LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2021 - 22 Autumn - 1

Student Name: Arjay Bikram Khand

Group: Networking (N1)

London Met ID: 21040602

College ID: NP01NT4A210026

Assignment Due Date: 7th August, 2022

Assignment Submission Date: 7th August, 2022

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 About project

The Bike Management System application is being written in Python, which is used to manage the project. In essence, it is a programme that groups bikes into different categories based on their cost, quantity, and other characteristics. Additionally, a software is used by the company to buy and sell bikes, and it has the function of settling all customer information from buying to selling with all necessary data. The bike's total selling price to customers or the importing company is also indicated. The use of this kind of software results in time and financial savings for the business by making work easier and more enjoyable. Additionally, it will deliver accurate outcomes based on the user-defined function. Such projects will assist in the creation of programmes, and in this project, the user-defined programme and programming language for the bike management system are Python.

1.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics (Anon., 2022). The level built-in data structures and slightly elevated designed architecture make it ideally suited for use as programming or object of agreement modules next to each other. Python's clear, practical grammar promotes accessibility and reduces programme maintenance. Python supports features like components and imports, which improve methods, practises, and reusability. The comprehensive classes and Python code can still be used and published without restriction, and this includes all widely used services.

1.2 Notepad

Notepad is a generic text editor included with all versions of Microsoft Windows that allows you to create, open, and read plaintext files (Anon., 2021). Even though they are not plain files, Notepad may be processing data that has special formatting.

1.3 Draw.io

Draw.io is open-sourced diagram and flowchart software, created for modern sensibilities and obligations in the professional world (J.Graph, 2022). A variety of shapes and templates are available on Draw.io for quick builds. Functioning is unaffected as well. Accurately manage and publish your work while importing and exporting additional data structures, such as the.xml and image formats. Both the web and portable devices are compatible with it.

1.4 Goals and Objectives

The target audience for this project, which is about a bike management system, is those who want to purchase bicycles. Customers can easily access the bikes of their choice. A user can log into the system and browse all the information about the motorcycles online if they are interested in any of them. It is not necessary to frequently stop by the showroom. Anyone with access to the system can get whatever information they need. Finally, the goals and objectives revolve around improving the user and business's ability to complete tasks quickly and easily. Additionally, to put the customer at ease while making the motorcycle purchase.

Users who require bike data but lack the time to search can use the Bike Management System, which is one of the best solutions available. Information and amenities for users are available without errors. The preferred bike models of users are immediately available. Visits to the showroom are not required on a regular basis. To obtain the data they require, anyone can use the system. The ultimate aim is to simplify and improve time efficiency for both the user and the business. To calm the buyer down as they buy bikes, as well.

2. Algorithm

An algorithm is a set of instructions for solving a problem or accomplishing a task (Downey, 2021).

Algorithm of Bike Management System

2.1 For purchasing bike :

Step 1: Start

Step 2: Take the user's input. If the input is a string, print "Please provide only integer values!" and proceed to step 3.

Step 3: Is user input correct? If so, proceed to step 5. If no, print "Please Enter Valid Number and Please Enter 1, 2, 3 Options."

Step 4: If the bike id is a string, print "Please provide only integer values!" then repeat step 4 and proceed to step 5.

Step 5: Is bike id a valid value? If so, proceed to step 6. If not, the message "Please provide a valid Bike ID" is printed.

Step 6: Take the user's input, such as name, address, and contact information, and proceed to step 7.

Step 7: If the quantity is a string, print "Please provide only integer values!", then repeat step 7 and proceed to step 8.

Step 8: Is the quantity correct? If so, proceed to step 9. If no, print "Please enter a valid quantity."

Step 9: Reduce inventory and create a new text file containing bike details, client information, and output. Then proceed to step 10.

Step 10: Do you want to get a new bike? If so, proceed to steps 4 and 5. Then, skip step 6 and proceed to step 8 to add details to an already existing text file. If not, proceed to Step 11.

Step 11: End

Algorithm of Bike Management System

2.2 Adding stock of bike

Step 1: Start

Step 2: Take user input If the input is a string, print "Please give integer value only!" and proceed to step 3.

Step 3: Is user input valid? If yes, proceed to step 5. If not, then print "Please enter a valid number n Please! enter 1,2,3 options"

Step 4 : If the bike id is a string, print "Please give integer value only!" then repeat step 4 and proceed to step 5.

Step 5 : Is bike id correct? If yes, proceed to step 6. If not, print "Please provide a valid Bike ID."

Step 6: Gather information from the dealer, such as the name of the shipping company and the cost of shipping.

Then proceed to step 7.

Step 7: If the quantity is a string, print "Please give integer value only!", then repeat step 7 and proceed to step 8.

Step 8: Is the quantity correct? If so, proceed to step 9. If no, print "Please enter a valid quantity."

Step 9: Increase the amount and Create a new text file with bike information and dealer information, and then display the results.Then proceed to step 10

Step 10: End

Algorithm of Bike Management System

2.3 Existing the program

Step 1: Start

Step 2: Take user input If the input is a string, print "Please give integer value only!" and proceed to step 3.

Step 3: Is user input valid? If yes, proceed to step 4. If not, then print "Please enter a valid number n Please! enter 1,2,3 options"

Step 4: Print "Thanks for using our System"

Step 5: End

3. Flowchart

A flowchart is a type of diagram representing a process using different symbols containing information about steps or a sequence of events (Anon., 2020).

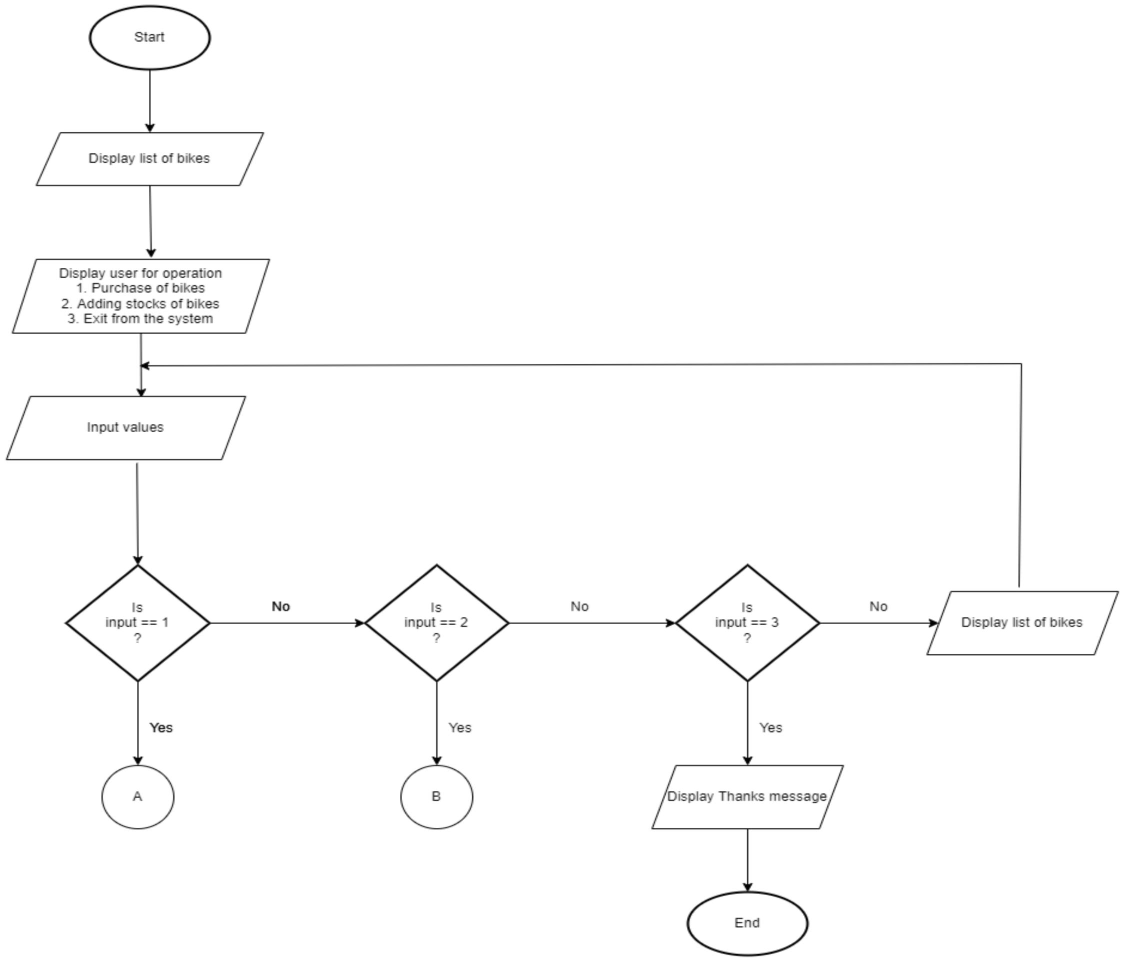


Figure 1: Main Flowchart of Bike Management System

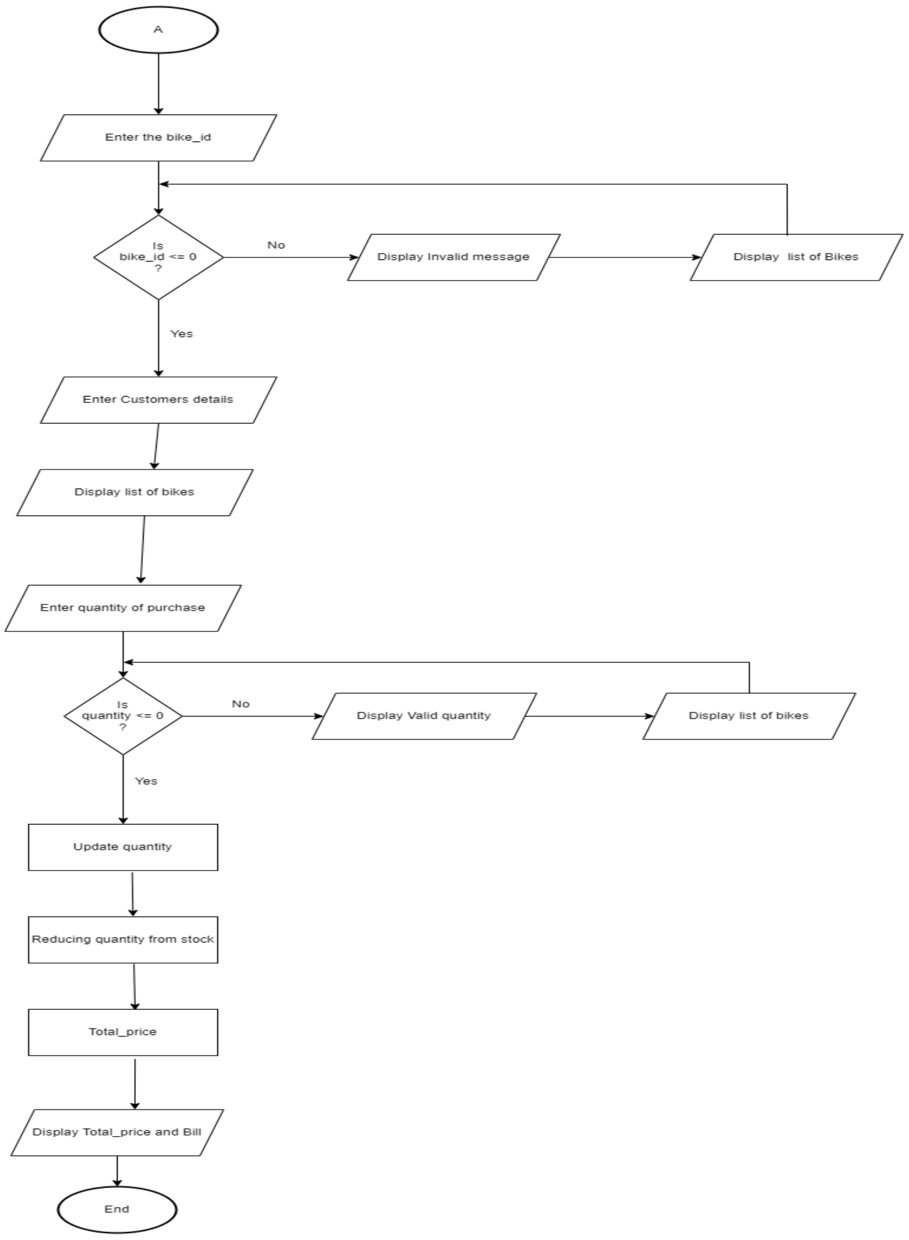


Figure 2: Flowchart of purchasing bike

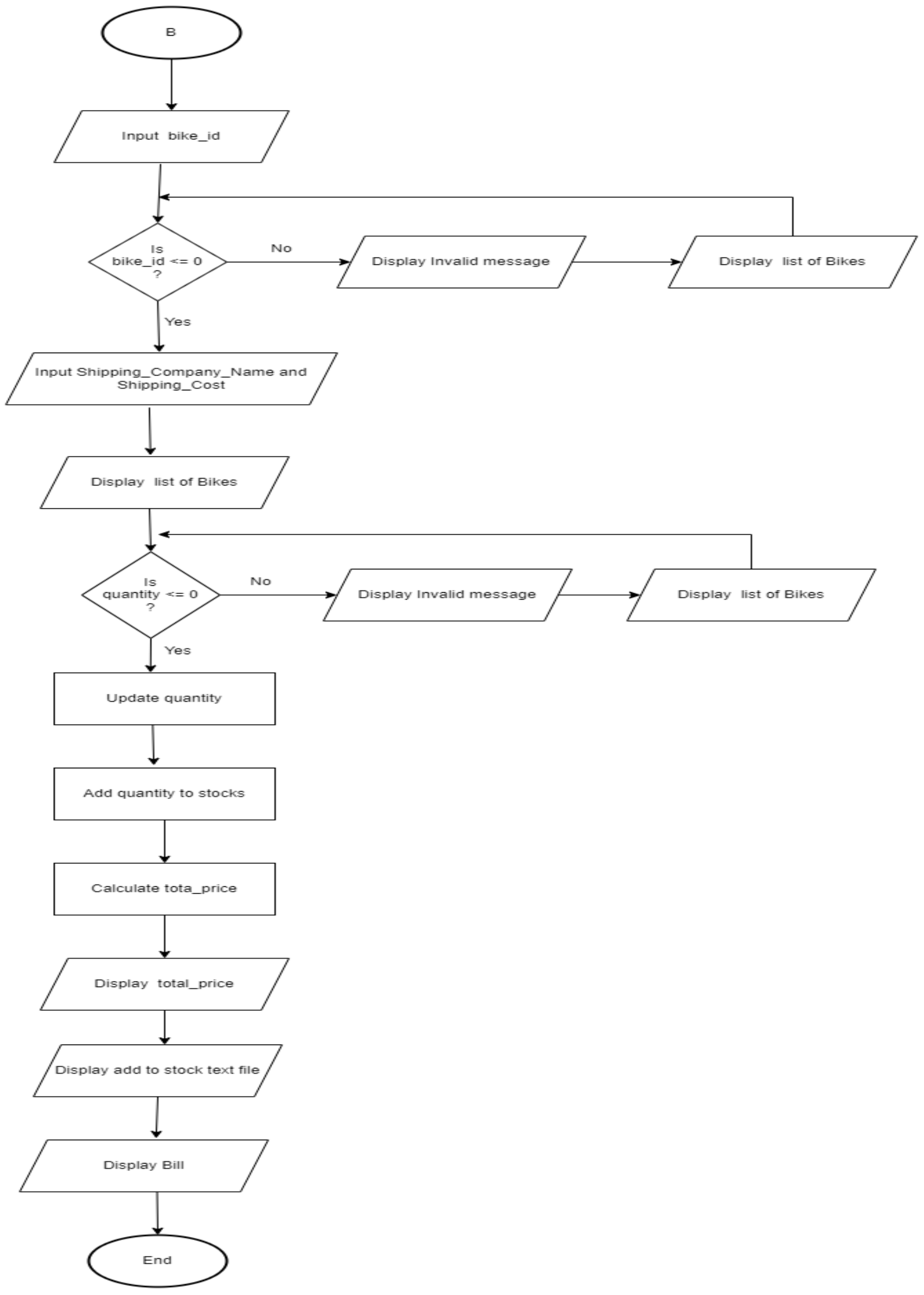


Figure 3: Flowchart of adding stocks of bikes

4. Pseudo Code

Pseudocode is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally styled natural language rather than in a programming language (Anon., 2020).

Pseudo Code of Bike Management System

```
DEFINE FUNCTION Welcome_to_program():

   OUTPUT("++++++++++++++++++++++++++"+" Welcome to the program "+"
++++++++++++++++++++++++++++++++++++++++++++++++++++")


DEFINE FUNCTION open_text_file():

   SET file TO open("Bikes.txt","r")

   RETURN file


DEFINE FUNCTION display_bike():


   OUTPUT("| Bike_ID \t|  Bike_Name \t\t| Company_Name \t\t|  Colour \t\t|  Quantity \t|
Price ")

   SET file TO open_text_file()

   SET id TO 1

   FOR line IN file:

      OUTPUT(id,"\t\t | "+ line.replace(",","\t\t | "))

      id += 1

   OUTPUT("\n\n")

   file.close()
```

```
DEFINE FUNCTION bike_to_2D_list():
        SET read_file TO open_text_file()
        SET My_list TO []
        FOR i IN read_file:
            SET i TO i.replace("\n", "")
            My_list.append(i.split(","))
        RETURN(My_list)

DEFINE FUNCTION user_for_operation():
    OUTPUT("++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++")
    OUTPUT("++++++++++ Enter 2 to add stock:            ++++++++++++++++++")
    OUTPUT("++++++++++ Enter 3 to exit:                 ++++++++++++++++++")


DEFINE FUNCTION user_INPUT():
     TRY:
        SET loop TO True
        WHILE loop EQUALS True:
         user_for_operation()
         SET user_INPUT TO int(INPUT("Enter the number: "))
         SET DT TO datetime.datetime.now()
         SET T TO DT.strftime("%H:%M:%S")
         SET D TO DT.strftime("%D/%M/%Y")
         SET S TO 0
```

```
IF user_INPUT EQUALS 1:
  SET Name TO INPUT("Enter your name :")
  SET Address TO INPUT("Enter address of customer: ")
  SET Contact TO INPUT("Enter contact number: ")
  SET the_bike_id TO validating_bike_id()
  OUTPUT("\n")
  SET the_q TO quantity_validation(the_bike_id)
  Sell(the_bike_id, the_q)
  OUTPUT("\n")
  SET total_price TO the_price(the_bike_id, the_q)
  OUTPUT(total_price)
  SET amount TO int(INPUT("Enter total amount paid by customer:"))
  Bill(the_bike_id, T, D, Name, str(amount), Address, Contact, str(the_q),
Customer_Detail)
ELSEIF user_INPUT EQUALS 2:
  display_bike()
  SET Name1 TO INPUT("Enter your name: ")
  SET the_bike_id TO validating_bike_id()
  SET Company_Details TO Company_Detail()
  SET Stock TO int(INPUT("Enter number of stocks: "))
  Stock_checking(Stock)
  Adding_stock(the_bike_id, Stock)
  OUTPUT("\n")
```

```
            SET cost TO int(INPUT("Enter importing cost: "))
            SellBill(the_bike_id, T, D, Name1, str(cost), str(Stock),Company_Details)
            bike_to_stock()
          ELSEIF user_INPUT EQUALS 3:
            exit_system()
            SET loop TO False
          ELSE:
            invalid_user_INPUT()
       except ValueError:
          OUTPUT("Please give us proper informations.")


DEFINE FUNCTION validating_bike_id():
    SET loop TO True
    WHILE loop:
        TRY:
            OUTPUT("\n\n")
            SET valid_id TO int(INPUT("Enter ID of bike you want to buy:"))
            OUTPUT("\n\n")
            WHILE valid_id <= 0 or valid_id > len(bike_to_2D_list()):
                OUTPUT("\n\n")

                OUTPUT("Please enter  valid Bike ID !")

                display_bike()
            SET Valid_id TO int(INPUT("Enter Id of the bike you want to buy: "))

         RETURN valid_id
      except ValueError:
         OUTPUT("\n\n")
         OUTPUT("please give integer value only! ")
         OUTPUT("\n\n")

DEFINE FUNCTION bike_to_stock():

   OUTPUT("The Bike has been added to the stock")

DEFINE FUNCTION purchasing_bike():

   OUTPUT("Thank you FOR purchasing the bike")

DEFINE FUNCTION invalid_user_INPUT():

   OUTPUT("Invalid INPUT!!!")
   OUTPUT("Please provide value as 1,2 or 3.")

DEFINE FUNCTION exit_system():

   OUTPUT("Thank you FOR using our system")

DEFINE FUNCTION Customer_Detail():
    SET Name TO INPUT("Enter your name : ")

    SET Address TO INPUT("Enter your address : ")

    SET Contact TO INPUT("Enter your contact number : ")

    RETURN Name,Address,Contact

DEFINE FUNCTION quantity_validation(the_bike_id):

    SET loop TO True

    WHILE loop:

        TRY:

            SET bike_list TO bike_to_2D_list()

            SET user_quantity TO int(INPUT("Enter the quantity you want to purchase: "))

            WHILE user_quantity <= 0 or user_quantity > int(bike_list[the_bike_id - 1][3]):


                OUTPUT("Please provide a  valid Qunatity ID !!!")


                display_bike()

                SET user_quantity TO int(INPUT("Enter the quantity you want to purchase: "))

                OUTPUT("\n")

            RETURN user_quantity

        except ValueError:

            OUTPUT("\n" "Only supports integer values!  " "\n")

DEFINE FUNCTION update_quantity(bike_list):

    SET file TO open("Bikes.txt","w")
```

```
   FOR i IN bike_list:
      file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) + "\n")
   file.close()
   display_bike()

DEFINE FUNCTION Sell(the_bike_id,the_q):
   SET bike_list TO bike_to_2D_list()
   SET bike_list[the_bike_id -1][3] TO int(bike_list[the_bike_id - 1][3]) - the_q
   update_quantity(bike_list)


DEFINE FUNCTION Update_stock(bike_list):
      SET file TO open("Bikes.txt", "w")
      FOR i IN bike_list:
         file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) +
"\n\n")
         file.close()
         display_bike()

DEFINE FUNCTION Stock_checking(Stock):
   WHILE Stock <= 0:
      OUTPUT("Enter valid quantity of stocks.")
      SET Stock TO int(INPUT("Enter number of stock to add"))
      display_bike()
   RETURN Stock
```

```
DEFINE FUNCTION Update_stock1(bike_list1):
        SET file TO open("Bikes.txt", "w")
        FOR i IN bike_list1:
            file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) +
"\n\n")
        file.close()
        display_bike()

DEFINE FUNCTION the_price(the_bike_id,the_q):
    SET bike_list TO bike_to_2D_list()
    SET total_price TO int(bike_list[the_bike_id - 1][4].replace("$","")) * int(the_q)
    OUTPUT("Total price is: : ", total_price)
    OUTPUT("\n")
    RETURN total_price


DEFINE FUNCTION Adding_stock(the_bike_id1,the_q1):
    SET bike_list TO bike_to_2D_list()
    SET bike_list[the_bike_id1 -1][3] TO int(bike_list[the_bike_id1 - 1][3]) + the_q1
    update_quantity(bike_list)


DEFINE FUNCTION Bill(the_bike_id, T, D, Name, amount, Address, Contact, the_q,
Customer_Details):
    SET bike_list TO bike_to_2D_list()
    SET the_price TO int(bike_list[the_bike_id-1][4].replace("$", ""))*the_q
```

```
with open("Purchase- " + Name + ".txt", "w+") as f:
    f.write("+++++++++++++++++++++++++++++++++++++++++++++++++++   \n\n")
    f.write("      Bike Management System               \n\n")
    f.write(" Name of Customer: " + Name + "         \n\n")
    f.write(" Address of Customer: " + Address + "       \n\n")
    f.write(" Contact number of Customer: " + Contact + " \n\n")
    f.write("Time of bike purchased is: " + T + "\n\n")
    f.write("Date of bike purchased is: " + D + "\n\n")
    f.write("Purchased Bike is: " + bike_list[the_bike_id - 1][0] + "\n\n")
    f.write("Purchased Bike Company is: " + bike_list[the_bike_id - 1][1] + "\n\n")
    f.write("Color of bike is: " + bike_list[the_bike_id - 1][2] + the_q + "\n\n")
    f.write("Quantity of bike is: " + the_q + "\n\n")
    f.write("Amount: " + amount +" \n\n")
    f.write("Total price is: " + the_price + "\n\n")


    SET display_file TO open("Purchase- " + Name + ".txt", "r")
    FOR output IN display_file:
        OUTPUT(output)
    purchase_more_bikes(the_bike_id, T, D, Name, amount, Address, Contact, the_q)
    user_for_operation()
```

```
DEFINE FUNCTION Bills(the_bike_id, T, D, Name, amount, Address, Contact, the_q,
Customer_Details):

    SET bike_list TO bike_to_2D_list()

    SET the_price1 TO int(bike_list[the_bike_id-1][4].replace("$", ""))*the_q


    with open("Purchase- " + Name + Contact+ Address + ".txt", "w+") as f:

        f.write("++++++++++++++++++++++++++++++++++++++++++++++++++++   \n\n")

        f.write("      Bike Management System               \n\n")

        f.write(" Name of Customer: " + Name + "         \n\n")

        f.write(" Address of Customer: " + Address + "      \n\n")

        f.write(" Contact number of Customer: " + Contact + " \n\n")

        f.write("Time of bike purchased is: " + T + "\n\n")

        f.write("Date of bike purchased is: " + D + "\n\n")

        f.write("Purchased Bike is: " + bike_list[the_bike_id -1][0] + "\n\n")

        f.write("Purchased Bike Company is: " +

                bike_list[the_bike_id -1][1] + "\n\n")

        f.write("Color of bike is: " +

                bike_list[the_bike_id -1][2] + the_q + "\n\n")

        f.write("Quantity of bike is: " + the_q + "\n\n")

        f.write("Total price is: " + the_price1 + "\n\n")


        #display of bill

        SET display_file TO open("Purchase- " + Name + ".txt", "a")

        FOR output IN display_file:
            OUTPUT(output)

        purchase_more_bikes(the_bike_id, T, D, Name, amount, Address, Contact, the_q)

        user_for_operation()



DEFINE FUNCTION purchase_more_bikes(the_bike_id, T, D, Name, amount, Address,
Contact, the_q):

    SET loop TO True

    WHILE loop:

        TRY:

            OUTPUT("Type Yes to Continue or No to Cancel.")

            SET user_quantity TO INPUT("Do you want to purchase another bike?: .")

            OUTPUT("\n\n")

            IF user_quantity.upper() EQUALS "Yes":

                SET the_bike_id TO validating_bike_id()

                display_bike()

                SET the_q TO quantity_validation(the_bike_id)

                sell(the_bike_id, the_q)

                SET the_price1 TO total_price(the_bike_id, the_q)

                Bill(the_bike_id, T, D, Name, amount, Address, Contact)

                OUTPUT("Total price of the bike is: ", the_price1)

                OUTPUT("\n\n")

            ELSE:

                user_quantity.upper() EQUALS "No"
            exit_system()

            SET loop TO False


        except ValueError:

            OUTPUT("\n\n", " Supports Yes/No Only! " "\n\n")

DEFINE FUNCTION validating_bike_id1():

    SET loop TO True

    WHILE loop:

        TRY:

            OUTPUT("\n")

            SET valid_id TO int(INPUT("Enter  id of bike you want to purchase: "))

            OUTPUT("\n")

            WHILE valid_id <= 0 or valid_id > len(bike_to_2D_list()):

                OUTPUT("  Please give a valid Bike_ID     ")

                OUTPUT("\n")

                display_bike()

                OUTPUT("\n")

                SET valid_id TO int(INPUT("Enter id of bike you want to purchase: "))

                OUTPUT("\n")

            RETURN valid_id

        except ValueError:

            OUTPUT("  Only supports integer values!  ")
```

```
DEFINE FUNCTION Company_Detail():
    SET Shipping_Company_name TO INPUT("Enter shipping company name: ")
    SET Shipping_Cost TO INPUT("Enter the shipping cost of the bikes: ")
    RETURN Shipping_Company_name,Shipping_Cost

DEFINE FUNCTION quantity_validation1(the_bike_id1):
    SET loop TO True
    WHILE loop:
        TRY:
            SET bike_list TO bike_to_2D_list()
            SET user_quantity TO int(INPUT("Enter quantity you want to add: "))
            WHILE user_quantity <= 0:
                OUTPUT("\n")
                OUTPUT("  Please give us valid quantity. ")
                OUTPUT("\n")
                display_bikes()
                SET user_quantity TO int(INPUT("Enter quantity you want to add: "))
                OUTPUT("\n")
            RETURN user_quantity
        except ValueError:
            OUTPUT("  Only supports integer values! ")


DEFINE FUNCTION the_price1(the_bike_id1,the_q1):
    SET bike_list TO bike_to_2D_list()

    SET total_price TO int(bike_list[the_bike_id1 - 1][4].replace("$","")) * int(the_q1)

    RETURN total_price


DEFINE FUNCTION SellBill(the_bike_id, T, D, Name1, cost, Stock, Company_Detail):

    SET bike_list1 TO bike_to_2D_list()

    SET Year TO str(datetime.datetime.now().year)

    SET Month TO str(datetime.datetime.now().month)

    SET Day TO str(datetime.datetime.now().day)

    SET Second TO str(datetime.datetime.now().second)



    with open("AddedStocks-" + Name1 + Year+""+Month+""+Day+""+Second+".txt",
"w+") as f:

        f.write("++++++++++++++++++++++++++++++++++++++++++++++   \n\n")

        f.write("     Bike Management System             \n\n")

        f.write("   Name of Shipping Company: "+Name1+"   \n\n")

        f.write(" Shipping Cost is: "+cost+"             \n\n")

        f.write(" Added bike stock is: "+T+"             \n\n")

        f.write(" Time of Stocks added is: "+T+"          \n\n")

        f.write(" Date of stocks added is: "+D+"          \n\n")

        f.write("Added stocks : " + str(Stock))

        f.write(" Stocked Bike Name is: " + bike_list1[the_bike_id - 1][0] + "\n\n")

        f.write(" Stocked Bike Company is: " + bike_list1[the_bike_id - 1][1] + "\n\n")

        f.write(" Stocked Bike Color is: " + bike_list1[the_bike_id - 1][2] + "\n\n")
        f.write(" Stocked Bike Price is: " + bike_list1[the_bike_id - 1][4] + "\n\n")


DEFINE FUNCTION exit_system():
    OUTPUT("\n\n")
    OUTPUT("++++++++++++++++++++++++++++++++++++++++++++++++++++")
    OUTPUT("Thank you FOR using our system")
    OUTPUT("++++++++++++++++++++++++++++++++++++++++++++++++++++")
    OUTPUT("\n\n")

DEFINE FUNCTION invalid_Input():
    OUTPUT("\n\n")
    OUTPUT("++++++++++++++++++++++++++++++++++++++++++++++++++++")
    OUTPUT("Invalid INPUT!!")
    OUTPUT(" Please give valid number! ")
    OUTPUT("++++++++++++++++++++++++++++++++++++++++++++++++++++")
    OUTPUT("\n\n")
Welcome_to_program()
display_bike()
user_INPUT()
```

5. Data Structure

A data structure is a logical way of organizing data in computer memory so that it can be used effectively (Anon., 2022). Data structures enable you to arrange your data in a way that enables you to store groups of data, relate them, and perform operations on them.

5.1 List

A list is defined as an ordered collection of items, and it is one of the essential data structures when using Python to create a project (Anon., 2021). It is utilised to simultaneously store a significant amount of data in []. The list data type in Python is the most flexible and can be expressed as a list of comma-separated values (items) enclosed in square brackets.

Eg: Bike_list= [1,2,3]

5.2 2-D Array

Two-dimensional array is an array within an array (Anon., 2021). This kind of array uses two indices rather than one to identify a data element's position. As a result, it represents a table with data organised into rows and columns. For the sake of convenience, economy, and effectiveness, we prefer to save the 2D data list. Our code was made clearer by using a 2-dimensional list. Due to this characteristic, we used data collection for 2-dimensional list storage so that we could implement our analyses. Debugging is quicker and easier. We can easily add and remove items from the two-dimensional list due to the flexibility of the data type.

For example,

Bike_list [Pulsar T20, Bajaj, Dark Blue, 20, $4000

CRF 300, Honda Bikes, Red White, 18, $14000

KTM Duke 250, KTM Bikes, Black Orange, 18, $3700]

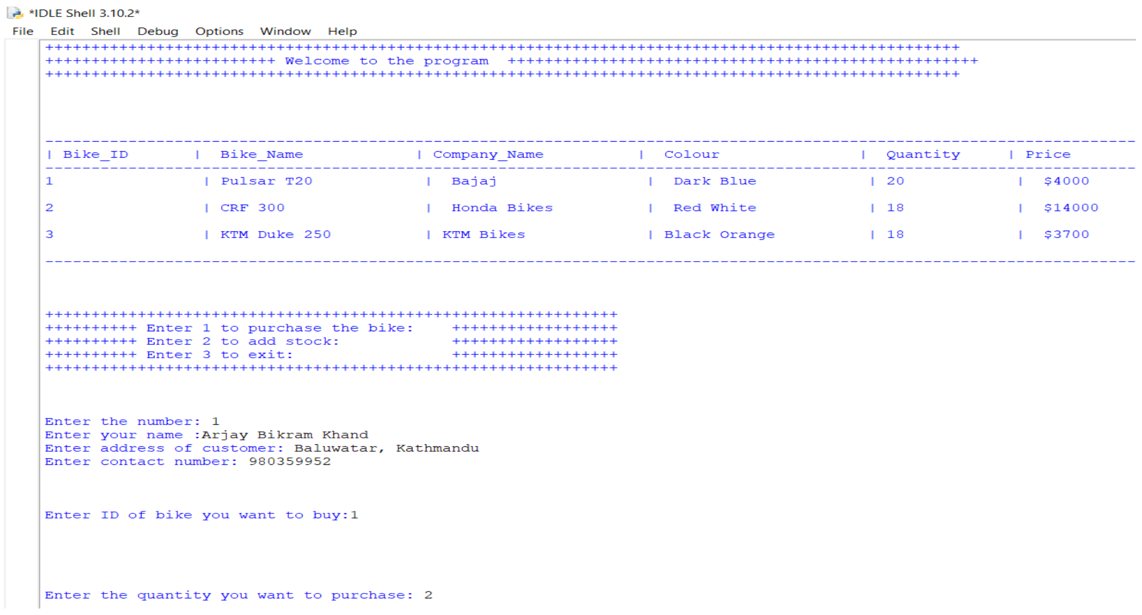
6. Program

6.1 Implementation of Program

The system's fundamental software module is this bike management system. The task of the programme is finished by this module. The program's initial options include purchasing motorcycles from the business or showroom. The second option is to give retailers access to dealer inventories of motorcycles. The system's module exit system is contained in the final selection.

6.2 Show selling of bike and adding the bikes in the stock

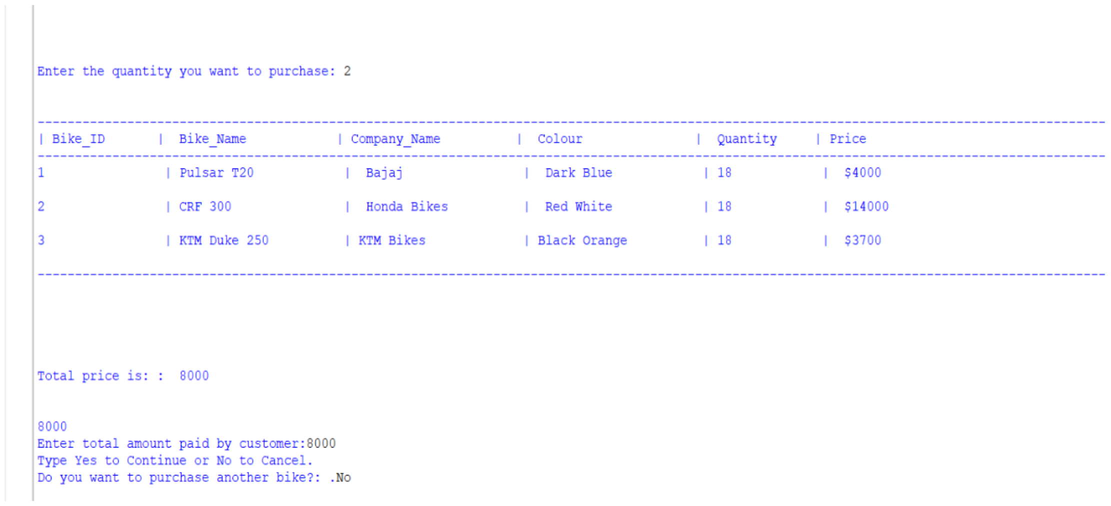
The Bike Management System, which consist of selling of the bikes to the customer and bike adding into the stocks of the showroom.



```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++ Welcome to the program  +++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



-----------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name             | Company_Name           |  Colour                |  Quantity      | Price
-----------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20             |  Bajaj                 |  Dark Blue             | 20             |  $4000

2                 | CRF 300                |  Honda Bikes           |  Red White             | 18             |  $14000

3                 | KTM Duke 250           | KTM Bikes              | Black Orange           | 18             |  $3700

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


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter the number: 1
Enter your name :Arjay Bikram Khand
Enter address of customer: Baluwatar, Kathmandu
Enter contact number: 980359952


Enter ID of bike you want to buy:1



Enter the quantity you want to purchase: 2
```

Figure 4: Selling of bike in program

The list of options and the list of motorcycles are displayed initially in order for the consumer to pick the choice desired. Following that, the number or choice is chosen. Following that, the customer's information was obtained. In addition, the id of the bike that the consumer wishes to purchase has been entered. The bike's quantity has also been entered. Moreover, the number of motorcycles purchased has decreased from the first list.



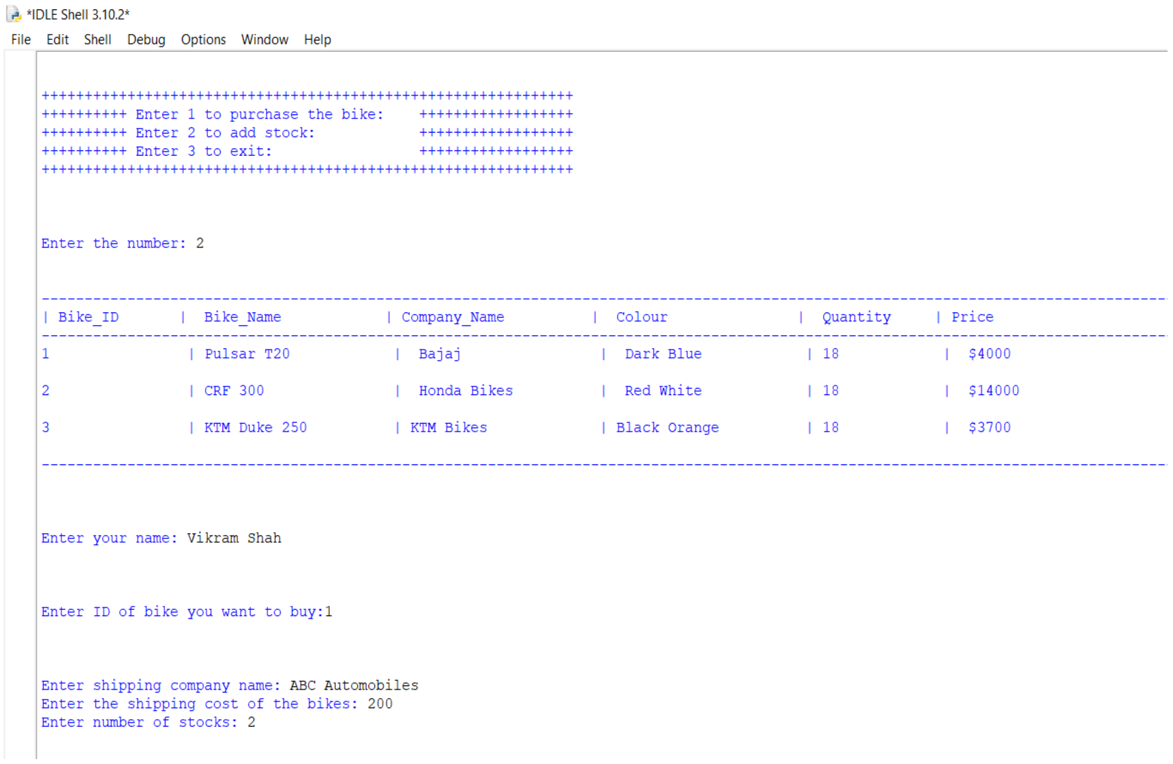
```
Enter the quantity you want to purchase: 2


------------------------------------------------------------------------------------------------------------------------------------------
| Bike_ID       |  Bike_Name            | Company_Name          |  Colour               |  Quantity     | Price
------------------------------------------------------------------------------------------------------------------------------------------
1                | Pulsar T20           |  Bajaj                |  Dark Blue            | 18            |  $4000

2                | CRF 300              |  Honda Bikes          |  Red White            | 18            |  $14000

3                | KTM Duke 250         | KTM Bikes             | Black Orange          | 18            |  $3700

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




Total price is: :  8000


8000
Enter total amount paid by customer:8000
Type Yes to Continue or No to Cancel.
Do you want to purchase another bike?: .No
```

Figure 5: Selling of bike program 2

The total cost of the motorcycle as well as the full amount the customer had paid was then output. It then asks whether you want to buy more motorcycles. The programme is replayed if the yes checkbox is selected; the system exits if the no checkbox is selected.

In addition, the system's other option, which would add bike inventories to the showroom, is currently being programmed.

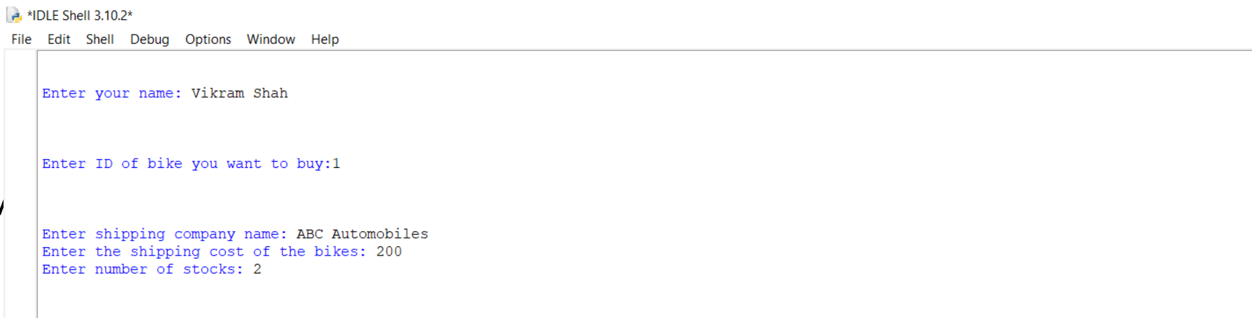
The list of bikes and available options is shown in the above figure. To add the bike stocks, a number has been entered. It is being done to enter the customer's name. The bike that is being added to stock has been given a bike ID. The total number of bike stocks was then increased. The list of bikes with newly added stock is also displayed.



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:     ++++++++++++++++++
++++++++++ Enter 2 to add stock:             ++++++++++++++++++
++++++++++ Enter 3 to exit:                  ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter the number: 2

-----------------------------------------------------------------------------------------------------------------------------------
| Bike_ID         |  Bike_Name              | Company_Name          |  Colour                 |  Quantity      | Price
-----------------------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20              |  Bajaj                |  Dark Blue              | 18             |  $4000

2                 | CRF 300                 |  Honda Bikes          |  Red White              | 18             |  $14000

3                 | KTM Duke 250            | KTM Bikes             | Black Orange            | 18             |  $3700

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


Enter your name: Vikram Shah


Enter ID of bike you want to buy:1


Enter shipping company name: ABC Automobiles
Enter the shipping cost of the bikes: 200
Enter number of stocks: 2
```

Figure 6: Adding stock of bike

The bicycle's importation costs are then entered. The list of options and bikes has also been re-displayed.



```
Enter your name: Vikram Shah


Enter ID of bike you want to buy:1


Enter shipping company name: ABC Automobiles
Enter the shipping cost of the bikes: 200
Enter number of stocks: 2
```

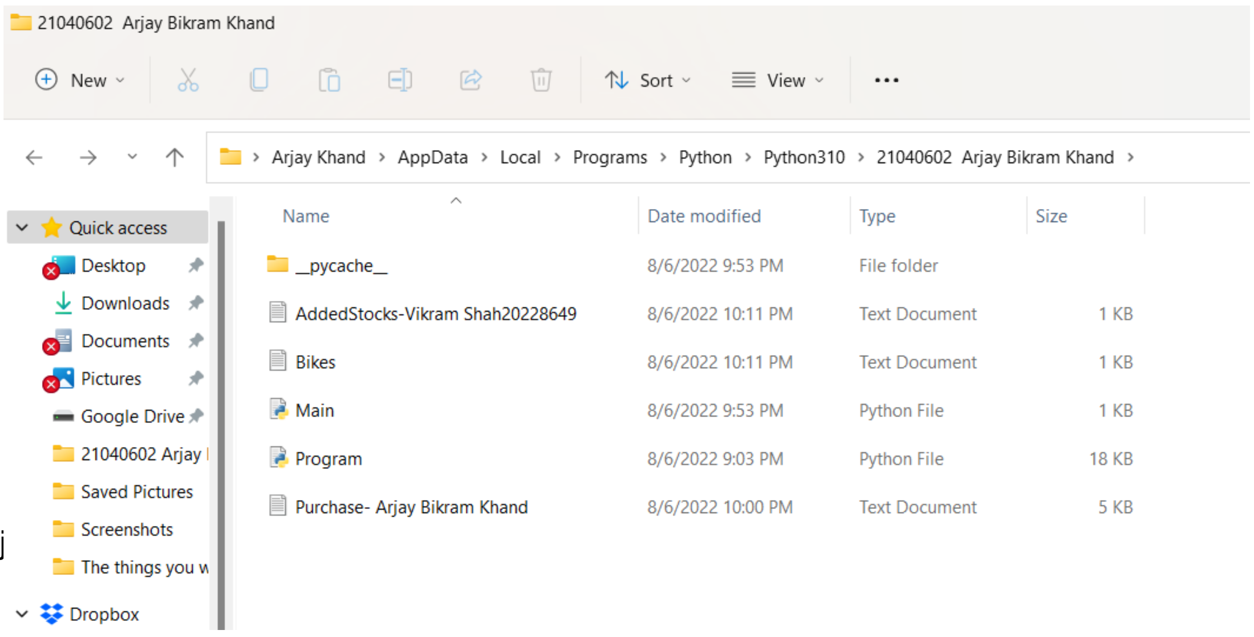
Figure 7: Adding stock of bike 2

6.3 Creation of text file

According to the system and programme, the text file for purchasing and adding stock of the bike is automatically generated in the folder containing the text file for the bike and the Python programme code.

\

Arj



21040602 Arjay Bikram Khand

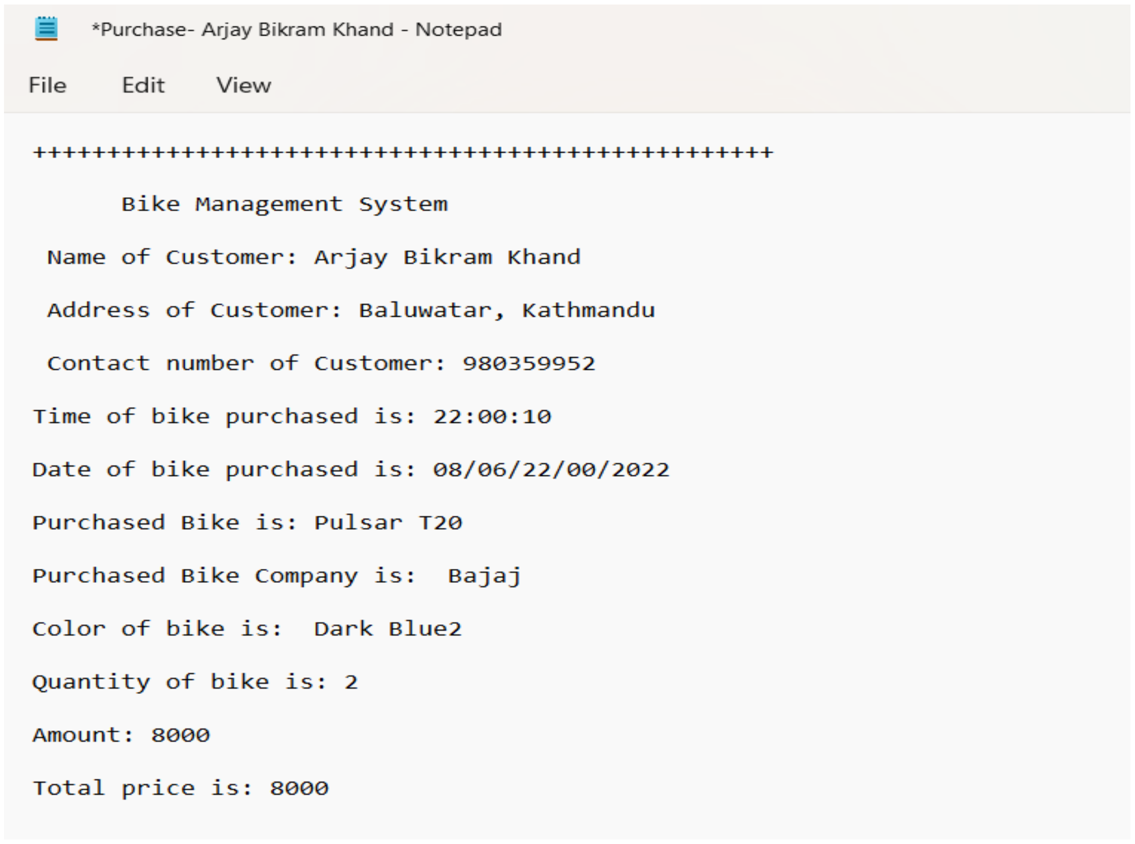
Arjay Khand › AppData › Local › Programs › Python › Python310 › 21040602 Arjay Bikram Khand ›

Name	Date modified	Type	Size
__pycache__	8/6/2022 9:53 PM	File folder	
AddedStocks-Vikram Shah20228649	8/6/2022 10:11 PM	Text Document	1 KB
Bikes	8/6/2022 10:11 PM	Text Document	1 KB
Main	8/6/2022 9:53 PM	Python File	1 KB
Program	8/6/2022 9:03 PM	Python File	18 KB
Purchase- Arjay Bikram Khand	8/6/2022 10:00 PM	Text Document	5 KB

Figure 8: Creation of text file

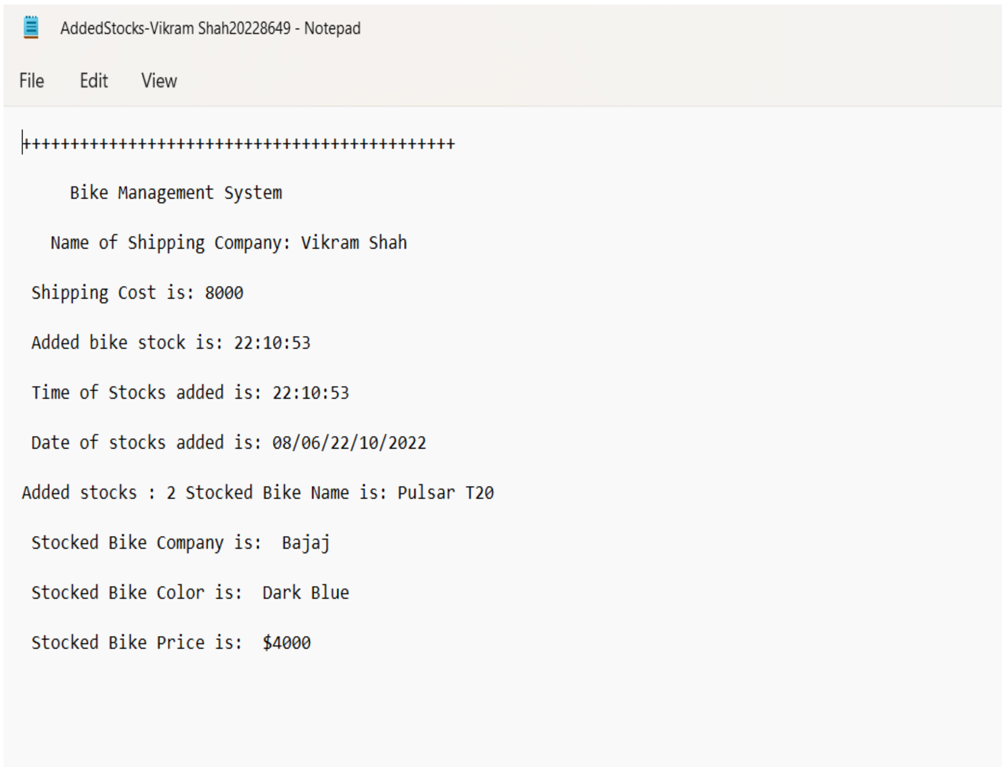
6.4 Opening text file and show the bill

The folder's text file containing the bike's text and Python code is automatically generated. The bill is automatically displayed as soon as the data or information is input. The bill of sale as well as a text file are also shown.



```
++++++++++++++++++++++++++++++++++++++++++++++++++

      Bike Management System

 Name of Customer: Arjay Bikram Khand

 Address of Customer: Baluwatar, Kathmandu

 Contact number of Customer: 980359952

Time of bike purchased is: 22:00:10

Date of bike purchased is: 08/06/22/00/2022

Purchased Bike is: Pulsar T20

Purchased Bike Company is:  Bajaj

Color of bike is:  Dark Blue2

Quantity of bike is: 2

Amount: 8000

Total price is: 8000
```

Figure 9: Display bill of purchase



AddedStocks-Vikram Shah20228649 - Notepad

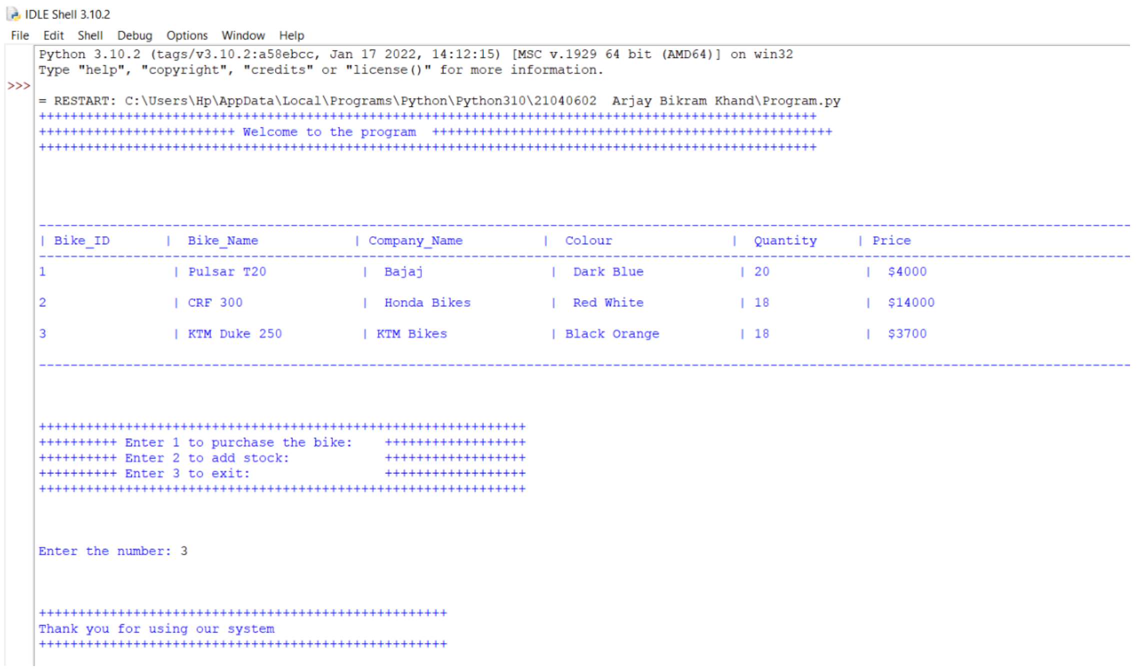
File Edit View

```
+++++++++++++++++++++++++++++++++++++++++++++

     Bike Management System

   Name of Shipping Company: Vikram Shah

 Shipping Cost is: 8000

 Added bike stock is: 22:10:53

 Time of Stocks added is: 22:10:53

 Date of stocks added is: 08/06/22/10/2022

Added stocks : 2 Stocked Bike Name is: Pulsar T20

 Stocked Bike Company is:  Bajaj

 Stocked Bike Color is:  Dark Blue

 Stocked Bike Price is:  $4000
```

Figure 10: Display bill of Added Stocks

6.5 Termination of program after selecting an option

Option 3 on the list of options is the one that will end the programme. It is utilised to end or exit the system.



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Hp\AppData\Local\Programs\Python\Python310\21040602  Arjay Bikram Khand\Program.py
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++ Welcome to the program  ++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



-------------------------------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name             | Company_Name           |  Colour                 |  Quantity       | Price
-------------------------------------------------------------------------------------------------------------------------------------------
1                | Pulsar T20             |  Bajaj                 |  Dark Blue              | 20              |  $4000

2                | CRF 300                |  Honda Bikes           |  Red White              | 18              |  $14000

3                | KTM Duke 250           | KTM Bikes              | Black Orange            | 18              |  $3700

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



++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



Enter the number: 3



++++++++++++++++++++++++++++++++++++++++++++++++++++
Thank you for using our system
++++++++++++++++++++++++++++++++++++++++++++++++++++
```

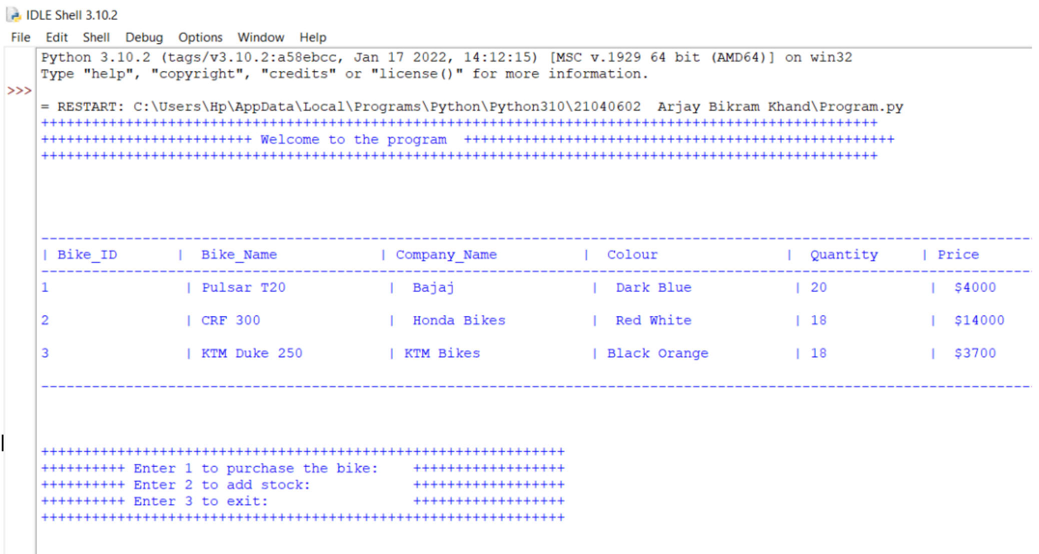
Figure 11: Termination of program

7. Testing

7.1 Test 1

Test 1	1
Action	Giving string variable in the place to input Bike ID.
Expected Result	An invalid input message will displayed.
Actual Result	The invalid input message is displayed.
Test Result	Successful.

Table 1: Test 1



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Hp\AppData\Local\Programs\Python\Python310\21040602  Arjay Bikram Khand\Program.py
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++ Welcome to the program  ++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



----------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name              | Company_Name            |  Colour                 |  Quantity      | Price
----------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20              |  Bajaj                   |  Dark Blue               | 20              |  $4000

2                 | CRF 300                 |  Honda Bikes             |  Red White               | 18              |  $14000

3                 | KTM Duke 250            | KTM Bikes                | Black Orange             | 18              |  $3700

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



++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

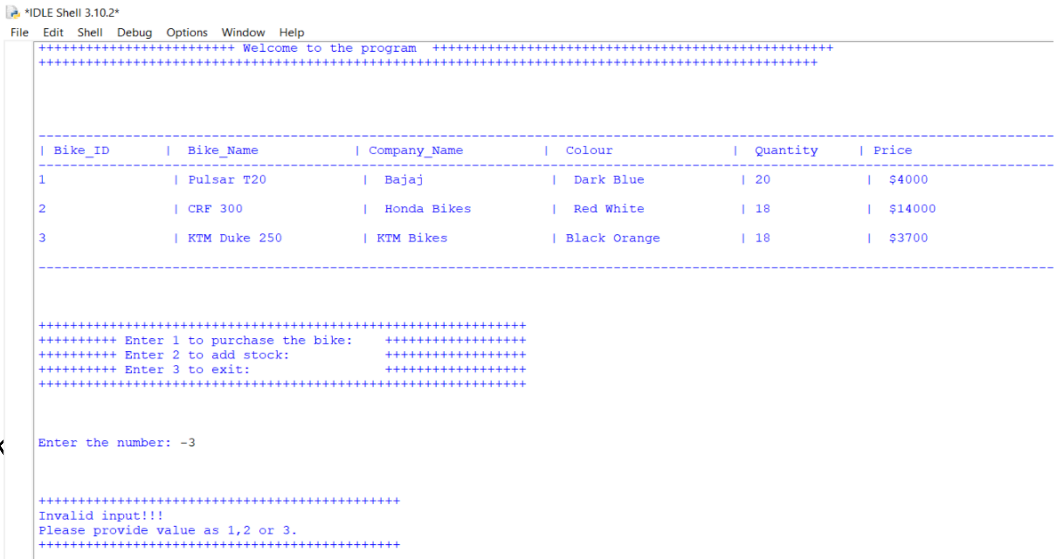
Figure 12: Screenshot of Test 1

7.2 Test 2

Adding bikes in stock

Test 2.1	2.1
Action	Providing as negavtive value as input
Expected Result	Input valid quantity message will displayed.
Actual Result	To input valid quantity is displayed.
Test Result	Successful.

Table 2: Test 2.1



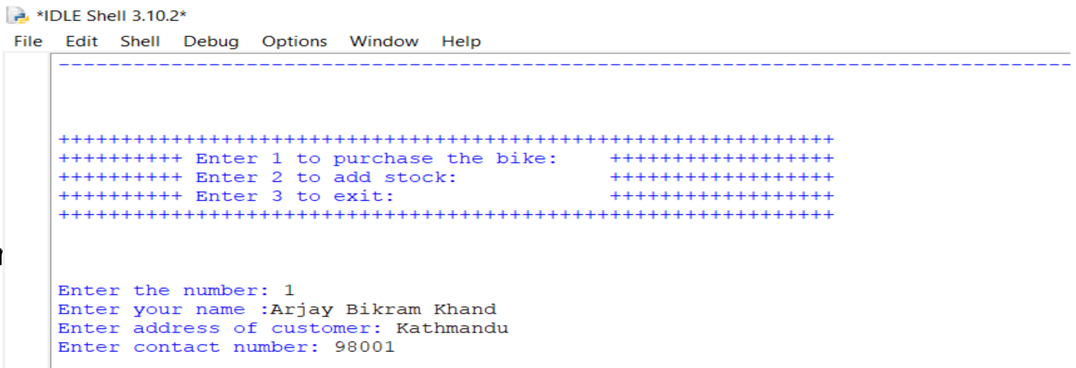
```
++++++++++++++++++++++++++ Welcome to the program  +++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



-------------------------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name              | Company_Name           |  Colour                 |  Quantity      | Price
-------------------------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20             |  Bajaj                 |  Dark Blue              | 20             |  $4000

2                 | CRF 300                |  Honda Bikes           |  Red White              | 18             |  $14000

3                 | KTM Duke 250           | KTM Bikes              | Black Orange            | 18             |  $3700

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


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter the number: -3


++++++++++++++++++++++++++++++++++++++++++++++++
Invalid input!!!
Please provide value as 1,2 or 3.
++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 13: Test 2.1

Selling of bikes

Test 2.2	2.2
Action	Providing negative and non-existed value as input
Expected Result	Input valid Id message will displayed.
Actual Result	To input valid ID is displayed.
Test Result	Successful.

Table 3: Test 2.2



```
*IDLE Shell 3.10.2*
File  Edit  Shell  Debug  Options  Window  Help
--------------------------------------------------------------------------------


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter the number: 1
Enter your name :Arjay Bikram Khand
Enter address of customer: Kathmandu
Enter contact number: 98001
```

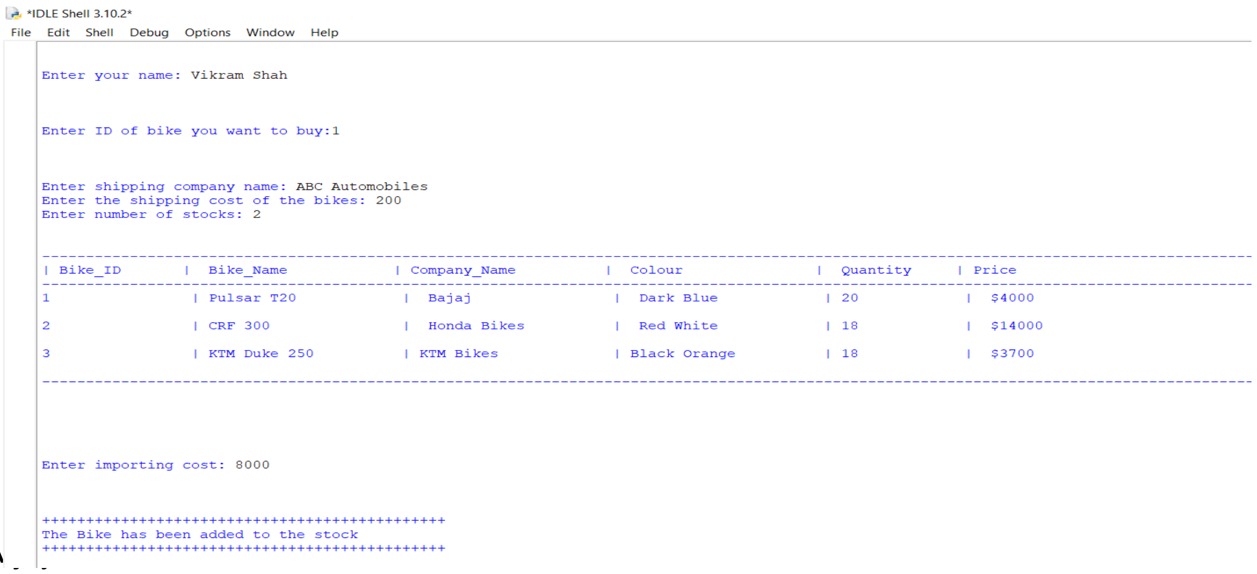
Figure 14: Test 2.2

7.3 Test 3

File generation of adding bikes in stock

Test 3	3
Action	Adding bikes in stock
Expected Result	It will take the information and display the bill in text file.
Actual Result	The text file is being automatically generated in folder of code.
Test Result	Successful.

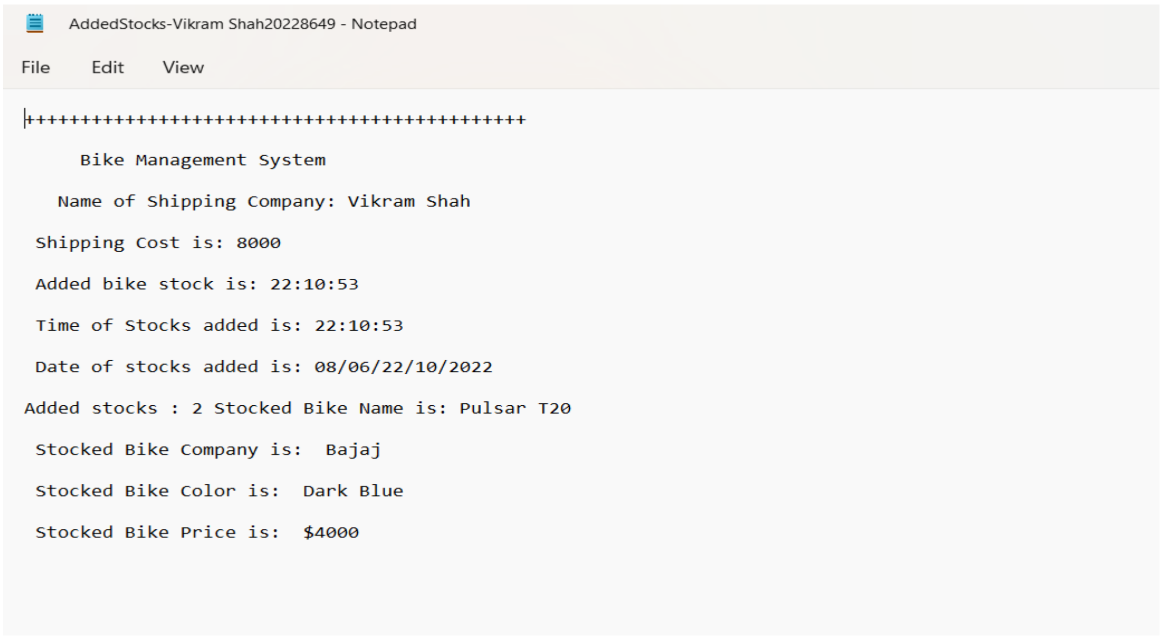
Table 4: Test 3



```
Enter your name: Vikram Shah


Enter ID of bike you want to buy:1


Enter shipping company name: ABC Automobiles
Enter the shipping cost of the bikes: 200
Enter number of stocks: 2


-------------------------------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name              | Company_Name            |  Colour                  |  Quantity      | Price
-------------------------------------------------------------------------------------------------------------------------------------------
1                | Pulsar T20              |  Bajaj                  |  Dark Blue               | 20             |  $4000

2                | CRF 300                 |  Honda Bikes            |  Red White               | 18             |  $14000

3                | KTM Duke 250            | KTM Bikes               | Black Orange             | 18             |  $3700

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




Enter importing cost: 8000


+++++++++++++++++++++++++++++++++++++++++++++++
The Bike has been added to the stock
+++++++++++++++++++++++++++++++++++++++++++++++
```

A

Figure 15: Test 3



AddedStocks-Vikram Shah20228649 - Notepad

File Edit View

```
++++++++++++++++++++++++++++++++++++++++++++++

     Bike Management System

   Name of Shipping Company: Vikram Shah

 Shipping Cost is: 8000

 Added bike stock is: 22:10:53

 Time of Stocks added is: 22:10:53

 Date of stocks added is: 08/06/22/10/2022

Added stocks : 2 Stocked Bike Name is: Pulsar T20

 Stocked Bike Company is:  Bajaj

 Stocked Bike Color is:  Dark Blue

 Stocked Bike Price is:  $4000
```

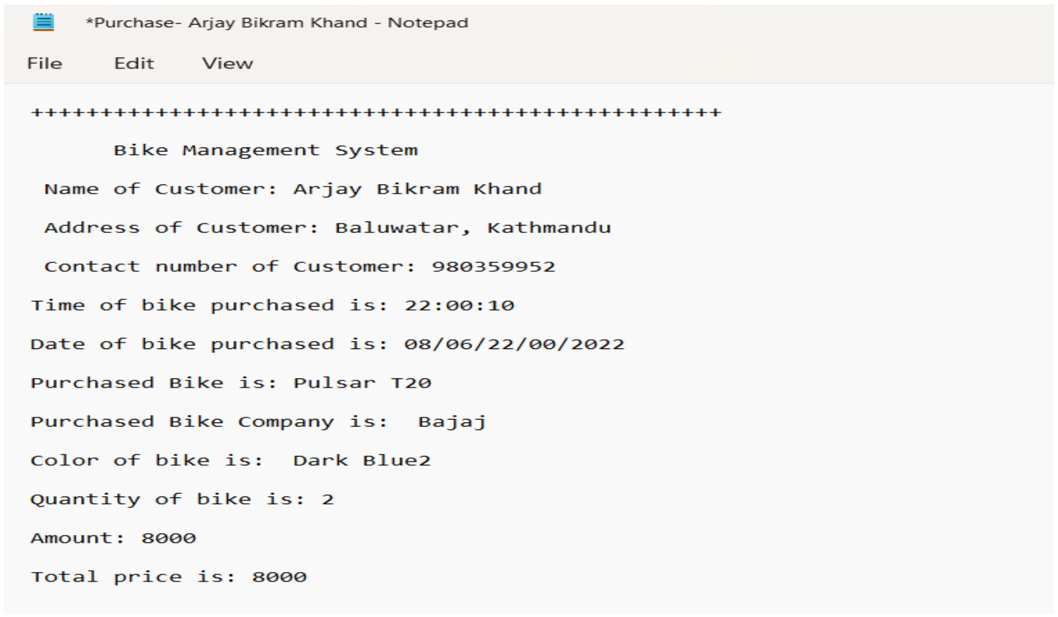
Figure 16: Display of bill of added stock in text file

7.4 Test 4

File generation of selling of bikes

Test 4	4
Action	Purchasing bikes
Expected Result	It will take the information and display the bill in text file.
Actual Result	The text file is being automatically generated in folder of code.
Test Result	Successful.

Table 5: Test 4



```
++++++++++++++++++++++++++++++++++++++++++++++++++
      Bike Management System
 Name of Customer: Arjay Bikram Khand
 Address of Customer: Baluwatar, Kathmandu
 Contact number of Customer: 980359952
Time of bike purchased is: 22:00:10
Date of bike purchased is: 08/06/22/00/2022
Purchased Bike is: Pulsar T20
Purchased Bike Company is:  Bajaj
Color of bike is:  Dark Blue2
Quantity of bike is: 2
Amount: 8000
Total price is: 8000
```

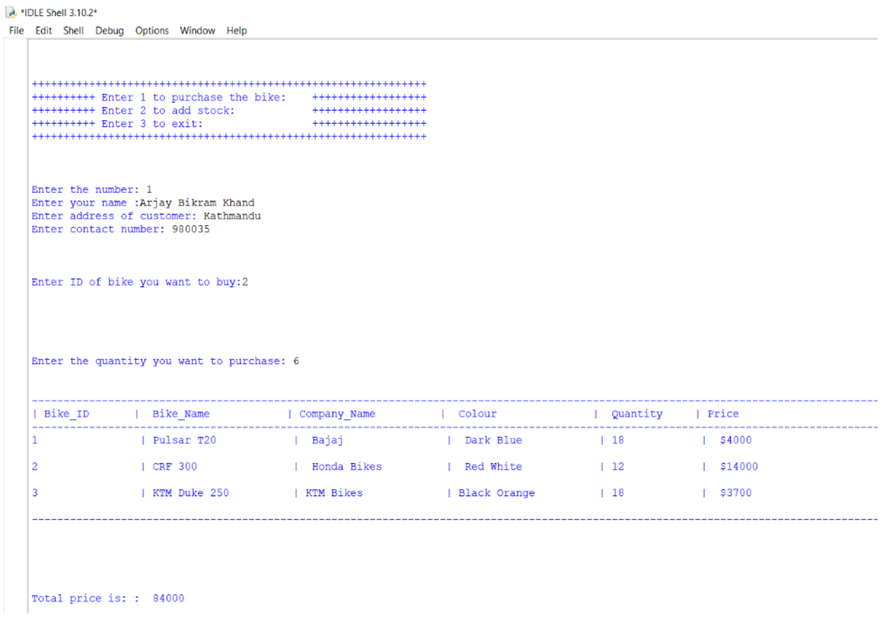
Figure 17: Display of bill of purchase bike in text file

7.5 Test 5

Show the update in stock of bike

Test 5.1	5.1
Action	Deduction of quantity by selling bike
Expected Result	Quantity of bike will be deducted after selling.
Actual Result	The quantity is being deducted.
Test Result	Successful.

Table 6: Test 5.1

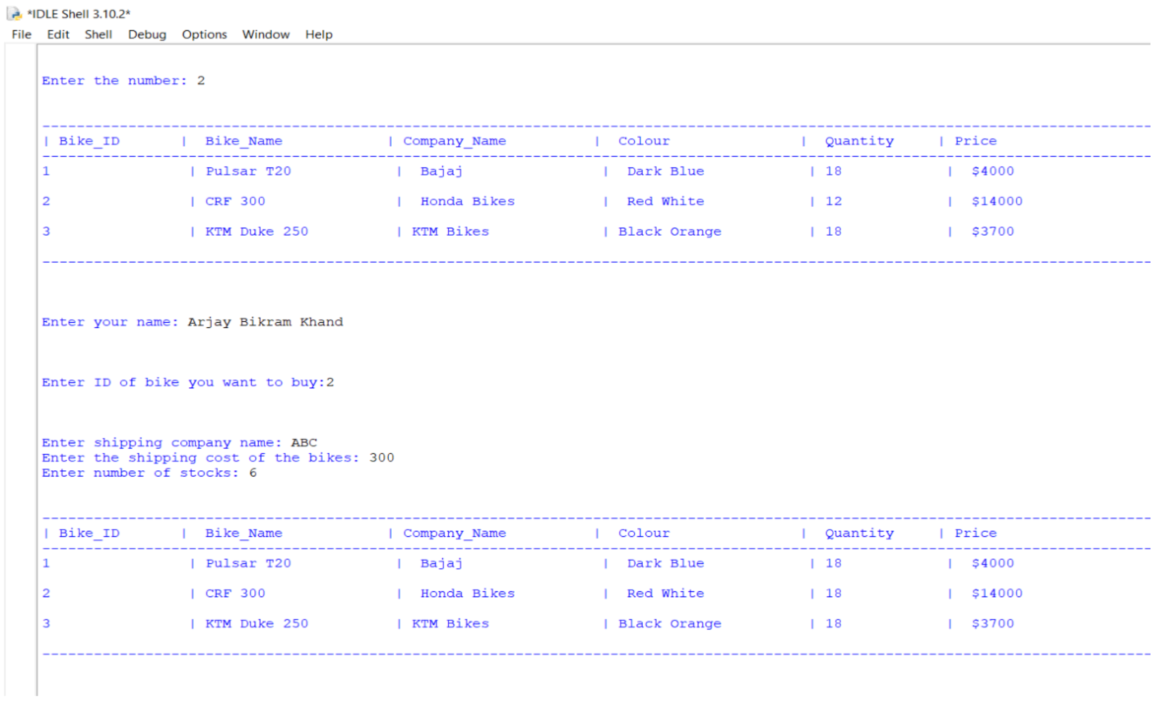


```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++
++++++++++ Enter 2 to add stock:            ++++++++++++++++++
++++++++++ Enter 3 to exit:                 ++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter the number: 1
Enter your name :Arjay Bikram Khand
Enter address of customer: Kathmandu
Enter contact number: 980035


Enter ID of bike you want to buy:2



Enter the quantity you want to purchase: 6

-------------------------------------------------------------------------------------------------------------------------------------
| Bike_ID       |  Bike_Name           | Company_Name          |  Colour               |  Quantity      | Price
-------------------------------------------------------------------------------------------------------------------------------------
1                | Pulsar T20           |  Bajaj                 |  Dark Blue             | 18             |  $4000

2                | CRF 300              |  Honda Bikes           |  Red White             | 12             |  $14000

3                | KTM Duke 250         | KTM Bikes              | Black Orange           | 18             |  $3700

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



Total price is: :  84000
```

Figure 18: Purchase of bike

Test 5.2	5.2
Action	Adding of stocks of bike
Expected Result	Quantity of bike will be increased
Actual Result	The quantity is being increased.
Test Result	Successful.

Table 7: Test 5.2



```
Enter the number: 2

----------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name              | Company_Name           |  Colour                 |  Quantity     | Price
----------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20              |  Bajaj                  |  Dark Blue               | 18            |  $4000

2                 | CRF 300                 |  Honda Bikes            |  Red White               | 12            |  $14000

3                 | KTM Duke 250            | KTM Bikes               | Black Orange             | 18            |  $3700

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


Enter your name: Arjay Bikram Khand


Enter ID of bike you want to buy:2


Enter shipping company name: ABC
Enter the shipping cost of the bikes: 300
Enter number of stocks: 6

----------------------------------------------------------------------------------------------------------------------
| Bike_ID        |  Bike_Name              | Company_Name           |  Colour                 |  Quantity     | Price
----------------------------------------------------------------------------------------------------------------------
1                 | Pulsar T20              |  Bajaj                  |  Dark Blue               | 18            |  $4000

2                 | CRF 300                 |  Honda Bikes            |  Red White               | 18            |  $14000

3                 | KTM Duke 250            | KTM Bikes               | Black Orange             | 18            |  $3700

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

Figure 19: Added stocks of bike

8. Conclusion

I learned about Python programming in this course. This assessment is the initial one. Python seems to have qualities that could make learning a programming language easier. Before choosing, though, it is important to carefully weigh all of the components of the introductory programming module. It's important to look into the effects of integrating Python into the current software. Try Python; it's an excellent programming language. You might want to use it for more than just finding your plane. It knows when to extend the plane and when to use a different approach. According to statistics, you will have content developers. Some developers can't work with planes because it's against the rules of their jobs.

Users who require bike data but lack the time to search can use the Bike Management System, which is one of the best solutions available. Information and amenities for users are available without errors. The preferred bike models of users are immediately available. Visits to the showroom are not required on a regular basis. To obtain the data they require, anyone can use the system. The final goal and objective is to simplify and improve time efficiency for both the user and the business. To calm the buyer down as they buy bikes, as well.

9. References

Anon., 2020. *Techopedia.* [Online]
Available at: www.techopedia.com
[Accessed 13 May 2022].

Anon., 2020. *TechTarget.* [Online]
Available at: www.techtarget.com
[Accessed 13 May 2022].

Anon., 2021. *CFI.* [Online]
Available at: www.corporatefinanceinstitute.com
[Accessed 13 May 2022].

Anon., 2021. *Computer Hope.* [Online]
Available at: www.computerhope.com
[Accessed 12 May 2022].

Anon., 2021. *Tutorials Point.* [Online]
Available at: www.tutorialspoint.com
[Accessed 13 May 2022].

Anon., 2022. *GeeksforGeeks.* [Online]
Available at: www.geeksforgeeks.org
[Accessed 13 May 2022].

Anon., 2022. *Python.* [Online]
Available at: www.python.org
[Accessed 12 May 2022].

Downey, L., 2021. *Investopedia.* [Online]
Available at: www.investopedia.com
[Accessed 12 May 2022].

J.Graph, 2022. *Technology Evaluation Centers.* [Online]
Available at: www3.technologyevaluation.com
[Accessed 12 May 2022].

10. Appendix

```
"""
Arjay Bikram Khand
Bike Management System
Fundamentals of Computing
"""


#importing datetime
import datetime



# the given below function displays the welcome message to the user
def Welcome_to_program():

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
    print("++++++++++++++++++++++++"+" Welcome to the program "+"
++++++++++++++++++++++++++++++++++++++++++++++++++")

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
    print("\n")


# to open text file
def open_text_file():
    file = open("Bikes.txt","r")
    return file


# the given function displays the bikes from the text file to the user
```

```
def display_bike():
   print("\n")

print("----------------------------------------------------------------------------------------------------------
------------------------------------------")
   print("| Bike_ID \t|  Bike_Name \t\t| Company_Name \t\t|  Colour \t\t|  Quantity \t| Price
")

print("----------------------------------------------------------------------------------------------------------
------------------------------------------")
   file = open_text_file()
   id = 1
   for line in file:
      print(id,"\t\t | "+ line.replace(",","\t\t | "))
      id += 1

print("----------------------------------------------------------------------------------------------------------
------------------------------------------")
   print("\n\n")
   file.close()

# the given function adds the bikes from the text file to the 2D list
def bike_to_2D_list():
      read_file = open_text_file()
      My_list = []
      for i in read_file:
         i = i.replace("\n", "")
         My_list.append(i.split(","))
      return(My_list)

# the given below function displays options for the users in the system
```

```
def user_for_operation():

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
")
   print("++++++++++ Enter 1 to purchase the bike:    ++++++++++++++++++")
   print("++++++++++ Enter 2 to add stock:            ++++++++++++++++++")
   print("++++++++++ Enter 3 to exit:                 ++++++++++++++++++")

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
")
   print("\n\n")


# the given below function checks if the input is 1,2 or 3 or anyother numbers
def user_input():
    try:
      loop = True
      while loop == True:
       user_for_operation()
       user_input = int(input("Enter the number: "))
       DT = datetime.datetime.now()
       T = DT.strftime("%H:%M:%S")
       D = DT.strftime("%D/%M/%Y")
       S = 0

       if user_input == 1:
         Name = input("Enter your name :")
         Address = input("Enter address of customer: ")
         Contact = input("Enter contact number: ")
         the_bike_id = validating_bike_id()
         print("\n")
```

```
            the_q = quantity_validation(the_bike_id)
            Sell(the_bike_id, the_q)
            print("\n")
            total_price = the_price(the_bike_id, the_q)
            print(total_price)
            amount = int(input("Enter total amount paid by customer:"))
            Bill(the_bike_id, T, D, Name, str(amount), Address, Contact, str(the_q),
Customer_Detail)
        elif user_input == 2:
            display_bike()
            Name1 = input("Enter your name: ")
            the_bike_id = validating_bike_id()
            Company_Details = Company_Detail()
            Stock = int(input("Enter number of stocks: "))
            Stock_checking(Stock)
            Adding_stock(the_bike_id, Stock)
            print("\n")
            cost = int(input("Enter importing cost: "))
            SellBill(the_bike_id, T, D, Name1, str(cost), str(Stock),Company_Details)
            bike_to_stock()
        elif user_input == 3:
            exit_system()
            loop = False
        else:
            invalid_user_input()
    except ValueError:
        print("Please give us proper informations.")


# the given below function validate the bike ID
```

```
def validating_bike_id():
    loop = True
    while loop:
        try:
            print("\n\n")
            valid_id = int(input("Enter ID of bike you want to buy:"))
            print("\n\n")
            while valid_id <= 0 or valid_id > len(bike_to_2D_list()):
                print("\n\n")
                print("++++++++++++++++++++++++++++++++++++++++++++++++++")
                print("Please enter  valid Bike ID !")
                print("++++++++++++++++++++++++++++++++++++++++++++++++++")
                print("\n\n")
                display_bike()
                print("\n\n")
                print("++++++++++++++++++++++++++++++++++++++++++++++++++")
                Valid_id = int(input("Enter Id of the bike you want to buy: "))
                print("++++++++++++++++++++++++++++++++++++++++++++++++++")
                print("\n\n")
            return valid_id
        except ValueError:
            print("\n\n")
            print("please give integer value only! ")
            print("\n\n")

# the given below function is called when user press 1
def bike_to_stock():
    print("\n\n")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("The Bike has been added to the stock")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")

# the given below function is called when user press 2
def purchasing_bike():
    print("\n\n")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("Thank you for purchasing the bike")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")

# the given below function is called when the user gives invalid input
def invalid_user_input():
    print("\n\n")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("Invalid input!!!")
    print("Please provide value as 1,2 or 3.")
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")

# the given below function is called when user press 3
def exit_system():
    print("\n\n")
    print("+++++++++++++++++++++++++++++++++++++++++++++")
    print("Thank you for using our system")
    print("+++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")


# details of the customers
def Customer_Detail():
    Name = input("Enter your name : ")
    Address = input("Enter your address : ")
    Contact = input("Enter your contact number : ")
    return Name,Address,Contact


# the given below function is called for validation of the quantity
def quantity_validation(the_bike_id):
    loop = True
    while loop:
        try:
            bike_list = bike_to_2D_list()
            user_quantity = int(input("Enter the quantity you want to purchase: "))
            while user_quantity <= 0 or user_quantity > int(bike_list[the_bike_id - 1][3]):
                print("\n")
                print("++++++++++++++++++++++++++++++++++++++++++++++++")
                print("Please provide a  valid Qunatity ID !!!")
                print("++++++++++++++++++++++++++++++++++++++++++++++++")
                print("\n")
                display_bike()
                user_quantity = int(input("Enter the quantity you want to purchase: "))
                print("\n")
            return user_quantity
        except ValueError:
            print("\n" "Only supports integer values!  " "\n")
```

```
## the given below function is called when quantity of bike is updated
def update_quantity(bike_list):
    file = open("Bikes.txt","w")
    for i in bike_list:
        file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) + "\n")
    file.close()
    display_bike()

# the given below function is called for finalization of sell of bikes
def Sell(the_bike_id,the_q):
    bike_list = bike_to_2D_list()
    bike_list[the_bike_id -1][3] = int(bike_list[the_bike_id - 1][3]) - the_q
    update_quantity(bike_list)

# the given below function is called again to update the stock of bike
def Update_stock(bike_list):
        file = open("Bikes.txt", "w")
        for i in bike_list:
            file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) + 
"\n\n")
            file.close()
            display_bike()

# the given below function is called to check the stock of bike
def Stock_checking(Stock):
    while Stock <= 0:
        print("Enter valid quantity of stocks.")
        Stock = int(input("Enter number of stock to add"))
        display_bike()
    return Stock


# the given below function is called to update stocks of bikes
def Update_stock1(bike_list1):
        file = open("Bikes.txt", "w")
        for i in bike_list1:
            file.write(str(i[0]) + "," + str(i[1]) + "," + str(i[2]) + "," + str(i[3]) + "," + str(i[4]) +
"\n\n")
        file.close()
        display_bike()


# the given below function is called to price of bikes
def the_price(the_bike_id,the_q):
    bike_list = bike_to_2D_list()
    total_price = int(bike_list[the_bike_id - 1][4].replace("$","")) * int(the_q)
    print("Total price is: : ", total_price)
    print("\n")
    return total_price


# the given below function is called to add the stocks of bike
def Adding_stock(the_bike_id1,the_q1):
    bike_list = bike_to_2D_list()
    bike_list[the_bike_id1 -1][3] = int(bike_list[the_bike_id1 - 1][3]) + the_q1
    update_quantity(bike_list)


# the given below function is called for bill of the bikes purchased
def Bill(the_bike_id, T, D, Name, amount, Address, Contact, the_q, Customer_Details):
    bike_list = bike_to_2D_list()
    the_price = int(bike_list[the_bike_id-1][4].replace("$", ""))*the_q

    with open("Purchase- " + Name + ".txt", "w+") as f:
        f.write("+++++++++++++++++++++++++++++++++++++++++++++++   \n\n")
        f.write("      Bike Management System              \n\n")
        f.write(" Name of Customer: " + Name + "         \n\n")
        f.write(" Address of Customer: " + Address + "      \n\n")
        f.write(" Contact number of Customer: " + Contact + " \n\n")
        f.write("Time of bike purchased is: " + T + "\n\n")
        f.write("Date of bike purchased is: " + D + "\n\n")
        f.write("Purchased Bike is: " + bike_list[the_bike_id - 1][0] + "\n\n")
        f.write("Purchased Bike Company is: " + bike_list[the_bike_id - 1][1] + "\n\n")
        f.write("Color of bike is: " + bike_list[the_bike_id - 1][2] + the_q + "\n\n")
        f.write("Quantity of bike is: " + the_q + "\n\n")
        f.write("Amount: " + amount +" \n\n")
        f.write("Total price is: " + the_price + "\n\n")

        #display of bill
        display_file = open("Purchase- " + Name + ".txt", "r")
        for output in display_file:
            print(output)
        purchase_more_bikes(the_bike_id, T, D, Name, amount, Address, Contact, the_q)
        user_for_operation()


def Bills(the_bike_id, T, D, Name, amount, Address, Contact, the_q, Customer_Details):
    bike_list = bike_to_2D_list()
    the_price1 = int(bike_list[the_bike_id-1][4].replace("$", ""))*the_q
    with open("Purchase- " + Name + Contact+ Address + ".txt", "w+") as f:
        f.write("++++++++++++++++++++++++++++++++++++++++++++++++++    \n\n")
        f.write("      Bike Management System               \n\n")
        f.write(" Name of Customer: " + Name + "         \n\n")
        f.write(" Address of Customer: " + Address + "       \n\n")
        f.write(" Contact number of Customer: " + Contact + " \n\n")
        f.write("Time of bike purchased is: " + T + "\n\n")
        f.write("Date of bike purchased is: " + D + "\n\n")
        f.write("Purchased Bike is: " + bike_list[the_bike_id -1][0] + "\n\n")
        f.write("Purchased Bike Company is: " +
                bike_list[the_bike_id -1][1] + "\n\n")
        f.write("Color of bike is: " +
                bike_list[the_bike_id -1][2] + the_q + "\n\n")
        f.write("Quantity of bike is: " + the_q + "\n\n")
        f.write("Total price is: " + the_price1 + "\n\n")

        #display of bill
        display_file = open("Purchase- " + Name + ".txt", "a")
        for output in display_file:
            print(output)
        purchase_more_bikes(the_bike_id, T, D, Name, amount, Address, Contact, the_q)
        user_for_operation()

# the given below function is called to purchase more bikes
def purchase_more_bikes(the_bike_id, T, D, Name, amount, Address, Contact, the_q):
    loop = True
    while loop:
        try:
            print("Type Yes to Continue or No to Cancel.")
            user_quantity = input("Do you want to purchase another bike?: .")
            print("\n\n")
            if user_quantity.upper() == "Yes":
                the_bike_id = validating_bike_id()
                display_bike()
                the_q = quantity_validation(the_bike_id)
                sell(the_bike_id, the_q)
                the_price1 = total_price(the_bike_id, the_q)
                Bill(the_bike_id, T, D, Name, amount, Address, Contact)
                print("Total price of the bike is: ", the_price1)
                print("\n\n")
            else:
                user_quantity.upper() == "No"
                exit_system()
                loop = False

        except ValueError:
                print("\n\n", " Supports Yes/No Only! " "\n\n")

# teh given below function is called to validate the bike id for other list
def validating_bike_id1():
    loop = True
    while loop:
        try:
            print("\n")
            valid_id = int(input("Enter  id of bike you want to purchase: "))
            print("\n")
            while valid_id <= 0 or valid_id > len(bike_to_2D_list()):
                print("  Please give a valid Bike_ID     ")
                print("\n")
                display_bike()
                print("\n")
                valid_id = int(input("Enter id of bike you want to purchase: "))
                print("\n")
            return valid_id
        except ValueError:
            print("  Only supports integer values!  ")

# details of the company
def Company_Detail():
    Shipping_Company_name = input("Enter shipping company name: ")
    Shipping_Cost = input("Enter the shipping cost of the bikes: ")
    return Shipping_Company_name,Shipping_Cost

# to show the given validate for the bike
def quantity_validation1(the_bike_id1):
    loop = True
    while loop:
        try:
            bike_list = bike_to_2D_list()
            user_quantity = int(input("Enter quantity you want to add: "))
            while user_quantity <= 0:
                print("\n")
                print("  Please give us valid quantity. ")
            print("\n")
            display_bikes()
            user_quantity = int(input("Enter quantity you want to add: "))
            print("\n")
         return user_quantity
      except ValueError:
         print("  Only supports integer values! ")



# to show the price of the bike of other
def the_price1(the_bike_id1,the_q1):
   bike_list = bike_to_2D_list()
   total_price = int(bike_list[the_bike_id1 - 1][4].replace("$","")) * int(the_q1)
   return total_price

#the given below function is called to sell the bike with bill
def SellBill(the_bike_id, T, D, Name1, cost, Stock, Company_Detail):
   bike_list1 = bike_to_2D_list()
   Year = str(datetime.datetime.now().year)
   Month = str(datetime.datetime.now().month)
   Day = str(datetime.datetime.now().day)
   Second = str(datetime.datetime.now().second)

   with open("AddedStocks-" + Name1 + Year+""+Month+""+Day+""+Second+".txt",
"w+") as f:
      f.write("+++++++++++++++++++++++++++++++++++++++++++++++   \n\n")
      f.write("     Bike Management System              \n\n")
      f.write("   Name of Shipping Company: "+Name1+"   \n\n")
      f.write(" Shipping Cost is: "+cost+"              \n\n")
```

```
        f.write(" Added bike stock is: "+T+"              \n\n")
        f.write(" Time of Stocks added is: "+T+"          \n\n")
        f.write(" Date of stocks added is: "+D+"          \n\n")
        f.write("Added stocks : " + str(Stock))
        f.write(" Stocked Bike Name is: " + bike_list1[the_bike_id - 1][0] + "\n\n")
        f.write(" Stocked Bike Company is: " + bike_list1[the_bike_id - 1][1] + "\n\n")
        f.write(" Stocked Bike Color is: " + bike_list1[the_bike_id - 1][2] + "\n\n")
        f.write(" Stocked Bike Price is: " + bike_list1[the_bike_id - 1][4] + "\n\n")
# the given below function is called when user press 3
def exit_system():
    print("\n\n")
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("Thank you for using our system")
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")

# the given below function is called when the user gives invalid input
def invalid_Input():
    print("\n\n")
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("Invalid input!!")
    print(" Please give valid number! ")
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("\n\n")

# calling function to execute the function
Welcome_to_program()
display_bike()
```

```
user_input()
```