CS4051NI/CC4059NI
Fundamental of Computing

60% Individual Coursework - 1

2023-24 Spring

Student Name: Anushma Kunwar

London Met ID: 22085497

College ID: NP01CP4S230051

Group: C19

Assignment Due Date: Monday, July 3, 2023

Assignment Submission Date: Friday, September 29, 2023

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Contents

1.Introduction 4
1.1 Intoduction to the project: 4
1.2 Tools used to complete this project: 4
IDLE 4
Diagrams.net 5
1.3 Goals and objectives: 6
2.Discussion and Analysis 6
2.1 Algorithm 6
Flowchart 8
Pseudocode 12
2.4 Data Structures: 17
Dictionary 17
List 17
Tuple 18
Set 18
Program 18
Testing 22
Test 1 22
Test 2: 23
Test 3: 24
Test 4 26
Test 5 28
Conclusion 31
Bibliography 32
Appendix 32

Table of Figures

Figure 1:IDLE LOGO 4
Figure 2:Draw.ioLogo 5
Figure 3:Ms Word 6
Figure 4:Flowchart 11
Figure 5:Dictionary used in program 17
Figure 6:List used in program 18
Figure 7:Choosing from these display 19
Figure 8:Renting process 19
Figure 9:Showing Rented Invoice 20
Figure 10:Returning Process 20
Figure 11:Showing Returned Text File 21
Figure 12:Showing exit message 21
Figure 13:Entering unappropriate value 22
Figure 14:Entering non existed value 24
Figure 15:Showing renting process of diff items 25
Figure 16:Showing information of rented equipments in shell 25
Figure 17:Showing file creation if renting invoice 25
Figure 18:Showing all the details of customers after renting in file 26
Figure 19:Retrning process 27
Figure 20:File creation of returning process 27
Figure 21:Showing all details of returning bills in text file 28
Figure 22:Renting process 29
Figure 23:Showing rented files 30
Figure 24:Returned process 30
Figure 25:Showing Returned detials in a file 30
Figure 26:Availabe equipments after update in stock 31

1.Introduction

1.1 Intoduction to the project:

This report and coursework is fully related to our knowlwdge based on python programming language.Python is a widely used general-purpose ,high level programming language.It is a dynamic,interpreted language.with several advantages,including programmers to read and write syntax .It’s source file use the ”.py” extension and are called as modules.It can work on different platforms.Python can be treated in a procedural way or an object-oriented way .We have completed this in a object-oriented way.Its practically lies in machine learning and AI,data analystics,data visualization,programming applications,web- development,game development,language development,finance sector,search engine optimization and so on. (Python:Geeksforkeeks)

1.2 Tools used to complete this project:

IDLE

Figure 1:IDLE LOGO

I have used IDLE 3.11 for coding part in this project.IDLE stands for Python’s Integrated Devlopment and Learning Environment.It allows us to write Python code there.It consists of Python shell window with coloring of code input ,output, and error messages, and multi-window text editor with multiple undo ,Python colorizing ,smart indent,call tips,auto completetiona and more other features.It is a goodlearning platform since it is lightweight and easy to use.It makes the language learning curve fast and Python programming friendly because it is a very simple editor that helps us to code .So,I have choose this software due to its simplicity for coding. (docs.python)

Diagrams.net

Figure 2:Draw.ioLogo

Diagram.net is a cross-platform graph drawing software which is a user friendly web application and is very easy to use.I have used this software to design flowchart for my report part.It was also used as a groundwork for creating the program.

Figure 3:Ms Word

Figure3:MS word logo

MS word is the most popular documentation program.It has many advanced features which allows us to format and edit your files and documentation in the best possible way.It creates user friendly environment so it is most popular software for creating reports.

1.3 Goals and objectives:

The focus of this project is to create customer rental shop which maintains information about the various available equipments in a text file.By using the python IDLE 3.11 this coursework is done.I had developed an application which reads all the text file and display the equipment that are available for renting purposes.Then,with each transcation date of rent and return ,a information will be kept in a notepad and will display information in the file.After each transaction,i.e the equipment rented,the quantity in the stock will decrease.and if the equipment is return then the quantity of the stock will increases.

2.Discussion and Analysis

2.1 Algorithm

An algorithm is a set of instructions for solving a probem and accomplishing a task step by step.

Step 1:Start

Step 3:Dsiplay welcome message in display

Step 5:Create variable check

Step 6:Input check if the transactions_dir exists or not;create if not

Step 6:Read equipment information from the file into the equipment_dict

Stp 7:Display a menu with options to rent,return or exit

Step 7:If check =1,call rent equipment()method,if check=2,call return equipment()methods,if check =3,exit the program and check if it is correct or display invalid input to customer.

Step 8:Initialize rented_items as empty list to store rented items.

9:Check if there is enough quantity of the selected equipment in equipment_dict

Step 10:If the quantity is enough qty=qty-rented quantity

Step 11:Store all the details of re nted equipments in rented_items.

Step 12:If userContinue than repeat step 8 to step 11

Step 13:If user press 0 then,create input variabes for customer details

Step 14:Input the costumers details and write them in transaction file.

Step 15:Again,Retuen to step 7

Step 16:Create variable and make user to input

Step 17:Make user to select items and quantites to return

Step 18:Calulate the rental duration and any fines for the late returns

Step 19:If noOfdays>5 then impose a fine Create variable fine =(rent_days – 5)*10

Step 20:Update the equipment quantities and generate a file for costumer return invoice.

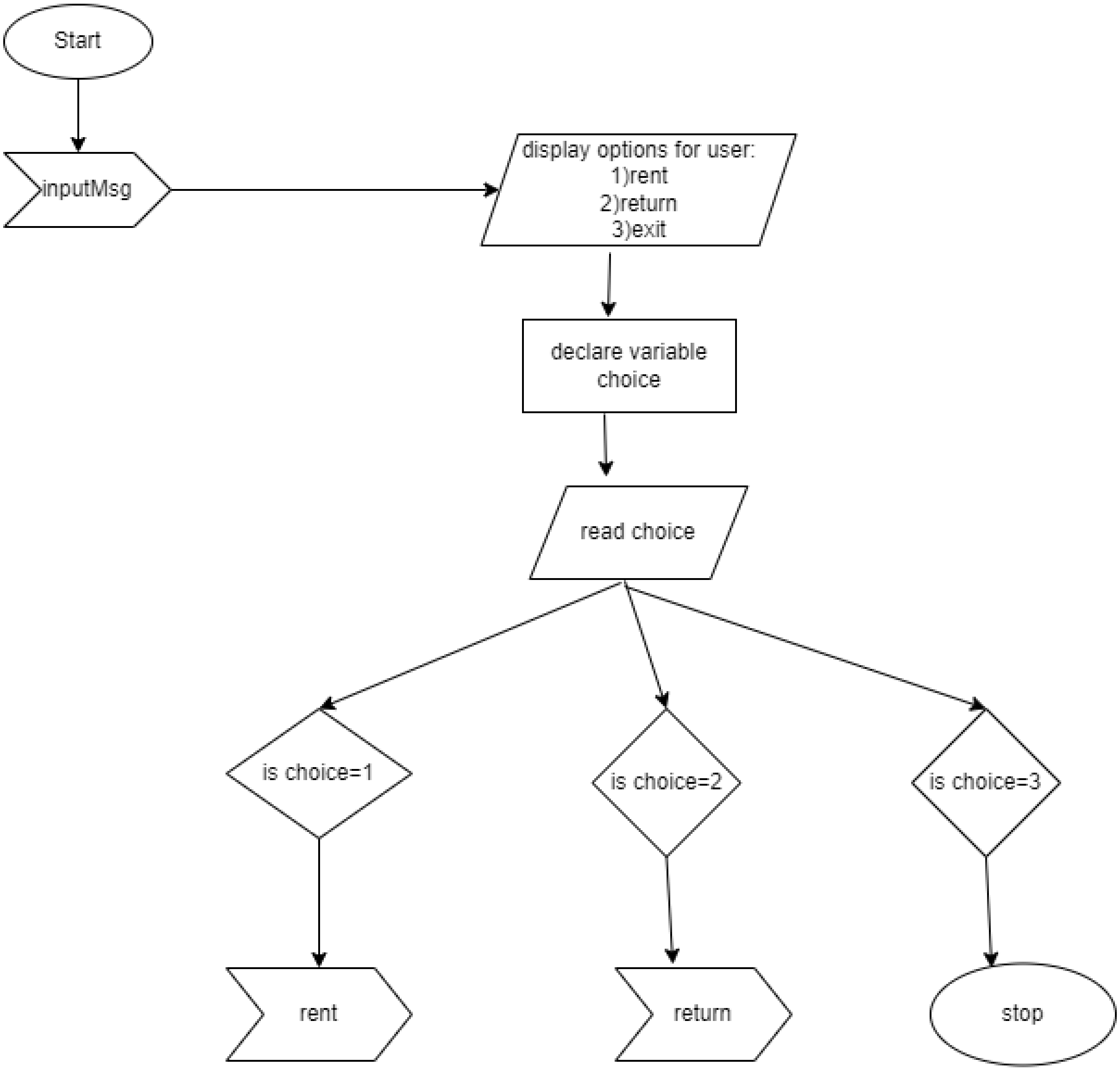
Step22:Write customer information,returned equipment details ,refund amount, and any fines to the return invoice file

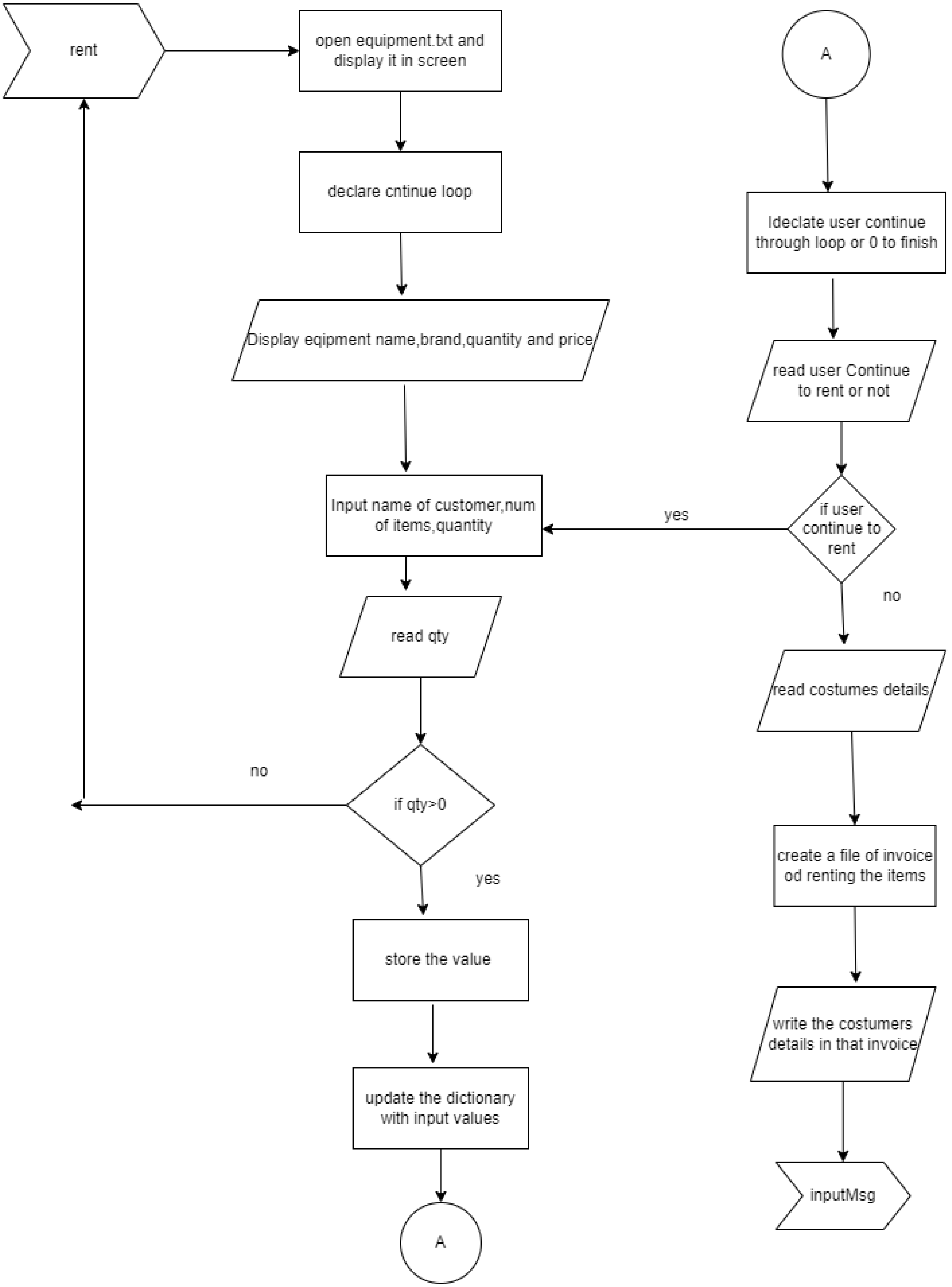
Step23:Go to step 3

Step24:If choice=3,exit the application with a message”Equipment information updated!Thankyou”

Flowchart

A flowchart is a diagram that separate steps of a process in sequential order.


Start
inputMsg
display options for user:
1)rent
2)return
3)exit
declare variable choice
read choice
is choice=1
is choice=2
is choice=3
rent
return
stop


rent
open equipment.txt and display it in screen
declare cntinue loop
Display eqipment name,brand,quantity and price
Input name of customer,num of items,quantity
read qty
if qty>0
no
yes
store the value
update the dictionary with input values
A
A
Ideclate user continue through loop or 0 to finish
read user Continue to rent or not
if user continue to rent
yes
no
read costumes details
create a file of invoice od renting the items
write the costumers details in that invoice
inputMsg

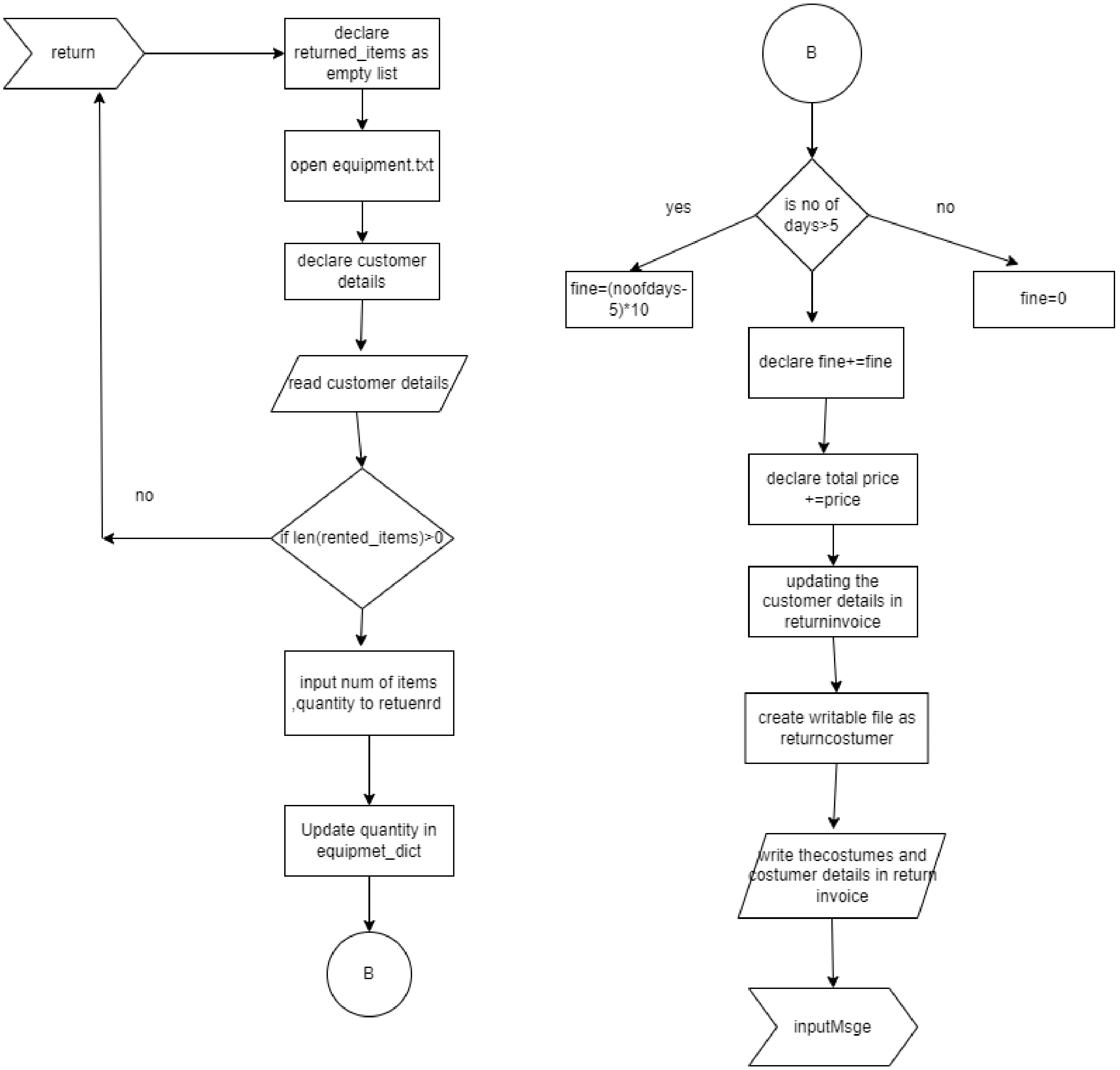

return
declare returned_items as empty list
open equipment.txt
declare customer details
read customer details
if len(rented_items)>0
no
input num of items ,quantity to retuenrd
Update quantity in equipmet_dict
B
B
is no of days>5
yes
no
fine=(noofdays-5)*10
fine=0
declare fine+=fine
declare total price +=price
updating the customer details in returninvoice
create writable file as returncostumer
write thecostumes and costumer details in return invoice
inputMsge

Figure 4:Flowchart

Pseudocode

Pseudocode is an informal way of programming description that doesnot require any strict programming language syntax or underlying technology considerations.It is used for creating an outline or a rough draft of a program.

```
DEFINE function as input_text:
 RETURN welcome_message


DEFINE function as main
    DISPLAY a welcome message as “Welcome to Equipment Rental Shop”
    DISPLAY the main option as
      1.Rent Equipment
      2.Return Equipment
      3.Exit
ENTER a choice and store it in dict
IF choice is equal to 1
    DISPLAY  All the available equipments
   CALL rent function
ELSE if check is equal to 2
    CALL return function
ELSE IF check is equal to 3:
    DISPLAY” Equipment information updated.Goodbye!”
    EXIT the file
ELSE
    DISPLAY “Please input from given choices.
DEFINE function as rent items
    SET rented_items as empty list to store items
    SET rent_date as current date and time
   CALL readFile function
    OPEN equipment.txt as readable file
    RETURN each line of equipmentlist as list item
   CLOSE equipmentList
   STORE rented equipments in a rented_items dict
   SET costumers to continue following process or 0 to stop
         INPUT name,quantity of the available equipments and store it in rented_items
         MODIFY rented_items
   IF availabe qty>=quantity
         UPDATE available quantity-quantity
          MODIFY equipment name,qty price in equipment dict
RETURN rented_items
ELSE
        DISPLAY”Not enough equipment available”
CALL generate_rentinvoice function
     SET timestamp as current date and time
    SET invoice name as customer name
    OPEN invoice_path in write mode
    WRITE “Customer:customer_name”to file
   UPDATE “Equipment as equipment (brand)” to file
   UPDATE “Quantity as quantity” to file
  UPDATE “Price per item as price”to file
  UPDATE increase total cost by price*quantity to file
  UPDATE Rent date to file
  UPDATE Total cost  to file
  UPDATE Rental Duration as 5days to file
  RETURN invoice_path
  CLOSE equipmentlist
```

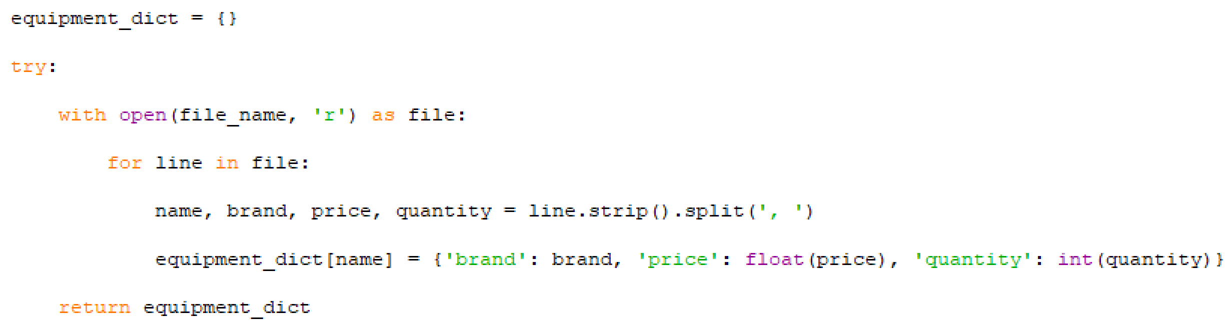
```
ENTER usercontinue forRent or 0 to stop
IF usercontinue forRent
    ENTER customername and store in rented_items
   ENTER number of items to rent
  ENTER quantity of items to rent
 UPDATE the details of customer in rent invoice file
 CLOSE the rentinvoice file

DEFINE function as return_items
   INITIALIZE returned_items as empty list
   INITIALIZE rented_items a empty list
   INITIALIZE return_date_time as empty list
   INITIALIZE rent_days as 0
   INITIALIZE customer_name s 0
   INITIALIZE fine as 0
   INITIALIZE fined_days as 0
TRY
        OPEN  rent invoice_path read as a file
UPDATE rent_days as datetime.datetime.now ()-datetime.datetime.striptime(rent_date)
IF rent_days>5
     CALCULATE fine=(rent_days-5)*10
ELSE
      FINE=0
      Fined_days=rent_days-5
 EXTRACT rented items information
 PRINT ("Rented itmes for customer") in shell
 SET items_to_return as empty
 IF  len(rented_items) > 0:
    procceed True
    WHILE procce:
          Enter the number of items to return
     IF "q" in choices
          BREAK
          FOR choice in choices:
  TRY
           CONVERT choice into int
             IF 1 is less or equal to choice or samller oor equal to len(rented_items):
             Quantity_to_return=Enter the quantity of item
 ELSE
              DISPLAY"Invalid quantity"
              Proceed =True
 ELSE
            DISPLAY"Invalid choice"
EXCEPT ValueError
          DISPLAY "Invalid input,choice
         BREAK
ELSE
    There are'nt any equipments to rent
UPDATE equipment quantities
 IF equipment in equipment_dict
       Equipment_dict quantity+=quantity to return
OPEN invoice _path write as file
      Tst_total_cost=0
     File.write as costumer name
     FOR equipment,brand,price,quantitytotal_cost,rent_date in rented_items
     UPDATE   file.write(f"Equipment: {equipment} ({brand}\n")
        UPDATE  file.writef"Quantity: {remaining}\n"
        UPDATE file.write(f"Price per item: ${price}\n"
        UPDATE  file.writef"Cost: ${price * remaining}\n"
        UPDATE   file.write '-' * 20 + '\n'
        UPDATE    tst_total_cost += price * remaining
        UPDATE   file.write f”Rental Duration :5days”
EXCEPT FilenotFoundError
        DISPLAY”Invoice not found”
RETURN returned_costumes,customer_name,,fine,fined days
```

2.4 Data Structures:

A data structure is a specialized format for organizing,processing,retrieving and storing of data.Data structure make it easy for users to access and work with the data they need in appropriate ways.They are the foundation of any programming language. In comparision to other programming language python makes it easier to learn the fundamentals of these data structures.The data structures that are used in this code are:

Dictionary

A dictionary is an unsorred collection of data which contains information in key-value pairs.It is surrounded by curly bracket.Dictionary is used in this program to store the item from the supplies,txt file.In this program we have used equipment_dict to store the information about available equipments.



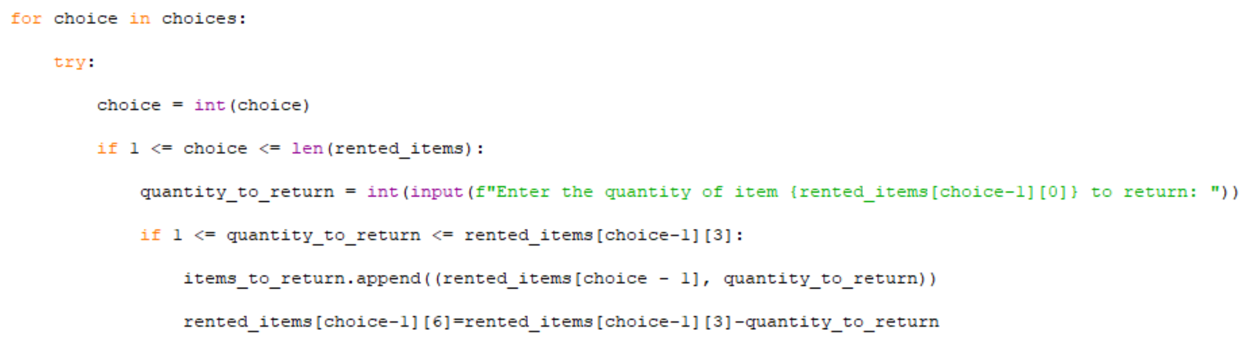
```
equipment_dict = {}

try:

    with open(file_name, 'r') as file:

        for line in file:

            name, brand, price, quantity = line.strip().split(', ')

            equipment_dict[name] = {'brand': brand, 'price': float(price), 'quantity': int(quantity)}

    return equipment_dict
```

Figure 5:Dictionary used in program

List

A list is an ordered data structure that stores elements sequentially and can be accessed by the index of the elements.It is also mutable It is created by the use of square brackets.

Here,we have used list to store the information about items that have been rented and returned by customer i.e rented_items or returned_items.



```
for choice in choices:
    try:
        choice = int(choice)
        if 1 <= choice <= len(rented_items):
            quantity_to_return = int(input(f"Enter the quantity of item {rented_items[choice-1][0]} to return: "))
            if 1 <= quantity_to_return <= rented_items[choice-1][3]:
                items_to_return.append((rented_items[choice - 1], quantity_to_return))
                rented_items[choice-1][6]=rented_items[choice-1][3]-quantity_to_return
```

Figure 6:List used in program

Other data structures are:

Tuple

Tuples are a data type which is ordered,or unchangeable,allow duplicates and can store different data types of single variable.()

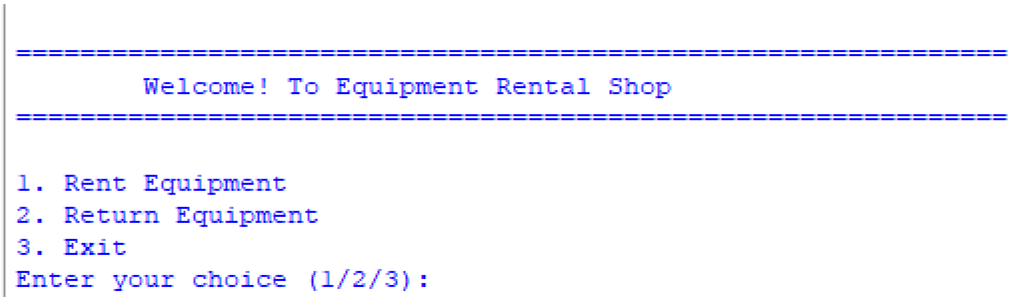
Set

They are the data type which is unordered,unchnagable,doesn't allow duplicates and can store different data types .{}

Program

This program is performed by doing a chunk of codes in different files and calling them and reusing them. I have performed mostly by using different loops a powerful technique of calling functions in python.

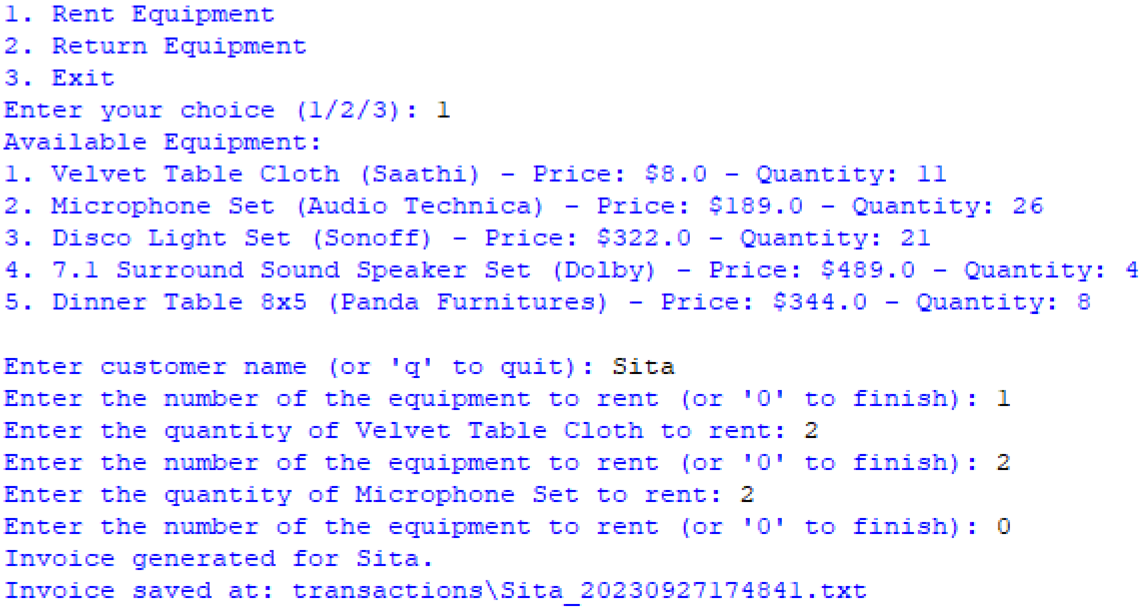
When the application is opend at first the greeting message appears. Then an option message will popup with option 1 to rent the equipments,option 2 to return the equipment and option 3 to exit form that.



```
==============================================================
        Welcome! To Equipment Rental Shop
==============================================================

1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3):
```

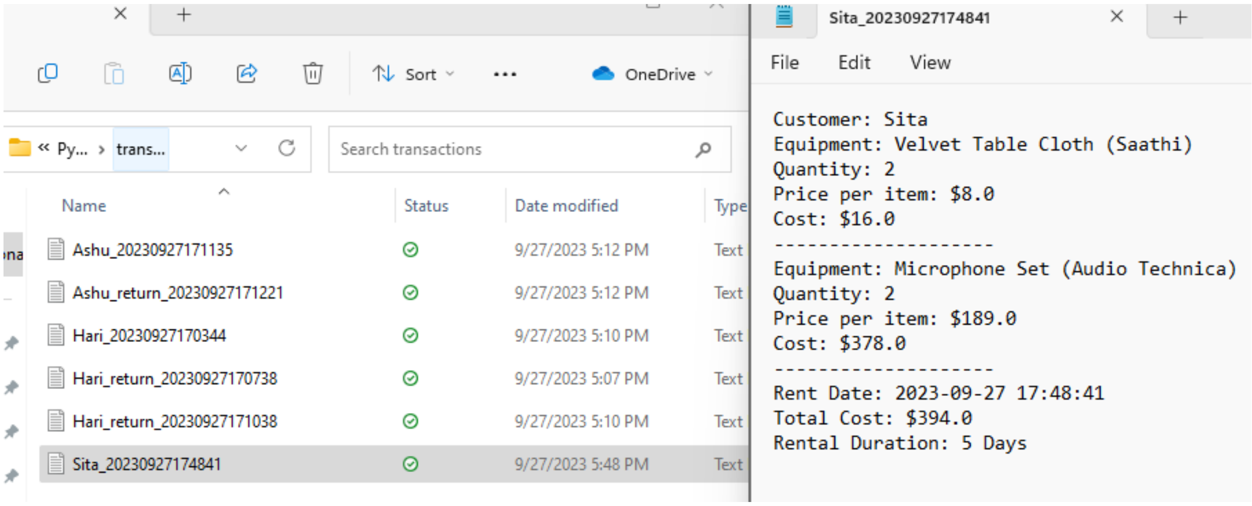
Figure 7:Choosing from these display

If the user decides to rent a n equipment,1 is pressed and availabe items is shown at the display. Filling up costumers detail they can rent many items at a time.At last if they want to stop the renting process they can press 0 after they are done by renting equipments..



```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 11
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 26
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 4
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8

Enter customer name (or 'q' to quit): Sita
Enter the number of the equipment to rent (or '0' to finish): 1
Enter the quantity of Velvet Table Cloth to rent: 2
Enter the number of the equipment to rent (or '0' to finish): 2
Enter the quantity of Microphone Set to rent: 2
Enter the number of the equipment to rent (or '0' to finish): 0
Invoice generated for Sita.
Invoice saved at: transactions\Sita_20230927174841.txt
```

Figure 8:Renting process

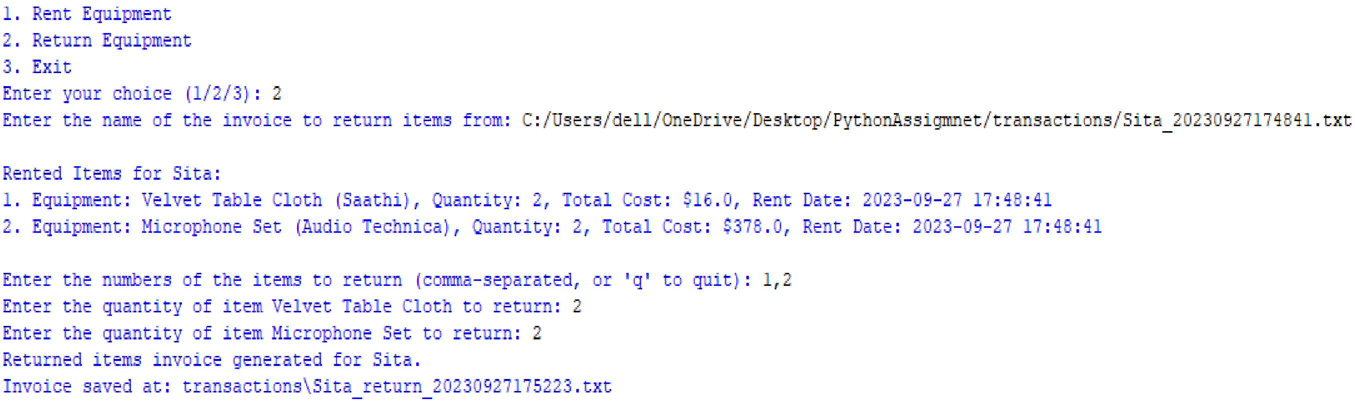


Name	Status	Date modified	Type
Ashu_20230927171135	✓	9/27/2023 5:12 PM	Text
Ashu_return_20230927171221	✓	9/27/2023 5:12 PM	Text
Hari_20230927170344	✓	9/27/2023 5:10 PM	Text
Hari_return_20230927170738	✓	9/27/2023 5:07 PM	Text
Hari_return_20230927171038	✓	9/27/2023 5:10 PM	Text
Sita_20230927174841	✓	9/27/2023 5:48 PM	Text

```
Customer: Sita
Equipment: Velvet Table Cloth (Saathi)
Quantity: 2
Price per item: $8.0
Cost: $16.0
--------------------
Equipment: Microphone Set (Audio Technica)
Quantity: 2
Price per item: $189.0
Cost: $378.0
--------------------
Rent Date: 2023-09-27 17:48:41
Total Cost: $394.0
Rental Duration: 5 Days
```

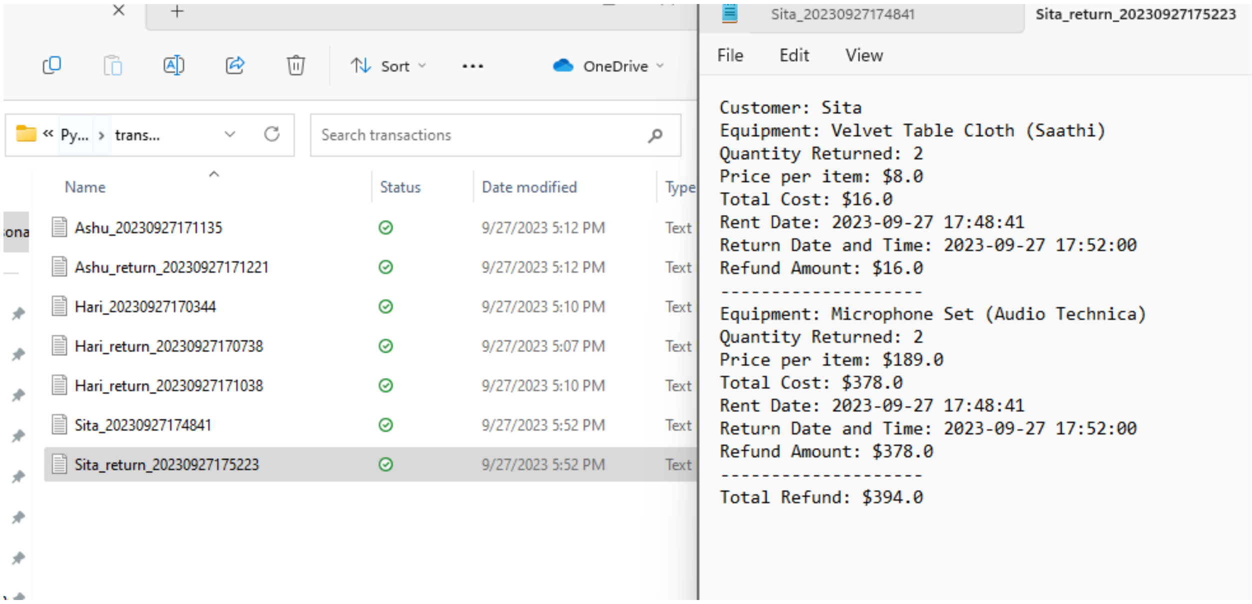
Figure 9:Showing Rented Invoice

Again,If the costumers want to return the items that they have rented then they can press return equipment choice i.e option 2 By filling details to return and a note or invoice will be genereated in a text file and that returned equipment quantity will be updated in the stock.



```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 2
Enter the name of the invoice to return items from: C:/Users/dell/OneDrive/Desktop/PythonAssigmnet/transactions/Sita_20230927174841.txt

Rented Items for Sita:
1. Equipment: Velvet Table Cloth (Saathi), Quantity: 2, Total Cost: $16.0, Rent Date: 2023-09-27 17:48:41
2. Equipment: Microphone Set (Audio Technica), Quantity: 2, Total Cost: $378.0, Rent Date: 2023-09-27 17:48:41

Enter the numbers of the items to return (comma-separated, or 'q' to quit): 1,2
Enter the quantity of item Velvet Table Cloth to return: 2
Enter the quantity of item Microphone Set to return: 2
Returned items invoice generated for Sita.
Invoice saved at: transactions\Sita_return_20230927175223.txt
```

Figure 10:Returning Process

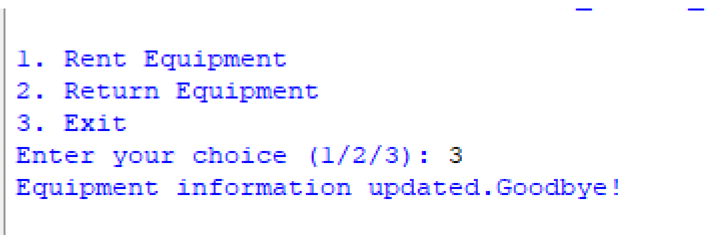


Name	Status	Date modified	Type
Ashu_20230927171135	✓	9/27/2023 5:12 PM	Text
Ashu_return_20230927171221	✓	9/27/2023 5:12 PM	Text
Hari_20230927170344	✓	9/27/2023 5:10 PM	Text
Hari_return_20230927170738	✓	9/27/2023 5:07 PM	Text
Hari_return_20230927171038	✓	9/27/2023 5:10 PM	Text
Sita_20230927174841	✓	9/27/2023 5:52 PM	Text
Sita_return_20230927175223	✓	9/27/2023 5:52 PM	Text

```
Customer: Sita
Equipment: Velvet Table Cloth (Saathi)
Quantity Returned: 2
Price per item: $8.0
Total Cost: $16.0
Rent Date: 2023-09-27 17:48:41
Return Date and Time: 2023-09-27 17:52:00
Refund Amount: $16.0
--------------------
Equipment: Microphone Set (Audio Technica)
Quantity Returned: 2
Price per item: $189.0
Total Cost: $378.0
Rent Date: 2023-09-27 17:48:41
Return Date and Time: 2023-09-27 17:52:00
Refund Amount: $378.0
--------------------
Total Refund: $394.0
```

Figure 11:Showing Returned Text File

At last,If the customers wants to exit from this rental process then they can enter choice as Exit i.e option 3.Then,a goodbye message is displayed updating all the information of equipments.



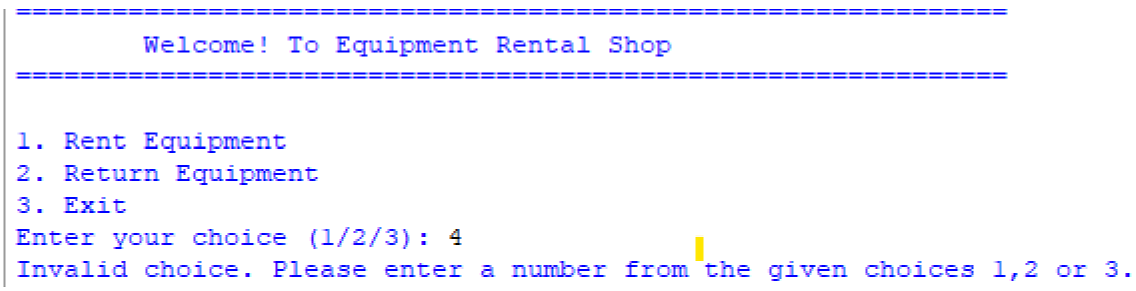
```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 3
Equipment information updated.Goodbye!
```

Figure 12:Showing exit message

Testing

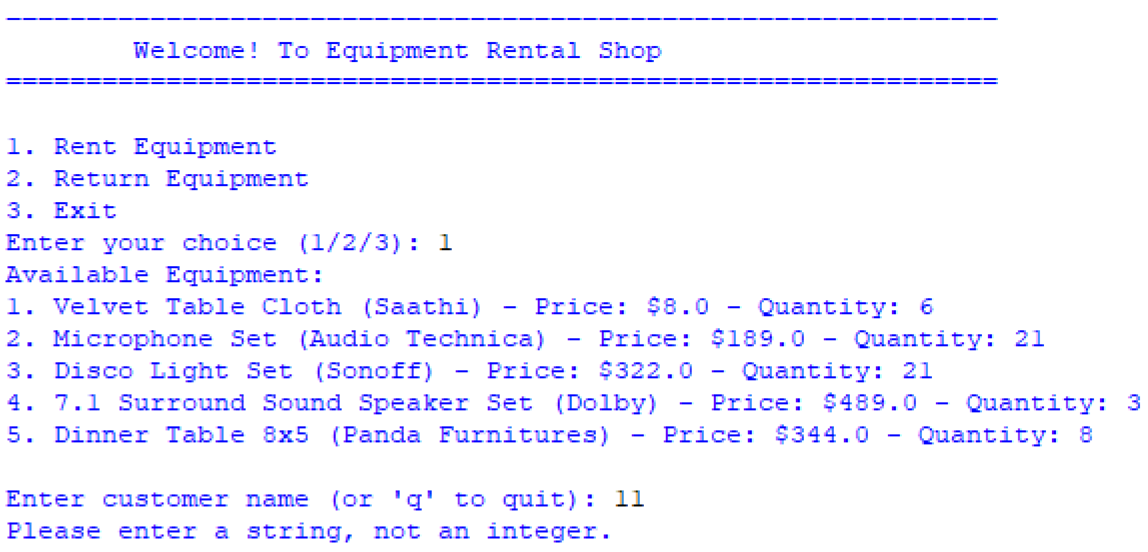
Test 1

Objective	**To show the implementation to try and except**
Action	**- Application is executed** **- Invalid number is enter**
Expected Result	**Error message showing Inavalid choice,Please enter a number of given choice**
Actual Result	**Error massege has shown in display**
Conclusion	**Yes,The test was sucessfull**



```
==============================================================
        Welcome! To Equipment Rental Shop
==============================================================

1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 4
Invalid choice. Please enter a number from the given choices 1,2 or 3.
```

Figure 13:Test 1 showing try and except



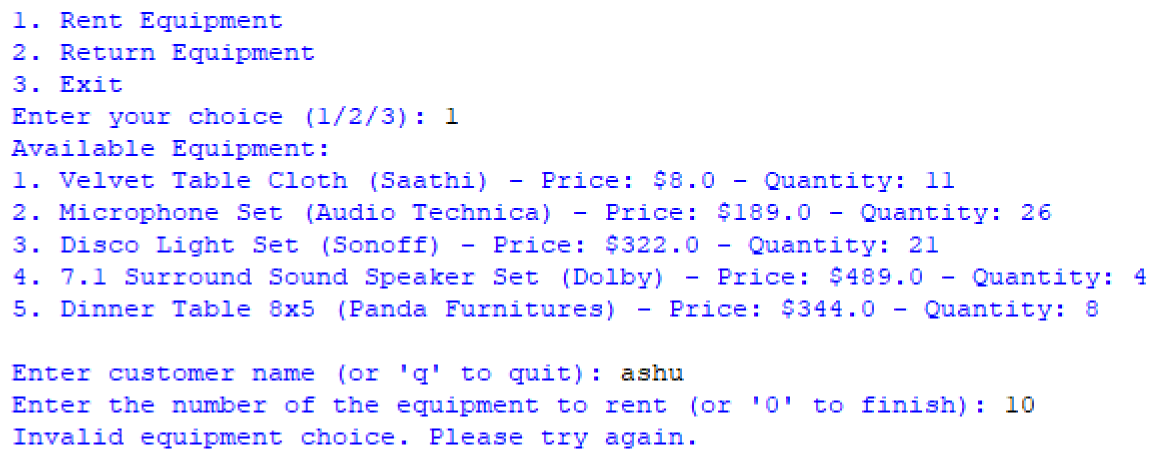
```
--------------------------------------------------------------
        Welcome! To Equipment Rental Shop
==============================================================

1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 6
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 21
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 3
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8

Enter customer name (or 'q' to quit): 11
Please enter a string, not an integer.
```

Figure 14:Test 2:Showing try except

Test 2:

Objective	**To Selection rent and returning of items**
Action	**Providing non existed value as input**
Expected Result	**Error message would be displayed Inavalid equipmet choice please try again.**
Actual Result	**Error message has displayed to try again.**

Conclusion	**The test was successfull**

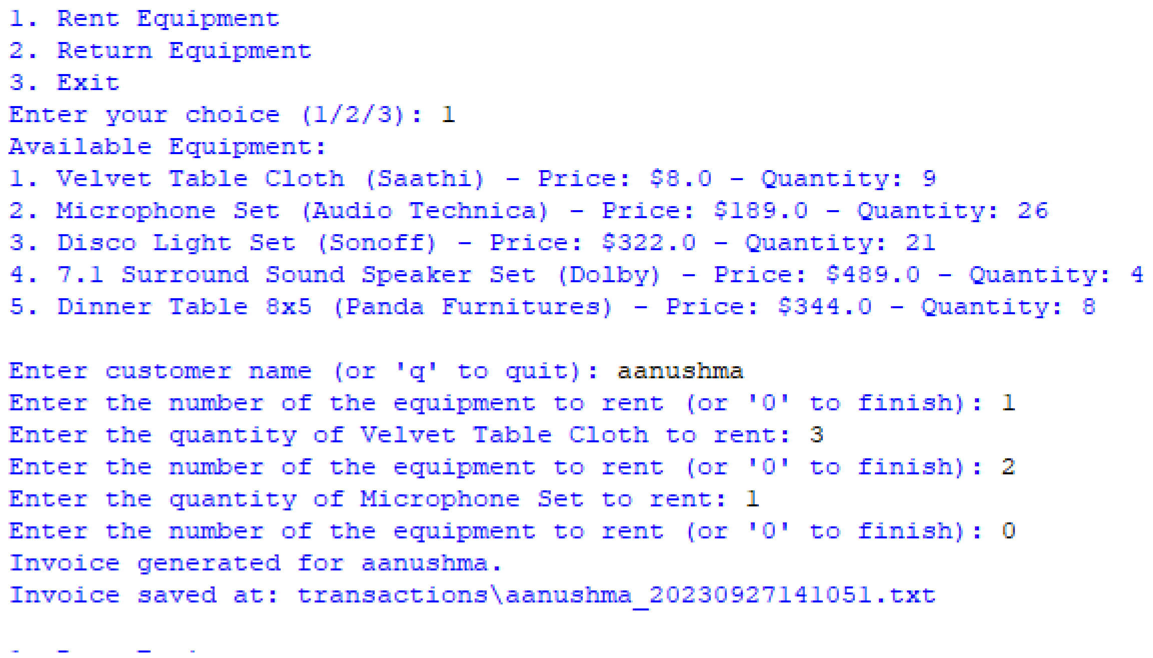


```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 11
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 26
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 4
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8

Enter customer name (or 'q' to quit): ashu
Enter the number of the equipment to rent (or '0' to finish): 10
Invalid equipment choice. Please try again.
```

Figure 15:Entering non existed value

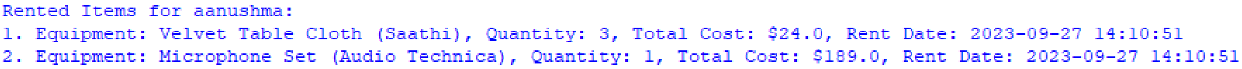
Test 3:

Objective	**File generation of item(s),Renting multiple items**
Action	**Application is executed** **1 is pressed to rent equipment** **All the necessary information are filled up**
Expected Result	**A file would be generated in a file while renting a custome and displayed in the screen as well**
Actual Reuslt	**As expected a file is generated in which the renting details os customer is stored.**
Conclusion	**The test is successfull**



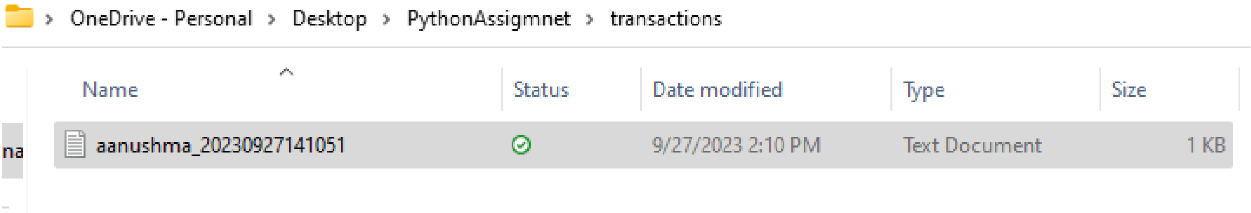
```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 9
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 26
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 4
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8

Enter customer name (or 'q' to quit): aanushma
Enter the number of the equipment to rent (or '0' to finish): 1
Enter the quantity of Velvet Table Cloth to rent: 3
Enter the number of the equipment to rent (or '0' to finish): 2
Enter the quantity of Microphone Set to rent: 1
Enter the number of the equipment to rent (or '0' to finish): 0
Invoice generated for aanushma.
Invoice saved at: transactions\aanushma_20230927141051.txt
```

Figure 16:Showing renting process of diff items



```
Rented Items for aanushma:
1. Equipment: Velvet Table Cloth (Saathi), Quantity: 3, Total Cost: $24.0, Rent Date: 2023-09-27 14:10:51
2. Equipment: Microphone Set (Audio Technica), Quantity: 1, Total Cost: $189.0, Rent Date: 2023-09-27 14:10:51
```

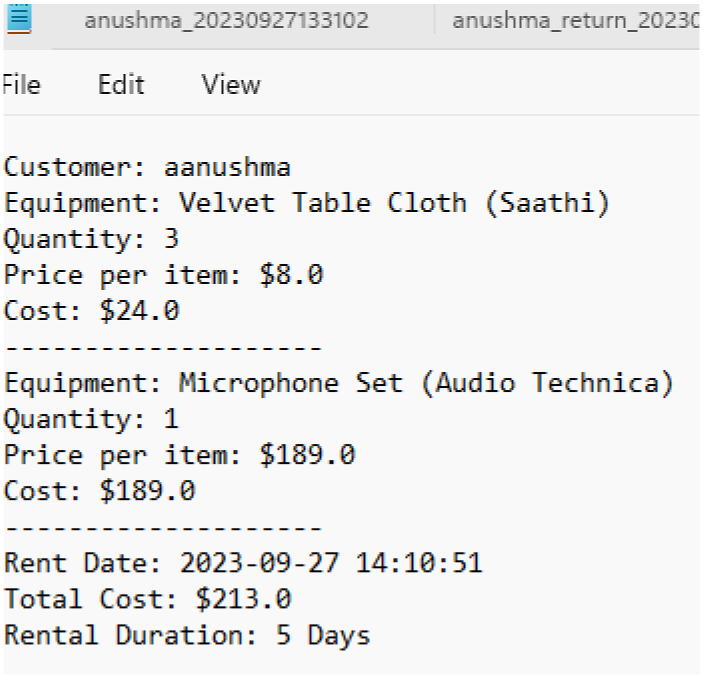
Figure 17:Showing information of rented equipments in shell



› OneDrive - Personal › Desktop › PythonAssigmnet › transactions

Name	Status	Date modified	Type	Size
aanushma_20230927141051	✓	9/27/2023 2:10 PM	Text Document	1 KB

Figure 18:Showing file creation if renting invoice

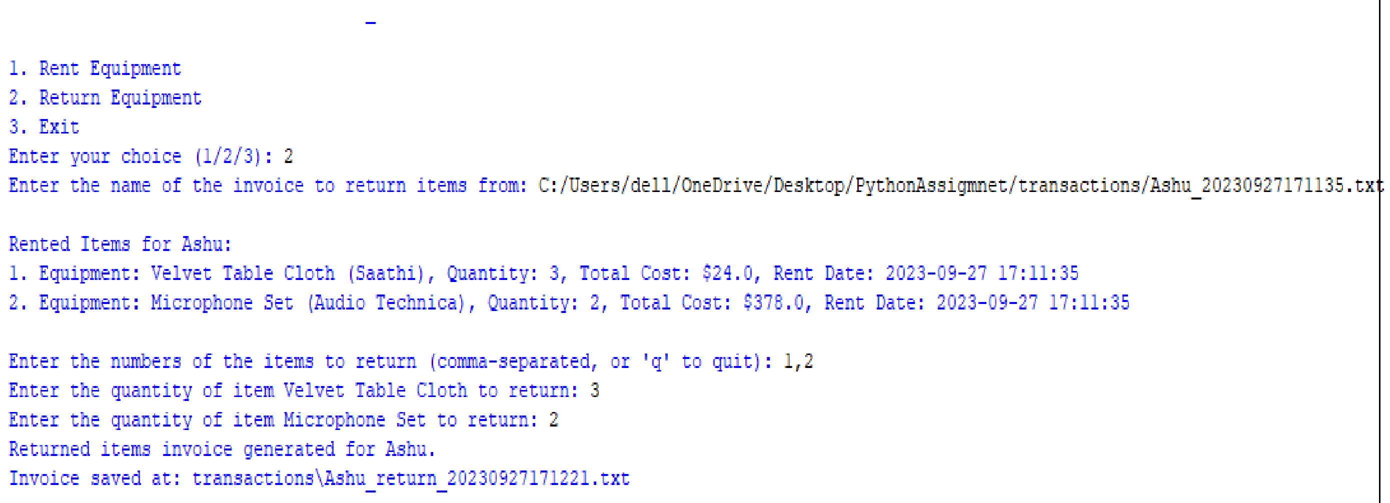


```
Customer: aanushma
Equipment: Velvet Table Cloth (Saathi)
Quantity: 3
Price per item: $8.0
Cost: $24.0
--------------------
Equipment: Microphone Set (Audio Technica)
Quantity: 1
Price per item: $189.0
Cost: $189.0
--------------------
Rent Date: 2023-09-27 14:10:51
Total Cost: $213.0
Rental Duration: 5 Days
```

Figure 19:Showing all the details of customers after renting in file

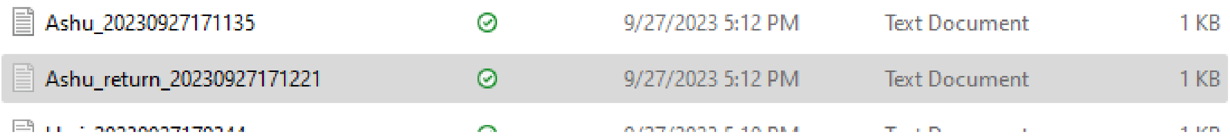
Test 4
To test generation of file after returning equipment

Objective	**To test file generation of return process of items**
Action	**Application is executed 2 is pressed so we can return All necessary information had filled**
Expected Result	**A returned file would generate in the folder with details of customer all equipments.**
Actual Result	**A file was genereated which contains information about to be returned equipments**
Conclusion	**The test was successfull**



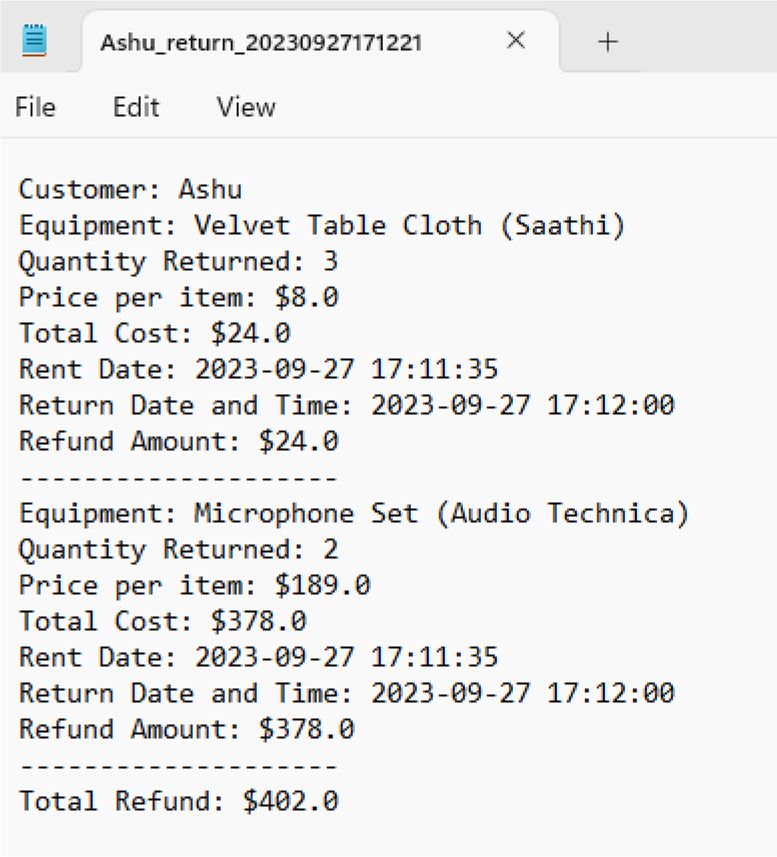
```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 2
Enter the name of the invoice to return items from: C:/Users/dell/OneDrive/Desktop/PythonAssigmnet/transactions/Ashu_20230927171135.txt

Rented Items for Ashu:
1. Equipment: Velvet Table Cloth (Saathi), Quantity: 3, Total Cost: $24.0, Rent Date: 2023-09-27 17:11:35
2. Equipment: Microphone Set (Audio Technica), Quantity: 2, Total Cost: $378.0, Rent Date: 2023-09-27 17:11:35

Enter the numbers of the items to return (comma-separated, or 'q' to quit): 1,2
Enter the quantity of item Velvet Table Cloth to return: 3
Enter the quantity of item Microphone Set to return: 2
Returned items invoice generated for Ashu.
Invoice saved at: transactions\Ashu_return_20230927171221.txt
```

Figure 20:Retrning process



```
Ashu_20230927171135           9/27/2023 5:12 PM    Text Document    1 KB
Ashu_return_20230927171221    9/27/2023 5:12 PM    Text Document    1 KB
```

Figure 21:File creation of returning process



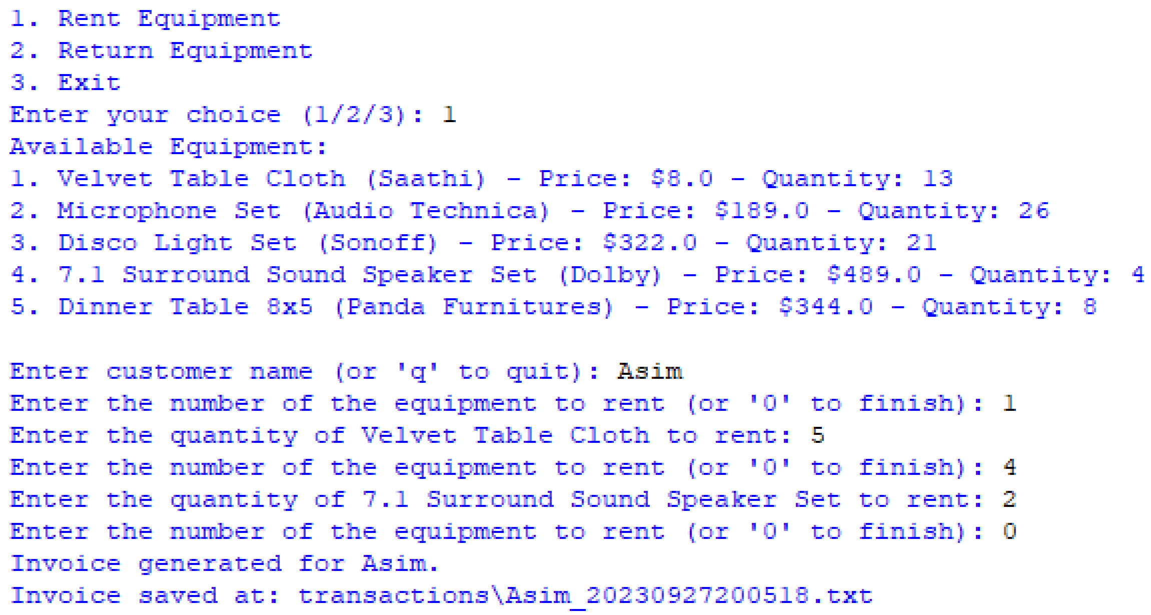
```
Customer: Ashu
Equipment: Velvet Table Cloth (Saathi)
Quantity Returned: 3
Price per item: $8.0
Total Cost: $24.0
Rent Date: 2023-09-27 17:11:35
Return Date and Time: 2023-09-27 17:12:00
Refund Amount: $24.0
--------------------
Equipment: Microphone Set (Audio Technica)
Quantity Returned: 2
Price per item: $189.0
Total Cost: $378.0
Rent Date: 2023-09-27 17:11:35
Return Date and Time: 2023-09-27 17:12:00
Refund Amount: $378.0
--------------------
Total Refund: $402.0
```

Figure 22:Showing all details of returning bills in text file

Test 5

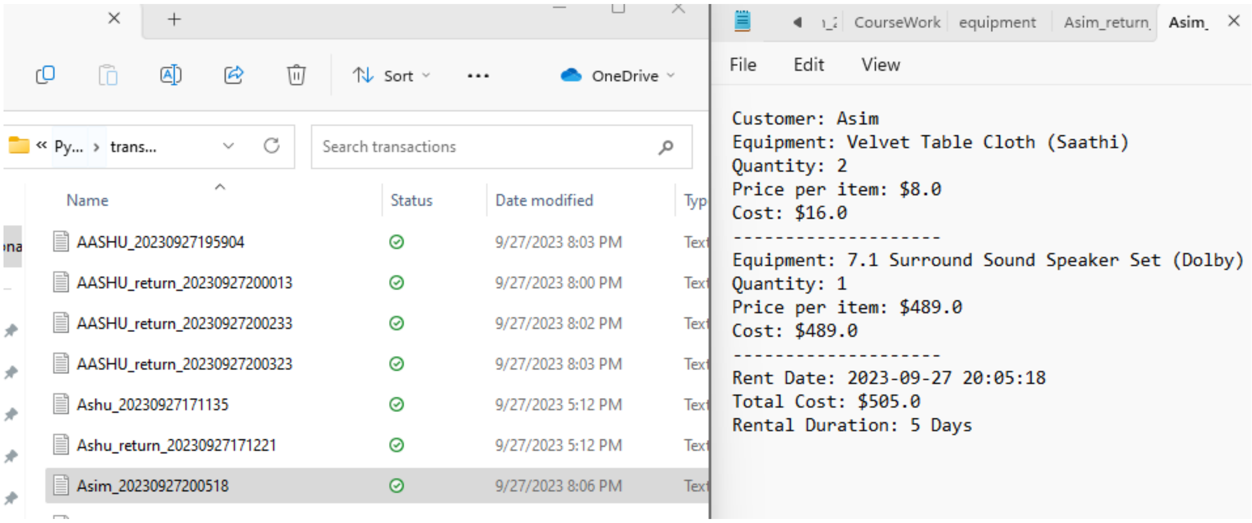
Objective	**To show update in stock items**
Action	**Application is executed 1 is entered to rent a equipment All details are filled . 5 Sathi are and 2Dolby are rented Now it is renting process is stoped Now,2 is pressed to return the equipment All details are filled appropriately and we have returned all the equipments. 3 sathi and 1 Dolby are returned.**
Expected Result	**The quantity in stock would be change**

Actual Result	**The quantity in the stock is chnaged**
Conclusion	**The test is successfull**



```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 13
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 26
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 4
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8

Enter customer name (or 'q' to quit): Asim
Enter the number of the equipment to rent (or '0' to finish): 1
Enter the quantity of Velvet Table Cloth to rent: 5
Enter the number of the equipment to rent (or '0' to finish): 4
Enter the quantity of 7.1 Surround Sound Speaker Set to rent: 2
Enter the number of the equipment to rent (or '0' to finish): 0
Invoice generated for Asim.
Invoice saved at: transactions\Asim_20230927200518.txt
```

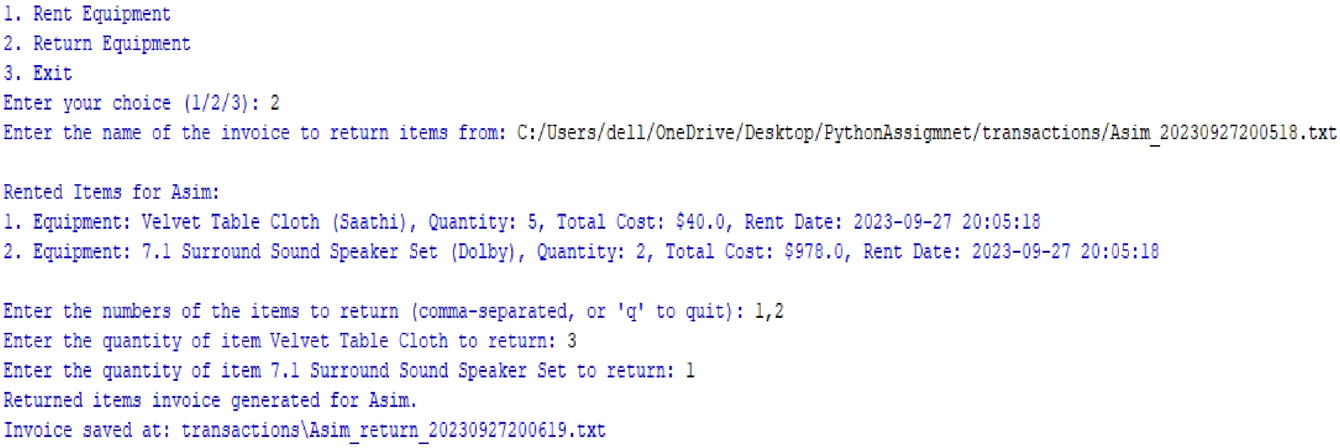
Figure 23:Renting process



Name	Status	Date modified	Typ
AASHU_20230927195904	✓	9/27/2023 8:03 PM	Text
AASHU_return_20230927200013	✓	9/27/2023 8:00 PM	Text
AASHU_return_20230927200233	✓	9/27/2023 8:02 PM	Text
AASHU_return_20230927200323	✓	9/27/2023 8:03 PM	Text
Ashu_20230927171135	✓	9/27/2023 5:12 PM	Text
Ashu_return_20230927171221	✓	9/27/2023 5:12 PM	Text
Asim_20230927200518	✓	9/27/2023 8:06 PM	Text

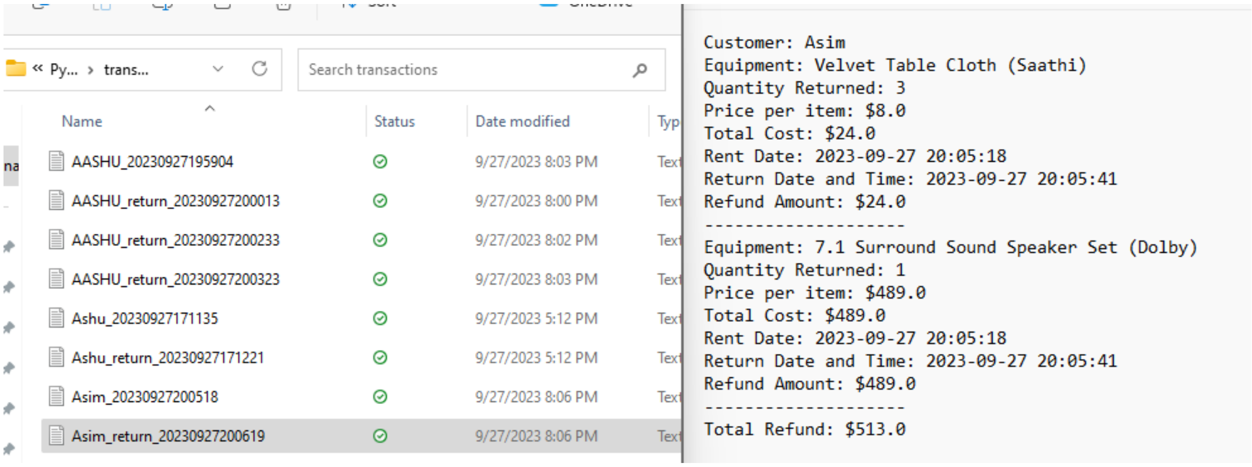
```
Customer: Asim
Equipment: Velvet Table Cloth (Saathi)
Quantity: 2
Price per item: $8.0
Cost: $16.0
--------------------
Equipment: 7.1 Surround Sound Speaker Set (Dolby)
Quantity: 1
Price per item: $489.0
Cost: $489.0
--------------------
Rent Date: 2023-09-27 20:05:18
Total Cost: $505.0
Rental Duration: 5 Days
```

Figure 24:Showing rented files



```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 2
Enter the name of the invoice to return items from: C:/Users/dell/OneDrive/Desktop/PythonAssigmnet/transactions/Asim_20230927200518.txt

Rented Items for Asim:
1. Equipment: Velvet Table Cloth (Saathi), Quantity: 5, Total Cost: $40.0, Rent Date: 2023-09-27 20:05:18
2. Equipment: 7.1 Surround Sound Speaker Set (Dolby), Quantity: 2, Total Cost: $978.0, Rent Date: 2023-09-27 20:05:18

Enter the numbers of the items to return (comma-separated, or 'q' to quit): 1,2
Enter the quantity of item Velvet Table Cloth to return: 3
Enter the quantity of item 7.1 Surround Sound Speaker Set to return: 1
Returned items invoice generated for Asim.
Invoice saved at: transactions\Asim_return_20230927200619.txt
```

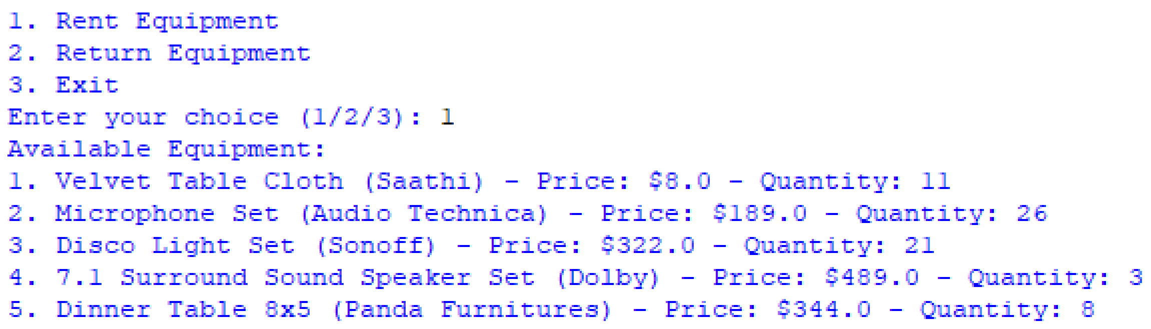
Figure 25:Returned process



Name	Status	Date modified	Typ
AASHU_20230927195904	✓	9/27/2023 8:03 PM	Text
AASHU_return_20230927200013	✓	9/27/2023 8:00 PM	Text
AASHU_return_20230927200233	✓	9/27/2023 8:02 PM	Text
AASHU_return_20230927200323	✓	9/27/2023 8:03 PM	Text
Ashu_20230927171135	✓	9/27/2023 5:12 PM	Text
Ashu_return_20230927171221	✓	9/27/2023 5:12 PM	Text
Asim_20230927200518	✓	9/27/2023 8:06 PM	Text
Asim_return_20230927200619	✓	9/27/2023 8:06 PM	Text

```
Customer: Asim
Equipment: Velvet Table Cloth (Saathi)
Quantity Returned: 3
Price per item: $8.0
Total Cost: $24.0
Rent Date: 2023-09-27 20:05:18
Return Date and Time: 2023-09-27 20:05:41
Refund Amount: $24.0
--------------------
Equipment: 7.1 Surround Sound Speaker Set (Dolby)
Quantity Returned: 1
Price per item: $489.0
Total Cost: $489.0
Rent Date: 2023-09-27 20:05:18
Return Date and Time: 2023-09-27 20:05:41
Refund Amount: $489.0
--------------------
Total Refund: $513.0
```

Figure 26:Showing Returned detials in a file



```
1. Rent Equipment
2. Return Equipment
3. Exit
Enter your choice (1/2/3): 1
Available Equipment:
1. Velvet Table Cloth (Saathi) - Price: $8.0 - Quantity: 11
2. Microphone Set (Audio Technica) - Price: $189.0 - Quantity: 26
3. Disco Light Set (Sonoff) - Price: $322.0 - Quantity: 21
4. 7.1 Surround Sound Speaker Set (Dolby) - Price: $489.0 - Quantity: 3
5. Dinner Table 8x5 (Panda Furnitures) - Price: $344.0 - Quantity: 8
```

Figure 27:Availabe equipments after update in stock

Conclusion

This coursework was totally based on our knowledge and understanding towards python programming language.Throughout this whole coursework I was able to learn many things which would missed I have'nt done this coursework.I have face many difficulties while doing this coursework .One of my weak point in this coursework was to define the function and use them in Python,which was fixed by taking the help from teachers,tutors and my friendsIt was very difficult to me to overcome from this .At first it really feels hard to do .But I realize that if we will not do task at a time we can get in problem at last Due to help of my tutor and friend I have overcome this problem and my understanding towards the python suddenly increased. I have also learn how to link those files in Python.As,I was developing the application ,I had severeal ideas for other things I might make with this programming language. And will do probably the same.Python is a user friendly program which offers me a best way for my future including a lot of potential.Due to this now I am totally aware about plagiarism which is a serious issues which I would never repeat that again.

Also, specially wanna thank to Islington College for giving us chance to design this application.More thankfull to my lecturers,tutors who were so friendly while teaching us and solving those problem that I have overcome it and creating such helpful ,friendly environment in college leads many student to ask those teachers without any hesitation, and assist us in a good way.

Bibliography

docs.python. (n.d.). Retrieved from docs.python: https://github.com/python/cpython/tree/3.11/Lib/idlelib/

Python:Geeksforkeeks. (n.d.). Retrieved from Geeksforgeeks: https://www.geeksforgeeks.org/

Appendix

Code of main.py file

```
#import necessary function and modules from other files

import os

from datetime import datetime

from read import read_equipment_file,display_available_equipment

from rent import rent_equipment,generate_invoice

from return_items import return_items,generate_return_invoice



def input_text():#Display welcome message

    welcome_message = """

================================================================

        Welcome! To Equipment Rental Shop

================================================================"""

    print(welcome_message)


def update_equipment_file(equipment_file,equipment_dict):


    with open(equipment_file, 'w') as file:
        for equipment, details in equipment_dict.items():

            file.write(f"{equipment}, {details['brand']}, {details['price']}, {details['quantity']}\n")



def main():

    equipment_file = "equipment.txt"#Define the equipment file and transaction directory

    transaction_dir = "transactions"



    if not os.path.exists(transaction_dir):#Check if the transaction directory exists,

        os.makedirs(transaction_dir)



    equipment_dict = read_equipment_file(equipment_file)#Read the equipment information from the equipment file



    if equipment_dict:
```

```
    input_text()#Display the elcome message
    while True:
        print("\n1. Rent Equipment")#Display the main menu
        print("2. Return Equipment")
        print("3. Exit")
        choice = input("Enter your choice (1/2/3): ")#Gets user choice

        if choice == '1':#Rent equipment
            display_available_equipment(equipment_dict)
            customer_name = input("\nEnter customer name (or 'q' to quit): ")
            if customer_name.lower() == 'q':
                break

            choices = []
            while True:

                equipment_choice = int(input("Enter the number of the equipment to rent (or '0' to finish): "))

                if equipment_choice == 0:

                    break



                if 1 <= equipment_choice <= len(equipment_dict):

                    quantity = int(input(f"Enter the quantity of {list(equipment_dict.keys())[equipment_choice - 1]} to rent: "))

                    choices.append((equipment_choice, quantity))

                else:

                    print("Invalid equipment choice. Please try again.")



            rented_items = rent_equipment(equipment_dict, choices)#Rented the selected items
```

```
            if rented_items:#Generate an invoice

                invoice_path = generate_invoice(customer_name, rented_items, transaction_dir)

                print(f"Invoice generated for {customer_name}.")

                print(f"Invoice saved at: {invoice_path}")

                update_equipment_file(equipment_file,equipment_dict)




        elif choice == '2':#Return equipment

            invoice_name = input("Enter the name of the invoice to return items from: ")

            invoice_path = os.path.join(transaction_dir, invoice_name)

            if os.path.exists(invoice_path):

                returned_items, customer_name,fine,fined_days = return_items(equipment_dict, invoice_path)

                if(len(returned_items)>0):
                    return_invoice_path = generate_return_invoice(customer_name, returned_items,fine,fined_days, transaction_dir)

                    print(f"Returned items invoice generated for {customer_name}.")

                    print(f"Invoice saved at: {return_invoice_path}")

                    update_equipment_file(equipment_file,equipment_dict)

            else:

                print(f"Invoice '{invoice_name}' not found.")



        elif choice == '3':#Exit the program
            print("Equipment information updated.Goodbye!")

        else:
            print("Invalid choice. Please enter a number from the given choices 1,2 or 3.")

            break



print(equipment_dict.items())  #For debugging process
```

```
if __name__ == "__main__":

    main()
```

Code of rent.py file

```
import os
import datetime #import datetime module  for timestamp generation


def rent_equipment(equipment_dict, choices):
    '''rent selected equipment  items and update their quantities  in the equipment dictionary.
    equipment_dict(dict):A dictionary containing equipment information.
    choices(list):A list of tuples,where each tuples contains the index of the the chosen equipmet and quantity to rent.
    '''

    rented_items = []#initialize an empty list to store rented items

    rent_date = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")#for getting current date and time

    for choice, quantity in choices:

        equipment_name = list(equipment_dict.keys())[choice - 1]

        details = equipment_dict[equipment_name]

        if details['quantity'] >= quantity:#checking if the enough items are avialbe to rent
```

```
            details['quantity'] -= quantity#Append the rented items to the list

            rented_items.append((equipment_name, details['brand'], details['price'], quantity, 
rent_date))

        else:

            print(f"Not enough {equipment_name} available.")

            return None

    return rented_items



def generate_invoice(customer_name, rented_items, transaction_dir):

    timestamp = datetime.datetime.now().strftime("%Y%m%d%H%M%S")

    invoice_name = f"{customer_name}_{timestamp}.txt"

    invoice_path = os.path.join(transaction_dir, invoice_name)



    total_cost = 0
```

```
with open(invoice_path, 'w') as file:
    file.write(f"Customer: {customer_name}\n")
    for equipment, brand, price, quantity,rent_date in rented_items:
        file.write(f"Equipment: {equipment} ({brand})\n")
        file.write(f"Quantity: {quantity}\n")
        file.write(f"Price per item: ${price}\n")
        file.write(f"Cost: ${price*quantity}\n")
        total_cost += price * quantity
        file.write('-' * 20 + '\n')
    file.write(f"Rent Date: {rent_date}\n")
    file.write(f"Total Cost: ${total_cost}\n")
    file.write(f"Rental Duration: 5 Days")
    return invoice_path
```

Code of return_items.py file

```
import os
import datetime
def return_items(equipment_dict, invoice_path):

    returned_items = []

    rented_items = []

    return_date_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    rent_days=0



    with open(invoice_path, 'r') as file:

        lines = file.readlines()

        customer_name = lines[0].split(": ")[1].strip()

        rent_date = lines[-3].split(": ")[1].strip()
        # Calculate the number of days rented

        rent_days = (datetime.datetime.now() - datetime.datetime.strptime(rent_date,"%Y-%m-%d %H:%M:%S")).days

        if rent_days > 5:

            fine = (rent_days - 5) * 10  # Assuming a $10 fine per day

        else:

            fine = 0

        fined_days=rent_days - 5

        # Extract rented items information

        rented_items_info = [lines[i:i+4] for i in range(1, len(lines)-3, 5)]



        print(f"\nRented Items for {customer_name}:")

        i=1

        for idx, item_info in enumerate(rented_items_info, start=1):

            if(int(item_info[1].split(": ")[1])>0):
            equipment_info = item_info[0].split(" (")

            equipment_name = equipment_info[0].split(": ")[1]

            brand = equipment_info[1][:-1]

            quantity = int(item_info[1].split(": ")[1])

            price = float(item_info[2].split(": $")[1])

            total_cost = float(item_info[3].split(": $")[1])

            rented_items.append([equipment_name, brand, price, quantity, total_cost,
rent_date,quantity])

            print(f"{i}. Equipment: {equipment_name} ({brand}, Quantity: {quantity}, Total
Cost: ${total_cost}, Rent Date: {rent_date}")

            i+=1




     # Prompt user to select items and quantity to return

      items_to_return = []
```

```
        if(len(rented_items)>0):

            proceed=True

            while proceed:

                choices = input("\nEnter the numbers of the items to return (comma-separated, or 'q' to quit): ").split(',')

                if 'q' in choices:

                    break




                for choice in choices:

                    try:

                        choice = int(choice)

                        if 1 <= choice <= len(rented_items):

                            quantity_to_return = int(input(f"Enter the quantity of item {rented_items[choice-1][0]} to return: "))

                            if 1 <= quantity_to_return <= rented_items[choice-1][3]:
                                items_to_return.append((rented_items[choice - 1], quantity_to_return))
                                rented_items[choice-1][6]=rented_items[choice-1][3]-quantity_to_return
                                proceed=False
                            else:
                                print(f"Invalid quantity. Please enter a value between 1 and {rented_items[choice-1][3]}.")
                                proceed=True
                        else:
                            print(f"Invalid choice: {choice}")
                            proceed=True
                    except ValueError:
                        print(f"Invalid input: {choice}")
                        break
        else:
            print(f"There aren't any items for {customer_name} to return")



        for (equipment, brand, price, quantity, total_cost, rent_date,remaining), quantity_to_return in items_to_return:

            # Update equipment quantities

            if equipment in equipment_dict:

                equipment_dict[equipment]['quantity'] += quantity_to_return

            else:

                equipment_dict[equipment] = {'brand': brand, 'price': price, 'quantity': quantity_to_return}



            returned_items.append((equipment, brand, price, quantity_to_return, price*quantity_to_return, return_date_time, rent_date))

        if(len(rented_items)>0):

            with open(invoice_path, 'w') as file:

                tst_total_cost=0
                file.write(f"Customer: {customer_name}\n")

                for equipment, brand, price, quantity, total_cost, rent_date,remaining in rented_items:

                    file.write(f"Equipment: {equipment} ({brand}\n")

                    file.write(f"Quantity: {remaining}\n")

                    file.write(f"Price per item: ${price}\n")

                    file.write(f"Cost: ${price*remaining}\n")

                    file.write('-' * 20 + '\n')

                    tst_total_cost+=price*remaining

                file.write(f"Rent Date: {rent_date}\n")

                file.write(f"Total Cost: ${tst_total_cost}\n")

                file.write(f"Rental Duration: 5 Days")

    return returned_items, customer_name,fine,fined_days
```

```
# Modify the main function accordingly to allow returning items


import datetime



def generate_return_invoice(customer_name, returned_items,fine,fined_days, transaction_dir):

    timestamp = datetime.datetime.now().strftime("%Y%m%d%H%M%S")

    invoice_name = f"{customer_name}_return_{timestamp}.txt"

    invoice_path = os.path.join(transaction_dir, invoice_name)



    total_refund = 0



    with open(invoice_path, 'w') as file:

        file.write(f"Customer: {customer_name}\n")

        for equipment, brand, price, quantity, total_cost, return_datetime, rent_date in returned_items:

            file.write(f"Equipment: {equipment} ({brand}\n")
    file.write(f"Quantity Returned: {quantity}\n")

    file.write(f"Price per item: ${price}\n")

    file.write(f"Total Cost: ${total_cost}\n")

    file.write(f"Rent Date: {rent_date}\n")

    file.write(f"Return Date and Time: {return_datetime}\n")

    refund_amount = total_cost

    total_refund += refund_amount

    file.write(f"Refund Amount: ${refund_amount}\n")

    file.write('-' * 20 + '\n')

file.write(f"Total Refund: ${total_refund}\n")

if fine > 0:

    file.write(f"Fine (for {fined_days} extra days): ${fine}\n")
    return invoice_path
```

Code of read.py file

```
import os


def read_equipment_file(file_name):

    equipment_dict = {}

    try:

        with open(file_name, 'r') as file:

            for line in file:

                name, brand, price, quantity = line.strip().split(', ')

                equipment_dict[name] = {'brand': brand, 'price': float(price), 'quantity': int(quantity)}

        return equipment_dict

    except FileNotFoundError:

        print(f"The file '{file_name}' does not exist.")

        return {}
```

```
def update_equipment_file(equipment_file,equipment_dict):

    with open(equipment_file, 'w') as file:

        for equipment, details in equipment_dict.items():

            file.write(f"{equipment}, {details['brand']}, {details['price']}, {details['quantity']}\n")



def display_available_equipment(equipment_dict):

    print("Available Equipment:")

    for idx, (equipment, details) in enumerate(equipment_dict.items(), start=1):

        print(f"{idx}. {equipment} ({details['brand']}) - Price: ${details['price']} - Quantity:
{details['quantity']}")
```

Thankyou