# Developer Manual – Sunny Days App

## 🛠️ How to Install the Application and All Dependencies

1. Clone the repository to your local environment:

git clone https://github.com/YOUR\_USERNAME/sunny-days.git

cd sunny-days

2. This project is a static web app. It doesn’t require Node.js or any backend server to run. All dependencies are loaded via CDN, including:

- FullCalendar

- Day.js

- Supabase JavaScript client

3. If you plan to scale or modularize the code (e.g., using Webpack, React, or Vite), initialize the project with:

npm init -y

4. No installation is required for API access — keys and configuration are embedded in the client-side script.

## 🚀 How to Run the Application on a Server

A. Locally (for testing):

- Open index.html directly in your browser. It will fetch weather and event data automatically.

B. Deployment (e.g., Vercel):

- Push your repository to GitHub.

- Connect it to https://vercel.com and import the repository.

- Set the deployment root to / and build output to public (or default settings).

- That’s it — the app will be live and responsive.

## 🧪 How to Run Tests

This project does not include automated testing. Manual QA steps are:

- Add an event using the New Event form → check for confirmation.

- Refresh the Home page → verify event persists and appears in the calendar.

- Switch between pages (Home, New Event, About) → verify animations and transitions.

- Deny location permissions → check fallback or error handling for weather display.

## 📡 API Usage – GET and POST

The application uses two APIs:

1. Supabase (custom backend)

- GET:

supabase.from('events').select('\*');

- POST:

supabase.from('events').insert([ { name, start\_time, end\_time } ]);

2. Open-Meteo (public weather forecast API)

- GET daily forecast:

https://api.open-meteo.com/v1/forecast?latitude=...&longitude=...&daily=temperature\_2m\_max,weathercode...

- GET hourly temperature:

Used to display current weather upon loading.

These are handled using fetch() in the front end.

## 🐛 Known Bugs and Roadmap

Known Issues:

- Requires browser geolocation permission.

- Calendar view doesn’t support editing or deleting events.

- No mobile responsiveness or touch optimization.

- No authentication or user-specific calendars.

Future Roadmap:

- Add user accounts via Supabase Auth

- Mobile-friendly layout using Flexbox/Grid media queries

- Drag-and-drop event editing

- Notification system for weather warnings

- React or Vue component migration

## 📁 Documentation Notes

- All application logic currently lives in index.html.

- You may modularize into separate .js and .css files as needed.

- Store this README in the root of your project repository.

- Supabase credentials are stored in the frontend. If needed, migrate to a secure backend or environment variable system for production.