WeatherVista — Real-Time Weather Monitoring Dashboard

A PROJECT REPORT

Submitted by

Ayush Kohli (23BCS11238)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Chandigarh University

Nov, 2025

BONAFIDE CERTIFICATE

Certified that this project report "**WeatherVista — Real-Time Weather Monitoring Dashboard"** is the bonafide work of **"Ayush Kohli"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. Sandeep Singh Kang	
BATCH HEAD	**SUPERVISOR**
BE-CSE	**BE-CSE**

TABLE OF CONTENTS

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Name:Ayush Kohli
UID: 23BCS11238
Branch: BE-CSE
Section/Group: KRG-3B
Subject name: Full Stack - 1
Subject code: 23CSH-339

1. PROJECT NAME

WeatherVista — Real-Time Weather Monitoring Dashboard

2. REFERENCE WEBSITE LINK

https://openweathermap.org/api

Other References:

- React Documentation (react.dev)
- Spring Boot REST API (spring.io)
- Recharts.js (recharts.org) for visualization

3. PROJECT DESCRIPTION

WeatherVista is an interactive **real-time weather dashboard** built using **React (frontend)** and **Java Spring Boot (backend)** that provides accurate weather updates, forecasts, and visual analytics. The app fetches live data from the **OpenWeather API** and presents it through an engaging, responsive, and animated interface with temperature graphs, humidity/rain charts, and dynamic themes (clear, cloudy, rainy).

Tech Stack Overview:

- **Frontend:** React.js + Axios + Recharts + React Icons
- **Backend:** Spring Boot (Java), RESTful APIs
- **Database (optional):** MySQL for user/city history (extendable)
- **API Integration:** OpenWeatherMap API

4. PROBLEM STATEMENT

Weather data is available through APIs but often lacks **visual appeal** and **user interactivity**.

Users need a unified dashboard that:

- Shows **real-time temperature, humidity, wind speed**, etc.
- Provides **hourly and daily trends** with visual graphs.
- Delivers **aesthetic visualization** with responsive, animated UI.

WeatherVista solves this by integrating weather APIs with a modern full-stack design to provide clear, interactive, and beautiful real-time weather analytics.

5. HIGH LEVEL DESIGN

Frontend

- **Technology:** React.js, HTML5, CSS3, JavaScript (ES6+), Recharts, Axios
- **Features:**
  - Responsive and animated UI with glassmorphism
  - Dynamic charts for temperature and humidity
  - Custom weather-based background animations

Backend

- **Technology:** Java Spring Boot (v3.x)
- **Endpoints:**
  - /api/weather?city={city} → returns current weather
  - /api/weather/forecast?city={city} → returns 5-hour forecast
- **Responsibilities:**
  - Consume OpenWeather API
  - Handle business logic
  - Serve data to React frontend

6. Key Features

User Features

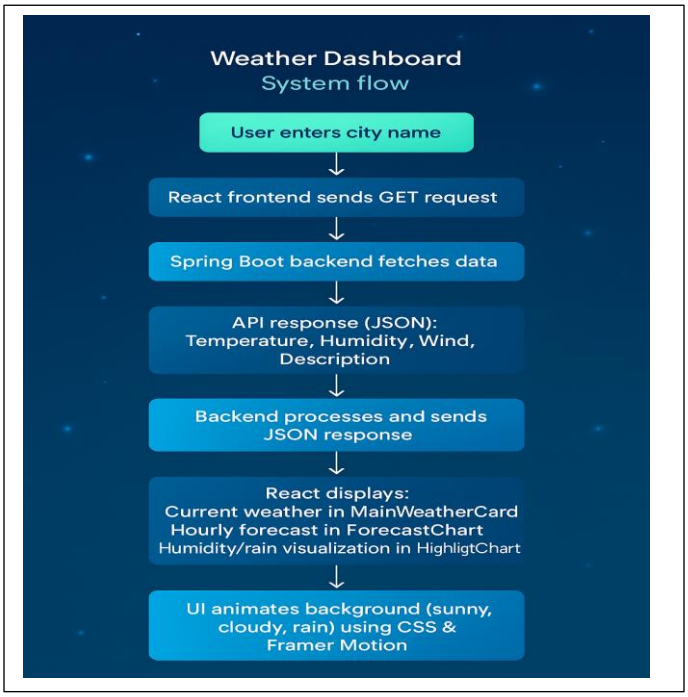
- **City Search:** Get instant weather by city name.
- **Forecast Graph:** Visualizes temperature trends (next 8 hours).
- **Highlights:** Displays humidity, rain probability, and wind speed.
- **Dynamic Theme:** Background and animations change with weather type.
- **Dual Chart Visualization**: Line + bar combo for humidity/rain patterns.

Technical Features

- REST API integration with OpenWeather.
- Axios-based frontend data fetching.
- React Hooks for state management.
- Spring Boot service layer for backend logic.
- Lightweight, animated CSS effects (aurora glow & particles).

SYSTEM FLOW

1. User enters a city name on the dashboard.
2. React frontend sends a GET request → http://localhost:8080/api/weather?city={city}.
3. Spring Boot backend fetches live data from OpenWeather API.
4. Backend processes and sends a JSON response with temperature, humidity, wind, and description.
5. React displays:
   - Current weather in the MainWeatherCard.
   - Hourly forecast in ForecastChart.
   - Humidity/rain visualization in HighlightsChart.
   -
6. UI animates background (sunny, cloudy, rain) using CSS & Framer Motion



FUTURE SCOPE

1. Add Geo-location detection (fetch user's current weather automatically).
2. Implement weekly forecast with interactive charts.
3. Integrate MySQL to store favorite cities or last searches.
4. Add AI-based weather prediction model using historical data.
5. Build mobile app version using React Native

.

6. Include air quality index (AQI) and UV index data.
7. Add 3D weather animations for a more immersive experience

Conclusion

WeatherVista demonstrates a modern, full-stack weather visualization system that merges **real-time API data**, **Spring Boot backend services**, and a **React-based animated UI**.
It provides users with a visually stunning and interactive way to explore weather data while showcasing key full-stack skills — RESTful backend development, API integration, and data-driven frontend visualization.

This project is ideal as a **Full Stack mini-project or portfolio piece**, highlighting practical API handling, UI/UX design, and integration between Java backend and React frontend.

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

ScreenShot