Assignment: Python Programming for DL

Name: S. SUVAN SENTHIL

Register Number:192324175

Department: B-TECH OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Date of Submission:17-07-2024

Problem 1: Real-Time Weather Monitoring System

Scenario:

The system fetches, processes, and displays up-to-date weather information, offering users an efficient way to stay informed about current weather conditions.

Tasks:

1. **Select a weather API service like OpenWeatherMap, Weatherstack, or Weather API. Sign up to get an API key**.
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.**

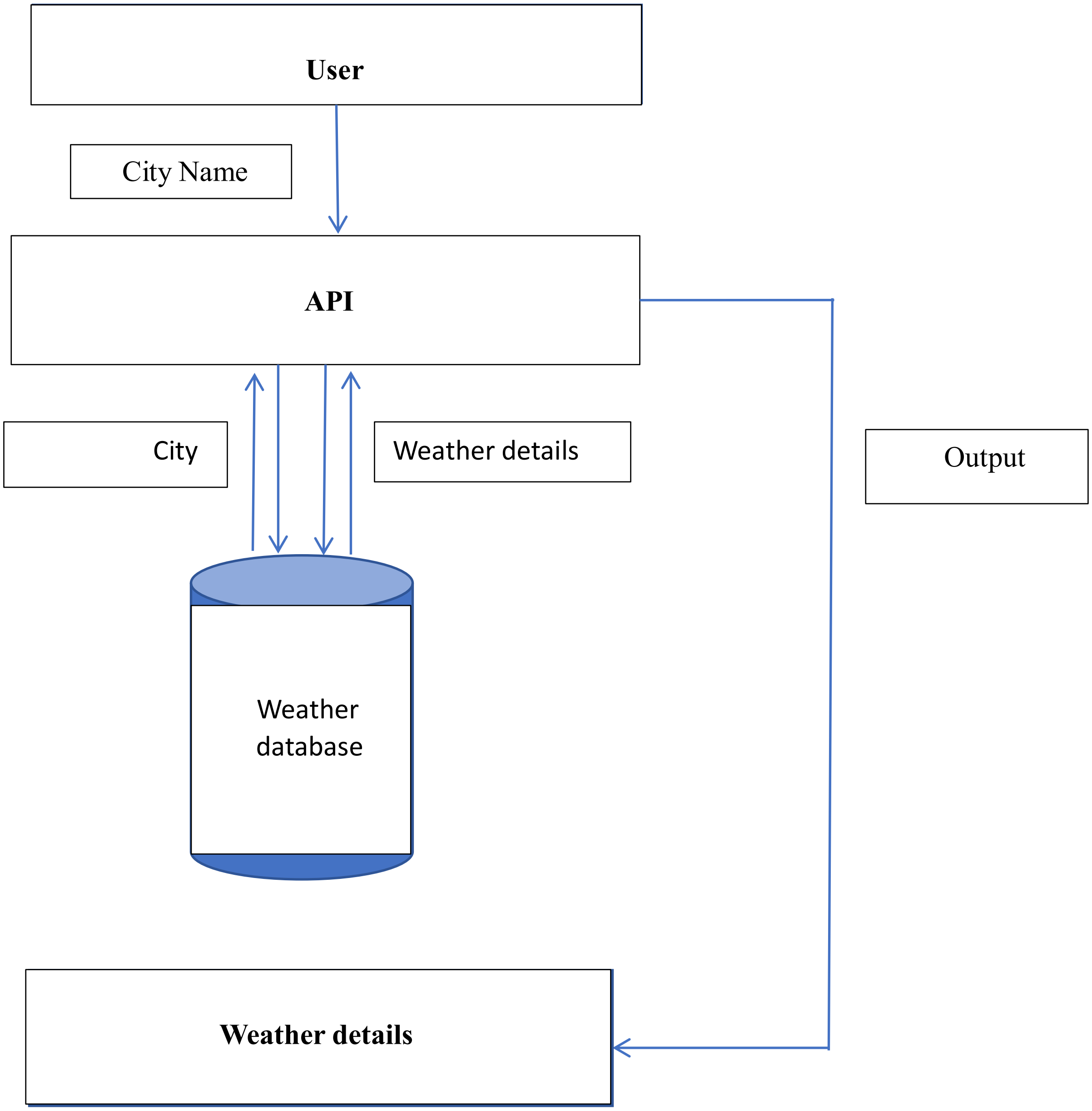
Deliverables:

- 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.

Solution:

Real-Time Weather Monitoring System

1.Data Flow Diagram



2.Implementation

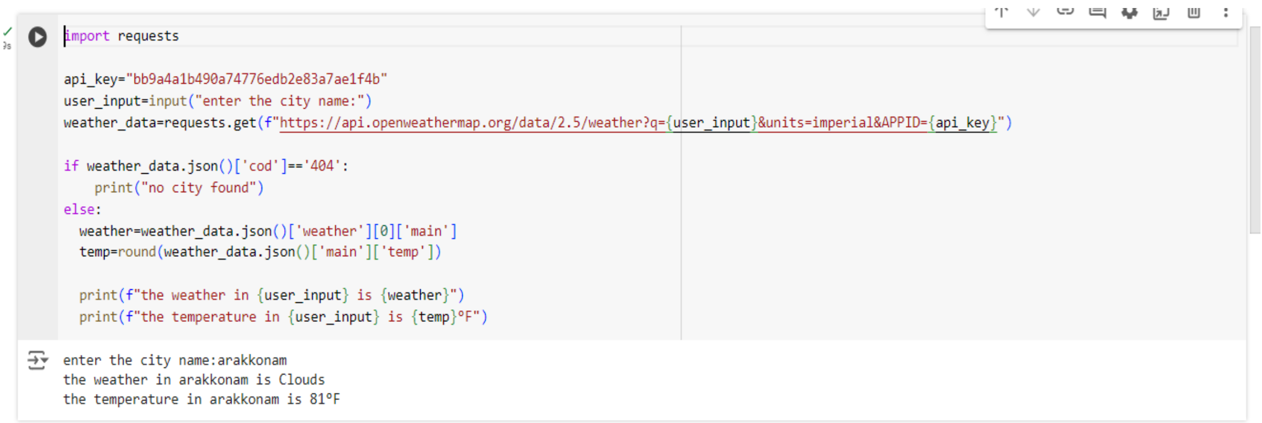
```
import requests
api key="bb9a4a1b490a74776edb2e83a7ae1f4b"
user input=input("enter the city name:")
weather_data=requests.get(f"https://api.openweathermap.org/data/2.5/
weather?q={user_input}&units=imperial&APPID={api_key}")

if weather data.json()['cod']=='404':
        print("no city found")
else:
       weather=weather_data.json()['weather'][0]['main']
       temp=round(weather_data.json()['main']['temp'])
       print(f"the weather in {user_input} is {weather}")
       print(f"the temperature in {user_input} is {temp}°F")
```

3.Display the Current weather information

enter the city: ARAKKONAM
Temperature (in FAHRENHEIT) = 82 °F
description = RAIN

4.User Input



```
import requests

api_key="bb9a4a1b490a74776edb2e83a7ae1f4b"
user_input=input("enter the city name:")
weather_data=requests.get(f"https://api.openweathermap.org/data/2.5/weather?q={user_input}&units=imperial&APPID={api_key}")

if weather_data.json()['cod']=='404':
    print("no city found")
else:
  weather=weather_data.json()['weather'][0]['main']
  temp=round(weather_data.json()['main']['temp'])

  print(f"the weather in {user_input} is {weather}")
  print(f"the temperature in {user_input} is {temp}°F")
```

```
enter the city name:arakkonam
the weather in arakkonam is Clouds
the temperature in arakkonam is 81°F
```

5.Documentation

Table of Contents

Introduction

This guide will walk you through creating a real-time weather monitoring system in Python using the Open Weather Map API. You will learn how to fetch, parse, and display weather data, and optionally plot it over time.

Set Up Your Environment

- ***Install Required Libraries***
- *First, install the required libraries. You'll need* ***requests*** *for API calls and* ***matplotlib*** *for plotting data.*

Get an API Key from OpenWeatherMap

- *Go to the OpenWeatherMap website.*
- *Sign up for a free account.*
- *Once logged in, go to the API keys section and generate a new API key.*

Fetch Weather Data

- You can use the OpenWeatherMap API to fetch real-time weather data. Here's how you can do it in Python.

Conclusion

- *Creating a real-time weather monitoring system using Python and the OpenWeatherMap API is a practical and rewarding project. By leveraging the capabilities of Python and readily available weather APIs, you can fetch, parse, and display real-time weather data for any location.*