Project Documentation: Air Quality Insights Dashboard

• Introduction

The Air Quality Insights Dashboard is an interactive web application designed to analyze air pollution trends across multiple cities. The project aims to provide actionable insights into air quality using real-time and historical data visualization. The dashboard is built using **Streamlit**, **Power BI**, and a **chatbot integration** to assist users in understanding air quality metrics.

• Project Scope

Inclusions:

- Development of an interactive **Streamlit-based dashboard**.
- Integration with **Power BI** for detailed visual analysis.
- Implementation of a **chatbot** to answer queries about air quality.
- **Data preprocessing and cleaning** to ensure accuracy.
- **Authentication system** for user access control.

Exclusions:

- Real-time data scraping from external sources.
- Advanced machine learning-based AQI predictions.
- Mobile app development.

Limitations & Constraints:

- Dataset only includes historical air quality data from 2015-2020.
- Chatbot responses are limited to predefined queries or dataset insights.

• Requirements

Functional Requirements:

- Secure **login/logout functionality**.
- **Dashboard visualization** for AQI and pollutant trends.
- **City-wise filtering and comparisons**.
- **Chatbot integration** for dataset-related queries.
- **User feedback submission** via email.

Non-Functional Requirements:

- **Scalability** for larger datasets.
- **User-friendly UI** with minimal response time.

- **Secure authentication** to prevent unauthorized access.

User Stories:

- *As a user, I want to compare AQI trends across cities so that I can understand pollution levels.*
- *As a researcher, I want detailed pollutant statistics so that I can analyze pollution trends.*
- *As an administrator, I want authentication control so that I can manage user access.*

Technical Stack

- **Programming Languages:** Python
- **Frameworks/Libraries:** Streamlit, Pandas, Matplotlib, Plotly, Seaborn, Power BI
- **Databases:** CSV-based data processing
- **Tools/Platforms:** Streamlit, Power BI, Google Generative AI (for chatbot)

Architecture/Design

System Architecture:

- The dashboard follows a **three-layer architecture**:
  - **Data Layer**: Handles data loading, preprocessing, and storage.
  - **Application Layer**: Implements business logic, user authentication, and chatbot responses.
  - **Presentation Layer**: Displays the Streamlit dashboard and Power BI reports.

Design Decisions:

- **Streamlit for dashboard** due to its interactive UI and Python compatibility.
- **Power BI for analytics** because of its advanced visualization capabilities.
- **Google Generative AI** for chatbot responses to enhance user engagement.

Development

- **Coding Standards:**
  - Followed **PEP 8** for Python coding.
  - Modularized code for reusability.
- **Challenges & Solutions:**
  - *Issue:* Large dataset loading time.
  - *Solution:* Used **@st.cache_data** for efficient caching.
  - *Issue:* Alignment issues in the UI.
  - *Solution:* Implemented consistent **CSS styling** and button sizing.

- ## Testing
  - **Unit Tests:** Checked individual functions (e.g., data loading, chatbot responses).
  - **Integration Tests:** Verified end-to-end flow between chatbot, dashboard, and Power BI.
  - **Results:**
    - Found & fixed minor **UI bugs**.
    - Improved chatbot response accuracy.
- ## Deployment
  - **Deployed via Streamlit Cloud**.
  - **Power BI dashboard** is embedded using Power BI service.
  - **Steps to Deploy:**
    - Push code to GitHub.
    - Host on Streamlit Cloud.
    - Embed Power BI report in the application.
- ## User Guide
  - **Login with credentials** to access the dashboard.
  - **Navigate using the sidebar** to explore AQI trends, datasets, and chatbot.
  - **Use filters** to analyze specific cities and pollutants.
  - **Interact with the chatbot** for AQI-related queries.
  - **Submit feedback** via the built-in form.
- ## Conclusion
  - Successfully built an **interactive air quality dashboard**.
  - Integrated **Power BI and chatbot** for enhanced analysis.
  - Key lessons:
    - Efficient **data caching** is crucial for large datasets.
    - **Consistent UI design** improves user experience.
    - **Chatbots add value** in guiding users through complex datasets.
- ## Appendices
  - **Research References:** AQI calculation standards, pollution datasets.

AQI Calculation Standards:

1. **U.S. Environmental Protection Agency (EPA) AQI Guide –** Defines AQI categories and pollutant concentration breakpoints.
   - Source: https://www.airnow.gov

2. **World Health Organization (WHO) Air Quality Guidelines –** Provides global recommendations for safe pollutant levels.
    - Source: https://www.who.int/health-topics/air-pollution
3. **India's National Air Quality Index (NAQI) Standards –** Specifies AQI calculation methodology used in India.
    - Source: https://cpcb.nic.in/

Pollution Datasets:

1. **Central Pollution Control Board (CPCB), India –** Provides real-time and historical air quality data.
    - Dataset: https://app.cpcbccr.com/ccr/#/caaqm-dashboard
2. **OpenAQ:** Open Air Quality Data – Global air quality data from multiple sources.
    - Dataset: https://openaq.org
3. **World Bank Air Pollution Data –** Comprehensive datasets on air pollution levels worldwide.
    - Dataset: https://data.worldbank.org/indicator/EN.ATM.PM25.MC.ZS
4. **NASA Earth Observatory –** Satellite-based air pollution measurements and analysis.
    - Dataset: https://earthdata.nasa.gov