**Database Development and Class Registration**

Hayden Herrig

CST499: Capstone for Computer Software Technology

Prof. Charmelia Butler

February 26, 2024

**Software System Development**

There is a lot that goes into developing a software system. It is often made up of several components that all come together to solve a problem. Developing the system is only part of the battle. The system also must be run to test and verify all the functionalities of the system. It is important to be able to understand what the source code of the system is doing and how to run the system in order to verify functionalities.

**How to Run a PHP File**

Assuming that PHP is already installed on the device running the files is relatively simple. For this particular example, to use PHP we are using XAMPP. The first step to opening a PHP file with XAMPP is to make sure the desired file is saved in the htdocs folder within the XAMPP installation. Next, open the XAMPP control panel. This allows different services installed with XAMPP to be activated or deactivated (Mikoluk, 2024). Once the control panel is open Apache needs to be activated to access the PHP files. Once Apache is up and running the next step is to access a web browser. Once the web browser is open the file can be accessed by typing localhost/filename.php. When actually opening a PHP file the name of the desired file should be used in place of filename.php (Mikoluk, 2024).

**Student Portal Database and Tables**

The first thing I did was develop the pages that would display to the user and allow them to navigate through the different pages of the site. The pages were a landing or home page, a registration page, and a log in page. All these pages contain a navbar that allows the user to navigate to any of the pages.

Once the pages had been developed, I created a database that would store all the user or student information as well as course information. The tbluser table is used to store all of the students’ registration information such as their first name, last name, email address, password, and phone number (SiteGround, 2024.). This table can then be referenced at the login page to allow students to log in to the system by referencing the username and password entered at the login page against the database. Another table was created to store course information. This table had the course ID, course name, which semester the course is available for, and how many students could register for the course. This table is referenced to present the available class options to the students.

To connect to the database the database was identified in a php configuration file by creating variable to identify the database host, database name, the user, and the password for the database. Then creating a variable called $con to create a connection to the database utilizing mysqli\_connect and the previously identified database information. The connection to the database can then be checked and allow information to be passed from the web application to the database.

**Registration Page**

The first step of building the registration was to identify what information users would need to submit on the registration page. The next step was to set up the form that users would fill out on the page. I did this by creating the necessary number of text inputs to allow the users to enter all the information. After creating the form, I used PhpMyAdmin to create the database to link to the website. I created the database and then the table that would store the registration page information entered by the users. Then I utilized the previously mentioned configuration file to connect the site to the database. I also utilized a registeruser php file to handle all of the information entered into the form and insert it into the database. This created a variable for each of the values that would be input into the form. After the information has been submitted, the user is then returned to the login page with all fields empty.

**Course Enrollment, Cancellation, and Schedule**

Building these pages was more difficult than I thought it would be. Firstly, the course enrollment page allows the student to select a course they wish to enroll in and then adds the course to their schedule. This requires querying the course table to add the course to the student’s schedule. First the student must enter the information for the course they wish to enroll in after hitting submit the course information is entered into the student’s schedule. Next was the cancellation system. This allows students to remove a course from their schedule. This is done by entering the information for the course the student wishes to drop and hitting submit. After submitting the database is queried to identify the correct course in the schedule to drop the course. Lastly, the schedule page displays the current courses the student is enrolled in. This is done by querying the database to look for courses that the student is enrolled in and displaying the courses the student has enrolled in.

**System Screenshots**A computer screen shot of a registration portal

Description automatically generated

**(FIG 1. Student Registration System Landing Page)**A blue and yellow sign in a computer screen

Description automatically generated

**(FIG 2. Student Registration System Login Page)**

A screenshot of a registration page

Description automatically generated

**(FIG 3. Student Registration System Registration Page)**

**A computer screen shot of a registration page

Description automatically generated**

**(FIG 4. Student Registration System Schedule Page)**

**A screen shot of a computer

Description automatically generated**

**(FIG 5. Student Registration System Course Registration)A screenshot of a computer

Description automatically generated**

**(FIG 6. Student Registration System Course Removal Page)**

A screen shot of a computer program

Description automatically generated

**(FIG 7. Student Registration System Configuration File)**

A screen shot of a computer program

Description automatically generated**(FIG 8. Student Registration System Landing Page File)**

A screenshot of a computer program

Description automatically generated

**(FIG 9. Student Registration System Login File)**

A screen shot of a computer program

Description automatically generated

**(FIG 10. Student Registration System File for Inserting User Data Into tbluser)**

A screenshot of a computer program

Description automatically generated

**(FIG 11. Student Registration System Registration Page File)**

**A screen shot of a computer program

Description automatically generated**

**(FIG 12. Student Registration System Current Schedule Page File)**

A screen shot of a computer program

Description automatically generated

**(FIG 13. Student Registration System Add Course Page File)**

**A screen shot of a computer program

Description automatically generated**

**(FIG 14. Student Registration System Course Removal Part 1)**

**A screen shot of a computer program

Description automatically generated**

**(FIG 15. Student Registration System Course Removal Part 2)**

**A screenshot of a computer

Description automatically generated(FIG 16. Database Containing Tables for Student Registration System)**

**A screenshot of a computer

Description automatically generated(FIG 17. Student Course Table Before Course Added)**

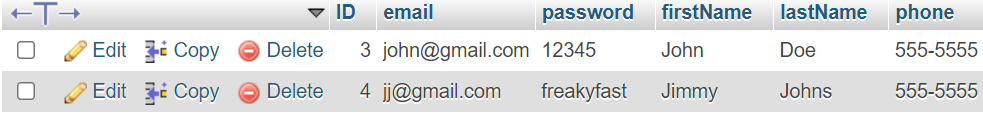
**A screenshot of a computer

Description automatically generated**

**(FIG 18. Student Course Table After Course Added)**

**A screenshot of a computer

Description automatically generated(FIG 19. Table for Storing Student Information Entered on Registration Page Before New Entry)**

****

**(FIG 20. Table for Storing Student Information Entered on Registration Page After New Entry)**

**Conclusion**

Breaking down how to run the files to operate the system helps to ensure that the system functions as intended and meets all the requirements. Breaking down how different parts of the system were developed provides a better understanding of how the system functions. The combination of being able to run the system and understand the systems parts helps ensure proper functionality of the system and easily identify issues within the system.

# References:

Mikoluk, K. (2024). XAMPP Tutorial: How to Use XAMPP to Run Your Own Web Server. Udemy. Retrieved February 19, 2024, from <https://blog.udemy.com/xampp-tutorial/>

SiteGround. (2024). PhpMyAdmin Database Management Tutorial. SiteGround Knowledge Resources. https://www.siteground.com/tutorials/phpmyadmin/database-management/