INTERNET OF THINGS – GROUP 5
PROJECT SUBMISSION
PART – 5

TITLE : NOISE POLLUTION MONITORING SYSTEM

INTRODUCTION:

The Noise Pollution Monitoring System is a project aimed at monitoring and analyzing noise pollution using IoT sensors. This system integrates IoT sensor deployment, mobile application development, web-based data visualization, and offers real-time noise monitoring and analysis. The primary objective is to provide accessible and actionable noise pollution data, empowering users to make informed decisions, promote environmental awareness, and work towards quieter, healthier communities.

OBJECTIVES :

1.Noise Data Collection :

- Collect real-time noise data from IoT sensors deployed in a specific area.
- Continuously monitor noise levels to provide accurate and up-to-date information.

2. Data Storage and Management :

- Store noise data efficiently and securely using Firebase.
- Enable data management, retrieval, and analytics for further study.

3. User Accessibility :

- Develop a user-friendly mobile app using MIT App Inventor for Android users.
- Create a web-based interface (HTML, CSS, and JavaScript) for desktop users.
- Ensure that users can easily access and understand the noise data.

4. Real-time Monitoring :

- Provide real-time access to noise data for users through the app and website.
- Enable users to monitor noise levels as they change over time.

5. Data Visualization :

- Display noise data in a visually comprehensible manner through graphs, charts, and maps.
- Aid users in identifying noise pollution trends and patterns.

6. Historical Data Analysis :

- Allow users to access historical noise data for trend analysis.
- Provide insights into long-term noise pollution patterns and changes.

7. User Authentication and Security :

- Implement user authentication and secure data access to protect sensitive information.
- Ensure the privacy and security of user accounts and data.

8. Sensor Simulation for Development :

- Use Wokwi for sensor simulation, facilitating system testing and development without physical sensors.

9. Environmental Awareness :

- Raise awareness about noise pollution in the monitored area.
- Educate the public on the impact of noise pollution on health and the environment.

10. Community Engagement :

- Encourage community involvement in noise pollution monitoring.

- Foster collaboration between local authorities, environmental organizations, and concerned citizens.

11. Data-Driven Decision-Making :

- Enable local authorities and policymakers to make informed decisions based on noise pollution data.

- Implement noise control measures and policies to reduce noise pollution in problem areas.

12. Scalability and Future Integration :

- Design the system with scalability in mind, allowing for the addition of more sensors or integration with other environmental monitoring devices.

- Plan for future enhancements and integration with additional sensors, mapping tools, and notification systems.

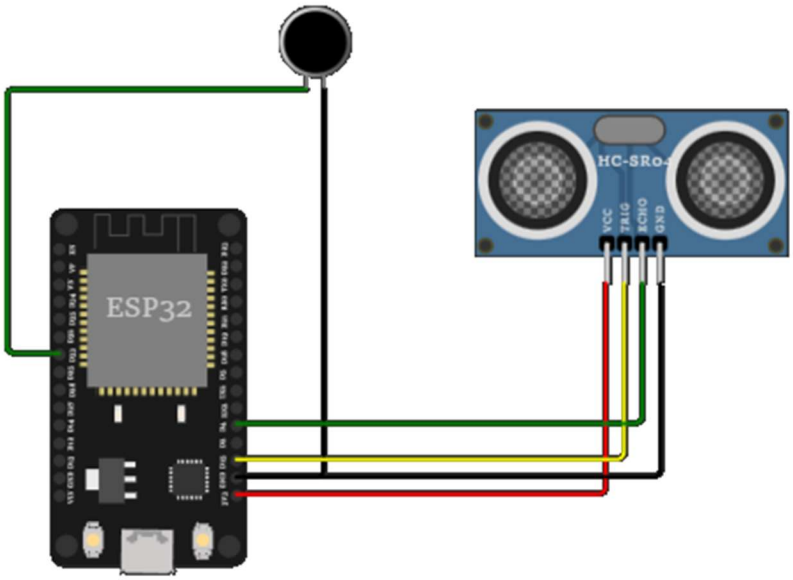
13. Community Health and Well-Being :

- Contribute to the improvement of community health and well-being by reducing noise pollution.

- Provide a tool for communities to actively address and mitigate the negative effects of excessive noise.

IOT SENSOR SIMULATION

We created a simple noise pollution monitoring system using wokwi.



Diagram

JSON :

```
{
  "version": 1,
  "editor": "wokwi",
  "parts": [
    {
      "type": "wokwi-esp32-devkit-v1",
      "id": "esp",
```

```
    "top": -139.3,
    "left": 14.2,
    "attrs": { "env": "micropython-20231005-v1.21.0" }
   },
   { "type": "wokwi-microphone", "id": "mic", "top": -141.78, "left": 158.19, "attrs": {} },
   {
    "type": "wokwi-hc-sr04",
    "id": "ultrasonic1",
    "top": -142.5,
    "left": 226.3,
    "attrs": { "distance": "88" }
   }
 ],
 "connections": [
   [ "esp:TX0", "$serialMonitor:RX", "", [] ],
   [ "esp:RX0", "$serialMonitor:TX", "", [] ],
   [ "mic:2", "esp:GND.1", "black", [ "v0" ] ],
   [ "ultrasonic1:VCC", "esp:3V3", "red", [ "v0" ] ],
   [ "ultrasonic1:TRIG", "esp:D15", "yellow", [ "v0" ] ],
   [ "ultrasonic1:ECHO", "esp:D4", "green", [ "v0" ] ],
   [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ],
   [ "mic:1", "esp:D32", "green", [ "v28.8", "h-153.6" ] ]
 ],
 "serialMonitor": { "display": "plotter" },
 "dependencies": {}
}
```

PROGRAM FOR SIMULATION :

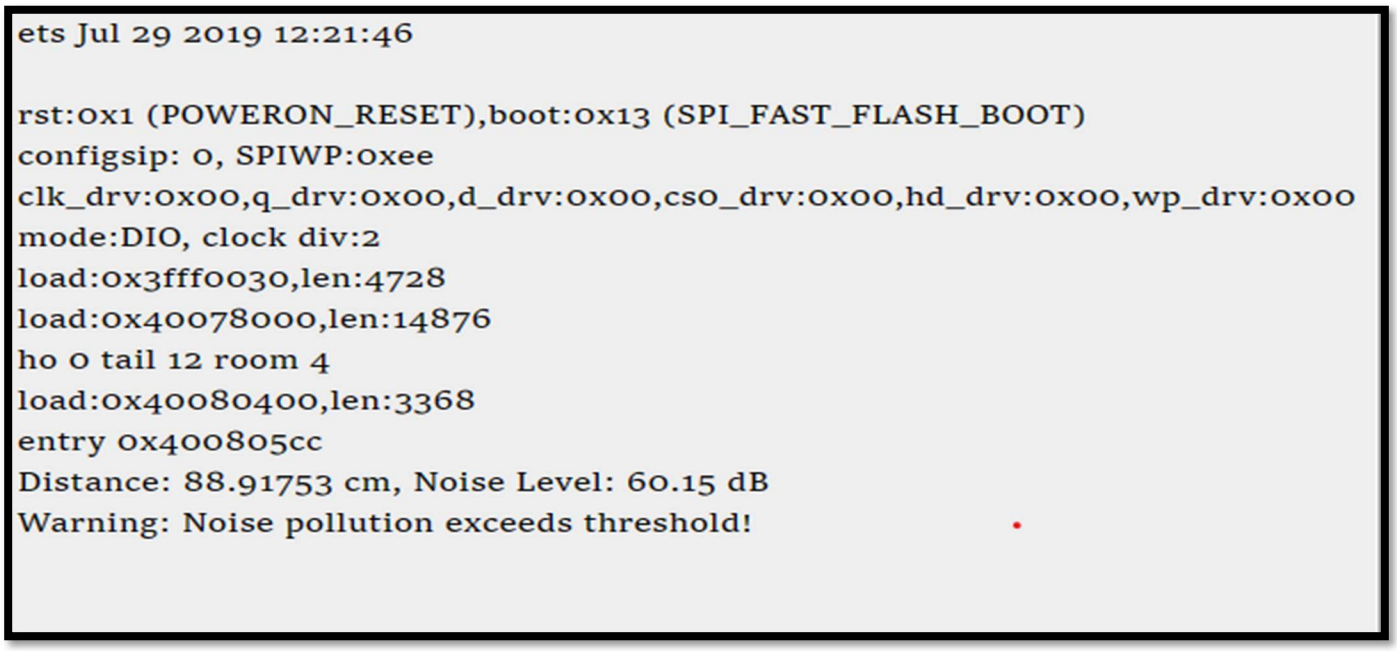
```
import machine
```

```
import time
import urequests
import ujson
import network
import math

wifi_ssid = 'Wokwi-GUEST'
wifi_password = ''

wifi = network.WLAN(network.STA_IF)
wifi.active(True)
wifi.connect(wifi_ssid, wifi_password)
while not wifi.isconnected():
    pass
ultrasonic_trig = machine.Pin(15, machine.Pin.OUT)
ultrasonic_echo = machine.Pin(4, machine.Pin.IN)
microphone=machine.ADC(machine.Pin(32))
calibration_constant = 2.0
noise_threshold = 60
firebase_url = 'https://noise-pollution-monitering-default-rtdb.firebaseio.com/'
firebase_secret = 'pZDccYblvZyOrZkhKroWFqosgpEmA24pbief6KNf'

def measure_distance():
    ultrasonic_trig.value(1)
    time.sleep_us(10)
    ultrasonic_trig.value(0)
    pulse_time = machine.time_pulse_us(ultrasonic_echo, 1, 30000)
   distance_cm = (pulse_time / 2) / 29.1
    return distance_cm
```

```
def measure_noise_level():
    noise_level = microphone.read()
    noise_level_db = 20 * math.log10(noise_level / calibration_constant)
    return noise_level, noise_level_db

def send_data_to_firebase(distance, noise_level_db):
    data = {
        "Distance": distance,
        "NoiseLevelDB": noise_level_db
    }
    url = f'{firebase_url}/sensor_data.json?auth={firebase_secret}'
   try:
        response = urequests.patch(url, json=data)
        if response.status_code == 200:
            print("Data sent to Firebase")
        else:
            print(f"Failed to send data to Firebase. Status code: {response.status_code}")
    except Exception as e:
        print(f"Error sending data to Firebase: {str(e)}")
   try:
    while True:
        distance = measure_distance()
        noise_level, noise_level_db = measure_noise_level()
        print("Distance: {} cm, Noise Level: {:.2f} dB".format(distance, noise_level_db))
        if noise_level_db > noise_threshold:
            print("Warning: Noise pollution exceeds threshold!")
            send_data_to_firebase(distance, noise_level_db)
            time.sleep(1)
```

```
except KeyboardInterrupt:
    print("Monitoring stopped")
```

Output for simulation:

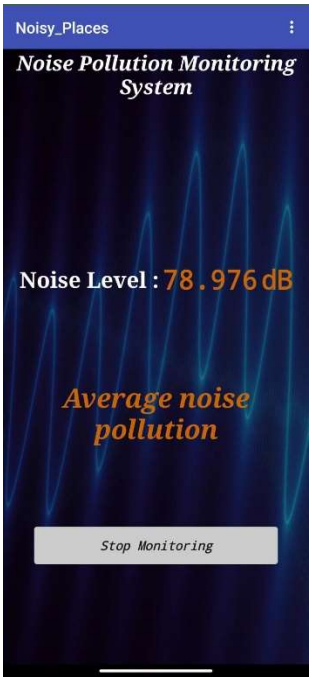


This is how we simulated the IoT sensor using wokwi site.

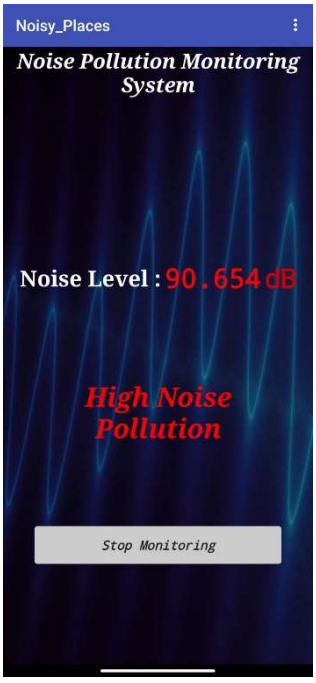
NOISE POLLUTION MONITORING APPLICATION

Front page of the application

Good Range Noise

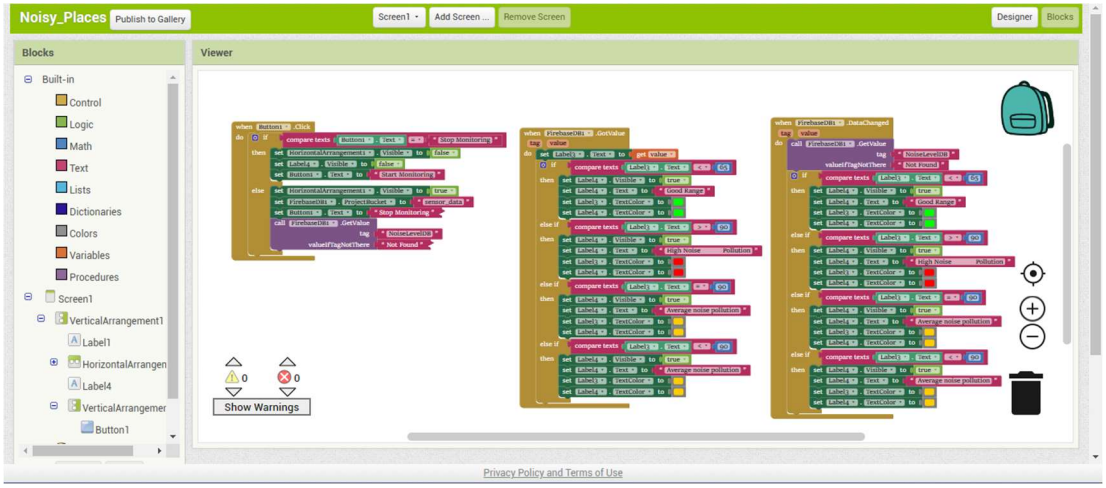


Average Range Noise



High Range Noise

Blocks of MIT app inventor



About the App :

We've created a noise pollution monitoring app using MIT App Inventor, and it's linked to Firebase for data management. It receives noise level data sensors and then stores this data in Firebase. Users can access real-time noise level information along with warnings based on predefined thresholds. This app shows the noise level and gives the alert along with it. This project aims to enhance awareness of noise pollution and empower individuals to take informed actions to mitigate its impact on health and well-being.

NOISE POLLUTION MONITORING WEB APPLICATION

Our project is a web-based noise pollution monitoring application developed using a stack of technologies that include HTML, CSS, JavaScript, Node.js, and EJS (Embedded JavaScript). Here's how it works:

1. Frontend Development:

We've created an interactive user interface using HTML and CSS. This provides users with a visually appealing and intuitive way to interact with the app.

2. Real-Time Data Display:

JavaScript is utilized to enable real-time data display on the web app. It connects to a backend server to fetch and update noise level data continuously.

3. Backend with Node.js:

We've used Node.js as our backend framework to handle data processing and manage server-side operations. Node.js is known for its efficiency and non-blocking I/O, making it a great choice for real-time applications.

4. Templating with EJS:

EJS (Embedded JavaScript) is employed for templating. It allows us to generate dynamic HTML content based on the data received from the server, enabling real-time updates in the user interface.

5. Data Collection and Firebase Integration:

The app collects noise level data from various sources, such as sensors or external APIs. This data is processed, analyzed, and stored in Firebase, a real-time database, allowing for efficient data management and retrieval.

6. User Interaction:

Users can access noise level information, receive warnings or alerts when noise levels exceed predefined thresholds, and customize their notification preferences. This empowers individuals to make informed decisions about their surroundings.

7. Scalability and Efficiency:

Node.js, with its asynchronous capabilities, ensures that our app is efficient and can handle multiple simultaneous users without performance degradation.

By using this technology stack, we've created a powerful web app that not only informs users about noise pollution but also provides a platform for raising awareness and actively addressing noise-related issues. The project demonstrates how modern web technologies can be harnessed to promote a quieter and healthier environment."

HTML code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Noisy_Places</title>
    <link rel="icon" href="bi.jpg">
    <style>body{
```

```
    text-align: center;
    background-image: url('bi.jpg');
    background-repeat:inherit;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;

}

h1{
    font-family:monospace;
    font-size: 50px;
    color: aliceblue;
}


#Noise{
    color: aliceblue;
    font-size: 30px;
    margin-top: 100px;
    display:none;
}

#NoiseTitle{
    font-weight: bold;
}
```

```
#NoiseTitle,#value{
    font-size: 40px;
    color: aliceblue;
}
 #warning{
    color: aliceblue;
    font-weight: bold;
    font-size: 2em;
}
a{
    font-size: 20px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
     color: #000000;
     text-decoration: none;
     background-color: aliceblue;
     padding: 5px 10px;
     border-radius: 7px;
     width: 20%;
     display: inline-block;
     margin: 80 auto;
     margin-top: 100px;
     transition: 0.2s cubic-bezier(0.075, 0.82, 0.165, 1);
}
a:hover{
    transform: scale(1.05);
```

```
    box-shadow: #000 -3px -3px;
}

button:hover{
    cursor: pointer;
}

details,summary{
    margin: 100px 50px 0px;
    color: aliceblue;
   font-size: larger;
   cursor: pointer;
   line-height: 1.7rem;
   transition: margin 500ms ease;
   font-family:cursive;
   font-size: 1.3rem;
   text-align: left;
}
summary{
    margin-top: 50px;
    font-size: 1.3rem;
    text-align: center;
    line-height: 1.2rem;
    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
}
```

```
    details{
        letter-spacing: 2px;
        text-align: center;
        text-indent: 1.2rem;
    }

    details[open] summary{
        margin-bottom: 10px;
    }
</style>
</head>
<body>
    <h1>Noise Pollution Monitoring system</h1><hr>
    <div id="Noise">
        <p id="NoiseTitle" style="display: inline;">Noise Level:</p>
        <p id="value" style="display: inline;"></p><br>
        <p id="warning"></p>
    </div>
</div>
   <a href="/monitoring">Start Monitoring</a>
   <details>
    <summary>Click Here To Know The Ways To Reduce Noise Pollution</summary><br>
```

Reducing noise pollution requires a multi-faceted approach. First, implementing and enforcing noise regulations in urban areas is essential, restricting noisy activities during nighttime hours. Second, encouraging the use of quieter technology and vehicles, such as electric cars and noise-reducing construction equipment, can make a significant difference. Third, promoting public awareness about noise pollution and its effects can lead to more considerate behavior. Fourth, soundproofing buildings and urban planning that separates

industrial and residential zones can help mitigate noise. Fifth, planting trees and creating green spaces in urban areas can act as natural sound barriers. Lastly, using noise barriers along highways and railways can reduce noise levels for nearby residents.

</details>

<script src="app.js"></script>

</body>

</html>

JAVA SCRIPT Code :

```
const express=require("express")

const app=express();

app.set('view engine','ejs');

var admin = require("firebase-admin");

const path = require("path");

app.set('views',path.join(__dirname+'/views'))


const firebaseConfig = {

  "type": "service_account",

  "project_id": "noise-pollution-monitering",

  "private_key_id": "0eb420a3f3d71492a36e7cc9de39f4c8bfa68b9f",

  "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCH3JzxUE/YyIe
i\nqbA0YNQCK3QfTHx9dPnIXxKGcn0le19geFkqTePWm09VIYhpUAJwTwLnP9NUnQih
\n4nF+K5LLUS35/QjMt0wqTOQN1XV4eCkb8JiGWEDMr95/VYqkuG2s5oR4h5M9XW2v
\nqodYhqfZ4iawMa+B2q81mt1Q70aj8rl+k81nQ26z6zVBCE25Zo8YrxqcLVpB06VP\n+kar
Y3xGSwUjNE7Eo6v4afAL3FyT968ad19hZQXoRkmExR+xC498gz8QErsfthMo\nXJN8oq5
TuDuH3egQpxy2RCi4MGX5qmT3ZIsBOs4+kjPFnnQNGxn7qccFF+BEj6qp\nxrI5dKMnA
gMBAAECggEACRpP7Z8q6mRtKgnig2/lZIvSyPIP+TvoLHOqQxXSqrs8\nZbNKjhoP5OY
dB6AqipvQvgOdNy9Ako++71HDtudme3bh9LjuaQDy+naSZRbjOwIa\nBwQ0w/OPy+cQq
```

klp86Vju8zqHUVAKUFkQvvS+6xx/Jg5bV622qVzTc0lfPG7MurM\nYuyuiXBWPe/xc5/IDcr/sAQdNBb3eo/G+cjWeV5njbJZ/LWdpWLzVNTaRBXLFiGK\nlme2Xgh+f9m2lbF75CQNJ5i7U0AzQHoOzSEzhyotZHRa7VL0QLKW45BqI9pXfZ4d\nbcZbjvuq8R+qFjkGLPgnBrFB3a9JJTFXRLn36dDiAQKBgQC8NcLH+M69OumypcxV\nKllgyVF5+wlfBUbmfCtrfRzxTfEp/3gM9kErl6LJrrz4h3TUdDDWpw3/DeySMfMG\nUPh9s6heUVQL9xj2ppBM3SkgWPQEW7a52ncvUxNkbng24hpbHG3zarVkQ+MnJQI/\ntGwoXYSxy1LzYpLqpvF2GUo4JwKBgQC4y/1rc4kPFpU48GYGV7h/7j9qKhRwrrA4\nLW16cn6w+f0s/aEvRC0sJUzUGsXFpjbnepctgGWSLCJr27Ul0LcrdSQquuy2dxGj\nlGVWDSgEaRY5DosEQVT4+v0hFnHaUxxsvYid7ETSHfqYYbJk2nFC9PBJszOXNemZ\ntA3srIIdAQKBgDxn3M4lsuTVBbCKuhwTbYA83OOTiJxwqyLKc30aOLHR3DcogTVX\ny+7byimE5a22e+68I/igwUM9CtKZKXC7iCWABefPnnQAqIhxSRsCWHLDWf4UGX9o\n+Ju9xBmVwwuKYf8gDsHzW6iEiWFE5YXguF30NeSCZ2sqFhEt542J7GI9AoGAGwhk\nOvq8uwNPsvOfyR+98qD0j+A1+0Hir2Ud1cK3+8WmHpW/pX1wqjuOoJyF/+LPt078\nnnIBi214vbt3GAxEkKmxJbSLJC+whHW/Q3ySvjO2efZw+A9JWztFQhC2XXBu6VcHo\nIUn1y1LtXKs2AIDf/q58FBvt2Rne4UgwnUWjBwECgYB5c07XS065nY9mrUEZZjPK\nCBtbtEAkkx3uEkiGr58YWb3WhH8ks2fnvOdpCBUnz/0vbLcBV3vzgc4814K5bRVA\nEc8XJOEbiPdfcHiYYYqj1X/ZFHAhK49tRdwfT2pNcXRqRpHWaao0eh2ag8eBNY12\n6NxpmIZ1S5mNMUcnHtLLww==\n-----END PRIVATE KEY-----\n",

"client_email": "firebase-adminsdk-defb1@noise-pollution-monitering.iam.gserviceaccount.com",

"client_id": "108808938227631267877",

"auth_uri": "https://accounts.google.com/o/oauth2/auth",

"token_uri": "https://oauth2.googleapis.com/token",

"auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",

"client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-defb1%40noise-pollution-monitering.iam.gserviceaccount.com",

"universe_domain": "googleapis.com"

};

admin.initializeApp({

credential: admin.credential.cert(firebaseConfig),

databaseURL: "https://noise-pollution-monitering-default-rtdb.firebaseio.com"

});

```
// Reference to the databas
const db = admin.database();
var data1
setInterval(()=>{let ref = db.ref('sensor_data/NoiseLevelDB');

ref.once('value',(snapshot) => {
  data1 = snapshot.val()
})
.catch((error) => {
  console.error('Error retrieving data:', error);
});

},1500)

app.get('/monitoring',(req,res)=>{
  res.render('index.ejs',{value:data1});
})

app.use(express.static(__dirname,""))


app.listen(3000,()=>{
  console.log('from 8080');
})
//End of this code
```

ABOUT :

This Node.js web app integrates with Firebase to retrieve and display real-time noise level data. It uses Express.js for server setup and serves an EJS template to show the data to users when they access the /monitoring route. The application periodically retrieves data from Firebase for up-to-date information.

EJS code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Noisy_Places</title>
    <link rel="icon" href="../bi.jpg">
    <style>
        Body
{
    text-align: center;
    background-image: url('../bi.jpg');
    background-repeat:inherit;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
```

```
}
h1{
    font-family:monospace;
    font-size: 50px;
    color: aliceblue;
}
#Noise{
    color: aliceblue;
    font-size: 30px;
    margin-top: 100px;
}
#NoiseTitle{
    font-weight: bold;
}
#NoiseTitle,#value{
    font-size: 40px;
    color: aliceblue;
}
#value{
    font-weight: 900;
    letter-spacing: 3px;
}
#warning{
    color: aliceblue;
    font-weight: bold;
    font-size: 2em;
```

```
}
a{
    font-size: 20px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
     color: #000;
     text-decoration: none;
     background-color: aliceblue;
     padding: 5px 10px;
     border-radius: 7px;
     width: 20%;
     display: inline-block;
    margin: 80 auto;
    margin-top: 100px;
    transition: 0.2s cubic-bezier(0.075, 0.82, 0.165, 1);
}
a:hover{
    transform: scale(1.05);
    box-shadow: #000 3px 3px;
}
button{
    font-size: 20px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
    border-radius: 7px;
    border: none;
    padding: 7px 0px;
```

```
    text-align: center;
    margin-top: 90px;
   height: 100%;
   width: 20%;
   }
button:hover{
    cursor: pointer;
}
   </style>
</head>
<body>
    <h1>Noise Pollution Monitoring system</h1><hr>
    <div id="Noise">
        <p id="NoiseTitle" style="display: inline;">Noise Level:</p>
        <p id="value" style="display: inline;"> <%= value %> </p><br>
        <p id="warning"></p>
    </div>
    <a href="/">Stop Monitoring</a>

    <script>
        const label2=document.querySelector('#value');
        const label3=document.querySelector('#warning');
        let data;
        setInterval(()=>{
        data =label2.textContent ;
        data=parseFloat(data);
```

```
        console.log(data);
        if (data <= 65.00) {
        label2.style.color = '#32ff77';
        label3.innerText = "Good Range";
        label3.style.color = '#32ff77';
        } else if (data >= 90.00) {
        label2.style.color = '#ff0000';
        label3.innerText = "High Noise Pollution";
        label3.style.color = '#ff0000';
        } else if (data > 65.00 && data < 90.00) {
        label2.style.color = '#ffc062';
        label3.innerText = "Average Range";
        label3.style.color = '#ffc062';
     }},100)
    setTimeout('location.reload(true)',5000)
    </script>
</body></html>
```

ABOUT :

This is an EJS file that's part of our web app for monitoring noise pollution. It's got a simple and clean design. Here's a quick rundown:

Background:

We've set a nice background image for the page to make it visually appealing.

Content:

You'll see the current noise level displayed right in the center of the page. We've styled it to be big and bold.

Styling:

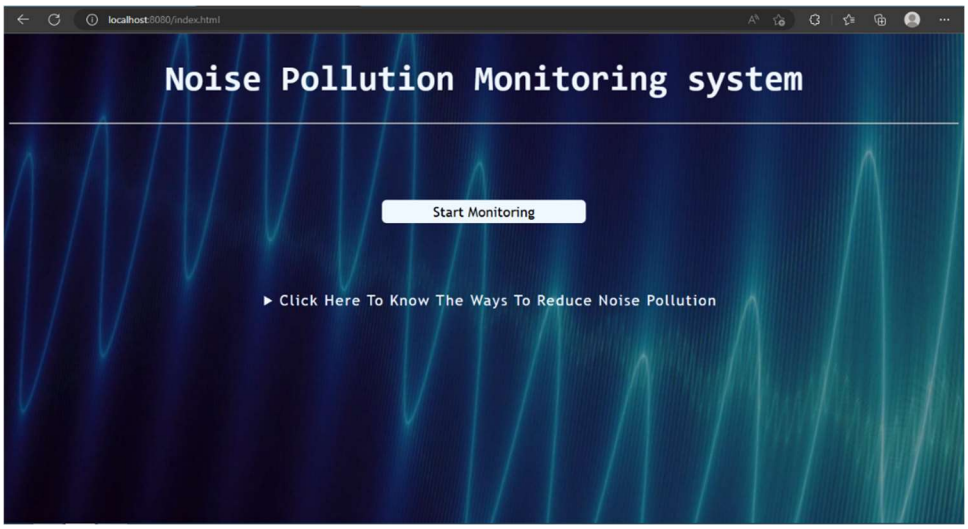
We've used CSS to make sure everything looks good. The noise level changes its text color based on how noisy it is, and there are clear warnings like "Good Range" or "High Noise Pollution" based on predefined thresholds.

User Interaction:

There's a button that lets users stop monitoring, and the page refreshes automatically every 5 seconds to keep the data up-to-date.

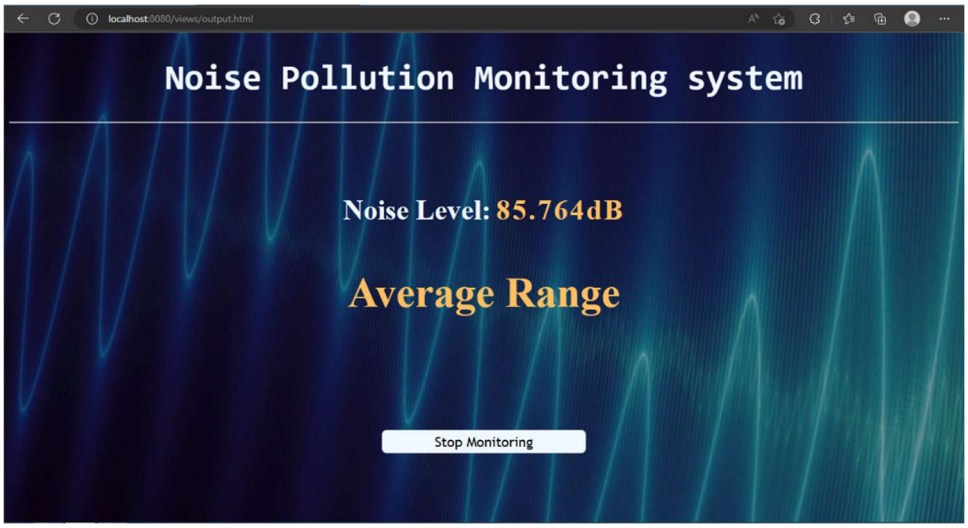
It's a simple and effective way to visualize noise pollution data in real-time.

OUR WEB PAGE



Front Page of our web page

With Summary



Output Result

CONCLUSION:

Innovating to raise awareness about noise pollution, our project combines an IoT sensor simulation, a user-friendly MIT App Inventor mobile app, and a web interface powered by HTML, CSS, JavaScript, and EJS. It provides real-time noise data, enabling users to make informed decisions regarding noise levels. The mobile app simplifies monitoring, while the web interface offers in-depth data visualization. We store data in Firebase for seamless data management, and Wokwi is utilized for simulation. Our system represents a crucial step toward understanding noise levels, promoting healthier environments, and empowering individuals and communities to take action.

THANK YOU