SUPPLY CHAIN MANAGEMENT FOR A GROCERY STORE

Optimizing operations for efficiency and customer satisfaction

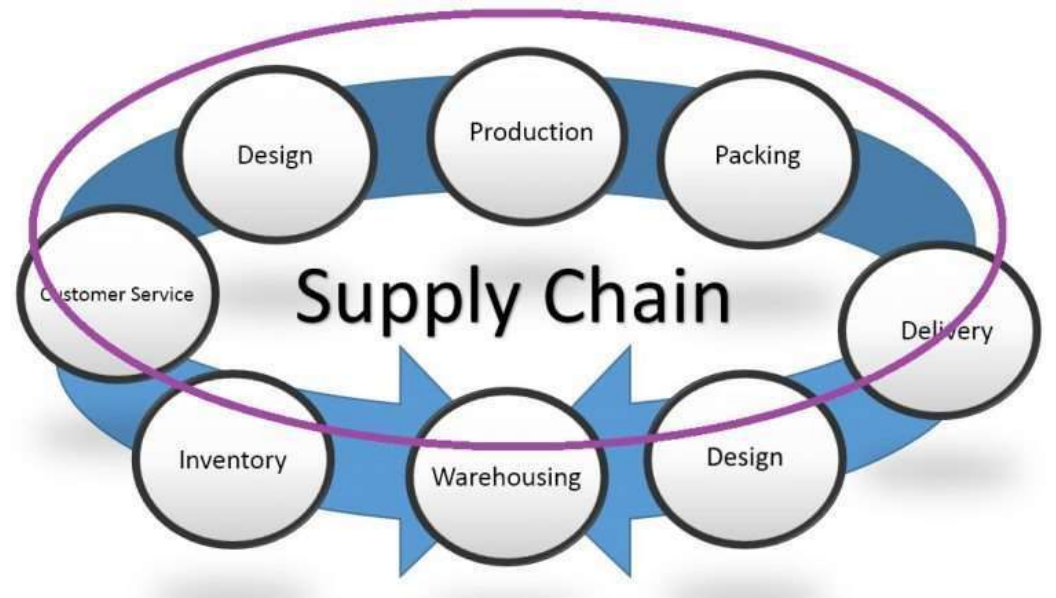
Presented by
R HARINI
ECA2333

content

- Introduction
- Technology used (in python)
- Workflow
- Pros and cons

INTRODUCTION

Grocery supply chains connect food producers and manufacturers with grocery retailers via transportation, warehousing, and inventory management processes



Real-time shelf monitoring, forecasting demand, using point-of-sale technology, and electronic shelf labeling are some of the technologies that can help meet inventory and supply chain needs. AI can also have a positive impact on environmental, social, and governance initiatives in the grocery and retail industry.

SUPPLY CHAIN MANAGEMENT

Optimizing Grocery Supply Chain Coordination

Ensuring Seamless Collaboration for Efficient Operations

Ensuring Resource Access for Producers

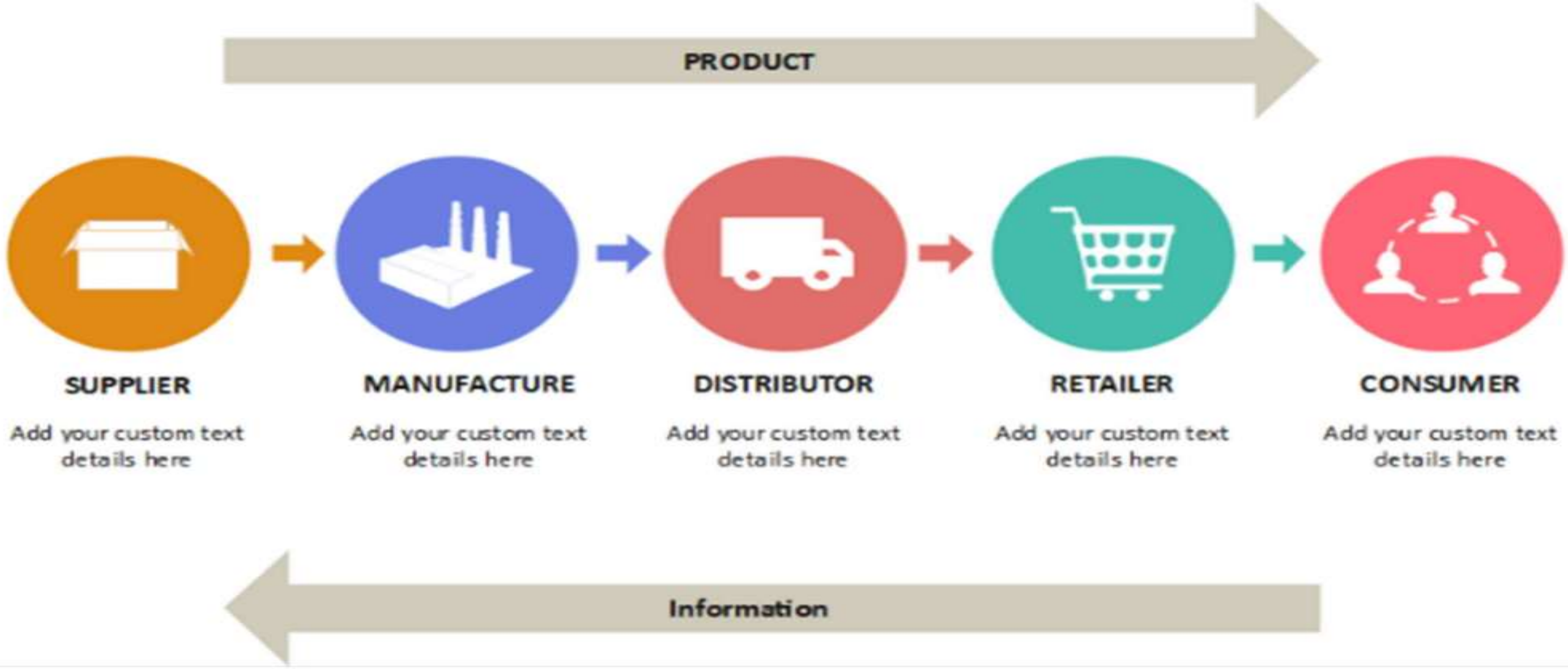
Facilitate producers with necessary resources for quality production

Timely Delivery of Quality Products

Guarantee on-time delivery of high-quality products to customers

Maintaining Adequate Stock Levels

Ensure stores maintain sufficient stock levels to meet demand



PRODUCT

SUPPLIER

Add your custom text details here

MANUFACTURE

Add your custom text details here

DISTRIBUTOR

Add your custom text details here

RETAILER

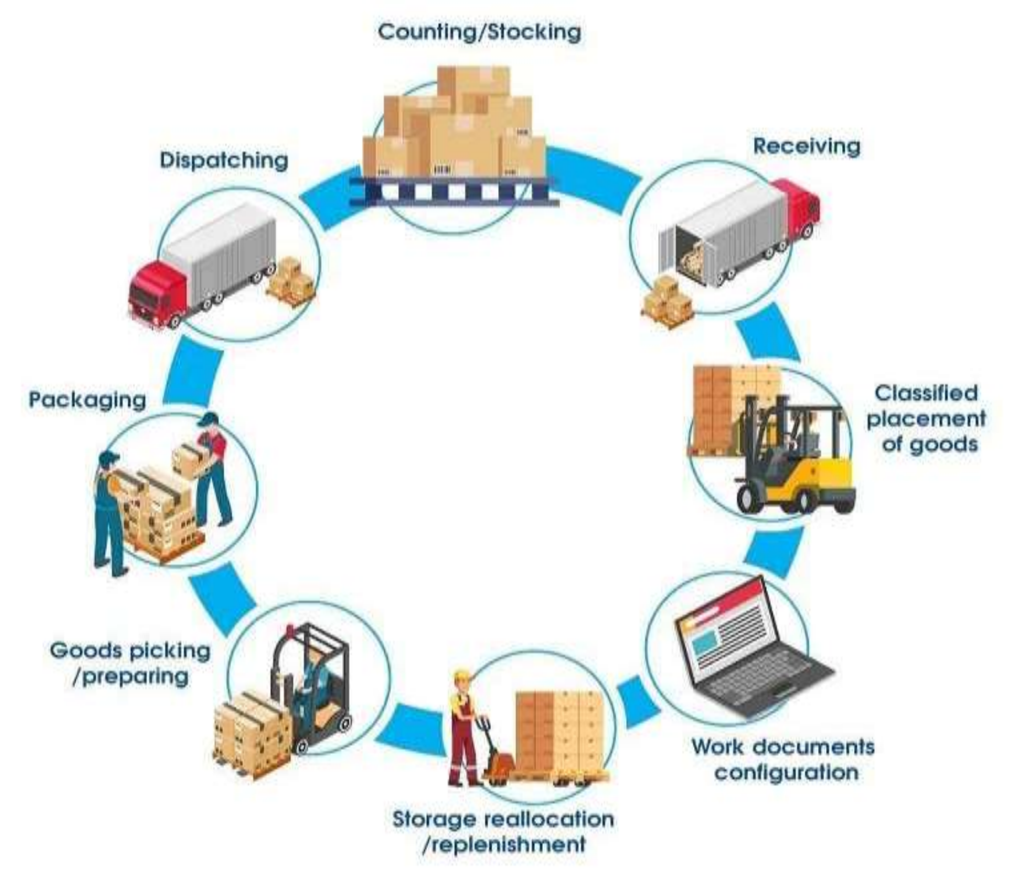
Add your custom text details here

CONSUMER

Add your custom text details here

Information

The grocery supply chain includes sourcing raw ingredients, processing and packaging them, and delivering them to stores for sale. Fresh fruits and vegetables don’t require much manufacturing or processing. They go from the farm to the store more or less as is, but perishable items must be refrigerated during shipping. Packaged goods—such as crackers, breakfast cereals, and margarines—follow a typical manufacturing journey. Raw ingredients are shipped from supplier to manufacturer, processed, combined, and cooked or baked (with varying degrees of automation along the assembly line). The final products are packaged and distributed to buyers, warehouses, or directly to retail grocers.


Counting/Stocking
Receiving
Classified placement of goods
Work documents configuration
Storage reallocation /replenishment
Goods picking /preparing
Packaging
Dispatching

❖ Pros and Cons of chain management

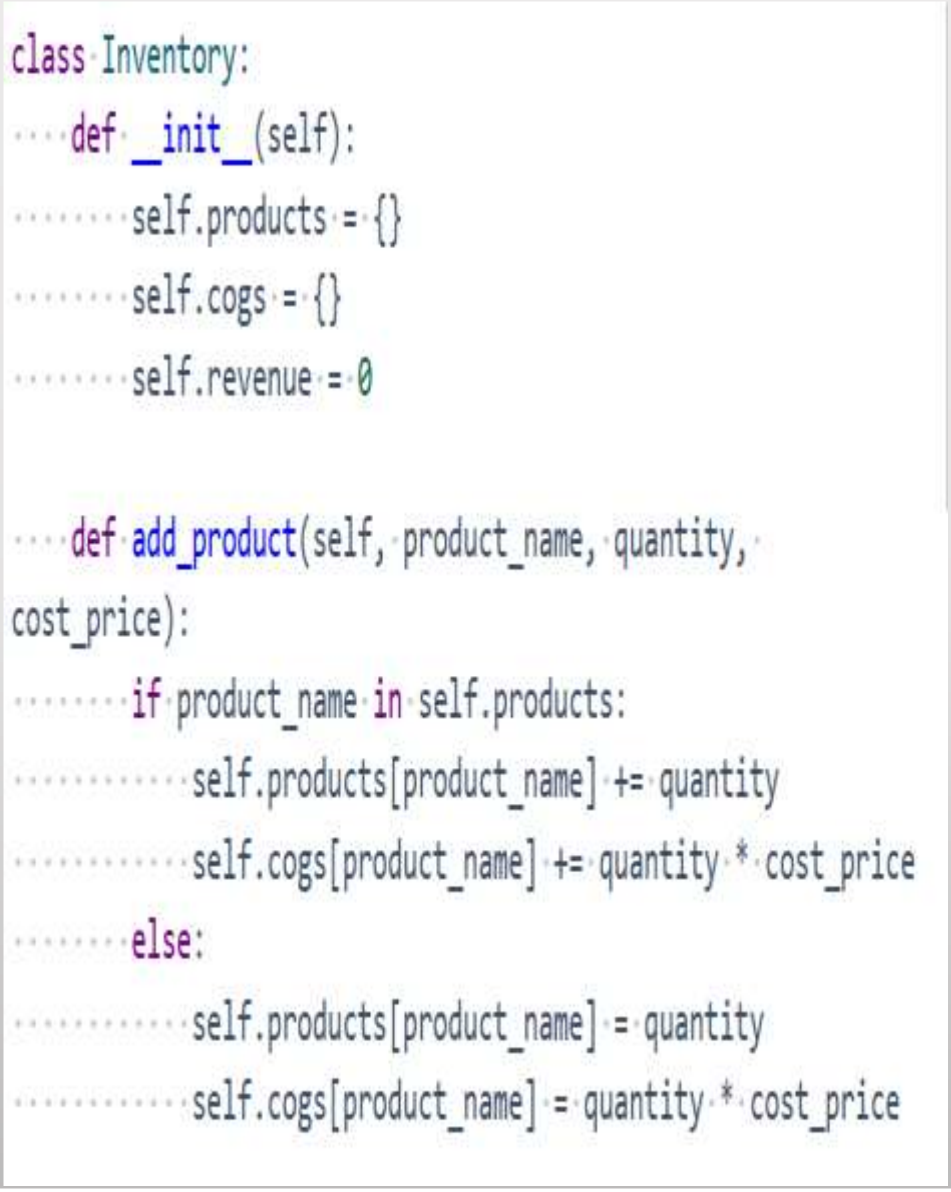
pros

- ✓Cost efficiency
- ✓Enhanced output
- ✓Avoids delays
- ✓Problem identification
- ✓Better collaboration

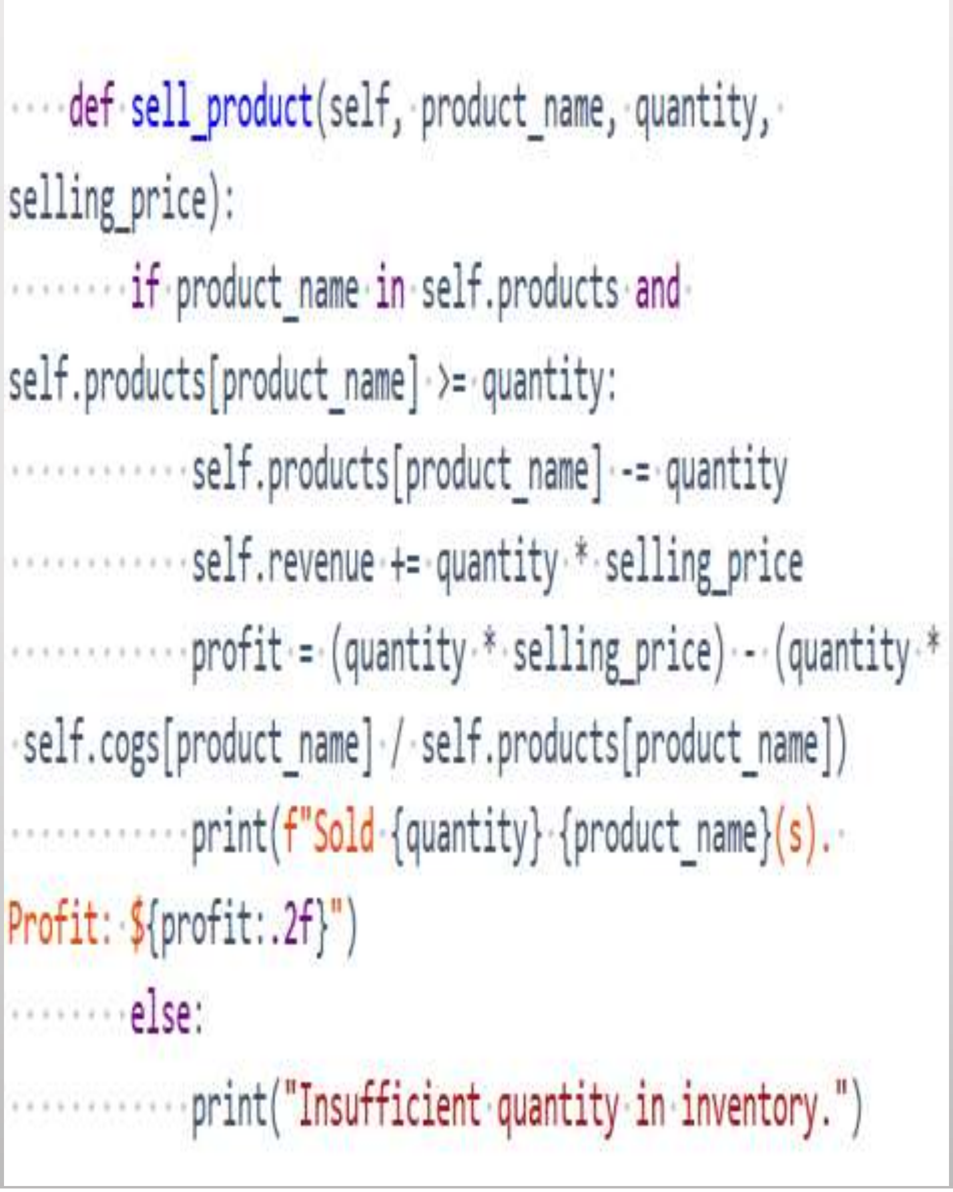
cons

- ▪ Expensive implementation
- ▪ complexity

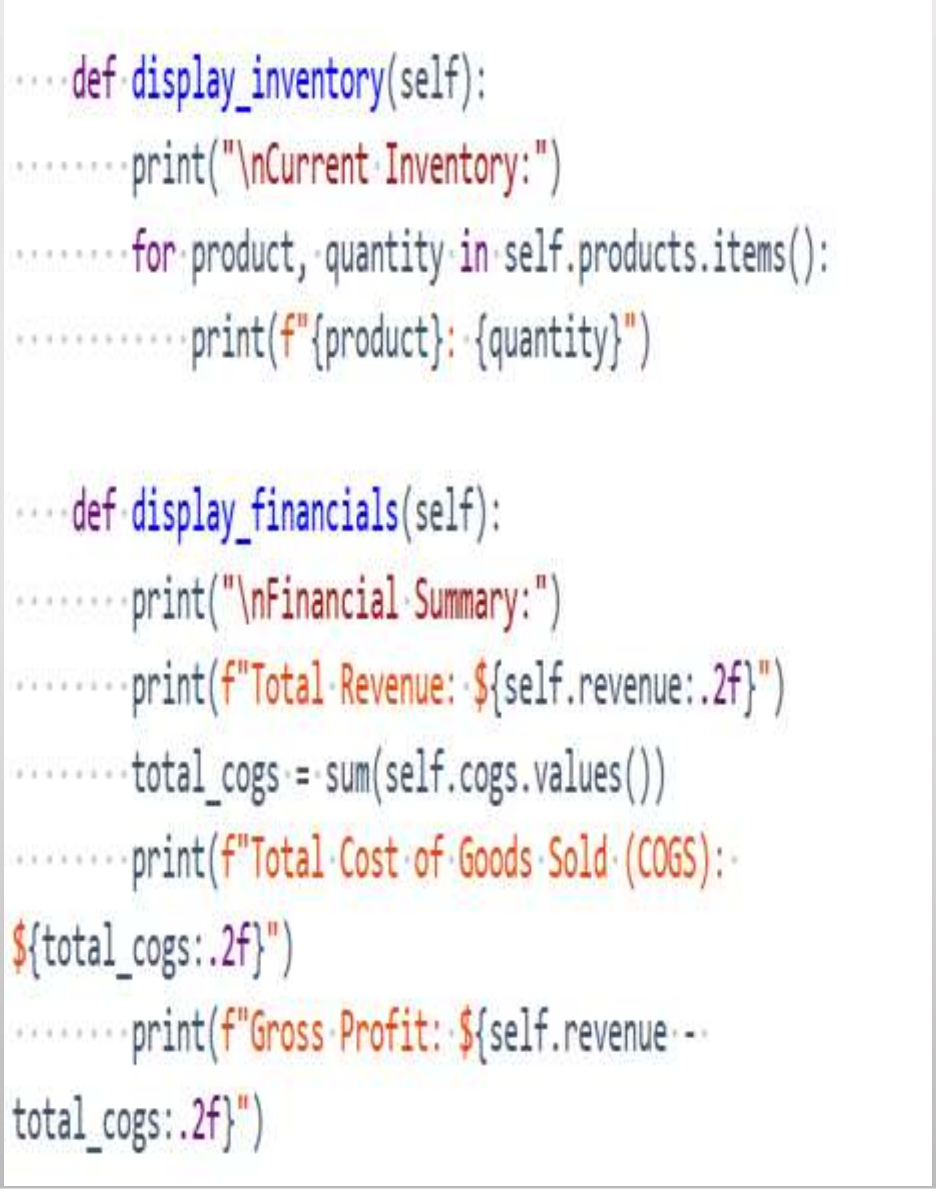
PROGRAM



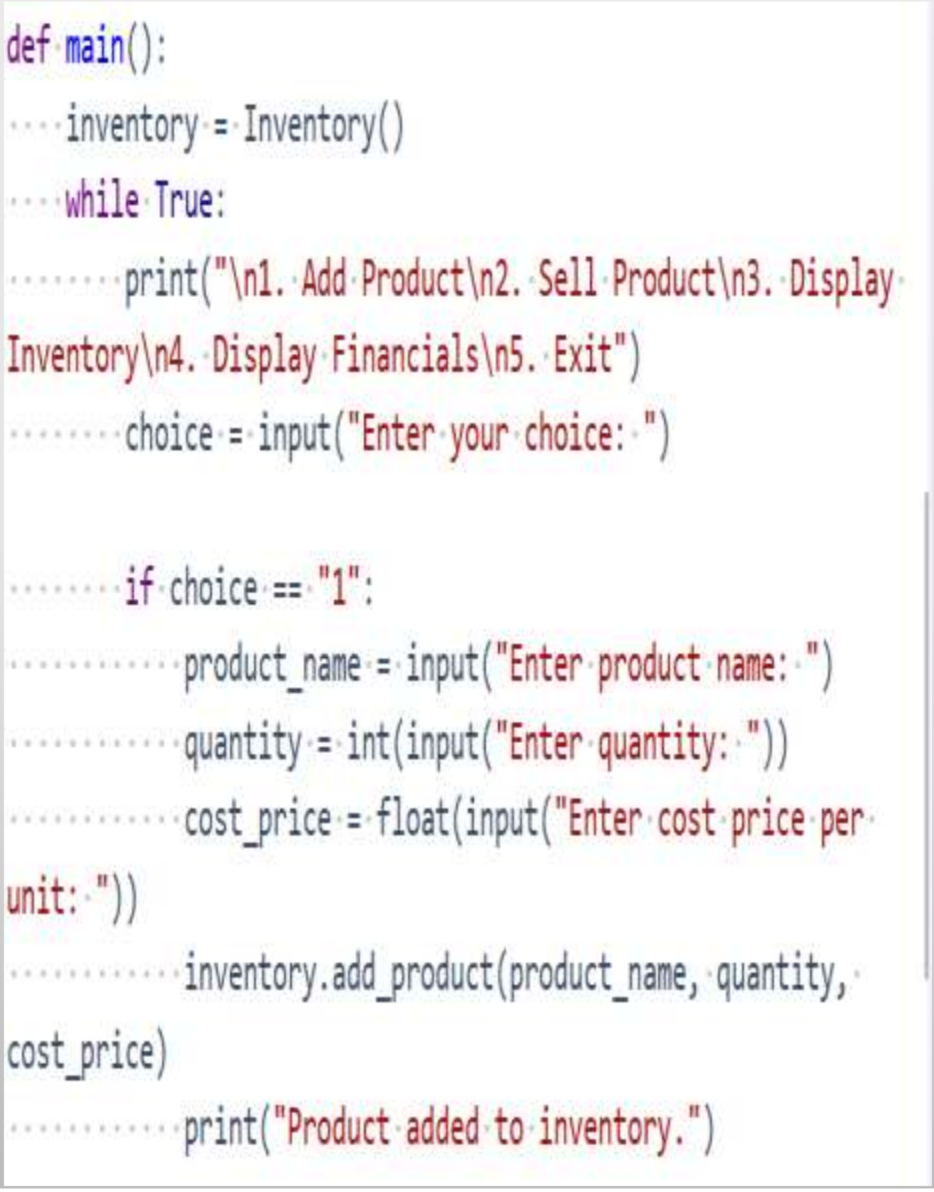
```
class Inventory:
    def __init__(self):
        self.products = {}
        self.cogs = {}
        self.revenue = 0

    def add_product(self, product_name, quantity, 
cost_price):
        if product_name in self.products:
            self.products[product_name] += quantity
            self.cogs[product_name] += quantity * cost_price
        else:
            self.products[product_name] = quantity
            self.cogs[product_name] = quantity * cost_price
```



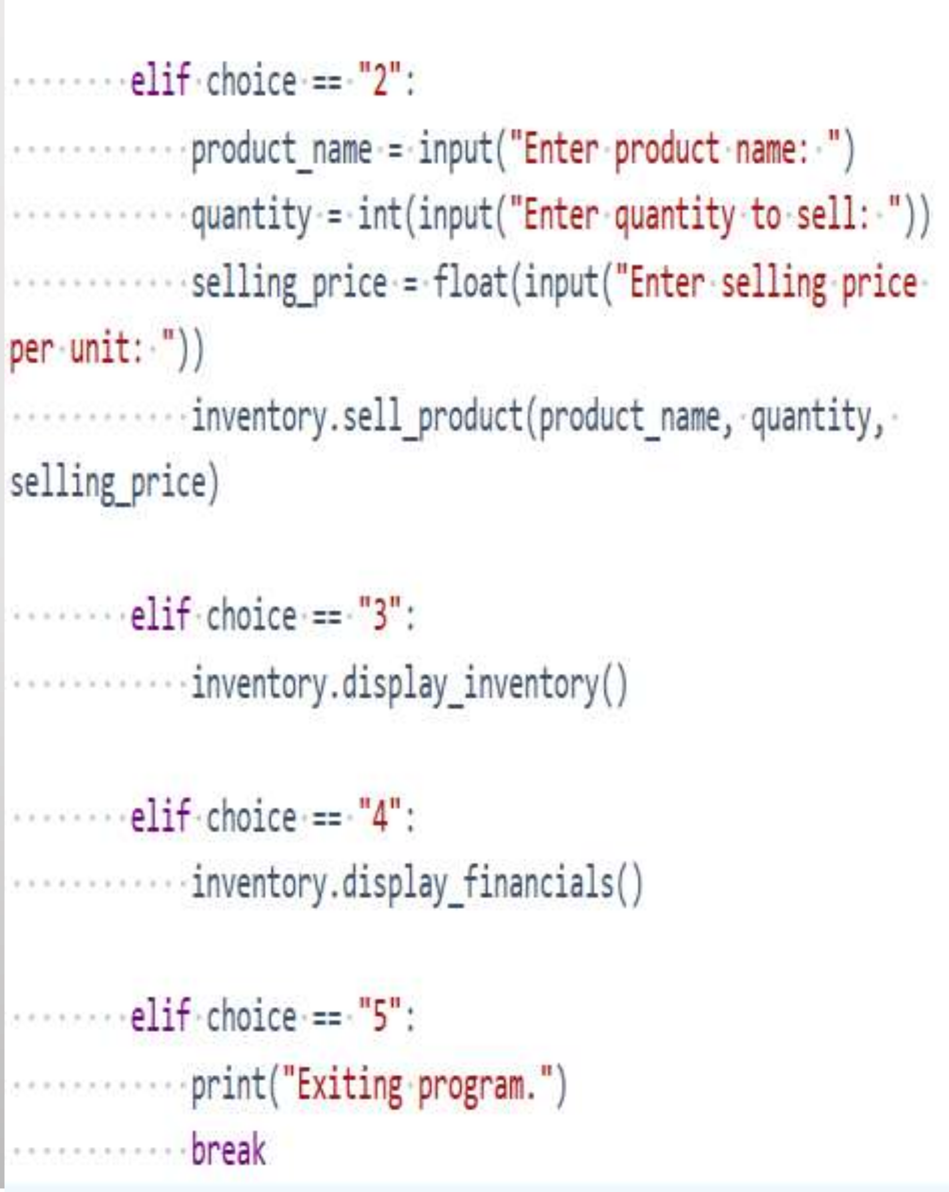
```
    def sell_product(self, product_name, quantity, 
selling_price):
        if product_name in self.products and 
self.products[product_name] >= quantity:
            self.products[product_name] -= quantity
            self.revenue += quantity * selling_price
            profit = (quantity * selling_price) - (quantity * 
 self.cogs[product_name] / self.products[product_name])
            print(f"Sold {quantity} {product_name}(s). 
Profit: ${profit:.2f}")
        else:
            print("Insufficient quantity in inventory.")
```



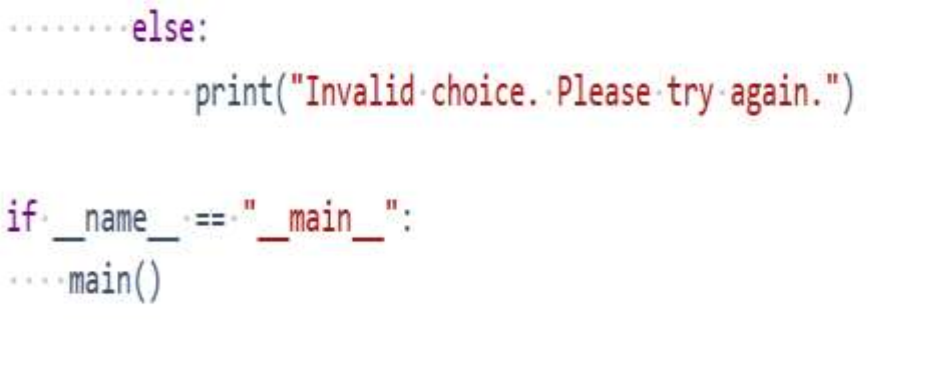
```
    def display_inventory(self):
        print("\nCurrent Inventory:")
        for product, quantity in self.products.items():
            print(f"{product}: {quantity}")

    def display_financials(self):
        print("\nFinancial Summary:")
        print(f"Total Revenue: ${self.revenue:.2f}")
        total_cogs = sum(self.cogs.values())
        print(f"Total Cost of Goods Sold (COGS): 
${total_cogs:.2f}")
        print(f"Gross Profit: ${self.revenue - 
total_cogs:.2f}")
```



```
def main():
    inventory = Inventory()
    while True:
        print("\n1. Add Product\n2. Sell Product\n3. Display 
Inventory\n4. Display Financials\n5. Exit")
        choice = input("Enter your choice: ")

        if choice == "1":
            product_name = input("Enter product name: ")
            quantity = int(input("Enter quantity: "))
            cost_price = float(input("Enter cost price per 
unit: "))
            inventory.add_product(product_name, quantity, 
cost_price)
            print("Product added to inventory.")
```



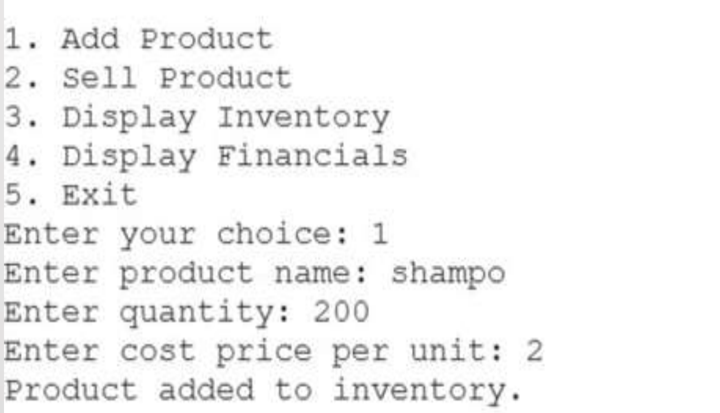
```
        elif choice == "2":
            product_name = input("Enter product name: ")
            quantity = int(input("Enter quantity to sell: "))
            selling_price = float(input("Enter selling price 
per unit: "))
            inventory.sell_product(product_name, quantity, 
selling_price)

        elif choice == "3":
            inventory.display_inventory()

        elif choice == "4":
            inventory.display_financials()

        elif choice == "5":
            print("Exiting program.")
            break
```



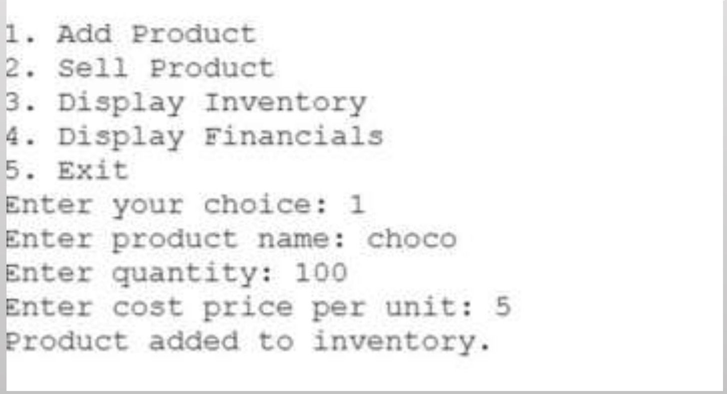
```
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

OUTPUT

Add Product

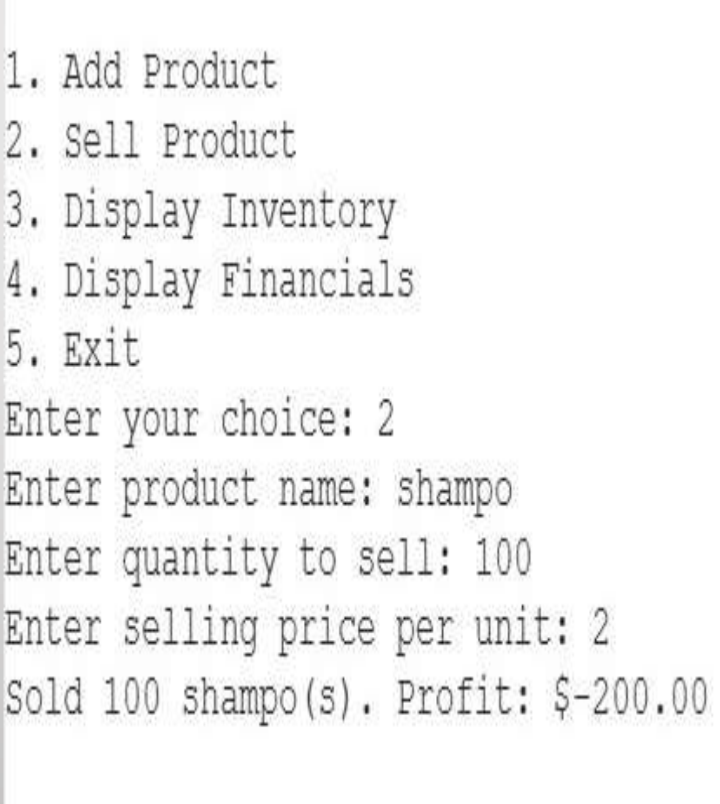


```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 1
Enter product name: shampo
Enter quantity: 200
Enter cost price per unit: 2
Product added to inventory.
```



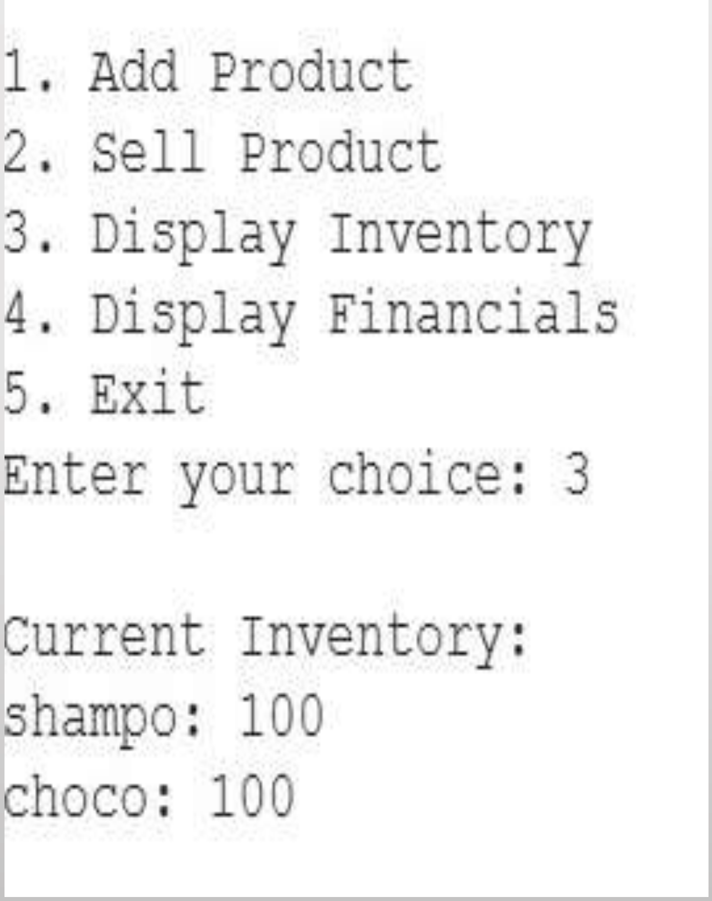
```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 1
Enter product name: choco
Enter quantity: 100
Enter cost price per unit: 5
Product added to inventory.
```

Sell Product



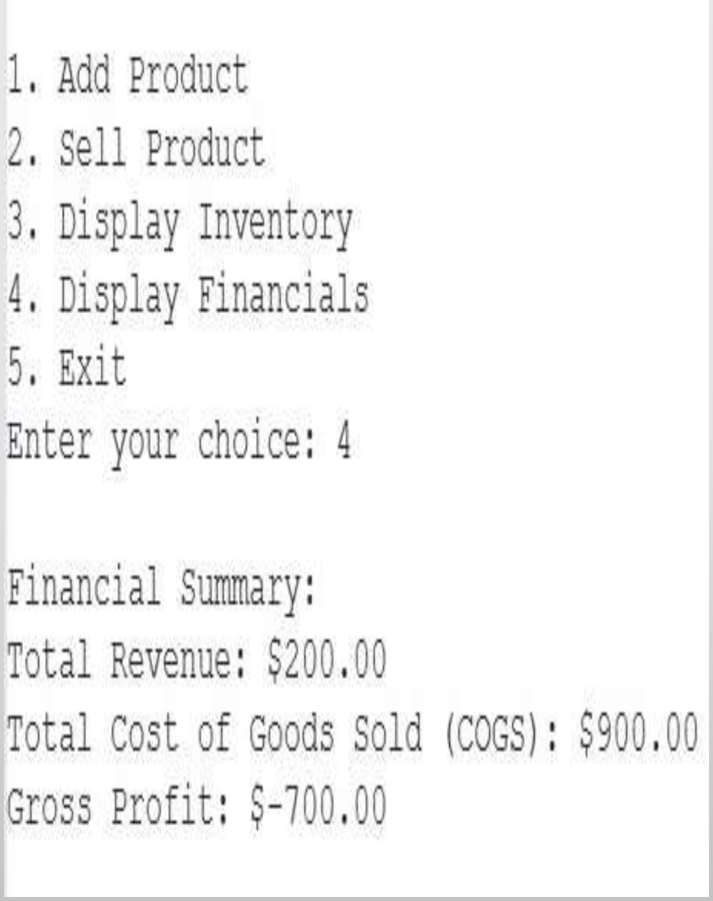
```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 2
Enter product name: shampo
Enter quantity to sell: 100
Enter selling price per unit: 2
Sold 100 shampo(s). Profit: $-200.00
```

Display Inventory



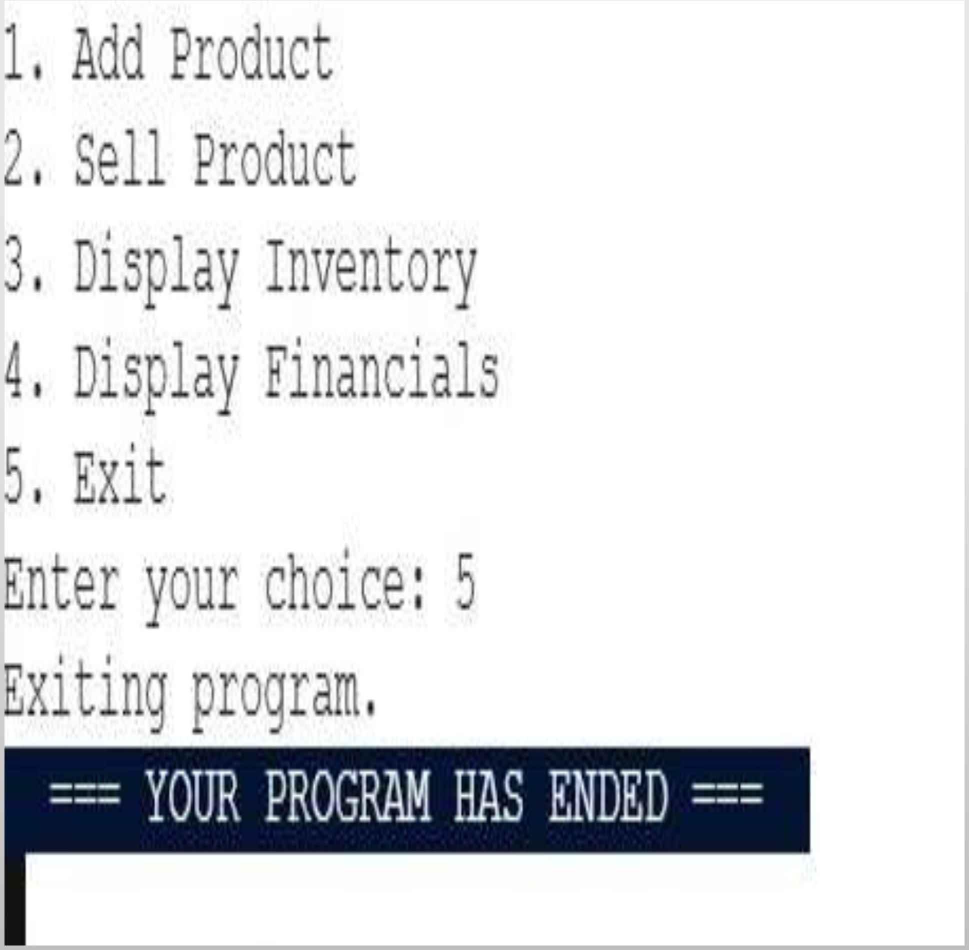
```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 3

Current Inventory:
shampo: 100
choco: 100
```

Display Financials



```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 4

Financial Summary:
Total Revenue: $200.00
Total Cost of Goods Sold (COGS): $900.00
Gross Profit: $-700.00
```

Exit



```
1. Add Product
2. Sell Product
3. Display Inventory
4. Display Financials
5. Exit
Enter your choice: 5
Exiting program.
 === YOUR PROGRAM HAS ENDED ===
```

Thank You!