IIT INFORMATICS INSTITUTE OF TECHNOLOGY

RGU ROBERT GORDON UNIVERSITY ABERDEEN

BSc (Hons) Artificial Intelligence and Data Science

Module Code : CM1601

Module Title : Programming Fundamentals

Module Leader : Ms.Sachinthani Perera

Stage : 2023 | January | Intake Semester 1

Assessment Type : Individual

Student ID (IIT) : 20222449

Student ID (RGU) : 2311688

Student Name : Payagalage Rumeth Sandinu

Contents

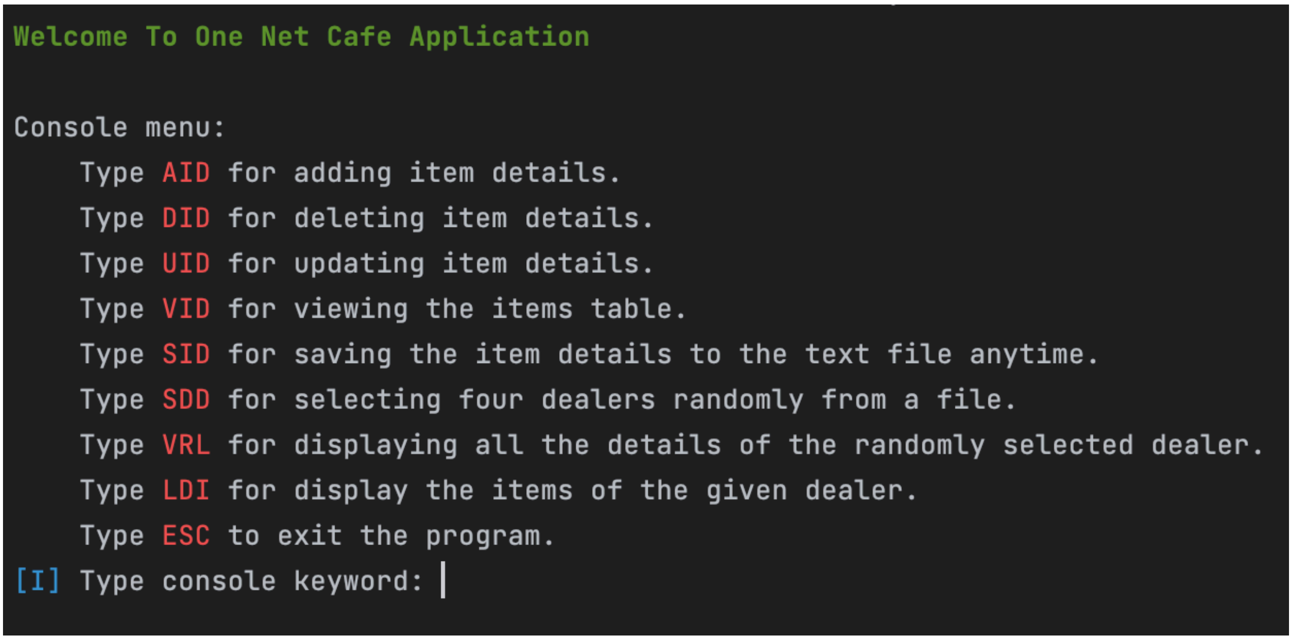
Introduction

Internet Cafes has become a popular business due to the rapid change in the technology and higher demand.Many people, especially students who do not have access to a personal computer or to the internet, go to internet cafes for completing their projects and assignments, and for entertainment.

This application helps the One Net Cafe to maintain business efficiently and smoothly.This application helps to add,delete and update items.Moreover this application will allow user to save data to a text file any time.User can view items in a table,select four dealers randomly,display all the details of the randomly selected dealers and display the items of the entered dealer can be easily done with this application.This application uses three text files to maintain data.

So it will be very easy to manage high volume of data about items by this application.

Application Interafce :



```
Welcome To One Net Cafe Application


Console menu:
    Type AID for adding item details.
    Type DID for deleting item details.
    Type UID for updating item details.
    Type VID for viewing the items table.
    Type SID for saving the item details to the text file anytime.
    Type SDD for selecting four dealers randomly from a file.
    Type VRL for displaying all the details of the randomly selected dealer.
    Type LDI for display the items of the given dealer.
    Type ESC to exit the program.
[I] Type console keyword: 
```

System overview

How The Application Works

This application requests for console keyword and then user have to enter the console keyword to do a specific task.User can input any values anytime if the values are not valid then the application will print an error message for the user and request again for values but the application will not be interrupted anyhow until the user enter ‘ESC’ as console keyword.There will be a validation check for the all values would be entered by the user.

Console menu :

- Type AID for adding item details.

- Type DID for deleting item details.

- Type UID for updating item details.

- Type VID for viewing the items table. (Sort according to the items code descending order) and print the current total.

- Type SID for saving the item details to the text file at any time.

- Type SDD for selecting four dealers randomly from a file.

- Type VRL for displaying all the details of the randomly selected dealers. (Sorted according to the location.)

- Type LDI for display the items of the given dealer.

- Type ESC to exit the program.

Assumptions :

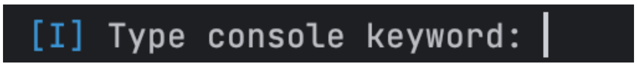
- Assume that user wants to keep Item Code unique.
- Assume that user purchased items after 2022 (Purchase Date >= 2022)
- Assume that Item Code,Item Price,Item Quantity,Item Date can not be null.
- Assume that user do not want to exit the program when he/she is inside a console.(when Application requests for console keyword)
- Assume that user want to save data to a file after adding,deleting or updating items.
- Assume user want to display the items of ther given dealers only entering the ‘LDI’ console keyword not in ‘VRL’ function.
- Assume that user want to keep Item Code same as before when updating item details.
- Assume that user wants not to change order in the text file after deleting or updating data.
- Assume that user wants the quantity of all the items when he input ‘VID’ as console keyword.
- Assume that user want to display the names of randomly selected dealers when he input ‘SDD’ as console keyword.
- Assume that the user want to display the items of any dealer in the text file in ‘LDI’.

User Interface

This application has enchanting user interface with colors and various text formats.This application has very user-friendly interface and it will be very interesting when dealing with this kind of colourful user interface.I have devided all text what will print by this appliocation to five groups.They are,

• Intputs

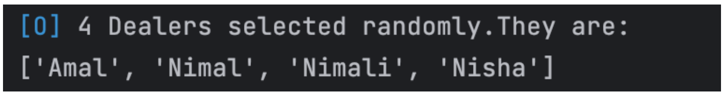
When application requests to get input from the user,there will be a '[I]' mark in blue color.User can see the cursor is blinking after the ':' mark.User can input any value there.



```
[I] Type console keyword: |
```

• Outputs

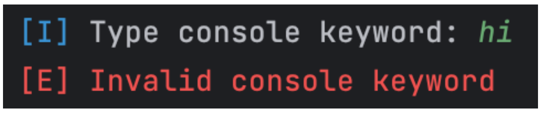
When application prints an output requested by user,there will be a ' [O]' mark in blue color.



```
[O] 4 Dealers selected randomly.They are:
['Amal', 'Nimal', 'Nimali', 'Nisha']
```

• Errors

If the values entered by user fails validation check there will be a '[E]' mark red in color.The error message also in red in color then the user can easily get an idea,that the value got failed in validation check without reading the message.

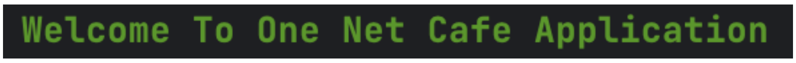


```
[I] Type console keyword: hi
[E] Invalid console keyword
```

• Welcome And Good bye Messages

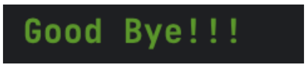
When user starts this application there will be an animation in green color with a worm welcome to the user.When user want to quit the progrom there will be a good bye message with an animation.Outputs are given bellow.

Welcome message :



```
Welcome To One Net Cafe Application
```

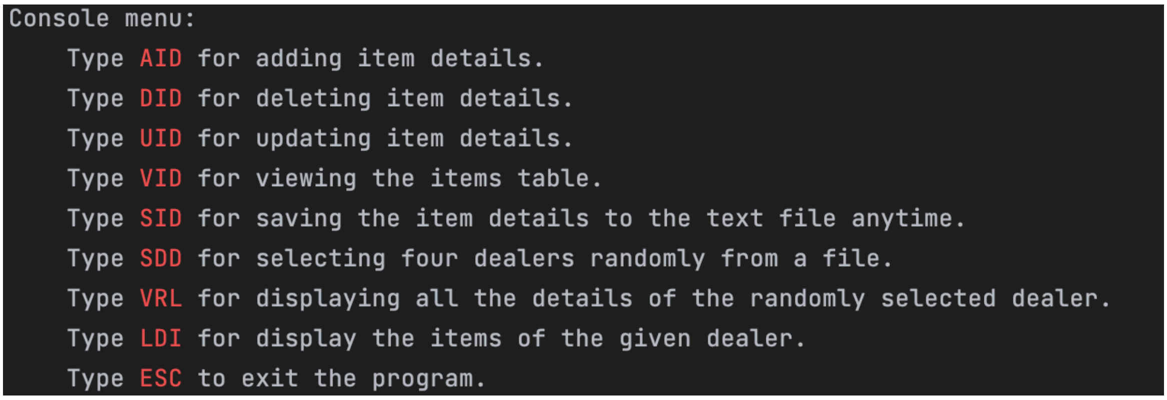
Good bye message :



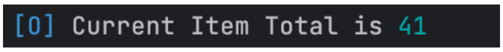
```
Good Bye!!!
```

• Important Text

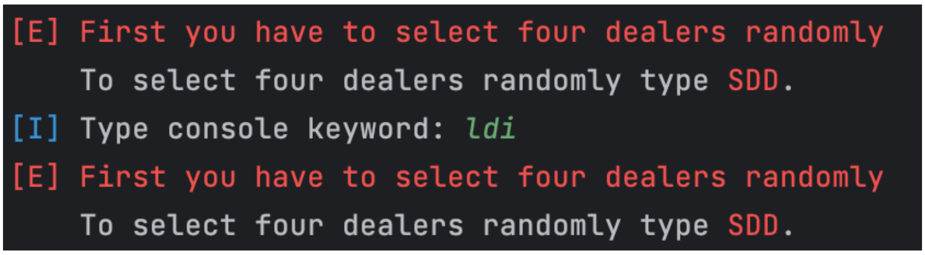
Important words and values are highlighted various with colors to get user attention for them.



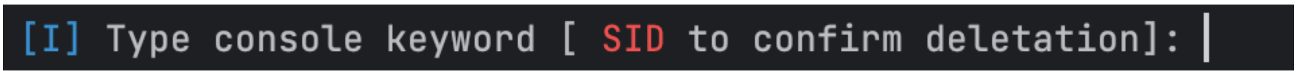
```
Console menu:
    Type AID for adding item details.
    Type DID for deleting item details.
    Type UID for updating item details.
    Type VID for viewing the items table.
    Type SID for saving the item details to the text file anytime.
    Type SDD for selecting four dealers randomly from a file.
    Type VRL for displaying all the details of the randomly selected dealer.
    Type LDI for display the items of the given dealer.
    Type ESC to exit the program.
```



```
[O] Current Item Total is 41
```



```
[E] First you have to select four dealers randomly
    To select four dealers randomly type SDD.
[I] Type console keyword: ldi
[E] First you have to select four dealers randomly
    To select four dealers randomly type SDD.
```



```
[I] Type console keyword [ SID to confirm deletation]:
```

Functionality

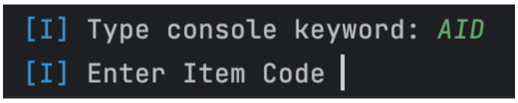
There are nine functions in this application.They are,

- AID
- DID
- UID
- VID
- SID
- SDD
- VRL
- LDI
- ESC

AID

This function allows user to enter Item details to the applications.
After entering ‘AID’ as console keyword this application will ask user to enter the item code.Then the user have to enter the Item Code.

Output :



```
[I] Type console keyword: AID
[I] Enter Item Code
```

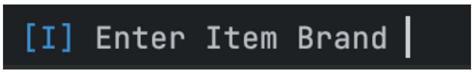
If Item Code is valid then the user will be asked for the Item Name.

Output :

```
[I] Enter Item Name
```

If the Item Name is valid then the user will be asked for the Item Brand.

Output :



```
[I] Enter Item Brand
```

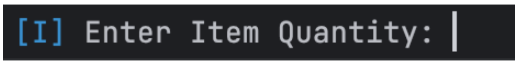
If the Item Brand is valid then the user will be asked for the Item Price.

Output :

```
[I] Enter Item Price:
```

If the Item Price is valid then the user will be asked for the Item Quantity.

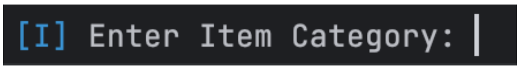
Output :



```
[I] Enter Item Quantity: 
```

If the Item Quantity is valid then the user will be asked for the Item Category.

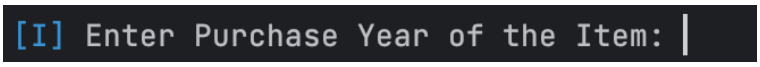
Output :



```
[I] Enter Item Category: 
```

If the Item Category is valid then the user will be asked for the Purchase Year of the Item.

Output :



```
[I] Enter Purchase Year of the Item: 
```

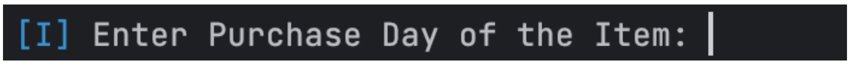
If the Purchase Year of the Item is valid then the user will be asked for the Purchase Month of the Item.

Output :

```
[I] Enter Purchase Month of the Item: 
```

If the Purchase Month of the Item is valid then the user will be asked for the Purchase Day of the Item.

Output :



```
[I] Enter Purchase Day of the Item: 
```

If the Purchase Month of the Item is valid then the user can save the entered data using 'SID' console to the file or can go for another function.

DID

This function allows user to delete a spesic item details saved to the text file.
After entering 'DID' as console keyword,application will ask Item Code of the item details user want to delete.

Output :

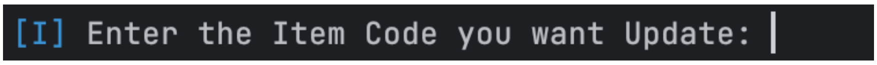
```
[I] Enter the Item Code you want to delete:
```

If the Item Code is valid then user will be asked to confirm the deletion.

UID

This function allows user to update all the details of a spesific item.User will be asked for the Item Code of Item that user want to update details.This will keep Item Code as same as before.

Output :



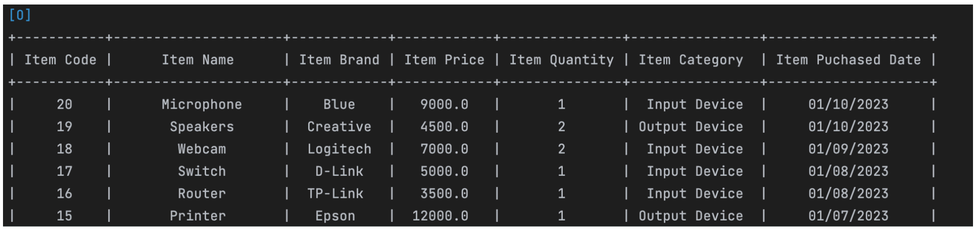
```
[I] Enter the Item Code you want Update: |
```

If the Item Code is valid,then the user will be asked to enter the all item details like ‘AID’ function without Item Code.After entering valid details user will be asked to save updated data in to the text file.

VID

This will print all the items and details in a table sorted according to the Item Code descending order.

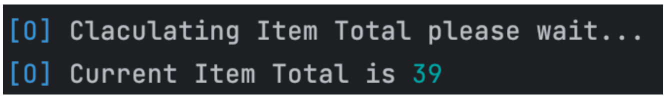
Output :



```
[O]
+-----------+---------------------+------------+------------+---------------+----------------+--------------------+
| Item Code |      Item Name      | Item Brand | Item Price | Item Quantity | Item Category  | Item Puchased Date |
+-----------+---------------------+------------+------------+---------------+----------------+--------------------+
|    20     |     Microphone      |    Blue    |   9000.0   |       1       |  Input Device  |     01/10/2023     |
|    19     |      Speakers       |  Creative  |   4500.0   |       2       | Output Device  |     01/10/2023     |
|    18     |       Webcam        |  Logitech  |   7000.0   |       2       |  Input Device  |     01/09/2023     |
|    17     |       Switch        |   D-Link   |   5000.0   |       1       |  Input Device  |     01/08/2023     |
|    16     |       Router        |  TP-Link   |   3500.0   |       1       |  Input Device  |     01/08/2023     |
|    15     |       Printer       |   Epson    |  12000.0   |       1       | Output Device  |     01/07/2023     |
```

After that this will print the total item quantity.

Output :



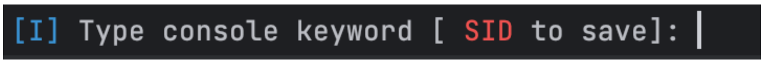
```
[O] Claculating Item Total please wait...
[O] Current Item Total is 39
```

The item quantity will be printed in cyan color to make it easily readable.

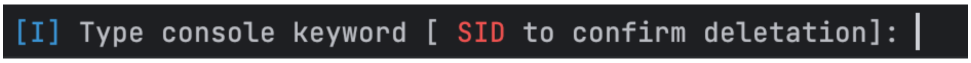
SID

User can save data using SID after entering,deleting or updating data.

Output :



```
[I] Type console keyword [ SID to save]: |
```

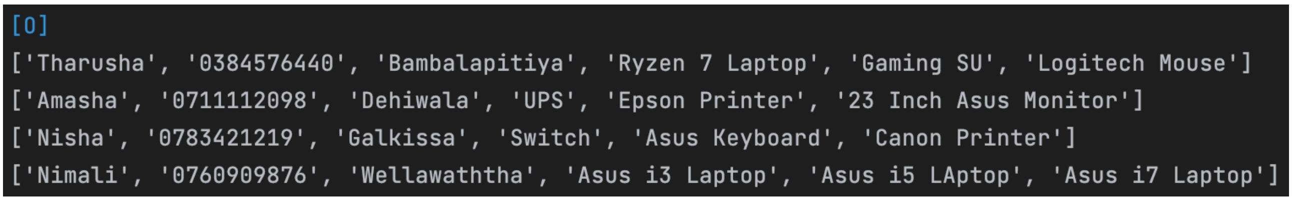


```
[I] Type console keyword [ SID to confirm deletation]: |
```

SDD

This function will select four dealers randomly from the Dealers.txt file and will display the names of the four dealers.

Output :

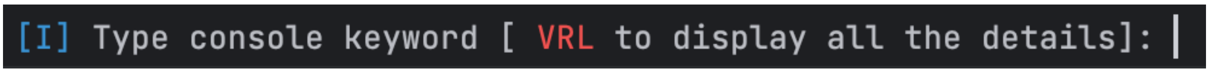


```
[O]
['Tharusha', '0384576440', 'Bambalapitiya', 'Ryzen 7 Laptop', 'Gaming SU', 'Logitech Mouse']
['Amasha', '0711112098', 'Dehiwala', 'UPS', 'Epson Printer', '23 Inch Asus Monitor']
['Nisha', '0783421219', 'Galkissa', 'Switch', 'Asus Keyboard', 'Canon Printer']
['Nimali', '0760909876', 'Wellawaththa', 'Asus i3 Laptop', 'Asus i5 LAptop', 'Asus i7 Laptop']
```

VRL

This function will display all the details of the randomly selected dealers.

Output :

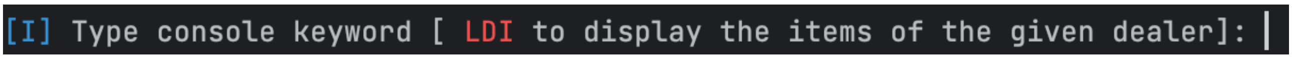


```
[I] Type console keyword [ VRL to display all the details]: 
```

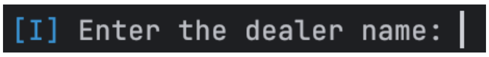
LDI

This funtion will ask user for the name of the dealer.If the name is valid this will print three items that the dealer sells

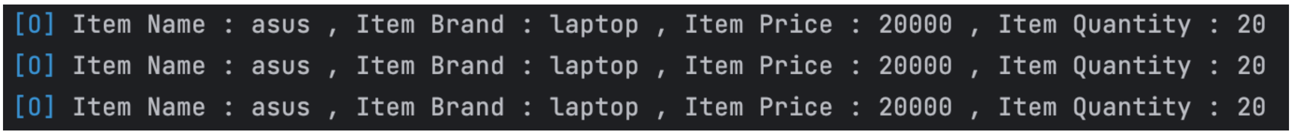
Output :



```
[I] Type console keyword [ LDI to display the items of the given dealer]: 
```



```
[I] Enter the dealer name: 
```

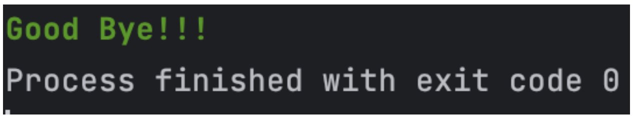


```
[O] Item Name : asus , Item Brand : laptop , Item Price : 20000 , Item Quantity : 20
[O] Item Name : asus , Item Brand : laptop , Item Price : 20000 , Item Quantity : 20
[O] Item Name : asus , Item Brand : laptop , Item Price : 20000 , Item Quantity : 20
```

ESC

This function will print ‘Good Bye!!!’ and terminates the program.

Output :



```
Good Bye!!!
Process finished with exit code 0
```

Testing And Validation

This applicaion has been tested many many times.User can get smooth experience by working on this application.All the values would be entered by the user will be validated.If the value is invalid then the application will print an error message with ‘[E]’ tag red in color and will ask again for the value again.

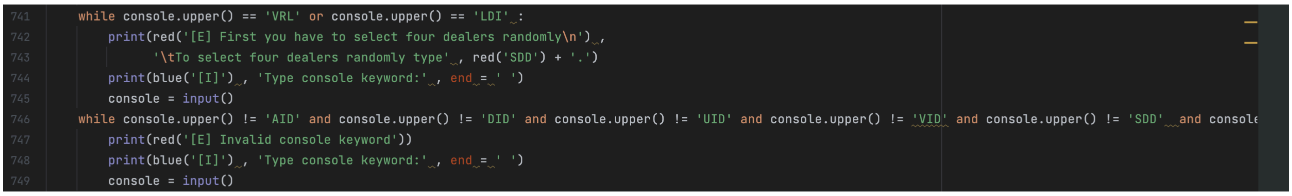
There will be validation check for every value enterd by the user.

- Console keyword
- AID
- DID
- UID
- LDI

Console Keyword

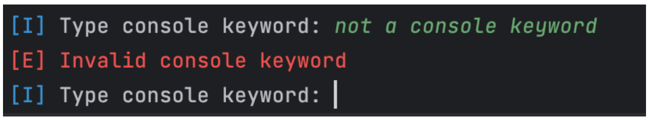
After starting the system or after finishing any function the application will ask the user to enter a console keyword.If the value entered by user is not a valid console keyword,then it will print an error message and will ask to enter a console keyword again.

Code :



```
    while console.upper() == 'VRL' or console.upper() == 'LDI' :
        print(red('[E] First you have to select four dealers randomly\n') ,
              '\tTo select four dealers randomly type' , red('SDD') + '.')
        print(blue('[I]') , 'Type console keyword:' , end = ' ')
        console = input()
    while console.upper() != 'AID' and console.upper() != 'DID' and console.upper() != 'UID' and console.upper() != 'VID' and console.upper() != 'SDD'  and console
        print(red('[E] Invalid console keyword'))
        print(blue('[I]') , 'Type console keyword:' , end = ' ')
        console = input()
```

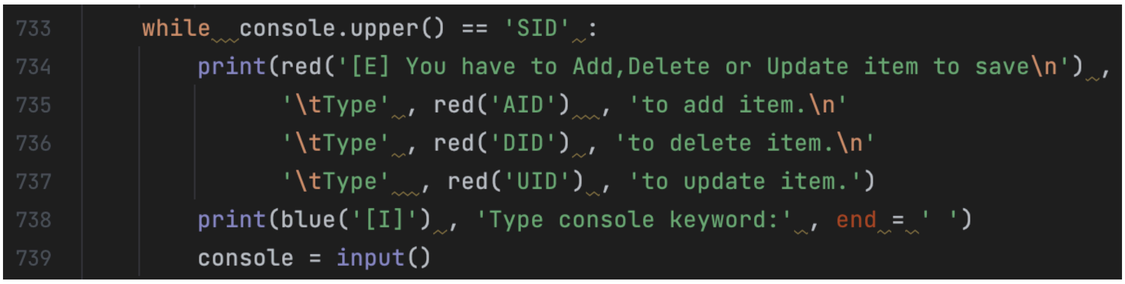
Output :



```
[I] Type console keyword: not a console keyword
[E] Invalid console keyword
[I] Type console keyword: 
```

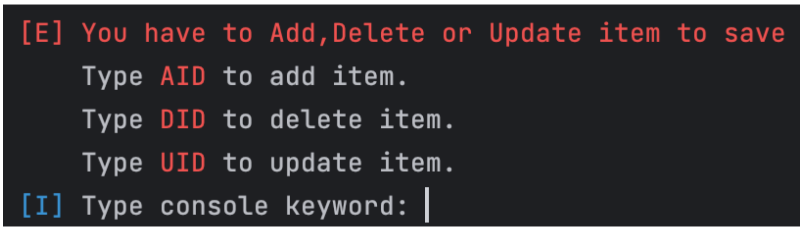
If user enter ‘SID’ console without adding,deleting or updating data,application will print an error message and will ask for console keyword.

Code :



```
    while  console.upper() == 'SID' :
        print(red('[E] You have to Add,Delete or Update item to save\n') ,
              '\tType' , red('AID')  , 'to add item.\n'
              '\tType' , red('DID') , 'to delete item.\n'
              '\tType'  , red('UID') , 'to update item.')
        print(blue('[I]') , 'Type console keyword:' , end = ' ')
        console = input()
```

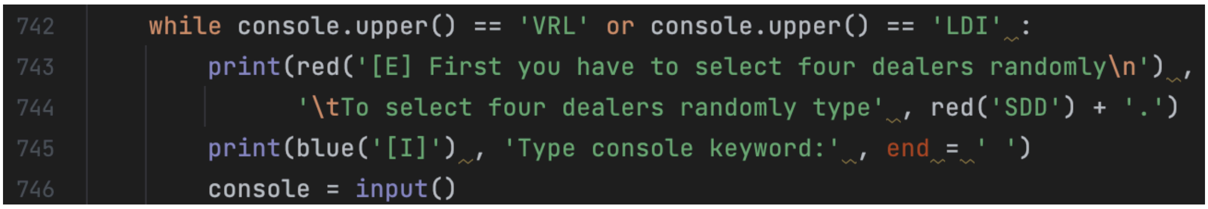
Output :



```
[E] You have to Add,Delete or Update item to save
    Type AID to add item.
    Type DID to delete item.
    Type UID to update item.
[I] Type console keyword: 
```

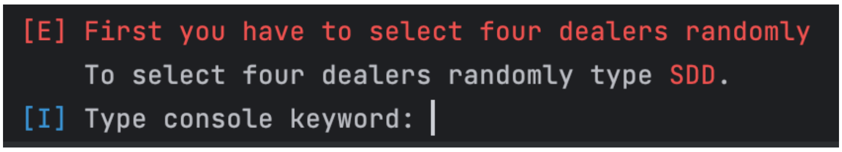
If user enter ‘VRL’ or ‘LDI’ console without randomly selecting dealers,application will print an error message and will ask for console keyword.

Code :



```
    while console.upper() == 'VRL' or console.upper() == 'LDI' :
        print(red('[E] First you have to select four dealers randomly\n') ,
              '\tTo select four dealers randomly type' , red('SDD') + '.')
        print(blue('[I]') , 'Type console keyword:' , end = ' ')
        console = input()
```

Output :



```
[E] First you have to select four dealers randomly
    To select four dealers randomly type SDD.
[I] Type console keyword: 
```

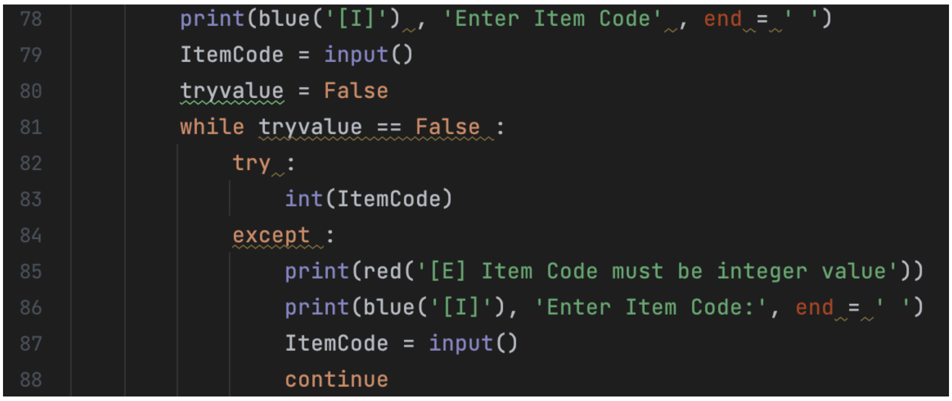
AID

There will be a validation check for each input in AID function.

Item Code

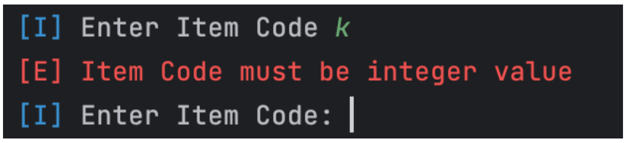
If the Item Code is not an integer the application will print an error message and will ask for an Item Code again.

Code :



```
        print(blue('[I]') , 'Enter Item Code' , end = ' ')
        ItemCode = input()
        tryvalue = False
        while tryvalue == False :
            try :
                int(ItemCode)
            except :
                print(red('[E] Item Code must be integer value'))
                print(blue('[I]'), 'Enter Item Code:', end = ' ')
                ItemCode = input()
                continue
```

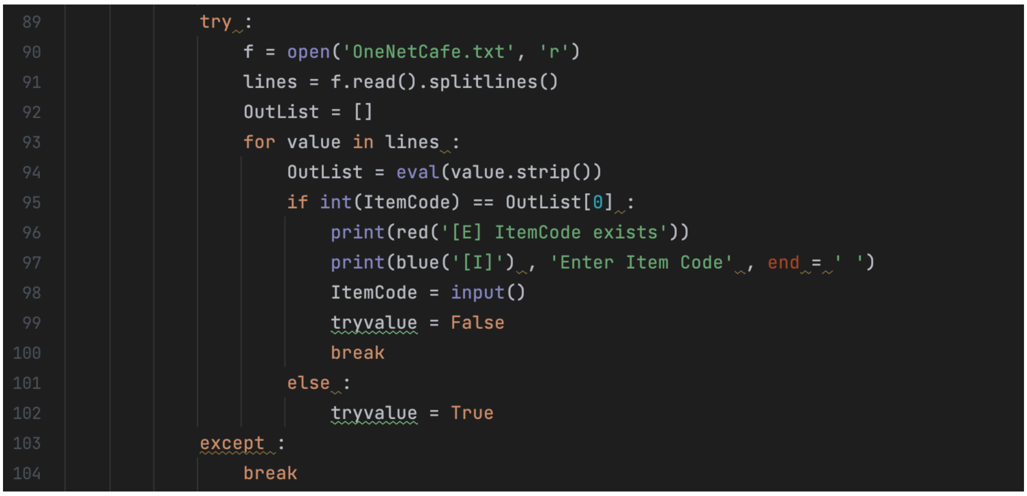
Output :



```
[I] Enter Item Code k
[E] Item Code must be integer value
[I] Enter Item Code: 
```

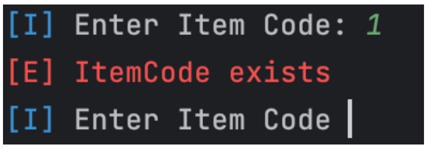
If the Item Code is an integer but it is already exists in the text file,then application will print an error message and it will ask for an Item Code again.

Code :



```
            try :
                f = open('OneNetCafe.txt', 'r')
                lines = f.read().splitlines()
                OutList = []
                for value in lines :
                    OutList = eval(value.strip())
                    if int(ItemCode) == OutList[0] :
                        print(red('[E] ItemCode exists'))
                        print(blue('[I]') , 'Enter Item Code' , end = ' ')
                        ItemCode = input()
                        tryvalue = False
                        break
                    else :
                        tryvalue = True
            except :
                break
```

Output :

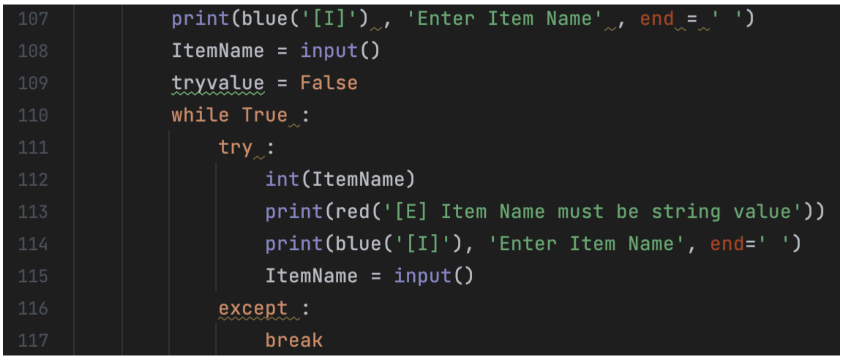


```
[I] Enter Item Code: 1
[E] ItemCode exists
[I] Enter Item Code 
```

Item Name

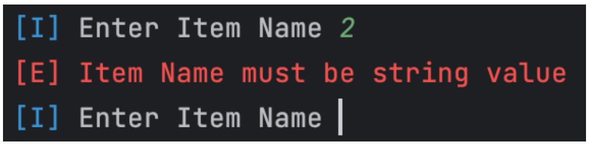
If the Item Code is valid then the application will ask for Item Name.If the Item Name is not an string value the application will print an error message and will ask for an Item Name.

Code :



```
        print(blue('[I]') , 'Enter Item Name' , end = ' ')
        ItemName = input()
        tryvalue = False
        while True :
            try :
                int(ItemName)
                print(red('[E] Item Name must be string value'))
                print(blue('[I]'), 'Enter Item Name', end=' ')
                ItemName = input()
            except :
                break
```

Output :

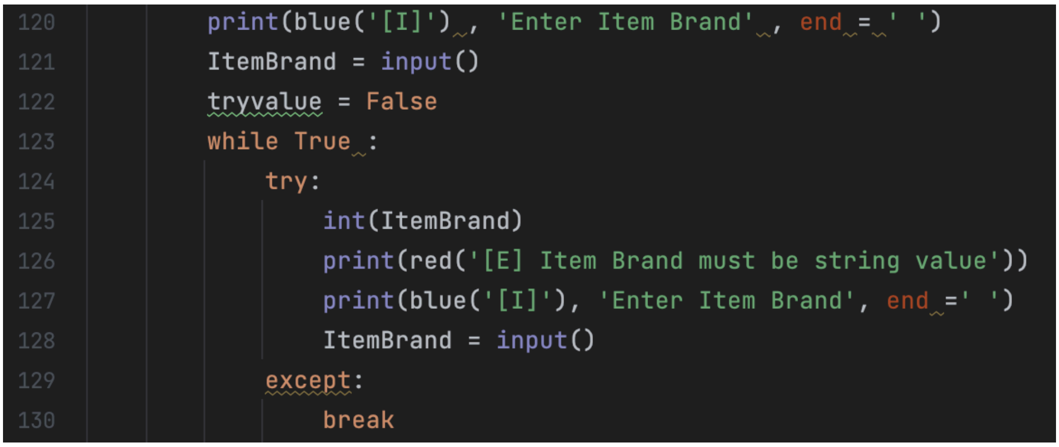


```
[I] Enter Item Name 2
[E] Item Name must be string value
[I] Enter Item Name
```

Item Brand

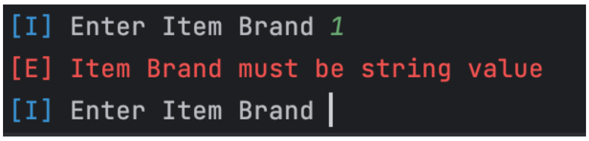
If the Item Name is valid then the application will ask for Item Brand.Item Brand must be an integer value.If not application will print an error message and will ask for an Item Brand.

Code :



```
        print(blue('[I]') , 'Enter Item Brand' , end = ' ')
        ItemBrand = input()
        tryvalue = False
        while True :
            try:
                int(ItemBrand)
                print(red('[E] Item Brand must be string value'))
                print(blue('[I]'), 'Enter Item Brand', end =' ')
                ItemBrand = input()
            except:
                break
```

Output :

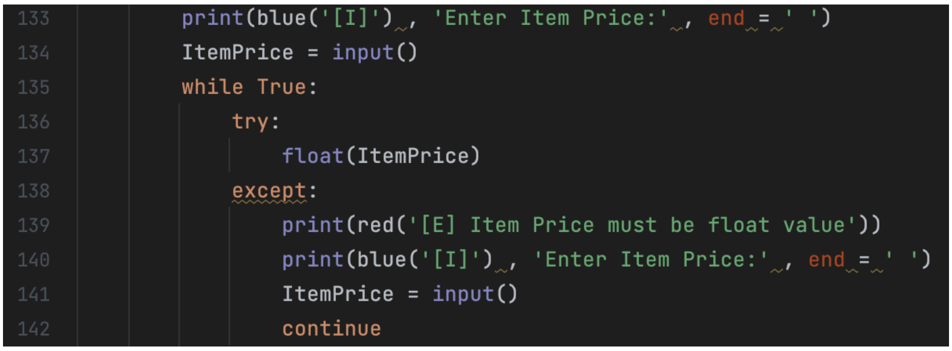


```
[I] Enter Item Brand 1
[E] Item Brand must be string value
[I] Enter Item Brand
```

Item Price

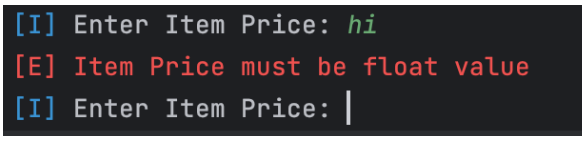
If the Item Brand is valid then the application will ask for Item Price.Item Price must be a float or integer value.If not application will print an error message and will ask for an Item Price.

Code :



```
        print(blue('[I]') , 'Enter Item Price:' , end = ' ')
        ItemPrice = input()
        while True:
            try:
                float(ItemPrice)
            except:
                print(red('[E] Item Price must be float value'))
                print(blue('[I]') , 'Enter Item Price:' , end = ' ')
                ItemPrice = input()
                continue
```

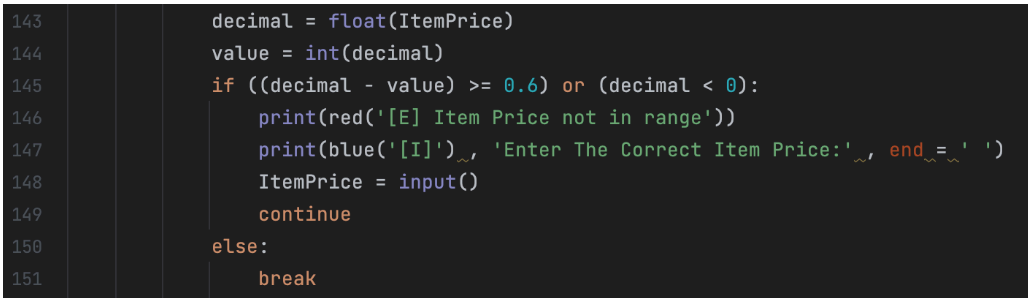
Output :



```
[I] Enter Item Price: hi
[E] Item Price must be float value
[I] Enter Item Price: |
```

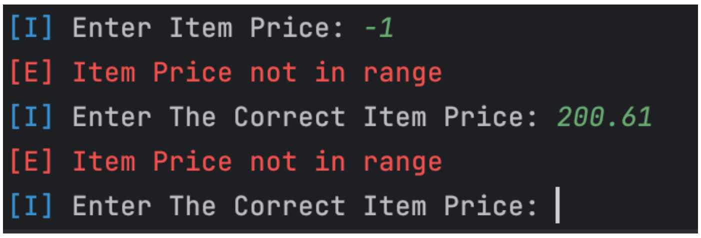
If the Item Price is an integer then the application will check weather the Item Price is greater than or equal to ‘0’ and the Decimal value is less than ‘.60’.If not application will print an error message and will ask for Item Price.

Code :



```python
            decimal = float(ItemPrice)
            value = int(decimal)
            if ((decimal - value) >= 0.6) or (decimal < 0):
                print(red('[E] Item Price not in range'))
                print(blue('[I]') , 'Enter The Correct Item Price:' , end = ' ')
                ItemPrice = input()
                continue
            else:
                break
```

Output :

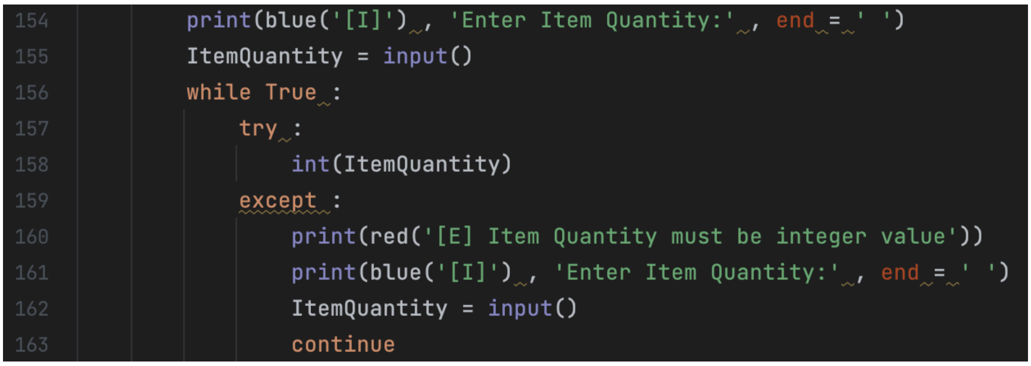


```
[I] Enter Item Price: -1
[E] Item Price not in range
[I] Enter The Correct Item Price: 200.61
[E] Item Price not in range
[I] Enter The Correct Item Price: |
```

Item Quantity

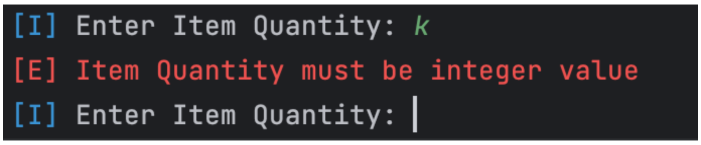
If the Item Price is valid then the application will ask for Item Quantity.Item Quantity must be an integer value.If not application will print an error message and will ask for an Item Quantity.

Code :



```python
        print(blue('[I]') , 'Enter Item Quantity:' , end = ' ')
        ItemQuantity = input()
        while True :
            try :
                int(ItemQuantity)
            except :
                print(red('[E] Item Quantity must be integer value'))
                print(blue('[I]') , 'Enter Item Quantity:' , end = ' ')
                ItemQuantity = input()
                continue
```

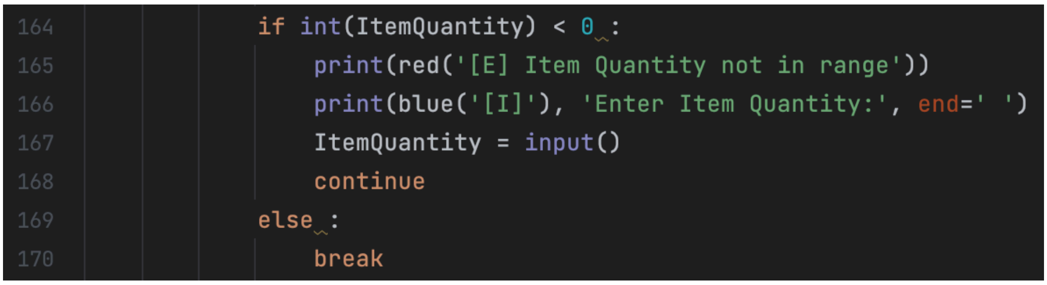
Output :



```
[I] Enter Item Quantity: k
[E] Item Quantity must be integer value
[I] Enter Item Quantity: 
```

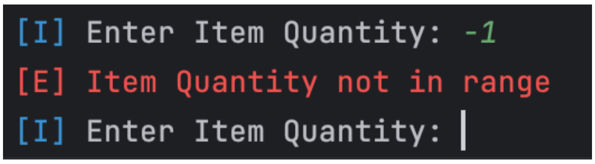
If the Item Quantity is an integer,then the application will check if the Quantity is greater than or equal ‘0’ if not application will print an error and will ask for Item Quantity.

Code :



```
            if int(ItemQuantity) < 0 :
                print(red('[E] Item Quantity not in range'))
                print(blue('[I]'), 'Enter Item Quantity:', end=' ')
                ItemQuantity = input()
                continue
            else :
                break
```

Output :

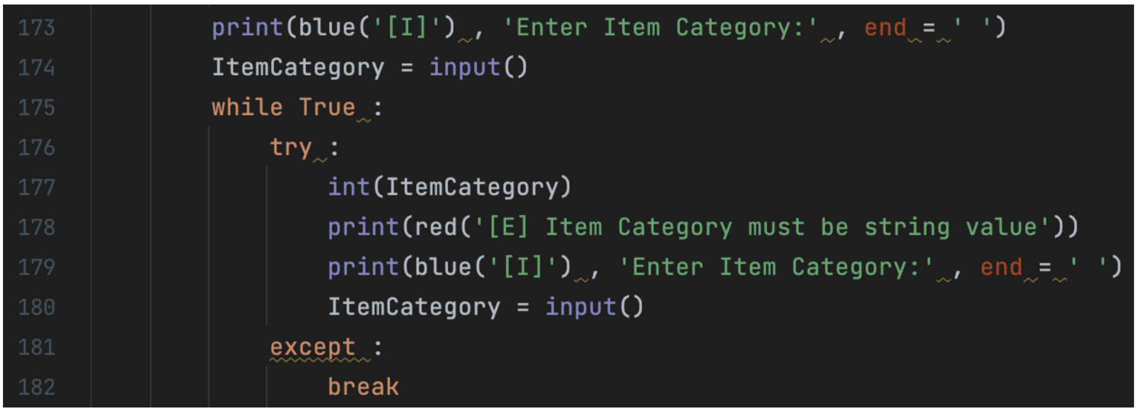


```
[I] Enter Item Quantity: -1
[E] Item Quantity not in range
[I] Enter Item Quantity: 
```

Item Category

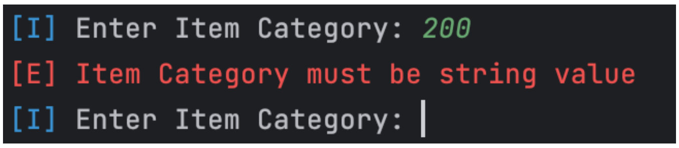
If the Item Quantity is valid then the application will ask for Item Category.Item Category must be a string value.If not application will print an error message and will ask for an Item Category.

Code :



```
        print(blue('[I]') , 'Enter Item Category:' , end = ' ')
        ItemCategory = input()
        while True :
            try :
                int(ItemCategory)
                print(red('[E] Item Category must be string value'))
                print(blue('[I]') , 'Enter Item Category:' , end = ' ')
                ItemCategory = input()
            except :
                break
```

Output :

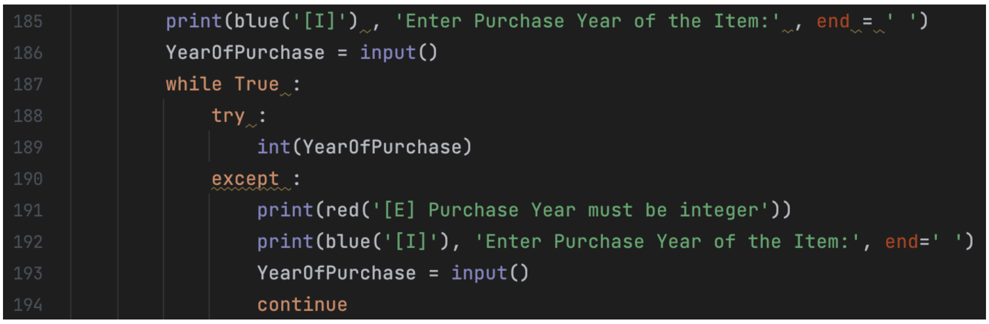


```
[I] Enter Item Category: 200
[E] Item Category must be string value
[I] Enter Item Category: |
```

Purchase Year of the Item

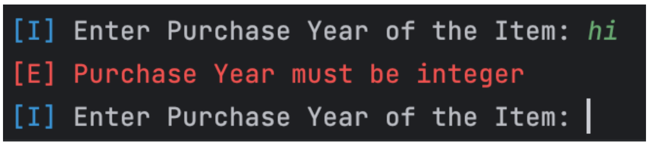
If the Item Quantity is valid, the the application will ask for the Purchase Year of the Item.Purchase Year must be an integer value if not.application will print an error messsage and will ask for Purchase Year.

Code :



```python
        print(blue('[I]') , 'Enter Purchase Year of the Item:' , end = ' ')
        YearOfPurchase = input()
        while True :
            try :
                int(YearOfPurchase)
            except :
                print(red('[E] Purchase Year must be integer'))
                print(blue('[I]'), 'Enter Purchase Year of the Item:', end=' ')
                YearOfPurchase = input()
                continue
```

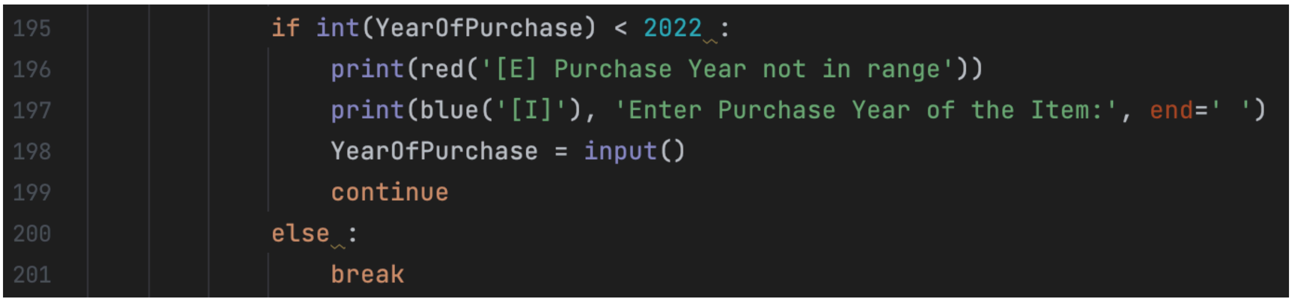
Output :



```
[I] Enter Purchase Year of the Item: hi
[E] Purchase Year must be integer
[I] Enter Purchase Year of the Item: |
```

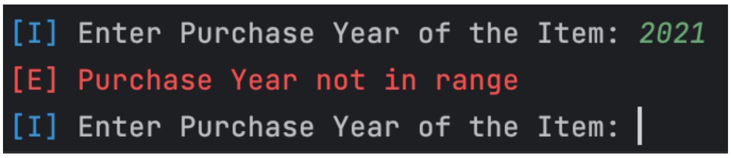
If the Purchase Year is an integer value,then the application will check weather the Purchase Year greater than or equal to 2022.If not,application will print an error message and will ask for Purchase Year.

Code :



```python
            if int(YearOfPurchase) < 2022 :
                print(red('[E] Purchase Year not in range'))
                print(blue('[I]'), 'Enter Purchase Year of the Item:', end=' ')
                YearOfPurchase = input()
                continue
            else :
                break
```

Output :

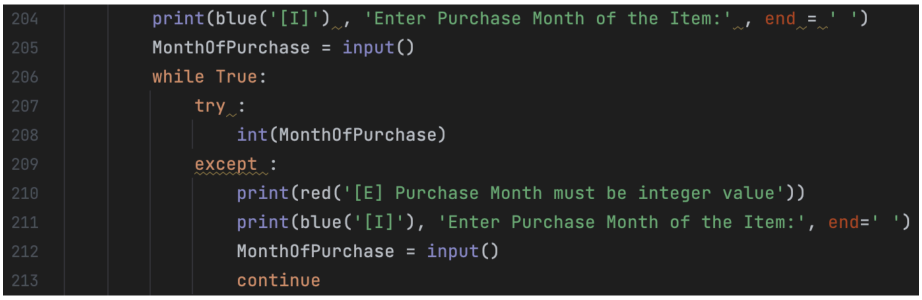


```
[I] Enter Purchase Year of the Item: 2021
[E] Purchase Year not in range
[I] Enter Purchase Year of the Item: 
```

Purchase Month of the Item

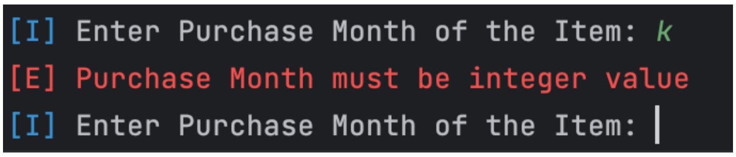
If the Purchase year is valid,then the application will ask for Purchase Month.Purchase month must be an integer.If not,application will print an error and will ask for Purchase Month.

Code :



```
        print(blue('[I]') , 'Enter Purchase Month of the Item:' , end = ' ')
        MonthOfPurchase = input()
        while True:
            try :
                int(MonthOfPurchase)
            except :
                print(red('[E] Purchase Month must be integer value'))
                print(blue('[I]'), 'Enter Purchase Month of the Item:', end=' ')
                MonthOfPurchase = input()
                continue
```

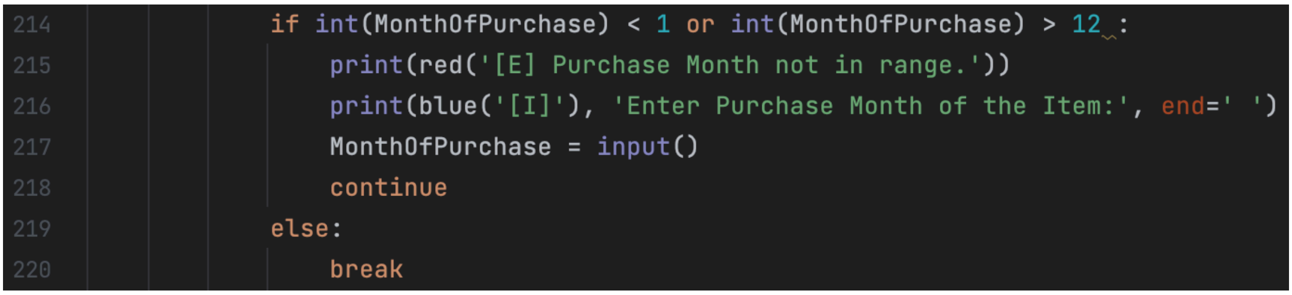
Output



```
[I] Enter Purchase Month of the Item: k
[E] Purchase Month must be integer value
[I] Enter Purchase Month of the Item: 
```

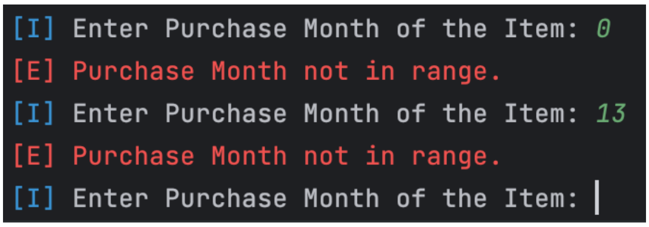
If the Purchase Month is an integer,then the application will check weather the Month greater than ‘0’ and less than ’13’.If not,application will print an error and will ask for Purchase Month.

Code :



```
            if int(MonthOfPurchase) < 1 or int(MonthOfPurchase) > 12 :
                print(red('[E] Purchase Month not in range.'))
                print(blue('[I]'), 'Enter Purchase Month of the Item:', end=' ')
                MonthOfPurchase = input()
                continue
            else:
                break
```

Output :

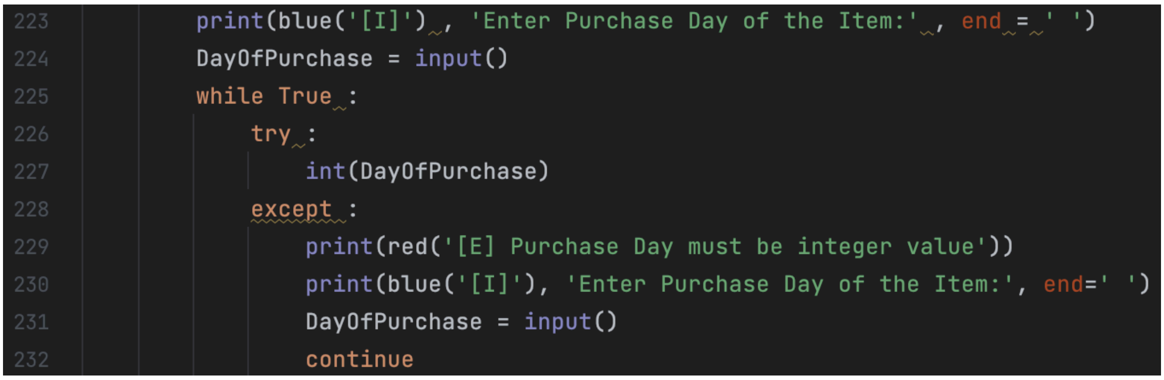


```
[I] Enter Purchase Month of the Item: 0
[E] Purchase Month not in range.
[I] Enter Purchase Month of the Item: 13
[E] Purchase Month not in range.
[I] Enter Purchase Month of the Item: 
```

Purchase Day of the Item

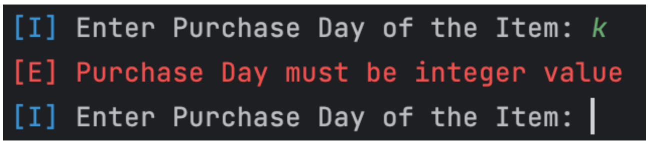
If the Purchase month is valid,then the application will ask for Purchase Day.Purchase Day must be integer.If not application will print an error and will ask for Purchase Day.

Code :



```python
print(blue('[I]') , 'Enter Purchase Day of the Item:' , end = ' ')
DayOfPurchase = input()
while True :
    try :
        int(DayOfPurchase)
    except :
        print(red('[E] Purchase Day must be integer value'))
        print(blue('[I]'), 'Enter Purchase Day of the Item:', end=' ')
        DayOfPurchase = input()
        continue
```

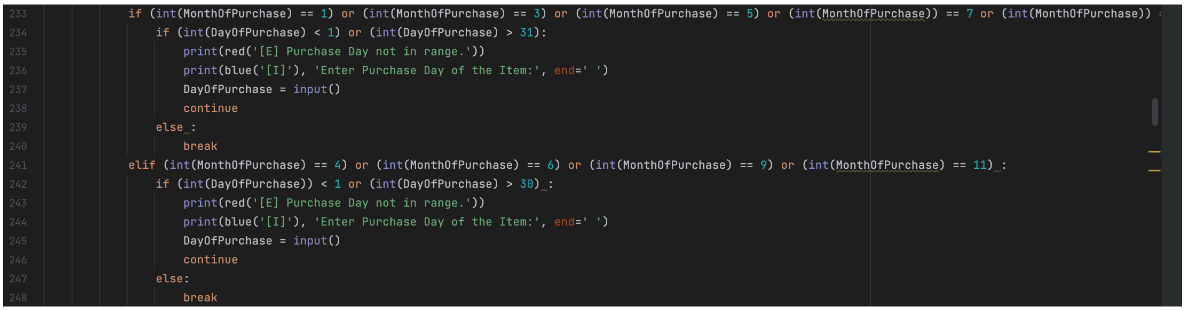
Output :



```
[I] Enter Purchase Day of the Item: k
[E] Purchase Day must be integer value
[I] Enter Purchase Day of the Item: 
```

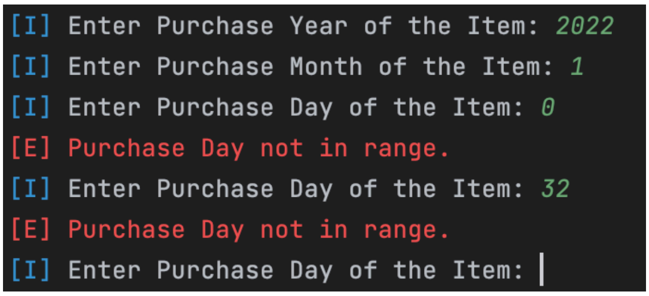
If the Purchase Day is an integer,then the application will check weather the month has 30 or 31 days.If the month is February then it will check weather it is a full year.If it is a full year the application will check the Purchase day in 0 - 29 range if not it will check the Purchase Day in 0 - 28 range and all days must be greater than 0.If not it will print an error and ask for Purchase Day.

Code :

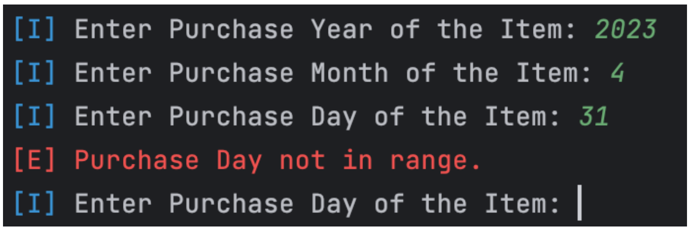


```python
if (int(MonthOfPurchase) == 1) or (int(MonthOfPurchase) == 3) or (int(MonthOfPurchase) == 5) or (int(MonthOfPurchase)) == 7 or (int(MonthOfPurchase))
    if (int(DayOfPurchase) < 1) or (int(DayOfPurchase) > 31):
        print(red('[E] Purchase Day not in range.'))
        print(blue('[I]'), 'Enter Purchase Day of the Item:', end=' ')
        DayOfPurchase = input()
        continue
    else :
        break
elif (int(MonthOfPurchase) == 4) or (int(MonthOfPurchase) == 6) or (int(MonthOfPurchase) == 9) or (int(MonthOfPurchase) == 11) :
    if (int(DayOfPurchase)) < 1 or (int(DayOfPurchase) > 30) :
        print(red('[E] Purchase Day not in range.'))
        print(blue('[I]'), 'Enter Purchase Day of the Item:', end=' ')
        DayOfPurchase = input()
        continue
    else:
        break
```

Output :

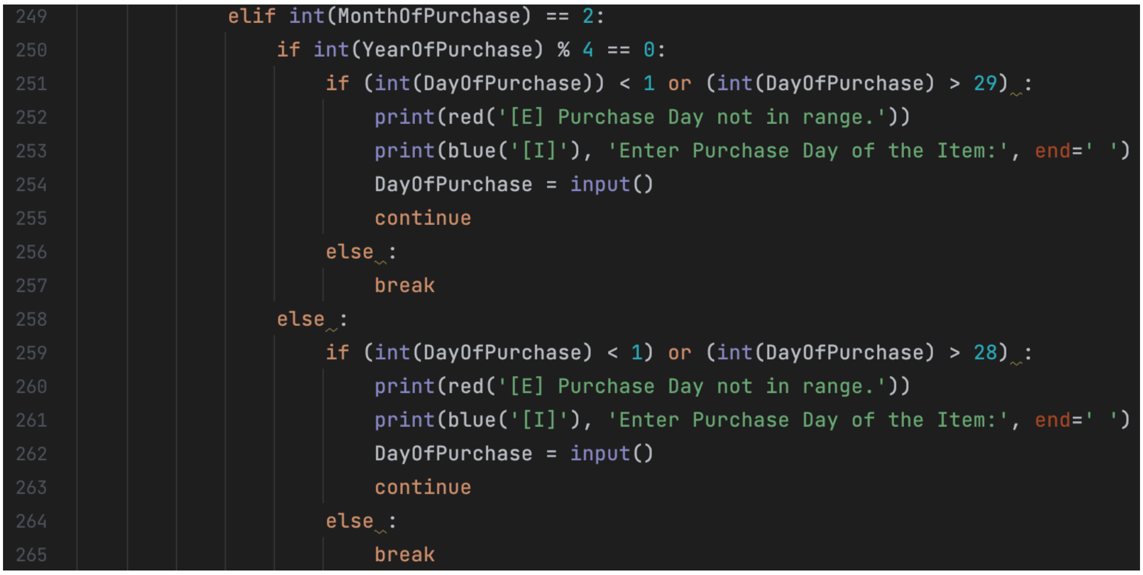


```
[I] Enter Purchase Year of the Item: 2022
[I] Enter Purchase Month of the Item: 1
[I] Enter Purchase Day of the Item: 0
[E] Purchase Day not in range.
[I] Enter Purchase Day of the Item: 32
[E] Purchase Day not in range.
[I] Enter Purchase Day of the Item: 
```



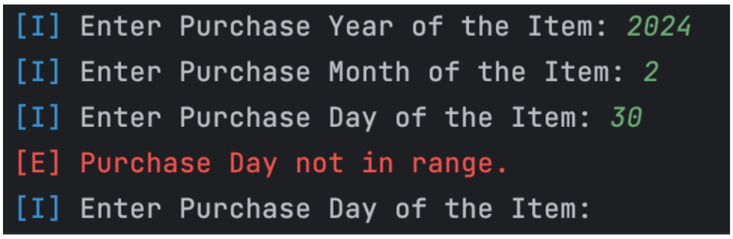
```
[I] Enter Purchase Year of the Item: 2023
[I] Enter Purchase Month of the Item: 4
[I] Enter Purchase Day of the Item: 31
[E] Purchase Day not in range.
[I] Enter Purchase Day of the Item: 
```

Code :

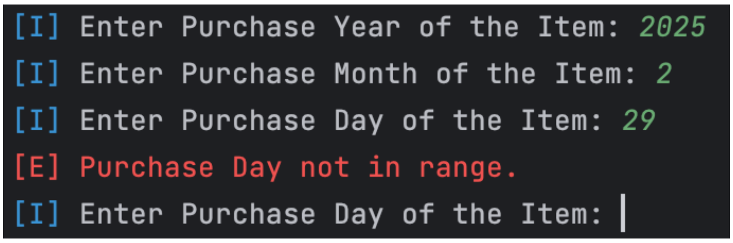


```python
            elif int(MonthOfPurchase) == 2:
                if int(YearOfPurchase) % 4 == 0:
                    if (int(DayOfPurchase)) < 1 or (int(DayOfPurchase) > 29) :
                        print(red('[E] Purchase Day not in range.'))
                        print(blue('[I]'), 'Enter Purchase Day of the Item:', end=' ')
                        DayOfPurchase = input()
                        continue
                    else :
                        break
                else :
                    if (int(DayOfPurchase) < 1) or (int(DayOfPurchase) > 28) :
                        print(red('[E] Purchase Day not in range.'))
                        print(blue('[I]'), 'Enter Purchase Day of the Item:', end=' ')
                        DayOfPurchase = input()
                        continue
                    else :
                        break
```

Output :



```
[I] Enter Purchase Year of the Item: 2024
[I] Enter Purchase Month of the Item: 2
[I] Enter Purchase Day of the Item: 30
[E] Purchase Day not in range.
[I] Enter Purchase Day of the Item:
```



```
[I] Enter Purchase Year of the Item: 2025
[I] Enter Purchase Month of the Item: 2
[I] Enter Purchase Day of the Item: 29
[E] Purchase Day not in range.
[I] Enter Purchase Day of the Item: 
```

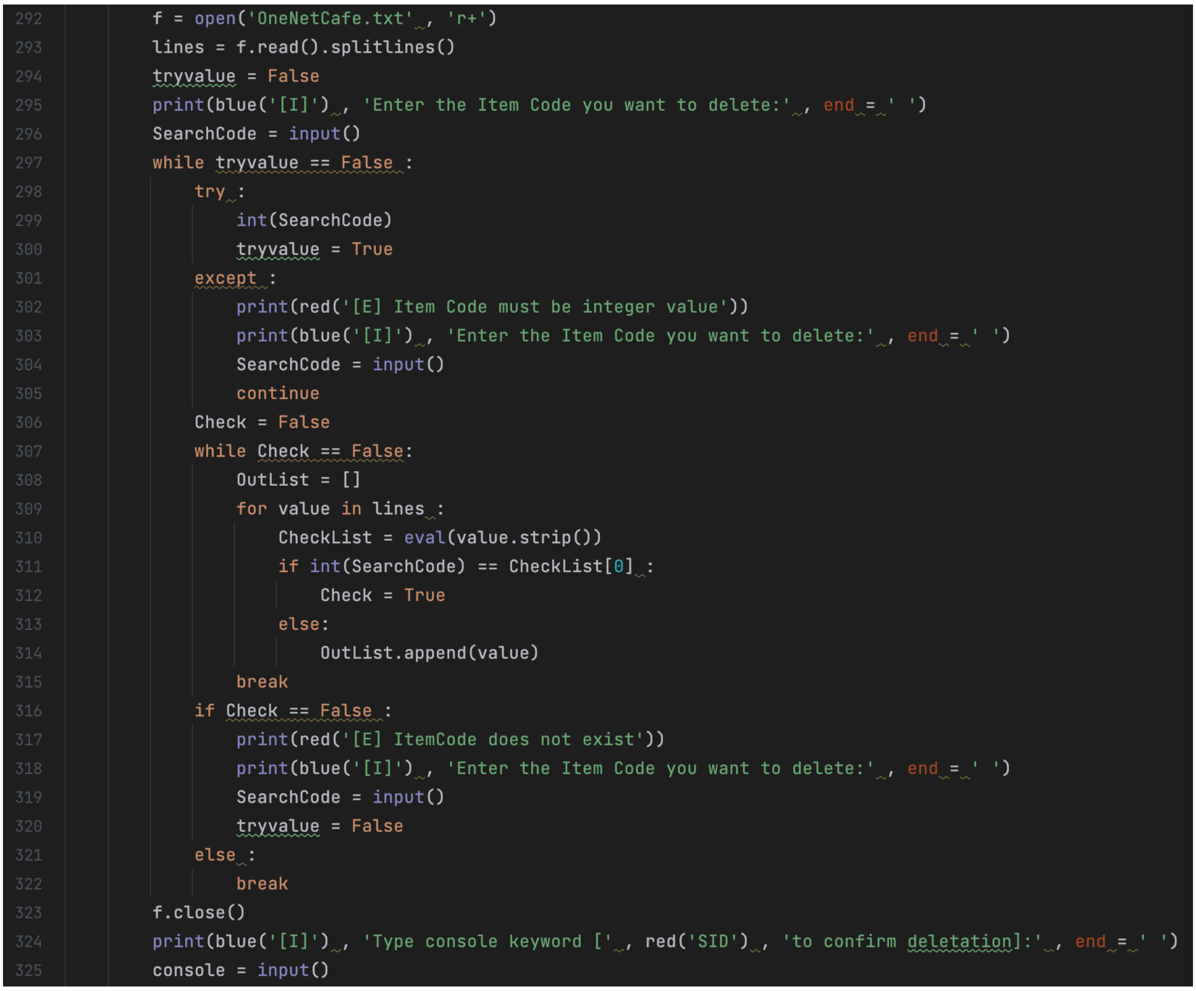
DID

There will be a validation check for each input in AID function.

Item Code

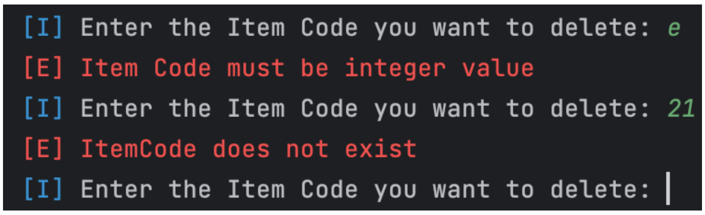
After user entering the Item Code,the application will check weather the Item code is an integer value.If not it will print an error message and will ask for Item Code.If the Item Code is valid then it will check weather the Item Code is in the saved file.If not it will print an error message and will ask for Item Code.

Code :



```
        f = open('OneNetCafe.txt' , 'r+')
        lines = f.read().splitlines()
        tryvalue = False
        print(blue('[I]') , 'Enter the Item Code you want to delete:' , end = ' ')
        SearchCode = input()
        while tryvalue == False :
            try :
                int(SearchCode)
                tryvalue = True
            except :
                print(red('[E] Item Code must be integer value'))
                print(blue('[I]') , 'Enter the Item Code you want to delete:' , end = ' ')
                SearchCode = input()
                continue
            Check = False
            while Check == False:
                OutList = []
                for value in lines :
                    CheckList = eval(value.strip())
                    if int(SearchCode) == CheckList[0] :
                        Check = True
                    else:
                        OutList.append(value)
                break
            if Check == False :
                print(red('[E] ItemCode does not exist'))
                print(blue('[I]') , 'Enter the Item Code you want to delete:' , end = ' ')
                SearchCode = input()
                tryvalue = False
            else :
                break
        f.close()
        print(blue('[I]') , 'Type console keyword [' , red('SID') , 'to confirm deletation]:' , end = ' ')
        console = input()
```

Output :



```
[I] Enter the Item Code you want to delete: e
[E] Item Code must be integer value
[I] Enter the Item Code you want to delete: 21
[E] ItemCode does not exist
[I] Enter the Item Code you want to delete: 
```

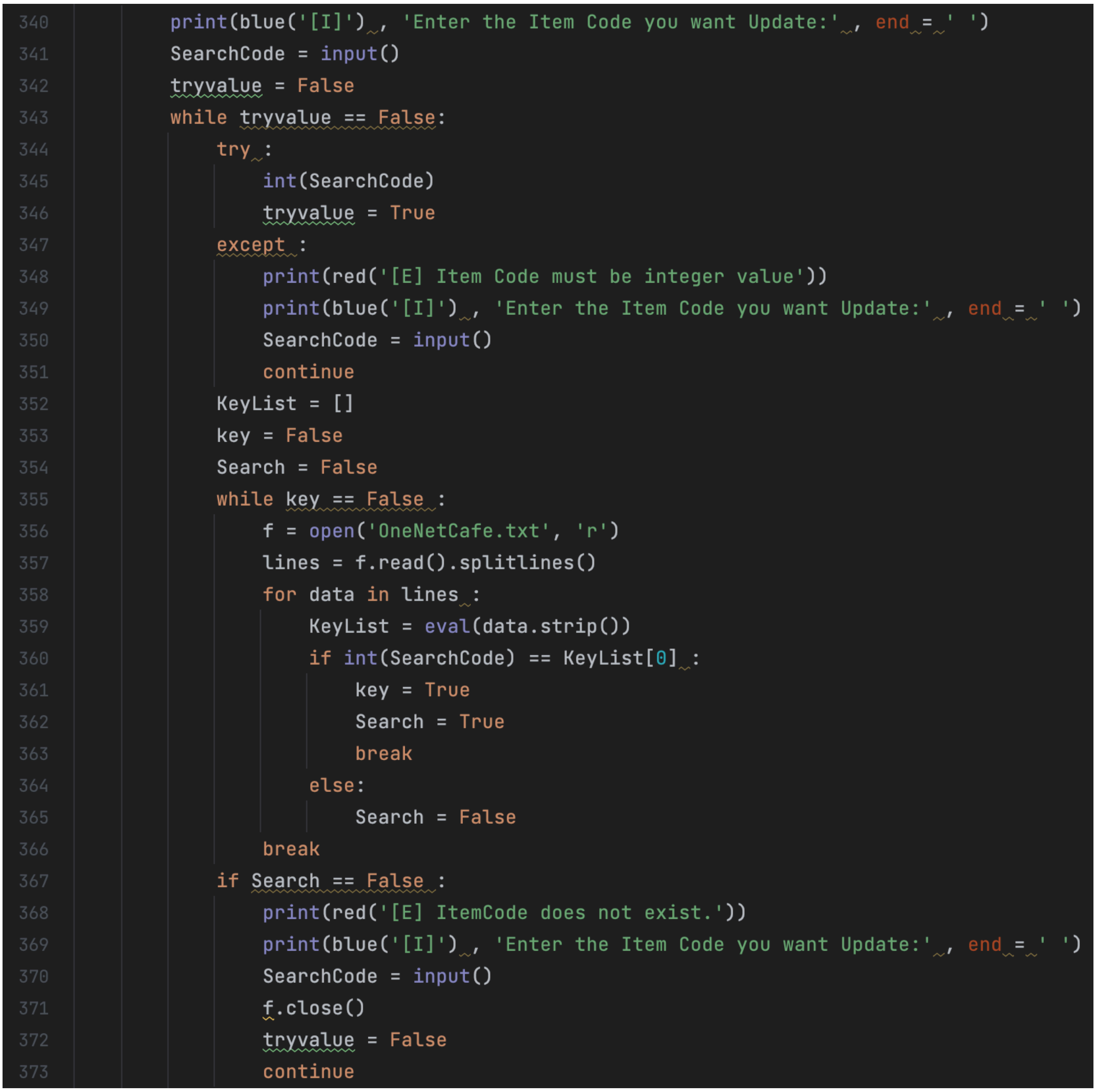
UID

There will be a validation check for each input in AID function.

Item Code

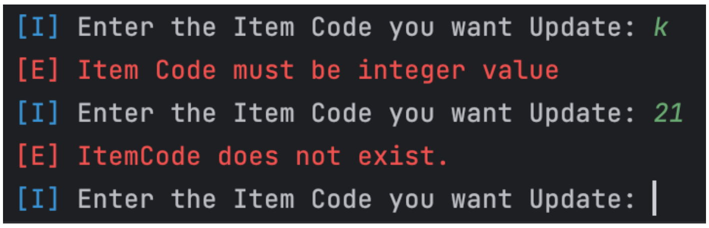
After user entering the Item Code,the application will check weather the Item code is an integer value.If not it will print an error message and will ask for Item Code.If the Item Code is valid then it will check weather the Item Code is in the saved file.If not it will print an error message and will ask for Item Code

Code :



```
        print(blue('[I]') , 'Enter the Item Code you want Update:' , end = ' ')
        SearchCode = input()
        tryvalue = False
        while tryvalue == False:
            try :
                int(SearchCode)
                tryvalue = True
            except :
                print(red('[E] Item Code must be integer value'))
                print(blue('[I]') , 'Enter the Item Code you want Update:' , end = ' ')
                SearchCode = input()
                continue
            KeyList = []
            key = False
            Search = False
            while key == False :
                f = open('OneNetCafe.txt', 'r')
                lines = f.read().splitlines()
                for data in lines :
                    KeyList = eval(data.strip())
                    if int(SearchCode) == KeyList[0] :
                        key = True
                        Search = True
                        break
                    else:
                        Search = False
                break
            if Search == False :
                print(red('[E] ItemCode does not exist.'))
                print(blue('[I]') , 'Enter the Item Code you want Update:' , end = ' ')
                SearchCode = input()
                f.close()
                tryvalue = False
                continue
```

Output :



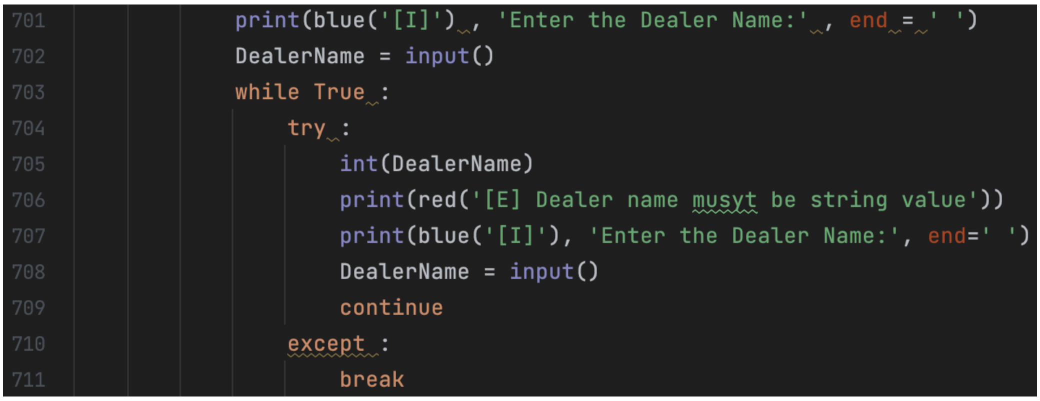
```
[I] Enter the Item Code you want Update: k
[E] Item Code must be integer value
[I] Enter the Item Code you want Update: 21
[E] ItemCode does not exist.
[I] Enter the Item Code you want Update: 
```

Other validations are same as AID function.

LDI

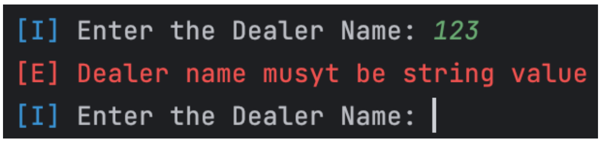
Dealer name must be a string value.If not application will print an error message and ask for Dealer Name.

Code :



```python
            print(blue('[I]') , 'Enter the Dealer Name:' , end = ' ')
            DealerName = input()
            while True :
                try :
                    int(DealerName)
                    print(red('[E] Dealer name musyt be string value'))
                    print(blue('[I]'), 'Enter the Dealer Name:', end=' ')
                    DealerName = input()
                    continue
                except :
                    break
```

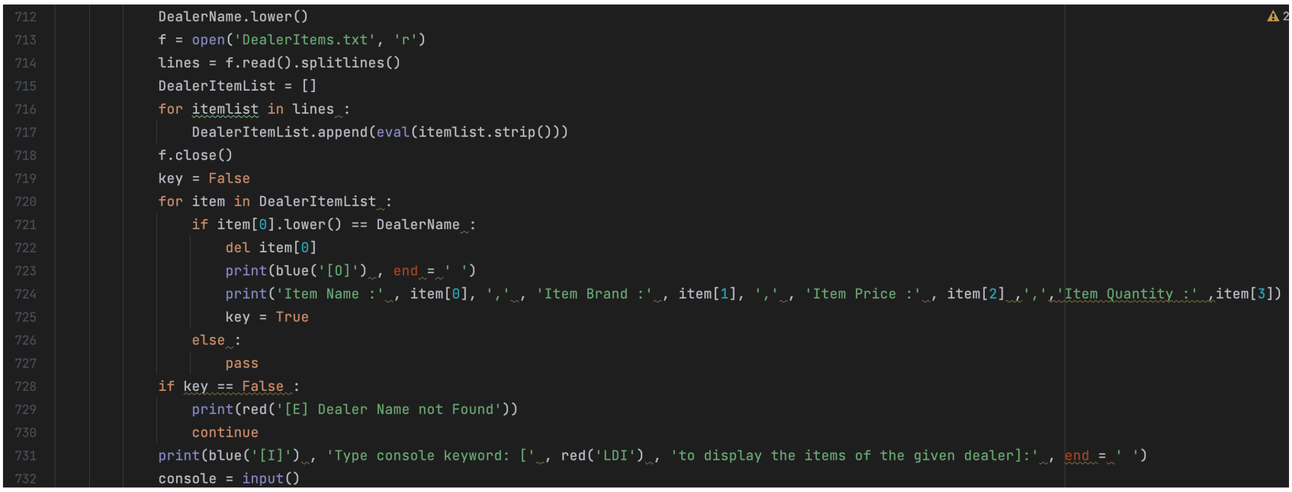
Output :



```
[I] Enter the Dealer Name: 123
[E] Dealer name musyt be string value
[I] Enter the Dealer Name: 
```

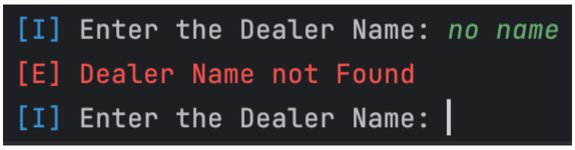
If the user enter an name not in the text file,application will print an error message and will ask for Dealer Name.

Code :



```
DealerName.lower()
f = open('DealerItems.txt', 'r')
lines = f.read().splitlines()
DealerItemList = []
for itemlist in lines :
    DealerItemList.append(eval(itemlist.strip()))
f.close()
key = False
for item in DealerItemList :
    if item[0].lower() == DealerName :
        del item[0]
        print(blue('[O]') , end = ' ')
        print('Item Name :' , item[0], ',' , 'Item Brand :' , item[1], ',' , 'Item Price :' , item[2] ,',','Item Quantity :' ,item[3])
        key = True
    else :
        pass
if key == False :
    print(red('[E] Dealer Name not Found'))
    continue
print(blue('[I]') , 'Type console keyword: [' , red('LDI') , 'to display the items of the given dealer]:' , end = ' ')
console = input()
```

Output :



```
[I] Enter the Dealer Name: no name
[E] Dealer Name not Found
[I] Enter the Dealer Name: 
```

Conclusion

In conclusion, the command line application designed for One Net Cafe provides an efficient and very user friendly system for managing the inventory. With features like adding, deleting, updating, and viewing item details, saving to a text file, selecting random suppliers, and displaying their details, the system is designed to make the inventory management process easier and more effective.

By using this system, User can make informed decisions about his inventory and ensure that his Internet cafe operates smoothly. Additionally, the sorting of items and dealers according to their locations respectively, ensures that John can quickly and easily find the information he needs.

Overall, the command line application provides a powerful tool for One Net Cafe to manage his inventory and operate it more efficiently, ultimately contributing to the success of his business.

References

- GeeksforGeeks (2023) *How to add time delay in Python*. Available at: https://www.geeksforgeeks.org/how-to-add-time-delay-in-python/.
- Sayon, S. (no date) *How to Print Bold Text in Python?* Available at: https://blog.finxter.com/how-to-print-bold-text-in-python/.
- *Add Colour to Text in Python | Data Science and Machine Learning* (no date). Available at: https://www.kaggle.com/general/273188.
- *View simple-colors on Snyk Open Source Advisor* (no date). Available at: https://snyk.io/advisor/python/simple-colors.
- *prettytable* (2023). Available at: https://pypi.org/project/prettytable/.