Class Registration System Implementation and Database Design

Nkosinomusa Gwalla

University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Professor Joseph Rangitsch

April 7, 2025

Introduction

In traditional academic settings, the process of registering for classes often involves manual paperwork and multiple approvals. To streamline this process, a web-based class registration system was proposed and prototyped in the Week 3 milestone, which established the initial interface design and a basic data schema for class registrations. Building on that foundation, the Week 4 phase focuses on implementing a MySQL database backend and dynamic PHP functionality to allow students to register for classes online. This paper presents the design and development of the class registration system’s database and core features, including how MySQL tables were created, how key PHP pages were implemented (for registering, listing, adding, and deleting classes), and reflections on the development experience. The goal is to demonstrate a fully functional system that simplifies course enrollment by leveraging a PHP-MySQL web application, resulting in a more user-friendly and efficient registration process.

MySQL Database Design and Implementation

The database schema includes three primary tables: users, courses, and registrations. The users table stores login credentials, the courses table holds class information, and the registrations table maps users to the courses they are enrolled in. The design enforces referential integrity and supports dynamic course management.

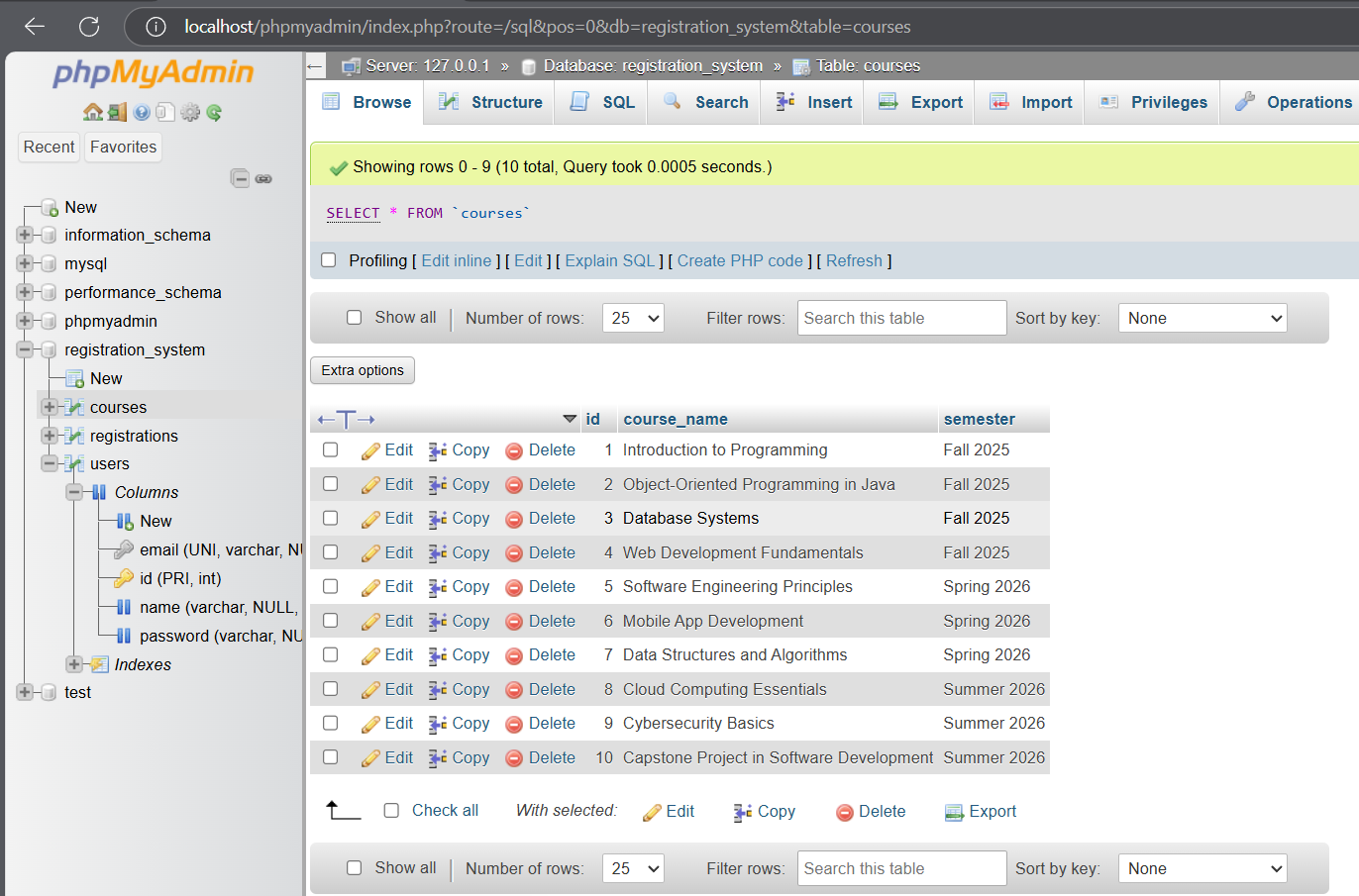
Figure 1

Users Table Structure in phpMyAdmin

A screenshot of a computer

AI-generated content may be incorrect.Figure 2

Courses Table After Inserting Sample Data



Implementation of Class Registration Features

PHP pages were developed for key functionalities:

- courses.php: Displays available classes with checkboxes.

- register\_course.php: Saves selected classes to the database.

- my\_courses.php: Lists classes a user is registered for.

- delete\_course.php: Allows users to drop a class.

Each of these pages connects to a MySQL database using a central Database.php file and prepared queries.

Figure 3

Course Registration Page (courses.php)



Figure 4

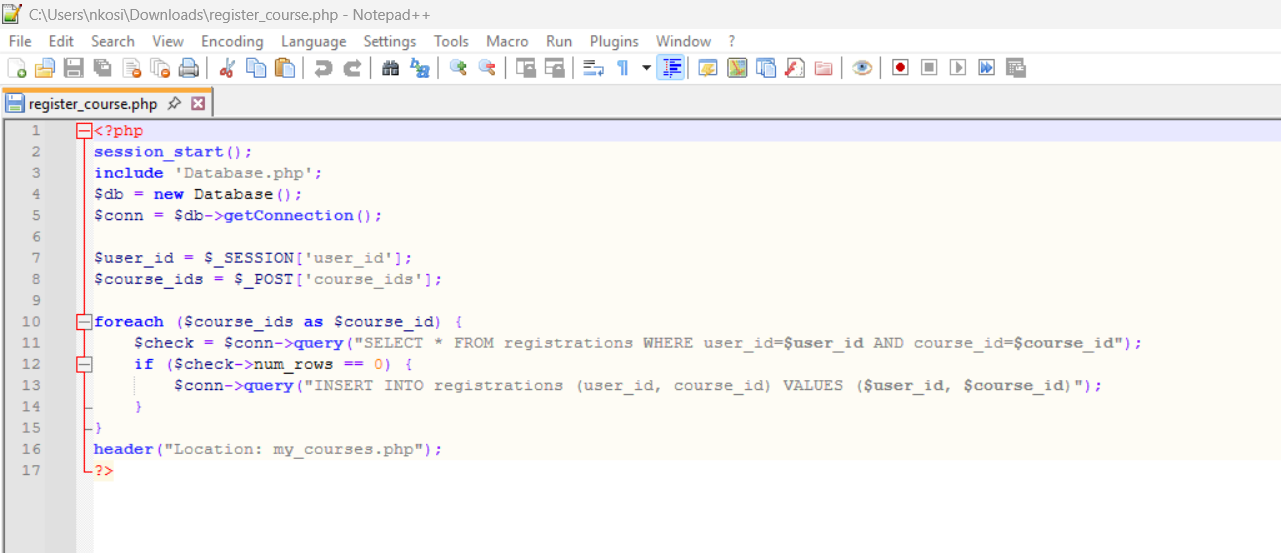
Registered Courses List (my\_courses.php)

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5

register\_course.php



.

Developer Experience: Challenges and Solutions

Challenges included debugging SQL syntax, connecting PHP to MySQL, and ensuring duplicate courses were not registered. Solutions included using phpMyAdmin for visual query building, validating records before insert, and using session variables to track logged-in users. This project provided real-world exposure to secure user login, database-driven logic, and dynamic webpage content.

Conclusion

The Week 4 milestone resulted in a complete, working system that supports course registration, listing, and management. The database was successfully connected to dynamic PHP pages, allowing users to register, add, and delete courses from their schedule. The implementation is functional, scalable, and adheres to best practices in both backend and frontend development. This system can serve as a foundation for more complex features like role-based access, instructor panels, and grade tracking.

References

Elmasri, R., & Navathe, S. B. (2017). Fundamentals of database systems (7th ed.). Pearson.

Schach, S. R. (2011). Essentials of software engineering (3rd ed.). McGraw-Hill Education.

PHP Documentation. (2023). password\_hash - PHP Manual. https://www.php.net/manual/en/function.p