Data Analysis Project using A.I by Gyan Prakash Patel

(AI Prompt Engineering for GUI-Based Data Analysis Software)

◇ **ROLE SETTING (Very Important)**

You are an expert Python software developer and data analyst.

You have strong experience in:

- Pandas for data analysis

- Tkinter for GUI applications

- Matplotlib for charts

- Building corporate-ready desktop automation tools

◇ **PROJECT CONTEXT**

I want to build a professional desktop software (GUI-based) for corporate users. The software should allow non-technical users to analyze Excel or CSV data without writing any Python code.

◇ **FUNCTIONAL REQUIREMENTS**

Write a complete Python program using Tkinter and Pandas that does the following:

1️⃣File Selection & Reading

- Allow the user to browse and select a CSV or Excel file.

- Add a separate "Read" button.

- When the Read button is clicked:

- Read the full dataset - Display:

- Total number of rows

- Total number of columns

- All column headings

2️⃣ Dynamic Column Detection

- Automatically detect:
- Text columns (object/string dtype)
- Numeric columns (int/float or numeric-looking text)

3️⃣ Report Builder (Interactive Analysis) -

Provide dropdowns for:

1. Group By Column (only text columns)
2. Aggregation Method (sum, mean, average, max, min, count, median)
3. Value Column (only numeric columns)

- When user clicks "Preview Report":
- Apply GroupBy + selected aggregation
- Sort results in descending order
- Display the result table inside the GUI (not Excel)

4️⃣ Export Report

- Add an Export button.
- Allow export format selection:
- Excel (.xlsx)
- CSV (.csv)
- Export the report automatically into the same folder where the input file is located.

5️⃣ Chart Builder (GUI-Based Visualization)

- Provide a dropdown to select chart type:
- Bar chart
- Column chart
- Line chart
- Pie chart

- When user clicks "Preview Chart":
- Generate the selected chart using the report data
- Display the chart inside the GUI window

- Add "Export Chart" option:
- Export chart as PNG image
- Save it in the same folder as the input file

6️⃣ Usability & Error Handling

- Show proper error messages if:
- File is not selected
- Read button is not clicked
- Dropdowns are not selected
- Reset previous outputs when a new report is generated
- Keep the GUI clean, professional, and easy to use

◇ **TECHNICAL CONSTRAINTS**

- Use Tkinter for GUI
- Use Pandas for data processing
- Use Matplotlib for charts
- Write clean, readable, well-commented code
- Structure the code so it can be converted into a .EXE using PyInstaller
- Do not require the end user to install Python

◇ **DELIVERABLE FORMAT**

- Provide the full Python code in one single block
- Ensure the program is production-ready
- Do not omit any required functionality