RETAIL ANALYTICS PROJECT – DATA ENGINEERING ASSIGNMENT

1. Objective

The goal of this project was to design and implement a data engineering pipeline for a retail analytics system. The pipeline performs data ingestion, cleaning, transformation, and reporting of store sales data. Finally, the processed data is visualized in Power BI through interactive dashboards to analyse performance metrics such as total revenue, profit, and sales trends.

2. Process Overview

Task 1 – Data Ingestion & Transformation

Created three datasets (stores.csv, products.csv, store_sales.csv) and loaded them into Azure Databricks using PySpark. Performed data cleaning, joined them into a single DataFrame (final_df), derived metrics like gross_amount, net_amount, and profit, and exported the cleaned data as cleaned_sales.csv for further analysis.

Task 2 – Data Pipeline and Logging

Developed an end-to-end ETL pipeline with stages for Ingestion, Transformation, and Storage. Implemented a logging system (log.txt) to record each stage's execution details and verified pipeline completion through Databricks logs.

Task 3 – Power BI Dashboard

Imported cleaned_sales.csv into Power BI, created DAX measures (Total Revenue, Profit Margin), and built an interactive dashboard with visualizations for store performance, top products, sales trends, and regional analysis. Ensured a clean layout with consistent colors, themes, and slicers for user interactivity.

Bonus – Daily Refresh & Summary Automation

A daily data refresh process can be simulated by appending new records with a load_date column using PySpark. This approach allows the dataset to automatically update each day, reflecting incremental data ingestion. Additionally, a simple script can be created to calculate and print a summary such as *"Yesterday's top store was <store_id> with total revenue of ₹<amount>."* The summary can also be logged in a file named summary_log.txt. This demonstrates how automated data pipelines can be designed to refresh data and generate daily insights similar to real-world enterprise workflows.

3. Challenges Faced

- Handling PySpark syntax errors and managing file paths in Databricks FileStore.
- Aligning visuals and formatting themes correctly within Power BI's limited customization options.
- Simulating automation within Databricks Community Edition (which lacks full scheduling support).

4. Learnings

- Understood how to design an end-to-end ETL pipeline using PySpark.
- Learned to connect Databricks outputs with Power BI for business analytics.
- Experienced data cleaning, transformation, and visualization workflows similar to a real Data Engineer project.
- Gained clarity on automation, logging, and data refresh strategies used in professional environments.

5. Outcome

Successfully built a **Retail Analytics Dashboard** with data-driven insights on sales, profit, and regional performance.