**Shopping Assistant Report**

**1. Project Overview**

**Objective**

The goal of this project was to build an intelligent shopping assistant using the LangGraph framework. The assistant facilitates product search, recommendations, cart management, and order-related queries while ensuring an engaging and natural conversational flow.

**Features**

**Product Search**: Answer product-related queries such as specifications, prices, and availability.

**Recommendations**: Suggest related or alternative products based on user preferences or queries.

**Cart Management**: Enable users to add, view, or remove items from the cart, with real-time updates.

**Order Assistance**: Respond to queries about checkout, delivery, payment options, and order status.

**2. Implementation**

**Overview**

This Amazon Shopping Assistant is a CLI-based chatbot powered by a language model and database-backed tools for interacting with an Amazon-like shopping platform. It enables users to search for products, manage their shopping cart, process checkouts, and handle various order-related queries.

**Key Components**

**1. Language Model (LLM)**

**Framework**: LangChain, with integration via ChatGroq.

**Model**: llama-3.1-70b-versatile

**Features**:

Conversational interface.

Dynamic tool invocation for task execution.

Handles fallback scenarios for better error recovery.

**2. Database**

**Technology**: PostgreSQL (psycopg2 for Python).

**Usage**:

Product search and retrieval.

Order creation and management.

Retrieval of delivery times and order statuses.

**3. Tools**

**Purpose**: Extend the assistant’s functionality by providing specific actions such as:

Searching for products.

Adding/removing/viewing items in the cart.

Checkout process.

Retrieving payment options, order status, and delivery time.

**4. State Management**

**Graph-Based Workflow**: Managed by StateGraph from LangGraph.

**Memory**: Persisted via MemorySaver to maintain session continuity.

**Modules and Functions**

**Core Assistant Logic**

1. **Class:** Assistant

Implements the conversational loop, invoking the language model and tools as needed.

Ensures proper formatting of model responses.

2. **Prompt Templates**

**Primary Prompt**: Provides the assistant with system instructions and integrates tool usage dynamically.

3. **Tools Integration**

Tools are bound to the LLM using bind\_tools.

**Tools**

1. **Product Search**

Queries the database for products based on user-defined filters (e.g., name, category, price range, rating).

Returns a formatted list of products or an appropriate error message.

2. **Cart Management**

**Add to Cart** :

Adds products to the cart or updates quantities if the product exists.

**Remove from Cart:**

Removes specific quantities or entire items from the cart.

**View Cart:**

Returns a detailed summary of items in the cart, including quantities, prices, and the grand total.

3 . **Checkout (**checkout**)**

Processes the cart items into an order stored in the database.

Clears the cart post-checkout and provides a delivery date.

4. **Order Information**

**Delivery Time**

Fetches the delivery date for a specific order.

**Order Status**

Retrieves the current status of an order.

5. **Payment Options**

Returns available payment methods.

**Error Handling**

Tools are wrapped with a fallback mechanism that prompts the LLM to provide alternative responses or error messages when issues occur.

**CLI Interaction**

The user interacts with the assistant through a CLI interface.

Commands:

• exit **/** quit: Ends the session.

• Free-form queries: Sent to the assistant for processing.

**Workflow**

1. **Initialization**

The assistant initializes with a unique thread ID and user-specific configurations.

2. **User Interaction**

User inputs are passed through the state graph, invoking the LLM or tools as required.

3. **Tool Execution**

If the LLM identifies a task requiring a tool, the corresponding tool is invoked.

4. **Response Generation**

Results from tools or LLM are returned to the user in a human-readable format.

Sample Interaction with ai assistant

A screenshot of a computer

Description automatically generated.

A screenshot of a computer

Description automatically generatedAs you can see, product search is being performed according to user query, also results are getting fetched from our database.

A screenshot of a computer

Description automatically generatedHere, you can see ai agent is deciding based on context of user query, which function needs to be called.For instance, cart management and checkout is being called in above example.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedFrom above illustrations one can see that Ai Agent is successfully understanding the context of user queries and based on that calling necessary tools to perform the task. For instance. Here agent is using tools relates order status and delivery time.

A screenshot of a computer program

Description automatically generatedHere, you can clearly see that agent was able to remember the context and performing accordingly.