PROJECT SUBMITTED BY:

NAME: POLISETTY OMKAR

REGISTER NO: 713921106038

TOPIC: NOISE POLLUTION MONITERING

MAIL ID: polisettyomkar@gmail.com

NM ID:au713921106038

COLLEGE CODE: 7139

**DEVELOPMENT OF NOISE POLLUTON MONITERING DEVICE USING IOT**

**Requirements:**

* Determine the specific goals of your noise pollution monitoring system (e.g., measuring sound levels in a specific area, identifying noise sources).
* Choose appropriate sound sensors (microphones or sound level sensors) based on your requirements.
* Decide on the communication protocol (Wi-Fi, LoRa, GSM) and the platform for data storage and analysis.

**Hardware:**

* **Sensors:** Choose sound level sensors capable of capturing the range of noise levels you want to monitor. Calibrate the sensors for accurate readings.
* **Microcontroller:** Use a microcontroller board like Arduino or Raspberry Pi to interface with the sensors and transmit data.
* **Communication Module:** Select a communication module compatible with your microcontroller and the desired range of data transmission.
* **Power Supply:** Design a power supply system (battery or mains power) based on the deployment location.

PYTHON SCRIPT

import spidev

import time

import requests

SPI\_PORT = 0

SPI\_DEVICE = 0

API\_ENDPOINT = "https://example.com/api/noise-level"

def read\_noise\_level(channel):

spi = spidev.SpiDev()

spi.open(SPI\_PORT, SPI\_DEVICE)

adc\_data = spi.xfer2([1, (8 + channel) << 4, 0])

noise\_level = ((adc\_data[1] & 3) << 8) + adc\_data[2]

spi.close()

return noise\_level

# Main function

def main():

try:

while True:

# Read noise level from MCP3008 ADC (channel 0 in this case)

noise\_level = read\_noise\_level(channel=0)

print("Noise Level: {}".format(noise\_level))

payload = {

"noise\_level": noise\_level

}

response = requests.post(API\_ENDPOINT, json=payload)

if response.status\_code == 200:

print("Data sent successfully!")

else:

print("Failed to send data!

time.sleep(60)

except KeyboardInterrupt:

print("Monitoring stopped by the user.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Conclusion and Future Improvements:**

* Benefits: IoT-based noise pollution monitoring systems offer valuable insights for urban planning, environmental research, and public health, aiding in noise pollution mitigation strategies.
* Continuous Improvement: Regular calibration, software updates, and feedback-driven enhancements are essential to maintain the accuracy and reliability of the monitoring device.
* Community Engagement: Involving communities and local authorities can lead to collaborative efforts in addressing noise pollution issues and implementing effective solutions.