Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS10020
Creating Web Applications

Assignment 2, TP2, 2019

Server-Side Programming

Important Dates:

Due Date	**End of week 12** (Late submission penalty: 10% of total available marks per day)
Demonstration	Your tutorial: Week 13 (session one)

Individual Assignment. Contribution to Final Assessment: 25%

Purpose of the assignment

In this part of the assignment you will further enhance the website you developed in Parts 1A and 1B. You will extend the functionality of the website creating server-side PHP scripts to process and store the quiz-attempt data sent from the Web form. This information will have been collected in your HTML Forms. It will involve the creation of simple MySQL tables to store, update and retrieve information from a website.

In addition, you will create a Web page that allows a quiz supervisor to view, update and delete attempts.

There will be an opportunity to enhance your website beyond the basic requirements.

Prerequisite

If you fail to meet the essential requirements of Part 1B of the Assignment, ***before*** this assignment is submitted and marked you need to demonstrate that you have fixed problems. Note that these fixes will **not** alter the mark you received for Part 1B.

It is advisable to get these fixes complete and signed off well before you hand in this part of the assignment. The tutor will check the fixes and sign-off that they have been completed.
Your tutor will be happy to advise you during labs or during consultation sessions if you need assistance fixing your Part 1B.

How to get your fixes signed off:

1. Arrange a time with your tutor to check your work during your allocated tutorial or during a consultation time.
2. Bring a copy of the assessment printout from Part 1B
3. Your tutor will check that the fixes to your assignment address the issues identified on the mark sheet.
4. If the fixes are successful, your tutor will record this and you will be eligible to have this assignment assessed. If there are issues that have not been fixed, your tutor will inform you of this and you will have *a further chance* to fix the assignment.

A: Specified Requirements

Use only MySQL commands in this assignment.

1. Use PHP to reuse common elements in your website

PHP provides us with techniques to modularize and reuse our web application code. You need to refactor your web pages so that common static HTML elements such a menu, header and footer are written in common text files which are then included into your web pages. Name the include file(s) with a **`.inc`** extension (but don't forget to rename your main pages with a **`.php`** extension!).

2. Create a Quiz Attempts table

Create a table **`attempts`** in your MySQL database. The information in each attempt record should include the following fields with appropriate data types:

- Attempt id (auto-generated primary key)
- Date and time of the attempt (generated by PHP)
- First name
- Last name
- Employee number
- The number of the attempt (1 -3)
- Score for the attempt

When a user submits a quiz attempt to the website, if an **`attempts`** table does not already exist in your database it should be programmatically created by your code.

3. Marking the quiz at the server and adding validated record to the `attempts` table

Adapt the quiz form (quiz.html) you developed in Parts 1A and 21Bto *add* quiz attempt records to the table. This form should submit the data to a php script called **`markquiz.php`**.

In Part 1B of the assignment you created some JavaScript to mark quizzes in the client browser. *This has an obvious security flaw*: a user could always spoof their result and send it to the server. Delete or disable the client-side marking script and reproduce this logic in a PHP script on the server. Make sure you do appropriate type-checking of the responses before marking them.

While you will have done client-side validation in Parts 1A & 1B, in order to preserve the integrity of the server data you should also implement server-side data format checking.

In addition to type-checking the quiz responses, check the integrity of the data input by the users. All input data should be *sanitized*. No required fields should be empty.

Field	Format requirement
Employee Id	7 or 10 digits
First name	max 20 alpha|space|hyphen characters
Last name	max 20 alpha|space|hyphen characters

Create a query that gets the number of attempts for a user. If the user already exists in the table and has already had 3 attempts, do not allow them to submit another attempt.

Provided the user has had less than 3 attempts at the quiz, when a valid quiz attempt has been submitted, checked and marked, add a record to the **`attempts`** table you have created.

When the table has had a valid record added, a webpage should be returned from the server that displays:

- The user name and id.
- The score achieved for this attempt.
- The number of attempts the user has made doing the quiz.
- If they have made less than 3 attempts at the quiz, display a hyperlink that allows them to have another attempt at the quiz.

If the quiz submission is not valid (e.g. an answer was not completed; the score was zero; the user has already had three attempts; etc.) use PHP to create a webpage that displays the reason(s) for the failure.

So we can test that server-side validation works correctly, we need to disable client-side HTML5 and JavaScript data checking.

1. Place the **`novalidate="novalidate"`** attribute into your forms.

2. You will need to temporarily disable any validate function(s) within your JavaScript.

   Hint: You can do this by making any *call* to the validate functions conditional. Put them in an **`if`** statement that evaluates a global Boolean variable you create and initialize. e.g.

   ```
   ...
   if (!debug)      {validate()};
   ...
   ```

   Set the flag variable debug to true or false depending on what mode you want to run the code in (or have a check box on the page to set the variable ☺).

4. Quiz supervisor queries.

Create a web page with a link on the menu that allows a supervisor to make the following queries of the **`attempts`** table:

- List all attempts.
- List all attempts for a particular Employee (given an Employee id OR name).
- List all Employees (id, first and last name) who got 100% on their first attempt.
- List all Employees (id, first and last name) got less than 50% on their third attempt.
- Delete all attempts for a particular employee (given an employee id).
- Change the score for a quiz attempt (given an employee id).

Create a php script called **`manage.php`** (and other appropriately named pages if necessary) to implemented access to the supervisor page and its queries.

B: Enhancements

You should complete the* Specified Requirements *before you attempt this part. See the marking Guide below.

Marks will be allocated to enhancements of your choice that go beyond the specified requirements. In this assignment we will consider PHP and MySQL enhancements. You are encouraged to be creative in thinking up possible enhancements.

Examples of PHP / MySQL enhancements you might make that will contribute a higher mark include:

- Store quiz questions in a database table and have the HTML dynamically created by PHP. Use this table to also store correct answers so responses can be marked without hard-coding.
- Normalise the structure of the data tables by, for example, creating separate `employee` and `attempt` tables. Creating a primary-foreign key link between these tables.
- Create a table that stores unique user-ids and passwords for quiz supervisors. Access to the supervisor web page should only be granted if a correct user name and password are entered.
- Provide more secure access to the supervisor page. Have access to the website disabled for user a period of time on, say, three or more invalid login attempts. Create a log out page with a link from the manage web page. Ensure the supervisor's webpage cannot be entered directly using a URL after logging out.
- Provide the quiz supervisor with the ability to select the field on which to sort the order in which the quiz attempt records are displayed.
- One or more enhancements of your own devising. If you plan such enhancements it would be worthwhile checking with your tutor first to ensure they are appropriate and non-trivial.

You must have a **PHP Enhancements** page that lists the enhancements you have implemented. For each extension briefly explain:

- how it goes beyond the specified requirements of the assignment
- what does a programmer have to do to implement the feature.

Any enhancements that are not listed on the PHP enhancements page will not be assessed.

A maximum of 2 extensions will be assessed. *Up to* 5 marks will be given per ***documented*** enhancement type. The filename of this page will be **`phpenhancements.html`** (or **`phpenhancements.php`** if it includes php script).

Website Folder Structure and Deployment Requirements

Create a website structured as specified in the previous assignments.

```
Assign2/              You must have this folder – case sensitive!
   index.php
   topic. php
   quiz. php
   enhancements.html
   enhancements2.html
   markquiz.php
   manage.php
   phpenhancements.php
   header.inc
   menu.inc
   footer.inc
    ...other php function and include pages

  scripts/            Folder for your JavaScript
  images/             Folder for images for your page content
  styles/              Folder for style.css other css files
  styles/images/      Folder for images referred to by your css files e.g. background
```

Note: All links to your files should be ***relative***. *Do not use absolute links*, as these links will probably be broken when files are transferred for marking. No marks will be allocated if links are broken.

Assignment Submission:

An electronic copy of your assignment should be submitted through Canvas submission link on or before the deadline.

- Make sure all your files are in the correct folders and compress your root folder with all your sub-folders with HTML, PHP, CSS, JavaScript, and image files into a zip file named “assign2.zip”. Submit this to ESP. When the zip file is decompressed, the entire website should be able to be run from index.html without needing to move any files.
- You can submit more than once through Canvas.
- Note that all deliverables must be submitted as softcopy. There is no need to submit an assignment cover sheet.

Demonstration Procedure:

1. Make sure you attend your allocated lab. You will demonstrate your assignment to the tutor in your allocated Tutorial in Week 13 (session 1). *You must attend this session to receive a mark for this assignment.* If you cannot attend your allocated tutorial due to illness you must provide a copy of the medical certificate to the convenor.
2. Before your demonstration starts
    a. Fill in and sign the Declaration on the Marking Sheet.
    b. Make sure you website is running on Mercury. (Your tutor will check the URL). All demonstrations will be done on Firefox.
    c. Load Web Developer in Firefox. Validate all your **new/altered** web pages for both HTML5 and XML and the results display in separate browser windows.

    Remember: this is HTML generated by your PHP - not the PHP code itself.

    d. Load MySQL Monitor command line client or the phpMyAdmin web interface so you can demonstrate the changes to your table as your demonstration progresses.
3. As you demonstrate your website your tutor will ask you to explain how you have implemented various aspects of it.

In the week following the demonstration tutorial your tutor will mark your source code and documentation.

Mark Sheet – Assignment 2 Marked by: ……

Assessed by demonstration in your tutorial **(total = 50 marks)**

Fill this in before you start

Student number ……. Student name ……..

Signature ……. Date ……

Tutorial Day …… Tutorial Time …….. Tutor Name …….

Declaration: I hereby confirm that none of my assignment files have been changed on Mercury after their submission to Canvas.

Signature ……. Date ……

Marker to complete: File check ☐ *Days late penalty if applicable (10%/day)* **……..**

Prerequisite	**Y/N**
All Essential requirements Assignment Part 1B met.	

Specified Requirements

Requirement	**Comment**	**Mark**
Menus and other common elements imported from include file		/4
attempts table - schema can store the necessary information with appropriate data types		/2
markquiz.php - text data inputs sanitised ☐ (2) - data formats checked using regex (3) First Name☐ Last Name☐ ID ☐ - number of attempts calculated, no more than 3 allowed ☐ (4) - answers not null and marked (1 x 5 = 5) - score correctly calculated ☐ (2) - table auto created if does not exist when accessed ☐ (2) - attempt correctly added to table ☐ (4) - On successful submission of quiz, page displays with required information including conditional retry hyperlink ☐ (4) - Page returned to the user with appropriate error information on ***unsuccessful*** submission (e.g. an answer was not completed; the score was zero; the >= attempts; etc.) ☐ (4)		/30
manage.php (2 each) •HTML page and table well presented ☐ • List all attempts for a student given id ☐ OR name ☐ • List all students (id, first and last name) with: - 100% on their first attempt ☐ - less than 50% on their third attempt ☐ • Delete all attempts for a particular student (given id) ☐ • Change the score for a quiz attempt (given id) ☐		/14
Sub-total		**/50**

PHP Enhancements listed in phpenhancements.html	Adequately described	Mark
		/5
		/5
Total Additions		**/10**

Requirement	Deduction if not met	Deduct
HTML		
- Valid HTML	-8	
- Well-formed XML	-4	
- Meta-data follows in-house standard	-4	
- Html has no Style information embedded	-2	
- HTML form elements follow in-house standard	-4	
- No deprecated elements/attributes used	-2	
- Comments adequate	-2	
PHP		
- Appropriate header comments as per in-house standard	-4	
- Line comments as appropriate	-2	
- Uses only mysqli commands	-4	
website		
- Directory Structure as specified, relative links	-4	
- All third party content acknowledged properly*	-60	
- Other deductions	-60	
Total Code Deductions		

* Note:

- Failure to acknowledge third party code or content *at all* is plagiarism and may result in zero marks for this assessment or other penalties in accord with Swinburne policy.
- Failure to be able to explain you code during the demonstration may result in up to a deduction of up to 100%.

Warning: All code is automatically checked for similarities to other submissions. Do not use *any* code from other students and make sure other 3rd party code is properly acknowledged and hyperlinked from comments within your source code.

A final assignment mark will *not* be provided during the demonstration. All code is inspected after the demonstration by your tutor before a final mark is allocated.

Comments: ……