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."

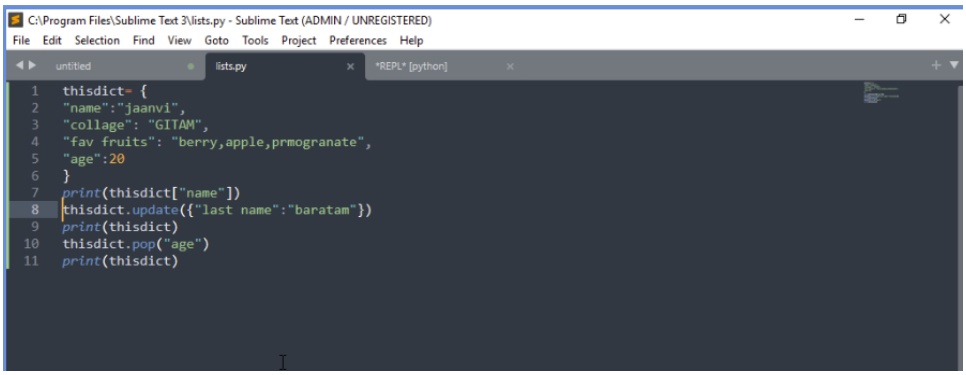
6. Edit Items (Choice 5): Users can edit the details of an existing item. They provide the item's name, and if it matches an item in the inventory, they can update its name, quantity, and price. The updated item details are displayed afterward.

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.



TASK-2 TRY EXCEPT:

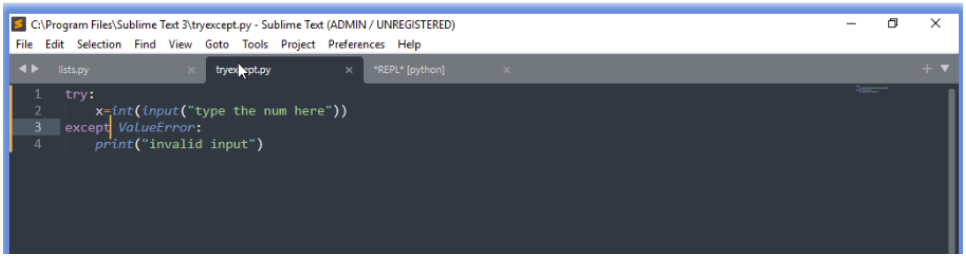
Here in try and except we mainly learn about exception handling and how it can be dealt in python. There are two main functions that can be done here. They are represented in blocks. They are:

1.The try block.

2.The except block.

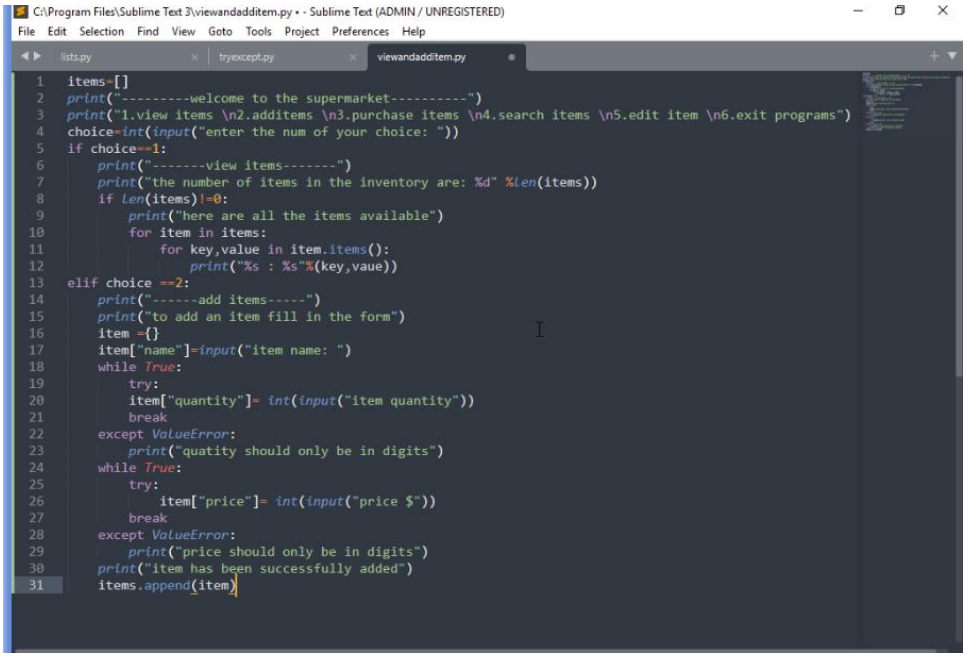
Its syntax is:

```
try:
    # Some Code
except:
    # Executed if error in the
    # try block
```



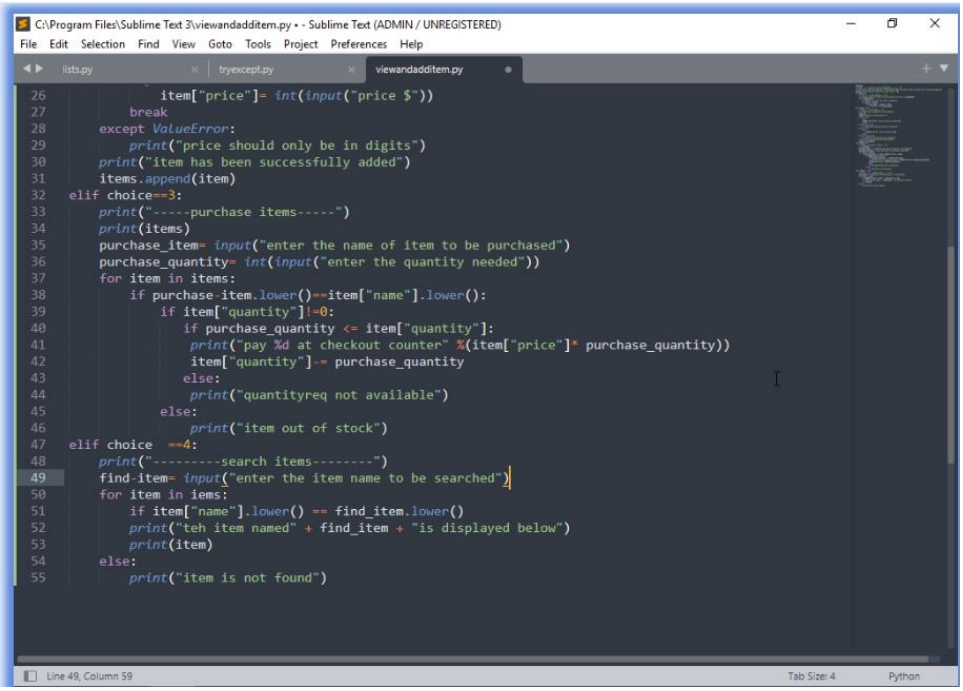
TASK-3 VIEWING AND ADDING ITEMS:

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.



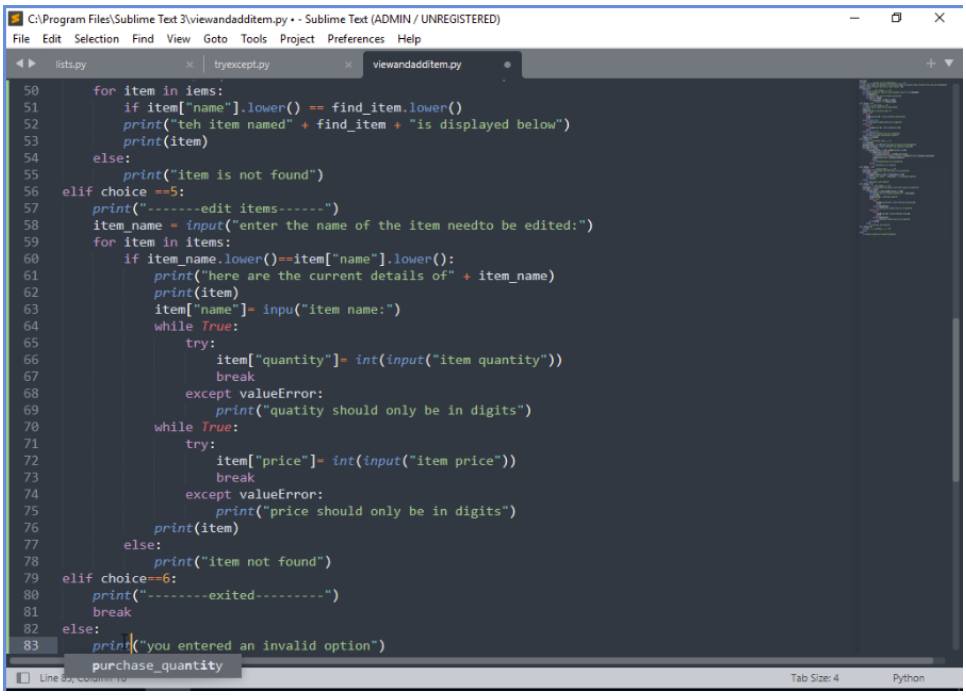
TASK-4 PURCHASING AND SERCHING FOR ITEMS:

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

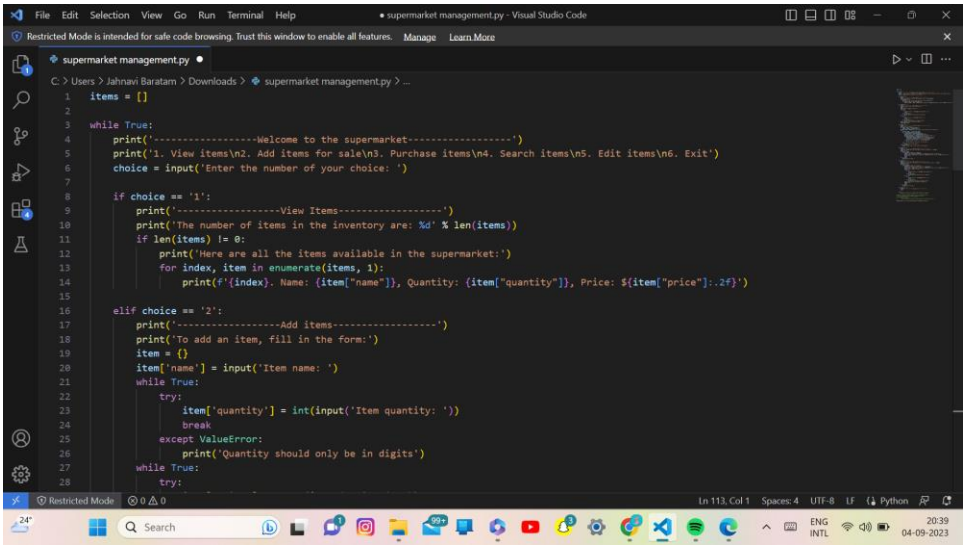


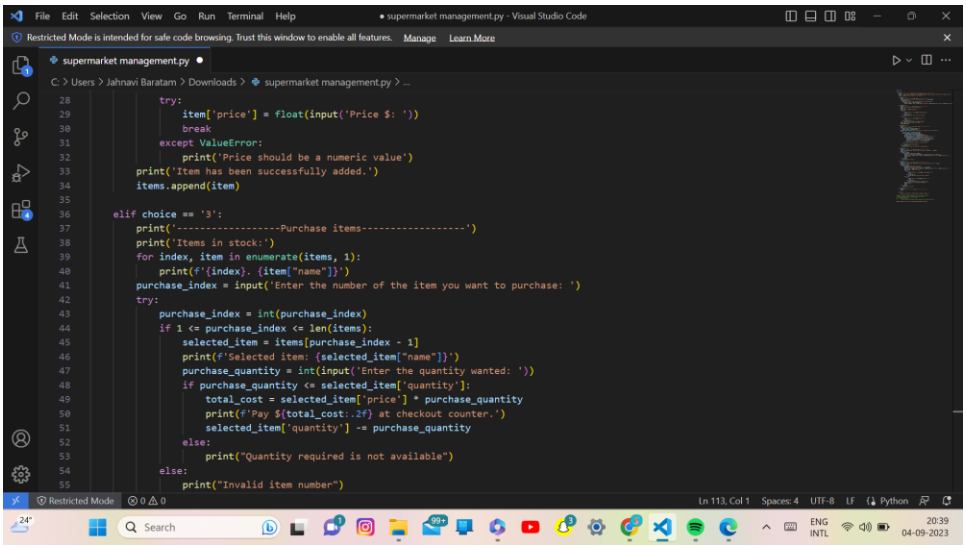
TASK-5 EDITING ITEMS AND EXITING:

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.

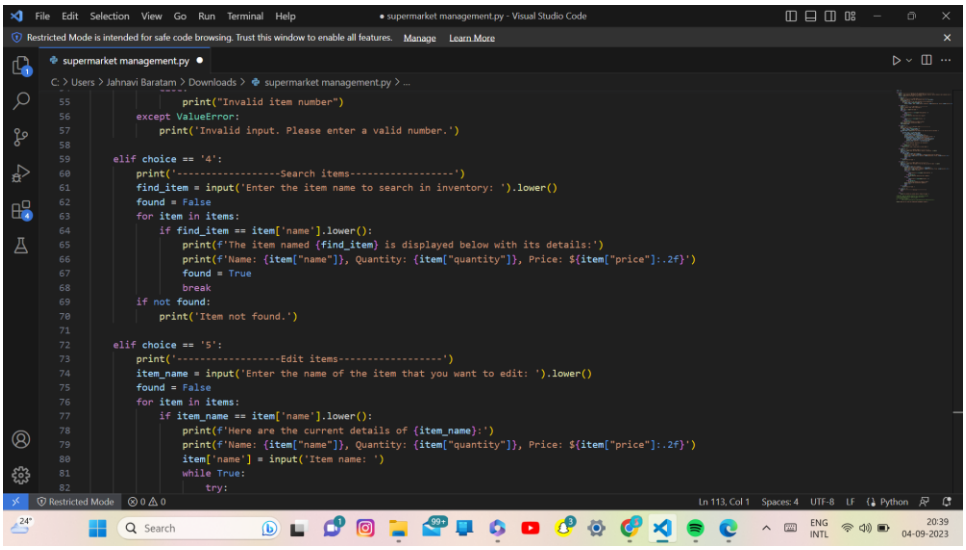


MODIFICATIONS ON EXISTING CODE:

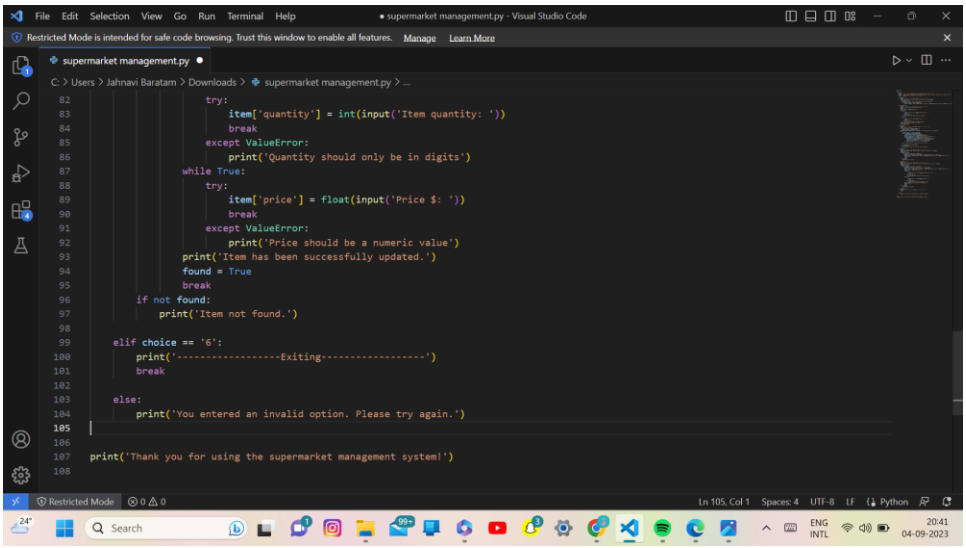




```
            try:
                item['price'] = float(input('Price $: '))
                break
            except ValueError:
                print('Price should be a numeric value')
        print('Item has been successfully added.')
        items.append(item)

    elif choice == '3':
        print('------------------Purchase items------------------')
        print('Items in stock:')
        for index, item in enumerate(items, 1):
            print(f'{index}. {item["name"]}')
        purchase_index = input('Enter the number of the item you want to purchase: ')
        try:
            purchase_index = int(purchase_index)
            if 1 <= purchase_index <= len(items):
                selected_item = items[purchase_index - 1]
                print(f'Selected item: {selected_item["name"]}')
                purchase_quantity = int(input('Enter the quantity wanted: '))
                if purchase_quantity <= selected_item['quantity']:
                    total_cost = selected_item['price'] * purchase_quantity
                    print(f'Pay ${total_cost:.2f} at checkout counter.')
                    selected_item['quantity'] -= purchase_quantity
                else:
                    print("Quantity required is not available")
            else:
                print("Invalid item number")
```

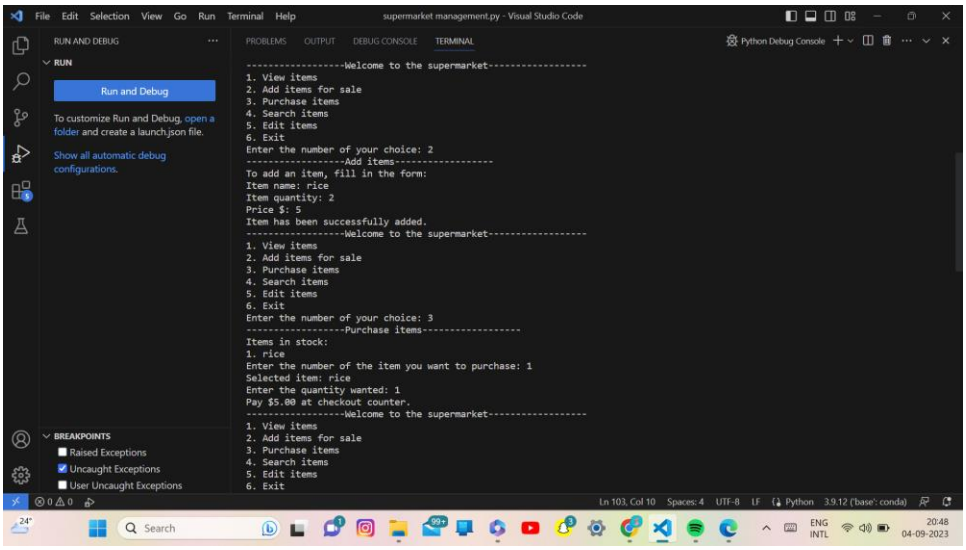


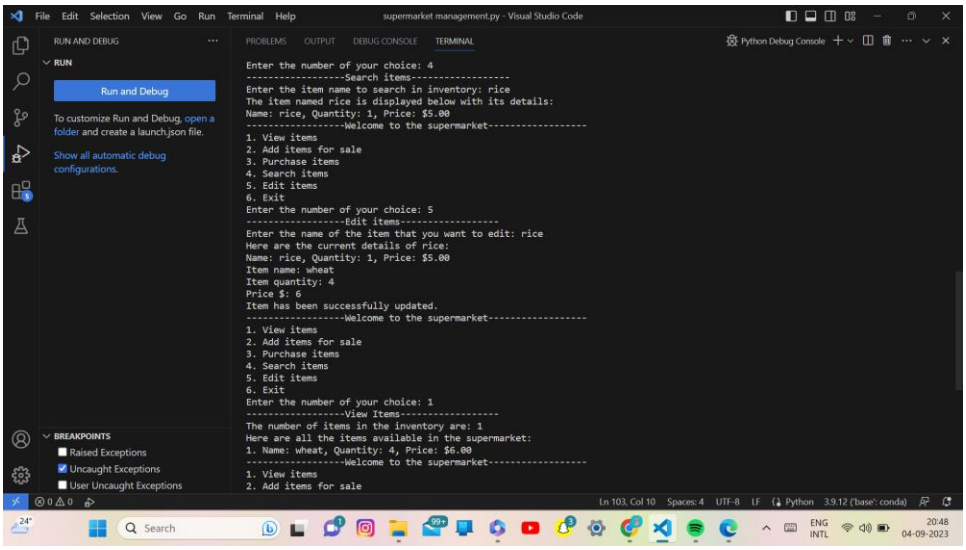
```
            print("Invalid item number")
        except ValueError:
            print('Invalid input. Please enter a valid number.')

    elif choice == '4':
        print('------------------Search items------------------')
        find_item = input('Enter the item name to search in inventory: ').lower()
        found = False
        for item in items:
            if find_item == item['name'].lower():
                print(f'The item named {find_item} is displayed below with its details:')
                print(f'Name: {item["name"]}, Quantity: {item["quantity"]}, Price: ${item["price"]:.2f}')
                found = True
                break
        if not found:
            print('Item not found.')

    elif choice == '5':
        print('------------------Edit items------------------')
        item_name = input('Enter the name of the item that you want to edit: ').lower()
        found = False
        for item in items:
            if item_name == item['name'].lower():
                print(f'Here are the current details of {item_name}:')
                print(f'Name: {item["name"]}, Quantity: {item["quantity"]}, Price: ${item["price"]:.2f}')
                item['name'] = input('Item name: ')
                while True:
                    try:
```

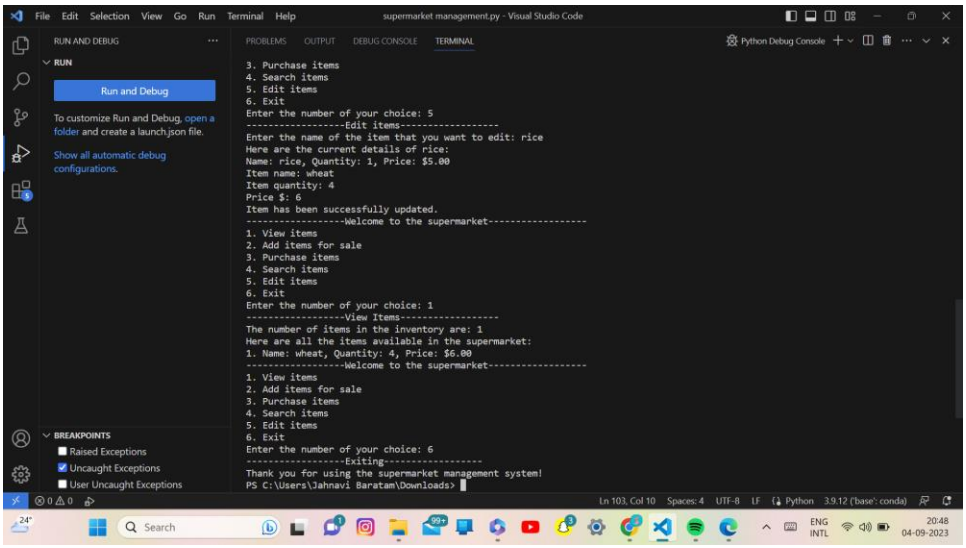


```
                        item['quantity'] = int(input('Item quantity: '))
                        break
                    except ValueError:
                        print('Quantity should only be in digits')
                while True:
                    try:
                        item['price'] = float(input('Price $: '))
                        break
                    except ValueError:
                        print('Price should be a numeric value')
                print('Item has been successfully updated.')
                found = True
                break
        if not found:
            print('Item not found.')

    elif choice == '6':
        print('------------------Exiting------------------')
        break

    else:
        print('You entered an invalid option. Please try again.')

print('Thank you for using the supermarket management system!')
```

THE OUTPUT OF THE MODIFIED CODE:






File Edit Selection View Go Run Terminal Help
supermarket management.py - Visual Studio Code
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
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
3. Purchase items
4. Search items
5. Edit items
6. Exit
Enter the number of your choice: 6
----------Exiting----------
Thank you for using the supermarket management system!
PS C:\Users\Jahnavi Baratam\Downloads>
Ln 103, Col 10 Spaces: 4 UTF-8 LF Python 3.9.12 ('base': conda)
Search
ENG INTL
20:48
04-09-2023