



CS4051NI

60% Individual Coursework

2023 Spring

Student Name: - Aasutosh Kumar Verma

London Met ID: -22085760

College ID: -NP01AI4S230020

Assignment Due Date: Monday, July 3, 2023

Assignment Submission Date: Friday, Aug 25, 2023

Word Count: 6166

Project File Links:

YouTube Link:	
Google Drive Link:	

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.

Contents

1. Introduction.....	1
1.1 Goals and Objectives:	1
2. Discussion and Analysis.....	3
2.1 Algorithm.....	3
2.2 Flowchart.....	5
2.3 Pseudocode	7
For Main_Files.py.....	7
For Write_Files.py.....	9
For Read_Files.py.....	9
For Print.py.....	10
For Invoice_notes.py.....	10
For Operation.py.....	12
2.4 Data Structures	16
Dictionary:.....	16
List:	16
TUPLES:	17
INTEGER:.....	17
STRING:	17
BOOLEAN:.....	17
3 Program.....	18
3.1 Overview of the program	18
3.2 Rules	18
3.2. Procedure of renting and returning costume:	19
4 Testing	28
4.1 Test 1	28
4.2 Test 2	29
4.3 Test 3	31
4.4 Test 4	33
4.5 Test 5	35
5 Conclusion.....	39
-Limitation: -	39
Solutions	40
6 References.....	40
7 Appendix	40

7.1 Operation_Files.py..... 40

7.2 Write_Files.py..... 43

7.3 Read_Files.py..... 43

7.4 Main_Files.py..... 44

7.5 Print.py..... 47

7.6 invoice_notes.py..... 50

List Of Figures

Figure: -1 Flowchart	5
Figure: -2 Flowchart	6
Figure: -3 Dictionary code	16
Figure: -4 Dictionary used	16
Figure: -5 List code	17
Figure: -6 List used	17
Figure: -7 Process of renting items or equipment's.	19
Figure: -8 Rent	
Figure: -9 Rent	
Figure: -10 Rent	
Figure: -11 Rent	
Figure: -12 Rent	
Figure: -13 Rent	
Figure: -14 Rent	
Figure: -15 Rent	23
Figure: -16 Process of returning items or equipment's.	23
Figure: -17 Return	
Figure: -18 Return	
Figure: -19 Return	
Figure: -20 Return	25
Figure: -21 Invoice rent. Txt & return.txt	26
Figure: -22 invoice rent	26
Figure: -23 invoice return	26
Figure: -24 ending	27
Figure: -25 Test -1 fig: - Try & except	28
Figure: -26 Test -1	
Figure: -27 Test-2 fig: - Negative\nnone exists value	29
Figure: -28 Test-3 fig: - Rent process	30
Figure: -29 Test-3	
Figure: -30 Test-3	
Figure: -31 Test-4 fig: -Return Process	31
Figure: -32 Test-4	
Figure: -33 Test-4	
Figure: -34 Test-5 fig: -Updating inventory after rented and after returning	35
Figure: -35 Test-5	
Figure: -36 Test-5	
Figure: -37 Test-5	
Figure: -38 Test-5	
Figure: -39 Test-5	
Figure: -40 Test-5	

List of Tables Table 1: test 1	28
Table 2: test 2	29
Table 3 : Test 3	31
Table 4: test 4	33
Table 5:test 5	35



1. Introduction

Python is an object-oriented programming language and high-level language. which uses dynamic semantics that is interpreted. Python is one of the most. popular programming languages and it is used for data science, software, and web. development.

This is the first time where we are required create an application using python. which can rent and return items. So, the requirement is to create an item. rental system. The basic function of the system is where users are able to rent. items and generate a bill and users are able to return the rented items and if they have rented the items for more than five days it fines the customer and displays the bill while executing the program and displays the bill in the file. I have divided the program into 6 files Main.py, Print.py, Write_Files.py, Read__Files.py, Operation.py, Invoice_notes.py. The executable file is Main.py file.

1.1 Goals and Objectives:

- ❖ To create a system where users can rent multiple items.
- ❖ To create a system where users can return multiple items.
- ❖ Generation of bill in both shell and txt file.
- ❖ The quantity should be subtracted while renting.
- ❖ The quantity should be added while returning.
- ❖ To provide a robust search functionality, enabling users to easily find specific items or browse categories.
- ❖ Implementation of secure payment gateways, ensuring safe and reliable transactions.

- ❖ Ability to track rental history, providing users with detailed insight into their past rentals and aiding in future recommendations.
- ❖ Creation of a user-friendly interface that is both visually appealing and functional, enhancing user interaction and satisfaction.
- ❖ Inclusion of a detailed help and support section, offering immediate assistance and guidance to users as needed.
- ❖ Integration with existing inventory and customer relationship management systems, ensuring a seamless workflow within the organization.
- ❖ Compliance with legal and industry standards, guaranteeing the system's integrity and adherence to applicable regulations and best practices.
- ❖ Optimization for various devices and platforms, allowing users to access the system from desktops, tablets, or mobile devices.
- ❖ Implementation of feedback and review mechanisms, fostering community engagement and continuous improvement of the system.
- ❖ Invalid message should be displayed while entering wrong data.
- ❖ Include the technologies used for the development of the project.

The main goal of this project is for students able to develop a system which is able.

rent and return a item which generates a bill which includes Total for rent and

a bill which includes Fine & Total for return. Both bills should be generated in text file. Main

goal is for students to be able to be more familiar with the language & to increase their creativity ideas based on real scenarios depends upon on their research.

2. Discussion and Analysis

- Explain how the program was developed with:

2.1 Algorithm

An algorithm is a set of instructions for solving a specific problem in programming. It takes set of input and output the desired results. It is visualization before writing a program.

STEP 1 – Display different options of the program.

STEP 2 – Take input form user and assign it to variable named options.

STEP 3 – If value of options is 1 go to step 4, if value of options is 2 go to step 18, if value of option is something else go to step 2.

STEP 4 – Take the content from the text file containing information about the items and store it in a data dictionary.

STEP 5 – Ask the user for the id of the equipment they want to rent and store it in variable customer name.

STEP 6 – Display items that are available to rent using the data dictionary.

STEP 7 – Ask user for num of items to rent one or multiple input required numbers. I

STEP 8 – If Num of items is invalid than go to step 7. Ask the user for the equipment code of the items.

STEP 9 – Ask users to enter valid quantities. If users had rented multiple items go to step 8.

STEP 10 – If invalid in equipment code or quantity than go to step 8.

STEP 11- subtracts the rented items from inventory stock.

STEP 12- update the data dictionary based on step 11.

STEP 13- Calculation of total amount based on step 7 to step 10 input.

STEP 14- stored the data used to create rented bills from step 5 to Step 12 in rental details.

STEP 15- Generate invoice inside shell & outside in .txt file format.

STEP 16- Generate Txt invoice contains Name & unique id number.

STEP 17- Display Updated inventory from step 12.

STEP 18- show the input options.

STEP 19-Ask users if they want to return or want to rent more with new invoice that contains unique id number,

STEP 20-If users choose option 2. Then ask users for it name.

STEP 21-Displayed updated inventory stock from rented details.

STEP 22- Ask users to enter num of items need to return.

STEP 23- If users input invalid than go to step 22.

STEP 24- Ask users to enter equipment code.

STEP 25-if invalid go to step24

STEP 26-Ask users to enter number quantity for the rented equipment.

STEP 27-if invalid go to step 24

STEP 28-check users name is matched with rentals details or not.

STEP 29-If not than go to step1.

STEP 30-Ask user to input date of return.

STEP 31-If invalid date format goes to step 30.

STEP 32- Calculate due_date & late_fee based on the condition define in program.

STEP 33-Calculate Total Amount if users return within 5 days of rented or before due date add fine zero but if failed than add late fee charged define In program.

STEP 34-Display currently inventory Stock.

STEP 35-Display invoice in shell along with company information.

STEP 36-Update inventory stock by adding return item in rented data dictionary item.txt file

STEP 37 Display updated inventory stock.

2.2 Flowchart

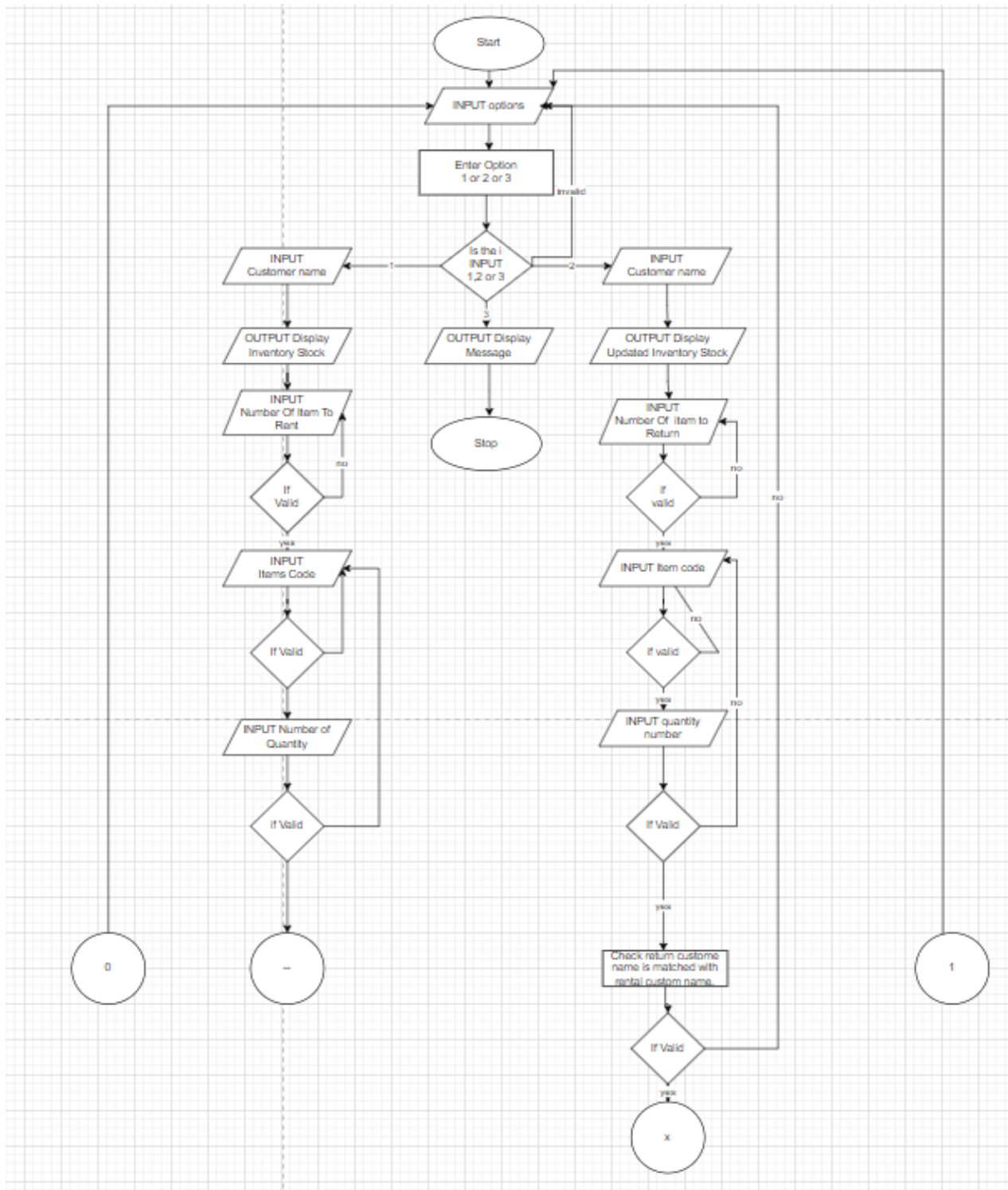


Figure:-1

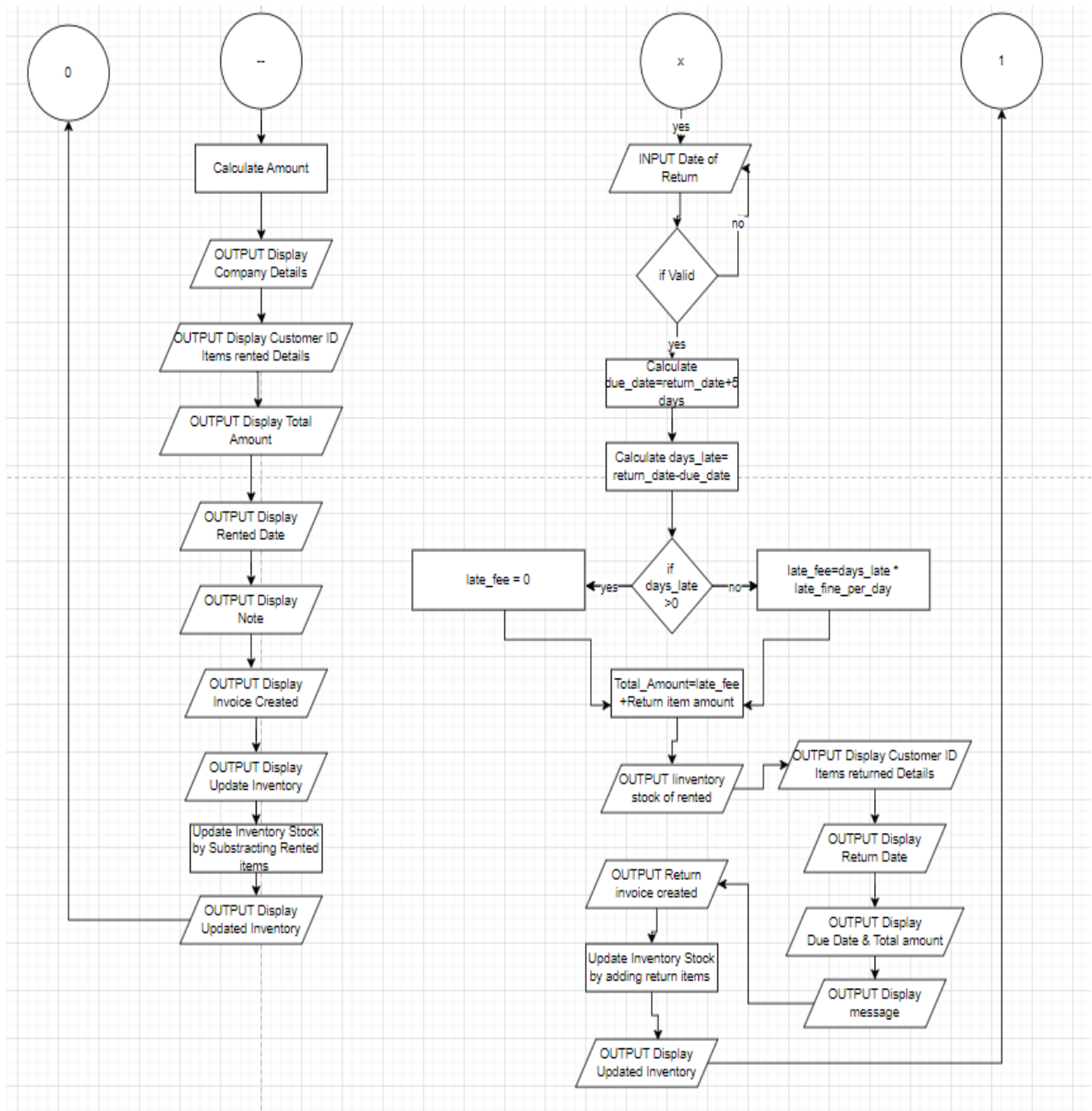


Figure:-2

2.3 Pseudocode

- Pseudocode is a simplified, human-readable representation of an algorithm or program's logic. It uses natural language and basic programming concepts to outline the steps and structure of the solution without using specific programming syntax.

For Main_Files.py

```
FUNCTION main():
    rented_items_dict = {} # Dictionary to store rented items for each customer
    returns = []
    rentals = []

    WHILE True:
        PRINT("\nMenu:")
        PRINT("1. Rent Equipment")
        PRINT("2. Return Equipment")
        PRINT("3. Exit")
        choice = INPUT("Choose an option (1/2/3): ")

        IF choice EQUALS '1':
            customer_name = INPUT("Enter Customer Name: ")

            PRINT("Available Items:")
            CALL print_inventory()

            WHILE True:
                num_items = INPUT("Number of Equipment Items to Rent: ")
                IF NOT IS_NUMERIC(num_items) OR TO_INT(num_items) <= 0:
                    PRINT("Please enter a valid positive integer.")
                    CONTINUE
                num_items = TO_INT(num_items)
                BREAK

            rentals = []
            FOR i FROM 1 TO num_items:
                LOOP = True
                WHILE LOOP EQUALS True:
                    TRY:
                        code = TO_INT(INPUT(f"Enter Equipment Code {i}: "))
                        quantity = TO_INT(INPUT(f"Enter Quantity {i}: "))
                        IF code <= 0 OR quantity <= 0:
                            PRINT("Please enter valid positive integers for code and quantity.")
                            CONTINUE
                    LOOP = False
```

```

        APPEND rentals, {'code': code, 'quantity': quantity}
        BREAK
    EXCEPT:
        PRINT("Please enter valid positive integers for code and quantity.")

# Store rented items in the dictionary
IF customer_name NOT IN rented_items_dict:
    rented_items_dict[customer_name] = []
EXTEND rented_items_dict[customer_name], rentals
# Update inventory table with new item quantities after adding them to rental list

result = CALL rent_equipment(customer_name, rentals, rented_items_dict)
PRINT(result)
CALL print_updated_inventory()

ELSE IF choice EQUALS '2':
    customer_name = INPUT("Enter Customer Name: ")
    CALL print_inventory() # Display inventory stock

WHILE True:
    num_items = INPUT("Number of Equipment Items to Return: ")
    IF NOT IS_NUMERIC(num_items) OR TO_INT(num_items) <= 0:
        PRINT("Please enter a valid positive integer.")
        CONTINUE
    BREAK

returns = []
FOR i FROM 1 TO num_items:
    LOOP = True
    WHILE LOOP EQUALS True:
        TRY:
            code = TO_INT(INPUT(f"Enter Equipment Code {i}: "))
            quantity = TO_INT(INPUT(f"Enter Quantity {i}: "))
            IF code <= 0 OR quantity <= 0:
                PRINT("Please enter valid positive integers for code and quantity.")
                CONTINUE
            LOOP = False
            APPEND returns, {'code': code, 'quantity': quantity}
            BREAK
        EXCEPT ValueError:
            PRINT("Only valid integers for code and quantity.")
# Check if entered customer name matches rented items' records
IF customer_name IN rented_items_dict:
    PRINT(f"Items for {customer_name}:")
    FOR item IN rented_items_dict[customer_name]:
        PRINT(f"Equipment Code: {item['code']}, Quantity: {item['quantity']}")

```

```

        result = CALL return_equipment(customer_name, returns, rented_items_dict)
        PRINT(result)
        CALL print_updated_inventory()
    ELSE:
        PRINT("Customer name does not match rented records. Please enter a valid customer
name.")

```

```

    ELSE IF choice EQUALS '3':
        BREAK
    ELSE:
        PRINT("Invalid choice, please select 1, 2, or 3.")

```

```

PRINT("Thanks for using our software!")

```

```

IF __name__ == "__main__":

```

```

    CALL main()

```

For Write_Files.py

```

FUNCTION write_inventory(inventory):

```

```

    WITH open('item.txt', 'w') AS file:

```

```

        FOR EACH item IN inventory:

```

```

            line = JOIN([TO_STRING(item[0]), item[1], item[2], '$' + TO_STRING(item[3]),
TO_STRING(item[4])), ',') + '\n'
            file.WRITE(line)

```

```

FUNCTION create_invoice(content, invoice_name):

```

```

    WITH open(invoice_name, 'w') AS file:

```

```

        file.WRITE(content)

```

For Read_Files.py

```

FUNCTION read_inventory():

```

```

    inventory = []

```

```

    WITH open('item.txt', 'r') AS file:

```

```

        FOR EACH line IN file:

```

```

            sn, name, description, price, quantity = SPLIT line BY ','

```

```

            price = TO_INTEGER(REMOVE_DOLLAR_SIGN(STRIP price))

```

```

            quantity = TO_INTEGER(STRIP quantity)

```

```

            inventory.APPEND([TO_INTEGER(sn), name, description, price, quantity])

```

RETURN inventory

For Print.py

```
IMPORT read_inventory FROM Read_Files
```

```
CONSTANT COMPANY_NAME = "Fashion Plaza"
```

```
CONSTANT ADDRESS = "LAHAN-1"
```

```
CONSTANT PHONE_NUMBER = "9824707814"
```

```
CONSTANT EMAIL = "npai4s230020@gmail.com"
```

```
CONSTANT MAX_RENT_ITEMS = 20
```

```
CONSTANT LATE_FINE_PER_DAY = 10
```

```
FUNCTION print_inventory():
```

```
    inventory = read_inventory()
```

```
    OUTPUT("\nInventory Stock:")
```

```
    OUTPUT("SN\tName\t\t\tDescription\t\tPrice\tQuantity")
```

```
    FOR EACH item IN inventory:
```

```
        OUTPUT(CONCATENATE item[0], "\t", item[1], "\t", item[2], "\t", item[3], "\t", item[4])
```

For Invoice_notes.py

```
IMPORT datetime
```

```
IMPORT timedelta
```

```
IMPORT COMPANY_NAME, ADDRESS, PHONE_NUMBER, EMAIL FROM Print
```

```
IMPORT create_invoice FROM Write_Files
```

```
FUNCTION generate_unique_invoice_id(customer_name):
```

```
    timestamp = FORMAT_DATE(TODAY, 'YYYYMMDDHHMMSS')
```

```
    RETURN CONCATENATE customer_name, "_", timestamp
```

```
FUNCTION generate_rent_invoice(customer_name, rented_item_details, total_amount):
```

```
invoice_content = CONCATENATE COMPANY_NAME, "\nAddress: ", ADDRESS, "\nPhone: ", PHONE_NUMBER, "\nEmail: ", EMAIL, "\nCustomer Name: ", customer_name, "\nDate of Rental: ", FORMAT_DATE(TODAY, 'YYYY-MM-DD'), "\n"
```

```
invoice_content += "Items Rented:\n"
```

```
FOR EACH item IN rented_item_details:
```

```
    invoice_content += CONCATENATE "- ", item[0], " - ", item[1], " - $", item[2], " - Quantity: ", item[3], "\n"
```

```
due_date = TODAY + timedelta(days=5)
```

```
invoice_content += CONCATENATE "Total Amount: $", total_amount, "\nRented Date: ", FORMAT_DATE(due_date, 'YYYY-MM-DD'), "\nNote: Please return the equipment within 5 days to avoid fines.\n"
```

```
unique_invoice_name = generate_unique_invoice_id(customer_name) + "_rent_invoice.txt"
```

```
create_invoice(invoice_content, unique_invoice_name)
```

```
RETURN CONCATENATE "Invoice created: ", unique_invoice_name
```

```
FUNCTION generate_return_invoice(customer_name, return_date, returned_item_details, due_date, late_fee, total_amount):
```

```
    invoice_content = CONCATENATE COMPANY_NAME, "\nAddress: ", ADDRESS, "\nPhone: ", PHONE_NUMBER, "\nEmail: ", EMAIL, "\nCustomer Name: ", customer_name, "\nDate of Return: ", return_date, "\n"
```

```
    invoice_content += "Items Returned:\n"
```

```
FOR EACH item IN returned_item_details:
```

```
    invoice_content += CONCATENATE "- ", item[0], " - ", item[1], " - $", item[2], " - Quantity: ", item[3], "\n"
```

```
invoice_content += CONCATENATE "Due Date: ", due_date, "\n"
```



```
IF late_fee > 0 THEN
```

```
    invoice_content += CONCATENATE "Late Fee: $", late_fee, "\n"
```

```
    invoice_content += CONCATENATE "Total Amount: $", total_amount + late_fee, "\nThank you  
for returning the equipment.\n"
```

```
RETURN invoice_content
```

For Operation.py

```
IMPORT datetime
```

```
IMPORT timedelta
```

```
IMPORT generate_unique_invoice_id, generate_rent_invoice, generate_return_invoice FROM  
Invoice_notes
```

```
IMPORT COMPANY_NAME, ADDRESS, PHONE_NUMBER, EMAIL, LATE_FINE_PER_DAY  
FROM Print
```

```
IMPORT write_inventory, create_invoice FROM Write_Files
```

```
IMPORT read_inventory FROM Read_Files
```

```
IMPORT print_inventory FROM Print
```

```
FUNCTION calculate_late_fee(return_date, due_date):
```

```
    days_late = (return_date - due_date).days
```

```
    IF days_late > 0 THEN
```

```
        RETURN days_late * LATE_FINE_PER_DAY
```

```
    ELSE
```

```
        RETURN 0
```

```
FUNCTION rent_equipment(customer_name, rentals, rented_item_details):
```

```
    inventory = read_inventory()
```

```
    total_amount = 0
```

```
    rented_item_details = []
```

```
invoice_content = CONCATENATE COMPANY_NAME, "\nAddress: ", ADDRESS, "\nPhone: ",  
PHONE_NUMBER, "\nEmail: ", EMAIL, "\nCustomer Name: ", customer_name, "\nDate of  
Rental: ", TODAY
```

```
FOR EACH rental IN rentals:
```

```
    item_found = False
```

```
    FOR EACH item IN inventory:
```

```
        IF rental['code'] == item[0] THEN
```

```
            item_found = True
```

```
            IF rental['quantity'] > item[4] THEN
```

```
                PRINT "Only", item[4], item[1], "available. Cannot rent", rental['quantity'], "as  
requested."
```

```
                CONTINUE
```

```
            END IF
```

```
            item[4] -= rental['quantity']
```

```
            total_amount += item[3] * rental['quantity']
```

```
            APPEND (item[1], item[2], item[3], rental['quantity']) TO rented_item_details
```

```
        END FOR
```

```
    IF NOT item_found THEN
```

```
        PRINT "Item with code", rental['code'], "not found in inventory."
```

```
    END IF
```

```
write_inventory(inventory)
```

```
IF SIZE OF rented_item_details > 0 THEN
```

```
    result = generate_rent_invoice(customer_name, rented_item_details, total_amount)
```

```
    RETURN result
```

```
ELSE
```

```
    RETURN "No items rented. Invoice not created."
```

```
END IF
```

```
FUNCTION return_equipment(customer_name, returns, rented_items_dict):
```

```
    valid_return = True
```

```
    inventory = read_inventory()
```

```
    loop = True
```

```
    WHILE loop == True:
```

```
        TRY:
```

```
            return_date_str = INPUT("Enter the Return Date (YYYY-MM-DD): ")
```

```
            return_date = PARSE_TO_DATE(return_date_str, 'YYYY-MM-DD')
```

```
            EXIT LOOP
```

```
        EXCEPT ValueError:
```

```
            PRINT "Please enter a valid date in the format YYYY-MM-DD."
```

```
    END WHILE
```

```
    print_inventory()
```

```
    invoice_content = CONCATENATE COMPANY_NAME, "\nAddress: ", ADDRESS, "\nPhone: ",  
    PHONE_NUMBER, "\nEmail: ", EMAIL, "\nCustomer Name: ", customer_name, "\nDate of  
    Return: ", return_date
```

```
    total_amount = 0
```

```
    rented_items = rented_items_dict.get(customer_name, [])
```

```
    returned_item_details = []
```

```
    FOR EACH returned_item IN returns:
```

```
        item_found = False
```

```
        FOR EACH item IN inventory:
```

```
            IF returned_item['code'] == item[0] THEN
```

```
                item_found = True
```

```
                IF returned_item['quantity'] > 0 THEN
```

```
                    IF returned_item['quantity'] > item[4] THEN
```

```
                        PRINT "Error: Cannot return more", item[1], "than were rented."
```

```

        valid_return = False
    ELSE
        APPEND (item[1], item[2], item[3], returned_item['quantity']) TO
returned_item_details
        item[4] += returned_item['quantity']
        total_amount += item[3] * returned_item['quantity']
    END IF
END IF
END FOR
IF NOT item_found THEN
    PRINT "Item with code", returned_item['code'], "not found in inventory."
END IF
END FOR

IF NOT valid_return THEN
    PRINT "Invalid return. Please make sure you are returning items you have rented."
    RETURN
END IF

write_inventory(inventory)

IF SIZE OF returned_item_details > 0 THEN
    due_date = return_date + timedelta(days=5)
    late_fee = calculate_late_fee(return_date, due_date)
    invoice_content = generate_return_invoice(customer_name, return_date,
returned_item_details, due_date, late_fee, total_amount)
    unique_invoice_name = generate_unique_invoice_id(customer_name) +
"_return_invoice.txt"
    create_invoice(invoice_content, unique_invoice_name)
    RETURN "Return Invoice created:", unique_invoice_name
ELSE

```

```
    RETURN "No items returned. Invoice not created."  
END IF
```

2.4 Data Structures

Efficient data manipulation, management, and organization lie at the core of programming. Data structures empower us to systematically arrange our data, allowing us to store collections of information and utilize them as needed. This program adeptly employs several essential built-in data structures provided by Python, including Integer, String, Boolean, List, and Dictionaries.

In this specific program, a data dictionary proves its value by efficiently storing rental details sourced from a text file. Furthermore, a 2D list serves as a versatile tool, serving various purposes such as tracking rented or returned costume ID, items details.

These data structures play a pivotal role in optimizing program performance. They provide a structured approach to data handling, enhancing the efficiency and effectiveness of coding and problem-solving endeavors. The following examples demonstrate the successful implementation of these data structures within the program.

Dictionary: In this data structure the data is stored in pairs of keys and values. Each value in a dictionary is assigned a key for more efficient management of data. This type of data structure is dynamic and mutable which means the values in the collection can be edited. The images below show examples of this data structure being implemented in the program

```
def main():  
    rented_items_dict = {} # Dictionary to store rented items for each customer  
  
def main():  
    rented_items_dict = {}  
    returns = []  
    rentals = []
```

Figure: -3 & Figure: -4

List: This type of data structure is used to store various data in a consecutive form. Each item in a list is assigned an address, which is called an index. The value of index begins from zero.

This type of data structure is mutable because it has various methods that lets us edit the elements of a list.

Lists are represented by big brackets []. The images below show examples of this data structure being implemented in the program.

```
returns = []
rentals = []

# Store rented items in the dictionary
if customer_name not in rented_items_dict:
    rented_items_dict[customer_name] = []
rented_items_dict[customer_name].extend(rentals)
```

Figure: -5 & Figure:-6

TUPLES: This type of data structure shares many similarities to the above mentioned, lists. Like a list each element is assigned with an index which starts from 0. The main factor that separates tuples from lists is that tuples are not mutable. The values in a Tuples cannot be edited. Tuples are represented by ().

INTEGER: This type of data structure can contain positive or negative whole numbers. This means it cannot contain fractions or decimals. In python an integer can be as long as you want.

STRING: This data structure is used to represent Unicode characters. It is most used to store alphabetic characters, but it can store a variety of characters.

BOOLEAN: This type of data structure is used to store one of the two built in values which are True and False. While assigning Boolean values the 'T' and 'F' of Truth and False respectively always must be capitalized, otherwise python will throw an error.

3 Program

3.1 Overview of the program

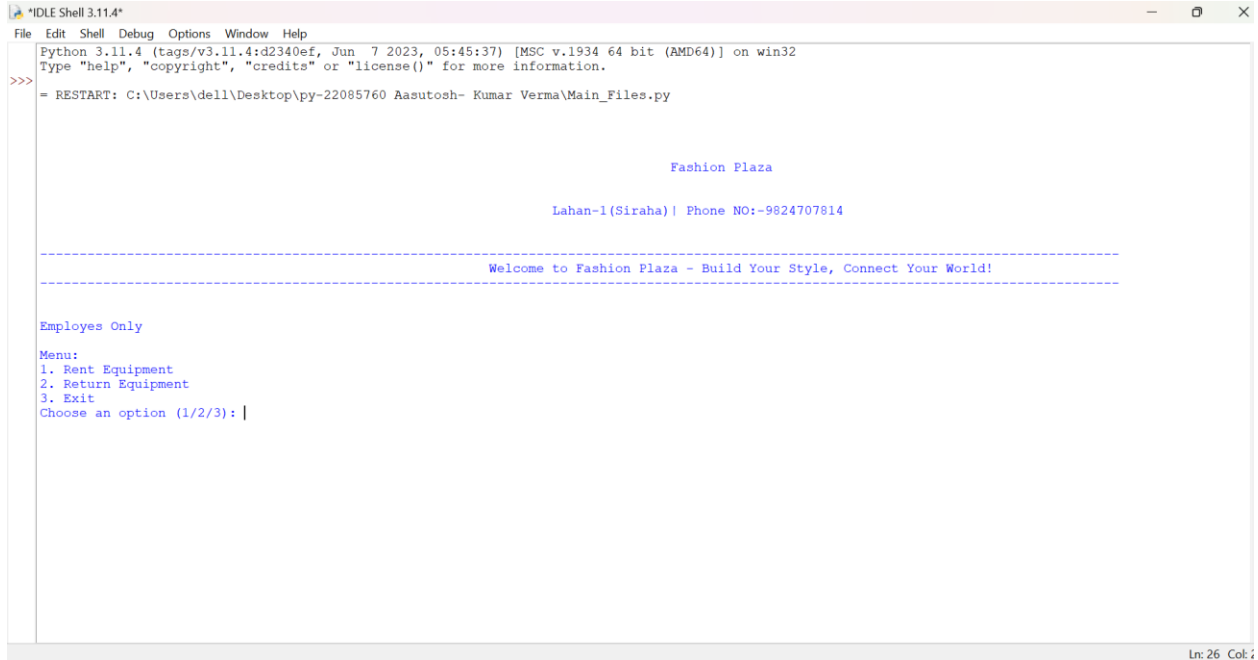
- This program was made for to handle the transaction of Item rental Shop. Along with updating inventory stocks that contains items details. It was developed based on real scenarios used in different shop or companies. The logic here used for basic operation which can used to run a small shop contains different items stocks. Firstly, the program gets information about items details from inventory located inside .txt files & stores it in a data dictionary for further logical operation. The program ask user to choose options whether they like to rent or return if they had rented before along their names or Id. The options 1 is Related to Rent Function which holds required fields for user input to generate bills or invoice based on the rented items which are available in stocks. After successfully rented multiple items than the users are required to return items within 5 days of rented date or else, they will be fine with \$10 per day. If their item's rented date is overdue date. During the process users are required to fill all the fields for returning process. The data is used for returning is check & compare with rented details to identify whether it's a valid customer or not along with items details to prevent getting caught in case of fraud or scam. After entering all require fields based on valid information a return invoice will be generated which includes name & unique Id Number In txt file format. The program operates under the following business.

3.2 Rules

- A customer is allowed to rent more than one type of items based on their selection, but the program is based on logic. So, there is max limit of 30 items for users to rent which is enough while maintain business operations smooth.
- A customers can only return if they had rented before, or else program will not process until a valid id is enter which match rented records along with valid items details.
- An invoice is generated after each transaction in txt file & in terminal to make sure the rent or return process is working.
- if an item is not returned within 10 days or Due date then a fine of \$10 for each day will be charged on wards until return.

3.3. Procedure of renting and returning costume:

3.3.1 Renting: To rent a items or equipment the user must first enter the Id of the costumer than inventory stock will show. After that users are required to choose items code and quantity based on stock. If users had selected multiples number of items than users need to enter multiple equipment code along with quantity based on users' decision.



```

Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): |
  
```

Figure:-7



```

Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: |
  
```

Figure:-8


```

IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:

-----
Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth    Saathi          8      14
2       Microphone Set        Audio Technica  189    13
3       Disco Light Set       Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5      Panda Furnitures 344    17
-----

Number of Equipment Items to Rent: k
Please enter a valid integer.
Number of Equipment Items to Rent: |

```

Ln: 41 Col: 3

Figure:-9

```

IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Equipment Code 1: 1
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1:

```

Ln: 29 Col: 2

Figure:-10

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:

-----
Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      14
2      Microphone Set      Audio Technica      189     13
3      Disco Light Set      Sonoff      322     8
4      7.1 Surround Sound Speaker Set      Dolby      489     20
5      Dinner Table 8x5      Panda Furnitures      344     17
-----

Number of Equipment Items to Rent: k
Please enter a valid integer.
Number of Equipment Items to Rent: 1
Enter Equipment Code 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: 1

Invoice created: Aas_20230824225006_rent_invoice.txt

-----
Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189     13
3      Disco Light Set      Sonoff      322     8
4      7.1 Surround Sound Speaker Set      Dolby      489     20
5      Dinner Table 8x5      Panda Furnitures      344     17
-----
Ln: 79 Col:

```

Figure:-11

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help

Number of Equipment Items to Rent: k
Please enter a valid integer.
Number of Equipment Items to Rent: 1
Enter Equipment Code 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: 1

Invoice created: Aas_20230824225006_rent_invoice.txt

-----
Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189     13
3      Disco Light Set      Sonoff      322     8
4      7.1 Surround Sound Speaker Set      Dolby      489     20
5      Dinner Table 8x5      Panda Furnitures      344     17
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 5
Invalid choice, please select 1, 2, or 3.
Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): |
Ln: 79 Col:

```

Figure:-12

```

IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17
-----
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 5
Invalid choice, please select 1, 2, or 3.
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: dsdd

Inventory Stock:
SN Name Description Price Quantity
1 Velvet Table Cloth Saathi 8 13
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17
Number of Equipment Items to Return: 1
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Customer name does not match rented records. Please enter a valid customer name.
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): |

```

Ln: 99 Cc

Figure:-13

```

IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
Enter Quantity 1: 1
Customer name does not match rented records. Please enter a valid customer name.
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: Aas

Inventory Stock:
SN Name Description Price Quantity
1 Velvet Table Cloth Saathi 8 13
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17
Number of Equipment Items to Return: d
Please enter a valid integer.
Number of Equipment Items to Return: 1
Enter Equipment Code 1: s
Only valid integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: d
Only valid integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Items for Aas:
Equipment Code: 1, Quantity: 1
Enter the Return Date (YYYY-MM-DD): 2023-05-09
Please enter a valid date in the format YYYY-MM-DD.
Enter the Return Date (YYYY-MM-DD): 2023-08-30

Inventory Stock:
SN Name Description Price Quantity
1 Velvet Table Cloth Saathi 8 13
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17

```

Ln: 166 Cc

Figure:-14

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
-----
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189    13
3      Disco Light Set      Sonoff      322    8
4      7.1 Surround Sound Speaker Set      Dolby      489    20
5      Dinner Table 8x5      Panda Furnitures      344    17

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Return: 2023-08-30
Items Returned:
- Velvet Table Cloth      - Saathi      - $8 - Quantity: 1
Due Date: 2023-09-04
Total Amount: $8
Thank you for returning the equipment.

Return Invoice created: Aas_20230824225210_return_invoice.txt
-----
Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      14
2      Microphone Set      Audio Technica      189    13
3      Disco Light Set      Sonoff      322    8
4      7.1 Surround Sound Speaker Set      Dolby      489    20
5      Dinner Table 8x5      Panda Furnitures      344    17

```

Figure:-15

3.3.2 Returning: To return an item or equipment the user must first enter the name of the customer. If it records is in rented or matches that they can process to return. Users can't return more than rented.

When returning user need to return within 5 days of rented than 0 fine will be added if the return is overdue date than per day 10 will be charged and will be added in total amounts along with items details.

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
-----
Number of Equipment Items to Rent: k
Please enter a valid integer.
Number of Equipment Items to Rent: 1
Enter Equipment Code 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: f
Please enter valid positive integers for code and quantity.
Enter Equipment Code 1: 1
Enter Quantity 1: 1

Invoice created: Aas_20230824225006_rent_invoice.txt
-----
Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189    13
3      Disco Light Set      Sonoff      322    8
4      7.1 Surround Sound Speaker Set      Dolby      489    20
5      Dinner Table 8x5      Panda Furnitures      344    17

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 5
Invalid choice, please select 1, 2, or 3.
Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): |

```

Figure:-16

```

IDE Shell 3.114*
File Edit Shell Debug Options Window Help
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17
-----
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 5
Invalid choice, please select 1, 2, or 3.
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: dsdd

Inventory Stock:
SN Name Description Price Quantity
1 Velvet Table Cloth Saathi 8 13
2 Microphone Set Audio Technica 189 13
3 Disco Light Set Sonoff 322 8
4 7.1 Surround Sound Speaker Set Dolby 489 20
5 Dinner Table 8x5 Panda Furnitures 344 17
Number of Equipment Items to Return: 1
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Customer name does not match rented records. Please enter a valid customer name.
Employees Only
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): |
Ln: 99

```

Figure:-17

```

C:\Users\user> python3 3.11.4\
File Edit Shell Debug Options Window Help
Enter Quantity l: 1
Customer name does not match rented records. Please enter a valid customer name.
Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: Aas

Inventory Stock:
SN      Name                        Description      Price  Quantity
1       Velvet Table Cloth             Saathi          8      13
2       Microphone Set                 Audio Technica  189    13
3       Disco Light Set               Sonoff          322    8
4       7.1 Surround Sound Speaker Set Dolby          489    20
5       Dinner Table 8x5              Panda Furnitures 344    17

Number of Equipment Items to Return: d
Please enter a valid integer.
Number of Equipment Items to Return: 1
Enter Equipment Code l: s
Only valid integers for code and quantity.
Enter Equipment Code l: 1
Enter Quantity l: d
Only valid integers for code and quantity.
Enter Equipment Code l: 1
Enter Quantity l: 1
Items for Aas:
Equipment Code: l, Quantity: 1
Enter the Return Date (YYYY-MM-DD): 2023-05-09
Please enter a valid date in the format YYYY-MM-DD.
Enter the Return Date (YYYY-MM-DD): 2023-08-30

Inventory Stock:
SN      Name                        Description      Price  Quantity
1       Velvet Table Cloth             Saathi          8      13
2       Microphone Set                 Audio Technica  189    13
3       Disco Light Set               Sonoff          322    8
4       7.1 Surround Sound Speaker Set Dolby          489    20
5       Dinner Table 8x5              Panda Furnitures 344    17

```

Figure:-18

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
Enter the Return Date (YYYY-MM-DD): 2023-08-09
Please enter a valid date in the format YYYY-MM-DD.
Enter the Return Date (YYYY-MM-DD): 2023-08-30

Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth   Saathi          8      13
2       Microphone Set       Audio Technica  189    13
3       Disco Light Set      Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5     Panda Furnitures 344    17

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Return: 2023-08-30
Items Returned:
- Velvet Table Cloth      - Saathi          - $8 - Quantity: 1
Due Date: 2023-09-04
Total Amount: $8
Thank you for returning the equipment.

Return Invoice created: Aas_20230824225210_return_invoice.txt
-----
Updated Inventory Stock:

Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth   Saathi          8      14
2       Microphone Set       Audio Technica  189    13
3       Disco Light Set      Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5     Panda Furnitures 344    17
-----

Employees Only
..
Ln: 166 Col

```

Figure:-19

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
-----
Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth   Saathi          8      13
2       Microphone Set       Audio Technica  189    13
3       Disco Light Set      Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5     Panda Furnitures 344    17

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Return: 2023-08-30
Items Returned:
- Velvet Table Cloth      - Saathi          - $8 - Quantity: 1
Due Date: 2023-09-04
Total Amount: $8
Thank you for returning the equipment.

Return Invoice created: Aas_20230824225210_return_invoice.txt
-----
Updated Inventory Stock:

Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth   Saathi          8      14
2       Microphone Set       Audio Technica  189    13
3       Disco Light Set      Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5     Panda Furnitures 344    17
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3):
Ln: 166 Col

```

Figure:-20

3.3.3 Text File generated for both rent & return system.

When a invoice is created a text file will be generated in respective files.


 Aas_20230824225006_rent_invoice	8/24/2023 10:50 PM	Text Document	1 KB
 Aas_20230824225210_return_invoice	8/24/2023 10:52 PM	Text Document	1 KB

Figure:-21

3.3.4 Invoice Display inside txt file format.

The details are included of rental process with rented items & total amount.

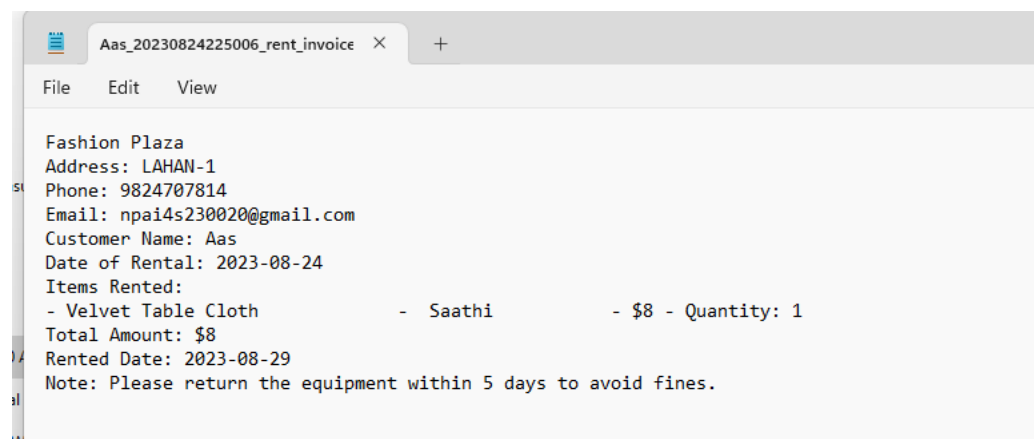


Figure:-22

The details is included of return process with return items & total amount if users failed to return before due date than fine will be charged but customers had return before due date so fine is zero added in total sum.

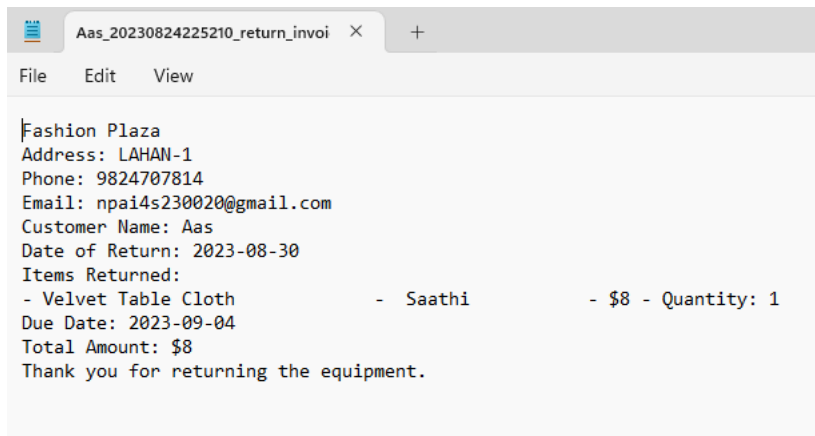


Figure:-23

3.3.5 Ending the program: After the user is done renting or returning equipment, they can exit the program by choosing the exit option in the main menu.

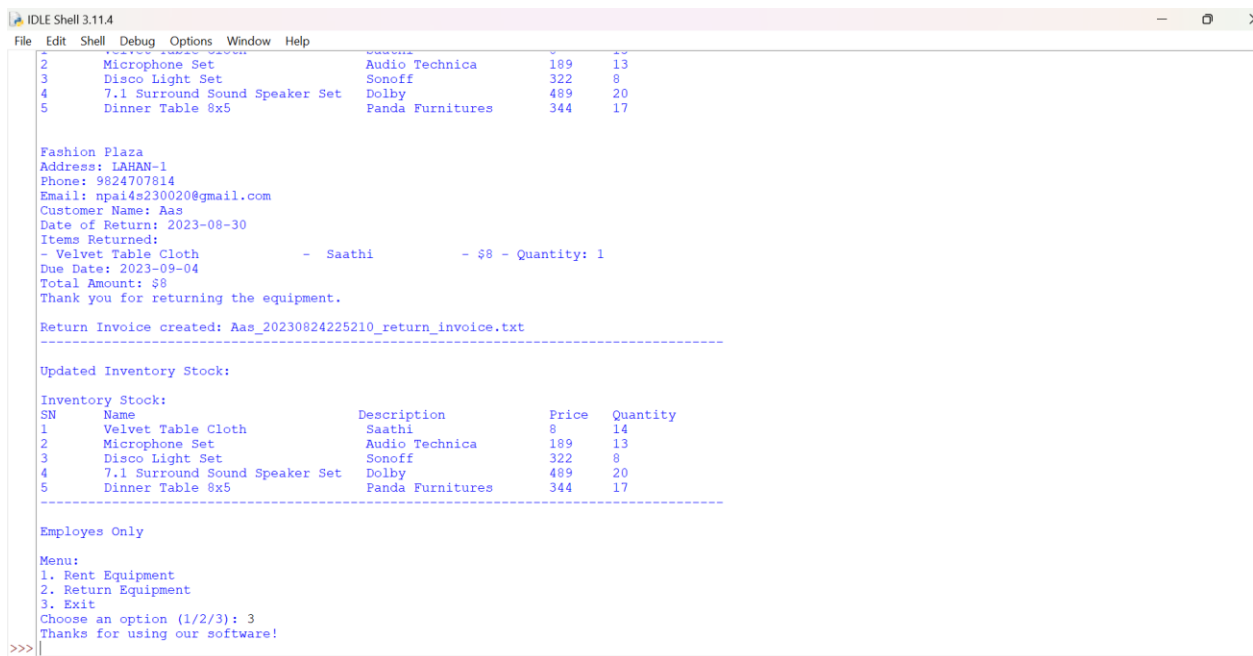


Figure:-24

4 Testing

4.1 Test 1

- Show implementation of try, except

Test no.	1
Objective	Testing try & except exception handling.
Action	Passing invalid input instead of string in equipment code
Expected Results	Display invalid message
Actual result	Invalid Message Displayed.
Conclusion	Test was successful.

```

Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:
-----
Inventory Stock:
SN      Name              Description      Price  Quantity
1      Velvet Table Cloth   Saathi          8       14
2      Microphone Set       Audio Technica   189     13
3      Disco Light Set      Sonoff           322     8
4      7.1 Surround Sound  Speaker Set     Dolby          489     20
5      Dinner Table 8x5     Panda Furnitures 344     17
-----
Number of Equipment Items to Rent: k
Please enter a valid integer.
Number of Equipment Items to Rent: |

```

Figure:-25

```

while loop == True:
    try:
        code = int(input(f"Enter Equipment Code {i+1}: "))
        quantity = int(input(f"Enter Quantity {i+1}: "))
        loop=False
        if code <= 0 or quantity <= 0:
            print("Please enter valid positive integers for code and
                continue
        returns.append({'code': code, 'quantity': quantity})
        break
    except ValueError:
        print("Only valid integers for code and quantity.")

```

Figure:-26

4.2 Test 2

- Selection rent and return of items

Test no.	2
Objective	Testing negative value /non-existed value
Action	Passing negative & non-exists value one in rent & one in return
Expected Results	Display invalid message in both rent & return as per user input.
Actual result	Invalid Message Displayed.
Conclusion	Test was successful.

```

Employes Only
|
Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:
-----

Inventory Stock:
SN      Name      Description      Price      Quantity
1      Velvet Table Cloth      Saathi      8      14
2      Microphone Set      Audio Technica      189      13
3      Disco Light Set      Sonoff      322      8
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17
-----

Number of Equipment Items to Rent: 3
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Enter Equipment Code 2: 2
Enter Quantity 2: 2
Enter Equipment Code 3: 3
Enter Quantity 3: -3
Please enter valid positive integers for code and quantity.
```

Figure:-27

4.3 Test 3

- File generation of renting of items(s) (Renting multiple items(s))

Test no.	3
Objective	Testing renting process & invoice in txt & Shell.
Action	Enter all valid details require in rentals system for creating rented invoices inside shell & outside in txt files format with name and unique id.
Expected Results	All details need to be valid for creating rentals system without leaving any field invoice will generate,
Actual result	All details were valid, rentals system created invoice generated,
Conclusion	Test was successful.

```

Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:\Users\dell\Desktop\py-22085760 Aasutosh- Kumar Verma\Main_Files.py

                                     Fashion Plaza

                               Lahan-1(Siraha) | Phone NO:-9824707814

-----
                               Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:
-----
Inventory Stock:
SN      Name                Description            Price  Quantity
1       Velvet Table Cloth      Saathi                8      14
2       Microphone Set          Audio Technica        189    13
3       Disco Light Set        Sonoff                322    8
4       7.1 Surround Sound Speaker Set  Dolby                489    20
5       Dinner Table 8x5        Panda Furnitures      344    17
-----

```

Figure:-28

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:
-----
Inventory Stock:
SN      Name                        Description      Price  Quantity
1       Velvet Table Cloth          Saathi          8      13
2       Microphone Set              Audio Technica  189    13
3       Disco Light Set             Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5             Panda Furnitures 344    17
-----
Number of Equipment Items to Rent: 1
Enter Equipment Code 1: 1
Enter Quantity 1: 1

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Rental: 2023-08-25

Invoice created: Aas_20230825092424_rent_invoice.txt
-----

Updated Inventory Stock:
Inventory Stock:
SN      Name                        Description      Price  Quantity
1       Velvet Table Cloth          Saathi          8      12
2       Microphone Set              Audio Technica  189    13
3       Disco Light Set             Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5             Panda Furnitures 344    17
-----

```

Figure:-29

```

Aas_20230824225006_rent_invoice X +
File Edit View

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Rental: 2023-08-24
Items Rented:
- Velvet Table Cloth          - Saathi          - $8 - Quantity: 1
Total Amount: $8
Rented Date: 2023-08-29
Note: Please return the equipment within 5 days to avoid fines.

```

Figure:-30

4.4 Test 4

- File generation of return process of items(s)

Test no.	4
Objective	Testing returning process & invoice in txt & Shell.
Action	Enter all valid details require in returns system for creating return invoices inside shell & outside in txt files format with name and unique id also input valid date of return based on condition due date & days late.
Expected Results	All details need to be valid for creating returns system without leaving any field invoice will generate, Total amounts will add late fine if users' field to return item before 5 days of being rented
Actual result	Invalid Message Displayed.
Conclusion	Test was successful.

```

File Edit Shell Debug Options Window Help

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: shyam

Inventory Stock:
SN      Name      Description      Price      Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189      11
3      Disco Light Set      Sonoff      322      5
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17
Number of Equipment Items to Return: 3
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Enter Equipment Code 2: 2
Enter Quantity 2: 2
Enter Equipment Code 3: 3
Enter Quantity 3: 3
Items for shyam:
Equipment Code: 1, Quantity: 2
Equipment Code: 1, Quantity: 2
Equipment Code: 2, Quantity: 2
Equipment Code: 3, Quantity: 3
Enter the Return Date (YYYY-MM-DD): 2023-08-30

Inventory Stock:
SN      Name      Description      Price      Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189      11
3      Disco Light Set      Sonoff      322      5
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17

```

Figure:-31

```

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: shyam
Date of Return: 2023-08-30
Items Returned:
- Velvet Table Cloth      - Saathi      - $8 - Quantity: 1
- Microphone Set      - Audio Technica      - $189 - Quantity: 2
- Disco Light Set      - Sonoff      - $322 - Quantity: 3
Due Date: 2023-09-04
Total Amount: $1352
Thank you for returning the equipment.

Return Invoice created: shyam_20230825102422_return_invoice.txt
-----

Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price      Quantity
1      Velvet Table Cloth      Saathi      8      14
2      Microphone Set      Audio Technica      189      13
3      Disco Light Set      Sonoff      322      8
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17

```

Figure:-32

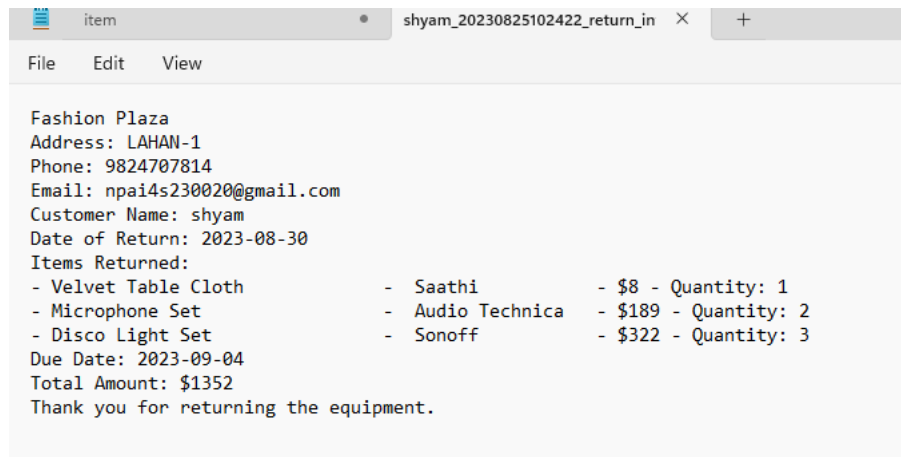


Figure:-33

4.5 Test 5

- Show the update in stock of items(s)

Test no.	5
Objective	Testing inventory update after renting & returning in txt & in shell
Action	Passing invalid input instead of string in equipment code
Expected Results	Display invalid message
Actual result	Invalid Message Displayed.
Conclusion	Test was successful.


```

Fashion Plaza

Lahan-1(Siraha) | Phone NO:-9824707814

-----
Welcome to Fashion Plaza - Build Your Style, Connect Your World!
-----

Employees Only|

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 1
Enter Customer Name: Aas
Available Items:
-----

Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth      Saathi          8      14
2       Microphone Set          Audio Technica  189    13
3       Disco Light Set         Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5        Panda Furnitures 344    17
-----

Number of Equipment Items to Rent: 1
Enter Equipment Code 1: 1
Enter Quantity 1: 1

```

Figure:-34

```

*IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Rental: 2023-08-25

Invoice created: Aas_20230825092859_rent_invoice.txt
-----

Updated Inventory Stock:

Inventory Stock:
SN      Name                Description      Price  Quantity
1       Velvet Table Cloth      Saathi          8      13
2       Microphone Set          Audio Technica  189    13
3       Disco Light Set         Sonoff          322    8
4       7.1 Surround Sound Speaker Set  Dolby          489    20
5       Dinner Table 8x5        Panda Furnitures 344    17
-----

```

Figure:-35

```

IDLE Shell 3.11.4*
File Edit Shell Debug Options Window Help

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3): 2
Enter Customer Name: Aas

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189      13
3      Disco Light Set      Sonoff      322      8
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17
Number of Equipment Items to Return: 1
Enter Equipment Code 1: 1
Enter Quantity 1: 1
Items for Aas:
Equipment Code: 1, Quantity: 1
Enter the Return Date (YYYY-MM-DD): 2023-08-30

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      13
2      Microphone Set      Audio Technica      189      13
3      Disco Light Set      Sonoff      322      8
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17

```

Figure:-36

```

Fashion Plaza
Address: LAHAN-1
Phone: 9824707814
Email: npai4s230020@gmail.com
Customer Name: Aas
Date of Return: 2023-08-30
Items Returned:
- Velvet Table Cloth      - Saathi      - $8 - Quantity: 1
Due Date: 2023-09-04
Total Amount: $8
Thank you for returning the equipment.

Return Invoice created: Aas_20230825092918_return_invoice.txt
-----

Updated Inventory Stock:

Inventory Stock:
SN      Name      Description      Price  Quantity
1      Velvet Table Cloth      Saathi      8      14
2      Microphone Set      Audio Technica      189      13
3      Disco Light Set      Sonoff      322      8
4      7.1 Surround Sound Speaker Set      Dolby      489      20
5      Dinner Table 8x5      Panda Furnitures      344      17
-----

Employees Only

Menu:
1. Rent Equipment
2. Return Equipment
3. Exit
Choose an option (1/2/3):

```

Figure:-37

In Txt File: -

Stock

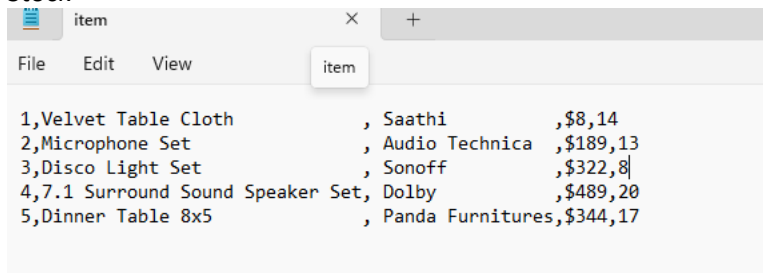


Figure:-38

After Rented

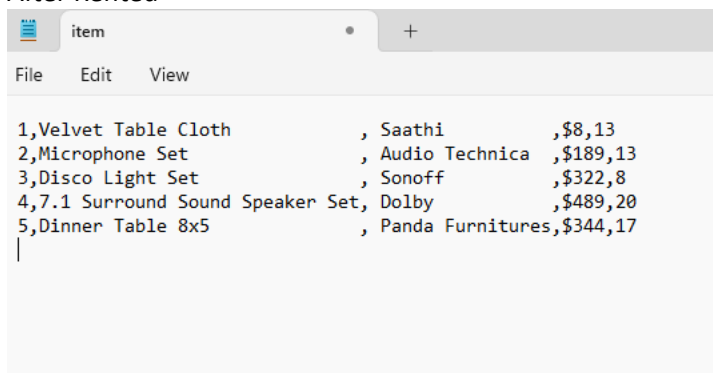


Figure:-39

After Returned

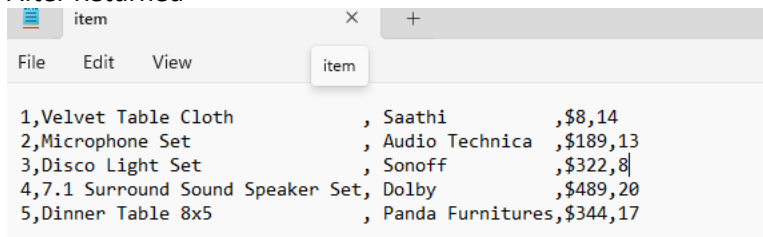


Figure:-40

5 Conclusion

The program main Objectives is to help student generates new ideas & logical concepts in higher level so, later the concept can be implemented in near future granting success in their journey. This course work was very interesting and challenging to do. I did lots of research based on my interested fields I took my time for learning about the working process & importance of pythons which can be uses in machine learning. Thanks to this course work I now feel comfortable with programming in python. & got many research and technology ideas or human lifestyle changing concepts I was able to understand the different concepts that were needed to do this course work.

I think that I now have a proper understanding of the various aspects of programming in python. & My imagination also got increased full of new concepts.

Now, I can assure that beside coding logic is what makes humans minds strong and experienced & promotes creativity that led young star like me towards success at first, I had a bit of trouble learning the concept of file handling but with a little bit of practice and guidance from my teachers, I am now confident in the topic. This course work was fun to do because I really enjoy problem solving. It was very satisfying to see a program that I made run smoothly and found the difference between python & java why this too holds different roles and making projects based on AI by combining both java & python languages the output results will be great.

-Limitation: -

- The late fee charged is calculated in days. If users failed to return items within 1 months than the program will not give accurate results.
- The invoice created while renting or returning process holds customers details which gets printed in invoice after completing system process but the bills or customer details which is store in .Txt files format is not being encrypted which limit the security options for customers safety.
- Users can't return items if he/she had not been registered in rented details.

- Its lack for operation in management System for higher organizations which is used to maintain records in advance level.
- Users can return more than rented items.

Solutions

To implement in real world there are many factors that needed to be concerned before implementing it.

1. In Shop or departments First prepare a analysis data that can overview that how data is used or handle based on that program logic should be developed before coding which is also a good practice.
2. There should be skilled human resources for handling different systems and operations.
3. Technology need to be improved and research based on useful gadgets need to be done for smooth business operations.
4. Security need to be improved as need to be implemented where required.

6 References

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

<https://www.techtarget.com/whatis/definition/algorithm>

<https://www.programiz.com/dsa/algorithm>

<https://study.com/learn/lesson/pseudocode-examples-what-is-pseudocode.html>

2 Appendix

7.1 Operation_Files.py

```
import datetime
from datetime import timedelta
from Invoice_notes import generate_unique_invoice_id, generate_rent_invoice,
generate_return_invoice
from Print import COMPANY_NAME, ADDRESS, PHONE_NUMBER, EMAIL, LATE_FINE_PER_DAY
from Write_Files import write_inventory, create_invoice
from Read_Files import read_inventory
from Print import print_inventory
```

```

#-----CalculateFee Function-----
def calculate_late_fee(return_date, due_date):
    days_late = (return_date - due_date).days
    if days_late > 0:
        return days_late * LATE_FINE_PER_DAY
    else:
        return 0
#-----Rent Function-----

def rent_equipment(customer_name, rentals, rented_item_details):
    inventory = read_inventory()
    total_amount = 0
    invoice_content = f'{COMPANY_NAME}\nAddress: {ADDRESS}\nPhone: {PHONE_NUMBER}\nEmail: {EMAIL}\nCustomer Name: {customer_name}\nDate of Rental: {datetime.datetime.today().date()}\n'

    rented_item_details = [] # List to store rented item details

    for rental in rentals:
        item_found = False
        for item in inventory:
            if rental['code'] == item[0]:
                item_found = True
                if rental['quantity'] > item[4]:
                    print(f"Only {item[4]} {item[1]} available. Cannot rent {rental['quantity']} as requested.")
                    continue
                item[4] -= rental['quantity']
                total_amount += item[3] * rental['quantity']
                rented_item_details.append((item[1], item[2], item[3], rental['quantity'])) # Store item details
        if not item_found:
            print(f"Item with code {rental['code']} not found in inventory.")

    write_inventory(inventory)

    if rented_item_details:
        print() # Print some empty lines
        print()
        result = generate_rent_invoice(customer_name, rented_item_details, total_amount)

        return result
    else:
        return "No items rented. Invoice not created."

#-----Return Function-----

```

```

def return_equipment(customer_name, returns, rented_items_dict):
    valid_return = True # Initialize valid_return here
    inventory = read_inventory()
    loop = True
    while loop == True:
        try:
            return_date_str = input("Enter the Return Date (YYYY-MM-DD): ")
            return_date = datetime.datetime.strptime(return_date_str, '%Y-%m-%d').date()

            break
        except ValueError:
            print("Please enter a valid date in the format YYYY-MM-DD.")

    print_inventory() # Display inventory stock after customer name is entered
    invoice_content = f'{COMPANY_NAME}\nAddress: {ADDRESS}\nPhone: {PHONE_NUMBER}\nEmail: {EMAIL}\nCustomer Name: {customer_name}\nDate of Return: {return_date}\n'

    total_amount = 0

    rented_items = rented_items_dict.get(customer_name, []) # Retrieve rented items for the customer
    returned_item_details = [] # List to store returned item details

    for returned_item in returns:
        item_found = False
        for item in inventory:
            if returned_item['code'] == item[0]:
                item_found = True

            if returned_item['quantity'] > 0:

                if returned_item['quantity'] > item[4]:
                    print(f"Error: Cannot return more {item[1]} than were rented.")
                    valid_return = False
                else:
                    returned_item_details.append((item[1], item[2], item[3], returned_item['quantity'])) # Store item details
                    item[4] += returned_item['quantity']
                    total_amount += item[3] * returned_item['quantity']

        if not item_found:
            print(f"Item with code {returned_item['code']} not found in inventory.")
    if not valid_return:
        print("Invalid return. Please make sure you are returning items you have rented.")
    return

write_inventory(inventory)

```

```

if returned_item_details:
    due_date = (return_date + timedelta(days=5))
    late_fee = calculate_late_fee(return_date, due_date)

    # Create the invoice content
    invoice_content = generate_return_invoice(customer_name, return_date, returned_item_details,
    due_date, late_fee, total_amount)

    print()
    print()
    print(invoice_content)

    unique_invoice_name = generate_unique_invoice_id(customer_name) + "_return_invoice.txt"
    create_invoice(invoice_content, unique_invoice_name) # Assuming create_invoice saves the
content to a file

    return f"Return Invoice created: {unique_invoice_name}"
else:
    return "No items returned. Invoice not created."

```

7.2 Write_Files.py

```

def write_inventory(inventory):
    with open('item.txt', 'w') as file:
        for item in inventory:
            line = ','.join([str(item[0]), item[1], item[2], '$' + str(item[3]), str(item[4])]) + '\n'
            file.write(line)

def create_invoice(content, invoice_name):
    with open(invoice_name, 'w') as file:
        file.write(content)

```

7.3 Read_Files.py

```

def read_inventory():
    inventory = []
    with open('item.txt', 'r') as file:
        for line in file:
            sn, name, description, price, quantity = line.strip().split(',')
            price = int(price[1:].strip())
            quantity = int(quantity.strip())
            inventory.append([int(sn), name, description, price, quantity])

```


return inventory

7.4 Main_Files.py

```
from Print import print_inventory, MAX_RENT_ITEMS
from Operation_Files import rent_equipment, return_equipment
```

```
print("\n")  
print("\n")  
print("\t\t\t\t\t\t\t\t\t\tFashion Plaza")  
print("\n")  
print("\t\t\t\t\t\t\t\t\tLahan-1(Siraha)| Phone NO:-9824707814")  
print("\n")  
print("-----"  
-----")  
print("\t\t\t\t\t\t\t\tWelcome to Fashion Plaza - Build Your Style, Connect Your World!")  
print("-----"  
-----")  
print("\n")
```

```
def main():
    rented_items_dict = {} # Dictionary to store rented items for each customer
    returns = []
    rentals = []
```

```
while True:
    print("Employees Only")
    print("\nMenu:")
    print("1. Rent Equipment")
    print("2. Return Equipment")
    print("3. Exit")
    choice = input("Choose an option (1/2/3): ")
```

```
if choice == '1':  
    customer_name = input("Enter Customer Name: ")
```

```
print("Available Items:")
print("-----")
print_inventory()
print("-----")
```

while True:

```

    num_items = input("Number of Equipment Items to Rent: ")
    if not num_items.isdigit():
        print("Please enter a valid integer.")
        continue
    num_items = int(num_items)
    if num_items > MAX_RENT_ITEMS:
        print(f"You can rent a maximum of {MAX_RENT_ITEMS} items at a time.")
        continue
    break

```

rentals = []

for i in range(num_items):

loop = True

while loop == True:

try:

code = int(input(f"Enter Equipment Code {i+1}: "))

quantity = int(input(f"Enter Quantity {i+1}: "))

loop=False

if code <= 0 or quantity <= 0:

print("Please enter valid positive integers for code and quantity.")

continue

rentals.append({'code': code, 'quantity': quantity})

break

except:

print("Please enter valid positive integers for code and quantity.")

Store rented items in the dictionary

if customer_name not in rented_items_dict:

rented_items_dict[customer_name] = []

rented_items_dict[customer_name].extend(rentals)

Update inventory table with new item quantities after adding them to rental list

result = rent_equipment(customer_name, rentals, rented_items_dict)

print(result)

print("-----")

print("\nUpdated Inventory Stock:")

print_inventory()

print("-----")

print()

elif choice == '2':

customer_name = input("Enter Customer Name: ")

print_inventory() # Display inventory stock

```
while True:
```

```
    num_items = input("Number of Equipment Items to Return: ")
    if not num_items.isdigit():
        print("Please enter a valid integer.")
        continue
    num_items = int(num_items)
    break
```

```
returns = []
```

```
for i in range(num_items):
```

```
    loop = True
```

```
    while loop == True:
```

```
        try:
```

```
            code = int(input(f"Enter Equipment Code {i+1}: "))
```

```
            quantity = int(input(f"Enter Quantity {i+1}: "))
```

```
            loop=False
```

```
            if code <= 0 or quantity <= 0:
```

```
                print("Please enter valid positive integers for code and quantity.")
```

```
                continue
```

```
            returns.append({'code': code, 'quantity': quantity})
```

```
            break
```

```
        except ValueError:
```

```
            print("Only valid integers for code and quantity.")
```

```
# Check if entered customer name matches rented items' records
```

```
if customer_name in rented_items_dict:
```

```
    print(f"Items for {customer_name}:")
```

```
    for item in rented_items_dict[customer_name]:
```

```
        print(f"Equipment Code: {item['code']}, Quantity: {item['quantity']}")
```

```
    result = return_equipment(customer_name, returns, rented_items_dict)
```

```
    print(result)
```

```
    print("-----")
```

```
    print("\nUpdated Inventory Stock:")
```

```
    print_inventory() # Display updated inventory stock after returns
```

```
    print("-----")
```

```
    print()
```

```
else:
```

```
    print("Customer name does not match rented records. Please enter a valid customer name.")
```

```
elif choice == '3':
```

```
    break
```

```
else:
```

```
    print("Invalid choice, please select 1, 2, or 3.")
```

```
print("Thanks for using our software!")
```

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

7.5 Print.py

```
from Print import print_inventory , MAX_RENT_ITEMS
from Operation_Files import rent_equipment, return_equipment
```

```
print("\n")  
print("\n")  
print("\t\t\t\t\t\t\t\t\t\tFashion Plaza")  
print("\n")  
print("\t\t\t\t\t\t\t\t\tLahan-1(Siraha)| Phone NO:-9824707814")  
print("\n")  
print("-----"  
-----")  
print("\t\t\t\t\t\t\t\tWelcome to Fashion Plaza - Build Your Style, Connect Your World!")  
print("-----"  
-----")  
print("\n")
```

```
def main():
    rented_items_dict = {} # Dictionary to store rented items for each customer
    returns = []
    rentals = []

    while True:
        print("Employees Only")
        print("\nMenu:")
        print("1. Rent Equipment")
        print("2. Return Equipment")
        print("3. Exit")
        choice = input("Choose an option (1/2/3): ")
```

```

if choice == '1':
    customer_name = input("Enter Customer Name: ")

    print("Available Items:")
    print("-----")
    print_inventory()
    print("-----")

    while True:

        num_items = input("Number of Equipment Items to Rent: ")
        if not num_items.isdigit():
            print("Please enter a valid integer.")
            continue
        num_items = int(num_items)
        if num_items > MAX_RENT_ITEMS:
            print(f"You can rent a maximum of {MAX_RENT_ITEMS} items at a time.")
            continue
        break

    rentals = []
    for i in range(num_items):
        loop = True
        while loop == True:
            try:
                code = int(input(f"Enter Equipment Code {i+1}: "))
                quantity = int(input(f"Enter Quantity {i+1}: "))
                loop=False
                if code <= 0 or quantity <= 0:
                    print("Please enter valid positive integers for code and quantity.")
                    continue
                rentals.append({'code': code, 'quantity': quantity})
                break
            except:
                print("Please enter valid positive integers for code and quantity.")

        # Store rented items in the dictionary
    if customer_name not in rented_items_dict:
        rented_items_dict[customer_name] = []
    rented_items_dict[customer_name].extend(rentals)
    # Update inventory table with new item quantities after adding them to rental list

    result = rent_equipment(customer_name, rentals, rented_items_dict)
    print(result)
    print("-----")

```

```

print("\nUpdated Inventory Stock:")
print_inventory()
print("-----")
print()

elif choice == '2':
    customer_name = input("Enter Customer Name: ")
    print_inventory() # Display inventory stock

while True:

    num_items = input("Number of Equipment Items to Return: ")
    if not num_items.isdigit():
        print("Please enter a valid integer.")
        continue
    num_items = int(num_items)
    break

returns = []
for i in range(num_items):
    loop = True
    while loop == True:
        try:
            code = int(input(f"Enter Equipment Code {i+1}: "))
            quantity = int(input(f"Enter Quantity {i+1}: "))
            loop=False
            if code <= 0 or quantity <= 0:
                print("Please enter valid positive integers for code and quantity.")
                continue
            returns.append({'code': code, 'quantity': quantity})
            break
        except ValueError:
            print("Only valid integers for code and quantity.")
    # Check if entered customer name matches rented items' records
    if customer_name in rented_items_dict:
        print(f"Items for {customer_name}:")
        for item in rented_items_dict[customer_name]:
            print(f"Equipment Code: {item['code']}, Quantity: {item['quantity']}")
        result = return_equipment(customer_name, returns, rented_items_dict)
        print(result)
        print("-----")
        print("\nUpdated Inventory Stock:")
        print_inventory() # Display updated inventory stock after returns
        print("-----")
        print()

```

```

    else:
        print("Customer name does not match rented records. Please enter a valid customer name.")

    elif choice == '3':
        break
    else:
        print("Invalid choice, please select 1, 2, or 3.")

    print("Thanks for using our software!")
if __name__ == "__main__":
    main()

```

7.6 invoice_notes.py

```

import datetime
from datetime import timedelta
from Print import COMPANY_NAME, ADDRESS, PHONE_NUMBER, EMAIL
from Write_Files import create_invoice

#-----generate_unique_invoice Function-----
def generate_unique_invoice_id(customer_name):
    timestamp = datetime.datetime.today().strftime('%Y%m%d%H%M%S')
    return f'{customer_name}_{timestamp}'

#-----generate_rent_invoice Function-----
def generate_rent_invoice(customer_name, rented_item_details, total_amount):

    invoice_content = f'{COMPANY_NAME}\nAddress: {ADDRESS}\nPhone: {PHONE_NUMBER}\nEmail: {EMAIL}\nCustomer Name: {customer_name}\nDate of Rental: {datetime.datetime.today().date()}\n'
    invoice_content += "Items Rented:\n"
    for item in rented_item_details:
        invoice_content += f"- {item[0]} - {item[1]} - ${item[2]} - Quantity: {item[3]}\n"

    invoice_content += f'Total Amount: ${total_amount}\nRented Date: {datetime.datetime.today().date() + timedelta(days=5)}\nNote: Please return the equipment within 5 days to avoid fines.\n'

    unique_invoice_name = generate_unique_invoice_id(customer_name) + "_rent_invoice.txt"
    create_invoice(invoice_content, unique_invoice_name)

```

```
    return f"Invoice created: {unique_invoice_name}"
#-----generate_return_invoice Function-----
-----
def generate_return_invoice(customer_name, return_date, returned_item_details, due_date, late_fee,
total_amount):

    invoice_content = f'{COMPANY_NAME}\nAddress: {ADDRESS}\nPhone: {PHONE_NUMBER}\nEmail:
{EMAIL}\nCustomer Name: {customer_name}\nDate of Return: {return_date}\n'
    invoice_content += "Items Returned:\n"
    for item in returned_item_details:
        invoice_content += f"- {item[0]} - {item[1]} - ${item[2]} - Quantity: {item[3]}\n"
    invoice_content += f'Due Date: {due_date}\n'
    if late_fee > 0:
        invoice_content += f'Late Fee: ${late_fee}\n'
    invoice_content += f'Total Amount: ${total_amount + late_fee}\nThank you for returning the
equipment.\n'

    return invoice_content
```