**Project Documentation: Negotiation Chatbot Using GPT-Based AI in VS Code**

**Objective:**

The goal of this project is to implement a terminal-based chatbot that simulates a negotiation process between a customer and a supplier, using GPT-based AI. The chatbot will handle price offers and user messages, responding dynamically with acceptance, rejection, or counteroffers. Additionally, sentiment analysis will be applied to modify the chatbot’s behavior based on the user’s tone.

**Step-by-Step Instructions for Client:**

**Step 1: Set Up VS Code**

1. **Install VS Code**: If not already installed, download and install Visual Studio Code from [here](https://code.visualstudio.com/).
2. **Install Python**: Ensure Python is installed on your system. If not, download and install Python from [here](https://www.python.org/downloads/).
3. **Open a New Folder**:
   * Open VS Code and create a new folder for the project. You can name it something like NegotiationChatbot.
   * Open the folder in VS Code by selecting File -> Open Folder....

**Step 2: Install Required Extensions**

1. **Python Extension**:
   * In VS Code, navigate to the Extensions Marketplace (on the left-hand sidebar).
   * Search for Python and install the Python extension by Microsoft. This will provide support for Python development in VS Code.
2. **Terminal Setup**:
   * You can access the terminal in VS Code by selecting Terminal -> New Terminal. This will be used to run Python commands and install packages.

**Step 3: Set Up Virtual Environment (Optional, but recommended)**

1. **Create a Virtual Environment**:
   * Open the terminal in VS Code.
   * Navigate to the project folder and run the following command to create a virtual environment:

bash

Copy code

python -m venv venv

* + This creates a folder called venv, which will isolate the project dependencies from the global environment.

1. **Activate the Virtual Environment**:
   * On **Windows**, run:

bash

Copy code

.\venv\Scripts\activate

* + On **macOS/Linux**, run:

bash

Copy code

source venv/bin/activate

**Step 4: Install Required Libraries**

1. **Install openai for GPT Model Access**:
   * In the terminal, run:

bash

Copy code

pip install openai

* + This installs the OpenAI library, which is used to interact with GPT-based AI models (e.g., GPT-3).

1. **Install nltk for Sentiment Analysis**:
   * In the terminal, run:

bash

Copy code

pip install nltk

* + This installs the Natural Language Toolkit (NLTK), which is required for analyzing sentiment in the user’s messages.

1. **Download the Sentiment Lexicon**:
   * After installing nltk, you need to download the VADER lexicon used for sentiment analysis. In the terminal, run:

bash

Copy code

python -m nltk.downloader vader\_lexicon

**Step 5: Configure OpenAI API Key**

1. **Obtain an OpenAI API Key**:
   * Sign up at [OpenAI](https://beta.openai.com/signup/) to get access to GPT-3.
   * Navigate to your account settings and generate an API key.
2. **Set the API Key in the Code**:
   * When you write your Python code, set the openai.api\_key variable to your API key (this step will be performed inside the code, not in VS Code). Make sure the key is kept confidential.

**Step 6: Write the Python Code**

1. **Create a Python File**:
   * In the project folder, create a new Python file (e.g., chatbot.py) where the logic for the negotiation chatbot will be written.
   * The chatbot will use OpenAI’s GPT-3 to simulate negotiation, and NLTK will be used for sentiment analysis.

**Step 7: Test the Chatbot**

1. **Run the Python File**:
   * After writing the chatbot code, save the file.
   * Run the script by typing the following command in the terminal:

bash

Copy code

python chatbot.py

1. **Interaction in the Terminal**:
   * You will be prompted in the terminal to enter your price offer and message. The chatbot will respond based on the negotiation logic and sentiment of your message.

**Step 8: Modify and Extend the Chatbot (Optional)**

1. **Add More Features**:
   * You can further extend the chatbot with additional logic, such as more complex negotiation strategies, discounts, or multi-round negotiations.
2. **Enhance Sentiment Analysis**:
   * If desired, you can fine-tune the sentiment analysis to provide different responses based on how positive or negative the user’s tone is.

**Step 9: Finalizing the Project**

1. **Testing**:
   * Test the chatbot multiple times to ensure it functions as expected. Test different price offers and tones to see how the chatbot reacts.
2. **Deployment (Optional)**:
   * If needed, this chatbot can later be converted into a web-based app or integrated with other systems. For now, it will function as a terminal-based chatbot.

**Deliverables:**

* **Python file (chatbot.py)**: Contains all the chatbot logic.
* **Requirements (openai, nltk)**: The necessary libraries installed via pip.
* **API Key Setup**: Instructions for setting up OpenAI API access.
* **Sentiment Analysis**: Sentiment handling through NLTK.

**Project Overview:**

This project is designed to work in a **local development environment** using **Visual Studio Code** and Python. It leverages the **OpenAI GPT model** for generating negotiation responses and **NLTK** for sentiment analysis. The interaction is purely terminal-based, with dynamic responses depending on user inputs.