**Group 42**

Assessment: Assignment 2

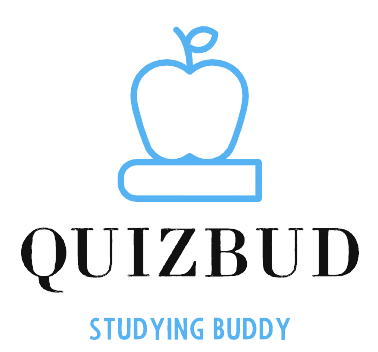
Student Names: **Jordan Chen (041050455), Kyle Ryc (040778889), Renato Simoes (040908766)**

Lab Professor Name: Hala Own

Lab Section Number: 303

Due Date: Sunday, December 4, 2022

# Project Overview



*Logo design*

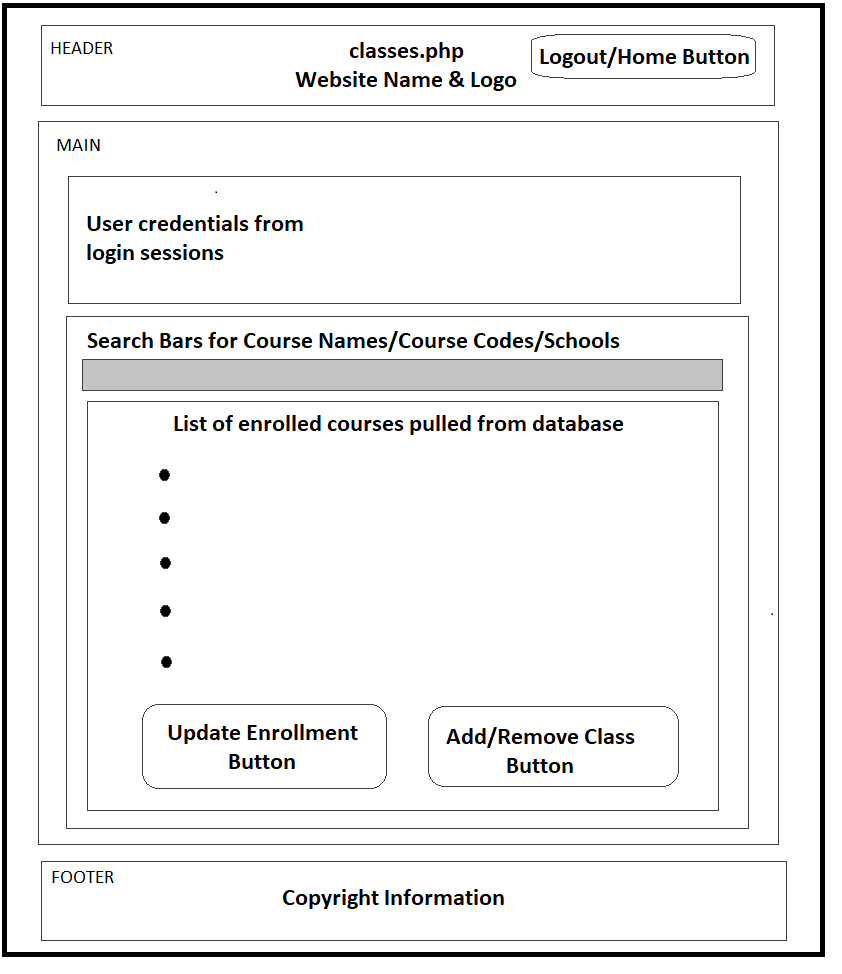
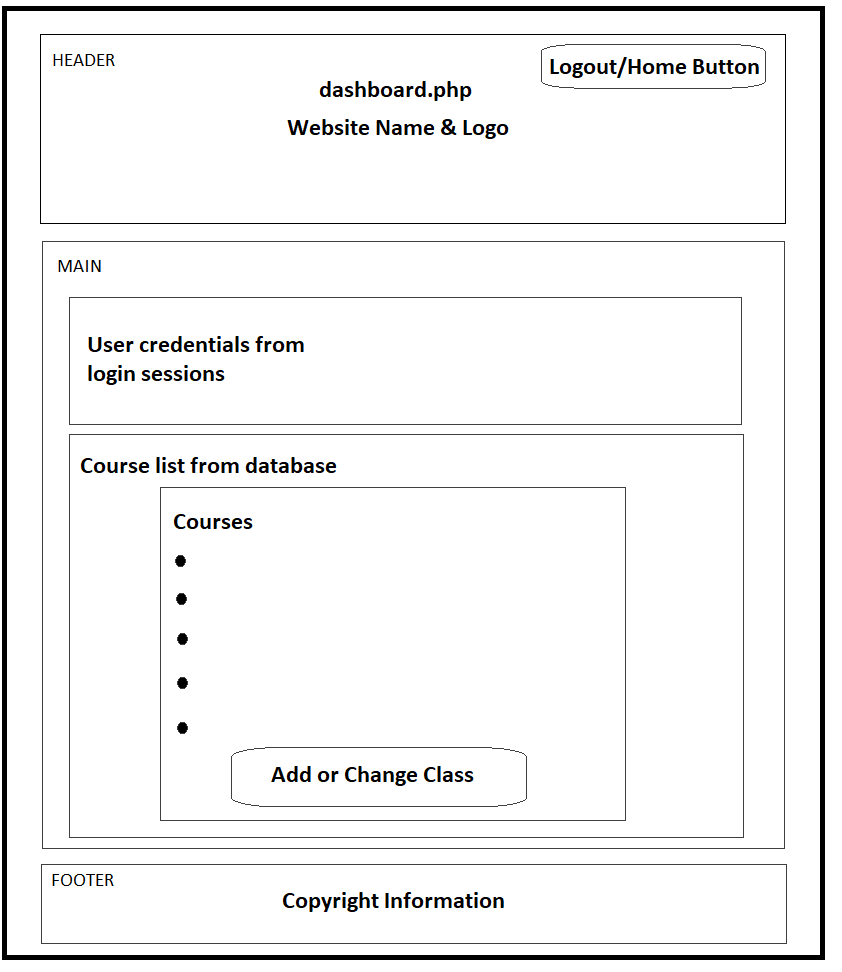
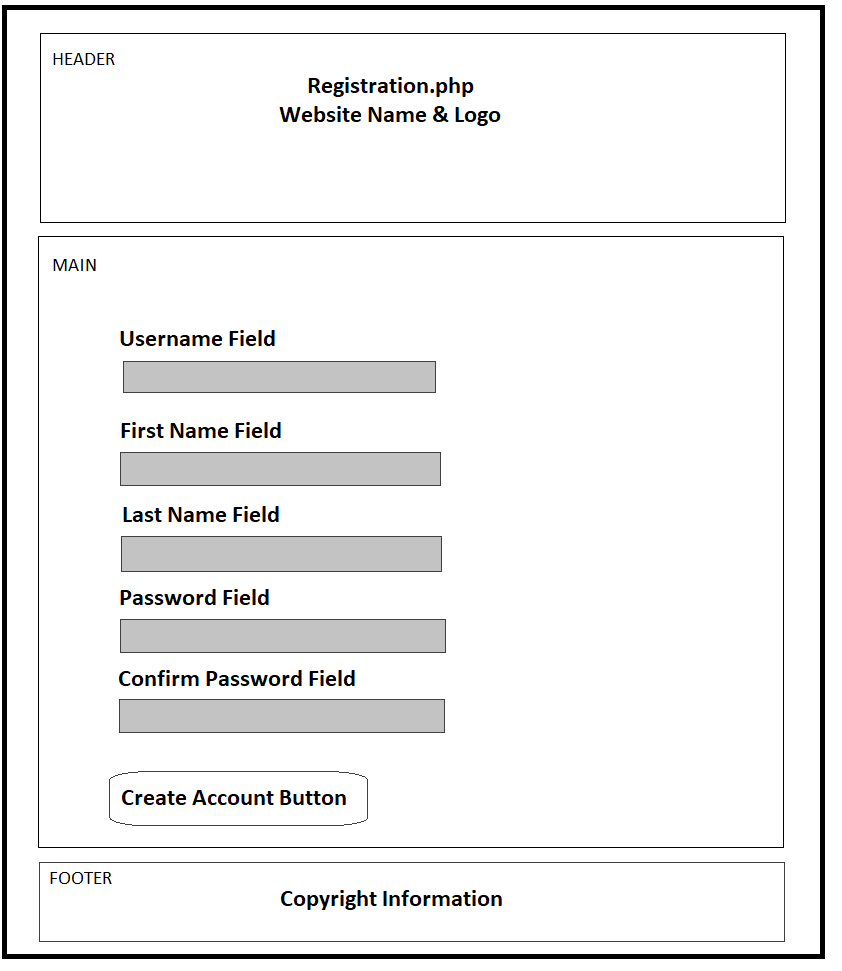
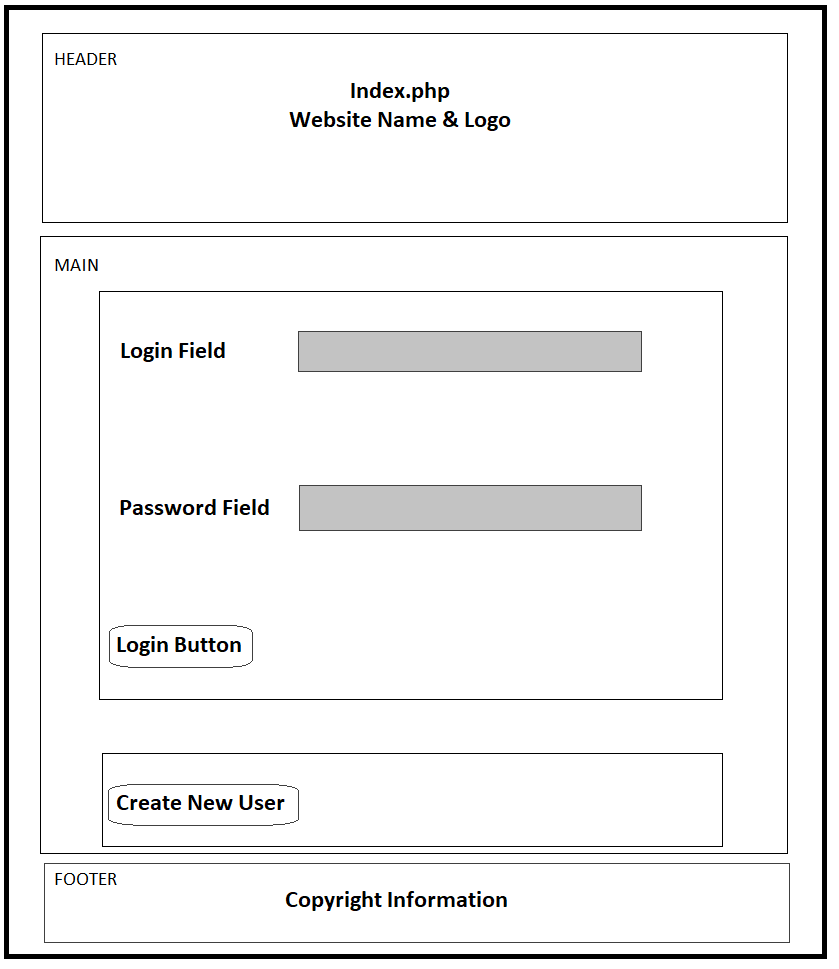
QuizBud is a website where users can create quizzes to help them develop their knowledge and study. It is tailored with college and university students in mind. The platform allows the registered users to register their courses and submit questions, which are stored in a database, which can later be accessed to generate a quiz. As it is collaborative in nature, a plural number of students can contribute to the quiz database, which can be filtered by class, thus increasing its efficacy as a study tool.

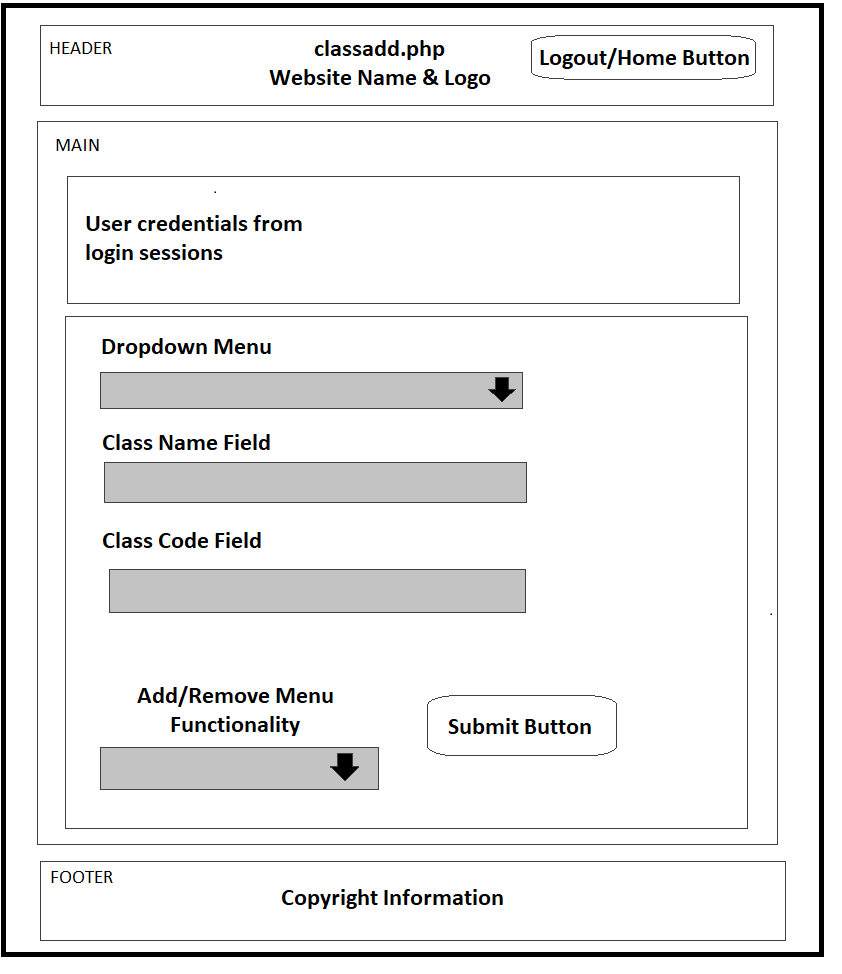
# Web Map

A picture containing diagram

Description automatically generated

# Wireframe





# Client-side Design (HTML/CSS)

## Make Question – makequestion.html

Form where the student can create a question to be added to the database, integrated via PHP

## Generate Quiz – generatequiz.html

Form where the student can generate a quiz from the database, integrated via PHP

## Created Quiz – createdquiz.html

Form where the student can access the generated quiz from the database, integrated via PHP and JavaScript

## Styles – styles.css

Single cascading style sheet responsible for determining the overall design of the website, used across the multiple HTML and PHP files

# Client-side Scripting (JavaScript)

## Validation – validation.js

Script responsible for validating the login process of the platform

## Registration Script – regscript.js

Script responsible for validating the user creation process of the platform

## Table Sorter – tableSorter.js

Clicking on the headings of the ClassList table allows you to sort each column alphabetically.

## Table Filter – tableSorter.js

In addition, the filtering fields hid any rows where the contents of the column do not contain the search query.

Script used to sort an internal HTML table

# Server-side Components (PHP/SQL) – Functionality Guide

## User Login – index.php

After form validation is completed via JavaScript, the student is either checked against the *Users* table to registration.php to verify if they are present. As this is a demonstration database, the passwords are not encrypted. If everything is good, the user’s fields are stored as a new user or to$\_SESSION variables, and then they are redirected to the dashboard.php if they have an existing account

## User Registration - registration.php

After form validation is completed via JavaScript, the student is entered into the *Users* table. If everything is good, the user’s fields are stored as $\_SESSION variables, and then they are redirected to the dashboard.

## Header and Footer

For the follow pages, PHP is used to create the information that is kept in a separate Header and Footer file. This includes posting the user’s name and ID # at the top of the page, from $\_SESSION variables.

## User Classes - dashboard.php & mycourses.php

Shows a list of all currently registered classes via an SQL query connecting *Users* with *Classes* via *Courses*, this is packaged in mycourses.php.

## Enrollment Changes – classes.php

Pulls a list of all available from *Classes*, showing a school name via *Schools*, and then shows a checkbox marked enabled for those the user has already signed up as determined by *Courses*. By changing the checkboxes and hitting submit, the user changes what classes they are enrolled in.

## Class List Changes – classadder.php

A simple form that allows the user to enter a class and either remove it from the currently available list, or add it. Results are pushed as insertions or deletions into the *Classes* table, cascading deletions in *Courses*.

## Connecting to the SQL Server – serverconnect.php

As seen above, it is very common to need to connect to the MySQL, so we’ve packaged up the connection script and returned the connection as a PHP variable $con, which you may encounter on basically every page.

## Session Ending and Logout – userLogout.php

In order to kill the current user session, the Logout button in the Header is configured to call a session\_destroy(), and redirect the user to the Index.php page.

# Database Structure

Diagram

Description automatically generatedDiagram

Description automatically generated

Figure Full quizdb EER Diagram

As QuizBud is designed to host quizzes, we have a complex set of interwoven tables built in the database as seen in **Figure 1 (**Full quizdb EER Diagram). However, Assignment 2 requirements allowed us to isolate our efforts to implementation of the course selection (which classes a student is enrolled in) and class list updates (which classes are maintained in the database). This limited area keeps us to the four tables in **Figure 2**.

Figure Assignment 2 quizdb EER Diagram

DDL and sample data can be implemented by running the script located at /public/sql/create-table.sql.

# Breakdown of Tasks

With the collaborative nature of this assignment, we chose to utilize both Discord and GitHub as collaboration tools, with GitHub being linked it to VS Code through the Remote Explorer option and submitting every modification through commits. The tasks would always involve hearing input from all members, and double checking the work done, ranging from the more accessory aspects of the assignment, such as determining the name and design of the platform being developed, to the preparatory steps, such as designing the wireframe of the site, to the more hands-on aspects of it, with the coding itself, like designing the HTML framework, creating and updating the CSS to design the entirety of the platform, creating the SQL script and making sure it worked, as well as populating it with enough information to test its contents and functionality.

Other than meeting in person to work on the project together, we remained connected during the vast majority of the work even remotely through discord, where we could interact through voice chat and share our screens, so we could more efficiently help one another and make sure the project proceeded as smoothly as possible.

# Authorship of Code

Please refer to the included git history for any clarifications of authorship. We apologize for the brusque nature of our commit messages, and will endeavour to write more clearly in the future.