# Project Report: Shop Assist- Laptop Recommender

1. **Project Background:**

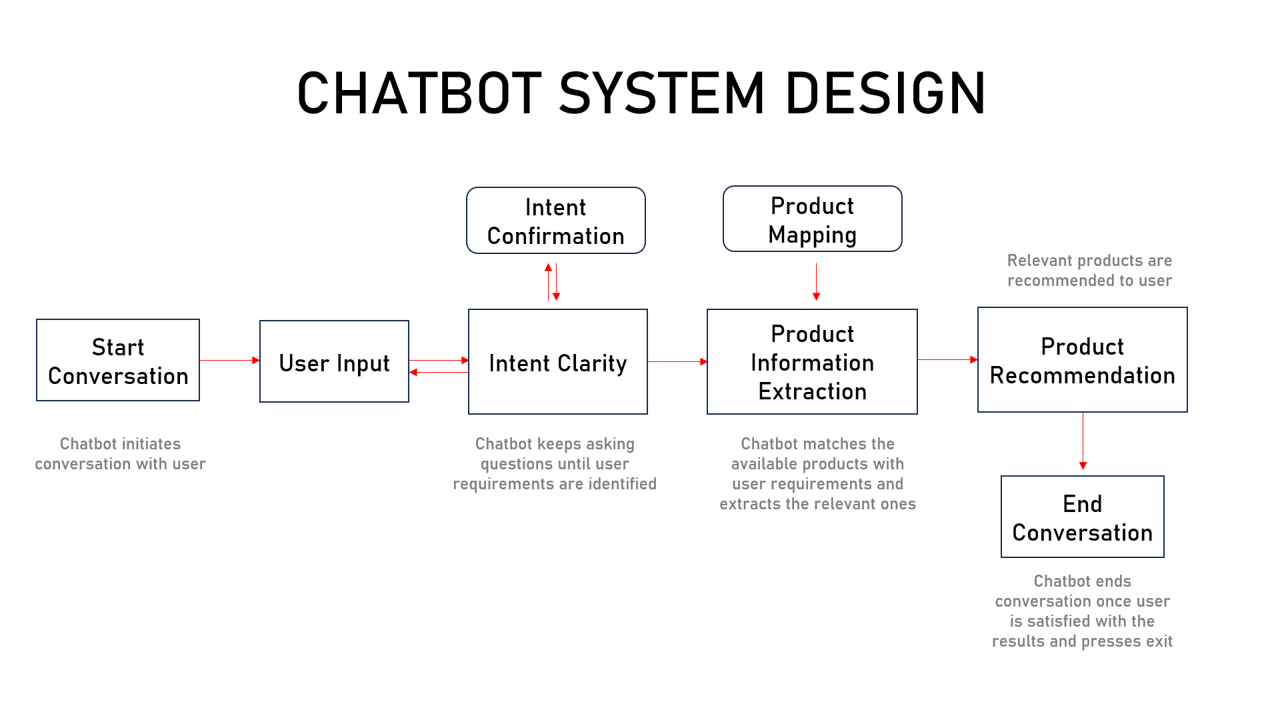
In today's digital age, online shopping has become the go-to option for many consumers. However, the overwhelming number of choices and the lack of personalized assistance can make the shopping experience daunting. To address this, we have developed ShopAssist AI, a chatbot that combines the power of large language models and rule-based functions to ensure accurate and reliable information delivery.

1. **Objective:**

 LLMs inherently struggle to generate standardised output consistently. While constructing the intent clarity layer, which required multiple layers to ensure that the output adhered to standard formats before invoking functions for database information extraction. OpenAI, the creator behind ChatGPT, has introduced an exciting innovation called 'function calling' within their GPT models. In this project we have opportunities to enhance the customer experience by crafting a superior and more intelligent chatbot

1. **Implementation:**

The project is implemented using Python programming language. The core functionality is powered by the OpenAI GPT-3.5 model, which is utilized for generating laptop recommendations based on user input. This diagram highlights the key components of the architecture:



1. **Enhancements**

Enhancements have been made in the Intent Clarity layer. Once Intent confirmation is completed. We use Function Calling to structure the user input in an API readable format. This reduces complex code to parse user input. It also enables users to update their requirements in any structure as a part of the conversations with the chat bot. following new functions are introduced:

1. **get\_chat\_completions\_func\_calling:** This method invokes function calling to structure the user input instead to using dictionary\_present() where the structure of the user input is explicitly formed.
2. **shopassist\_custom\_functions:** this method describes the function that will be used by Open AI function calling. It specifies the name of the function, what it does and input parameters required.
3. **extract\_user\_info:** captures key laptop preferences such as GPU intensity, display quality, portability, multitasking capability, processing speed, and budget from the user’s input. By clearly defining these parameters, the function can tailor recommendations based on individual needs.

This approach helps by:

**Structuring User Input**: It breaks down the user's request into key categories (e.g., GPU intensity or budget), ensuring precise extraction of relevant details.

**Automation and Personalization**: The API can then automatically provide personalized suggestions, streamlining the decision-making process without needing extensive manual input.

**Enhanced User Experience**: It transforms vague user queries into actionable insights, improving the accuracy and relevance of the laptop recommendations.

1. **dialogue\_mgmt\_system\_func\_call:** The main function facilitates user interaction by prompting users to input their preferences. The system checks for flagged content using the moderation function and generates recommendations accordingly. The change here is that it uses the **get\_chat\_completions\_func\_calling** which in turn uses the function calling API
2. **Conclusion:**

The Shop Assist 2.0 project demonstrates the capabilities of natural language processing and AI in providing personalized recommendations in the restaurant domain. By leveraging advanced language models like GPT-3.5, the system is able to understand and respond to user queries effectively. With further refinement utilising Function calling, we are able to structure user inputs for the API invocation thereby making user interactions more conversational.