**Assignment Documentation**

**Summary:**

The integration of the OpenAI model into negotiation chatbot leverages several key components, with a focus on structuring the flow of conversation, maintaining context, and ensuring the model's outputs align with the negotiation logic.

**Dependencies:**

**FastAPI**: For creating the API endpoints.

**LangChain**: For managing conversation chains and logic interaction with ChatGPT.

**OpenAI**: To integrate with OpenAI’s ChatGPT.

**spaCy:** For natural language processing and extracting information from user messages.

**Uvicorn**: For running the FastAPI application.

**1.Loading and Setting up the Model**

The OpenAI language model (GPT-3.5-turbo) is accessed through the **ChatOpenAI** class from LangChain, which is initialized with the API key:

**2.Maintaining Conversation Memory**

Memory is a crucial part of a conversational agent, as it allows the model to maintain the context of previous messages. This is handled using **ConversationBufferMemory**, which stores the ongoing conversation history:

The prompt ensures that the model understands its role as a negotiator and responds with strict price adjustments or acceptance without unnecessary elaboration. The variables like **current\_price**, **min\_price**, and **discount\_step** are dynamically injected based on the state of the conversation.

**3.Defining the Negotiation Prompt**

The negotiation logic is defined as a **template** using the PromptTemplate class. This ensures the model understands its role in the negotiation process:

**4.Handling Conversation with ConversationChain**

Here use Conversationchain which is designed to manage both the conversation memory and the invocation of the language model.

**5.Invoking the Model with Structured Input**

Instead of passing a raw string to the model, Here I pass structured messages that include both the user's input and the system prompt:

**6.Dynamic Negotiation Logic**

After the bot generates a response, the code dynamically adjusts the current\_price based on the user's requests.

**7.FastAPI for API Endpoint Creation**

The API is built with **FastAPI** to handle HTTP requests. The /negotiate endpoint processes user input, applies negotiation logic, and returns a response formatted by the ChatGPT model using LangChain.

**8. Running the API and Generating Responses**

The model's output is structured as a response based on the negotiation rules.

**This setup allows the model to follow strict negotiation rules, dynamically adjust prices, and maintain conversation context.**