GenAI-Based Shopping Assistant: ShopAssist AI

1. Background

In today's fast-paced digital age, online shopping has become the preferred method for consumers worldwide. However, the sheer volume of choices, coupled with the lack of personalized guidance, often makes the shopping experience overwhelming and time consuming. To address these challenges, we have developed **ShopAssist AI**, an intelligent chatbot that combines the power of **large language models (LLMs)** and **rule-based functions** to deliver accurate, personalized, and reliable product recommendations.
ShopAssist AI is designed to simplify the online shopping experience by acting as a virtual shopping assistant. It leverages advanced natural language processing (NLP) capabilities to understand user preferences, ask relevant questions, and provide tailored recommendations. By integrating conversational AI with a structured dataset, ShopAssist AI ensures that users find the best products that match their needs.

2. Problem Statement

Given a dataset containing detailed information about laptops (e.g., product names, specifications, descriptions, etc.), the goal is to build a chatbot that can:

1. **Parse the dataset** to understand product features and descriptions.
2. **Engage in natural conversations** with users to gather their requirements (e.g., budget, preferred specifications, usage scenarios).
3. **Provide accurate and personalized laptop recommendations** based on the user's inputs.
4. **Handle user queries** efficiently and guide them through the decision-making process.

The chatbot must balance the flexibility of conversational AI with the precision of rule-based logic to ensure reliable and relevant recommendations.

3. Approach

The development of ShopAssist AI follows a structured approach to ensure seamless functionality and user satisfaction:

3.1 Conversation and Information Gathering

• The chatbot uses **large language models (LLMs)** to engage users in natural, human-like conversations.• It asks targeted questions to gather essential information about the user's preferences,
such as:

o Budget range
o Desired screen size
o Processing power (e.g., Intel i5, i7, AMD Ryzen)
o RAM and storage requirements
o Intended usage (e.g., gaming, work, casual use)

- The conversational flow is designed to be intuitive and user-friendly, ensuring that users feel guided rather than interrogated.

3.2 Information Extraction

- Once the chatbot collects sufficient information, **rule-based functions** are employed to extract the most relevant data from the dataset.
- These functions analyze the dataset (laptop_data.csv) and filter out laptops that match the user's requirements.
- The system identifies the **top 3 laptops** that best align with the user's preferences.

3.3 Personalized Recommendation

- The chatbot engages in further dialogue with the user to refine the recommendations.
- It provides detailed explanations for each recommendation, highlighting key features and benefits.
- Users can ask follow-up questions, and the chatbot will respond with accurate and context-aware answers.

4. System Functionalities

ShopAssist AI offers a range of functionalities to enhance the user experience:

4.1 User Interface

- A **user-friendly web interface** allows users to interact with the chatbot seamlessly.
- The interface is designed to be intuitive, ensuring that users of all technical levels can navigate it effortlessly.

4.2 Conversational AI

- The core of ShopAssist AI is powered by **OpenAI's chat model**, which enables natural and engaging conversations.
- The chatbot guides users through the process of specifying their requirements and provides real-time responses.

4.3 User Input Moderation

• To ensure a safe and respectful interaction, user inputs are moderated using **OpenAI's moderation API**.
• Inappropriate or offensive content is flagged, and the chatbot responds appropriately to maintain a positive user experience.

4.4 User Profile Extraction

• The chatbot extracts key information from the conversation to build a **user profile**.
• Using **OpenAI's function calling mechanism**, the chatbot converts user requirements into a structured JSON object, which is then used to filter and recommend products.

4.5 Dataset Integration

• The system leverages a dataset (laptopData.csv) that contains detailed information about various laptops, including:
o Product names
o Specifications (e.g., processor, RAM, storage, screen size)
o Descriptions (e.g., usage scenarios, unique features)
• The chatbot parses the Description column using LLMs to provide context-aware Recommendations.

5. System Architecture

ShopAssist AI follows a **client-server architecture** to ensure scalability and efficiency:

5.1 Client-Side

• Users interact with the chatbot through a **web-based interface**.
• The interface is hosted on a server and accessible via a browser.

5.2 Server-Side

• The server runs a **Flask application** that handles user requests and interacts with external APIs.
• The Flask application performs the following tasks:
o **Routing:** Maps user requests to appropriate functions based on URLs.
o **Conversation Management:** Manages conversation initiation, response generation, and history maintenance.
o **User Input Processing:** Captures user input, performs moderation checks, and extracts user profiles.o **Recommendation Logic:** Compares user profiles with laptop data and generates

personalized recommendations.

5.3 External APIs

- The system interacts with **OpenAI’s API** for:
  - **Conversation generation:** Generating natural and context-aware responses.
  - **Moderation:** Ensuring user inputs are appropriate.
  - **Function calling:** Converting user requirements into structured JSON objects.

5.4 Data Storage

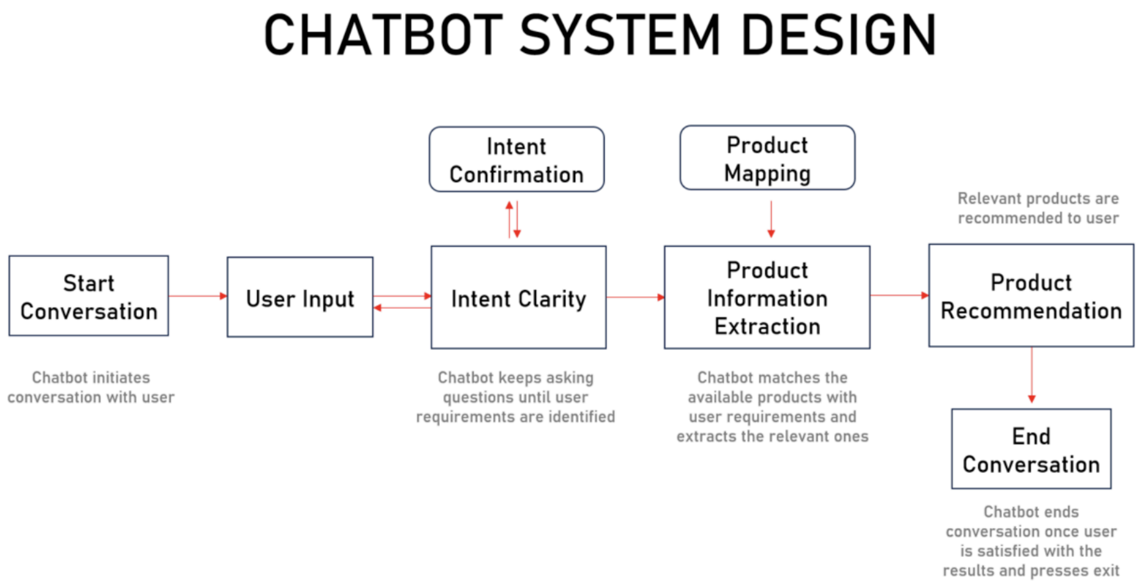
- The laptop dataset (laptopData.csv) is stored externally and accessed by the Flask application for filtering and recommendation purposes.

Chatbot System Design

Workflow Overview

1. **Start Conversation**
   - The chatbot initiates a conversation with the user.
2. **User Input**
   - The chatbot asks a series of questions to gather the user's requirements.
3. **Intent Confirmation**
   - The chatbot confirms the user's intent by asking follow-up questions to ensure all requirements are accurately identified.
4. **Product Mapping**
   - The chatbot maps the user's requirements to the available products in the dataset.
5. **Product Information Extraction**
   - The chatbot extracts relevant product information based on the user's requirements.
6. **Product Recommendation**
   - The chatbot recommends the most relevant products to the user.
7. **End Conversation**
   - The chatbot ends the conversation once the user is satisfied with the recommendations and chooses to exit.

This structured workflow ensures that the chatbot effectively gathers user requirements, provides accurate product recommendations, and delivers a seamless user experience.


CHATBOT SYSTEM DESIGN
Intent Confirmation
Product Mapping
Relevant products are recommended to user
Start Conversation
User Input
Intent Clarity
Product Information Extraction
Product Recommendation
Chatbot initiates conversation with user
Chatbot keeps asking questions until user requirements are identified
Chatbot matches the available products with user requirements and extracts the relevant ones
End Conversation
Chatbot ends conversation once user is satisfied with the results and presses exit

6. Implementation Details

6.1 Flask Application Functionalities

The Flask application is the backbone of ShopAssist AI and includes the following key Functionalities:

6.1.1 Routing
- Maps user requests to appropriate functions based on URLs.
- Ensures seamless navigation and interaction.

6.1.2 Conversation Management
- Handles the initiation of conversations, response generation, and maintenance of conversation history.
- Uses OpenAI's chat model to generate natural and engaging responses.

6.1.3 User Input Processing
- Captures user input and performs moderation checks using OpenAI's moderation API.
- Extracts user profiles by converting user input strings into JSON objects using OpenAI's function calling mechanism.

6.1.4 Recommendation Logic
- Compares user profiles with laptop data to identify the top 3 recommendations.
- Validates recommendations and generates recommendation text for display.

6.2 Major Functions

- **initialize_conversation():** Initializes the conversation with a system message.
- **get_chat_completions():** Takes the ongoing conversation as input and returns the assistant’s response.
- **moderation_check():** Checks if user or assistant messages are inappropriate and ends the conversation if necessary.
- **intent_confirmation_layer():** Evaluates whether the chatbot has accurately captured the user’s profile.
- **dictionary_present():** Checks if the user’s profile is returned as a Python dictionary.
- **compare_laptops_with_user():** Compares the user’s profile with laptop data and returns the top 3 recommendations.
- **initialize_conv_reco():** Initializes the recommendations conversation.

7. Getting Started

To set up and run ShopAssist AI, follow these steps:

7.1 Prerequisites

- **Python 3.7+**
- **OpenAI API Key:** Add your OpenAI API key to the config.yaml file.

7.2 Installation and Setup

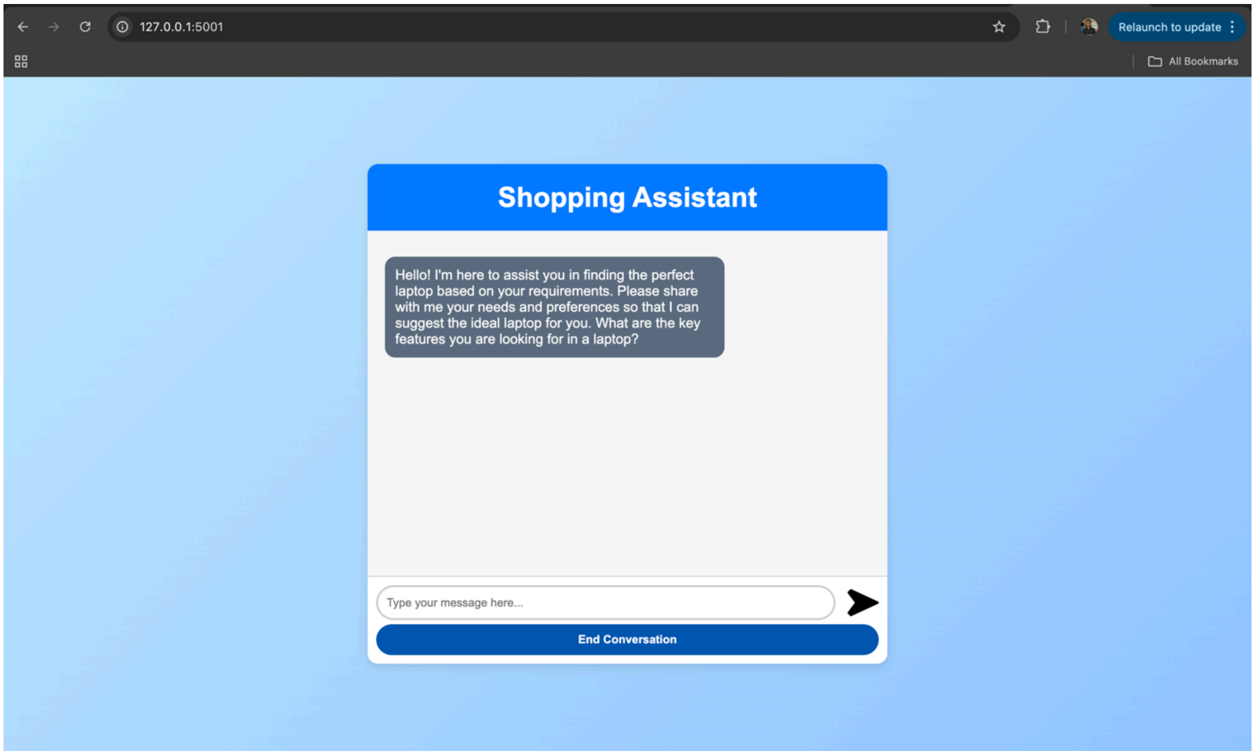
1. **Clone the repository:**
2. git clone git https://github.com/anshul1311/ShopAssist-AI.git
3. cd Shop-Assist-2.0
4. **Launch VS Code from Anaconda:**
   - o Open the ShopAssistAI folder in VS Code.
   - o Open a terminal in VS Code (Ctrl+`` or go to Terminal>New Terminal`).
5. **Install dependencies:**
6. pip install -r requirements.txt
7. **Run the application:**
8. python app.py

7.3 Note

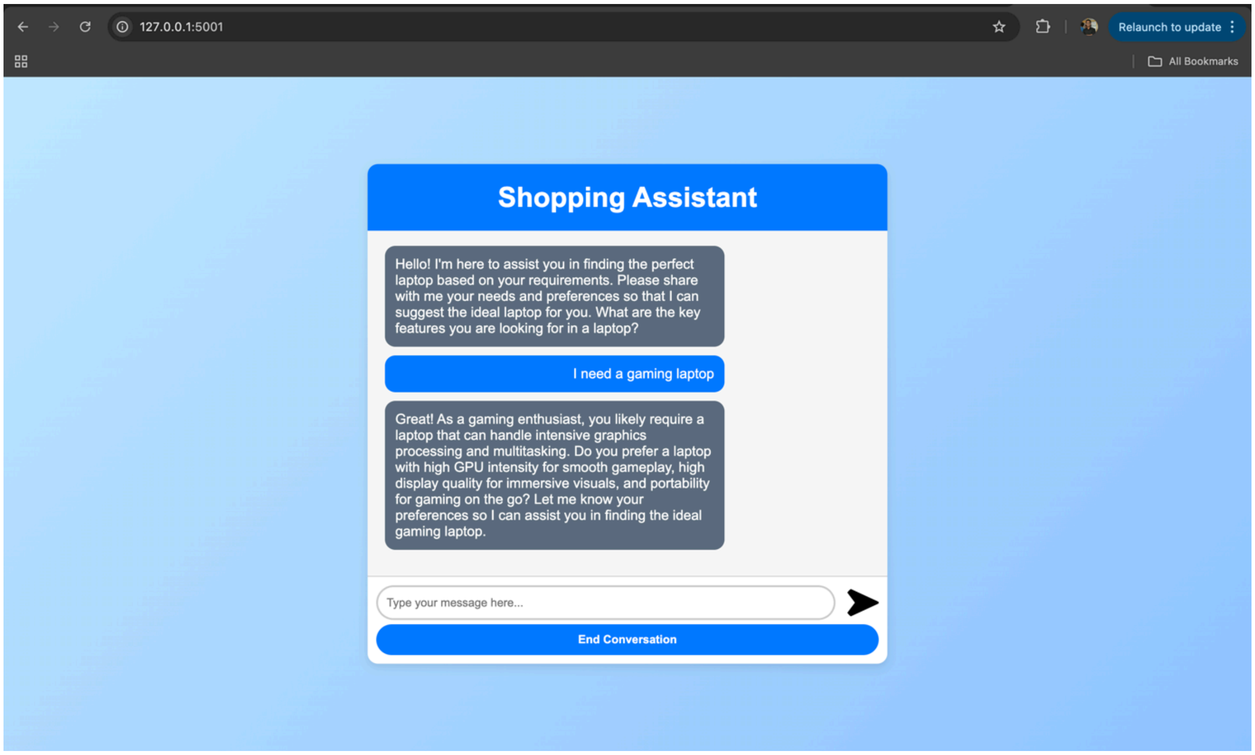
- This version includes steps to create and activate a Conda environment with **Python 3.11.9 or above** to ensure compatibility and correct setup before running the application.

8. Appendix

8.1 User Output Example Screenshots



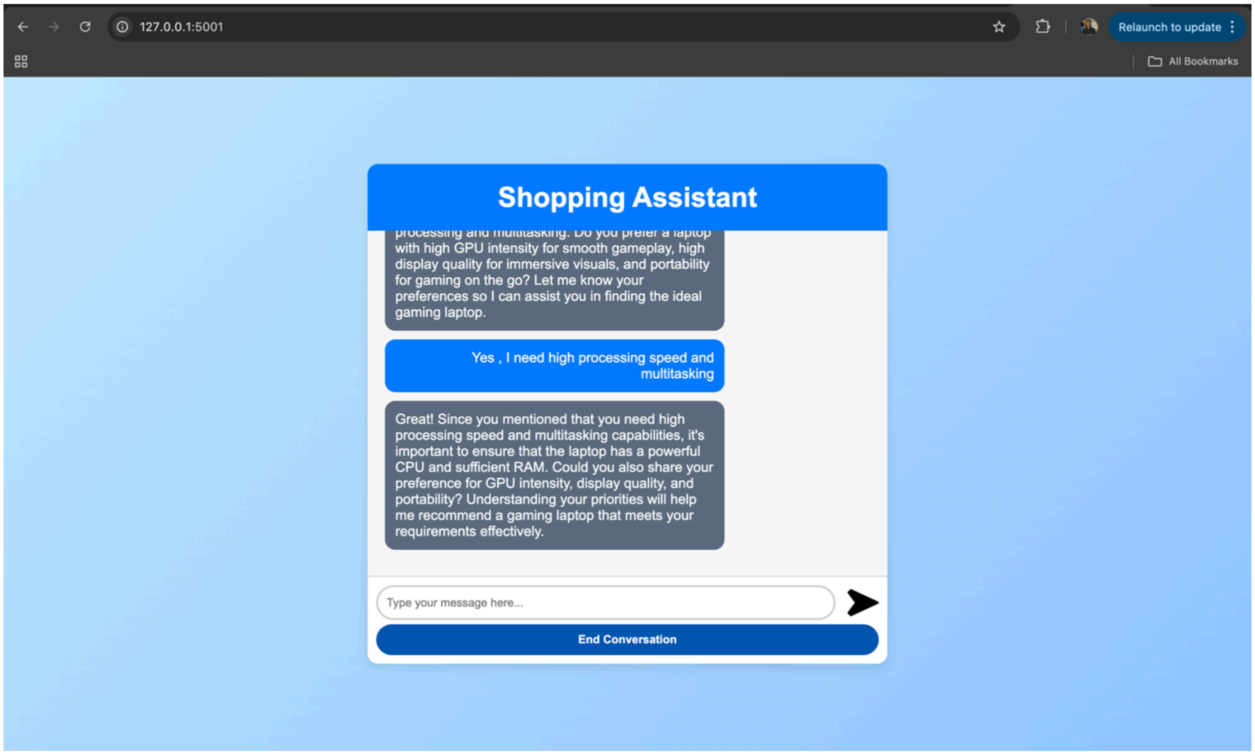
Hello! I'm here to assist you in finding the perfect laptop based on your requirements. Please share with me your needs and preferences so that I can suggest the ideal laptop for you. What are the key features you are looking for in a laptop?



Hello! I'm here to assist you in finding the perfect laptop based on your requirements. Please share with me your needs and preferences so that I can suggest the ideal laptop for you. What are the key features you are looking for in a laptop?

I need a gaming laptop

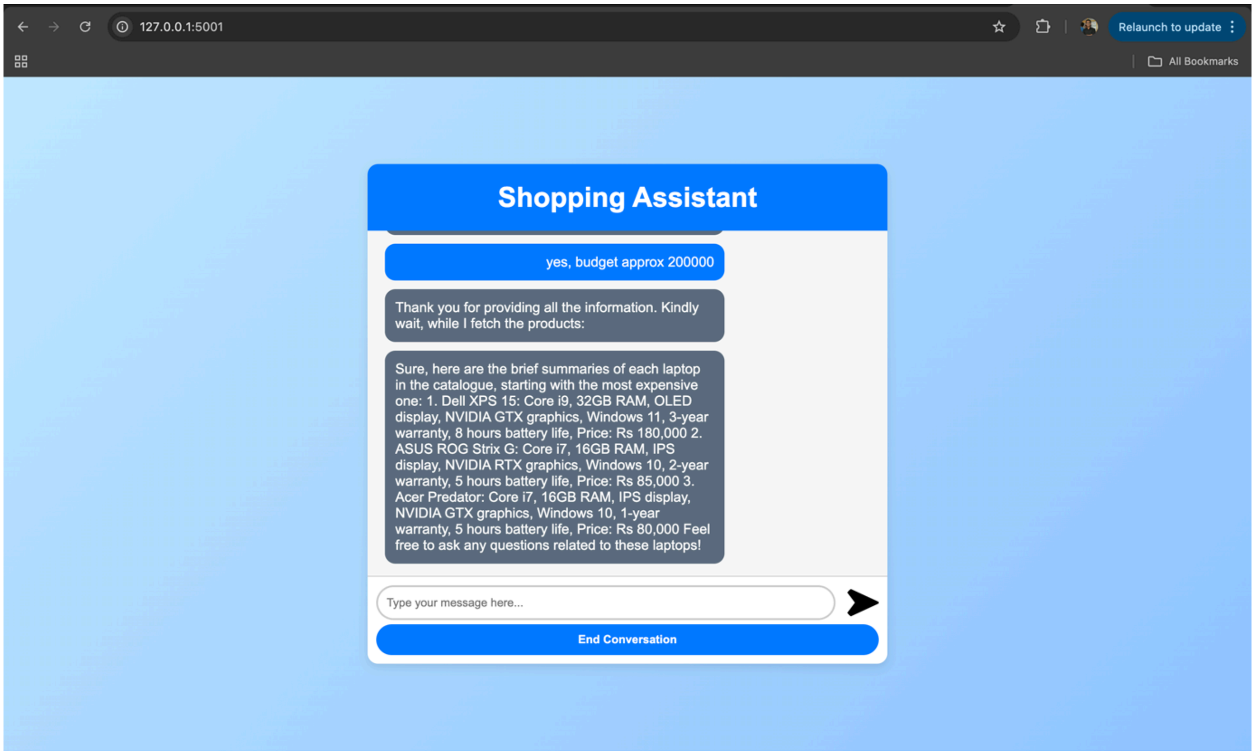
Great! As a gaming enthusiast, you likely require a laptop that can handle intensive graphics processing and multitasking. Do you prefer a laptop with high GPU intensity for smooth gameplay, high display quality for immersive visuals, and portability for gaming on the go? Let me know your preferences so I can assist you in finding the ideal gaming laptop.



processing and multitasking. Do you prefer a laptop with high GPU intensity for smooth gameplay, high display quality for immersive visuals, and portability for gaming on the go? Let me know your preferences so I can assist you in finding the ideal gaming laptop.

Yes , I need high processing speed and multitasking

Great! Since you mentioned that you need high processing speed and multitasking capabilities, it's important to ensure that the laptop has a powerful CPU and sufficient RAM. Could you also share your preference for GPU intensity, display quality, and portability? Understanding your priorities will help me recommend a gaming laptop that meets your requirements effectively.



yes, budget approx 200000

Thank you for providing all the information. Kindly wait, while I fetch the products:

Sure, here are the brief summaries of each laptop in the catalogue, starting with the most expensive one: 1. Dell XPS 15: Core i9, 32GB RAM, OLED display, NVIDIA GTX graphics, Windows 11, 3-year warranty, 8 hours battery life, Price: Rs 180,000 2. ASUS ROG Strix G: Core i7, 16GB RAM, IPS display, NVIDIA RTX graphics, Windows 10, 2-year warranty, 5 hours battery life, Price: Rs 85,000 3. Acer Predator: Core i7, 16GB RAM, IPS display, NVIDIA GTX graphics, Windows 10, 1-year warranty, 5 hours battery life, Price: Rs 80,000 Feel free to ask any questions related to these laptops!

ShopAssist AI represents a significant step forward in enhancing the online shopping experience. By combining the power of conversational AI with structured data analysis, it provides users with a seamless, personalized, and efficient way to find the products they need.

Chatbot Functionalities , Limitations & Challenges

1. Limited Number of Functions: There is a limit to the number of functions that can be defined within a model's context.
2. Chatbot can cater to irrelevant requests from the users apart from shop assist.It can cater to offensive/prohibited content generated from both user and assistant through moderation checks
3. Data has a limitation, hence the chatbot asks the question for origin giving only these laptops as options.
4. One limitation of the chatbot is that it looks for the exact value entered by the user in the database and returns if there are matching rows in the data. It is unable to recommend any alternatives to keep the user engaged.
5. Rate Limits

- APIs may restrict the number of requests per minute/hour.

- Implement caching for frequent queries.

6.Chatbot can handle a scenario wherein the user provides the requirements that do not match with the dataset and replies appropriately.
8.One particular challenge faced was when I tried to increase the dataset size, the OpenAI timeout error was quite frequent and hence had to use only limited rows.
9. Sometimes the bot still times out when trying to fetch the results, therefore please try again should you face any such errors.
10. Authentication & Permissions :

- API keys or OAuth tokens may be required.

- Some data (like user purchase history) might be restricted due to privacy concerns.

11. Product Search & Recommendation :

- Suggest products based on user preferences.

- Filter products by category, price, or brand.

- Compare product features.

12. Latency & Downtime:

- API response times can vary.
- Have fallback responses if an API is unavailable.

13. There are misclassifications in the laptop’s specifications, even after specifying clear rules for LLM. You can fine-tune an open-source LLM to make its understanding & performance better.