NOISE POLLUTION MONITORING

PROJECT DOCUMENTATION

Project title: Noise Pollution Monitoring

Project phase: Phase 3-Development Part 1

Objective:

The objective of IOT-based noise pollution monitoring is to accurately measure and analyse ambient noise levels in real-time, providing valuable data for effective urban planning, environmental management, and ensuring public health and well-being.

Sensor used for monitoring:

Noise level monitoring with IOT often involves the use of sound or noise sensors. Some common types include MEMS (Micro-Electro-Mechanical Systems) microphones or specialized noise sened for environmental monitoring. These sensors can detect and measure sound levels, and when integrated into an IOT system, they provide a continuous stream of data for analysis and decision-making.

Libraries Used:

Pyaudio: For audio input.

Numpy: For data processing.

Matplotlib: For data visualization.

Code Steps:

Import Required Libraries-

```
Import pyaudio

Import numpy as np

Import matplotlib.pyplot as plt
```

Initialize Microphone:

Initialize the microphone and configure its parameters (e.g., sample rate and chunk size).

Code:

```
Audio = pyaudio.PyAudio()

Stream = audio.open(format=pyaudio.paInt16, channels=1, rate=44100, input=True, frames_per_buffer=1024)
```

Data Collection and Analysis:

Capture audio data in chunks and analyze it over time. In this example, we'll calculate and visualize the sound levels in decibels (dB).

Code :

```
 # Initialize an empty list to store noise level data

 Noise_levels = []

# Capture data over a specific duration (e.g., 10 seconds)

Duration = 10 # Adjust as needed

For _ in range(0, int(44100 / 1024 * duration)):

 Data = stream.read(1024)

Audio_data = np.frombuffer(data, dtype=np.int16)

 # Calculate the root mean square (RMS) as a noise level indicator

Rms = np.sqrt(np.mean(audio_data**2))

Decibels = 20 * np.log10(rms)

# Append the noise level to the list Noise_levels.append(decibels)
```

Data Visualization:

Plot the recorded noise levels to visualize the noise pollution over time.

Code:

```
# Create a time axis (in seconds)

Time = np.arange(0, duration, duration / len(noise_levels))

 # Plot the noise levels

Plt.figure(figsize=(10, 5))
```

```
 Plt.plot(time, noise_levels)

 Plt.xlabel('Time (s)')

Plt.ylabel('Noise Level (dB)')

 Plt.title('Noise Pollution Monitoring')

 Plt.grid(True)

 Plt.show()
```

Cleanup and End:

Release the microphone and close the audio stream when done.

Code :

```
# Cleanup

 Stream.stop_stream()

 Stream.close()

   Audio.terminate()
```

This code collects and visualizes noise data over a specified duration. Keep in mind that this is a basic example, and real noise pollution monitoring systems may employ more advanced techniques and hardware for accurate and continuous data collection and analysis.

Conclusion :

Implementing noise pollution monitoring using IoT offers a comprehensive solution for real-time data collection and analysis. By deploying smart sensors that continuously measure ambient noise levels, this system provides valuable insights for:

- ➢ Urban Planning
- ➢ Environmental Management
- ➢ Public Health
- ➢ Data-Driven Decision Making
- ➢ Efficient Interventions

In essence, noise pollution monitoring through IoT contributes to creating smarter, healthier, and more livable urban environments by leveraging technology to address a critical aspect of environmental quality.