

CHAPTER 1:**INTRODUCTION**

The Sales Dashboard is a dynamic data visualization tool built with Streamlit, offering insights into sales performance based on a dataset from an Excel file. Users can customize their analysis by filtering data according to city, customer type, gender, and payment type. The dashboard presents key metrics like total sales, average rating, and average sales per transaction, providing a comprehensive overview of sales performance. Interactive bar charts visualize sales data by product line and hour, as well as tax information by product line, enhancing the understanding of sales trends and patterns.

CHAPTER 2:**OBJECTIVES**

1. Data Import: Import sales data from an Excel file into a pandas DataFrame for analysis.
2. Data Filtering: Allow users to filter data based on city, customer type, gender, and payment type.
3. Key Metrics: Calculate and display key metrics such as total sales, average rating, and average sales per transaction.
4. Sales by Product Line: Visualize sales data by product line using a horizontal bar chart.
5. Sales by Hour: Display sales data by hour using a bar chart to identify peak sales hours.
6. Tax Information: Present tax information by product line in a bar chart for better understanding of tax contributions.
7. Interactive Visualization: Provide interactive charts that update based on user-selected filters for enhanced user experience.
8. Streamlit Styling: Use Streamlit's styling options to hide default elements like the header and footer for a cleaner dashboard interface.

CHAPTER 3:**PROBLEM STATEMENT**

To design and develop a dynamic Sales Dashboard using Streamlit, enabling users to analyze and visualize sales data effectively, including key metrics, sales by product line and hour, and tax information, with interactive features for a user-friendly experience.

CHAPTER 4:**SCOPE OF PROJECT WORK**

1. Importing sales data from an Excel file into a pandas DataFrame.
2. Creating filters for city, customer type, gender, and payment type.
3. Calculating key metrics such as total sales, average rating, and average sales per transaction.
4. Visualizing sales data by product line and hour using interactive bar charts.
5. Presenting tax information by product line in a bar chart.
6. Implementing Streamlit's styling options to enhance the dashboard's appearance and usability.
7. Testing the dashboard to ensure data accuracy and functionality.
8. Deploying the dashboard for use in a production environment.

CHAPTER 5:**SOFTWARE AND HARDWARE REQUIREMENTS****Software Requirements:**

- Python (3.6+)
- Streamlit library (latest version)
- Pandas library
- Plotly Express library
- Openpyxl library
- MongoDB (for caching data)
- IDE (e.g., Visual Studio Code, PyCharm)
- Web browser

Hardware Requirements:

- Computer with at least 4GB of RAM (8GB recommended)
- Sufficient storage space for data and software
- Internet connection (for data import and Streamlit deployment)
- Operating System: Windows, macOS, or Linux

CHAPTER 6:**ALGORITHM****Step 1: Data Import**

Read sales data from an Excel file into a pandas DataFrame.

Step 2: Data Filtering

Create interactive filters for city, customer type, gender, and payment type.

Step 3: Calculate Key Metrics

Calculate total sales, average rating, and average sales per transaction.

Step 4: Visualize Sales by Product Line

Create a horizontal bar chart to visualize sales data by product line.

Step 5: Visualize Sales by Hour

Create a bar chart to display sales data by hour.

Step 6: Visualize Tax Information

Create a bar chart to show tax information by product line.

Step 7: Streamlit Styling

Use Streamlit's styling options to enhance the dashboard's appearance and layout.

Step 8: Testing and Deployment

Test the dashboard to ensure data accuracy and functionality, then deploy it for use in a production environment.

CHAPTER 7:**MODULE DESCRIPTION**

Data Import Module: Responsible for reading sales data from an Excel file into a pandas DataFrame. It uses the `pd.read_excel()` function from the pandas library.

Filtering Module: Implements interactive filters for city, customer type, gender, and payment type. It uses Streamlit's `st.sidebar.multiselect()` function to create the filters.

Key Metrics Module: Calculates and displays key metrics such as total sales, average rating, and average sales per transaction. It uses pandas' DataFrame methods like `sum()`, `mean()`, and `groupby()`.

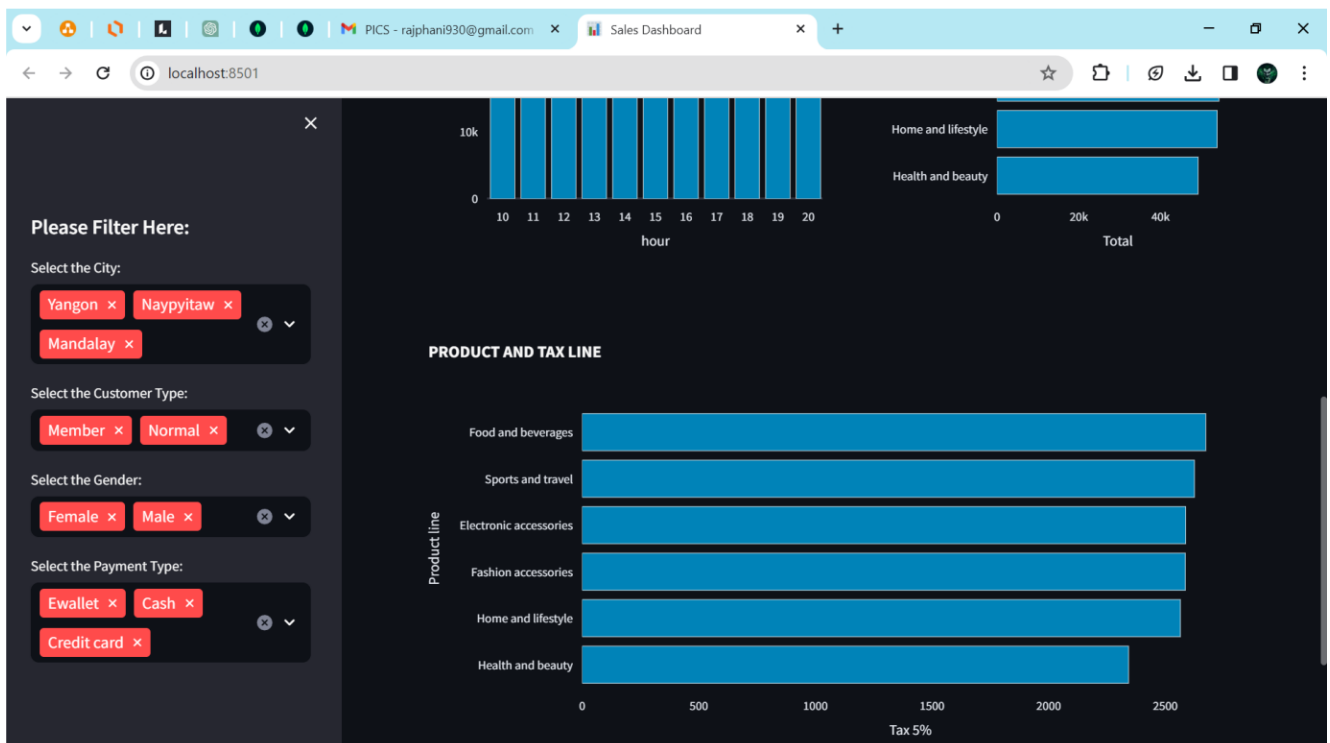
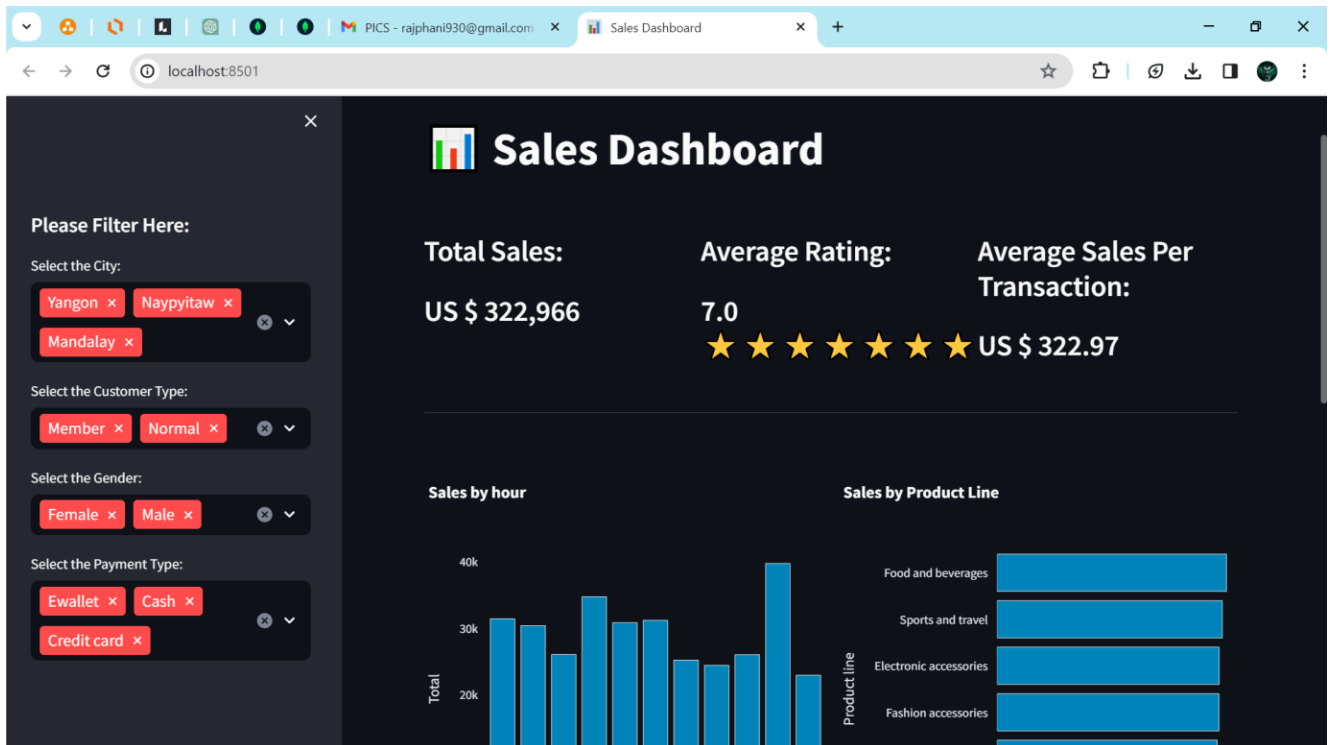
Sales by Product Line Module: Visualizes sales data by product line using a horizontal bar chart. It uses Plotly Express's `px.bar()` function to create the chart.

Sales by Hour Module: Displays sales data by hour using a bar chart to identify peak sales hours. It also uses Plotly Express's `px.bar()` function.

Tax Information Module: Presents tax information by product line in a bar chart. It uses Plotly Express's `px.bar()` function.

Streamlit Styling Module: Implements Streamlit's styling options to hide default elements like the header and footer for a cleaner dashboard interface. It uses HTML/CSS styling within Streamlit's `st.markdown()` function.

Testing and Deployment Module: Tests the dashboard to ensure data accuracy and functionality, then deploys it for use in a production environment. It involves running the Streamlit app locally or deploying it to a web server.

CHAPTER 8:**RESULTS AND DISSICUSSION**

CHAPTER 9:**CONCLUSION**

In conclusion, the Streamlit sales dashboard provides a user-friendly interface for analyzing sales data. It allows users to filter data based on various criteria, view key performance indicators, and visualize sales trends using interactive charts. The dashboard utilizes Python libraries such as Pandas, Plotly Express, and Streamlit to create an interactive and insightful tool for sales analysis. While the dashboard has limitations such as data size restrictions and limited filtering options, it serves as a valuable tool for gaining insights into sales performance and making data-driven decisions. Overall, the dashboard demonstrates the power of Python and its libraries for creating effective data visualization tools for business analytics.

CHAPTER 10:**REFERENCES**

1. Streamlit Documentation: <https://docs.streamlit.io/>
2. Pandas Documentation: <https://pandas.pydata.org/docs/>
3. Plotly Express Documentation: <https://plotly.com/python/plotly-express/>
4. OpenPyXL Documentation: <https://openpyxl.readthedocs.io/en/stable/>
5. Python Programming Language: <https://www.python.org/>