**Phase 3 project**

**Project Title: AIR QUALITY MONITORING**

**Project ID:** proj\_223737\_Team\_3

**College:** Gnanamani College of Technology

**College code:** 6208

**Branch:** B.Tech-Information Technology

**Year:** IIIrd year

Team Members:

Balachandar.K(620821205006)

Jana.V(620821205020)

Kathirvanan.S(620821205028)

Pravinkumar.K(620821205041)

Srikanth.S(620821205056)

**AIR QUALITY MONITORING**

**Definition:**

An IoT-based air pollution monitoring system is an ideal solution that can provide real-time data and insights about the air quality in a particular area. An IoT based air pollution monitoring system consists of several hardware and software components that works together to collect and process data.

**PHASE 3**

To configure IoT devices like sensors for measuring air quality parameters and develop a Python script to send the collected data to a data-sharing platform, you'll need some hardware components (e.g., sensors, microcontroller), software libraries, and a data-sharing platform. In this example, I'll use the Raspberry Pi as an IoT device and a fictional data-sharing platform. You can adapt this example to your specific hardware and platform.

**Hardware Components:**

1. Raspberry Pi (or any suitable microcontroller)
2. Air quality sensors (e.g., PM2.5/PM10 sensors, gas sensors)
3. Wi-Fi/Internet connection

**Software Dependencies:**

1. Python (pre-installed on the Raspberry Pi)
2. Appropriate Python libraries for your sensors (e.g., Adafruit CircuitPython libraries, if using Adafruit sensors)
3. MQTT library for data communication (e.g., Paho MQTT)
4. Any additional libraries specific to your data-sharing platform

**Steps to Configure IoT Device and Develop Python Script:**

1. Set up your Raspberry Pi with the necessary sensors. Make sure the sensors are correctly connected to the GPIO pins on the Raspberry Pi.
2. Install the required Python libraries on your Raspberry Pi using pip. For example, if you are using the Adafruit BME280 sensor, you can install the library like this:

pip install adafruit-circuitpython-bme280

1. Create a Python script to read data from the sensors and send it to your data-sharing platform. Here's a basic example using MQTT for data communication:

import time

import board

import adafruit\_bme280

import paho.mqtt.client as mqtt

# Initialize the sensor

i2c = board.I2C()

sensor = adafruit\_bme280.Adafruit\_BME280\_I2C(i2c)

# MQTT configurations

mqtt\_broker = "your.mqtt.broker.com"

mqtt\_port = 1883

mqtt\_topic = "air\_quality\_data"

mqtt\_client\_id = "air\_quality\_sensor"

# Connect to the MQTT broker

client = mqtt.Client(client\_id=mqtt\_client\_id)

client.connect(mqtt\_broker, mqtt\_port)

while True:

try:

# Read sensor data

temperature = sensor.temperature

humidity = sensor.humidity

pressure = sensor.pressure

# Send data via MQTT

data = {

"temperature": temperature,

"humidity": humidity,

"pressure": pressure

}

client.publish(mqtt\_topic, json.dumps(data))

time.sleep(60) # Send data every 60 seconds

except Exception as e:

print("Error: ", str(e))

time.sleep(60) # Retry after 60 seconds in case of an error

1. Replace the placeholders in the script with your actual MQTT broker details and customize the data format to match your platform's requirements.
2. Run the Python script on your Raspberry Pi. This script will continuously read data from the sensor and publish it to the specified MQTT topic.
3. Set up a data-sharing platform that subscribes to the MQTT topic to receive and process the sensor data.