Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS10011 / COS60004 / COS60007
Creating Web Applications (and Databases)

Assignment Part 3, Semester 1, 2016

Server-Side Programming

Important Dates:

Due Date ESP	**5pm Monday 23 May 2014** (Late submission penalty 10% of total available marks per day)
Demonstration	Your tutorial: Week 12, 23 May 2016

Individual Assignment. Contribution to Final Assessment: 14%

Purpose of the assignment

In this part of the assignment you will further enhance the Web site you developed in Parts 1 and 2. You will extend the functionality of the Web site by creating server-side PHP scripts to process and store the job application 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 for storing, updating and retrieving information from a Web site.

In addition you will create a Web page that allows the Human Resources (HR) manager at the company to view, update and delete applications.

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

Prerequisite

If you fail to meet the essential requirements of Part 2 of the Assignment, ***before*** this assignment is submitted and marked you must demonstrate that you have fixed problems. Note that these fixes will **not** alter the marks you received for Part 2.

It is advisable to get these fixes completed 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 with fixing your Part 2.

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 ESP assessment printout from Part 2.
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 Part 2 of the assignment.

A: Specified Requirements

Use only mysqli commands in this assignment.

1. Use PHP to reuse common elements in your Web site

PHP provides us with techniques to modularise and reuse our web application code. You need to refactor your web pages so that common static HTML elements, such as menu, header and footer, are written in common HTML/PHP files which are then included into your Web pages.

2. Create an EOI table (expressions of interest)

Create a table **`eoi`** in your MySQL database. The information in each attempt record should include the following:

- **`EOInumber`** (auto-generated id)
- Job Reference number
- First name
- Last name
- Address
    - Street address
    - Suburb/town
    - State
    - Postcode
- Email address
- Phone number
- Skills (Note: the data here will depend on your job description. Although there are better ways to design this, you could just have a number of generic fields called skill1, skill2, ... etc.)
- Other skills. Text description

In addition to the above information, each record should have a **Status** field. The values in this field can **New, Current** or **Final**. When a EOI record is first created the **Status** is set to **New**.

When a user accesses the site, if an EOI table does not already exist in your database it should be programmatically created by your code.

3. Adding validated records to the EOI table

Use (or adapt) the application form (**`apply.html`**) you developed in Assignment Part 1 and Part 2 to *add* EOI records to the table. When the database has accepted the expression of interest from the form, a Web page should display a confirmation message with the unique auto-generated **`EOInumber`** to the user.

While you will have done client-side validation in Parts 1 & 2, in order to preserve the integrity of the server data you should also implement server-side data format checking.

Check the integrity of the data input by the users. No required fields should be empty.

Field	Format requirement
Job reference number	exactly 6 alphanumeric characters
First name	max 25 alpha characters
Last name	max 25 alpha characters
Date of birth	dd/mm/yyyy between 15 and 80
Gender	Selected
Street Address	max 40 characters
Suburb/town	max 40 characters
State	One of VIC,NSW,QLD,NT,WA,SA,TAS,ACT
Postcode	exactly 4 digits – matches state
Email address	validate format
Phone number	8 to 12 digits, or spaces
Other skills	not empty if check box selected

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. Because we will still need JavaScript to handle client-side storage, we cannot disable it entirely. 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. HR manager queries

Create a web page that allows a manager to make the following queries of the **`eoi`** table:

- List all EOIs.
- List all EOIs for a particular position (given a job reference number).
- List all EOIs for a particular applicant given their first name, last name or both.
- Delete all EOIs with a specified job reference number
- Chang the Status of an EOI.

The name of this web page should be **`manage.php`**

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 job descriptions in a database table and have the HTML dynamically created by PHP.
- Normalise the structure of the dataset by, for example, creating a primary-foreign key link between the `eoi` and `job_description` tables; `job_description` . and skills, etc. Ensure that integrity of the database is preserved, for example an EOI cannot be created if the job reference number does not exist in the `job_description` table.
- Provide the manager with the ability to select the field on which to sort the order in which the EOI records are displayed.
- Create a manager registration page with server side validation requiring unique username and a password rule, and store this information in a table. Control access to manage.html by checking username and password. Have access to the web site 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 manager's web site cannot be entered directly using a URL after logging out.
- 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).

Web Site Folder Structure and Deployment Requirements

Create a website structured as specified in the previous assignments.

> **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 ESP at https://esp.ict.swin.edu.au 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 "assign3.zip". Submit this to ESP. When the zip file is decompressed, the entire Web site should be able to be run from index.html without needing to move any files.
- You can submit more than once through ESP.
- Note that all deliverables must be submitted as softcopy. There is no need to submit an assignment cover sheet as ESP generates a receipt upon successful submission.

Demonstration Procedure:

1. Make sure you attend your allocated lab. You will demonstrate your assignment to the tutor in your allocated Tutorial in Week 12. *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 web site 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. Display a copy of the submission receipt from ESP (or have a soft copy available on for viewing on your computer).
    e. 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 Web site your tutor will ask you to explain how you have implemented various aspects of it.

After the demonstration tutorial your tutor will mark your source code and documentation.

Mark Sheet – Assignment Part 3 Marked by: ……

Assessed by demonstration in your tutorial

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 ESP.

Signature ……. Date …….

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

Prerequisite	**Y/N**
Errors in Assignment 2 fixed	

Specified Requirements

Requirement	**Comment**	**Mark**
HTML Menus created and other common elements from PHP includes		/5
`EOI` table - schema can store the necessary information		/2
application.php - records added from Web site ☐ (3) - table automatically created if does not exist when accessed ☐ (2) - status added = New ☐ (1) - **`EOInumber`** programmatically generated ☐ (2)		/8
data formats checked at server (1 each) JobRefNo☐; Name☐; Age format ☐; Age range ☐; Address/suburb ☐; State ☐; Pcode ☐; State-pcd match ☐; Email ☐; Phone ☐; Other skills not empty if checked ☐		/11
manage.php (2 each) **-** HTML well presented ☐ - List all EOIs ☐ - List all EOIs for a particular position ☐ - List all EOIs for a particular applicant ☐ - Change the status an EOI for a particular applicant ☐ - Change the status of all EOIs with a specified job reference ☐ - Delete all EOIs with a specified job reference number ☐		/14
Sub-total		**/40**

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

Requirement	Deduction if not met	Deduct
HTML (manage.php and		
- 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	
Web site		
- Directory Structure as specified, relative links	-4	
- All third party content acknowledged properly*	-50	
- Other deductions	-50	
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: ……