Requirements Document

WeatherNow Calendar Integration

Pascal Putz
pascal.putz@study.thws.de
5123135

Gunn Kataria
gunn.kataria@study.thws.de
9125072

Katrina Alex
katrina.alex@study.thws.de
9125071

Manuel Stöth
manuel.stoeth@study.thws.de
5123045

Marvin Kraus
marvin.kraus@study.thws.de
5123143

April 2025

1 Author Information

Name	**Email**	**StudyID**
Pascal Putz	pascal.putz@study.thws.de	5123135
Gunn Kataria	gunn.kataria@study.thws.de	9125072
Katrina Alex	katrina.alex@study.thws.de	9125071
Manuel Stöth	manuel.stoeth@study.thws.de	5123045
Marvin Kraus	marvin.kraus@study.thws.de	5123143

2 Project Overview

2.1 Project Description

WeatherNow Calendar Integration is a web-based application designed to simplify free-time planning by combining real-time weather data with the user's personal calendar schedule. By leveraging the OpenWeatherMap API and the Google Calendar API, the app allows users to check current and forecasted weather conditions alongside their availability. Based on weather statistics and free days, WeatherNow suggests personalized activities, helping users make the most of their time—whether it's a sunny outdoor hike or a cozy indoor museum visit. The frontend is built using React.js to ensure a modern, responsive, and smooth user experience across both mobile and desktop devices.

2.2 Application Goals

- Enable users to view accurate current weather and detailed forecasts for any location worldwide.
- Integrate the user's Google Calendar to identify free and busy periods automatically.
- Suggest context-aware activities based on weather conditions and user availability.
- Provide a user-friendly, accessible, and visually appealing interface with light/dark mode options.
- Ensure fast, reliable, and error-tolerant performance for all weather and calendar interactions.
- Build a modular and scalable frontend that can be extended with additional APIs or features in the future.

2.3 Target Users

- **Travelers and Tourists** planning outdoor or sightseeing activities.
- **Daily Commuters** preparing for weather disruptions during workdays.
- **Outdoor Enthusiasts** (hikers, cyclists, sports players) relying on forecasts.
- **Busy Professionals and Students** maximizing free time with suitable recommendations.
- **Families and Groups** organizing weekend outings or vacations dependent on weather.

3 Key Features

Must-Have

- **City Search & Current Weather:** Users search a city, see temperature, wind, humidity, conditions and icon.
- **5-Day Forecast:** Forecast in 3-hour intervals with daily high/low summaries.
- **Geolocation Lookup:** Automatically fetch weather for user's current coordinates.
- **Unit Toggle:** Switch between °C and °F.
- **Theme Toggle:** Light and dark modes, persisted across sessions.
- **Interactive Calendar:** Select any date to trigger analysis.
- **Activity Suggestions:** Recommend indoor/outdoor activities based on weather rules.
- **Error Handling:** Friendly messages for invalid input, denied permissions, or API failures.
- **Responsive UI:** Adapts to mobile, tablet, and desktop breakpoints.
- **Client-Side Navigation:** React Router routes for Home, Forecast, and Calendar.

Nice-to-Have

- Store and recall recently searched cities.
- Sync and overlay multiple Google calendars.
- Localize UI text and weather descriptions (e.g., EN/DE).
- Lookup historical weather for past dates.
- Push notifications for severe weather or upcoming free days.

4 User Roles and Interactions

Primary role: General user The system shall enable users to:

- Enter or select a city to view weather and forecast.
- Grant calendar access to retrieve busy/free days.
- Select a date on the calendar to fetch weather and activity suggestions.
- Toggle units (°C/°F) and theme (light/dark).
- Navigate via header/menu to Home, Forecast, and Calendar.
- Receive inline or toast notifications on errors.

5 User Stories / Use Cases

1. As a user, I want to enter a city name so I can view current weather conditions.
2. As a user, I want to see a five-day forecast so I can plan upcoming days.
3. As a user, I want to grant calendar access so the app knows my free days.
4. As a user, I want to click a date in my calendar to get weather stats and activity suggestions.
5. As a user, I want to switch between °C and °F based on my preference.
6. As a user, I want dark mode at night to reduce eye strain.
7. As a user, I want to be notified if an API call fails or I enter an invalid city.

6 Non-Functional Requirements

- **Usability:** Clear labels, consistent icons, minimal learning curve.
- **Responsiveness:** Fluid layout using CSS Flexbox/Grid for all viewports.
- **Accessibility:** ARIA labels, sufficient contrast, keyboard navigation.
- **Performance:** Data fetch and render complete under 1.5 s on average networks.
- **Security:** HTTPS only; API keys in environment variables.
- **Maintainability:** Modular React components; documented code.
- **Scalability:** Architecture supports adding new APIs/features with minimal refactoring.

7 Technology Assumptions

- **Frontend:** React.js (ES6+).
- **Styling:** CSS Modules or Tailwind CSS.
- **Routing:** React Router DOM.
- **APIs:** OpenWeatherMap REST, Google Calendar OAuth2.
- **State Management:** React Context or Redux.
- **Build/Deploy:** Docker multi-stage build; Nginx serves static files.

8 Project Constraints

- OpenWeatherMap free tier: max 60 calls/min.
- Google Calendar API quotas and OAuth consent requirements.
- Requires explicit user permission for location/calendar access.
- No custom backend development; all logic in frontend.
- Requirements Document due: April 30, 2025.

9 Acknowledgement of AI Tools

This document was drafted and refined using GPT-4o based on an outline provided by the authors. The authors reviewed and enhanced the output for coherence and alignment with course requirements.