# AI-Powered Healthcare Assistant and Supply Chain Assistant

# ----------------------------------

# Healthcare Assistant Section

# ----------------------------------

# Sample medical dataset (can be expanded)

medical\_data = {

"fever": "You may have a viral infection. Stay hydrated and rest. If symptoms persist, consult a doctor.",

"cough": "You might have a respiratory infection. Drink warm fluids and consider a medical checkup.",

"headache": "Rest in a dark room and stay hydrated. Seek medical advice if the headache is severe or persistent.",

"cold": "It could be a common cold. Rest, drink fluids, and consider over-the-counter remedies.",

"sore throat": "Gargle with warm salt water. If it continues, consult a physician.",

"fatigue": "Ensure you're getting enough sleep and nutrients. Persistent fatigue should be evaluated by a doctor."

}

# Simple NLP-based symptom checker

def analyze\_symptoms(user\_input):

user\_input = user\_input.lower()

responses = []

for symptom, advice in medical\_data.items():

if symptom in user\_input:

responses.append(f"{symptom.title()}: {advice}")

if not responses:

responses.append("Symptoms not recognized. Please consult a medical professional.")

return responses

# Healthcare Assistant Interaction

def healthcare\_assistant():

print("AI Healthcare Assistant\nType your symptoms below (e.g., 'I have a headache and cough'):")

user\_input = input("You: ")

responses = analyze\_symptoms(user\_input)

print("\nAI Assistant:")

for response in responses:

print(response)

# ----------------------------------

# Supply Chain Assistant Section

# ----------------------------------

# Sample demand forecast for each month (can be expanded)

demand\_data = {

1: 120, 2: 130, 3: 140, 4: 150, 5: 292, 6: 160,

7: 170, 8: 180, 9: 190, 10: 200, 11: 210, 12: 220

}

# Sample sensor data (for 'iot' command)

sensor\_data = {

'shipment\_id': 'SHIP123',

'temperature': 22.57,

'humidity': 53.91,

'location': 'Warehouse A'

}

def get\_demand\_for\_month(month):

# Returns the estimated demand for the given month

if 1 <= month <= 12:

return demand\_data.get(month, "No data for this month.")

else:

return "Invalid month. Enter number between 1 and 12."

def get\_sensor\_data():

# Returns the current sensor data for shipment

return sensor\_data

# Supply Chain Assistant Interaction

def supply\_chain\_assistant():

print("Supply Chain Assistant is ready.")

while True:

user\_input = input("Enter month (1-12), 'iot' for sensor data, 'exit': ")

if user\_input.isdigit():

month = int(user\_input)

print(f"Estimated demand for month {month}: {get\_demand\_for\_month(month)}")

elif user\_input.lower() == 'iot':

print("Current Sensor Data:")

for key, value in get\_sensor\_data().items():

print(f"{key.title()}: {value}")

elif user\_input.lower() == 'exit':

print("Exiting Supply Chain Assistant.")

break

else:

print("Please enter a valid input.")

# ----------------------------------

# Main Program: Choose Assistant

# ----------------------------------

def main():

print("Welcome to the AI Assistant Suite.")

print("Type '1' for Healthcare Assistant or '2' for Supply Chain Assistant.")

choice = input("Enter choice: ")

if choice == '1':

healthcare\_assistant()

elif choice == '2':

supply\_chain\_assistant()

else:

print("Invalid choice. Please restart the program.")

# Run the main function

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

main()