**Week 3 Assignment: Landing, Login, and Enrollment Pages Development**

Shaun Hoadley

CST499: Capstone for Computer Software Technology

Professor Amr Elchouemi

January 31, 2022

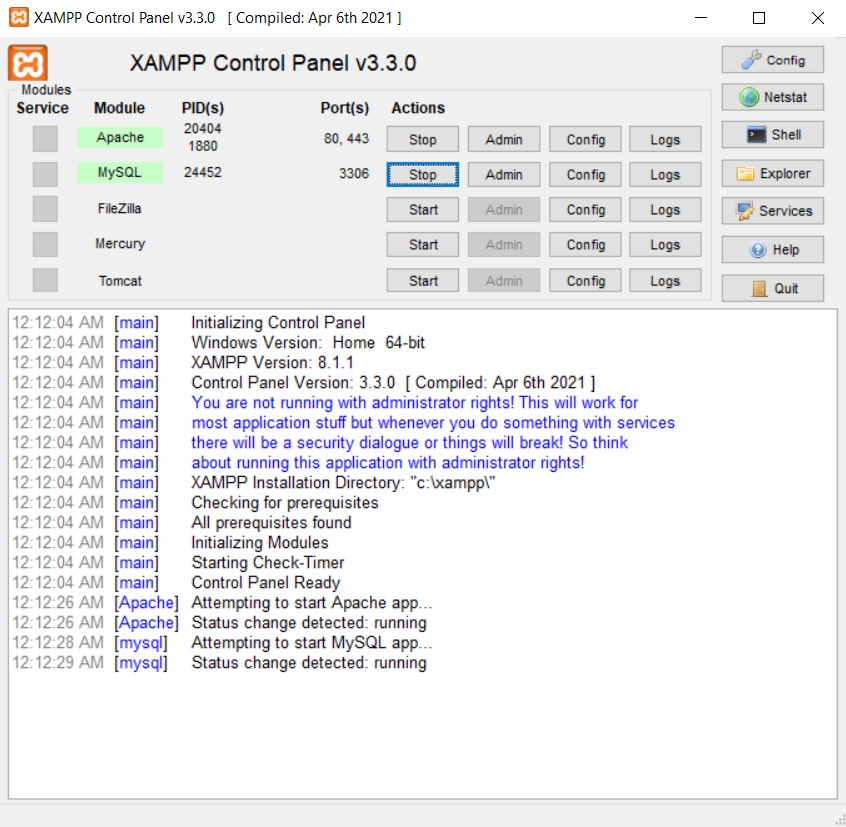
**Landing, Login, and Enrollment Pages Development**

**Introduction**

The purpose of this paper is to begin the implementation of the student course enrollment system that is the final project for the class. Before beginning, we must first gater the tools we need to work with. For the MySQL database and web hosting, we will be using XAMPP because it contains an Apache web server and MySQL application. Additionally, we need an application to do the coding work, for which there are many routes we can take. We can use an integrated development environment (IDE) like Eclipse for PHP or a simple text editor. I have elected to use the Sublime text editor, which has additional functionality over something like NotePad, for formatting programming code and checking the syntax.

**Explain how to run a PHP file in XAMPP**

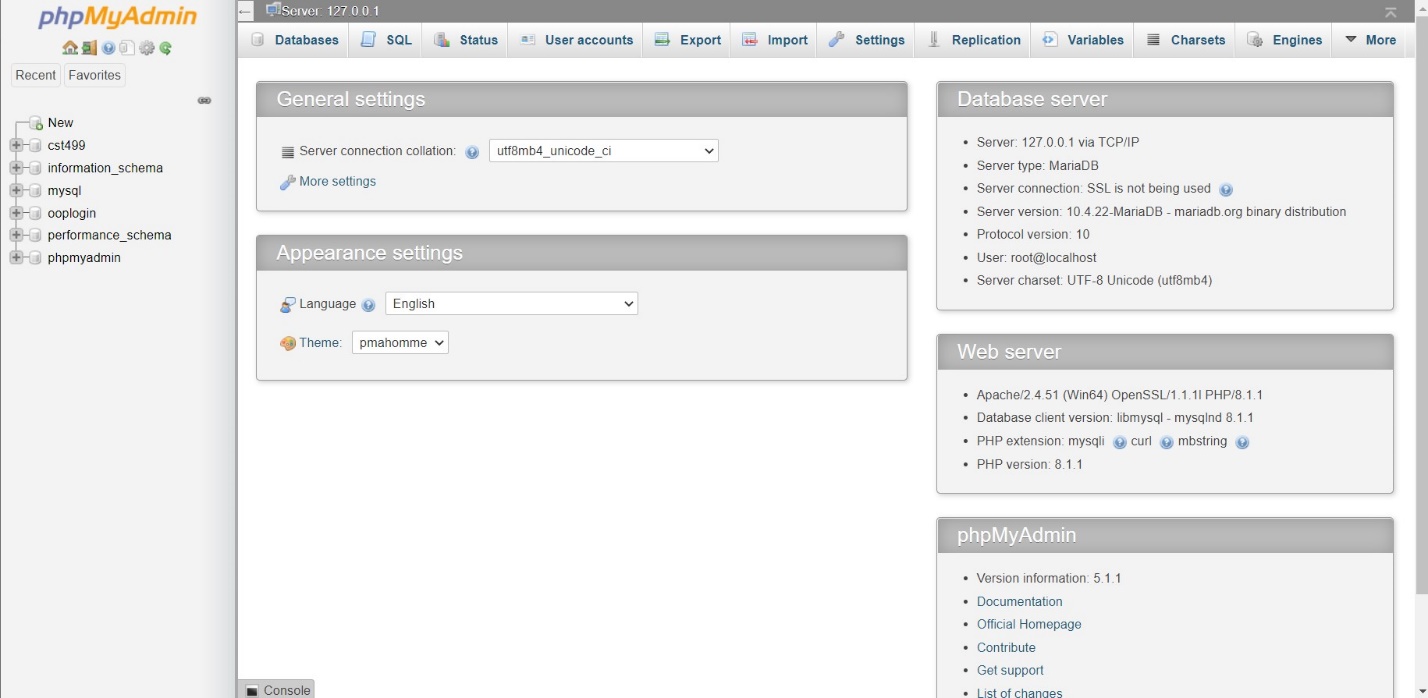
After installing XAMPP, we open the XAMPP Control Panel to start the Apache and MySQL services as seen in Figure 1 below. When the both services have started up, we can test to see if they are working by opening a brower (in my case, Chrome) and typing “localhost” or “127.0.0.1” in the browser navigation bar to get the XAMPP Welcome page (Figure 2 below) and “localhost/phpMyAdmin” or “127.0.0.1/phpMyAdmin” to open the MySQL database application (Mikoluk, 2013). To run any PHP/HTML files we create we will need to save them to the xampp/htdocs folder (default installation is C:\xampp\htdocs). With all of that done, it is time to begin creating our project.



**Figure 1: XAMPP Control Panel**

****

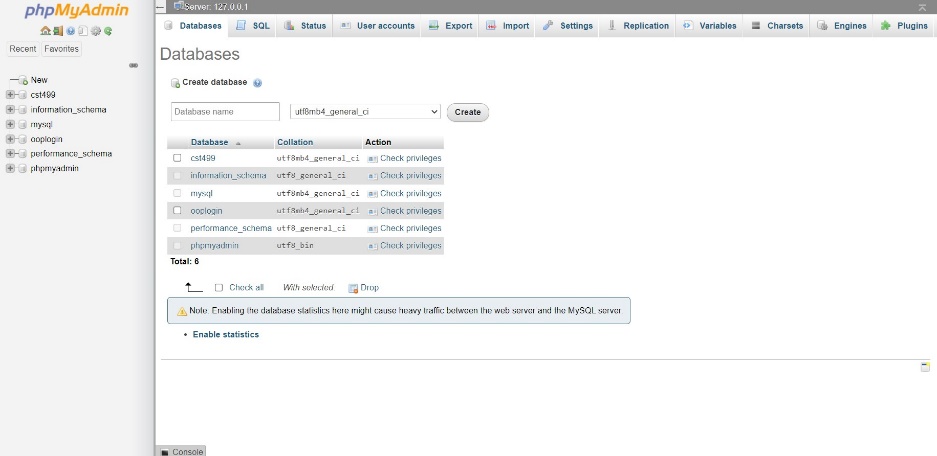
**Figure 2: XAMPP Welcome Screen**

****

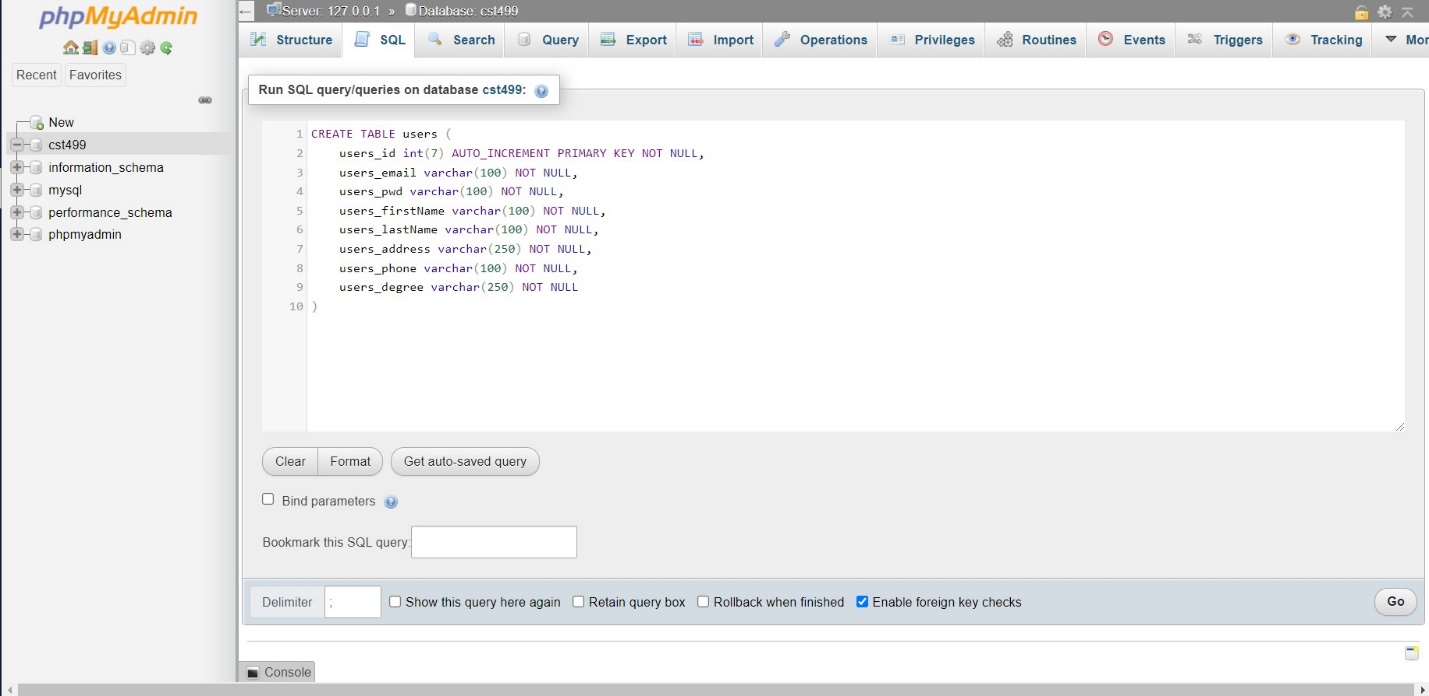
**Figure 3: phpMyAdmin Screen**

**Creating the Database and Users Table**

To create the database for the project, click “New” in the left side panel and enter the name for the database (mine is called cst499 as seen in Figure 4) and select “create”. Next, select the newly created database in the left side panel to show the contents of it (currently empty). We now need to create the users table. Creating the table can be done by filling out the “create table” form, but I find it easier to use SQL code as shown in Figure 5.

****

**Figure 4: Create the Database**

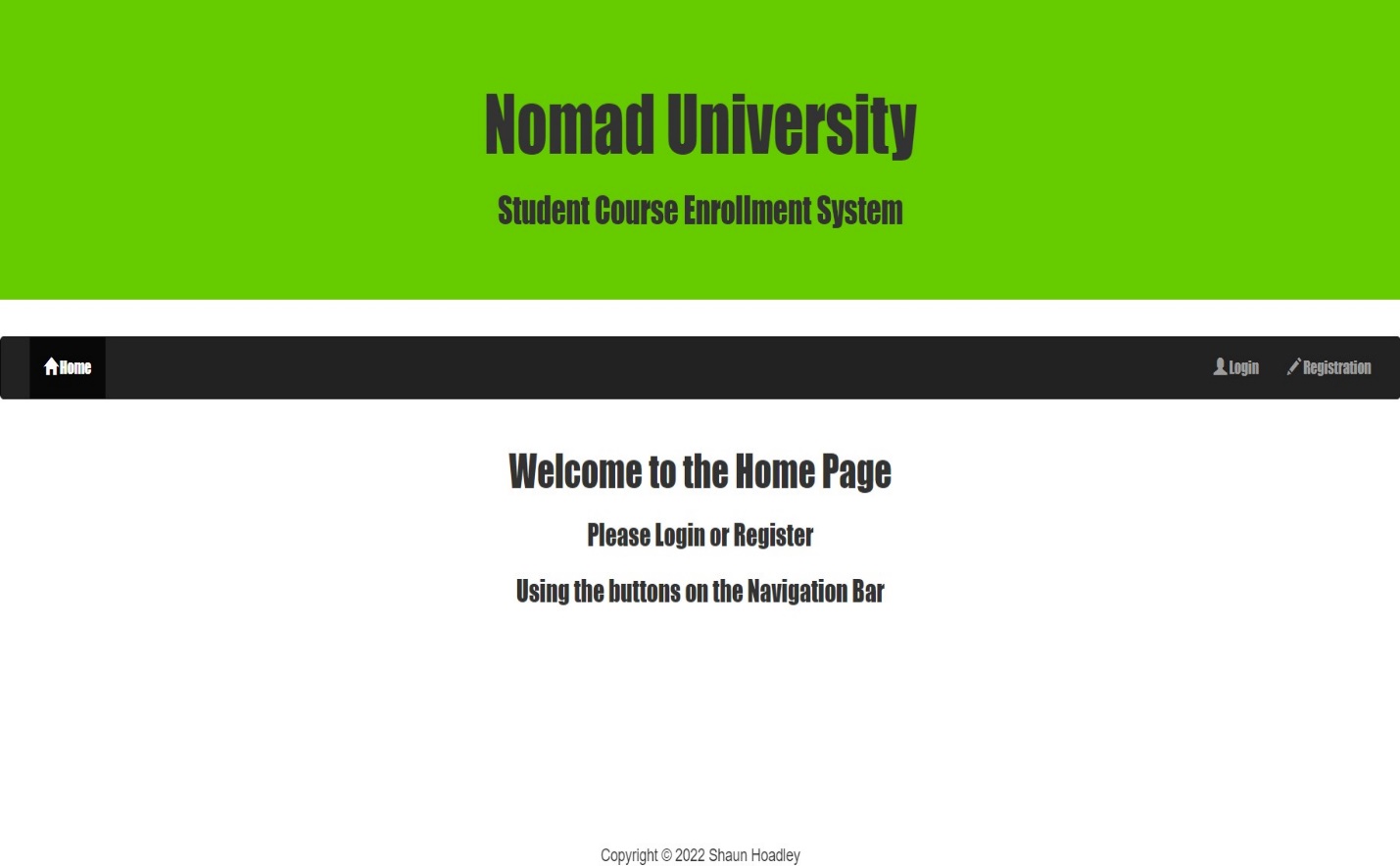
****

**Figure 5: Create the Users Table**

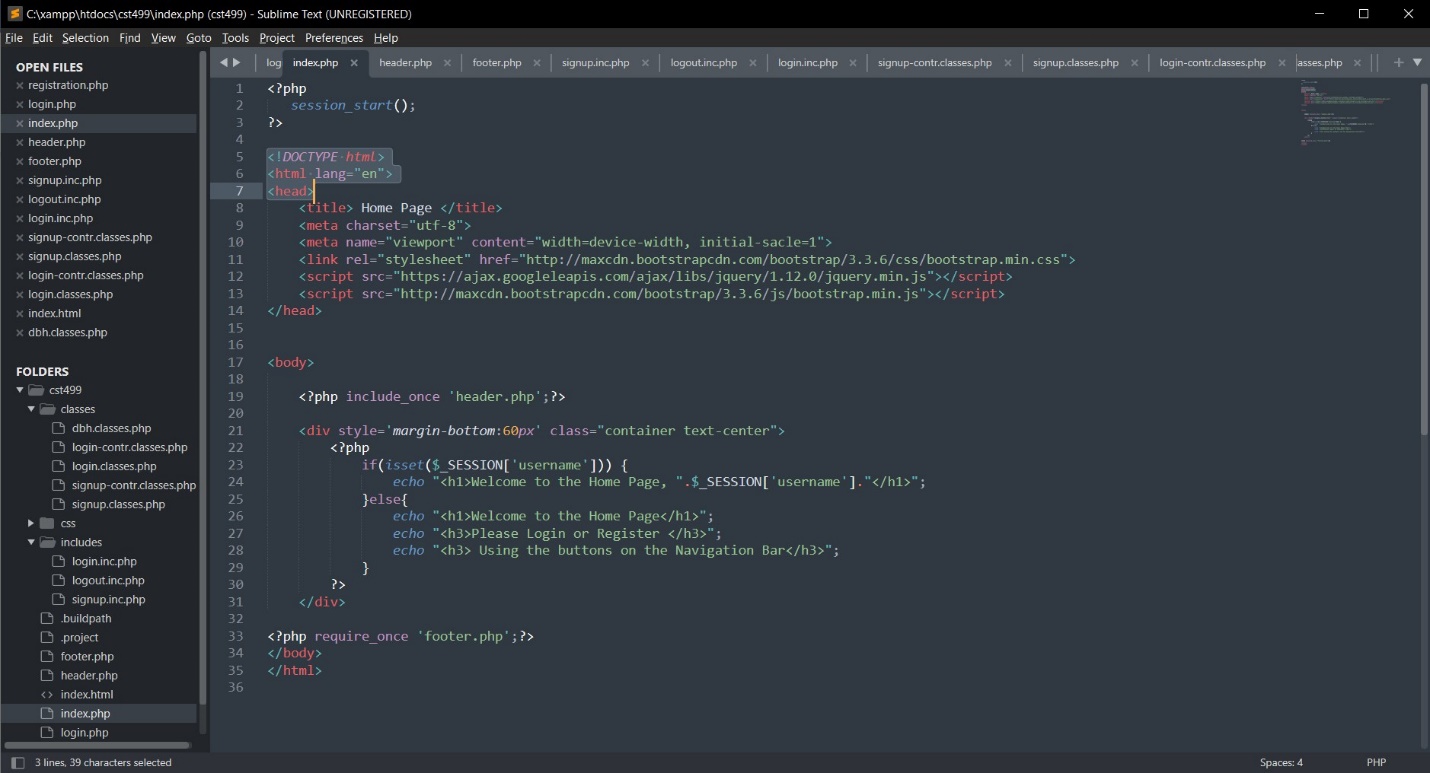
As a note, I could have created the database using SQL code as well, but it really is easier the way I did it.

**Creating the Landing, Login, and Registration Pages**

Now our initial database setup is complete and we can begin creating the project pages that will interact with it. Most students use procedural PHP (which I did in CST310), so I decided to tackle this project using object-oriented PHP instead. We will begin with the landing page. As long as the user has the correct address for the landing page, you can call it whatever you want, though it is most commonly called index.html, index.php, home.html, or home.php. This next part is not strictly necessary, but I created both an index.html and index.php page, with the index.html page doing nothing but redirecting the user to the index.php page (Figure 6). The layout and source code for index.php are Figures 7 and 8 respectively.

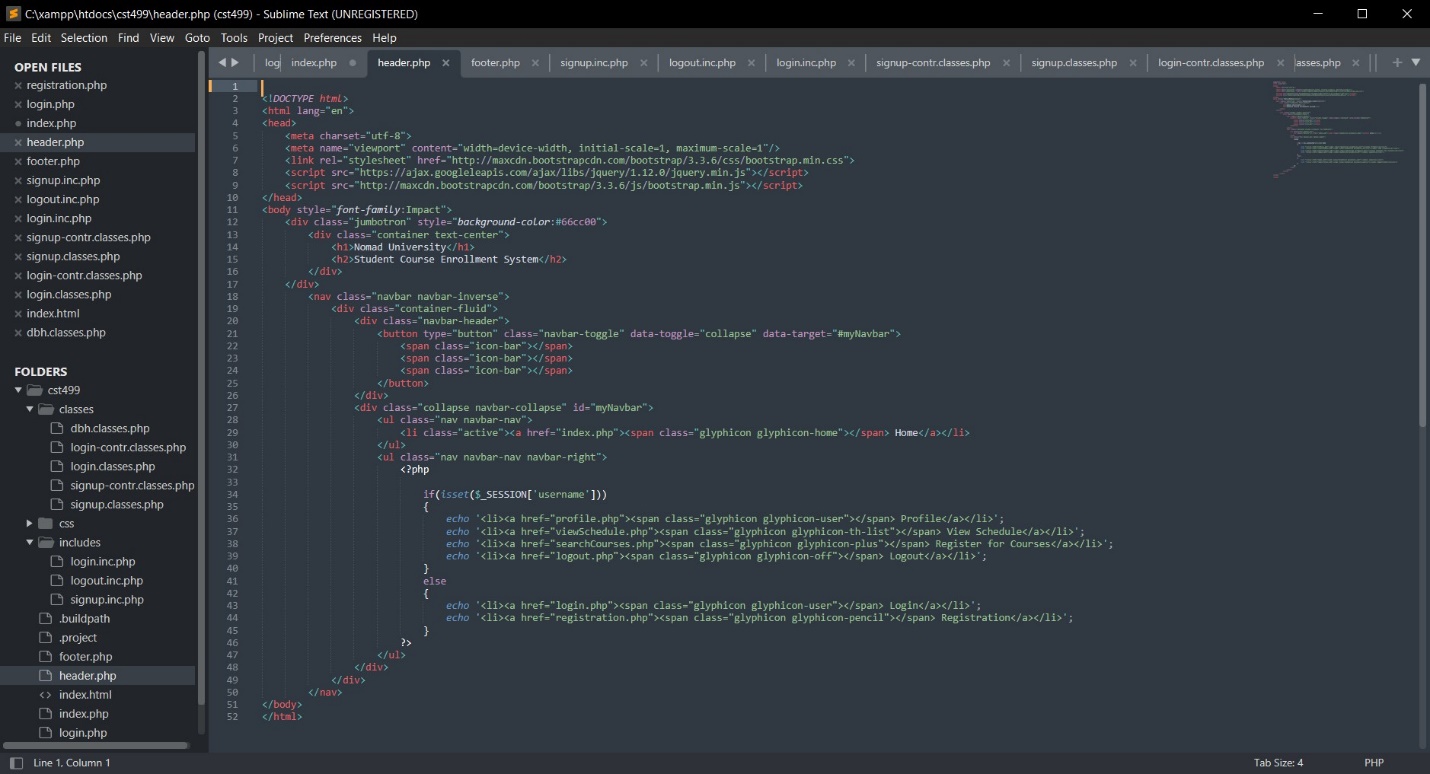


**Figure 7: Landing Page Layout**

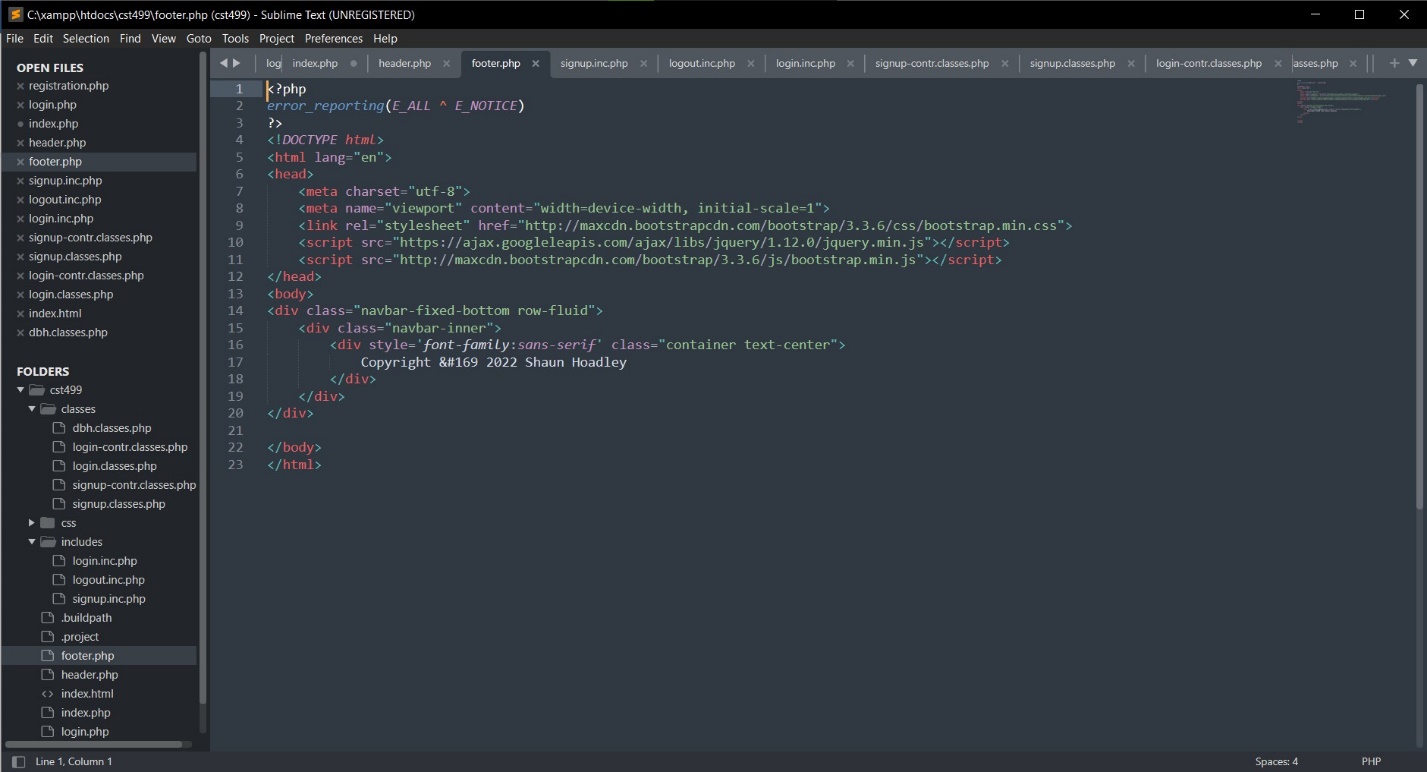
****

**Figure 8: Landing Page Source Code**

There are elements that we want to stay the same regardless of which page we are on, such as the navigation bar and the copywrite. Therefore, I have separate header and footer pages that are included in any page that will be displayed to the users (Figure 9 & 10).

****

**Figure 9: Header Source Code**

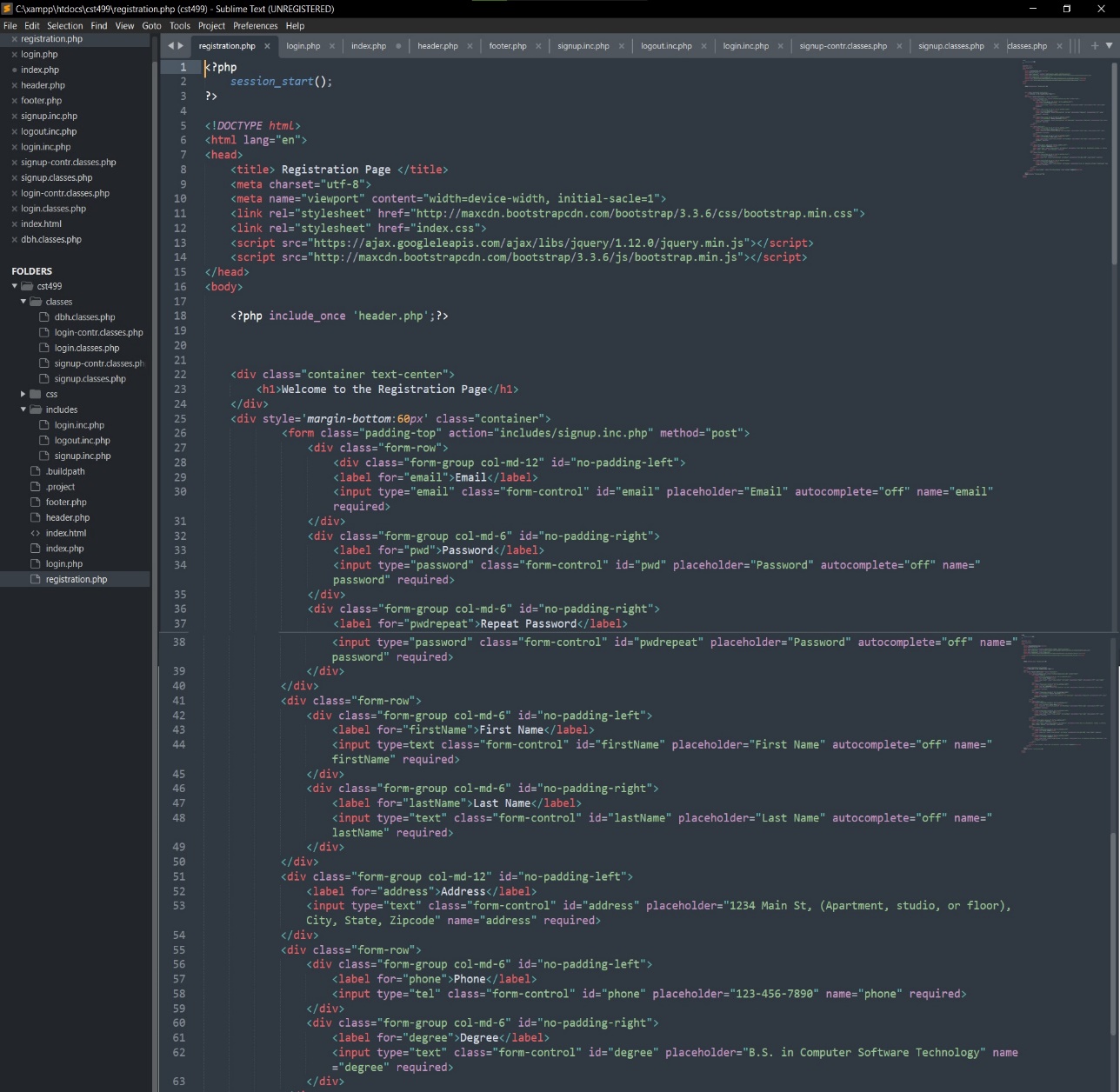
****

**Figure 10: Footer Source Code**

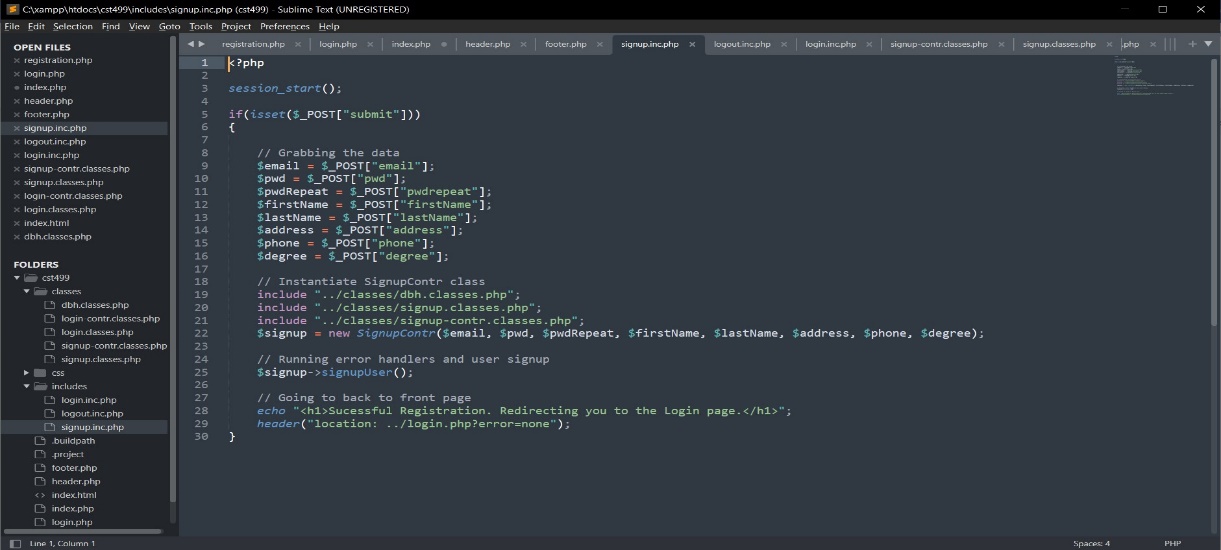
After completing the landing page, I moved to the registration page and supporting files. Figure 11 shows the layout for the registration page and Figure 12 shows the source code for it. Figures 13, 14, 15, and 16 are the source code for the signup includes file, signup controller file, signup classes file, and the database handler classes file. Graphical user interface, text

Description automatically generated

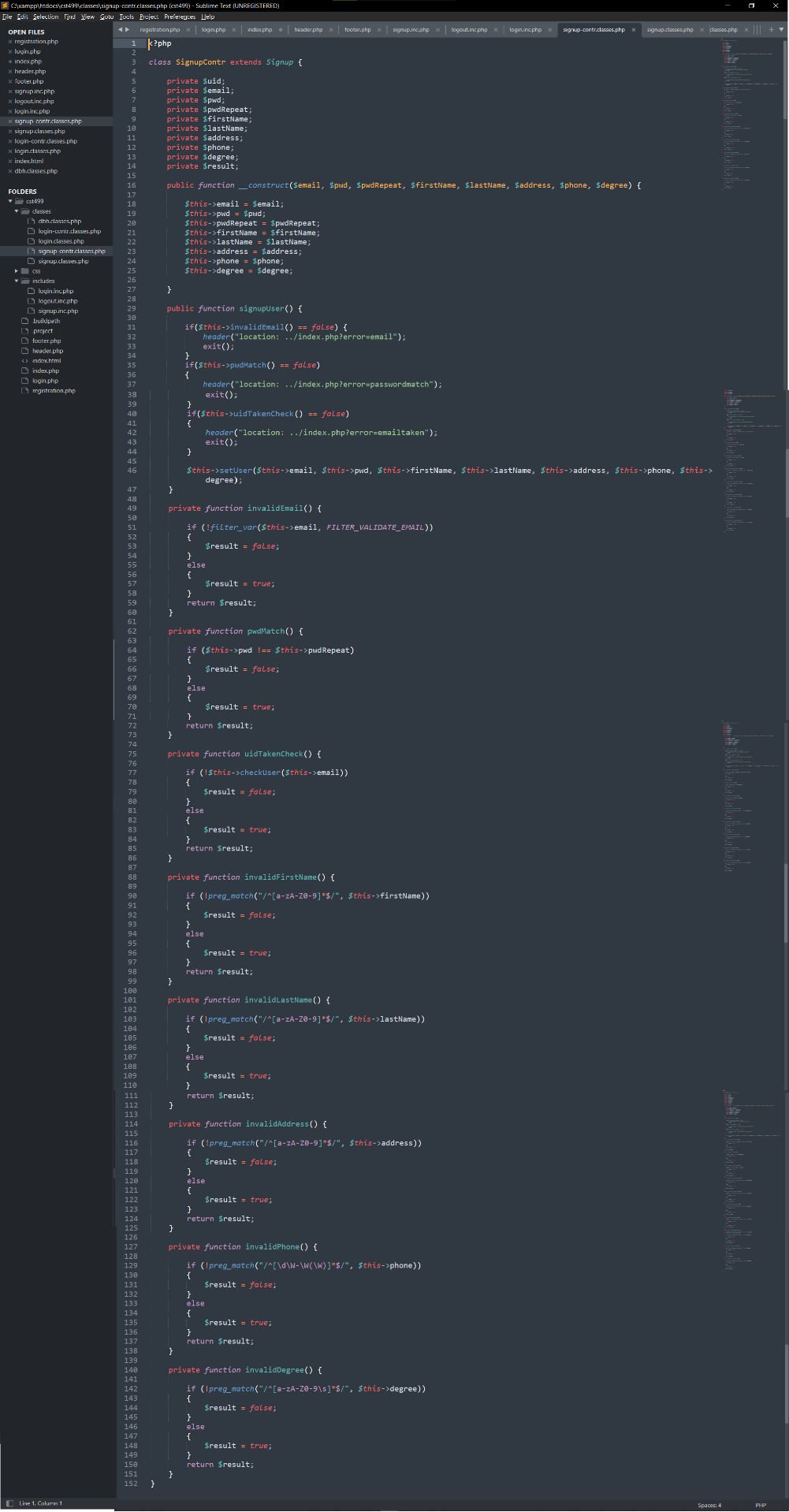
**Figure 11: Registration Page Layout**

****

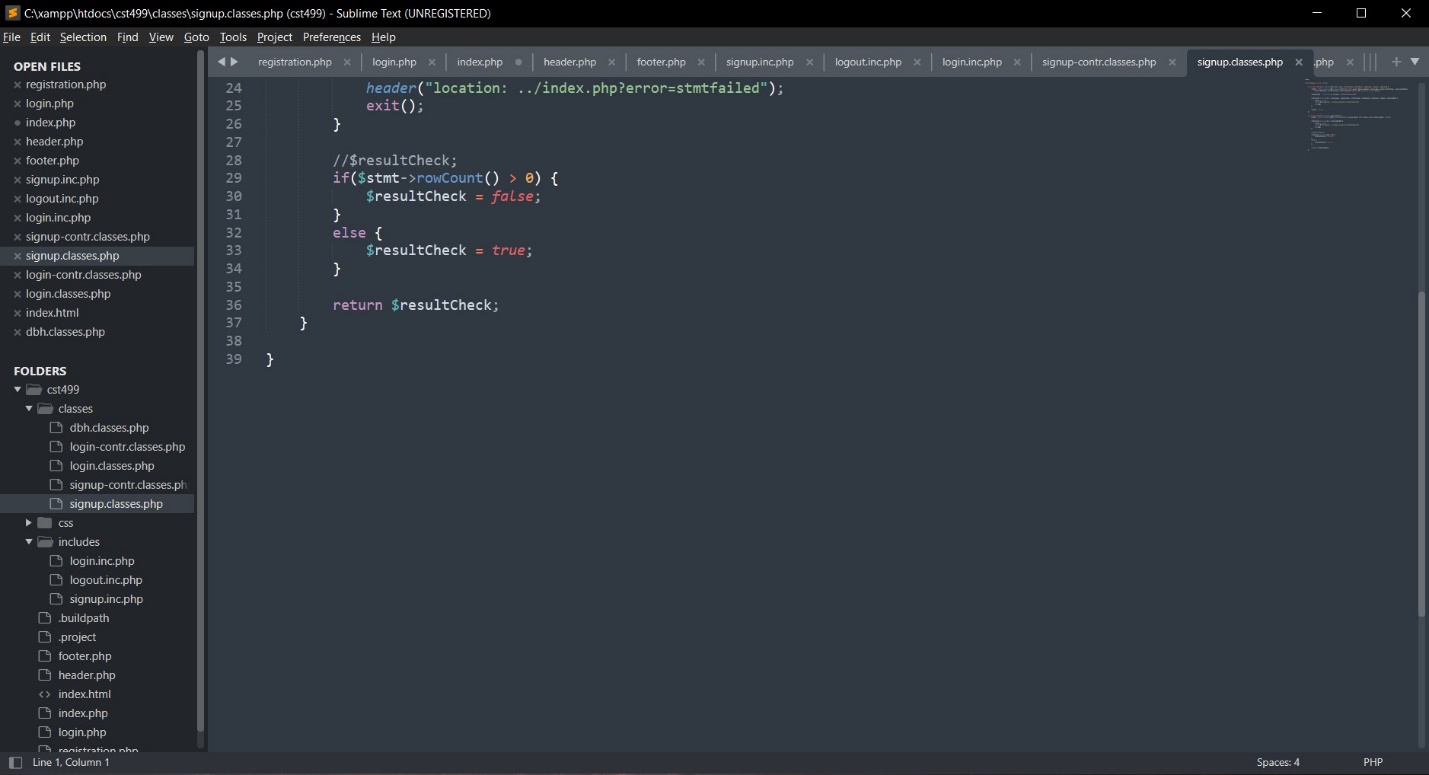
**Figure 12: Registration Page Source Code**

****

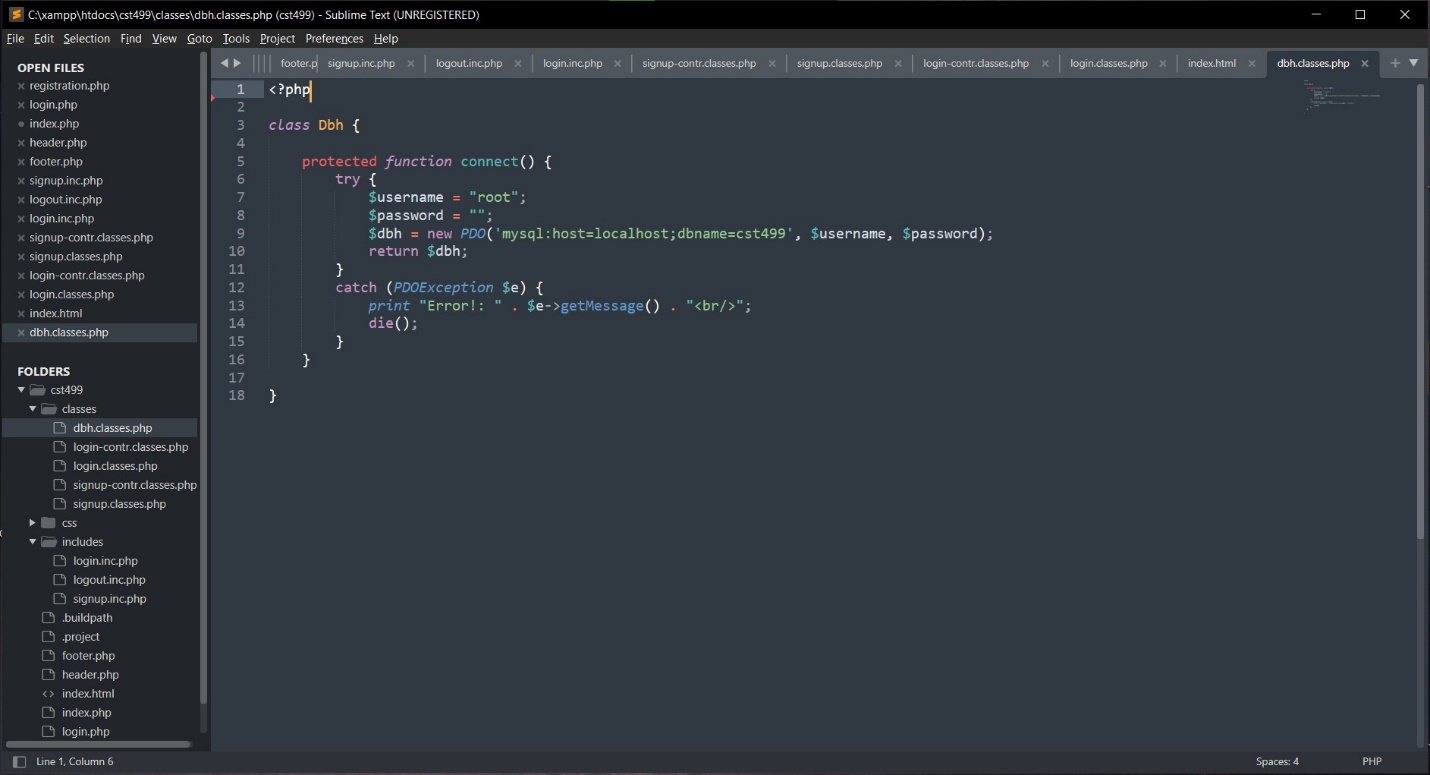
**Figure 13: Signup Includes Source Code**

****

**Figure 14: Signup Controller Source Code**

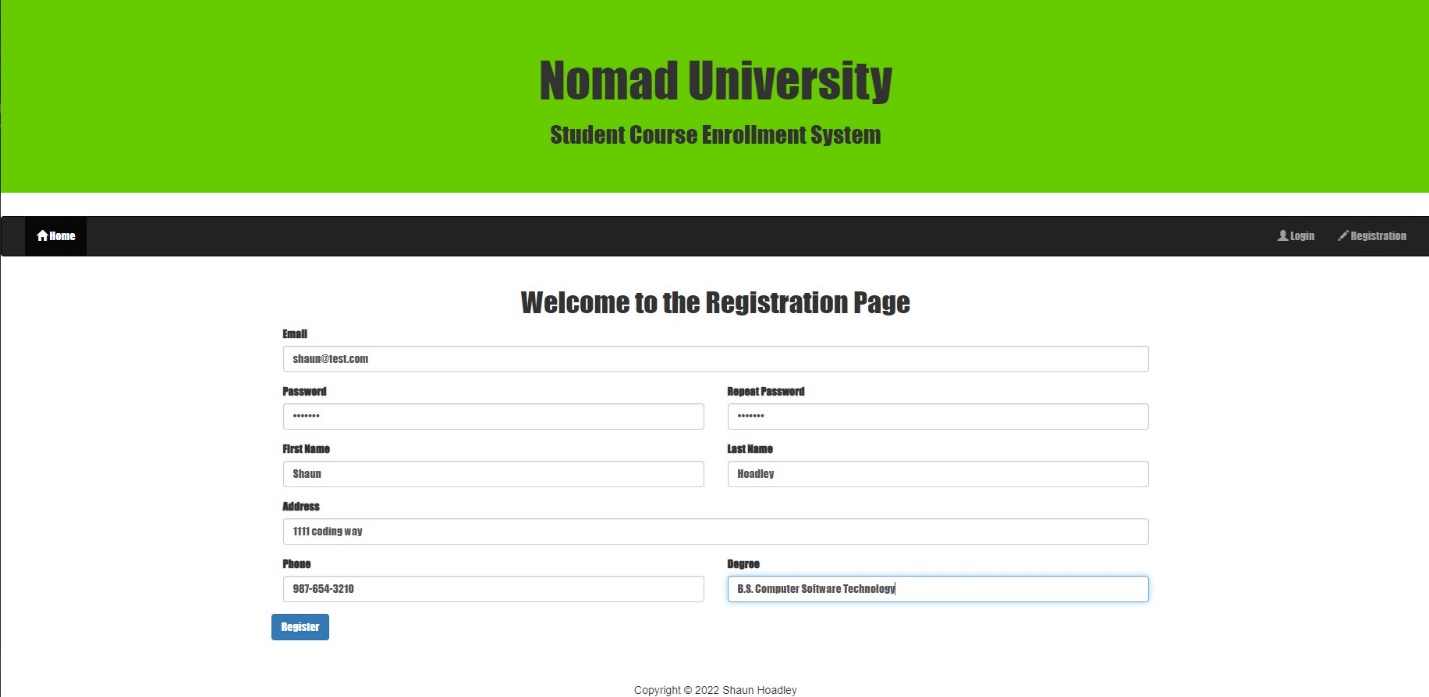
****

**Figure 15: Signup Classes Source Code**

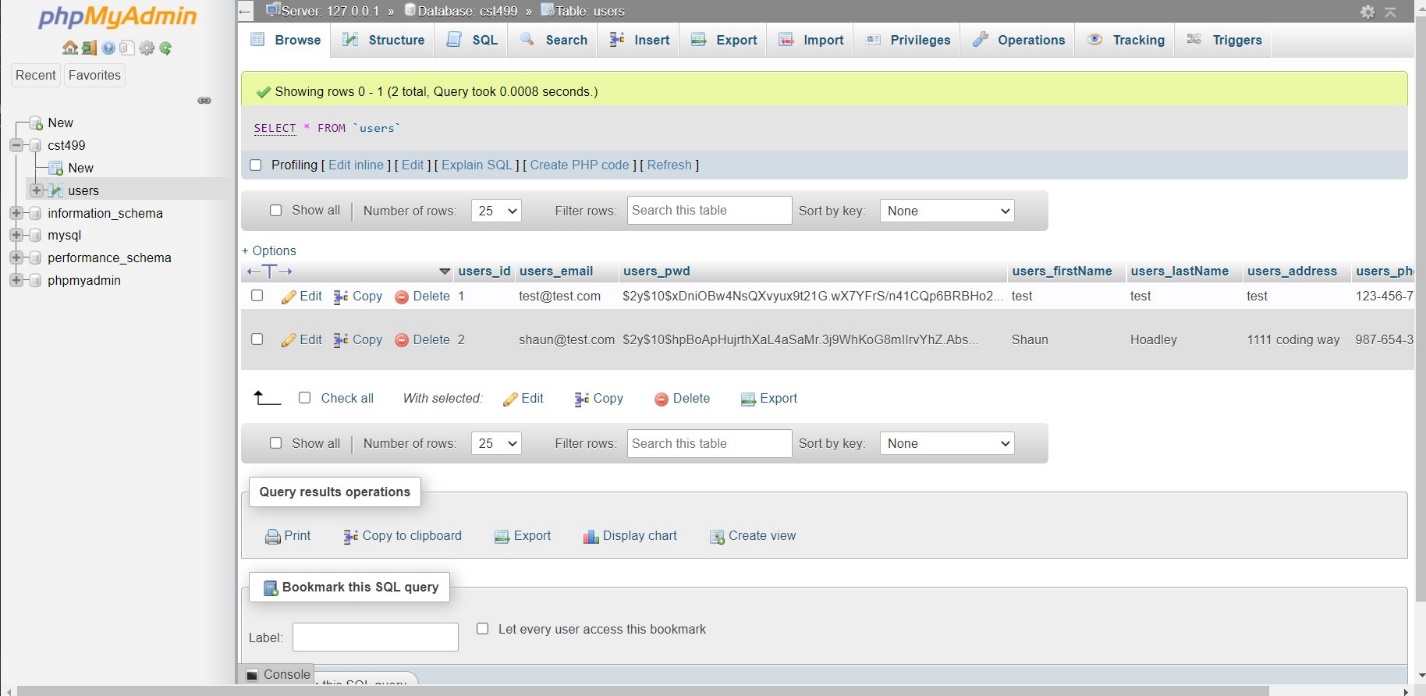
****

**Figure 16: Database Handler Classes Source Code**

After the user fills out the form data on the registration page and selects the submit button, their data is transferred and saved to the database. Upon successful completion, the user is then redirected to the Login Page to sign into the system. Figure 17 shows the registration form filled out prior to selecting submit and Figure 18 shows the database table with the data successfully entered into it.

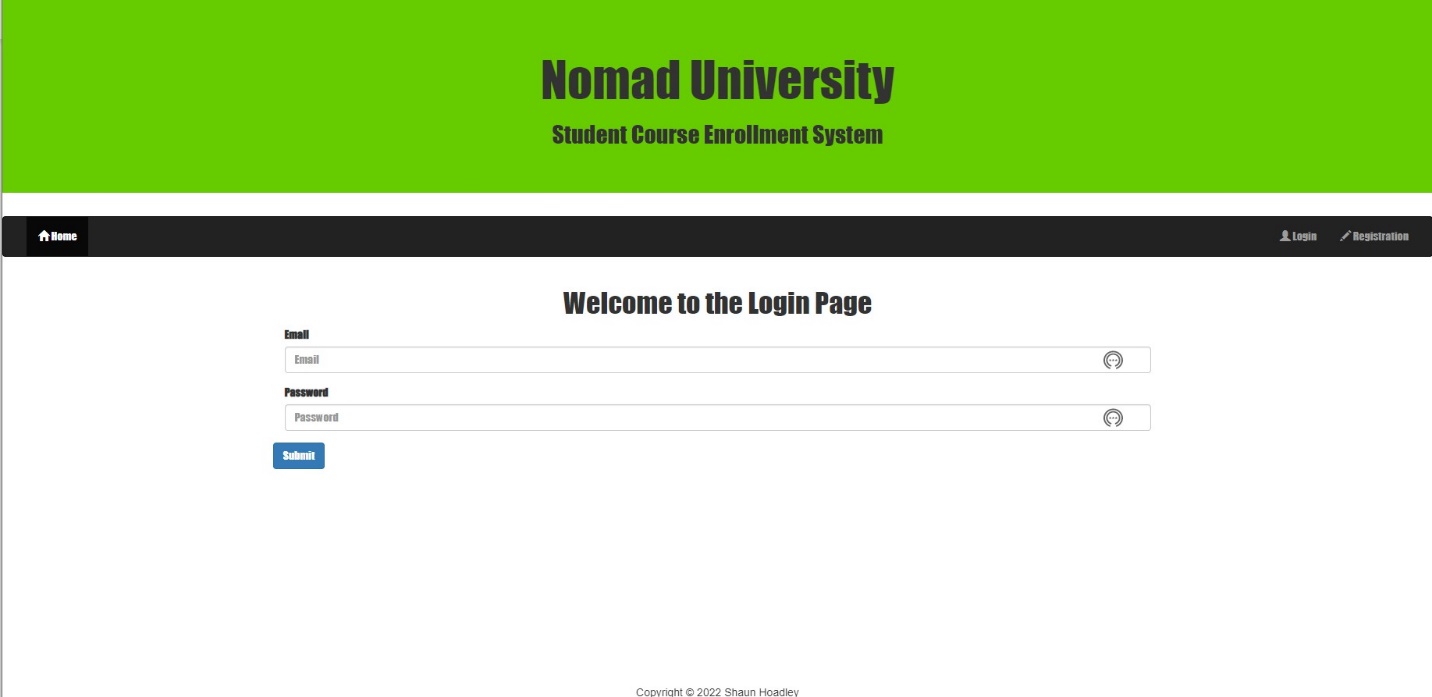
****

**Figure 17: Completed Registration Form**

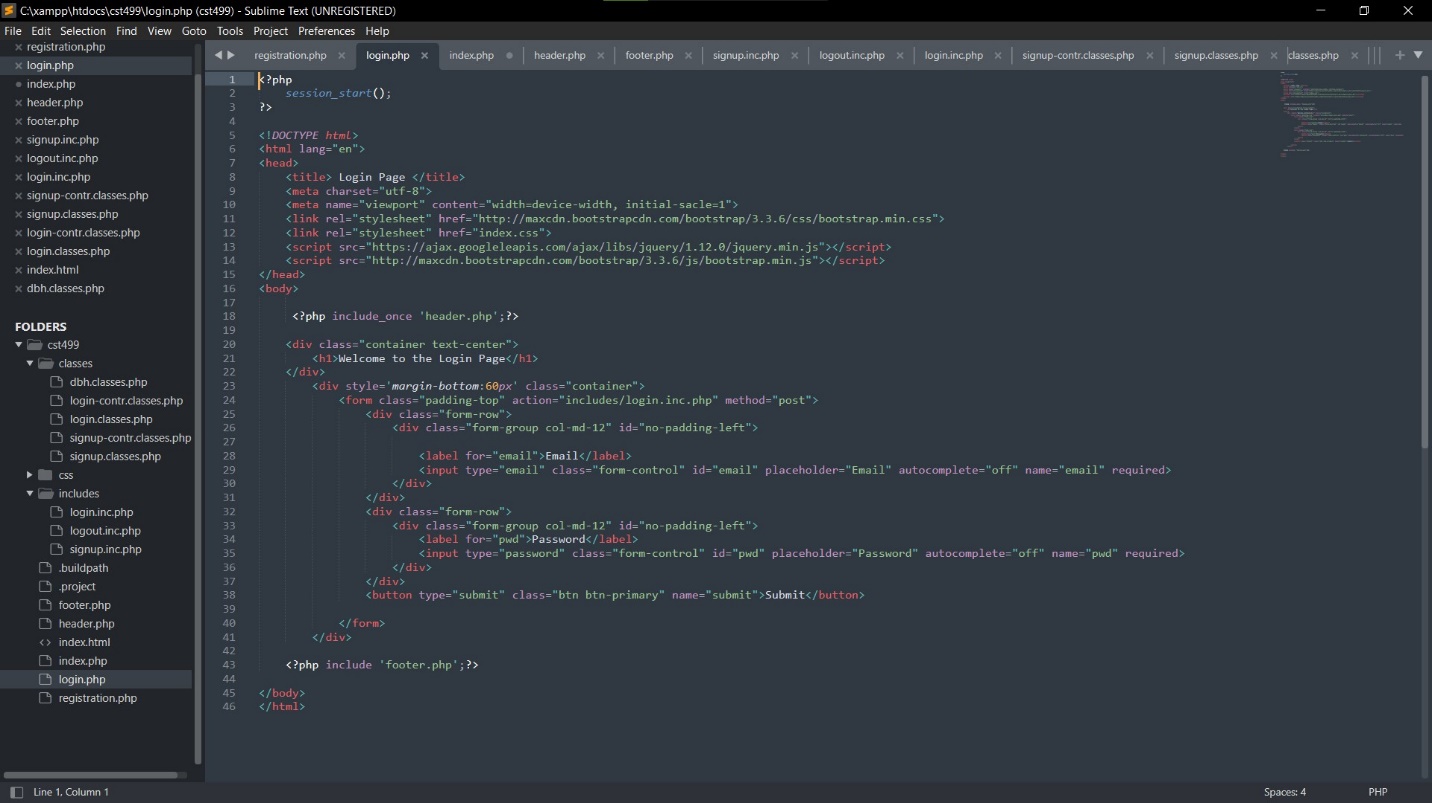
****

**Figure 18: User Table With Form Data Saved**

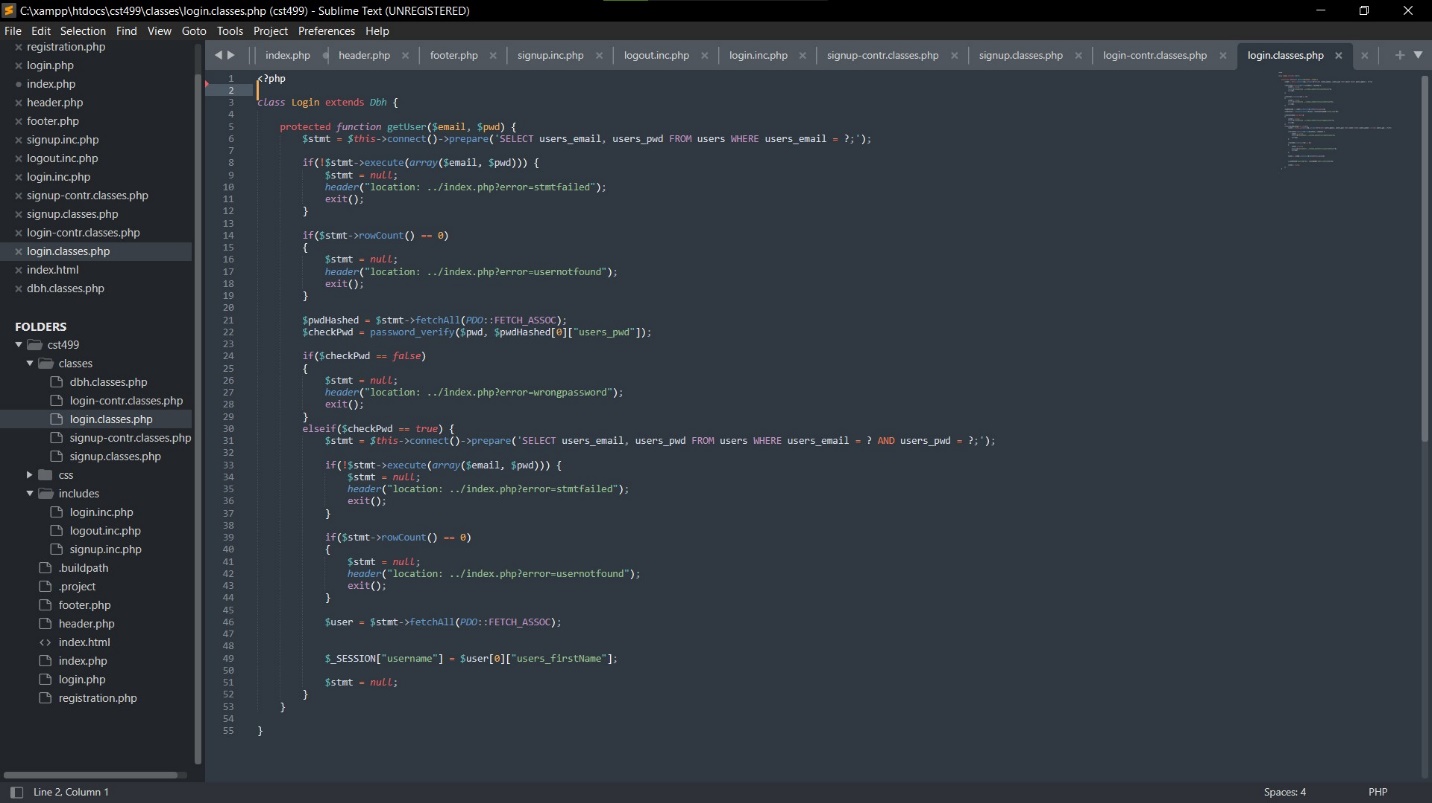
Now that the Registration page is completed, it was time to create the Login Page. The layout of the Login Page is demonstrated in Figure 19, followed by the source code for it in Figure 20. As with the registration page, there are additional files to handle the processing of the Login data. The login classes file source code is Figure 21, the login controller file is Figure 22, and the login includes file is Figure 23.

****

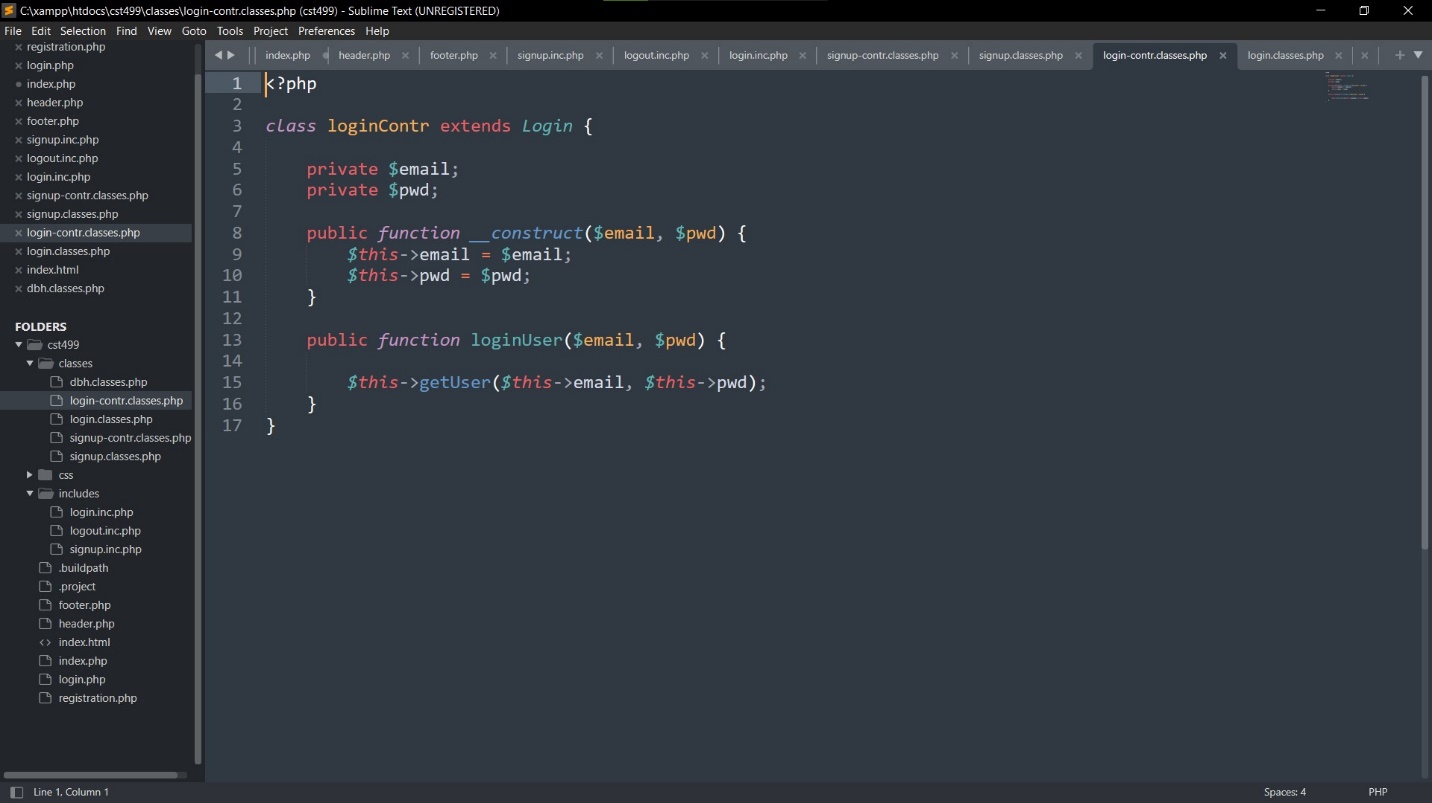
**Figure 19: Login Page Layout**

****

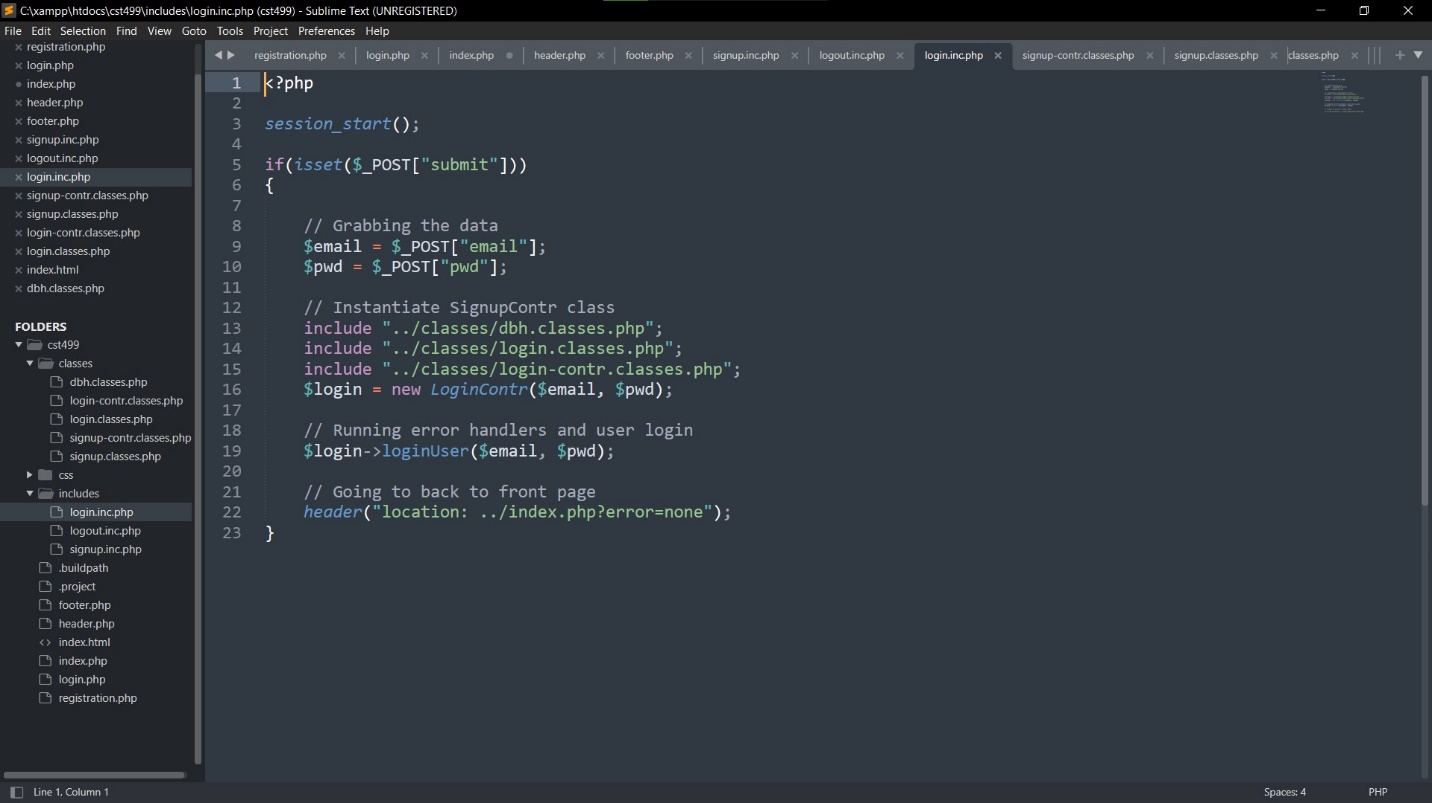
**Figure 20: Login Page Source Code**

****

**Figure 21: Login Classes Source Code**

****

**Figure 22: Login Controller Source Code**

****

**Figure 23: Login Includes Source Code.**

It is with the login page that I have hit a wall. What the login page is supposed to do is check the data entered into the form against the database and after verification, redirect the user to the landing page (for the time being) with the navigation bar options changed to reflect a logged in users options. I still do not have it working properly but I am confident I will have it figured out soon.

**References**

Mikoluk, K. (2013). XAMPP tutorial: How to use XAMPP to run your own web server [Blog post]. Retrieved from https://blog.udemy.com/xampp-tutorial