## **1. Introduction**

The Weather App project is a web-based application that retrieves and displays real-time weather data using the OpenWeatherMap API. The app was developed as part of a full-stack development bootcamp to demonstrate core competencies in frontend technologies, API integration, UI/UX design, testing, and user interaction.

The objective was to build an interactive, visually appealing application that allows users to search for weather conditions in any city and view essential details such as temperature, humidity, weather conditions, wind speed, and a short-term forecast.

## **2. Project Requirements & Phases**

The development process was organized into six structured phases:

### **Phase 1: Project Setup & Basic Layout**

**Objective:** To establish the project folder structure, create a static HTML layout, and apply initial CSS styling.

**Tasks Completed:**

* Created index.html for basic layout with a title, input box, and result section.
* Applied Flexbox and padding in style.css for layout and spacing.
* Styled the container and elements for desktop and mobile responsiveness.

**Files:**

* index.html
* style.css

### **Phase 2: Weather API Integration**

**Objective:** To fetch real-time weather data using the OpenWeatherMap API and dynamically display it in the UI.

**Tasks Completed:**

* Registered for and used a personal API key.
* Implemented JavaScript (script.js) to fetch data from two endpoints:  
  + Current weather data
  + 5-period forecast
* Converted temperature from Kelvin to Celsius.
* Displayed fetched weather data in the app: city name, temperature, humidity, condition, icon, and wind info.
* Handled errors like invalid city input or API fetch failure.

**Tools Used:**

* JavaScript fetch()
* OpenWeatherMap API (v2.5)

### **Phase 3: Data Display & UI Enhancement**

**Objective:** To refine the presentation of the weather data and ensure clarity and visual appeal.

**Enhancements:**

* Displayed temperature with °C symbol.
* Added labels for humidity, wind speed, and description.
* Used API-based weather icons.
* Applied CSS effects (box-shadow, spacing, padding, etc.)
* Implemented a styled forecast section for the next 5 periods.

**UI Design Choices:**

* Responsive Flexbox layout
* Color theme: soft whites, violet accent buttons, background image (space.png)
* Weather icon visuals

### **Phase 4: User Interaction & Optional Features**

**Objective:** To enhance user experience by enabling smoother interaction.

**Features Implemented:**

* "Enter" key triggers the weather search.
* A "Reset" button clears all input and weather display data.
* Reset button is disabled until data is fetched, improving UX.

**Optional Feature Status:**

* Geolocation: **Not implemented**

### 

### **Phase 5: Testing & Bug Fixing**

**Objective:** To verify that the app handles all expected inputs and behaves correctly.

**Manual Testing Performed:**

| **Test Case** | **Status** |
| --- | --- |
| Valid city input | ✅ Passed |
| Invalid city input (e.g., "Xyzabc") | ✅ Passed |
| Empty city input | ✅ Passed |
| Weather image/icon loads correctly | ✅ Passed |
| Reset button functionality | ✅ Passed |
| Enter key search functionality | ✅ Passed |
| Responsive layout on mobile/desktop | ✅ Passed |

**Common Bugs Handled:**

* Re-fetching issue if no input
* Typo or symbols in city name
* Forecast clearing upon bad input

### **Phase 6: Documentation & Finalization**

**Objective:** To finalize the project with clean, readable code and create documentation.

**Tasks Completed:**

* Source code was cleaned and commented (especially in script.js)
* README file was created with instructions, features, and file structure
* Project structured for GitHub or ZIP submission

## **4. Learning Outcomes**

Through this project, the following skills were strengthened:

* DOM manipulation using JavaScript
* API integration and handling JSON responses
* Form input validation and error handling
* CSS Flexbox-based responsive design
* Debugging and testing techniques for web apps
* Clean code documentation and user-centric design principles

## **5. Screenshots**

 