SUPERMARKET BILL GENERATION

Project Report: Supermarket Bill Generation Mini-Project

1. Introduction:

The Supermarket Bill Generation Mini-Project is a simple Python application that simulates the operations of a supermarket. It allows users to add items to a virtual inventory, make purchases by adding items to a cart, remove items from the cart, and generate bills for the purchases made.

2. Features:

- Inventory Management: The application allows users to add items to the inventory along with their prices and quantities.

- Purchase Management: Users can add items to their shopping cart, remove items from the cart, and view the items currently in the cart.

- Bill Generation: Users can generate a bill for the items in their cart, which includes the item names, quantities, prices, and the total bill amount.

- User Interface: The application provides a simple command-line interface for interaction, with options displayed for each step of the process.

3. Implementation:

- Supermarket Class: Defines the structure and methods for managing the supermarket operations, including adding items to the inventory, removing items, and generating bills.

CODE:  
 class Supermarket:

def \_\_init\_\_(self):

self.inventory = {}

def add\_item(self, item\_name, price, quantity):

try:

if item\_name in self.inventory:

self.inventory[item\_name]['price'] = price

self.inventory[item\_name]['quantity'] += quantity

else:

self.inventory[item\_name] = {'price': price, 'quantity': quantity}

except Exception as e:

print("Error:", "invalid product please enter correct item")

def remove\_item(self, item\_name, quantity):

try:

if item\_name in self.inventory:

if self.inventory[item\_name]['quantity'] >= quantity:

self.inventory[item\_name]['quantity'] -= quantity

else:

print("Insufficient quantity of", item\_name)

else:

print(item\_name, "not found in inventory")

except Exception as e:

print("Error:", "invalid product please enter correct")

def generate\_bill(self, cart):

total\_bill = 0

print("\*\*\*\* Your Bill \*\*\*\*")

try:

for item, quantity in cart.items():

if item in self.inventory:

item\_price = self.inventory[item]['price']

item\_total = item\_price \* quantity

print(item, "x", quantity, "=", item\_total)

total\_bill += item\_total

else:

print(item, "not found in inventory")

print("Total Bill:", total\_bill)

except Exception as e:

print("Error:", "invalid product please enter correct")

def main():

try:

supermarket = Supermarket()

# Adding items to inventory

supermarket.add\_item("Apple", 2.5, 100)

supermarket.add\_item("Banana", 1.0, 150)

supermarket.add\_item("Milk", 3.0, 50)

supermarket.add\_item("Bread", 2.0, 75)

cart = {}

while True:

print("\nAvailable Items:")

for item, details in supermarket.inventory.items():

print(item, "-", details['price'], "per item")

print("\n1. Add Item to Cart")

print("2. Remove Item from Cart")

print("3. Generate Bill")

print("4. Exit")

choice = input("Enter your choice: ")

if choice == '1':

item = input("Enter item name: ")

quantity = int(input("Enter quantity: "))

if item in supermarket.inventory:

if item in cart:

cart[item] += quantity

else:

cart[item] = quantity

else:

print("Item not found in inventory")

elif choice == '2':

item = input("Enter item name: ")

quantity = int(input("Enter quantity: "))

if item in cart:

if quantity >= cart[item]:

del cart[item]

else:

cart[item] -= quantity

else:

print("Item not found in cart")

elif choice == '3':

if cart:

supermarket.generate\_bill(cart)

break

else:

print("Cart is empty")

elif choice == '4':

break

else:

print("Invalid choice")

except Exception as e:

print("Error:", "invalid product please enter correct")

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

main()

OUTPUT:

I).Available Items:

Apple - 25 per item

Banana - 10 per item

Milk - 30 per item

Bread - 20 per item

1. Add Item to Cart

2. Remove Item from Cart

3. Generate Bill

4. Exit

Enter your choice: 1

Enter item name: Apple

Enter quantity: 4

Available Items:

Apple - 25 per item

Banana - 10 per item

Milk - 30 per item

Bread - 20 per item

1. Add Item to Cart

2. Remove Item from Cart

3. Generate Bill

4. Exit

Enter your choice: 1

Enter item name: Bread

Enter quantity: 3

Available Items:

Apple - 25 per item

Banana - 10 per item

Milk - 30 per item

Bread - 20 per item

1. Add Item to Cart

2. Remove Item from Cart

3. Generate Bill

4. Exit

Enter your choice: 3

\*\*\*\* Your Bill \*\*\*\*

Apple x 4 = 100

Bread x 3 = 60

Total Bill: 160

II).Available Items:

Apple - 25 per item

Banana - 10 per item

Milk - 30 per item

Bread - 20 per item

1. Add Item to Cart

2. Remove Item from Cart

3. Generate Bill

4. Exit

Enter your choice: 1

Enter item name: Apple

Enter quantity: hg

---------------------------------------------------------------------------

ValueError Traceback (most recent call last)

Cell In[1], line 89

85 print("Invalid choice")

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

---> 89 main()

Cell In[1], line 58, in main()

56 if choice == '1':

57 item = input("Enter item name: ")

---> 58 quantity = int(input("Enter quantity: "))

59 if item in supermarket.inventory:

60 if item in cart:

ValueError: invalid literal for int() with base 10: 'hg'

III).Available Items:

Apple - 2.5 per item

Banana - 1.0 per item

Milk - 3.0 per item

Bread - 2.0 per item

1. Add Item to Cart

2. Remove Item from Cart

3. Generate Bill

4. Exit

Enter your choice: 1

Enter item name: Banana

Enter quantity: f

Error: invalid product please enter correct

3. Conclusion:

The Supermarket Bill Generation Mini-Project provides a basic yet functional implementation of supermarket operations using Python. It serves as a foundation for further exploration and extension into more complex inventory management and retail systems. This project demonstrates the versatility and power of Python in developing practical applications for real-world scenarios.

PROJECT DONE BY

Y.KARTHIK

B.KARTHEEK

K.MARUTHI PRASANNA

K.GOPI CHANDU