**Week 3-5 – Landing, Login, and Enrollment Pages Development**

Ben Merritt

The University of Arizona Global Campus

CST 449: Capstone for Computer Software Technology

Instructor: Dr. Charmelia Butler

December 11, 2023

**Week 3 – Landing, Login, and Enrollment Pages Development**

This week, I started developing the Course Enrollment System for CST499. The beginning of this project was similar to prior coursework that I completed at the University of Arizona Global Campus, especially in the course CST310. This was an exciting project to begin since it brings back more aspects of web development, which I am very interested in continuing my development. To complete this assignment, the overall process, which will be broken down further, was to download XAMPP to host a local server, create a database within PHPmyadmin, develop php files as webpages, and connect the database and webpages to provide inputs (Tsui, 2018).

**How to Run a PHP File in XAMPP**

The software platform XAMPP was downloaded and installed from the official website (<https://www.apachefriends.org/index.html>) on my computer during the CST 310 course. XAMPP is a cross-platform local host server that many developers utilize to create test environments for software development (Javatpoint, 2021). In the XAMPP control panel, I clicked the "Start" button next to "Apache." This started the Apache web server. Next, I clicked on the "Start" button next to "MySQL." This started the database. Next, I navigated the local system to the folder XAMPP was saved under (C:\xampp) and located the 'htdocs' folder. Within this folder, the PHP files can be created and saved to utilize a web browser to view. I opted to create a folder 'CST\_499\_Week\_3\_Bmerritt\_Assignment\_Course\_Registration\_System' folder within htdocs to serve as the repository for all the upcoming PHP files created for the course registration system. A PHP file can be made within Microsoft Visual Studio Code or Notepad++, saved within the file structure, and named 'filename.php'. Then Google Chrome can open 'http://localhost/filename.php' to ensure the appropriate page is loaded (Mikoluk, 2013).

**MySQL Functions and Steps to Create Database**

Before creating the database, I opened PhpMyAdmin, which would be used on the back end to store student information. Utilizing XAMPP, I selected the 'Admin' button next to MySQL to open the phpMyAdmin local server. Once the admin page loaded, I created a new SQL document in Notepad++ named CST499\_Week\_3.sql and Imported it into phpMyAdmin (SiteGround, n.d.). The imported document provided the MySQL code, where a database named **CST499\_ben** is created; this unique name will be utilized later when establishing the connection between the PHP files and the database. The primary table, **tbluser**, is defined with columns such as user ID, email, password, and role information. An Admin user with the highest role is inserted into this table.

Additionally, roles and permissions are managed through the **roles** and **permissions** tables, respectively. A relationship between roles and permissions is established with the **role\_perm** table, while the **user\_role** table defines the relationship between users and roles. A table named **contact\_messages** is created to store messages. Furthermore, a trigger called **after\_tbluser\_insert** is defined to automatically insert a corresponding user-role record when a new user is added. These components contribute to a well-organized database schema that supports user management, access control, and message storage functionalities (SiteGround, n.d.). Additional Tables will be added to the database for Course enrollment information.

**Developing the PHP Files**

This was one of the most challenging portions of the project this week; however, I could fortunately reference much of the course material from CST310 to develop the web page similarly. I required quite a few files to be created to design the webpage and handle the inputs. The first document I made was the config.php file. This dealt with the connection information to the specific 'cst499\_ben' database with the servername, username, password, and database name. Next, the index.php or landing page also established the function' $pdo', which could be used on other pages to connect to the database. This page is the first page loaded and requires config.php and master.php. It also changes from the initial home page to the User Session School Home Page when a user logs in on the login.php page.

The following file I began work on was the master.php file. This was the base file for the homepage and provided the links to the additional pages. This page consisted of a header and a banner with the appropriate links (login, logout, profile, etc.) and pointed to the additional footer.php file to hold the copyright information. The master.php file also enables the user to log into individual profile sessions and will adjust the page as necessary based on the user. For example, the home page will prompt a login, while if a user is logged into their session, it will prompt them to view their profile or log out. The final two portions of the assignment gave me the most difficulty: the registration and the login pages. The registration page file 'registration.php' serves as the leading new user input boxes to place text in and a button to submit the data. The code uses $connString and $pdo to establish the link between the table and those text boxes (Krossing, 2015). Once data is inputted into the text boxes on the webpage and the submit button is pressed, the registration.php file conducts several checks with Javascript and PHP to verify the information is in the correct format, that user information is not already in the database, and tells the verified text where to be placed into the table (Connolly & Hoar 2018). The user is automatically directed to the Login page. The login.php starts a session and connects to the table to read the data within the table and identify if the inputs match, the inputs being a user's email and password. I also created the logout page to end the user's session and return to the landing homepage. Bootstrap was used to setup the pages format and html code (Oh, n.d.). Included is the admin panel to change user roles and other information, which allows access to additional pages based on role name.

**Conclusion**

In conclusion, this week's development efforts focused on creating the Course Enrollment System for CST499. The project drew parallels to previous coursework, particularly in CST310, and marked an exciting venture into web development, an area of keen interest for further exploration. The process involved setting up a local server using XAMPP, creating a database through PHPMyAdmin, and developing PHP files to establish the necessary web pages. The steps to run a PHP file in XAMPP were outlined, emphasizing the importance of the XAMPP control panel and the 'htdocs' folder for file storage and access. MySQL functions played a crucial role in database management, with detailed steps provided for creating the database and tables. Developing PHP files, including config.php, index.php, master.php, registration.php, and login.php, presented challenges, but leveraging past course materials facilitated the process. Noteworthy components include the master.php file, which serves as the foundation for the homepage, and the registration.php and login.php files, which are responsible for user input validation and login functionality. Bootstrap was utilized for page formatting and HTML code structure. Overall, the project laid the groundwork for a comprehensive Course Enrollment System, encompassing user management, access control, and message storage functionalities within a well-structured database schema.

**Screenshots**

Below are screenshots of the database and table, the various pages, and the PHP files.

**Figure 1.** Landing or initial Home Page

A screen shot of a computer

Description automatically generated

**Figure 2**. Login Page.

A screenshot of a computer

Description automatically generated

**Figure 3.** Registration Page.

A screenshot of a computer

Description automatically generated

**Figure 4**. Input Entered in Registration Page.

A screenshot of a computer

Description automatically generated

**Figure 5.** Admin page.

A screenshot of a computer

Description automatically generated

**Figure 6.** Profile page

A screenshot of a computer

Description automatically generated

**Figure 7.** Manage page.

A screen shot of a computer

Description automatically generated

**Figure 8.** Database cst499\_ben

A screenshot of a computer

Description automatically generated

**Figure 9.** Table User with data.

A screenshot of a computer

Description automatically generated

**Figure 10.** Current User Role Table

A screenshot of a computer

Description automatically generated

**Figure 11.** Role permissions.

A screenshot of a computer

Description automatically generated

**Figure 12.** Roles.

A screenshot of a computer

Description automatically generated

**Figure 13.** New User Added Default Applicant Role.

A screenshot of a computer

Description automatically generated

**Figure 14.** Index.php

A computer screen shot of a program code

Description automatically generated

**Figure 15.** Master.php

A computer screen with text on it

Description automatically generated

**Figure 16.** Master.php

A computer screen with text and images

Description automatically generated

**Figure 17.** Master.php

A screen shot of a computer program

Description automatically generated

**Figure 18.** Master.php

A computer screen with many coding

Description automatically generated

**Figure 19.** Master.php

A screen shot of a computer

Description automatically generated

**Figure 20.** Master.php

A computer screen with text on it

Description automatically generated

**Figure 21.** config.php

A screenshot of a computer

Description automatically generated

**Figure 22.** footer.php

A screenshot of a computer

Description automatically generated

**Figure 23.** Login.php

A computer screen with text on it

Description automatically generated

**Figure 24.** Registration.php

A computer screen with many lines of text

Description automatically generated

**Figure 25.** CST499.js

A screenshot of a computer

Description automatically generated

**Figure 26.** CST499\_Week\_3.sql

A computer screen shot of a computer code

Description automatically generated

**Figure 27.** CST499\_Week\_3.sql

A computer screen with text on it

Description automatically generated

References

Connolly, R., & Hoar, R. (2018). [*Fundamentals of web development*](https://uagc.instructure.com/courses/124945/modules/items/6353220)(2nd ed.). Pearson.

Javatpoint. (2021). *Xampp tutorial*. www.javatpoint.com. Retrieved 2023, from https://www.javatpoint.com/xampp

Krossing, D. (2015, November 15). *31: Mysql Insert into database | php tutorial | learn php programming | php for Beginners*. YouTube. Retrieved 2023, from https://www.youtube.com/watch?v=IQy8SenpdfM

Mikoluk, K. (2013, September 18). [XAMPP tutorial: How to use XAMPP to run your own web server](XAMPP%20tutorial:%20How%20to%20use%20XAMPP%20to%20run%20your%20own%20web%20server). *Udemy.*https://blog.udemy.com/xampp-tutorial

Oh, S. (n.d.). [*Bootstrap: Tutorial 7*](http://www.cs.toronto.edu/~mashiyat/csc309/Tutorial/6/Bootstrap.pdf). https://www.cs.toronto.edu/~mashiyat/csc309/Tutorial/6/Bootstrap.pdf

SiteGround. (n.d.). [*phpMyAdmin create and populate tables tutorial*.](phpMyAdmin%20create%20and%20populate%20tables%20tutorial.) https://www.siteground.com/tutorials/phpmyadmin/create-populate-tables/

Tsui, F., Karam, O., & Bernal, B. (2018). [*Essentials of software engineering*](https://uagc.instructure.com/courses/124945/modules/items/6353220)(4th ed.). Jones & Bartlett Learning.