Weather Dashboard with Customizable Alerts

Project Proposal

WDD330 – Nelson Monteiro

Overview

What is the problem we are trying to solve?

Users often need real-time, location-specific weather data and alerts to plan their day effectively. Many apps offer this service, but not all allow users to fully customize alerts based on specific conditions like temperature, humidity, wind speed, or severe weather warnings.

Why are we doing this?

This project aims to create a weather dashboard that gives users an easy way to access current weather conditions, forecasts, and customizable alerts. It will be useful for anyone who needs timely weather information, such as commuters, outdoor enthusiasts, or people in areas prone to severe weather.

Who is the audience?

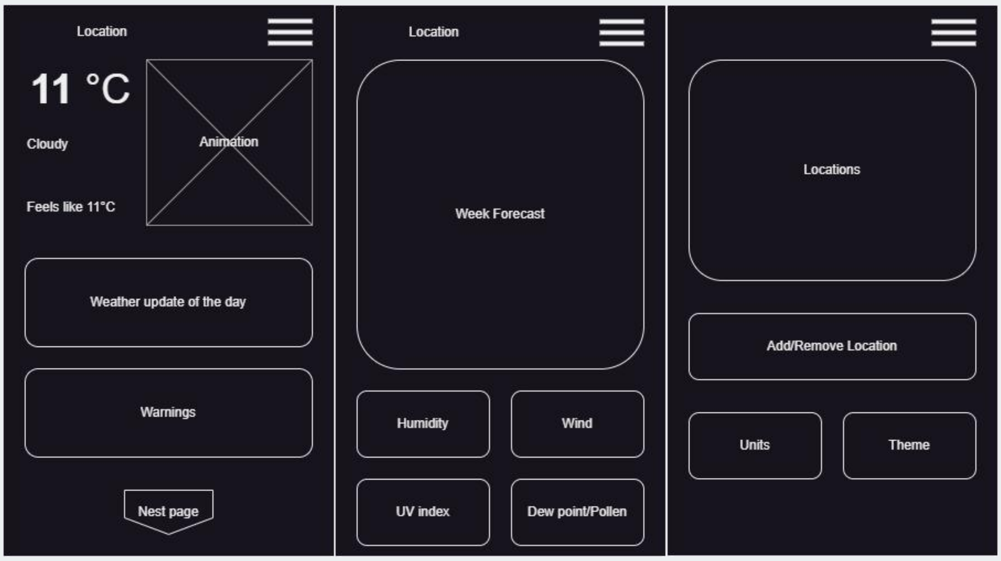
- General public who wants an easy-to-use weather tool.
- Users who are planning outdoor activities (hiking, sports, etc.).
- Users in regions prone to severe weather who need quick alerts.

List of the major functions of the application

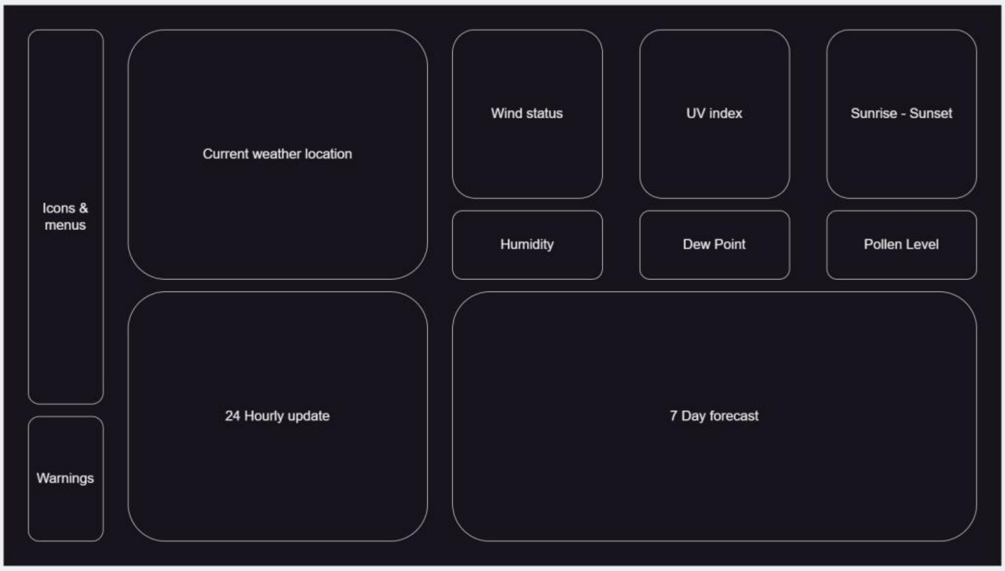
- **Display Current Weather**: Shows current temperature, wind speed, humidity, and other conditions for a specified location.
- **7-Day Forecast**: Provides a forecast for the upcoming week, displaying daily highs, lows, and general conditions (rain, sun, etc.).
- **Customizable Alerts**: Users can set up notifications based on specific weather conditions (e.g., "Alert me if the temperature goes below 4°C" or "Send a notification for severe weather warnings").
- **Search by Location**: Allows users to input a city or use their geolocation to get weather data for their current location.
- **Mobile and Desktop Views**: The design will be responsive and mobile-first, ensuring that it works well on any device.
- **Weather Map**: Integrates a visual map showing weather patterns (optional, depending on API availability).
- **Animations**: Dynamic animations will be included to enhance the user experience, such as animated weather icons, transitions when switching views, and fading in of weather data after an API fetch.

Wireframes of the Major Views

- Mobile Wireframes:



- Desktop Wireframes:



External API Data Source(s)

- OpenWeatherMap API: This API will provide the current weather, forecast, and weather alerts. It is widely used, reliable, and has excellent documentation. We'll use their free tier to start.
- WeatherAPI.com: As a potential alternative for more detailed forecast data and customizable options like air quality and severe weather alerts.

Initial Module List

1. App.js: The main entry point that initializes the app, handles user interactions, and calls the relevant modules.

2. **WeatherService.js:** A module for handling all API requests, including fetching the current weather, forecasts, and alerts.
3. **AlertManager.js:** Manages the logic for setting and triggering custom weather alerts.
4. **UIController.js:** Handles rendering weather data to the DOM, updating the UI, and displaying notifications and alerts.
5. **Animations.js:** Provides helper functions for adding animations like weather icons, transitions, and fading effects.
6. **Utils.js:** General utility functions, such as formatting dates, converting temperature units, and storing user preferences (e.g., saved locations, alert settings) in local storage.

Colours/Typography/Specific Element Styling

- **Colours:**
  - **Primary Colour:** Light Blue (#5DADE2) – Represents calm, clear weather.
  - **Secondary Colour:** Dark Gray (#273849) – For text and icons.
  - **Alert Colour:** Red (#E74C3C) – For severe weather alerts.
  - **Background Colour:** Light Gray (#ECF0F1) – For clean, minimalist design.
- **Typography:**
  - **Font:** Roboto.
  - **Headers:** Bold, slightly larger font (H1, H2).
  - **Body Text:** Regular weight, smaller for secondary info like humidity and wind.

Schedule (Mile Markers)

1. **Week 1 (Setup and Core Features):**
   - Finalize API choice and set up the basic project structure using **Vite**.
   - Implement initial layout and basic API calls to fetch current weather.
   - Complete the functionality for current weather and 7-day forecast display.
   - Ensure data is fetched from the API and rendered correctly.
2. **Week 2 (Custom Alerts & UI Enhancements):**
   - Implement the ability to customize alerts and notifications.
   - Add animations to enhance user experience.
3. **Week 3 (Final Touches & Testing):**
   - Test the app for bugs, refine styling, and ensure everything works across mobile and desktop.
   - Final code cleanup, ensuring everything passes ESLint checks.

Link to a Trello Board

https://trello.com/b/yHYpTsxF/weather-dashboard