Week 3 Assignment—Landing, Login, and Enrollment Pages Development

Maria Reyes

University of Arizona Global Campus

CST 499 Capstone for Computer Software Technology

Professor Charmelia Butler

October 11, 2023

As with any software project, careful implementation must take place. After creating UML diagrams and communicating with the team and stakeholders about requirements, implementation goes into using software programs and programming languages to construct the individual components. Those components will form part of a subsystem, and they will eventually form part of the entire software system build. This process is likewise utilized in the enrollment software system build. Within this paper, the development of the landing page, the registration page, and the login page will be discussed as well as screenshots of their scripts will be provided.

In order for this project to run scripts through the web, a certain web development environment is used. This open-source tool is called XAMPP. Programmers can use it to create dynamic websites and applications without purchasing a license. The services that this tool provides is connecting to a database platform such as PHPmyAdmin and Apache. Now, for this software build, PHP code is utilized within the Notepad++ application to develop the script. These key technologies are a way of "using multiple methods to capture the complex requirements [which] is the first step in building a complex web-based system" (Sun et al., 2003, p. 100). These methods will allow the performance of the PHP file scripts to interact with the database, PHPmyAdmin and Apache. According to Connolly & Hoar (2018), "PHP is a dynamically typed language that can be embedded directly within the HTML, and supports most common object-oriented features such as classes and inheritance" (p. 496). These technologies run smoothly with the aid of XAMPP.

In connection, it is important to explain how to run a PHP file in XAMPP. First of all, of course, is the download of the XAMPP web-development environment. During the download process, the prompt will ask which programming languages to install into the server. One of the

boxes that must be checked is PHP. Once it is installed, saving the file incurs further steps. If using a Windows operating system, the following steps will apply to create a PHP file:

1). Click on Windows (C:)

2). Click on the XAMPP folder.

3). Click on the htdocs folder.

4). Create a new folder in the htdocs folder.

5). Open the new folder and right-click to add a new text document.

6). Name the new document and end with the .php extension.

7). If the document is still referred to as a text file, then simply open the file with Notepad++ and save the file as a PHP file.

From there, the programmer has access to create the script for the landing, registration, and log in enrollment pages. It is important that the path that was used to create the files is noted because it is necessary when displaying the webpages within the localhost. The localhost enables the output of the PHP files and shows if there are any errors present in the scripts.

Furthermore, PHPmyAdmin is coded by using MySQL functions. In order for a PHP file to connect with the database, a custom connection class must be created. It acts as the "foundation program" because the "web server dynamically connects database system or file system on basis of parameters from client browser's requests" (Zhao et al., 2005, p. 3168). In the case of the enrollment system build, a separate PHP file was created to form the connection. Within the CST499 Project folder, a file was created called "connectpage.php" and it allowed for certain functions to perform by using a variety of parameters. The source code includes creating

a class called “$connect” within the PHP code. Then, the script calls for the “mysqli_connect” function, which defines the localhost, username, password, and the specific database name it should connect to since the server may have more than one database saved. It reads the following:

“$connect = mysqli_connect(‘localhost’, ‘root’, ‘’, ‘enrollmentdb’);”

This phrase sets the base for the connection process. Within the parenthesis, the username is “root”, the password is empty ‘’, and the database we are calling is “enrollmentdb”. After, an if-else statement is necessary to check if the connection was established or not. The source code that does just that is the following:

```
if($connect){
        Echo “Success”;
}else{
        Echo “Failed to connect”;}
```

The source code should enable the connection to be successful and display “success” in the web page. To check, in a web browser, the path to the connection page created in Notepad++ must be copied. For instance, the path that shows the echoed phrased being displayed from the connection file is “localhost/CST499Project/connectpage.php”.

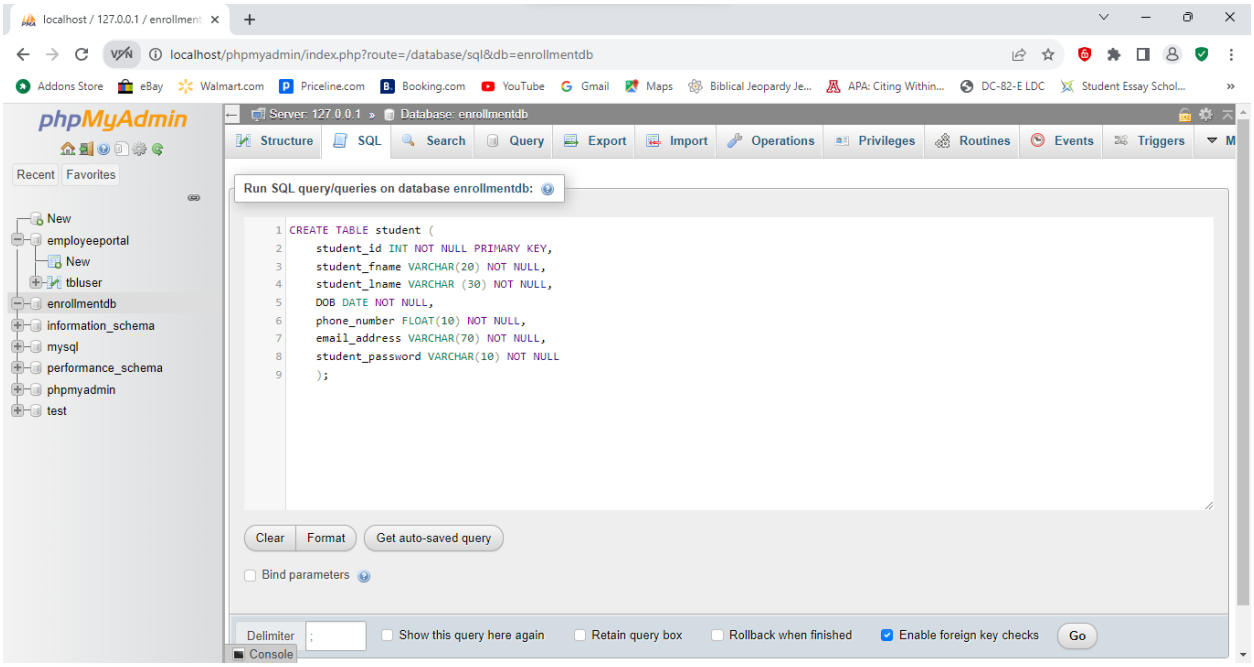
A major part of the enrollment software system build is the registration process for a new user. Once the registration PHP file is created by using the steps above, the script must be created to allow users to interact with the system. Within the HTML code to create the webpage, a form class has to be inserted. The method is POST in order for the new user to enter their information.

A division class has to be included as well in order for the rows to display in a readable format. From there, the first heading and a paragraph is added. Within the form, the labels are created for each row such as: Student ID, First Name, Last Name, Date of Birth, Phone Number, Email Address, and the user also is asked to create their own password with a minimum character count. Once all those labels are scripted along with their input class, then the code for the submit button must be created. The last step is to close the html code block. This form starts being active when the user can save their information into the database. Hence, the following will discuss the steps taken to allow this process to perform. Within the PHP code has to be activated with the "session_start" function. This form must also include the custom connection page that was created called: "connectpage.php", which will connect the registration page with the database, specifically the "student" table. Then, the server is request with an if-else statement. In between the statement, each variable has to be defined from the different fields such as "$studentid", "$student_fname", "$student_lname", "$phone_number", "$email_address", and "password", which performs with the $_POST method. Another if-statement is needed to ensure that each of those variables were not empty. Now, the global variable that actually inserts the information created by the new user into the database is "$query". Along with each student variable, the values are stated from the student table. To direct the user to the login page once they completed the registration page, the "mysqli_query" function is called with two parameters: "$connect" and "$query". The header function states that the login page is where the location is headed towards, but if one of those fields were empty, then the statement "Invalid information" is displayed from the else statement. The final task is to close the PHP code.

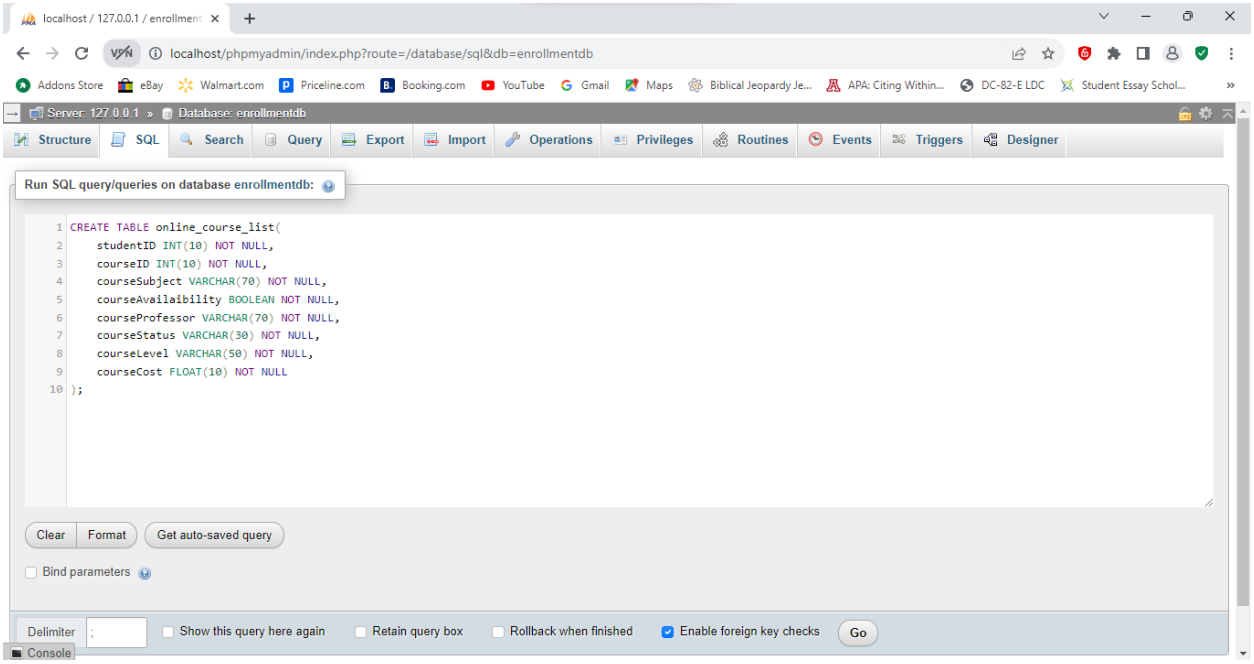
Since the requirements of the enrollment software system build was completed, the implementation became the next step. Within this paper, the skeleton of this software system was

produced. The three main pages that was created was the landing page, registration page, and login page. The other page was also vital, which was the connection page. This page is where the custom connection class was created in order to connect the PHP code with the database. It was necessary to do so because it will establish the new user to register and their information is saved in the database, where it will also verify their information when logging in. Therefore, the MySQL database functions were defined and the steps taken to create the registration page were explained. With these pages completed, the enrollment software system can begin its first steps.

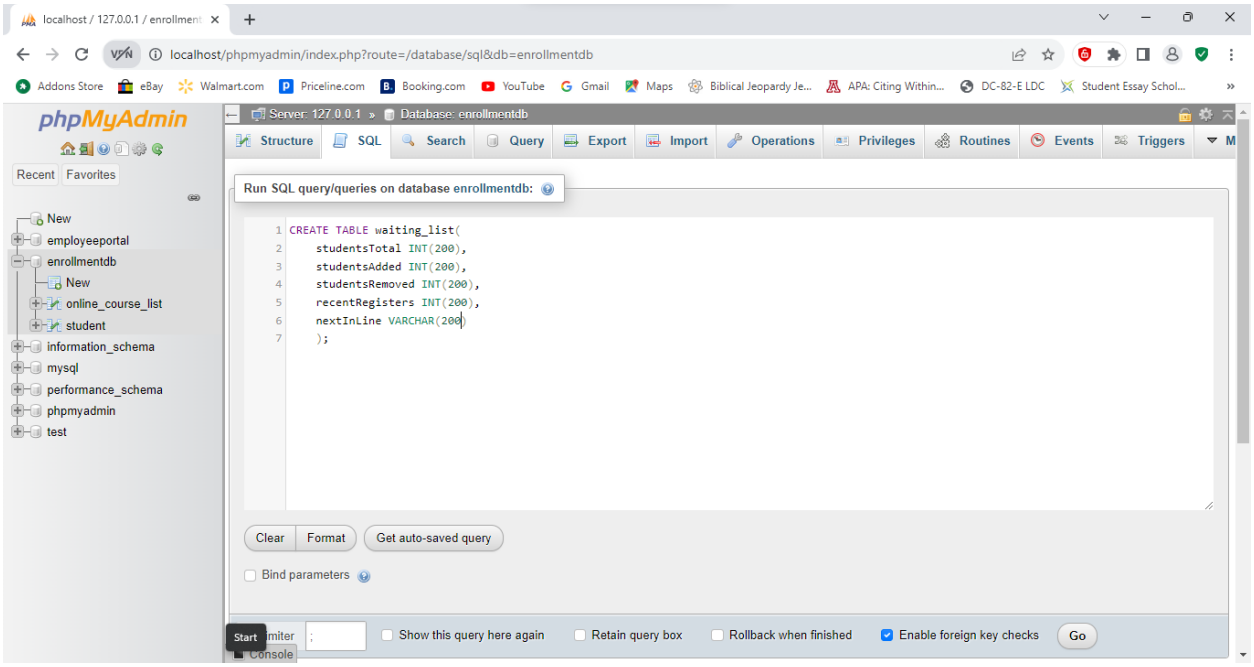
Create student table script:



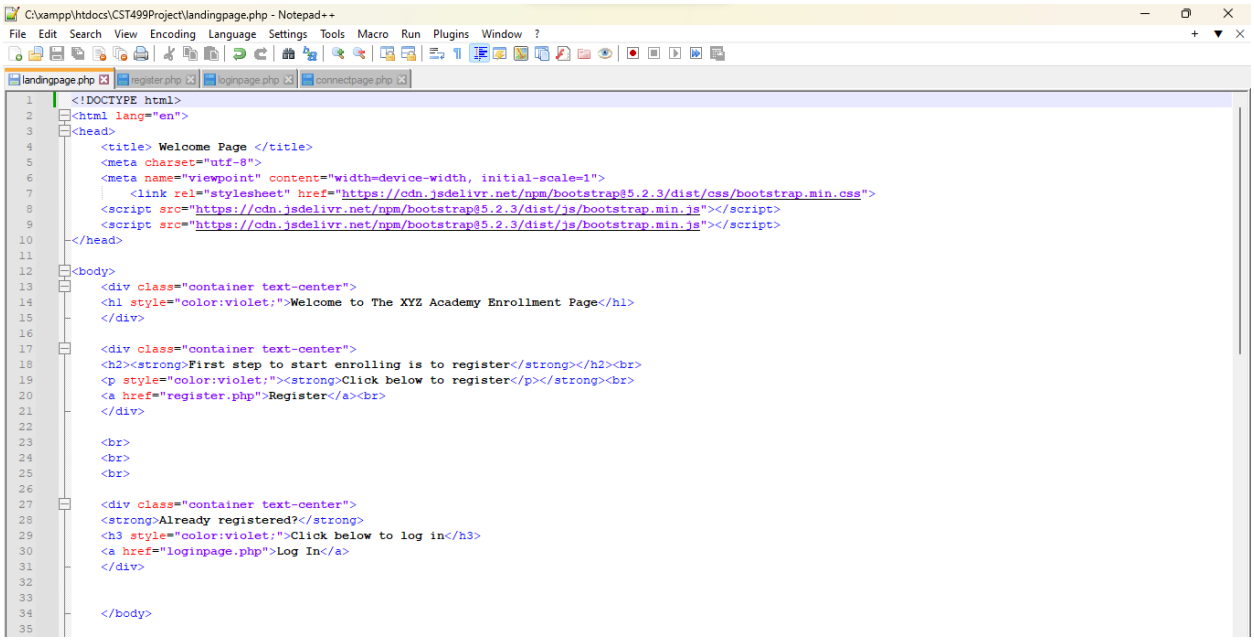
Create online_course_list table script:



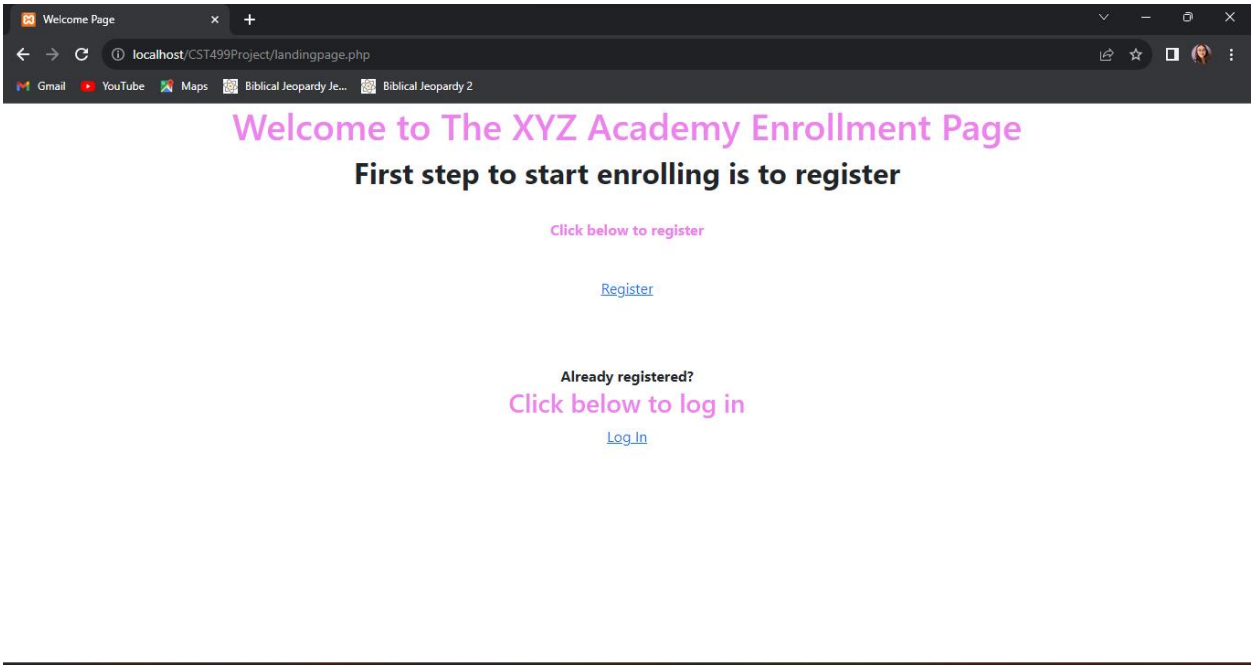
Create waiting_list table script:



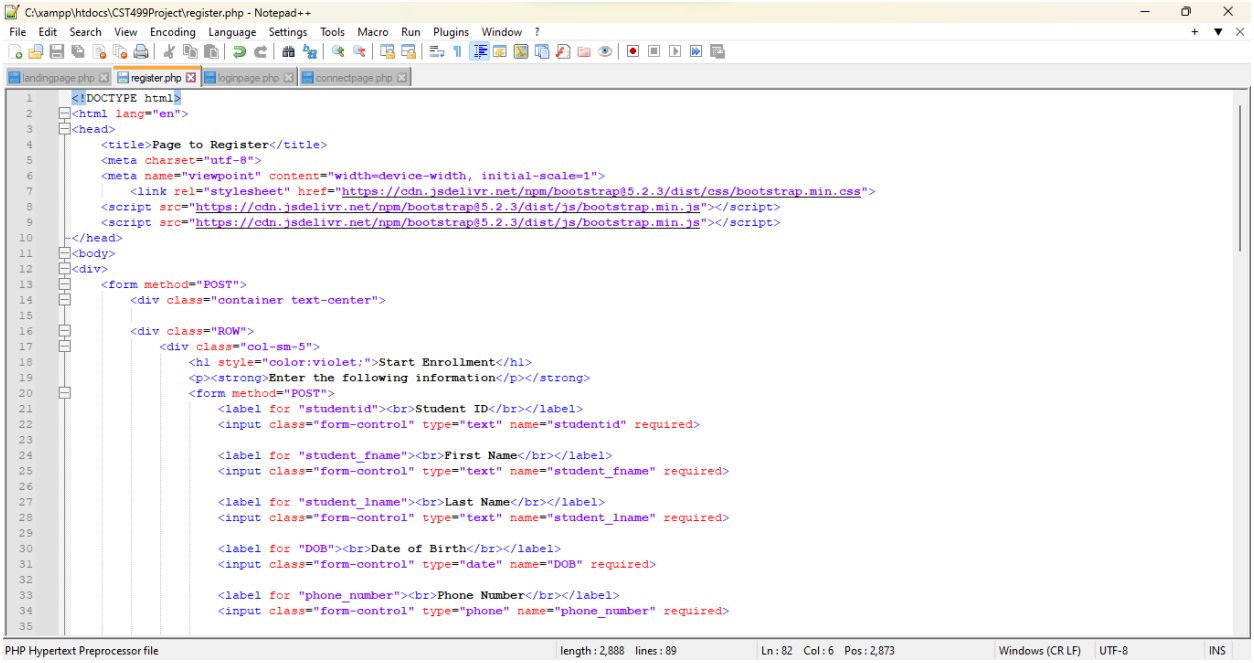
Landing Page script and output:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title> Welcome Page </title>
    <meta charset="utf-8">
    <meta name="viewpoint" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
</head>

<body>
    <div class="container text-center">
    <h1 style="color:violet;">Welcome to The XYZ Academy Enrollment Page</h1>
    </div>

    <div class="container text-center">
    <h2><strong>First step to start enrolling is to register</strong></h2><br>
    <p style="color:violet;"><strong>Click below to register</p></strong><br>
    <a href="register.php">Register</a><br>
    </div>

    <br>
    <br>
    <br>

    <div class="container text-center">
    <strong>Already registered?</strong>
    <h3 style="color:violet;">Click below to log in</h3>
    <a href="loginpage.php">Log In</a>
    </div>


    </body>
```



Registration scripts and output:



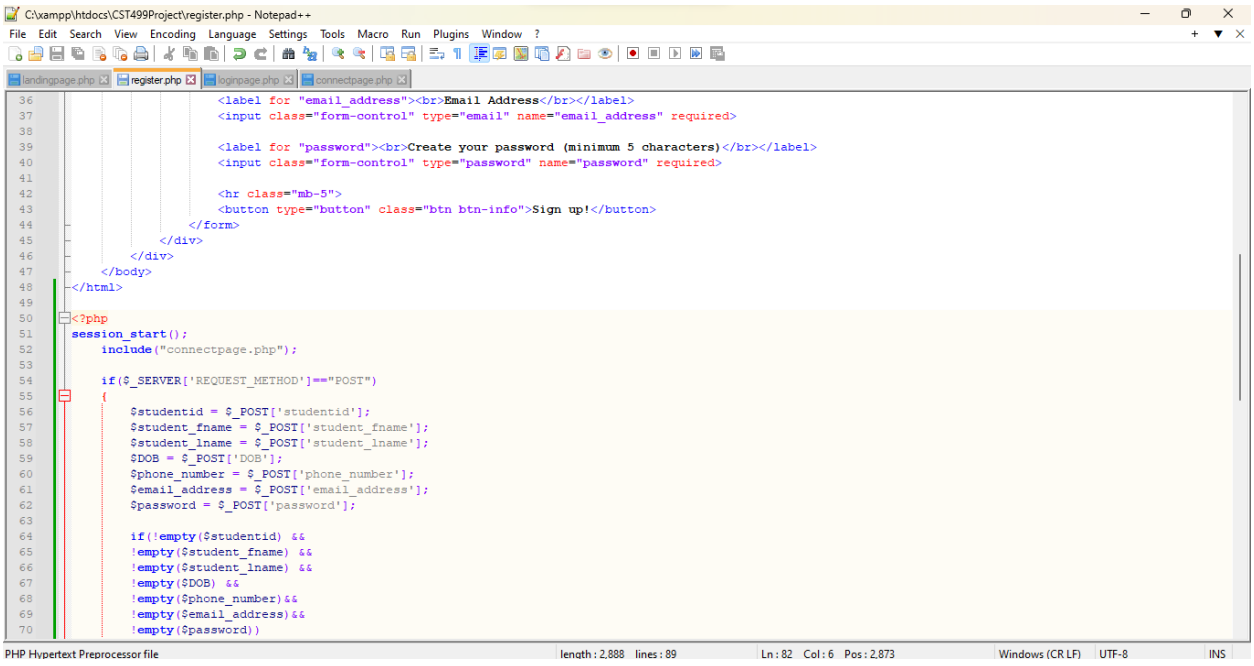
C:\xampp\htdocs\CST499Project\register.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

landingpage.php | register.php | loginpage.php | connectpage.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Page to Register</title>
    <meta charset="utf-8">
    <meta name="viewpoint" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
</head>
<body>
<div>
    <form method="POST">
        <div class="container text-center">

        <div class="ROW">
            <div class="col-sm-5">
                <h1 style="color:violet;">Start Enrollment</h1>
                <p><strong>Enter the following information</p></strong>
                <form method="POST">
                    <label for "studentid"><br>Student ID</br></label>
                    <input class="form-control" type="text" name="studentid" required>

                    <label for "student_fname"><br>First Name</br></label>
                    <input class="form-control" type="text" name="student_fname" required>

                    <label for "student_lname"><br>Last Name</br></label>
                    <input class="form-control" type="text" name="student_lname" required>

                    <label for "DOB"><br>Date of Birth</br></label>
                    <input class="form-control" type="date" name="DOB" required>

                    <label for "phone_number"><br>Phone Number</br></label>
                    <input class="form-control" type="phone" name="phone_number" required>
```

PHP Hypertext Preprocessor file | length : 2,888 lines : 89 | Ln : 82 Col : 6 Pos : 2,873 | Windows (CR LF) | UTF-8 | INS



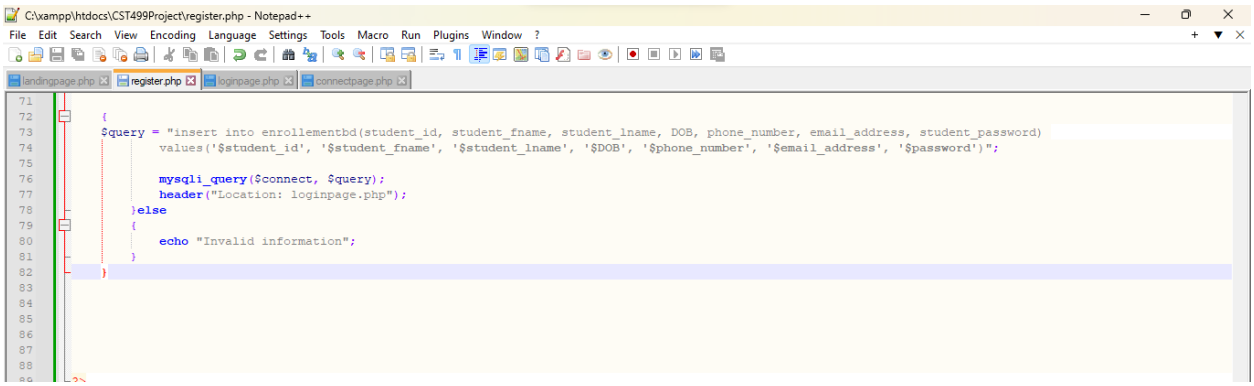
C:\xampp\htdocs\CST499Project\register.php - Notepad++

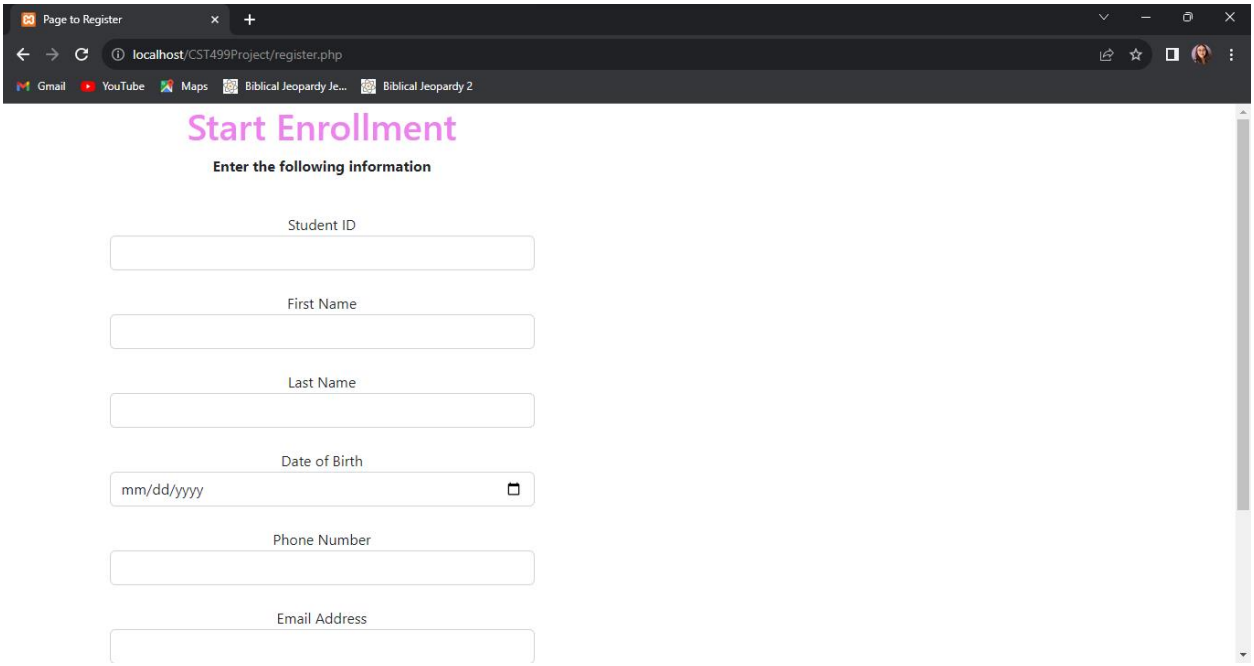
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

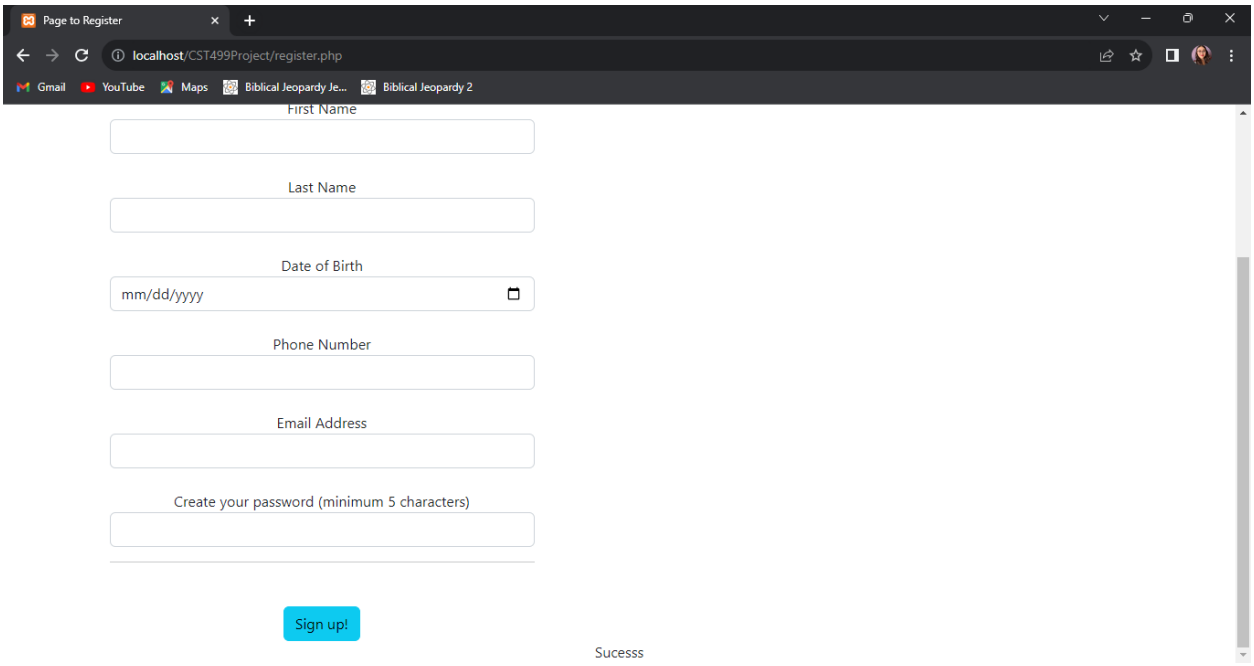
landingpage.php | register.php | loginpage.php | connectpage.php

```
                    <label for "email_address"><br>Email Address</br></label>
                    <input class="form-control" type="email" name="email_address" required>

                    <label for "password"><br>Create your password (minimum 5 characters)</br></label>
                    <input class="form-control" type="password" name="password" required>

                    <hr class="mb-5">
                    <button type="button" class="btn btn-info">Sign up!</button>
                </form>
            </div>
        </div>
    </body>
</html>

<?php
session_start();
    include("connectpage.php");

    if($_SERVER['REQUEST_METHOD']=="POST")
    {
        $studentid = $_POST['studentid'];
        $student_fname = $_POST['student_fname'];
        $student_lname = $_POST['student_lname'];
        $DOB = $_POST['DOB'];
        $phone_number = $_POST['phone_number'];
        $email_address = $_POST['email_address'];
        $password = $_POST['password'];

        if(!empty($studentid) &&
        !empty($student_fname) &&
        !empty($student_lname) &&
        !empty($DOB) &&
        !empty($phone_number)&&
        !empty($email_address)&&
        !empty($password))
```

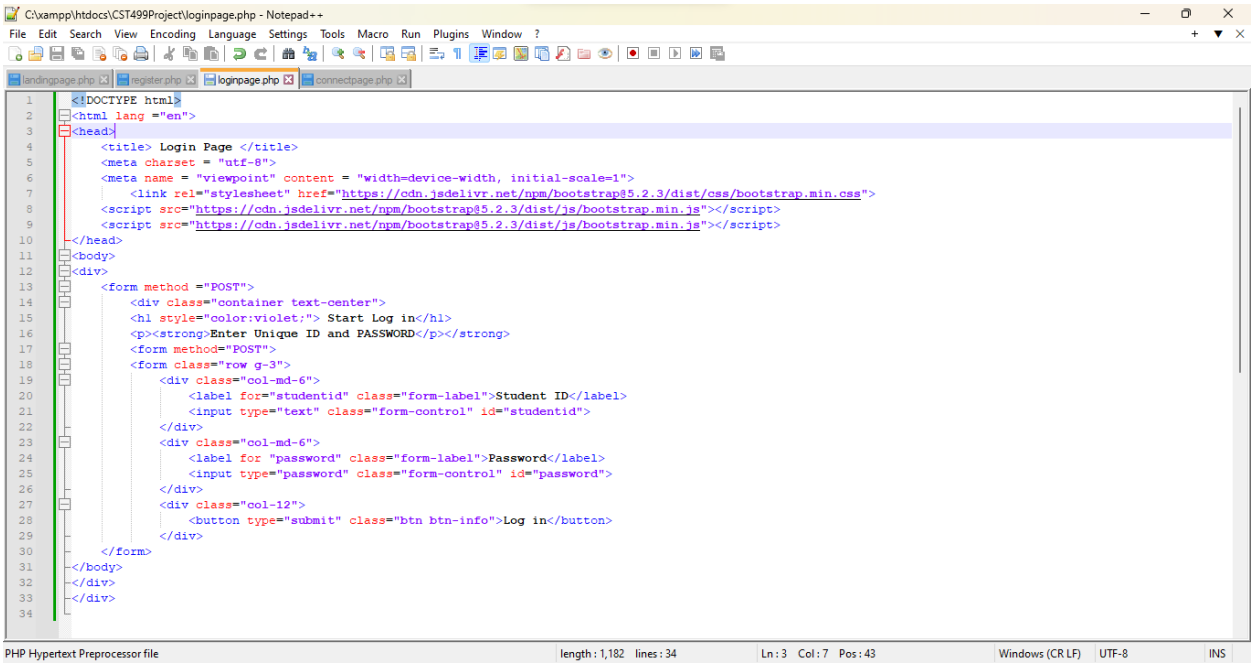
PHP Hypertext Preprocessor file | length : 2,888 lines : 89 | Ln : 82 Col : 6 Pos : 2,873 | Windows (CR LF) | UTF-8 | INS

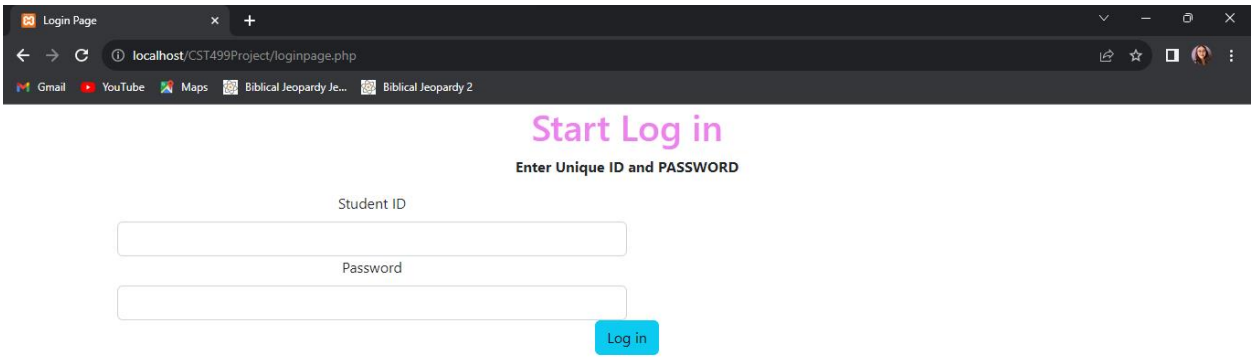

C:\xampp\htdocs\CST499Project\register.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
landingpage.php
register.php
loginpage.php
connectpage.php
{
$query = "insert into enrollementbd(student_id, student_fname, student_lname, DOB, phone_number, email_address, student_password)
values('$student_id', '$student_fname', '$student_lname', '$DOB', '$phone_number', '$email_address', '$password')";
mysqli_query($connect, $query);
header("Location: loginpage.php");
}else
{
echo "Invalid information";
}
}


Page to Register
localhost/CST499Project/register.php
Gmail
YouTube
Maps
Biblical Jeopardy Je...
Biblical Jeopardy 2
Start Enrollment
Enter the following information
Student ID
First Name
Last Name
Date of Birth
mm/dd/yyyy
Phone Number
Email Address

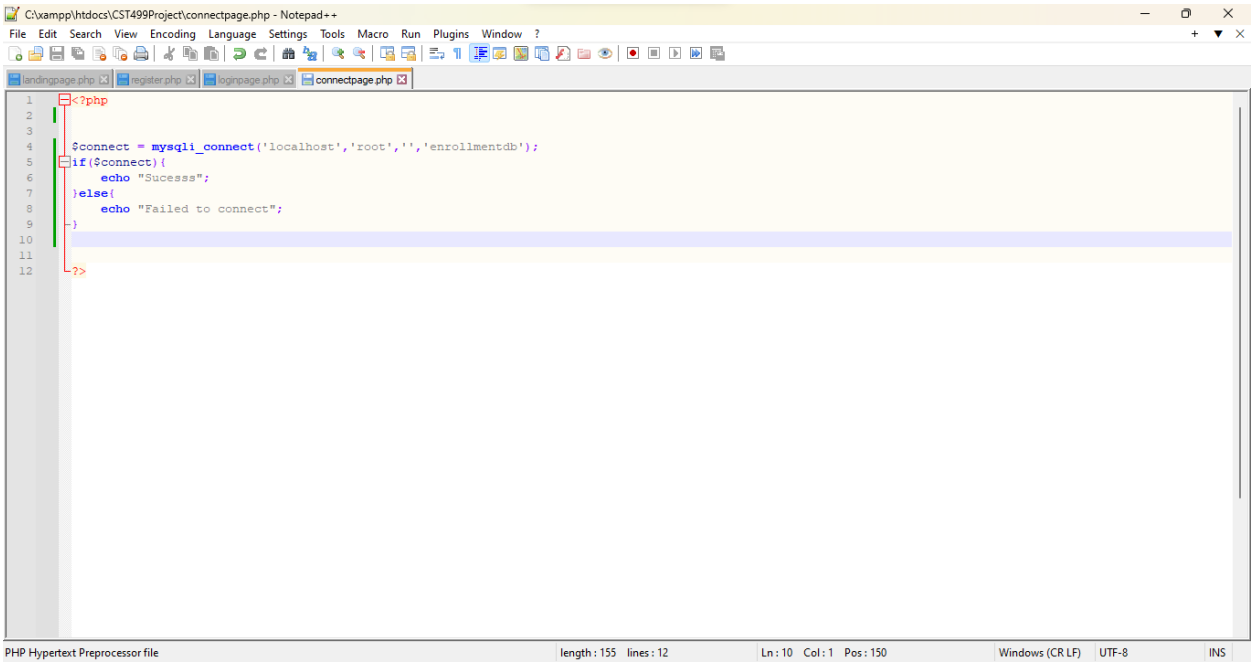


Log in script and output:





Connection page script:



References

Connolly, R., & Hoar, R. (2018). *Fundamentals of web development* (2nd ed.). Pearson.

Sun, D., Wong, K., & Moise, D. (2003). Lessons learned in Web site architectures for public utilities. *Proceedings Fifth IEEE International Workshop on Web Site Evolution, 2003. Theme: Architecture*, 93–100. https://doi.org/10.1109/WSE.2003.1234013

Zhao, F. J., Zhang, J. F., & Cao, D. Y. (2005). Dynamic database connection and dynamic web map service for internet mapping. *Proceedings 2005 IEEE International Geoscience & Remote Sensing Symposium, 2005 (IGARSS '05)*, 3167–3169. https://doi.org/10.1109/IGARSS.2005.1526512