

# Title: Air Quality Dashboard Application Documentation

## Introduction:

This documentation provides an overview of the code structure and functionalities of the Air Quality Dashboard application. The application aims to visualize and analyze air quality data using Streamlit and various data visualization libraries such as Seaborn and Matplotlib.

## Code Overview:

The code consists of a Python script containing the main functionality of the Air Quality Dashboard application. It imports necessary libraries such as Seaborn, Streamlit, Pandas, and Matplotlib, along with custom modules for generating different types of graphs and analysis.

## Functionality:

1. Data Loading:
  - Users can upload a CSV, TXT, XLSX, or XLS file containing air quality data using the file uploader widget.
  - If no file is uploaded, a sample CSV file is loaded as the default dataset.
2. Menu Navigation:
  - The application provides a sidebar menu with various options for data analysis and visualization.
  - Users can navigate between different sections of the dashboard using the menu options.
3. Home Page:
  - The home page displays initial visualizations and summary statistics of the air quality data.
  - It includes line plots, heatmaps, and violin plots to represent different aspects of the data.

4. Time Graphs:
  - This section displays time series graphs of air quality measurements over time.
5. Measurements by Pollutants:
  - Users can explore measurements grouped by pollutants using visualizations tailored for this analysis.
6. Alerts and Regulatory Thresholds:
  - This section provides insights into alerts and regulatory thresholds related to air quality.
7. Pollution Sources Analysis:
  - Users can analyze the sources of pollution and their impact on air quality using specialized visualizations.
8. Regional Comparisons:
  - Comparative analysis of air quality across different regions is presented in this section.
9. History and Trends:
  - Users can explore historical trends and patterns in air quality data through visualizations and analysis.

## Dependencies:

- Seaborn
- Streamlit
- Pandas
- Matplotlib

## Explanation of Graph Functions:

### 1. Time Graphs:

- **Function Name:** `time_graphs(df)`
- **Description:** This function generates a bar chart representing the air quality measurements over time for a selected pollutant.
- **Parameters:**

- `df`: DataFrame containing air quality data with columns 'end\_date', 'Polluant', and 'valeur'.
- Interaction:
  - Users can select a pollutant from a dropdown menu to view its measurements over time.

## 2. Measurements by Pollutants:

- **Function Name:** `measurements_by_pollutants(df)`
- **Description:** This function creates a line chart displaying the measurements of a selected pollutant over time.
- **Parameters:**
  - `df`: DataFrame containing air quality data with columns 'end\_date', 'Polluant', and 'valeur'.
- **Interaction:**
  - Users can select a pollutant from a dropdown menu to visualize its measurements over time.
  - Data for the selected pollutant can be downloaded as a CSV file using the provided download button.

## 3. Alerts and Regulatory Thresholds:

- **Function Name:** `alerts_and_regulatory_thresholds()`
- **Description:** This function generates a bar chart showing various parameters with indicators for exceeding regulatory thresholds.
- **Interaction:**
  - The function highlights bars exceeding a predefined regulatory threshold with text annotations.
  - A horizontal line represents the regulatory threshold, aiding in visual comparison.

## 4. Pollution Sources Analysis:

- **Function Name:** `pollution_sources_analysis(df)`
- **Description:** This function produces line plots to analyze the evolution of air quality over time for different types of pollution sources.
- **Parameters:**

- `df`: DataFrame containing air quality data with columns 'type d'influence', 'Date de début', and 'valeur'.
- Interaction:
  - Users can observe trends in air quality indices associated with various pollution source types.

## 5. Regional Comparisons:

- **Function Name:** `regional_comparisons(df)`
- **Description:** This function generates line plots comparing air quality across different regions.
- **Parameters:**
  - `df`: DataFrame containing air quality data with columns 'Zas', 'Date de début', and 'valeur'.
- **Interaction:**
  - Users can visualize and compare air quality trends among different regions.

## 6. History and Trends:

- **Function Name:** `history_and_trends(df)`
- **Description:** This function creates line plots illustrating the historical evolution of air quality over time.
- **Parameters:**
  - `df`: DataFrame containing air quality data with columns 'Date de début', 'type d'influence', and 'valeur'.
- **Interaction:**
  - Users can explore historical trends and patterns in air quality indices associated with different influencing factors.

## Conclusion:

**Conclusion:** The Air Quality Dashboard application provides a user-friendly interface for visualizing and analyzing air quality data. By leveraging Streamlit and various data visualization libraries, users can gain valuable insights into air quality metrics, trends, and regulatory compliance. Further enhancements and customizations can be made to adapt the application to specific use cases and data sources.