NORTHERN UNIVERSITY BANGLADESH

Project Report: Weather App Using OpenWeatherMap API

Course Title: Software Development III

Course Code: CSE 3292

SUBMITTED BY:

Group – 51

SUBMITTED TO:

Md. Mahadi Hasan

Lecturer, Dept of CSE

Northern University Bangladesh

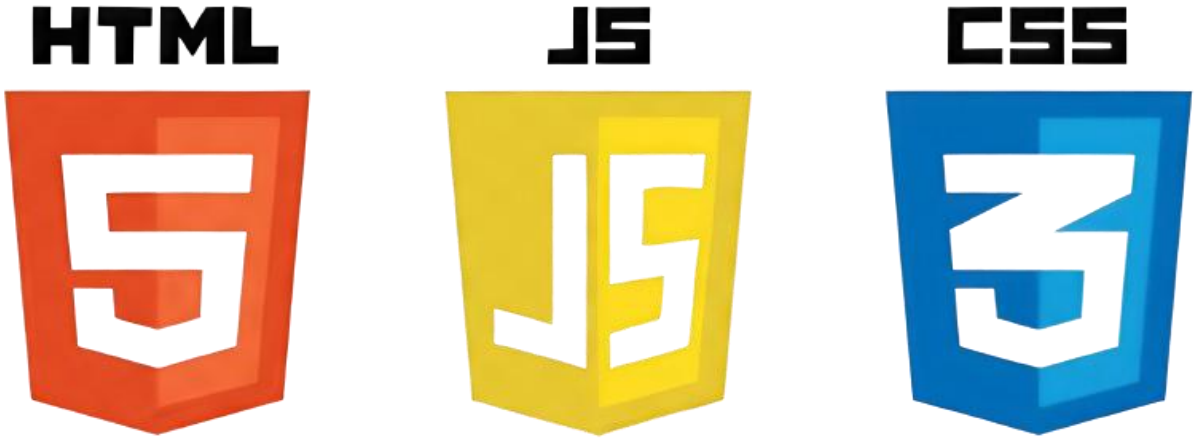
TABLE OF CONTENTS

1. **Project Title**

Weather App Using OpenWeatherMap API

2. **Introduction**

This project is a simple and interactive weather application that allows users to check the current weather of any city in the world. The app fetches real-time data from the OpenWeatherMap API and displays temperature, humidity, wind speed, and weather conditions with dynamic icons.

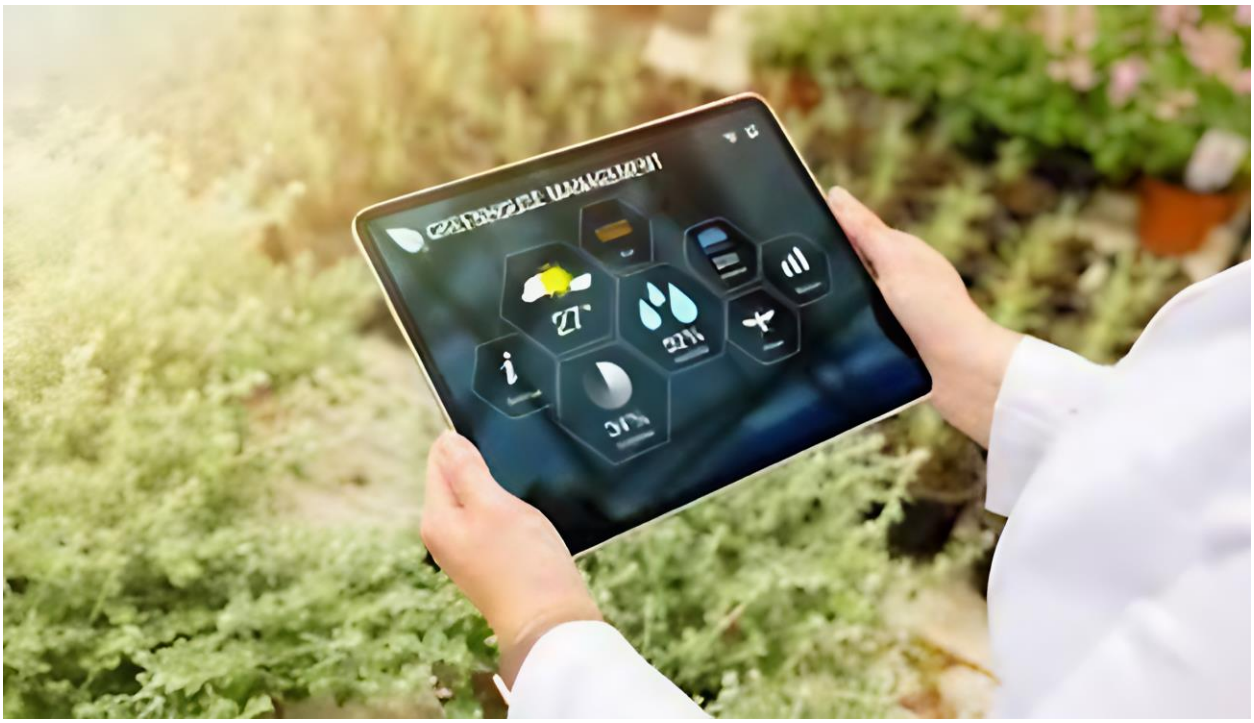


3. **Objective**

The main objective of this project is to create a responsive weather information app using HTML, CSS, and JavaScript, while learning how to integrate and use third-party APIs effectively.

4. **Tools and Technologies Used**

- **Frontend**: HTML5, CSS3, JavaScript (ES6)
- **API**: OpenWeatherMap (RESTful API)
- **Image Assets**: Weather icons (cloud, sun, rain, etc.)
- **Text Editor**: VS Code
- **Browser**: Firefox (for testing)



5. System Design and Features

*a) **User Interface**:*

- **Search bar** for entering the city name
- **Search button** to initiate weather fetch
- **Dynamic display** of:
    - City name
    - Temperature (°C)
    - Humidity (%)
    - Wind speed (km/h)
    - Weather condition with icon
    - Allow location-based weather using the Geolocation API.
    - Weather forecast for the next 5 days.
    - weather forecast for the every 3 hours.

*b) **Error Handling**:*

- If the city name is invalid, an error message is shown.
- API call failures are handled gracefully.

*c) **Visuals**:*

- Weather condition icons change dynamically based on the API response (`Clear`, `Clouds`, `Rain`, etc.).

6. Implementation Details

- The `fetch()` API is used to get weather data in JSON format.
- DOM is updated using `document.querySelector()` and `.innerHTML` methods.
- Different weather conditions are handled using conditional logic and appropriate icons.
- Input is taken from a text box, and a button click triggers the main function.

Code Overview:

- HTML: Structures the layout with search input and weather display blocks.
- CSS (style.css): Styles the card, input field, weather box, and icons.
- JavaScript:
  - Fetches data using OpenWeatherMap API.
  - Updates weather information dynamically.
  - Handles user interactions and invalid inputs.

7. Sample Output

- **Valid Input**:
  Input: London
  Output: Temperature: 15°C, Humidity: 68%, Wind Speed: 10 km/h, Icon: Cloudy

- **Invalid Input**:
  Input: `FakeCity123`
  Output: "Invalid city name..." error message displayed.

8. **Conclusion**

This project demonstrates how to build a real-time weather application using HTML, CSS, and JavaScript. It showcases API integration, DOM manipulation, event handling, and basic error checking. It is scalable and can be further enhanced with features like multi-day forecasts, geolocation-based weather, and dark/light themes.

9. **Future Enhancements**

- Add loading animations and improve responsive design.
- Support for multiple languages and units (Celsius/Fahrenheit).