

Streamlit chat app

Code:

```
app.py 2 x
C: > streamlit-chat-app > app.py > ...
1 import streamlit as st
2 import os
3 from groq import Groq
4
5 # Set up the page configuration
6 st.set_page_config(page_title="Groq Chat App", layout="wide")
7
8 # Get API key from secrets.toml
9 groq_api_key = st.secrets["GROQ_API_KEY"]
10 client = Groq(api_key=groq_api_key)
11
12 # Select Groq model
13 model = st.sidebar.selectbox("Select a Groq Model", ["llama3-8b-8192", "llama3-70b-8192", "mixtral-8x7b-32768", "gemma-7b-it"])
14
15 st.title("🗨️ Groq Chat App")
16
17 # Initialize chat history
18 if "messages" not in st.session_state:
19     st.session_state.messages = []
20
21 # Display chat messages from history
22 for message in st.session_state.messages:
23     with st.chat_message(message["role"]):
24         st.markdown(message["content"])
25
26 # Accept user input
27 if prompt := st.chat_input("What is up?"):
28     # Add user message to chat history
29     st.session_state.messages.append({"role": "user", "content": prompt})
30     # Display user message in chat message container
31     with st.chat_message("user"):
32         st.markdown(prompt)
33
34     # Display assistant response in chat message container
35     with st.chat_message("assistant"):
36         # Display assistant response in chat message container
37         with st.chat_message("assistant"):
38             message_placeholder = st.empty()
39             full_response = ""
40
41             # Construct the messages payload for the API
42             messages_payload = [
43                 {"role": m["role"], "content": m["content"]}
44                 for m in st.session_state.messages
45             ]
46
47             # Stream the response from Groq with the corrected method
48             try:
49                 stream = client.chat.completions.create(
50                     model=model,
51                     messages=messages_payload,
52                     stream=True
53                 )
54                 for chunk in stream:
55                     (variable) full_response: Any | LiteralString
56                     full_response += chunk.choices[0].delta.content or ""
57                     message_placeholder.markdown(full_response + "▮")
58             except Exception as e:
59                 st.error(f"An error occurred: {e}")
60                 full_response = "I'm sorry, an error occurred while processing your request."
61
62             # Finalize the response display
63             message_placeholder.markdown(full_response)
64
65 # Add assistant response to chat history
66 st.session_state.messages.append({"role": "assistant", "content": full_response})
```

Streamlit chat app

Output:

