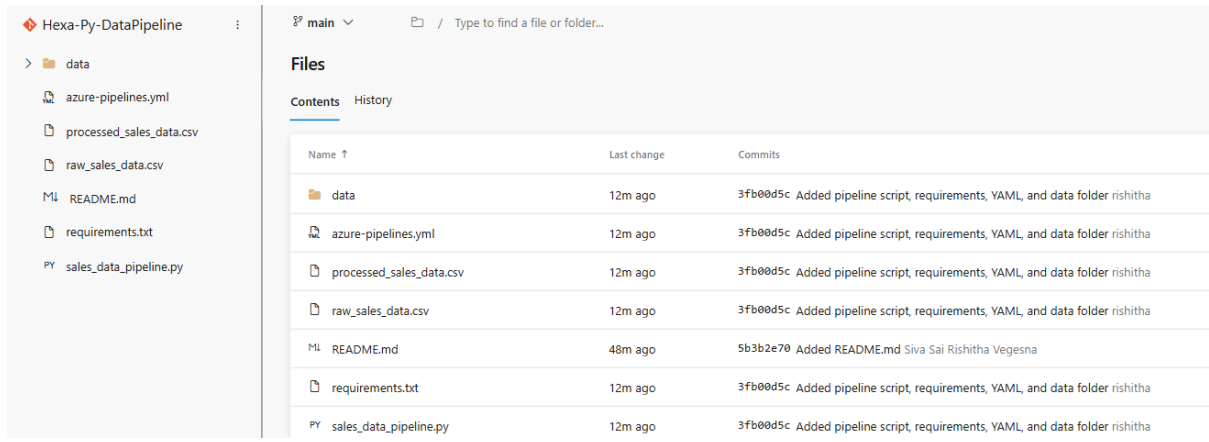


TASK-2 SCREENSHOTS

REPO STRUCTURE:



The screenshot shows a file explorer interface for a repository named 'Hexa-Py-DataPipeline'. The left sidebar lists the files and folders: 'data', 'azure-pipelines.yml', 'processed_sales_data.csv', 'raw_sales_data.csv', 'README.md', 'requirements.txt', and 'sales_data_pipeline.py'. The main area displays a table of files with columns for Name, Last change, and Commits.

Name	Last change	Commits
data	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha
azure-pipelines.yml	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha
processed_sales_data.csv	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha
raw_sales_data.csv	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha
README.md	48m ago	5b3b2e70 Added README.md Siva Sai Rishitha Vegesna
requirements.txt	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha
sales_data_pipeline.py	12m ago	3fb00d5c Added pipeline script, requirements, YAML, and data folder rishitha

Local VS Code terminal showing script run success.

```
>>
[✓] Files created: raw_sales_data.csv, processed_sales_data.csv
[Blob Upload] Skipped (no Azure credentials): raw_sales_data.csv
[Blob Upload] Skipped (no Azure credentials): processed_sales_data.csv
PS C:\Users\Rishitha\Desktop\Hexa-Py-DataPipeline>
```

Showing all the files

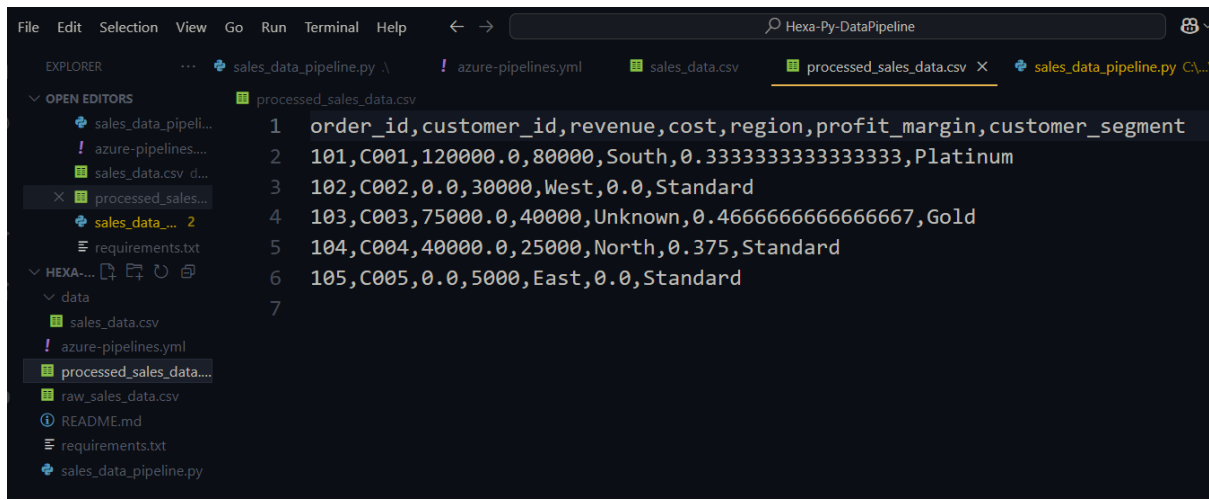
```
PS C:\Users\Rishitha\Desktop\Hexa-Py-DataPipeline> dir

Directory: C:\Users\Rishitha\Desktop\Hexa-Py-DataPipeline

Mode                LastWriteTime         Length Name
----                -
d-----          25-08-2025   15:28             data
-a----          25-08-2025   15:44             844 azure-pipelines.yml
-a----          25-08-2025   16:09             308 processed_sales_data.csv
-a----          25-08-2025   16:09             204 raw_sales_data.csv
-a----          25-08-2025   15:26             985 README.md
-a----          25-08-2025   15:30              28 requirements.txt
-a----          25-08-2025   15:43            2685 sales_data_pipeline.py

PS C:\Users\Rishitha\Desktop\Hexa-Py-DataPipeline>
```

Cleaned processed_sales_data.csv



```
1 order_id,customer_id,revenue,cost,region,profit_margin,customer_segment
2 101,C001,120000.0,80000,South,0.3333333333333333,Platinum
3 102,C002,0.0,30000,West,0.0,Standard
4 103,C003,75000.0,40000,Unknown,0.4666666666666667,Gold
5 104,C004,40000.0,25000,North,0.375,Standard
6 105,C005,0.0,5000,East,0.0,Standard
7
```

BONUS QUESTIONS

1. Why is storing cleaned data in Azure Blob Storage important for real-time pipelines?

- Azure Blob Storage acts as a centralized, scalable, and cost-effective data lake.
- Storing cleaned data ensures that downstream systems (like Databricks, Synapse, Power BI, or ML models) consume consistent and high-quality data.
- It enables real-time analytics because multiple services can access the same “single source of truth” simultaneously.
- It also supports versioning, durability, and global accessibility, which are critical for production pipelines.

2. What’s the difference between pipeline artifacts and Blob Storage uploads?

- Pipeline Artifacts:
 - Used inside Azure DevOps to share files between pipeline stages/jobs.
 - Temporary, for CI/CD workflow execution.
 - Accessed only within Azure DevOps.
- Blob Storage Uploads:

- Stored in Azure Storage account, outside DevOps.
- Persistent, production-ready storage.
- Accessible by applications, analytics platforms, and external services.

3. How would you handle failures in file uploads in a production setup?

- **Retry mechanism:** Implement automatic retries with exponential backoff.
- **Error handling & logging:** Log all upload attempts and errors for debugging.
- **Alerts/Monitoring:** Use Azure Monitor / Application Insights to trigger alerts when uploads fail.
- **Idempotent uploads:** Always overwrite or use unique file names to avoid duplicates.
- **Fallback strategy:** Store failed uploads in a “dead-letter queue” or backup location for later reprocessing.
- **Fail pipeline stage if critical:** Ensure pipeline doesn’t silently succeed if uploads are mandatory.