**The Landing, Login, and Enrollment Pages Development Paper**

Raul Machuca

## CST 499: Capstone for Computer Software Technology

Charmelia Butler

June 16, 2025

I implemented several PHP-based web pages and a MySQL database using XAMPP to develop a functional student course registration portal. This paper explains how to run PHP files in XAMPP, describes the developed landing, login, and registration pages, and outlines the structure of the MySQL database and supporting scripts used to register new students into the system.

XAMPP is a cross-platform development environment that enables the local testing of PHP applications. To run PHP files, I first downloaded and installed XAMPP from https://www.apachefriends.org. Once installed, I opened the XAMPP Control Panel and started the Apache and MySQL services. I then placed all project files into the htdocs folder under C:\xampp\htdocs\CST499. To access the application in a browser, I navigated to http://localhost/CST499/home.php. This setup enabled me to develop and test the landing, login, and registration pages without uploading to an external server.

The landing page (home.php) displays a welcoming message and includes two buttons for login and registration. This page introduces the portal's purpose and appropriately guides new or returning users. The login page (login.php) authenticates students by checking their email and password against stored database records. Upon successful login, users are redirected to their profile. The registration page (registration.php) allows new users to input their first name, last name, email, password, address, and phone number.

The MySQL database for this project is named cst499. It includes two primary tables: students and students\_courses. The students table holds all the students’ personal information, while the students\_courses table manages course enrollments. This structure supports a many-to-many relationship where each student can enroll in multiple courses, and each course can have multiple students.

The students table is defined with the following SQL statement:

CREATE TABLE students (

student\_id VARCHAR(10) PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

address TEXT NOT NULL,

phone VARCHAR(20),

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL

);

The students\_courses table creates the relationship between students and the courses they select:

CREATE TABLE students\_courses (

id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id VARCHAR(10),

course\_code VARCHAR(10),

FOREIGN KEY (student\_id) REFERENCES students(student\_id)

);

I used PHP Data Objects (PDO) to connect to the database securely. In database.php, the connectDB() function establishes a secure connection to the MySQL database and returns a PDO object. Another function, executeQuery(), accepts student data and performs an INSERT operation into the students table using prepared statements to prevent SQL injection. A helper function, generateStudentId(), generates a unique student ID by combining a prefix and a randomly generated numeric value, ensuring that each ID is unique within the system.

The layout of the registration page was created using Bootstrap 3. The form is divided into several rows to organize input fields in a readable and responsive format. The email and password fields are displayed side by side, while the phone and address fields occupy their lines. The course selection field uses a multi-select dropdown. The "Register" button is placed in a centered position beneath the form.

The registration.php script includes a form-handling section that listens for a POST request. Upon form submission, it collects all the student inputs, generates a new student ID, and invokes the executeQuery() function to save the data. Error messages or success confirmations are displayed in a Bootstrap alert box above the form.

The students table is the principal repository for user information. The student\_id is the primary key, and each email must be unique. Passwords are stored in the table using secure hashing methods, ensuring data privacy and security. This structure supports future enhancements such as password resets and profile updates.

This development project introduced the key components of a student course registration system. I created and tested the landing, login, and registration pages using PHP and MySQL with XAMPP. The portal allows new users to register and existing users to log in.

database.php script

A computer screen shot of text

AI-generated content may be incorrect.

login.php script

A screen shot of a computer program

AI-generated content may be incorrect.

index.php

A screen shot of a computer program

AI-generated content may be incorrect.

Tables

Students

A screenshot of a computer

AI-generated content may be incorrect.

Student\_courses

A screenshot of a computer

AI-generated content may be incorrect.

Landing Page

A screenshot of a computer

AI-generated content may be incorrect.

Registration Page

A screen shot of a computer

AI-generated content may be incorrect.

Login Page

A screen shot of a computer

AI-generated content may be incorrect.

**References**

Bootstrap Documentation. (n.d.). Forms. Retrieved from <https://getbootstrap.com/docs/4.0/components/forms/>

Bootstrap Documentation. (n.d.). Tables. Retrieved from <https://getbootstrap.com/docs/4.0/content/tables/>

Connolly, R., & Hoar, R. (2022). *Fundamentals of web development* (3rd ed.). Pearson.

GeeksforGeeks. (n.d.). PHP number\_format() function. Retrieved from <https://www.geeksforgeeks.org/php-number_format-function/>

PHP Manual. (n.d.). foreach loop. Retrieved from <https://www.php.net/manual/en/control-structures.foreach.php>

PHP Manual. (n.d.). password\_hash – Create a password hash. Retrieved from <https://www.php.net/manual/en/function.password-hash.php>

PHP Manual. (n.d.). password\_verify – Verifies that a password matches a hash. Retrieved from <https://www.php.net/manual/en/function.password-verify.php>

PHP Manual. (n.d.). PDO prepared statements. Retrieved from <https://www.php.net/manual/en/pdo.prepared-statements.php>

PHP Manual. (n.d.). PDOStatement::fetch. Retrieved from <https://www.php.net/manual/en/pdostatement.fetch.php>

PHP Manual. (n.d.). session\_start – Start new or resume an existing session. Retrieved from <https://www.php.net/manual/en/function.session-start.php>

TraversyMedia. (2019). *How to create a simple user registration system using PHP & MySQL* [Video]. YouTube. Retrieved from <https://www.youtube.com/watch?v=oVJ0anq8yLA&ab_channel=TraversyMedia>

Tutorial Republic. (n.d.). PHP MySQL login system. Retrieved from <https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>

W3Schools. (n.d.). HTML5 tags. Retrieved from <https://www.w3schools.com/tags/default.asp>