Climate Visualizer Project Report

Climate Visualizer Project Report

Project Overview

Climate Visualizer is a full-stack web application for exploring, analyzing, and visualizing global air quality data.

It consists of:

- A backend (Python, Flask) that serves air quality data and analysis via a REST API.
- A frontend (React) that provides interactive dashboards, charts, and maps for users to explore air quality trends, pollutant composition, temporal patterns, and correlations.

Backend (Flask API)

Location: backend/

Key Features:

- Data Source: Uses a CSV file (global_air_quality_dataset.csv) containing air quality and weather data for various cities and countries.
- Data Cleaning: Loads and cleans the data (removes missing values, sorts by country/city/date).
- API Endpoints:
  - /api/countries - List all countries in the dataset.
  - /api/cities - List cities, optionally filtered by country.
  - /api/pollutants - List all pollutants tracked (e.g., PM2.5, PM10, NO2, SO2, CO, O3).
  - /api/data - Get air quality data, filterable by country, city, pollutant, and date range.
  - /api/city-aqi-trends - Get monthly/yearly AQI trends, pollution spikes, and trend direction for each city.
  - /api/pollutant-composition - For each city, get average, max, and percentage contribution of each pollutant to AQI.
  - /api/temporal-patterns - Analyze AQI by month, weekday/weekend, daily/weekly trends, and pollutant levels.
  - /api/correlation-analysis - Statistical correlations between AQI, pollutants, and weather variables (temperature, humidity, wind speed).
  - /api/forecast - Placeholder for future forecasting features.
- Analysis Functions:
  - Summary statistics for each country/city.
  - AQI trends and pollution spikes detection.

  - Correlation analysis (Pearson/Spearman) between pollutants and weather.

Frontend (React App)

Location: frontend/

Key Features:

- Modern UI: Built with React, styled with Tailwind CSS, and uses charting libraries (Chart.js, react-chartjs-2) and mapping (react-leaflet).
- Navigation: Sidebar with links to Dashboard, City Trends, Pollutant Composition, Temporal Patterns, and Correlation Analysis.
- Global Search: Modal for searching cities, countries, pollutants, and navigating to different analysis screens.

Main Screens/Components:

- Dashboard: Summary cards (average AQI, number of cities, latest date, dominant pollutant), filters, trend charts, and map.
- City Trends: Select a city to view monthly/yearly AQI trends, pollution spikes, and trend direction.
- Pollutant Composition: Select a city to view average, max, and percentage contribution of each pollutant.
- Temporal Patterns: Analyze AQI by month, weekday/weekend, daily/weekly trends.
- Correlation Analysis: View statistical correlations between AQI, pollutants, and weather variables.
- Map Section: Interactive map with city markers colored by AQI.

Data Flow:

- The frontend fetches data from the backend API based on user selections (filters, city, pollutant, etc.).
- Data is visualized using charts and maps, with real-time updates as filters change.

Technologies Used

- Backend: Python, Flask, Pandas, NumPy, SciPy, Flask-CORS
- Frontend: React, Tailwind CSS, Chart.js, react-chartjs-2, react-leaflet, react-router-dom, lucide-react (icons)
- Data: CSV file with air quality and weather data

How It Works (User Flow)

1. User opens the app (frontend).
2. Dashboard loads with summary statistics and a map.
3. User applies filters (country, city, pollutant, date range) to explore specific data.
4. User navigates to different analysis screens:

   - City Trends: See AQI trends and spikes for a city.
   - Pollutant Composition: See which pollutants dominate in a city.
   - Temporal Patterns: See how AQI changes over time and by day type.
   - Correlation Analysis: See how pollutants and weather variables are related.

5. All data and charts are fetched from the backend API, which processes and analyzes the CSV data on demand.

Summary Table

Layer	Technology	Main Purpose/Features
Backend	Flask, Pandas	Serve air quality data, perform analysis, expose REST API
Frontend	React, Tailwind	Interactive dashboards, charts, maps, filters, navigation
Data	CSV	Global air quality and weather data (city, country, date, AQI, pollutants, weather)

Conclusion

Climate Visualizer is a data-driven web app for exploring and analyzing global air quality.

It provides interactive visualizations and statistical insights, making it useful for researchers, policymakers, and the public interested in air pollution trends and their relationships with weather.