Fundamentals of computing

60% Individual Coursework

2023 spring

Student Name: Apil thapa

London Met ID: 22067753

College ID: NP01cp4a220164

Assignment Due Date: Sunday, March 19, 2023

Assignment Submission Date: Friday, May 12, 2023

Word Count: 242

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

Table of Contents

1. Introduction:

Simply, a laptop shop is a place where customers can purchase laptops and other computer accessories, with the increasing demand of laptop in our market especially when we are kepping track of quantity stcoks it is a difficult task so we are creating a python program for faster access and more effiecnt work to make customize designed software we are creating this software using python programming language. This is a python project in which we have to create a laptop shop which can purchased and sell laptops. The laptop shop can be created with the help of our own implementation, program using tab spaces not by using.format also.i have learned a lot while doing this project. This project shows me the whole structure of program how python helps to develop such type of project.python has become more popular language in today's field whether it has been used in different fields such as industry, data science where python plays big role to structured data. By doing this coursework I have learned about dictionary, list how can we use the key and values to extract data? This all concept have provided me better understanding for python. Without using libraries like pandas how can we split those textfile data into table format? We have learned a lot of things positive things with this project which can helps us build more extra base in python programming language. This project can be done more easily by the help of pandas for displaying and .format in order to create more easy work in displaying data.later we can create more shops like this for grocery store or something else .we can create program for others too. This laptop shop project includes only command line code if we can add graphical user interface into this we can do better in for this code.overall this coursework or laptop shop project helps us to build logic and understanding about data structures which also includes algorithm and flowcharts and how to implement this logics. We have tried our best to make our project more interactive with users which won’t exit if user don’t want to exit from program.

1.1. Main goals of our project.

The primary goal of using python in a laptop shop is to streamline its operations and increase its efficency. By developing customized software, bussiness can automate various processes,such as inventory management , sales tracking and customer data management .its play main role in bussinesss and customer data management to keep tracks of varoius data using this python programming to create software Our main goals is to create a laptop shop using python programming language without help of additional libraries such as pandas and extra formats.laptop shop project defines that how we can purchased those laptops which are available in out stocks but how can we add those available laptops in out stock and how we can sell those laptops too which should decrease our quantity from stocks. We can use our own logic to add more like we can add more laptops to our stock and we can search for our laptops too and additional many more things. We are creating the store which sells some laptop from stocks which decreases quantity after buying also purchasing stocks which add more quantity in out stocks we are generating invoice,bill to the customer for purchasing items we are generating invoice with vat and gross amount whereas for selling items we are asking customers either they wants with shipping cost or not if they wants with shipping cost then invoice,bill with shipping cost added should be generated else bill with non shipping cost is generated.our main goals is to create a laptop shop and which helps user to buy laptops according to their need. We have done our best so that user won't face any problems while purchasing or selling laptops. We have generated invoices according to user’s transaction details.

1.2. Objectives for that project.

We are main focusing to develop a software system to manage inventory, by using python it helps to manage the system for tracking quantity and stocks. We are tracking for the sales how much we have sales and how much quantity is taken by our customer, for purchasing those laptops we are working such as for updating quantity, from the stocks and generating invoice for the purchased items including vat amount and etc. firstly, we are asking everything for the user either they wants to buy more laptops or not if they wants to buy more laptops then program will run go and generate bill, invoice as per customer needs it will execute on the basis of user requirement if they want to execute it will run else it will exit if user wants it to.At first, Developing a software system to manage inventory While creating our python shops project we have readed the textfile which has details of laptops lists modify those details and append it to dictionary on which we are extracting those details with the help of keys and values.first we are displaying those details on console to the user and gives an option for the user if he wants to sell or buy laptops.if he wants to sell program will ask for the name.phone number and I'd which is used as keys to extract details of the laptops.and checks all stocks if available then it will further procced to give choice for the user it he wants to purchase more if not then program will goes on generating an invoice which includes all vat amount gross amount and net amount on the product.the program implements the proper try, except to catch if any types of exception occurs during the execution of the program so that users won't faced any difficulties while.having transaction. Same goes for sell also program will ask the user for their personal details which I'd he wants to access laptop details and program will ask for the user if he wants to sell more laptops or not if yes then program will runs in a loop but if not then it will ask for the user's if he wants to add shipping cost to his item or not if yes then invoice with shipping cost added is generated else shipping cost is not added to invoice.the program will run till until user wants to exit from it otherwise it will runs it won't break from middle of the program simply if user wants it to exit then only it will exit . This steps or in this way we are able to achieve the goals what we have decided to do for.

2. Discussion and analysis.

2.1. Algorithm:

Step by step representation of our program is known as algorithm. As we are developing something extra we need to create an algorithm first to structurised how it looks and to which flow it goes on. Many people who develops different inventions wrote their algorithm first to know at least how my program looks like.

Here is an algorithm for our coursework to create laptop shop's program.

Step-1. Start.

Step-2 Initialize the dictionary named product dict which store details.

Step-3. Read the text file that contains all the information about the available laptops and append it to dictionary.

Step-4. Display menus of user's choice.

Step-5.Ask user if they wants to (1. buy, 2.sell or 3.exit).

Step-6 .if user selects ‘1’, Asks for the user for their personal details (such as name and

Their phone Numbers.)

Step-7. Asks the user which number of laptop id he wants to buy.

Step-8 ask user what number of quantity of laptop he needs.

Step-9 If a user provide valid details such as product id of laptop available in our stock

Then only program will continue otherwise, it runs in a loop which ask for users

Details again until they are valid.

Step-10 If quantity is available in our stock then only it will continue otherwise it runs in a loop ask for quantity again until

Valid.

Step-11 Calculates the vat amount (13% of total amount) and gross amount (amount

Without any vat) and net amount.

Step-12. It will update the quantity of orderd laptops in dictionary.

Step-13 Create a new list of items that the users has ordered and append it to a new list. Which also displays the list to the user with full details of each laptops, prices and quantity.

Step-14. The program gives an option for the user to buy more laptops or not.

Step-15. If the user wants to buy more laptops they just need to click ‘y’ which will return to step 7 again otherwise, it will

Break the while loop.

Step-16 Generate an invoice/receipt of the laptops that the user bought and write it to the unique textfile. It also prints the invoice on the users screen/console.

Step-17 when program will break then it will return to step 5 in which user has three

Choices either he wants to buy, sell or exit.

Step-18. If user choose option ‘2’.

Step-19. Asks for the customer for their personal details (such as name and their phone Numbers.)

Step-20. If the customer wants to sell the laptops, the program asks for the product ID, which should be greater than zero

And less than length of dictionary.

Step-21 Ask for the customer in what number of quantity he needs.

Step-22. If the customer enters valid product id and user quantity which is in our stock, the program continues, otherwise

it Runs in a Loop until it is valid.

Step-23. Update the quantity of the laptops sold and displayed it on the user’s screen.

Step-24 create a new list of items that the customer has sold and append it to a new list.

It also displays the list to the user with details of each laptops.

Step-25. Asks for the customer, if he wants to add shipping cost to his laptop or not.

Step-26. If customer permit to add shipping cost, Invoice/receipt with shipping cost added and writes to a unique textfile.

Step-27 it generates an invoice/receipt without shipping cost added and

Writes to a unique textfile.

Step-28 it gives a choice for customer if he wants to sell any other laptops or not if yes then it

Will return to Step 20.

Step-29 otherwise, it will return to the step 5.

Step -30 if user choose option ‘3’ then,

Step-31. The program will Exit / Close.

Step-32. The program implements the proper use of try/except. So that user won’t face any exception during execution of the program.

2.2. Flowchart:

A flowchart can be defined as a pictorial reperesentation of an algorithm. As we have written already an algorithm we represent it to a flowchart how it looks like simply which is flowchart.

Here in flowchart diagram use of many shapes like input/output box, process box to display to read input and outputs etc.to link from one page to another also one part to another part of the diagram use of on page reference and off reference is implemented in our flowchart not to make our charts lengthy and more thinner when user saw it.

Here’s is flowchart diagram for representation of Laptop’s shop program.

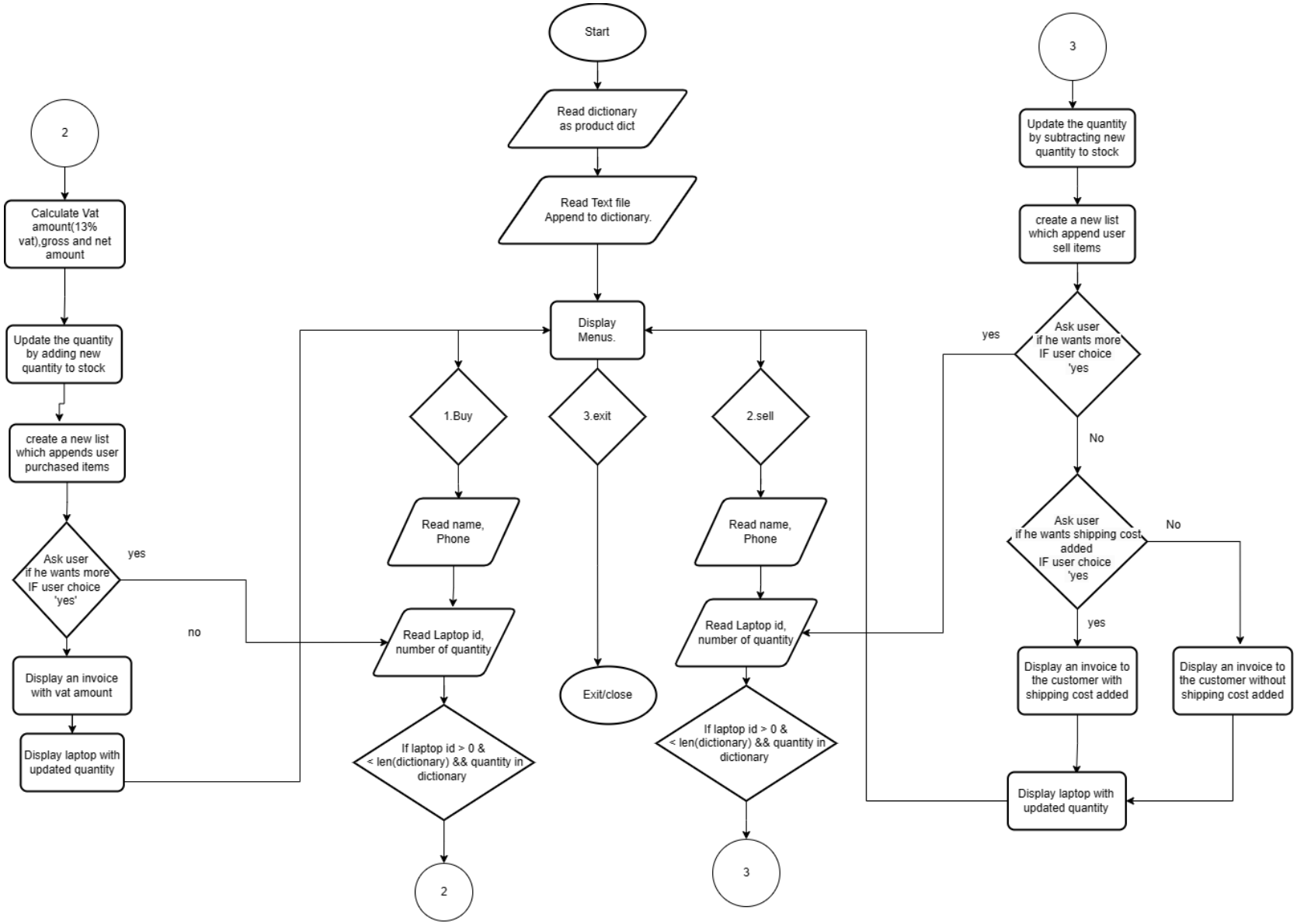

Figure 1 Flowchart

2.3. Pseudocode:

2.3.1 for read.py module.

Initialize the list named as **user requirement.**

Initialize the list named as **user need.**

Initialize the dictionary named as product dict.

Define function display products with no parameter

Initialize id to one.

Open textfile named ‘product info’ with read mode.

Iterate over each line of the file starting from one with enumerate function.

Remove whitespaces and store it in variable named line1.

Store in dictionary with key named id and values splitted with comma

Display new line twice on the screen.

Display named ‘brandname’,’laptopname’,’price’,’quantity’ and’processor’ to the user with Proper Tabspaces on the screen.

Iterate over dictionary dot items with keys and values.

Display key in string format with each values over dictionary by taking length of ‘brandname’,’laptopname’,’price’,’quantity’ and’processor’ and subtracting to the Values using indexing present in Dictionary.

2.3.2 for write.py module.

Import the date and time module.

From read.py module import dictionary named product dict with two list user need and user requirement.

Define function bill generate with parameter as (name, phone, now and user need).

Initialize a varaible named to ‘lucky buyer’.

Get the current date and time using a built in function.

Convert the year, month, day, hour, minute and second to strings.

Concatenate the year, month, day, hour, minute and seconds strings to form a single string

Representation of current date and time.

Assign the concatenated string to a variable name interesting time.

Set the total price to zero.

Display name and titile of shop.

Display current date using built in library.

Display name of buyer.

Display phone number of the buyer.

Set the total pric to zero.

FOR EACH product in user need list:

Get the name of laptop and store it in variable as ‘laptop name’.

Get the quantity of the laptop and store it into variable named ‘laptop quant’.

Get the price of the laptop and store it into variable named ‘total price’.

Get the individual price of the laptop and store it into variable named ‘total Fullamt’.

Get the total price of each laptop which user has buyed and store it into variable Named ’total fullamt’.

Calculate the net amount by multiplying and price of each laptop and store it into Variable named ‘net amount’.

Calculate the vat amount by multiplying net amount * 0.13 percent and store it Into Variable named ‘vat amount’.

Add the total fullamt to the ‘total pric’ variable converted to an integer.

Display name of laptop user has selected converted to an string.

Display number of quantity user has selected converted to an string.

END FOR

Display net amount.

Display vat amount.

Display gross amount.

Display total price for the user.

Concatenate name, underscore, status, and interesting time with ‘.txt’ and store it into variable Named add.

Open a file named ‘add’ in write mode, and assign it to a variable ‘bill’.

Write a horizontal line seperator to a file.

Write the title of the bill.

Write the current date in the format as ‘%y-%m-%d %h-%m-%s’.

Write the name of buyer converted to a string.

Write the phone number of the buyer converted to an int again to string.

FOR EACH product in user need list:

    Write a name of the laptop to a file.

    Write a number of quantity to a file.

END FOR

Write a horizontal line seperator to a file.

Write a horizontal line seperator to a file.

Write the net amount to the file.

Write the gross amount to the file.

Write the vat amount to the file.

Write the total price to the file, with a label.

Write a horizontal line seperator to a file.

Close the file.

Display ‘your details of the shop’ with proper label.

Display "\tbrandname |\t\tlaptopname |\t\t price |\t\tquantity |\t processor |\t graphics".

FOR EACH key and value in product dict dot items:

Display horizontal line seperator.

Display each updated products in table format with key and values.

Display horizontal line seperator.

Display new line seperator.

END FOR.

2.3.3 for operations.py module

From write.py module import function bill generate and shippbill generate.

Import date and time built in library.

Define function named buy products with parameter as (product dict, now and user need list).

Repeat until valid name is entered.

    Ask user to enter their name.

    IF name is not empty:

        If name contains only alphabet:

            Exit the loop

        OTHERWISE:

            Display ‘name can’t be empty’.

        END IF

    END IF

Repeat until valid phone is entered:

    Try:

        Ask user to enter phone number.

Exit the loop

Except:

Display ‘please enter valid information’.

Initalize variable loop to true:

While loop is equals to true:

Repeat until valid product id entered.

Try:

Ask for the product id converted to an integer.

Exit the loop.

Except:

Display ‘please enter valid information’.

While product id converted to an integer less than or equals to zero or greater than length of a dictionary:

Display ‘please enter valid product id’.

Ask for the product id converted to an integer.

Repeat until valid quantity is entered by user:

Try:

Ask for the user quantity converted to an integer.

Exit the loop.

Except:

Display ‘please enter integer values’.

Update the quantity in the dictionary by adding quantity which is in the stock converted to an integer with user selected quantity.

Get the value of product name from a dictionary based on the product id.

Retrieve thr dictionary entry for the given product id.

Get the product name from the entry by assuming it is stored in index 1.

ASSIGN the product name to the variable ‘product name’.

Convert the user’s selected quantity to an integer and assign it to variable quantity selcted.

Calculate the total price of the product by multiplying quantity user has selected converted to an integer with price of each laptop converted to an integer.

Product name, quantity selected, each price and total price is putted in a list and store it in a variable called full item.

Append full item to a list called user need.

Display line seperator to the screen.

Display message’user buyed item list’.

Display ""+"Laptopname"+" "+" quantity"+" "+"Individal price"+""+" total price" with proper formatting using tab spaces.

FOR EACH item in user need list:

    Display each user buyed list with proper spacing and formatting using tabspaces.

    Display line seperator to the screen.

END FOR

Ask user if he wants to buy more laptops or not and store it into variable named choice.

IF choice in lower is equals to ‘n’:

Loop variable is set to false which will break while loop.

Calling bill generate function from write.py module with parameter as (name, phone, now and user need list.)

Return variable name and phone number.

Define function named sell products with parameter as (product dict, now and user requirement list):

    Repeat until valid name is entered.

Ask user to enter their name.

IF name is not **empty**:

If name contains only alphabet:

Exit the loop

END IF

OTHERWISE:

Display ‘name can’t be empty’.

END OTHERWISE

END IF

Repeat until valid phone is entered:

Try:

Ask user to enter phone number.

Exit the loop

Except:

Display ‘please enter valid information’.

Initialize variable loop to true:

While loop is equals to true:

Repeat until valid product id entered.

Try:

Ask for the product id converted to an integer.

Exit the loop.

Except:

Display ‘please enter valid information’.

While product id converted to an integer less than or equals to zero or greater than length of a dictionary:

Display ‘please enter valid product id’.

Ask for the product id converted to an integer.

Repeat until valid quantity is entered by user:

Try:

Ask for the user quantity converted to an integer.

Exit the loop.

Except:

Display ‘please enter integer values’.

Update the quantity in the dictionary by subtracting the quantity which is in the stock converted to an integer with user selected quantity.

Get the value of product name from a dictionary based on the product id.

Retrieve thr dictionary entry for the given product id.

Get the product name from the entry by assuming it is stored in index 1.

ASSIGN the product name to the variable ‘product name’.

Convert the user’s selected quantity to an integer and assign it to variable quantity selcted.

Calculate the total price of the product by multiplying quantity user has selected converted to an integer with price of each laptop converted to an integer.

Product name, quantity selected, each price and total price is putted in a list and store it in a variable called full item.

Append full item to a list called user requirement.

Display line seperator to the screen.

Display message’user selled item list’.

Display ""+"Laptopname"+" "+" quantity"+" "+"Individal price"+""+" total price" with proper formatting using tab spaces.

FOR EACH item in user need list:

> **Display** each user buyed list with proper spacing and formatting using tabspaces.
>
> **Display** line seperator to the screen.

END FOR

Ask user if he wants to buy more laptops or not and store it into variable named choice.

IF choice in lower is equals to ‘n’:

Loop variable is set to false which will break out while loop.

Calling shippbill generate function from write.py module to generate invoice after user selled item with patrameter as (name, phone now and user requirement list.)

Return variable as name and phone number.

END IF.

2.3.4 for main.py module

From read.py module Import product dict user need and user requirement list.

From operations.py module import buy products function.

From write.py module import bill generate function.

Import read module.

Import write module.

Get the current date and time using built in Library called datetime store it in variable now.

Display titile of the shop.

Display new line seperator with tab spaces.

While true:

Show user menus as (buy, sell and exit).

Display name of the shop with welcome message.

Give a choice to user either wants to buy, sell or exit.

Display new line seperator to the console.

Ask user to enter either they wants to buy, sell or exit and store it into variable named user value.

IF user value present in user menus then:

IF user value equals to ‘1’:

Calling display products functions with read.py module.

Display docstring of read.py module.

Calling buy products function with parameter as (product dict, now and userneed list) from operations .py module and store it into two variable named name and phone.

Display docstring of operations module.

END IF

ELIF user value equals to ‘2’:

Calling the display products function with read module.

Call sell products function with paramter as (product dict, now and user requirement) list with Operations.py module.

Display the docstrings for operations module.

END ELIF

OTHERWISE

Exit the program using **exit ()** fucntion.

OTHERWISE:

Display new line sepearator to the screen.

Display message ‘option not available’.

END IF.

2.4. Data structures.

Data structures are the way of managing and arranging data in theprogram they can be used to mnage data in diiferent ways such as accessing, adding, deleting and searching of the data.data structures are said to be containers of data types that store data in a particular way.

2.4.1 Dictionary:

A dictionary is a type of data structures which has two pairs key and values.we can access values from key in the Dictionary.it is represented by {}. To create our own dictionary type a key, followed by a colon. Followed by the Key's value.use commas to separate key value pairs and surround thw whole thing with curly bracket.the most Simple thing which we can do is acessing values of dictionary. We use key to get its values.we just need to put the Keys in the bracket after dictionary name. In our this course work we have implemented keys and values conceptin Dictionary to obtain laptop details using its id as a keys. We have used laptop id as keys so that we can get values Such as laptop details after giving id which is better and easier things/way to solve given problem.dictionary is one Of the best way to use because it is easy to use firstly then , values can be access through keys which is a good benefit for us.we have implemented dictionary in our coursework and used laptop id as a key to accesss whole details of laptops.

read.py - C:\Users\apil xetri\OneDrive\Desktop\Coursework(220167755)foc\read.py (

File Edit Format Run Options Window Help

```
user_requirement=[]
user_need=[]
product_dict={}
def display_products():

    '''This program reads the textfile and display the qua
    returns empty and does not contain any parameter or ar
```

Figure 2 dictionary used

2.4.2 Tuples.

Tuples are the data structure which is simple and unchangeable .tuples are generally denoted by () which are generally seperated by commas.They are the types of sequences like strings.but unlike strings tuples can contains elements of type.using tuples we can proceed our coursework also but we havent implemented tuples because for beginners it can be dfficult for us to understood it.we can create tuples for both strings and numbers values. list is seen as same as tuples but ne difference between lists and tuples can be tuples are unmutable wehereas lists can be mutuable.same as list we can use different built int function with tuples too such as we can use len function to calculate length of tuples, whereas we can also use in operator with tuples too indexing tuples works with indexing strings x,we can use slicing techniques with tuples too.like strings tuples are also immutable which we cant change it.we can concatenate tuples easily like lists. Tuples is also good and easier data structures but one negative point about tuples is that we need to convert it to list or something else to insert or to bring change in it which is the reason behind less use of tuples.

2.4.3 Lists.

Lists are the collection of values which conatin data.generally list are denoted by square brackets and each elements inside list are seperated by commas.we have used list in our coursework to store data of user details.which later on we have read those list and access the values.list are two types single dimensional and two dimensional. In our coursework lists are used as to store details of laptop which user has purchased as well as user hass selled also to generate invoice we have used list. First we read those list we displayed those list using for loop and all. Lists can be one dimensiuonal as well as two dimesional we are using two dimensional list in our coursework. We can use different functions for different type of operation in list we can use len functional to calculate how many elements are there in list, also we can access elemenets presenet in list through indexing of the list.we can slice those lists and get the output what we wants.we can concatenate lists .lists are generally mutuable list is seen as same as tuples but ne difference between lists and tuples can be tuples are unmutable wehereas lists can be mutuable. We can assign new elements to the list through indexing also.we can assiign new list with slicing too.Deleting in a list can also be possible .we can do those things in a list we can also use different list methods such as append, sort, reverse, count etc, so lists is a one of the important and more usuable data structures in python programming language.

read.py - C:\Users\apil xetri\OneDrive\Desktop\Coursework(220167755)foc\read.py (

File Edit Format Run Options Window Help

```
user_requirement=[]
user_need=[]
product_dict={}
def display_products():

    '''This program reads the textfile and display the qua
    returns empty and does not contain any parameter or ar
```

Figure 3 list used

2.4.4 Sets.

Sets are also types of data structures. They are generally unordered. They are generally created using curly braces and they are the set of unique elements only.it used different method like add methods you can add elements to a set using the add function. Removing elements from a set you can remove elements froma set using remove function.sets also supports various mathmatical operations like addition, subtraction, division etc.

2.4.5 Strings

Strings are also data structures because they are the sequence of characters that are immutable and which are created using single or double quotes.they are immutable which means that once created they cannot be changed but we can perform different operations to manipulate and extract information from them.

3. Program

The program starts with initializing new dictionary to which we are storing our shop details.first of all we read textfile and store it into dictionary. we are assigning laptop id as a key and accessing details of the laptop by assumming as values. First we give a choice to the user from the user menus. If user choose ‘1’ the program will go for the buyer.

Program will ask for the users to enter their name and phone number after that it will ask for the laptop id, if laptop id is valid (less than or equals to zero or greater than length of dictionary) it will ask for user quantity if user quantity is also valid or available in our stock then it will it will display list of user purchased item and ask for the user if they want to buy more laptops if they click ‘y’ it will run in a loop otherwise it will calculate vat amount, gross amount, and net amount and Generate an invoice / receipt with vat, gross and net amount included in both console and textfile.

Again if user choice is ‘2’ then the program will go for the seller. Program will ask for the users to enter their name and phone number after that it will ask for the laptop id, if laptop id is valid (less than or equals to zero or greater than length of dictionary) it will ask for user quantity if user quantity is also valid or available in our stock then user selled item list is displayed also program will choices for the customer if they wants to sell more laptop or not if yes then program will run in a loop otherwise it will ask again user if they wants to add shipping cost to their item or not again if yes then invoice with shipping cost added will be generated in console as well as textfile otherwise shipping cost is not added to an invoice / bill.simply if user wants to exit from the program he will need to choose option 3.in another way we can illustrate thst

Developing a software system to manage inventory While creating our python shops project we have readed the textfile which has details of laptops lists modify those details and append it to dictionary on which we are extracting those details with the help of keys and values.first we are displaying those details on console to the user and gives an option for the user if he wants to sell or buy laptops.if he wants to sell program will ask for the name.phone number and I'd which is used as keys to extract details of the laptops.and checks all stocks if available then it will further procced to give choice for the user it he wants to purchase more if not then program will goes on generating an invoice which includes all vat amount gross amount and net amount on the product.the program implements the proper try, except to catch if any types of exception occurs during the execution of the program so that users won't faced any difficulties while.having transaction. Same goes for sell also program will ask the user for their personal details which I'd he wants to access laptop details and program will ask for the user if he wants to sell more laptops or not if yes then program will runs in a loop but if not then it will ask for the user's if he wants to add shipping cost to his item or not if yes then invoice with shipping cost added is generated else shipping cost is not added to invoice.the program will run till until user wants to exit from it otherwise it will runs it won't break from middle of the program simply if user wants it to exit then only it will exit.

as textfile.Implementation in code is shown below:

```
------------------------------|
Welcome to our laptop shop
------------------------------|
1. buy_products
2. sell_products
3. exit


------------------------------|

please choose an option: 1
```

```
                                                    Welcome to our shop

-------------------------------------------------------------------------------------------------------------------------------------|
        brandname |             laptopname |              price |               quantity |        processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------|


1     Razer Blade                  Razer                  $2000                   20              i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------|


2     XPS                          Dell                   $1976                   15              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|


3     Alienware                    Alienware              $1978                   24              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|


4     Swift 7                      Acer                   $900                    12              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|


5     Macbook Pro 16               Apple                  $3500                   10              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: apil
```

Figure 4 buy 1 figure

```
Enter name of our customer: apil
Enter phone number of our customer: 9889428234
Enter id for laptop you want to buy: 5
Enter number of quantity you want to buy: 3


                user buyed item list
-------------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------------
  Apple         3        3500           10500
-------------------------------------------------------------------------|

Do you want to buy another laptop(yes/No)Press 'y' for yes and 'n' for No: y

Enter id for laptop you want to buy: 5
Enter number of quantity you want to buy: 4


                user buyed item list
-------------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------------
  Apple         3        3500           10500
-------------------------------------------------------------------------|

  Apple         4        3500           14000
-------------------------------------------------------------------------|
```

Figure 5 buy 2 figure

In sell generally program will ask for the customers name and phone number.and product id also user quantity if they all are valid it will ask us if we want to buy other laptops too if yes then it will run in a loop if not then it will give us choice either we wants to add shipping charge to our item or not invoice will be generated accodingly.

```
File  Edit  Shell  Debug  Options  Window  Help
1. buy_products
2. sell_products
3. exit


-----------------------------|

please choose an option: 2




                                                        Welcome to our shop

-------------------------------------------------------------------------------------------------------------------------------------------------------|
        brandname |             laptopname |              price |               quantity |        processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------------------------|


1     Razer Blade                  Razer                  $2000                   20              i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------------------------|


2     XPS                          Dell                   $1976                   15              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------|


3     Alienware                    Alienware              $1978                   24              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------|


4     Swift 7                      Acer                   $900                    12              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------|


5     Macbook Pro 16               Apple                  $3500                   10              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------|
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: apil
Enter phone number of our customer: 213433245
Enter id for laptop : 4
Enter number of quantity : 3
```

Figure 6 sell figure

```
                user sell item list
-------------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------------
  Acer          3        900            2700
-------------------------------------------------------------------------|


Do you want to sell any other laptops?(y/n): n


Dear user will you be allowing us to add shipping cost to your pc(y/N)?  :  y
```

Figure 7 sell 2 figure

Textfile is created with open method

write.py - C:\Users\apil xetri\OneDrive\Desktop\Coursework(220167755)foc\write.py (3.11.2)

File Edit Format Run Options Window Help

```
with open(add,"w")as customerbill:



    ("Bill with shipping cost included ")
    customerbill.write("-------------------------------------------------------------------------\n")
    customerbill.write("\t\tBill of our shop | Apil electronics\n")
    customerbill.write("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
    customerbill.write("\t\t--------------------------------------\n\n")
    customerbill.write("\t\tname of  our customer is: ")
    customerbill.write("\t       "+str(name)+"\n")

    customerbill.write("\t\tphone number of our customer is: ")
    customerbill.write("\t " +str(int(phone))+"\n")

    for j in user_requirement:

        customerbill.write("\t\tThe name of laptop is\t\t      "+str(j[0])+"\n")
        customerbill.write("\t\tThe number of quantity is\t\t "+str(j[1])+"\n")

    customerbill.write("\t\tThe shipping cost is: ")
    customerbill.write("\t\t\t "+str(DElivery_cost))
    customerbill.write("\n\n")
    customerbill.write("-------------------------------------------------------------------------------")
    customerbill.write("\n")
    customerbill.write("\t\tThe total price is \t\t\tRS "+str(total_cost_is))
    customerbill.write("\n")
    customerbill.write("-------------------------------------------------------------------------------")
```

Figure 8 creating textfile

```
-----------------------------------------------------------------------
            Bill of our shop | Apil electronics
                  Date: 2023-05-08 21:45:05
            ---------------------------------------

            name of  our customer is:            apil
            phone number of our customer is:     9867004146
            The name of laptop is                Apple
            The number of quantity is            4
            The name of laptop is                Acer
            The number of quantity is            9
            The shipping cost is:                1000

-----------------------------------------------------------------------
            The total price is                  RS 23100
-----------------------------------------------------------------------
```

Figure 9 showing the bill

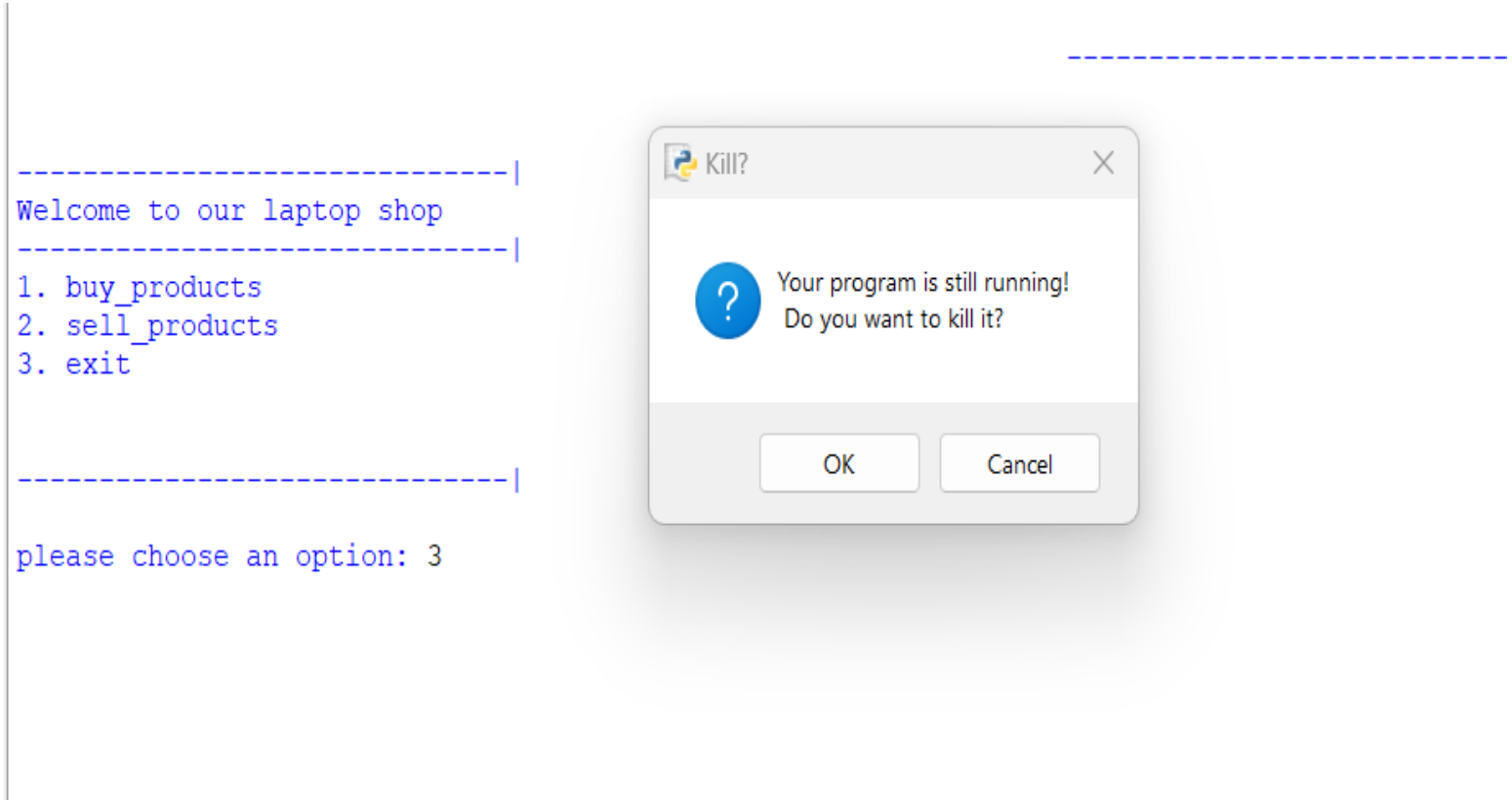

Figure 10 Termination of the program

When user clicks ‘3’ the program will exit / terminate.

4. Testing

4.1 Table 1

Table 1 to test for showing implementation of try/except.

Test	**1**
Objective	To show implementation of try/except.
Action	➡ To enter invalid input and show the message.
Expected result	Should display the message when invalid input is entered.
Actual result	Message is displayed when invalid value is entered.
conclusion	This test is successful.

```
while True:

    try:

        phone = int(input("Enter phone number of buyer: "))
        break
    except:
        print("Please enter valid information")
```

Figure 11 show implementaion of try/except

```
Enter phone number of buyer: as
Please enter valid information
Enter phone number of buyer:
```

Figure 12 expected output implementing try/except

4.2 Table 2

Table 2 To test to provide negative and non existed value for sale and purchase laptops.

Objective	To provide negative and non existed value for sale and purchase laptops.
Action	➡ We provide negative value for sell and buy products. ➡ We provide non existed value for sell and buy products.
Expected result	Should display the message also need ask for input again.
Actual result	Message displayed and asked for input value again.

conclusion	This test is successful.

4.2.1 for purchase

```
Enter phone number of our customer: -4
please enter positive value
Enter phone number of our customer:
```

Figure 13 when negative value entered

```
Enter id for laptop you want to buy: 8
Please enter valid id which is available in our stock
Enter id for laptop you want to buy:
```

Figure 14 non existed value entered as input

4.2.2 for sell

```
File  Edit  Shell  Debug  Options  Window  Help
------------------------------|
Welcome to our laptop shop
------------------------------|
1. buy_products
2. sell_products
3. exit


------------------------------|

please choose an option: 2




                                                        Welcome to our shop

-------------------------------------------------------------------------------------
        brandname |              laptopname |              price |               quan
-------------------------------------------------------------------------------------


1     Razer Blade                  Razer                  $2000                   20
-------------------------------------------------------------------------------------


2     XPS                          Dell                   $1976                   15
-------------------------------------------------------------------------------------


3     Alienware                    Alienware              $1978                   24
-------------------------------------------------------------------------------------


4     Swift 7                      Acer                   $900                    12
-------------------------------------------------------------------------------------


5     Macbook Pro 16               Apple                  $3500                   10
-------------------------------------------------------------------------------------
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: apil
Enter phone number of our customer: -6
please enter positive value
Enter phone number of our customer:
```

Figure 15 for negative value

```
*IDLE Shell 3.11.2*
File  Edit  Shell  Debug  Options  Window  Help
1. buy_products
2. sell_products
3. exit


-------------------------------|

please choose an option: 2




                                                        Welcome to our shop

-------------------------------------------------------------------------------------
        brandname |              laptopname |              price |                quant
-------------------------------------------------------------------------------------


1     Razer Blade                  Razer                  $2000                    20
-------------------------------------------------------------------------------------


2     XPS                          Dell                   $1976                    15
-------------------------------------------------------------------------------------


3     Alienware                    Alienware              $1978                    24
-------------------------------------------------------------------------------------


4     Swift 7                      Acer                   $900                     12
-------------------------------------------------------------------------------------


5     Macbook Pro 16               Apple                  $3500                    10
-------------------------------------------------------------------------------------
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: apil
Enter phone number of our customer: -6
please enter positive value
Enter phone number of our customer: 2
Enter id for laptop : 9
Please enter valid id which is available in our stock
Enter id for laptop :
```

Figure 16 non existed value entered

4.3 table 3

Table 3 to test to show complete purchase process, output and purchased details.

Objective	To show complete purchase process. To show output in the shell as well. To show the purchased laptop details in a textfile.
Action	➡ Complete purchase process is shown. ➡ Output is shown In the shell. ➡ Purchased laptop details is shown in a textfile.
Expected result	Multiple items need to be purchased. Output should be there in console/screen. Output should be in textfile.
Actual result	Multiple items was purchased successfully.. Output displayed in screen successfully. Output displayed in the form of textfile also.
conclusion	This test is successful.

```
*IDLE Shell 3.11.2*
File  Edit  Shell  Debug  Options  Window  Help

------------------------------|
Welcome to our laptop shop
------------------------------|
1. buy_products
2. sell_products
3. exit


------------------------------|

please choose an option: 1




                                                        Welcome to our shop

------------------------------------------------------------------------------------------------------------------------------------------------------|
        brandname |             laptopname |              price |               quantity |        processor |     graphics
------------------------------------------------------------------------------------------------------------------------------------------------------|


1     Razer Blade                  Razer                  $2000                   20              i7 7th Gen      GTX 3060
------------------------------------------------------------------------------------------------------------------------------------------------------|


2     XPS                          Dell                   $1976                   15              i5 9th Gen      GTX 3070
------------------------------------------------------------------------------------------------------------------------------------------------------|


3     Alienware                    Alienware              $1978                   24              i5 9th Gen      GTX 3070
------------------------------------------------------------------------------------------------------------------------------------------------------|


4     Swift 7                      Acer                   $900                    12              i5 9th Gen      GTX 3070
------------------------------------------------------------------------------------------------------------------------------------------------------|


5     Macbook Pro 16               Apple                  $3500                   10              i5 9th Gen      GTX 3070
------------------------------------------------------------------------------------------------------------------------------------------------------|
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: ApilThapa
Enter phone number of our customer: 98670041456
Enter id for laptop you want to buy: 5
```

Figure 17 purchase process 1

```
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------
  Apple          6         3500            21000
-------------------------------------------------------------------|


Do you want to buy another laptop(yes/No)Press 'y' for yes and 'n' for No: y


Enter id for laptop you want to buy: 4
Enter number of quantity you want to buy: 9


                 user buyed item list
-------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------
  Apple          6         3500            21000
-------------------------------------------------------------------|


  Acer           9         900             8100
-------------------------------------------------------------------|


Do you want to buy another laptop(yes/No)Press 'y' for yes and 'n' for No: y


Enter id for laptop you want to buy: 2
Enter number of quantity you want to buy: 5


                 user buyed item list
-------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------
  Apple          6         3500            21000
-------------------------------------------------------------------|


  Acer           9         900             8100
-------------------------------------------------------------------|


  Dell           5         1976            9880
```

Figure 18 purchase process 2

```
-----------------------------------------------------------------------
                Bill of our shop | Apil electronics
                        Date: 2023-05-10 22:38:18
                ----------------------------------------

                name of  buyer is:                  ApilThapa
                phone number of buyer is:           98670041456
                The name of laptop is               Apple
                The number of quantity is           6
                The name of laptop is               Acer
                The number of quantity is           9
                The name of laptop is               Dell
                The number of quantity is           5
                net_amount is:                      9880

                vat_amount is:                     1284.4

                gross_amount is:                    11164.4

-----------------------------------------------------------------------
                The total price is                RS 38980
-----------------------------------------------------------------------
```

Figure 19 Bill in console

```
-----------------------------------------------------------------------
            Bill of our shop | Apil electronics
                  Date: 2023-05-10 22:38:18
            ---------------------------------------

            name of  buyer is:              ApilThapa
            phone number of buyer is:       98670041456
            The name of laptop is           Apple
            The number of quantity is       6
            The name of laptop is           Acer
            The number of quantity is       9
            The name of laptop is           Dell
            The number of quantity is       5
            net_amount is:                  9880
            vat_amount is:                  1284.4
            gross_amount is:               11164.4
-------------------------------------------------------------------------
            The total price is             RS 38980
-------------------------------------------------------------------------
```

Figure 20 Bill in textfile

4.4 Table 4

Table 4 To test to show complete sell process, output and selled details list.

Objective	To show complete sell process. To show output in the shell as well. To show the selled laptop details in a textfile.
Action	➡ Complete sell process is shown. ➡ Output is shown In the shell. ➡ Selled laptop details is shown in a textfile.
Expected result	Multiple items need to be selled. Output should be there in console/screen. Output should be in textfile.
Actual result	Multiple items was selled successfully. Output displayed in screen successfully. Output displayed in the form of textfile.
conclusion	This test is successful.

```
                                                                                    ----------------------------------------------------
-------------------------------|
Welcome to our laptop shop
-------------------------------|
1. buy_products
2. sell_products
3. exit


-------------------------------|

please choose an option: 2




                                                                                    Welcome to our shop

-------------------------------------------------------------------------------------------------------------------------------------------------------------|
        brandname |                laptopname |                price |                quantity |        processor |      graphics
-------------------------------------------------------------------------------------------------------------------------------------------------------------|


1     Razer Blade                    Razer                    $2000                    20              i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------------------------------|


2     XPS                            Dell                     $1976                    15              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------------|


3     Alienware                      Alienware                $1978                    24              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------------|


4     Swift 7                        Acer                     $900                     12              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------------|


5     Macbook Pro 16                 Apple                    $3500                    10              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------------------------------|
```

Figure 21 sales process 1

```
*IDLE Shell 3.11.2*
File   Edit   Shell   Debug   Options   Window   Help
5      Macbook Pro 16                Apple                    $3500                    10
--------------------------------------------------------------------------------------------
This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments
Enter name of our customer: apilThapa
Enter phone number of our customer: 9867004146
Enter id for laptop : 5
Enter number of quantity : 4




                user sell item list
-------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------
  Apple         4        3500           14000
-------------------------------------------------------------------|


Do you want to sell any other laptops?(y/n): y
Enter id for laptop : 3
Enter number of quantity : 4




                user sell item list
-------------------------------------------------------------------|
Laptopname  quantity Individal price   total price
-------------------------------------------------------------------
  Apple         4        3500           14000
-------------------------------------------------------------------|


  Alienware     4        1978           7912
-------------------------------------------------------------------|


Do you want to sell any other laptops?(y/n): n
```

Figure 22 sales proces 2

```
-----------------------------------------------------------------------
                Bill without shipping cost
-----------------------------------------------------------------------
                Bill of our shop | Apil electronics
                        Date: 2023-05-10 22:57:03
                ---------------------------------------

                name of  buyer is:                        apilThapa
                phone number of buyer is:                 9867004146
                The name of laptop is                     Apple
                The number of quantity is                 4
                The name of laptop is                     Alienware
                The number of quantity is                 4
-----------------------------------------------------------------------
                The total price is                      RS  21912
-----------------------------------------------------------------------
```

Figure 23 console bill for non shipping

```
-------------------------------------------------------------------------------
          Bill of our shop | Apil electronics
                Date: 2023-05-10 22:57:03
          ----------------------------------------

          name of  our customer is:           apilThapa
          phone number of our customer is:    9867004146
          The name of laptop is               Apple
          The number of quantity is           4
          The name of laptop is               Alienware
          The number of quantity is           4


-------------------------------------------------------------------------------
          The total price is                  RS 21912
-------------------------------------------------------------------------------
```

Figure 24 Bill in textfile for non shipping

```
                Bill with shipping cost included
-------------------------------------------------------------------

                Bill of our shop | Apil electronics

                        Date: 2023-05-10 23:05:57

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

                name of  our customer is:                apilThapa

                phone number of our customer is:         9867004146

                The name of laptop is                    Apple

                The number of quantity is                4

                The name of laptop is                    Alienware

                The number of quantity is                4

                Shipping cost is:                        1000

-------------------------------------------------------------------------
                The total price is                     RS  22912
-------------------------------------------------------------------------
```

Figure 25 Console bill with shipping cost

```
-----------------------------------------------------------------
          Bill of our shop | Apil electronics
                Date: 2023-05-10 22:57:03
          ---------------------------------------

          name of  our customer is:           apilThapa
          phone number of our customer is:    9867004146
          The name of laptop is               Apple
          The number of quantity is           4
          The name of laptop is               Alienware
          The number of quantity is           4


-----------------------------------------------------------------
          The total price is                  RS 21912
-----------------------------------------------------------------
```

Figure 26 Bill in textfile with shipping cost

4.5 Table 5

Table 5 To test to show quantity being deducted while selling and added while purchasing.

Objective	To show Quantity being added while purchasing laptops. To show Quantity being deducted while selling laptops.
Action	➡ Purchase laptop with some amount of quantity. ➡ Sells laptop with some amount of quantity.
Expected result	Stock need to be updated while purchasing laptops. Stock need to be deducted while selling laptops.
Actual result	Quantity is added to previous stock after user purchased. Quantity is deducted to previous stock after selled.
conclusion	This test is successfull.

4.5.1 for purchased

```
                                        Welcome to our shop

-------------------------------------------------------------------------------------------------------------------------------------|
        brandname |            laptopname |             price |              quantity |        processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------|

1     Razer Blade                Razer                $2000                  20             i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------|

2     XPS                        Dell                 $1976                  15             i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

3     Alienware                  Alienware            $1978                  24             i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

4     Swift 7                    Acer                 $900                   12             i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

5     Macbook Pro 16             Apple                $3500                  10             i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|
```

Figure 27 Stock before user purchased

```
                                    Your || Details of shop

-------------------------------------------------------------------------------------------------------------------------------------|
        brandname |             laptopname |              price |               quantity |        processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------|

1     Razer Blade                  Razer                  $2000                   20              i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------|

2     XPS                          Dell                   $1976                   15              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

3     Alienware                    Alienware              $1978                   24              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

4     Swift 7                      Acer                   $900                    12              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

5     Macbook Pro 16               Apple                  $3500                   60              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|
```

Figure 28 Stock update after user purchase 50 macbook in console

```
                                    Your || Details of shop
-------------------------------------------------------------------------------------------------------------
      brandname       laptopname            price     quantity              processor         graphics
-------------------------------------------------------------------------------------------------------------
1    Razer Blade          Razer              $2000     20                i7 7th Gen          GTX 3060
      -------------------------------------------------------------------------------------------------------

2    XPS                  Dell               $1976     15                i5 9th Gen          GTX 3070
      -------------------------------------------------------------------------------------------------------

3    Alienware            Alienware                $1978     24                i5 9th Gen          GTX 3070
      -------------------------------------------------------------------------------------------------------

4    Swift 7                Acer                   $900      12                i5 9th Gen          GTX 3070
      -------------------------------------------------------------------------------------------------------

5    Macbook Pro 16             Apple              $3500     60                i5 9th Gen          GTX 3070
      -------------------------------------------------------------------------------------------------------
```

Figure 29 Stock update ater user purchase in textfile

4.5.2 for sale

```
                                        Welcome to our shop

-------------------------------------------------------------------------------------------------------------------------------------|
        brandname |            laptopname |            price |            quantity |       processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------|

1     Razer Blade                Razer                $2000                20            i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------|

2     XPS                        Dell                 $1976                15            i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

3     Alienware                  Alienware            $1978                24            i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

4     Swift 7                    Acer                 $900                 12            i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|

5     Macbook Pro 16             Apple                $3500                10            i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|
```

Figure 30 stock before user sell

```
                                    Your || Details of shop

-------------------------------------------------------------------------------------------------------------------------------------|
        brandname |             laptopname |              price |               quantity |        processor |     graphics
-------------------------------------------------------------------------------------------------------------------------------------|

1     Razer Blade                  Razer                  $2000                   20              i7 7th Gen      GTX 3060
-------------------------------------------------------------------------------------------------------------------------------------|



2     XPS                          Dell                   $1976                   15              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|



3     Alienware                    Alienware              $1978                   24              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|



4     Swift 7                      Acer                   $900                   4                i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|



5     Macbook Pro 16               Apple                  $3500                   10              i5 9th Gen      GTX 3070
-------------------------------------------------------------------------------------------------------------------------------------|
```

Figure 31Stock in console after selled

```
                                    Your || Details of shop
-------------------------------------------------------------------------------------------------------------------
     brandname       laptopname            price     quantity              processor         graphics
-------------------------------------------------------------------------------------------------------------------
1    Razer Blade          Razer              $2000     20                i7 7th Gen          GTX 3060
     -------------------------------------------------------------------------------------------------------------

2    XPS                  Dell               $1976     15                i5 9th Gen          GTX 3070
     -------------------------------------------------------------------------------------------------------------

3    Alienware            Alienware                $1978     24                i5 9th Gen          GTX 3070
     -------------------------------------------------------------------------------------------------------------

4    Swift 7                Acer                   $900      4                i5 9th Gen          GTX 3070
     -------------------------------------------------------------------------------------------------------------

5    Macbook Pro 16             Apple              $3500     10                i5 9th Gen          GTX 3070
     -------------------------------------------------------------------------------------------------------------
```

Figure 32 after selled in textfile

5. Conclusion.

Simply, we have created a laptop shop program using python in which user can purchased and sale their laptops.

our main objectives of this project is to develop a program which can buy, sell and generated invoice after users buy or sell the laptops.our main goal is to create a shop which keeps tracks of stock in the shop generallly it asked user for the sell or buy laptops after that repesctive details is inputed and users can do transaction based on the stock in the shop, user will not exited until they want to do so.

It generates an invoice immediately after transaction with full details, price, quantity everything. main key features for this project is that user wont face any difficulties while buying or selling items in our shop.we have put it out better functionality so that user wont be bored and can transact in a good way with no errors and exceptions.

We have faced many difficulties while creating this laptop shop program.it uses dictionary so I have faced many keyerrors at first while storing details.many synatx errors and problems in while loops after user wants to purchase more laptops,it was difficult to handle out each carefully.while asking for the input at first we need to manage user can enter any value so that could exited the program so I have putted that inside try and except,also user need to asked input until valid input is entered by user , that should also be handled carefully.while creating text file for writing bile into that textfile it was difficult to manage all details is same format as titile,because we are not allowed to use string formatting methods it was very challenging for us to manage those details in a same formats . We need to try that through hit and trial which killed my

more time.that was also one challenge for me. Other is while generating invoice by reading user purchased and selled list appending to those details to the variable and also to textfile was also a great challenge. Overall many difficulties were seen, but one thing we have learned from this project is we are able to understood dictionary, list also other dsata structures, how project is carried out and how we need to think logically to solve the problem, this kind of skills were develoed through this coursework.

Many difficulties were faced but also certain things I implement to overcome those difficulties were going to discuss here. First we need to understood concept theroritically about those data structures or method else anything which is essential while solving those problems. After that we need to try to write code at least what we have learned theroitically, without practicing those concepts it won’t be possible to learn those programming.i learned dictionary and list from different online materials and try to implement those things but it is difficult to implement , at first how we can create logics in programming it Is very hard to find out those things.i wrote program for buy products I know that how we take input from the user and try to implement that featues and I successfully did also but after that things weren’t clear that how can we generate bills,invoice etc and how that user input is kept under try/except block to avoid unnecessary values. There was a confusion for me in the while loop till that time. Whenever I started to understood about the concepts how it is done then again I started to write those code. Many errors like I have used dictionary so keyerror was occurred for those I lost many

days solving that errors.after that it was very difficult for me to understood how program can be break out in different modules. Concepts of global and local variable with functions was very tough to understand. Another challanging factors for me is to call from main.py module and need to acess all those modules right after buying or selling invoice should be generated so by overcoming all those difficulties I have successfully created an python laptop shop program.

6. Appendix

6.1 for main.py module

```
from read import product_dict,user_need,user_requirement

from operations import buy_products

from write import bill_generate

import read

import operations

import write

import datetime



now = datetime.datetime.now()

print("--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------\n")

print("\t\t\t\t\t\t\t\t\tApil Laptop and Pc shop || 9867004146\n")

print("\t\t\t\t\t\t\t\t------------------------------------------------------------")
```

```
while True:
    user_menus=('1','2','3')
    print("\n")
    print("------------------------------|")
    print("Welcome to our laptop shop ")
    print("------------------------------|")
    print("1. buy_products")
    print("2. sell_products")
    print("3. exit")
    print("\n")
    print("------------------------------|"+"\n")
    user_value = input("please choose an option: ")

    if user_value in user_menus:

        if user_value == '1':

            read.display_products()
            print(read.display_products.__doc__)
```

```
    name ,phone = operations.buy_products(product_dict,now,user_need)
    print(operations.buy_products.__doc__)




elif user_value == '2':
    read.display_products()

    print(read.display_products.__doc__)

    operations.sell_products(product_dict,now,user_requirement)
    print(operations.sell_products.__doc__)
```

```
    else:
        exit()
else:
    print("\n")
    print("Option Not available ")
```

6.2 For operations.py module

```
from write import bill_generate,shippbill_generate
import datetime
def buy_products(product_dict,now,user_need):
    '''This is a buy products function in which user give their details including which lapop he wants to buy
        Takes dictionary date now and another list as arguments
       return name and phone of the user'''
    while True:



       name=input("Enter name of our customer: ")
       if name.strip():
          if name.isalpha():
             break
          else:

             print("please enter valid name")
             name=input("Enter name of our customer: ")
```

```
    else:
        print("Name cant be empty")

while True:

    try:

        phone = int(input("Enter phone number of our customer: "))
        if phone < 0:
            print("please enter positive value")
        else:
            break
    except:
        print("Please enter valid information")
loop = True
while loop == True:



    while True:
        try:
```

```
        product_id =int(input("Enter id for laptop you want to buy: "))
        break
    except:
        print("please enter valid information")



while int(product_id) <= 0 or int(product_id) > len(product_dict):
    print("Please enter valid id which is available in our stock")
    product_id =int(input("Enter id for laptop you want to buy: "))

while True:

    try:
        user_quantity = int(input("Enter number of quantity you want to buy: "))
        break

    except:
        print("please enter integer value")

get_quantity = int(product_dict[product_id][3])
```

```
 #update file
product_dict[product_id][3] = int(product_dict[product_id][3]) + int(user_quantity)
product_name=product_dict[product_id][1]
quantity_selected=int(user_quantity)
each_price=product_dict[product_id][2].replace("$","").strip()
total_price=int(quantity_selected)*int(each_price)

#putting this value in list and appending to user requirement

full_item=[product_name,quantity_selected,each_price,total_price]
user_need.append(full_item)
print("\n\n")
print("\t\t"+"user buyed item list")
print("-----------------------------------------------------------------------|")
print(""+"Laptopname"+" "+" quantity"+" "+"Individal price"+""+"   total price")
print("-----------------------------------------------------------------------")
for item in user_need:
    print("",str(item[0])," \t"+str(item[1]),"\t",str(item[2]),"\t\t"+str(item[3]),"\t\t\t  ")
    print("-----------------------------------------------------------------------|")
```

```
            print("\n")


        choice= input("Do you want to buy another laptop(yes/No)Press 'y' for yes and 'n' for No: ")
        print("\n")
        if choice.lower() == 'n':
            loop = False
            bill_generate(name,phone,now,user_need)

            print("\n")

    return name,phone




def sell_products(product_dict,now,user_requirement):
    '''This is a sell products function which give the details of our customer what he wants to sell
        Takes dictionary date now and another list as arguments
        return name and phone number of the customer'''
```

```
while True:

    name=input("Enter name of our customer: ")
    if name.strip():

        if name.isalpha():
            break

        else:
            print("please enter valid name")
            name=input("Enter name of our customer: ")

    else:
        print("Name cant be empty")

while True:

    try:
        phone = int(input("Enter phone number of our customer: "))
```

```
        if phone < 0:
            print("please enter positive value")
        else:
            break



    except:
        print("Please enter valid information")

main=True
while main == True:

    while True:



        try:


            product_id =int(input("Enter id for laptop : "))
            break
        except:
```

```
        print("please enter valid information")
while int(product_id) <= 0 or int(product_id) > len(product_dict):
    print("Please enter valid id which is available in our stock")
    product_id =int(input("Enter id for laptop : "))




while True:

    try:
        user_quantity = int(input("Enter number of quantity : "))
        break
    except:
        print("please enter integer value")


take_quantity = int(product_dict[product_id][3])
```

```
while int(user_quantity) <= 0 or int(user_quantity) > take_quantity:
    print("Please enter valid quantity available in our stock")
    user_quantity = input("Enter number of quantity : " )



product_name=product_dict[product_id][1]
quantity_selected=int(user_quantity)
each_price=product_dict[product_id][2].replace("$","").strip()
total_price=int(quantity_selected)*int(each_price)

full_item=[product_name,quantity_selected,each_price,total_price]
user_requirement.append(full_item)


#total_cost = int(user_quantity) * int(product_dict[product_id][2].replace("$","").strip())
```

```
    #update quantity
    print("\n\n")
    product_dict[product_id][3]=int(product_dict[product_id][3])-int(user_quantity)
    print("\n\n")
    print("\t\t"+"user sell item list")
    print("-----------------------------------------------------------------------|")
    print(""+"Laptopname"+" "+" quantity"+" "+"Individual price"+""+"   total price")
    print("-----------------------------------------------------------------------")
    for item in user_requirement:
        print("",str(item[0])," \t"+str(item[1]),"\t",str(item[2]),"\t\t"+str(item[3]),"\t\t\t  ")
        print("-----------------------------------------------------------------------|")
        print("\n")
    choice=input("Do you want to sell any other laptops?(y/n): ")
    if choice.lower() == "n":
        main=False
        shippbill_generate(name,phone,now,user_requirement)

        #putting this value in list and appending to user requirement

return name,phone
```

6.3 for write.py module

```
import datetime
from read import product_dict,user_need,user_requirement

def bill_generate(name,phone,now,user_need):

    '''This is a function which generates bill of buying laptops from manufacturers
        Takes name ,phone number of manufacturers and list as arguments
        returns empty'''

    status="the_lucky_buyer"
    year=str(datetime.datetime.now().year)
    month=str(datetime.datetime.now().month)
    day=str(datetime.datetime.now().day)
    hour=str(datetime.datetime.now().hour)
    minute=str(datetime.datetime.now().minute)
    seconds=str(datetime.datetime.now().second)
    interesting_time=year+month+day+hour+minute+seconds
    total_price=0
```

```
print("-----------------------------------------------------------------------\n")
print("\t\tBill of our shop | Apil electronics\n")
print("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
print("\t\t-------------------------------------\n\n")
print("\t\tname of  buyer is: ",end="")
print("\t\t  "+str(name)+"\n")
print("\t\tphone number of buyer is: ",end="")
print("\t  " +str(int(phone))+"\n")
total_pric=0
for product in user_need:

    laptop_name=product[0]
    laptop_quant=product[1]
    total_price=product[2]
    total_fullamt=product[3]
    net_amount=int(product[1])*int(product[2])
    vat_amount=net_amount*0.13
    gross_amount=vat_amount+net_amount
    total_pric+=int(total_fullamt)
```

```
    print("\t\tThe name of laptop is\t\t",str(laptop_name)+"\n")
    print("\t\tThe number of quantity is\t ",str(laptop_quant)+"\n")
print("\t\tnet_amount is: ",end="")
print("\t\t\t ",net_amount)
print("\n")
print("\t\tvat_amount is: ",end="")
print("\t\t\t",vat_amount)
print("\n")
print("\t\tgross_amount is: \t\t ",gross_amount)

print("\n")
print("-------------------------------------------------------------------------------")
print("\t\tThe total price is \t\t RS",total_pric)
print("-------------------------------------------------------------------------------")
add = name + "_"+status+"_"+interesting_time+".txt"

with open(add,"w")as bill:
```

```
bill.write("-----------------------------------------------------------------------\n")
bill.write("\t\tBill of our shop | Apil electronics\n")
bill.write("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
bill.write("\t\t--------------------------------------\n\n")
bill.write("\t\tname of  buyer is: " )
bill.write("\t\t "+str(name)+"\n")
bill.write("\t\tphone number of buyer is: ")
bill.write("\t " +str(int(phone))+"\n")
for i in user_need:
    bill.write("\t\tThe name of laptop is\t\t"+str(i[0])+"\n")
    print("\n")
    bill.write("\t\tThe number of quantity is\t "+str(i[1])+"\n")
bill.write("\t\tnet_amount is:\t\t\t "+str(net_amount))
bill.write("\n")
bill.write("\t\tvat_amount is:\t\t\t "+str(vat_amount))
bill.write("\n")
bill.write("\t\tgross_amount is: \t\t\t"+str(gross_amount))
```

```
    bill.write("\n")
    bill.write("-----------------------------------------------------------------------------")
    bill.write("\n")
    bill.write("\t\tThe total price is \t\tRS "+str(total_pric))
    bill.write("\n")
    bill.write("-----------------------------------------------------------------------------")

    with open("update.txt","w")as billupdate:

        billupdate.write("\n")
        billupdate.write("\t\t\t\t\t \t   Your || Details of shop")
        billupdate.write("\n")
        billupdate.write("-------------------------------------------------------------------------------------------------------------------------------
|")
        billupdate.write("\n")
        billupdate.write("\tbrandname       laptopname   \t       price   \tquantity      \t    processor   \t    graphics")
        billupdate.write("\n")
        billupdate.write( "-------------------------------------------------------------------------------------------------------------------------------
-|")

        for key, value in product_dict.items():
```

```
        billupdate.write("\n")

        billupdate.write("       \t".join([str(key) +"     "+ str(value[0])," "*(9-len(value[0]))+" "+str(value[1])+" "*(9-
len(value[1]))+" \t\t  "+str(value[2])+"\t   "*(4-len(str(value[2])))+"   \t"+str(value[3])+" "*(7-len(str(value[3])))+"\t\t"+

            str(value[4])+"\t "*(3-len(str(value[4])))+"\t\t  "+str(value[5])+"  \t"*(2-len(str(value[5])))+"\t"]))

        billupdate.write("------------------------------------------------------------------------------------------------------------------------------
----|")

        billupdate.write("\n")




    print("\n")

    print("\t\t\t\t\t    Your || Details of shop")

    print("\n")

    print("------------------------------------------------------------------------------------------------------------------------------|")

    print("\tbrandname |\t\tlaptopname |\t\t  price |\t\tquantity |\t  processor |\t  graphics")

    print( "------------------------------------------------------------------------------------------------------------------------------|")


    for key, value in product_dict.items():

      print("\n")
```

```
        print(str(key) +"     ", value[0]," "*(9-len(value[0]))+"\t\t ",value[1]," "*(9-len(value[1])),"\t\t",value[2],""*(4-len(str(value[2]))),"\t\t",value[3]," "*(7-len(str(value[3]))),"\t",
              value[4]," "*(8-len(str(value[4]))),"\t",value[5]," "*(7-len(str(value[5]))))
        print("-----------------------------------------------------------------------------------------------------------------------------------------------|")
        print("\n")



def shippbill_generate(name,phone,now,user_requirement):



    '''This function generates bill of selling including shipping cost to an item,it also takes values as a parameter such as name ,phone number
        datetime now with list as arguments or parameter
          also returns empty'''

    status="the_lucky_customer"
    year=str(datetime.datetime.now().year)
    month=str(datetime.datetime.now().month)
    day=str(datetime.datetime.now().day)
    hour=str(datetime.datetime.now().hour)
    minute=str(datetime.datetime.now().minute)
```

```
seconds=str(datetime.datetime.now().second)
interesting_time=year+month+day+hour+minute+seconds

add = name + "_"+status+"_"+interesting_time+".txt"

print("\n")


shipped = input("Dear user will you be allowing us to add shipping cost to your pc(y/N)?  :  ")
print("\n\n\n")
if shipped == "y":


    DElivery_cost=1000
    total =0

    #amount_total=int(total_cost)
    print("\t\tBill with shipping cost included ")
    print("----------------------------------------------------------------------\n")
    print("\t\tBill of our shop | Apil electronics\n")
```

```
print("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
print("\t\t-------------------------------------\n\n")
print("\t\tname of  our customer is: ",end="")
print("\t\t  "+str(name)+"\n")
print("\t\tphone number of our customer is: ",end="")
print("\t  " +str(int(phone))+"\n")


for pc in user_requirement:

    laptops_is=str(pc[0])
    quantity_is=int(pc[1])
    indiv_qu=int(pc[2])
    cost_is=int(pc[3])
    total+=int(pc[3])

    total_cost_is=int(total)+int(DElivery_cost)

    print("\t\tThe name of laptop is\t\t\t",str(laptops_is)+"\n")
    print("\t\tThe number of quantity is\t\t ",str(quantity_is)+"\n")
```

```
print("\t\tShipping cost is: ",end="")
print("\t\t\t ",DElivery_cost)
print("\n")
print("-------------------------------------------------------------------------------")
print("\t\tThe total price is \t\t\tRS ",total_cost_is)
print("-------------------------------------------------------------------------------")


with open(add,"w")as customerbill:



    ("Bill with shipping cost included ")
    customerbill.write("-------------------------------------------------------------------------\n")
    customerbill.write("\t\tBill of our shop | Apil electronics\n")
    customerbill.write("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
    customerbill.write("\t\t--------------------------------------\n\n")
    customerbill.write("\t\tname of  our customer is: ")
    customerbill.write("\t       "+str(name)+"\n")
```

```
customerbill.write("\t\tphone number of our customer is: ")
customerbill.write("\t " +str(int(phone))+"\n")

for j in user_requirement:

    customerbill.write("\t\tThe name of laptop is\t\t      "+str(j[0])+"\n")
    customerbill.write("\t\tThe number of quantity is\t\t "+str(j[1])+"\n")


customerbill.write("\t\tThe shipping cost is: ")
customerbill.write("\t\t\t "+str(DElivery_cost))
customerbill.write("\n\n")
customerbill.write("--------------------------------------------------------------------------------")
customerbill.write("\n")
customerbill.write("\t\tThe total price is \t\t\tRS "+str(total_cost_is))
customerbill.write("\n")
customerbill.write("--------------------------------------------------------------------------------")


with open("update.txt","w")as billupdatesell:
```

```
billupdatesell.write("\n")
billupdatesell.write("\t\t\t\t\t\t \t   Your || Details of shop")
billupdatesell.write("\n")
billupdatesell.write("-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|")
billupdatesell.write("\n")
billupdatesell.write("\tbrandname       laptopname   \t        price   \tquantity      \t    processor   \t    graphics")
billupdatesell.write("\n")
billupdatesell.write( "-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|")

for key, value in product_dict.items():
    billupdatesell.write("\n")
    billupdatesell.write("       \t".join([str(key) +"    "+ str(value[0])," "*(9-len(value[0]))+" "+str(value[1])+" "*(9-len(value[1]))+" \t\t  "+str(value[2])+"\t   "*(4-len(str(value[2])))+"   \t"+str(value[3])+" "*(7-len(str(value[3])))+"\t\t"+
            str(value[4])+"\t "*(3-len(str(value[4])))+"\t\t  "+str(value[5])+"  \t"*(2-len(str(value[5])))+"\t"]))
    billupdatesell.write("-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|")
    billupdatesell.write("\n")
```

```
print("\n\n\n")
print("\t\t\t\t\t\t    Your || Details of shop")
print("\n")
print("-----------------------------------------------------------------------------------------------------------------------------------------------------|")
print("\tbrandname |\t\tlaptopname |\t\t  price |\t\tquantity |\t  processor |\t  graphics")
print( "-----------------------------------------------------------------------------------------------------------------------------------------------------|")

for key, value in product_dict.items():
    print("\n")
    print(str(key) +"    ", value[0]," "*(9-len(value[0]))+"\t\t ",value[1]," "*(9-len(value[1])),"\t\t",value[2],""*(4-len(str(value[2]))),"\t\t",value[3]," "*(7-len(str(value[3]))),"\t",
        value[4]," "*(8-len(str(value[4]))),"\t",value[5]," "*(7-len(str(value[5]))))
    print("-----------------------------------------------------------------------------------------------------------------------------------------------------|")
```

```
        print("\n")


else:

    totals=0
    print("-----------------------------------------------------------------\n")
    print("\t\tBill without shipping cost ")
```

```
print("-----------------------------------------------------------------------\n")
print("\t\tBill of our shop | Apil electronics\n")
print("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
print("\t\t--------------------------------------\n\n")
print("\t\tname of  buyer is: ",end="")
print("\t\t\t  "+str(name)+"\n")
print("\t\tphone number of buyer is: ",end="")
print("\t\t  " +str(int(phone))+"\n")

for pc in user_requirement:
    laptops_is=str(pc[0])
    quantity_is=int(pc[1])
    indiv_qu=int(pc[2])
    cost_is=int(pc[3])
    totals+=int(pc[3])

    total_cost_is=int(totals)

    print("\t\tThe name of laptop is\t\t\t",str(laptops_is)+"\n")
    print("\t\tThe number of quantity is\t\t ",str(quantity_is)+"\n")
```

```
print("-----------------------------------------------------------------------------")
print("\t\tThe total price is \t\t\t RS ",total_cost_is)
print("-----------------------------------------------------------------------------")




print("\n")
print("\t\t\t\t\t\t    Your || Details of shop")
print("\n")
print("-------------------------------------------------------------------------------------------------------------------------------------|")
print("\tbrandname |\t\tlaptopname |\t\t  price |\t\tquantity |\t  processor |\t  graphics")
print( "-------------------------------------------------------------------------------------------------------------------------------------|")


for key, value in product_dict.items():
    print("\n")
    print(str(key) +"    ", value[0]," "*(9-len(value[0]))+"\t\t ",value[1]," "*(9-len(value[1])),"\t\t",value[2],""*(4-
len(str(value[2]))),"\t\t",value[3]," "*(7-len(str(value[3]))),"\t",
            value[4]," "*(8-len(str(value[4]))),"\t",value[5]," "*(7-len(str(value[5]))))
    print("-------------------------------------------------------------------------------------------------------------------------------------|")
    print("\n")
```

```
with open(add,"w")as customerbill:


    ("Bill without shipping cost included ")
    customerbill.write("-----------------------------------------------------------------------\n")
    customerbill.write("\t\tBill of our shop | Apil electronics\n")
    customerbill.write("\t\t\tDate: " + now.strftime("%Y-%m-%d %H:%M:%S") + "\n")
    customerbill.write("\t\t--------------------------------------\n\n")
    customerbill.write("\t\tname of  our customer is: ")
    customerbill.write("\t      "+str(name))
    customerbill.write("\n")
    customerbill.write("\t\tphone number of our customer is: ")
    customerbill.write("   "+str(int(phone))+"\n")
```

```
for j in user_requirement:
    customerbill.write("\t\tThe name of laptop is\t\t     "+str(j[0])+"\n")
    customerbill.write("\t\tThe number of quantity is\t\t"+str(j[1])+"\n")

customerbill.write("\n")
customerbill.write("\n")
customerbill.write("-------------------------------------------------------------------------------")
customerbill.write("\n")
customerbill.write("\t\tThe total price is\t\t\tRS "+str(total_cost_is))
customerbill.write("\n")
customerbill.write("-------------------------------------------------------------------------------")

with open("update.txt","w")as billupdatesellother:



    billupdatesellother.write("\n")
    billupdatesellother.write("\t\t\t\t\t \t   Your || Details of shop")
    billupdatesellother.write("\n")
    billupdatesellother.write("-------------------------------------------------------------------------------------------------------------------------------------------------------|")
    billupdatesellother.write("\n")
```

```
billupdatesellother.write("\tbrandname      laptopname   \t       price   \tquantity      \t    processor   \t
graphics")

billupdatesellother.write("\n")

billupdatesellother.write( "-----------------------------------------------------------------------------------------------------------------
-------------------|")


for key, value in product_dict.items():

    billupdatesellother.write("\n")

    billupdatesellother.write("       \t".join([str(key) +"    "+ str(value[0])," "*(9-len(value[0]))+" "+str(value[1])+"
"*(9-len(value[1]))+" \t\t  "+str(value[2])+"\t   "*(4-len(str(value[2])))+"   \t"+str(value[3])+" "*(7-len(str(value[3])))+"\t\t"+

            str(value[4])+"\t "*(3-len(str(value[4])))+"\t\t  "+str(value[5])+"  \t"*(2-len(str(value[5])))+"\t"]))

    billupdatesellother.write("-----------------------------------------------------------------------------------------------------------------
----------------------|")

    billupdatesellother.write("\n")
```

6.4 for read.py module

```
user_requirement=[]
user_need=[]
product_dict={}
def display_products():

    '''This program reads the textfile and display the quantity,products in tables format
    returns empty and does not contain any parameter or arguments'''

    id = 1
    with open("product_info.txt", "r") as file:
        for id,line in enumerate(file,start=1):
            line1=line.strip()
            product_dict[id]=line1.split(",")



    print("\n\n\n\n")
```

```
    print("\t\t\t\t\t\t\tWelcome to our shop  "+"\n")
    print("-----------------------------------------------------------------------------------------------------------------------------------------------------------|")
    print("\tbrandname |\t\tlaptopname |\t\t  price |\t\tquantity |\t  processor |\t  graphics")
    print( "-----------------------------------------------------------------------------------------------------------------------------------------------------------|")

    for key, value in product_dict.items():

        print("\n")
        print(str(key) +"    ", value[0]," "*(9-len(value[0]))+"\t\t ",value[1]," "*(9-len(value[1])),"\t\t",value[2],""*(4-
len(value[2])),"\t\t",value[3]," "*(7-len(value[3])),"\t",
                value[4]," "*(8-len(value[4])),"\t",value[5]," "*(7-len(value[5])))
        print("-----------------------------------------------------------------------------------------------------------------------------------------------------------|")
```