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

Jonah Hannett

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology (CSF2535A)

Joseph Rangitsch

9/15/2025

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

Building the student registration page is a great first step as it is the student’s first step as well. They need to first register their profile before they can schedule their classes, and most of the pages will be built during this step as well. I will still need to add the pages that let the students schedule their classes and view their schedule, but at this time, they will be able to see a landing page, the login page, the registration page, and some semblance of a custom user profile. In this paper we will go over the steps to build a local database and website, as well as how to send data between them.

**Running PHP on XAMPP**

Running PHP files in XAMPP is a simple process once you have installed the XAMPP software and Apache/MySQL. After installation you need to find your XAMPP file which will probably be in your C: drive. In there you will want to look for htdocs so the full path will be c:\xampp\htdocs. In this folder you just need to create a new folder and title it whatever you want, possibly “project”. To run a PHP file, you just need to add a text file and give it a .php extension (e.g. test.php). To run the file, open XAMPP, start Apache, go to a browser and type in localhost/project/test.php and if valid PHP/HTML is in the file it will run.

**Setting up the Database and Connection Class**

Because MySQL was set up when installing XAMPP, it needed to be started along with Apache to create a database connection between the site and the available tables. To set up the MySQL connection, I needed to open the config ‘my.ini’ file and change a few bits, specifically the port from the default 3306 to 3307 as it wasn’t connecting with the 3306 port. Once this was set, I could access MySQL from XAMPP by clicking the admin button. To access the database through the browser, I needed to make a separate file in my StudentScheduler project folder titled config.php, which I use as a reference whenever I need to connect to the database in my code. The file contains PHP code that defines the connection using DBHOST, DBNAME, DBUSER, and DBPASS followed by a $con function that uses mysqli\_connect passing the above information and creating a connection.

**Saving User Registration Information to Database**

Creating a registration page required both HTML and PHP code to create an interface that someone could interact with. I started by adding the common html header info and included a text color specifically for errors. Once I had the html I used PHP code to first include my config.php file so I could call the database. I wanted to require every field so I started with including null errors which would be filled in and displayed next to the forms if something was empty. Next I include null values which will be filled in if valid before being sent to the database. I then use forms in html to ask for inputs, and the PHP reads the inputs and validates they are clean and there are no duplicate emails since I am currently using emails as the username, and they need to be unique. If there are no errors with any of the forms, I throw a message saying the student was registered and then use a SQL command to insert the information into the student\_class\_scheduler database under the tblStudent table. I let the ID auto increment since it will be one of the most important values to determine which student took which action. Once the values are sent to the database I send the user to the login page so they can input the new user information and log in. The database needed to be available first so I could send the information to it, so before I tested the registration page or even built it, I opened MySQL through XAMPP, and it shows phpMyAdmin in a browser. I clicked on new database and named it student\_class\_scheduler, then made my tblStudent table to store the information. I used INT for my student ID field, but the rest of the columns used VARCHAR to allow for a variety of characters which I could sanitize in my code. Student IDs automatically increment when loaded into the database so I don’t need to send a value or apply a count somehow in my code which could cause issues inadvertently.

**Code and Database Screenshots**

A computer screen shot of a program

AI-generated content may be incorrect.

Figure : config.php. This is the file used to connect to the database.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Landing page. The index.php file holds the code for this webpage.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : index.php. This file holds the code for the landing page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Registration page. The registration.php file holds the code for this page and is used to send new student info to the database.

A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

Figure : registration.php. The code for the registration page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Login page. Webpage used to create the login session and redirects to user profile.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

Figure : login.php. The code for the login page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Welcome page. This is the first page a student sees when they log in and is created with the welcome.php file.

A computer screen shot of a program code

AI-generated content may be incorrect.

Figure : welcome.php. The code for the welcome page for students.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Student profile page. Handled by the profile.php file and shows the information entered in the registration page.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure : profile.php. The code that runs the student profile page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Logout page. This page shows up when a student logs out of the system and is handled with the logout.php file.

A computer screen shot of a program code

AI-generated content may be incorrect.

Figure : logout.php. This is the code that handles the user logout process and disconnects the session.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Login request page. This is used as an intermediate page that appears if you try to go to the profile.php or welcome.php page without generating a valid session and is handled by the pleaselogin.php file.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure : pleaselogin.php. This is the code that displays the page before sending you back to the landing page.

A screenshot of a computer

AI-generated content may be incorrect.

Figure : tblStudent. This is the beginning of the database which holds the user registration information.

**Conclusion**

In this paper we have covered what it takes to set up XAMPP along with Apache and MySQL, which allows you to build a local website that communicates with a database. In my code I set up a configuration file used to open the connection with the database which can be called in all other files to facilitate all database communication whether sending or reading data. The code also includes files that cover registration, login, a landing page, and a student profile page. There are still many improvements I can make such as proper password hashing and additional SQL injection protection, but this is a great start, and I look forward to completing the project.