

Project Title:

Retail Analysis Dashboard – Design and Implementation.

DESCRIPTION:

The Retail Analysis Dashboard provides clear and actionable insights into sales, customers, and product performance using interactive charts and KPIs, offering a clean, intuitive, and user-friendly interface.

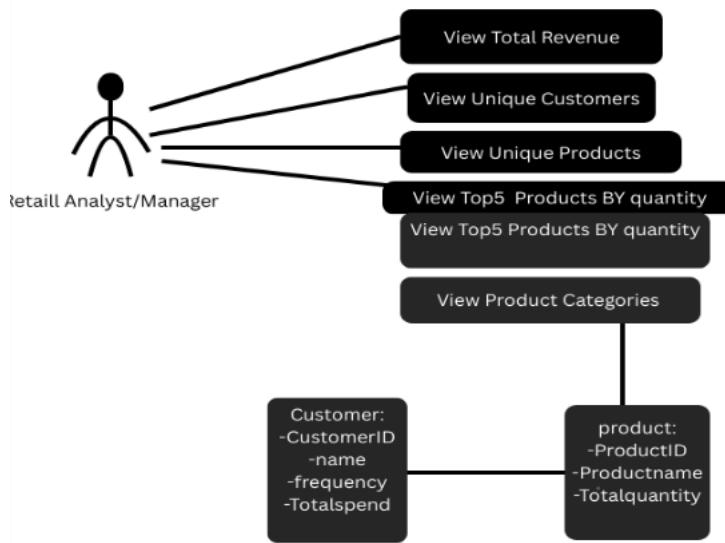
Project Overview:

The dashboard displays key metrics for a retail business, including unique customers, unique products, total revenue, top products, top customers, customer frequency, and revenue by product category. It is built to be interactive, visually appealing, and easy to use.

Use-case Explanations:

- **KPI Display:** Shows metrics such as total revenue, unique customers, and unique products.
- **Top Products:** Lists the top 5 products based on quantity sold.
- **Top Customers:** Lists the top 5 customers based on total spending.
- **Customer Frequency:** Shows purchase frequency for each customer.
- **Revenue by Category:** Displays revenue aggregated by product category in a table format.

UML DIAGRAM:



Front-end Interface Design:

- *Dashboard Header: Displays project title*
- *KPI Cards: Unique Customers, Unique Products, Total Revenue*
- *Tables: Top products, top customers, customer frequency, revenue by category*
- *Styling: Light gray background, white elevated cards, rounded corners, shadows, dark table headers, hover effects.*

CODE:

Flask-Backend(app.py)

```
◆ app.py > dashboard
 1  from flask import Flask, render_template
 2  import pandas as pd
 3
 4  app = Flask(__name__)
 5
 6  # Load dataset
 7  df = pd.read_csv(r"C:\Users\saisr\Desktop\retail_analysis\data\transactions.csv")
 8
 9  @app.route('/')
10  def dashboard():
11      # Summary KPIs
12      unique_customers = df["CustomerID"].nunique()
13      unique_products = df["ProductID"].nunique()
14      df["TotalAmount"] = df["Quantity"] * df["Price"]
15      total_revenue = df["TotalAmount"].sum()
16      avg_transaction_value = df["TotalAmount"].mean()
17
18      # Top 5 products by quantity
19      top_products = (
20          df.groupby(['ProductID', 'ProductCategory'])['Quantity']
21              .sum()
22              .reset_index()
23              .sort_values(by='Quantity', ascending=False)
24              .head(5)
25      )
26
27
28
29      # Top 5 customers by spending
30      top_customers = (
31          df.groupby(['CustomerID', 'ProductCategory'])['TotalAmount']
32              .sum()
33              .reset_index() # convert Series to DataFrame
34              .sort_values(by='TotalAmount', ascending=False)
35              .head(5) # top 5
36  )
```

HTML Template(index.html):

```

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Retail Analysis Dashboard</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body>
    <div class="container-fluid">
      <!-- Dashboard Header -->
      <div class="d-flex justify-content-between align-items-center bg-gray-800 text-white p-4 mb-4 rounded-lg">
        <h2 class="text-center text-3xl font-bold leading-tight">Retail Analysis Dashboard</h2>
      </div>
      <!-- Grids -->
      <div class="row text-center mb-4">
        <div class="col-md-4">
          <div class="card p-4 shadow-sm bg-white">
            <h3 class="text-muted">Unique Customers</h3>
            <div>
              {{ unique_customers }}</div>
            </div>
          </div>
        <div class="col-md-4">
          <div class="card p-4 shadow-sm bg-white">
            <h3 class="text-muted">Unique Products</h3>
            <div>
              {{ unique_products }}</div>
            </div>
          </div>
        <div class="col-md-4">
          <div class="card p-4 shadow-sm bg-white">
            <h3 class="text-muted">Total Revenue</h3>
            <div>
              ${{ total_revenue }}</div>
            </div>
          </div>
        </div>
      <!-- Top 5 Products -->
      <div class="card p-3 mb-4">
        <table class="table table-striped table-hover">
          <thead>
            <tr>
              <th>Product</th>
              <th>ProductCategory</th>
              <th>Total Quantity</th>
            </tr>
          </thead>
          <tbody>
            <% for row in top_products %>
            <tr>
              <td>{{ row['ProductID'] }}</td>
              <td>{{ row['ProductCategory'] }}</td>
              <td>{{ row['Quantity'] }}</td>
            </tr>
            <% endfor %>
          </tbody>
        </table>
      </div>
      <!-- Top Customers -->
      <div class="card p-3 mb-4">
        <h3 class="text-4xl font-bold text-white mb-3">Top 5 Customers (by spending)</h3>
      </div>
      <div class="table-container">
        <table class="table table-striped table-hover">
          <thead>
            <tr>
              <th>Customer ID</th>
              <th>Product Category</th>
              <th>Total Spend</th>
            </tr>
          </thead>
          <tbody>
            <% for row in top_customers %>
            <tr>
              <td>{{ row['CustomerID'] }}</td>
              <td>{{ row['ProductCategory'] }}</td>
              <td>${{ row['TotalAmount'] }}</td>
            </tr>
            <% endfor %>
          </tbody>
        </table>
      </div>
      <!-- Customer Frequency -->
      <div class="card p-3 mb-4">
        <h3 class="text-4xl font-bold text-white mb-3">Customer Frequency</h3>
      </div>
      <div class="table-container">
        <table class="table table-striped table-hover">
          <thead>
            <tr>
              <th>Customer ID</th>
              <th>Number of Purchases</th>
            </tr>
          </thead>
          <tbody>
            <% for row in category_revenue %>
            <tr>
              <td>{{ row['CustomerID'] }}</td>
              <td>{{ row['TotalAmount'] }}</td>
            </tr>
            <% endfor %>
          </tbody>
        </table>
      </div>
    </div>
  </body>
</html>

```

CSS Styling:

Style.css

```

static > style.css > top_products.csv > index.html > app.py

4 body {
5   background-color: #e0e0e0; /* light gray background */
6   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
7   font-size: 16px;
8   color: #000000;
9   background-color: f8f9fa;
10  margin: 0;
11  padding: 0;
12 }

/* ===== */
13 | Dashboard Header
14 ===== */
15 .dashboard-header {
16   background-color: #2a2a2a;
17   border: 2px solid #333;
18   color: #ffffff;
19   padding: 20px;
20   text-align: center;
21   border-radius: 12px;
22   margin-bottom: 20px;
23   text-align: center;
24   box-shadow: 0 3px 10px rgba(0,0,0,0.5);
25 }
26 .dashboard-container {
27   width: 100%;
28   max-width: 1700px;
29   margin: 0 auto;
30   overflow-x: auto;
31 }

style.css > top_products.csv > index.html > app.py

tatic > # style.css > h3
67 .table thead {
68   background-color: #111111;
69   color: #ffffff;
70   font-weight: 600;
71   text-align: center;
72 }
73 .table thead th {
74   font-size: 1.25rem !important;
75   font-weight: 700 !important;
76   color: #000000 !important;
77 }
78 /* --- ONLY Revenue by Category headings white --- */
79 body > div > div:nth-child(6) thead th {
80   color: #white !important;
81   font-weight: bold !important;
82   text-align: center;
83 }
84 }

85 table.table tbody tr td:first-child {
86   font-size: 1.3rem !important;
87 }

41 box-shadow: 0 4px 8px rgba(128, 136, 134, 0.5);
42 padding: 20px;
43 transition: transform 0.2s;
44 color: #000000;
45 }
46 .card:hover {
47   transform: translateY(-5px);
48 }

/* ===== */
50 ===== */
51 Table Styling
52 ===== */
53 .table {
54   border-collapse: separate;
55   border-spacing: 0;
56   border-radius: 8px;
57   overflow: hidden;
58   background-color: #2a2a2a;
59   color: #ffffff;
60 }
61 .table-container {
62   max-height: 300px;
63   width: 100%;
64   overflow-y: auto;
65   overflow-x: auto;
66 }

104 ===== */
105 .table th, td {
106   font-size: 1.1rem !important;
107   text-align: center;
108   vertical-align: middle;
109 }
110 /* ===== */
111 | KPIs Text
112 ===== */
113 h6 {
114   font-weight: 600;
115   color: #cccccc;
116   margin-bottom: 10px;
117   text-transform: uppercase;
118 }
119 }

120 h3 {
121   font-size: 2rem;
122   font-weight: 700;
123   color: #1f2937;
124   margin: 0;
125 }
126 }

127 /* ===== */
128 | Responsive Tables
129 ===== */
130

```

Explanation of the Code:

Backend: Loads CSV data, calculates KPIs, aggregates data for tables, passes it to HTML template

Frontend: Uses Bootstrap and Tailwind for layout, CSS for styling, Jinja templating to display dynamic data

Interactivity: Hover effects on tables, responsive layout, easy readability of metrics

Output with Explanation:

Dashboard Header and KPIs: Shows total revenue, unique customers, unique products

Table: Displays top 5 products by quantity

Top Customers Table: Displays top 5 customers by total spending

Customer Frequency Table: Displays number of purchases per customer

Revenue by Category Table: Displays revenue grouped by product category

Retail Analysis Dashboard

Unique Customers: 44

Unique Products: 28

Total Revenue: \$49680

Top 5 Products (by Quantity)

Product	ProductCategory	Total Quantity
P601	Grocery	77
P701	Clothing	24
P702	Clothing	3
P704	Clothing	3
P803	Electronics	3

Top 5 Customers (by spending)

Customer ID	Product Category	Total Spend
C131	Furniture	\$3900
C114	Furniture	\$3600
C139	Furniture	\$2600
C110	Furniture	\$2500
C127	Furniture	\$2400

Customer Frequency

Customer ID	Number of Purchases
C101	4
C102	4
C103	1
C104	1
C105	1

Revenue by Category

Product Category	Revenue
Furniture	\$25600
Electronics	\$17150
Clothing	\$5390
Grocery	\$1540

25°C Light rain

Search web & PC

16:31 03-09-2025



Bibliography:

1. **W3Schools.** "Python Pandas Tutorial." Available at:
<https://www.w3schools.com/python/pandas>
2. **GeeksforGeeks.** "Working with CSV Files in Python." Available at:
<https://www.geeksforgeeks.org/working-csv-files-python/>
3. **TutorialsPoint.** "HTML Tutorial." Available at:
<https://www.tutorialspoint.com/html/index.html>
4. **W3Schools.** "CSS Tutorial." Available at: <https://www.w3schools.com/css/>
5. **Bootstrap.** "Bootstrap 5 Documentation." Available at:
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>
6. **Python Software Foundation.** "Python Language Reference, version 3.10." Available at: <https://www.python.org>

Visual Studio Code. "Code Editing. Redefined." Available at: <https://code.visualstudio.com>