Ahmadu Bello University, Zaria

Department of Mathematics

**2009/2010 First Semester Examination**

**COSC 405: Web Application Engineering II**

**Date**: May, 2010 **Time Allowed**: 120 Minutes

**Instructions**:

1. Attempt ANY FOUR questions.
2. Write all your answers in the spaces provided on this Question Paper.

**Student’s Registration Number**:………………………………………………………………… **Signature**:……………………………………..

**Date of Examination**:…………………………………………………………………………………**Time**:………………………………………………

**Scores**:

|  |  |  |
| --- | --- | --- |
| **Question** | **Maximum Scores** | **Scores Obtained** |
| 1 | 20 |  |
| 2 | 20 |  |
| 3 | 20 |  |
| 4 | 20 |  |
| 5 | 20 |  |
| 6 | 20 |  |
| **Total** | **80** |  |

1. Answer both the following questions:
	1. (**10 marks**). Compare and contrast the use of PHP and the use of Java applets for accessing a database server.
	2. (**10 marks**). Identify three different ways of accessing form variables from other PHP files. Explain relative merit of each over the others.
2. Answer both the following questions.
	1. (**14 marks**). Show the HTML-code resulting from a call to the PHP-function my\_universities, presented below, with the array $univs given as argument

|  |
| --- |
| $univs = array (array ("KASU", 2), array ("ABU", 7),array("UMYU",3));function my\_universities ($list) { $res = "<table width=50%>"; for ($i = 0; $i < count($list); $i++) { $name = $list [$i] [0]; $count = $list [$i] [1]; $n = $count; $bar = ""; while ($n > 0) { $bar = $bar . "\*"; $n = $n - 1; } $res = $res . "<tr><td width=25%>$name</td> <td width=25%>$count: $bar</td> </tr>"; } return "$res.</table>";} |

 **Answer:**

* 1. (**6 marks**). Write a PHP function max\_rating that takes as argument an array of university rankings and returns the maximum ranking number.

Hint: calling the function with the following array should return the value 7:

array (array ("KASU", 4), array ("ABU", 7),array("UMYU",5))

It can be advantageous (but not required) to use the foreach command to traverse the array.

**Answer:**

1. (**20 marks**). Study the following PHP code carefully and write down its output.

|  |
| --- |
| <?php $responses = array(1,2,6,4,8,5,9,7,8,10, 1,6,3,8,6,10,3,8,2,7, 6,5,7,6,8,6,7,5, 6,6, 5,6,7,5,6,4,8,6,8,10); for ( $i= 0; $i <= count($responses); $i=$i+1 ) { $result[$responses[$i]] += 1; } $answer = "<table border='border'><tr>"; for ( $t=1 ; $t <= 10 ; $t=$t+1 ) { $datum = $result[$t]; $answer = $answer."<td>$datum</td>"; } echo "$answer.'</tr></table>'";?> |

**Answer:**

1. Answer both the following questions.
	1. (**10 marks**). A *well-formed time* consists of two parts separated by a colon (:). The first part should be a number between 0 and 23 (both numbers inclusive). The second part should be a number between 0 and 59 (both numbers inclusive). For the first part, a prefixed zero (0) is optional if the number is between 0 and 9 (both numbers inclusive). For the second part, a prefixed zero (0) is mandatory if the number is between 0 and 9 (both numbers inclusive).

Examples of well-formed times are 23:59, 5:09, and 03:00. Examples of strings that are not well-formed times are 23:2 and 24:00.

Construct a regular expression that is matchable only by well-formed times.

**Answer:**

* 1. (**10 marks**). Construct a PHP function validate\_time for checking whether a time is well-formed, as described in (a) above. If the argument passed to the function is a well-formed time, the function should return immediately. Otherwise, the function should display an error message in the user’s browser and stop the program by calling the built-in PHP function exit.

Use the PHP function ereg to check that the password is well-formed.

**Answer:**

1. Answer both the following questions
	1. (**10 marks**). Identify and write short notes on two forms of vulnerabilities that can occur in the database layer of an application.
	2. Consider the following MySQL database table called Expense:

|  |  |  |  |
| --- | --- | --- | --- |
| **kind** | **dept** | **year** | **value** |
| Salary | Acquisition | 2001 | 490,000 |
| Salary | Sales | 2002 | 1,500,000 |
| Salary | Acquisition | 2002 | 500,000 |
| Coffee | Acquisition | 2003 | 800 |
| Coffee | Sales | 2003 | 300 |
| Salary | Sales | 2003 | 1,600,000 |
| Salary | Acquisition | 2003 | 510,00 |

Write MySQL queries to find

1. (**5 marks**). for each year, the total expenses of each kind.
2. (**5 marks**). the total expenses grouped by department, kind, and year for the years 2002-2003 in which the sum of the expenses is less than 1,000,000.
3. Consider the following site map for an online registration Web application.



Unlabeled arrows represent hyperlinks to a new HTML page, possibly generated by a PHP script. Labeled arrows represent transactions that update the database by running a PHP script.

Answer the following questions:

1. (**10 marks**). The following is the code for the file maillist\_add.php which adds a user to a maillist table. Complete the missing code by writing in the empty rectangles following the comments.

|  |  |  |  |
| --- | --- | --- | --- |
| <?php // Establish database connection $db = mysql\_connect("localhost", "root"); mysql\_select\_db("mail", $db) or die (mysql\_error($db)); // read new member’s Name and Email from the maillist\_add.html form

|  |
| --- |
|   |

 //Add the new member’s Name and Email into the maillist table of the database

|  |
| --- |
|  |

 // Jump to maillist.php

|  |
| --- |
|  |

?> |

1. (**10 marks**). The following is the code for the file maillist.php that displays the updated list of members in the maillist.

|  |  |  |
| --- | --- | --- |
| <html><title>Members</title><body><h2>Members</h2><ul><?php  // Establish database connection $db = mysql\_connect("localhost","root"); mysql\_select\_db('mail', $db) or die (mysql\_error($db)); // Extract rows from the table

|  |
| --- |
|  |

  // Iterate through the rows and display each row as a list item as shown in the figure above.

|  |
| --- |
|  |

 ?></ul><p><a href="maillist\_add.html">Add Yourself</a></body></html> |