DATA ANALYZER AND VISUALIZER

Name : Rajkumar G. Shahu
Exam No : 30004
Project Guide : Prof. Hirav Joshi

Project Profile :

SARDAR PATEL UNIVERSITY

C. P. Patel F. H. Shah Commerce (Autonomous) Collage, Anand
B.C.A, B.B.A (ITM), PGDCA & M.Sc. (Data Science) Department

M.Sc. (DS) Semester - 3 PG03CMDS05 (2025 - 2026)

Full Name	**RAJKUMAR GAURANGBHAI SHAHU**
Project Title	**DATA ANALYZER AND VISUALIZER**
Roll No.	**04**
University Exam No.	**30004**
Front End Tool	**STREAMLIT**
Back End Tool	**PYTHON**
Project Guide	**Prof. Hirav Joshi**

Index

Sr. No	Description of Topic
1	Acknowledgement
2	Project Overview
3	Features
4	Installation & Setup
5	Quick Start Guide
6	Supported File Formats
7	Visualization Types
8	User Interface
9	Conclusion

Acknoledgement

- incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

- I am grateful to our project guide "Prof. Hirav Joshi" for the guidance, inspiration and constructive suggestion that helpful us in the preparation of this project.

- I am also thankful to our Head of The Department "Dr. Nayan Patel" for his continuous guidance.

- I am also obliged to our Principal "Dr. Tejal Tanna" for creative support.

- I also thank my colleagues who have helped in successful completion of the project.

Project Overview

The **Data Analysis & Visualization Dashboard** is a comprehensive web-based application built with Streamlit that enables users to upload, analyze, and visualize data without writing any code. The dashboard provides an intuitive interface for exploratory data analysis (EDA), interactive visualizations, and data export capabilities.

Target Users

- Data analysts and scientists
- Business analysts
- Students and researchers
- Non-technical stakeholders who need data insights

Features

Core Functionality

- **Multi-format file upload** (CSV, Excel, JSON, Parquet)
- **Interactive data exploration** with filtering and sorting
- **Comprehensive statistical analysis** and summaries
- **12+ visualization types** with customization options
- **Real-time data filtering** and transformation
- **Export capabilities** (CSV, Excel formats)
- **Responsive design** for different screen sizes

Analysis Capabilities

- Descriptive statistics
- Data type detection and conversion
- Missing value analysis
- Correlation analysis
- Distribution analysis
- Categorical data analysis

Visualization Features

- Interactive charts with Plotly
- Customizable color schemes
- Dynamic filtering and grouping
- Responsive chart sizing
- Professional styling

Installation & Setup

- **Prerequisites :**
- Python 3.7 or higher
- pip package manager
- 2GB RAM minimum (8GB recommended)
- Modern web browser

- **Step 1: Environment Setup**
- # Create virtual environment (recommended)
- python -m venv streamlit_dashboard
- source streamlit_dashboard/bin/activate
- # On Windows: streamlit_dashboard\Scripts\activate
-
- # Or using conda
- conda create -n streamlit_dashboard python=3.9
- conda activate streamlit_dashboard

Installation & Setup

- **Step 2: Install Dependencies**
- # Install required packages
- pip install streamlit
- pip install pandas
- pip install plotly
- pip install seaborn
- pip install matplotlib
- pip install scipy
- pip install openpyxl
- pip install pyarrow

- **Step 3: Download and Run**
- # Save the dashboard code as 'r1.py'
- streamlit run r1.py

Quick start Guide

- **1. Launch the Application**
- streamlit run r1.py
- The dashboard will open in your default browser at http://localhost:8501

- **2. Upload Your Data**
- Click **"Choose a file"** in the sidebar
- Select your data file (CSV, Excel, JSON, or Parquet)
- Wait for the upload confirmation

- **3. Explore Your Data**
- Review the **Data Overview** section for basic statistics
- Use the **Data Preview** to examine your dataset
- Check **Data Types** and **Missing Values**

Quick start Guide

- **4. Create Visualizations**
- Select a **Visualization Type** from the dropdown
- Choose appropriate **columns** for X and Y axes
- Customize colors, groupings, and aggregations
- View your interactive chart

- **5. Filter and Export**
- Use the **Data Filtering** section to subset your data
- Export filtered results as **CSV** or **Excel**

Supported File Formats

- **CSV Files (.csv)**
- # Reading parameters
- df = pd.read_csv(file,
- encoding='utf-8',
- parse_dates=True,
- infer_datetime_format=True
-)

- **Excel Files (.xlsx, .xls)**
- # Multi-sheet support
- df = pd.read_excel(file, sheet_name=0) # First sheet

- **JSON Files (.json)**
- # Nested JSON flattening
- df = pd.json_normalize(json_data)

- **Parquet Files (.parquet)**
- # High-performance columnar format
- df = pd.read_parquet(file)

Visualization types

- **1. Scatter Plot**
- **Purpose**: Explore relationships between two numeric variables
- **Parameters:**
- X-axis: Numeric column
- Y-axis: Numeric column
- Color: Categorical column (optional)
- Size: Numeric column (optional)
- **Use Cases:**
- Correlation analysis
- Outlier detection
- Pattern identification

- **2. Line Chart**
- **Purpose**: Show trends over time or ordered categories
- **Parameters:**
- X-axis: DateTime or numeric column
- Y-axis: Numeric column
- Group by: Categorical column (optional)
- **Use Cases:**
- Time series analysis
- Trend visualization
- Performance tracking

Visualization types

- **3. Bar Chart**
- **Purpose**: Compare categories or show distributions
- **Parameters:**
- Category: Categorical column
- Value: Numeric column
- Aggregation: sum, mean, count, median
- **Use Cases:**
- Category comparison
- Distribution analysis
- Ranking visualization

- **4. Histogram**
- **Purpose**: Show distribution of a single numeric variable
- **Parameters:**
- Column: Numeric column
- Bins: Number of bins (10-50)
- Overlay: Normal distribution curve
- **Use Cases:**
- Distribution analysis
- Data quality assessment
- Statistical analysis

Visualization types

- **5. Box Plot**
- **Purpose**: Show distribution quartiles and outliers
- **Parameters:**
- Y-axis: Numeric column
- Group by: Categorical column (optional)
- **Use Cases:**
- Outlier detection
- Distribution comparison
- Statistical summary

- **6. Heatmap**
- **Purpose**: Show correlation between numeric variables
- **Parameters:**
- Correlation method: Pearson, Spearman
- Color scale: Various options
- Annotations: Show correlation values
- **Use Cases:**
- Feature selection
- Multicollinearity detection
- Pattern recognition

Visualization types

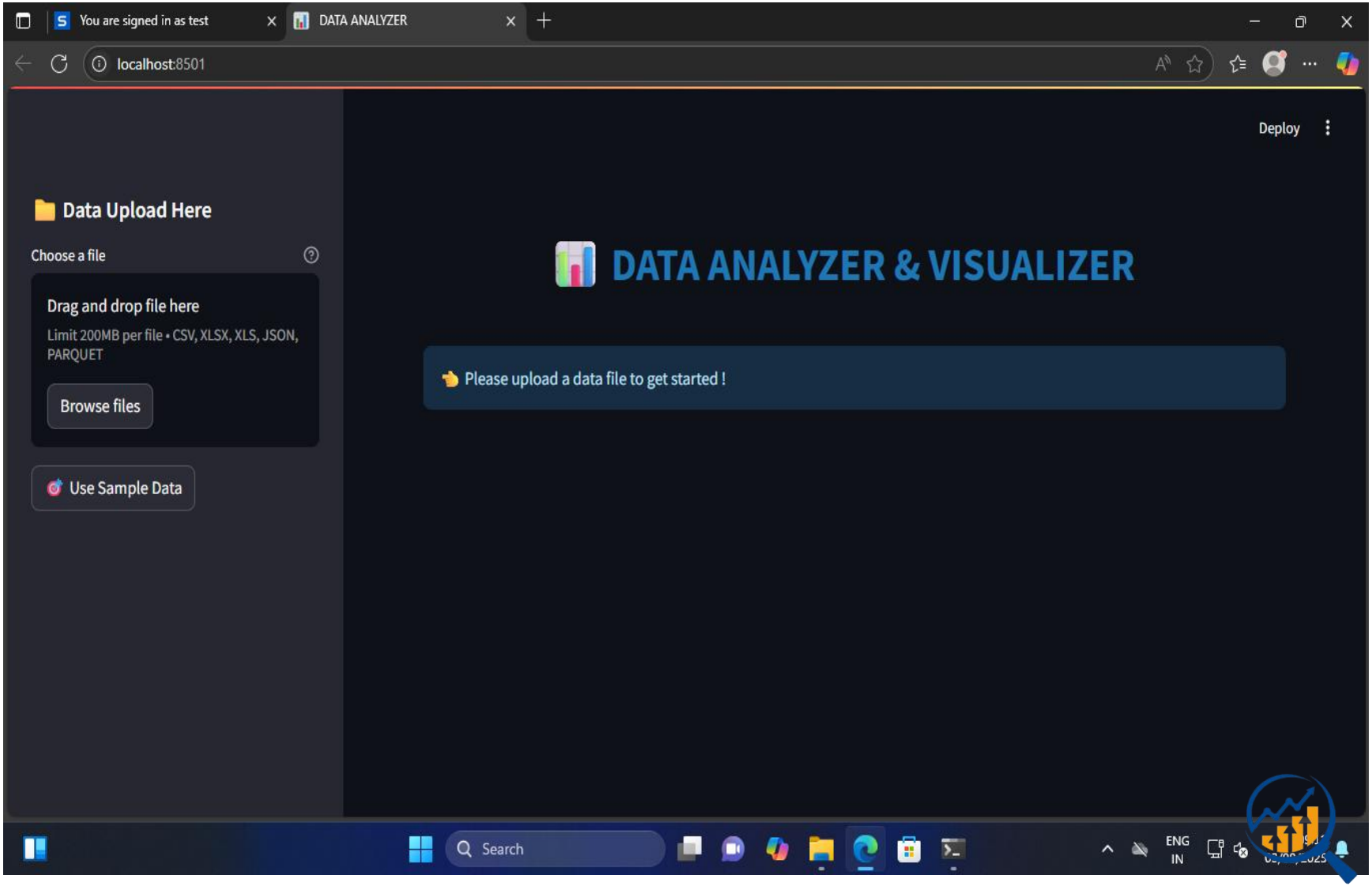
- **7. Pie Chart**
- **Purpose**: Show proportions of categorical data
- **Parameters:**
- Category: Categorical column
- Limit: Top N categories
- **Use Cases:**
- Market share analysis
- Category distribution
- Budget allocation

- **8. Distribution Plot**
- **Purpose**: Combine histogram with density curve
- **Parameters:**
- Column: Numeric column
- KDE: Kernel density estimation overlay
- **Use Cases:**
- Distribution shape analysis
- Statistical modeling
- Data exploration

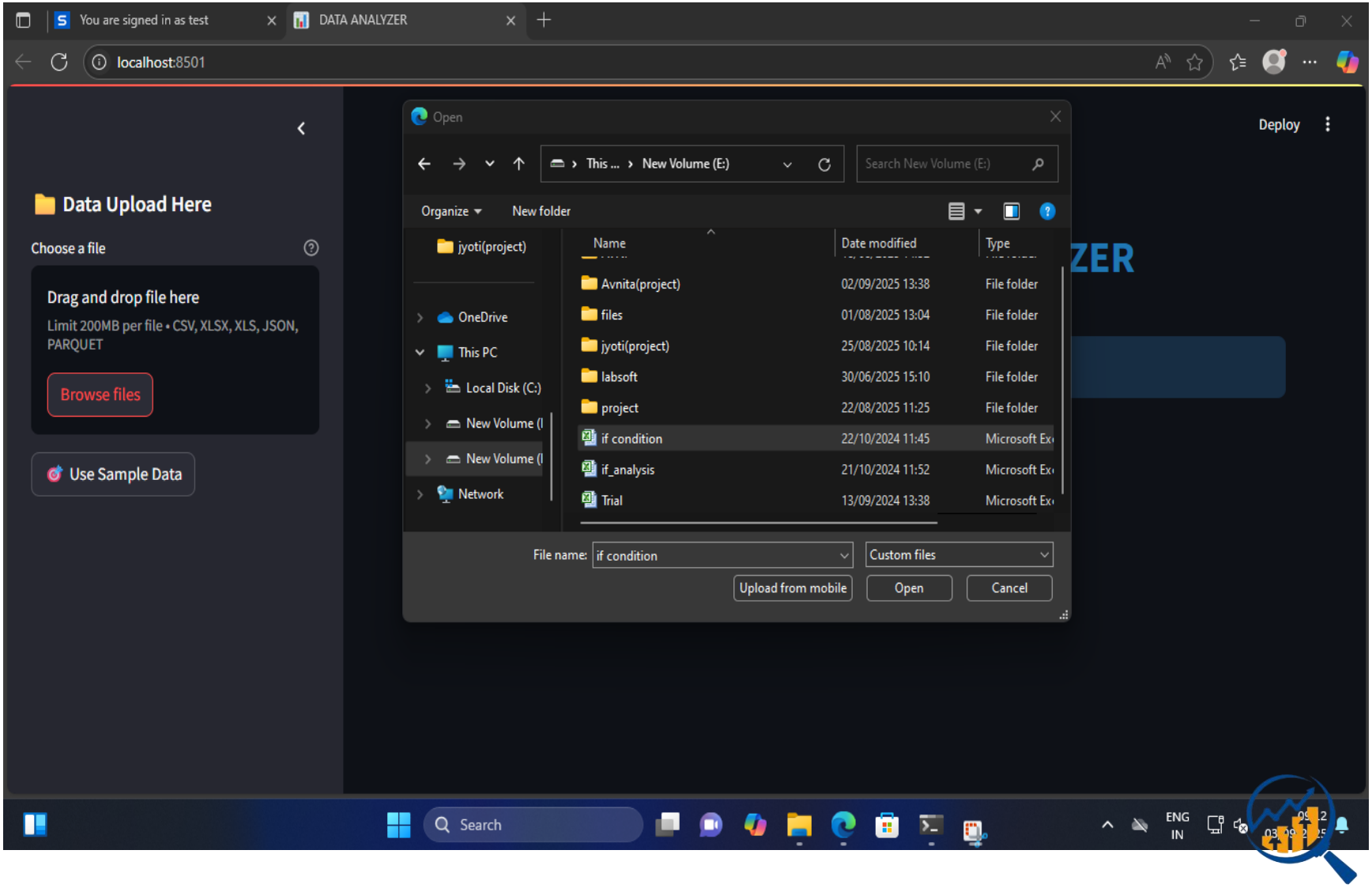
Visualization types

- **9. Time Series**
- **Purpose**: Specialized line chart for temporal data
- **Parameters:**
- Date column: DateTime column
- Value column: Numeric column
- Resampling: Daily, weekly, monthly
- **Use Cases:**
- Temporal trend analysis
- Seasonal pattern detection
- Forecasting preparation

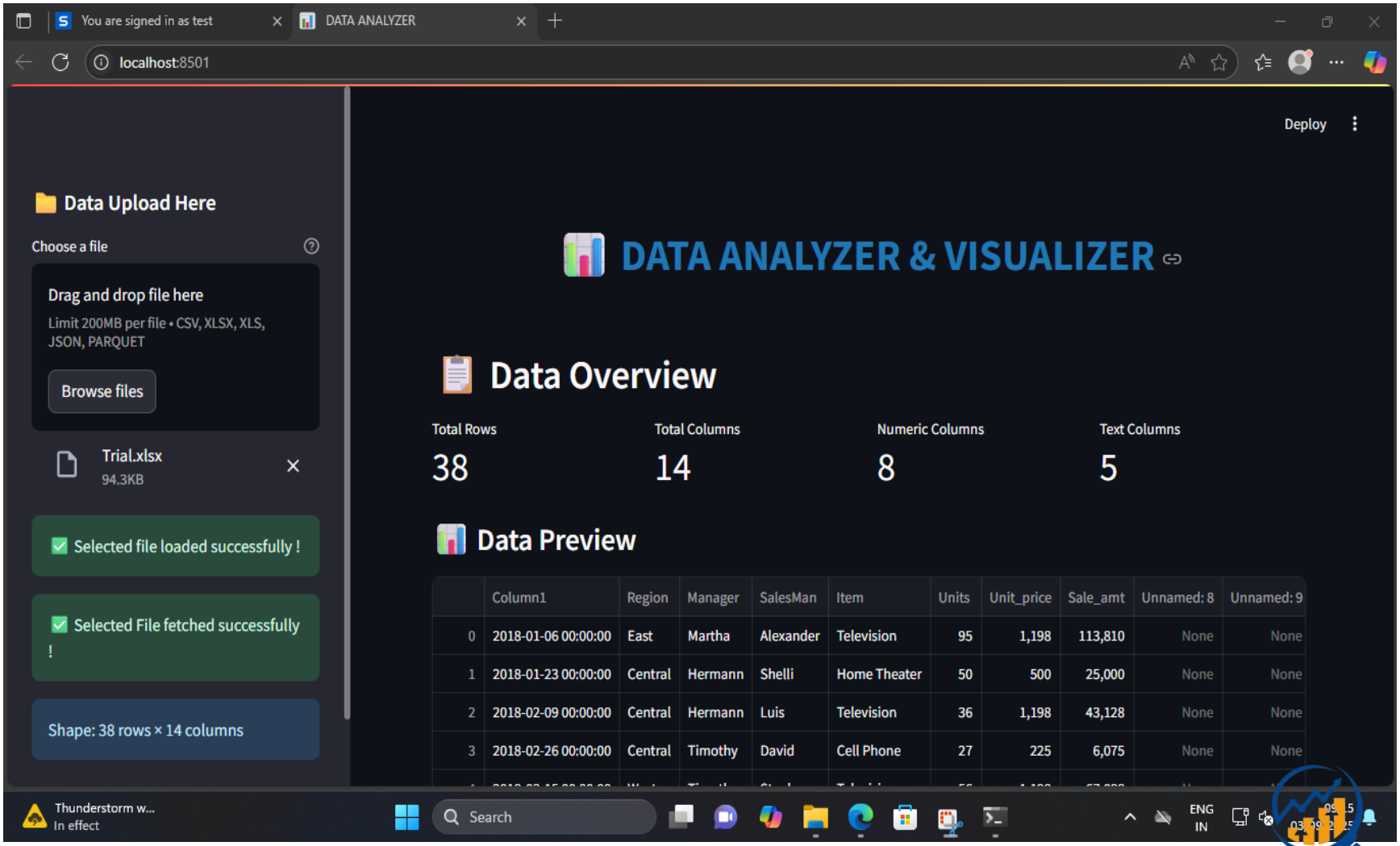
User Interface



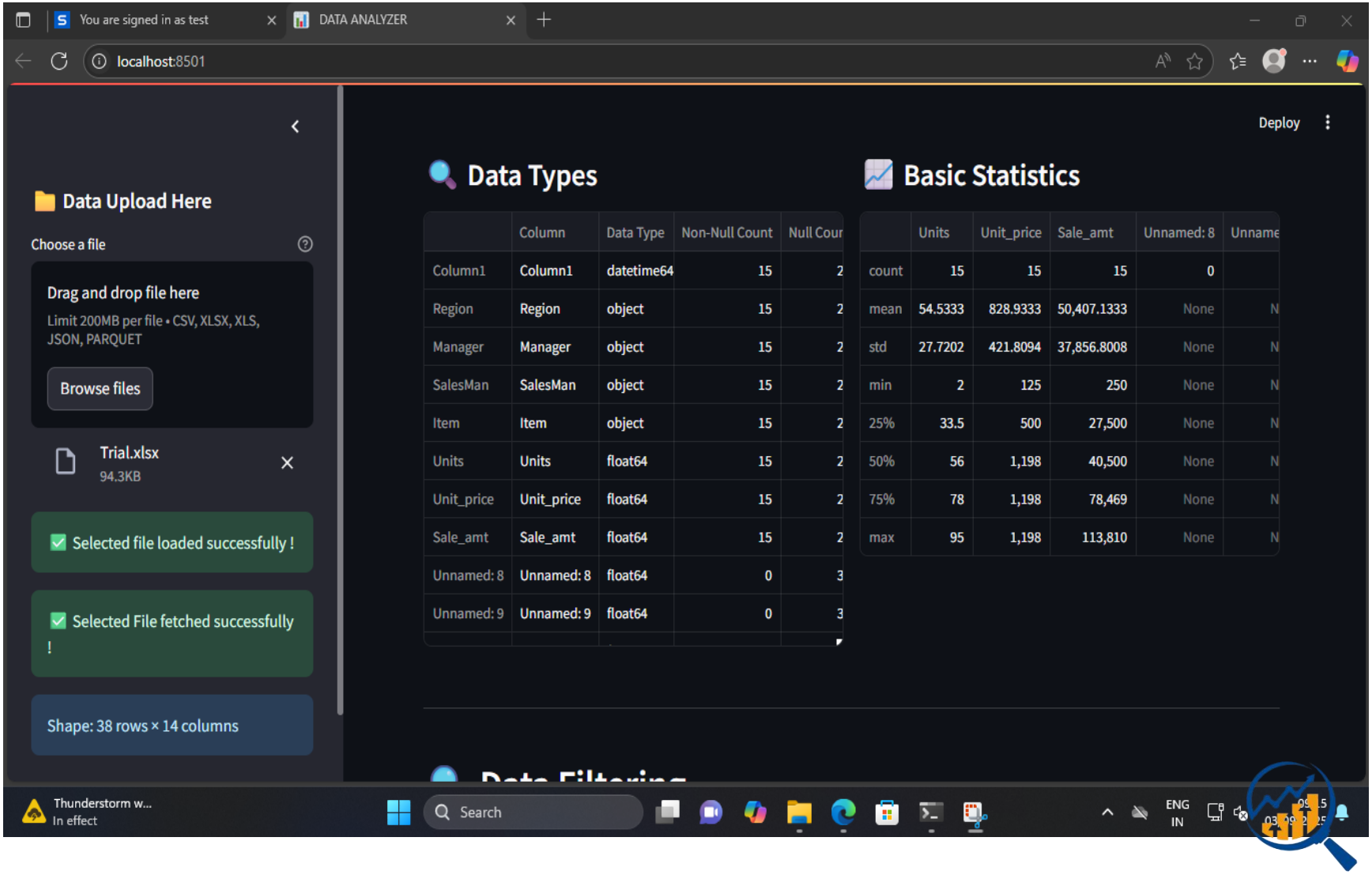
User Interface



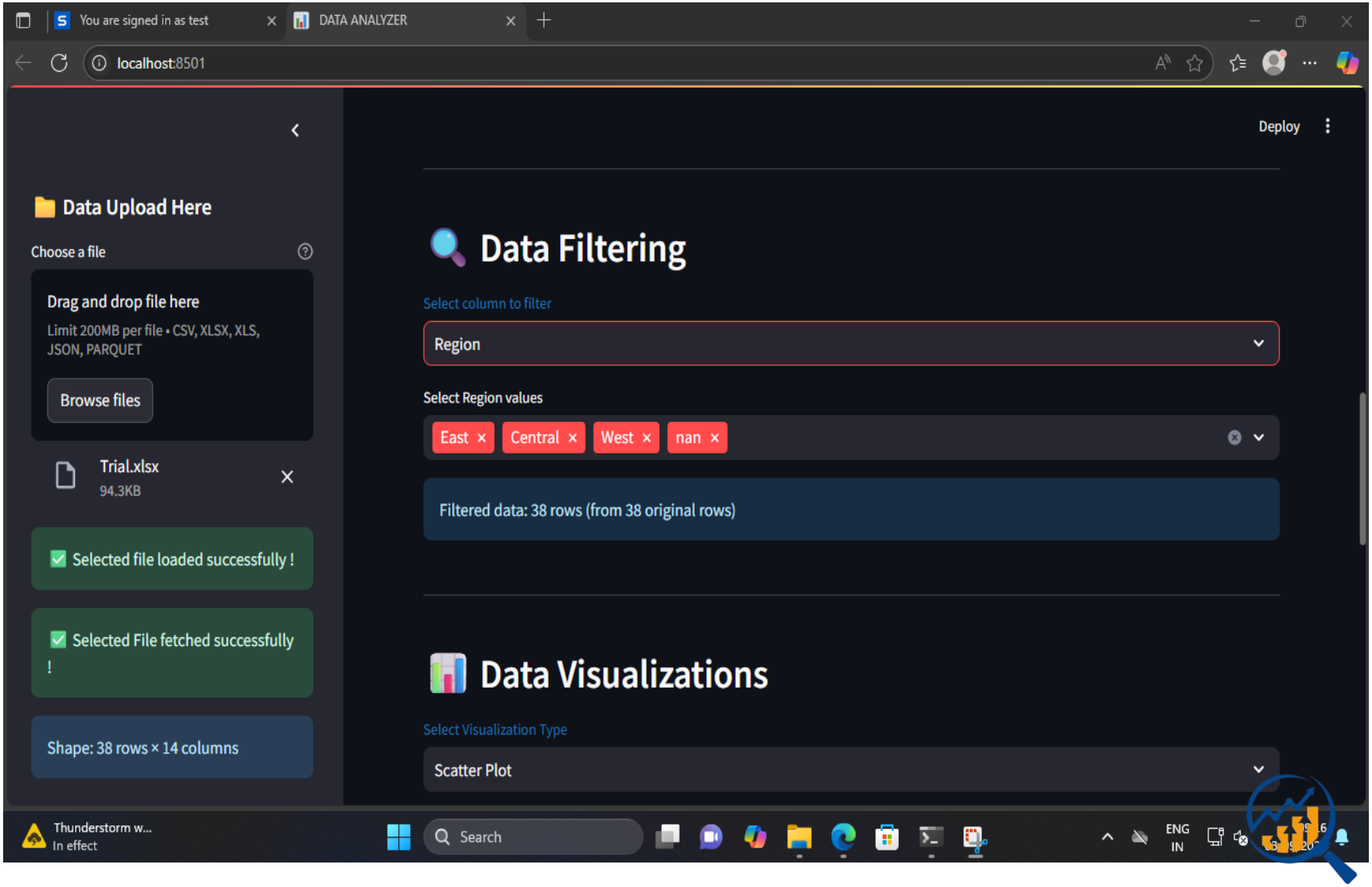
User Interface



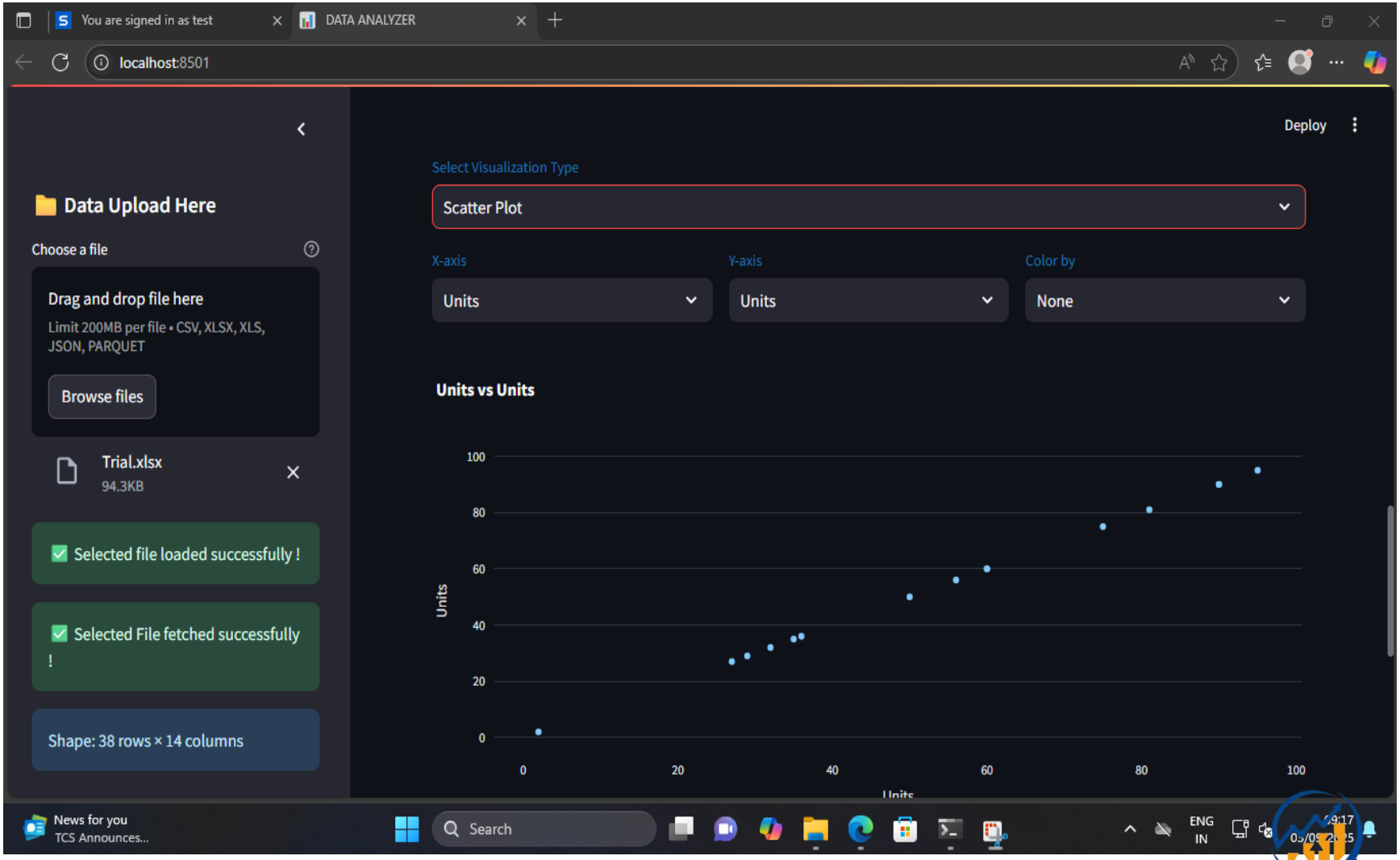
User Interface



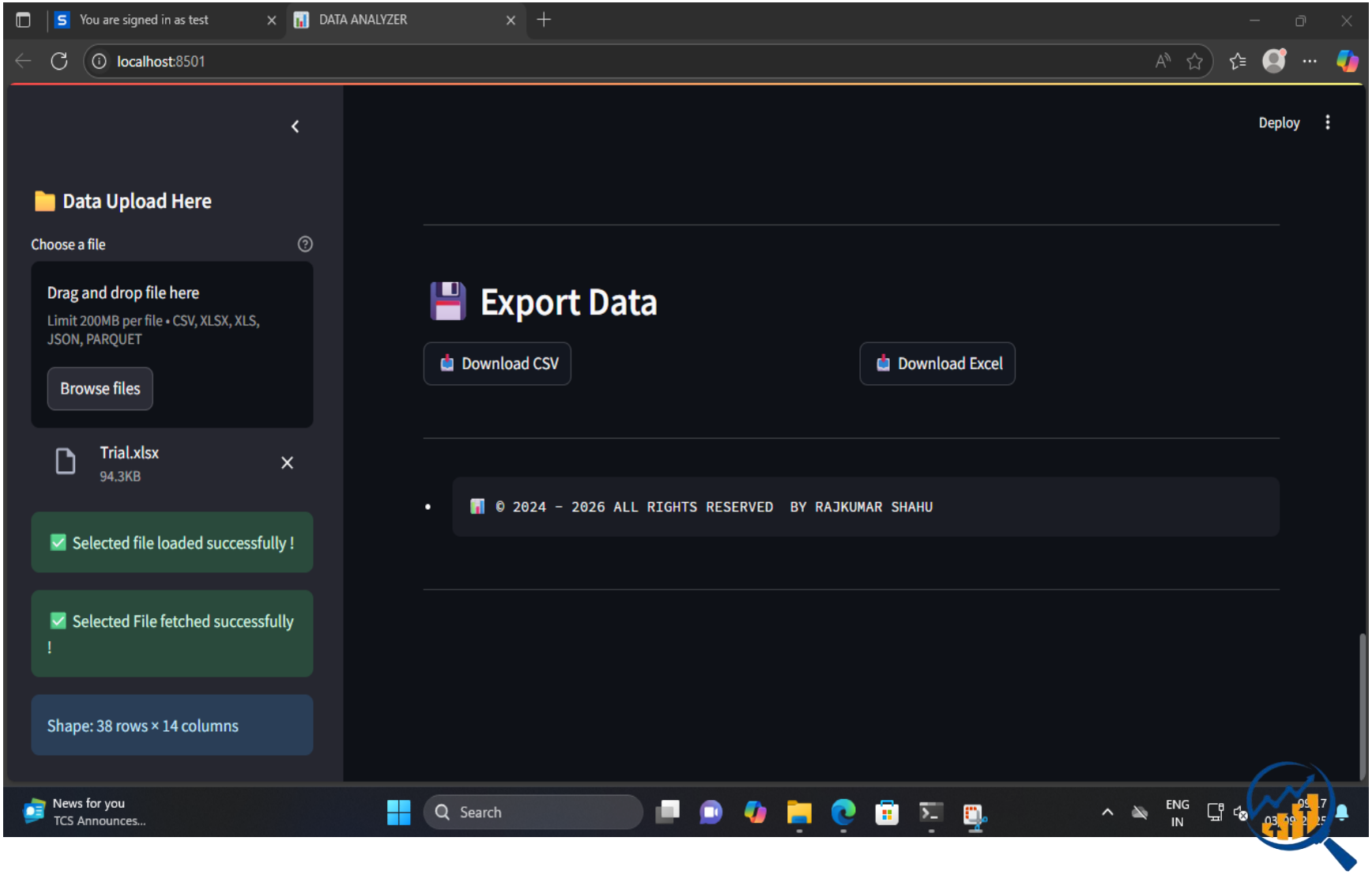
User Interface



User Interface



User Interface



Conclusion :

- In this project, we developed an interactive **Data Analyzer and Visualizer** application using **Python** and **Streamlit**. The tool allows users to upload datasets, explore them through summary statistics, clean data, and generate a variety of insightful visualizations with ease.

- By integrating libraries such as **Pandas**, **Matplotlib**, **Seaborn**, and **Plotly**, we provided dynamic data exploration capabilities, including:

- Viewing dataset structure and summary statistics
- Handling missing values and filtering data
- Creating visualizations like bar charts, histograms, scatter plots, heatmaps, and more
- Exporting cleaned or filtered data for further use

Conclusion :

- This project demonstrates how data science tools can be made accessible through user-friendly web interfaces. It serves as a foundation for more advanced analytics applications, where users can gain insights from their data without writing any code.

- In future iterations, this tool can be enhanced with features like:

- Machine learning model integration
- Real-time data updates
- Dashboard export capabilities
- Collaboration tools or database integration

- Overall, the project showcases the power of combining Python's data stack with Streamlit's simplicity to democratize data analysis for both technical and non-technical users.

Thank You

Created By : Rajkumar G. Shahu