Capstone Project: Concept Note and Implementation Plan

Project Title: Predicting PM2.5/AQI for Chiang Rai Using Weather & Seasonal Data

Team Members:

- Khant Nyar Ko Ko
- Myo Zin Thant
- Kay Khine Maw
- Swan Htut Oakkar Aung

Concept Note

1. Project Overview

This project aims to address the significant public health challenge of severe air pollution in Chiang Rai, Thailand, which is often exacerbated by seasonal agricultural burning. The goal is to develop a machine learning model that can accurately forecast the next-day PM2.5 concentrations and the corresponding Air Quality Index (AQI) category. The project's primary deliverable will be a simple, publicly accessible web dashboard that displays these daily forecasts, providing actionable information to the local community, including students at Mae Fah Luang University. This initiative directly supports several UN Sustainable Development Goals: SDG 3 (Good Health & Well-Being) by providing early warnings to minimize exposure to pollutants, SDG 11 (Sustainable Cities & Communities) by offering data to inform local air quality management, and SDG 13 (Climate Action) by highlighting the link between seasonal activities and pollution events.

2. Objectives

The specific objectives of this project are:

- To develop and train a machine learning model capable of accurately predicting next-day PM2.5 concentrations in Chiang Rai.
- To use meteorological and historical pollution data as the primary inputs for the predictive model.
- To evaluate multiple machine learning algorithms (Linear Regression, Random Forest, XGBoost) to identify the best-performing model.
- To create a lightweight, user-friendly web dashboard to display the daily AQI forecast and provide simple health recommendations.
- To translate complex environmental data into an easily understandable format (e.g., "Good," "Moderate," "Unhealthy") to empower the community to take protective measures.

3. Background

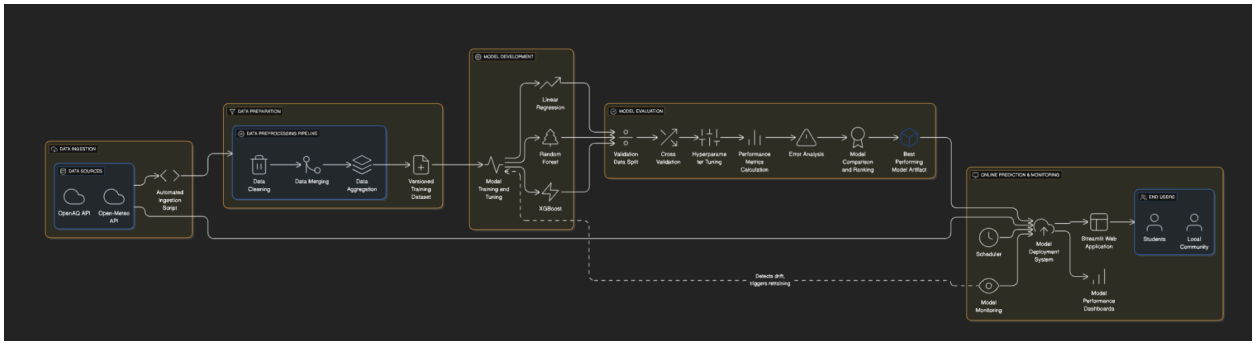
Chiang Rai province in Northern Thailand experiences pronounced seasonal air quality degradation, primarily due to biomass burning, which leads to high concentrations of particulate matter. This poses a direct threat to public health. Previous research in the region has successfully demonstrated the viability of using meteorological data to predict air pollution levels. For instance, a 2021 study by P. S. La-ong-muang et al. used Multiple Linear Regression (MLR) to forecast PM10 concentrations in Chiang Rai, finding a strong correlation with weather variables like temperature and humidity, especially during the summer haze season. While this established a strong precedent, this project will advance this work by focusing on the more hazardous PM2.5 pollutant and employing more sophisticated, non-linear machine learning models. A machine learning approach is necessary to capture the complex, non-linear interactions between weather patterns and pollution that linear models cannot fully represent, with the goal of achieving higher predictive accuracy.

4. Methodology

The project will employ classical machine learning techniques for time-series forecasting. The core methodology involves training models to predict a future value (next-day PM2.5) based on a sequence of past observations (historical weather and pollution data).

- **Models:** We will implement three models for comparative evaluation:
    1. **Linear Regression:** To serve as a simple, interpretable baseline and for direct comparison with previous regional studies.
    2. **Random Forest:** An ensemble model known for its robustness and ability to handle non-linear relationships.
    3. **XGBoost (Extreme Gradient Boosting):** A state-of-the-art gradient boosting model renowned for its high accuracy and performance, which will be our primary advanced model. We will also implement a hybrid trend-residual modeling strategy with XGBoost to better handle time-series trends, where a linear model captures the long-term trend and XGBoost models the remaining fluctuations.
- **Evaluation:** The regression task (predicting PM2.5 concentration) will be evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The classification task (predicting AQI category) will be evaluated using Accuracy and the F1-Score.

5. Architecture Design Diagram



1. Overall Purpose

This diagram illustrates the complete, end-to-end architecture of the AQI Prediction System. It is organized into distinct logical zones, or "swimlanes," showing the two primary workflows:

1. The "Offline" Training Pipeline: The top-level flow that moves from left to right, showing how raw data is collected, prepared, and used to build and select the best-performing model.
2. The "Online" Prediction Pipeline: The bottom-level flow showing how the trained model is deployed in a live system, triggered by a scheduler, and used to serve daily forecasts to end-users.

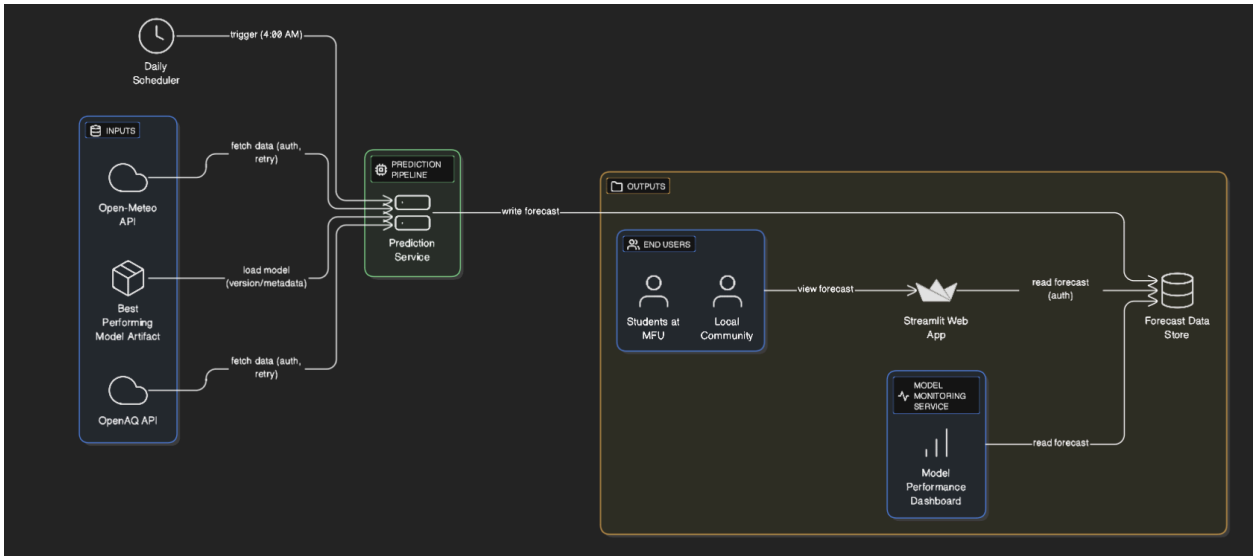
2. Components Explanation:

- **DATA INGESTION:**
    - Purpose: To acquire all necessary raw data from external sources.
    - Components:
        - Data Sources: This includes the OpenAQ API (for historical PM2.5 data) and the Open-Meteo API (for historical weather data).
        - Automated Ingestion Script: A script that runs to fetch the data from both APIs.
- **DATA PREPARATION**:
    - Purpose: To clean, transform, and structure the raw data into a usable format for machine learning.
    - Components:
        - Data Preprocessing Pipeline: This is a sequence of steps.
        - Data Cleaning: Handles missing values (e.g., interpolation) and inconsistencies.
        - Data Merging: Joins the air quality and weather data into a single dataset.
        - Data Aggregation: Transforms hourly data into daily averages.
    - Final Output: Versioned Training Dataset. This is the final, clean dataset stored and versioned to ensure reproducible training.
- **MODEL DEVELOPMENT:**

    - Purpose: To use the clean data to train the three candidate machine learning models.
    - Components:
        - Model Training and Tuning: The main process that takes the Versioned Training Dataset as input.
        - Linear Regression / Random Forest / XGBoost: The three specific models that are trained in parallel to be compared.
- **MODEL EVALUATION:**
    - Purpose: To rigorously and systematically test the trained models to select the single best one.
    - Components (in logical order):
        - Validation Data Split: The training data is split into training and validation sets.
        - Cross Validation: A procedure used to test model performance robustly.
        - Hyperparameter Tuning: The process of optimizing each model's settings to achieve the best performance.
        - Performance Metrics Calculation: The models are scored using the metrics defined in the plan (e.g., MAE, RMSE, F1-Score) .
        - Error Analysis: A manual or automated review of where the models are making mistakes.
        - Model Comparison and Ranking: The final step to objectively rank the models and select the winner.
    - Final Output: Best Performing Model Artifact. This is the final, trained model file (e.g., model.pkl) that is saved for deployment.
- **ONLINE PREDICTION & MONITORING:**
    - Purpose: This is the live, "production" system that serves daily forecasts to the public.
    - Components:
        - Scheduler: A tool (like cron) that triggers the prediction process automatically every day.
        - Model Deployment System: The "backend" of the application. It loads the Best Performing Model Artifact and fetches live data to generate a new forecast.
        - Streamlit Web Application: The "frontend" or user interface that the public interacts with.
        - Model Monitoring: A service that tracks the live model's performance to detect drift .
        - Model Performance Dashboards: A visual dashboard for the project team to see the live model's accuracy and health.
- **END USERS:**
    - Purpose: The final stakeholders and consumers of the project's deliverable.
    - Components: This includes `Students` (at Mae Fah Luang University) and the `Local Community` in Chiang Rai.

3. Data and Logic Flow (Interactions)

1. **The "Offline" Training Flow:** This is the main horizontal path.
    - Data Sources are fed into the Automated Ingestion Script.
    - The script's output goes to the Data Preprocessing Pipeline.
    - The pipeline's output is the Versioned Training Dataset.
    - This dataset is the input for Model Training and Tuning.
    - The three trained models are then fed into the Model Evaluation pipeline.
    - The evaluation pipeline's final output is the Best Performing Model Artifact. This "bridge" artifact is then passed to the Model Deployment System in the online pipeline.
2. **The "Online" Prediction Flow:** This path shows the live system in action.
    - **Trigger:** The Scheduler sends a daily trigger to the Model Deployment System.
    - **Data Ingestion:** The Model Deployment System simultaneously fetches *live data* from the Data Sources (OpenAQ/Open-Meteo).
    - **Prediction:** The Model Deployment System loads the Best Performing Model Artifact, combines it with the live data, and generates a new forecast.
    - **Display:** The Streamlit Web Application reads this new forecast from the deployment system and displays it to the End Users.
    - **Monitoring:** The Model Monitoring service also observes the Model Deployment System's predictions to populate the Model Performance Dashboards.
3. **The "Monitoring & Retraining" Feedback Loop:**
    - The dashed line shows the Model Monitoring service sending a signal "Detects drift, triggers retraining" back to the Model Training and Tuning process. This completes the MLOps lifecycle, ensuring that when the live model's performance degrades, the system can automatically trigger a new model to be trained on the latest data.



1. Overall Purpose

This diagram provides a "zoomed-in" view of the "Online Prediction Pipeline." Its purpose is to show the architecture of the live, deployed system. It illustrates how the model *created* in the offline pipeline is *used* to generate a fresh forecast every day and serve it to the public .

2. Components Explanation

This architecture is brilliantly designed with a decoupled (separated) approach, which is a best practice.

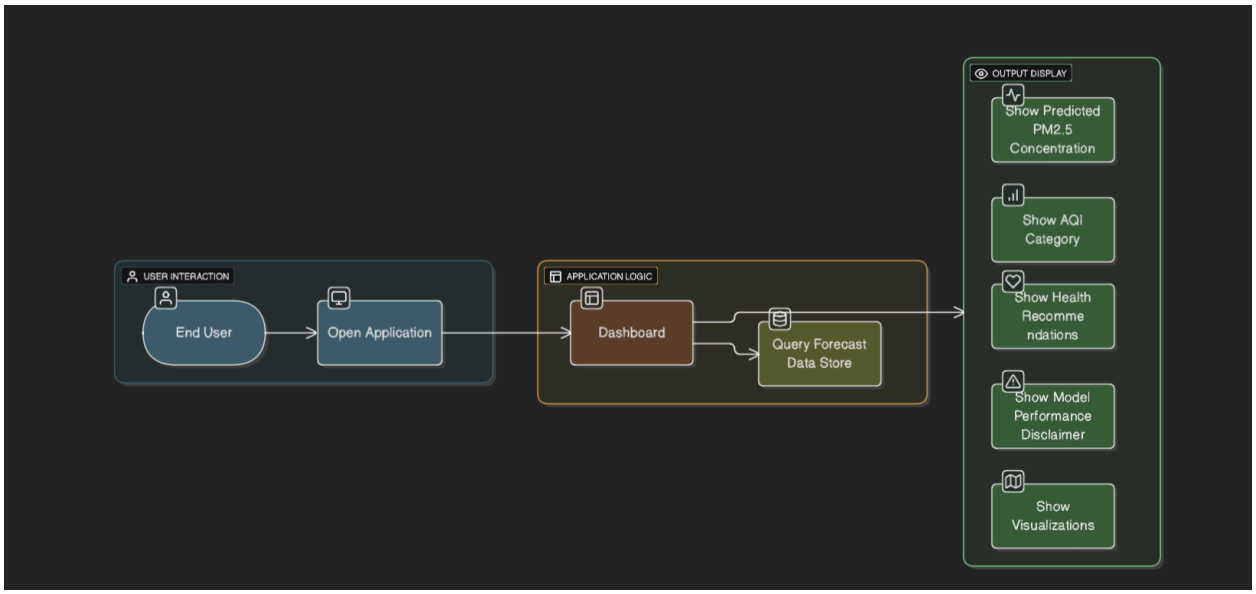
- **Daily Scheduler:** This is the trigger for the entire system. It is an automated process (like a cron job) set to run once per day at a specific time (4:00 AM) to ensure the forecast is ready before users wake up.
- **INPUTS:** These are the three "ingredients" the system needs to make a new prediction.
    - **Best Performing Model Artifact:** This is the final, trained model file (e.g., model.pkl) that was the output of the offline training pipeline. The "version/metadata" label indicates the system tracks *which* model is being used.
    - **Open-Meteo API & Open-AQ API:** These are the sources for *live, real-time data* (not the historical data used for training). The "(auth, retry)" label shows the system is robust, handling API authentication and automatically retrying if a network call fails, as planned in your mitigations .
- **PREDICTION PIPELINE:**
    - **Prediction Service:** This is the "engine" or "backend" of the system. It's a script that is activated by the Scheduler. Its sole job is to load the three inputs (model, live weather, live AQI), run the prediction, and save the result.
- **OUTPUTS:** This section contains all the "consumers" of the new forecast.
    - Forecast Data Store: This is the central hub and the most critical part of this design. It is a database or file that stores the new forecast. By having the

Prediction Service *write* to this store, and all other applications *read* from it, you have decoupled your system. This makes the web app fast and reliable, as it doesn't have to wait for the model to run.

- Streamlit Web App: This is the user-facing dashboard (the "frontend") .
- END USERS : The final stakeholders (Students at MFU and the Local Community) who will view the dashboard.
- Model Monitoring Service: An internal service that tracks the live model's performance by checking its predictions in the data store .
- Model Performance Dashboard: The visual tool for your team to see the monitoring service's results and check for model drift.

3. Data and Logic Flow (Interactions)

1. Trigger: The entire process begins at 4:00 AM when the Daily Scheduler sends a "trigger" signal to the Prediction Service.
2. Activation: The Prediction Service "wakes up" and simultaneously pulls in its three inputs:
    - It "loads model (version/metadata)" from the Best Performing Model Artifact.
    - It "fetches data (auth, retry)" from both the Open-Meteo API and Open-AQ API.
3. Generation: Once it has the model and the new data, the Prediction Service generates the next-day forecast and "writes" this single new prediction into the Forecast Data Store.
4. Consumption (User-Facing): At any time of day, an End User can "view forecast" by opening the Streamlit Web App. The app, in turn, "reads" the pre-computed forecast from the Forecast Data Store. This ensures the user gets an instant response.
5. Consumption (Internal): In parallel, the Model Monitoring Service also "reads" the forecast from the data store to update the Model Performance Dashboard, allowing your team to monitor the system's health without affecting the user's experience.



1. Overall Purpose

This diagram illustrates the User Interface (UI) and User Experience (UX) Flow of the Streamlit web application. It "zooms in" on the Streamlit Web App component from the previous diagrams to show exactly what happens when an end-user visits the dashboard and what information they are presented with .

2. Components Explanation

- USER INTERACTION:
    - Purpose: This represents the user and the initial action they take to begin the process.
    - Components:
        - End User: The individual (e.g., local community member, student at Mae Fah Luang University) who wants to know the air quality forecast.
        - Open Application: This is the user's action of navigating to the web dashboard's URL in their browser.
- APPLICATION LOGIC:
    - Purpose: This represents the "backend" logic of the Streamlit application itself as it processes the user's request.
    - Components:
        - Dashboard: The main application page that the user sees.
        - Query Forecast Data Store: The specific function the Dashboard runs *immediately* upon loading. It fetches the latest data from the Forecast Data Store (which we identified in the previous "Online Pipeline" diagram).
- OUTPUT DISPLAY:

- Purpose: This represents all the visual components on the dashboard that present the final, translated information to the user. This is the primary deliverable of the project.
- Components: This is a list of all the information blocks the user will see:
  - Show Predicted PM2.5 Concentration: The specific numerical forecast.
  - Show AQI Category: The translated, easy-to-understand category (e.g., "Good," "Moderate," "Unhealthy") .
  - Show Health Recommendations: The actionable guidance (e.g., "Limit outdoor activity").
  - Show Model Performance Disclaimer: An important ethical component to be transparent with users about the model's accuracy and uncertainty.
  - Show Visualizations: A simple graph, gauge, or color-coded card to help the user quickly understand the data.

3. Data and Logic Flow (Interactions)

1. Start: The entire process begins when the End User takes the action to Open Application.
2. Query: This action immediately loads the Dashboard. The Dashboard's first and only logic step is to Query Forecast Data Store to retrieve the latest pre-computed prediction.
3. Display: The Dashboard then uses the data from that query to populate all the components in the OUTPUT DISPLAY section, presenting a complete and easy-to-understand forecast to the End User.

6. Data Sources

The project will utilize two primary, publicly accessible data sources, which will be merged into a single time-series dataset. Air quality data, specifically historical PM2.5 concentrations for Chiang Rai, will be programmatically acquired from the OpenAQ platform via its REST API. Corresponding meteorological data—including temperature, relative humidity, wind speed, wind direction, precipitation, and atmospheric pressure—will be sourced from the Open-Meteo API, which provides extensive historical weather data without requiring an API key for non-commercial use. The raw hourly data will be aggregated to daily averages, and a preprocessing pipeline will be implemented to merge the datasets, handle missing values through interpolation, and normalize numerical features for model training.

7. Literature Review

The project's methodology is grounded in existing regional air quality research. A key precedent is the work of P. S. La-ong-muang et al. (2021), which successfully used Multiple Linear Regression and meteorological data to predict PM10 concentrations in Chiang Rai. Their findings confirmed a strong seasonal dependency and the predictive power of local weather variables, validating the core approach of this project. Our work extends this foundation by focusing on the more hazardous PM2.5 pollutant and, most significantly, by employing

advanced, non-linear machine learning models like Random Forest and XGBoost. This methodological shift is intended to capture the complex, non-linear atmospheric dynamics that linear models may miss, with the hypothesis that these more sophisticated techniques will yield a significant improvement in predictive accuracy and provide a more nuanced understanding of pollution drivers in the region.

Implementation Plan

1. Technology Stack

- **Programming Language:** Python
- **Data Manipulation & Analysis:** Pandas
- **Machine Learning Libraries:** Scikit-learn (for Linear Regression and data preprocessing), XGBoost
- **Data Acquisition:** py-openaq (for OpenAQ API), Python Requests library (for Open-Meteo API)
- **Web Application Framework:** Streamlit
- **Version Control:** Git / GitHub

2. Timeline

The project is planned over a 7-week period. The timeline and task breakdown are visualized in the Gantt chart below.

Gantt Chart

Phase	Task	Duration (Weeks)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
1. Project Planning	**Define Scope & Objectives**	**1**	■						

	Literature & Tech Review	**1**	■						
2. Data Management	**Develop Data Ingestion Scripts**	**1**		■					
	Data Collection & Cleaning	**2**		■	■				
	Exploratory Data Analysis	**1**			■				
3. Model Development	**Implement Baseline (Linear Reg.)**	**1**				■			
	Implement Advanced Models (RF, XGBoost)	**2**				■	■		
	Hyperparameter Tuning	**1**					■		

4. Evaluation	**Model Evaluation & Selection**	**1**						■	
	Final Report Writing	**2**						■	■
5. Deployment	**Develop Streamlit Dashboard**	**2**						■	■
	Final Presentation Prep	**1**							■

Task Distribution Matrix (RACI)

- **R** - Responsible: The person who does the work.
- **A** - Accountable: The person who owns the task.
- **C** - Consulted: Provides input and feedback.
- **I** - Informed: Kept up-to-date on progress.

Task	Khant Nyar Ko Ko	Myo Zin Thant	Kay Khine Maw	Swan Htut Oakkar Aung
Project Planning & Scope	A	R	C	C
Literature & Tech Review	R	A	I	I
Data Ingestion & Collection	C	R	A	R
Data Cleaning & EDA	I	A	R	R
Baseline Model Implementation	R	C	R	A
Advanced Model Implementation	A	R	R	C
Model Evaluation & Selection	R	R	A	C
Streamlit Dashboard Dev.	C	A	C	R

Final Report & Presentation	R	R	R	A

3. Milestones

- **Week 2:** Completion of data ingestion scripts and initial data collection.
- **Week 4:** Completion of data preprocessing and exploratory data analysis.
- **Week 5:** Baseline and advanced models implemented and initial training completed.
- **Week 6:** Final model selected after comprehensive evaluation.
- **Week 7:** First functional prototype of the Streamlit dashboard is live; final project report, presentation, and codebase are submitted.

4. Challenges and Mitigations

- **Challenge: Data Quality and Availability.** APIs may have downtime, or the collected data may have significant gaps or inconsistencies.
    - **Mitigation:** We will implement robust error handling in our data ingestion scripts. For missing data, we will use appropriate imputation techniques, such as linear interpolation or forward-fill, to maintain the time-series integrity.
- **Challenge: Model Performance.** The initial models may not achieve the desired level of accuracy for a reliable forecast. Tree-based models are also inherently unable to extrapolate beyond the range of training data, which is a risk for time-series with trends.
    - **Mitigation:** We will perform careful feature engineering to extract more predictive signals from the data. The hybrid trend-residual modeling approach is specifically designed to mitigate the extrapolation issue. Extensive hyperparameter tuning will be conducted to optimize the final model.
- **Challenge: Technical Constraints.** We may encounter API rate limits or require significant computational resources for model training.
    - **Mitigation:** We have selected Open-Meteo for its generous free tier (10,000 calls/day), which should be sufficient. We will start training on smaller subsets of data to iterate quickly and will leverage efficient libraries like XGBoost, which is optimized for performance.

5. Ethical Considerations

- **Data Bias and Equity:** Air quality monitoring stations may not be evenly distributed, potentially leading to models that are more accurate for some areas than others. This could create a data gap for vulnerable or under-resourced communities. We will mitigate this by being transparent about the locations of the data sources and acknowledging this as a limitation. The project's goal is to identify pollution trends, not to stigmatize specific neighborhoods.
- **Communication of Risk:** Providing forecasts about public health risks can induce anxiety or "warning fatigue" if not communicated responsibly. Our web dashboard will pair forecasts with clear, simple, and actionable guidance. We will also be transparent about the model's performance and uncertainty to avoid presenting the forecast as an infallible prediction.
- **Data Privacy:** This project exclusively uses open-source, aggregated data from public monitoring stations. No personally identifiable information is collected or used, so individual data privacy is not a concern.
- **Scientific Integrity:** We commit to being transparent about our methodology, data sources, and model performance. The project will be fully documented to ensure our results are reproducible and can be scrutinized by others.

6. References

- La-ong-muang, P. S., et al. (2021). Particulate Matter (PM10) Prediction Based on Multiple Linear Regression: A Case Study in Chiang Rai Province, Thailand. *National Center for Biotechnology Information*.
- OpenAQ. (2025). OpenAQ Platform. Retrieved from https://openaq.org/
- Open-Meteo. (2025). Open-Meteo Weather API. Retrieved from https://open-meteo.com/