

# AI-Powered Product Support Assistant Architecture

## Problem Statement:

ShopWise Solutions, an e-commerce platform in Austin, Texas, aims to enhance customer support through an AI-powered product support assistant. This assistant will serve as the first line of support for customer inquiries regarding products, orders, returns, and refunds, all by interfacing with our existing e-commerce database.

## Requirements:

ShopAssistant AI capabilities:

- Natural Language Understanding
- Database Integration
- Personalized Responses
- Multi-Turn Dialogues
- Avoidance of hallucinations
- Order Management

## Final Decision:

Implement an AI-powered conversational chatbot using:

- OpenAI GPT-4o
- LangChain for agent-based interactions
- Streamlit for web interface

## Brainstorming Sessions (*Journey to Final Decision*):

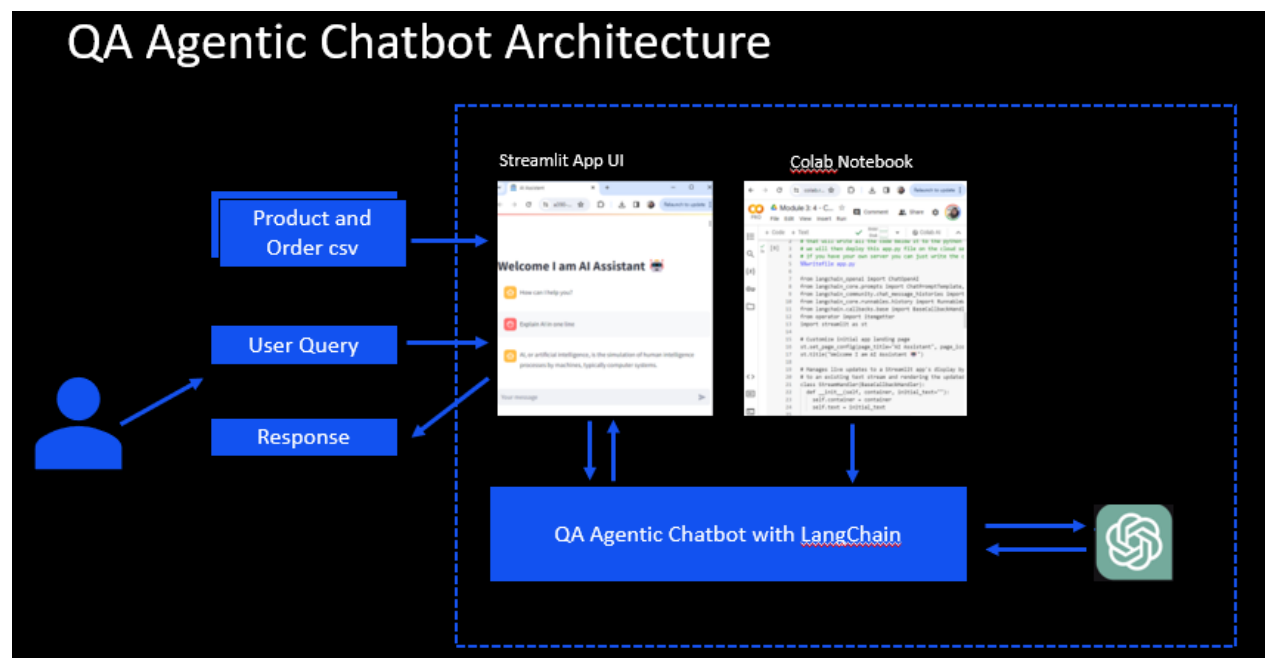
- I. **The Problem looked like a RAG solution but it wasn't.**
  - In our initial discussion after studying the problem statement, we thought that a basic RAG Chatbot would help solve this problem.
  - After using Open AI Assistant platform for quick testing, we realized that the RAG won't work here for the following reasons:
    1. The dataset is a structured data in a CSV format, where we want to filter or compare as we would do in a SQL or Python.
    2. It's not a text-based file where we want to retrieve information.
    3. Every row will become a chunk and that will increase the size as the dataset will become larger.
- II. **AI Agents**
  - We brainstormed if we want to make a chatbot to retrieve order and product information. A human would do writing a SQL or Python query and fetch the data.
  - If Agents given access to these tools and functions then they would also perform these actions.
  - We used LangChain Framework for Agent based interactions and OpenAI's LLM.

### III. Which LLM to Choose?

- We started out with cheaper versions of GPTs for cost effectiveness since there will be many customers asking multiple questions. For e.g. GPT-4o-mini and it's versions.
- But GPT-4o-mini was not performing on Agentic framework. It was taking time to respond and many times the answer was incorrect.
- Tried GPT-4 but the results were not satisfactory.
- Hence, we finalized on GPT-4o, the model had good contextual understanding and able to answer product comparison effectively.

### Architecture Components

- Large Language Model: OpenAI GPT-4o
- Framework: LangChain
- Web Interface: Streamlit
- Deployment: Google Colab and Tunnelling through NGROK



### Key Capabilities

- Natural language understanding
- Personalized responses
- Multi-turn dialogue management
- Order status tracking
- Product information retrieval

## **Future Considerations**

1. Implement robust error handling
2. Develop comprehensive fallback mechanisms
3. Develop guardrails