Events Platform

Project Overview

A small community business has reached out to you to create a platform where they can create and share events with members of the community.

You have been tasked with building and hosting a platform (either a website or a mobile app) that allows community members to view, sign up for, and add events to their own personal calendars. Staff members should have additional functionality to create and manage events.

Minimum Viable Product (MVP)

Your platform must fulfil the following functionality:

1. Display a list of events for users to browse.
2. Allow users to sign up for an event.
3. Allow users to add events to their Google Calendar after signing up.
4. Enable staff members to sign-in to create and manage events.

See Completion and Submission Requirements for more details.

Tech Choices:

- The platform should be built using **JavaScript** or **TypeScript**.
- **Event Data**: You can use either a freely available API for event data or create your own event data. Research and decide on which API to use prior to starting. The focus is on building the platform, not on data generation.
- **Calendar API**: You'll need to sign up for the Google Calendar API (or an equivalent) using a free developer account. This will allow users to add events to their calendars.
- Implement security best practices for **user authentication**
- Host the project on a **free platform** (e.g., GitHub Pages for web, Expo for mobile apps).

The following technologies and tools are **suggestions**, not requirements:

- **React** or **React Native** for the frontend.
- **TypeScript** for a new challenge.

- **Google Calendar API** for calendar integration.
- **ExpoDev** for hosting a mobile React Native apps - This platform will provide a QR code and URL so your project can be accessed via the ExpoDev app.

UI Requirements

- Ensure the design is **responsive** and works well across different screen sizes.
- **Accessibility** must be considered for users with disabilities (e.g., screen readers, keyboard navigation).
- The platform should clearly communicate **errors** to the users (e.g., failed requests, missing fields).
- Loading states should be obvious when content is being fetched.
- The user interface should be intuitive, making it easy to find, sign up for, and create events.

Completion and Submission Requirements

The due date will be advised, but it will be no later than four weeks after the project commencement.

Your project must meet the following criteria to be considered complete:

1. The project must be **hosted** and publicly accessible (web or mobile).
2. The README must include:
    - A summary of the project
        - you may consider recording a **video walkthrough** of your platform, highlighting key features. Host this video on a free platform (e.g., YouTube) and include a link in your README.
    - Test account access details
    - Clear instructions on how to run the project locally, including any setup steps (e.g., installing dependencies, setting up environment variables).
3. Meet the MVP requirements outlined above.

Failure to do this may result in the project being rejected.

Optional Extensions

If you have time once you have completed the MVP requirements, consider adding the following features:

1. **Payment platform integration**: Implement payments via Stripe, Google Pay, etc.
2. **Confirmation emails**: Automatically send confirmation emails to users who sign up for an event.
3. **Social media integration**: Allow users to share events on social platforms.
4. **Cross-platform**: Build both a website and a mobile app.
5. **Google/Social login**: Allow users to sign up using their Google or social media accounts.