1. Project Setup

- Choose **JavaFX** (modern and feature-rich) or **Swing** (simpler and widely used).
- Use **Maven** or **Gradle** for dependency management.
- Install **JDK 17+** and set up your IDE (**IntelliJ IDEA** or **Eclipse**).

---

2. API Integration

- **OpenWeatherMap API** (register and get an API key).
- Use `HttpURLConnection` or `OkHttp` to make API requests.
- Parse JSON responses using **Jackson** or **Gson**.

---

3. GUI Design (JavaFX or Swing)

- **Input Field:** Users enter a city name.
- **Search Button:** Triggers API request.
- **Weather Display:** Labels showing temperature, humidity, wind speed, conditions.
- **Icons:** Represent different weather conditions (e.g., ☀️ for sunny).
- **Unit Conversion:** Celsius ↔ Fahrenheit, m/s ↔ km/h.
- **Dynamic Backgrounds:** Adjust background based on time.

---

4. Features Implementation

✅ **API Handling:** Fetch real-time weather.
✅ **Input Validation:** Prevent empty or invalid entries.
✅ **Error Handling:** Handle API failures, network errors, or invalid locations.

✅ **History Tracking:** Store past searches with timestamps in a local file.
✅ **Forecast Display:** Show 3-hourly or daily forecasts.

5. Code Example (JavaFX)

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Scanner;
import org.json.JSONObject;

public class WeatherApp extends Application {
    private TextField cityField;
    private Label weatherLabel;

    @Override
    public void start(Stage stage) {
        cityField = new TextField();
        cityField.setPromptText("Enter City Name");

        Button searchButton = new Button("Get Weather");
        searchButton.setOnAction(e -> fetchWeather());

        weatherLabel = new Label("Weather info will appear here");

        VBox layout = new VBox(10, cityField, searchButton, weatherLabel);
        Scene scene = new Scene(layout, 300, 200);
        stage.setScene(scene);
        stage.setTitle("Weather App");
        stage.show();
    }

    private void fetchWeather() {
        String city = cityField.getText().trim();
        if (city.isEmpty()) {
            weatherLabel.setText("Please enter a city!");
            return;
        }
```

```
        String apiKey = "YOUR_OPENWEATHERMAP_API_KEY";
        String urlString = "https://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid="
+ apiKey + "&units=metric";

        try {
            URL url = new URL(urlString);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.connect();

            Scanner sc = new Scanner(url.openStream());
            StringBuilder inline = new StringBuilder();
            while (sc.hasNext()) {
                inline.append(sc.nextLine());
            }
            sc.close();

            JSONObject json = new JSONObject(inline.toString());
            double temp = json.getJSONObject("main").getDouble("temp");
            String condition =
json.getJSONArray("weather").getJSONObject(0).getString("description");

            weatherLabel.setText("Temperature: " + temp + "°C\nCondition: " + condition);

        } catch (IOException e) {
            weatherLabel.setText("Error fetching weather data.");
        }
    }

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