**Weather App – Project Report**

**1. Project Title**

**Weather App** – A web-based application to fetch and display current weather information using city names or ZIP codes.

**2. Objective**

The main objective of this project is to create a simple and responsive web application that allows users to check the current weather of any city or ZIP code worldwide using real-time data from the OpenWeatherMap API.

**3. Technologies Used**

* **HTML** – For structuring the web pages.
* **CSS** – For styling the app and making it responsive.
* **JavaScript** – For interactivity and fetching API data asynchronously.
* **OpenWeatherMap API** – To retrieve real-time weather data including temperature, description, and icons.

**4. Features**

* Input field for **city name** or **ZIP code** with country code.
* Submit button to fetch current weather data.
* Display of:
  + Temperature (°C)
  + Weather description
  + Weather icon
* Responsive layout compatible with mobile, tablet, and desktop.
* Error handling for invalid locations or API issues.

**5. System Overview**

The Weather App works by taking user input (city name or ZIP code) and sending a request to the OpenWeatherMap API. The API returns JSON data containing weather information, which is then displayed dynamically on the web page, including:

1. Temperature in Celsius.
2. Weather description (e.g., “Clear Sky”, “Partly Cloudy”).
3. A corresponding weather icon.

**6. Implementation**

**6.1 HTML**

* Provides the input field for city/ZIP code.
* Includes a button to fetch weather and a container to display results.
* Structured for accessibility and responsive layout.

**6.2 CSS**

* Styles the input, button, and result container.
* Implements responsive design to support different screen sizes.
* Adds visual cues like weather icons and background colors.

**6.3 JavaScript**

* Uses fetch() and async/await to call the OpenWeatherMap API.
* Processes the API response to extract temperature, description, and icon.
* Updates the HTML dynamically with retrieved weather data.
* Handles errors and invalid input gracefully.

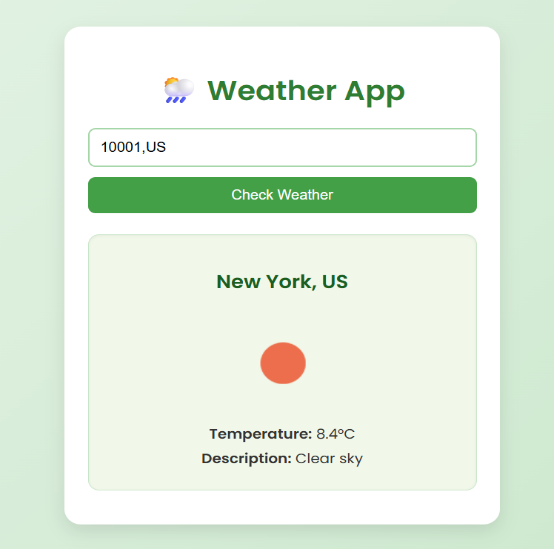
**7. How to Run**

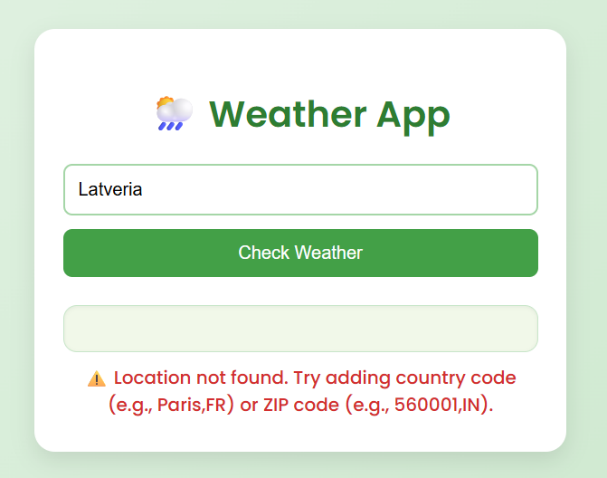
1. Clone or download the repository.
2. Navigate to the project directory.
3. Replace the apiKey in script.js with your OpenWeatherMap API key.
4. Open index.html in a web browser, or use a local server (e.g., Live Server in VS Code) for live reload.

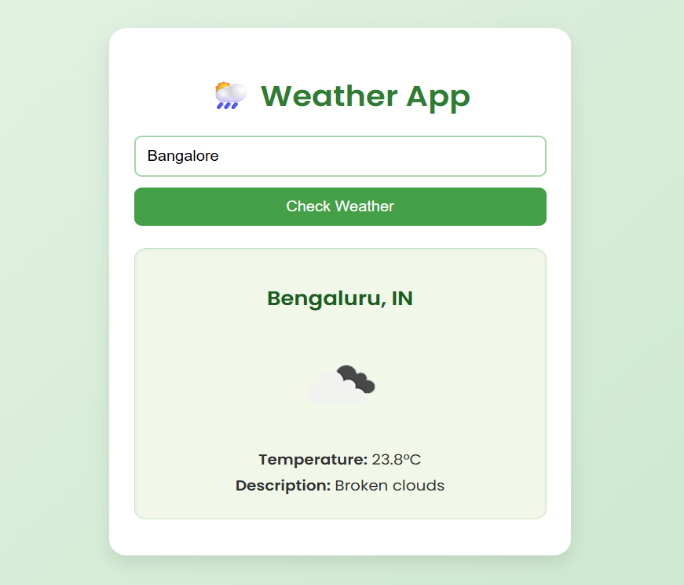
**8. Testing**

* Tested with city names like "Bengaluru,IN", "Paris,FR", "London,UK".
* Tested with ZIP codes like "560001,IN" and "10001,US".
* Verified that invalid locations (e.g., "Latveria") display appropriate error messages.
* Tested on multiple devices for responsive design (desktop, tablet, mobile).

**9. Screenshots**

****

****

****

**10. Challenges**

* Ensuring ZIP code coverage for all regions; some API responses may vary.
* Handling asynchronous API calls and updating the DOM dynamically.
* Making the design responsive across all screen sizes.

**11. Conclusion**

The Weather App successfully demonstrates how to fetch and display live weather data using an API. It provides a clean, responsive interface for users to check weather information for cities or ZIP codes. Error handling ensures a smooth user experience even with invalid input.

**12. References**

* [OpenWeatherMap API](https://openweathermap.org/)