**IOT\_phase 4**

**Noise pollution monitoring**

Team members

Aravindan p(210621106006)

Ajai m(210621106003)

Avinash k(210621106008)

Bathmanaban v(21062106009)

**INTRODUCTION**

Noise pollution monitoring is essential for assessing and addressing the impact of excessive noise on the

environment and human health.

\*\*Hardware Setup:\*\*

* Choose suitable IoT noise sensors (e.g., microphones) and microcontrollers (e.g., Raspberry Pi or

Arduino).

* Assemble the hardware components and connect the sensors to the microcontroller.

2. \*\*Software Development:\*\*

* Develop a Python script for the microcontroller to capture noise level data from the sensors.
* Implement data preprocessing and filtering to ensure accurate readings.
* Use libraries like `RPi.GPIO` for Raspberry Pi or `Adafruit\_CircuitPython` for Adafruit boards to

Interface with sensors.

3. \*\*Data Transmission:\*\*

* Set up communication protocols (e.g., MQTT, HTTP, or WebSocket) to send data from the sensors to

The central platform.

4. \*\*Central Platform:\*\*

* Create a cloud-based or local server to receive and store the incoming noise data.
* Develop a web application or API for data visualization and analysis.
* MongoDB to store data.
* 5. \*\*Real-time Data Processing:\*\*

- Implement real-time data processing for immediate insights.

- You may want to use tools like Apache Kafka or RabbitMQ for message queuing and processing.

* 6. \*\*User Interface:\*\*

- Create a user-friendly dashboard to visualize noise pollution data.

- Use HTML, CSS, and JavaScript for the frontend, and libraries like D3.js for data visualization.

* 7. \*\*Alerts and Notifications:\*\*

- Add alerting mechanisms to notify relevant authorities or the public when noise levels exceed

* Predefined thresholds.
* 8. \*\*Security:\*\*
* - Ensure data security and authentication measures to protect the system from unauthorized access.
* 9. \*\*Power Management:\*\*
* - Optimize power usage for IoT devices, especially if they run on batteries.
* 10. \*\*Testing and Calibration:\*\*

- Thoroughly test the system, calibrate sensors, and validate data accuracy.

* 11. \*\*Deployment:\*\*

- Deploy IoT sensors in public areas, ensuring they are well-protected from environmental conditions.

- Monitor and maintain the system to ensure its continued functionality.

**NOISE POLLUTION MONITORING APP**

Creating a noise pollution monitoring app would be a valuable project.

We would need to incorporate features like real-time noise level

measurements, location tagging, and data visualization. We can

Consider using a smartphone's built-in microphone and GPS capabilities

for data collection. Additionally, you could allow users to report noise

disturbances and provide educational resources on noise pollution.

**Features**:

**Real-time Noise Level Monitoring**

Utilize the smartphone's microphone to measure and display current

noise levels in decibels (dB).Provide a continuous real-time graph to

show noise level fluctuations.

**Noise Disturbance Reporting:**

Enable users to report noise disturbances by recording audio snippets and providing descriptions

**Development Considerations**:

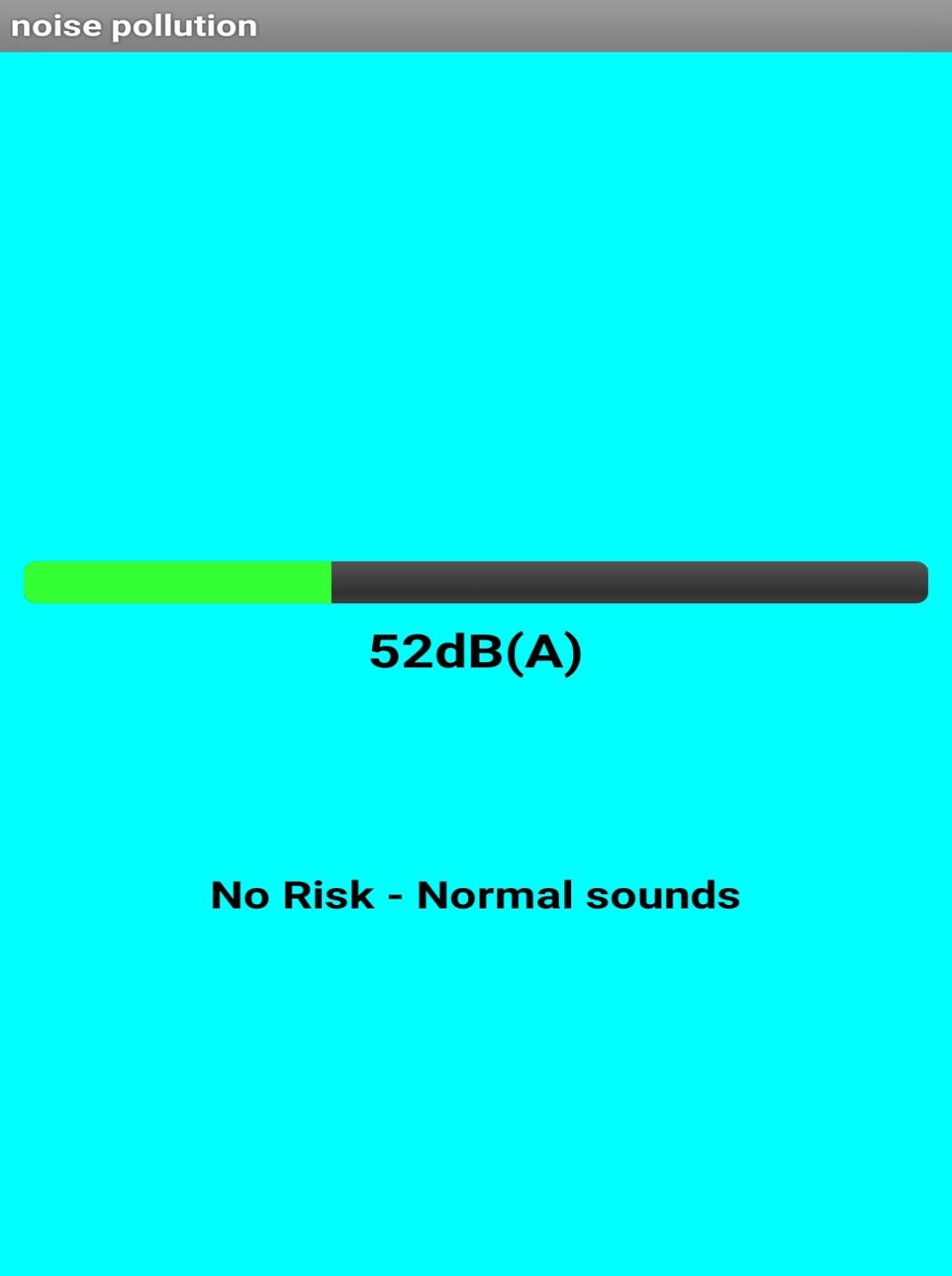
Choose a suitable platform for app development (iOS, Android, or

both).Utilize sound processing algorithms to accurately measure noise

levels.Implement data storage and management, considering

scalability.Secure user data and ensure privacy compliance.Test the app

extensively to ensure accurate noise measurements and user-friendly functionality



Link : <https://drive.google.com/drive/folders/1YjhPlc9ZAk867RiJxe7JytnUMnwMsZus>

**WEB CODE**

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Real-Time Noise Level Data</title>

<script>

// Function to update the noise level

function updateNoiseLevel() {

// Simulate real-time data (replace with actual data retrieval)

const noiseLevel = Math.floor(Math.random() \* 101); // Random noise level between 0 and 100 dB

// Update the displayed noise level

document.getElementById('noise-level').textContent = `Noise Level: ${noiseLevel} dB`;

}

// Update the noise level every 5 seconds (adjust as needed)

setInterval(updateNoiseLevel, 5000);

// Initial update

updateNoiseLevel();

</script>

</head>

<body>

<h1>Real-Time Noise Level Data</h1>

<div id="noise-data">

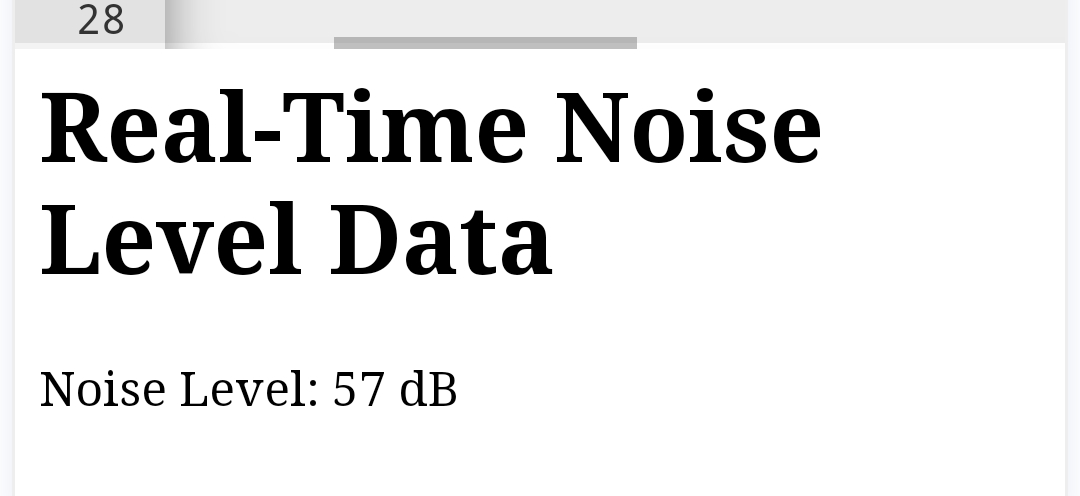
<p id="noise-level">Loading...</p>

</div>

</body>

</html>

**OUTPUT**



**CONCLUSION**

This app and the html code above will record the sound using the microphone and give the real time noise level data in db.