🔥 Your PySpark Project: "Retail Sales Data Pipeline"

🛒 Project Title:

"Retail Data Analysis with PySpark: ETL & Insights Pipeline"

📦 Project Summary:

You'll build an ETL pipeline using PySpark that:

- Ingests raw **retail sales CSV data**
- Performs transformations: cleaning, aggregations, filtering
- Applies window functions and joins to analyze customer behavior
- Generates final output reports as CSV/Parquet for dashboards

🗺️ Project Roadmap (7 Days — Full Hands-on)

Day	Phase	Goals
1	**Setup & Data Ingestion**	Install PySpark, load sample data, setup dev environment
2	**Data Cleaning & Schema Design**	Handle nulls, fix types, define schema
3	**Transformations & Aggregations**	Apply select, filter, groupBy, agg, joins
4	**Window Functions & Business Logic**	Use rank, dense_rank, lag, rolling sum

5	**Customer Insights & Analysis**	Build reports: CLTV, repeat customers, sales trends
6	**Output & Save Results**	Save as Parquet/CSV files, generate plots (optional)
7	**Documentation & GitHub Setup**	Push code to GitHub with README + pipeline diagram

🧪 Breakdown of Each Day

📍 Day 1 – Setup + Ingest

- Install PySpark locally or use Google Colab (with Spark magic)
- Load CSV file into a Spark DataFrame
- Print schema and preview data

📍 Day 2 – Cleaning + Schema

- Define schema explicitly (`StructType`)
- Handle missing/null values
- Cast columns to correct types (`date`, `float`, `int`)
- Create additional columns (like `total_price = unit_price * quantity`)

📍 Day 3 – Transform + Aggregate

- `groupBy("category").agg(sum("total_price"))`
- `groupBy("customer_id").count()`
- Join with customer data if available
- Filter data (e.g., top 5 cities by revenue)

📍 Day 4 – Window Functions

- Use `Window.partitionBy().orderBy()` to:
    - Get top 3 products by revenue per category
    - Get previous purchase date per customer (`lag`)
    - Calculate running total sales over time

📍 Day 5 – Insights & Use Cases

- Customer Lifetime Value (CLTV)
- Repeat Customers (order count > 1)
- Top categories/products by month
- Daily sales trend

📍 Day 6 – Save & Output

- Write output DataFrames to:
    - CSV
    - Parquet
- (Optional) Generate plots using pandas/matplotlib

(Optional) Store in GCS or local folder structure like:

```
/data/raw/

/data/processed/

/data/output/
```

-

📍 Day 7 – Polish + GitHub Upload

- Write README:

    - Project intro
    - Dataset link
    - Pipeline steps
    - Sample outputs
    - Screenshots (optional)
- Add pipeline diagram using draw.io or Excalidraw
- Push to GitHub with folder structure

📈 Bonus Challenge (Optional After Main Project)

- Run the same ETL using **SparkSQL**: register your DataFrame as a temporary view and reuse your SQL queries.
- Scale the project by adding **multiple CSVs** and reading them as a stream (to simulate real-time).

📘 Example Final Folder Structure

```
retail-data-pipeline/
├── data/
│   ├── raw/
│   ├── processed/
│   └── output/
├── notebooks/
│   └── retail_analysis.ipynb
├── scripts/
│   └── etl_pipeline.py
```

```
├── README.md
└── pipeline_diagram.png
```