SwiftSupply Logistics Azure Data Platform

# 1. Project Overview

**Company Name:** SwiftSupply Logistics Pvt Ltd (Fictional)  
**Industry:** Retail & E-commerce supply chain management  
**Project Goal:** Build an Azure Data Platform to consolidate batch and streaming data from warehouses to improve inventory forecasting and shipment monitoring.

**Architecture Diagram (Placeholder):**  
[Insert diagram: ADLS → Bronze/Silver/Gold layers → Databricks → Power BI]

**Data Flow Diagram (Placeholder):**  
[Insert diagram: Batch CSV + IoT JSON → Bronze → Silver → Gold → Power BI]

# 2. Data Sources

## A. Batch Data (Historical)

**Files:** inventory.csv, sales\_orders.csv  
**Columns:** - inventory.csv: warehouse\_id, product\_id, product\_name, category, quantity\_in\_stock, last\_restock\_date - sales\_orders.csv: order\_id, product\_id, quantity\_sold, sale\_date, warehouse\_id

**Source:** Kaggle “Retail Product Sales” dataset or manually created CSVs (10–50 rows)

**Sample Screenshot (Placeholder):**  
[Insert screenshot of inventory.csv in Excel or ADLS]

## B. Streaming Data (IoT Shipments)

**Files:** JSON format, generated via iot\_stream\_simulator.py **Schema:** shipment\_id, warehouse\_id, status, location, timestamp **Status Values:** In Transit, Delivered, Delayed **Location:** Warehouse cities or random GPS coordinates

**Sample Screenshot (Placeholder):**  
[Insert screenshot of sample shipment JSON file]

# 3. Transformation Layers

| Layer | Input | Transformation Logic | Output Format |
| --- | --- | --- | --- |
| Bronze | Raw CSV / JSON | Load as-is into ADLS, maintain original schema | Delta |
| Silver | Bronze Delta | Remove nulls, convert dates to yyyy-MM-dd, numeric columns to integers | Delta |
| Gold | Silver batch + streaming | Join on warehouse\_id, filter In Transit & Delayed, add stock\_status (Low Stock <50), alert delayed shipments, retain last 90 days | Delta |

**Sample Screenshots (Placeholder):**  
[Insert screenshot: Bronze Delta table in Databricks]  
[Insert screenshot: Silver Delta table with transformations]  
[Insert screenshot: Gold Delta curated table ready for BI]

# 4. Azure Data Factory Pipelines

## Batch Pipeline

* Trigger: Daily
* Steps:
  1. Check new files in S3
  2. Copy to ADLS /raw/batch
  3. Databricks Notebook: Bronze → Silver → Gold
  4. Archive processed files

**Screenshot Placeholder:**  
[Insert screenshot of ADF Batch Pipeline workflow]

## Streaming Pipeline

* Trigger: Continuous
* Steps:
  1. Event Hub → ADLS /raw/streaming
  2. Databricks Streaming Job: Bronze → Silver → Gold
  3. Save curated Delta for BI

**Screenshot Placeholder:**  
[Insert screenshot of ADF Streaming Pipeline workflow]

## Gold Pipeline

* Merge Silver batch and streaming
* Apply alert and stock rules
* Save managed Gold Delta tables

**Screenshot Placeholder:**  
[Insert screenshot of ADF Gold Pipeline or Databricks Gold Notebook execution]

# 5. Utilities

* **adls\_utils.py:** Upload/download ADLS, move processed files to archive
* **azure\_keyvault\_utils.py:** Fetch credentials from Key Vault
* **data\_validation.py:** Check nulls, schema, date formats

**Screenshot Placeholder:**  
[Insert screenshot of code snippet from utilities folder]

# 6. Notebooks

* bronze\_to\_silver\_batch.ipynb: Batch CSV → Bronze → Silver Delta
* silver\_to\_gold\_batch.ipynb: Merge batch and streaming Silver → Gold Delta
* bronze\_to\_silver\_streaming.ipynb: Event Hub JSON → Bronze → Silver
* silver\_to\_gold\_streaming.ipynb: Merge streaming Silver with batch, apply alerts, save Gold

**Screenshot Placeholder:**  
[Insert screenshot of notebook showing transformation logic]

# 7. Git Repository Structure

swift-supply-data-platform/  
│  
├── README.md  
├── docs/  
│ └── SwiftSupply\_Project\_Overview.docx  
├── src/  
│ ├── batch/  
│ │ ├── notebooks/  
│ │ └── scripts/  
│ ├── streaming/  
│ │ ├── notebooks/  
│ │ └── scripts/  
│ └── utilities/  
├── data/  
│ ├── raw/  
│ ├── processed/  
│ └── curated/  
├── pipelines/  
├── tests/  
└── requirements.txt

**Git Guidelines:** - Branches: main (prod), dev (development), feature/ (features) - Commit messages: [feature|fix|docs] - .gitignore: **pycache**/, *.pyc,* .log, .ipynb\_checkpoints/

# 8. Testing

* test\_batch\_transforms.py: Validate batch transformations
* test\_stream\_transforms.py: Validate streaming parsing & transformations
* Use pytest or unittest

**Screenshot Placeholder:**  
[Insert screenshot of test results or Pytest execution]

# 9. Business Rules

1. Daily Inventory Update: quantity\_in\_stock < 50 → Low Stock
2. Shipment Delay Alert: status = Delayed → alert CSV /curated/alerts/
3. Retention: Keep last 90 days of shipment data

# 10. Screenshots & Documentation

* Include screenshots of Databricks notebooks, ADF pipelines, Delta Lake tables
* Highlight transformations, curated Gold tables, alerts
* Placeholder images included throughout document for visual aid

# 11. References

* Azure Databricks Docs
* Azure Data Factory Docs
* Delta Lake Docs
* Event Hub Python SDK