UNIVERSITY OF MAURITIUS

FACULTY OF ENGINEERING

SECOND SEMESTER/YEARLY EXAMINATIONS

MAY 2009

PROGRAMME	**BSc (Hons) Computer Science and Engineering** **Bsc (Hons) Information Systems** **BSc (Hons) Information and Communication Technologies**		
MODULE NAME	**Web Technologies**		
DATE	**Thursday** **14 May 2009**	**MODULE CODE**	**CSE 2003Y(3)**
TIME	**13.30 – 16.30 hrs**	**DURATION**	**3 hrs**
NO. OF QUESTIONS SET	**6**	**NO. OF QUESTIONS TO BE ATTEMPTED**	**5**

INSTRUCTIONS TO CANDIDATES

This paper consists of two sections. Section A and Section B.

Answer All Questions from Section A and any one question from Section B.

All questions refer to the Feedback case study discussed in class.

SECTION A

All questions refer to the Feedback case study as described:

Consider a feedback application that is being developed to collect feedback from students about different modules. The following schema has been defined for the feedback database, implemented in MySQL.

classsizes (classsize, classsizedesc)
feedbacks (email, modulecode, moduleyear, programme, classsize, delivery, labs, Othercomments, moderation)
modules (modulecode, moduledesc)
users (username, pass)

Notes:

- *classsize* and *modulecode* in the *feedbacks* table are foreign keys from the classsizes and *modules* tables respectively.
- The *moderation* field contains values 'a' for approved, 'p' for pending, and 'r' for rejected. The default value is 'p'.
- The values for the *classsizes* table are 'a', 'p' , 'f' , 'e' in the *classsize* field for 'adequate', 'poor', 'fair' and 'excellent' respectively in the *classsizedesc* field.
- *Delivery* and *labs* fields are boolean and indicate whether improvements are required in the delivery and labs (a value of 1 indicates that improvement is required)

Question 1

Students post feedback about different modules using the *Feedback11.php* page shown in Figure 1, the codes of which are shown in *listing 1*. One of the problems with the above design is that the students see all the modules that are available when they have to submit feedback, even those they did not register for. You propose a change in your table design and introduce the table *registers* as follows.

registers (username, module, moduleyear)

Listing 1

```
1   <?php
2    //connecting to database
3    include("db_connect.php");
4
5    $Rs = mysql_query("SELECT * FROM modules");
6
7    if (mysql_num_rows($Rs)<0)
8    {
9     mysql_close($con);
10    header( 'Location: error.html') ;
11   } //retrieving module information
12
13   $Rs1 = mysql_query("SELECT * FROM classsizes");
14   if (mysql_num_rows($Rs1)<0)
15   {
16    mysql_close($con);
17    header( 'Location: error.html') ;
```

```
 18      }
 19     ?>
 20
 21     <html>
 22     <head>
 23      <title>Feedback</title>
 24     </head>
 25
 26     <body onload='alert("Welcome");'>
 27     <script type="text/javascript" src=validations.js></script>
 28     <script type="text/javascript">
 29     <!--
 30     var curdate=new Date();
 31     document.write("<H1>Hello</H1>");
 32     document.write(curdate.getHours()+":"+curdate.getMinutes());
 33     function validateForm()
 34     {
 35      if (!validateNumeric(document.forms[0].txt_username.value,'Username'))
 36        return false;
 37      if (!validateBlank(document.forms[0].txt_password.value,'Password'))
 38        return false;
 39      return true;
 40     }
 41
 42     -->
 43     </script>
 44      <form id=frm_feedback action="feedback_pro9.php" method="post" onsubmit="return validateForm()">
 45      <table cellspacing=10 >
 46       <tr>
 47        <td>Username</td>
 48        <td><input type=text name="txt_username" maxlength=40 size=40 ></td>
 49        <td>Password</td>
 50        <td><input type=password name="txt_password" maxlength=40 size=40></td>
 51       </tr>
 52       <tr>
 53        <td>Email Address</td>
 54        <td><input type=text name="txt_email" maxlength=60 size=60 ></td>
 55        <td>Programme</td>
 56        <td><input type=text name="txt_programme" maxlength=60 size=60></td>
 57       </tr>
 58       <tr>
 59        <td>Module</td>
 60        <td><select name=txt_module>
 61            <? while ($rows = mysql_fetch_array($Rs)){?>
 62            <option value=<?echo $rows['modulecode'];?>><?echo $rows['moduledesc'];?></option>
 63            <?}
 64                    ?>
 65         </select></td>
 66        <td></td>
 67        <td></td>
 68       </tr>
 69       <tr>
 70        <td>Class size</td>
 71        <td colspan=3>
 72          <?while ($rows = mysql_fetch_array($Rs1)){?>
 73           <input type=radio name="txt_classsize" value='<?echo $rows["classsize"];?>'
 74           <?if ($rows["classsize"]=='-1') echo " checked";?>>
 75           <?
 76            echo $rows["classsizedesc"];
 77           }?>
 78        </td>
 79       </tr>
 80       <tr>
 81        <td>Improvements needed in</td>
 82        <td colspan=3><input type=checkbox name=txt_delivery value="ON">Delivery Mode
 83         <input type=checkbox name=txt_labs value="ON">Labs</td>
 84       </tr>
 85       <tr>
 86        <td>Other Comments</td>
 87        <td><textarea name="txt_others" cols=40 rows=3></textarea></td>
 88        <td></td>
 89        <td></td>
 90       </tr>
```

```
91        <tr>
92         <td><input type=submit value="Submit"></td>
93         <td><input type=reset value=reset></td>
94         <td></td>
95         <td></td>
96        </tr>
97       </table>
98       </body>
99      </html>
100     <?
101      mysql_close($con);
102     ?>
```

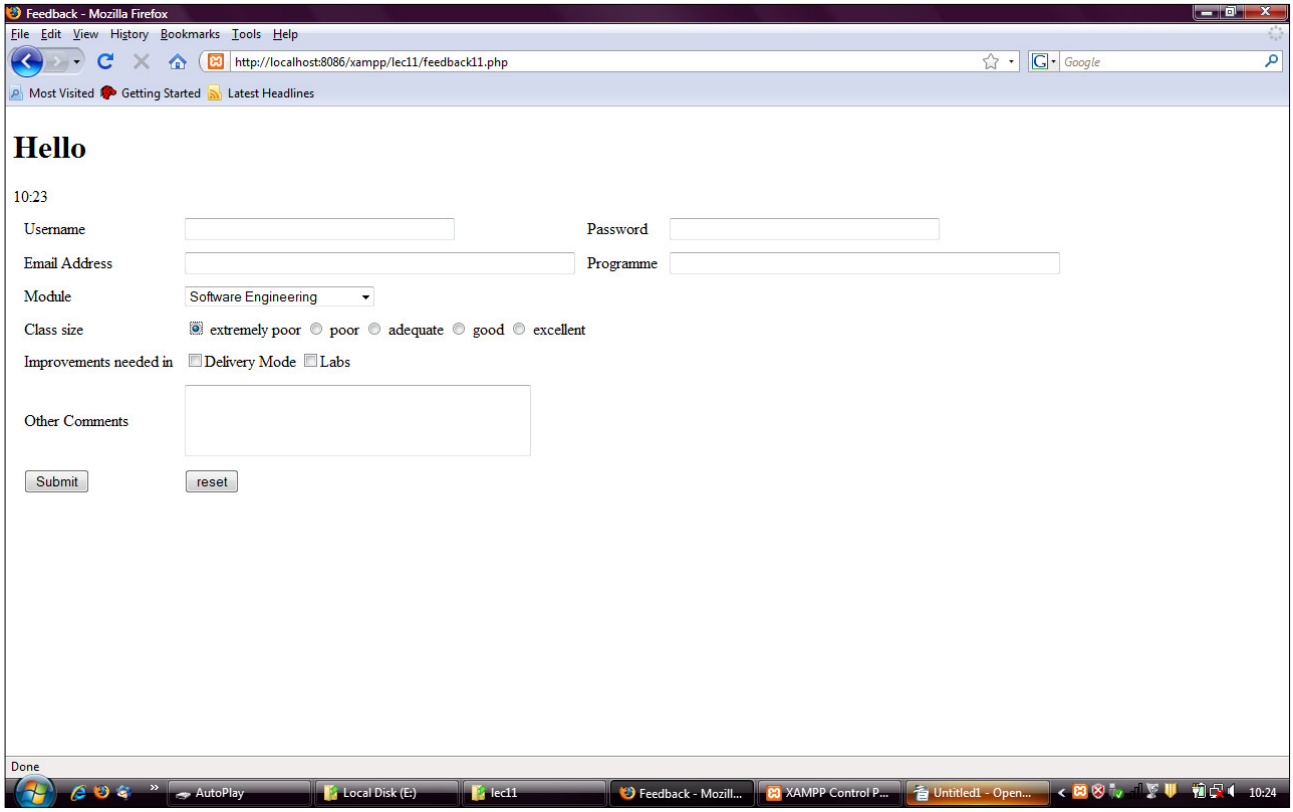

Figure 1: Feedback11.php

a) Change the codes for listing 1 such that only the modules that the user has taken for the current year are displayed in the drop down list for modules.

- You are not required to rewrite the entire set of codes. However, indicate where your changes will fit. [You may use the line numbers to indicate where your changes will fit]
- Assume that in the new proposed design, the student will log in first by submitting his username and password before being redirected to the above page and his username will be stored in a session variable called *username*.
- Make sure that your page is **logical** given that the user will already have logged in before being redirected to the feedback page.

[12 marks]

b) Identify another problem with the design above and propose how you will change the design.

[8 marks]

Question 2

Consider the page **feedback.php** illustrated in Figure 2 below:

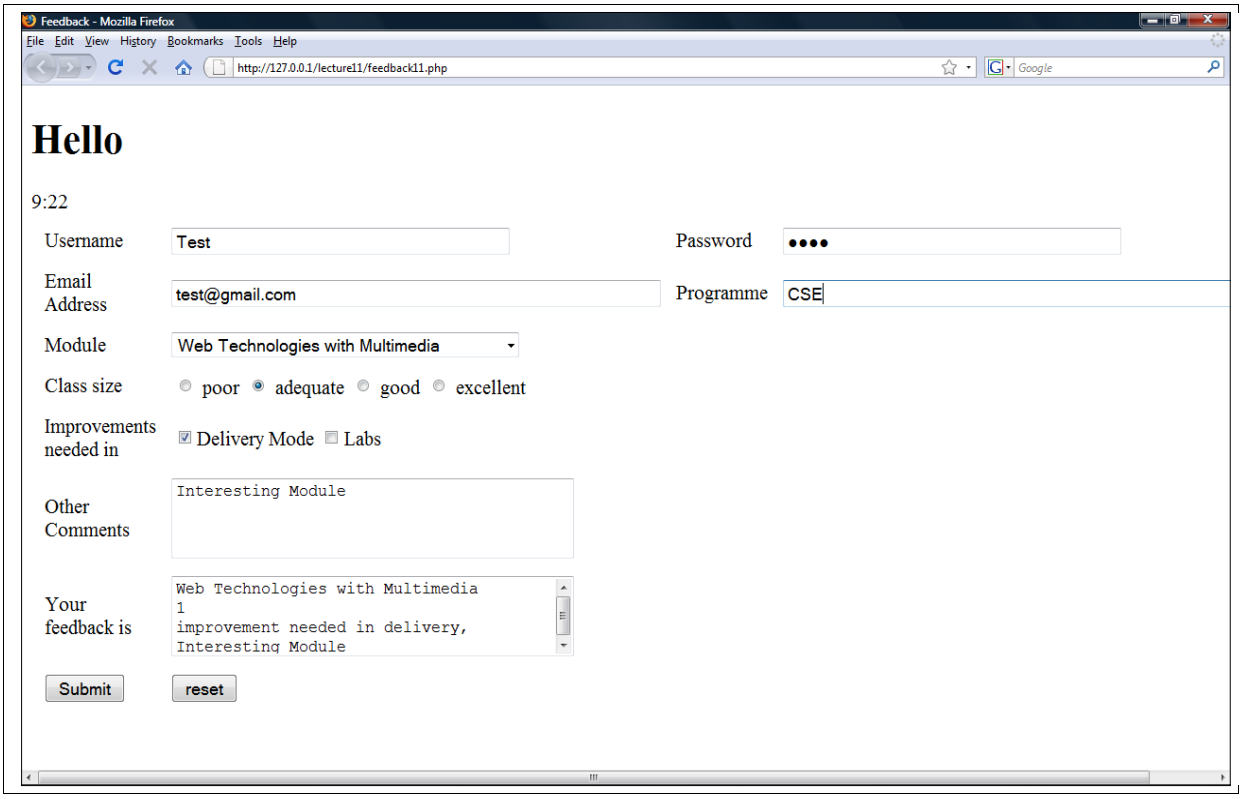

Figure 2: feedback.php

The codes for the page **feedback.php** have been provided in *listing 2* below:

Listing 2

```
<?php
 //connecting to database
 include("db_connect.php");

 $Rs = mysql_query("SELECT * FROM modules");

 if (mysql_num_rows($Rs)<0)
 {
  mysql_close($con);
  header( 'Location: error.html') ;
 } //retrieving module information

 $Rs1 = mysql_query("SELECT * FROM classsizes");
 if (mysql_num_rows($Rs1)<0)
 {
  mysql_close($con);
  header( 'Location: error.html') ;
 }
?>
<html>
<head><title>Feedback</title></head>

<body>
<script type="text/javascript" src=validations.js></script>
<script type="text/javascript">
<!--
var curdate=new Date();
```

```
document.write("<H1>Hello</H1>");
document.write(curdate.getHours()+":"+curdate.getMinutes());
function validateForm()
{
  if (!validateNumeric(document.forms[0].txt_username.value,'Username'))
    return false;
  if (!validateBlank(document.forms[0].txt_password.value,'Password'))
    return false;
  return true;
}
function showFeedback()
{
//code comes here
}-->></script>

<form name="frm_feedback" id=frm_feedback action="feedback_pro9.php?id=test" method="post" onsubmit="return validateForm()">
<table cellspacing=10 >
<tr>
<td>Username</td>
<td><input type=text name="txt_username" maxlength=40 size=40 ></td>
<td>Password</td>
<td><input type=password name="txt_password" maxlength=40 size=40></td>
</tr>
<tr>
<td>Email Address</td>
<td><input type=text name="txt_email" maxlength=60 size=60 ></td>
<td>Programme</td>
<td><input type=text name="txt_programme" maxlength=60 size=60></td>
</tr>
<tr>
<td>Module</td>
<td><select name="txt_module">
     <? while ($rows = mysql_fetch_array($Rs)){?>
<option value=<?echo $rows['modulecode'];?>><?echo $rows['moduledesc'];?>
</option>
<?}?>
  </select></td>
<td></td><td></td>
</tr>
<tr>
<td>Class size</td>
<td colspan=3>
  <?while ($rows = mysql_fetch_array($Rs1)){?>
   <input type=radio name="txt_classsize" value='<?echo $rows["classsize"];?>'>
   <? echo $rows["classsizedesc"];?>
</td>
</tr>
<tr>
<td>Improvements needed in</td>
<td colspan=3><input type="checkbox" name="txt_delivery" value="ON" >Delivery Mode
<input type="checkbox" name="txt_labs" value="ON" >Labs</td>
</tr>
<tr>
<td>Other Comments</td>
<td><textarea name="txt_others" cols=40 rows=3></textarea></td>
<td></td><td></td>
</tr>
<tr>
<td>Your feedback is</td>
<td><textarea name="txt_feedback" cols=40 rows=3 ></textarea></td>
<td></td><td></td>
</tr>
<tr>
<td><input type=submit value="Submit"></td>
<td><input type=reset value=reset></td>
<td></td><td></td>
</tr>
 </table>
 </body>
</html>
<?
mysql_close($con);
?>
```

As the user enters his feedback, the information provided is displayed in the textarea, *txt_feedback*, as shown above. Only the *module*, the *classsize*, the *improvements needed in* and the *other comments* are displayed.

a) Write the function *showFeedback()* that captures the various information as the user is entering them and displays them in the *txt_feedback* textarea.
- You may find the following functions/events useful: onClick, onChange, length, text, value

[10 marks]

b) Indicate how the function *showFeedback()* is called in the page **feedback.php**.

[3 marks]

c) There is one small problem in the example that has been provided. When the user selects the various classsizes, the values -1, 1, 2 and 3 are displayed instead of poor, adequate, good and excellent (refer to Figure 2 above). Explain why this is the case.

[3 marks]

d) One solution to this problem would be to save the data in the *classsizes* table in an xml file instead of a table. The **classsize.xml** has been provided below.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<feedback>
    <classsize>
    <description>poor</description>
    <value>-1</value>
    </classsize>
    <classsize>
    <description>adequate</description>
    <value>1</value>
    </classsize>
    <classsize>
    <description>good</description>
    <value>2</value>
    </classsize>
    <classsize>
    <description>excellent</description>
    <value>3</value>
    </classsize>
</feedback>
```

i. Write the **classsize.xsd** schema to validate the above xml file.

[6 marks]

ii. Write the code required to link the **classsize.xsd** schema to the above XML file.

[1 mark]

iii. The **feedback.php** can now be rewritten using the xml file instead of the table *classsizes*. Write the **classsize.xsl** file which will read the **classsize.xml** file and display the radio buttons with their values as shown in Figure 3 below. There is no need to write the code to integrate **classsize.xsl** and **classsize.xml** in **feedback.php**.

Note: When used within an attribute assignment "{path}" has the exact same effect as <xsl:value-of select="path"/> used outside of attribute assignments.

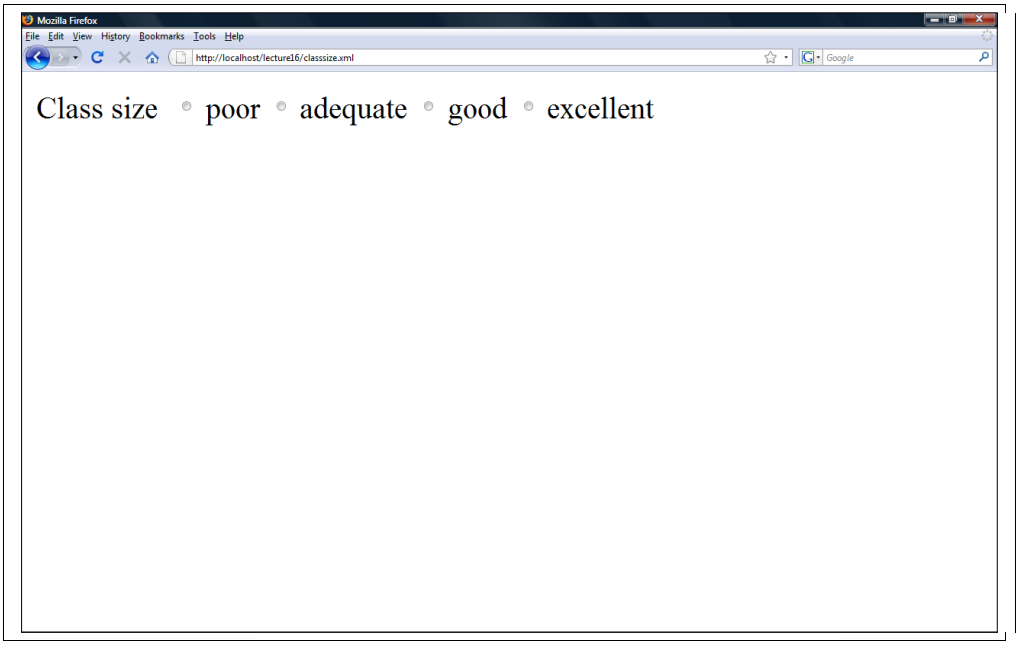

Figure 3 – classsize.xml

[6 marks]

iv. Write the code required to link the **classsize.xsl** to the above XML file.

[1 mark]

Question 3

Answer the following questions with regard to Selenium and testing. Use a couple of sentences to answer each question.

Listing 3

<!-- Web page source for Selenium testing question -->

```
<html>
<head>
  <title>Feedback</title>
</head>
<body>
<script type="text/javascript" src=validations.js></script>
<script type="text/javascript">
<!--
function validateForm()
 {
  if (!validateNumeric(document.forms[0].txt_username.value,'Username'))
     return false;
  if (!validateBlank(document.forms[0].txt_password.value,'Password'))
     return false;
  return true;
 }
 -->
 </script>
  <form id=frm_feedback action="sb_feedback_pro.php?id=test"
       method="post" onsubmit="return validateForm()">
   <table cellspacing=10 >
     <tr>
       <td>Username</td>
       <td><input type=text name="txt_username" maxlength=40 size=40 ></td>
       <td>Password</td>
       <td><input type=password name="txt_password" maxlength=40 size=40></td>
    </tr>
    <tr>
       <td>Module</td>
       <td><select name="txt_module">
             <option value="CSE2001Y" Selected>Software Engineering</option>
             <option value="CSE2002Y">Database Systems</option>
             <option value="CSE2003Y">Web Technologies</option>
          </select></td>
       <td></td>
       <td></td>
    </tr>
    <tr>
       <td>Class size</td>
       <td colspan=3><input type=radio name="txt_classsize" value='-1'>Poor
                    <input type=radio name="txt_classsize" value='1'>Adequate
                    <input type=radio name="txt_classsize" value='2'>Good</td>
    </tr>
    <tr>
       <td><input type=submit value="Submit"></td>
       <td><input type=reset value=reset></td>
       <td></td>
       <td></td>
    </tr>
   </table>
</body>
</html>
```

(a) Suppose that you were writing a selenium script to test the web page displaying the HTML code entitled *Web page source for Selenium testing question* shown as *Listing 3*. Assume that this script currently behaves exactly according to the application's specifications. Your Selenium tester needs to identify changes that affect the correct behavior of this page. List the 6 most meaningful tests that you could perform on the web page, without being able to look at the page returned by the server after successfully posting the data. That is, your tests must be done on this page alone, and cannot look at the server response. For each of the tests, indicate which line number(s) in the HTML code are being tested. You do not need to write the "Selenese", just describe the test.

[6 marks]

(b) What is the purpose of regression testing? What is meant by "regression"? (Be brief.)

[3 marks]

(c) Using the terms "precondition", "postcondition" and "program state", explain how unit testing works. (Be brief.)

[3 marks]

(d) In the context of unit testing, what is test coverage? Why is test coverage important? (Be brief.)

[3 marks]

Question 4

Answer the following questions with respect to web application security. Use a few sentences to answer each question. The module feedback application mentioned in the questions includes the simplified web page example entitled *Web page source for Selenium testing question* (shown as *Listing 3*).

(a) One method for attacking the security of a web application – such as the module evaluation program – is to directly access the PHP files via the browser. It is quite common to name PHP included files to have a ".inc" extension (e.g., "db.inc"). First, explain why a file such as "db.inc" is a security risk. Next, identify two methods for reducing that security risk. Finally, of the two methods you identified, explain which is the more secure and why.

[5 marks]

(b) One source of web application attacks is SQL injection. First, explain what SQL injection is. Next, give an example of how entering the username in the feedback page of the module feedback program might be used to perform SQL injection. Last, explain what you could do to prevent SQL-injection attacks via user-entered data such as the username.

[5 marks]

(c) Explain the security principle, Defense in Depth. Give a plausible explanation of how Defense in Depth could be achieved by the module feedback application. Present three methods. This is not a request to list all possible defense elements, but to list three defense elements that help illustrate Defense in Depth.

[5 marks]

SECTION B

Answer only one of the following two questions. Both questions pertain to the following AJAX problem.

Consider the following set of codes for *feedbackAjax.php* in *listing 4*, that makes use of AJAX to save the feedback data and adds the saved data to the lower part of the page. *savefeedback.php* captures all the data posted and tries to insert them into the database and returns the following codes:

- 200 – Insert was successful
- 1024 – feedback already inserted for user for particular
- -200 – Some error occurred

Listing 4

```
<?php
 //validate whether the user is connected first
 include("validate_session.php");
 include("db_connect.php");
 $Rs1 = mysql_query("SELECT * FROM modules");

 if (mysql_num_rows($Rs1)<0)
 {
  mysql_close($con);
  header( 'Location: error.html') ;
 } //retrieving module information

 $Rs2 = mysql_query("SELECT * FROM classsizes");
 if (mysql_num_rows($Rs2)<0)
 {
  mysql_close($con);
  header( 'Location: error.html') ;
 }
?>

<html>
<head>
 <title>Feedback</title>
</head>

<body>
<script type="text/javascript" src=validations.js></script>
<script type="text/javascript">
<!--
 document.write("<H1>Hello</H1>");

//Function taken from http://www.captain.at/howto-ajax-form-post-request.php
  var http_request;
  function makePOSTRequest(url, parameters) {
      http_request = false;
      if (window.XMLHttpRequest) { // Mozilla, Safari,...
         http_request = new XMLHttpRequest();
         if (http_request.overrideMimeType) {
```

```
            // set type accordingly to anticipated content type
          //http_request.overrideMimeType('text/xml');
          http_request.overrideMimeType('text/html');
        }
      } else if (window.ActiveXObject) { // IE
        try {
          http_request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
          try {
            http_request = new ActiveXObject("Microsoft.XMLHTTP");
          } catch (e) {}
        }
      }
      if (!http_request) {
        alert('Cannot create XMLHTTP instance');
        return false;
      }

      http_request.onreadystatechange = processContents;
      http_request.open('POST', url, true);
      http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
      http_request.setRequestHeader("Content-length", parameters.length);
      http_request.setRequestHeader("Connection", "close");
      http_request.send(parameters);
    }

    function processContents() {
      if (http_request.readyState == 4) {
        if (http_request.status == 200) {
          //alert(http_request.responseText);
          result = http_request.responseText;
          if (result=="200")
                {

           tablestr= "<tr><td width=150>"+document.forms['frm_feedback'].txt_module.value+"</td><td
width=150>"+getCheckedValue(document.forms['frm_feedback'].elements['txt_classsize'])+"</td> <td width=200>";
           if (document.forms['frm_feedback'].txt_delivery.checked)
                        tablestr=tablestr +"Improvement needed";
                else
                        tablestr=tablestr +"No Improvement needed";

           tablestr=tablestr + "</td><td width=200>";
           if(document.forms['frm_feedback'].txt_labs.checked)
            tablestr=tablestr +"Improvement needed";
                else
                        tablestr=tablestr +"No Improvement needed";

           tablestr=tablestr +"</td><td width=250>";

           tablestr=tablestr + document.forms['frm_feedback'].txt_others.value+"</td></tr>";
           table_modules.innerHTML=table_modules.innerHTML+ tablestr;

                document.forms['frm_feedback'].txt_delivery.checked=false;
                document.forms['frm_feedback'].txt_labs.checked=false;
                document.forms['frm_feedback'].txt_others.value="";

//              //div_records.style.backgroundColor="red";
           div_records.style.visible="visible";
          }
          if   (result=="1024")
          {
            alert("You already posted feedback for this module");
          }
          //document.getElementById('myspan').innerHTML = result;
        } else {
          alert('There was a problem with the request.');
        }
      }
    }

 function get(obj) {

  if (!validateBlank(document.forms['frm_feedback'].txt_email.value, "email"))
```

```
    return false;
  var poststr = "txt_email=" + escape(encodeURI(document.getElementById("txt_email").value)) + "&txt_module=" +
escape(encodeURI(document.getElementById("txt_module").value)) + "&txt_classsize="
 + escape(encodeURI(getCheckedValue(document.forms['frm_feedback'].elements['txt_classsize']))) +
"&txt_others="+escape(encodeURI(document.getElementById("txt_others").value));

 if (document.forms['frm_feedback'].txt_delivery.checked)
  poststr = poststr + "&txt_delivery="+escape(encodeURI("on"));
 if (document.forms['frm_feedback'].txt_labs.checked)
  poststr = poststr + "&txt_labs="+escape(encodeURI("on"));
 makePOSTRequest('savefeedback.php', poststr);
}

//getCheckedValue(document.forms['radioExampleForm'].elements['number']))

// return the value of the radio button that is checked
// return an empty string if none are checked, or
// there are no radio buttons
//source = http://www.somacon.com/p143.php
function getCheckedValue(radioObj) {
        if(!radioObj)
                return "";
        var radioLength = radioObj.length;
        if(radioLength == undefined)
                if(radioObj.checked)
                        return radioObj.value;
                else
                        return "";
        for(var i = 0; i < radioLength; i++) {
                if(radioObj[i].checked) {
                        return radioObj[i].value;
                }
        }
        return "";
}

-->
</script>

<div>
 <form id="frm_feedback" name="frm_feedback" action="javascript:get(document.getElementById('frm_feedback'))">
 <table cellspacing=10 >
  <tr>
   <td>Email Address</td>
   <td><input type=text name="txt_email" id="txt_email" maxlength=60 size=60 value=<?echo $email;?>></td>
   <td>Module</td>
   <td><select name=txt_module id=txt_module>
       <? while ($rows = mysql_fetch_array($Rs1)){?>
       <option value=<?echo $rows['modulecode'];?>><?echo $rows['moduledesc'];?></option>
       <?}//end while
              ?>
    </select></td>
  </tr>
  <tr>
   <td>Class size</td>
   <td colspan=3>
     <?while ($rows = mysql_fetch_array($Rs2)){?>
      <input type=radio name="txt_classsize" id="txt_classsize" value='<?echo $rows["classsize"];?>'
      <?if ($rows["classsize"]=='-1') echo " checked";?>>
      <?
       echo $rows["classsizedesc"];
      }//end while?>
   </td>
  </tr>
  <tr>
   <td>Improvements needed in</td>
   <td colspan=3><input type=checkbox name=txt_delivery id=txt_delivery >Delivery Mode
    <input type=checkbox name=txt_labs id=txt_labs >Labs</td>
  </tr>
  <tr>
   <td>Other Comments</td>
   <td><textarea name="txt_others" id="txt_others" cols=40 rows=3></textarea></td>
   <td></td>
```

```
  <td></td>
 </tr>
 <tr>
  <td><input type="button" name=button onclick="javascript:get(this.parentNode);" value=Submit></td>
  <td><input type=reset value=Reset></td>
  <td></td>
  <td></td>
 </tr>
</table>

</div>

<div id=div_records style="background-color:lightblue">
<HR>
 These are feedbacks you have already posted<br><br>

<table border=1 id=table_modules>
 <tr>
  <th width=150>Module Code</th>
  <th width=150>Class Size</th>
  <th width=200>Delivery Mode</th>
  <th width=200>Labs</th>
  <th width=250>Other Comments</th>
 </tr>
</table>
</div>

<br>
</form>
 </body>
</html>

<?
 mysql_close($con);
?>
```

Question 5

One problem with this page is that it works with Firefox but not with Internet Explorer. Internet Explorer generates an '*unknown runtime error*'

Briefly explain why the above codes do not work with Internet Explorer.

[5 marks]

Modify these codes such that *feedbackAjax.php* works with both Firefox and Internet Explorer.

[15 marks]

Question 6

Another set of problems with this page are as follows:

- The table headers are displayed even if there are no feedbacks that have been posted as shown in Figure 4.
- It might be that a user has already input feedback for different modules and these feedback should be listed when the page is loaded.

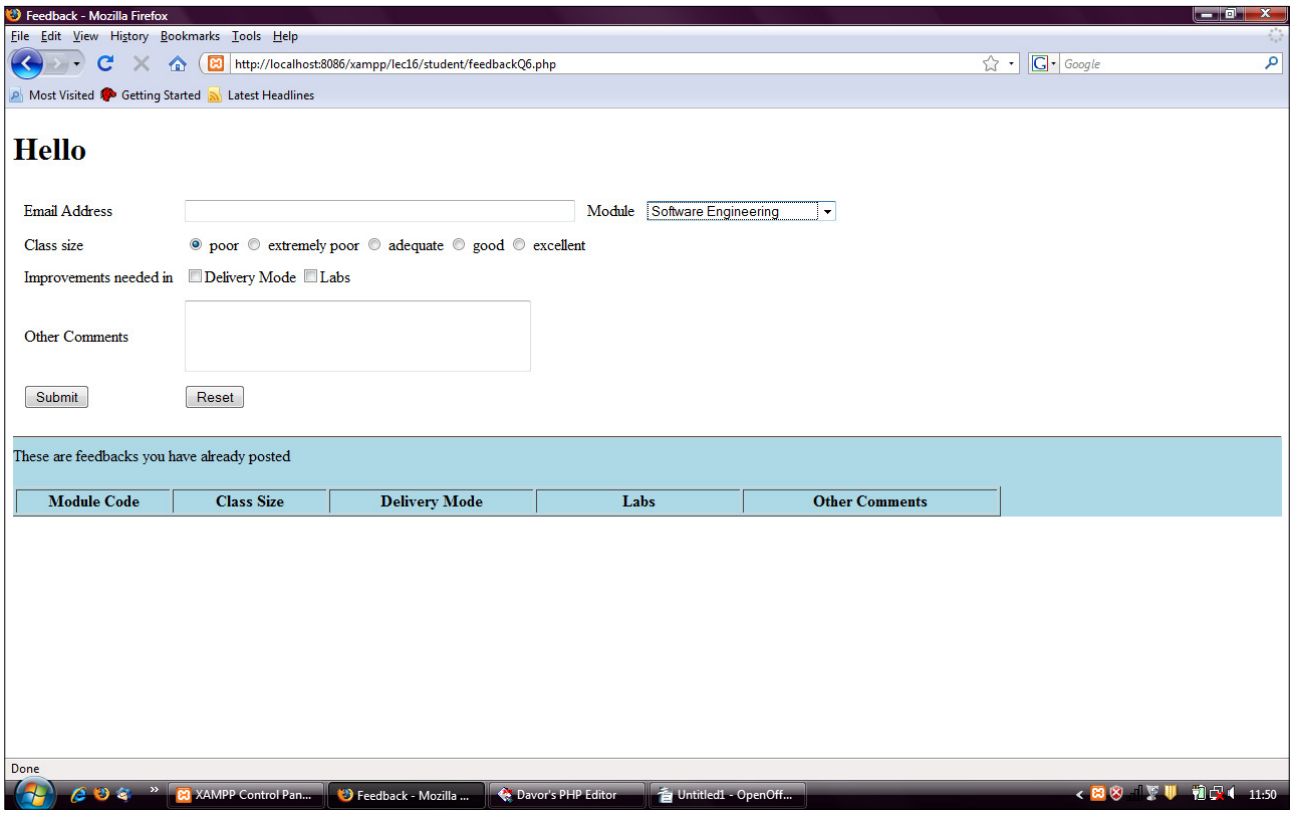

Figure 4: feedbackAjax.php

The new database schema of the system is proposed as follows.
classsizes (classsize, classsizedesc)
feedbacks (email, modulecode, moduleyear, classsize, delivery, labs, Othercomments, moderation)
modules (modulecode, moduledesc)
users (username, pass, email, programme)

After logging into the system, the system keeps the email of the user in a session variable *email*. This data is used to find the feedbacks the user has already posted.

Modify the codes such that:

- The table details are only displayed in case there are feedbacks already submitted by the user in a previous session.
- In case any feedback exists, it should be added to the table.

Note: Make sure that you have codes that will render the table visible once the first feedback has been submitted.
Your codes may only work in Firefox

[20 marks]

END OF QUESTION PAPER

/kp