

Project Title	Live PM2.5 Nowcast and Forecast with Streamlit
Skills take away From This Project	<ul style="list-style-type: none"> <li>• API integration (OpenAQ, Open-Meteo)</li> <li>• Data preprocessing and feature engineering</li> <li>• Time series forecasting (recursive prediction)</li> <li>• Anomaly detection using Isolation Forest</li> <li>• Model building with Scikit-learn</li> <li>• Streamlit app development for ML visualization</li> <li>• Data quality validation and monitoring</li> </ul>
Domain	Environmental Data Science Air Quality & Pollution Monitoring Time Series Forecasting Applied Machine Learning

### Problem Statement:

Design a system that fetches **live PM2.5 air quality data** and **weather features**, cleans and processes it, and builds an ML model to forecast **next 24–48 hours PM2.5 concentration**. Provide anomaly detection and a user-friendly visualization through Streamlit.

### Business Use Cases:

**Public Health Monitoring:** Early warning for pollution spikes.

**Smart City Applications:** Traffic control and urban planning.

**Environmental Policy:** Data-driven air quality policy decisions.

**IoT Integration:** Edge-based PM2.5 forecast in air filters, purifiers.

**Climate Research:** Studying urban air quality patterns and weather correlation.

**Approach:**

**Data Ingestion:** Pull PM2.5 via OpenAQ (geo-radius) or fallback Open-Meteo Air Quality API.

**Weather Features:** Fetch hourly weather (temperature, humidity, wind, pressure).

**Feature Engineering:** Create lag features, rolling statistics, and interactions.

**Data Quality & Anomaly Detection:** Use IsolationForest to flag unusual readings.

**Modeling:** Train Gradient Boosting Regressor to predict PM2.5.

**Recursive Forecasting:** Generate forecasts up to 48 hours.

**Visualization:** Build Streamlit UI with metrics, charts, and feature importance.

**Results:**

An interactive Streamlit dashboard showing:

- **Last observed PM2.5**
- **Next hour forecast**
- **24–48h prediction line chart**
- **Anomaly indicators**
- **Feature importance ranking**

Validated model performance with **MAE (Mean Absolute Error)**.

**Project Evaluation metrics:**

**MAE (Mean Absolute Error)** on validation set.

Coverage % of valid PM2.5 readings.

Accuracy of anomaly detection.

Forecast horizon reliability (number of valid steps).

Code quality and adherence to project guidelines.

### Technical Tags:

Python Streamlit Machine Learning Scikit-learn Air Quality OpenAQ Open-Meteo Time Series Forecasting Environmental Data Science

### Data Set:

**Source:** OpenAQ API (PM2.5), Open-Meteo Weather API, Open-Meteo Air Quality API (fallback).

**Format:** JSON (converted to Pandas DataFrame).

### Variables:

- PM2.5 concentration ( $\mu\text{g}/\text{m}^3$ )
- Temperature ( $^{\circ}\text{C}$ )
- Relative Humidity (%)
- Wind Speed (m/s)
- Wind Direction ( $^{\circ}$ )
- Pressure (hPa)
- Engineered features: lags, rolling means/stds, anomaly flags

### Data Set Explanation:

PM2.5 values are aggregated hourly.

Missing values filled with forward-fill.

Outliers flagged via anomaly detection.

Weather data synchronized with PM2.5 timeline.

Final dataset prepared with engineered features for ML.

### Project Deliverables:

app.py (Streamlit source code)  
requirements.txt (dependencies)  
Documentation (this project report)  
Screenshots of working dashboard (optional)

### Project Guidelines:

Use **version control (Git)** for commits.

Maintain **PEP8 coding standards**.

Document functions with comments.

Use caching (`st.cache_data`) to avoid repeated API calls.

Modularize code (data fetch, feature engineering, modeling).

### Timeline:

**Week 1:** Setup environment, fetch API data, preprocess.

**Week 1:** Feature engineering, anomaly detection, initial modeling.

**Week 2:** Implement recursive forecasting, validate model.

**Week 2:** Build Streamlit dashboard, polish UI.

**Final Submission:** Code + documentation + working app demo.

## PROJECT DOUBT CLARIFICATION SESSION ( PROJECT AND CLASS DOUBTS)

**About Session:** The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.

**Note: Book the slot at least before 12:00 Pm on the same day**

**Timing: Monday-Saturday (4:00PM to 5:00PM)**

**Booking link :** <https://forms.gle/XC553oSbMJ2Gcfug9>

**For DE/BADM project/class topic doubt slot clarification session:**

**Booking link :** <https://forms.gle/NtkQ4UV9cBV7Ac3C8>

**Session timing:**

**For DE: 04:00 pm to 5:00 pm every saturday**

**For BADM 05:00 to 07:00 pm every saturday**

## LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

**About Session:** The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

**Note: This form will Open only on Saturday (after 2 PM ) and Sunday on Every Week**

**Timing:**

**For BADM and DE**

**Monday-Saturday (11:30AM to 1:00PM)**

**For DS and AIML**

**Monday-Saturday (05:30PM to 07:00PM)**

**Booking link :** <https://forms.gle/1m2Gsro41fLtZurRA>



