Landing, Login, and Enrollment Pages Development

Jamain Hughes

The University of Arizona Global Campus

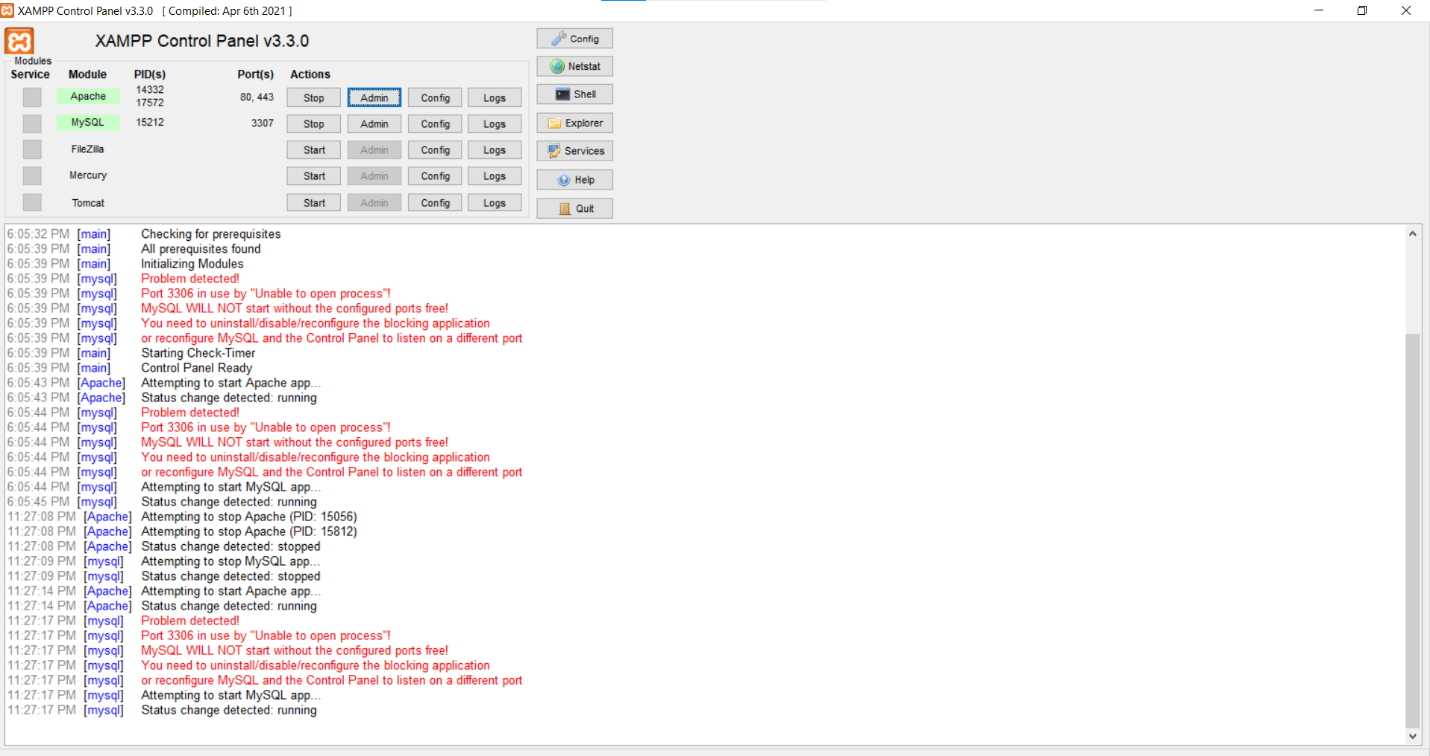
CST 499: Capstone for Computer Software Technology

Professor Amjad Alkilani

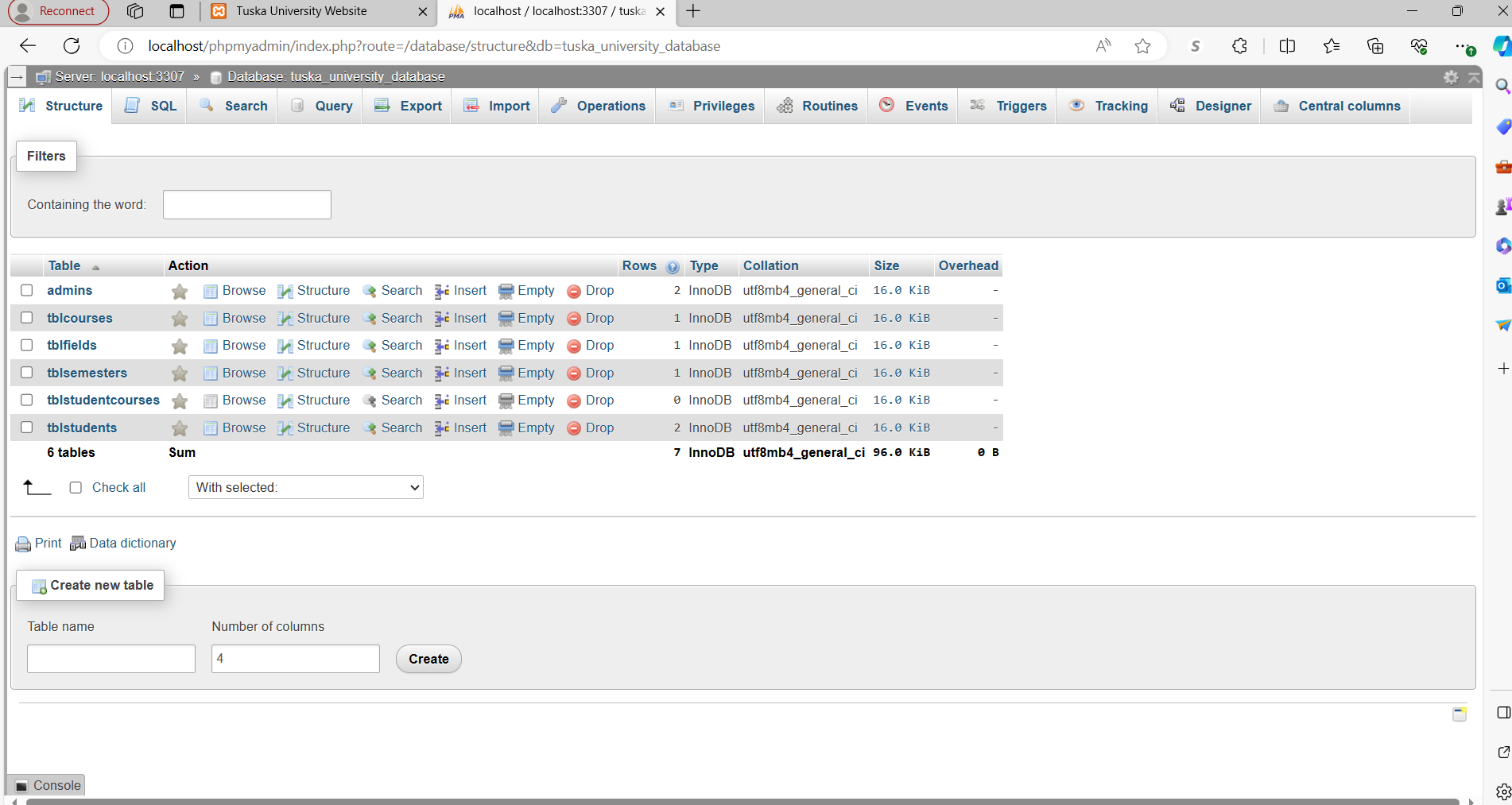
January 15, 2024

The development of an efficient online student registration system is very important for educational institutions. This paper describes steps involved in creating a student registration website for Tuska University. This paper also highlights the key elements from the execution of PHP files to the establishment of MySQL databases and tables. The narrative unfolds through the setup of essential technologies, the intricacies of database design, and the implementation of PHP coding, culminating in a robust registration system.

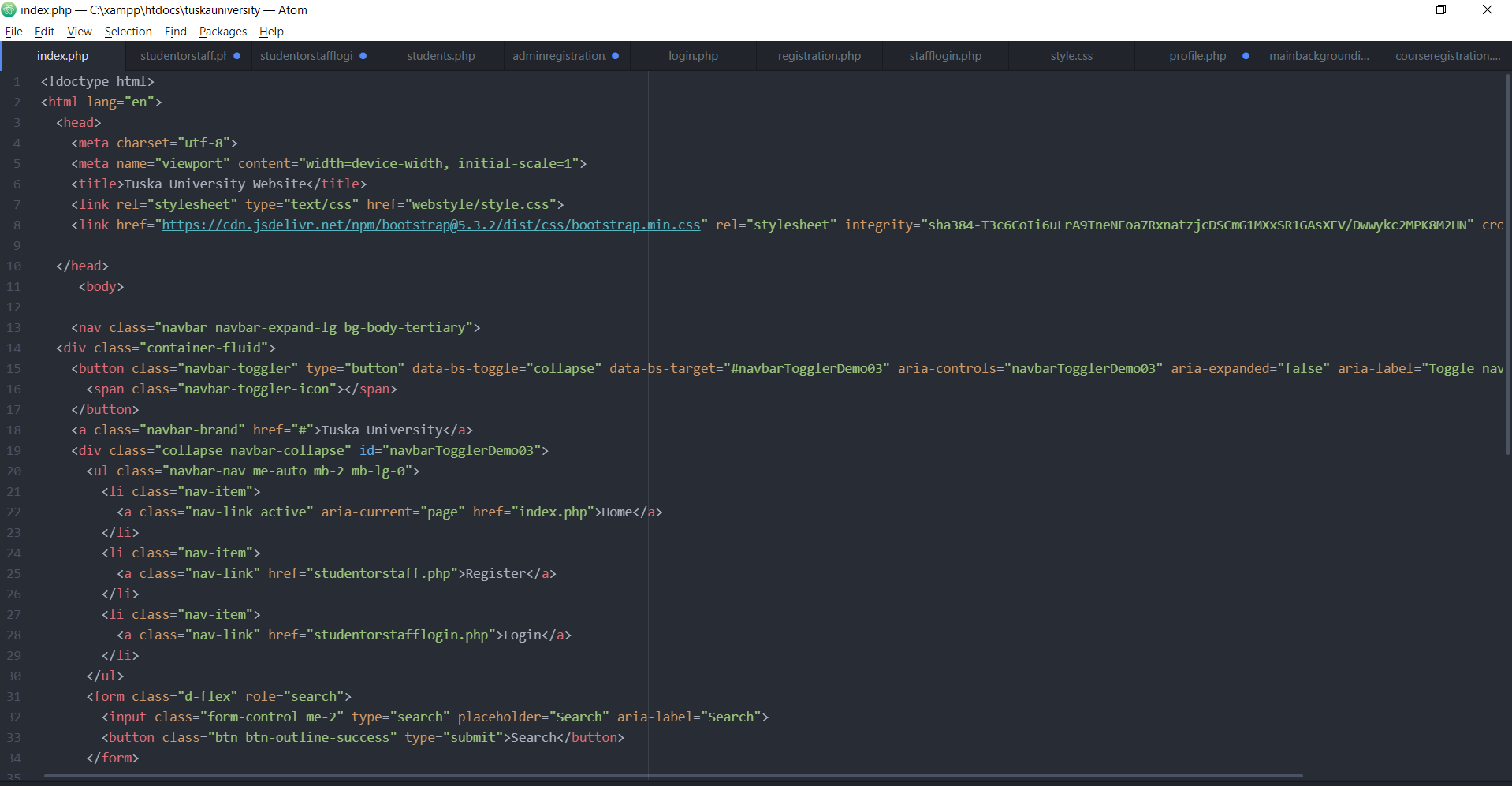
For this week’s assignment I created the landing page, registration page, and login pages for the online college website. I also created the MySQL database and tables for the website. Before beginning the implementation stage, I prepared the key technologies to create the dynamic website that connects to my MySQL database. For this project I used two main technologies to implement my code. One of the main technologies that I needed to prepare were a open-source code editor called Atom. I have been using Atom for quite some time so, I didn’t need to install it into my computer however, I did need to make sure it was updated before beginning implementing my student registration system. The second technology that I need to begin this project is a PHP development environment called XAMPP. XAMPP allows me to create a local web server environment to aid me in developing and testing my web application. I also use XAMPP regularly, so I didn’t need to install it. I needed to ensure that it was updated, and all of the ports were properly configured. After configuring everything I went to my XAMPP control panel and started the Apache and MYSQL server. After starting both servers, I created a new file in my code editor and named it confirm.php and saved it into a folder called demo. This confirm.php file was created to ensure that everything is working properly. To do so, I coded the file to print out “Hello World” in PHP. After typing the code, I save the file within my XAMPP directory inside of the htdocs folder and opened my web browser and typed <http://localhost/demo/cofirm.php/> in order to execute the code. This code execution was a successful attempt, and I was then ready to begin the implementation stage of my software system.

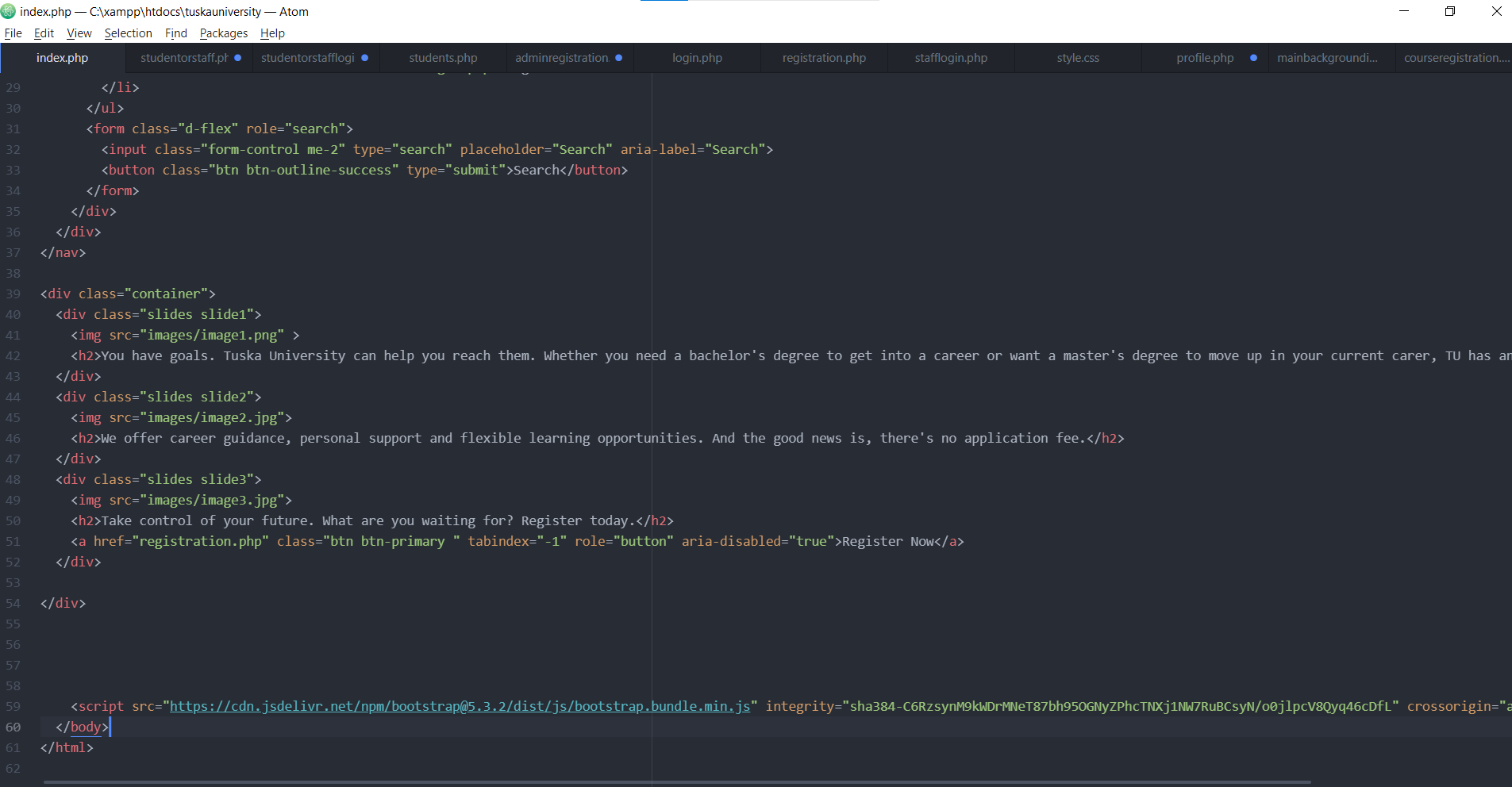


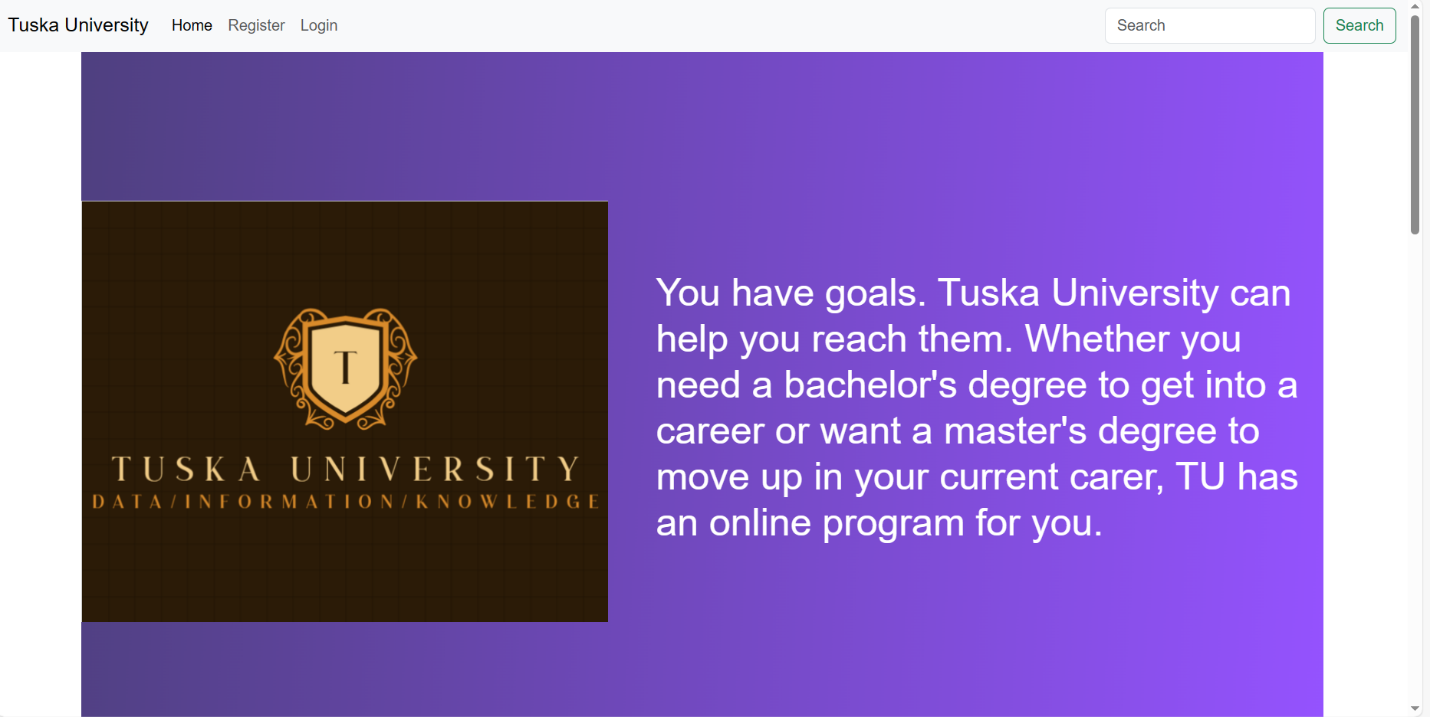
From there I created another folder inside the XAMPP htdocs folder and named it tuskauniversity. This file will hold all the documents pertaining to the student registration system. Before beginning my landing page, I opened MySQL admin and created a database schema called tuska\_university\_database. This database has six tables. These tables are labeled admins, tblcourses, tblfields, tblsemesters, tblstudentcourses, and tblstudents. The table that is labeled admin holds all of the data for the administrative staff. This table saves various types of information like the administrator ID, username, password, email, and role. This table will allow administrators to create their own profile in order to add courses to the school’s schedule and assist students with problems they are having regarding registration. The table labeled tblcourses holds all of the information regarding all of the courses that students will be registering for. The rows in this table consist of courseID, courseName, courseCode, fieldID, and semesterID. The table labeled tblfields contain the data of the different fields of study that students will be able to apply to. This table will allow students to easily find the courses that they need to take within their specific programs. This table contains two rows labeled fieldID and fieldName. The table labeled tblsemesters will hold all of the data pertaining to the summer, fall, and winter semesters. The rows within this table consist of semesterID and semesterName. The table labeled tblstudentcourses will have all the data pertaining to the different courses available within the school’s schedule. This table consist of five rows labeled studentCourseID, studentID, courseID, semesterID, and fieldID. The table labeled tblstudents will hold all the data pertaining to every student that has successfully registered within the school’s website. The rows of this table are labeled studentID, studentFirstName, studentLastName, studentPhoneNumber, studentEmail, studentPassword, fieldName, studentAddress, profilePicture, semesterName, and courseName. After creating all the necessary tables I needed to combine records from all the tables that have matching values in common fields through an INNER JOIN.

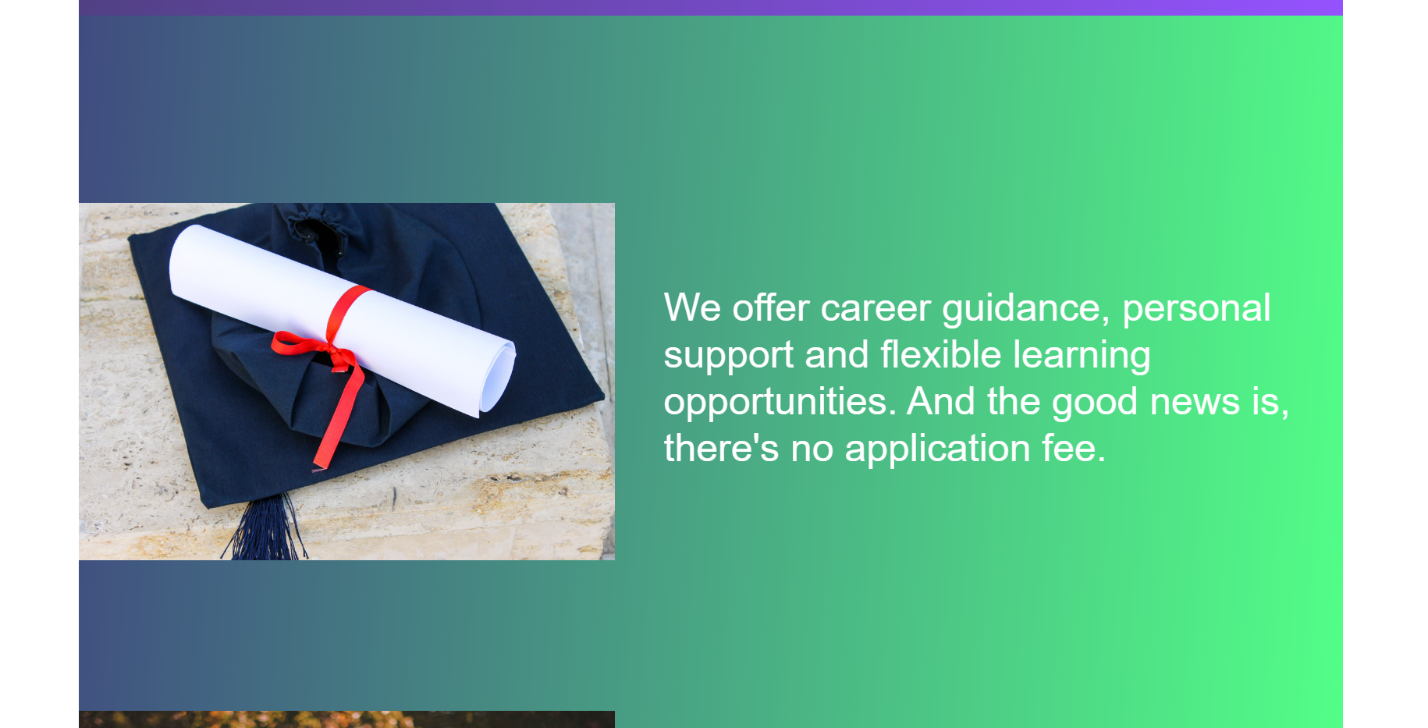


After creating my database schema, I began working on the landing page. I created a new file and named it index.php and saved it to my tuskaunversity folder. My index file is the homepage of the website that contains a navbar with clickable links that direct the user to a registration, login, and home page. The design of this page is rather simplistic with three images that has a brief description of what the college has to offer followed by a registration button on the bottom of the page.

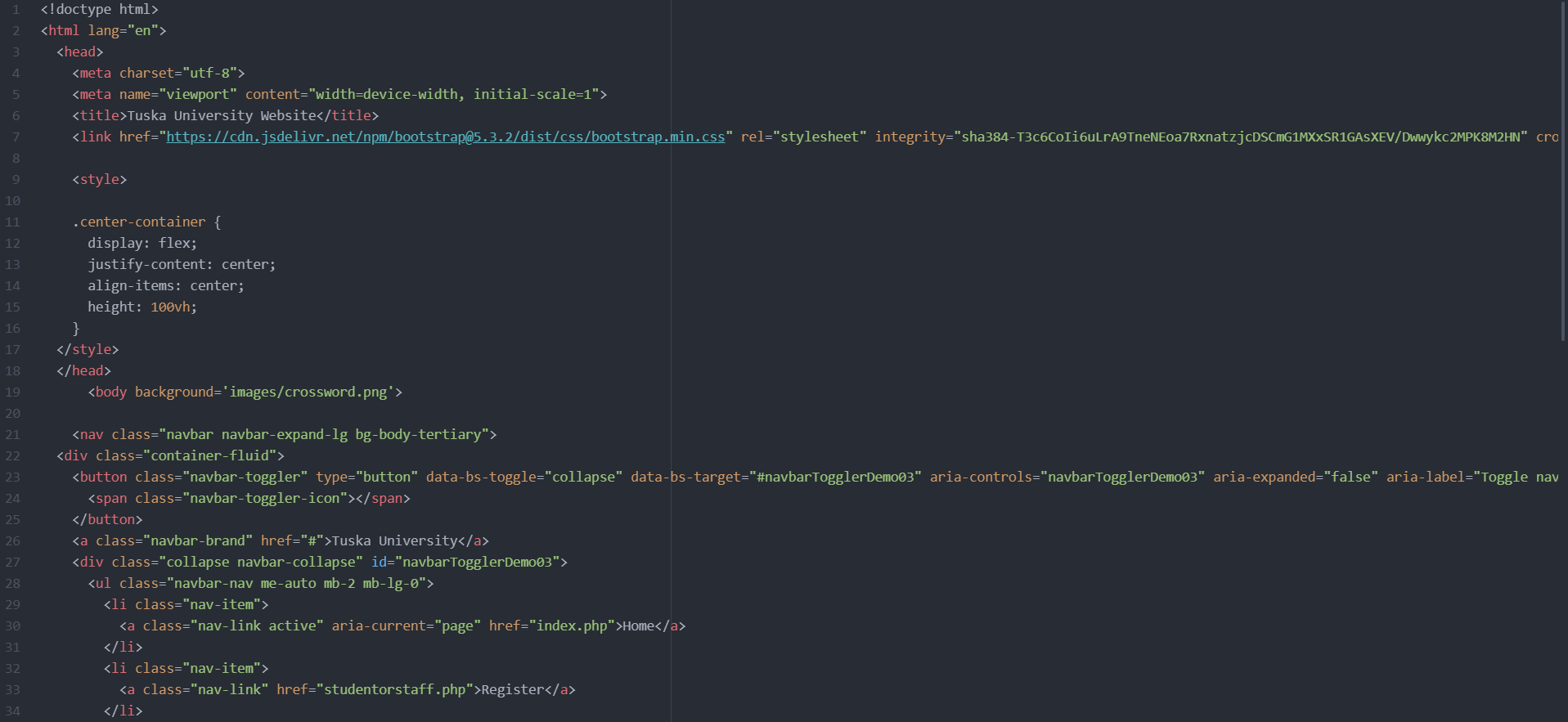


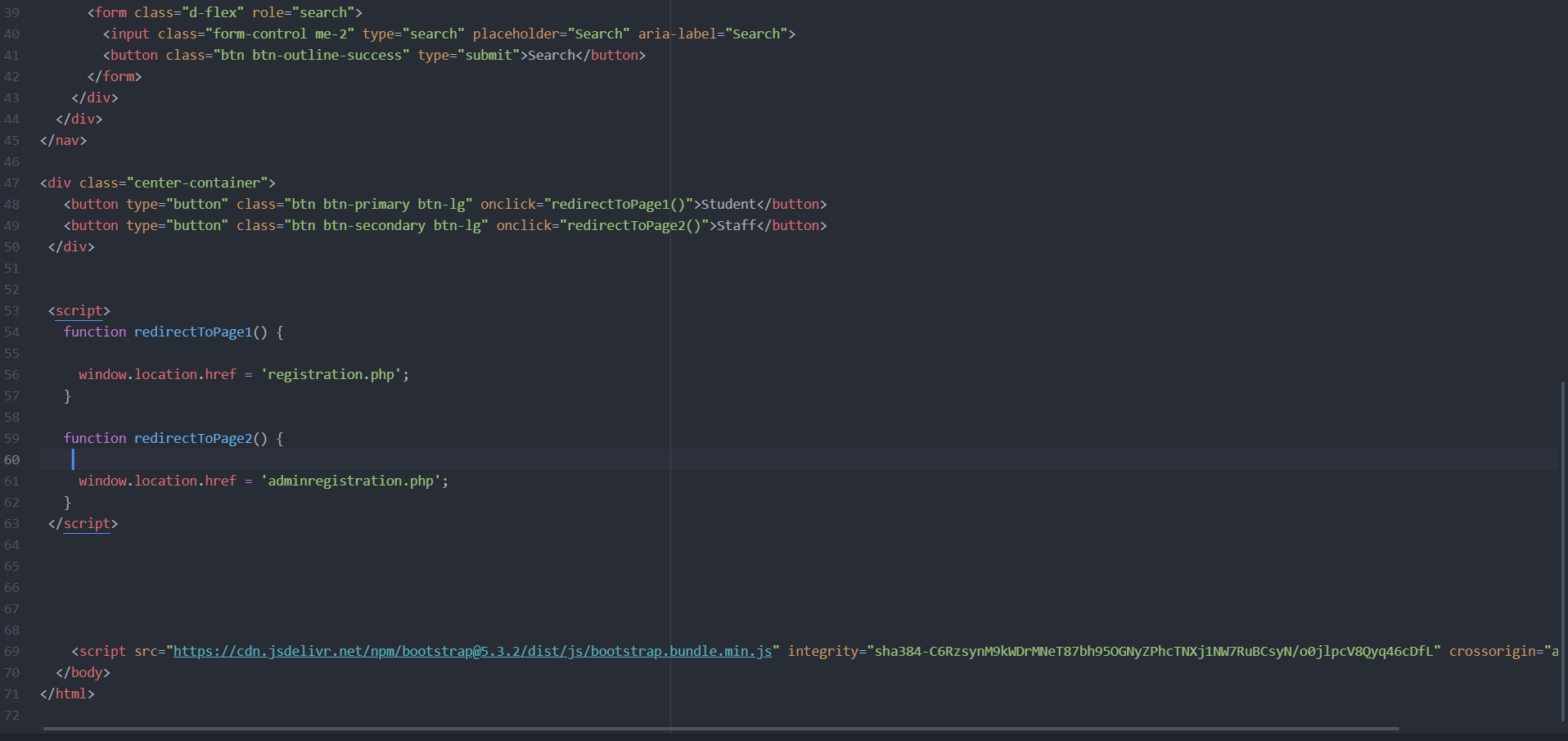




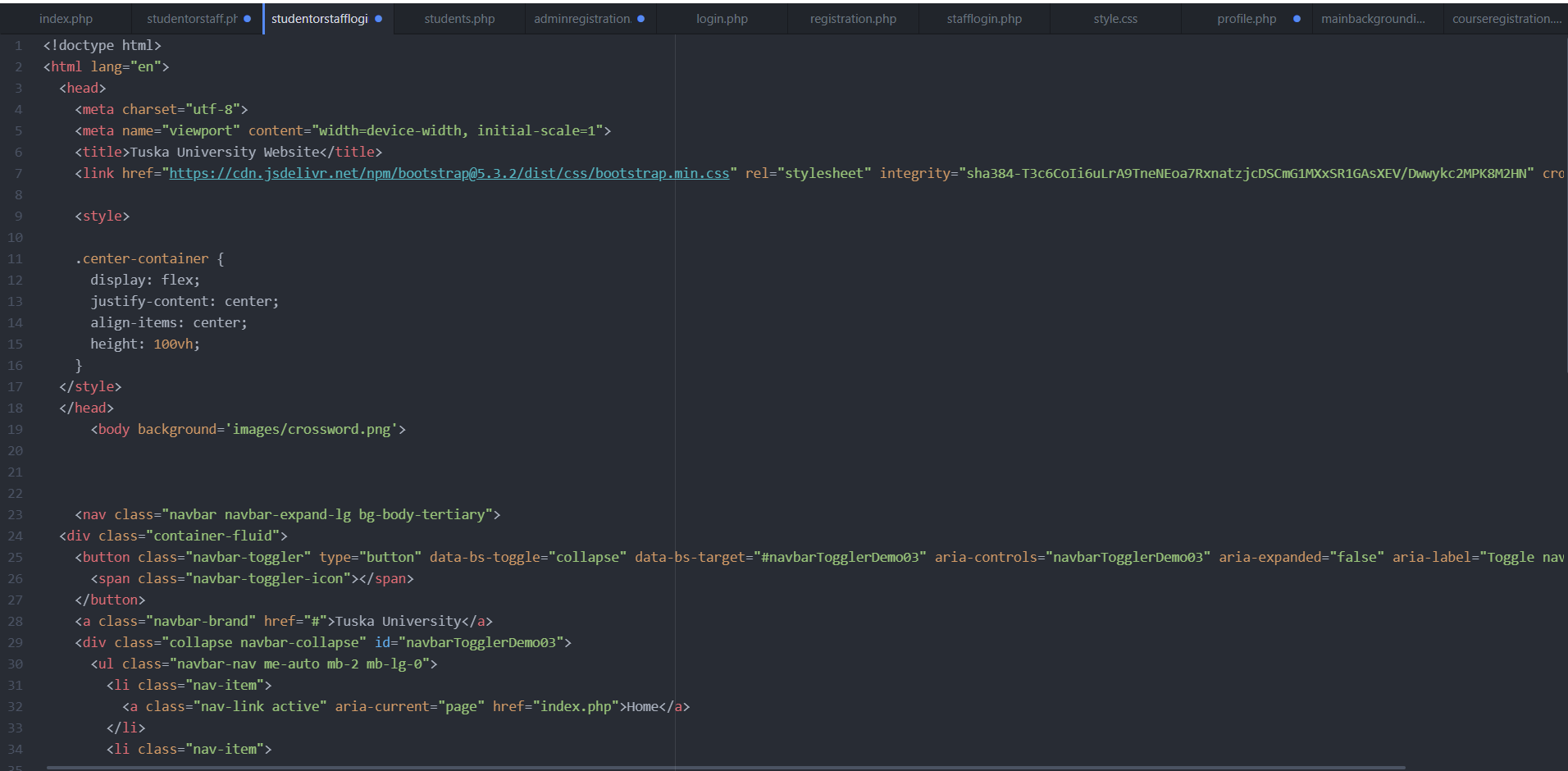


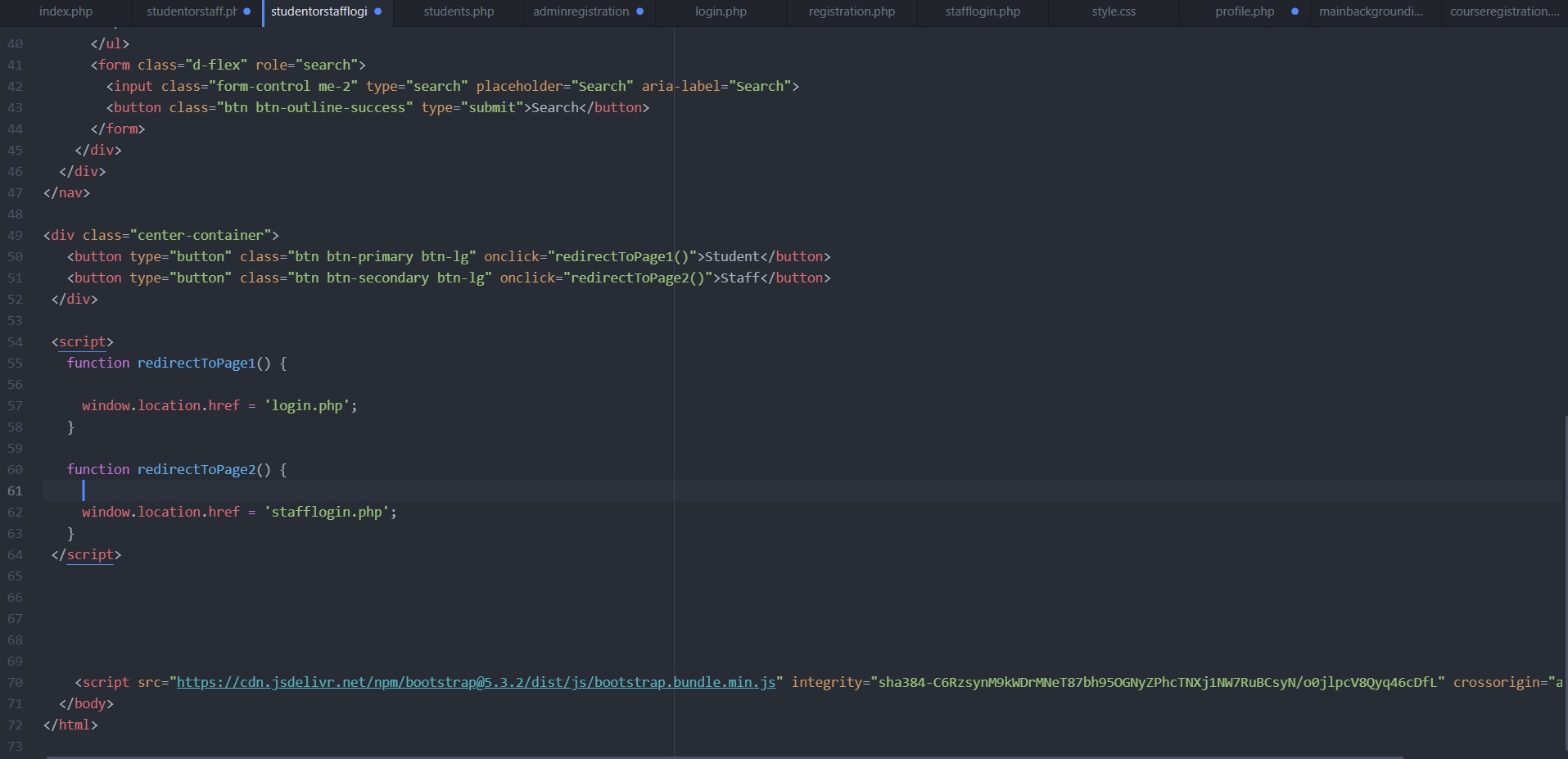


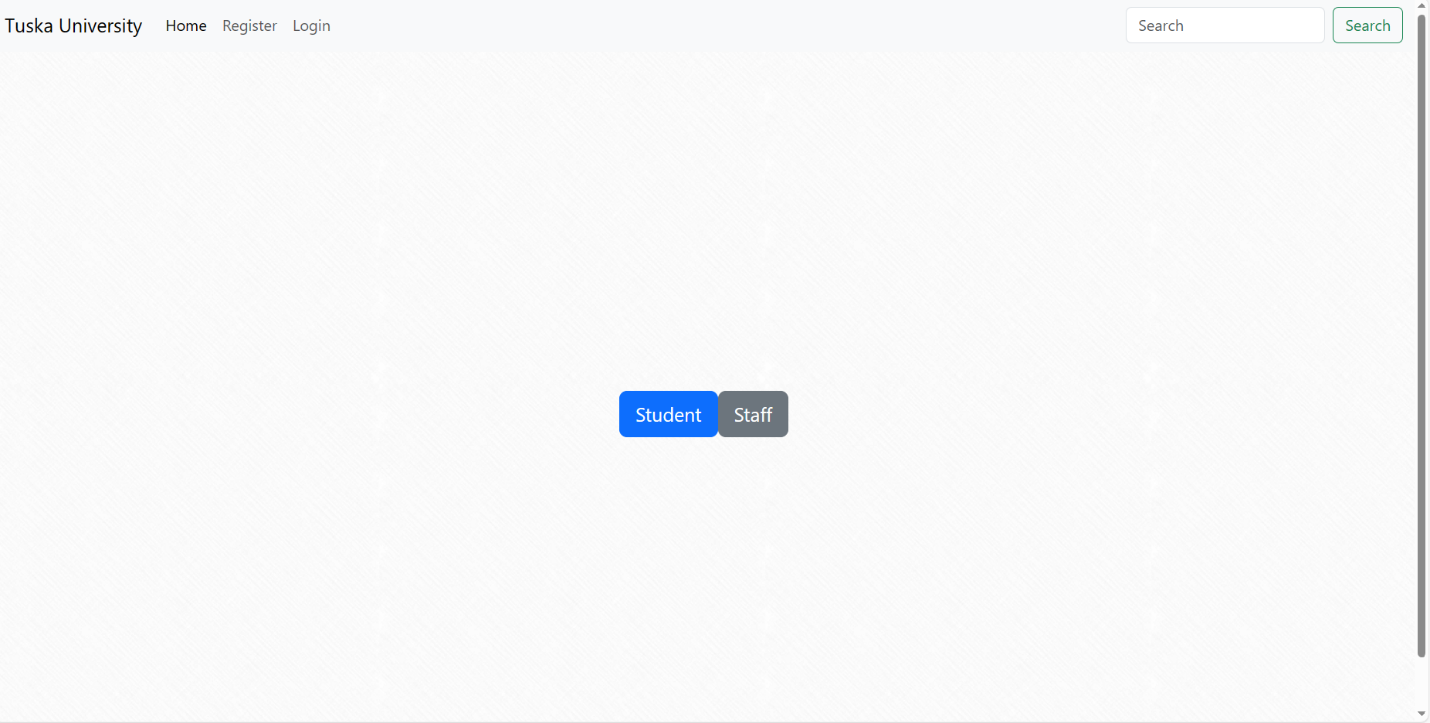




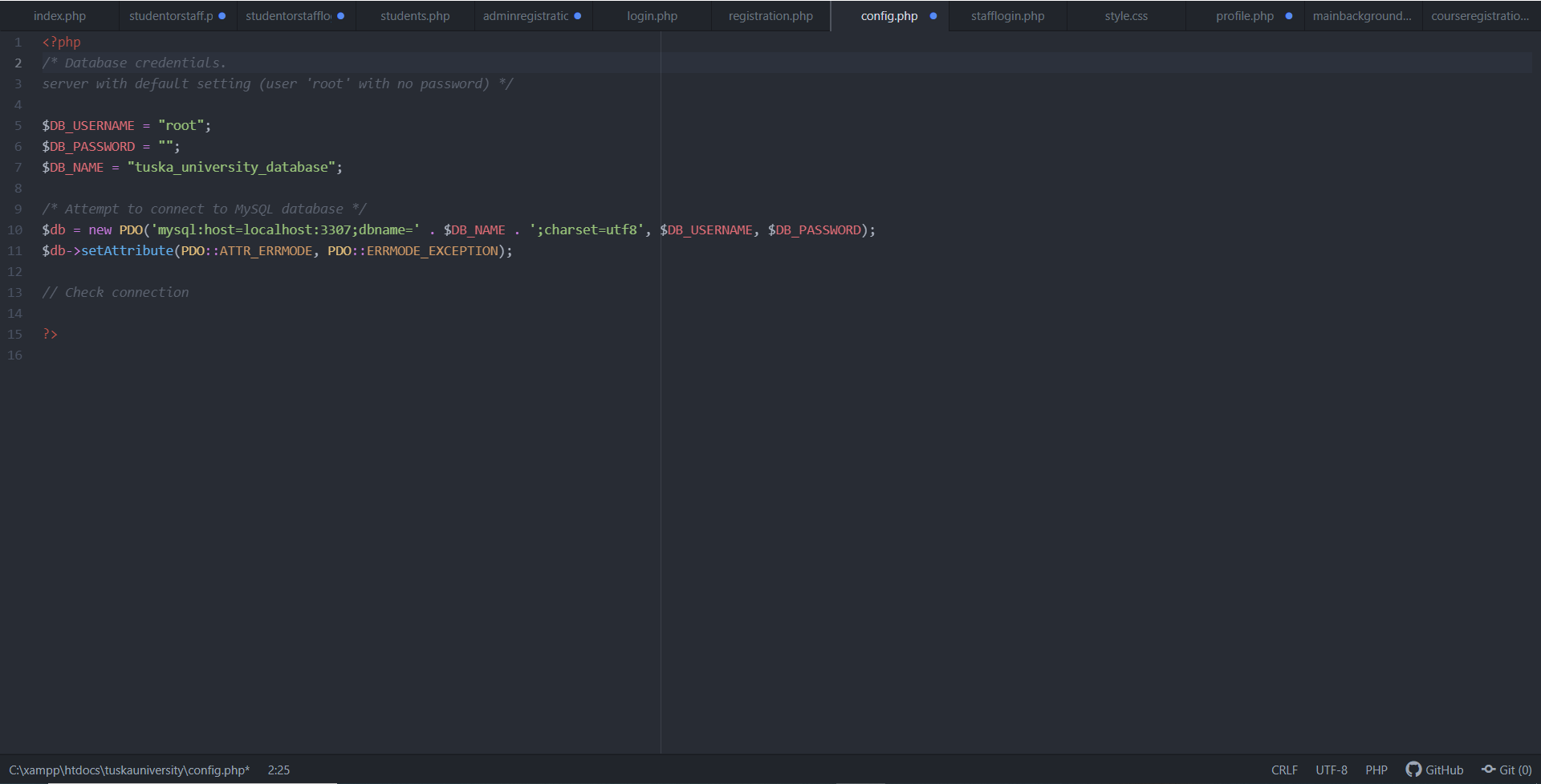
This website will not only allow students to create an account, but administrators will also be able create their accounts as well. To allow both students and administrators to create their accounts I created a page called studentorstaff.php. This page has two clickable buttons. One button is labeled student. This button will allow students to classify that they are looking to register as students. The other button is labeled staff. This button will allow administrators to classify that they are registering as staff. I also oriented the login process the same way. I understand that I could have programmed a dropdown menu allowing students and staff to select who they are but, I programmed the website in such a way to make it look more professional.



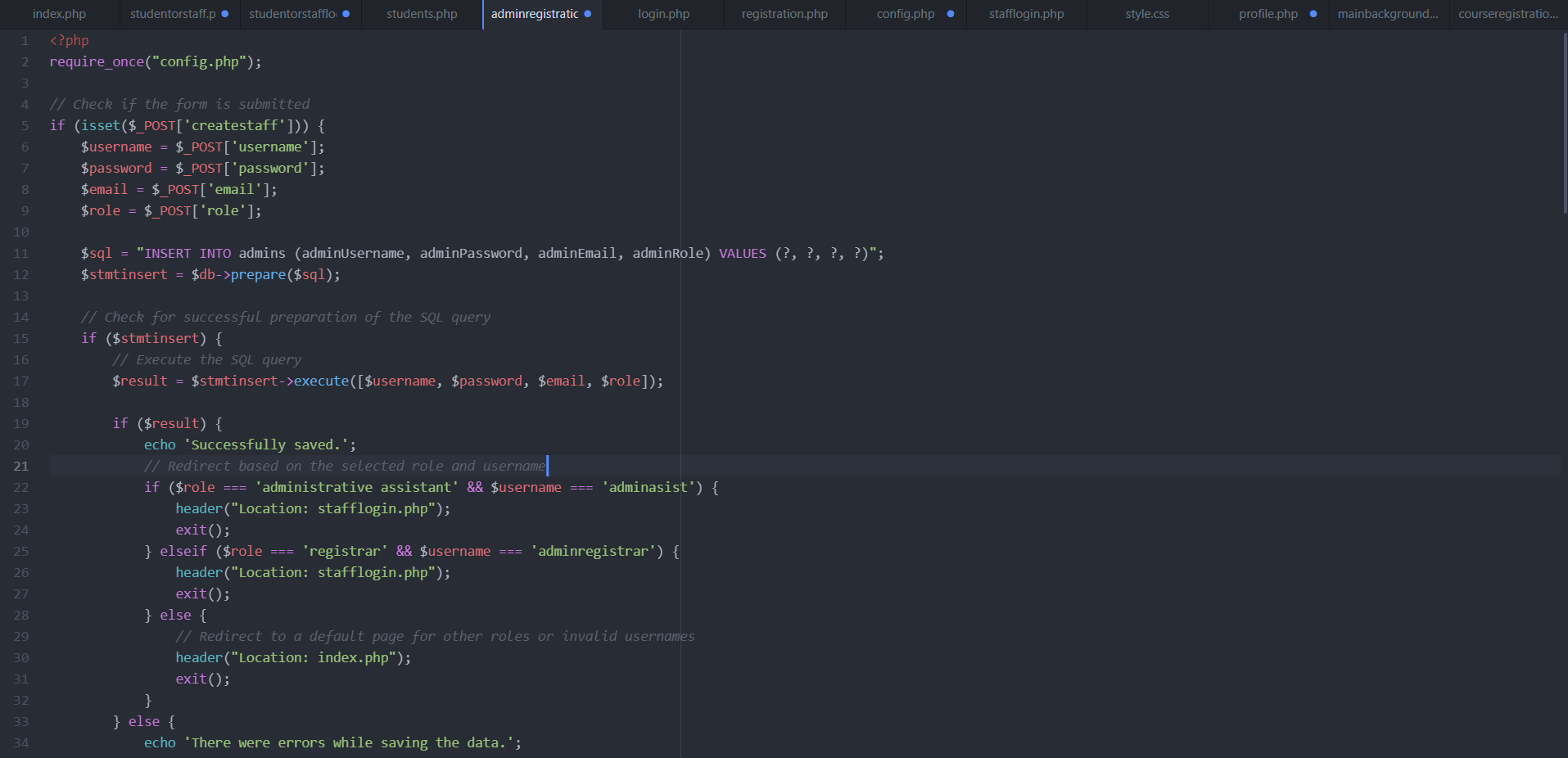


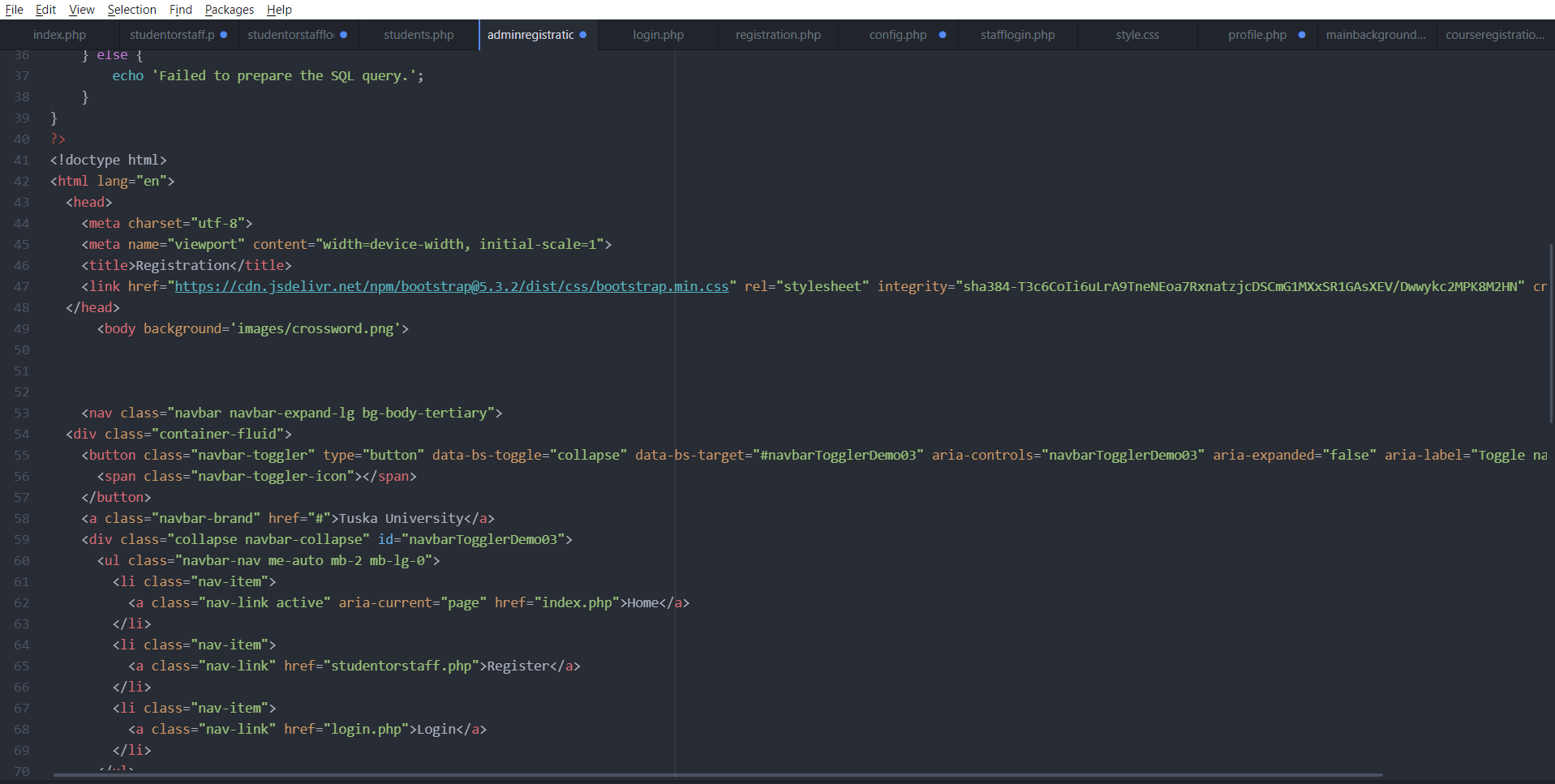


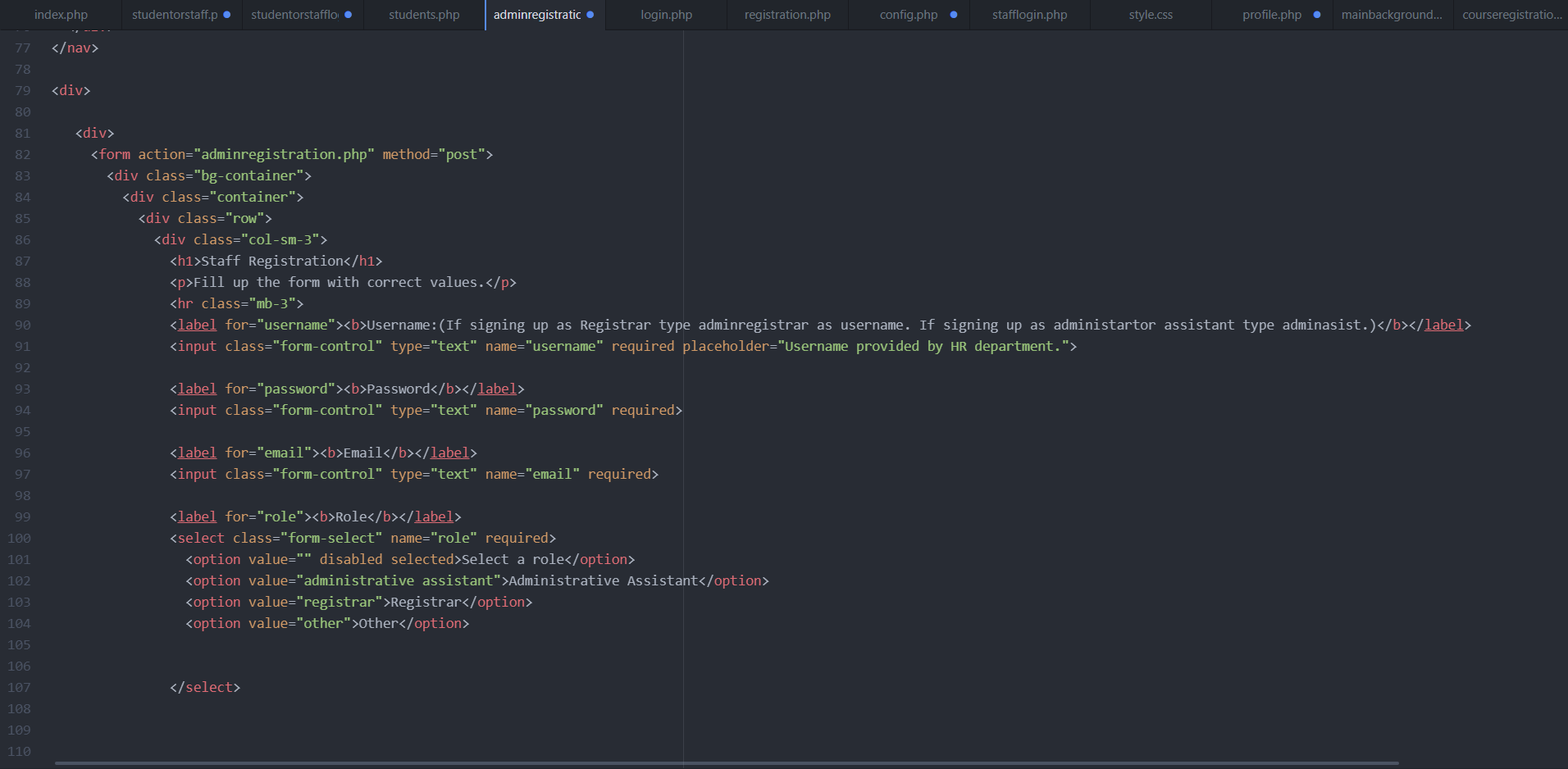
Both administrators and students will need to register to use this software system. In order to achieve this goal, there needs to be a connection to the database that I created. Through the database, the software system will be able to save the data of each student and administrative staff so that they can login into the system whenever they need to use it. To establish a connection to the database, I created a file called config.php. This configuration file holds the database credentials and it also connects to the MySQL database and check for the connecting. For this logic I used a PDO approach using PHP. The reason why I decided to use the PDO approach instead of the object-oriented approach because it offers a data-access abstraction layer, which means that I can issue queries and fetch data using the same functions regardless of which database I’m using. The PDO approach also provides protection from SQL injection because it uses a prepared statement.

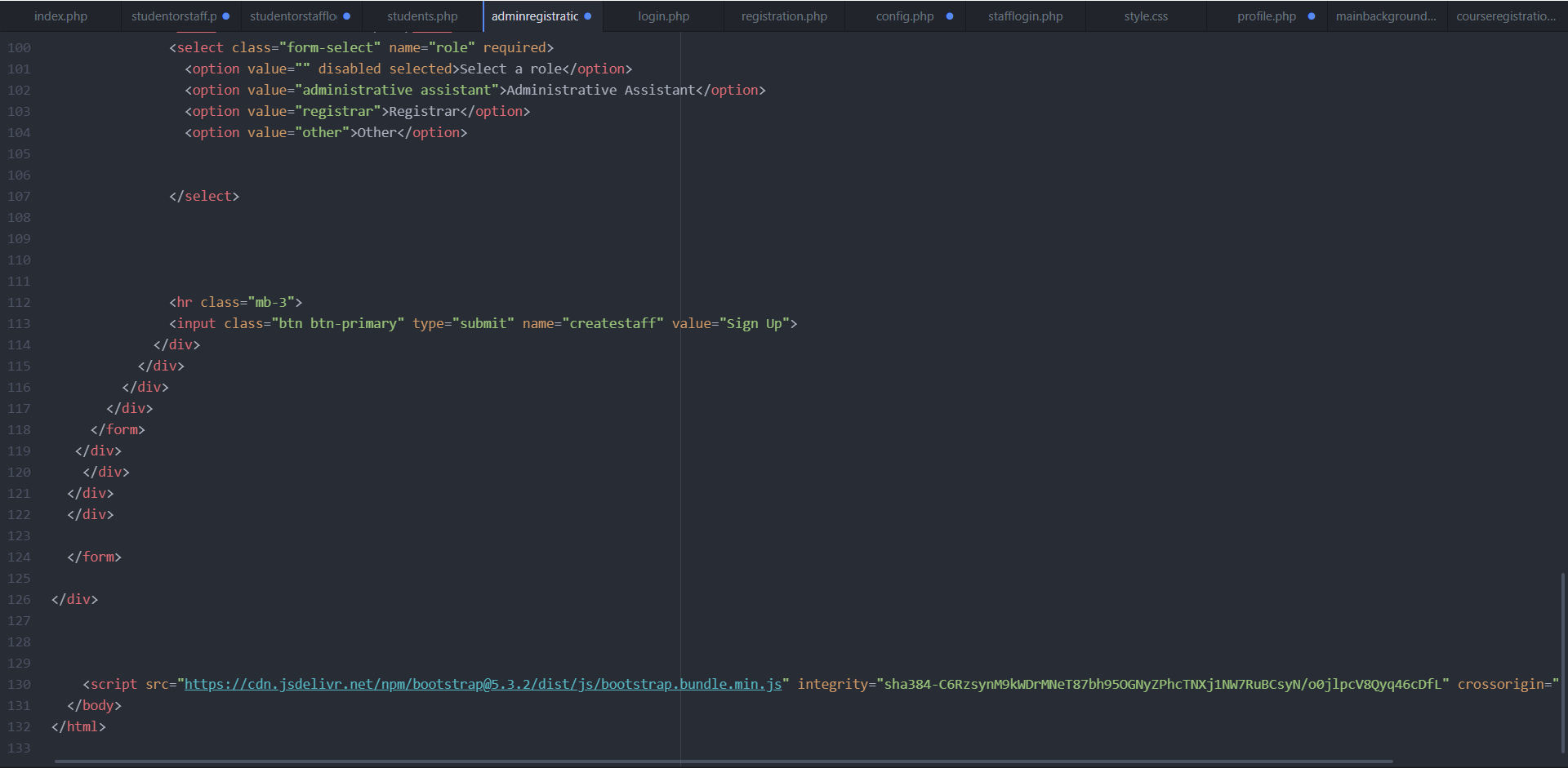


After creating my config.php file I created a file called registration.php. This file will be responsible for allowing new students to register as Tuska University alumni. This registration page is a simple form nested inside of a smooth light background called crossword. The form asks for students’ first name, last name, physical address, phone number, email address, and password. For this file I used HTML5, PHP, and CSS in order to make a functioning registration form. Through PHP I was able to begin a session and include the config.php file to allow the page to connect to my database. From there I programmed the page to take in the information entered by the students and save it to the database after the student have submitted their information. I have structured the administrative staff registration page in a similar way to the student registration page. The only difference between the two pages is the data that each admin must type to register. To promote security posture, administrators are given a username provided by the HR department. Once they begin the register, they need to insert their given username into the registration form and if that username is a match, then they are successfully registered as a user. This form also asks for them to create a password, email address, and their specific administrative role. To make this process as simplistic as possible, I typed a prompt above the username box giving out each the correct usernames for each administrator based on their role.

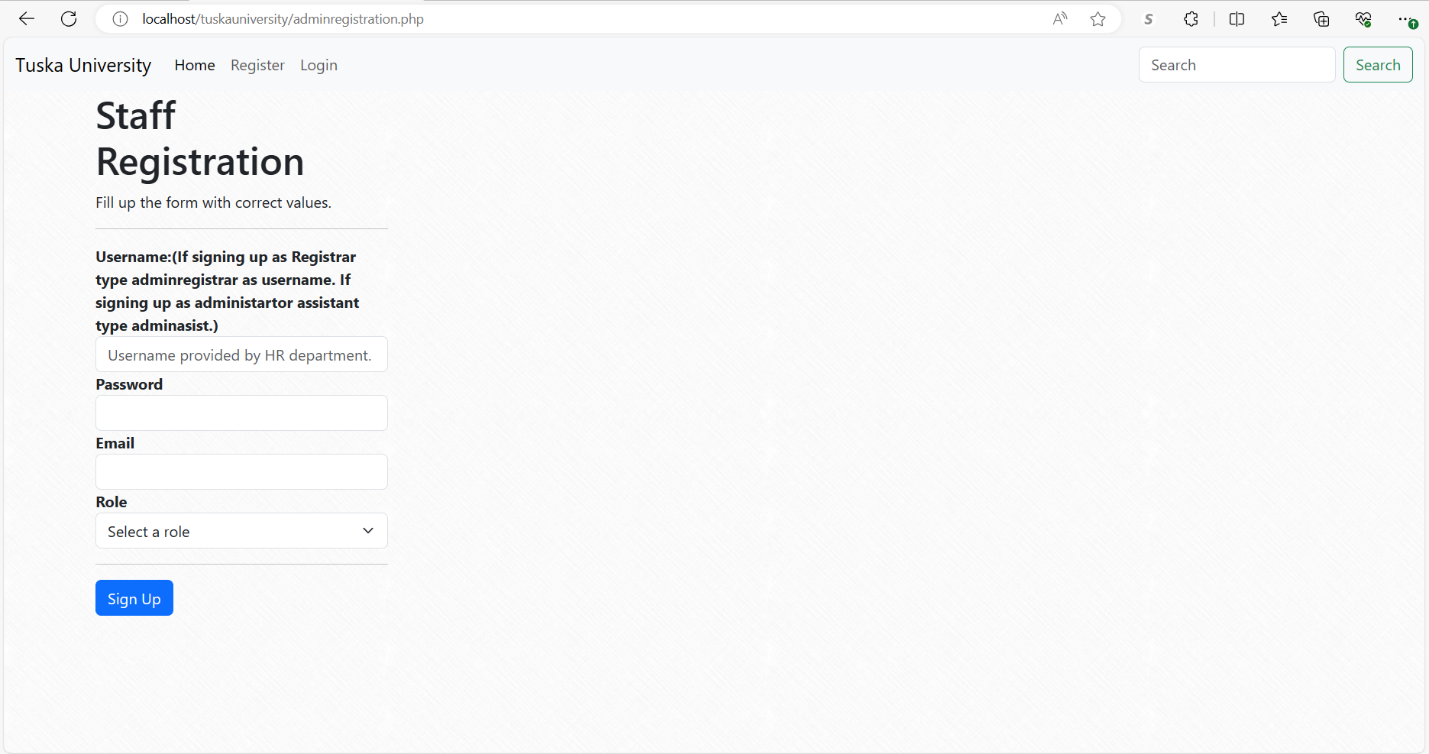




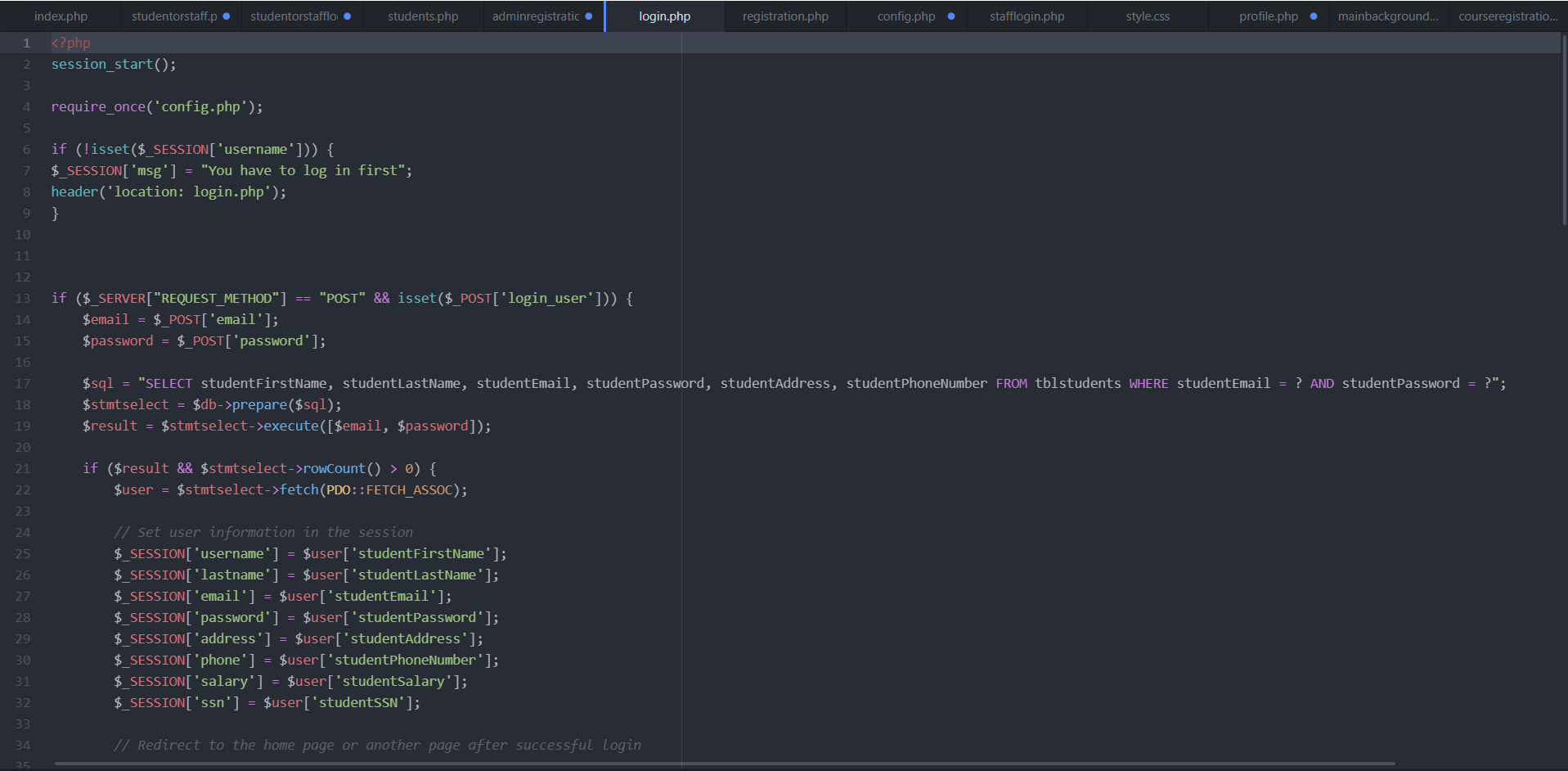


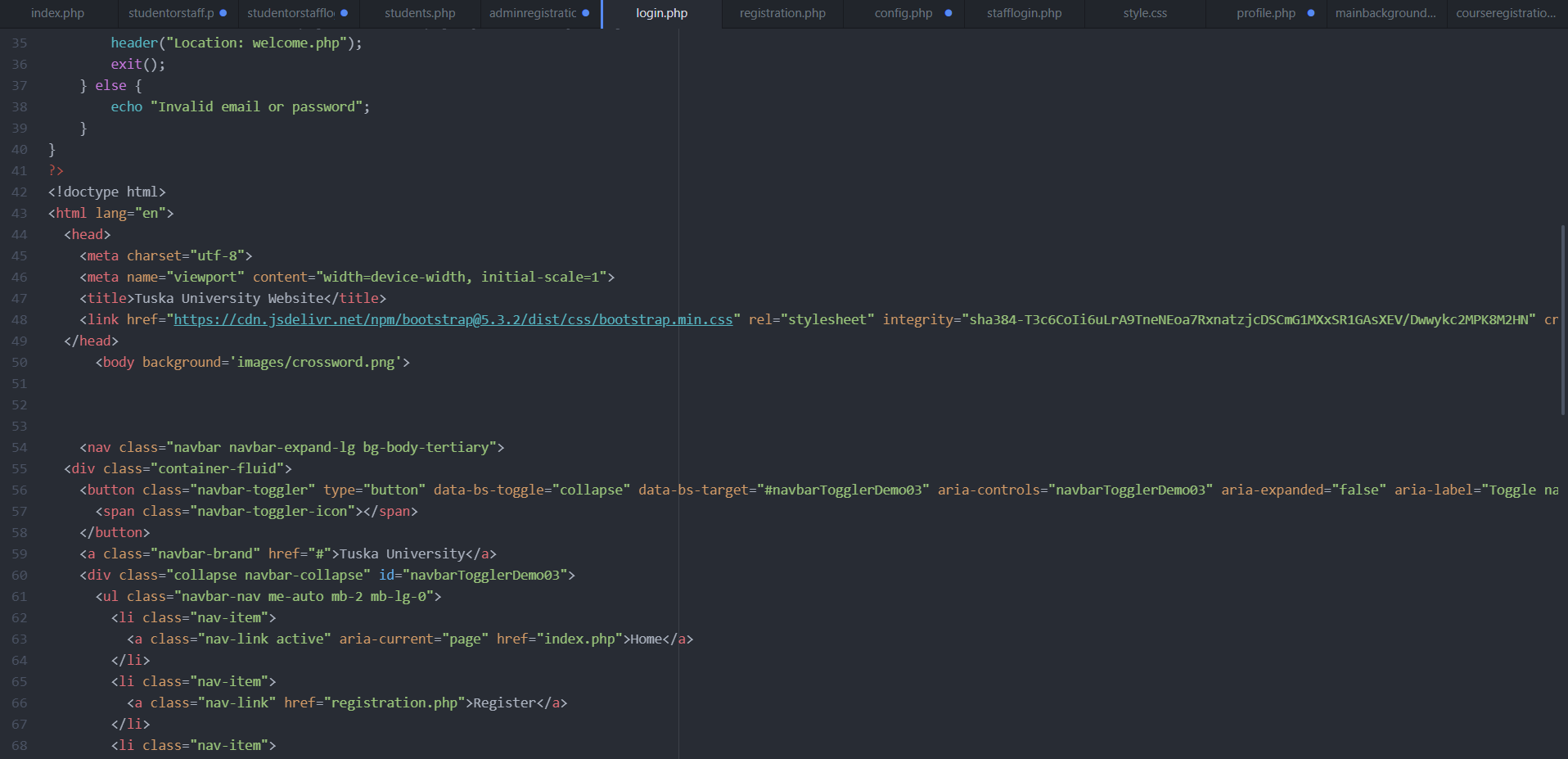


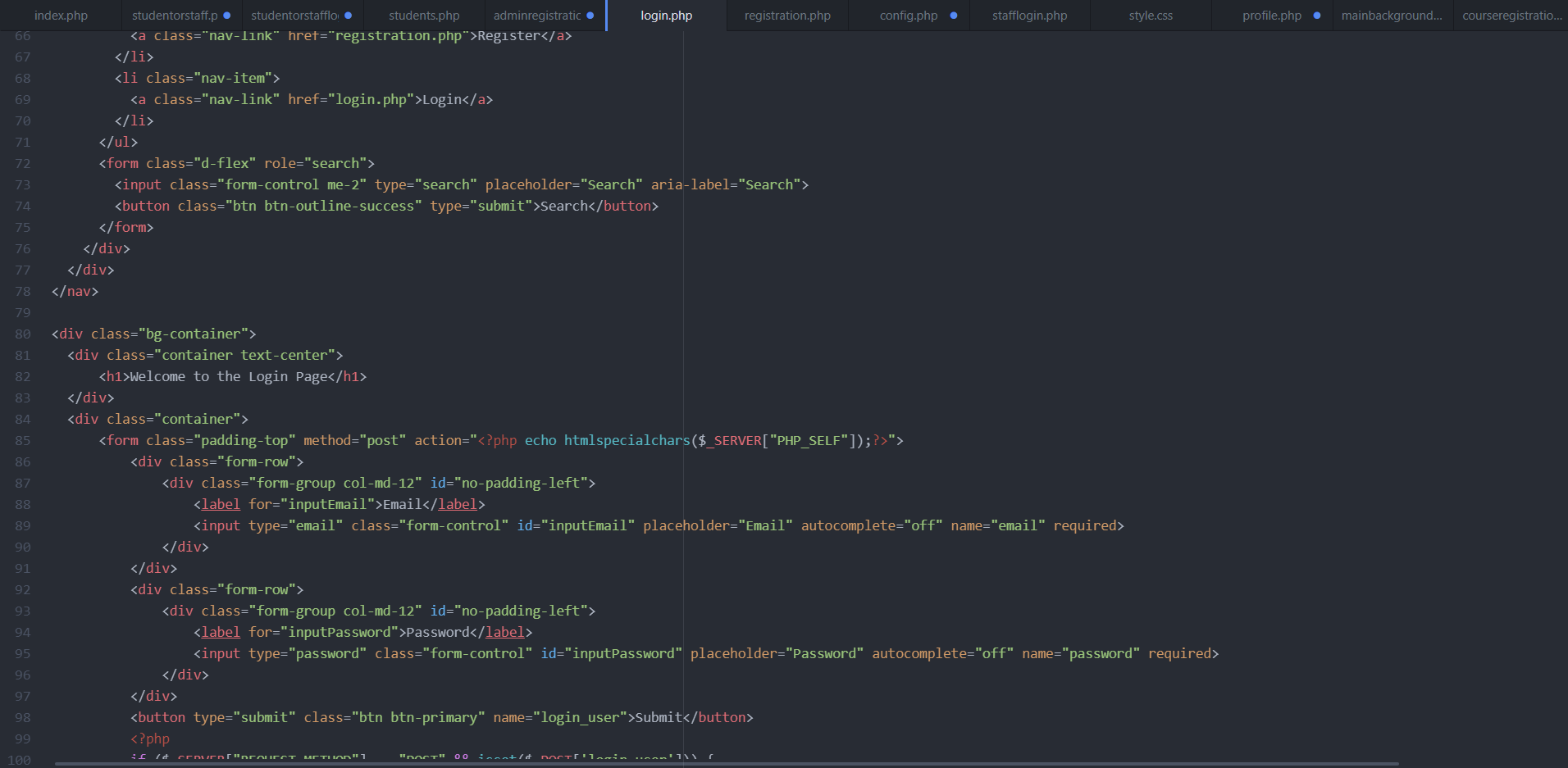


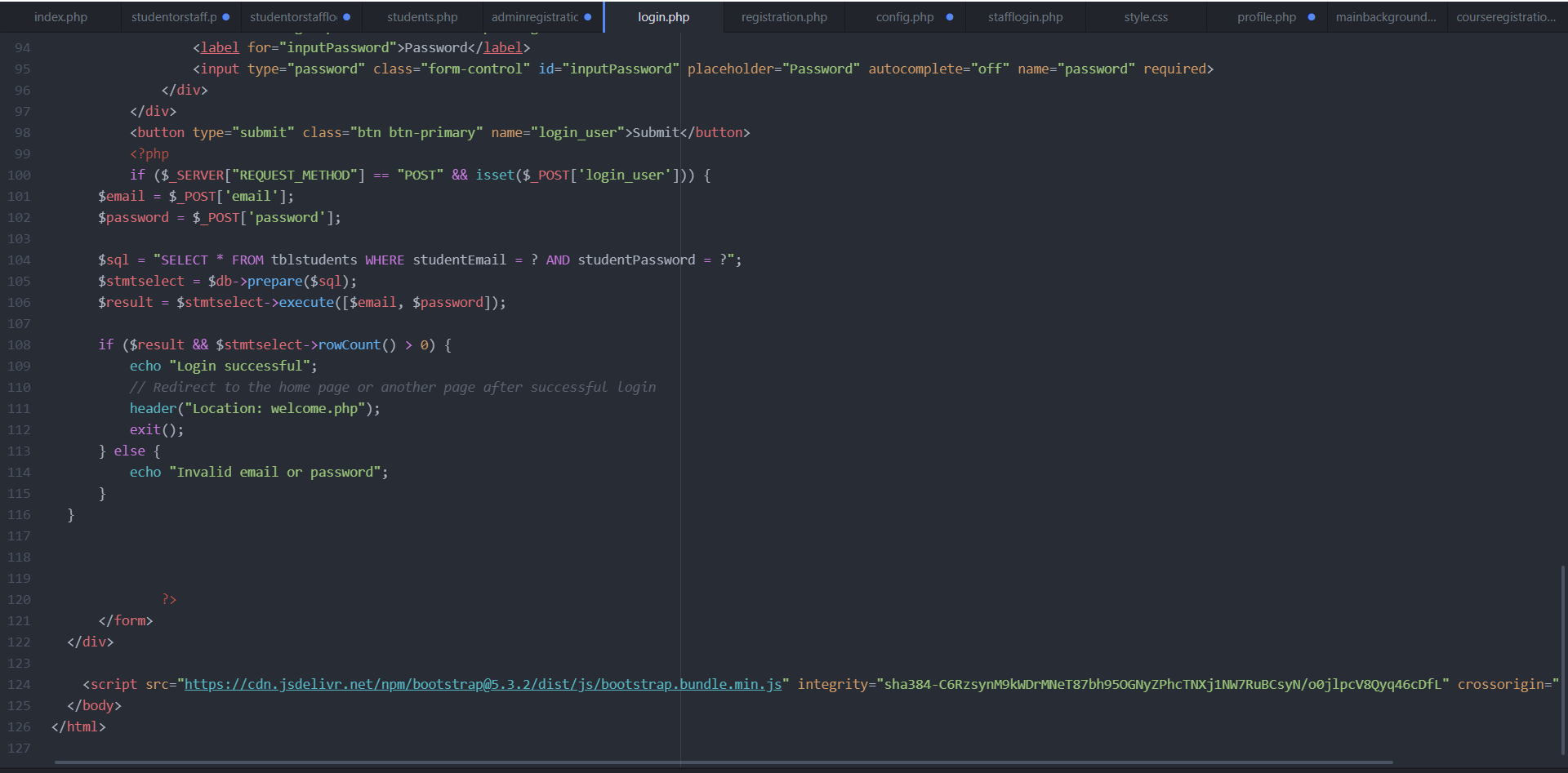


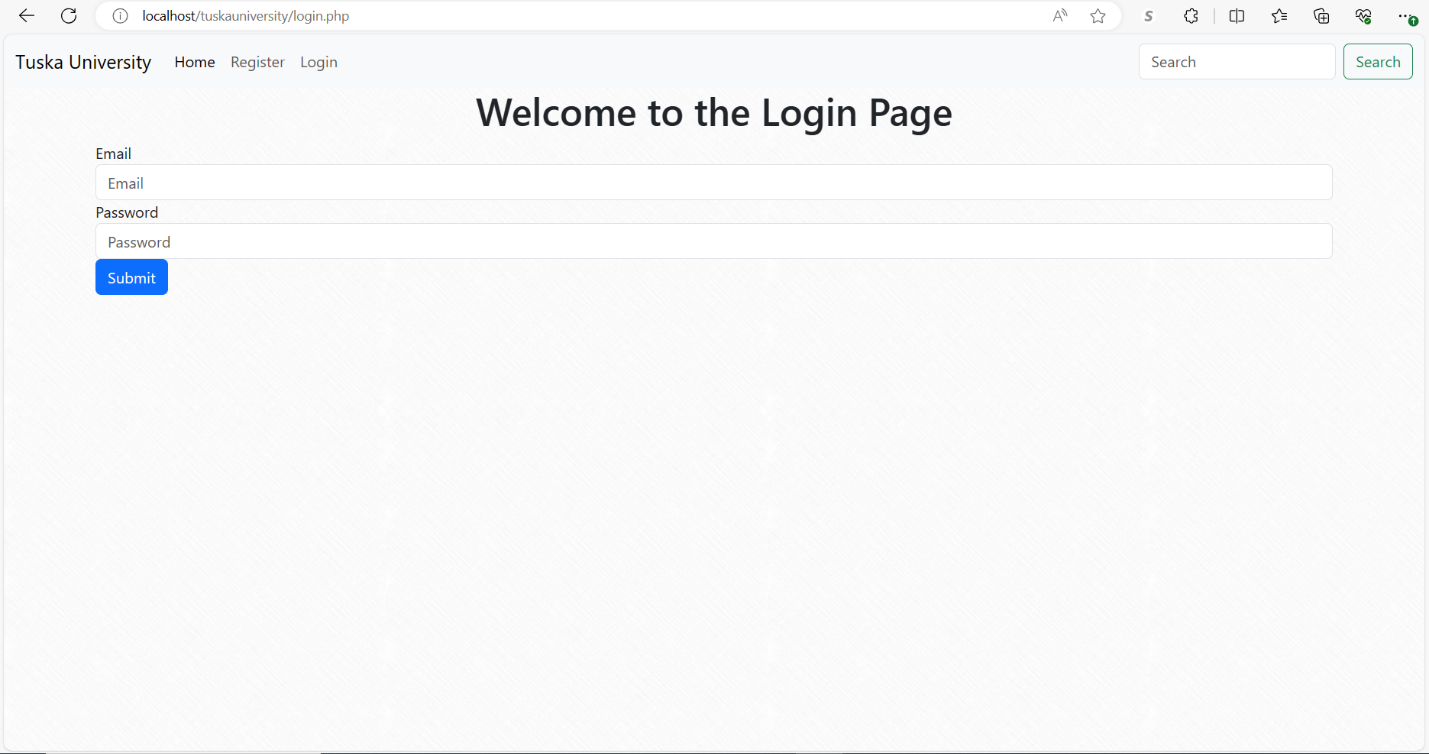
After creating the registration forms for the students and the administrative staff I needed to created a login form to allow students and administrative staff to log in to the system whenever they need to. In order to do this, I created a new file and named it login.php. In the beginning of the file, I initiated a session start and included my config.php file because that particular page will need to access the database in order to verify a user’s information. Then I scripted the webpage to check if a username key is in session. If it is not set the user is not registered. If the user is not registered after attempting to login they will be redirected back to the homepage. I have structured the administrator login page the exact same way.











Within this course I was able to understand that a student registration system has a substantial amount of importance to online universities. This paper has thoroughly explained the important steps involved in the development process, spanning technology setup, database design, and web page creation. The resulting system not only addresses the fundamental needs of student registration but also accommodates administrative staff, presenting a comprehensive and user-centric online experience. To complete this project, I used several online college websites to gain a better idea of the structure of an online university. I also used several books to aid me in creating the logic behind the front end and the back end of my website. Although I have created a functional website, I didn’t get to focus on the security and error handling aspect of my website. I understand the importance of error handling and security for software systems that handle sensitive information of users therefore, I will be sure to implement that into the next iteration of my software product in order to improve its functionalities.

References

DeBarros, A. (2022). Practical SQL: A beginner’s guide to storytelling with data. No Starch Press.

Duckett, John. (2014). HTML & CSS: Design and build websites. John Wiley and Sons.

Duckett, Jon, Ullman, C., & Stone, E. (2022). PHP & MySQL: Server-side web development. John Wiley & Sons.

Mikoluk, K. (2013, September 18). XAMPP tutorial: How to use XAMPP to run your own web serverLinks to an external site. Udemy.https://blog.udemy.com/xampp-tutorial.

SiteGround. (n.d.). phpMyAdmin create and populate tables tutorial. Links to an external site. 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/126521/modules/items/6439323) (4th ed.). Jones & Barlett Learning.