PHASE 3 Development part1

NOISE POLLUTION MONITORING

Introduction:

Noise may be measured using a sound level meter at the source of the noise. Alternatively, an organization or company may measure a person's exposure to environmental noise in a workplace via a noise dosimeter. The measurements taken using either of these methods will be evaluated according to the standards below.

Deploy noise sensors in public areas to measure noise levels.

These sensors can be strategically placed in different locations to capture real-time noise data. By leveraging technology, the sensors can transmit the collected data wirelessly to a central system for analysis and storage. The deployment of noise sensors offers several benefits. Firstly, it enables continuous monitoring of noise levels providing a comprehensive understanding of noise pollution patterns throughout the day and across different areas. This data can be used to identify noise hotspots, assess the effectiveness of noise control measures, and make informed decisions regarding urban planning and noise mitigation strategies. Additionally, noise sensors can facilitate data-driven decision-making by providing accurate and reliable information. The collected data can be analysed to identify trends, patterns, and potential sources of noise pollution. This information can be valuable for policymakers, urban planners, and environmental agencies in developing targeted interventions to reduce noise pollution and improve the quality of life for residents. It is important to ensure that the deployment of noise sensors complies with privacy regulations and guidelines. Proper data anonymization and security measures should be implemented to protect the privacy of individuals while still allowing for effective noise monitoring and analysis. Overall, deploying noise sensors in public areas is a proactive and data-driven approach to address noise pollution, enabling evidence-based decision-making and fostering healthier and more sustainable urban environments.

Develop python script

Creating a Python script for IoT sensors to send real-time noise level data to a noise pollution information platform involves several steps. Here's a basic outline of how you can achieve this using a Raspberry Pi as an example hardware platform. Keep in mind that the exact implementation may vary depending on your specific hardware and the platform you're sending the data to.

Hardware setup

You'll need an IoT device with a noise level sensor (e.g., a microphone) and network connectivity (e.g., Wi-Fi or Ethernet). In this example, we'll use Connect the noise level sensor to your device as per the manufacturer's instructions.

Install Necessary Libraries: Install any necessary Python libraries for working with your sensor. For a noise sensor, you might need a library for reading data from the sensor, such as sound device.

Set Up Network Connection:[4]

Ensure your IoT device is connected to the internet. Configure Wi-Fi or Ethernet settings as needed. Create a Noise Sensor Script:Write a Python script to read data from the noise sensor. This script should use the appropriate library to collect noise

PYTHON SCRIPT:

```
      import time

import requests

import sound_sensor_library  # Replace with the actual library for your noise sensor

# Configuration

SENSOR_ID = "sensor_001"  # Unique identifier for the sensor

API_ENDPOINT = "https://your-api-endpoint.com/data"  # Replace with your API endpoint

# Initialize the noise sensor (use the appropriate library or code for your specific sensor)

sound_sensor = sound_sensor_library.initialize()

# Function to read noise level

def read_noise_level():

    noise_level = sound_sensor.get_noise_level()  # Replace with the appropriate method

    return noise_level

# Function to send data to the API

def send_data_to_api(data):

    headers = {"Content-Type": "application/json"}

    payload = {

        "sensor_id": SENSOR_ID,

        "timestamp": int(time.time()),

        "noise_level": data,

    }

    try:

        response = requests. Post(API_ENDPOINT, json=payload, headers=headers)

        if response.status_code == 200:

            print("Data sent successfully")

        else:

            print("Failed to send data. Status code:", response.status_code)

    except Exception as e:
        print("Error:", str(e))

# Main loop

while True:

    noise_level = read_noise_level()

    send_data_to_api(noise_level)

    time. Sleep(60)  # Send data every 60 seconds (adjust as needed)Send Data to Noise Pollution Platform
```

:

Replace

"https://your-noise-platform.com/api/send_data" with the actual URL of your noise pollution information platform. Adjust the payload structure as per the API requirements of the platform.

Run the Script:

Execute the Python script on your IoT device, and it will record noise level data and send it to the platform in real-time.

Data Processing on the Platform:

On the noise pollution information platform, make sure you have a way to receive and process the data sent by the IoT device.

Conclusion:

The IoT-based air pollution monitoring system provides several benefits over traditional air pollution monitoring systems. It can collect real-time data from multiple locations, which then analyzed to identify the sources of pollution. It helps to take necessary measures to reduce it.

SUBMITTED BY:

S KAVIYA

(311421106026)