Project Report

1) Weather Forecast App

Introduction

The Weather Forecast App is designed to fetch and display real-time weather information using the **OpenWeatherMap API**. It demonstrates how external APIs can be integrated with a Java-based application. A Spring Boot REST API fetches and serves weather details, and optionally, a JavaFX UI can display them visually.

Abstract

The app consumes real-time weather data for a given city, parses JSON responses, and presents it to the user in a clean interface. It also handles invalid input and network issues gracefully. This project showcases the integration of Java, APIs, JSON parsing, and RESTful services.

Tools Used

- **Java 11+/17+**
- **Spring Boot** (for backend REST API)
- **JavaFX** (for optional GUI front-end)
- **Gson/Jackson** (for JSON parsing)
- **OpenWeatherMap API** (for live weather data)
- **Eclipse/IntelliJ** (IDE)

Steps Involved in Building the Project

1. **Register & API Key** – Obtain API key from OpenWeatherMap.
2. **Controller & Service** – Create a REST endpoint /api/weather/{city} using WeatherController and WeatherService.
3. **HTTP Request** – Fetch live weather data from OpenWeatherMap using HttpURLConnection/RestTemplate.
4. **Parse JSON** – Extract fields like city, temperature, condition, and icon code using Gson/Jackson.
5. **UI Display** – Show results in JavaFX UI or return JSON response in Spring Boot API.
6. **Error Handling** – Handle cases like invalid city names, API errors, or no internet connection.

Conclusion

The Weather Forecast App highlights how to consume third-party APIs in Java projects and present real-time data effectively. It is scalable, can be extended for multi-day forecasts, and demonstrates both backend (Spring Boot) and frontend (JavaFX) integration.