Unit 6G4Z2101: Introduction to Web Design and Development

Laboratory Instructions - the PHP tasks

**Important: For the final unit tests you are permitted to take in these instructions AND any notes you make on this laboratory sheet. You MUST also have the most up to date version of your laboratory attempts available at the final tests.**

1. **Learning materials**

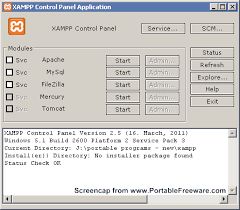
The recommended text for this part of the unit is:   
  
"Learning PHP, MySQL and Javascript: A step by step guide to creating dynamic websites" by Robin Nixon and published by O'Reilly.   
  
  
PHP, MySQL and Apache are open source projects and as such there is a lot of free material and support available on the internet. Key websites are:   
  
http://www.php.net   
  
and   
  
http://dev.mysql.com/   
  
These are definitive websites as both PHP and Mysql are open source.   
  
  
**During the laboratories, you will create two web pages:**  
  
1] The first web page will allow you demonstrate that you can operate an "AMP" based development environment and create a webpage based on PHP, HTML and Javascript.   
  
2] A second web page will require you to build a MYSQL based website. Instructions for this page will be presented half way through the term. You are expected to have completed the first web page to at least “rectangle 8” by halfway through the term.  
  
Before you start creating dynamic webpages you will need to become familiar with the development environment.

1. **Setting up the development environment**

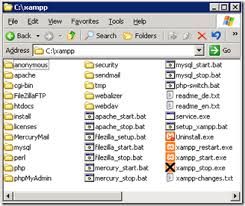
You will find a "WAMP" (Windows Apache, Mysql, PHP) system available. Typical examples are Mowes, Xammp, WAMMP and MAMP.

There are some slight difference between “WAMPS” but generally they all have a control panel and directory structure as indicated below

**The control panel of the Xammp WAMP**



**The directory structure of a typical WAMP**



Start the server and MYSQL processes from the control panel. Then type the following into the URL bar of a browser:

http://localhost

You should find that the above URL calls up the default WAMP web page in the browser.

The default localhost webpage for the WAMP as displayed in a browser

1. **Testing that your development environments works**

Copy and Paste Place the code below (Generate a md5 hash of a random number) in the correct directory in your AMP (this can vary between systems: directories named WWW, public\_HTML, or htdocs are common).

Open this script in a text editor and then change the "seed" for a random number from 1021 to a new number, 1234. Once you have done this successfully, your browser will display a "hash" code.

<?php

// Generate an md5of a random number in order to test your WAMP and introduce you to seeded random //numbers

// All you have to do is find the code by setting the seed number to 1234

// set the seed

$a = 1021;

srand($a);

//convert the random number into a string then convert into an md5 hash

$b = rand();

$c = strval($b); // Note the md5 function only workd on strings not numbers.

$d = md5($c);

if ($d == "3de36aabe0a47900e74f4a55a03db1fc"){

$d= "You need to change the seed number!";

Echo " ", $d;

}

else {

Echo " ", "SUCCESS!!. This is the code that you enter into Moodle ONCE you have altered the seed number::"," ", $d;

echo "<br>";

}

?>

**One further test of your development environment (needed for the MYSQL laboratories)**

Open up "PHPMyadmin" in your browser. This tests the full capability of the system.

1. **The PHP,HTML, Javascript web page.**

All PHP,HTML and Javascript should be contained in a single file called

"YourNameHTML.php"..

e.g

“FredBlogsHTML,php”

At the end of the unit you will be required to upload this file – hence the file name structure.

**Always store a copy of your most recent work on your student area. In this way you have access to it and you will need access to it during the final tests.**

You will need to be familiar with a number of PHP, HTML commands. These are listed below. With a view to preparing for these sessions you need to go to PHP.Net and make notes in the area provided as to what each command does.

1. **Your Notes – you need these for the final unit test, make them good.**

|  |  |
| --- | --- |
| **Commands** | Notes on the use of the command (you complete – I did <table> for you) |
| **HTML:** |  |
| <Table> | Creates a table of a certain border style. Note the table in its basic form extends depending on what is in each “rectangle”. The table has to be populated with <td> (table data) and <tr> (table row tags to define the number of rows and elements. |
| < Form action=”PhpFileThaIisRun.php” method=”post”> | [Note: how is this related to the php command “$\_POST”?] |
| <Input type=”submit” name=”SomeStringYouMadeUp”> | [Note: how is this related to the php command “$\_POST”?] |
| **PHP:** |  |
| echo |  |
| rand |  |
| srand |  |
| while |  |
| str\_rot13 |  |
| $\_POST[“name”] | [Note: how is this related to the html commands Form and Input?)] |
| Other commands found useful |  |
| Other commands found useful |  |
| Other commands found useful |  |
| Other commands found useful |  |

1. **Setting up the PHP Laboratory work**

Copy and paste the code below into your working WAMP area (usually /httpdocs). Then serve the file (go to a browser then put <http://localhost/yourfilename.php> into the address bar).

<?php  
// The HTML starts here

echo " <table border='1'>

<tr>

<td>Rectangle 1 Your Name and registration number</td>

<td>Rectangle 2</td>

</tr>

<tr>

<td>Rectangle 3</td>

<td>Rectangle 4</td>

</tr><td>Rectangle 5</td>

<td>Rectangle 6</td>

</tr>

<tr>

<td>Rectangle 7</td>

<td>Rectangle 8</td>

</tr>

<tr>

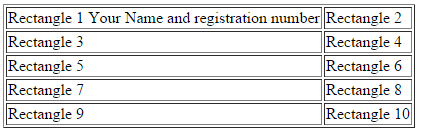
<td>Rectangle 9</td>

<td>Rectangle 10</td>

</tr>

</table> ";  
?>

You should see:



1. **Laboratory Task – this should take five weeks. These all need to be working for the final test**

Create a PHP, HTML, Javascript web page structured on table structure given above which is served from an "WAMP" development system. The following should appear in each rectangle.

Each rectangle must be labelled as "Rectangle 4" etc.

|  |
| --- |
| Rectangle 1: Your name and student registration number and CURRENT date. |
| Rectangle 2: The result of adding the three numbers 3+4+5. DON’T just “echo” 12 to screen – do the calculation. |
| Rectangle 3: The result of using the seeded random number function "srand" using the seed "3034" |
| Rectangle 4: The result of using a conditional loop to display all the even numbers between 30 and 40. |
| Rectangle 5: An "input field" that allows you to input a four digit integer number or a four letter string and store it in a variable "$x". A user will put the numbers/letters in a box and click on a button labelled "submit" in order to enter the number/letter into the variable. |
| Rectangle 6: An output that indicates if a number "$x" input by the user is even or odd. |
| Rectangle 7: The result of using $x in as a seed for the "srand" function. In this part, you may be asked to input a four digit number which then displays a new random number. |
| Rectangle 8: The result of encoding $x (letters only) in "rot13" using in built PHP functions. |
| Rectangle 9: The result of encoding $x in "rot13" using Javascript (find suitable code on the internet - this is not part of the course devoted to teaching javascript). |
| Rectangle 10: An md5 password being cracked by using the brute force method.  The output displayed here is the result of manually putting the required md5 hash into the raw PHP code and should be of the form “letter’m’ and four digit number” e.g “m5463”.  **The scenario for Rectangle 10**  You are the website administrator for a website that requires a low level password for entry. The previous owner of the website forget to code in a password recovery feature and as such you now have to recover the passwords for a number of users of the website. You do have access to the website database BUT you are NOT allowed to manually change the passwords for users in the database as those passwords are heavily used in defining what permissions a user of the website has. e.g. Moderators for the website forum all have a password beginning with the lower case letter "m". ALL letters are lowercase. Passwords were allocated to all users and the method of choice was to create a password, which was built on two parts - a lowercase letter "m" and a number between 1 and 10000. An md5 hash was then created to form a 32 digit string. e.g. the user with password "m1234" is represented in the database as the md5 hash:  77c12394ef7d4f23a8fa07d87309afd9  The simplest thing to do is to write some PHP code that uses a while loop to increment through the numbers 0 to 10000, concatenate that number with the "m" and then compare the md5 hash to that you recovered (by looking at it using PhpMyAdmin) from the database. If the two hashes are the same then you have "cracked" the password. This technique is known as a "brute force" method of cracking passwords. |

**All PHP,HTML and Javascript should be contained in a single file called**

**"YourNameHTML.php". e.g “FredBlogsHTML,php”**

**At the end of the unit you will be required to upload this file – hence the file name structure.**

**Always store a copy of your most recent work on your student area. In this way you have access to it and you will need access to it during the final tests.**

**8. Additional resources**

There are a number of good quality videos available on the internet but many of them focus on the higher level application of PHP. CodeIgnite and the Zend framework tutorials are common and well worth watching once you've finished this unit.