Weather Forecast Application - Project Documentation

---

1. Objective

The **Weather Forecast Application** is a web-based project that allows users to check real-time weather conditions for any location. The application utilizes the **OpenWeatherMap API** to retrieve weather data and dynamically displays information such as temperature, humidity, wind speed, and weather conditions. The system also provides an extended 5-day weather forecast and supports geolocation for fetching weather data of the user's current location.

---

2. Functionalities Implemented

- **Search by City Name:**
  Users can search for weather conditions by entering a city name in the search bar.
- **Current Location Weather Data:**
  The application uses the Geolocation API to retrieve the user's current latitude and longitude and fetch weather data accordingly.
- **Recent Search Dropdown:**
  The system maintains a list of recently searched cities in a dropdown menu using **local storage**, allowing users to quickly revisit previous locations.
- **5-Day Weather Forecast:**
  Displays an extended forecast with details such as temperature, weather icons, wind speed, and humidity.
- **UI for Multiple Devices:**
  Responsive design to ensure compatibility with desktops, tablets, and mobile devices.
- **Error Handling and Input Validation:**
  Validates search queries and handles API errors gracefully, displaying appropriate error messages when required.
- **Interactive User Interface:**
  Buttons, dropdowns, and real-time UI updates for an enhanced user experience.

---

3. File Structure & Semantic Breakdown

📁 File Structure

```
weather-forecast-app/
├── src/
│   ├── app.js             # JavaScript logic for
│   ├── index.html         # Main HTML structure
│   ├── input.css          # Tailwind CSS input file
│   └── output.css         # Compiled Tailwind CSS output file
├──package.json           # Node.js package configuration
```

HTML - (index.html)

- Utilizes semantic HTML elements such as `header`, `section`, `article`, and `footer` for better structure and accessibility.
- Contains input fields, buttons, and a dropdown menu for searching and selecting locations.
- Displays current weather conditions along with a 5-day weather forecast.

CSS - (input.css)

- Implemented using **Tailwind CSS** for responsive and modern design.
- Ensures a clean and minimalistic layout with appropriate spacing and color consistency.
- Applied media queries for mobile, tablet, and desktop views.
- Added scroll functionality to handle extended forecast results.
- Styled dropdowns and buttons with hover effects to enhance user experience.

JavaScript - (app.js)

- **Fetch Weather Data:**
  Retrieves weather data from the OpenWeatherMap API based on city name or

current location.

- **DOM Manipulation:**
  Dynamically updates the weather data, error messages, and UI changes based on user interactions.

- **Local Storage Management:**
  Stores and retrieves recently searched cities using `localStorage`.

- **Input Validation:**
  Prevents invalid or empty search queries and provides appropriate alerts.

- **Error Handling:**
  Displays appropriate error messages when an invalid location is entered or API fails.

Setup Instructions

To run the Weather Forecast Application on your local machine, follow the steps below:

1. **Clone the GitHub Repository** :

   ```
   git clone
   https://github.com/SravanGunaganti/weather-forecast-application.git
   ```
2. **Navigate to the Project Folder**:
   ```
   cd weather-forecast-application
   ```
3. **Install Dependencies**:

   ```
   npm install
   ```
   Run the following command to install Tailwind CSS and other dependencies

4. **Run the Development Server**:
   To build and watch the Tailwind CSS file for changes, use the following command:
   ```
   npm run start
   ```

   This will watch for any changes in your `src/input.css` and compile them to `output.css`

5. **Open the Application**:
   Open the `src/index.html` file in your browser to view the application:

4. Design Choices

- **Minimalist User Interface:**
  Designed with a clean and structured UI for intuitive navigation and interaction.
- **Color Palette and Styling:**
  Utilized a light blue and white color scheme to align with the theme of a weather application.
- **Responsive Design:**
  Ensured compatibility across multiple devices using Tailwind CSS media queries.
- **Dropdown for Recent Searches:**
  Dropdown UI implemented for displaying recently searched cities, making it easier for users to revisit locations.
- **Real-Time Error Alerts:**
  Error messages dynamically update when invalid search terms or API errors occur, improving the user experience.

---

5. Challenges Faced & Solutions

1. Maintaining Data on Page Refresh

Challenge: Ensuring that recent searches persist after page reload.
Solution: Implemented `localStorage` to store and retrieve recently searched cities dynamically.

2. Displaying Extended Forecast

Challenge: Organizing and displaying a 5-day forecast in a structured and readable format.
Solution: Used a grid layout to display forecast data clearly with icons, dates, and temperatures.

3. Geolocation Access Restrictions

Challenge: Handling cases where users deny geolocation access.
Solution: Provided appropriate error messages and allowed fallback to manual city search.

---

6. Live Demo & GitHub Repository

- **Live Demo:** https://accurate-weatherforecast.netlify.app/

- **GitHub Repository:**https://github.com/SravanGunaganti/weather-forecast-application.git

7. Conclusion

The **Weather Forecast Application** successfully implements essential features such as location-based weather forecasts, extended weather reports, and error-handling mechanisms. The project effectively integrates JavaScript for API requests, DOM manipulation, and local storage management. Tailwind CSS ensures a responsive and aesthetically pleasing user interface.