**Week 3 Assignment**

**Landing, Login, and Enrollment Pages Development**

Eric Williams

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Professor Joseph Rangitsch

Due April 29, 2024

**Week 3 Assignment**

**Landing, Login, and Enrollment Pages Development**

## How to Run a PHP File In XAMPP

Running a PHP file in XAMPP first requires having XAMPP installed on your machine. The instructions for completing this task can be found via any search engine and will vary depending on the operating system running on your computer. Another prerequisite requirement is some sort of text editor or specialized IDE for inputting your code. Notepad ++ is a great example of a text editor that displays HTML in an easy-to-read manner.

Next, ensure that XAMPP is running by running the XMAPP Control Panel. Click on the “Start” button for both Apache and MySQL to ensure those services are running. You can now create a PHP file within your text editor or IDE (for example, master.php) that contains the code for your project. Save the file in the htdocs directory for XAMPP, which is usually at C:\xampp\htdocs.

You should now be able to open the file within your web browser. XAMPP uses the address of <http://localhost/> as the test server, and that URL will open the default page for XAMPP. To view your PHP file, just add the filename after the URL. For example, <http://localhost/master.php> will load the above-mentioned file and run the PHP script contained within.

## Creating the Registration Page:

Creating the registration page was a two-step process, beginning with creating a functioning PHP code that would execute our expected behaviors. The most basic expectation for our behavior is that the user enters data into several inputs on form, then that data is transferred via POST to the appropriate columns in our tblUser table in the college\_registration database. Below, the code for the registration.php file is broken down by line and function to explain how this was accomplished.

1. Require\_once “server.php”; allows the registration.php file to access any classes, functions, or constants that are defined in the server.php file.

2. $db = new Database("localhost", "college\_registration", "root", ""); creates a new instance of the Database object (from server.php)

3. if ($db->connect()) is a conditional check that will only allow the contained code to run if there is an active connection with $db, which should have been done during the previous step.

4. if ($\_SERVER['REQUEST\_METHOD'] == 'POST') is another conditional check that verifies if a POST request has been made to the server. If so, we continue running the code block to retrieve the data entered by the user into the form.

5. $sql = "INSERT INTO tblUser…; This block prepares an SQL statement that will insert the data into the tblUser table. For security reasons, to help prevent malicious SQL injection attacks, this block uses placeholders for the actual values, which will be replaced during execution.

6. $result = $db->executeQuery($sql…; This block executes the SQL query that was prepared above by passing an array (the second argument) that contains the actual values of our placeholders.

7. if ($result > 0)…; This is a status check. If the query is executed successfully, the registration page will echo “Registration successful!”. If the code doesn’t successfully execute, it will echo “Registration failed. Please try again.”.

8. $db->disconnect(); Lastly, we disconnect from the database object created earlier.

Once the function has been established and tested, esthetics and usability are addressed. This is done through our master.php and footer.php files created last week to form the basic layout and toolbar, along with the code to create the POST form.

1. <form method="post"> is used to define the form information as POST data.

2. Each individual field in the form contains a label and input field, along with a unique ID and name attribute. The label informs the user which data they should input into which field, whereas the other attributes are used by the POST method to transfer the data to the appropriate locations in the table.

3. <button type="submit">Register</button> is a simple button that executes the submit command in POST.

4. Lastly, we close the form and container with the appropriate close tags, </form> and </div> respectively.

## Screenshots

Home

A computer screen shot of a registration

Description automatically generated

Login

A computer screen shot of a registration page

Description automatically generated

Registration

A computer screen shot of a registration page

Description automatically generated

## Code Snippets

Home Page HTML

A screenshot of a computer

Description automatically generated

Registration Page PHP

A screenshot of a computer program

Description automatically generated

Registration Page HTML

A screenshot of a computer

Description automatically generated

Login PHP

A screenshot of a computer

Description automatically generated

Login HTML

A screenshot of a computer

Description automatically generated

MySQL Table with data from Registration Form

A screenshot of a computer

Description automatically generated

## References

Connolly, R., & Hoar, R. (n.d.). Fundamentals of Web Development, 2/e. Retrieved from <https://platform.virdocs.com/>

Johari, A. (2023, March 14). How to run a PHP program in XAMPP? step by step guide. Edureka. https://www.edureka.co/blog/how-to-run-a-php-program-in-xampp/