# **Week 5 – Compilation.**

# **Software Project.**

Danny Escalante

University of Arizona Global Campus

CST-499: Capstone for Computer Software Technology

Instructor: Joseph Rangitsch

May, 13th 2024.

**Project Test Plan.**

The main objective of a test plan for an online class registration project is to guarantee the successful implementation of a dependable, secure, and user-friendly software solution that streamlines class registration processes. Below I break this project into a Test Plan Template (TPT) based on the Guru99 Sample Document Example.

1. **Introduction.**

This test plan aims to delineate the testing strategy, scope, resources, and timeline for the online class registration software.

* 1. Scope.

This test plan includes the functional and non-functional testing of the online class registration software, encompassing the registration process, user authentication, class selection, payment processing, and system performance.

* + 1. In Scope.
       1. Functional and non-functional testing of the online class registration software.
       2. Registration process, user authentication, class selection, payment processing, and system performance.
    2. Out of Scope.
       1. Any features not related to class registration.
       2. Third-party integrations beyond payment processing.
  1. Quality Objective.

The objective is to ensure that the online class registration software meets the following quality standards: Functional correctness, reliability, performance efficiency, and security.

* 1. Roles and Responsibilities.
* Project Manager: Approve test plan, allocate resources, and monitor progress.
* Development Team Lead: Provide necessary support and resources for testing.
* Quality Assurance Lead: Develop test cases, execute tests, and generate test reports.
  1. Exit Criteria.

Outlines the conditions that must be met for testing to be considered complete (Geeksforgeeks, 2023). For functional testing, this includes ensuring critical functionalities like user registration and payment processing work without major defects, with high-priority defects either resolved or documented with acceptable workarounds.

1. **Test Methodology.**
   1. Overview.

This test plan follows a systematic approach to testing, covering multiple test levels and types to ensure comprehensive coverage.

* 1. Test Levels
* Unit testing and integration testing
* System performance testing
* Acceptance and usability testing
  1. Bug Triage.

Bugs will be categorized based on severity and priority to facilitate efficient resolution.

* 1. Suspension Criteria and Resumption Requirements.
* Testing will be suspended if critical defects hinder testing progress.
* Testing will resume once the defects are resolved.
  1. Test Estimated Effort.

Testing will be considered complete when all test cases have been executed, and the acceptance criteria have been met. The estimated effort section allocates resources and provides a breakdown of the time required for various testing activities. This helps in planning and budgeting for testing activities throughout the project lifecycle.

1. **Test Deliverables.**

Include all of the necessary records and reports produced during the testing process, such as the test plan that describes the testing strategy, parameters, and goals. Deliverables play a crucial role in monitoring the progress of testing, guaranteeing comprehensive test coverage, and promoting efficient communication among project participants.

1. **Resource & Environment Needs**
   1. Incident Report Strategy.

Outlines how defects or incidents discovered during testing are reported, categorized, and resolved (Tsui, et. al., 2018). Incidents are classified based on severity and regular status updates are provided until incidents are resolved, and retesting ensures satisfactory resolution before closure, maintaining the quality of the software product.

* 1. Testing Tools.
* JMeter(performance) and OWASP ZAP (security).
  1. Test Environment.
* Desktops, laptops for client-side testing, and servers with sufficient processing power and memory for server-side testing.
* Operating systems: Windows, macOS, Linux.
* Browsers: Chrome, Firefox, Safari, and Edge based on student’s preferences.

The online class registration software's effective development and implementation depend on several benefits that come with using this project test plan. The strategy makes sure that all functional and non-functional software features are thoroughly tested by methodically defining the test objectives, methodology, environment, and resources. The strategy enables the timely identification and resolution of faults by using stringent testing procedures at several stages, including unit, integration, system, and acceptance testing.

**UML DIAGRAMS.**

An online course system's structure, behavior, and relationships can be represented standardized and methodically using UML (Unified Modeling Language), by breaking down complex concepts into manageable modules and providing interactive resources like video lectures, tests, and hands-on exercises, students can gain a firm grasp of UML diagrams at their own pace. Furthermore, the flexibility of the online format lets people fit learning around their current obligations.

**Class diagrams** provide a structured representation of system syntax, and important components like classes, attributes, and methods. Providing class structures, relationships, and inheritance hierarchies clearly and concisely using a combination of interactive tutorials, instructional films, and hands-on exercises (TutorialsPoint, n.d.).

Diagram

Description automatically generated

**Sequence Diagrams** provide a sequential representation of the messages and interactions between objects, which facilitates the analysis, design, and communication of intricate software systems. (Padmanabhan, 2012).

Diagram

Description automatically generated with medium confidence

**State diagrams** symbolize a system's various states and the circumstances that cause these states to change. State diagram notation and symbols, along with methods for handling concurrent states and capturing complex behaviors, are essential in an online course. (Tsui, et. al., 2018).

Diagram

Description automatically generated with medium confidence

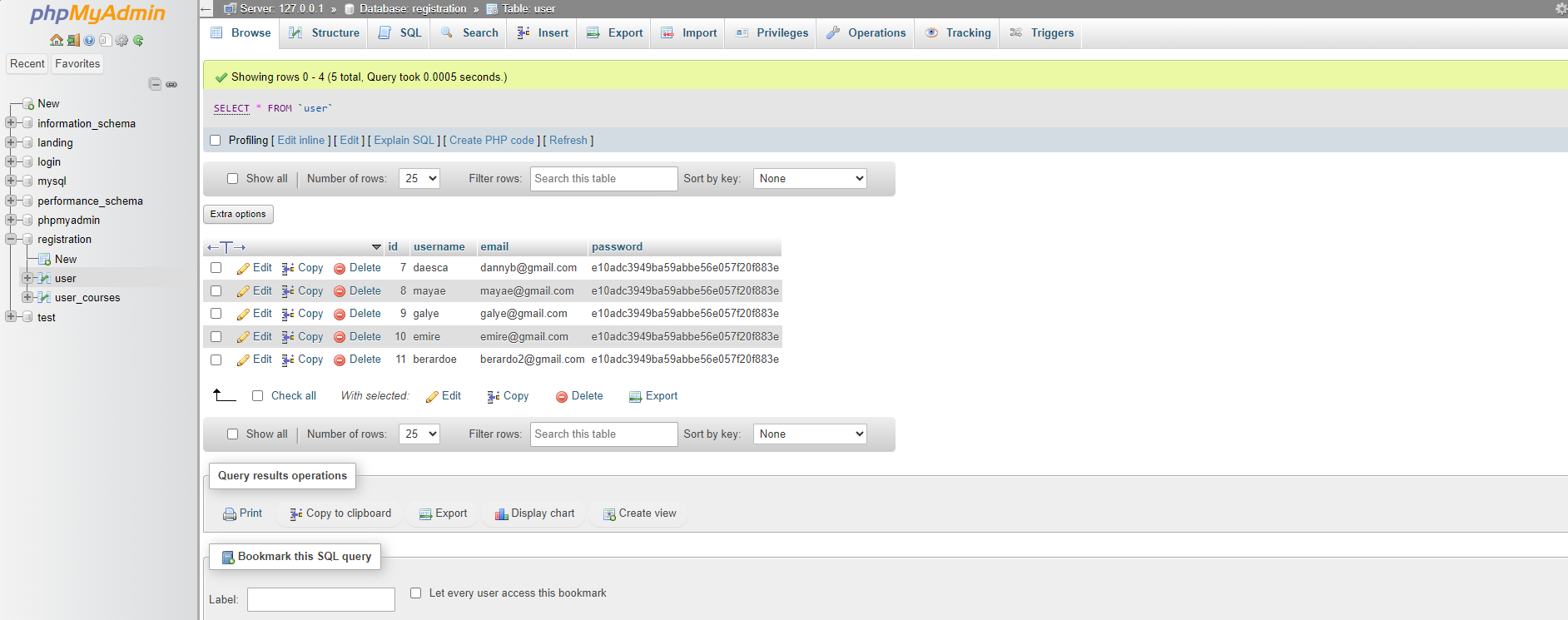
**User Case diagrams**, serve as a vital tool for users to interact with a system and are an essential teaching software engineering topic. (Spillner & Schaefer, 2014). Here is a representation of this UML diagram:

Diagram

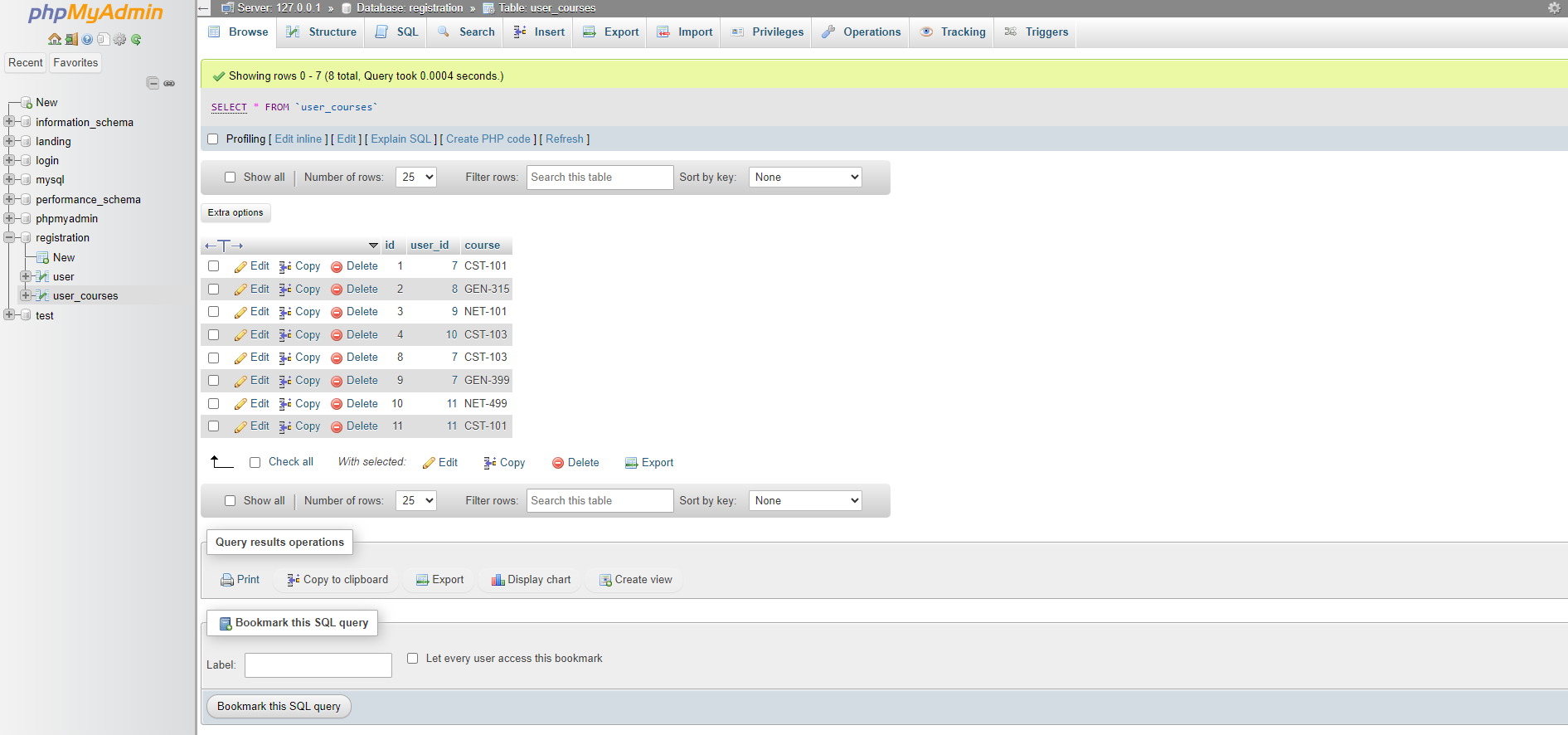
Description automatically generated

**Online Classes Enrollment PHP XAMP.**

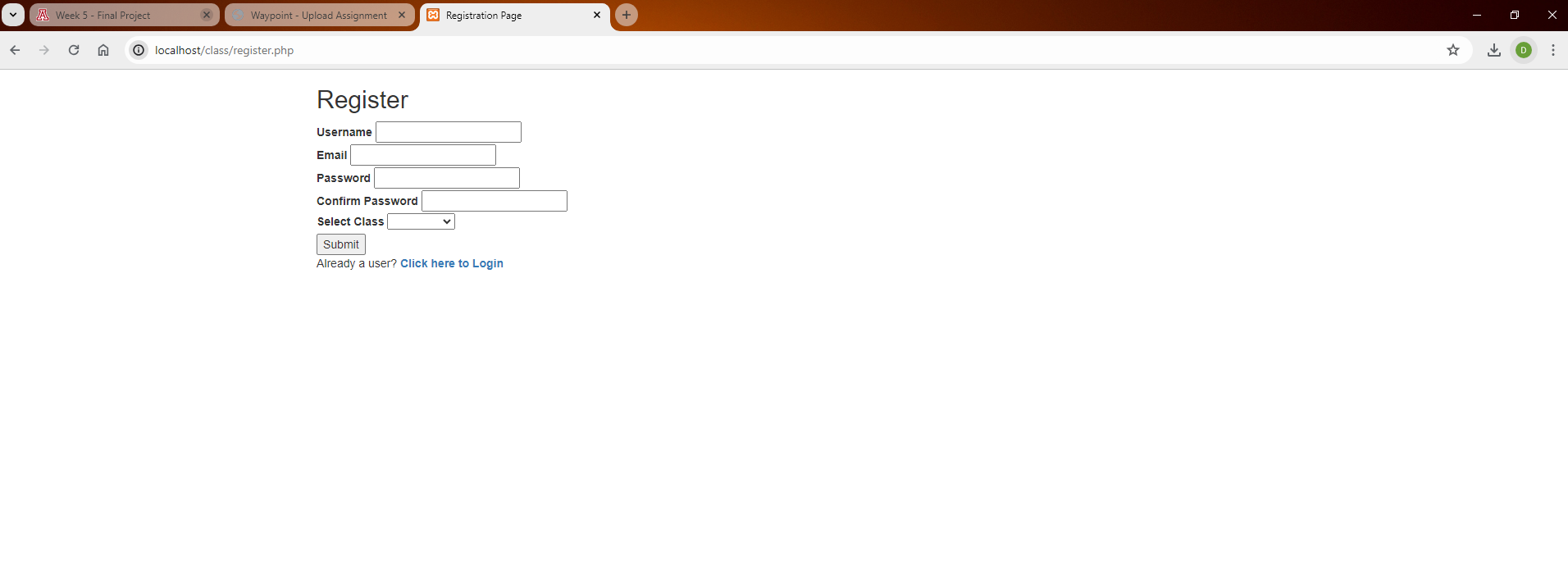
phpMyAdmin – User database:

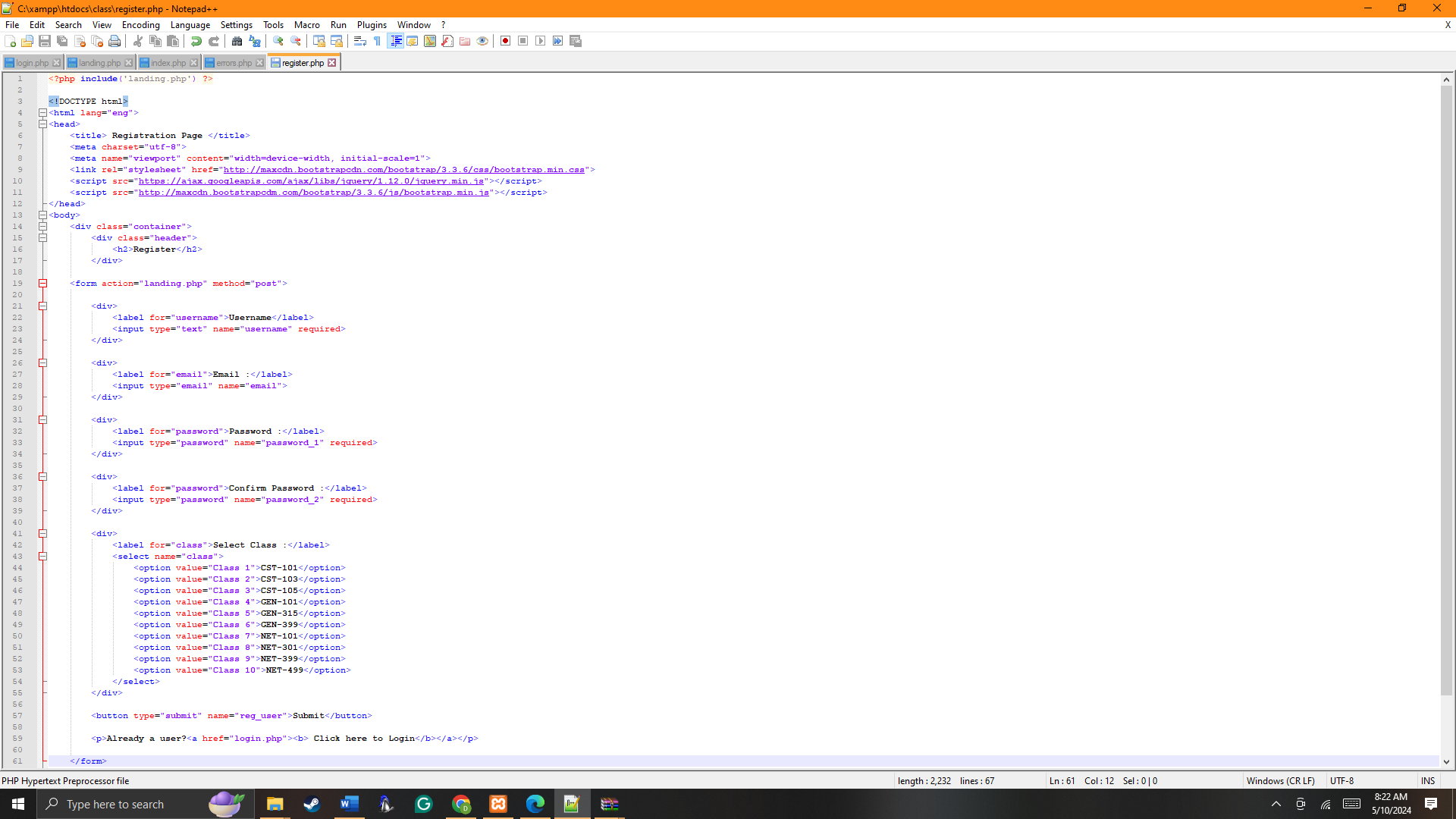


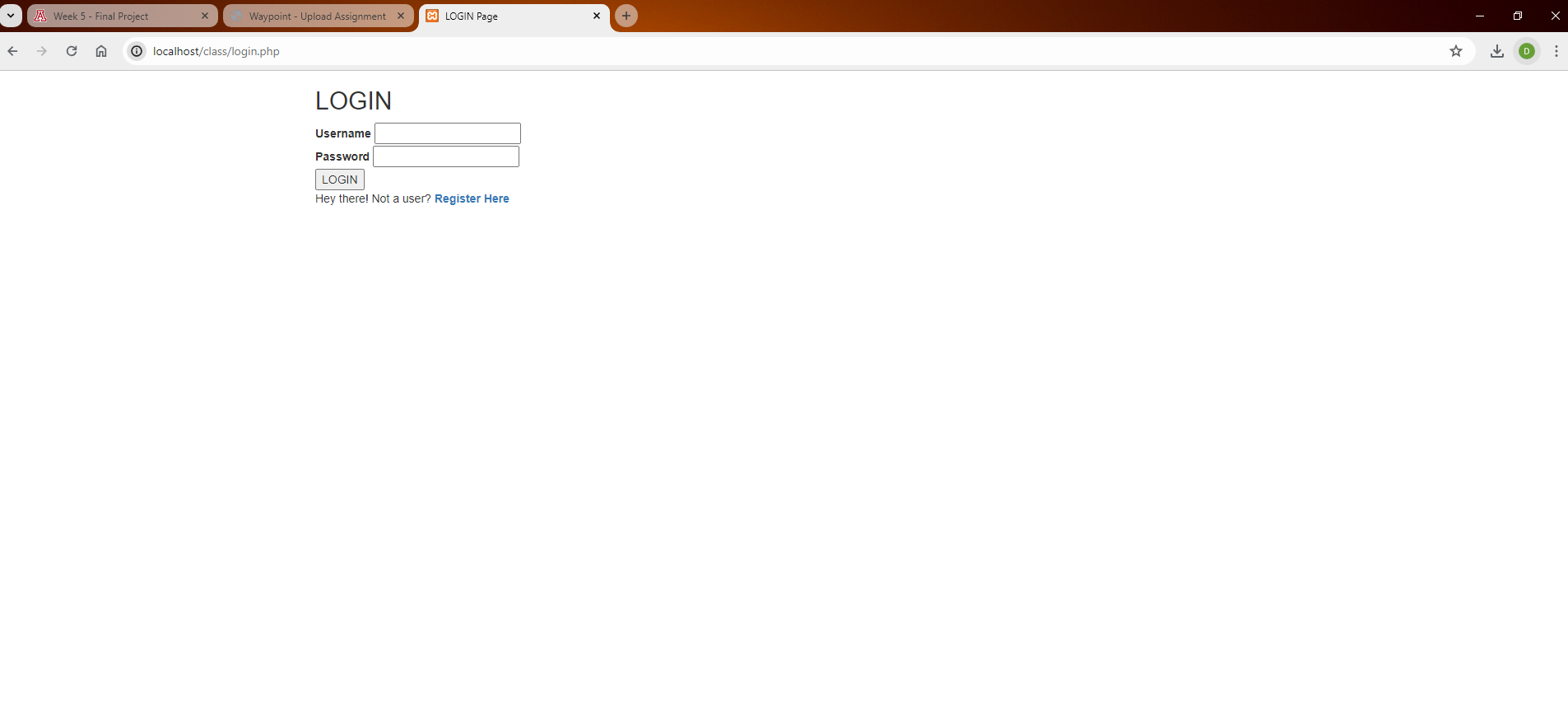
phpMyAdmin – User\_courses database:

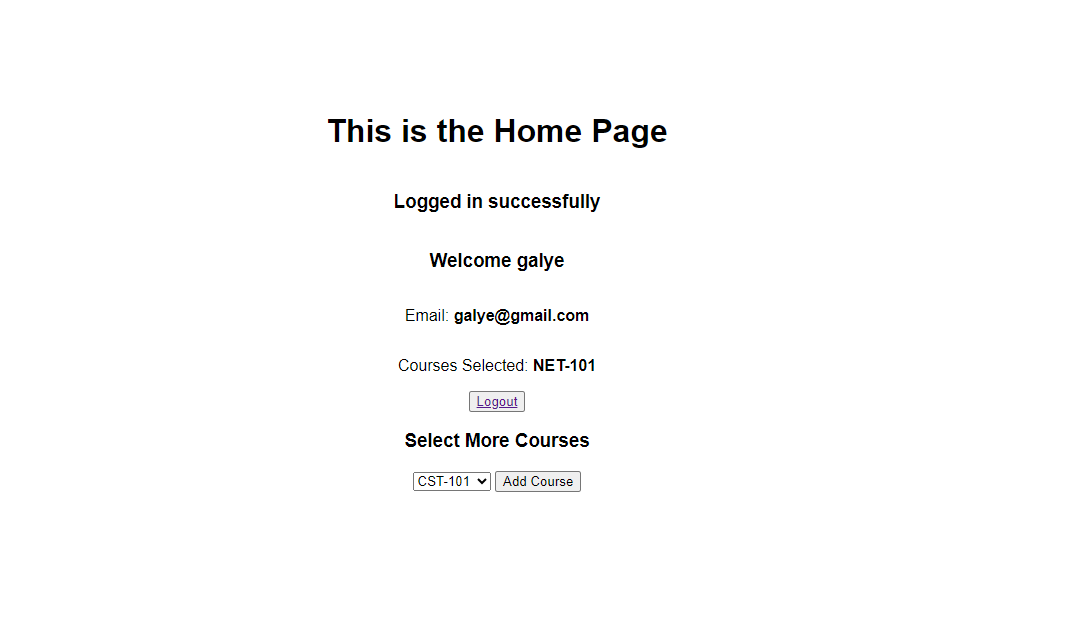


**Register.php**

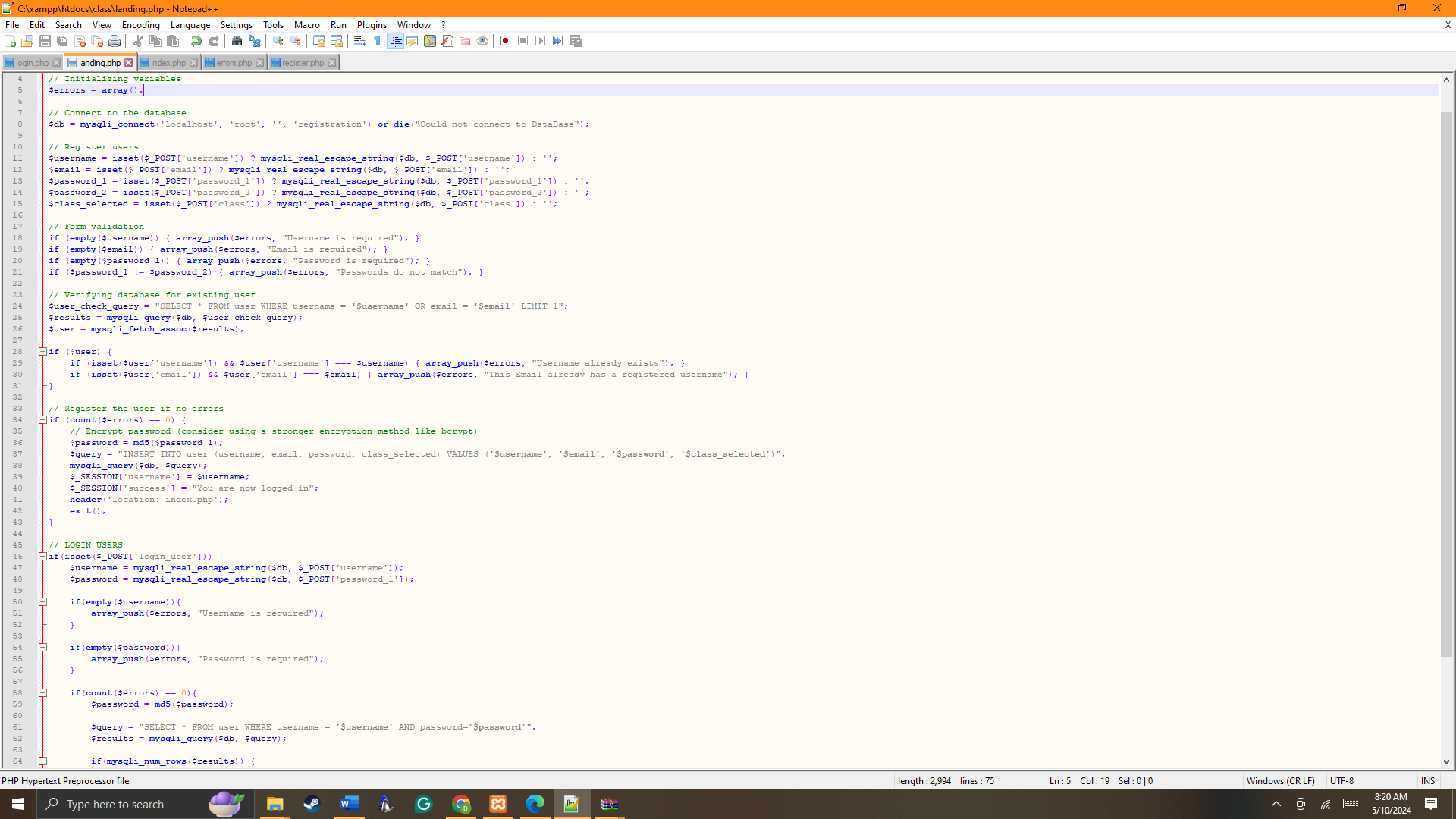




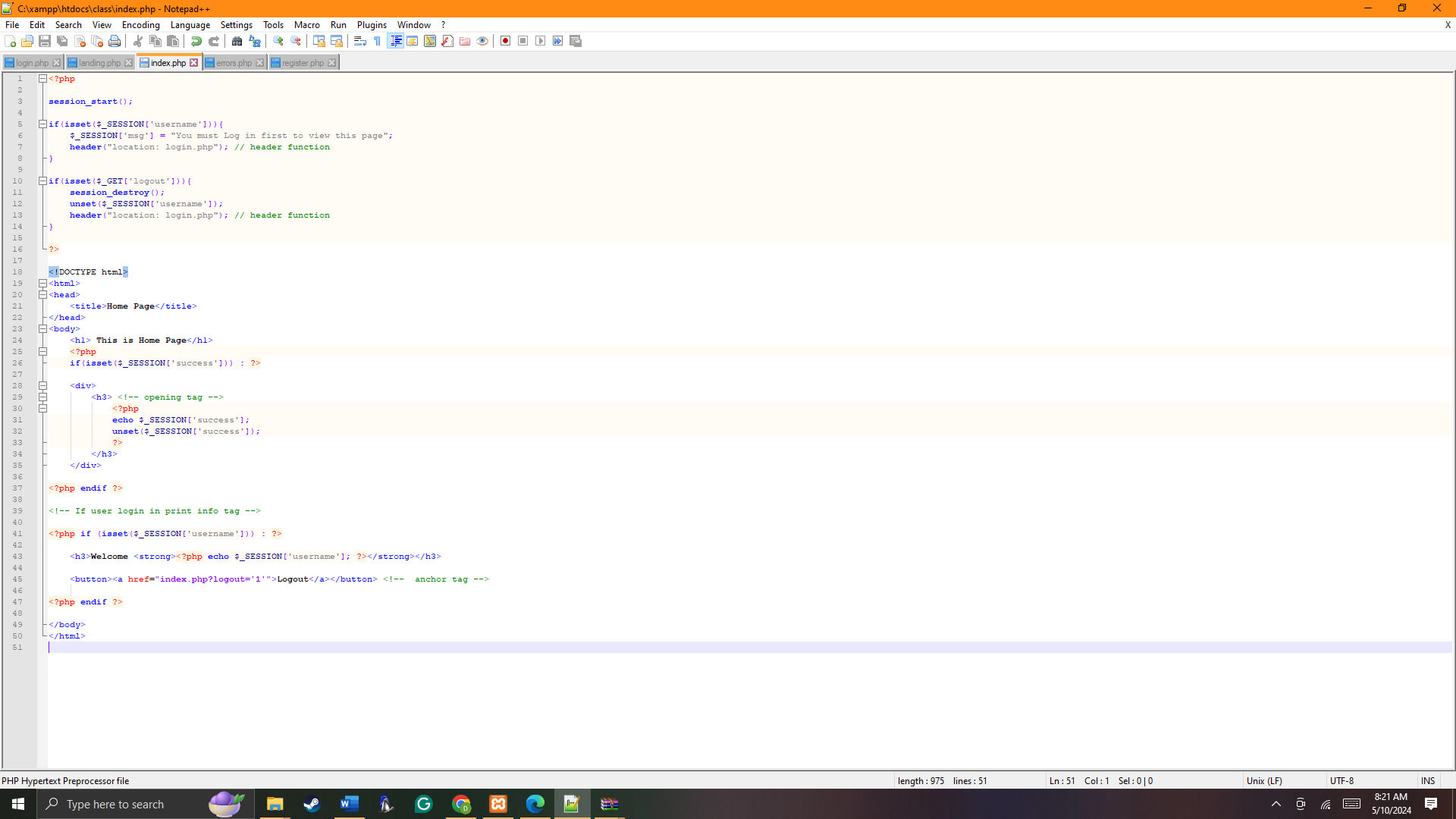
Login.php: 



Landing.php



**Index.php**



**Running the PHP file in XAMPP.**

We can run PHP (Hypertext Protocol) files using “.php” as the file extension. In this case, I used Notepad++ to write the Login, Registration, and Landing Website code. Once the XAMPP is installed, run the Apache and MYSQL program, and then click on “Admin”. A site will open up as “localhost” or type “http://localhost”. All the “.php” files in the XAMPP were saved in the default directory “C:\xampp\htdocs”. Lastly, start Apache admin.

MySQL databases in PHPMyAdmin, I utilized various functions like creating databases, and tables and executing queries. To connect to a MySQL database in PHP, I followed these steps: first, I defined a class with functions for connecting to the database, executing queries, fetching results, and closing the connection (Conolly, et. al., 2018). Then, I implemented the connection function using PHP’s MySQL extension, specifying the hostname, username, password, and database name ensuring secure and efficient communication with the MySQL database.

To create a registration page in PHPMyAdmin, I followed these steps: First, I designed the registration form using HTML. Then, I created a PHP script to handle form submission. Within this script, I established a connection to the MySQL database. Next, I executed an SQL INSERT query to store the user’s information in the database table designated for user registrations. Finally, I verified the code, to identify errors encountered during the process, ensuring a smooth and user-friendly registration experience (Mikoluk, 2013).

**Implementing a PHP file in XAMPP.**

The supplied database development class code makes it easier for users to register and log in to a website. A username, email address, and password can be entered throughout the registration procedure. There's also a dropdown menu from which they can choose a class. When the form is submitted, the information is transferred to the server, where it is checked to make sure the passwords match and all fields are filled in. The user's data is added to a MySQL database with the chosen class if validation succeeds. In the login.php file, a Username and password are required to access the site Overall, the code helped me find lots of information and support in the current course’s resources helped me to experience coding registration with class selection, and secure login functionality for an online class.

**References:**

* Jones & Bartlett Learning.W3schools (n.d.). HTML Element References. Retrieved from: <https://www.w3schools.com/TAGS/default.asp>
* Mikoluk, K. (2013, September 18). XAMPP tutorial: How to use XAMPP to run your web server. Udemy.
* Padmanabhan B. (2012). Unified Modeling Language (UML) Overview. Retrieved from: <https://people.eecs.ku.edu/~hossein/810/Readings/UML-diagrams.pdf>
* Spillner, A., Linz, T., & Schaefer, H. (2014). Software testing foundations: A study guide for the certified tester exam (4th ed.). Rocky Nook.
* TutorialsPoint. (n.d.). UML - Model. <https://www.tutorialspoint.com/uml/uml_modeling_types.htm>
* Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.).