



PBEL Virtual Internship

Weather Application

Submitted

By

Devraj singh

Moradabad Institute of Technology

Submitted

To

MR . Karan Raj

Declaration

IDevraj singh hereby declare that the project report entitled “ Weather Application ” done by me under the guidance of Mr. Karan Raj (Mentor) at IBM PBEL Virtual Internship, is submitted in partial fulfillment of the requirements for the award of IBM PBEL Virtual Internship Certification.

DATE: 12-july-2025

Devraj singh

PLACE: MORADABAD

SIGNATURE OF CANDIDATE

Acknowledgement

I am deeply thankful to my respected guide, Mr. Karan Raj , whose guidance, mentorship, and motivation were invaluable throughout the course of this internship project. His timely suggestions, valuable insights, and constructive feedback enabled me to complete the project successfully.

Iwould also like to express my gratitude to the IBM PBEL Virtual Internship Team for providing this platform to learn real-world technologies and improve technical skills. I am equally thankful to my college, Moradabad Institute of Technology , for supporting and encouraging students to participate in such industry-oriented internships. Last but not least, I thank my peers and family who provided constant support.

Table of Contents

Acknowledgement.....	3
Table of content.....	4
Abstract.....	5
Introduction.....	6
Project Objective.....	7
Technologies Involved.....	8
1. React.js.....	8
2. JavaScript (ES6+).	8
3. HTML & CSS.....	8
4. OpenWeatherMap API.....	8
5. Git & GitHub.....	8
Application Architecture.....	9
1. App Component.....	9
2. SearchBox Component.....	9
3. InfoBox Component.....	9
4. WeatherApp.css.....	9
Features of the Application.....	10
Problems Faced & Solution Implementation.....	11
1.API key Handling.....	11
2.Error management.....	11
3.Styling and Responsiveness.....	11
4.Managing Componet state.....	11
Output.....	12
Screenshot:1.....	12
Screenshot:2.....	13
Screenshot:2.....	14

Abstract

The Weather Application is a responsive and interactive web-based tool that provides users with current weather information for any city across the globe. Built using React.js and powered by the OpenWeatherMap API, the application delivers real-time data such as temperature, humidity, and general weather conditions through a user-friendly interface.

The primary goal of this project is to demonstrate the practical application of modern front-end development skills, including API integration, component-based architecture, and responsive design principles. The project also emphasizes clean code structure, reusable components, and effective error handling to ensure a robust user experience.

This application was developed as part of the IBM PBEL Virtual Internship, with an aim to bridge the gap between academic learning and real-world project development. It serves as a strong foundation for understanding full-cycle front-end development and working with external data sources in a scalable and maintainable way.

Introduction

The Weather Application is a real-time, responsive, and interactive web-based application that allows users to get current weather details of any city across the world. It was developed using React.js, a modern JavaScript library for building user interfaces. The data is fetched from the OpenWeatherMap API which provides comprehensive weather data including temperature, humidity, weather description, and more.

The project was developed under the PBEL Virtual Internship program offered by IBM, with the goal of improving practical software development skills. Through this project, I have applied my knowledge of React.js, REST APIs, asynchronous data handling, and responsive UI design.

Project Objectives

- To build a responsive web application using React.js.
- To consume an external weather API and integrate it with the frontend.
- To allow users to search for any city and get real-time weather information.
- To implement good UI/UX design practices and responsive layout.
- To handle errors gracefully, such as invalid input or network failure.

Technologies Involved

1. React.js: Used for building the UI components and managing the app state.
2. JavaScript (ES6+): Core programming language used for logic and API calls.
3. HTML & CSS: For structuring and styling the web application.
4. OpenWeatherMap API: To fetch current weather details via HTTP requests.
5. Git & GitHub: Version control and code repository management.

Application Architecture

The application follows a component-based architecture provided by React.js. It is divided into several logical components:

1. App Component: The root component that contains the main application logic.
2. SearchBox Component: Provides input field for users to enter a city name and triggers API call.
3. InfoBox Component: Displays the weather data such as temperature, humidity, weather description, etc.
4. WeatherApp.css: Responsible for styling and layout.

When a user enters a city and submits the form, an API call is made to the OpenWeatherMap server using `fetch()`. The data is then stored in React state and rendered dynamically inside the InfoBox component.

Features of the Application

- Real-time weather search by city name.
- Displays temperature, humidity, and general weather conditions.
- Input validation and error messages for invalid city entries.
- Attractive and responsive UI compatible with both mobile and desktop.
- Clean and minimal codebase with reusable components.

Problems Faced & Solution Implementation

1. API Key Handling

Problem: The API key was exposed in the frontend.

Solution: Added API key through environment variable for secure usage during development.

2. Error Management

Problem: Application would crash if the city entered did not exist.

Solution: Implemented try-catch block with error state to display friendly error messages.

3. Styling and Responsiveness

Problem: UI elements were misaligned on smaller screens.

Solution: Used Flexbox and media queries to enhance responsiveness for all screen sizes.

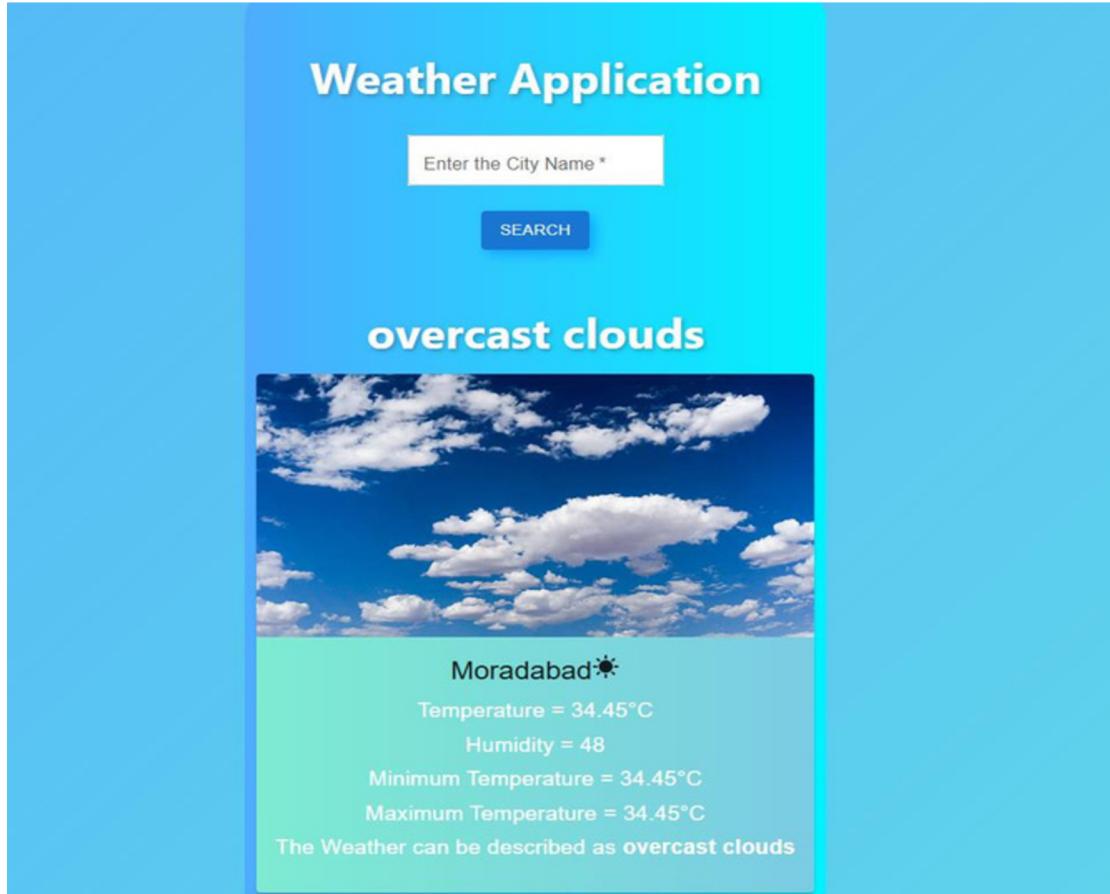
4. Managing Component State

Problem: Passing data between components while maintaining separation of concerns.

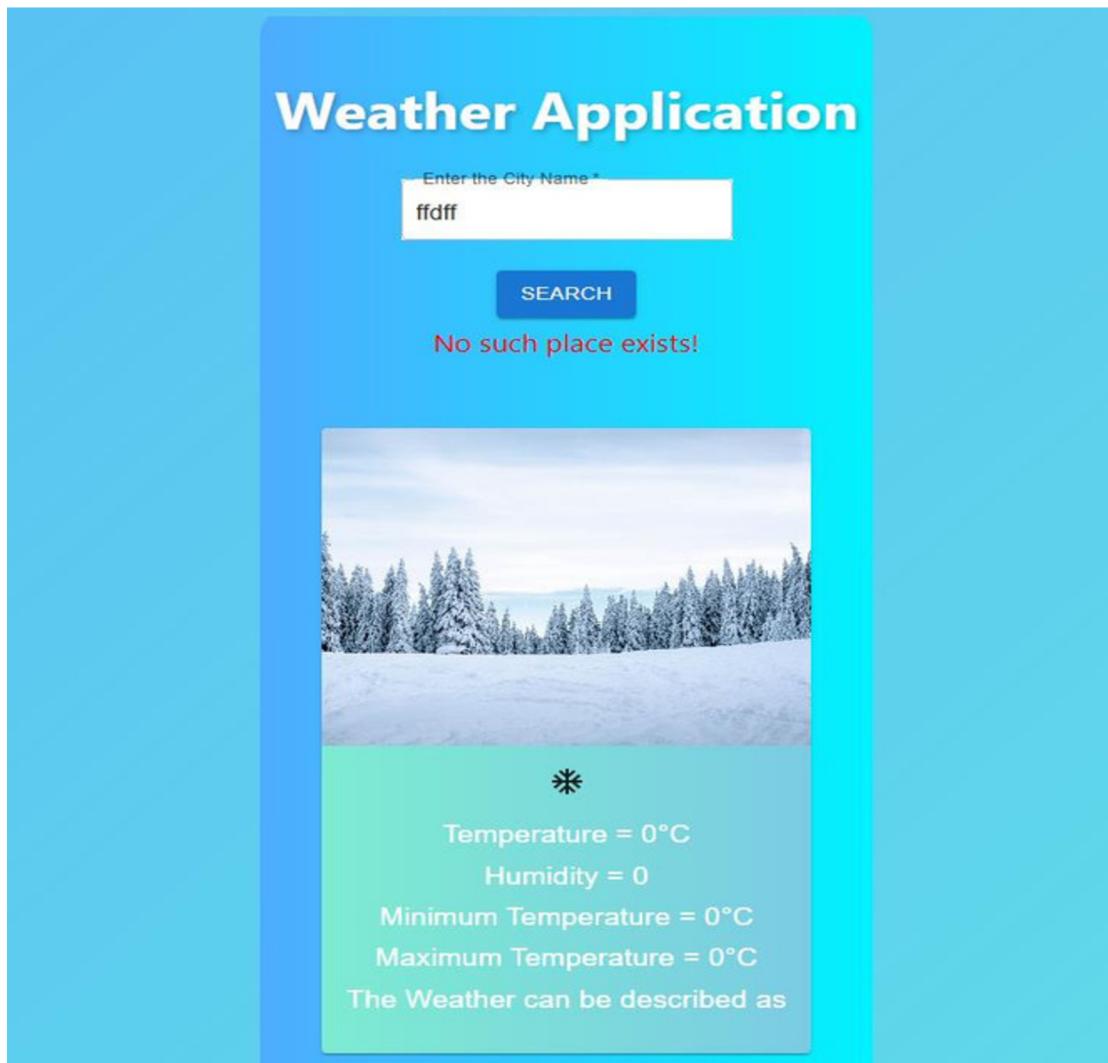
Solution: Utilized React hooks ('useState', 'useEffect') and props to pass and update data effectively.

OUTPUT

Screenshot:1 Weather details for 'Moradabad'



Screenshot:2 Error message on invalid city name



Screenshot:3 Mobile view of the application

