NOISE POLLUTION MONITORING

Project Objectives:

The objective of the Noise Pollution Monitoring project is to create a real-time system for monitoring noise levels in urban areas. This system aims to:

- Measure noise pollution using IoT sensors deployed across the city.
- Transmit data to a centralized platform for processing and analysis.
- Develop a user-friendly mobile app for accessing noise pollution information.
- Raise public awareness about noise pollution and its impact.
- Contribute to noise pollution mitigation through informed decision-making.

IoT Sensor Deployment:

IoT sensors are strategically deployed across the city, including in residential, commercial, and industrial areas. These sensors consist of microphones and data transmission modules. They continuously capture sound data, which is sent to the central platform for analysis.

Platform Development:

The central platform serves as the core of the system. It collects, processes, and analyzes data from IoT sensors. The platform is hosted on cloud servers and includes a web-based dashboard for administrators. It features a database to store historical data, real-time data processing, and visualization tools for noise level data.

Mobile App Development:

A mobile app is developed for iOS and Android platforms to allow residents to access noise pollution information. The app includes the following features:

- Real-time noise level data display on a map.
- Historical noise data trends and statistics.
- Notification alerts for high noise levels.
- Reporting tools to submit noise complaints.
- Educational content on noise pollution and its impact.

Code Implementation:

- IoT Sensors: The sensor firmware is implemented to record audio data and transmit it to the platform using MQTT or HTTP protocols.
- Platform: The platform is developed using a combination of technologies like Node.js, databases (e.g., MongoDB), and cloud services (e.g., AWS or Azure).
- Mobile App: Native or cross-platform development frameworks like React Native are used to create the mobile app.

Promoting Public Awareness and Mitigation:

The real-time noise level monitoring system promotes public awareness in several ways:

Accessible Information: The mobile app provides residents with easy access to real-time and historical noise data, helping them understand the noise levels in their area.

Alerts and Notifications: The app can send alerts when noise levels exceed acceptable limits, prompting residents to take action or modify their behavior.

Complaint Reporting: Users can report noise complaints through the app, which are then processed by local authorities for action.

Educational Content: The app includes educational content about the health and environmental impacts of noise pollution, raising awareness about its significance.

Data for Decision-Making: The platform provides valuable noise data to city planners and policymakers for informed decision-making in urban planning, zoning, and noise pollution mitigation strategies.

By increasing public awareness and providing data for decision-makers, the system contributes to noise pollution mitigation by promoting responsible behavior and informed urban development.

1. Deploy IoT Sensors:

Choose Hardware:

Select appropriate IoT sensor hardware, such as microphones and microcontrollers. - **Programming Sensors**: Program the sensors to capture audio data at regular intervals and transmit it to a central platform.

Example output:

```
# Example sensor code for data transmission
 import mqtt # or HTTP library
import microphone # capture audio
 import time
while True: audio_data = microphone.capture_audio()
 mqtt.publish(audio_data) # Publish data to the MQTT broker
time.sleep(60) # Send data every minute
```

2. **Develop the Noise Pollution Information Platform**:

- **Select Technology:** Choose a technology stack for your platform (e.g., Node.js, MongoDB, AWS).

- **Set Up a Database:** Create a database to store sensor data. Design the schema for data storage. –
- **Create API Endpoints:** Develop APIs to receive and process sensor data
- **Data Visualization**: Implement data visualization tools for real-time and historical noise data. Example platform UI output:

3. Develop Mobile Apps:

- **Choose a Framework:** Decide whether to build native apps or use cross-platform frameworks like React Native.
- **Implement Features:** Develop the mobile app with features such as real-time data display, historical data trends, noise level alerts, and reporting tools.

4. Integrate Using Python:

Data Transmission:

Write Python scripts to receive and process data from IoT sensors and transmit it to the platform. You can use libraries like paho-mqtt for MQTT or requests for HTTP communication.

```
import paho.mqtt.client as mqtt
import requests
def on_message(client, userdata, message):
audio_data = message.payload.decode
# Send data to the platform
response = requests.post('platform_api_endpoint', json={'audio_data': audio_data})
client = mqtt.Client()
client.on_message = on_message client.connect('mqtt_broker_url', 1883)
client.subscribe('sensor_data_topic')
client.loop_forever()
```

Data Processing:

Use Python to process and analyze the data on the platform. You can employ data analysis libraries and frameworks such as Pandas or NumPy.

5. Example Outputs:

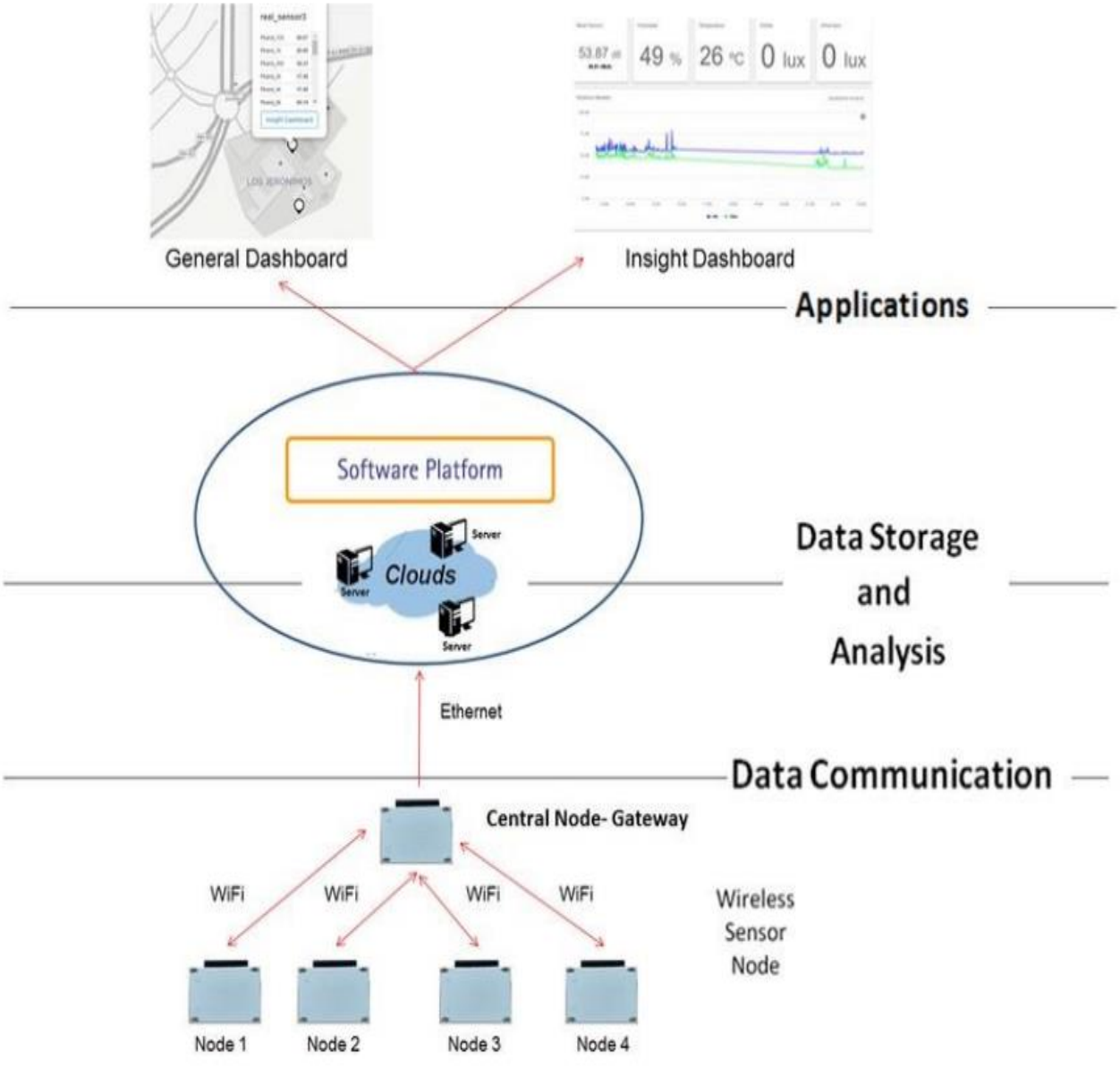
- Example IoT Sensor Data:

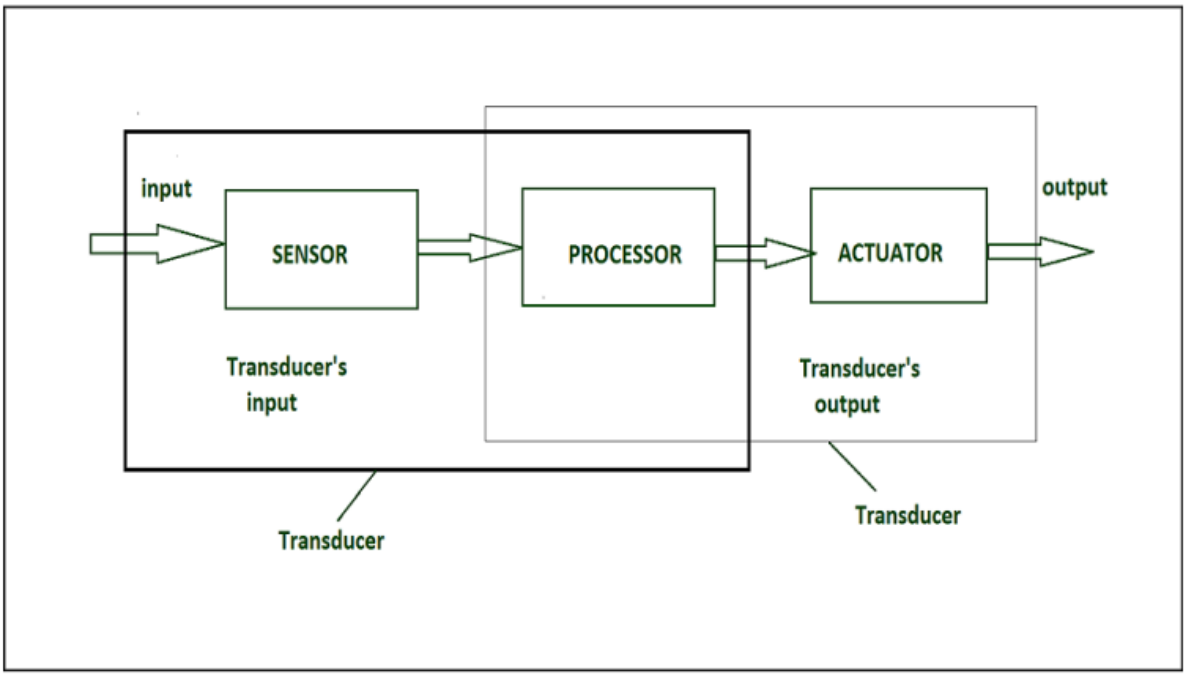
{"timestamp": "2023-01-01T12:00:00", "noise_level": 68.5}

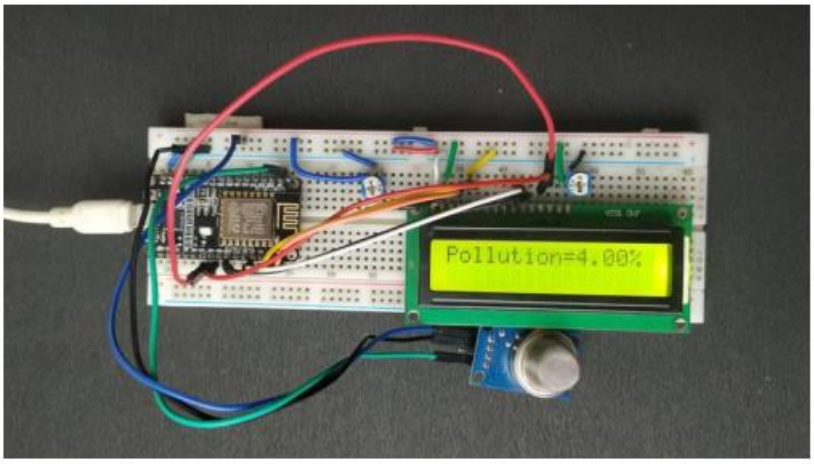
- Example Platform UI (visualizing noise data):

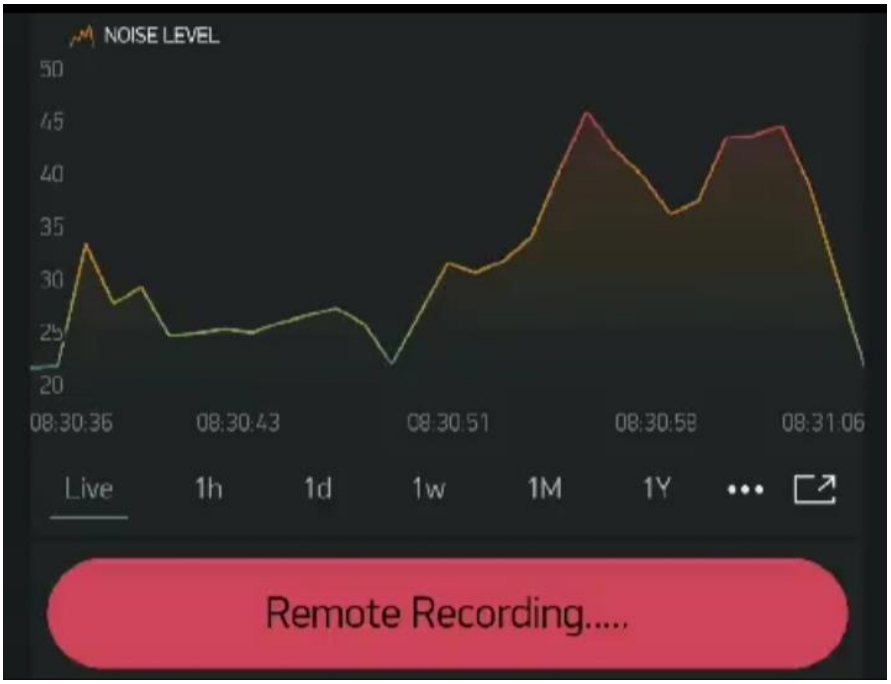
- Example Mobile App Interface (real-time noise display):

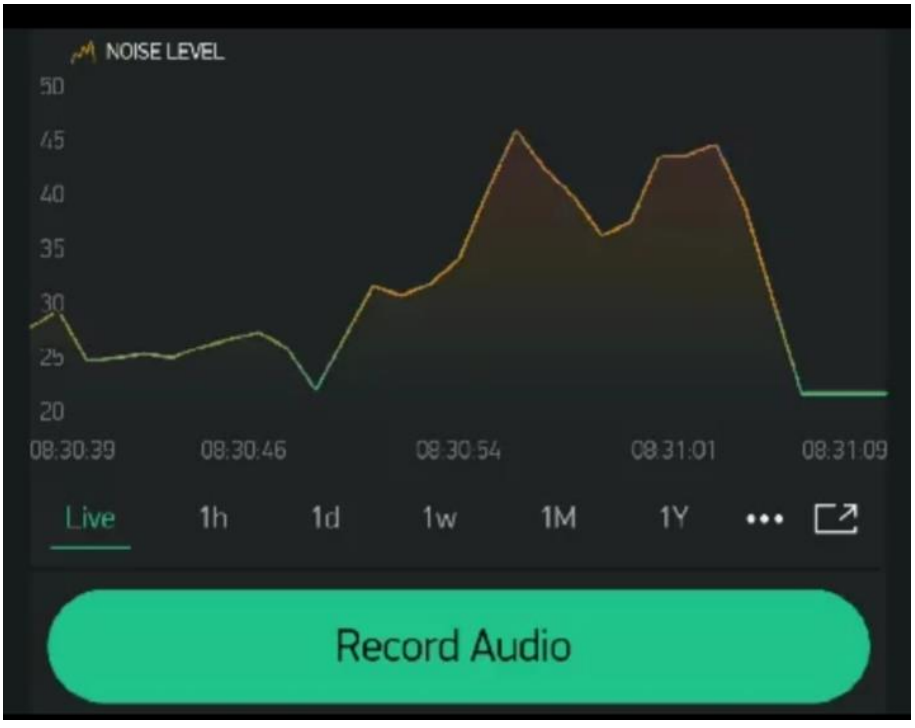
Diagrams and Screenshots



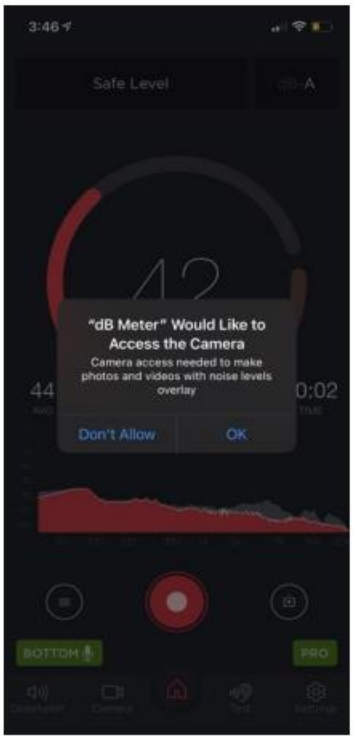

input
SENSOR
PROCESSOR
ACTUATOR
output
Transducer's
input
Transducer's
output
Transducer
Transducer


Pollution=4.00%


NOISE LEVEL
50
45
40
35
30
25
20
08:30:36
08:30:43
08:30:51
08:30:58
08:31:06
Live
1h
1d
1w
1M
1Y
Remote Recording.....


NOISE LEVEL
50
45
40
35
30
25
20
08:30:39
08:30:46
08:30:54
08:31:01
08:31:09
Live
1h
1d
1w
1M
1Y
Record Audio


3:46
Safe Level
dB-A
42
44
42
45
00:02
BOTTOM
PRO


3:46
Safe Level
dB-A
"dB Meter" Would Like to Access the Camera
Camera access needed to make photos and videos with noise levels overlay
Don't Allow
OK
BOTTOM
PRO