Weather Information App using Python & API

# Overview:

This project is a Python-based script designed to retrieve and display real-time weather information for any location provided by the user. It uses the wttr.in API to fetch the weather data and applies regular expressions (regex) to extract the current temperature in Celsius. Based on the temperature, the script categorizes the day as either hot, cold, or moderate.

# Features:

- Real-Time Weather Data: Users can input any city or region to retrieve the current weather information.  
- Temperature Extraction: The script uses regex to parse and extract temperature details from the API response.  
- Weather Condition Feedback: Provides feedback based on the current temperature such as ‘hot’, ‘cold’, or ‘moderate’.  
- Simple and Clear Output: Displays the weather information in a user-friendly format.

# Technologies Used:

* **Python:** Main programming language.
* **Requests module:** Used to send HTTP requests and retrieve weather data from the wttr.in API.
* **Regular Expressions (Regex):** Used to extract temperature data from the API response.
* **API (wttr.in):** Provides schematic weather information for cities/regions.

# Code :



# Output:

# 

# OUTPUT 2:

# 

# How It Works:

1. User Input: The script asks the user to input a location (city or region).  
2. Fetching Data: Using the requests library, it sends a GET request to wttr.in, which provides the current weather details in a simple text format.  
3. Extracting Temperature: The script uses a regular expression to extract the temperature in Celsius.  
4. Categorizing the Weather: Based on the temperature, it classifies the weather as hot, cold, or moderate.  
5. Output: The weather details and corresponding feedback are printed to the console.

# Future Enhancements:

- Add graphical visualization for temperature changes.  
- Allow users to specify units (Celsius or Fahrenheit).  
- Display additional weather details like wind speed, humidity, etc.

# How to Run:

1. Install Python (if not already installed).  
2. Install the required libraries using pip:  
 pip install requests  
3. Copy the Python script into a file (e.g., weather\_app.py).  
4. Run the script:  
 python weather\_app.py  
5. Enter a location when prompted, and the weather information will be displayed.