# UNIVERSITY OF MAURITIUS

# FACULTY OF ENGINEERING

## SECOND SEMESTER EXAMINATIONS

## MAY 2010

| | | | |
|---|---|---|---|
| **PROGRAMME** | **BSc (Hons) Computer Science**<br>**BSc (Hons) Information Systems** | | |
| **MODULE NAME** | **Web Technologies I** | | |
| **DATE** | **Wednesday**<br>**26 May 2010** | **MODULE CODE** | **CSE 1041(1)** |
| **TIME** | **13.30 – 15.30 Hrs** | **DURATION** | **2 Hours** |
| **NO. OF QUESTIONS SET** | **4** | **NO. OF QUESTIONS TO BE ATTEMPTED** | **4** |

INSTRUCTIONS TO CANDIDATES

**Answer all 4 questions.**

**All questions carry equal marks.**

**All questions refer to the same feedback application and on the Feedbacks database design provided.**

Consider the following Database to keep track of Feedbacks on different modules.

**users(**username:*varchar(100)*, password:*varchar(100)***)**
**modules(**modulecode:*varchar(10)*, moduledesc:*varchar(100)***)**
**classsizes(**classsize:*varchar(10)*, classsizedesc:*varchar(100)***)**
**feedbacks(**username:*varchar(100)*, modulecode:*varchar(10)*, moduleyear:*varchar(4)*, programme:*varchar(100)*, classsize:*varchar(10)*, delivery:*tinyint(1)*, labs:*tinyint(1)*, othercomments:*varchar(100)*, moderation: *enum('a', 'r', 'p')***)**

Note:

- **feedbacks**.*username* refers to **users**.*username*
- **feedbacks**.*modulecode* refers to **modules**.*modulecode*
- **feedbacks**.*classsize* refers to **classsizes**.*classsize*
- All primary keys are underlined

## Question 1

We would like to create a page, **deletefeedback.php**, to delete **Approved Feedbacks** that are stored in the database. A screenshot of the page is shown in **Figure 1.**

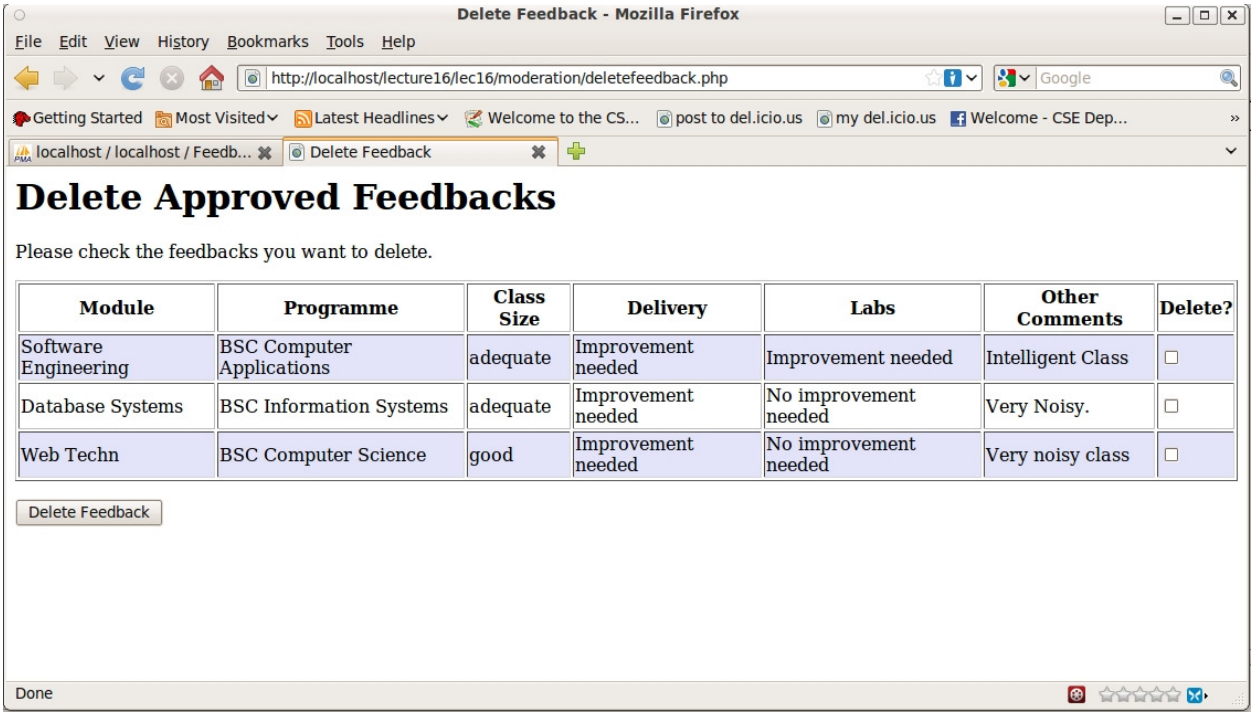


**Figure 1**: *deletefeedback.php*

***../Cont'd next page***

**Question 1** ***(Cont'd)***

When clicking on '**Delete Feedback**' button, the page, **del_feedback_pro.php**, is called. Part of the codes for the page, **del_feedback_pro.php**, is given below. Comments are given to missing portion of codes.

**del_feedback_pro.php**

```
<?
/*
---     the codes for retrieving the session variable 'username' and     assigning it to $user
---     validation to redirect the user to login.php if session is invalid
*/

include("db_connect.php");
$del_me = $_POST['dellist'];
?>

<html>
<head>
  <title>Deleting Feedbacks</title>
</head>
<body>

<?
  foreach ($del_me as $feed)
  {
        $delete = "delete from feedbacks where username='$user' and modulecode='$feed'";

        if (!mysql_query($delete))
              throw new Exception('Feedback could not be deleted');
        else
        {
              echo "Logged in as $user <br>";
              echo "Deleted Feedback of module code $feed<br>";
        }
  }
?>
</body>
</html>
<?
        mysql_close($con);
?>
```

Assuming that the pages **db_connect.php, login.php** and **error.html** already exist, you are required to write all the codes for the page, **deletefeedback.php**, such that it calls the page **del_feedback_pro.php** above. In addition, you should ensure the following:

***.../Cont'd next page***

**Question 1** ***(Cont'd)***

- The session variable for the username is valid, otherwise redirect the user to the **login.php** page.
- The proper SQL query for retrieving the user's approved feedbacks if available.
- Error handling should be taken care where user is redirected to an error page, **error.html** where required.
- The status of delivery and labs field should be displayed accordingly, since they store either 0 or 1 as values:
    - if delivery is 0, status is '*No improvement needed*', otherwise '*Improvement needed*'
    - if labs is 0, status is '*No improvement needed*', otherwise '*Improvement needed*'

**[25 marks]**

## Question 2

Consider the following page, **process_module.php**, which makes use of Ajax technology to dynamically add and delete modules to the database and reload part of the page.

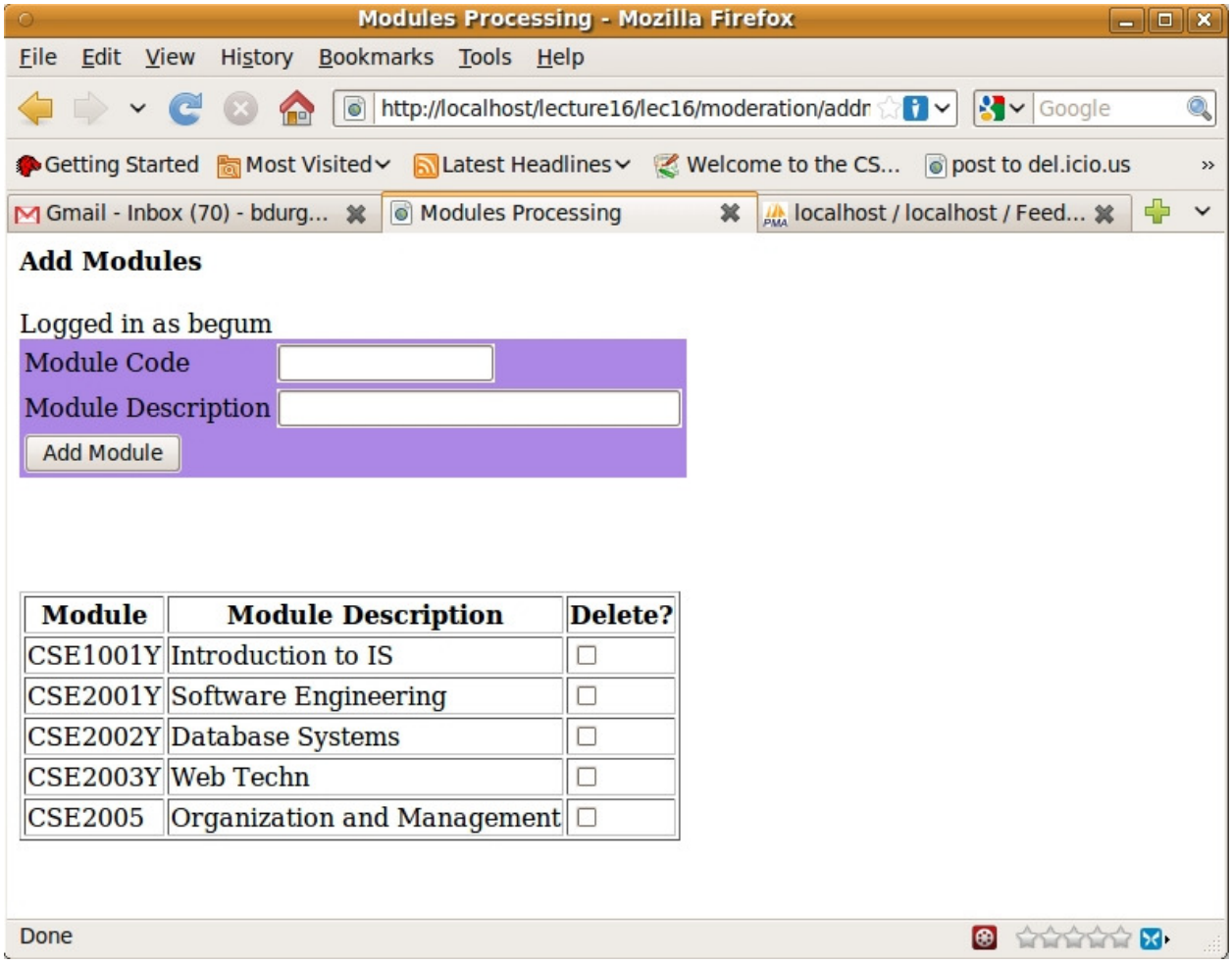


**Figure 2**: *process_module.php*

Part of the codes for the page, **process_module.php**, is given below and comments are given to missing portion of codes. Also, we assume that **ajax_functions.js** and **db_connect.php** already exist.

*.../Cont'd next page*

**Question 2** ***(Cont'd)***

**process_module.php**

```
<?
/*
---     the codes for retrieving the session variable 'username' and     assigning it to $user
---     validation to redirect the user to login.php if session is invalid

*/

<?
include("db_connect.php");
$sql_select = "SELECT * FROM modules";
$res = mysql_query($sql_select);
?>
<html>
<head>
 <title>Modules Processing</title>
  <script src="ajax_functions.js" type="text/JavaScript"></script>
  <script>
  <!--
 /*Function to set url, params and invoking a request to add a new module*/
  function addModule() { }

 /*Function to set url, params and invoking a request to delete a module*/
  function delModule() { }
  -->
  </script>
</head>
<body>
<h4>Add Modules</h4>
Logged in as <?echo $user;?>
<form name="frm_modules">
<table>
  <tr>
     <td>Module Code</td>
     <td><input type=text name="txt_modulecode" id="txt_modulecode"
          size=15></td>
   </tr>
<tr>
     <td>Module Description</td>
     <td><input type=text name="txt_moduledesc" id="txt_moduledesc"    size=30></td>
  </tr>
  <tr>
    <td><input type=button name="add_btn" value="Add Module"></td>
    <td> </td>
  </tr>
```

```
 </table>
</form>
<br><br><br>
<span id="myspan" name="myspan">
<table border="1">
 <tr>
  <th>Module</th>
  <th>Module Description</th>
  <th>Delete?</th>
 </tr>
<?
  While ($rows = mysql_fetch_array($res)) {
?>
  <tr>
   <td><? echo $rows["modulecode"]; ?></td>
    <td><? echo $rows["moduledesc"]; ?></td>
    <td><input type="checkbox" name="dellist" value="<?echo $rows['modulecode']?>"></td>
  </tr>
<?  } //end while
?>
</table>
</span>
</body>
</html>
<?
mysql_close($con)
?>
```

The **ajax_functions.js** contains the following codes:

**ajax_functions.js**

```
//Function taken from http://www.captain.at/howto-ajax-form-post-request.php

  var http_request; //global variable

  function makePOSTRequest(url, parameters) {
    http_request = false;

          /* For Firefox*/
          if (window.XMLHttpRequest) {
      http_request = new XMLHttpRequest();
          } else /*For IE*/
            if (window.ActiveXObject) {
          try {
            /*For some versions of IE*/
          http_request = new ActiveXObject("Msxml2.XMLHTTP");
          } catch (e) {
```

```
        try {
          /*For some other versions of IE*/
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e) {}
      }
    }

    if (!http_request) {
      alert('Cannot create XMLHTTP instance');
      return false;
    }

     //Add the request
    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.onreadystatechange = theResponse;
    http_request.send(parameters);
  }

/*working with the server response*/
function theResponse() {
        if (http_request.readyState == 4) {
                if (http_request.status == 200) {
                        result = http_request.responseText;
                } else {
                        alert('There was a problem with the request.');
                }
        }
}
```

a) The purpose of the page, **process_module.php**, is to dynamically add new modules and delete existing modules. **addmodule_pro.php** and **delmodule_pro.php** are processing pages to add a new module and delete an existing module respectively.

(i) Write the codes for the Javascript function **addModule()**, such that the corresponding page and parameter(s) are sent to a XMLHTTPRequest instance. **[4 marks]**

(ii) Write the codes for the Javascript function **delModule()**, such that the corresponding page and parameter(s) if any are sent to a XMLHTTPRequest instance. You are required to prompt the user to continue or stop before deleting a module. Assume that the **delModule()** function is invoked upon checking a checkbox. **[6 marks]**

***.../Cont'd next page***

**Question 2** ***(Cont'd)***

(iii) Write the line of code that would call the function **delModule()** defined in (ii) above. Indicate where you would add the codes.

**[2 marks]**

b) The following is a screenshot of the page, **process_module.php**, after adding a new module. Notice that the new module now appears in the lower table.

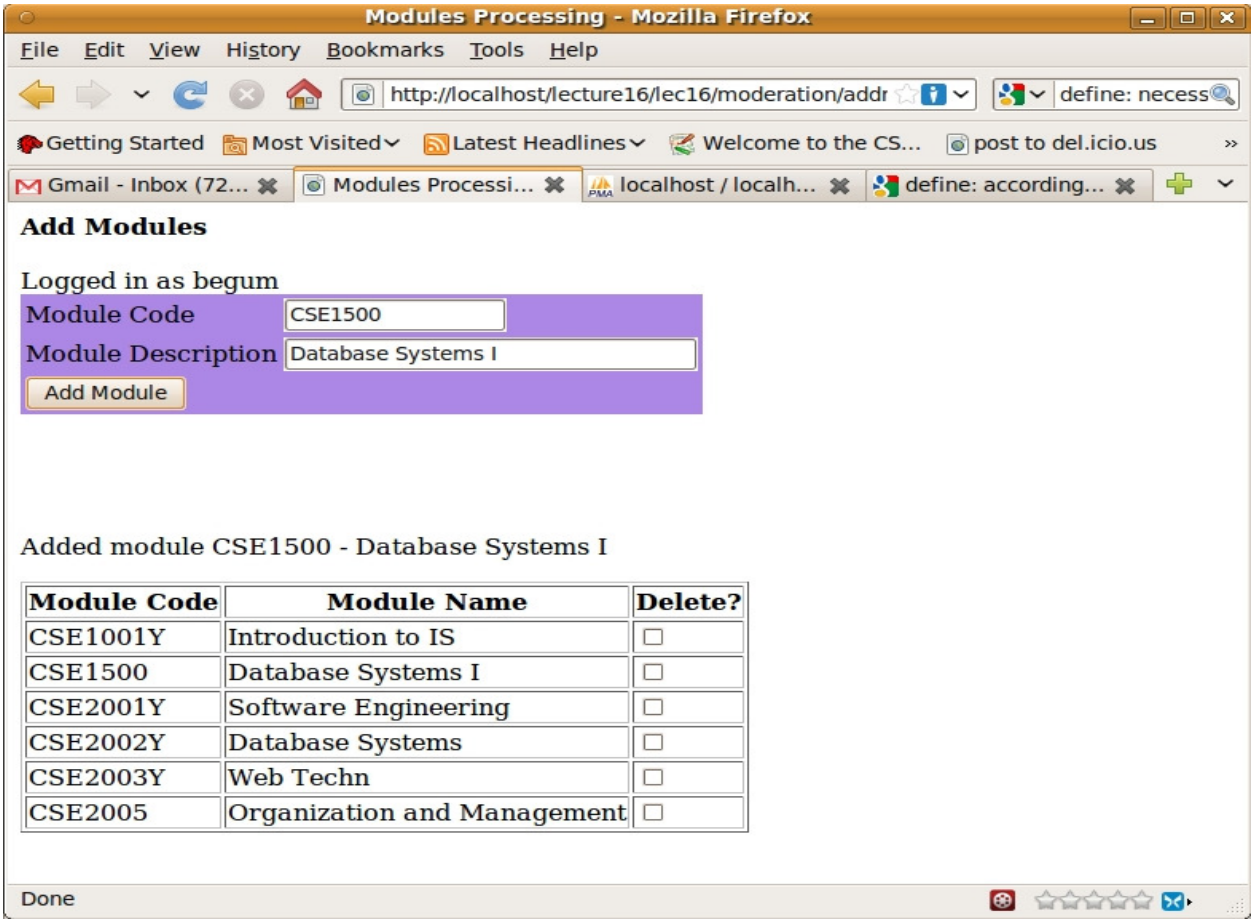


**Figure 3**: *process_module.php* after a successful module addition

(i) You are required to write the code for the page, **addmodule_pro.php**, to ensure the resulting screen of **Figure 3**. **[10 marks]**

(ii) The intended behavior for the above is NOT working properly and the problem resides in function **theResponse()** in **ajax_functions.js**. Give the change that is required for it to work properly. **[3 marks]**

**Question 3**

a) In the page **process_module.php** above, if the user leaves the textboxes **txt_modulecode** and **txt_moduledesc** blank and clicks on the **Add Module** button, an empty row is added to the table on the page. Write the JavaScript function **checkBlank()** that validates the 2 textboxes. Indicate any change in your code to call the function **checkBlank()**. **[4 + 3 marks]**

b) In **Figure 1**, **deletefeedback.php**, each odd row of the table should have a background color of #E3E4FA. Modify your codes to reflect this. You need only write the few lines of code needed for that. You should make use of inline styles.

**[4 marks]**

c) **Process_module.php**, in **Figure 3,** has been formatted using an external style sheet, **style.css**, found in the same folder. The formatting is as follows:

- the background colour of the table is #AD87E6
- the colour of the button is #FFE4C4.

i. Write the external style sheet, **style.css,** defining 2 classes **tbl** and **btn**.

ii. Write the codes to link the **process_module.php** page to the **style.css** page.

iii. Indicate any changes to be made to your codes to use the 2 classes just created.

**[3 + 1 + 2 marks]**

*.../Cont'd next page*

**Question 3** ***(Cont'd)***

d) Consider the page **enterfeedback.html** shown in **Figure 4** and the code that follows.

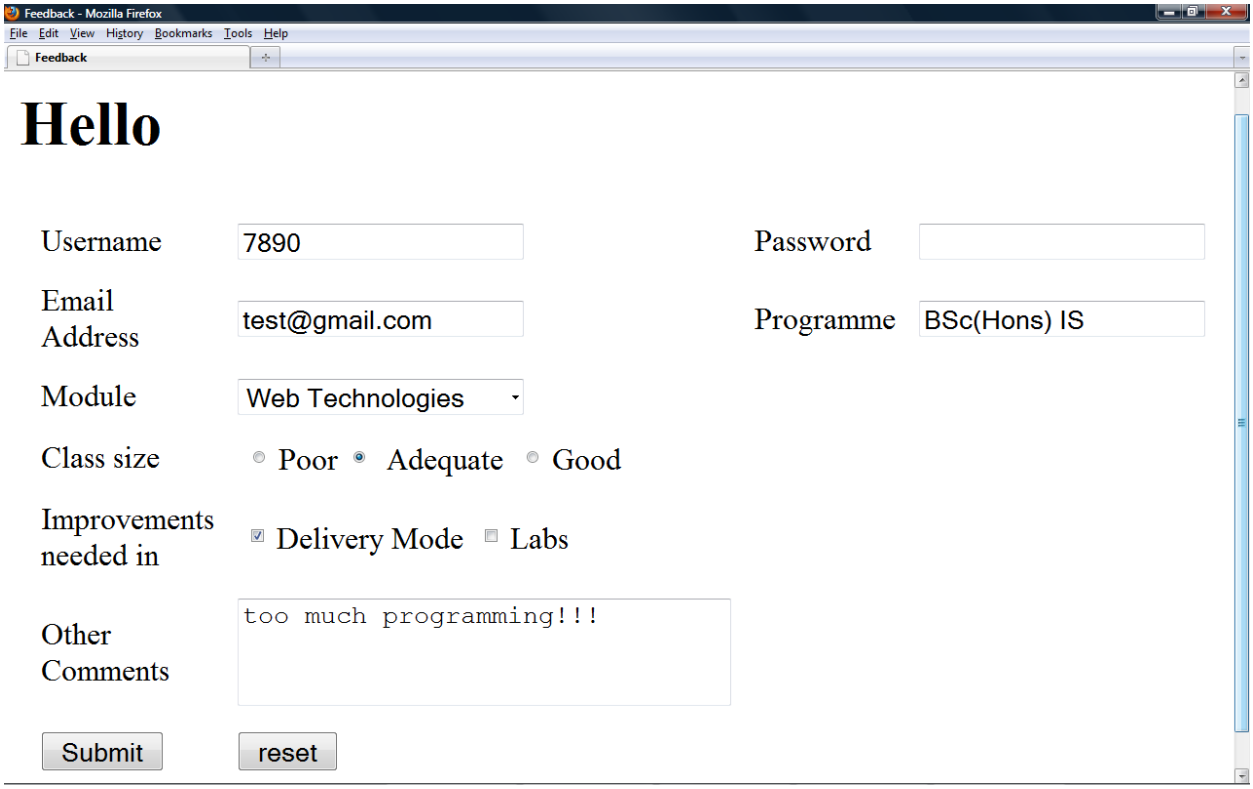


**Figure 4**: *enterfeedback.html*

```
<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 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="feedback_pro9.php" method="get" 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>
         <option value="CSE2002Y">Database Systems</option>
         <option value="CSE2003Y">Web Technologies</option>
         <option value="CSE2001Y" Selected>Software Engineering</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>Improvements needed in</td>
     <td colspan=3><input type=checkbox name=txt_delivery>Delivery Mode      <input
type=checkbox name=txt_labs>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><input type=submit value="Submit"></td>
    <td><input type=reset value=reset></td>
    <td></td>
```

```
    <td></td>
  </tr>
 </table>
</body>
</html>
```

i. Predict the URL that will appear in your browser on clicking the 'Submit' button in the **Figure 4** above. (The password entered is web).

**[5 marks]**

ii. Consider the JavaScript code which is found in the code above. The code has been modified as shown below. What difference does it make if any?

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

**[3 marks]**

## Question 4

a) For administrative staff to access the various feedback pages above, they need to login first. The desired behavior is that after successful login, the user can access the different web pages and in this case he can access the **managefeedbacks.html** page. The page **login.php** is displayed in **Figure 5** below. There is a combo box on the page dynamically displaying all the users from the table **users** in the **Feedback** database. When the user clicks on the submit button (Login), all information is sent to the **login_pro.php** page for processing.

Write the codes for **login.php**. You may make any assumption you wish.

**[7 marks]**

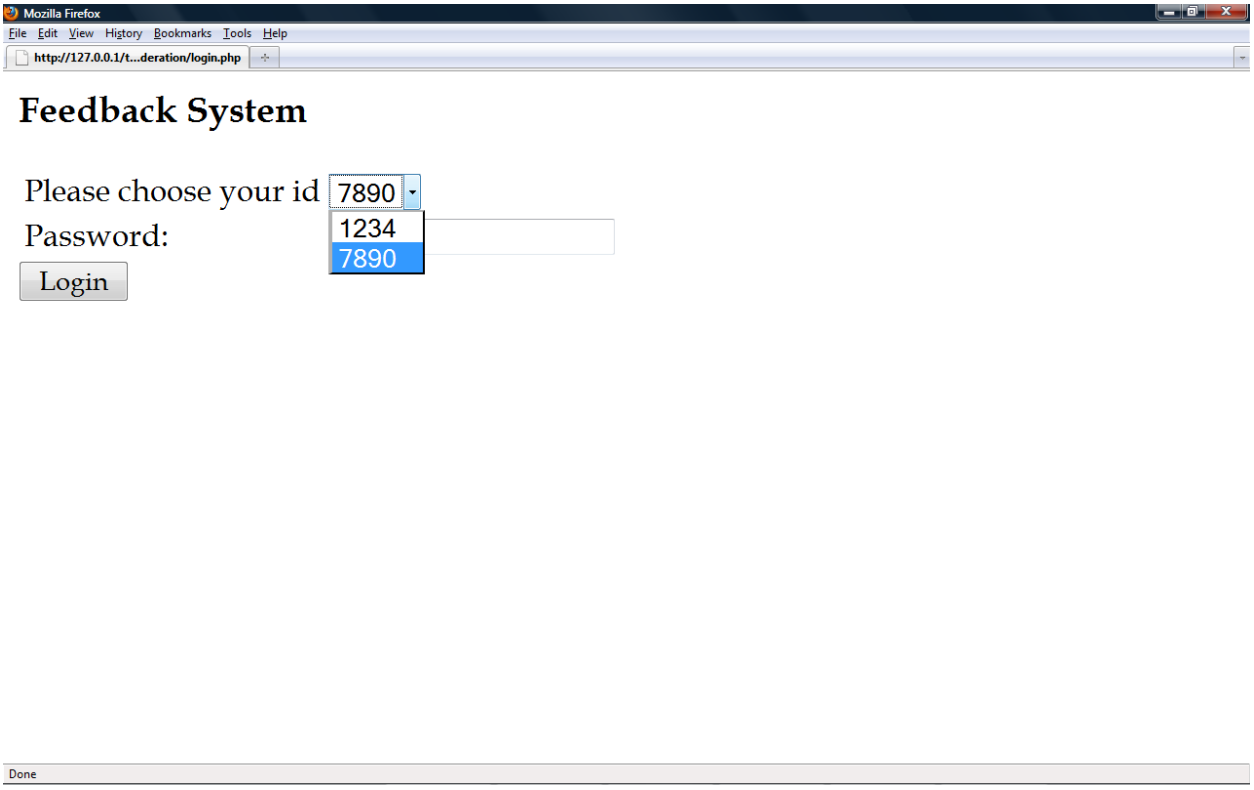


**Figure 5**: *login.php*

b) **Figure 6** displays the page **managefeedbacks.html**. There are 4 frames with their respective names and source files as follows: **banner.html** (frm_banner), **menu.html** (frm_menu), **main.html** (frm_main) and **copyright.html** (frm_copy). Write the codes for **managefeedbacks.html**.

*.../Cont'd next page*

**Question 4** ***(Cont'd)***

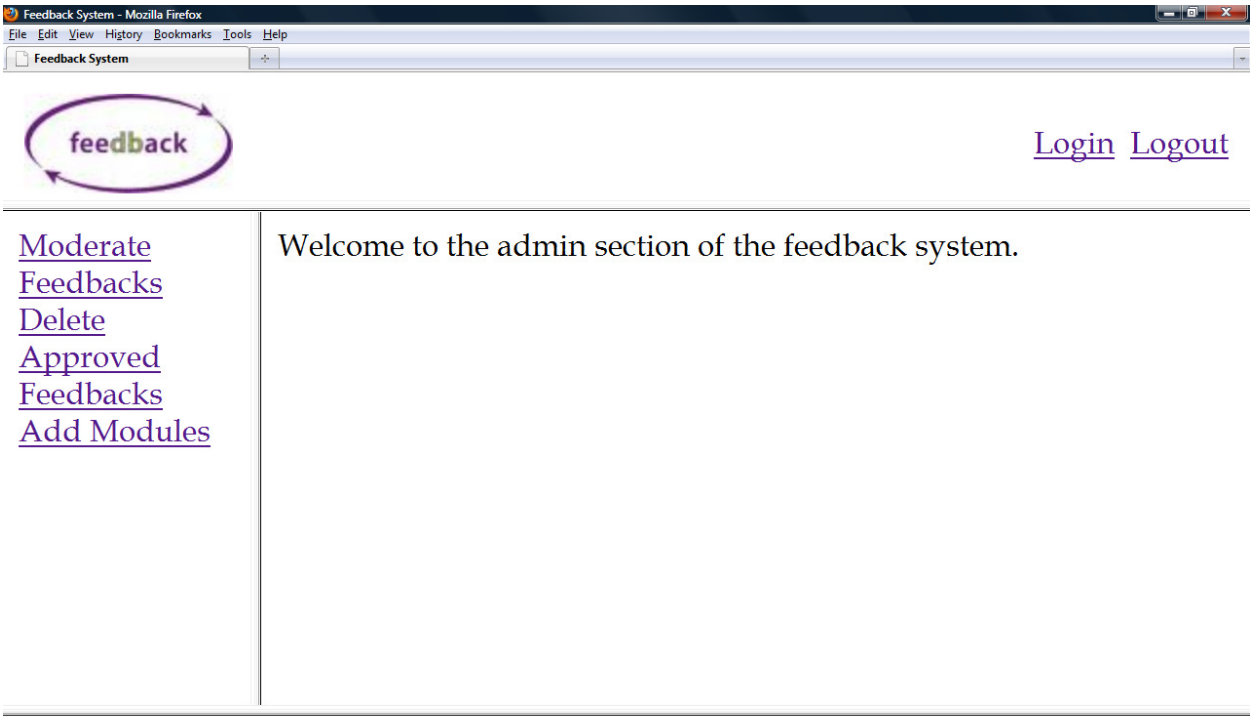


**Figure 6**: *managefeedbacks.html*

**[6 marks]**

c) There are 2 problems with the **managefeedbacks.html** page as displayed in **Figure 6** and as implemented in sections **a)** and **b)**. Identify the problems and provide the solutions. **[6 marks]**

d) Write the codes for **menu.html** given that the link **Moderate Feedbacks** opens the page **moderate.php**, the link **Delete Approved Feedbacks** opens the page **deletefeedback.php** and the link **Add Modules** opens the page **process_module.php**. All pages open in the main frame. **[4 marks]**

e) Consider the codes for **copyright.inc** shown below. Server Side Includes can be used for parametrization.

```
<h5 align="center">&copy;University of Mauritius 2010</h5>
```

What are the implications of using such a file with respect to an HTML file?

**[2 marks]**

**END OF QUESTION PAPER**

/ph