```
import speech_recognition as sr

from gtts import gTTS

import playsound

import os

import uuid

import datetime

import random

from geopy.geocoders import Nominatim

import geocoder


recognizer = sr.Recognizer()


# ---------------- MEMORY STORAGE ----------------

saved_memories = []


# ---------------- WELLNESS TIPS ----------------

wellness_tips = [

    "Please take slow deep breaths. You are safe.",

    "Try naming five things you can see around you.",

    "Listening to familiar music can help you feel calm.",

    "A short walk can refresh your mind.",

    "You are doing well. Take things one step at a time."

]


# ---------------- ALZHEIMER’S MEDICINE SCHEDULE ----------------

medicine_schedule = {

    "morning": [

        {

            "name": "Donepezil",

            "instruction": "Take one tablet after breakfast",

            "purpose": "Helps improve memory and thinking ability"
        },
        {
            "name": "Vitamin B12",
            "instruction": "Take one tablet after breakfast",
            "purpose": "Supports brain and nerve health"
        }
    ],
    "afternoon": [
        {
            "name": "Memantine",
            "instruction": "Take one tablet after lunch",
            "purpose": "Reduces confusion and supports daily functioning"
        }
    ],
    "night": [
        {
            "name": "Memantine",
            "instruction": "Take one tablet after dinner",
            "purpose": "Helps with memory and evening behavior"
        },
        {
            "name": "Calcium",
            "instruction": "Take one tablet before sleep",
            "purpose": "Strengthens bones and prevents weakness"
        }
    ]
}

# ---------------- TEXT TO SPEECH ----------------
def speak(text):
    filename = f"{uuid.uuid4()}.mp3"
    tts = gTTS(text=text, lang='en')
    tts.save(filename)
    playsound.playsound(filename)
    os.remove(filename)

# ---------------- SPEECH TO TEXT ----------------
def listen():
    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source, duration=0.4)
        audio = recognizer.listen(source)
        try:
            return recognizer.recognize_google(audio).lower()
        except:
            speak("Sorry, I did not understand. Please say it again slowly.")
            return None

# ---------------- READ MEDICINES ----------------
def read_medicines(time_of_day):
    meds = medicine_schedule[time_of_day]
    for med in meds:
        speak(med["name"])
        speak(med["instruction"])
        speak(med["purpose"])
    speak("Please take medicines only as prescribed by your doctor.")

# ---------------- LOCATION FUNCTIONS ----------------
def get_full_address():
    try:
        # Get approximate coordinates using IP
        g = geocoder.ip('me')
        lat, lng = g.latlng
        # Reverse geocode using Nominatim
        geolocator = Nominatim(user_agent="mindtrace_assistant")
        location = geolocator.reverse((lat, lng), language='en')
        address = location.address if location else "Address not found"

        # Google Maps link (for caregiver / logging only)
        maps_link = f"https://www.google.com/maps/search/?api=1&query={lat},{lng}"
        return address, maps_link
    except Exception as e:
        print("Error getting location:", e)
        return "Unable to determine your location", None

# ---------------- MAIN ASSISTANT ----------------
def assistant():
    speak("Hello. I am your MindTrace Care assistant. How may I help you today?")

    while True:
        command = listen()
        if not command:
            continue

        # TIME
        if "time" in command:
            speak("The current time is " +
                  datetime.datetime.now().strftime("%I:%M %p"))

        # DATE
        elif "date" in command:
            speak("Today is " +
                  datetime.datetime.now().strftime("%B %d, %Y"))
```

```
# SAVE MEMORY
elif "save memory" in command or "remember this" in command:
    speak("Please tell me what I should remember.")
    memory = listen()
    if memory:
        saved_memories.append(memory)
        speak("I have saved it.")

# RECALL MEMORY
elif "recall memory" in command or "show memory" in command:
    if saved_memories:
        speak("Here are your memories.")
        for m in saved_memories:
            speak(m)
    else:
        speak("You do not have any saved memories.")

# MORNING MEDICINE
elif "morning medicine" in command:
    speak("It is time for your morning medicines.")
    read_medicines("morning")

# AFTERNOON / NOON MEDICINE
elif "afternoon medicine" in command or "noon medicine" in command:
    speak("It is time for your afternoon medicines.")
    read_medicines("afternoon")

# NIGHT MEDICINE
elif "night medicine" in command:
    speak("It is time for your night medicines.")
    read_medicines("night")

# ALL MEDICINES
elif "all medicines" in command:
    speak("Here is your complete medicine schedule.")
    speak("Morning medicines.")
    read_medicines("morning")
    speak("Afternoon medicines.")
    read_medicines("afternoon")
    speak("Night medicines.")
    read_medicines("night")

# WHAT MEDICINE NOW
elif "what medicine should i take now" in command:
    hour = datetime.datetime.now().hour
    if 5 <= hour < 12:
        speak("It is morning time.")
        read_medicines("morning")
    elif 12 <= hour < 18:
        speak("It is afternoon time.")
        read_medicines("afternoon")
    else:
        speak("It is night time.")
        read_medicines("night")

# WELLNESS HELP
elif "help me feel better" in command or "tip" in command:
    speak(random.choice(wellness_tips))

# GREETING
elif "hello" in command or "hi" in command:
```

```
            speak("Hello. I am here to help you.")

        # LOCATION FOR PATIENT (speaks only full address)
        elif "where am i" in command or "my location" in command:
            address, maps_link = get_full_address()
            speak(f"You are currently at: {address}")
            # Do not speak maps link aloud, but print for caregiver/logging
            if maps_link:
                print(f"Caregiver Google Maps link: {maps_link}")

        # EMERGENCY
        elif "emergency" in command:
            speak("Please stay calm. I am alerting your caregiver.")
            address, maps_link = get_full_address()
            speak(f"You are currently at: {address}")
            # Print the link only for caregiver
            if maps_link:
                print(f"Caregiver Google Maps link: {maps_link}")

        # EXIT
        elif "exit" in command or "stop" in command or "quit" in command:
            speak("Goodbye. Take care.")
            break

        # UNKNOWN
        else:
            speak("Please say that again slowly.")

# ---------------- RUN ASSISTANT ----------------
assistant()
```