Weather App Documentation

Introduction

The Weather App is a simple web application that allows users to check the current weather conditions of any city. It retrieves real-time weather data from the OpenWeatherMap API and displays information such as temperature, humidity, and weather conditions.

Features

- Search for the weather of any city
- Displays temperature, humidity, and weather condition
- User-friendly interface with a frosted glass effect
- Dynamically updates background based on temperature

Technologies Used

- **HTML** for structuring the web page
- **CSS** for styling and layout (including a frosted glass effect)
- **JavaScript** for fetching and displaying weather data
- **OpenWeatherMap API** for retrieving real-time weather information

Project Structure

```
Weather App/
├── index.html      # Main HTML file
├── styles.css      # Stylesheet for UI design
├── script.js       # JavaScript file for API handling
```

Setup Instructions

Prerequisites

- A web browser (Google Chrome, Firefox, etc.)
- A text editor (VS Code, Sublime Text, etc.)
- An active internet connection
- API Key from OpenWeatherMap (Replace `YOUR_API_KEY` in the script.js file)

Steps to Run the Project

1. Clone or download the project.
2. Open the `index.html` file in a browser.
3. Enter a city name and click the "Get Weather" button.
4. The weather information will be displayed.

Code Explanation

1. HTML (index.html)

- Contains an input field for entering the city name.
- A button to trigger the API request.
- A section to display weather details.

2. CSS (styles.css)

- Defines styles for the weather app, including a background gradient and frosted glass effect.
- Ensures responsive design.

3. JavaScript (script.js)

- Fetches weather data from OpenWeatherMap API.
- Updates the UI dynamically with the retrieved weather information.
- Handles errors if an invalid city name is entered.

API Call Example:

```
const apiUrl =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${api
Key}&units=metric`;
```

Possible Improvements

- Display an icon representing the weather condition.
- Implement dynamic background images based on temperature.
- Add a loading animation while fetching data.
- Improve UI responsiveness for mobile devices.

Conclusion

This Weather App provides a simple yet effective way to fetch and display weather details. With enhancements, it can be made more interactive and visually appealing.