**Integration of OpenAI GPT-3.5 with FastAPI for Price Negotiation**

**This FastAPI-based application integrates the OpenAI GPT-3.5 model to handle user price negotiations. Here’s how the integration works:**

**Purpose**

**The app allows users to submit a proposed price for a product and receive a negotiation response. The system generates a counteroffer based on the user’s price and a predefined acceptable price range. It also simulates a discount to make the counteroffer more appealing.**

**Key Components**

**- FastAP: A modern web framework used to build APIs. It handles user requests and sends the appropriate responses.**

**-OpenAI GPT-3.5 Mode: The model is responsible for generating conversational responses during the negotiation. It acts as a virtual assistant that helps in responding to the user’s price proposals.**

**-Negotiation Logic The app uses a price range to determine whether the user’s offer is acceptable. If the price is too low or too high, the system responds directly. If the price falls within the acceptable range, GPT-3.5 generates a negotiation response, and a random discount is applied to the proposed price to create a counteroffer.**

**- Dynamic Counteroffers: After GPT-3.5 generates a conversational response, the system randomly generates a discount (between 0-20%) and applies it to the user’s offer to propose a counteroffer, simulating a negotiation process.**

**WorkFlow**

**-Price Submissio: The user submits a price via a POST request to the `/negotiate` endpoint.**

**-Price Evaluation: The app checks whether the submitted price is within the predefined range. If it's below or above the range, the app responds with a message indicating the minimum or maximum price allowed.**

**- GPT-3.5 Response Generation: If the price is within range, the OpenAI GPT-3.5 model generates a response. The model is pre-configured with context, telling it to act as a negotiation assistant, making the conversation more realistic.**

**- Counteroffer: The app applies a discount to the user’s offer (using a random discount percentage), and this is included in the final response to the user.**

**- Offer Acceptance: If the user accepts the counteroffer, they can submit the final price via the `/accept` endpoint, and the app responds by confirming the acceptance.**

**EndPoints**

**- /negotiate: This endpoint processes the user’s proposed price and returns a negotiation response, including the counteroffer generated by GPT-3.5.**

**- /accept: This endpoint handles the user’s acceptance of the counteroffer, finalizing the negotiation process.**

**Model**

**GPT-3.5 is used to generate human-like negotiation responses. It receives input from the user’s price proposal and generates a conversation tailored around negotiating that price. This makes the system feel more interactive and engaging.**

**The application includes basic error handling for requests that fall outside the acceptable price range. It also catches potential issues related to the GPT-3.5 model (such as API errors or rate limits) to ensure that the app responds gracefully.**

**App**

**The app runs locally using Uvicorn, a high-performance ASGI server. This allows the FastAPI app to listen for incoming requests, process them, and return the appropriate responses.**

**Conclusion**

**This application combines FastAPI for building the web API, OpenAI's GPT-3.5 for conversational responses, and basic negotiation logic to provide an interactive price negotiation experience. The integration enables users to negotiate product prices in real time with the help of AI, making it a dynamic and responsive system.**