ShopAssist AI 2.0

(Gaurav Kumar Sahoo)

This project builds an AI-powered conversational assistant using Open AI APIs for helping users find the best laptop based upon their needs.

It is an upgrade from ShopAssist AI 1.0, improving conversational flow and reducing code complexity by using function calls.

Goal:

The primary objective is to assess the user's requirements like GPU intensity, display quality, portability, multitasking capability, processing speed and budget, then recommend the most suitable laptop. If no optimal match is found, the user is redirected to human assistance.

Enhancements in Version 2.0:

- Removed dependency on predefined example user dictionaries in prompts. So we now no longer require hardcoded sample dictionaries such as

  example_user_dict = {"gpu_intensity": "high", "display_quality": "high", "portability": "low", "multitasking": "high", "processing_speed": "high", "budget": "150000"}

- Removed explicit `response_format` specification when calling the Chat Completion API, as function calling guarantees structured JSON output.

- Removed the Intent Confirmation step as with function calling, the model only invokes functions when all required parameters are present and valid, removing the need for a separate validation layer.

- Removed the User Requirements Dictionary step as function calling directly returns structured JSON arguments matching the defined schema.

- Implemented function calling in Product Mapping step to ensure consistent and accurate mapping of product features.

- Reduced prompt verification requirements as guaranteed structured output minimizes the need for extensive post-processing checks.

- Refined prompts for improved recommendation accuracy, leading to more precise and reliable results.

- Upgraded model from GPT-3.5-Turbo to GPT-4-0613 which improved reasoning ability and overall performance.

Block Diagram representing Function Call integration:
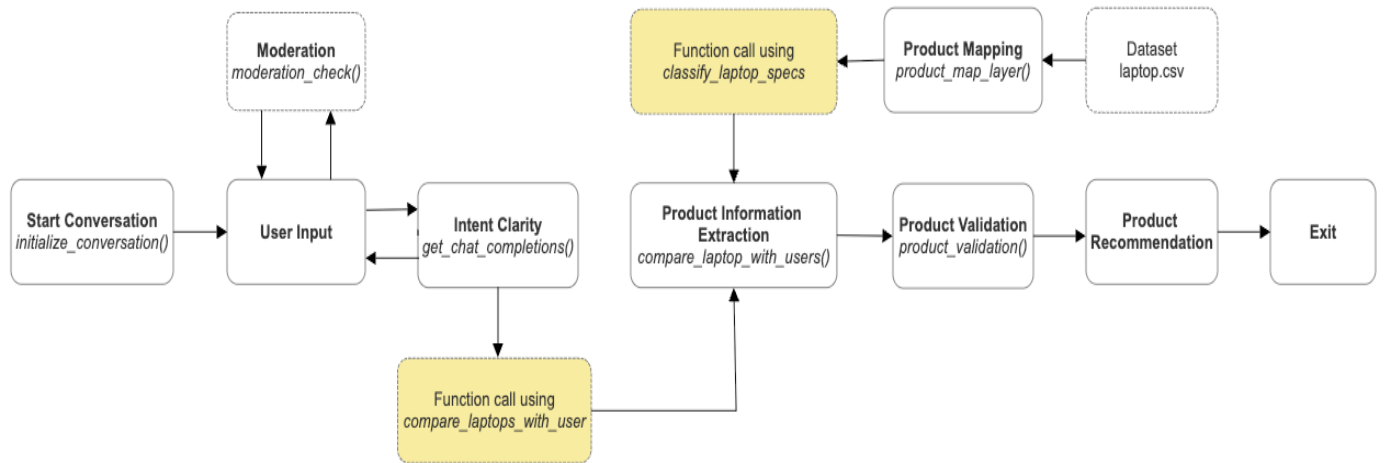
Conversation Flow:

- System role initiates the interaction by understanding the user's needs, asking no more than two feature-related questions at a time.

- Each user response undergoes a moderation check to ensure it contains safe and non-harmful input.

- The assistant continues to gather information until all required fields are collected, then triggers the function call with complete and validated data.

- The function output is compared against the updated laptop dataset containing features to identify the most suitable match.

- Updated laptop dataset is generated in product mapping step via function calls, extracting feature details from existing product descriptions.

- Best match selection is validated against a defined similarity threshold to ensure it closely meets the user's requirements.

- Recommendation is presented to the user in the desired format specified in the prompt.

- Conversation then may continue or end depending on whether the user wants to explore further options.

Importance of Using Function Calling in OpenAI API:

Function calling helps in reliably extracting structured data (like JSON) from the model's responses without depending on parsing. In ShopAssist AI 2.0, this is important because:

- Consistency - Instead of relying on the AI to follow output formatting in plain text, function calling enforces a schema which ensuring the output always has expected fields like "gpu_intensity" or "budget".

- Simpler Code and Automation - Once the model outputs structured data, it can directly be used in code functions without regex or manual parsing.

- Reduced Errors - Eliminates misinterpretations due to typos, formatting changes, or variations in phrasing.

- Better UX - The assistant can switch between asking questions and making recommendations smoothly because the data it collects is already structured.

Key Benefits of ShopAssist Version 2.0:

- Enhanced reliability in data extraction through the use of function calling.
- Reduced need for manual parsing of model responses.
- Easier integration with databases.
- Upgraded to GPT-4-0613 which offers better reasoning and smoother conversations.