import os

import streamlit as st

import google.generativeai as genai

# Initialize the Gemini-Pro model

os.environ['GOOGLE\_GEMINI\_KEY']="AIzaSyBoVQYDOeYH8L4GZRL6b82nJAZJi-A"

genai.configure(api\_key=os.getenv('GOOGLE\_GEMINI\_KEY'))

model = genai.GenerativeModel('gemini-pro')

# Define the function to send the user's message to the model and get the response

def get\_model\_response(user\_input):

prompt = "act only as a professional mental health therapist answer my query with your best possible response in short not more than one paragraph, can ask questions like a therapist..."

response = model.generate\_content(user\_input)

return response.text

# Define a function to convert the role of the message to a Streamlit chat message role

def role\_to\_streamlit(role):

return "Therapist" if role == "Bot" else "User"

# Display the chat interface

def main():

st.set\_page\_config(page\_title="Therapisia",

page\_icon='👩‍⚕️',

layout='centered',

initial\_sidebar\_state='collapsed')

st.title(" Therapisia- Mental Health Counselor ChatBot 👩‍⚕️")

st.subheader("Express and feel free to talk with Me")

st.subheader("I'm here to help you :sunglasses:")

# Initialize chat history

if 'chat\_history' not in st.session\_state:

st.session\_state.chat\_history = []

# Display user input box and send button

user\_input = st.text\_input("You: ", key="user\_input")

if st.button("Send") and user\_input:

st.session\_state.chat\_history.append(("User", user\_input))

bot\_response = get\_model\_response(user\_input)

st.session\_state.chat\_history.append(("Bot", bot\_response))

# Display chat history

if st.session\_state.chat\_history:

for role, message in st.session\_state.chat\_history:

message\_html = f"""

<div class="{'user-message' if role == 'User' else 'bot-message'}">

<div class="message">{message}</div>

</div>

"""

st.markdown(message\_html, unsafe\_allow\_html=True)

# Adjust layout

st.markdown("""

<style>

.stTextInput>div>div {

width: 70%;

}

.user-message {

display: flex;

justify-content: flex-end;

margin: 10px;

}

.bot-message {

display: flex;

justify-content: flex-start;

margin: 10px;

}

.message {

max-width: 60%;

padding: 10px;

border-radius: 10px;

background-color: #dcf8c6; /\* User message background \*/

color: #000;

font-size: 16px;

}

.bot-message .message {

background-color: #ececec; /\* Bot message background \*/

}

</style>

""", unsafe\_allow\_html=True)

if \_\_name\_\_ == "\_\_main\_\_":

main()