Project Report: Weather Dashboard Website

Weather Dashboard Website

Submitted by: Ankit Patil

Date: July 17, 2025

Introduction & Objectives

Introduction & Objectives:

The Weather Dashboard is a full-stack web application developed to provide real-time weather updates for any user-specified location or the user's current location. The objective is to integrate weather API services with a responsive and dynamic user interface that makes the data accessible and understandable. It is ideal for learning API integration and building a full-stack project.

Technologies & Architecture

Technologies & Architecture:

- Front-end: HTML, CSS, JavaScript
- Back-end: Node.js with Express.js
- Hosting: Netlify (for front-end), Render/Glitch/Heroku (for back-end)
- API: OpenWeatherMap for weather data

The architecture involves a front-end sending requests to the back-end server, which communicates with the OpenWeatherMap API, processes data, and sends it back to the client.

Key Features & Implementation Details

Key Features & Implementation Details:

- Responsive UI: Built with CSS for adaptive layouts across devices.

- Form Handling: Input field allows user to search for city-specific weather.

- Routing: Express handles API request routing on the backend.

- Common city data displayed in a tabular layout with temperature, humidity, and wind info.

Project Structure

Project Structure:

- /public: Contains HTML, CSS, and static assets.

- /server.js: Node.js + Express backend server for handling API requests.

- /scripts: JavaScript code for DOM manipulation and API fetch.

- .env: Contains environment variables like API key.

All code is modularized for easy maintainability.

Testing & Deployment

Testing & Deployment:

- Local: Run using `npm start`

- Deployment:

  - Frontend: Netlify

  - Backend: Can be deployed using Render/Heroku

- GitHub Repo: https://github.com/AP8113/weather-dashboard

Contributions & Future Enhancements

Contributions & Future Enhancements:

- All design and development done independently, including UI, backend, and API integration.
- Future enhancements may include:
  - Weekly forecast support
  - Location history
  - Dark mode
  - Enhanced loading and error states

Your Role and Responsibilities

Role and Responsibilities:

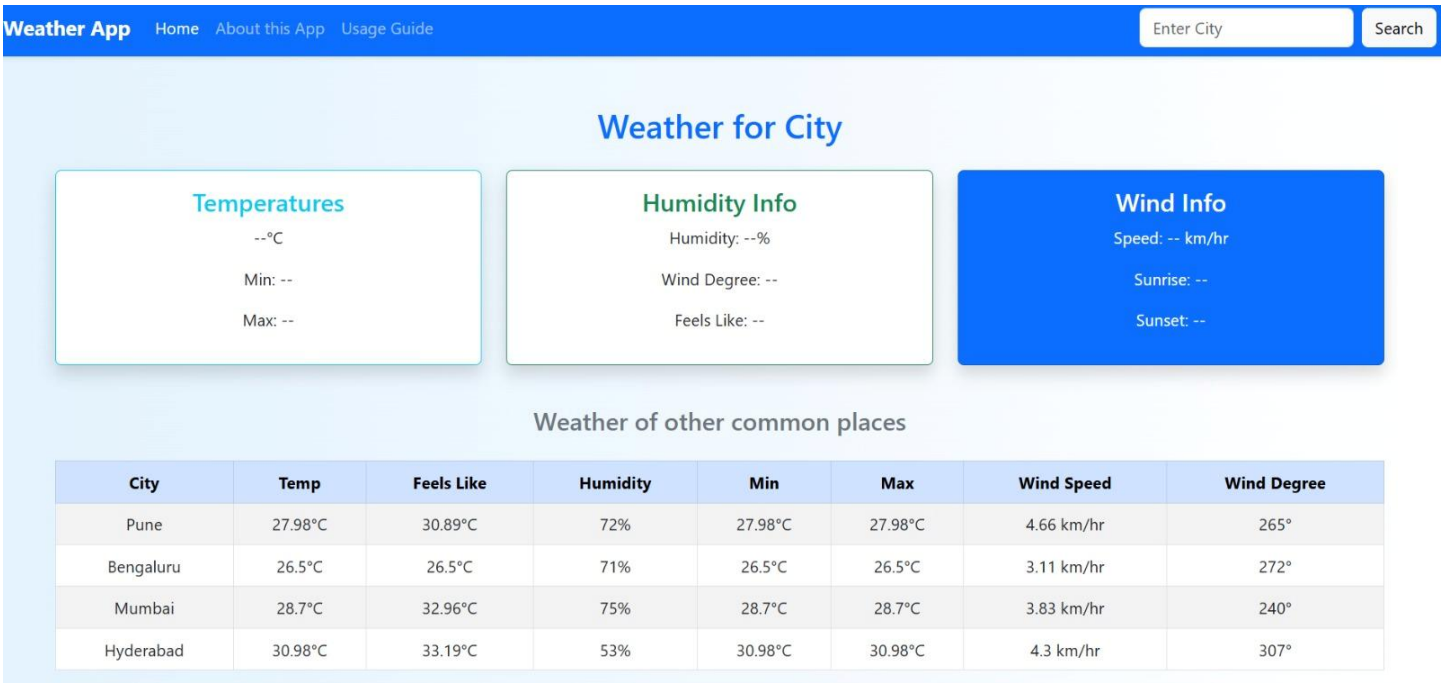
- Designed and developed the entire UI using HTML/CSS.
- Implemented JavaScript logic for DOM updates and user input handling.
- Built the back-end using Node.js and Express to handle API requests securely.
- Managed deployment of both frontend and backend.

npm Scripts and APIs

npm Scripts and APIs Used:

- npm start: Runs the local server for development.
- OpenWeatherMap API: Used to fetch weather data.
- Express: Node.js framework for server routing.
- dotenv: To manage API keys securely.

GitHub Repo: https://github.com/AP8113/weather-dashboard



Conclusion

Conclusion:

The project successfully demonstrates a complete weather dashboard with real-time data, built entirely by one developer. It serves as a solid example of how front-end and back-end technologies can be combined to create a functional, informative, and user-friendly application.