Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS60004
Creating Web Applications

Assignment Part 3, Semester 2, 2021

Server-Side Programming

Important Dates:

Due Date	**10pm, Monday, Week 12** (Late submission penalty 10% of total available marks per calendar day)
Demonstration	Your tutorial: Week 12

Individual Assignment. Contribution to Final Assessment: 40%

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.
- Analyse the accessibility of your website.

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

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. 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 an .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 **`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
- Street address
- Suburb
- State
- Postcode
- Email address
- Phone number
- Skills (the checkbox data)
- Other skills (Text description, the textarea data)

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

In order to 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 **`eoi`** 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 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.php 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. **(If you have a username and password, please include the username and password on the login page, so the markers can login to your website.)**
- 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. 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`.

```
assign3/              You must have this folder – case sensitive!
  index.php
  jobs.php
  apply.php
  about.php
  enhancements.php
  enhancements2.php
  phpenhancements.php
  header.inc
  menu.inc
  footer.inc
  settings.php
  ..other php function and include pages
  processEOI.php
  manage.php
  ..other php pages
  scripts/            Folder for your JavaScript files
  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.

C: Analytical Report on Web Accessibility

This section requires that you analyse the accessibility of the website you built in and write a report to present the results of the analysis.

The accessibility of the website should be analysed against the principles introduced in WCAG 2.0, including:

1. **Perceivable** – Information and user interface components must be presentable to users in ways they can perceive.
2. **Operable** - User interface components and navigation must be operable.
3. **Understandable** - Information and the operation of user interface must be understandable.
4. **Robust** - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Your report must be professionally written (**1000~1500 words**). Table 1 presents the suggested structure for the report and the content for each section of the report.

Table 1. Components of Report and Requirements

Component	Purpose and Content
Title	• Report title • Your name • Student ID
Executive Summary	• Overview of the report • Findings
Introduction	• Website introduction • Objective of the report • Outline of the report's structure
Accessibility Analysis	• Method of analysis • Tools used for analysis • Accessibility analysis of the website • Data and information collected in the analysis • Findings of the analysis • Discussion of findings
Conclusion	• Conclusion drawn from the analysis • Recommendations for improvement
References	• List of reference materials used during analysis
Appendix	• Information that supports but is not essential to the report

The following evaluation tools can help you analyse the accessibility of the website:

- **Chrome**
  - Accessibility Developer Tools
- **Firefox**
  - Accessibility Evaluation Toolbar
  - Accessibility Scripts Toolbar
  - WCAG Contrast Checker
  - Juicy Studio Accessibility Toolbar

- **Online Accessibility Testing**
  - Functional Accessibility Evaluator
  - AMP Express

Referencing

In this report you are expected to comprehensively reference the third-party materials you used during the analysis. Make sure any quotes are clearly indicated.

All references must be in Swinburne Harvard Style.

Resources

There are a large number of resources on the Internet. Practical resources include:

- Web Content Accessibility Guidelines (WCAG) 2.0
- Web Accessibility in Mind
- Web Accessibility at the U-M
- Accessibility and Usability at Penn State

Submission

Submit your assignment in PDF format to Canvas. **Turnitin has been integrated into Canvas. Turnitin allows the marker to identify any similarities between your work and that of other students and from the Internet in general.**

Assignment Submission (Mercury + Canvas)

Research Report: Please submit your research report in a PDF file to Canvas.

Web Application: Your web application should be uploaded to Mercury on or before your deadline. An electronic copy of your assignment should be submitted through Canvas on or before your deadline.

- Make sure all your files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, JavaScript, PHP and images into a zip file named "assign3.zip". Submit this to Canvas. 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. Your last submission will be marked.
- Note that all deliverables must be submitted electronically. There is no need to submit an assignment cover sheet.

Make sure you complete your Canvas submission process.

The code that is assessed in your demonstration ***must be identical*** to that you submit to Canvas.

Mark Sheet – Assignment Part 3 Marked by: ……
Assessed by demonstration in your tutorial

Fill this in before you start

Student number ……. Student name ……..

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 ……

Assignment Part 3 100 marks

Specified Requirements

Requirement	Comment	Mark
HTML Menus created and other common elements from PHP includes		/7
`EOI` table - schema can store the necessary information		/3
processEOI.php - records added from Web site ☐ (6) - table automatically created if does not exist when accessed ☐ (6) - status added = New ☐ (2) - **`EOInumber`** programmatically generated ☐ (4) - Redirect if tried to access directly via URL ☐ (2)		/20
data sanitised and formats checked at server (4 each) JobRefNo; Name; Suburb; ☐; Age format; Age range ☐; State; Postcode; State-postcode match ☐; Email; Phone; ☐; Other skills not empty if checked ☐		/20
manage.php (4 each) - List all EOIs ☐ - List all EOIs for a particular position (given Ref num) ☐ - List all EOIs for a particular applicant given firstname, lastname or both ☐ - Change the status of EOI for a particular applicant ☐ - Delete all EOIs with a specified job reference number ☐		/20
Sub-total		**/70**

Enhancements

Enhancements listed in phpenhancements.html/php	Adequately described	Mark
		/10
		/10
Total Additions		**/20**

Research Report

Research Report	Mark
Comments:	/10

Other Deductions *based on demonstration, documentation, code and file inspection*

Requirement	**Max Deduction** if requirements not met	**Deduct**
HTML *(deduct up to 4 marks each)* -- Deprecated elements/attributes have been used - Inappropriate use of HTML semantics (e.g. use of when <section> <article> should be used) - HTML usability does not follow standards (e.g. alt on images, label in forms, tables) - HTML has embedded Style markup. CSS is not fully separated from HTML - Code comments inadequate to inform later code understanding/ maintenance	-20	
PHP *(deduct up to 4 marks each)* - Inappropriate header comments, do not match in-house standard. - Code comments inadequate to inform later code understanding/ maintenance - Uses only mysqli commands	-12	
Web site - Directory and file structure not as specified - Third party content inadequately acknowledged *- 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.*	-4 -4 - All marks	
Total Deductions		

Research Report Assessment Criteria

Grade	Criteria	Mark
Excellent	• Full understanding of Web accessibility. • All required contents are covered. • Report is presented in an excellent manner.	10
Very good	• Thorough knowledge of Web accessibility. • All required contents are covered; • Report is presented in a proper manner.	8
Average	• Average comprehension of Web accessibility. • Most required contents are covered. • Report is presented in an acceptable manner.	5
Unsatisfactory	• Inadequate understanding of Web accessibility. • Most required contents are not covered. • Report is of poor quality.	3
Failing	• Failure to submit assignment, or • Poor understanding of Web accessibility. • Report is of extremely poor quality.	0

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.