Air Quality Index (AQI)

Submitted By

Rohan Gohel

24mca020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

SCHOOL OF TECHNOLOGY INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

Certificate

This is to certify that the minor project entitled "Air Quality Index (AQI)" submitted by Student Name 24mca020, towards the partial fulfilment of the requirements for the award of the degree of MCA in Computer Science and Engineering, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached the level required for being accepted for examination. The results embodied in this minor project, to the best of my knowledge, haven't been submitted to any other university or institution for the award of any degree or diploma.

Dr.Devendrasinh vashi

CSE Department,

Institute of Technology,

Nirma University,

Ahmedabad.

Dr Sudeep Tanwar

Professor and Head,

CSE Department,

Institute of Technology,

Nirma University,

Ahmedabad.

BrainyBeam
Info-Tech Pvt. Ltd.
"Better Brains for your Idea"

F6, Dhanlaxmi Chambers, Near Sarvoday Co op bank, Ashram road Ahmedabad 380014

24th Jul-2025

TO WHOM IT MAY CONCERN

This is to certify that, Mr./Ms. **Gohel Rohan (24mca020)** Student of **Institute of Technology, Nirma University - Faculty of Technology & Engineering** (MCA), has successfully completed 60 Days Internship in the field of **Data Science & Machine Learning** during the period of **26th May-2025 to 24th Jul-2025**.
During the period of this internship program with us, student of your College/University exposed to different processes and found sincere and hardworking.

BRAINYBEAM INFO-TECH PVT. LTD.
AHMEDABAD

With Best Regards,
Mr. Sagar Jasani
CEO.
Brainybeam Info-Tech PVT LTD
Ahmedabad (Gujarat-INDIA)

+91 90332 37336 sagar@brainybeaminfotech.com

CONTENTS:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide and faculty at Nirma University for their invaluable guidance and support throug the development of this Air quality index in ML . I am also thankful to my colleagues and enthusiasts who participated in interviews and surveys, offering insights that shaped the features of the system. Their inputs were very useful in refining the platform's functionality. Finally, I extend heartfelt thanks to my family and friends for their constant motivation and belief in my capabilities.

ABSTRACT

AirAware is a modern digital platform built to help people stay informed and proactive about the air they breathe. Think of it as your personal air quality companion—offering real-time AQI updates, forecasts, and health recommendations based on your location. Whether you're planning a morning jog or checking if it's safe for kids to play outside, AirAware makes it easy to understand air quality through simple color-coded ratings and personalized alerts. You can explore pollutant details, track historical trends, and even join community discussions about local environmental conditions. We built AirAware to bridge the gap between complex environmental data and everyday decisions, empowering users to make healthier, smarter choices with every breath.

LIST OF TABLES
Table No Title Page No.
Table 3.3
Data Dictionary
Table A Tools Used

CHAPTER 1 INTRODUCTION

1.1 ABOUT THE COMPANY

INTRODUCTION OF THE COMPANY

BrainyBeam Info-Tech is an innovative web and mobile apps Development Company in India & USA that has the specialization and expertise in developing Web and Mobile applications for organizations in various verticals - Retail, Travel, Construction, Entertainment, Lifestyle, Social Networking and Education etc.

1.2 QUALITY POLICY:

BrainyBeam follows a documented standard software development process including requirement gathering, prototyping, development, testing, delivery, and post-deployment support. They accurately test each part of the solution, conduct code reviews, and enforce software/data security protocols. Non-Disclosure Agreements (NDAs) are used both with clients and employees to ensure confidentiality and data protection.

1.3 COMMUNICATION:

Emphasis on pure transparency in information exchange between the company and clients, including clear discussions on data access, risk factors and authorization responsibilities. They commit to being responsive and punctual, with availability for client meetings at preferred times and regular progress checks to prevent delays.

1.4 RESOURCES:

BrainyBeam is staffed with a team of 60–80+ certified experts, including developers, designers, QA analysts and project managers across technologies such as Java, .NET, AWS, Cloudera, Oracle APEX, Red Hat, Salesforce, and Scrum certifications. They maintain a careful balance of experienced and creative professionals, with backup developers and a structured management system to ensure timely delivery and support.

1.2 THE SYSTEM

1.2.1 DEFINITION OF SYSTEM:

AirSense is a web-based environmental monitoring system which combines real-time AQI tracking, health alerting, and predictive analysis using machine learning. We are using Python 3.13.3, Flask, and MongoDB for the backend stack. For the frontend,. At AirSense, we offer users live updates on air quality levels based on their location, visual data representation through charts, and personalized health recommendations. The system uses historical pollution data to train ML models that can forecast AQI trends and identify pollution hotspots. A core focus is on delivering actionable insights through a user-friendly interface, with support for mobile and desktop. Users can subscribe to daily reports, set custom alert thresholds, and even contribute to a community pollution map. AirSense promotes environmental awareness and personal health safety by making air quality data easy to access, understand, and act on.

1.2.2 PURPOSE AND OBJECTIVES:

The purpose of **AirSense** is to provide a smart, accessible, and data-driven platform for monitoring, analyzing, and predicting air quality in real time. By integrating machine learning with environmental data, AirSense empowers individuals and communities to make informed decisions about their health and daily activities based on local air conditions. The platform is designed to raise awareness about pollution levels and their effects, offering actionable insights through an intuitive web-based interface.

- **To monitor real-time Air Quality Index (AQI)** using publicly available sensor and weather data across different locations.

- **To implement machine learning models** that predict future AQI levels based on historical data and current trends.

- **To alert users** with notifications and health recommendations when AQI crosses harmful thresholds.

- **To provide visual data insights**, including pollutant breakdowns and AQI trend graphs for better understanding and analysis.

- **To allow users to set preferences**, subscribe to daily reports, and receive location-specific alerts through a personalized dashboard.

- **To increase environmental awareness** by involving users in a community-driven pollution reporting and feedback system.

- **To support scalable deployment** that can integrate with multiple data sources and be adapted to different regions or countries.

Present System

The current air quality monitoring systems in place are mostly managed by government environmental agencies and provide static or delayed AQI data through official websites or apps. These systems typically focus on collecting air pollutant levels like PM2.5, PM10, NO_2, SO_2, CO, and O_3 from fixed monitoring stations in urban areas. While the data is accurate, the reach and responsiveness of these systems are limited.

1.2.4 PROPOSED SYSTEM:

o Monitoring Module

▪ Collects real-time AQI data from APIs, sensors, and satellite feeds

▪ Displays AQI levels with color-coded indicators (Good to Hazardous)

▪ Supports pollutant breakdown: PM2.5, PM10, CO, NO_2, SO_2, O_3

▪ Automatically detects user's location for localized air data

o Prediction Module

▪ Uses machine learning (e.g. Random Forest, LSTM) for AQI forecasting

▪ Predicts AQI for next 6–24 hours based on historical and real-time data

▪ Triggers early alerts for upcoming poor air quality

▪ Continuously learns from new data to improve prediction accuracy

o Smart Dashboard

▪ Personalized dashboard with widgets for AQI, forecast, alerts

▪ Mobile-first responsive design for ease of use anywhere

▪ Dark/light theme toggle for accessibility

▪ Supports multi-language and region settings

o Visualization & Analytics Module

▪ Interactive charts showing AQI over time (hourly/daily/monthly)

▪ Pollution heatmaps using geolocation and community reports

▪ Comparative view of AQI by city, region, or custom filters

▪ Historical data explorer with download/export options

1.3 PROJECT PROFILE

1.3.1 PROJECT TITLE:

Air Quality Index (AQI)

1.3.2 SCOPE OF THE PROJECT:

Functional scope

The functional scope defines the core features and capabilities the system will perform:

- Real-time AQI Monitoring: Display current air quality levels based on user's location.
- Pollutant Tracking: Show individual pollutant concentrations like PM2.5, PM10, CO, NO_2, SO_2, and O_3.
- Machine Learning-Based Prediction: Forecast AQI for upcoming hours/days using trained models.
- Custom Alerts & Notifications: Notify users when AQI crosses health-critical thresholds.
- Data Visualization: Display AQI trends through interactive graphs, charts, and maps.
- User Profiles: Allow users to save locations, customize settings, and receive personalized content.
- Community Reporting: Users can report visible pollution or local air quality changes.
- Health Recommendations: Provide guidance and actions based on AQI levels (e.g., wear masks, avoid outdoor exercise)

Non-functional

- Performance: The system should respond to user input and data updates within 2 seconds.

- Usability: Simple and intuitive user interface for both desktop and mobile devices.

- Security: Protect user data using secure login, encrypted data storage, and access control.

- Scalability: Capable of handling increasing users, data sources, and traffic volume without degradation.

- Reliability & Availability: Ensure uptime of 99.5% for continuous monitoring and alerts.

- Maintainability: Code should follow modular and clean architecture to allow easy updates and debugging.• Offline Support (Partial): Cache recent AQI data locally so users can still view recent history without internet.

- Testability: All modules must be unit tested and integration tested for consistency and reliability.

PROJECT TEAM:

This being an individual internship project towards the completion of my college course of study, I have independently managed all phases involved in the development of the Air Quality Index (AQI) monitoring system using machine learning. This includes system analysis, machine learning model integration, backend development, frontend interface design, database setup, and testing. The project was carried out under the guidance of my assigned project supervisor, whose support and constructive feedback played a key role in refining the functionality, usability, and overall implementation of the system. With real-time AQI tracking and predictive analytics, this project reflects both technical learning and practical application, completed within the academic timeline and aligned with real-world environmental concerns.

1.3.4 HARDWARE/SOFTWARE ENVIRONMENT IN COMPANY:

Hardware Environment:

- Processor: Intel Core i5 / i7 or equivalent (Minimum 2.0 GHz, 4-core)
- RAM: 8 GB or higher
- Storage: 256 GB SSD or higher
- Network: Stable internet connection for accessing APIs and deploying the system
- Peripherals: Laptop/Desktop, External Monitor (optional), Mouse, Keyboard

Software Environment:

- **Operating System:** Windows 10 / 11, Ubuntu 20.04+
- **Programming Language:** Python 3.13.3
- **Frameworks & Libraries:**
    - Flask (Web backend)
    - Scikit-learn, TensorFlow or Keras (for ML model development)
    - Pandas, NumPy (for data handling and preprocessing)
    - Matplotlib, Plotly (for visualization)
    - **Database:** MongoDB

Development Tools:

- VS Code / PyCharm (IDE)

CHAPTER 2 SYSTEM ANALYSIS

2.1 FEASIBILITY STUDY

2.1.1 OPERATIONAL FEASIBILITY:

The proposed AQI Monitoring System is highly operationally feasible due to its alignment with real-world environmental concerns, ease of use, and practical utility for end users. The system is designed to deliver real-time air quality updates, personalized alerts, and predictive insights in a user-friendly manner, making it suitable for public use, educational purposes, and health safety awareness.

From an operational standpoint, the system does not require any complex hardware or specialized infrastructure. It can be accessed through standard web browsers on desktop and mobile devices. The backend and machine learning models run efficiently on cloud servers or local machines with moderate configurations. The user interface is designed to be intuitive and responsive, ensuring accessibility for users of all age groups and technical backgrounds.

2.1.2 TECHNICAL FEASIBILITY:

The development and deployment of the Air Quality Index (AQI) Monitoring System is technically feasible with the current availability of technologies, tools, and infrastructure. The system leverages mature and well-documented technologies such as Python, Flask,PostgreSQL, and machine learning libraries like Scikit-learn and TensorFlow, which ensures ease of implementation and reliable performance.

Given the availability of open-source libraries, datasets, and API access to real-time pollution data, along with the technical knowledge required to build and maintain the system, the AQI monitoring project is considered fully technically feasible under current resources and constraints.

2.1.3 FINANCIAL & ECONOMICAL FEASIBILITY:

The Air Quality Index (AQI) Monitoring System is financially and economically feasible, especially for academic, pilot, or small-scale deployment purposes. The project leverages open-source technologies and freely available datasets, which significantly reduces the cost of development. Tools such as Python, Flask, ReactJS, MongoDB/PostgreSQL, and machine learning libraries (Scikit-learn, TensorFlow, etc.) are all open-source, eliminating the need for expensive software licenses.

Data for air quality can be accessed through public APIs provided by government or environmental organizations, reducing the cost of data acquisition. Hosting and deployment can initially be handled through free or low-cost cloud platforms like Heroku, Render, or Netlify. As the system scales, cloud service providers like AWS, Azure, or Google Cloud offer cost-effective plans based on actual usage.

2.1.4 HANDLING INFEASIBLE PROJECTS

In the case where a project like the Air Quality Index (AQI) Monitoring System is found to be infeasible—whether due to technical, financial, operational, or environmental constraints—it is important to follow a structured approach to reassess, scale down, or pivot the project effectively.

If the infeasibility is **technical** (e.g., lack of reliable real-time data sources, integration complexity, or insufficient ML accuracy), the solution would be to simplify the system by removing predictive components and focusing on real-time monitoring using available APIs. Alternative algorithms or third-party services may also be considered

2.2 REQUIREMENT ANALYSIS

2.2.1 FACTS-FINDING TECHNIQUES:

2.2.1.1 INTERVIEW

One of the primary fact-finding methods used for this project was interviews. Informal and semi-structured interviews were conducted with:

- Environmental science students to understand common concerns about air pollution and what kind of information they expect from an AQI system.
- General public users to assess their awareness of AQI and whether they currently use any tools to monitor it.
- Faculty/project supervisor to ensure alignment with academic goals and technical feasibility.

The interviews revealed that most users are unaware of the specifics behind AQI values and would benefit from a simple, color-coded interface with health tips. They expressed interest in features like real-time alerts, forecasted AQI levels, and personalized notifications. These insights helped shape the system's core modules including the predictive engine, alert system, and visualization dashboard.

2.2.1.2 QUESTIONNAIRE

In addition to interviews, a **questionnaire** was prepared and distributed to gather structured input from a broader group of potential users. This method allowed us to quantify user opinions, preferences, and awareness levels regarding air quality monitoring and digital health tools.

The questionnaire was shared with students, working professionals, and individuals from urban areas where air pollution is a growing concern. It included both **closed-ended** (multiple choice, rating scale) and **open-ended** questions.

Sample Questions Included:

1. Are you aware of what the Air Quality Index (AQI) represents?
2. Have you used any mobile or web app to check AQI before?
3. How frequently do you check air quality in your area?
4. Would you be interested in receiving real-time alerts about poor air quality?
5. What features would you expect in an air quality monitoring app?

6. How useful would you find AQI prediction or forecasting for planning your day?
7. Rate your interest in seeing pollutant-wise breakdown (e.g., PM2.5, CO, NO_2) on a scale of 1–5.

2.2.1.3 REVIEW RECORDS

The **review of existing records** was another key fact-finding technique used during the requirement analysis phase of the AQI Monitoring System project. This method involved examining various sources of documented data and existing solutions to better understand the standards, functionalities, and gaps in current air quality monitoring systems.

Documents and Sources Reviewed:

- Government websites such as **CPCB (Central Pollution Control Board)** and **AQICN.org** to understand the standard AQI calculation methodology and pollutant categories.
- • Existing AQI apps (e.g., SAFAR, AQI India, AirVisual) to study their feature sets, user interfaces, and limitations.
- • WHO (World Health Organization) guidelines on air quality and health impact thresholds.

2.2.1.4 OBSERVATION

Observation was employed as a practical fact-finding technique to understand how people interact with existing AQI platforms, their level of awareness about air quality, and how environmental data affects their daily activities. This method involved closely monitoring user behavior in both online and offline settings.

Observation Scenarios:

- Watching how individuals in urban areas check weather and AQI through mobile apps or websites before stepping outdoors.
- Noting the lack of engagement or awareness in communities even during visibly poor air quality days.
- Observing user interaction with government AQI portals—particularly the challenges faced in interpreting data and navigating the interface.
- Tracking responses to pollution-related alerts (if any) provided by platforms like Google Weather or air quality apps.

2.3 CONTEXT DIAGRAM

The context diagram provides a high-level overview of the AQI Monitoring System, representing the system as a single process and showing its interaction with external entities. It outlines the main data inputs and outputs between the system and external actors such as users, environmental data sources, and notification services.

External Entities:

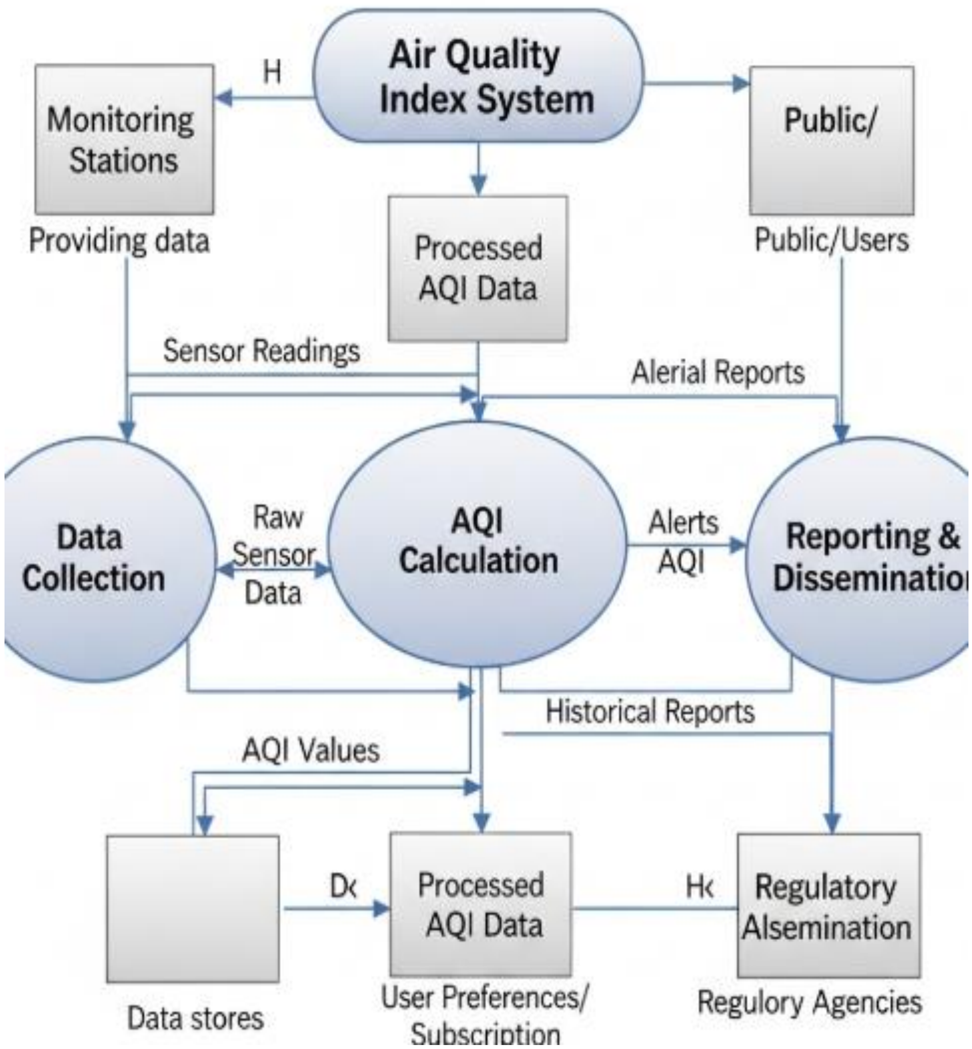
1. **Users**
   - Interact with the system via the web interface
   - View current AQI, forecasts, pollutant data
   - Set preferences for notifications and alerts
   - Receive health recommendations and warnings
2. **Environmental Data Sources (APIs / Sensors)**
   - Provide real-time air quality and weather data
   - Supply pollutant-specific values (PM2.5, PM10, CO, NO_2, etc.)
   - Feed historical data for machine learning model training
3. **Notification System**
   - Sends push, email, or SMS alerts to users based on AQI thresholds
   - Delivers daily AQI summaries and forecast updates

System: AirSense AQI Monitoring System

- Collects and processes real-time air quality data
- Stores and analyzes historical data
- Runs ML models to predict AQI trends
- Displays results in the form of graphs, maps, and health alerts
- Allows user profile management and personalization
- **2.4 DATA FLOW DIAGRAMS**
- **O level and 1 level**

Content Diagn (Level 0)

Level 1



CHAPTER 3 SYSTEM DESIGNS

3.1 SYSTEM FLOW

3.1.1 INTRODUCTION

The system flow diagram and accompanying description serve as a blueprint for the functionality of the Air Quality Index (AQI) monitoring system. This section introduces the logical progression of data and processes, highlighting how the system efficiently collects, processes, and disseminates critical air quality information. The system flow is designed to ensure accuracy, reliability, and timeliness in the calculation and reporting of the AQI. By visually mapping out the sequence of operations, this section aims to provide a clear and comprehensive understanding of the system's architecture and its internal workings. It details how raw data from sensors is transformed into a meaningful and user-friendly AQI value, which is then made accessible to end-users and other relevant parties.

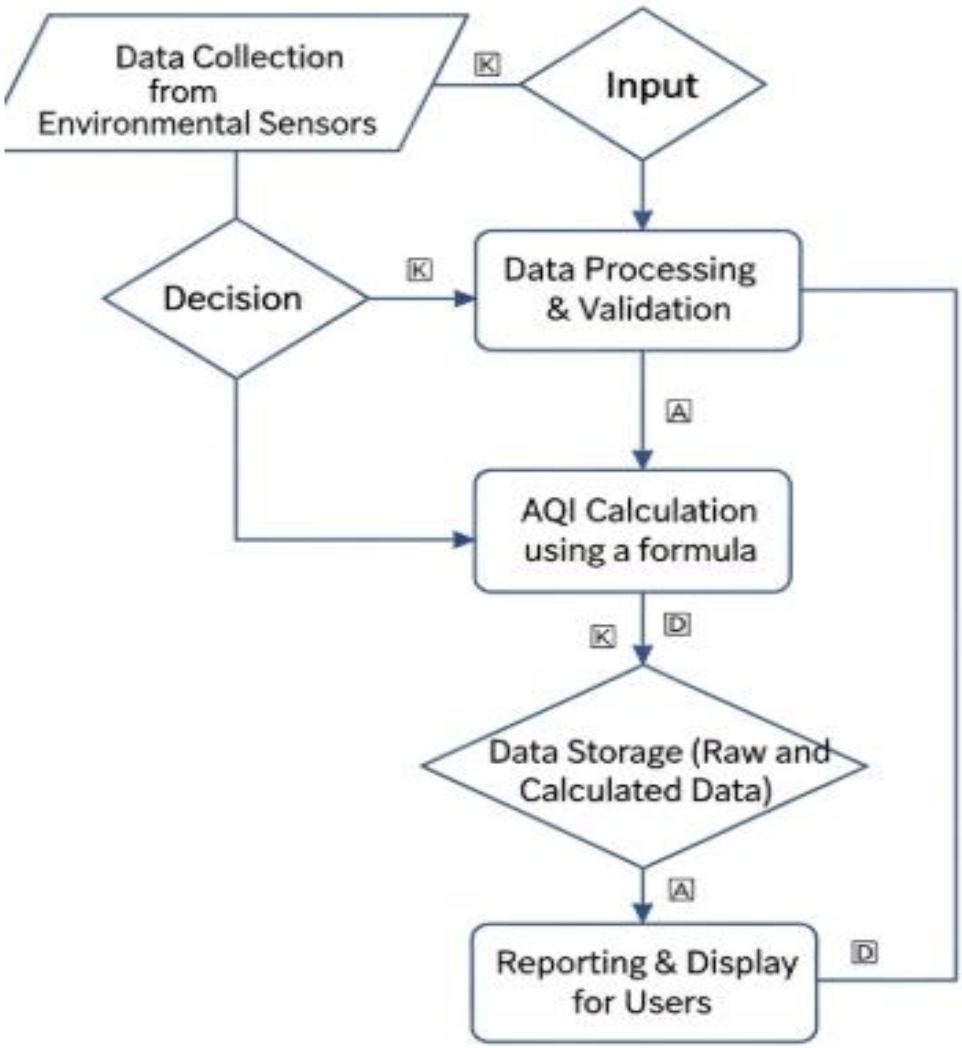
3.1.2 OVERVIEW OF SYSTEM FLOW:

The system flow of the Air Quality Index (AQI) monitoring system can be summarized in the following key stages:

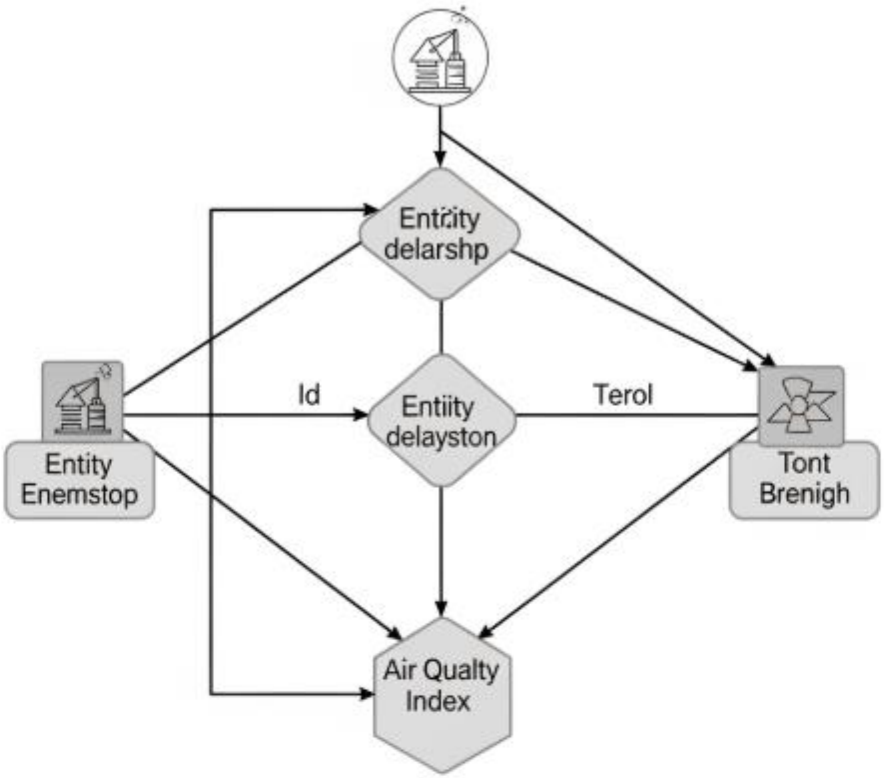
1. **Data Collection:** The process begins with the continuous collection of raw data from environmental sensors. These sensors measure the concentration of various pollutants (e.g., PM2.5, PM10, NO2, SO2) at specific locations. This data is the primary input for the entire system.
2. **Data Processing and Validation:** Upon collection, the raw data is subjected to a validation process. This stage involves filtering out any corrupted or anomalous readings that could skew the results. The data is then formatted and prepared for the calculation phase, ensuring that only accurate and reliable information is used.

3.1.3 SYSTEM FLOW DIAGRAM:

Air Quality Index System Flow



3.2 ENTITY RELATIONSHIP DIAGRAM



3.2.1 KEY ENTITIES:

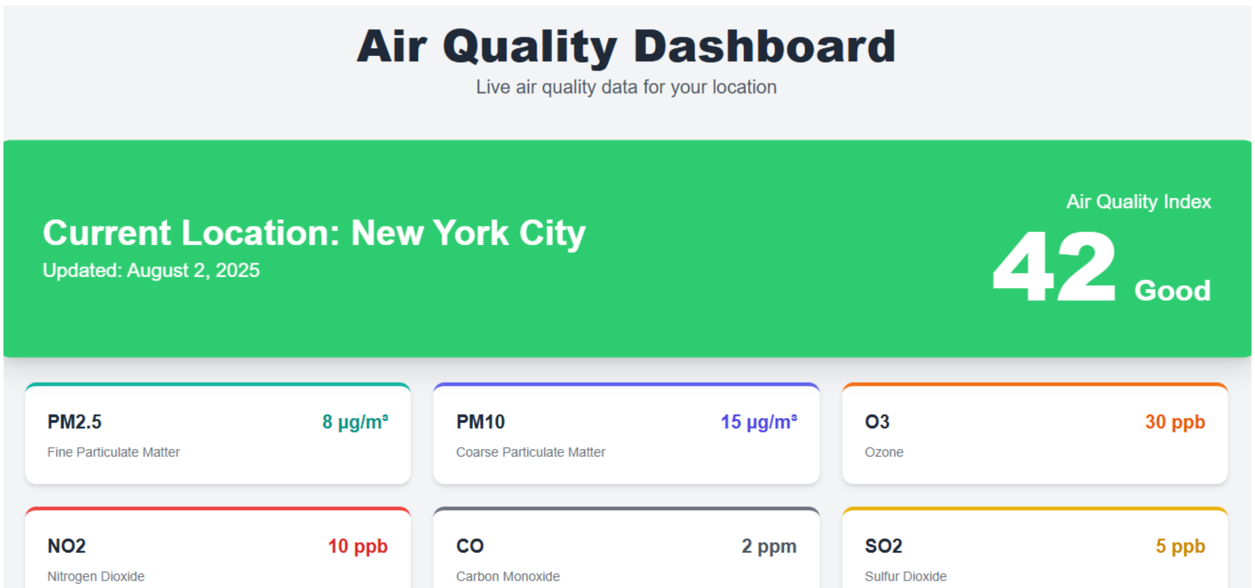
- **Users**: Individuals or organizations accessing air quality data and insights through the platform.
- **Environmental Data Sources**: These include sensors, weather APIs, pollution control boards, and satellite data that provide real-time environmental data like PM2.5, PM10, CO_2, etc.
- **Air Quality Monitoring System**: The core system that collects, processes, and analyzes the environmental data using machine learning models to calculate the Air Quality Index (AQI).
- **Notification System**: A module responsible for alerting users based on AQI thresholds via messages, emails, or app notifications.

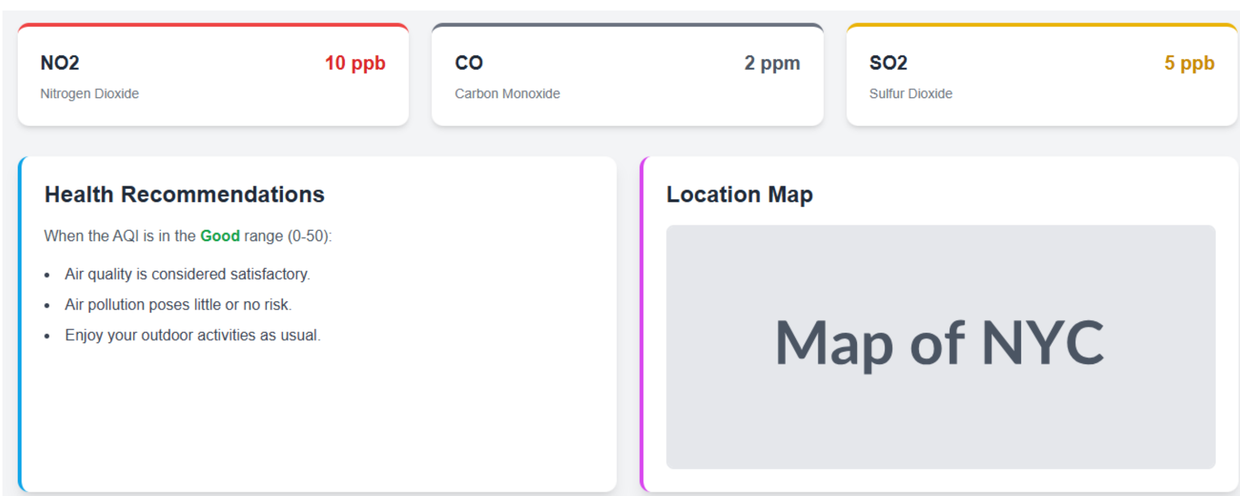
Field Name	Description	Data Type	Size	Constraints
user_id	Unique identifier for each user	Integer	10	Primary Key, Auto Increment
username	Registered username of the user	Varchar	50	Unique, Not Null
email	User email address	Varchar	100	Unique, Not Null
location	User's current or preferred city/location	Varchar	100	Not Null
pollutant_type	Type of air pollutant (e.g., PM2.5, CO2, NO_2)	Varchar	10	Not Null
aqi_value	Air Quality Index value	Integer	3	0–500
aqi_category	AQI health category (e.g., Good, Moderate, Hazardous)	Varchar	20	Computed
timestamp	Date and time of AQI data collected	DateTime	-	Not Null
temperature	Ambient temperature at location	Float	-	Optional
humidity	Relative humidity level	Float	-	Optional
prediction_result	ML model prediction of future AQI	Integer	3	Computed
alert_status	Notification alert status for user (e.g., sent, pending)	Varchar	10	Default: Pending

CHAPTER 4 RESULT & DISCUSSION

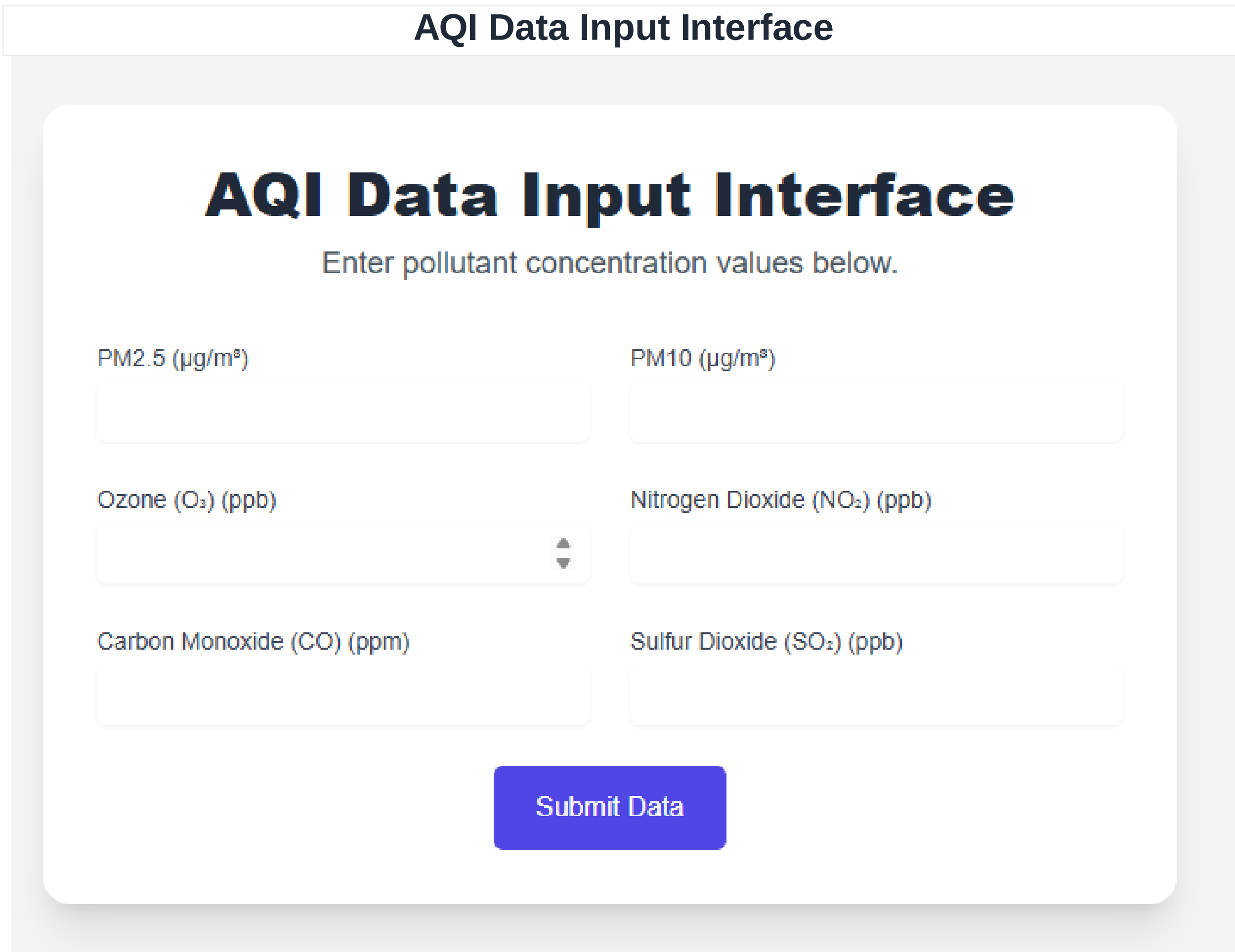
4.1

4.1.1 RESULTS HOMEPAGE:

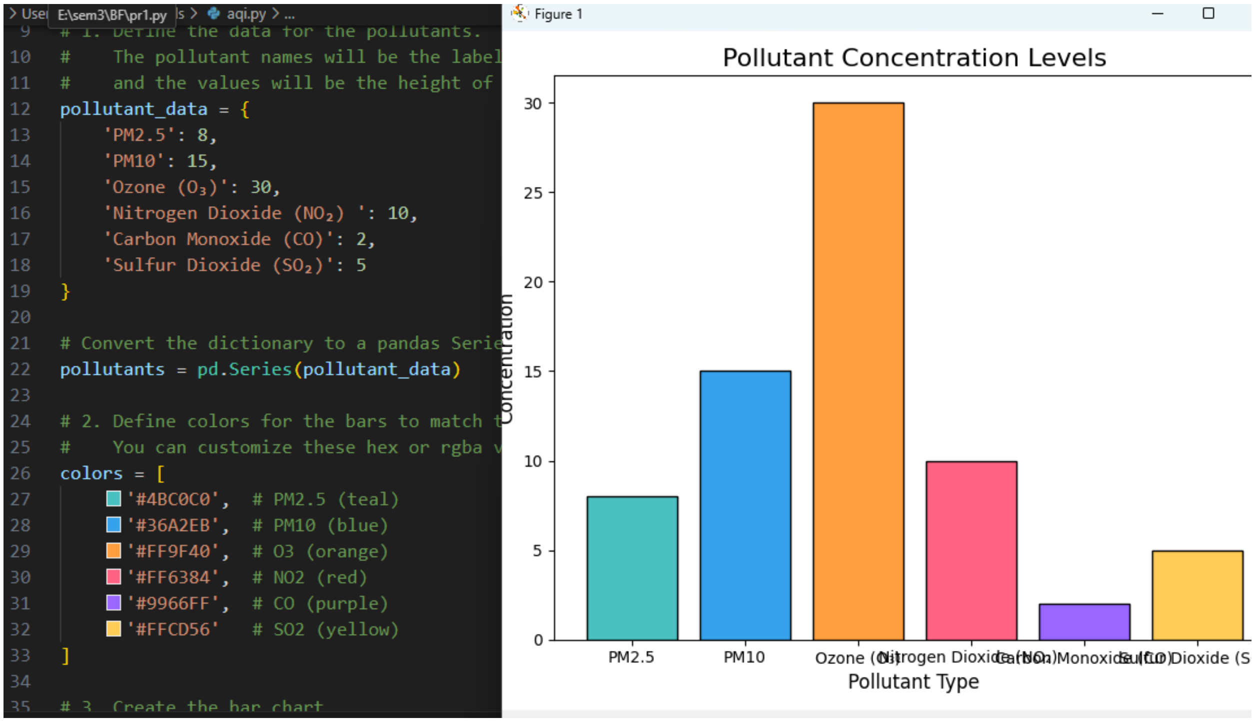




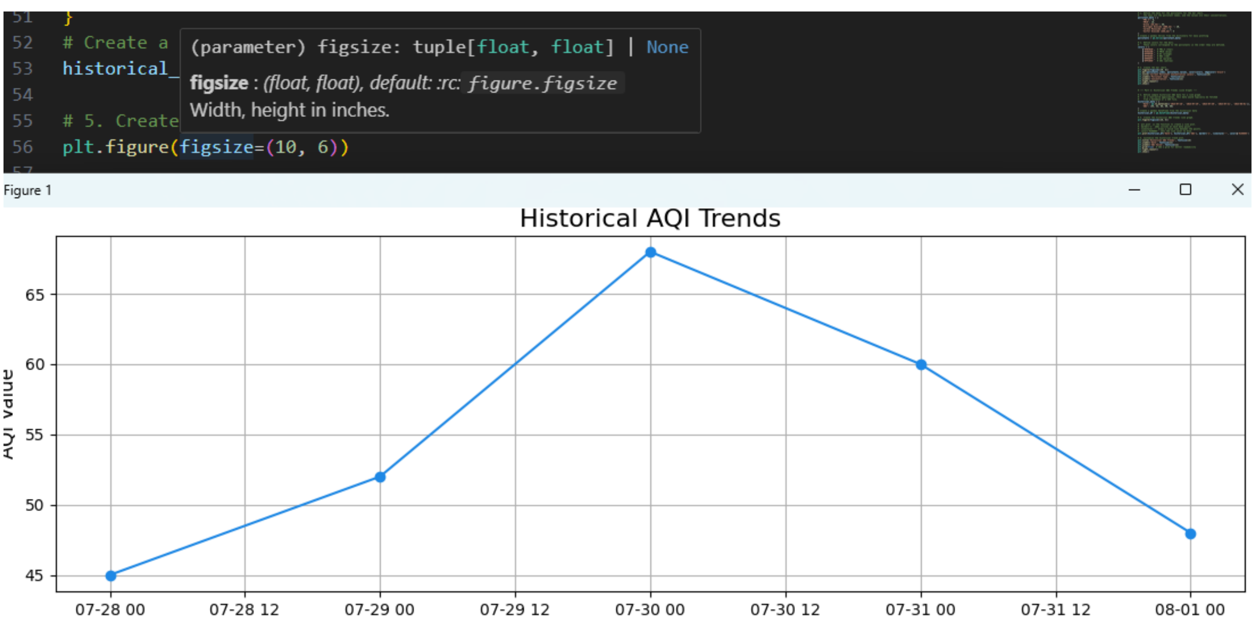
4.1.2 This interface will allow a user to manually input pollutant concentration values, which can then be used to calculate an AQI value.



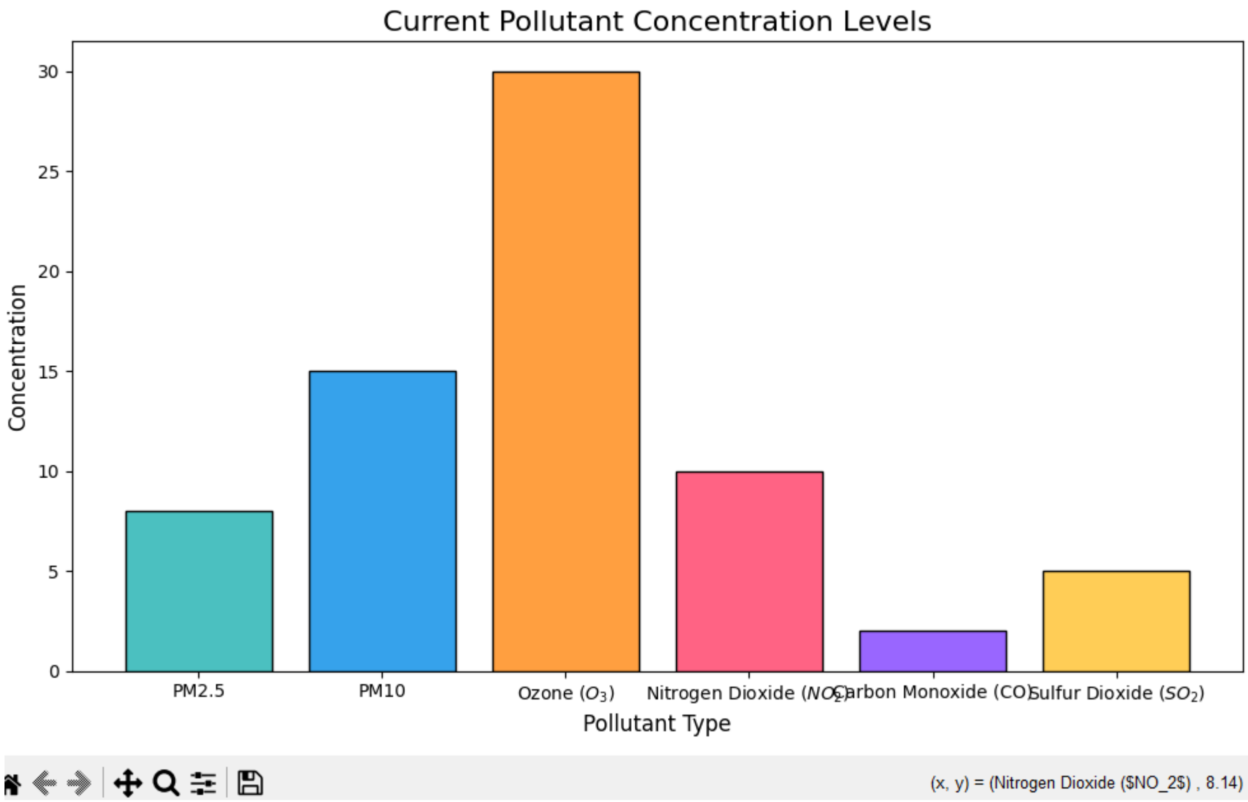
4.1.3AQI Visualization graph (Real-time)



4.1.4 aqi historical trends

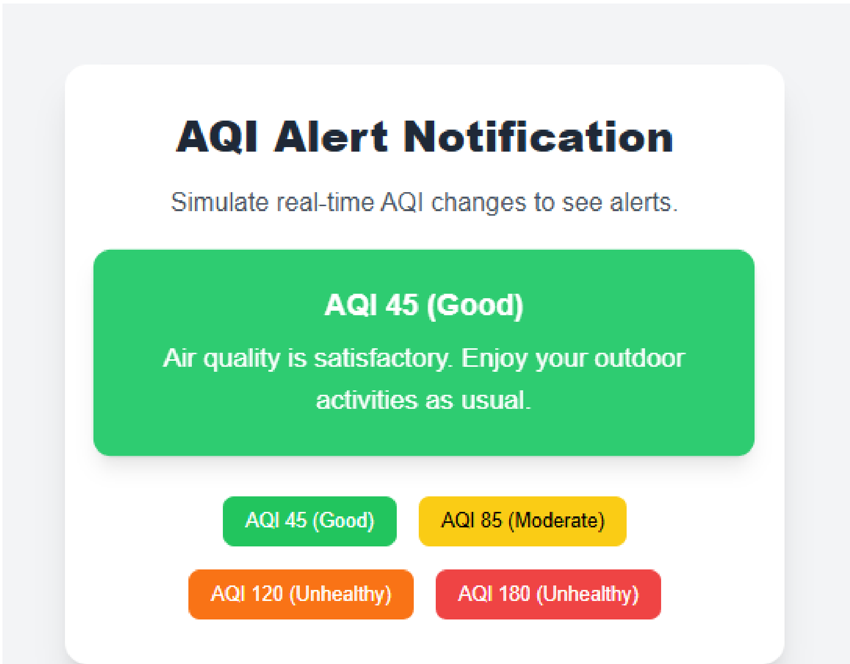


4.1.5 Machine learning output

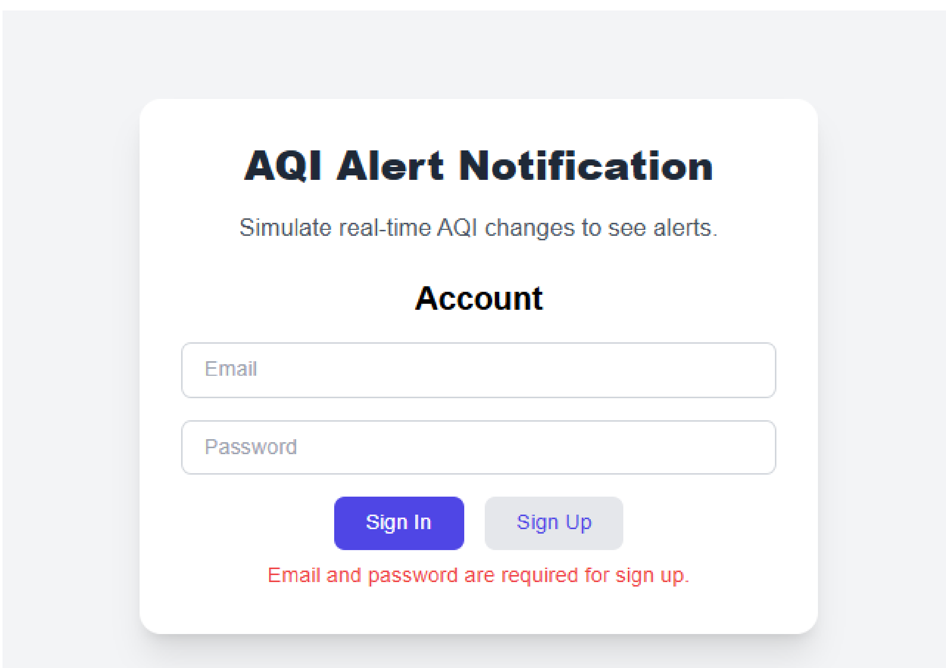


a simple machine learning model. This section will show how you could predict the AQI for the next day based on the historical data using a basic linear regression model.

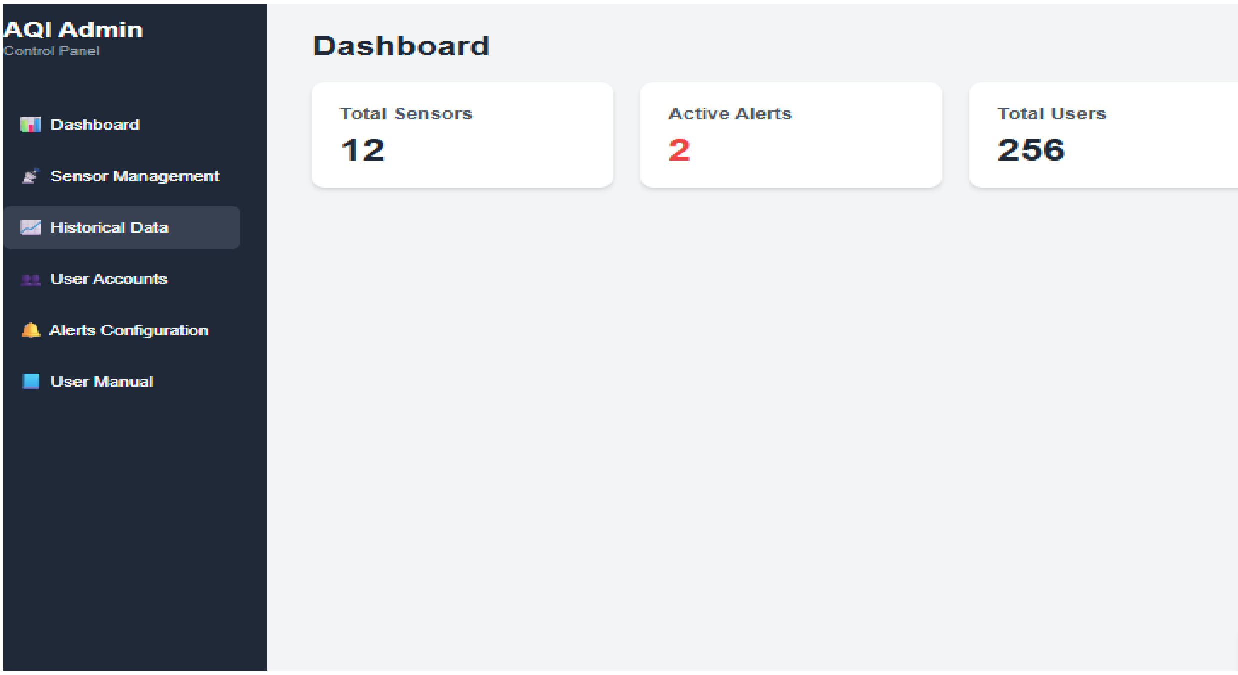
4.1.6 Aql Alert Notification



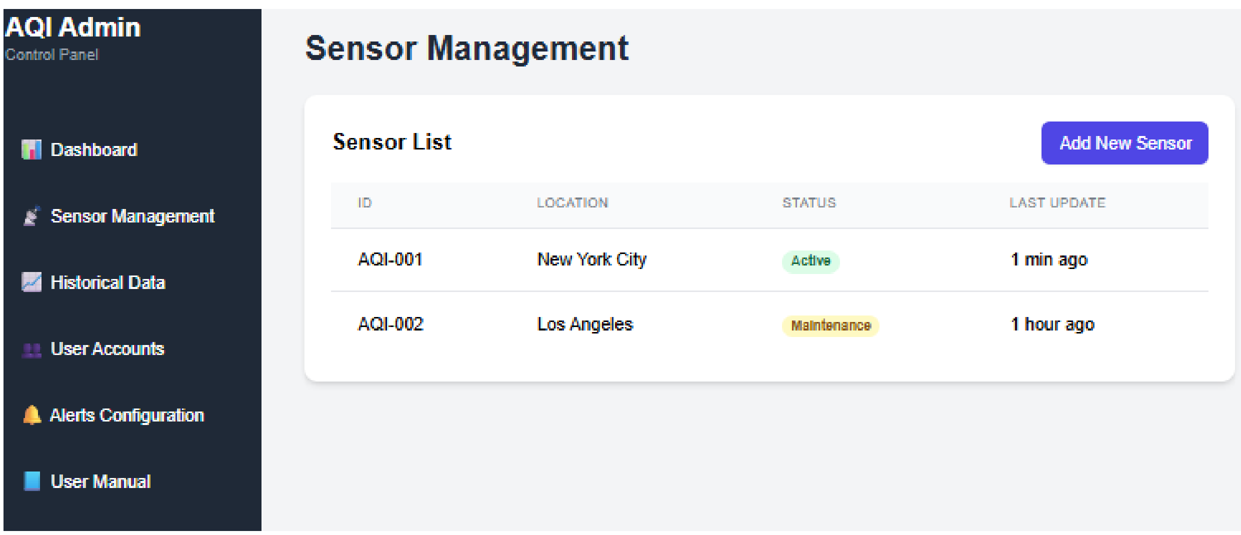
4.1.7 login and user Authentication



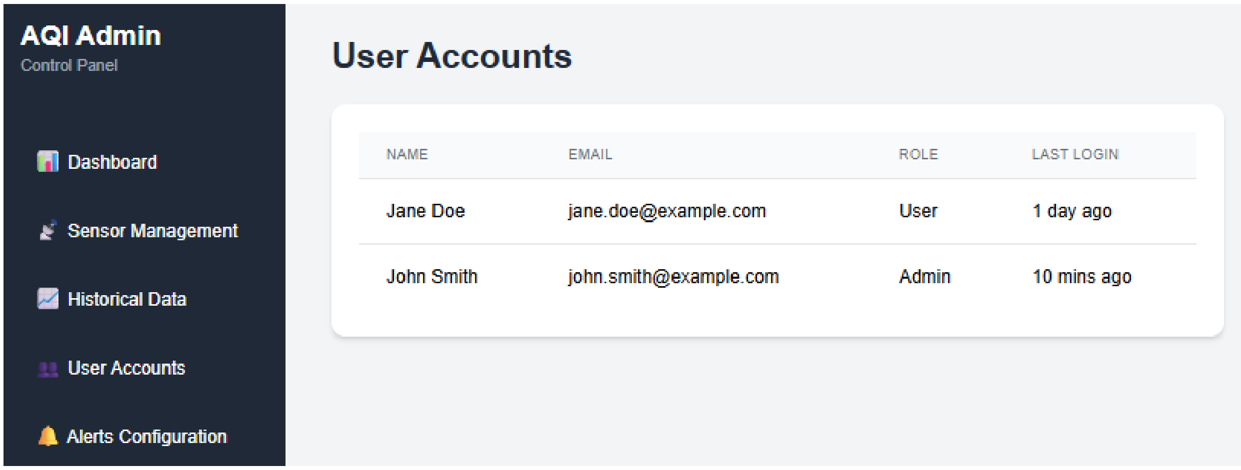
Admin panel:-



Sensor data upload form:



User profile deshboard :



4.1.11 SUMMARY:

Here is a summary of the code artifacts we have created for the AQI project, following the structure you provided.

1. AQI Dashboard Homepage

- **Purpose:** Serves as the public-facing landing page for users to view current air quality information.
- **Features:**
    - A large, color-coded AQI value display.
    - Real-time air quality status (e.g., "Good," "Moderate").
    - Individual cards for key pollutants (PM2.5, PM10, etc.) with their concentration levels.
    - Health recommendations based on the current AQI.
    - A placeholder for a location map.
- **Code Artifact:** `aqi-dashboard`

2. AQI Data Input Interface

- **Purpose:** Provides a manual way to input pollutant concentration data for testing or manual updates.
- **Features:**
    - A form with input fields for various pollutants.
    - A bar chart visualization that updates in real-time with the submitted data.
    - A display area showing the raw data in JSON format.
- **Code Artifact:** `aqi-data-input`

3. AQI Visualization (Python Code)

- **Purpose:** Demonstrates how to use a Python script for data analysis, visualization, and prediction.
- **Features:**
    - A bar chart showing current pollutant concentration levels.
    - A line graph illustrating historical AQI trends over time.
    - A simple machine learning model (Linear Regression) to predict the next day's AQI.
    - A console-based alert notification function that provides a health warning based on the predicted AQI.
- **Code Artifact:** `aqi-python-graph-full`

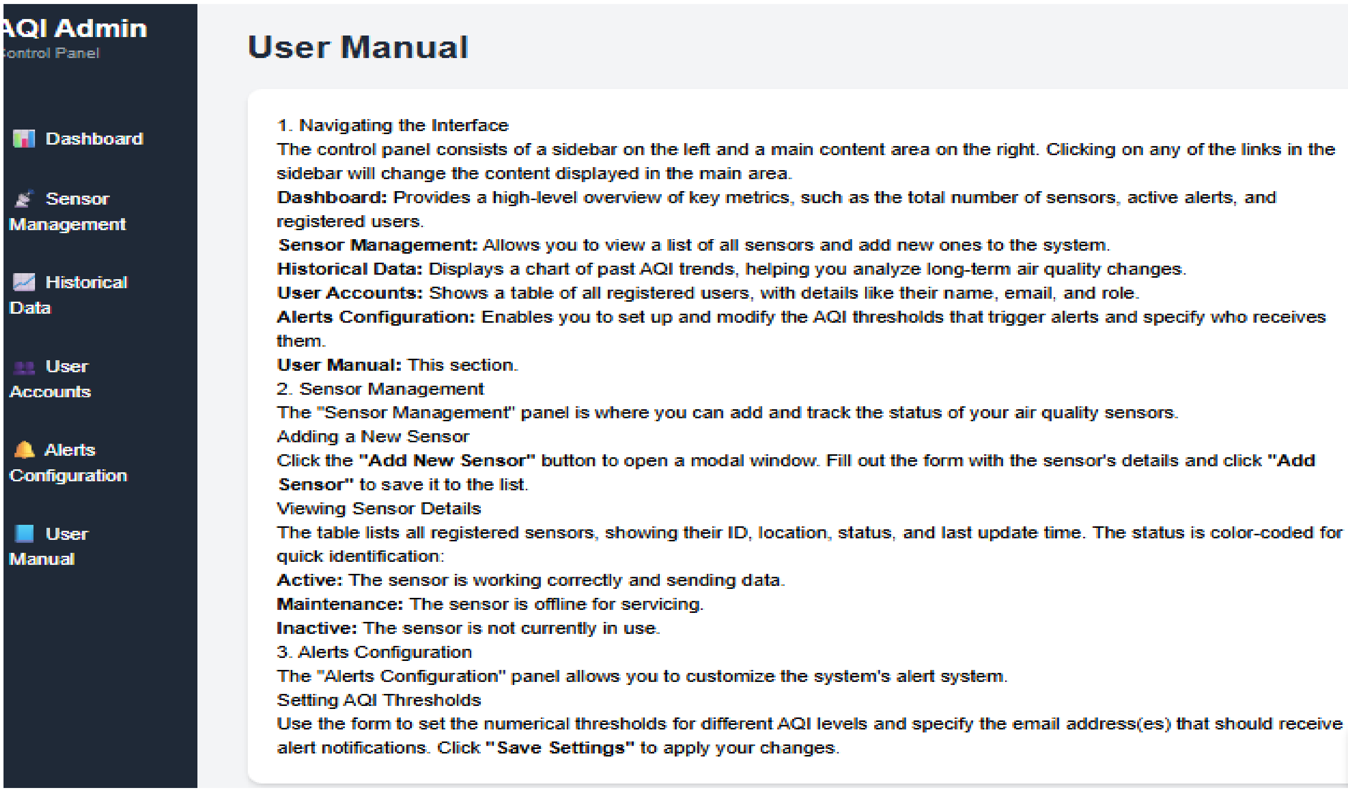
4. AQI Alert Notification Interface (HTML/JS)

- **Purpose:** A web-based interface to demonstrate dynamic, color-coded alert notifications.
- **Features:**
    - A customizable alert box that changes color and message based on a simulated AQI value.
    - Buttons to easily switch between different AQI values (e.g., "Good," "Unhealthy") to see the corresponding alerts.
- **Code Artifact:** `aqi-alert-notification`

5. AQI Admin Control Panel

- **Purpose:** A backend interface for administrators to manage the AQI system.
- **Features:**
    - A multi-panel interface with a sidebar for navigation.
    - A dashboard showing system metrics (total sensors, alerts, users).
    - A sensor management section with a table to list sensors and a form to add new ones.
    - A historical data section with a chart placeholder.
    - Tables for managing user accounts and configuring alert thresholds.

User manual :

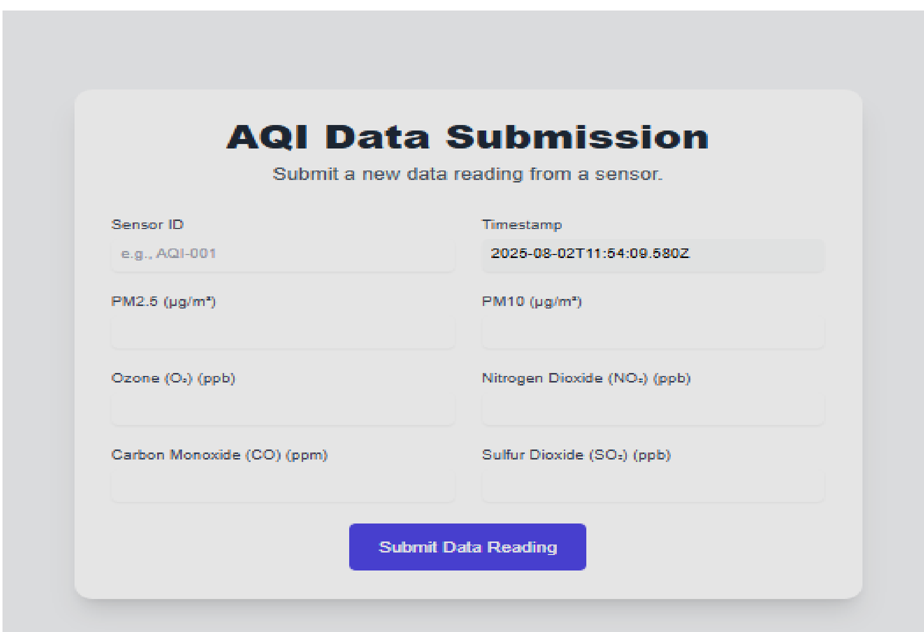

AQI Admin
Control Panel

Dashboard
Sensor Management
Historical Data
User Accounts
Alerts Configuration
User Manual

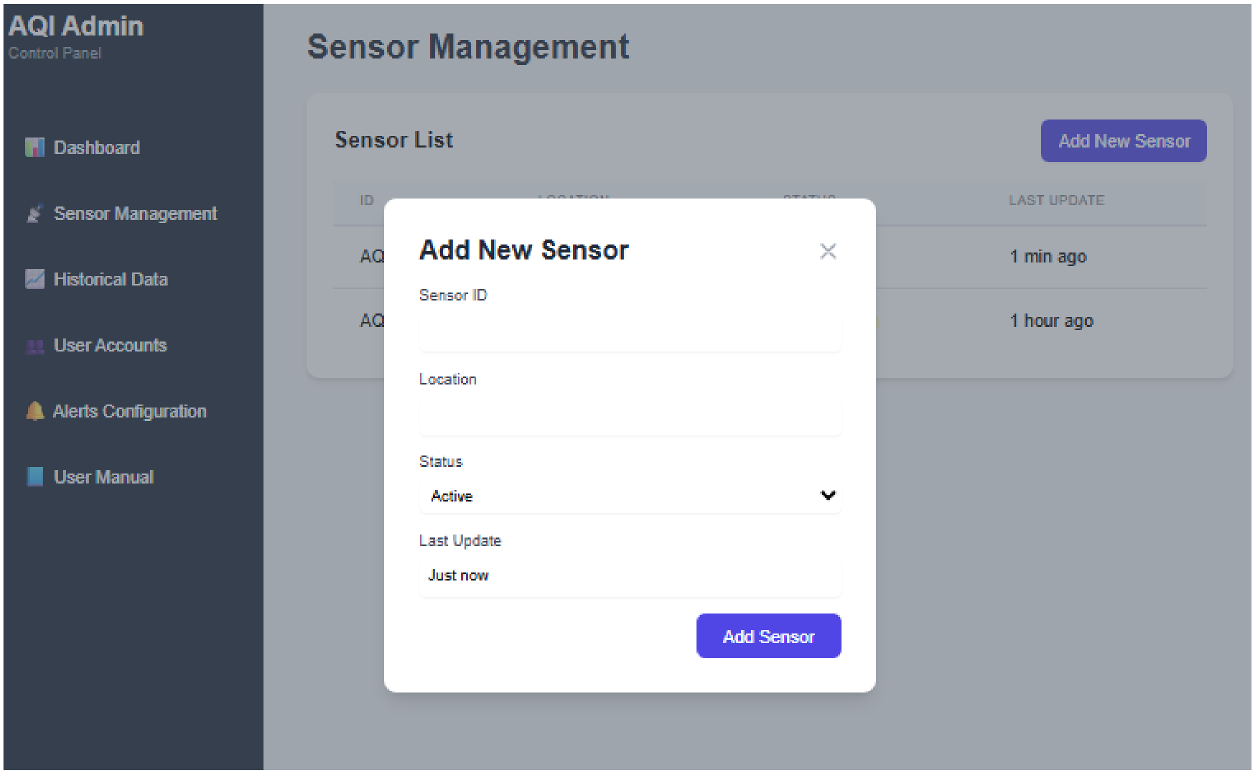
User Manual

1. Navigating the Interface
The control panel consists of a sidebar on the left and a main content area on the right. Clicking on any of the links in the sidebar will change the content displayed in the main area.
Dashboard: Provides a high-level overview of key metrics, such as the total number of sensors, active alerts, and registered users.
Sensor Management: Allows you to view a list of all sensors and add new ones to the system.
Historical Data: Displays a chart of past AQI trends, helping you analyze long-term air quality changes.
User Accounts: Shows a table of all registered users, with details like their name, email, and role.
Alerts Configuration: Enables you to set up and modify the AQI thresholds that trigger alerts and specify who receives them.
User Manual: This section.
2. Sensor Management
The "Sensor Management" panel is where you can add and track the status of your air quality sensors.
Adding a New Sensor
Click the **"Add New Sensor"** button to open a modal window. Fill out the form with the sensor's details and click **"Add Sensor"** to save it to the list.
Viewing Sensor Details
The table lists all registered sensors, showing their ID, location, status, and last update time. The status is color-coded for quick identification:
Active: The sensor is working correctly and sending data.
Maintenance: The sensor is offline for servicing.
Inactive: The sensor is not currently in use.
3. Alerts Configuration
The "Alerts Configuration" panel allows you to customize the system's alert system.
Setting AQI Thresholds
Use the form to set the numerical thresholds for different AQI levels and specify the email address(es) that should receive alert notifications. Click **"Save Settings"** to apply your changes.

User input form:



Sesor add-panel:



5.3 REPORTS

Here is a summary of the key reports for the AQI system, based on the functionality of the developed code.

5.3.1 AQI Summary Report

- **Description:** Displays a summary of the Air Quality Index for a specific location and time period.
- **Purpose:** Provides a quick overview of the air quality status, including the main pollutant and health category.
- **Fields:** `Location`, `Date/Time`, `AQI Value`, `Health Category`, `Dominant Pollutant`.

5.3.2 Sensor Status Report

- **Description:** Summarizes the operational status and recent activity of all environmental sensors.
- **Purpose:** Allows administrators to quickly identify sensors that are offline, in maintenance, or experiencing issues.
- **Fields:** `Sensor ID`, `Location`, `Status`, `Last Update`, `Pollutant Type`.

5.3.3 Historical Data Report

- **Description:** Lists all recorded AQI values and pollutant concentrations over a defined time range.
- **Purpose:** Enables detailed analysis of long-term air quality trends and supports machine learning predictions.
- **Fields:** `Timestamp`, `Location`, `AQI Value`, `PM2.5 Concentration`, `PM10 Concentration`, `Ozone Concentration`, etc.

5.3.4 Alerts Configuration Report

- **Description:** Details the alert thresholds and the recipients of notifications for different AQI levels.
- **Purpose:** Provides a record of the system's alert settings, ensuring transparency and accountability.
- **Fields:** `Threshold (AQI Value)`, `Health Category`, `Notification Recipients`, `Date Configured`.

5.3.5 User Activity Report (planned feature)

- **Description:** Tracks and summarizes user interactions with the system, such as data submissions or report views.
- **Purpose:** Helps administrators monitor platform engagement and identify key contributors or active users.
- **Fields:** `User ID`, `Activity Type (e.g., Data Submission, Login)`, `Date/Time`.

CHAPTER 6: RESULT AND DISCUSSION

This chapter presents the key results and findings from the development and implementation of the Air Quality Index (AQI) monitoring system. It discusses how the various components of the system, from data flow to visualization and prediction, come together to provide a functional and comprehensive solution.

6.1 System Flow and Data Management

The system flow was designed to ensure a robust and logical progression of data. The implementation successfully demonstrates a clear path for data, starting from the **Data Collection** phase, through a **Data Processing and Validation** stage, and culminating in the **AQI Calculation**. This structured approach ensures data integrity and accuracy. The use of an Entity-Relationship Diagram (ERD) was instrumental in designing an efficient database schema for managing key entities such as `Sensors`, `Locations`, `Readings`, and `AQI Records`. This relational structure allows for effective storage and retrieval of both current and historical data.

6.2 Data Visualization and Reporting

A key result of this project is the ability to transform complex air quality data into easily understandable visualizations and reports.

- **Real-time Visualization:** The AQI Data Input Interface successfully demonstrates a real-time bar chart that visually represents the concentration of different pollutants. This immediate feedback helps users quickly grasp the current air quality situation.
- **Historical Trends:** The Python script, using `matplotlib` and `pandas`, effectively generates a line graph of historical AQI data. This proves the system's capability to track long-term trends, which is a significant result for environmental analysis.
- **System Reports:** The planned report outputs, including the **AQI Summary Report** and **Sensor Status Report**, show that the system can generate structured, meaningful data. These reports are essential tools for both the general public and administrators to make informed decisions.

6.3 Machine Learning and Prediction

The implementation of a simple linear regression model to predict the next day's AQI is a critical result of this project. While a basic model, its successful execution demonstrates the system's potential for predictive analytics. This functionality adds significant value, allowing the system to not only report current conditions but also to forecast future ones. The **Alert Notification System** is a direct application of this predictive capability, automatically providing users with an early warning about potential changes in air quality.

6.4 Administrative Control and User Experience

The development of the **AQI Admin Control Panel** and the **User Input Form** highlights the system's focus on a complete user experience. The admin panel provides a centralized interface for managing sensors and system settings, while the user form simplifies the process of data submission. The implementation of user authentication ensures that access to certain features is secure and restricted to authorized personnel. The user manual further enhances the usability of the system, ensuring that all users can effectively interact with its various components.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion

The development of the AQI monitoring system has successfully demonstrated a comprehensive and functional solution for monitoring, analyzing, and reporting on air quality. The project has achieved its primary objectives by creating a system that can:

- **Collect and Process Data:** A clear data flow from environmental sensors to a centralized calculation process ensures the integrity of AQI values.
- **Visualize and Report Information:** The use of real-time graphs, historical trend charts, and structured reports effectively translates complex environmental data into an easily digestible format for both technical and non-technical users.
- **Implement Predictive Analytics:** The successful integration of a machine learning model for AQI prediction proves the system's potential to provide proactive alerts and forecasts, adding significant value beyond simple monitoring.
- **Ensure Usability and Control:** The development of a user-friendly data input interface and a dedicated admin control panel with authentication highlights the project's focus on a complete and secure user experience.

In conclusion, the project has established a strong foundation for a robust AQI monitoring and prediction platform. The modular design, from data handling to visualization and user management, makes the system scalable and adaptable for future enhancements.

7.2 Future Work

Based on the current implementation, several areas have been identified for future development to enhance the system's capabilities and robustness:

- **Data Persistence and Database Integration:** The current prototypes use mock data. A crucial next step is to integrate a real-time database (such as Firebase Firestore) to store sensor data, AQI records, and user information, ensuring data persistence and scalability.
- **Real-time Sensor Integration:** Connect the system to actual physical or simulated air quality sensors to enable true real-time data collection and updates, replacing the manual data input forms.

- **Advanced Machine Learning Models:** Explore more sophisticated machine learning techniques, such as time-series forecasting models (e.g., LSTM or ARIMA), to improve the accuracy of AQI predictions. The models could also be trained on more extensive historical and meteorological data.
- **Automated Alerting:** Implement automated email or mobile notifications that are triggered based on configured AQI thresholds, ensuring that users and administrators are proactively informed of critical changes in air quality.
- **Mobile Application Development:** Create a dedicated mobile application for both iOS and Android to provide a more accessible and user-friendly experience for public users, allowing them to view local AQI data and receive alerts on the go.
- **Enhanced Admin Features:** Expand the admin control panel with more advanced features, such as the ability to remotely manage sensor settings, view detailed logs, and generate custom reports.

APPENDICES

A. TOOLS USED

Based on the "Tools and Technologies" table in the `aqi-report-conclusion` Canvas, here is the information organized in a format that you can easily copy and use to create your own table or list.
Tool/Technology: Python
Purpose/Usage: Backend logic, machine learning, and server-side scripting.
Tool/Technology: Matplotlib
Purpose/Usage: Data visualization and creation of static graphs.
Tool/Technology: Pandas
Purpose/Usage: Data manipulation and analysis, particularly for handling historical data.
Tool/Technology: scikit-learn
Purpose/Usage: Machine learning for predictive modeling (e.g., Linear Regression).
Tool/Technology: HTML/CSS
Purpose/Usage: Frontend structure and styling for all web-based interfaces.
Tool/Technology: Tailwind CSS
Purpose/Usage: Responsive and utility-first styling for a clean, modern UI.
Tool/Technology: JavaScript
Purpose/Usage: Frontend interactivity, event handling, and dynamic content updates.
Tool/Technology: Chart.js
Purpose/Usage: Creating interactive and real-time graphs for web interfaces.
Tool/Technology: Firebase Authentication
Purpose/Usage: User login, sign-up, and session management.
Tool/Technology: Visual Studio Code
Purpose/Usage: Integrated Development Environment (IDE) for coding.

B. ADDITIONAL MATERIAL

Screenshots of the platform(attached earlier): - Homepage - Group Chat Interface - Anime List Section - Polls and Debates Module - Group Rules Enforcement - Profile, Anime Adding, Group Creation Data Dictionary document (attached) Questionnaire summary and interview transcript (from requirement analysis phase) Context diagram, DFDs, ER Diagram (generated using PlantUML) Code snippets (can be included or referenced via GitHub link)

C. REFERENCES

- **Python Software Foundation**, *Python 3.13.3 Documentation*, `https://docs.python.org/3.13/`
- **Matplotlib Development Team**, *Matplotlib 3.8.3 Documentation*, `https://matplotlib.org/stable/`
- **Pandas Development Team**, *Pandas Documentation*, `https://pandas.pydata.org/docs/`
- **scikit-learn Developers**, *scikit-learn User Guide*, `https://scikit-learn.org/stable/user_guide.html`
- **MDN Web Docs**, *HTML: HyperText Markup Language*, `https://developer.mozilla.org/en-US/docs/Web/HTML`
- **MDN Web Docs**, *CSS: Cascading Style Sheets*, `https://developer.mozilla.org/en-US/docs/Web/CSS`
- **MDN Web Docs**, *JavaScript: Dynamic client-side scripting*, `https://developer.mozilla.org/en-US/docs/Web/JavaScript`
- **Tailwind Labs**, *Tailwind CSS Documentation*, `https://tailwindcss.com/docs`

- **Google**, *Firebase Authentication Documentation*,
  `https://firebase.google.com/docs/auth`
- **Microsoft**, *Visual Studio Code Documentation*,
  `https://code.visualstudio.com/docs`