

CO551 Open Source Systems



CW2: COMPLETE REPORT

By: Martin Konecny
Student ID: 100128860 / 22045464



Table Of Contents

Youtube Presentation	3
Task 1 - Seeding 5 Students Into Database	3
Task 2 - Displaying Students Within HTML Table	6
Task 3 - Added Checkboxes & Deletion System	7
Task 4 - Adding New Student Form	11
Task 5 - Uploading Images To Students	14
Task 6 - Improving The Interface Using CSS	18
Login Page	18
Landing Page	21
My Modules Page	24
Assign Module Page	26
My Details Page	28
Students Page	31
Add Student Page	34
Task 7 - Protecting SQL From SQL Injections	39
Task 8 - Evaluation & Research	42
Part A - 2 Different Server-Side Approaches To Development	42
PHP	42
ASP.NET Core	42
Conclusion	42
Part B - Security Considerations In Personal Data Management Systems	43
Cross-Site Scripting (XSS)	43
Insider Threat	43
References:	44



Youtube Presentation

https://youtu.be/TGz6q_Lk_D0

Task 1 - Seeding 5 Students Into Database

In this task I have seeded 5 different students into a database. I did not use faker as I was unable to install all the necessary components into my machine as the tutorial on blackboard has been deleted. I have added 5 different students in the back-end and when I ran studentseed.php on my web browser, all 5 students have been automatically inserted in the database I have created.

Output

Record for student ID 3105 inserted successfully.
Record for student ID 1702 inserted successfully.
Record for student ID 2307 inserted successfully.
Record for student ID 0104 inserted successfully.
Record for student ID 0505 inserted successfully.

Output screenshot of studentseed.php being ran in a web browser. All 5 students have been seeded in the database successfully.



PhpMyAdmin

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=co551cw2&table=student

Server: 127.0.0.1 » Database: co551cw2 » Table: student

Browse Structure SQL Search Insert Export Import Privileges Operations Track

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

SELECT * FROM `student`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	studentid	password	dob	firstname	lastname
<input type="checkbox"/>	0104	\$2y\$10\$B421337/fnFDx9/Udwge0u3lsdUPTZ6Jx.pE5F6E18r...	2003-04-01	Nino	Nakano
<input type="checkbox"/>	0505	\$2y\$10\$.CQIYZajJOGaJeeNMhA7kOngn13ljkR.YE29VjNYaNX...	1999-05-05	Salma	Hayek
<input type="checkbox"/>	1702	\$2y\$10\$92J4rPLlp4SCKmwoG2Z9OBWpRDZc5tztNn5CAiQ/Hy...	1997-02-17	Greg	Jenko
<input type="checkbox"/>	20000000	\$2y\$10\$.LJBOl64nZWEVVE/v5mgNuzR01zx1zoyXuGJUa/zp2U...	1974-11-10	Jon	Smith
<input type="checkbox"/>	2307	\$2y\$10\$Qp9XzUlPB4kkn791aZ07w.kG8Op7cSYeZLfZpmOAt3G...	2002-07-23	Joey	Bezigher
<input type="checkbox"/>	3105	\$2y\$10\$uqY5rPwX8RZi6By.NktWVeh3p7ZxFopAgCv5aGhIPD...	1998-05-31	Martin	Konecny

Screenshot from PhpMyAdmin which shows 5 students that have been automatically seeded in the database. Passwords are hashed as well for robust security.



Code

```
index.php studentseed.php students.php

studentseed.php
1 <?php
2 // these three lines include necessary configuration, database connection, and functions files
3 include("_includes/config.inc");
4 include("_includes/dbconnect.inc");
5 include("_includes/functions.inc");
6
7 // student details which will be seeded to the database. Each student is represented as an associative array
8 $students = [
9     ['studentid' => '3105', 'password' => password_hash('test', PASSWORD_DEFAULT), 'dob' => '1998-05-31', 'firstname' => 'Martin',
10    ['studentid' => '1702', 'password' => password_hash('test', PASSWORD_DEFAULT), 'dob' => '1997-02-17', 'firstname' => 'Greg',
11    ['studentid' => '2307', 'password' => password_hash('test', PASSWORD_DEFAULT), 'dob' => '2002-07-23', 'firstname' => 'Joey',
12    ['studentid' => '0104', 'password' => password_hash('test', PASSWORD_DEFAULT), 'dob' => '2003-04-01', 'firstname' => 'Nino',
13    ['studentid' => '0505', 'password' => password_hash('test', PASSWORD_DEFAULT), 'dob' => '1999-05-05', 'firstname' => 'Salma'],
14];
15
16 // This block of code loops through each student in an array. Creates an SQL INSERT query for the current student's details
17 foreach ($students as $student) {
18     $sql = "INSERT INTO student (studentid, password, dob, firstname, lastname, house, town, county, country, postcode) VALUES (
19         '{$student['studentid']}',
20         '{$student['password']}',
21         '{$student['dob']}',
22         '{$student['firstname']}',
23         '{$student['lastname']}',
24         '{$student['house']}',
25         '{$student['town']}',
26         '{$student['county']}',
27         '{$student['country']}',
28         '{$student['postcode']}'
29     );";
30
31     // Execute the SQL query and check if it was successful, displays either success message or an error message
32     if (mysqli_query($conn, $sql)) {
33         echo "Record for student ID {$student['studentid']} inserted successfully.<br>";
34     } else {
35         echo "Error inserting record for student ID {$student['studentid']}: " . mysqli_error($conn) . "<br>";
36     }
37 }
38
39 // closes data base connection
40 mysqli_close($conn);
41 ?>
42
```

This is the commented code, I have also included an error message as a form of validation (see task 3 screenshots) **The error message is also previewed in video presentation.**



Task 2 - Displaying Students Within HTML Table

In this task I have successfully created a new php file called students.php which adds all of the student records within the database inside a well formatted html table and displays it as long as the user is logged in (if they are not logged in, they will be redirected to login page). The order of each field and column is identical to the table within the database.

Output

Student ID	First Name	Last Name	Date of Birth	House	Town	County	Country	Postcode
0104	Nino	Nakano	2003-04-01	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ
0505	Salma	Hayek	1999-05-05	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH
1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS
20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT
2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE
3105	Martin	Konecny	1998-05-31	6 Bellevue Close	Nelson	Lancashire	United Kingdom	BB9 8TX

Code

```
EXPLORER index.php studentseed.php students.php
CO551CW2 _includes
config.inc dbconnect.inc functions.inc
passwordLib.php passwordLibClass.php
_sql college.dump
css
img
js
templates
partials
default.php
login.php
assignmodule.php
authenticate.php
details.php
index.php
logout.php
modules.php
password_hash.php
README.md
students.php
studentseed.php

index.php
studentseed.php
students.php

1 <?php
2     include("_includes/config.inc");
3     include("_includes/dbconnect.inc");
4     include("_includes/functions.inc");
5
6     // This block of code checks if the admin / user is logged in
7     // if the user isn't logged in, they will be redirected to login page
8     // they must be logged in to view table with all the students.
9     if (isset($_SESSION['id'])) {
10         echo template("templates/partials/header.php");
11         echo template("templates/partials/nav.php");
12
13         $sql = "SELECT * FROM student";
14         $result = mysqli_query($conn, $sql);
15
16         // Check if any records were returned from the database query
17         if (mysqli_num_rows($result) > 0) {
18             $data['content'] = "<h2>Student Records</h2>";
19             $data['content'] .= "<table border='1'>";
20             $data['content'] .= "<tr>
21                             <th>Student ID</th>
22                             <th>First Name</th>
23                             <th>Last Name</th>
24                             <th>Date of Birth</th>
25                             <th>House</th>
26                             <th>Town</th>
27                             <th>County</th>
28                             <th>Country</th>
29                             <th>Postcode</th>
30                         </tr><tr>
31                         <td>0104</td>
32                         <td>Nino</td>
33                         <td>Nakano</td>
34                         <td>2003-04-01</td>
35                         <td>45 King Street</td>
36                         <td>Manchester</td>
37                         <td>Greater Manchester</td>
38                         <td>United Kingdom</td>
39                         <td>M2 4LQ</td>
40                     </tr><tr>
41                         <td>0505</td>
42                         <td>Salma</td>
43                         <td>Hayek</td>
44                         <td>1999-05-05</td>
45                         <td>67 High Street</td>
46                         <td>Aylesbury</td>
47                         <td>Buckinghamshire</td>
48                         <td>United Kingdom</td>
49                         <td>HP20 1SH</td>
50                     </tr><tr>
51                         <td>1702</td>
52                         <td>Greg</td>
53                         <td>Jenko</td>
54                         <td>1997-02-17</td>
55                         <td>20 High St</td>
56                         <td>Manchester</td>
57                         <td>Greater Manchester</td>
58                         <td>United Kingdom</td>
59                         <td>M14 4SS</td>
60                     </tr><tr>
61                         <td>20000000</td>
62                         <td>Jon</td>
63                         <td>Smith</td>
64                         <td>1974-11-10</td>
65                         <td>23 Victoria Road</td>
66                         <td>High Wycombe</td>
67                         <td>Bucks</td>
68                         <td>UK</td>
69                         <td>HP11 1RT</td>
70                     </tr><tr>
71                         <td>2307</td>
72                         <td>Joey</td>
73                         <td>Bezigher</td>
74                         <td>2002-07-23</td>
75                         <td>123 Baker Street</td>
76                         <td>London</td>
77                         <td>Greater London</td>
78                         <td>United Kingdom</td>
79                         <td>NW1 6XE</td>
80                     </tr><tr>
81                         <td>3105</td>
82                         <td>Martin</td>
83                         <td>Konecny</td>
84                         <td>1998-05-31</td>
85                         <td>6 Bellevue Close</td>
86                         <td>Nelson</td>
87                         <td>Lancashire</td>
88                         <td>United Kingdom</td>
89                         <td>BB9 8TX</td>
90                     </tr>"
```



```
32     // Fetch each record as an associative array
33     while ($row = mysqli_fetch_assoc($result)) {
34         $data['content'] .= "<tr>
35             <td>$row['studentid']</td>
36             <td>$row['firstname']</td>
37             <td>$row['lastname']</td>
38             <td>$row['dob']</td>
39             <td>$row['house']</td>
40             <td>$row['town']</td>
41             <td>$row['county']</td>
42             <td>$row['country']</td>
43             <td>$row['postcode']</td>
44         </tr>";
45     }
46
47     $data['content'] .= "</table>";
48 } else {
49     $data['content'] = "<p>No student records found.</p>";
50 }
51
52 // Outputs the final table by using a template to render the page
53 echo template("templates/default.php", $data);
54 } else {
55     header("Location: index.php");
56 }
57
58 echo template("templates/partials/footer.php");
59 ?>
60
```

Task 3 - Added Checkboxes & Deletion System

In this task I have improved the student records table by adding checkboxes into each row where the user can select a student or multiple students and have them deleted. I have also implemented validation, if the user selects a student and clicks on delete. Rather than have the student record be deleted straight away a pop up box will appear which will ask the user if they are sure about wanting to delete the student.

Output

The screenshot shows a web browser window with the URL `localhost/CO551CW2/students.php`. The page title is "Student Records". Below the title is a table with columns: Select, Student ID, First Name, Last Name, Date of Birth, House, and Postcode. There are 10 rows of data. In the first two rows, the "Select" column contains checked checkboxes. A modal dialog box is overlaid on the page, containing the text "localhost says" and "Are you sure you want to delete the selected students?". At the bottom right of the dialog are "OK" and "Cancel" buttons.

Select	Student ID	First Name	Last Name	Date of Birth	House	Postcode			
<input checked="" type="checkbox"/>	0104	Nino	Nakano	2003-04-01	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ
<input checked="" type="checkbox"/>	0505	Salma	Hayek	1999-05-05	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH
<input type="checkbox"/>	1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS
<input type="checkbox"/>	20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT
<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE
<input type="checkbox"/>	3105	Martin	Konecny	1998-05-31	6 Bellevue Close	Nelson	Lancashire	United Kingdom	BB9 8TX

Delete Selected

In this screenshot I have selected students “0104” and “0505” and clicked on delete. Rather than the students being deleted a pop up window was opened which asks the user if they are sure about deleting selected students.



localhost/CO551CW2/students.php

[My Modules](#) | [Assign Module](#) | [My Details](#) | [Logout](#)

Student , ID 0104 has been successfully deleted!
Student , ID 0505 has been successfully deleted!

Student Records

Select	Student ID	First Name	Last Name	Date of Birth	House	Town	County	Country	Postcode
<input type="checkbox"/>	1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS
<input type="checkbox"/>	20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT
<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE
<input type="checkbox"/>	3105	Martin	Konecny	1998-05-31	6 Bellevue Close	Nelson	Lancashire	United Kingdom	BB9 8TX

[Delete Selected](#)

Once the user confirms that they are sure about wanting to delete the students, a message will appear saying the students have been successfully deleted from the database. The table will also be updated and appear without the students.

PhpMyAdmin

Server: 127.0.0.1 » Database: co551cw2 » Table: student

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#)

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

```
SELECT * FROM `student`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	studentid	password	dob	firstname	lastname
<input type="checkbox"/>	1702	\$2y\$10\$92J4rPLIp4SCKmwoGZ9OBWpRDZc5tztNn5CAiQ/Hy...	1997-02-17	Greg	Jenko
<input type="checkbox"/>	20000000	\$2y\$10\$LJBOl64nZWEVVE/v5mgNuR01zx1zoyXuGUJa/zp2U...	1974-11-10	Jon	Smith
<input type="checkbox"/>	2307	\$2y\$10\$Qp9XzUIpB4kkn791aZ07w.kG8Op7cSYeZLfZpmOAt3G...	2002-07-23	Joey	Bezigher
<input type="checkbox"/>	3105	\$2y\$10\$uqY5rPwX8RZi6ByNktWVeh3p7ZxFopAgCv5aGhIPD...	1998-05-31	Martin	Konecny

[Edit](#) [Copy](#) [Delete](#) With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

This screenshot shows the database after two students were deleted.



Seeding removed students back

< > ⌂ localhost/CO551CW2/studentseed.php

Error inserting record for student ID 3105: Duplicate entry '3105' for key 'PRIMARY'
Error inserting record for student ID 1702: Duplicate entry '1702' for key 'PRIMARY'
Error inserting record for student ID 2307: Duplicate entry '2307' for key 'PRIMARY'
Record for student ID 0104 inserted successfully.
Record for student ID 0505 inserted successfully.

Back to task 1, since I have deleted the students, I have seeded them back. This screenshot also shows an error message which says 3 students with primary key (ID) already exist so they were not added to avoid duplication.

Code

```
index.php studentseed.php students.php X deletedstudents.php

students.php
1
2 <?php
3
4     include("_includes/config.inc");
5     include("_includes/dbconnect.inc");
6     include("_includes/functions.inc");
7
8     // This block of code checks if the admin / user is logged in
9     // if the user isn't logged in, they will be redirected to login page
10    // they must be logged in to view table with all the students.
11    if (isset($_SESSION['id'])) {
12        echo template("templates/partials/header.php");
13        echo template("templates/partials/nav.php");
14
15        // Display feedback message
16        if (isset($_SESSION['delete_message'])) {
17            echo "<p>{$_SESSION['delete_message']}</p>";
18            unset($_SESSION['delete_message']); // Clear message after displaying
19        }
20
21        $sql = "SELECT * FROM student";
22        $result = mysqli_query($conn, $sql);
23
24        // Check if any records were returned from the database query
25        if (mysqli_num_rows($result) > 0) {
26            $data['content'] = "<h2>Student Records</h2>";
27            $data['content'] .= "<form action='deletedstudents.php' method='post'>";
28            $data['content'] .= "<table border='1'>";
29            $data['content'] .= "<tr>";
30                $data['content'] .= "<th>Select</th>";
31                $data['content'] .= "<th>Student ID</th>";
32                $data['content'] .= "<th>First Name</th>";
33                $data['content'] .= "<th>Last Name</th>";
34                $data['content'] .= "<th>Date of Birth</th>";
35                $data['content'] .= "<th>House</th>";
36                $data['content'] .= "<th>Town</th>";
37                $data['content'] .= "<th>County</th>";
38                $data['content'] .= "<th>Country</th>";
39                $data['content'] .= "<th>Postcode</th>";
40            $data['content'] .= "</tr>";
```



```
41     // Fetch each record as an associative array
42     while ($row = mysqli_fetch_assoc($result)) {
43         $data['content'] .= "<tr>
44             <td><input type='checkbox' name='students[]' value='{$row['studentid']}'></td>
45             <td>{$row['studentid']}</td>
46             <td>{$row['firstname']}</td>
47             <td>{$row['lastname']}</td>
48             <td>{$row['dob']}</td>
49             <td>{$row['house']}</td>
50             <td>{$row['town']}</td>
51             <td>{$row['county']}</td>
52             <td>{$row['country']}</td>
53             <td>{$row['postcode']}</td>
54         </tr>";
55     }
56
57     $data['content'] .= "</table>";
58     $data['content'] .= "<input type='submit' value='Delete Selected' onclick='return confirm(\"Are you sure you want to delete the selected students?\")';>";
59     $data['content'] .= "</form>";
60 } else {
61     $data['content'] = "<p>No student records found.</p>";
62 }
63 // Outputs the final table by using a template to render the page
64 echo template("templates/default.php", $data);
65 } else {
66     header("Location: index.php");
67 }
68
69 echo template("templates/partials/footer.php");
70 ?>
71
```



Task 4 - Adding New Student Form

In this task I have created a simple HTML form where the user can add a new student into the database, the user can also create login details for the student which are stored inside the database. The password that is going to be set for the student will be hashed in the database for extra security. **Validation is showcased in the video presentation.**

Output

< > ⌂ ↗ localhost/CO551CW2/addstudent.php

Add Student

Student ID:

Password:

Date of Birth:

First Name:

Last Name:

Address:

Town/City:

County:

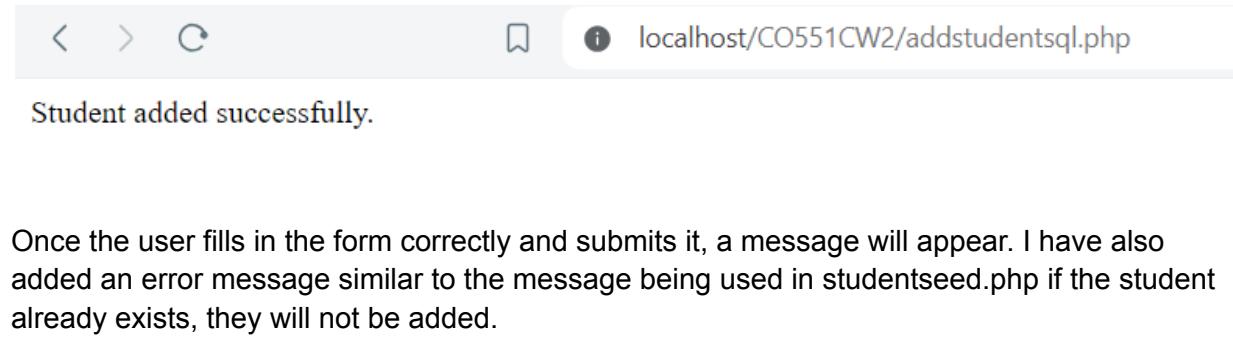
Country:

Postcode:

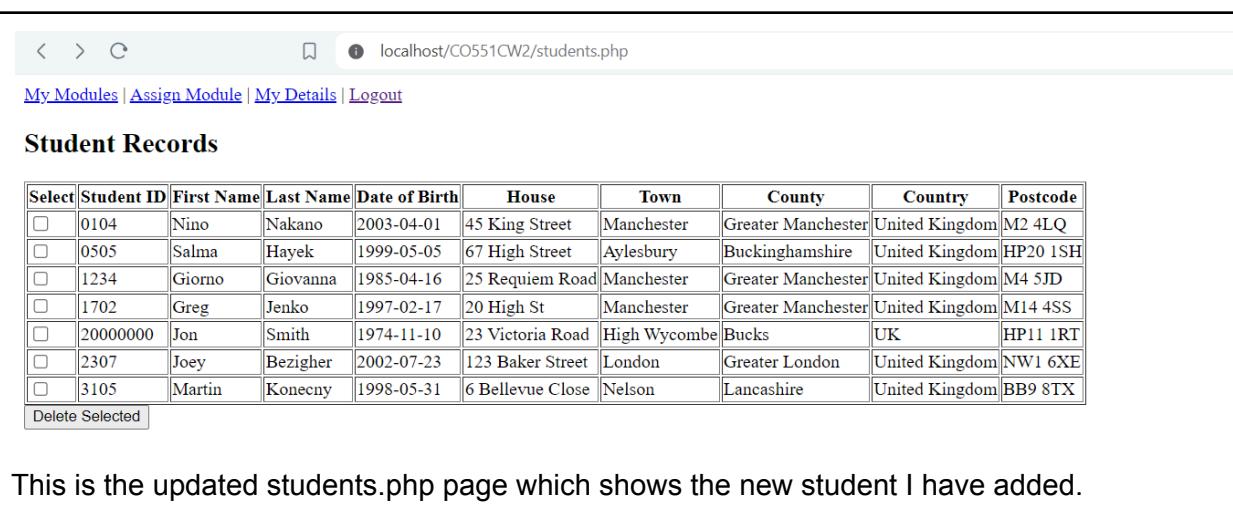
This is a simple HTML form which I have already filled up. Once the submit button is clicked the student will be added into the database and students.php table. For the validation I have decided only numbers can be used when picking student ID and only letters can be accepted when filling up first and last name (**showcased in video presentation**).



Output Message

A screenshot of a web browser window. The address bar shows "localhost/CO551CW2/addstudentsql.php". The page content says "Student added successfully." Below this message, there is a note: "Once the user fills in the form correctly and submits it, a message will appear. I have also added an error message similar to the message being used in studentseed.php if the student already exists, they will not be added."

Students Page With New Student

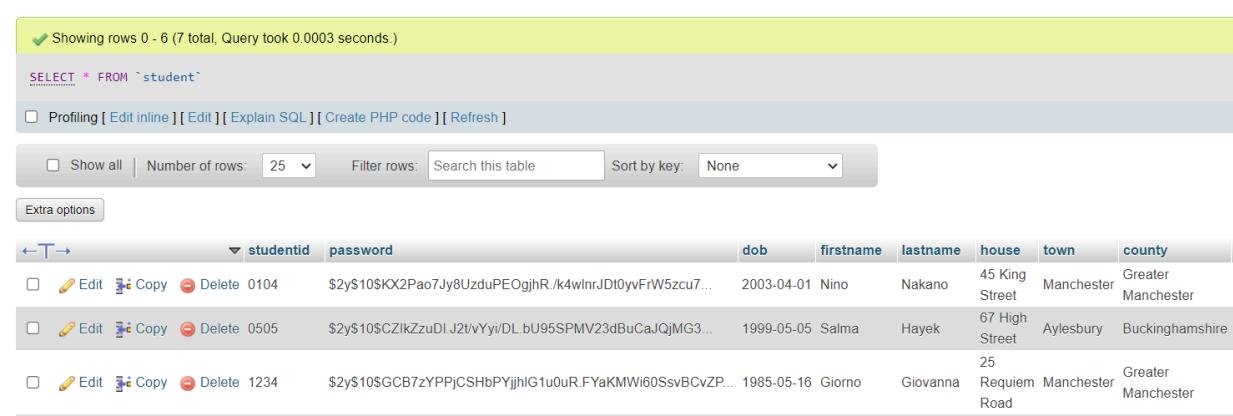
A screenshot of a web browser window titled "Students Page With New Student". The address bar shows "localhost/CO551CW2/students.php". The page contains a table of student records with columns: Select, Student ID, First Name, Last Name, Date of Birth, House, Town, County, Country, and Postcode. A new row is added for "Martin Konecny" with Student ID 3105. Below the table is a button labeled "Delete Selected".

Select	Student ID	First Name	Last Name	Date of Birth	House	Town	County	Country	Postcode
<input type="checkbox"/>	0104	Nino	Nakano	2003-04-01	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ
<input type="checkbox"/>	0505	Salma	Hayek	1999-05-05	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH
<input type="checkbox"/>	1234	Giorno	Giovanna	1985-04-16	25 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD
<input type="checkbox"/>	1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS
<input type="checkbox"/>	20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT
<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE
<input type="checkbox"/>	3105	Martin	Konecny	1998-05-31	6 Bellevue Close	Nelson	Lancashire	United Kingdom	BB9 8TX

Delete Selected

This is the updated students.php page which shows the new student I have added.

PhpMyAdmin

A screenshot of the PhpMyAdmin interface. It shows a table named "student" with columns: studentid, password, dob, firstname, lastname, house, town, and county. The table contains three rows of data. The "password" column shows hashed values. The "Extra options" dropdown is open, showing "Edit", "Copy", and "Delete" buttons for each row.

	studentid	password	dob	firstname	lastname	house	town	county
<input type="checkbox"/>	0104	\$2y\$10\$KX2Pao7Jy8UzduPEOghR./k4wlnrJDt0yFrW5zcu7...	2003-04-01	Nino	Nakano	45 King Street	Manchester	Greater Manchester
<input type="checkbox"/>	0505	\$2y\$10\$CZIkZzuDIJ2t/vYyi/DLbU95SPMV23dBuCaJQjMG3...	1999-05-05	Salma	Hayek	67 High Street	Aylesbury	Buckinghamshire
<input type="checkbox"/>	1234	\$2y\$10\$GCB7zYPPjCSHbPYjhIG1u0uR.FYaKMW60SsvBCvZP...	1985-05-16	Giorno	Giovanna	25 Requiem Road	Manchester	Greater Manchester

This screenshot shows that the password I set for Giorno is hashed and securely stored.



Code

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
  </head>
  <body>
    <h1>Add Student</h1>
    <form action="addstudentsql.php" method="post">
      <label for="studentid">Student ID:</label><br>
      <input type="text" id="studentid" name="studentid" required><br>
      <br>
      <label for="password">Password:</label><br>
      <input type="password" id="password" name="password" required><br>
      <br>
      <label for="dob">Date of Birth:</label><br>
      <input type="date" id="dob" name="dob" required><br>
      <br>
      <label for="firstname">First Name:</label><br>
      <input type="text" id="firstname" name="firstname" required><br>
      <br>
      <label for="lastname">Last Name:</label><br>
      <input type="text" id="lastname" name="lastname" required><br>
      <br>
      <label for="house">Address:</label><br>
      <input type="text" id="house" name="house" required><br>
      <br>
      <label for="town">Town/City:</label><br>
      <input type="text" id="town" name="town" required><br>
      <br>
      <label for="county">County:</label><br>
      <input type="text" id="county" name="county" required><br>
      <br>
      <label for="country">Country:</label><br>
      <input type="text" id="country" name="country" required><br>
      <br>
      <label for="postcode">Postcode:</label><br>
      <input type="text" id="postcode" name="postcode" required><br>
      <br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

A simple HTML form



A screenshot of a code editor showing a PHP script named addstudentsql.php. The code includes various MySQLi functions for escaping strings and inserting data into a 'student' table. It also includes error handling and message output. The code uses the \$_POST superglobal to get form data and mysqli_real_escape_string to prevent SQL injection.

```
index.php x studentseed.php students.php addstudent.php addstudentsql.php ● deletestudents.php

addstudentsql.php
1 <?php
2     include("includes/config.inc");
3     include("_includes/dbconnect.inc");
4
5     // This block of code gets the data from the form
6     $studentid = mysqli_real_escape_string($conn, $_POST['studentid']);
7     $password = mysqli_real_escape_string($conn, $_POST['password']);
8     $dob = mysqli_real_escape_string($conn, $_POST['dob']);
9     $firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
10    $lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
11    $house = mysqli_real_escape_string($conn, $_POST['house']);
12    $town = mysqli_real_escape_string($conn, $_POST['town']);
13    $county = mysqli_real_escape_string($conn, $_POST['county']);
14    $country = mysqli_real_escape_string($conn, $_POST['country']);
15    $postcode = mysqli_real_escape_string($conn, $_POST['postcode']);
16
17    // Hashes the password for extra security
18    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
19
20    // Insert data into student table
21    $sql = "INSERT INTO student (studentid, password, dob, firstname, lastname, house, town, county, country, postcode) VALUES ('$studentid', '$hashedPassword', '$dob', '$firstname', '$lastname', '$house', '$town', '$county', '$country', '$postcode')";
22
23    // Displays message
24    if (mysqli_query($conn, $sql)) {
25        echo "Student added successfully.";
26    } else {
27        echo "Error adding student: " . $sql . "<br>" . mysqli_error($conn);
28    }
29
30    mysqli_close($conn);
31 ?>
```

PHP code This screenshot was taken after course work was completed so it includes mysqli_real_escape_string() to avoid sql injections

Task 5 - Uploading Images To Students

I was encountering errors with this task and completed it after finishing task 6 hence why form & table look complete as they use full styling.

In this task I have added the option to upload a student's image when adding a new student. So once again the user fills in the form but this time they can upload a file (for validation i have made it so only certain file types are accepted such as jpeg and png). The image is then stored directly inside the database as a blob rather than being stored in a local file. **Validation is showcased in the video presentation.**

Output

✉ Postcode:

📷 Upload Image: Choose file No file chosen

Submit

"Choose file" option is located at the end of the form, the user can only select jpeg and png images as well as images of a certain size. Large images will not be accepted.



Output Students.php

<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE	
<input type="checkbox"/>	3105	Sabrina	Thompson	1998-05-31	6 Pine Close	Nelson	Lancashire	United Kingdom	BB9 5TX	
<input type="checkbox"/>	7654	Sarah	Smith	1997-05-10	19 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD	

This screenshot shows the updated table where the image of a student is displayed correctly

PhpMyAdmin

dob	firstname	lastname	house	town	county	country	postcode	image_path
2003-04-01	Nino	Perez	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ	[BLOB - 50.7 KiB]
1999-05-05	Salma	Hayek	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH	[BLOB - 64.0 KiB]
1997-02-17	Greg	Jenko	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS	[BLOB - 62.9 KiB]
1998-05-31	Martin	Konecny	6 Bellevue	Nelson	Lancashire	United Kingdom	BB9 8TX	[BLOB - 31.7 KiB]
1974-11-10	Jon	Smith	28 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT	NULL
2002-07-23	Joey	Bezigher	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE	[BLOB - 40.1 KiB]
1998-05-31	Sabrina	Thompson	6 Pine Close	Nelson	Lancashire	United Kingdom	BB9 5TX	[BLOB - 39.0 KiB]
1997-05-10	Sarah	Smith	19 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD	[BLOB - 64.0 KiB]

None

Jon Smith does not have an image assigned to them as that student is a default student from the forked template hence why the column shows NULL.



Code

```
<script>
  document.addEventListener("DOMContentLoaded", function() {
    const form = document.forms['frmdetails'];

    form.addEventListener('submit', function(event) {
      let valid = true;

      // Clear previous error messages
      const errorMessages = document.querySelectorAll('.error-message');
      errorMessages.forEach(function(message) {
        message.textContent = '';
      });

      // Validate Student ID (numbers only)
      const studentid = form['studentid'].value;
      if (!/^\d+$/.test(studentid)) {
        document.getElementById('studentid-error').textContent = "Student ID must contain only numbers.";
        valid = false;
      }

      // Validate First Name (letters only)
      const firstname = form['firstname'].value;
      if (!/^[a-zA-Z]+$/i.test(firstname)) {
        document.getElementById('firstname-error').textContent = "First Name must contain only letters.";
        valid = false;
      }

      // Validate Last Name (letters only)
      const lastname = form['lastname'].value;
      if (!/^[a-zA-Z]+$/i.test(lastname)) {
        document.getElementById('lastname-error').textContent = "Last Name must contain only letters.";
        valid = false;
      }

      if (!valid) {
        event.preventDefault(); // Prevent form submission if invalid
      }
    });
  });
</script>
```

Javascript Validation



```
addstudentsql.php
1 <?php
2
3     function template($template_path, $data) {
4         ob_start();
5         include $template_path;
6         return ob_get_clean();
7     }
8
9     include("_includes/config.inc");
10    include("_includes/dbconnect.inc");
11
12    echo template("templates/partials/header.php", $data);
13    echo template("templates/partials/nav.php", $data);
14 ?>
15
16 <div class="main-content" style="margin-left: 240px; font-family: 'Roboto', sans-serif; font-size: 18px; background-color: #f4f4f4;">
17     <?php
18         // Sanitize and escape user input to prevent SQL injection attacks
19         $studentid = mysqli_real_escape_string($conn, $_POST['studentid']);
20         $password = mysqli_real_escape_string($conn, $_POST['password']);
21         $dob = mysqli_real_escape_string($conn, $_POST['dob']);
22         $firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
23         $lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
24         $house = mysqli_real_escape_string($conn, $_POST['house']);
25         $town = mysqli_real_escape_string($conn, $_POST['town']);
26         $county = mysqli_real_escape_string($conn, $_POST['county']);
27         $country = mysqli_real_escape_string($conn, $_POST['country']);
28         $postcode = mysqli_real_escape_string($conn, $_POST['postcode']);
29
30         // File upload
31         $imageData = file_get_contents($_FILES['image']['tmp_name']);
32         $imageData = mysqli_real_escape_string($conn, $imageData); // real escape string to prevent sql attacks
33
34         // Hash the password for security
35         $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
36
37         // Insert data into student table
38         $sql = "INSERT INTO student (studentid, password, dob, firstname, lastname, house, town, county, country, postcode, image_path)
39             VALUES ('$studentid', '$hashedPassword', '$dob', '$firstname', '$lastname', '$house', '$town', '$county', '$country', '$postcode', '$imageData')";
40
41         // Execute SQL query
42         if (mysqli_query($conn, $sql)) {
43             // Display message for student addition
44             echo "<div style='position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); text-align: center;'>";
45             echo "<p>Student has been added.</p>";
46             echo "</div>";
47         } else {
48             echo "Error adding student: " . mysqli_error($conn);
49         }
50
51         mysqli_close($conn);
52     ?>
53 </div>
```

This screenshot was taken after course work was completed so it includes `mysqli_real_escape_string()` to avoid sql injections



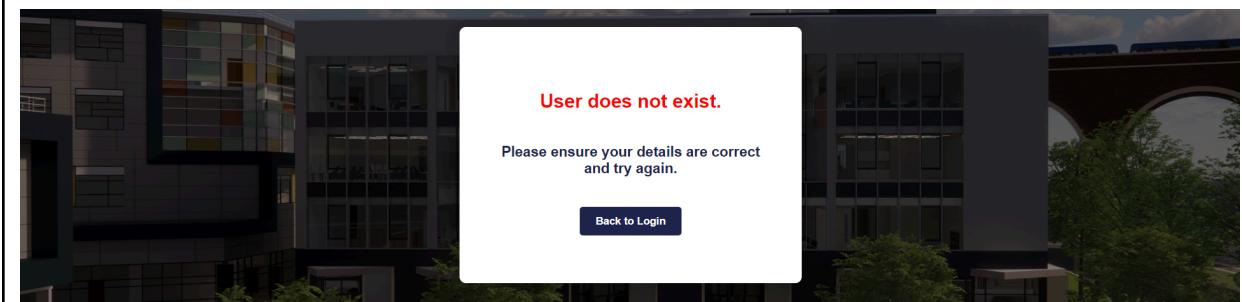
Task 6 - Improving The Interface Using CSS

In this task I have used modern and emerging HTML & CSS to greatly improve the user interface. To guarantee consistency and a cohesive user experience throughout the entire web application, I have adhered to the same colour palette. I have used websites such as W3School and CodePen to research different styles suitable for web applications such as this one.

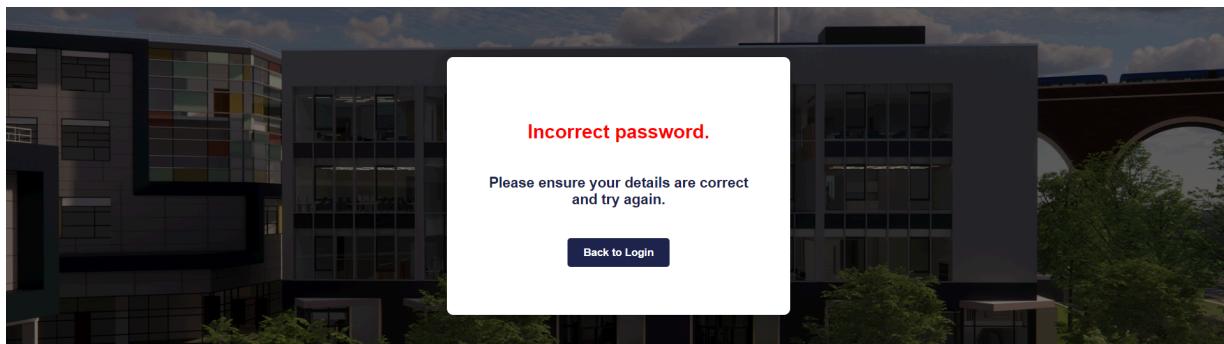
Login Page

A screenshot of a web browser displaying a login form. The form is white with a dark blue header and footer. It contains fields for 'Student ID' and 'Password', a terms and conditions link, and a 'LOGIN' button. The background is a blurred image of a modern building with glass windows and greenery.

This is the login page, the existing user within the database must input their login credentials in order to access the landing page, if the user inputs incorrect credentials they will be redirected into the error page.



If user inputs ID that does not exist in the database this message will appear



If user inputs incorrect password, this message will appear

Code

```
<body>
    <!-- Entire login box section -->
    <div class="container">
        <!-- Login Form -->
        <form class="login_form" method="POST" name="frm_login" action="authenticate.php">
            <!-- Header Section -->
            <header class="login_header">
                <h1 class="login_title">Login</h1>
            </header>
            <!-- Main Section -->
            <main class="login_main">
                <!-- Student ID Input Group -->
                <div class="login_group">
                    <input class="login_input" type="text" name="txtid" required>
                    <label class="login_label">Student ID</label>
                    <div class="login_bar"></div>
                </div>
                <!-- Password Input Group -->
                <div class="login_group">
                    <input class="login_input" type="password" name="txtpwd" required>
                    <label class="login_label">Password</label>
                    <div class="login_bar"></div>
                </div>
                <!-- Terms and Privacy Policy -->
                <p class="login_terms">
                    By Logging In, I confirm that I have read and agreed to the
                    <a href="#">Terms</a> and
                    <a href="#">Privacy Policy</a>
                </p>
            </main>
            <!-- Footer section of the box -->
            <footer class="login_footer">
                <input class="login_button" type="submit" name="btnlogin" value="Login">
            </footer>
        </form>
    </div>
```

This is the login page HTML code. It uses a simple login form inside a box. Image with opacity is set as a background



```


```



```
<style>
    /* Styling for the body */
    body {
        font-family: 'Roboto', sans-serif;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
        background-image: url('loginpage.jpg');
        background-size: cover;
        background-repeat: no-repeat;
        position: relative;
    }

    /* Adjustment for image opacity here */
    .overlay {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background-color: rgba(0, 0, 0, 0.5);
    }

    /* Container which holds the error message */
    .error-container {
        background-color: #ffffff;
        box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);
        padding: 10px 40px;
        border-radius: 8px;
        max-width: 350px;
        width: 100%;
        text-align: center;
        position: relative;
        z-index: 1;
        height: auto;
    }

    /* Error message styling */
    .error-message {
        color: #ff0000;
        margin-bottom: 20px;
    }
</style>
```

Styling for the error message page.

Landing Page

This is the landing page, I have used the image as a header and then I added another image side by side with text. On the left hand side I have customised the nav.php template and turned it into a side navigation bar because the majority of today's popular admin panels use side navigation bars, for example blackboard.



Code (Navigation Bar)

```

8      <style>
9      /* Import Montserrat font */
10     @import url("https://fonts.googleapis.com/css2?family=Roboto:wght@400,700&display=swap");
11
12     /* Side Navigation Styles */
13     .sidenav {
14         height: 100px;
15         width: 220px;
16         position: fixed;
17         left: 0;
18         top: 0;
19         left: 0;
20         background-color: #E6EAF2;
21         overflow-x: hidden;
22         padding-top: 50px;
23         text-align: center;
24     }
25
26     /* Logo Styles */
27     .logo img {
28         width: 200px;
29         height: auto;
30         margin-bottom: 20px;
31     }
32
33     /* Navigation Links styles */
34     .sidenav a {
35         padding: 10px 0px;
36         text-decoration: none;
37         font-size: 16px;
38         color: #C00000;
39         display: flex;
40         align-items: center;
41         -webkit-transition: font-size .3s ease;
42         -moz-transition: font-size .3s ease;
43         -o-transition: font-size .3s ease;
44         transition: font-size .3s ease;
45         margin-bottom: 8px;
46     }
47
48     /* Icon Styles */
49     .sidenav a img {
50         width: 32px;
51         height: 32px;
52         margin-right: 10px;
53     }
54
55     /* Text Styles */
56     .sidenav a span {
57         font-family: 'Roboto', sans-serif;
58         padding-left: 8px;
59     }
60
61     /* Hover Styles */
62     .sidenav a:hover {
63         color: #f1f1f1;
64         cursor: pointer;
65         font-size: 1.1em;
66     }
67
68     /* Main Content Styles */
69     .main-content {
70         margin-left: 120px;
71         padding: 20px;
72     }
73   </style>

```

```

76   <body>
77     <!-- Div for entire side navigation bar -->
78     <div class="sidenav">
79       <!-- Section for logo -->
80       <div class="logo">
81           
82       </div>
83       <!-- Navigation links -->
84       <a href="index.php">
85           <!-- font awesome icons -->
86           <i class="fa fa-info-circle"></i>
87           <span>Index</span>
88       </a>
89       <a href="modules.php">
90           <i class="fa fa-graduation-cap"></i>
91           <span>My Modules</span>
92       </a>
93       <a href="assignmodule.php">
94           <i class="fa fa-desktop"></i>
95           <span>Assign Module</span>
96       </a>
97       <a href="details.php">
98           <i class="fa fa-edit"></i>
99           <span>My Details</span>
100      </a>
101      <a href="students.php">
102          <i class="fa fa-users"></i>
103          <span>Students</span>
104      </a>
105      <a href="addstudent.php">
106          <i class="fa fa-user-plus"></i>
107          <span>Add Student</span>
108      </a>
109      <a href="logout.php">
110          <i class="fa fa-power-off"></i>
111          <span>Logout</span>
112      </a>
113   </div>
114 </body>

```

This is the source code for my nav bar template, it uses effects such as text zoom if hovered over.

Code

```

1  index.php
2  <?php
3  include("_includes/config.inc");
4  include("_includes/dbconnect.inc");
5  include("_includes/functions.inc");
6
7  // If user is logged in, display admin page
8  if (isset($_SESSION['id'])) {
9      $data['content'] = "
10      <div class='centered-image'>
11          <img src='indexheader.png' alt='Hero Image'>
12      </div>
13      <div class='centered-box' style='width: 80%;'>
14          <div class='box-left'>
15              <img src='indexpic.jpg' alt='Left Image' style='width: 100%;'>
16          </div>
17          <div class='box-right'>
18              <h1>WELCOME TO ADMIN PAGE</h1>
19              <h2>Please select an option from the navigation bar</h2>
20              <p>Nullam vel dapibus lacus. Sed nec justo id leo rutrum eleifend. Curabitur nec metus a libero ullamcorper tincidunt. Sed a aliquam nisi. Nullam dignissim eros vel commodo dapibus. Sed ac posuere ligula. Vivamus euismod nisl in sollicitudin vulputate. Integer tristique, dui ut blandit tempor, arcu velit faucibus lacus, nec fermentum nisi orci eu risus.</p>
21              <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ac tortor in velit tempor ullamcorper nec sed felis. Fusce eget nisi sed justo fringilla rhoncus. Mauris in sapien ut elit eleifend ultrices.</p>
22          </div>
23      </div>";
24
25      // Templates
26      echo template("templates/partials/nav.php");
27      echo template("templates/default.php", $data);
28  } else {
29      // If user is not logged in, display login page
30      echo template("templates/login.php", $data);
31  }
32 ?>

```



```
<style>
  @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap');

  /* Font family */
  body {
    font-family: 'Roboto', sans-serif;
  }

  /* Positions the image in the center of the page */
  .centered-image {
    position: absolute;
    top: 10%;
    left: 44.78%;
    transform: translate(-50%, -50%);
  }

  /* Centers the box horizontally and vertically */
  .centered-box {
    position: absolute;
    top: 70%;
    left: 58%;
    transform: translate(-50%, -50%);
    display: flex;
    justify-content: center;
  }

  /* Sets the width of the left box */
  .box-left {
    flex: 1;
  }

  /* Set the width and padding for the right box */
  .box-right {
    flex: 1;
    padding-left: 20px;
  }

  /* Set the font size for h1 */
  .box-right h1 {
    font-size: 40px;
  }

  /* Set the font size for h2 */
  .box-right h2 {
    font-size: 30px;
  }

  /* Set the font size for p */
  .box-right p {
    font-size: 20px;
  }
</style>
```

Above is the PHP code that brings out all the templates such as the navigation bar template. I included html code inside the php code to avoid an error i was encounetrng.This is the CSS code for styling on this page



My Modules Page

This is modules.php page it shows current modules assigned to the user. For this page i used standard styling to create a container in the centre of the page with heading and a table, the table has hover effects.

Code

```
modules.php
1  <?php
2    // Includes necessary configuration, functions and database connection files
3    include("includes/config.inc");
4    include("includes/dbconnect.inc");
5    include("_includes/functions.inc");
6
7    // Check if the user is logged in,
8    // if they are logged in they can see the content
9    if (isset($_SESSION['id'])) {
10      // Build SQL statement to select a student's modules
11      $sql = "SELECT * FROM studentmodules sm, module m
12      WHERE m.modulecode = sm.modulecode AND
13      sm.studentid = " . $_SESSION['id'] . ";
14      // Executes the sql query
15      $result = mysqli_query($conn, $sql);
16
17      // Prepares table content
18      $tableContent = "<table class='module-table'>";
19      $tableContent .= "<tbody><tr><th>Code</th><th>Type</th><th>Level</th></tr>";
20
21      // Displays the modules assigned with the student inside html table
22      while($row = mysqli_fetch_array($result)) {
23        $tableContent .= "<tr><td>$row[modulecode]</td><td>$row[name]</td>";
24        $tableContent .= "<td>$row[level]</td></tr>";
25      }
26      $tableContent .= "</tbody></table>";
27
28      // includes templates
29      echo template("templates/partials/header.php");
30      echo template("templates/partials/nav.php"); //
31      echo "
32      <div class='table-wrapper'>
33        <div class='table-container'>
34          <h1 class='modules-heading'>MY CURRENT MODULES</h1> <!-- Heading above the table --&gt;
35          $tableContent &lt;!-- Table content --&gt;
36        &lt;/div&gt;
37      &lt;/div&gt;;
38    } else {
39      // if user isn't logged in they will be redirected to login page
40      header("Location: index.php");
41    }
42  &gt;&gt;</pre>
```

This php code checks and retrieves assigned modules to a signed in student and displays them in a table.



```
<style>
/* CSS Styles */
body {
    font-family: 'Roboto', sans-serif;
    margin: 0;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #f4f4f4;
}

.table-wrapper {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100%;
    width: 100%;
}

/* Actual styling for the table*/
.module-table {
    width: 100%;
    border-collapse: collapse;
    margin: 0 auto;
    font-family: 'Roboto', sans-serif;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    overflow: hidden;
    background-color: #fff;
}

.module-table th, .module-table td {
    padding: 12px;
    text-align: left;
    transition: background-color 0.3s;
}

.module-table th {
    background-color: #f2f2f2;
    font-size: 18px;
    border-bottom: 2px solid #ddd;
}

.module-table td {
    font-size: 16px;
    border-bottom: 1px solid #ddd;
}

/* Add alternating row colors */
.module-table tbody tr:nth-child(even) {
    background-color: #fafafa;
}

.module-table tbody tr:nth-child(odd) {
    background-color: #ffffff;
}

/* hover effect */
.module-table tbody tr:hover {
    background-color: #e0e0e0;
}

/* Container which holds the table */
.table-container {
    max-width: 1400px;
    margin-left: 100px;
    padding: 20px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.2);
    text-align: center;
}

/* Module heading styles */
.modules-heading {
    font-size: 40px;
    margin-bottom: 20px;
    color: #333;
}

</style>
```

All of the styles for modules page. It includes a white box which holds the table. It uses shadow and has round edges. Table has different styles such as the hoover styles and styles for the font inside it.



Assign Module Page

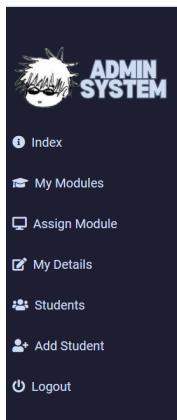
This is the assigned module page, in this page users can assign a module to themselves.

A screenshot of a web browser window showing the 'Assign Module' page. The left sidebar has links for Index, My Modules, Assign Module (which is selected), My Details, Students, Add Student, and Logout. The main content area has a form titled 'Select a module to assign' with a dropdown menu showing 'Programming Development' and a 'Save' button below it.

Drop Down box and save button

A screenshot of a web browser window showing the 'Assign Module' page. The left sidebar has links for Index, My Modules, Assign Module (which is selected), My Details, Students, Add Student, and Logout. The main content area has a form titled 'Select a module to assign' with a dropdown menu showing three options: 'Programming Development', 'Programming Principles', and 'Internet Systems Development'. The 'Programming Development' option is highlighted with a blue background.

Three different modules which can be selected



Once again a standard styling is used here that creates a container in the centre of the screen with heading and dropdown box and a submit button. Once the submit button is clicked a message will appear prompting the user that module was successfully assigned to the student.

Code

```
assignmodule.php
1  <?php
2      // Includes neccessary configuration, functions and database conection files
3      include("_includes/config.inc");
4      include("_includes/dbconnect.inc");
5      include("_includes/functions.inc");
6
7      // check if logged in
8      if (isset($_SESSION['id'])) {
9
10         echo template("templates/partials/header.php");
11         echo template("templates/partials/nav.php");
12
13         // If a module has been selected
14         if (isset($_POST['selmodule'])) {
15             // Sanitize session id and selected module to prevent SQL injection
16             $studentId = mysqli_real_escape_string($conn, $_SESSION['id']);
17             $selectedModule = mysqli_real_escape_string($conn, $_POST['selmodule']);
18
19             $sql = "insert into studentmodules values ('$studentId', '$selectedModule')";
20             $result = mysqli_query($conn, $sql);
21             $data['content'] .= "<p>The module $selectedModule has been assigned to you</p>";
22         } else {
23
24             // Build SQL statement that selects all the modules
25             $sql = "select * from module";
26             $result = mysqli_query($conn, $sql);
27
28             $data['content'] .= "<div class='form-container'>";
29             $data['content'] .= "<form name='frmassignmodule' action='.' method='post'>";
30             $data['content'] .= "<label for='selmodule'>Select a module to assign</label><br/>";
31             $data['content'] .= "<select name='selmodule' id='selmodule'>";
32             // Display the module names in a drop down selection box
33             while ($row = mysqli_fetch_array($result)) {
34                 $data['content'] .= "<option value='".$row['modulecode']."'>" . $row['name'] . "</option>";
35             }
36             $data['content'] .= "</select><br/>";
37             $data['content'] .= "<input type='submit' name='confirm' value='Save' />";
38             $data['content'] .= "</form>";
39             $data['content'] .= "</div>";
40         }
41
42         // includes templates
43         echo template("templates/default.php", $data);
44
45         // if user isn't logged in they will be redirected to lofin page
46     } else {
47         header("Location: index.php");
48     }
49
50 >>
```

This php code assigns modules to a logged on student. It is sent to a database. After the module is selected and saved, the user will see a message.



```
<style>
  /* CSS Styles */
  @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap');

  body {
    font-family: 'Roboto', sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }

  /* Container which holds the drop down with modules*/
  .form-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    max-width: 600px;
    width: 100%;
    padding: 40px;
    border: 1px solid #ddd;
    border-radius: 10px;
    background-color: #fff;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease;
    margin-left: -60px;
  }

  /* Hoover effect for the drop down */
  .form-container:hover {
    box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
  }

  label {
    font-size: 24px;
    margin-bottom: 20px;
    width: 100%;
    text-align: left;
  }

  select {
    width: 100%;
    padding: 16px;
    margin-bottom: 20px;
    margin-top: 10px;
    border: 1px solid #ddd;
    border-radius: 8px;
    font-size: 20px;
    background-color: #fafafa;
    transition: border-color 0.3s ease;
  }

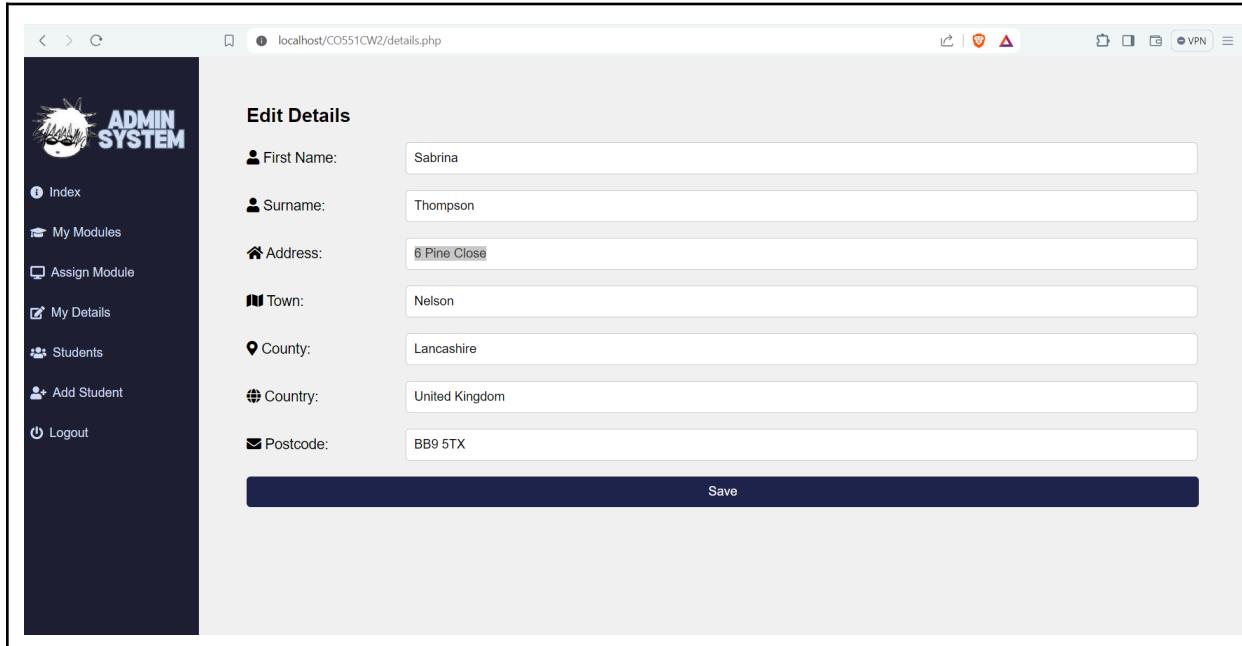
  select:focus {
    border-color: #4CAF50;
    outline: none;
  }

  /* Button styles */
  input[type='submit'] {
    width: 100%;
    padding: 16px;
    margin-top: 10px;
    border: none;
    border-radius: 8px;
    background-color: #22254E;
    color: white;
    font-size: 20px;
    cursor: pointer;
    transition: background-color 0.3s ease;
  }

  input[type='submit']:hover {
    background-color: #CEDDF4;
  }
</style>
```

Styles for the assign a module page. I stuck to the same design so I used a white box in the centre of the screen which uses shadows and has round corners, then inside the box I have put a drop down box where the user can select a module and assign it to themselves.

My Details Page



The screenshot shows a web browser window with the URL `localhost/CO551CW2/details.php`. The page title is "Edit Details". On the left, there is a sidebar with a logo and navigation links: Index, My Modules, Assign Module, My Details (which is selected), Students, Add Student, and Logout. The main content area has a form titled "Edit Details" with the following fields:

Field	Value
First Name:	Sabrina
Surname:	Thompson
Address:	6 Pine Close
Town:	Nelson
County:	Lancashire
Country:	United Kingdom
Postcode:	BB9 5TX

A large blue "Save" button is at the bottom of the form.



This is my details page, where the user can edit their details such as their address. Once they make the changes and click on submit a message will appear saying the changes were made. Once again a very simple css is used here. Simple form with font awesome icons. When it comes to the message I have added it inside a container.

Code

```

details.php
1 <?php
2   include("_includes/config.inc");
3   include("_includes/dbconnect.inc");
4   include("_includes/functions.inc");
5
6   // Check if logged in
7   if (isset($_SESSION['id'])) {
8
9     echo template("templates/partials/header.php");
10    echo template("templates/partials/nav.php");
11
12   // If the form has been submitted
13   if (isset($_POST['submit'])) {
14
15     // Sanitize user inputs to prevent SQL injection
16     $firstname = mysql_real_escape_string($conn, $_POST['txtfirstname']);
17     $lastname = mysql_real_escape_string($conn, $_POST['txtlastname']);
18     $house = mysql_real_escape_string($conn, $_POST['txthouse']);
19     $town = mysql_real_escape_string($conn, $_POST['txtown']);
20     $county = mysql_real_escape_string($conn, $_POST['txtcountry']);
21     $country = mysql_real_escape_string($conn, $_POST['txtcountry']);
22     $postcode = mysql_real_escape_string($conn, $_POST['txtpostcode']);
23
24     $studentid = mysql_real_escape_string($conn, $_SESSION['id']);
25
26     // Build an SQL statement to update the student details
27     $sql = "UPDATE student SET
28       firstname = '$firstname',
29       lastname = '$lastname',
30       house = '$house',
31       town = '$town',
32       county = '$county',
33       country = '$country',
34       postcode = '$postcode'
35     WHERE studentid = '$studentid'";
36
37     $result = mysql_query($conn, $sql);
38
39     $data['content'] = "<div class='message-container'><h3>Your details have been updated!</h3></div>";
40
41   } else {
42
43     // Build an SQL statement to return the student record with the id that matches that of the session variable
44     $studentid = mysql_real_escape_string($conn, $_SESSION['id']);
45     $sql = "SELECT * FROM student WHERE studentid=$studentid";
46     $result = mysql_query($conn, $sql);
47     $row = mysql_fetch_array($result);
48
49   }
50
51   // Using <<<EOO notation to allow building of a multi-line string
52   $data['content'] = <<<EOO
53
54   <div class="main-content">
55     <div>Details</div>
56     <form name="frmdetails" action="" method="post">
57       <div class="form-row">
58         <label for="txtfirstname"><i class="fas fa-user"></i> First Name:</label>
59         <input name="txtfirstname" type="text" value="$row['firstname']" />
60       </div>
61       <div class="form-row">
62         <label for="txtlastname"><i class="fas fa-user"></i> Surname:</label>
63         <input name="txtlastname" type="text" value="$row['lastname']" />
64       </div>
65       <div class="form-row">
66         <label for="txthouse"><i class="fas fa-home"></i> Address:</label>
67         <input name="txthouse" type="text" value="$row['house']" />
68       </div>
69       <div class="form-row">
70         <label for="txtown"><i class="fas fa-map"></i> Town:</label>
71         <input name="txtown" type="text" value="$row['town']" />
72       </div>
73       <div class="form-row">
74         <label for="txtcountry"><i class="fas fa-globe"></i> County:</label>
75         <input name="txtcountry" type="text" value="$row['county']" />
76       </div>
77       <div class="form-row">
78         <label for="txtpostcode"><i class="fas fa-envelope"></i> Postcode:</label>
79         <input name="txtpostcode" type="text" value="$row['postcode']" />
80       </div>
81       <div class="submit-btn">
82         <input type="submit" value="Save" name="submit"/>
83       </div>
84     </form>
85   EOO;
86
87   // Render the template
88   echo template("templates/default.php", $data);
89
90   } else {
91     header("Location: index.php");
92   }
93
94   echo template("templates/partials/footer.php");
95 
```

PHP code which allows changes to be made to the students details, current details are retrieved from the database and updated details are sent back to the database. User gets a message as well after they submit new details. It includes a form with icons.



```

95 <style>
96   /* General Styles */
97   body {
98     font-family: 'Roboto', sans-serif;
99     background-color: #f2f2f2;
100    margin: 0;
101    padding: 0;
102  }
103
104  .container {
105    max-width: 1200px;
106    margin: 20px auto;
107    padding: 0 20px;
108  }
109
110  /* Form Styles */
111  .form-container {
112    background-color: #fff;
113    border-radius: 10px;
114    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
115    padding: 20px;
116    margin-bottom: 20px;
117  }
118
119  .form-row {
120    margin-bottom: 20px;
121    display: flex;
122    align-items: center;
123  }
124
125  .form-row label {
126    width: 200px;
127    flex: 0 0 200px;
128    font-size: 18px;
129    margin-bottom: 0;
130  }
131
132  .form-row input[type="text"] {
133    flex: 1;
134    height: 40px;
135    font-size: 16px;
136    padding: 10px;
137    box-sizing: border-box;
138    border: 1px solid #ccc;
139    border-radius: 5px;
140  }
141
142  /* Button Styles */
143  .submit-btn input[type="submit"] {
144    background-color: #22254E;
145    color: white;
146    border: none;
147    padding: 10px 20px;
148    text-align: center;
149    text-decoration: none;
150    display: inline-block;
151    font-size: 16px;
152    border-radius: 5px;
153    cursor: pointer;
154    transition: background-color 0.3s;
155    width: 98.5%;
156  }
157
158  .submit-btn input[type="submit"]:hover {
159    background-color: #CED0F4;
160  }
161
162  /* Message Styles */
163  .message-container {
164    background-color: #fff;
165    border-radius: 10px;
166    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
167    padding: 20px;
168    margin: 20px auto;
169    width: 400px;
170    height: 250px;
171    position: absolute;
172    top: 50%;
173    left: 50%;
174    transform: translate(-50%, -50%);
175    display: flex;
176    justify-content: center;
177    align-items: center;
178  }
179
180  .message-container h1 {
181    font-size: 24px;
182    color: #333;
183  }
184
185 </style>

```

These are the styles I used. For this page I took a different approach, rather than having it inside a container I decided to just have it in the actual page to save some space as the form is quite large. I used icons and stuck to the same theme. It also has some effects such as the hover effect on the button. I used white container for the message, once again round edges with shadows.



Students Page



- [Index](#)
- [My Modules](#)
- [Assign Module](#)
- [My Details](#)
- [Students](#)
- [Add Student](#)
- [Logout](#)

Select	Student ID	First Name	Last Name	Date of Birth	House	Town	County	Country	Postcode	Image
<input type="checkbox"/>	0104	Nino	Perez	2003-04-01	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ	
<input type="checkbox"/>	0505	Salma	Hayek	1999-05-05	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH	
<input type="checkbox"/>	1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS	
<input type="checkbox"/>	1998	Martin	Konecny	1998-05-31	6 Bellevue	Nelson	Lancashire	United Kingdom	BB9 8TX	
<input type="checkbox"/>	20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT	Student Image
<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE	
<input type="checkbox"/>	3105	Sabrina	Thompson	1998-05-31	23 Pine Close	Nelson	Lancashire	United Kingdom	BB9 5TX	
<input checked="" type="checkbox"/>	31055555	Giormos	Giovanna	2000-12-10	25 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD	
<input type="checkbox"/>	7654	Sarah	Smith	1997-05-10	19 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD	

[Delete Selected](#)



- [Index](#)
- [My Modules](#)
- [Assign Module](#)
- [My Details](#)
- [Students](#)
- [Add Student](#)
- [Logout](#)

Student Records										
Select	Student ID	First Name	Last Name	Date of Birth	House	Town	County	Country	Postcode	Image
<input type="checkbox"/>	0104	Nino	Perez	2003-04-01	45 King Street	Manchester	Greater Manchester	United Kingdom	M2 4LQ	
<input type="checkbox"/>	0505	Salma	Hayek	1999-05-05	67 High Street	Aylesbury	Buckinghamshire	United Kingdom	HP20 1SH	
<input type="checkbox"/>	1702	Greg	Jenko	1997-02-17	20 High St	Manchester	Greater Manchester	United Kingdom	M14 4SS	
<input type="checkbox"/>	1998	Martin	Konecny	1998-05-31	6 Bellevue	Nelson	Lancashire	United Kingdom	BB9 8TX	
<input type="checkbox"/>	20000000	Jon	Smith	1974-11-10	23 Victoria Road	High Wycombe	Bucks	UK	HP11 1RT	Student Image
<input type="checkbox"/>	2307	Joey	Bezigher	2002-07-23	123 Baker Street	London	Greater London	United Kingdom	NW1 6XE	
<input type="checkbox"/>	3105	Sabrina	Thompson	1998-05-31	23 Pine Close	Nelson	Lancashire	United Kingdom	BB9 5TX	
<input type="checkbox"/>	7654	Sarah	Smith	1997-05-10	19 Requiem Road	Manchester	Greater Manchester	United Kingdom	M4 5JD	

Student ID 31055555 has been successfully deleted!

This is a students page which displays all students in a table with the correct image assigned to each student, it has a checkbox to select a student which the user wants to delete. Before deleting the student the user will be asked if they wish to proceed and if they do a red message will appear on the top of the screen saying the user was deleted.

Code



```

10 // Displays message
11 if (isset($_SESSION['delete_message'])) {
12     echo '<p class="error-message">' . $_SESSION['delete_message'] . '</p>';
13     // Clear message after displaying
14     unset($_SESSION['delete_message']);
15 }
16
17 $sql = "SELECT * FROM student";
18 $result = mysqli_query($conn, $sql);
19
20 // Check if any records were returned from the database query
21 if (mysqli_num_rows($result) > 0) {
22     $data['content'] = "<div class='table-container'>";
23     $data['content'] .= "<h2>Student Records</h2>";
24     $data['content'] .= "<form action='deletestudents.php' method='post'>";
25     $data['content'] .= "<table>";
26     $data['content'] .= "<tr>";
27         $data['content'] .= "<th>Select</th>";
28         $data['content'] .= "<th>Student ID</th>";
29         $data['content'] .= "<th>First Name</th>";
30         $data['content'] .= "<th>Last Name</th>";
31         $data['content'] .= "<th>Date of Birth</th>";
32         $data['content'] .= "<th>House</th>";
33         $data['content'] .= "<th>Town</th>";
34         $data['content'] .= "<th>County</th>";
35         $data['content'] .= "<th>Country</th>";
36         $data['content'] .= "<th>Postcode</th>";
37         $data['content'] .= "<th>Image</th>";
38     $data['content'] .= "</tr>";
39     // Fetch each record as an associative array
40     while ($row = mysqli_fetch_assoc($result)) {
41         $imageData = base64_encode($row['image_path']);
42         $data['content'] .= "<tr>";
43             $data['content'] .= "<td><input type='checkbox' name='students[]' value='{$row['studentid']}'></td>";
44             $data['content'] .= "<td>" . $row['studentid'] . "</td>";
45             $data['content'] .= "<td>" . $row['firstname'] . "</td>";
46             $data['content'] .= "<td>" . $row['lastname'] . "</td>";
47             $data['content'] .= "<td>" . $row['dob'] . "</td>";
48             $data['content'] .= "<td>" . $row['house'] . "</td>";
49             $data['content'] .= "<td>" . $row['town'] . "</td>";
50             $data['content'] .= "<td>" . $row['county'] . "</td>";
51             $data['content'] .= "<td>" . $row['country'] . "</td>";
52             $data['content'] .= "<td>" . $row['postcode'] . "</td>";
53             $data['content'] .= "<td><img src='data:image/jpeg;base64," . $imageData . "' alt='Student Image' style='width: 50px; height: auto;'/></td>";
54         $data['content'] .= "</tr>";
55     }
56 }
57
58 $data['content'] .= "</table>";
59 $data['content'] .= "<div class='button-container'>";
60 $data['content'] .= "<input type='submit' value='Delete Selected' onclick='return confirm(\"Are you sure you want to delete the selected students?\");' class='delete-button'>";
61 $data['content'] .= "</div>";
62 $data['content'] .= "</form>";
63 $data['content'] .= "</div>";
64 } else {
65     $data['content'] = "<p>No student records found.</p>";
66 }
67 // Outputs the final table by using a template to render the page
68 echo template("templates/default.php", $data);
69 } else {
70     header("Location: index.php");
71 }
72
73 echo template("templates/partials/footer.php");

```

PHP code which displays the students inside a table, the students are being displayed from a database. It also features a message after the user deletes a student.

```

1 # deletestudents.php
2 <?php
3 include("_includes/config.inc");
4 include("_includes/dbconnect.inc");
5 include("_includes/functions.inc");
6
7 // Check if the user is logged in
8 if (isset($_SESSION['id'])) {
9     $message = "";
10    // Check if any students were selected to be deleted
11    if (isset($_POST['students']) && is_array($_POST['students'])) {
12        // Loop through selected students and delete them.
13        foreach ($_POST['students'] as $studentid) {
14            $studentid = mysqli_real_escape_string($conn, $studentid);
15            $sql = "DELETE FROM student WHERE studentid = '$studentid'";
16            if (mysqli_query($conn, $sql)) {
17                $message .= "Student ID $studentid has been successfully deleted! <br>";
18            }
19        }
20
21        // Store message in session to display on students.php
22        $_SESSION['delete_message'] = $message;
23        header("Location: students.php");
24    } else {
25        header("Location: index.php");
26    }
27 ?>

```

PHP code in a separate file which allows students to be deleted and generates the message.



```
<style>
    /* CSS Styles for table container */
    .table-container {
        max-width: 90%;
        margin: 0 auto;
        overflow-x: auto;
        padding: 20px;
        margin-left: 100px; /* Adjusted margin-left */
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        background: #fff;
        border-radius: 8px;
        font-family: 'Roboto', sans-serif; /* Apply Roboto font */
    }

    /* Additional CSS styles for the table */
    table {
        width: 100%;
        border-collapse: collapse;
        border: 1px solid #ddd;
        margin-top: 20px;
    }

    th, td {
        padding: 8px;
        text-align: left;
        border-bottom: 1px solid #ddd;
        font-size: 14px;
    }

    th {
        background-color: #f2f2f2;
        position: sticky;
        top: 0;
        z-index: 2;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }

    tr:hover {
        background-color: #BCBAC5;
        transition: background-color 0.3s ease;
    }

    /* Styles for the submit button */
    .delete-button {
        background-color: #22254E;
        color: white;
        padding: 12px 200px; /* Adjusted padding for wider button */
        border: none;
        border-radius: 4px;
        cursor: pointer;
        transition: background-color 0.3s ease;
        display: block;
        margin: 20px auto 0;
        width: auto; /* Adjusted width to auto */
        font-family: 'Roboto', sans-serif;
        font-size: 15px;
    }

    .delete-button:hover {
        background-color: #CEDDF4;
    }

    /* Styles for error message */
    .error-message {
        color: red;
        margin-top: 10px;
        text-align: center;
        font-family: 'Roboto', sans-serif;
        font-size: 20px;
    }
</style>
```

Styles used for this page. It's a table inside a white box with round edges and shadows. I kept consistency with the designs to ensure the page looks similar to other pages. The error message is displayed above the table and it is red.



Add Student Page

A screenshot of a web browser displaying the 'ADD NEW STUDENT' form. On the left is a dark sidebar with 'ADMIN SYSTEM' branding and links for Index, My Modules, Assign Module, My Details, Students, Add Student, and Logout. The main form area has a title 'ADD NEW STUDENT'. It includes fields for Student ID (with a placeholder 'Upload Image'), Password, First Name, Last Name, Date of Birth (dd/mm/yyyy), Address, Town, Postcode, County, and Country. A 'Submit' button is at the bottom. A dashed box highlights the area where an image can be uploaded.

This is the add a new student form, I have designed it in a way where it replicates a basic admin panel. Using CSS i have created a container where users can upload the image, i used javascript as well so the image can be previewed and removed.

A screenshot of the same web browser showing the 'ADD NEW STUDENT' form with validation errors. The Student ID field contains 'dfghj', which has a red error message: 'Student ID must contain only numbers.' The First Name field contains '2345', with a red error message: 'First Name must contain only letters.' The Date of Birth field contains '11/11/0001', with a red error message: 'Date of Birth must be a valid date.' The rest of the form appears to be empty or correct. The sidebar and overall layout remain the same.

This is validation I have used in this form. I decided not to use validation for passwords as the initial password is just a "test". But I used validation for an ID. Only numbers can be an ID, full name can't have numbers in it and DoB must be a valid date.



A screenshot of a web browser window. The URL is `localhost/CO551CW2/addstudentsql.php`. The page title is "ADMIN SYSTEM". On the left, there is a sidebar with the following menu items: Index, My Modules, Assign Module, My Details, Students, Add Student, and Logout. The main content area displays the message "Student has been added.".

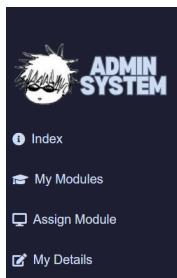
Once a user adds a new student, a message will appear on the screen.

A screenshot of a web browser window. The URL is `localhost/CO551CW2/addstudent.php`. The page title is "ADD NEW STUDENT". The sidebar on the left is identical to the previous screenshot. The main form contains fields for Student ID (3105), First Name (Giorno), Last Name (Giovanna), Date of Birth (10/10/2000), Address (25 Requiem Road), Town (Manchester), Postcode (M4 5JD), County (Greater Manchester), and Country (United Kingdom). The "Last Name" field is highlighted with a red border and a tooltip "Please fill in this field.".

If user forgets to fill in field, they will be prompted to do so before they can submit the form

A screenshot of a web browser window. The URL is `localhost/CO551CW2/addstudentsql.php`. The page title is "ADMIN SYSTEM". The sidebar on the left is identical. The main content area displays the error message "Error adding student: Got a packet bigger than 'max_allowed_packet' bytes".

Another form of validation, if a user uploads a large image, the database will reject it as only a certain number of bytes is allowed for the BLOB database.



Error adding student: Duplicate entry '3105' for key 'PRIMARY'

If the user selects an ID which is already in use an error message will appear saying that the user with that ID already exists. ID is used as a primary key in my database hence why there can't be any duplicates

Code

```
wedstudent.php
1 <?php
2     include("../includes/config.inc");
3     include("../includes/functions.inc");
4     include("../includes/functions.inc");
5
6     // check logged in
7     if (!isset($_SESSION['id'])) {
8         echo template("template/partials/header.php");
9         echo template("template/partials/nav.php");
10    }
11
12    <!-- Page Header -->
13    <div class="main-contents">
14        <!-- Header section -->
15        <div class="row">
16            <div style="background-color: #f0f0f0; text-align: center; padding: 20px; margin-top: -20px; width: calc(100% + 240px); margin-left: -120px;">
17                <h2>ADD NEW STUDENT</h2>
18            </div>
19        </div>
20        <!-- End of header section -->
21
22        <!-- Form for adding a student -->
23        <form name="student" action="addstudentsql.php" method="post" enctype="multipart/form-data">
24            <div class="form-container">
25                <!-- Row containing the image upload section and form fields -->
26                <div class="row">
27                    <div class="image-upload square" style="margin-top: 25px;">
28                        <img alt="Image placeholder" style="width: 100%; height: 100%; object-fit: cover; border-radius: 10px;"/>
29                    </div>
30                    <div class="form-column">
31                        <label for="image">Upload Image</label>
32                        <input type="file" id="image" name="image" accept="image/jpeg, image/png" required style="display: none; position: absolute; bottom: 10px; left: 10px; width: 10px; height: 10px; border: none; border-radius: 50%; opacity: 0; transition: none;"/>
33                        <button type="button" id="remove-image" style="display: none; position: absolute; bottom: 10px; left: 10px; width: 10px; height: 10px; border: none; border-radius: 50%; background-color: transparent; color: inherit; font-size: inherit; font-weight: inherit; line-height: inherit; font-family: inherit; outline: none; transition: none;"/>
34                    </div>
35                </div>
36                <div class="form-container">
37                    <div class="row">
38                        <div class="form-column">
39                            <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 10px;">
40                                <label for="studentid">Student ID:</label>
41                                <input name="studentid" type="text" required />
42                                <span class="error-message" id="studentid-error"></span>
43                            </div>
44                            <div class="form-column">
45                                <label for="password">Password:</label>
46                                <input name="password" type="password" required />
47                                <span class="error-message" id="password-error"></span>
48                            </div>
49                        </div>
50                        <div class="row">
51                            <div class="form-column">
52                                <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 10px;">
53                                    <label for="firstname">First Name:</label>
54                                    <input name="firstname" type="text" required />
55                                    <span class="error-message" id="firstname-error"></span>
56                                </div>
57                                <div class="form-column">
58                                    <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 10px;">
59                                        <label for="lastname">Last Name:</label>
60                                        <input name="lastname" type="text" required />
61                                        <span class="error-message" id="lastname-error"></span>
62                                    </div>
63                                </div>
64                            </div>
65                        </div>
66                    </div>
67                </div>
68            </div>
69            <!-- Third row of form columns -->
70            <div class="form-column">
71                <label for="dob">Date of Birth:</label>
72                <input name="dob" type="date" placeholder="YYYY-MM-DD" required />
73                <span class="error-message" id="dob-error"></span>
74            </div>
75            <div class="form-column">
76                <label for="house">Address:</label>
77                <input name="house" type="text" required />
78            </div>
79        </div>
80        <!-- New row for Postcode, County, and Country -->
81        <div class="form-row">
82            <div class="form-column small-column postcode-column" style="width: 30%; "><!-- Adjusted width -->
83                <label for="town">Town:</label>
84                <input name="town" type="text" required />
85            </div>
86            <div class="form-column county-column" style="width: 20%; "><!-- Adjusted width -->
87                <label for="postcode">Postcode:</label>
88                <input name="postcode" type="text" required />
89            </div>
90            <div class="form-column country-column" style="width: 20%; "><!-- Adjusted width -->
91                <label for="country">Country:</label>
92                <input name="country" type="text" required />
93            </div>
94        </div>
95        <!-- Submit button -->
96        <div class="form-row submit-btn">
97            <input type="submit" value="Submit" />
98        </div>
99    </div>
100
```

HTML code which adds the form into the page. The user must fill the form in order to add a student. It also has basic php configuration files as well as php templates



```

<script>
  document.addEventListener("DOMContentLoaded", function() {
    const form = document.forms['frmdetails'];

    // Image preview functionality
    const imageInput = document.getElementById('image');
    const imagePreview = document.getElementById('image-preview');
    const imageLabel = document.querySelector('.image-label');

    const removeImageButton = document.getElementById('remove-image');
    // Display image preview when image is selected
    imageInput.addEventListener('change', function() {
      const file = this.files[0];
      if (file) {
        const reader = new FileReader();
        reader.onload = function(e) {
          imagePreview.src = e.target.result;
          imagePreview.style.display = 'block';
          imageLabel.style.display = 'none';
          removeImageButton.style.display = 'block';
        }
        reader.readAsDataURL(file);
      }
    });
    // Remove the image preview when remove button is clicked
    removeImageButton.addEventListener('click', function() {
      imageInput.value = '';
      imagePreview.src = '';
      imagePreview.style.display = 'none';
      imageLabel.style.display = 'flex';
      removeImageButton.style.display = 'none';
    });
  });

  // Form validation
  form.addEventListener('submit', function(event) {
    let valid = true;

    // Clear previous error messages
    const errorMessages = document.querySelectorAll('.error-message');
    errorMessages.forEach(function(message) {
      message.textContent = '';
    });

    // Validate Student ID (numbers only)
    const studentid = form['studentid'].value;
    if (!(/\d{4}/).test(studentid)) {
      document.getElementById('studentid-error').textContent = "Student ID must contain only numbers.";
      valid = false;
    }

    // Validate First Name (letters only)
    const firstname = form['firstname'].value;
    if (!(/\w[a-zA-Z]/).test(firstname)) {
      document.getElementById('firstname-error').textContent = "First Name must contain only letters.";
      valid = false;
    }

    // Validate Last Name (letters only)
    const lastname = form['lastname'].value;
    if (!(/\w[a-zA-Z]/).test(lastname)) {
      document.getElementById('lastname-error').textContent = "Last Name must contain only letters.";
      valid = false;
    }

    // Validate Date of Birth (not a future date or date older than 120 years)
    const dob = new Date(form['dob'].value);
    const today = new Date();
    const maxAge = new Date();
    maxAge.setFullYear(today.getFullYear() - 1900);
    if (dob > today || dob < maxAge) {
      document.getElementById('dob-error').textContent = "Date of Birth must be a valid date.";
      valid = false;
    }

    // Prevent form submission if invalid
    if (!valid) {
      event.preventDefault();
    }
  });
});

```

Javascript code which is used for form validation, so when a user fills in the form incorrectly a set of error messages will appear and the form won't be submitted. It also features functionality which previews a selected image and deletes the preview when a button is clicked.

```

addstudent.php
188 <style>
189   .postcode-column,
190   .county-column,
191   .country-column {
192     position: relative;
193     top: -10px;
194   }
195   .error-message {
196     color: red;
197     font-size: 14px;
198     margin-top: 5px;
199     display: block;
200   }
201
202   .submit-btn {
203     text-align: center;
204     margin-top: -10px;
205   }
206
207   .submit-btn input[type="submit"] {
208     background-color: #22254E;
209     color: white;
210     border: none;
211     padding: 12px 25px;
212     font-size: 18px;
213     border-radius: 5px;
214     cursor: pointer;
215     transition: background-color 0.3s;
216   }
217
218   .submit-btn input[type="submit"]:hover {
219     background-color: #CED0F4;
220   }
221
222   .form-container {
223     display: flex;
224     flex-direction: column;
225     align-items: center;
226     padding: 30px;
227     margin: 45px auto;
228     width: 85%;
229     max-width: 1200px;
230     background-color: #F4F4F4;
231     border-radius: 10px;
232     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
233     height: 450px;
234   }

```

```

.form-row {
  display: flex;
  justify-content: space-between;
  width: 100%;
  margin-bottom: 20px;
}

.form-column {
  display: flex;
  flex-direction: column;
  width: 48%;
  margin-bottom: 20px;
}

.form-column.small-column {
  width: 30%;
}

.form-row label {
  font-size: 18px;
  margin-bottom: 5px;
}

.form-row input[type="text"],
.form-row input[type="password"],
.form-row input[type="date"],
.form-row input[type="file"] {
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
}

.main-contents {
  margin-left: 240px;
  padding: 0 20px;
  font-family: 'Roboto', sans-serif;
}

.image-upload-square {
  position: relative;
  width: 200px;
  height: 200px;
  border: 2px dashed #ccc;
  display: flex;
  justify-content: center;
  align-items: center;
  border-radius: 10px;
  margin-right: 20px;
  margin-bottom: 50px;
  overflow: hidden;
}

.image-upload-square label {
  font-size: 16px;
  color: #333;
  cursor: pointer;
  width: 100%;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
}

.image-upload-square input[type="file"] {
  display: none;
}

.form-fields-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  width: calc(100% - 220px);
}

#remove-image {
  background-color: #22254E;
  color: white;
  border: none;
  padding: 5px 10px;
  font-size: 14px;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s;
}

#remove-image:hover {
  background-color: #CED0F4;
}

```

Styles I used for this page. It is mainly used for positioning and overall layout of this form. I decided I want to replicate the basic admin page so I added a square for image upload and aligned the form next to it and under it.



```
<div class="main-content" style="margin-left: 240px; font-family: 'Roboto', sans-serif; font-size: 18px; background-color: #f4f4f4;">
    <?php
        // Collect form data and sanitize
        $studentid = mysqli_real_escape_string($conn, $_POST['studentid']);
        $password = mysqli_real_escape_string($conn, $_POST['password']);
        $dob = mysqli_real_escape_string($conn, $_POST['dob']);
        $firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
        $lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
        $house = mysqli_real_escape_string($conn, $_POST['house']);
        $town = mysqli_real_escape_string($conn, $_POST['town']);
        $county = mysqli_real_escape_string($conn, $_POST['county']);
        $country = mysqli_real_escape_string($conn, $_POST['country']);
        $postcode = mysqli_real_escape_string($conn, $_POST['postcode']);
        $file = $_FILES['image'];

        // Validate file upload
        $allowedTypes = ['image/jpeg', 'image/png'];
        $fileType = mime_content_type($file['tmp_name']);
        if (!in_array($fileType, $allowedTypes)) {
            echo "<p>Error: Invalid file type. Only JPEG and PNG are allowed.</p>";
            exit;
        }

        // File upload
        $imageData = file_get_contents($file['tmp_name']);
        $imageData = mysqli_real_escape_string($conn, $imageData);

        // Hash the password for security
        $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

        // Insert data into student table
        $sql = "INSERT INTO student (studentid, password, dob, firstname, lastname, house, town, county, country, postcode, image_path)
                VALUES ('$studentid', '$hashedPassword', '$dob', '$firstname', '$lastname', '$house', '$town', '$county', '$country', '$postcode', '$imageData')";

        // Execute SQL query
        if (mysqli_query($conn, $sql)) {
            // Display message for student addition
            echo "<div style='position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); text-align: center;'>";
            echo "<p>Student has been added.</p>";
            echo "</div>";
        } else {
            echo "Error adding student: " . mysqli_error($conn);
        }

        mysqli_close($conn);
    ?>
</div>
```

PHP codes which allows a new student to be added into a database. It adds all required details and stores an image inside the database for the student. It also has file validation so only jpeg and png images can be uploaded (**the validation is showcased in a youtube presentation**)



Task 7 - Protecting SQL From SQL Injections

Authenticate.php

Before

```
authenticate.php
1  <?php
2  include("_includes/config.inc");
3  include("_includes/dbconnect.inc");
4  include("_includes/functions.inc");
5
6  session_start();
7
8  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
9      $studentid = $_POST['txtid'];
10     $password = $_POST['txtpwd'];
11
12     // Unsafe query
13     $sql = "SELECT * FROM student WHERE studentid = '$studentid'";
14     $result = mysqli_query($conn, $sql);
15     $row = mysqli_fetch_assoc($result);
16
17     if ($row) {
18         // User exists, check password
19         if (password_verify($password, $row['password'])) {
20             // Successful login
21             $_SESSION['id'] = $row['studentid'];
22             header("Location: index.php");
23             exit;
24         } else {
25             // Incorrect password
26             $error_message = "Incorrect password.";
27         }
28     } else {
29         // User does not exist
30         $error_message = "User does not exist.";
31     }
32
33     // Redirect to error.php with error message
34     header("Location: error.php?error=" . urlencode($error_message));
35     exit;
36 }
37 >>
```

After

```
authenticate.php
1  <?php
2  include("_includes/config.inc");
3  include("_includes/dbconnect.inc");
4  include("_includes/functions.inc");
5
6  session_start();
7
8  // Sanitize user inputs to prevent malicious attacks such as SQL injections.
9  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
10     $studentid = mysqli_real_escape_string($conn, $_POST['txtid']);
11     $password = mysqli_real_escape_string($conn, $_POST['txtpwd']);
12
13     // Check user credentials
14     $sql = "SELECT * FROM student WHERE studentid = ?";
15     $stmt = mysqli_prepare($conn, $sql);
16     mysqli_stmt_bind_param($stmt, "s", $studentid);
17     mysqli_stmt_execute($stmt);
18     $result = mysqli_stmt_get_result($stmt);
19     $row = mysqli_fetch_assoc($result);
20
21     if ($row) {
22         // User exists, check password
23         if (password_verify($password, $row['password'])) {
24             // Successful login
25             $_SESSION['id'] = $row['studentid'];
26             header("Location: index.php");
27             exit;
28         } else {
29             // Incorrect password
30             $error_message = "Incorrect password.";
31         }
32     } else {
33         // User does not exist
34         $error_message = "User does not exist.";
35     }
36
37     // Redirect to error.php with error message
38     header("Location: error.php?error=" . urlencode($error_message));
39     exit;
40 }
```

Before I implemented `mysqli_real_escape_string()` there was a possibility of an SQL injection where an attacker could manipulate the input (student id) to execute SQL commands where they could get around authentication by entering `123' OR '1='1` as the login id.

Whereas after I have added `mysqli_real_escape_string()` the code is now protected from SQL Injection as special characters in the login details are properly escaped. For instance if an attacker inputs `123' OR '1='1` it would be safely escaped to `123' OR '\1\'=\'\1` which would prevent the attacker from injecting SQL.

There was no change in functionality of my web application. It runs well and is now protected from SQL Injection attacks.



Assignmodules.php

Before

```
assignmodule.php
1 <?php
2     include("../includes/config.inc");
3     include("../includes/dbconnect.inc");
4     include("../includes/functions.inc");
5
6     // check logged in
7     if (!isset($_SESSION['id'])) {
8
9         echo template("templates/partials/header.php");
10        echo template("templates/partials/nav.php");
11
12        // If a module has been selected
13        if (isset($_POST['selmodule'])) {
14            $sql = "Insert into studentmodules values ('" . $_SESSION['id'] . "','" . $_POST['selmodule'] . "');";
15            $result = mysqli_query($conn, $sql);
16            $data['content'] .= "<p>The module " . $_POST['selmodule'] . " has been assigned to you</p>";
17        } else {
18
19            // Build SQL statement that selects all the modules
20            $sql = "Select * from module";
21            $result = mysqli_query($conn, $sql);
```

After

```
assignmodule.php
1 <?php
2     include("../includes/config.inc");
3     include("../includes/dbconnect.inc");
4     include("../includes/functions.inc");
5
6     // check logged in
7     if (!isset($_SESSION['id'])) {
8
9         echo template("templates/partials/header.php");
10        echo template("templates/partials/nav.php");
11
12        // If a module has been selected
13        if (isset($_POST['selmodule'])) {
14            // Sanitize input to prevent SQL injection
15            $studentId = mysqli_real_escape_string($conn, $_SESSION['id']);
16            $selectedModule = mysqli_real_escape_string($conn, $_POST['selmodule']);
17
18            $sql = "Insert into studentmodules values ('$studentId', '$selectedModule')";
19            $result = mysqli_query($conn, $sql);
20            $data['content'] .= "<p>The module $selectedModule has been assigned to you</p>";
21        } else {
```

Before: If a malicious user could somehow inject SQL code into `$_SESSION['id']` or `$_POST['selmodule']`, it could result in arbitrary SQL execution, potentially exposing or damaging data.

After: The input is sanitised, preventing SQL injection. The database queries will only interpret the sanitised input as strings, ensuring the queries execute as intended without any unexpected behaviour.

There was no change in functionality of my web application. It runs well and is now protected from SQL Injection attacks.

Details.php

Before

```
details.php
1 <?php
2
3     include("../includes/config.inc");
4     include("../includes/dbconnect.inc");
5     include("../includes/functions.inc");
6
7     // Check if logged in
8     if (!isset($_SESSION['id'])) {
9
10        echo template("templates/partials/header.php");
11        echo template("templates/partials/nav.php");
12
13        // If the form has been submitted
14        if (isset($_POST['submit'])) {
15
16            // Build an SQL statement to update the student details
17            $sql = "UPDATE student SET
18                firstname = '" . $_POST['txtfirstname'] . "' ,
19                lastname = '" . $_POST['txtnickname'] . "' ,
20                house = '" . $_POST['txthouse'] . "' ,
21                town = '" . $_POST['txttown'] . "' ,
22                county = '" . $_POST['txtcountry'] . "' ,
23                country = '" . $_POST['txtcountry'] . "' ,
24                postcode = '" . $_POST['txtpostcode'] . "' ,
25                WHERE studentId = '" . $_SESSION['id'] . "' ;";
26
27            $result = mysqli_query($conn, $sql);
28
29            $data['content'] = "<div class='message-container'><h1>Your details have been
30            updated</h1></div>";
31
32        } else {
33
34            // Build an SQL statement to return the student record with the id that matches that of the session variable
35            $sql = "SELECT * FROM student WHERE studentId='" . $_SESSION['id'] . "' ;";
36            $result = mysqli_query($conn, $sql);
37            $row = mysqli_fetch_array($result);
```

After

```
details.php
1 <?php
2     include("../includes/config.inc");
3     include("../includes/dbconnect.inc");
4     include("../includes/functions.inc");
5
6     // Check if logged in
7     if (!isset($_SESSION['id'])) {
8
9         echo template("templates/partials/header.php");
10        echo template("templates/partials/nav.php");
11
12        // If the form has been submitted
13        if (isset($_POST['submit'])) {
14
15            // Sanitize user inputs to prevent SQL injection
16            $firstname = mysqli_real_escape_string($conn, $_POST['txtfirstname']);
17            $lastname = mysqli_real_escape_string($conn, $_POST['txtnickname']);
18            $house = mysqli_real_escape_string($conn, $_POST['txthouse']);
19            $town = mysqli_real_escape_string($conn, $_POST['txttown']);
20            $county = mysqli_real_escape_string($conn, $_POST['txtcountry']);
21            $country = mysqli_real_escape_string($conn, $_POST['txtcountry']);
22            $postcode = mysqli_real_escape_string($conn, $_POST['txtpostcode']);
23            $studentId = mysqli_real_escape_string($conn, $_SESSION['id']);
24
25            // Build an SQL statement to update the student details
26            $sql = "UPDATE student SET
27                firstname = '$firstname',
28                lastname = '$lastname',
29                house = '$house',
30                town = '$town',
31                county = '$county',
32                country = '$country',
33                postcode = '$postcode',
34                WHERE studentId = '$studentId'";
35
36            $result = mysqli_query($conn, $sql);
37
38            $data['content'] = "<div class='message-container'><h1>Your details have been updated</h1></div>";
39
40        } else {
41
42            // Build an SQL statement to return the student record with the id that matches that of the session variable
43            $studentId = mysqli_real_escape_string($conn, $_SESSION['id']);
44            $sql = "SELECT * FROM student WHERE studentId='$studentId'";
45            $result = mysqli_query($conn, $sql);
46            $row = mysqli_fetch_array($result);
```



Before: Without carrying out any sanitization, the code simply enters input from the user (`$_POST['txtfirstname']`, `$_POST['txtlastname']`) into SQL queries. For instance the hacker could input malicious SQL commands in the form fields to extract student data or even delete them from the database.

After: Now that I added `mysqli_real_escape_string()`: the user inputs are sanitised before being used in SQL queries, which escapes special characters in the input string fields. This protects my web application from malicious hackers. Consequently this lowers any risk of malicious hackers gaining access to the database to steal or delete sensitive information.

There was no change in functionality of my web application. It runs well and is now protected from SQL Injection attacks.

addstudentsql.php

Before

```
addstudentsql.php
1 <?php
2     function template($template_path, $data) {
3         ob_start();
4         include($template_path);
5         return ob_get_clean();
6     }
7
8     include("../includes/config.inc");
9     include("../includes/dbconnect.inc");
10
11    echo template("templates/partials/header.php", $data);
12    echo template("templates/partials/nav.php", $data);
13
14    $studentId = $_POST['studentId'];
15    $password = $_POST['password'];
16    $dob = $_POST['dob'];
17    $firstname = $_POST['firstname'];
18    $lastname = $_POST['lastname'];
19    $house = $_POST['house'];
20    $town = $_POST['town'];
21    $county = $_POST['county'];
22    $country = $_POST['country'];
23    $postcode = $_POST['postcode'];
24
25    // Hash the password for security
26    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
27
28    // Prepare SQL statement
29    $sql = "INSERT INTO Student (studentId, password, dob, firstname, lastname, house, town, county, country, postcode, imageData)
30          VALUES ('$_', '$', '$', '$', '$', '$', '$', '$', '$', '$', '')";
31
32    // Create a prepared statement
33    $stmt = mysqli_prepare($conn, $sql);
34
35    // Bind parameters to statement
36    mysqli_stmt_bind_param($stmt, "ssssssssss", $studentId, $hashedPassword, $dob, $firstname, $lastname, $house, $town, $county, $country, $postcode);
37
38    // Get image data
39    $imageData = file_get_contents($_FILES['image']['tmp_name']);
40
41    if (mysqli_stmt_execute($stmt)) {
42        // Display message
43        echo "<div> style='position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); text-align: center;'>";
44        echo "<p>Student has been added.</p>";
45        echo "</div>";
46    } else {
47        echo "Error adding student: " . mysqli_error($conn);
48    }
49
50    mysqli_stmt_close($stmt);
51    mysqli_close($conn);
52 }
```

```
addstudentsql.php
1 <?php
2     function template($template_path, $data) {
3         ob_start();
4         include($template_path);
5         return ob_get_clean();
6     }
7
8     include("../includes/config.inc");
9     include("../includes/dbconnect.inc");
10
11    echo template("templates/partials/header.php", $data);
12    echo template("templates/partials/nav.php", $data);
13
14
15 <div class="main-content" style="margin-left: 240px; font-family: 'Roboto', sans-serif; font-size: 18px; background-color: #f0f0f0;>
16
17     // Sanitize and escape user input to prevent SQL injection attacks
18     $studentId = mysqli_real_escape_string($conn, $_POST['studentId']);
19     $password = mysqli_real_escape_string($conn, $_POST['password']);
20     $dob = mysqli_real_escape_string($conn, $_POST['dob']);
21     $firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
22     $lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
23     $house = mysqli_real_escape_string($conn, $_POST['house']);
24     $town = mysqli_real_escape_string($conn, $_POST['town']);
25     $county = mysqli_real_escape_string($conn, $_POST['county']);
26     $country = mysqli_real_escape_string($conn, $_POST['country']);
27     $postcode = mysqli_real_escape_string($conn, $_POST['postcode']);
28
29
30     // File upload
31     $imageData = file_get_contents($_FILES['image']['tmp_name']);
32     $imageData = mysqli_real_escape_string($conn, $imageData); // real escape string to prevent sql attacks
33
34     // Hash the password for security
35     $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
36
37     // Insert data into student table
38     $sql = "INSERT INTO student (studentId, password, dob, firstname, lastname, house, town, county, country, postcode, imageData)
39          VALUES ('$_', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '')";
40
41     // Execute SQL query
42     if (mysqli_query($conn, $sql)) {
43         // Display message
44         echo "<div> style='position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); text-align: center;'>";
45         echo "<p>Student has been added.</p>";
46         echo "</div>";
47     } else {
48        echo "Error adding student: " . mysqli_error($conn);
49    }
50 }
```

Before: The code collects input from the user straight from `$_POST` variables without any escape or sanitization. Which makes it very vulnerable to SQL injection attacks because the query uses the input values directly.

After: The code uses `mysqli_real_escape_string()` which makes the input escape and sanitise. This prevents malicious hackers from injecting special characters in the input string fields. The password which the user sets is hashed before being stored in the database.

There was no change in functionality of my web application. It runs well and is now protected from SQL Injection attacks.



Task 8 - Evaluation & Research

Part A - 2 Different Server-Side Approaches To Development

PHP

PHP is a popular and widely used server-side programming language that simplifies building web applications and allows integration directly into HTML. PHP is cross platform and can be run on multiple different operating systems. It's free to use, easy to learn and is arguably the most popular programming language. Despite all these advantages there are many issues mainly security flaws which include SQL injection hence why necessary security measures must be taken into consideration to reduce risks. (*What is php*)

ASP.NET Core

ASP.NET Core was developed by Microsoft, it is a versatile framework that allows development of enterprise-level web apps. "It is known for its strong security features and seamless integration with the .NET ecosystem" (*Albahari et al., 2020*). It is also known for its scalability, high-performance and support for multiple programming languages such as C# or Java. The major disadvantage of ASP.NET is its heavy dependence on Microsoft technologies and support. Developers must follow Microsoft guidelines which restricts flexibility. It is also difficult to learn as a new programmer.

Conclusion

In conclusion PHP is ideal for simple and small projects due to being free-to-use and easy to learn. However it will not perform in complex enterprise-level applications like ASP.NET Core which is scalable, secure and excels in high performance. But it can be costly and difficult to learn.



Part B - Security Considerations In Personal Data Management Systems

Cross-Site Scripting (XSS)

XSS is a type of security vulnerability which allows hackers to inject malicious scripts into web pages which are viewed by users. This vulnerability occurs when the web application does not validate / escape input made by the user. This allows the hacker to successfully execute script in users web browser. The main risk of XSS is the fact that it can lead to data leaks, data theft and session hijacking. To mitigate this risk developers must escape and sanitise their queries and user inputs before they are used on web pages. CSP (Content security policy) will help restrict any execution of unauthorised scripts. (*Rapid7, 2024*)

Insider Threat

Insider is someone who works for the organisation or another individual who wants to steal data or initiate an internal attack on the organisation. To mitigate insider threats it is necessary to monitor staff behaviours frequently, for example Implementing logging mechanisms such as `$log_entry` (PHP) which logs all of users activity to a file, with actions and other activities they perform with corresponding time stamps. Implementing mechanisms like this helps with monitoring users and detecting suspicious activities. Staff training must be provided in order to raise awareness of different risks and behaviours an individual can show. "In a networked world, trust is the cornerstone of security" (*Stallings, 2024*).



References:

Albahari, J., Albahari, B., & Albahari, O. (2020) 'ASP.NET Core', in *ASP.NET Security*. Sebastopol, California: O'Reilly Media, Inc, USA.

What is cross-site scripting (XSS)? definition and prevention (no date) Rapid7. Available at: <https://www.rapid7.com/fundamentals/cross-site-scripting/> (Accessed: 10 May 2024).

Insider threat mitigation (no date) Insider Threat Mitigation | Cybersecurity and Infrastructure Security Agency CISA. Available at: <https://www.cisa.gov/insider-threat-mitigation> (Accessed: 18 May 2024).

What is php? (no date) php. Available at: <https://www.php.net/manual/en/intro-whatis.php> (Accessed: 25 May 2024).

What is ASP.NET?: .NET (no date) Microsoft. Available at: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet> (Accessed: 13 May 2024).

Artists, M. (no date) Database Images, Stock Photo. Available at: <https://www.istockphoto.com/> (Accessed: 05 May 2024).

Nunes, P. (2017) 'Login Inspiration', Codepen. Available at: <https://codepen.io/phnunes/pen/pwXYxG> (Accessed: 10 May 2024).