Project Guide:

Title: Real-Time AQI Visualizer and Emission Pattern Validation System

---

📌 Overview

This project builds a system to continuously monitor air quality using two data sources:

- Official AQI data from public APIs (like OpenWeather, Data.gov.in)
- AI-estimated particulate matter (PM2.5, PM10) values derived from haze levels in real-time surveillance images.

Both datasets are stored in a cloud database and visualized on a live map. Discrepancies are flagged when actual air pollution and reported industry AQI data don't match patterns.

📊 System Workflow

1️⃣ Real-Time AQI Data Collection:

- Use APIs from OpenWeather and Data.gov.in to fetch real-time PM2.5 and PM10 values at defined intervals (e.g., every 15 minutes).
- Store this data in **Supabase** along with timestamp and location.

2️⃣ Image Data Collection for AI Model Training:

- Place an AQI sensor (capable of measuring PM2.5 and PM10) near a fixed-position surveillance camera.
- At regular intervals, capture an image and simultaneously log the sensor's PM values.

- Each image and reading must share the **same timestamp**.
- Example:
  `image_20250701_1300.jpg` → PM2.5: 54 µg/m³, PM10: 89 µg/m³
- Collect data under different conditions (day/night, clear/hazy) over multiple days.

3️⃣ Train a Haze Detection AI Model:

- Use **Python + OpenCV** to preprocess images (resize, normalize, augment).
- Design a **CNN (Convolutional Neural Network) regression model** using PyTorch or TensorFlow.
- Train the model to predict PM2.5 and PM10 values from haze intensity in images.
- Track performance using metrics like **Mean Absolute Error (MAE)** and **correlation coefficient (R^2)**.
- Save the trained model for deployment.

4️⃣ Deploy and Store Data in Supabase:

- Store AI-predicted PM values and official API data in a **Supabase** table with timestamps.
- Use Python Supabase SDK (`supabase-py`) for database communication.

5️⃣ Compare and Correlate Both Data Sources:

- Regularly retrieve both datasets.

- Perform correlation analysis using **Pearson Correlation** and **Dynamic Time Warping (DTW)**.

- Identify deviations where factory data trends do not match public or AI-estimated patterns.

- Set thresholds (e.g., if correlation falls below 0.6 over a 1-hour window, trigger an alert).

6️⃣ Live Map Visualization:

- Use **Leaflet.js** to display AQI data points on a map.

- Show color-coded markers for AQI levels.

- On marker click, show both official and AI-estimated values, pollutant types, and any flagged discrepancies.