**Database Development and Class Registration**

Dennis Walters

University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Dr. Amjad Alkilani

November 15, 2021

**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. At some point in the future additional functionality may be added to the Portal to provide additional information, or access to other applications like course material, or grades.

**Steps Taken to Combine All Functionality**

Throughout this course I chose to build each piece of functionality upon what had been previously built rather than building each function separately then combine them together into a single application at the end. Thus, I added the new functionality to the previous week’s completed website. In week 3 I took the provided website template and installed into the instance of XAMPP I had installed on my computer then tested it to ensure it was working as expected. I then took the website template and rewrote it in a Model/View/Controller framework then created the user registration page and a database to store the user registration information in. I continued by adding a student login page to the previous website code so that the Student Portal consisted of both the user registration and login functionalities. I finished the week out by adding the student profile page which allows the logged in user to see their own information they entered during registration and modify it. The last thing I added in week 3 is the logout functionality to the Student Portal allowing users who have logged in to log out and secure their information. To do this I cleared out the $\_SESSION array and the session cookie then used a PHP function called destroy\_session to clean up everything else.

In week 4 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 full, notify a student on the waitlist when they are enrolled in a class that has gained an opening, and remove that same student from the waitlist. In order to accommodate these functions, I had to add several new pages and database tables. Please see the screenshots below.

**Figure 1**

User home page

Graphical user interface, text, application

Description automatically generated

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

**Figure 2**

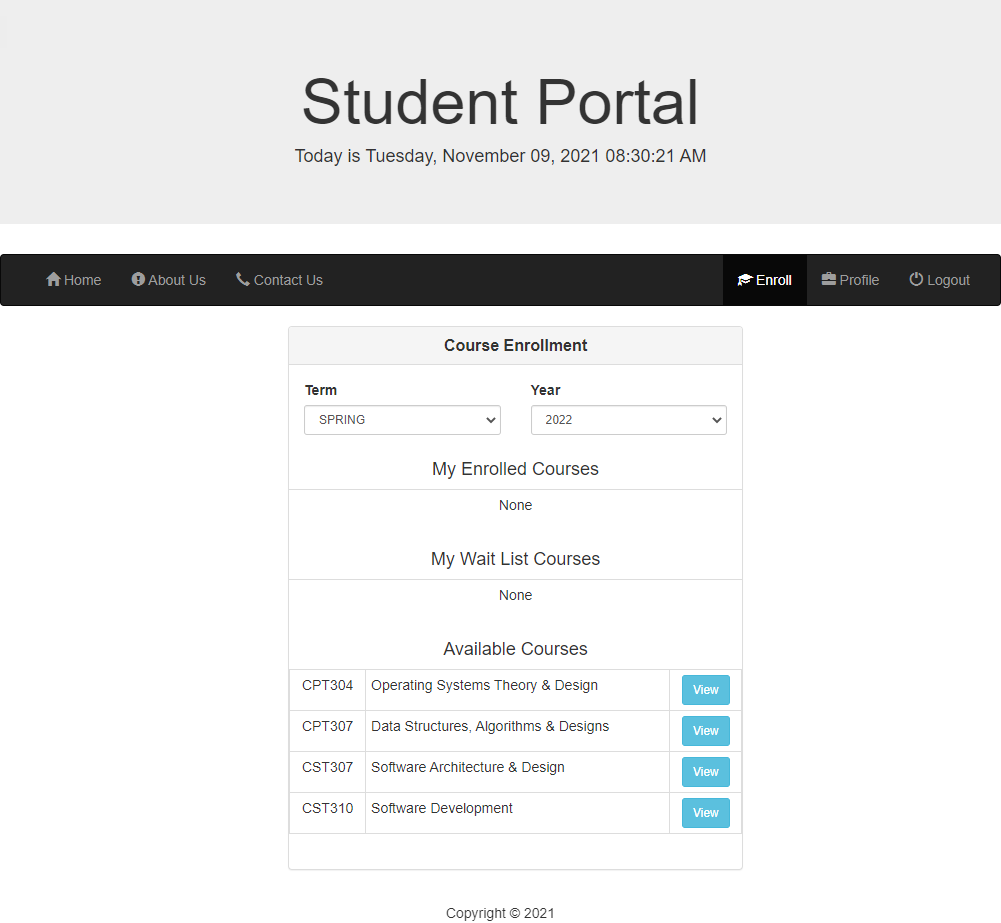
Course enrollment page

Graphical user interface, application

Description automatically generated

**Figure 3**

Select Term and Year



**Figure 4**

Course details

Graphical user interface, application

Description automatically generated

**Figure 5**

Course details of a course that is full.

Graphical user interface, application

Description automatically generated

*Note.* The student is notified in the details when a class is full. In addition, the ***Enroll*** button becomes a ***Add To Waitlist*** button when the class has reached the maximum number of enrolled students allowed.

**Figure 6**

Notification of successful enrollment.

Graphical user interface

Description automatically generated

**Figure 7**

Notification of successfully being added to the waitlist.

Graphical user interface, application

Description automatically generated

**Figure 8**

Course details after successful enrollment.

Graphical user interface, text, application

Description automatically generated

*Note.* The ***Enroll*** button changes to ***Drop Course*** after successful enrollment in the course.

**Figure 9**

Notification of successfully dropping a course.

Graphical user interface

Description automatically generated

**Figure 10**

Course details after successfully being added to the waitlist.

Graphical user interface, application

Description automatically generated

**Figure 11**

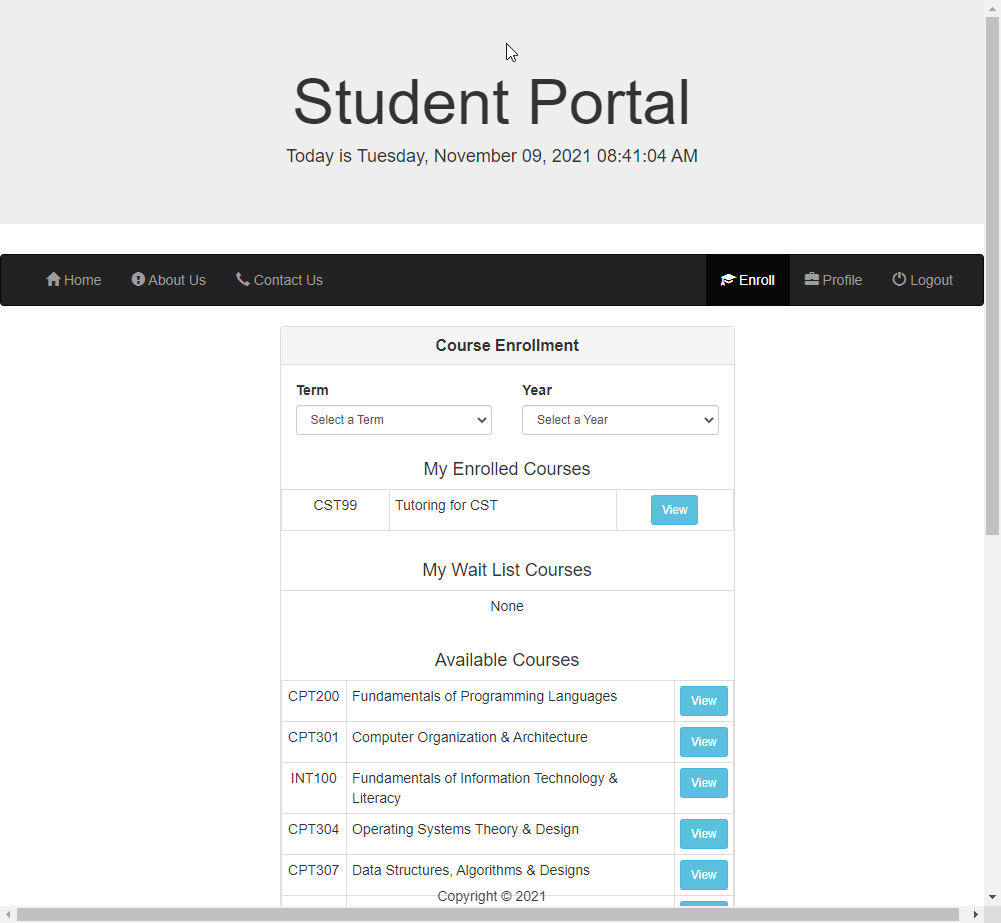
Notification upon login that user has been moved from the waitlist to enrolled in the course.

Graphical user interface, text

Description automatically generated

**Figure 12**

Enrollment page after student has been moved from the waitlist to enrolled.



**Figure 13**

mySQL before creation of database.

Graphical user interface, text, application, email

Description automatically generated

**Figure 14**

mySQL create database

Graphical user interface, text, application, Word

Description automatically generated

**Figure 15**

Create tables

Graphical user interface, text, application

Description automatically generated

**Figure 16**

Populate tables

Graphical user interface, text, application, Word

Description automatically generated

**Figure 17**

Home.php Controller

Text

Description automatically generated

**Figure 18**

Enroll.php Controller

Text

Description automatically generated

**Figure 19**

Login.php Controller

Text

Description automatically generated

**Figure 20**

Enrollment.php Model

Text

Description automatically generated

**Figure 21**

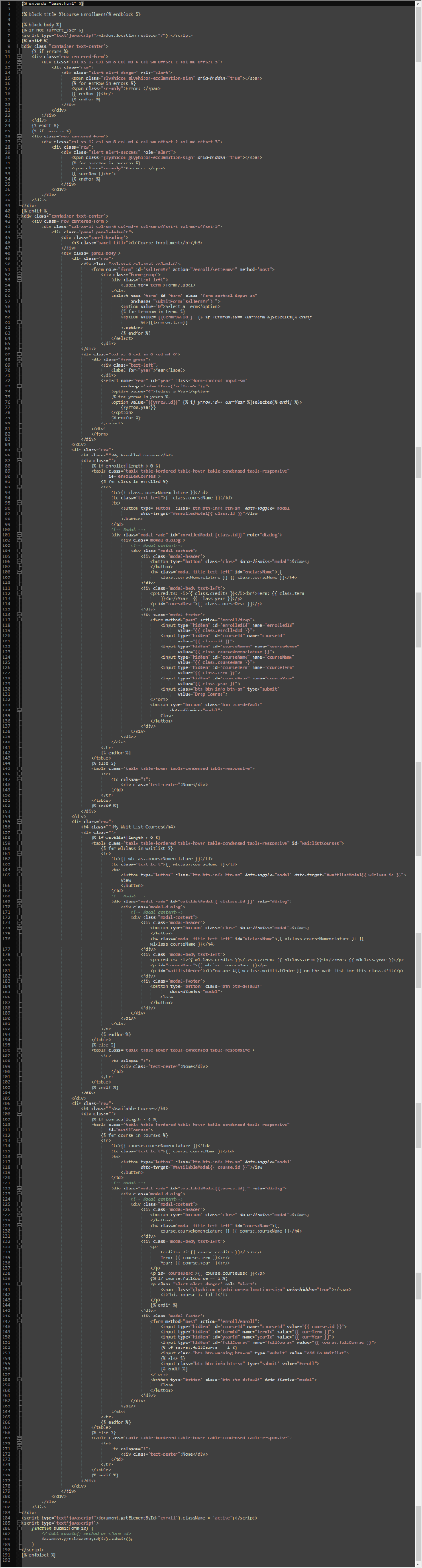
User.php Model

Text

Description automatically generated

**Figure 22**

Enroll\index.html View



**Figure 23**

Home\index.html View

Text

Description automatically generated

**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>.