***Muni.***

***Synopsis:***

Muni is a Single Page Application that allows users to view a live leaderboard for the municipal golf courses in the city they have chosen. It exists solely to compare a users daily or weekly scores with other local players. I believe this will be a good addition to my portfolio to give centralized exposure to most, if not all, of the current technologies I comprehend.

***< Front-End Management >***

* The front-end is to be built with React and will allow for a user to login (via a local or OAuth method), access the leaderboard, add/remove a score, and edit their profile. React router will be utilized for front-end routing
* GH Pages will host the application

***< Back-End Management>***

* Express will be used as the backend framework to define and handle routes.
* PostgreSQL will be the database provider and will manage user information, course information, and scores.
* Sequelize will be used as the ORM on the back-end to define schemas and communicate with a PostgreSQL database.

### ***1. User Story: View the Leaderboard***

**As a** user,

**I want to** view a live leaderboard for the municipal golf courses in my selected city,

**so that** I can compare my daily or weekly scores with other local players.

**Acceptance Criteria:**

* Users can choose their city from a dropdown or search field.
* The leaderboard updates dynamically with the latest scores.
* Scores are sortable by criteria such as date, course, or rank.

### ***2. User Story: Add or Remove Scores***

**As a** user,

**I want to** add or remove my golf scores,

**so that** my position is reflected on the leaderboard.

**Acceptance Criteria:**

* Users can input their scores via a form that includes date, course, and score.
* Validation ensures scores are in the proper format before submission.
* Scores can be removed or updated, with the changes reflected in real-time.

### ***3. User Story: Manage My Profile***

**As a** user,

**I want to** edit my profile details,

**so that** my information and display preferences are up to date.

**Acceptance Criteria:**

* Users can update their name, city, and other personal information.
* Profiles include an avatar that users can upload or change.
* Changes are saved to the database and reflected across the application.