import time

import random

from datetime import datetime, timedelta

# Sample reminders data for health clock

reminders = {

'medication': '08:00 AM',

'meal': '12:00 PM',

'appointment': '03:00 PM',

}

# Sample health data (In a real scenario, this would come from sensors)

def get\_health\_data():

heart\_rate = random.randint(60, 100) # Heart rate in beats per minute

activity\_level = random.choice(['Low', 'Medium', 'High']) # Activity level

sleep\_hours = random.randint(4, 8) # Sleep hours

return heart\_rate, activity\_level, sleep\_hours

# Function to send alerts (in a real system, you would integrate notifications)

def send\_alert(message):

print(f"ALERT: {message}")

# AI-based health monitoring and prediction (Simplified)

def ai\_health\_check(heart\_rate, activity\_level):

# Basic prediction: if heart rate is too high or activity level is low, send an alert

if heart\_rate > 90:

send\_alert("High heart rate detected. Please check your health!")

elif activity\_level == 'Low':

send\_alert("Low activity detected. Consider moving around for your health.")

# Function to display reminders

def show\_reminders():

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

print(f"\nCurrent Time: {current\_time}")

for reminder, time in reminders.items():

if current\_time == time:

print(f"Reminder: Time for {reminder.capitalize()}")

# Health clock main loop (simplified for demo)

def health\_clock():

while True:

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

# Show reminders based on the time

show\_reminders()

# Get health data and perform AI health check

heart\_rate, activity\_level, sleep\_hours = get\_health\_data()

print(f"Heart Rate: {heart\_rate} bpm | Activity Level: {activity\_level} | Sleep Hours: {sleep\_hours} hours")

# AI-driven health check

ai\_health\_check(heart\_rate, activity\_level)

# Wait before checking again (simulating real-time updates every minute)

time.sleep(60)

# Run the health clock

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

print("Health Clock is running...")

health\_clock()