**Database Development and Class Registration**

Charlie Farris

University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Dr. Charmelia Butler

November 29, 2022

**Database Development and Class Registration**

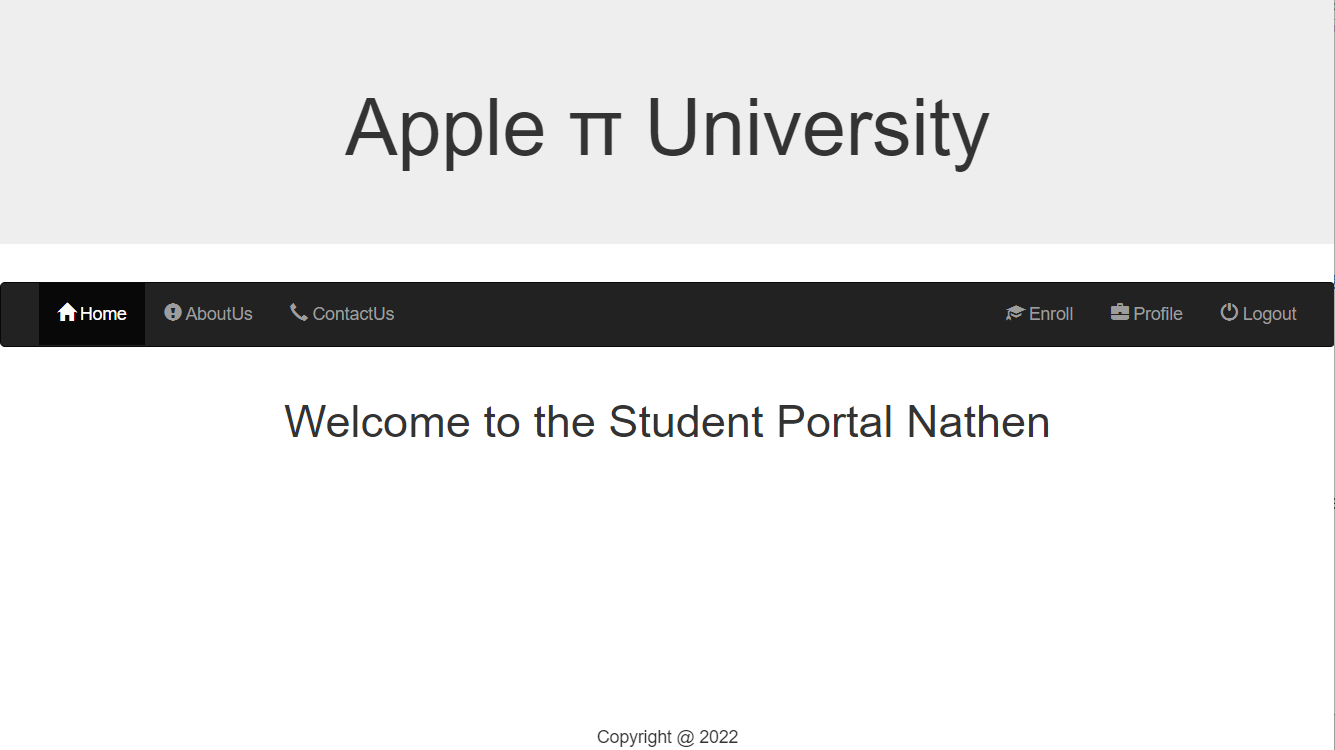
The purpose of the Student Portal Website is to provide a means for each student to register, login, maintain their information, and register for courses. In the future, additional functionality may be added to the Portal to provide additional information or access to other applications like course Transcript information, course materials, and grades.

**Steps Taken to Combine All Functionality**

Throughout this course, I built each leg of functionality upon what had previously been built rather than building each function separately and then combining them into a single application at the end. Hence, I added the new functionality to the previous week’s completed website. In week three, I took the provided website template, installed it into the instance of XAMPP already on my computer, and then tested the template to ensure it was working as expected. From there, I created the user registration page and a database to store the user registration information. The student login page was then added to the previous website code so that the Student Portal consisted of both the user registration and login functionalities. I completed the week by adding the student profile page allowing the logged-in user to see the information they entered during registration. The last thing I added in week three is the logout functionality to the Student Portal. This functionality allows users who have logged in to log out and secure their information. This was accomplished by clearing out the $\_SESSION array and the session cookie. I then used a PHP function called destroy\_session to clean up everything else.

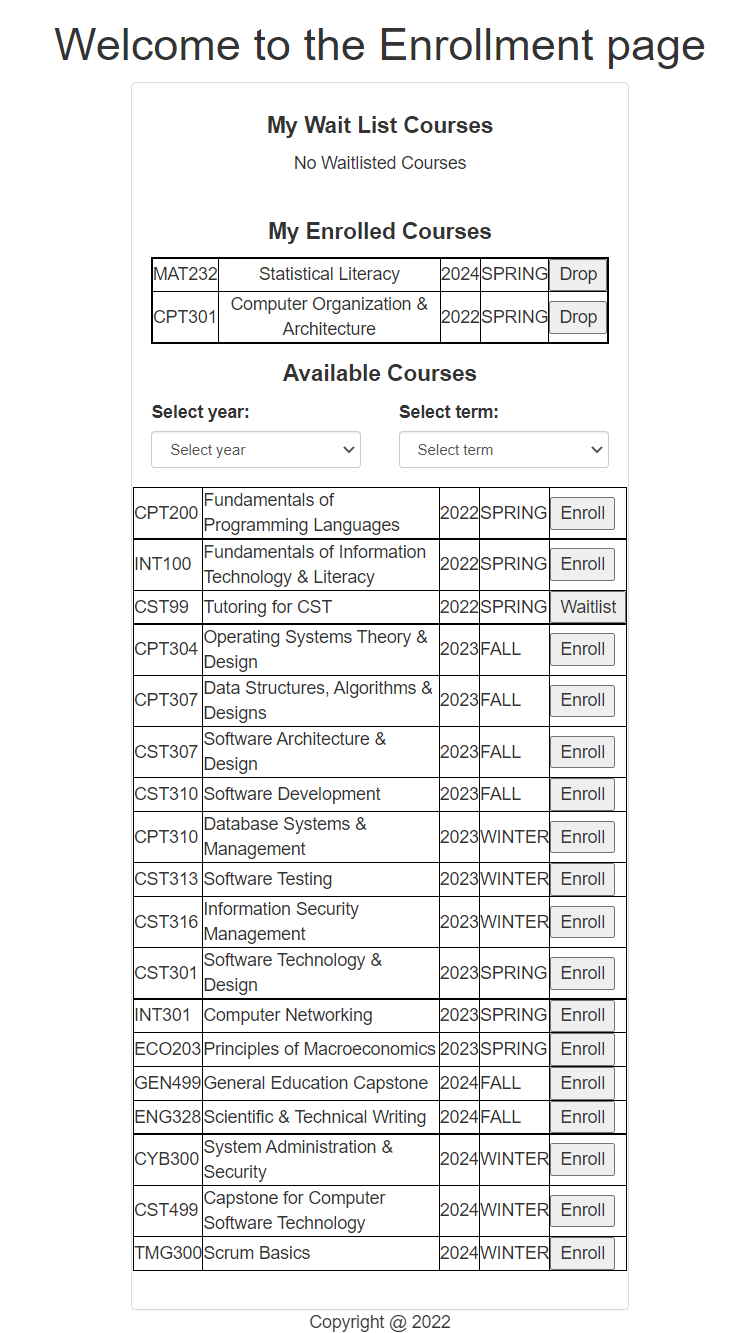
In week four, I built on the previously developed site adding functionality to allow the user to enroll in courses, drop courses, add themselves to a waitlist if a course is at max capacity, and remove themselves from a waitlist. In order to accommodate these features, I added several new pages and database tables.

**Figure 1** User home page

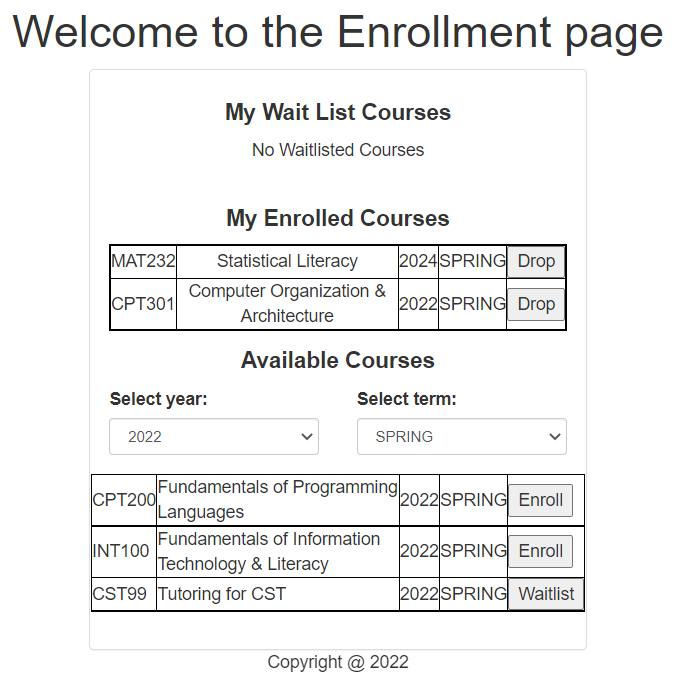


*Note.* The Student Portal website home page after the user has logged in. A welcome message is personalized with the user’s first name.

**Figure 2** Course enrollment page

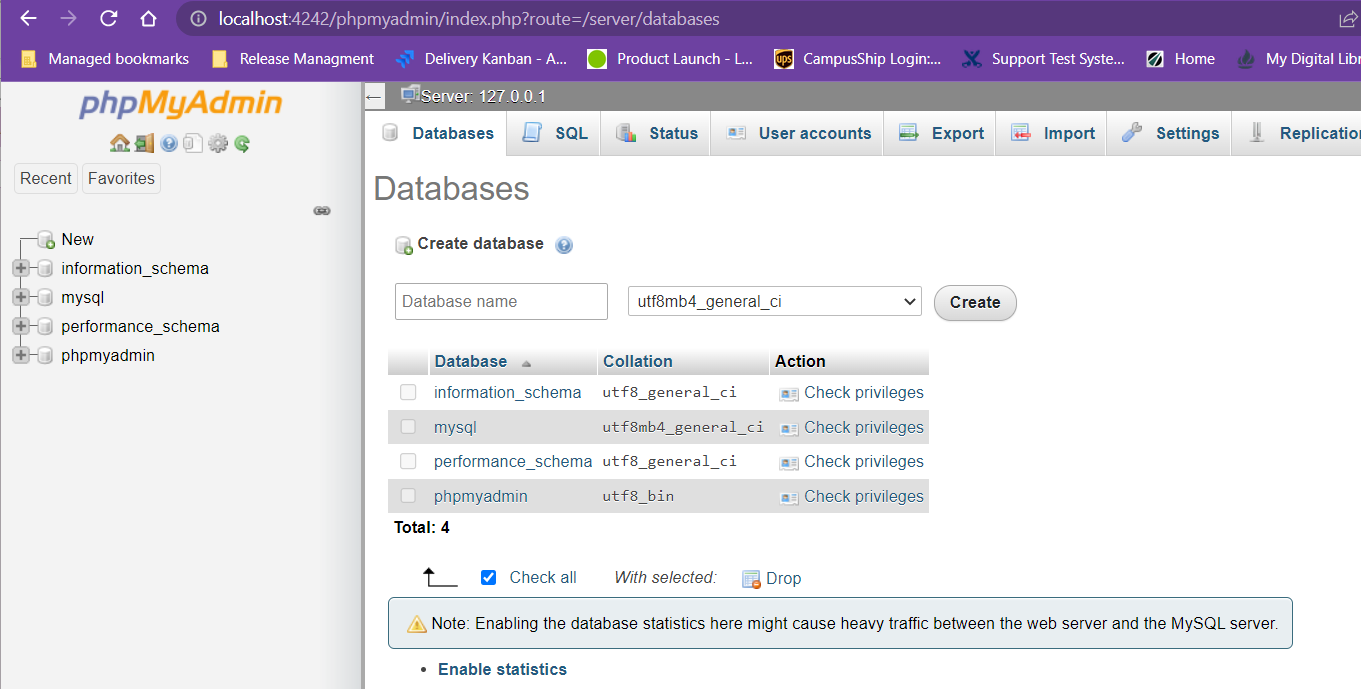


**Figure 3** Select Term and Year

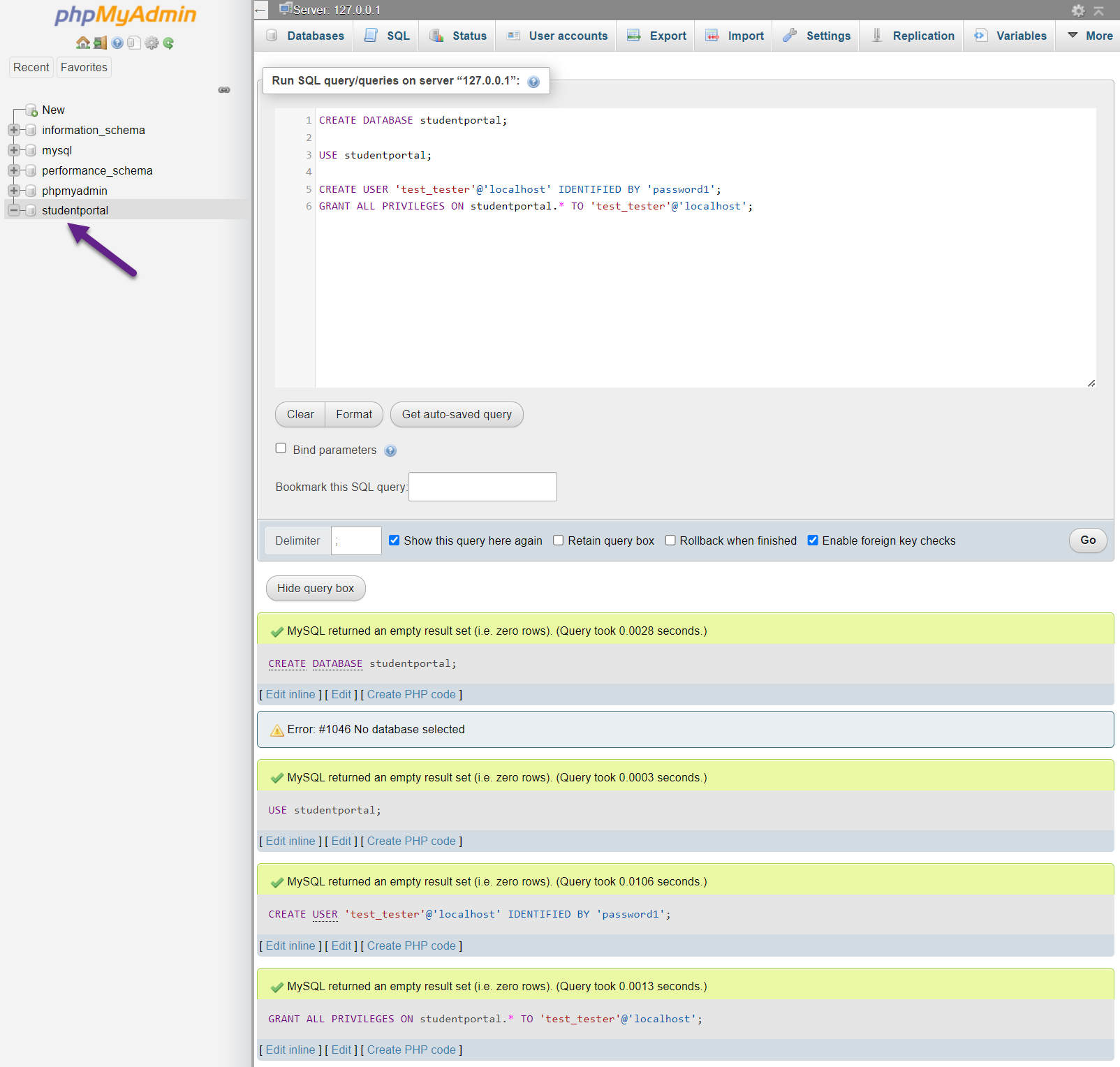


*Note:* The student can tell when a class is full as the ***Enroll*** button becomes a ***Waitlist*** button when the class has reached the maximum number of enrolled students allowed.

**Figure 4:** mySQL before creation of database.

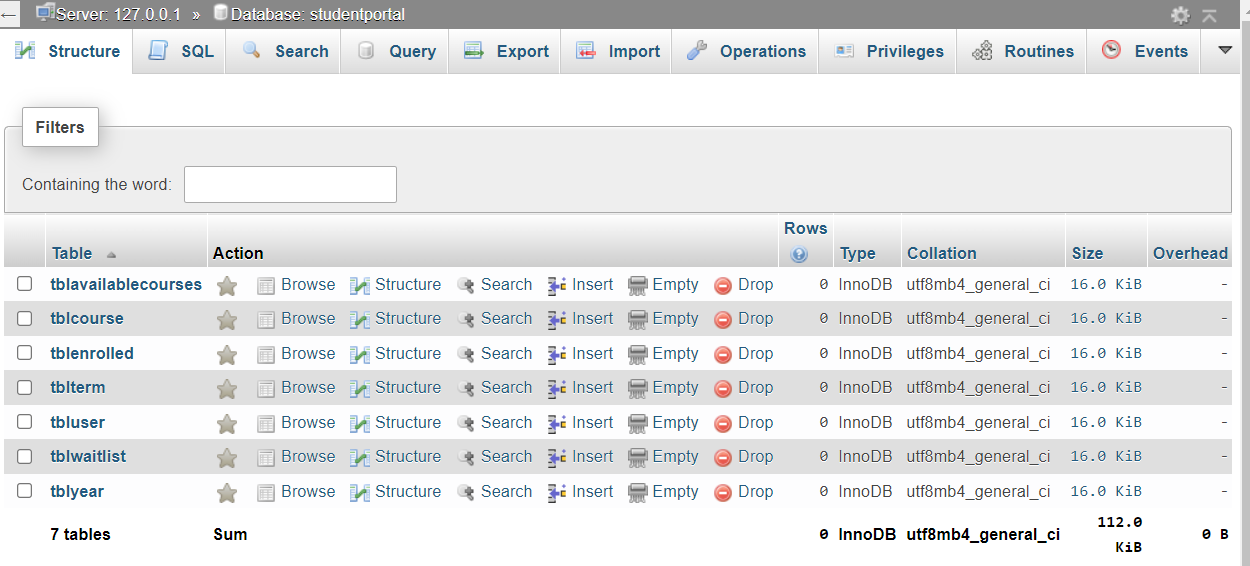


**Figure 5:** mySQL create database and user that is used by the php code to make the connection



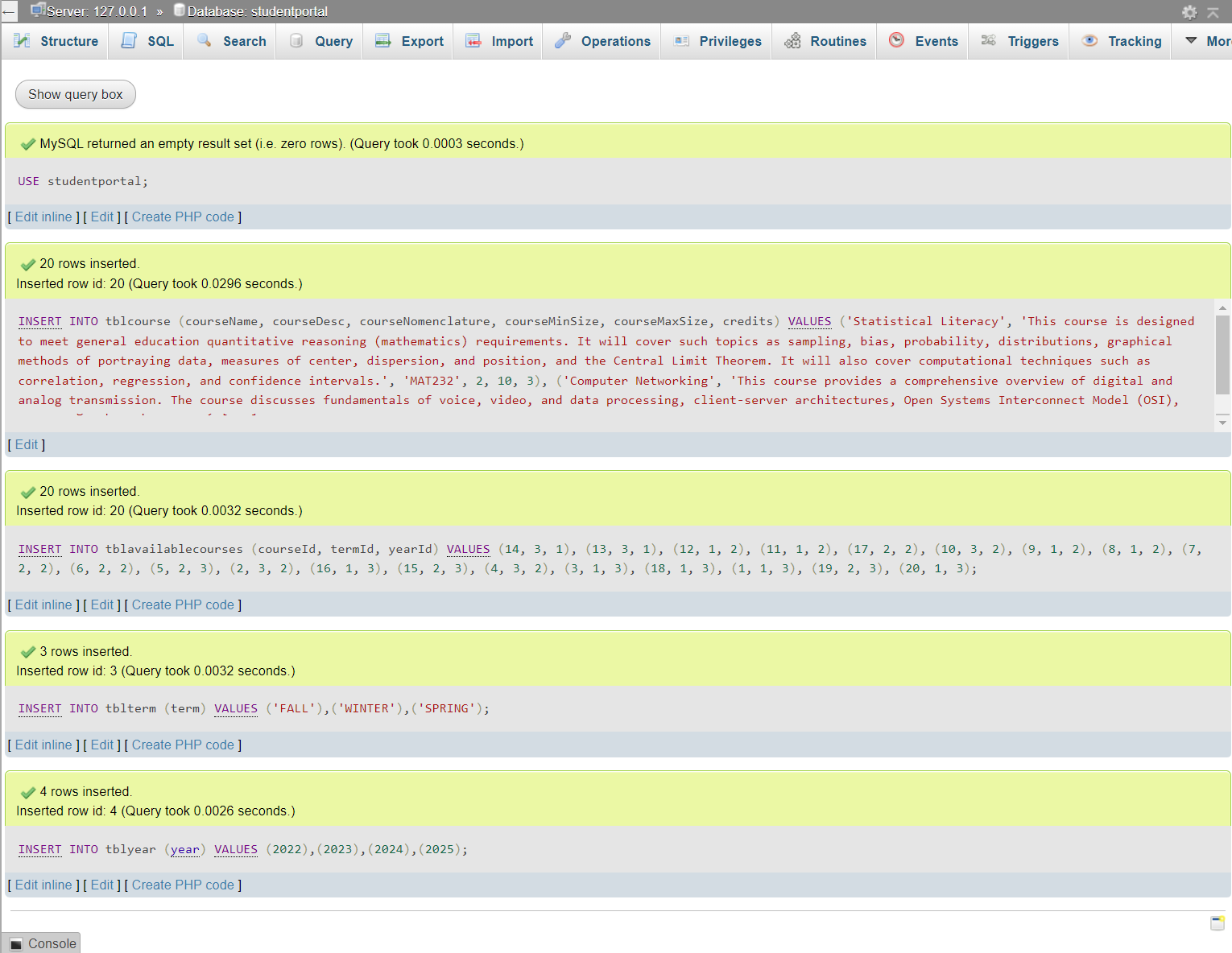
**Figure 6:** Create tables



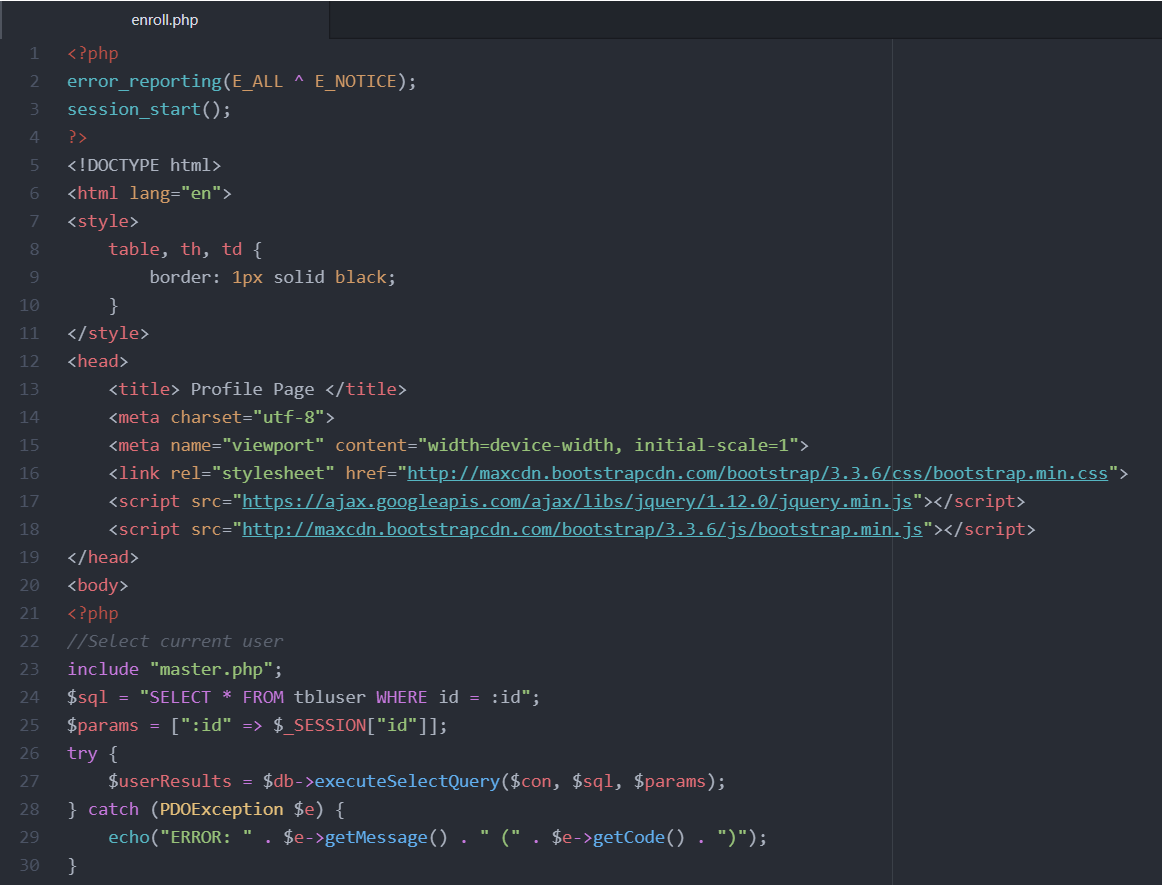


**Figure 7:** Populate tables



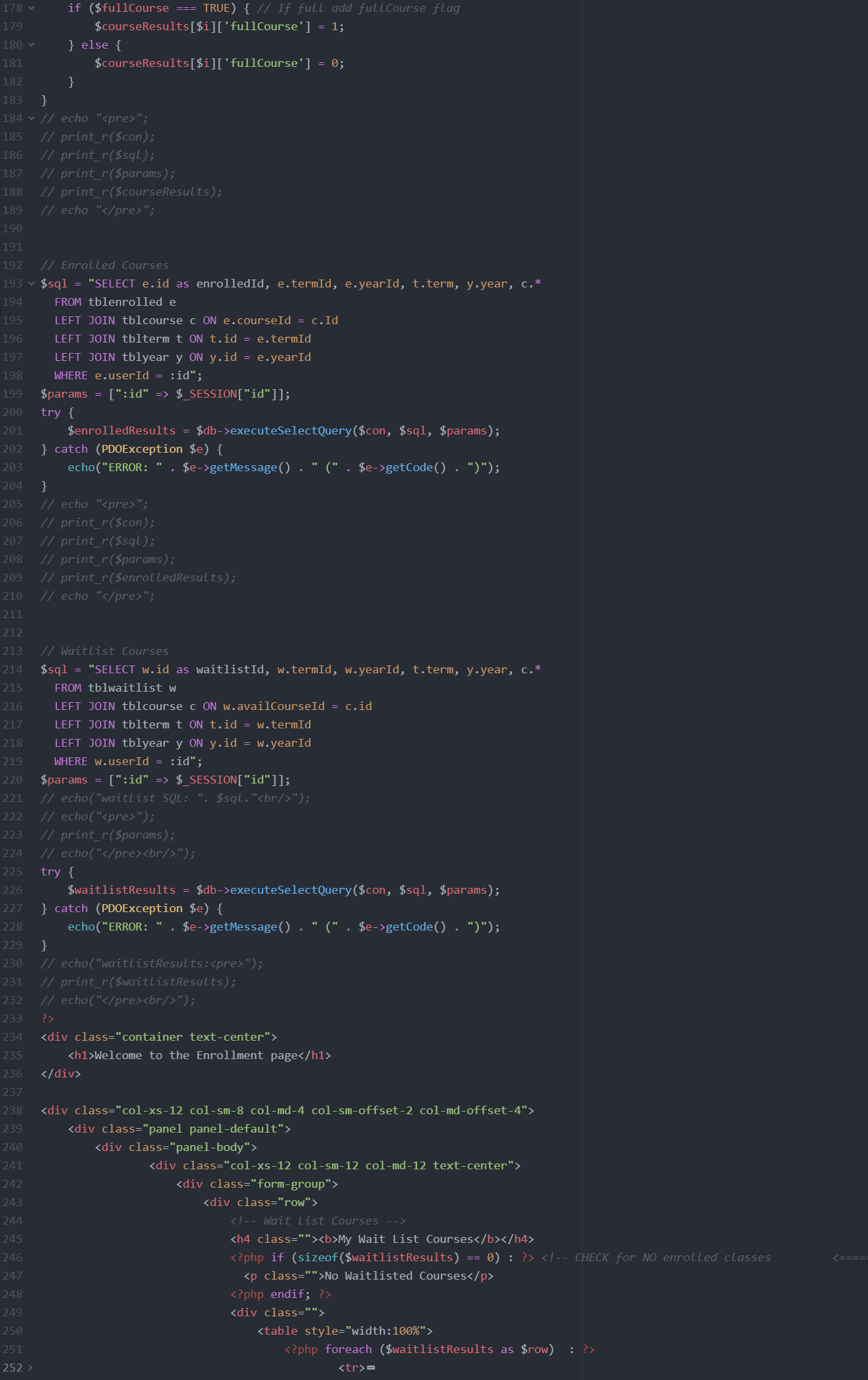


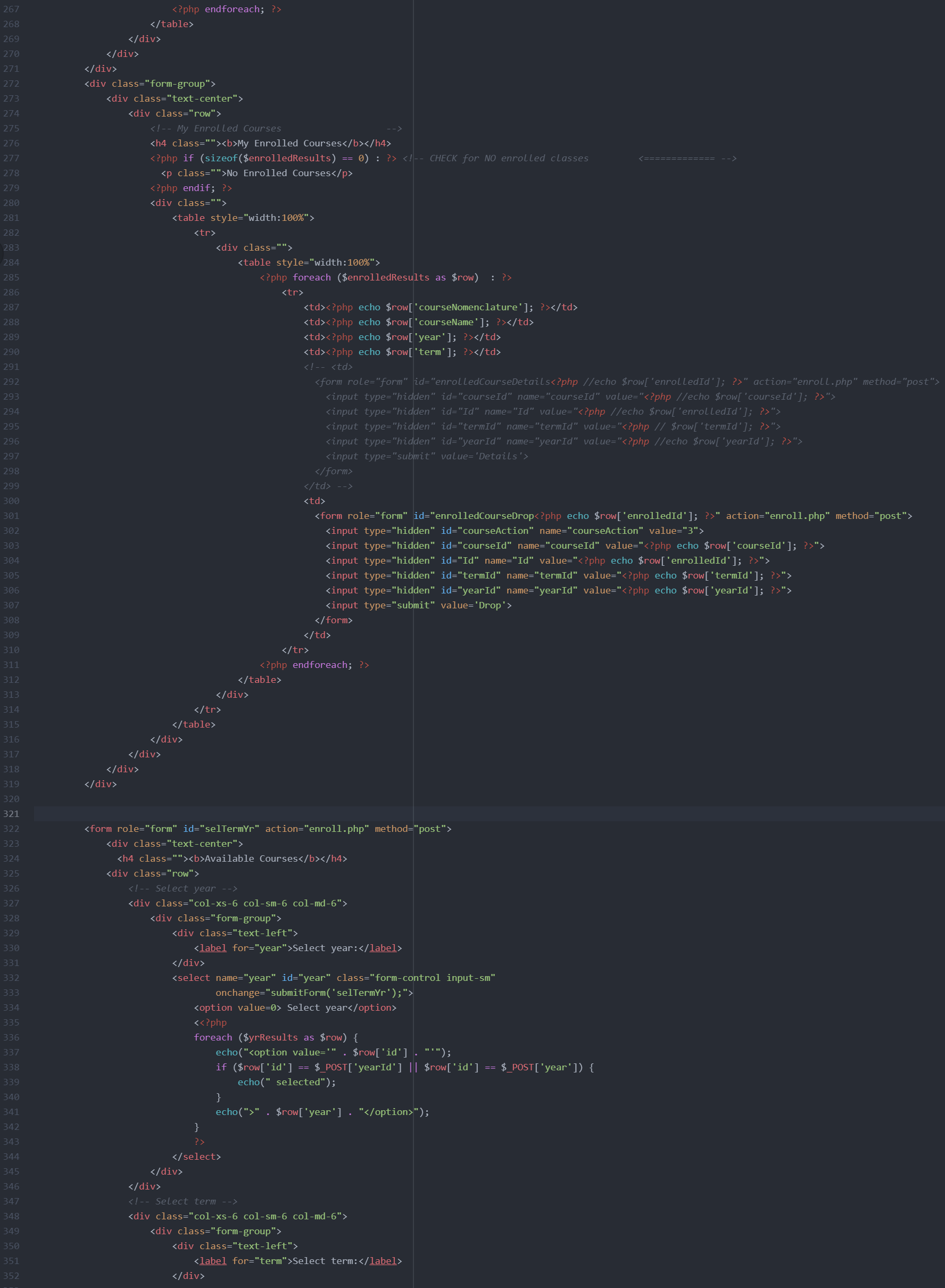
**Figure 8:** Enrollment.php

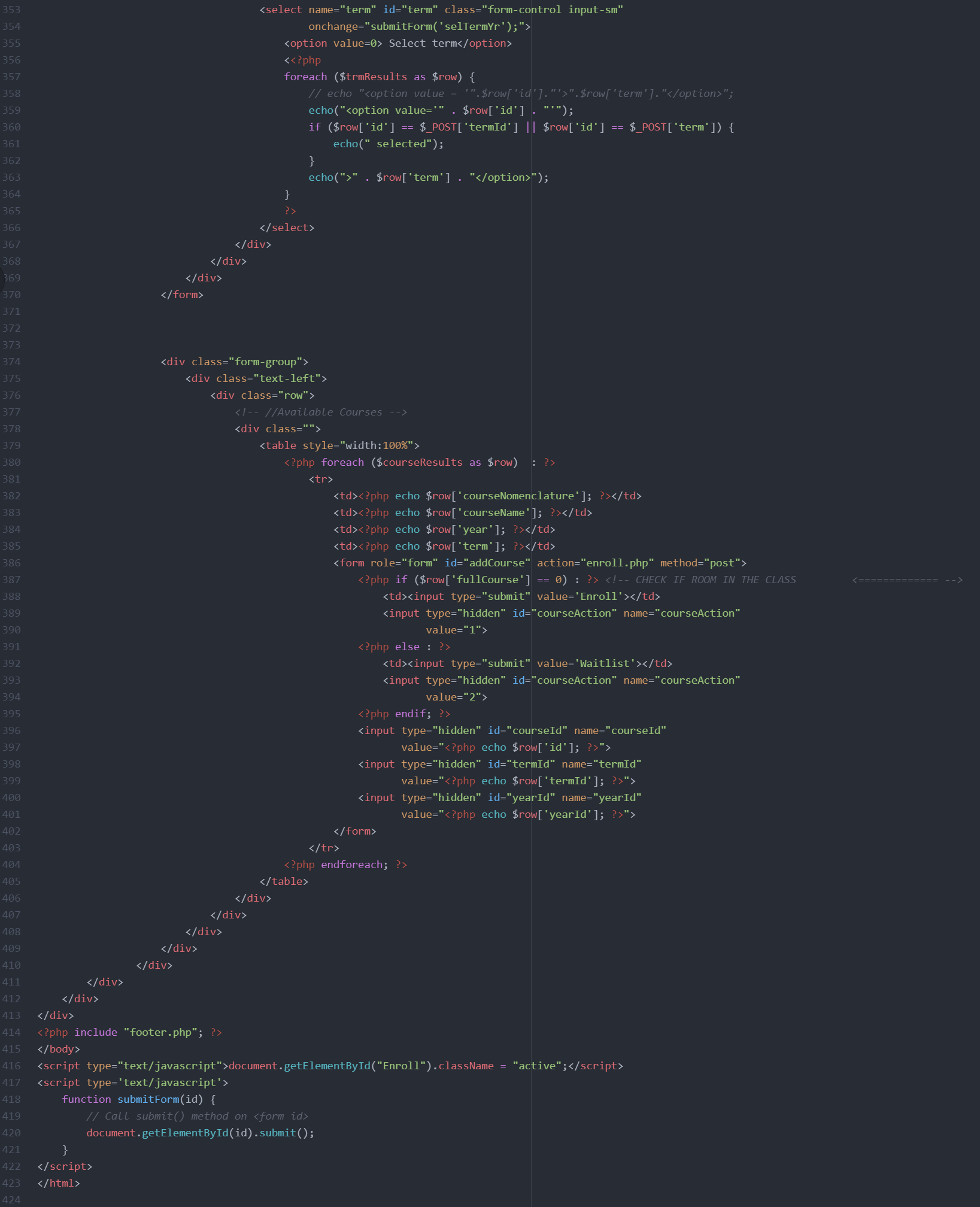












**References**

*Bootstrap · The world’s most popular mobile-first and responsive front-end framework. 3.3.6*

*Documentation—BootstrapDocs*. (n.d.). Retrieved from <https://bootstrapdocs.com/v3.3.6/docs/>.

*php.net – Documentation.* (n.d.). Retrieved from <https://www.php.net/docs.php>.

Php tutorials. (n.d.). *GeeksforGeeks*. Retrieved from

<https://www.geeksforgeeks.org/php-tutorials/>.

*Php tutorial*. (n.d.). Retrieved from <https://www.w3schools.com/php/default.asp>.