

Milestone Two

Team 104-4 | Squad 4

Project Feature List

- Login Page: This is the initial screen that shows up. The user can log in to their existing account, which will take them to the home page. If they do not have an account, there will be a button to register that will take user to register page.
- Register page: User creates an account. Once they are registered, they are taken to the home page
- How to play page: Instructions on how to play game
- Score page: Shows leaderboard of everyone who has played and their scores.
- Game page: This is where the game is played. It starts with a timer that counts down to start the game.
- Menu page: The menu page features 3 buttons that take to the how to play page, scores page, play page
- Database: PostgreSQL database that stores email, username, password, score
- Integration Layer: Uses Node JS to connect our database to the front-end
- Sound Design: Sound effects for game

Requirements

- Login Page Requirements:
 - Be easy to understand and use.
 - The login page must be able to take input from the user, a username and password.
 - This page must send the input from the user to the backend using Node JS, check it against the PostgreSQL database, and respond accordingly with javascript (wrong user/password, redirect to another page on success).
- Register Page Requirements:
 - Be easy to understand and use.
 - Register page is similar to the login page in that it will take input from the user in the form of a username and a password, and maybe some other additional fields.
 - It will have to send the information to the backend with Node JS, and add an entry to the PostgreSQL database, then redirect the user to a new page.
- Score Page:
 - Be easy to read and understand.

- Pull a certain number of entries from the database base on highest score and use them to populate a table or similar structure on the front end page.
- Game Requirements:
 - Be fun to play and be well designed graphically.
 - Use JS to trigger queues for the user, visual or audio, then read keyboard input from the player and determine whether to continue the game, if the player succeeded at reacting to the queue, or end it if they didn't.
 - Update the database on an ended game with new statistics like high score, games played, etc.
- Menu Page Requirements:
 - Be easy to understand and helpful for the user to navigate the website.
- Database Requirements:
 - Contain data for each user, such as username, password, email, high score, games played, etc.
 - Be able to integrate with Node JS to pull data, and store new data.

Project Plan

Kanban Board: <https://github.com/kaeleng/3308-project/projects/1>