IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Real-Time Weather Detection

Miss. Bhumika Balu Nale
Godavari Institute of Management and Research, Jalgaon India
Under the guidance of
Prof. Pradnya Baviskar
Godavari Institute of Management and Research, Jalgaon India

Abstract: *This paper presents the design and development of a Real-Time Weather Detection System utilizing HTML, CSS, JavaScript, and API integration. The system connects to the OpenWeatherMap API to collect and display current weather data for any selected location. The project highlights the integration of API-based services into front-end development, showcasing how client-side technologies can effectively handle real-time data visualization for user convenience. By emphasizing simplicity, responsiveness, and accuracy, the project contributes to the field of web-based environmental information systems, demonstrating how modern web technologies can support real-time data accessibility and user engagement.*

Keywords: Real-Time Weather Detection, OpenWeatherMap API, HTML, CSS, JavaScript, Weather Forecasting

I. INTRODUCTION

Weather plays an important role in our daily life. Knowing the weather in advance helps people plan their work, travel, and outdoor activities. The Real-Time Weather Detection Project is a small web-based system that shows live weather information such as temperature, humidity, pressure, and wind speed of any city using the OpenWeatherMap API.
The Real-Time Weather Detection Project provides a practical implementation of this idea. By using the OpenWeatherMap API, the project collects live weather data and displays it dynamically on a web page built with HTML, CSS, and JavaScript. The API provides structured weather data in JSON format, which can be easily parsed and displayed.

1.1 Background and Basics

In old times, weather prediction was done using traditional instruments like barometers and thermometers. Today, with the help of modern technology and the internet, we can get weather information instantly.
The OpenWeatherMap API provides free access to real-time weather data from around the world. It sends weather details in JSON format, which can easily be used in web applications through programming. Our project uses this API to display weather information based on the user's city input.

1.2 Motivation

Nowadays, everyone wants instant and accurate weather updates. Many websites and mobile apps provide this information but are full of advertisements or unnecessary details.
This motivated us to create a simple and easy-to-use weather app that shows only the important weather details in real time.

1.3 Problem Definition

Most people face difficulties in getting quick and clear weather updates. Some apps take time to load, while some show too many details.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Problem Statement:

To design a simple web-based system that shows real-time weather details of any city by using the OpenWeatherMap API and displays the information in a clear and user-friendly way.

II. PROJECT PLANNING AND MANAGEMENT

Project Planning and Management is the most important phase of the Software Development Life Cycle (SDLC).
It involves defining the project scope, setting objectives, preparing schedules, allocating resources, estimating cost, and managing risks to ensure that the project is completed successfully within time and budget.
Proper planning helps in achieving the project goals efficiently and ensures that all tasks are carried out in a systematic manner.

2.1 Objective

To build a web application that detects and displays live weather data.
To use OpenWeatherMap API for fetching weather information.
To display data in a clean and responsive web page.
To learn API integration and JSON data handling.
To help users get quick and accurate weather updates.

2.2 Feasibility Study

A feasibility study is conducted before the development process begins.
It helps determine whether the project is practical, achievable, and worth investing time and resources in.
It examines different factors like technical, operational, economic, and time feasibility.

(a) Technical Feasibility

Technical feasibility checks if the current technology, software, and hardware can support the system.
In this project, the OpenWeatherMap API provides real-time weather data in JSON format which can be easily integrated using PHP and JavaScript. Tools such as XAMPP, HTML, CSS, JavaScript, and PHP are used all of which are open-source and easily available.

(b) Operational Feasibility

Operational feasibility examines how well the proposed system works in the real environment.
The Real-Time Weather Detection system provides an interactive and user-friendly interface where users can simply enter the name of a city to get updated weather details like temperature, humidity, wind speed, and condition.

(c) Economic Feasibility

Economic feasibility (also known as cost-benefit analysis) ensures that the project is cost-effective.
This project uses free software and APIs, so the total cost of development is very low. There are no licensing fees or server charges involved. The only minor cost is internet usage and documentation. Therefore, it is economically feasible for small developers or students.

(d) Time Feasibility

Time feasibility checks whether the project can be completed within the available time period.
Since the project involves limited modules such as input, API fetching, display, and report generation, it can be completed within 10–15 days if properly scheduled. The short development time makes the project time-feasible.
An easy way to comply with the Journal paper formatting requirements is to use this document as a template and simply type your text into it.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

2.3 Risk Analysis

Risk analysis is the process of identifying possible problems that may affect the success of the project and planning ways to reduce or manage them. It is important to handle risks at an early stage to avoid failure or loss later.

Types of Risks and Their Management:

Risk Description	Type of Risk	Impact	Possible Solution / Mitigation
API not responding or server downtime	Technical	Medium	Implement fallback handling or display last fetched data
Internet connection issues	Operational	High	Use offline storage or cached data
Incorrect data or delayed response from API	External	Medium	Validate data and refresh automatically
Programming errors during development	Technical	High	Conduct regular debugging and testing
Lack of time or improper scheduling	Managerial	Medium	Follow project schedule strictly
Browser compatibility issues	Technical	Low	Test the application in multiple browsers

Risk Management Strategy:

Risks are continuously monitored throughout the project life cycle. Proper documentation, version control, and testing are maintained to reduce unexpected problems.

2.4 Project Scheduling

Project scheduling defines the timeline for various tasks and helps ensure that the project is completed on time. It divides the project into small, manageable activities and assigns start and end dates for each.

Schedule:

Phase	Activity Description	Duration (Days)
Phase 1	Requirement Gathering and Analysis	12
Phase 2	System Design (UI layout, page structure)	5
Phase 3	API Integration and Backend Coding	16
Phase 4	Testing and Debugging	7
Phase 5	Documentation and Final Presentation	5

Total Duration: 45 Days

2.5 Cost Estimation

Cost estimation predicts the total expense required to complete the project. It includes hardware, software, internet, and other miscellaneous costs. Estimating cost helps to keep the project within budget and avoid overspending.

Estimated Cost Table:

Item	Description	Estimated Cost (₹)
Hardware	Computer/Laptop (already available)	0
Software	XAMPP, Browser, Text Editor	Free

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

API Service	OpenWeatherMap (Free Tier)	0
Internet	For API and development work	200
Documentation	Printing, Stationery, Binding	100
Miscellaneous	Backup, maintenance	100

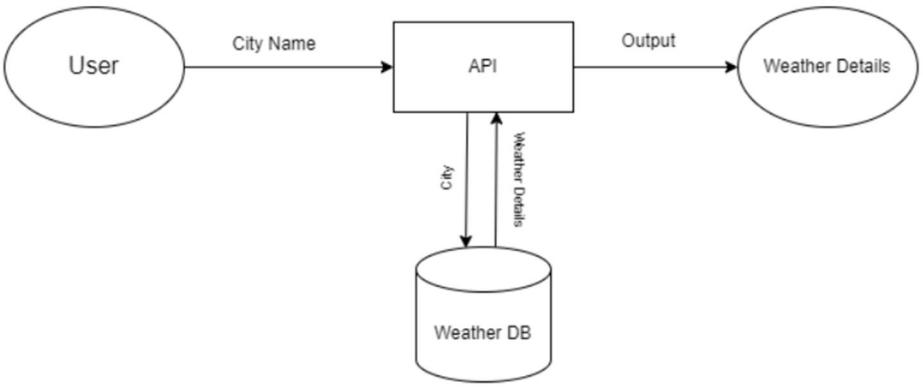
Total Estimated Cost: ₹400

III. SYSTEM DESIGN

3.1 System Architecture

The architecture follows a client-server model.
Client Side: The user interacts with a browser interface.
Server Side: The OpenWeatherMap API processes user requests.
Communication: The browser sends HTTP GET requests with the city name; the API responds with JSON weather data.

3.2 Data Flow Diagram



3.3 Components

HTML: Provides structure and layout of the web page.
CSS: Adds design, color, and responsiveness.
JavaScript: Handles data fetching, parsing, and dynamic updates.
OpenWeatherMap API: Provides the real-time data source.

IV. IMPLEMENTATION

The implementation phase is where the actual development of the system takes place.
It is the stage where all the designs and plans are converted into working code.
The main goal of this phase is to develop a functional and reliable software product that meets all the requirements defined in the analysis phase.
During this phase, the developer writes code, integrates different modules, connects with the OpenWeatherMap API, and tests each component to ensure the system works correctly.

4.1 Algorithm / Steps:

The algorithm or step-by-step process for the Real-Time Weather Detection System.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Algorithm:

Start the program.
Accept input from the user (city name).
Create an API request URL by combining the base URL, API key, and city name.
Send the HTTP request to the **OpenWeatherMap API** server.
Receive the JSON response containing the weather details.
Parse the JSON data to extract information like:
Temperature
Humidity
Wind Speed
Pressure
Weather Condition
Display the weather information on the webpage in a user-friendly format.
If the city name is invalid or API fails, display an error message.
End the program.

V. TESTING

5.1 Testing Methodologies

Black Box Testing: Tested by giving different city names and checking output.
White Box Testing: Verified JavaScript functions, API calls, and data flow logic.

5.2 Test Cases

Test Case	Input	Expected Output	Actual Output	Result
TC01	Jalgaon	Weather details for Jalgaon	Same	Pass
TC02	London	Real-time data for London	Same	Pass
TC03	Invalid City	Error message	Error displayed	Pass
TC04	Empty Input	Prompt for input	Prompt shown	Pass

VI. RESULTS AND DISCUSSION

The developed system successfully fetches and displays weather information in real time. It performs consistently across browsers like Chrome, Edge, and Firefox. The interface is visually appealing, responsive, and easy to navigate.
During testing, the system provided accurate data within seconds. The use of asynchronous JavaScript eliminated page reloads, enhancing user experience. Additionally, the project demonstrates the practical integration of APIs, showcasing how open-source data can be used to build real-world applications efficiently.
This project also serves as an educational example for web development students to understand API communication, JSON parsing, and DOM manipulation.

VII. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The Real-Time Weather Detection System was successfully designed, developed, and implemented using the OpenWeatherMap API. The main objective of the project was to provide real-time and accurate weather information to users in a simple, fast, and user-friendly manner and this goal has been effectively achieved.
The system allows users to enter any city name and instantly receive details such as temperature, humidity, wind speed, and weather conditions. The use of open-source web technologies like HTML, CSS, JavaScript, and PHP makes the project lightweight, cost-effective, and easily maintainable.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Through testing and validation, it was observed that the system performed efficiently, displaying accurate data with quick response times. The design of the interface is intuitive, ensuring that even non-technical users can operate it without difficulty.

Overall, the project successfully demonstrates how API-based integration can be used to develop real-time, data-driven web applications. It also showcases the importance of real-time information systems in our daily lives, especially for planning travel, agriculture, and outdoor activities.

7.2 Future Scope

Forecasting Feature:

Extend the system to display 5-day or 7-day weather forecasts using advanced API features.

Graphical Representation:

Include charts or graphs to visualize temperature, humidity, and pressure trends over time.

Location Detection:

Use GPS or browser-based geolocation to automatically detect the user's current location and display the weather without manual input.

Mobile Application Integration:

Develop a mobile app version using Android or Flutter to make the system accessible on smartphones.

Notification and Alerts System:

Implement features that send weather alerts or notifications for extreme conditions such as heavy rainfall, storms, or heatwaves.

Multilingual Support:

Add multiple language options so users from different regions can use the system easily.

AI-based Prediction:

Integrate artificial intelligence or machine learning to predict upcoming weather changes based on historical data.

Enhanced Security and Data Storage:

Store user preferences securely and provide options for personalized dashboards and saved locations.

VIII. ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who contributed to the successful completion of my research paper titled **"Real-Time Weather Detection Project using the OpenWeatherMap API"**.

First and foremost, I would like to thank my **guide Prof.Pradnya Baviskar**, for their valuable guidance, continuous support, and constructive feedback throughout the course of this project. Their expertise and encouragement greatly enhanced the quality of my work.

I am also grateful to the **Godavari Institute of Management and Research, Jalgaon**, for providing the necessary resources and a supportive learning environment to carry out this research successfully.

Special thanks to my classmates and friends for their helpful discussions, moral support, and collaboration during this project.

Finally, I would like to express my heartfelt appreciation to my **family** for their constant encouragement, understanding, and motivation, which inspired me to complete this work with dedication and enthusiasm.

This research paper is a result of collective effort, guidance, and support, and I extend my sincere gratitude to everyone who played a part in making it possible.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

REFERENCES

[1]. OpenWeatherMap API Documentation: https://openweathermap.org/api
[2]. W3Schools Tutorials: https://www.w3schools.com/
[3]. MDN Web Docs: https://developer.mozilla.org/
[4]. GeeksforGeeks –https://www.geeksforgeeks.org/
– For tutorials related to API integration, error handling, and data processing.
[5]. R. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill Education