Project Document

CFG Degree Fullstack Autumn 2023

Team: 5

Project Name:

Run to the Moon

Introduction

Aims and Objectives

'Run to the Moon' is a browser based app. Its main goal is to promote physical activity and healthy habits in school-aged children and their families by facilitating an enjoyable and rewarding approach to running. We aim to have a positive impact on children, families, schools, and the wider community by nurturing a love for running and promoting healthy competition.

Project Roadmap

1. Background
2. Specifications and Design
3. Implementation and Execution
4. Testing and Evaluation
5. Conclusion

1. Background

It is essential to encourage children from an early age to participate in physical activities, such as running, in a fun and supportive environment as it offers numerous benefits. These include, improved cardiovascular health, better immune system, cognitive development, and increased confidence. Therefore, we aim to promote healthy competition with this application.

How it works:

School children will be able to participate in groups and compete against each other as teams. Each child and their families will run as much as possible and submit their miles into the app via an admin user for their team. Whichever team submits enough miles to reach the Moon first is declared the winner. The winning team will receive exciting prizes and all participating groups will be given smaller prizes.

2. Specifications and Design

Project Requirements

1) **User Interface**

   Simple and inviting UI through the layout and the use of bright colours and images suitable for children. Required pages include:

   - Landing page with animation, mission statement, and calls to action
   - Login page
   - Sign up page which currently shows an option to join the waitlist

- Team Dashboard where groups can submit their new miles and see their progress
- Leaderboard showing the progress of all competing teams
- Not found page

2) **Database**

There are four tables in the database to allow for normalisation of the data being collected and stored. Tables are as follows:

- Teams: Team ID, TeamName
- TeamCredentials: Credential ID, Team ID, Email, Password
- MilesRun: Run ID, Team ID, Miles Completed, Submission Date
- Waitlist: FullName, Email

3) **Backend Server**

The Express server provides API endpoints to deal with fetching and posting from and to the database.

- To fetch team credentials and authenticate teams via the Login page
- To collate waitlisted contact details
- To send submissions of new miles from the Team Dashboard page
- To provide the current total miles ran by teams to be displayed in the Dashboard and Leaderboard pages

Design and Architecture

1) **Languages and Libraries:**
    - UI and UX built with React.
    - React Bootstrap for styling and responsive design.
    - MySQL database design.
    - Server-side logic in Node.js with Express
2) **Wireframe:**
    - The intended design of our web application was mapped out on a wireframe using Miro. You can find the wireframe in the repository.
    - We focussed on usability and layout when designing the architecture of each page using colours and calls to action to inform the user of their next steps. We did this by following the heuristics and principles addressed by Norman and Nielsen.
3) **Colour Choices:**

Our focus was on creating an engaging page suitable for children. The main colour scheme is blue and yellow to represent the sky and stars, respectively. Brighter shades were chosen to keep the page vibrant and suitable for children.

- Blue: Peaceful and represents stability and structure, thus creating a calm atmosphere with its use in the application.
- Yellow: Incites positive emotions as it symbolises happiness and imagination. Following Norman's principle, the colour is used to encourage the user's eyes to

focus on the next steps on the page, working in conjunction with the calls to action attached.

- Other: For the progress bars, different colours such as green, yellow, and red, are used to help differentiate between the teams on the Dashboard and Leaderboard.
- The shades used were tested to see if they aligned with WCAG standards, considering both the background and foreground elements for colour contrast.

3. Implementation and Execution

Development Approach

Iterative development with a focus on MVP components
The project was broken down into non-negotiable features that would provide a base-level experience for the user with frequent reviews of commits and pull requests to preview the website functionality. Any improvements or additions were discussed as a group and added incrementally.

Task breakdown and allocation based on SWOT analysis
Each team member conducted an individual SWOT analysis to establish strengths and weaknesses so that each member could contribute to the project with confidence and equally learn from others; in the areas they felt they needed more guidance. The individual SWOTs helped to inform the overall project SWOT so that we were able to make an informed decision about the project choice, implementation, and execution.

Tasks were managed on Trello, a task management workspace, where each member was allocated to a particular task marked with a status indication so that other members were aware of the project and individual progress.

Collaboration and peer reviewed via Github
By using Github, reviews of code took place via informative Pull Requests (PR) which explained the rationale for changes and updates. Equally, in team meetings live discussions of PRs helped to evaluate and improve our work. We used branches to isolate updates and added different team members as ‘Reviewers’ to review our PRs before merging to the main branch.

Tools

- Slack for ad-hoc communication, collaboration and clarification.
- Zoom for group calls.
- WhenAvailable for team availability scheduling.
- Trello for task management and visualisation.
- Miro for SWOT analysis and wireframing.

- GitHub for code collaboration.

Implementation Process

Process

1. Code was implemented in a manner that allowed team members to begin work independently of one another on sections that were not reliant on others. Once the application framework was in place, we moved onto *end-to-end functionality*, before applying *responsive design*.
2. End-to-End Functionality
    - Submission of miles → updating database → returning new total miles
    - Login form → username and password gathered → check against database → team logged in and dashboard made available
    - Leaderboard presenting total miles → rearranged in descending order
    - Logout button becomes available when team logged in → selected and team logged out
3. Responsive Design
    - Updates to styling once product nearing completion
    - Media queries applied at a standard mobile screen size breakpoint

Achievements	**Challenges**	**Changes made**
- A working server - Successful axios calls from the front-end to the database(DB) - Inclusion of enhancements beyond MVP e.g. waitlist DB table	- Getting everyone's SQL connection working - Attempting Bcrypt (see branch feature/password-bcrypt) - Failing tests	- Simplified animation on home page - Personalised coloured progress bars rather than moving team icons

4. Testing and Evaluation

Testing Strategy

Testing file for unit tests

Unit tests were implemented for the majority of pages and components within our project.

- Test Suites: 7 failed, 3 passed
- Tests: 1 failed, 6 passed

There were issues with error codes on a number of our tests that we were unable to rectify, namely around out of date 'babel' modules and use of axios within our components.

Usability testing for user experience.

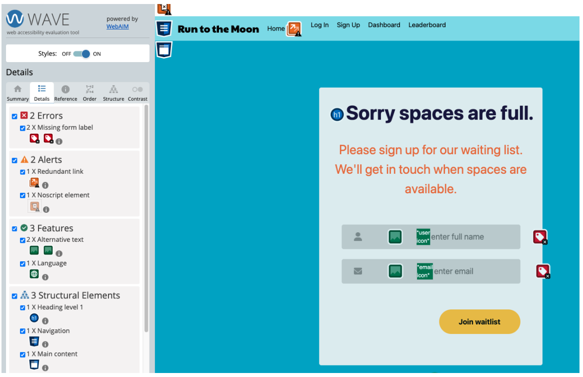
Once we were happy with the application, we carried out usability testing. Checking that links acted as expected, and screens, messages and pages rendered at the correct times. During testing, we identified that the ‘See Your Progress’ link on the homepage took you to the ‘LogIn’ page when the user was already logged in. A fix was implemented using Redux; the link will now direct the user to the ‘LogIn’ page if they are not already logged in, and to the Dashboard page if they are.

Responsive design was also tested at varying breakpoints; issues with page height were identified and rectified on the Dashboard and Home page.

HTML Accessibility Testing

Accessibility checks were carried out using the ‘WAVE Evaluation’ Chrome Extension. The checks highlighted errors on the Login and Sign Up pages. The errors advised that label names were required on the input fields. These errors were rectified in the final product

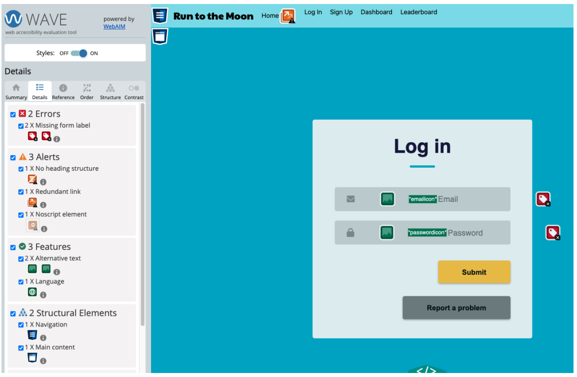
Before Amendments



☑ 2 Errors
☑ 2 X Missing form label
☑ 2 Alerts
☑ 1 X Redundant link
☑ 1 X Noscript element
☑ 3 Features
☑ 2 X Alternative text
☑ 1 X Language
☑ 3 Structural Elements
☑ 1 X Heading level 1
☑ 1 X Navigation
☑ 1 X Main content

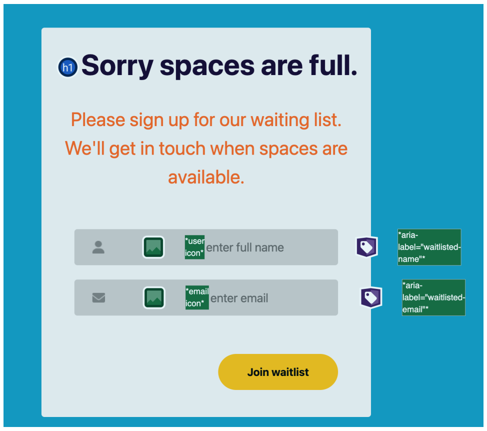
Sorry spaces are full.

Please sign up for our waiting list. We'll get in touch when spaces are available.



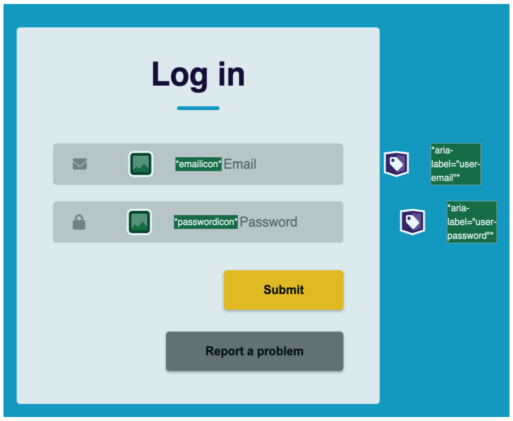
☑ 2 Errors
☑ 2 X Missing form label
☑ 3 Alerts
☑ 1 X No heading structure
☑ 1 X Redundant link
☑ 1 X Noscript element
☑ 3 Features
☑ 2 X Alternative text
☑ 1 X Language
☑ 2 Structural Elements
☑ 1 X Navigation
☑ 1 X Main content

After Amendments



Sorry spaces are full.

Please sign up for our waiting list. We'll get in touch when spaces are available.



The remainder of the pages received no error warnings, but did include suggestions for improvements. The suggestions were considered but subsequently not applied as we felt the user experience would be negatively affected by the changes.

System Limitation

We considered the following limitations in our design;

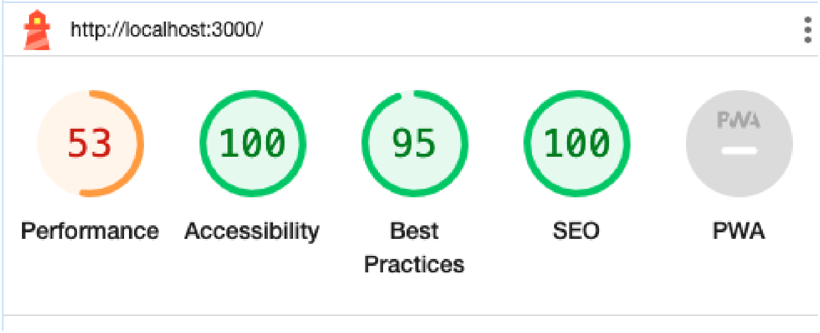
- Connection to MySQL database remaining open and stable
- User's internet connection reliability.

5. Conclusion

We successfully developed our ‘Run to the Moon’ Minimum Viable Product, as well as some of the enhancements that were discussed at design stage (such as the Sign Up page creating a waitlist table within the SQL database.)

Future enhancements and considerations

- Get sign up to actually sign new teams up rather than add to waitlist
- Allow teams to update their account details and have multiple login details per team (Database is already normalised to facilitate this)
- Add a GPS API so that users can sync Strava and other running apps to give a more accurate and reliable representation of miles run.
- Have dynamic application of colours applied to the progress bars
- Add password bcrypt
- Further animation within the application to engage users further
- Hosting the web application
- Interrogating findings from Google Lighthouse website quality tool; specifically with a focus on improving the performance of the web app;



http://localhost:3000/

53 Performance

100 Accessibility

95 Best Practices

100 SEO

PWA