IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Minada Vinuwara Amarasinghe
Student Reference Number: 10952558

Module Code: PUSL2020	Module Name: Software Development Tools and Practices
Coursework Title: Air Quality Monitoring Dashboard – Testing Report	
Deadline: 29th April 2025	Member of staff responsible for coursework: Dr. Rasika Ranaweera/Ms. Pavithra Subhashini
Programme: BSc. (Hons.) Software Engineering	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: **Group 52**

Plymouth University ID	Name
10952558	Minada Vinuwara Amarasinghe
10952556	Damithu Dinlaka Samarasinha
10952751	Ruchika Kalhara Ekanayake
10953037	Simeshaa Barskaran
10952712	Bejunge Thusith Dhananjaya Warnakulasooriya
10952976	Gamage Basthiyan Appuhamillage Praveen Tharusha Munasinghe

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: 

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***
Signed:

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.
I *have ~~used~~/not used translation software.
If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

Table of Contents

GitHub repo link:

https://github.com/minababo/AirQualityDashboard.git

Individual Contribution Breakdown

Plymouth ID	Name	Contributions
10952558	Minada Amarasinghe	- Simulation engine - Frontend charting - Database integration - Test case documentation - Critical test analysis
10952556	Damithu Samarasinha	- Sensor CRUD - Admin Panel - Wrote functional test plans - Integration of test cases
10953037	Simeshaa Barskaran	- Leaflet mapping - Home map display - Designed UI validation tests and mock objects
10952751	Edirisinghe Ekanayake	- Data filtering - Pagination - Tested data persistence and regression testing
10952712	Bejunge Warnakulasooriya	- AQI trend chart - Data sorting - Conducted integration testing and coverage reporting
10952976	Gamage Munasinghe	- Alert Configuration and display - Created unit tests for alert evaluation logic

Introduction

This testing report documents the testing strategies our team used while developing the Real Time Air Quality Monitoring Dashboard. It also includes the test cases and the results for the whole development phase. Testing was done to make sure that the functionalities, performance, the usability and reliability of the application were met. Some of the key features incorporate real-time AQI data simulation, alert thresholds, map-based and chart-based data visualization.

System Overview

The backend is built using ASP.NET Core (.NET 8), which handled CRUD operations, data simulations, and alert generations.

Microsoft SQL Server is used for the database schema as it provides robust performance and has native integration with .NET system.

The frontend includes responsive Razor Pages using HTML/CSS/JS, Chart.js for visual trends, and Leaflet for visual mappings.

Public users can access the home page, sensor list page, as well as each sensors' details pages. Admins must register and login to access the admin panel where they can manage sensors, see system statuses, control simulation, configure simulation thresholds, and monitor real-time alerts for AQI levels.

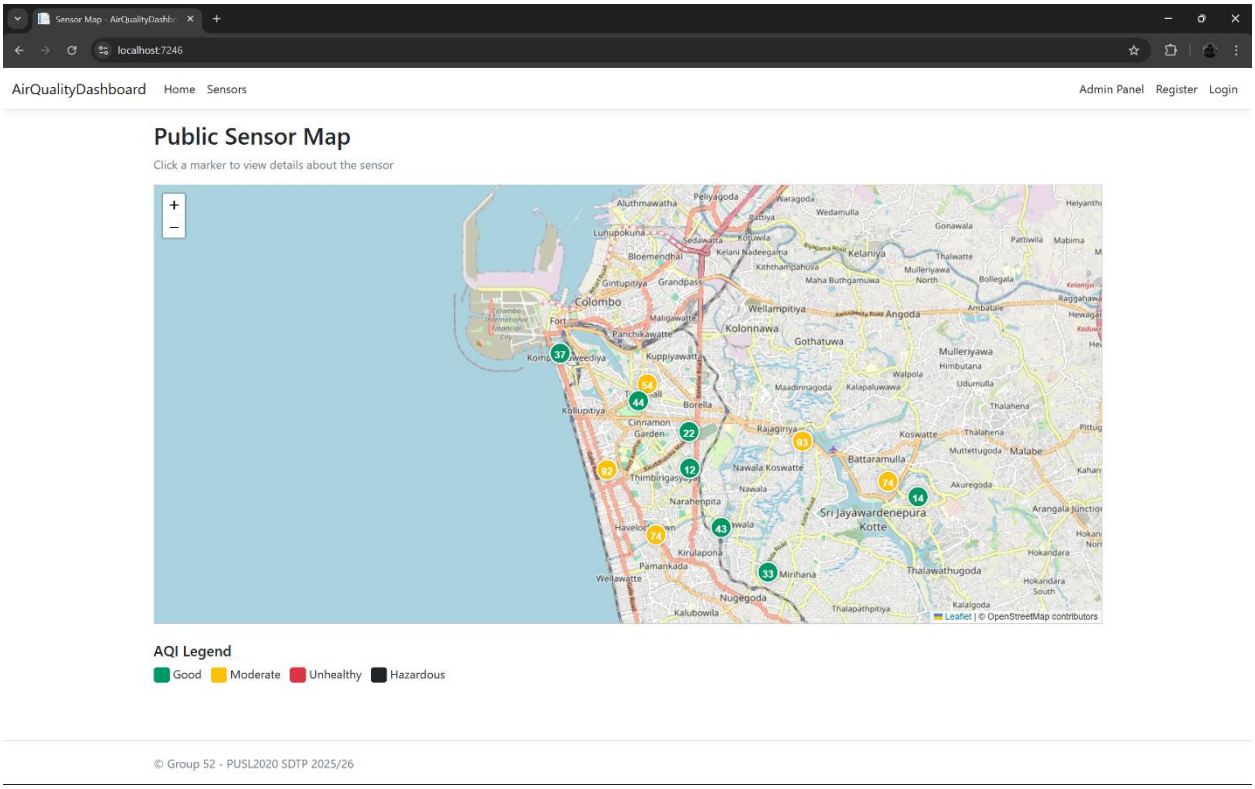
AQI data values are automatically simulated every 5 minutes in order to keep consistency

Technologies Used

Component	Tool
Backend	ASP.NET Core (C#)
Database	Microsoft SQL Server
Frontend	HTML, CSS, JS, Leaflet, Chart.js
Data Generation	Our own simulation service
Mapping Library	Leaflet.js

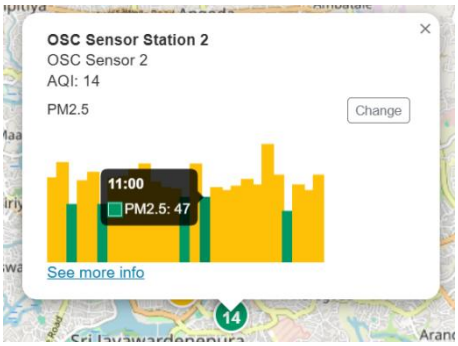
Development Evidence and Screenshots

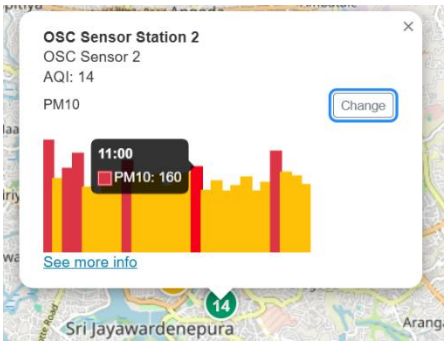
Home page with public map view



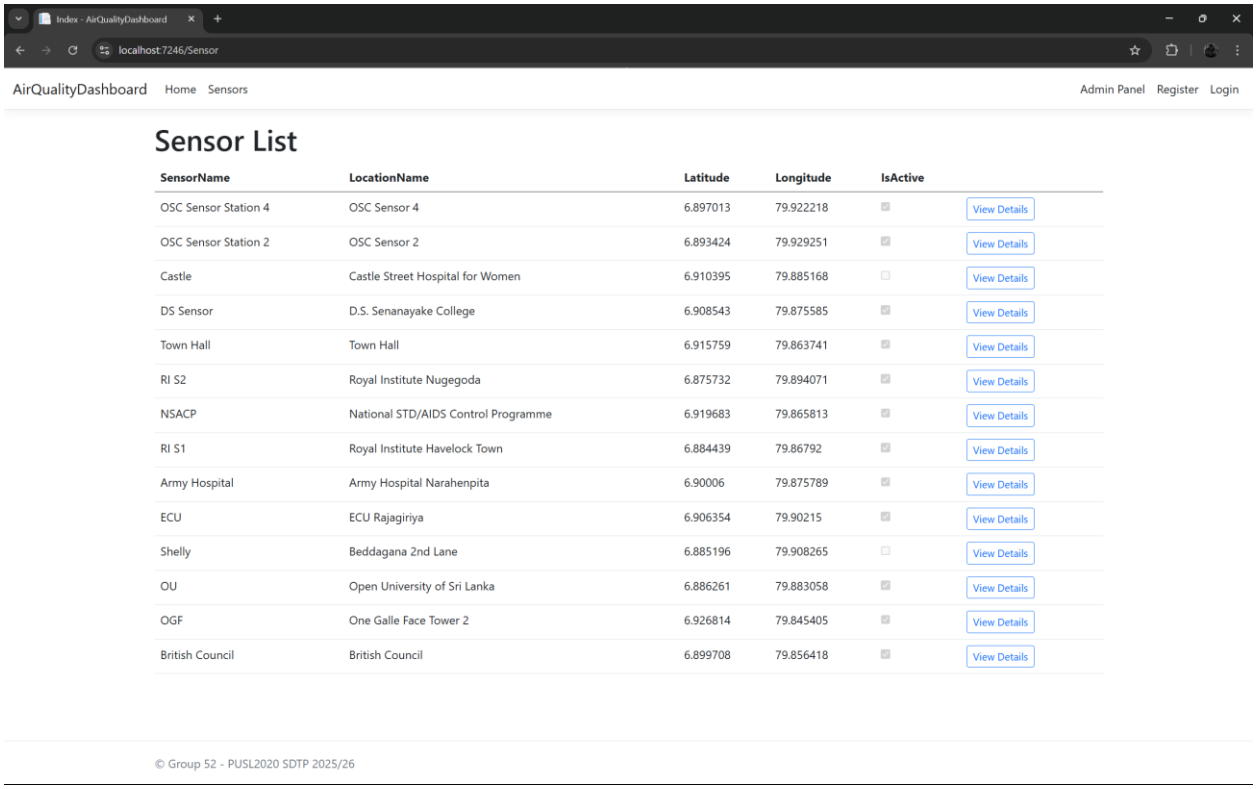
Developers: Minada and Simeshaa

- This page displays a public map of the sensors within Colombo using markers.
- Each sensor marker is color-coded and showcases the AQI PM2.5 reading of that sensor.
- A legend is provided below the map so that the users know whether the current levels are ‘Good’, ‘Moderate’, ‘Unhealthy’, or ‘Hazardous’.
- Once a marker is clicked a small popup will be shown of that sensor’s details including its name, location, current AQI reading, along with a chart showing recent historical AQI trends of PM2.5 and PM10. The two reading’s views switch automatically after every 30 seconds but can also be changed manually using the change button given in the popup.
- The popup also contains the link to that certain sensor’s Details page.





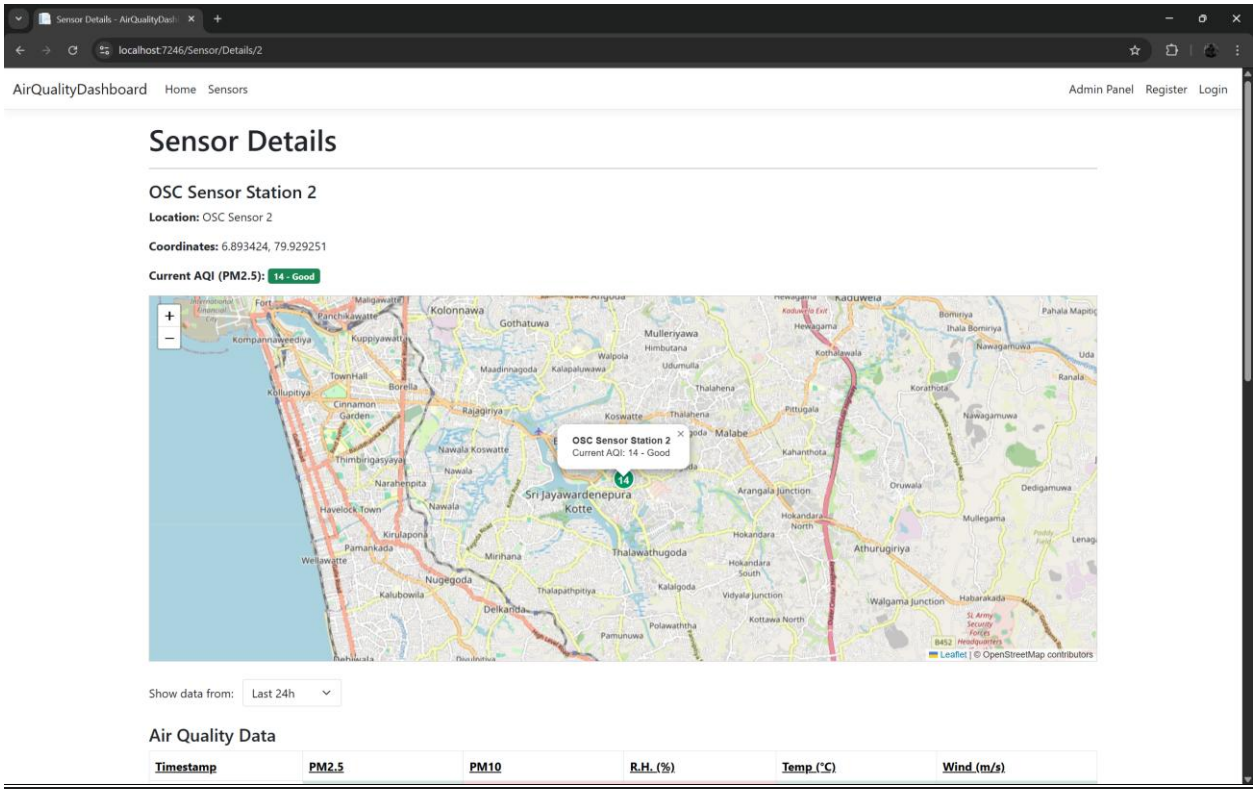
Sensor Listings (Public)

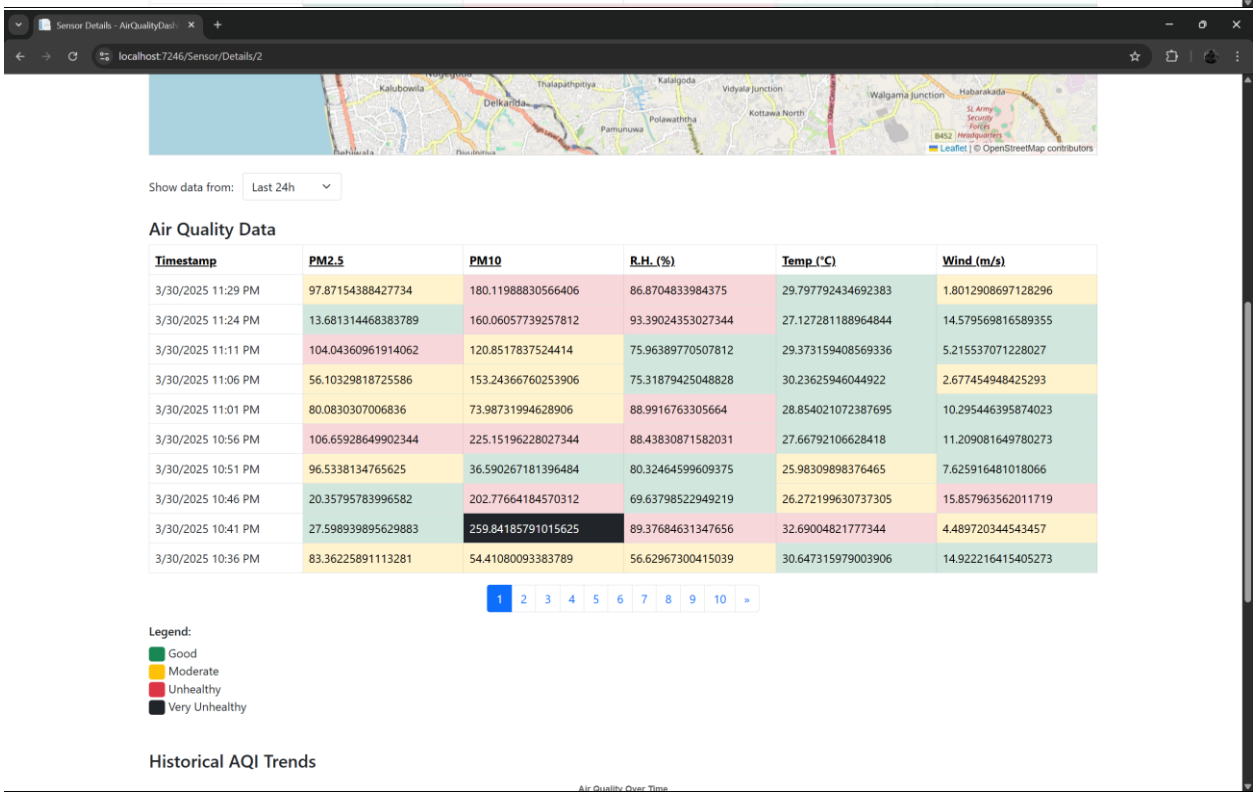


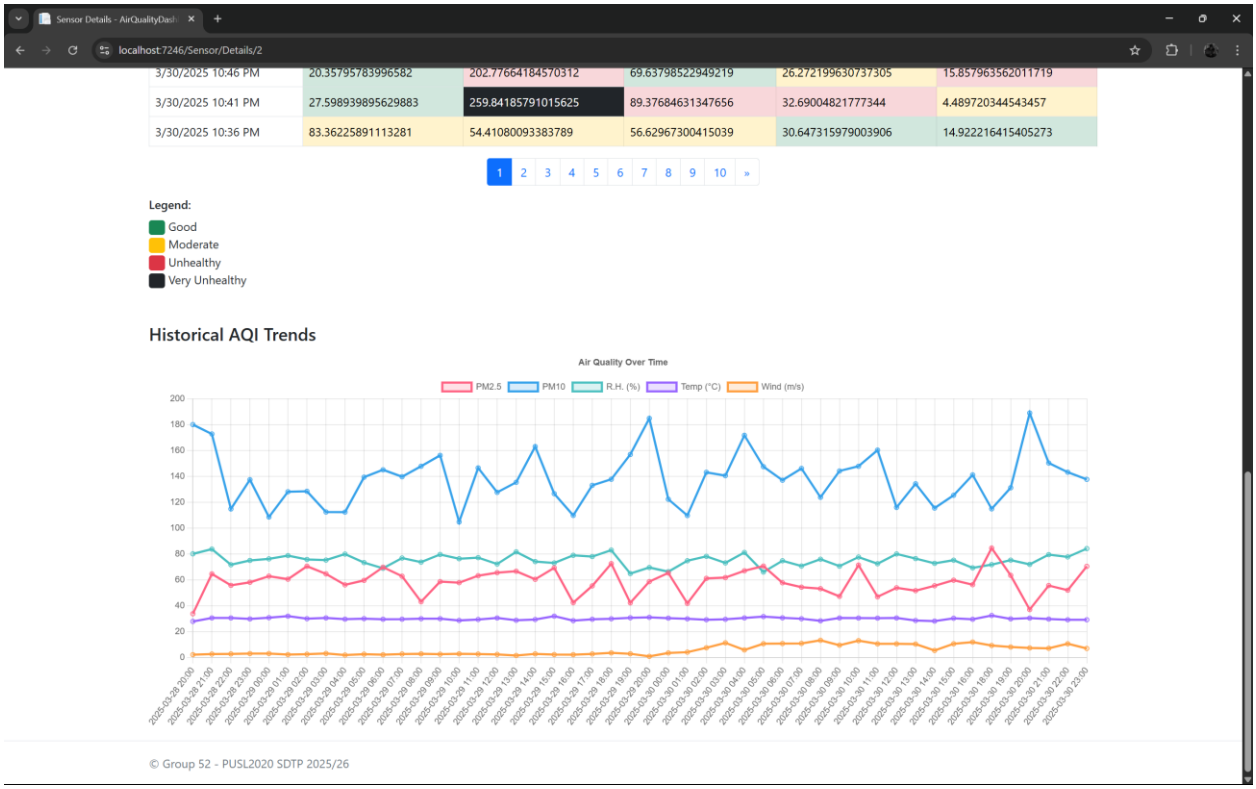
Developer: Simeshaa

- This page displays all the sensors that are in the system whether they are active or inactive.
- A link to each sensor’s details page is provided through the ‘View Details’ button.

Detailed AQI View

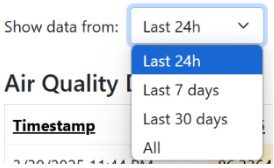






Developers: Ruchika and Thusith

- This page has three sections including basic details about the sensor and the location on the map, Air Quality Data on a table, as well as a chart displaying the Historical AQI Trends.
- Users can filter the data being displayed on the table and chart to show the all the data from the last 24 hours, last 7 days, last 30 days or even all.



- We also implemented a sorting functionality so users can sort the required data by ascending or descending order.
- A pagination feature was also used to reduce cluttering of data within the page.

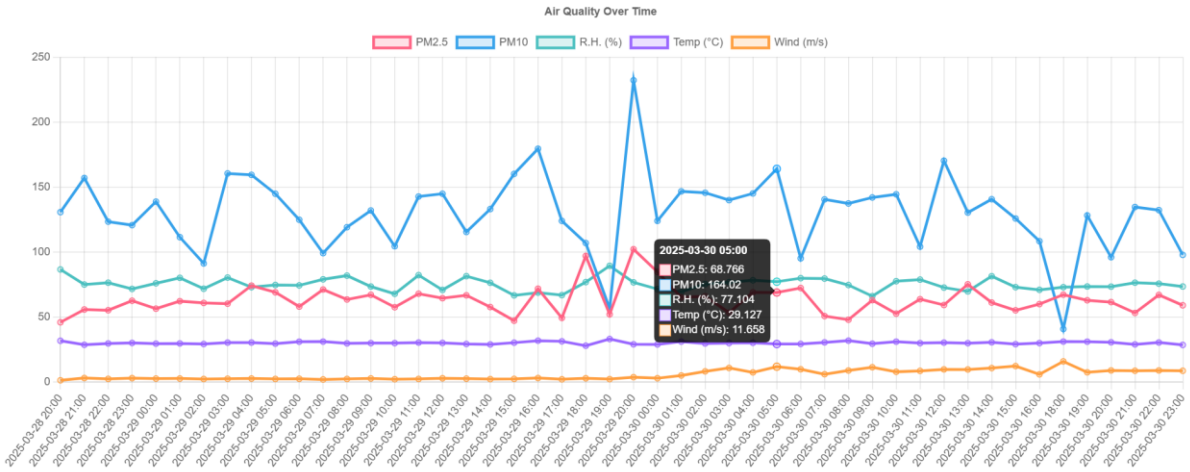
Air Quality Data

Timestamp	PM2.5 ↑	PM10	R.H. (%)	Temp (°C)	Wind (m/s)
3/30/2025 11:29 PM	10.025225639343262	139.64549255371094	55.45015335083008	29.916872024536133	9.251155853271484
3/30/2025 8:41 PM	11.9668550491333	231.6153564453125	95.111572265625	29.420642852783203	11.088501930236816
3/30/2025 8:11 AM	12.751749992370605	206.88381958007812	99.40458679199219	32.465110778808594	4.764034271240234
3/30/2025 9:26 AM	12.898783683776855	52.16794204711914	63.38166809082031	32.607486724853516	14.646935462951166
3/30/2025 9:01 AM	13.364786148071289	73.56887817382812	55.52970886230469	28.758390426635742	9.017989158630371
3/30/2025 7:11 PM	13.705099105834961	88.31914520263672	94.84929656982422	25.456106185913086	2.213207483291626
3/30/2025 11:54 AM	14.086128234863281	54.95329284667969	58.223419189453125	28.518539428710938	15.141676902770996
3/30/2025 8:06 AM	14.242352485656738	61.897560119628906	52.503997802734375	33.93632507324219	8.663663864135742
3/30/2025 4:01 AM	14.470894813537598	157.4715576171875	54.71873474121094	32.10086822509766	7.489993095397949
3/30/2025 12:09 PM	14.6093168258667	161.19931030273438	60.599884033203125	28.247467041015625	2.055905342102051

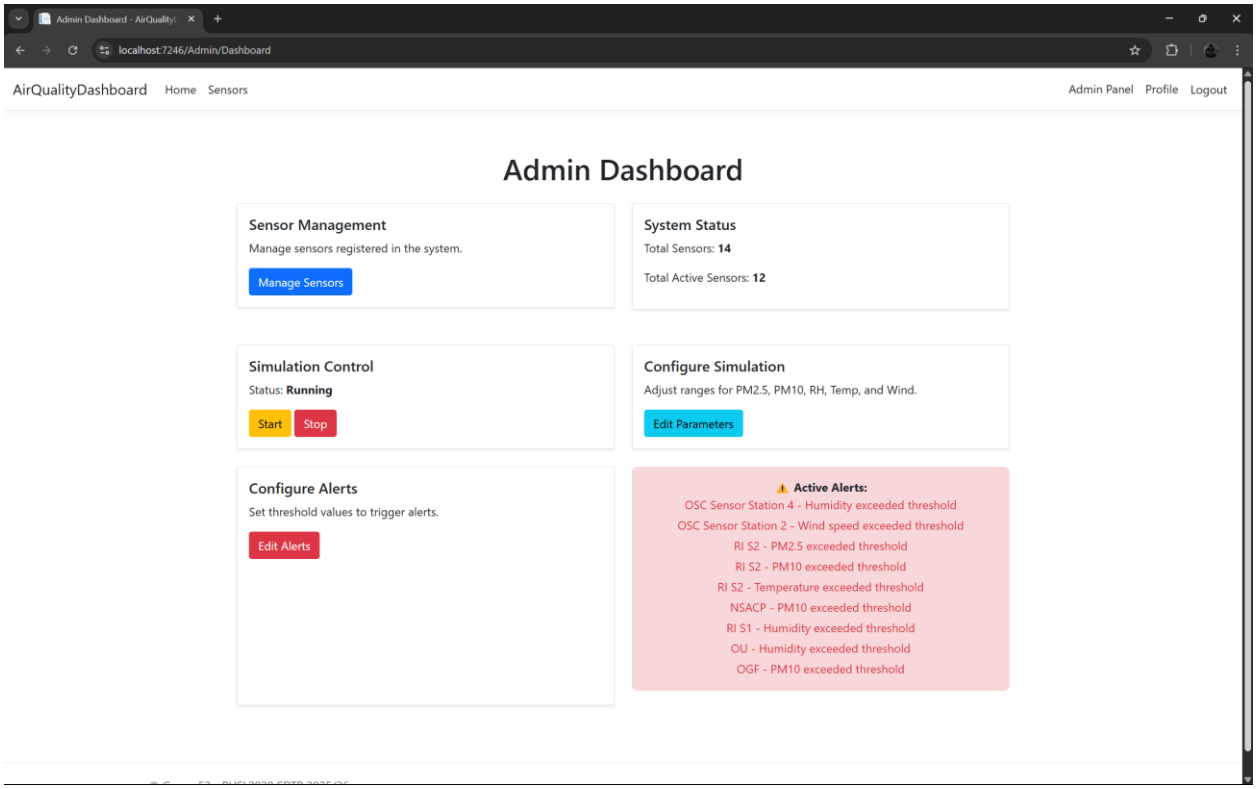
1 2 3 4 5 6 7 8 9 10 »

- Charting was used to display Historical AQI Trends by taking the average AQI PM2.5 value each hour. Each AQI reading is clearly shown using colours and hovering the mouse on a point in the chart displays a clear view of all the reading averages from that hour.

Historical AQI Trends



Admin Dashboard

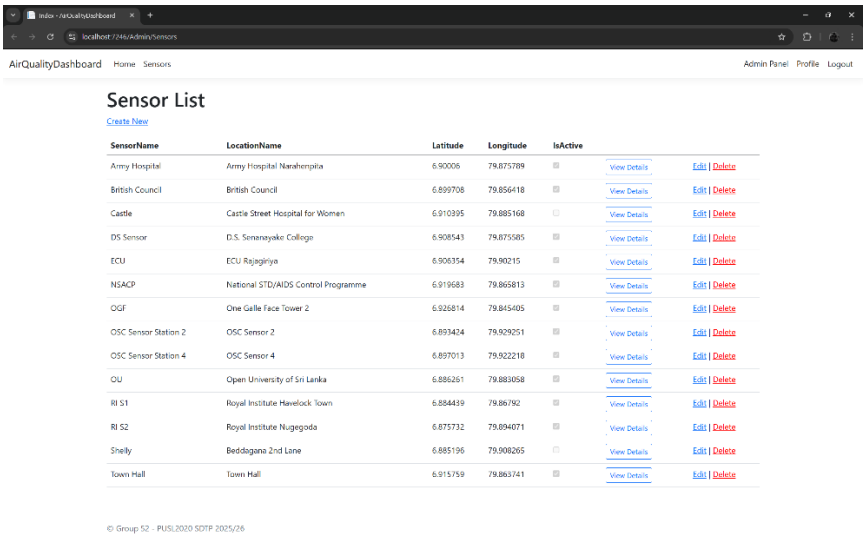


Developers: Damithu, Minada and Praveen

- This page includes all the actions admins can perform in cards.

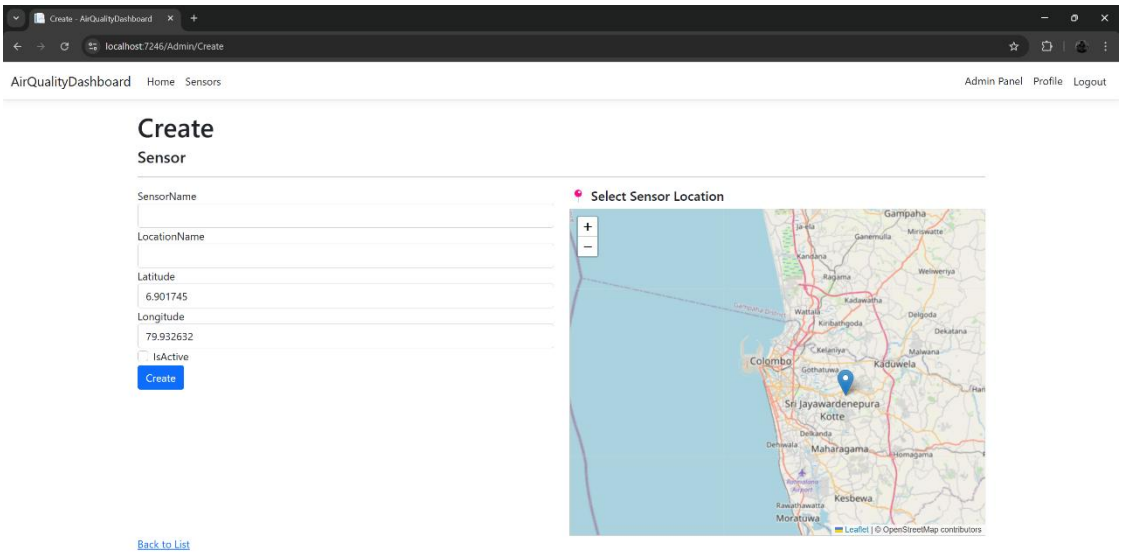
Sensor Management

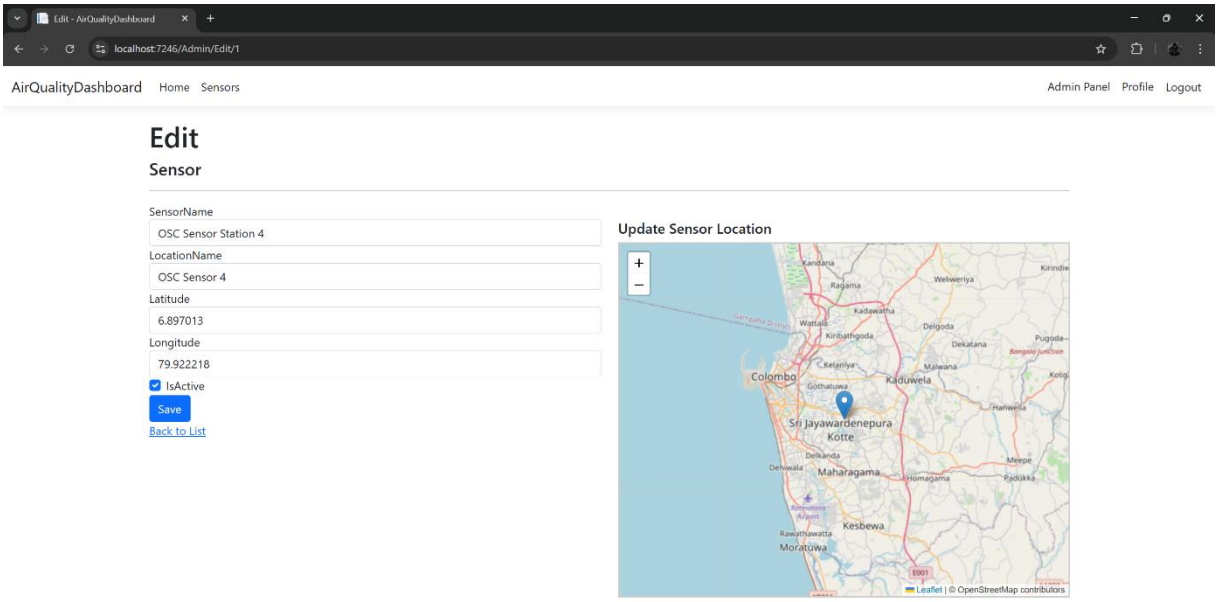
- Clicking the ‘Manage Sensors’ button redirects to the sensor list page but including Sensor CRUD operations this time.

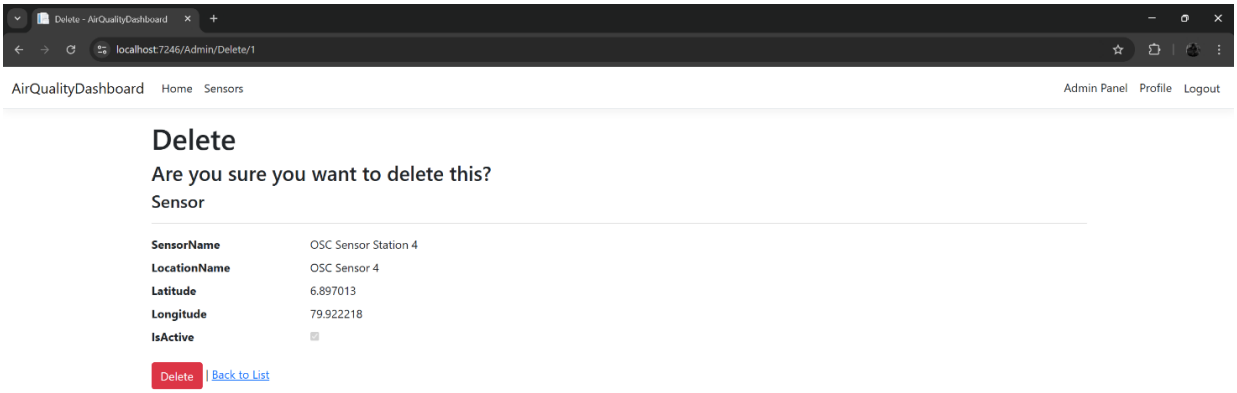


Developer: Damithu

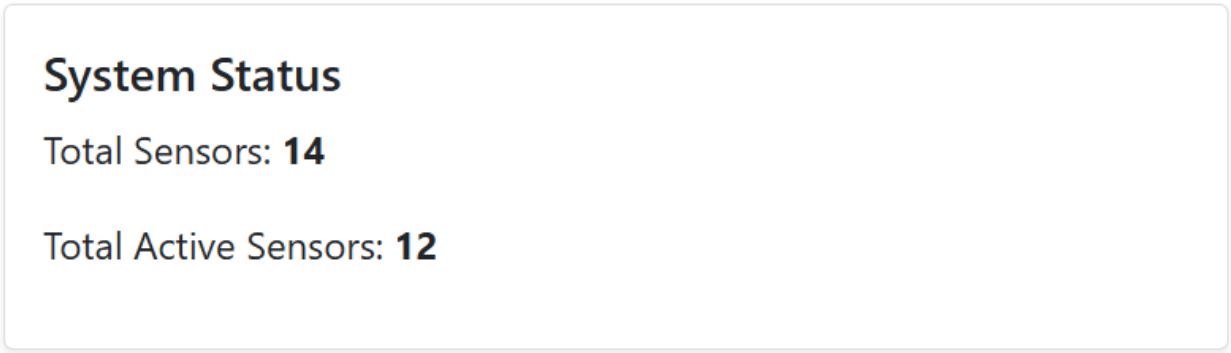
- When creating a sensor, the admin can select the location by clicking anywhere on the map which will provide the latitude and longitude on the fields. Same logic applies for when editing.







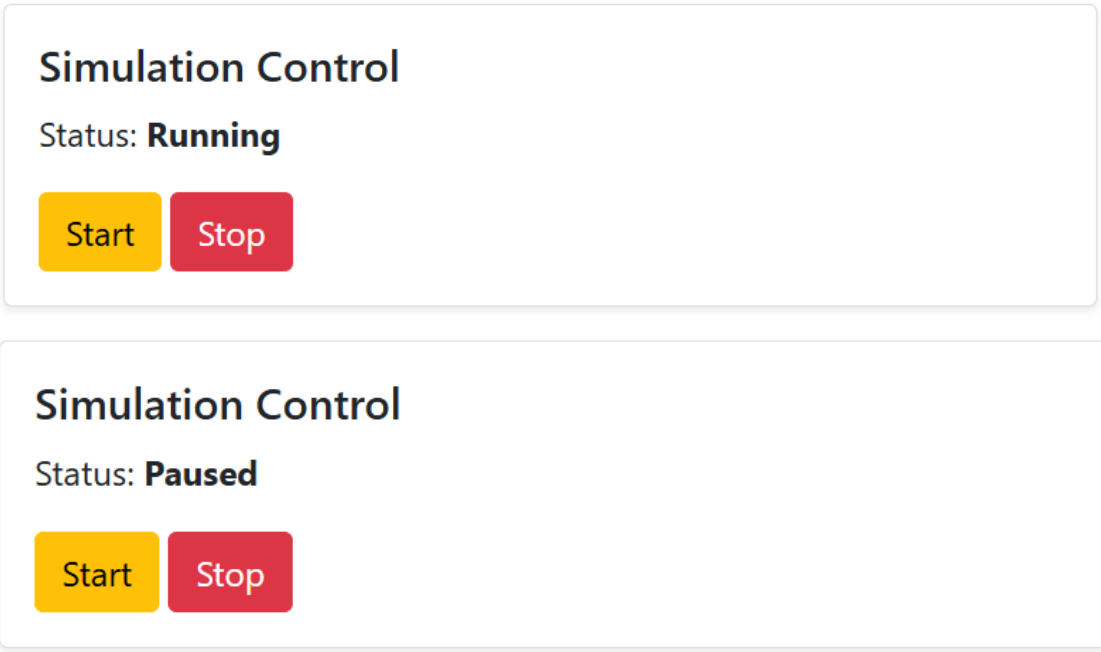
System Status



Developer: Damithu

- Displays the current total and active sensor counts

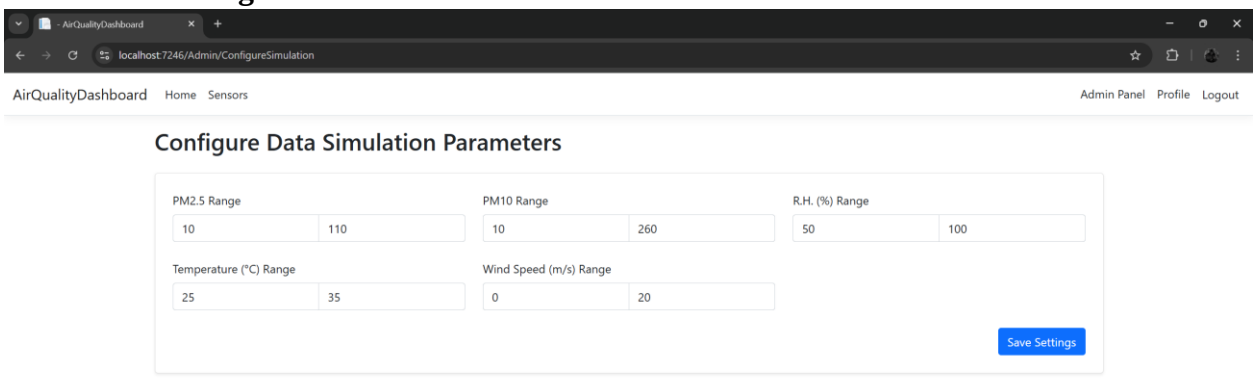
Simulation Control



Developer: Minada

- This functionality allows the admin to start/stop the automated AQI data generation service that is happening in the background.

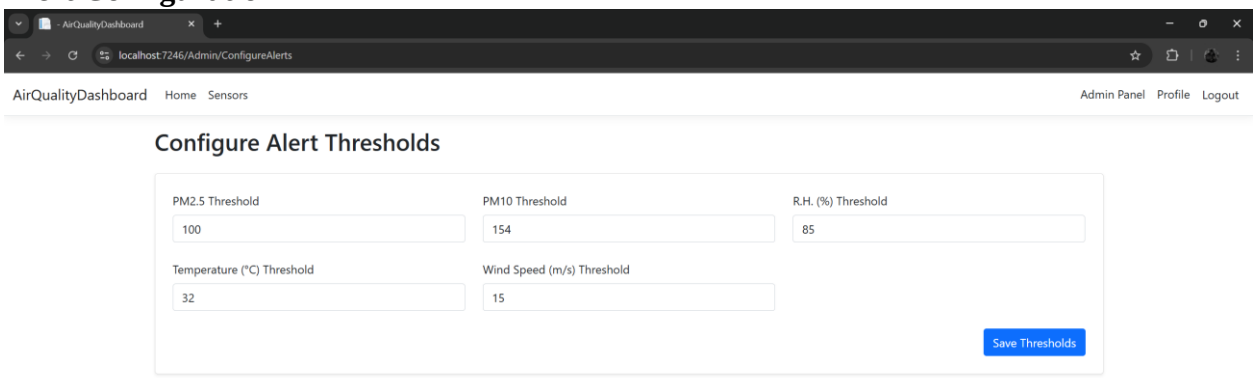
Simulation Configuration



Developer: Minada

- This feature can be used to change the range of values each AQI data simulates. By providing the two end points in the given fields and saving the settings, the data that is being simulated ranges from those points.

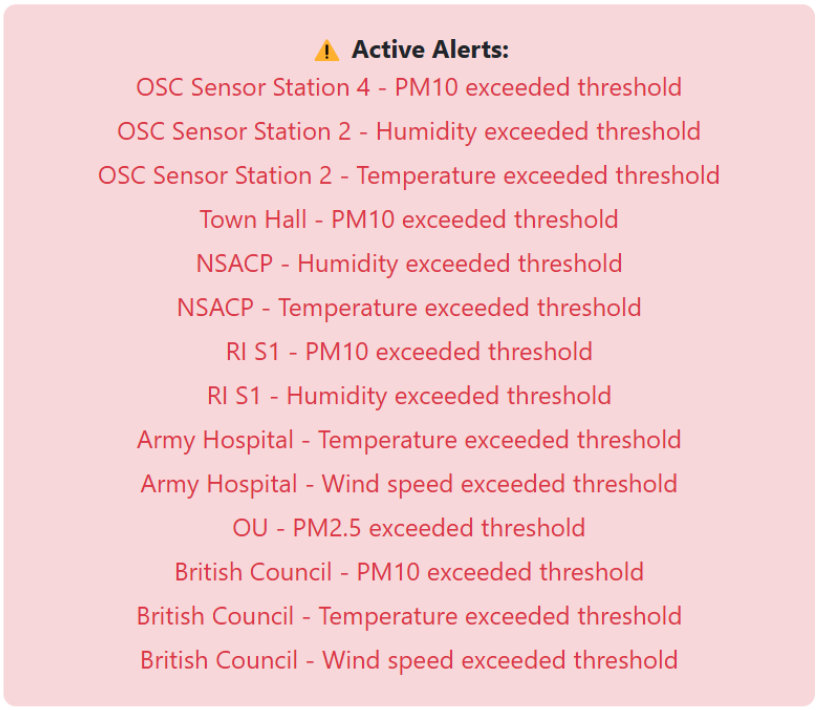
Alert Configuration



Developer: Praveen

- Using this the admin can change the threshold which is used to give out alerts for each AQI reading value that exceeds the given threshold.

Alert Notifications



Developer: Praveen

- This function auto-refreshes every 15 seconds to dynamically show alerts of the AQI readings that exceed the given threshold from the previous feature. Admin can also be redirected to any wanted sensor that is shown in this notification board by clicking on it.

Description of Test Cases

A thorough set of test cases were used to evaluate the air quality monitoring system's performance, functionality, usability, and accuracy.

Key testing categories included:

- **Sensor management** – creating, editing, viewing, and deleting sensors
- **AQI data simulation and presentation** – graphs, tables, and real-time updates
- **Mapping** – verifying that all sensor markers, complete with popups and colour-coded values appear on the map accurately
- **Sorting, pagination and filtering** – ensure that the right information is displayed based on user needs
- **View access** – ensuring that admins and public users have the appropriate access permissions
- **Admin simulation controls** – examining how start/stop actions impact on data generation
- **Threshold alerts** – confirming that alerts are appropriately triggered by admin-configured values

Every test case included the following stages – description of test, actions done to perform the test, expected results, and actual results and status of the test. Some examples consisted of:

Test Case ID	Description	Action	Expected Result	Actual Result	Status
TC01	Add a new sensor	Enter sensor data and location	Sensor saved and appears in the list and public map	As expected	✓
TC02	Trigger alert threshold	Set low threshold	Dashboard displays alert notifications	As expected	✓
TC03	Filter AQI Data by 'Last 7 Days'	Select 'Last 7 Days' filter on the dropdown	Only AQI data relevant to the last 7 days are displayed in the table	As expected	✓
TC04	Sort AQI table by PM10	Click on PM10 header	Visually verify the ordering after sorting the AQI table by ascending order of PM10	As expected	✓
TC05	View sensor marker popups	Click on a marker on the map	Popup information and graph rendered properly	As expected	✓
TC06	Paginate AQI data	Click page number	Verify each page has the right entries	As expected	✓

TC-07	Sensor with invalid coordinates (e.g. null)	Submit new sensor without coordinates	System rejects submission and shows validation error	As expected	✓
TC-08	Changing alert thresholds mid-simulation	Vary thresholds for AQI readings while simulating	System immediately adapts alert detection logic	As expected	✓
TC-09	Map pop-up chart validation	PM10/PM2.5 chart toggle repeatedly clicked, and then waiting 30 seconds	Chart updates consistently, no crashes or stuck states. After 30 seconds it automatically switches between the two readings	As expected	✓

Structure and Role of Mock Object

The system made use of simulation-based mock objects to facilitate testing of different components without depending on real-time sensor hardware. The sensor data readings that were automatically generated and periodically saved in the database were represented by these mocks.

A background service that produced AQI data for every sensor served as the primary mock component.
The previously mentioned admin panel's adjustable range was used by this service.

Structure:
- Data for every active sensor was generated using the mock generator
- To replicate real-time monitoring, data points were generated every several seconds
- Within predetermined min/max constraints, values were randomized

Role in Testing:
- Enabled repeatable testing of real-time updates
- Allowed to have hardware-free functional testing of AQI trend charts
- Made sure that controlled situations could be used to validate the alert system
- Facilitated testing of database and user interface interaction under demand

Unit and Integration Tests

Our system includes multiple validation stages for both backend and frontend components. Tests were conducted in this manner:

- Backend logic was tested by manually executing controller activities and validating database state before and after operations
- Integration testing between views and databases was carried out by monitoring system behavior as real-time mock data passed through
- Simulation parameters and alert levels were modified using the admin UI and tested with visual indicators and alert displays
- The AQI data table and trend graphs were validated using live and historical data from real-world interactions

All components were set up locally via Microsoft SQL Server and tested in browser across numerous scenarios.

Functional Test Plans

The plan covers the following functional test areas:

- Create and validate sensors
- Store and retrieve AQI data
- Render live dashboards and maps
- Configure admin pages
- Trigger and display alerts
- Create and view graphs
- Sort, filter and paginate data

The system's functional testing strategy consists of structured scenario-based tests that simulate real-world activities. Each key feature, such as sensor creation, AQI data filtering, and map marker display, was verified to ensure that they performed as expected. Boundary instances, such as missing coordinates or severe data, were also verified. Administrative privilege checks were included to guarantee that only authorized users may change configurations. Public user’s access to the admin dashboard was restricted to ensure correct authorization flow. Sensor creation involves entering data, selecting a location using the map, and validate accurate database and UI updates.

Each functionality's success and failure conditions were tested thoroughly and appropriately.

Regression Testing

To make sure the features that were already there continued to work, we ran tests again each time after adding a new feature. For example, chart plotting and map popups were there from the beginning and so when adding sorting, alerts, and pagination features, we checked them again to see if anything was affected and to assure that all the functions worked properly.

Cross-Browser and Responsive Testing

To ensure accessibility and reliability, the application was tested on a variety of modern browsers, including Chrome, Edge, and Firefox. Functional features such as charts, filters, and sorting were tested for consistency. The layout and popups on the public sensor map were also tested on multiple screen sizes (desktop, tablet, and mobile) to ensure responsiveness and usability. Map tiles, AQI markers, and data tables are properly scaled across resolutions.

Testing Tools Used

Tool	Purpose
Web Browser	Manually check functions
Browser DevTools	Manual frontend and performance checks
Chart.js	Visual confirmation of AQI trends
Leaflet	Verifying correct display of markers

Critical Analysis of the Test Strategy

The test technique emphasizes realism and repeatability, employing controlled simulation for real-time data and confirming behavior through browser-based observation and UI replies.

This approach enabled testing with continuously updated data.
- Reproduction of known edge cases.
- Comprehensive coverage of the data flow pipeline, including simulation, storage, visualization, and alerting.

Limitations:
- Hardware integration was not possible due to a lack of linked sensors.
- Manual test validation is necessary for many UI interactions.

Despite its limitations, the approach provided complete verification of basic functionality, scalability, and responsiveness.