# PHASE -5

# NOISE POLLUTION MONITORING

# 

COLLEGE CODE : 5113

COLLEGE NAME : KINGSTON ENGINEERING

DOMAIN : IOT

PROJECT TITLE : NOISE POLLUTION

MONITORING SYSTEM

PROJECT MEMBER: NAAN MUDHALVAN ID

D NAVIN(LEADER) 511321104062

D NAMDEV 511321104060

B PURUSHOTH 511321104076

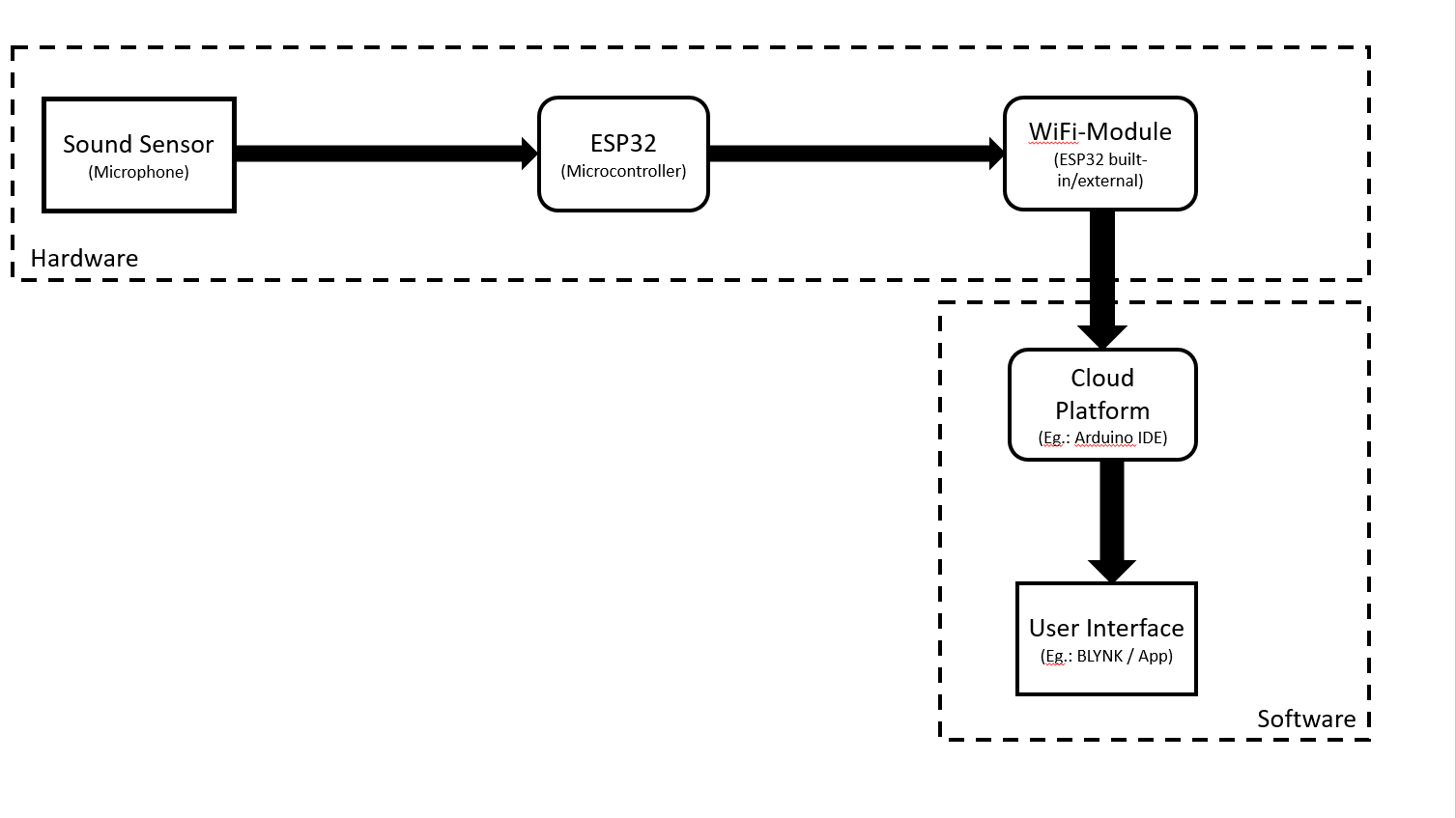
H SAI JASWANTH 511321104083

T VAMSI KRISHNA 511321104108

**OBJECTIVES :**

**The objectives of the project is noise pollution monitoring systems use connected devices, such as sensors and microphones, to continuously measure and track sound levels in real-time. It also known as environmental noise or sound pollution, is the propagation of noise with harmful impact on the activity of human or animal life. To regulate and control noise producing and generating sources. Maintaining the ambient air quality standards in respect of noise.**

**BLOCK DIAGRAM :**

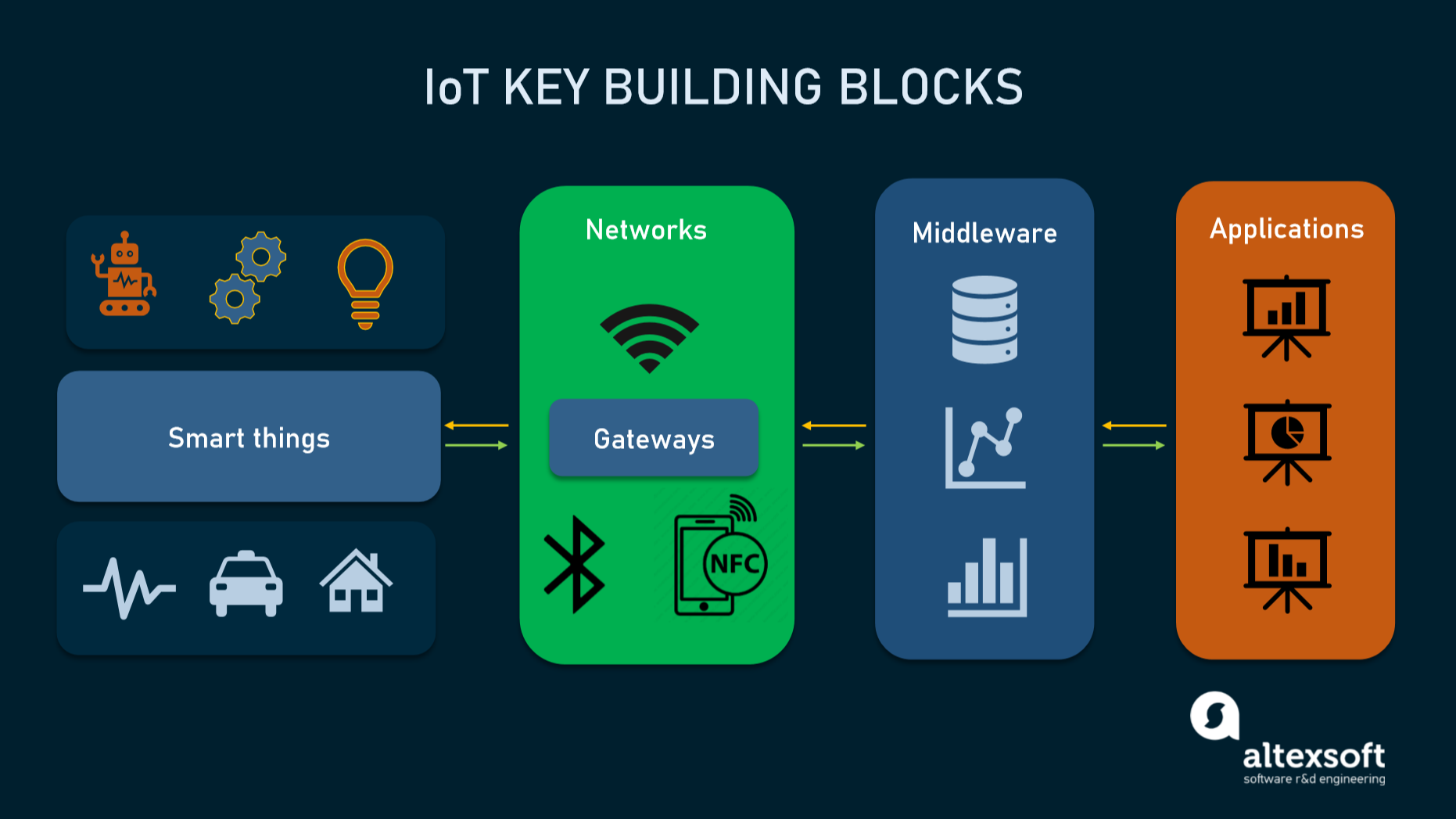


**IoT DEVICE SETUP :**

**Microphone sensor: It involves choosing appropriate sensors (like microphones), integrating them with a microcontroller or a single-board computer, programming it to collect**

**noise data, transmitting this data to a central server or cloud for storage and analysis, and finally, creating a user interface or system for accessing and visualizing the noise level data.**

**It's crucial to ensure accurate sensor placement, proper calibration, and data security measures for effective noise monitoring.**

****

**PLATFORM DEVELOPMENT :**

**Hardware Selection: Choose appropriate sensors capable of capturing noise levels. Common choices include microphones or sound level meters. Ensure they're compatible with IoT platforms.**

**sensor screenshot:**



Choose appropriate noise sensors (e.g., microphones or sound level meters) with the necessary sensitivity and accuracy to measure noise pollution effectively.

Consider factors such as frequency range, resolution, and dynamic range to ensure the sensors capture a wide range of noise levels accurately.

Develop a sensor calibration procedure to ensure data accuracy.

**Prototyping: Develop a prototype system by connecting the sensors to a microcontroller (like Arduino or Raspberry Pi) capable of gathering and transmitting data.**

**Data Transmission: Utilize IoT communication protocols (like MQTT or HTTP) to send the collected data from the sensors to a central server or cloud platform.**

**Cloud Integration: Create a cloud infrastructure (using platforms like AWS, Azure, or Google Cloud) to receive, store, and process the data. Implement security measures for data integrity.**

**Data Processing and Analysis: Use software to process and analyze the collected noise data. This might involve signal processing, machine learning, or statistical analysis to derive insights.**

**User Interface Development: Design a user interface, such as a web or mobile application, to display the noise data. Ensure it's user-friendly and provides meaningful insights and alerts.**

**Alerting and Reporting: Implement a system that alerts relevant parties (authorities, users, etc.) when noise levels exceed predefined thresholds. Create reports or visualizations for comprehensive analysis.**

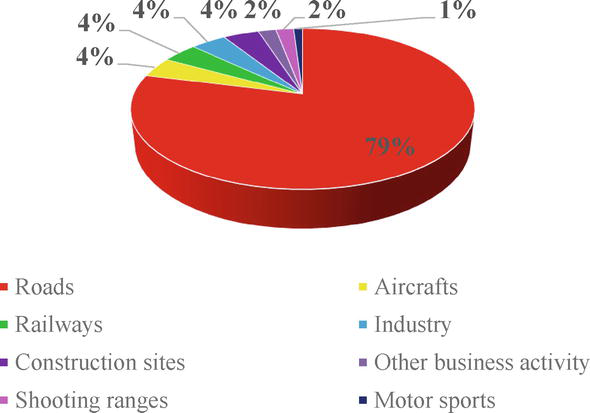
**Testing and Calibration: Thoroughly test the system to ensure accuracy and reliability. Calibrate sensors and algorithms for optimal performance.**

**Deployment: Once tested, deploy the system in the desired locations for real-time noise monitoring.**

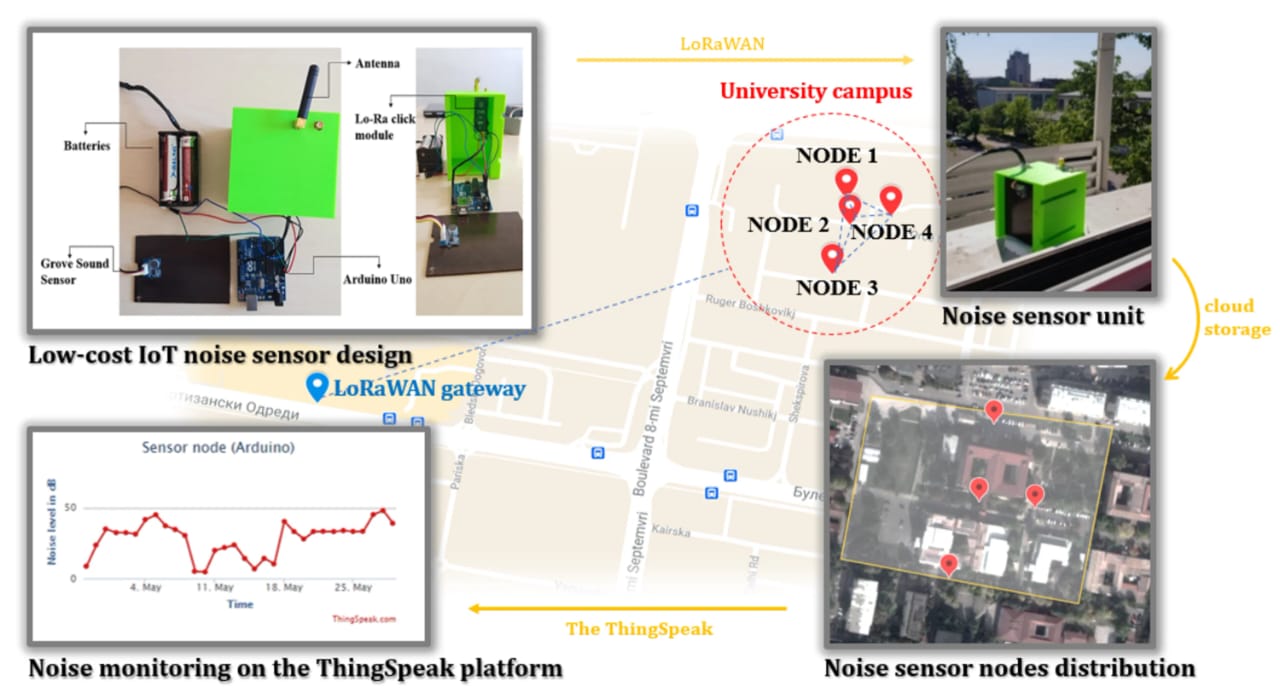
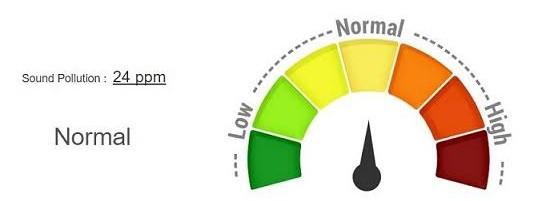
**Maintenance and Upgrades: Regularly maintain the system, update software, and make necessary improvements based on user feedback and changing requirements.**

**EXPLANATION DETAIL :**

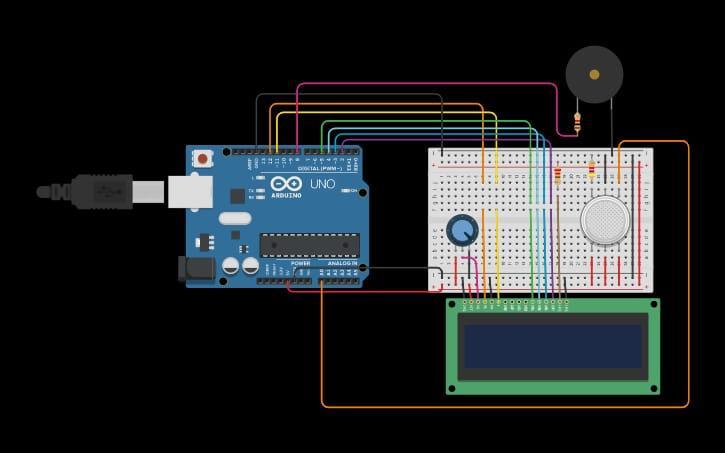
**The system will collect real time data at workplace machines, traffic,and vehicles create occupational noise. Employees and occupants are exposed to this harmful noise. Due to this occupants can face many health problems such as headache, hearing impairment, hypertension, heart problem, annoyance and sleep disorder. to Avoid this situation Perfect Pollucon Services conducts Noise Testing in the company or home to measure high noise within premises. Actions can be taken once the locations and source are identified which are causing the noise. it helps to keep your employee or family safe from Harmful Noise.**

****

**Noise or sound level monitoring or measurement is a process to measure the magnitude of Noise in industries and residential area. Data collected from Noise level monitoring & Testing helps us to understand trends and action can be taken to reduce noise pollution. Noise pollution is Low or High-frequency sound that can cause/harm the activity of human life. It can be caused by various industrial Machines, Motor Vehicles and Craft etc. Noise Pollution Monitoring process is a part of Environmental Monitoring & Testing as noise pollution is also increasing exponentially in recent years.**

****

**CIRCUIT FOR NOISE MONITORING :**

****

**CODE IMPLEMENTATION:**

**To replicate the project ,you would need to follow these steps:**

**1.iot sensor setup**

**2.mobile app development**

**3.raspberry pi integration**

**4.code implementation example**

**Install the necessary libraries on the raspberry pi,such as request ,for making http requests.**

**Write a python script on raspberry pi to collect sensor data and send it to transit information platform’s api endpoint .here’s an example**

**PYTHON CODE:**

import sounddevice as sd

import numpy as np

import csv

import time

# Define the parameters

duration = 3600 # Recording duration in seconds (1 hour)

sample\_rate = 44100 # Sampling rate in Hz

channels = 1 # Mono audio

# Create an empty list to store the sound level data

sound\_levels = []

# Function to calculate the decibel level

def calculate\_decibel\_level(audio\_data):

rms = np.sqrt(np.mean(audio\_data\*\*2))

decibel\_level = 20 \* np.log10(rms)

return decibel\_level

# Callback function for recording audio

def audio\_callback(indata, frames, time, status):

if status:

print("Error:", status)

decibel\_level = calculate\_decibel\_level(indata) sound\_levels.append(decibel\_level)

# Start recording

with sd.InputStream(callback=audio\_callback, Channels=channels, samplerate=sample\_rate):

print(f"Recording for {duration} seconds...")

sd.sleep(duration \* 1000) # Sleep for the recording duration

# Save the recorded data to a CSV file

timestamp = time.strftime("%Y%m%d%H%M%S")

filename = f"sound\_levels\_{timestamp}.csv"

with open(filename, 'w', newline='') as csvfile:

csv\_writer = csv.writer(csvfile)

csv\_writer.writerow(['Time', 'Decibel Level (dB)'])

for i, level in enumerate(sound\_levels):

csv\_writer.writerow([i / sample\_rate, level])

print(f"Data saved to {filename}"

**5.develop the mobile app ui**

**Certainly !here are some exaamples outputs of raspberry pi data transmission and a mobile app user interface for a traffic management system**

**UI CODE IMPLEMENTATION:**

HTML CODE:

<!DOCTYPE html>

<html>

<head>

<title>Noise Pollution Monitoring App</title>

<style>

/\* Add CSS styles for the layout and design \*/

</style>

</head>

<body>

<h1>Noise Pollution Monitoring</h1>

<div id="noise-level">

<h2>Real-time Noise Level</h2>

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

</div>

<!-- Add any additional elements for user interaction or data visualization -->

<script>

// JavaScript for fetching and displaying real-time noise data

// Function to fetch noise data from your backend (simulated in this example)

function fetchNoiseData() {

// Replace with actual API endpoint to get real-

time data

const apiUrl = "https://your-api-url.com/noise-

data";

// Simulated response (replace with actual API

request)

const simulatedResponse = {

noiseLevel: 70, // Replace with actual noise

level data

timestamp: new Date().toLocaleString(),

};

// Update the noise level value on the page

const noiseValueElement =

document.getElementById("noise-value");

noiseValueElement.textContent =

`${simulatedResponse.noiseLevel} dB (as of

{simulatedResponse.timestamp})`;

}

// Fetch noise data initially and set up periodic

updates

fetchNoiseData();

setInterval(fetchNoiseData, 60000);

// Update every minute (adjust as needed)

</script>

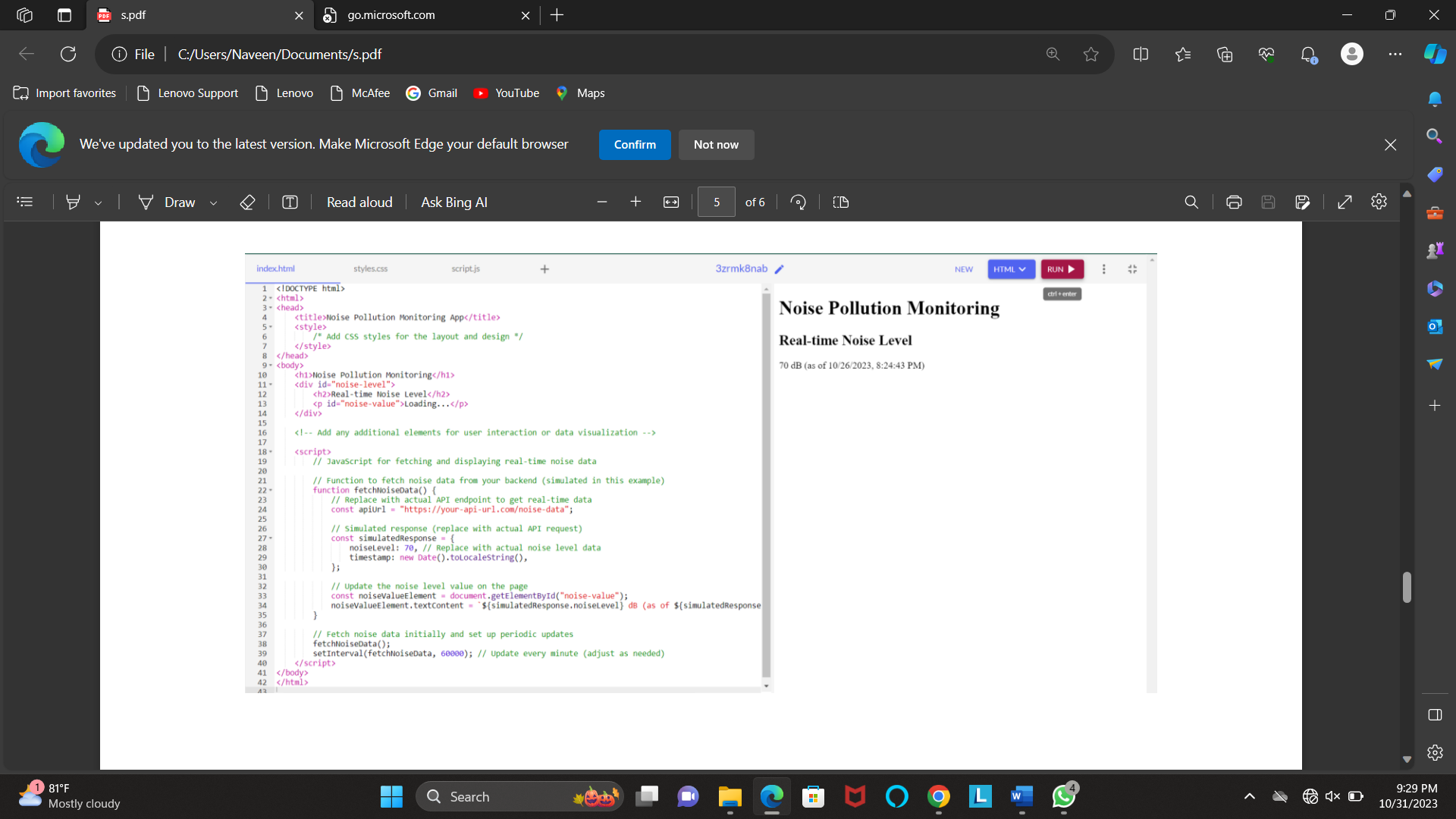
</body>

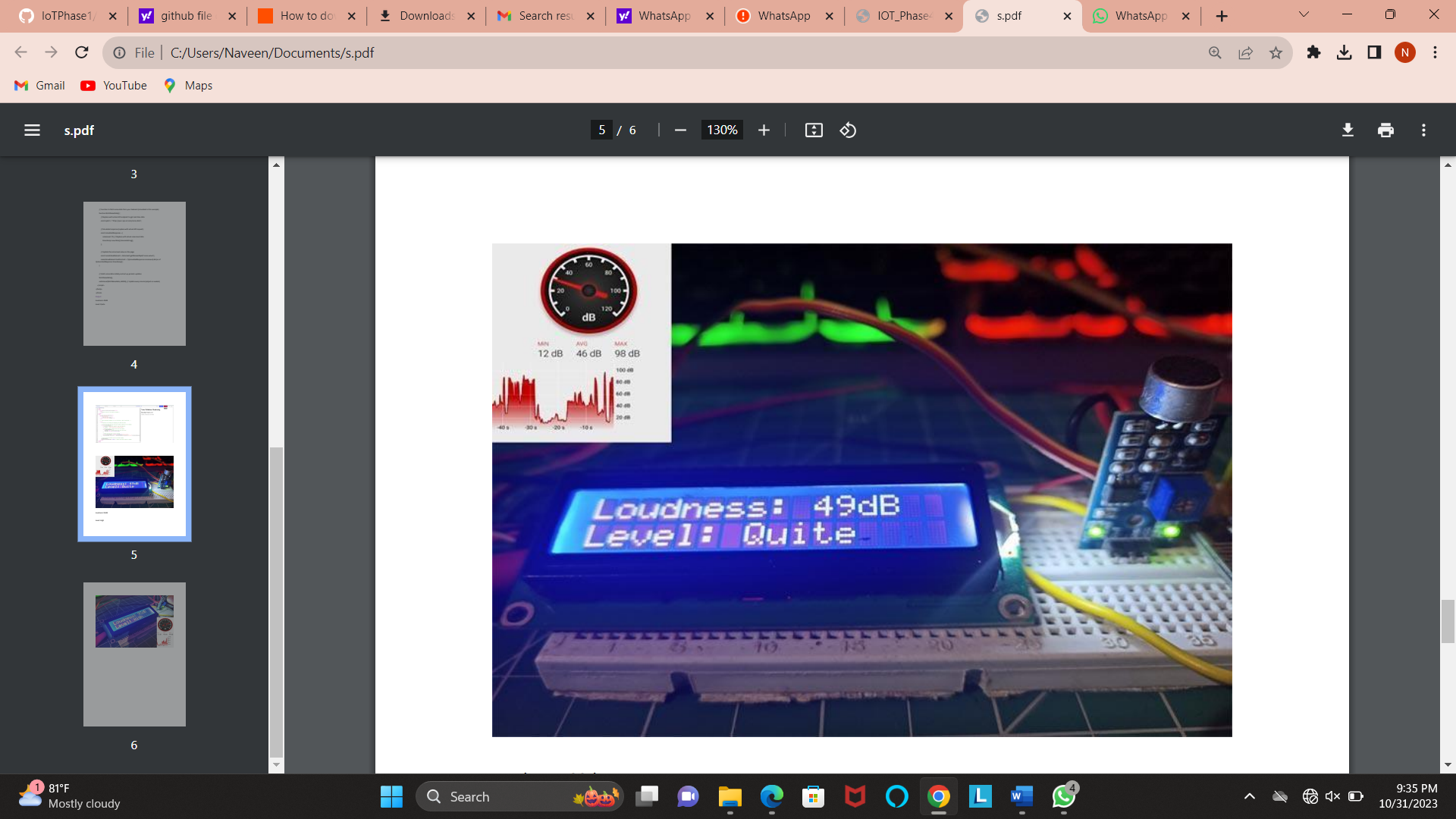
</html>

Output:

Loudness: 49dB

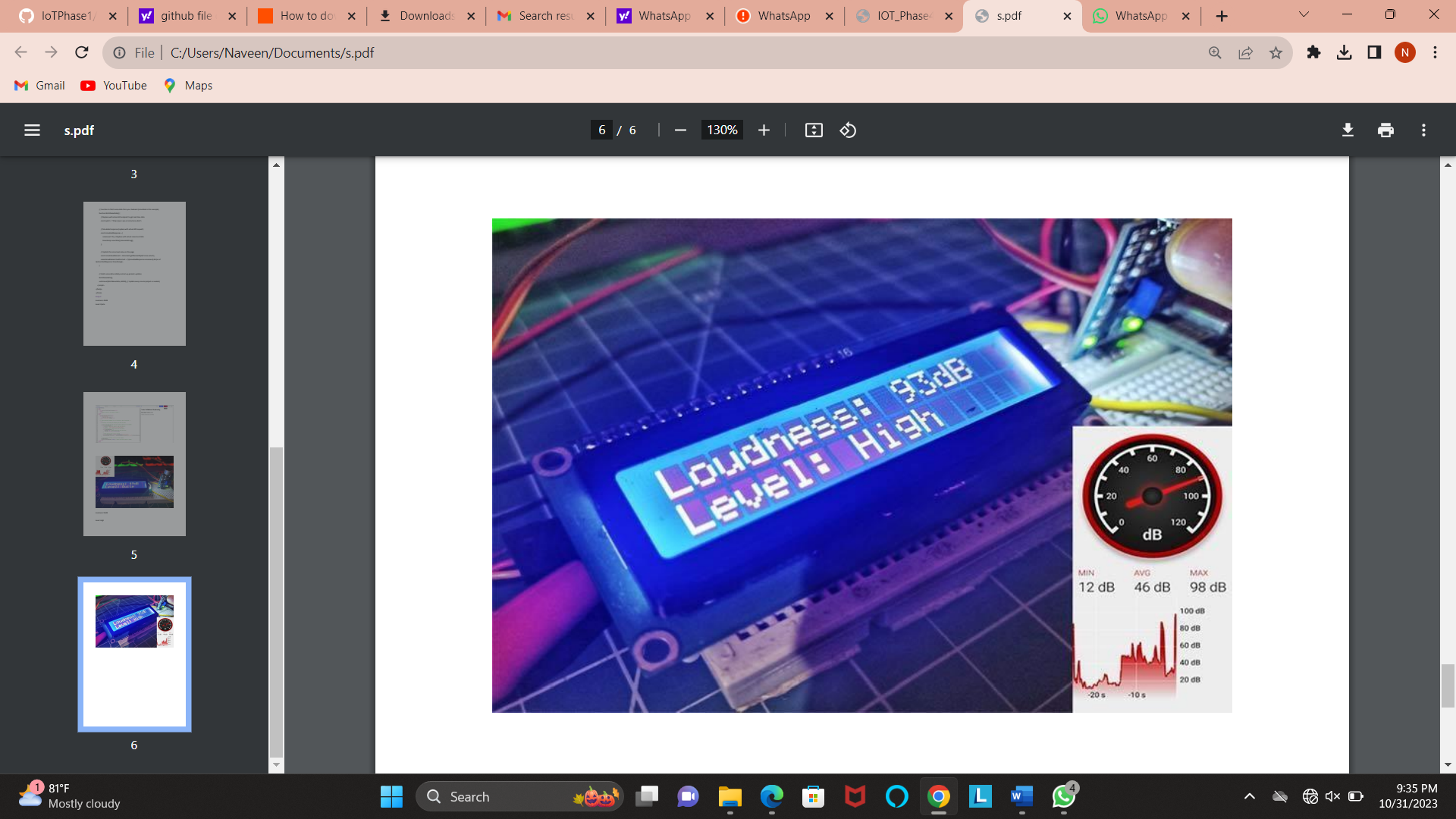
Level: Quite





Loudness: 93dB

Level: High



**CONCLUSION :**

**The monitoring noise pollution is essential for understanding its impact on the environment and public health. By employing advanced monitoring techniques and data analysis, we can identify sources, assess exposure levels, and implement effective mitigation strategies. Continued research and monitoring efforts are crucial to creating healthier, more peaceful environments for communities worldwide.**