ShopAssist AI 2.0

System Design Considerations:

Designing an AI system requires thoughtful planning to ensure it meets its goals while being efficient, secure, and reliable. One of the key choices is selecting the right Large Language Model (LLM), weighing the flexibility of open-source options against the dependability of proprietary solutions. Considerations like ease of use, cost, model performance, and handling of context must align with the specific needs of the domain. At the same time, strong safeguards for data protection and user privacy are essential to address risks such as unsafe outputs, harmful responses, or LLM hallucinations. Consistent and accurate results tailored to the domain are not just a technical goal—they're crucial for building trust in the system.

Basic design:

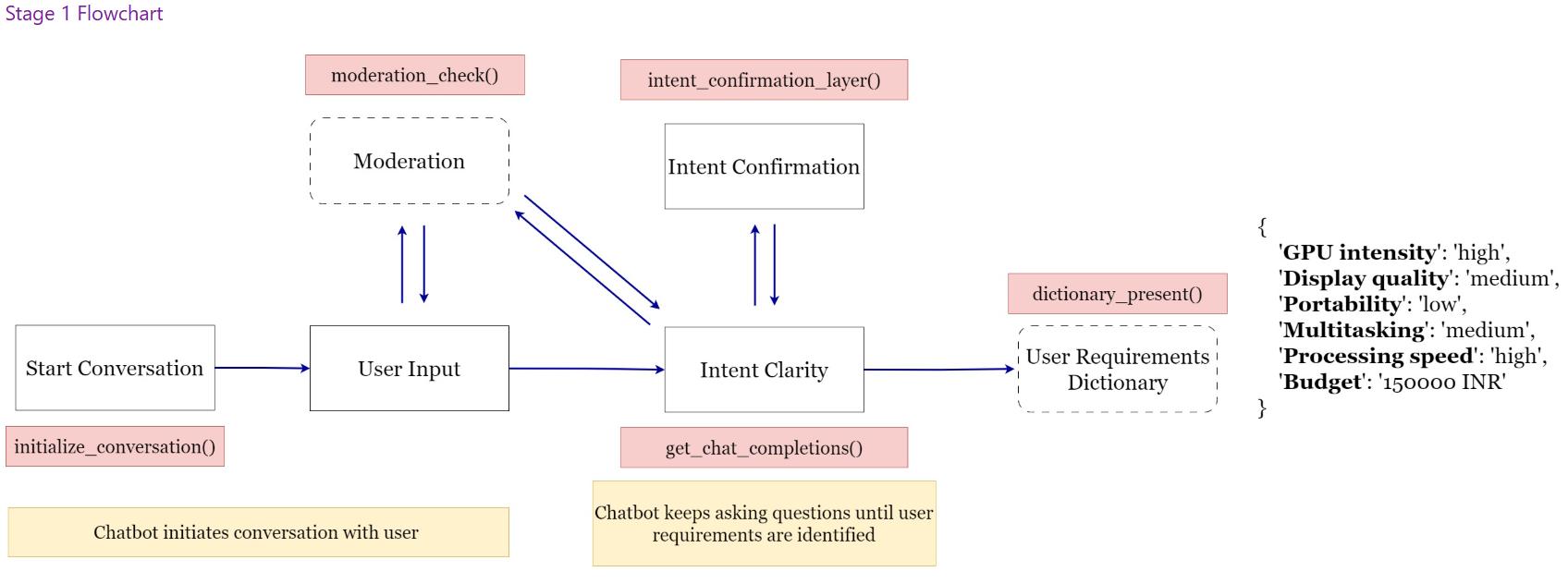
Laptop-database(.csv)

Product recommendation

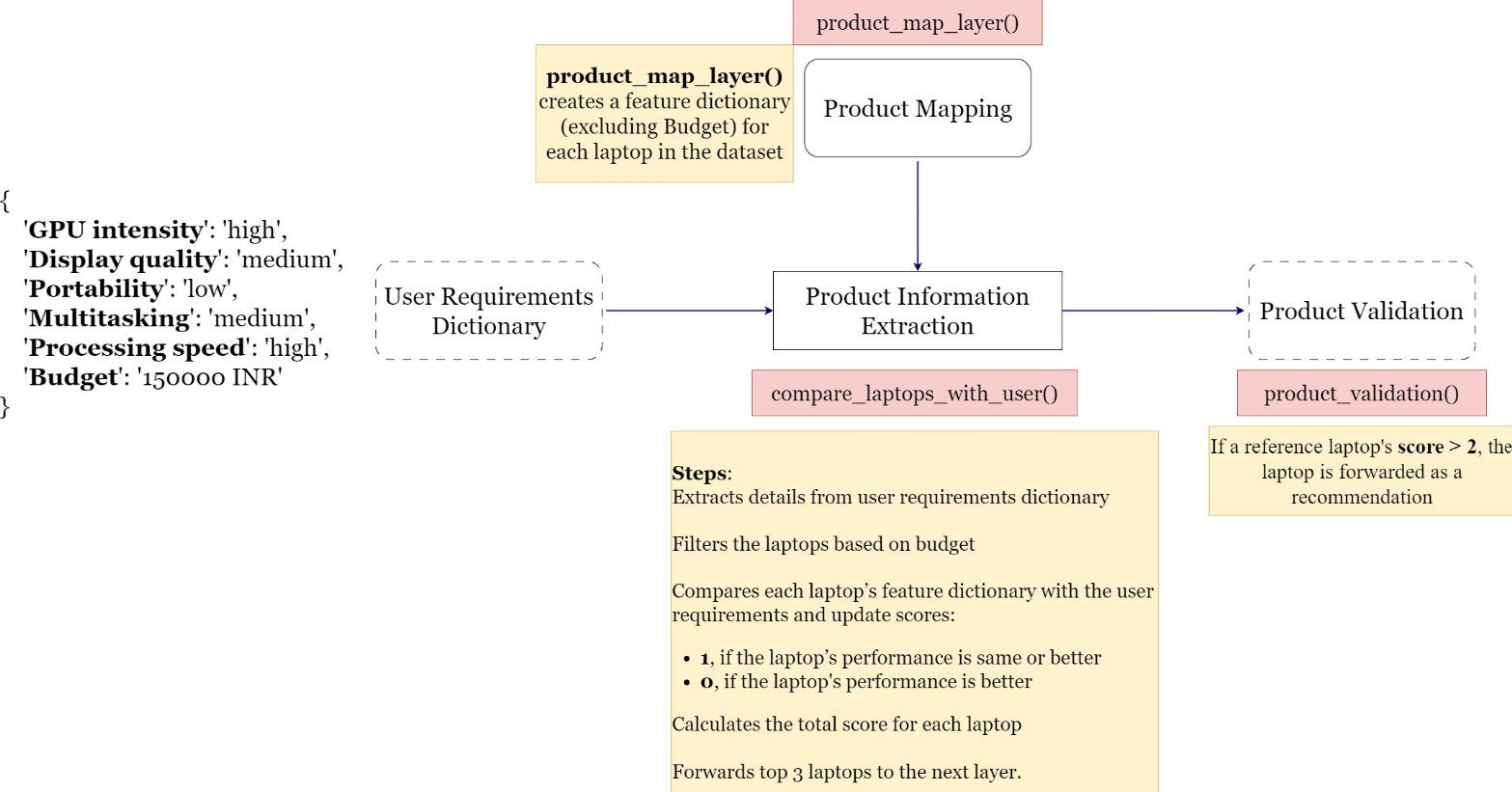
User Input

Chat bot retrieves information based on the input from the database.

Stage 1:



Stage 2:



We couldn’t reliably extract the required input parameters from users with 100% accuracy, often due to LLM hallucinations or incomplete or unclear input during conversations. To tackle this issue, we’re introducing a new feature called **function calling** as part of ShopAssist 2.0. This enhancement will allow us to better interpret and process user inputs seamlessly. Additionally, we’re implementing improvements to the user experience, making it easier for users to provide all necessary information upfront before we present the top 3 recommendations.

System Design:

Simplifying the solution by using function calling to integrate functions developed.

We couldn’t reliably extract the required input parameters from users with 100% accuracy, often due to LLM hallucinations or incomplete or unclear input during conversations. To address this challenge, we’re introducing **function calling**, a powerful feature in ShopAssist 2.0 that augments the capabilities of Large Language Models (LLMs).

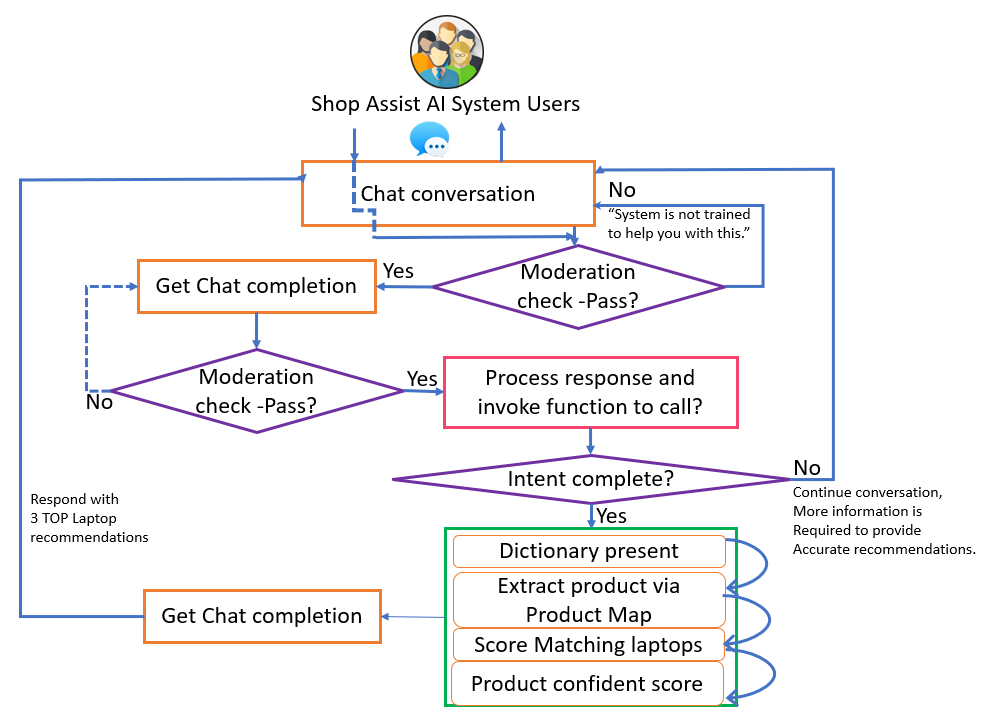
Function calling bridges the gap between natural language understanding and structured task execution. While LLMs are excellent at processing and interpreting human input, they can sometimes misinterpret requests or produce inaccurate outputs due to ambiguities in user input. Function calling simplifies this process by allowing the LLM to identify when a specific task needs to be performed and delegating the execution of that task to pre-defined, reliable functions.

For instance, when a user provides details about their shopping preferences, the LLM can recognize that it needs to retrieve product recommendations. Instead of attempting to generate the results itself, the LLM triggers the appropriate function—integrated into our system—that has been explicitly developed to handle such tasks. These functions are designed to fetch accurate data, validate inputs, and return structured outputs, ensuring reliability and precision.

This approach not only reduces errors caused by LLM hallucinations but also ensures that the system remains robust when handling complex or incomplete queries. By offloading critical tasks to well-defined functions, we can:

1. **Improve Accuracy**: Function calls ensure that computations and data retrieval are handled by tested logic rather than relying solely on LLM interpretations.
2. **Enhance Efficiency**: The system seamlessly connects user inputs to the correct functions, streamlining operations.
3. **Simplify Development**: Developers can focus on creating and refining specific functions for core tasks, knowing the LLM will call them when needed.
4. **Elevate User Experience**: Users benefit from clear, accurate outputs and a guided process for providing the necessary information.

In ShopAssist 2.0, function calling will not only handle core tasks like retrieving and ranking product recommendations but also help guide users through the input process by prompting for missing details. This ensures that by the time we generate the top 3 recommendations, all the necessary information has been collected and validated, providing a smoother and more reliable experience.



**More improvements in Data analysis:**

**Exploratory Data Analysis (EDA)**

We have a very small data set for the inventory to choose from and therefore we have few choices to choose from for a varied user’s requirements. User experience can be improved by expanding the inventory data available.

Here is a sample data conversion and analysis to improve system performance and predictive capabilities. The conversion of these fields and ranking them as Low, Medium, and High provides us ability to choose the best possible laptop/computer configuration based on the user needs. We are considering these 6 parameters in our design

**GPU**(Core), **Display quality** (Display Type , Display Size, screen resolution, Graphics Processor) , **Portability** ( Laptop weight, Average Battery Life), **Multitasking** , **Processing speed**(Clock Speed, RAM size), **budget**(Price, Warranty)

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Sample Value** | **Units/numeric** |
| **Brand** | VARCHAR(50) | Dell |  |
| **Model Name** | VARCHAR(100) | Inspiron |  |
| **Core** | VARCHAR(20) | i5 |  |
| **CPU Manufacturer** | VARCHAR(50) | Intel |  |
| **Clock Speed** | VARCHAR(20) | 2.4 GHz | GHz |
| **RAM Size** | VARCHAR(10) | 8GB | GB |
| **Storage Type** | VARCHAR(50) | SSD |  |
| **Display Type** | VARCHAR(50) | LCD |  |
| **Display Size** | VARCHAR(20) | 15.6" | Inches |
| **Graphics Processor** | VARCHAR(50) | Intel UHD |  |
| **Screen Resolution** | VARCHAR(20) | 1920x1080 | pixels |
| **OS** | VARCHAR(50) | Windows 10 |  |
| **Laptop Weight** | VARCHAR(20) | 2.5 kg | kg |
| **Special Features** | TEXT | Backlit Keyboard |  |
| **Warranty** | VARCHAR(20) | 1 year | years |
| **Average Battery Life** | VARCHAR(20) | 6 hours | hours |
| **Price** | DECIMAL(10,2) | 35,000 | currency |
| **Description** | TEXT | The Dell Inspiron is a versatile laptop that combines powerful performance and affordability. |  |

Display final message to the customer with the top 3 choices along with the description and how they can specifically solve for their use case/requirements. If we do not find the best match, direct to human assistant.

Function Calling capability with ChatGPT 3.5 and GPT-4 models, which allows calling user-defined functions as input to generate structured outputs. This eliminates the need for complex regular expressions and prompt engineering, thus making it easier for user to interact with AI systems.