WeatherAPI.java

```
// WeatherAPI.java
package com.weatherapp.api;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;

public class WeatherAPI {
    private static final String API_KEY = "e421f607f8064a25f51ccbl69ae06c81";
    private static final String BASE_URL = "https://api.openweathermap.Org/data/2.5/weather";

    public static JsonObject getWeatherData(String city) {
        try {
            String urlString = BASE_URL + "?q=" + city + "&appid=" + API_KEY;
            URL url = new URL(urlString);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");

            BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
            StringBuilder response = new StringBuilder();
            String line;

            while ((line = reader.readLine()) != null) {
                response.append(line);
            }
            reader.close();

            JsonParser parser = new JsonParser();
            return parser.parse(response.toString()).getAsJsonObject();

        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

WeatherUtils.java

```
// WeatherUtils.java
package com.weatherapp.utils;

public class WeatherUtils {
    public static double kelvinToCelsius(double kelvin) {
        return kelvin - 273.15;
    }

    public static double kelvinToFahrenheit(double kelvin) {
        return (kelvin - 273.15) * 9/5 + 32;
    }
}
```

WeatherAppGULjava

```
// WeatherAppGUI.java
package com.weatherapp.gui;

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.*;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import com.google.gson.JsonObject;
import com.weatherapp.api.WeatherAPI;
import com.weatherapp.utils.WeatherUtils;
import java.time.LocalTime;
import java.util.ArrayList;
import java.util.List;

public class WeatherAppGUI extends Application {
```

```
private VBox historyBox = new VBox();
private List<String> searchHistory = new ArrayList<>();
private boolean isCelsius = true;
private Label temperatureLabel = new Label();
private Label humidityLabel = new Label();
private Label windSpeedLabel = new Label();
private Label conditionLabel = new Label();
private ImageView weatherIcon = new ImageView();
private BorderPane root = new BorderPane();

@Override
public void start(Stage primaryStage) {
    TextField cityInput = new TextField();
    cityInput.setPromptText("Enter city name");
    Button searchButton = new Button("Search");
    Button toggleUnitButton = new Button("Switch °C/°F");

    searchButton.setOnAction(e -> fetchWeather(cityInput.getText()));
    toggleUnitButton.setOnAction(e -> toggleTemperatureUnit());

    VBox infoBox = new VBox(10, temperatureLabel, humidityLabel, windSpeedLabel, conditionLabel, weatherIcon);
    infoBox.setPadding(new Insets(10));

    VBox leftPanel = new VBox(10, new Label("Search History:"), historyBox);
    leftPanel.setPadding(new Insets(10));
    leftPanel.setPrefWidth(200);

    VBox inputBox = new VBox(10, cityInput, searchButton, toggleUnitButton);
    inputBox.setPadding(new Insets(10));

    root.setTop(inputBox);
    root.setCenter(infoBox);
    root.setLeft(leftPanel);

    updateBackground();

    Scene scene = new Scene(root, 600, 400);
    primaryStage.setTitle("Weather Information App");
    primaryStage.setScene(scene);
    primaryStage.show();
}

private void fetchWeather(String city) {
    if (city == null || city.trim().isEmpty()) {
        showAlert("Input Error", "City name cannot be empty.");
        return;
    }

    JsonObject data = WeatherAPI.getWeatherData(city);
    if (data == null || !data.has("main")) {
        showAlert("API Error", "Unable to fetch weather data.");
        return;
    }

    double tempKelvin = data.getAsJsonObject("main").get("temp").getAsDouble();
    double temp = isCelsius ? WeatherUtils.kelvinToCelsius(tempKelvin) : WeatherUtils.kelvinToFahrenheit(tempKelvin);
    int humidity = data.getAsJsonObject("main").get("humidity").getAsInt();
    double wind = data.getAsJsonObject("wind").get("speed").getAsDouble();
    String condition = data.getAsJsonArray("weather").get(0).getAsJsonObject().get("main").getAsString();

    temperatureLabel.setText("Temperature: " + String.format("%.1f", temp) + (isCelsius ? " °C" : " °F"));
    humidityLabel.setText("Humidity: " + humidity + "%");
    windSpeedLabel.setText("Wind Speed: " + wind + " m/s");
    conditionLabel.setText("Condition: " + condition);
    weatherIcon.setImage(new Image(getClass().getResourceAsStream("/icons/" + condition.toLowerCase() + ".png")));

    String timestamp = java.time.LocalDateTime.now().toString();
    searchHistory.add(city + " at " + timestamp);
    updateHistory();
}

private void toggleTemperatureUnit() {
    isCelsius = !isCelsius;
}

private void updateHistory() {
    historyBox.getChildren().clear();
    for (String entry : searchHistory) {
        historyBox.getChildren().add(new Label(entry));
    }
}

private void updateBackground() {
    LocalTime now = LocalTime.now();
    if (now.isAfter(LocalTime.of(18, 0)) || now.isBefore(LocalTime.of(6, 0))) {
        root.setBackground(new Background(new BackgroundFill(Color.DARKSLATEBLUE, CornerRadii.EMPTY, Insets.EMPTY)));
    } else {
        root.setBackground(new Background(new BackgroundFill(Color.LIGHTSKYBLUE, CornerRadii.EMPTY, Insets.EMPTY)));
    }
}
```

```
    private void showAlert(String title, String message) {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle(title);
        alert.setContentText(message);
        alert.showAndWait();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Please turn over

Weather Information App – README

Overview

The Weather Information App is a desktop application built in Java using JavaFX. It provides real-time weather data for cities around the world using the OpenWeatherMap API. The app features a clean user interface, dynamic visuals, and user-friendly controls.

Features

- • Real-Time Weather: Displays temperature, humidity, wind speed, and condition.
- • Search Input: Enter any city name to get weather data.
- • Temperature Unit Toggle: Switch between Celsius (°C) and Fahrenheit (°F).
- • Weather Icons: Shows an image based on current weather (e.g., sun, cloud, rain).
- • Error Handling: Alerts if input is invalid or API fails.
- • Search History: Keeps a list of previously searched cities with timestamps.
- • Dynamic Background: Background color changes depending on time (day/night).

Requirements

- • Java Development Kit (JDK) – version 11 or above
- • JavaFX SDK – for GUI support
- • Gson Library – for parsing JSON data
- • Internet Connection – for API access

Project Structure

com.weatherapp.api → WeatherAPI.java
com.weatherapp.utils → WeatherUtils.java
com.weatherapp.gui → WeatherAppGUI.java
resources/icons/ → Contains PNG images (e.g., clear.png, clouds.png)

Setup Instructions (For Eclipse)

- • Open Eclipse IDE
- • Import Project:
  Go to File > Import > Existing Projects into Workspace
  Select the folder where the source code is located

- • Add JavaFX and Gson:
  Right-click project → Build Path > Configure Build Path
  Add gson.jar and JavaFX libraries under External JARs
- • Run App:
  Navigate to WeatherAppGUI.java
  Right-click → Run As > Java Application

Example Output

If you search for London, you might see:
Temperature: 14.2 °C
Humidity: 65%
Wind Speed: 3.4 m/s
Condition: Clouds

Icons Note

Make sure you place weather icons in a folder called 'icons' inside 'resources'. Filenames should be lowercase (e.g., clear.png, rain.png, clouds.png).

Troubleshooting

Issue	Solution
JavaFX error	Check that JavaFX is set up in Run Config
Missing icons	Add appropriate .png files to icons folder
API not responding	Check your API key and internet connection

License