**Week 3-5 – Database Development and Class Registration**

Ben Merritt

The University of Arizona Global Campus

CST 449: Capstone for Computer Software Technology

Instructor: Dr. Charmelia Butler

December 11, 2023

**Week 3-5 – Database Development and Class Registration**

This week, I revised the work from week 3, developing the Course Enrollment System for CST499. I created and updated the following tables: contact\_messages, courses, enrollment, messages, offerings, roles, rooms, semesters, students, subject\_tracks, teachers, and users. I researched many sample codes on GitHub, which helped reuse code to build this project. The main issue with reusing code is that it may still have to be troubleshot to ensure it will fit in the project. To complete this assignment, the overall process was to create database tables within PHPmyadmin, develop php files as webpages, and connect the database and webpages to provide inputs (Tsui, 2018). In the XAMPP, I created a folder 'CST\_499\_Week\_4\_Bmerritt\_Assignment\_Course\_Registration\_System' folder within htdocs to serve as the repository for all the new PHP files created for the course registration system. I coded in Microsoft Visual Studio Code and Notepad++ (Mikoluk, 2013).

**MySQL Functions and Steps to Create Database**

Before creating the database, I opened PhpMyAdmin, created a new SQL document in Notepad++ named CST499\_Week\_4.sql, and Imported it into phpMyAdmin (SiteGround, n.d.). The imported document provided the MySQL code, where a database named **CST499\_ben** is created; this unique name will be utilized later when establishing the connection between the PHP files and the database. The primary table, **users**, is defined with columns such as user name, password, and first and last name. An Admin user with the highest role is inserted into this table. I used the registration page to add a user, which defaults to the applicant's role. The applicant can log in but can't do anything until an administrator logs in and changes the user's role to student or Teacher. Once there is at least one teacher and student, the teacher can create classes, and the student can enroll or be put on a waiting list. When another student unenrolls from a class, the highest-level student on the waiting list is auto-enrolled in the class. These components contribute to a well-organized database schema that supports user management, access control, and message storage functionalities (SiteGround, n.d.). Additional testing will ensure data is getting in and out of the database for Course enrollment information.

**Developing the PHP Files**

This was one of the most challenging portions of the project this week; however, I could fortunately reference Github course registration projects. I required several files to be created to design the webpage and handle the inputs and outputs. I had used $pdo for my connect and found no projects that used it. $pdo will work with many database server platforms. I wrote some MySQL connection code and used both to see if it made any difference. As long as variables are used for the database connections, there is not much difference between MySQL and $pdo (Krossing, 2015). Once data is inputted into the text boxes on the webpage and the submit button is pressed, the registration.php file conducts several checks with Javascript and PHP to verify the information is in the correct format, that user information is not already in the database, and tells the verified text where to be placed into the table (Connolly & Hoar 2018). Bootstrap was used to set up the page's format and HTML code (Oh, n.d.). Included is the admin panel to change user roles and other information, which allows access to additional pages based on role name.

The free and open-source PHP script security model is based on standard cryptology models and provides high protection for both source code and opcode levels. We can write secure PHP files by providing a theoretical framework and a practical implementation of a robust and efficient encryption system for PHP scripts. (Jevremovic et al., 2013)

**Conclusion**

In conclusion, this week's development efforts focused on creating the database tables for the Course Enrollment System for CST499. The process involved using a local server, XAMPP, creating multiple tables in a database through PHPMyAdmin, and developing PHP files to establish the necessary web pages. The steps to run a PHP file in XAMPP were outlined, emphasizing the importance of the XAMPP control panel and the 'htdocs' folder for file storage and access. MySQL functions played a crucial role in database management, with detailed steps provided for creating the database and tables. Developing PHP files, including classes.php, adminpan.php, courses.php, messages.php, student.php, teacher.php, and schedule.php, presented challenges, but leveraging past course materials facilitated the process. Noteworthy components include the profile.php file, which serves as the foundation for the roles of the different users. Bootstrap was utilized for page formatting and HTML code structure. Overall, the project laid the groundwork for a comprehensive Course Enrollment System, encompassing user management, access control, and message storage functionalities within a well-structured database schema.

**Screenshots**

Below are screenshots of the database and table, the various pages, and the PHP files.

**Figure 1.** Database tables

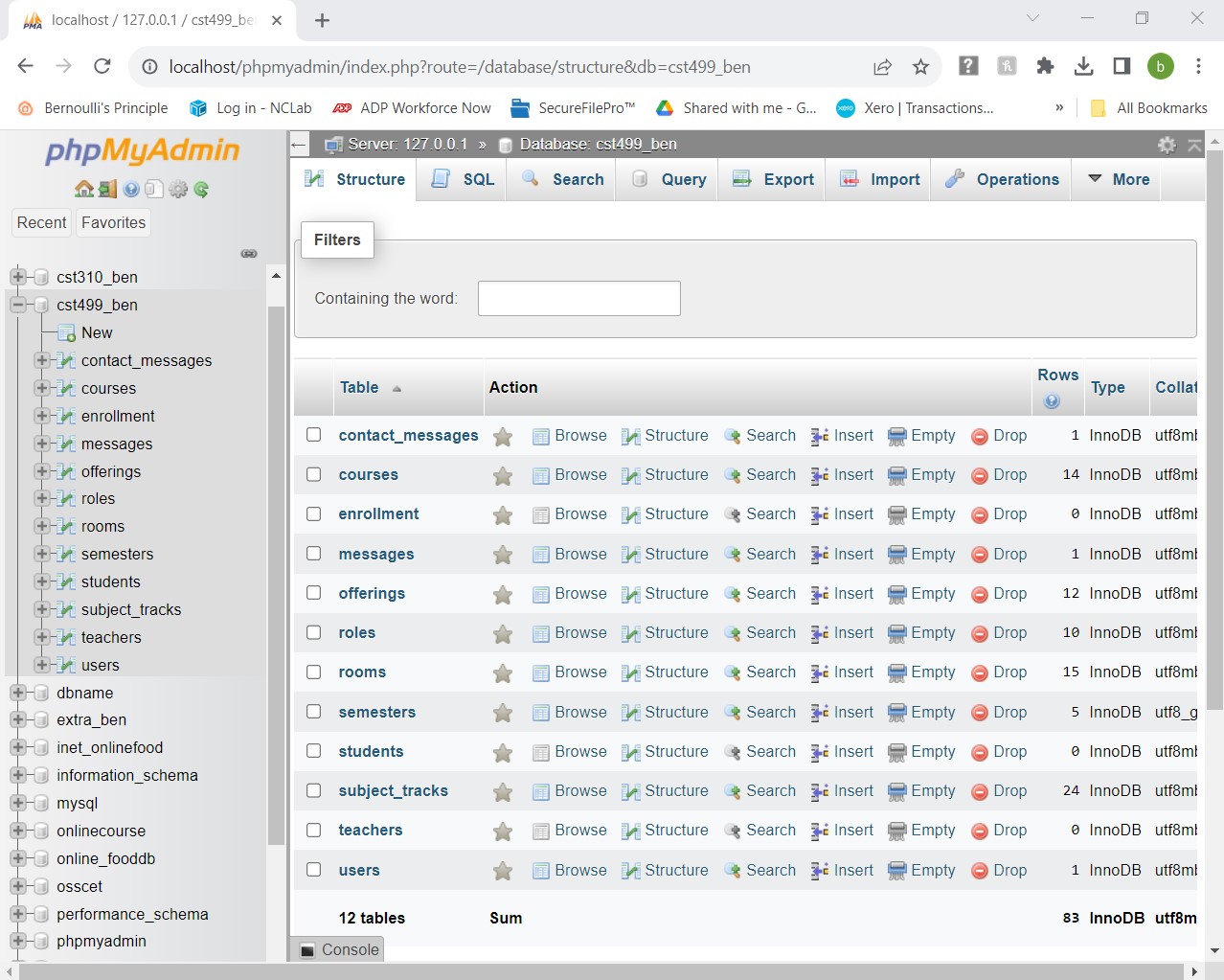


Figure . SQL code for cst499\_ben Database click to expand embedded document.



**Figure 3**. Contact Us Page.

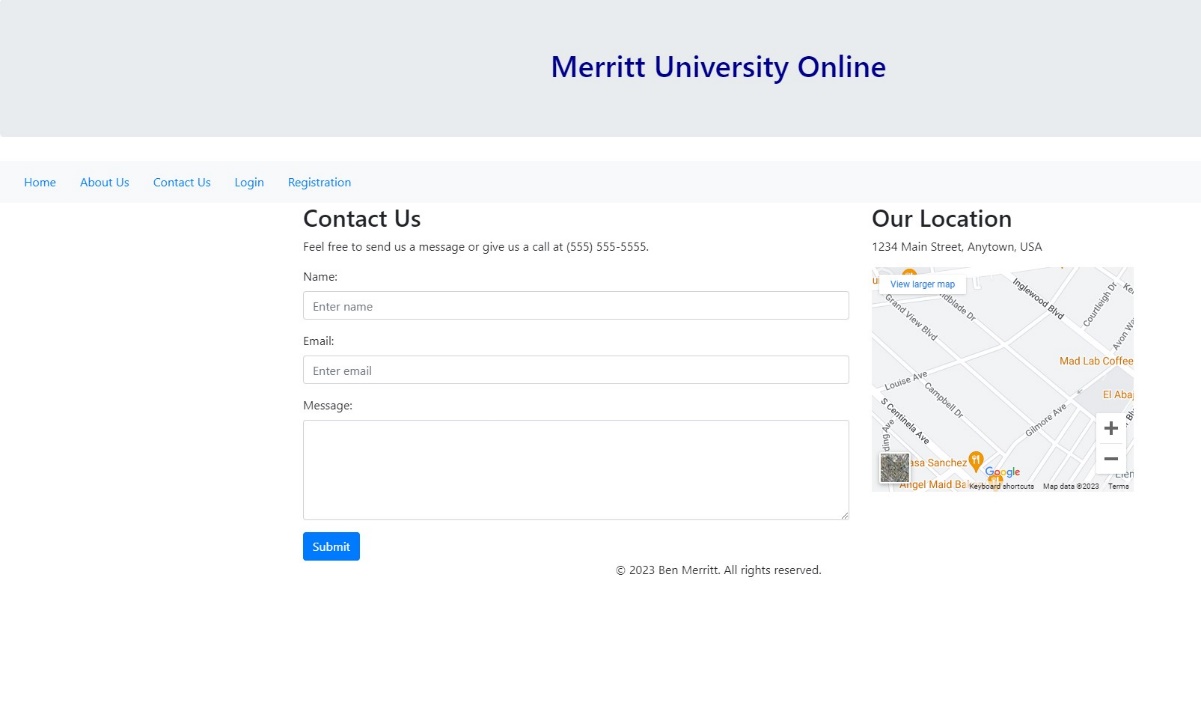
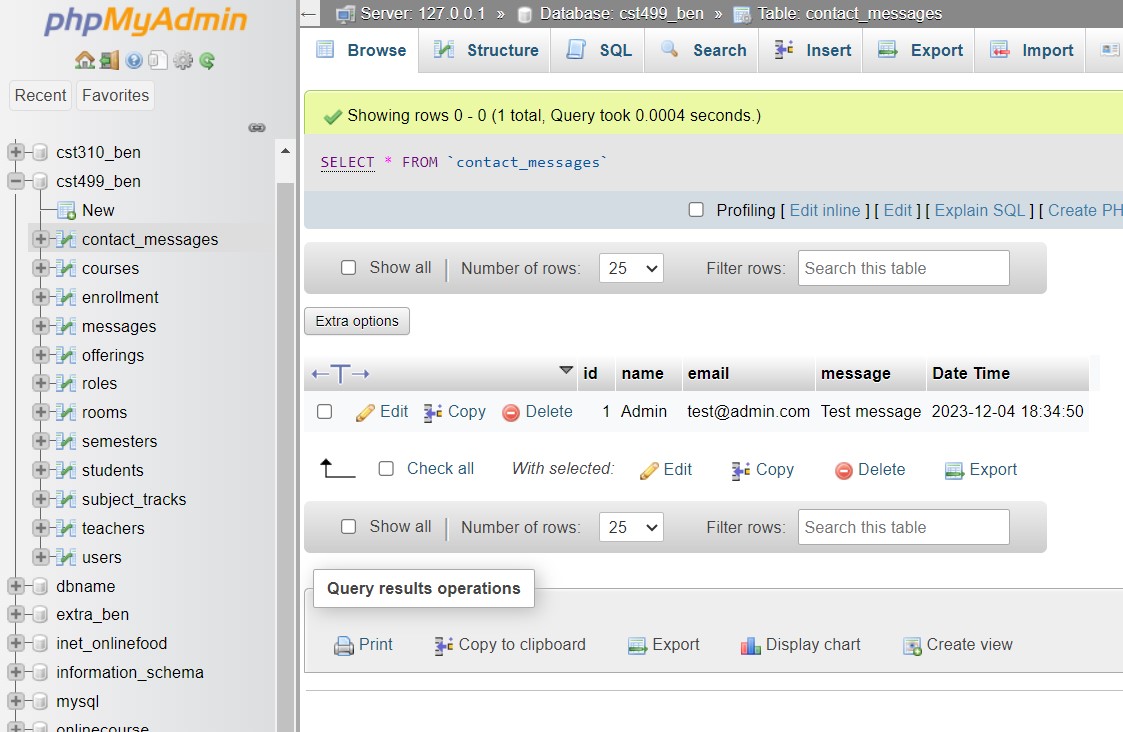


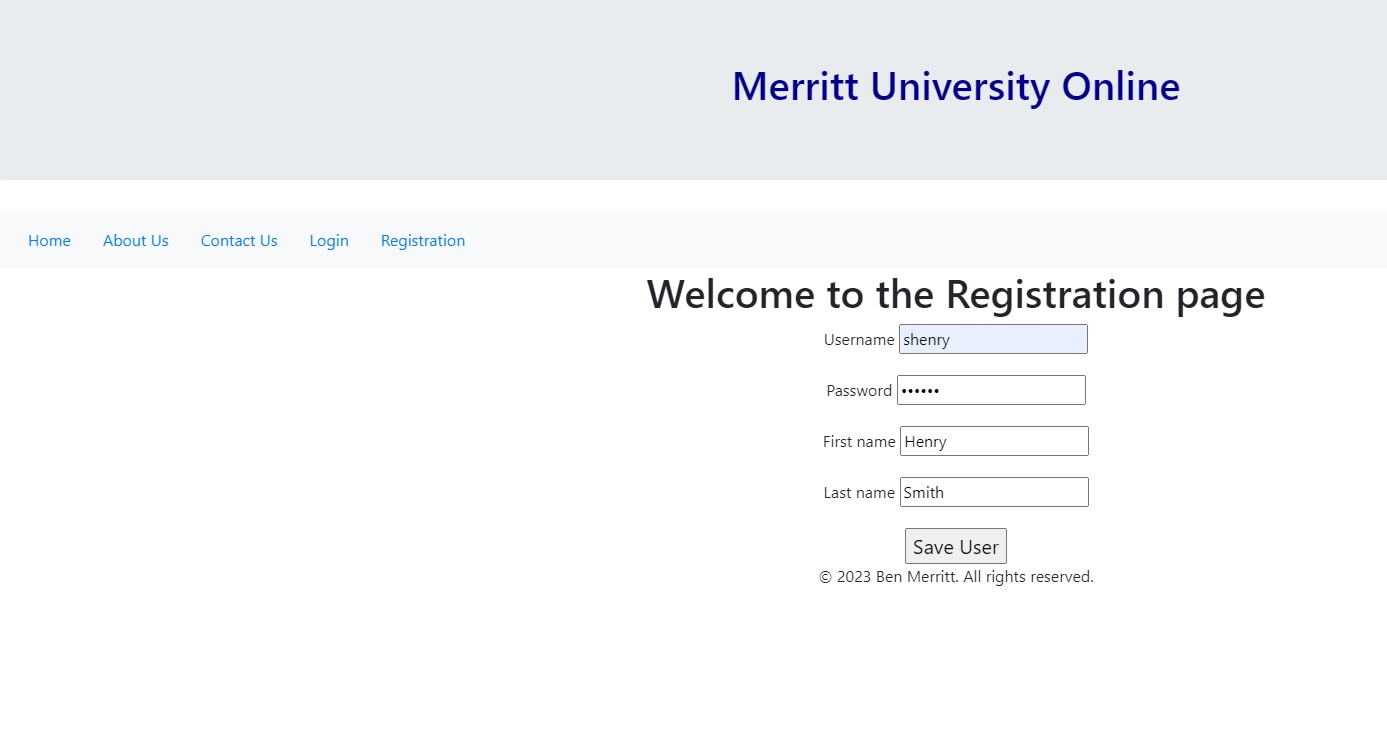
Figure . Contact Us code click to expand embedded document



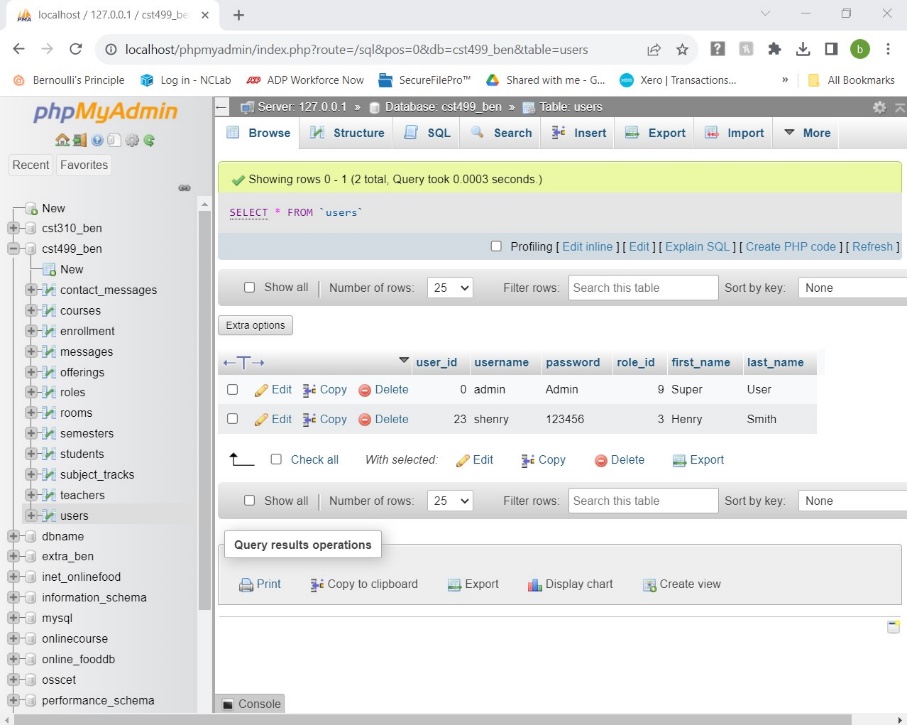
**Figure 5.** Contact Us Page contact\_messages Table.



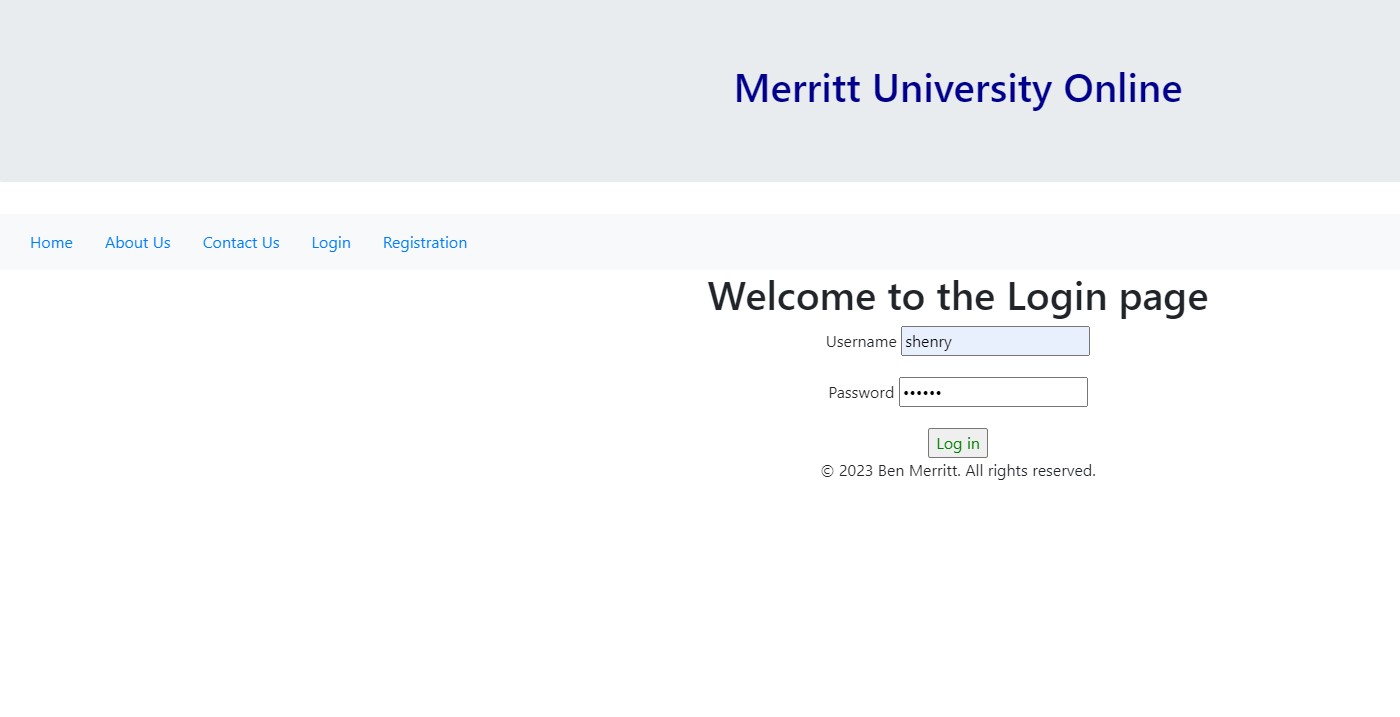
**Figure 6**. Input Entered in Registration Page.



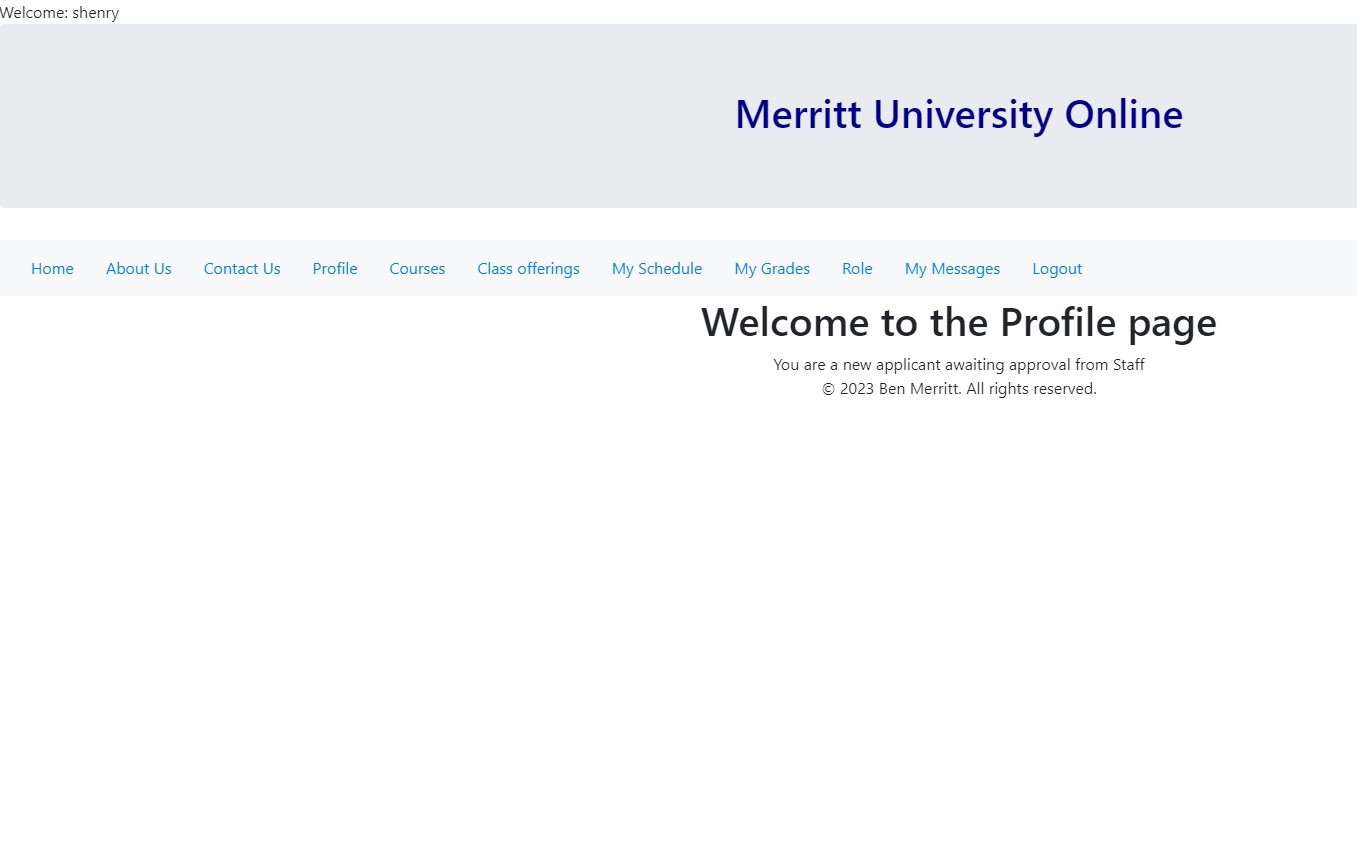
**Figure 7.** users Table new user roll applicant number 3.



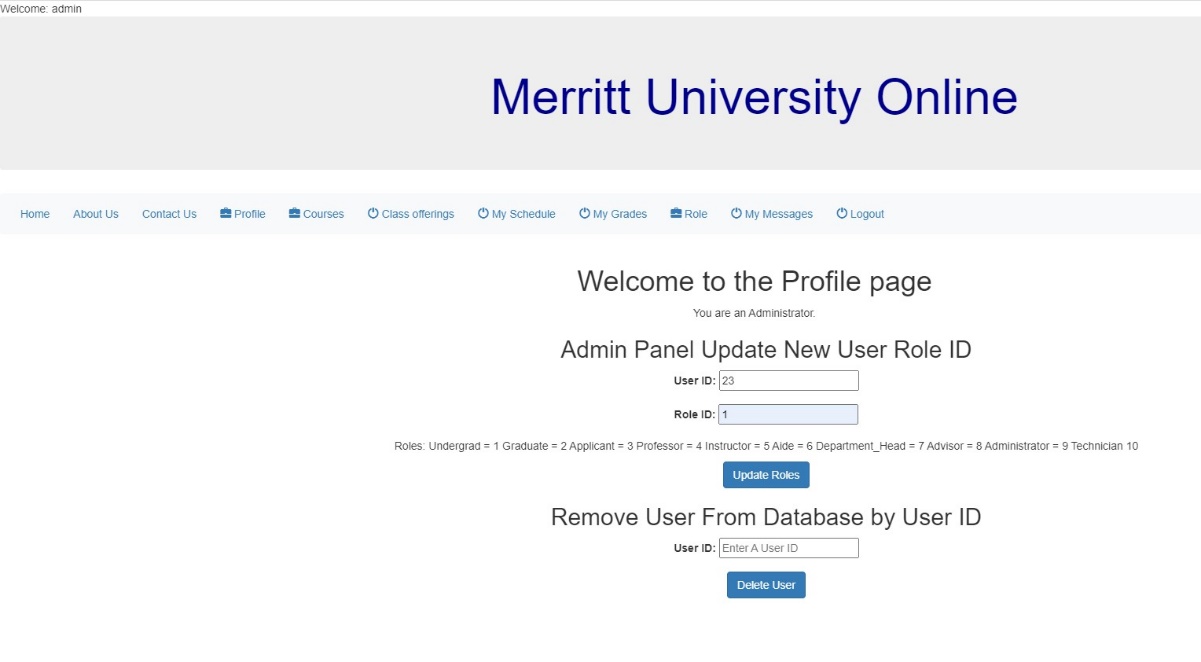
**Figure 8.** Login new user Applicant



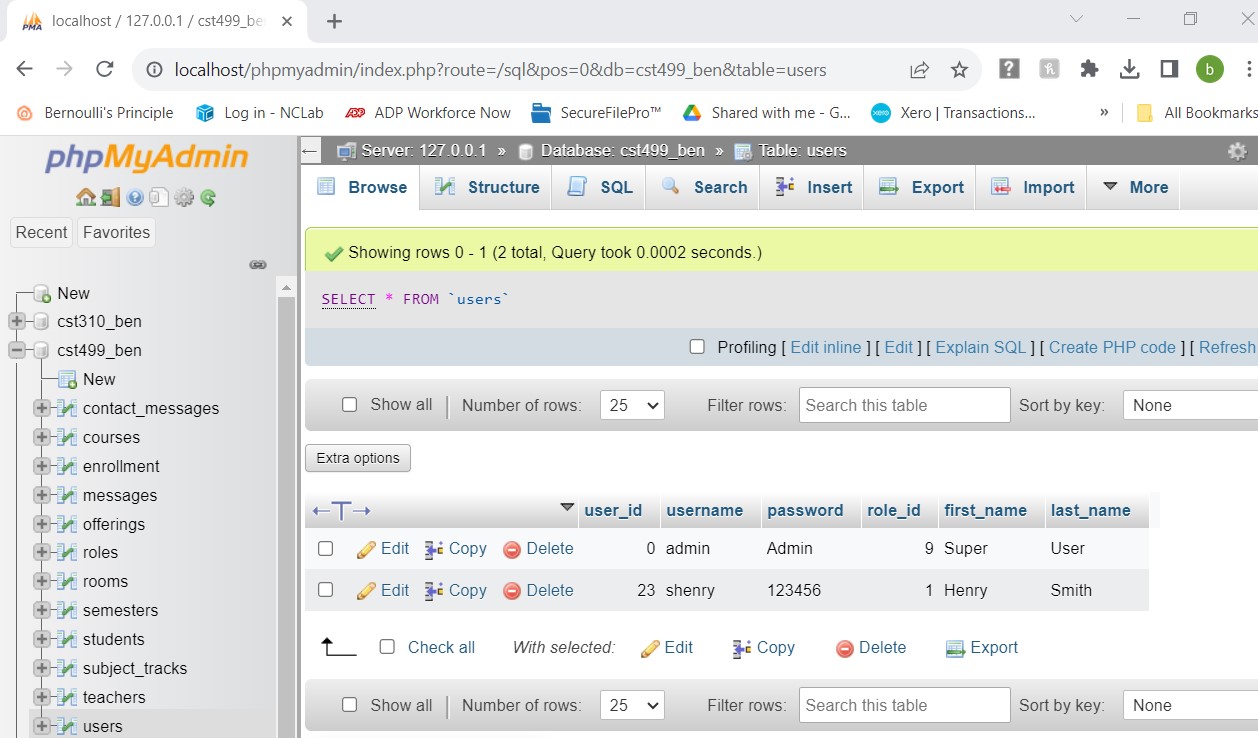
**Figure 9.** New user requires Admin approval.



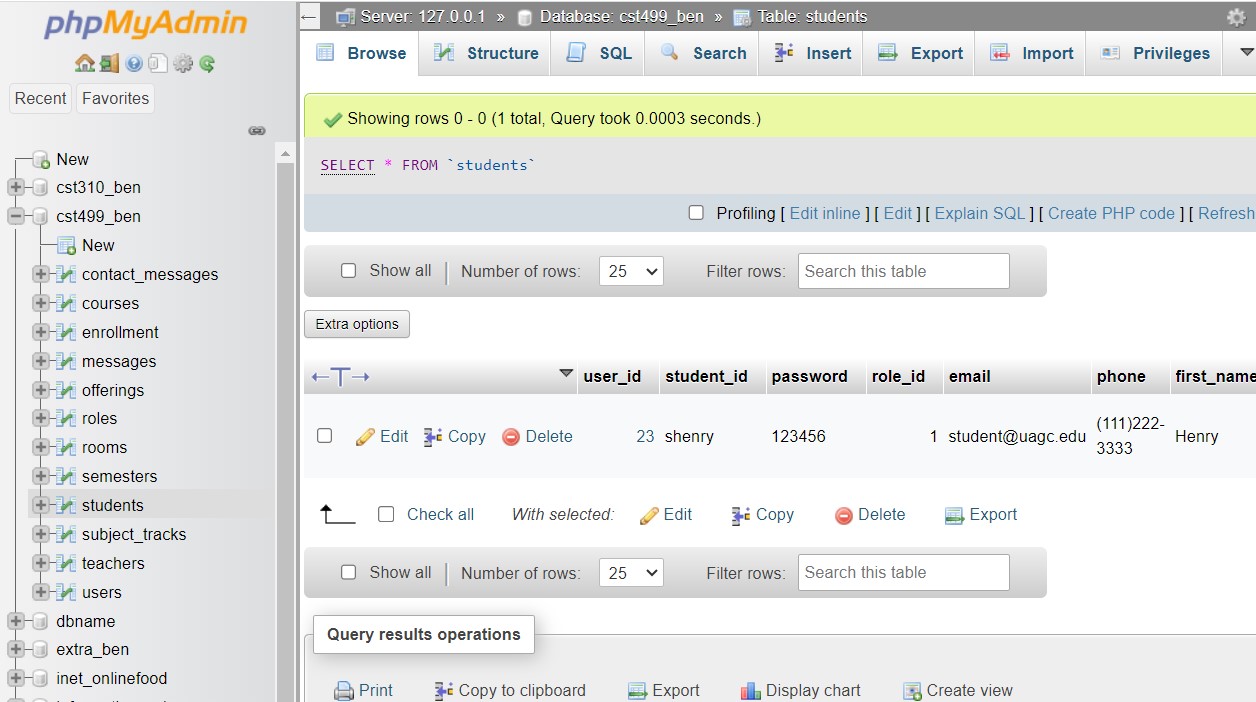
**Figure 10.** Admin changes user 23 shenry from 3 role to Undergrad student 1



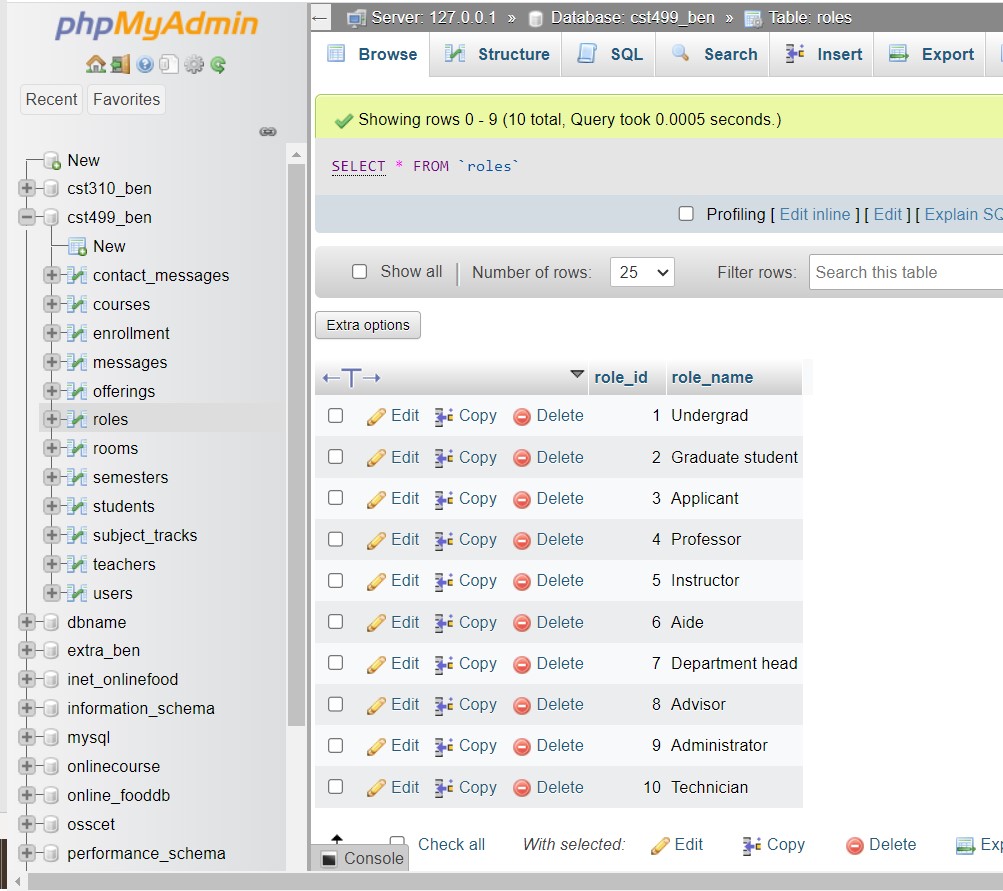
**Figure 11.** Table User 23 with role data changed to role 1 undergrad student.



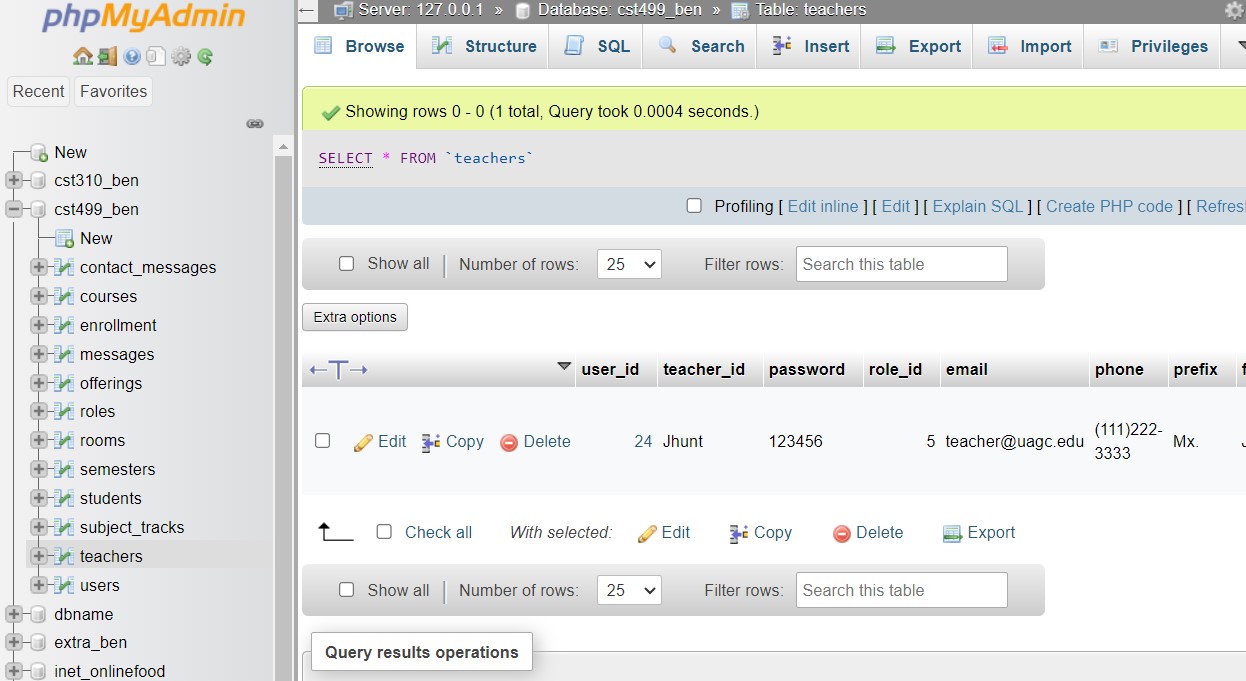
**Figure 12.** User 23 added to Table Students Role 1



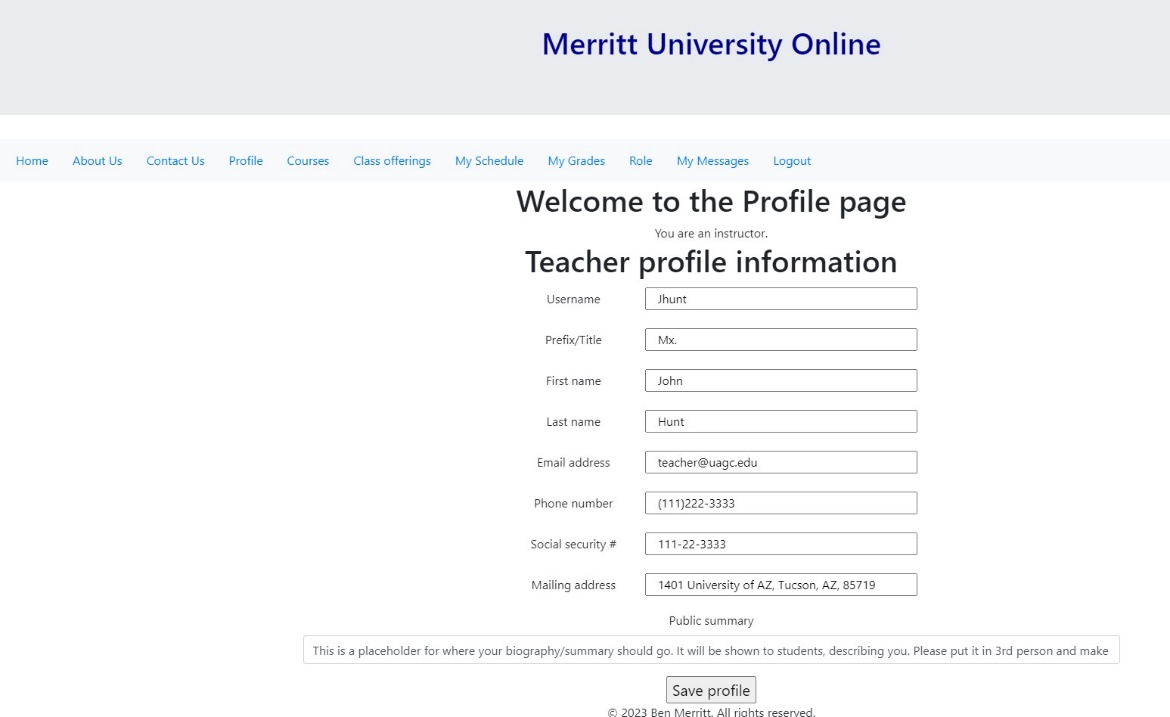
**Figure 13.** Roles Table.



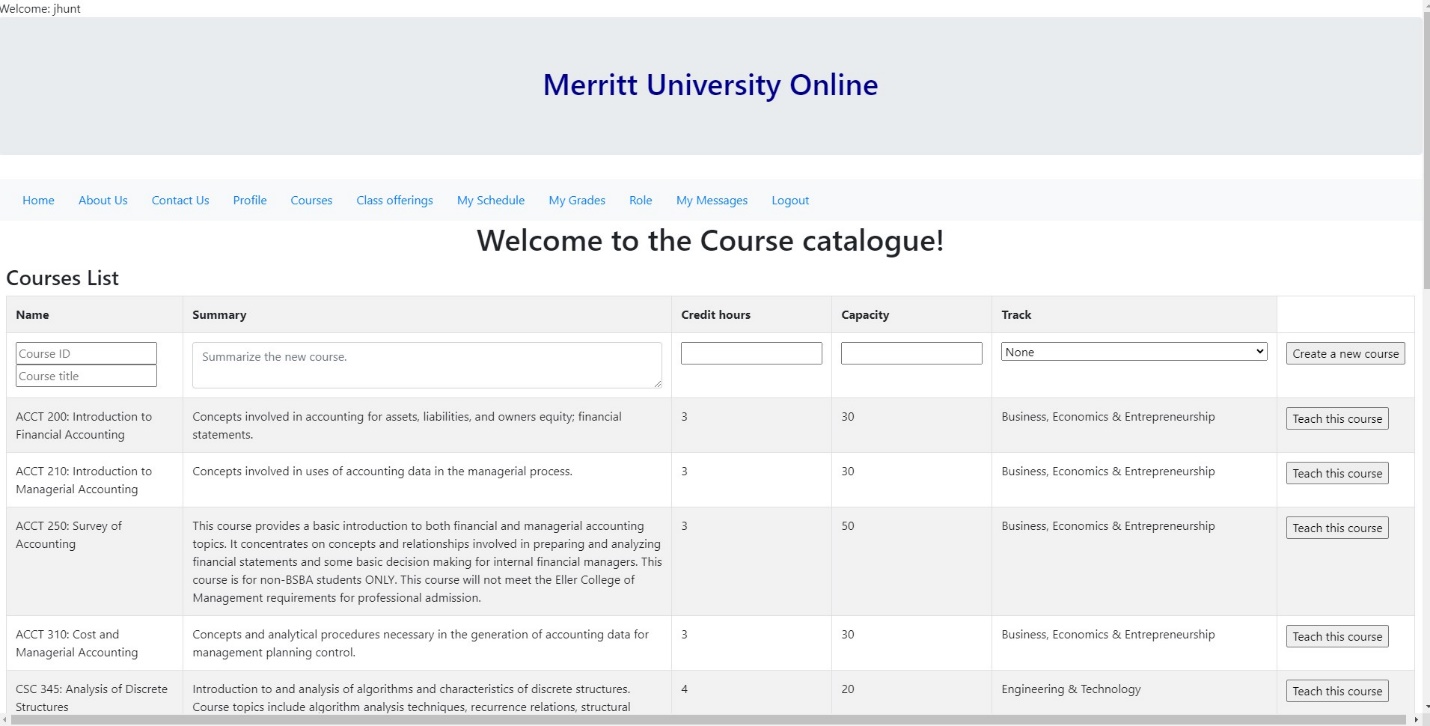
**Figure 14.** Teachers Table with new Teacher added by Admin.



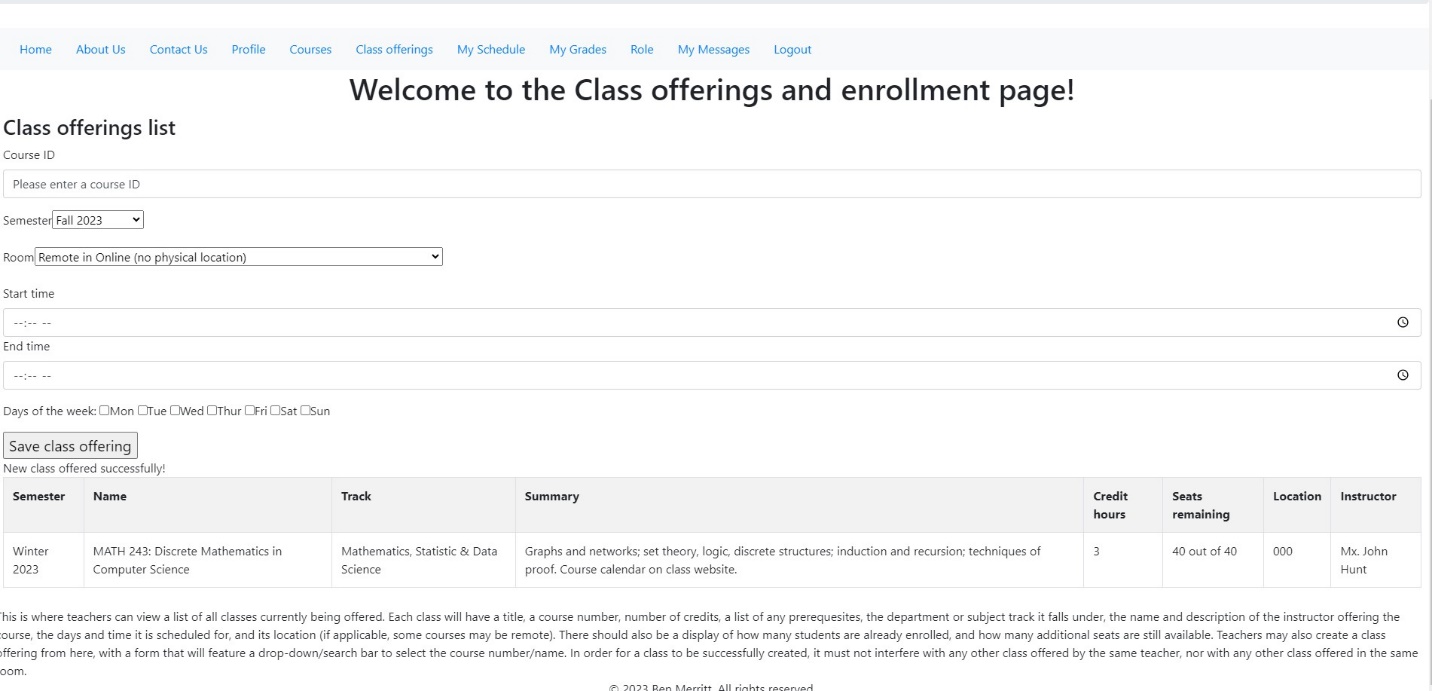
**Figure 15.** Teacher Profile.



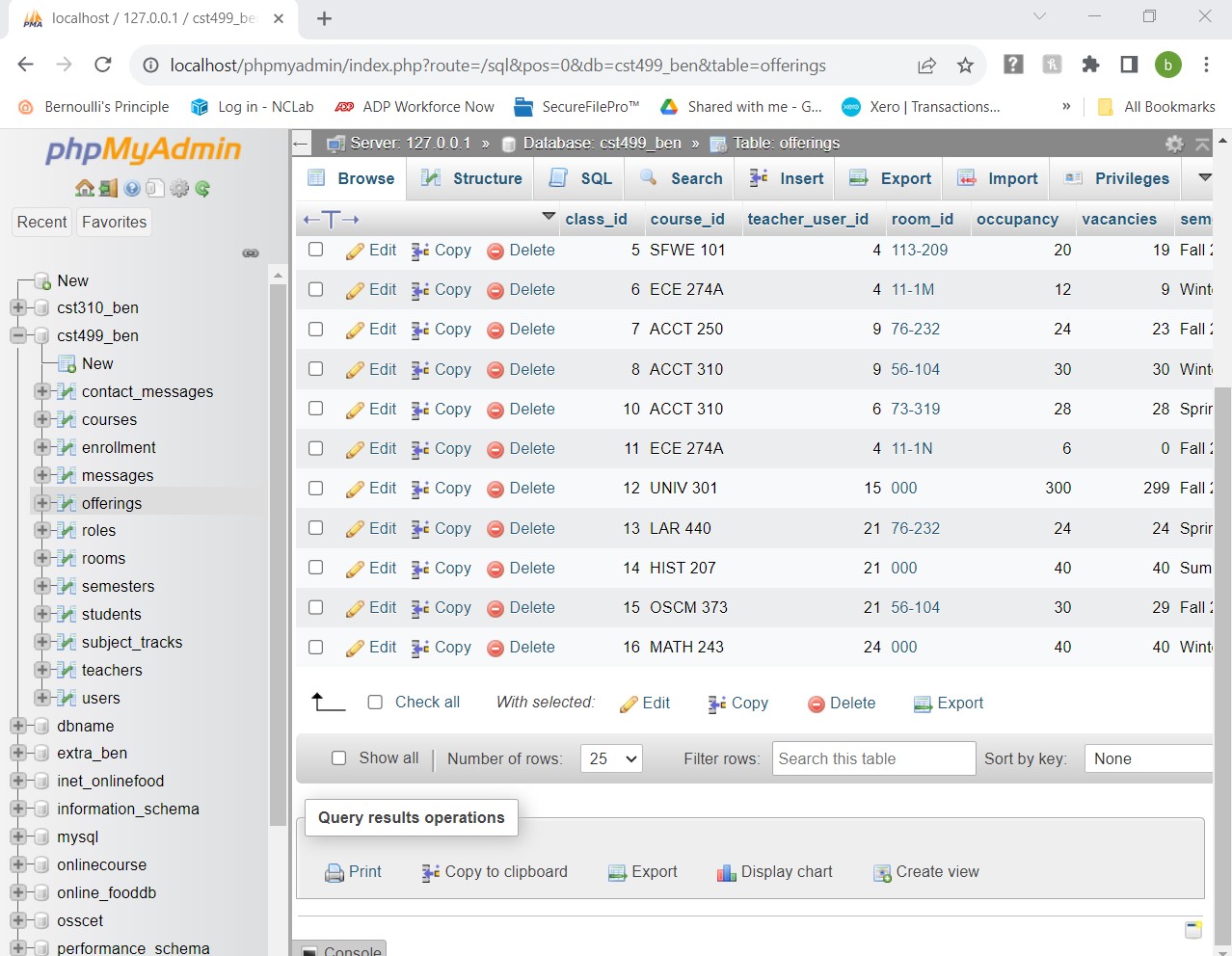
**Figure 16.** Teacher can add, teach, and edit a course



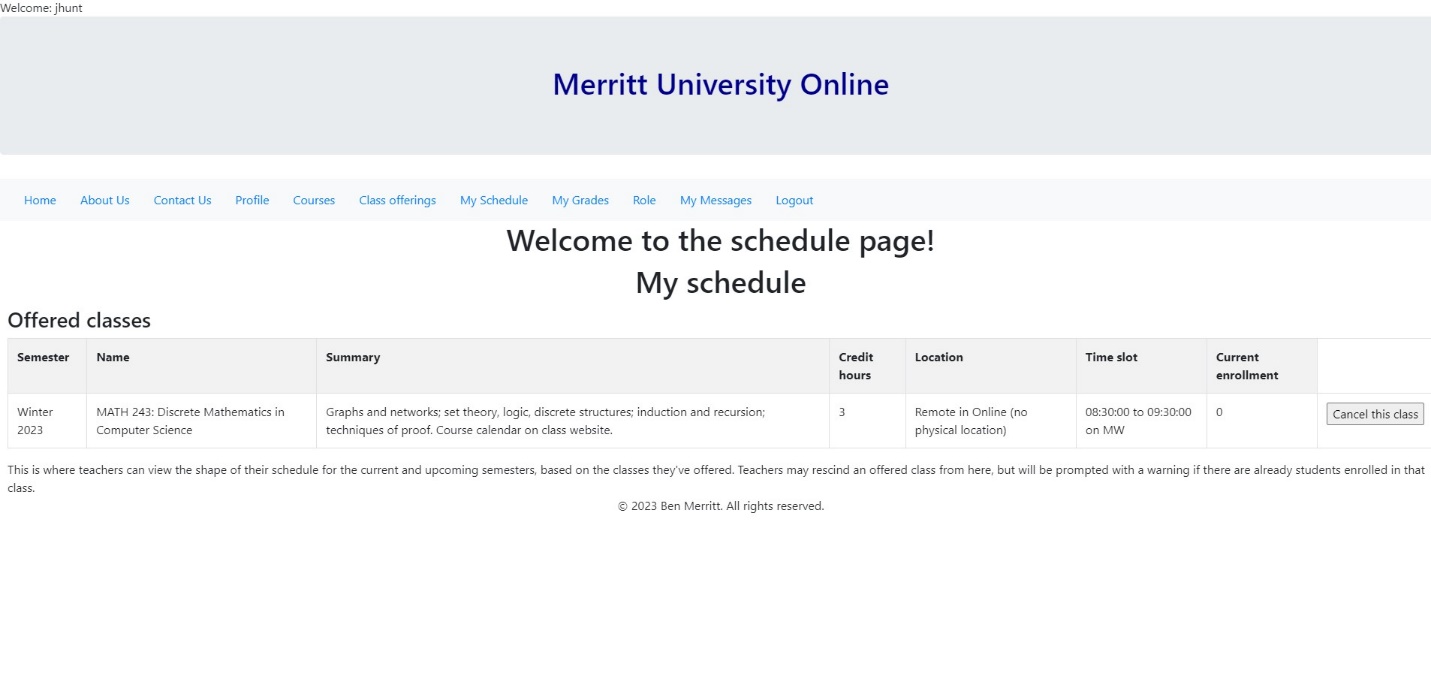
**Figure 17.** Teacher creates a Course



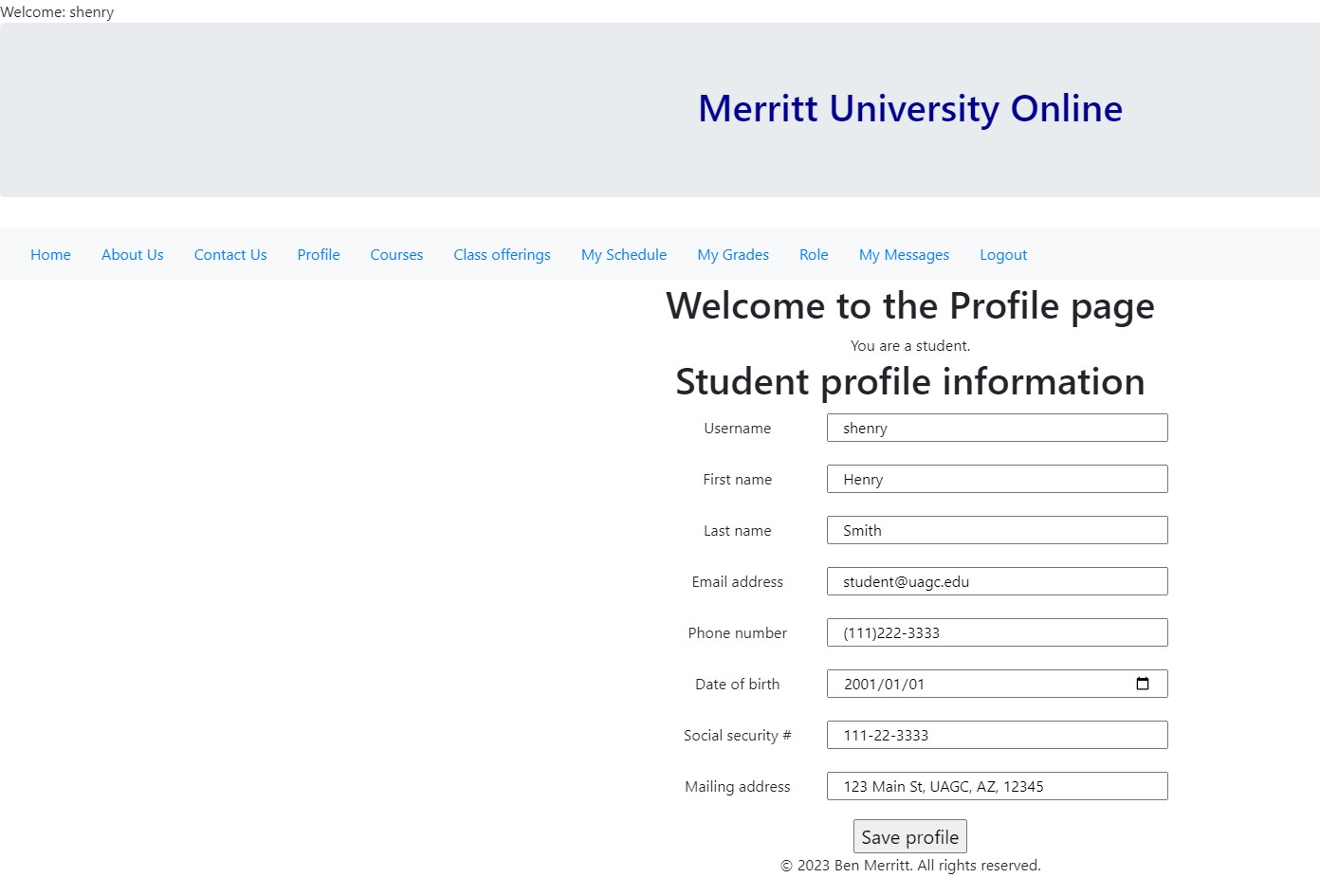
**Figure 18.** Math243 is offered along with other courses



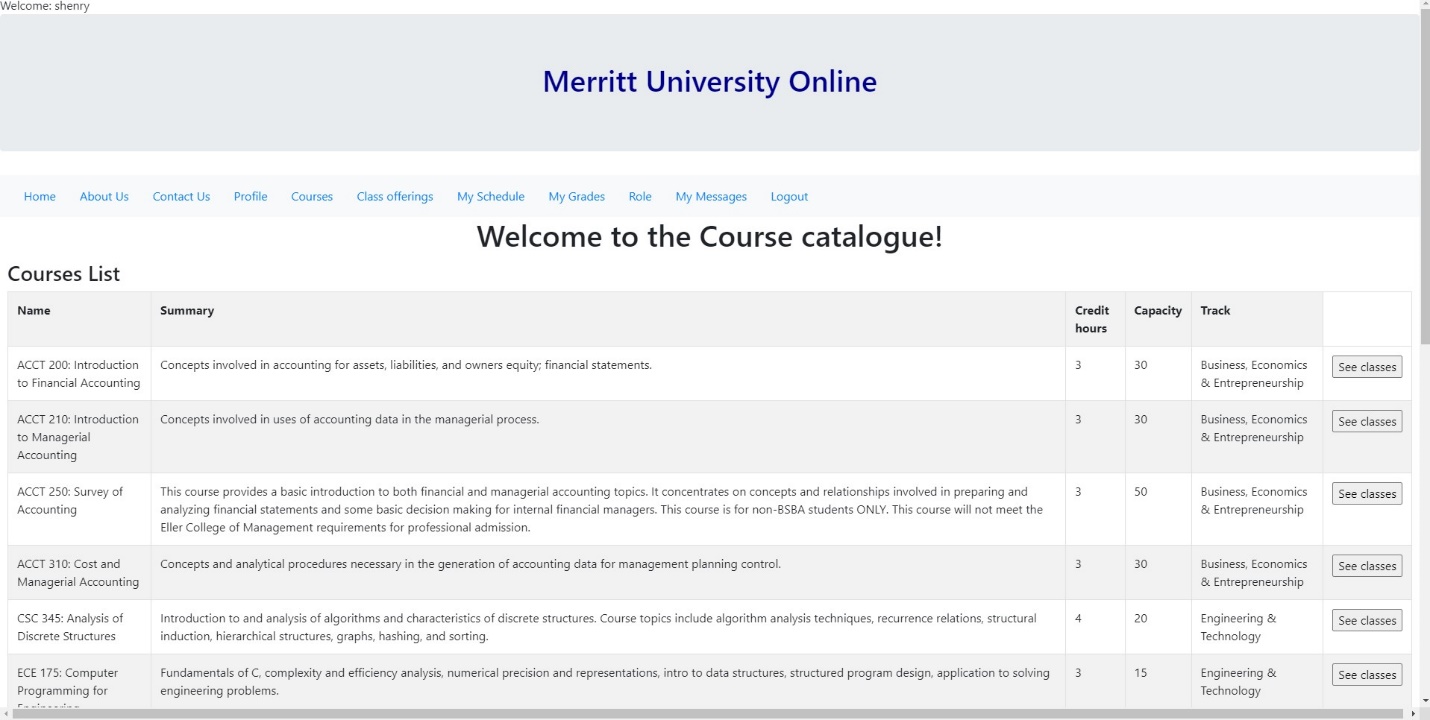
**Figure 19.** Teachers Schedule shows Math 243



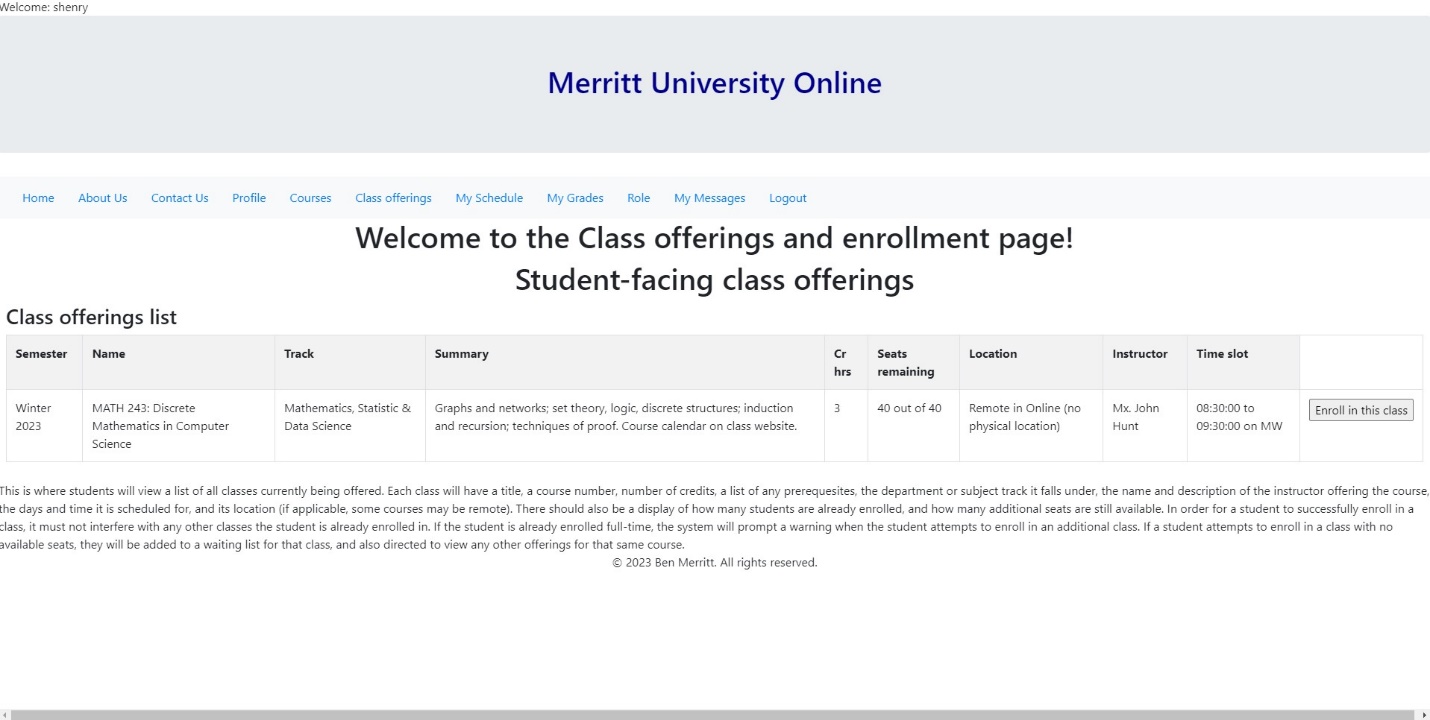
**Figure 20.** Henry Profile as a Student.



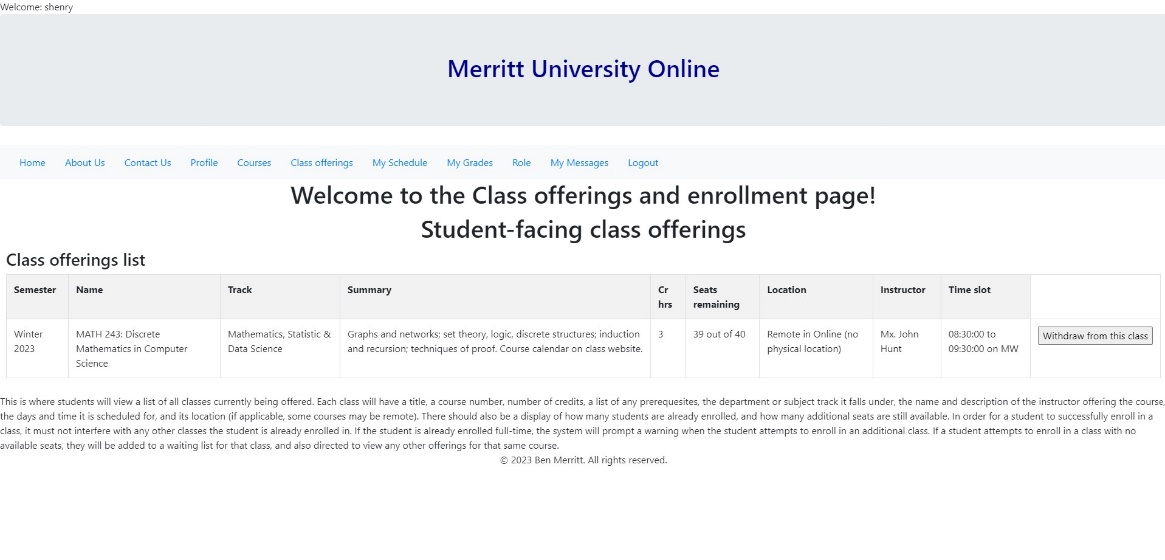
**Figure 21.** Courses View



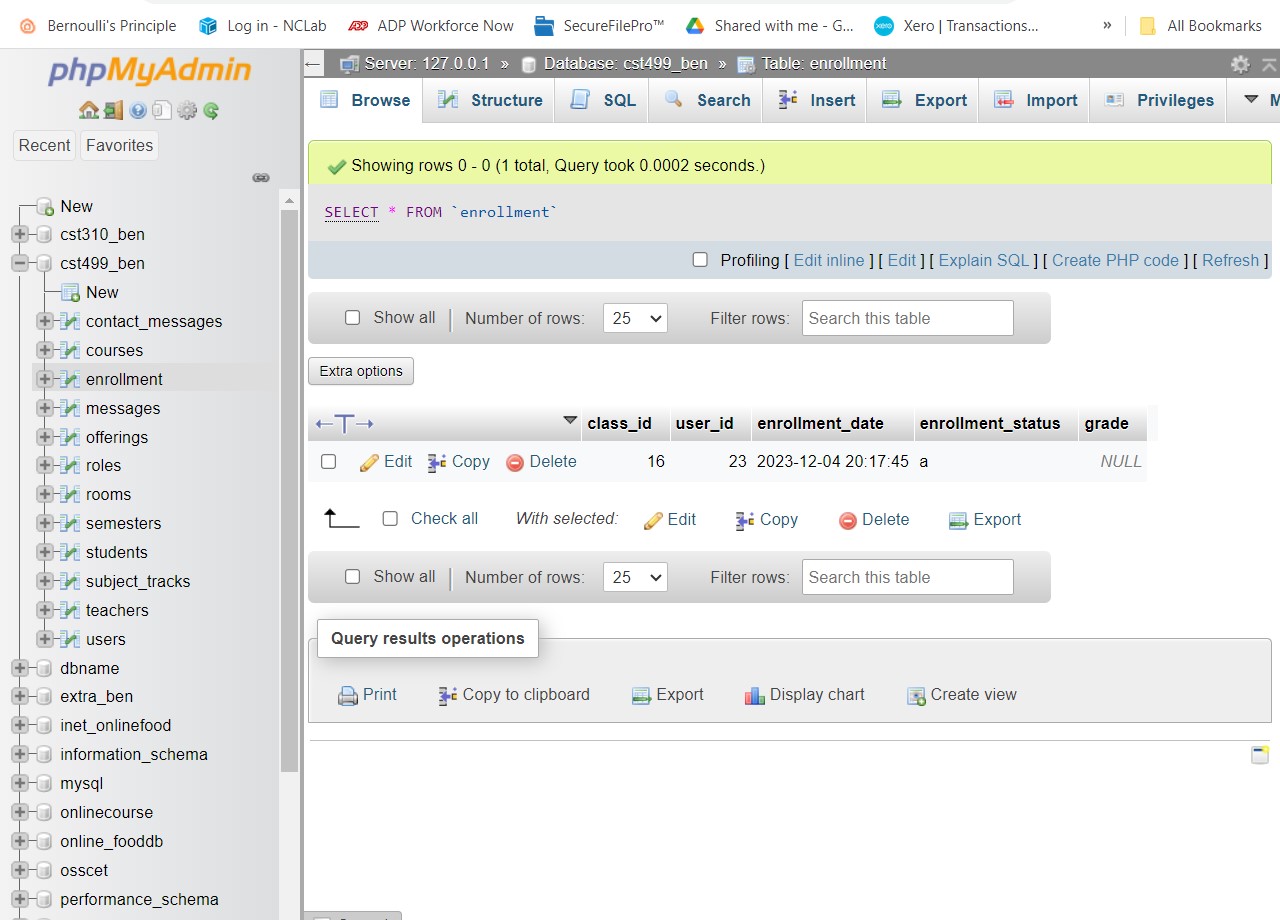
**Figure 22.** Henry Enrolls in course Math 243 offering



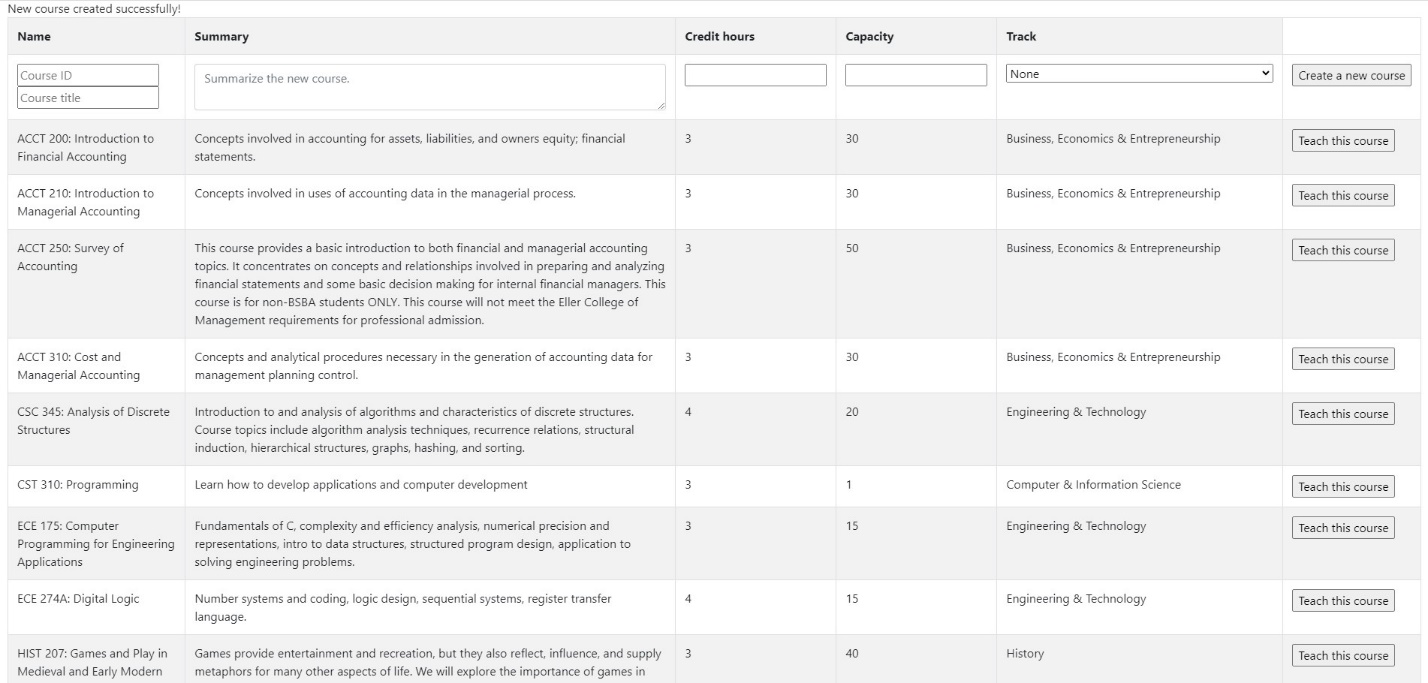
**Figure 23.** Henry is enrolled and can withdraw from the class if wanted.



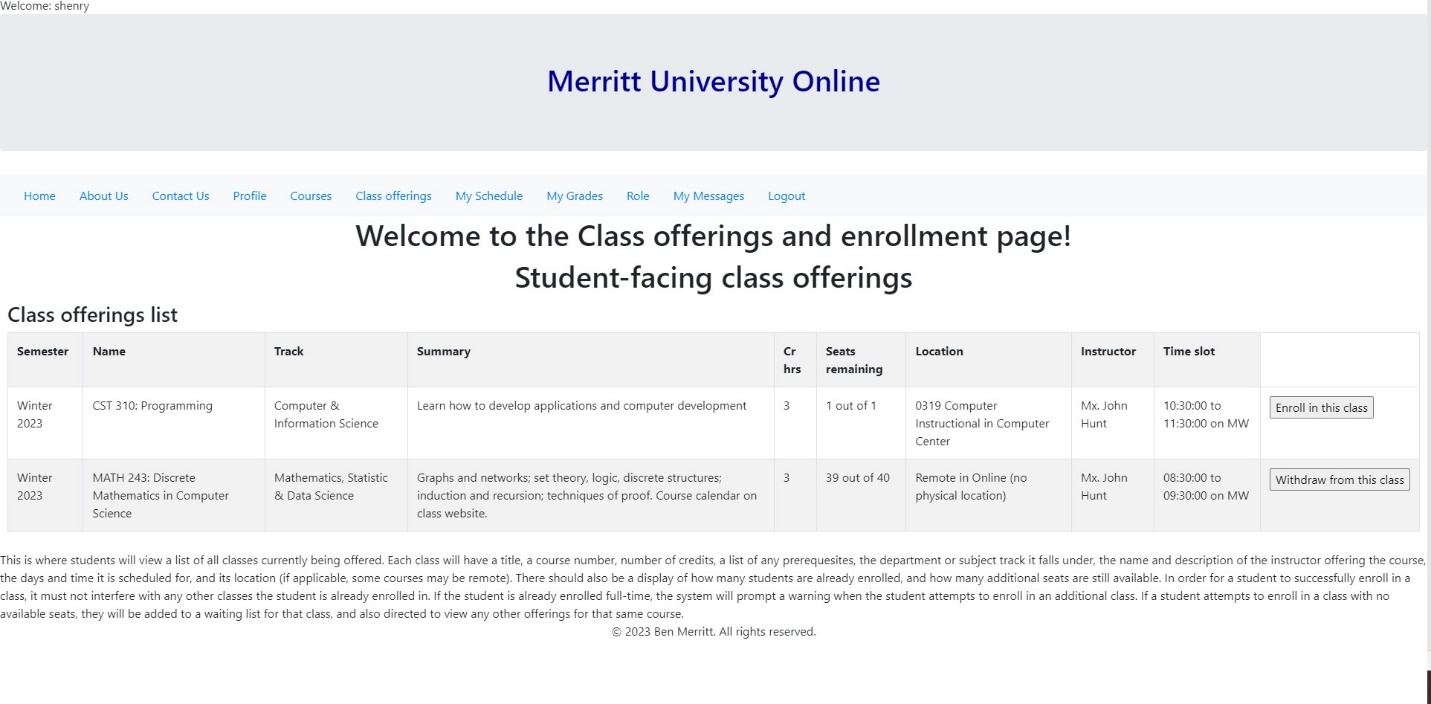
**Figure 24.** Table Enrollment User 23 Henry is enrolled in class 16 Math 243 status "a" Active



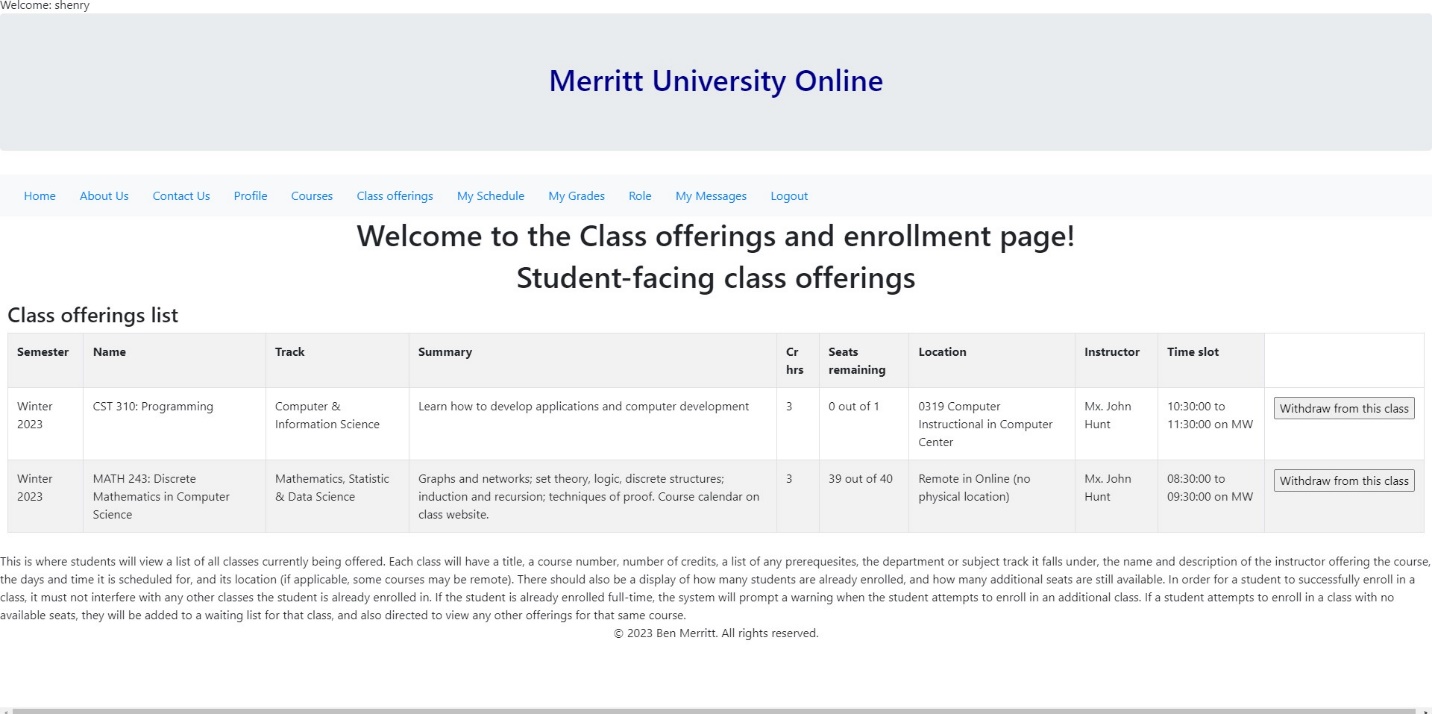
**Figure 25.** CST 310 course created with only 1 seat.



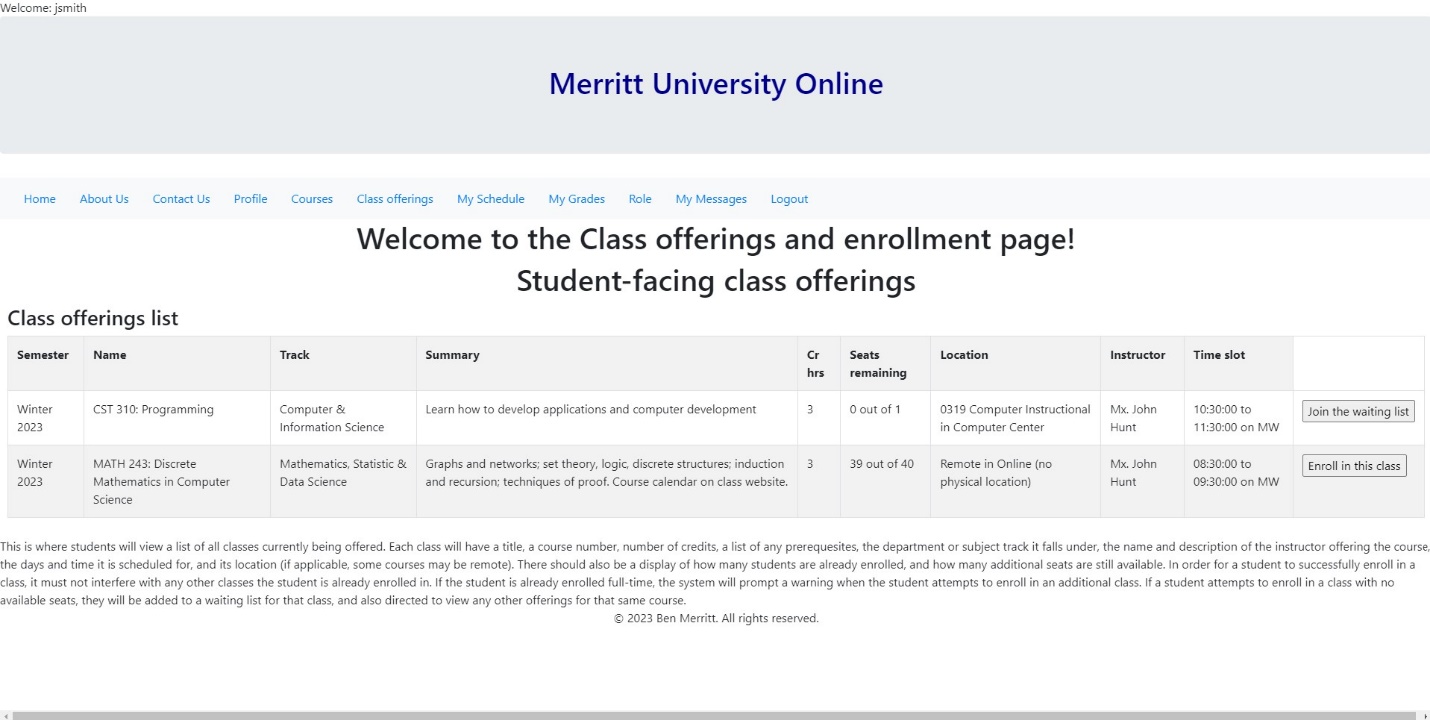
**Figure 26.** Henry enrolls in CST 310 class with 1 seat



**Figure 27.** CST 310 course has 0 seats.



**Figure 28.** Jane Smith jsmith Joins the waiting list for CST 310

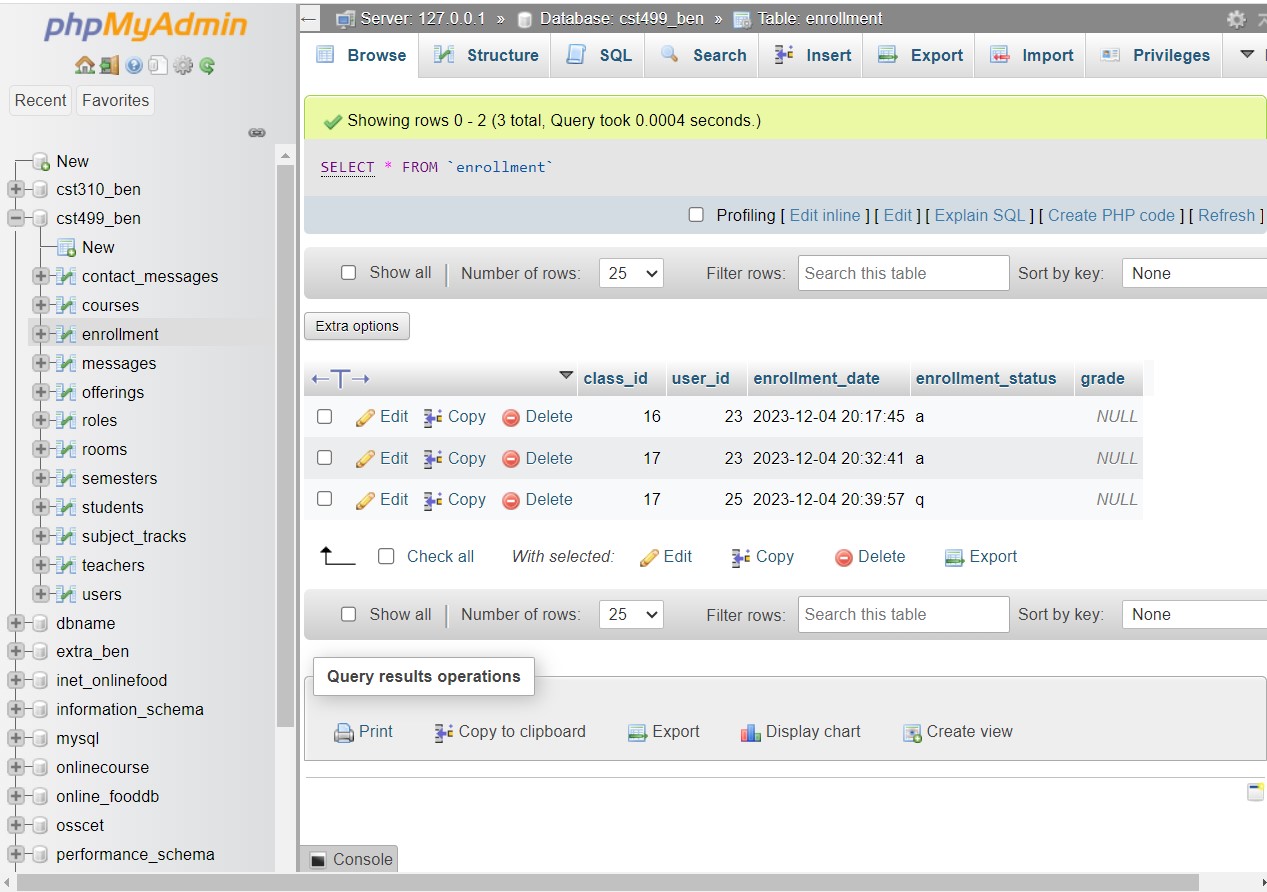


**Figure 29.** Jane Smith jsmith is on the waiting list for CST 310 and can leave if wanted.

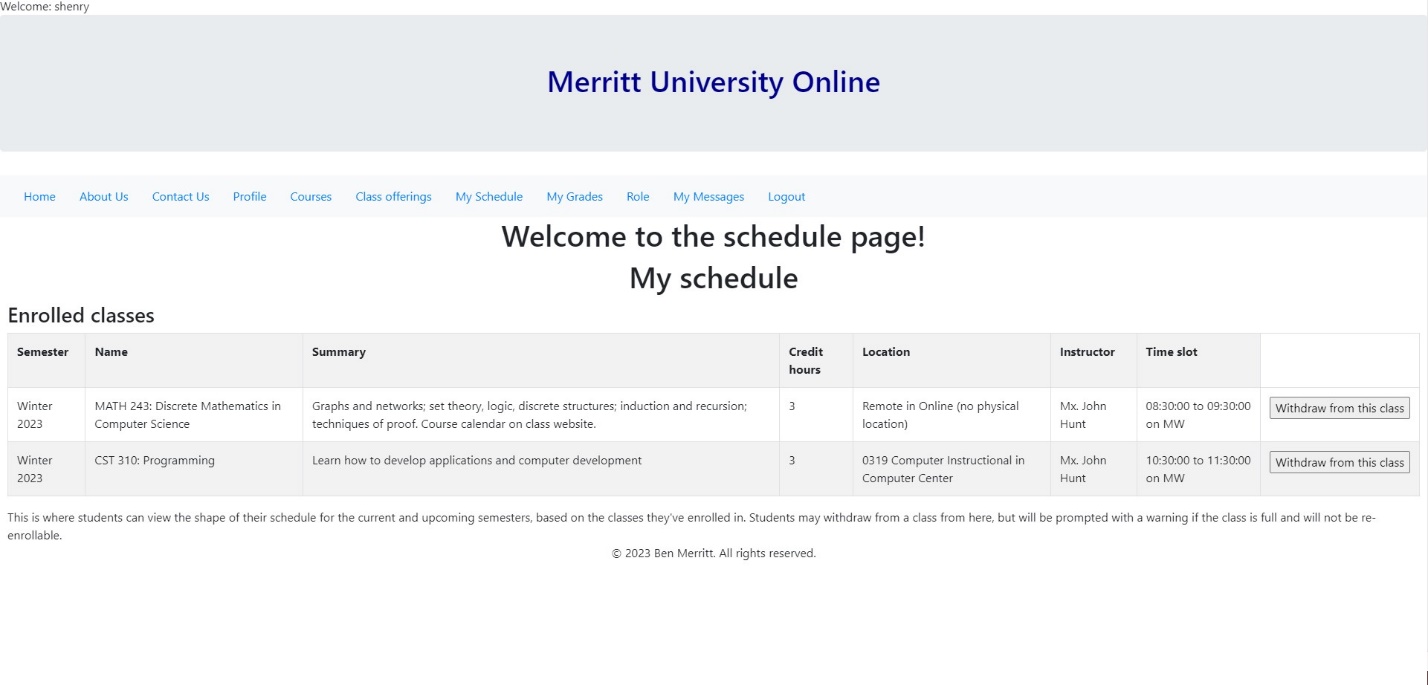
A screenshot of a university online

Description automatically generated

**Figure 30.** Table enrollment Henry got last seat and Jane is waiting with 'q' status.



**Figure 31.** Henry withdraws from CST 310.



**Figure 32.** Jane is Automatically Enrolled in CST 310

A screen shot of a computer

Description automatically generated

**Figure 33.** Henry can Join the waiting list for CST310 if wanted.

A screenshot of a online application

Description automatically generated

**Figure 34.** Table enrollment Jane is enrolled in CST310 & Henry can jion waiting list.

A screenshot of a computer

Description automatically generated

**Figure 35.** course.php code click to expand embedded document



**Figure 36** courses\_student.php code click to expand embedded document.



**Figure 37** courses\_teacher.php code click to expand embedded document



**Figure 38.** messages.php code click to expand embedded document



**Figure 39.** message\_compose.php code click to expand embedded document



**Figure 40.** schedule.php code. click to expand embedded document



**Figure 41.** teacher.php code click to expand embedded document



**Figure 42** schedule\_teacher.php code click to expand embedded document.



**Figure 43.** student.php code click to expand embedded document.



**Figure 44.** shedule\_student.php code click to expand embedded document



**Figure 45** Database code after entering data click to expand embedded document.



**Figure 46** Jane has message she was autoenrolled after being on waiting list.

A screenshot of a computer

Description automatically generated

**Figure 47** Admin shows Jane's message was sent.

A screenshot of a computer

Description automatically generated

**Figure 48.** User 25 Jane got auto-enrolled message in database.

A screenshot of a computer

Description automatically generated

References

Connolly, R., & Hoar, R. (2018). [*Fundamentals of web development*](https://uagc.instructure.com/courses/124945/modules/items/6353220)(2nd ed.). Pearson.

Javatpoint. (2021). *Xampp tutorial*. www.javatpoint.com. Retrieved 2023, from https://www.javatpoint.com/xampp

Jevremovic, A., Ristic, N., & Veinovic, M. (2013). Improving protection of PHP source code using cryptology models. *2013 11th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2013 11th International Conference On*, *02*, 409–412. https://doi.org/10.1109/TELSKS.2013.6704410

Krossing, D. (2015, November 15). *31: Mysql Insert into database | php tutorial | learn php programming | php for Beginners*. YouTube. Retrieved 2023, from https://www.youtube.com/watch?v=IQy8SenpdfM

Mikoluk, K. (2013, September 18). [XAMPP tutorial: How to use XAMPP to run your own web server](XAMPP%20tutorial:%20How%20to%20use%20XAMPP%20to%20run%20your%20own%20web%20server). *Udemy.*https://blog.udemy.com/xampp-tutorial

Oh, S. (n.d.). [*Bootstrap: Tutorial 7*](http://www.cs.toronto.edu/~mashiyat/csc309/Tutorial/6/Bootstrap.pdf). https://www.cs.toronto.edu/~mashiyat/csc309/Tutorial/6/Bootstrap.pdf

SiteGround. (n.d.). [*phpMyAdmin create and populate tables tutorial*.](phpMyAdmin%20create%20and%20populate%20tables%20tutorial.) 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/124945/modules/items/6353220)(4th ed.). Jones & Bartlett Learning.