

---

# INTRODUCTION

---

The project ShopZen(An Online Retail Store) created in Python aims to develop a complete application that will enable customers to buy products online.

There are 2 Stakeholders(modes of operation):

1. ADMIN
2. USER

## ADMIN:

Further admin has the following functionalities:

### ➤ STOCK MANAGEMENT:

**Add, Modify or Delete Products:** This option helps the admin to update their stock by displaying, inserting, modifying or deleting records(products and categories).

### ➤ STOCK ANALYSIS:

Admin is able to visualize stock, and monthly revenue with the help of graphs.

## USER:

### ➤ SIGN UP:

A new customer can register himself by entering a few details and creating their USERID and PASSWORD.

### ➤ LOGIN:

Existing customers can log in to their IDs by entering USERIDs and PASSWORDS, after which they are directed to various options.

---

# MODULES IMPORTED

---

S NO.	MODULE NAME	DESCRIPTION
1.	MySQL. Connector	To connect Python with MySQL.
2.	Tkinter	Python's standard Graphical User Interface (GUI) package.
3.	Pyplot	To plot different types of graphs.
4.	Random	To generate Random Numbers

---

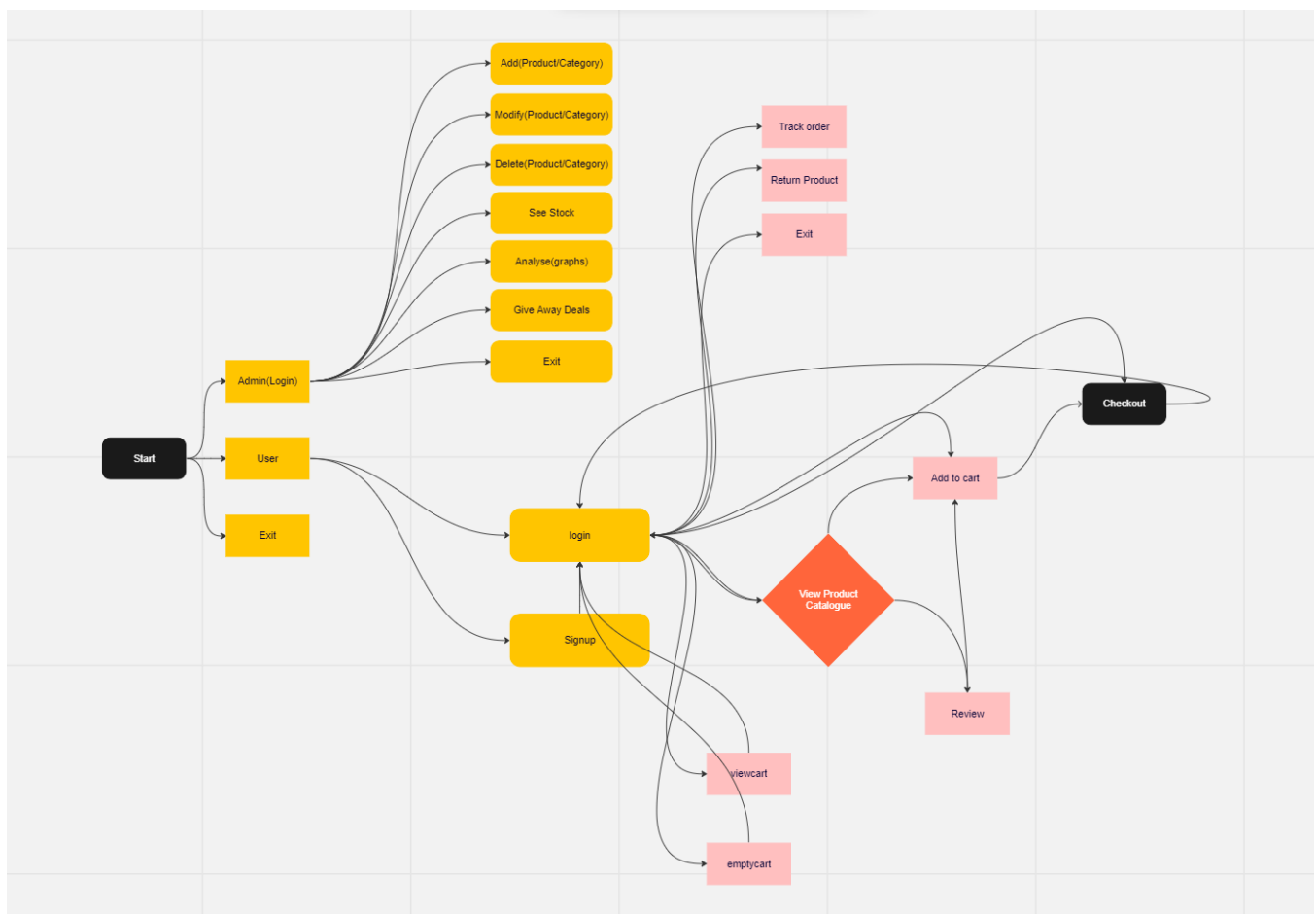
# USER-DEFINED FUNCTIONS

---

S NO.	FUNCTION	DESCRIPTION
1.	adminlogin()	Opens the admin window with admin options.
2.	custologin()	Opens the user window asking the user to sign in.
3.	Addp(),Addc()	Used to add products and categories.
4.	modifyp(),modify()	Used to modify product and categories.
5.	stodisplay()	Used to display records.
6.	deletep(),deletec()	Used to delete product and categories.
7.	Stograph()	Used to generate product vs qty graph.
8.	Sale()	Used to generate monthly revenue graph.
9.	Giveawaydeals()	Used to set deals and discounts.
10.	signup()	Open the account creation window.
11.	menu()	Used to display product catalogue to user.

<b>12.</b>	<code>addtocart()</code>	Used to add products to card.
<b>13.</b>	<code>Showavailabledeals()</code>	Used to show available deals.
<b>14.</b>	<code>Viewcart()</code>	Used to view items in the cart.
<b>15.</b>	<code>Viewmembership()</code>	Used to check membership status.
<b>16.</b>	<code>Emptycart()</code>	Used to empty cart by the user.
<b>17.</b>	<code>Viewreview()</code>	Used to view review of a product.
<b>18.</b>	<code>Trackorder()</code>	Used to track an order.
<b>19.</b>	<code>Checkout()</code>	Used to exitcart and final billing.
<b>20.</b>	<code>Returnproduct()</code>	Used to set return product status to yes.
<b>21.</b>	<code>User()</code>	Open the home window with login options.
<b>22.</b>	<code>supermainmenu()</code>	Opens the main page of our store.

# Flow Diagram



# SOURCE CODE

```
from tkinter import *
import mysql.connector
import random
print("\t", "*" * 75)
print("\t\tS H O P   Z E N   -   O N L I N E   R E T A I L   S T O R
E")
print("\t", "*" * 75)

'''done -
Addc(),Addp(),Modifyc(),Modifyp(),deletec(),deletep(),stodisplay(),stograph(),sale(),gi
veawaydeals()'''

conn = mysql.connector.connect(host="localhost", user="root", passwd="root",
database="group62")
cur=conn.cursor()

"""Admin portal"""
def Addc():
    print("")
    print("\t\t\t", "-" * 4, "Category", "-" * 4)
    print("\t\t\t", "_" * 41)
    # try:
    # Id = input("\t\t Enter Category Id\t\t\t:")
    Name = input("\t\t Enter Category Name\t\t\t:")
    Desc = input("\t\t Enter Description\t\t\t:")
    sql = "insert into category values ({},'{'','{'})".format(
        "NULL", Name, Desc)
    cur.execute(sql)
    cur.execute("commit")
    print("")
    print("\t\t\t", "-" * 32, sep="")
    print("\t\t\t\t|Category Inserted Successfully|")
    print("\t\t\t", "-" * 32, sep="")
    # except:
    #     print()
    #     print("\t\t\t", "*" * 38)
    #     print("\t\t\t\t*Category Id already exist, Try again*")
    #     print("\t\t\t", "*" * 38)
    #     print("\t\t\t", "_" * 41)
```

```
# Addc()

def Addp():
    conn = mysql.connector.connect(host="localhost", user="root", passwd="root",
database="group62")
    cur = conn.cursor()
    print("")
    print("\t\t\t", "-"*4, "Adding Product", "-"*4)
    print("\t\t", "_"*41)
    try:
        # Id = input("\t\t Enter Product Id\t\t :")
        CId = input("\t\t Enter Category Id\t\t :")
        if CId.isdigit():
            Name = input("\t\t Enter Product Name\t\t :")
            qty = input("\t\t Enter Qty\t\t\t :")
            price = input("\t\t Enter Price\t\t\t :")
            compname = input("\t\t Enter Company Name\t\t :")
            Desc = input("\t\t Enter Description\t\t :")
            dom = input("\t\t Enter Dom\t\t:")
            doe = input("\t\t Enter Doe\t\t:")
            sql = "insert into product values
({},{},'{}',{},{},'{}','{}','{}',{})".format(
                "NULL", CId,Name, qty, price, compname, Desc, dom, doe)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*31, sep="")
            print("\t\t\t|Product Inserted Successfully|")
            print("\t\t\t", "-"*31, sep="")
        else:
            print("\t\t\tInvalid Id, should be a Integer")
    except:
        print("\t\t\tCategory Id does not exist, Try again")
    Addp()

def modifyc():
    print("")
    print("\t\t\t", "-"*4, "Modifying Category", "-"*4)
    while True:
        print("\t\t", "_"*41)
        print("\t\t\tChoice\t\t|\tAction")
        print("\t\t", "_"*41)
        print("\t\t\t1.\t\t|\tCategory Name")
        print("\t\t\t2.\t\t|\tDescription")
        print("\t\t\t3.\t\t|\tExit")
        print("\t\t", "_"*41)
        x=int(input("\t\t\tEnter Your Choice\t:"))
        print("\t\t", "_"*41)
        if x==1:
```

```

Name=input("\t\t Enter Category Id To Be Modified\t:")
if Name.isdigit():
    query="select catid from category where catid={}".format(Name)
    cur.execute(query)
    rec=cur.fetchall()
    if rec!=[]:
        Namex=input("\t\t Enter New Category Name\t\t:")
        sql="update category set catname='{}' where
catid={}".format(Namex,Name)
        cur.execute(sql)
        cur.execute("commit")
        print("")
        print("\t\t\t", "-"*30, sep="")
        print("\t\t\t|Record Modified Successfully|")
        print("\t\t\t", "-"*30, sep="")
    else:
        print("\t\t\tId Does Not Exist")
else:
    print("Invalid Id, should be a Integer")
elif x==2:
    Ide=input("\t\t Enter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select catid from category where catid={}".format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Price=input("\t\t Enter New Description\t\t:")
            sql="update category set Description='{}' where
catid={}".format(Price,Ide)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*30, sep="")
            print("\t\t\t|Record Modified Successfully|")
            print("\t\t\t", "-"*30, sep="")
        else:
            print("\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be a Integer")
else:
    break

def modifyp():
    print("")
    print("\t\t\t", "-"*4, "Modifying product", "-"*4)
    while True:
        print("\t\t\t", "-"*41)

```



```

print("\t\t\tChoice\t\t|\tAction")
print("\t\t", "_"*41)
print("\t\t\t1.\t\t|\tProduct_Name")
print("\t\t\t2.\t\t|\tQuantity")
print("\t\t\t3.\t\t|\tPrice")
print("\t\t\t4.\t\t|\tCompanyName")
print("\t\t\t5.\t\t|\tdom")
print("\t\t\t6.\t\t|\tdoe")
print("\t\t\t7.\t\t|\tExit")
print("\t\t", "_"*41)
x=int(input("\t\t\tEnter Your Choice\t:"))
print("\t\t", "_"*41)
if x==1:
    Name=input("\t\t Enter Id To Be Modified\t:")
    if Name.isdigit():
        query="select pid from product where pid='{}'.format(Name)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Namex=input("\t\t Enter New Name\t\t:")
            sql="update product set pname='{}' where
pname='{}'.format(Name,Namex)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*30, sep="")
            print("\t\t\t|Record Modified Successfully|")
            print("\t\t\t", "-"*30, sep="")
        else:
            print("\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be an integer")
elif x==2:
    Ide=input("\t\t Enter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select pid from product where pid='{}'.format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            qty=input("\t\t Enter New Qty\t\t:")
            sql="update product set qty={} where pid='{}'.format(qty,Ide)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*30, sep="")
            print("\t\t\t|Record Modified Successfully|")
            print("\t\t\t", "-"*30, sep="")
        else:
            print("\t\t\tId Does Not Exist")

```

```

else:
    print("Invalid Id, should be an integer")
elif x==3:
    Ide=input("\t\t Enter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select pid from product where pid='{ }'".format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Price=input("\t\t Enter New Price\t\t:")
            sql="update product set Price={ } where pid='{ }'".format(Price,Ide)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*30, sep="")
            print("\t\t\t|Record Modified Successfully|")
            print("\t\t\t", "-"*30, sep="")
        else:
            print("\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be an integer")
elif x==4:
    Ide=input("\t\t Enter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select pid from product where pid='{ }'".format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Category=input("\t\t Enter New Company Name\t\t:")
            sql="update product set companyname='{ }' where
pid='{ }'".format(Category,Ide)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t", "-"*30, sep="")
            print("\t\t\t|Record Modified Successfully|")
            print("\t\t\t", "-"*30, sep="")
        else:
            print("\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be an integer")
elif x==5:
    Ide=input("\t\t Enter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select pid from product where pid='{ }'".format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Category=input("\t\t Enter New Dom\t\t:")

```

```

        sql="update product set dom='{}' where
pid='{}'".format(Category,Ide)
        cur.execute(sql)
        cur.execute("commit")
        print("")
        print("\t\t\t\t\t","-"*30,sep="")
        print("\t\t\t\t\t|Record Modified Successfully|")
        print("\t\t\t\t\t","-"*30,sep="")
    else:
        print("\t\t\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be an integer")
elif x==6:
    Ide=input("\t\t\t\t\tEnter Id To Be Modified\t:")
    if Ide.isdigit():
        query="select pid from product where pid='{}'".format(Ide)
        cur.execute(query)
        rec=cur.fetchall()
        if rec!=[]:
            Category=input("\t\t\t\t\tEnter New Doe\t\t:")
            sql="update product set doe='{}' where
pid='{}'".format(Category,Ide)
            cur.execute(sql)
            cur.execute("commit")
            print("")
            print("\t\t\t\t\t","-"*30,sep="")
            print("\t\t\t\t\t|Record Modified Successfully|")
            print("\t\t\t\t\t","-"*30,sep="")
        else:
            print("\t\t\t\t\tId Does Not Exist")
    else:
        print("Invalid Id, should be an integer")
else:
    break

def stodisplay():
    sql="select * from product"
    cur.execute(sql)
    rec=cur.fetchall()
    query="select catid,catname from category"
    cur.execute(query)
    recnew=cur.fetchall()
    ltemp=[]
    dtemp={}
    l=[]
    for i in rec:

```



```

try:
    Id=input("\t\t Enter Category Id To Be Deleted:")
    if Id.isdigit():
        sql="delete from category where catid={}".format(Id)
        cur.execute(sql)
        cur.execute("commit")
        print("")
        print("\t\t\t","-"*29,sep="")
        print("\t\t\t|Record Deleted Successfully|")
        print("\t\t\t","-"*29,sep="")
    else:
        print("\t\t\tInvalid Id, should be a integer")
except:
    print("\t\t\tCategory Id does not exist, Try Again")
    deletec()

def stograph():
    import matplotlib.pyplot as plt
    sql="select qty from product;"
    cur.execute(sql)
    rec=cur.fetchall()
    query="select pname from product;"
    cur.execute(query)
    rec1=cur.fetchall()
    l1=[]
    l2=[]
    for i in rec:
        for j in i:
            l1.append(int(j))
    for i in rec1:
        for j in i:
            l2.append(j)
    plt.title("ITEM STOCKS")
    plt.xlabel("ITEM")
    plt.ylabel("STOCKS")
    plt.bar(l2,l1,color=["crimson","aqua","deepskyblue","tomato"],align="center",width=
0.3)
    plt.xticks(rotation="vertical")
    plt.show()

# def sale():
#     import matplotlib.pyplot as plt
#     conn = mysql.connector.connect(host="localhost", user="root", passwd="root",
database="group62")
#     cur=conn.cursor()
#     print("\t\t","-"*41)
#     d=input("\t\t Enter Sales Date\t\t:")
#     sql="select sales, product from sale where sale_date='{ }' group by
product_name".format(d)

```

```

#     cur.execute(sql)
#     rec=cur.fetchall()
#     l1=[]
#     l2=[]
#     for i in rec:
#         l2.append(i[0])
#         l1.append(i[1])
#     plt.title("DAILY SALES REPORT-"+str(d))
#     plt.xlabel("ITEM")
#     plt.ylabel("SALES")
#     plt.bar(l1,l2,color=["crimson","aqua","deepskyblue","tomato"],align="center",width
h=0.1)
#     plt.show()

# def category_sale():
#     import matplotlib.pyplot as plt
#     conn = mysql.connector.connect(host="localhost", user="root", passwd="root",
database="group62")
#     cur=conn.cursor()
#     print("\t\t","_"*41)
#     sql="""SELECT c.catname, DATE_FORMAT(o.dop, '%m') AS month, SUM(p.price *
ca.qty)"revenue"
# FROM category c
# INNER JOIN product p ON c.catid = p.catid
# INNER JOIN orderprod op ON p.pid = op.pid
# INNER JOIN payment o ON op.orderid = o.orderid
# INNER JOIN cart ca ON ca.cid=o.cid
# GROUP BY c.catname, DATE_FORMAT(o.dop, '%m') with rollup;"
#     SELECT c.catname, DATE_FORMAT(o.dop, '%m') AS month, SUM(p.price *
ca.qty)"revenue""
#     cur.execute(sql)
#     rec=cur.fetchall()
#     l1=[]
#     l2=[]
#     for i in rec:
#         if i[1]!=None and i[2]!=0:
#             l1.append(i[0])
#             l2.append(i[2])
#     plt.title("Revenue Across Category")
#     plt.xlabel("Category")
#     plt.ylabel("Revenue")
#     plt.bar(l1,l2)
#     plt.xticks(rotation="vertical")
#     plt.show()
# # category_sale()

def sale():
    import matplotlib.pyplot as plt
    print("\t\t","_"*41)

```

```

    sql="""SELECT DATE_FORMAT(dop, '%m') AS month, SUM(totamount) AS revenue
FROM orders
JOIN payment ON orders.orderid = payment.orderid
GROUP BY month
ORDER BY month ASC;"""
    cur.execute(sql)
    rec=cur.fetchall()
    l1=[]
    l2=[]
    for i in rec:
        l1.append(i[0])
        l2.append(i[1])
    plt.title("Revenue Across Month")
    plt.xlabel("Month")
    plt.ylabel("Revenue")
    plt.bar(l1,l2)
    # plt.xticks(rotation="vertical")
    plt.show()

def giveawaydeals():
    while True:
        amount=input("\t\t Enter the amount(discount will be applied on order above
this amount):")
        discount=input("\t\t Enter the discount\t\t\t\t\t\t\t :")
        sql="insert into deals values ({},{})".format(amount,discount)
        cur.execute(sql)
        cur.execute("commit")
        print("\t\t\t-----")
        print("\t\t\tDeal Added Successfully")
        print("\t\t\t-----")
        ans=input("\t\t Do you want to continue(y/n)\t\t\t\t : ")
        if ans.lower()=="y":
            continue
        elif ans.lower()=="n":
            break
        else:
            print("\t\tInvalid choice, Try again")

"""Admin portal done"""#####

"""Customer Portal"""#####

def addtocart(pid,cid):
    print("\n")
    print("\t\t\t\t\t","_"*41)
    print('')
    \t\t\t\t\t

```

[illegible]

```
def menu(cid):
    sql="select * from category"
    cur.execute(sql)
    rec=cur.fetchall()
    print("\t\t\t\t\tProduct Catalogue")
    print("\t+-----+-----+-----+-----+-----+-----")
    print("\t|      Choice      |          Category_Name          |           Description           ")
    print("\t+-----+-----+-----+-----+-----+-----")
    count=1
    for i in rec:
        print('\t|'.format(str(count).rjust(10),str(i[1]).rjust(35),str(i[2]).rjust(45)))
        print("\t+-----+-----+-----+-----+-----+-----")
        count+=1
    ch=0
    while True:
        print("\t\t\t","*" * 70)
        choice=int(input("\t\t\tEnter Your Choice Of Category to See Related Products\t:"))
        print("\t\t\t","*" * 70)
        res=rec[choice-1]
        # print(res)
        sql="select * from product where catid='{ }'".format(res[0])
        cur.execute(sql)
        rec1=cur.fetchall()
```



```

        print("\t+-----+-----+-----+-----+")
        print("\t| Product_Id | Product_Name | price | Company")
        print("\t| Description |")
        print("\t+-----+-----+-----+-----+")
        for i in rec1:
            print('\t|
{}|{}|{}|{}|{}|'.format(str(i[0]).rjust(11),str(i[2]).rjust(25),str(i[4]).rjust(15),str
(i[5]).rjust(25),str(i[6]).rjust(32)))
            print("\t+-----+-----+-----+-----+")
        count+=1
        review=input("\t\t\tDo you also want to see review of the product(y/n) : ")
        if review.lower()=="y":
            rpid=input("\t\t\tEnter product id for the review of product you want to
see:")
            qurev="select review from review where pid={}".format(rpid)
            cur.execute(qurev)
            reco1=cur.fetchall()
            quname="select pname from product where pid={}".format(rpid)
            cur.execute(quname)
            reco2=cur.fetchall()
            print("\n\t\t\t\t-----")
            print("\t\t\t\tProduct Name | Description")
            print("\t\t\t\t-----")
            print("\t\t\t\t",reco2[0][0],"\t\t\t",reco1[0][0])
            print("\t\t\t\t-----")
            y=input("\n\t\t\tAdd to Cart or View More(For Cart(C)/For More(M))? :")
            if y.lower()=="m":
                continue
            else:
                print("\n")
                print("\t\t\t\t-----")
                prodid=input("\t\t\tEnter product id for the product that you want to add
to cart:")
                print("\t\t\t\t-----")
                addtocart(prodid,cid)
                return

def showavailabledeals():
    sql="select * from deals;"
    cur.execute(sql)
    rec=cur.fetchall()
    print("\n\t\t\tDeals : ")
    print("\t\t\t+-----+-----+-----+-----+")
    print("\t\t\t| Orders Above Amount | Discount |")

```

```

print("\t\t+-----+-----+")
for i in rec:
    print('\t\t| {}|{}|'.format(str(i[0]).rjust(20),str(i[1]).rjust(15)))
    print("\t\t+-----+-----+")
return

def viewmembership(cid):
    sql="select membership from customer where cid = {}".format(cid)
    cur.execute(sql)
    rec=cur.fetchall()
    print(rec)
    if rec[0][0]=='yes':
        print("\t\t\t\tYou are a member")
    else:
        print("\t\t\t\tYou are currently not a member")

def viewcart(cid):
    query="select * from cart where cid={}".format(cid)
    cur.execute(query)
    rec=cur.fetchall()
    print("\n\t\tCart : ")
    print("\t\t+-----+-----+-----+")
    print("\t\t|   Cart Id |   Product Name   |   Quantity   |")
    print("\t\t+-----+-----+-----+")
    for i in rec:
        sql="select pname from product where pid={}".format(i[0])
        cur.execute(sql)
        recnew=cur.fetchall()
        print('\t\t|
{}|{}|{}|'.format(str(i[2]).rjust(9),str(recnew[0][0]).rjust(21),str(i[1]).rjust(15)))
        print("\t\t+-----+-----+-----+")
    return

def emptycart(cid):
    try:
        id=input("\t\t\tEnter cartid for the product you want to delete :")
        sql="delete from cart where cid={}".format(cid)
        cur.execute(sql)
        cur.execute("commit")
        print("\t\t\tRecord Deleted Succesfully")
    except:
        print("\t\t\tId does not exist, Try Again")

def viewreview(pid):
    querev="select review,cid from review where pid={}".format(pid)
    cur.execute(querev)
    reco1=cur.fetchall()
    quaname="select pname from product where pid={}".format(pid)

```

```

cur.execute(quname)
reco2=cur.fetchall()
print("\n\t\tReview : ")
print("\t\t\t+-----+-----+-----+")
print("\t\t\t|      Product Name      |      Review      |      User Name      |")
print("\t\t\t+-----+-----+-----+")
for i in reco1:
    sql="select cname from customer where cid={}".format(i[1])
    cur.execute(sql)
    recsql=cur.fetchall()
    print('\t\t\t|
{}|{}|{}|'.format(str(reco2[0][0]).rjust(20),str(i[0]).rjust(18),str(recsql[0][0]).rjus
t(21)))
    print("\t\t\t+-----+-----+-----+")
return

def trackorder(cid):
    sql="select orderid from orders where cid={}".format(cid)
    cur.execute(sql)
    rec=cur.fetchall()
    d=dict()
    for i in rec:
        cur.execute("select empid from delivery_details where orderid={}".format(i[0]))
        emprec=cur.fetchall()
        cur.execute("select empname from delivery_person where
empid={}".format(emprec[0][0]))
        empnamerec=cur.fetchall()
        cur.execute("select paystatus,dod from delivery_details where
orderid={}".format(i[0]))
        rec=cur.fetchall()
        d[i[0]]=tuple([empnamerec[0][0],rec[0][0],rec[0][1]])
    print("\n\t\tTrack Order : ")
    print("\t\t\t+-----+-----+-----+
-----+")
    print("\t\t\t|      orderid      |      Employee Name      |      Pay Status      |      Date of
Delivery |")
    print("\t\t\t+-----+-----+-----+
-----+")
    for i in d:
        print('\t\t\t|
{}|{}|{}|{}|'.format(str(i).rjust(14),str(d[i][0]).rjust(21),str(d[i][1]).rjust(19),str
(d[i][2]).rjust(19)))
        print("\t\t\t+-----+-----+-----+
-----+")
    # except:
    #     print("Id does not exist, Try Again")
    return

count=0

```



```

am=sum(tamount)+tax
print("\n\t Customer Name\t\t :",name[0][0])
print("\t Total Amount\t\t\t :",sum(tamount))
print("\t Delivery Charge(@10%\t :",tax)
print("\t Amount To Be Paid\t\t :",am)
print("\n")
print("\t\t\t+-----+")
print("\t\t\t|Thank You,Repay your Visit|")
print("\t\t\t+-----+")
print("\t ", " _____ "*10)
print("\t ", " / _____ / "*10)

return

def returnproduct(cid):
    cur.execute("select orderid,cid from orders where cid={}".format(cid))
    rec=cur.fetchall()
    print("\t\t\t-----")
    print("\t\t\t Orderid | cid")
    print("\t\t\t-----")
    for i in rec:
        print("\t\t\t ",i[0],"\t | \t",i[1])
    print("\t\t\t-----")
    id=input("\t\t\tEnter orderid that you want to return : ")
    cur.execute("select empid from delivery_details where orderid={}".format(id))
    recid=cur.fetchall()
    cur.execute("insert into returnsp value
({},{},{},'{}','{}','{}')".format(id,recid[0][0],"NULL","NO","NO","NO"))
    cur.execute("commit")
    print("\n\t\t\t -----")
    print("\t\t\t Return Initiated Successfully")
    print("\t\t\t -----")
    return

"""Customer Portal Done"""#####

def adminlogin(window):
    conn = mysql.connector.connect(
        host="localhost", user="root", passwd="root", database="group62")
    cur = conn.cursor()
    from tkinter import messagebox
    window.destroy()
    window1 = Tk()
    window1.geometry("750x530")
    window1.resizable(width=False, height=False)
    photo = PhotoImage(file='C:\\Users\\tusha\\Downloads\\adbgimage.png')
    l = Label(window1, image=photo)
    l.pack()
    image = PhotoImage(file='C:\\Users\\tusha\\Downloads\\adminp.png')
    l1 = Label(window1, image=image)

```

```

11.place(x=280, y=90)
12 = Label(window1, text="ADMIN LOGIN", font=(
    "Lucida handwriting", 18, "bold"), bg="black", fg="sky blue")
12.place(x=230, y=10, height=70, width=250)
13 = Label(window1, text="ENTER ADMINID      :", font=(
    "Lucida", 20, "bold"), bg="black", fg="khaki")
13.place(x=100, y=250)
14 = Label(window1, text="ENTER PASSWORD:", font=(
    "Lucida", 20, "bold"), bg="black", fg="khaki")
14.place(x=100, y=340)
us = StringVar()
ps = StringVar()
x = Entry(window1, textvariable=us, font=(
    "Lucida", 16, "bold"), borderwidth=5)
x.place(x=450, y=250)
y = Entry(window1, textvariable=ps, font=(
    "Lucida", 16, "bold"), show="*", borderwidth=5)
y.place(x=450, y=340)

def show():
    hidebutton = Button(window1, image=hideimage,
        command=hide, borderwidth=5)
    hidebutton.place(x=670, y=343)
    y.config(show="")

def hide():
    showbutton = Button(window1, image=showimage,
        command=show, borderwidth=5)
    showbutton.place(x=670, y=343)
    y.config(show="*")
showimage = PhotoImage(file='C:\\Users\\tusha\\Downloads\\showimage.png')
hideimage = PhotoImage(file='C:\\Users\\tusha\\Downloads\\hideimage.png')
showbutton = Button(window1, image=showimage, command=show, borderwidth=5)
showbutton.place(x=670, y=343)

def myfun():
    conn = mysql.connector.connect(
        host="localhost", user="root", passwd="root", database="group62")
    cur = conn.cursor()
    usid = us.get()
    pas = ps.get()
    query = "select aid from admin where aid = '{}'.format(usid)"
    cur.execute(query)
    reco = cur.fetchall()
    if reco != []:
        sql = "select apassword from admin where aid='{}'.format(usid)"
        cur.execute(sql)
        rec = cur.fetchall()
        if rec[0][0] == pas:

```

```

messagebox.showinfo("Admin Message", "Sign in Successfully")
window1.destroy()
while True:
    print("\t\t", "_"*41)
    print("\t\t", "^"*8, "\tAdmin Portal\t", "^"*9)
    print("\t\t", "_"*41)
    print("\t\t\t\tChoice\t\t|\tAction")
    print("\t\t", "_"*41)
    print("\t\t\t1.\t\t|\tAdd")
    print("\t\t\t2.\t\t|\tModify")
    print("\t\t\t3.\t\t|\tDisplay")
    print("\t\t\t4.\t\t|\tDelete")
    print("\t\t\t5.\t\t|\tSet Give Away Deals")
    print("\t\t\t6.\t\t|\tAnalysis")
    print("\t\t\t7.\t\t|\tExit")
    print("\t\t", "_"*41)
    choice = int(input("\t\t\tEnter Your Choice\t:"))
    if choice == 1:
        ques = input("\t\t\tDo you want to add product or
category(p/c):")

        print("\t\t", "_"*41)
        if ques.lower() == 'p':
            Addp()
        elif ques.lower() == 'c':
            Addc()
        else:
            print("\t\t\tInvalid choice, Try again!")
    elif choice == 2:
        ques = input("\t\t\tDo you want to modify product or
category(p/c):")

        print("\t\t", "_"*41)
        if ques == 'p':
            modifyp()
        elif ques == 'c':
            modifyc()
        else:
            print("\t\t\tInvalid choice, Try again!")
    elif choice==3:
        print("\t\t", "_"*41)
        stodisplay()
    elif choice==4:
        ques = input("\t\t\tDo you want to delete product or
category(p/c):")

        print("\t\t", "_"*41)
        if ques == 'p':
            deletep()
        elif ques == 'c':
            deletec()
        else:

```

```

        print("\t\t\tInvalid choice, Try again!")
    elif choice==5:
        giveawaydeals()
    elif choice==6:
        while True:
            print("\t\t","_"*41)
            print("\t\t\tChoice\t\t|\tAction")
            print("\t\t","_"*41)
            print("\t\t\t1.\t\t|\tStockVsItem")
            print("\t\t\t2.\t\t|\tMonthly Revenue")
            print("\t\t\t3.\t\t|\tExit")
            print("\t\t","_"*41)
            Choice=int(input("\t\t\tEnter Your Choice\t:"))
            if Choice==1:
                stograph()
            elif Choice==2:
                sale()
            else:
                break
    elif choice==7:
        window2=Tk()
        mainmenu(window2)
        break
    else:
        print("\t\t","_"*41)
        print("\t\t\tInvalid Entry")
        print("")
        ans=input("\t\t\tDo You Want To Cont.(y/n):")
        if ans=="y" or ans=="Y":
            continue
        else:
            break

    else:
        messagebox.showerror("Error","Incorrect Password, Try Again")
else:
    messagebox.showerror("Error","User Does Not Exist")
    Button(text="Log
In",command=myfun,bg="black",font=("elephant",20),fg="khaki").place(x=160,y=420,height=
80,width=180)
    Button(text="BACK <--
",command=lambda:mainmenu(window1),bg="black",font=("elephant",20),fg="khaki").place(x=
400,y=420,height=80,width=180)
    window1.mainloop()

```



```

def signup(window):
    window.destroy()
    conn=mysql.connector.connect(host="localhost",user="root",passwd="root",database="group62")
    cur=conn.cursor()
    from tkinter import messagebox
    window1=Tk()
    window1.resizable(width=False,height=False)
    image=PhotoImage(file='C:\\Users\\tusha\\Downloads\\custsignimage.png')
    l1=Label(window1,image=image)
    l1.pack()
    l2=Label(window1,text="Customer Signup",font=("lucida",20,"bold"),bg="black",fg="linen")
    l2.place(x=200,y=5,height=70,width=300)
    l3=Label(window1,text="ENTER YOUR NAME",font=("Lucida",20,"bold"),bg="black",fg="linen")
    l3.place(x=50,y=100)
    l4=Label(window1,text="ENTER PHONE NO",font=("lucida",20,"bold"),bg="black",fg="linen")
    l4.place(x=50,y=150)
    l5=Label(window1,text="ENTER USER-ID",font=("lucida",20,"bold"),bg="black",fg="linen")
    l5.place(x=50,y=200)
    l8=Label(window1,text="ENTER PASSWORD",font=("Lucida",20,"bold"),bg="black",fg="linen")
    l8.place(x=50,y=250)
    l7=Label(window1,text="Sex(M/F)",font=("Lucida",20,"bold"),bg="black",fg="linen")
    l7.place(x=50,y=300)
    l6=Label(window1,text="ENTER DOB(yy-mm-dd)",font=("lucida",20,"bold"),bg="black",fg="linen")
    l6.place(x=50,y=350)
    name=StringVar()
    ph=StringVar()
    uid=StringVar()
    upswd=StringVar()
    usex=StringVar()
    udb=StringVar()
    try:
        x1=Entry(window1,textvariable=name,font=("Lucida",18,"bold"),borderwidth=5)
        x1.place(x=450,y=100)
        x2=Entry(window1,textvariable=ph,font=("Lucida",18,"bold"),borderwidth=5)
        x2.place(x=450,y=150)
        x3=Entry(window1,textvariable=uid,font=("Lucida",18,"bold"),borderwidth=5)
        x3.place(x=450,y=200)
        x4=Entry(window1,textvariable=upswd,font=("Lucida",18,"bold"),borderwidth=5)
        x4.place(x=450,y=250)
        x5=Entry(window1,textvariable=usex,font=("Lucida",18,"bold"),borderwidth=5)
        x5.place(x=450,y=300)

```

```

x6=Entry(window1,textvariable=udb,font=("Lucida",18,"bold"),borderwidth=5)
x6.place(x=450,y=350)
def myfun():
    conn=mysql.connector.connect(host="localhost",user="root",passwd="root",database="group62")
    cur=conn.cursor()
    sql="insert into customer values
('{'','{'','{'','{'','{'','{'','{'')'.format(uid.get(),upswd.get(),name.get(),udb.get(),
usex.get(),"no",ph.get())
    cur.execute(sql)
    cur.execute("commit")
    messagebox.showinfo("Admin Message","Sign up Successfully")
    user(window1)
    conn.close()
    Button(text="Sign
up",command=myfun,bg="black",fg="linen",font=("lucida",22)).place(x=160,y=430)
    Button(text="BACK <--
",command=lambda:user(window1),bg="black",fg="linen",font=("lucida",22)).place(x=450,y=
430)
    window1.mainloop()
except:
    print("\t\t\t\t-----")
    print("\t\t\t\tWrong Date Format , Try Again")
    print("\t\t\t\t-----")

def custologin(window2):
    conn=mysql.connector.connect(host="localhost",user="root",passwd="root",database="g
roup62")
    cur=conn.cursor()
    from tkinter import messagebox
    window2.destroy()
    window=Tk()
    window.resizable(width=False,height=False)
    hoto=PhotoImage(file='C:\\Users\\tusha\\Downloads\\custloginbg.png')
    l1=Label(window,image=hoto)
    l1.pack()
    image=PhotoImage(file='C:\\Users\\tusha\\Downloads\\custlogin.png')
    l1=Label(window,image=image)
    l1.place(x=310,y=90)
    l2=Label(window,text="USER
LOGIN",font=("Lucida",20,"bold"),bg="darkorange4",fg="snow")
    l2.place(x=240,y=10,height=70,width=300)
    l3=Label(window,text="ENTER
USERID      :",font=("Lucida",20,"bold"),bg="black",fg="khaki1")
    l3.place(x=110,y=280)
    l4=Label(window,text="ENTER
PASSWORD:",font=("Lucida",20,"bold"),bg="black",fg="khaki1")
    l4.place(x=110,y=330)

```

```

us=StringVar()
ps=StringVar()
x=Entry(window,textvariable=us,borderwidth=5,font=("Lucida",18,"bold"))
x.place(x=420,y=280)
y=Entry(window,textvariable=ps,show="*",borderwidth=5,font=("Lucida",18,"bold"))
y.place(x=420,y=330)
def show():
    hidebutton = Button(window, image=hideimage,
                        command=hide, borderwidth=5)
    hidebutton.place(x=660, y=335)
    y.config(show="")

def hide():
    showbutton = Button(window, image=showimage,
                       command=show, borderwidth=5)
    showbutton.place(x=660, y=335)
    y.config(show="*")
showimage = PhotoImage(file='C:\\Users\\tusha\\Downloads\\showimage.png')
hideimage = PhotoImage(file='C:\\Users\\tusha\\Downloads\\hideimage.png')
showbutton = Button(window, image=showimage, command=show, borderwidth=5)
showbutton.place(x=660, y=335)
def myfun():
    conn=mysql.connector.connect(host="localhost",user="root",passwd="root",database="group62")
    cur=conn.cursor()
    usid=us.get()
    pas=ps.get()
    query="select cid from customer where cid={}".format(usid)
    cur.execute(query)
    reco=cur.fetchall()
    if reco!=[]:
        sql="select cpassword from customer where cid='{}'.format(usid)
        cur.execute(sql)
        rec=cur.fetchall()
        if rec[0][0]==pas:
            messagebox.showinfo("Admin Message","Logged In Successfully")
            window.destroy()
            while True:
                print("\t\t", "_"*41)
                print("\t\t", "^"*9, "\tShopZen\t", "^"*9)
                print("\t\t", "_"*41)
                print("\t\t\tChoice\t\t|\tAction")
                print("\t\t\t1. \t\t|\tExplore Product Catalogue")
                print("\t\t\t2. \t\t|\tShow Available Deals")
                print("\t\t\t3. \t\t|\tAdd Product to Cart")
                print("\t\t\t4. \t\t|\tView Membership Status")
                print("\t\t\t5. \t\t|\tView Cart")
                print("\t\t\t6. \t\t|\tEmpty Cart")
                print("\t\t\t7. \t\t|\tView Review")

```

```

        print("\t\t\t8. \t\t|\tExit Cart/Check Out")
        print("\t\t\t9. \t\t|\tTrack your order")
        print("\t\t\t10. \t\t|\tReturn a product")
        print("\t\t\t>10.\t\t|\tExit")
        print("\t\t", "_"*41)
        choice=int(input("\t\t\tEnter Your Choice\t:"))
        print("\t\t", "_"*41)
        if choice==1:
            menu(usid)
        elif choice==2:
            showavailabledeals()
        elif choice==3:
            prodid=input("\t\t\tEnter product id for the product that you
want to add to cart:")
            addtocart(prodid,usid)
        elif choice==4:
            viewmembership(usid)
        elif choice==5:
            viewcart(usid)
        elif choice==6:
            emptycart()
        elif choice==7:
            prodid=input("\t\t\tEnter product id for the product that you
want to add to cart:")
            viewreview(prodid)
        elif choice==8:
            checkout(usid)
        elif choice==9:
            trackorder(usid)
        elif choice==10:
            returnproduct(usid)
        else:
            window1=Tk()
            mainmenu(window1)
            break

    else:
        messagebox.showerror("Error","Incorrect Password, Try Again")
    else:
        messagebox.showerror("Error","User Does Not Exist")
    Button(text="Log
In",command=myfun,bg="black",fg="khaki",font=("elephant",22)).place(x=130,y=430)
    Button(text="BACK <--
",command=lambda:user(window),bg="black",fg="khaki",font=("elephant",22)).place(x=450,y
=430)
    window.mainloop()

```

```

def mainmenu(window):
    global hoto
    global photo
    global oto
    window.destroy()
    window1=Tk()
    window1.resizable(width=False,height=False)
    hoto=PhotoImage(file='C:\\Users\\tusha\\Downloads\\sebimage.png')
    Label(window1,image=hoto).pack()
    photo=PhotoImage(file='C:\\Users\\tusha\\Downloads\\adimage.png')
    Label(window1,image=photo).place(x=100,y=40)
    oto=PhotoImage(file='C:\\Users\\tusha\\Downloads\\usimage.png')
    Label(window1,image=oto).place(x=900,y=40)
    Button(text="ADMIN",command=lambda:adminlogin(window1),bg="black",fg="khaki",font=(
"lucida",26)).place(x=50,y=240,width=300,height=80)
    Button(text="USER",command=lambda:user(window1),font=("lucida",26),bg="black",fg="k
haki").place(x=850,y=240,width=300,height=80)
    def myfun1():
        window1.destroy()
        Button(text="EXIT",command=myfun1,bg="black",fg="khaki",font=("lucida",26)).place(x
=450,y=450,width=300,height=80)

def user(window):
    global photo
    window.destroy()
    window2=Tk()
    window2.geometry("800x510")
    window2.resizable(width=False,height=False)
    photo=PhotoImage(file='C:\\Users\\tusha\\Downloads\\usercolimage.png')
    Label(window2,image=photo).pack()
    l2=Label(window2,text="USER'S COLUMN",font=("Lucida
Handwriting",26,"bold"),bg="black",fg="khaki")
    l2.place(x=210,y=10,height=70,width=400)
    a=StringVar()
    a.set("Login")
    Button(window2,text="Signup",command=lambda:signup(window2),bg="bisque2",font=("lu
cida",26,"bold")).place(x=30,y=170,width=300,height=100)
    Button(window2,text="Login",command=lambda:custologin(window2),bg="bisque2",font=(
"lucida",26,"bold")).place(x=485,y=170,width=300,height=100)
    Button(text="BACK <--",command=lambda:mainmenu(window2),bg="slate
gray",font=("elephant",26)).place(x=250,y=400,width=300,height=100)
    window2.mainloop()

def supermainmenu():
    global photo,hoto
    window=Tk()

```

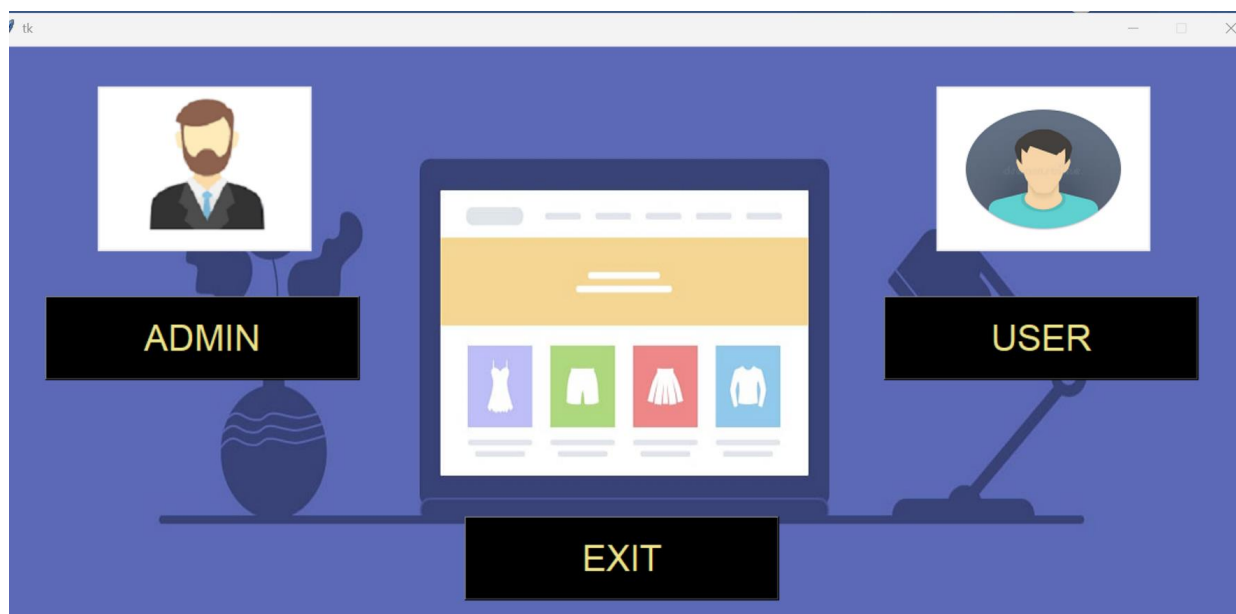
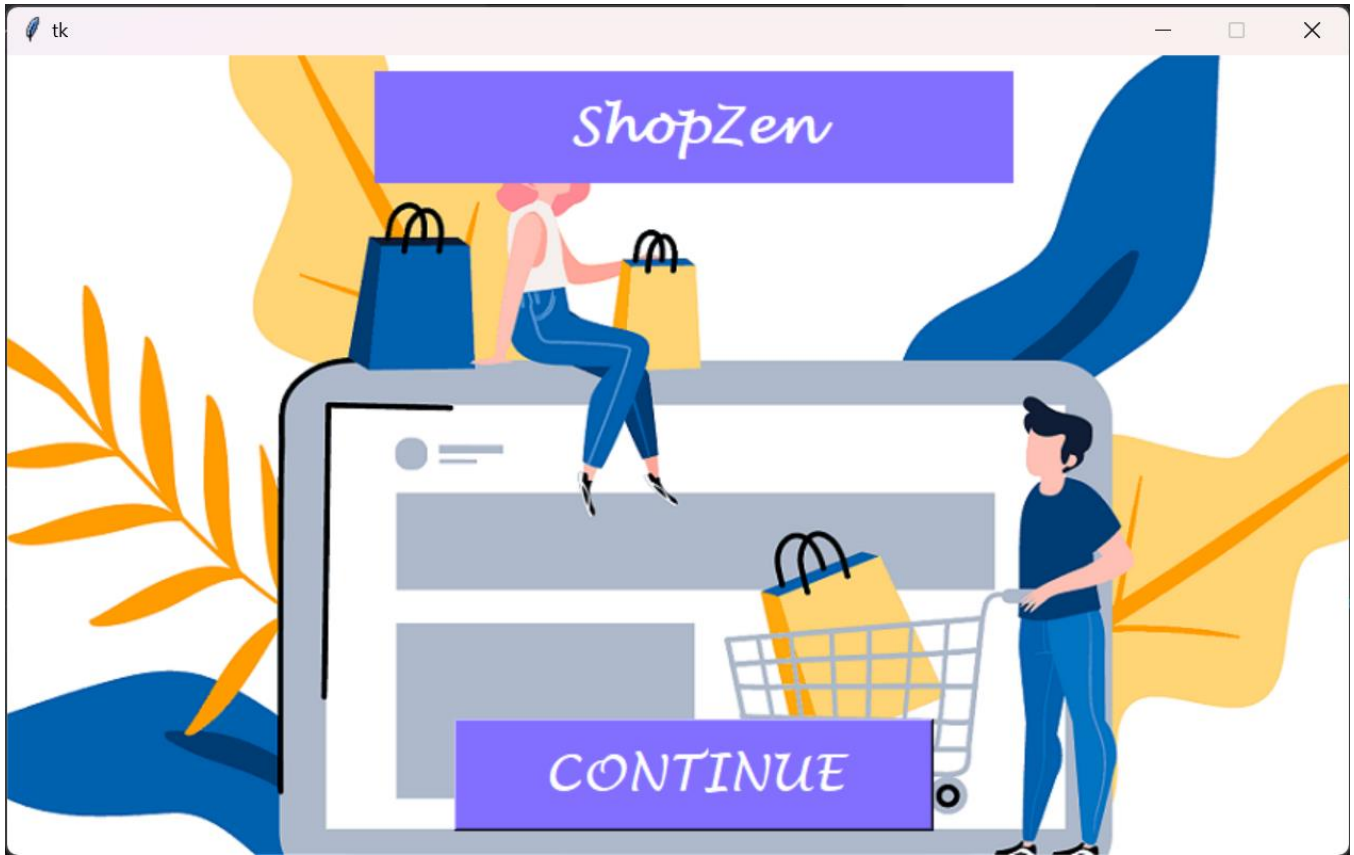
```
window.geometry("840x500")
# window.geometry("940x590")
window.resizable(width=False,height=False)
photo=PhotoImage(file='C:\\Users\\tusha\\Downloads\\inimage.png')
Label(window,image=photo).pack()
l2=Label(window,text="ShopZen",font=("Lucida
Handwriting",24,"bold"),bg="Slateblue1",fg="white")
l2.place(x=230,y=10,height=70,width=400)
Button(text="CONTINUE",command=lambda:mainmenu(window),bg="Slateblue1",fg="white",f
ont=("lucida handwriting",24)).place(x=280,y=415,height=70,width=300)
window.mainloop()

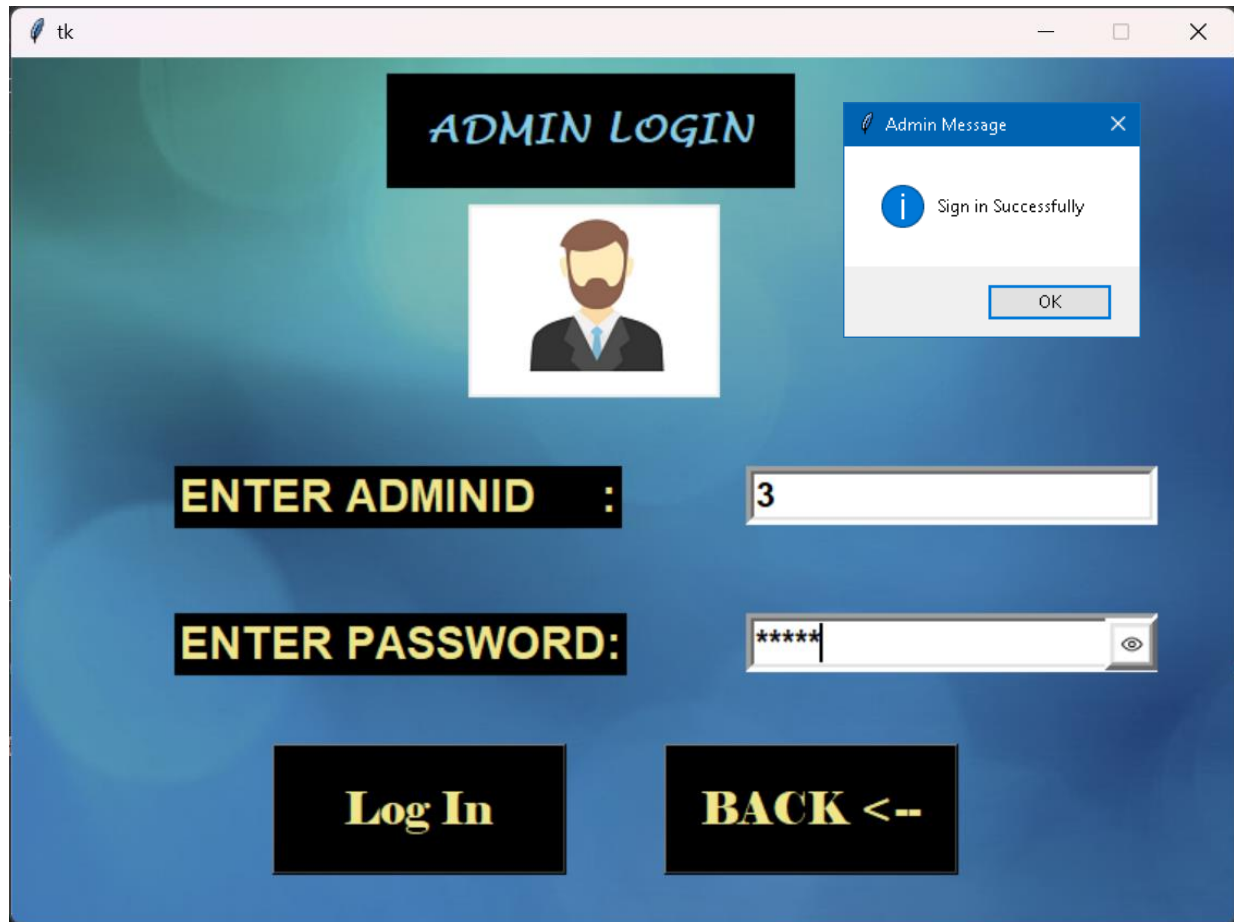
supermainmenu()
```

---

## OUTPUTS (PYTHON)

---







```

*****
S H O P Z E N - O N L I N E R E T A I L S T O R E
*****

```

```

^^^^^^^^^ Admin Portal ^^^^^^^^^^

```

Choice	Action
1.	Add
2.	Modify
3.	Display
4.	Delete
5.	Set Give Away Deals
6.	Analysis
7.	Exit

```

-----
Enter Your Choice      :1
Do you want to add product or category(p/c):c
-----

```

```

---- Category ----

```

```

-----
Enter Category Name      :electrical
Enter Description        :electrical appliances
-----

```

```

-----
|Category Inserted Successfully|
-----

```

3.	Display
4.	Delete
5.	Set Give Away Deals
6.	Analysis
7.	Exit

```

-----
Enter Your Choice      :2
Do you want to modify product or category(p/c):p
-----

```

```

---- Modifying product ----

```

Choice	Action
1.	Product_Name
2.	Quantity
3.	Price
4.	CompanyName
5.	dom
6.	doe
7.	Exit

```

-----
Enter Your Choice      :1
-----
Enter Id To Be Modified :2
Enter New Name          :heater
-----

```

```

-----
|Record Modified Successfully|
-----

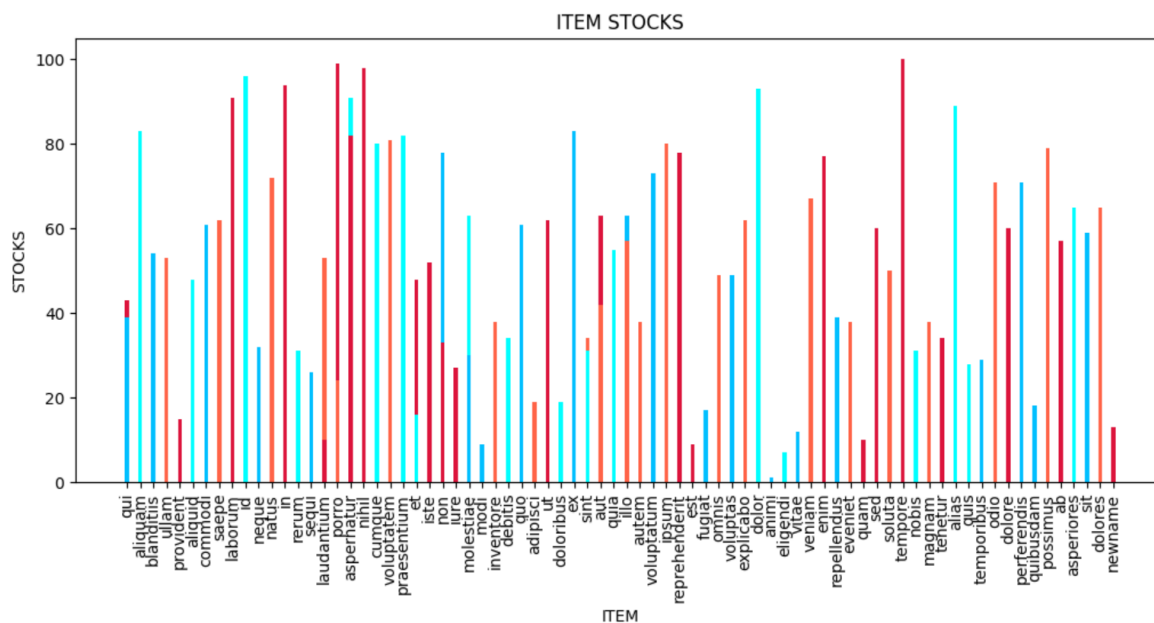
```

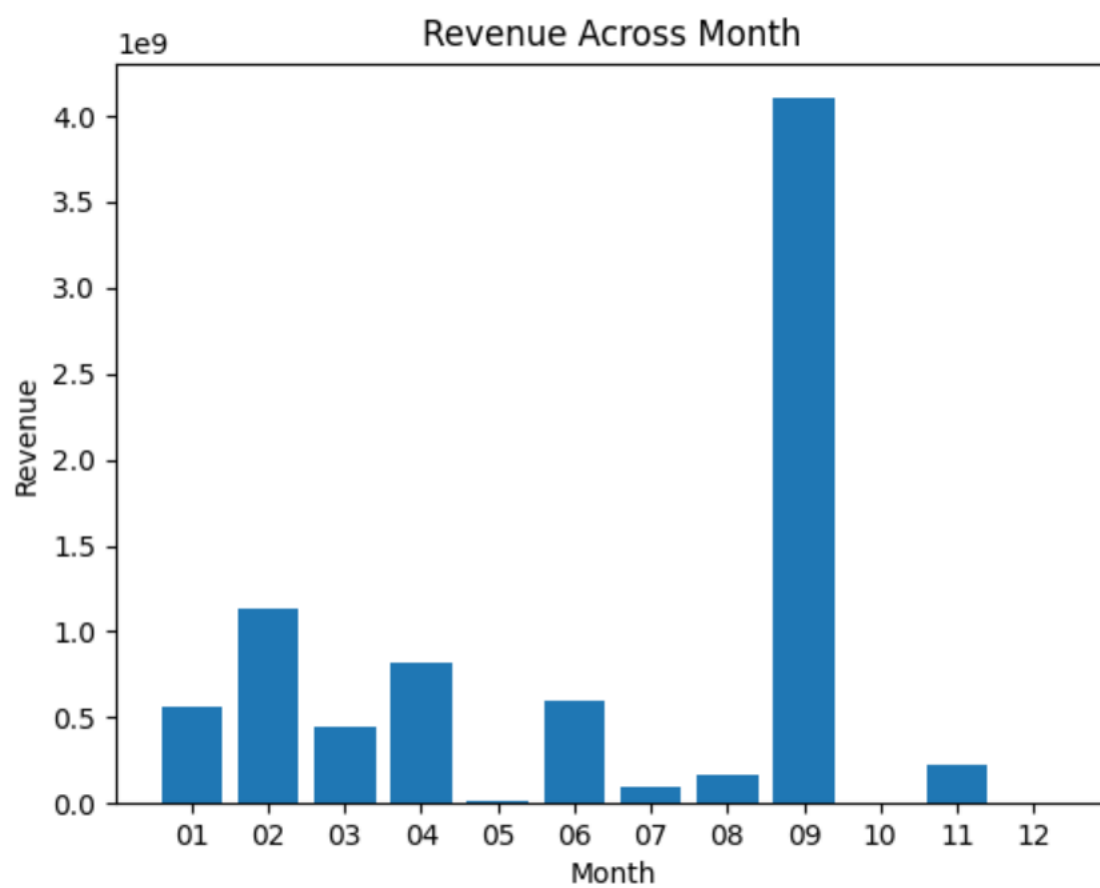
-----								
Enter Your Choice :3								
-----								
Records :								
-----								
Product_ID	Product_Name	Category_Name	qty	price	Company Name	dom	doe	
-----								
1	qui	Wunsch-Marks	43	683134.0	Greenfelder, Mo	1998-06-17	1993-03-31	
-----								
2	aliquam	Stokes Group	83	2704660.0	Anderson, Strom	2016-11-14	1983-05-27	
-----								
3	blanditiis	Berge-Greenfelder	54	0.0	Walsh-Jenkins	1982-09-06	2015-10-08	
-----								
4	ullam	Reinger Inc	53	477498.0	Stoltenberg Ltd	2015-06-10	1999-02-28	
-----								
5	provident	Donnelly-Little	15	57697800.0	Ebert, Predovic	2001-11-11	1997-12-16	
-----								
6	aliquid	Zemlak PLC	48	24435800.0	Schmitt, Dickin	1970-05-29	2007-01-24	
-----								
7	commodi	Doyle Ltd	61	62907.2	Kreiger, Hartma	1976-01-03	2005-09-03	
-----								
8	saepe	Marvin, Harris and Schuppe	62	31.6	Waters-Hermisto	1977-09-12	2002-10-31	
-----								
9	laborum	King and Sons	91	1011170.0	Lueilwitz-Hilpe	1979-12-29	1987-03-02	
-----								
10	id	Murray, Hauck and Ledner	29	1835920.0	Dibbert Ltd	2004-10-24	2000-09-15	
-----								
11	neque	Howell, McLaughlin and Schowal	32	0.0	Ebert-Larson	1973-05-10	2015-02-14	
-----								

Admin Portal	
Choice	Action
1.	Add
2.	Modify
3.	Display
4.	Delete
5.	Set Give Away Deals
6.	Analysis
7.	Exit
-----	
Enter Your Choice :4	
Do you want to delete product or category(p/c):c	
-----	
---- Deleting Category ----	
-----	
Enter Category Id To Be Deleted:56	
-----	
Record Deleted Successfully	
-----	

Choice	Action
1.	Add
2.	Modify
3.	Display
4.	Delete
5.	Set Give Away Deals
6.	Analysis
7.	Exit

Enter Your Choice :5  
Enter the amount(discount will be applied on order above this amount):15000  
Enter the discount :25  
-----  
Deal Added Successfully  
-----  
Do you want to continue(y/n) : n





### Customer Signup

ENTER YOUR NAME :

tushar

ENTER PHONE NO :

4328953

ENTER USER-ID :

120

ENTER PASSWORD :

tushar

Sex(M/F) :

M

ENTER DOB(yy-mm-dd):

23-12-11

Sign up

BACK <--


Admin Message

i

Sign In Successfully

OK

### USER LOGIN



ENTER USERID :

120

ENTER PASSWORD:

tushar

Log In

BACK <--

Admin Message

i

Logged In Successfully

OK

```

ShopZen
-----
Choice      |      Action
1.           |      Explore Product Catalogue
2.           |      Show Available Deals
3.           |      Add Product to Cart
4.           |      View Membership Status
5.           |      View Cart
6.           |      Empty Cart
7.           |      View Review
8.           |      Exit Cart/Check Out
9.           |      Track your order
10.          |      Return a product
>10.         |      Exit

-----
Enter Your Choice      :1

```

```

Product Catalogue
-----
+-----+-----+-----+
| Choice | Category_Name | Description |
+-----+-----+-----+
| 1 | Wunsch-Marks | Odio amet eos sit et. Au |
+-----+-----+-----+
| 2 | Stokes Group | Quas molestiae dolorum a |
+-----+-----+-----+
| 3 | Berge-Greenfelder | Repellendus cum assumend |
+-----+-----+-----+
| 4 | Reinger Inc | Excepturi exercitationem |
+-----+-----+-----+

```

```

+-----+-----+-----+
| 101 | fhasklfas | fsda;kjfasd;klas;ld |
+-----+-----+-----+
| 102 | electrical | electrical appliances |
+-----+-----+-----+

```

```

*****
Enter Your Choice Of Category to See Related Products :4
*****

```

```

+-----+-----+-----+-----+-----+
| Product_Id | Product_Name | price | Company Name | Description |
+-----+-----+-----+-----+-----+
| 4 | ullam | 477498.0 | Stoltenberg Ltd | Numquam vel totam sint dolorem |
+-----+-----+-----+-----+-----+
| 104 | newname | 31242.0 | mycom | fsjaf |
+-----+-----+-----+-----+-----+

```

```

Do you also want to see review of the product(y/n) : y
Enter product id for the review of product you want to see:4

```

```

-----
Product Name | Description
-----
ullah | excelent
-----

```

```

Enter Your Choice      :2

```

```

Deals :

```

```

+-----+-----+
| Orders Above Amount | Discount |
+-----+-----+
| 5000 | 5 |
+-----+-----+
| 8000 | 10 |
+-----+-----+
| 12000 | 20 |
+-----+-----+
| 15000 | 25 |
+-----+-----+

```

```

      \ | \ | \ | \
    $$$$ \ $$$$ $ $ $ $
    $$_/ $ $ $ $ $ $
    $ $ $ $ $ $ $ $
    $$$$ \ $ $ $ $ $
    $$_/ $ $ $ $ $ $
    $ $ $ $ $ $ $ $
    $$$$ \ $$$$ \ $$$$ \ $$$$
  
```

```

+-----+-----+-----+-----+
| Product_Name | Unit Price | Quantity | Price |
+-----+-----+-----+-----+
| provident    | 57697800.0| 33       | 1904027400.0|
+-----+-----+-----+-----+
  
```

```

Customer Name      : Elyse
Total Amount       : 1904027400.0
Delivery Charge(@10%) : 190402740.0
Amount To Be Paid  : 2094430140.0
  
```

```

+-----+
|Thank You,Repay your Visit|
+-----+
  
```

```

6.      | Empty Cart
7.      | View Review
8.      | Exit Cart/Check Out
9.      | Track your order
10.     | Return a product
>10.   | Exit
  
```

```

-----
Enter Your Choice      :10
-----
  
```

```

-----
Orderid | cid
-----
1       | 5
29      | 5
132     | 5
133     | 5
-----
  
```

```

Enter orderid that you want to return : 29
  
```

```

-----
Return Initiated Sucessfully
-----
  
```

# Olap queries and Triggers :

```
select paytype,dop,count(*)"Amount" from payment
where cid is not null
group by paytype,dop with rollup;
```

```
SELECT catname, DATE_FORMAT(doe, '%M')"month", COUNT(*) AS num_expired
FROM category
JOIN product USING (catid)
WHERE doe < NOW()
GROUP BY catname,DATE_FORMAT(doe, '%M') with rollup;
```

```
SELECT cname, DATE_FORMAT(dop, '%M')"month", COUNT(orderid) AS num_orders
FROM customer
JOIN payment USING (cid)
GROUP BY cname, DATE_FORMAT(dop, '%M') with rollup;
```

```
SELECT c.catname, DATE_FORMAT(o.dop, '%m') AS month, SUM(p.price * ca.qty)"revenue"
FROM category c
INNER JOIN product p ON c.catid = p.catid
INNER JOIN orderprod op ON p.pid = op.pid
INNER JOIN payment o ON op.orderid = o.orderid
INNER JOIN cart ca ON ca.cid=o.cid
GROUP BY c.catname, DATE_FORMAT(o.dop, '%m') with rollup;
```

delimiter //

```
CREATE TRIGGER update_product_stock
AFTER INSERT ON cart
```





```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE product
```

```
    SET qty = qty - NEW.qty
```

```
    WHERE pid = NEW.pid;
```

```
END //
```

```
delimiter ;
```

```
drop trigger update_product_stock;
```

```
select * from cart;
```

```
select * from product;
```

```
insert into cart values(57,10,104,51)
```

```
delimiter //
```

```
CREATE TRIGGER update_membership
```

```
AFTER UPDATE ON orders
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    if old.totamount>5000 then
```

```
        update customer
```

```
        set membership="yes"
```

```
        where cid=new.cid;
```

```
END if;
```

```
END //
```

```
delimiter ;
```

# Conflicting Transactions

## Transaction T1

### A)

- i.)  
Select \* from cartid where cartid=1;
- ii.)  
Update cart set qty=qty+1  
Where cartid=1;
- iii.)  
Select cid from customer  
Where cid=1;
- iv.)  
Update customer set cname="fdskj"  
Where cid = 1;

## Transaction T2

### B)

- i.)  
Select \* from cartid where cartid=1;
- ii.)  
Update cart set qty=qty+1  
Where cartid=1;
- iii.)  
Select cid from customer  
Where cid=1;
- iv.)  
Update customer set cname="fdskj"  
Where cid = 1;

The two transactions in this case conflicting because they are changing the same tables and rows. In this situation, a conflicting serializable schedule can be built upon by interchanging non-conflicting queries, and likewise, a non-conflicting schedule can also be created by exchanging the conflicting statements in the given transactions.

# Non-Conflicting Transaction

use group62;

## **Transaction T1**

Select \* from products where cid=1 and pid=1;

Commit;

## **Transaction T2**

Select paytype from payment where ordered=1 and cid=1;

Commit;

## **Transaction T3**

Update cart set qty=100 where cid=1;

Commit;

## **Transaction T4**

Insert into deals values

(100,10) ;

Commit;

The four transactions will not produce conflict other since they are reading and writing different parts of the database and are only reading from the same memory locations. They do not involve any shared data, which can cause conflicts if accessed at the same time.

Transactions T1 and T2 here are accessing completely different tables and therefore is no overlap in the data that they are accessing; hence there, both transactions can be executed simultaneously.

Transaction T3 and T4 do not have any overlapping data as the table they are accessing is different for both of them. T3 is updating the cart, while T4 is inserting values to deals which is completely different from T1, T2, T3. Hence, all the transactions can be executed simultaneously without conflicts.

As a result, there are no interference between the transactions in this case. Additionally, to ensure data consistency, concurrency control techniques like shared and exclusive locks can be utilised which further prevents conflicts.