Landing, Login, and Enrollment Pages Development

Dallas Spencer

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Instructor: Charmelia Butler

January 13th, 2025

The primary project for this course is developing a student portal where we focus on the core function of the course adder. To do this, we have to leverage several technologies to get the software off the ground. In this essay we will analyze the tools used and evaluate the source code for the first legs of the project.

In order to test and access this project locally, we have to utilize an open-source software called XAMPP. This is short for Cross-platform, Apache, MySQL, PHP, and Perl (javatpoint, n.d). This program allows anyone to set up a web server from their local device and imitate the experience of your webpage, database, etc.

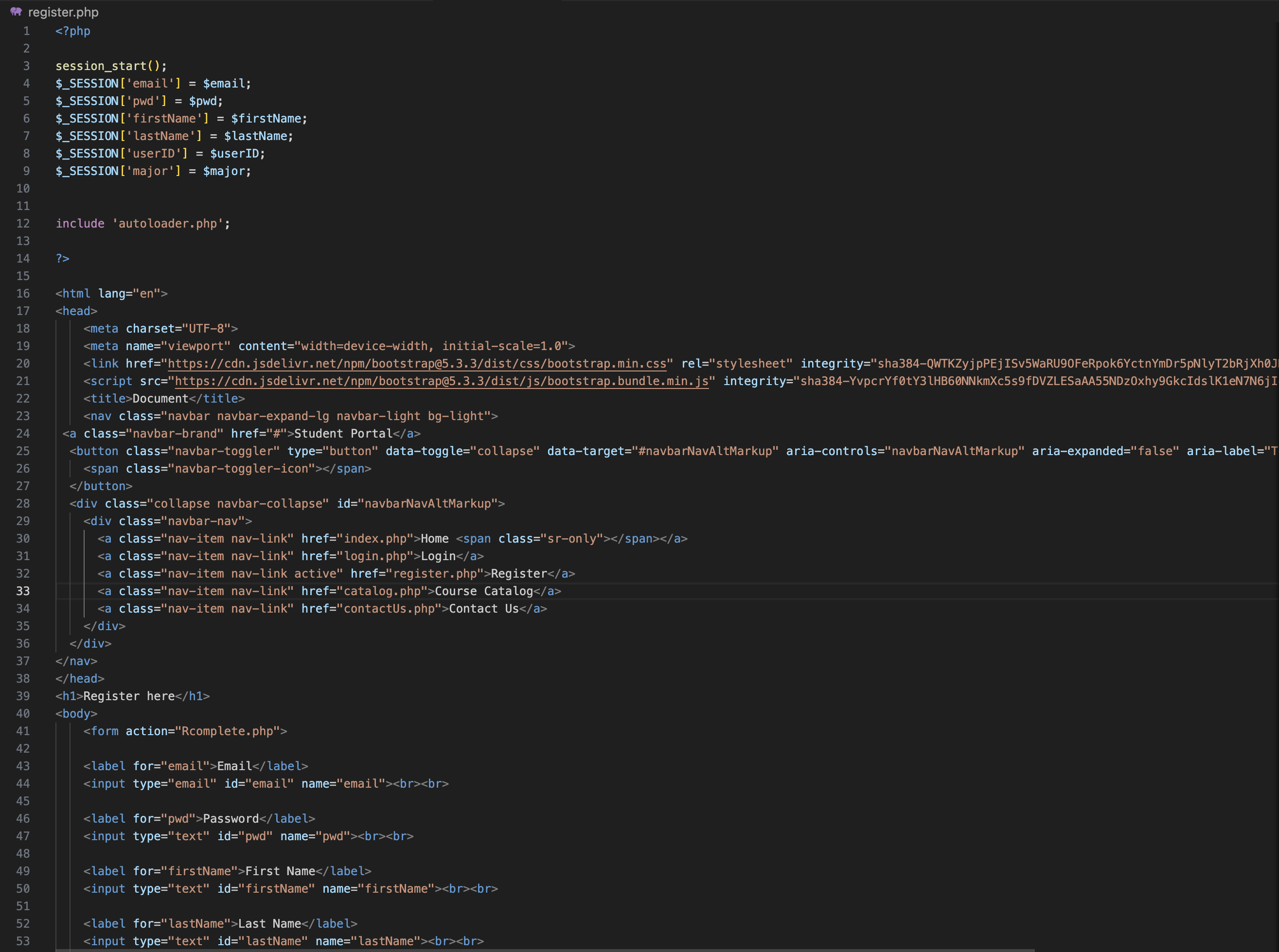
To run a PHP file in XAMPP, you have to create a folder on the computer, under “htdocs” folder within the XAMPP file tree. The folder can be named anything that you need it to be. From there, you will place your documents within the folder and save it in your selected integrated development environment (IDE). Once you have the documents built, you’ll open your browser of choice while the XAMPP server is running locally and enter in the following web address: “127.0.0.1/{foldername}”. This allows the browser to point to the local host and access the proper folder where the project will exist. You can go further and type in the specific file names if you’re testing specific functions (such as rejected due to no authentication).

One of the core functions to XAMPP is allowing for a database with MySQL. MySQL database is managed through phpmyadmin. To allow for a PHP document to communicate with the database, you have run a function called “mysqli\_connect”. This function accepts the variables assigned to connect to and access the database to connect, login, and authenticate (W3schools, n.d). Once connected to the database, mysqli commands allow for querying the database and manipulating any data as needed.

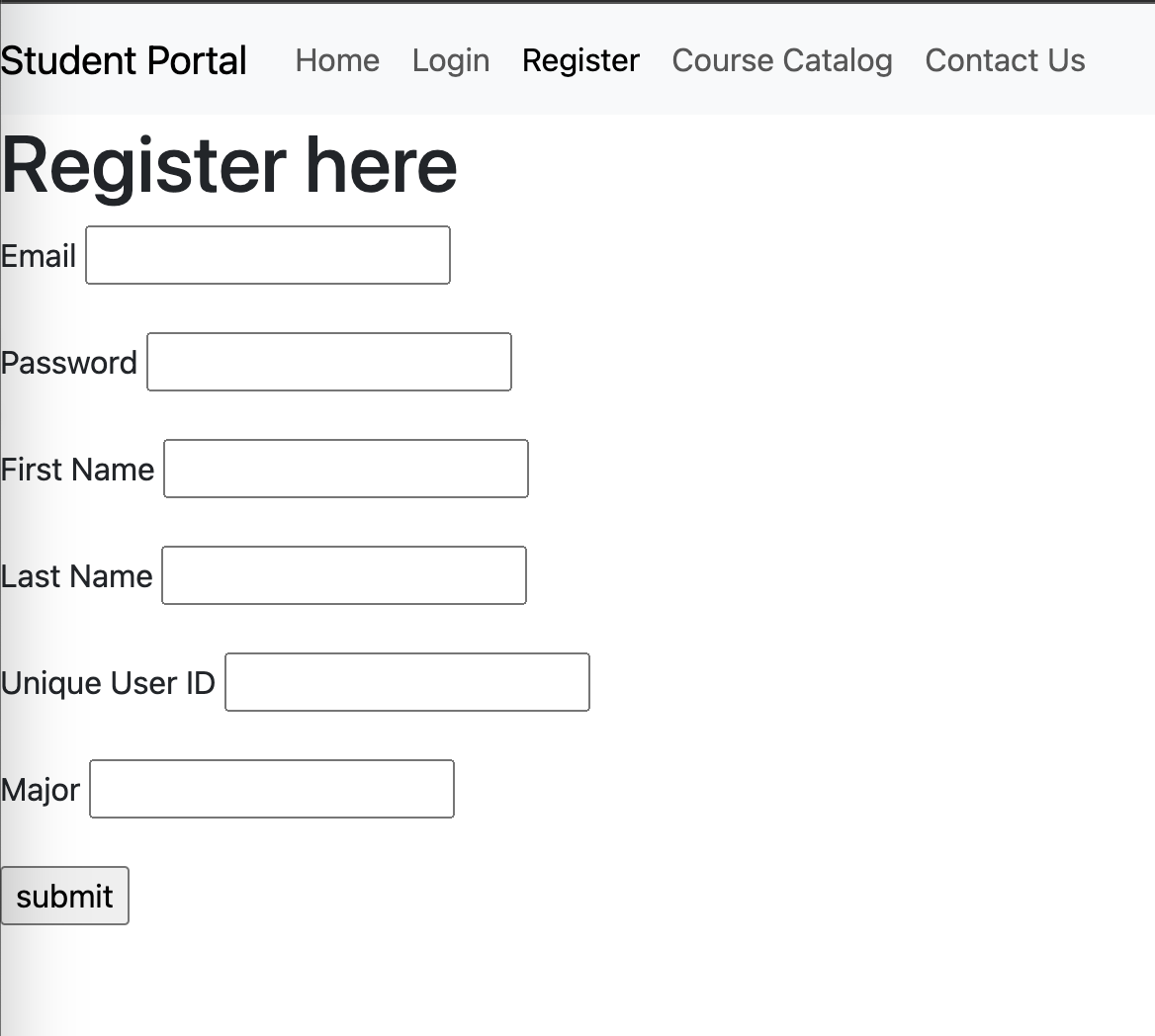
Now that we have established a connection to the database, the registration page leverages this connection to run SQL commands against the table, tblStudent. Leveraging bootstrap, the page has a CSS framework with HTML input boxes to capture the data. Each of the data inputs are held in variables defined at the start of the page in a session ID and passed to a separate page to register the information. This page securely authenticates the database and creates an SQL command with the input data. One of the requirements for the project is to have unique user IDs, to accomplish this we set the primary key as the userID column and force the ID to be unique. Error redundancy is in the registration form so any errors MySQL throws, the form will capture and dynamically handle.

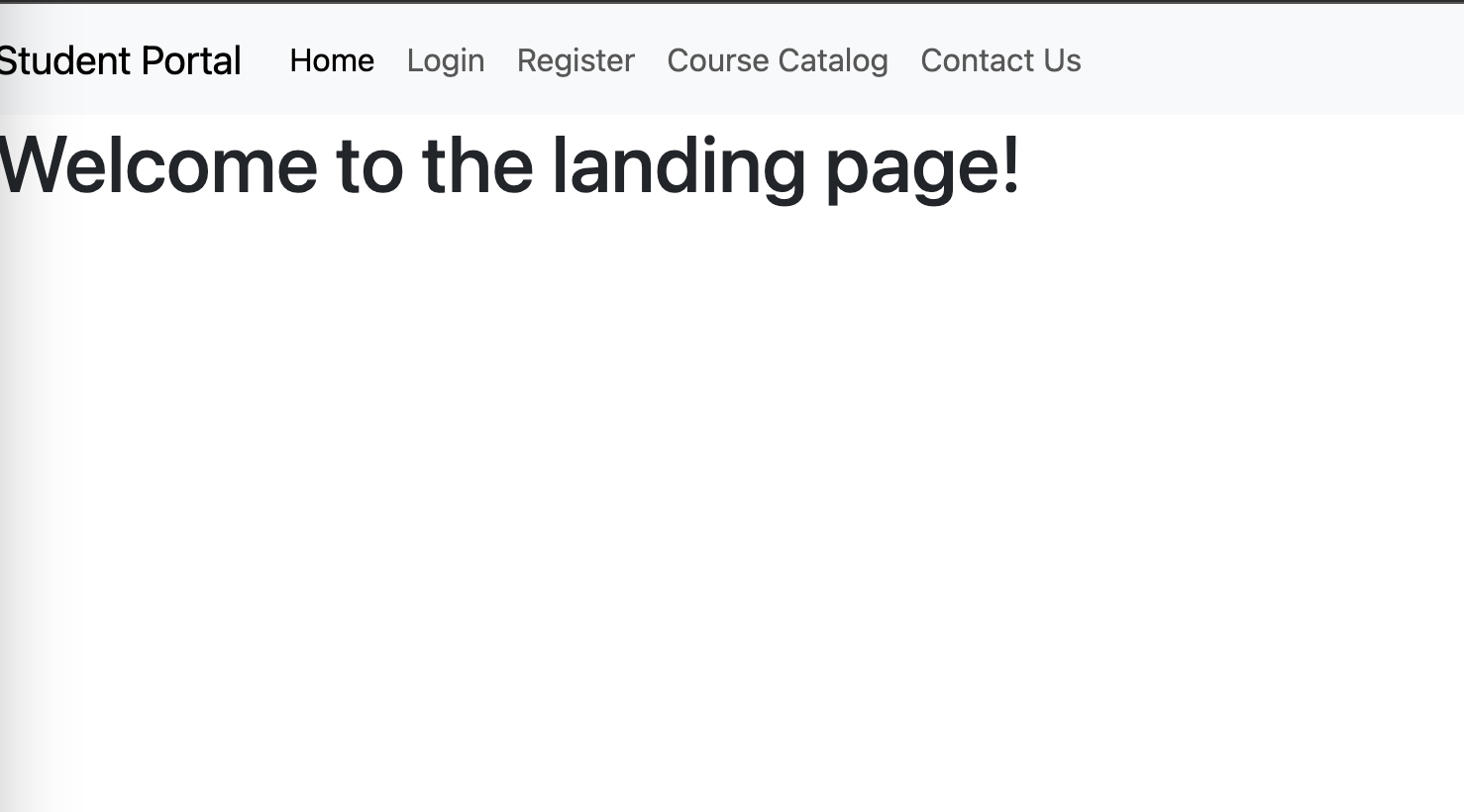
In conclusion, we leveraged multiple functions and interconnected a database in the software to achieve a functional registration system. This system will easily allow for the future expandability of the project.

Source Code:

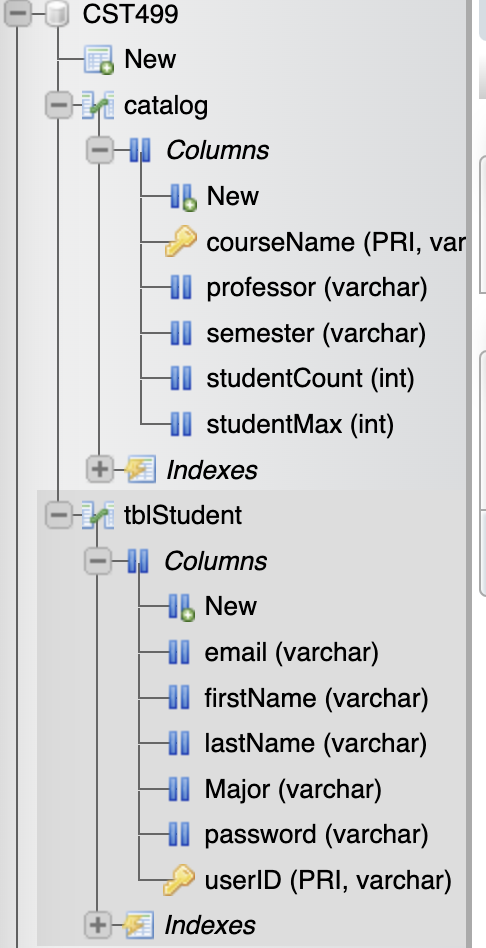


  
*Figure 1 and 2: registration.php source code*

*Figure 3: Registration page HTML+CSS*

*Figure 4: Landing page*

*Figure 5: Rcomplete.php, back end function that registration.php passes the information to create the data into the table.*



*Figure 6: phpmyadmin database with tables “tblStudent” and “catalog”*

Resources:

*XAMPP Tutorial - javatpoint*. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/xampp>

*W3Schools.com*. (n.d.). <https://www.w3schools.com/php/php_ref_mysqli.asp>