URBAN NOISE POLLUTION MANAGEMENT

Syntax Error Squad

Jaison Edward B

Sharvesh S

Sanjith M

Objective

Create a platform to monitor and reduce noise pollution in urban areas, using sensors and data analysis to identify noise hotspots and implement mitigation strategies.

Reasons to Solve the Problem

- **Health Impacts:** Prolonged exposure to high levels of noise pollution can lead to various health problems such as stress, sleep disturbances, hearing loss, and cardiovascular issues.

- **Quality of Life:** Excessive noise negatively affects the daily lives of residents by reducing comfort in homes, workplaces, and public spaces.

- **Regulatory Compliance:** Many cities have noise regulations, but without real-time data, enforcement is challenging.

- **Informed Urban Planning:** Data-driven insights into noise levels can help urban planners design better infrastructure, zoning regulations, and traffic management strategies to reduce noise pollution at its source.

- By solving this problem, we can create healthier, more liveable urban environments while providing city officials and planners with the tools they need to make informed decisions about noise mitigation.

Proposed Idealogy
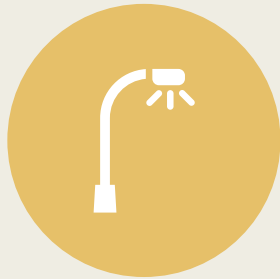
We will be developing a smart, scalable platform to monitor and manage urban noise pollution using a network of noise detection stations.
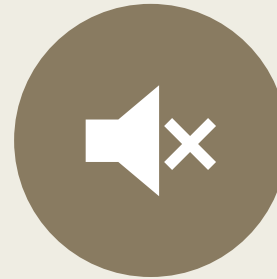
Each station, built using ESP32 microcontrollers and sound sensors, will capture real-time sound data from different locations across the city.

This data will be sent to Firebase, where it will be stored and processed.

Our Raspberry Pi-based web dashboard will retrieve this data, displaying it on the web dashboard.
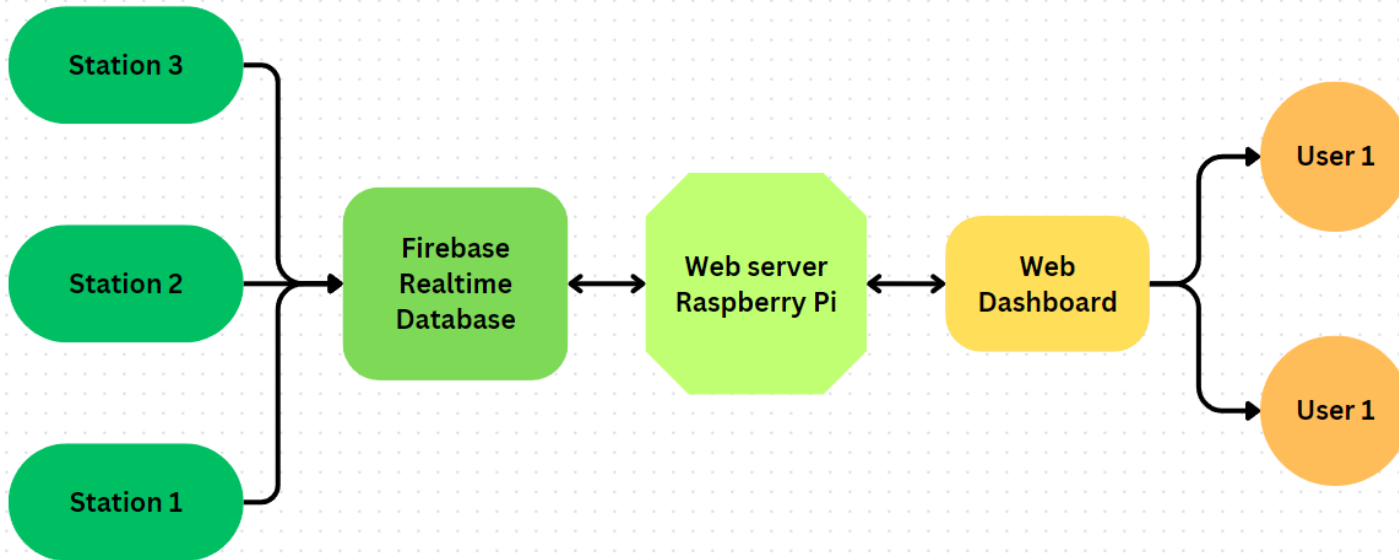
Components and Software to be used

Hardware Components

– *ESP32 DEV boards*

– *LM393 based Sound Sensors*

– *Raspberry Pi 4B*

Software and Languages

– *Arduino IDE, C/C++*

– *Apache Web Server, HTML, CSS, Java Script*

– *Firebase*

- Noise Detection Station that use ESP32s and Noise sensors (Hardware)

- Data Storage and Transmission using Firebase real-time database.

- Raspberry Pi will be hosting the Web Server.

- Data Visualization & Analysis will be made possible through the web dashboard.

- With this approach, we will create a dynamic and scalable platform that will help monitor, visualize, and manage noise pollution in urban areas, making it easier for city planners to take action

Individual Noise Station Block Diagram

- Each noise station will consist of an ESP32 microcontroller

- One or two LM393 sound sensors will be connected to each ESP32 module

- The sound sensors will continuously monitor ambient noise levels in real-time, measuring the intensity of noise in decibels (dB) at the station's location.

- Multiple noise stations can be deployed across different urban locations, each functioning as an independent unit
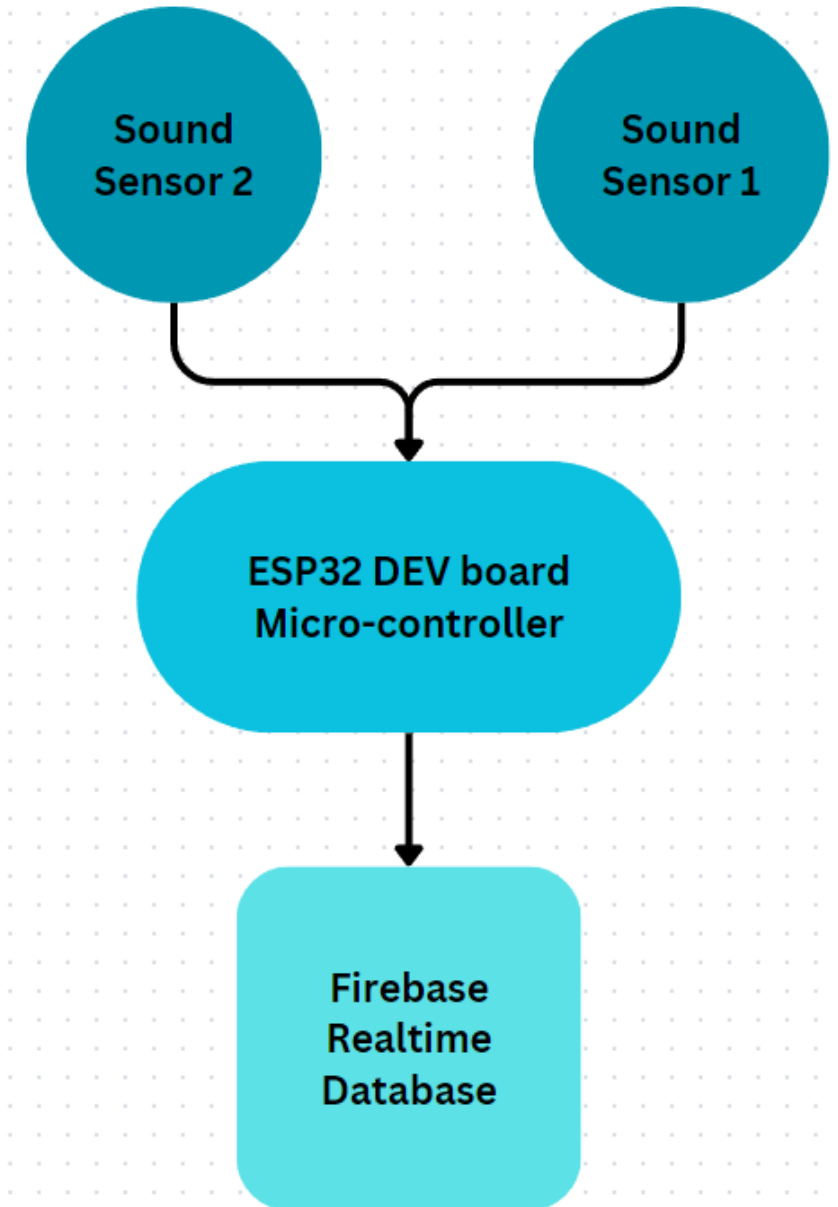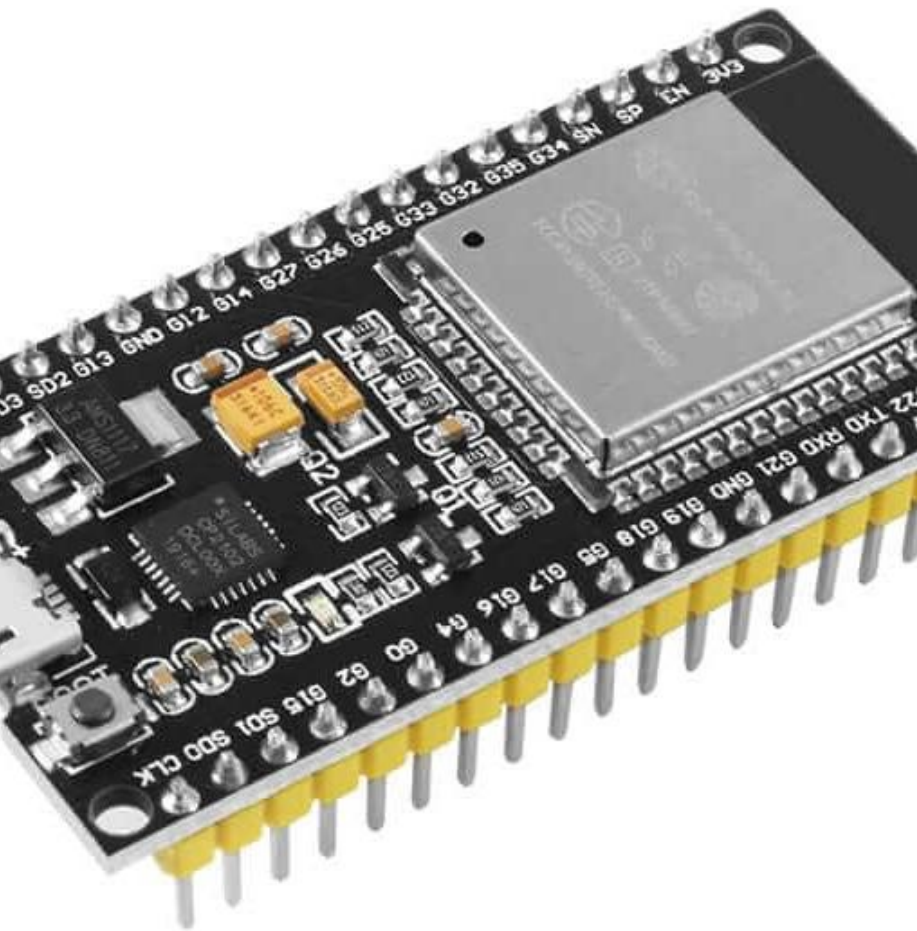
ESP32 (Detection module) Pseudo Code

1. **Setup:**
   - Initialize peripherals.
   - Connect to Wi-Fi and Authenticate with Database.

2. **Main Loop:**
   - Every 1 second:
       - Read noise levels from both sensors (calcDecibels()).
       - Process levels (LOW, MEDIUM, HIGH) using processLevel().
       - Print and display values on OLED (displayOled()).
   - Every 20 seconds:
       - Send sensor data (sen1, sen2) to Firebase.

3. **Helper Functions:**
   - calcDecibels(pin): Read sensor, calculate and return decibel.
   - processLevel(dB): Return noise level (0=LOW, 1=MEDIUM, 2=HIGH).
   - displayOled(sen1, sen2, lvl1, lvl2): Update OLED with sensor values.

Raspberry Pi (Web server) Pseudo Code

1. **Setup**
   - Import Firebase and initialize with config
   - Connect to database

2. **Function classifyNoise(value):**
   - Return noise level

3. **Function fetchData():**
   - Get data from Firebase
   - Display noise levels and draw graph

4. **Function drawNoiseGraph(sensor1, sensor2):**
   - Draw bar graph

   - Call fetchData()

Business Model

Value Proposition:

Innovative urban noise pollution management through smart technologies and real-time data analytics, enhancing public health and urban livability.

Key Products & Services:

- **IoT-Based Noise Monitoring:** Real-time tracking and alerts.
- **Consulting Services:** Custom noise mitigation strategies.
- **Business Solutions:** Noise audits and certifications.
- **Smart Products (Future):** Noise-canceling technologies.

Target Market:

- City Governments
- Real Estate Developers
- Noise-Sensitive Businesses
- Environmental Agencies
- Urban Residents

Revenue Streams:

- **Subscription Model (SaaS):** Tiered access to data and analytics.

- **Consulting Fees:** Noise audits and strategies.

- **Installation Services:** Sensor setup and maintenance.

- **Data Licensing:** Anonymized data sales.

- **Noise Certification Program:** Annual fees for compliance.

Key Activities:

- **R&D:** Continuous improvement of technology.

- **Platform Maintenance:** Ensure efficient operations.

- **Customer Support:** Ongoing assistance for clients.

- **Business Development:** Building partnerships and promoting services.

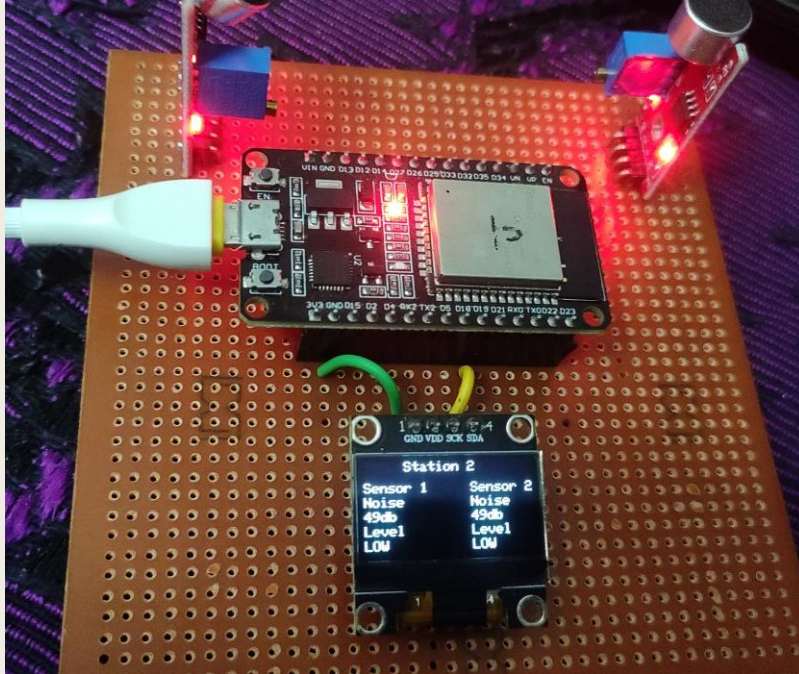- **Regulatory Compliance:** Aligning with local noise regulations.
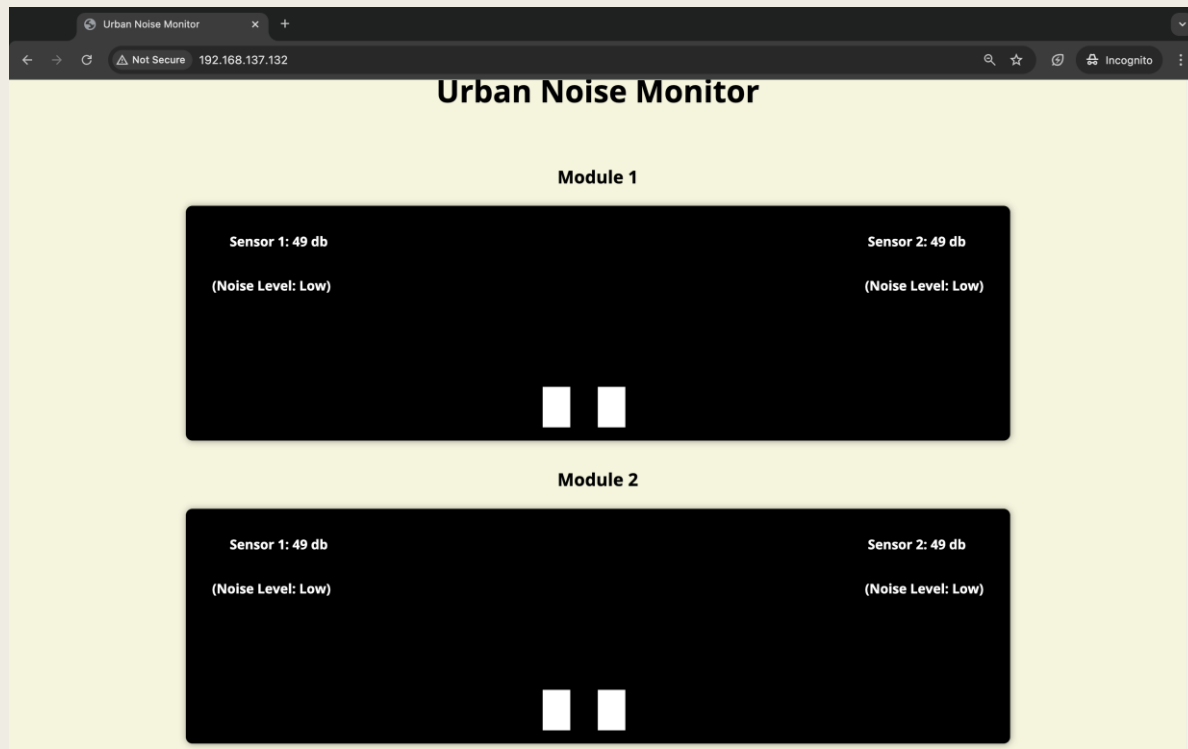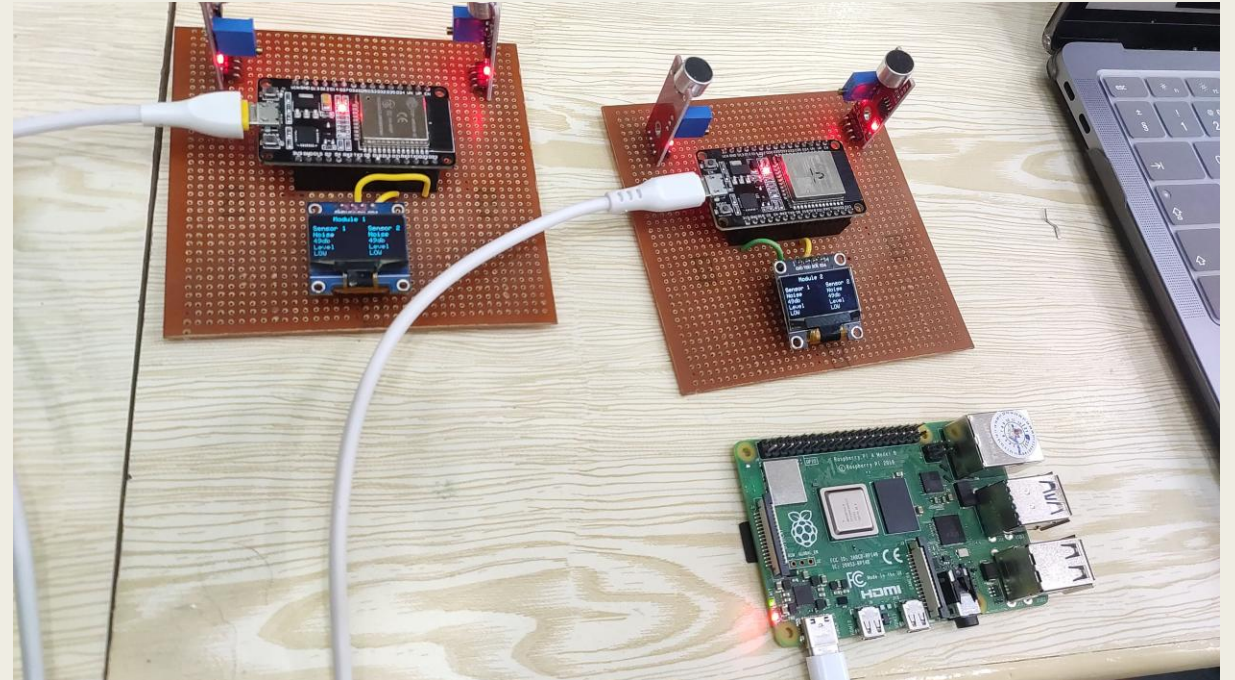
Web dashboard hosted on Raspberry Pi
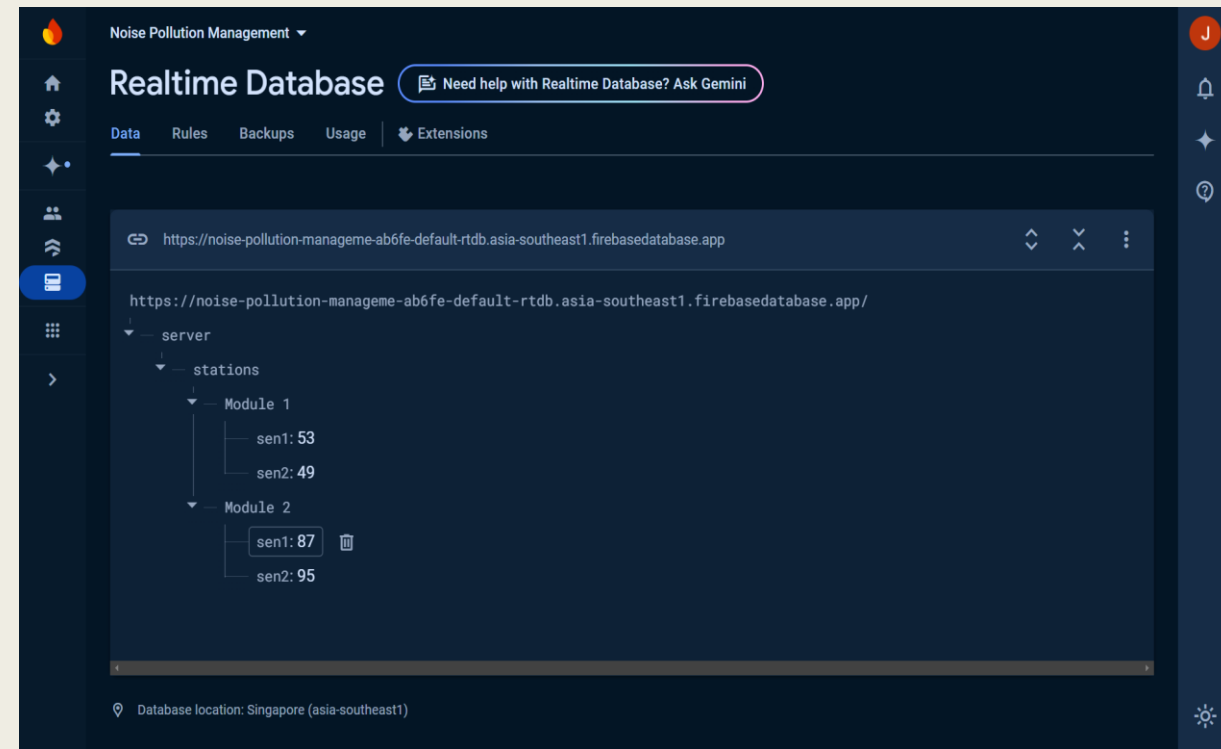
Firebase real-time DB at work

RESULT

Individual noise detection module

Entire hardware of the system

RESULT