### PROJECT TITLE: NOISE POLLUTION MONITORING

### IOT PHASE 3: Development Part 1

### 

### DESCRIPTION

Building an IoT-enabled Noise Pollution Monitoring system involves setting up the IoT noise sensors to measure noise levels and sending this data to a central platform for processing and analysis. In this phase, we'll focus on developing a Python script to run on the IoT sensors to collect noise level data and send it to a central server using MQTT protocol.

**Hardware Setup**

**Noise Sensors (e.g., LM393):**

Deploy noise sensors (e.g., LM393 sound sensors) in public areas where you want to monitor noise levels. Connect these sensors to a microcontroller (e.g., Raspberry Pi, Arduino) that will read the noise levels.

**Microcontroller (ARDUINO UNO):**

Use a microcontroller to interface with the noise sensors, read the noise levels, and communicate with the central server. Connect the noise sensor(s) to appropriate pins on the microcontroller.

**Software Development**

**Python Script on IoT Sensors**:

Develop a Python script that runs on the IoT sensors (microcontroller) to read noise levels from the sensors.

It sends this data to a central server using a communication protocol like MQTT.

**Python script**

**PROGRAM**

import time

import RPi.GPIO as GPIO

import paho.mqtt.client as mqtt

SOUND\_SENSOR\_PIN = 18 # GPIO pin where the sound sensor is connected

BROKER = 'mqtt.example.com' # MQTT broker address

PORT = 1883 # MQTT broker port

TOPIC = 'noise\_level'

def setup\_gpio():

GPIO.setmode(GPIO.BCM)

GPIO.setup(SOUND\_SENSOR\_PIN, GPIO.IN)

def read\_noise\_level():

while True:

noise\_level = GPIO.input(SOUND\_SENSOR\_PIN)

client.publish(TOPIC, str(noise\_level))

print(f'Sent noise level: {noise\_level}')

time.sleep(5) # Publish data every 5 seconds

def on\_connect(client, userdata, flags, rc):

print(f'Connected to MQTT broker with result code {rc}')

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

client = mqtt.Client()

client.on\_connect = on\_connect

client.connect(BROKER, PORT, 60)

setup\_gpio()

client.loop\_start()

read\_noise\_level()

**Central Server / Platform**

**Noise Pollution Information Platform**:

Set up a central server or cloud platform to receive the noise level data from the IoT sensors. This platform will process, analyze, and store the data for monitoring and visualization.

**MQTT Broker:**

Set up an MQTT broker (e.g., Mosquitto) on the central server to receive data from the IoT sensors.

**Data Processing and Analysis:**

Implement data processing and analysis logic on the platform to process the received noise level data, detect patterns, and derive insights.

**Deployment and Monitoring**

**Deployment:**

Deploy the IoT noise sensors in the desired public areas.

**Monitoring and Visualization:**

Monitor the noise level data collected from the IoT sensors on the platform. Create visualizations and alerts to effectively monitor noise pollution in real-time.

\*By following these steps, you can begin building your IoT-enabled Noise Pollution Monitoring system and start collecting and monitoring noise level data from public areas.