import tkinter as tk

from tkinter import scrolledtext

import random

import sqlite3

# Database function to set up and populate the doctor information if not present

def setup\_database():

conn = sqlite3.connect('healthcare.db')

cursor = conn.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS doctors (

id INTEGER PRIMARY KEY,

name TEXT,

specialty TEXT,

location TEXT,

contact TEXT

)

''')

# Insert sample data if table is empty

cursor.execute("SELECT COUNT(\*) FROM doctors")

if cursor.fetchone()[0] == 0:

doctors\_data = [

('Dr. Alice Johnson', 'Cardiologist', 'New York', 'alice@example.com'),

('Dr. Bob Smith', 'Pediatrician', 'Los Angeles', 'bob@example.com'),

('Dr. Carol Lee', 'Dermatologist', 'Chicago', 'carol@example.com'),

]

cursor.executemany('INSERT INTO doctors (name, specialty, location, contact) VALUES (?, ?, ?, ?)', doctors\_data)

conn.commit()

conn.close()

# Call the setup function to initialize the database

setup\_database()

# Function to fetch doctor information from the database

def get\_doctor\_info(specialty=None, location=None):

conn = sqlite3.connect('healthcare.db')

cursor = conn.cursor()

# Build query dynamically based on parameters

query = "SELECT name, specialty, location, contact FROM doctors WHERE 1=1"

params = []

if specialty:

query += " AND specialty LIKE ?"

params.append(f"%{specialty}%")

if location:

query += " AND location LIKE ?"

params.append(f"%{location}%")

cursor.execute(query, params)

doctors = cursor.fetchall()

conn.close()

# Format the response

if doctors:

response = "Here are some doctors that might help:\n"

for doc in doctors:

response += f"Name: {doc[0]}, Specialty: {doc[1]}, Location: {doc[2]}, Contact: {doc[3]}\n"

return response

else:

return "I'm sorry, I couldn't find any doctors matching your criteria."

# Main healthcare response logic

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

user\_input = user\_input.lower()

# Check for doctor inquiries

if "doctor" in user\_input:

specialty = None

location = None

# Identify specialty and location from the user input

if "cardiologist" in user\_input:

specialty = "Cardiologist"

elif "pediatrician" in user\_input:

specialty = "Pediatrician"

elif "dermatologist" in user\_input:

specialty = "Dermatologist"

if "new york" in user\_input:

location = "New York"

elif "los angeles" in user\_input:

location = "Los Angeles"

elif "chicago" in user\_input:

location = "Chicago"

# Get doctor information based on the specialty and location

return get\_doctor\_info(specialty, location)

# Basic hardcoded responses for symptoms

elif "fever" in user\_input:

return "If you're uncomfortable, you can take over-the-counter medications like acetaminophen or ibuprofen."

elif "headache" in user\_input:

return "Headaches can be caused by stress or dehydration. Over-the-counter pain relievers like acetaminophen, ibuprofen, or aspirin can help relieve pain."

elif "cough" in user\_input:

return "Adding a pinch of turmeric powder to warm milk or water can help control mucus production and soothe a sore throat."

elif "cold" in user\_input or "sneeze" in user\_input:

return "Gargle with warm salt water, or try a hot lemon and honey drink. Decongestants, antihistamines, and pain relievers can help."

elif "tired" in user\_input or "fatigue" in user\_input:

return "Fatigue could be due to lack of sleep or stress. Make sure to get enough rest and consult a doctor if it persists."

elif "heart attack" in user\_input:

return "Call 911 or your local emergency number. Take aspirin, if recommended, and follow emergency procedures."

else:

return random.choice([

"I'm sorry, I didn't understand that. Can you describe your symptoms in another way?",

"Could you please provide more details about your symptoms?",

"I'm here to help. Can you tell me more about what you're experiencing?"

])

# Function to handle user input

def send\_message():

user\_input = user\_entry.get()

if user\_input.strip() == "":

return

# Display user message

chat\_display.config(state=tk.NORMAL)

chat\_display.insert(tk.END, f"You: {user\_input}\n")

chat\_display.config(state=tk.DISABLED)

user\_entry.delete(0, tk.END)

# Get chatbot response

response = get\_healthcare\_response(user\_input)

# Display chatbot response

chat\_display.config(state=tk.NORMAL)

chat\_display.insert(tk.END, f"Bot: {response}\n")

chat\_display.config(state=tk.DISABLED)

# Scroll to the end of chat display

chat\_display.yview(tk.END)

# Initialize GUI

root = tk.Tk()

root.title("Healthcare Chatbot")

# Chat display area

chat\_display = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=50, height=20, font=("Arial", 12))

chat\_display.grid(row=0, column=0, columnspan=2, padx=10, pady=10)

chat\_display.config(state=tk.DISABLED) # Start with display disabled

# User entry area

user\_entry = tk.Entry(root, width=40, font=("Arial", 12))

user\_entry.grid(row=1, column=0, padx=10, pady=10)

# Send button

send\_button = tk.Button(root, text="Send", command=send\_message, font=("Arial", 12))

send\_button.grid(row=1, column=1, padx=10, pady=10)

# Enable chat display area to accept new messages

chat\_display.config(state=tk.NORMAL)

chat\_display.insert(tk.END, "Bot: Hello! I am a healthcare assistant. How can I help you today?\n")

chat\_display.config(state=tk.DISABLED)

# Run the application

root.mainloop()