PROJECT:

GROCERY STORE

By – Kushagra Lakhwani

github.com/KorigamiK

INDEX

INTROUDUCTION

Grocery shoppers have high expectations: they want to experience welcoming surroundings, large product variety, value for money, sales efficiency and much more. The competition is intense and the margins small. That’s why how you manage your daily operations can make or break your business.

Every food retailer needs to constantly face the complexities of handling a large and extremely varied product mix. Out-of-stock products are the most common cause for lost sales in the grocery business. At the same time, grocery products are often perishables, and can have very different life-cycles, thus further complicating ordering and inventory management.

Grocery retailers are also expected to provide high-quality, speedy service. Fast and efficient check-out is critical; each check-out line needs to keep moving while it processes many items per minute, under pressure.

A Software like this is crucial for any everyday grocery store or a café.

ACKNOWLEDGEMENT

I aknowledge the greatness of python as a programming language.

SOURCE CODE

- ## Setup

```
import mysql.connector as sql

con = sql.connect(

    host="localhost",
    user=input("enter user (root)"),
    password=input("enter password"),
)
print("{: ^40}".format("Origami"))
cur = con.cursor()

cur.execute("CREATE DATABASE IF NOT EXISTS kushagra_project")
cur.execute("use kushagra_project")
d1 = ["Items", "Price"]
q = "CREATE TABLE {2}(Sno int, {0} Varchar(60), {1} int, Item_Code varchar(50));"
cur.execute(q.format(d1[0], d1[1], "Menu_1"))
cur.execute("alter table Menu_1 add primary key(Item_Code);")
con.commit()

dict_1 = {
    "Lacinato Kale": 30,
    "Baby Spinach": 34,
    "Cauliflower": 23,
    "Avocados": 24,
    "Apples": 43,
    "Strawberries": 12,
    "Organic Eggs": 30,
    "Kefir": 34,
    "Almond Milk": 23,
    "Yogurt": 24,
    "Sheep's Milk": 43,
}


def rq(x):
    return '"{}"'.format(x)


for i in range(len(dict_1)):
    cur.execute(
        "INSERT into Menu_1 Values({0}, {1}, {2}, ".format(
            i, rq(list(dict_1.keys())[i]), dict_1[list(dict_1.keys())[i]]
        )
        + rq("#GS{:03d}".format(i))
        + ");"
    )
con.commit()

d2 = [
    "SNACKS",
    "Price",
]
dict_2 = {"Popcorn": 30, "Hummus": 34, "Dark Chocolate": 23, "Dried Fruit": 24}

q = "CREATE TABLE {2}(Sno int, {0} Varchar(60), {1} int, Item_Code varchar(50));"
cur.execute(q.format(d2[0], d2[1], "Menu_2"))
cur.execute("alter table Menu_2 add primary key(Item_Code);")

for i in range(len(dict_2)):
    cur.execute(
        "INSERT into Menu_2 Values({0}, {1}, {2}, ".format(
            i, rq(list(dict_2.keys())[i]), dict_2[list(dict_2.keys())[i]]
        )
        + rq("#GS{:03d}".format(i + 11))
        + ");"
    )
con.commit()

quer1 = """CREATE TABLE cust_details(
CustID Varchar(30),
Name varchar(20),
Address varchar (40),
EmailID varchar(35),
pincode int,
PhoneNO varchar(40),
pass varchar(50)
);"""
cur.execute(quer1)
cur.execute("alter table cust_details add primary key(CustID);")
cur.execute(
    'insert into cust_details values("ADMIN", "origami", "asdf", "admin@asdf.com"
, 1234, "100", "origami");'
)
cur.execute(
    "create table gifts(Sno int, CustID varchar(60), Voucher_type varchar(60), Value int);"
)
cur.execute("alter table gifts add primary key(CustID);")
con.commit()
```

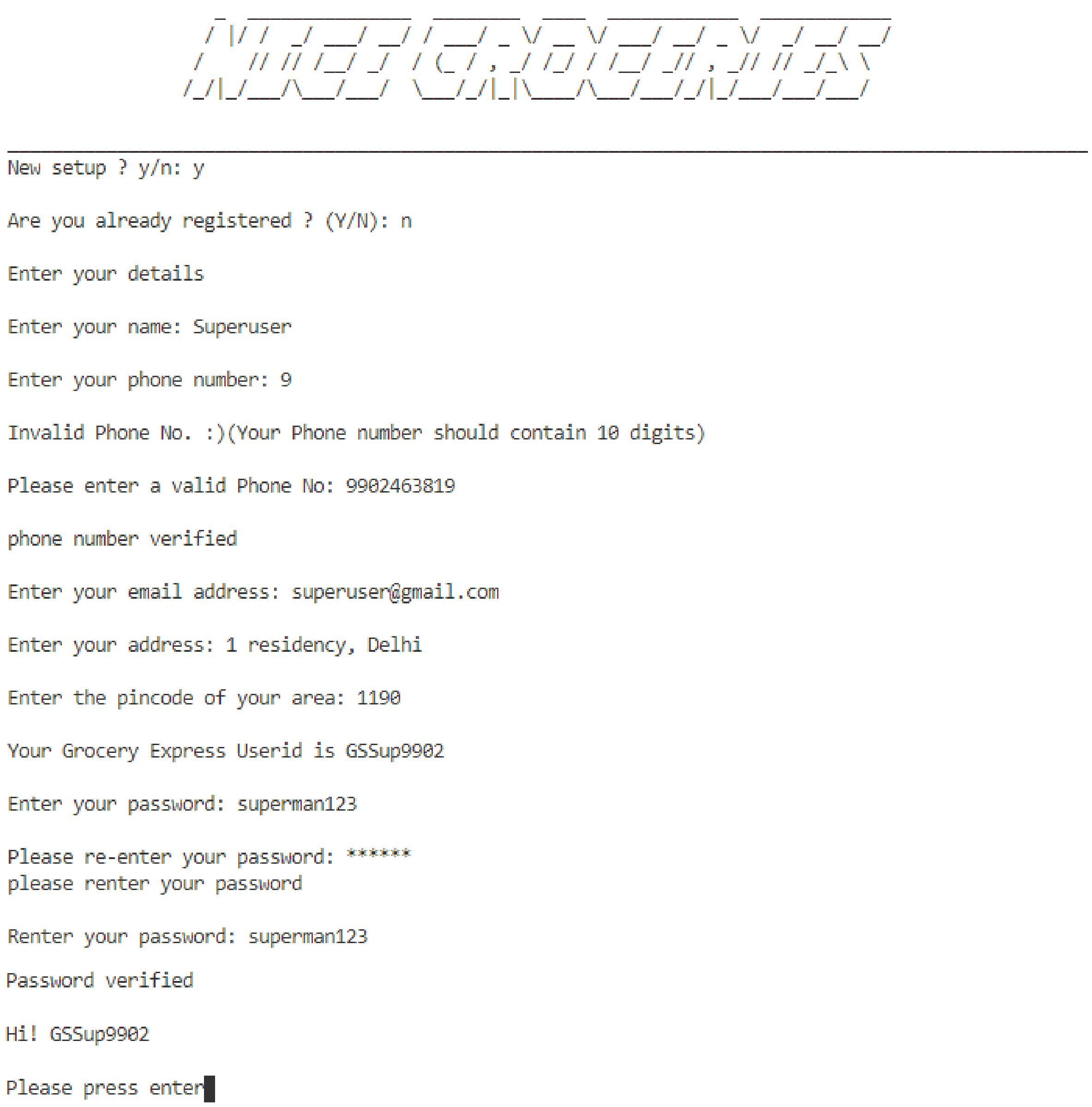
▪ Code

```
import mysql.connector as sq
from prettytable import PrettyTable as pt
import math
import time

name_of_store = """
\t\t   _  _______________  ________  ____  ____________  ____________
\t\t  / |/ /  _/ ___/ __/ / ___/ _ \/ __ \/ ___/ __/ _ \/  _/ __/ __/
\t\t /    // // /__/ _/  / (_ / , _/ /_/ / /__/ _// , _// // _/_\ \
\t\t/_/|_/___/\___/___/  \___/_/|_|\____/\___/___/_/|_/___/___/___/
"""
print(name_of_store)
print(f"{'':_^90s}")
# exit()
e = input("New setup ? y/n: ")

print()

if e != "y":
    con = sq.connect(host="localhost", user="root", database="project")
else:
    con = sq.connect(
        host="localhost",
        # user=input("Enter user (root): "),
        # password=input("Enter password: "),
        user="root",
        password="origami",
        database="kushagra_project",
    )

cur = con.cursor()


def table_print(y):
    cur.execute("Select * from {}".format(y))
    a = cur.fetchall()
    x = pt()
    cur.execute("desc {}".format(y))
    f = cur.fetchall()
    d = []
    for i in f:
        d += [list(i)[0]]

    g = [[] for i in range(len(a[0]))]
    for i in list(a):
        for j, k in enumerate(list(i)):
            g[j].append(k)
    for i, j in enumerate(g):
        x.add_column(d[i], j)
    x.title = y
    return x


# cur.execute("select * from Menu_1;")
# a=cur.fetchall()
# print(a)

ask = input("Are you already registered ? (Y/N): ")
print()
a = 1
b = 1
try:
    from msvcrt import getch

    def getpass(prompt):
        """Replacement for getpass.getpass() which prints asterisks for each character typed"""
        print(prompt, end="", flush=True)
        buf = b""
        while True:
            ch = getch()
            if ch in {b"\n", b"\r", b"\r\n"}:
                print("")
                break
            elif ch == b"\x08":  # Backspace
                buf = buf[:-1]
                print(
                    f'\r{(len(prompt)+len(buf)+1)*" "}\r{prompt}{"*" * len(buf)}',
                    end="",
                    flush=True,
                )
            elif ch == b"\x03":  # Ctrl+C
                raise KeyboardInterrupt
            else:
                buf += ch
                print("*", end="", flush=True)
        return buf.decode(encoding="utf-8")


except ImportError:
    from getpass import getpass
while b == 1:
    if ask == "Y" or ask == "y":
        print("Please give your login details\n")
        # verifying the userid
        while a == 1:
            enterid = str(input("Enter your userid: "))
            print()
            tup = (enterid,)
            cur.execute("Select CustID from cust_details")
            varc = cur.fetchall()
            # varc is a list of all CustID stored in Cust_details in tuple databases
            for c in varc:
                if c == tup:
                    print("UserID found\n")
                    a = 2
                    break
            else:
                print("No userid found\n")

        psswd = getpass("Enter you password: ")
        print()
        check = (enterid, psswd)
        cur.execute("Select CustID, pass from cust_details")
        varp = cur.fetchall()
        a = 1
        h = 1
        while a == 1:
            for j in varp:
                if j == check:
                    print("\nPassword verified\n")
                    a = 2
                    break
            else:
                print("Incorrect Password\n")
                psswd = getpass("Please type the correct password: ")
                check = (enterid, psswd)
                h += 1
                if h > 2:
                    print(
                        "You entered the wrong password too many times, create a
new id\n"
                    )
                    ask = "n"
        b = 2

    elif ask == "N" or ask == "n":
        print("Enter your details\n")
        name = input("Enter your name: ")
        print()
        phoneno = str(input("Enter your phone number: "))
        print()

        while True:
            if len(phoneno) == 10:
                print("phone number verified\n")
                break
            else:
                print(
                    "Invalid Phone No. :)(Your Phone number should contain 10 dig
its)\n"
                )
                phoneno = str(input("Please enter a valid Phone No: "))
                print()
        email = input("Enter your email address: ")
        print()
        add = input("Enter your address: ")
        print()
        while True:
            pincode = input("Enter the pincode of your area: ")
            try:
                int(pincode)
                break
            except:
                print("Enter correct pincode")
        print()
        time.sleep(2)
        user_id = "GS" + name[0:3] + phoneno[0:4]
        print("Your Grocery Express Userid is", user_id + "\n")
        psswd = input("Enter your password: ")
        print()
        recheck = getpass("Please re-enter your password: ")

        while True:
            if psswd == recheck:
                time.sleep(2)
                print("Password verified\n")
                break
            else:
                print("please renter your password\n")
                recheck = input("Renter your password: ")
                print()
        # Entering the data given by user into the Database
        query1 = "Insert into cust_details values(%s, %s, %s, %s, %s, %s, %s)"
        cur.execute(query1, (user_id, name, add, email, pincode, phoneno, psswd))
        con.commit()  # veeeery important
        b = 2
    else:
        ask = input("Are you already registered ? (Y or N): ")
        print()
try:
    cur.execute(
        'select name, CustID from cust_details where CustID="{}"'.format(enterid)
    )
    z = cur.fetchall()
    name = z[0][0]
    user_id = z[0][1]
except:
    name = user_id


def get_voucher(id_of_user):
    cur.execute(f'select Value from gifts where CustID="{user_id}";')
    data = cur.fetchall()
    if len(data) != 0:
        return data[0][0]
    else:
        return 0


print("Hi! {}".format(name), "\n")
time.sleep(1)
input("Please press enter")
print()
print(table_print("Menu_1"))
print()
print(table_print("Menu_2"))
time.sleep(1)
cur.execute("select Items, Price from Menu_1;")
```

```
dict_1 = dict(cur.fetchall())
cur.execute("select SNACKS, Price from Menu_2;")
dict_2 = dict(cur.fetchall())

list_1 = []
ans = "y"


def checker(x):
    z = list(x.capitalize())
    w = ""
    for i in range(len(z)):
        try:
            if i == x.index(" "):
                z[i + 1] = z[i + 1].upper()
        except:
            continue
    for j in range(len(z)):
        w += z[j]
    x = w
    if x in dict_1:
        list_1.append(x)
    elif x in dict_2:
        list_1.append(x)
    else:
        print(
            "Sorry sir, We do not have ",
            x,
            " in stock yet, Leave your requests in the feedback section.\n",
        )


while True:
    while ans == "y":
        o = input("\nWhat do you want to purchase, Sir? :  ")
        print()
        checker(o)
        ans = input("Want something else, Sir(Y/N): ")
        print()
    else:
        yes = input("Do you want to continue shopping ? (y/n): ")
        if yes != "y":
            break
        else:
            ans = yes
            pass
print()

# print(list_1) #good for debugging


def quant(y):
    for i in range(len(y)):
        a = int(input("Quantity of item " + y[i] + ": "))
        print()
        list_2[i] = a


def feed():
    input("Please tell us about your visit, suggestions, complaints or requests:
\n")
    print("Thank you for you cooperation, we will get back to you as soon as poss
ible ")


def truncate(number, digits) -> float:
    stepper = 10.0 ** digits
    return math.trunc(stepper * number) / stepper


if len(list_1) != 0:

    list_2 = list(list_1)
    quant(list_1)

    list_3 = [[[] for b in range(6)] for a in range(len(list_1) + 1)]

    list_3[0][0] = "S. NO."
    list_3[0][1] = "NAME OF ITEMS (Price Per Item)"
    list_3[0][2] = "QUANTITY"
    list_3[0][3] = "TAX"
    list_3[0][4] = "PRICE"
    list_3[0][5] = "EFFECTIVE PRICE"

    sum = 0
    z = 18
    for n in range(1, len(list_1) + 1):

        list_3[n][0] = str(n)
        list_3[n][1] = str(list_1[n - 1])
        list_3[n][2] = list_2[n - 1]
        list_3[n][3] = str(z) + "%"
        try:
            str(list_1[n - 1]) in dict_1
            list_3[n][4] = list_2[n - 1] * dict_1[str(list_1[n - 1])]
            sum += list_2[n - 1] * dict_1[str(list_1[n - 1])] * (1 + (18 / 100))
            list_3[n][5] = (
                "Rs."
                + str(list_2[n - 1] * dict_1[str(list_1[n - 1])] * (1 + (18 / 100
)))[
                    :13
                ]
            )
        except:
            list_3[n][4] = list_2[n - 1] * dict_2[str(list_1[n - 1])]
            sum += list_2[n - 1] * dict_2[str(list_1[n - 1])] * (1 + (18 / 100))
            list_3[n][5] = (
                "Rs."
                + str(list_2[n - 1] * dict_2[str(list_1[n - 1])] * (1 + (18 / 100
)))[
                    :13
                ]
            )

    y = pt()

    b = []

    for i in range(len(list_3[0])):
        b.append(list_3[0][i])
    y.field_names = b

    for i in range(1, len(list_3)):
        try:
            list_3[i][1] += " ({})".format(str(dict_1[list_1[i - 1]]))
        except:
            list_3[i][1] += " ({})".format(str(dict_2[list_1[i - 1]]))

    for k in range(1, len(list_3)):
        y.add_row(list_3[k])

    print(y)
    print()
    voucher = get_voucher(user_id)
    if voucher != 0:
        print(f"Applying your voucher of Rs.{voucher} \n")
        time.sleep(2)
        final_amount = round(sum) - voucher
    else:
        print("FREE TIP! Shop for Rs. 500 or more to get special discounts\n")
        time.sleep(1)
        final_amount = round(sum)
        pass
    print("Your total bill is Rs.", final_amount)
    time.sleep(1)
    print()
    print("Collect your recipt at the counter.\n")
    print("Thank you for the visit", name + "\n")
    print("Please Enter feedback and you can cllect your bill \n ")
    voucher_maybe = (
        f"Congratulations!! You got discount of Rs.{voucher} on you bill of Rs.{r
ound(sum)}\n"
        if voucher != 0
        else ""
    )
    with open("{}_bill.txt".format(name), "a") as f:
        f.writelines(
            [
                "{:_^88s}\n\n".format("Thank you for shopping with us !"),
                str(y) + "\n",
                voucher_maybe,
                "The total payable amount is: "
                + f"Rs.{str(final_amount)}/- "
                + "\n\n\n",
            ]
        )
    time.sleep(2)
    feed()

    def get_random():
        from random import randint

        return randint(10, 90)

    if final_amount >= 500:
        print("\nYou are elegible for our gift voucher. Shop again to claim it.")
        cur.execute("select CustID, Value from gifts;")
        for i in cur.fetchall():
            if user_id in i:
                cur.execute(
                    f'update gifts set Value={get_random()} where CustID="{user_i
d}";'
                )
                con.commit()
                break
        else:
            cur.execute(
                f'insert into gifts values(1,"{user_id}", "off rupees", {get_rand
om()});'
            )
            con.commit()
    con.close()
print("\nLookin' forward to you next visit !")


def stopper():
    if not input("Press enter to exit.."):
        print()
        exit()
    else:
        stopper()


stopper()
```

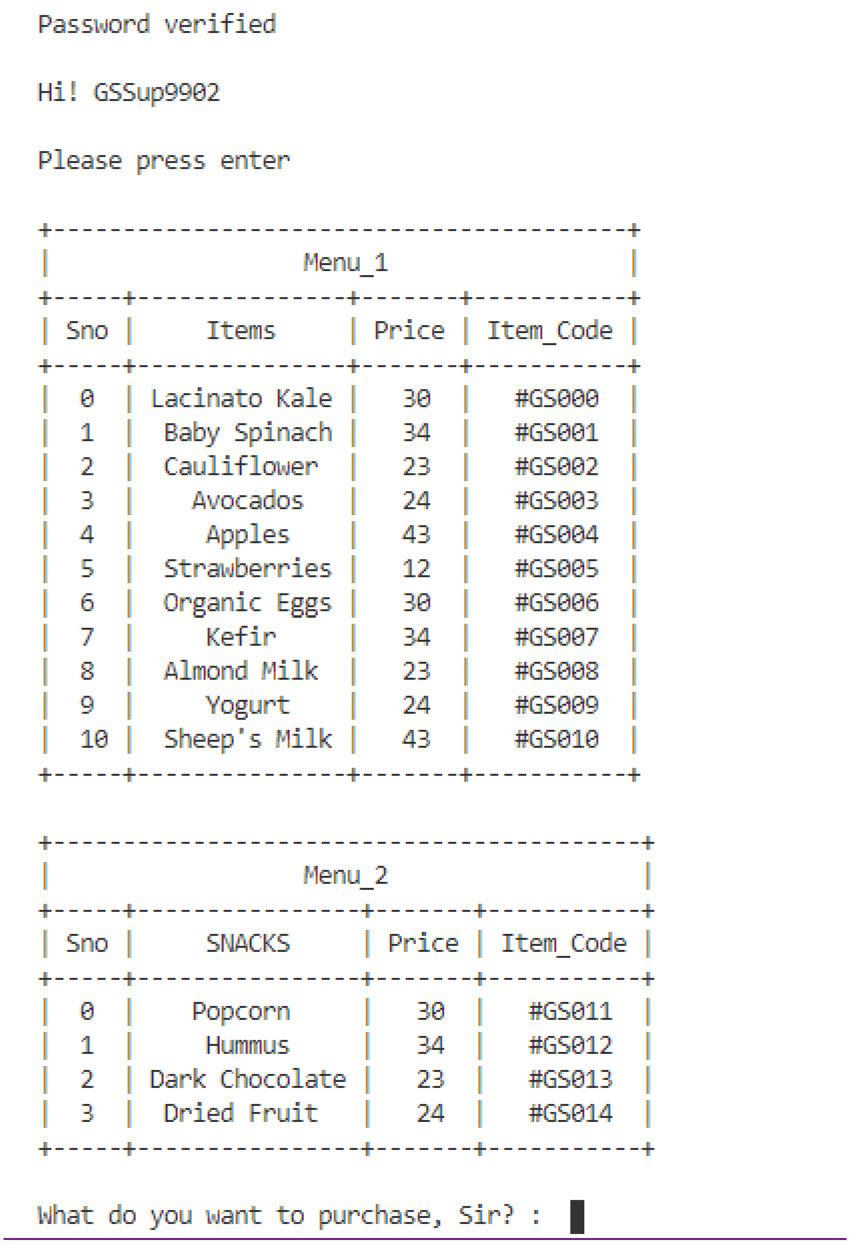
USAGE EXAMPLES

- ## Creating UserID for new customers:



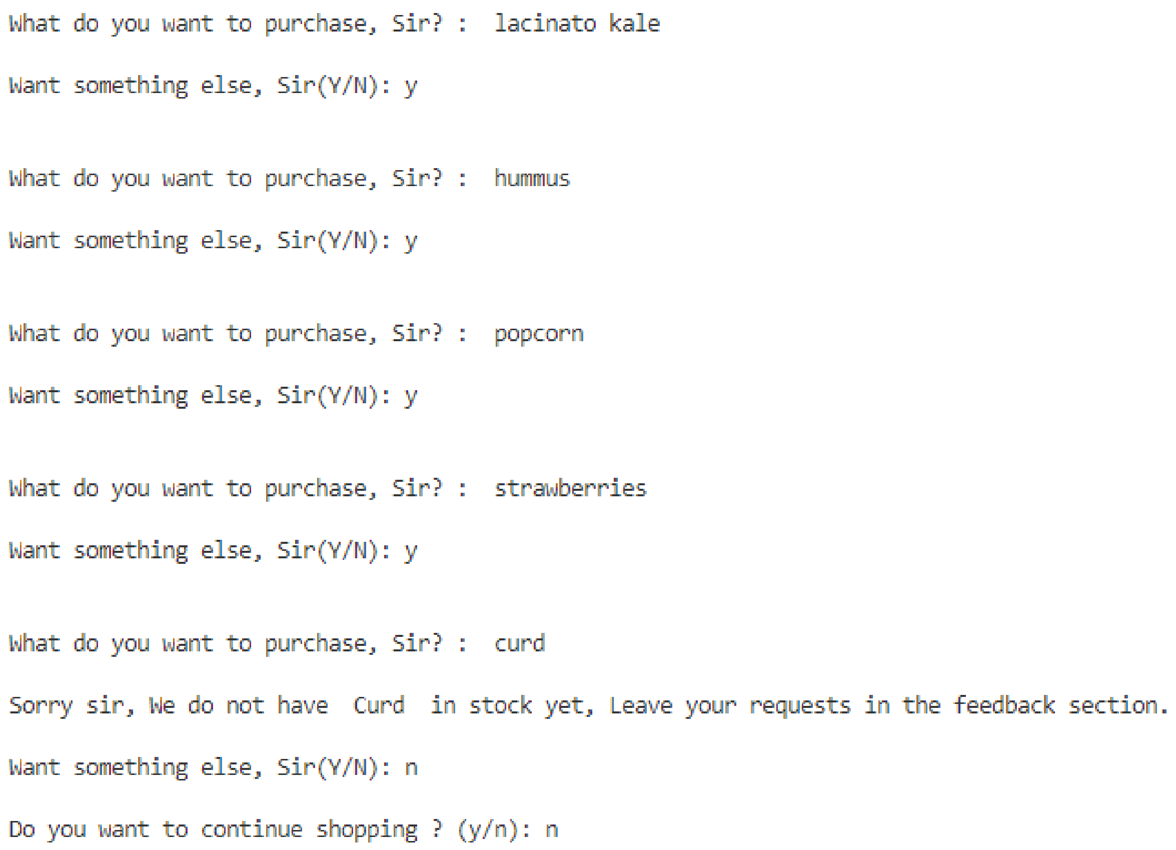
```
   _  ___________  _______  ____  _________  ______________
  / |/ /  _/ ___/ __/ / ___/ _ \/ __ \/ ___/ __/ _ \/  _/ __/ __/
 /    // // /__/ _/  / (_ / , _/ /_/ / /__/ _// , _// // _/_\ \
/_/|_/___/\___/___/  \___/_/|_|\____/\___/___/_/|_/___/___/___/

________________________________________________________________________________
New setup ? y/n: y

Are you already registered ? (Y/N): n

Enter your details

Enter your name: Superuser

Enter your phone number: 9

Invalid Phone No. :)(Your Phone number should contain 10 digits)

Please enter a valid Phone No: 9902463819

phone number verified

Enter your email address: superuser@gmail.com

Enter your address: 1 residency, Delhi

Enter the pincode of your area: 1190

Your Grocery Express Userid is GSSup9902

Enter your password: superman123

Please re-enter your password: ******
please renter your password

Renter your password: superman123

Password verified

Hi! GSSup9902

Please press enter
```

Showing the Menus:



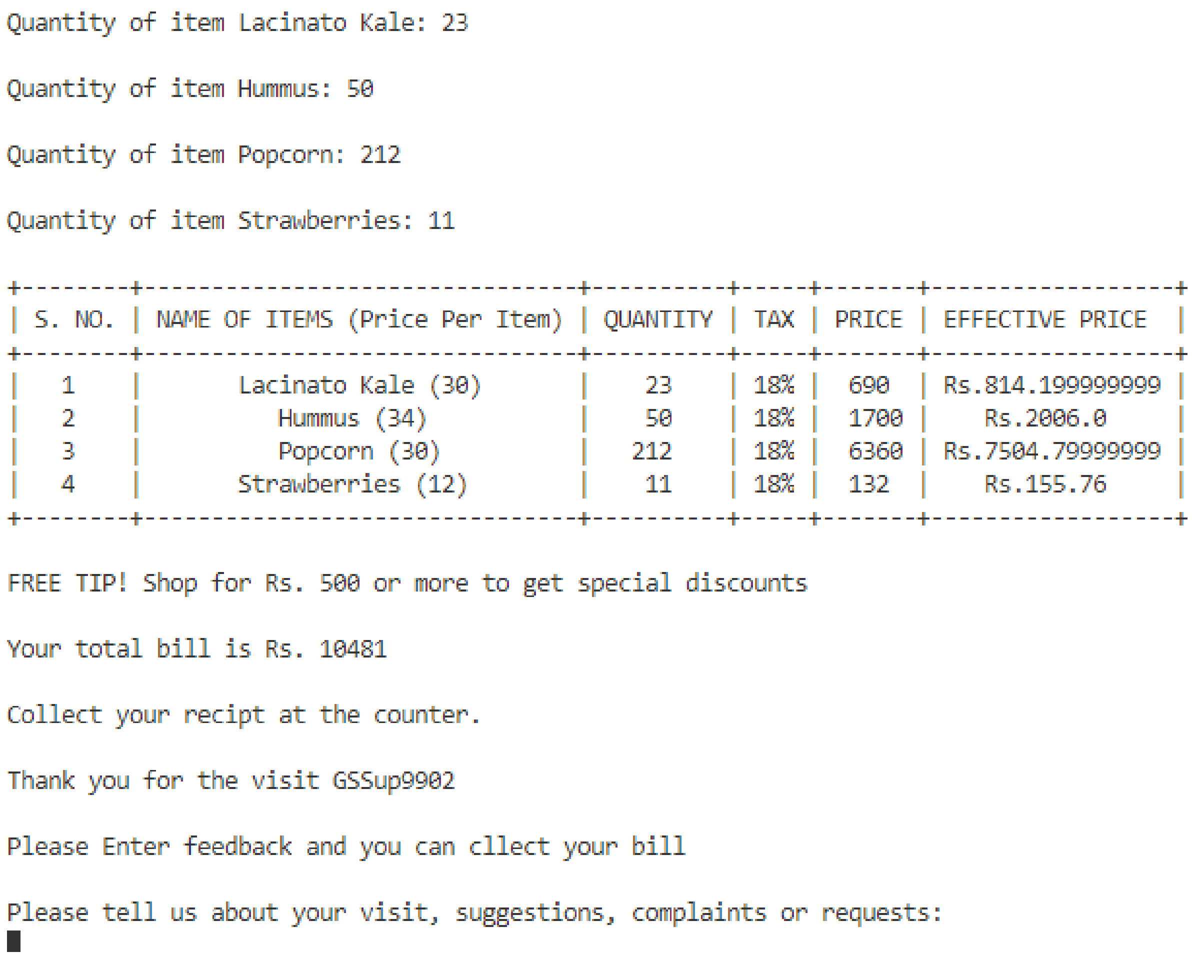
```
Password verified

Hi! GSSup9902

Please press enter

+-----------------------------------------+
|                  Menu_1                 |
+-----+---------------+-------+-----------+
| Sno |     Items     | Price | Item_Code |
+-----+---------------+-------+-----------+
|  0  | Lacinato Kale |   30  |   #GS000  |
|  1  |  Baby Spinach |   34  |   #GS001  |
|  2  |  Cauliflower  |   23  |   #GS002  |
|  3  |    Avocados   |   24  |   #GS003  |
|  4  |     Apples    |   43  |   #GS004  |
|  5  |  Strawberries |   12  |   #GS005  |
|  6  |  Organic Eggs |   30  |   #GS006  |
|  7  |     Kefir     |   34  |   #GS007  |
|  8  |  Almond Milk  |   23  |   #GS008  |
|  9  |     Yogurt    |   24  |   #GS009  |
|  10 |  Sheep's Milk |   43  |   #GS010  |
+-----+---------------+-------+-----------+

+------------------------------------------+
|                  Menu_2                  |
+-----+----------------+-------+-----------+
| Sno |     SNACKS     | Price | Item_Code |
+-----+----------------+-------+-----------+
|  0  |    Popcorn     |   30  |   #GS011  |
|  1  |     Hummus     |   34  |   #GS012  |
|  2  | Dark Chocolate |   23  |   #GS013  |
|  3  |  Dried Fruit   |   24  |   #GS014  |
+-----+----------------+-------+-----------+

What do you want to purchase, Sir? :
```

- ## Selecting Items:



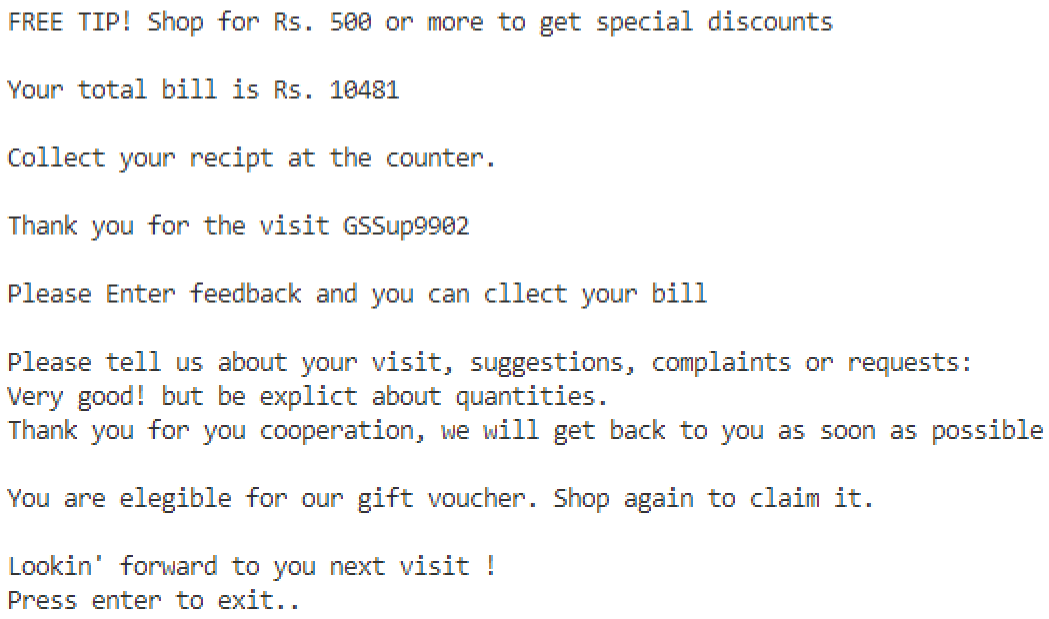
```
What do you want to purchase, Sir? :  lacinato kale

Want something else, Sir(Y/N): y

What do you want to purchase, Sir? :  hummus

Want something else, Sir(Y/N): y

What do you want to purchase, Sir? :  popcorn

Want something else, Sir(Y/N): y

What do you want to purchase, Sir? :  strawberries

Want something else, Sir(Y/N): y

What do you want to purchase, Sir? :  curd

Sorry sir, We do not have  Curd  in stock yet, Leave your requests in the feedback section.

Want something else, Sir(Y/N): n

Do you want to continue shopping ? (y/n): n
```

▪ Proceding to checkout:



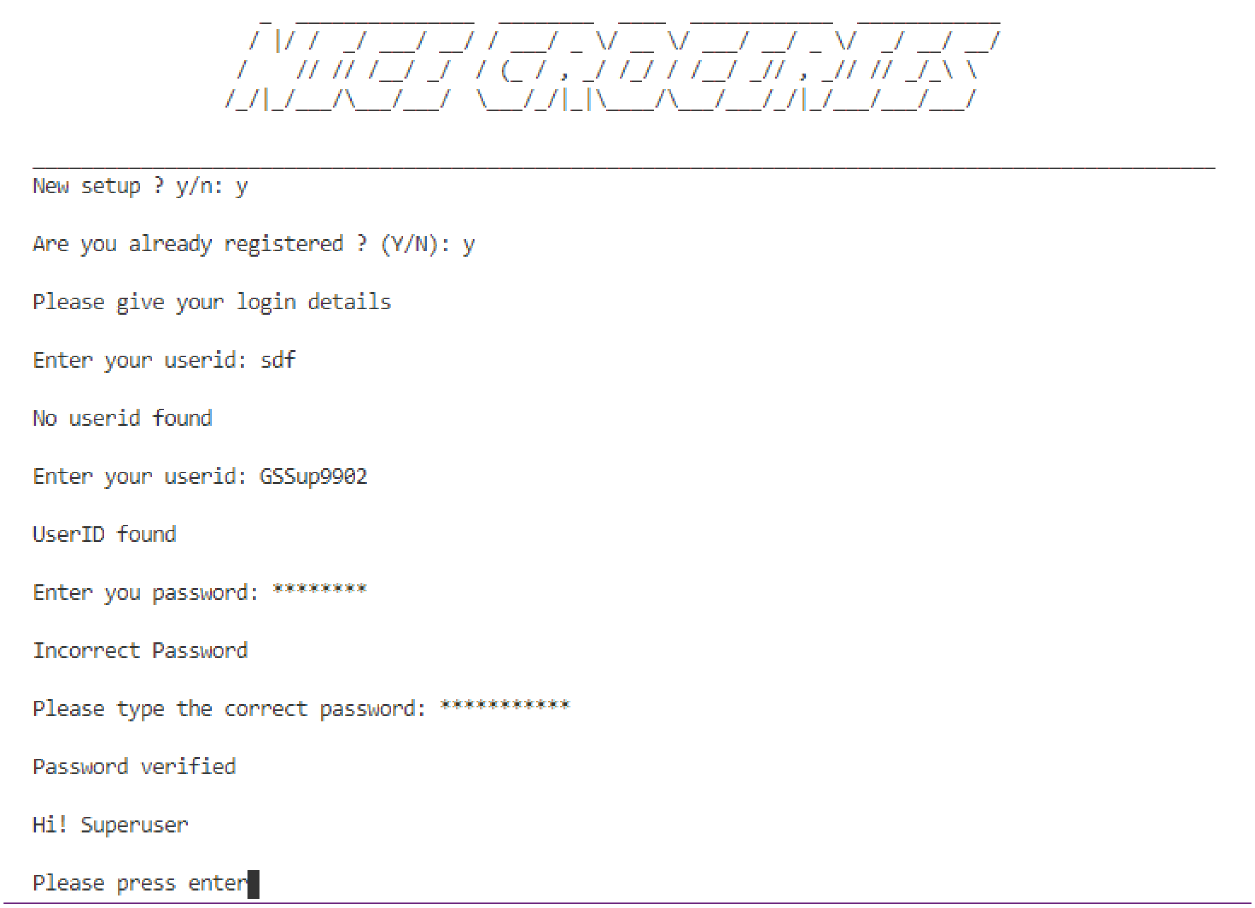
```
Quantity of item Lacinato Kale: 23

Quantity of item Hummus: 50

Quantity of item Popcorn: 212

Quantity of item Strawberries: 11

+--------+--------------------------------+----------+-----+-------+------------------+
| S. NO. | NAME OF ITEMS (Price Per Item) | QUANTITY | TAX | PRICE | EFFECTIVE PRICE  |
+--------+--------------------------------+----------+-----+-------+------------------+
|   1    |       Lacinato Kale (30)       |    23    | 18% |  690  | Rs.814.199999999 |
|   2    |          Hummus (34)           |    50    | 18% | 1700  |    Rs.2006.0     |
|   3    |          Popcorn (30)          |   212    | 18% | 6360  | Rs.7504.79999999 |
|   4    |       Strawberries (12)        |    11    | 18% |  132  |    Rs.155.76     |
+--------+--------------------------------+----------+-----+-------+------------------+

FREE TIP! Shop for Rs. 500 or more to get special discounts

Your total bill is Rs. 10481

Collect your recipt at the counter.

Thank you for the visit GSSup9902

Please Enter feedback and you can cllect your bill

Please tell us about your visit, suggestions, complaints or requests:
█
```

- Getting feedback and giving vouchers:



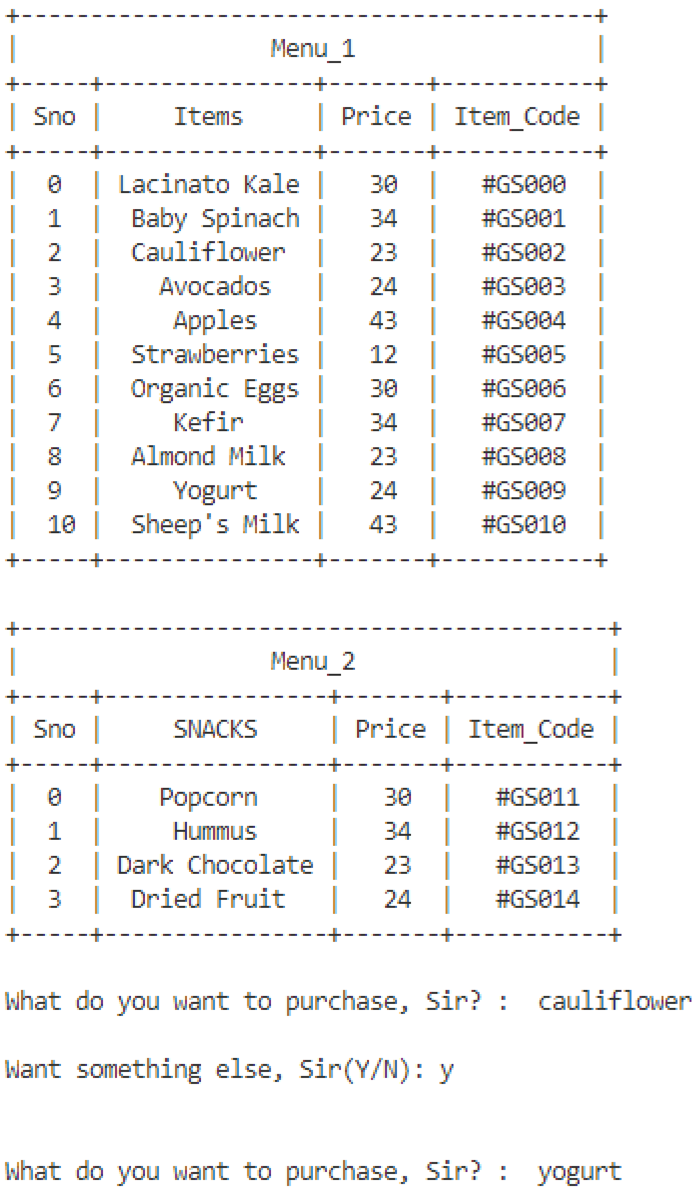
```
FREE TIP! Shop for Rs. 500 or more to get special discounts

Your total bill is Rs. 10481

Collect your recipt at the counter.

Thank you for the visit GSSup9902

Please Enter feedback and you can cllect your bill

Please tell us about your visit, suggestions, complaints or requests:
Very good! but be explict about quantities.
Thank you for you cooperation, we will get back to you as soon as possible

You are elegible for our gift voucher. Shop again to claim it.

Lookin' forward to you next visit !
Press enter to exit..
```

- Returning customers:

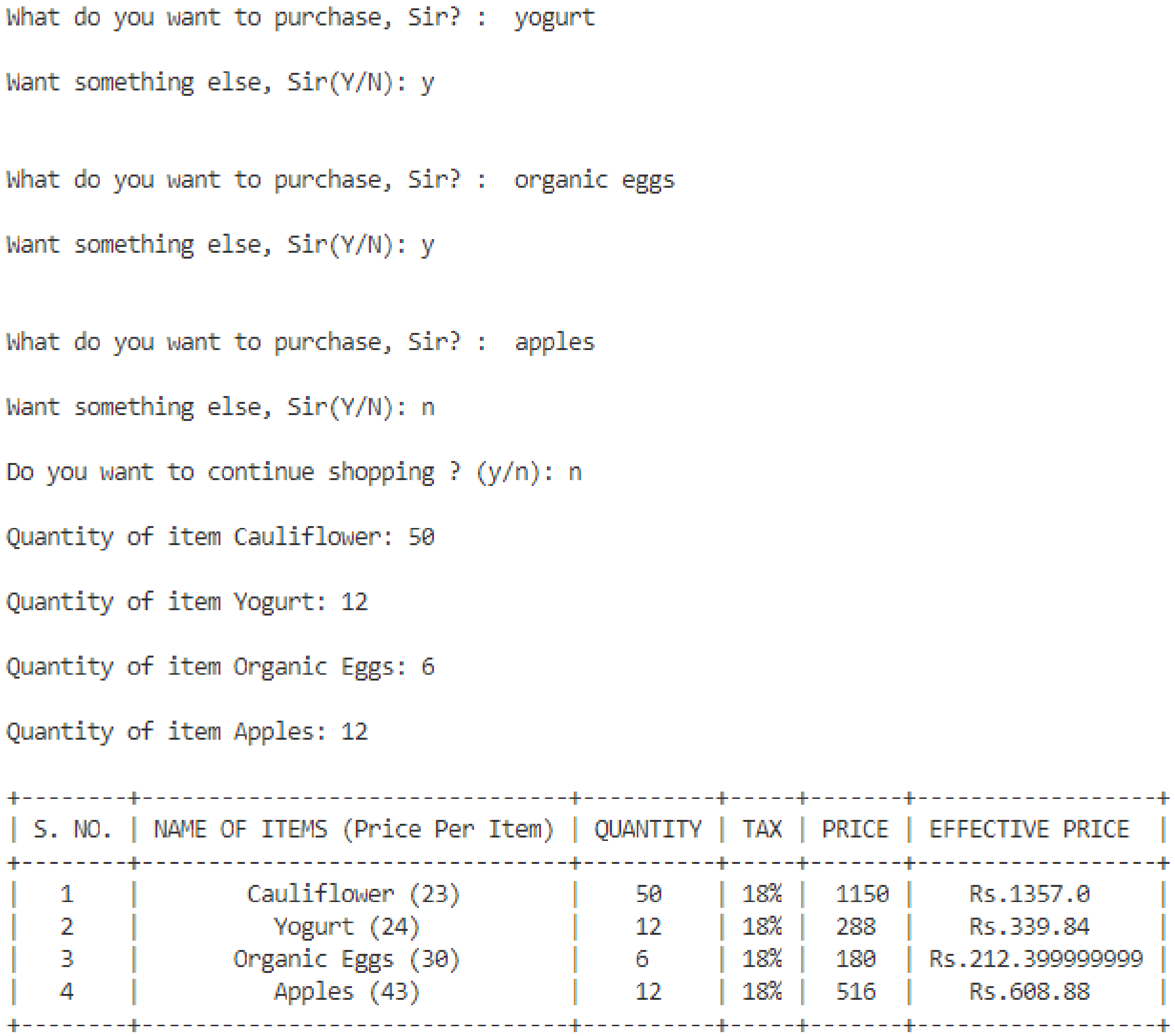


```
   _  ______________  _______  ____  __________  __________
  / |/ /  _/ ___/ __/ / ___/ _ \/ __ \/ ___/ __/ _ \/  _/ __/ __/
 /    // // /__/ _/  / (_ / , _/ /_/ / /__/ _// , _// // _/_\ \
/_/|_/___/\___/___/  \___/_/|_|\____/\___/___/_/|_/___/___/___/

New setup ? y/n: y

Are you already registered ? (Y/N): y

Please give your login details

Enter your userid: sdf

No userid found

Enter your userid: GSSup9902

UserID found

Enter you password: ********

Incorrect Password

Please type the correct password: ***********

Password verified

Hi! Superuser

Please press enter
```

- ## Buying items:

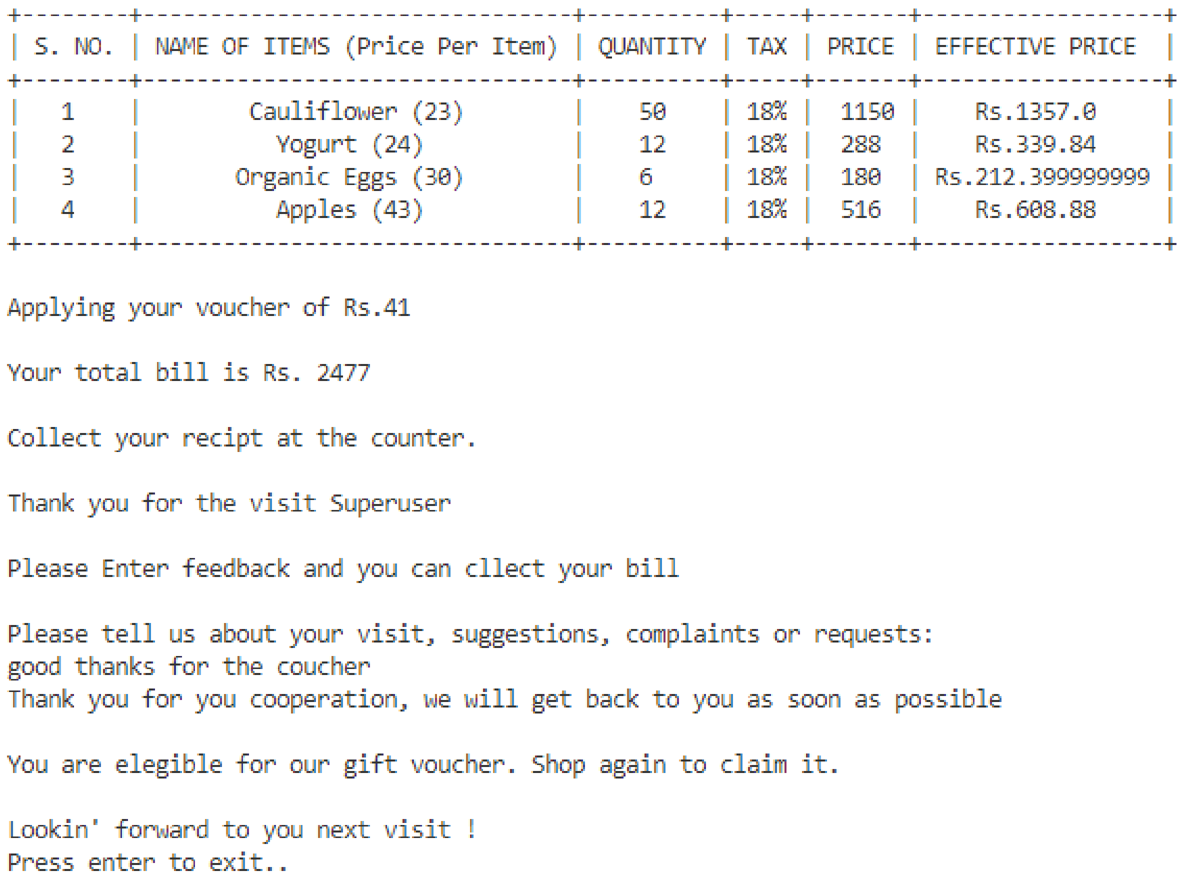


```
+-----------------------------------------+
|                  Menu_1                 |
+-----+---------------+-------+-----------+
| Sno |     Items     | Price | Item_Code |
+-----+---------------+-------+-----------+
|  0  | Lacinato Kale |   30  |   #GS000  |
|  1  |  Baby Spinach |   34  |   #GS001  |
|  2  |  Cauliflower  |   23  |   #GS002  |
|  3  |    Avocados   |   24  |   #GS003  |
|  4  |     Apples    |   43  |   #GS004  |
|  5  |  Strawberries |   12  |   #GS005  |
|  6  |  Organic Eggs |   30  |   #GS006  |
|  7  |     Kefir     |   34  |   #GS007  |
|  8  |  Almond Milk  |   23  |   #GS008  |
|  9  |     Yogurt    |   24  |   #GS009  |
|  10 |  Sheep's Milk |   43  |   #GS010  |
+-----+---------------+-------+-----------+

+------------------------------------------+
|                  Menu_2                  |
+-----+----------------+-------+-----------+
| Sno |     SNACKS     | Price | Item_Code |
+-----+----------------+-------+-----------+
|  0  |    Popcorn     |   30  |   #GS011  |
|  1  |     Hummus     |   34  |   #GS012  |
|  2  | Dark Chocolate |   23  |   #GS013  |
|  3  |  Dried Fruit   |   24  |   #GS014  |
+-----+----------------+-------+-----------+

What do you want to purchase, Sir? :  cauliflower

Want something else, Sir(Y/N): y


What do you want to purchase, Sir? :  yogurt
```



```
What do you want to purchase, Sir? :  yogurt

Want something else, Sir(Y/N): y


What do you want to purchase, Sir? :  organic eggs

Want something else, Sir(Y/N): y


What do you want to purchase, Sir? :  apples

Want something else, Sir(Y/N): n

Do you want to continue shopping ? (y/n): n

Quantity of item Cauliflower: 50

Quantity of item Yogurt: 12

Quantity of item Organic Eggs: 6

Quantity of item Apples: 12

+--------+------------------------------+----------+-----+-------+------------------+
| S. NO. | NAME OF ITEMS (Price Per Item) | QUANTITY | TAX | PRICE | EFFECTIVE PRICE  |
+--------+------------------------------+----------+-----+-------+------------------+
|   1    |       Cauliflower (23)       |    50    | 18% |  1150 |    Rs.1357.0     |
|   2    |         Yogurt (24)          |    12    | 18% |  288  |    Rs.339.84     |
|   3    |      Organic Eggs (30)       |    6     | 18% |  180  | Rs.212.399999999 |
|   4    |         Apples (43)          |    12    | 18% |  516  |    Rs.608.88     |
+--------+------------------------------+----------+-----+-------+------------------+
```

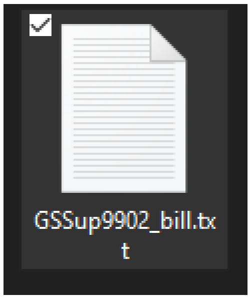
- Applying Vouchers from previous visits:

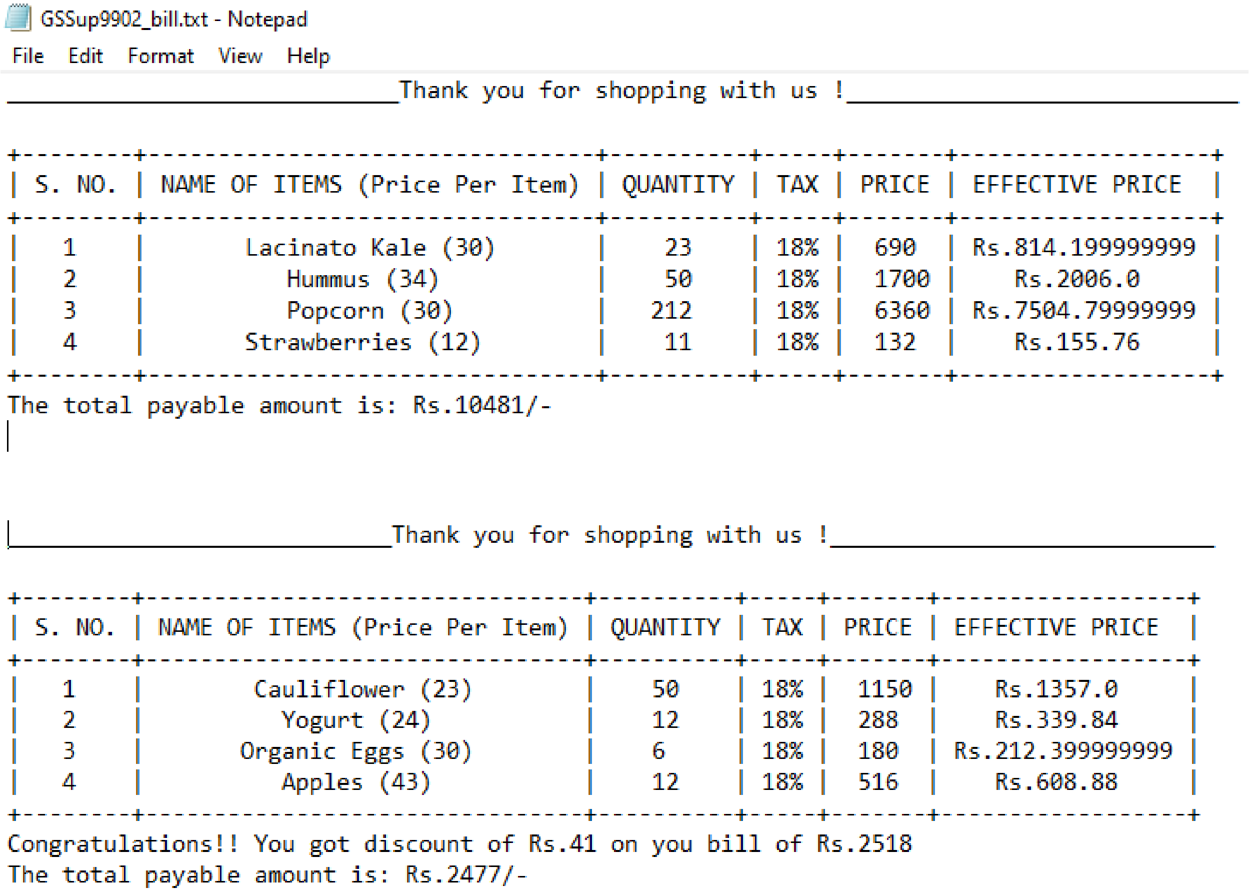


```
+--------+--------------------------------+----------+-----+-------+------------------+
| S. NO. | NAME OF ITEMS (Price Per Item) | QUANTITY | TAX | PRICE | EFFECTIVE PRICE  |
+--------+--------------------------------+----------+-----+-------+------------------+
|   1    |        Cauliflower (23)        |    50    | 18% |  1150 |    Rs.1357.0     |
|   2    |          Yogurt (24)           |    12    | 18% |  288  |    Rs.339.84     |
|   3    |       Organic Eggs (30)        |    6     | 18% |  180  | Rs.212.399999999 |
|   4    |          Apples (43)           |    12    | 18% |  516  |    Rs.608.88     |
+--------+--------------------------------+----------+-----+-------+------------------+

Applying your voucher of Rs.41

Your total bill is Rs. 2477

Collect your recipt at the counter.

Thank you for the visit Superuser

Please Enter feedback and you can cllect your bill

Please tell us about your visit, suggestions, complaints or requests:
good thanks for the coucher
Thank you for you cooperation, we will get back to you as soon as possible

You are elegible for our gift voucher. Shop again to claim it.

Lookin' forward to you next visit !
Press enter to exit..
```

Here a voucher is given if the purchase if greater than or equal to Rs. 500/-

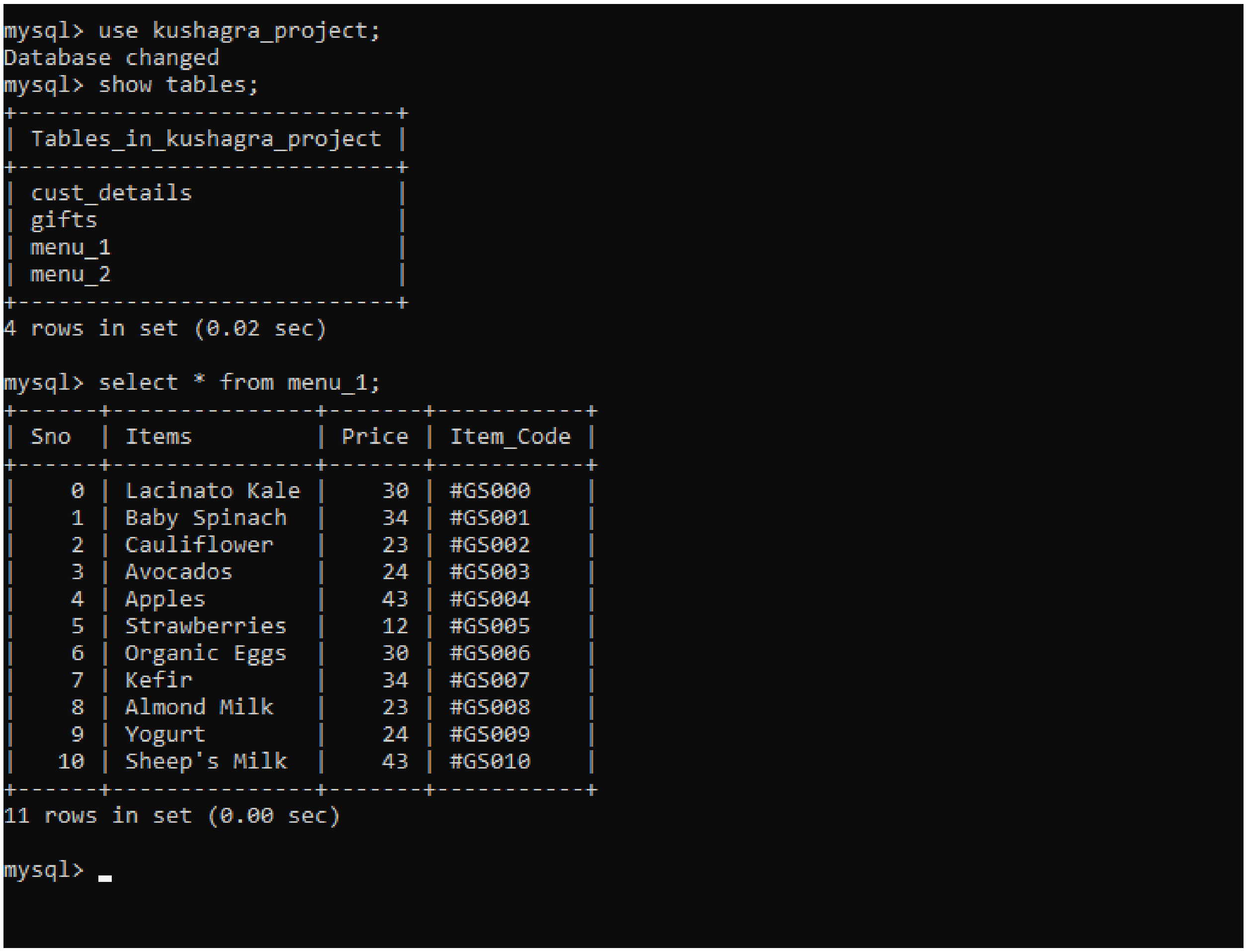
- Seeing the purchase history and collecting the bills of visits:



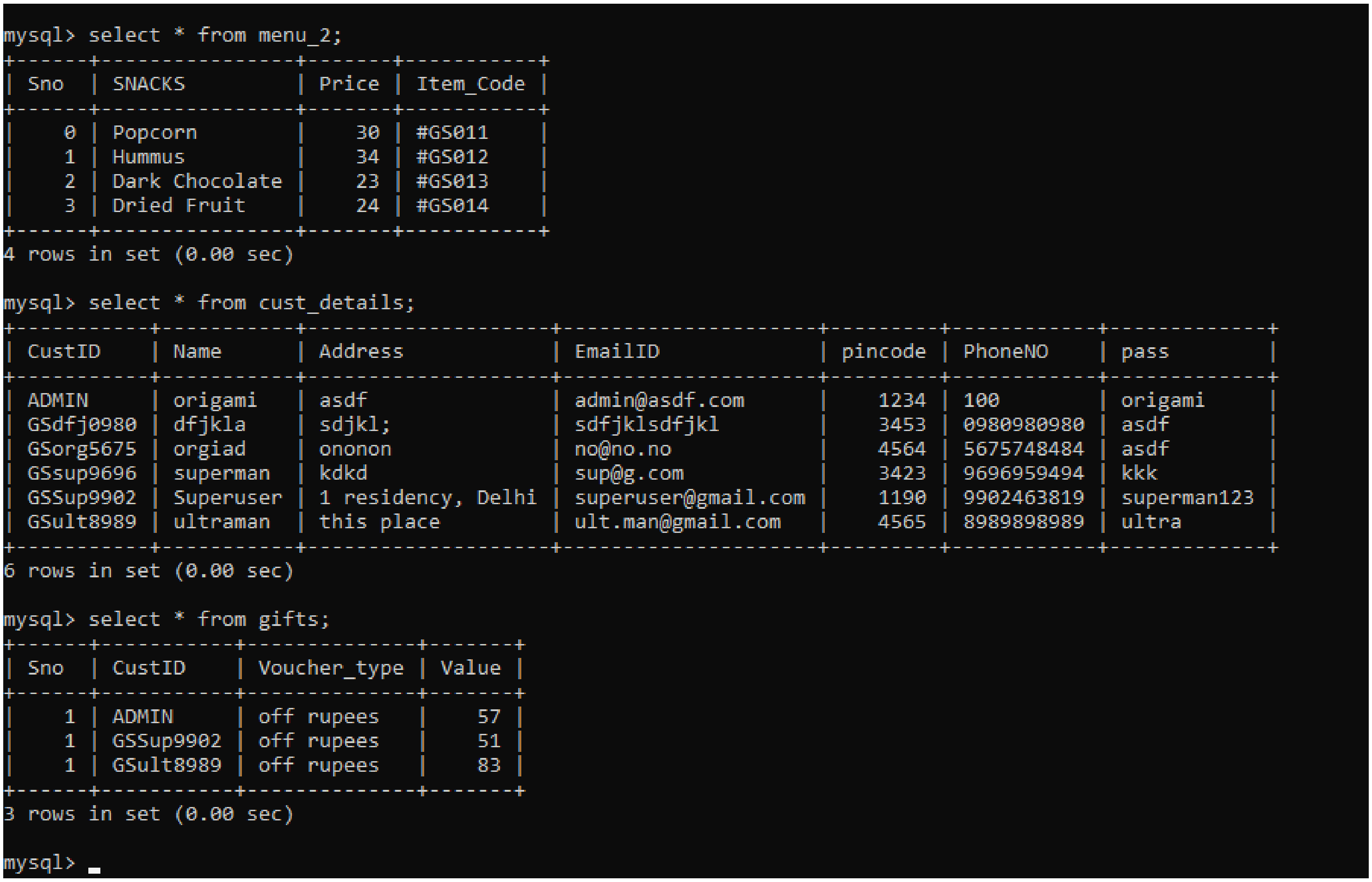


```
____________________________Thank you for shopping with us !____________________________

+--------+---------------------------------+----------+-----+-------+------------------+
| S. NO. | NAME OF ITEMS (Price Per Item) | QUANTITY | TAX | PRICE | EFFECTIVE PRICE  |
+--------+---------------------------------+----------+-----+-------+------------------+
|   1    |       Lacinato Kale (30)        |    23    | 18% |  690  | Rs.814.199999999 |
|   2    |          Hummus (34)            |    50    | 18% | 1700  |    Rs.2006.0     |
|   3    |          Popcorn (30)           |   212    | 18% | 6360  | Rs.7504.79999999 |
|   4    |        Strawberries (12)        |    11    | 18% |  132  |    Rs.155.76     |
+--------+---------------------------------+----------+-----+-------+------------------+
The total payable amount is: Rs.10481/-


____________________________Thank you for shopping with us !____________________________

+--------+--------------------------------+----------+-----+-------+-----------------+
| S. NO. | NAME OF ITEMS (Price Per Item) | QUANTITY | TAX | PRICE | EFFECTIVE PRICE |
+--------+--------------------------------+----------+-----+-------+-----------------+
|   1    |        Cauliflower (23)        |    50    | 18% |  1150 |    Rs.1357.0    |
|   2    |           Yogurt (24)          |    12    | 18% |  288  |    Rs.339.84    |
|   3    |        Organic Eggs (30)       |    6     | 18% |  180  | Rs.212.399999999 |
|   4    |           Apples (43)          |    12    | 18% |  516  |    Rs.608.88    |
+--------+--------------------------------+----------+-----+-------+-----------------+
Congratulations!! You got discount of Rs.41 on you bill of Rs.2518
The total payable amount is: Rs.2477/-
```

The database created:



```
mysql> use kushagra_project;
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_kushagra_project |
+----------------------------+
| cust_details               |
| gifts                      |
| menu_1                     |
| menu_2                     |
+----------------------------+
4 rows in set (0.02 sec)

mysql> select * from menu_1;
+------+---------------+-------+-----------+
| Sno  | Items         | Price | Item_Code |
+------+---------------+-------+-----------+
|    0 | Lacinato Kale |    30 | #GS000    |
|    1 | Baby Spinach  |    34 | #GS001    |
|    2 | Cauliflower   |    23 | #GS002    |
|    3 | Avocados      |    24 | #GS003    |
|    4 | Apples        |    43 | #GS004    |
|    5 | Strawberries  |    12 | #GS005    |
|    6 | Organic Eggs  |    30 | #GS006    |
|    7 | Kefir         |    34 | #GS007    |
|    8 | Almond Milk   |    23 | #GS008    |
|    9 | Yogurt        |    24 | #GS009    |
|   10 | Sheep's Milk  |    43 | #GS010    |
+------+---------------+-------+-----------+
11 rows in set (0.00 sec)

mysql> _
```



```
mysql> select * from menu_2;
+------+----------------+-------+-----------+
| Sno  | SNACKS         | Price | Item_Code |
+------+----------------+-------+-----------+
|    0 | Popcorn        |    30 | #GS011    |
|    1 | Hummus         |    34 | #GS012    |
|    2 | Dark Chocolate |    23 | #GS013    |
|    3 | Dried Fruit    |    24 | #GS014    |
+------+----------------+-------+-----------+
4 rows in set (0.00 sec)

mysql> select * from cust_details;
+-----------+-----------+--------------------+---------------------+---------+------------+-------------+
| CustID    | Name      | Address            | EmailID             | pincode | PhoneNO    | pass        |
+-----------+-----------+--------------------+---------------------+---------+------------+-------------+
| ADMIN     | origami   | asdf               | admin@asdf.com      |    1234 | 100        | origami     |
| GSdfj0980 | dfjkla    | sdjkl;             | sdfjklsdfjkl        |    3453 | 0980980980 | asdf        |
| GSorg5675 | orgiad    | ononon             | no@no.no            |    4564 | 5675748484 | asdf        |
| GSsup9696 | superman  | kdkd               | sup@g.com           |    3423 | 9696959494 | kkk         |
| GSSup9902 | Superuser | 1 residency, Delhi | superuser@gmail.com |    1190 | 9902463819 | superman123 |
| GSult8989 | ultraman  | this place         | ult.man@gmail.com   |    4565 | 8989898989 | ultra       |
+-----------+-----------+--------------------+---------------------+---------+------------+-------------+
6 rows in set (0.00 sec)

mysql> select * from gifts;
+------+-----------+--------------+-------+
| Sno  | CustID    | Voucher_type | Value |
+------+-----------+--------------+-------+
|    1 | ADMIN     | off rupees   |    57 |
|    1 | GSSup9902 | off rupees   |    51 |
|    1 | GSult8989 | off rupees   |    83 |
+------+-----------+--------------+-------+
3 rows in set (0.00 sec)

mysql>
```

ENHANCEMENTS

- The characteristic of a good program is the ability for it to be user friendly. I believe a good documentation for this could be made for it to become more accessible to any person
- A mode for staff only could also be made. Staff can then monitor the stats on how well a grocery is selling and monitor the expiry date
- The ability for users to withdraw their name from our database can also be implemented. This is to ensure the privacy of our customers

HARDWARE USED/ SOFTWARE USED

- MX linux
- Dell inspiron 3000
- 500gb HDD
- 4gb RAM

- Python 3.7.6
- IDLE
- MySql / MariaDB

BIBLIOGRAPHY

- Docs.python.org