



## Swinburne University of Technology

School of Science, Computing and Emerging Technologies

### COS10026 Web Technology Project

Project Part 2, Semester 1,  
2025 Server-Side & Dynamic  
Web Programming

#### Important Information

|   |  |
|---|--|
| <b>Due Date</b>                         | <b>11 pm on Monday in Week 12</b><br>(Late submission penalty: 10% of total available marks per day) |
| <b>Submission Method</b>                | <b>Canvas + Mercury</b>  |
| <b>Contribution to Final Assessment</b> | 50% (70% Group Tasks, 30% Individual Tasks)  |

#### Purpose of the assignment

In this part of the assignment you will further enhance the Web site you developed in Parts

1. 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.

#### Group Tasks: A, B, C , & D

##### A: Specified Requirements

###### 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 another 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 local 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
- Address:
  - Street address
  - Suburb/town
  - State
  - Postcode
- Email address
- Phone number
- Skills (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 (process\_eoi.php)

Use (or adapt) the application form what you developed in Project Part 1 so that the form data is sent to a PHP script (`process_eoi.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 created by your code. The “process\_eoi.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, 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 | Drop down selection including all jobs ref numbers |
| First name           | max 20 alpha characters                            |
| Last name            | max 20 alpha characters                            |
| Date of birth        | dd/mm/yyyy   |
| Gender               | radio inputs grouped using a fieldset and legend   |
| Street Address       | max 40 characters                                  |
| Suburb/town          | max 40 characters                                  |

|                               |                                  |
|-------------------------------|----------------------------------|
| State                         | 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        |
| Required Technical Skill List | Checkbox input                   |
| 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 data checking.

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

## 5. HR manager queries (manage.php)

Create a web page **manage.php** that allows a manager to make the following queries of the **eois** 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.

## 6. Update about page (about.php)

Update members contribution on the about page.

## 7. Jobs Description

Create a table named jobs in your MySQL database. Store job descriptions in a database table and have the HTML dynamically created by PHP.

## 8: Enhancements

You should complete the Specified Requirements before you attempt this part. Up to 8 marks will be given per documented enhancement type. The filename of this page will be enhancements.php. In this page you need to list and explain what you have done from the following options.

### Options:

- 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.

## B: Web Site Folder Structure and Deployment Requirements

Create a website structured as specified in the project part 1. Your website folder structure should follow the structure below

```
project2/                                You must have this folder – case sensitive!
index.php
jobs.php
apply.php
about.php
enhancements.php
header.inc
nav.inc
footer.inc
settings.php
process_eoi.php
manage.php
..other html and php pages
```

```
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: Project Group Presentation (Assessed individually)

For this presentation, you will showcase your group project to your classmates in 10 minutes during week 12 class. The presentation slides will be uploaded with the group submission but the presentation will be assessed individually, and each student is expected to cover the following points:

**Introduction:** Each member should present a section of the introduction, focusing on their own contribution to the project's purpose and objectives.

**Development Process:** Each student will discuss a particular technology they worked with during development.

**Database Interaction:** Each member should explain how their part of the website interacts with the database (e.g., a team member responsible for job listings explains that interaction).

**Conclusion:** Each member should briefly summaries what they learned from the project, focusing on their individual contributions and experiences.

## D: Peer Review & Feedback (Individual Task)

Every student needs to complete the peer and self-evaluation. This task is crucial for providing constructive feedback to your peers and reflecting on your own contribution. The link can be found in the assessments page under "Project Part 2". The Peer & Self Evaluation results will be reported to each member anonymously to provide an opportunity for improvement. You must provide respectful and constructive feedback. This should be done individually.

## Assignment Submission (Canvas & GitHub)

The marks are allocated 70% for team task and 30% for individual task in this assignment.

Deliverables:

- The web application (Group task - Codes, SQL file, GitHub repository link and presentation slides)
- The presentation (Delivered in a group but assessed individually - 30%)
- Peer & self evaluation from (Individual task)

Make sure all your website files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, PHP, db file and images into a zip file named "groupName\_part2.zip". **Submit this to Canvas by your group leader into "Portfolio Part 2 (Group Submission)".** the entire website should be able to be run from index.php without needing to move any files.

Every student needs to present their part in the project during week 12. **Failure to attend this class will result in a score of zero for the presentation mark.**

Every student needs to do their peer and self evaluation in Canvas.

You can submit more than once through Canvas. Your last submission will be marked.

Note that all deliverables must be submitted electronically (except doing the presentation in class). **Make sure you complete your Canvas submission process.**

### Submission Summary:

- Demonstration of your website to your tutor during Week 11.

Project Part 2 (Group Submission):

- Zip file named groupName\_part2.zip
- Presentation Slides
- GitHub repository link as a submission comment.

Project Part 2 (Individual Submission):

- Presentation
- Peer & Self Evaluation