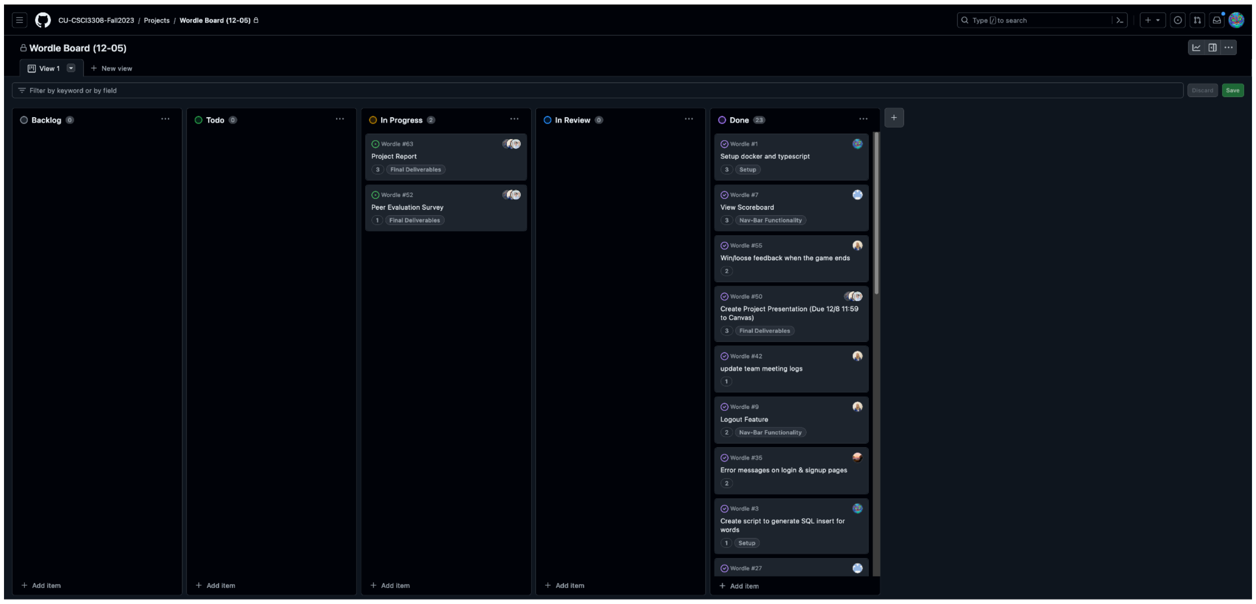
Wordle App

Developers: Tai Karuna, Fernando Picoral, Nathan So, Ella Arnold, Aidan Youell

Description: Our application is a guessing game where the user has six chances to guess a five-letter word. The user will type a word, and the application will respond by telling the user which letters they guessed correctly and if they are in the correct location. The user's goal is to guess the word in the fewest possible number of guesses. The five-letter word is selected randomly from a database with thousands of possible choices. After a word is selected from the database, it will be removed so that a player cannot guess the same word twice. The player can create their account, log in and out of their account, view their statistics, view the instructions for the game, and see a scoreboard. The project went very smoothly because we kept a strict but consistent and organized schedule. We hosted weekly meetings as a team and with our TA to ensure we were keeping a good pace to complete the project. These meetings allowed everyone to understand what they had to do and gave us opportunities to collaborate on issues. We also created a group chat to allow easy and clear communication.

Project Tracker: https://github.com/orgs/CU-CSCI3308-Fall2023/projects/43



CU-CSCI3308-Fall2023 / Projects / Wordle Board (12-05)

Wordle Board (12-05)

View 1 | New view

Filter by keyword or by field

Backlog 0

Todo 0

In Progress 2

- Wordle #63 — Project Report — 3 — Final Deliverables
- Wordle #52 — Peer Evaluation Survey — 1 — Final Deliverables

In Review 0

Done 23

- Wordle #1 — Setup docker and typescript — 3 — Setup
- Wordle #7 — View Scoreboard — 3 — Nav-Bar Functionality
- Wordle #55 — Win/loose feedback when the game ends — 2
- Wordle #50 — Create Project Presentation (Due 12/8 11:59 to Canvas) — 3 — Final Deliverables
- Wordle #42 — update team meeting logs — 1
- Wordle #9 — Logout Feature — 2 — Nav-Bar Functionality
- Wordle #35 — Error messages on login & signup pages — 2
- Wordle #3 — Create script to generate SQL insert for words — 1 — Setup
- Wordle #27

VCS: https://github.com/CU-CSCI3308-Fall2023/Wordle

Video: 012-5 Video Demonstration

Contributions:

Tai:

- Wrote html for the game board
- Added custom styles to the board and contributed to styles for the keyboard
- Contributed to javascript for the game logic
- Created and styled error messages for login and register failures

Fernando:

- Implemented the following endpoints, including necessary queries and type definitions:
    - /game/start
        - Start a new game
    - /game/guess
        - Add a guess to an existing game
    - /scoreboard
        - Get the ranking/points for all users in the system
    - /auth/login
        - Login
    - /auth/signup
        - Signup/register
- Wrote tests for all /game/* and /auth/* endpoints
- Setup Dockerfile for prod, dev, and test environments
- Setup dev tools (Typescript, ESLint, Prettier, etc)
- Created database seed files
- Wrote frontend JS to handle the following regarding the game logic:
    - Request to start a new game
    - Request to add a guess to the current game
    - Coloring of the letters on the game grid and on-screen keyboard based on the response from the server
    - Event listeners to allow typing with the physical keyboard

Nathan:

- Wrote the HTML code and embedded javascript code for scoreboard view
    - app/src/ui/views/scoreboard.ejs
- Wrote the CSS for scoreboard

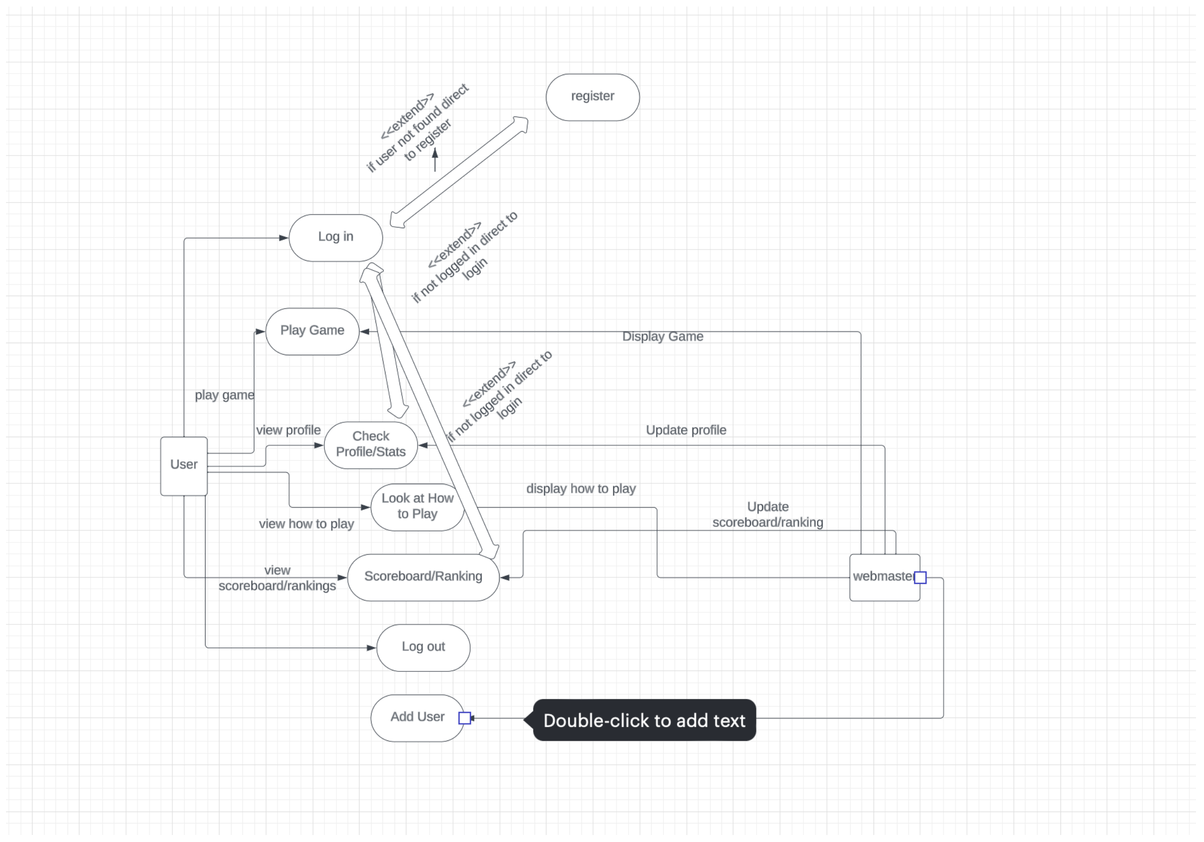
    - app/src/public/css/scoreboard.css
- Handled deployment on azure for lab 13, changed line in json file to help deploy
- Created logout endpoint
- Wrote HTML for nav-bar.ejs(did not style it)

Ella:

- Wrote the HTML code and some CSS for instructions view
    - app/src/ui/views/instructions.ejs
- Implemented the route for instructions.ejs
    - app/src/index.ts
- Implemented the Header and Footer partials
    - Continuously added any classes and libraries needed for styling and functionality of the front end
    - app/src/ui/partials/header.ejs, footer.ejs
- Completed the UAT document deliverable for lab 11
- Continuously updated Team Meeting Logs document throughout the project
- Created and styled the onscreen keyboard in the game view
- Used Bootstrap to implement the end-of-game pop up modal that appears to give the user feedback on their game
    - Allows user to either start new game or navigate to the scoreboard

Aidan:
- Wrote the HTML and CSS for the following views: login, register, and instructions
    - app/src/ui/views/instructions.ejs
    - app/src/ui/views/login.ejs
    - app/src/ui/views/register.ejs
    - app/src/public/css/instructions.css
    - app/src/public/css/loginStyles.css
    - app/src/public/css/registerStyles.css
- Implemented the nav-bar partial and styles for it
    - app/src/ui/partials/nav-bar.ejs
    - app/src/public/css/nav-bar.css
- Helped to create the UAT document
- Completed project presentation

Use Case Diagram:



Test Results:

UAT:
https://github.com/CU-CSCI3308-Fall2023/Wordle/blob/fe419d1d2010c25ee98ddff8408a64810bfeaa24/milestones/uat.md

For Lab 11 (see link above), we defined three tests: one for login, one for signup, and one for the scoreboard.

For the login and signup, we developed automated tests that are passing based on our acceptance criteria. For the scoreboard, we changed our approach from automated tests to manual testing. Nevertheless, the acceptance criteria were still met after performing multiple manual tests.

Furthermore, we also developed automated tests for all the game-related endpoints (/game/start and /game/guess), which were not on our UAT. As for these tests, the acceptance criteria were the following:

1. It should not start games with words that the user has already guessed
2. It should correctly compute the points after each guess

3. It should correctly provide the feedback for all letter in the guess (correct position/in the word but wrong position/not in the word)
4. It should return HTTP 400 if either the guess is invalid or the same guess was used before in the current game
5. It should return HTTP 404 if either the user doesn't own the game (the game ID references a game started by another user) or if the game is already finished (the user either already won or lost/used all guesses)

The tests we developed for the game-related endpoints met all the above requirements.

Deployment:

https://github.com/CU-CSCI3308-Fall2023/Wordle/blob/fe419d1d2010c25ee98ddff8408a64810bfeaa24/milestones/lab-13-deployment.md