

COMPUTER SCIENCE PROJECT 2020-21



Developed By:

Mukund Sarda (12-H)

Kartik Singh (12-H)

St. Mark's Sr. Sec. Public School Meera Bagh, New Delhi



INDEX

S.No.	Topic	Page No.
1	Certificate	3
2	Acknowledgement	4
3	References	5
4	Introduction	6
5	General Information	7
6	Source Code	8-20
7	Outputs	21-47
8	Requirements	48

CERTIFICATE

This is to certify that Mukund Sarda of class XII-H has developed the project on the topic **“GROCERY SHOP MANAGEMENT SYSTEM”**.

The project is found worthy of acceptance as final project for the subject computer science of class XII. The project has been done under my supervision in the computer lab of St. Mark's Sr. Sec. Public School, Meera Bagh in the session 2020-2021. The work done by him is original.

Mr. Naveen Gupta
(PGT Computer Science)

Date: _____

ACKNOWLEDGEMENT

I, **MUKUND SARDA**, of class **XII-H**, would like to express a deep sense of thanks and gratitude to **Mr. Naveen Gupta**, my computer science teacher, for guiding me immensely through the course of the project. His untiring help, constructive advice, constant motivation and encouragement is the reason for success of the project. I also thank him for teaching us computer science from the very basics to the complex which not only strengthened my knowledge but also helped me completing the project.

I would like to thank my parents for their motivation and support. I would also like to thank my friends for their valuable suggestions and timely help and support.

REFERENCE

I am highly indebted to these sources of information that helped me a lot to improve my project:

1. Together With Computer Science Python
2. Class Notes
3. NCERT Computer Science Textbook
4. www.google.com
5. wiki.python.org
6. www.python.org
7. www.mysql.com

INTRODUCTION

Grocery shops are present everywhere across the country. But have you ever thought how these shops are managed? Management of these shops with the old existing manual management system has become a tedious and tiresome work due to increase in number of both the customers and variety of products. Thus, in order to avoid errors while billings of items and challenges in managing the information of products, suppliers, customers, etc that prevails in the old management system, I have created a new “Grocery Shop Management System”. The purpose of this system is to automate the existing system by the help of software which helps the shopkeeper fulfilling all his requirements by providing all types of create, read and update operations. It also helps to store the data and information in form of computerised records for a longer period of time with easy accessing and manipulation of the same. This in turn would make the management more efficient by reducing the manual work and also provide better utilization of resources. It will also help in providing better services to the customers.

GENERAL INFORMATION

1. Python Program:

- a. Imported Libraries: pymysql
- b. Number Of Functions: 30
- c. Number Of Lines: 670

2. MySQL Database:

- a. Database Name: Grocery_Shop_Management
- b. Number Of Tables: 10 (Bakery, Dairy, Cereals_And_Spices, Beverages, Fruits_And_Vegetables, Bathroom_And_Toiletries, Snacks, Orders, Sales, Users)

SOURCE CODE

```
# GROCERY SHOP MANAGEMENT SYSTEM
# Version 1.0
# Mukund Sarda & Kartik Singh (12-H)
# St. Mark's Sr. Sec. Public School, Meera Bagh

import pymysql

def userid():
    print()

    a=input("Are you an existing user:")

    if a=='Y' or a=='y' or a=='Yes' or a=='YES' or a=='yes':
        existinguser()

    elif a=='N' or a=='n' or a=='No' or a=='NO' or a=='no':
        h=input("Enter the product key:")

        if h=="MSKS2021":
            print("Access Granted")

            i=input("Enter the new User ID :")
            j=input("Enter the new Password :")

            k="INSERT INTO Users(User_Id,Password) VALUES('%s','%s')"%(i,j)
            cur.execute(k)
            db.commit()

            print("New User Created")

            existinguser()

        else:
            print("Access Denied")
            print("Incorrect Product Key")

    else:
        print("Invalid Input")

    return

def existinguser():
    print()

    b=input("Enter User ID:")
```



```

c=input("Enter Password:")

try:
    d="SELECT User_Id FROM Users WHERE Password='%s'"%(c)
    cur.execute(d)
    e=cur.fetchall()
    h=e[0][0]
    i=h

    f="SELECT Password FROM Users WHERE User_Id='%s'"%(b)
    cur.execute(f)
    g=cur.fetchall()
    j=g[0][0]
    k=j

    if b==i and c==k:
        print("Successfully Connected")
        print()

        final_1()

except:
    print("User ID or Password must be incorrect.")

return

def categories():
    print()

    print("1. Bakery")
    print("2. Dairy")
    print("3. Cereals_And_Spices")
    print("4. Beverages")
    print("5. Fruits_And_Vegetables")
    print("6. Bathroom_And_Toiletries")
    print("7. Snacks")

    return

def menu1():
    print()

    print("WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:")
    categories()

    return

def menu2(k):
    print()

```

```

print("THINGS TO DO:")
print("1. Add Products To",k)
print("2. Update Products In",k)
print("3. To Check Which Product Have Less Quantity In",k)
print("4. To View All Products Of",k)
print("5. To Sell Products Of",k)
print("6. To Manage Order List")
print("7. To Manage Sales List")

return

def add(k):
    print()

    a=input("Enter the Product Code:")
    b=input("Enter the Product Name:")
    c=int(input("Enter the Cost Price:"))
    f=int(input("Enter the Selling Price:"))
    d=int(input("Enter the Quantity:"))

    e="INSERT INTO "+k+"(Product_Code,Product_Name,Cost_Price,Quantity,Selling_Price)
VALUES('%s','%s','%s','%s','%s')"%(a,b,c,d,f)
    cur.execute(e)
    db.commit()

    return

def add_final(k):
    print()

    ans='yes'

    while ans=='yes' or ans=='Yes' or ans=='Y' or ans=='y' or ans=='YES':
        add(k)
        print()
        ans=input("Do You Want To Add More Products (Yes/No):")

    return

def updatemenu():
    print()

    print("THINGS TO DO:")
    print("1. To Update The Cost Price")
    print("2. To Update The Selling Price")
    print("3. To Update The Quantity")
    print("4. To Delete A Product")

    return

def update_cost_price(k):

```

```
print()
```

```
a=input("Enter The Product Name For Which Cost Has To Be Updated:")
```

```
b=int(input("Enter The Updated Cost:"))
```

```
c="UPDATE "+k+" SET Cost_Price='%s' WHERE Product_Name='%s'"%(b,a)
```

```
d=cur.execute(c)
```

```
db.commit()
```

```
return
```

```
def update_selling_price(k):
```

```
    print()
```

```
    a=input("Enter The Product Name For Which Selling Price Has To Be Updated:")
```

```
    b=int(input("Enter The Updated Cost:"))
```

```
    c="UPDATE "+k+" SET Selling_Price='%s' WHERE Product_Name='%s'"%(b,a)
```

```
    d=cur.execute(c)
```

```
    db.commit()
```

```
    return
```

```
def update_quantity(k):
```

```
    print()
```

```
    a=input("Enter The Product Name For Which Quantity Has To Be Updated:")
```

```
    b=int(input("Enter The Updated Quantity:"))
```

```
    c="UPDATE "+k+" SET Quantity='%s' WHERE Product_Name='%s'"%(b,a)
```

```
    d=cur.execute(c)
```

```
    db.commit()
```

```
    return
```

```
def delete_product(k):
```

```
    print()
```

```
    a=input("Enter The Product Name Which Has To Be Deleted:")
```

```
    b="DELETE FROM "+k+" WHERE Product_Name='%s'"%(a)
```

```
    c=cur.execute(b)
```

```
    db.commit()
```

```
    return
```

```
def less_quantity(k):
```

```
    print()
```

```
    a=int(input("Enter The Minimum Quantity:"))
```

```

print("The "+k+" Products Whose Quantity Is Less Than",a,"Are:")
b="SELECT * FROM "+k+" WHERE Quantity<'%s'"%(a)
c=cur.execute(b)
d=cur.fetchall()

for i in d:
    print(i)

return

def less_quan_order(k):
    print()

    a=input("Do You Want To Place Order Of These Products:")

    if a=='Y' or a=='y' or a=='Yes' or a=='YES' or a=='yes':
        order_add_final(k)

    elif a=='N' or a=='n' or a=='No' or a=='NO' or a=='no':
        print()

    else:
        print("Incorrect Input")

    return

def view_all(k):
    print()

    a="SELECT * FROM "+k
    b=cur.execute(a)
    c=cur.fetchall()

    for i in c:
        print(i)

    return

def order_menu():
    print()

    print("THINGS TO DO:")
    print("1. Add Order To Order List")
    print("2. To Update The Quantity Of A Order")
    print("3. To Delete A Order")
    print("4. To View Complete Order List")

    return

def order_add(k):
    print()

```

```
a=input("Enter The Name Of The Product To Be Ordered:")
```

```
b=int(input("Enter The Quantity:"))
```

```
d="SELECT Product_Code FROM "+k+" WHERE Product_Name='%s'"%(a)
```

```
cur.execute(d)
```

```
e=cur.fetchall()
```

```
f=e[0][0]
```

```
g=f
```

```
h="SELECT Cost_Price FROM "+k+" WHERE Product_Name='%s'"%(a)
```

```
cur.execute(h)
```

```
i=cur.fetchall()
```

```
j=i[0][0]
```

```
l=j
```

```
m=b*l
```

```
n="INSERT INTO Orders(Product_Category,Product_Name,Quantity,Product_Code,Cost_Price,Amount)
VALUES('%s','%s','%s','%s','%s','%s')"%(k,a,b,g,l,m)
```

```
cur.execute(n)
```

```
db.commit()
```

```
return
```

```
def order_add_final(k):
```

```
    print()
```

```
    ans='yes'
```

```
    while ans=='yes' or ans=='Yes' or ans=='Y' or ans=='y' or ans=='YES':
```

```
        order_add(k)
```

```
        print()
```

```
        ans=input("Do You Want To Add More Products (Yes/No):")
```

```
    return
```

```
def order_update_quantity():
```

```
    print()
```

```
    a=input("Enter The Name Of The Product Whose Quantity Has To Be Updated:")
```

```
    b=int(input("Enter The Updated Quantity:"))
```

```
    c="UPDATE Orders SET Quantity='%s' WHERE Product_Name='%s'"%(b,a)
```

```
    d=cur.execute(c)
```

```
    db.commit()
```

```
    h="SELECT Cost_Price FROM Orders WHERE Product_Name='%s'"%(a)
```

```
    cur.execute(h)
```

```
    i=cur.fetchall()
```

```
    j=i[0][0]
```

```
    l=j
```

```
m=b*I
```

```
e="UPDATE Orders SET Amount='%s' WHERE Product_Name='%s'"%(m,a)
cur.execute(e)
db.commit()
```

```
return
```

```
def order_delete():
    print()
```

```
a=input("Enter The Name Of The Product Whose Order Is To Be Deleted:")
```

```
b="DELETE FROM Orders WHERE Product_Name='%s'"%(a)
c=cur.execute(b)
db.commit()
```

```
return
```

```
def order_view_all():
    print()
```

```
a="SELECT * FROM Orders"
b=cur.execute(a)
c=cur.fetchall()
```

```
for i in c:
    print(i)
```

```
return
```

```
def sales_menu():
    print()
```

```
print("THINGS TO DO:")
print("1. Add Record To Sales List")
print("2. To Delete Sales Before A Particular Date")
print("3. To View Complete Sales List")
print("4. To View Total Profit Earned")
```

```
return
```

```
def sales_add(k):
    print()
```

```
a=input("Enter The Name Of The Product To Be Sold:")
b=int(input("Enter The Quantity:"))
```

```
d="SELECT Product_Code FROM "+k+" WHERE Product_Name='%s'"%(a)
cur.execute(d)
e=cur.fetchall()
```

```
f=e[0][0]
g=f
```

```
h="SELECT Cost_Price FROM "+k+" WHERE Product_Name='%s'"%(a)
cur.execute(h)
i=cur.fetchall()
j=i[0][0]
l=j
```

```
m="SELECT Selling_Price FROM "+k+" WHERE Product_Name='%s'"%(a)
cur.execute(m)
n=cur.fetchall()
o=n[0][0]
p=o
```

```
q=p*b
r=l*b
s=q-r
```

```
t="INSERT INTO
Sales(Product_Category,Product_Name,Quantity,Product_Code,Selling_Price,Amount,Profit,Date)
VALUES('%s','%s','%s','%s','%s','%s','%s',CURDATE())"%(k,a,b,g,p,q,s)
cur.execute(t)
db.commit()
```

```
u="UPDATE "+k+" SET Quantity=Quantity-'%s' WHERE Product_Name='%s'"%(b,a)
cur.execute(u)
db.commit()
```

```
return
```

```
def sales_add_final(k):
    print()
```

```
ans='yes'
```

```
while ans=='yes' or ans=='Yes' or ans=='Y' or ans=='y' or ans=='YES':
    sales_add(k)
    print()
    ans=input("Do You Want To Sell More Products (Yes/No):")
```

```
return
```

```
def delete_sales():
```

```
a=input("Enter The Year(YYYY):")
b=input("Enter The Month(MM):")
c=input("Enter The Date(DD):")
```

```
d=a+b+c
e=int(d)
```

```
f="DELETE FROM Sales WHERE Date<'%s'"%(e)
```

```
cur.execute(f)
db.commit()
```

```
return
```

```
def sales_view_all():
    print()
```

```
    a="SELECT * FROM Sales"
    b=cur.execute(a)
    c=cur.fetchall()
```

```
    for i in c:
        print(i)
```

```
return
```

```
def sales_profit():
    print()
```

```
    a="SELECT SUM(Profit) FROM Sales"
    cur.execute(a)
    b=cur.fetchall()
```

```
    print("Total Profit Earned:")
    print(b[0][0])
```

```
return
```

```
def end():
    print()
```

```
    print("~~~~~")
    print("Thank You For Using Our GROCERY SHOP MANAGEMENT SYSTEM")
    print("~~~~~")
```

```
    print()
```

```
    print("""BY:
Mukund Sarda
Kartik Singh
(XII-H)""")
    return
```

```
def final_1():
    menu1()
    print()
```

```
    a=int(input("Enter Your Choice:"))
```

```
    if a==1:
```



```
k="Bakery"
elif a==2:
    k="Dairy"
elif a==3:
    k="Cereals_And_Spices"
elif a==4:
    k="Beverages"
elif a==5:
    k="Fruits_And_Vegetables"
elif a==6:
    k="Bathroom_And_Toiletries"
elif a==7:
    k="Snacks"
else:
    print("Incorrect Input")

print()

menu2(k)

print()

c=int(input("Enter Your Choice:"))

if c==1:
    add_final(k)

elif c==2:
    updatemenu()
    print()

    d=int(input("Enter Your Choice:"))

    if d==1:
        update_cost_price(k)

    elif d==2:
        update_selling_price(k)

    elif d==3:
        update_quantity(k)

    elif d==4:
        delete_product(k)

    else:
        ("Incorrect Input!")

elif c==3:
    less_quantity(k)

print()
```

```
less_quan_order(k)

elif c==4:
    view_all(k)

elif c==5:
    sales_add_final(k)

elif c==6:
    order_menu()
    print()

    e=int(input("Enter Your Choice:"))

    if e==1:
        order_add_final(k)

    elif e==2:
        order_update_quantity()

    elif e==3:
        order_delete()

    elif e==4:
        order_view_all()

    else:
        ("Incorrect Input!")

elif c==7:
    sales_menu()
    print()

    f=int(input("Enter Your Choice:"))

    if f==1:
        sales_add_final(k)

    elif f==2:
        delete_sales()

    elif f==3:
        sales_view_all()

    elif f==4:
        sales_profit()

    else:
        ("Incorrect Input!")
```

```

else:
    print("Incorrect Input!")

return

def grocery_shop_management():
    z="yes"

    while z=="yes" or z=="Yes" or z=="Y" or z=="y" or z=="YES":
        userid()

        print()

        z=input("Do You Want To Run The Program Again (Yes/No):")

    else:
        end()

    return

print("~~~~~")
print("WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM")
print("~~~~~")

print()

print("-----")
print("PRODUCT CATEGORIES AVAILABLE")
print("-----")

categories()

print()

x=input("Enter The Password Of Your Database:")

dr=pymysql.connect(host="localhost",user="root",password=x)
cr=dr.cursor()

cr.execute("CREATE DATABASE IF NOT EXISTS Grocery_Shop_Management")

db=pymysql.connect(host="localhost",user="root",password=x,db="Grocery_Shop_Management")
cur=db.cursor()

cur.execute("CREATE TABLE IF NOT EXISTS Bakery(Product_Code VARCHAR(50),Product_Name
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")

cur.execute("CREATE TABLE IF NOT EXISTS Dairy(Product_Code VARCHAR(50),Product_Name
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")

cur.execute("CREATE TABLE IF NOT EXISTS Cereals_And_Spices(Product_Code VARCHAR(50),Product_Name
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")

```

```
cur.execute("CREATE TABLE IF NOT EXISTS Beverages(Product_Code VARCHAR(50),Product_Name  
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Fruits_And_Vegetables(Product_Code VARCHAR(50),Product_Name  
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Bathroom_And_Toiletries(Product_Code  
VARCHAR(50),Product_Name VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Snacks(Product_Code VARCHAR(50),Product_Name  
VARCHAR(50),Cost_Price INT,Selling_Price INT,Quantity INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Orders(Product_Category VARCHAR(50),Product_Code  
VARCHAR(50),Product_Name VARCHAR(50),Quantity INT,Cost_Price INT,Amount INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Sales(Product_Category VARCHAR(50),Product_Code  
VARCHAR(50),Product_Name VARCHAR(50),Date DATE,Quantity INT,Selling_Price INT,Amount INT,Profit INT);")
```

```
cur.execute("CREATE TABLE IF NOT EXISTS Users(User_Id VARCHAR(50),Password VARCHAR(50));")
```

```
print("! Database Created Successfully !")
```

```
grocery_shop_management()
```

```
db.commit()
```

OUTPUTS

A. Database Tables:

```
mysql> USE Grocery_Shop_Management;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_grocery_shop_management |
+-----+
| bakery                             |
| bathroom_and_toiletries            |
| beverages                          |
| cereals_and_spices                 |
| dairy                              |
| fruits_and_vegetables              |
| orders                             |
| sales                              |
| snacks                             |
| users                              |
+-----+
10 rows in set (0.02 sec)
```

B. Tables Description:

```
mysql> DESC Bakery;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Product_Code   | varchar(50)   | YES  |     | NULL     |       |
| Product_Name   | varchar(50)   | YES  |     | NULL     |       |
| Cost_Price     | int           | YES  |     | NULL     |       |
| Selling_Price  | int           | YES  |     | NULL     |       |
| Quantity       | int           | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> DESC Dairy;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Product_Code   | varchar(50)   | YES  |     | NULL     |       |
| Product_Name   | varchar(50)   | YES  |     | NULL     |       |
| Cost_Price     | int           | YES  |     | NULL     |       |
| Selling_Price  | int           | YES  |     | NULL     |       |
| Quantity       | int           | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> DESC Cereals_And_Spices;
```

Field	Type	Null	Key	Default	Extra
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Cost_Price	int	YES		NULL	
Selling_Price	int	YES		NULL	
Quantity	int	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESC Beverages;
```

Field	Type	Null	Key	Default	Extra
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Cost_Price	int	YES		NULL	
Selling_Price	int	YES		NULL	
Quantity	int	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESC Fruits_And_Vegetables;
```

Field	Type	Null	Key	Default	Extra
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Cost_Price	int	YES		NULL	
Selling_Price	int	YES		NULL	
Quantity	int	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESC Bathroom_And_Toiletries;
```

Field	Type	Null	Key	Default	Extra
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Cost_Price	int	YES		NULL	
Selling_Price	int	YES		NULL	
Quantity	int	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESC Snacks;
```

Field	Type	Null	Key	Default	Extra
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Cost_Price	int	YES		NULL	
Selling_Price	int	YES		NULL	
Quantity	int	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESC Orders;
```

Field	Type	Null	Key	Default	Extra
Product_Category	varchar(50)	YES		NULL	
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Quantity	int	YES		NULL	
Cost_Price	int	YES		NULL	
Amount	int	YES		NULL	

6 rows in set (0.00 sec)

```
mysql> DESC Sales;
```

Field	Type	Null	Key	Default	Extra
Product_Category	varchar(50)	YES		NULL	
Product_Code	varchar(50)	YES		NULL	
Product_Name	varchar(50)	YES		NULL	
Date	date	YES		NULL	
Quantity	int	YES		NULL	
Selling_Price	int	YES		NULL	
Amount	int	YES		NULL	
Profit	int	YES		NULL	

8 rows in set (0.00 sec)

```
mysql> DESC Users;
```

Field	Type	Null	Key	Default	Extra
User_Id	varchar(50)	YES		NULL	
Password	varchar(50)	YES		NULL	

2 rows in set (0.00 sec)

C. Tables Data:

```
mysql> SELECT * FROM Bakery;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BA01	White Bread	20	25	17
BA02	Brown Bread	25	30	19
BA03	Fruit Cake	30	40	13
BA04	Burger Buns	30	35	15
BA05	Cookies	40	55	21
BA06	Biscuit	30	50	23

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Dairy;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
DA01	Milk	40	45	21
DA02	Butter	85	100	11
DA03	Cheese	80	95	13
DA04	Yogurt	35	40	18
DA05	Ice Cream	85	90	12
DA06	Buttermilk	20	25	19

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Cereals_And_Spices;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
CS01	Turmeric	30	40	11
CS02	Rice	90	110	15
CS03	Salt	18	20	21
CS04	Chickpeas	50	65	17
CS05	Pepper	35	45	13
CS06	Gram	60	75	19

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Beverages;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BE01	Soda	15	20	23
BE02	Juice	25	30	19
BE03	Soft Drinks	30	35	21
BE04	Lemonade	10	15	17
BE05	Coffee	15	25	20
BE06	Water	10	12	25

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Fruits_And_Vegetables;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
FV01	Orange	42	50	13
FV02	Banana	32	40	11
FV03	Onion	30	35	18
FV04	Tomato	20	25	14
FV05	Apple	65	80	12
FV06	Potato	15	20	16

6 rows in set (0.00 sec)


```
mysql> SELECT * FROM Bathroom_And_Toiletries;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BT01	Soap	45	60	23
BT02	Shampoo	210	250	13
BT03	Toothpaste	70	80	17
BT04	Face Wash	130	150	18
BT05	Hair Oil	85	100	15
BT06	Toothbrush	55	60	25

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Snacks;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
SN01	Chips	25	30	14
SN02	Chocolates	65	80	16
SN03	Nachos	35	40	19
SN04	Noodles	25	30	11
SN05	Soup	40	45	12
SN06	Candy	10	15	23

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455
Dairy	DA02	Butter	5	90	450
Bakery	BA03	Fruit Cake	5	30	150
Beverages	BE04	Lemonade	3	10	30
Snacks	SN04	Noodles	8	25	200
Bathroom_And_Toiletries	BT02	Shampoo	3	210	630

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Sales;
```

Product_Category	Product_Code	Product_Name	Date	Quantity	Selling_Price	Amount	Profit
Bakery	BA06	Biscuit	2021-02-10	3	50	150	60
Bathroom_And_Toiletries	BT01	Soap	2021-02-11	2	60	120	30
Snacks	SN03	Nachos	2021-02-11	4	40	160	20
Dairy	DA04	Yogurt	2021-02-11	1	40	40	5
Beverages	BE06	Water	2021-02-11	5	12	60	10
Fruits_And_Vegetables	FV03	Onion	2021-02-11	2	35	70	10

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Users;
+-----+-----+
| User_Id | Password |
+-----+-----+
| Mukund  | 14062003 |
| Kartik  | 22112002 |
+-----+-----+
2 rows in set (0.00 sec)
```

1. Database Creation:

```
~~~~~
WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM
~~~~~
```

```
-----
PRODUCT CATEGORIES AVAILABLE
-----
```

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

```
Enter The Password Of Your Database:m k s 15
! Database Created Successfully !
```

2. Non-Existing User:

```
~~~~~  
WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM  
~~~~~
```

```
-----  
PRODUCT CATEGORIES AVAILABLE  
-----
```

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

```
Enter The Password Of Your Database:m k s 15  
! Database Created Successfully !
```

```
Are you an existing user:No  
Enter the product key:MSKS2021  
Access Granted  
Enter the new User ID :Mukund  
Enter the new Password :14062003  
New User Created
```

```
Enter User ID:Mukund  
Enter Password:14062003  
Successfully Connected
```

3. Existing User:

```
~~~~~  
WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM  
~~~~~  
  
-----  
PRODUCT CATEGORIES AVAILABLE  
-----  
  
1. Bakery  
2. Dairy  
3. Cereals_And_Spices  
4. Beverages  
5. Fruits_And_Vegetables  
6. Bathroom_And_Toiletries  
7. Snacks  
  
Enter The Password Of Your Database:m k s 1 2  
! Database Created Successfully !  
  
Are you an existing user:Yes  
  
Enter User ID:Kartik  
Enter Password:22112002  
Successfully Connected  
  
WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:  
  
1. Bakery  
2. Dairy  
3. Cereals_And_Spices  
4. Beverages  
5. Fruits_And_Vegetables  
6. Bathroom_And_Toiletries  
7. Snacks  
  
Enter Your Choice:
```

4. Incorrect Product Key:

```
~~~~~  
WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM  
~~~~~  
  
-----  
PRODUCT CATEGORIES AVAILABLE  
-----  
  
1. Bakery  
2. Dairy  
3. Cereals_And_Spices  
4. Beverages  
5. Fruits_And_Vegetables  
6. Bathroom_And_Toiletries  
7. Snacks  
  
Enter The Password Of Your Database:mukesh152  
! Database Created Successfully !  
  
Are you an existing user:No  
Enter the product key:msks  
Access Denied  
Incorrect Product Key
```

5. Incorrect Password:

```
~~~~~  
WELCOME TO GROCERY SHOP MANAGEMENT SYSTEM  
~~~~~  
  
-----  
PRODUCT CATEGORIES AVAILABLE  
-----  
  
1. Bakery  
2. Dairy  
3. Cereals_And_Spices  
4. Beverages  
5. Fruits_And_Vegetables  
6. Bathroom_And_Toiletries  
7. Snacks  
  
Enter The Password Of Your Database:mukesh152  
! Database Created Successfully !  
  
Are you an existing user:Yes  
  
Enter User ID:Mukund  
Enter Password:1  
User ID or Password must be incorrect.
```

6. Add Products:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:1

THINGS TO DO:

1. Add Products To Bakery
2. Update Products In Bakery
3. To Check Which Product Have Less Quantity In Bakery
4. To View All Products Of Bakery
5. To Sell Products Of Bakery
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:1

Enter the Product Code:BA07

Enter the Product Name:Pastry

Enter the Cost Price:55

Enter the Selling Price:70

Enter the Quantity:12

Do You Want To Add More Products (Yes/No):No

```
mysql> SELECT * FROM Bakery;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BA01	White Bread	20	25	17
BA02	Brown Bread	25	30	19
BA03	Fruit Cake	30	40	13
BA04	Burger Buns	30	35	15
BA05	Cookies	40	55	21
BA06	Biscuit	30	50	23
BA07	Pastry	55	70	12

```
7 rows in set (0.00 sec)
```

7. Update:

a. Cost Price:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:2

THINGS TO DO:

1. Add Products To Dairy
2. Update Products In Dairy
3. To Check Which Product Have Less Quantity In Dairy
4. To View All Products Of Dairy
5. To Sell Products Of Dairy
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:2

THINGS TO DO:

1. To Update The Cost Price
2. To Update The Selling Price
3. To Update The Quantity
4. To Delete A Product

Enter Your Choice:1

Enter The Product Name For Which Cost Has To Be Updated:Butter

Enter The Updated Cost:90

```
mysql> SELECT * FROM Dairy;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
DA01	Milk	40	45	21
DA02	Butter	90	100	11
DA03	Cheese	80	95	13
DA04	Yogurt	35	40	18
DA05	Ice Cream	85	90	12
DA06	Buttermilk	20	25	19

6 rows in set (0.00 sec)

b. Selling Price:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:3

THINGS TO DO:

1. Add Products To Cereals_And_Spices
2. Update Products In Cereals_And_Spices
3. To Check Which Product Have Less Quantity In Cereals_And_Spices
4. To View All Products Of Cereals_And_Spices
5. To Sell Products Of Cereals_And_Spices
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:2

THINGS TO DO:

1. To Update The Cost Price
2. To Update The Selling Price
3. To Update The Quantity
4. To Delete A Product

Enter Your Choice:2

Enter The Product Name For Which Selling Price Has To Be Updated:Rice

Enter The Updated Cost:105

```
mysql> SELECT * FROM Cereals_And_Spices;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
CS01	Turmeric	30	40	11
CS02	Rice	90	105	15
CS03	Salt	18	20	21
CS04	Chickpeas	50	65	17
CS05	Pepper	35	45	13
CS06	Gram	60	75	19

```
6 rows in set (0.00 sec)
```


c. Quantity:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:4

THINGS TO DO:

1. Add Products To Beverages
2. Update Products In Beverages
3. To Check Which Product Have Less Quantity In Beverages
4. To View All Products Of Beverages
5. To Sell Products Of Beverages
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:2

THINGS TO DO:

1. To Update The Cost Price
2. To Update The Selling Price
3. To Update The Quantity
4. To Delete A Product

Enter Your Choice:3

Enter The Product Name For Which Quantity Has To Be Updated:Coffee
Enter The Updated Quantity:15

```
mysql> SELECT * FROM Beverages;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BE01	Soda	15	20	23
BE02	Juice	25	30	19
BE03	Soft Drinks	30	35	21
BE04	Lemonade	10	15	17
BE05	Coffee	15	25	15
BE06	Water	10	12	25

```
6 rows in set (0.00 sec)
```

d. Delete Product:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:1

THINGS TO DO:

1. Add Products To Bakery
2. Update Products In Bakery
3. To Check Which Product Have Less Quantity In Bakery
4. To View All Products Of Bakery
5. To Sell Products Of Bakery
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:2

THINGS TO DO:

1. To Update The Cost Price
2. To Update The Selling Price
3. To Update The Quantity
4. To Delete A Product

Enter Your Choice:4

Enter The Product Name Which Has To Be Deleted:Pastry

```
mysql> SELECT * FROM Bakery;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BA01	White Bread	20	25	17
BA02	Brown Bread	25	30	19
BA03	Fruit Cake	30	40	13
BA04	Burger Buns	30	35	15
BA05	Cookies	40	55	21
BA06	Biscuit	30	50	23

6 rows in set (0.00 sec)

8. Less Quantity:

a. Order Placed:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:5

THINGS TO DO:

1. Add Products To Fruits_And_Vegetables
2. Update Products In Fruits_And_Vegetables
3. To Check Which Product Have Less Quantity In Fruits_And_Vegetables
4. To View All Products Of Fruits_And_Vegetables
5. To Sell Products Of Fruits_And_Vegetables
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:3

Enter The Minimum Quantity:14

The Fruits_And_Vegetables Products Whose Quantity Is Less Than 14 Are:

('FV01', 'Orange', 42, 50, 13)
('FV02', 'Banana', 32, 40, 11)
('FV05', 'Apple', 65, 80, 12)

Do You Want To Place Order Of These Products:Yes

Enter The Name Of The Product To Be Ordered:Apple

Enter The Quantity:7

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455

1 row in set (0.00 sec)

b. Order Not Placed:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:6

THINGS TO DO:

1. Add Products To Bathroom_And_Toiletries
2. Update Products In Bathroom_And_Toiletries
3. To Check Which Product Have Less Quantity In Bathroom_And_Toiletries
4. To View All Products Of Bathroom_And_Toiletries
5. To Sell Products Of Bathroom_And_Toiletries
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:3

Enter The Minimum Quantity:18

The Bathroom_And_Toiletries Products Whose Quantity Is Less Than 18 Are:

('BT02', 'Shampoo', 210, 250, 13)
('BT03', 'Toothpaste', 70, 80, 17)
('BT05', 'Hair Oil', 85, 100, 15)

Do You Want To Place Order Of These Products:No

Do You Want To Run The Program Again (Yes/No):No

9. View All Products:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:7

THINGS TO DO:

1. Add Products To Snacks
2. Update Products In Snacks
3. To Check Which Product Have Less Quantity In Snacks
4. To View All Products Of Snacks
5. To Sell Products Of Snacks
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:4

```
('SN01', 'Chips', 25, 30, 14)
('SN02', 'Chocolates', 65, 80, 16)
('SN03', 'Nachos', 35, 40, 19)
('SN04', 'Noodles', 25, 30, 11)
('SN05', 'Soup', 40, 45, 12)
('SN06', 'Candy', 10, 15, 23)
```

10. Sell Product:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:1

THINGS TO DO:

1. Add Products To Bakery
2. Update Products In Bakery
3. To Check Which Product Have Less Quantity In Bakery
4. To View All Products Of Bakery
5. To Sell Products Of Bakery
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:5

Enter The Name Of The Product To Be Sold:Biscuit

Enter The Quantity:3

Do You Want To Sell More Products (Yes/No):No

```
mysql> SELECT * FROM Bakery;
```

Product_Code	Product_Name	Cost_Price	Selling_Price	Quantity
BA01	White Bread	20	25	17
BA02	Brown Bread	25	30	19
BA03	Fruit Cake	30	40	13
BA04	Burger Buns	30	35	15
BA05	Cookies	40	55	21
BA06	Biscuit	30	50	20

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Sales;
```

Product_Category	Product_Code	Product_Name	Date	Quantity	Selling_Price	Amount	Profit
Bakery	BA06	Biscuit	2021-02-10	3	50	150	60

1 row in set (0.00 sec)

11. Manage Order List:

a. Add Order:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:2

THINGS TO DO:

1. Add Products To Dairy
2. Update Products In Dairy
3. To Check Which Product Have Less Quantity In Dairy
4. To View All Products Of Dairy
5. To Sell Products Of Dairy
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:6

THINGS TO DO:

1. Add Order To Order List
2. To Update The Quantity Of A Order
3. To Delete A Order
4. To View Complete Order List

Enter Your Choice:1

Enter The Name Of The Product To Be Ordered:Butter

Enter The Quantity:5

Do You Want To Add More Products (Yes/No):No

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455
Dairy	DA02	Butter	5	90	450

2 rows in set (0.00 sec)

b. Update Order Quantity:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:2

THINGS TO DO:

1. Add Products To Dairy
2. Update Products In Dairy
3. To Check Which Product Have Less Quantity In Dairy
4. To View All Products Of Dairy
5. To Sell Products Of Dairy
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:6

THINGS TO DO:

1. Add Order To Order List
2. To Update The Quantity Of A Order
3. To Delete A Order
4. To View Complete Order List

Enter Your Choice:2

Enter The Name Of The Product Whose Quantity Has To Be Updated:Butter
Enter The Updated Quantity:6

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455
Dairy	DA02	Butter	5	90	450
Bakery	BA03	Fruit Cake	5	30	150
Beverages	BE04	Lemonade	3	10	30
Snacks	SN04	Noodles	8	25	200
Bathroom_And_Toiletries	BT02	Shampoo	3	210	630

6 rows in set (0.01 sec)

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455
Dairy	DA02	Butter	6	90	540
Bakery	BA03	Fruit Cake	5	30	150
Beverages	BE04	Lemonade	3	10	30
Snacks	SN04	Noodles	8	25	200
Bathroom_And_Toiletries	BT02	Shampoo	3	210	630

6 rows in set (0.00 sec)

c. Delete Order:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:4

THINGS TO DO:

1. Add Products To Beverages
2. Update Products In Beverages
3. To Check Which Product Have Less Quantity In Beverages
4. To View All Products Of Beverages
5. To Sell Products Of Beverages
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:6

THINGS TO DO:

1. Add Order To Order List
2. To Update The Quantity Of A Order
3. To Delete A Order
4. To View Complete Order List

Enter Your Choice:3

Enter The Name Of The Product Whose Order Is To Be Deleted:Lemonade

```
mysql> SELECT * FROM Orders;
```

Product_Category	Product_Code	Product_Name	Quantity	Cost_Price	Amount
Fruits_And_Vegetables	FV05	Apple	7	65	455
Dairy	DA02	Butter	6	90	540
Bakery	BA03	Fruit Cake	5	30	150
Snacks	SN04	Noodles	8	25	200
Bathroom_And_Toiletries	BT02	Shampoo	3	210	630

5 rows in set (0.00 sec)

d. View Complete Order List:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:5

THINGS TO DO:

1. Add Products To Fruits_And_Vegetables
2. Update Products In Fruits_And_Vegetables
3. To Check Which Product Have Less Quantity In Fruits_And_Vegetables
4. To View All Products Of Fruits_And_Vegetables
5. To Sell Products Of Fruits_And_Vegetables
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:6

THINGS TO DO:

1. Add Order To Order List
2. To Update The Quantity Of A Order
3. To Delete A Order
4. To View Complete Order List

Enter Your Choice:4

```
('Fruits_And_Vegetables', 'FV05', 'Apple', 7, 65, 455)
('Dairy', 'DA02', 'Butter', 6, 90, 540)
('Bakery', 'BA03', 'Fruit Cake', 5, 30, 150)
('Snacks', 'SN04', 'Noodles', 8, 25, 200)
('Bathroom_And_Toiletries', 'BT02', 'Shampoo', 3, 210, 630)
```

12. Manage Sales List:

a. Add Sales:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:6

THINGS TO DO:

1. Add Products To Bathroom_And_Toiletries
2. Update Products In Bathroom_And_Toiletries
3. To Check Which Product Have Less Quantity In Bathroom_And_Toiletries
4. To View All Products Of Bathroom_And_Toiletries
5. To Sell Products Of Bathroom_And_Toiletries
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:7

THINGS TO DO:

1. Add Record To Sales List
2. To Delete Sales Before A Particular Date
3. To View Complete Sales List
4. To View Total Profit Earned

Enter Your Choice:1

Enter The Name Of The Product To Be Sold:Soap

Enter The Quantity:2

Do You Want To Sell More Products (Yes/No):No

```
mysql> SELECT * FROM Sales;
```

Product_Category	Product_Code	Product_Name	Date	Quantity	Selling_Price	Amount	Profit
Bakery	BA06	Biscuit	2021-02-10	3	50	150	60
Bathroom_And_Toiletries	BT01	Soap	2021-02-11	2	60	120	30

2 rows in set (0.00 sec)

b. Delete Sales By Date:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:7

THINGS TO DO:

1. Add Products To Snacks
2. Update Products In Snacks
3. To Check Which Product Have Less Quantity In Snacks
4. To View All Products Of Snacks
5. To Sell Products Of Snacks
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:7

THINGS TO DO:

1. Add Record To Sales List
2. To Delete Sales Before A Particular Date
3. To View Complete Sales List
4. To View Total Profit Earned

Enter Your Choice:2

Enter The Year(YYYY):2021

Enter The Month(MM):02

Enter The Date(DD):11

Do You Want To Run The Program Again (Yes/No):No

```
mysql> SELECT * FROM Sales;
```

Product_Category	Product_Code	Product_Name	Date	Quantity	Selling_Price	Amount	Profit
Bakery	BA06	Biscuit	2021-02-10	3	50	150	60
Bathroom_And_Toiletries	BT01	Soap	2021-02-11	2	60	120	30
Snacks	SN03	Nachos	2021-02-11	4	40	160	20
Dairy	DA04	Yogurt	2021-02-11	1	40	40	5
Beverages	BE06	Water	2021-02-11	5	12	60	10
Fruits_And_Vegetables	FV03	Onion	2021-02-11	2	35	70	10

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM Sales;
```

Product_Category	Product_Code	Product_Name	Date	Quantity	Selling_Price	Amount	Profit
Bathroom_And_Toiletries	BT01	Soap	2021-02-11	2	60	120	30
Snacks	SN03	Nachos	2021-02-11	4	40	160	20
Dairy	DA04	Yogurt	2021-02-11	1	40	40	5
Beverages	BE06	Water	2021-02-11	5	12	60	10
Fruits_And_Vegetables	FV03	Onion	2021-02-11	2	35	70	10

5 rows in set (0.00 sec)

c. View Complete Sales List:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:1

THINGS TO DO:

1. Add Products To Bakery
2. Update Products In Bakery
3. To Check Which Product Have Less Quantity In Bakery
4. To View All Products Of Bakery
5. To Sell Products Of Bakery
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:7

THINGS TO DO:

1. Add Record To Sales List
2. To Delete Sales Before A Particular Date
3. To View Complete Sales List
4. To View Total Profit Earned

Enter Your Choice:3

```
('Bathroom_And_Toiletries', 'BT01', 'Soap', datetime.date(2021, 2, 11), 2, 60, 120, 30)
('Snacks', 'SN03', 'Nachos', datetime.date(2021, 2, 11), 4, 40, 160, 20)
('Dairy', 'DA04', 'Yogurt', datetime.date(2021, 2, 11), 1, 40, 40, 5)
('Beverages', 'BE06', 'Water', datetime.date(2021, 2, 11), 5, 12, 60, 10)
('Fruits_And_Vegetables', 'FV03', 'Onion', datetime.date(2021, 2, 11), 2, 35, 70, 10)
```

d. View Total Profit:

WHICH PRODUCT CATEGORY YOU WANT TO ACCESS:

1. Bakery
2. Dairy
3. Cereals_And_Spices
4. Beverages
5. Fruits_And_Vegetables
6. Bathroom_And_Toiletries
7. Snacks

Enter Your Choice:3

THINGS TO DO:

1. Add Products To Cereals_And_Spices
2. Update Products In Cereals_And_Spices
3. To Check Which Product Have Less Quantity In Cereals_And_Spices
4. To View All Products Of Cereals_And_Spices
5. To Sell Products Of Cereals_And_Spices
6. To Manage Order List
7. To Manage Sales List

Enter Your Choice:7

THINGS TO DO:

1. Add Record To Sales List
2. To Delete Sales Before A Particular Date
3. To View Complete Sales List
4. To View Total Profit Earned

Enter Your Choice:4

Total Profit Earned:

75

Do You Want To Run The Program Again (Yes/No):No

13. Ending:

Do You Want To Run The Program Again (Yes/No):No

~~~~~  
Thank You For Using Our GROCERY SHOP MANAGEMENT SYSTEM  
~~~~~

BY:

Mukund Sarda

Kartik Singh

(XII-H)

REQUIREMENTS

1. Hardware Requirements:

- a. Intel Core i3/i5/i7/i9 Or Any Equivalent
- b. At least 256 MB RAM
- c. 2 MB Free Space On Hard Disk

2. Software Requirements:

- a. MySQL 8.0
- b. Python 3.0 / Higher (Thonny / Python IDLE)
- c. Any Operating System