LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2021-22 Summer

Student Name: Himanshu Yadav

Group: L1C5

London Met ID: 21049513

College ID: NP01CP4A210166

Assignment Due Date: 26th August 2022

Assignment Submission Date: 26th August 2022

I confirm that I understand my coursework needs to be submitted online via Google Classroom 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 mark of zero will be awarded.

Table of Contents

List of Figures

List of Tables

1. Introduction

Although there is a higher demand for costumes around Halloween, many customers have costume needs all year round. A costume shop provides its clients with wigs, makeup, and costumes for theatrical productions, festivals, and special occasions. Depending on the demands of their clients, successful stores offer a mix of things that can be rented and purchased. (TRUic, 2017)

A costume rental business is a fun opportunity, and the additional yearly income it can generate is outstanding. Once the item has paid for itself with five or six rentals, the revenue generated by the rental item from that point on is almost all profit the best way to market a costume rental business is to create a marketing brochure describing the costumes in stock and distribute the marketing brochure and rental rate sheet to organizations within the community, such as sports organizations, community theatre groups, charity associations, and colleges and universities. Halloween will be a huge time of year for you, so be sure to ramp up your marketing efforts in September and October. Once established, this type of rental business will easily be supported by word-of-mouth advertising and repeat customers. (Entrepreneur, 2022)

The Costume Rental System is a simple project intended to help rental stores manage costumes. A Costume Rental System is developed using the Python programming language, allowing the shop owner to maintain the rental and return process for costumes. This project demonstrates how to create a user interface for a Costume Rental System without using any Python GUI toolkits. The interface is designed simply, and the output is rendered on the terminal. There are three types of modules in the Costume Rental System that are responsible for performing the functional activities of the system. The main module, rentCostume Module, returnCostume Module, and costumes.txt file contain details concerning costumes such as costume name, costume brand,

price and quantity. To specify the shop items, the system also uses the idea of structures. Additionally, it efficiently uses a variety of Python principles, including functions for manipulating strings, looping, and file operations.

The rent and return module generate a note for each transaction. The name of the customer, the name of the costume rented with Brand name, cost, and quantity, as well as the date and time the costume was rented by the user, are generated on the rent note and return note, respectively, when someone rents a costume. Additionally, the letter generates the total amount due for the rental. Additionally, the rent module's conditional function and Python loops are used to accomplish the multiple costume renting condition. Therefore, if a person chooses to rent numerous costumes, each costume should be listed on the note, along with the total cost, which is also calculated on the note. A note ought to be created and added to the file once more when a costume is returned to the store. The note includes the customer's name, the costume's name and brand, the number of costumes being returned, the date, and the time of the return. The loan period is set at 5 days, and if the costume is returned after that time, a daily $5 fine is added to the return note.

1.1 Goals of Costume Rental Shop

The primary goal of this project was to create a system for costume rentals that stores costume information in a text file. It was necessary to create an application that will read the text file and show every outfit that is rentable. Then, a note for the unique borrower should be generated and recorded in a file for each transaction involving renting and returning. The costume supply should also be updated after every transaction.

Some of the important points are listed below:

1. An application needs to be developed which will read the text file and display all the costumes available for renting.

2. Then with each transaction renting a note should be generated for the particular rental and should be written into a file.
3. The stock of the costumes should also be updated after each transaction according to the number of costumes taken by the user and its quantity must be decreased by the number of the costumes taken by the user.
4. In the case of returning a costume, a note should again be generated for the person returning the costume.
5. After the user return costume, the stock should also be updated.
6. The quantity of the costume returned should be increased in the costume stock.
7. The maximum delay late for costume return should be 5 days.
8. If the borrower returns the costume late, then a fine should be imposed of $5 by each day.

1.2 Objective of Costume Rental Shop

The project mainly focuses on the following objectives:

1. To create a project using python programming and its features
2. To implement features like control statements, structures, and exception handlings
3. To make the program interface easy and user friendly
4. To get an idea about making a simple project using python.
5. To learn to be able to develop complex programs aimed at solving real-life projects like Costume Rental System and get knowledge to solve real problem-based projects.
6. To be able to solve problem-based problems by developing the python script, its compiling, and debugging.

1.3 Tools Used

- **Word:**

  MS Word is a word processer developed by Microsoft which is used to male professional quality document, letters, reports etc. It has advanced features which allow you to format and edit your files and documents in the best possible way. It is one of the office productivity applications included in the Microsoft Office suite. Originally developed by Charles Simonyi and Richard Brodie, it was first released in 1983. MS-Word lets you create professional quality documents, reports, letter and resumes. Unlike a plain text editor, MS Word has features including spell check, grammar check, text and font formatting, HTML support, image support, advanced page layout, and more.

- **Python (idle):**

  IDLE stands for Integrated development and learning environment. IDLE provides an integrated environment for a python. When you install a python in your desktop IDLE is installed by default. It works the same in all platforms like Linux, Windows, and macOS. It is coded purely in python, using the tinker Graphical User Interface (GUI) toolkit. It is cross-platform. It works smoothly and almost similarly on the operating Systems. It has Python Shell window in which you can write code, you get an error message and output also. A multi-window editor is also available in IDLE.

- **Draw. io:**

  Draw.io is proprietary software for making diagrams and charts. The Software lets you choose from an automatic layout function or create a custom layout. They have a large selection of sharps of shapes and hundreds of visual elements

to make your diagram or charts one of a kind. The drag and drop features male its simples to create a greater looking diagram or chart.

2. Discussion and Analysis

2.1 Algorithm

An algorithm is a producer used for solving problem or performing a computation. It acts as an ex-act list of instructions that conduct specified actions step by step. Algorithms are widely used throughout all areas of IT. It can be simplex and complex depending on what you want to achieve. It can be understood by taking the example of cooking a new recipe. To cook a new recipe, one read the instructions and steps and excites them one by one in the given sequence. The result thus obtained is the new dish cooked perfectly.

Step 1. Start
Step 2. Display the welcome screen
Step 3. Input user option
Step 4. if user input = 1, then go to step 5, elif user input = 2, then go to step 9, elif user input is 3, then go to step 13, else go to step 3
Step 5. Display the costumes list and ask for serial number and quantity for renting purpose.
Step 6. Ask user if they want to rent more costumes or not.
Step 7. if user wants more costume, then go to step 5, elif user doesn't want more costume then go to step 8
Step 8. Ask username, address, phone Number for generating rent bill and generate the rent, Bill.
Step 9. Display the costumes list and ask for serial number and quantity for returning purpose.
Step 10. Ask user if they want to return more costumes or not.

Step 11. if user wants to return more costume, then go to step 9, elif user doesn't want more costume then go to step 12

Step 12. Ask username, address, phone Number for generating rent bill and generate the return, Bill.

Step 13. Display a welcome screen and the exit the program

Step 14. Stop

2.2 Flowchart

Flowchart is a graphical representation of an algorithm. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting", or program-planning.

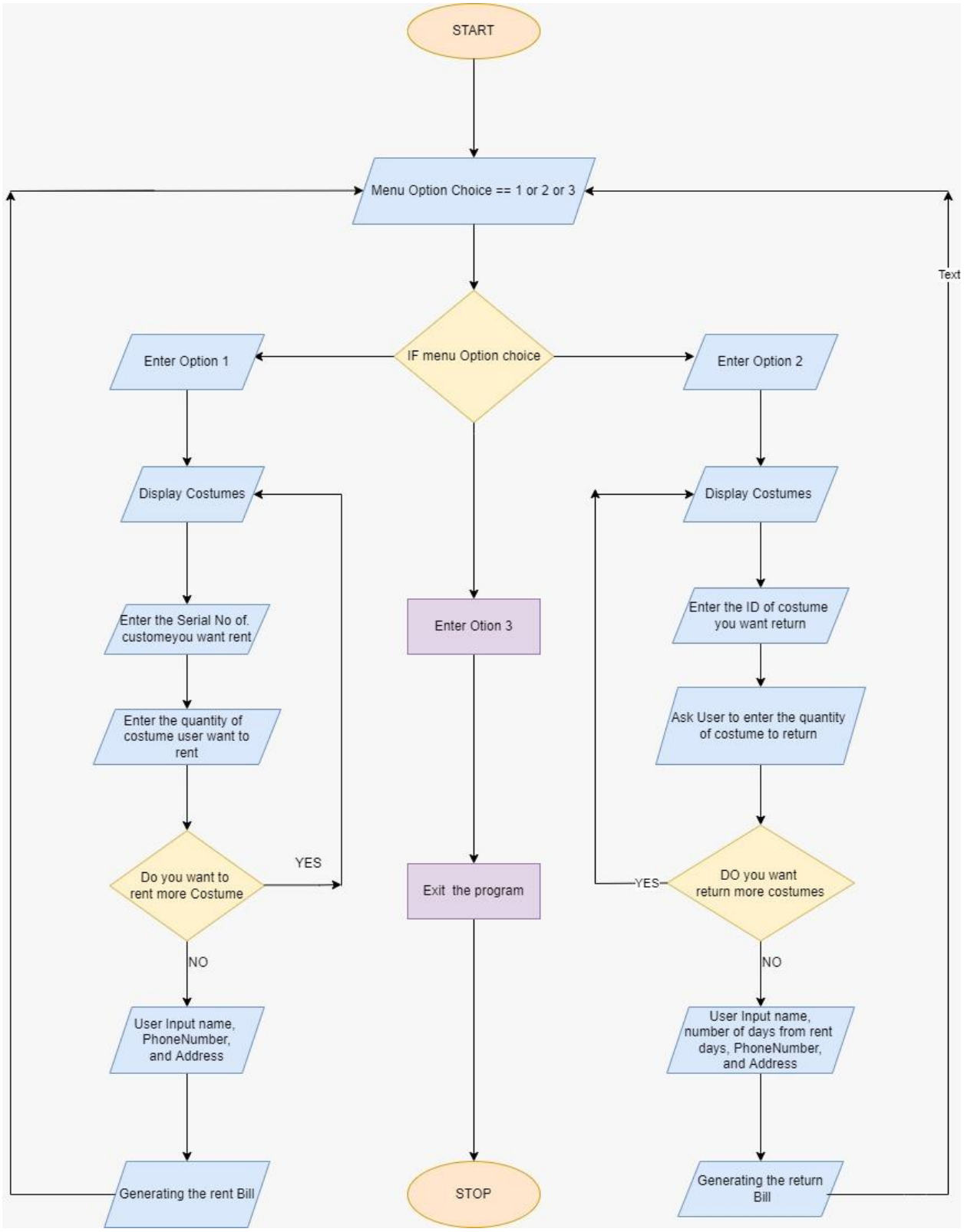


Figure 1: Flowchart of program

2.3 Pseudocode

Pseudocode is a method of planning which enables the programmer to plan without worrying about syntax. It is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft of a program. Pseudocode summarizes a program's flow but excludes underlying details. System designers write pseudocode to ensure that programmers understand a software project's requirements and align code accordingly.

Main Class

START
IMPORT rentCostume, returnCostume
CREATE Non parameterized function name as welcome
PRINT ----------
PRINT an empty line
PRINT Welcom**e** to costume rental application
PRINT Designed by Himanshu Yadav :)
PRINT an empty line
PRINT ----------
CREATE non parameterized function name as displayingMessage
WHILE True
PRINT Select a desirable option
PRINT (1) || Press 1 to rent a costume.
PRINT (2) || Press 2 to return a costume.
PRINT (3) || Press 3 to exit.
CREATE a variable name selectedOption and **INPUT** Enter a Option
IF selecte**DO**ption is equal to 1
PRINT \n
PRINT Let's rent a costume😊
PRINT \n
CALLING a non-parameterized function named as rentCostume.rentCostume()

ELIF selecte**DO**ption is equal to 2
PRINT \n
PRINT Let's **RETURN** a costume😊
PRINT \n

```
                    CALLING a non-parameterized function named as
    RETURNCostume.RETURNCostume()

                ELIF selectedOption is equal to 3
                    PRINT Thank You FOR using our application :)"
                    EXIT
                ELSE
                    PRINT Invalid INPUT! ☹
                    PRINT Please select the value as per the provided options :)
                ENDIF
    CALLa function welcome
    CALL a function  displayMessage
END
```

rentCostume class

```
    START
        IMPORTdatetime, rando m
        CREATE a non-parameterized function name as getFileContent
          DO
            File OPEN from costumes.txt in r Format
            CREATE a  data with file.readlines
            CLOSE the file
            RETURN data
        CREATE a parameterized function getDictionary by fileContent
            CREATE an empty SET of data
            FOR index in range(length of fileContent)
                Data[index+1] is equal to fileContent[index].repalcing \n
    .spliitng by ,
            RETURN data
        CREATE a parameterized function name PRINTCostumes by mainData
            PRINT ---------------------------------------------------------------------------
            PRINT S.No., "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t",
    "Quantity"
            PRINT---------------------------------------------------------------------------
            FOR key,value in mainData.items():
                PRINTstr(key)+str("."), "\t", value[0], "\t\t", value[1], "\t\t",
    value[2], "\t\t", value[3]
            PRINT---------------------------------------------------------------------------
            RETURN
        CREATE a parameterized function name getValidSno by mainData
            CREATE a variable validSno with False value
            WHILE validSno is equal to False
```

DO
CREATE a variable SNo and **INPUT** Enter Serial Number:
IF SNo.isdigit()
CREATE a variable SNo as int SNo
IF SNo is grater than 0 and SNo is les than equal length of mainData
IF int mainData[SNo][3] is equal to 0
PRINT an empty line
PRINT Costume is out of stock!
PRINT an empty line
PRINT Try another!
PRINT an empty line
PRINT PRINT Costumes(mainData)

CONTINUE
ELSE

validSno = True
PRINT f"The serial number of the costume is {SNo}.
PRINT an empty line
PRINT----------
PRINT The costume is available. ")

PRINT ----------
RETURN SNo

ELSE
PRINT your number is out of the Options provided.
PRINT \n
ELSE
PRINT Please type a number next time.
PRINT \n
ENDO
ENDWHILE

CREATE a parameterized function name getValidQuantity by mainData, SNo
CREATE an empty list name cart
CREATE an empty list tempRntBill
CREATE a variable validQuantity as false
WHILE validQuantity is equal to false
DO

CREATE a variable quantity and **INPUT** Enter the total number of fresses you want to rent
IF quantity.isdigit
Quantity is equal to int quantity
IF quantity is greater than 0 and quantity is less than equal to int mainData[SNo] [3]
validQuantity is equal to true
mainData [SNo][3] is equal to String (int(mainData [SNo] [3] - quantity))
RETURN quantity
ELSE
PRINT Quantity provided is greater than we have in Stock
PRINT So, Please enter the quantity which **DO**esn't goes exceeding our stocks
PRINT \n
ELSE
PRINT Please type a number next time
PRINT \n
CREATE a non parameterized function named as rentcostumes
CREATE a variable userWantsClothes as True
CREATE an empty list as Cart
CREATE an empty list as tempRentBill
WHILE userWantsClothes is equal to true
DO
PRINT PRINT Costumes with maindata
INITIALIZE SNo as getValidSNowith mainData
INITIALIZE quantity as getValidQuantity with mainData and SNo
CREATE a variable flag as TRUE
FOR costume in cart
IF costume index 0 is equalt to SNo variable
Costume[1] +qquantity

Flag is equal to false
IF flag
APPEND mainData[SNo][0] and quantity to cart
APPEND mainData[SNo][0], mainData[SNo][1], mainData[SNo][2] and quantity
Valid_**INPUT** is equal to false
WHILE valid_**INPUT** is equal to false
DO
INITIALIZE a variable wantAnother and **INPUT** wanna rent more(yes/no)?
IF wantAnother.lower is equal to Yes
PRINT Youe cart: {cart}

```
                                        Valid_INPUT is equal to true
                                        BREAK
                                    ENDIF
                                    ELIF wantAnother.lower is equal to no
                                        CALL generateBill with tempRentBill
                                        INITALIZE usewantclothes as false
                                        INITALIZE Valid_INPUT as false
                                    ENDEIF
                                    ELSE
                                        PRINT Invalid INPUT !!
                                        CONTINUE
                                    ENDELSE
                                ENDDO
                            ENDWHILE
                            CALL updateTextFile with mainData function
                        ENDDO
                    ENDWHILE
        CREATE a parameterize function named as generatBill by tempRentBill
            INITIALIZE validName as False
            WHILE validName is equal to false
              DO
                INITIALIZE a variable name as String and INPUT Enter Your Name
                IF Name. replace “” isalpha
                    Validname is equal to True
                ENDIF
              ENDO
            ENDWHILE
            ValidPhoneNumber is equal to False
            WHILE validPHonenumber is equal to false
              DO
                INITIALIZE a variable phonenumber as String and INPUT Entr you
PhoneNumber
                IF phoneNumber. Isdigit
                    ValidPhoneNumber is equal to False
                ENDIF
              ENDDO
            ENDWHILE
            INITIALIZE a variable Address as String and INPUT Enter your Address:
            INITIALIZE a variable GST as random.randint(1000, 5000)
            INITIALIZE a variable Year as datetime.datetime.now().year
            INITIALIZE a variable Month as datetime.datetime.now().Month
            INITIALIZE a variable Day as datetime.datetime.now().Day
            INITIALIZE a variable Hour as datetime.datetime.now().hour
            INITIALIZE a variable Minute as datetime.datetime.now().minute
            INITIALIZE a variable Second as datetime.datetime.now().second
```

```
        INITIALIZE a variable microsecon as
datetime.datetime.now().microsecond

        PRINT ----------Costume Rental Shop----------
        PRINT                    Kathmandu, Nepal
        PRINT f"                   GST.NO:-{GST}
        PRINT ------------------------------Rent Bill-------------------------
        PRINTf"Date:- {Year}-{Month}-{Day}
Time:- {Hour}:{Minute}:{Second}:{microSecond}
        PRINT f"Customer Name: {Name}
        PRINT f"Phone Number: {phoneNumber}"
        PRINT S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Rate", "\t\t",
"Quantity", "\t\t", "Amount
        INITAILIZE a variable row as Empty string
        INITIALIZE a variable counter as 0
        INITILIZE a variable finalPrice as 0
        FOR i in range as length of tempRentBill
                Counter += 1
                FOR j in range as length as tempRentBill[i]
                        INITIALIZE variable dollarpice as Float with
tempRentBill[i][2].replace("$","")
                        INITIALIZE variable priceDetail as dollaprice *
tempRentBill[i][j]
                        INITIALIZE variable row as row plus tempRentBill[i][j] as
String
                PRINT counter,"\t",row, "\t", priceDetail
                Finalprice is equal to finalprice + priceDetail
                INITAILIZE a variable row as Empty string
        PRINT f"                    Total Price: ${finalPrice}
        PRINT                  Thank you for vising our store
        PRINT                       Visit again :)
        PRINT an empty line
        PRINT ---------Bill has also been generated in txt file.---------


        Text is equal to Rent –{Name} to txt file
        OPEN file
        SET file.write("\n")
        SET file.write                          Kathmandu, Nepal
        SET file.write("\n")
        SET file.write                          GST.NO:-{GST}
        SET filewrite("\n")
        SET file.write ----------Rent Bill----------
        SET file.write("\n")
        SET file.write f"Date:- {Year}-{Month}-{Day}
Time:- {Hour}:{Minute}:{Second}:{microSecond}
```

```
            SET file.write("\n")
            SET file.write f"Customer Name: {Name}"
            SET file.write("\n")
            SET file.write f"Phone Number: {phoneNumber}
            SET file.write("\n")
            SET file.write("\n")
            SET file.write S.No. \t Costume Name \t\t Brand \t\t\t price \t\t Quantity \t\t
      A
            INITIALIZE a variable row as empty

            INITIALIZE a variable counter as 0
            INITILIZE a variable finalPrice as 0
            FOR i in range as length of tempRentBill
                  Counter += 1
                  FOR j in range as length as tempRentBill[i]
                        INITIALIZE variable dollarpice as Float with
      tempRentBill[i][2].replace("$","")
                        INITIALIZE variable priceDetail as Dolla price *
      tempRentBill[i][j]
                        INITIALIZE variable row as row plus tempRentBill[i][j] as
      String
                  PRINT counter,"\t",row, "\t", priceDetail
                  Finalprice is equal to finalprice + priceDetail
                  INITAILIZE a variable row as Empty string
            PRINT f"                              Total Price: ${finalPrice}
            PRINT                           Thank you for vising our store
            PRINT                                 Visit again :)
            CLOSE file



      CREATE a parameterize function updateTextFile by mainData
            OPEN file
            FOR I in mainData. Value
                  file.write(str(value[0]) + "," + str(value[1]) + "," + str(value[2]) + "," +
str(value[3]) + "\n")
            CLOSE file
      SET fileContent to getFileContent
      SET mainData to getDictionaryby parameter fileContent
END


returnCostume class

      START
            IMPORT datatime, random
```

CREATE a non parameterize function name as getfilecontent
OPEN file
SET data as file.readlines
CLOSE file
CREATE a parameterize function name as getDictionary by filecontent
SET a variable data as empty set
FOR index in range by length of fileContent
INITIALIZE a variable data[index+1] fileContent[index].replaceing “” and spliting by ,
RETURN data
CREATE a parameterize function name as **PRINT**Costume by mainData
PRINT S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t", "Quantity
FOR key,value in mainData.item
PRINT str(key)+str("."), "\t", value[0], "\t\t", value[1], "\t\t", value[2], "\t\t", value[3]
RETURN
CREATE a parameterize function name as getValidID by mainData
INITIALIZE validID as false
WHILE validID is equal to false
DO
INITIALIZE a variable ID and **INPUT** Enter the costumeID to return
IF ID.digit
INITIALIZE a variable ID as Integer ID
IF ID is greater than 0 and ID is less than equal to length of mainData
INITALIZE validID to True
RETURN ID
BREAK
ENDIF
ELSE
PRINT CREATE a parameterize function name as **PRINT**Costume by mainData
ENDELSE
ELSE
PRINT Please type a number next time.
ENDELSE
CREATE a parameterize function name as getValidreturnQuantity by mainData and ID
INITAILIZE renturn cart as empty list
INITIALIZE validQuad to false
WHILE validQuad is equal to False
INITIALIZE a variable quantity and **INPUT** Enter the quantity you wanna return:
IF quantity. Isdigit

INITIALIZE quantity as INT quantity
INITIALIZE validQuad to True
SET mainData[ID][3] as int(mainData[ID][3]) + quantity
APPEND mainData[ID][0], mainData[ID][1], mainData[ID][3] to RETURNcart
ENDIF
ELSE
PRINT Please enter a number not anything **ELSE**!
ENDELSE
ENDWHILE
CREATE a non parameterize function name as RETURNCostume
INITIALIZE variable userRETURNClothes to false
INITIALIZE a variable RETURNcart as empty ,list
WHILE user**RETURN**Clothes is equal to True
PRINT PRINT Costumes(mainData)
SET variable ID as getValidID with mainDAta
SET variable quantity as getValidRETURNQuantity with mainData and ID
CREATE a variable flag as TRUE
FOR costume in RETURNcart
IF costume index 0 is equalt to IDvariable
Costume[1] +q=quantity

Flag is equal to false
IF flag
APPEND mainData[SNo][0], mainData[SNo][1], mainData[SNo][2] and quantity to RETURNcart
Valid_**INPUT** is equal to false
WHILE valid_**INPUT** is equal to false
DO
INITIALIZE a variable RETURNAnother and **INPUT** wanna **RETURN** more(yes/no)?
IF RETURN Another is equal to Yes and CONVERT as lower
Valid_**INPUT** is equal to true
BREAK
ENDIF
ELIF RETURN Another is equal to no and CONVERT as lower
CALL generate**RETURN**Bill with **RETURN**cart
INITALIZE user**RETURN**clothes as false
INITALIZE Valid_**INPUT** as True
ENDE**IF**
ELSE
PRINT Please enter a option from given options only!

CONTINUE
ENDELSE
ENDDO
ENDWHILE
CALL updateTextFile with mainData function
ENDDO
ENDWHILE
CREATE a parameterize function name as generate**RETURN**Bill by retuencart
INITIALIZE variable validname as false
WHILE validName is equal to false
DO
INITIALIZE a variable Name as STeing and **INPUT** please enter your name
IF Name.replace(" ", "").isalpha():
INITIALIZE a variable validName toTrue
ENDIF
ELSE
PRINT Sorry, You mistakely typed your name wrong!
ENDELSE
ENDDO
ENDWHILE

INITIALIZE variable validDay as false
WHILE validDay is equal to false
DO
INITIALIZE a variable day and **INPUT** please enter number of Day from rent days:
IF day.isdigit
INITIALIZE a variable day to INT day
ENDIF
ELSE
PRINT Please enter the days which is always in number
ENDELSE
ENDDO
ENDWHILE
INITIALIZE variable validphonenumber as false
WHILE validphonenumber is equal to false
DO
INITIALIZE a variable phonenumber and **INPUT** please enter your phone number
IF phonenumber.isdigit
INITIALIZE a variable validphonenumber to True
ENDIF
ELSE

PRINT Sorry, You mistakely typed your Phone Number wrong! number
ENDELSE
ENDDO
ENDWHILE
INITIALIZE a variable Address as String and **INPUT** Enter your Address:
INITIALIZE a variable GST as ran**DO**m.randint(1000, 5000)
INITIALIZE a variable Year as datetime.datetime.now().year
INITIALIZE a variable Month as datetime.datetime.now().Month
INITIALIZE a variable Day as datetime.datetime.now().Day
INITIALIZE a variable Hour as datetime.datetime.now().hour
INITIALIZE a variable Minute as datetime.datetime.now().minute
INITIALIZE a variable Second as datetime.datetime.now().second
INITIALIZE a variable microsecon as datetime.datetime.now().microsecond

PRINT ----------Costume Rental Shop----------
PRINT Kathmandu, Nepal
PRINT f" GST.NO:-{GST}
PRINT ----------Rent Bill----------
PRINTf"Date:- {Year}-{Month}-{Day} Time:- {Hour}:{Minute}:{Second}:{microSecond}
PRINT f"Customer Name: {Name}
PRINT f"Phone Number: {phoneNumber}"
PRINT S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Rate", "\t\t", "Quantity", "\t\t", "Amount
INITAILIZE a variable row as Empty string

INITIALIZE a variable counter as 0

IF day is greater than 5
SET a variable newDay to day minus 5
SET a variable fine as new * 5
ENDIF
ELSE
INITIALIZE variable fine as zero
ENDELSE

FOR I in range upto length of **RETURN**cart
Counter += 1
FOR j in range as length as tempRentBill[i]

INITIALIZE variable row as row plus tempRentBill[i][j] as String
PRINT counter,"\t",row, "\t", priceDetail

INITAILIZE a variable row as Empty string
PRINT f" fine: ${fine}
PRINT Thank you **FOR** vising our store
PRINT Visit again :)
PRINT an empty line
PRINT ---------Bill has also been generated in txt file.---------

Text is equal to **RETURN**–{Name} to txt file
OPEN file
SET file.write("\n")
SET file.write Kathmandu, Nepal
SET file.write("\n")
SET file.write GST.NO:-{GST}
SET filewrite("\n")
SET file.write ----------Rent Bill----------
SET file.write("\n")
SET file.write f"Date:- {Year}-{Month}-{Day}
Time:- {Hour}:{Minute}:{Second}:{microSecond}
SET file.write("\n")
SET file.write f"Customer Name: {Name}"
SET file.write("\n")
SET file.write f"Phone Number: {phoneNumber}
SET file.write("\n")
SET file.write("\n")
SET file.write S.No. \t Costume Name \t\t Brand \t\t\t price \t\t
Quantity \t\t A
INITIALIZE a variable row as empty

INITIALIZE a variable counter as 0
INITILAIZE a variable finalPrice as 0
FOR i in range as length of **RETURN**cart
Counter += 1
FOR j in range as length as **RETURN**cart[i]

INITIALIZE variable row as row plus tempRentBill[i][j]
as String

INITAILIZE a variable row as Empty string
SET file.write f" fine: ${fine}
SET file.write Thank you **FOR** vising our
store
SET file.write Visit again :)
CLOSE file

```
        CREATE a parameterize function updateTextFile by mainData
                OPEN file
                FOR I in mainData. Value
                        file.write(str(value[0]) + "," + str(value[1]) + "," + str(value[2]) + "," +
str(value[3]) + "\n")
                CLOSE file
        SET fileContent to getFileContent
        SET mainData to getDictionaryby parameter fileContent
         ENDDO
END
```

2.4 Data Structures

A customized format called a data structure is used to store data in an ordered manner. A data structure may be used in computer programming to hold data so that several methods can be applied to it. We have two different types of data structures: collection data types, which include objects like tuples and dictionaries, and primitive data types, which include objects like String, Integer, and Float. Some of the most fundamental Python data structures are lists, sets, tuples, and dictionaries. Each data structure is unique in and of itself. Data structures act as "containers" for classifying and organizing data according to type. Lists, sets, and tuples are the three fundamental types of data structures in the Python programming language. The ability to modify something after it has been generated, or mutability, is one of the variations between data structures.

Lists and tuples are the most practical data types, and they can be found in practically any Python program.

Primitive Date Types

The most basic data structures are primitive data types. They are the building block for data manipulation. There are four primitive types: integers, floats, Booleans, and strings.

a) Integer

When we want to represent numeric data, especially whole numbers, we use integers. Whole numbers can also be negative. When our variable name has many words, we can use an underscore to separate them. In the Library management system program it is used everywhere we must ask the whole number as input and even while calculation the decreasing and increasing in quantity. Also, Integer is used when we want to ask integer value from the user like a choice option in the library menu. The costume ID while renting the costumes as well as returning the costumes.

```
def getValidSno(mainData):

    """we are validating the serial number."""

    validSno = False
    while validSno == False:
        SNo = input("Enter Serial Number: ")
        try:
            if SNo.isdigit():
                SNo = int(SNo)
                if SNo > 0 and SNo <= len(mainData):
                    if int(mainData[SNo][3]) == 0:
                        print()
                        print("Costume is out of stock!")
                        print()
                        print("Try another!")
                        print()
                        print(printCostumes(mainData))
                        continue
```

Figure 2: Screenshot of Program showing use of Integer data type

b) Float

The floating-point numbers are represented by afloat. It works with decimals and rational numbers like 1.5 and 2.5. Float is almost similar to integer, but it also

represents the decimal number. In the program, a float is used to calculate the cost to be paid while renting and returning the costumes as there may be float numbers in the cost of costumes in the shop.

```
counter = 0
finalPrice = 0
for i in range(len(tempRentBill)):
    counter += 1
    for j in range(len(tempRentBill[i])):
        dollarprice = float(tempRentBill[i][2].replace("$",""))
        priceDetail = dollarprice * tempRentBill[i][3]
        row = row + str(tempRentBill[i][j]) + "\t\t"
    print(counter,"\t",row, "\t", priceDetail)
    finalPrice = finalPrice + priceDetail
    row = ""
```

Figure 3: Screenshot of Program showing use of Float data type

c) String

Strings are character data sequences. In Python, the string term can be defined as str. Single or double quotes can be used to separate string literals. The string includes all characters between the opening delimiter and the matching closing delimiter. A string holds the value which can be either a number or alphabet or any symbols but here the number is like an alphabet, not a number. Here, the number loses its addition subtraction and all other properties. In the costume management system program costumes names and brands are written in the string and even costume quantity and its price are converted in the string after all the calculations.

```
validName = False
while validName == False:
    Name = str(input("Enter your name: "))
    if Name.replace(" ", "").isalpha():
        validName = True
```

Figure 4: Screenshot of Program showing use of String data type

d) Boolean

True and False are the two possible values for a Boolean data type. They can in useful when building conditional and comparison expressions. One of Python's built-in data types is the Boolean type. It's used to represent an expression's truth value. The bool keyword is used to declare a Boolean data type, which can only take the values true or false. True = 1 and false = 0 when the value is returned.

```
while userWantsClothes == True:
    print(printCostumes(mainData))
    SNo = getValidSno(mainData)
    quantity = getValidQuantity(mainData,SNo)

    flag = True
    for costume in cart:
        if costume[0] == SNo:
            costume[1] += quantity
            flag = False
    if flag:
        cart.append([mainData[SNo][0], quantity])
        tempRentBill.append([mainData[SNo][0], mainData[SNo][1], mainData[SNo][2], quantity])
```

Figure 5: Screenshot of Program showing use of Boolean data type

Collection Data Types

In this program, we have used some collection data types. There are four types of collection data types: List, Dictionary, Set, and Tuple. In this program, we have used List and Dictionary as collection data types.

1. **List**

   The list is one of the most commonly used collection data types. In the list, data stored are stored in sequence with the proper index. List are mutable data types so can be modified easily. In the program, a list is used to display the costume name, Brand, costume quantity, cost from the text file, and while writing.

```
    cart = []
    tempRentBill = []
    validQuantity = False

    while validQuantity == False:
        quantity = input("Enter the total number of dresses you want to rent: ")
        try:
            if quantity.isdigit():
                quantity = int(quantity)
                if quantity > 0 and quantity <= int(mainData[SNo][3]):
                    validQuantity = True
                    mainData[SNo][3] = str(int(mainData[SNo][3]) - quantity)
                    return quantity
```

Figure 6: Screenshot of Program showing use of List data type

2. **Dictionary**

   Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is ordered*, changeable and do not allow duplicates. Dictionaries are written with curly brackets, and have keys and values. Dictionary items are ordered, changeable, and does not allow duplicates. Dictionary items are presented in key:value pairs, and can be referred to by using the key name. (W3schools.com, 2020)

```
def getDictionary(fileContent):

    """Here, we are arranging the data from which is in list to Dictionary."""

    data = {}
    for index in range(len(fileContent)):
        data[index+1] = fileContent[index].replace("\n","").split(",")
    return data


def printCostumes(mainData):

    """Here, we create a table which enhances the user experince."""

    print("-------------------------------------------------------------------------------------")
    print("S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t", "Quantity")
    print("-------------------------------------------------------------------------------------")
    for key,value in mainData.items():
        print(str(key)+str("."), "\t", value[0], "\t\t", value[1], "\t\t", value[2], "\t\t", value[3])
    print("-------------------------------------------------------------------------------------")
    return ""
```

Figure 7: Screenshot of Program showing use of Dictionary data type

3. Program

A. Implementation of Program

Python programming was used to construct the project, which will assist the costume business in maintaining its inventory. Various data structures were used in the creation of this software. The project shows how to build a costume rental system's user interface without using any Python GUI toolkits; instead, the interface is simply constructed, and the output is shown on the terminal. This program is made by developing many modules and functions for various goals. The system's three different sorts of modules control the costume management system's operational tasks. They are the Main Module, Rent Module, Return Module, and finally, a file called costumes.txt that provides information on the costumes, like their name, brand, quantity, and price. To specify the costume items, the system also uses the idea of structures. The application scans the text file and prints every costume that is offered from it. The application operates in accordance with user input, such as whether the user wishes to access the main menu, select the option to rent a costume, or return the rented outfit. The costume rental or return process does not cause the software to shut down; rather, the user must choose to do so. The application can read a text file to display the costumes and the number of them that are available. It can also update when a costume is leased or returned. The invoice is generated in a new text file each time a costume is rented or returned by a user. Exception handling is used in the program so if the user input the wrong value program won’t get closed.

B. Showing renting, returning, Bill generating of costume

Rent Module

The rent function is created in this module, and when a user rents costumes, information like a user name and phone number are requested. Additionally, the User Name and Phone Number are validated. The costume ID must be entered when the user wishes to choose the costume for rental. The text file is then used to create a notice or an invoice. A unique note is issued for each rent paid by a distinct user. When a user hires a costume, the rent function is called, reading the text file and writing the costume's details there. Also When a user rents costumes, the number of costumes in the file containing already-existing costumes is reduced by the amount of costumes the user has rented. To prevent this and ensure the program runs well, the validation for renting additional costumes and costume ID is also completed. Additionally, the rent module's conditional function and Python loops are used to implement the multiple costume rent condition. Therefore, if a person chooses to rent numerous costumes, each costume should be listed on the note, along with the total cost, which is also calculated on the note.

```
-----------------------------------------------------

        Welcome to costume rental application
            Designed by Himanshu Yadav :)

-----------------------------------------------------


Select a desirable option
(1) || Press 1 to rent a costume.
(2) || Press 2 to return a costume.
(3) || Press 3 to exit.
Enter a option: 1


Let's rent a costume :)


-------------------------------------------------------------------------------------
S.No.    Costume Name           Brand                  Price          Quantity
-------------------------------------------------------------------------------------
1.       Formal Suits           MegaPlex               $14.5          500
2.       Fairy Costume          DollarSmart            $18            500
3.       Thor Costume           Marvel INC             $23            500
4.       Ant Costume            DC Comics              $25            500
-------------------------------------------------------------------------------------

Enter Serial Number: 1
The serial number of the costume is 1.

-----------------------------------------
      The costume is available.
-----------------------------------------


Enter the total number of dresses you want to rent: 100
```

Figure 8: Screenshot of program showing renting process

```
-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-4123
-------------------------------------------------Rent Bill-------------------------------------------------
Date:- 2022-8-26                                                              Time:- 0:3:39:726430
Customer Name: Himanshu Yadav
Phone Number: 9852832265
-----------------------------------------------------------------------------------------------------------
S.No.    Costume Name           Brand                  Rate           Quantity               Amount
-----------------------------------------------------------------------------------------------------------
1        Formal Suits           MegaPlex               $14.5          100                    1450.0
-----------------------------------------------------------------------------------------------------------
                                      Total Price: $1450.0
-----------------------------------------------------------------------------------------------------------
                                Thank you for vising our store
                                      Visit again :)

------------------------------Bill has also been generated in txt file.------------------------------
```

Figure 9: Screenshot of program printing Rent Invoice in terminal

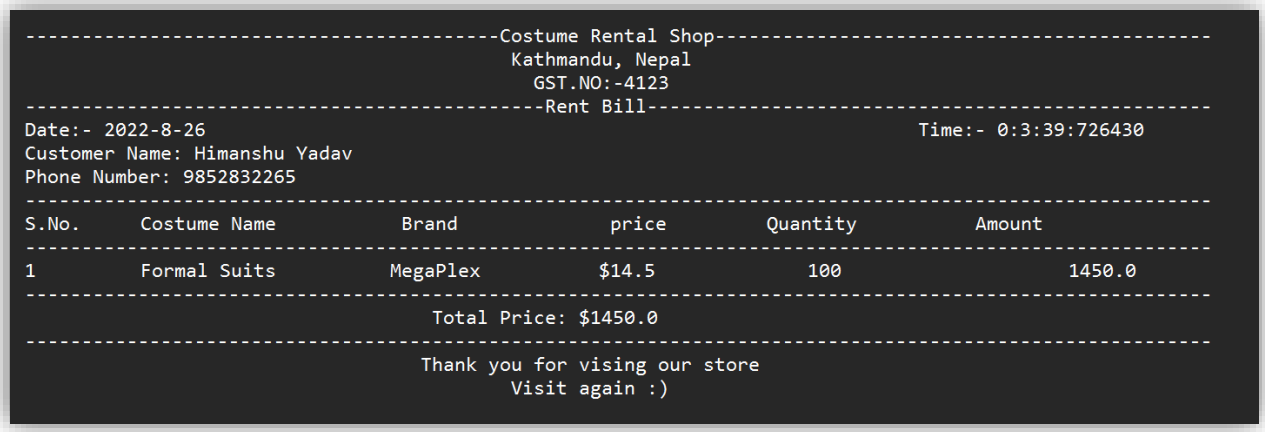


Figure 10: Screenshot of program printing Rent Invoice in txt format

Return Module

The return function is created in this module, and when a user returns costumes, information like a user name and phone number are requested. Additionally, the User Name and Phone Number are validated. The costume ID must be entered when the user wishes to choose the costume for return. When the user returns the costume number of quantity of costumes also gets updated by adding the no. of returned costumes to the existing number of quantity of costumes in the costumes file. When the user returns the costumes, the borrower's name and costume ID is asked. Moreover, the maximum number date to borrow the costume is set to 5 days, and if the return date is expired, the fine of $5 is imposed per day and is displayed with a note after the total amount calculation.

```
------------------------------------------------------

        Welcome to costume rental application
            Designed by Himanshu Yadav :)

------------------------------------------------------


Select a desirable option
(1) || Press 1 to rent a costume.
(2) || Press 2 to return a costume.
(3) || Press 3 to exit.
Enter a option: 2


Let's return a costume :)


-------------------------------------------------------------------------------------
S.No.    Costume Name            Brand                   Price          Quantity
-------------------------------------------------------------------------------------
1.       Formal Suits            MegaPlex                $14.5          400
2.       Fairy Costume           DollarSmart             $18            500
3.       Thor Costume            Marvel INC              $23            500
4.       Ant Costume             DC Comics               $25            500
-------------------------------------------------------------------------------------

Enter the costume ID to return: 1
Enter the quantity you wanna return: 100
```

Figure 11: Screenshot of program showing returning process

```
-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-2599
-----------------------------------------------Return Bill-----------------------------------------------
Date:- 2022-8-26                                                        Time:- 0:29:51:126767
Customer Name: Himanshu Yadav
Phone Number: 9852565812
-------------------------------------------------------------------------------------------------------
S.No.           Costume Name            Brand                   Price           Quantity
-------------------------------------------------------------------------------------------------------
1               Formal Suits           MegaPlex               $14.5           100
-------------------------------------------------------------------------------------------------------
                                          Fine: $50
-------------------------------------------------------------------------------------------------------
                                Thank you for vising our store
                                      Visit again :)

------------------------------Bill has also been generated in txt file.--------------------------------
```

Figure 12: Screenshot of program printing Return Invoice in terminal

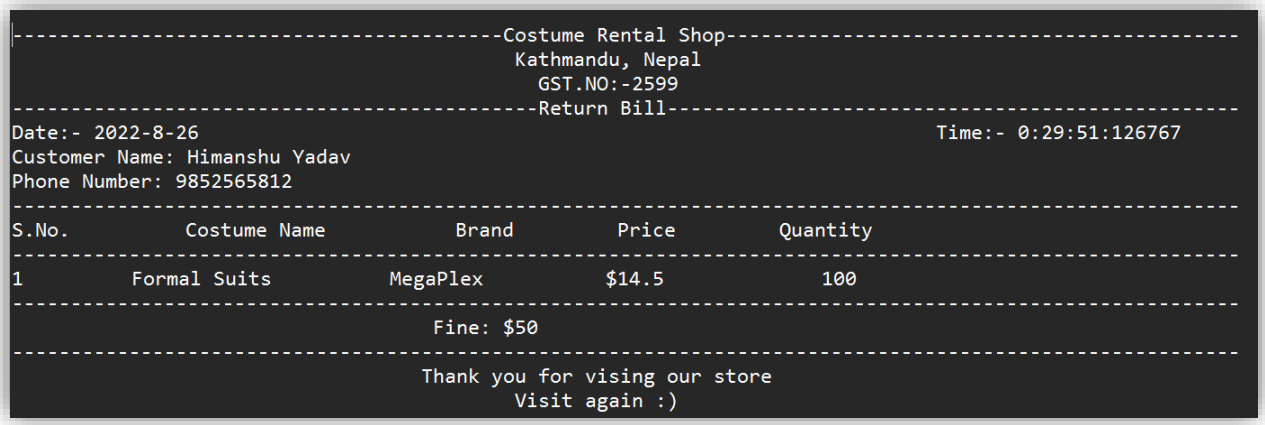


Figure 13: Screenshot of program printing Return Invoice in txt format

4. Testing

A. Test 1

Test No.	1
Objective	To show the implementation of try, except when the user inputs an invalid number
Action	➢ Costume Menu Press 1 to rent a costume. Press 2 to return a costume. Press 3 to exit. ➢ Enter a option: 5 When the user gives 5 as input in Costume Menu
Expected Result	When the user inputs an invalid choice, it will throw an error exception and show an error message.
Actual Result	Invalid input! :(Please select the value as per the provided options :)
Conclusion	The test was successful.

Table 1: Test table of implementation of try-except

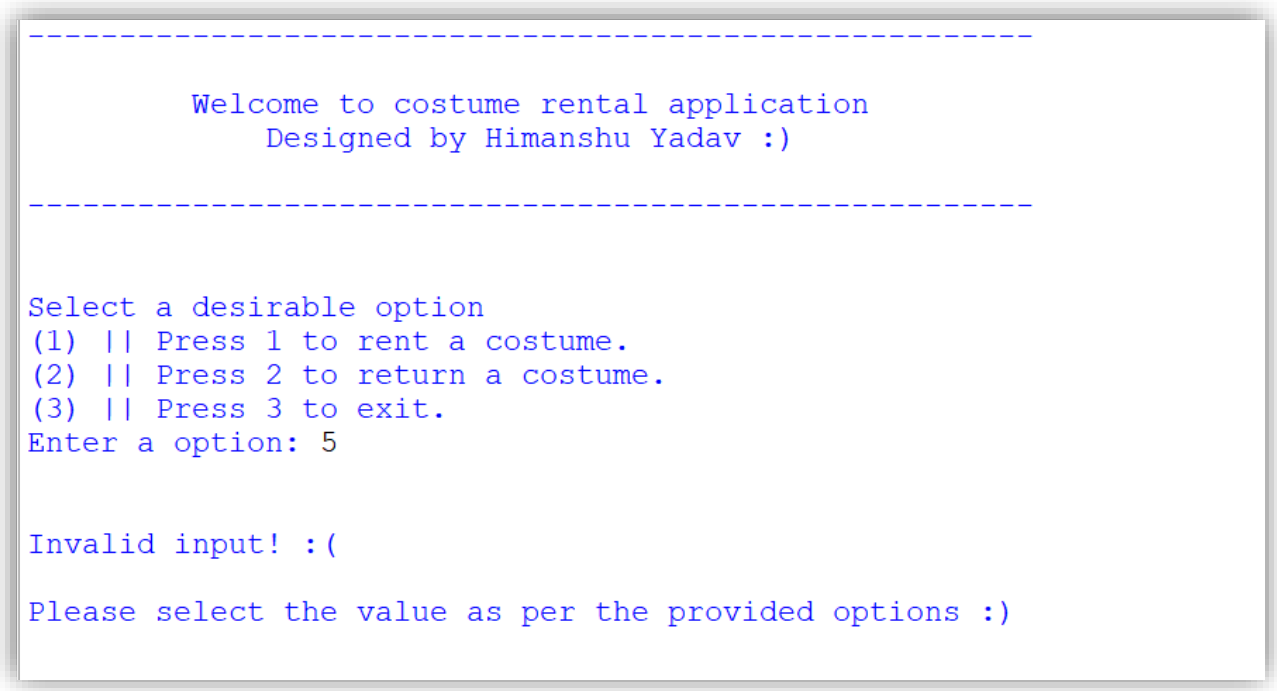


Figure 14: screenshot of implementation of try-except

B. Test 2

Test No.	2
Objective	Provide Negative value as input Provide the non-existed value as input
Action	1. Enter the total number of dresses you want to rent: -100 2. Enter the costume ID to return: 0
Expected Result	1. Please type a number next time. 2. Your number is out of the options provided.
Actual Result	1. Please type a number next time. 2. Your number is out of the options provided.
Conclusion	The test was successful.

Table 2: Test table of rent and return process

```
Let's rent a costume :)


-------------------------------------------------------------------------------
S.No.    Costume Name           Brand                  Price          Quantity
-------------------------------------------------------------------------------
1.       Formal Suits           MegaPlex               $14.5          500
2.       Fairy Costume          DollarSmart            $18            500
3.       Thor Costume           Marvel INC             $23            500
4.       Ant Costume            DC Comics              $25            500
-------------------------------------------------------------------------------

Enter Serial Number: 1
The serial number of the costume is 1.

-----------------------------------------
       The costume is available.
-----------------------------------------


Enter the total number of dresses you want to rent: -100
Please type a number next time.
```

Figure 15: Screenshot of program while providing negative value as input

```
Let's return a costume :)


-------------------------------------------------------------------------------
S.No.    Costume Name           Brand                  Price          Quantity
-------------------------------------------------------------------------------
1.       Formal Suits           MegaPlex               $14.5          500
2.       Fairy Costume          DollarSmart            $18            500
3.       Thor Costume           Marvel INC             $23            500
4.       Ant Costume            DC Comics              $25            500
-------------------------------------------------------------------------------

Enter the costume ID to return: 0
Your number is out of the options provided.
```

Figure 16: Screenshot of program while providing non existed value as input

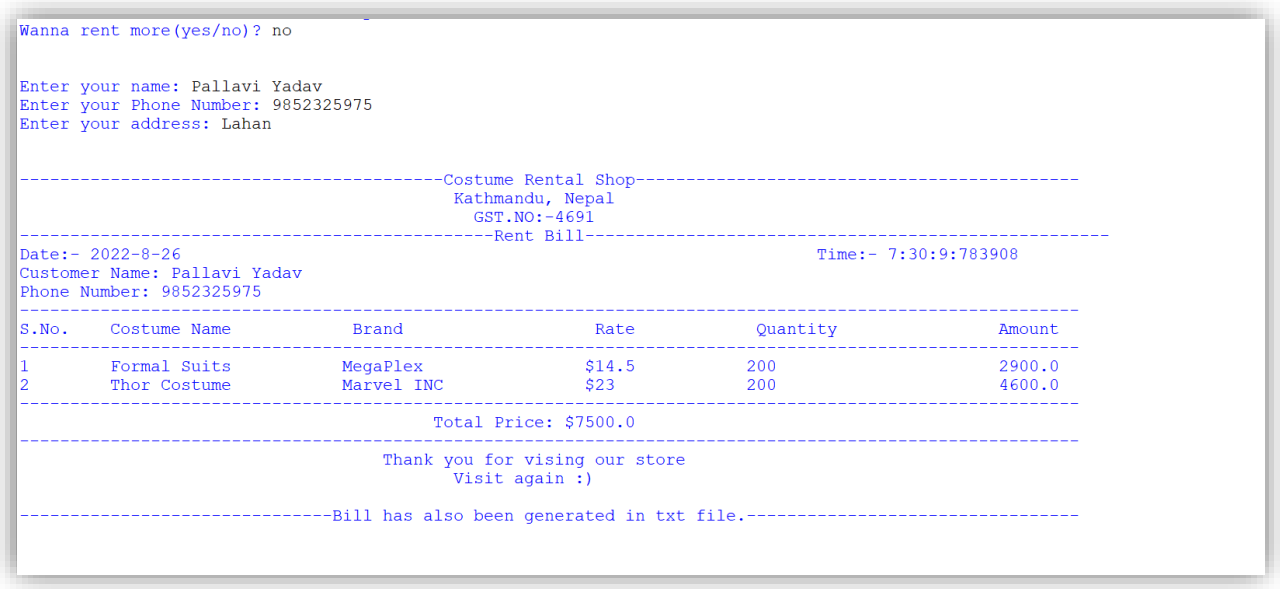
C. Test 3

Test No.	3
Objective	To show file generation of rent
Action	➢ Costume Menu Press 1 to rent a costume. Press 2 to return a costume. Press 3 to exit. ➢ Enter Serial Number: 1 The serial number of the costume is 1. ➢ Enter the total number of dresses you want to rent: 200
Expected Result	The rent-user.txt file will be generated.
Actual Result	The rent-user.txt file is generated.
Conclusion	The test was successful.

Table 3: Test table of file generation of rent process

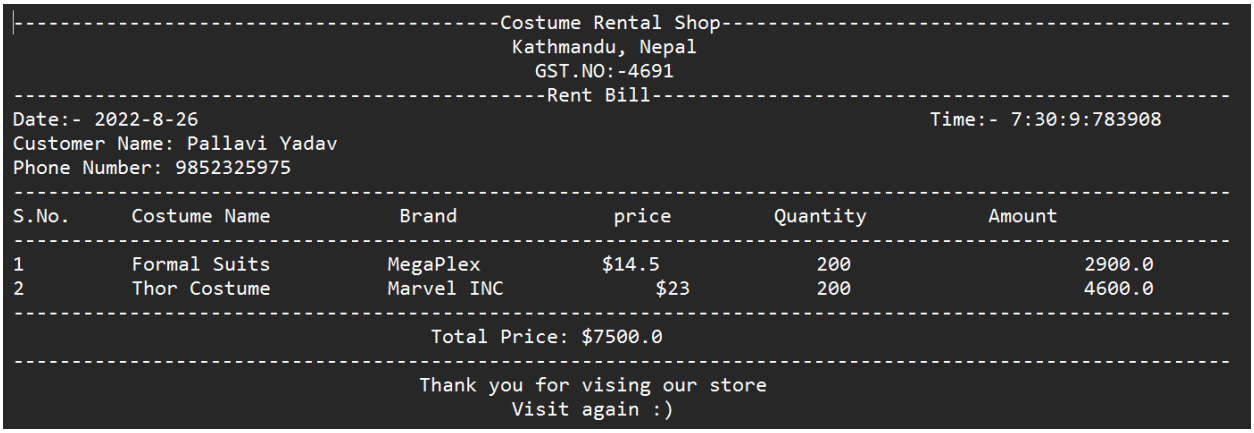
```
Let's rent a costume :)


-------------------------------------------------------------------------------
S.No.    Costume Name           Brand                  Price          Quantity
-------------------------------------------------------------------------------
1.       Formal Suits           MegaPlex               $14.5          500
2.       Fairy Costume          DollarSmart            $18            500
3.       Thor Costume           Marvel INC             $23            500
4.       Ant Costume            DC Comics              $25            500
-------------------------------------------------------------------------------

Enter Serial Number: 1
The serial number of the costume is 1.

-----------------------------------------
      The costume is available.
-----------------------------------------


Enter the total number of dresses you want to rent: 200
Wanna rent more(yes/no)? yes

Your Cart: [['Formal Suits', 200]]
```

Figure 17: Screenshot of program while renting the costume



```
Wanna rent more(yes/no)? no


Enter your name: Pallavi Yadav
Enter your Phone Number: 9852325975
Enter your address: Lahan


-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-4691
-------------------------------------------Rent Bill-------------------------------------------------
Date:- 2022-8-26                                                          Time:- 7:30:9:783908
Customer Name: Pallavi Yadav
Phone Number: 9852325975
-----------------------------------------------------------------------------------------------------
S.No.    Costume Name            Brand                  Rate           Quantity               Amount
-----------------------------------------------------------------------------------------------------
1        Formal Suits           MegaPlex               $14.5          200                    2900.0
2        Thor Costume           Marvel INC             $23            200                    4600.0
-----------------------------------------------------------------------------------------------------
                                    Total Price: $7500.0
-----------------------------------------------------------------------------------------------------
                                Thank you for vising our store
                                       Visit again :)

------------------------------Bill has also been generated in txt file.--------------------------------
```

Figure 18: Screenshot of Invoice of rent in the terminal



```
-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-4691
-------------------------------------------Rent Bill-------------------------------------------------
Date:- 2022-8-26                                                          Time:- 7:30:9:783908
Customer Name: Pallavi Yadav
Phone Number: 9852325975
-----------------------------------------------------------------------------------------------------
S.No.     Costume Name           Brand             price         Quantity          Amount
-----------------------------------------------------------------------------------------------------
1         Formal Suits          MegaPlex          $14.5             200                    2900.0
2         Thor Costume          Marvel INC            $23           200                    4600.0
-----------------------------------------------------------------------------------------------------
                                   Total Price: $7500.0
-----------------------------------------------------------------------------------------------------
                                Thank you for vising our store
                                       Visit again :)
```

Figure 19: Screenshot of Invoice of rent in the txt form

D. Test 4

Test No.	4
Objective	To show file generation of return
Action	➢ Costume Menu Press 1 to rent a costume. Press 2 to return a costume. Press 3 to exit. ➢ Enter Serial Number: 1 The serial number of the costume is 1. ➢ Enter the total number of dresses you want to return: 200
Expected Result	The return-user.txt file will be generated.
Actual Result	The return-user.txt file is generated.
Conclusion	The test was successful.

Table 4: Test table of file generation of return process

```
Let's return a costume :)


-------------------------------------------------------------------------------------
S.No.    Costume Name            Brand                   Price           Quantity
-------------------------------------------------------------------------------------
1.       Formal Suits            MegaPlex                $14.5           300
2.       Fairy Costume           DollarSmart             $18             500
3.       Thor Costume            Marvel INC              $23             300
4.       Ant Costume             DC Comics               $25             500
-------------------------------------------------------------------------------------

Enter the costume ID to return: 1
Enter the quantity you wanna return: 200
Wanna return more(yes/no)? yes
```

Figure 20: Screenshot of file generation of return

```
Please enter your name: Pallavi Yadav
Enter number of Day from rent days: 15
Enter your Phone Number: 985203245
Enter your address: Lahan

-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-3412
-----------------------------------------Return Bill-------------------------------------------------
Date:- 2022-8-26                                                              Time:- 7:35:55:195602
Customer Name: Pallavi Yadav
Phone Number: 985203245
-----------------------------------------------------------------------------------------------------
S.No.           Costume Name           Brand                  Price          Quantity
-----------------------------------------------------------------------------------------------------
1               Formal Suits           MegaPlex               $14.5          200
2               Thor Costume           Marvel INC             $23            200
-----------------------------------------------------------------------------------------------------
                                          Fine: $50
-----------------------------------------------------------------------------------------------------
                                  Thank you for vising our store
                                         Visit again :)

------------------------------Bill has also been generated in txt file.------------------------------
```

Figure 21: Screenshot of Invoice of return in the terminal

```
-----------------------------------------Costume Rental Shop-----------------------------------------
                                        Kathmandu, Nepal
                                          GST.NO:-3412
-----------------------------------------Return Bill-------------------------------------------------
Date:- 2022-8-26                                                              Time:- 7:35:55:195602
Customer Name: Pallavi Yadav
Phone Number: 985203245
-----------------------------------------------------------------------------------------------------
S.No.          Costume Name           Brand         Price         Quantity
-----------------------------------------------------------------------------------------------------
1         Formal Suits          MegaPlex          $14.5             200
2         Thor Costume          Marvel INC             $23          200
-----------------------------------------------------------------------------------------------------
                                    Fine: $50
-----------------------------------------------------------------------------------------------------
                                  Thank you for vising our store
                                         Visit again :)
```

Figure 22: Screenshot of Invoice of return in the txt form

E. Test 5

Test No.	5
Objective	Show the update in costume file 1. Show the quantity being deducted while renting the costumes 2. Show the quantity being added while returning the costumes
Action	Costume Menu Press 1 to rent a costume. Press 2 to return a costume. Press 3 to exit. ➢ Enter Serial Number: 1 The serial number of the costume is 1. ➢ Enter the total number of dresses you want to rent: 200 Costume Menu Press 1 to rent a costume. Press 2 to return a costume. Press 3 to exit. ➢ Enter Serial Number: 1 The serial number of the costume is 1. ➢ Enter the total number of dresses you want to return: 200
Expected Result	The costumes file will be updated.
Actual Result	The costumes file is updated.
Conclusion	The test was successful.

Table 5: Test table of costumes update in txt file

```
Formal Suits,MegaPlex,$14.5,500
Fairy Costume,DollarSmart,$18,500
Thor Costume,Marvel INC,$23,500
Ant Costume,DC Comics,$25,500
```

Figure 23: Screenshot of text file before renting

```
Select a desirable option
(1) || Press 1 to rent a costume.
(2) || Press 2 to return a costume.
(3) || Press 3 to exit.
Enter a option: 1


Let's rent a costume :)


-------------------------------------------------------------------------------------
S.No.    Costume Name            Brand                  Price          Quantity
-------------------------------------------------------------------------------------
1.       Formal Suits            MegaPlex               $14.5          500
2.       Fairy Costume           DollarSmart            $18            500
3.       Thor Costume            Marvel INC             $23            500
4.       Ant Costume             DC Comics              $25            500
-------------------------------------------------------------------------------------

Enter Serial Number: 1
The serial number of the costume is 1.

-----------------------------------------
       The costume is available.
-----------------------------------------


Enter the total number of dresses you want to rent: 200
Wanna rent more(yes/no)? no
```

Figure 24: Screenshot of program while renting the costume

```
Formal Suits,MegaPlex,$14.5,300
Fairy Costume,DollarSmart,$18,500
Thor Costume,Marvel INC,$23,500
Ant Costume,DC Comics,$25,500
```

Figure 25: Screenshot of txt file after renting

```
Formal Suits,MegaPlex,$14.5,300
Fairy Costume,DollarSmart,$18,500
Thor Costume,Marvel INC,$23,500
Ant Costume,DC Comics,$25,500
```

Figure 26: Screenshot of text file before returning

```
Select a desirable option
(1) || Press 1 to rent a costume.
(2) || Press 2 to return a costume.
(3) || Press 3 to exit.
Enter a option: 2


Let's return a costume :)


--------------------------------------------------------------------------------
S.No.    Costume Name            Brand                   Price           Quantity
--------------------------------------------------------------------------------
1.       Formal Suits            MegaPlex                $14.5           300
2.       Fairy Costume           DollarSmart             $18             500
3.       Thor Costume            Marvel INC              $23             500
4.       Ant Costume             DC Comics               $25             500
--------------------------------------------------------------------------------

Enter the costume ID to return: 1
Enter the quantity you wanna return: 200
Wanna return more(yes/no)? no
```

Figure 27: Screenshot of program while returning the costume

```
Formal Suits,MegaPlex,$14.5,500
Fairy Costume,DollarSmart,$18,500
Thor Costume,Marvel INC,$23,500
Ant Costume,DC Comics,$25,500
```

Figure 28: Screenshot of txt file after returning the costume

5. Conclusion

This was the first coursework assigned to us in the second semester. We have learned almost all the basic stuff of python. We have learned different new things after this coursework about that we can develop different projects easily with python-like one of them is developed already on our coursework. Similarly, It has helped me to develop my imagination. Before I didn't know how to use python to develop the system GUI in the console but now, we can think up to that level and learned the huge scope of python in the future. After hard work and research, we can complete this coursework which even helped me to increase my research skills too. Even different test was done to find the bugs and errors in the code developed as it is obvious that there will be some bugs in the program for sure. I have completed this coursework in time after a lot of research and hard work also that to the module leader who always helped me to complete my coursework.

From the overall coursework, I learned different new things about python programming language and also about different new things except for python which may help in another programming language too like how to create a flowchart, pseudocode. Algorithm and many more which may also help in other programming languages and making reports in future work. This coursework was a bit hard than I have thought because It took almost one month to complete the task. Thanks to the module leaders and different teachers of this module who helped me to complete the coursework.

Hence, the experience to complete this coursework was awesome and has helped to develop our imagination and showed the future and wide scope of the python programming language. It has helped me to develop different abilities and skills which may also help in the future.

6. Appendix

Main Module

```
import rentCostume
import returnCostume

def welcome():
    print("-------------------------------------------------------")
    print()
    print("         Welcome to costume rental application")
    print("             Designed by Himanshu Yadav :) ")
    print()
    print("-------------------------------------------------------")


def displayingMessage():


    while True:
        print("\n")
        print("Select a desirable option")
        print("(1) || Press 1 to rent a costume.")
        print("(2) || Press 2 to return a costume.")
        print("(3) || Press 3 to exit.")
        selectedOption = input("Enter a option: ")
        if selectedOption == "1":
            print("\n")
            print("Let's rent a costume :)")
            print("\n")
            rentCostume.rentCostume()

        elif selectedOption == "2":
            print("\n")
            print("Let's return a costume :)")
            print("\n")
            returnCostume.returnCostume()

        elif selectedOption == "3":
            print("\n")
            print("         Greetings and thank you for visiting our store :)")
            exit()

        else:
            print("\n")
```

```
            print("Invalid input! :( ")
            print()
            print("Please select the value as per the provided options :)")

    welcome()
    displayingMessage()
```

Rent Module

```
    import datetime
    import random

    def getFileContent():

        file = open("costumes.txt","r")
        data = file.readlines()
        file.close()
        return data


    def getDictionary(fileContent):

        data = {}
        for index in range(len(fileContent)):
            data[index+1] = fileContent[index].replace("\n","").split(",")
        return data


    def printCostumes(mainData):

        print("-------------------------------------------------------------------------------")
        print("S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t", "Quantity")
        print("-------------------------------------------------------------------------------")
        for key,value in mainData.items():
            print(str(key)+str("."), "\t", value[0], "\t\t", value[1], "\t\t", value[2], "\t\t", value[3])
        print("-------------------------------------------------------------------------------")
        return ""

    def getValidSno(mainData):

        validSno = False
        while validSno == False:
            SNo = input("Enter Serial Number: ")
            try:
```

```
            if SNo.isdigit():
                SNo = int(SNo)
                if SNo > 0 and SNo <= len(mainData):
                    if int(mainData[SNo][3]) == 0:
                        print()
                        print("Costume is out of stock!")
                        print()
                        print("Try another!")
                        print()
                        print(printCostumes(mainData))
                        continue
                    else:
                        validSno = True
                        print(f"The serial number of the costume is {SNo}.")
                        print()
                        print("-----------------------------------------")
                        print("       The costume is available.          ")

                        print("-----------------------------------------")
                        print("\n")
                    return SNo
                else:
                    print("Your number is out of the options provided.")
                    print("\n")
            else:
                print("Please type a number next time.")
                print("\n")
        except:
            print("Invalid Serial Number!")

    def getValidQuantity(mainData,SNo):

        cart = []
        tempRentBill = []
        validQuantity = False

        while validQuantity == False:
            quantity = input("Enter the total number of dresses you want to rent: ")
            try:
                if quantity.isdigit():
                    quantity = int(quantity)
                    if quantity > 0 and quantity <= int(mainData[SNo][3]):
                        validQuantity = True
                        mainData[SNo][3] = str(int(mainData[SNo][3]) - quantity)
                        return quantity
                    else:
```

```
                print("Quantity provided is greater than we have in stock.")
                print("So, Please enter the quantity which doesn't goes exceeding our
stocks.")
                print("\n")
            else:
                print("Please type a number next time.")
                print("\n")
        except:
            print("Invalid Quantity!")

    def rentCostume():

        userWantsClothes = True
        cart = []
        tempRentBill = []

        while userWantsClothes == True:
            print(printCostumes(mainData))
            SNo = getValidSno(mainData)
            quantity = getValidQuantity(mainData,SNo)

            flag = True
            for costume in cart:
                if costume[0] == SNo:
                    costume[1] += quantity
                    flag = False
            if flag:
                cart.append([mainData[SNo][0], quantity])
                tempRentBill.append([mainData[SNo][0], mainData[SNo][1],
mainData[SNo][2], quantity])

            valid_input = False
            while valid_input == False:
                wantAnother = input("Wanna rent more(yes/no)? ")
                if wantAnother.lower() == "yes":
                    print("\n")
                    print(f"Your Cart: {cart}")
                    print("\n")
                    valid_input = True
                    break
                elif wantAnother.lower() == "no":
                    print("\n")
                    generateBill(tempRentBill)
                    userWantsClothes = False
                    valid_input = True
                else:
```

```
                print("Invalid Input !!")
                print("\n")
                continue
            updateTextFile(mainData)
            print("\n")


    def generateBill(tempRentBill):

        validName = False
        while validName == False:
            Name = str(input("Enter your name: "))
            if Name.replace(" ", "").isalpha():
                validName = True

        validPhoneNumber = False
        while validPhoneNumber == False:
            phoneNumber = str(input("Enter your Phone Number: "))
            if phoneNumber.isdigit():
                validPhoneNumber = True


        Address = str(input("Enter your address: "))
        GST = random.randint(1000, 5000)
        Year = (datetime.datetime.now().year)
        Month = (datetime.datetime.now().month)
        Day = (datetime.datetime.now().day)
        Hour = (datetime.datetime.now().hour)
        Minute = (datetime.datetime.now().minute)
        Second = (datetime.datetime.now().second)
        microSecond = (datetime.datetime.now().microsecond)
        print("\n")
        print("-----------------------------------------Costume Rental Shop----------------------------
---------------")
        print("                                  Kathmandu, Nepal
")
        print(f"                                    GST.NO:-{GST}                                    ")
        print("-----------------------------------------------Rent Bill--------------------------------------
----------")
        print(f"Date:- {Year}-{Month}-{Day}                                                    Time:-
{Hour}:{Minute}:{Second}:{microSecond}")
        print(f"Customer Name: {Name}")
        print(f"Phone Number: {phoneNumber}")
        print("-----------------------------------------------------------------------------------------------
-----")
```

```
    print("S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Rate", "\t\t", "Quantity",
"\t\t", "Amount")
    print("-----------------------------------------------------------------------------------------------------
-----")
    row = ""

    counter = 0
    finalPrice = 0
    for i in range(len(tempRentBill)):
        counter += 1
        for j in range(len(tempRentBill[i])):
            dollarprice = float(tempRentBill[i][2].replace("$",""))
            priceDetail = dollarprice * tempRentBill[i][3]
            row = row + str(tempRentBill[i][j]) + "\t\t"
        print(counter,"\t",row, "\t", priceDetail)
        finalPrice = finalPrice + priceDetail
        row = ""

    print("-----------------------------------------------------------------------------------------------------
-----")
    print(f"                                Total Price: ${finalPrice}
")
    print("-----------------------------------------------------------------------------------------------------
-----")
    print("                              Thank you for vising our store
")
    print("                                   Visit again :)                                    ")
    print()
    print("-------------------------------Bill has also been generated in txt file.-----------------
---------------")


    text = f"Rent-{Name}.txt"
    file = open(text,"w")
    file.write("----------------------------------------Costume Rental Shop----------------------
--------------------")
    file.write("\n")
    file.write("                                  Kathmandu, Nepal
")
    file.write("\n")
    file.write(f"                                    GST.NO:-{GST}
")
    file.write("\n")
    file.write("-----------------------------------------------Rent Bill-------------------------------------
------------")
    file.write("\n")
```

```
    file.write(f"Date:- {Year}-{Month}-{Day}
Time:- {Hour}:{Minute}:{Second}:{microSecond}")
    file.write("\n")
    file.write(f"Customer Name: {Name}")
    file.write("\n")
    file.write(f"Phone Number: {phoneNumber}")
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write("S.No. \t Costume Name \t\t Brand \t\t\t price \t\t Quantity \t\t Amount")
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    row = ""

    counter = 0
    finalPrice = 0
    for i in range(len(tempRentBill)):
        counter += 1
        for j in range(len(tempRentBill[i])):
            dollarprice = float(tempRentBill[i][2].replace("$",""))
            priceDetail = dollarprice * tempRentBill[i][3]
            row = row + str(tempRentBill[i][j]) + "\t\t\t"
        file.write(f"{counter} \t\t {row} \t\t{priceDetail}")
        file.write("\n")
        finalPrice = finalPrice + priceDetail
        row = ""

    file.write("-------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write(f"                              Total Price: ${finalPrice}
")
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write("                              Thank you for vising our store
")
    file.write("\n")
    file.write("                                   Visit again :)                                  ")
    file.close()
```

```
def updateTextFile(mainData):

    file = open("costumes.txt", "w")
    for value in mainData.values():
        file.write(str(value[0]) + "," + str(value[1]) + "," + str(value[2]) + "," + str(value[3])
+ "\n")
    file.close()

fileContent = getFileContent()
mainData = getDictionary(fileContent)
```

Return Module

```
import datetime
import random

def getFileContent():

    file = open("costumes.txt","r")
    data = file.readlines()
    file.close()
    return data


def getDictionary(fileContent):

    data = {}
    for index in range(len(fileContent)):
        data[index+1] = fileContent[index].replace("\n","").split(",")
    return data


def printCostumes(mainData):

    print("-------------------------------------------------------------------------------------")
    print("S.No.", "\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t", "Quantity")
    print("-------------------------------------------------------------------------------------")
    for key,value in mainData.items():
        print(str(key)+str("."), "\t", value[0], "\t\t", value[1], "\t\t", value[2], "\t\t", value[3])
    print("-------------------------------------------------------------------------------------")
    return ""

def getValidID(mainData):

    validId = False
    while validId == False:
```

```
        ID = input("Enter the costume ID to return: ")
        if ID.isdigit():
            ID = int(ID)
            if ID > 0 and ID <= len(mainData):
                validID = True
                return ID
                break
            else:
                print("Your number is out of the options provided.")
                print("\n")
        else:
            print("Please type a number next time.")
            print("\n")



    def getValidReturnQuantity(mainData,ID):

        returnCart = []

        validQuad = False
        while validQuad == False:
            quantity = input("Enter the quantity you wanna return: ")
            if quantity.isdigit():
                quantity = int(quantity)
                validQuad = True
                mainData[ID][3] = str(int(mainData[ID][3]) + quantity)
                returnCart.append([mainData[ID][0], mainData[ID][1], mainData[ID][3]])
                return quantity
            else:
                print("Please enter a number not anything else!")
                print("\n")


    def returnCostume():

        userReturnsClothes = True
        returnCart = []

        while userReturnsClothes == True:
            print(printCostumes(mainData))
            ID = getValidID(mainData)
            quantity = getValidReturnQuantity(mainData,ID)

            flag = True
            for costume in returnCart:
```

```
            if costume[0] == ID:
                costume[1] += quantity
                flag = False
        if flag:
            returnCart.append([mainData[ID][0], mainData[ID][1], mainData[ID][2],
quantity])

        valid_input = False
        while valid_input == False:
            returnAnother = input("Wanna return more(yes/no)? ")
            if returnAnother.lower() == "yes":
                print("\n")
                valid_input = True
                break
            elif returnAnother.lower() == "no":
                print("\n")
                generateReturnBill(returnCart)
                userReturnsClothes = False
                valid_input = True
            else:
                print("Please enter a option from given options only!")
                print("\n")
                continue
        updateTextFile(mainData)
        print("\n")

    def generateReturnBill(returnCart):

        validName = False
        while validName == False:
            Name = str(input("Please enter your name: "))
            if Name.replace(" ", "").isalpha():
                validName = True
            else:
                print("Sorry, You mistakely typed your name wrong!")
                print("\n")

        validDay = False
        while validDay == False:
            day = input("Enter number of Day from rent days: ")
            if day.isdigit():
                day = int(day)
                validDay = True
            else:
                print("Please enter the days which is always in number!")
                print("\n")
```

```
    validPhoneNumber = False
    while validPhoneNumber == False:
        phoneNumber = str(input("Enter your Phone Number: "))
        if phoneNumber.isdigit():
            validPhoneNumber = True
        else:
            print("Sorry, You mistakely typed your Phone Number wrong!")
            print("\n")

    Address = str(input("Enter your address: "))
    GST = random.randint(1000, 5000)
    Year = (datetime.datetime.now().year)
    Month = (datetime.datetime.now().month)
    Day = (datetime.datetime.now().day)
    Hour = (datetime.datetime.now().hour)
    Minute = (datetime.datetime.now().minute)
    Second = (datetime.datetime.now().second)
    microSecond = (datetime.datetime.now().microsecond)
    print("\n")
    print("-----------------------------------------Costume Rental Shop----------------------------
---------------")
    print("                                  Kathmandu, Nepal
")
    print(f"                                    GST.NO:-{GST}                                    ")
    print("-----------------------------------------------Return Bill--------------------------------
---------")
    print(f"Date:- {Year}-{Month}-{Day}                                                    Time:-
{Hour}:{Minute}:{Second}:{microSecond}")
    print(f"Customer Name: {Name}")
    print(f"Phone Number: {phoneNumber}")
    print("-------------------------------------------------------------------------------------------------
-----")
    print("S.No.", "\t\t", "Costume Name", "\t\t", "Brand", "\t\t\t", "Price", "\t\t",
"Quantity")
    print("-------------------------------------------------------------------------------------------------
-----")
    row = ""

    counter = 0

    if day > 5:
        newDay = day - 5
        fine = newDay*5
    else:
        fine = 0
```

```
    for i in range(len(returnCart)):
        counter += 1
        for j in range(len(returnCart[i])):
            row = row + str(returnCart[i][j]) + "\t\t"
        print(counter,"\t\t",row)
        row = ""

    print("-------------------------------------------------------------------------------------------------
-----")
    print(f"                                    Fine: ${fine}                                    ")
    print("-------------------------------------------------------------------------------------------------
-----")
    print("                                Thank you for vising our store
")
    print("                                    Visit again :)                                    ")
    print()
    print("-------------------------------Bill has also been generated in txt file.-----------------
--------------")


    text = f"Return-{Name}.txt"
    file = open(text,"w")
    file.write("-----------------------------------------Costume Rental Shop----------------------
---------------------")
    file.write("\n")
    file.write("                                    Kathmandu, Nepal
")
    file.write("\n")
    file.write(f"                                      GST.NO:-{GST}
")
    file.write("\n")
    file.write("-----------------------------------------Return Bill-----------------------------------
-------------")
    file.write("\n")
    file.write(f"Date:- {Year}-{Month}-{Day}
Time:- {Hour}:{Minute}:{Second}:{microSecond}")
    file.write("\n")
    file.write(f"Customer Name: {Name}")
    file.write("\n")
    file.write(f"Phone Number: {phoneNumber}")
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write("S.No. \t\t Costume Name \t\t Brand \t\t Price \t\t Quantity")
```

```
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    row = ""

    counter = 0
    finalPrice = 0
    for i in range(len(returnCart)):
        counter += 1
        for j in range(len(returnCart[i])):
            row = row + str(returnCart[i][j]) + "\t\t\t"
        file.write(f"{counter} \t\t {row}")
        file.write("\n")
        row = ""

    file.write("-------------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write(f"                              Fine: ${fine}                                ")
    file.write("\n")
    file.write("-------------------------------------------------------------------------------------------------
----------")
    file.write("\n")
    file.write("                            Thank you for vising our store
")
    file.write("\n")
    file.write("                                  Visit again :)                                 ")
    file.close()

def updateTextFile(mainData):

    file = open("costumes.txt", "w")
    for value in mainData.values():
        file.write(str(value[0]) + "," + str(value[1]) + "," + str(value[2]) + "," + str(value[3])
+ "\n")
    file.close()

fileContent = getFileContent()
mainData = getDictionary(fileContent)
```

Bibliography

Entrepreneur, 2022. *Entrepreneur.* [Online]
Available at: https://www.entrepreneur.com/businessideas/costume-rentals
[Accessed 15 8 2022].
TRUic, 2017. *HowToStartAnLLC.com.* [Online]
Available at: https://howtostartanllc.com/business-ideas/costume-rental#:~:text=A%20costume%20business%20offers%20its,the%20needs%20of%20their%20customers.
[Accessed 15 8 2022].
W3schools.com, 2020. *W3schools.com.* [Online]
Available at: https://www.w3schools.com/python/python_dictionaries.asp
[Accessed 15 8 2022].