Java Developer Internship

Organization: Elevate Labs

Aug – Sep 2025

Weather Forecast App – Project Report

Introduction

The Weather Forecast App is a desktop-based application designed to provide real-time weather information to users in a visually appealing and interactive way. Developed in Java using JavaFX for the user interface, it integrates with the OpenWeatherMap API to fetch live weather data. Users can enter the name of a city, and the app instantly displays details such as temperature, weather conditions, and corresponding icons. Additionally, the app uses dynamic background themes that adjust to different weather conditions such as sunny, cloudy, rainy, and snowy days. This creates a smooth and engaging experience for the user while learning the basics of API integration and UI design in Java.

Abstract

This project aims to demonstrate how real-world applications can be built using JavaFX for graphical interfaces combined with external APIs for data retrieval. The Weather Forecast App showcases essential software development practices such as making HTTP requests, parsing JSON responses, and handling exceptions gracefully. By utilizing the Google Gson library, the JSON response from the OpenWeatherMap API is processed and displayed in a clean format on the user interface. The project highlights the importance of modular programming, user experience design, and effective integration of third-party services. Beyond being an academic exercise, this project mirrors real-world industry applications where live data is used to deliver value to end users.

Tools Used

- **Programming Language:** Java
- **Framework:** JavaFX (for building the graphical interface)
- **API:** OpenWeatherMap API (for fetching live weather data)
- **Library:** gson (for parsing JSON responses)
- **IDE:** IntelliJ IDEA / Eclipse / NetBeans - **Server Runtime:** JDK 8 or later

Steps Involved in Building the Project

1. Environment Setup: Install JDK and configure JavaFX libraries. Add Gson library to handle JSON.

2. User Interface Design: Create input fields, buttons, and a weather card layout using JavaFX components such as VBox, HBox, Label, and ImageView.

3. API Integration: Use Java's HttpURLConnection to connect with the OpenWeatherMap API. Send requests with the city name and API key.

4. Data Extraction: Parse the JSON response using the Gson library to retrieve temperature, description, humidity, and weather icon code.

5. Dynamic UI Update: Update labels, icons, and background colors dynamically according to weather conditions (sunny, cloudy, rainy, snowy).

6. Error Handling: Display user-friendly error messages in case of invalid city names, failed connections, or API errors.

7. Testing: Test the application by entering different cities and checking results for accuracy and UI responsiveness.

8. Deployment: Package the application for end users to run on desktop systems with minimal setup.

Conclusion

The Weather Forecast App demonstrates how desktop applications can provide real-time services by integrating APIs and building intuitive interfaces. It highlights fundamental skills such as user interface design with JavaFX, handling HTTP requests, parsing JSON data, and presenting information in a user-friendly manner. This project also underlines the significance of user experience by dynamically changing themes based on weather conditions. The application can be further extended to show 5-day forecasts, provide weather alerts, or be adapted into a mobile version. Overall, the project is a practical and valuable exercise that bridges academic learning with real-world application development.