Week 4 Assignment: Database Development and Class Registration

Joshua Hohendorf

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Prof. Joseph Rangitsch

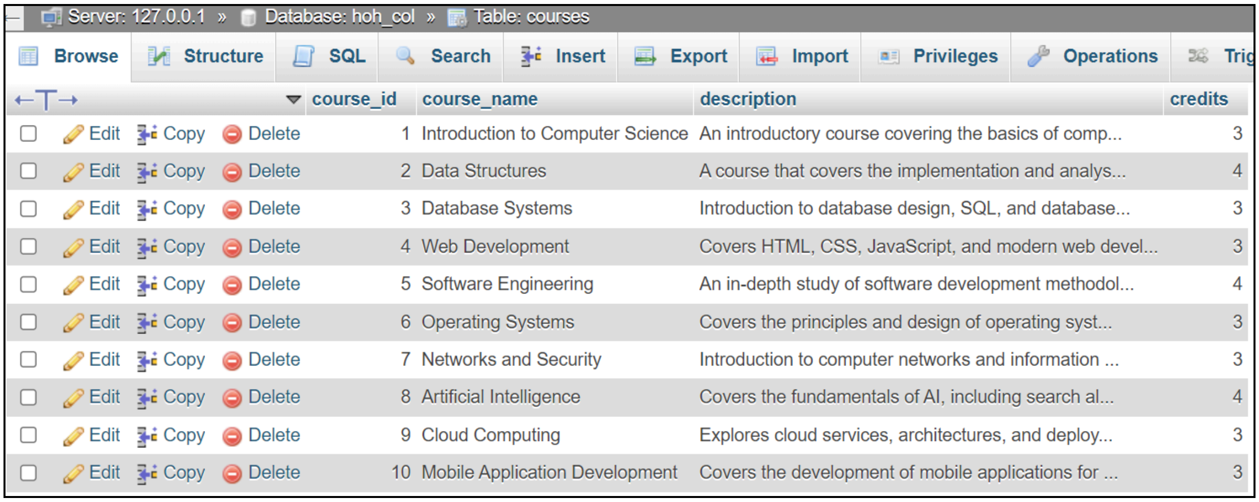
August 19th, 2024

Week 4 Assignment: Database Development and Class Registration

A structured approach was taken to enhance the MySQL database and the web application to meet the project's requirements. I first created the additional tables, one for courses and a registrations table to link users to their selected courses. Each table was designed to store essential data, such as user credentials, course details, and registration information. Figure 1 depicts the course table. The example course information was added via a SQL script. Figure 2 depicts the registration table. Data would exist once a student registers for a course.

Figure 1

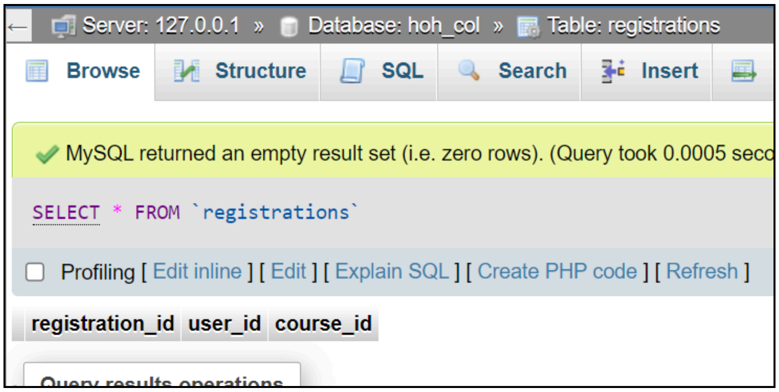
The courses table with example course information.



←T→				▼ course_id	course_name	description	credits
□	Edit	Copy	Delete	1	Introduction to Computer Science	An introductory course covering the basics of comp...	3
□	Edit	Copy	Delete	2	Data Structures	A course that covers the implementation and analys...	4
□	Edit	Copy	Delete	3	Database Systems	Introduction to database design, SQL, and database...	3
□	Edit	Copy	Delete	4	Web Development	Covers HTML, CSS, JavaScript, and modern web devel...	3
□	Edit	Copy	Delete	5	Software Engineering	An in-depth study of software development methodol...	4
□	Edit	Copy	Delete	6	Operating Systems	Covers the principles and design of operating syst...	3
□	Edit	Copy	Delete	7	Networks and Security	Introduction to computer networks and information ...	3
□	Edit	Copy	Delete	8	Artificial Intelligence	Covers the fundamentals of AI, including search al...	4
□	Edit	Copy	Delete	9	Cloud Computing	Explores cloud services, architectures, and deploy...	3
□	Edit	Copy	Delete	10	Mobile Application Development	Covers the development of mobile applications for ...	3

Figure 2

The registration table.



Server: 127.0.0.1 » Database: hoh_col » Table: registrations
Browse
Structure
SQL
Search
Insert
Query results operations

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seco

```
SELECT * FROM `registrations`
```

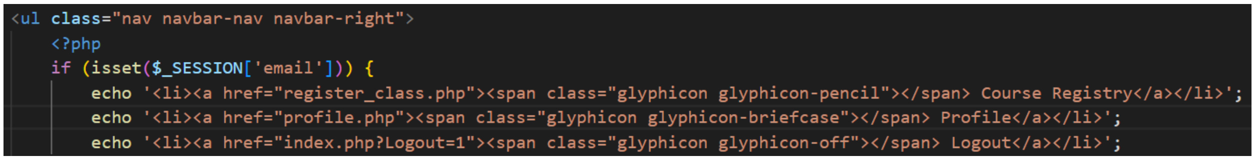
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

registration_id	user_id	course_id

Next, I connected the backend PHP pages to the MySQL database, ensuring seamless communication between the front-end forms and the stored data. First was the addition of a new navigation button, *Course Registry*. This button will only show once a user has successfully logged in after creating a new user account. Figure 3 shows the code, and Figure 4 shows the button on the front end.

Figure 3

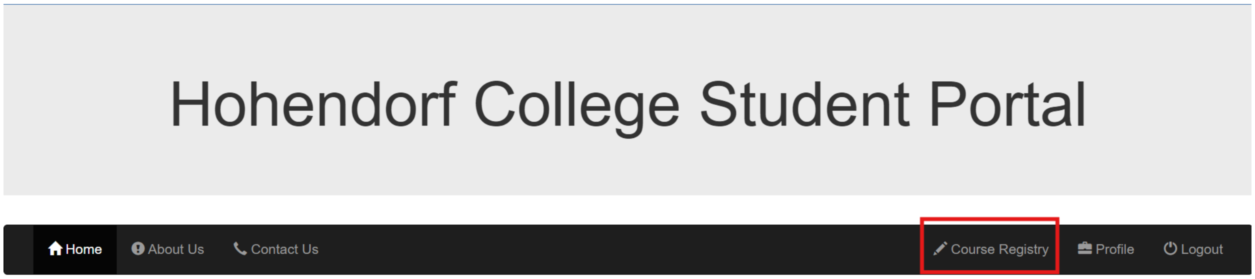
PHP code of Course Registry navigation button.



```
<ul class="nav navbar-nav navbar-right">
    <?php
    if (isset($_SESSION['email'])) {
        echo '<li><a href="register_class.php"><span class="glyphicon glyphicon-pencil"></span> Course Registry</a></li>';
        echo '<li><a href="profile.php"><span class="glyphicon glyphicon-briefcase"></span> Profile</a></li>';
        echo '<li><a href="index.php?Logout=1"><span class="glyphicon glyphicon-off"></span> Logout</a></li>';
```

Figure 4

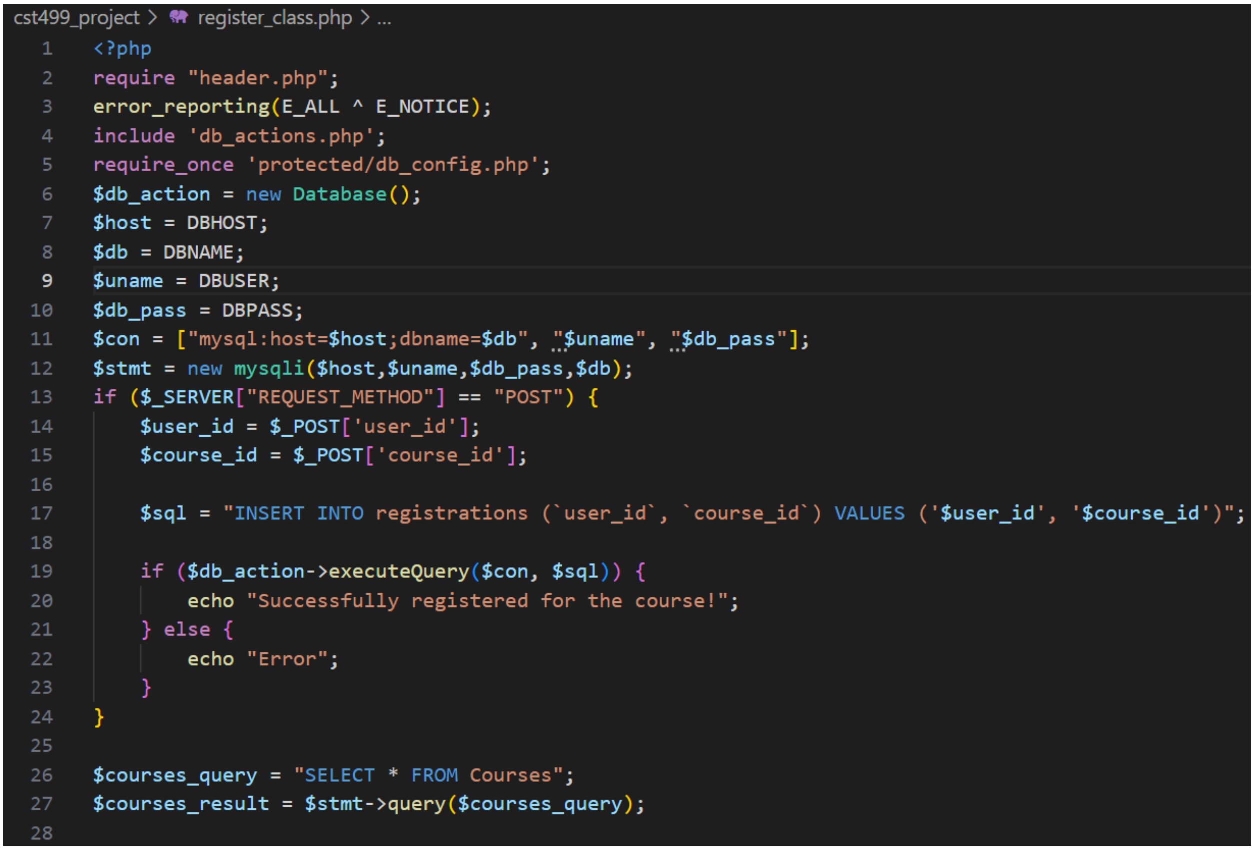
Screenshot of Course Registry navigation button.



The *Course Registry* button would call a newly created page called register_class.php. Figures 5 & 6 show the code that the new file contains. There is a dropdown list of available courses for the user to select. Each selection will reload the page with a message at the top stating whether the action was successful or not. During this point in development, I switched from PDO to MySQLi. The reason for this change was that, while examining different methods of execution, I needed help getting PDO with full utilization of the db_action class and the proper passing of required information. Figure 7 is a screenshot of the webpage with the dropdown menu. Figure 8 shows the success message when a student registers for a course. Finally, Figure 9 exhibits the updated registrations table.

Figure 5

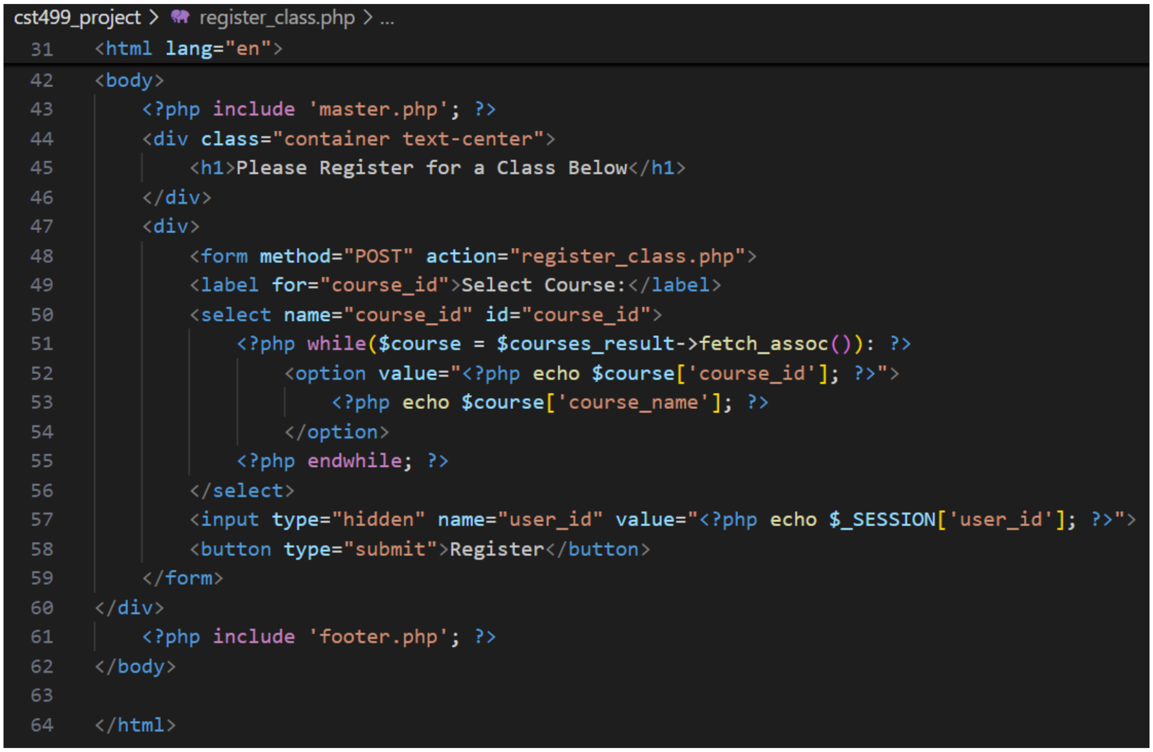
PHP code of register_class.php part 1.



```php
<?php
require "header.php";
error_reporting(E_ALL ^ E_NOTICE);
include 'db_actions.php';
require_once 'protected/db_config.php';
$db_action = new Database();
$host = DBHOST;
$db = DBNAME;
$uname = DBUSER;
$db_pass = DBPASS;
$con = ["mysql:host=$host;dbname=$db", "$uname", "$db_pass"];
$stmt = new mysqli($host,$uname,$db_pass,$db);
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user_id = $_POST['user_id'];
    $course_id = $_POST['course_id'];

    $sql = "INSERT INTO registrations (`user_id`, `course_id`) VALUES ('$user_id', '$course_id')";

    if ($db_action->executeQuery($con, $sql)) {
        echo "Successfully registered for the course!";
    } else {
        echo "Error";
    }
}

$courses_query = "SELECT * FROM Courses";
$courses_result = $stmt->query($courses_query);

```

Figure 6

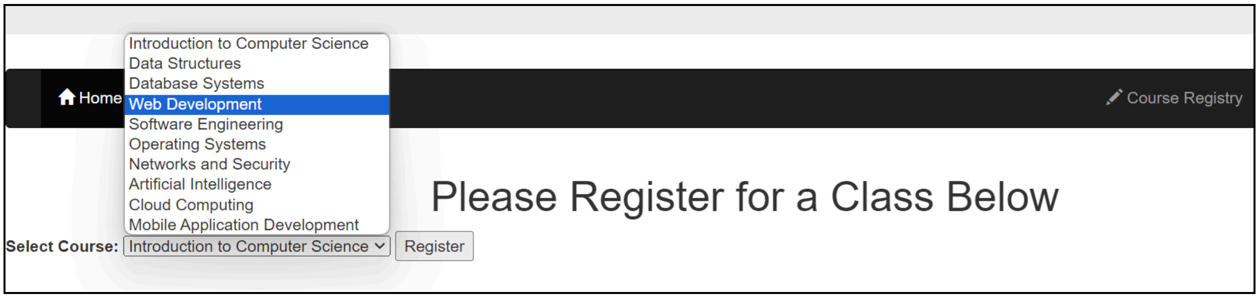
PHP code of register_class.php part 2.



```
<html lang="en">
<body>
    <?php include 'master.php'; ?>
    <div class="container text-center">
        <h1>Please Register for a Class Below</h1>
    </div>
    <div>
        <form method="POST" action="register_class.php">
        <label for="course_id">Select Course:</label>
        <select name="course_id" id="course_id">
            <?php while($course = $courses_result->fetch_assoc()): ?>
                <option value="<?php echo $course['course_id']; ?>">
                    <?php echo $course['course_name']; ?>
                </option>
            <?php endwhile; ?>
        </select>
        <input type="hidden" name="user_id" value="<?php echo $_SESSION['user_id']; ?>">
        <button type="submit">Register</button>
    </form>
</div>
    <?php include 'footer.php'; ?>
</body>

</html>
```

Figure 7

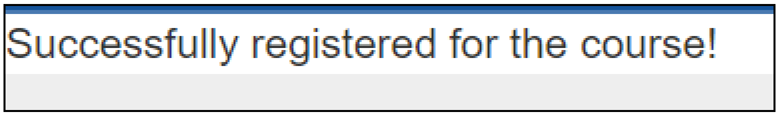
Menu of available courses for a student to register for.



Please Register for a Class Below

Figure 8

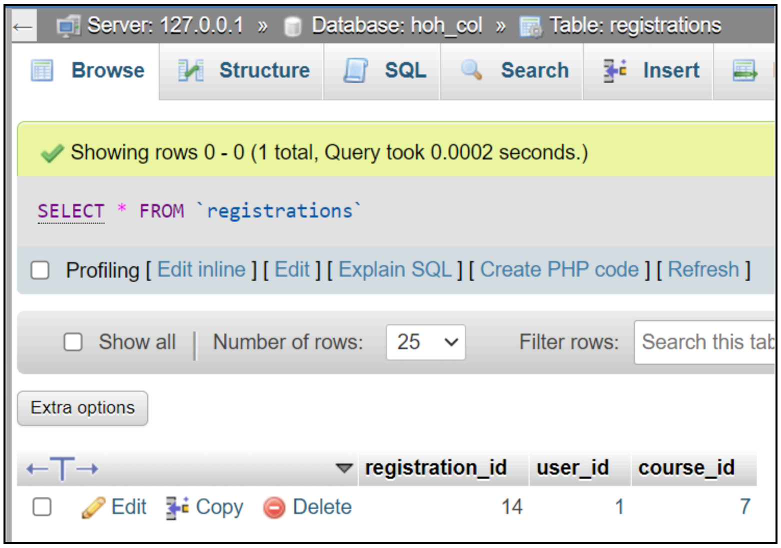
PHP code of register_class.php part 2.



Successfully registered for the course!

Figure 9

Updated registrations table.


Server: 127.0.0.1 » Database: hoh_col » Table: registrations
Browse
Structure
SQL
Search
Insert
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this tab
Extra options
Edit
Copy
Delete

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `registrations`
```

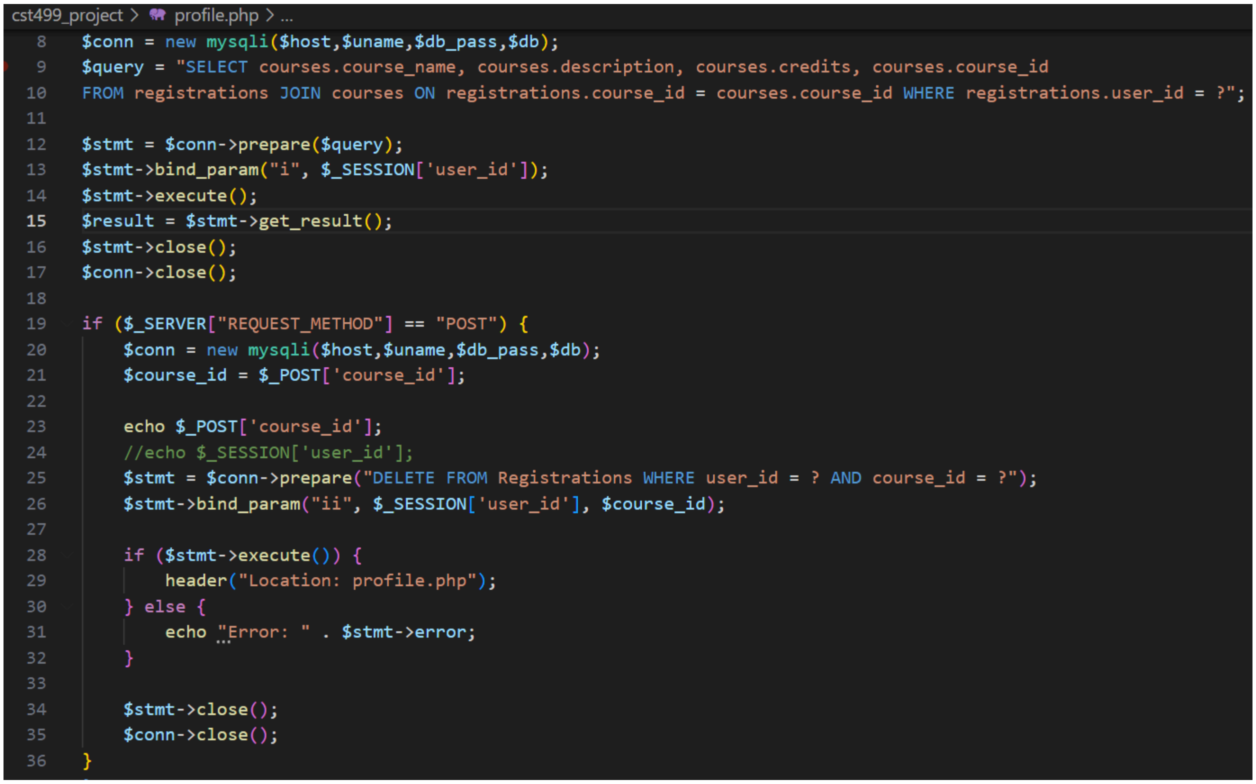
	registration_id	user_id	course_id
□ Edit Copy Delete	14	1	7

Additionally, I developed a feature to list all the courses a user is currently registered for by querying the database and displaying the results in a clear format. Displaying the course information was pretty straightforward. A recursive method creates a new line for each course.

The final development piece was to allow the users to delete their registration. A form with just a button was designed that passed the course_id to an awaiting header method. This method would delete the registration from the registrations table.

Figure 10

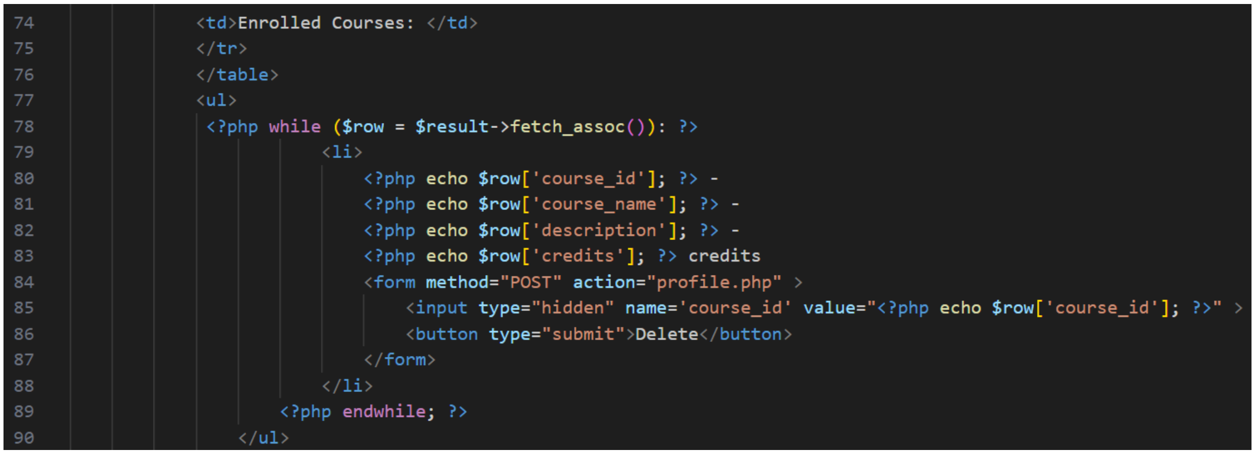
PHP code of updated profile.php part 1.



```
$conn = new mysqli($host,$uname,$db_pass,$db);
$query = "SELECT courses.course_name, courses.description, courses.credits, courses.course_id
FROM registrations JOIN courses ON registrations.course_id = courses.course_id WHERE registrations.user_id = ?";

$stmt = $conn->prepare($query);
$stmt->bind_param("i", $_SESSION['user_id']);
$stmt->execute();
$result = $stmt->get_result();
$stmt->close();
$conn->close();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $conn = new mysqli($host,$uname,$db_pass,$db);
    $course_id = $_POST['course_id'];

    echo $_POST['course_id'];
    //echo $_SESSION['user_id'];
    $stmt = $conn->prepare("DELETE FROM Registrations WHERE user_id = ? AND course_id = ?");
    $stmt->bind_param("ii", $_SESSION['user_id'], $course_id);

    if ($stmt->execute()) {
        header("Location: profile.php");
    } else {
        echo "Error: " . $stmt->error;
    }

    $stmt->close();
    $conn->close();
}
```

Figure 11

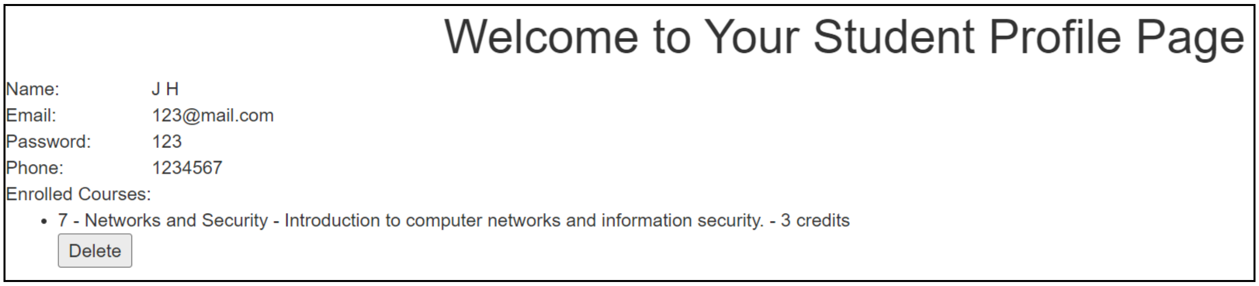
PHP code of updated profile.php part 2.



```
            <td>Enrolled Courses: </td>
            </tr>
            </table>
            <ul>
             <?php while ($row = $result->fetch_assoc()): ?>
                        <li>
                            <?php echo $row['course_id']; ?> -
                            <?php echo $row['course_name']; ?> -
                            <?php echo $row['description']; ?> -
                            <?php echo $row['credits']; ?> credits
                            <form method="POST" action="profile.php" >
                                <input type="hidden" name='course_id' value="<?php echo $row['course_id']; ?>" >
                                <button type="submit">Delete</button>
                            </form>
                        </li>
                    <?php endwhile; ?>
                </ul>
```

Figure 12

Updated profile page with registered courses.



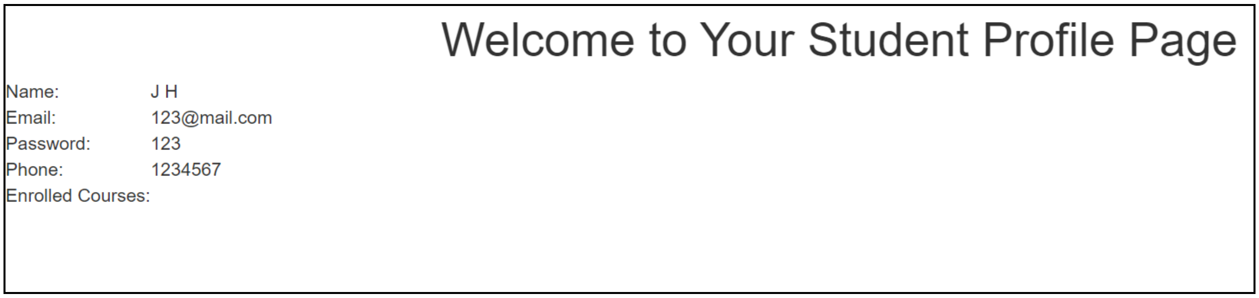
Welcome to Your Student Profile Page

Name: J H
Email: 123@mail.com
Password: 123
Phone: 1234567
Enrolled Courses:

- 7 - Networks and Security - Introduction to computer networks and information security. - 3 credits

Figure 13

Profile page after deleting registration record.



Welcome to Your Student Profile Page

Name: J H
Email: 123@mail.com
Password: 123
Phone: 1234567
Enrolled Courses:

This comprehensive approach completed the project, delivering a fully functional course registration system that met all the specified requirements. Unfortunately, due to the difficulty I experienced with the course registration piece, I could not develop the waitlist feature.

References

Gharehyazie, M., Ray, B., & Filkov, V. (2017). Some from Here, Some from There: Cross-Project Code Reuse in GitHub. *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, 291–301. doi:10.1109/MSR.2017.15

PHP Documentation Group. (n.d.). *MySQLi extension*. PHP Manual. https://www.php.net/manual/en/book.mysqli.php

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones & Bartlett Learning.