**Infosys Internship 5.0**

**Project Documentation:**

**ClearView: Interactive Air Quality Insights**

**By –** Nagilla Shyam Kumar

**Introduction:**

**Project Overview**: The "Interactive Air Quality Monitoring" project aims to develop an accessible Power BI dashboard for visualizing air quality metrics. Utilizing historical air quality data and meteorological variables, the dashboard empowers users to analyze trends and assess air quality levels regionally and temporally. This is integrated into a user-friendly Streamlit web application, catering to environmentalists, policymakers, and the general public.

**Objectives**:

* **Data Collection and Preprocessing:** Collect and preprocess historical AQI data (PM2.5, PM10, NO2, SO2, O3) to ensure accuracy and consistency.
* **Interactive Dashboard:** Create an interactive Power BI dashboard with actionable insights.
* **Trend Analysis:** Facilitate AQI trend exploration and regional comparisons.
* **Accessibility:** Integrate the dashboard into a Streamlit application.
* **Robustness:** Ensure reliability through extensive testing.

**Significance:**

* Promotes awareness of air quality and its health implications.
* Aids in identifying high-pollution zones for targeted interventions.
* Provides tools for analyzing trends and disparities.
* Engages the public with an intuitive interface.

**Team Members:**

Mentor name: Bhargava

Coordinator Name: Jothiraghavan

Team -A: Nallana Sai Ganesh, Aryan Sharma, Keerthan K M, Nagilla Shyam Kumar, Tejas Abhay Kulkarni, Shashi Ranjan, JeyaSuba M, Keerthana Porika, Diwakar N, Jose Riyan A, Janani B S, Allu Sai Shri Govardhinee.

**Project Scope:**

**Inclusions:**

1. **Data Collection and Preprocessing:**

* + Integrating historical AQI data and pollutant metrics.
  + Cleaning inconsistencies and ensuring data reliability.

2. **Dashboard Creation:**

* + Designing an interactive Power BI dashboard.
  + Features include time-series analysis, regional filters, and pollutant-specific insights.

3. **Web Application:**

* + Embedding the dashboard into a responsive Streamlit application.
  + Ensuring accessibility across devices.

4. **Visualization Tools:**

* + Incorporating graphs and comparative analysis tools.

5. **Testing & Documentation:**

* + Comprehensive testing for functionality.
  + User documentation for smooth onboarding.

**Exclusions:**

* Real-time AQI data integration.
* Predictive analytics and machine learning.
* Mobile app development.

**Constraints:**

* Dependence on historical data accuracy.
* Limited geographical scope.
* Scalability adjustments for additional data sources.
* Assumes basic user familiarity with dashboards.

**Requirements:**

**Functional Requirements:**

1. Import and preprocess historical AQI data.
2. Implement interactive visualizations in Power BI.
3. Embed the dashboard in a Streamlit application.
4. Provide role-based data access.
5. Conduct thorough testing for usability.

**Non-Functional Requirements:**

1. Ensure responsive design for multiple devices.

2. Maintain quick load times for large datasets.

3. Prioritize data security and privacy.

4. Compatibility with modern web browsers.

**Technical Stack:**

* **Programming Languages:** Python, DAX (Power BI)
* **Frameworks:** Streamlit
* **Libraries:** Pandas, matplotlib, smtplib
* **Tools:** Power BI, Visual Studio Code

**Architecture/Design:**

The architecture of the Clearview project follows a modular approach to ensure scalability and efficiency. Each stage is carefully designed to handle specific tasks while ensuring seamless integration with other components.

1. **Data Collection:**

* Historical air quality and meteorological data are sourced from reliable repositories and APIs.
* The data includes key pollutants (PM2.5, PM10, NO2, SO2, O3) and auxiliary parameters like temperature and humidity.

1. **Data Processing:**

* Preprocessing scripts standardize datasets by handling missing values, normalizing units, and resolving inconsistencies.
* A structured format is established to facilitate smooth downstream integration.

1. **Data Analysis:**

* Statistical analysis identifies key patterns and anomalies within the dataset.
* Optional exploratory methods provide a foundation for future predictive modeling.

1. **Visualization:**

* Power BI dashboards are designed to visualize trends, regional comparisons, and pollutant-specific insights.
* Interactive charts, graphs, and heatmaps enhance the user experience by offering drill-down and filter options.

1. **Web Application Integration:**

* The dashboard is embedded into a Streamlit application using iframe integration.
* Responsive design principles ensure usability across devices such as desktops and tablets.

1. **User Interaction Layer:**

* Filters and selectors in the Streamlit app allow users to customize views, such as selecting regions, time periods, and specific pollutants.
* Features like tooltips and legends provide additional context for data insights.

1. **Feedback Mechanism:**

* A feedback loop enables users to report issues or suggest enhancements.
* This iterative process helps maintain data accuracy and improves user satisfaction.

1. **Data Flow:**

* Data moves seamlessly from the collection layer through preprocessing and analysis stages to the visualization module.
* Dynamic updates are facilitated through efficient database queries and API calls.

1. **Security Considerations:**

* All user interactions and data exchanges are encrypted to maintain privacy and integrity.
* Role-based access controls ensure that sensitive data is available only to authorized users.

**Workflow in Summary:**

1. Start with raw air quality data.
2. Preprocess the data (clean, normalize, encode).
3. Visualize it using Power BI.
4. Integrate the visualization into a Streamlit web app.
5. Deploy the app for users to interact and analyze air quality insights.

**Development:**

**Technologies Used:**

* Streamlit for creating the website.
* Power BI for building interactive dashboards.

**Coding Standards:**

* Used consistent indentation and meaningful variable names.
* Followed modular design for reusability and maintainability.

**Challenges and Solutions:**

1. **Data Cleaning:** Resolved inconsistencies with preprocessing scripts.
2. **Integration:** Ensured seamless embedding of Power BI in Streamlit.
3. **Interactivity:** Enhanced user experience with clear filters and intuitive navigation.

**Testing:**

**Approach:**

1. Unit testing for core modules.
2. Integration testing for Streamlit and Power BI components.
3. System testing for overall application reliability.

**Results:**

* Resolved compatibility and responsiveness issues.
* Achieved accurate data visualization.
* Verified system stability across devices.

**Deployment:**

**Process:**

1. Hosted the Streamlit app locally for demonstration purposes.
2. Embedded the Power BI dashboard URL directly into the app.

**Instructions:**

* Install required Python libraries (e.g., streamlit, Firebase).
* Run the Streamlit app using streamlit run app.py.
* Ensure Power BI dashboard URL is accessible for embedding.

**User Guide:**

1. **Setup to Use:**
   1. Run the application using:

streamlit run app.py

* 1. Navigate through login/signup and explore the dashboard.
  2. Use filters to customize visualizations.

1. **Troubleshooting:**
   * Ensure database availability.
   * Verify internet connectivity for Power BI dashboards.

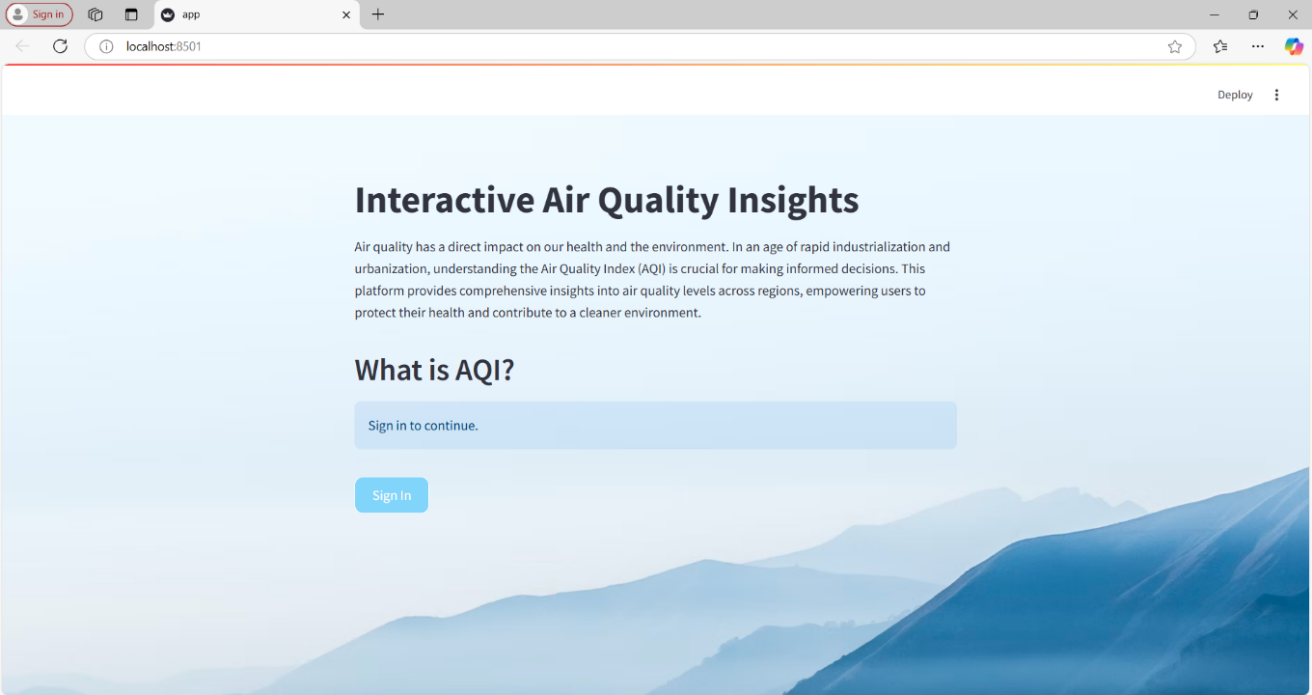
**Conclusion:**

The ClearView project successfully developed an interactive platform for visualizing AQI data, enabling insightful analysis and fostering awareness. Key areas for improvement include expanding database coverage and optimizing performance for broader accessibility.

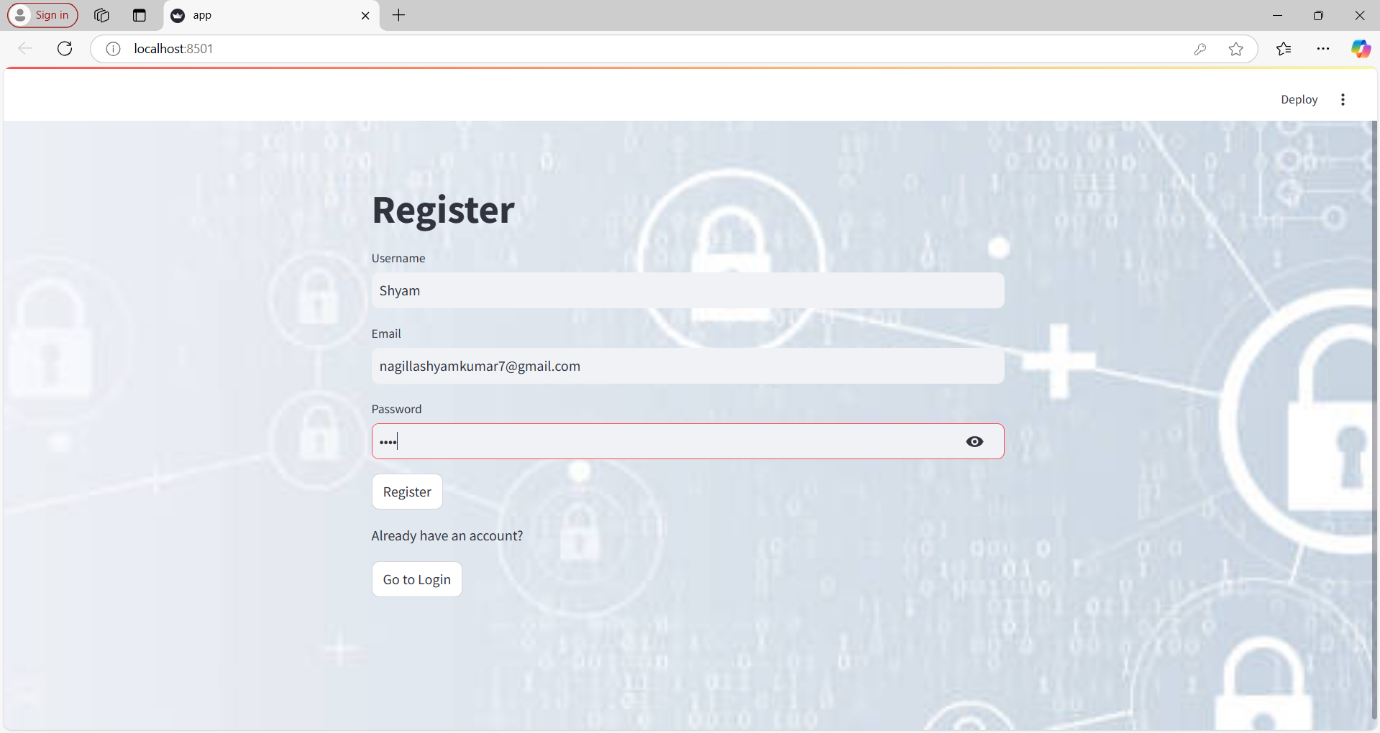
**Appendices:**

The below are the screenshots of the UI.

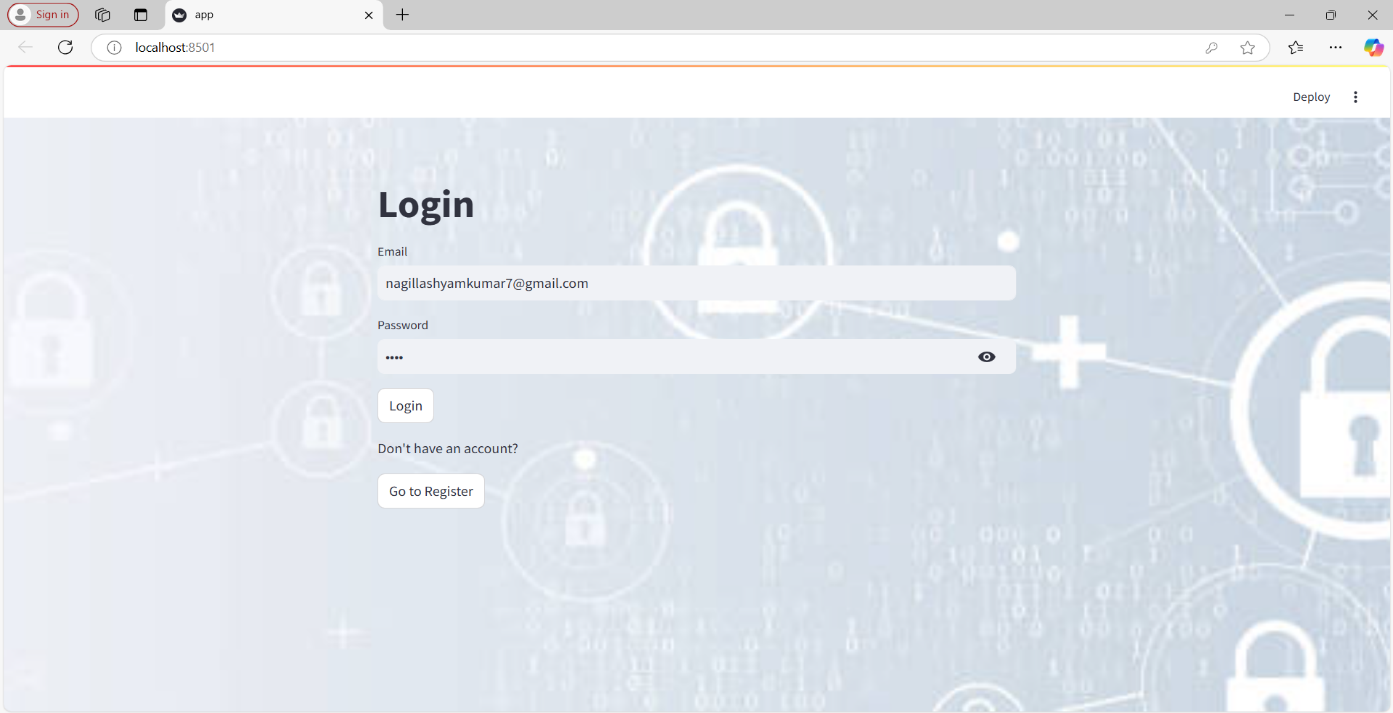
* 1. Landing page:



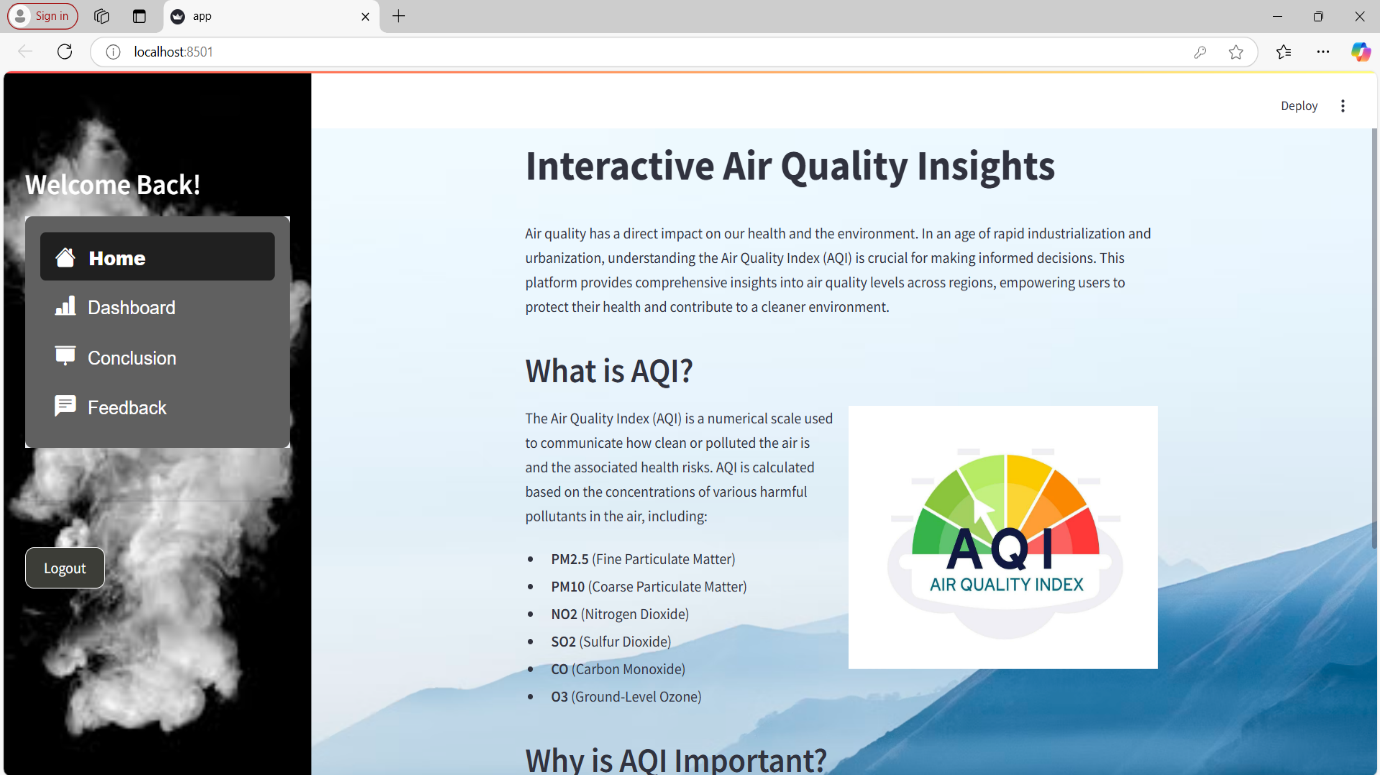
* 1. Register page:



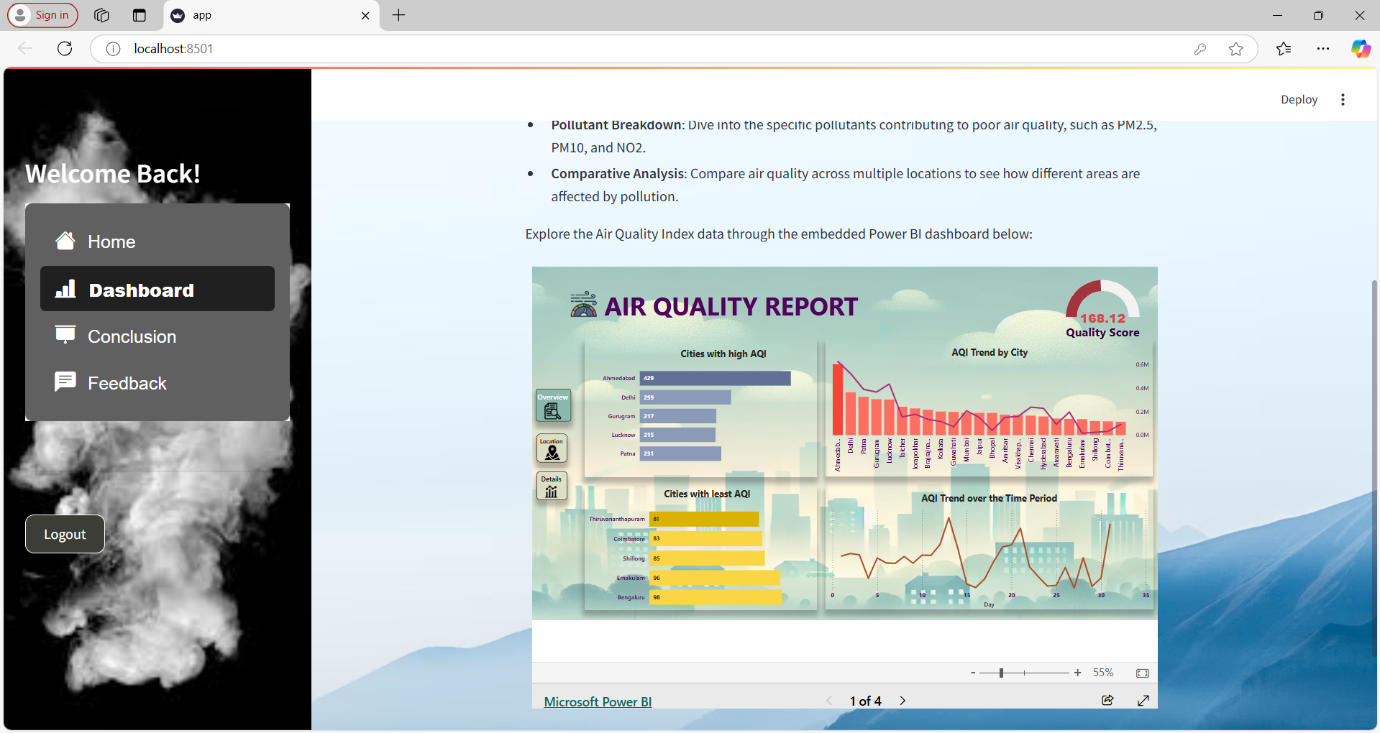
* 1. Login page:



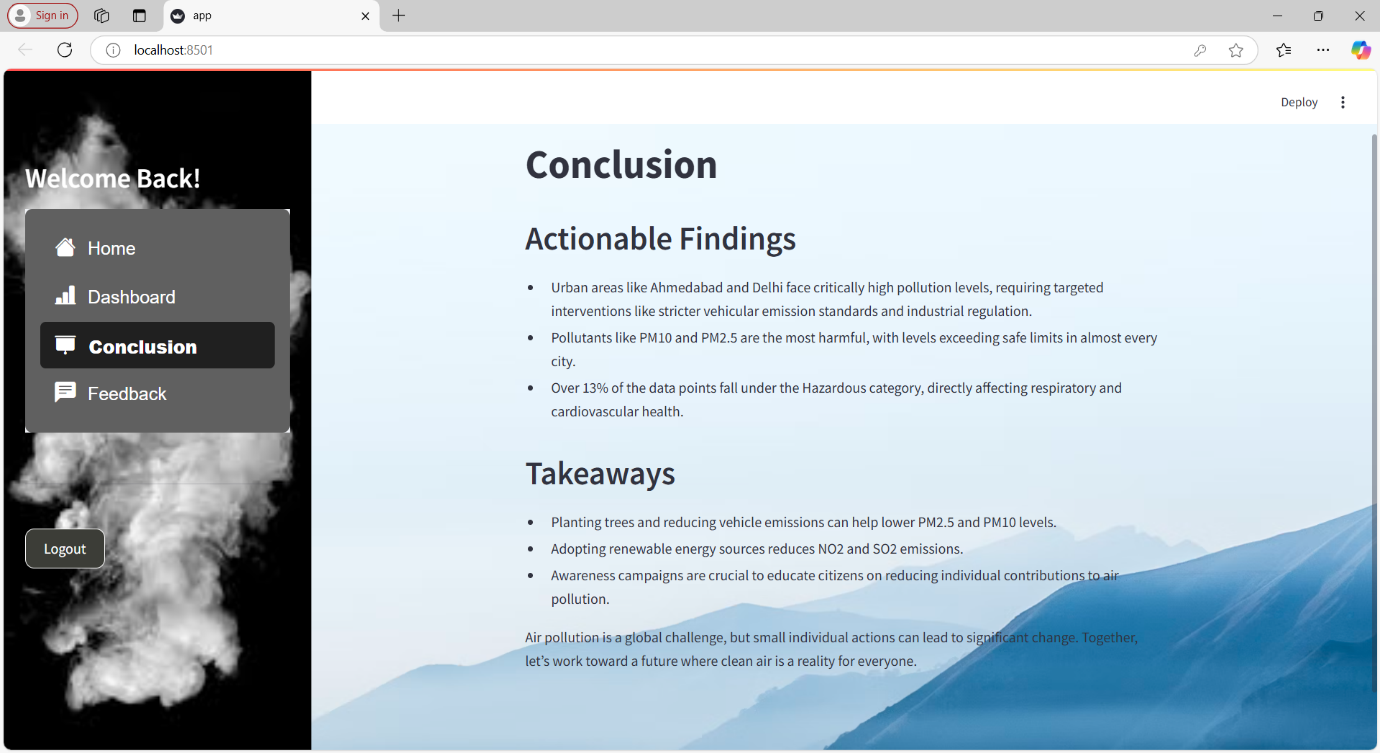
* 1. Home page:



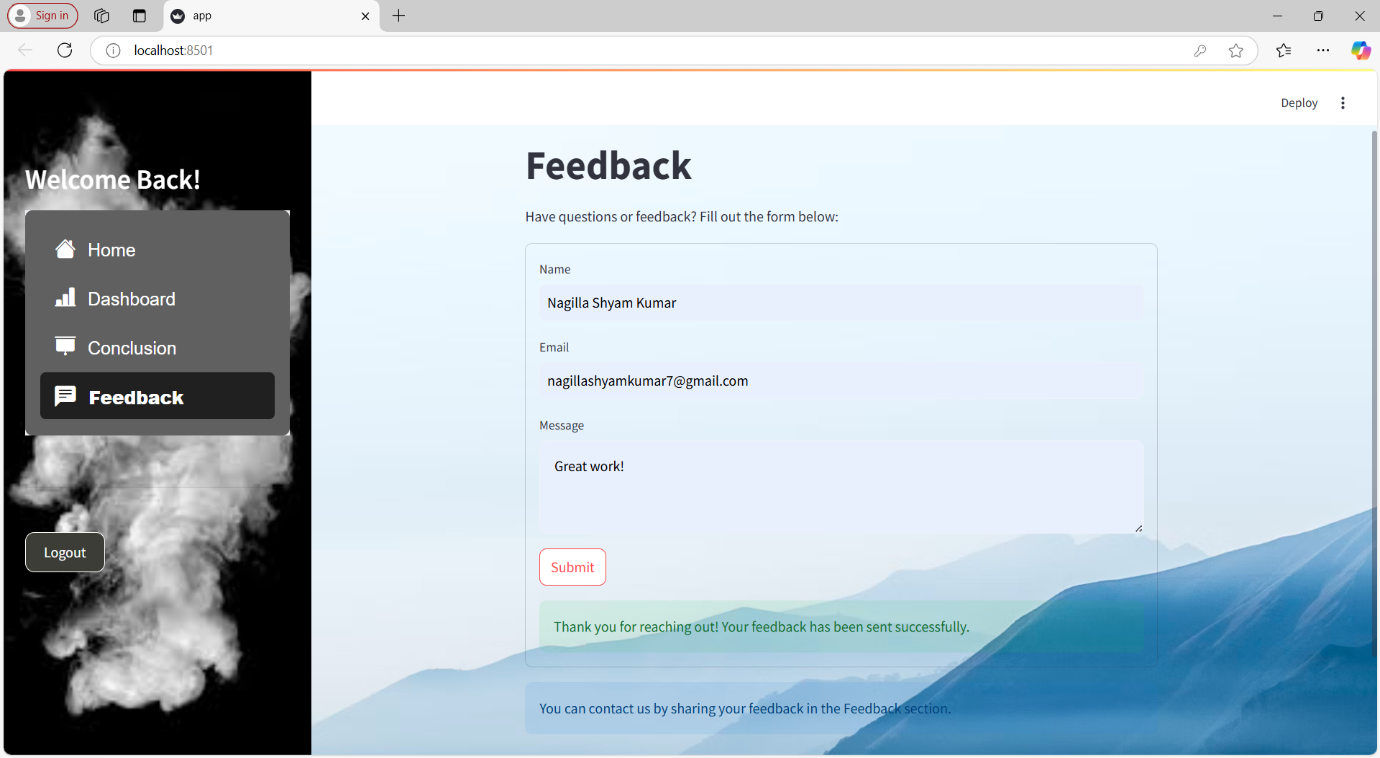
* 1. Dashboard page:



* 1. Conclusion page:



* 1. Feedback page:



* 1. Feedback Mail:

