**PROJECT TITLE : MONITORING NOISE POLLUTION –IOT**

**PHASE 2: INNOVATION**

## PROBLEM STATEMENT

Deploy IoT sensors to measure noise pollution in public areas, providing real-time noise level data accessible to the public through a platform or mobile app.Sound meter is connected with Arduino board to detect the sound and noise in the locality.This data is fed to cloud using esp8266 module. This data is analyzed for noise pollutionlevels and the result is made available in a mobile app.

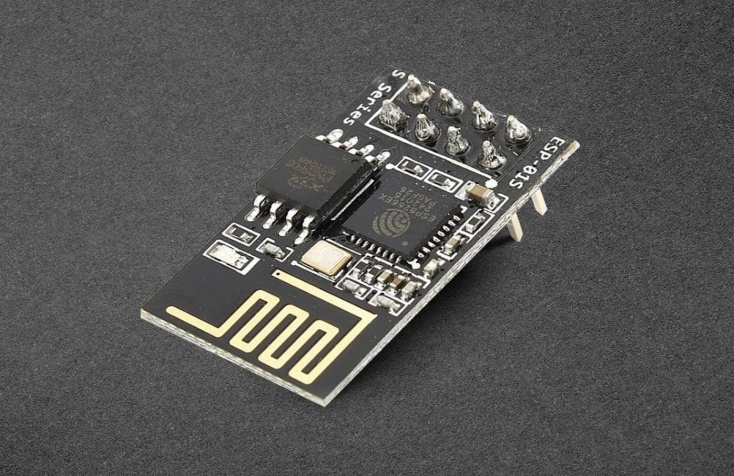
## COMPONENTS REQUIRED:

**Noise Sensors** : Noise sensors capable of measuring sound levels. The LM393 sound detecting sensors is used.

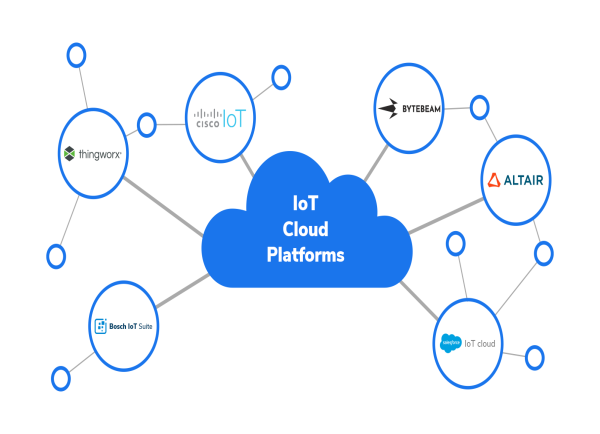
**IoT Hardware:** To connect the sensors to the internet, you'll need microcontroller boards or IoT development kits.



**Data Communication**: It is a way to transmit the data from the sensors to a central server or a cloud platform. ESP8266 WiFi module is used



**Cloud Platform**: A cloud is used platform to receive and store data from the sensors



**Power Supply**: we may need a power source, such as batteries, solar panels, or a power outlet and also a USB adapter.



## STEPS FOR PROJECT DESIGN:

1. **Sensor Setup:**

**Connect the noise sensors to the microcontroller. The specific connections will depend on the sensor you choose. Ensure that you can accurately measure and record noise levels.**

1. **Microcontroller Programming:**

**Write code for the microcontroller to collect data from the sensors and send it to the cloud platform. You'll also need to handle data processing, error handling, and power management.**

1. **Data Transmission:**

**Use Wi-Fi, cellular, or LoRa to send the data to the cloud platform. Ensure that the data is encrypted for security.**

1. **Cloud Platform Configuration:**

**Set up your cloud platform to receive and store the incoming data. You may also need to create a database to store historical noise data.**

1. **Data Analysis:**

**Implement data analysis tools to process and visualize the noise data. You can create dashboards to monitor real-time noise levels and generate reports or alerts when certain thresholds are exceeded.**

1. **User Interface:**

**Develop a user interface that allows users to access noise level data, historical records, and other relevant information. This can be a web or mobile app.**

1. **Power Management:**

**If you're using batteries or solar panels, ensure efficient power management to prolong the system's life.**

1. **Deployment:**

**Install the sensors at the desired monitoring locations. Ensure they are securely mounted and protected from environmental factors.**

1. **Maintenance:**

**Regularly maintain and calibrate the sensors to ensure accurate data collection. Update the software as needed**

**10.Data Visualization and Reporting:**

**Provide data visualization and reporting options, such as charts, maps, and alerts for users and local authorities to make informed decisions.**