**PROJECT 4 : NOISE POLLUTION MONITORING SYSTEM**

**PHASE 1 : Problem definition :**

The project involves deploying IOT sensors to measure noise pollution in public areas and providing real-time noise level data accessible to the public through a platform or mobile app.

**Key Objectives :**

* **Data Collection:** To collect and record noise pollution data from multiple sensors deployed in different locations, providing a comprehensive view of noise levels within an area.
* **Environmental Protection:** Monitor noise levels in sensitive natural areas to protect ecosystems , wildlife and human health from noise pollution.
* **Threshold Alarms**: Set predefined noise level thresholds and trigger alarms or notifications when noise levels exceed these limit**.**
* **Noise Mapping**: Create noise maps to visualize noise distribution and identify high-noise areas.
* **Mitigation Planning**: Develop strategies and policies for noise pollution reduction and mitigation.
* **Noise Control Measures**: Evaluate the effectiveness of noise control measures and technologies.

Design Thinking :

* **Understand Users:**

Identify who will use the system and understand their needs.

* **Define the Problem:**

Clearly state the issues related to noise pollution to be addressed.

* **Generate Ideas:**

Brainstorm creative solutions, considering both tech and non-tech options.

* **Build a Prototype:**

Create a basic version of the monitoring system for testing.

* **Test in Real Environment:**

Implement the prototype in a real-world setting for feedback.

* **Refine and Iterate:**

Improve the design based on user feedback and testing.

* **Develop Final Version:**

Create the complete noise pollution monitoring system.

* **Evaluate Impact:**

Assess how well the system addresses noise pollution.

* **Ensure Sustainability**:

Plan for ongoing support and maintenance.

* **Communicate Benefits:**

Clearly communicate the system's advantages to the public and authorities.

**PHASE 2 : INNOVATION :**

After thorough research and analysis, we arrived at an innovative solution to solve the above problem as detailed in phase 1 of our project. We will be using the ESP8266 micro controller as well as Arduino UNO microcontroller as both these suit the best for our project.

Sensors**:**

**Microphone sensor (LM 393):** Employing IOT sensors is crucial for accurate data collection and analysis. Consider using sound level meters or microphones (LM393 sensor) as primary sensors, coupled with a microcontroller such as Arduino for signal processing and data transmission. These devices can capture real-time audio data, measuring noise levels in decibels.

Connectivity **:**

Ensure connectivity through a robust network infrastructure, integrating sensors with IOT devices. Utilize Wi-Fi, cellular, or LPWAN technologies for data transmission to a central server for real-time monitoring and analysis.

Cloud :

Beeceptor is used for rapid API prototyping, endpoint validation, and simulation of diverse data scenarios. While beneficial for testing, it complements, rather than replaces, dedicated cloud platforms that provide robust infrastructure for data storage, analysis, and real-time processing.

Protocol **:**

Implementation using MQTT (Message Queuing Telemetry Transport) protocol for efficient data transmission. This lightweight and reliable protocol ensure seamless connectivity, making it ideal for monitoring and managing noise levels effectively.

Public platform :

Design an easy-to-use interface for seamless navigation. Encourage user participation by enabling data contributions and insights sharing on social media. Educate the public about noise pollution's effects through informative resources. Support innovation with API access for developers to create applications using the data. Ensure broad accessibility through web browsers and mobile apps for widespread public awareness.

TEAM MEMBERS:

1. Jayanth N - (2021504523)
2. Lavanya V - (2021504525)
3. Mathana K - (2021504528)
4. Sudhanthira P - (2021504548)
5. Vidhya SS - (2021504557)