WEATHER APP

Full Stack Web Development Project Report
(Unified Mentor Internship)

---

1. Title Page

Project Title: Weather App
Domain: Full Stack Web Development
Technologies Used: HTML, CSS, JavaScript, Public Weather API (OpenWeatherMap)
Difficulty Level: Hard
Internship Organization: Unified Mentor

Github: https://github.com/vai938/weather_app
Intern Name: *Vaibhav Thapliyal*
Duration: *Nov 2025 – Feb 2025*

---

2. Abstract

The Weather App is a web-based application designed to provide real-time weather information to users based on their selected city or location. The application leverages a public weather API to fetch current weather conditions and presents them through an intuitive and responsive user interface. Key information such as temperature (in Celsius), weather description, and corresponding weather icons are displayed dynamically.

The primary goal of this project is to strengthen frontend development skills while integrating external APIs using JavaScript. The project emphasizes asynchronous programming, error handling, responsive UI design, and user-centric interaction. This report details the system design, implementation, challenges faced, and future scope of the Weather App.

---

3. Introduction

Weather information plays a crucial role in daily planning, travel, agriculture, and disaster management. With the rise of web technologies, it has become increasingly feasible to provide real-time weather updates through lightweight web applications.

The Weather App project aims to build a simple yet functional weather application using core web technologies. This project demonstrates practical usage of HTML for structure, CSS for styling and responsiveness, and JavaScript for logic, API communication, and dynamic rendering of data.

This project was undertaken as part of the internship at Unified Mentor to gain hands-on experience in building real-world web applications and integrating third-party services through APIs.

---

4. Problem Statement

Users often require quick access to accurate and up-to-date weather information for different locations. Existing weather platforms may be heavy, cluttered with advertisements, or not developer-customizable.

Problem:
To design and develop a lightweight, responsive web application that allows users to input a city name and retrieve current weather information reliably, with proper handling of invalid inputs and API errors.

---

5. Objectives

The main objectives of this project are:

- To design a user-friendly web interface for weather data retrieval.
- To integrate a public weather API for real-time data.
- To implement asynchronous JavaScript for API calls.
- To display weather details, including temperature, description, and icons.
- To implement error handling for invalid locations and network/API issues.
- To ensure responsiveness across devices and screen sizes.

---

6. Scope of the Project

In Scope:

- City-based weather search
- Display of current temperature (in Celsius)
- Weather description and icon
- Responsive UI
- Error messages for invalid inputs

Out of Scope (Future Enhancements):

- 7-day weather forecast
- Location-based automatic weather detection (GPS)
- User authentication
- Saving favorite locations

---

7. Literature Review (Brief)

Several web-based weather applications exist that rely on public APIs such as OpenWeatherMap. These applications typically use RESTful APIs and JSON responses. The project draws conceptual inspiration from commonly used weather dashboards but implements a simplified version focusing on learning objectives: API integration, frontend responsiveness, and asynchronous JavaScript programming.

---

8. System Architecture

High-Level Architecture:

```
User → Web Interface (HTML/CSS) → JavaScript Logic
                               → Weather API (OpenWeatherMap)
                               ← JSON Weather Data
                               → UI Rendering
```

Components:

- Frontend UI (HTML + CSS)
- API Integration Layer (JavaScript Fetch API)
- Data Presentation Layer (DOM Manipulation)

---

9. Technology Stack

Layer	Technology Used
Frontend	HTML5, CSS3
Scripting	JavaScript (ES6)
API Service	OpenWeatherMap API
Tools	Browser, VS Code

---

10. Functional Requirements

- The system shall allow users to input a city name.
- The system shall fetch weather data using a public API.
- The system shall display temperature, description, and icon.
- The system shall handle invalid city names gracefully.
- The system shall display error messages when API fails.

---

11. Non-Functional Requirements

- **Usability:** Simple and intuitive UI.
- **Performance:** Quick API response handling.
- **Reliability:** Proper error handling.
- **Responsiveness:** Works on mobile, tablet, and desktop devices.
- **Scalability:** Can be extended with forecasts and additional features.

---

12. Implementation Details

12.1 Frontend (HTML & CSS)

HTML is used to structure the layout, including:

- Input field for city name
- Submit button

- Display container for weather details

CSS ensures:

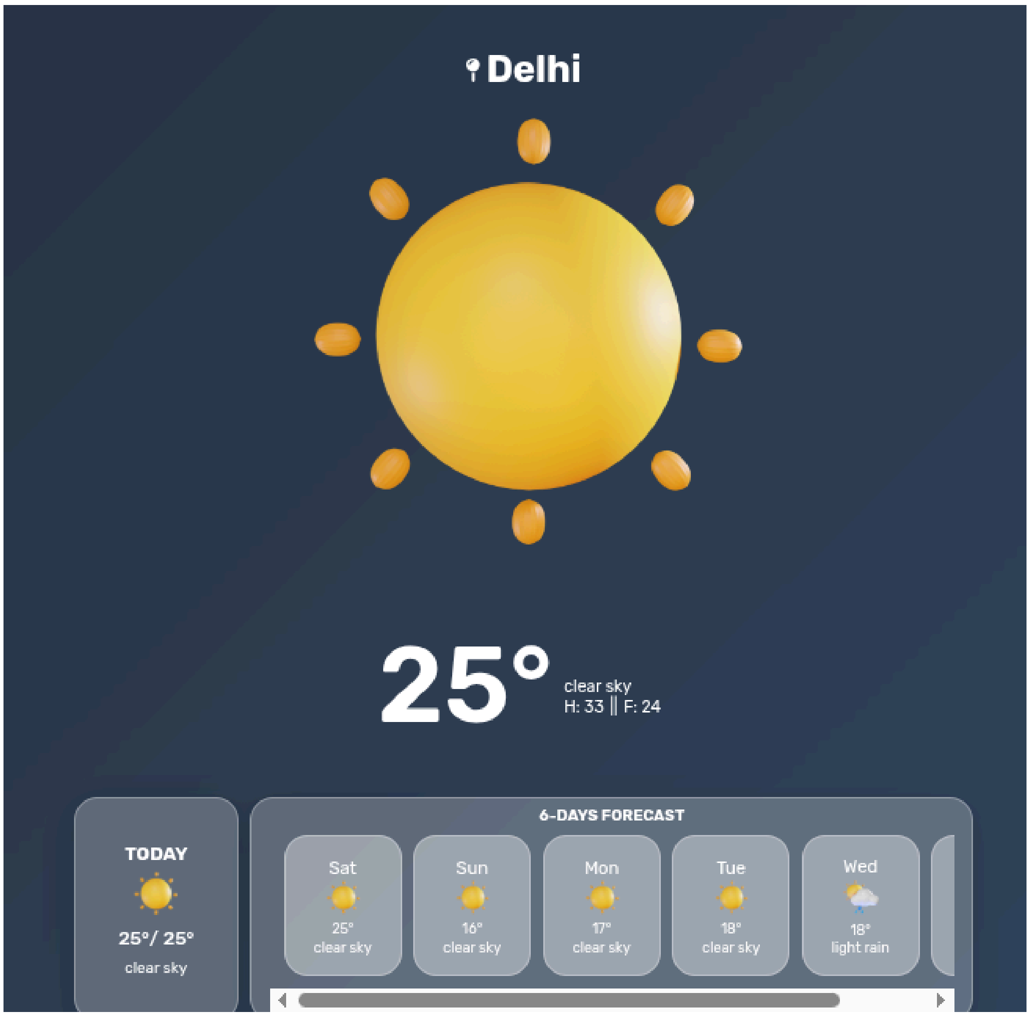
- Responsive layout
- Visual appeal
- Proper alignment and spacing
- Mobile-friendly design

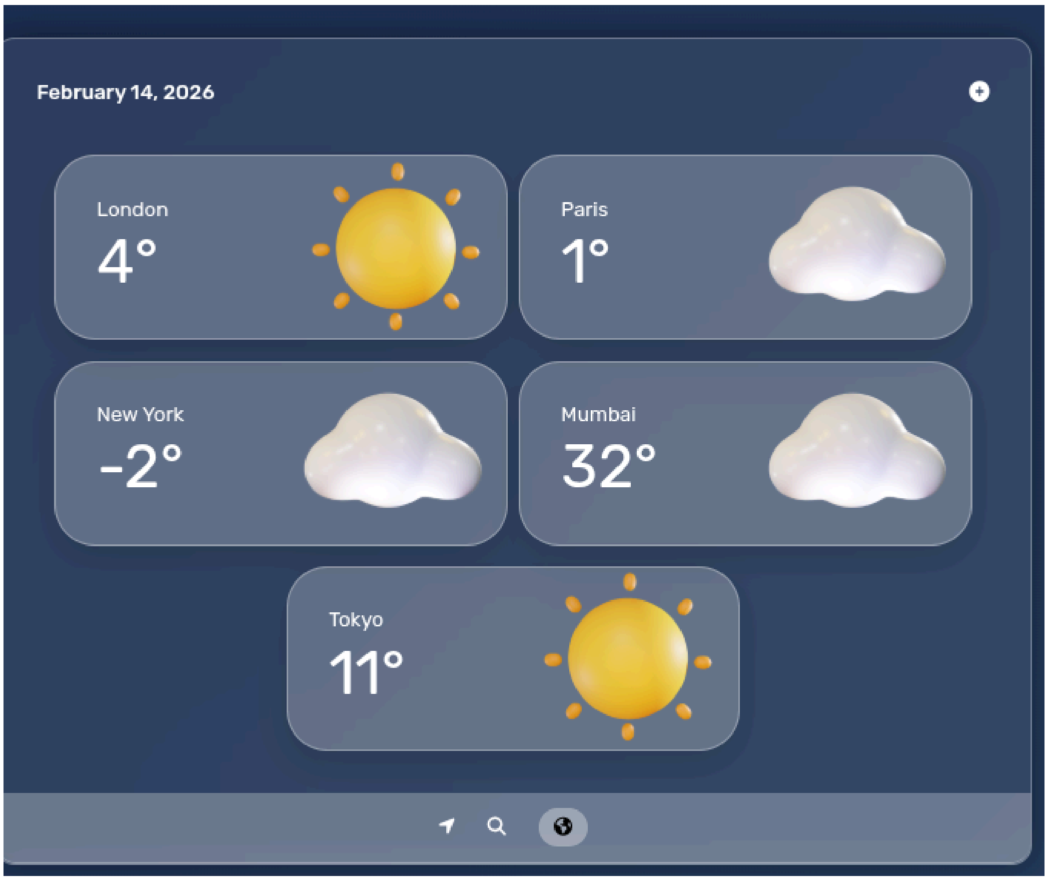
12.2 JavaScript Logic

JavaScript handles:

- Capturing user input
- Making asynchronous API calls using `fetch()` or `async/await`
- Parsing JSON response
- Updating DOM elements dynamically
- Handling API and input errors

13. Output Screens (Placeholders)





February 14, 2026

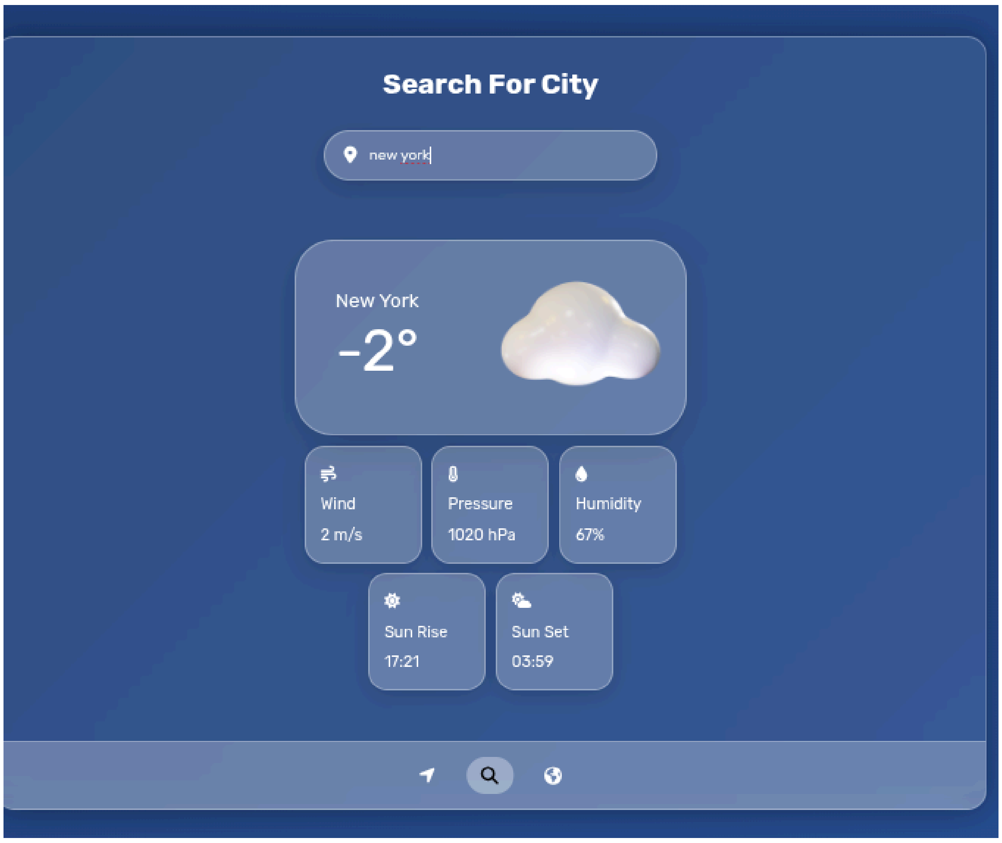
London
4°

Paris
1°

New York
-2°

Mumbai
32°

Tokyo
11°



Search For City

new york

New York
-2°

Wind
2 m/s

Pressure
1020 hPa

Humidity
67%

Sun Rise
17:21

Sun Set
03:59

- Home Screen
- Weather Result Display
- Error Message Screen

---

14. Testing

The application was tested on:

- Multiple browsers (Chrome, Edge, Firefox)
- Different screen sizes
- Valid and invalid city names
- Network failure simulation

---

15. Challenges Faced

- Handling API response errors
- Managing asynchronous calls and promises
- Designing a responsive UI
- Debugging incorrect city inputs
- Ensuring correct temperature unit conversion

---

16. Results and Outcomes

The Weather App successfully retrieves and displays real-time weather data for user-provided locations. The application meets all functional requirements and provides a clean, user-friendly interface. This project improved understanding of:

- API integration
- Frontend architecture
- Error handling strategies
- Asynchronous JavaScript programming

---

17. Future Enhancements

- Add multi-day forecast feature
- Enable location-based auto-detection
- Add unit toggle (Celsius/Fahrenheit)
- Store recent searches
- Deploy the app online

---

18. Conclusion

The Weather App project successfully demonstrates the practical application of frontend web development concepts integrated with external APIs. It serves as a foundational project for building more complex, data-driven web applications. The project provided hands-on experience in real-world development scenarios, aligning well with internship learning outcomes at Unified Mentor.