# DEPARTMENT OF ELECTRONICS AND COMMUNICATION

**Project name :** Air Quality Monitoring

**Team name : MNSN_TEAM**

**Team members :**

Mallikarjuna.M(111421106029)

Navya.M(111421106030)

Sai Teja.N(111421106031)

Nithish.G(111421106032)

# Problem Statement

Air quality is a critical concern in many urban areas, impacting the health and well-being of residents. Monitoring air quality is crucial for identifying pollution sources and implementing effective measures to improve it. This project aims to develop an IoT-based system for real-time air quality monitoring. The system will collect data from various sensors and provide a user-friendly platform for users to access and visualize this data.

# Objectives:

1.**Real-time Air Quality Monitoring:** Develop an IoT system that can continuously measure key air quality parameters, including particulate matter (PM2.5 and PM10), carbon monoxide (CO), nitrogen dioxide (NO2), and ozone (O3).
2.**Data Collection and Transmission:** Set up IoT devices equipped with appropriate sReal-time Air Quality Monitoring: Develop an IoT system that can continuously measure key air quality parameters, including particulate matter (PM2.5 and PM10), carbon monoxide (CO), nitrogen dioxide (NO2), and ozone (O3).

3.**Data Collection and Transmission:** Set up IoT devices equipped with appropriate sensors to collect air quality data and transmit it to a central platform.

4.**Data Visualization:** Create a user-friendly data-sharing platform where users can access real-time air quality information and historical data through an intuitive user interface.

5.**Alerting and Reporting:** Implement an alerting system that notifies users when air quality parameters exceed predefined thresholds. Additionally, generate reports for historical data analysis.ensors to collect air quality data and transmit it to a central platform.
6.**Data Visualization:** Create a user-friendly data-sharing platform where users can access real-time air quality information and historical data through an intuitive user interface.
7.**Alerting and Reporting:** Implement an alerting system that notifies users when air quality parameters exceed predefined thresholds. Additionally, generate reports for historical data analysis.

# IoT Device Setup:

- For the IoT device setup, we will use the following components and sensors:

- Microcontroller: Raspberry Pi or Arduino for data processing and communication.
- Air Quality Sensors: Sensors capable of measuring PM2.5, PM10, CO, NO2, and O3 levels.
- Data Transmission: Wi-Fi or GSM module to send data to the central platform.
- Power Supply: Battery or an external power source depending on the deployment location

# Platform Development:

- Database Setup: Create a database to store the incoming air quality data.

- Web Application: Develop a web-based application that allows users to access and visualize the data. The platform should have user registration, login, and data representation features.

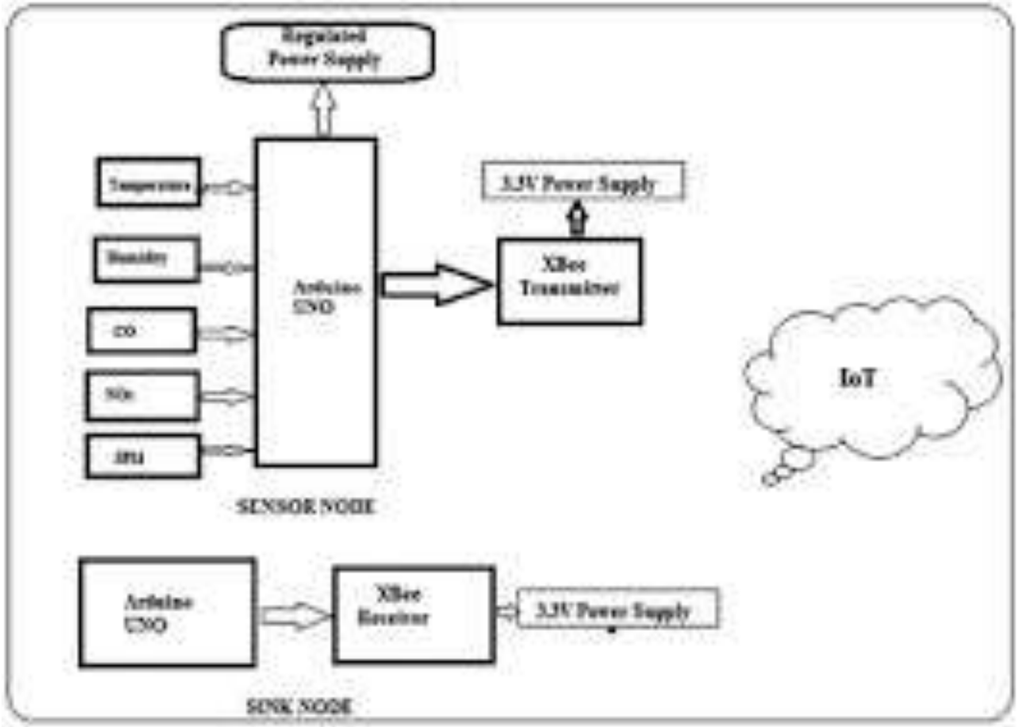
- Threshold Configuration: Implement a feature that enables users to set alert thresholds for each air quality parameter.

- Real-time Updates: Ensure that the platform receives and updates data in real-time.

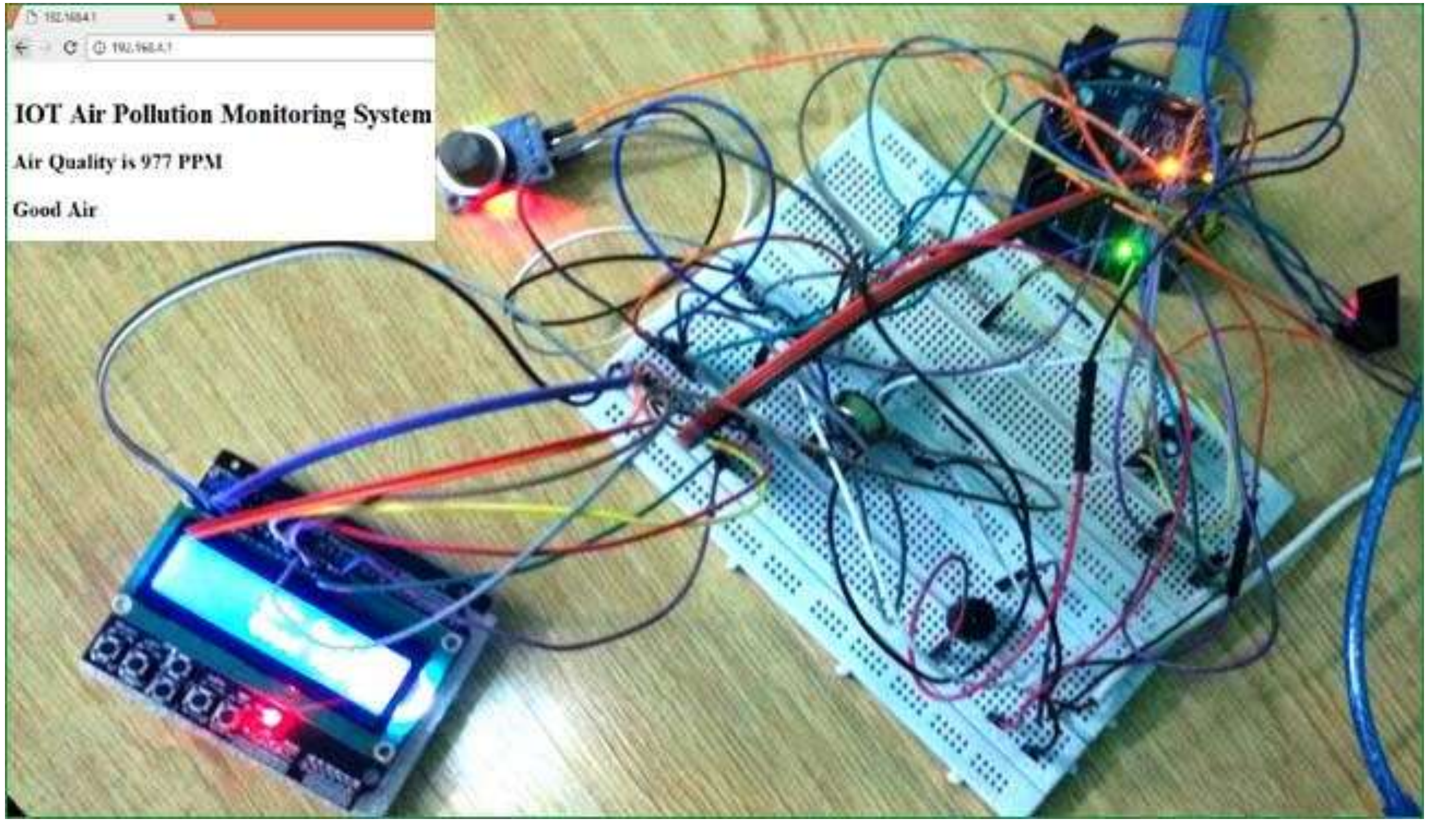
# Code

```
import random

# Function to generate random sensor data
def generate_sensor_data():
    pm25 = random.uniform(0, 100)
    co = random.uniform(0, 5)
    no2 = random.uniform(0, 1)
    o3 = random.uniform(0, 0.1)
    return pm25, co, no2, o3

# Function to calculate the Air Quality Index (AQI)
def calculate_aqi(pm25, co, no2, o3):
    # Replace with your AQI calculation logic
    # This is a simplified example; use an appropriate formula for AQI calculation
    aqi = (pm25 + co + no2 + o3) / 4
    return aqi

# Function to classify air quality based on AQI
def classify_air_quality(aqi):
    if aqi <= 50:
        return "Good"
    elif aqi <= 100:
        return "Moderate"
    elif aqi <= 150:
        return "Unhealthy for Sensitive Groups"
    elif aqi <= 200:
        return "Unhealthy"
    else:
        return "Very Unhealthy"

# Generate sensor data
pm25, co, no2, o3 = generate_sensor_data()

# Calculate AQI
aqi = calculate_aqi(pm25, co, no2, o3)

# Classify air quality
air_quality = classify_air_quality(aqi)

# Output the results
print(f"PM2.5: {pm25} µg/m³")
print(f"CO: {co} ppm")
print(f"NO2: {no2} ppm")
print(f"O3: {o3} ppm")
print(f"AQI: {aqi}")
print(f"Air Quality: {air_quality}")
```

# BLOCK DIAGRAM



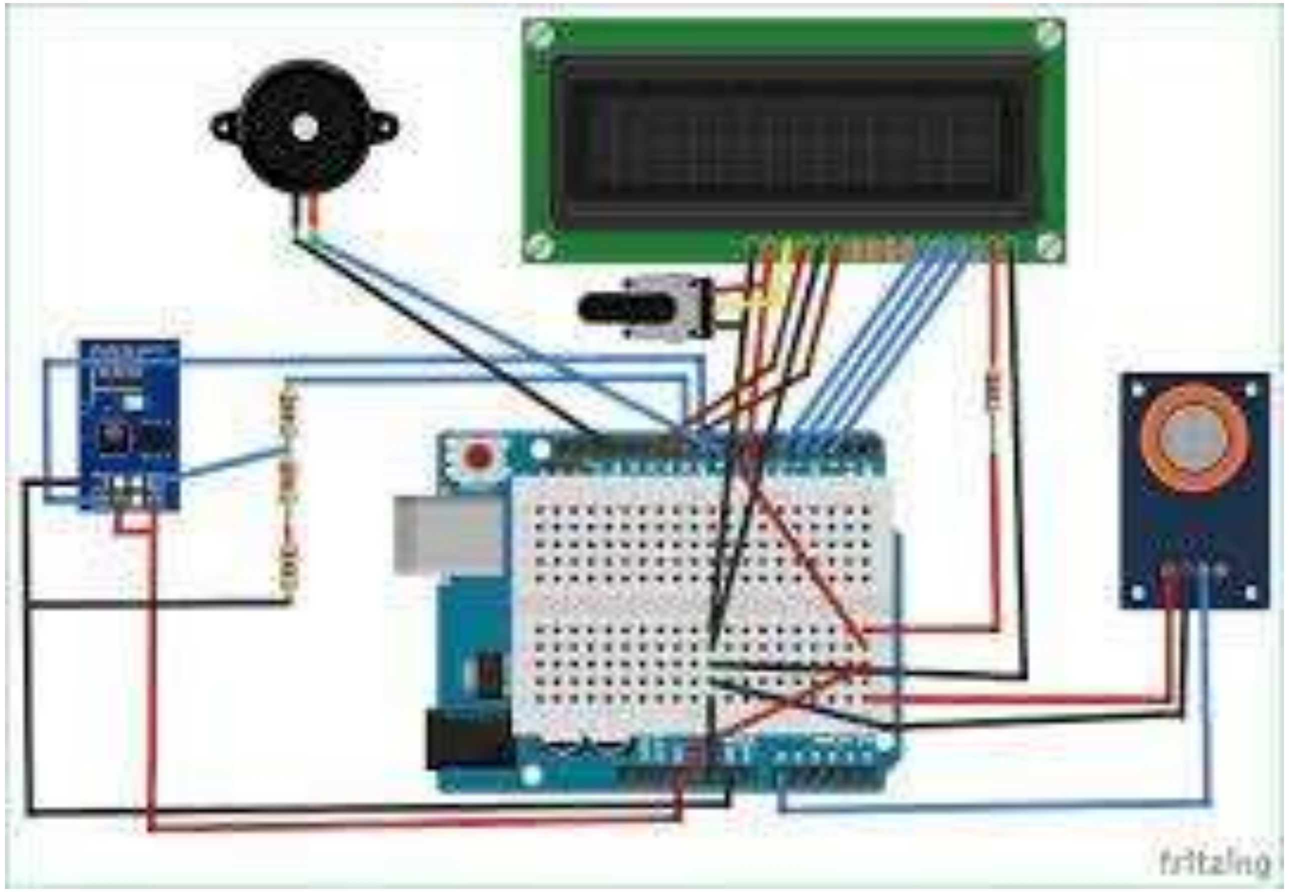
# screenshots of the IoT devices



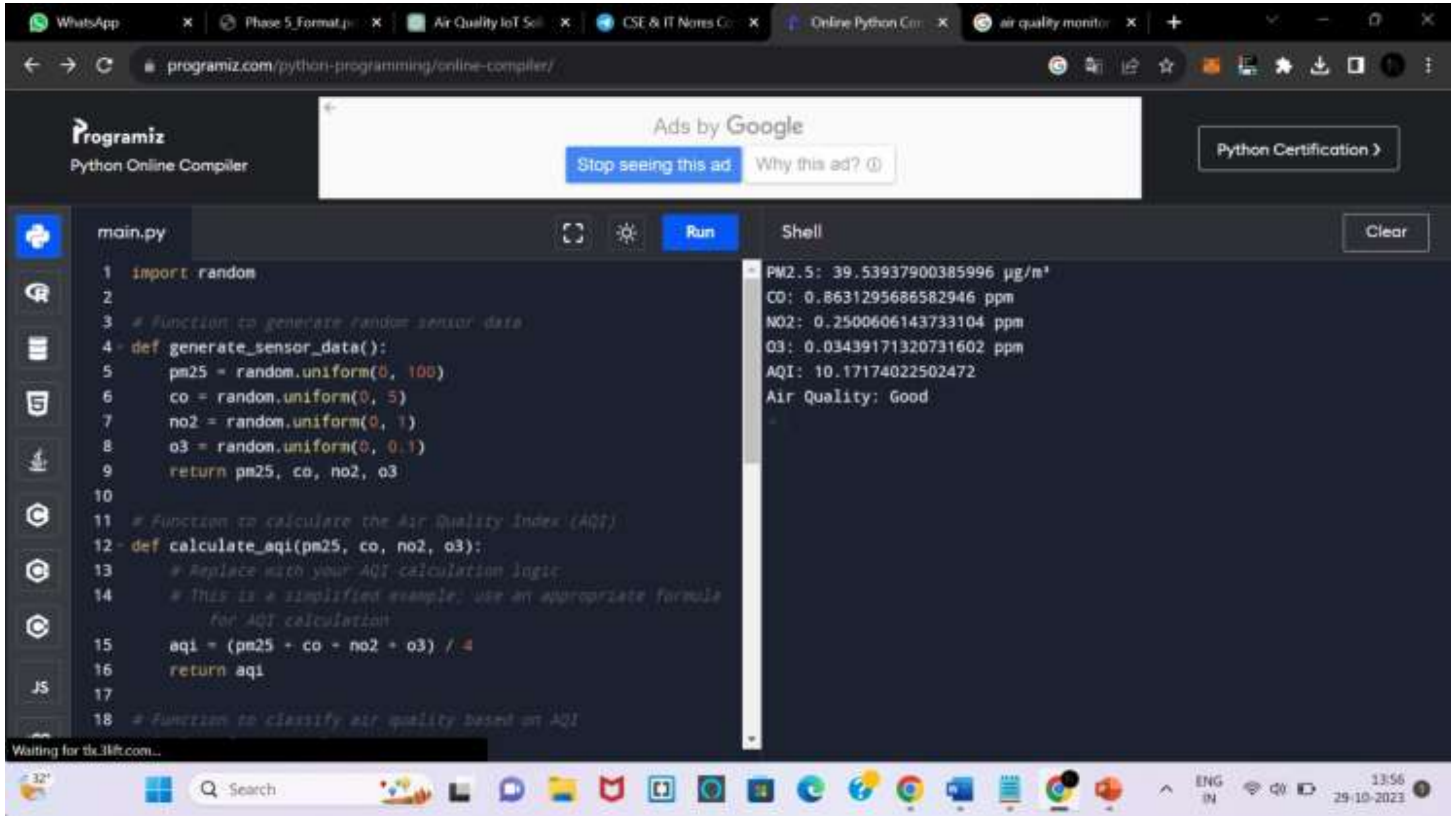
IOT Air Pollution Monitoring System

Air Quality is 977 PPM

Good Air

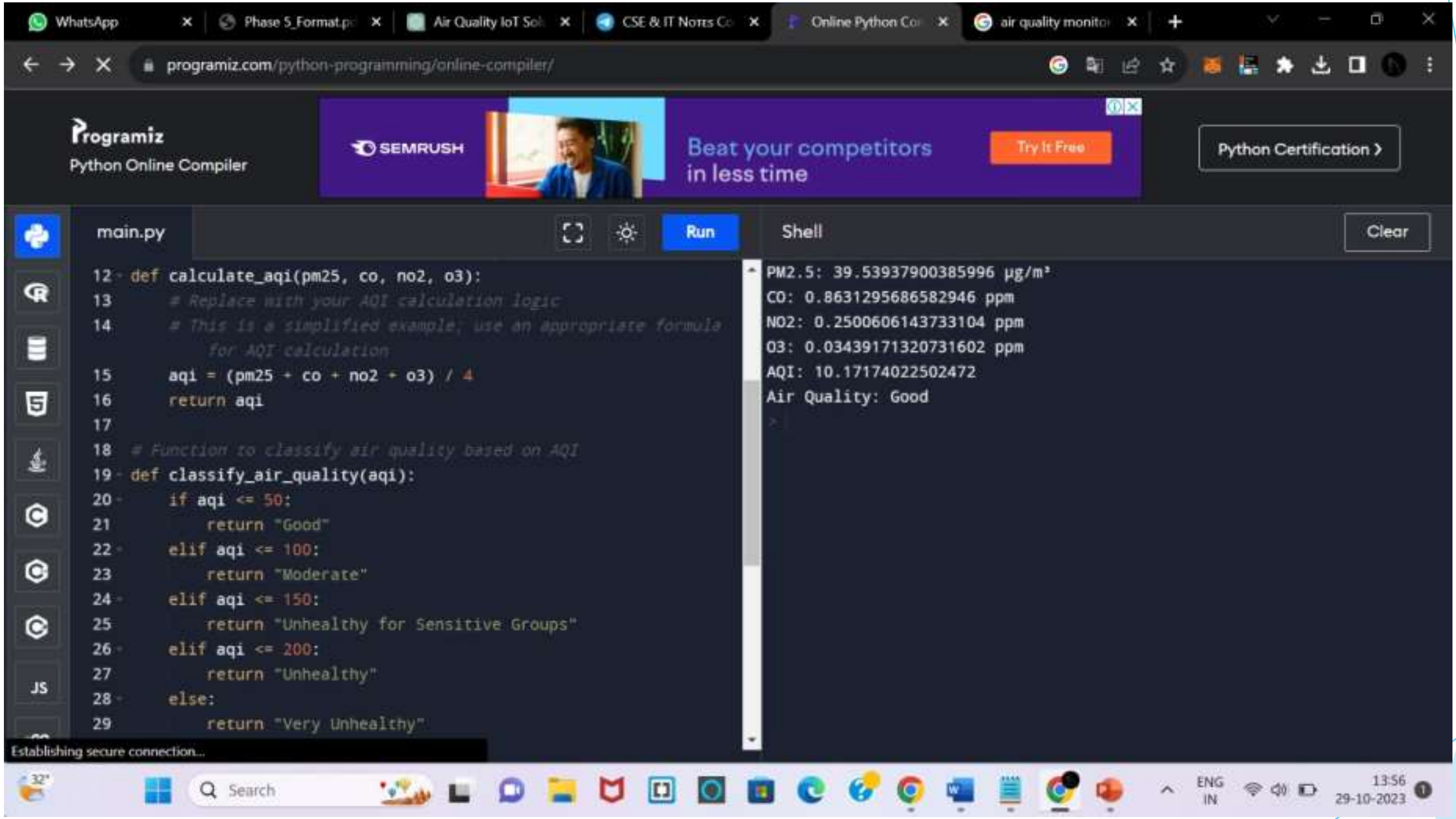

fritzing

# SCREENSHOT OF PYTHON CODE OUTPUT:



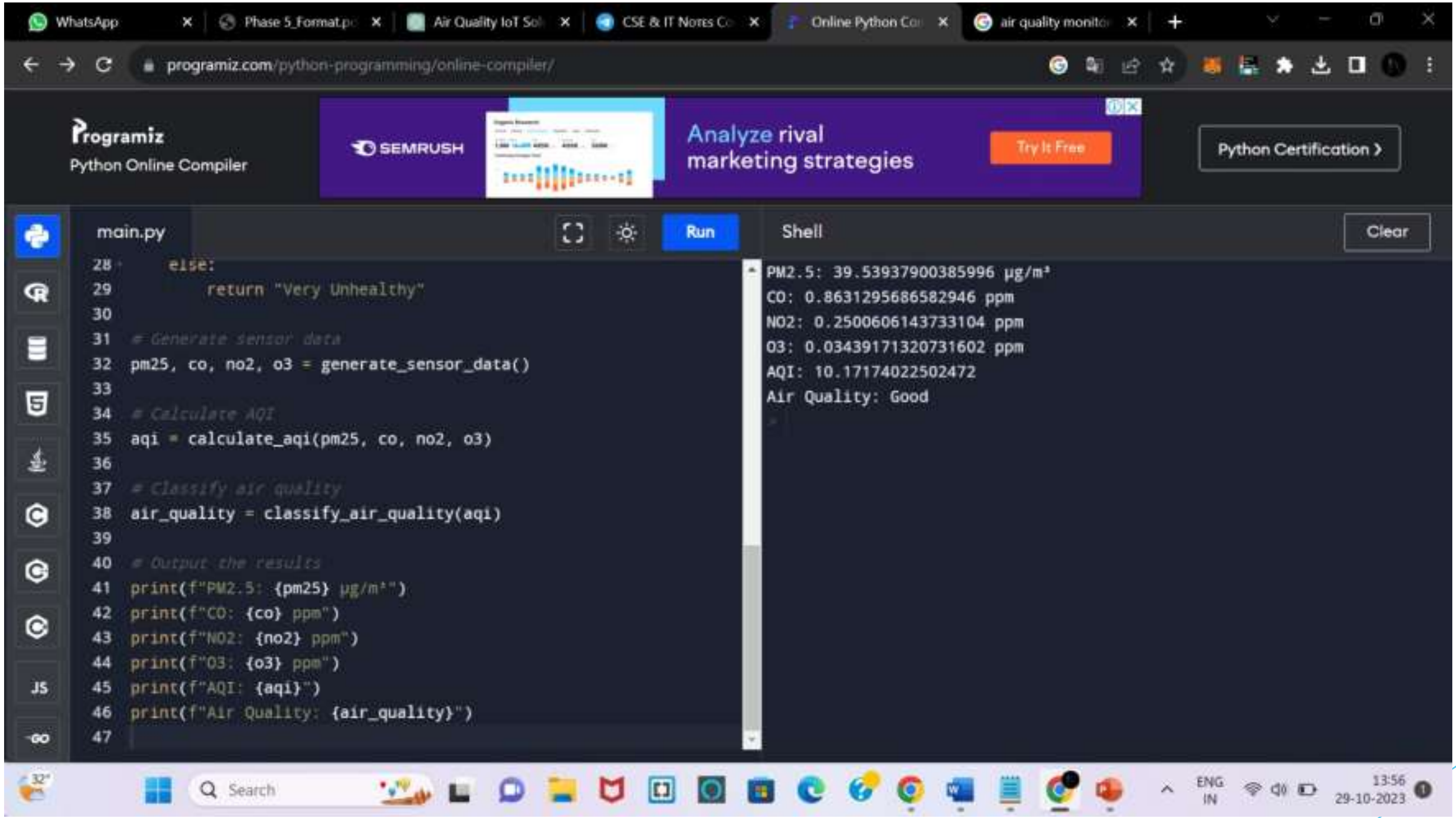
```python
import random

# Function to generate random sensor data
def generate_sensor_data():
    pm25 = random.uniform(0, 100)
    co = random.uniform(0, 5)
    no2 = random.uniform(0, 1)
    o3 = random.uniform(0, 0.1)
    return pm25, co, no2, o3

# Function to calculate the Air Quality Index (AQI)
def calculate_aqi(pm25, co, no2, o3):
    # Replace with your AQI calculation logic
    # This is a simplified example; use an appropriate formula
        for AQI calculation
    aqi = (pm25 + co + no2 + o3) / 4
    return aqi

# Function to classify air quality based on AQI
```

```
PM2.5: 39.53937900385996 µg/m³
CO: 0.8631295686582946 ppm
NO2: 0.2500606143733104 ppm
O3: 0.03439171320731602 ppm
AQI: 10.17174022502472
Air Quality: Good
```



```python
def calculate_aqi(pm25, co, no2, o3):
    # Replace with your AQI calculation logic
    # This is a simplified example; use an appropriate formula for AQI calculation
    aqi = (pm25 + co + no2 + o3) / 4
    return aqi

# Function to classify air quality based on AQI
def classify_air_quality(aqi):
    if aqi <= 50:
        return "Good"
    elif aqi <= 100:
        return "Moderate"
    elif aqi <= 150:
        return "Unhealthy for Sensitive Groups"
    elif aqi <= 200:
        return "Unhealthy"
    else:
        return "Very Unhealthy"
```

```
PM2.5: 39.53937900385996 µg/m³
CO: 0.8631295686582946 ppm
NO2: 0.2500606143733104 ppm
O3: 0.03439171320731602 ppm
AQI: 10.17174022502472
Air Quality: Good
>
```



```python
    else:
        return "Very Unhealthy"

# Generate sensor data
pm25, co, no2, o3 = generate_sensor_data()

# Calculate AQI
aqi = calculate_aqi(pm25, co, no2, o3)

# Classify air quality
air_quality = classify_air_quality(aqi)

# Output the results
print(f"PM2.5: {pm25} µg/m³")
print(f"CO: {co} ppm")
print(f"NO2: {no2} ppm")
print(f"O3: {o3} ppm")
print(f"AQI: {aqi}")
print(f"Air Quality: {air_quality}")

```

```
PM2.5: 39.53937900385996 µg/m³
CO: 0.8631295686582946 ppm
NO2: 0.2500606143733104 ppm
O3: 0.03439171320731602 ppm
AQI: 10.17174022502472
Air Quality: Good
```

# IoT Device Code:

- Initialize and configure sensors.
- Set up data transmission protocols (e.g., MQTT or HTTP).
- Implement error handling and retry mechanisms for data transmission.
- Continuously collect and send data to the central platform.

# Data Sharing Platform UI:

- The platform's user interface should include the following components:

- User Registration and Login: Secure user authentication for data access.
- Dashboard: A user-friendly dashboard displaying real-time air quality data.
- Historical Data Visualization: Charts and graphs for historical data analysis.
- Alert Configuration: A settings page to set and manage alert thresholds.
- Notifications: Real-time alerts and notifications when air quality parameters exceed thresholds.

# EXPLANATION

**Air quality monitoring is essential for safeguarding public health and the environment, especially in urban areas facing air pollution challenges. This project presents an IoT-based Air Quality Monitoring System designed to continuously collect data from IoT devices equipped with sensors measuring key air quality parameters, including PM2.5, CO, NO2, and O3. The project's primary objectives are to send this data to a central platform, calculate an Air Quality Index (AQI), and classify air quality conditions. Users can access this information via a user-friendly platform, set alert thresholds, and receive notifications when air quality deteriorates. This system contributes to real-time environmental awareness and empowers individuals and communities to make informed decisions regarding their well-being and environmental impact.**

# CONCLUSION

The development of an IoT air quality monitoring system is crucial for public health and environmental monitoring. By integrating air quality sensors with web technologies, this project aims to provide a user-friendly platform for individuals and authorities to access real-time air quality data. The platform will enhance awareness of air quality issues and assist in making informed decisions for environmental management. It is expected that the successful completion of this project will contribute to improved air quality monitoring and management in various locations.

# THANK YOU