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

Benjamin Brown

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Professor Joseph Rangitsch

March 3, 2025

In order to create a secure and efficient web application, several key steps must be taken, including making sure the right environment is set up and managing user data appropriately. The purpose of this paper is to demonstrate how to construct a simple user registration and login system using PHP and MySQL, running on a local server environment such as XAMPP. You will learn how to set up XAMPP, develop a custom database connection class, and develop the registration and login processes. Throughout the application, the focus is on security, particularly preventing SQL injection, and ensuring that it is both reliable and scalable. To build a secure web application that can be tested locally, this guide provides a clear, step-by-step approach.

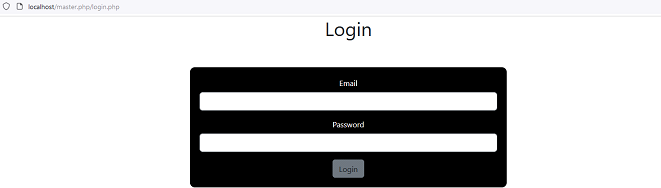
It is imperative that the version of XAMPP used for executing PHP files matches the operating system being used. The first step in executing PHP files with XAMPP is to download and install the software. As soon as the XAMPP Control Panel is accessed, the Apache server must be started by selecting the "Start" button adjacent to the "Apache" option. Following this, the 'htdocs' directory within the XAMPP installation folder, commonly located at C:/xampp/htdocs on Windows or /Applications/XAMPP/htdocs on macOS, should be identified. The files of a project can be organized in a new folder within this directory. An appropriate folder for a related project might be "StudentPortal". The PHP file, such as index.php, should then be saved within this folder, resulting in a file path like C:/xampp/htdocs/StudentPortal/index.php. To run the PHP file, a web browser must be opened, and the URL <http://localhost/StudentPortal/index.php> entered.

The URL indicates that the PHP file should be retrieved from the local server by the browser. When all configurations are correct, the Apache server processes the PHP file and renders its output. It is possible to develop and test PHP scripts within a local environment by following this procedure.

Database connection custom classes were created using MySQL functions such as mysqli\_connect() and mysqli\_prepare(). In order to ensure modularity and maintainability, a reusable PHP class was developed to encapsulate the connection logic. Using parameters such as hostname, username, password, and database name, the mysqli\_connect() function allowed the connection to be made, while error-handling mechanisms enabled potential connection problems to be identified. Moreover, the class included methods for executing queries, using mysqli\_prepare() to securely prepare SQL statements, thereby mitigating SQL injection risks. An application using this structured approach was able to manage database interactions in a secure and robust manner.

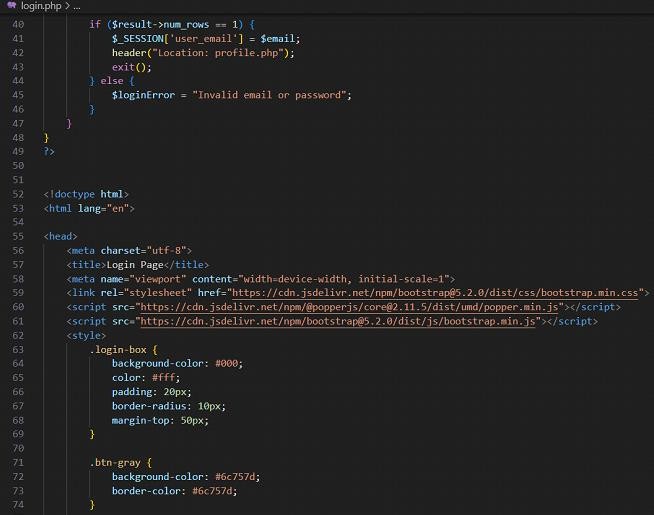
In order to create the registration page and save user information in the database, several steps were taken. In the first step, a registration form was designed using HTML to capture essential information about the user, such as name, address, billing address, phone number, and student details. A PHP script processed the form data after submission. MySQL functions are used by the PHP script to establish a secure connection between the application and the database. As soon as the connection was established, the script retrieved the form data and validated it to ensure that all required fields were filled in correctly. The user information was then inserted into the relevant database tables (such as the user table) using a SQL INSERT query.

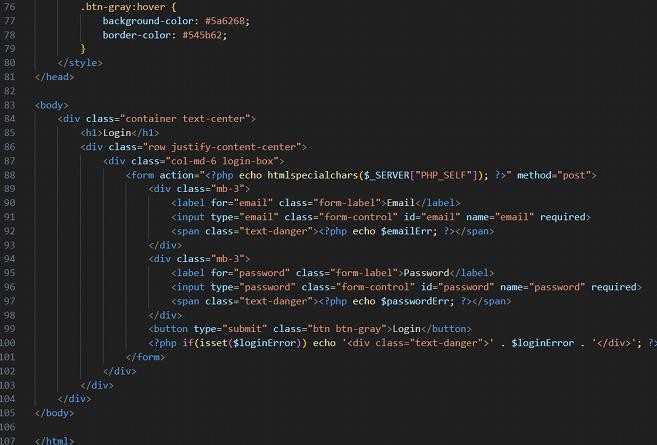
In order to ensure data security, the script used prepared statements to prevent SQL injection. As soon as the data was successfully inserted, the PHP script provided feedback to the user, confirming successful registration or displaying an error message if an error occurred.

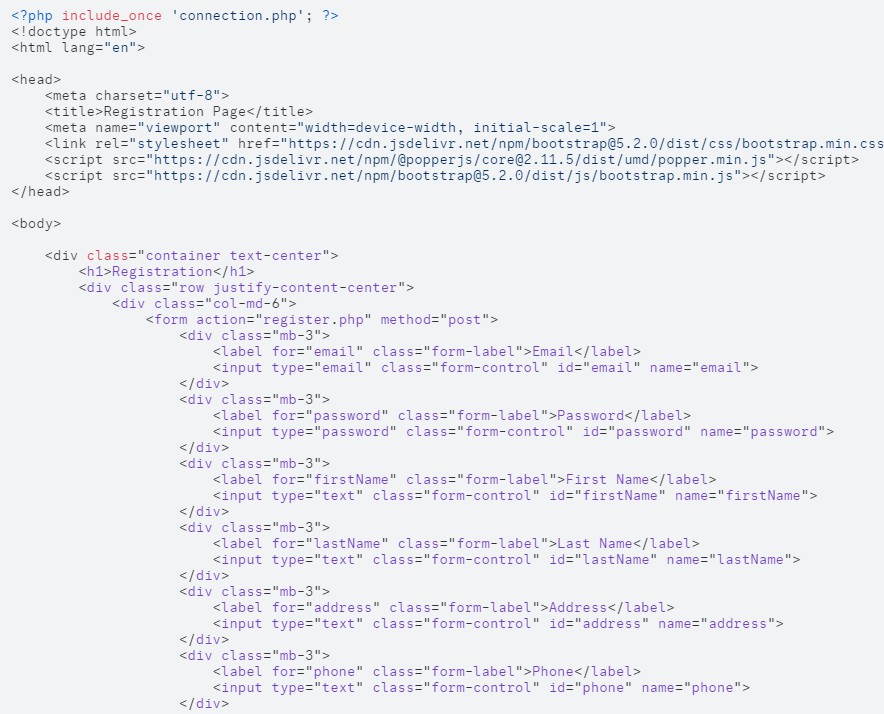
**Login Page Layout**

**Login PHP Code**



****

**Registration page**

**User table**

****

**Conclusion**

To summarize, PHP, MySQL, and XAMPP provide a solid foundation for developing secure web applications. In addition to ensuring the system is secure, developers can manage user data appropriately by following the steps outlined. With XAMPP, you can test and develop locally, which speeds up the development process. Anyone looking to build secure, dynamic web applications that handle user data safely and efficiently can start with this approach.

References

Jaycocks, T., et al. (2024). *What are the steps to creating a PHP login form?* LinkedIn. [https://www.linkedin.com/advice/0/what-steps-creating-php-login-form-skills-web-development-](https://www.linkedin.com/advice/0/what-steps-creating-php-login-form-skills-web-development-eww7c) [eww7c](https://www.linkedin.com/advice/0/what-steps-creating-php-login-form-skills-web-development-eww7c)

Mikoluk, K. (2013). *XAMPP tutorial: How to use XAMPP to run your own web server*. Udemy. <https://blog.udemy.com/xampp-tutorial/>