**Noise Pollution Monitoring**

**Abstract:**

Noise pollution is a growing concern in urban and industrial areas, affecting the well-being of communities. To address this issue, the Noise Pollution Monitoring project aims to develop an IoT-based system for monitoring and analyzing noise levels in various locations. This system collects data from noise sensors, processes it in a central platform, and presents real-time information to both the public and relevant authorities. This document provides an in-depth exploration of the project's objectives, IoT device setup, platform development, code implementation, and more.

**Objectives:**

The primary objectives of the Noise Pollution Monitoring project are as follows:

**Real-time Noise Monitoring**: To provide real-time noise level data from multiple locations, allowing citizens and authorities to be aware of their surrounding noise environment.

**Data Analysis:** To process and analyze the collected data to identify trends, patterns, and sources of noise pollution.

**Alerting and Notifications:** To send alerts and notifications when noise levels exceed predefined thresholds or unusual noise patterns are detected.

**Data Accessibility:** To make the noise data accessible to the public, local authorities, and urban planners for informed decision-making.

**IoT Device Setup:**

**Noise Sensors:**

The core of the IoT device setup is the noise sensor. These sensors are strategically placed in various locations where noise monitoring is required. They are equipped with the following features:

**Sound Level Measurement:** Noise sensors are capable of measuring sound levels in decibels (dB) with a high degree of accuracy.

**Wireless Connectivity:** The sensors are equipped with wireless communication capabilities, such as Wi-Fi or LoRa, to transmit data to the central platform.

**Microcontroller:**

Each noise sensor is connected to a microcontroller (e.g., Arduino or Raspberry Pi). The microcontroller plays a vital role in the IoT device setup:

**Data Collection:** It collects data from the noise sensors at regular intervals, ensuring the integrity of the data collected.

**Data Transmission:** The microcontroller manages the transmission of data to the central platform, employing secure communication protocols.

**Power Management:** The microcontroller can manage power consumption efficiently, potentially enabling battery or solar-powered solutions.

**Connectivity:**

The data collected by the noise sensors is transmitted to the central platform via different connectivity options:

**Wi-Fi:** In urban environments, Wi-Fi connectivity can be utilized to transmit data from the sensors to the platform.

**LoRa (Long-Range):** In rural or remote areas, where Wi-Fi is not available, LoRa technology can provide long-range communication capabilities.

**Cellular Networks:** In cases where wide coverage is required, sensors can be connected to cellular networks.

**Power Supply:**

Power supply options vary depending on the deployment location:

**Battery-Powered:** In remote or temporary monitoring setups, the IoT devices can be powered by batteries with efficient power management.

**Solar-Powered:** For sustainable and long-term solutions, solar panels can be used to charge the devices.

**Local Power Supply:** In urban areas, devices can be directly connected to the local power grid.

**Data Storage:**

The data collected by the IoT devices is transmitted to a central storage system, which can either be cloud-based or a local database:

**Cloud-Based:** Cloud solutions offer scalability and accessibility, ensuring that data is securely stored and easily accessible from anywhere.

**Local Database:** In cases where cloud infrastructure is not suitable, a local database may be employed for data storage.

**Platform Development:**

The development of the data-sharing platform is a critical component of the Noise Pollution Monitoring project. The platform provides the following functionalities:

**Data Collection:**

**Device Registration:** IoT devices need to be registered with the platform to establish their identity and connection.

**Authentication and Authorization:** Secure methods of authentication and authorization are implemented to ensure that only authorized devices can send data to the platform.

**Data Ingestion:** APIs and endpoints are created to receive data from the IoT devices.

**Data Storage:**

**Database Design:** The database is designed to efficiently store noise data, with metadata such as location, timestamp, and sensor ID being associated with each data point.

**Data Security:** Data encryption and access control mechanisms are implemented to protect the integrity and confidentiality of the data.

**Data Analysis:**

**Real-Time Analysis:** Algorithms for real-time data analysis are developed to identify current noise levels and detect patterns in the data.

**Historical Analysis:** The platform can perform historical data analysis to determine trends, peak noise levels, and potential sources of noise pollution.

**Anomaly Detection:** To identify unusual noise patterns and trigger alerts, anomaly detection algorithms are implemented.

**User Interface:**

The platform offers a user-friendly web or mobile application for different user groups, including citizens, administrators, and local authorities:

**Real-Time Data Display:** Users can view real-time noise level data, displayed in an intuitive and easily understandable format, such as charts and graphs.

**Geospatial Visualization:** Users can explore noise data on maps, allowing them to pinpoint specific locations and patterns.

**User Notifications:** Alerts and notifications are provided to users when noise levels exceed predefined thresholds or unusual patterns are detected.

**Alerts and Notifications:**

**Threshold Alerts:** Users can set predefined noise level thresholds, and the platform sends alerts when these thresholds are exceeded.

**Custom Notifications:** Users can receive notifications via email, SMS, or mobile app notifications, with the ability to customize notification preferences.

**Code Implementation:**

The successful implementation of the Noise Pollution Monitoring project relies on well-structured code for both the IoT devices and the platform:

**IoT Device Code:**

**Sensor Configuration:** The code configures the noise sensors, ensuring accurate noise level measurements.

**Data Transmission:** Communication protocols like MQTT or HTTP are implemented for efficient data transmission to the platform.

**Power Management:** The code optimizes power usage, especially for battery-powered or solar-powered IoT devices.

**Platform Code:**

#include <LiquidCrystal.h> // include the LiquidCrystal library

const int micPin1 = A0; // define the pin for the first microphone

const int micPin2 = A1; // define the pin for the second microphone

const int micPin3 = A2; // define the pin for the third microphone

const int buzzerPin = 9; // define the pin for the buzzer

const int ledPin = 6; // define the pin for the LED

const int contrast = 50; // define the LCD contrast

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // initialize the LCD display

void setup() {

  pinMode(buzzerPin, OUTPUT); // set the buzzer pin as output

  pinMode(ledPin, OUTPUT); // set the LED pin as output

  lcd.begin(16, 2); // initialize the LCD display

  analogWrite(6,contrast); // set the LCD contrast

**Serial**.begin(9600); // initialize the serial monitor

}

void loop() {

  // read the values from the microphones

  int micValue1 = analogRead(micPin1);

  int micValue2 = analogRead(micPin2);

  int micValue3 = analogRead(micPin3);

// calculate the sound levels in dB for each microphone

  float voltage1 = micValue1 \* 5.0 / 1024.0; // convert the first microphone value to voltage (5V reference)

  float voltage2 = micValue2 \* 5.0 / 1024.0; // convert the second microphone value to voltage (5V reference)

  float voltage3 = micValue3 \* 5.0 / 1024.0; // convert the third microphone value to voltage (5V reference)

  float dB1 = 20 \* log10(voltage1/0.0063); // calculate the sound level in dB for the first microphone

  float dB2 = 20 \* log10(voltage2/0.0063); // calculate the sound level in dB for the second microphone

  float dB3 = 20 \* log10(voltage3/0.0063); // calculate the sound level in dB for the third microphone

  // calculate the average sound level in dB for all microphones

  float averageDB = (dB1 + dB2 + dB3) / 3;

  // display the sound level on the LCD display and the serial monitor

  lcd.setCursor(0, 0); // set the cursor to the first row of the LCD display

  lcd.print("Sound Level: "); // print the text "Sound Level: " on the LCD display

  lcd.setCursor(0, 1); // set the cursor to the second row of the LCD display

  lcd.print(averageDB); // print the average sound level on the LCD display

**Serial**.print("Sound Level: "); // print the text "Sound Level: " on the serial monitor

**Serial**.println(averageDB); // print the average sound level on the serial monitor

  // control the LED and the buzzer based on the sound level

  if (averageDB > 70) { // if the sound level is higher than 70 dB

    digitalWrite(ledPin, HIGH); // turn the LED on

    tone(buzzerPin, 1000, 500); // turn the buzzer on

  } else { // if the sound level is lower than 70 dB

    digitalWrite;

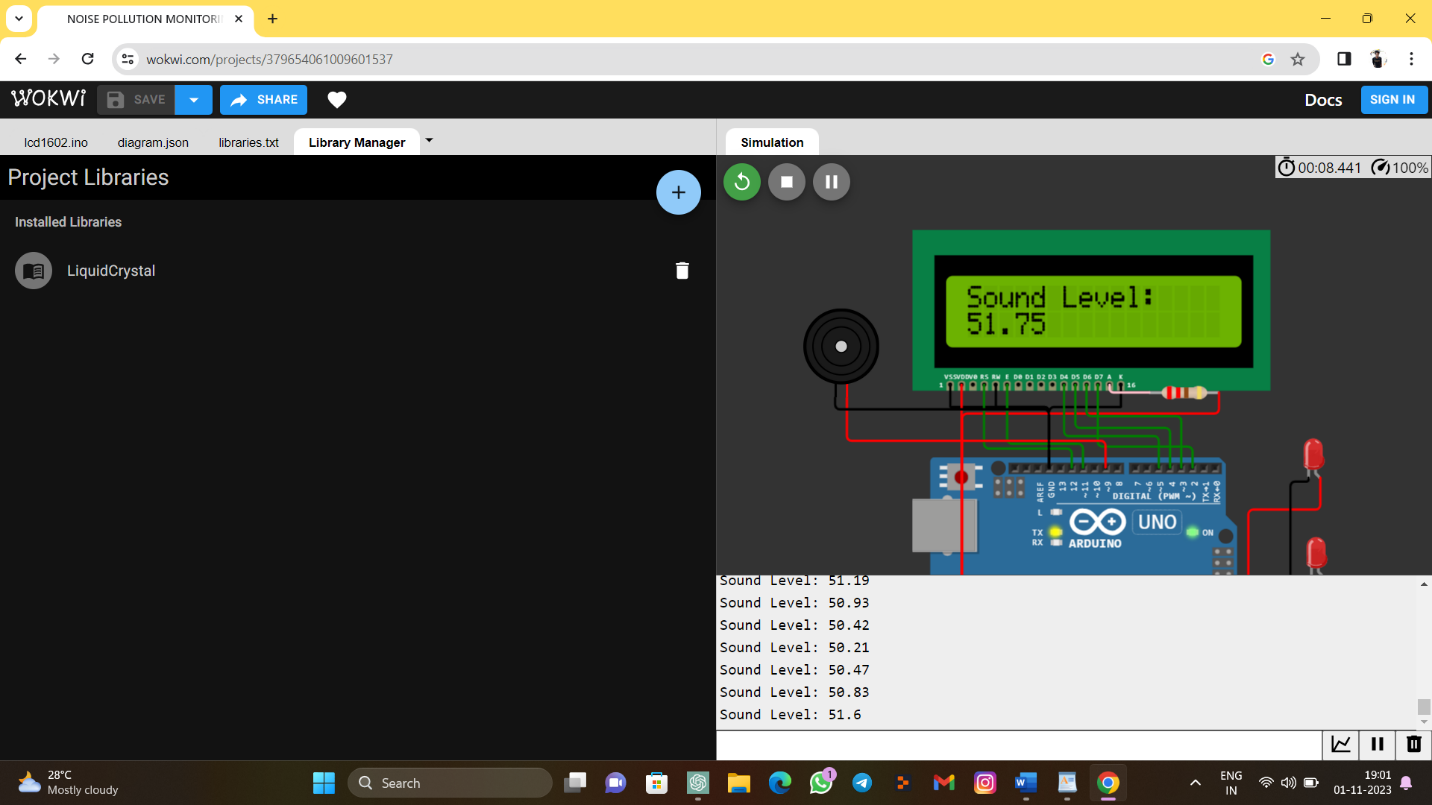
  }

}

**Diagrams and Screenshots:**

A computer screen shot of a computer

Description automatically generated



{

  "version": 1,

  "author": "Uri Shaked",

  "editor": "wokwi",

  "parts": [

    { "type": "wokwi-arduino-uno", "id": "uno", "top": 200, "left": 20, "attrs": {} },

    { "type": "wokwi-lcd1602", "id": "lcd", "top": 8, "left": 20, "attrs": {} },

    { "type": "wokwi-resistor", "id": "r1", "top": 140, "left": 220, "attrs": { "value": "220" } },

    {

      "type": "wokwi-buzzer",

      "id": "bz1",

      "top": 66.16,

      "left": -72.28,

      "attrs": { "volume": "0.1" }

    },

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": 175.61,

      "left": 339.36,

      "attrs": { "color": "red" }

    },

    {

      "type": "wokwi-led",

      "id": "led2",

      "top": 259.31,

      "left": 341.33,

      "attrs": { "color": "red" }

    },

    {

      "type": "wokwi-led",

      "id": "led3",

      "top": 329.23,

      "left": 345.27,

      "attrs": { "color": "red" }

    }

  ],

  "connections": [

    [ "uno:GND.1", "lcd:VSS", "black", [ "v-51", "\*", "h0", "v18" ] ],

    [ "uno:GND.1", "lcd:K", "black", [ "v-51", "\*", "h0", "v18" ] ],

    [ "uno:GND.1", "lcd:RW", "black", [ "v-51", "\*", "h0", "v18" ] ],

    [ "uno:5V", "lcd:VDD", "red", [ "v16", "h-16" ] ],

    [ "uno:5V", "r1:2", "red", [ "v16", "h-118", "v-244", "h50" ] ],

    [ "r1:1", "lcd:A", "pink", [] ],

    [ "uno:12", "lcd:RS", "green", [ "v-16", "\*", "h0", "v20" ] ],

    [ "uno:11", "lcd:E", "green", [ "v-20", "\*", "h0", "v20" ] ],

    [ "lcd:D4", "uno:5", "green", [ "v43.53", "h76.86" ] ],

    [ "lcd:D5", "uno:4", "green", [ "v36.63", "h75.24" ] ],

    [ "lcd:D6", "uno:3", "green", [ "v26.79", "h78.54" ] ],

    [ "lcd:D7", "uno:2", "green", [ "v52.39", "h79.87" ] ],

    [ "bz1:2", "uno:9", "red", [ "v36.28", "h220.75" ] ],

    [ "bz1:1", "uno:GND.1", "black", [ "v9.69", "h180.53", "v54.16" ] ],

    [ "led1:A", "uno:A0", "red", [ "v26.92", "h-60.72", "v166.42", "h-77.79" ] ],

    [ "led2:A", "uno:A1", "red", [ "v18.06", "h-50.87", "v108.32", "h-81.73" ] ],

    [ "led3:A", "uno:A2", "red", [ "v67.3", "h-125.71" ] ],

    [ "led3:C", "uno:GND.2", "black", [ "v47.6", "h-169.87" ] ],

    [ "led2:C", "uno:GND.2", "black", [ "v32.84", "h-17.24", "v86.66", "h-147.71" ] ],

    [ "led1:C", "uno:GND.2", "black", [ "v3.29", "h-15.27", "v198.92", "h-150.66" ] ]

  ],

  "dependencies": {}

}

**Conclusion:**

The Noise Pollution Monitoring project is a comprehensive effort to address the issue of noise pollution in urban and industrial areas. By combining IoT devices for real-time data collection with a sophisticated data-sharing platform, the project enables citizens and authorities to make informed decisions about noise pollution mitigation. The project's objectives, IoT device setup, platform development, and code implementation ensure that it provides a powerful and user-friendly tool for noise pollution monitoring.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Reg No** | **Name** | **NM ID** |
| 1. | 2021504305 | GUNALAN S | 962A49B13521634B4384D1D50691ABA5 |
| 2. | 2021504309 | RAVI R | 432DC798E64F0C5AEFA469F90743D2D0 |
| 3. | 2021504703 | SALMAN FARSI A | D4B74181B73528093CC41C9799D0AE0A |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*