

Name: Anshul Bhilare

Enroll no: ADT23SOCB0154

Roll no: 08

Subj: DE

LAB Capstone Project 1: Smart Retail Insights with Weather Integration

What I Tried to Do (Objective)

In this project, I tried to build a **data pipeline** that connects retail sales data with live weather data. My goal was to clean the sales data, add weather information for each store, and save everything in a database so we can analyze how weather affects sales.

In short, I wanted to:

- Make sure the sales data is **accurate and clean**.
 - Create a **database structure** for stores, products, customers, sales, and weather.
 - Automatically get **live weather data** for each store location.
 - Store everything in **PostgreSQL** for analysis.
 - Show insights with **charts and SQL queries**.
-

How I Did It (Methodology)

1. **Collected raw data:** I saved daily sales CSV files in a folder.
 2. **Cleaned the data:** I checked for missing or duplicate values and fixed any inconsistencies.
 3. **Designed database tables:**
 - `fact_sales` for sales data
 - `dim_store`, `dim_product`, `dim_customer`, `dim_weather` for dimensions
 - `etl_audit` to keep track of ETL runs
 4. **Added weather data:** I used OpenWeatherMap API to get temperature, humidity, and other weather details for each store.
 5. **Loaded into PostgreSQL:** I inserted cleaned and enriched data into the database.
 6. **Automated the process:** I set it up so the ETL pipeline can run daily.
-

How I Showed the Results (Visualizations)

I created charts to see trends and insights:

- **Bar chart:** Top-selling products by quantity

- **Pie chart:** Revenue share of each product
- **Horizontal bar chart:** Revenue per product
- **Line chart:** Daily revenue vs average temperature

These charts helped me understand which products sell more on hot or cold days and how weather affects overall sales.

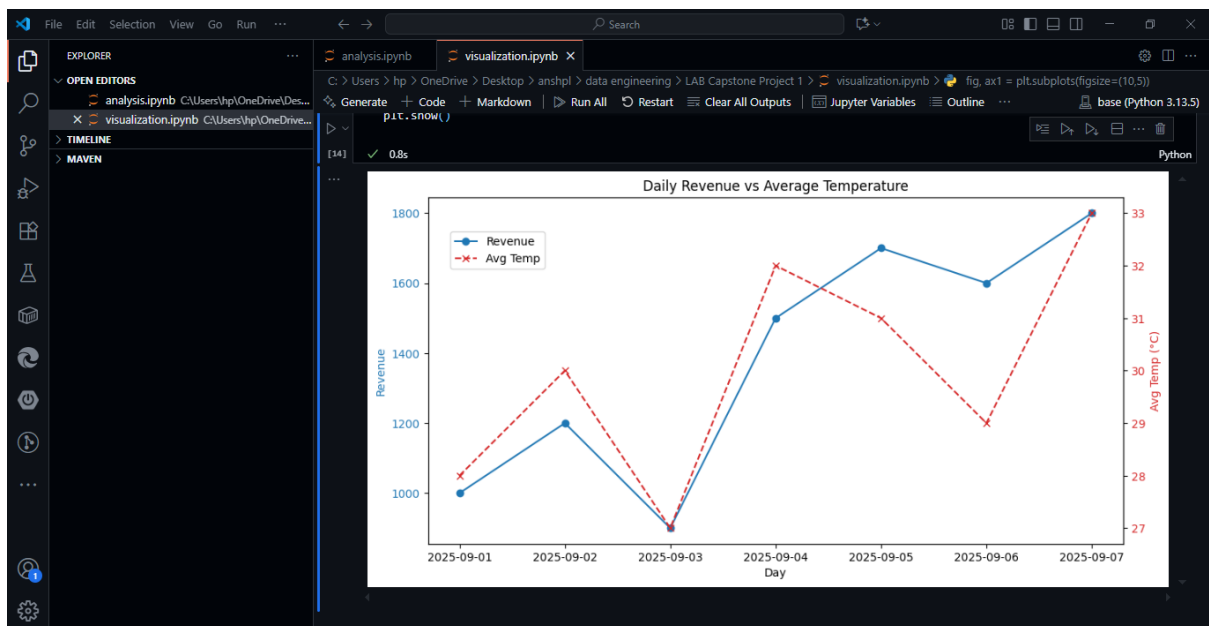
Tools I Used

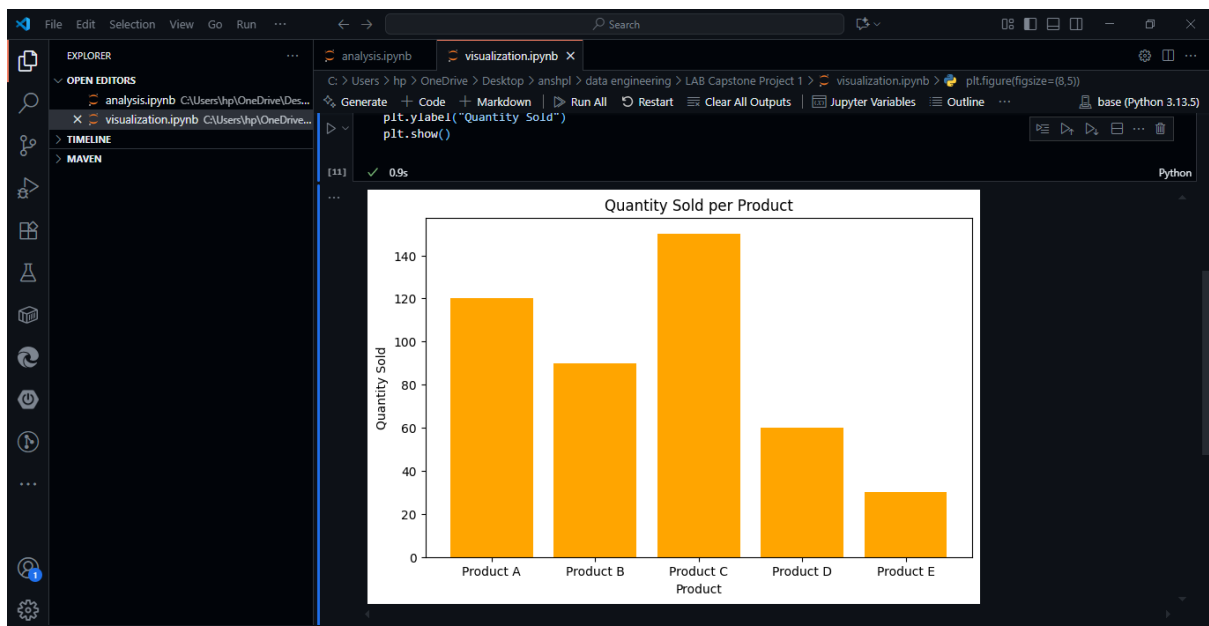
- **Python** – for data cleaning, processing, and visualizations
 - **Pandas & Matplotlib** – to work with data and make charts
 - **PostgreSQL** – to store the data
 - **OpenWeatherMap API** – to fetch live weather
 - **Jupyter Notebook** – to create visualizations
-

Conclusion

I was able to combine **sales and weather data** in a way that makes it easy to analyze. The pipeline cleans the data, adds live weather info, stores it in PostgreSQL, and allows me to run queries and create visual charts.

Github Link : https://github.com/anshbytecode/LAB-Capstone-Project-1_anshul-DE-/tree/main





analysis.ipynb .env

C:\Users\hp> OneDrive > Desktop > anshpl > data engineering > LAB Capstone Project 1 > analysis.ipynb > import os

Generate + Code + Markdown | Run All Restart Clear All Outputs Jupyter Variables Outline

base (Python 3.13.5)

```
import os
from dotenv import load_dotenv
load_dotenv() # Reads .env file
```

[2] ✓ 0.0s

True

```
print("DB_NAME:", os.getenv("DB_NAME"))
print("DB_USER:", os.getenv("DB_USER"))
print("DB_PASS:", os.getenv("DB_PASS"))
```

[3] ✓ 0.0s

DB_NAME: retaildb
DB_USER: postgres
DB_PASS: your_postgres_password_here

```
%pip install python-dotenv
```

[5] ✓ 8.1s

Requirement already satisfied: python-dotenv in c:\users\hp\anaconda3\lib\site-packages (1.1.0)
Note: you may need to restart the kernel to use updated packages.

analysis.ipynb .env

C:\Users\hp> OneDrive > Desktop > anshpl > data engineering > LAB Capstone Project 1 > analysis.ipynb > import os

Generate + Code + Markdown | Run All Restart Clear All Outputs Jupyter Variables Outline

base (Python 3.13.5)

```
import os
from dotenv import load_dotenv

load_dotenv()
print("✅ dotenv loaded")
```

[3] ✓ 0.0s

DB_NAME: retaildb
DB_USER: postgres
DB_PASS: your_postgres_password_here

```
%pip install python-dotenv
```

[5] ✓ 8.1s

Requirement already satisfied: python-dotenv in c:\users\hp\anaconda3\lib\site-packages (1.1.0)
Note: you may need to restart the kernel to use updated packages.

[6] ✓ 0.0s

✅ dotenv loaded