**Assignment: Python Programming for GUI Development**

Name: M.Nirosha

Register Number: 192311411

Department: Computer science and engineering

Date of Submission: 26.08.2024

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.

**Solution:**

**Real-Time Weather Monitoring System**

**1.Data Flow Diagram**

Weather App

Weather API

User

1)Input Location

2) Fetch Current Weather Data

3) Fetch Historical Weather Data

4) Parse API Response

5) Display Data

**2. Implementation**

import requests

API\_KEY = '26f6ac4550c9cd2d2d6b010857a4685f'

def get\_weather(city):

API\_URL = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API\_KEY}&units=metric"

try:

response = requests.get(API\_URL)

response.raise\_for\_status()

if response.status\_code == 200:

data = response.json()

print(f"Current Weather in {city.capitalize()}:")

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

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

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

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

print(f"Date and Time of Data: {data['dt']}")

else:

print(f"Failed to retrieve data: {response.status\_code}, {response.text}")

except requests.RequestException as e:

print(f"An error occurred: {e}")

if \_name\_ == '\_main\_':

city = input('Enter city name: ')

get\_weather(city)

**3.Display the Current weather information**

Enter city name: bangalore

Current Weather in Bangalore:

Temperature: 23.17°C

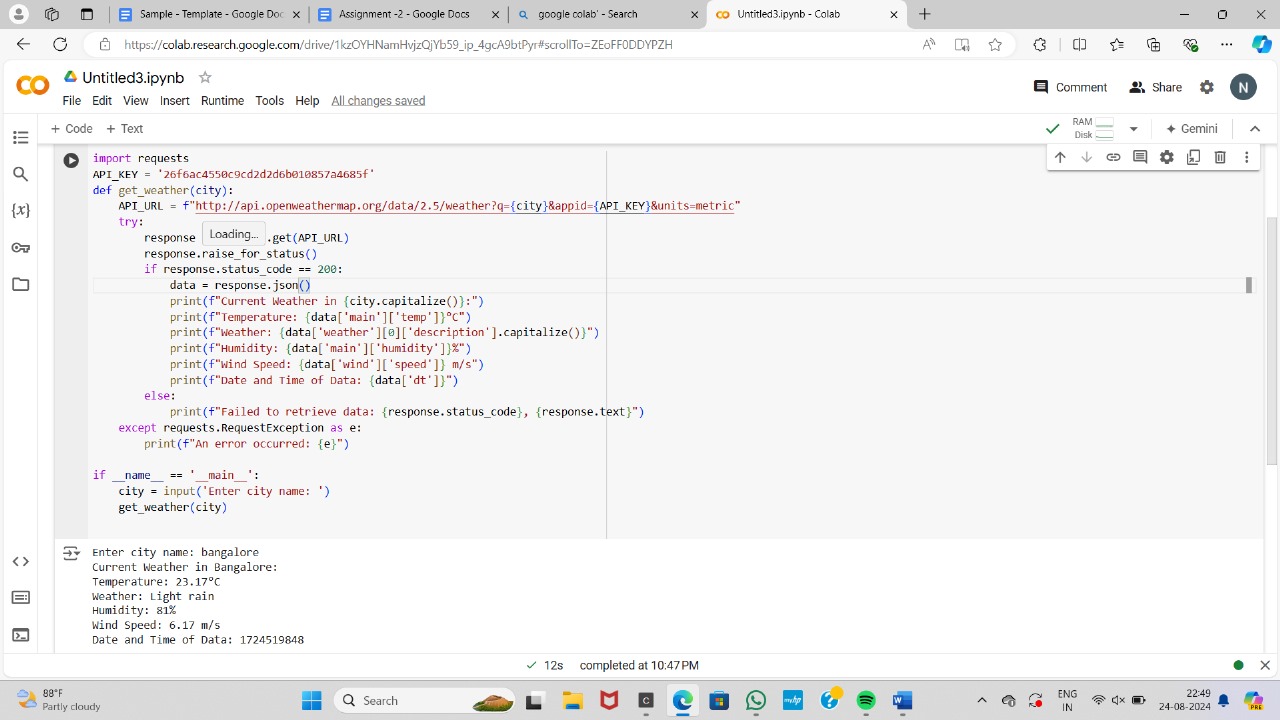
Weather: Light rain

Humidity: 81%

Wind Speed: 6.17 m/s

Date and Time of Data: 1724519848

**4.User Input**



**5.Documentation**

**Detailed explanation of the actual code:**

* 1. **Fetch weather data:** This function takes a location as input and fetches the weather data from the Open Weather Map 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.

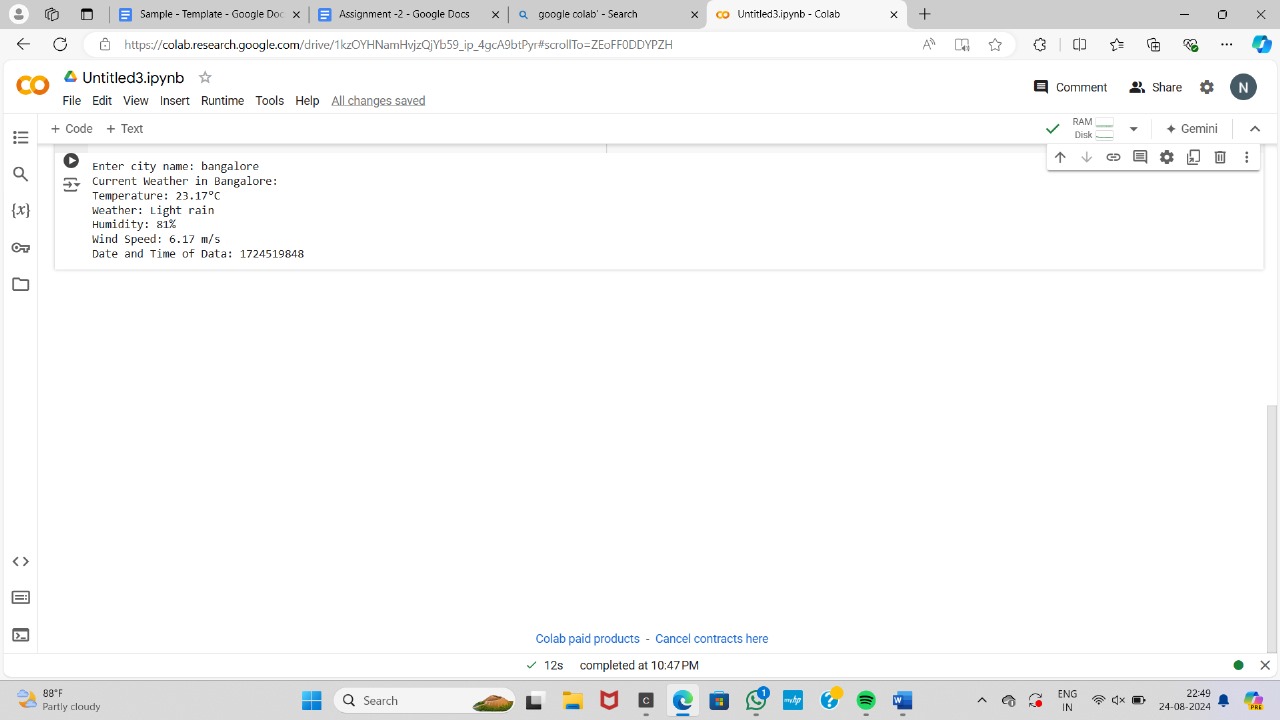
**Assumptions made (if any):**

1. The user will always enter a valid location.
2. The Open Weather Map 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.

**Sample Output / Screen Shots:**



**Graph on Number of Death Age Difference:**

Graph to visualize temperature trends over a week for a given location. We'll use the Matplotlib library for this purpose. First, we need to extend our application to fetch historical weather data for the past week. Open Weather Map provides a "One Call API" that includes historical data.

