Week 4 Assignment

Israel Balderas

CST499 Capstone for Computer Software Technology

University Of Arizona Global Campus

Professor: Amjad Alkilani

November 9, 2021

For this week's change to the student portal, we needed to establish a connection to the MySQL database. It also required additional data manipulation from the PHP script to the backend Mysql platform. Last week's assignment included some code that would send data to one table for new users. This week required being able to view available classes, select a class, add it to the student's profile, and, last of all, drop the class.

For the first step establishing a connection to the database, I had already developed this script in last week's assignment. I created a script the hosted the connection in the desired database and called in database.php. This way, I could call this script whenever I needed it without re-creating the connection repeatedly. Following this up, I needed to go back to the MySQL student center database and create two additional tables. I created one table which included all the classes that would be available. The second table would host the classes that were assigned to the student. I used the "CREATE TABLE classes and assignedclass" and added the new column items course, title, and credits, and set the required parameters. Now, this completes the work portion needed in the database; the rest of the work would occur on the PHP side.

Graphical user interface, text, application

Description automatically generated

I had two create two new pages to integrate into the website. The first page would display the available classes and would pull the data from the classes table in the database. The second page would be built similar to the first page but would only display the classes the student had been enrolled into. Because both pages would be very similar, I used the same coding type for both by pulling different tables. I did this by creating a new query into the database that would pull the data from the database and display it in a table. I then added the form action function and added the ability to add select each class. I created another script to handle the selected class called "classregistration.php."



The classregistraion script captured the selected class ID's using the \_POST function. I then queried the classes table and used the fetch function to pull the information from "courseID, Course, and Title row. I added this information and assigned it to the variables. Following this, I created a new variable to insert the new variables into the assignedclass database. This completed the work needed for the available classes page.

Returning to the current classes page, I made a similar approach that I used for the class registration page. Instead, this time I made the form selections would need to drop the classes selected. I reused the code I used for classregistration.php. This time I used the Delete function that captured the selected courseID and deleted it from the table.



This wraps up the basics of the code, but there are still many bugs in the system. If there is to be continued development, several items would need to be revisited to correct issues. For now the system functions and meets the requirements off adding a class or dropping a class. Below are screenshots of the page in progress.

**Adding a class:**

Table

Description automatically generated  
**Class Added**

Text

Description automatically generated

**Viewing a available classes and dropping a class.**

Table

Description automatically generated