A

Synopsis

on

Weather Forcasting

in partial fulfillment of the requirement for the degree
of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MURALIDHAR (2301331530098)
AMARJEET KUMAR (2301331530022)

Under the supervision of

Mr. Rajat Kumar

(Asst. Professor ,CSE)

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

GREATER NOIDA 2025-26

Index

Supervisor Sign:

1. Introduction

1.1 Background and Overview

Weather plays a crucial role in human activities, ranging from agriculture, travel aviation, tourism, and disaster management to the simple planning of daily routines. A farmer depends on weather updates for irrigation scheduling, a traveller checks the forecast before planning a trip, and government authorities rely on weather data for early disaster warnings.

With the advancement of technology, weather forecasting has become more precise. However, delivery of weather data to end-users still faces challenges. Many existing apps provide too much technical data, are bloated with advertisements, or require premium subscriptions for complete features. This reduces accessibility, usability, and inclusiveness.

The Pro Weather App has been developed to overcome these issues by focusing on:

- Simplicity – clean interface with only relevant details.
- Accuracy – powered by the Open Weather Map API.
- Responsiveness – adapts to mobile, tablet, and desktop devices.
- Visual Engagement – dynamic background images reflecting real-time conditions.

This project is an academic yet practical implementation of web technologies (HTML5, CSS3, JavaScript) and API integration. It emphasizes frontend development, real-time data handling, user interface design,

and error management, making it valuable for students, developers, and general users alike.

1.2 Objective of the Project

The objectives of the Pro Weather App are:

1. To develop lightweight, responsive, and user-friendly weather forecasting application.
2. To integrate OpenWeatherMap API for fetching real-time weather updates.
3. To provide visual representation of weather conditions through adaptive background images.
4. To ensure cross-device compatibility using responsive web design.
5. To demonstrate academic application of modern web technologies.

1.3 Scope of the Project

- End users (students, travelers, general public).
- Developers (demonstration of API and JavaScript integration).
- Educational purposes (mini-projects and academic reports).
- Real-time application (daily life weather checking).

1.4 Contribution of the Project

- Provides ad-free, accessible, and free weather data.
- Serves as a learning resource for web technology students.
- Offers potential for future scalability and enhancements (GPS, multi-day forecasts, voice assistance).

2. Literature Review / Existing Systems

Weather applications have been studied extensively for their role in delivering climate and forecast data to users. Some major systems include:

System	Features	Limitations
AccuWeather	Detailed forecasts, maps, alerts	Ads, premium subscription needed
Weather.com	Hourly/daily forecasts, radar	Cluttered UI, slow load times
Yahoo Weather	Visual appeal, Flickr images	Limited accuracy, basic free tier
Google Weather	Fast results via search	Minimal detail, lacks customization

Gaps Identified:

- Poor balance between data accuracy and user experience.
- Heavy reliance on advertisements.
- Limited personalization and responsiveness.

Pro Weather App fills this gap by offering a clean, responsive, ad-free interface powered by a reliable API.

3. Problem Statement

The challenges with existing weather systems are:

- Complexity: Interfaces overloaded with data and ads.
- Accessibility issues: Some require subscriptions for detailed reports.
- Visualization gap: Many lack contextual visuals (rain → rainy theme).
- Error handling: Users often see confusing API errors when entering invalid cities.
- Responsiveness: Not all systems adapt well to smaller devices.

Problem

Definition:
There is a need for a lightweight, accessible, and visually appealing weather app that provides real-time, accurate, and easy-to-understand information for everyday users.

4. Proposed Methodology

4.1 Modules

1. City Input Module
   - Accepts user input, validates city name.
   - Prevents blank submissions.
2. API Integration Module
   - Fetches weather data from OpenWeatherMap using Fetch API.
   - Extracts temperature, weather description, country, and city details.

3. Dynamic Background Module
    - Background changes (sunny, rainy, cloudy, cold) for visual realism.
4. Error Handling Module
    - Shows descriptive errors like *"City not found"*.
5. Responsive UI Module
    - Uses CSS media queries and flexible layouts for all devices.

4.2 Diagrams (suggested for report)

- Use Case Diagram:
    - User → Enter City → App → Fetch API → Display Results.
- DFD (Level 0):
    - User Input → Weather App → API Request → API Response → Display Output.
- ER Diagram (for academic purpose):
    - Entities: User, City, Weather Data.
    - Attributes: temperature, condition, timestamp.

4.3 Algorithm (Pseudocode)

```
BEGIN
  INPUT city_name
  IF city_name IS NOT EMPTY
```

```
    CALL OpenWeatherMap API
    IF response = success
      DISPLAY temperature, condition, location
      CHANGE background image
    ELSE
      DISPLAY "City not found"
  ELSE
    DISPLAY "Enter city name"
END
```

5. Feasibility Study

5.1 Technical Feasibility

- Built with HTML, CSS, JS (widely supported).
- OpenWeatherMap API ensures reliable data.
- Hosting possible on free cloud platforms.

5.2 Operational Feasibility

- Easy for any user (enter city → get results).
- Minimal learning curve.
- Can be extended for institutional or commercial use.

5.3 Economic Feasibility

- Free to build using open-source tools.
- API has a free tier suitable for student projects.
- Deployment cost negligible on GitHub Pages or Netlify.

6. Facilities Required

Software Requirements

- HTML5, CSS3, JavaScript
- API: OpenWeatherMap
- Editor: Visual Studio Code
- GitHub/Netlify for hosting

Hardware Requirements

- Developer System: Intel i5/i7, 8GB RAM, 100GB storage.
- Deployment Server: Cloud-based, minimal specs (2 vCPU, 4GB RAM).

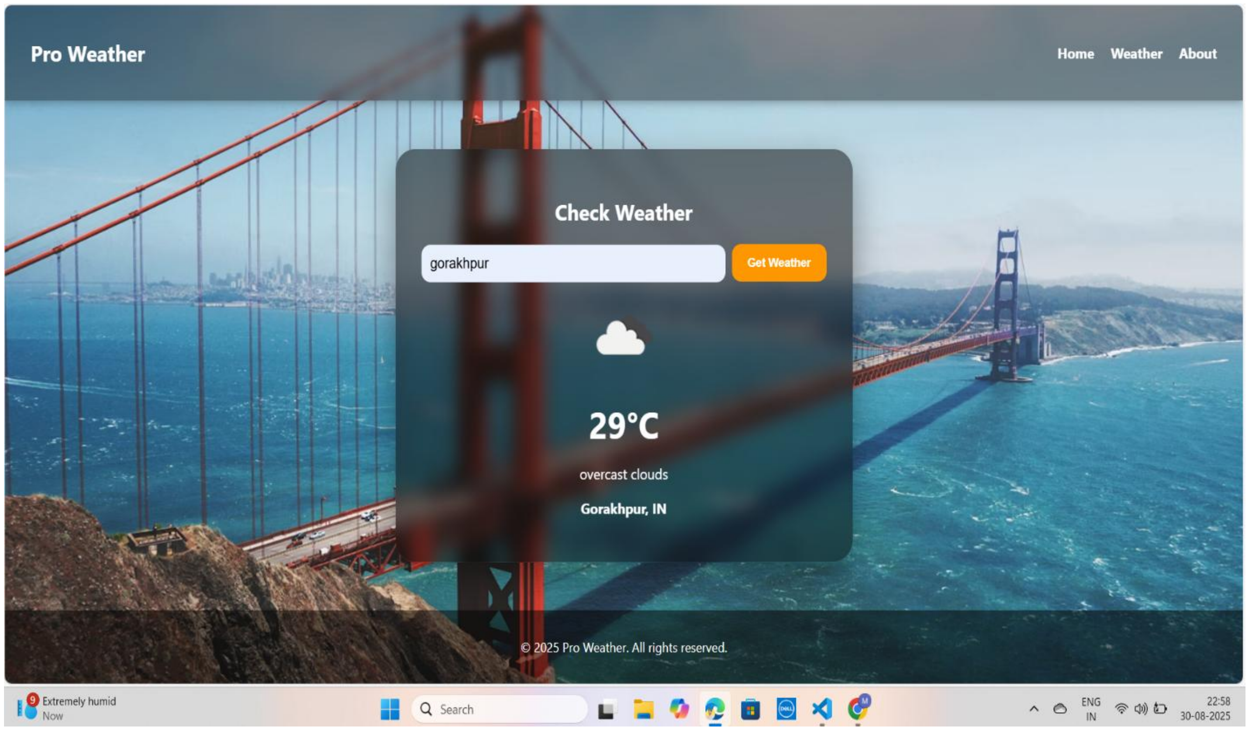
Other Requirements

- High-speed internet.
- Modern browser (Chrome/Edge/Firefox).

7. System Design & Implementation

- Architecture: Client-side app communicating with REST API.
- Request-Response Cycle:
  - User input → Fetch Request → API Response → DOM Update.
- **Implementation Phases:**
  - Phase 1: Basic UI with input form.
  - Phase 2: API integration.
  - Phase 3: Dynamic visuals.

- Phase 4: Error handling & responsiveness.



8. Testing & Results

Test Cases Table

Test Case	Input	Expected Output	Result
Valid City	“Delhi”	Weather data displayed	Pass
Invalid City	“Xyzabc”	Error: City not found	Pass
Empty Input	“”	Error: Enter city name	Pass
Cloudy Weather	“London”	Cloudy BG shown	Pass

The app successfully passed all test cases.

9. Conclusion & Future Enhancements

The Pro Weather App demonstrates how modern web technologies and APIs can deliver real-time visually appealing, and accurate weather forecasts.
It balances simplicity, accuracy, and aesthetics.

Future Enhancements:

- Multi-day forecast (5–7 days).
- GPS-based auto-location detection.
- Voice-enabled search.
- Mobile app version (React Native/Flutter).
- AI-based prediction for severe weather events.

10. References

1. OpenWeatherMap API Documentation
2. Mozilla Developer Network – HTML, CSS, JavaScript Documentation
3. W3Schools – Web Tutorials
4. Unsplash – Weather Background Images
5. Visual Studio Code – Official Docs
6. Netlify & GitHub Pages Hosting Guides
7. Research papers on Web-based Weather Applications (IJRTE, IEEE)