# Case Analysis Dashboard

A comprehensive and interactive web application built with Streamlit for analyzing and visualizing case management data from an Excel file. This dashboard provides key insights into case volumes, resolution times, backlog trends, and performance metrics through a user-friendly interface.

## Features

* **Interactive Dashboard**: A clean, wide-layout interface for easy data exploration.
* **Flexible Time Frame Selection**: Filter data by a custom date range or by a specific week.
* **Dynamic KPI Metrics**: View high-level metrics for cases opened and closed within the selected period.
* **Weekly Summaries**: Get a quick overview of opened vs. closed cases on a weekly basis.
* **Detailed Breakdowns**: Analyze opened and closed cases by Product Line and Case Reason using data tables and interactive pie charts.
* **Drilldown Analysis**: Select a specific Product Line to see a further breakdown by Product Model, Case Reason, and Case Owner.
* **Queue Analysis**: Isolate and analyze cases that are currently sitting in a support queue.
* **Year-to-Date (YTD) Analysis**:
  + Visualize long-term trends for Average Case Age and Average Resolution Time.
  + Analyze the current open case backlog from the beginning of the year.
* **Data and Chart Export**: Every chart and graph has options to download the visualization as a PNG image and the underlying data as an Excel file.

## Setup and Installation

To run this application on your local machine, please follow these steps.

1. **Prerequisites**:
   * Python 3.8 or newer.
2. Clone the Repository:  
   Download the app.py and requirements.txt files and place them in the same project directory.
3. Create a Virtual Environment (Recommended):  
   It's best practice to create a virtual environment to manage project dependencies.  
   # For Windows  
   python -m venv venv  
   venv\Scripts\activate  
     
   # For macOS/Linux  
   python3 -m venv venv  
   source venv/bin/activate
4. Install Dependencies:  
   Install all the required Python packages using the requirements.txt file.  
   pip install -r requirements.txt

## How to Run the Application

1. Navigate to the project directory in your terminal.
2. Run the following command:  
   streamlit run app.py
3. Your web browser will automatically open a new tab with the running application.

## How to Use the Dashboard

1. **Upload Data**: Click the "Browse files" button to upload your case report in .xlsx format.
2. **Select Time Frame**: Use the sidebar on the left to filter the data. You can choose to filter by a specific **Date Range** or by **Week**.
3. **Analyze**: The dashboard will dynamically update to reflect the data within your selected time frame.
4. **Download**: Use the "Download Chart" and "Download Data" buttons beneath any visualization to export it for your reports.

## Input Excel File Format

For the dashboard to function correctly, your uploaded Excel file **must** contain the following columns. The column names must be an exact match.

| **Column Name** | **Data Type** | **Description** | **Required?** |
| --- | --- | --- | --- |
| Opened Date | Date/Datetime | The date the case was created. The script assumes dayfirst=True format (e.g., DD/MM/YYYY). | **Yes** |
| Status | Text | The current status of the case. The script specifically looks for statuses defined in the app's code. | **Yes** |
| Product Line | Text | The primary product line associated with the case. | **Yes** |
| Case Last Modified Date | Date/Datetime | The date the case was last updated. Required for resolution time and closed case calculations. | **Yes** |
| Case Reason | Text | The reason the case was opened. | **Yes** |
| Case Owner | Text | The person or queue the case is assigned to. | **Yes** |
| Product Model | Text | The specific model of the product. | **Yes** |

**Note**: The application's logic for "Open" and "Closed" cases is based on the OPEN\_STATUSES and CLOSED\_STATUSES lists defined at the top of app.py. You can edit these lists to match the specific status names used in your reports.