PROJECT LOG - AIRAWARE SMART AIR QUALITY PREDICTION SYSTEM

-by N.N. Sai Sreya

1.DAY 1

Topic: AirAware Smart Air Quality Prediction System

1.1 Overview:

AirAware is a smart air quality monitoring and prediction system that uses real-time pollution data and machine learning. It focuses on predicting PM2.5, CO2, NO2 levels and giving awareness about hazardous air conditions. The project includes frontend, backend, database, dashboards, and ML integration.

1.2 Objectives:

- Collect dataset containing PM2.5, CO2, NO2, etc.

- Forecast air quality using ML models such as Linear Regression, Random Forest, SVM.

- Compare Predicted vs Actual values.

- Show 3-4 months of data with analytics on a dashboard.

- Alert users if air quality becomes hazardous.

- Use Kaggle datasets or similar sources.

- Preprocess data (cleaning, normalization) and store in database.

- Build UI + API-based full-stack application.

1.3 Milestones:

1st – 25%

2nd – 50%

3rd – 75%

4th – 100%

Each milestone must include a working demo.

2. DAY 2

2.1 Team Formation:

Team 1 – N.N.Sai Sreya,Rajalaksmi, Rahul, Lokesh, Divija Nandana.

Dataset chosen: Delhi Air Quality Dataset (Kaggle).

2.2 Key Decisions:

- Build AirAware website to predict future pollution levels.

- Predict air quality for upcoming years and show trends in dashboard.

- React recommended for a beautiful UI dashboard.

- Dashboard to include last year's air quality, current quality, and predictions.

- API calls for communication between frontend and backend.

- Backend receives payload from frontend, processes it, and returns prediction results.

2.3 Additional Features:

- Graph of past few years.

- Current weather display.

- Potential integration with Google weather API.

- Accuracy visualization using heatmaps.

- Dummy sensor data input for now.

- Future enhancement: real sensors for surrounding air quality.

3. DAY 3

3.1 Topics Covered:

- Pandas, NumPy, Matplotlib

- Heatmap for model accuracy

3.2 API Concepts:

Two main Python APIs: Flask and FastAPI

- FastAPI:

  ```
  from fastapi import FastAPI

  app = FastAPI()

  @app.get("/")
  ```

- Flask API:

  ```
  from flask_api import FlaskAPI

  from flask import request

  app = FlaskAPI(__name__)

  @app.route("/", methods=['GET'])
  ```


3.3 Key Points:

- GET method sends payload in header.

- POST method sends payload in body.

- Postman is used to test backend API endpoints.

- API connects frontend and backend through requests and responses.

- Frontend payload structure must be known before writing backend.

- Changing payload breaks API functionality.

- FastAPI supports async, better performance, high concurrency.

3.4 Additional:

- WebSocket, streaming responses.

- Deployment discussions pending.

3.5 Task:

Build basic working version of AirAware dashboard with API connectivity.

4. Day 4

Topic: Git and Version Control

4.1 Commands learned:

- git add . - Adds all new/changed files to staging.

- git add <file> - Adds only the selected file to staging.

- git commit -m "msg" - Saves changes with a message.

- git push - Uploads your commits to the GitHub repo.

- git branch - Shows the branch you are currently on.

- git branch –all - Shows all branches in the project.

- git fetch –all - Gets updates from all remote branches.

- git pull - Fetches and merges latest changes from the remote branch.

- git stash - Temporarily saves uncommitted changes without pushing.

- git checkout -b <branch> - Creates and switches to a new branch.

- git checkout <branch> - Switches to an existing branch.

4.2 Virtual Environment:

- python -m venv venv

- venv/Scripts/Activate

- pip install -r requirements.txt

- deactivate

5. Day 5

Topic: Database Concepts

5.1 DB & DBMS

CRUD operations: Create, Rename, Update, Delete

5.2 Types of Databases:

Structured – MySQL, SQL, PostgreSQL

Unstructured – MongoDB

5.3 Normalization Forms:

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

Concepts: Partial dependency, Transitive dependency

5.4 Keys:

Primary key, Foreign key, Composite key

5.5 Data Types:

- Master data (static)

- Transaction data (dynamic)

5.6 Frontend Pages:

- Analytical Dashboard

- About Page

- Report Page

- Login Page

- Admin Page

5.7 AI Tools:

OpenAI, Gemini, Grok, Llama, Vertex AI

5.8 ML Concepts:

- Supervised (Regression, Classification)

- Unsupervised (Clustering)

- Reinforcement learning

5.9 NLP Concepts:

- Embeddings

- Vectors

- NLTK

6. Day 6
- Project Progress Planning

6.1 Milestone Deadlines:

- 25% on 20-21 November

- 50% by 30 November

- No UI template changes after 30 November

- Final output on 1 December

- Then enhancements begin

6.2 Documentation:

- 45-page detailed project document required

- Each page should include Day and Topic

- Include Git, API, DB, Project overview, ML, and daily logs

6.3 GitHub:

- Push code to main repo

- Team repo for demo

- Personal repo for presentation

- Add teammates as contributors

- Maintain backup copy

6.4 Corporate Culture:

- Learn to communicate freely

- Understand professional workflows

7. Day 7:

7.1 Doubt Clearance & Additional Instructions

Day 7 focused mainly on clarifying students' doubts related to the AirAware project, the backend–frontend workflow, API integration, dataset usage, and Git practices. The instructor addressed questions regarding Python APIs (Flask/FastAPI), UI design guidelines, database selection, and how to structure payloads between frontend and backend.

Additional instructions were also given for the project timeline, including how to approach Milestone-1, how to organize the GitHub repository, and how each team should maintain transparency in UI development and backend logic.

Students were reminded to follow the planned structure, avoid unnecessary UI template changes after deadlines, and ensure the project was progressing toward the 25% milestone target.

On this day, we also created both the team GitHub repository and our individual personal GitHub repositories for code collaboration and documentation.

Personal Github repository: https://github.com/SaiSreya96/saisreya_personal_repo

Team Github repository: https://github.com/SaiSreya96/grp1_team_repo_infosys

8. Day 8

8.1 Preparation for Milestone-1 Presentation

On Day 8, students were given dedicated time to prepare for the Milestone-1 presentation scheduled for the next day. No new topics were taught, as the entire session was focused on getting ready for the first project review.

Since this day was fully allotted for milestone preparation, I utilized the time to work on the essential components required for the presentation. I updated and organized the **log document**, created and refined the **project PowerPoint presentation**, and made the necessary **UI changes** to improve the overall appearance and flow of the AirAware dashboard.

The main purpose of Day 8 was to ensure that the project was in a presentable and polished state for the milestone evaluation, and I worked on documentation, presentation, and UI enhancements accordingly.

9. DAY 9

Milestone 1

10.DAY 10

Milestone 1

11. Day 11

11.1. Artificial Intelligence (AI)

- AI is a set of methods that enables machines to perceive data, perform tasks, and make predictions—tasks that normally need human intelligence.

11.2. Key Subfields of AI

- ML – Machine Learning

- DL – Deep Learning

- NLP – Natural Language Processing

- RL – Reinforcement Learning

- Knowledge-based systems

11.3. Deep Learning (DL)

- DL uses **neural networks** like:

  o RNN

  o CNN