NOISE POLLUTION MONITORING

TEAM MEMBER
822121106002:S.AJAY
PHASE 1 : DOCUMENT SUBMISSION

PROJECT DEFINITION:

This project involves installing IoT sensors in various public areas. These sensors are designed to measure and record noise levels in their respective locations. The primary goal is to provide real time noise level data. The recorded data will be available to the public through a mobile app or any platform. Each sensor is capable of detecting and measuring sound and noise levels in its vicinity. These sound meters are connected to Arduino boards, which act as microcontrollers to gather data from the sensors. These data are uploaded to the cloud and are analyzed for noise pollution. This analysis involves processing and interpreting the raw sound level measurements to determine the degree of noise pollution in the area. The results of the data analysis, which represent noise pollution levels, are made available through a mobile app. Users can access this app to check the noise levels in different public areas.

Design Thinking:

Hardware setup:

1. Select components:
   a. Choose IoT sensors with sound level measurement like decibelmeter.
   b. Acquire arduino board and ESP8266 module.
   c. Gather power supplies, cables and protective cover for the hardware
2. Connect sound sensor to arduino:
   a. Connect the sensors to the Arduino board.
   b. Write arduino code to read sound level from the sensors and transmit this data to the ESP8266 module.

3. ESP8266 integration:
    a. Connect the ESP8266 module to connect to wifi networks.
    b. Write Arduino code to establish a wifi connection and sensor data to a cloud server.

Cloud Setup:
1. Choosing microsoft azure or google cloud for data storage and analysis.
2. Set up an account and create a new project or environment.
3. Create a cloud based database to store the incoming noise level data.
4. Design a data ingestion system that can receive data from the ESP8266 module and store it in the cloud.
5. Writing scripts or functions to analyze the data stored in the cloud.
6. Calculate noise pollution level based on collected sound data.

Mobile app development:
1. Create the app's user interface. Design screens to display noise pollution data in real-time.
2. Connect the app to the cloud service where the noise data is stored.
3. Write code to fetch real time noise pollution data from the cloud and display it.
4. Ensure the app works correctly and debug any issues.

Deployment and accessibility:
1. Install the IoT sensors in the selected public areas. Ensuring that they are securely mounted and protected from environmental factors.
2. Publish the mobile app on app stores to make it accessible to the public.
3. Ensuring that the data collected and analyzed in the cloud is accessible to the public through the mobile app or web platform.
4. Regularly monitor the system's performance and update the app as needed.