SHIV NADAR SCHOOL GURUGRAM

COMPUTER SCIENCE PROJECT

GROCERY MANAGEMENT SYSTEM

SHIV NADAR SCHOOL
Education for Life

Submitted by:
Kushi Arun Kumar, Kasmya Bhatia

COMPUTER SCIENCE INVESTIGATORY PROJECT

GROCERY MANAGEMENT SYSTEM

SUBMITTED BY:

Kushi Arun Kumar, Kasmya Bhatia

Under the guidance of

Mr. Himanshu Arora

(PGT CS)

INDEX:

CERTIFICATE

This is to certify that Kasmya Bhatia and Kushi Arun Kumar of Class XII have prepared the report on a Project entitled "GROCERY MANAGEMENT SYSTEM". The report is the result of their efforts and endeavors. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII as prescribed by CBSE for AISSCE 2022-2023 in Shiv Nadar School, Pahari Road, Sector 26, DLF City, Phase -1, Block -E, Gurugram, Haryana 122011.

Roll Number: **Date:**

______________ ______________

Signature of **Signature of**

Internal Examiner: **External Examiner:**

__________________ __________________

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Mr. Himanshu Arora, PGT(CS) as well as our Principal Ms. Monica Sagar who gave me the golden opportunity to do this wonderful project on the topic "Grocery Management System", which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends/classmates for their help and wishes for the successful completion of this project within the limited period.

Kushi Arun Kumar,
Kasmya Bhatia

DECLARATION

I hereby declare that the project work entitled “GROCERY MANAGEMENT SYSTEM” submitted to Shiv Nadar School, Gurugram for the subject Computer Science under the guidance of Mr. Himanshu Arora, PGT(CS) is a record of original work done by me.

I further declare that this project or any part of it has not been submitted elsewhere for any other class.

Class:

Place:

Date:

INTRODUCTION TO PYTHON

Python is an interpreted, high level, general purpose programming language which was created by Guido van Rossum and first released in 1991. Python has a design philosophy that emphasizes code readability. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features Dynamic Typing system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural and has a large and comprehensive standard library and has various other libraries that can be downloaded externally.

Python interpreters are available for various operating systems. CPython, the reference implementation of Python, is an open source software and has a community based development model, as do nearly all of Python’s other implementations. Python and CPython are managed by the non-profit Python Software Foundation. One can also use other implementations like Spyder IDE or Python IDLE.

SQL CONNECTIVITY:

SQL stands for Structured Query Language. SQL databases store collections of information that is organised in the form of tables and can be easily accessed, managed and updated. MySQL is a free-to-use, open-source database that facilitates efficient management of databases by connecting them to the software of python.

Connectivity of SQL to python can be of a high advantage to us due to the following factors. Storing data using SQL can reduce chances of data redundancy and inconsistency, sharing of data can also be easily facilitated while integrity of data being maintained.

HARDWARE AND SOFTWARE USED:

HARDWARE REQUIREMENTS:

Processer	AMD Ryzen 4000 SERIES
RAM	4GB
Hard Disk Drive	500GB
Monitor	18.5 LED monitor
Mouse	Logitech
Keyboard	104 Keys Keyboard
Port	USB Port

SOFTWARE REQUIREMENTS:

Operating System	Windows 10
Front End Tool	Python
Report tool	Data Report/Microsoft Word

INTRODUCTION TO PROJECT:

The purpose of the Grocery Shop Management System is to automate the existing manual system by using computerized software and equipment. The motive is to fulfill requirements of a grocery store by enabling storage of valuable data for a longer period of time and making it easily manageable, accessible and . Grocery Shop Management System can also lead to an error free, secure, reliable and fast way of management. As it can assist the user to concentrate on other activities rather than concentrating on record keeping. Thus it can help organizations in better utilization of resources such as time.

MODULES USED:

1. Product module:
   All the details regarding the products that are being sold in the store will be stored on a database in an organized manner to make this kind of information accessible and comprehensible. These details may include their price, manufacturer, product ID, category, quantity and so on. Every time a sale is made, the system registers this information and stores it which will help the system track the product quantity (sold and left), the customer details, the profit made and the loss experienced from this sale, and many more.

2. Stock management module:
   The system is equipped to manage the stock and the inventory of the store, details regarding the products and their quantity (as mentioned above) at a given time will be stored on a database and when a

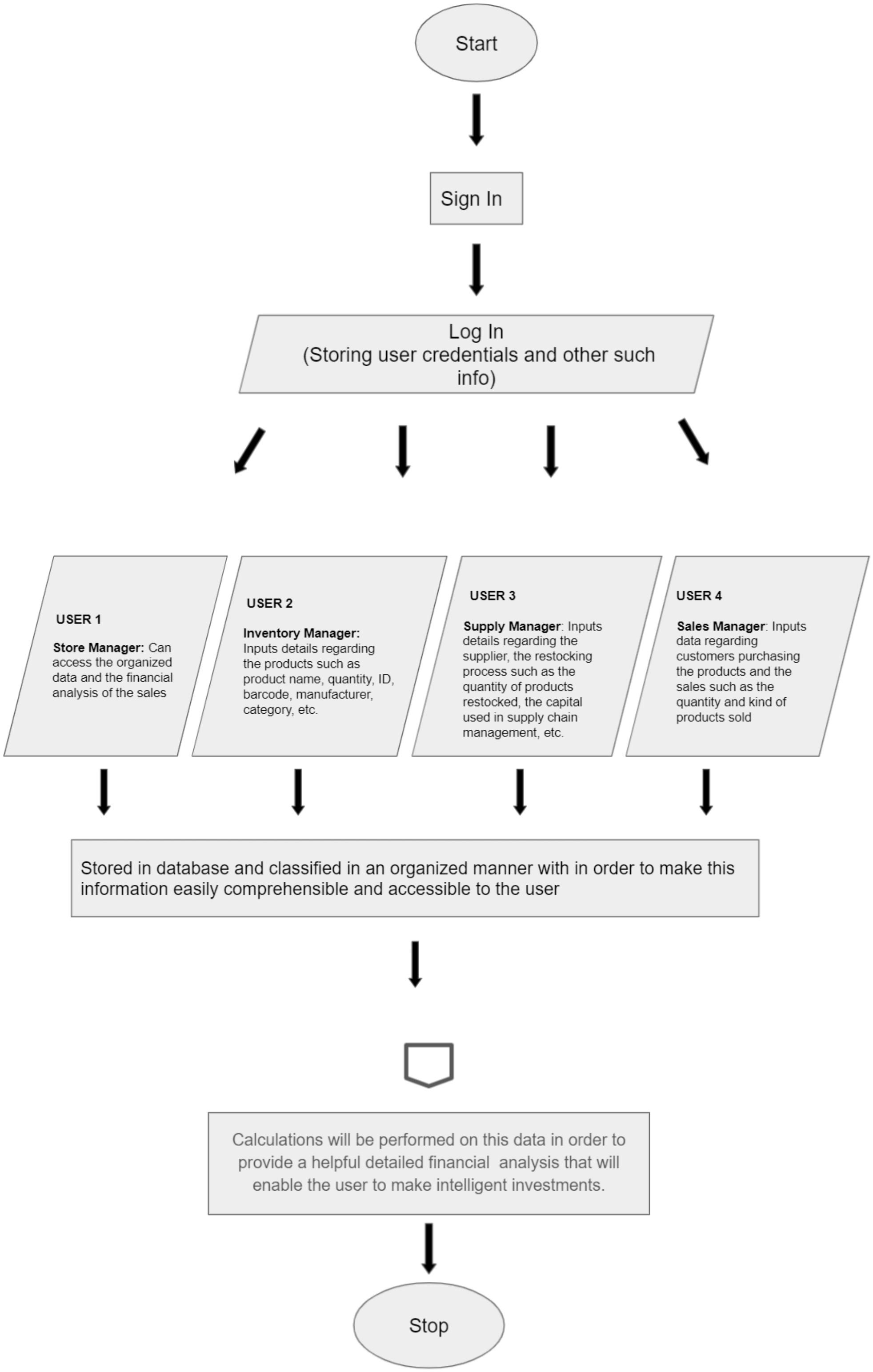
product quantity reaches a certain limit, it will notify the user to restock and order it again.

3. Financial management module:
   The system will also provide a detailed analysis of the profit, loss and total revenue of the store on a monthly basis. Along with that, it will provide a helpful evaluation on the customer demand based on information of the categories and the quantities of the particular products sold in a given period of time which will allow the user to invest smartly, and will allow them to maximize their profits by targeting these customer expectations and desires.

4. Customer and Supplier module:
   The system will store necessary information and details regarding the customers who are buying from the store and the suppliers who provide the business with the products. This will ease communication and make the process painless and more efficient. Moreover, it will enable the user to target customers through email/sms advertising and marketing and will ultimately help increase sales.

5. Login and system users module:
   This module will handle information regarding the login details and the different users of the system. A portal by which the user, and only the registered user, can login where and when required.

FLOWCHART:


Start
Sign In
Log In
(Storing user credentials and other such info)
USER 1
Store Manager: Can access the organized data and the financial analysis of the sales
USER 2
Inventory Manager: Inputs details regarding the products such as product name, quantity, ID, barcode, manufacturer, category, etc.
USER 3
Supply Manager: Inputs details regarding the supplier, the restocking process such as the quantity of products restocked, the capital used in supply chain management, etc.
USER 4
Sales Manager: Inputs data regarding customers purchasing the products and the sales such as the quantity and kind of products sold
Stored in database and classified in an organized manner with in order to make this information easily comprehensible and accessible to the user
Calculations will be performed on this data in order to provide a helpful detailed financial analysis that will enable the user to make intelligent investments.
Stop

Code

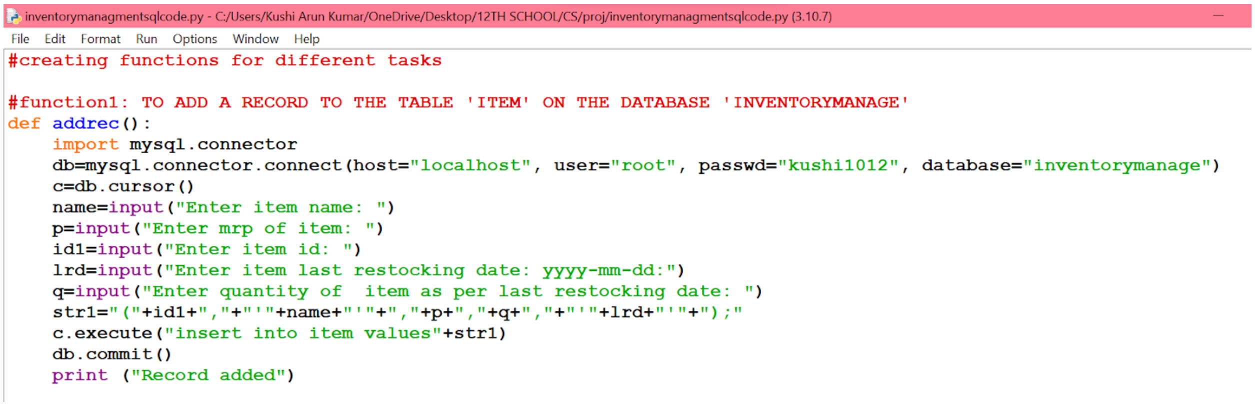
#creating functions for different tasks

- **FUNCTION1: TO ADD A RECORD TO THE TABLE 'ITEM' ON THE DATABASE 'INVENTORYMANAGE'**

In the following code, Connection between the MySQL database and Python is made using the connect() function from the mysql.connector module. Information such as database details like HostName, username, and the password is passed. The connection object is returned. Input from the user is taken in order to store in the form of records.

Code:

```
def addrec():
   import mysql.connector
   db=mysql.connector.connect(host="localhost", user="root",
passwd="kushi1012", database="inventorymanage")
   c=db.cursor()
   name=input("Enter item name: ")
   p=input("Enter mrp of item: ")
   id1=input("Enter item id: ")
   lrd=input("Enter item last restocking date: yyyy-mm-dd:")
   q=input("Enter quantity of  item as per last restocking date: ")
   str1="("+id1+","+"'"+name+"'"+","+p+","+q+","+"'"+lrd+"'"+");"
   c.execute("insert into item values"+str1)
   db.commit()
   print ("Record added")
```

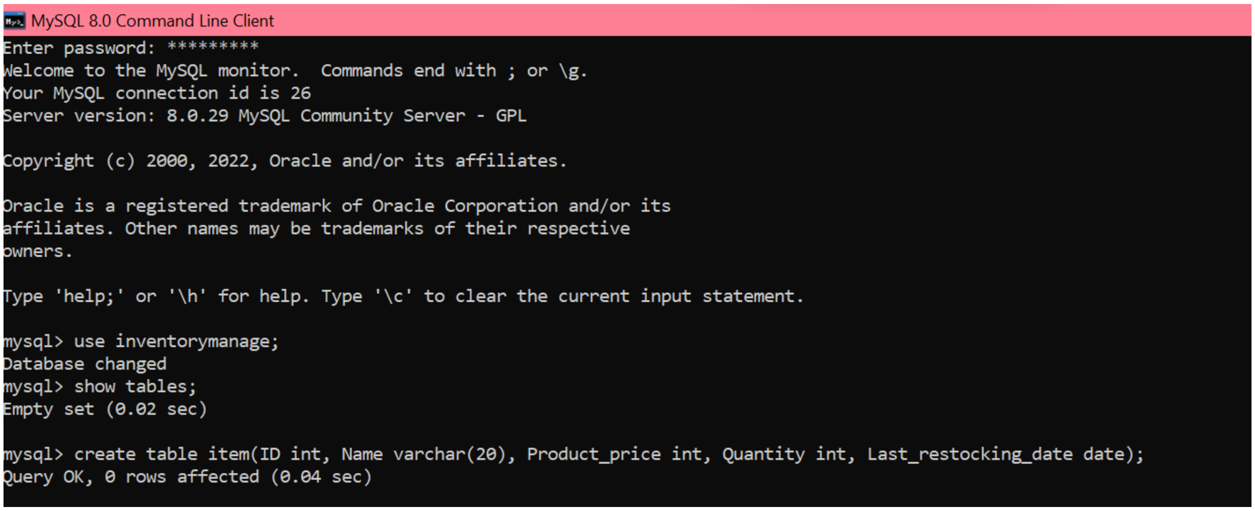


inventorymanagmentsqlcode.py - C:/Users/Kushi Arun Kumar/OneDrive/Desktop/12TH SCHOOL/CS/proj/inventorymanagmentsqlcode.py (3.10.7)
File Edit Format Run Options Window Help

```python
#creating functions for different tasks

#function1: TO ADD A RECORD TO THE TABLE 'ITEM' ON THE DATABASE 'INVENTORYMANAGE'
def addrec():
    import mysql.connector
    db=mysql.connector.connect(host="localhost", user="root", passwd="kushi1012", database="inventorymanage")
    c=db.cursor()
    name=input("Enter item name: ")
    p=input("Enter mrp of item: ")
    id1=input("Enter item id: ")
    lrd=input("Enter item last restocking date: yyyy-mm-dd:")
    q=input("Enter quantity of  item as per last restocking date: ")
    str1="("+id1+","+"'"+name+"'"+","+p+","+q+","+"'"+lrd+"'"+");"
    c.execute("insert into item values"+str1)
    db.commit()
    print ("Record added")
```

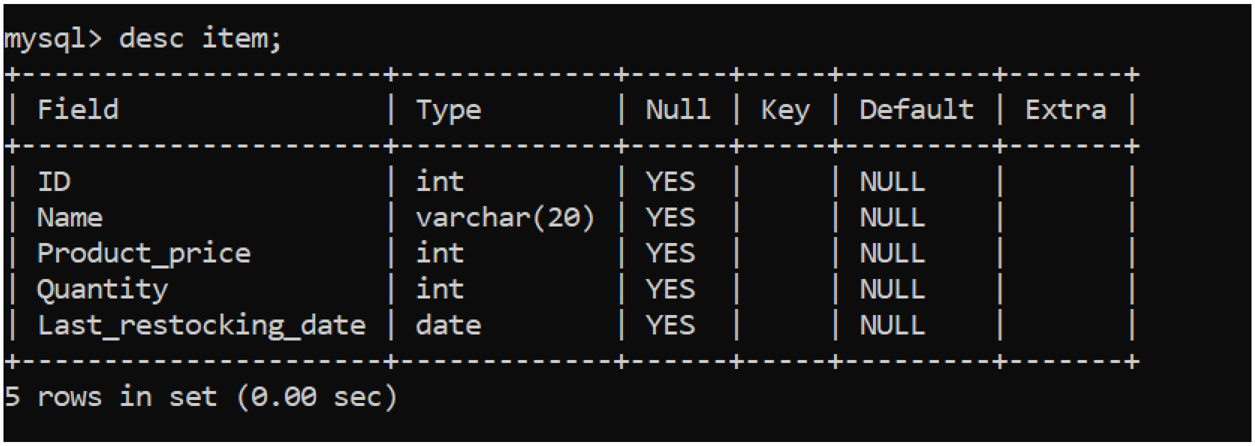
- ## CREATING TABLE USING 'INVENTORYMANAGE' DATABASE

The table named item is created in order to store the records.



```
Enter password: *********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.29 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use inventorymanage;
Database changed
mysql> show tables;
Empty set (0.02 sec)

mysql> create table item(ID int, Name varchar(20), Product_price int, Quantity int, Last_restocking_date date);
Query OK, 0 rows affected (0.04 sec)
```

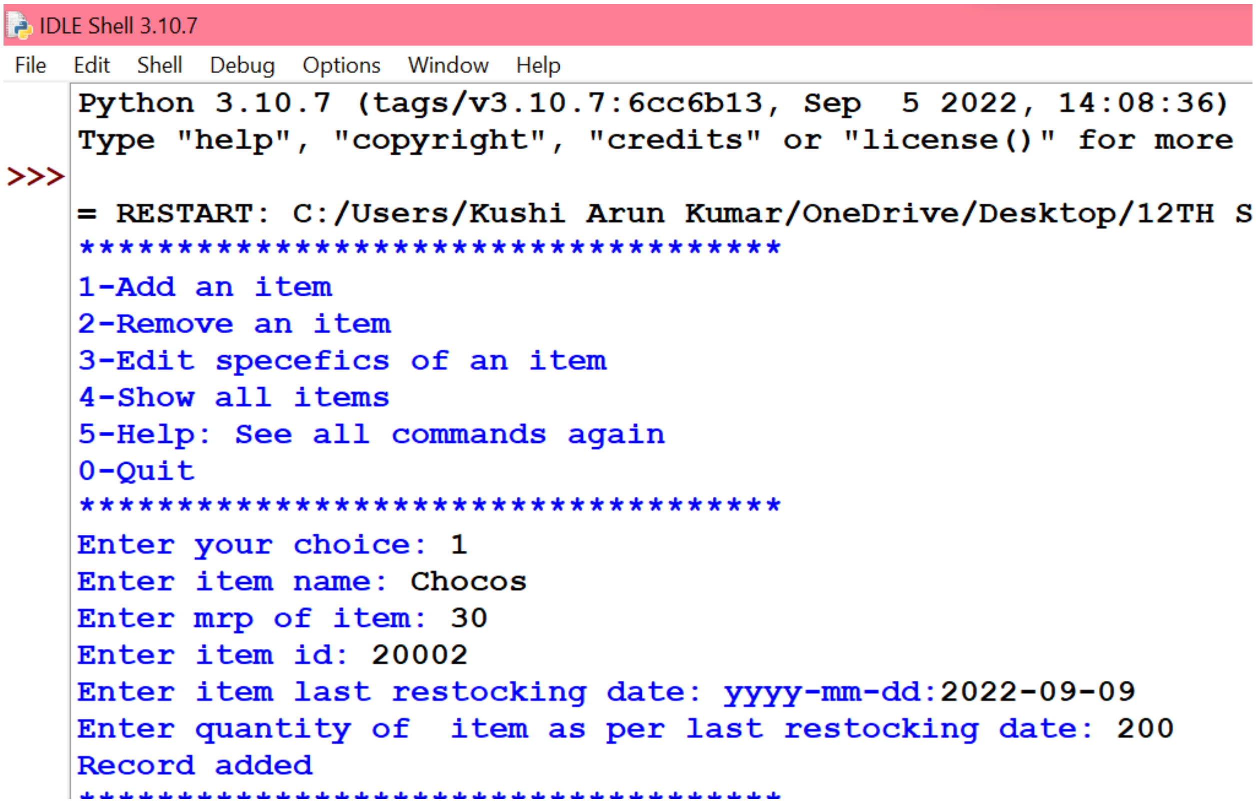
Here the schema of the table item is displayed using desc query.



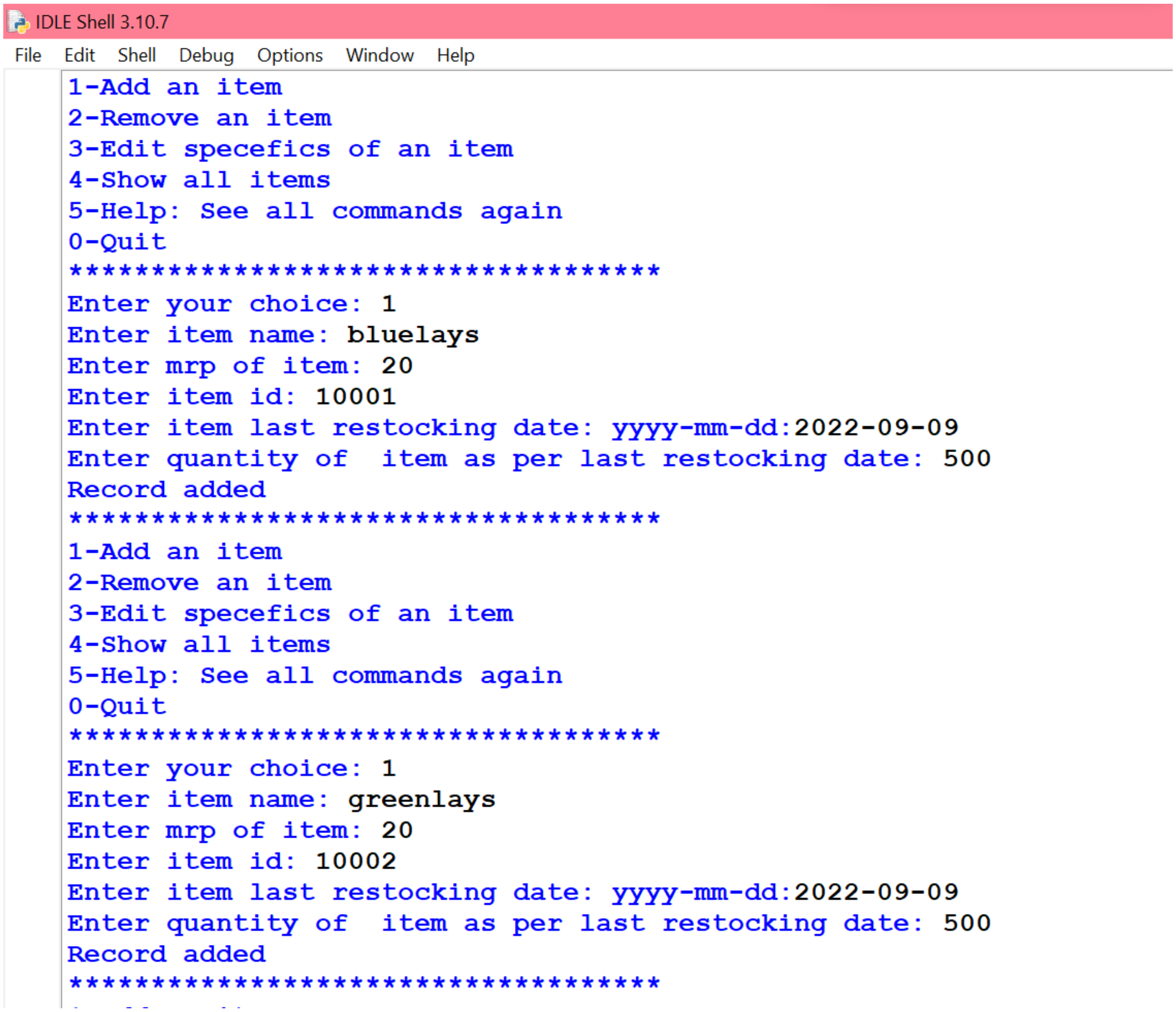
```
mysql> desc item;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| ID                  | int         | YES  |     | NULL    |       |
| Name                | varchar(20) | YES  |     | NULL    |       |
| Product_price       | int         | YES  |     | NULL    |       |
| Quantity            | int         | YES  |     | NULL    |       |
| Last_restocking_date| date        | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Output & Implementation:

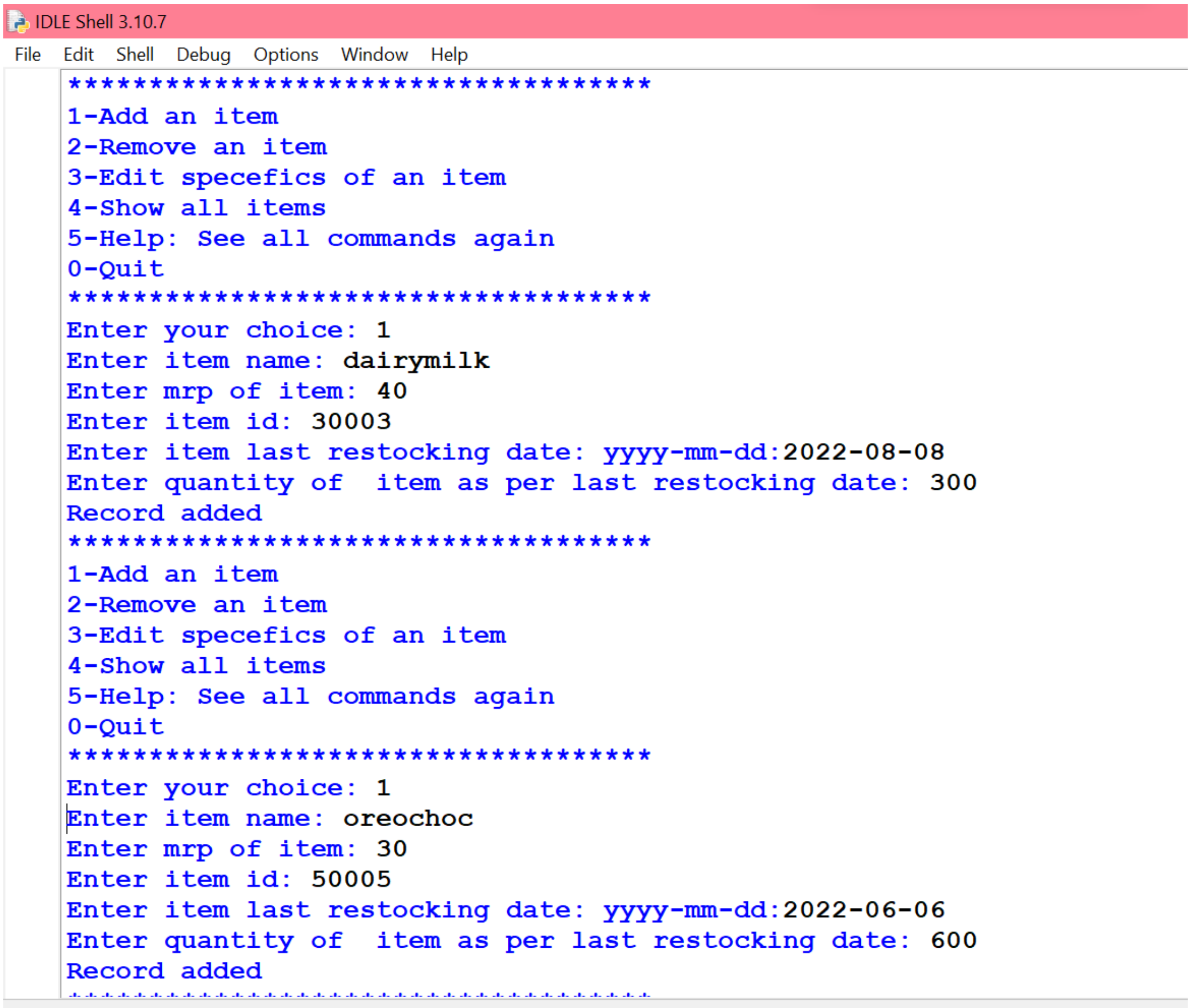
The following input data is added in a record as a result of running the above written code.



```
    Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36)
    Type "help", "copyright", "credits" or "license()" for more
>>> 
    = RESTART: C:/Users/Kushi Arun Kumar/OneDrive/Desktop/12TH S
    ************************************
    1-Add an item
    2-Remove an item
    3-Edit specefics of an item
    4-Show all items
    5-Help: See all commands again
    0-Quit
    ************************************
    Enter your choice: 1
    Enter item name: Chocos
    Enter mrp of item: 30
    Enter item id: 20002
    Enter item last restocking date: yyyy-mm-dd:2022-09-09
    Enter quantity of  item as per last restocking date: 200
    Record added
    ************************************
```

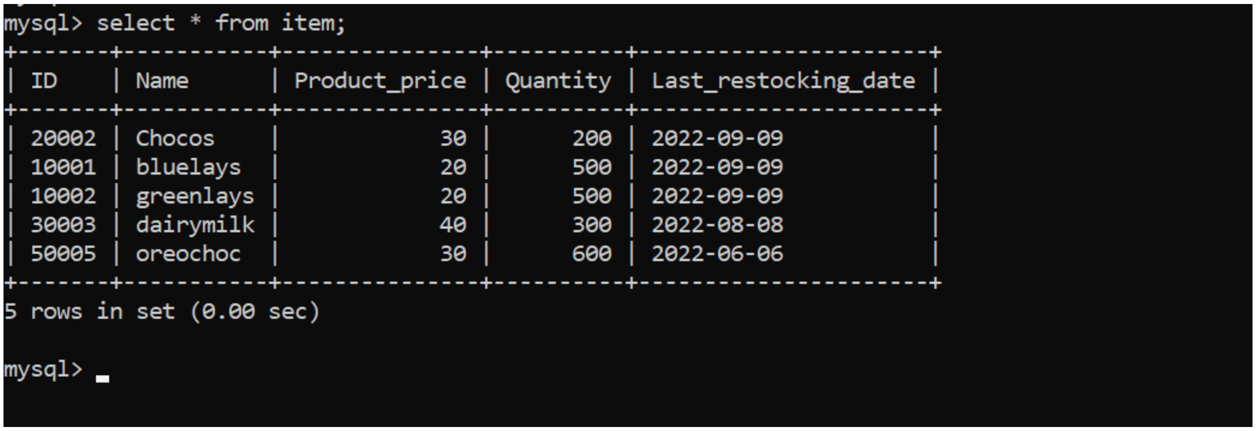


```
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 1
Enter item name: bluelays
Enter mrp of item: 20
Enter item id: 10001
Enter item last restocking date: yyyy-mm-dd:2022-09-09
Enter quantity of  item as per last restocking date: 500
Record added
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 1
Enter item name: greenlays
Enter mrp of item: 20
Enter item id: 10002
Enter item last restocking date: yyyy-mm-dd:2022-09-09
Enter quantity of  item as per last restocking date: 500
Record added
************************************
```



```
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 1
Enter item name: dairymilk
Enter mrp of item: 40
Enter item id: 30003
Enter item last restocking date: yyyy-mm-dd:2022-08-08
Enter quantity of  item as per last restocking date: 300
Record added
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 1
Enter item name: oreochoc
Enter mrp of item: 30
Enter item id: 50005
Enter item last restocking date: yyyy-mm-dd:2022-06-06
Enter quantity of  item as per last restocking date: 600
Record added
************************************
```

By using the select query, the stored information can be viewed in the table item.

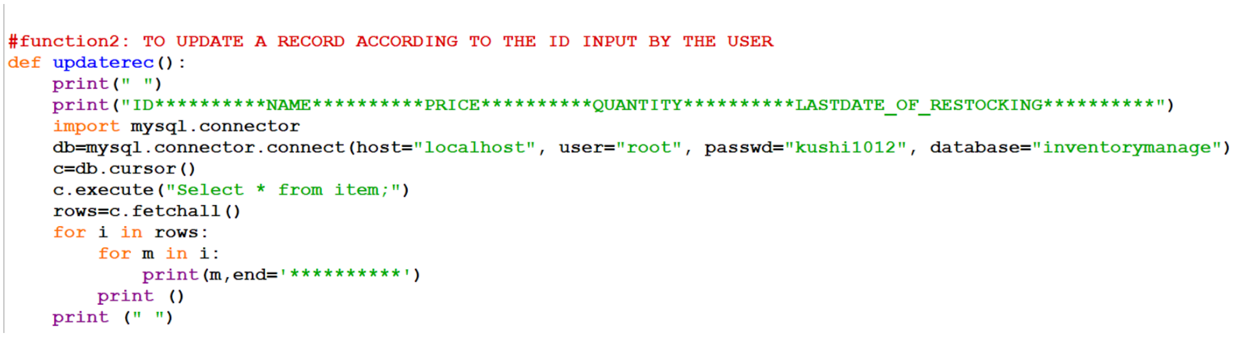


```
mysql> select * from item;
+-------+-----------+---------------+----------+---------------------+
| ID    | Name      | Product_price | Quantity | Last_restocking_date |
+-------+-----------+---------------+----------+---------------------+
| 20002 | Chocos    |            30 |      200 | 2022-09-09          |
| 10001 | bluelays  |            20 |      500 | 2022-09-09          |
| 10002 | greenlays |            20 |      500 | 2022-09-09          |
| 30003 | dairymilk |            40 |      300 | 2022-08-08          |
| 50005 | oreochoc  |            30 |      600 | 2022-06-06          |
+-------+-----------+---------------+----------+---------------------+
5 rows in set (0.00 sec)

mysql>
```

- **FUNCTION2: TO UPDATE A RECORD ACCORDING TO THE ID INPUT BY THE USER**

Code:

```
def updaterec():
   print(" ")

print("ID**********NAME**********PRICE**********QUANTITY*******
***LASTDATE_OF_RESTOCKING**********")
   import mysql.connector
   db=mysql.connector.connect(host="localhost", user="root",
passwd="kushi1012", database="inventorymanage")
   c=db.cursor()
   c.execute("Select * from item;")
   rows=c.fetchall()
   for i in rows:
      for m in i:
         print(m,end='**********')
      print ()
   print (" ")
```



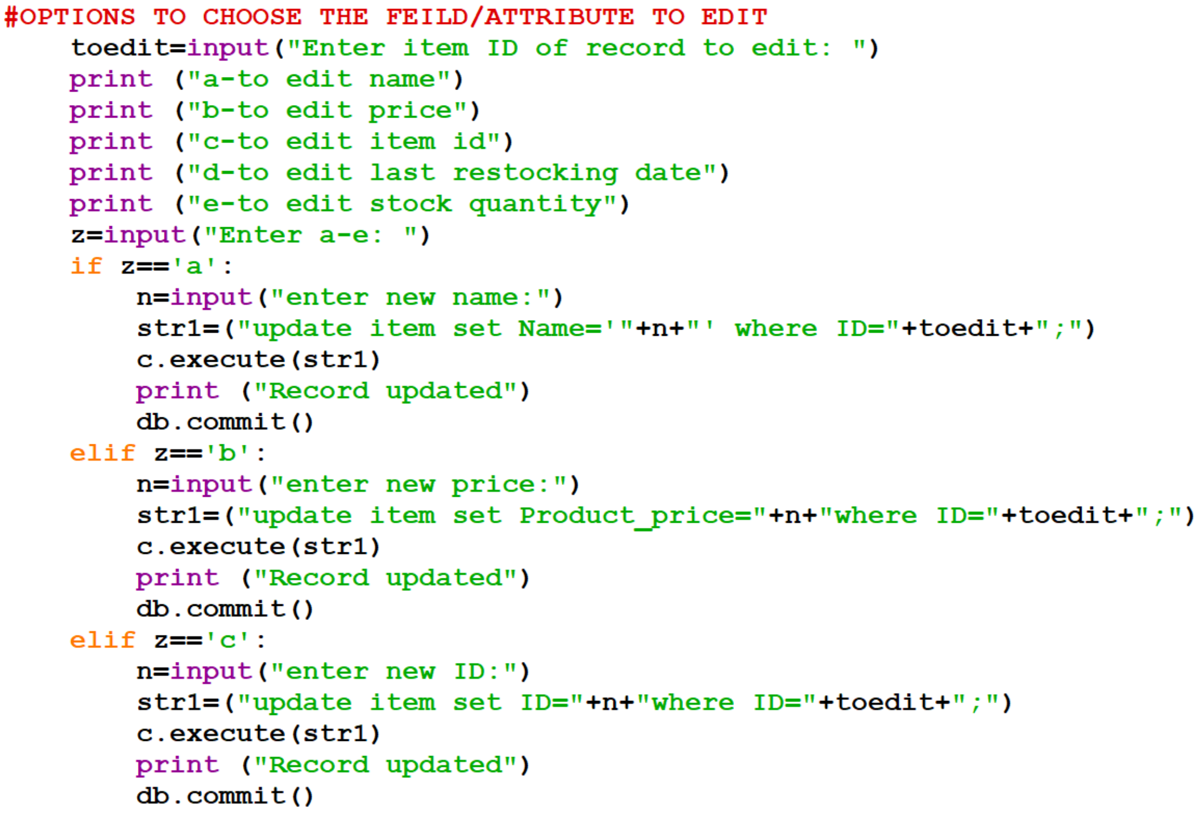
```
#function2: TO UPDATE A RECORD ACCORDING TO THE ID INPUT BY THE USER
def updaterec():
    print(" ")
    print("ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**********")
    import mysql.connector
    db=mysql.connector.connect(host="localhost", user="root", passwd="kushi1012", database="inventorymanage")
    c=db.cursor()
    c.execute("Select * from item;")
    rows=c.fetchall()
    for i in rows:
        for m in i:
            print(m,end='**********')
        print ()
    print (" ")
```

- **OPTIONS TO CHOOSE THE FIELD/ATTRIBUTE TO EDIT**

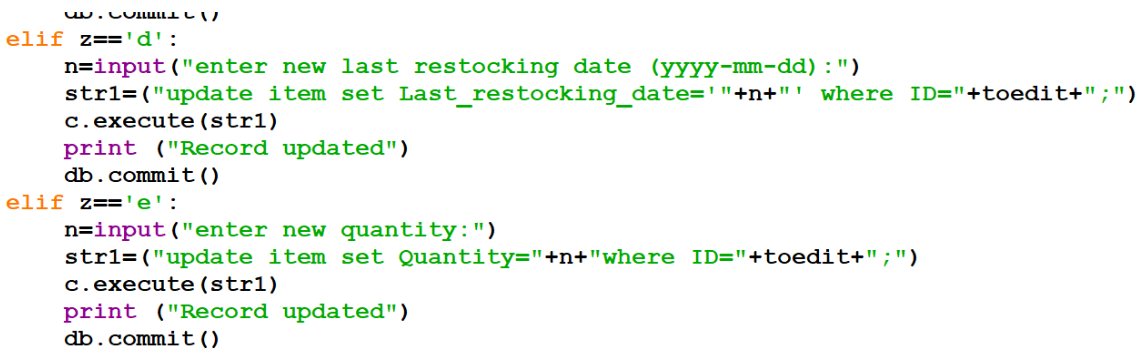
As per user’s requirement edits can be made to the previous information stored in a record. The user is needed to choose a valid option from the following. Information is updated according to the selected option entered by the user as input. If, elif and else commands are used for decision making.

Code:

```
toedit=input("Enter item ID of record to edit: ")
    print ("a-to edit name")
    print ("b-to edit price")
    print ("c-to edit item id")
    print ("d-to edit last restocking date")
    print ("e-to edit stock quantity")
    z=input("Enter a-e: ")
    if z=='a':
        n=input("enter new name:")
        str1=("update item set Name='"+n+"' where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='b':
        n=input("enter new price:")
        str1=("update item set Product_price="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='c':
        n=input("enter new ID:")
        str1=("update item set ID="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='d':
        n=input("enter new last restocking date (yyyy-mm-dd):")
        str1=("update item set Last_restocking_date='"+n+"' where
ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='e':
        n=input("enter new quantity:")
        str1=("update item set Quantity="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
```



```
#OPTIONS TO CHOOSE THE FEILD/ATTRIBUTE TO EDIT
    toedit=input("Enter item ID of record to edit: ")
    print ("a-to edit name")
    print ("b-to edit price")
    print ("c-to edit item id")
    print ("d-to edit last restocking date")
    print ("e-to edit stock quantity")
    z=input("Enter a-e: ")
    if z=='a':
        n=input("enter new name:")
        str1=("update item set Name='"+n+"' where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='b':
        n=input("enter new price:")
        str1=("update item set Product_price="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='c':
        n=input("enter new ID:")
        str1=("update item set ID="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
```



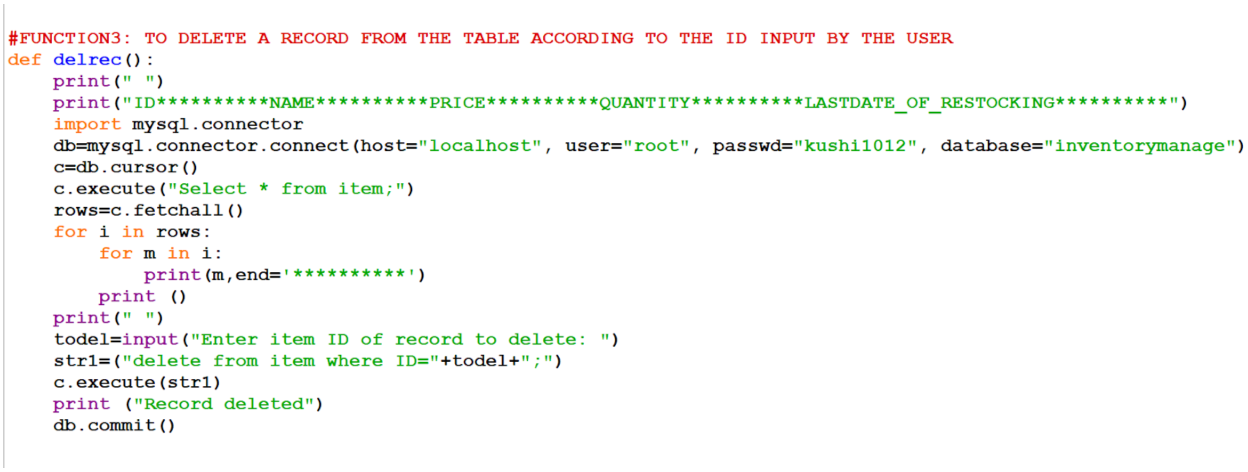
```
        db.commit()
    elif z=='d':
        n=input("enter new last restocking date (yyyy-mm-dd):")
        str1=("update item set Last_restocking_date='"+n+"' where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
    elif z=='e':
        n=input("enter new quantity:")
        str1=("update item set Quantity="+n+"where ID="+toedit+";")
        c.execute(str1)
        print ("Record updated")
        db.commit()
```

- **FUNCTION3: TO DELETE A RECORD FROM THE TABLE**

According to the item id entered by the user in the form of input, the record will be found. The selected record corresponding to the entered item id would be deleted.

Code:

```
def delrec():
   print(" ")

print("ID**********NAME**********PRICE**********QUANTITY*******
***LASTDATE_OF_RESTOCKING**********")
   import mysql.connector
   db=mysql.connector.connect(host="localhost", user="root",
passwd="kushi1012", database="inventorymanage")
   c=db.cursor()
   c.execute("Select * from item;")
   rows=c.fetchall()
   for i in rows:
      for m in i:
         print(m,end='**********')
      print ()
   print(" ")
   todel=input("Enter item ID of record to delete: ")
   str1=("delete from item where ID="+todel+";")
   c.execute(str1)
   print ("Record deleted")
   db.commit()
```

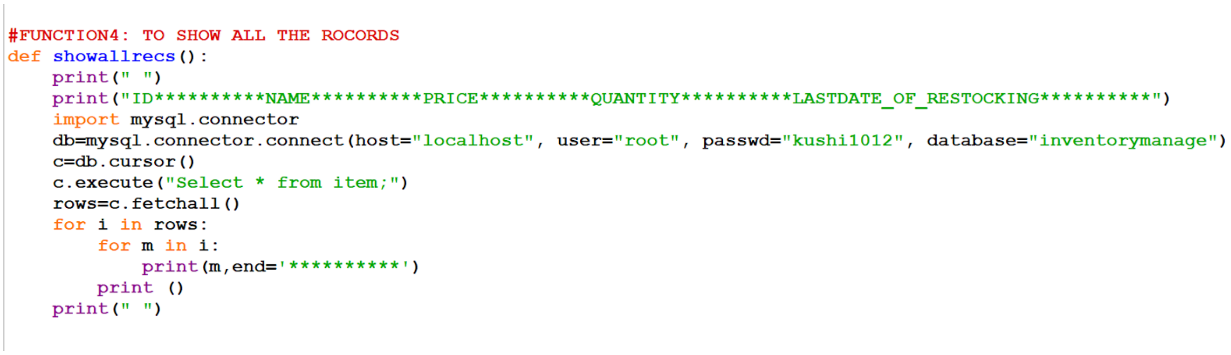


```
#FUNCTION3: TO DELETE A RECORD FROM THE TABLE ACCORDING TO THE ID INPUT BY THE USER
def delrec():
    print(" ")
    print("ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**********")
    import mysql.connector
    db=mysql.connector.connect(host="localhost", user="root", passwd="kushi1012", database="inventorymanage")
    c=db.cursor()
    c.execute("Select * from item;")
    rows=c.fetchall()
    for i in rows:
        for m in i:
            print(m,end='**********')
        print ()
    print(" ")
    todel=input("Enter item ID of record to delete: ")
    str1=("delete from item where ID="+todel+";")
    c.execute(str1)
    print ("Record deleted")
    db.commit()
```

- **<u>FUNCTION4: TO SHOW ALL THE RECORDS</u>**

<u>Code:</u>

```
def showallrecs():
    print(" ")
print("ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**********")
    import mysql.connector
    db=mysql.connector.connect(host="localhost", user="root", passwd="kushi1012", database="inventorymanage")
    c=db.cursor()
    c.execute("Select * from item;")
    rows=c.fetchall()
    for i in rows:
        for m in i:
            print(m,end='**********')
        print ()
    print(" ")
```

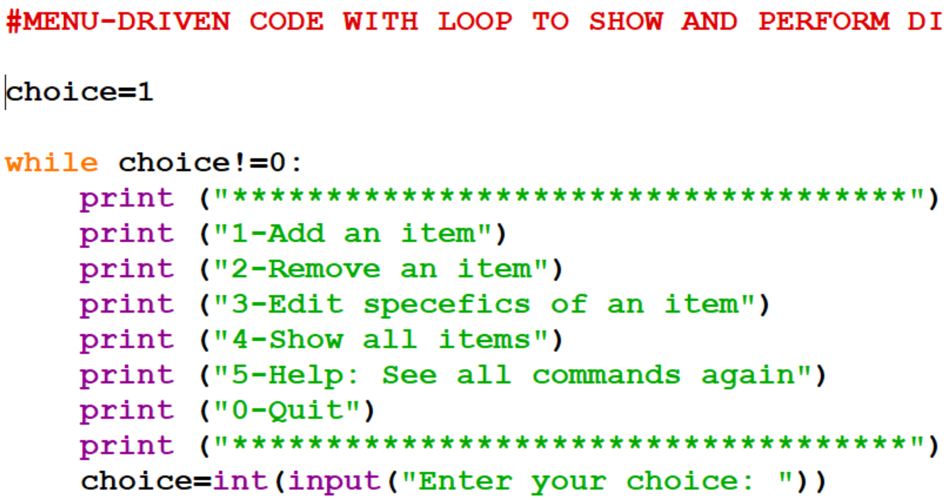


```
#FUNCTION4: TO SHOW ALL THE ROCORDS
def showallrecs():
    print(" ")
    print("ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**********")
    import mysql.connector
    db=mysql.connector.connect(host="localhost", user="root", passwd="kushi1012", database="inventorymanage")
    c=db.cursor()
    c.execute("Select * from item;")
    rows=c.fetchall()
    for i in rows:
        for m in i:
            print(m,end='**********')
        print ()
    print(" ")
```

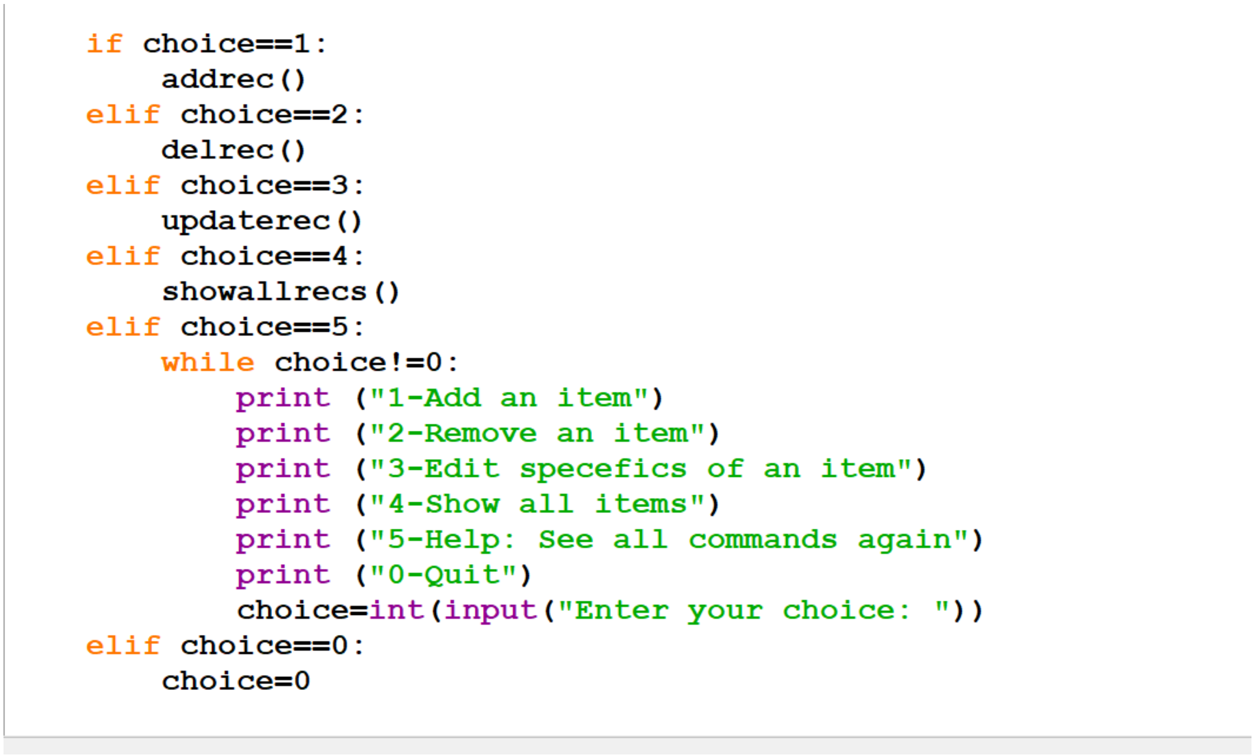
- **<u>MENU-DRIVEN CODE WITH LOOP TO SHOW AND PERFORM DIFFERENT TASKS</u>**

<u>Code:</u>

```
choice=1

while choice!=0:
   print ("************************************")
   print ("1-Add an item")
   print ("2-Remove an item")
   print ("3-Edit specefics of an item")
   print ("4-Show all items")
   print ("5-Help: See all commands again")
   print ("0-Quit")
   print ("************************************")
   choice=int(input("Enter your choice: "))


   if choice==1:
      addrec()
   elif choice==2:
    delrec()
  elif choice==3:
    updaterec()
  elif choice==4:
    showallrecs()
  elif choice==5:
    while choice!=0:
      print ("1-Add an item")
      print ("2-Remove an item")
      print ("3-Edit specefics of an item")
      print ("4-Show all items")
      print ("5-Help: See all commands again")
      print ("0-Quit")
      choice=int(input("Enter your choice: "))
  elif choice==0:
    choice=0
```



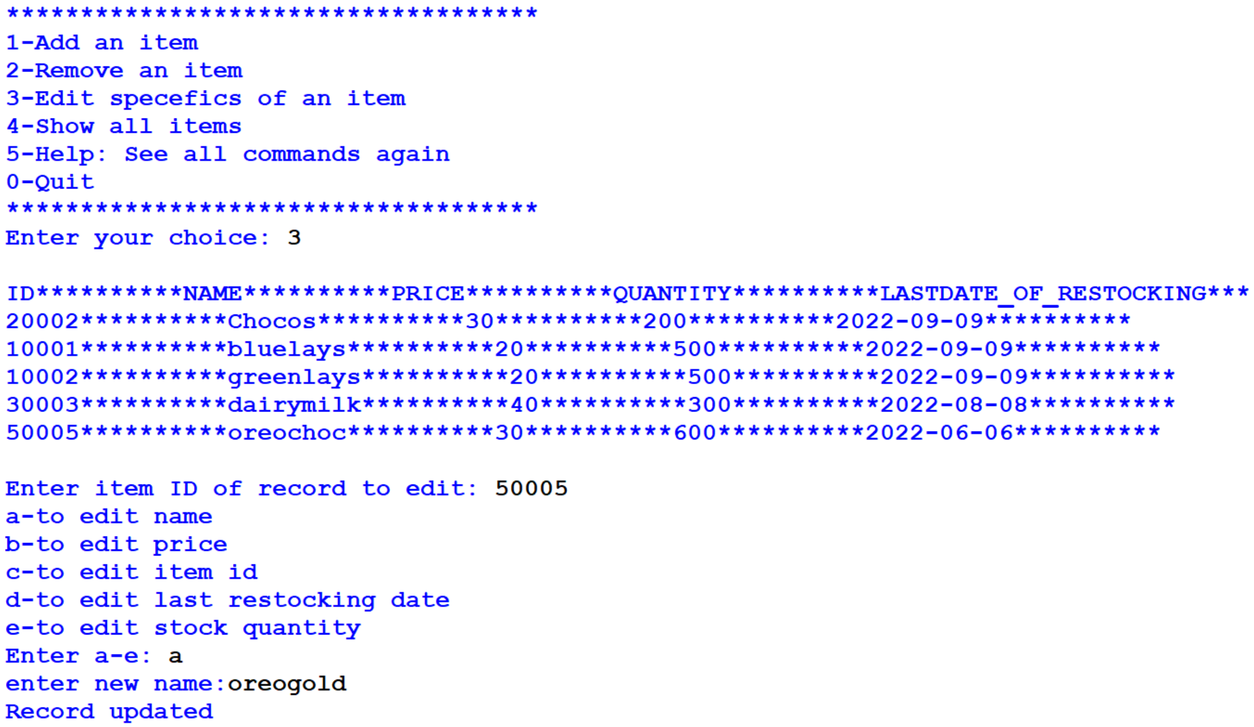
```
#MENU-DRIVEN CODE WITH LOOP TO SHOW AND PERFORM DIFFERENT TASKS

choice=1

while choice!=0:
    print ("************************************")
    print ("1-Add an item")
    print ("2-Remove an item")
    print ("3-Edit specefics of an item")
    print ("4-Show all items")
    print ("5-Help: See all commands again")
    print ("0-Quit")
    print ("************************************")
    choice=int(input("Enter your choice: "))
```



```python
if choice==1:
    addrec()
elif choice==2:
    delrec()
elif choice==3:
    updaterec()
elif choice==4:
    showallrecs()
elif choice==5:
    while choice!=0:
        print ("1-Add an item")
        print ("2-Remove an item")
        print ("3-Edit specefics of an item")
        print ("4-Show all items")
        print ("5-Help: See all commands again")
        print ("0-Quit")
        choice=int(input("Enter your choice: "))
elif choice==0:
    choice=0
```

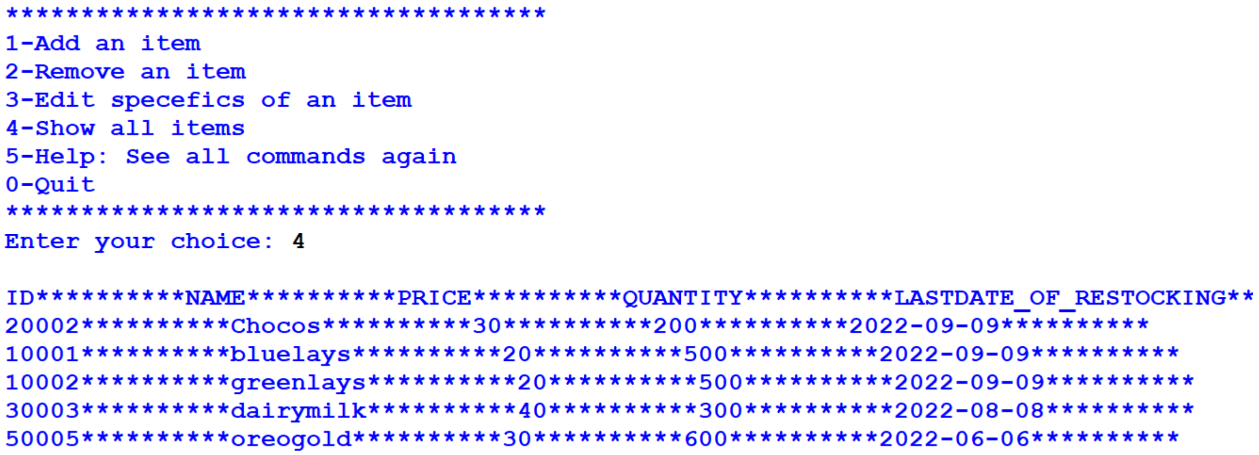
Output & Implementation

Option 3 is selected by the user. According to the item id the corresponding record is selected. As the user has selected option a the edited name is entered by the user and the record is updated.



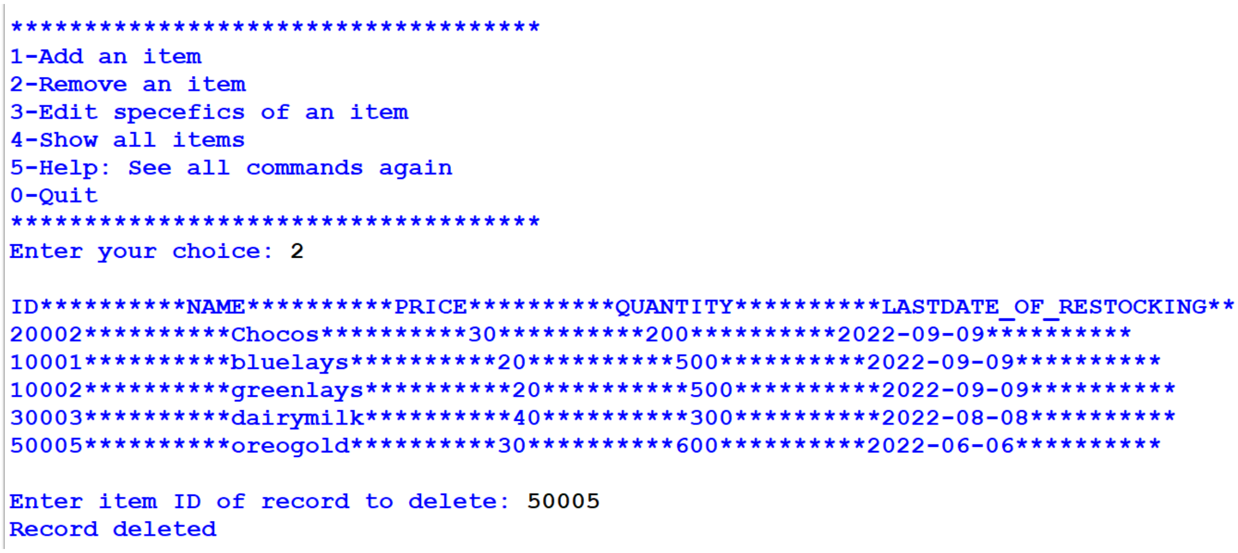
```
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 3

ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING***
20002**********Chocos**********30**********200**********2022-09-09**********
10001**********bluelays**********20**********500**********2022-09-09**********
10002**********greenlays**********20**********500**********2022-09-09**********
30003**********dairymilk**********40**********300**********2022-08-08**********
50005**********oreochoc**********30**********600**********2022-06-06**********

Enter item ID of record to edit: 50005
a-to edit name
b-to edit price
c-to edit item id
d-to edit last restocking date
e-to edit stock quantity
Enter a-e: a
enter new name:oreogold
Record updated
```

Option 4 is selected by the user. Hence, all items and information regarding those items is displayed.



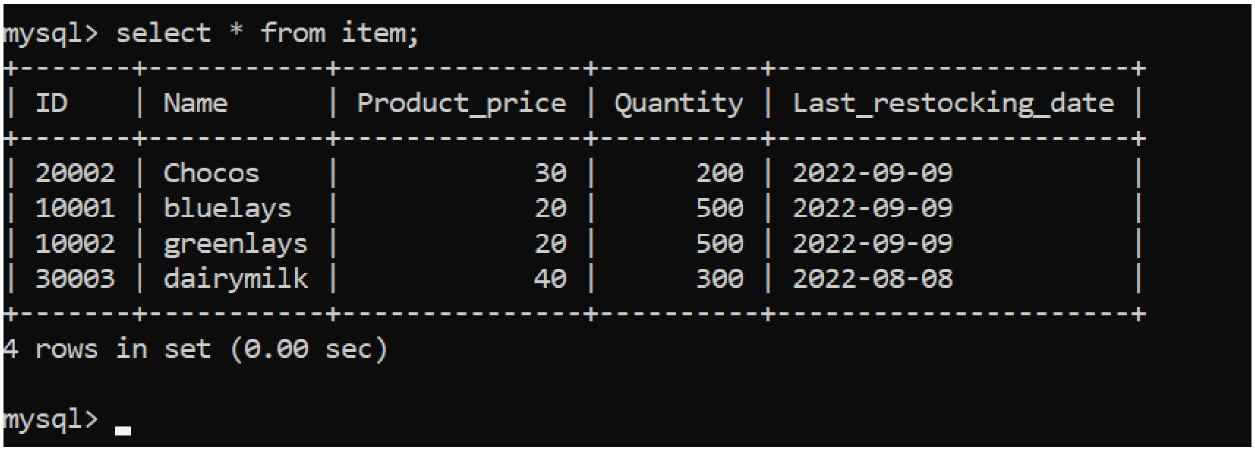
```
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 4

ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**
20002**********Chocos**********30**********200**********2022-09-09**********
10001**********bluelays**********20**********500**********2022-09-09**********
10002**********greenlays**********20**********500**********2022-09-09**********
30003**********dairymilk**********40**********300**********2022-08-08**********
50005**********oreogold**********30**********600**********2022-06-06**********
```

Option 2 is selected by the user. According to the item id entered by the user, the record corresponding to that item id is deleted.



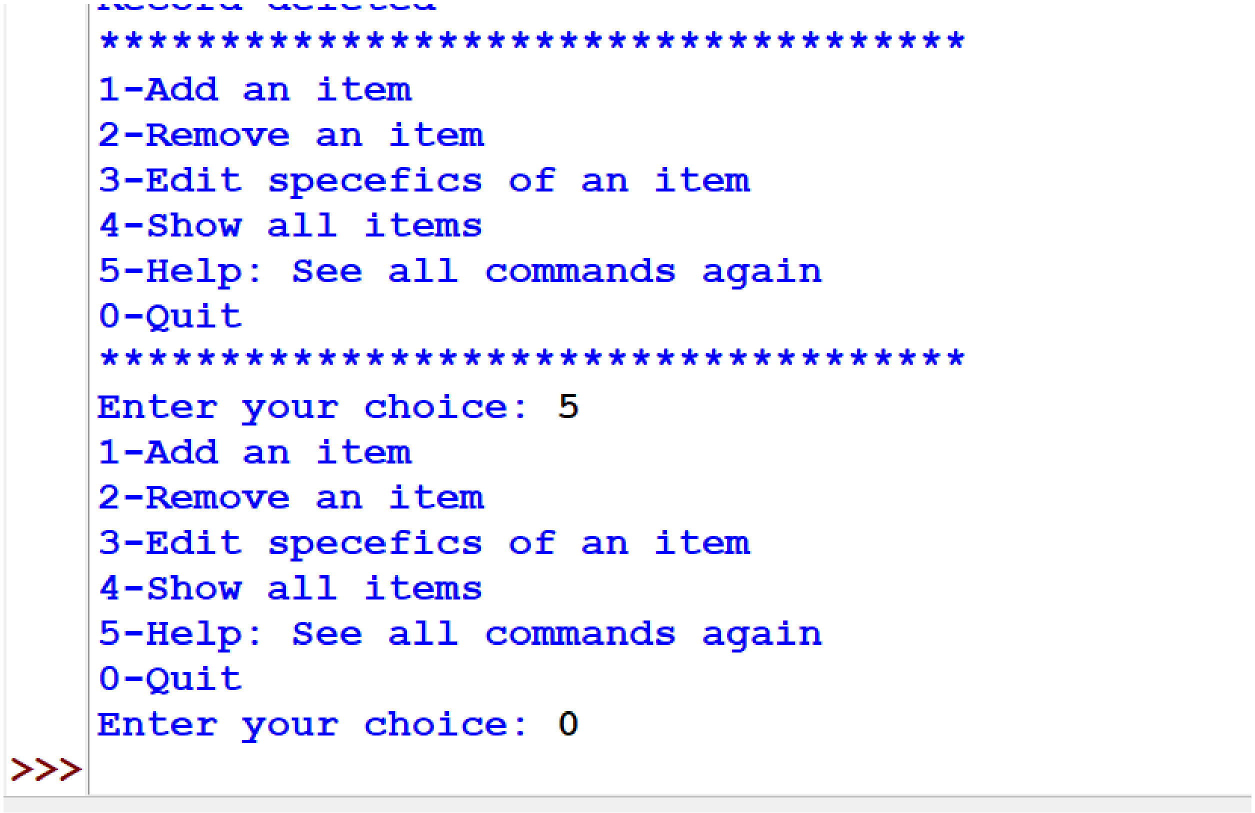
```
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 2

ID**********NAME**********PRICE**********QUANTITY**********LASTDATE_OF_RESTOCKING**
20002**********Chocos**********30**********200**********2022-09-09**********
10001**********bluelays**********20**********500**********2022-09-09**********
10002**********greenlays**********20**********500**********2022-09-09**********
30003**********dairymilk**********40**********300**********2022-08-08**********
50005**********oreogold**********30**********600**********2022-06-06**********

Enter item ID of record to delete: 50005
Record deleted
```

Using the select query it can be inferred that the record is deleted from the table item.



```
mysql> select * from item;
+-------+-----------+---------------+----------+----------------------+
| ID    | Name      | Product_price | Quantity | Last_restocking_date |
+-------+-----------+---------------+----------+----------------------+
| 20002 | Chocos    |            30 |      200 | 2022-09-09           |
| 10001 | bluelays  |            20 |      500 | 2022-09-09           |
| 10002 | greenlays |            20 |      500 | 2022-09-09           |
| 30003 | dairymilk |            40 |      300 | 2022-08-08           |
+-------+-----------+---------------+----------+----------------------+
4 rows in set (0.00 sec)

mysql> _
```

Lastly if Option 0 is selected the code stops running.



```
Record deleted
************************************
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
************************************
Enter your choice: 5
1-Add an item
2-Remove an item
3-Edit specefics of an item
4-Show all items
5-Help: See all commands again
0-Quit
Enter your choice: 0
>>>
```

Bibliography

- Computer science with python - Sumita Arora class XI
- Computer science with python - Preeti Arora class XII
- www.tutorial.com
- www.w3school.com