Weather Forecast App

Title : Online Quiz System

Name : BUKYA NANDINI

Institution: Mohan Babu University

Objective: Fetch and display real-time weather using public API.

Tools: Java, JavaFX, OpenWeatherMap API

**Introduction**

The Weather Forecast App is a desktop application built using Java and JavaFX that enables users to fetch and display real-time weather information for any city. Utilizing the OpenWeatherMap API, the app retrieves current weather data such as temperature, humidity, weather description, and displays corresponding icons to provide a clear, user-friendly experience. This project serves as an excellent introduction to API integration, GUI development, and JSON parsing in Java.

**Abstract**

This project demonstrates how to develop a Java-based application that interacts with an external weather API to obtain live data. The app features a simple and intuitive GUI where users input a city name to receive current weather details. It showcases asynchronous network requests, JSON data handling using the Gson library, and dynamic UI updates with JavaFX. Error handling is implemented to manage invalid city entries or network failures. The project emphasizes clean architecture, usability, and effective API consumption.

**Tools Used**

* **Java 11+**: Programming language for application logic.
* **JavaFX**: For building the graphical user interface.
* **OpenWeatherMap API**: Provides real-time weather data.
* **Gson**: Library used for parsing JSON data.
* **HttpURLConnection**: For HTTP requests.
* **IDE (Eclipse/IntelliJ)**: Development environment for coding and debugging.

**Steps Involved in Building the Project**

1. **API Registration**: Created an account on OpenWeatherMap and obtained a free API key.
2. **Setup JavaFX UI**: Designed the user interface including text input for city names, buttons, labels, and an image view to display weather icons.
3. **Network Request Handling**: Implemented HTTP GET requests using HttpURLConnection to fetch weather data from the OpenWeatherMap endpoint.
4. **JSON Parsing**: Used the Gson library to parse JSON responses and extract relevant fields such as temperature, humidity, weather description, and icon codes.
5. **UI Update**: Updated the JavaFX UI dynamically using Platform.runLater() to ensure thread safety.
6. **Error Management**: Added error handling to manage invalid inputs, API errors, and connection issues by displaying appropriate messages.
7. **Testing & Debugging**: Conducted tests for various city inputs and network scenarios to ensure robustness.

**Conclusion**

The Weather Forecast App successfully integrates a public weather API with a Java desktop interface to provide users with real-time weather information. The project demonstrates fundamental concepts of API interaction, multithreading, JSON parsing, and GUI updates in JavaFX. It provides a practical experience in building user-friendly applications that communicate with external services. Future enhancements could include adding a multi-day forecast, geolocation support, or a more sophisticated UI design.