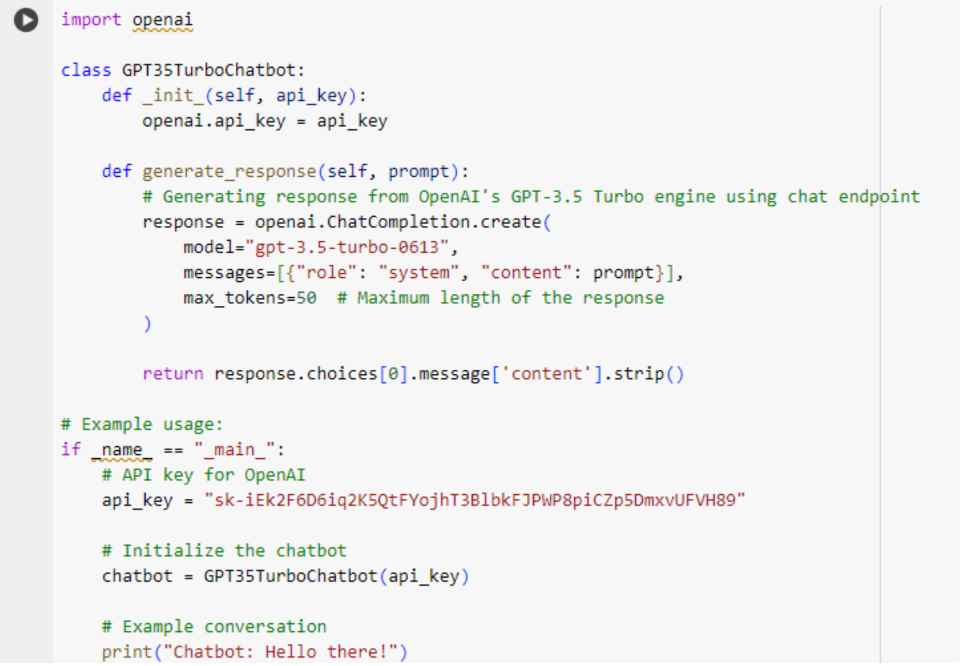
Lawrence
Technological
University®

NAME: - ANUPAMA AVUSULA

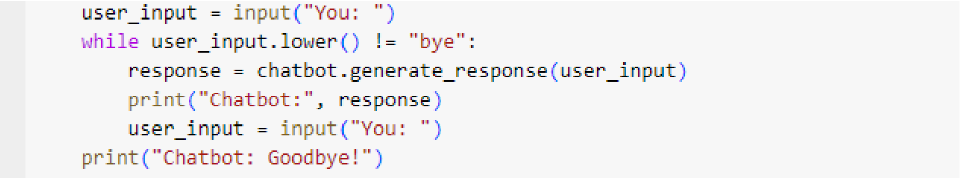
BANNER ID: - 000786453

API Integrated Chatbot AI

CODE: -

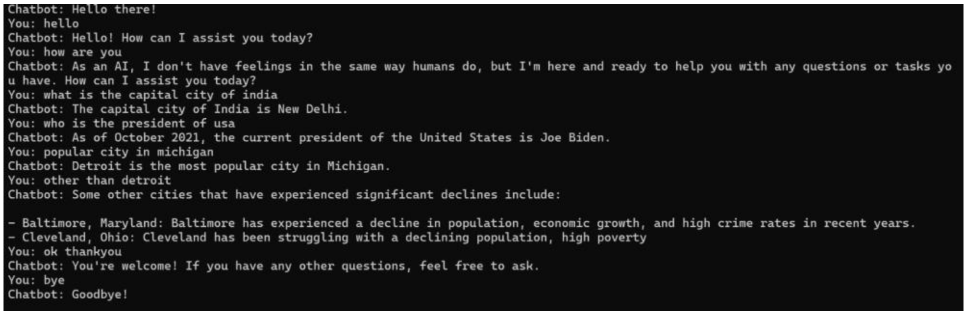


```python
import openai

class GPT35TurboChatbot:
    def _init_(self, api_key):
        openai.api_key = api_key

    def generate_response(self, prompt):
        # Generating response from OpenAI's GPT-3.5 Turbo engine using chat endpoint
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo-0613",
            messages=[{"role": "system", "content": prompt}],
            max_tokens=50  # Maximum length of the response
        )

        return response.choices[0].message['content'].strip()

# Example usage:
if _name_ == "_main_":
    # API key for OpenAI
    api_key = "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"

    # Initialize the chatbot
    chatbot = GPT35TurboChatbot(api_key)

    # Example conversation
    print("Chatbot: Hello there!")
```



```python
    user_input = input("You: ")
    while user_input.lower() != "bye":
        response = chatbot.generate_response(user_input)
        print("Chatbot:", response)
        user_input = input("You: ")
    print("Chatbot: Goodbye!")
```

OUTPUT: -



```
Chatbot: Hello there!
You: hello
Chatbot: Hello! How can I assist you today?
You: how are you
Chatbot: As an AI, I don't have feelings in the same way humans do, but I'm here and ready to help you with any questions or tasks yo
u have. How can I assist you today?
You: what is the capital city of india
Chatbot: The capital city of India is New Delhi.
You: who is the president of usa
Chatbot: As of October 2021, the current president of the United States is Joe Biden.
You: popular city in michigan
Chatbot: Detroit is the most popular city in Michigan.
You: other than detroit
Chatbot: Some other cities that have experienced significant declines include:

- Baltimore, Maryland: Baltimore has experienced a decline in population, economic growth, and high crime rates in recent years.
- Cleveland, Ohio: Cleveland has been struggling with a declining population, high poverty
You: ok thankyou
Chatbot: You're welcome! If you have any other questions, feel free to ask.
You: bye
Chatbot: Goodbye!
```

IMPORTING LIBRARIES: -

To write and execute the code for this integrated chatbot we need to install a library named “openai”, for interacting with OpenAi’s GPT-3.5 API.

Openai: - For many different natural language processing applications, developers can easily use OpenAI's robust artificial intelligence models, such GPT-3, through the OpenAI library. It provides features for text generation, completion, and classification, enabling users to effortlessly include advanced AI capabilities into their apps.

SET UP AND API KEY: -

Firstly, we need to install python before starting the code.Next, use pip install openai to install the OpenAI library. Get an API key from the OpenAI platform, then enter it in the code. In order to communicate with OpenAI's GPT-3.5 Turbo model for conversational answers, the chosen key should authenticate queries.

API INTEGRATION: -

To integrate the OpenAI API, obtain an API key from OpenAI's platform and replace it in the code. Make sure the **openai** library is installed (**pip install openai**). This key authenticates requests to OpenAI's servers, enabling interaction with their GPT-3.5 Turbo model. With the key set, the code initializes a chatbot class that interacts with the model for generating conversational responses. The **generate_response** method sends prompts to the model, receiving responses based on the provided input. This integration empowers users to leverage OpenAI's advanced natural language processing capabilities easily within their applications.

PERSONALITY CUSTOMIZATION: -

Introduce a personality parameter in the GPT35TurboChatbot class to allow for personality customisation. Combine user questions with this personality feature before submitting them to the GPT-3.5 Turbo model for response creation. For example, "Friendly Bot" is the set personality. When this feature is used, the chatbot's conversation takes on unique characteristics such as professionalism, friendliness, or comedy, which improves user engagement and interaction experiences by providing customized responses that reflect the selected personality.

MODEL EXPLAINATION: -

A cutting-edge natural language processing model from OpenAI called GPT-3.5 Turbo is used in the code. It communicates with the model using the ChatCompletion API endpoint by utilizing the openai package. A chatbot's personality can be customized using the GPT35TurboChatbot class, which initializes using an API key. The GPT-3.5 Turbo model, which creates conversational responses, receives cues from the generate_response method together with the provided personality. With the help of this integration, users may engage with the potent AI model with ease, opening up a variety of applications for chatbot development, automated customer support, and text generating activities that produce responses that resemble those of a human.

PROMPT ENGINEERING: -

In order to effectively direct conversations, prompt engineering for this chatbot entails creating interesting and contextually relevant prompts. Begin by extending a cordial welcome to build rapport, incorporating personal qualities to influence answers, and offering background information to enhance comprehension of user inquiries. To retain clarity and to promote user participation, utilize open-ended questions and acknowledge user feedback. Include prompts asking users to provide feedback so that the chatbot can continuously improve. You may build more meaningful and engaging conversations, improving the chatbot's overall effectiveness in meeting user needs and improving the user experience overall, by designing prompts that are specific to the chatbot's personality and purpose.

CONCEPTUAL TRAINING EXPLANATION: -

The training procedure conceptually makes use of OpenAI's GPT-3.5 Turbo model, which has been widely pre-trained on conversational data. In order to reduce the difference between the generated and ground truth replies, data is gathered, preprocessed, and then input into the model. Text production and processing are facilitated by the transformer design. Even though the offered code doesn't directly require fine-tuning, model refinement on certain datasets or tasks is a regular technique. Using OpenAI's API, the model generates answers to user inquiries after it has been trained. Real-time meaningful interactions are made possible by the procedure, which endows the model with the capacity to produce coherent and contextually relevant responses.