# Retail Business Performance & Profitability Analysis

Project Overview

This report analyzes retail sales performance using a structured transactional dataset.
The goal is to identify trends in revenue, profit, category-level performance, and regional behavior.
The report also highlights insights that help improve decision-making, inventory planning, and marketing efforts.

This project is designed to be simple, clear, and beginner-friendly while still demonstrating strong analytical skills.

Tools Used:

- Python (pandas, matplotlib, seaborn)
- SQL (SQLite / MySQL)
- Optional: Power BI / Tableau for dashboard

Deliverables:

- Notebook / Python script
- SQL queries
- Power BI/Tableau dashboard (structure included)
- Insights report
- README.md

README.md

# Retail Business Performance & Profitability Analysis

## Overview

This project analyzes retail sales performance using transactional data.

It explores revenue, profit, order trends, top-selling categories, and regional performance.

The goal is to help business owners identify strong areas and opportunities for growth.

## Project Structure

```
project/
│── data/
│   └── retail_sales.csv
│   ├── revenue_trend.png
│   ├── profit_trend.png
│   ├── category_sales.png
│   ├── region_sales.png
│── retail_analysis.py
│── queries.sql
│── insights.txt
│── dashboard
│── README.md
```

## Tools Used

- Python (pandas, matplotlib, seaborn)
- SQL (SQLite/MySQL)
- Power BI or Tableau

## Key Analysis Performed

- Monthly Revenue Trend
- Monthly Profit Trend
- Category Performance
- Region Performance
- Profitability Ratio
- Top 10 Products
- Customer Order Behavior

## How to Run

1. Install dependencies:

```
pip install pandas numpy matplotlib seaborn
```

2. Place `retail_sales.csv` inside the `data/` folder.

3. Run the script:

```
python retail_analysis.py
```

4. Dashboard images are saved in `images/`.

## Deliverables

- Clean dataset
- Python analysis script
- SQL query file
- Images for dashboard
- Insights report

## Final Notes

This project is simple, effective, and perfect for beginners wanting to showcase data analysis skill.

Dataset (retail_sales.csv) — Structure

Order_ID	Date	Product	Category	Region	Quantity	Price	Profit
1001	2023-01-01	Notebook	Office Supplies	East	3	12.50	5.10

SQL Queries (queries.sql)

```
-- Total Revenue
SELECT SUM(Price * Quantity) AS Total_Revenue FROM retail_sales;
-- Total Profit
SELECT SUM(Profit) AS Total_Profit FROM retail_sales;
```

```
-- Monthly Revenue Trend
SELECT strftime('%Y-%m', Date) AS month,
       SUM(Price * Quantity) AS revenue
FROM retail_sales
GROUP BY month
ORDER BY month;
-- Category Performance
SELECT Category,
       SUM(Price * Quantity) AS revenue
FROM retail_sales
GROUP BY Category
ORDER BY revenue DESC;
-- Region Sales
SELECT Region,
       SUM(Price * Quantity) AS revenue
FROM retail_sales
GROUP BY Region;
```

Python Script: retail_analysis.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Load data
df = pd.read_csv("data/retail_sales.csv", parse_dates=["Date"])

df["Revenue"] = df["Price"] * df["Quantity"]
df["Month"] = df["Date"].dt.to_period("M").dt.to_timestamp()
```

```
# Monthly Revenue
monthly_revenue = df.groupby("Month")["Revenue"].sum()
plt.figure(figsize=(10,5))
sns.lineplot(x=monthly_revenue.index, y=monthly_revenue.values)
plt.title("Monthly Revenue Trend")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("images/revenue_trend.png")
plt.close()

# Profit Trend
monthly_profit = df.groupby("Month")["Profit"].sum()
plt.figure(figsize=(10,5))
sns.lineplot(x=monthly_profit.index, y=monthly_profit.values, color="green")
plt.title("Monthly Profit Trend")
plt.xlabel("Month")
plt.ylabel("Profit")
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("images/profit_trend.png")
plt.close()

# Category Sales
cat_sales = df.groupby("Category")["Revenue"].sum()
plt.figure(figsize=(8,5))
sns.barplot(x=cat_sales.index, y=cat_sales.values)
```

```
plt.title("Sales by Category")
plt.xlabel("Category")
plt.ylabel("Revenue")
plt.tight_layout()
plt.savefig("images/category_sales.png")
plt.close()

# Region Sales
reg_sales = df.groupby("Region")["Revenue"].sum()
plt.figure(figsize=(8,5))
sns.barplot(x=reg_sales.index, y=reg_sales.values, palette="viridis")
plt.title("Sales by Region")
plt.xlabel("Region")
plt.ylabel("Revenue")
plt.tight_layout()
plt.savefig("images/region_sales.png")
plt.close()

print("Images generated successfully.")
```

Dashboard Visuals (description)

These will be saved into the images/ folder when you run the script:

1. **revenue_trend.png**
   Monthly revenue line chart
2. **profit_trend.png**
   Monthly profit line chart
3. **category_sales.png**
   Bar chart of revenue by category

4. **region_sales.png**
   Bar chart of revenue by region

If you want, I can create a ready-made Power BI layout too.

Insights (insights.txt)

1. Revenue consistently increases in Q3 and Q4, suggesting seasonal demand.

2. The Technology category contributes the highest revenue.

3. East and West regions show stronger performance than Central and South.

4. Profit margins vary by category; Office Supplies has lower margins than Technology.

5. Most orders fall between the 2–5 quantity range.