“Weather Wise AI – A Full Stack Weather App”

A Project Report submitted in partial Fulfillment of the Requirements for an internship at Elevate Labs.

By,

Raj Shantaram Parsharam

Zeal College of Engineering & Research, Pune

2024-25

Abstract

Weather Wise AI is a full stack web application that allows users to search weather data by entering a city name. The backend fetches real-time weather information from a weather API and displays it on the frontend. A unique feature is AI-powered summarization: when the user clicks the “Summarize with AI” button, the backend sends the weather details to the Gemini AI API which generates a concise 100-word summary and returns it to the UI. The frontend, designed using React.js and Lottie animations, ensures a smooth and engaging user experience. The project is deployed live on Render.

Introduction

The role of technology in our daily lives has increased significantly. One such example is weather forecasting applications, which help people plan their daily activities. Weather Wise AI enhances the user experience by not only showing raw weather data but also providing a concise AI-generated summary. Problem Definition: Traditional weather apps often present too much technical information. Users require a short, meaningful summary to understand quickly.

Objectives: -

- Enable users to check live weather data by entering a city name.
- Provide concise AI summaries for easier understanding.
- Demonstrate full stack development using React and Spring Boot.
- Deploy a production-ready application on Render.

System Analysis

System Requirements:

- Hardware: Minimum i3 Processor, 4GB RAM, 500MB storage, internet connection.
- Software: React.js, Spring Boot, Java, Render. Feasibility, Gemini
- Technical: Uses well-supported open-source technologies.
- Operational: Simple user interface ensures ease of use.
- Economic: Free-tier deployment on R
- ender minimizes cost.

System Design

System Architecture: The system is divided into three layers

frontend (React.js with Lottie), backend (Spring Boot), and AI layer (Gemini API).

Flow of Execution: User enters city name → Backend fetches weather data → Data displayed → User clicks Summarize → Gemini AI generates summary → Sent back to UI.

Database Design: Although primarily API-based, the system can be extended with a database for logging queries and storing user search history

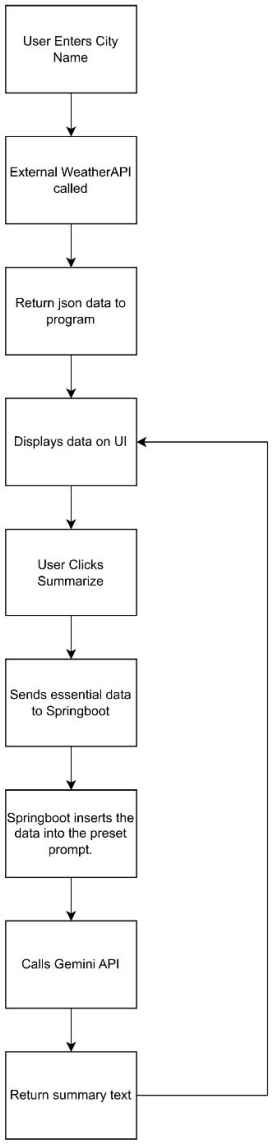

Fig. Flow of Program

Implementation

Frontend Implementation:

Developed using React.js with components for input, display, and AI summarization. Axios is used for API calls. Lottie animations enhance visual experience. Eclipse IDE uses with jdk 21.0.7.

Backend Implementation:

Built using Spring Boot with REST endpoints for handling weather data requests and AI summarization requests. Typed code in Vs code editor.

AI Integration:

The Gemini AI API is called with a crafted prompt, and the response is processed into a user-friendly 100-word summary. Get the API key from Google Gemini and implemented with Spring AI using the latest Spring AI documentation and Gemini documentation.

Deployment:

Both frontend and backend deployed on Render, ensuring live availability. Having a separate repo that is integrated with Render.

Testing:

For API testing Postman tool is used. Every react component is test after build.

Results

Results:

- Successful fetching of weather data by city name.
- Summarization feature returns accurate 100-word AI summaries.
- Lottie animations display seamlessly.
- Deployment tested successfully on Render with live URL.
- Error handles correctly with toast.

Live Demo:

https://weatherwiseai-frontend.onrender.com/

Live Screenshots

Conclusion & Future Scope

Conclusion:

Weather Wise AI successfully integrates real-time weather information with AI-based summarization, enhancing user experience.

Future Scope:

- User authentication and personalization.
- Multi-language support for summaries.
- Voice assistant integration.
- Data storage for analytics and historical trends
- Weather map

References

1. React.js Documentation – https://reactjs.org/
2. Spring Boot Documentation – https://spring.io/projects/spring-boot
3. Render Deployment – https://render.com/
4. Gemini AI API Documentation – Google AI
5. OpenWeatherMap API – https://openweathermap.org/