



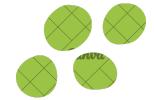
Air-Aware Smart Air Quality Prediction System

-Smarter Air, Cleaner Lives

Presented By Aleena Noor



Project Overview



Purpose

- Predict AQI & pollutant levels using time-series models
- Provide visual dashboards for trends & future air quality
- Generate alerts when predicted AQI becomes unsafe
- Analyze pollutant contributions and seasonal patterns
- Allow admins to update data & retrain forecasting models

Tools and Technologies used

- Dataset : Kaggle World Air Quality Dataset
- Python
- HTML, CSS and Chart.js(Javascript) for dashboards
- Streamlit for Interactive Dashboard
- Pandas
- Flask
- Google Colab

What is AQI ?

- AQI (Air Quality Index) is a number that shows how clean or polluted the air is.
- Helps understand whether air is safe, moderate, or harmful to breathe.

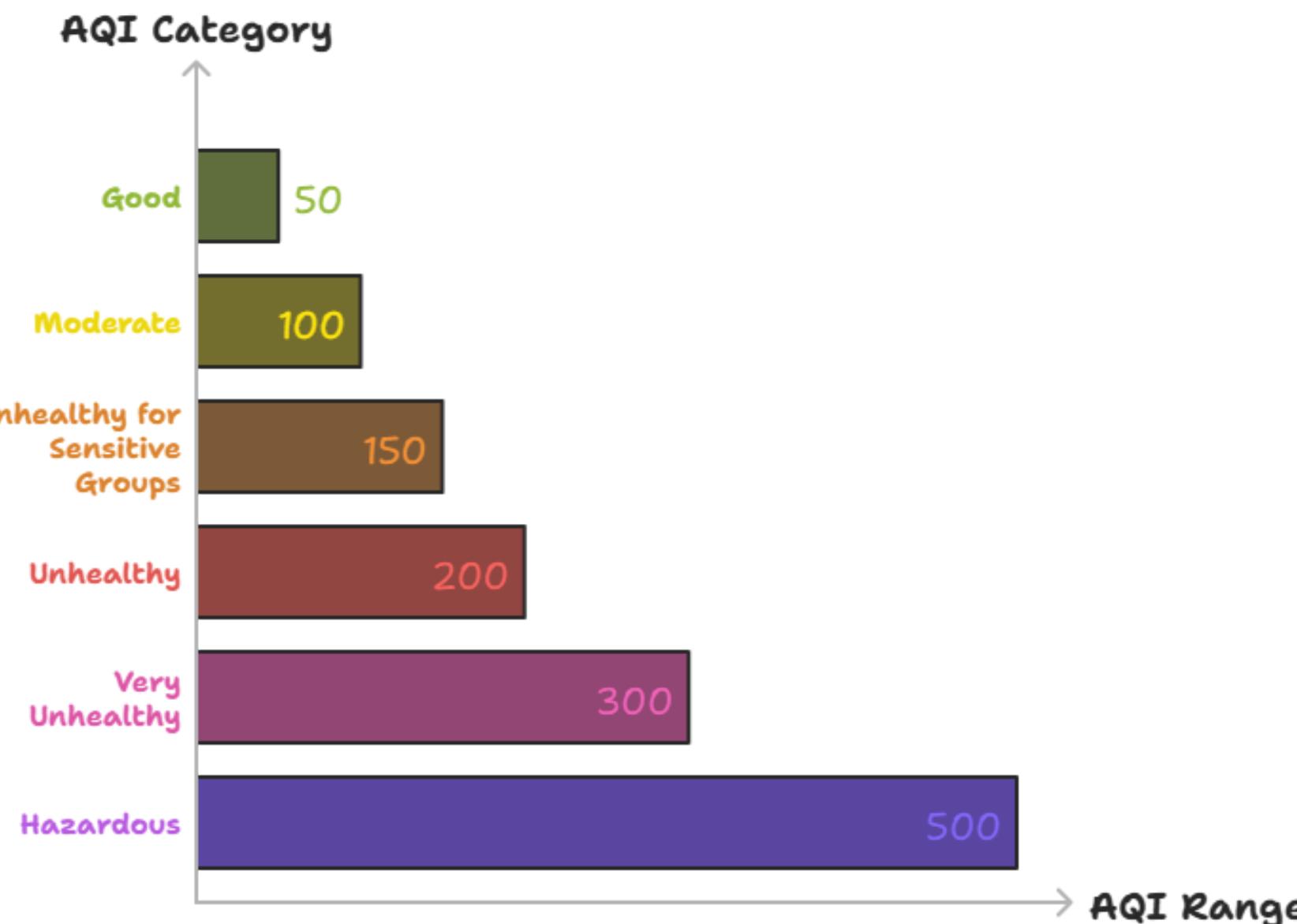
Factors affecting AQI

The major pollutants that determine AQI:

- **PM2.5** – Fine particles
- **PM10** – Dust & coarse particles
- **CO** – Carbon monoxide
- **SO₂** – Sulphur dioxide
- **NO₂** – Nitrogen dioxide
- **O₃** – Ozone

These pollutants contribute to the final AQI score

Air Quality Index Scale



AQI Measurement & Agencies



How pollutants are measured?

- **Dust Samplers** (PM_{2.5}, PM₁₀): Measure fine and coarse particulate matter in the air.
- **Gas Analyzers** (CO, SO₂, NO₂): Detect concentration of harmful gases like Carbon Monoxide, Sulphur Dioxide, and Nitrogen Dioxide.
- **UV Photometry** (O₃): Measures ozone levels using ultraviolet absorption.
- **CPCB Monitoring Stations**: Real-time air quality monitoring stations placed across different cities.

Agencies / Apps That Provide AQI

- **CPCB** – Central Pollution Control Board
- **SAFAR** – System of Air Quality and Weather Forecasting And Research
- **IMD** – India Meteorological Department
- Google AQI
- IQAir

Major Contributions to Air Pollution

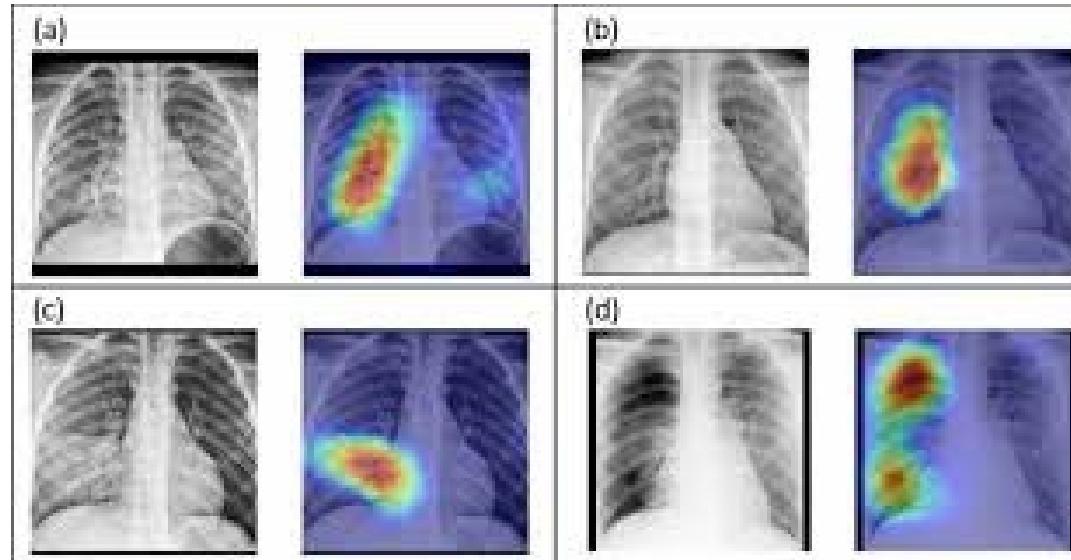
- **Vehicles**
 - Emissions from daily transport
- **Industries**
 - Smoke from factories
- **Stubble burning**
 - Crop residue burning
- **Solid fuel & biomass burning**
 - Wood, coal, and waste burning
- **Construction dust**
 - Dust from building activities
- **Brick kilns**
 - Smoke from brick-making units



Impact of Air Pollution & Possible Solutions

Effects on Humans

- Asthma & breathing problems
- Lung and heart diseases
- Eye and throat irritation
- Fatigue & low immunity



Effects on the Environment

- Smog formation
- Acid rain
- Climate change
- Reduced visibility



Solution to reduce Air Pollution

Use air purifiers & masks to reduce exposure to polluted air



Increase afforestation and protect green areas to improve air quality

Conduct public awareness programs on the importance of clean air



PUBLIC AWARENESS



Promote cleaner fuels like CNG, LPG, and EVs to reduce emissions

Milestone 1

Data Preprocessing and EDA

- Download and clean raw air-quality datasets
- Handle missing values, outliers, and inconsistent timestamps
- Perform EDA to study pollutant trends & correlations
- Resample data (hourly/daily) for forecasting
- Extract and prepare features for model training

Github Link : Milestone 1

Milestone 2

Model Training & Evaluation

- Train forecasting models: **ARIMA, Prophet, LSTM**
- Evaluate performance using **RMSE, MAE**
- Compare results across pollutants (**PM2.5, PM10, O₃ etc.**)
- Select the best-performing model
- Save trained models for prediction/inference

[Github Link : Milestone 2](#)

Dashboard 1 : Data Explorer

Dashboard Features

- Analyze **time-series trends of PM2.5 pollutant** over selected dates
- Visualize pollutant intensity using **bubble charts**
- View **statistical summaries** (mean, median, min, max, std dev)
- Explore **value distribution** through bar charts
- Filter data by **location, time range, and pollutant type**
- Check **data quality indicators** (completeness & validity)

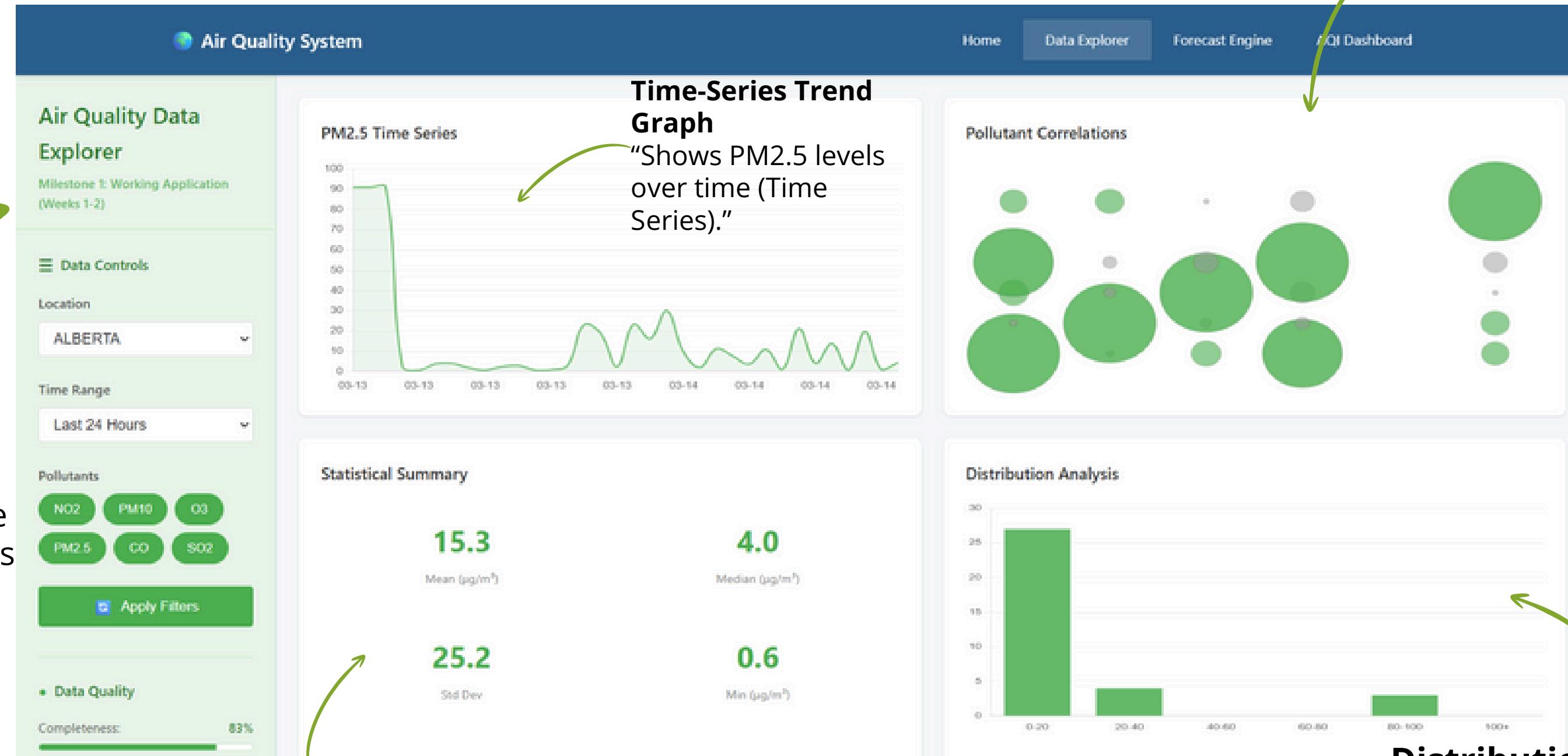
Tools & Technologies used

- **HTML** – structure of the dashboard
- **CSS** – styling, layout, theme design
- **JavaScript** – interactive functionality
- **Chart.js** – for charts (line chart, bubble chart, bar chart)
- **JSON / CSV Data** – loading AQI dataset
- **VS Code** – for development

Dashboard 1: Data Explorer

Location & Time Filters (Left Sidebar)

"Select location, time range, and pollutants to explore the AQI dataset."



Statistical Summary
"Mean, median, min, max & standard deviation."

Bubble Visualization
"Pollutant intensity represented by bubble size."

Distribution Analysis
"Distribution of pollutant values."

Dashboard 2 : Forecast Engine

Dashboard Features

- **Forecast** comparison of models (**ARIMA, Prophet, LSTM, XGBoost**)
- **RMSE and MAE** bar chart to compare model errors.
- **PM2.5 forecasting graph showing Actual vs Predicted values** for different model type (ARIMA / LSTM / Prophet / XGBoost) with **forecast horizon** (12h, 24h, 48h)
- **Best Model Table** showing best performing model for each pollutant
- **Forecast accuracy curves** comparing all models across different horizons

Tools and Technology used

- **HTML** – for dashboard layout
- **CSS** – for styling & UI design
- **JavaScript** – for interactivity
- **Chart.js** – for RMSE/MAE bar charts, accuracy curves & forecast graphs
- **Machine Learning Models** : ARIMA, Prophet, LSTM and XGBoost
- **JSON / CSV** – for forecast data
- **Python** – used to build, train, and evaluate forecasting models
- **Flask** – backend API to send forecast results to the dashboard.
- **VS Code** – for development

Dashboard 2 : Forecast Engine



Model RMSE/MAE Chart

"Model error comparison (RMSE) for each pollutant."

PM2.5 Forecast Graph

"Actual vs Forecasted PM2.5 levels over time for different models and on different horizons"

Dashboard 2 : Forecast Engine



Best Model Table
"Shows best-performing model per pollutant."

Forecast Accuracy Chart
"Accuracy comparison across forecasting horizons."

Dashboard 3 : AQI Dashboard

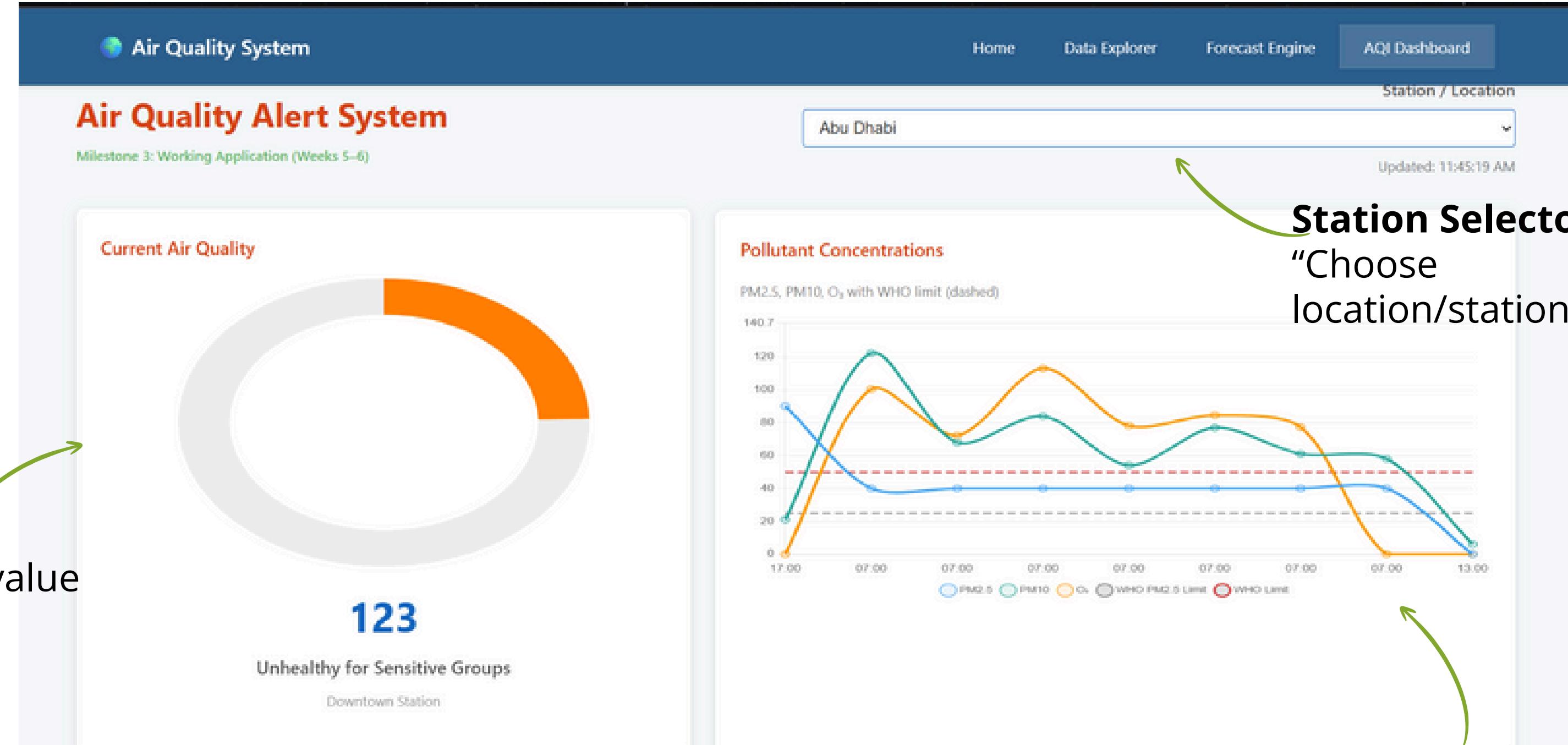
Dashboard Features

- Current Air Quality gauge showing real-time AQI value
- Live pollutant readings (PM2.5, PM10, O₃, etc.) with WHO limits
- 7-Day Air Quality Forecast with daily AQI category
- Active Alerts panel showing warnings for high pollutant levels
- Primary pollutant detection (e.g., O₃ or PM2.5)
- Health advisory messages for sensitive groups
- Location selector to switch between stations
- Last updated timestamp for real-time monitoring

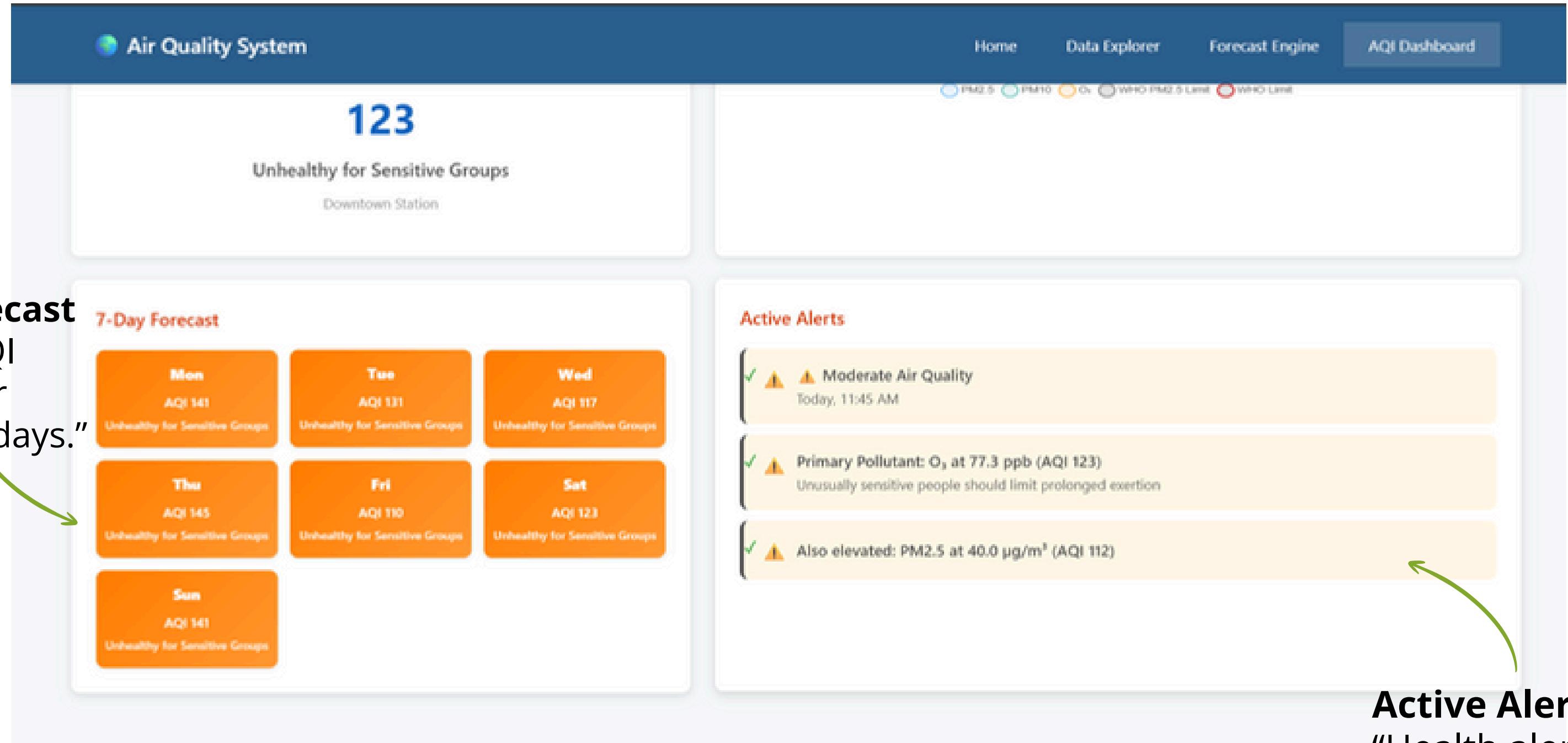
Tools & Technology used

- HTML – dashboard structure
- CSS – UI design, layout & color coding
- JavaScript – interactivity & alert logic
- Chart.js – pollutant line chart & AQI gauge
- Flask (Python backend) – serves real-time AQI data & alerts
- Machine Learning Output – used for 7-day AQI forecast
- JSON / CSV – data exchange format
- AQI Standard Categories (WHO limits)

Dashboard 3 : AQI Dashboard & Alert System



Dashboard 3 : AQI Dashboard



Active Alerts
"Health alerts for elevated pollutants."

Dashboard 4 : Streamlit Dashboard

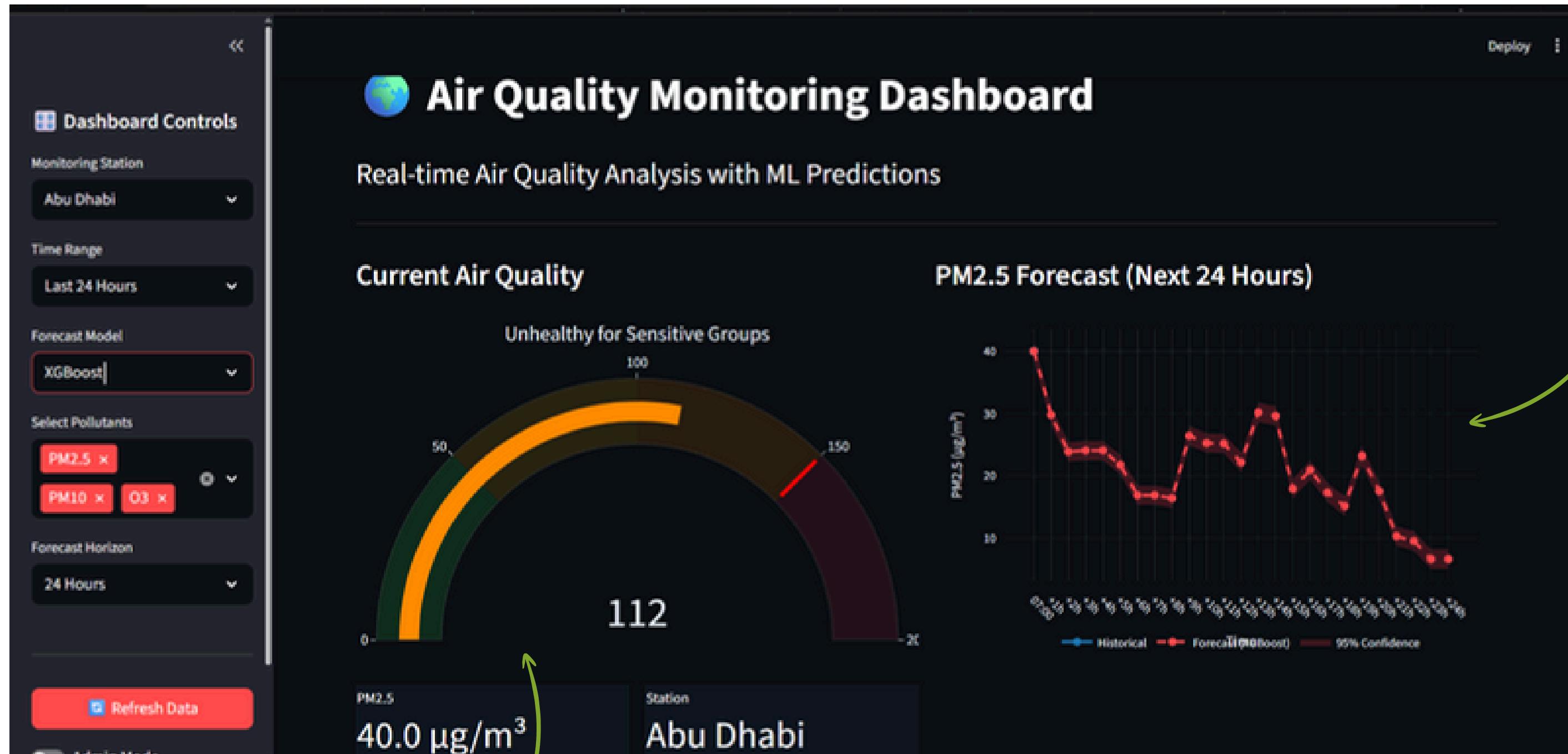
Dashboard Features

- **Interactive Filters Sidebar** : Users can select station, time range, pollutants, forecast model, and forecast horizon.
- **Real-Time AQI Gauge** : Displays current PM2.5-based AQI with color-coded categories (Good / Moderate / Unhealthy / etc.).
- **PM2.5 Forecast (Next 24 Hours)** : Uses selected ML model, shows: Historical vs forecast curve, Confidence interval shading, etc.
- **Weekly Pollutant Trends** : Line graphs for selected pollutants (PM2.5, PM10, NO₂, O₃, SO₂, CO).
- **Smart Alert Notifications** : Auto-generated alerts based on AQI severity
- Admin Panel (Advanced) : Dataset summary & statistics, Model performance visualization, System health indicators, Upload/replace CSV files (main, predictions, forecast), Data integrity checks, Download buttons

Tools and Technology used

- **Streamlit** – UI + dashboard framework
- **Python** – backend logic
- **Pandas, NumPy** – data processing
- **Plotly** – gauge, line charts, forecast graphs
- **Machine Learning Models**: LSTM, ARIMA, Prophet, XGBoost
- **Joblib** – reading model metadata
- **CSV Files** – dataset, predictions, forecast
- **Custom CSS** – dark theme & styling

Dashboard 4 : Streamlit Dashboard



Sidebar Filters
"Station, time range, pollutants, model selection."

AQI Gauge

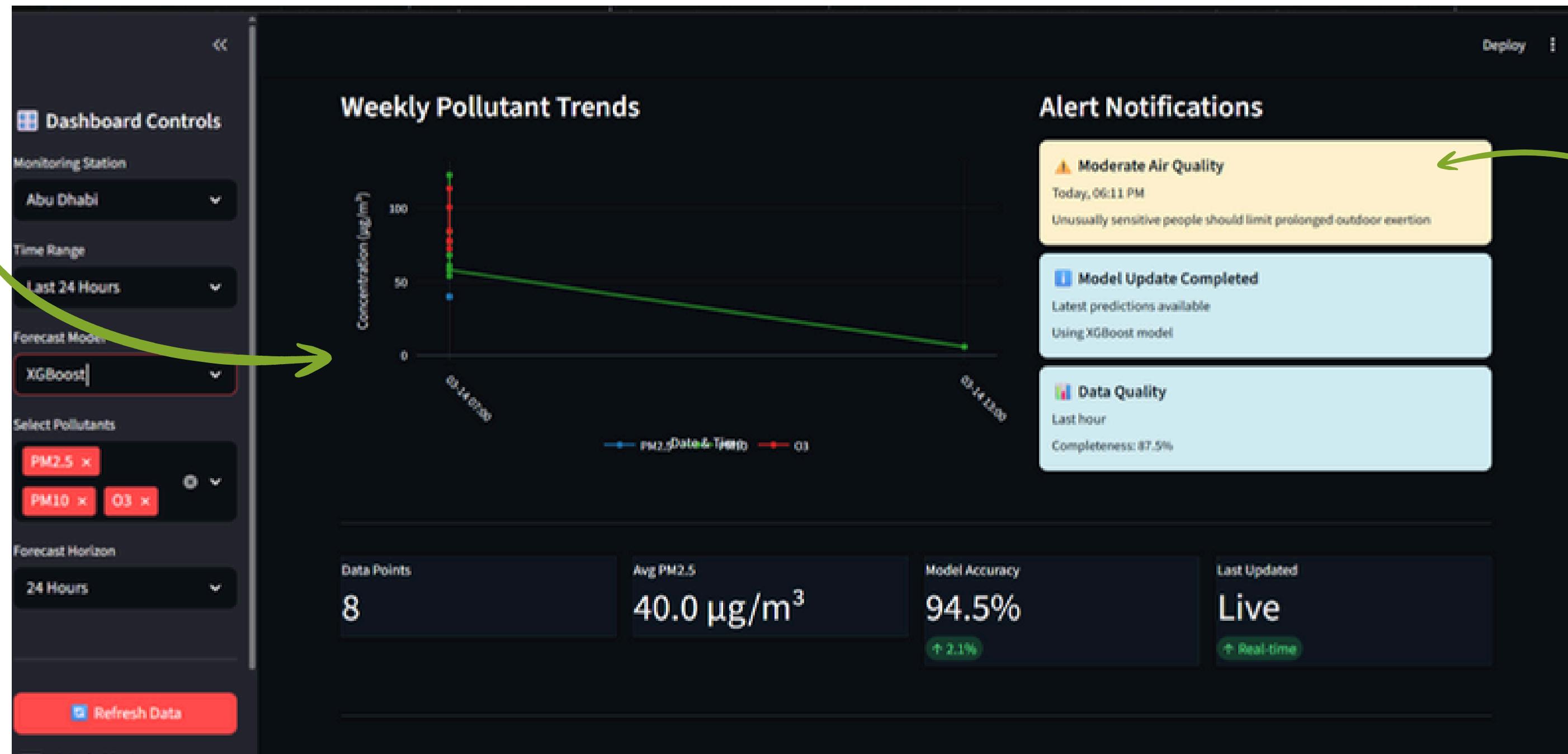
"Real-time AQI with category & color coding."

Forecast Graph
"Next 24-hour PM2.5 forecast with confidence interval."

Dashboard 4 : Streamlit Dashboard

Weekly Trends

"Last 7-day pollutant comparison (PM, NO₂, O₃...)."



Alerts Box
"Auto-generated health alerts based on AQI level."

Dashboard 4 : Streamlit Dashboard

Top Tabs : Switch between Data Summary / Model Performance / System Health / Data Management

Admin mode toggle on

Select Data Type or upload csv files

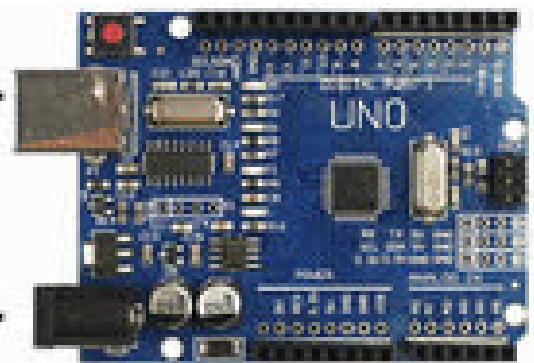
The screenshot shows the Streamlit Admin Panel interface. On the left, there's a sidebar with a 'Forecast Horizon' dropdown set to '24 Hours', a 'Refresh Data' button, and an 'Admin Mode' toggle switch which is turned on. Below that is a section for 'Upload Data' with a 'Select Data Type' dropdown set to 'Main Air Quality Data' and a file upload area for CSV files. A green arrow points from the 'Admin mode toggle on' text to the 'Admin Mode' switch. Another green arrow points from the 'Select Data Type or upload csv files' text to the 'Upload Data' section. The main content area is titled 'Admin Panel' and contains tabs for 'Data Summary' (which is active), 'Model Performance', 'System Health', and 'Data Management'. Under 'Data Summary', there's a section for 'Current Dataset Statistics' with a table showing various metrics like count, mean, min, max, etc., for columns BC, CO, NO, NO2, NOX, O3, PM1, PM10, PM2.5, RELATIVEHUMIDITY, SO2, TEMPERATURE, UMO3, and year. Below that is a 'Data Info' section with 'Total Rows' (8), 'Total Columns' (28), and 'Missing Values' (1).

	Last Updated	BC	CO	NO	NO2	NOX	O3	PM1	PM10	PM2.5	RELATIVEHUMIDITY	SO2	TEMPERATURE	UMO3	year
count	8	8	8	8	8	8	8	8	8	7	8	8	8	8	8
mean	2024-03-14 07:45:00	3.4034	118.308	1.43	22.4162	16.4025	65.6424	11	66.3103	40	35.5531	11.8913	19.0871	122.352	2024
min	2024-03-14 07:00:00	3.4034	90	1	6.08	13	-0.999	11	6.13	40	35.5531	3.37	19.0871	122.352	2024
25%	2024-03-14 07:00:00	3.4034	122.352	1	9.095	13	54.2395	11	57	40	35.5531	7.8125	19.0871	122.352	2024
50%	2024-03-14 07:00:00	3.4034	122.352	1	23.665	13	77.715	11	64.5	40	35.5531	10	19.0871	122.352	2024
75%	2024-03-14 07:00:00	3.4034	122.352	1	28.2575	13	88.6275	11	78.75	40	35.5531	15.8275	19.0871	122.352	2024
max	2024-03-14 13:00:00	3.4034	122.352	4.44	52.67	49.22	113.24	11	122.352	40	35.5531	24.14	19.0871	122.352	2024
std	None	0	11.4382	1.2162	15.5566	9.6237	42.9745	0	32.6041	0	0	6.823	0	0	0

Data Info

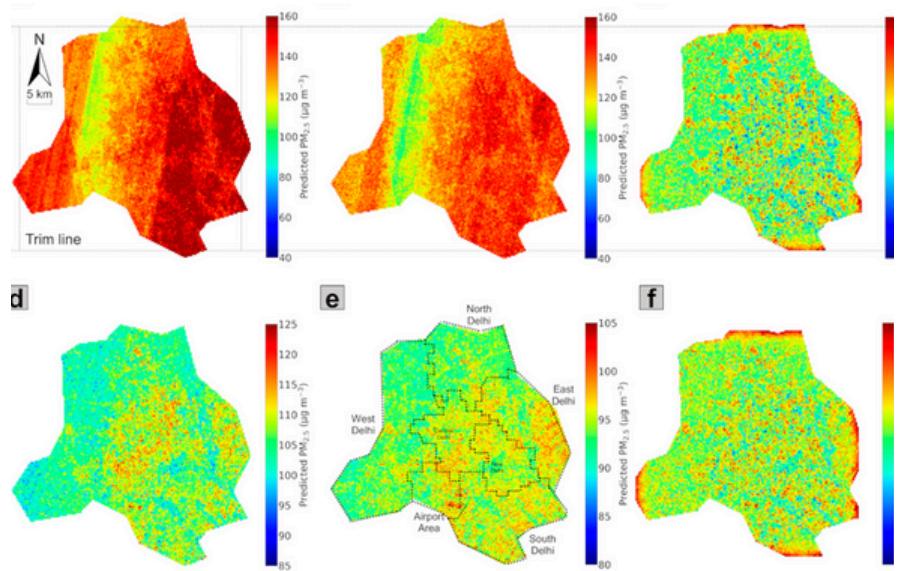
Total Rows	Total Columns	Missing Values
8	28	1

Future Enhancements



Real-Time Sensor

Connect live air-quality sensors to update AQI and pollutant levels automatically.



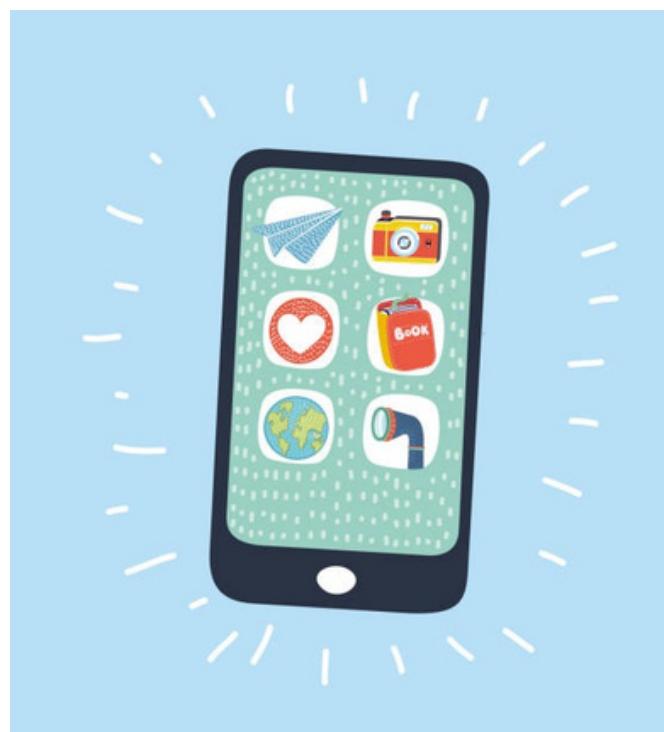
Predictive Hotspot Detection

Use ML to identify future pollution hotspots (areas likely to become highly polluted in coming hours/days), helping authorities take early action.



Deploy as a Public Web Portal with User Profiles

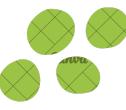
Allow users to create accounts, save locations, set alert preferences, and view personalized dashboards.



Mobile App

Create a mobile app that sends instant notifications when AQI becomes unhealthy.

Project Execution Guide



Steps to run the project

1. Download the Project Folder

👉 Download the complete folder: **all_dashboards**

2. Extract the Folder

👉 Unzip the file

👉 Open the folder in your system

3. Requirements

Make sure dependencies are installed:

pip install -r requirements.txt

4. Run the Project

👉 **Terminal 1** — Start Flask Backend

Run this inside the project folder: **python app.py**

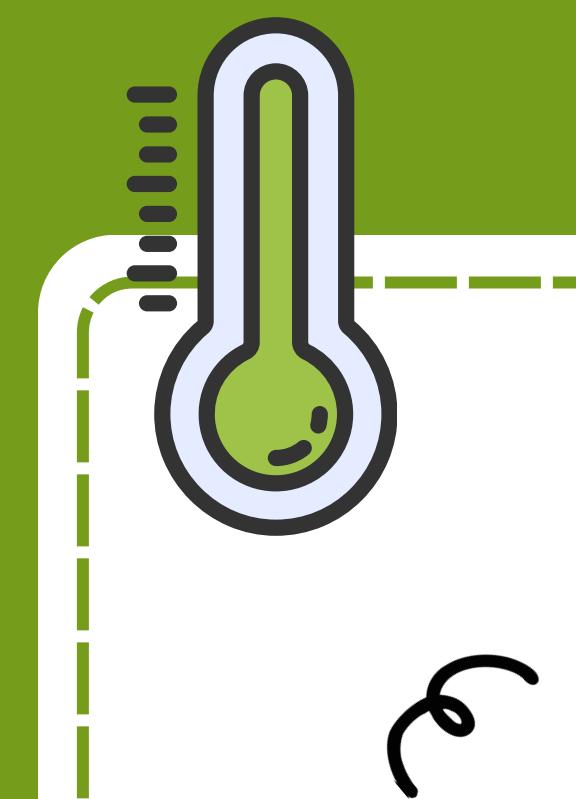
👉 **Terminal 2** — Start Streamlit Dashboard

Run this inside the same folder: **python -m streamlit run streamlit_dashboard.py --server.port=8501 --server.address=127.0.0.1**

Cithub Links

- The complete project folder **all_dashboards** is provided as part of the submission.
- [All Dashbords Folder](#)

- **Word Air Quality Dataset Kaggle link**
- [Dataset Link](#)



Thank You

