DEP Unified AI Dashboard

Student: Harshali Kadam **Course:** Experiential Learning Lab **Project:** Smart City Dashboard

1. DEMO VIDEO (Recommended)

The easiest way to see the project in action is to watch the included demo video: `final project demo .mov`

This video shows the complete, full-stack application running, including the connection to the live MongoDB database, the real-time dashboard updates, and the Generative AI (Gemini) analyst.

2. Project Components

This project is a full-stack application with three main components:

1. `index.html` : The frontend single-page application (SPA). It's a "fat client" that contains all UI, visualization (Chart.js, Leaflet), and business logic (the "Risk Engine" and AI calls).
2. `api.py` : The backend microservice (Python/Flask). This server's only job is to securely connect to the MongoDB Atlas database, fetch the 12,000+ documents, and serve them to the frontend as a JSON API.
3. **MongoDB Atlas**: The cloud-hosted database (the "single source of truth") that stores the raw city data.

The `system_architecture.md` file provides a more detailed visual breakdown.

3. How to Run (Advanced)

Note: This project *cannot* be run by just double-clicking the `index.html` file. It requires the live `api.py` server to be running.

Prerequisites:

- Python 3 installed
- An internet connection

Step 1: Install Dependencies

Open a terminal and run:

```
pip install Flask flask-pymongo flask-cors
```

Step 2: Launch the Backend API Server

1. In your terminal, navigate to this project folder.

2. Run the server:

```
python3 api.py
```

3. The server will be live and running at `http://127.0.0.1:5001/` . **Leave this terminal running.**

Step 3: Launch the Frontend

1. Open `index.html` in your web browser (like Google Chrome or Safari).
2. The dashboard will load, call the live API, and populate with the real data from MongoDB.

Security Note: The `api.py` file contains the MongoDB connection string. The Gemini API key is in the `index.html` file.