WHETHER ANALYSIS

BY

ASWINI.M

Agenda

- Introduction
- Objective
- Code implementation
- Module used
- Output
- Future scope

Introduction

Overview of the Basic Python Weather Project

Project Overview

This project aims to create a simple yet effective weather application using Python, which fetches real-time weather data from an external API.

Purpose of the Application

The primary goal is to provide users with immediate access to weather information, enhancing their ability to plan daily activities based on current conditions.

Significance of Weather Information

Accurate weather data is crucial for various sectors, including agriculture, transportation, and event planning, as it influences decision-making processes.

Objective

Goals of the Weather Application

- **Real-Time Weather Updates:** The application is designed to fetch and display current weather conditions, ensuring users receive the most up-to-date information available.
- **Learning Basic Python Programming:** This project serves as an educational tool, allowing users to practice fundamental Python concepts such as variables, functions, and control structures.
- **Understanding API Usage:** By integrating an external weather API, users will gain hands-on experience in making HTTP requests and processing JSON data, which are essential skills in modern programming.

Photo by Timothy Ah Koy on Unsplash

Code implementation

```
import requests

def get_weather(city):
    api_key = "https://wttr.in"  # using wttr.in (no API key needed)
    url = f"{api_key}/{city}?format=j1"

    response = requests.get(url)
    data = response.json()

    # Extract important info
    area = data['nearest_area'][0]['areaName'][0]['value']
    region = data['nearest_area'][0]['region'][0]['value']
    country = data['nearest_area'][0]['country'][0]['value']
    temperature = data['current_condition'][0]['temp_C']
    weather_desc = data['current_condition'][0]['weatherDesc'][0]['value']

    print(f"\nWeather Report for {area}, {region}, {country}")
    print("----------------------------------------")
    print(f"Temperature: {temperature}°C")
    print(f"Condition: {weather_desc}")
    print("----------------------------------------")

# Get input from user
city_name = input("Enter city name: ")
get_weather(city_name)
```

Modules used

Key Libraries for Functionality

Requests Module

This module is essential for making HTTP requests to the weather API, enabling the application to fetch real-time data effortlessly.

JSON Module

The JSON module is utilized to parse and handle the JSON data returned by the API, allowing for easy extraction of relevant information.

Role of Each Module

Each module plays a critical role in the application, with requests handling data retrieval and JSON managing data formatting and accessibility.

Output

```
Enter city name: chennai

Weather Report for Chennai, Tamil Nadu, India
----------------------------------------
Temperature: 30°C
Condition: Patchy rain nearby
----------------------------------------
```

Results of the Weather Application

Displaying Program Output

The output of the program presents the current weather conditions in a user-friendly format, making it easy to read and understand.

Example Output for Chennai

For instance, the application retrieves and displays temperature, humidity, and wind speed for Chennai, providing users with essential weather metrics.

Explaining the Output Format

The output is structured to highlight key weather indicators, ensuring that users can quickly grasp the current conditions at a glance.

Future scope

Enhancements and New Features

- **Potential Improvements:** Future iterations of the project could include enhancements such as a more sophisticated user interface and additional weather metrics.
- **Adding Forecast Features:** Incorporating weather forecasts would provide users with not only current conditions but also predictions for upcoming days, enhancing the application's utility.
- **Utilizing Different APIs:** Exploring various weather APIs could enrich the data available to users, offering more comprehensive insights into weather patterns and trends.

Photo by Jonathon Young on Unsplash