

 **Simple CLI Supermarket Cart with Remove Option:**

python

CopyEdit

```
from datetime import datetime
```

```
# Welcome message
```

```
print("\n----- Welcome to Malar Supermarket -----\\n")
```

```
name = input("Enter your name: ")
```

```
# Items and their prices
```

```
items = {
```

```
    "rice": 45,
```

```
    "oil": 120,
```

```
    "salt": 20,
```

```
    "soap": 45,
```

```
    "shampoo": 85,
```

```
    "milk": 25,
```

```
    "doll": 125,
```

```
    "paneer": 100,
```

```
    "paste": 20,
```

```
    "maggie": 40,
```

```
    "sugar": 42,
```

```
    "onion": 40,
```

```
    "juice": 85
```

```
}
```

```
# Show items
```

```
print("\nAvailable Items:")

for item, price in items.items():
    print(f"{item} - Rs {price} each")

# Cart data
cart = []

# Main loop
while True:
    print("\nOptions:")
    print("1. Add item")
    print("2. Remove item")
    print("3. Checkout")

    choice = input("Enter your choice (1/2/3): ")

    if choice == "1":
        item = input("Enter item name to add: ").lower()
        if item in items:
            qty = int(input("Enter quantity: "))
            price = items[item] * qty
            cart.append([item, qty, price])
            print(f"{item} added to cart.")

        else:
            print("Item not available.")

    elif choice == "2":
        if not cart:
```

```
print("Cart is empty.")

else:
    print("\nItems in your cart:")
    for i, c in enumerate(cart, 1):
        print(f"{i}. {c[0]} - Qty: {c[1]}")
    remove_name = input("Enter item name to remove: ").lower()
    found = False
    for i in range(len(cart)):
        if cart[i][0] == remove_name:
            del cart[i]
            print(f"{remove_name} removed from cart.")
            found = True
            break
    if not found:
        print("Item not found in cart.")

elif choice == "3":
    if not cart:
        print("Cart is empty. Nothing to bill.")
        break
    total = sum([item[2] for item in cart])
    gst = total * 0.05
    final = total + gst

    print("\n----- BILL -----")
    print("Name:", name)
    print("Date:", datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
    print("\nItems:")
```

```
for i, c in enumerate(cart, 1):
    print(f"{i}. {c[0]} - Qty: {c[1]} - Rs {c[2]}")
    print("-----")
print(f"Total: Rs {total}")
print(f"GST (5%): Rs {gst:.2f}")
print(f"Final Amount: Rs {final:.2f}")
print("-----")
print("Thank you for shopping with us!")
break
```

else:

```
    print("Invalid choice. Please try again.")
```



Example Output:

mathematica

CopyEdit

Options:

1. Add item
2. Remove item
3. Checkout

Enter your choice (1/2/3): 1

Enter item name to add: rice

Enter quantity: 2

rice added to cart.

Options:

1. Add item
2. Remove item

3. Checkout

Enter your choice (1/2/3): 2

Items in your cart:

1. rice - Qty: 2

Enter item name to remove: rice

rice removed from cart.

Would you like this version to also **ask again after checkout** if they want to shop more, or just exit?

Ask ChatGPT

You said:

superb can you explain me step by step about this program

ChatGPT said:

Absolutely! Let me explain the **simple supermarket cart program** step-by-step so you can fully understand how each part works.

SIMPLE SUPERMARKET CART — STEP-BY-STEP BREAKDOWN

1. Welcome and Setup

python

CopyEdit

from datetime import datetime

- This imports the **datetime** module so we can print the date and time on the bill.

python

CopyEdit

```
print("\n----- Welcome to Malar Supermarket -----\\n")
```

```
name = input("Enter your name: ")
```

- Shows a welcome message.

- Asks for the customer's name.
-

2. Define Items and Prices

python

CopyEdit

```
items = {
```

```
    "rice": 45,
```

```
    "oil": 120,
```

```
    ...
```

```
}
```

- A Python dictionary where each **key is the item name** (like "rice") and the **value is its price** (like 45).
 - You can easily **add or remove items** here.
-

3. Show Available Items

python

CopyEdit

```
print("\nAvailable Items:")
```

```
for item, price in items.items():
```

```
    print(f"{item} - Rs {price} each")
```

- Loops through the dictionary and prints each item with its price.
-

4. Initialize the Cart

python

CopyEdit

```
cart = []
```

- Creates an **empty list** named cart to store what the user adds to their shopping cart.
- Each item in the cart will be stored as a list: [item_name, quantity, total_price].

5. Main Menu Loop

python

CopyEdit

while True:

```
    print("\nOptions:")
    print("1. Add item")
    print("2. Remove item")
    print("3. Checkout")
```

...

- This loop keeps running until the user selects **Checkout**.
- Shows a menu with 3 choices:
 1. Add item to cart
 2. Remove item from cart
 3. Checkout and exit

6. Add Item to Cart

python

CopyEdit

```
if choice == "1":
```

```
    item = input("Enter item name to add: ").lower()
```

...

```
    price = items[item] * qty
```

```
    cart.append([item, qty, price])
```

- Asks for item name and quantity.
- Converts item name to **lowercase** to avoid case mismatch.
- Calculates the total price (price = rate × qty) and **adds it to cart**.

- The cart now contains a list of lists like:
[['rice', 2, 90], ['milk', 3, 75]]
-

7. Remove Item from Cart

python

CopyEdit

```
elif choice == "2":  
  
    ...  
  
    for i in range(len(cart)):  
  
        if cart[i][0] == remove_name:  
  
            del cart[i]
```

- First checks if the cart is empty.
 - Then lists current items and asks for an item name to remove.
 - Searches for that item by name in the cart.
 - If found, removes it using del.
-

8. Checkout (Billing)

python

CopyEdit

```
elif choice == "3":  
  
    ...  
  
    total = sum([item[2] for item in cart])  
  
    gst = total * 0.05  
  
    final = total + gst
```

- Calculates:
 - total: sum of all item prices in the cart
 - gst: 5% tax
 - final: total + gst

Then it prints the bill:

python

CopyEdit

```
for i, c in enumerate(cart, 1):
```

```
    print(f"{i}. {c[0]} - Qty: {c[1]} - Rs {c[2]}")
```

- Prints each item from the cart with its quantity and total price.

Finally:

python

CopyEdit

```
print("Thank you for shopping with us!")
```

```
break
```

- Exits the loop and ends the program after billing.
-