Assignment: Python Programming for GUI Development

Name: R.Sathish

Register Number: 192324204

Department: B.tech AI &DS

Date of Submission:26/08/2024

Problem 5: AIR POLLUTION

Scenario:

An air quality monitoring app leverages an AQI (Air Quality Index) API to provide real-time data on air pollution. Users can input their location or share GPS coordinates, prompting the app to fetch the AQI along with concentrations of pollutants like PM2.5 and CO. The app then displays the AQI value, color-coded for easy interpretation, and provides health recommendations based on air quality levels. Additional details, such as pollutant levels and the source of the data, are also presented. Users can then decide whether to go outdoors or take precautions. The API supports location-based queries and returns data in a standardized format. Enhancements could include historical trends, notifications, and personalized health advice.

DELIVARABLE:

1. Project Documentation:

   - Project plan, architecture, and API details.

2. Source Code:

   - Python/JavaScript code for API integration and data display.

3. User Interface:

   - Dashboard with AQI indicators, pollutant data, and location input.

4. Deployment Package:

   - Setup instructions, config files, and Docker support (if applicable)

5. Test Cases:

   - Unit and end-to-end tests for functionality and error handling.

6. User Guide:

   - How-to guide with AQI explanations and troubleshooting tips.

7. Reporting and Analytics:

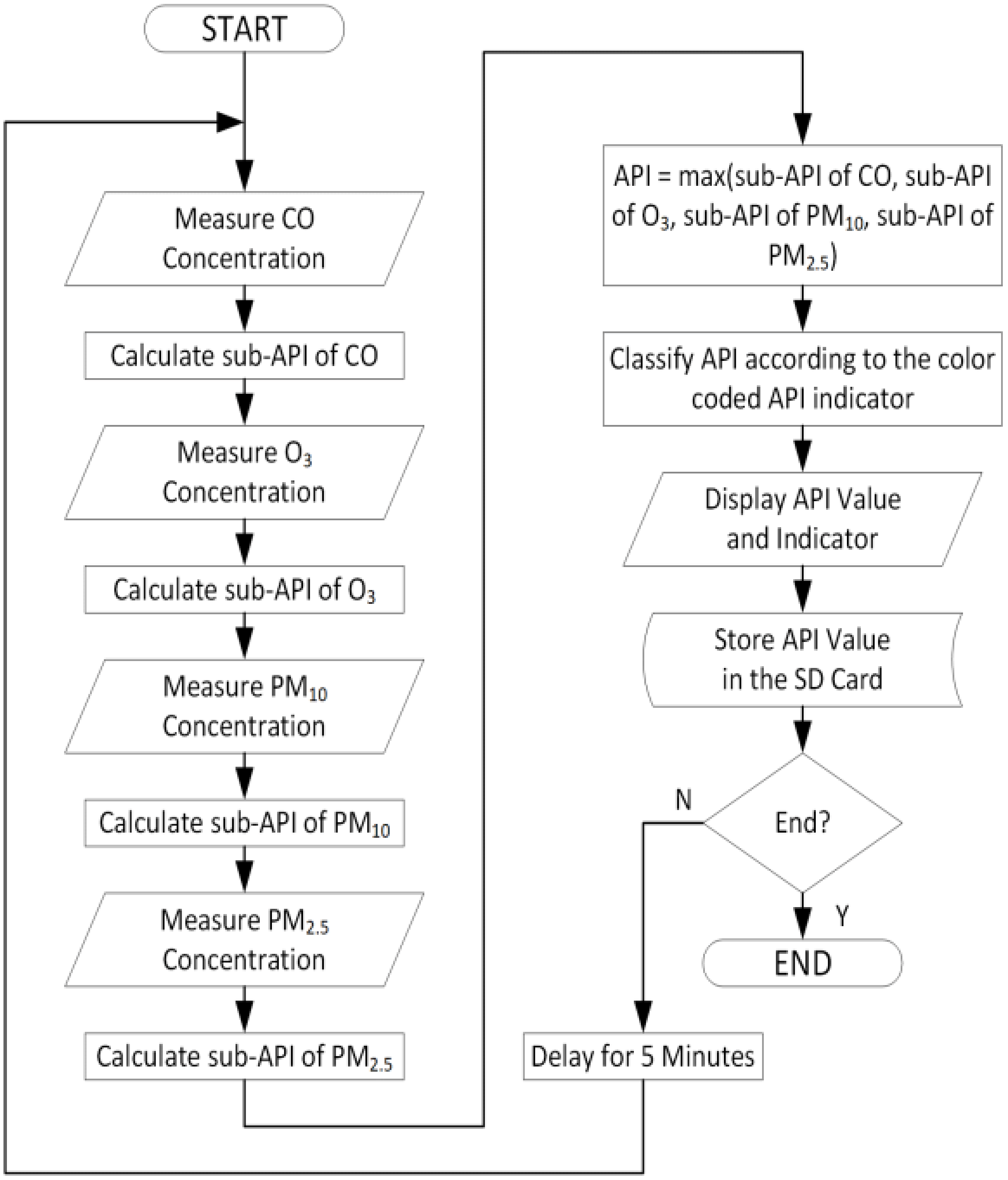
   - Air quality trend reports and data export (e.g., CSV).

8. Hosting and Deployment:

   - Cloud-hosted app with public URL access.

SOLUTION:

1.FLOW CHART


START
Measure CO Concentration
Calculate sub-API of CO
Measure O_3 Concentration
Calculate sub-API of O_3
Measure PM_{10} Concentration
Calculate sub-API of PM_{10}
Measure $PM_{2.5}$ Concentration
Calculate sub-API of $PM_{2.5}$
API = max(sub-API of CO, sub-API of O_3, sub-API of PM_{10}, sub-API of $PM_{2.5}$)
Classify API according to the color coded API indicator
Display API Value and Indicator
Store API Value in the SD Card
End?
N
Y
END
Delay for 5 Minutes

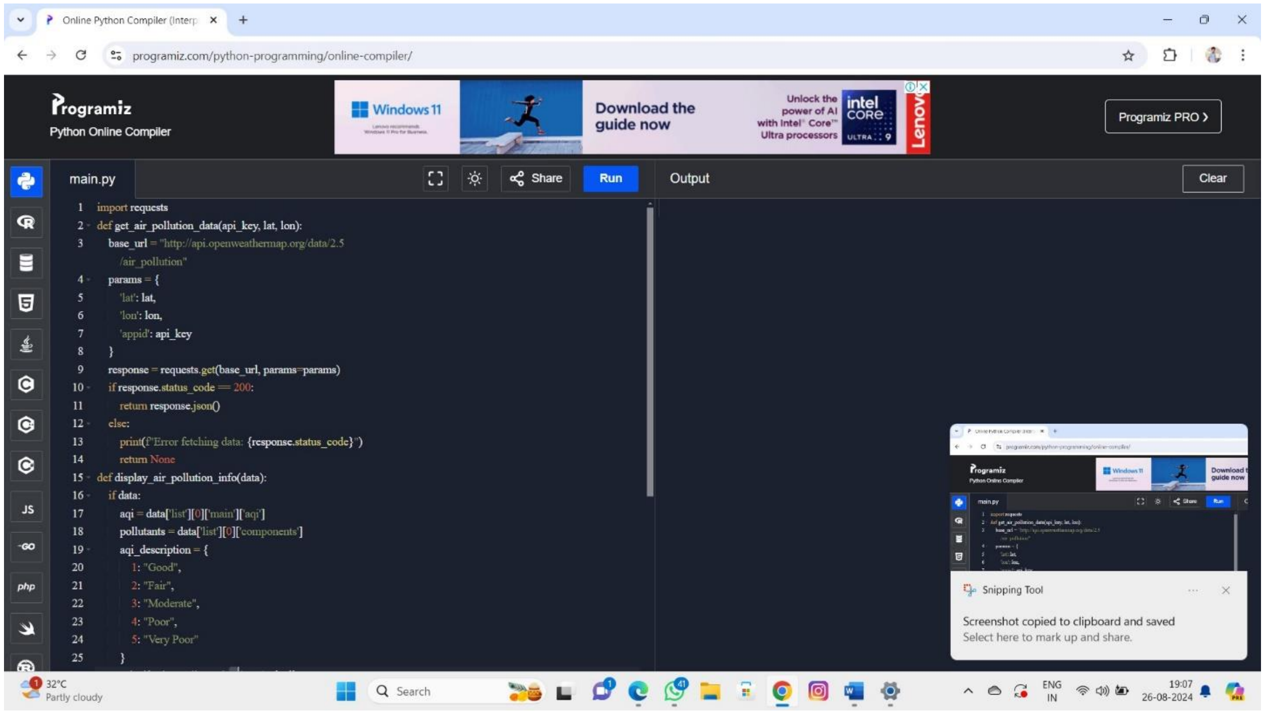
2.CODE IMPLEMENTATION:

```
 import requests


def get_air_pollution_data(api_key, lat, lon):
    base_url = "http://api.openweathermap.org/data/2.5/air_pollution"
    params = {
        'lat': lat,
        'lon': lon,
        'appid': api_key
    }

    response = requests.get(base_url, params=params)

    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching data: {response.status_code}")
        return None

def display_air_pollution_info(data):
    if data:
        aqi = data['list'][0]['main']['aqi']
        pollutants = data['list'][0]['components']
        aqi_description = {
            1: "Good",
            2: "Fair",
            3: "Moderate",
            4: "Poor",
            5: "Very Poor"
        }

        print(f"Air Quality Index (AQI): {aqi} - {aqi_description[aqi]}")
        print("Pollutant Levels (in µg/m³):")
        print(f"PM2.5: {pollutants['pm2_5']}")
        print(f"PM10: {pollutants['pm10']}")
        print(f"Carbon Monoxide (CO): {pollutants['co']}")
        print(f"Ozone (O3): {pollutants['o3']}")
        print(f"Nitrogen Dioxide (NO2): {pollutants['no2']}")
        print(f"Sulfur Dioxide (SO2): {pollutants['so2']}")
        print(f"Ammonia (NH3): {pollutants['nh3']}")
    else:
        print("No air pollution data available.")

def main():
    api_key = "your_openweathermap_api_key"  # Replace with your actual API key
    lat = input("Enter latitude: ").strip()
    lon = input("Enter longitude: ").strip()


    air_pollution_data = get_air_pollution_data(api_key, lat, lon)
    display_air_pollution_info(air_pollution_data)


if __name__ == "__main__":
    main()
```

3. DISPLAY THE CURRENT INFORMATION:

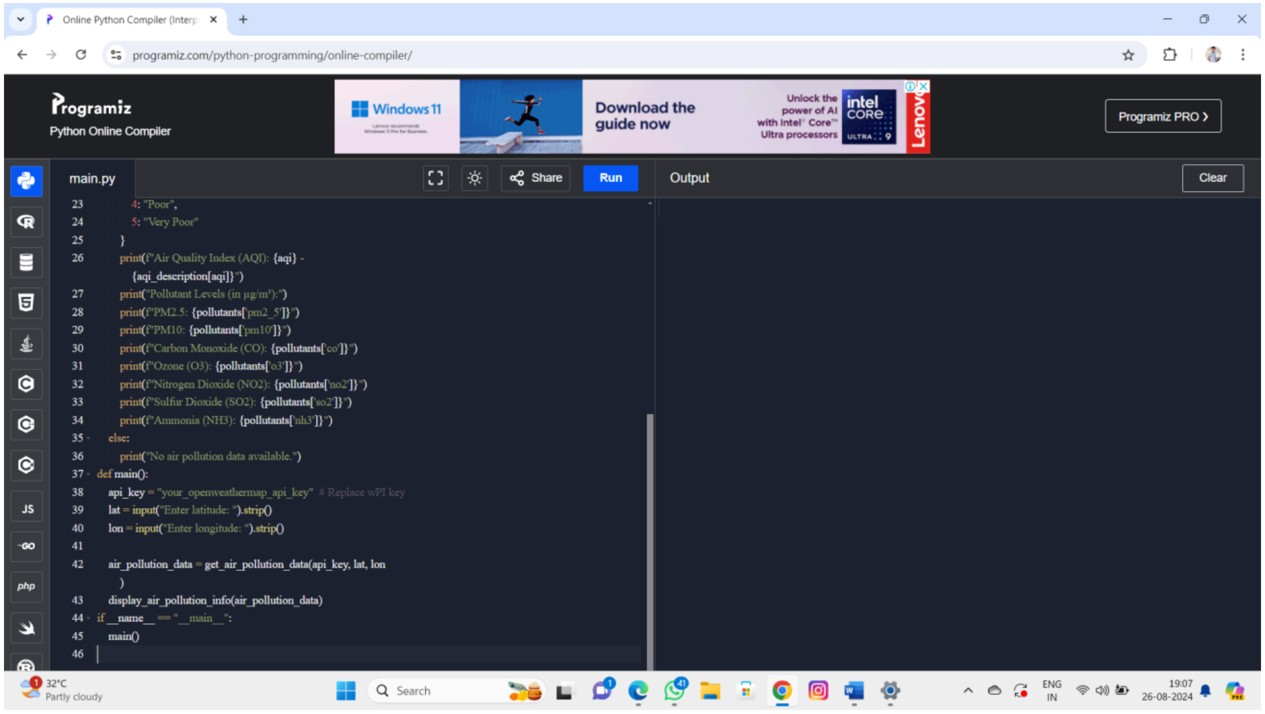
```
CO Concentration: 62.88
O₃ Concentration: 22.09PM₁₀
Concentration: 56.65PM₂.₅
Concentration: 89.80API
Value: 44.90
API Classification: Good
```

4.USER INPUT:



```python
import requests
def get_air_pollution_data(api_key, lat, lon):
    base_url = "http://api.openweathermap.org/data/2.5
      /air_pollution"
    params = {
        'lat': lat,
        'lon': lon,
        'appid': api_key
    }
    response = requests.get(base_url, params=params)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching data: {response.status_code}")
        return None
def display_air_pollution_info(data):
    if data:
        aqi = data['list'][0]['main']['aqi']
        pollutants = data['list'][0]['components']
        aqi_description = {
            1: "Good",
            2: "Fair",
            3: "Moderate",
            4: "Poor",
            5: "Very Poor"
        }
```

Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.



```python
            4: "Poor",
            5: "Very Poor"
        }
        print(f"Air Quality Index (AQI): {aqi} -
          {aqi_description[aqi]}")
        print("Pollutant Levels (in µg/m³):")
        print(f"PM2.5: {pollutants['pm2_5']}")
        print(f"PM10: {pollutants['pm10']}")
        print(f"Carbon Monoxide (CO): {pollutants['co']}")
        print(f"Ozone (O3): {pollutants['o3']}")
        print(f"Nitrogen Dioxide (NO2): {pollutants['no2']}")
        print(f"Sulfur Dioxide (SO2): {pollutants['so2']}")
        print(f"Ammonia (NH3): {pollutants['nh3']}")
    else:
        print("No air pollution data available.")
def main():
    api_key = "your_openweathermap_api_key"  # Replace wPI key
    lat = input("Enter latitude: ").strip()
    lon = input("Enter longitude: ").strip()

    air_pollution_data = get_air_pollution_data(api_key, lat, lon
      )
    display_air_pollution_info(air_pollution_data)
if __name__ == "__main__":
    main()
```

5,DOCUMENTATION:

ALGORITHM:

To fetch and display air pollution data, first, prompt the user to input the latitude and longitude of their location. Then, send a GET request to the Air Pollution API using these coordinates and an API key. Check if the response is successful (status code 200); if not, display an error message. If successful, extract the Air Quality Index (AQI) and levels of pollutants such as PM2.5, PM10, CO, NO2, SO2, and O3 from the response. Finally, present the AQI value along with its category (e.g., Good, Moderate) and the pollutant levels to the user.

HOW HISTORICAL DATA INFLUENCES DECISION ON AIR POLLUTION:

Historical air pollution data is crucial for informed decision-making regarding air quality management and public health. By analyzing past trends, decision-makers can identify long-term patterns in pollution levels, helping to detect increases or decreases over time and assess their impact. This data is essential for evaluating the health effects of air pollution, which informs health advisories and preventive measures for at-risk groups. It also guides the development and adjustment of air quality policies, such as stricter emission controls or public awareness campaigns. Additionally, historical data aids in emergency response planning by highlighting periods of severe pollution, helping to prepare for future events. Public awareness is enhanced through access to historical data, encouraging community action and support for better air quality policies. Lastly, historical data supports informed urban planning and infrastructure development by considering potential air quality impacts and integrating mitigation strategies.

:

CONCLUSION:

In conclusion, integrating an Air Pollution API into applications offers valuable real-time insights into air quality, empowering users to make informed decisions about their health and activities. By providing current data on pollutants and AQI values, such APIs enable users to respond proactively to changing air conditions and take necessary precautions. Additionally, such tools support public awareness, facilitate effective health advisories, and contribute to better environmental management. Overall, the Air Pollution API enhances the ability to monitor and improve air quality, fostering a healthier and more informed community.