# COS60004 Creating Web Applications Part-3

By Ethan PP Cutting – 100942775

**Due Date:** 10 am on Thursday in week 12

**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.

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

# Contents

## Executive Summary

The purpose of this assignment is to enhance a web application build off parts 1 and 2 assignment by adding server-side PHP scripts to process job application data, creating MySQL tables to store and retrieve information, and as well developing a web page for the HR manager to view, update, and delete job applications.

The finding I found was how to connect to a database using PHP scripts, how to build a table inside MariaDB, and call from the table so the user can pull data from the tables or change and delete from the EOIs table.

## Introduction

The U.I.D job application website has been enhanced to streamline the job application process and improve HR management. We incorporated server-side PHP scripts and a MySQL database with an EOI table. Additionally, a dedicated HR manager interface was developed to simplify applicant management. This report highlights the implemented features, emphasizing their impact on user experience and efficient HR operations.
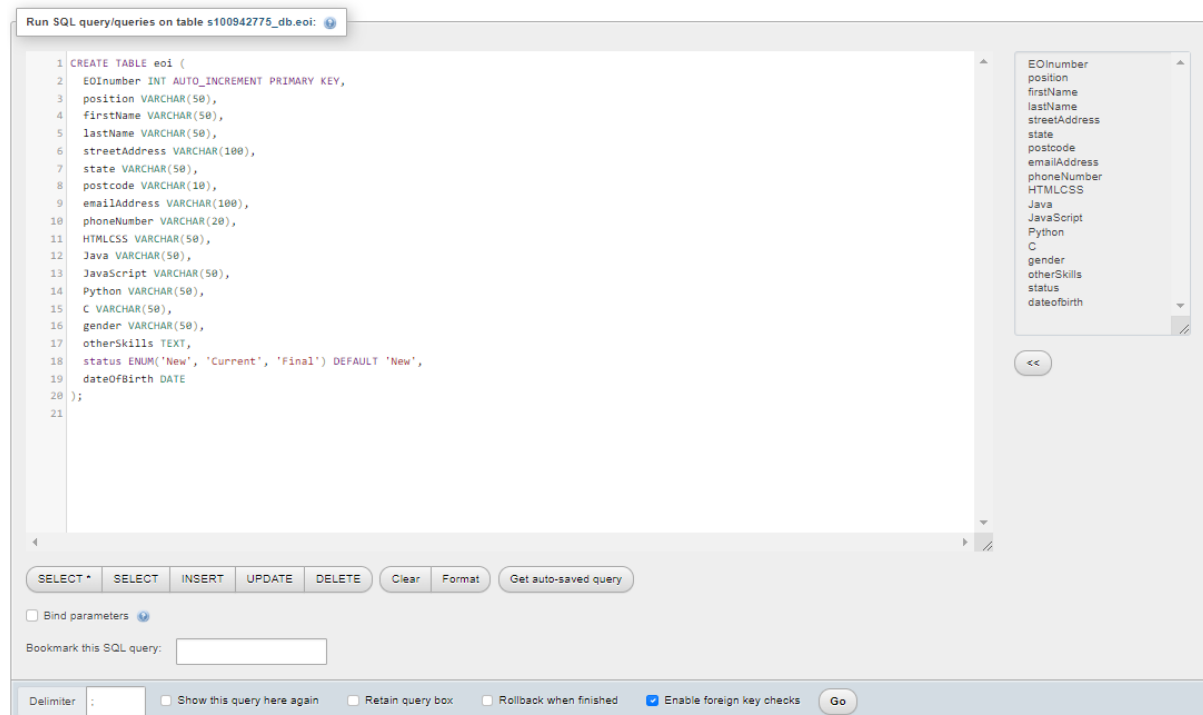
## Accessibility Analysis

### Method of analysis

The analysis for this assignment was going through lab07 – PHP Basic, Lab08 – Processing form data with PHP, Lab09 – Databases, MySQL, and PHP, and well Lab10 – Managing State and SQL Injection. By doing this lab it gave me the knowledge to finish this assignment.

### Tools used for analysis.

I developed the website using Visual Studio Code, PHP for server-side scripting, and MariaDB (MySQL) as the database management system. PhpMyAdmin is a great tool and user-friendly MySQL database management. Here's a concise summary of creating the EOI table using PhpMyAdmin.



To test the web page, you need to connect with Cisco any connect secure mobility client.
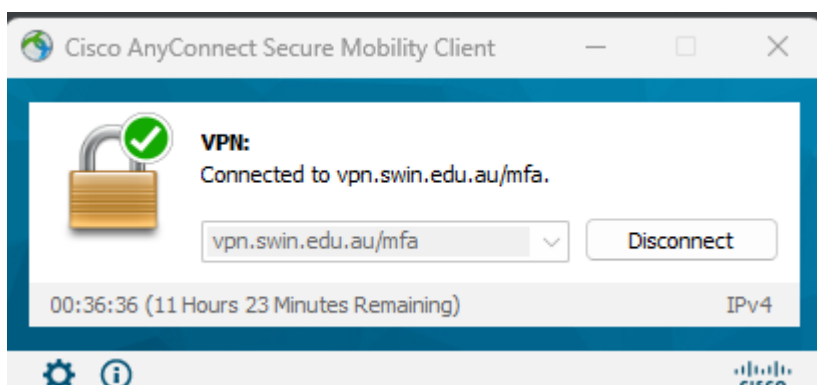
Connecting to WinSCP to upload your folders to the server you will log in like this.
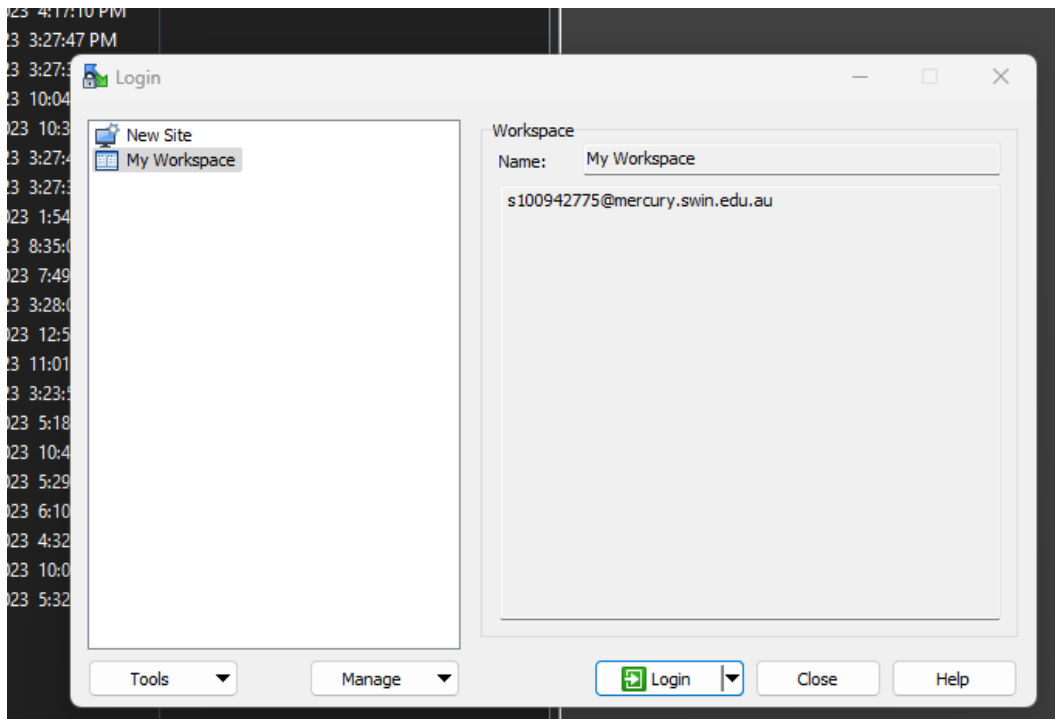


To test the server the web browser I used Firefox.
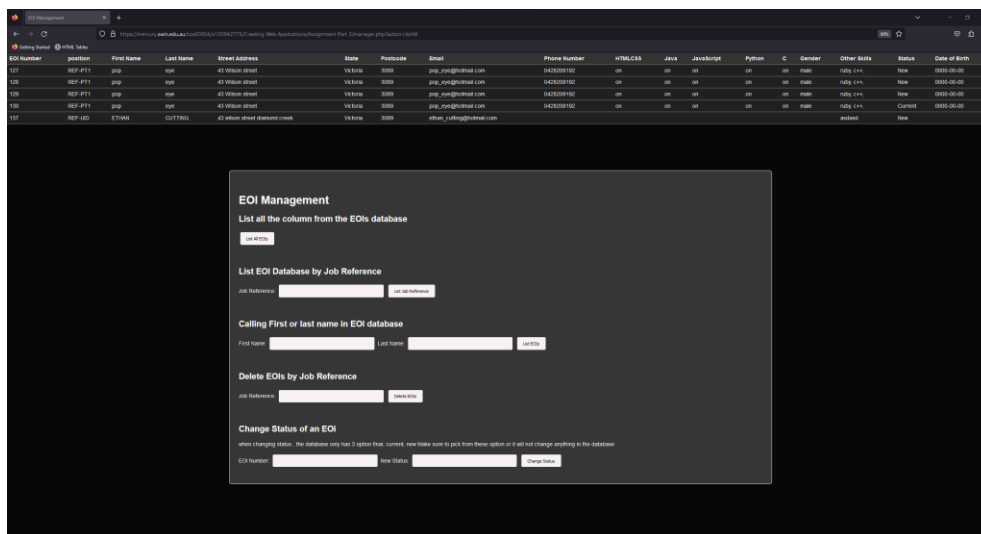


*Accessibility analysis of the website.*

Accessibility of U.I.D website is very straight forwarded you start on the index page and the user will have the option to access all the pages with the top nav bar or there are 3 images with a herf link that will take the user to the about us page, apply page, and jobs pages.

Testing different browsers overall Firefox and Google Chromes as the website is compatible with both web browsers as well desktops. As it is accessible on all devices. Checking as well for keyboard accessibility by testing the web pages that are accessed and operated by using a keyboard alone.

*Data and information collected in the analysis.*

The user will go to apply page from the index page and then enter the contact info and Education of the applicant here is an example:

Then after filling out the information of the user and then click apply button and it will send to the EOI table and the HR manager is able to pull up the applicant.

The HR manager will be able to delete an applicant by job reference.

If the HR manager is trying to find an applicant, then they can pull the applicant up by first name or last name.

The HR manager will be able to pull up applicants by Job Reference in the EOI table.

The HR manager can update the applicant's status in the EOI table when a new application is received.



The HR manager has three options: new, current, and final. If none of these options are chosen, no changes will be made to the database table.

**ProcessEOI.php:**

**Data Sanitization:** inside my code, it will include 'sanitise_input()' this function will perform basic data sanitization by removing leading/trailing spaces, and backslashes and converting special characters to HTML entities using 'htmlspecialchars()'. This will help with stopping security vulnerabilities like SQL injection and as well cross-site scripting (XSS) attacks.

**Form Data Retrieval:** using '$_POST' super global array to retrieve the form data submitted via the HTTP POST method it will check if the required fields for an example '$_POST["fname"]') will set and assigns them to variables after sanitizing the input.

**Checkbox Values:** This code will checks if checkboxes, for example, HTMLCSS,Java, JavaScript, or Python will check using the 'isset()' function and assigns their values accordingly to the database.

**Database Connection:** This script establishes a database connection by using 'mysqli_connect()' by using the host, username, password, and database name by adding this it will check if the connection was successful before proceeding with further operations.

**SQL Query and Insertion**: Create an SQL query using the retrieved form data and inserts it into the 'eoi' tables by using the 'mysqli_query()' function. Then the query uses the 'VALUES' clause to insert the form data into the

**Query Execution:** will check if the query execution was a success by using the returned result from 'mysqli_query()'. If it is successful it will display a success message and if it's not successful an error message will display with the specific error by 'mysqli_error()'.

**Manager.php:**

**Database Connection:**

**Conditional Actions:** There are several conditional blocks that will use different actions based on the different values of the 'action' parameter and will pass through GET or POST requests.

**List All EOIs:** When the list button is pressed, the code will then run a query to retrieve all rows from the EOI table.

**List EOIs by Job Reference:** When the list job reference button triggers a query that will pull based on the specific job reference number that was entered.

**List EOIs by User First Name and Last Name:** This will allow the user to enter their first name or last name and pull the user they are looking for from the EOIs table.

**Delete EOIs Status:** This will enable deleting EOIs based on a job reference number for example REF-PT1 OR REF-UID.

**Change EOI Status:** The change status will allow the HR manager to update the status of the applicant in the EOI table.

**Execute Query Function:** The 'executeQuery()' function is called to execute the query and it will display the results in an HTML table.

**HTML and CSS:** using HTML to build the form for the buttons and inputs and using CSS to style the forms and field sets.

**Settings.php:**

**Database Connection settings:** these will include variables that will hold the necessary database connection settings, like the host, username, password, and the database name in MariaDB.

**Connection Establishment:** will use the 'mysqli_connect()' function this will connect to the MySQL database server.

**Connection Check:** Will look after if the code checks the connection status by using the 'if(!$conn)' and if the connection fails, then the script will include an error message using 'die()' and as well 'mysqli_connect_errror()' function and this will give the user more details about the connection error.

**Error Handling:** will look after the case where the database connection fails by using 'die()' this will complete the script and then it will display the error message.

## Conclusion

In conclusion, the U.I.D job application website has been enhanced with PHP scripts, MySQL integration, and an HR manager interface, improving efficiency and user experience. These additions ensure secure data handling and streamline the job application process, benefiting both applicants and HR personnel.

Reference

https://swinburne.instructure.com/courses/49148/pages/week-8-php-part-1?module_item_id=3155323

https://swinburne.instructure.com/courses/49148/pages/week-9-php-part-2?module_item_id=3155324

https://swinburne.instructure.com/courses/49148/pages/week-10-php-and-mysql-part-1?module_item_id=3155325

https://swinburne.instructure.com/courses/49148/pages/lab-10-managing-state-and-sql-injection?module_item_id=3155337

https://www.w3schools.com/php/php_mysql_connect.asp

https://www.geeksforgeeks.org/connect-php-to-mysql/