import openai

import schedule

import time

import datetime

# ========== Configuration ==========

openai.api\_key = "YOUR\_OPENAI\_API\_KEY"

# Simulated patient data

patient\_data = {

"name": "Alice",

"age": 30,

"symptoms": [],

"medications": {

"Metformin": "08:00",

"Vitamin D": "20:00"

}

}

# ========== AI Chat Function ==========

def ask\_health\_ai(question):

prompt = f"You are a helpful and cautious healthcare assistant. A patient asks: '{question}'. Provide a simple, informative, and safe response."

response = openai.ChatCompletion.create(

model="gpt-4",

messages=[

{"role": "system", "content": "You are a medical assistant, not a doctor."},

{"role": "user", "content": prompt}

],

temperature=0.4,

max\_tokens=300

)

return response.choices[0].message.content.strip()

# ========== Symptom Checker ==========

def basic\_symptom\_checker(symptoms):

known\_conditions = {

"fever": "You may have an infection. Stay hydrated and monitor your temperature.",

"headache": "It could be due to stress, dehydration, or something else. Rest and hydrate.",

"cough": "This could indicate a cold, flu, or allergies. If persistent, see a doctor.",

"chest pain": "Seek medical attention immediately.",

"fatigue": "Ensure adequate rest, nutrition, and hydration. It could be stress-related."

}

advice = []

for symptom in symptoms:

matched = known\_conditions.get(symptom.lower(), "Symptom not recognized. Please consult a doctor.")

advice.append(f"- {symptom.capitalize()}: {matched}")

return "\n".join(advice)

# ========== Medication Reminder ==========

def medication\_reminder():

now = datetime.datetime.now().strftime("%H:%M")

for med, time\_str in patient\_data["medications"].items():

if now == time\_str:

print(f"💊 Reminder: Time to take your medication: {med}")

# ========== Chatbot Interface ==========

def chatbot():

print(f"👩‍⚕️ Hello {patient\_data['name']}! I'm your Health AI Assistant.")

print("Type 'exit' to end.\n")

while True:

user\_input = input("You: ")

if user\_input.lower() in ["exit", "quit"]:

print("👋 Take care! Remember to follow your treatment plan.")

break

elif "symptom" in user\_input.lower():

symptoms = input("Enter symptoms separated by commas: ").split(",")

symptoms = [s.strip() for s in symptoms]

result = basic\_symptom\_checker(symptoms)

print(f"🩺 Symptom Analysis:\n{result}")

else:

reply = ask\_health\_ai(user\_input)

print(f"Health AI: {reply}")

# ========== Scheduler for Medication Alerts ==========

def start\_scheduler():

for med, time\_str in patient\_data["medications"].items():

schedule.every().day.at(time\_str).do(lambda: print(f"💊 Reminder: Take {med} now!"))

while True:

schedule.run\_pending()

time.sleep(60) # Check every minute

# ========== Run Program ==========

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

# Option: background medication reminder

# import threading

# threading.Thread(target=start\_scheduler).start()

chatbot()