Project Title: Air & Noise Pollution Monitoring Dashboard

Objective:

To design a web-based application that visualizes real-time air and noise pollution levels across different city zones using map overlays. The system will allow citizens to report pollution sources and enable authorities to monitor trends, take action, and publish alerts.

Problem Statement:

Urban areas face increasing challenges from air and noise pollution, yet real-time public access to this data is limited. People remain unaware of health risks in their localities, while authorities struggle with scattered reports and manual monitoring. There is a need for an intelligent system that empowers both citizens and officials through real-time, location-aware data.

Proposed Solution:

- Displays AQI and noise levels on an interactive map (using OpenStreetMap or Google Maps API).
- Allows users to:
  - View current pollution levels (color-coded)
  - Report pollution incidents (with photos, auto-location, and descriptions)
  - Receive alerts based on location
- Allows admin/staff to:
  - Update or validate reports
  - Access trends and heatmaps
  - View maintenance or inspection logs
  - Analyze data through zone-based analytics dashboard

Key Features:

Module	Description
User Module	Register/login, view AQI/Noise data, report pollution, view alerts, give feedback
Admin Module	Login, manage pollution reports, validate user inputs, view historical logs, send alerts
Map Integration	Live AQI/Noise data visualization, location tagging for reports and sensors
Reporting	Upload images, auto-capture GPS location, timestamp, select pollution category
Dashboard	Real-time charts, city-zone breakdowns, user complaints, trend analysis

Tech Stack:

- **Frontend: HTML,CSS, Bootstrap.**
- **Backed**: Java REST APIs
- **Database**: MySQL
- **Authentication**: JWT / Spring Security
- **Optional**: Google Maps API, Firebase for image storage

Impact:

- Enhances public awareness about health risks from pollution.
- Encourages civic participation in maintaining cleaner environments.
- Helps urban planners and pollution control boards take timely actions.
- Enables data-driven policies and alerts in high-risk areas.

Software Requirements Specification (SRS)

Title: Air & Noise Pollution Monitoring Dashboard

1. Introduction

1.1 Purpose

The purpose of this project is to develop a user-friendly dashboard that monitors and visualizes air and noise pollution levels in a given area. The system aims to raise awareness about environmental quality by providing historical data on key pollutants and noise levels. This can help individuals, communities, and authorities make informed decisions to ensure a healthier and more sustainable environment.

1.2 Scope

1. Data Collection:

   Using either publicly available APIs or sample datasets to gather air and noise pollution data

   Focused on selected pollutants (e.g., PM2.5, PM10, CO2) and sound levels in decibels.

2. Data Processing & Display:

   Converting raw data into a readable and meaningful format.

   Representing the data using interactive charts, tables, and/or maps.

3. Dashboard Development:

   Creating a web-based interface (using HTML, CSS, and JavaScript or a framework like React).

   Allowing users to view pollution levels over time or by location.

4. Educational Value:

   Helping students understand how environmental data can be used in real-world applications.

   Encouraging awareness and discussion on pollution and its health impacts.

5. Technical Learning:

   Applying frontend and/or backend development skills.

   Working with APIs, data visualization libraries, and basic data handling techniques.

1.3 Definitions, Acronyms, Abbreviations

Term	Description
GPS	Global Position System
UI	User Interface
API	Application Programming Interface
User	General public accessing the system

1.4 References

1. OpenAQ – Open Air Quality Data Platform

https://openaq.org/

2. World Air Quality Index Project – API for Global AQI

https://aqicn.org/api/

3. WHO – Air Pollution Guidelines

World Health Organization. Air quality guidelines – Global update 2021

https://www.who.int/publications/i/item/9789240034228

2. Overall Description

2.1 Product Perspective

This dashboard is a standalone system that integrates with environmental monitoring APIs to visualize air quality and noise pollution levels. It incorporates mapping services (e.g., Google Maps or Leaflet.js) to display pollution data geographically.

2.2 Product Functions

- Interactive, map-based data visualization
- Historical data analysis and trends
- Alert system for hazardous pollution levels
- Feedback and reporting mechanism for users

2.3 User Classes and Characteristics

User Type	Description
1. General Users	View pollution data for awareness or planning. Basic tech skills.
2. Health-Conscious	Monitor pollution levels to make lifestyle or travel decisions.
3.City Planners	Analyze trends, generate reports, and respond to high-risk areas.
4Admins	Configure sensors, manage alerts, and respond to feedback.

2.4 Operating Environment

- Devices: Android/iOS Mobile Devices, Desktop Web Browsers (Chrome, Firefox, Safari)
- Backend: Java with MySQL Database

- Frontend: Web-based dashboard

2.5 Design and Implementation Constraints

- Dependence on external API uptime and sensor data accuracy
- Limited access to proprietary environmental data sources
- Map API usage limitations (e.g., call quotas and licensing)

2.6 User Documentation

- In-app tutorials and tooltips
- Downloadable PDF user manual

3. System Features

3.1 Map-Based Visualization

Functional Requirements:

- Integrate with map services (e.g., Google Maps, Leaflet.js).
- Use geolocation to show user’s current area data.
- Allow zoom, pan, and location search functionality.

3.3 Historical Data and Trends

Functional Requirements:

- Provide charts (line, bar) for daily, weekly, monthly data.
- Support export of data as CSV or PDF.
- Allow filtering by date, location, and pollutant type.

3.4 Alerts and Notifications

Functional Requirements:

- Define customizable thresholds for AQI and noise levels.
- Send alerts to users when levels exceed thresholds.
- Display alert banners on dashboard.

3.5 User Feedback and Reporting System

Functional Requirements:

- Feedback form.
- Store feedback in backend database.
- Admin panel to view, filter, and respond to reports.

4. External Interface Requirements

4.1 User Interface

- The System Will Feature A Responsive Web Interface Built Using Html, Css, And Bootstrap.
- Includes Interactive Pollution Maps Using Google Maps Or Openstreetmap Apis.

4.2 Hardware Interface

- User Devices: Smartphones Or Computers With Internet And Location Services.
- Admin Devices: Pc, Laptop, Or Tablet With Modern Web Browser.

4.3 Software Interface

- Google Maps Api / Openstreetmap For Location Tagging And Pollution Visualization.
- Firebase (Optional) For Cloud Image Storage From Pollution Reports.
- Mysql Database For Storing User Data, Pollution Logs, Reports, And Feedback.

4.4 Communication Interfaces

- All Frontend-Backend Interactions Will Use HTTPS-Based Restful Apis.
- Geo-Location Services Used To Auto-Tag User Reports

5. Appendices

A. Tools and Technologies

- Frontend: React Native / Flutter / React.js
- Backend: Node.js / Django / Flask
- Database: PostgreSQL
- Map: Google Maps / Leaflet
- Hosting: AWS / GCP / Azure

B. Future Enhancements

- AI-powered route suggestions based on historical data
- Offline mode for saved routes
- Integration with ride-sharing services