Test

### **Objective**

Design and implement an LLM-powered agent to manage E-Commerce product data and frequently asked questions. The agent should:

* Accurately retrieve product statistics based on user queries with filter options.
* Answer general customer queries using a predefined FAQ dataset.
* Utilize LangChain tools to route queries to the appropriate functions effectively.

### **Input Data**

1. **Product\_Statistics.csv** Contains product data with the following columns:
   * Product ID, Name, Category, Price, Sales Count, Rating, Stock Level, etc.
2. **FAQ.csv** Contains frequently asked questions and their respective answers:
   * Question, Answer

### **Functional Requirements**

#### **1. Product Information Retrieval**

* Implement a function that allows querying the product statistics using filters such as:
  + Category
  + Price Range
  + Stock Level
  + Rating
* Ensure support for natural language input like:  
   “Show me top-rated electronics under $500 in stock.”

#### **2. FAQ Handling**

* Implement a function to answer general user queries using the FAQ dataset.
* Match user queries to the most relevant FAQ and return the corresponding answer based on the closest question found in the dataset.

#### **3. Framework Integration**

* Create compatible tools in any Framework or Library of your choice.
* Build a basic Agent or WorkFlow that uses the appropriate tool based on user intent.
* Use prompt engineering to guide the agent’s decision-making and ensure clarity.

### **Deliverables**

* Python scripts implementing the solution. Please make sure your code is properly modularized.
* Mentioned interaction flows demonstrating:
  + Product search with multiple filters.
  + Correct FAQ retrieval.
  + Concise System Instructions
  + Agentic Orchestration selecting tool / tools based on input.
* Evaluation Criteria of the proposed pipeline in accordance to responses, tool selection, retrieval and edge cases.
* **Bonus:** Create a Frontend with Streamlit or Gradio
* README with setup instructions and usage examples.