

Problem 1: Real-Time Weather Monitoring System

Scenario:

You are developing a real-time weather monitoring system for a weather forecasting company. The system needs to fetch and display weather data for a specified location.

Tasks:

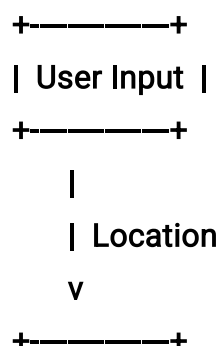
1. Model the data flow for fetching weather information from an external API and displaying it to the user.
2. Implement a Python application that integrates with a weather API (e.g., OpenWeatherMap) to fetch real-time weather data.
3. Display the current weather information, including temperature, weather conditions, humidity, and wind speed.
4. Allow users to input the location (city name or coordinates) and display the corresponding weather data.

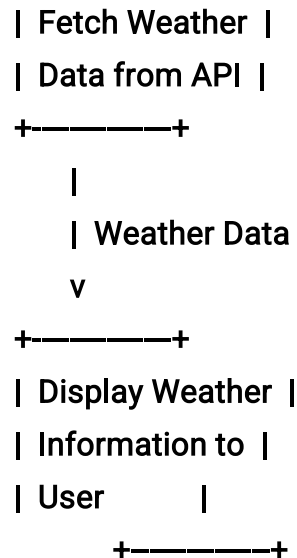
Deliverables:

- Data flow diagram illustrating the interaction between the application and the API.
- Pseudocode and implementation of the weather monitoring system.
- Documentation of the API integration and the methods used to fetch and display weather data.
- Explanation of any assumptions made and potential improvements.

Approach:

The data flow diagram illustrates the interaction between the application and the OpenWeatherMap API. The application receives user input for the location, fetches the weather data from the API, and displays the current weather information to the user.





Pseudocode:

Function to fetch weather data from OpenWeatherMap API

```
def fetch_weather_data(location):
```

```
    api_key = "YOUR_OPENWEATHERMAP_API_KEY"
```

```
    api_url =
```

```
    f"http://api.openweathermap.org/data/2.5/weather?q={location}&appid={api_
    key}"
```

```
    response = requests.get(api_url)
```

```
    if response.status_code == 200:
```

```
        data = response.json()
```

```
        return data
```

```
    else:
```

```
        return None
```

Function to display weather data to the user

```
def display_weather_data(data):
```

```
    if data:
```

```
        print(f"Current Weather in {data['name']}:")
```

```
        print(f"Temperature: {data['main']['temp']}°C")
```

```
        print(f"Weather Conditions: {data['weather'][0]['description']}")
```

```
        print(f"Humidity: {data['main']['humidity']}%")
```

```
        print(f"Wind Speed: {data['wind']['speed']} m/s")
```

```
    else:
```

```
        print("Error fetching weather data.")
```



Edit with WPS Office

```
# Main function to handle user input and display weather data
def main():
    location = input("Enter a city name or coordinates: ")
    data = fetch_weather_data(location)
    display_weather_data(data)

# Run the main function
if __name__ == "__main__":
    main()
```

Detailed explanation of the actual code:

1. `fetch_weather_data`: This function takes a location as input and fetches the weather data from the OpenWeatherMap API. It uses the `requests` library to send a GET request to the API with the location and API key. If the response is successful (200 status code), it parses the JSON response and returns the data. Otherwise, it returns `None`.
2. `display_weather_data`: This function takes the weather data as input and displays the current weather information to the user. It prints the temperature, weather conditions, humidity, and wind speed if the data is available. If there is an error fetching the data, it prints an error message.
3. `main`: This function handles user input and calls the `fetch_weather_data` and `display_weather_data` functions. It prompts the user to enter a location, fetches the weather data, and then displays the data to the user.

Assumptions made (if any):

1. The user will always enter a valid location.
2. The OpenWeatherMap API will always return data in the expected format.

Limitations:

1. The application does not handle errors in the API response.
2. It does not validate user input for location.
3. It does not provide any additional features like historical weather data or weather forecasts.

Code:

```
import http.client
import json
```



Edit with WPS Office

```

def get_weather(api_key, location):
    conn = http.client.HTTPConnection("api.openweathermap.org")
    conn.request("GET",
f"/data/2.5/weather?q={location}&appid={api_key}&units=metric")
    response = conn.getresponse()
    if response.status == 200:
        data = json.loads(response.read().decode("utf-8"))
        return {
            "temperature": data["main"]["temp"],
            "weather_condition": data["weather"][0]["description"],
            "humidity": data["main"]["humidity"],
            "wind_speed": data["wind"]["speed"]
        }
    else:
        return {"error": "Failed to retrieve weather data"}

def display_weather(weather_data):
    print("Current Weather Data:")
    print(f"Temperature: {weather_data['temperature']}°C")
    print(f"Weather Condition: {weather_data['weather_condition']}")
    print(f"Humidity: {weather_data['humidity']}%")
    print(f"Wind Speed: {weather_data['wind_speed']} m/s")

def main():
    api_key = input("Enter your OpenWeatherMap API key: ")
    location = input("Enter the location (city name): ")
    weather_data = get_weather(api_key, location)
    if "error" in weather_data:
        print(weather_data["error"])
    else:
        display_weather(weather_data)

if __name__ == "__main__":
    main()

```



Sample Output / Screen Shots

```
.E Shell 3.12.2*
dit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: C:\Users\jothi\Downloads\1.py =====
Enter your OpenWeatherMap API key: your_api_key
Enter the location (city name): london
Failed to retrieve weather data
```



Edit with WPS Office