**Database Development for Registering and Cancelling Classes**

Jonah Hannett

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology (CSF2535A)

Joseph Rangitsch

9/22/2025

**Database Development for Registering and Cancelling Classes**

When building this functionality, I needed to implement 3 additional new tables in SQL so that I could store the professor information, the classes that could be scheduled, and a list of the classes that were scheduled by the students. I wanted to separate the professors and the classes so that I wouldn’t have to make separate classes depending on the professor. To bring these tables to life in the website with PHP, I needed to create 4 new php files cancelclass.php, classes.php, getclassinfo.php, and scheduleclasses.php. The classes.php and scheduleclasses.php files are webpages that the student will interact with. Classes.php holds the currently assigned classes and offers a cancellation button. It queries the SQL tables for class information and status (enrolled/waitlisted#) and puts it into a table with a form cancel button at the end. If you click the cancel button, it calls the code in the cancelclass.php file which looks at the status (enrolled/waitlisted#) and either removes them from enrolled and sets the next in line as enrolled, or if they were waitlisted it just moves the waitlist up. It does call from the function in the scheduleclasses.php file which I need to move over. The scheduleclasses.php file has the general functionality of selecting a class depending on the semester and day of the week. I had to implement javascript to dynamically update the page so I could add more than one class and enroll them at the same time. This makes it look cleaner, and I can add any number of classes even if some are set to enroll and others are going to be put on a waitlist. The pictures of the database and the code are below.

**Code and Database Screenshots**

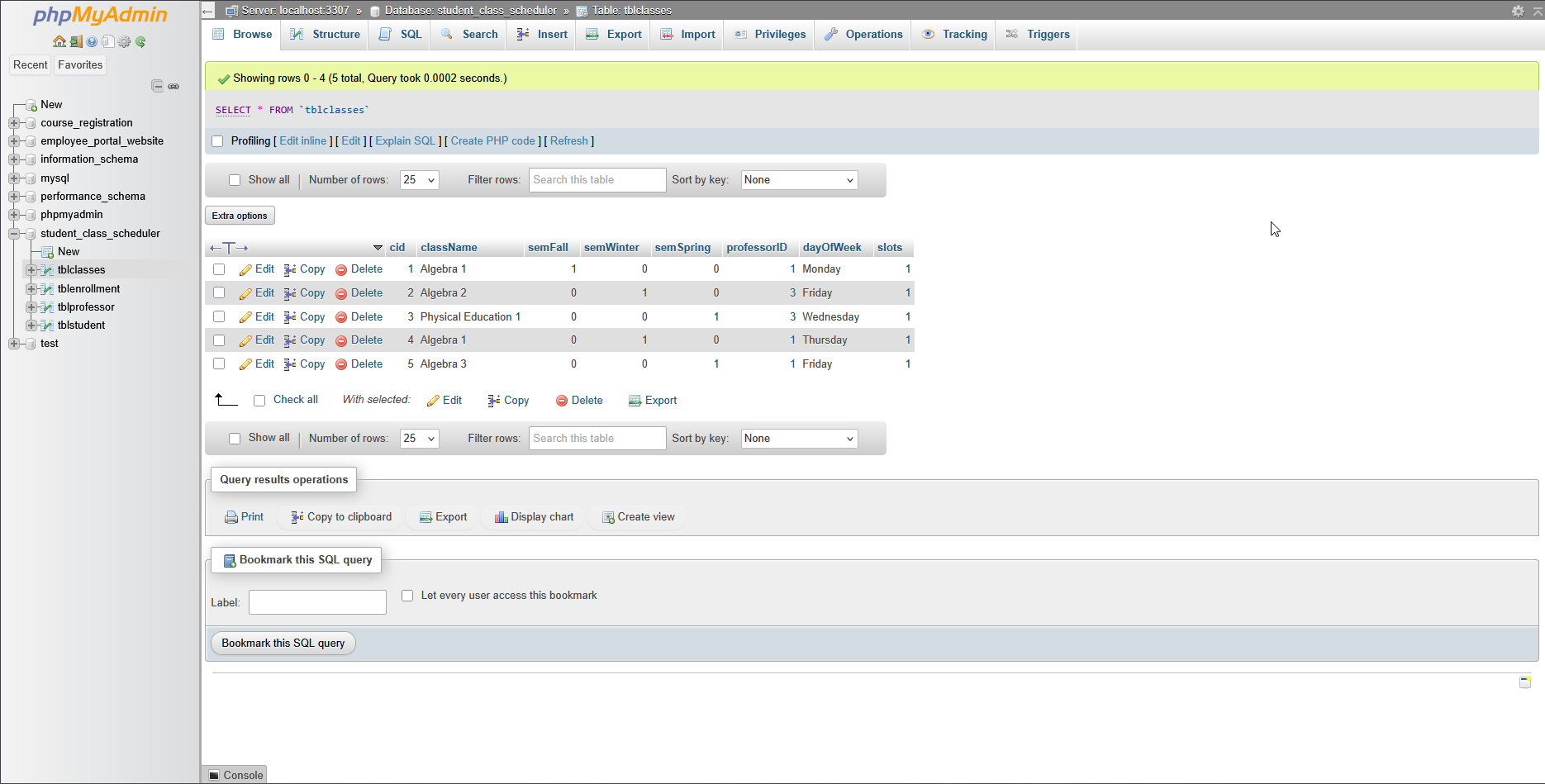


Figure .tblClasses. This table holds the class ID (AutoIncrementing) class name, which semester it is available for, professor assigned, day of the week, and how many class slots are available

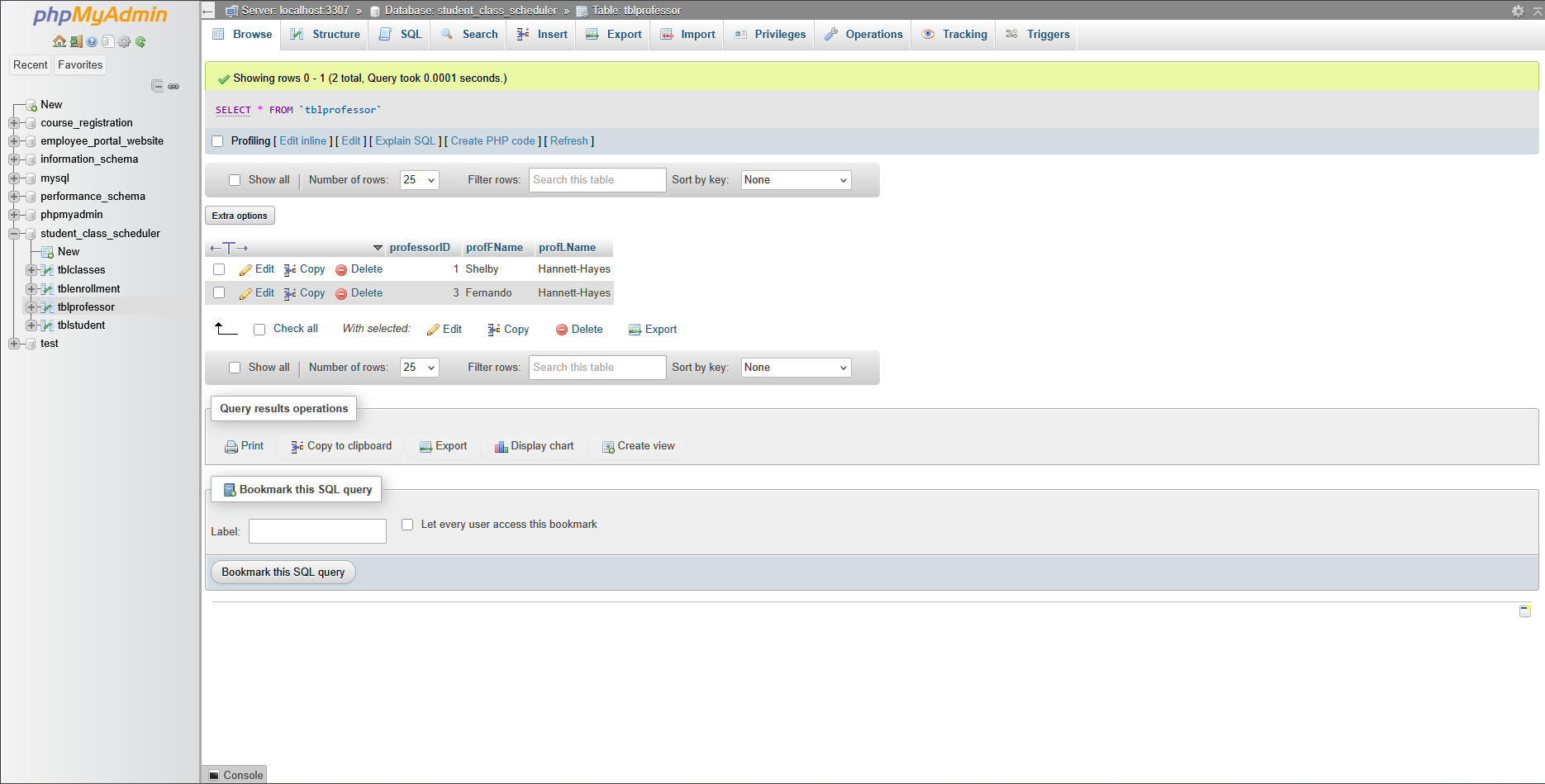


Figure .tblProfessor. Holds the professorID, and the professor first name and last name. Connected to tblEnrolled with a foreign key

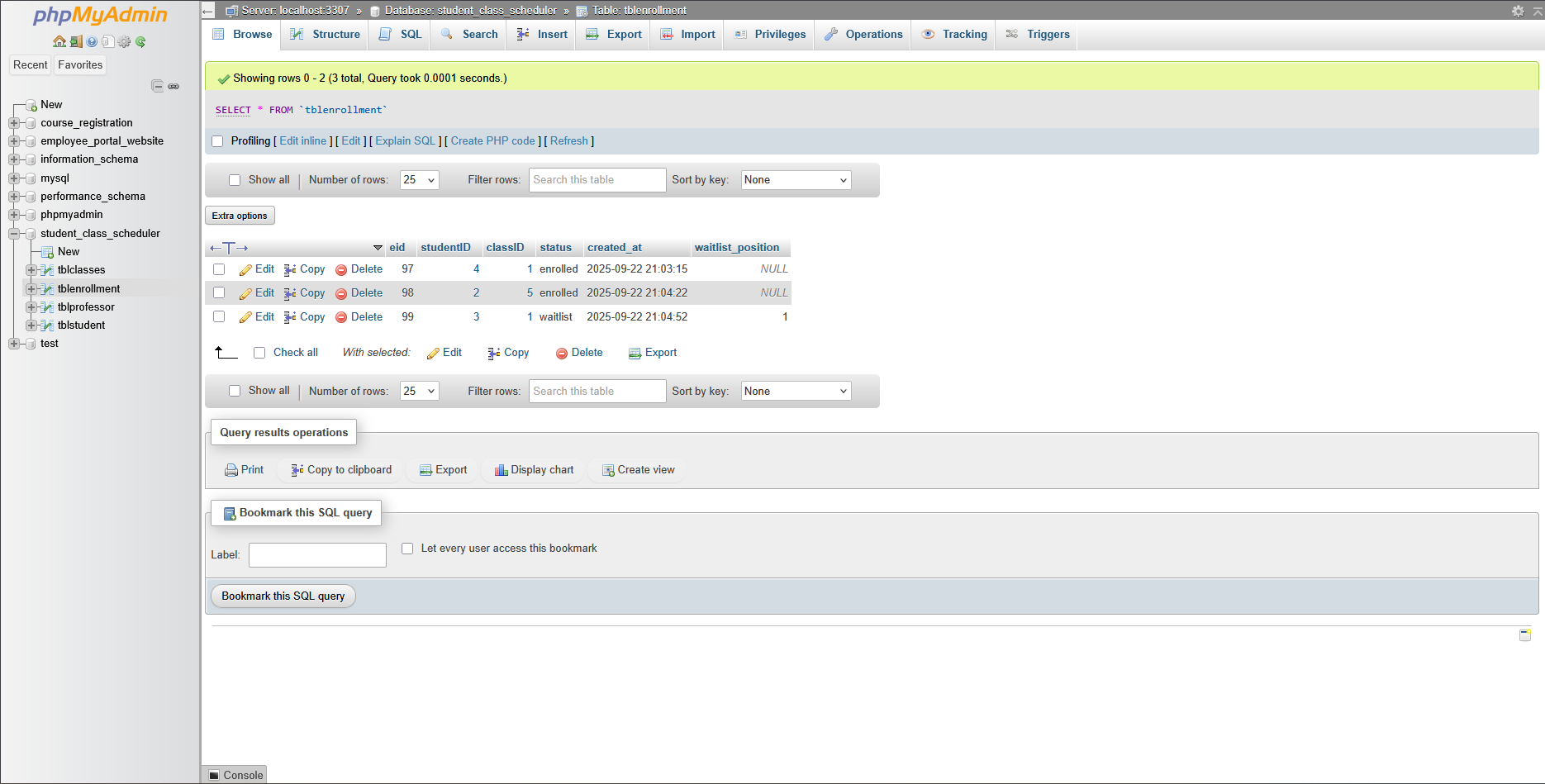


Figure . tblEnrolled before adding new classes. This table has the enrolled ID (eid) the student ID of the enrolled/waitlisted student, the classID, the status of the student (enrolled/waitlisted#), a timestamp of enrollment, and the position in the waitlist.

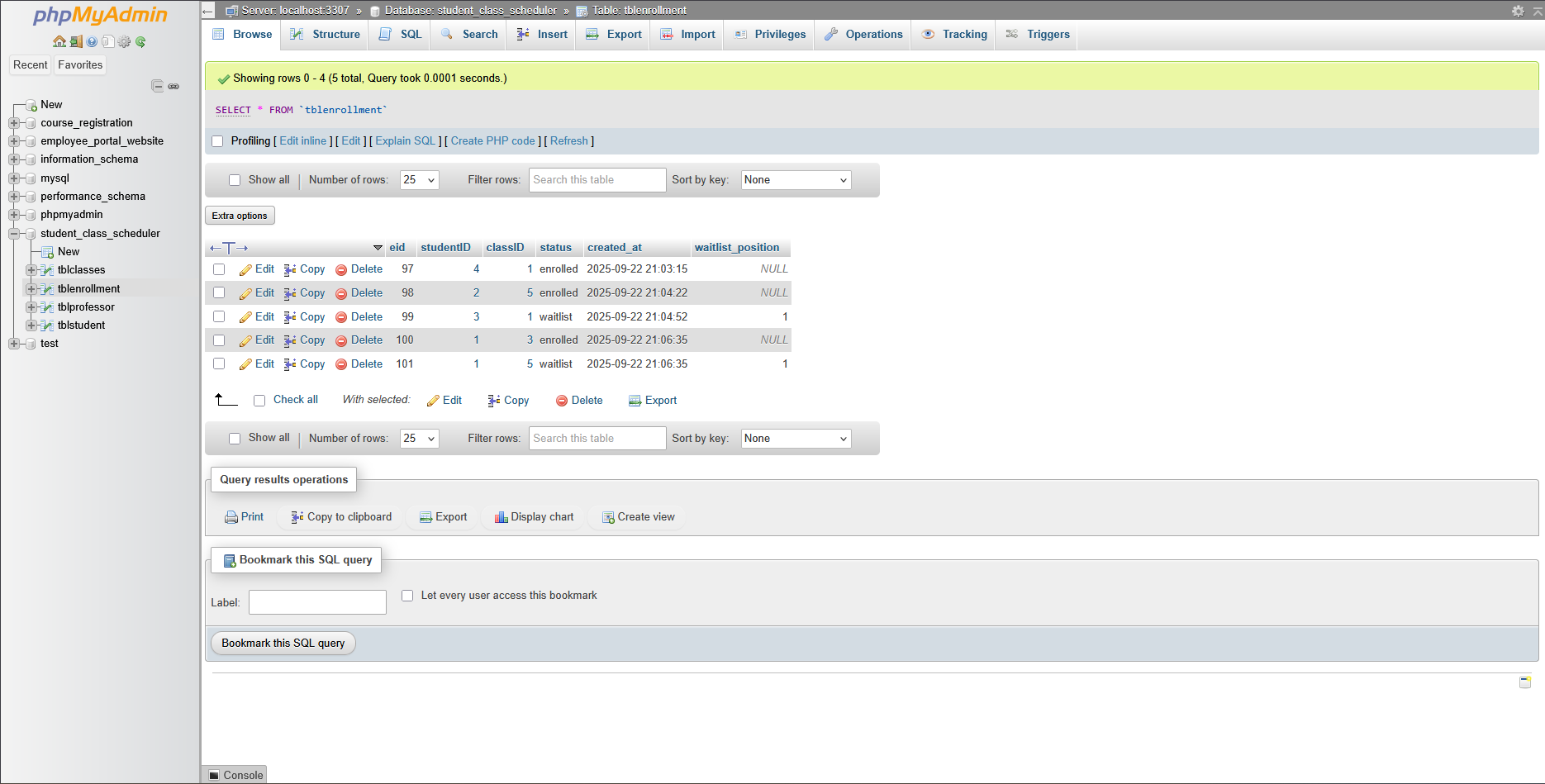


Figure . tblEnrolled after enrolling. The same tblEnrolled table but showing how student 1 enrolled to class 3 and 5 at the same time with one enrolled and the other waitlisted.

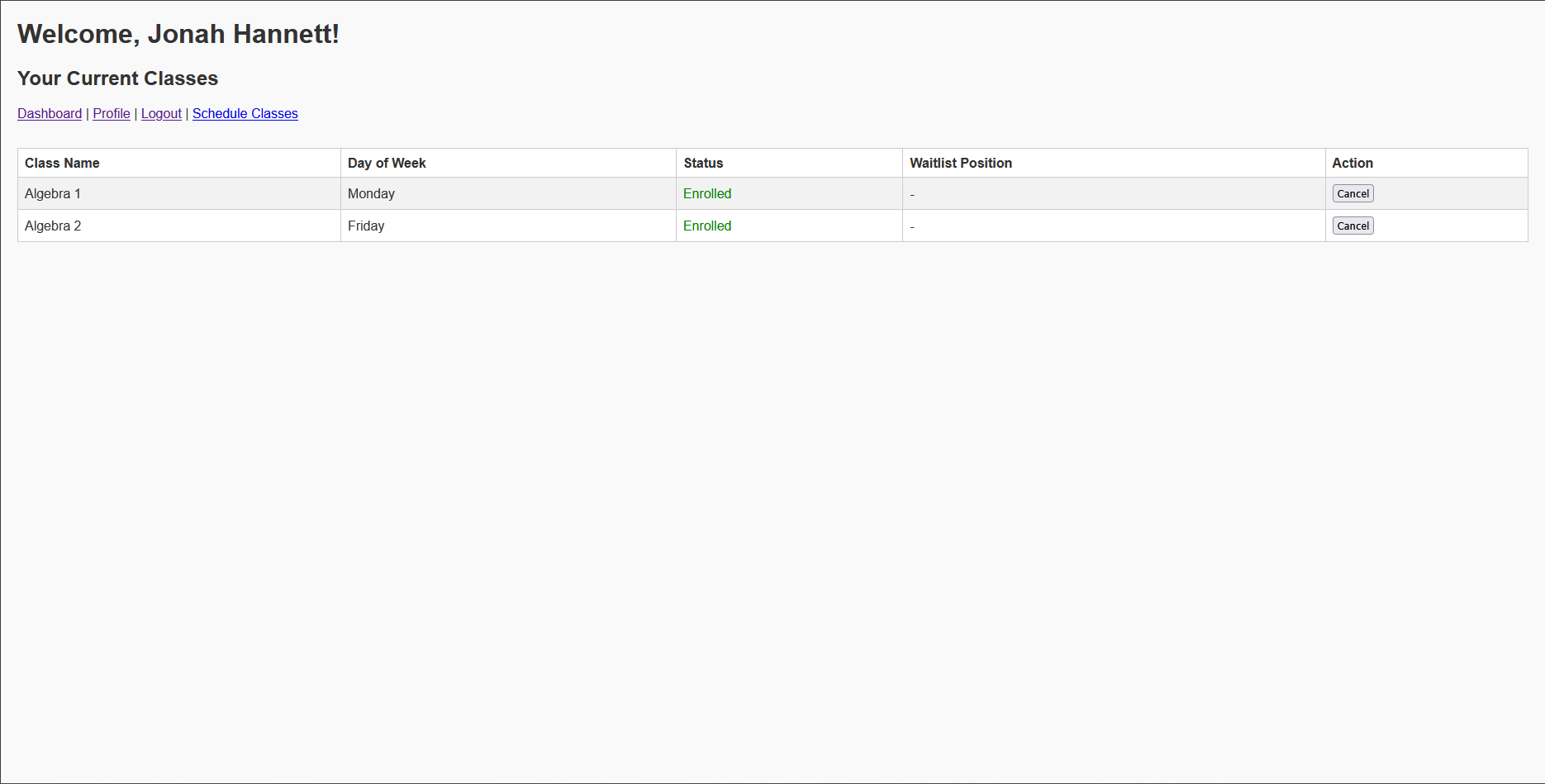


Figure . The classes.php page. This page is tied to classes.php and shows the current classes enrolled with the option to cancel.

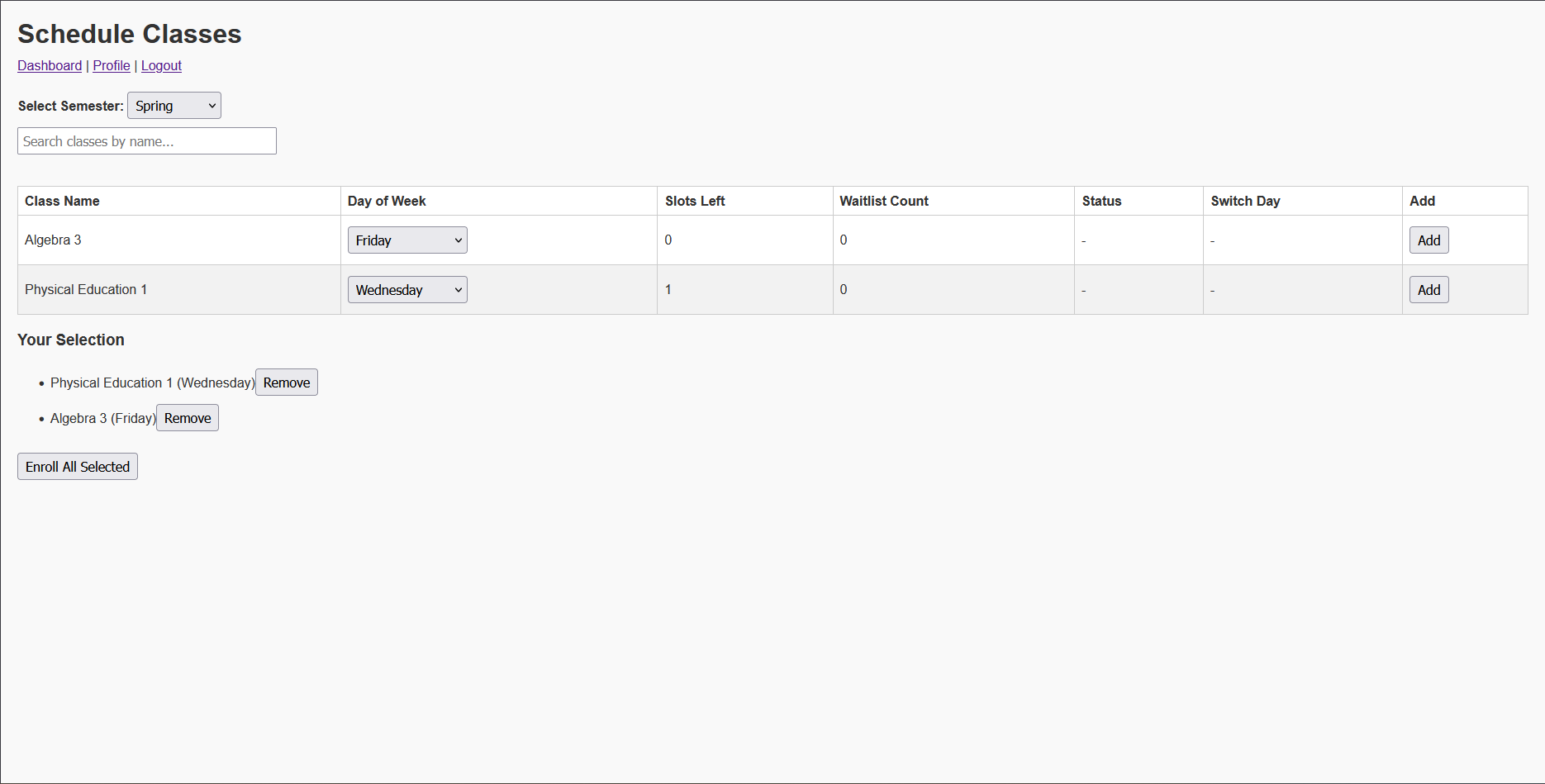


Figure . The scheduleclasses.php page. This page allows the student to select the semester then select classes from the available options.

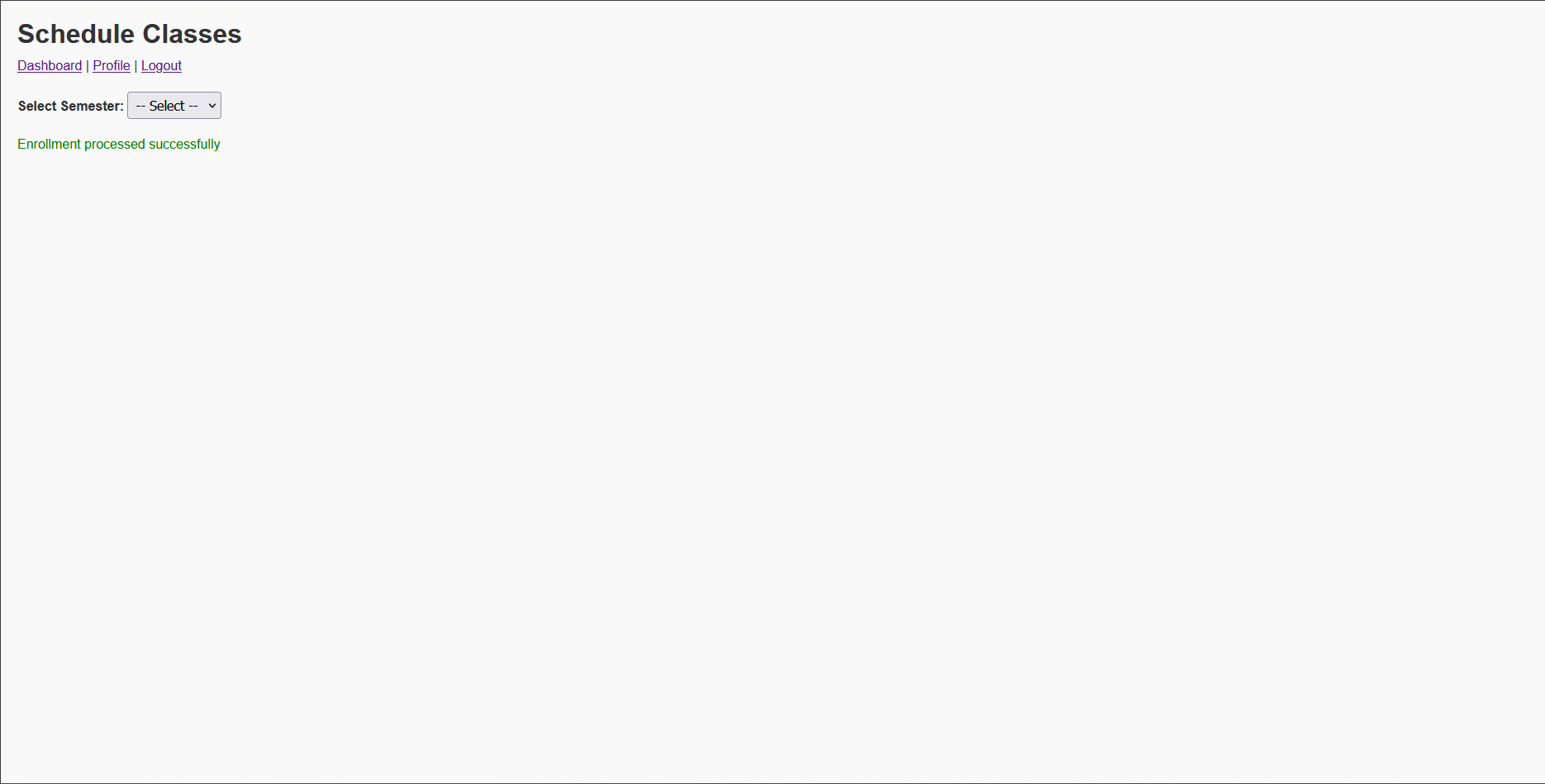


Figure . Scheduleclasses.php after enrolling. This is the message showing that the selected classes were scheduled. Figure 4 shows the result



Figure . Classes.php. This is the code for classes.php.



Figure . cancelclasses.php. This is the code to help cancel classes from classes.php



Figure . getclassinfo.php. This code mainly grabs enrollment/waitlist counts for javascript functionality

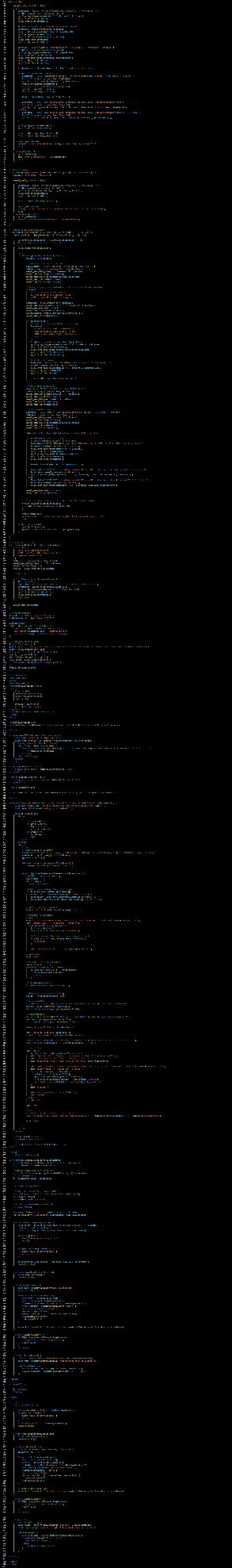


Figure . scheduleclasses.php. This is the code for the scheduleclasses.php page and will probably be refactored for the final submission.

**Conclusion**

In this paper we have covered the implementation of class scheduling and cancellation for students in the student scheduling website. I hope to continue to build a visually pleasing and functional system through additional research of CSS and refactoring.