import smtplib

from datetime import datetime, timedelta

from email.mime.text import MIMEText

import pandas as pd

import random

import re

import os

from difflib import get\_close\_matches

import streamlit as st

# Load the dataset from a local file

csv\_file = "zocdoc.csv"  # Replace with the actual path to your CSV file

if not os.path.exists(csv\_file):

    st.error(f"Dataset file '{csv\_file}' not found! Please ensure the file exists in the correct location.")

    st.stop()

# Load the dataset

df = pd.read\_csv(csv\_file)

# Dictionary to store doctors by specialty

doctors\_by\_specialty = df.groupby('speciality')['Doctor\'s Name'].apply(list).to\_dict()

# Mapping of symptoms to specialties

symptom\_to\_specialty = {

    "fever": "General Physician",

    "cough": "Pulmonologist",

    "chest pain": "Cardiologist",

    "skin rash": "Dermatologist",

    "eye pain": "Ophthalmologist",

    "toothache": "Dentist",

    "joint pain": "Orthopedic",

    "stomach pain": "Gastroenterologist"

}

# Initialize session state for appointments and user appointments

if "appointments" not in st.session\_state:

    st.session\_state["appointments"] = {}

if "user\_appointments" not in st.session\_state:

    st.session\_state["user\_appointments"] = {}

def send\_email(to\_email, subject, body):

    sender\_email = "doctorappointmentchatbot@gmail.com"

    password = "tpvb nebt xkak azog"

    msg = MIMEText(body)

    msg["Subject"] = subject

    msg["From"] = sender\_email

    msg["To"] = to\_email

    try:

        with smtplib.SMTP\_SSL("smtp.gmail.com", 465) as server:

            server.login(sender\_email, password)

            server.sendmail(sender\_email, to\_email, msg.as\_string())

        st.success("Email sent successfully!")

    except Exception as e:

        st.error(f"Error sending email: {e}")

def validate\_mobile(mobile):

    return re.match(r'^\d{10}$', mobile) is not None

def validate\_name(name):

    return re.match(r'^[A-Za-z\s]+$', name) is not None

def parse\_date(date\_str):

    for fmt in ("%Y-%m-%d", "%d-%m-%Y", "%m/%d/%Y", "%d/%m/%Y"):

        try:

            return datetime.strptime(date\_str, fmt).strftime("%Y-%m-%d")

        except ValueError:

            continue

    return None

def parse\_time(time\_str):

    try:

        return datetime.strptime(time\_str, "%I:%M %p").strftime("%I:%M %p")

    except ValueError:

        return None

def correct\_spelling(input\_word, word\_list):

    matches = get\_close\_matches(input\_word, word\_list, n=1, cutoff=0.6)

    return matches[0] if matches else input\_word

def is\_past\_date(date\_str):

    today = datetime.today().strftime("%Y-%m-%d")

    return date\_str < today

def is\_past\_time(date\_str, time\_str):

    if date\_str == datetime.today().strftime("%Y-%m-%d"):

        current\_time = datetime.now().strftime("%I:%M %p")

        return time\_str < current\_time

    return False

def is\_same\_day(date\_str):

    return date\_str == datetime.today().strftime("%Y-%m-%d")

def is\_time\_slot\_available(date\_str, time\_str, doctor):

    if date\_str in st.session\_state["appointments"]:

        for booked\_time, booked\_doctor in st.session\_state["appointments"][date\_str].items():

            if booked\_doctor == doctor:

                booked\_time\_dt = datetime.strptime(booked\_time, "%I:%M %p")

                new\_time\_dt = datetime.strptime(time\_str, "%I:%M %p")

                if abs((new\_time\_dt - booked\_time\_dt).total\_seconds()) < 1200:  # 20 minutes in seconds

                    return False

    return True

def handle\_chat():

    if "step" not in st.session\_state:

        st.session\_state["step"] = None

    if "appointment\_details" not in st.session\_state:

        st.session\_state["appointment\_details"] = {}

    user\_input = st.chat\_input("Type your message here...")

    if user\_input:

        if st.session\_state["step"] is None:

            st.session\_state["step"] = "welcome"

            st.session\_state["messages"] = []

        if st.session\_state["step"] == "welcome":

            st.session\_state["messages"].append({"role": "assistant", "content": "Welcome to the Hospital Appointment System! How can I assist you today?"})

            st.session\_state["messages"].append({"role": "assistant", "content": "Please choose an option by typing the number:\n1. Book Appointment\n2. Reschedule Appointment\n3. Cancel Appointment\n4. Medical Info\n5. Exit"})

            st.session\_state["step"] = "options"

        elif st.session\_state["step"] == "options":

            if user\_input == "1":

                st.session\_state["step"] = "name"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Sure! Let's book an appointment. What's your name?"})

            elif user\_input == "2":

                st.session\_state["step"] = "reschedule\_email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Sure! Let's reschedule your appointment. What's your email address?"})

            elif user\_input == "3":

                st.session\_state["step"] = "cancel\_email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Sure! Let's cancel your appointment. What's your email address?"})

            elif user\_input == "4":

                st.session\_state["step"] = "medical\_info"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Sure! What disease are you looking for information about?"})

            elif user\_input == "5":

                st.session\_state["messages"].append({"role": "assistant", "content": "Thank you for using the Hospital Appointment System. Goodbye!"})

                st.session\_state["step"] = None

            else:

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Invalid option. Please choose a number from 1 to 5."})

        elif st.session\_state["step"] == "name":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "options"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the main menu. Please choose an option by typing the number:\n1. Book Appointment\n2. Reschedule Appointment\n3. Cancel Appointment\n4. Medical Info\n5. Exit"})

            else:

                st.session\_state["appointment\_details"]["name"] = user\_input

                st.session\_state["step"] = "email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Got it! What's your email address?"})

        elif st.session\_state["step"] == "email":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "name"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your name?"})

            else:

                st.session\_state["appointment\_details"]["email"] = user\_input

                st.session\_state["step"] = "mobile"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Great! What's your mobile number?"})

        elif st.session\_state["step"] == "mobile":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your email address?"})

            elif not validate\_mobile(user\_input):

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Invalid mobile number. Please enter a 10-digit number."})

            else:

                st.session\_state["appointment\_details"]["mobile"] = user\_input

                st.session\_state["step"] = "age"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Thanks! What's your age?"})

        elif st.session\_state["step"] == "age":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "mobile"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your mobile number?"})

            elif not user\_input.isdigit():

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Invalid age. Please enter a number."})

            else:

                st.session\_state["appointment\_details"]["age"] = user\_input

                st.session\_state["step"] = "gender"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Got it! What's your gender? (Male/Female/Transgender)"})

        elif st.session\_state["step"] == "gender":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "age"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your age?"})

            elif user\_input.lower() not in ["male", "female", "transgender"]:

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Invalid gender. Please enter Male, Female, or Transgender."})

            else:

                st.session\_state["appointment\_details"]["gender"] = user\_input

                st.session\_state["step"] = "symptoms"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Thanks! What symptoms are you experiencing?"})

        elif st.session\_state["step"] == "symptoms":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "gender"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your gender?"})

            else:

                corrected\_symptoms = correct\_spelling(user\_input, symptom\_to\_specialty.keys())

                if corrected\_symptoms != user\_input:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Did you mean '{corrected\_symptoms}'?"})

                    st.session\_state["appointment\_details"]["symptoms"] = corrected\_symptoms

                else:

                    st.session\_state["appointment\_details"]["symptoms"] = user\_input

                suggested\_specialty = symptom\_to\_specialty.get(st.session\_state["appointment\_details"]["symptoms"], "General Physician")

                st.session\_state["messages"].append({"role": "assistant", "content": f"Based on your symptoms, we recommend a {suggested\_specialty}."})

                if suggested\_specialty not in doctors\_by\_specialty:

                    st.session\_state["messages"].append({"role": "assistant", "content": "No doctors available for the suggested specialty."})

                    st.session\_state["step"] = None

                else:

                    doctors = doctors\_by\_specialty[suggested\_specialty]

                    st.session\_state["appointment\_details"]["doctors"] = doctors

                    st.session\_state["step"] = "select\_doctor"

                    st.session\_state["messages"].append({"role": "assistant", "content": "Here are the available doctors:"})

                    for idx, doctor in enumerate(doctors):

                        st.session\_state["messages"].append({"role": "assistant", "content": f"{idx + 1}. {doctor}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Please select a doctor by typing the number."})

        elif st.session\_state["step"] == "select\_doctor":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "symptoms"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What symptoms are you experiencing?"})

            else:

                try:

                    doctor\_choice = int(user\_input) - 1

                    if doctor\_choice < 0 or doctor\_choice >= len(st.session\_state["appointment\_details"]["doctors"]):

                        st.session\_state["messages"].append({"role": "user", "content": user\_input})

                        st.session\_state["messages"].append({"role": "assistant", "content": "Invalid selection. Please try again."})

                    else:

                        st.session\_state["appointment\_details"]["selected\_doctor"] = st.session\_state["appointment\_details"]["doctors"][doctor\_choice]

                        st.session\_state["step"] = "appointment\_date"

                        st.session\_state["messages"].append({"role": "user", "content": user\_input})

                        st.session\_state["messages"].append({"role": "assistant", "content": "Great! When would you like to book the appointment? (YYYY-MM-DD or DD-MM-YYYY or MM/DD/YYYY or DD/MM/YYYY)"})

                except ValueError:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid input. Please enter a number."})

        elif st.session\_state["step"] == "appointment\_date":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "select\_doctor"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. Please select a doctor by typing the number."})

            else:

                parsed\_date = parse\_date(user\_input)

                if not parsed\_date:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid date format. Please use YYYY-MM-DD, DD-MM-YYYY, MM/DD/YYYY, or DD/MM/YYYY."})

                elif is\_past\_date(parsed\_date):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Cannot book an appointment for a past date!"})

                else:

                    st.session\_state["appointment\_details"]["appointment\_date"] = parsed\_date

                    st.session\_state["step"] = "appointment\_time"

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Got it! What time would you like to book the appointment? (HH:MM AM/PM)"})

        elif st.session\_state["step"] == "appointment\_time":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "appointment\_date"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. When would you like to book the appointment? (YYYY-MM-DD or DD-MM-YYYY or MM/DD/YYYY or DD/MM/YYYY)"})

            else:

                parsed\_time = parse\_time(user\_input)

                if not parsed\_time:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid time format. Please use HH:MM AM/PM."})

                elif is\_past\_time(st.session\_state["appointment\_details"]["appointment\_date"], parsed\_time):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Cannot book an appointment for a past time!"})

                elif not is\_time\_slot\_available(st.session\_state["appointment\_details"]["appointment\_date"], parsed\_time, st.session\_state["appointment\_details"]["selected\_doctor"]):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "This time slot is already booked. Please choose another time."})

                else:

                    st.session\_state["appointment\_details"]["appointment\_time"] = parsed\_time

                    st.session\_state["step"] = "confirm\_appointment"

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Great! Here are your appointment details:"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Name: {st.session\_state['appointment\_details']['name']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Email: {st.session\_state['appointment\_details']['email']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Mobile: {st.session\_state['appointment\_details']['mobile']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Age: {st.session\_state['appointment\_details']['age']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Gender: {st.session\_state['appointment\_details']['gender']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Symptoms: {st.session\_state['appointment\_details']['symptoms']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Doctor: {st.session\_state['appointment\_details']['selected\_doctor']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Date: {st.session\_state['appointment\_details']['appointment\_date']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": f"Time: {st.session\_state['appointment\_details']['appointment\_time']}"})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Type 'confirm' to book the appointment or 'back' to go back."})

        elif st.session\_state["step"] == "confirm\_appointment":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "appointment\_time"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What time would you like to book the appointment? (HH:MM AM/PM)"})

            elif user\_input.lower() == "confirm":

                email = st.session\_state["appointment\_details"]["email"]

                selected\_doctor = st.session\_state["appointment\_details"]["selected\_doctor"]

                parsed\_date = st.session\_state["appointment\_details"]["appointment\_date"]

                appointment\_time = st.session\_state["appointment\_details"]["appointment\_time"]

                if not is\_time\_slot\_available(parsed\_date, appointment\_time, selected\_doctor):

                    st.session\_state["messages"].append({"role": "assistant", "content": "This time slot is already booked. Please choose another time."})

                    st.session\_state["step"] = "appointment\_time"

                else:

                    if parsed\_date not in st.session\_state["appointments"]:

                        st.session\_state["appointments"][parsed\_date] = {}

                    st.session\_state["appointments"][parsed\_date][appointment\_time] = selected\_doctor

                    if email not in st.session\_state["user\_appointments"]:

                        st.session\_state["user\_appointments"][email] = set()

                    st.session\_state["user\_appointments"][email].add((parsed\_date, selected\_doctor))

                    appointment\_id = f"APPT{len(st.session\_state['user\_appointments'])}"

                    user\_body = f"""Hello {st.session\_state['appointment\_details']['name']},

Your appointment with {selected\_doctor} has been successfully booked.

Appointment Date: {parsed\_date}

Appointment Time: {appointment\_time}

Appointment ID: {appointment\_id}

Regards,

Hospital Management

"""

                    send\_email(email, "Appointment Confirmation", user\_body)

                    st.session\_state["messages"].append({"role": "assistant", "content": "Appointment booked successfully! Confirmation email sent."})

                    st.session\_state["step"] = None

            else:

                st.session\_state["messages"].append({"role": "assistant", "content": "Invalid input. Type 'confirm' to book the appointment or 'back' to go back."})

        elif st.session\_state["step"] == "reschedule\_email":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "options"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the main menu. Please choose an option by typing the number:\n1. Book Appointment\n2. Reschedule Appointment\n3. Cancel Appointment\n4. Medical Info\n5. Exit"})

            elif user\_input not in st.session\_state["user\_appointments"] or not st.session\_state["user\_appointments"][user\_input]:

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "No existing appointments found for this email. Please try again."})

            else:

                st.session\_state["appointment\_details"]["email"] = user\_input

                st.session\_state["step"] = "reschedule\_select"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Here are your current appointments:"})

                appointment\_list = list(st.session\_state["user\_appointments"][user\_input])

                for idx, (date, doctor) in enumerate(appointment\_list):

                    st.session\_state["messages"].append({"role": "assistant", "content": f"{idx + 1}. {doctor} on {date}"})

                st.session\_state["messages"].append({"role": "assistant", "content": "Please select an appointment to reschedule by typing the number."})

        elif st.session\_state["step"] == "reschedule\_select":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "reschedule\_email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your email address?"})

            else:

                try:

                    choice = int(user\_input) - 1

                    appointment\_list = list(st.session\_state["user\_appointments"][st.session\_state["appointment\_details"]["email"]])

                    if choice < 0 or choice >= len(appointment\_list):

                        st.session\_state["messages"].append({"role": "user", "content": user\_input})

                        st.session\_state["messages"].append({"role": "assistant", "content": "Invalid selection. Please try again."})

                    else:

                        st.session\_state["appointment\_details"]["old\_date"], st.session\_state["appointment\_details"]["doctor"] = appointment\_list[choice]

                        st.session\_state["step"] = "reschedule\_date"

                        st.session\_state["messages"].append({"role": "user", "content": user\_input})

                        st.session\_state["messages"].append({"role": "assistant", "content": "Got it! What's the new appointment date? (YYYY-MM-DD or DD-MM-YYYY or MM/DD/YYYY or DD/MM/YYYY)"})

                except ValueError:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid input. Please enter a number."})

        elif st.session\_state["step"] == "reschedule\_date":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "reschedule\_select"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. Please select an appointment to reschedule by typing the number."})

            else:

                parsed\_new\_date = parse\_date(user\_input)

                if not parsed\_new\_date:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid date format. Please use YYYY-MM-DD, DD-MM-YYYY, MM/DD/YYYY, or DD/MM/YYYY."})

                elif is\_past\_date(parsed\_new\_date):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Cannot reschedule to a past date!"})

                else:

                    st.session\_state["appointment\_details"]["new\_date"] = parsed\_new\_date

                    st.session\_state["step"] = "reschedule\_time"

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Great! What's the new appointment time? (HH:MM AM/PM)"})

        elif st.session\_state["step"] == "reschedule\_time":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "reschedule\_date"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's the new appointment date? (YYYY-MM-DD or DD-MM-YYYY or MM/DD/YYYY or DD/MM/YYYY)"})

            else:

                parsed\_new\_time = parse\_time(user\_input)

                if not parsed\_new\_time:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid time format. Please use HH:MM AM/PM."})

                elif is\_past\_time(st.session\_state["appointment\_details"]["new\_date"], parsed\_new\_time):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Cannot reschedule to a past time!"})

                elif not is\_time\_slot\_available(st.session\_state["appointment\_details"]["new\_date"], parsed\_new\_time, st.session\_state["appointment\_details"]["doctor"]):

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "This time slot is already booked. Please choose another time."})

                else:

                    old\_time = None

                    for time, doc in st.session\_state["appointments"].get(st.session\_state["appointment\_details"]["old\_date"], {}).items():

                        if doc == st.session\_state["appointment\_details"]["doctor"]:

                            old\_time = time

                            break

                    st.session\_state["appointments"][st.session\_state["appointment\_details"]["old\_date"]].pop(old\_time, None)

                    st.session\_state["appointments"].setdefault(st.session\_state["appointment\_details"]["new\_date"], {})[parsed\_new\_time] = st.session\_state["appointment\_details"]["doctor"]

                    st.session\_state["user\_appointments"][st.session\_state["appointment\_details"]["email"]].remove((st.session\_state["appointment\_details"]["old\_date"], st.session\_state["appointment\_details"]["doctor"]))

                    st.session\_state["user\_appointments"][st.session\_state["appointment\_details"]["email"]].add((st.session\_state["appointment\_details"]["new\_date"], st.session\_state["appointment\_details"]["doctor"]))

                    st.session\_state["messages"].append({"role": "assistant", "content": "Appointment rescheduled successfully! Confirmation email sent."})

                    reschedule\_body = f"""Hello,

Your appointment with {st.session\_state['appointment\_details']['doctor']} has been rescheduled.

New Appointment Date: {st.session\_state['appointment\_details']['new\_date']}

New Appointment Time: {parsed\_new\_time}

Regards,

Hospital Management

"""

                    send\_email(st.session\_state["appointment\_details"]["email"], "Appointment Rescheduled", reschedule\_body)

                    st.session\_state["step"] = None

        elif st.session\_state["step"] == "cancel\_email":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "options"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the main menu. Please choose an option by typing the number:\n1. Book Appointment\n2. Reschedule Appointment\n3. Cancel Appointment\n4. Medical Info\n5. Exit"})

            elif user\_input not in st.session\_state["user\_appointments"] or not st.session\_state["user\_appointments"][user\_input]:

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "No existing appointments found for this email. Please try again."})

            else:

                st.session\_state["appointment\_details"]["email"] = user\_input

                st.session\_state["step"] = "cancel\_select"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Here are your current appointments:"})

                appointment\_list = list(st.session\_state["user\_appointments"][user\_input])

                for idx, (date, doctor) in enumerate(appointment\_list):

                    st.session\_state["messages"].append({"role": "assistant", "content": f"{idx + 1}. {doctor} on {date}"})

                st.session\_state["messages"].append({"role": "assistant", "content": "Please select an appointment to cancel by typing the number."})

        elif st.session\_state["step"] == "cancel\_select":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "cancel\_email"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the previous step. What's your email address?"})

            else:

                try:

                    choice = int(user\_input) - 1

                    appointment\_list = list(st.session\_state["user\_appointments"][st.session\_state["appointment\_details"]["email"]])

                    if choice < 0 or choice >= len(appointment\_list):

                        st.session\_state["messages"].append({"role": "user", "content": user\_input})

                        st.session\_state["messages"].append({"role": "assistant", "content": "Invalid selection. Please try again."})

                    else:

                        cancel\_date, doctor = appointment\_list[choice]

                        for time, doc in st.session\_state["appointments"].get(cancel\_date, {}).items():

                            if doc == doctor:

                                del st.session\_state["appointments"][cancel\_date][time]

                                break

                        st.session\_state["user\_appointments"][st.session\_state["appointment\_details"]["email"]].remove((cancel\_date, doctor))

                        st.session\_state["messages"].append({"role": "assistant", "content": "Appointment canceled successfully! Confirmation email sent."})

                        cancel\_body = f"""Hello,

Your appointment with {doctor} on {cancel\_date} has been successfully canceled.

Regards,

Hospital Management

"""

                        send\_email(st.session\_state["appointment\_details"]["email"], "Appointment Cancellation", cancel\_body)

                        st.session\_state["step"] = None

                except ValueError:

                    st.session\_state["messages"].append({"role": "user", "content": user\_input})

                    st.session\_state["messages"].append({"role": "assistant", "content": "Invalid input. Please enter a number."})

        elif st.session\_state["step"] == "medical\_info":

            if user\_input.lower() == "back":

                st.session\_state["step"] = "options"

                st.session\_state["messages"].append({"role": "user", "content": user\_input})

                st.session\_state["messages"].append({"role": "assistant", "content": "Returning to the main menu. Please choose an option by typing the number:\n1. Book Appointment\n2. Reschedule Appointment\n3. Cancel Appointment\n4. Medical Info\n5. Exit"})

            else:

                disease = user\_input

                today = datetime.today().strftime("%Y-%m-%d")

                simulated\_count = random.randint(100, 5000)

                st.session\_state["messages"].append({"role": "assistant", "content": f"As of {today}, approximately {simulated\_count} patients are affected by {disease} in India."})

                st.session\_state["step"] = None

    # Display chat messages

    for message in st.session\_state.get("messages", []):

        with st.chat\_message(message["role"]):

            st.write(message["content"])

def main():

    st.title("Hospital Appointment Chatbot")

    handle\_chat()

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

    main()