

Creating a Supermarket System Using Python

About the Project:

In this Coursera guided project, the hands-on project on Creating a Supermarket System Using Python is divided into following tasks:

Task 1: Dictionaries

Task 2: Try Except

Task 3: Viewing and Adding Items

Task 4: Purchasing and Searching for Items

Task 5: Editing Items and Exiting

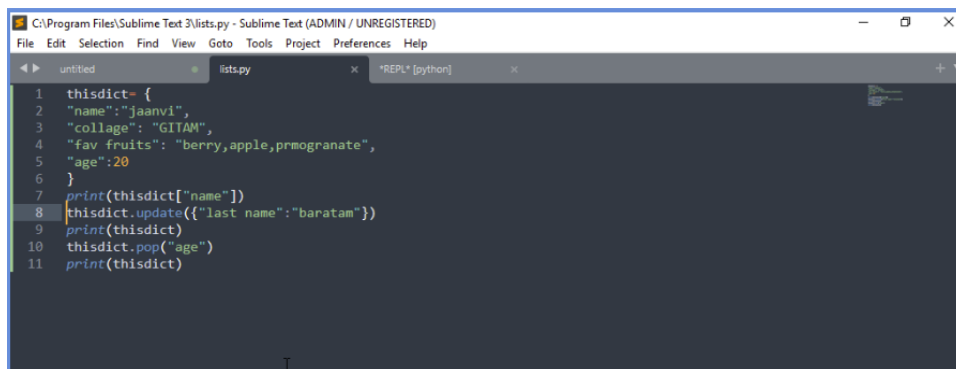
The whole program can be explained as:

1. Main Loop: Your program runs in an infinite loop (``while True``) until the user chooses to exit.
2. View Items (Choice 1): This option displays the current items in the inventory. It iterates through the list of items and prints each item's details.
3. Add Items (Choice 2): Users can add new items to the inventory. The code prompts users for the item's name, quantity, and price and then adds the item to the ``items`` list.
4. Purchase Items (Choice 3): Users can purchase items by specifying the item name and quantity they want to buy. If the item is in stock and the requested quantity is available, it calculates the total cost and deducts the purchased quantity from the inventory. It also handles cases where the item is out of stock, or the requested quantity is not available.
5. Search Items (Choice 4): Users can search for items by entering the item's name. The code iterates through the inventory and prints the details of the matching item. If no match is found, it prints "item not found."

6. Edit Items (Choice 5): Users can edit the details of an existing item. They provide the item's name, and if it matches an item in the inventory, they can update its name, quantity, and price. The updated item details are displayed afterward.
7. Exit (Choice 6): Choosing this option exits the program by breaking out of the loop.
8. Invalid Input Handling: Your code also handles cases where the user enters an invalid option by displaying a message.

TASK-1 DICTIONARIES:

Here we learn how to create dictionaries in python and how to access, edit, update, delete the given information in the dictionaries.



```
C:\Program Files\Sublime Text 3\lists.py - Sublime Text (ADMIN / UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
lists.py
1 thisdict = {
2     "name": "jaanvi",
3     "collage": "GITAM",
4     "fav fruits": "berry,apple,prmgrogrante",
5     "age": 20
6 }
7 print(thisdict["name"])
8 thisdict.update({"last name": "baratam"})
9 print(thisdict)
10 thisdict.pop("age")
11 print(thisdict)
```

TASK-2 TRY EXCEPT:

Here in try and except we mainly learn about exception handling and how it can be dealt in python. There are two main functions that can be done here. They are represented in blocks. They are:

- 1.The try block.
- 2.The except block.

Its syntax is:

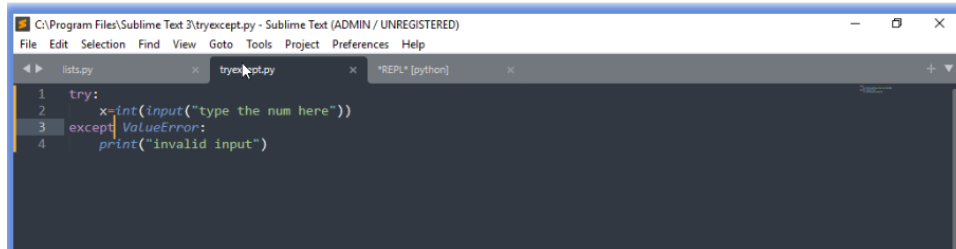
try:

Some Code

except:

Executed if error in the

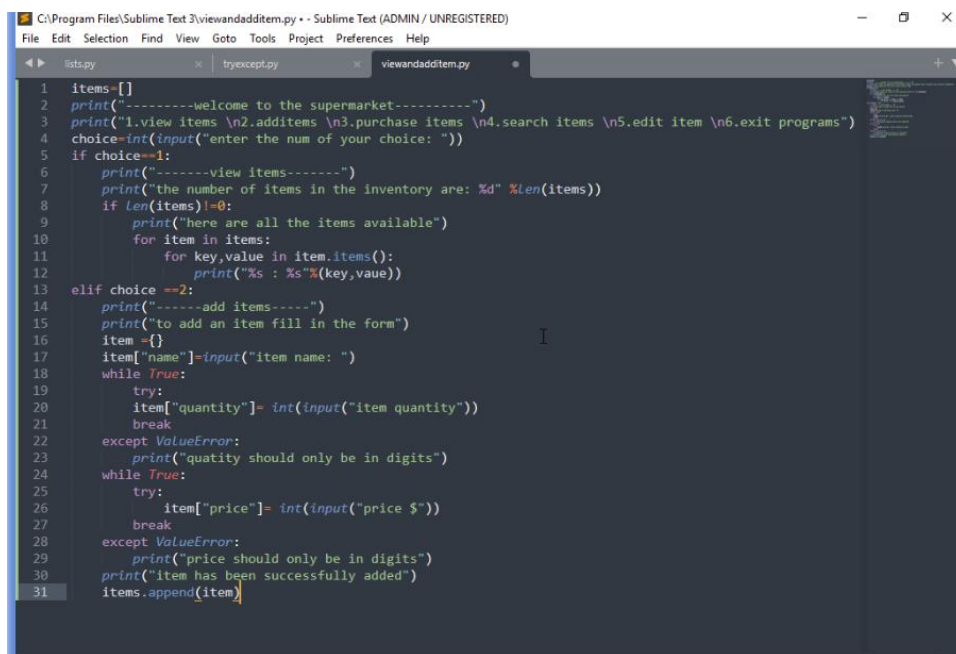
try block



```
C:\Program Files\Sublime Text 3\tryexcept.py - Sublime Text (ADMIN / UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
lists.py tryexcept.py *REPL* [python]
1 try:
2     x=int(input("type the num here"))
3 except ValueError:
4     print("invalid input")
```

TASK-3 VIEWING AND ADDING ITEMS:

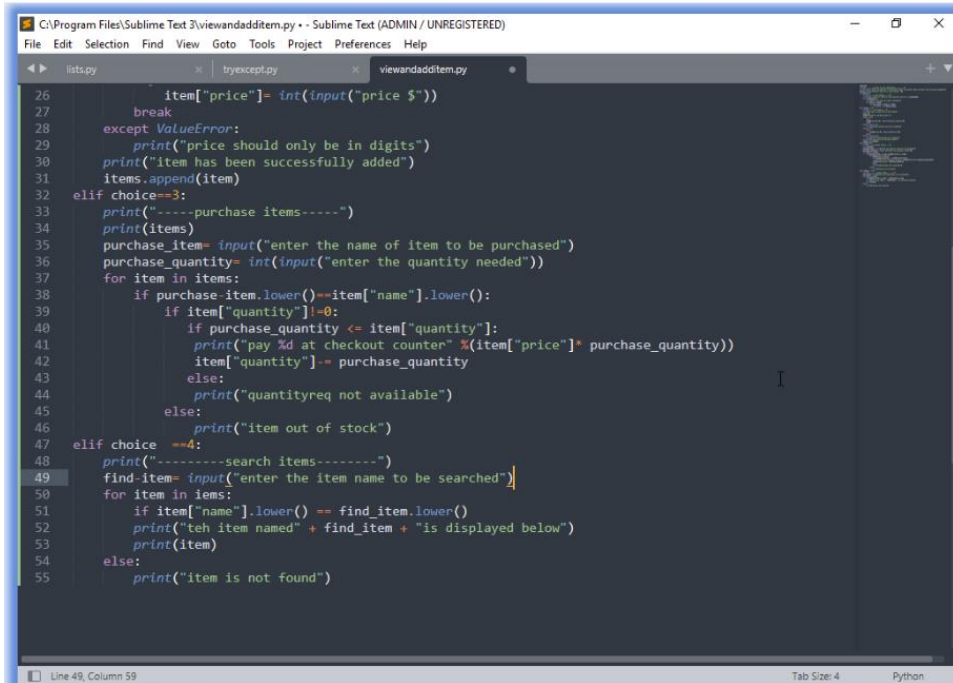
Here we use the help of dictionaries and try except methods for creating the main code. The first part is viewing and adding items in the list of the items.



```
C:\Program Files\Sublime Text 3\viewandadditem.py - Sublime Text (ADMIN / UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
lists.py tryexcept.py viewandadditem.py
1 items=[]
2 print("-----welcome to the supermarket-----")
3 print("1.view items \n2.additems \n3.purchase items \n4.search items \n5.edit item \n6.exit programs")
4 choice=int(input("enter the num of your choice: "))
5 if choice==1:
6     print("-----view items-----")
7     print("the number of items in the inventory are: %d" %len(items))
8     if len(items)!=0:
9         print("here are all the items available")
10        for item in items:
11            for key,value in item.items():
12                print("%s : %s"%(key,vauue))
13 elif choice ==2:
14     print("-----add items-----")
15     print("to add an item fill in the form")
16     item ={}
17     item["name"]=input("item name: ")
18     while True:
19         try:
20             item["quantity"]-= int(input("item quantity"))
21             break
22         except ValueError:
23             print("quacity should only be in digits")
24     while True:
25         try:
26             item["price"]-= int(input("price $"))
27             break
28         except ValueError:
29             print("price should only be in digits")
30     print("item has been successfully added")
31     items.append(item)
```

TASK-4 PURCHASING AND SERCHING FOR ITEMS:

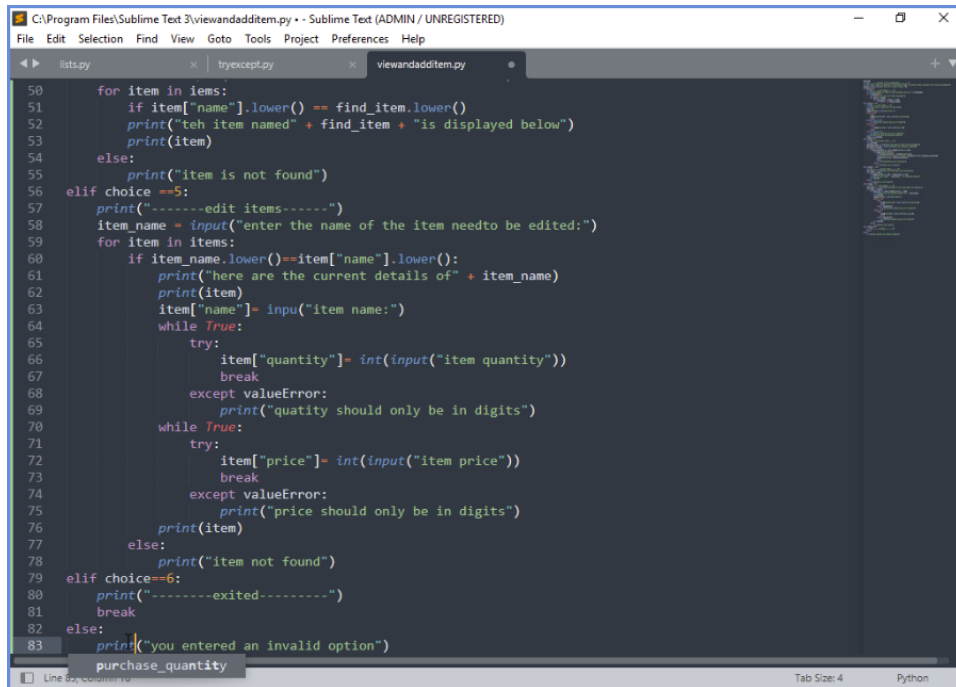
Here we use the for, if, else, loops to run the code. The second part of the code is purchasing and searching for items.



```
C:\Program Files\Sublime Text 3\viewandadditem.py - Sublime Text (ADMIN / UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
lists.py tryexcept.py viewandadditem.py
26 item["price"] = int(input("price $"))
27 break
28 except ValueError:
29     print("price should only be in digits")
30     print("item has been successfully added")
31     items.append(item)
32 elif choice==3:
33     print("-----purchase items-----")
34     print(items)
35     purchase_item= input("enter the name of item to be purchased")
36     purchase_quantity= int(input("enter the quantity needed"))
37     for item in items:
38         if purchase_item.lower()==item["name"].lower():
39             if item["quantity"]!=0:
40                 if purchase_quantity <= item["quantity"]:
41                     print("pay %d at checkout counter" %(item["price"]* purchase_quantity))
42                     item["quantity"]-= purchase_quantity
43                 else:
44                     print("quantityreq not available")
45             else:
46                 print("item out of stock")
47 elif choice ==4:
48     print("-----search items-----")
49     find_item= input("enter the item name to be searched")
50     for item in items:
51         if item["name"].lower() == find_item.lower():
52             print("teh item named" + find_item + "is displayed below")
53             print(item)
54         else:
55             print("item is not found")
```

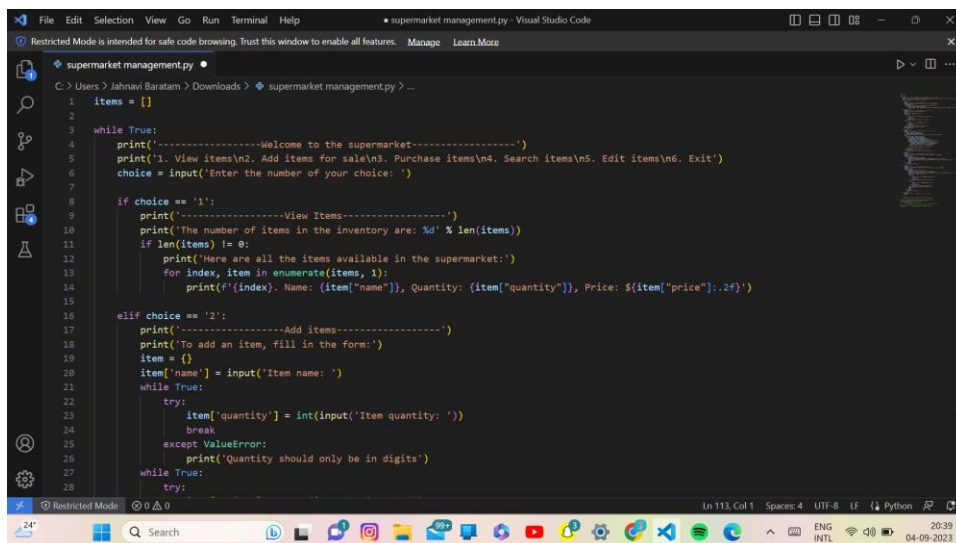
TASK-5 EDITING ITEMS AND EXITING:

Here we add a new function called the while true loop in which the try and except functions are present. We end the loop using the break function. The third part of the code is about editing items and exiting from the program.



```
50     for item in items:
51         if item["name"].lower() == find_item.lower():
52             print("teh item named" + find_item + "is displayed below")
53             print(item)
54     else:
55         print("item is not found")
56 elif choice == 5:
57     print("-----edit items-----")
58     item_name = input("enter the name of the item need to be edited:")
59     for item in items:
60         if item_name.lower() == item["name"].lower():
61             print("here are the current details of" + item_name)
62             print(item)
63             item["name"] = input("item name:")
64             while True:
65                 try:
66                     item["quantity"] = int(input("item quantity"))
67                     break
68                 except ValueError:
69                     print("quantity should only be in digits")
70             while True:
71                 try:
72                     item["price"] = int(input("item price"))
73                     break
74                 except ValueError:
75                     print("price should only be in digits")
76             print(item)
77         else:
78             print("item not found")
79 elif choice == 6:
80     print("-----exited-----")
81     break
82 else:
83     print("you entered an invalid option")
84     purchase_quantity
```

MODIFICATIONS ON EXISTING CODE:



```
1 items = []
2
3 while True:
4     print('-----Welcome to the supermarket-----')
5     print('1. View items\n2. Add items for sale\n3. Purchase items\n4. Search items\n5. Edit items\n6. Exit')
6     choice = input('Enter the number of your choice: ')
7
8     if choice == '1':
9         print('-----View Items-----')
10        print('The number of items in the inventory are: %d' % len(items))
11        if len(items) != 0:
12            print('Here are all the items available in the supermarket:')
13            for index, item in enumerate(items, 1):
14                print(f'{index}. Name: {item["name"]}, Quantity: {item["quantity"]}, Price: ${item["price"]:.2f}')
15
16    elif choice == '2':
17        print('-----Add items-----')
18        print('To add an item, fill in the form:')
19        item = {}
20        item["name"] = input('Item name: ')
21        while True:
22            try:
23                item["quantity"] = int(input('Item quantity: '))
24                break
25            except ValueError:
26                print('Quantity should only be in digits')
27        while True:
28            try:
```

```
File Edit Selection View Go Run Terminal Help • supermarket management.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

supermarket management.py
C:\Users> Jahnvi Baratham > Downloads > supermarket management.py > ...

28         try:
29             item['price'] = float(input('Price $: '))
30             break
31         except ValueError:
32             print('Price should be a numeric value')
33         print('Item has been successfully added.')
34         items.append(item)
35
36     elif choice == '3':
37         print('-----Purchase items-----')
38         print('Items in stock:')
39         for index, item in enumerate(items, 1):
40             print(f'{index}. {item["name"]}')
41         purchase_index = input('Enter the number of the item you want to purchase: ')
42         try:
43             purchase_index = int(purchase_index)
44             if 1 <= purchase_index <= len(items):
45                 selected_item = items[purchase_index - 1]
46                 print(f'Selected item: {selected_item["name"]}')
47                 purchase_quantity = int(input('Enter the quantity wanted: '))
48                 if purchase_quantity <= selected_item['quantity']:
49                     total_cost = selected_item['price'] * purchase_quantity
50                     print(f'Pay ${total_cost:.2f} at checkout counter.')
51                     selected_item['quantity'] -= purchase_quantity
52                 else:
53                     print("Quantity required is not available")
54             else:
55                 print("Invalid item number")
```

```
File Edit Selection View Go Run Terminal Help • supermarket management.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

supermarket management.py
C:\Users> Jahnvi Baratham > Downloads > supermarket management.py > ...

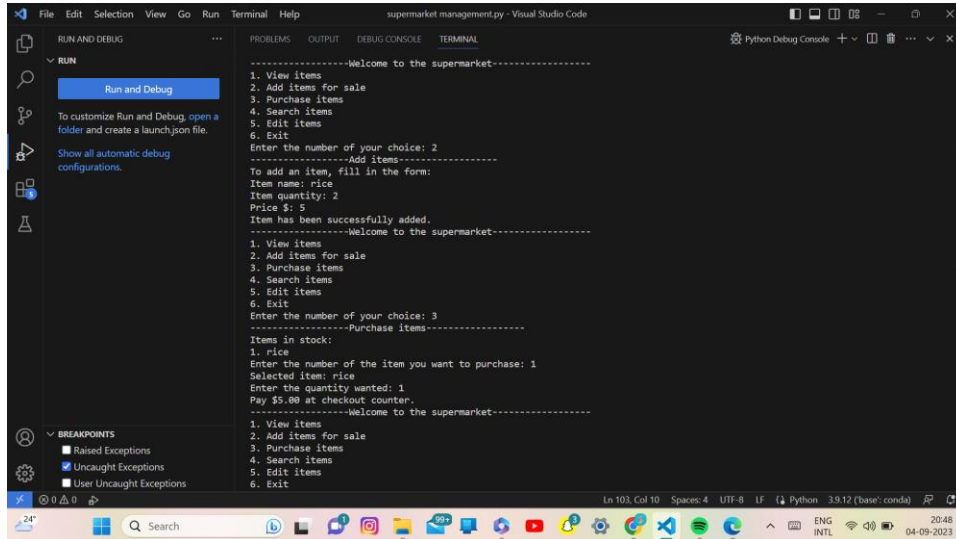
55         print("Invalid item number")
56     except ValueError:
57         print('Invalid input. Please enter a valid number.')
58
59     elif choice == '4':
60         print('-----Search items-----')
61         find_item = input('Enter the item name to search in inventory: ').lower()
62         found = False
63         for item in items:
64             if find_item == item['name'].lower():
65                 print(f'The item named {find_item} is displayed below with its details:')
66                 print(f'Name: {item["name"]}, Quantity: {item["quantity"]}, Price: ${item["price"]:.2f}')
67                 found = True
68                 break
69         if not found:
70             print('Item not found.')
71
72     elif choice == '5':
73         print('-----Edit items-----')
74         item_name = input('Enter the name of the item that you want to edit: ').lower()
75         found = False
76         for item in items:
77             if item_name == item['name'].lower():
78                 print(f'Here are the current details of {item_name}:')
79                 print(f'Name: {item["name"]}, Quantity: {item["quantity"]}, Price: ${item["price"]:.2f}')
80                 item['name'] = input('Item name: ')
81                 while True:
82                     try:
```

```
File Edit Selection View Go Run Terminal Help • supermarket management.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

supermarket management.py
C:\Users> Jahnvi Baratham > Downloads > supermarket management.py > ...

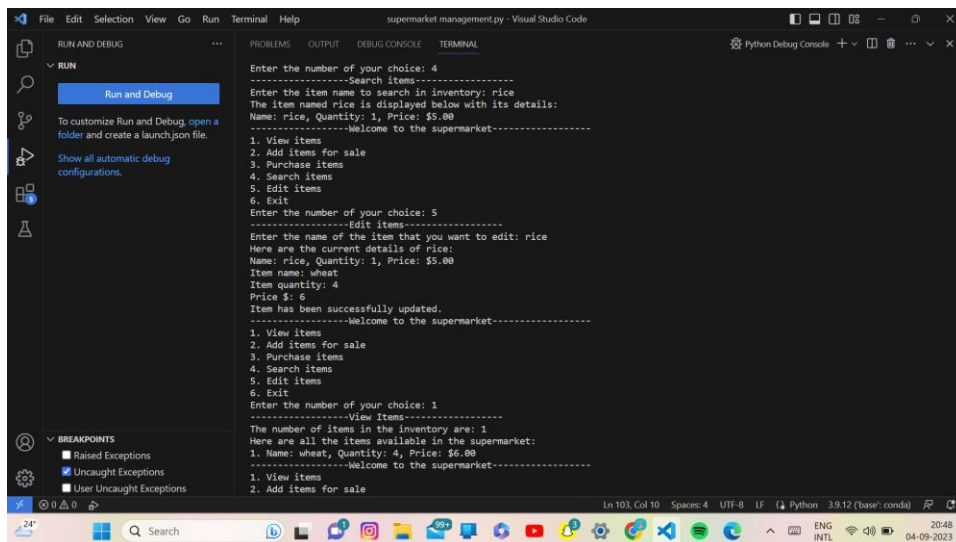
82         try:
83             item['quantity'] = int(input('Item quantity: '))
84             break
85         except ValueError:
86             print('Quantity should only be in digits')
87     while True:
88         try:
89             item['price'] = float(input('Price $: '))
90             break
91         except ValueError:
92             print('Price should be a numeric value')
93         print('Item has been successfully updated.')
94         found = True
95         break
96     if not found:
97         print('Item not found.')
98
99     elif choice == '6':
100         print('-----Exiting-----')
101         break
102
103     else:
104         print('You entered an invalid option. Please try again.')
105
106
107 print('Thank you for using the supermarket management system!')
108
```

THE OUTPUT OF THE MODIFIED CODE:



The screenshot shows the Visual Studio Code interface with the 'Terminal' tab active. The terminal output displays the program's execution flow: a welcome message, a menu of options (1-6), and user input for option 2. It then prompts for item details (name, quantity, price) and confirms the addition. The process repeats for option 3 (purchase), showing the item in stock and the checkout process. The terminal status bar at the bottom indicates 'Ln 103, Col 10' and 'Python 3.9.12 (base:conda)'.

```
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 2
-----Add items-----
To add an item, fill in the form:
Item name: rice
Item quantity: 2
Price $: 5
Item has been successfully added.
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 3
-----Purchase items-----
Items in stock:
1. rice
Enter the number of the item you want to purchase: 1
Selected item: rice
Enter the quantity wanted: 1
Pay $5.00 at checkout counter.
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
```



The screenshot shows the Visual Studio Code interface with the 'Terminal' tab active. The terminal output displays the program's execution flow: a welcome message, a menu of options (1-6), and user input for option 4. It then prompts for the item name to search, displays the item details (Name: rice, Quantity: 1, Price: \$5.00), and returns to the main menu. The process repeats for option 5 (edit), showing the current details of rice and allowing the user to update the item name to 'wheat' and the price to \$6.00. The terminal status bar at the bottom indicates 'Ln 103, Col 10' and 'Python 3.9.12 (base:conda)'.

```
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 4
-----Search items-----
Enter the item name to search in inventory: rice
The item named rice is displayed below with its details:
Name: rice, Quantity: 1, Price: $5.00
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 5
-----Edit items-----
Enter the name of the item that you want to edit: rice
Here are the current details of rice:
Name: rice, Quantity: 1, Price: $5.00
Item name: wheat
Item quantity: 4
Price $: 6
Item has been successfully updated.
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 1
-----View items-----
The number of items in the inventory are: 1
Here are all the items available in the supermarket:
1. Name: wheat, Quantity: 4, Price: $6.00
-----Welcome to the supermarket-----
1. View items
2. Add items for sale
```

