**User Documentation for the Enhanced Weather App**

**Overview**

The Enhanced Weather App is a simple, web-based application built with Python and Flask. It allows users to retrieve current weather information for any city worldwide. The app features:

* A text input for custom city searches.
* Quick-access buttons for predefined cities.
* Temperature unit selection (Celsius or Fahrenheit).
* Additional weather details, including humidity and wind speed.

This documentation explains how to use the app effectively.

**Features**

**1. Search Weather by City Name**

* Enter the name of any city in the text box (e.g., "London," "New York") and click **Get Weather**.
* The app retrieves the current weather for the entered city.

**2. Predefined City Buttons**

* Quickly fetch weather data for common cities by clicking a button:
  + **New York**
  + **London**
  + **Tokyo**
  + **Paris**

**3. Temperature Unit Selection**

* Choose the desired temperature unit from the dropdown:
  + **Celsius (°C)**: Default option for most regions.
  + **Fahrenheit (°F)**: Commonly used in the United States.

**4. Weather Details**

* The app displays:
  + City Name
  + Temperature
  + Weather Description (e.g., "Clear sky")
  + Weather Icon (e.g., a sun or cloud image)
  + Humidity Percentage
  + Wind Speed (in meters per second or miles per hour, depending on the unit selected)

**5. Error Handling**

* If the city name is invalid, the app displays an error message:  
  *"City not found. Please try again."*
* For empty city input, an error is shown:  
  *"Please enter a city name."*

**Getting Started**

**Accessing the App**

1. Launch the application by running the following command in your terminal:
2. python app.py
3. Open a web browser and navigate to:
4. http://127.0.0.1:5000/

**Using the App**

**Step 1: Search for a City**

1. **Enter a city name**:
   * Type the name of the city (e.g., "London") into the text box.
2. Select a temperature unit (Celsius or Fahrenheit) from the dropdown menu.
3. Click **Get Weather**.

**Step 2: Quick Access Cities**

1. Below the form, you will see buttons for predefined cities.
2. Click a button (e.g., **London**) to get the weather for that city without typing.

**Step 3: Viewing Weather Details**

After submitting a query, the app displays:

1. **City Name**: The queried city.
2. **Temperature**: Current temperature in the selected unit.
3. **Weather Icon**: A visual representation of the weather (e.g., sun, clouds).
4. **Weather Description**: A text description of the weather (e.g., "Clear sky").
5. **Humidity**: Current humidity percentage.
6. **Wind Speed**: Wind speed in meters/second (m/s) or miles/hour (mph), depending on the unit selected.

**Step 4: Handling Errors**

1. If no city is entered, the app prompts:  
   *"Please enter a city name."*
2. If an invalid city is entered, the app shows:  
   *"City not found: [Entered City]. Please try again."*

**Examples**

**Example 1: Search for "London"**

* Enter "London" in the text box.
* Select **Celsius** and click **Get Weather**.
* The app displays:
  + Temperature: 15°C
  + Weather: "Cloudy"
  + Humidity: 70%
  + Wind Speed: 3 m/s

**Example 2: Use Predefined City Button**

* Click the **New York** button.
* The app automatically retrieves the weather for New York without requiring input.

**Example 3: Switch to Fahrenheit**

* Enter "Tokyo" in the text box.
* Select **Fahrenheit** from the dropdown menu.
* Click **Get Weather**.
* The app displays:
  + Temperature: 68°F
  + Weather: "Sunny"
  + Humidity: 60%
  + Wind Speed: 10 mph

**Troubleshooting**

**Common Issues**

1. **Invalid City Name**:
   * Ensure correct spelling (e.g., "São Paulo," not "Sao Paolo").
   * For cities with spaces, simply type them (e.g., "New York").
2. **API Key Issues**:
   * If weather data doesn’t load, ensure the API key is valid and active.
3. **Application Not Running**:
   * Ensure Python and Flask are installed:
   * pip install flask requests
4. **Unit Mismatch**:
   * Verify the unit selected (Celsius or Fahrenheit).

**Extending the App**

**Developers can:**

1. **Add More Cities**: Modify the index.html template to include more quick-access buttons.
2. **Forecast Data**: Use the OpenWeatherMap Forecast API to provide a 5-day weather forecast.
3. **Geolocation**: Auto-detect the user’s location and show weather data for their city by default.
4. **Mobile Compatibility**: Enhance the UI with responsive design for better mobile usability.

**Support**

If you encounter issues, check:

* OpenWeatherMap FAQ: <https://openweathermap.org/faq>
* Flask Documentation: <https://flask.palletsprojects.com/>

For further assistance, feel free to contact support or share logs for debugging.