Landing, Login, and Enrollment Pages Development

Carlon Bullock

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Professor: Joseph Rangitsch

April 28, 2025

**Introduction**

Building dynamic, database-driven applications has become more accessible in learning software engineering and web development. Leveraging skills refined during prior classes, this project focused on implementing the foundational components of the Online Course Registration and Enrollment System (OCRES). The work included developing a landing page, login page, and registration page connected to a MySQL database using PHP. Working on a macOS environment introduced some unique technical considerations, particularly the decision to use MAMP as the local server environment instead of XAMPP, which is more commonly demonstrated on Windows systems. Despite these initial adjustments, the familiarity gained from previous experiences made navigating MAMP, setting up Apache and MySQL servers, and deploying PHP scripts a smoother process than it once was.

This paper documents the development of a basic dynamic website, illustrating the steps taken to configure MAMP, create server-side pages, build a database for user registration, and integrate all components into a functioning system. It highlights how iterative learning and practical exposure to PHP and MySQL development tools can significantly ease the complexity of full-stack implementation. This paper aims to present the development process of a web-based registration system built in MAMP using PHP and MySQL while reflecting on the smoother navigation of technologies compared to earlier learning experiences.

**Running PHP Files in MAMP**

Setting up the local development environment for the OCRES project required using MAMP, as macOS does not natively support XAMPP as seamlessly as Windows systems do. Launching MAMP provided a familiar experience compared to previous coursework, but with slight differences that had to be navigated carefully. The Apache and MySQL servers were started through the MAMP control panel, and the project folder, ocres\_project, was created within the htdocs directory to host all necessary PHP and HTML files. Accessing the application locally involved entering http://localhost:8888/ocres\_project/landing.html into the browser, which successfully loaded the project’s landing page. From there, hyperlinks to the login and registration pages were accessible. Running PHP files through MAMP required ensuring each file had the correct .php extension and that the folder names matched exactly since macOS and web servers are case-sensitive.

Compared to earlier experiences setting up web environments, this process felt more intuitive due to previous practice in CST310. Challenges like misnaming files, missing server starts, or incorrect directory paths were avoided more confidently. Additionally, understanding how MAMP bundles Apache, PHP, and MySQL together into a local ecosystem helped reinforce key concepts of side-server development discussed by Tsui, Karam, and Bernal (2022, Ch. 12). This smoother navigation ultimately allowed more focus on the core development of the login and registration logic, rather than spending excessive time on server setup hurdles.

**Landing, Login, and Registration Page Development**

Developing the web front-end for the OCRES system involved creating three essential pages: a landing page, a login page, and a registration page. Each page played a distinct role in facilitating user interaction with the system. Although building these pages involved revisiting concepts learned in earlier courses, the process was significantly more fluid due to improved familiarity with PHP forms, hyperlink navigation, and local server management.

**Landing Page Development**

The landing page served as the entry point for users accessing the system. It presented a simple, user-friendly interface containing hyperlinks to either the login or registration pages. Created as a basic HTML file, the landing page emphasized functionality and clarity over complexity. One of the challenges encountered during development involved ensuring that hyperlink paths were matched correctly to the case-sensitive file names and directory structure enforced by the MAMP environment. Once the correct file structure and navigation were verified, the landing page loaded successfully and served as the project’s central access hub.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1. The landing page for the OCRES system shows navigation links to login and registration pages.

**Login Page Development**

The login page was developed to give users a simple form requiring their registered email and password. The project focused on developing the form structure at this stage without implementing full authentication validation. The form was built in PHP to ensure server-side compatibility, allowing future expansion for validating credentials against the MySQL user table. A minor technical adjustment ensured that the login page’s extension was .php rather than .html, a distinction that had previously caused a “404 Not Found” error during initial testing. Correcting the file extension allowed for seamless navigation from the landing page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2. Login form for OCRES system prompting users for email and password.

**Registration Page Development**

The registration page required a more sophisticated approach, combining HTML input fields with embedded PHP scripting to handle form submission and database insertion. Fields for name, email, phone number, and password were provided. Upon submission, PHP scripts validated the input and inserted the new user’s information into the user table of the ocres\_db database. One improvement over earlier experiences was structuring the registration PHP file to use secure password hashing with PHP’s password\_hash() function. This reflects stronger data protection practices than more basic examples from earlier coursework. After successfully submitting the registration form, the database reflected the newly added user. This integration between front-end form and back-end database marked a significant milestone in building a dynamic, database-connected web application.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3. Registration form for new users, capturing name, email, phone, and password.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4. A successful registration message confirms user insertion into the database.

**MySQL Database Functions and Custom Connection Class**

Working with MySQL databases through PHP required an understanding of how to establish connections, execute queries, and safely handle user data. The registration process of the OCRES project utilized MySQL functions embedded within PHP scripts to manage these interactions. A key part of the workflow was creating a direct database connection within the register.php page using the MySQL extension, a standard procedural method for PHP-MySQL integration. The mysqli\_connect() function established the connection to the local MySQL server, specifying parameters such as server address, username, password, and database name. This connection allowed the project to securely insert new user data into the user table. The real\_escape\_string() method was used to sanitize input data, preventing SQL injection vulnerabilities—an essential improvement over earlier coursework where security practices may have been less emphasized.

The user password was protected using PHP’s password\_hash() function, which encrypted the password before insertion into the database. The final insertion was handled through a standard SQL INSERT INTO statement executed by the query() method of the connection object. A custom database connection class was considered better for organizing database operations in future expansions of OCRES. This class would encapsulate the connection logic into reusable methods, improving the modularity and maintainability of the codebase (Tsui et al., 2022, Ch. 14). A screenshot of the register.php PHP source code shows the integration of form handling, database connection, and user data insertion.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 5. Sublime Text editor showing PHP source code for the registration page, including database insertion logic.

**Registration Workflow Explanation**

The registration workflow for the OCRES system was designed to provide a simple, secure, and efficient method for users to create new accounts. The process followed a logical sequence from user interaction to database insertion, ensuring data validation and security at every step. The workflow begins when a user navigates from the landing page to the registration page. Upon arriving at the registration form, the user enters their name, email, phone number, and password into the corresponding input fields. When the user submits the form, the PHP script embedded within register.php processes the request. The first action of the script is to establish a database connection using mysqli\_connect(). If the connection fails, the script immediately terminates with an error message. Upon successful connection, the PHP script retrieves the submitted form data and sanitizes it using the real\_escape\_string() method to prevent SQL injection attacks. The password field is hashed using PHP’s password\_hash() function, applying secure encryption before storage.

Next, an SQL INSERT INTO statement is constructed with the sanitized and hashed data, and the query() function is used to execute it against the users’ table in the ocres\_db database. If the insertion succeeds, a success message is displayed to the user. If any error occurs—such as attempting to register with an already existing email address—the script reports the error, allowing the user to adjust and resubmit. Finally, the inserted user record becomes visible inside the database when viewed through phpMyAdmin, confirming that the registration process is completed as intended. This end-to-end workflow reinforces best practices in secure PHP development and mirrors real-world database interaction models found in scalable web applications, as described by Tsui, Karam, and Bernal (2022, Ch. 14).

**Conclusion**

Developing the Online Course Registration and Enrollment System (OCRES) web application provided an opportunity to apply both foundational and newly acquired skills in dynamic web development. The project reflected the cumulative growth achieved through prior coursework and hands-on practice by setting up the MAMP environment, designing core pages, connecting to a MySQL database, and handling user registration securely with PHP. While earlier experiences with local server management and PHP scripting were challenging, navigating these systems during this project was markedly smoother.

Familiarity with server configurations, file structure conventions, and PHP form handling significantly reduced potential friction points. Additionally, reinforcing best practices for security, such as password hashing and input sanitization, demonstrated an evolution toward more professional coding standards. Overall, this phase of the OCRES project underscores the importance of iterative learning in technical education. Setting up a local development environment, building structured pages, and interacting securely with databases are critical skills for modern software engineers—skills that continue to develop through consistent application and reflection.

**Reference:**

Gemino, A., Reich, B. H., & Serrador, P. (2020). Agile, traditional, and hybrid approaches to project success: Is hybrid a poor second choice? Project Management Journal, 51(2), 177–194. https://doi.org/10.1177/8756972820973082

Ginikanwa, N. F. (2024). A comparative analysis of agile and traditional project management methodologies and their impact on information-system development projects: A Nigerian banking system case study. International Journal of Science and Research, 13(8), 1682–1690.

Tsui, F., Karam, O., & Bernal, B. (2022). Essentials of software engineering (5th ed.). Jones & Bartlett Learning.