ShopAssist AI Project

This document is a supplementary reading material for the ShopAssist AI project. This document explains the various functions and stages of the project.

The key learning objectives of this project are:

- Understanding the multi-stage architecture behind the fully functioning chatbot system,
- Integrate OpenAI’s **Chat Completion APIs** to provide a natural and contextually relevant conversation,
- Build an end-to-end chatbot solution that seamlessly interacts with users, understands their needs and delivers accurate and relevant recommendations.

You are required to be familiar with the following concepts to fully understand/ appreciate this session.

- **API Calls and GPT-3.5 Model:** You learnt how to make basic API calls using Python to interact with the GPT-3.5 model of OpenAI. The model has been extensively used in this course to create the AI-based Shop Assist system.
- **Prompt Engineering:** You learnt the principles of prompt engineering, including providing clear instructions and enhancing the reasoning capabilities of the artificial intelligence (AI) model. A five-component framework (task, role, context, guidelines and output format) is utilised to create effective prompts for better results.
- **Enhancing the Reasoning Capabilities of the LLM:** You explored two techniques for enhancing the reasoning capabilities of the AI model – chain of thought prompting and few-shot prompting. These techniques will be demonstrated in the Shop Assist AI system.
- **Designing End-to-End LLM Systems:** You covered the design of end-to-end LLM systems comprising multiple components. Shop

Assist AI serves as an example of such a system, incorporating conversation and information gathering, data processing and product recommendation.

In the ShopAssist project, your task is to build ShopAssist AI, a laptop recommendation chatbot that can:

- Interact with users interactively,
- Understand the user's laptop requirements, and,
- Recommend the most suitable laptops based on their needs and preferences.

Project Background:

In today's digital age, online shopping has become the preferred option for many consumers. However, the vast array of choices and the lack of personalised assistance can make the shopping experience overwhelming and challenging. To address this issue, we present ShopAssist AI. This chatbot combines the power of LLMs and rule-based functions to provide accurate and reliable recommendations during the online laptop shopping experience.

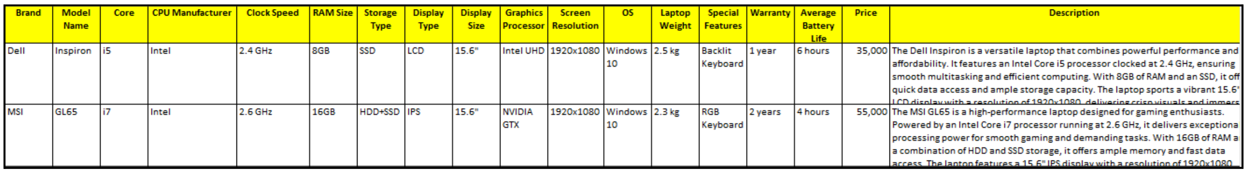
Problem Statement:

Given a dataset containing laptop information (product names, specifications, descriptions, etc.), build a chatbot that parses the dataset and provides accurate laptop recommendations based on user requirements. This chatbot, named ShopAssist AI, will

- Interact with users,
- Understand their laptop requirements and
- Recommend the most suitable laptops from a dataset based on their needs and preferences.

You can download the files used in this project from the following link. The stub notebooks for the live sessions can be accessed from the following link - Starter Notebooks

Dataset:
Let’s take a look at a quick overview of the laptop dataset used in this project.



Brand	Model Name	Core	CPU Manufacturer	Clock Speed	RAM Size	Storage Type	Display Type	Display Size	Graphics Processor	Screen Resolution	OS	Laptop Weight	Special Features	Warranty	Average Battery Life	Price	Description
Dell	Inspiron	i5	Intel	2.4 GHz	8GB	SSD	LCD	15.6"	Intel UHD	1920x1080	Windows 10	2.5 kg	Backlit Keyboard	1 year	6 hours	35,000	The Dell Inspiron is a versatile laptop that combines powerful performance and affordability. It features an Intel Core i5 processor clocked at 2.4 GHz, ensuring smooth multitasking and efficient computing. With 8GB of RAM and an SSD, it off quick data access and ample storage capacity. The laptop sports a vibrant 15.6" LCD display with a resolution of 1920x1080, delivering crisp visuals and immers
MSI	GL65	i7	Intel	2.6 GHz	16GB	HDD+SSD	IPS	15.6"	NVIDIA GTX	1920x1080	Windows 10	2.3 kg	RGB Keyboard	2 years	4 hours	55,000	The MSI GL65 is a high-performance laptop designed for gaming enthusiasts. Powered by an Intel Core i7 processor running at 2.6 GHz, it delivers exceptiona processing power for smooth gaming and demanding tasks. With 16GB of RAM a a combination of HDD and SSD storage, it offers ample memory and fast data access. The laptop features a 15.6" IPS display with a resolution of 1920x1080

Primarily, your chatbot will analyse the ‘Description’ column for each laptop, understand whether the user’s requirements match the laptop's specifications and then forward a relevant laptop as a recommendation.

Let’s take a look at a sample laptop description.

“The Dell Inspiron is a versatile laptop that combines powerful performance and affordability. It features an Intel Core i5 processor clocked at 2.4 GHz, ensuring smooth multitasking and efficient computing. With 8 GB of RAM and an SSD, it offers quick data access and ample storage capacity. The laptop sports a vibrant 15.6” LCD display with a resolution of 1920 × 1080, delivering crisp visuals and an immersive viewing experience. Weighing just 2.5 kg, it is highly portable, making it ideal for on-the-go usage. Additionally, it boasts an Intel UHD GPU for decent graphical performance and a backlit keyboard for enhanced typing convenience. With a one-year warranty and a battery life of up to 6 hours, the Dell Inspiron is a reliable companion for work or entertainment. All these features are packed at an affordable price of 35,000, making it an excellent choice for budget-conscious users.”

The overall approach you will follow throughout this chatbot creation journey is as follows:

- **Conversation and Information Gathering:** The chatbot will utilise language models to understand and generate natural responses. Through a conversational flow, it will ask relevant questions to gather information about the user's requirements.
- **Information Extraction:** Once the essential information is collected, we will utilise LLM's natural language understanding and a few rule-based reasoning to extract the top three laptops that best match the user's needs.
- **Personalised Recommendation:** Leveraging the extracted information, the chatbot engages in further dialogue with the user, efficiently addressing their queries and aiding them in finding the perfect laptop solution.

Why?

The important question you need to ask is the necessity of creating such a complex and extensive system design. It should be noted that multiple design considerations have been addressed while designing the chatbot system. These include:

- **LLM Hallucinations**: Reduce or remove hallucinations from LLM responses. For more information about hallucinations in LLMs, you can refer to this link.
- **LLM consistency and reproducibility**: Ensure the output responses from the language model (OpenAI) are reproducible, consistent for the same set of input parameters and correct. For example, OpenAI's chat completion API has provided functionality such as JSON output parsing and seed argument. You can refer to the following links for more information:
    - How to make your completions outputs consistent with the new seed parameter | OpenAI Cookbook
    - OpenAI API Guide: Using JSON Mode
    - OpenAI Documentation - Text generation -JSON Mode
- Utilise the **LLM reasoning capabilities** for certain high-level tasks. This includes advanced prompt engineering

techniques you've covered previously such as Few-Shot prompting, Chain-of-Thought prompting, Chain-of-Thought Few-Shot prompting, Self-consistency prompting etc.

- **Account for the limitations of LLMs**: A key factor in building LLM-based applications is understanding the capabilities of LLMs. Throughout this project, you will encounter various limitations of using simple prompts for LLMs and dive deeper into various prompt engineering concepts and prompt tuning to get the output response you need for the given input and ensure that each layer functions as part of a cohesive system and the overall system works optimally.
  A design choice that has been used in the ShopAssist project is the addition of a heuristic/ rule-based logic for assigning scores to laptops based on the user's requirement.

System Design Overview - ShopAssist AI

The system design for the project is illustrated in the diagram below.

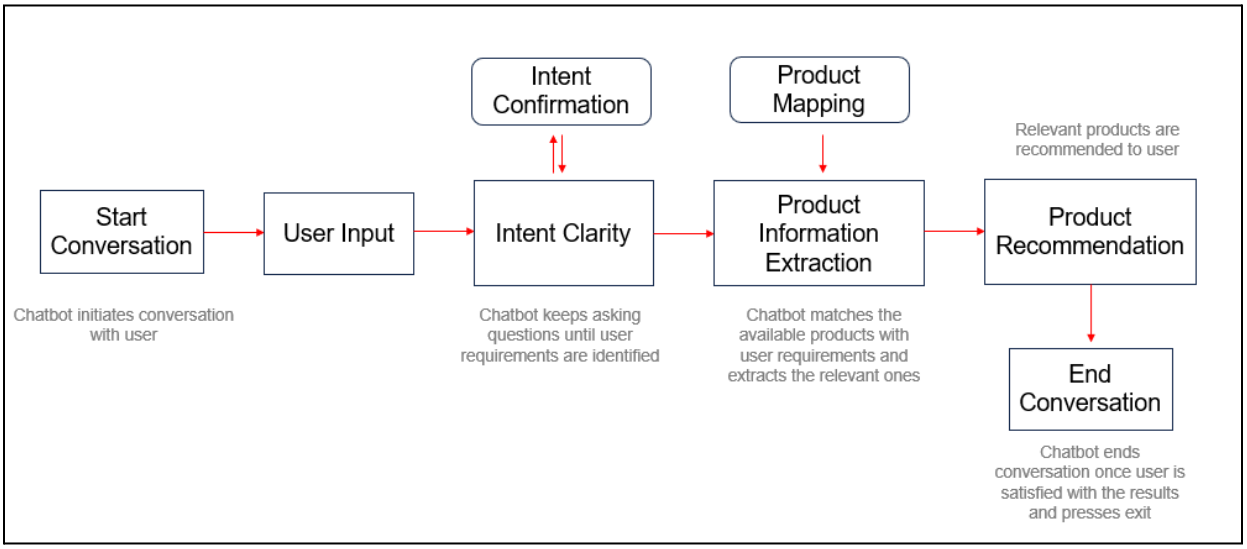

Intent Confirmation
Product Mapping
Relevant products are recommended to user
Start Conversation
User Input
Intent Clarity
Product Information Extraction
Product Recommendation
Chatbot initiates conversation with user
Chatbot keeps asking questions until user requirements are identified
Chatbot matches 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

Illustration: Stages of ShopAssist AI System

For ease, the system design has been compartmentalised into three stages as shown in the diagram below.

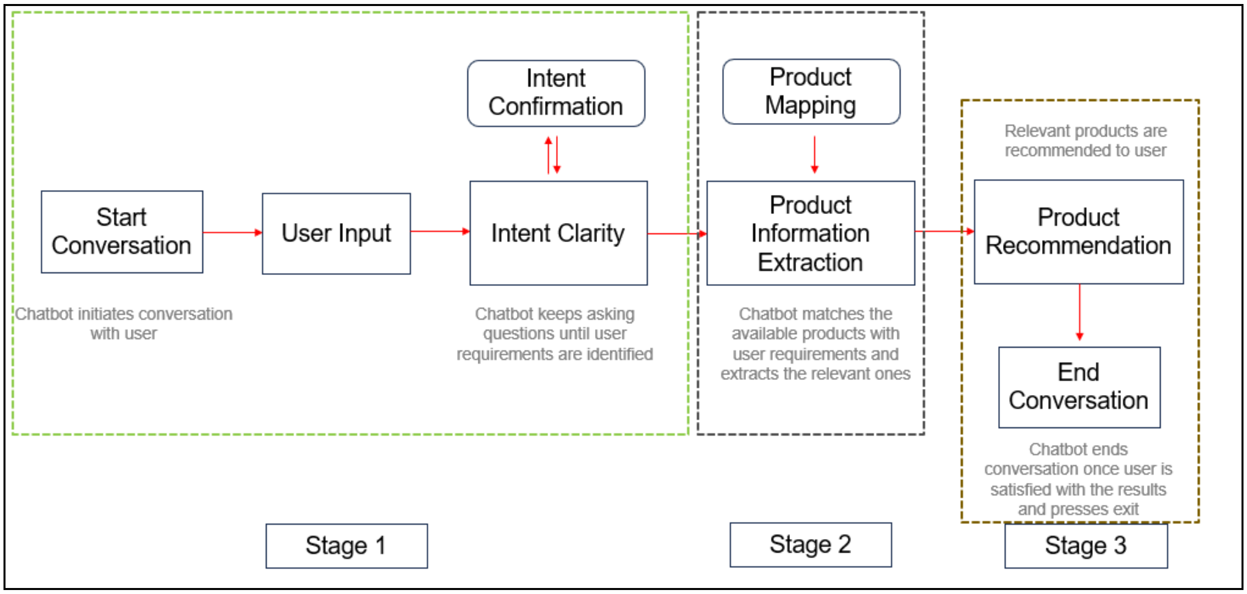
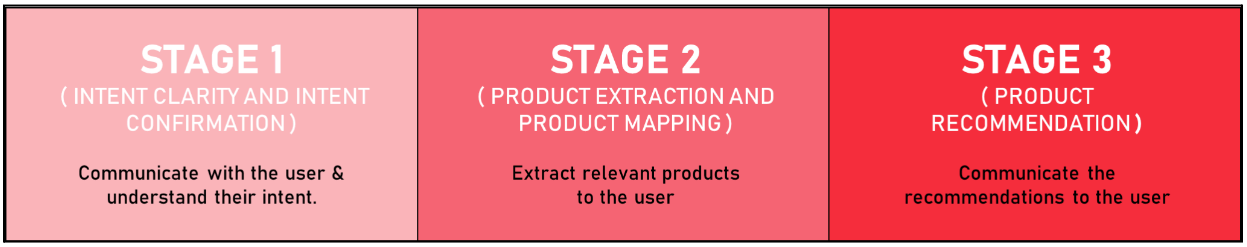

Start Conversation
Chatbot initiates conversation with user
User Input
Intent Confirmation
Intent Clarity
Chatbot keeps asking questions until user requirements are identified
Product Mapping
Product Information Extraction
Chatbot matches the available products with user requirements and extracts the relevant ones
Relevant products are recommended to user
Product Recommendation
End Conversation
Chatbot ends conversation once user is satisfied with the results and presses exit
Stage 1
Stage 2
Stage 3

Illustration: Stages of ShopAssist AI System

As you can see in this image, there are three stages of the chatbot, which are as follows:

- **Stage 1:** Intent Clarity and Intent Confirmation
- **Stage 2:** Product Extraction and Product Mapping
- **Stage 3**: Product Recommendation



STAGE 1 (INTENT CLARITY AND INTENT CONFIRMATION)	STAGE 2 (PRODUCT EXTRACTION AND PRODUCT MAPPING)	STAGE 3 (PRODUCT RECOMMENDATION)
Communicate with the user & understand their intent.	Extract relevant products to the user	Communicate the recommendations to the user

Illustration: Stages of ShopAssist AI System

Stage 1 - Intent Clarity and Intent Confirmation

The first stage involves a conversation between the user and the AI system. Python functions such as **initialize_conversation()**, trigger the conversation, and those such as **get_chat_completions()** allows the conversation to continue with each conversation via LLM calls.

The stage includes an additional layer called **moderation_check()** to flag and discontinue conversations that contain unsafe or sensitive content.

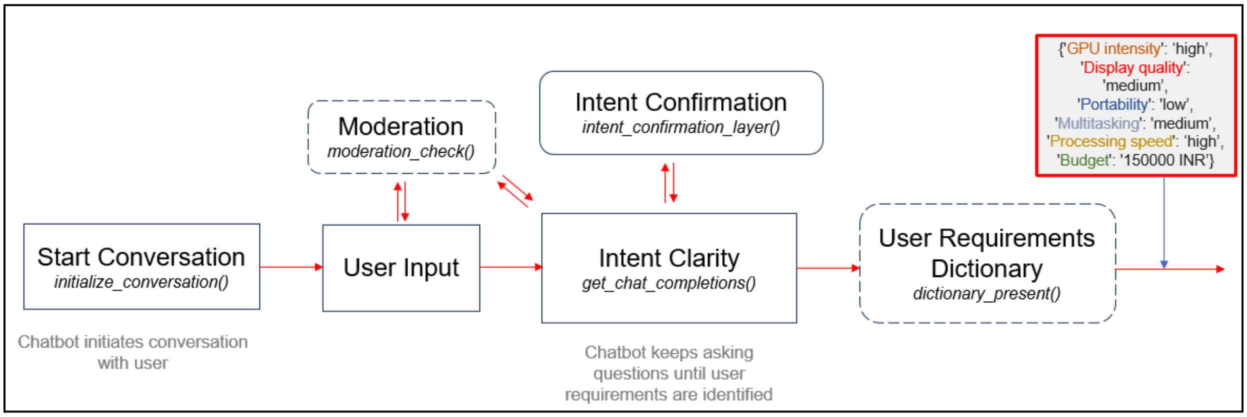

Start Conversation
initialize_conversation()
Chatbot initiates conversation with user
User Input
Moderation
moderation_check()
Intent Confirmation
intent_confirmation_layer()
Intent Clarity
get_chat_completions()
Chatbot keeps asking questions until user requirements are identified
User Requirements Dictionary
dictionary_present()
{'GPU intensity': 'high', 'Display quality': 'medium', 'Portability': 'low', 'Multitasking': 'medium', 'Processing speed': 'high', 'Budget': '150000 INR'}

Illustration: Stage 1 - Intent Clarity and Intent Confirmation

The output of this stage is a dictionary that has captured all the needs of the user (User Requirements Dictionary) in key-value pairs or specifically JSON. The reason for JSON output is for output parsing convenience - the AI system aims to collect user requirements and store them in a Python dictionary for further processing rather than a string. It should be noted that LLMs are token producers and output tokens are primarily in string format rather than Python objects, unless specifically mentioned.

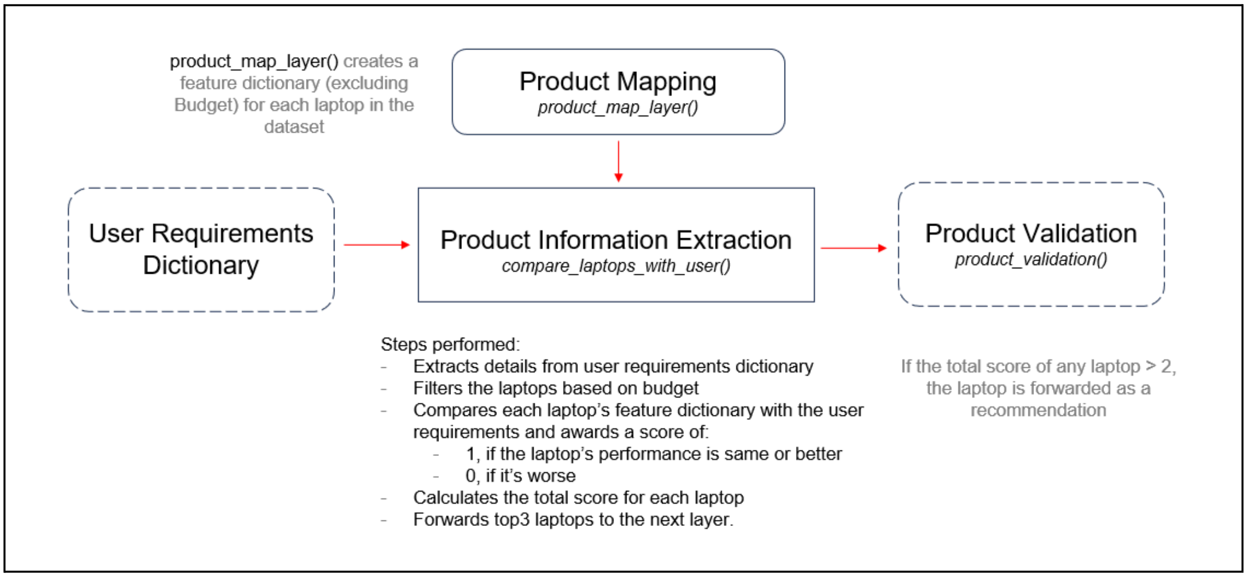
Therefore, to guard the scenario where we do not have a dictionary output, a Python function called 'dictionary_present()' is created. It is used to convert the LLM output, which might resemble a dictionary in string format, into an actual Python dictionary.

The overall functioning of the chatbot in this stage happens in the following steps:

- **Initialising conversation and chat model completion:** The conversation starts with the ‘**initialize_conversation()**’ function, where the AI system introduces itself and asks for the user’s requirements. The subsequent messages from the user and the AI system are attributed to ‘**chat_model_completion()**’ function where the OpenAI’s chat completion API is utilised for continuing the user conversation, until the requirements have been completely identified.
- **Intent confirmation:** The AI system uses an ‘**Intent Confirmation layer’ (intent_confirmation_layer())** as a flag (yes or no) to indicate whether all requirements of the user have been captured. During the conversation, if the system receives ‘no,’ it understands that further questions are needed to capture all the requirements.
- **User requirements dictionary:** Once the ‘**Intent Confirmation**’ **layer** confirms that all requirements of the user have been captured (*i.e.* ‘yes’ flag), it passes the requirements to the ‘User Requirements Dictionary’ (**dictionary_present() function**) to be converted into a dictionary/ dictionary-like object to the **user_req** variable.

Stage 2 - Product Mapping and Information Extraction

The second stage of the system is the ‘Product Mapping and Information Extraction’ stage. This stage filters the laptops as per the user requirements dictionary captured in the previous stage and uses it to present the top three laptop recommendations to the user. The illustrations below shows functions and steps in Stage 2.


product_map_layer() creates a feature dictionary (excluding Budget) for each laptop in the dataset
Product Mapping
product_map_layer()
User Requirements Dictionary
Product Information Extraction
compare_laptops_with_user()
Product Validation
product_validation()
If the total score of any laptop > 2, the laptop is forwarded as a recommendation

Steps performed:

- Extracts details from user requirements dictionary
- Filters the laptops based on budget
- Compares each laptop's feature dictionary with the user requirements and awards a score of:
  - 1, if the laptop's performance is same or better
  - 0, if it's worse
- Calculates the total score for each laptop
- Forwards top3 laptops to the next layer.

Illustration: Stage 2 Product Map and Information Extraction Layer

The entire laptop filtering process in Stage 2 can divided into two parts:

Part 1 (Product Mapping): You create a feature dictionary for each laptop from its given description. This is done using the **product_map_layer()** function, which extracts key features and criteria from laptop descriptions.

- This function extracts the primary laptop features from the detailed description of any laptop (such as GPU intensity, display quality, portability, multitasking, processing speed, etc.). These entries are stored as key-value pairs of a dictionary **lap_spec**.
- Once these values are extracted, they are mapped with the appropriate classification value **(low, medium or high)** defined inside rules.
- To perform this mapping, you need a detailed set of rules from your heuristics, which the LLM model is not privy to.
- Since this operation is independent from any user input, you need to execute this function once for all the laptops.

Part 2 (Information Extraction): So, till this stage, you have two dictionaries: **lap_spec** and **user_req**. All you need to do now is to determine how similar a laptop's features (stored in lap_spec) are to the user's requirements (stored in **user_req**). But before you compare these two dictionaries, they need to be converted from a string of dictionary to just a dictionary. This is done using the **dictionary_present()** function.

Then, they are passed to a rule-based function **compare_laptops_with_user()**.

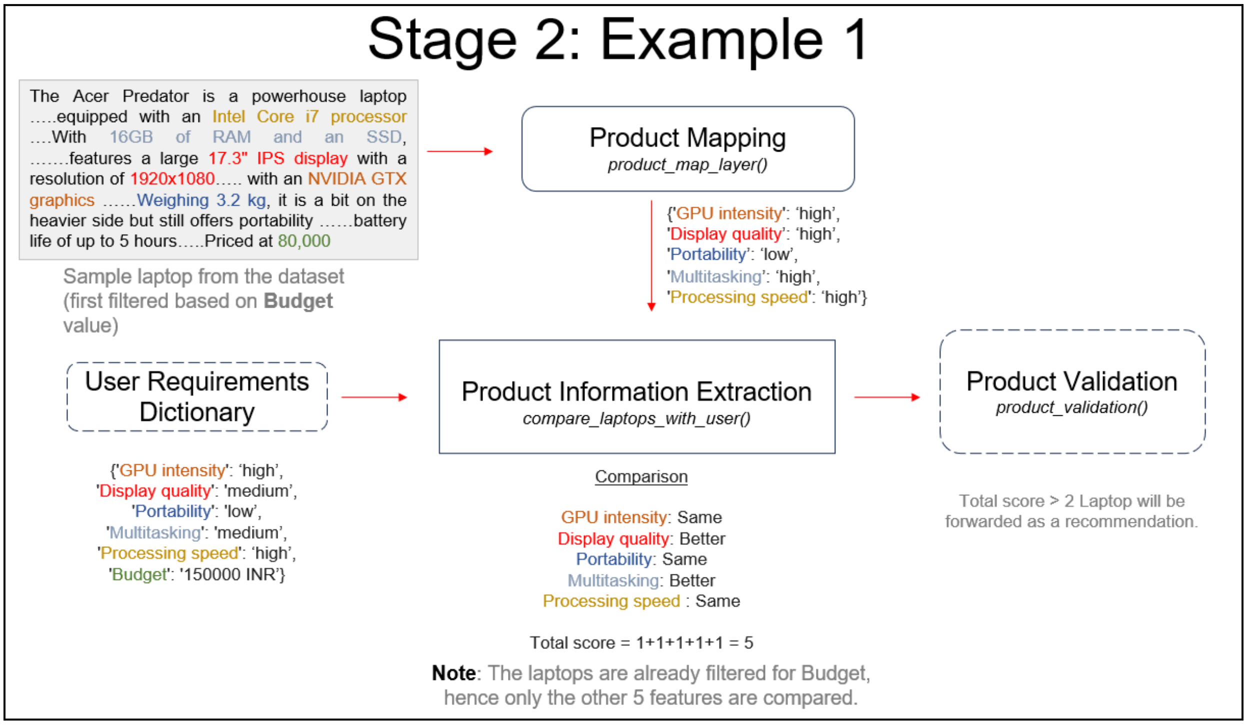
- For each feature, a score of 1 is assigned if the feature is the same or better than the user's requirement. Otherwise, a score of 0 is assigned.
- The scoring is performed for all the laptops iteratively. Once the scoring is done, the scores are then used to rank and identify the top three laptops as recommendations for the user.
- This scoring is not fixed, as it can be modified as per your requirement for any particular use case.

Product Validation Layer: Once the top three laptops are extracted, the list is sent to the product validation layer, which ensures that only relevant products are forwarded as recommendations.

The purpose of the product validation layer is to ensure that only laptops with a **score of three or above** are recommended to the user. The score threshold of three is arbitrary but is chosen based on the desire for at least three features to meet or exceed the user's requirements.

So, if a laptop's total score is greater than two (indicating that it meets or exceeds user's requirements in at least two features), it is considered a recommendation

The images below shows the entire product mapping and product information extraction for two sets of **lap_spec** variables.



Stage 2: Example 1

The Acer Predator is a powerhouse laptop …..equipped with an Intel Core i7 processor ….With 16GB of RAM and an SSD, …….features a large 17.3" IPS display with a resolution of 1920x1080….. with an NVIDIA GTX graphics ……Weighing 3.2 kg, it is a bit on the heavier side but still offers portability ……battery life of up to 5 hours…..Priced at 80,000

Sample laptop from the dataset (first filtered based on **Budget** value)

Product Mapping
product_map_layer()

{'GPU intensity': 'high',
'Display quality': 'high',
'Portability': 'low',
'Multitasking': 'high',
'Processing speed': 'high'}

User Requirements Dictionary

{'GPU intensity': 'high',
'Display quality': 'medium',
'Portability': 'low',
'Multitasking': 'medium',
'Processing speed': 'high',
'Budget': '150000 INR'}

Product Information Extraction
compare_laptops_with_user()

Comparison

GPU intensity: Same
Display quality: Better
Portability: Same
Multitasking: Better
Processing speed : Same

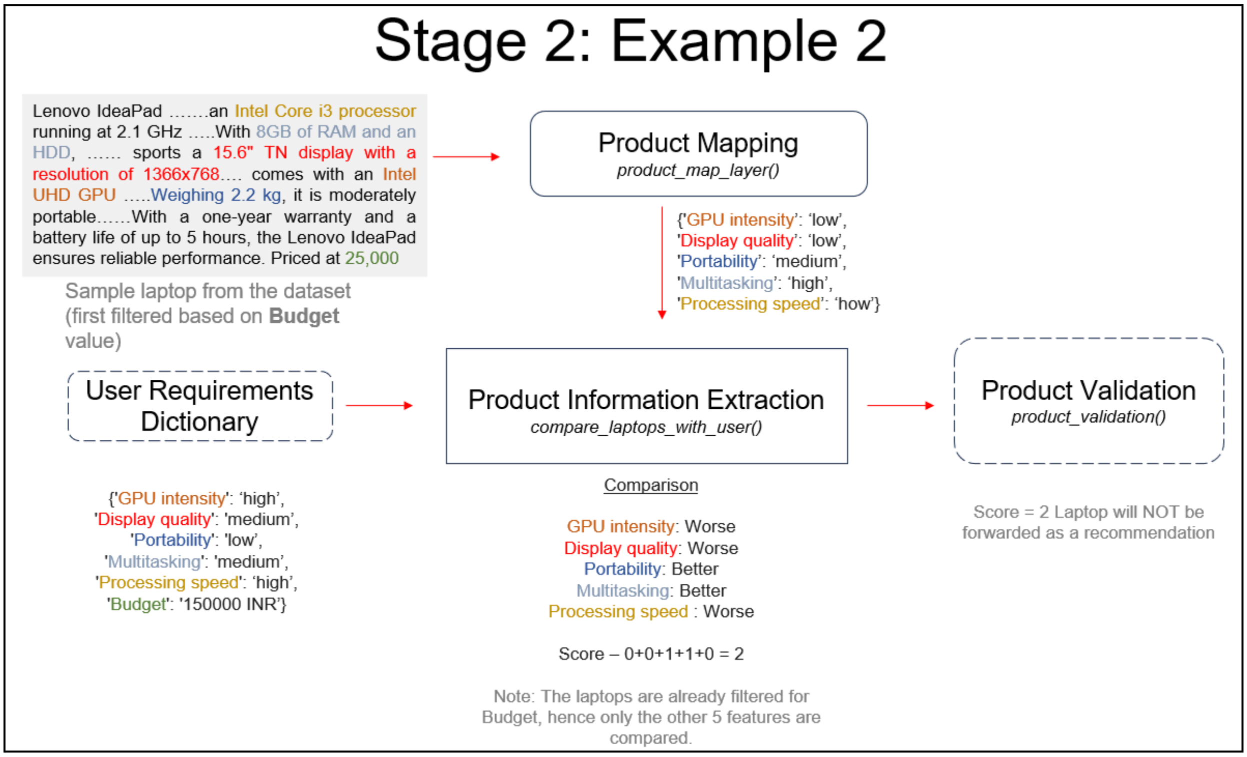
Total score = 1+1+1+1+1 = 5

Note: The laptops are already filtered for Budget, hence only the other 5 features are compared.

Product Validation
product_validation()

Total score > 2 Laptop will be forwarded as a recommendation.

Illustration: Laptop Example 1



Stage 2: Example 2

Lenovo IdeaPad …….an Intel Core i3 processor running at 2.1 GHz …..With 8GB of RAM and an HDD, …… sports a 15.6" TN display with a resolution of 1366x768…. comes with an Intel UHD GPU …..Weighing 2.2 kg, it is moderately portable……With a one-year warranty and a battery life of up to 5 hours, the Lenovo IdeaPad ensures reliable performance. Priced at 25,000

Sample laptop from the dataset (first filtered based on **Budget** value)

Product Mapping
product_map_layer()

{'GPU intensity': 'low',
'Display quality': 'low',
'Portability': 'medium',
'Multitasking': 'high',
'Processing speed': 'how'}

User Requirements Dictionary

{'GPU intensity': 'high',
'Display quality': 'medium',
'Portability': 'low',
'Multitasking': 'medium',
'Processing speed': 'high',
'Budget': '150000 INR'}

Product Information Extraction
compare_laptops_with_user()

Comparison

GPU intensity: Worse
Display quality: Worse
Portability: Better
Multitasking: Better
Processing speed : Worse

Score – 0+0+1+1+0 = 2

Note: The laptops are already filtered for Budget, hence only the other 5 features are compared.

Product Validation
product_validation()

Score = 2 Laptop will NOT be forwarded as a recommendation

Illustration: Laptop Example 2

Stage 3 - Product Recommendation

Finally, you have reached the product recommendation layer. It takes the output from the 'compare_laptops_with_user' function in the previous layer and provides the recommendations to the user. The broader process happening in this stage can be summarised as follows:

- The product validation layer will recommend a maximum of three laptops to the product recommendation layer. This ensures that the user is presented with a manageable number of laptop options to choose from.
- It is possible that no laptops meet the score threshold of three, in which case the product validation layer will feed 'None' or 'No laptops matched' to the product recommendation layer. In such cases, the AI system will be instructed to connect the user to a human expert.

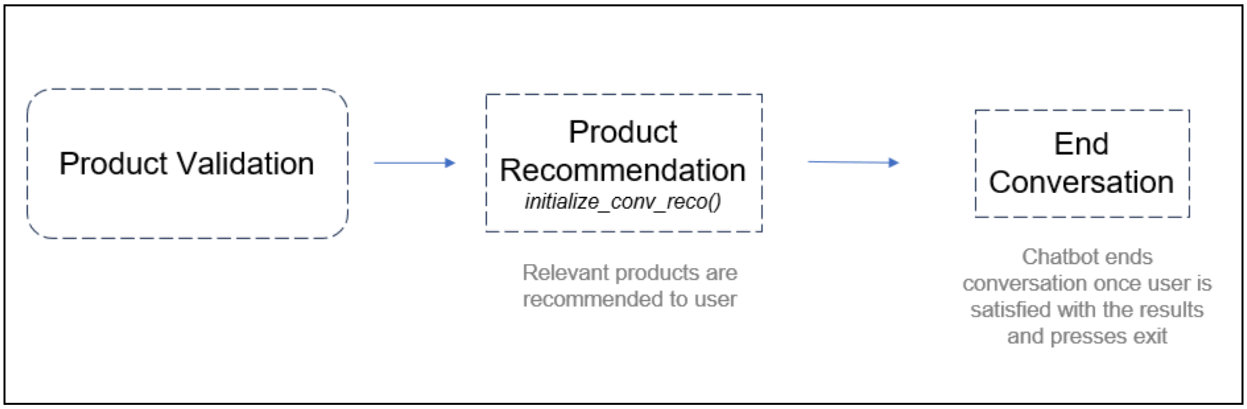

Product Validation
Product Recommendation
initialize_conv_reco()
Relevant products are recommended to user
End Conversation
Chatbot ends conversation once user is satisfied with the results and presses exit

Illustration: Stage 3 - Product Recommenation Stage

The conversation flow of ShopAssist AI is shown below:

Unset

```
Chatbot Assistant:Hello! I am an intelligent laptop gadget
expert and I am here to help you find the best laptop for
your needs. Please let me know what you are looking for in
a laptop and any specific requirements you may have.

User: I need a laptop for coding.

Intent Confirmation Yes/No: No

Chatbot Assistant: Great! As a coder, having a laptop with
good processing speed is crucial for running multiple
development tools and applications smoothly. Do you
typically work on large codebases or complex algorithms
that require intense processing power? Understanding the
intensity of your coding tasks will help me recommend the
most suitable laptop for you.

Intent Confirmation Yes/No: No

User: I work with image processing and deep learning
applications.
```

Intent Confirmation Yes/No: No

Chatbot Assistant: Thank you for providing that information. Working with image processing and deep learning applications requires a laptop with high processing speed and GPU intensity. These tasks can be computationally demanding, so a strong GPU is essential for efficient performance. Do you also require a high-resolution display for better visualization of images and data? Understanding your requirements will help me tailor my recommendations accordingly.

User: Yes. A high-resolution display would be ideal.

Intent Confirmation Yes/No: No

Chatbot Assistant: Great to know that a high-resolution display is important to you. Along with high GPU intensity and processing speed, having high display quality is crucial for working with image processing and deep learning applications. To better understand your needs, do you prefer a laptop that is highly portable and easy to carry around, or do you primarily work from a fixed location? Understanding your preference for portability will help me recommend the most suitable laptop for you.

User: Portability is not a priority but ideally the laptop should weigh around 2 kgs.

Intent Confirmation Yes/No: No

Chatbot Assistant: Got it, portability is not a top priority but you prefer a laptop that weighs around 2 kgs. This indicates that you need a balance between portability and performance. Given your focus on image processing and deep learning, high GPU intensity, processing speed, and

```
display quality are still crucial factors. Lastly, could
you please specify your budget for the laptop? This
information will help me narrow down the options within
your price range while meeting your specified requirements.

User: around 150000

Intent Confirmation Yes/No: Yes

{'GPU intensity': 'high', 'Display quality': 'high',
'Portability': 'medium', 'Multitasking': 'medium',
'Processing speed': 'high', 'Budget': 150000}

Variables extracted!

Chatbot Assistant:

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

1. Acer Predator: Core i7 processor, 16GB RAM, SSD storage,
NVIDIA GTX graphics, 17.3" IPS display (1920x1080), Windows
10, 3.2 kg, Dual Cooling Fans, 5 hours battery life, 1-year
warranty, Price: Rs. 80,000

2. MSI GL65: Core i7 processor, 16GB RAM, HDD+SSD storage,
NVIDIA GTX graphics, 15.6" IPS display (1920x1080), Windows
10, 2.3 kg, RGB Keyboard, 4 hours battery life, 2-year
warranty, Price: Rs. 55,000

3. Lenovo ThinkPad: Ryzen 7 processor, 16GB RAM, SSD
storage, NVIDIA GTX graphics, 14" IPS display (2560x1440),
Linux, 1.6 kg, Backlit Keyboard, 6 hours battery life,
3-year warranty, Price: Rs. 60,000

Based on your user profile and the laptops available, the
Acer Predator seems to align well with your requirements.
```

It offers high GPU intensity and processing speed with a high-quality display. The laptop also provides decent portability and multitasking capabilities within your budget.

User: what GPU does it use?

Chatbot Assistant: The Acer Predator laptop uses an NVIDIA GTX graphics processor.

User: exit

Chatbot Assistant:If you have any more questions in the future, feel free to ask. Have a great day!