Web Application Development using PHP & MySQL

Authors
Khirulnizam Abd Rahman, Hafiz Mohd Hanafi &
Che Wan Shamsul Bahri C.W. Ahmad

Web server: Apache

Server-side script: PHP

Database server: MySQL

Database administration: phpMyAdmin

Web Application Development using PHP & MySQL
Authors: Khirulnizam Abd Rahman, Hafiz Mohd Hanafi & Che Wan Shamsul Bahri C.W.Ahmad
Bandar Seri Putra, Bangi, Selangor, MALAYSIA

First published electronically: June 2007
Updated and Published in printed form: May 2016
Published by Faculty of Information Sciences & Technology
Selangor International Islamic University College
Bandar Seri Putra, Bangi, 43000 Kajang
Selangor, MALAYSIA.

Book Website http://kerul.net
Cover by Muizz Salleh

About the Authors

KHIRULNIZAM ABD RAHMAN is currently a Computer Science lecturer in the Faculty of Information Science and Technology, KUIS. He has been handling programming classes since 2000 (16 years). Graduated Bachelor in IT (Computer Science) Universiti Kebangsaan Malaysia (UKM) in 1999, and finishes Masters in IT (Computer Science) UKM in 2009. Web Programming is his favorite subject, and has been lecturing the subject since 2005. Java and .NET are also other programming languages that he is interested in. He has also published Android apps in the year 2010, and his early apps among others are Malay Proverb Dictionary (Peribahasa) and m-Mathurat. There are also Hijra of the Prophet Muhammad app in the Windows Phone store, and the latest apps published is SmartSolat.com.
Website: KERUL.net
Email: khirulnizam@gmail.com

HAFIZ MOHD HANAFI graduated from Universiti Teknologi Malaysia in Bachelor Electric and Electronic Engineering. He has master in Information Technology from Open University Malaysia. Since 2006, almost 10 years' experiences teaching in Web Programming and consult several industrial project. Currently he is working as a lecturer at Kolej Universiti Islam Antarabangsa Selangor. There are several companies he managed to consult in web base system and electronic circuit design and development. The projects he had consulted are Celcom Tower Security and Monitoring System, Automated Station Announcement using GPS for Keretapi Tanah Melayu (KTM) and latest project is Police Mobile Unit Monitoring Application for Royal Police of Malaysia. All those projects are collaborative efforts with GreenSilicon Technology.
Email: hafizhanafi@kuis.edu.my

CHE WAN SHAMSUL BAHRI BIN C.W.AHMAD graduated from Universiti Teknologi Malaysia (UTM) in Diploma in Computer Science and Bachelor in Science (Computer System). He has Master in Information Technology from Universiti Kebangsaan Malaysia (UKM). Since 2000, almost 16 years' experiences teaching in IT Subjects and consult several industrial project. Currently he is working as a lecturer at Kolej Universiti Islam Antarabangsa Selangor (KUIS). He also appointed as tutor at Open University Malaysia (OUM) Bangi and has experience in teaching and learning for Institut Komunikasi dan Elektronik Tentera Darat (IKED) with OUM program.
Email: cwshamsul@kuis.edu.my

Module Content: Web Application Development using PHP and MySQL

Chapter 01: PHP Introduction

Chapter 02: PHP Syntax

Chapter 03: Form Interactions

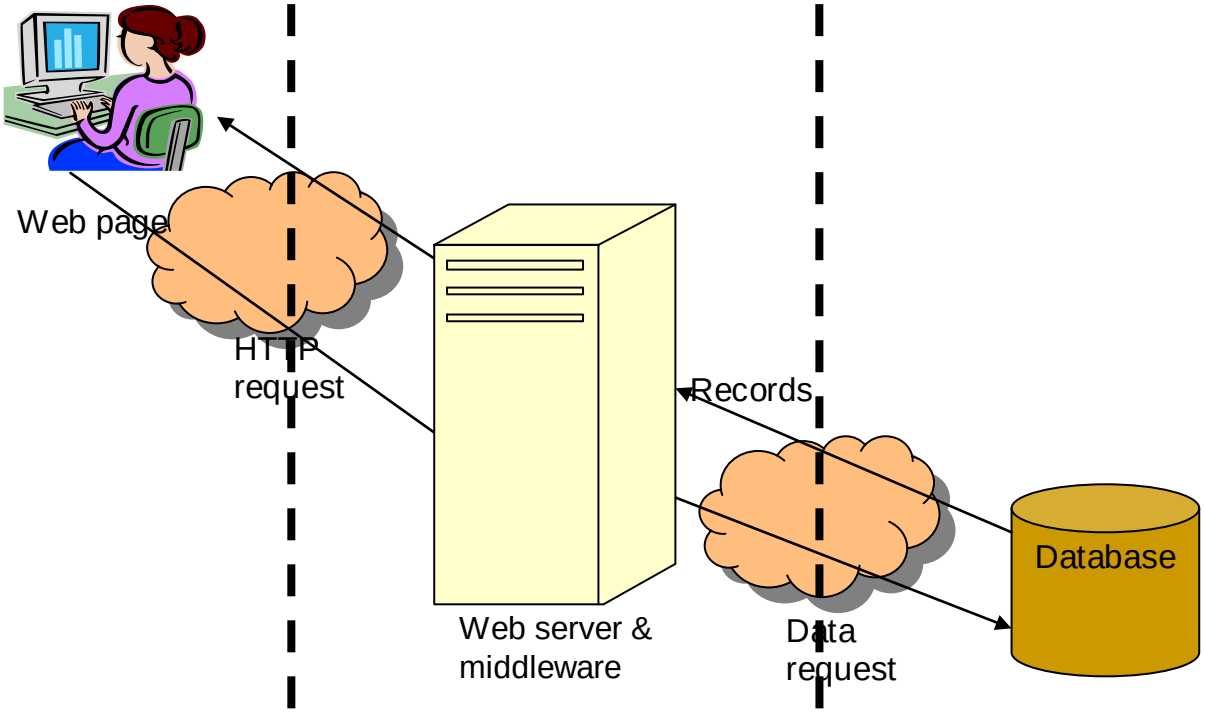
Chapter 04: PHP Operators

Chapter 19: File Uploads

introduction

3-tier Application Environment

- User interaction is separated from the database through a middleware that act as a connector.
- **Client :** Used to display the interface of the application. The interface downloaded form the server. Some logical operations done here. Eg: Web Browsers (IE, Firefox, Opera, Safari, etc)
- **Web Server+Middleware :** Store and host the interface. Provide the interface to the client. Most operations are on data retrieval and manipulations.
  Eg: Apache + PHP, IIS + ASP.
- **Database server :** Store and host data/records. Receive request and send the data/record to the web server. Eg : MySQL, SQL Server, PostgreSQL, etc…



What is dynamic website / active server pages?

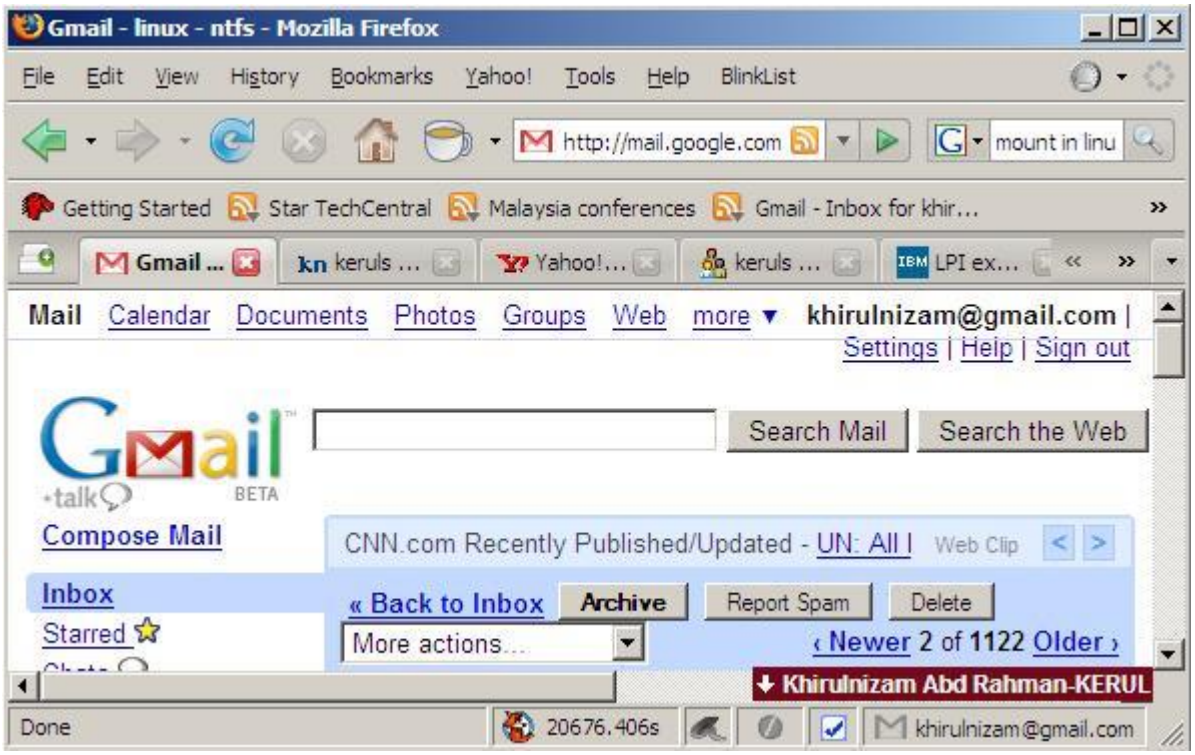
Web site that provide web pages that the content may change with regard to the change of the variables, such as users, browsers, geography, records in the database, etc…

What is a web server?

- Program / computer application that receive request from the browsers.
- Provide documents to requesting browsers.
- It's a slave program that only acts if a browser request.
- Example of web servers;
  - Apache
  - IIS
  - Xitami
  - Apache Tomcat
  - ColdFusion

What is a web browser?

- Applications that act as clients on the web.
- Request document from the web server.
- The server locates the document.
- The server sends the document to the web browser.



- Examples of web browsers;
  - Google Chrome.
  - Firefox from Mozilla.
  - Safari from Apple Inc.
  - Opera.

What is server-side scripting?

- The script embedded inside the HTML page, interpreted and executed inside the server (the web server) and the result/output of the execution is sent to the browser.
- The script interpreted and executed inside the server.
- The script won't be sent to the client.
- Example: PHP, ASP .NET, ASP, JSP, Perl, Phyton, etc…

What is client-side scripting?

- The script embedded inside the HTML page, downloaded with the HTML page and executed on the client (browser).
- JavaScript (EcmaScript), VBScript, Jscript.

What is web programming?

- Combination of server-side and client-side to develop a dynamic website. The application is deployed in the web environment and can be access by multiple concurrent users.

Exercise 1

1. What are the advantages and disadvantages of developing a web based application? List and explain five advantages and five disadvantages.
2. Differentiate between server-side and client-side scripting. Explain five items.

A:. INTRO TO PHP

What is PHP?

- PHP stands for PHP Hypertext Preprocessor.
- Used to develop dynamic website
- It is a server-side language.
- The PHP script embedded in the web page.
- The script is run (interpreted) on your web server, and the output from the process is inserted to the web page as a part of the content before transmitted to the browser.
- The script won't be sent to the browser which request the web page, so the script is not visible to the users.
- It supports many database management system (Oracle, Postgre, DB2, Microsoft SQL Server, etc).
- And it's free and powerful, that's why PHP is very popular.

B:. THE PHP DEVELOPMENT TOOLS

by apachefriends.org

The text will refer the XAMPP compilation tools for web development using Apache as the web server, PHP as the server script interpreter and MySQL as the database server. These three major tools are provided in the XAMPP and available for you in a single installation.

Installing XAMPP as the development tool.

- Developed by a group of volunteer programmer as a tool to develop web application using os/fs on Windows.
- * in latest Linux distribution, Apache web server, MySQL database, and PHP interpreter are installed as default for the server edition.

- Before this, developer has to download and install 3 major applications for developing web application using os/fs tools.
- (Apache web server, MySQL database, and PHP interpreter)
- phpDev combines three of them (plus few aditional software) in one installation.
- download this compilation at http://www.apachefriends.org/en/xampp-windows.html

Running XAMPP

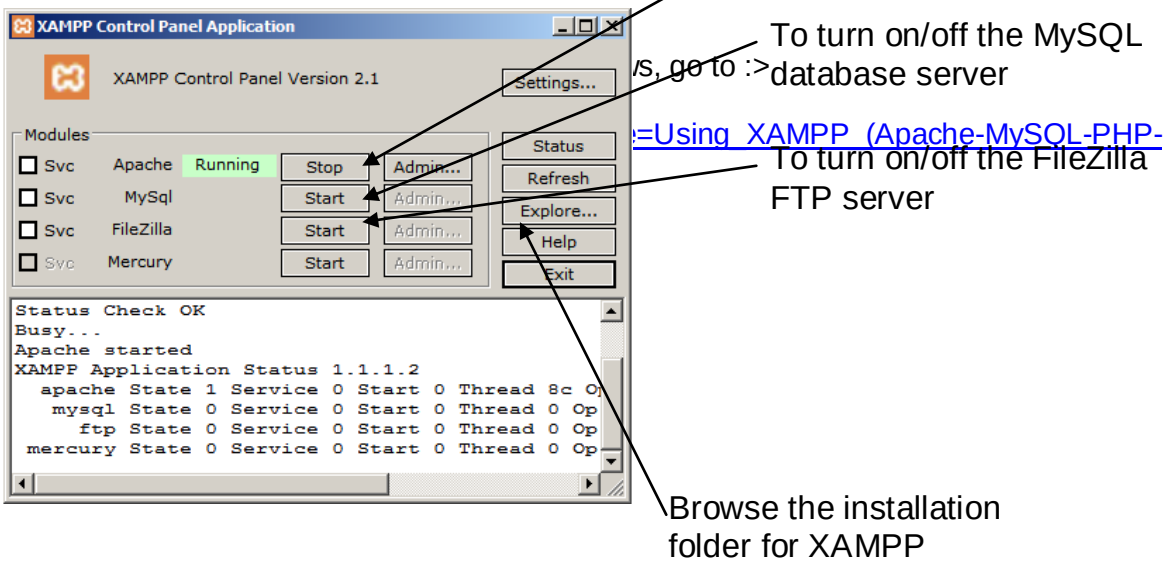
- After you downloaded and installed XAMPP for Windows on your system, turn on all the server needed by clicking the
  Start>Programs>XAMPP>Control Server Panel

***Pls stop the IIS service if you have installed it. Because Apache server is using port 80 by default (similar as IIS).

- After that, start the Apache server to use the web server application.
- If you need to use MySQL, start the MySQL service.
- The testing page will appear (if the installation succeed).
- The web root is
  c:\Program Files\xampp\htdocs
- Save all your php files inside
  c:\Program Files\xampp\htdocs\yourFolder

to create your own web application project.

XAMPP Control Server Panel



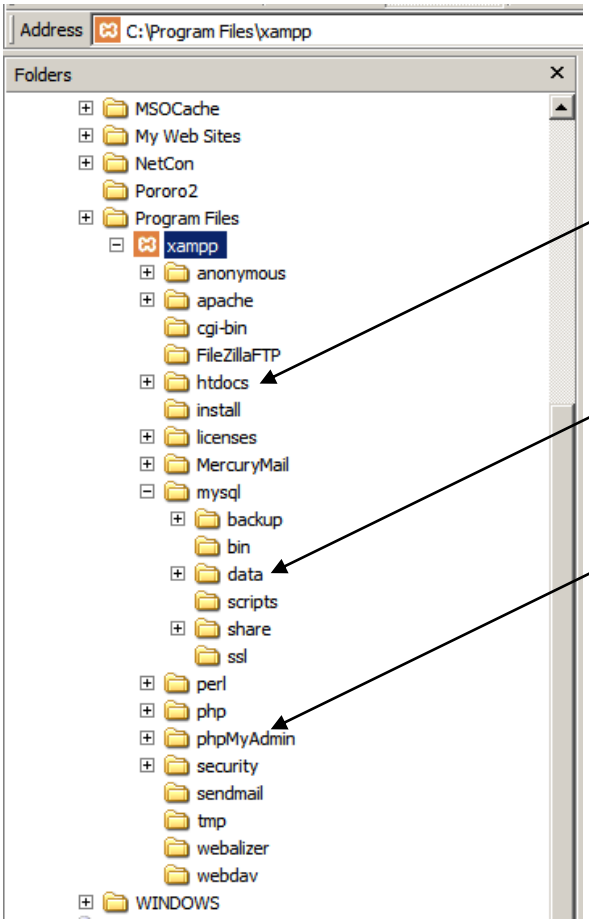
To turn on/off the Apache web server

To turn on/off the MySQL database server

To turn on/off the FileZilla FTP server

Browse the installation folder for XAMPP

XAMPP application folder



htdocs – the web root (this is where we dump all the HTML pages and the PHP scripts)

mysql/data – where MySQL store the database

phpMyAdmin – the web based MySQL administrator

C:. TESTING YOUR FIRST PHP SCRIPT

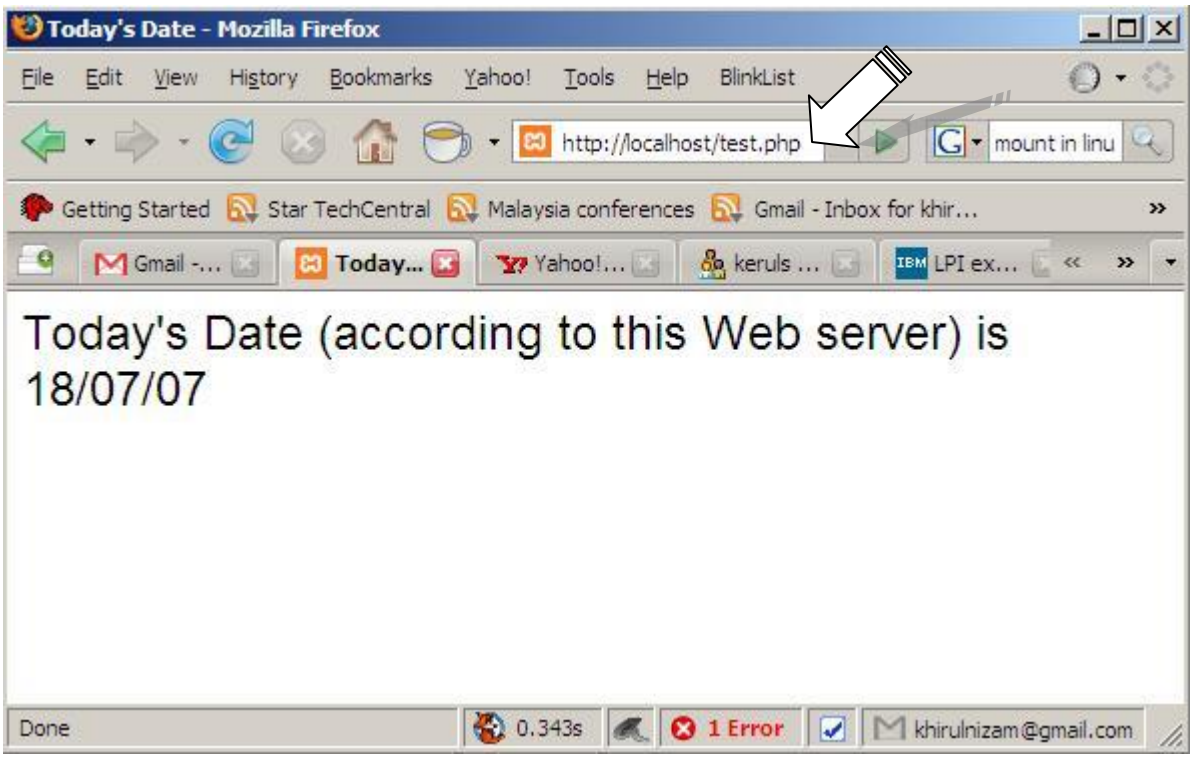
Inserting PHP script inside HTML document.
The script below will display today's date as available in the computer system.

```
<html>
<head><title>Today's Date</title></head>
<body>
Today's Date (according to this Web server) is <br>
<?php
      echo( date("d/m/y") );
?>
</body>
</html>
```

- Use your favorite web page editor such as Dreamweaver , or Notepad, or PHPDev (phpdev.org), or Eclipse for PHP (easyeclipse.org). From my own experience, FrontPage doesn't support server side scripts very good. If you need to do PHP development under Visual Studio, there is a PHP plugin you can download from .
- Save the file inside the webroot of your web server as **test.php**. In this case, we are using the XAMPP compilation. So by default the webroot is in
  c:/Program Files/Xampp/htdocs

Output:
Go to your favorite web browser and type the address of your localhost plus the filename in the address box. It would be;

http://localhost/test.php



References :

- Sebesta, Robert W. 2003. *Programming the World Wide Web*. Pearson Education, Inc.
- http://wikipedia.org
- http://apachefriends.org
- http://dhost.info/kerul/webdev/

syntax

Objectives: This chapter will cover the syntax (grammar) of PHP language. To learn a new programming language, we should know the syntax first. After that only we can master the language.
Topics that we're going to discuss are;

A:. PHP script tag.
B:. Comment.
C:. Statements.
D:. Expressions.
E:. Keyword/reserved words.
F:. Variable,
G:. Data types.
H:. Lateral value.
I:. Printing Messages.
J:. Concatenation.
K:. Simple debugging technique.

PHP SCRIPT TAG

PHP scripts is a segment of program written in the HTML file. It is embedded inside the HTML code in order to program the HTML page to act accordingly to the dynamic changes set by the programmer.

<?php ?> is the tag to segregate the PHP script from the rest of the HTML tags. Between the '<?php' and '?>' lies the PHP script.

```
Example :

<html>
<head>
<title>Today's Date</title>
</head>
<body>
Today's Date (according to this Web server) is <br>
<?php
     print("Hello World!!<br>");
     echo( date("d/m/y") );
?>

</body>
</html>
```

php script inside a HTML page

B:. COMMENTS

- Comment is code that will not be compiled (ignored when the program is compiled).
- It will not effect the code.
- It is a good practice to write the comment.
- Comment is useful the tell others what is the programming doing.
- Helps programmer to memorize what is he doing.

3 types of comments.

1. Line comment.
   Symbol : //
   The statements written after the double slash will be ignored.

```
Example:
    //display variable value
    $num1=70;
    echo $num1;
```

2. Block comment – to comment statements or programmers' remark for more that one lines of codes.
   Symbol : /*……*/

Example:

```
<?php
/*    Author      : Khirulnizam Abd Rahman
      Email       : khirulnizam@gmail.com
      Website     : http://dhost.info/kerul/webdev/
      Purpose of script
        Setting the connection for database host
*/
      $db=mysql_connect("172.16.12.1 ", "root ", "kuis123");
?>
```

C:. STATEMENTS

Any statements must be stopped by a semicolon ';'. PHP interpreter will generate an error message if the statement is not ended with the semicolon, the compiler will tell you there is syntax error.

```
Example:

$huruf='a';
echo ("Hello $nama, you're entering PHP world.");
```

D:. EXPRESSIONS

Expression is a statement that produces value.

```
Example:

      $jumlah = $hasil1 + $hasil2;
      if ($jumlah == 0)
      while ($n < 10)
```

E:. VARIABLE

- Memory location to hold a value (a number, string, object, array, etc).
- PHP is a **loosely typed** language.
- means that a single variable may contain any type of data, (integer, floating value, string) In the process, the variable changes type: where it used to contain a number, it
- The same variable can hold different data type at different time
- No need to declare
- Must have dollar sign ($) in front of variable

Eg :

```
$testvariable = "Three";
$testvariable = 3;
```

Variable Naming Rules

A variable name must start with a dollar sign ($).

- A variable name can only contain alpha-numeric characters and underscores (a-Z, 0-9, and _)
- A variable name should not contain spaces. If a variable name should be more than one word, it should be separated with underscore ($my_string), or with capitalization ($myString)

Examples of valid variable name.

1. $nom2
2. $jumlah
3. $huruf
4. $_nama
5. $for
6. $FLOAT
7. $duitringgit
8. $makanApa
9. $myName
10. $tiada2_

Examples of invalid identifier.

1. 2nom (begins with numeric)
2. $jumlah nombor (there are space [special character])
3. $1234 (all characters are numeric)
4. $kerul@hotmail (@ is invalid character)
5. $too.much (‘.’ is invalid character)

F:. KEYWORD/RESERVED WORD

- Words that are used as a special instruction in the programming language.
- These words have special meaning in PHP. Some of them represent things which look like functions, some look like constants, and so on--but they're not, really: they are language constructs. You cannot use any of the following words as constants, class names, function or method names. Using them as variable names is generally OK, but could lead to confusion.

Table K.1. List of PHP Keywords

And	or	xor	__FILE__	
__LINE__	**array()**	as	break	case
class	const	continue	declare	default
Die()	do	**echo()**	else	elseif
empty()	enddeclare	endfor	endforeach	endif
endswitch	endwhile	**eval()**	**exit()**	extends
For	foreach	function	global	if
include()	**include_once()**	**isset()**	**list()**	new
print()	**require()**	**require_once()**	**return()**	static
switch	**unset()**	use	var	while
__FUNCTION__	__CLASS__	__METHOD__		
PHP 5 Only		exception	final	php_user_filter
interface	implements	extends	public	private
protected	abstract	clone	try	catch
throw				
PHP 4 Only		cfunction	old_function	this

G:. LATERAL VALUES

- Lateral is the exact/actual value (what you see is what you get).
- The value could be number(integer or real), character, string, or Boolean.

Integer

- Integer is round number.
- For example, 10 is an integer.

```
Example ;
      $nomborBesar = 10;
```

- Use negative sign to for negative number.

```
Example ;
      $num = -10;
```

Float

```
Example :
      $temp = 37.98789;
      $product = 0.0;
```

Characters

- To assign a character value into a *char* variable, the value must be put inside a open and close quote.
  Example : 'A', '?', '7'

```
Example :
      $huruf = 'b';
      $grade= 'A';
```

These are the list of hidden character in PHP (similar like in C).

\n	New line
\t	Tab (a tab space)
\b	Backspace
\'	Single quote
\"	Double quote

Boolean

- There are only two values for Boolean variable,
- **true** and
- **false**

```
Example :
      $eatalready = true;
      $isInteger = false;
```

String

- A group of characters.
- Marked with double quote ("……").

```
      Example:
            "123434"
            "T"
            "I love PHP"
            "\tPHP tak suka saya."
            "\n I am using PHP4"
            "We need PHP to do web programming."
            "This string will \n be printed \n in 3 lines"
```

H:. PRINTING MESSAGES

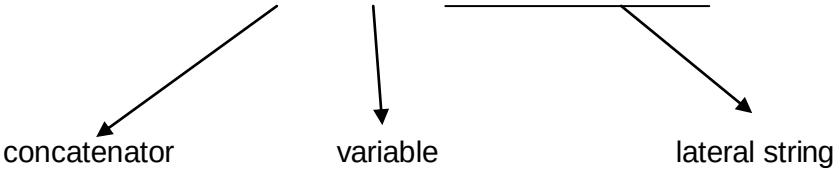
<?
$welcome_text = "Hello and welcome to my website. ";
print $welcome_text;
?>
We can also use echo

<?
$welcome_text = "Hello and welcome to my website.";
echo $welcome_text;
?>

I:. CONCATENATION

- Concatenation is to join(or to merge) several strings into one.
- In PHP we usually use "." (dot) to do concatenation.

```
<?
      $name = "Kerul";
      $age=21;
      echo "My name is ".$name. "<br>";
      echo "I\'m ".$age. "years old…<br>";
?>
```



Simpler way of doing concatenation…

Used to merge a lateral string and value of a variable.

```
<?
      $name = "Kerul";
      $age=21;
      echo "My name is $name <br>";
      echo "I am $age years old…<br>";
?>
```

Example:

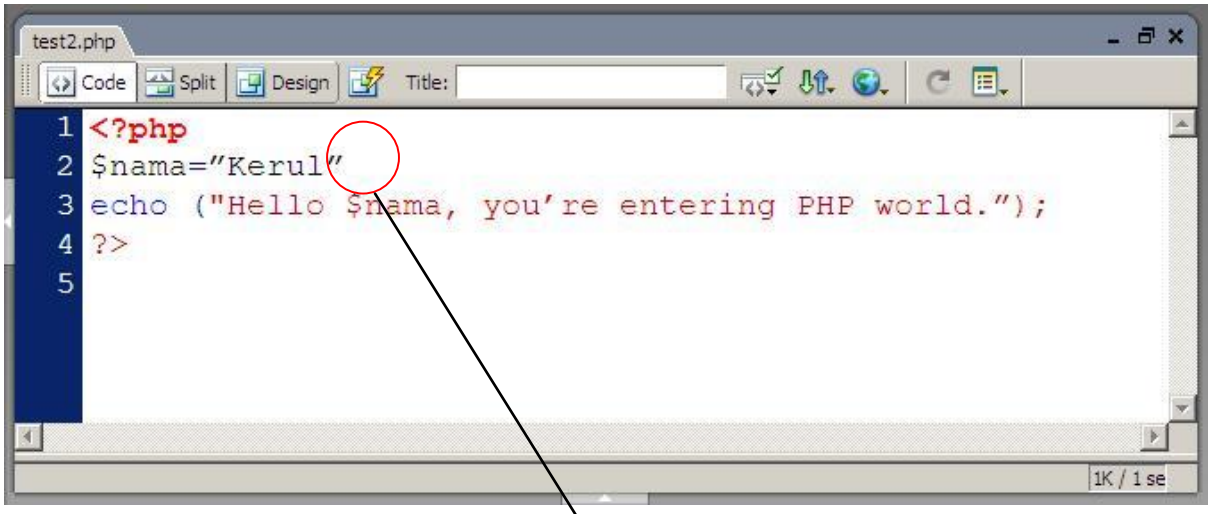
```
<html>
<head>
<title>Variable Testing </title>
</head>
<body>
This is the second test<br>
<?php
      $name = "ur name";
      $age=21;
      echo ("My name is $name <br>");
      echo ("I\'m $age years old…<br>");
?>
</body>
</html>
```

J:. SIMPLE DEBUGGING TECHNIQUE

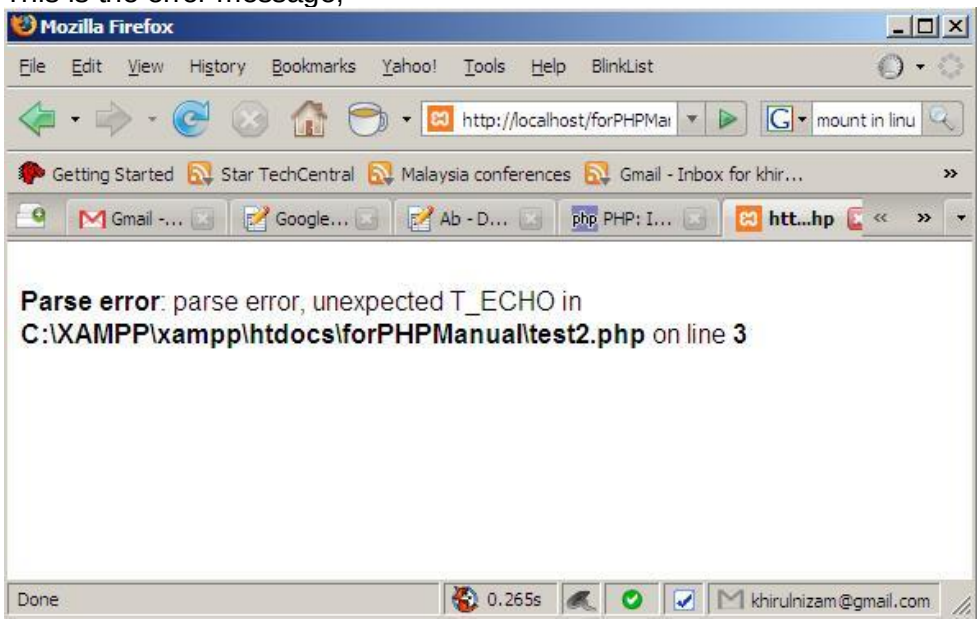
If let say, you forgot to put the semicolon, there will be an error message. You need to find and correct the error, save the script file, and refresh the page in the browser.

Example : this is the script with error.

```
<?php
      $nama=”Kerul”
      echo ("Hello $nama, you’re entering PHP world.”);
?>
```



This is the error message;



Go to the script editor, and add a semicolon at the end of the variable initialization. Save the script and refresh the content in the browser. The same method goes to any error occur during the coding. Good luck, you need it very much.

Exercise 2

1. Decide whether the following variables are **valid** or **invalid**. Give your reason.
   a) $apa
   b) $123lekuk
   c) $what-is-the-number
   d) $_uji1
   e) $nama saya
   f) $For
   g) IF
   h) $float
   i) $variable_to_hold_names_or_matrix_for_students
   j) $hasil*nombor

2. Decide whether the following initializations are **valid** or **invalid**. Give your reason.
   a) $nom1 = 1231234;
   b) $num3 = 23.7684;
   c) $var1 = 123.0;
   d) $dah_makan = TRUE;
   e) $huruf = 't';
   f) $alphabet = "B";
   g) $pi = 0.0D;
   h) $besar =12345674567L;
   i) $Lekuk123 = 12.5f;
   j) $name = "Mamat';

Please refer the the declaration below to answer question 3 and 4.

```
$nom1 = 10;
$nomPerpFLOAT = 10.0055;
$nama_saya = "CRACKER";
```

3. Write the output for the following statements.
   a) echo ("Saya baru nak belajar PHP.");
   b) echo ("Hari ini saya bangun pukul " . $nom1 . " pagi.");
   c) echo (" Where do you wanna go, ". $nama_saya);
   d) echo ("Welcome to PHP \n world");

4. Write the *echo( )* statement to display the following message. Use only one *echo( )* function to display the message. Replace the value with the variable name from the above declaration.
   a) Saya suka nombor **10**.
   b) My nick name is **CRACKER**.
      How about yours?
   c) Casting is needed to do this operation : **10/10.0055**.

form interactions

Objectives:

1. To introduce the concept of passing users' value in the web programming environment.
2. To create form to receive user's input, and target file in order to process the information sent to the server.

Typing the user input in a form is the most common way to interact with the script inside a web server. There is also another way to send data, by embedding the data inside the URL.

Input Through the HTML Form using GET method.

```
<form name="form_name"  method="send_method" action="target_file">

        inputs….

</form>
```

This is the typical structure of a form. It contains name, method and action as the attributes.

- **name** : is the name of the form.
- **method** : how the data is transferred (using POST or GET), in this case use GET.
- **action** : is the target file resides in the web server that receive the data from this form.
- inputs : are the input elements (eg: text box, button, radio, checkbox, etc)

Text box

```
<input   type="text"   name="firstname">
```

First name:

Last name:

- This is the tag for most of the input.
- The attribute 'type' defines what type of input element to be displayed to the user in the web page. In this case, the input type is text box.
- The name is a very important attribute where it represents the value entered by the user.
- The main input element used in form is text box, and followed by a submit button to transmit user's information to the server.
- The text box is capable of receiving single-line string, and may contain any character.
- The data is sent to the server and received by the target file, which is defined in the form's action.

Submit button

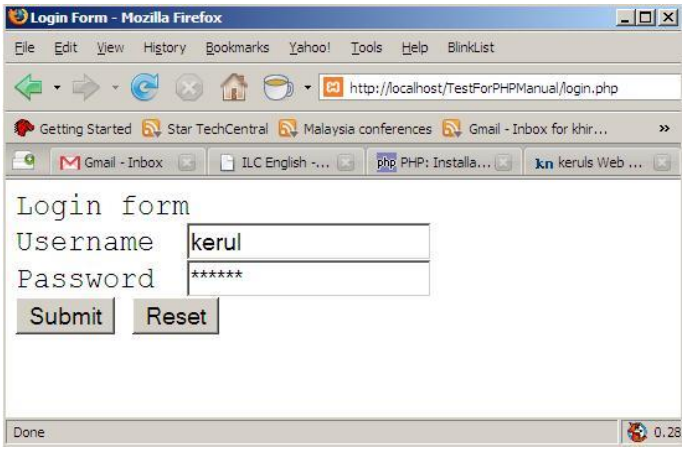
```
<input  type="submit"  value="Submit">
```

- The submit button is normally used to activate the process of sending/ transmitting the data to the target file (to the server).
- Make sure the 'type' is **submit**. The attribute 'value' will be shown as the caption of the button.

EXAMPLE 1

Step 1.Create a HTML page with a form to receive username and the password of a dummy system. Save the file as *login.php*.

```
<html>
<head><title>Login Form</title></head>
<body>
Login form <br>
<form name="frmLogin"  method= "get"  action="displayInfo.php">
  Username
        <input name="txtUsername"  type="text" size="20"><br>
  Password
        <input name="txtPassword"  type="password"  size="20"><br>
        <input name="btnSubmit"  type="submit"  value="Submit">
        <input name="btnReset"  type="reset"  value="Reset">
</form>
</body>
</html>
```

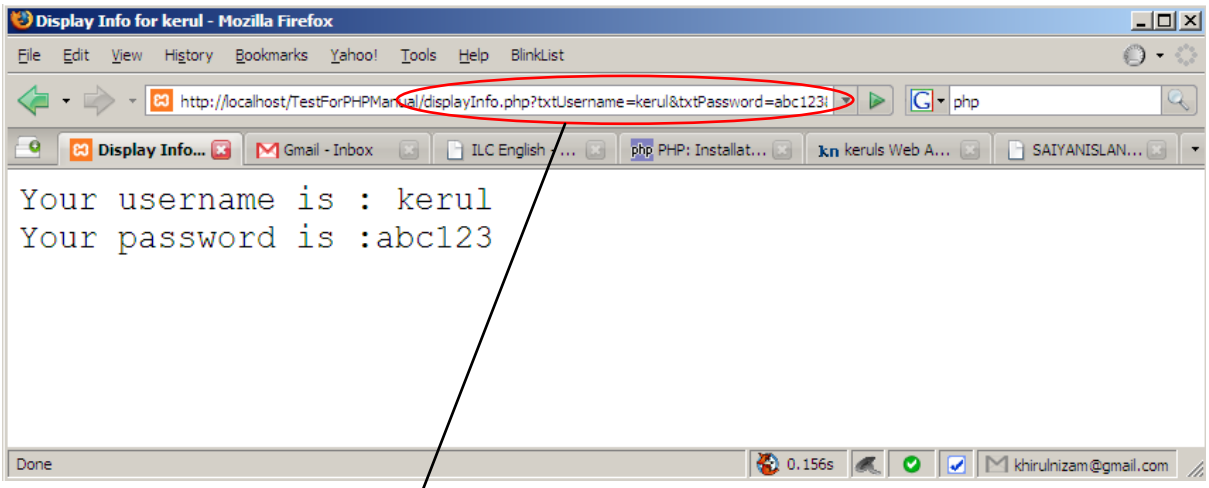


Step 2.Create a target file to receive username and the password from the form. The information will be extracted and displayed to the user.

This is the target file, named *displayInfo.php*.

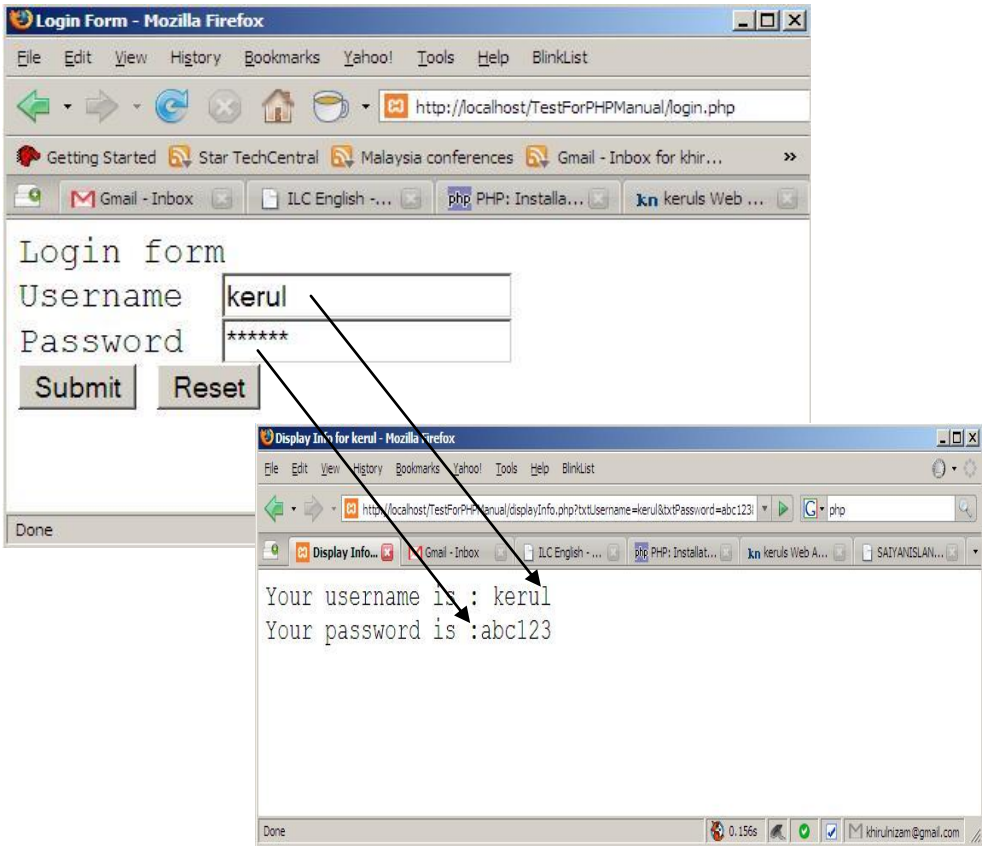
```
<html>
<head>
<?php
        $username=$_GET["txtUsername"]; //username is extracted from the querystring
        $pword=$_GET["txtPassword"];  //password is extracted from the querystring
?>
<title>Display Info for <?php echo $username;?> </title>
</head>
<body>
Your username is : <?php echo $username;?> <br>
Your password is :<?php echo $pword;?> <br>
</body>
</html>
```

*Observe closely how the username and the password entered by the user is extracted by the $_GET server variable.



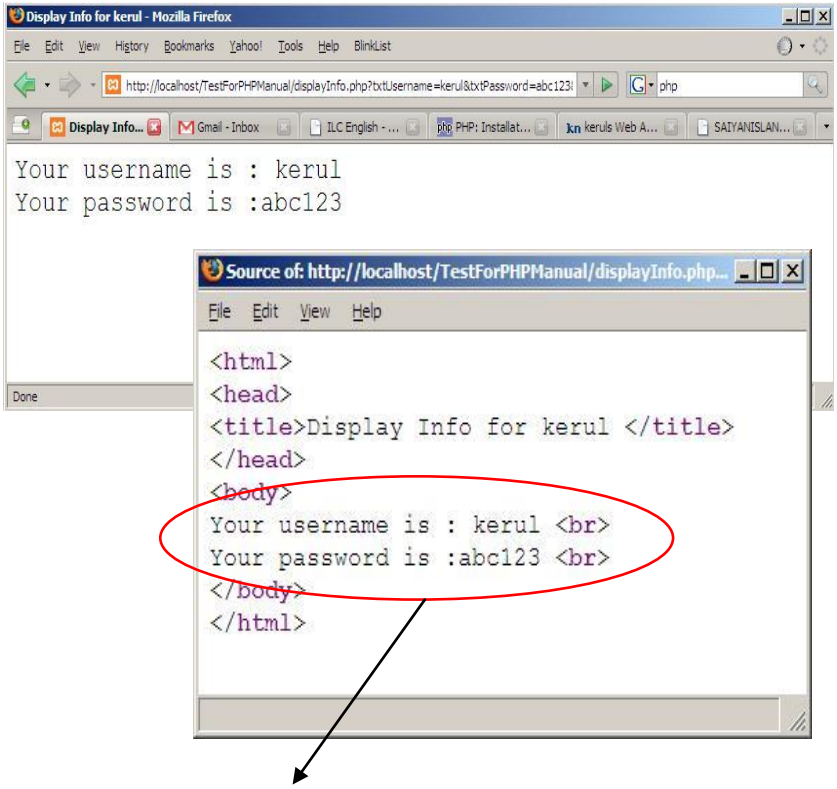
The query string enlarged:
http://localhost/TestForPHPManual/displayInfo.php?**txtUsername=kerul**&
txtPassword=abc123&btnSubmit=Submit

See how the pages exchange the values of the user's input. The values keyed-in by the user in the first page (the page with form – *login.php*). The form transmits the values through the query string and the second page (*displayInfo.php*) extracts the values using $_GET, and display them.



Since PHP is a server-side scripting, it's interpreted/executed inside the server. The output of the script execution is embedded inside the page which contains the script. At the end the process, the HTML page, together with the output generated, is sent to the browser (client). The script will not be included in the HTML page sent to the browser.

Try to view the HTML source of the *displayInfo.php* from the browser. You will not see the PHP script. Compare the HTML source with the page with the PHP script in page 2. *Do they have any differences?*



See! No PHP script in the HTML page source!
WHY???

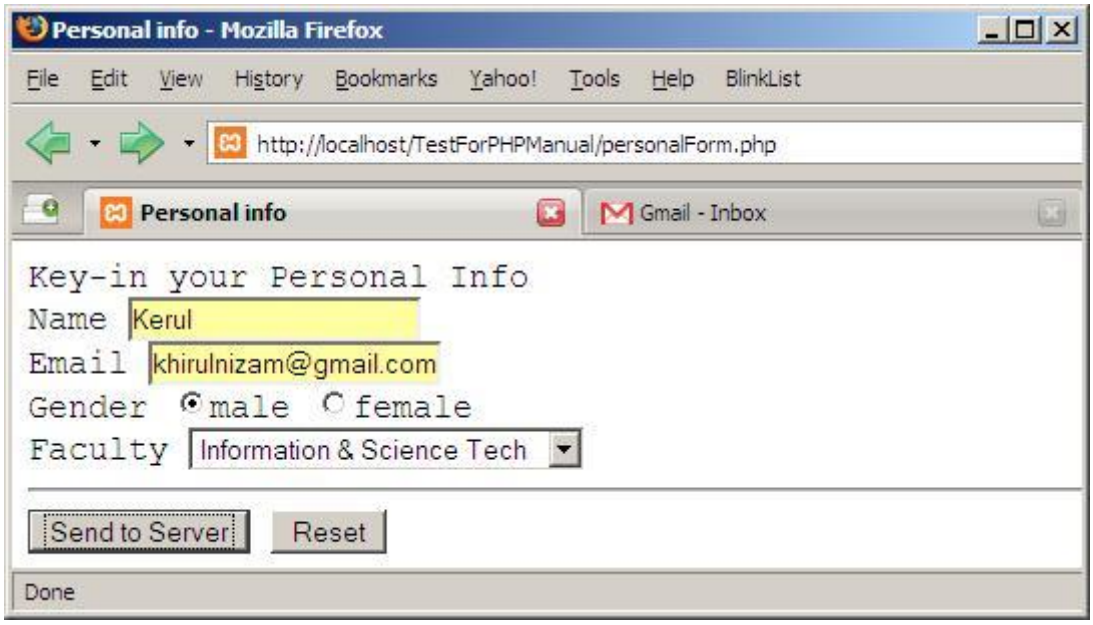
EXAMPLE 2
(This is an example with an error in the PHP script.)

Step 1.Create a HTML page with a form to receive user's information. Use the following HTML codes.

Step 2.Save as *personalForm.php*, and this page will be sent to another page with the name processPersonal.php as mentioned in the form's action.

```
<html>
<head><title>Personal info</title></head>
<body>
Key-in your Personal Info<br>
<form name="frmInfo" method="get" action="processPersonal.php">
  Name<input name="txtName"  type="text" ><br>
  Email <input name="txtEmail" type="text"  ><br>
  Gender        <input name="rGender"  type="radio" value="male">male
                <input name="rGender"  type="radio" value="female">female<br>
  Faculty       <select name="cmbFaculty">
                        <option value="FTSI">Information & Science Tech</option>
                        <option value="FPM">Management & Muamalah</option>
                        <option value="FPPI">Islamic Studies</option>
                        <option value="FBK">Language & Communication</option>
                </select>
          <hr>
                <input name="btnSubmit"  type="submit"  value="Send to Server">
                <input name="btnReset"  type="reset"  value="Reset">
</form>
</body>
</html>
```

This is the output of the HTML code above.

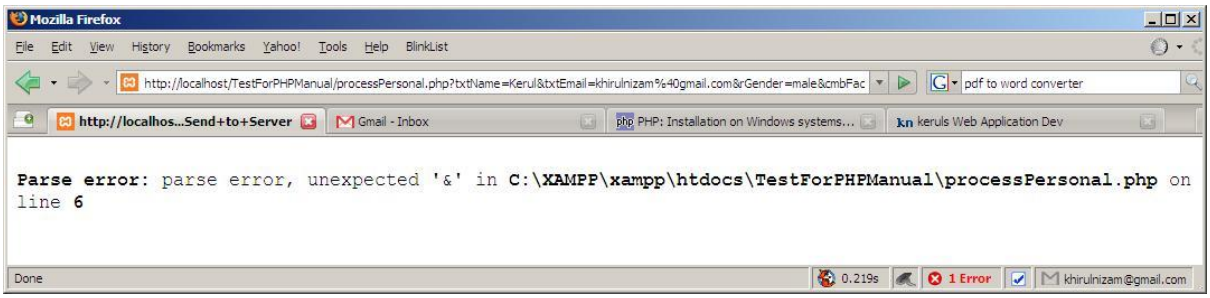


Step 3.Create another page and write the code.

Step 4.Save the page as *processPersonal.php*.

```
1.  <html>
2.  <head>
3.  <?php
4.      $name=$_GET["txtName"];
5.      $email=$_GET["txtEmail"];
6.      &gender=$_GET["rGender"];
7.      $faculty=$_GET["cmbFaculty"];
8.  ?>
9.  <title>Display Personal Info for <?php echo $name;?></title></head>
10. <body>
11. <b>These are my personal details</b>
12. <hr>
13. My name is <?php echo $name;?>.<br>
14. I am <?php echo $gender;?>.<br>
15. I am a student of faculty <?php echo $faculty;?>.<br>
16. Do contact me for any comment...<a href="mailto:<?php echo $email;?>">
17. <?php echo $email;?> </a>
18. </body>
19. </html>
```

Step 5.When the page is viewed in the browser, there's a massage saying that the page (processPersonal.php) has an error in **line 6**.



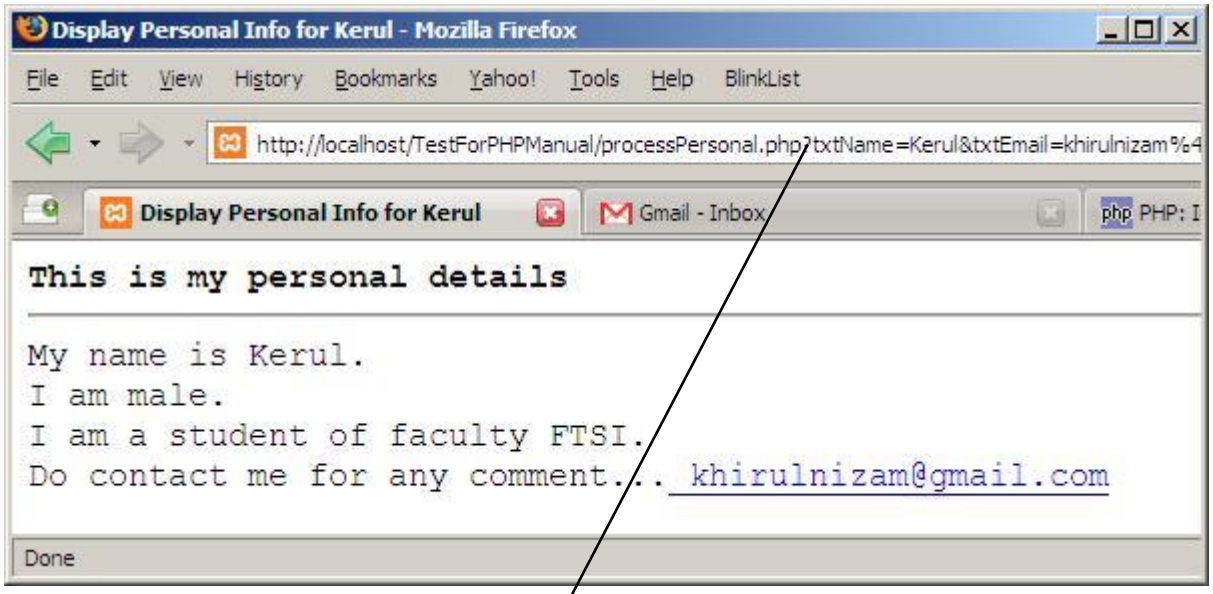
Step 6.Open the processPersonal.php in the code editor (Dreamwever) and go to the **line 6**, and fix the error. There are not suppose to have '&' before *gender*. Change the '&' into '$'.

```
1.  <html>
2.  <head>
3.  <?php
4.      $name=$_GET["txtName"];
5.      $email=$_GET["txtEmail"];
6.      &gender=$_GET["rGender"];
7.      $faculty=$_GET["cmbFaculty"];
8.  ?>
9.  <title>Display Personal Info for <?php echo $name;?></title></head>
10. <body>
11. <b>These are my personal details</b>
12. <hr>
13. My name is <?php echo $name;?>.<br>
14. I am <?php echo $gender;?>.<br>
15. I am a student of faculty <?php echo $faculty;?>.<br>
16. Do contact me for any comment...<a href="mailto:<?php echo $email;?>">
17. <?php echo $email;?> </a>
18. </body>
19. </html>
```

This is the error, change to $ as the beginning of any variable.

Step 7.Save the file.

Step 8.Go to the web browser, and refresh the page.



The query string enlarged:
http://localhost/TestForPHPManual/processPersonal.php?**txtName=Kerul**&**txtEmail=khirulnizam%40gmail.com**&**rGender=male**&**cmbFaculty=FTSI**&btnSubmit=Send+to+Server

Step 9.Observe at the address bar, the querystring is much longer than the querystring in Example 1. **Why is this happening?**

Input Through the HTML Form using POST method.

EXAMPLE 3

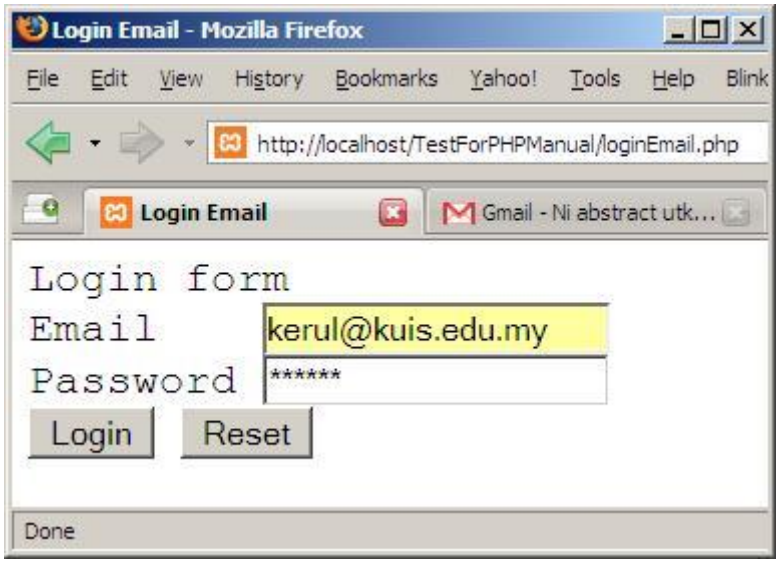
Step 1.Create a HTML page with a form to receive email address and the password. Write the following HTML codes and save as *loginEmail.php*. Make sure the method in the form is POST.

```
<html>
<head><title>Login Email</title></head>
<body>
Login form <br>
<form name="loginEmail"  method= "POST"  action="checkPassword.php">
  Email
        <input name="txtEmail"  type="text"><br>
  Password
        <input name="txtPassword"  type="password">
              <br>
        <input name="btnSubmit"  type="submit"  value="Login">
        <input name="btnReset"  type="reset"  value="Reset">
</form>
</body>
</html>
```

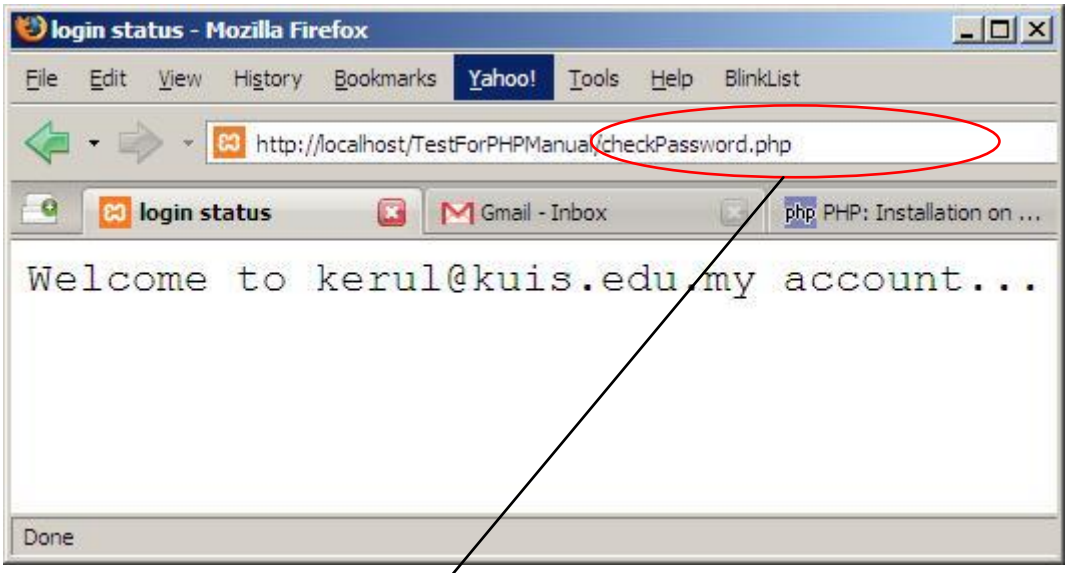
Step 2.Create a another HTML page with the PHP script to retrieve the user's email and password.. Write the following HTML codes and save as checkPassword.*php*. Observe closely how the user's email and password are retrieved.

```
<html>
<head>
<?php
        $email=$_POST["txtEmail"]; //email is extracted
        $pword=$_POST["txtPassword"];  //password is extracted
?>
<title>Login status </title>
</head>
<body>
<?php
        $systemPassword="abc123";

        if ($pword==$systemPassword){ //case 1: user password matches system
password
              echo "Welcome to $email account...<br>";
        }
        else{ //case 1: user passwrod do not match system password
              echo "Password does not match...<br>";
        }
?>
</body>
</html>
```

Step 3.Go to your browser and open the file for the form. Key in your email address, and the password (use *abc123*), and click the *Login* button.



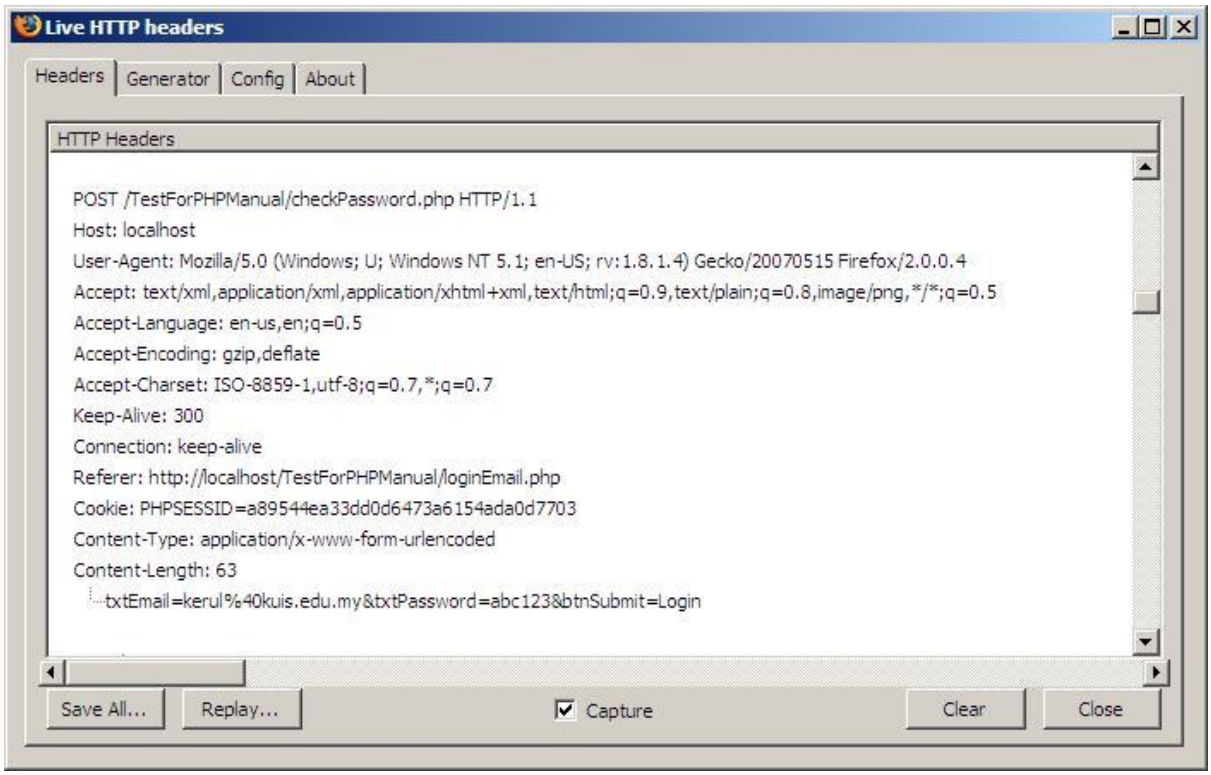
Step 4.If you keyed-in the right password (abc123) then you will get this page.



The values from the previous form (in step 3) do not appear here. This is one of the advantages by using POST method to submit confidential data. No one get to see the values in the URL*.

Live HTTP headers

Although the values are not included in the URL, still this is not the most secure internet transaction. It still visible if you have the right tools to uncover the values transmitted through POST method. Example: use the Live HTTP Headers from http://livehttpheaders.mozdev.org/ to view all the information submitted to the server. Refer to the next page.



Differences between GET and POST method.

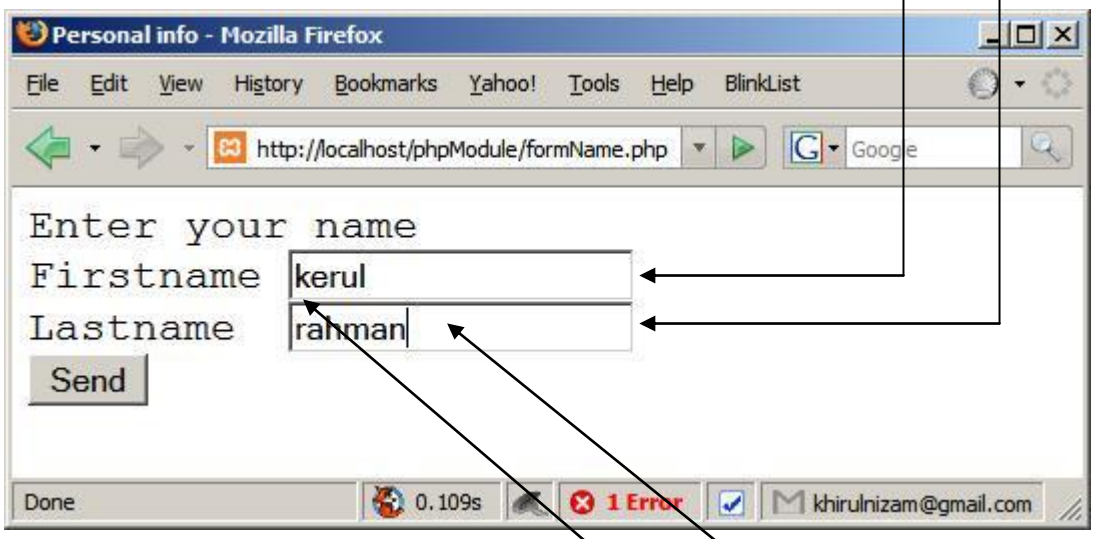
(Extracted from: http://www.cs.tut.fi/~jkorpela/forms/methods.html)

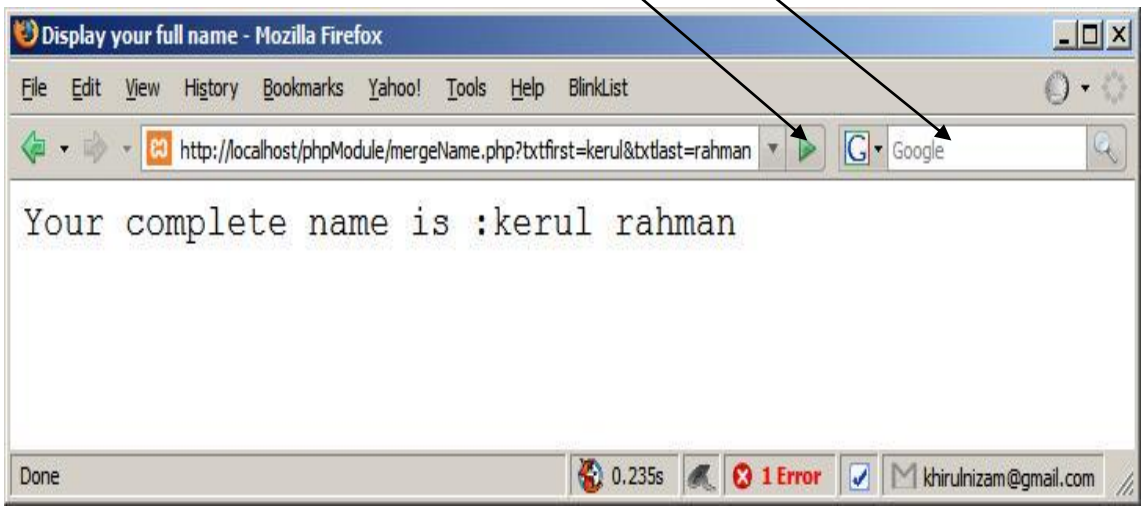
- The HTML specifications have *technically* define the difference between "GET" and "POST"
- **GET** is basically for just getting (retrieving) data.
- The variable name and the value appear in the URL and they are visible to the naked eyes.
- GET - form data is to be encoded (by a browser) into at the end of the URL of the target file.
- **POST** - form data is embedded within a message body.
- POST may involve anything, like storing files to the server, updating data, ordering a product, or sending an e-mail.
- The variable name and the value do not appear in the URL.

Extracting the form's information using $_GET

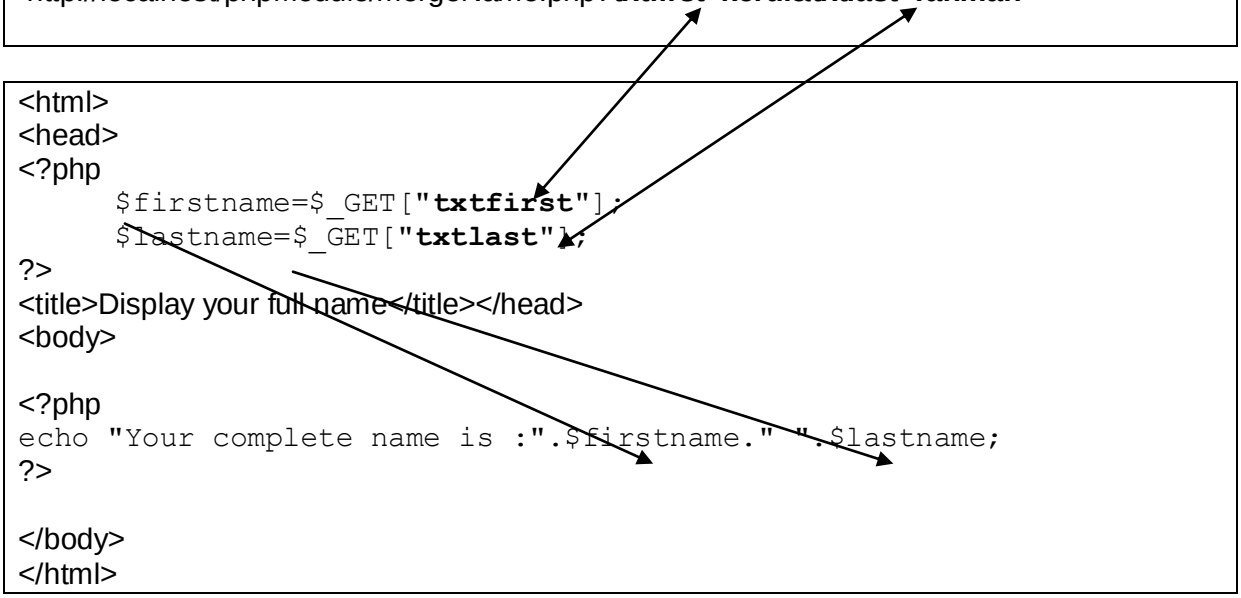
When a form is submitted to the target file, the input element names and their respective values are embedded in the URL to form a querystring. The query string is ????.

```
<html>
<head><title>Name</title></head>
<body>
Enter your name<br>
<form name="frmName" method="get" action="mergeName.php">
  Firstname  <input  name="txtfirst"  type="text"><br>
  Lastname  <input  name="txtlast"  type="text"><br>
          <input name="btnSubmit" type="submit"  value="Send">
</form>
</body>
</html>
```





The query string enlarged:
http://localhost/phpModule/mergeName.php?**txtfirst=kerul**&**txtlast=rahman**



```
<html>
<head>
<?php
      $firstname=$_GET["txtfirst"];
      $lastname=$_GET["txtlast"];
?>
<title>Display your full name</title></head>
<body>

<?php
echo "Your complete name is :".$firstname." ".$lastname;
?>

</body>
</html>
```

Displaying variable's value using echo

There are many functions capable of displaying (generating output) in PHP. The most popular function is echo. Below are few descriptions taken from the official PHP Manual about the echo. (Adapted from http://php.net/echo)

Echo outputs all parameters.

Examples

```
<?php
echo "Hello World";

echo "This spans
multiple lines. The newlines will be
output as well";

echo "This spans\nmultiple lines. The newlines will be\noutput as
well.";

echo "Escaping characters is done \"Like this\".";

// You can use variables inside of an echo statement
$foo = "foobar";
$bar = "barbaz";

echo "foo is $foo"; // foo is foobar

// Using single quotes will print the variable name, not the value
echo 'foo is $foo'; // foo is $foo

// Some people prefer passing multiple parameters to echo over
concatenation.
echo 'This ' . 'string ' . 'was ' . 'made ' . 'with concatenation.'
. "\n";
?>
```

References

1. Live HTTP headers is provided by mozdev.org and can be downloaded freely at http://livehttpheaders.mozdev.org/
2. The article on Differences between GET and POST provided by Jukka "Yucca" Korpela, and available from http://www.cs.tut.fi/~jkorpela/forms/methods.html
3. The official documentation (help/manual) is available from http://php.net. Simply add the function needed for elaboration at the end of the address. Example php.net/echo to know more about echo, how to use it, some examples, and users' contributed notes.

EXERCISE 3

Exercise 3.1

You're given this code for the form (the first page), so the user could key-in the book information needed. If the user clicks the "Save Book" button, the information will be sent to another file named *displayBookInfo.php*. Retrieve the values from the form, and display them on the second page.

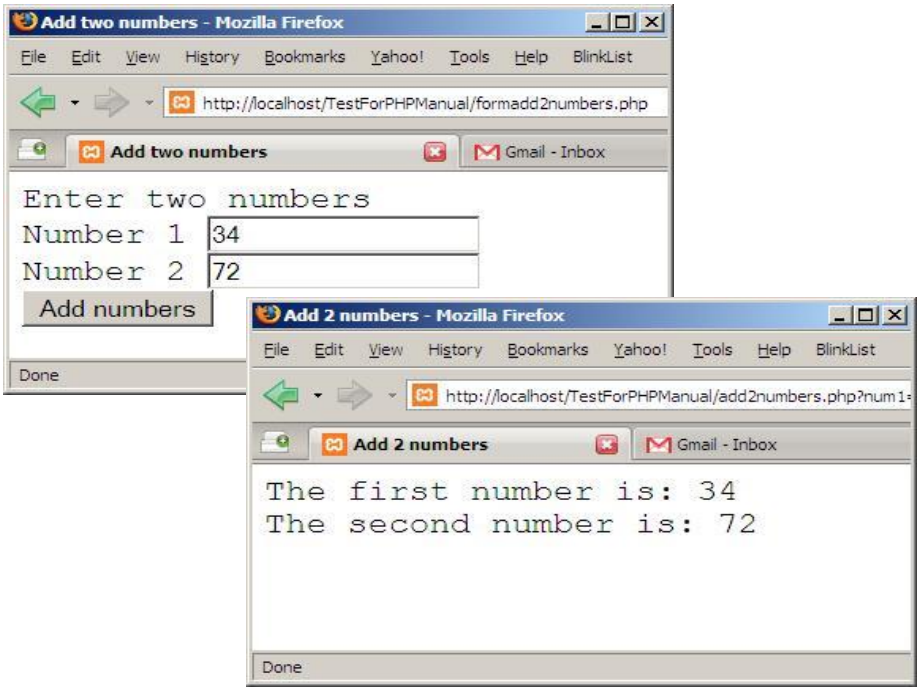
```
<html>
<head>
<title>Add New Book</title>
</head>
<body>
Add New Book<br>
<form method="GET" name="formNew" action=" displayBookInfo.php">
        ISBN        <input type="text" name="isbn"><br>
        Title       <input type="text" name="title"><br>
        Author      <input type="text" name="author"><br>
        Publisher   <input type="text" name="publisher"><br>
        Year        <input type="text" name="year"><br>
        Quantity    <input type="text" name="number"><br>
        <input type="submit" value="Save Book" name="submit">
        <input type="reset" value="Reset" name="reset">
</form>
</body>
</html>
```

Exercise 3.2

Create the form for the user to enter the information for his/her name, home address, e-mail, mobile phone number, and also a submit button to send the information to the target file. Create another page (the target file) to receive the user's input, and display all the values.

Exercise 3.3

Below are the pages for a simple web calculator. The form will receive two numbers, sent the numbers to the target page, and display the numbers on the target page. Write the HTML tags and the PHP scripts for both pages.



operators

Objectives:

1. To introduce and use the PHP operators.
2. The categories of PHP operators are assignment, arithmetic, logical and comparison operators.
3. Assigning variables.
4. Operations on numbers using arithmetic operators.
5. Comparing values.
6. Combining statements using Boolean logical operators.

Assignment Operators.

- Assign means to give value to a variable to be stored.
- The symbol is "**=**".
- Equal to "**= =**" is different from assign "**=**", so don't get confused!!

Assignment Operators	**Meaning**	**Example**
Simple assignments		
*Var***=***expr*	Assign the value of expression into the variable.	`$nom=3;` (value 3 is assign the variable *nom*) `$a=50;` `$b=$a;` ($b receives the value of $a, which is 30) `$hasil=($nom*3)+10;` (value produced on the right expression will be assigned to variable *hasil*)
*Var****=***expr*	Multiplication assignment.	`$nom*=3;` (similar as $nom=$nom*3;)
*Var***/=***expr*	Division assignment.	`$nom1/=$nom2;` (similar as $nom1=$nom1/$nom2;)
*Var***%=***expr*	Modulus assignment.	`$num%=2;` (similar as $num=$num%2;)
*Var***+=***expr*	Addition assignment.	`jum+=var1;` (similar as $jum=$jum+$var1;)
Var **-=***expr*	Subtraction assignment.	`$jum-=$var1;` (similar as $jum=$jum-$var1;)

Arithmetic Operators.

These operators are used in the calculation of numerical values.

Operator	Operation	Example	
		Mathematical expression	**PHP expression**
+	Addition	num1 + 3	`$num1 + 3`
-	Subtraction	pi - y	`$pi - $y`
*****	Multiplication	width x height	`$width * $height`
/	Division	a/b or a ÷ b	`$a/$b`
%	Modulus	nom modulo 3	`$nom%3`

Operators order of evaluation (precedence)

Operators	Order of evaluation
( )	First
*, /, %	Second
+, -	Last

Example of mathematical expressions transform into PHP expressions.

Mathematical expression	PHP expression
average = $\frac{a + b + c + d}{4}$	average = (a + b + c + d) / 4;
volumeSphere=4/3 x 3.142 x radius^3	volumeSphere=(4/3)*3.142 * radius * radius*radius;
Fahrenheit = (Celcius x 9/5) +32	Fahrenheit = (Celcius *(9/5)) +32;
volumeCylinder=3.142 x radius^2 x height	volumeCylinder=3.142 * radius * radius * height;
parameterCircle = 2 x 3.142 x radius	parameterCircle = 2.0 * 3.142 * radius;

EXAMPLES for Arithmetic Operations

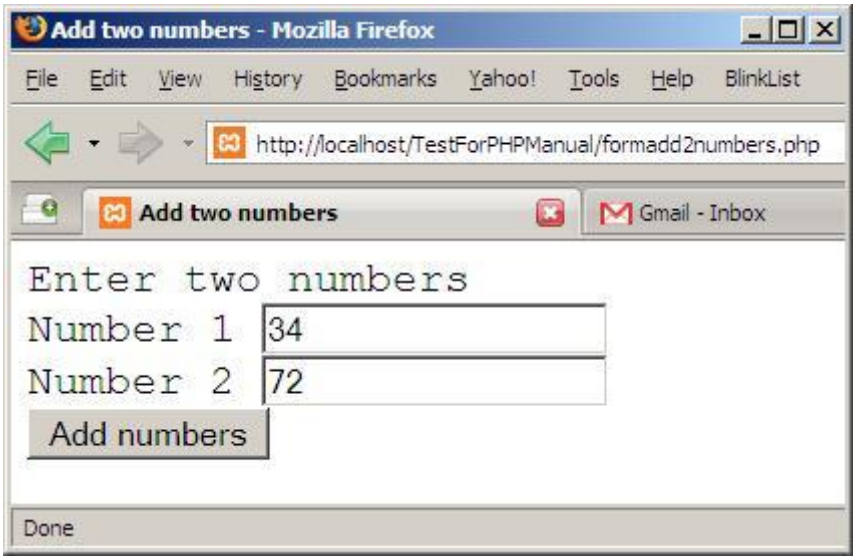
Example 1: Add 2 numbers

You're to develop a simple web application to receive two numbers from the user. The numbers is entered through a HTML form, sent to the server, and another file will receive the number, add both of them and display the result of the operation.

Step 1. Create a HTML page with a form to receive two numbers, and a submit button. Name it *formadd2numbers.php*.

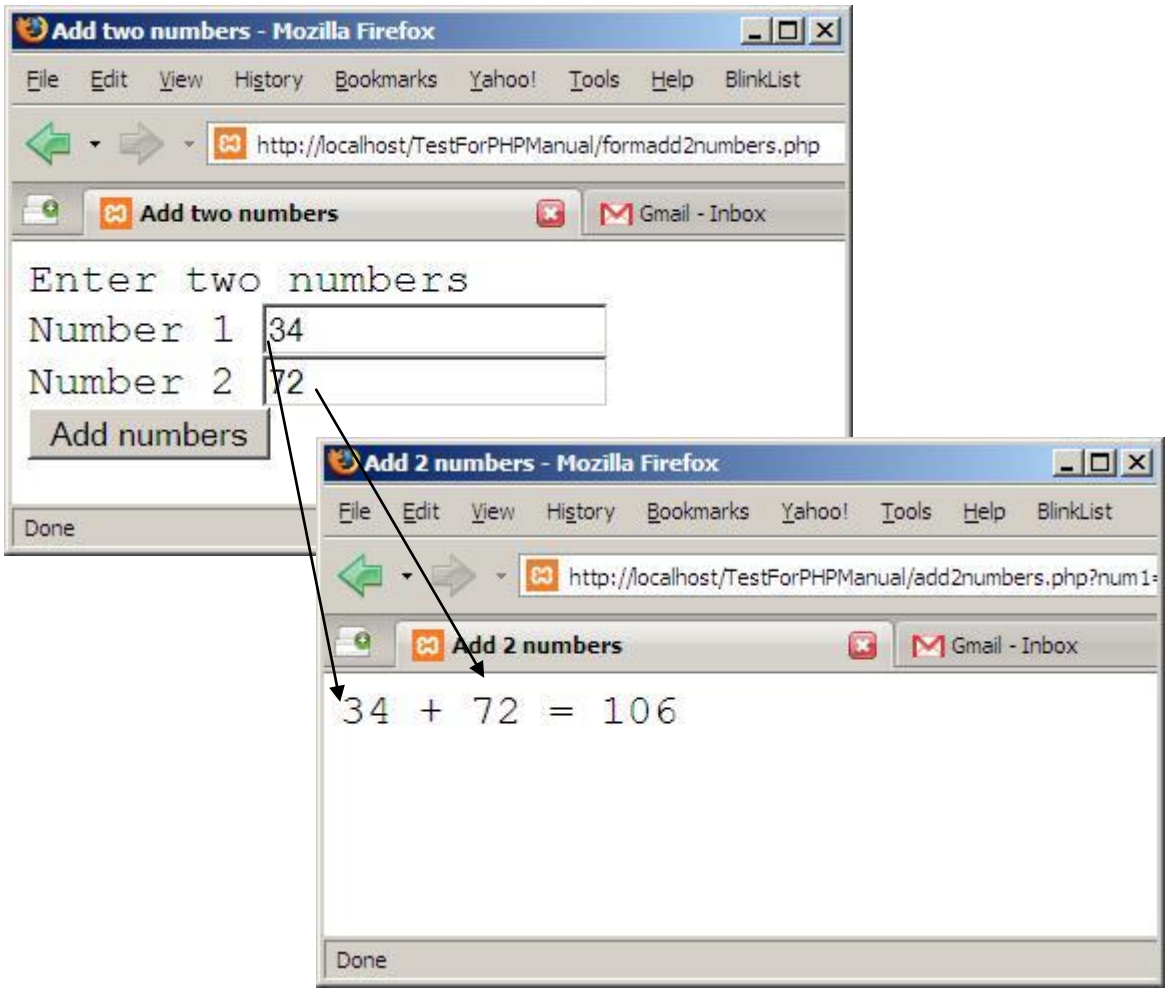
```
<html>
<head>
<title>Add two numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="formNew" action="add2numbers.php">
        Number 1      <input type="text" name="num1"><br>
        Number 2      <input type="text" name="num2"><br>
        <input type="submit" name="btnAdd" value="Add numbers">
</form>
</body>
</html>
```

Step 2. View the page in the browser. Key in any number. **Don't click the button yet.** Go to Step 3, finish the Step 3 and then you can hit the button.



Step 3. Create a new page, write the code below, and save as *add2numbers.php* .

```
<html>
<head>
<title>Add 2 numbers </title>
</head>
<body>
<?php
        $n1=$_GET["num1"]; //retrieve the first number
        $n2=$_GET["num2"];  //retrieve the second number
        $hasil=$n1+$n2; // here is the operation
        echo " $n1 + $n2 = $hasil"; //display all the values
?>
</body>
</html>
```



Example 2: Currency converter

Convert a currency value from RM into USD. Given RM1 is equivalent to USD3.40.

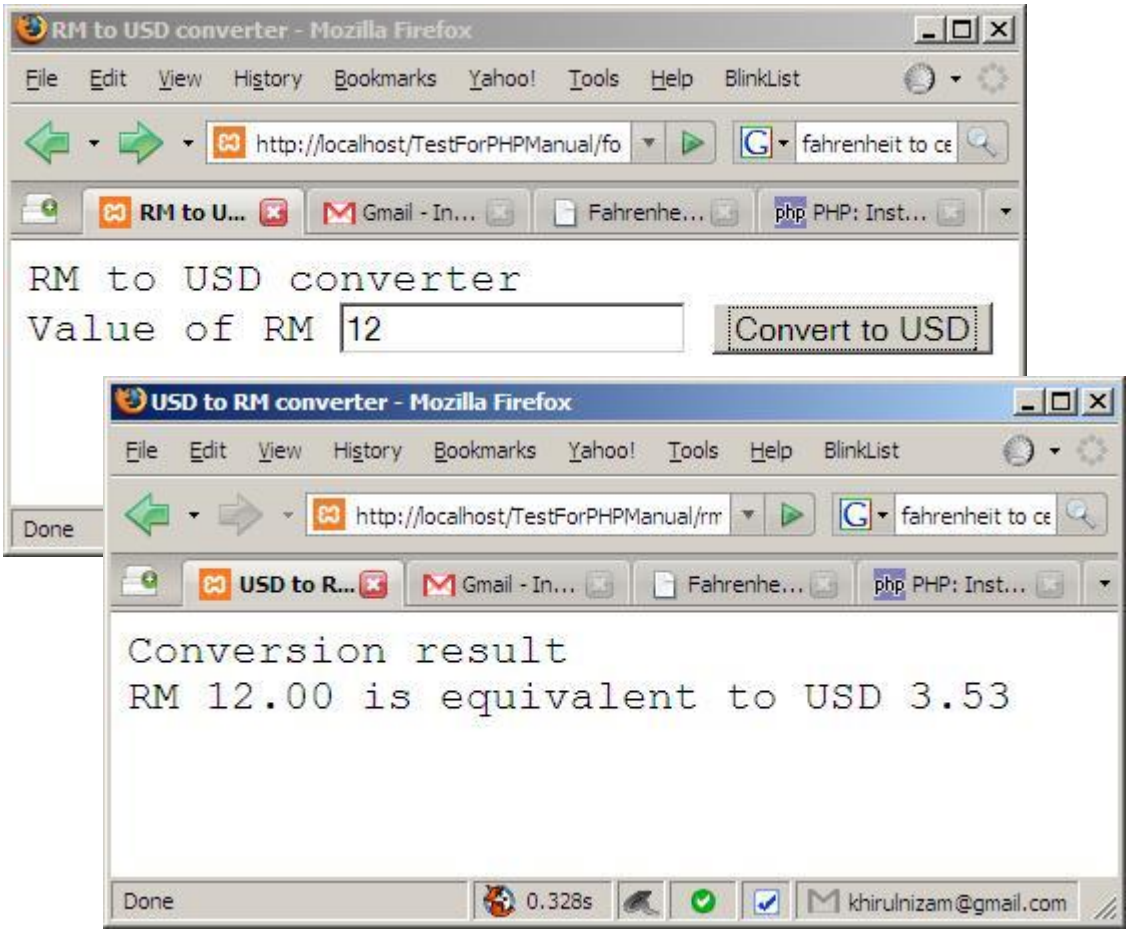
Step 1. Create a HTML page with a form to receive a value in Malaysian ringgit, and a submit button. Name it *formRMUSDConverter.php*. Use the code below.

```
<html>
<head>
<title>RM to USD converter</title>
</head>

<body>
RM to USD converter<br>
<form name="formconvert" action="rmusdconverter.php" method="get">
Value of RM <input  name="txtrm"  type="text">
        <input name="btnconvert" type="submit" value="Convert to USD">

</form>

</body>
</html>
```

Step 2. Next, create a HTML page to receive the value in Malaysian ringgit, and cenvert the value into USD. Save and name it as *rmusdconverter.php*. Use the code below.

```
<html>
<head>
<title>USD to RM converter</title>
</head>

<body>
Conversion result<br>
<?php
        $rmvalue=$_GET["txtrm"];

        $usdvalue=$rmvalue/3.4;
        printf("RM %.2f",$rmvalue);
        echo " is equivalent to ";
        printf("USD %.2f",$usdvalue);

?>
</body>
</html>
```

Step 3. Preview the page (in the browser) with the name *formRMUSDConverter.php,* key in a value in the text box and click the button “Convert to USD”. The browser should go to the next page named *rmusdconverter.php* (as defined in the form’s action) and display the value converted from RM into USD.


RM to USD converter - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/fo
fahrenheit to ce
RM to U...
Gmail - In...
Fahrenhe...
PHP: Inst...
RM to USD converter
Value of RM 12
Convert to USD
Done
USD to RM converter - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/rm
fahrenheit to ce
USD to R...
Gmail - In...
Fahrenhe...
PHP: Inst...
Conversion result
RM 12.00 is equivalent to USD 3.53
Done
0.328s
khirulnizam@gmail.com

Example 3: Simple statistical web application for calculating sum and average of five numbers.

Step 1. Create the form to receive five numbers from the user. Use the code below, and save as *form5numbers.php*.

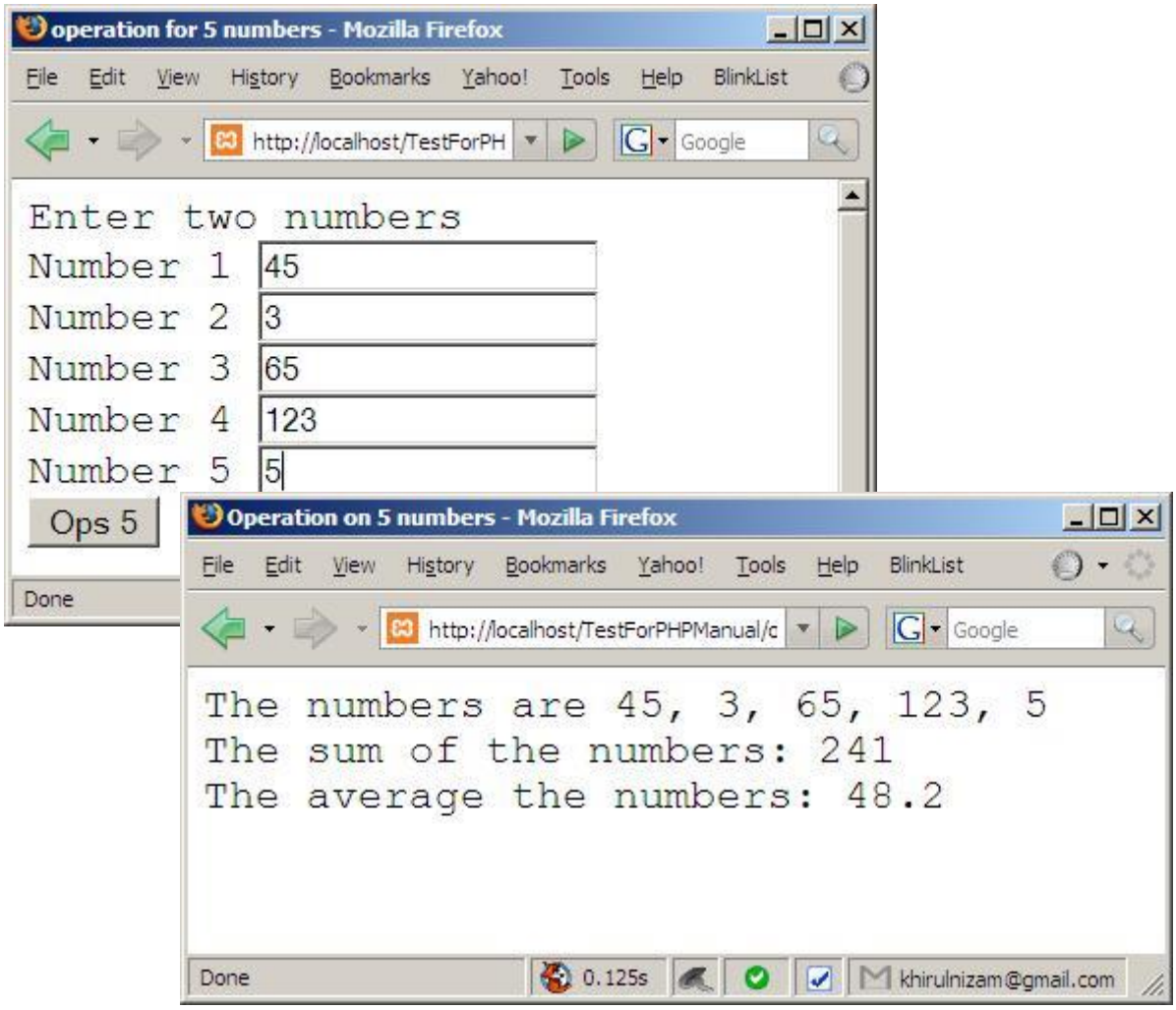
```
<html>
<head>
<title>operation for 5 numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="form5Numbers" action="ops5numbers.php">
        Number 1    <input type="text" name="num1"><br>
        Number 2    <input type="text" name="num2"><br>
        Number 3    <input type="text" name="num3"><br>
        Number 4    <input type="text" name="num4"><br>
        Number 5    <input type="text" name="num5"><br>
        <input type="submit" name="btnOps" value="Ops 5">
</form>
</body>
</html>
```

Step 2. Create another page to receive the numbers entered by the user. Save as *ops5numbers.php*.

```
<html>
<head>
<title>Operation on 5 numbers </title>
</head>
<body>
<?php
        //retrieve the numbers
        $n1=$_GET["num1"];
        $n2=$_GET["num2"];
        $n3=$_GET["num3"];
        $n4=$_GET["num4"];
        $n5=$_GET["num5"];
        $sum=$n1+$n2+$n3+$n4+$n5;
        $average=$sum/5;
        echo "The numbers are $n1, $n2, $n3, $n4, $n5<br>";
        echo "The sum of the numbers: $sum<br>";
        echo "The average the numbers: $average<br>";
?>
</body>
</html>
```

Step 3. View the form (*form5numbers.php*) in the browser and key-in a number in each of the boxes, and click the button.

Step 4. The sum and the average of the five numbers are displayed.


operation for 5 numbers - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPH
Google
Enter two numbers
Number 1 45
Number 2 3
Number 3 65
Number 4 123
Number 5 5
Ops 5
Done
Operation on 5 numbers - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/c
Google
The numbers are 45, 3, 65, 123, 5
The sum of the numbers: 241
The average the numbers: 48.2
Done
0.125s
khirulnizam@gmail.com

Exercise 4.1: Mathematical Expression

1. What is the numerical value of each of the following expressions as evaluated by the PHP scripting language?
   i. (20/2)%2
   ii. 3%2
   iii. 10%(10/5)
   iv. 8%5*3+1
   v. 2+4*5
   vi. 10/(4+1)%3
   vii. 1+5*3/3+5%4

2. Convert the following mathematical formula into PHP expression.
   ii. $y = b^2-4ac$
   iii. $b = 2bx^2 + 3(a+b)x + 3a$
   iv. $m = \frac{y_2 - y_1}{x_2 - x_1}$
   v. $n = a^3+b^2$
   vi. $d = (x+10)(x-3)$
   vii. $C = \frac{5}{9}(F-32)$

*(*For all the problems in question 3-10, you are required to create a HTML form for the input page, and another page to display the result of the calculation).*

3. Multiplication of two numbers entered by user.

4. Receive a length in kilometer, convert the length into meter.

5. Convert a currency value from US dollar (USD) into Malaysian ringgit (RM).

6. Convert a distance in mile into kilometer.
   Given: 1 mile is equivalent to 1.609344 kilometers.

7. Convert temperature value from Fahrenheit into Celsius.
   Given: Celsius=(5/9)*(Fahrenheit-32)

8. You are given the following page with a form to receive the price of item and the quantity for the user to buy the item.

9. You are to develop a loan payment calculator. The first page is the form for the user to key in the sum of loan, the interest rate, and the number of years to settle the payment. After the user entered all the information, the user will click a button and the information will be submitted to the server. In the server there is another file waiting to calculate and display the monthly installment to be paid, and the total sum of payment to be made.

Comparison Operators.

The comparison operators are used to compare values.

Operator	Meaning	Example
==	Equal to	($nombor == 100)
!=	Not equal to	($huruf != 'Z')
<	Less than	($nom1 < $nom2)
>	Greater than	($markah > 80)
<=	Less than or equal to	($bil <= 10)
>=	Greater than or equal to	($bil >= $input)

Logical Operators.

The logical operators are used for Boolean expression or logical comparison. The Boolean expression produces **`true`** if the statement is true and **`false`** if the statement is false.

Operator	Meaning	Example
&&	Logical AND	($x==1) && ($y==2)
||	Logical OR	($huruf=='A') \|\| ($nom>10)
!	Logical NOT	(!EOF) !($nombor > 100)

Examples of expression:

Statement	convert to PHP expression
x is between 10 to 50. (*Pembolehubah x bernilai di antara 10 hingga 50*)	`($x>=10)&&($x<=50)`
y is not equal to f or q. (y tak sama dengan f atau q)	`($y!=$f)||( $y!=$q)`

z is multiplication of 5.
(*z adalah gandaan 5*)
 	
z is not multiplication of 5.
(*z bukan gandaan 5*)
	`($z%5 == 0)`
	
	`($z%5 != 0)`

Var1 is negative number. (*Var1 ialah nombor negatif*)	`($Var1 < 0)`
Var2 is between 0 to 100, but 20 to 50 are excluded. (*Var2 adalah antara 0 hingga 100, tapi antara 20 hingga 50 tak termasuk*)	`(($Var2>=0)&&($Var2<=100)&&` `(!(($Var2>=20)&&($Var2<=50)))`

Other Operators

Operator	Meaning	Example

++	Unary pre-increment (add 1 to the value in a variable)
	Unary post-increment (add 1 to the value in a variable *later*)
		`$nom++;`
		
		`++$nom;`

--	Unary pre-decrement (subtract 1 from the value in a variable)
	Unary post-decrement (subtract 1 from the value in a variable *later*)
		`$jum--;`
		
		`--$jum;`

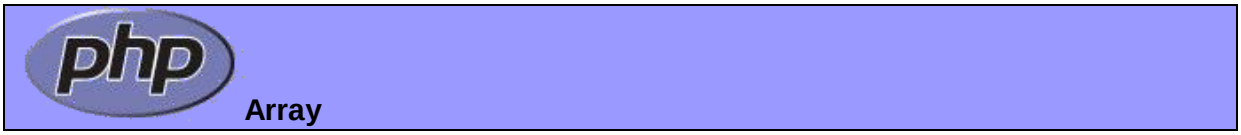
{ }	Parenthesis (block of statements)	`if($val==0){`
		`.....`
		}

[ ]	Array subscript (index number for array variable)	`$j[1]=100;`
.	Concatenate (merge several separated strings)	`$name="Kerul"." Rahman";` `$name+=" Abd Rahman";`
;	Semicolon (to end a statement)	`$a=5;` `$a+=10;`
( )	Bracket As the arithmetic expression separator.	 `$average=($a+$b+$c)/3;`
//	Line comment	
/* …*/	Block comment	

Exercise 4.2: Boolean Expressions

1. What is the value of the following Boolean expressions (true/false)?
   a. `4 > 1`
   b. `!((4+1)>5)`
   c. `'c'=='C'`
   d. `!(false)&&(true)`
   e. `(4>4)||(100%10==5)`

2. Create the Boolean expression (in PHP format) for each of the following statements.
   a. $X is positive integer.
   b. $Z is negative integer.
   c. $W is between –100 to 100.
   d. $Y is multiplication of 10.
   e. $A is multiplication of 3.
   f. $num1 is not equal to 10.
   g. $num2 is even number.
   h. $num3 is odd number.
   i. $num4 is a prime number.
   j. $var2 is devisable by 5.
   k. $gaji is greater than 5000.
   l. $huruf1 is vocal.
   m. $huruf2 is consonant.
   n. $huruf3 is upper case.
   o. $huruf4 is lower case.
   p. $huruf5 is equal to 'm'.
   q. $huruf6 is not equal to 'Z'.



Array

In computer science an **array** is a data structure consisting of a group of elements that are accessed by indexing. In most programming languages each element has the same data type and the array occupies a contiguous area of storage. Most programming languages have a built-in array data type.

Some programming languages support array programming which generalizes operations and functions to work transparently over arrays as they do with scalars, instead of requiring looping over array members.

Well as we can see here guys, array can be divided into 3 categories:

- **Numeric array**
- **Associative array**
- **Multidimensional array** (sorry, will not be discussed here)

We're going to define one by one of the categories that provided above with the example for each of it.

Numeric array

When we discuss about numeric ID it can be done using 2 ways. It is :

- Assign using Automatic method
- Assign using Manual method

A numeric array stores each element with a numeric ID key. Using the numeric ID, the index of the array can be stored using 2 ways. The below example is automatically assigned into the array.

Example 1:

```
 7
 8  <body>
 9  <?
10      $names = array("ali","abu","bakar");
11
12  ?>
13  </body>
14  </html>
15
```

You can see the variable $names that used as array that access by its element. This is how when a string has been initialized into the array. Because array by default is empty and to be cleared every array is start from 0. If we declared the array with the size of 9 actually the max for it is only 8, because all array is start from 0. Next using the same technique, we see in the second example.

Example 2:

```
<body>
<?php

    $number = 10;
    $sekret = array("Example",7,$number);
    echo $sekret[0]; //prints: Example
    echo $sekret[1]; //prints: 7
    echo $sekret[2]; //prints: 10

?>
</body>
</html>
```

As you can see, elements in array can be any type scalar of data (string, number, variable) and so on. So, here the advantage when using array. You can initialize any type data into it as long it follows the rule using the array.

The second way stored the data in numeric ID can be done manually. As shown in the example below:

Example 1:

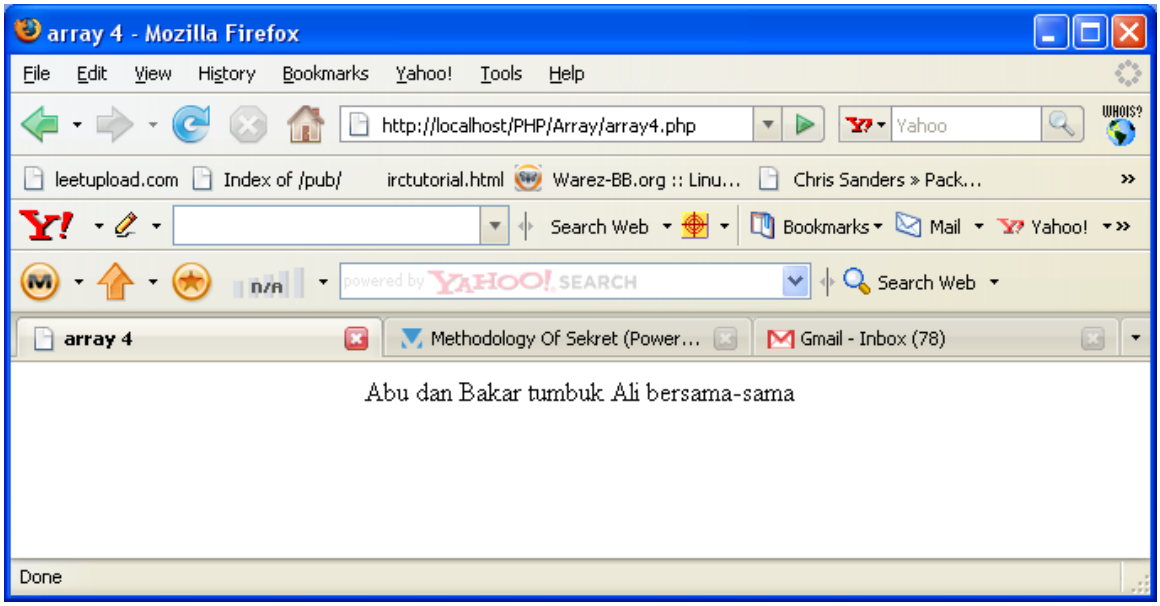
```

<body>
<?
    $names[0] = "Ali";
    $names[1] = "Abu";
    $names[2] = "Bakar";
?>
</body>
</html>

```

As we look the example above, every element is initialized using the manual method. “peter” was initialize to Array number 1 a.k.a names[0] and so on . So that’s the example how it done using manual method.

The ID key can also be used in script. Refer to the example below.

```
<body>
<?php

    $names[0] = "Ali";
    $names[1] = "Abu";
    $names[2] = "Bakar";

    echo $names[1] . " dan " . $names[2] .
    " bakar ". $names[0] . "bersama-sama";
?>
</body>
</html>
```

Output :



Associative array

Basic concept for associative array is quite similar with numeric array. The difference between those arrays is the way how to initialize ID key. Each ID key is associated with value. When storing data about specific named values, a numerical array is not a best way to do it.

By using associative array, value can be assign as keys, and assign value to them. Refer to example 1 and 2; there are 2 methods to initiate this array.

```
<?php

        $ages = array("Ali"=>32 , "Ahmad"=>25, "Abu"=> 28);

?>
```

Example 1: First method to initiate associative array

```
<?php

        $ages["Ali"] = 32;
        $ages["Ahmad"] = 25;
        $ages["Abu"] = 28;

?>
```

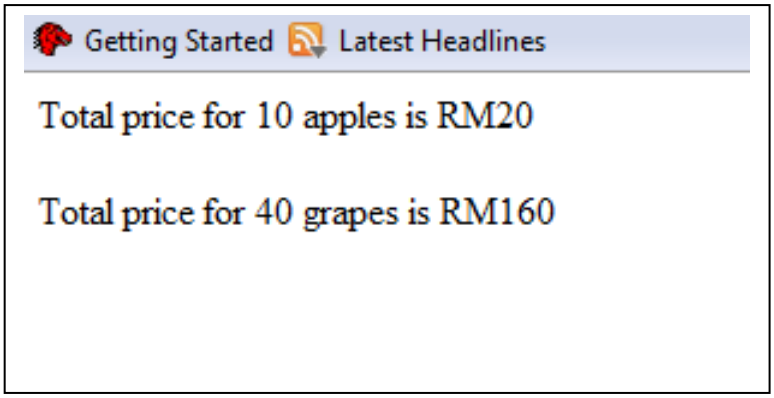
Example 2: Second method to initiate associative array

Refer to example 3, the application sample code to assign array using first and second methods, and output for this type of array.

Sample code:

```
<?php

  //first method
  $fruit = array("Apple" => 2, "Pineapple" => 3, "Orange" => 1, "Grape"=>4);

  //second method
  $quantity["Apple"] = 10;
  $quantity["Pineapple"] = 20;
  $quantity["Orange"] = 30;
  $quantity["Grape"] = 40;

  //application or operation
  $price["Apple"] = $fruit["Apple"] * $quantity["Apple"];

  $price["Grape"] = $fruit["Grape"] * $quantity["Grape"];

  //output to screen
  echo "Total price for $quantity['Apple'] apples is RM$price['Apple']<br><br>";

  echo "Total price for $quantity['Grape'] grapes is RM$price['Grape']<br><br>";

?>
```

Output :



selection structures

SELECTION IN PHP

if construct.

- *if* is a PHP reserved words to perform selection.
- There is another selection, that is *switch…case*. We will only discuss *if* for this course.
- There are 4 ways of using *if*.
  - *i)* Simple *if*
  - *ii)* *If … else*
  - *iii)* *if … else if … else if … else*
  - iv) Nested *if*

A. simple *if*

(one choice selection)

The format:

```
if (condition) {
        statements;
}
```

- Condition is a Boolean expression that will produce ***true*** or ***false***.
- All the statements between the if *block* will only be executed if the value produced by the condition is true.

Example: *makan* and *kenyang*.
Study the sentence below;

If I eat, then full.

If there is a programming language in plain English, the sentence above will become like this if it is translated into program.

```
If I eat, then
        I'm full.
```

I'm full will happen if only *I eat* is true.

Another example :
Below is a simple algorithm,

If *gred* is 'A', then display "You are great!".

gred is a character variable to hold a character for a student's grade. If the character inside *gred* is 'A', then the program will display message "You are great".

In PHP, the program will be like this;

```
if  ($gred = = 'A')
        echo ("You are great!");
```

B. *if …. else*

(Used when the selection have only two choices)

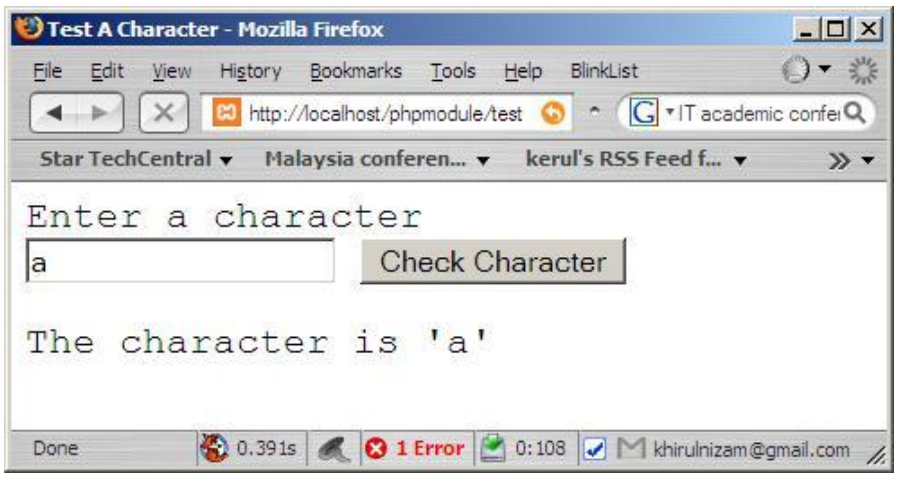
Example :
The example below shows the usage of simple *if...else* .
Algorithm :

1. receive a character from a user
2. if the character is 'a'
   then display "Huruf yang anda masukkan ialah a"
3. if the character is other than 'a'
   then display "Huruf yang anda masukkan bukan a"

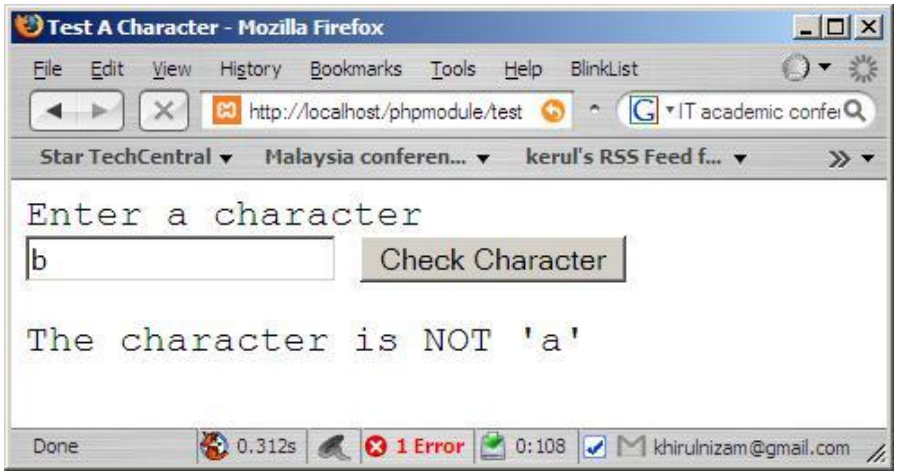
Below is the complete program

```
<html>
<head><title>Test An Alphabet</title></head>
<body>
Enter a small alphabet<br>
<form name="frmChar" method="get" action="testCharacter.php">
  <input name="txtchar" type="text">
  <input name="btnSubmit" type="submit"  value="Check Char">
</form>

<?php
if ($_GET['txtchar'] !=NULL){
    if ($_GET['txtchar']=='a'){
        echo "The alphabet is 'a'<br>";
    }
    else{
        echo "The alphabet is NOT 'a'<br>";
    }
}
?>

</body>
</html>
```

If user enter an ‘a’, then the message “The character is a ” will appear. However if the input is other than ‘a’, the message is “The character is NOT a ”.



This is the output if user enters ‘a’ for the input.



This is the output if user enters other than ‘a’ for the input.

C. *if …. else if …..else if …..else if*

(Used to handle multiple-choice selection. No limitation on the number of the selection)

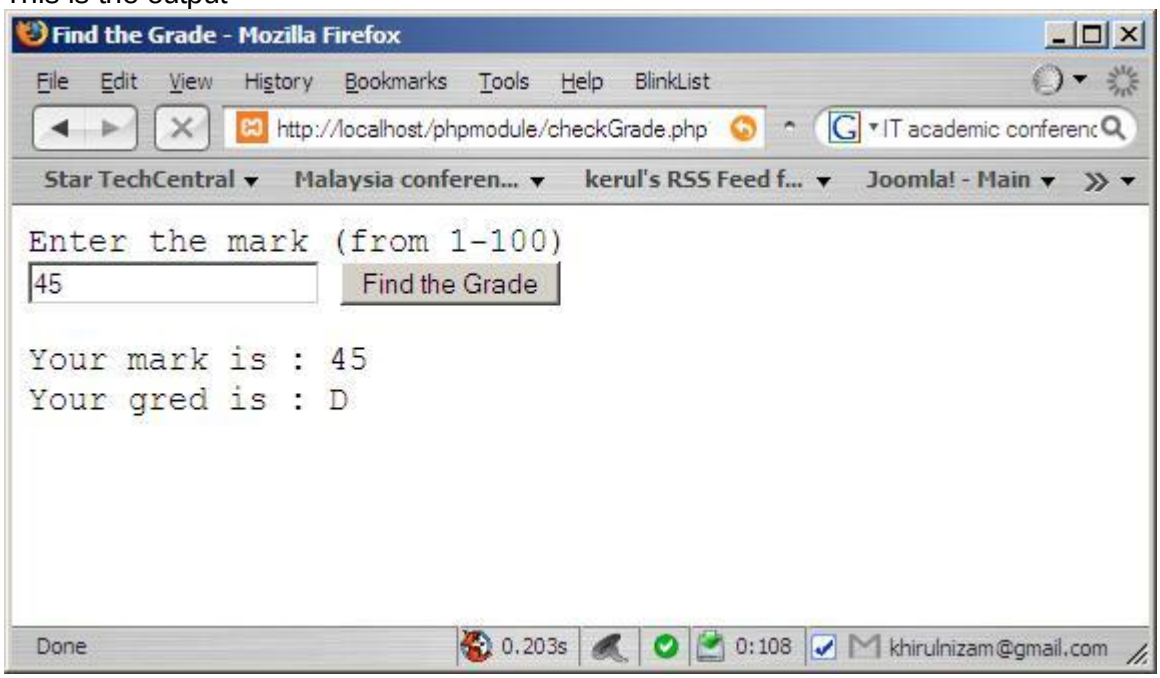
Example:
The program will display the grade for the entered mark.
Algorithm:

1. Get the mark entered by user.
2. if mark is between 100 to 80,
   then grade is A
   else, if mark is between 76 to 60,
   then grade is B
   else, if mark is between 56 to 50,
   then grade is C
   else, if mark is between 46 to 40,
   then grade is D
   else, if mark is between 36 to 0,
   then grade is F
   else,
   then "INPUT NOT VALID!!!!"
3. Display mark and grade.

```
<html>
<head><title>Find the Grade</title></head>
<body>
Enter the mark (from 1-100)<br>
<form name="frmMark" method="get" action="checkGrade.php">
    <input name="txtmark" type="text">
     <input name="btnSubmit" type="submit"  value="Find the Grade">
</form>
```

```
<?php
$mark=$_GET['txtmark'];
if ($mark != NULL){
    //mark is between 80-100
    if (($mark >=80)&&($mark <=100))
        $gred='A';
    //mark is between 60-80
    else if (($mark >=60)&&($mark <80))
        $gred='B';
    //mark is between 50-60
    else if (($mark >=50)&&($mark <60))
        $gred='C';
    //mark is between 40-50
    else if (($mark >=40)&&($mark <50))
        $gred='D';
    //mark is between 0-40
    else if (($mark >=0)&&($mark <40))
        $gred='F';
    //mark is out of range
    else
        $gred='input is not valid';
  //display result
  echo "Your mark is : $mark <br>";
  echo "Your gred is : $gred <br>";
}?> </body></html>
```

This is the output



D. *if….else* nested.

There are another *if* statements inside an *if* statement.

```
if (…..){
      ……..
      if (……..){
            ……….
            ……….
       }
      else if (…….){
            ……….
            ……….
       }
      else {
           ……….
       }
}
else if (…..){
       ……….
       ……….
}
else {
       ……….
       ……….
}
```

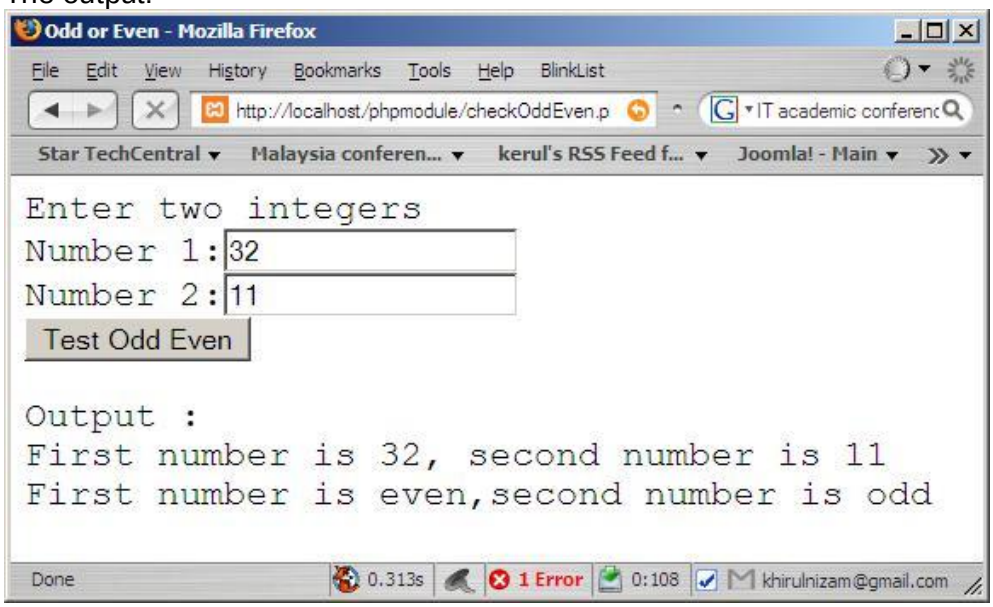
Example:

The program in page 6 will ask the user to enter two integers. Then it will decide whether both numbers are odd or even or the numbers either one is odd or even.

Algorithm:
1. receive 2 integers.
2. if the first number is even
   and the second number is even, then
   display “Kedua-duanya genap”,
   but if the second number is odd, then
   display “Nombor pertama genap, kedua ganjil”,
3. else if the first number is odd
   and the second number is odd, then
   display “Kedua-duanya ganjil”,
   but if the second number is even, then
   display “Nombor pertama ganjil, kedua genap”,

```
<html>
<head>
<title>Odd or Even</title>
</head>
<body>
Enter two integers<br>
<form name="frmMark" method="get" action="checkOddEven.php">
  Number 1:<input name="txtnum1" type="text">
  <br>
  Number 2:<input name="txtnum2" type="text">
  <br>
  <input name="btnSubmit" type="submit"  value="Test Odd Even">
</form>
<?php
$num1=$_GET['txtnum1'];
$num2=$_GET['txtnum2'];

if ($num1 != NULL || $num2 != NULL){
      echo "Output :<br>";
      echo "First number is $num1, second number is $num2<br>";
      if ($num1%2 == 0){//first number is even
            if ($num2%2 == 0)//and second number also even
                  echo("Both numbers are even.");
            else //second number is odd
                  echo("First number is even, second number is odd");
      }
      else {//first number is odd
            if ($num2%2 != 0)//and second also odd
                  echo("Both numbers are odd.");
            else //second number even
                  echo("First number is odd, second number is even");
      }
}
?>
</body>
</html>
```

The output:


Odd or Even - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmodule/checkOddEven.p
IT academic conferenc
Star TechCentral
Malaysia conferen...
kerul's RSS Feed f...
Joomla! - Main
Enter two integers
Number 1: 32
Number 2: 11
Test Odd Even
Output :
First number is 32, second number is 11
First number is even,second number is odd
Done
0.313s
1 Error
0:108
khirulnizam@gmail.com

FORM VALIDATION

Validating input/data from a form is a very important step in order to make sure the input given by the user is not garbage / junk. The is no point saving the input from a form to the database if the is no data inside the input element. Or trying to verify user's username and password if there are no username or password entered by the user.

Form Validating Example 1 : Validating Username and Password
The following example is to make sure the user key-in the username and password, not sending a blank form.

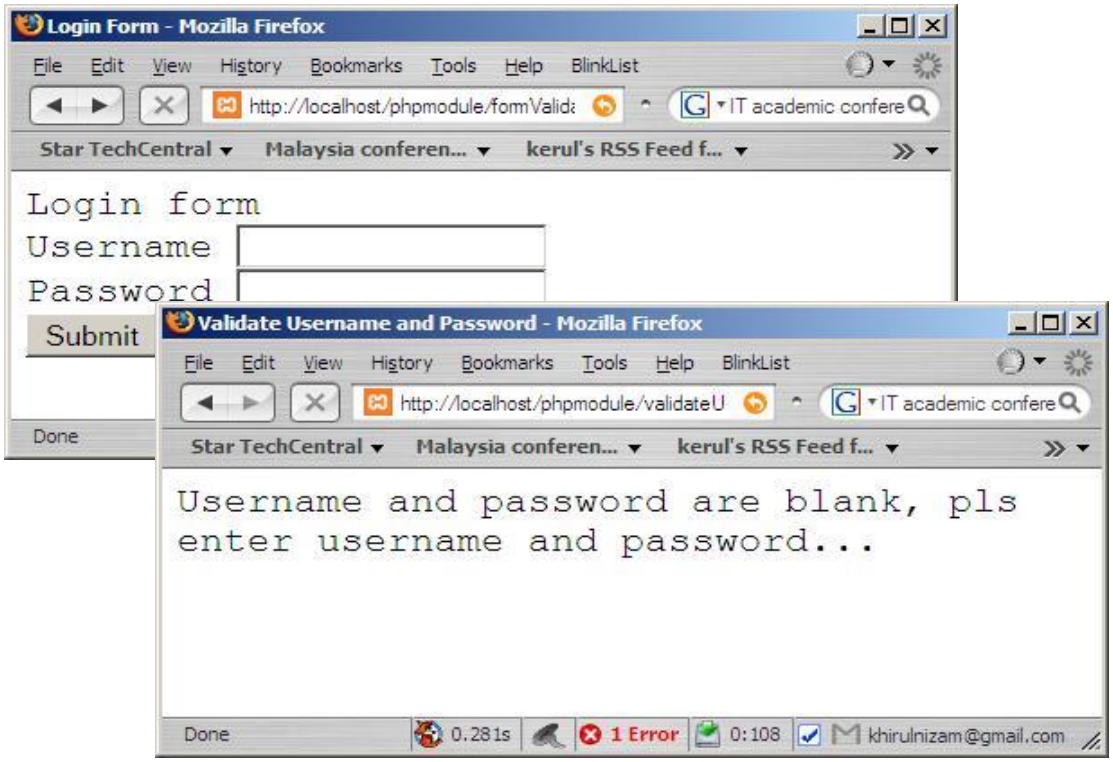
```
<html>
<head><title>Login Form</title></head>
<body>
Login form <br>
<form name="frmLogin"  method="get" action="validateUP.php">
  Username
     <input name="txtUsername"  type="text"><br>
  Password
     <input name="txtPassword"  type="password"><br>
     <input name="btnSubmit"  type="submit"  value="Submit">
  <input name="btnReset"  type="reset"  value="Reset">
</form>
</body>
</html>
```

Target file : validateUP.php

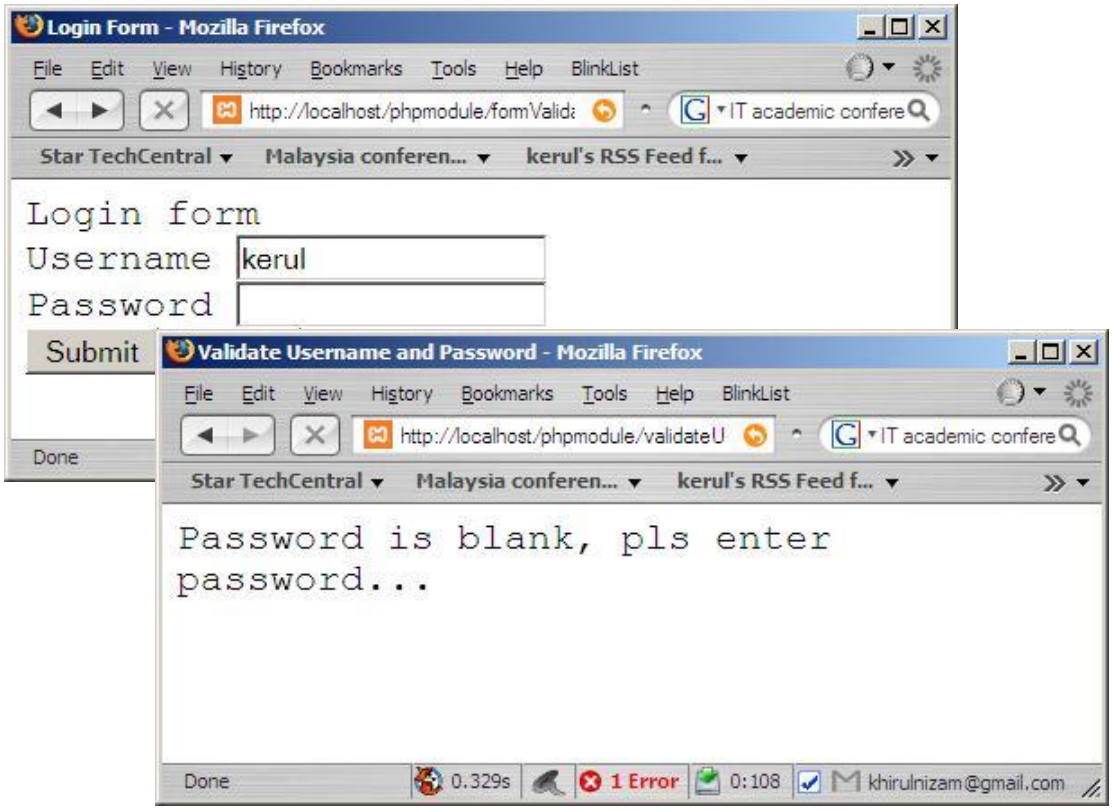
```
<html>
<head>
      <title>Validate Username and Password</title>
</head>
<body>
<?php
  $username=$_GET["txtUsername"];
  $pword=$_GET["txtPassword"];
  if($username==NULL && $password==NULL){
     echo "Username and password blank,
           pls enter username and password...";
  }
  else if($username==NULL){
     echo "Username is blank, pls enter username...";
  }
  else if($password==NULL){
     echo "Password is blank, pls enter password...";
  }
  else{
     echo "Your username is : $username <br>";
     echo "Your password is : $pword <br>";
  }
?>

</body>
</html>
```

The outputs:

Form both with username and password are blank.



Form both with password blank.



Form Validating Example 2 : Validating Numbers

This is how we make sure the user enter only number for the input.

```
<html>
<head>
<title>Add two numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="form2numbers"     action="add2numbers-
validate.php">
      Number 1   <input type="text" name="num1"><br>
      Number 2   <input type="text" name="num2"><br>
      <input type="submit" name="btnAdd" value="Add numbers">
</form>
</body>
</html>
```

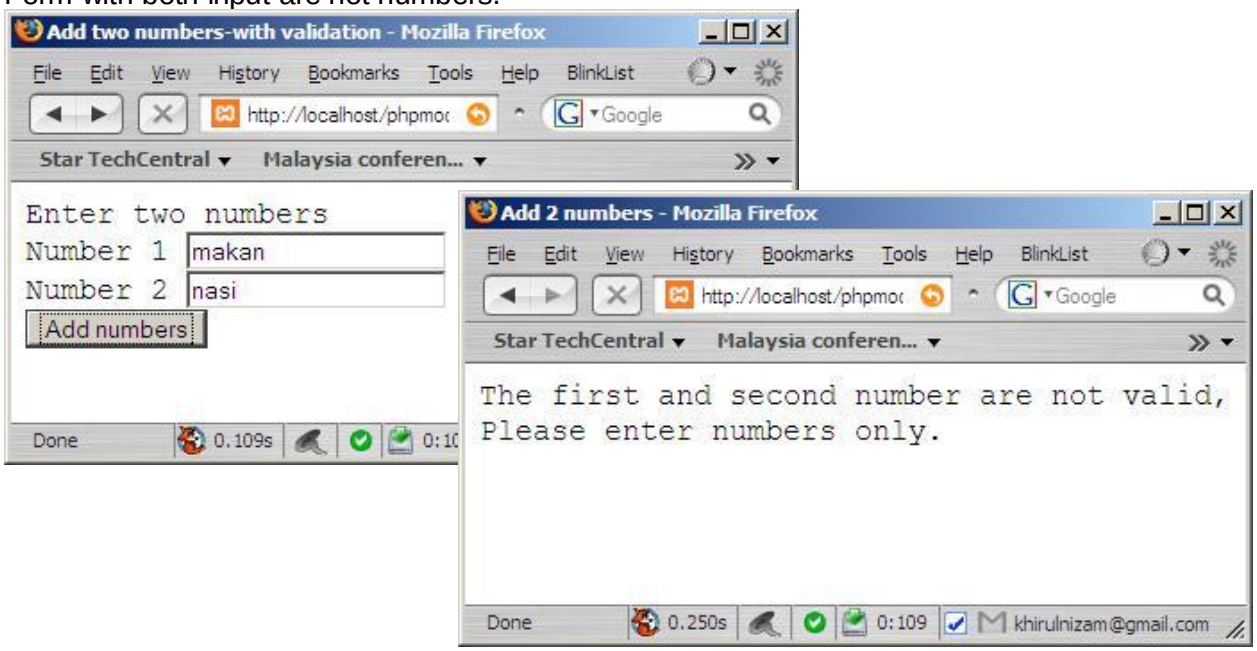
The target file : add2numbers-validate.php

```
<html>
<head>
<title>Add 2 numbers </title>
</head>
<body>
<?php
  $n1=$_GET["num1"]; //retrieve the first number
  $n2=$_GET["num2"];  //retrieve the second number

  //check so that both input are not blank
  if ($n1!=NULL || $n2!=NULL){
     //if both are numbers
     if(ctype_digit($n1) && ctype_digit($n2)){
          echo "The first number is: $n1 <br>";
          echo "The second number is: $n2";
          $hasil=$n1+$n2;
          echo " $n1 + $n2 = $hasil";
     }
     //if both are notnumbers
     else if(!ctype_digit($n1) && !ctype_digit($n2)){
          echo "The first and second number are not valid,<br>
                 Please enter a number only";
     }
     //if the first input is not number
     else if(!ctype_digit($n1)){
          echo "The first number is not a digit <br>";
          echo "Please enter a number only";
     }
     //if the second input is not number
     else if(!ctype_digit($n2)){
          echo "The first second is not a digit <br>";
          echo "Please enter a number only";
```

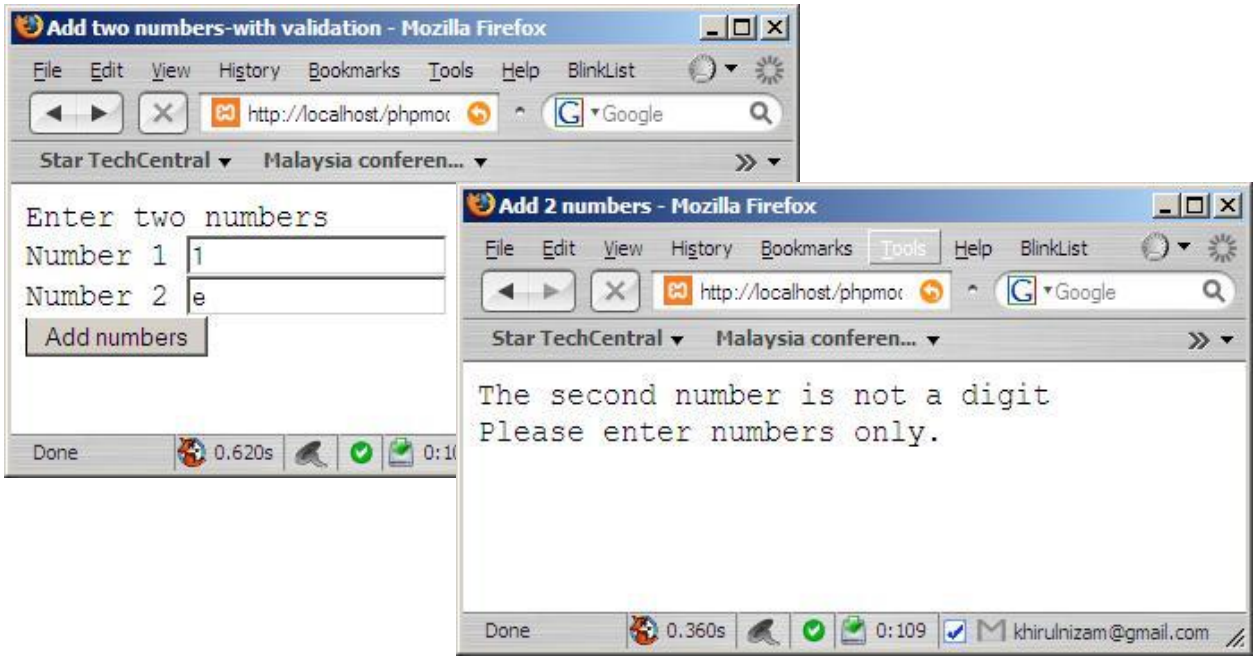
```
        }
    }
    else{//if both inputs are blanks
        echo "Please make sure to enter the both numbers<br>";
    }
?>
</body>
</html>
```

The outputs:

Form with both input are not numbers.

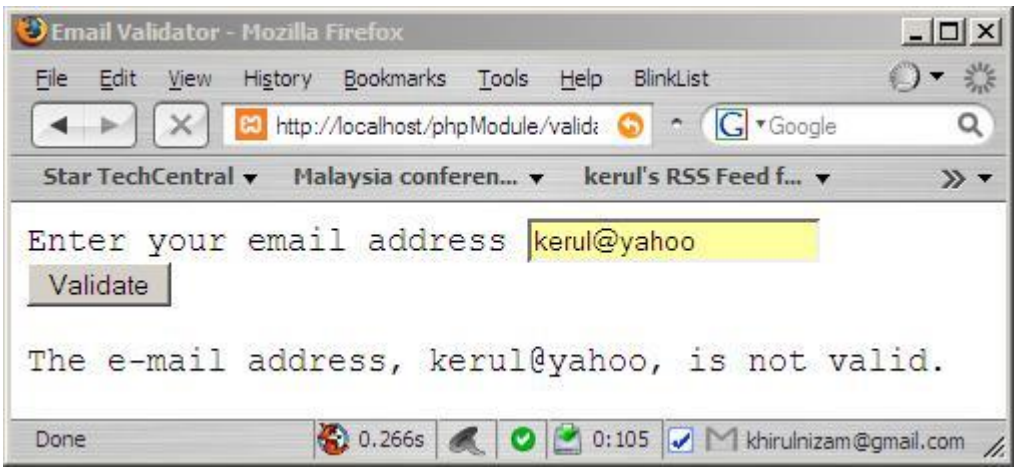


Form with both input are not numbers.



Form Validating Example 3 : Validating Email Address using Regular Expression

```
<html>
<head>
<title>Email Validator</title>
</head>
<body>
<form action="validateEmail.php" method="get">
Enter your email address
      <input name="txtemail" type="text"><br>
      <input name="btnsubmit" type="submit" value="Validate">

</form>
<?php
$emailadd=$_GET['txtemail'];
if ($emailadd != NULL){
   $email=$_GET["txtemail"];
   $emelRegexp ="^[a-z0-9,!#\$%&'\*\+/=\?\^_`\{\|}~-]+(\.[a-z0-
9,!#\$%&'\*\+/=\?\^_`\{\|}~-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*\.([a-z]{2,})$
";//in one line
      if(!eregi($emelRegexp, $emailadd)){
         echo "The e-mail address, $emailadd,  is not valid.";
      }
      else {
         echo "The e-mail address, $emailadd,  is valid.";
      }
}
else{
      echo "No e-mail address yet to validate...";
}
?>
</body>
</html>
```



*For further information on *eregi*, please refer to php.net/eregi.
**The example on regular expression to validate e-mail address is provide by Markus Sipilä from user contributed note in http://www.markussipila.info/pub/emailvalidator.php.

Exercises for Selection in PHP

Question 1
Write a program in PHP to check an integer whether it is odd or even.

Question 2
Write a program in PHP to check whether a number is positive or negative.

Question 3

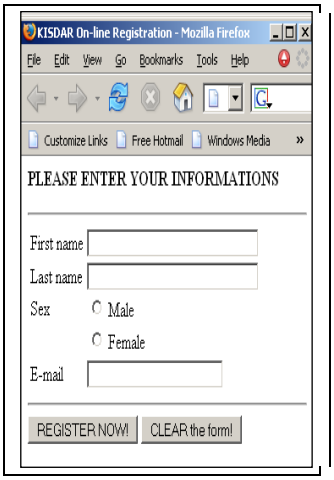


Figure 2

a. You are required to write the HTML code for the HTML document shown in Figure 2 above. Create a form named *formInfo* and send the data to a file named *processInfo.php* and the method is GET. Place the input components of the form inside a table and arrange them neatly. Give a name to each of the input component. "REGISTER NOW!" is the submit button and "CLEAR the form!" is the reset button.

b. Write a validation script in the *processInfo.php* file to make sure the user does not leave any input blank (user must key in all the information). If there is any invalid input or blank, inform the user by giving him/her messages.

repetition structures

Repetition control structures are used to execute a code block over and over again until the stopping condition is fulfill. It means you don't have to copy and paste your code many times in the file just use a right loop statement.

```
<?php
      echo "1";
      echo "2";
      echo "3";
      echo "4";
      echo "5";
      echo "6";
      echo "7";
      echo "8";
      echo "9";
      echo "10";
?>
```

As you can see the above statements are written without repetition control structure. The program tries to display a set of numbers from 1 to 10. The programmer needs to copy, paste and modify the echo command in order to print the ten numbers. This program will be a whole lot simpler if the programmer choose to use repetition structure, as below.

```
<?php
      $i=1;
      do {
            echo $i;
            $i++;
      }while ($i<=10);
?>
```

There are many ways of doing repetition. We will discuss three of the most popular repetition structure in PHP.

1. for
2. while and do … while
3. foreach

for

General format of *for* structure.

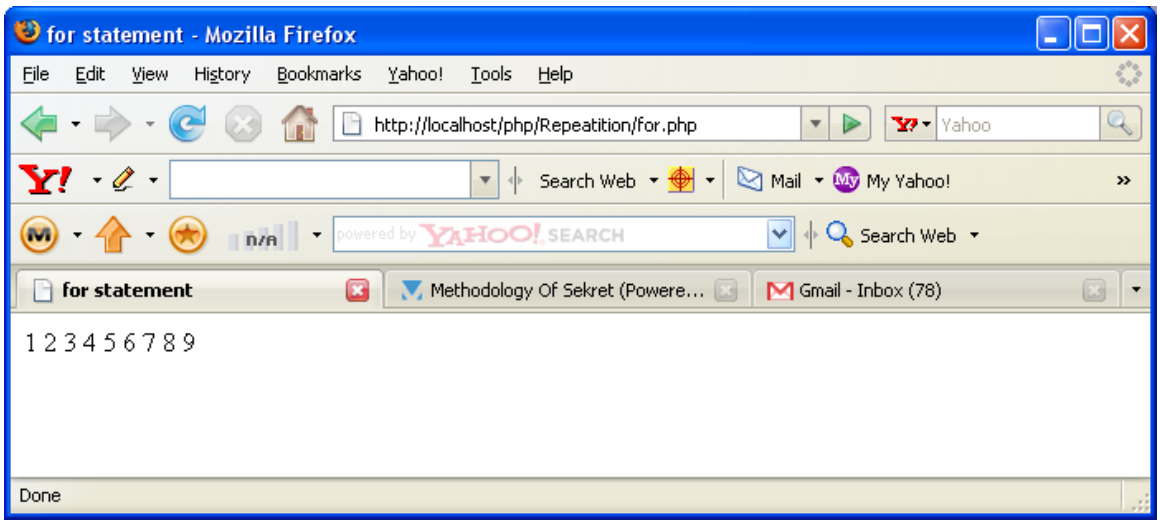
```
for (initialize a counter; conditional statement; increment the
counter){
      codes;
}
```

Example using the for loop:

```
7
8   <body>
9
10  <?php
11  for ($i=1;$i<=9;$i++){
12  echo " $i ";
13  }
14  ?>
15
16  </body>
17  </html>
18
```

Well as you can see both example above will given the same output:

Output:



*As you can see the solution with loop is much better. It is shorter, easier to understand. Besides this in most cases you don't know during the coding how many times the code block needs to be executed.
While Loop

while

General format of *while* structure.

```
while (condition)
code to be executed;
```

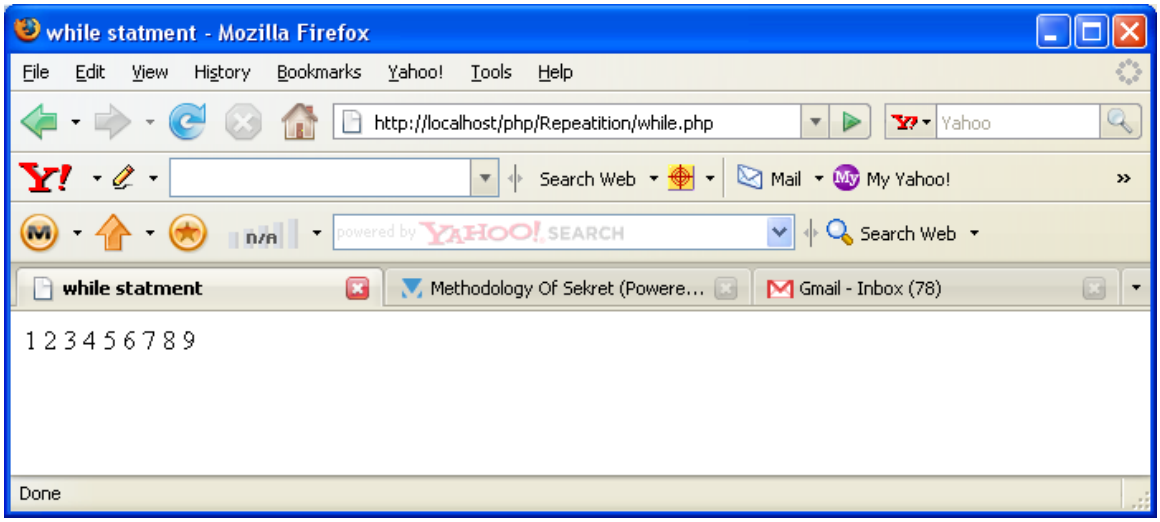
Example using the while loop

```
7
8
9   <body>
10  <?php
11          $i=1;
12          while ($i<=9){
13          echo " $i ";
14          $i++;
15          }
16  ?>
17  </body>
18  </html>
19
```

The code block will be executed until the condition is true. It means that it can happen that it will never execute. To implement our first example with while loop looks like this:

The output will be exactly the same as before. However if you initialize $i variable with 10 ($i=10) then nothing will be displayed. If you forgot to increment the variable it will result an endless loop as the condition will be never changed and it is always true.

Output:



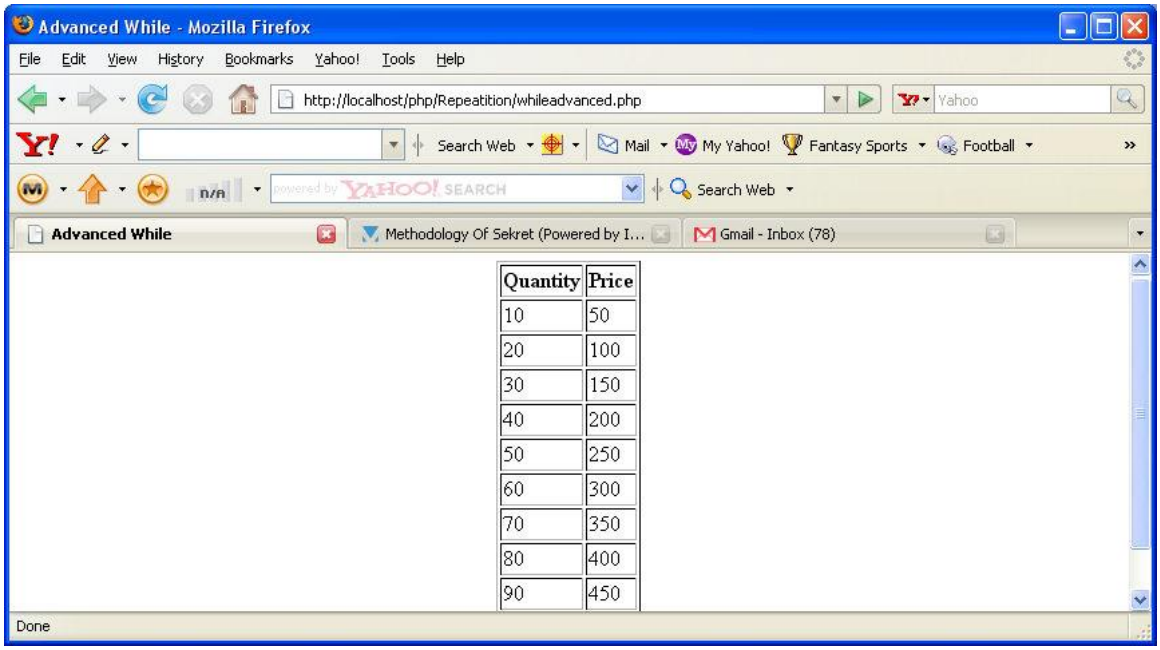
Another while example:

Imagine that you are running an art supply store. You would like to print out the price chart for number of brushes and total cost. You sell brushes at a flat rate, but would like to display how much different quantities would cost. This will save your customers from having to do the mental math themselves.

You know that a while loop would be perfect for this repetitive and boring task. Here is how to go about doing it.

```
8   <body>
9   <?php
10      $brush_price = 5;
11      $counter = 10;
12
13      echo "<table border=\"1\" align=\"center\">";
14      echo "<tr><th>Quantity</th>";
15      echo "<th>Price</th></tr>";
16      while ( $counter <= 100 ) {
17          echo "<tr><td>";
18          echo $counter;
19          echo "</td><td>";
20          echo $brush_price * $counter;
21          echo "</td></tr>";
22          $counter = $counter + 10;
23      }
24      echo "</table>";
25  ?>
26  </body>
27  </html>
```

Output:



Quantity	Price
10	50
20	100
30	150
40	200
50	250
60	300
70	350
80	400
90	450

Pretty cool, huh? The loop created a new table row and its respective entries for each quantity, until our counter variable grew past the size of 100. When it grew past 100 our conditional statement failed and the loop stopped being used. Let's review what is going on.

1. We first made a $brush_price and $counter variable and set them equal to our desired values.
2. The table was set up with the beginning table tag and the table headers.
3. The while loop *conditional statement* was checked, and $counter (10) was indeed smaller or equal to 100.
4. The code inside the while loop was executed, creating a new table row for the price of 10 brushes.
5. We then added 10 to $counter to bring the value to 20.
6. The loop started over again at step 3, until $counter grew larger than 100.
7. After the loop had completed, we ended the table.

do…while

This loop variant is very similar to the while loop. However there is one important difference. With do while loop the code block will be executed at least once. This is because in case of a do while loop PHP checks the condition only after the first iteration. It is clearly visible from the syntax:

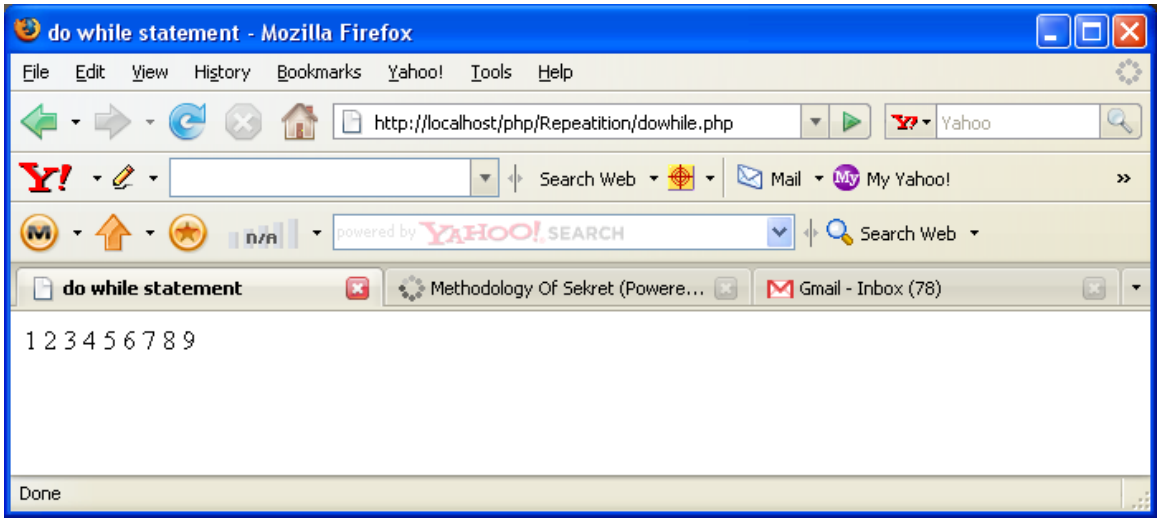
General format of *for* structure.

```
do {
code block
} while (condition)
```

Besides this there is no more difference to the basic while loop. Here is the example how to use it:

```
7
8   <body>
9   <?php
10          $i=1;
11          do {
12          echo " $i ";
13          $i++;
14          } while ($i<=9)
15  ?>
16  </body>
17  </html>
18
```

Output:



The last loop structure in PHP is the foreach. This is a special loop as you can use it only for arrays. The goal of the foreach loop to iterate over each element of an array. If you try to use it with a normal variable you will get an error.

foreach

General format of foreach structure.

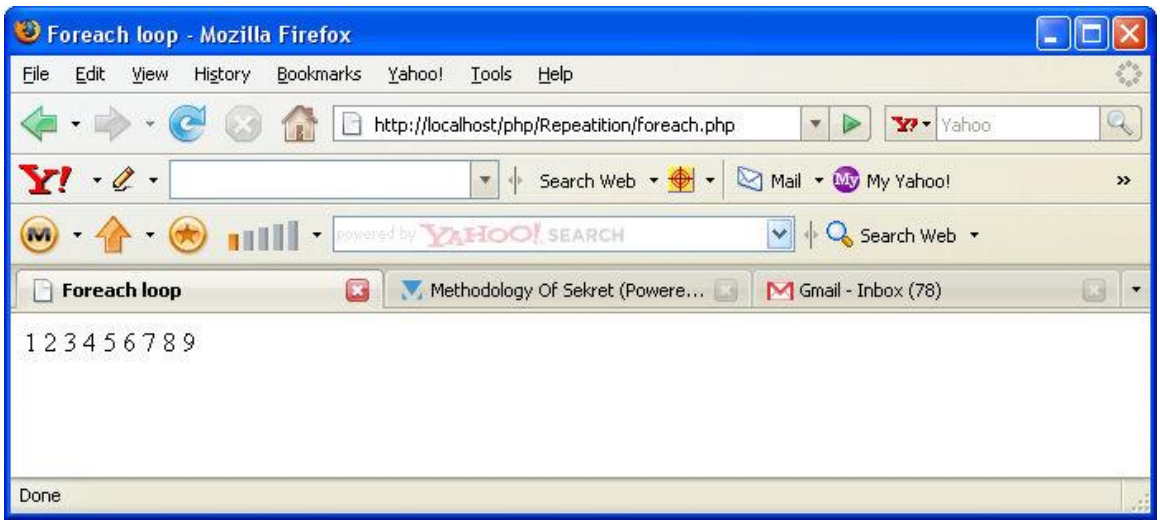
```
foreach (array as $value)
code executed;
```

It means that in each iteration the actual array value will be copied to the $value variable and you can use it in the code executed.

Example using the for each loop:

```
7
8   <body>
9   <?php
10        $myList = array(1,2,3,4,5,6,7,8,9);
11
12        foreach ($myList as $value) {
13        echo " $value ";
14        }
15  ?>
16  </body>
17  </html>
18
```

Output:

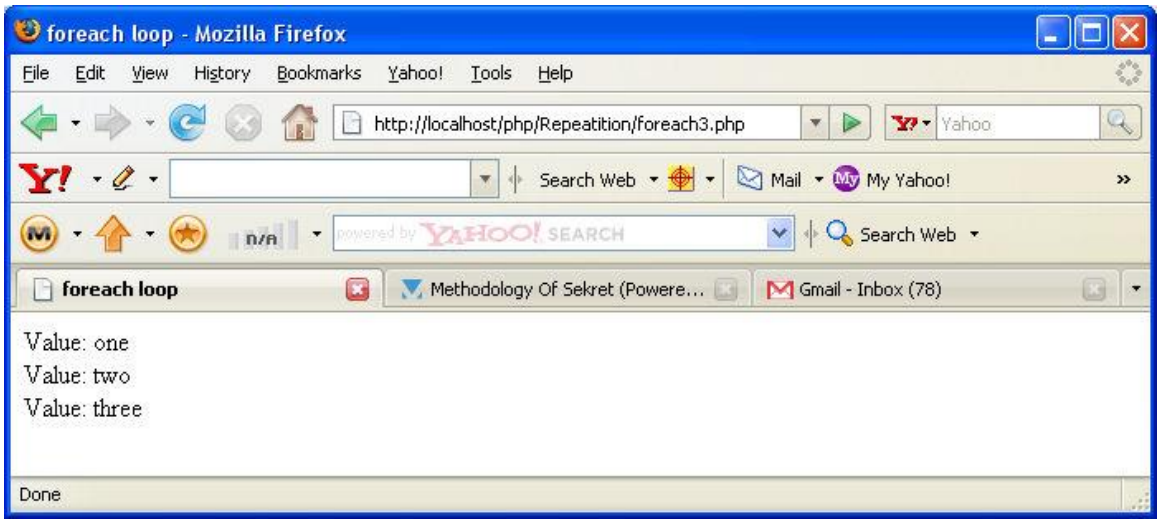


Another simple example for the *foreach* loop:

```
7
8   <body>
9   <?php
10          $arr=array("one", "two", "three");
11
12          foreach ($arr as $value)
13          {
14            echo "Value: " . $value . "<br />";
15          }
16  ?>
17  </body>
18  </html>
19
```

Well as we can see here we declared $arr as our test variable. And we declared the value of the variable in the array. And it is “one”, “two”, and “three”. So from the code also we can see that the array that contain in the variable $arr is being copied into the variable $value using the foreach loop ($arr as $value).And the last thing we see here it echo the variable $echo and the output will be :

Output:



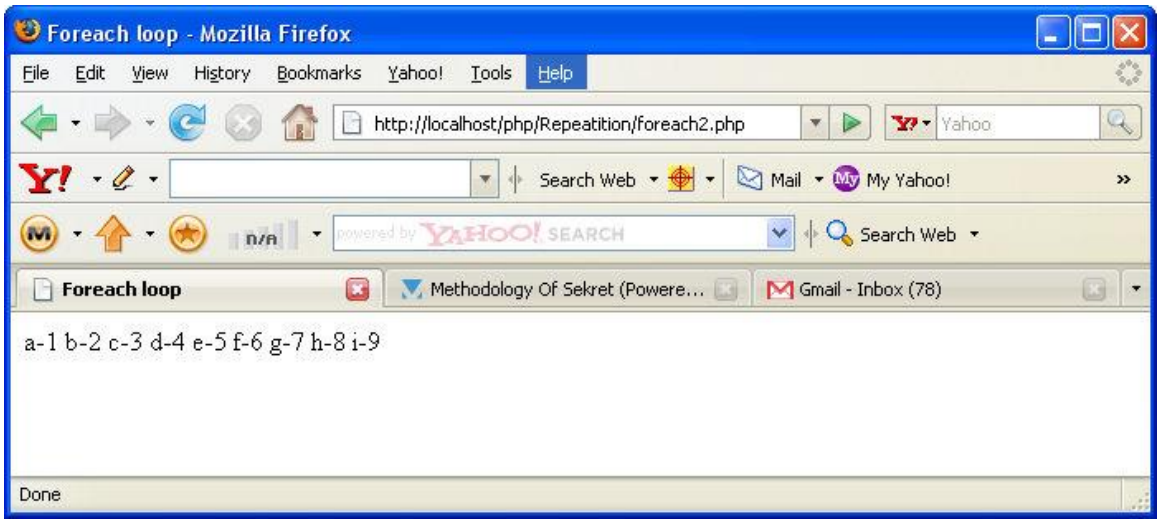
There is an alternative syntax of the foreach loop to handle associative arrays. You can use this if you want to know not only the actual element value but the key as well. The syntax is:

```
foreach (array as $key => $value)
code executed;
```

In this case you can use both information in your code execute like this:

```
<body>
<?php
        $myList = array('a'=>1,'b'=>2,'c'=>3,'d'=>4,'e'=>5,
        'f'=>6,'g'=>7,'h'=>8,'i'=>9);

        foreach ($myList as $key => $value) {
        echo " $key-$value ";
        }
?>
</body>
</html>
```

Output:



Repetition Applied Example 1 : Generating Date Input using ComboBox

In this example, we would like to show how to generate option list for date input, consist of day, month and year.

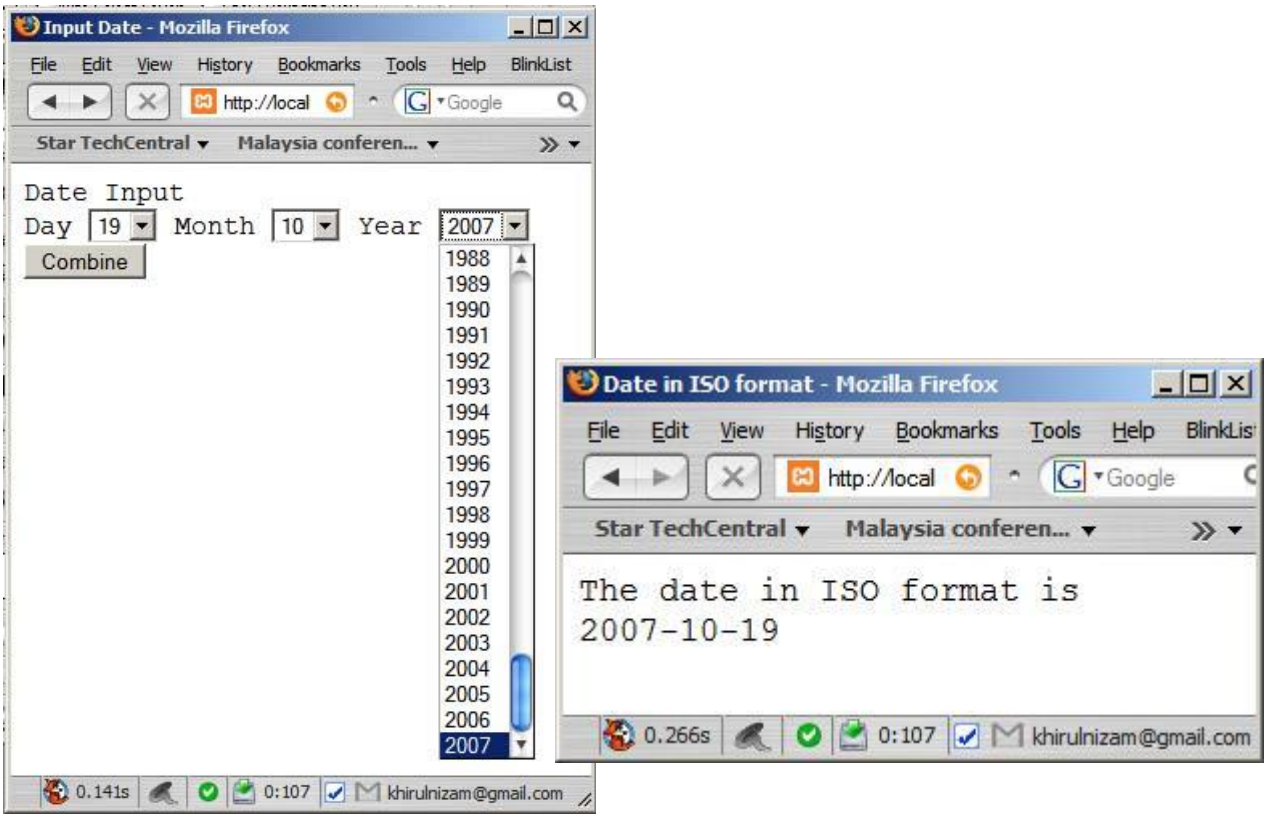
```
<html>
<head>
<title>Input Date</title>
</head>
<body>
<form name="forminsert" method=get action="isodate.php">
Date Input <br>
Day
<select name="day">
<?php
     for ($i=1;$i<=31;$i++){
          echo "<option value='$i'> $i </option>\n";
     }//end for
?>
</select>

Month
<select name="month">
<?php
     for ($i=1;$i<=12;$i++){
          echo "<option value='$i'> $i </option>\n";
     }//end for
?>
</select>

Year
<select name="year">
<?php
     for ($i=1900;$i<=date('Y');$i++){
          echo "<option value='$i'> $i </option>\n";
     }//end for
?>
</select>

<br>
<input type="submit" value="Combine">

</form>
<br>

</body>
</html>
```

From the items selected, generate ISO format for the date (yyyy-mm-dd).

```
<html>
<head>
<title>Date in ISO format</title>
</head>
```

```
<body>
The date in ISO format is <br>
<?php
        $day=$_GET['day'];
        $month=$_GET['month'];
        $year=$_GET['year'];
        $isodate = sprintf("%4d-%2d-%2d", $year, $month, $day);
        echo $isodate;
?>

</body>
</html>
```

*For further information on sprintf function, go to http://php.net/sprintf.



Exercises for Repetition in PHP

Question 1

Let say you have this array in your program;

```
$name= array ("Kerul", "Amir", "Luqman", "Muna", "Acik");
```

Create the program in PHP to print all the values in the array by using repetition.

Question 2

What is the output of the following code execution?

```
<?php
$no=5;
for ($i=0; $i<20; $i++){
      echo ("$no <br>");
      $no--;
}
?>
```

Question 3

Generate a combo box (selection list) from the array of states in Malaysia. Use for statement to generate the combo box as shown in the picture.

$state = array ("Johor","Melaka","N Sembilan", "Selangor", "Perak", "Kedah", "Pulau Pinang", "Perlis", "Kelantan", "Terengganu", "Pahang", "Sabah", "Sarawak", "Kuala Lumpur", "Labuan", "Putrajaya");

This is the sample :

additional tools for web database application development

PHP is widely known for fast server-side scripting execution. It can connect to any database servers provided the drivers are available. Most common database server used to be PHP sparring partner is MySQL. For this tutorial you need a database server, which is MySQL, and the MySQL administrator, we'll be using phpmyadmin.

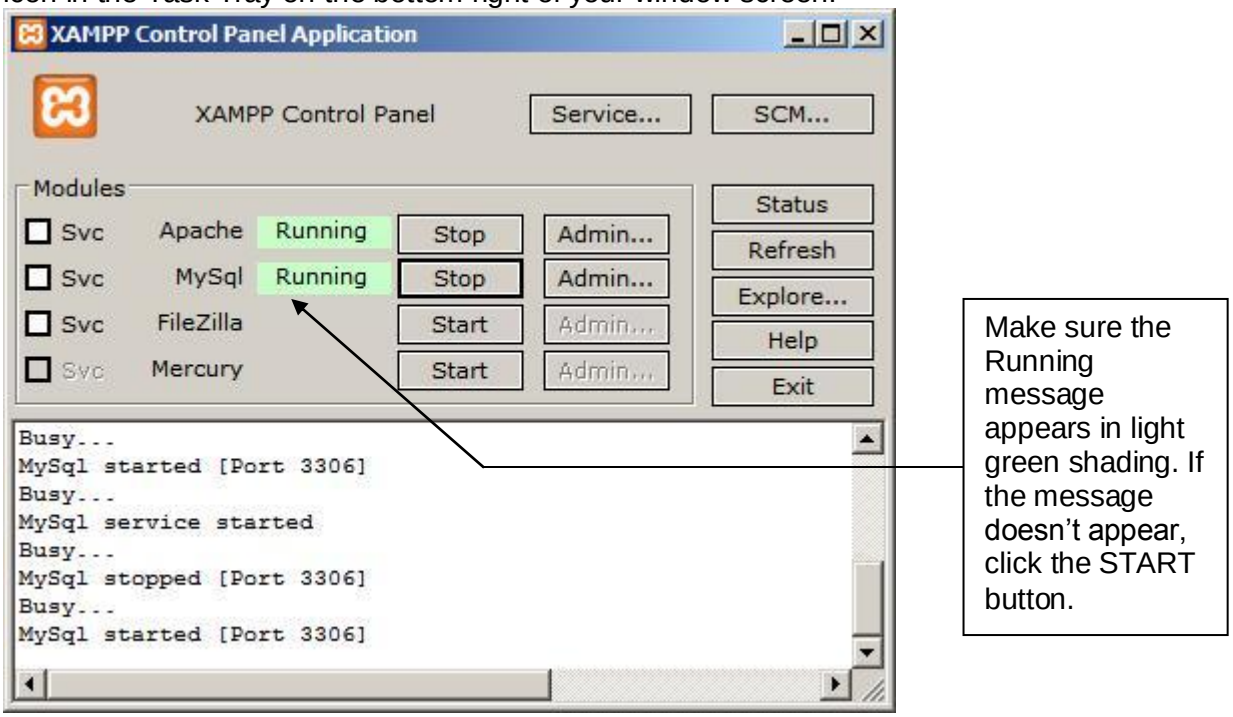
The database server, MySQL Community Edition

There are two major versions of MySQL database server, the first and the most popular is MySQL Community Server, and the second is the MySQL Enterprise Edition. Throughout this module, we'll be using the MySQL Community Server which is free.
*The MySQL Community Server version 5 is available for download at http://dev.mysql.com/downloads/mysql/5.0.html#downloads .

**Please refer http://dev.mysql.com/doc/refman/5.0/en/index.html for more information on MySQL database server.

ON the MySQL database server

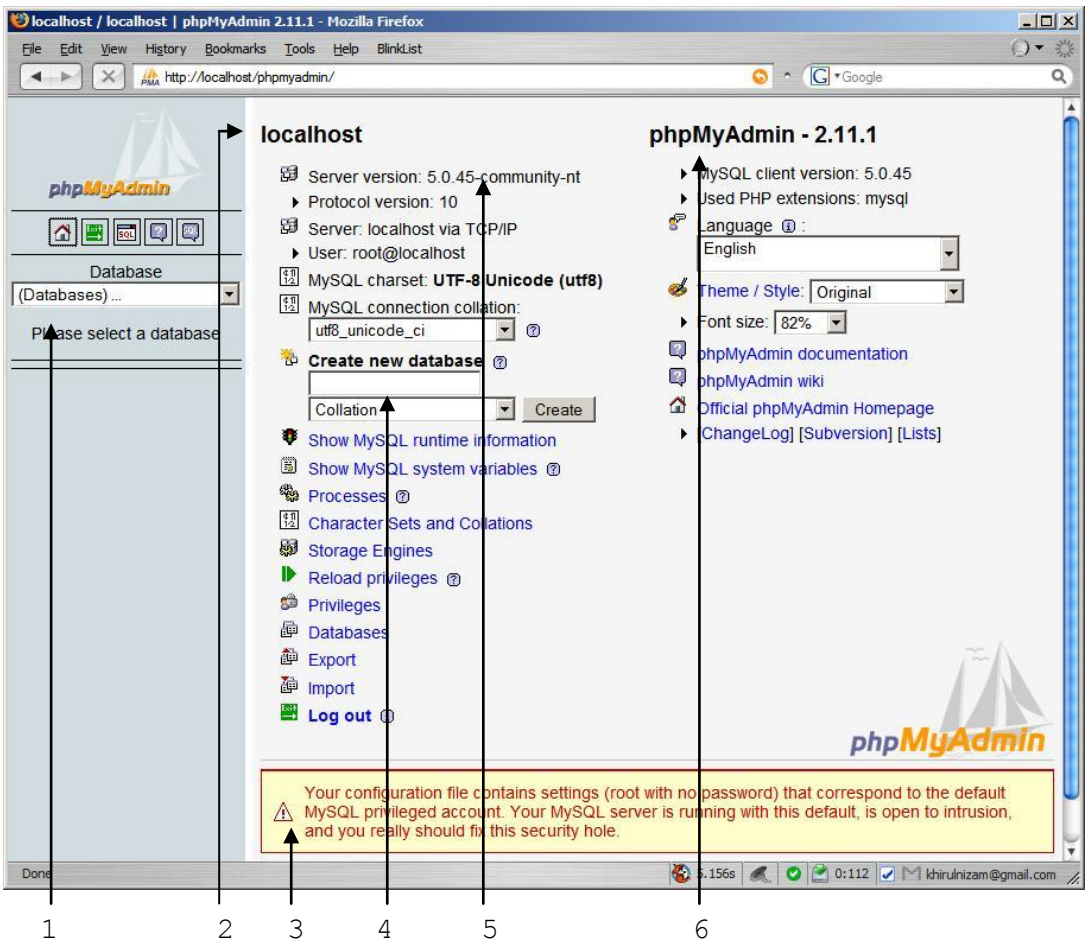
To make sure the database server is on, open the XAMPP Control Panel. Find the ( ) icon in the Task Tray on the bottom right of your window screen.



The database administrator

- phpmyadmin is an open-source web-based MySQL administrator.
- For the installation for the latest version, download the application from http://www.phpmyadmin.net and extract the zip file to the web-root.
- This tutorial is based on release v2.11.1, the default installation in XAMPP 1.6.4.
- To access from your machine; **http://localhost/phpmyadmin** .

Tutorial 1: A brief introduction to phpmyadmin main page.

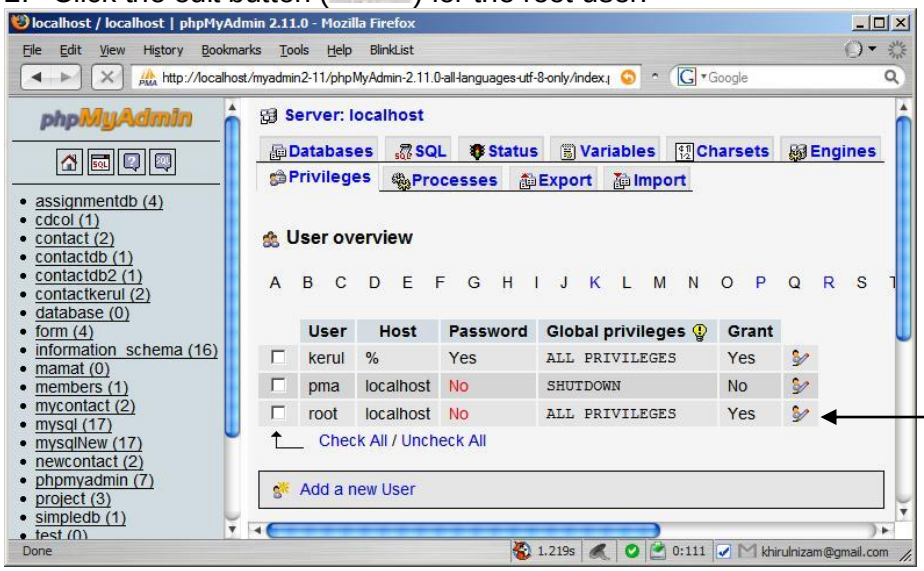


Legend :

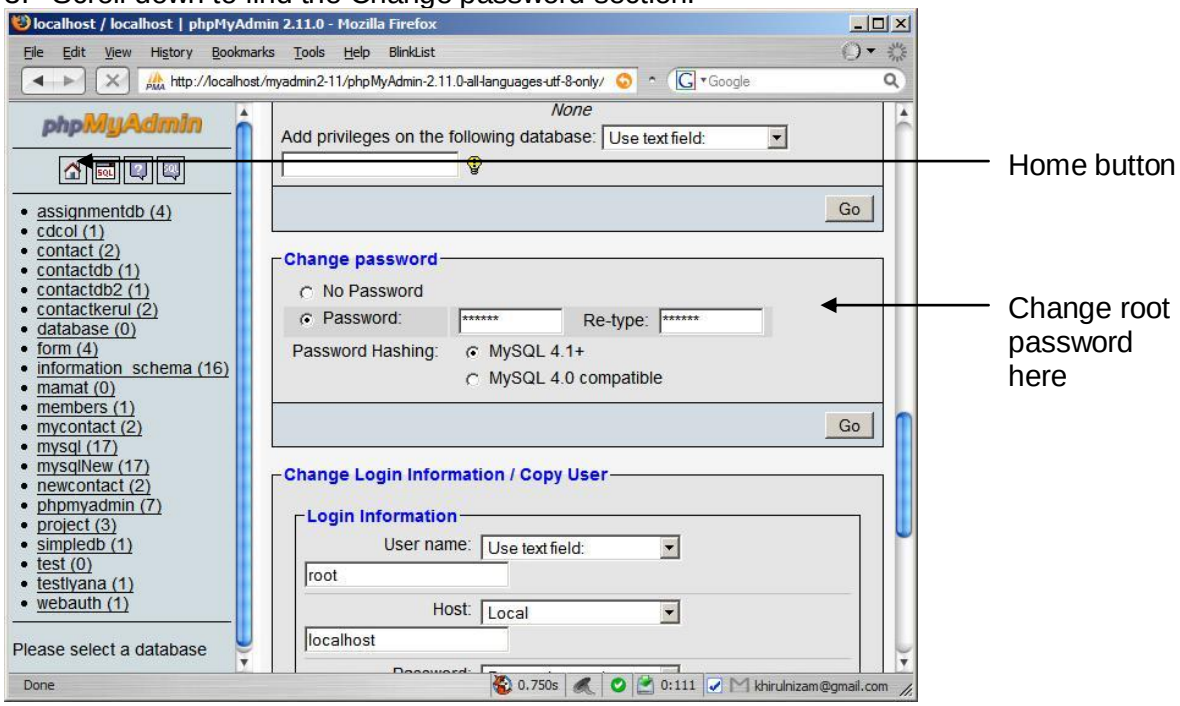
1. Listing of all databases available in the database server. Choose one of the database to administer the content. It will change into listing in the form of combo box, once you have selected a database.
2. The database address. Sometimes in the form of IP address.
3. Warning: The database server root password is default (no password). You must change the password for better security. A simple tutorial is included to change the password through phpmyadmin.
4. Type a new database name, and hit Create to create a new database. Make sure the name is unique and please avoid special characters.
5. The MySQL server version (this tutorial is using 5.0.25-community-nt).
6. The phpmyadmin version (this tutorial is using 2.11.1, default version provided in XAMPP 1.6.4).

Tutorial 2: Changing the MySQL root password.

1. Click the Privileges menu.
2. Click the edit button () for the root user.

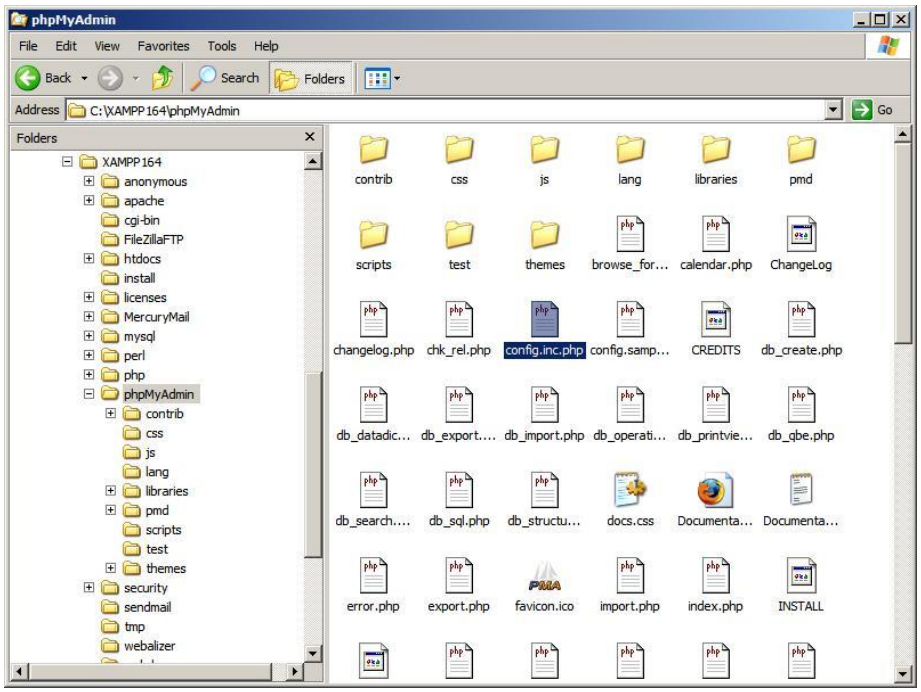


3. Scroll down to find the Change password section.



4. After you have changed the root password (let say *abc123*) try to hit Home button () , on the left side frame. You will get this message;

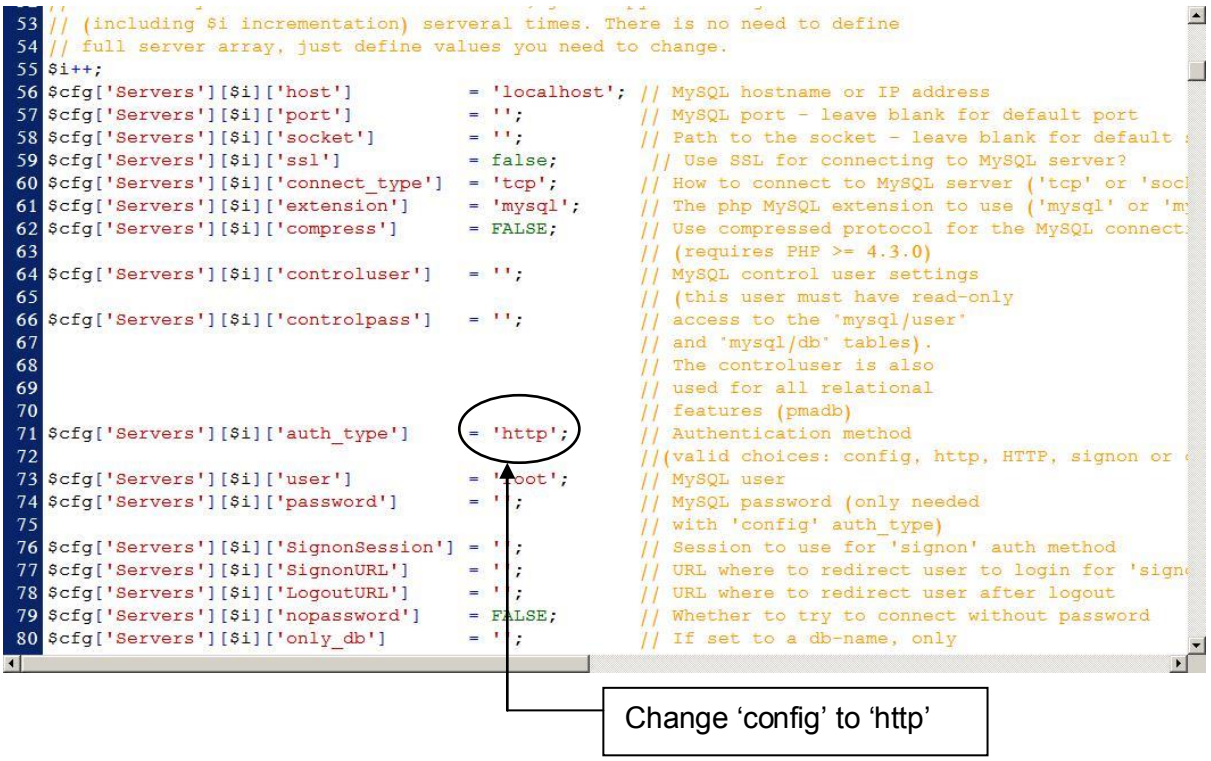
5. You need to modify the phpmyadmin configuration setting so phpmyadmin could access the MySQL database server with the new password. Note that, phpmyadmin use the privileges set in the MySQL database users table.
6. Choose the *config.inc.php* file and open with your favorite text editor.



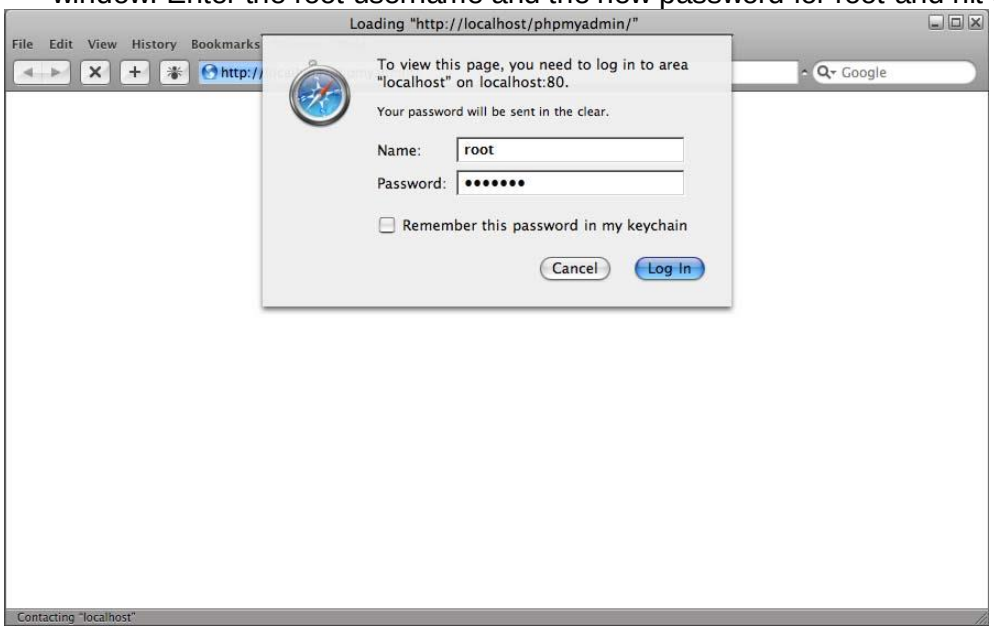
7. Edit the information near these lines :

```
$cfg['Servers'][$i]['auth_type'] = 'config';      // Authentication method
                                  //(valid choices: config, http, HTTP, signon or cookie)
$cfg['Servers'][$i]['user']          = 'root';          // MySQL user
```

8. Change *config* to *http*.



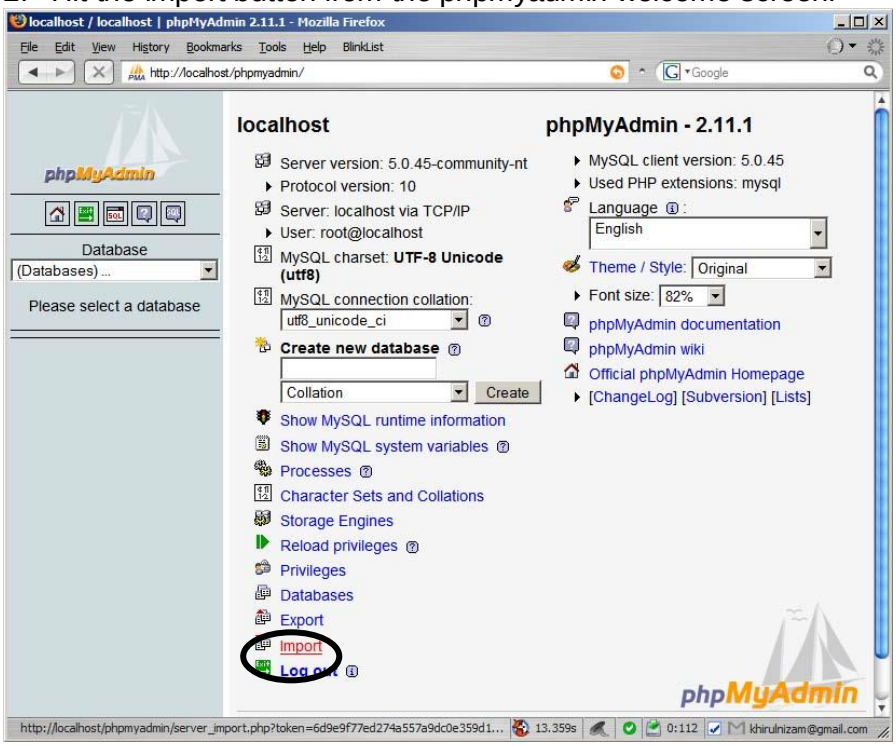
9. Go back to your browser and refresh/reload the page. You will be greeted with this window. Enter the root username and the new password for root and hit enter.



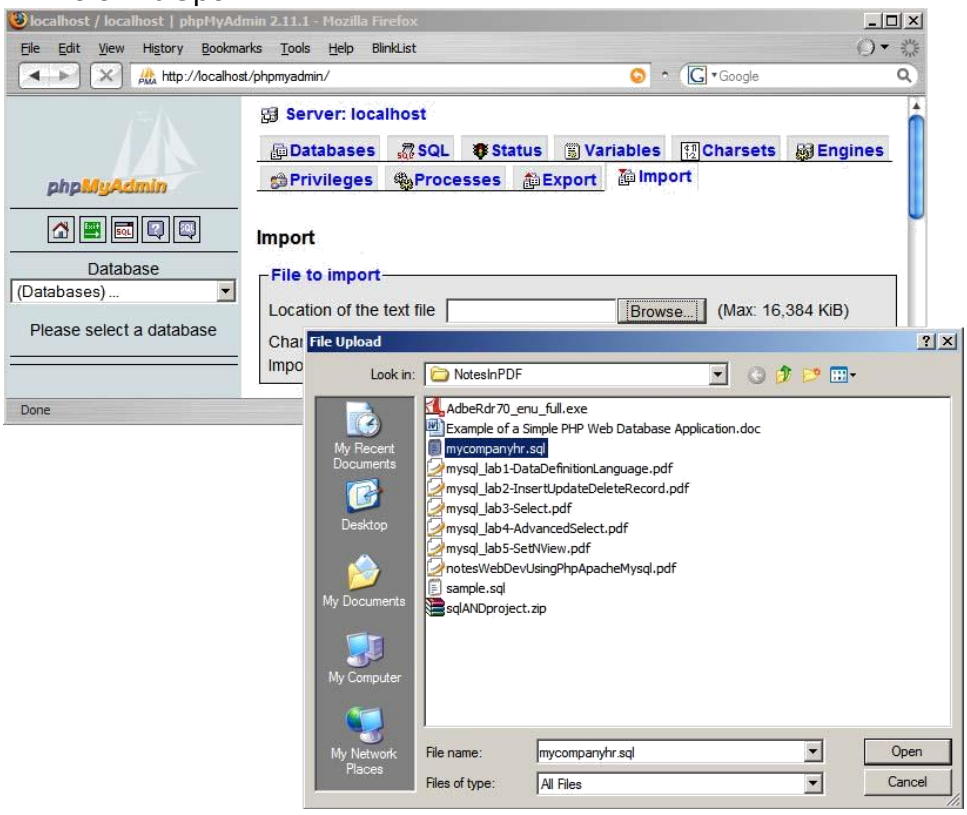
10. Then your phpmyadmin is in control again.

Tutorial 3 : Importing an existing database using phpmyadmin.

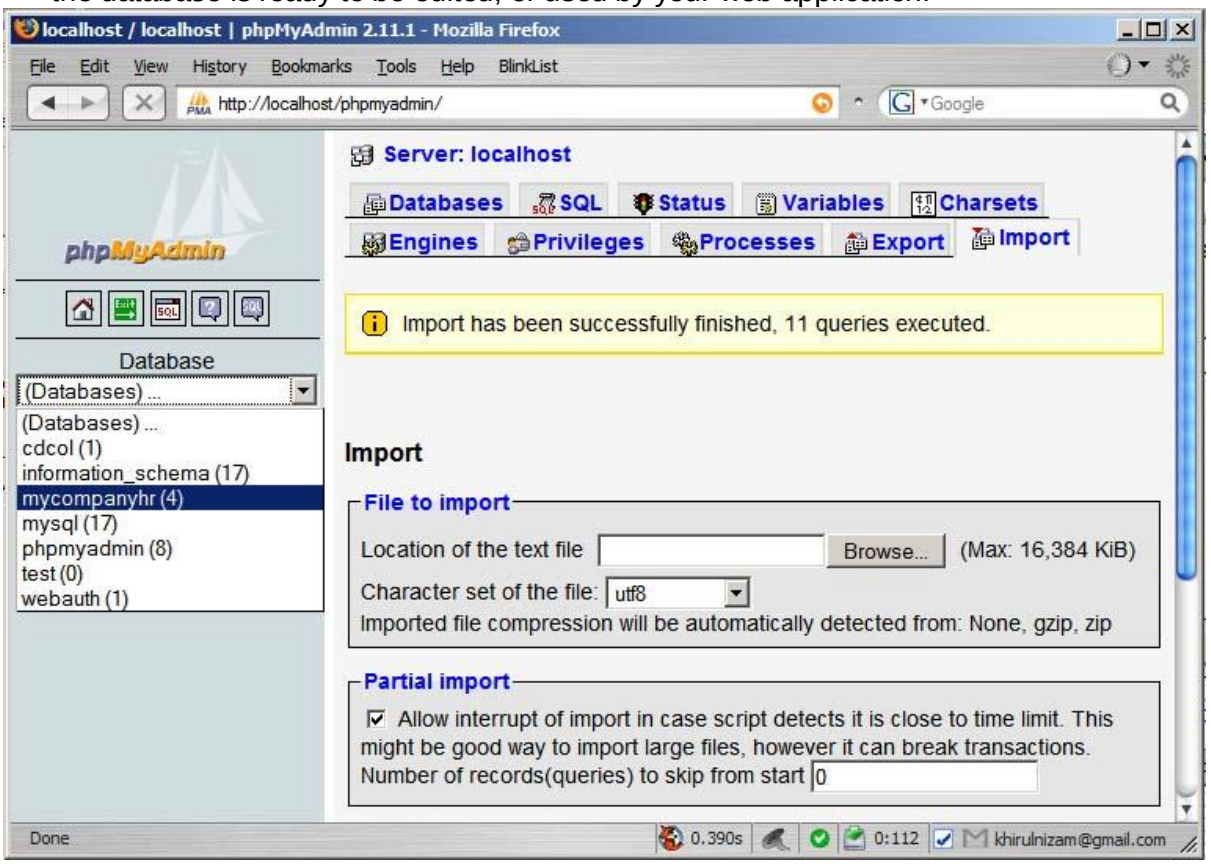
1. Download the sql file from dhost.info/kerul/sql/mycompanyhr.sql. And save the file somewhere in your computer.
2. Hit the import button from the phpmyadmin welcome screen.



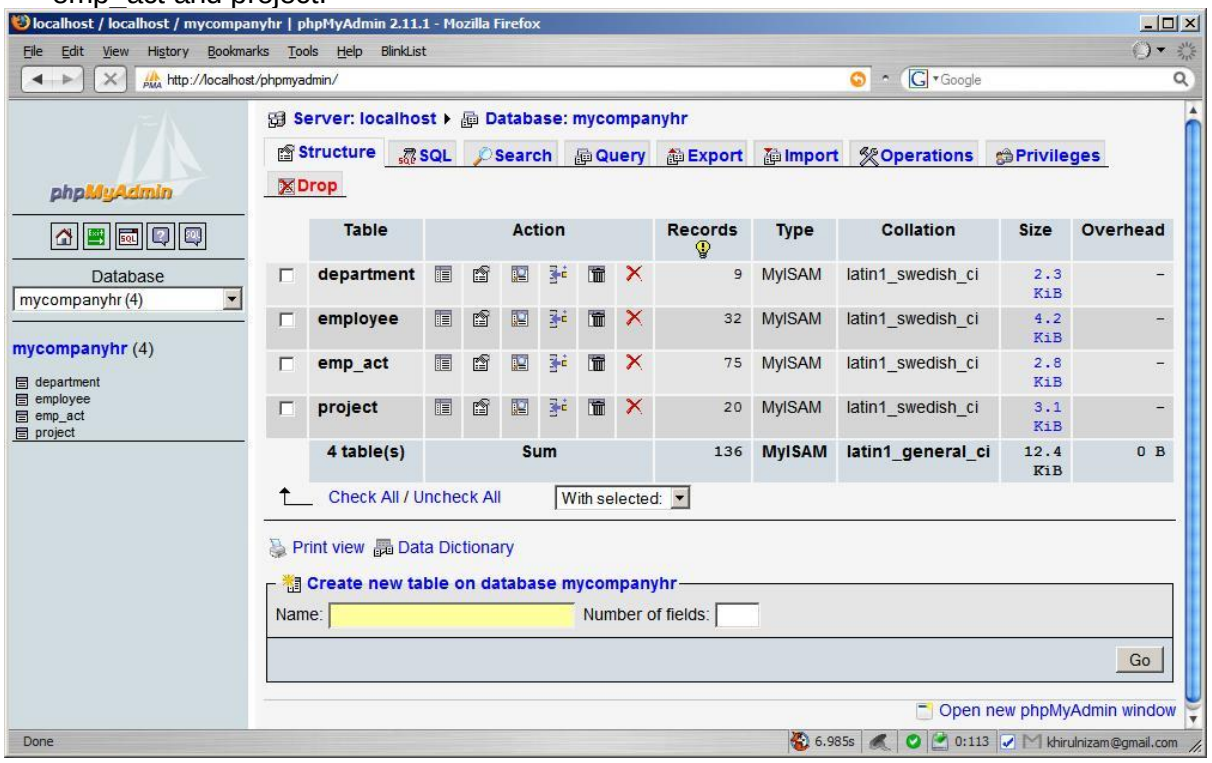
3. Click browse and find the files you have just downloaded previously. Double-click the file or hit Open.



4. If there are no sql errors, the screen should appear like this. Go to the left section of the page and click the combo box. Choose the database named mycompanyhr and the database is ready to be edited, or used by your web application.



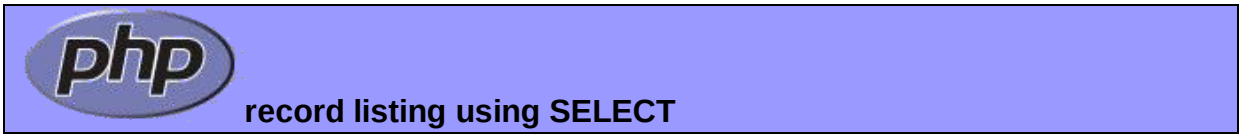
5. As you can see on the left, the database consists of 4 tables; department, employee, emp_act and project.



MySQL Improved Extension

Through out this web database development exercises, we will be covering a lot of API for MySQL database connection. MySQL has introduced an improved version of mysql functions prefixed with mysqli_ . All the functions are design to connect and manipulate information from MySQL database server version 4.1.3 and above. For further information, go to http://php.net/mysqli .

Among the functions that will be used in this tutorial are;
mysqli_connect — Open a new connection to the MySQL server.
mysqli_select_db — Selects the default database for database queries.
mysqli_affected_rows — Gets the number of affected rows in a previous MySQL operation.
mysqli_close — Closes a previously opened database connection.
mysqli_errno — Returns the error code for the most recent function call.
mysqli_error — Returns a string description of the last error.
mysqli_execute — Executes a prepared Query.
mysqli_fetch_array — Fetch a result row as an associative, a numeric array, or both.
mysqli_fetch_row — Get a result row as an enumerated array.
mysqli_num_fields — Get the number of fields in a result.
mysqli_num_rows — Gets the number of rows in a result.
mysqli_query — Performs a query on the database.
*These function list are extracted from http://php.net/mysql.



record listing using SELECT

(Disclaimer: This tutorial is based on XAMPP version 1.6.4. The result may vary if you're using different version of XAMPP).

Connect to MySQL database server, and the database using mysqli_ functions.

```
$db=mysqli_connect("localhost","root","abc123","mycompanyhr");
```

The database server address
The username for the database server
Password for the username
The database name used

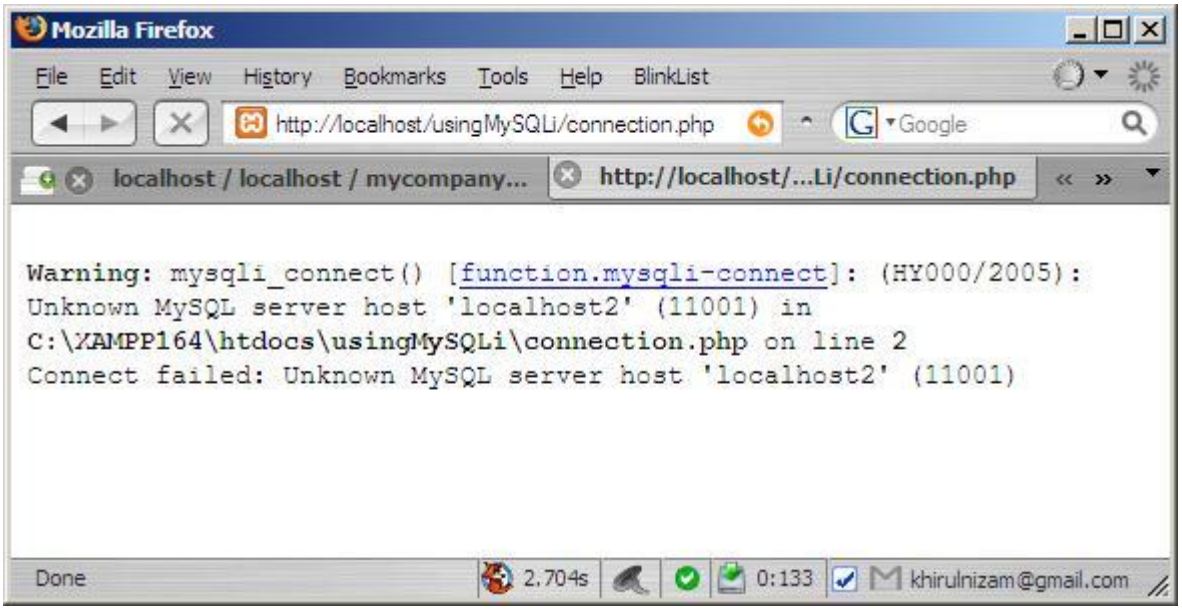
Check the connection

This function provides the connection error number.

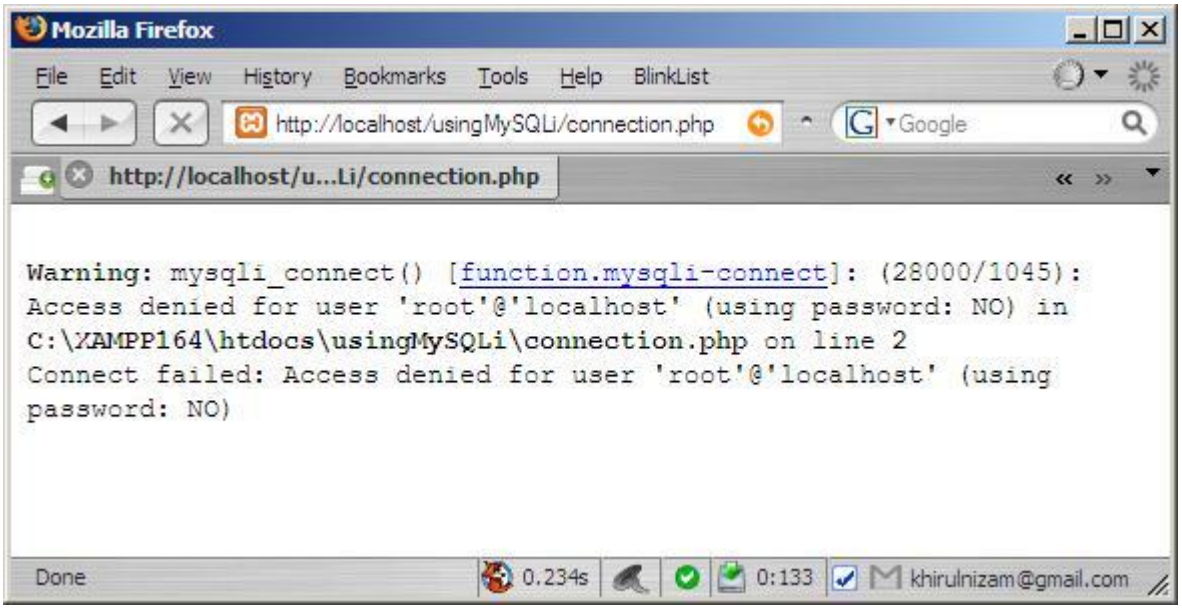
```
if (mysqli_connect_errno($db)) {
    printf("Connect failed: %s\n", mysqli_connect_error($db));
    exit();
}
```

This function provide the connection error details.

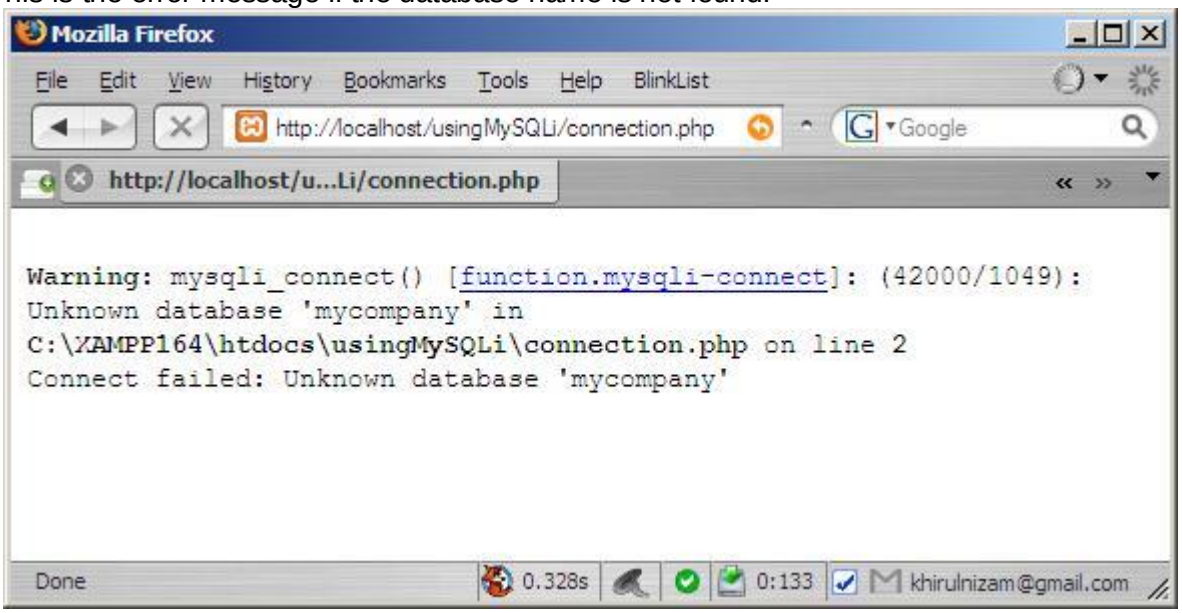
This is the error message if the database server provided in the address parameter is not found.



This is the error message if the user's password doesn't match.



This is the error message if the database name is not found.



At the end, this is the script for the database connection. Save this script in a file named *connection.php*, to be included later in other pages.

```
<?php
$db=mysqli_connect("localhost","root","abc123","mycompanyhr");
//Check connection
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error($db));
    exit();
}
?>
```

Create SQL query

```
$query="select EMPNO, FIRSTNAME, LASTNAME, DEPT, PHONENO, EMAIL
        from employee";
```

This will fetch all the records that contain only the fields listed in the query, from a table named *employee*, inside the database *mycompanyhr*.

Execute the query

```
$qr=mysqli_query($db, $query);
if($qr==false){
    echo ("Query cannot be executed!<br>");
    echo ("SQL Error : ".mysqli_error($db));
}
```

The SQL command.

The result set.

mysql_query function will return a result set consist of all the records from the table employee. However, it will return false if the sql command cannot be executed (whether is there any syntax error on the sql command, or cannot find the table specified, etc), and the error message will appear.

* An SQL **result set** is a set of rows from a database, as well as meta-information about the query such as the column names, and the types and sizes of each column.
Extracted from: http://en.wikipedia.org/wiki/Result_set

So *$qr* will hold the result set from the sql execution (pic 9.1). This result set have not appear in your page yet. It's just a visualization for what resides in the your machine's memory.

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	PHONENO
000010	CHRISTINE	HAAS	A00	3978
000020	MICHAEL	THOMPSON	B01	3476
000030	SALLY	KWAN	C01	4738
000050	JOHN	GEYER	E01	6789
000060	IRVING	STERN	D11	6423
000070	EVA	PULASKI	D21	7831
000090	EILEEN	HENDERSON	E11	5498
000100	THEODORE	SPENSER	E21	0972
000110	VINCENZO	LUCCHESSI	A00	3490
000120	SEAN	O'CONNELL	A00	2167
000130	DOLORES	QUINTANA	C01	4578
000140	HEATHER	NICHOLLS	C01	1793
000150	BRUCE	ADAMSON	D11	4510
000160	ELIZABETH	PIANKA	D11	3782
000170	MASATOSHI	YOSHIMURA	D11	2890
000180	MARILYN	SCOUTTEN	D11	1682

Pic 9.1

Check the records effected

If there is no record effected, display a message telling the user that there is no record selected from the sql execution.
mysql_num_rows() is a function to check how many records are selected from a particular sql commad. If no record is selected, it will return 0.

```
if(mysqli_num_rows($qr)==0){
      echo ("Sorry, no record fetched...<br>");
}
```

…to be continued

Fetch a record and display

If the value returned from the mysqli_num_rows() is not 0, then try fetch one record from the result set (which is *$qr*), and display.

A record.
The result set.

… continued from the previous code segment

```
else{//there is/are record(s)
      $rekod=mysqli_fetch_array($qr);
      echo (" Employee no: ".$rekod['EMPNO']."<br>");
      echo (" First name: ".$rekod['FIRSTNAME']."<br>");
      echo (" Last name: ".$rekod['LASTNAME']."<br>");
      echo (" Department code: ".$rekod['WORKDEPT']."<br>");
      echo (" Phone no: ".$rekod['PHONENO']."<br>");
}
```

`mysqli_fetch_array($qr)` is a function to get a record from the result set. The function will return a single array that contains the information for each field. The array is stored in `$rekod`.

Observe how the data is accessed using the fieldname and combined in the echo statement.

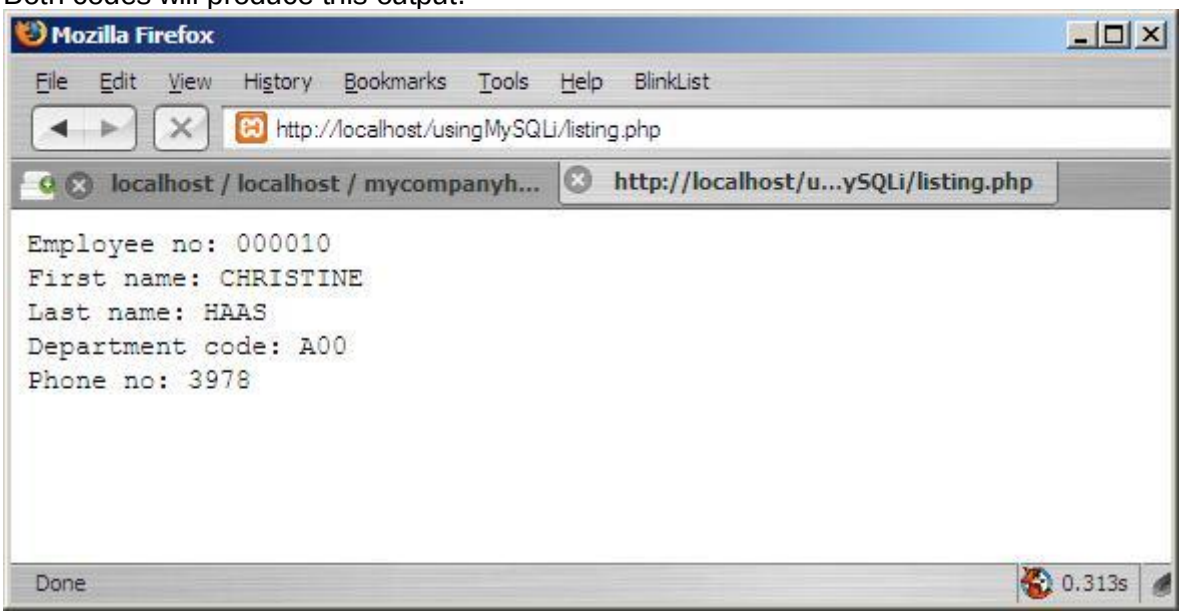
```
$rekod['EMPNO']
$rekod['FIRSTNAME']
$rekod['LASTNAME']
$rekod['WORKDEPT']
$rekod['PHONENO']
```

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	PHONENO
000010	CHRISTINE	HAAS	A00	3978
000020	MICHAEL	THOMPSON	B01	3476
000030	SALLY	KWAN	C01	4738
000050	JOHN	GEYER	E01	6789
000060	IRVING	STERN	D11	6423
000070	EVA	PULASKI	D21	7831
000090	EILEEN	HENDERSON	E11	5498

The data is accessible through associative array of index array handling. By default, the first field is indexed 0. This is how to access the record using index array handling.

```
else{//there is/are record(s)
      $rekod=mysqli_fetch_array($qr);
      echo (" Employee no: ".$rekod[0]."<br>");
      echo (" First name: ".$rekod[1]."<br>");
      echo (" Last name: ".$rekod[2]."<br>");
      echo (" Department code: ".$rekod[3]."<br>");
      echo (" Phone no: ".$rekod[4]."<br>");
}
```

Both codes will produce this output.



This is the complete code for *listing.php*. Observe the first line, it contains code inclusion from another file named *connection.php*.

```
<?php
//Include the connection details
include ("connection.php");

//Create SQL query
$query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
           from employee";

//Execute the query
$qr=mysqli_query($db,$query);
if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
}

//Check the record effected, if no records,
//display a message
if(mysqli_num_rows($qr)==0){
     echo ("No record fetched...<br>");
}
else{//there is/are record(s)
     $rekod=mysqli_fetch_array($qr);
     echo (" Employee no: ".$rekod['EMPNO']."<br>");
     echo (" First name: ".$rekod['FIRSTNAME']."<br>");
     echo (" Last name: ".$rekod['LASTNAME']."<br>");
     echo (" Department code: ".$rekod['WORKDEPT']."<br>");
     echo (" Phone no: ".$rekod['PHONENO']."<br>");
}
?>
```

Multiple Records Listing

To list all the records selected from the query, you just need a repetition structure to redo the fetch and display record for all the records available in the result set.

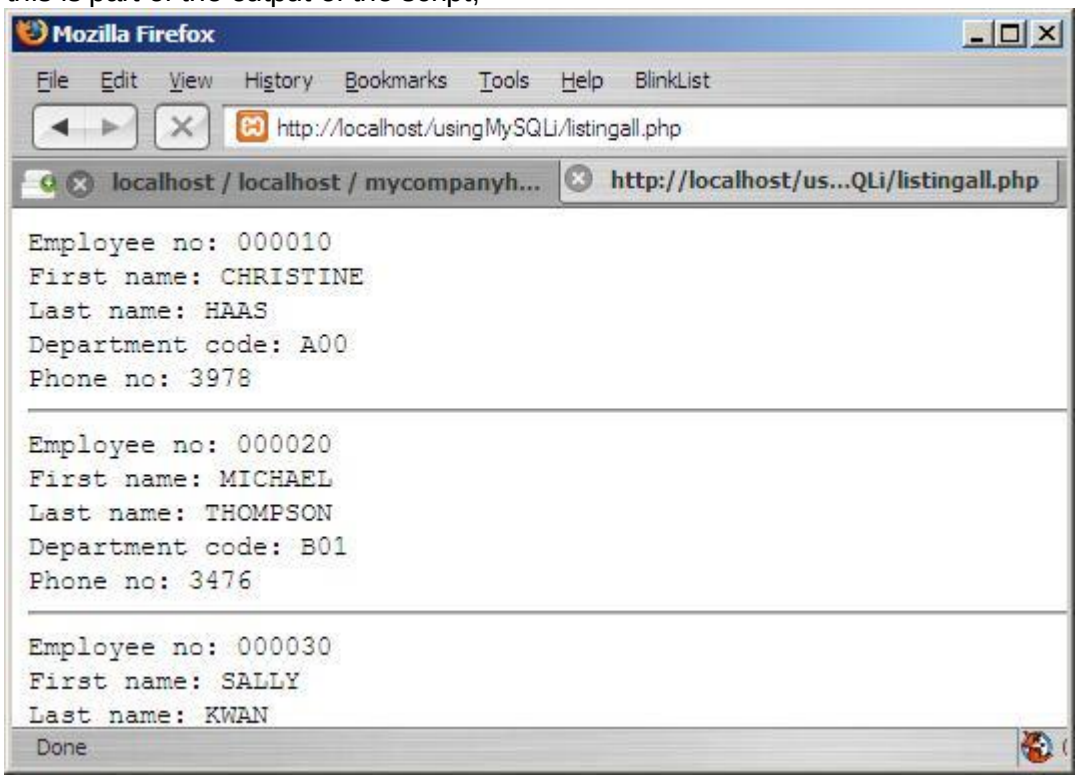
The script will fetch a record, display all the data, and do the same procedure again until the end of the result set. If the returned value from the `mysqli_fetch_array()` function is false, this happens when it comes to the end of result set, the repetition will stop.

```
if(mysqli_num_rows($qr)==0){
     echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
     while ($rekod=mysqli_fetch_array($qr)){//redo to other records
          echo (" Employee no: ".$rekod['EMPNO']."<br>");
          echo (" First name: ".$rekod['FIRSTNAME']."<br>");
          echo (" Last name: ".$rekod['LASTNAME']."<br>");
          echo (" Department code: ".$rekod['WORKDEPT']."<br>");
          echo (" Phone no: ".$rekod['PHONENO']."<hr>");
     }//end of records
}//end if there are records
```

This is the full script of the page (*listingall.php*);

```
<?php
//Include the connection details
include ("connection.php");

//Create SQL query
$query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
           from employee";

//Execute the query
$qr=mysqli_query($db,$query);
if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
}

//Check the record effected, if no records, display a message
if(mysqli_num_rows($qr)==0){
     echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
     while ($rekod=mysqli_fetch_array($qr)){//redo to next records
          echo (" Employee no: ".$rekod['EMPNO']."<br>");
          echo (" First name: ".$rekod['FIRSTNAME']."<br>");
          echo (" Last name: ".$rekod['LASTNAME']."<br>");
          echo (" Department code: ".$rekod['WORKDEPT']."<br>");
          echo (" Phone no: ".$rekod['PHONENO']."<hr>");
     }//end of records
}//end if there are records

?>
```

And, this is part of the output of the script;



Populating the Records inside a Table

The output in the previous exercise is not neat enough. It's more user friendly if we use table to manage all the records to be displayed in the web page.

We need table with 5 columns, since we have 5 fields to display. If we have more fields, just add another column.

The header of the table is situated before the repetition to extract and display the record.

```
if(mysqli_num_rows($qr)==0){
      echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
?>
<table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>


<?php
      while ($rekod=mysqli_fetch_array($qr)){//redo to other records
?>
      <tr>
            <td><?php echo $rekod['EMPNO']; ?></td>
            <td><?php echo $rekod['FIRSTNAME']; ?></td>
            <td><?php echo $rekod['LASTNAME']; ?></td>
            <td><?php echo $rekod['WORKDEPT']; ?></td>
            <td><?php echo $rekod['PHONENO']; ?></td>
      </tr>
<?php
      }//end of while
?>
</table>
<?php
}//end if there are records

?>
```

The table header

The data displayed inside the table's cells

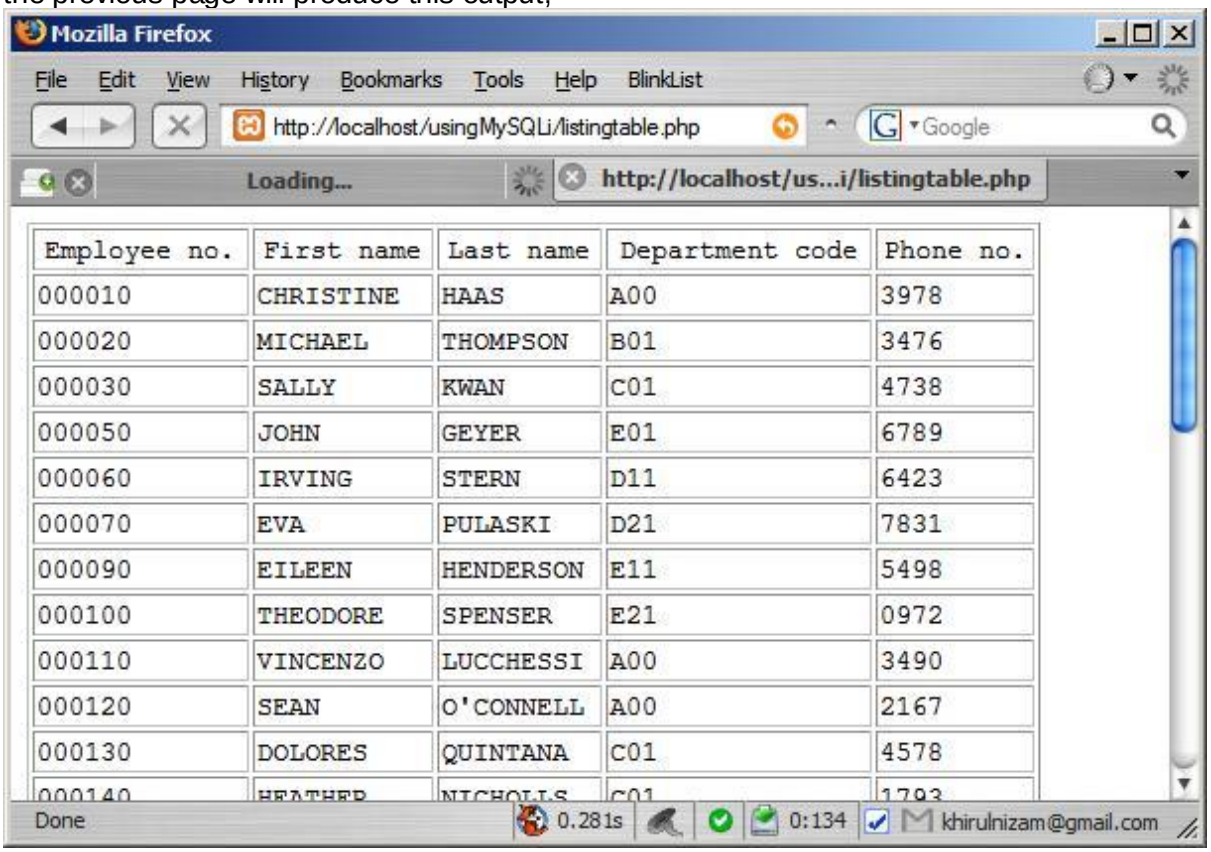
There is much simpler code to display a value from a variable, without having to use echo command.

```
<?=$var ?>
```

So we can simply change all the echo (only to display a variable value) with these;

```
            <td><?=$rekod['EMPNO']?></td>
            <td><?=$rekod['FIRSTNAME']?></td>
            <td><?=$rekod['LASTNAME']?></td>
            <td><?=$rekod['WORKDEPT']?></td>
            <td><?=$rekod['PHONENO']?></td>
```

Copy the script from the previous page, and save the file as *listingtable.php*. The script from the previous page will produce this output;

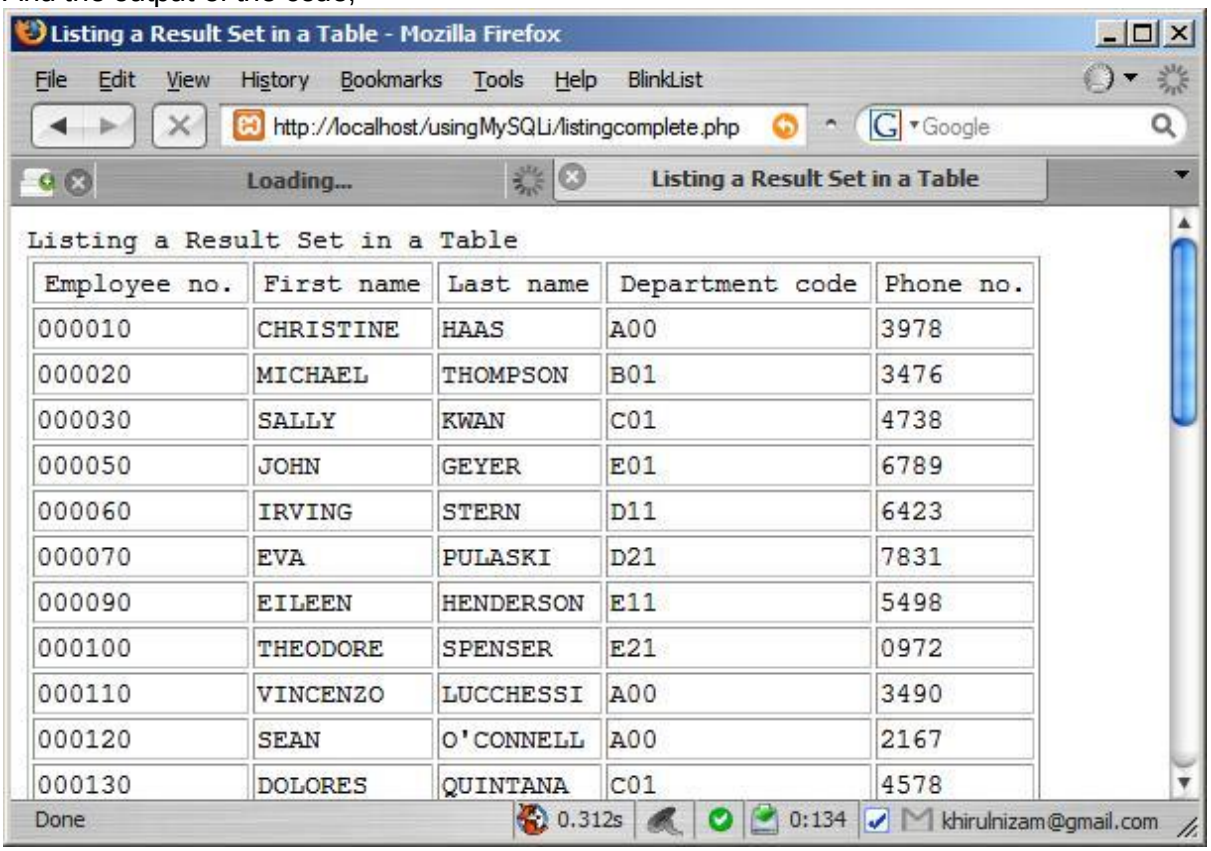


Employee no.	First name	Last name	Department code	Phone no.
000010	CHRISTINE	HAAS	A00	3978
000020	MICHAEL	THOMPSON	B01	3476
000030	SALLY	KWAN	C01	4738
000050	JOHN	GEYER	E01	6789
000060	IRVING	STERN	D11	6423
000070	EVA	PULASKI	D21	7831
000090	EILEEN	HENDERSON	E11	5498
000100	THEODORE	SPENSER	E21	0972
000110	VINCENZO	LUCCHESSI	A00	3490
000120	SEAN	O'CONNELL	A00	2167
000130	DOLORES	QUINTANA	C01	4578
000140	HEATHER	NICHOLLS	C01	1793

The complete code with HTML to list all the information in the result set to a HTML table (give the file name as *listingcomplete.php*).

```
<?php
//Include the connection details
include ("connection.php");

//Create SQL query
$query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
            from employee";

//Execute the query
$qr=mysqli_query($db,$query);
if($qr==false){
      echo ("Query cannot be executed!<br>");
      echo ("SQL Error : ".mysqli_error($db));
}
?>
<html>
<head>
<title>Listing a Result Set in a Table</title>
</head>
<body>
Listing a Result Set in a Table<br>

<?php
//Check the record effected, if no records,
//display a message
if(mysqli_num_rows($qr)==0){
      echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
?>
<table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>

<?php
      while ($rekod=mysqli_fetch_array($qr)){//redo to other records
?>
      <tr>
            <td><?=$rekod['EMPNO']?></td>
            <td><?=$rekod['FIRSTNAME']?></td>
            <td><?=$rekod['LASTNAME']?></td>
            <td><?=$rekod['WORKDEPT']?></td>
            <td><?=$rekod['PHONENO']?></td>
      </tr>
<?php
      }//end of records
?>
</table>
<?php
}//end if there are records
?>
</body>
</html>
```

And the output of the code;



Listing a Result Set in a Table

Employee no.	First name	Last name	Department code	Phone no.
000010	CHRISTINE	HAAS	A00	3978
000020	MICHAEL	THOMPSON	B01	3476
000030	SALLY	KWAN	C01	4738
000050	JOHN	GEYER	E01	6789
000060	IRVING	STERN	D11	6423
000070	EVA	PULASKI	D21	7831
000090	EILEEN	HENDERSON	E11	5498
000100	THEODORE	SPENSER	E21	0972
000110	VINCENZO	LUCCHESSI	A00	3490
000120	SEAN	O'CONNELL	A00	2167
000130	DOLORES	QUINTANA	C01	4578

Exercise

Create a database named BOOKSHOP that contains three tables namely; BOOKS, AUTHORS and PUBLISHERS. Develop a page (for each table below) by using PHP scripts to extract all the records from each of the table and display all the records inside in the page.

Database name: BOOKSHOP

isbn	title	authorid	publisherid	year	number
1047154354	Computer Security	DG	JW	2000	5
7223020711	Anti Hacker Toolkit	MS	MGH	2003	3
0201440997	Computer Security Art and Science	MB	AW	2002	3
0722274274	Hack Notes	MB	AW	2004	4
0772228691	Hacking Exposed	SML	MGH	2003	3
0772228123	Protect Your PC	HA	MGH	2002	3
1568847009	Internet Security Handbook	WS	JW	2003	3
1047154600	Computer Security: 2nd Ed	DG	JW	2004	2
0772229003	Protect Your PC II	HA	MGH	2004	3
0130293636	Visual Basic .NET	HMD	PH	2002	1
0130294022	C# How to Program	HMD	PH	2003	3

Table 1 : BOOKS

authorid	authorname	country
DG	Dieter Gollman	Germany
MS	Mike Shema	France
MB	Matt Bishop	USA
SML	Stuart McLure	USA
HA	Hackers Associate	Multiple
WS	William Stalling	UK
HMD	H. M. Dietel	USA

Table 2 : AUTHORS

publisherid	publishername	specialization
JW	John Wiley	Info tech
MGH	MacGraw Hill	info tech
AW	Addison Wesley	info tech
PH	Prentice Hall	info tech

Table 3 : PUBLISHERS

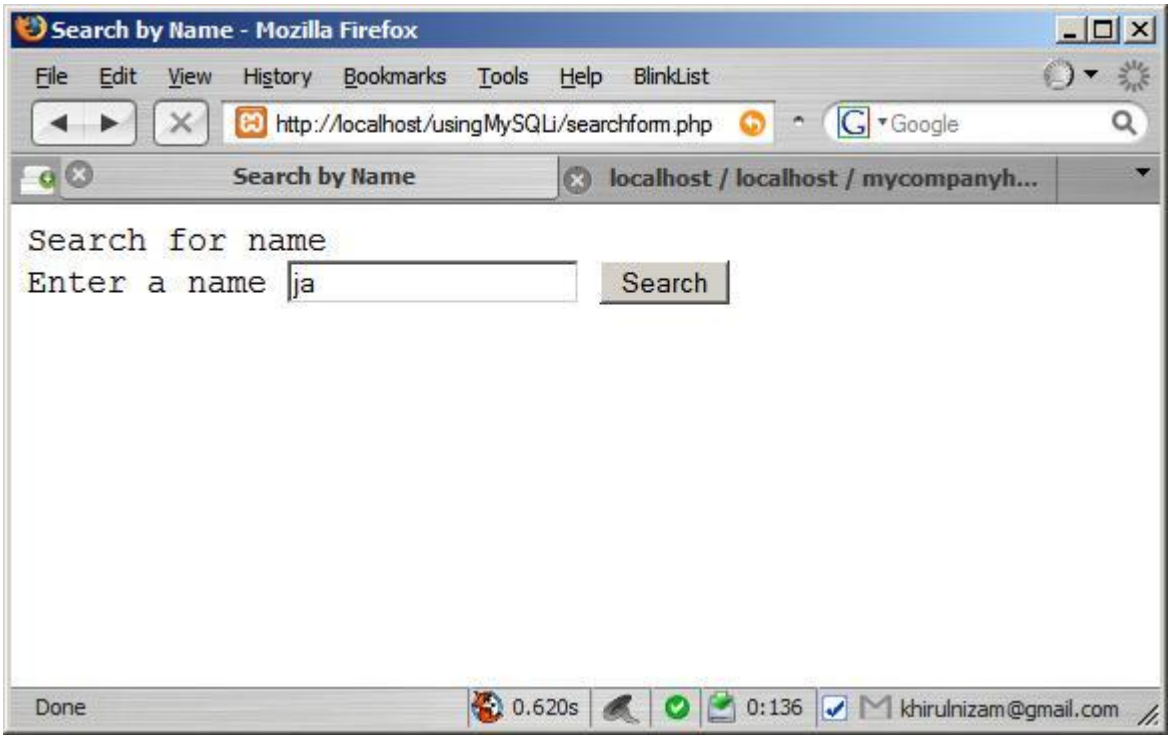
search record using SELECT

Simple search

Create a HTML form to receive the user's request. In this example, we will be doing search based on the first name, from the table *employee* of the database *mycompanyhr*.

Create the file below, and save it as *searchform.php*.

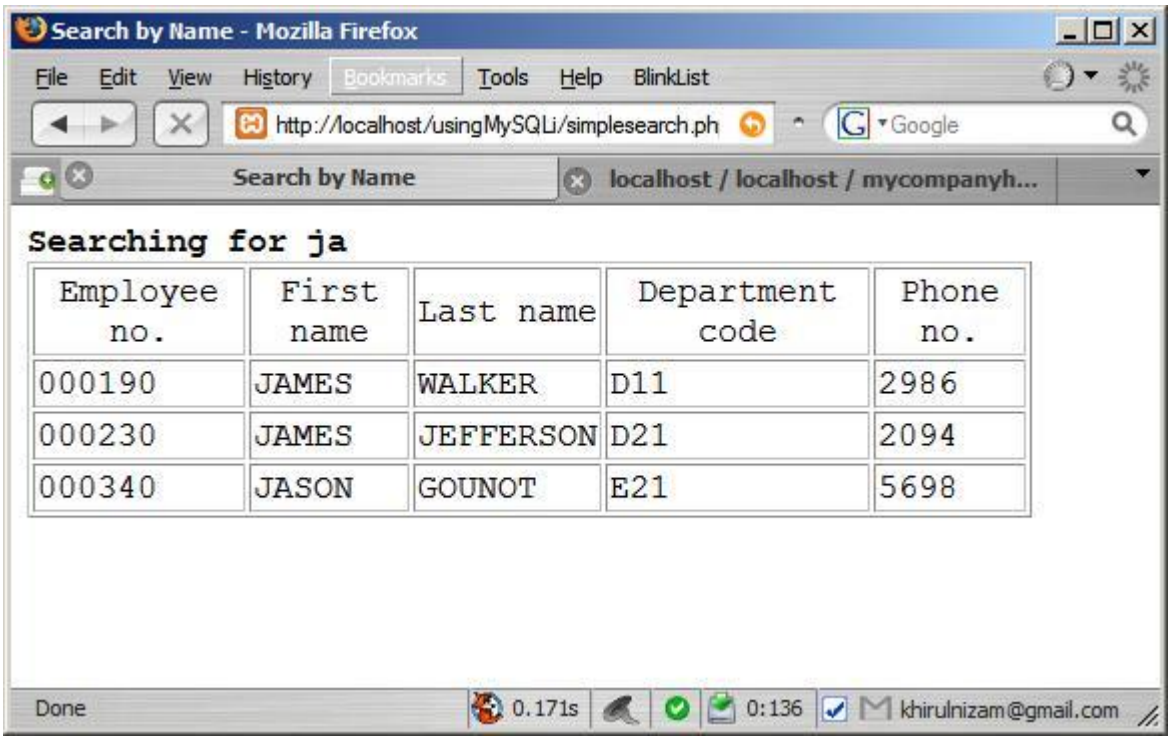
```
<html>
<head>
<title>Search by Name</title>
</head>
<body>
Search for name<br>

<form action="simplesearch.php" method="get" name="formsearch">
Enter a name <input name="txtsearch" type="text">
<input name="btnsearch" type="submit" value="Search">
</form>

</body>
</html>
```



Create another file to receive the name entered by the user. The filename if you refer to the action attribute in the form is *simplesearch.php*.

```
<html>
<head>
<title>Search by Name</title>
</head>

<body>
<?php
  $searchName=$_GET['txtsearch'];
  include ("connection.php");

  //Create SQL query, add WHERE clause to narrow listing
  $query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
          from employee
          where FIRSTNAME like '%$searchName%'";

  //Execute the query
  $qr=mysqli_query($db,$query);
  if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
  }

  //Check the record effected, if no record,
  //display a message
  if(mysqli_num_rows($qr)==0){
     echo ("No record by that name: $searchName...<br>");
  }//end no record
  else{//there is/are record(s)
  ?>
  <b>Searching for <?=$searchName?></b><br>
  <table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>

  <?php
     while ($rekod=mysqli_fetch_array($qr)){//redo to other records
  ?>
      <tr>
    <td><?=$rekod['EMPNO']?></td>
    <td><?=$rekod['FIRSTNAME']?></td>
    <td><?=$rekod['LASTNAME']?></td>
    <td><?=$rekod['WORKDEPT']?></td>
    <td><?=$rekod['PHONENO']?></td>
      </tr>
  <?php
     }//end of records
  ?>
```

```
  </table>
  <?php
  }//end if there are records
?>
</body>
</html>
```



Combine search form and the search result

It's much more user friendly if we use a single page for the user's request and the record listing.

```
<html>
<head>
<title>Listing a Result Set in a Table</title>
</head>

<body>
Search for name<br>

<form action="simplesearch.php" method="get" name="formsearch">
Enter a name <input name="txtsearch" type="text">
<input name="btnsearch" type="submit" value="Search">
</form>

<?php
$searchName=$_GET['txtsearch'];
//this will execute if a name is inserted for search
if( $searchName!= NULL){
  //Include the connection details
  include ("connection.php");

  //Create SQL query, add WHERE clause to narrow listing
  $query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
          from employee
          where FIRSTNAME like '%$searchName%'";

  //Execute the query
  $qr=mysqli_query($db,$query);
  if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
  }

  //Check the record effected, if no records,
  //display a message
  if(mysqli_num_rows($qr)==0){
     echo ("No record by that name: $searchName...<br>");
  }//end no record
  else{//there is/are record(s)
  ?>
  <b>Searching for <?=$searchName?></b><br>

  <table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>
```

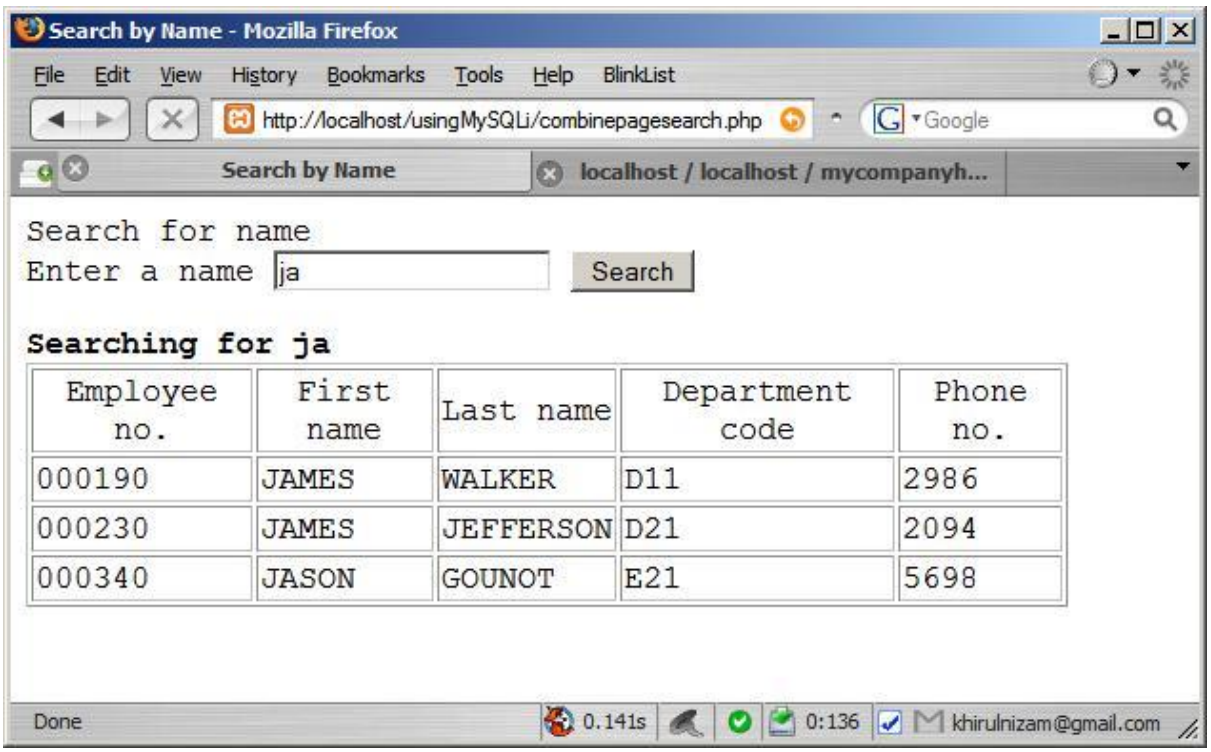
1 2 3 4 5

```
  <?php
     while ($rekod=mysqli_fetch_array($qr)){//redo to other records
  ?>
      <tr>
    <td><?=$rekod['EMPNO']?></td>
    <td><?=$rekod['FIRSTNAME']?></td>
    <td><?=$rekod['LASTNAME']?></td>
    <td><?=$rekod['WORKDEPT']?></td>
    <td><?=$rekod['PHONENO']?></td>
      </tr>
  <?php
     }//end of records
  ?>
  </table>
  <?php
  }//end if there are records

}//end if a name is inserted
?>
</body>
</html>
```

The legend for the number in circle in the previous script.

1. The form for the user to enter the name to search.
2. To extract the value entered in the textbox of the form in 1.
3. If the value in 2 is NOT NULL (user keyed in some text), the search process will execute. If the value is NULL, then nothing will happen.
4. Add the WHERE clause to select only the records that fulfill the condition.
5. Just a simple output to display the name searched by the user.

And the rest of the script is very much similar to the listing procedure in Chapter 9.



Search by using preferred criteria

To provide a much easier searching facility for the user, we can prepare a multiple option to choose. Instead of search by name only, user can choose search by their preferred field.

Create the form to receive user's query. Prepare a text box and a combo box/dropdown list. The drop down list will contain all criteria to choose.

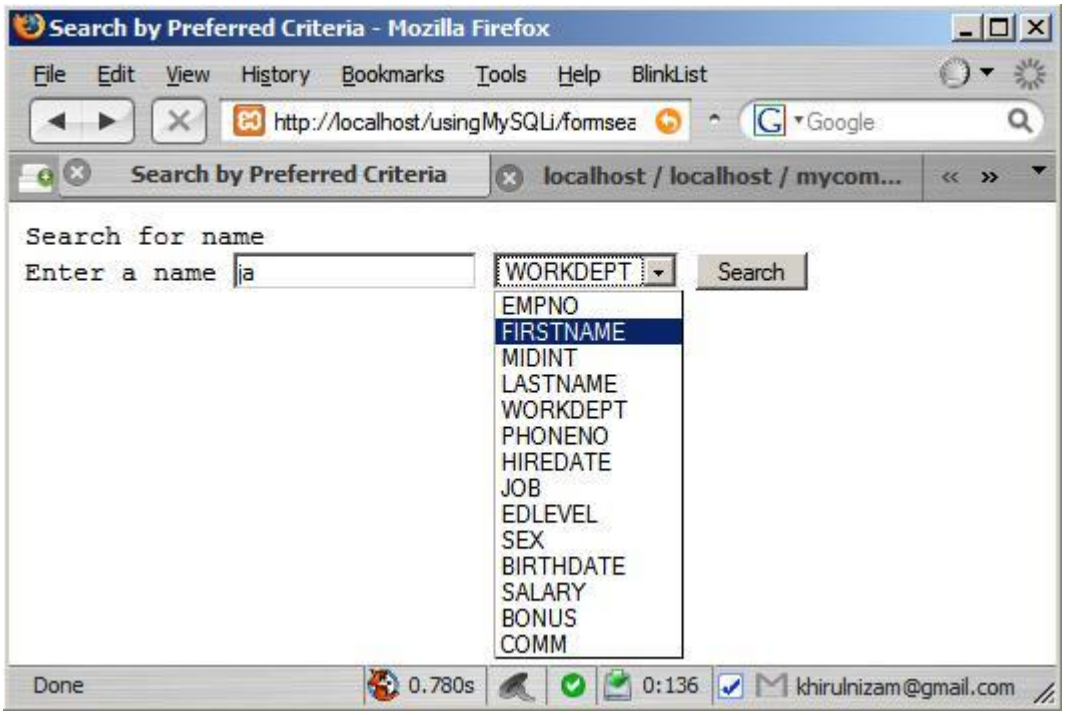
This is the code for the form, save as *formsearchcriteria.php*.

```
<html>
<head>
<title>Search by Preferred Criteria </title>
</head>

<body>
Search for name<br>

<form action="selectcriteriasearch.php" method="get"
name="formsearch">
Enter a name <input name="txtsearch" type="text">
<?php
      //Include the connection details
      include ("connection.php");
      //list all the fields in the table employee
      $qfields="SHOW COLUMNS FROM employee";
      /*the sql query above will return only the fieldnames,
      without any record*/
?>
<select name="cmbfield">
<?php
      //create a combo box for list of fields
      $rsfield=mysqli_query($db,$qfields);
      while($field=mysqli_fetch_array($rsfield)){
        $fieldname=$field['Field'];
        echo "<option value='$fieldname'>$fieldname</option>\n";
}
?>
</select>

<input name="btnsearch" type="submit" value="Search">
</form>

</body>
</html>
```

1
2

The legend for the number in circle in the previous script.

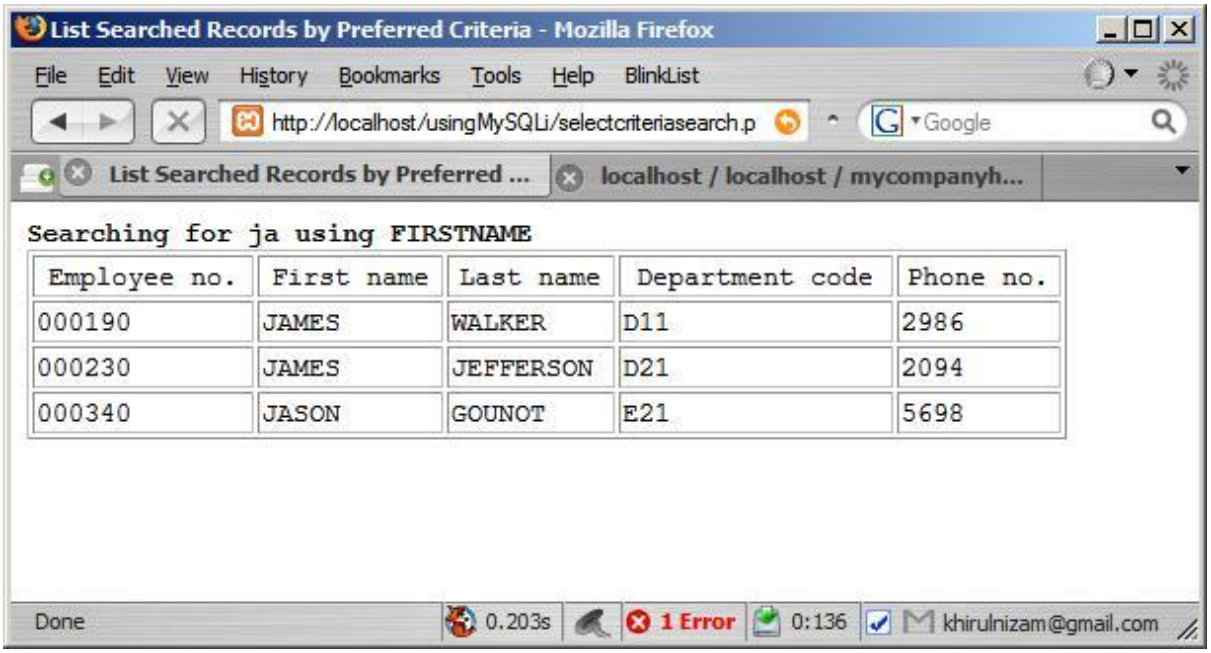
1. This SQL clause is used to list all the fields from the *employee* table.
2. Create the combo-box / drop-down-list, form the field list in 1, for the user to select their preferred criteria.



The file to process the user query is named *selectcriteriasearch.php*

```
<html>
<head>
<title>List Searched Records by Preferred Criteria </title>
</head>

<body>
<?php
//fetch the search item
$searchitem=$_REQUEST['txtsearch'];
//fetch the search criteria
$searchcriteria=$_REQUEST['cmbfield'];

//Include the connection details
include ("connection.php");
  //Create SQL query, add WHERE clause to narrow listing
  $query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
          from employee
          where $searchcriteria like '%$searchitem%'";

  //Execute the query
  $qr=mysqli_query($db,$query);
  if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
  }

  //Check the record effected, if no records,
  //display a message
  if(mysqli_num_rows($qr)==0){
     echo ("No record by that query: $searchitem...<br>");
  }//end no record
  else{//there is/are record(s)
```

```
  ?>
  <b>Searching for <?=$searchitem?> using
<?=$searchcriteria?></b><br>
  <table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>


  <?php
     while ($rekod=mysqli_fetch_array($qr)){//redo to other records
  ?>
     <tr>
          <td><?=$rekod['EMPNO']?></td>
          <td><?=$rekod['FIRSTNAME']?></td>
          <td><?=$rekod['LASTNAME']?></td>
          <td><?=$rekod['WORKDEPT']?></td>
          <td><?=$rekod['PHONENO']?></td>
     </tr>
  <?php
     }//end of records
  ?>
  </table>
  <?php
  }//end if there are records
?>
</body>
</html>
```

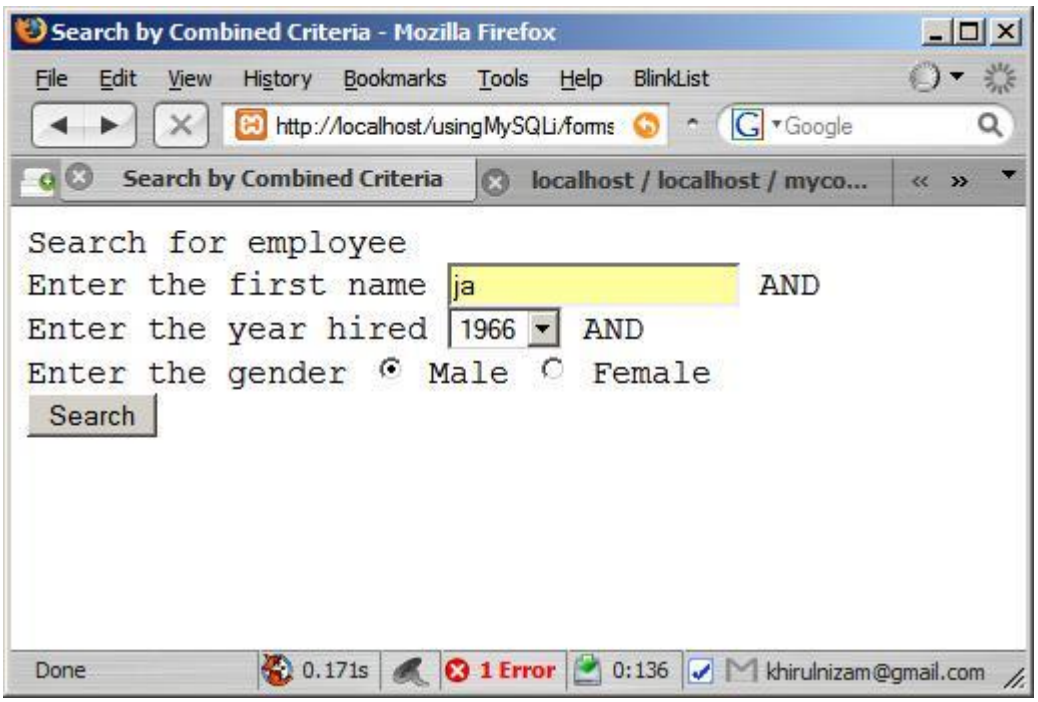


Search by using combination of criteria

Search the *employee* table with the combination of first name, hired year and gender. Create the first file to receive user's request, filename *formsearchcombinecriteria.php*.

```
<html>
<head>
<title>Search by Combined Criteria </title>
</head>

<body>
Search for employee<br>

<form action="searchcombinecriteria.php" method="get"
name="formsearch">
Enter the first name <input name="txtfirstname" type="text"> AND
<br>
Choose the year hired
    <select name="cmbyearhired">
      <?php
            for ($i=1945;$i<=date('Y');$i++){
                 echo "<option value='$i'> $i</option>";
            }
      ?>
      </select> AND <br>
Choose the gender
      <input name="rsex" type="radio" value="M"> Male
      <input name="rsex" type="radio" value="F"> Female
      <br>
<input name="btnsearch" type="submit" value="Search">
</form>

</body>
</html>
```

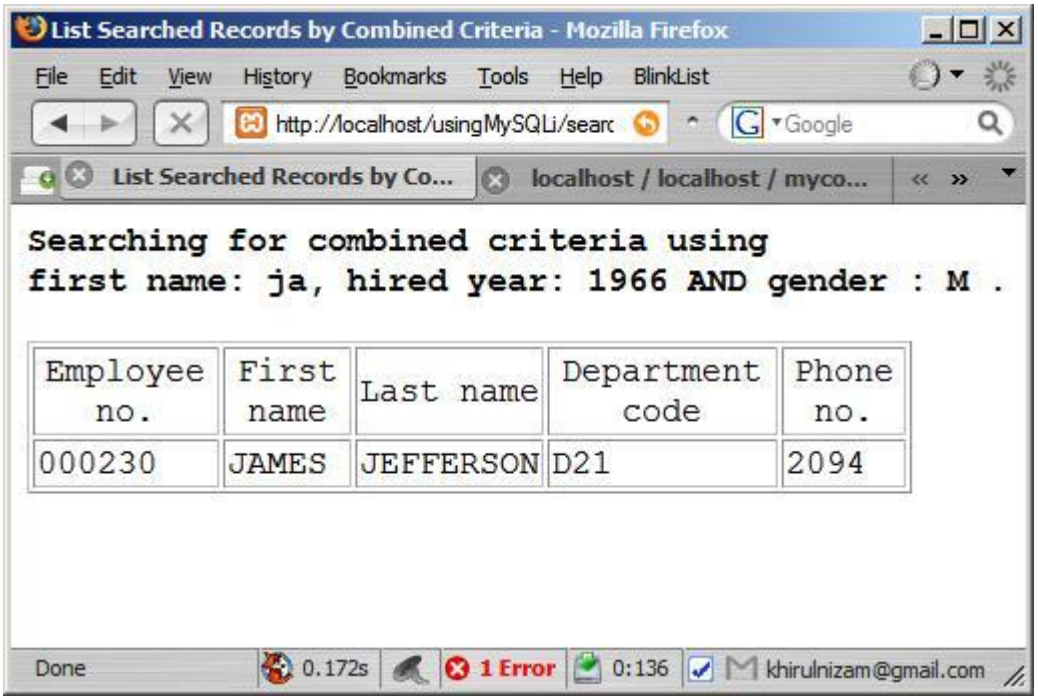
This generates the year list from 1946 to the current year.



This is the page to receive the search criteria and list the match records. The filename is *searchcombinecriteria.php* .

```
<html>
<head>
<title>List Searched Records by Combined Criteria </title>
</head>

<body>
<?php
$firstname=$_REQUEST['txtfirstname'];
$hiredyear=$_REQUEST['cmbyearhired'];
$sex=$_REQUEST['rsex'];

//Include the connection details
  include ("connection.php");
  //Create SQL query, add WHERE clause to narrow listing
  $query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO,
          SEX, HIREDATE
          from employee
          where FIRSTNAME like '%$firstname%' AND
          year(HIREDATE) = '$hiredyear' AND
          SEX = '$sex'";
  //Execute the query
  $qr=mysqli_query($db,$query);
  if($qr==false){
     echo ("Query cannot execute!<br>");
     echo ("SQL Error : ".mysqli_error($db));
  }

  //Check the record effected, if no records,
  //display a message
  if(mysqli_num_rows($qr)==0){
     echo ("No record by that query...<br>");
  }//end no record
  else{//there is/are record(s)
  ?>
  <b>Searching for  combined criteria using
  <br>first name: <?=$firstname?>,
  hired year: <?=$hiredyear?> AND
  gender : <?=$sex?> .
  </b><br>
  <table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>

  <?php
     while ($rekod=mysqli_fetch_array($qr)){//redo to other records
  ?>
     <tr>
```

```
            <td><?=$rekod['EMPNO']?></td>
            <td><?=$rekod['FIRSTNAME']?></td>
            <td><?=$rekod['LASTNAME']?></td>
            <td><?=$rekod['WORKDEPT']?></td>
            <td><?=$rekod['PHONENO']?></td>
        </tr>
    <?php
        }//end of records
    ?>
    </table>
    <?php
    }//end if there are records
?>
</body>
</html>
```



inserting a new record using INSERT

Simple Record Insertion

We are still using the *employee* table from the *mycompanyhr* database. For simple insertion, all the information will be keyed-in using the textbox. The user will plainly insert the data without any validation process, or automated input. Later in this chapter, we will discuss on how to do validation, and automation on the input.

We will not be using all the fields in the table *employee*, instead we will only choose EMPNO, FIRSTNAME, LASTNAME, WORKDEPT and PHONENO. For this simple insert exercise, you need to create the following form. The form has been arranged neatly in a table.

Filename: ***forminsert.php***

```
<html>
<head><title>Insert Process</title></head>
<body>
<strong>Insert a new Employee </strong><br>
<form name="insert" action="insertrecord.php" method="get">

<table border="0">
  <tr>
    <td width="40%">Employee no</td>
    <td width="60%"><input name="EMPNO" type="text"></td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Work department</td>
    <td><input name="WORKDEPT" type="text" maxlength="3"></td>
  </tr>
  <tr>
    <td>Phone ext no</td>
    <td><input name="PHONENO" type="text" maxlength="4"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="save" type="submit" value="Save Record"></td>
  </tr>
</table>
</form>
</body>
</html>
```

Observe closely on Work department and Phone extension number input there is an additional <input> properties which is the *maxlength*. This properties will restrict only the defined length of characters can be entered by the user. For example in **<input name="WORKDEPT" type="text" maxlength="3">**. The maximum number of characters is only three. This is because all the department code consist of three characters only.

General format of SQL insert command:

```
INSERT INTO table_name (field_list)
        VALUES (values to be inserted in the respective fields)
```

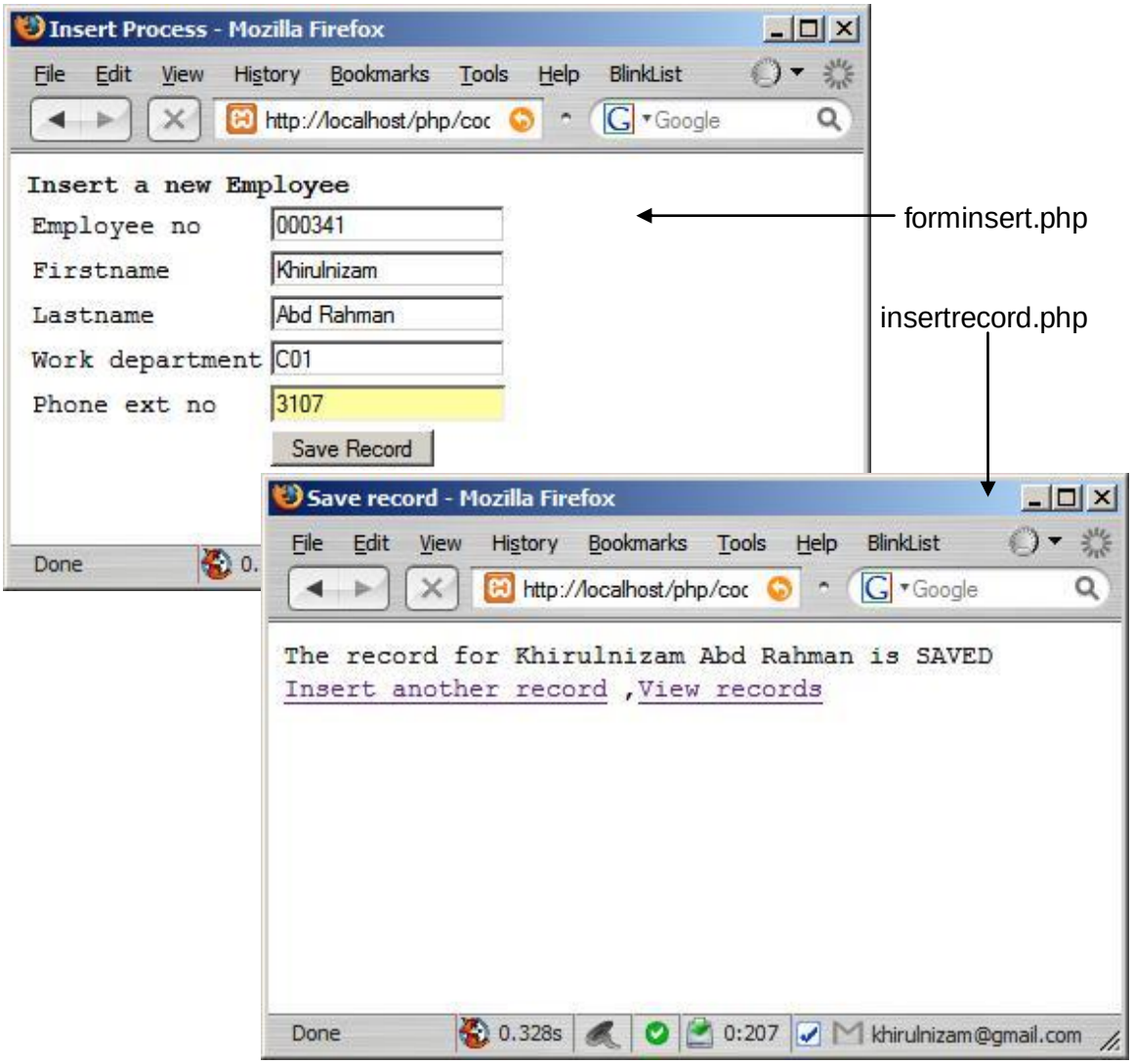
Example:

```
INSERT INTO employee
      (EMPNO, FIRSTNAME, LASTNAME,WORKDEPT, PHONENO)
      values ('000345','Khirulnizam','Abd Rahman','C01','3017')
```

The target file for the form is ***insertrecord.php*** as defined in form's action.

```
<?php
include "connection.php";

$EMPNO=$_GET['EMPNO'];
$FIRSTNAME=$_GET['FIRSTNAME'];
$LASTNAME=$_GET['LASTNAME'];
$WORKDEPT=$_GET['WORKDEPT'];
$PHONENO=$_GET['PHONENO'];

$sql="insert into employee
      (EMPNO, FIRSTNAME, LASTNAME,WORKDEPT, PHONENO)
      values ('$EMPNO','$FIRSTNAME','$LASTNAME',
      '$WORKDEPT','$PHONENO')";

$rs=mysqli_query($db, $sql);
?>

<html>
<head>
<title>Save record</title>
</head>
<body>
<?php
if($rs==true){
      echo "The record for $FIRSTNAME $LASTNAME is SAVED<br>";
      echo "<a href='forminsert.php'>Insert another record</a> ,";
      echo "<a href='listing.php'>View reords</a>";
}
else{
      echo "The record CANNOT be SAVED<br>";
}
?>
</body>
</html>
```

These are the screen shots of the simple insert application.



However, the simple form in the previous exercise doesn't help the user much in inserting a new record. The data entry clerk need to look for the last employee number in order to insert the new employee number. Since the EMPNO is a primary key, inserting a new record with a duplicate EMPNO will not permit the insertion of the new record. The data entry clerk also need to follow the EMPNO specification, which is must be 6 digits, and padded with zero in front of the employee number.

Users will be having problem in entering the work department of the new employee. They need to refer what is the code for each department as specified in the *department* table. This will incur another problem of entering wrong department code, or assigning the employee to a department that does not exist.

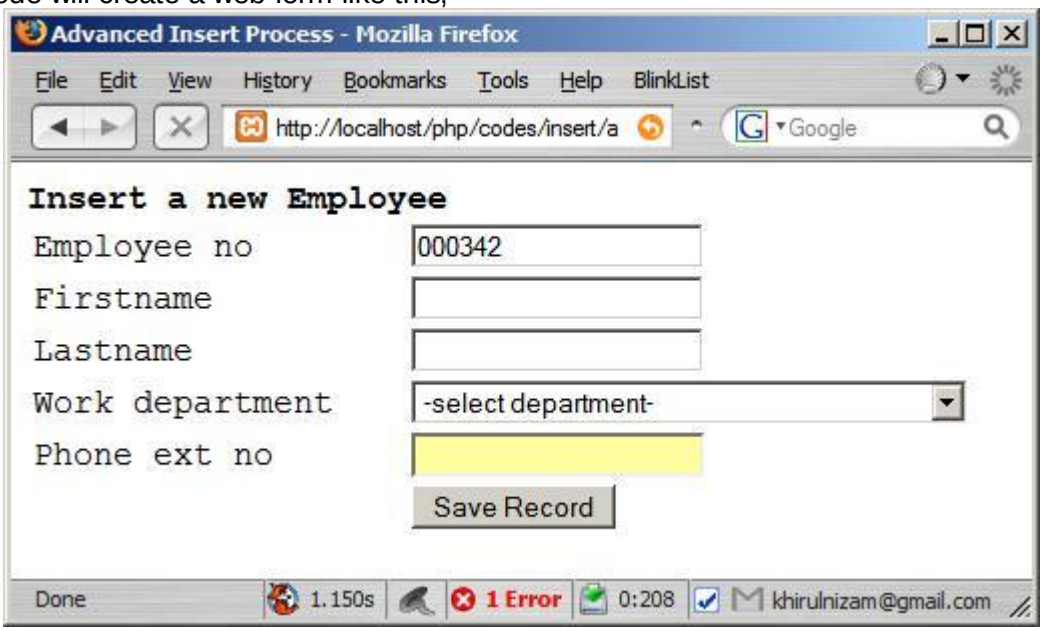
Advance Insert Application Exercise

This exercise is to facilitate users with more functions in the form interface. The objective of this exercise is to avoid user from entering false data such as wrong employee number, or entering a department code that doesn't exist in the *department* table.

```
<html>
<head>
<title>Advanced Insert Process</title>
</head>
<body>
<strong>Insert a new Employee </strong><br>
<form name="insert" action="insertrecord.php" method="get">

<table border="0">
  <tr>
  <?php
     //this script will determine the EMPNO automatically
     include "connection.php";
     $sql="select EMPNO from employee order by EMPNO asc";
     $rs=mysqli_query($db,$sql);
     $numrecords=mysqli_num_rows($rs);
     if($numrecords==0){
          //if no record exist
          $EMPNO="000001";
     }
     else{//thera are records already
          while($record=mysqli_fetch_array($rs)){
               $EMPNO=$record['EMPNO'];
          }
          $EMPNO++;//increase one from the last EMPNO
          $EMPNO=sprintf("%06d", $EMPNO );
          //this to make sure the new EMPNO is zero padded infront,
          //to make sure consistencey from the previous EMPNO
     }
  ?>
    <td width="40%">Employee no</td>
    <td width="60%"><input name="EMPNO" type="text" readonly
                                value="<? echo $EMPNO?>"></td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text"></td>
  </tr>
  <tr>


    <td>Work department</td>
    <td>
      <?php
          //this script is to generate option list
```

```
            //of department code and name
            $sql="select DEPTNO, DEPTNAME from department";
            $rs=mysqli_query($db,$sql);
      ?>
      <select name="WORKDEPT">
            <option value=''>-select department-</option>
            <?php
              while ($record=mysqli_fetch_array($rs)){
                  $deptno=$record['DEPTNO'];
                  $deptname=$record['DEPTNAME'];
                  echo "<option value='$deptno'>
                              $deptno - $deptname
                        </option>\n";
                  }
            ?>
      </select>

      </td>
  </tr>
  <tr>
    <td>Phone ext no</td>
    <td><input name="PHONENO" type="text" maxlength="4"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="save" type="submit" value="Save Record"></td>
  </tr>
</table>
</form>
</body>
</html>
```

The code will create a web form like this;



DELETE an Existing record

General format of SQL delete command:

```
DELETE FROM table_name
        WHERE conditions
```

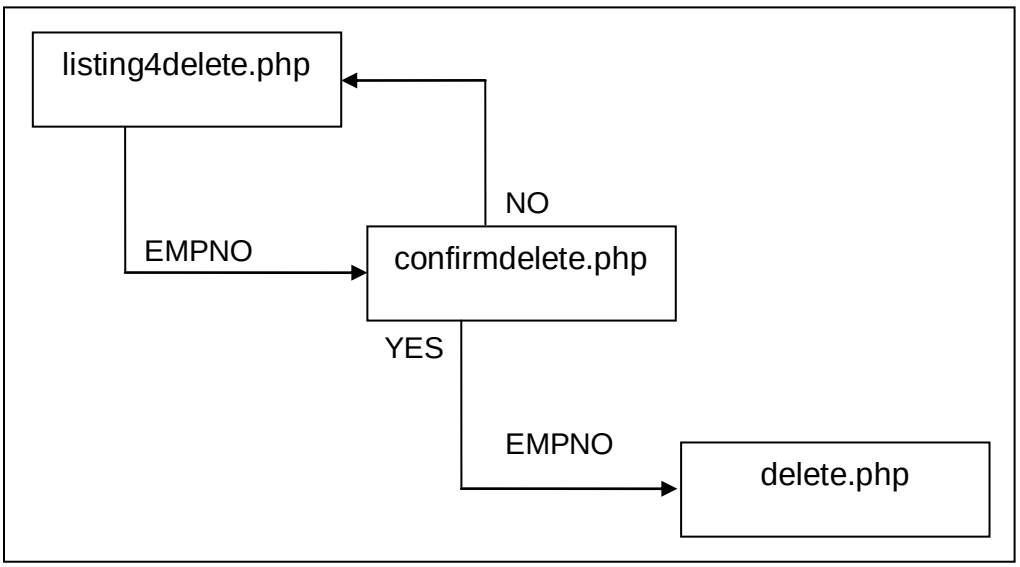
Example:

```
DELETE FROM employee
      WHERE EMPNO=='000345'
```

Deleting an Existing Record

The term DELETE in SQL is a command to delete a complete record from a specific table.

In this *delete* exercise we will create three different pages that will communicate to each other sequentially. Refer to the diagram below.



The first page is to list records with a delete button so the user could select which record to delete. The next page in the sequence will confirm the record deletion. This is to make sure the user is aware of the record they are going to delete. And the last page is the page where the SQL command for delete is executed and the selected record is really wiped out from the database.

Filename: ***listing4delete.php***

```
<?php
//include the connection details
include ("connection.php");

//Create SQL query
$query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
           from employee
           order by EMPNO asc";

//Execute the query
$qr=mysqli_query($db,$query);
if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
}
?>
<html>
<head>
<title>Listing to Edit</title>
</head>

<body>
Listing a Table for Editing<br>
<?php
//Check the record effected, if no records,
//display a message
if(mysqli_num_rows($qr)==0){
     echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
?>
<table width="90%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>


<?php
     while ($rekod=mysqli_fetch_array($qr)){
?>
 <tr>
    <td><?=$rekod['EMPNO']?></td>
    <td><?=$rekod['FIRSTNAME']?></td>
    <td><?=$rekod['LASTNAME']?></td>
    <td><?=$rekod['WORKDEPT']?></td>
    <td><?=$rekod['PHONENO']?></td>
    <td>
     <a href="confirmdelete.php?EMPNO=<?=$rekod['EMPNO']?>">
          delete
     </a>
```

```
    </td>
  </tr>
<?php
     }//end of records
?>
</table>
<?php
}//end if there are records
?>
</body>
</html>
```

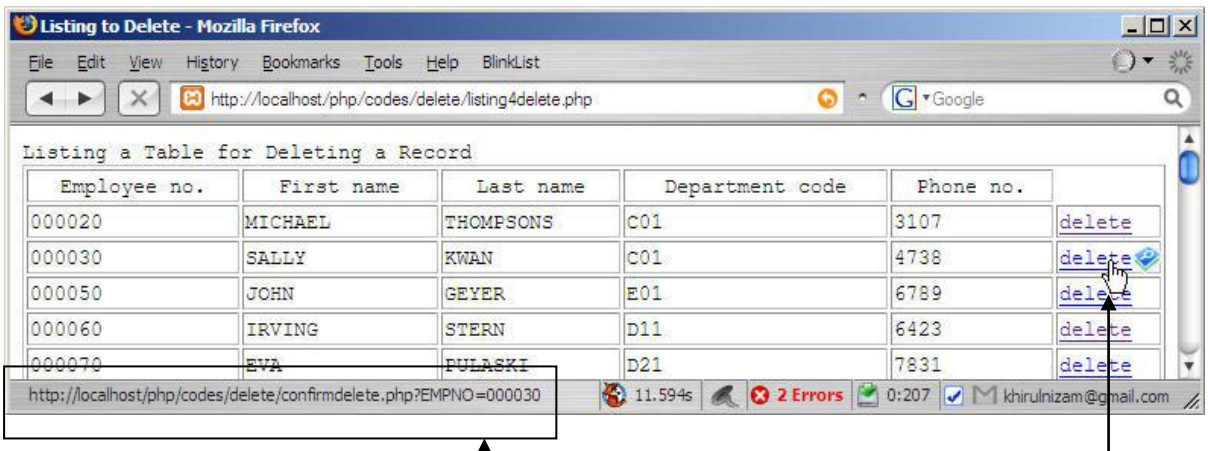
Observe closely the bolded codes in the *listing4delete.php*. The HTML tag will prepare a hyperlink that will direct user to another file named *confirmdelete.php* and bringing along a variable named EMPNO for the chosen employee record.

```
<a href="confirmdelete.php?EMPNO=<?=$rekod['EMPNO']?>">
   delete
</a>
```

the target file where user is asked a confirmation

the EMPNO variable with the value of the selected employee record to be deleted

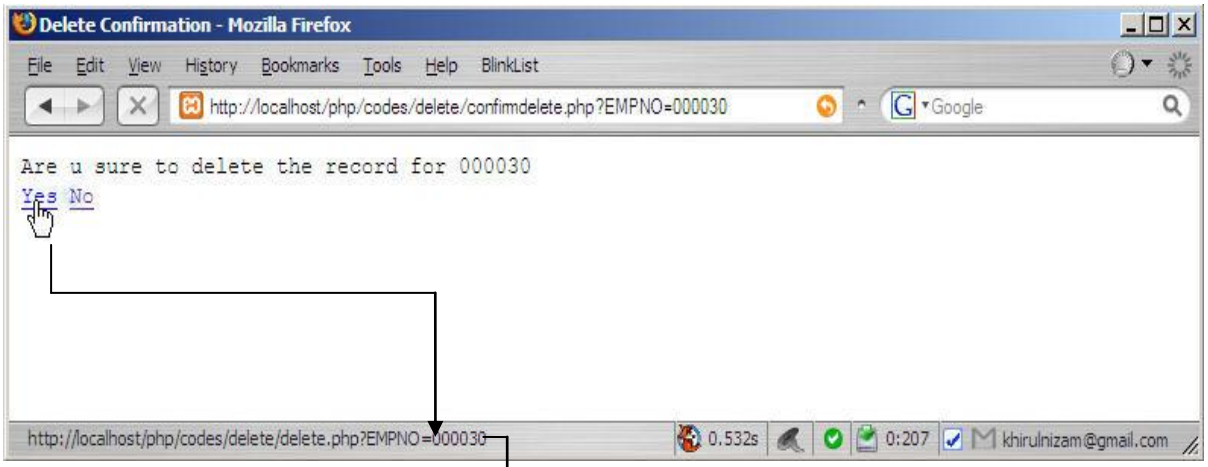
If you execute the script on the browser, it will generate a hyperlink, with a direction to another page named *confirmdelete.php*, by sending a variable EMPNO with the employee number of the selected employee record.



Each “delete” hyperlink will generate unique querystring based on the EMPNO for each record.

User choose one record to be deleted by clicking the “delete” hyperlink.

Once the user click the “delete” hyperlink, it will go the page *confirmdelete.php*, with the EMPNO to delete. When the EMPNO arrived at this page, it will be extracted and another hyperlink will be generated.



When the user click the “Yes” hyperlink, it will submit an employee number to another page named *delete.php*.

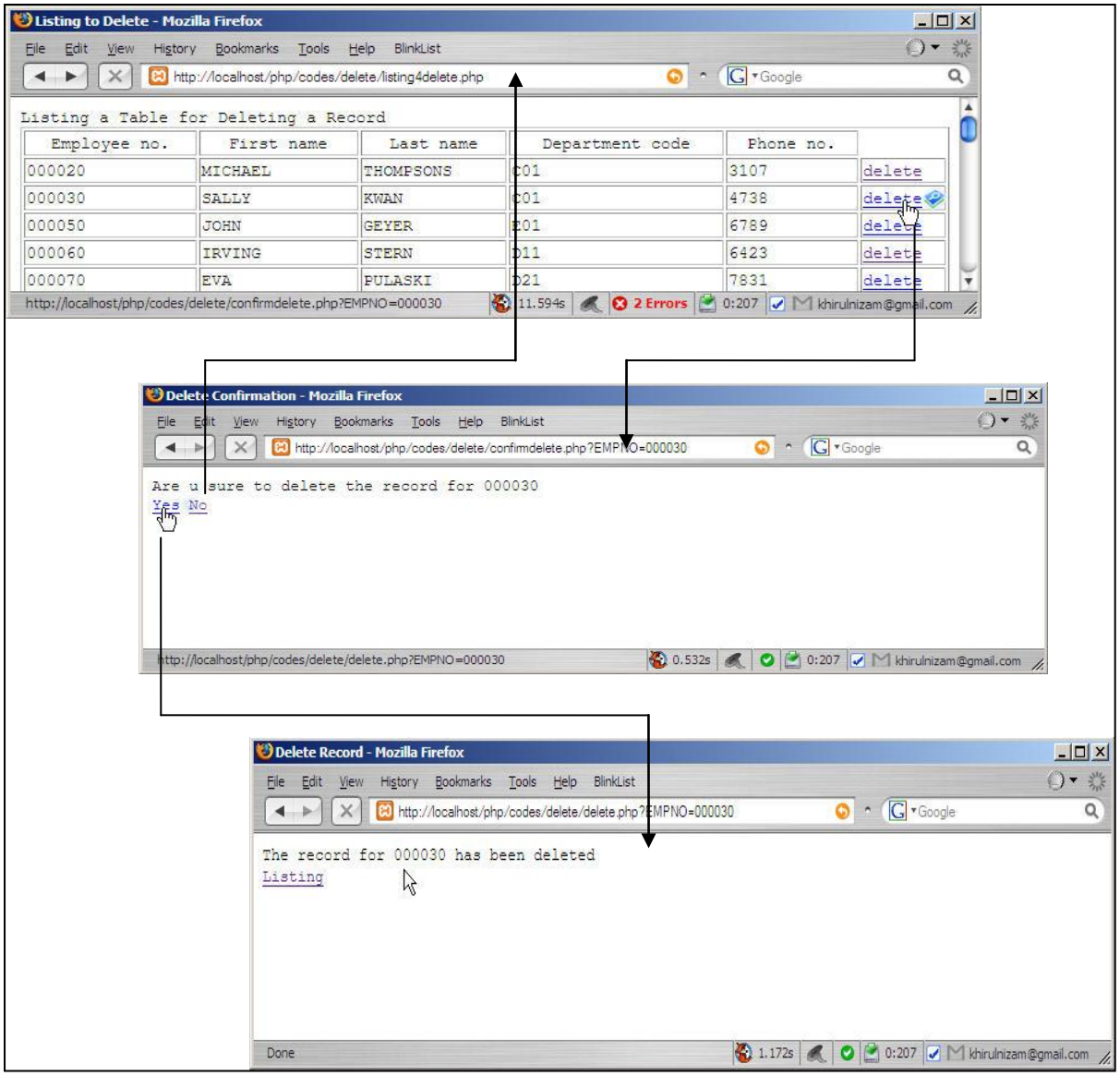
The first hyperlink is “Yes”, and the second one is “No”. If the user really want to delete the record, click “Yes”. This will direct the user to another file *delete.php* where the SQL *delete* command lies. However if the user would like to cancel the deletion process, click “No” and it will direct user back to the *listing4delete.php* file.

Filename: ***confirmdelete.php***.

```
<html>
<head>
<title>Delete Confirmation</title>
</head>
<body>
<?
      $EMPNO=$_GET['EMPNO'];
?>
Are u sure to delete the record for <? echo $EMPNO?>
<br>
<a href="delete.php?EMPNO=<? echo $EMPNO?>">Yes</a>
<a href="listing4delete.php">No</a>
<br>
</body>
</html>
```

The last step is to delete the selected record. This is the script to really delete the record. It contains in the file *delete.php*.

```
<html>
<head>
<title>Delete Record</title>
</head>
<body>
<?
      include "connection.php";
      $EMPNO=$_GET['EMPNO'];

      $sql="delete from employee
            where EMPNO='$EMPNO' ";
      $q=mysqli_query($db,$sql);

      if ($q==true){
            echo "The record for $EMPNO has been deleted";
      }
      else{
            echo "Fail to delete record for $EMPNO";
            echo mysqli_error($db);
      }
?>
<br>
<a href="listing4delete.php">Listing</a>
<br>
</body>
</html>
```


Listing to Delete - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/delete/listing4delete.php
Google
Listing a Table for Deleting a Record
Employee no. | First name | Last name | Department code | Phone no.
000020 | MICHAEL | THOMPSONS | C01 | 3107 | delete
000030 | SALLY | KWAN | C01 | 4738 | delete
000050 | JOHN | GEYER | E01 | 6789 | delete
000060 | IRVING | STERN | D11 | 6423 | delete
000070 | EVA | PULASKI | D21 | 7831 | delete
http://localhost/php/codes/delete/confirmdelete.php?EMPNO=000030
11.594s
2 Errors
0:207
khirulnizam@gmail.com
Delete Confirmation - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/delete/confirmdelete.php?EMPNO=000030
Google
Are u sure to delete the record for 000030
Yes No
http://localhost/php/codes/delete/delete.php?EMPNO=000030
0.532s
0:207
khirulnizam@gmail.com
Delete Record - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/delete/delete.php?EMPNO=000030
Google
The record for 000030 has been deleted
Listing
Done
1.172s
0:207
khirulnizam@gmail.com

UPDATE an existing record

General format of SQL update command:

```
UPDATE table_name
        SET field_name='new data'
        WHERE conditions
```

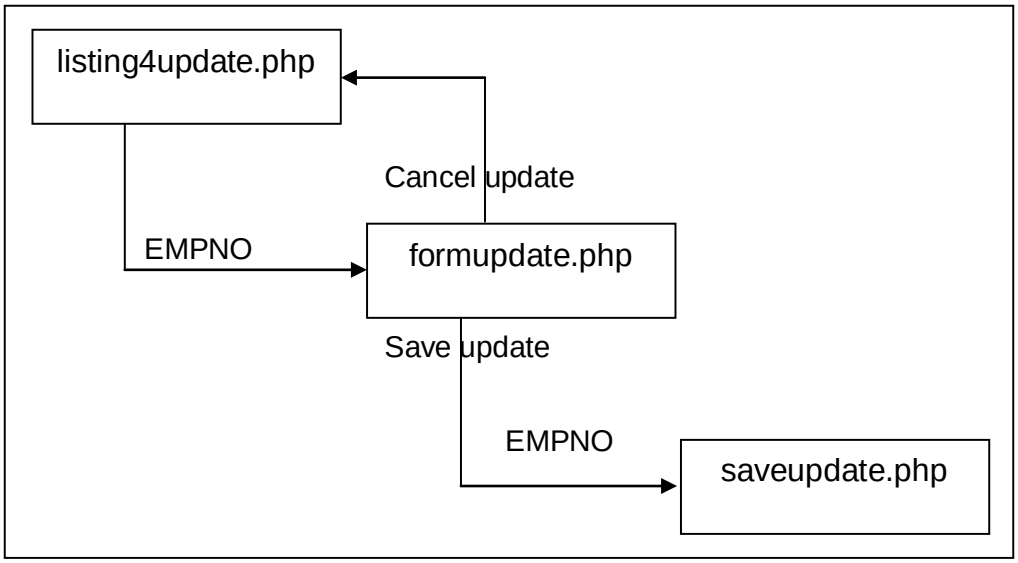
Example:

```
UPDATE employee
SET FIRSTNAME='Khirulnizam', LASTNAME='Abd Rahman', WORKDEPT='C01',
PHONENO='1110'
WHERE EMPNO='000345'
```

Simple Update Exercise

The term UPDATE in SQL is a command to modify an exiting record from a specific table.

In this *update* exercise we will create three different pages that will communicate to each other sequentially. Refer to the diagram below.



The first page is to list records with update button so the user could select which record to update. The next page in the sequence is the update form. The existing data is populated in the form. This is very useful for the user to see the existing data in order to make the changes. The new data is than sent to another page that will save the changes to the database.

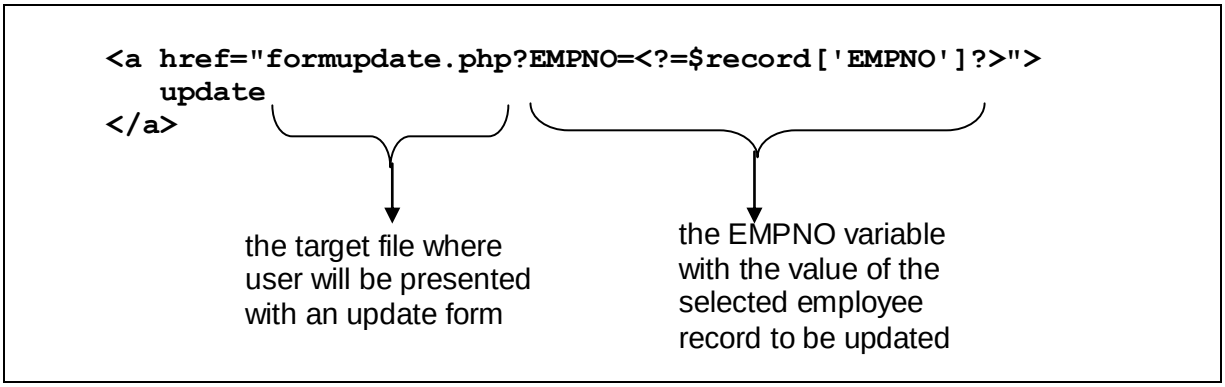
The first file is the *listing4update.php*. This is the same file as we used in the Chapter 12: Delete an Existing Record. The difference is just the last column that contains the “update” hyperlink, which connects to the page named *formupdate.php*, contains the form to change the existing data.

Filename: ***listing4update.php***

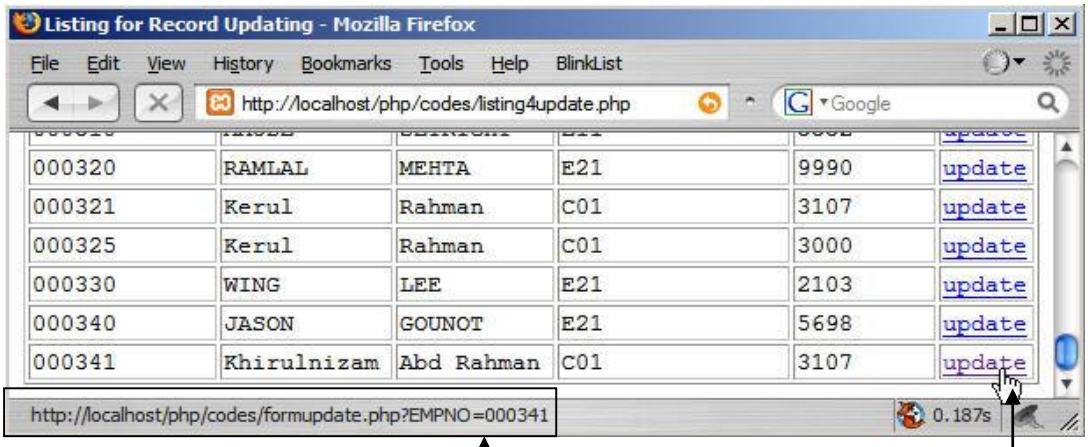
```
<?php
//include the connection details
include ("connection.php");

//Create SQL query
$query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
           from employee
           order by EMPNO asc";

//Execute the query
$qr=mysqli_query($db,$query);
if($qr==false){
     echo ("Query cannot be executed!<br>");
     echo ("SQL Error : ".mysqli_error($db));
}
?>
<html>
<head>
<title>Listing for Record Updating</title>
</head>

<body>
Listing for Record Updating<br>
<?php
//Check the record effected, if no records,
//display a message
if(mysqli_num_rows($qr)==0){
     echo ("No record fetched...<br>");
}//end no record
else{//there is/are record(s)
?>
<table width="100%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>


<?php
     while ($record=mysqli_fetch_array($qr)){
?>
 <tr>
    <td><?=$record['EMPNO']?></td>
    <td><?=$record['FIRSTNAME']?></td>
    <td><?=$record['LASTNAME']?></td>
    <td><?=$record['WORKDEPT']?></td>
```

```
    <td><?=$record['PHONENO']?></td>
    <td>
      <a href="formupdate.php?EMPNO=<?=$record['EMPNO']?>">
          update
      </a>
    </td>

  </tr>
<?php
     }//end of records
?>
</table>
<?php
}//end if there are records
?>
</body>
</html>
```

Observe closely the bolded codes in the *listing4update.php*. The HTML tag will prepare a hyperlink that will direct user to another file named *formupdate.php* and bringing along a variable named EMPNO for the chosen employee record. This process is similar to the delete procedure.



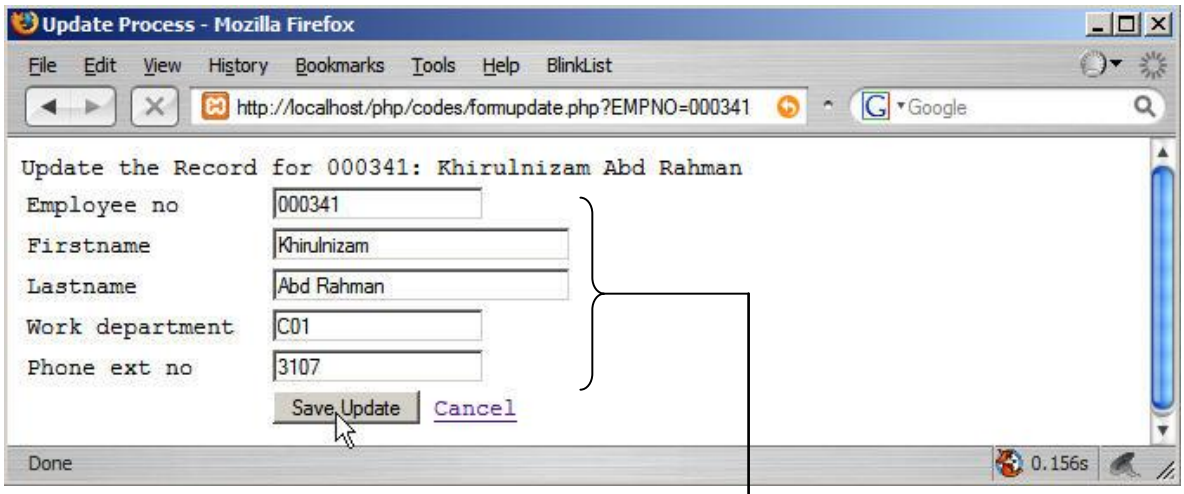
If you execute the script in the browser, it will generate a hyperlink, with a direction to another page named *formupdate.php*, by sending a variable EMPNO with the employee number of the selected employee record.



Each “delete” hyperlink will generate unique querystring based on the EMPNO for each record.

User choose one record to be deleted by clicking the “delete” hyperlink.

Once the user click the “delete” hyperlink, it will go the page *confirmdelete.php*, with the EMPNO to delete. When the EMPNO arrived at this page, it will be extracted and another hyperlink will be generated.



The existing data for the record appeared in the textboxes, to provide convenience for the user to change.

The first hyperlink is “Yes”, and the second one is “No”. If the user really want to delete the record, click “Yes”. This will direct the user to another file *delete.php* where the SQL *delete* command lies. However if the user would like to cancel the deletion process, click “No” and it will direct user back to the *listing4update.php* file.

Filename: ***formupdate.php***.

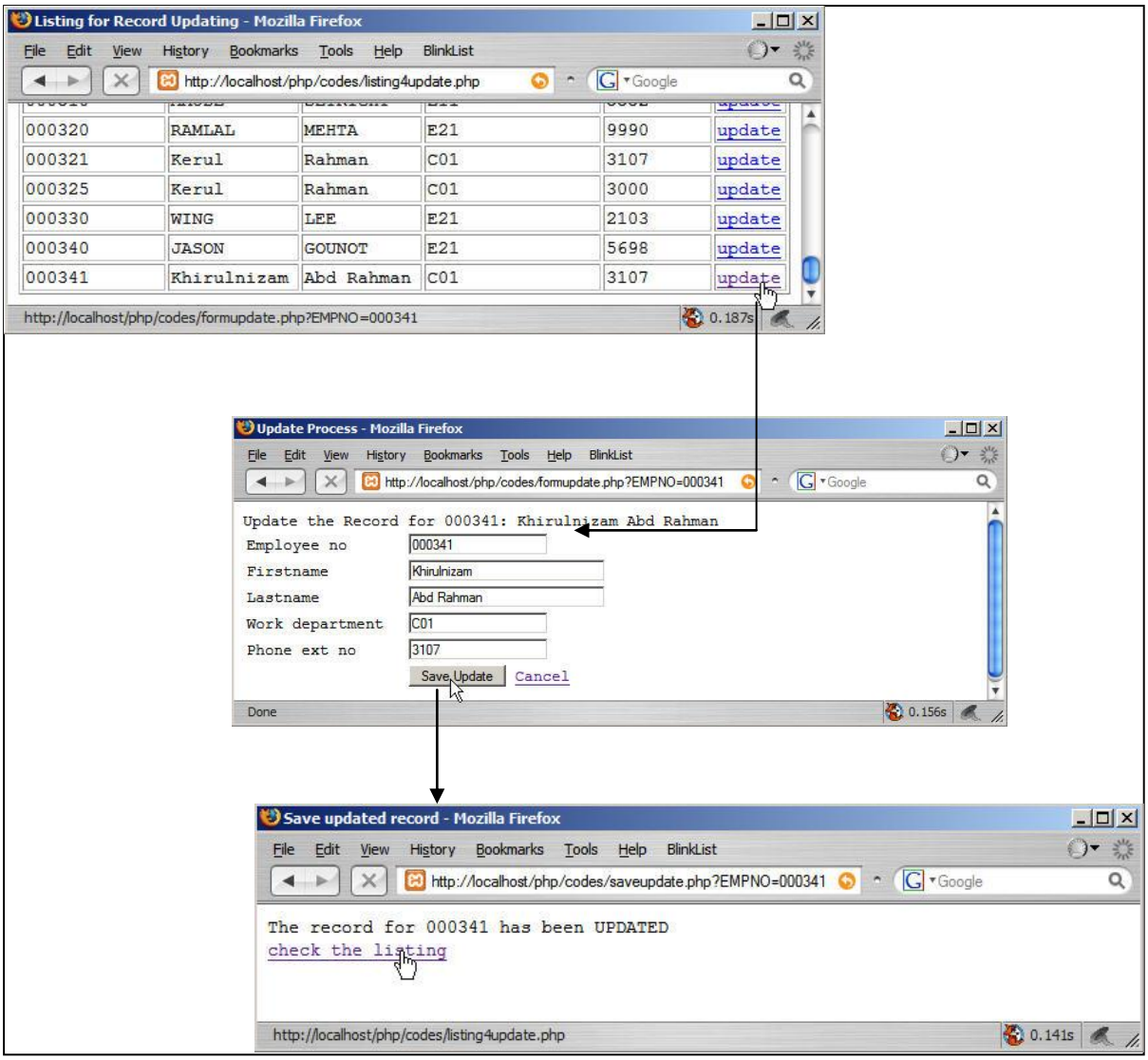
```
<html>
<head>
<title>Update Process</title>
</head>
<body>
<?php
include "connection.php";
$EMPNO=$_GET['EMPNO'];
$sql="select * from employee
          where EMPNO='$EMPNO'";

$rs=mysqli_query($db,$sql);
$record=mysqli_fetch_array($rs);
?>
Update the Record for <?=$record['EMPNO']?>:
<?=$record['FIRSTNAME']?> <?=$record['LASTNAME']?> <br>

<form name="insert" action="saveupdate.php" method="get">
  <table width="100%"  border="0">
  <tr>
    <td width="30%">Employee no</td>
    <td width="70%">
        <input name="EMPNO" type="text" readonly
          value="<? echo $record['EMPNO']?>" >
     </td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text" size="30"
          value="<?=$record['FIRSTNAME']?>" >
     </td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text" size="30"
          value="<?=$record['LASTNAME']?>" >
     </td>
  </tr>
  <tr>
    <td>Work department </td>
    <td><input name="WORKDEPT" type="text"
          value="<?=$record['WORKDEPT']?>" >
     </td>
  </tr>
  <tr>
    <td>Phone ext no</td>
    <td><input name="PHONENO" type="text"
          value="<?=$record['PHONENO']?>" maxlength="4" >
     </td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="save" type="submit" value="Save Update">
      <a href="listing4update.php">Cancel</a> </td>
```

```
  </tr>
</table>
</form>

</body>
</html>
```

The last step is to extract all the data from the form and change the old data of the record to the new one.

Filename: ***update.php***

```
<?php
$EMPNO=$_GET['EMPNO'];
$FIRSTNAME=$_GET['FIRSTNAME'];
$LASTNAME=$_GET['LASTNAME'];
$WORKDEPT=$_GET['WORKDEPT'];
$PHONENO=$_GET['PHONENO'];

$sql="update employee set FIRSTNAME='$FIRSTNAME',
           LASTNAME='$LASTNAME',
           WORKDEPT='$WORKDEPT',
           PHONENO='$PHONENO'
           where EMPNO='$EMPNO'";

include "connection.php";
$rs=mysqli_query($db, $sql);
?>

<html>
<head>
<title>Save updated record</title>
</head>
<body>
<?php
if($rs==true){
     echo "The record for $EMPNO has been UPDATED<br>";
     echo "<a href='listing4update.php'>check the listing</a><br>";
}
else{
     echo "The update is NOT SUCCESSFUL!<br>";
}
?>
</body>
</html>
```


Listing for Record Updating - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/listing4update.php
000320 RAMLAL MEHTA E21 9990 update
000321 Kerul Rahman C01 3107 update
000325 Kerul Rahman C01 3000 update
000330 WING LEE E21 2103 update
000340 JASON GOUNOT E21 5698 update
000341 Khirulnizam Abd Rahman C01 3107 update
http://localhost/php/codes/formupdate.php?EMPNO=000341
0.187s
Update Process - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/formupdate.php?EMPNO=000341
Update the Record for 000341: Khirulnizam Abd Rahman
Employee no 000341
Firstname Khirulnizam
Lastname Abd Rahman
Work department C01
Phone ext no 3107
Save Update Cancel
Done
0.156s
Save updated record - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/php/codes/saveupdate.php?EMPNO=000341
The record for 000341 has been UPDATED
check the listing
http://localhost/php/codes/listing4update.php
0.141s

Advanced Update an Existing Record Exercise

The first file is almost the same as the file to do the record listing. However there is a form to search for a record with three search options; by EMPNO, FIRSTNAME or LASTNAME. The found records will be provide an update hyperlink which points to another file named *xformupdate.php*. We name this file as *xlisting4update.php* to differentiate from the simple update in the previous discussion.

Filename: ***xlisting4update.php***

```
<html>
<head>
<title>Listing for Record Updating</title>
</head>

<body>
Search for Record to be Updated<br>
<form action="" method="get" name="searchupdate">
  Please enter only ONE of the criteria below<br>

  <table width="90%"  border="0">
    <tr>
      <td>EMPNO</td>
      <td><input name="EMPNO" type="text" maxlength="6"></td>
    </tr>
    <tr>
      <td>FIRSTNAME</td>
      <td><input name="FIRSTNAME" type="text"></td>
    </tr>
    <tr>
      <td>LASTNAME</td>
      <td><input name="LASTNAME" type="text"></td>
    </tr>
     <tr>
      <td> </td>
      <td><input name="btnsearch" type="submit" value="SEARCH"></td>
    </tr>
  </table>
</form>
<br>
<?php
include ("connection.php");
$EMPNO=$_GET['EMPNO'];
$FIRSTNAME=$_GET['FIRSTNAME'];
$LASTNAME=$_GET['LASTNAME'];
if (!($EMPNO==NULL && $FIRSTNAME==NULL && $LASTNAME==NULL)){
      //check for any user search
  echo "<strong>Listing for Record Updating</strong><br>";

  //build search condition
  if ($EMPNO!=NULL){
        $condition="where EMPNO LIKE '%$EMPNO%'";
  }
  else if ($FIRSTNAME!=NULL){
        $condition="where FIRSTNAME LIKE '%$FIRSTNAME%'";
  }
```

1

2

```
  else if ($LASTNAME!=NULL){
        $condition="where LASTNAME LIKE '%$LASTNAME%'";
  }
   //Create SQL query
  $query="select EMPNO, FIRSTNAME, LASTNAME, WORKDEPT, PHONENO
          from employee
          $condition
          order by EMPNO asc";

  //Execute the query
  $qr=mysqli_query($db,$query);
  if($qr==false){
        echo ("Query cannot be executed!<br>");
        echo ("SQL Error : ".mysqli_error($db));
  }

  /*Check the record effected, if no records,
  display a message*/
  if(mysqli_num_rows($qr)==0){
        echo ("No record found...<br>");
  }//end no record
  else{//there is/are record(s)
  ?>

 <table width="100%"  border="1">
  <tr align="center">
    <td>Employee no.</td>
    <td>First name</td>
    <td>Last name</td>
    <td>Department code</td>
    <td>Phone no.</td>
  </tr>

  <?php
        while   ($record=mysqli_fetch_array($qr)){//redo   to   other
records
  ?>
      <tr>
    <td><?=$record['EMPNO']?></td>
    <td><?=$record['FIRSTNAME']?></td>
    <td><?=$record['LASTNAME']?></td>
    <td><?=$record['WORKDEPT']?></td>
    <td><?=$record['PHONENO']?></td>
      <td>
      <a href="xformupdate.php?EMPNO=<?=$record['EMPNO']?>">
      update</a>
      </td>
  </tr>
  <?php
      }//end of records
  ?>
</table>
<?php
  }//end if there are records
```

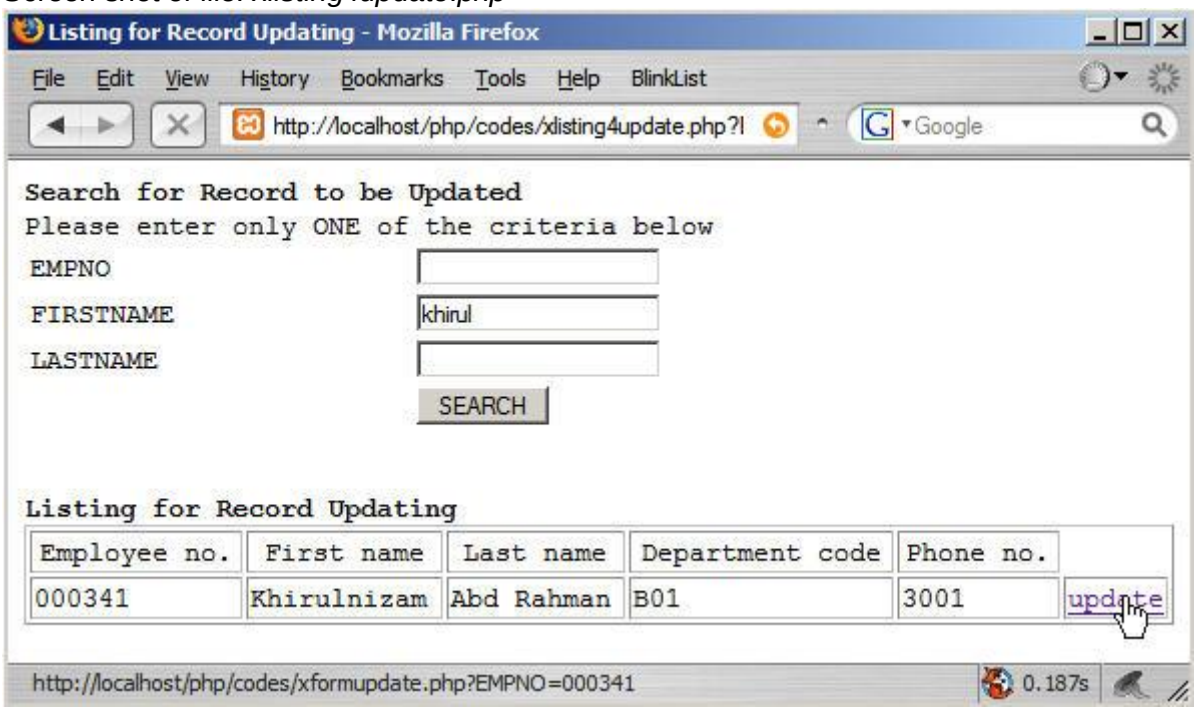
3

```
}//end if any user input
?>
</body>
</html>
```

Description for bolded and gray-shadowed text in *xlisting4update.php* codes.

1. Search and listing will only be done if one of the field for EMPNO, FIRSTNAME or LASTNAME is entered by the user.
2. This script will generate the condition of the SQL select statement to search for the record the user request, using the criteria chosen.
3. This will generate the hyperlink to direct user to another page named xformupdate.php, and bringing along the EMPNO.

Screen shot of file: *xlisting4update.php*



This example will consider the data validating and filtering. The important part in this exercise is to provide user with a more user-friendly form. For instance, the work department must be listed using drop down list. This is to prevent the user from typing the wrong department code. The phone number must be all digits, without any special character or alphabet.

This is the improved version of formupdate.php with the name *xformupdate.php*.
Filename: ***xformupdate.php***

```
<html>
<head>
<title>Update Process</title>
</head>
<body>
<?php
include "connection.php";
$EMPNO=$_GET['EMPNO'];
$sql="select * from employee
           where EMPNO='$EMPNO'";

$rs=mysqli_query($db,$sql);
$record=mysqli_fetch_array($rs);
?>
Update the Record for <?=$record['EMPNO']?>:
<?=$record['FIRSTNAME']?> <?=$record['LASTNAME']?> <br>

<form name="insert" action="xsaveupdate.php" method="get">
  <table width="100%"  border="0">
  <tr>
    <td width="30%">Employee no</td>
    <td width="70%"><input name="EMPNO" type="text" readonly
                      value="<? echo $record['EMPNO']?>" ></td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text" size="30"
                      value="<?=$record['FIRSTNAME']?>" ></td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text" size="30"
                      value="<?=$record['LASTNAME']?>" ></td>
  </tr>
  <tr>
    <td>Work department </td>
    <td>
      <?php
      /*this srcipt generates dropdown list
        with the selected record WORKDEPT as the deault value*/
      $employeeWORKDEPT=$record['WORKDEPT'];
      $sqldept="select DEPTNO, DEPTNAME from department";
      $rsdept=mysqli_query($db,$sqldept);
      ?>
      <select name="WORKDEPT">
         <?php
           while ($dept=mysqli_fetch_array($rsdept)){
                $deptno=$dept['DEPTNO'];
                $deptname=$dept['DEPTNAME'];
                /*if the DEPTNO equals to the employee's WORKDEPT,
                option list is default*/
                if ($employeeWORKDEPT==$deptno){
                     echo "<option value='$deptno' selected>
```

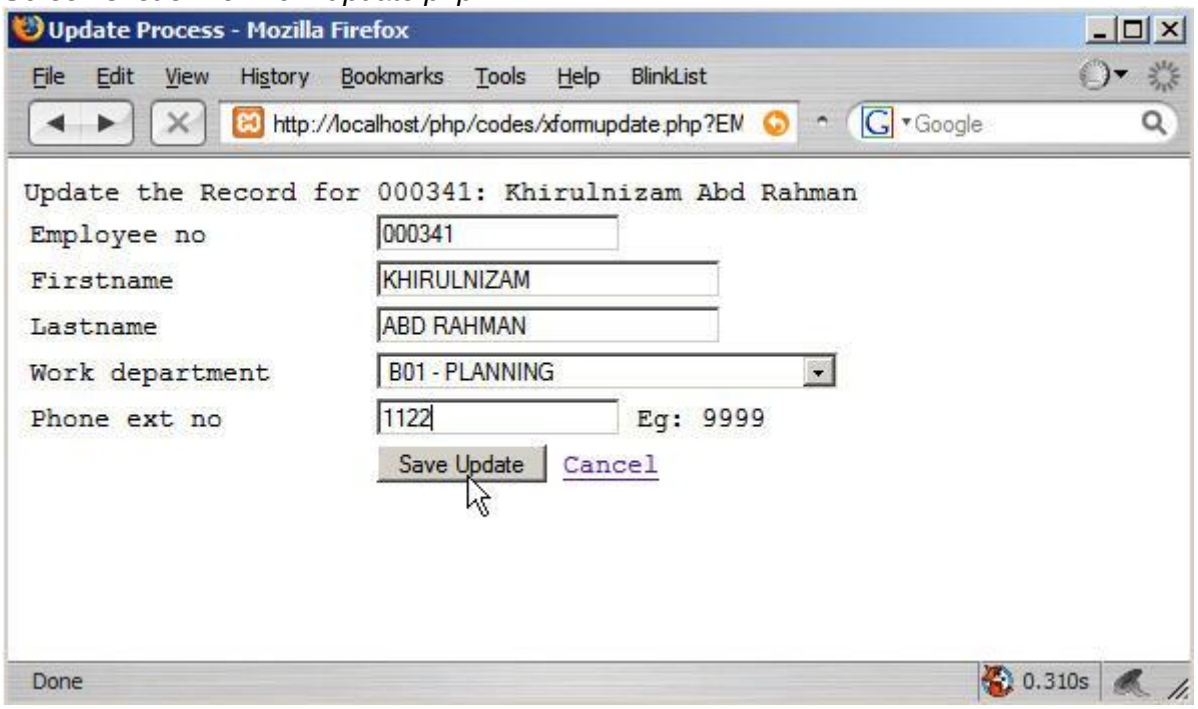
1

```
                              $deptno - $deptname </option>\n";
                    }else{
                          echo "<option value='$deptno'> $deptno -
                              $deptname </option>\n";
                    }
               }//end while
               ?>
          </select>
          </td>
     </tr>
     <tr>
       <td>Phone ext no</td>
       <td><input name="PHONENO" type="text"
              value="<?=$record['PHONENO']?>" maxlength="4" >
              Eg: 9999</td>
     </tr>
     <tr>
       <td> </td>
       <td><input name="save" type="submit" value="Save Update">
         <a href="listing4update.php">Cancel</a> </td>
     </tr>
</table>
</form>
</body>
</html>
```

Description for bolded and gray-shadowed text in *xformupdate.php* codes.

1. The script generates a option list (drop down combo box). The default selected item will be based on the WORKDEPT in the employee's record.

Screen shot of file: *xformupdate.php*

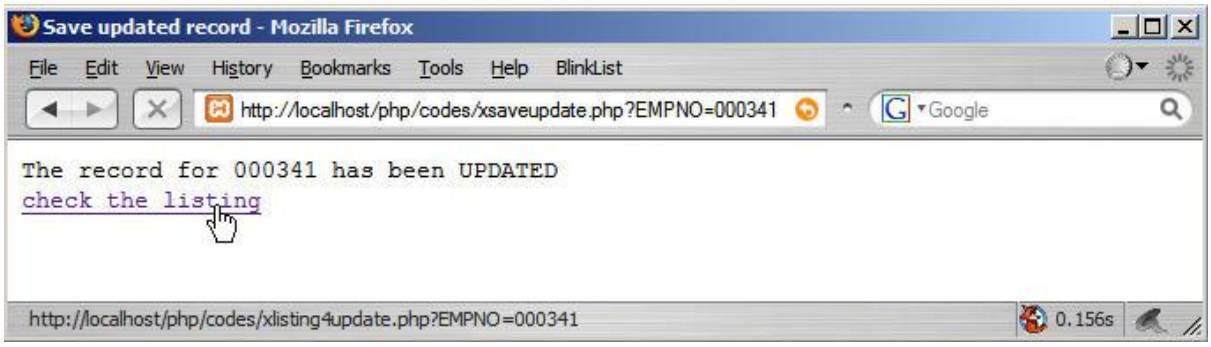


And the last process is to extract all the data, do some data filtering. If there's nothing wrong with the data save the changes. However if the data is invalid, go back to the form.

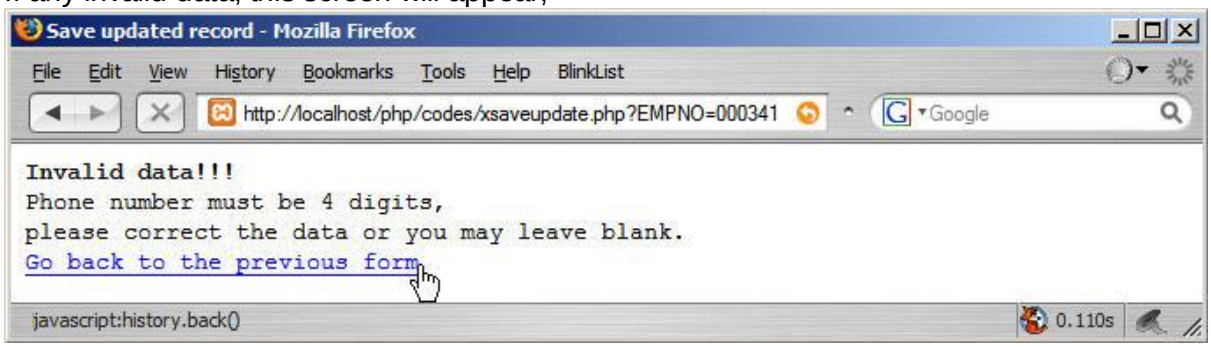
Filename: ***xsaveupdate.php***

```
<?php
include "connection.php";
$EMPNO=$_GET['EMPNO'];
$FIRSTNAME=$_GET['FIRSTNAME'];
$LASTNAME=$_GET['LASTNAME'];
$WORKDEPT=$_GET['WORKDEPT'];
$PHONENO=$_GET['PHONENO'];

//check for input error (filtering)
$dataerror=false;
if ($PHONENO!=NULL &&(!ctype_digit($PHONENO)||strlen($PHONENO)!=4)){
      /* phone number not blank and (not digit or length not 4)*/
      $dataerror=true;
}
?>

<html>
<head>
<title>Save updated record</title>
</head>
<body>
<?php
if ($dataerror==false){
      $sql="update employee set FIRSTNAME='$FIRSTNAME',
            LASTNAME='$LASTNAME',
            WORKDEPT='$WORKDEPT',
            PHONENO='$PHONENO'
            where EMPNO='$EMPNO'";
      $rs=mysqli_query($db, $sql);
      if($rs==true){
            echo "The record for $EMPNO has been UPDATED<br>";
            echo "<a href='xlisting4update.php?EMPNO=$EMPNO'>
                  check the listing</a><br>";
      }
      else{
            echo "The update is NOT SUCCESSFUL!<br>";
      }
}else{
      echo "<strong>Invalid data!!!<br></strong>";
      echo "Phone number must be 4 digits,<br>";
      echo "please correct the data or you may leave blank. <br>";
      echo "<a href='javascript:' onclick='history.back()'>
              Go back to the previous form </a><br>";
      /*this provides a hyperlink similar to Back button in
      the browser*/
}
?>
</body>
</html>
```

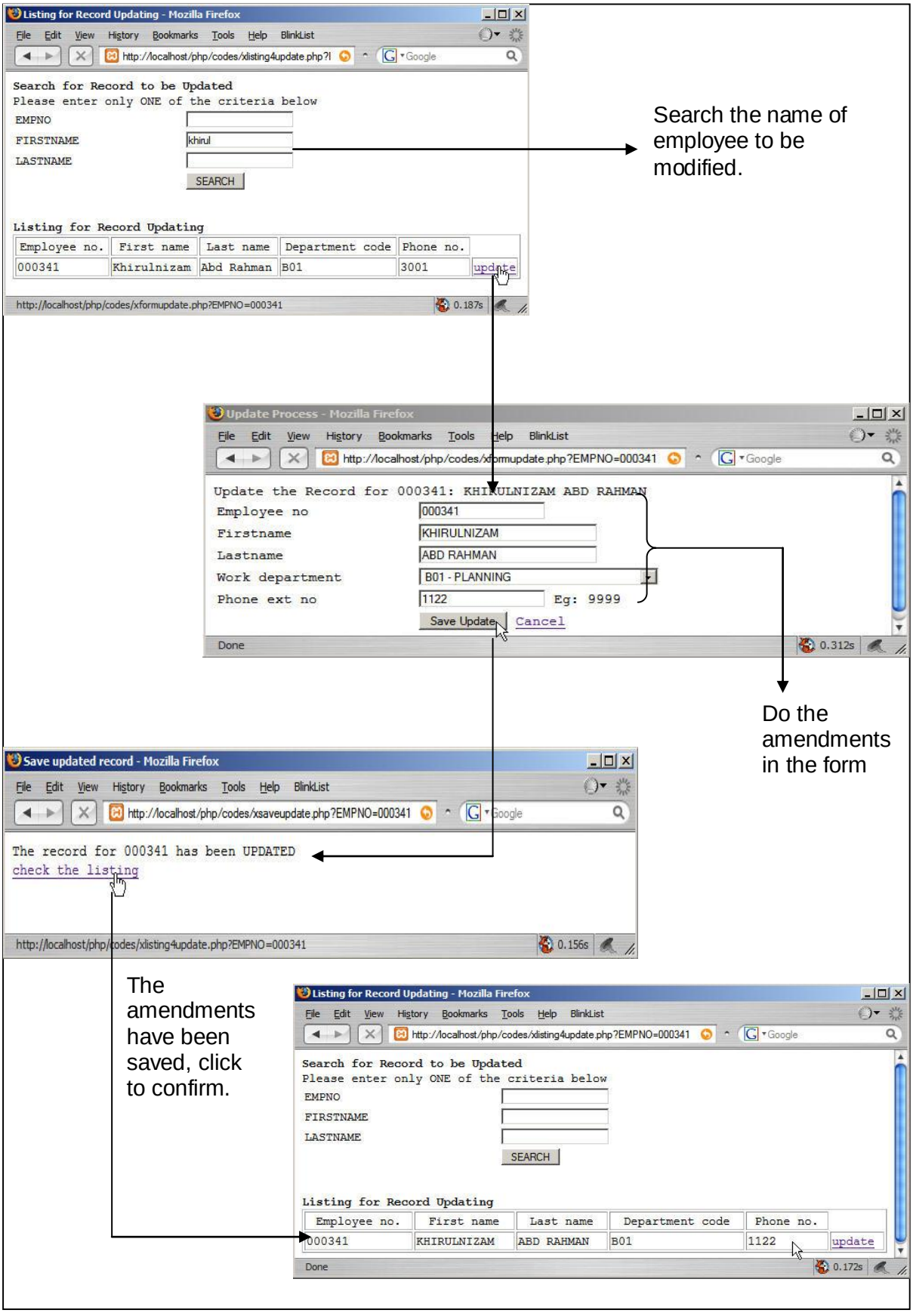
Screen shot of file: *xsaveupdate.php*



If any invalid data, this screen will appear;



Screen shots for the whole process;



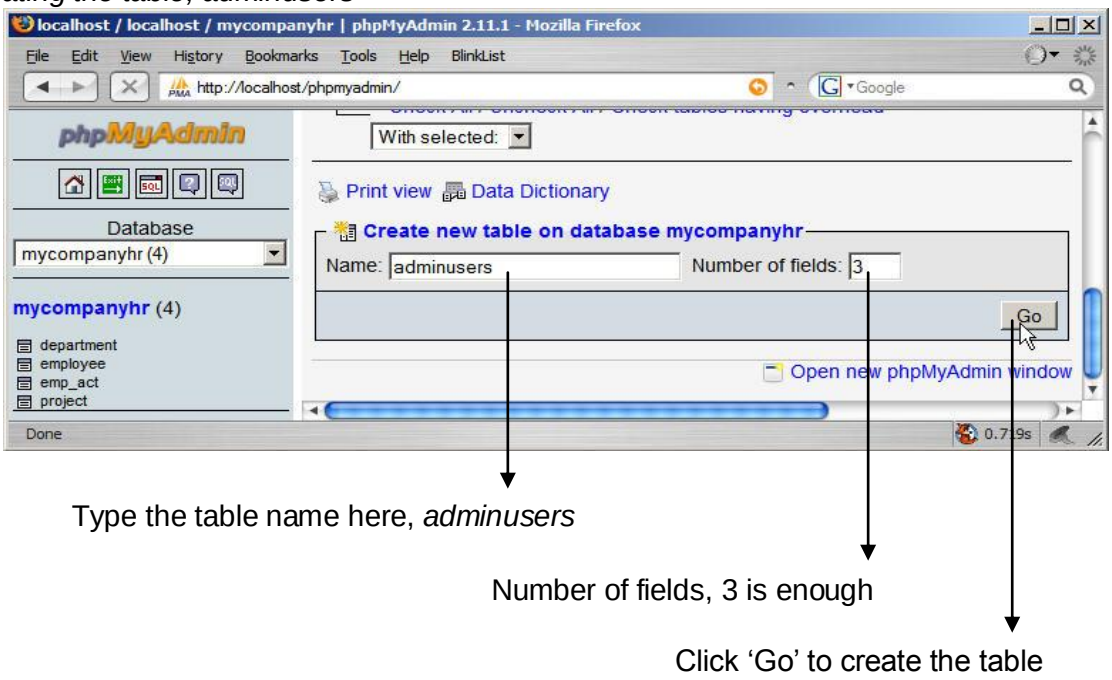
Logging in using Username and Password

Creating the *adminusers* table

For this exercise, we will be using the EMPNO as the username and a password given to the username. The information on the user's password and authorization level will be stored in another table named *adminusers* in the same *mycompanyhr* database.

So now lets create the *adminusers* table in the *mycompanyhr* table. Go to the phpmyadmin in your favorite browser and choose the database *mycompanyhr*.

Creating the table, adminusers

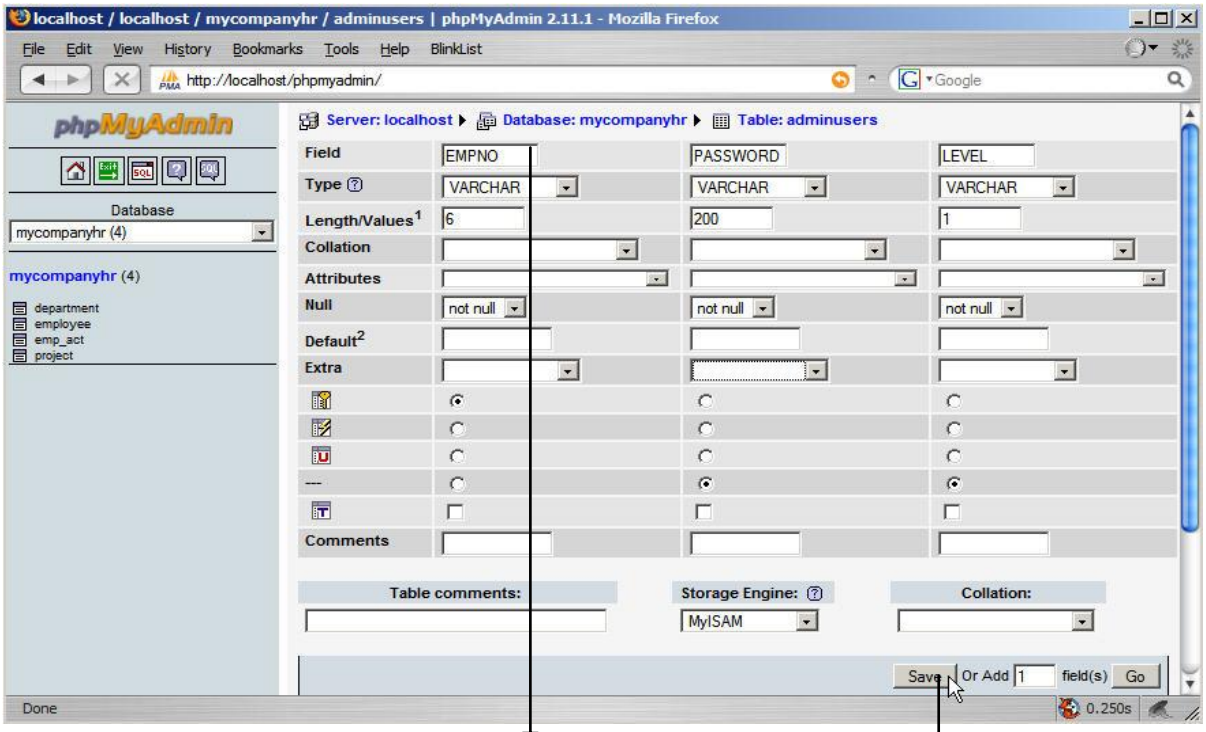


Type the table name here, *adminusers*

Number of fields, 3 is enough

Click 'Go' to create the table

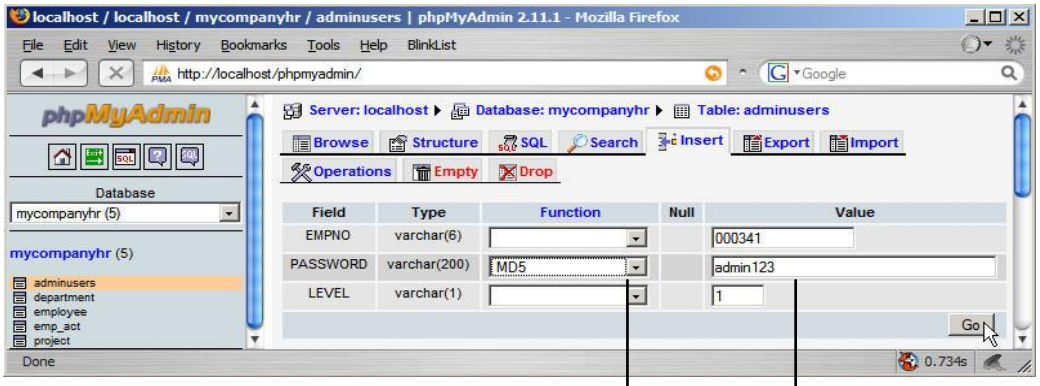
Setting the fields specifications, for table adminusers



Use the **EMPNO** to maintain the employer's id. This way, only the employee in listed in the employee table has the right to become an administrator.

Save the fields' specifications.

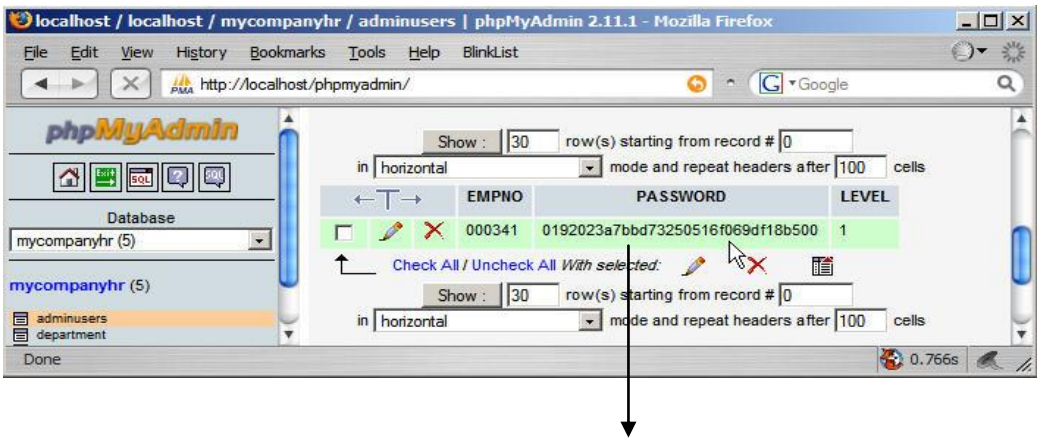
Insert an admin user with MD5 password



Set the additional MD5 function to be performed to the password

Do remember the password

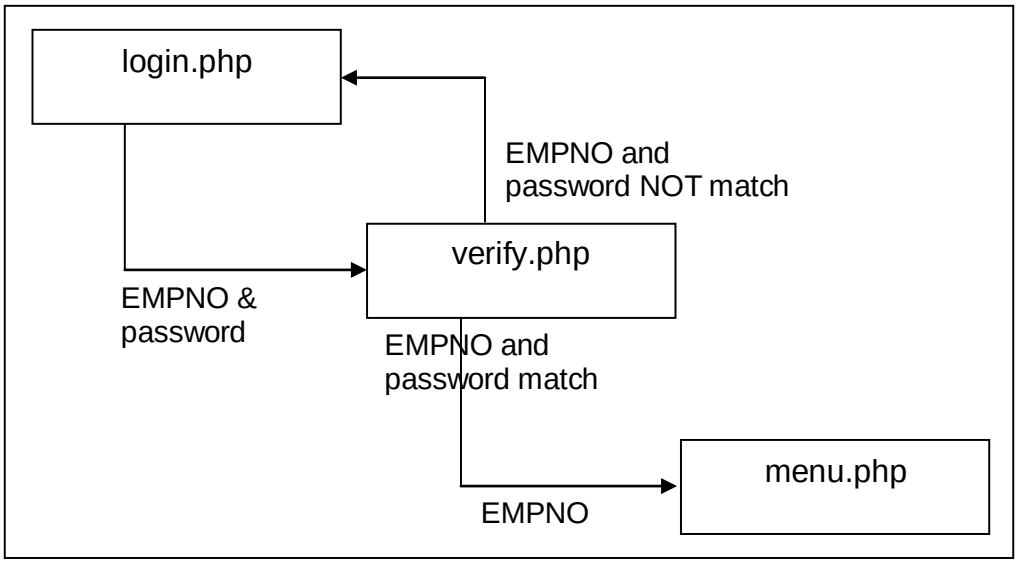
The admin user with MD5 password



The MD5 password (the actual data is admin123)

Simple Login page

This login facility contains a login form, a verification script and a simple system menu to access the other administration parts.

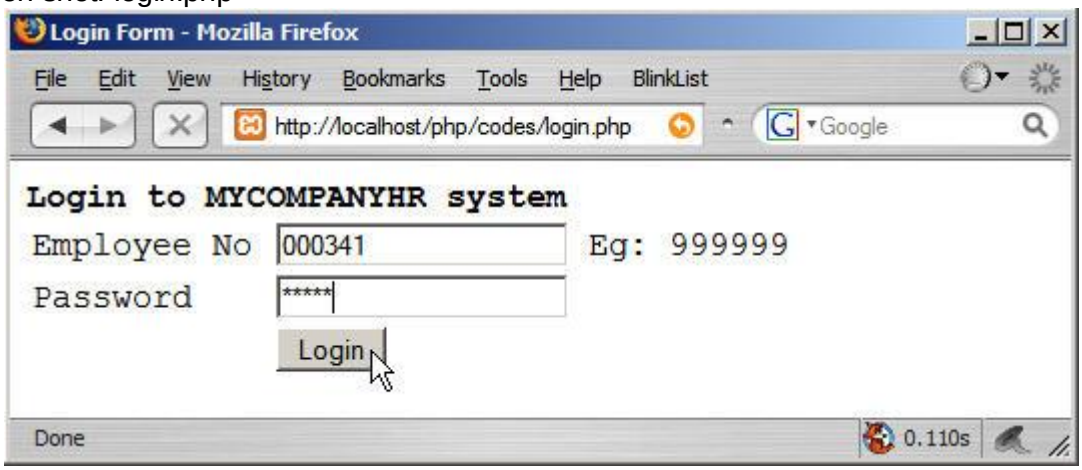


The first page is *login.php* which contains a form to receive user's username and password. The username and password are then sent to another page (*verify.php*) for verification. In this verification process the username and the password entered by the user will be compared to the username and the password stored in the table. The user will be directed to the administration menu (*menu.php*) if and only if the username and the password are match.

Filename: ***login.php***

```
<html>
<head>
<title>Login Form</title>
</head>
<body>

<strong>Login to MYCOMPANYHR system</strong>
<form name="formlogin" method="post" action="verify.php">
<table width="400"  border="0">
  <tr>
    <td>Employee No</td>
    <td><input name="EMPNO" type="text" maxlength="6">
      Eg: 999999 </td>
  </tr>
  <tr>
    <td>Password</td>
    <td><input name="PASSWORD" type="password"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="submit" type="submit" value="Login"></td>
  </tr>
</table>
 </form>

</body>
</html>
```

Screen shot: login.php



Verification process

The next page is verification page which consists of script to compare the entered user's password against the password stored in the database. If the username (in this case the EMPNO) and password match, this script will direct the user to the administration menu page. If not, the user will be asked to re-enter the username and the password.

Filename: ***verify.php***

```
<?php
$EMPNO=$_POST['EMPNO'];
$PASSWORD=$_POST['PASSWORD'];
include "connection.php";
$sql="select * from adminusers where EMPNO='$EMPNO'";
$rs=mysqli_query($db, $sql);
?>
<html>
<head>
<title>MyCOMPANYHR-verify</title>
</head>

<body>
<strong>Verify employee number and password</strong><br>
<?php
if(mysqli_num_rows($rs)==1){ //found one user
      $record=mysqli_fetch_array($rs);
      $DBPASSWORD=$record['PASSWORD'];//password from database
      $USERPASSWORD=md5($PASSWORD);//MD5 password key-in by user
      if($DBPASSWORD==$USERPASSWORD){
            //compare password from database against password entered by user
         echo "Username and password match,<br>";
         echo "WELCOME $EMPNO !!!<br>";
         echo "<a href='menu.php?EMPNO=$EMPNO'>
                  Click to ADMIN Menu </a><br>";
      }

      else{
         echo "Username found, but password NOT match,<br>";
         echo "<a href='javascript:history.back()'>
                  re-enter password</a><br>";
      }
}

else{
      echo "Username NOT found,<br>";
      echo "<a href='javascript:history.back()'>re-login</a><br>";
}
?>
</body>
</html>
```

1

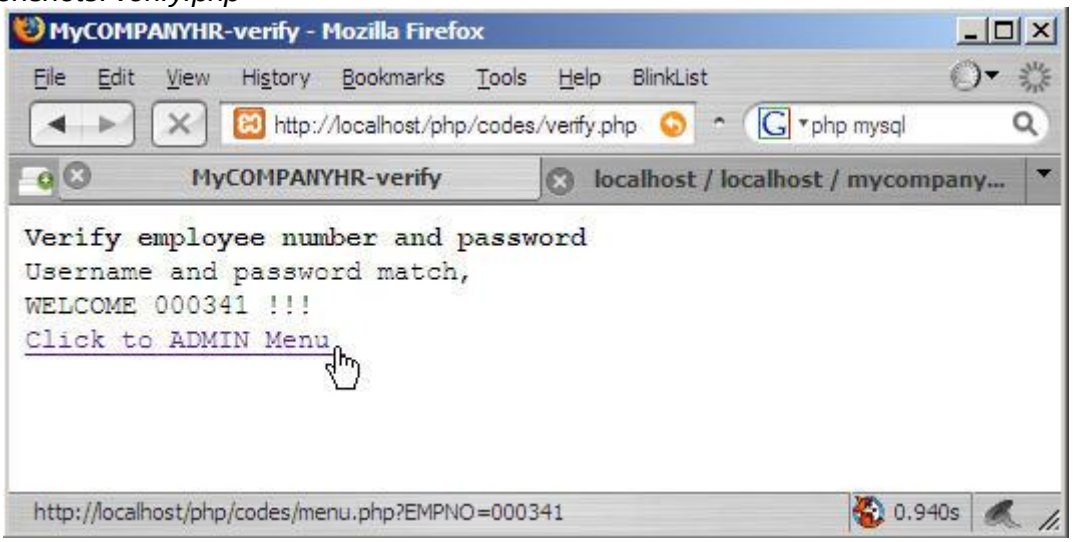
2

3

Code descriptions: *verify.php*

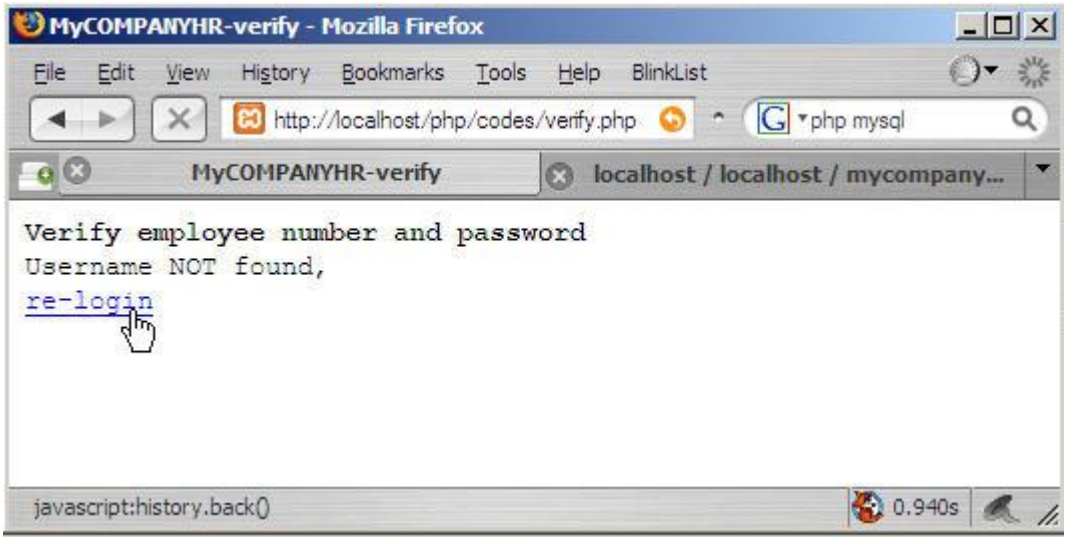
1. `$DBPASSWORD=$record['PASSWORD'];` – This script extracts the password (for the user by the EMPNO) from the table *adminusers*.
   `$USERPASSWORD=md5($PASSWORD);` – Meanwhile the password entered by the user from the previous form is extracted and decoded using MD5 function that also available in PHP.
   `if($DBPASSWORD==$USERPASSWORD){` – If the password entered by the user (after MD5 encoding) is equal to the MD5 encoded password in the *adminusers* table, than the username and password are matched.
2. This is the actions if the user is found in the *adminusers* table, but the password provided by the user is not match to the password stored in the table.
3. While this is the action if the username (provided by the user's EMPNO) is not found in the *adminusers* table.

Screenshots: *verify.php*



Screen shots: *verify.php* – verification succeed.

Screen shots: *verify.php* – password does not match.



Screen shots: *verify.php* – username not found.

Menu page for full administration features and limited access

The authorised user is directed to this page. Before the menu is displayed to the user, full information regarding the user's access level, his first name, last name and working department is retrieved.

The list of pages authorised to the user are depend on the user access level. For this exercise there are user level 1 and 2. The users with access level 1 has the full administration access. The menu contains all the administration features such as searching, insert, update and delete employees' record. While the users with access level 2 is limited access users, they can only search the employees' records and update their own personel particulars.

This is the SQL to select user's level, firstname, lastname, workdept, deptname. Use JOIN statement since most of the informations are separated in different table.

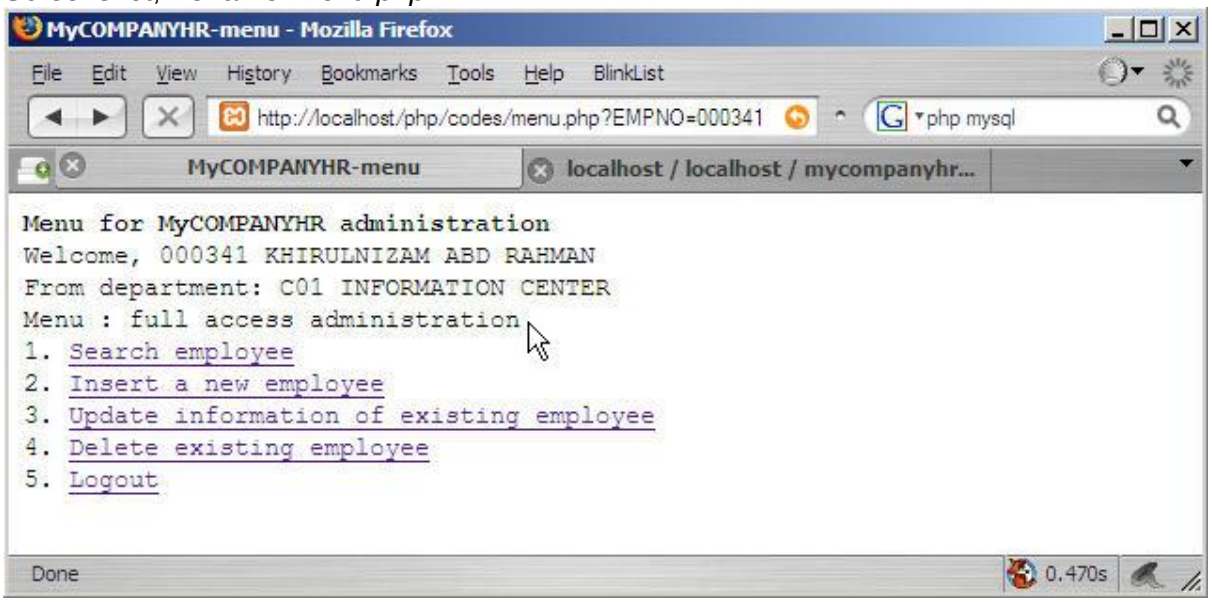
```
SELECT
      adminusers.EMPNO,
      adminusers.LEVEL,
      employee.FIRSTNAME,
      employee.LASTNAME,
      employee.WORKDEPT,
      department.DEPTNAME
FROM
adminusers
      INNER JOIN employee
            ON adminusers.EMPNO = employee.EMPNO
      INNER JOIN department
            ON employee.WORKDEPT = department.DEPTNO
WHERE adminusers.EMPNO='000341'
```

Filename: ***menu.php***

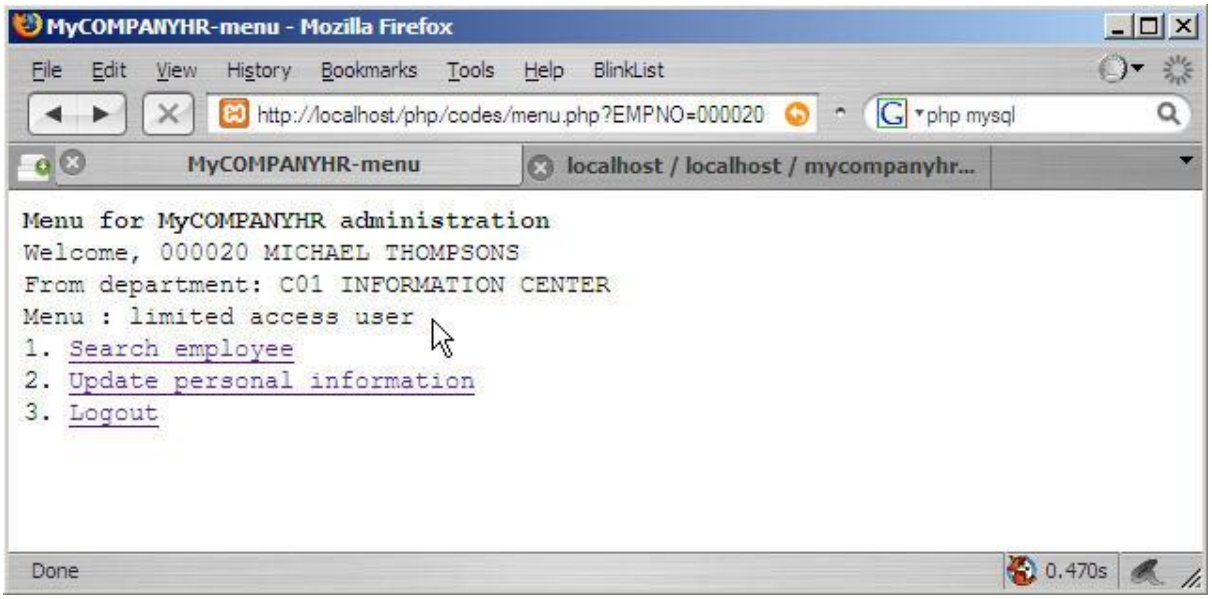
```
<?php
include "connection.php";
?>
<html>
<head>
<title>MyCOMPANYHR-menu</title>
</head>

<body>
<strong>Menu for MyCOMPANYHR administration</strong><br>
<?php
//script to display employees information
$EMPNO=$_GET['EMPNO'];
/*this SQL commad will fetch the employee's administration level,
firstname, lastname, workdept, deptname in their respective tables*/
$sql="SELECT
            adminusers.EMPNO,
            adminusers.LEVEL,
            employee.FIRSTNAME,
            employee.LASTNAME,
            employee.WORKDEPT,
            department.DEPTNAME
      FROM adminusers
            INNER JOIN employee
                  ON adminusers.EMPNO = employee.EMPNO
            INNER JOIN department
                  ON employee.WORKDEPT = department.DEPTNO
      WHERE adminusers.EMPNO='$EMPNO'";

$rs=mysqli_query($db, $sql);
$record=mysqli_fetch_array($rs);
$level=$record['LEVEL'];
$firstname=$record['FIRSTNAME'];
$lastname=$record['LASTNAME'];
$workdept=$record['WORKDEPT'];
$deptname=$record['DEPTNAME'];

?>
Welcome, <? echo "$EMPNO $firstname $lastname"?> <br>
From department: <? echo "$workdept $deptname"?> <br>

<?php
//this menu displays depending on the users level
//if level is 1, full access
//if level is 2, limited access
if($level==1){
?>
      Menu : full access administration<br>
      1. <a href="searchform.php">Search employee</a><br>
      2. <a href="forminsert.php">Insert a new employee</a><br>
      3. <a href="listing4update.php">Update information of
                  existing employee</a><br>
      4. <a href="listing4delete.php">Delete existing
                  employee</a><br>
```

```
      5. <a href="logout.php">Logout</a><br>
<?php
}//end level 1
else if($level==2){
?>
      Menu : limited access user <br>
      1. <a href="searchform.php">Search employee</a><br>
      2. <a href="formupdate-personal.php">Update personal
                information</a><br>
      3. <a href="logout.php">Logout</a><br>
<?php
}//end level 2
?>
</body>
</html>
```

Screenshot, filename: *menu.php*



Menu for users with access level 1.



Menu for users with access level 2.

Server side SESSION

So, what is the SESSION?
Session in term of server-side scripting concern is the connection between a client (browser) and the server that hosting the web application. Each time a client connects to a web server, there is a connection happens. The connection can be registered in the server and there are a lot of information regarding the connection that can be recorded in the web server. For example the web browser being used or the IP number.

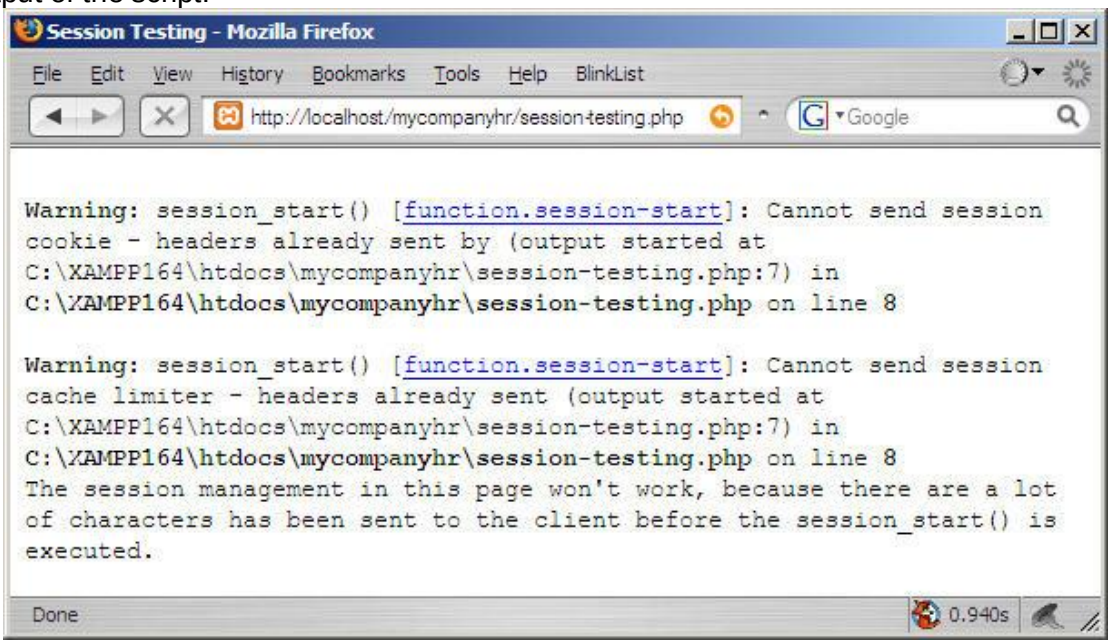
Starting a session
Each unique connection provides a unique session id. To enable the session management of a page, first the *session_start*() function must be executed. This function can only work if there is no information from the page was sent to the client, not even a space or a single bit.

For example, try the following code.

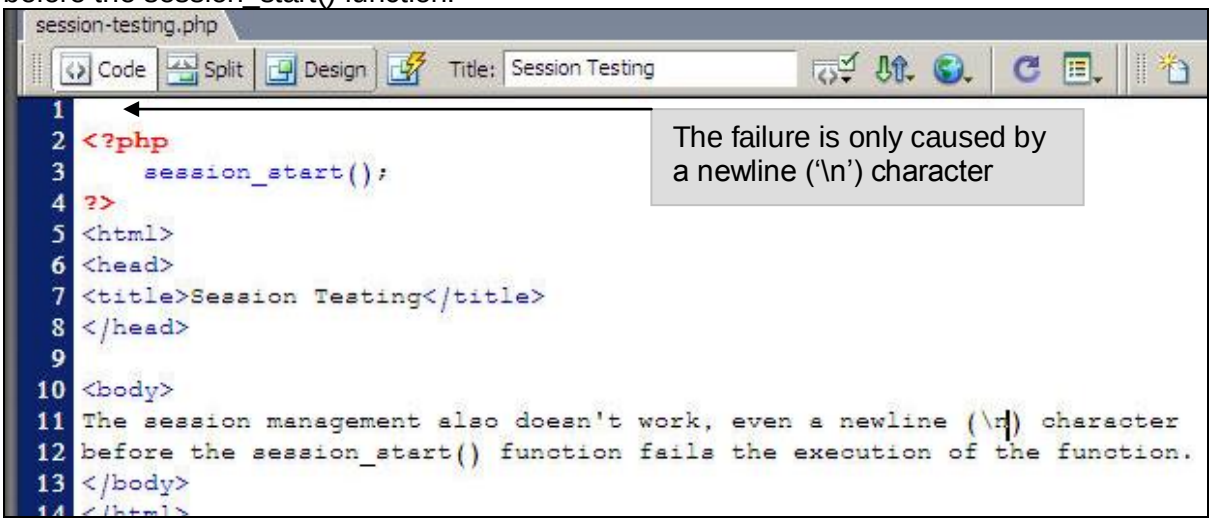
```
<html>
<head>
<title>Session Testing</title>
</head>

<body>
<?php
      session_start();
      //trying to start session management
?>
The session management in this page won't work, because there are a
lot of characters has been sent to the client before the
session_start() is executed.
</body>
</html>
```

This information has been sent to the client. It is the reason why the sesion_start() function fail to executed.

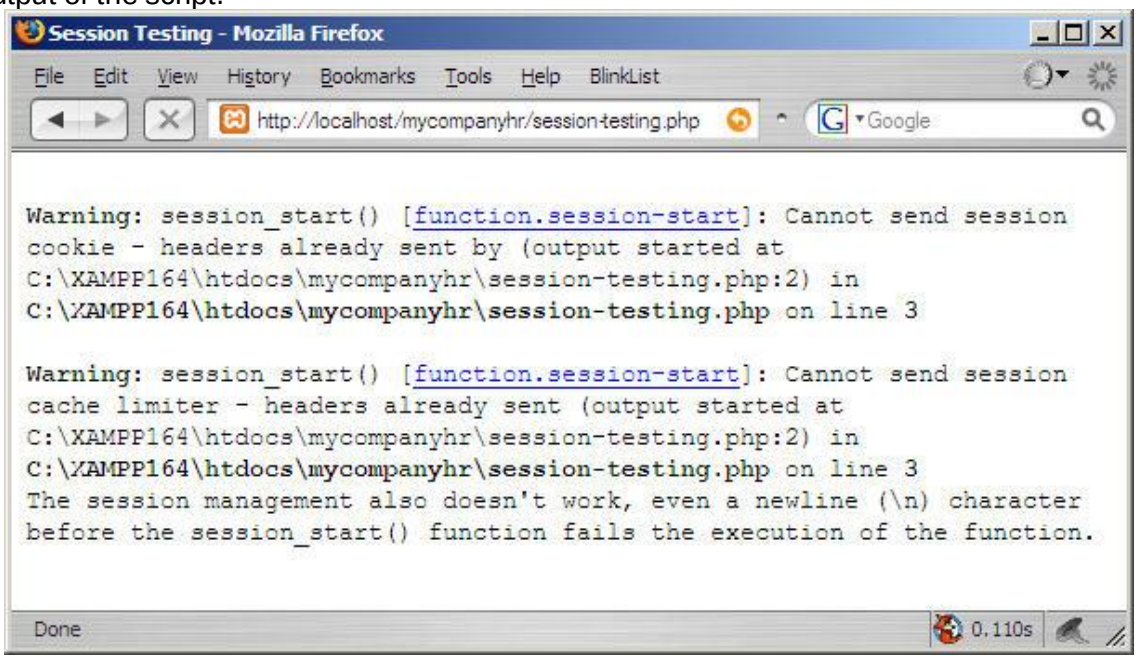
Output of the script.



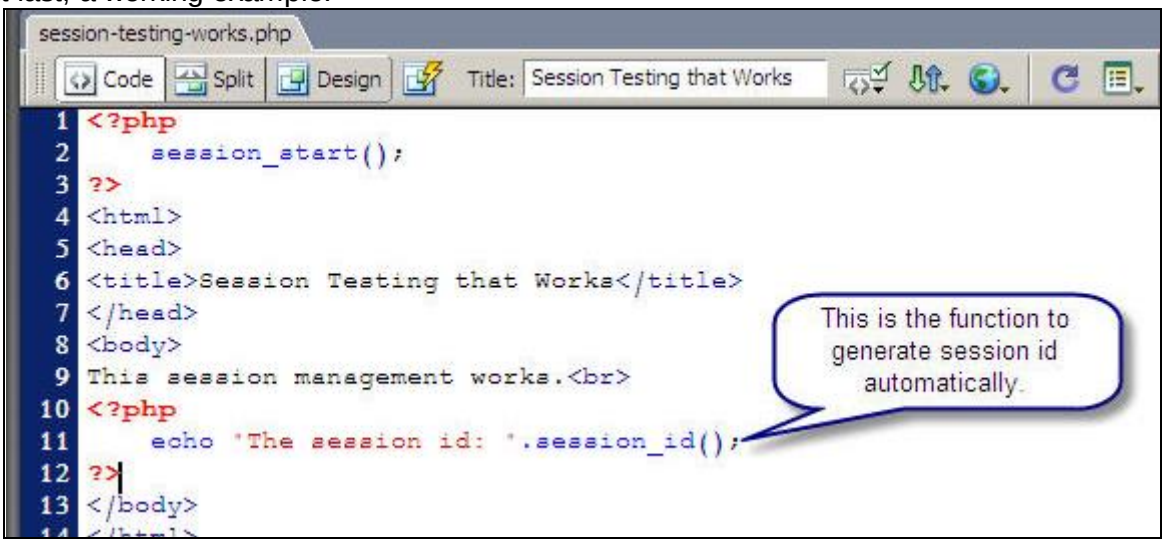
This example also doesn't work. The problem is caused by only a newline ('\n') character before the session_start() function.



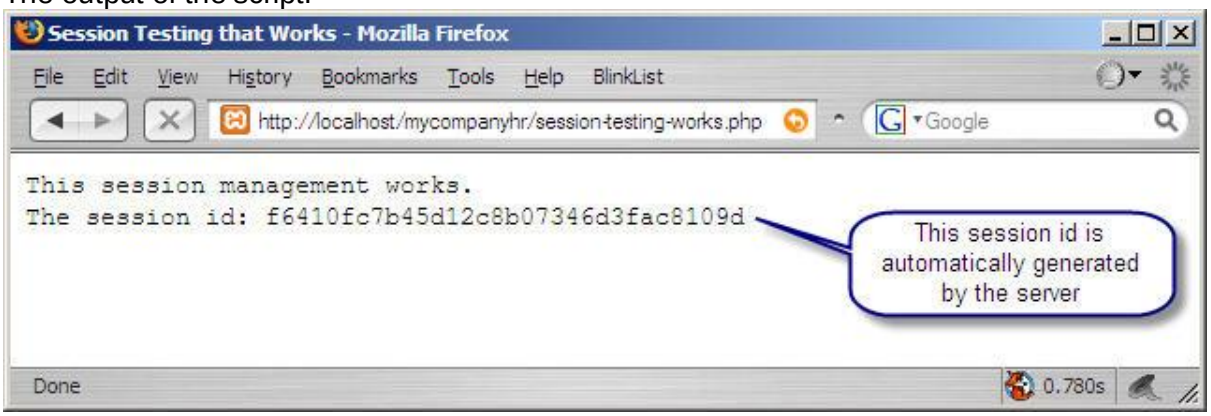
Output of the script.



At last, a working example.

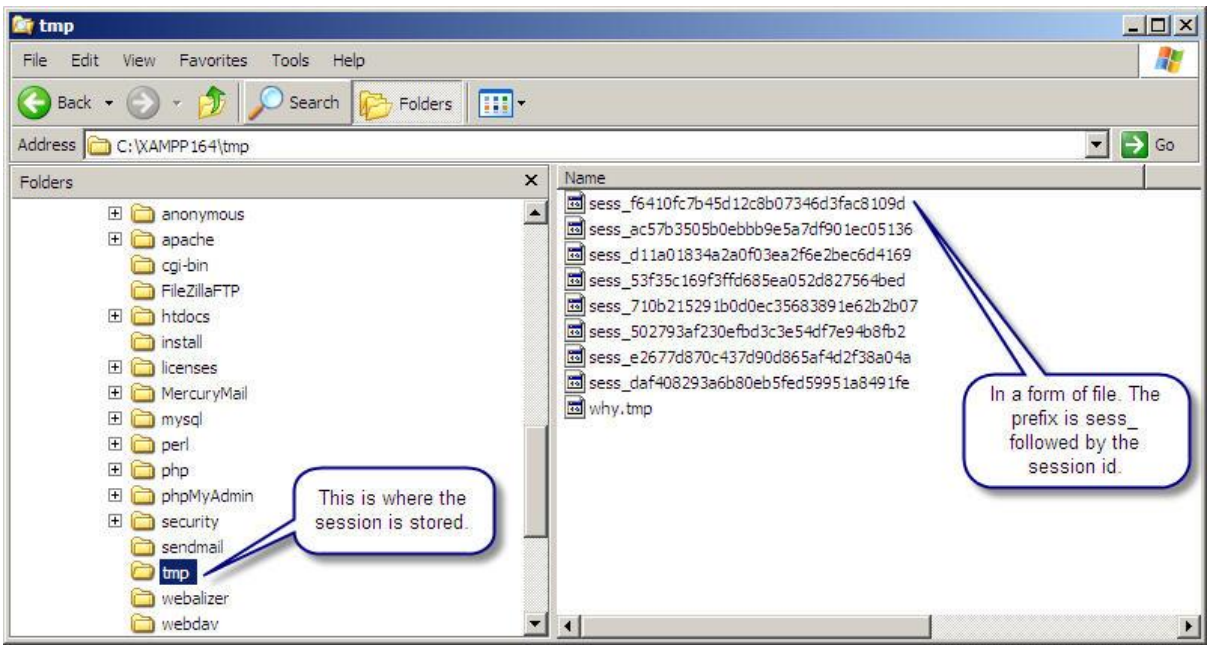


The output of the script.



So where is this session information is stored?

Currently we are using XAMPP. By default the Apache web server store the session information in the tmp directory. Each session will generate a file with the sess_ as the prefix, and followed by the session id.



Try to open the file and you will nothing is stored in the file where the latest session was created using the script in previous page. This is because there is no session variable is registered.

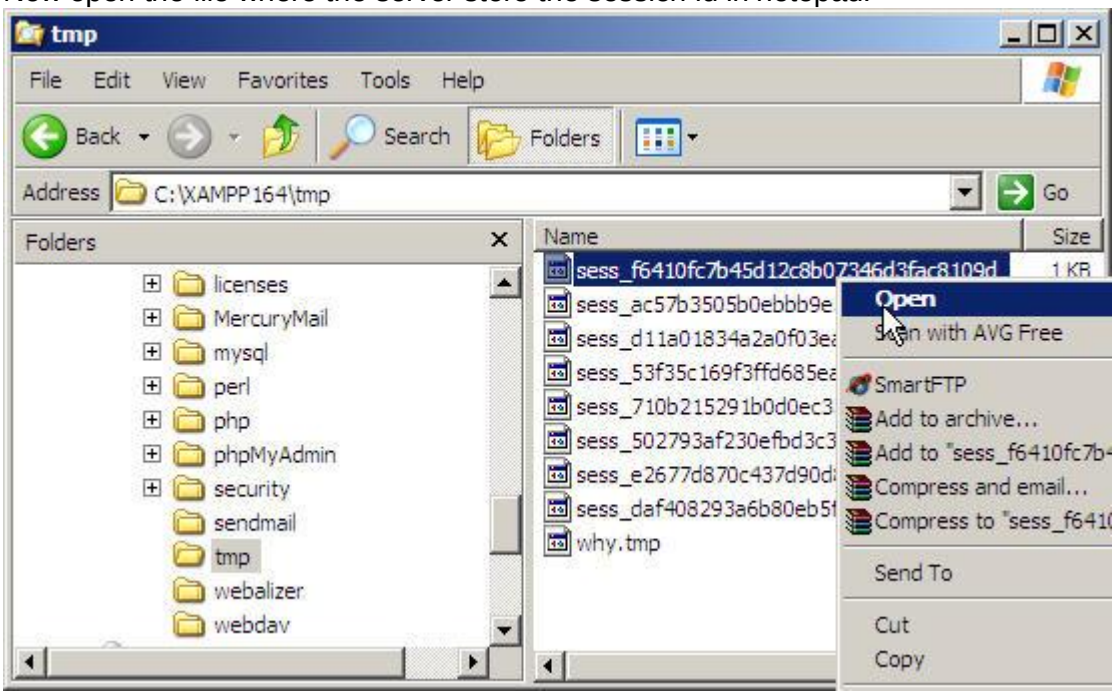
Session variable registration need to be done in order to store values to be used as long as the session (or the connection) is established. For example, in a web application there are few files connected to each other. In order to maintain a certain amount of value to be used for every pages in the system, we need to store it in the session variable. For example the username of the user who are using the system.

Registering a session variable
To store the username (so that all the pages in the system will be displaying the same username) we need to register a session variable. The following script is to register few session variable.

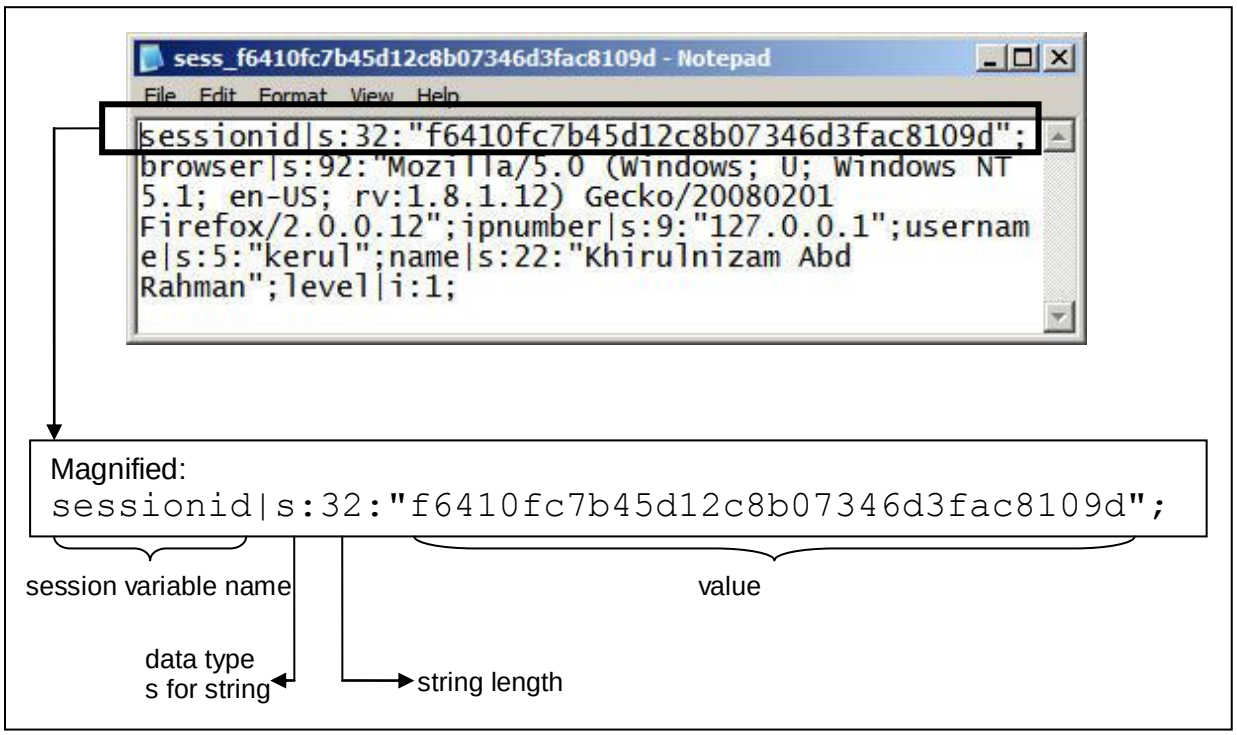
```
<?php  session_start();  ?>
<html>
<head>
<title>Session Registration</title>
</head>
<body>
This page is to register user's information.<br>
<?php
if(!isset($_SESSION['sessionid'])){
      $_SESSION['sessionid']=session_id();//session id
      $_SESSION['browser']=$_SERVER['HTTP_USER_AGENT'];//browser
      $_SESSION['ipnumber']=$_SERVER['REMOTE_ADDR'];//client's ip
      $_SESSION['username']="kerul";//the username
      $_SESSION['name']="Khirulnizam Abd Rahman";//full name
      $_SESSION['level']=1;//user access level
}
?>
</body>
</html>
```

The script provide the registration of four session variable namely; *sessionid*, *username*, *name* and *level*, with their respective value.

Now open the file where the server store the session id in notepad.



This is the content of the session file.



Using the session variable.

```
<?php  session_start(); ?>
<html>
<head>
<title>Session Usage</title>
</head>
<body>
This page is to use user's information stored in session.<br>
<?php
      echo $_SESSION['sessionid'].'<br>';
      echo $_SESSION['browser'].'<br>';
      echo $_SESSION['ipnumber'].'<br>';
      echo $_SESSION['username'] .'<br>';
      echo $_SESSION['name'] .'<br>';
      echo $_SESSION['level'] .'<br>';
?>
</body>
</html>
```

Checking the session variable.

```
<?php
      session_start();
?>
<html>
<head>
<title>Session Checking</title>
</head>
<body>
This page is to check whether user's information are stored in the
session.<br>
<?php
      if (isset($_SESSION['sessionid'])){
            echo $_SESSION['sessionid'] .'<br>';
            echo $_SESSION['browser'].'<br>';
            echo $_SESSION['ipnumber'].'<br>';
            echo $_SESSION['username'] .'<br>';
            echo $_SESSION['name'] .'<br>';
            echo $_SESSION['level'] .'<br>';
      }
      else{
            $_SESSION['sessionid']=session_id();
            $_SESSION['browser']=$_SERVER['HTTP_USER_AGENT'];
            $_SESSION['ipnumber']=$_SERVER['REMOTE_ADDR'];
            $_SESSION['username']="kerul";
            $_SESSION['name']="Khirulnizam Abd Rahman";
            $_SESSION['level']=1;
      }
?>
</body>
</html>
```

Destroy the session variable.

```
<?php
      session_start();
?>
<html>
<head>
<title>Session Checking</title>
</head>
<body>
This page is to check whether user's information are stored in the
session.<br>
<?php
      if (isset($_SESSION['sessionid'])){
            session_destroy();//this to destroy all session info
      }
?>
</body>
</html>
```

Securing the System's Parameter using SESSION

So, why use SESSION instead? Don't you already have the user login facilities?

User login facility is just the guard upon the entrance. A thief doesn't really need permission from the guard to enter. They simply hope over the fence, cut a hole in the fence, or dig a passage under the fence to sneak in to a premise.

So, for the sake of the premise, we should position a guard for each of the building in the premise compound. It would act as a double layer protection against the unauthorized person. Although a thief could possibly bypass the security guard at the entrance, or sneak in through a hole in the fence, they need to bypass another security personel to enter a building.

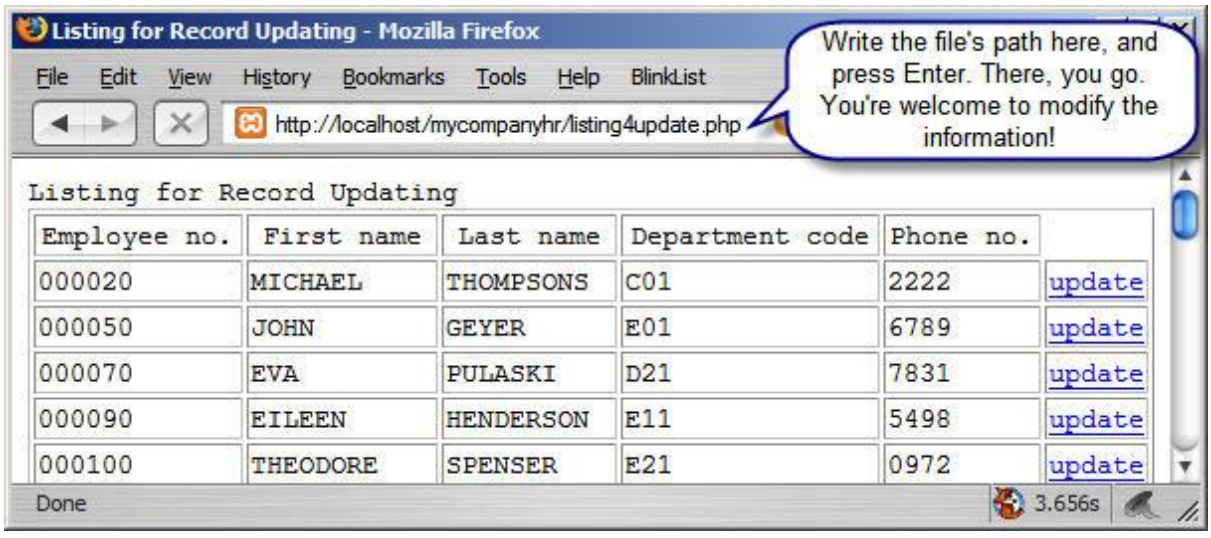
As for the web application concern, we have implemented the first layer protection by applying the user login process. This login will determine only the authorized personel is allowed to access the menu page. This menu page consists of all the links to another pages that provides administration facilities.

However, a hacker doesn't really need the menu page. Anybody can access the administration application if he/she knows the path and the file name of the exact page. For example, the page to update an employee details are *listing4update.php* (we are still using the *mycompanyhr* project for this discussion). We can simply view the address of the page from the address bar (that is why it is important not to disclose the file's path while viewing the file's content in the browser-this issue will be discussed in the later chapter).

So, try typing this path;

http://localhost/mycompanyhr/listing4update.php

in the address bar with the project sample installed in your localhost. Although you did not login (by providing username and passwrod), still you could possibly view the content and are permitted to even update the information.



Applying check session in the system.

Before that lets modify the login.php file. Change the form's action to *verify-session.php*.

```
<html>
<head>
<title>MyCOMPANYHR-Login Form</title>
</head>
<body>

<strong>Login to MYCOMPANYHR system</strong>
<form name="formlogin" method="post" action="verify-session.php">
<table width="400"  border="0">
  <tr>
    <td>Employee No</td>
    <td><input name="EMPNO" type="text" maxlength="6">
      Eg: 999999 </td>
  </tr>
  <tr>
    <td>Password</td>
    <td><input name="PASSWORD" type="password"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="submit" type="submit" value="Login"></td>
  </tr>
</table>
</form>

</body>
</html>
```

In order to implement the session control to enhance you system's security, register the connection details at the point where the system has verify the username exists and the password match. So we need to get back on the verification script as in *verify.php* page in Chapter 14.

Rename the file with *verify-session.php*.

```
<?php
//session begins
session_start(); //set the users full name
$EMPNO=$_POST['EMPNO'];
$PASSWORD=$_POST['PASSWORD'];
include "connection.php";
$sql="select * from adminusers where EMPNO='$EMPNO'";
$rs=mysqli_query($db, $sql);
?>
<html>
<head>
<title>MyCOMPANYHR-verify</title>
</head>

<body>
<strong>Verify employee number and password</strong><br>
<?php
if(mysqli_num_rows($rs)==1){ //found one user
      $record=mysqli_fetch_array($rs);
      $DBPASSWORD=$record['PASSWORD'];//password from database
      $USERPASSWORD=md5($PASSWORD);//MD5 password key-in by user
      if($DBPASSWORD==$USERPASSWORD){
            //compare password from database against password entered by user
            echo "Username and password match,<br>";
            echo "WELCOME $EMPNO !!!<br>";
            //register the session for the user
            $_SESSION["sessionid"]=session_id();
            $_SESSION["empno"]=$EMPNO;
            echo "<a href='menu-session.php'>ADMIN Menu </a><br>";
      }
      else{
            echo "Username found, but password NOT match,<br>";
            echo "<a href='javascript:history.back()'>re-enter
                  password</a><br>";
      }
}
else{
      echo "Username NOT found,<br>";
      echo "<a href='javascript:history.back()'>re-login</a><br>";
}
?>
</body>
</html>
```

After this point of verification, we need to implement session checking for all the administration pages.

Try this script for session checking mechanism. And then include this *checksession.php* file in all the pages involved.

```
<?php
//this script is to check session to verify user login
session_start();
if(!isset($_SESSION["empno"])){ //if session NOT set
      echo "You are not logged in,
            <a href='login-session.php'>click here to login.</a>";
            exit(0);
}
?>
```

`if(!isset($_SESSION["empno"]))` – this line checks whether the session for the user with the particular employee number is registered in the server session. If the session is not set, then user will be asked to go to the login page.

`exit(0);` – this statement terminate the execution the current page. This is to make sure the page is not sent to the unauthorized user.

Reconfigure the *menu.php*, and save as *menu-session.php*.

```
<?php
include "checksession.php";
include "connection.php";
?>
<html>
<head>
<title>MyCOMPANYHR-menu</title>
</head>

<body>
<strong>Menu for MyCOMPANYHR administration</strong><br>
<?php
//script to display employees information
$empno=$_SESSION['empno'];
/*this SQL commad will fetch the employee's administration level,
firstname, lastname, workdept, deptname in their respective tables*/
$sql="SELECT
            adminusers.EMPNO,
            adminusers.LEVEL,
            employee.FIRSTNAME,
            employee.LASTNAME,
            employee.WORKDEPT,
            department.DEPTNAME
      FROM adminusers
            INNER JOIN employee
                  ON adminusers.EMPNO = employee.EMPNO
            INNER JOIN department
                  ON employee.WORKDEPT = department.DEPTNO
      WHERE adminusers.EMPNO='$empno'";
$rs=mysqli_query($db, $sql);
$record=mysqli_fetch_array($rs);
$_SESSION['level']=$record['LEVEL'];
```

```
$_SESSION['fullname']=$record['FIRSTNAME']." ".$record['LASTNAME'];
$_SESSION['workdept']=$record['WORKDEPT'];
$_SESSION['deptname']=$record['DEPTNAME'];

?>
Welcome, <? echo "$empno ".$_SESSION['fullname']?> <br>
From department:
     <? echo $_SESSION['workdept']." ".$_SESSION['deptname']?> <br>

<?php
//this menu displays depending on the users level
//if level is 1, full access
//if level is 2, limited access
if($_SESSION['level']==1){
?>
     Menu : full access administration<br>
     1. <a href="searchform.php">Search employee</a><br>
     2. <a href="forminsert.php">Insert a new employee</a><br>
     3. <a href="listing4update.php">Update information of existing
          employee</a><br>
     4. <a href="listing4delete.php">Delete existing
          employee</a><br>
     5. <a href="logout.php">Logout</a><br>
<?php
}
else if($_SESSION['level']==2){
?>
     Menu : limited access user <br>
     1. <a href="searchform.php">Search employee</a><br>
     2. <a href="formupdate-personal.php">Update personel
          information</a><br>
     3. <a href="logout.php">Logout</a><br>
<?php
}

//display footer
include "footer.php";
?>
</body>
</html>
```

So what is in *footer.php* file? The file contains the script to display the current user's information, a link to *menu-session.php*, and also a link to the *logout.php* file. The purpose of this template is to provide easier navigation for the user.

Filename; *footer.php*.

```
<style type="text/css">
<!--
.Copyright {
      color: #FF6600;
      font-weight: bold;
      font-family: Arial, Helvetica, sans-serif;
      font-size: 12px;
}
.style2 {font-size: 14px}
.style4 {font-size: 12px}
-->
</style>
<table width="100%"  border="0">
  <tr>
    <td><div align="center" class="Copyright">
      <div align="left">Copyright  <? echo date('Y');?> &copy; All
Rights Reserved to Kerul </div>
    </div></td>
  </tr>
</table>
<table width="100%"  border="0" bgcolor="#FF6600">
  <tr>
    <td         bordercolor="#FF9933"         bgcolor="#FF9933"><div
align="left"><span class="style2">
      <a href="menu-session.php"> menu</a> -
      <a href="logout.php">logout</a> </span>
      <strong>MyCOMPANYHR</strong>-
      <span class="style4">user (<? echo $_SESSION['fullname']; ?>)
      </span></div></td>
  </tr>
</table>
```

Logout facility. – this is to destroy the session information stored in the server. It is a very important to delete all the session information should the user intend to leave the web application.

Filename; logout.php

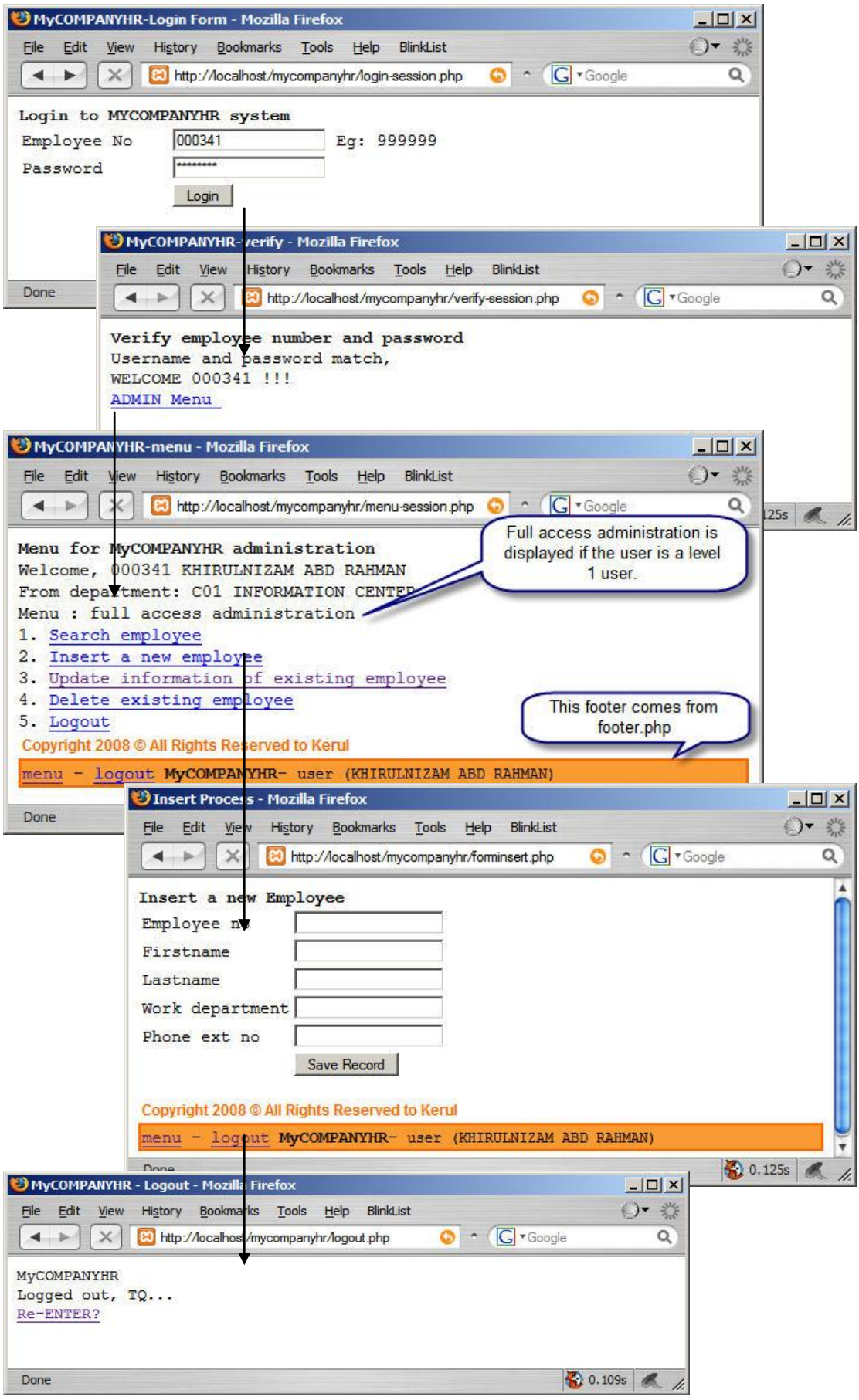
```
<? //to destroy session
      session_start();
      session_destroy();
?>
<html>
<head>
<title>MyCOMPANYHR - Logout </title>
</head>

<body>
MyCOMPANYHR<br>
Logged out, TQ...<br>
<a href="login-session.php">Re-ENTER?</a><br>
</body>
</html>
```

The next step is to include *checksession.php* and *logout.php* in all the pages involved in insert, update and delete facilities. If you think the data is highly confidential, you might also include the files in the listing or search facilities. Normally searching or record listing is open to public.

Let insert both files to the *forminsert.php*.

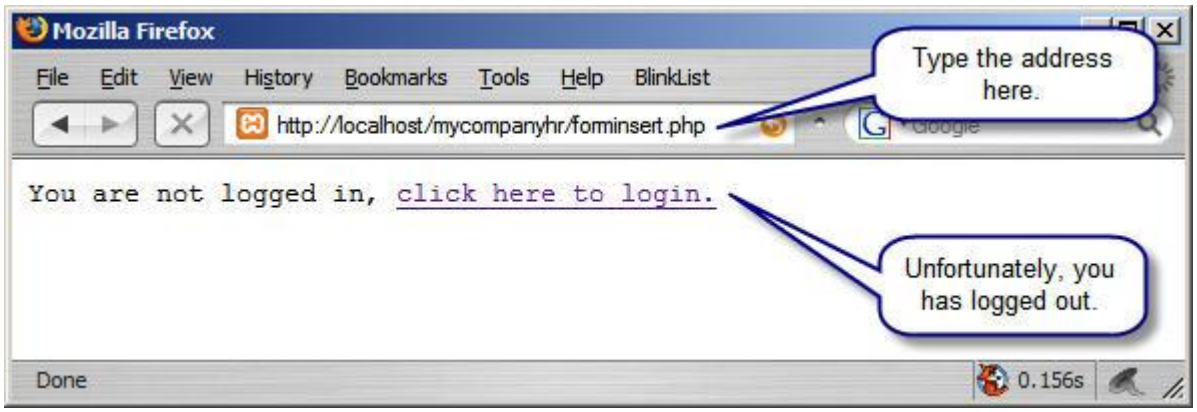
```
<?php include "checksession.php"; ?>
<html>
<head>
<title>Insert Process</title>
</head>
<body>
<strong>Insert a new Employee </strong><br>
<form name="insert" action="insertrecord.php" method="get">

<table border="0">
  <tr>
    <td width="40%">Employee no</td>
    <td width="60%"><input name="EMPNO" type="text"></td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Work department</td>
    <td><input name="WORKDEPT" type="text" maxlength="3"></td>
  </tr>
  <tr>
    <td>Phone ext no</td>
    <td><input name="PHONENO" type="text" maxlength="4"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="save" type="submit" value="Save Record"></td>
  </tr>
</table>
</form>
<?php include "footer.php"; ?>
</body>
</html>
```

Screen shots.



Right after the logout process, try to type this path in the address bar;
http://localhost/mycompanyhr/forminsert.php

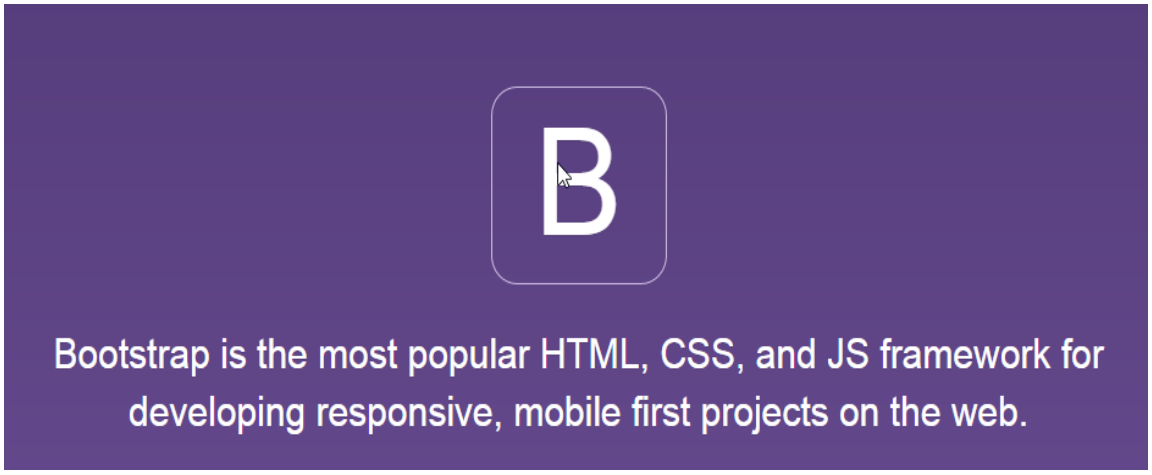
Unfortunately you cannot see the form to insert new employee. Why this is happening? The checksession.php restrict the user from viewing the page, since the user has logged out. The user need to login again in order to get the session registered.



What is BootStrap?

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. (http://www.w3schools.com/bootstrap)

Available here - http://getbootstrap.com/



One framework, every device.

Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.

Full of features

With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

HTML without framework is dull. Doing hard-coded CSS and JS are quite difficult with no promising result on cross platform compatibility. So we decided to explore BootStrap as others said it is the most popular web framework.

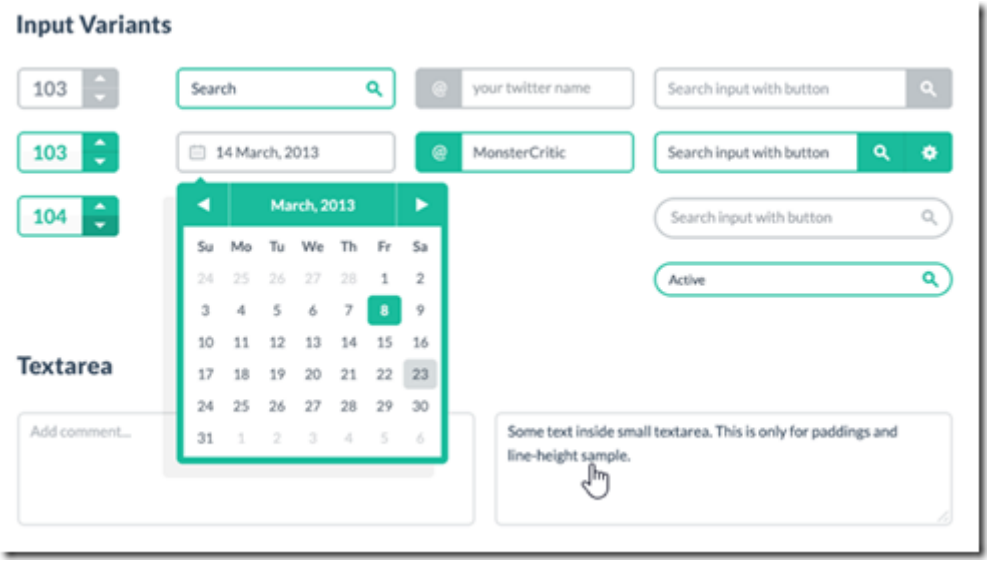
Why you need Flat-UI? Seems like a beautiful theme to make my site look professional. Anyway you could get variety of BootStrap theme out there, feel free to select here http://bootstraphero.com/the-big-badass-list-of-twitter-bootstrap-resources/

Flat-UI is from DesignModo - http://designmodo.com/flat/

What Flat-UI from DesignModo have to offer you?


Button States
Turquoise Button
Static
Static
Static
Turquoise Button
Hover
Hover
Hover
Turquoise Button
Pressed
Pressed
Pressed
Cool Button
Hot Button
Bright Button
Disabled
Cool Button
Hot Button
Bright Button
Disabled
Cool Button
Hot Button
Bright Button
Disabled
Button Sizes
Huge Button
Large Button
Standard Button
Small Button


Dropdowns
My Account
My Account
My Account
My Profile
My Friends
Messages
My Settings
Log Out
Dropdown Button
Dropdown Button
Dropdown Button
Dropdown Button
Sub Menu Element
Sub Menu Selected
Sub Menu Selected
Drama
Comedy
Thriller
Western
Icon & Color
Element One
Element Selected
Element Three


Input Variants
103
103
104
Search
14 March, 2013
March, 2013
your twitter name
MonsterCritic
Search input with button
Search input with button
Search input with button
Active
Textarea
Add comment...
Some text inside small textarea. This is only for paddings and line-height sample.

What is Responsive web design?

Definition: Responsive Web Design is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good (visually appealing) on any screen size such as desktop, tablet, phablet or mobile phone.

Example of Responsive Web Layout

Both websites below are displayed from the same HTML code source, however the size of the display are different. The left displayed on desktop screen, while the right displayed on a mobile phone. Previously web developer need to have two different HTML source code; one source is the main desktop layout, and another one is downgraded code to suite mobile phone small screen size. With responsive web layout design, this will be an issue of the past.

Example of Responsive Layout: the display on the left is desktop view, and the right one is mobile phone view.

HTLM code requirements

```
<!DOCTYPE html>
<html lang="en">
<head>
      <title>Bootstrap Case 2-Columns</title>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-
      scale=1">
      <link rel="stylesheet"
      href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/boots
      trap.min.css">
      <script
      src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquer
      y.min.js">
      </script>
      <script
      src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstr
      ap.min.js">
      </script>
</head>
<body>
      … Content is here…
</body>
</html>
```

This is an important meta information that is required in your responsive HTML head;

```
      <meta name="viewport" content="width=device-width, initial-
      scale=1">
```

This line define that your HTML page is optimized for mobile device screen.

Importing the Bootstrap framework.
Below are three framework calling for Bootstrap implementation in your HTML page. These three libraries are imported from the cloud. You can also download and store the libraries in your own server, which for us personally the better way to do it. Refer to the project example for hosting the framework in your own server.

1. CSS library

```
<link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstra
p.min.css">
```

2. JavaScript library jQuery

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.m
in.js">
</script>
```

3. JavaScript library for BootStrap

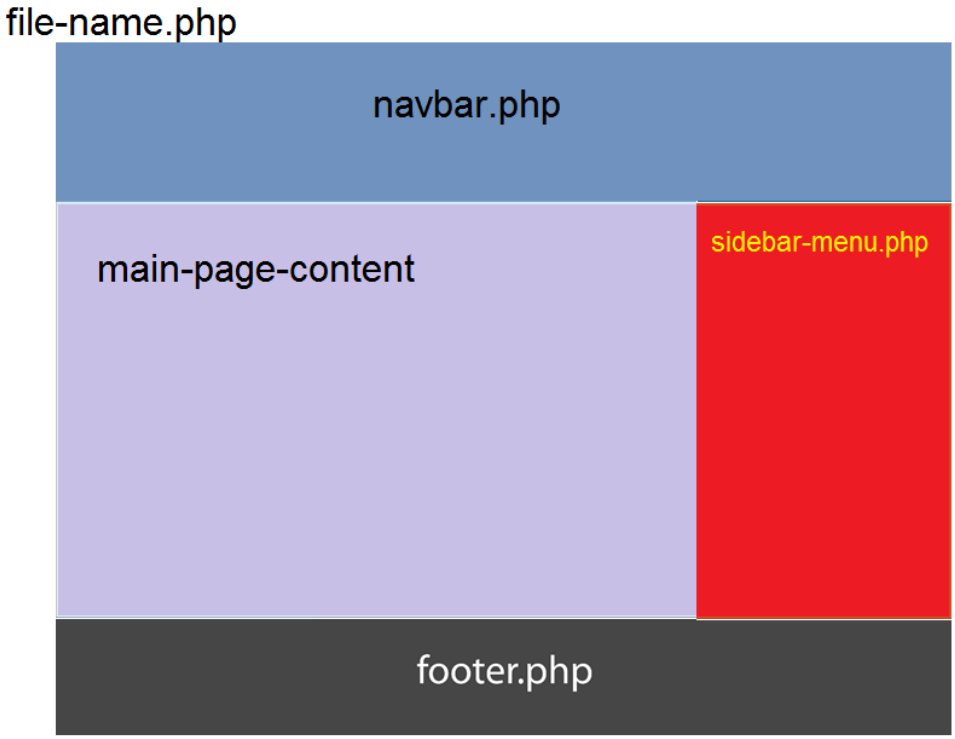
```
<script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.
min.js">
</script>
```

Project Example

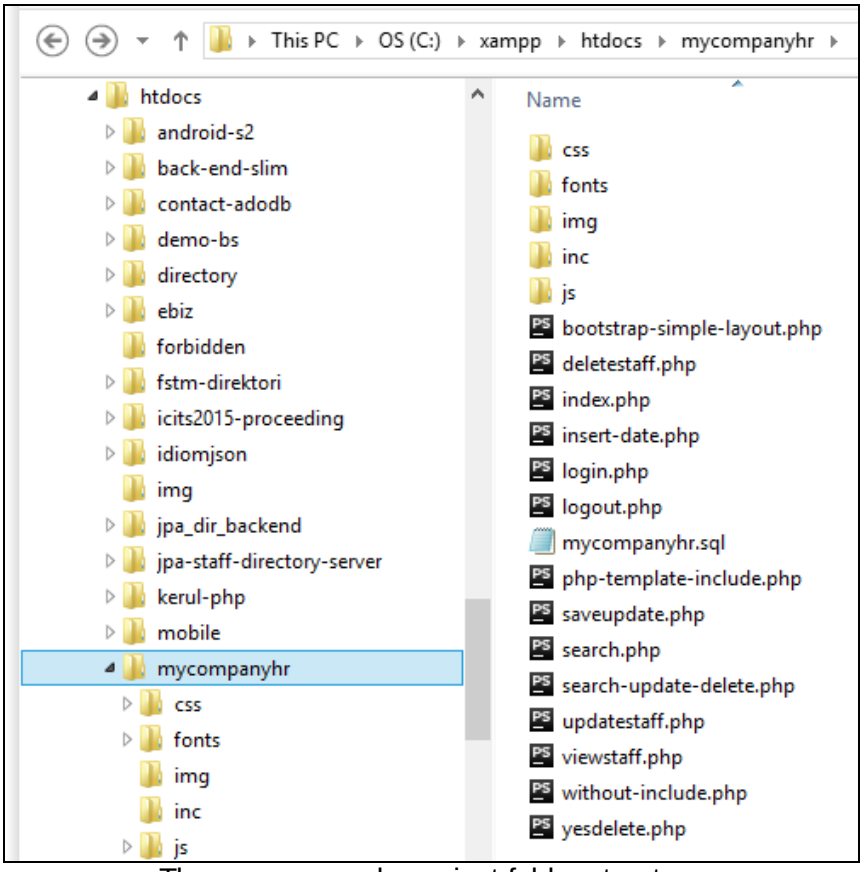
We would like to brief the project MyCOMPANYHR – a simple web based system to manage employees' profiles.

Below is the simple layouting. And we use a lot of PHP file include in the project.

- The include statement includes and evaluates the specified file.
- When a file is included, the code it contains inherits the variable scope of the line on which the include occurs. Any variables available at that line in the calling file will be available within the called file, from that point forward. However, all functions and classes defined in the included file have the global scope.
- To learn more, head to php.net/include



The main layout of MyCOMPANYHR project.



The mycompanyhr project folder structure

Working with the project template.

- Download the template here - http://bit.ly/mycompanyhr
- You can import the MySQL database by importing the *mycompanyhr.sql* into your MySQL server. Use import feature available in the *phpMyAdmin*.
- The template file is php-db-template.php
- The files for inclusion are available at folder ***inc***
    - dbconn.php
    - navbar.php
    - sidebar-menu.php
    - footer.php
- CSS and JS folders are also available, contain the FLAT-UI BootStrap framework.

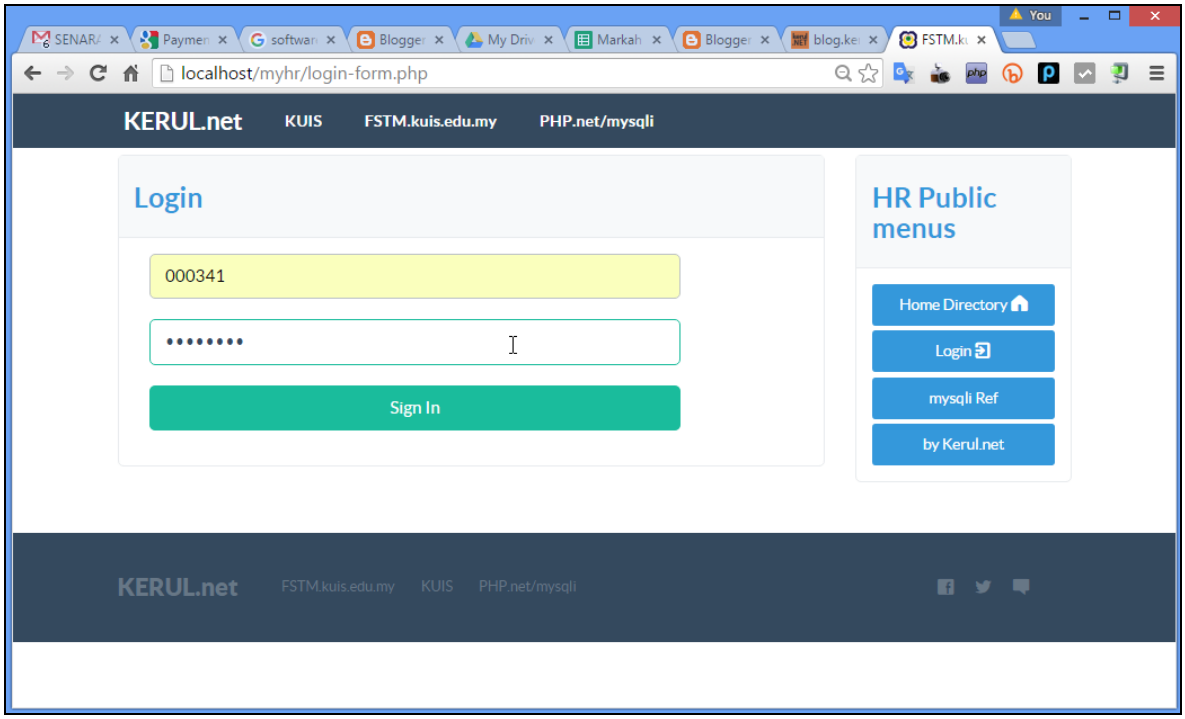
The main PHP files contains code as below.

```
<?php
//include the database connectivity setting
include ("inc/dbconn.php");
//include the navigation bar
include ("inc/header-navbar.php");?>

<div class="container">
      <br>
      <br>
  <div class="row">
        <div class="col-md-9" name="maincontent" id="maincontent">
            <!-- ***********Edit your content STARTS from here******** -->

            THE PAGE CONTENTS ARE HERE

            <!-- ***********Edit your content ENDS here******** -->

      </div><!-- end main content -->


      <?php
            //include the sidebar menu
            include ("inc/sidebar-menu.php");
      ?>
  </div><!-- end row -->

</div><!-- end container -->


<?php
//include the footer
include ("inc/footer.php");?>
```

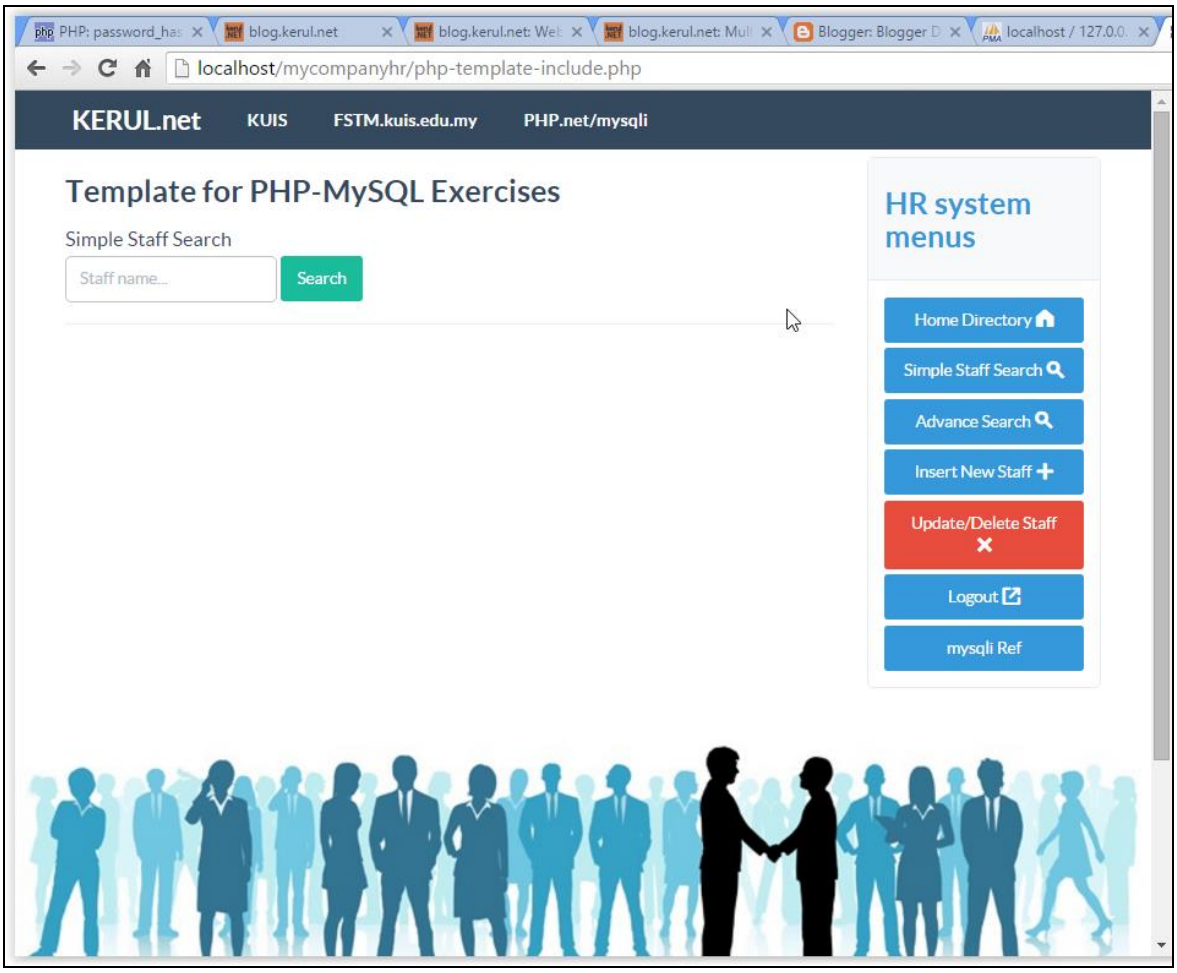
Notice the comment below, those are HTML comment. Write your PHP codes inside the border.

```
<!-- ***********Edit your content STARTS from here******** -->

            THE PAGE CONTENTS ARE HERE

<!-- ***********Edit your content ENDS here******** -->
```

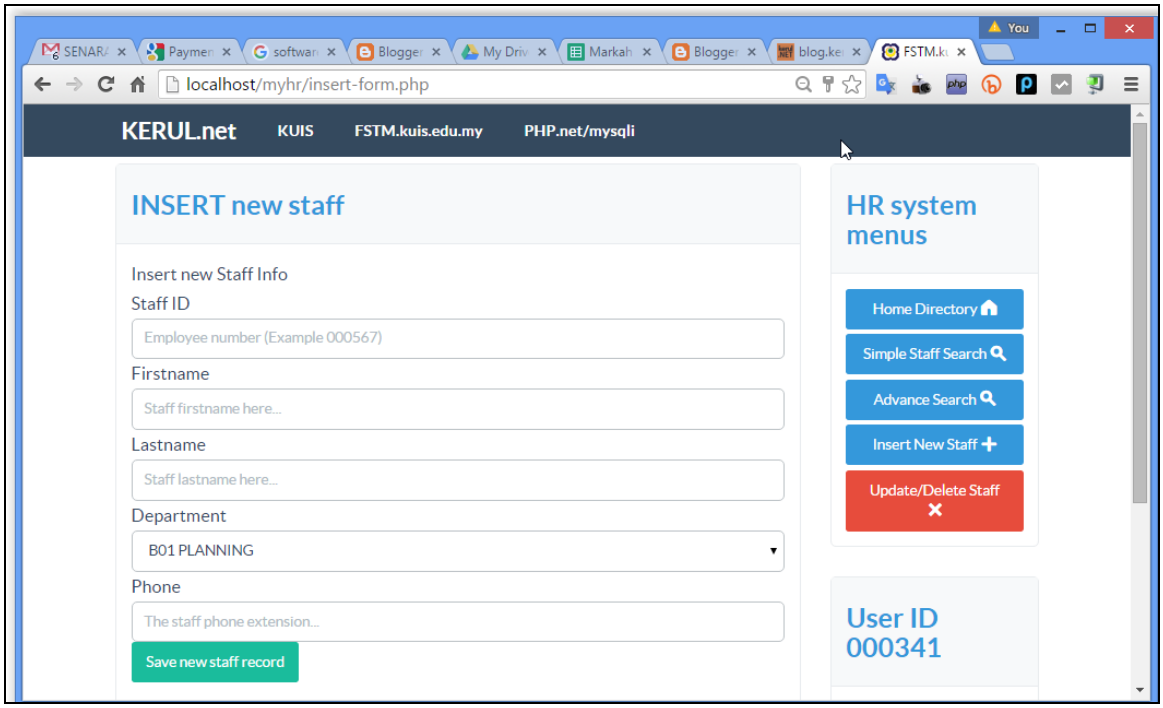
Some screenshots of the available features of the simple system.



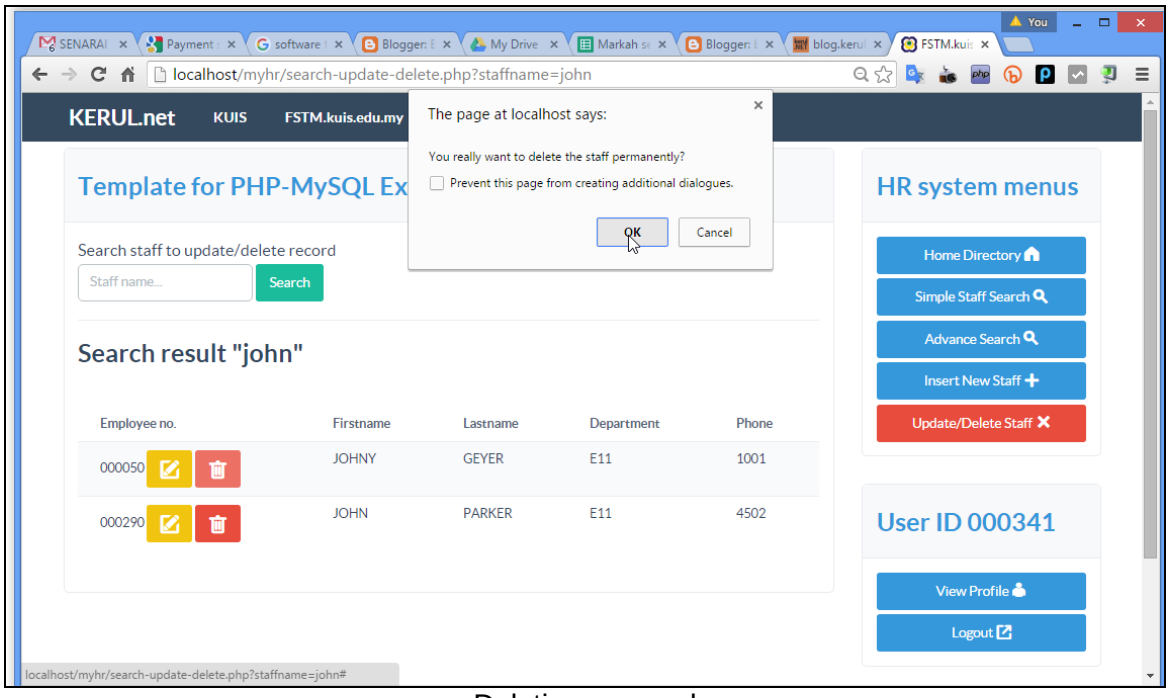
Login page



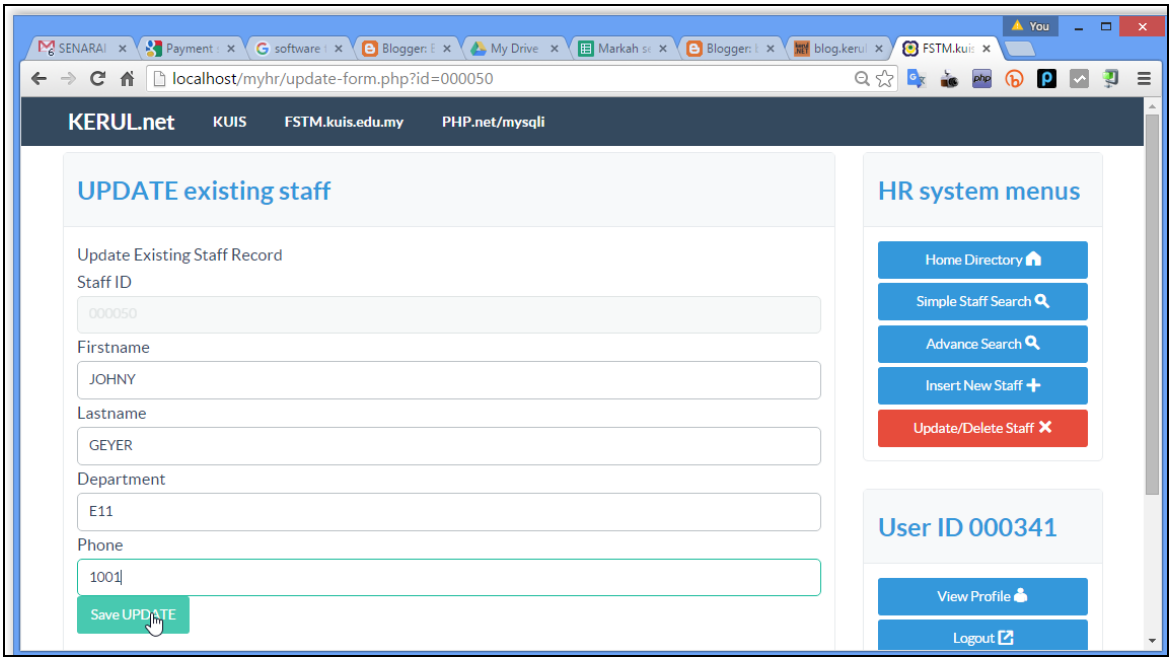
Employee search feature



Inserting a new record



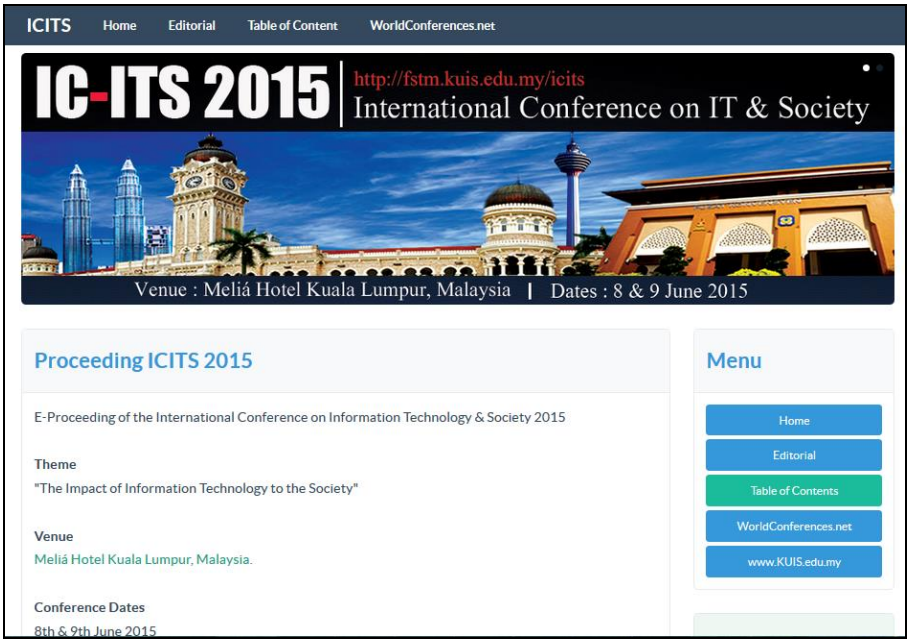
Deleting a record



Updating a record

EXERCISE

Develop a simple proceeding management system to facilitate an administrator to manage the proceedings paper of an academic conference. An academic proceeding paper is paperwork that is presented in an academic conference.
Administrator's page: The system could store (at least) information of the PDF file, paper id, paper title, name of authors, institution, corresponding email, abstract and keywords in the database.
The public page could display the list of proceedings as below to the public user.



Template for the exercise – Proceeding Management System.
Download here –
https://drive.google.com/file/d/0B5_Hw_xzXWcXaElxeENBTDB2aXc/view?usp=sharing

String Functions

STRING FUNCTIONS

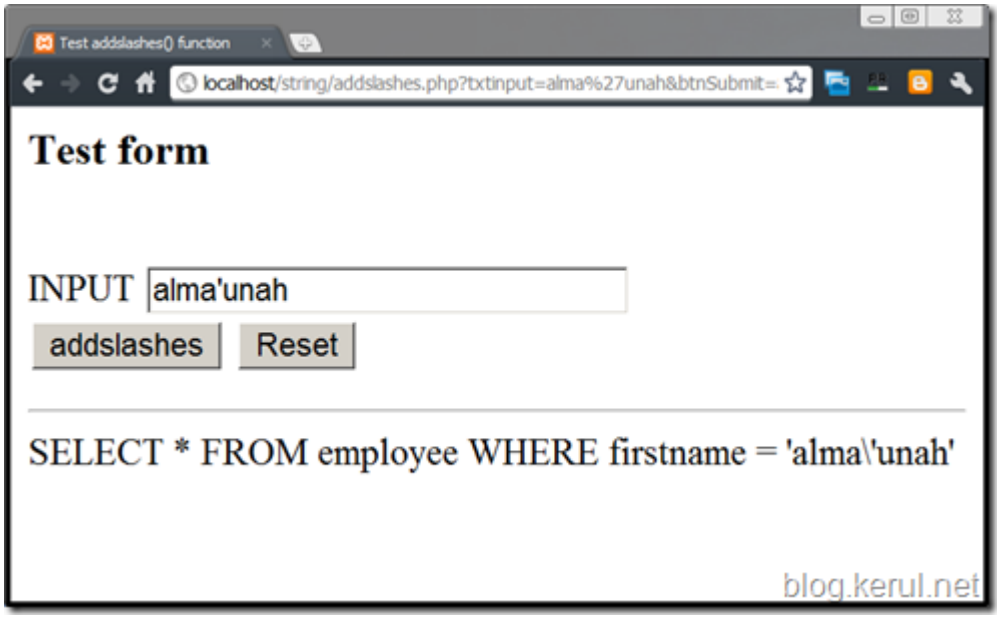
String functions are used in computer programming languages to manipulate a string or query information about a string (some do both).

Frequently used STRING functions

EXAMPLE 1
`addslashes` — Quote string with slashes

In the example, the addslashes function prevents the SQL string from being misinterpreted by the single quote in the middle of the search criteria.

```
<html>
<head><title>Test addslashes() function</title></head>
<body>
<h3>Test form </h3><br>
<form name="formtest"  method= "GET"  action="">
  INPUT
      <input name="txtinput"  type="text" size="30"><br>
      <input name="btnSubmit"  type="submit"  value="addslashes">
      <input name="btnReset"  type="reset"  value="Reset">
</form>
<hr>
<?php
    $in=$_GET["txtinput"];
    if ($in==NULL){
        echo "Pls enter a string with quote";
    }
    else{
        $sql="SELECT * FROM employee WHERE firstname =
'".addslashes($in)."'";
        echo $sql;
    }
?>

</body>
</html>
```



Example for **`addslashes`** function

`stripslashes`– this is the inverse of `addslashes` where you remove backslashes from a string.

EXAMPLE 2

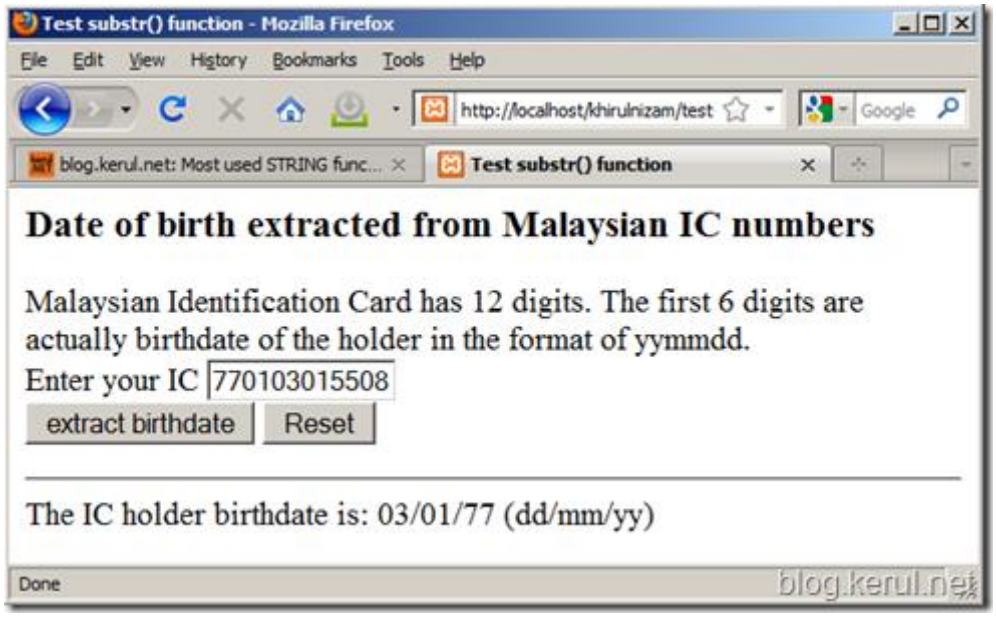
`strlen` — Get string length

`substr` — Return part of a string

In the example, we are trying to extract birthdate information from a Malaysian IC numbers.

```
<html>
<head><title>Test substr() function</title></head>
<body>
<h3>Date of birth extracted from Malaysian IC numbers </h3>
Malaysian Identification Card has 12 digits.
The first 6 digits are actually birthdate of the holder
in the format of yymmdd.
<br>
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text" size="12" maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="extract birthdate">
      <input name="btnReset"  type="reset"  value="Reset">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        // $ic!=NULL - to check there's actually input in the textbox
        // ctype_digit($ic) - all characters are number
        // strlen($ic)==12 - to check there are 12 digits
        $year=substr($ic,0,2);//extract digit 1 and 2
        $month=substr($ic,2,2);//extract digit 3 and 4
```

```
        $day=substr($ic,4,2);//extract digit 5 and 6

        echo "The IC holder birthdate is: ";
        echo "$day/$month/$year (dd/mm/yy)";// date format dd/mm/yy
    }
    else{
        echo "Enter an IC number (must be in 12 digits format)";
    }
?>

</body>
</html>
```



Examples for **`strlen`** and **`substr`** function

EXAMPLE 3

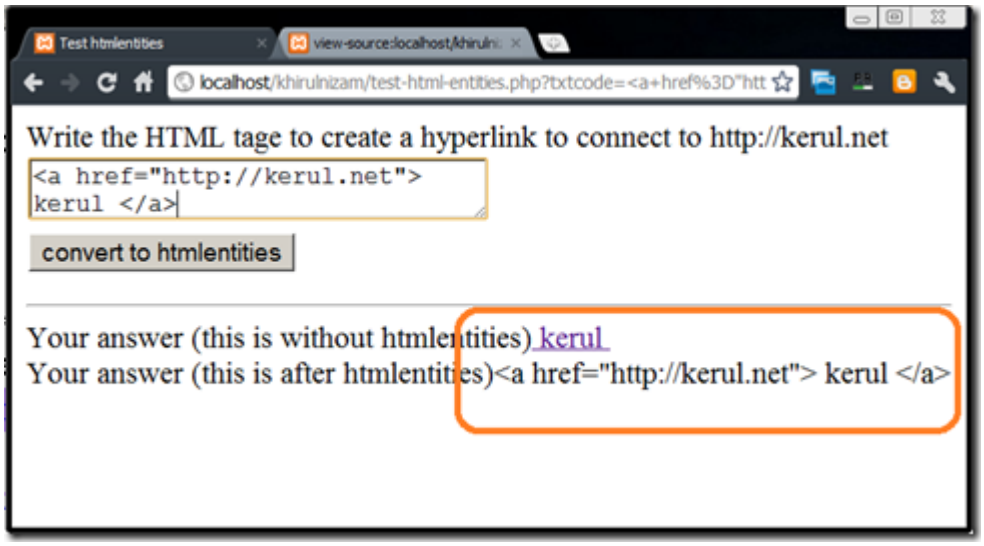
`htmlentities` — Convert all applicable characters to HTML entities

In the example, we are trying to print a HTML tag into the HTML page.

```
<html>
<head>
<title>Test htmlentities</title>
</head>

<body>
<form name="formtest"  method= "GET"  action="">
  Write the HTML tage to create a hyperlink to connect to
http://kerul.net<br>
      <textarea name="txtcode"></textarea>
      <br>
      <input name="btnSubmit" type="submit" value="convert to
htmlentities">
</form>
```

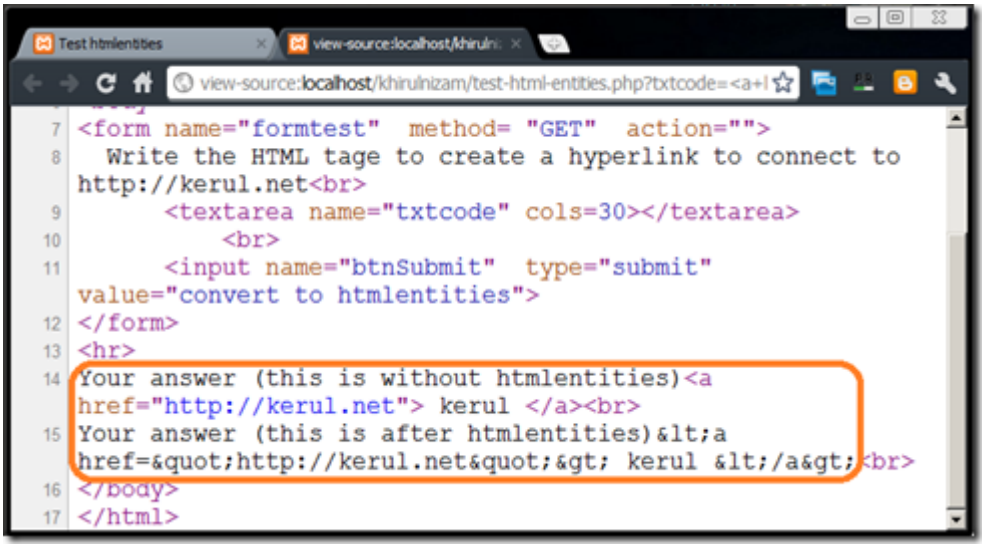
```
<hr>
<?php
    $code=$_GET["txtcode"];
    //echo " $ic <br>";
    if ($code!=NULL ){

        echo "Your answer (this is without htmlentities)";
        echo $code;
        echo "<br>\n";
        echo "Your answer (this is after htmlentities)";
        echo htmlentities($code);
        echo "<br>\n";


    }else{
        echo "Please provide the code";
    }
?>
</body>
</html>
```

The output;



Example for **`htmlentities`** function

See the HTML tag for the page,



HTML tag for the page

html_entity_decode — Convert all HTML entities to their applicable characters

EXAMPLE 4 – String to number (big integer)

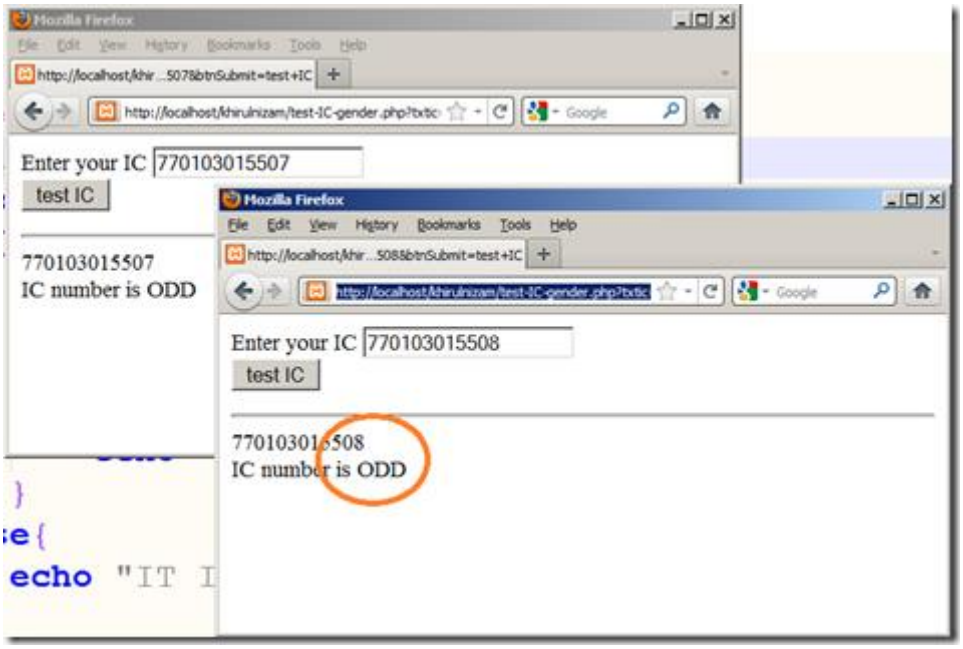
FYI, the maximum range for an integer value in PHP variable is 2147483647. I got this from the code;

```
<?php
echo PHP_INT_MAX; //to get the maximum integer value PHP variable can hold
?>
```

Guest what happen if I run this code to check Malaysian IC number whether it's ODD or EVEN (input comes from a textbox – means it in STRING). Malaysian IC number consists of 12 digits number.

```
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text"maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="test IC">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    echo " $ic <br>";
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        if ($ic%2==0){//IC is even
            echo "IC number is EVEN";
        }
```

```
        else{
            echo "IC number is ODD";
        }
    }else{
        echo "IT IS NOT A VALID IC NUMBER";
    }
?>
```



It will always tell you the number is ODD…

Why? Not quite sure why. Yet the solution is simple. Add 0 to the $ic as in the following code, and problem solved. They said it something to do with changing the integer variable into unsigned..

```
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text"maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="test IC">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    $ic=$ic+0;
    echo " $ic <br>";
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        if ($ic%2==0){//IC is even
            echo "IC number is EVEN";
        }
        else{
            echo "IC number is ODD";
        }
    }else{
        echo "IT IS NOT A VALID IC NUMBER";
    } ?>
```

EXAMPLE 5

`array_count_values()` - returns an array using the values of array as keys and their frequency in array as values.

```
<?php
$array = array(1, "hello", 1, "php", "hello");
print_r(array_count_values($array));
?>
```

The output for above example will

```
Array
(
    [1] => 2
    [hello] => 2
    [php] => 1
)
```

The following is example for word

EXAMPLE 6

`mb-strlen()` - When counting the length of an UTF-8 string.

```
if (mb_strlen($name, 'UTF-8') < 3) {
    $error .= 'Name is required. Minimum of 3 characters required';
}
```

EXAMPLE 7

`mb_strtok()` - String tokenizer for Multibyte

This is a simple function to implement some kind of `mb_strtok()` in PHP. As maybe you all are aware the `mb_strtok` function does not available for multibyte string (aka Unicode string). So this is my attempt to solve the problem. Anyway, there are bugs where the program halt if the input text is too long (how long? not sure yet). Maybe you could improve to provide better result.

String tokenization

The PHP code *mb_strtok.php* ;

```
<html>
    <head>
        <title>String token for MB_STRING</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    </head>
    <body>
        <h1>String token for MB_STRING</h1>
       <h2><a href="http://kerul.net">kerul.net</a></h2>

<form method="GET" ACTION="">
Input text <br>
<textarea name="txtinput" cols=30 rows=10></textarea>
<br>
<input type="submit" >
</form>

<?php
$in=$_GET["txtinput"];
$inputlen=mb_strlen($in, 'UTF-8');
echo ("Input length: $inputlen characters. <br>\n");

$tokens=mb_strtok(" /n/t?\'.", $in);
echo ("List of TOKENS<br>\n");
//echo $tokens;
for($i=0; $i<count($tokens); $i++){
    echo ("[$i] -> ".$tokens[$i] ." <br> \n");
```

```
}

function mb_strtok($delimiters, $str=NULL)
{
    static $pos = 0; // Keep track of the position on the string for each
subsequent call.
    static $string = "";
    static $listtoken=array();
    // If a new string is passed, reset the static parameters.
    if($str!=NULL)
    {
        $pos = 0;
        $string = $str;
    }

    // Initialize the token.
    $token = "";

    while ($pos < mb_strlen($string,'UTF-8'))//loop till end of input
string
    {

        $char = mb_substr($string, $pos, 1);//fetch one character, pos =
char position
        $pos++;
        //echo ("Char at $pos => $char <br>\n");//trace character at
position

        if(mb_strpos($delimiters, $char)===FALSE)//if character is not
delimeter
        {
            $token .= $char;//put character in the token node
        }
        else
        {
           //if arrive at delimeter, push token to listtoken
           array_push($listtoken, $token);
           $token="";//clear the token node
        }
    }
    // return the list of tokens
    if ($listtoken!="")
    {
        return $listtoken;
    }
    else
    {
        return false;
    }
}
?>
  </body>
</html>
```

There is another one, this time the separator (.,;:) will be stored in the list of token (listtoken).

```
<html>
    <head>
        <title>String token for MB_STRING</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    </head>
    <body>
        <h1>String token for MB_STRING</h1>
       <h2><a href="http://kerul.net">kerul.net</a></h2>

    <form method="GET" ACTION="">
    Input text <br>
    <textarea name="txtinput" cols=30 rows=10></textarea>
    <br>
    <input type="submit" >
    </form>

    <?php
    $in=$_GET["txtinput"];
    $inputlen=mb_strlen($in, 'UTF-8');
    echo ("Input length: $inputlen characters. <br>\n");

    $tokens=mb_strtok(" /n/t/f", $in);//delimeter by whitespace only
    echo ("List of TOKENS<br>\n");
    //echo $tokens;
    for($i=0; $i<count($tokens); $i++){
        echo ("[$i] -> ".$tokens[$i] ." <br> \n");
    }

    function mb_strtok($delimiters, $str=NULL)
    {
        static $pos = 0; // Keep track of the position on the string for
each subsequent call.
        static $string = "";
        static $listtoken=array();
        // If a new string is passed, reset the static parameters.
        if($str!=NULL)
        {
            $pos = 0;
            $string = $str;
        }

        // Initialize the token.
        $token = "";

        while ($pos < mb_strlen($string,'UTF-8'))//loop till end of input
string
        {

            $char = mb_substr($string, $pos, 1, 'UTF-8');//fetch one
character, pos = char position

            echo ("Char at $pos => $char <br>\n");//trace character at
position


            if(mb_strpos($delimiters, $char)===FALSE)//if character is not
delimeter
```

```
            {
                if($char=="." || $char==";"||$char==":"||$char==","){
                    echo "Token detected $token <br>\n";
                    array_push($listtoken, $char);
               //$token="";//clear the token node
                }else{
                    $token .= $char;//put character in the token node
                }
            }
            else
            {
               //if arrive at delimeter, push token to listtoken
               echo "Token detected $token <br>\n";
               array_push($listtoken, $token);
               $token="";//clear the token node
            }
            $pos++;
        }
        return $listtoken;
        // return the list of tokens
        if ($listtoken!="")
        {
            return $listtoken;
        }
        else
        {
            return false;
        }

    }
    ?>
      </body>
    </html>
```

Other string functions

`similar_text` — Calculate the similarity between two strings

`md5` — Calculate the md5 hash of a string

`money_format` — Formats a number as a currency string

`echo` — Output one or more strings

`printf` — Output a formatted string

`str_word_count` — Return information about words used in a string

`strtok` — Tokenize string

`trim` — Strip *whitespaces (or other characters) from the beginning and end of a string

`ucwords` — Uppercase the first character of each word in a string

*Whitespaces are hidden characters such as space, tab and newline.

EXERCISE – String Functions

Exercise 1: Accept a date in dd/mm/yyyy format and convert into ISO date (yyyy-mm-dd). Make sure the input format is correct before converting to another format.

Answer:

```
<form method="get" action="">
Date <input type="text" name="txtdate">
    format (dd/mm/yyyy)<br>
    <input type="submit">
</form>
<?php
$date=$_GET['txtdate'];
if (strlen($date)==10){

    $day=substr($date, 0, 2);
    $month=substr($date, 3, 2);
    $year=substr($date, 6, 4);

    if(ctype_digit($day)&&ctype_digit($month)&&ctype_digit($year)){
        //display
        echo "Input date: $date<br>";
        echo "ISO date: $year-$month-$day";
    }
    else{
        echo "Date format not valid <br>";
    }
}
else{
    echo "Date format not valid <br>";
}
?>
```

Exercise 2: Accept a date in ISO format (yyyy-mm-dd) and convert to dd MONTH yyyy. Make sure the input format is correct before converting to another format.
Eg: Accept 2011-07-14 as the input, and output 14 JULY 2011

Answer:

```
<form method="GET" action="">
Task: Convert ISO date (yyyy-mm-dd) into dd NAME_OF_MONTH yyyy <br>
Input ISO date<input type="text" maxlength=10 name="txtisodate">
(format: yyyy-mm-dd)<br>
<input type="submit" value="Convert">

</form>
<hr>
Output
<br>

<?php
$date=$_GET['txtisodate'];
if (strlen($date)==10){

    $year=substr($date, 0, 4);
    $month=substr($date, 5, 2);
    $day=substr($date, 8, 2);
```

```
    if(ctype_digit($day)&&ctype_digit($month)&&ctype_digit($year)){
        //display
        echo "Input ISO date: $date<br>";
        if($month=="01")
            $monthname="January";
        else if ($month=="01")
            $monthname="February";
        else if ($month=="02")
            $monthname="February";
        else if ($month=="03")
            $monthname="March";
        else if ($month=="04")
            $monthname="April";
        else if ($month=="05")
            $monthname="May";
        else if ($month=="06")
            $monthname="June";
        else if ($month=="07")
            $monthname="July";
        else if ($month=="08")
            $monthname="August";
        else if ($month=="09")
            $monthname="September";
        else if ($month=="10")
            $monthname="October";
        else if ($month=="11")
            $monthname="November";
        else if ($month=="12")
            $monthname="December";
        else
            $month="NOT_VALID_VALUE";
        echo "After conversion: $day $monthname $year";

    }
    else{
        echo "Date format not valid <br>";
    }
}
else{
    echo "Date format not valid <br>";
}
?>
```

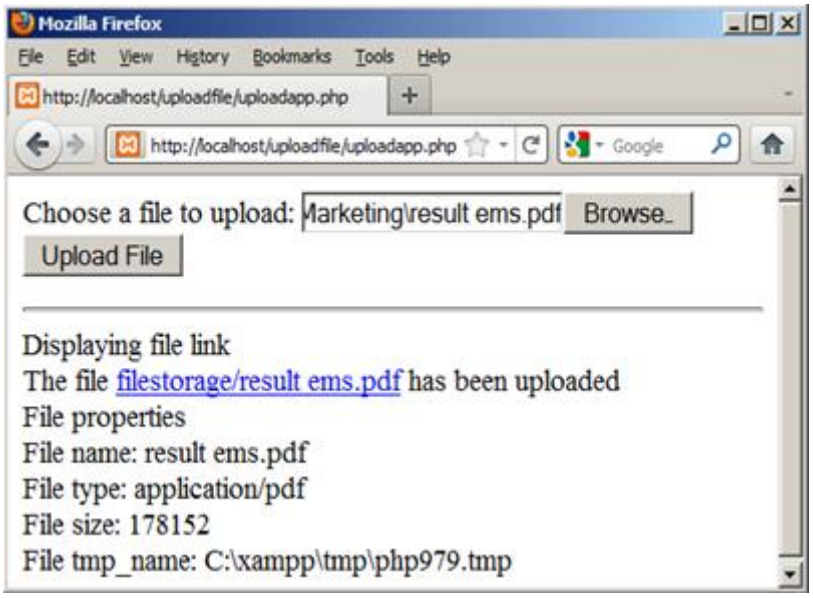
File Upload

Upload file to the server
A very useful aspect of PHP is its ability to manage file uploads to your server. Allowing users to upload a file to your server opens a whole can of worms, so please be careful when enabling file uploads.

EXERCISE 1: UPLOADING A FILE WITHOUT RESTRICTION

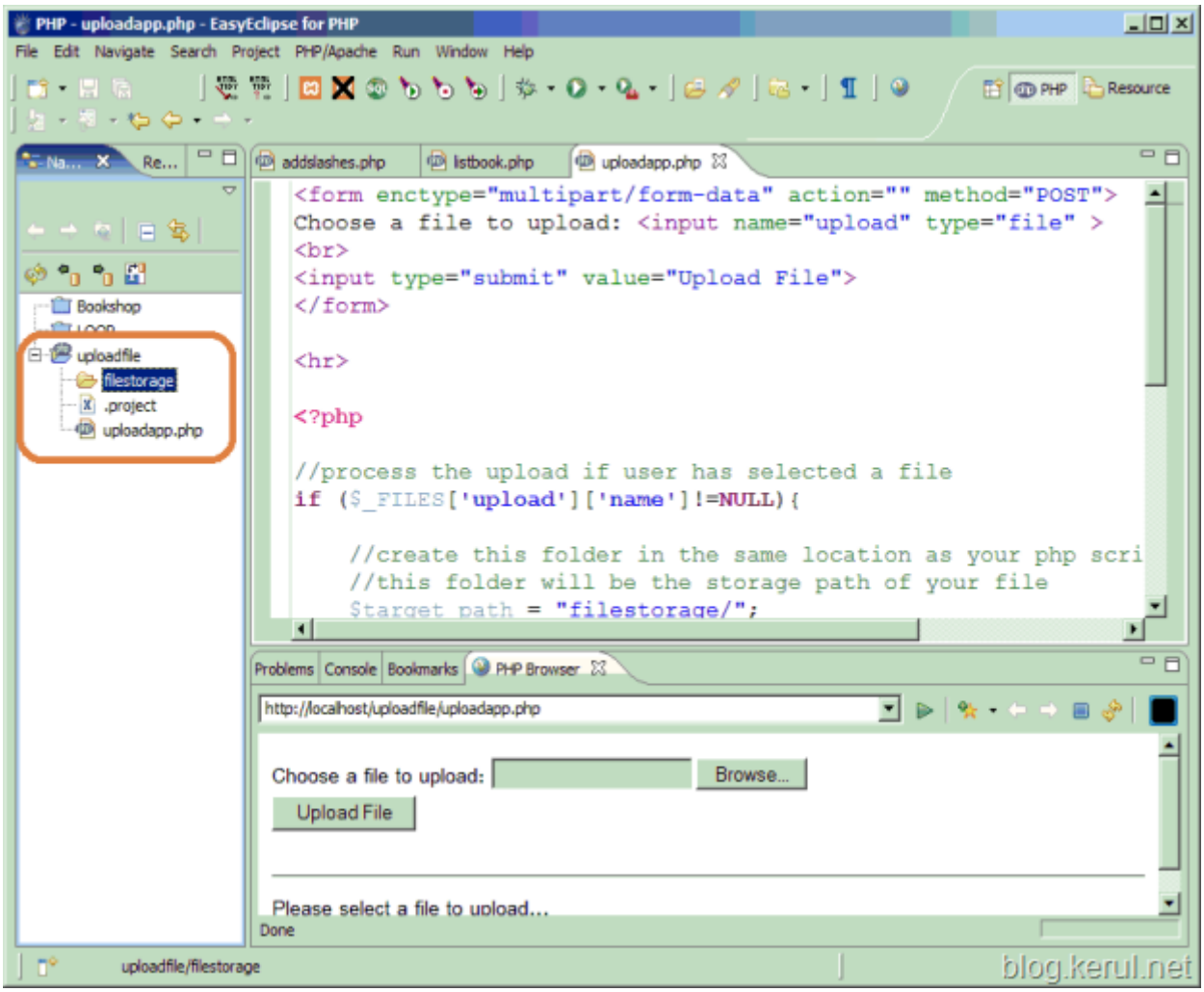
WARNING: This is a very simple file uploading exercise, intentionally created for training purposes to expose students the concept of file uploading processes. DO NOT use this in production, because it has no security feature…

What it does: This application will have the capability to upload any selected file into the web server.
Output: This application will look like this in Firefox;



Upload selected file into the web server

For the purpose of this exercise, you need to create another folder in the project folder. This folder will serve as the storage for whatever files the user uploaded.



Create new folder in the project folder

The PHP code: It is a single file PHP script.

```
<form enctype="multipart/form-data" action="" method="POST">
Choose a file to upload: <input name="upload" type="file" >
<br>
<input type="submit" value="Upload File">
</form>
<hr>

<?php
//process the upload if user has selected a file
if ($_FILES['upload']['name']!=NULL){

    //create this folder in the same location as your php script
    //this folder will be the storage path of your file
    $target_path = "filestorage/";
    $target_path = $target_path . $_FILES['upload']['name'];

    if(move_uploaded_file($_FILES['upload']['tmp_name'], $target_path)) {
        //Display file link
        echo "Displaying file link<br>";
        echo "The file <a href='$target_path'>$target_path</a> " .
             "has been uploaded<br>";
        echo "File properties <br>";
        echo "File name: ". $_FILES['upload']['name']."<br>";
        echo "File type: ". $_FILES['upload']['type']."<br>";
        echo "File size: ". $_FILES['upload']['size']."<br>";
        echo "File tmp_name: ". $_FILES['upload']['tmp_name']."<br>";
    }
    else{
        //uploading error
        echo "There was an error uploading the file, " .
             "please try again!<br>";
        echo "File error code: ". $_FILES['upload']['error'];
    }
}
else{//no file selected
    echo "Please select a file to upload...";
}

?>
```

Explanation: elaboration of what's happening in the PHP code.

The form elements;

- **`enctype="multipart/form-data`**" - Necessary for our to-be-created PHP file to function properly (read more). It is used for submitting forms that contain files, non-ASCII data, and binary data.
- **`method="POST"`** - Informs the browser that we want to send information to the server using POST. It won't work if you set it to GET.
- **`input name="upload"`** - upload is how we will access the file in our PHP script.

The server processes;

When the PHP script is executed, the uploaded file exists in a temporary storage area on the server. If the file is not moved to a different location it will be destroyed! To save the file we need to make use of the **$_FILES** associative array.
The **$_FILES** array is where PHP stores all the information about files. There are two elements of this array that we will need to understand for this example.

- **upload** - *upload* is the reference we assigned in our HTML form. We will need this to tell the **$_FILES** array which file we want to play around with.

- **$_FILES['upload']['name']** - name contains the original path of the user uploaded file.

- **$_FILES['upload']['tmp_name']** - *tmp_name* contains the path to the temporary file that resides on the server. The file should exist on the server in a temporary directory with a temporary name.

- **if ($_FILES['upload']['name']!=NULL)** – means if there is no file selected by the user.

- **$target_path** – is the full path of the file saved in the server. By right it is supposed to be in *filestorage/your_file.ext*

- **move_uploaded_file($_FILES['upload']['tmp_name'],$target_path)** – a PHP function to ensure that the file designated by filename **($_FILES['upload']['tmp_name'])** is a valid upload file (meaning that it was uploaded via PHP's HTTP POST upload mechanism). If the file is valid, it will be moved to the filename given by destination (in **$target_path**). (read more –> http://php.net/move_uploaded_file)

**Credit to TiZag.com. Modified from the tutorial in http://www.tizag.com/phpT/fileupload.php
.

EXERCISE 2: UPLOADING A FILE WITH FILE TYPE RESTRICTIONS

This exercise will impose file type restrictions. It will only receive image files with *PNG, GIF or JPG format*. Before uploading the file to the server, check the file type. Display the image in the output.

```
<form enctype="multipart/form-data" action="" method="POST">
Choose an image to upload (PNG, JPG, GIF): <input name="upload" type="file"
>
<br>
<input type="submit" value="Upload Image">
</form>
<hr>

<?php
//process the upload if user has selected a file
if ($_FILES['upload']['name']!=NULL){

    //create this folder in the same location as your php script
    //this folder will be the storage path of your file
    $target_path = "filestorage/";
    $target_path = $target_path . $_FILES['upload']['name'];
    if ($_FILES['upload']['type']== 'image/x-png'
        || $_FILES['upload']['type']== 'image/png'
        || $_FILES['upload']['type']== 'image/jpeg'
        || $_FILES['upload']['type']== 'image/pjpeg'
        || $_FILES['upload']['type']== 'image/gif'){
    if(move_uploaded_file($_FILES['upload']['tmp_name'], $target_path)) {
        //Display file link
        echo "Displaying file link<br>";
        echo "The file <a href='$target_path'>$target_path</a> " .
             "has been uploaded<br>";
        echo "File properties <br>";
        echo "File name: ". $_FILES['upload']['name']."<br>";
        echo "File type: ". $_FILES['upload']['type']."<br>";
        echo "File size: ". $_FILES['upload']['size']."<br>";
        echo "File tmp_name: ". $_FILES['upload']['tmp_name']."<br>";
    }
    else{
        //uploading error
        echo "There was an error uploading the file, " .
             "please try again!<br>";
        echo "File error code: ". $_FILES['upload']['error'];
    }
    }
}
else{//no file selected
    echo "Please select a file to upload...";
}

?>
```

EXERCISE 3: UPLOADING A FILE WITH FILE TYPE AND SIZE RESTRICTIONS

Create a file uploading application that only receives a *PDF file* with *file size smaller* than *100KB*. Display the link of the transferred file.

```
<form enctype="multipart/form-data" action="" method="POST">
Choose a PDF file less than 100KB: <input name="upload" type="file" >
<br>
<input type="submit" value="Upload PDF">
</form>

<hr>

<?php

//process the upload if user has selected a file
if ($_FILES['upload']['name']!=NULL){

    //create this folder in the same location as your php script
    //this folder will be the storage path of your file
    $target_path = "filestorage/";

    if ($_FILES["upload"]["type"] == "application/pdf"
            && $_FILES["upload"]["size"] <=102400){
        $target_path = $target_path . $_FILES['upload']['name'];
        if(move_uploaded_file($_FILES['upload']['tmp_name'], $target_path))
{
            //Display file link
            echo "Displaying file link<br>";
            echo "The file <a href='$target_path'>$target_path</a> has been
uploaded";
        }
        else{
            //uploading error
            echo "There was an error uploading the file, please try
again!<br>";
            echo "File type: ".$_FILES["upload"]["type"]."<br>";
            echo "File size: ".$_FILES["upload"]["size"]."<br>";
        }
    }
    else {
        echo "Only PDF file <100KB is allowed. Try again<br>";
        echo "File type: ".$_FILES["upload"]["type"]."<br>";
        echo "File size: ".$_FILES["upload"]["size"]."<br>";
    }
}
else{//no file selected
    echo "Please select an image file to upload...";
}
?>
```