**Database Development and Class Registration**

Augustine Ugwu

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Dr. Charmelia Butler

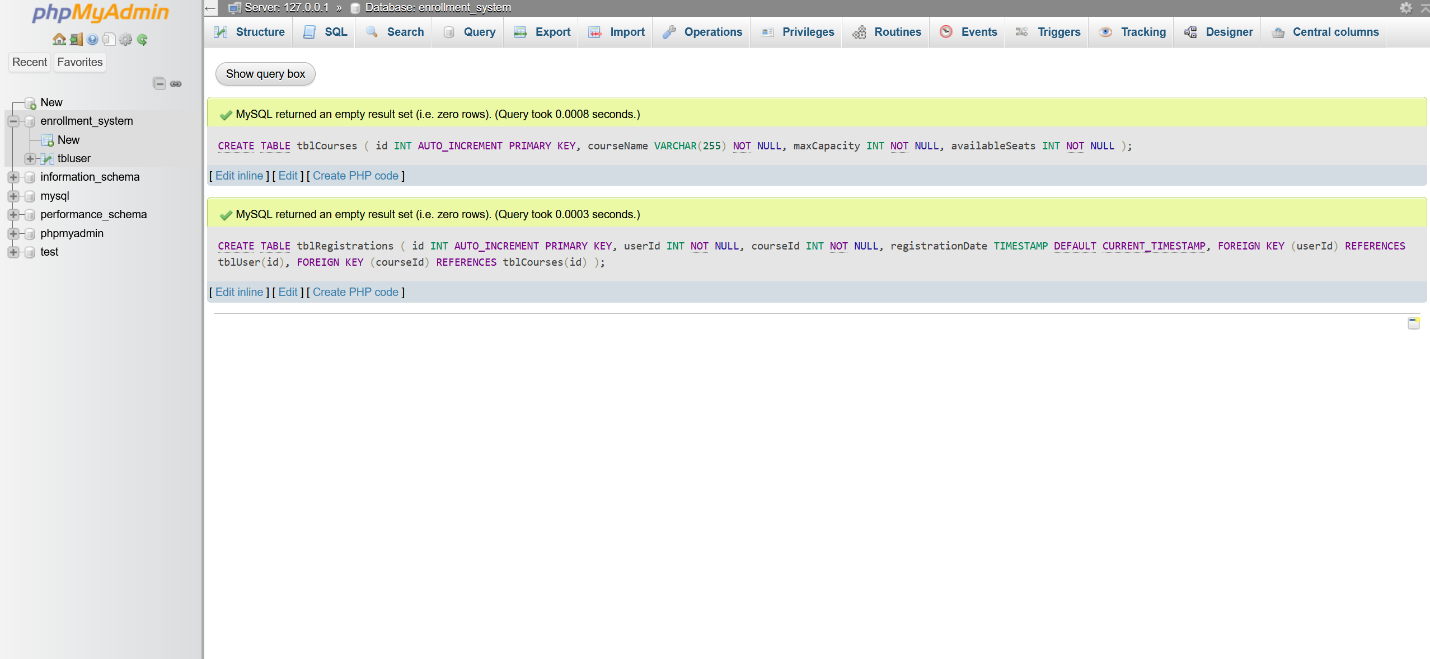
December 12, 2024

For Week 4 of the Online Course Enrollment System project, I built on the foundation laid in Week 3, enhancing the system's capabilities by integrating the MySQL database and adding course registration functionalities. Based on my instructor's suggestions, I changed critical Week 3 elements to increase security and usability. This week, I worked on creating the necessary tables in the MySQL database, connecting the application to the backend database, and developing the functionality for course registration, listing enrolled classes, adding more courses, removing classes from the schedule, and introducing a user dashboard for improved navigation. Following the instructor's criticism, I enhanced the security of the registration and login forms by validating all user inputs and hashing passwords with password\_hash(). I also streamlined the flow between the front-end and back-end, resulting in speedier database transactions, and I used Bootstrap to make the registration and login pages more polished and responsive.

**Database Development**

To meet the system’s requirements, I created the following tables in the MySQL database:

1. **tblCourses:** Stores course details such as course name, maximum capacity, and **available seats.**
2. **tblRegistrations:** Tracks student registrations for courses, linking the student (via userId) and the course ( courseId).

****

After creating the tables, I added sample data for courses to test the registration functionality.

**A screenshot of a computer

Description automatically generated**

To ensure the registration system works efficiently, I connected the PHP pages to the backend MySQL database, linking user registration with course enrollment and the listing of enrolled classes. The backend logic was developed using prepared statements to enhance security and prevent SQL injection. The course registration page dynamically displays available courses, allowing users to register with a single click. Upon registration, the system checks if seats are available, and if so, the user is successfully registered for the course while the seat count in the database is updated accordingly.

A screen shot of a computer program

Description automatically generated

**Created Pages**

I developed the following pages to meet the project’s requirements:

**Dashboard Page**: A centralized hub for users to view and access their registration details, enrolled classes, and options to add/drop courses. The dashboard improves usability by consolidating all key functionalities into one intuitive page.

A screenshot of a computer

Description automatically generated

**Course Registration Page**: Displays a list of courses and allows users to register.

A screenshot of a computer

Description automatically generated

**Registered Classes Page**: Displays the list of courses the user is currently enrolled in.

A white background with black text

Description automatically generated

**Add/Drop Classes Page**: Enables users to add new courses or drop existing one.  
A screenshot of a computer

Description automatically generated

**Source Code**

**A computer screen shot of a black screen

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A black background with colorful lines

Description automatically generated with medium confidence**

**A screen shot of a computer

Description automatically generated**

**A black screen with white text

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

This implementation step helped me better grasp how to integrate front-end pages with back-end databases. I learnt how to deal with real-time data adjustments, such as reducing available seats in a course and ensuring data integrity through prepared statements. Creating the dashboard page taught me the value of building user-friendly interfaces to improve usability and streamline navigation. One of the issues I encountered was ensuring that the data was properly handled upon registration and that the database reflected the changes. After completing these tasks successfully, I gained confidence in full-stack development, particularly in PHP and MySQL integration. In the future, I see room for improvement, such as adding error handling for edge cases and refining user interactions, such as notifying users on a waitlist when a course is full. In addition, I intend to improve the dashboard by using visual components like as charts or progress bars to create a more dynamic experience.

**References**

Tsui, F., Karam, O., & Bernal, B. (2018). [*Essentials of software engineering*](https://uagc.instructure.com/courses/139106/modules/items/7086725) (4th ed.). Jones & Bartlett Learning.

GeeksforGeeks. (2022, May 30). How to create admin login page using PHP? Retrieved from  <https://www.geeksforgeeks.org/how-to-create-admin-login-page-using-php/>