# ****Weather App GUI****

## ****Introduction****

This is a simple Weather App that I made using Java. It shows real-time weather information of any city that the user types. For example, if someone writes "Lahore", the app connects to an online weather service, gets the data, and shows it on the screen. It tells you about the temperature, weather condition (like sunny or cloudy), humidity, and wind speed.

I have used **Java Swing** to create the graphical interface of the app. This means that the app opens in a window with buttons, text boxes, and labels, just like any normal desktop application.

## ****Technologies and Tools Used****

**Java 18** – The programming language I used to create this application.

**Java Swing** – This is the GUI (Graphical User Interface) technology in Java which I used to create the windows, buttons, and layout.

**HTTPURLConnection** – A built-in Java class that helped me connect with the online weather service to fetch data.

**JSON Simple Library** – This is a small library that helped me read the weather data, which comes in JSON format.

**OpenWeatherMap API** – I used this online service to get weather information of cities. It gives data in JSON, which I read and display.

## ****How the App Works****

* 1.When you open the app, it shows a window with a text box.
* 2.You can type the name of any city in the box.
* 3.Then you press a button.
* 4.The app connects to the internet and searches for the weather of that city.
* 5.It collects temperature, wind, humidity, and condition (like Rainy or Sunny).
* 6.The app then shows that weather information on the screen in an easy way.
* 7.It’s like a small Google Weather, but made by me using Java.

## ****Main Parts of the Project****

### ****1. AppLauncher****

This is the starting point of the app.When we run the project, this class runs first and starts the GUI (the app window).

### ****2. WeatherAppGui****

1. This class is responsible for creating the interface.It includes all the visible parts like:
2. The text field (where user writes the city)
3. The button (to fetch weather)
4. The labels (where weather details are shown)
5. It controls what happens when you click the button and how the weather data is shown.

### ****3. WeatherApp****

1. This class handles all the background work.
2. It connects with the weather API, fetches the JSON data, and processes it.
3. It converts that data into useful information (like temperature in °C and weather type like "Cloudy").
4. Then it sends that information back to the GUI for display.

## ****What This App Show****

1. Current Temperature
2. Weather Condition (like Sunny, Rainy, Cloudy)
3. Wind Speed (in meters per second)
4. Humidity (in percentage)

**CREATED BY:**

* **MUHAMMAD HOZAIFA NAEEM (FA24-BCS-049)**