# Define an expanded product catalog with subcategories

product\_catalog = [

{"name": "Football", "category": "Sport\_Balls", "price": 20, "description": "A standard football for outdoor play"},

{"name": "Basketball", "category": "Sport\_Balls", "price": 25, "description": "An official size basketball"},

{"name": "Volleyball", "category": "Sport\_Balls", "price": 18, "description": "Durable volleyball for indoor and outdoor play"},

{"name": "Soccer", "category": "Sport\_Balls", "price": 22, "description": "High-quality soccer ball for all surfaces"},

{"name": "Tennis Racket", "category": "Sports\_Equipment", "price": 50, "description": "Lightweight racket for all levels"},

{"name": "Running Shoes", "category": "Sports\_Equipment", "price": 80, "description": "Comfortable shoes for long-distance running"},

{"name": "Golf Set", "category": "Sports\_Equipment", "price": 300, "description": "Complete set of golf clubs and bag"},

{"name": "Smartwatch", "category": "Gadgets", "price": 150, "description": "Tracks fitness and sports activities"},

{"name": "Yoga Mat", "category": "Fitness", "price": 30, "description": "Non-slip mat for yoga and stretching exercises"},

{"name": "Skipping Rope", "category": "Fitness", "price": 10, "description": "Adjustable speed skipping rope for cardio"},

{"name": "Hula Hoop", "category": "Fitness", "price": 15, "description": "Weighted hula hoop for core workouts"},

{"name": "Fitbit", "category": "Gadgets", "price": 120, "description": "Wearable fitness tracker for monitoring health"}

]

# Function to recommend products based on context (user's intent)

def recommend\_products(intent):

recommendations = []

for product in product\_catalog:

if intent.lower() in product["category"].lower() or intent.lower() in product["description"].lower():

recommendations.append(product)

return recommendations

# Function for an interactive session to refine recommendations

def refine\_recommendations(recommendations):

print("\nWe have found the following products:")

for idx, product in enumerate(recommendations, 1):

print(f"{idx}. {product['name']} - ${product['price']}: {product['description']}")

while True:

user\_input = input("\nWould you like to filter the results by price (yes/no)? ").strip().lower()

if user\_input == "yes":

max\_price = float(input("Enter your maximum budget: $"))

recommendations = [p for p in recommendations if p["price"] <= max\_price]

if not recommendations:

print("No products found within your budget. Try increasing your budget.")

else:

print("\nFiltered products:")

for idx, product in enumerate(recommendations, 1):

print(f"{idx}. {product['name']} - ${product['price']}: {product['description']}")

else:

break

return recommendations

# Main function to simulate the GenAI recommender system

def personalized\_recommendation():

print("Welcome to the GenAI-powered product recommender!")

user\_name = input("What's your name? ").strip()

gift\_receiver = input("Who are you buying a gift for? ").strip()

interest = input(f"What does {gift\_receiver} like (e.g., sports, fitness, gadgets)? ").strip()

print(f"\nThank you, {user\_name}. Let's find the perfect gift for {gift\_receiver}.")

recommendations = recommend\_products(interest)

if not recommendations:

print("Sorry, we couldn't find any products matching that interest. Please try a different search.")

return

final\_recommendations = refine\_recommendations(recommendations)

print("\nFinal Recommendations:")

for idx, product in enumerate(final\_recommendations, 1):

print(f"{idx}. {product['name']} - ${product['price']}: {product['description']}")

print("\nThank you for using the GenAI recommender. Happy shopping!")

# Run the personalized recommender system

personalized\_recommendation()