CS4051NI\CC4059NI Fundamental of Computing

60% Individual Coursework

2023-24 Summer

Student Name: Rijan Karki

London Met ID: 23056320

College ID: NP01CP4S240074

Assignment Due Date: Sunday, August 18, 2024

Assignment Submission Date: Sunday, August 18, 2024

Word Count: 3875

Project File Links:

YouTube Link:	Keep Unlisted YouTube URL of your Project Here
Drive Link:	Keep Google Drive URL of your Project Here with Anyone in Organization can View Option Enabled

Table of contents

Table of Figures

Table of tables

Introduction

Introduction to the project

This project involves utilizing Python’s file handling methods to develop a comprehensive program for managing the inventory of a furniture store. The program is designed to keep track of various furniture items, including the ID’s manufacturers, product names, quantities, and prices. By using this system, store administrators can effectively monitor stock levels, update inventory with new products or adjust quantities when items are sold, and generate detailed invoices for transactions.

This program reads data from a text file containing information about the furniture items. It allows users to view current inventory levels, process sales and returns, and update stock details as necessary. Additionally, the system provides functionality for generating invoices that include customer information, purchased items, quantities, and any applicable shipping costs.

Overall, this project aims to streamline the operations of a furniture store, offering an effective solution for managing inventory and handling sales transactions.

Tools used for completion of the project

IDLE

Figure 1: IDLE logo

IDLE is an Integrated Development Environment (IDE) that is included with Python. It is designed to provide a user-friendly interface for writing and executing Python code. IDLE stands for Integrated Development and Learning Environment. IDLE provides many useful features for Python developers, including syntax highlighting, auto-indentation, code completion, and debugging tools. It also includes a Python shell that allows us to interactively execute Python code and see the output in real-time. Overall, IDLE is a great tool for anyone looking to write and execute Python code.

Diagram.net

Figure 2: Diagram.net Logo

Diagrams.net (also known as draw.io) is a free, web-based tool for creating diagram and flowcharts. It provides a user-friendly interface for creating various types of diagrams, including flowcharts, network diagrams, UML diagrams, and many more. It is designed to be flexible and customizable, allowing us to create diagram that meet our specific needs. It provides a wide range of shapes, icon, and connectors that we can use to create our diagrams. One of the key features of Diagrams.net is its collaboration capabilities. We can share our diagrams with others, either by inviting them to edit diagram or by sharing a read-inly link. This makes it easy to work with others.

MS Word

Figure 3: MS Word Logo

Microsoft Word is a popular word processing software developed by Microsoft Corporation. MS Word allows users to create and edit text-based documents, such as letters, reports, essays, and more. It provides a wide range of formatting tools, allowing users to change font size, style, and colour, add headers and footers, insert images, and apply various styles to text. One of the advantages of MS Word is its user-friendly interface. It provides a simple and intuitive environment that makes it easy for users to get started with creating and formatting documents. Overall, MS Word is a powerful and versatile tool for creating and editing text-based documents with user friendly interface.

Goals and Objectives

The goals and objectives of my Python Project are as follows:

1. Manage Inventory: Track and update stock levels for different furniture items, ensuring accurate inventory records.
2. Handle Sales Transactions: Process sales by reducing the stock quantity, applying pricing, and generating detailed invoices for each transaction.
3. Facilitate Purchases: Update stock levels when new furniture is added to the inventory, reflecting the changes in the system.
4. Generate Invoices: Produce clear and detailed invoices for both sales and purchases, including necessary transaction details.
5. Validate Inputs: Implement robust error handling and input validation to prevent invalid entries and ensure reliable operations.
6. Continuous Operation: Maintain a running loop for managing multiple transactions within a single session, with an option to exit when needed.
7. Display Information: Provide well-formatted output for furniture details, ensuring readability of stock and pricing information.

Discussion and Analysis

Algorithm

Step 1: Start

Step 2: Print welcome Message and Shop Information

Step 3: Display Menu Options i.e., Buy Furniture, Sell Furniture and Exit

Step 4: Read furnituredetails.txt and Store Data in data-list

Step 5: Take user Input for choice

```
Step 6: If Choice is 1(Buy Furniture):
Step 7: Read furnituredetails.txt
Step 8: Print the information to the shell in a managed way
Step 9: Take user input for furniture ID to buy
Step 10: If the furniture ID is in the list:
Step 11: store details of the selected furniture in new-list
Step 12: Take user input for quantity
Step 13: Confirm order:
Step 14: If confirmation is yes:
Step 15: Store the choice and quantity in a 2d list
Step 16: Update the quantity in new-list
Step 17: Ask user if they want to buy more:
Step 18: If y, go to Step 9
Step 19: if n, go to Step 22
Step 20: If confirmation is not yes, go to Step 9
Step 21: Update furnituredetails.txt with new quantities
Step 22: Calculate net amount for the transaction and generate invoice:
Step 23: Take user name and current date/time
Step 24: Create a unique text file for the invoice
Step 25: Print invoice details to the text file
Step 26: Go to Step 5
Step 27: If choice is 2(sell furniture):
Step 28: Read furnituredetails.txt
Step 29: Print the information to the shell in a managed way
Step 30: Take user input for furniture Id to sell
Step 31: If the furniture Id is in the list:
Step 32: Store details of the selected furniture in new list
Step 33: Take user input for quantity
Step 34: If the quantity is available:
Step 35: Confirm order:
Step 36: If the confirmation is yes:
Step 37: Store the choice and quantity in 2D list
Step 38: Update the quantity
Step 39: Ask user if they want to sell more:
Step 40: If yes, go to step 30
Step 41: If no, go to step 44
Step 42: If confirmation is not yes or no go to Step 30
Step 43: Update furnituredetails.txt with new quantities
Step 44: Calculate net amount for the transaction and generate invoice:
Step 45: Take user name and current date/time
Step 46: Create a unique text file for the invoice
Step 47: Print invoice details to the text file and shell
Step 48: Go to Step 5
Step 49: If the choice is 3(Exit):
Step 50: End
Step 51: If the choice is invalid, display an error message and go to Step 5
```

Flowchart

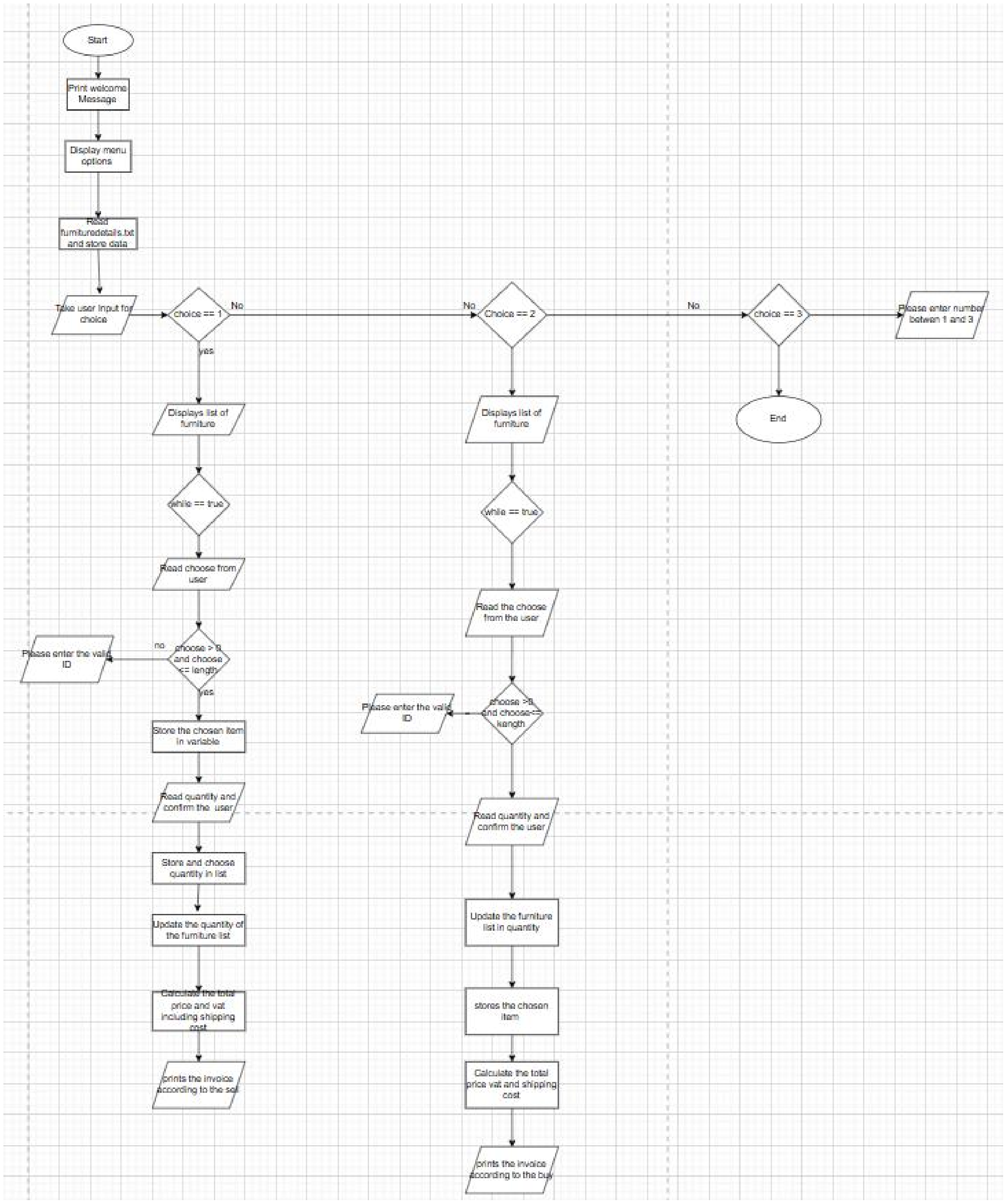

Start
Print welcome Message
Display menu options
Read furnituredetails.txt and store data
Take user input for choice
choice == 1
No
yes
Displays list of furniture
while == true
Read choose from user
choose > 0 and choose <= length
no
Please enter the valid ID
yes
Store the chosen item in variable
Read quantity and confirm the user
Store and choose quantity in list
Update the quantity of the furniture list
Calculate the total price and vat including shipping cost
prints the invoice according to the sell
No
Choice == 2
Displays list of furniture
while == true
Read the choose from the user
choose >0 and choose<= length
Please enter the valid ID
Read quantity and confirm the user
Update the furniture list in quantity
stores the chosen item
Calculate the total price vat and shipping cost
prints the invoice according to the buy
No
choice == 3
Please enter number betwen 1 and 3
End

Figure 4: Flowchart

Pseudocode

```
DEFINE FUNCTION file open ()
  OPEN furnituredetails.txt in read mode
  CLOSE the file
 DEFINE FUNCTION listt ()
   OPEN the file furnituredetails.txt in read mode
    INITIALIZE an empty list rec
    FOR each in the file
       REMOVE newline characters from the file
       SPLIT the line by commas and append the resulting list to rec
  CLOSE the file


DEFINE FUNCTION write_furniture_details(furniture_details)

   OPEN furnituredetials.txt in write mode

   FOR each furniture item in furniture_details

        WRITE the id, manufacturer, product, quantity and price to file

   CLOSE the file

DEFINE FUNCTION write_invoice(invoice_details)

   GENERATE a unique invoice_id using generate_invoice_id()

   OPEN a new file named after the invoice_id in write mode

   WRITE Invoice Details and customer details (name, address, contact
number) to the file

    WRITE details of the purchased furniture items to the file

    CALCULATE total_amount as the sum of the price times quantity for
each item

     CALCULATE vat_amount as 13% of total_amount

     ADD shipping_cost from invoice_details

    CALCULATE total_amouont_with_vat_shipping as the sum of
total_amount, vat_amount and shipping_cost

     WRITE the total_amount, vat_amount, shipping_cost, and
total_amount_with_vat_shipping to the file

      CLOSE the file
```

```
          DEFINE FUNCTION generate_invoice_id ()
             GET the current date and time
             RETURN a string INVOICE followed by the current date and time
formatted as YYYYMMDDHHMMSS
          DEFINE FUNCTION decrease_quantity (furniture_list, furniture_id,
quantity)
              FOR each item in furniture_list
                IF the furniture_id matches the current item ID
                   IF the items quantity is greater than or equal to quantity
                      SUBTRACT quantity from the item's quantity
                      RETURN True
                   ELSE
                      PRINT Not enough quantity available.
                      RETURN False
            PRINT FurnitureID is not Found.
            RETURN False
DEFINE FUNCTION increase_quantity (furniture_list, furniture_id, quantity)
   FOR each item in furniture_list
     IF the function_id matches the current items ID
        ADD quantity to the item's quantity
        RETURN True
    PRINT Furniture ID is not found.
    RETURN False
DEFINE FUNCTION check_availablility (furniture_list, furniture_id)
  FOR each item in furniture_list
     IF the furniture_id matches the current items ID
        IF the items quantity is greater than 0
           RETURN Available
         ELSE
```

```
                RETURN Not Available
        PRINT Furniture ID not Found
        RETURN Not Available
IMPORT MODULES read, write, and operation
DEFINE FUNCTION get_customer_details ()
    LOOP until valid input is received
        PROMPT user to enter their name, address, and contact_no
        IF contact_no is numeric
            RETURN name, address, and contact_no
         ELSE
            PRINT Invalid contact number. Please enter a numeric value.
DEFINE FUNCTION display_furniture_details(furniture_details)
     PRINT table headers and separators
     FOR each furniture item in furniture_details
         DISPLAY the id, manufacturer, product, quantity, and price fields
DEFINE FUNCTION main ()
    INITIALIZE furniture_details by calling read. listt ()
    LOOP until user chooses to exit
        PRINT Welcome to the BRJ Furniture and display menu options
             PRINT Sell Furniture
             PRINT Buy Furniture
             PRINT Exit
        PROMPT user to enter their selection
      IF selection is equals to 1 then
             CALL display_furniture_details(furniture_details)
              PROMPT user to enter furniture_id and quantity
              IF operation. decrease_quantity (furniture_details, furniture_id,
quantity) then
              CALL get_customer_details () to obtain customer details
```

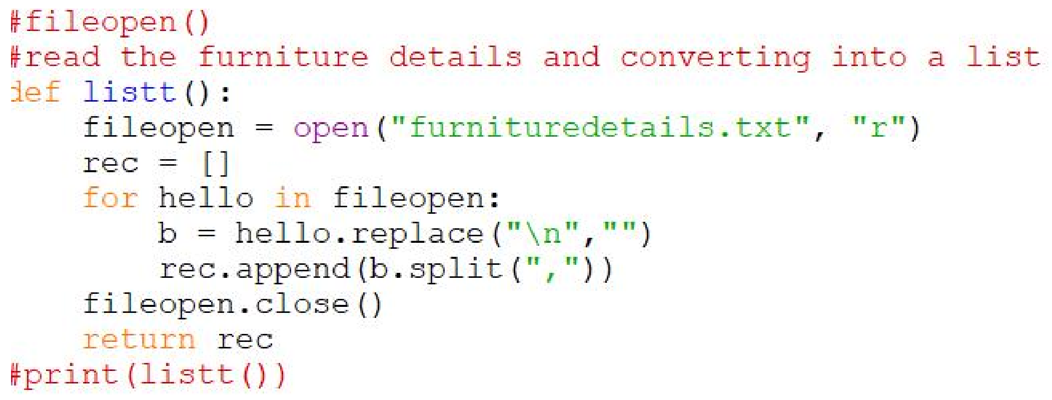
```
                    CREATE invoice_details dictionary:
                        ADD customer details
                        SET shipping_cost to a fixed value
                        FOR each item in furniture_details
                            IF item [index 0] is equals to furniture_id then
                                ADD items details to invoice_details[items]
                      CALL write. write_invoice(invoice_details)
                      CALL write. write_furniture_details(furniture_details)
                      PRINT Product has been sold.
        ELSE IF selection is equal to 2 then
            CALL display_furniture_details(furniture_details)
            PROMPT user to enter furniture_id and quantity
            IF operation. Increase_quantity (furniture_details, furniture_id, quantity)
        then
                PRINT Furniture returned successfully.
                CALL write. write_furniture_details(furniture_details)
        ELSE IF selection is equals to 3 then
                PRINT Thank you for visiting BRJ Furniture. Have a good day!!
                BREAK the loop to exit the program
         ELSE
               PRINT Invalid choice. Please enter a number between 1 and 3.
         ERROR HANDLING
               IF an exception occurs
                  PRINT An error occurred. Please enter a numeric value.
CALL main ()
```

Data Structure

In Python, data structures are objects used to store and organize data. Python provides a variety of built-in data structures that make it easy to manipulate and work with data in various ways. Some of the commonly used data structures in Python include lists, tuples, sets, and dictionaries. The building blocks of computer programs are data structures, which offer distinct ways of organizing data for efficient access depending on the specific use case. Python comes with a comprehensive collection of data structures in its standard library.

List

In Python, a list is a data structure used to hold a sequence of data with different types. It can be described as a collection of items or values that can be changed or modified after it has been created. The elements of a list are enclosed within square brackets [] and separated by commas. As a mutable type, lists allow for easy modification of their elements.



```
#fileopen()
#read the furniture details and converting into a list
def listt():
    fileopen = open("furnituredetails.txt", "r")
    rec = []
    for hello in fileopen:
        b = hello.replace("\n","")
        rec.append(b.split(","))
    fileopen.close()
    return rec
#print(listt())
```

Figure 5: List used in the code

Other data structures that were not used in this program are:

Dictionary

A dictionary in python is a data structure consisting of a collection of key-value pairs, where the values can be of any Python object. The keys of dictionary must be immutable Python objects, such as numbers, strings, or tuples. Dictionaries are useful when we need to look up values based on their keys. Example of dictionary is:

Person = {‘name’: ‘Ram’:, ‘age’:, 30, ‘city’:, ‘New York’}

Tuples

A tuple is like a list, but it is immutable, meaning that once it is created, its values cannot be changed, Tuples are often used to group related data together, such as the coordinates of a point or the data and time of an event. To create a tuple in Python, we can use parentheses () and separate the values with commas. Example of tuple is:

Coordinates = (10,20)

Sets

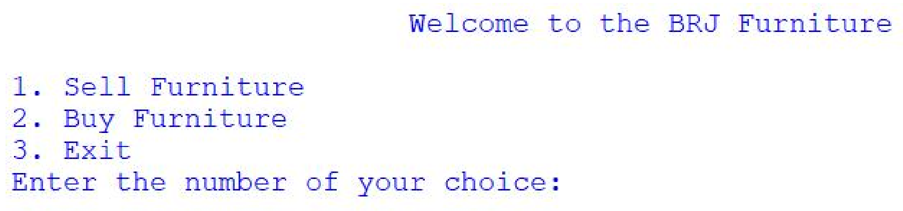
A set is an unordered collection of unique values. Sets are useful when we need to remove duplicates from a list or perform set operations such as union, intersection, and difference. To create a set in Python, you can use curly braces {} or the set () function.

Example: my _set = {1, 2, 3, 4}

Program

This program is designed by writing bunch of codes into different function and calling and reusing them as needed. While making the program 2d list is used to store data of the furniture, exception handling is done where needed, looping is done to achieve some required goals.

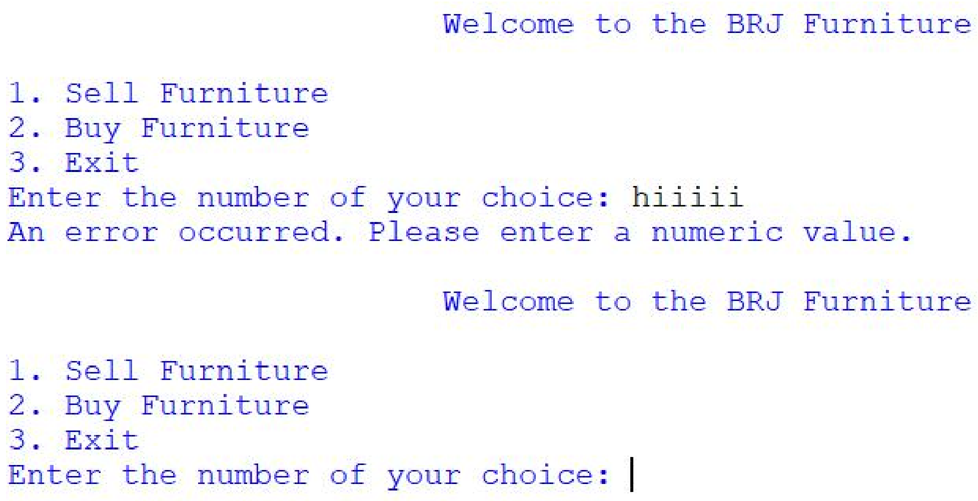
The program is runed first where the welcome message appears in the shell asking the user if they want to sell buy or exit from the system.



```
                         Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice:
```

Figure 6: Interface when code is run

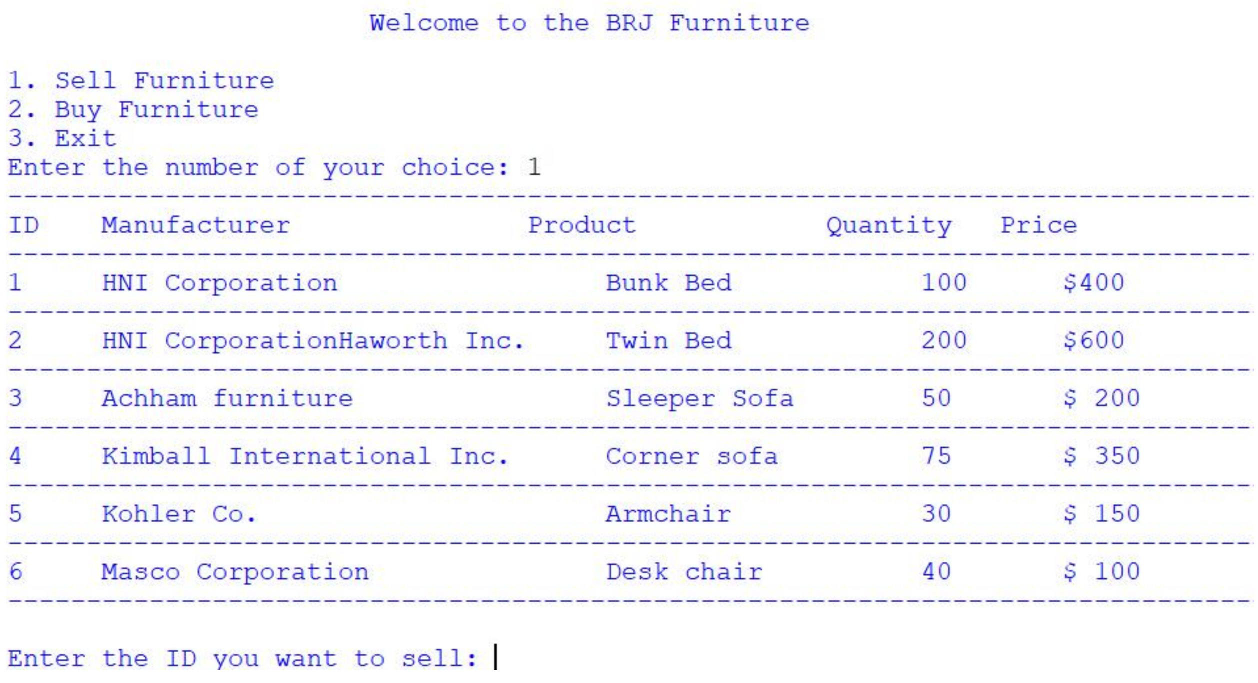
When user provides a string where there is need of integer, due to exceptional handling the program doesn't stop it displays an error message and continues the program.



```
                              Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: hiiiii
An error occurred. Please enter a numeric value.

                              Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 
```

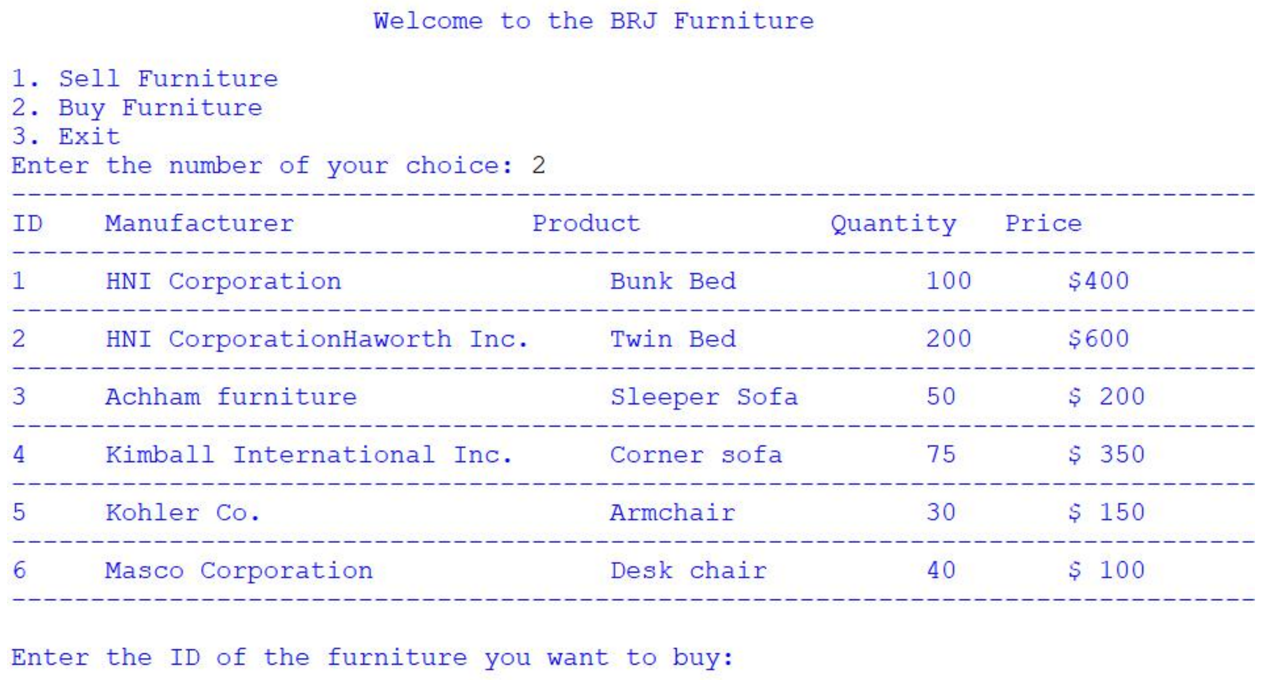
Figure 7: Implementation exception handling

If we give 1 then it displays the list of the furniture and then asks the user to write the required furniture ID then the quantity, they want to sell then again, the program asks the users name, address and contact no. If the users input 2 then it again display the list of the furniture's and the same process as per the sales and when the user inputs 3 then the program will be ended with a gentle goodbye.



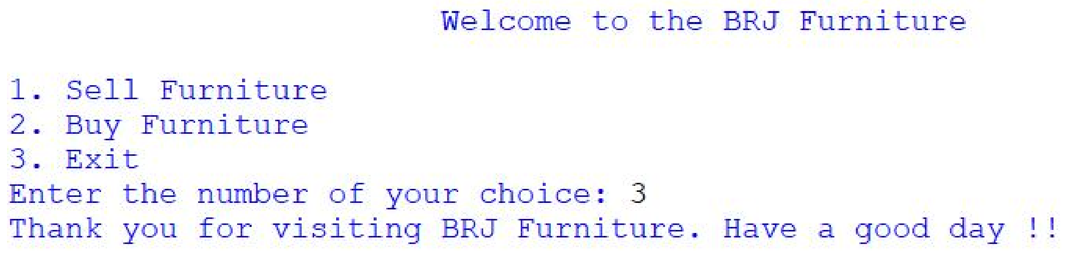
```
                           Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 1
-------------------------------------------------------------------------------
ID    Manufacturer                 Product            Quantity    Price
-------------------------------------------------------------------------------
1     HNI Corporation                   Bunk Bed            100       $400
-------------------------------------------------------------------------------
2     HNI CorporationHaworth Inc.       Twin Bed            200       $600
-------------------------------------------------------------------------------
3     Achham furniture                  Sleeper Sofa        50        $ 200
-------------------------------------------------------------------------------
4     Kimball International Inc.        Corner sofa         75        $ 350
-------------------------------------------------------------------------------
5     Kohler Co.                        Armchair            30        $ 150
-------------------------------------------------------------------------------
6     Masco Corporation                 Desk chair          40        $ 100
-------------------------------------------------------------------------------

Enter the ID you want to sell: 
```

Figure 8: Selling a furniture



```
                       Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 2
-------------------------------------------------------------------------------
ID    Manufacturer                 Product            Quantity   Price
-------------------------------------------------------------------------------
1     HNI Corporation                   Bunk Bed            100      $400
-------------------------------------------------------------------------------
2     HNI CorporationHaworth Inc.       Twin Bed            200      $600
-------------------------------------------------------------------------------
3     Achham furniture                  Sleeper Sofa        50       $ 200
-------------------------------------------------------------------------------
4     Kimball International Inc.        Corner sofa         75       $ 350
-------------------------------------------------------------------------------
5     Kohler Co.                        Armchair            30       $ 150
-------------------------------------------------------------------------------
6     Masco Corporation                 Desk chair          40       $ 100
-------------------------------------------------------------------------------

Enter the ID of the furniture you want to buy:
```

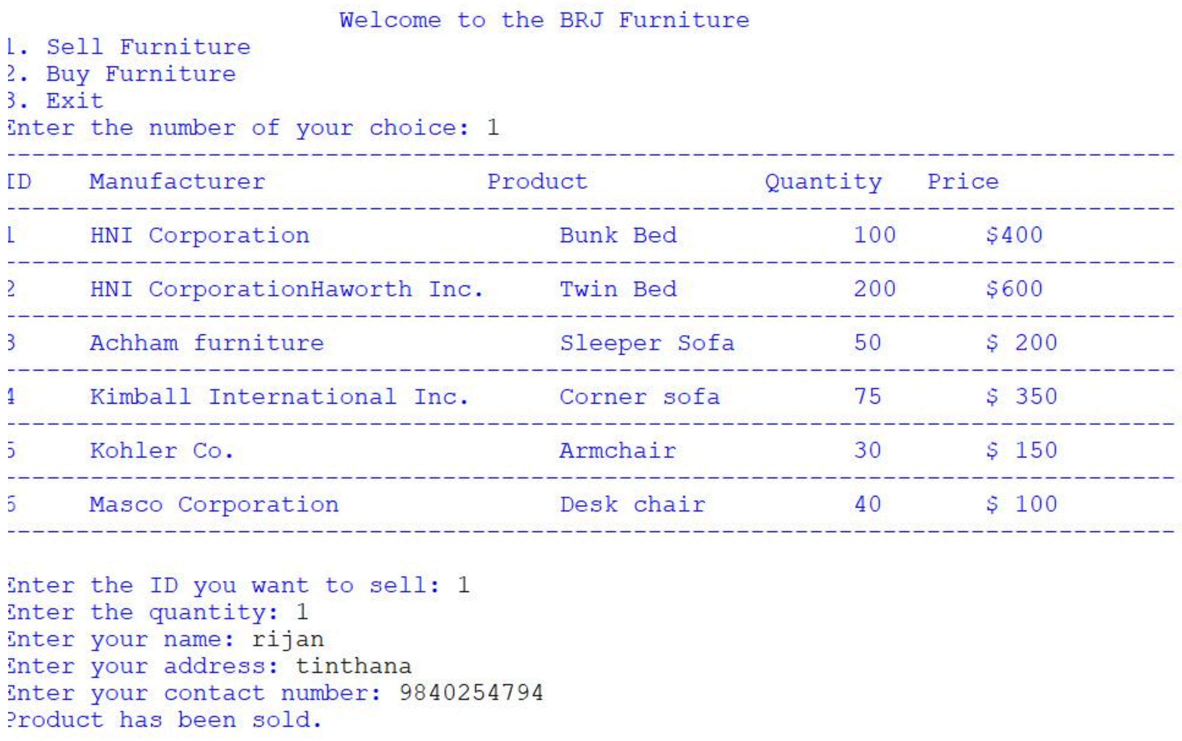
Figure 9:Buying a furniture



```
                    Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 3
Thank you for visiting BRJ Furniture. Have a good day !!
```

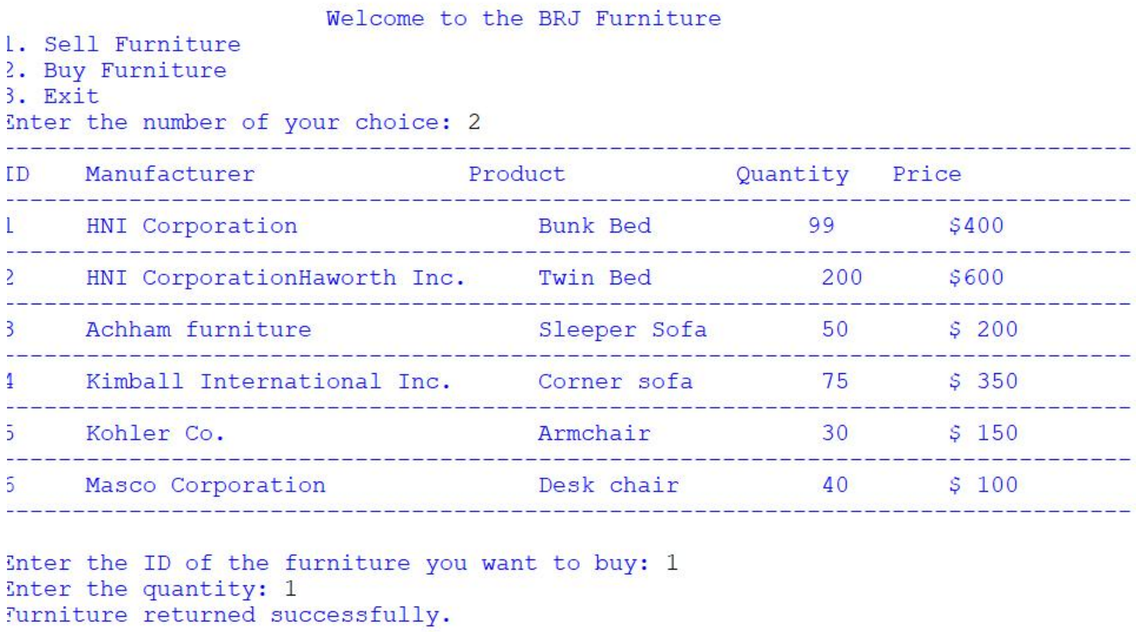
Figure 10: Exiting the program

After pressing 1 or 2 it displays the furniture list and then asks the user the id of the furniture and the quantity the want of their needs after that the user information like name address and the phone number is need to placed for the detail enquiry and art last there is a message pop up like the product is sold or furniture returned.



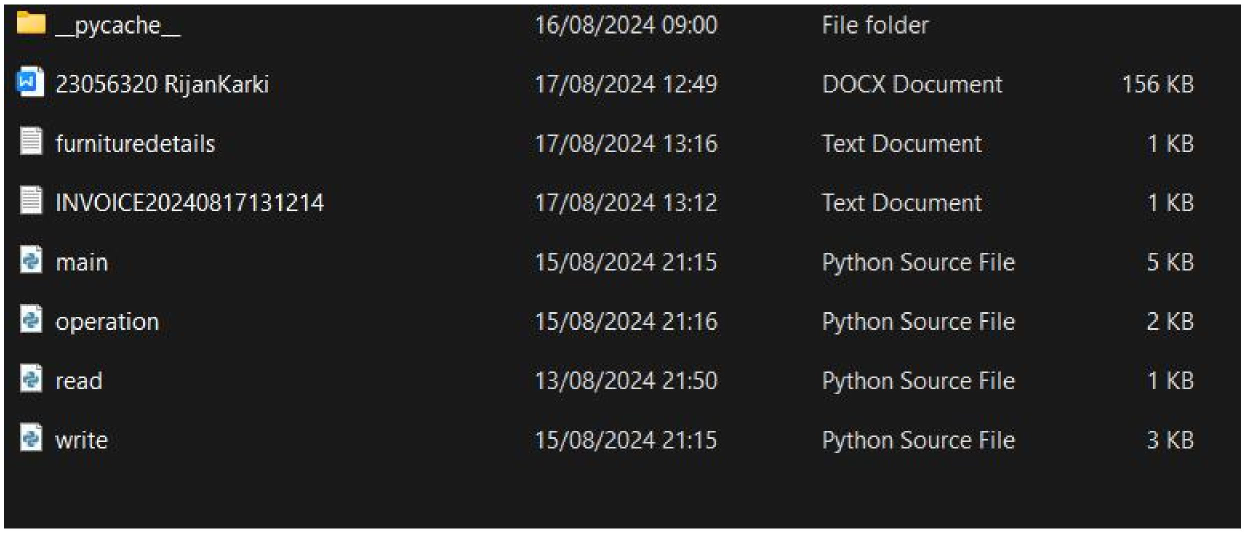
```
                      Welcome to the BRJ Furniture
l. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 1
--------------------------------------------------------------------------------
ID    Manufacturer                 Product            Quantity   Price
--------------------------------------------------------------------------------
l     HNI Corporation                   Bunk Bed            100      $400
--------------------------------------------------------------------------------
2     HNI CorporationHaworth Inc.       Twin Bed            200      $600
--------------------------------------------------------------------------------
3     Achham furniture                  Sleeper Sofa        50       $ 200
--------------------------------------------------------------------------------
4     Kimball International Inc.        Corner sofa         75       $ 350
--------------------------------------------------------------------------------
5     Kohler Co.                        Armchair            30       $ 150
--------------------------------------------------------------------------------
6     Masco Corporation                 Desk chair          40       $ 100
--------------------------------------------------------------------------------

Enter the ID you want to sell: 1
Enter the quantity: 1
Enter your name: rijan
Enter your address: tinthana
Enter your contact number: 9840254794
Product has been sold.
```

Figure 11: Process while selling a product



```
                      Welcome to the BRJ Furniture
l. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 2
-------------------------------------------------------------------------------
ID    Manufacturer                Product           Quantity   Price
-------------------------------------------------------------------------------
l     HNI Corporation                 Bunk Bed           99        $400
-------------------------------------------------------------------------------
2     HNI CorporationHaworth Inc.     Twin Bed            200       $600
-------------------------------------------------------------------------------
3     Achham furniture                Sleeper Sofa        50        $ 200
-------------------------------------------------------------------------------
4     Kimball International Inc.      Corner sofa         75        $ 350
-------------------------------------------------------------------------------
5     Kohler Co.                      Armchair            30        $ 150
-------------------------------------------------------------------------------
5     Masco Corporation               Desk chair          40        $ 100
-------------------------------------------------------------------------------

Enter the ID of the furniture you want to buy: 1
Enter the quantity: 1
Furniture returned successfully.
```

Figure 12: Process while buying a product



__pycache__	16/08/2024 09:00	File folder	
23056320 RijanKarki	17/08/2024 12:49	DOCX Document	156 KB
furnituredetails	17/08/2024 13:16	Text Document	1 KB
INVOICE20240817131214	17/08/2024 13:12	Text Document	1 KB
main	15/08/2024 21:15	Python Source File	5 KB
operation	15/08/2024 21:16	Python Source File	2 KB
read	13/08/2024 21:50	Python Source File	1 KB
write	15/08/2024 21:15	Python Source File	3 KB

Figure 13: Invoice in a txt file

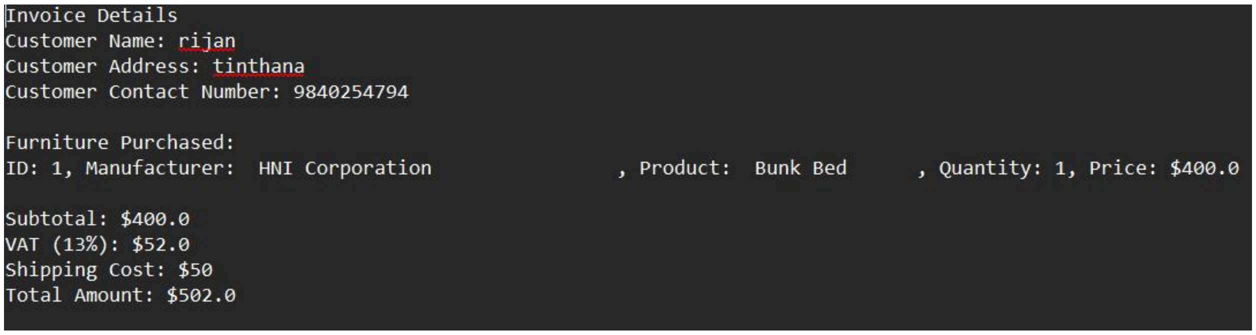

```
Invoice Details
Customer Name: rijan
Customer Address: tinthana
Customer Contact Number: 9840254794

Furniture Purchased:
ID: 1, Manufacturer:  HNI Corporation                , Product:  Bunk Bed      , Quantity: 1, Price: $400.0

Subtotal: $400.0
VAT (13%): $52.0
Shipping Cost: $50
Total Amount: $502.0
```

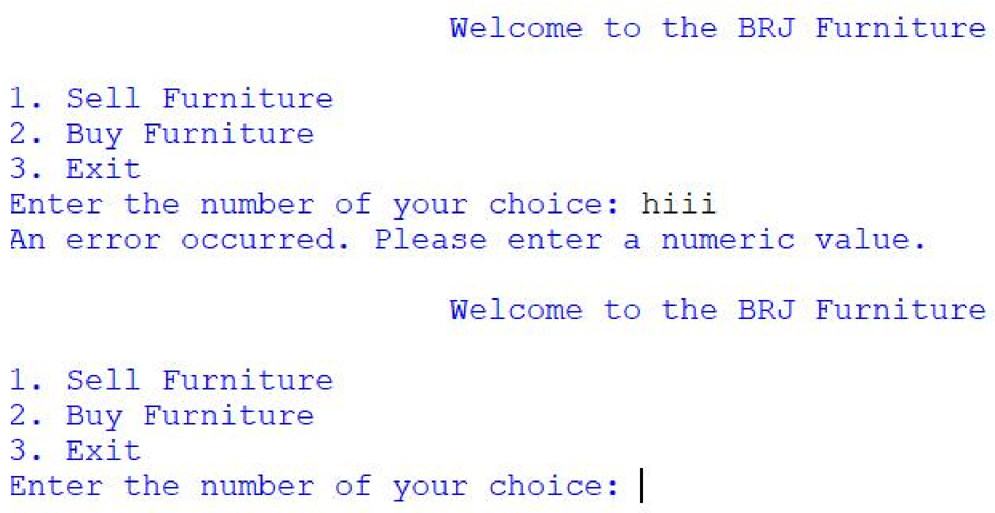
Figure 14: Creating invoice

Testing

Implementation of try, except

Objective	To check whether try and except works or not
Action	- Program is executed. - Assigning random string value where integer is required.
Expected result	Error message should appear saying please enter a numeric number.
Actual result	Error message appears saying please enter a numeric number
Conclusion	Test successful

Table 1: Test of implementation of try and except



```
                      Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: hiii
An error occurred. Please enter a numeric value.

                      Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 
```

Figure 15: Entering String value where Integer value is required

Test 2: Selection buy and sell of furniture

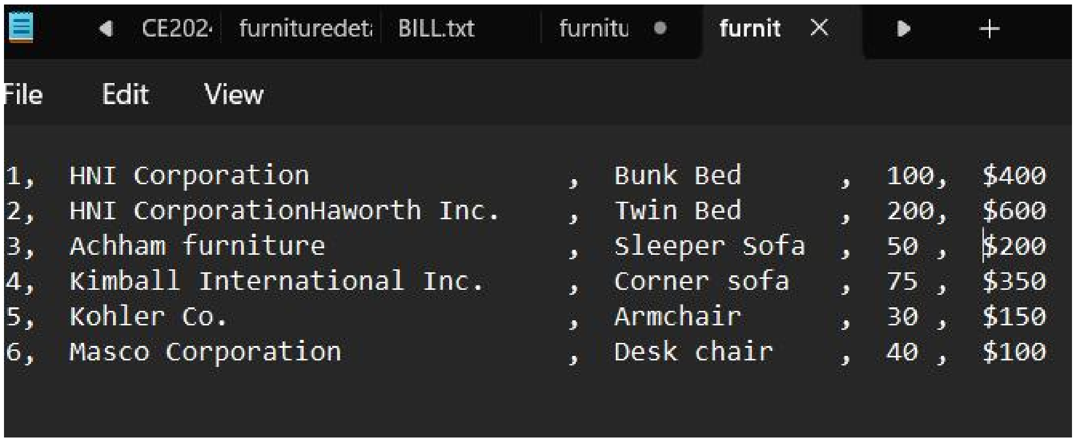
Objective	To check input validation while buying and selling
Action	- Program is executed. - Going to buying section - Negative value is entered in the field where id of the product should be entered. - Application runs in loop. - Providing non existence id number of the furniture.
Expected result	Error message should be displayed saying to enter a valid id number
Actual Result	Error message is not being displayed.

Table 2: Test to input validation

Test 3: Show the update in stock of furniture

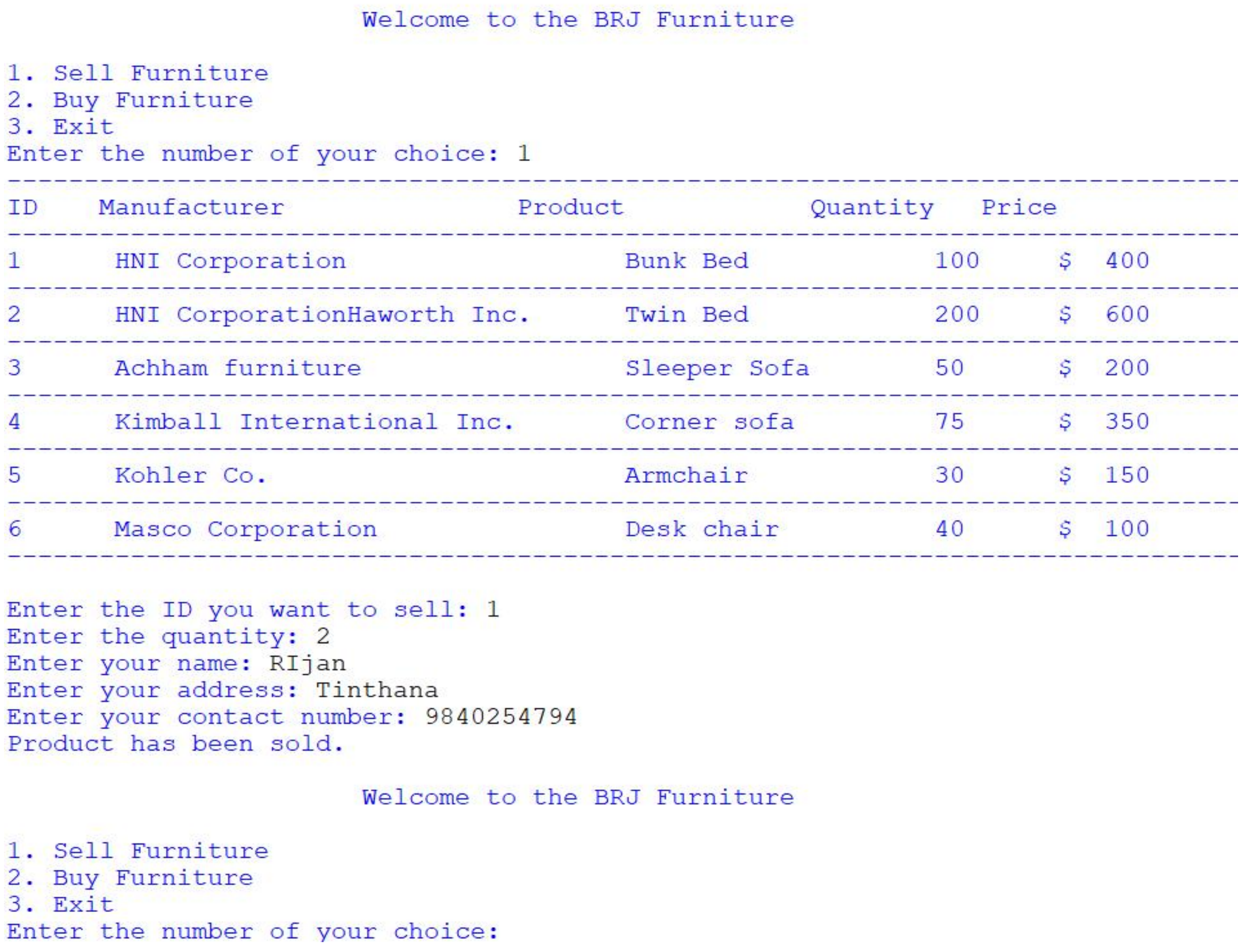
Objective	To check and update in the stock after purchase and sell of furniture
Action	- Program is executed. - 1 is pressed to sale the furniture - Filling up the required things - 2 items are bought. - 2 is pressed to buy the laptop - Filling up the required things. - 2 items are sold
Expected result	The quantity in the txt file should be changed.
Actual result	The quantity in the txt file is changed.
Conclusion	Test successful

Table 3: Test of stock update



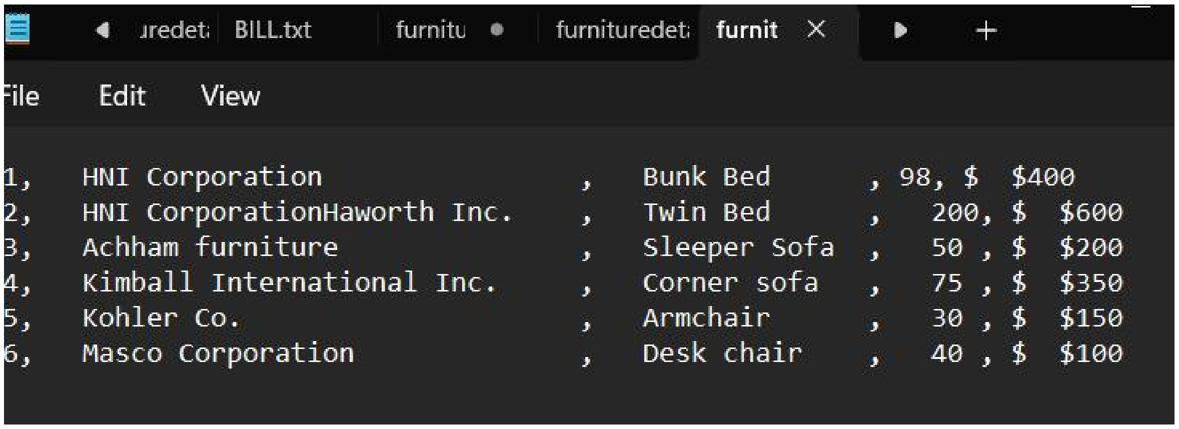
```
1,  HNI Corporation                 ,  Bunk Bed      ,  100,  $400
2,  HNI CorporationHaworth Inc.     ,  Twin Bed      ,  200,  $600
3,  Achham furniture                ,  Sleeper Sofa  ,  50 ,  $200
4,  Kimball International Inc.      ,  Corner sofa   ,  75 ,  $350
5,  Kohler Co.                      ,  Armchair      ,  30 ,  $150
6,  Masco Corporation               ,  Desk chair    ,  40 ,  $100
```

Figure 16: Data before buy or sell



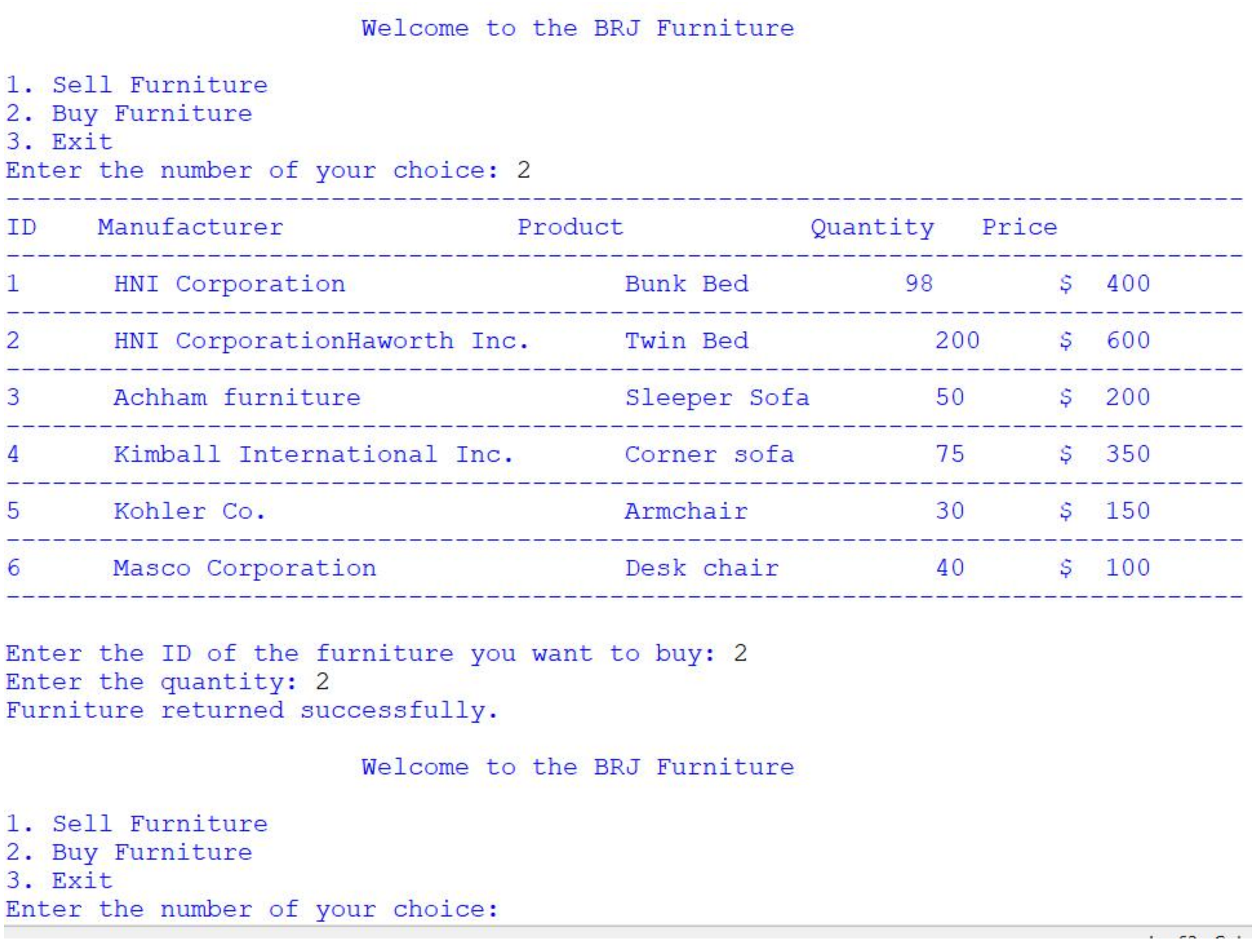
```
                       Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 1
--------------------------------------------------------------------------------
ID    Manufacturer                 Product            Quantity   Price
--------------------------------------------------------------------------------
1      HNI Corporation                    Bunk Bed            100       $  400
--------------------------------------------------------------------------------
2      HNI CorporationHaworth Inc.        Twin Bed            200       $  600
--------------------------------------------------------------------------------
3      Achham furniture                   Sleeper Sofa        50        $  200
--------------------------------------------------------------------------------
4      Kimball International Inc.         Corner sofa         75        $  350
--------------------------------------------------------------------------------
5      Kohler Co.                         Armchair            30        $  150
--------------------------------------------------------------------------------
6      Masco Corporation                  Desk chair          40        $  100
--------------------------------------------------------------------------------

Enter the ID you want to sell: 1
Enter the quantity: 2
Enter your name: RIjan
Enter your address: Tinthana
Enter your contact number: 9840254794
Product has been sold.

                       Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice:
```

Figure 17: Selling two furniture



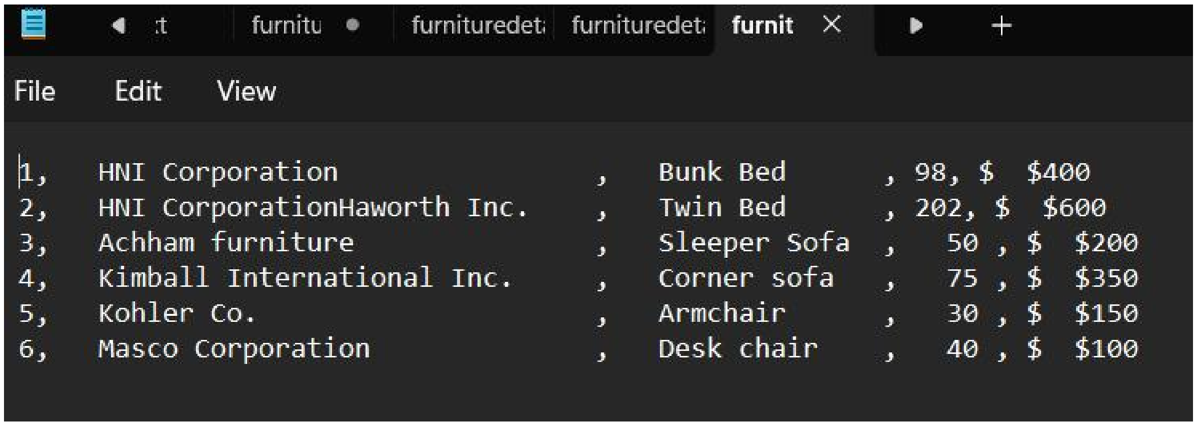
```
1,   HNI Corporation                  ,  Bunk Bed      , 98, $  $400
2,   HNI CorporationHaworth Inc.      ,  Twin Bed      ,   200, $  $600
3,   Achham furniture                 ,  Sleeper Sofa  ,   50 , $  $200
4,   Kimball International Inc.       ,  Corner sofa   ,   75 , $  $350
5,   Kohler Co.                       ,  Armchair      ,   30 , $  $150
6,   Masco Corporation                ,  Desk chair    ,   40 , $  $100
```

Figure 18: Data after selling two furniture of ID 1



```
                       Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice: 2
--------------------------------------------------------------------------------
ID    Manufacturer                 Product            Quantity   Price
--------------------------------------------------------------------------------
1      HNI Corporation                    Bunk Bed           98         $  400
--------------------------------------------------------------------------------
2      HNI CorporationHaworth Inc.        Twin Bed             200      $  600
--------------------------------------------------------------------------------
3      Achham furniture                   Sleeper Sofa         50       $  200
--------------------------------------------------------------------------------
4      Kimball International Inc.         Corner sofa          75       $  350
--------------------------------------------------------------------------------
5      Kohler Co.                         Armchair             30       $  150
--------------------------------------------------------------------------------
6      Masco Corporation                  Desk chair           40       $  100
--------------------------------------------------------------------------------

Enter the ID of the furniture you want to buy: 2
Enter the quantity: 2
Furniture returned successfully.

                       Welcome to the BRJ Furniture

1. Sell Furniture
2. Buy Furniture
3. Exit
Enter the number of your choice:
```

Figure 19: Buying two furniture



```
1,    HNI Corporation                  ,   Bunk Bed       , 98, $  $400
2,    HNI CorporationHaworth Inc.      ,   Twin Bed       , 202, $  $600
3,    Achham furniture                 ,   Sleeper Sofa   ,   50 , $  $200
4,    Kimball International Inc.       ,   Corner sofa    ,   75 , $  $350
5,    Kohler Co.                       ,   Armchair       ,   30 , $  $150
6,    Masco Corporation                ,   Desk chair     ,   40 , $  $100
```

Figure 20: Data after buying two furniture of ID 2

Conclusion

In conclusion, this python project is an excellent example of how programming can be used to streamline business operations. The project involves creating a program that can manage information about available furniture in a text file, which can be used to update stock levels and generate invoices for sales and purchases.

The program is designed to buy purchase and sell the furniture from manufacturers to the customers. With each transaction, the program generates an invoice that contains all the relevant details such as the manufacturer, product name, quantity and price details.

Overall, this python project is an excellent example of how programming can be used to simplify and automate business operations. By automating repetitive tasks., the furniture shop can focus on more critical aspects of its business, such as providing customer service and expanding its operations.

References

https://www.w3schools.com/python/python_reference.asp

https://www.educba.com/python-references/

https://www.geeksforgeeks.org/shared-reference-in-python/

https://docs.python.org/3/reference/index.html

https://wiki.python.org/moin/ReferenceBooks

Appendix

Main.py

```
import read

import write

import operation


def get_customer_details():
    """
    Asks the user for their details and returns them.
    Handles invalid contact number input with retry.
    """
    while True:
        try:
            name = input("Enter your name: ")
            address = input("Enter your address: ")
            contact_no = int(input("Enter your contact number: "))
            return name, address, contact_no
        except:
            print("Invalid contact number. Please enter a numeric value.")


def display_furniture_details(furniture_details):
    """
    Display the details of furniture in a formatted table.
    """
    print("-" * 80)
    print("ID    Manufacturer               Product            Quantity   Price")
    print("-" * 80)
    for i in range(len(furniture_details)):
        id_field = furniture_details[i][0] + " " * (5 - len(furniture_details[i][0]))
        manufacturer_field = furniture_details[i][1] + " " * (30 - len(furniture_details[i][1]))
        product_field = furniture_details[i][2] + " " * (20 - len(furniture_details[i][2]))
        quantity_field = furniture_details[i][3] + " " * (10 - len(furniture_details[i][3]))
        price_field = furniture_details[i][4].replace('$', '') + " " * (10 -
len(furniture_details[i][4].replace('$', '')))

        print(id_field + manufacturer_field + product_field + quantity_field + "$" +
price_field)
        print("-" * 80)

def main():
    """
    Main function to manage the furniture store operations.
    Provides a menu to sell, buy furniture, or exit the program.
    """
    furniture_details = read.listt()

    while True:
        print("\n                    Welcome to the BRJ Furniture                              ")
        print("1. Sell Furniture")
        print("2. Buy Furniture")
        print("3. Exit")

        try:
            selection = int(input("Enter the number of your choice: "))
```

```
if selection == 1:
    display_furniture_details(furniture_details)
    furniture_id = input("\nEnter the ID you want to sell: ")
    quantity = int(input("Enter the quantity: "))

    if operation.decrease_quantity(furniture_details, furniture_id, quantity):
        name, address, contact_no = get_customer_details()
        invoice_details = {
            'customer_name': name,
            'customer_address': address,
            'customer_contact': contact_no,
            'items': [],
            'shipping_cost': 50  # Example shipping cost
        }
        for item in furniture_details:
            if item[0] == furniture_id:
                invoice_details['items'].append({
                    'id': item[0],
                    'manufacturer': item[1],
                    'product': item[2],
                    'quantity': quantity,
                    'price': float(item[4].replace('$', ''))
                })
        write.write_invoice(invoice_details)
        write.write_furniture_details(furniture_details)
        print("Product has been sold.")

elif selection == 2:
```

```
            display_furniture_details(furniture_details)
            furniture_id = input("\nEnter the ID of the furniture you want to buy: ")
            quantity = int(input("Enter the quantity: "))

            if operation.increase_quantity(furniture_details, furniture_id, quantity):
                print("Furniture returned successfully.")
                write.write_furniture_details(furniture_details)

        elif selection == 3:
            print("Thank you for visiting BRJ Furniture. Have a good day !!")
            break

        else:
            print("Invalid choice. Please enter a number between 1 and 3.")

    except:
        print("An error occurred. Please enter a numeric value.")

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

read.py

```
#open and read the furniture details of a file
def fileopen():
    fileopen = open("furnituredetails.txt", "r")
    #print(fileopen.read())
    fileopen.close()
#fileopen()
```

```
#read the furniture details and converting into a list
def listt():
    fileopen = open("furnituredetails.txt", "r")
    rec = []
    for hello in fileopen:
        b = hello.replace("\n","")
        rec.append(b.split(","))
    fileopen.close()
    return rec
#print(listt())
 write.py
import datetime

def write_furniture_details(furniture_details):
    """
    Write the updated furniture details to the file.
    """
    file = open("furnituredetails.txt", "w")
    for furniture in furniture_details:
        file.write(furniture[0] + ", " +
                   furniture[1] + ", " +
                   furniture[2] + ", " +
                   furniture[3] + ", " +
                   "$" + furniture[4] + "\n")
    file.close()

def write_invoice(invoice_details):
    """
    Write the invoice details to a new file with a unique invoice ID.
    """
    invoice_id = generate_invoice_id()  # Function to generate a unique invoice ID
    file = open(invoice_id + ".txt", "w")
    file.write("Invoice Details\n")
    file.write("Customer Name: " + invoice_details['customer_name'] + "\n")
    file.write("Customer Address: " + invoice_details['customer_address'] + "\n")
    file.write("Customer Contact Number: " + str(invoice_details['customer_contact']) +
"\n")
    file.write("\nFurniture Purchased:\n")

    total_amount = 0
    for item in invoice_details['items']:
        file.write("ID: " + item['id'] + ", " +
                   "Manufacturer: " + item['manufacturer'] + ", " +
                   "Product: " + item['product'] + ", " +
                   "Quantity: " + str(item['quantity']) + ", " +
                   "Price: $" + str(item['price']) + "\n")
        total_amount += item['price'] * item['quantity']

    vat_amount = total_amount * 0.13
    shipping_cost = invoice_details['shipping_cost']
    total_amount_with_vat_shipping = total_amount + vat_amount + shipping_cost

    file.write("\nSubtotal: $" + str(total_amount) + "\n")
    file.write("VAT (13%): $" + str(vat_amount) + "\n")
    file.write("Shipping Cost: $" + str(shipping_cost) + "\n")
    file.write("Total Amount: $" + str(total_amount_with_vat_shipping) + "\n")
    file.close()
```

```
def generate_invoice_id():
    """
    Generate a unique invoice ID.
    """
    now = datetime.datetime.now()
    return "INVOICE" + now.strftime("%Y_%m_%d_%H_%M_%S")
```

operation.py

```
def decrease_quantity(furniture_list, furniture_id, quantity):
    """
    Decrease the quantity of the specified furniture in the inventory.
    """
    for i in range(len(furniture_list)):
        if furniture_list[i][0] == furniture_id:
            if int(furniture_list[i][3]) >= quantity:
                furniture_list[i][3] = str(int(furniture_list[i][3]) - quantity)
                return True
            else:
                print("Not enough quantity available.")
                return False
    print("Furniture ID not found.")
    return False


def increase_quantity(furniture_list, furniture_id, quantity):
    """
    Increase the quantity of the specified furniture in the inventory.
    """
    for i in range(len(furniture_list)):
        if furniture_list[i][0] == furniture_id:
            furniture_list[i][3] = str(int(furniture_list[i][3]) + quantity)
            return True
    print("Furniture ID not found.")
    return False

def check_availability(furniture_list, furniture_id):
    """
    Check the availability of furniture based on the ID.
    """
    for i in range(len(furniture_list)):
        if furniture_list[i][0] == furniture_id:
            if int(furniture_list[i][3]) > 0:
                return "Available"
            else:
                return "Not Available"
    print("Furniture ID not found.")
    return "Not Available"
```