

ChatGPT

You said:

ChatGPT said:

Got it 👍 — this is a TCS AI Fridays – AI for SDLC exercise document.

Here's a summary of what it's asking you to do:

Use Story: Smart Retail Operations Platform

Goal:

Develop a simple AI-based inventory

classification tool using **ABC analysis**.



Objective

Build a basic software module that helps retail teams classify inventory items (A, B, C) based on their value and quantity, so they can manage stock effectively.



ABC Classification Logic

- **A items:** High value, low quantity → need tight control (e.g., premium electronics).
- **B items:** Moderate value & quantity → need regular monitoring.
- **C items:** Low value, high quantity → minimal oversight (e.g., accessories).



Key Features

- **Rule Engine:** Define rules for A/B/C categorization (e.g., value thresholds).
- **Manual Data Entry:** Input item name, quantity, and unit cost.
- **Automatic Classification:** System auto-assigns category (A/B/C).
- **Item Threshold:** Define reorder and max inventory limits per category.
- **Dashboard:** Display categorized items with filters and totals.
- **Export Option:** Download categorized data as CSV/PDF.





Benefits

- Prioritize inventory control easily.
- Reduce overstock of low-value items.
- Improve focus on high-impact (A) items.

If you'd like, I can help you **design a working AI prototype** for this — e.g. using **Python (Streamlit + Pandas)** or a **low-code AI tool** (like Gemini or ChatGPT API).

Would you like me to create:

-  a **Python Streamlit prototype** (with data upload, ABC classification & dashboard), or
-  a **flow diagram + AI**

design explaining the logic for presentation?

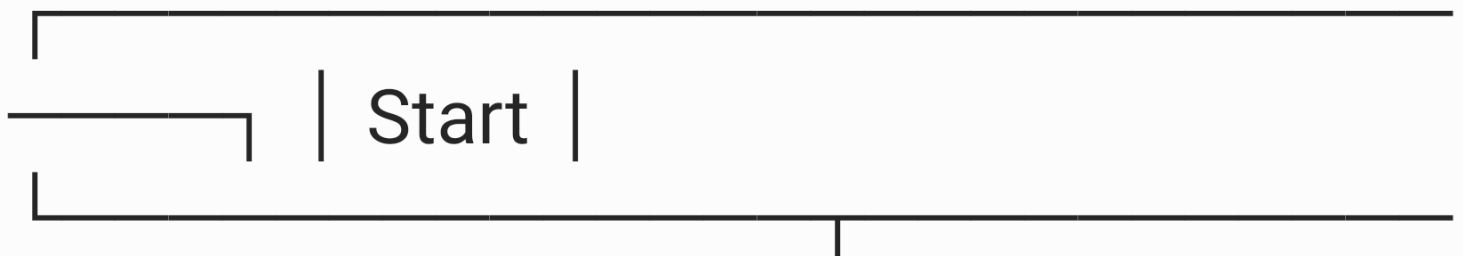
You said:

A flow diagram

ChatGPT said:

Perfect 👍 — here's a clear **flow diagram** for your TCS AI Fridays use case: **"Smart Retail Operations Platform – ABC Inventory Classification using AI"**

 **Flow Diagram (Text Representation)**



▼

User Inputs Item Data
(Item Name, Quantity, Unit Cost)

▼

Calculate Total Value
per Item → $\text{Total Value} = \text{Quantity} \times \text{Cost}$

▼

Apply Rule Engine / AI
Model • Determine thresholds for A/B/
C • Use rules or ML-based classifier

▼

Automatic
Classification → Assign item as A / B

/ C |

 ▼ | Check Inventory

Thresholds | | • Min / Max reorder levels

 ▼ | Update Dashboard |

| • Show categorized list | | • Summary

totals & filters |

 ▼ | Export Option | | •

Download CSV / PDF report |

 ▼



Optional AI Integration Ideas

You can mention these in your explanation:

- **AI rule optimization:** Use machine learning to dynamically adjust A/B/C thresholds based on historical sales data.
- **Predictive reorder suggestion:** Use regression or time-series models to forecast stockouts.
- **Natural language interface:** Allow user to query inventory via chat (e.g., “Show all A items below reorder level”).

Would you like me to create a **visual version (diagram image)** of this flow for presentation or submission?

You said:

ChatGPT said:

Thanks — this second page provides the **instructions and deliverables** for your TCS AI Fridays task.

Here's a clear summary of what it says (so you can combine it with the first page for your submission):



TCS AI Fridays – AI for SDLC

Instructions


Use AI Tools or Build Your Own AI Agents (BYOAA) to develop the software and deliverables.

Your team must:

- Use AI tools (like ChatGPT, Gemini, Copilot, or other AI coding assistants)
- Or **build your own simple AI agent** to automate parts of the development lifecycle (like rule engine, UI design, test cases, or documentation).

Deliverables

You are expected to produce the following:

- User Persona
- User Story
- Functional Flow (→ the flow diagram we discussed fits here )
- UX/UI Design
- Data Design / Data Models
- Implementation Document
- Working Software
- Production-Level Quality Code
- Test Design
- Production Test Scripts

Evaluation Criteria

Area	Focus
Selection of Tools	Use AI tools best suited for your scenario
Quality Standards	Apply consistent coding patterns, high-quality design, code reviews
Integration	Integrate with other systems or databases
Testing	Include test cases and functional validation
Documentation	Provide detailed documentation of design, models, and data
User Experience	Simple, intuitive interface with smooth navigation

Tips for a Good Submission

- ✓ Keep UI simple and responsive
- ✓ Focus on working prototype, not just concept
- ✓ Document rule engine, logic, and AI

model clearly



Provide screenshots or live demo link if possible



Export dashboard as CSV/PDF as mentioned in use case