OpenAI API and GPT-4o Mini Chatbot Implementation

From basics to building a Chatbot using GPT-4o Mini and Streamlit

By: Shruti Kavishwar
Guided By: Prof. Henry Chang
San Francisco Bay University

GitHub: [GenAI/ChatGPT at main · ShrutiK02/GenAI (github.com)]

What is OpenAI API ?

- **Overview**: The OpenAI API allows developers to integrate AI-powered text generation capabilities into applications using models like GPT-4o and GPT-4o Mini.
- **GPT 4o vs GPT- 4o Mini:**
  - GPT-4o Mini is faster and more affordable.
  - GPT-4o is more powerful and accurate, suitable for complex tasks.
- **Customization in AI**: While tools like ChatGPT are great, creating custom models via API gives better control.

Why Build a Custom AI Chatbot?

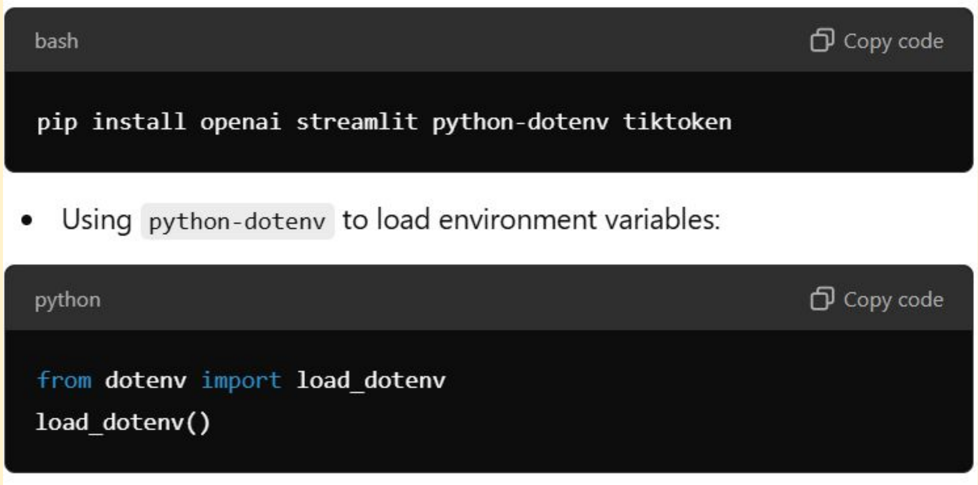
- **Why Custom Chatbots?** Pre-built AI tools are limited in customization.
- **AI Customization:** By using OpenAI API, developers can tailor the chatbot's behavior, appearance, and capabilities.
- **Message Roles:**
    - **User:** Sends queries.
    - **Assistant:** Provides responses.
    - **System:** Sets the overall behavior of the assistant (e.g., tone or format of responses).
- **Example**: A chatbot customized for handling customer support with a friendly tone.
-

Setting up the Environment

Project Setup:

- Create project folder.
- Install required libraries: OpenAI, Streamlit, Python-dotenv, and tiktoken.
- Generate OpenAI API key and store it in .env file.

Code Example:



```bash
pip install openai streamlit python-dotenv tiktoken
```

- Using `python-dotenv` to load environment variables:

```python
from dotenv import load_dotenv
load_dotenv()
```

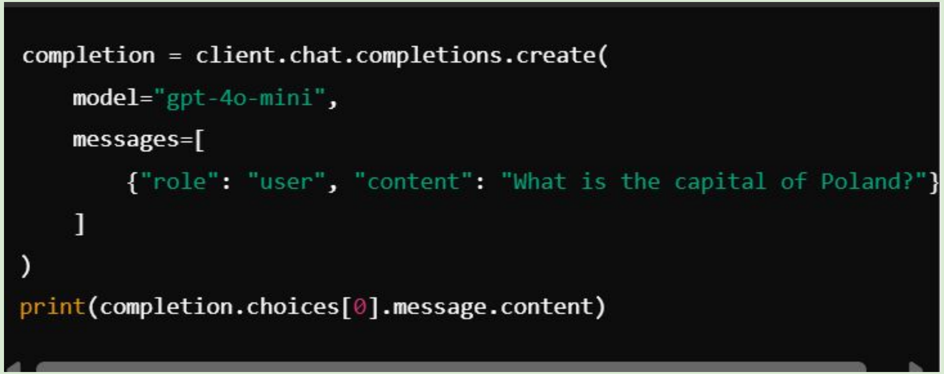
Making API Calls with GPT-4o Mini

Basic API Call Structure:

- Using the OpenAI API to send queries and retrieve responses.
- Roles involved: User, Assistant, and System.

Example Query:

- **Prompt**: "What is the capital of Poland?"
- **API Call:**



```
completion = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "user", "content": "What is the capital of Poland?"}
    ]
)
print(completion.choices[0].message.content)
```

- **Response**: "The capital of Poland is Warsaw."

Tuning GPT -4o Mini with Parameters

Temperature: Controls randomness in responses.

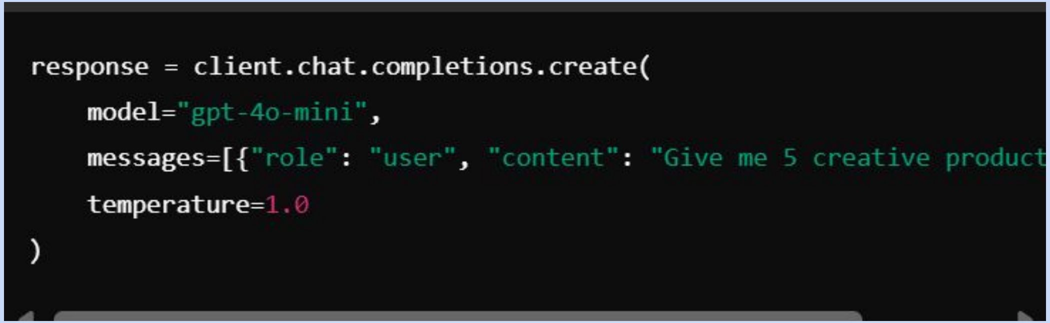
- Low temperature (0.1): More predictable, fact-based responses.
- High temperature (1.0): More creative, random responses.

Max Tokens: Limit the number of tokens returned in the response.
Seed: Reproduce specific results by setting a fixed random seed.

Example: Generating creative product names for eco-friendly sportswear.

- Low temperature result: Conservative names.
- High temperature result: Creative names.

Code Example for Temperature Control



```
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": "Give me 5 creative product
    temperature=1.0
)
```

Managing Tokens in GPT -4o Mini

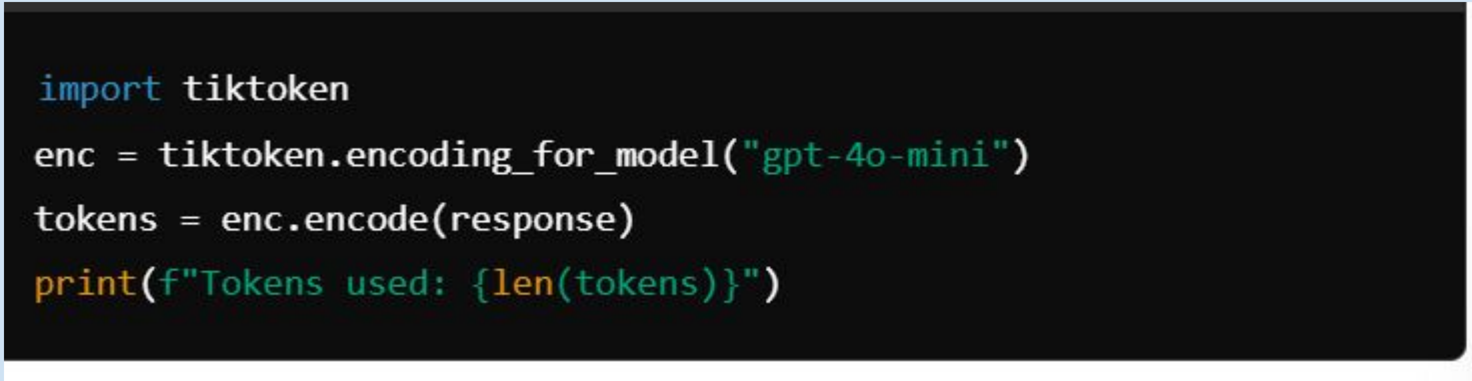
What are Tokens?

- Tokens are chunks of text used by language models (words, punctuation, symbols).
- Rough rule: 1 token ≈ 3/4 of a word.

Counting Tokens in Python:

- Example response token count using tiktoken library.

Token Usage Example:

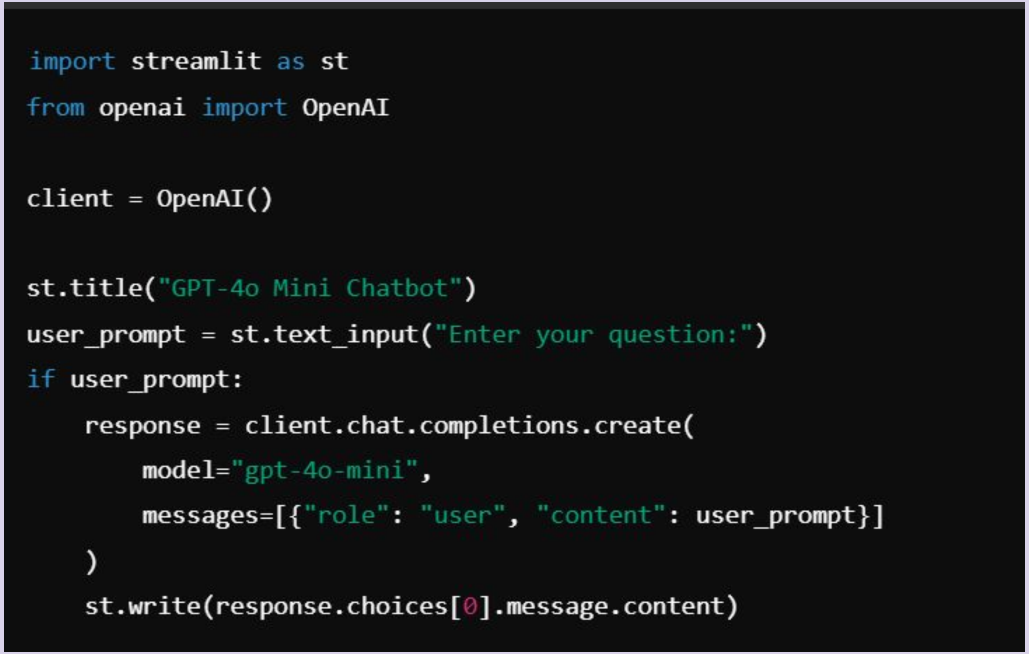


```python
import tiktoken
enc = tiktoken.encoding_for_model("gpt-4o-mini")
tokens = enc.encode(response)
print(f"Tokens used: {len(tokens)}")
```

Building the GPT -4o Mini Chatbot UI

Setting up the UI with Streamlit:

- Use Streamlit to create a user-friendly interface for chatbot interaction.

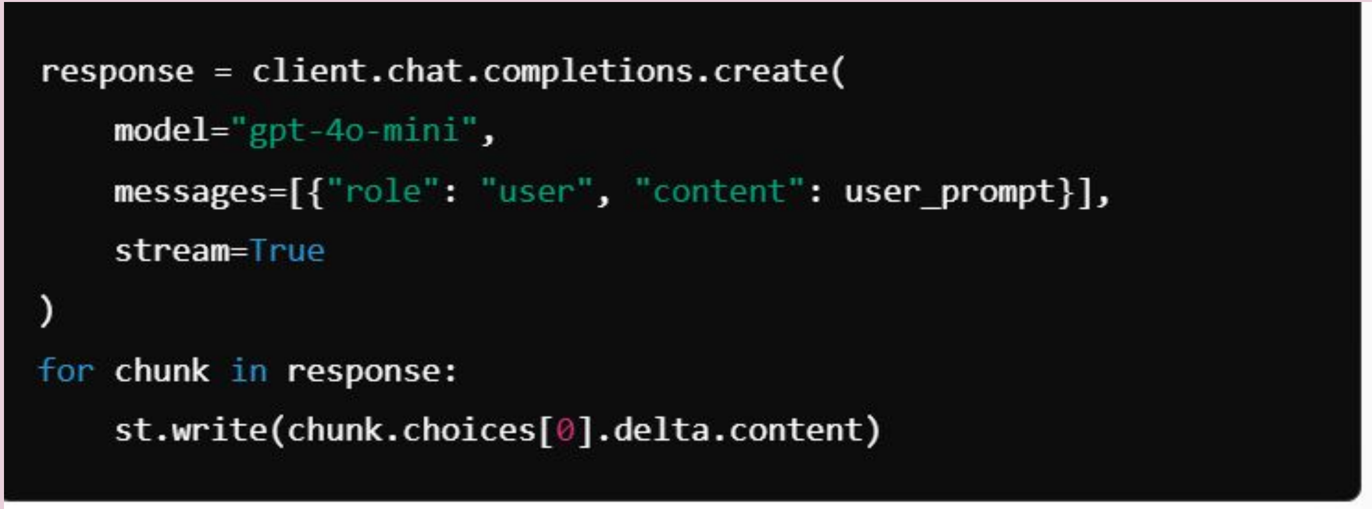


```python
import streamlit as st
from openai import OpenAI

client = OpenAI()

st.title("GPT-4o Mini Chatbot")
user_prompt = st.text_input("Enter your question:")
if user_prompt:
    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[{"role": "user", "content": user_prompt}]
    )
    st.write(response.choices[0].message.content)
```

Real-Time Response Streaming

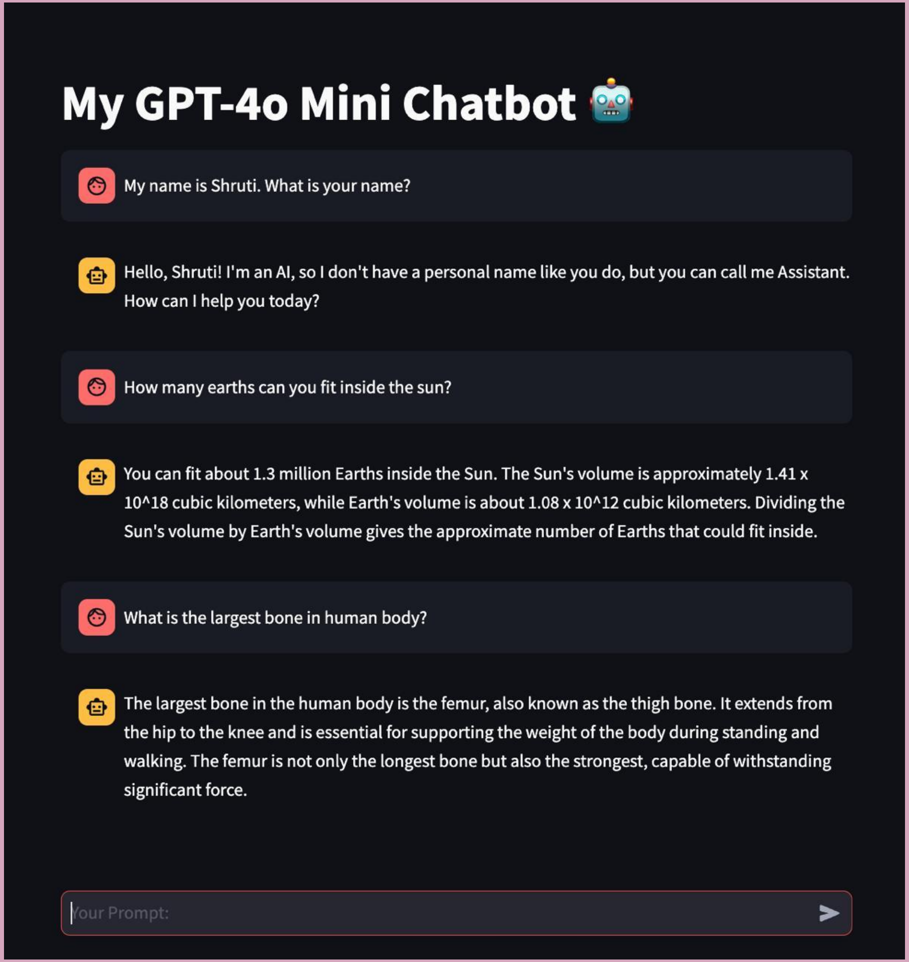
Streaming vs Non-Streaming:

- Non-streaming: Users wait for the entire response before seeing output.
- Streaming: Users see output as it's being generated.



```
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": user_prompt}],
    stream=True
)
for chunk in response:
    st.write(chunk.choices[0].delta.content)
```

Results



My GPT-4o Mini Chatbot 🤖

My name is Shruti. What is your name?

Hello, Shruti! I'm an AI, so I don't have a personal name like you do, but you can call me Assistant. How can I help you today?

How many earths can you fit inside the sun?

You can fit about 1.3 million Earths inside the Sun. The Sun's volume is approximately 1.41 x 10^18 cubic kilometers, while Earth's volume is about 1.08 x 10^12 cubic kilometers. Dividing the Sun's volume by Earth's volume gives the approximate number of Earths that could fit inside.

What is the largest bone in human body?

The largest bone in the human body is the femur, also known as the thigh bone. It extends from the hip to the knee and is essential for supporting the weight of the body during standing and walking. The femur is not only the longest bone but also the strongest, capable of withstanding significant force.

Recap and Next Steps

- **Summary of Learnings:**
  - Building custom chatbots with GPT-4o Mini.
  - Managing language model parameters and token usage.
  - Creating user interfaces with Streamlit.
- **Next Steps:**
  - Experiment with building your own chatbot.
  - Explore advanced features of GPT models.