



Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS10011
Creating Web Applications

Assignment Part 3, Semester 1, 2019

Server-Side Programming

Important Dates:

Due Date ESP	Week 12 – 8 am, Day of your tutorial (27 – 31 May 2019) (Late submission penalty 10% of total available marks per day)
Demonstration	Your tutorial: Week 12, (27 – 31 May 2019)

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 job application data sent from the Web forms you created in the previous parts of the assignment.
- Create simple MySQL tables for storing, updating and retrieving information from a Web site.
- 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 mysql 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. Rewrite your web pages so that the common static HTML elements such a menu, header and footer are written in common text files that are then “included” back into your web pages. Name the include file(s) with a .inc extension, replace the sections of HTML in your main pages with ‘include’ statements, and rename your main pages with a .php extension, so the php includes will be included.

2. Create a file to store your database connection variables “settings.php”

As you have done in the labs use a PHP include file “**settings.php**” that contains the host, user, password and database name connection variables, and use this in your PHP to connect to your MySQL database on the feenix-mariadb database server.

3. Create an **EOI** table (expressions of interest)

Create a table **eo_i** 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 be **New**, **Current** or **Final**. When an EOI record is first created the **Status** is set to **New**.

4. Adding validated records to the **EOI** table (processEOI.php)

Use (or adapt) the application form you developed in Assignment Part 1 and Part 2 so that the form data is sent to a PHP script (**processEOI.php**) that *adds* an EOI record 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.

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

The “processEOI.php” page, should not be able to be accessed directly by url through a browser. *Hint: check what data has been set and redirect.*

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. All data should be sanitized to remove leading and trailing spaces, backslashes and HTML control characters. No required fields should be empty. If the data does not validate an appropriate user-friendly error page should be displayed to the user.

Field	Format requirement
Job reference number	exactly 5 alphanumeric characters
First name	max 20 alpha characters
Last name	max 20 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 ☺).

5. HR manager queries (manage.php)

Create a web page **manage.php** that allows a manager to make the following queries of the **eo_i** table and returns a web page with the appropriate results.

- 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
- Change the Status of an EOI.

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 `eo`i 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.

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.

Your website folder structure should follow the structure below. All files should be under a folder /assign3.

<code>assign3/</code>	<i>You must have this folder – case sensitive!</i>
<code>index.php</code>	
<code>jobs.php</code>	
<code>apply.php</code>	
<code>about.php</code>	
<code>enhancements.php</code>	
<code>enhancements2.php</code>	
<code>phpenhancements.php</code>	
<code>header.inc</code>	
<code>menu.inc</code>	
<code>footer.inc</code>	
<code>settings.php</code>	
<code>..other php function and include pages</code>	
<code>processEOI.php</code>	
<code>manage.php</code>	
<code>..other php pages</code>	
<code>scripts/</code>	<i>Folder for your JavaScript files</i>
<code>images/</code>	<i>Folder for images for your page content</i>
<code>styles/</code>	<i>Folder for style.css other css files</i>
<code>styles/images/</code>	<i>Folder for images referred to by your css files e.g. background</i>

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.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.

Submission Reminders:

- The code that is assessed in your demonstration **must be identical** to that you submit to ESP. This will be checked before the demonstration.
- Make sure you complete your ESP submission process. You should get a PDF receipt if you have submitted successfully.

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 your 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

Assessed by demonstration in your tutorial

Marked by:

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
processEOI.php - records added from Web site <input type="checkbox"/> (3) - table automatically created if does not exist when accessed <input type="checkbox"/> (2) - status added = New <input type="checkbox"/> (1) - EOInumber programmatically generated <input type="checkbox"/> (2) - Redirect if tried to access directly via URL <input type="checkbox"/> (2)		/10
data sanitised and formats checked at server (1 each) JobRefNo <input type="checkbox"/> ; Name <input type="checkbox"/> ; Age format <input type="checkbox"/> ; Age range <input type="checkbox"/> ; Address/suburb <input type="checkbox"/> ; State <input type="checkbox"/> ; Pcode <input type="checkbox"/> ; State-pcd match <input type="checkbox"/> ; Email <input type="checkbox"/> ; Phone <input type="checkbox"/> ; Other skills not empty if checked <input type="checkbox"/>		/11
manage.php (2 each) - HTML well presented <input type="checkbox"/> - List all EOIs <input type="checkbox"/> - List all EOIs for a particular position (given Ref num) <input type="checkbox"/> - List all EOIs for a particular applicant given firstname, lastname or both <input type="checkbox"/> - Change the status an EOI for a particular applicant <input type="checkbox"/> - Delete all EOIs with a specified job reference number <input type="checkbox"/>		/12
Sub-total		/40

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

Requirement	Deduction if requirement not met	Deduct
HTML and PHP		
- Invalid HTML	-8	
- Meta-data does not follow in-house standard	-4	
- HTML has Style information embedded	-4	
- HTML form elements do not follow in-house standard	-4	
- Deprecated elements/attributes used	-2	
- Comments inadequate	-2	
- Lack of appropriate header comments as per in-house standard	-4	
- Lack of appropriate Line comments	-2	
- Uses other than mysqli commands	-4	
Web site		
- Directory Structure not as specified, absolute links	-4	
- Third party content inadequately acknowledged	-50	
- Other deductions	-50	
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.	- All marks	
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 your 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: