

DS565 - Generative AI-Driven Intelligent Apps Development

Project: ChatBot Using Streamlit

Course: DS565

Professor: Dr. Henry Chang

Presented by: Karan Shrestha (20087)



Introduction to OpenAI API

- **What is OpenAI API?**
 - Provides access to powerful language models (GPT series)
 - Applications: Text generation, chatbots, translation, etc.
- **Why GPT-4o Mini?**
 - Efficient, lightweight version of GPT-4, optimized for quick interactions



Streamlit Overview

- **What is Streamlit?**
 - Python framework for building web apps quickly
 - Focuses on simplicity and real-time interaction
- **Why Use Streamlit?**
 - Easy to integrate with OpenAI for chatbot development
 - Great for prototyping interactive apps



Chatbot Architecture

- **Key Components**
 - **OpenAI API:** Powers the chatbot responses
 - **Streamlit:** Manages the user interface
 - **Session State:** Maintains the conversation history
- **Flow of Interaction**
 - User input → OpenAI API → Response → Display in Streamlit



Code Walkthrough (Key Parts)

- **1. Setting Up**
 - Import necessary libraries: `openai`, `streamlit`, and `dotenv`
 - Initialize the OpenAI client using `OpenAI()`
- **2. Capturing User Input**
 - Use `st.chat_input("Your Prompt:")` for user interaction
 - Store conversation in `st.session_state["messages"]`



Code Walkthrough (Response Generation & Display)

- **1. Sending Input to OpenAI**
 - Use `client.chat.completions.create` to send prompts to GPT-4o Mini
 - Stream responses in real time
- **2. Displaying Messages**
 - Loop through conversation history and display with `st.markdown()`
 - Real-time conversation updates

▼ CHATBOT_STREAMLIT

⚙️ .env

🔗 chatbot.py

🔗 chatbot.py > ...

```
1  import streamlit as st
2  from openai import OpenAI
3
4  from dotenv import load_dotenv
5
6  load_dotenv()
7
8  client = OpenAI()
9
10 st.title("My GPT-4o Mini Chatbot 🤖")
11
12 # Initialize messages in the session state
13 if "messages" not in st.session_state:
14     st.session_state.messages = []
15
16 # Display messages
17 for message in st.session_state["messages"]:
18     with st.chat_message(message["role"]):
19         st.markdown(message["content"])
20
21 if user_prompt := st.chat_input("Your Prompt:"):
22     st.session_state.messages.append({"role": "user", "content": user_prompt})
23     with st.chat_message("user"):
24         st.markdown(user_prompt)
25
26     with st.chat_message("assistant"):
27         chatbot_msg = st.empty()
28         full_response = ""
29         stream = client.chat.completions.create(
30             model="gpt-4o-mini",
31             messages=f
```



Running the Chatbot

- **Run Command:** `streamlit run chatbot.py`
- **Local URL:** View in browser via local/remote network URL provided by Streamlit

My GPT-4o Mini Chatbot



hi



Hello! How can I assist you today?

Your Prompt:





Write a code in python for prime numbers.



Sure! Here is a simple Python code to check for prime numbers and also to generate a list of prime numbers up to a given limit.

Checking if a single number is prime:

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Example usage
number = 29
if is_prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

Your Prompt:

