**AISSCE 2019-20**

**COMPUTER SCIENCE PROJECT**

**ON**

**ONLINE SHOPPING**

CERTIFICATE

This is to certify that Anushree Bajaj, Roll no: \_\_\_\_\_\_\_\_\_\_ Class XII Science, Year 2019-20 of Sushila Birla Girls’ School, Kolkata has successfully completed the project titled ‘Online Shopping’.

(Internal examiner) (External examiner)

ACKNOWLEDGEMENT

I, Anushree Bajaj would like to express my sincere gratitude towards my computer teacher Mrs.S.Seth for her inspiration and support in completion of the project.

CONTENTS

* INTRODUCTION
* HARDWARE , SOFTWARE USED
* PROJECT DESIGN

FRONT END

BACK END

* TECHNICAL DESIGN
* SOURCE CODE
* OUTPUT
* BIBLIOGRAPHY

INTRODUCTION

The project, **Online shopping,** caters to the working of an online shopping website. The application allows user to view items add it in his/her cart and proceed to pay for the same. The application displays the details of the past orders.

The application allows user to create profile, insert update, delete data in their profile. Registered users can update, delete item in their carts, and then proceed to pay. Tentative date of delivery and additional distance charges are calculated on the basis of the address entered by the user.

The proposed system is interactive, fast and user friendly for the end users. The purpose of the whole application is to ease the daily or regular activities of online shopping.

HARDWARE

* MONITOR
* CPU
* KEYBOARD
* MOUSE
* LASER JET PRINTER
* INTEL CORE i7
* PROCESSOR SPEED -2.81 GHZ

SOFTWARE

* OPERATING SYSTEM – MICROSOFT WINDOWS 10
* FRONT END – PYTHON 3.7.4
* BACKEND – MYSQL 8.0 SERVER

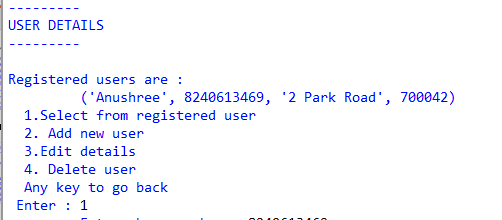
**FRONTEND**

**USER INTERFACE**

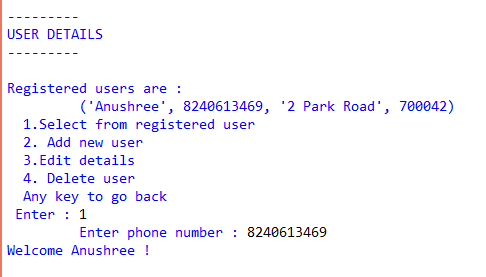
**Module** **1** **:** **Home**



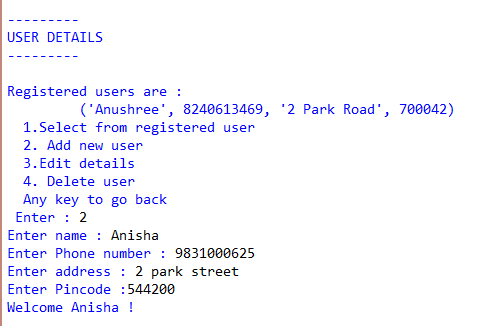
**Module** **2** **:** **User** **Profile**



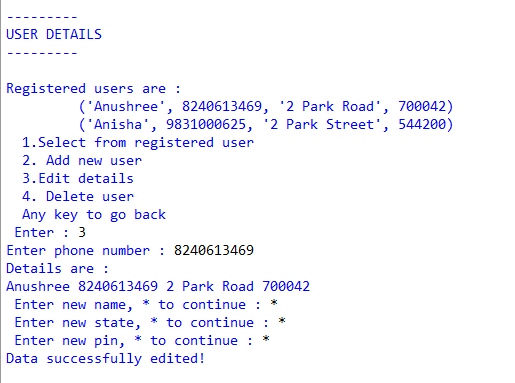
1.Selecting user



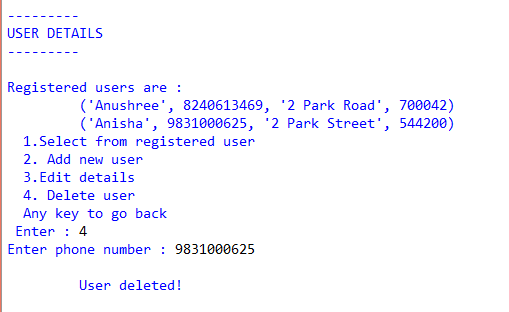
2.Adding new user



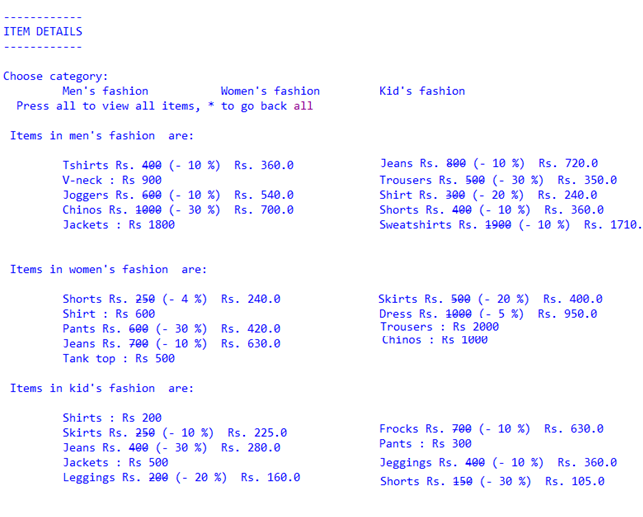
3.Updating user profile



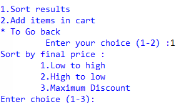
4.Deleting user profile



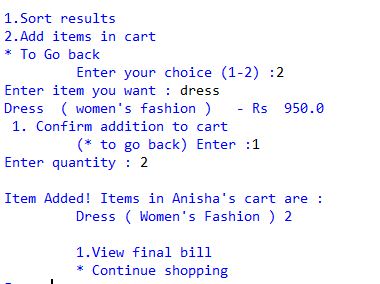
**Module** **3** **:** **Item** **Details** 1.Displaying Items



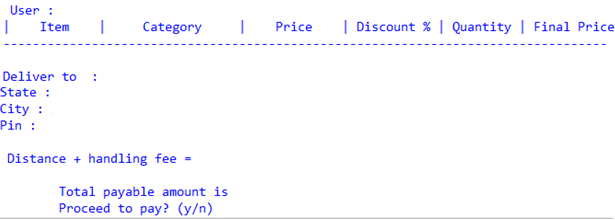
2.Sorting items



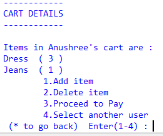
3.Addition to cart



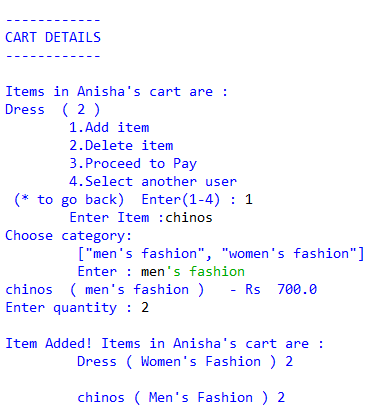
4.Check out



**Module** **4** **:** **Cart** **Details**

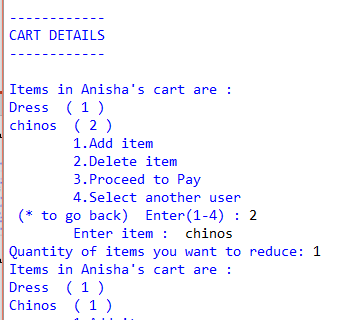


1.Addition to cart

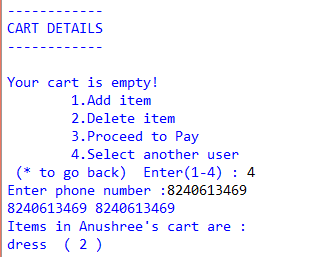


2.Delete from cart

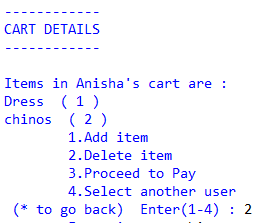
c



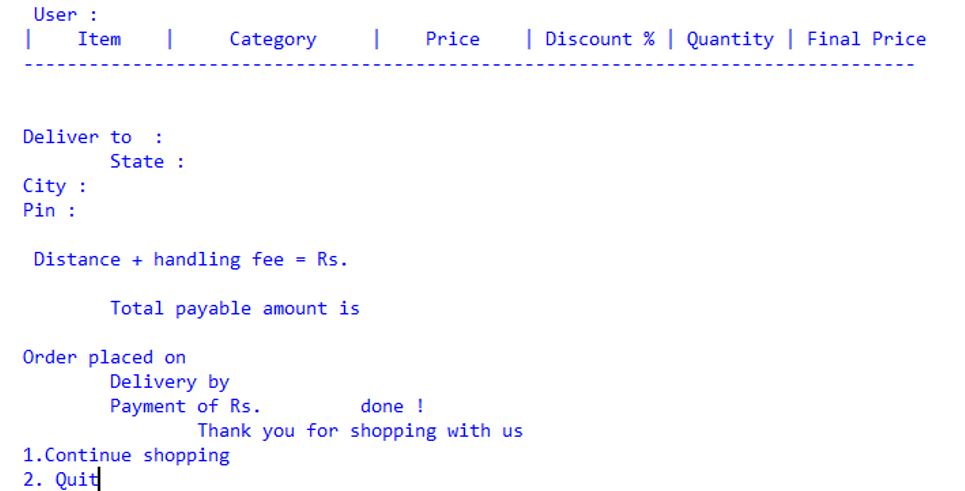
3.Selecting another user



4.Check out



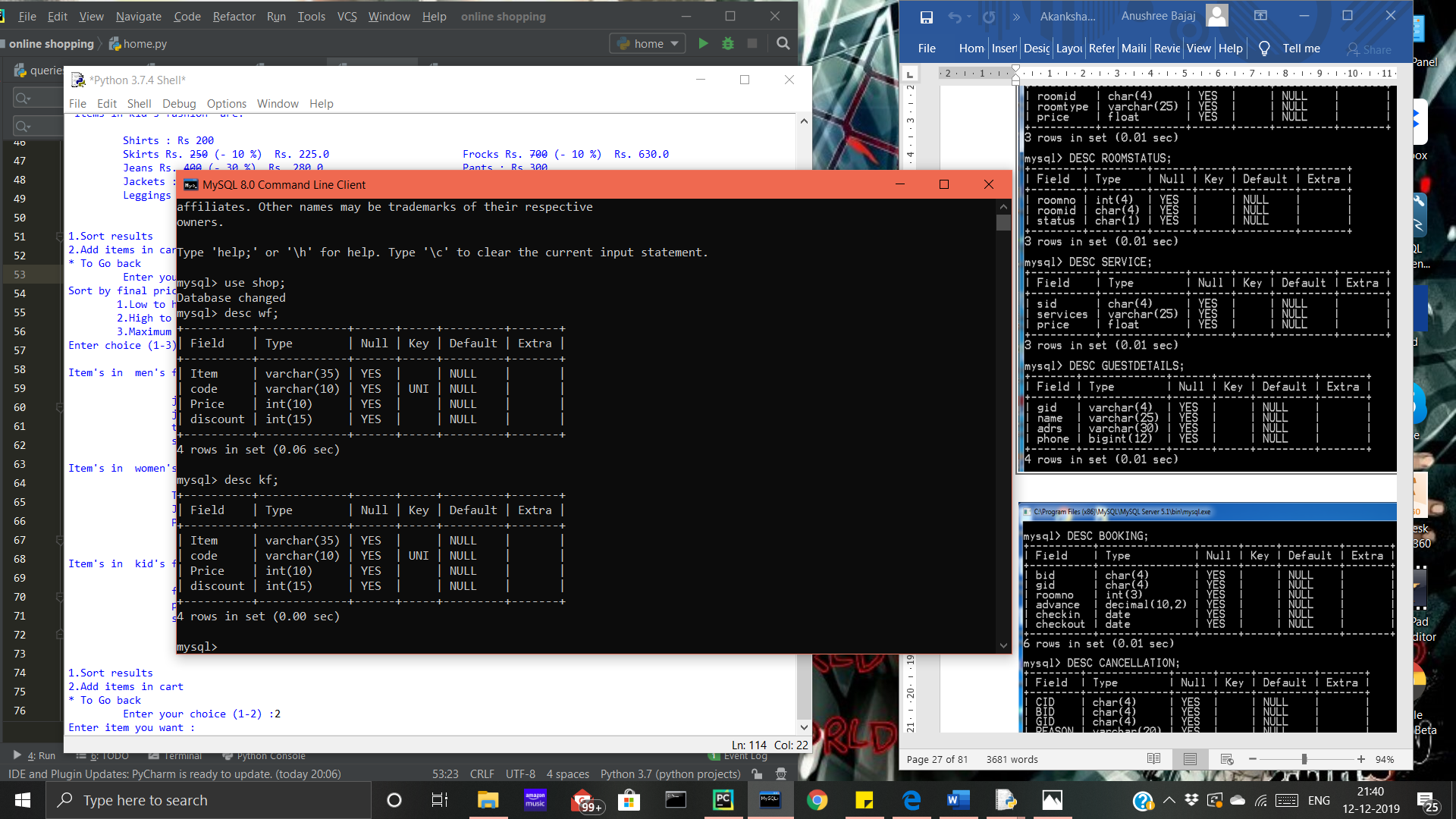
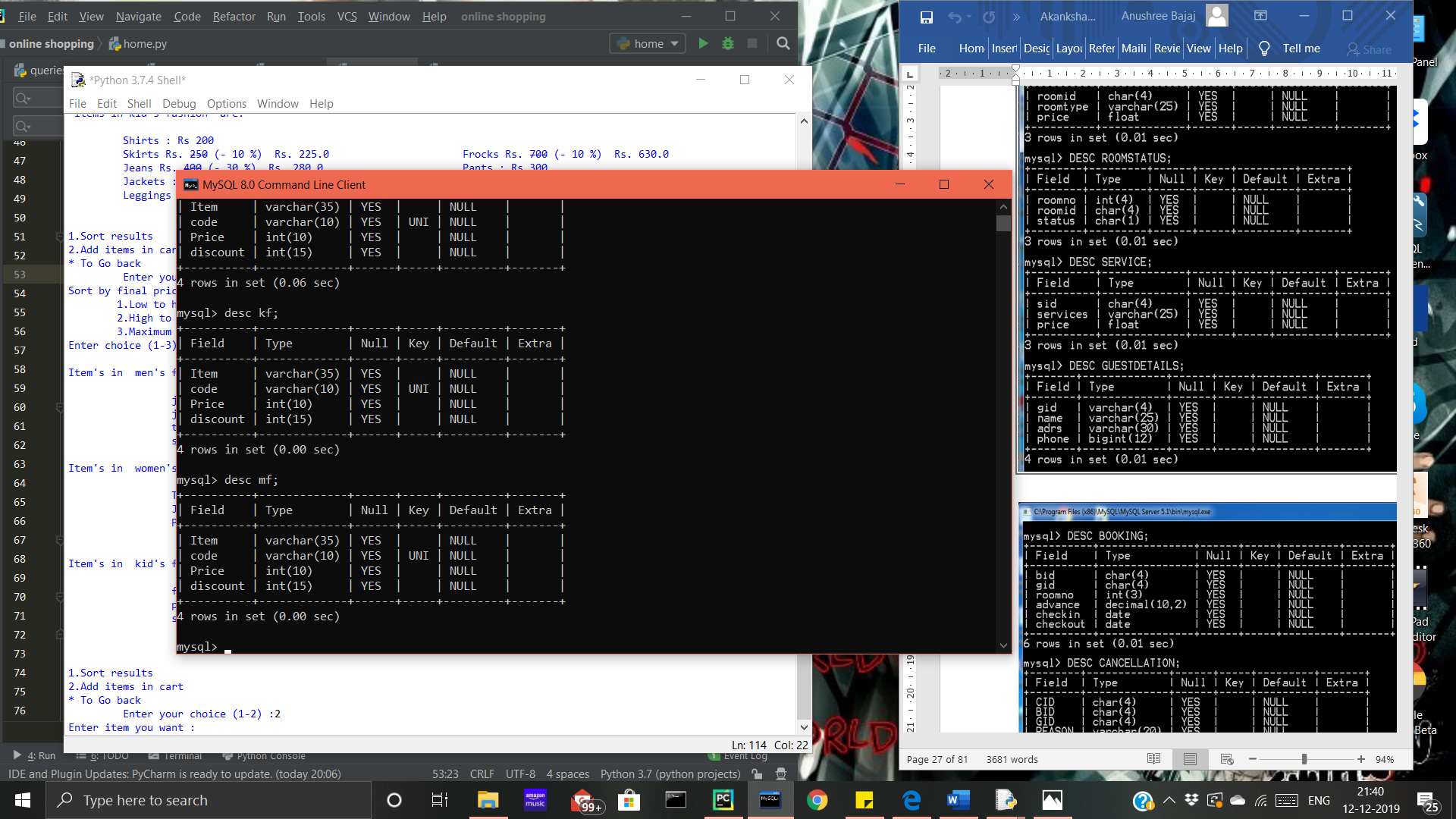
**Module** **5** **:** **Check** **Out**

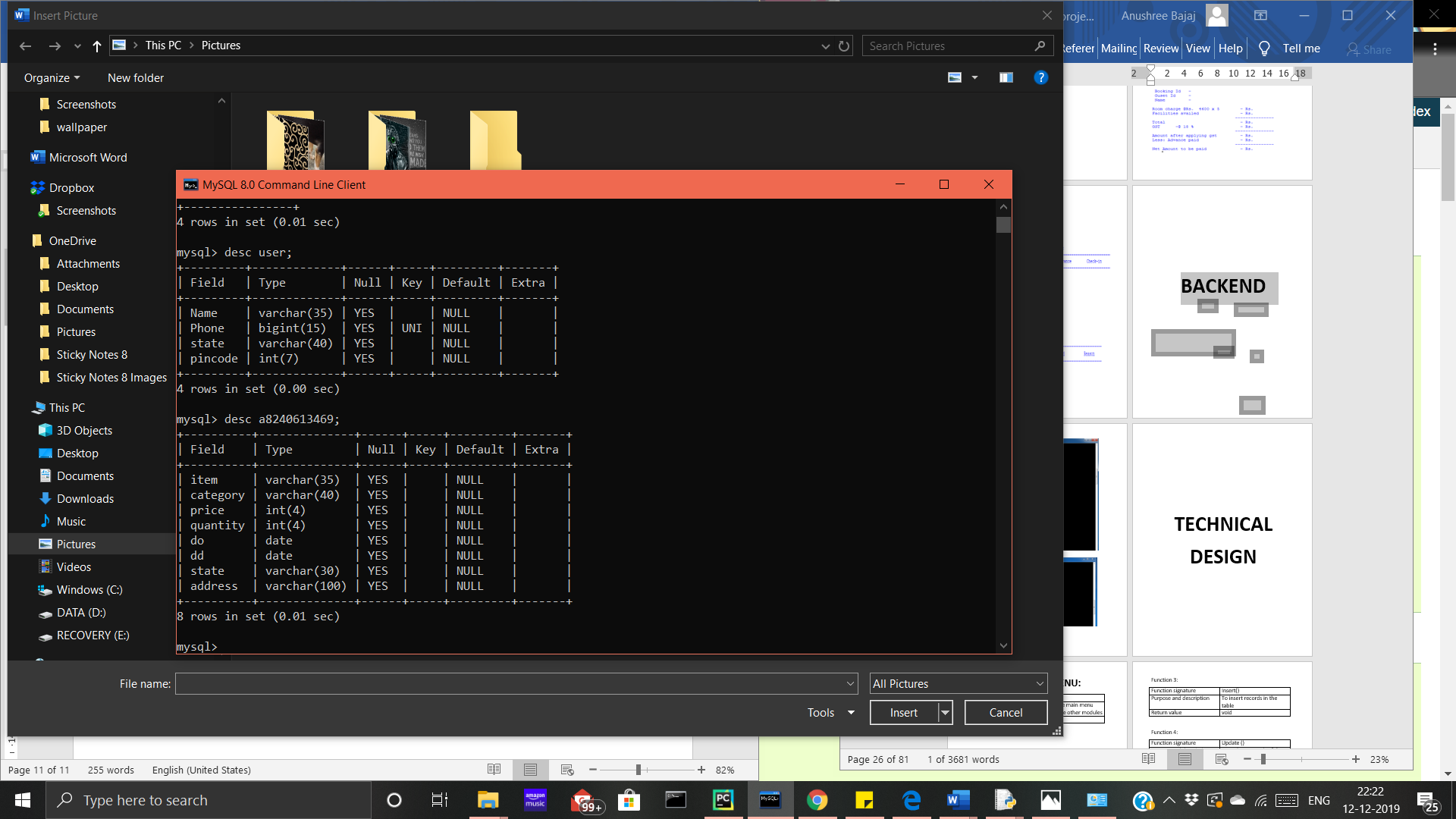


**Module** **6** **:** **View** **past** **orders**



**BACKEND**





**TECHNICAL**

**DESIGN**

**MODULE 1 - HOME:**

Function 1:

|  |  |
| --- | --- |
| Function signature | main |
| Purpose and description | Displays the main menu and calls the other modules |
| Return value | void |

Function 2:

|  |  |
| --- | --- |
| Function signature | ai(c) |
| Purpose and description | Check’s if category and item inputted by user is valid |
| Return value | void |

Function 3:

|  |  |
| --- | --- |
| Function signature | view(q) |
| Purpose and description | Displays categories by which items are divided |
| Return value | void |

**MODULE 2 – USER PROFILE:**

Function 1:

|  |  |
| --- | --- |
| Function signature | usertable() |
| Purpose and description | Creates table of users, if it does not exist |
| Return value | void |

Function 2:

|  |  |
| --- | --- |
| Function signature | User() |
| Purpose and description | Displays details of users and calls other functions in the module |
| Return value | void |

Function 3:

|  |  |
| --- | --- |
| Function signature | insertuser() |
| Purpose and description | Creates profile of new user |
| Return value | void |

Function 4:

|  |  |
| --- | --- |
| Function signature | update() |
| Purpose and description | Updates details of registered user |
| Return value | void |

Function 5:

|  |  |
| --- | --- |
| Function signature | delete() |
| Purpose and description | Deletes user |
| Return value | void |

Function 6:

|  |  |
| --- | --- |
| Function signature | selecteduser() |
| Purpose and description | Selects user |
| Return value | void |

**MODULE 3 – ITEM DETAILS:**

Function 1:

|  |  |
| --- | --- |
| Function signature | view(tkey) |
| Purpose and description | Displays all items and their price of chosen category |
| Return value | void |

Function 2:

|  |  |
| --- | --- |
| Function signature | sort(tkey, order, check) |
| Purpose and description | Sorts the items in order selected by user |
| Return value | void |

Function 3:

|  |  |
| --- | --- |
| Function signature | itemcategory (item, tkey) |
| Purpose and description | Deletes user |
| Return value | void |

Function 4:

|  |  |
| --- | --- |
| Function signature | itemdetail(item, cat) |
| Purpose and description | Displays final price and category of chosen item and category |
| Return value | void |

Function 5:

|  |  |
| --- | --- |
| Function signature | strike(text) |
| Purpose and description | Strikes the original price |
| Return value | string |

**MODULE 4 – CART DETAILS:**

Function 1:

|  |  |
| --- | --- |
| Function signature | add(user, ctgofitem, item) |
| Purpose and description | Adds items in cart |
| Return value | void |

Function 2:

|  |  |
| --- | --- |
| Function signature | delete(item, user) |
| Purpose and description | Deletes item in cart |
| Return value | void |

Function 3:

|  |  |
| --- | --- |
| Function signature | view(user) |
| Purpose and description | Displays items in cart |
| Return value | void |

Function 4:

|  |  |
| --- | --- |
| Function signature | bill(user) |
| Purpose and description | Displays the final bill |
| Return value | void |

Function 5:

|  |  |
| --- | --- |
| Function signature | tableprint(data, user) |
| Purpose and description | Displays the data in tabular form |
| Return value | void |

Function 6:

|  |  |
| --- | --- |
| Function signature | Check(state) |
| Purpose and description | Checks if delivery is available in the desired region |
| Return value | Boolean |

Function 7:

|  |  |
| --- | --- |
| Function signature | rtd(): |
| Purpose and description | Checks if selected user is registered and displays user details |
| Return value | void |

**MODULE 5 – CHECK OUT:**

Function 1:

|  |  |
| --- | --- |
| Function signature | time(state) |
| Purpose and description | Calculates time required for delivery |
| Return value | integer |

Function 2:

|  |  |
| --- | --- |
| Function signature | payment(total, user, address, state) |
| Purpose and description | Confirms payment, adds distance fee and sets delivery date |
| Return value | void |

**MODULE 6 – VIEW PAST ORDERS:**

Function 1:

|  |  |
| --- | --- |
| Function signature | viewpayment(user) |
| Purpose and description | Displays past orders |
| Return value | void |

**MODULE 7 : CREATION OF TABLES**

Function 1:

|  |  |
| --- | --- |
| Function signature | createdb() |
| Purpose and description | Creates database shop if it does not exist |
| Return value | void |

Function 2:

|  |  |
| --- | --- |
| Function signature | createtable (c) |
| Purpose and description | Creates tables containing item details if they don’t exist |
| Return value | void |

Function 3:

|  |  |
| --- | --- |
| Function signature | takedata (c) |
| Purpose and description | Imports details of items from csv files |
| Return value | void |

Function 4:

|  |  |
| --- | --- |
| Function signature | insertvalues () |
| Purpose and description | Inserts item details into tables |
| Return value | void |

**SOURCE CODE**

**MODULE 1: HOME.PY**

import queries\_data2 as qdata

import profile\_data as pdata

import cart\_data as cdata

import check\_out as cout

import Past\_order as porders

pdata.usertable()

def ai(c):

for i in categories:

cword = list(i.split(" "))

checkper = []

check = 0

for j in (list(c.split(" "))):

for k in cword:

inpletter = list(j)

for ctgletter in k:

if ctgletter in inpletter:

check += 1

check = check\*100/len(k)

checkper.append(check)

i = checkper.index(max(checkper))

s = "Did you mean ?", categories[i], "(y/n)"

x = input(s)

if x == "y" or x == "Y":

return i, True

else:

return i, False

def view(q):

key = []

for i in q:

i = i.strip()

i = i.lower()

for j in range(len(categories)):

if categories[j] == i:

key.append(j)

if i not in categories:

print("Category ", i, "does not exist")

j, check = ai(i)

if check:

key.append(j)

return key

categories = ["men's fashion", "women's fashion","kid's fashion"]

# accessories",

print("---------------")

print("ONLINE SHOPPING")

print("---------------")

while True:

print()

print("---------")

print("MAIN MENU")

print("---------")

print()

print("1.Your profile\n2.View items \n3.View cart \n4.Check out\n5.View Orders \n6.Quit ")

choice = input("\t Enter choice (1-6) :")

choice = choice.strip()

if choice == "1":

print()

print("---------")

print("USER DETAILS")

print("---------")

print()

pdata.user()

if choice == "2":

print()

print("------------")

print("ITEM DETAILS")

print("------------")

print()

main = True

while main:

print("Choose category:")

c = 0

for i in categories:

print("\t", i.capitalize(), end=" ")

c = c + 1

print()

choice1 = (input(" Press all to view all items, \* to go back "))

choice1 = choice1.lower()

choice1 = choice1.strip()

if choice1 == "\*":

break

else:

if choice1 == "all":

key = [x for x in range(len(categories))]

qdata.view(key)

else:

choice1 = list(choice1.split(","))

key = view(choice1)

if len(key) != 0:

qdata.view(key)

while len(key) != 0:

print()

print("1.Sort results \n2.Add items in cart \n\* To Go back")

corf = input(" \t Enter your choice (1-2) :")

corf = corf.strip()

if corf == "1":

print("Sort by final price : \n \t1.Low to high \n \t2.High to low \n\t3.Maximum Discount")

c = input("Enter choice (1-3): ")

c = c.strip()

if c == "1":

order = "asc"

qdata.sort(key, order, True)

elif c == "2":

order = "desc"

qdata.sort(key, order, True)

elif c == "3":

order = "desc"

qdata.sort(key, order, False)

else:

print("Wrong choice!")

print()

elif corf == "2":

item = input("Enter item you want : ")

item = item.strip()

item = item.title()

if item == "\*":

pass

else:

itemadded, ctgofi, enter = qdata.itemcategory(item, key)

while enter:

corf2 = input(" 1. Confirm addition to cart \n\t (\* to go back) Enter :")

if corf2 == "1":

if len(pdata.users) == 0:

print("Please register ")

pdata.insertuser()

select = pdata.selecteduser()

cdata.add(select, ctgofi, itemadded)

cp = input("\t 1.View final bill \n\t \* Continue shopping \nEnter:")

if cp == "1":

total,address,state = cdata.bill(select)

finalc = input("\tProceed to pay? (y/n) ")

if finalc == "y":

cout.payment(total,select,address,state)

ff = input("1.Continue shopping \n2. Quit")

if ff == "1":

choice = "3"

elif ff == "2":

choice = "6"

else:

print("Please choose from given options")

key = []

main = False

break

elif finalc == "n":

pass

else:

print("Please choose from given options!")

else:

pass

break

.

elif corf2 == "\*":

break

else:

print("Please choose from given options.")

elif corf == "\*":

break

else:

print("Wrong choice!")

if choice == "3":

print()

print("------------")

print("CART DETAILS")

print("------------")

print()

if len(pdata.users) == 0:

print("No registered Users!")

else:

user = pdata.selecteduser()

cdata.view(user)

while True:

print("\t1.Add item\n\t2.Delete item\n\t3.Proceed to Pay \n\t4.Select another user")

c = input(" (\* to go back) Enter(1-4) : ")

c = c.strip()

if c == "1":

item = input("\tEnter Item :")

item = item.strip()

itemadded, ctgofi, enter = qdata.itemcategory(item)

if enter:

cdata.add(user, ctgofi, itemadded)

elif c == "2":

itemdel = input("\tEnter item : ")

itemdel = itemdel.strip()

itemdel = itemdel.capitalize()

cdata.delete(itemdel, user)

elif c == "3":

if user not in cdata.cart :

print("Cart empty!!")

else:

total, address, state = cdata.bill(user)

cout.payment(total, user, address, state)

ff = input("1.Continue shopping \n2. Quit")

if ff == "1":

choice = "3"

elif ff == "2":

choice = "6"

else:

print("Please choose from given options")

break

elif c == "4":

enteruser = input("Enter phone number :")

enteruser = enteruser.strip()

if int(enteruser) in pdata.users:

user = enteruser

print(user,enteruser)

cdata.view(enteruser)

elif c =="\*":

break

else:

print("Choose from given options")

if choice == "4":

print()

print("---------")

print("CHECK OUT ")

print("---------")

print()

if pdata.users == []:

print("No registered users!\n Cart empty")

else:

user= pdata.selecteduser()

total, address, state = cdata.bill(user)

cout.payment(total, user, address, state)

ff = input("1.Continue shopping \n2. Quit")

if ff == "1":

choice = "3"

print(choice)

elif ff == "2":

choice = "6"

else:

print("Please choose from given options")

if choice == "5":

print()

print("-------------------")

print("PAST ORDERS DETAILS")

print("-------------------")

print()

user = pdata.selecteduser()

porders.viewpayment(user)

while True:

c = input("\n1.View orders by different user \n(\* to continue) Enter : ")

if c == "1":

cdata.rtd()

else:

break

if choice == "6":

print("Thank You")

break

else:

if choice not in ["1","2","3","4","5"]:

print("Wrong Choice")

else:

pass

**MODULE 2: CART DETAILS**

**Cart\_data.py**

import mysql.connector

import csv

import datetime

itemlist = []

ctglist = []

cart = {}

def add(user,ctgofitem,item):

qty = int(input("Enter quantity : "))

if user in cart:

c = 0

for i in cart[user]:

if i[0] == ctgofitem and i[1] == item:

qty = i[2]+1

cart[user].pop(c)

c = c+1

if item in cart[user]:

cart[user].append([ctgofitem,item,qty])

else:

cart[user].append([ctgofitem, item, qty])

else:

cart[user]=[[ctgofitem,item,qty]]

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

print("\nItem Added! Items in "+name+"'s cart are :")

x = cart[user]

for i in x: # x- nested list i- list

for k in range(len(ctg)):

if ctg[k] == i[0]:

cat = categories[k]

print("\t",i[1], "(", cat, ")", i[2])

print()

def view(user):

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

if user in cart:

print("Items in "+name+"'s cart are :")

x = cart[user]

for i in x:

print(i[1]," (",i[2],")")

else:

print("Your cart is empty!")

def delete(item,user):

if len(cart) == 0:

print("Cart empty! No item to delete.")

else:

c=0

for i in cart[user]:

i[1] = i[1].capitalize()

if str(i[1]) == str(item) :

if i[2]>1:

c= i[2]

while c >= i[2]:

c = int(input("Quantity of items you want to reduce: "))

if c == i[2]:

cart[user].pop(c)

break

else:

i[2] = i[2]-c

break

else:

cart[user].pop(c)

break

c = c+1

if cart[user] == []:

cart.pop(user)

break

else:

print("Item not in cart")

view(user)

def bill(user):

if user in cart:

x = cart[user] #nested list

data =[]

for i in x: #list

y=[]

e = "select \* from "+str(i[0])+" where item ='"+str(i[1])+"'"

cur.execute(e)

a = cur.fetchall()

item = a[0][0]

for k in range(len(ctg)):

if ctg[k] == i[0]:

catgry = categories[k]

break

hp = a[0][2]

discount = a[0][3]

qty = i[2]

if discount == 0:

fp = hp\*qty

else:

fp = (hp-hp\*discount/100) \* qty

y = [item,catgry,hp,discount,qty,fp]

data.append(y)

t,a,s = tableprint(data,user)

return t,a,s

else:

print("Cart empty! :'(") ## $-P $$

return None,None,None

def tableprint(data,user):

headers = ["Item", "Category", "Price", "Discount %", "Quantity", "Final Price"]

total = 0

hspaces = [8,13,9,11,9,12]

dspaces = [12,19,9,12,10,12]

output =""

for i in range(len(headers)):

output += "|"+" "\*(hspaces[i]-len(headers[i])) + headers[i] + " "\*(hspaces[i]-len(headers[i]))

output += "\n"+"-"\*(82)

for i in data:

output += "\n"

for j in range(len(i)):

no = (dspaces[j]-len(str(i[j])))//2+1

output +=" "\*no+ str(i[j]) + " "\*no

total += i[5]

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

print("\n User : ",name)

print(output)

state = input("Deliver to : \n\tState : ")

while check(state) == False:

state = input("\tState : ")

city = input("City : ")

pin = input("Pin : ")

t = time(state)

print("\n Distance + handling fee = Rs.", 10 + 5\*t)

total += 10 + 5\*t

print("\n\tTotal payable amount is ",total)

address = state + ","+ city + " "+ pin

return(total,address,state)

def time(state):

with open("distances.csv","r") as file:

data = csv.reader(file)

for row in data:

if row[0] == state:

time = row[1]

break

else:

time = 0

time = float(time)//105+2

if time > 10:

time = time-10

return time

def check(state):

with open("distances.csv","r") as file:

data = csv.reader(file)

for i in data:

if i[0] == state:

break

else:

print("Sorry we do not deliver in entered location!")

return False

def rtd():

e = "select name, phone from user"

cur.execute(e)

a = cur.fetchall()

output = " "\*7 + "Name " + " "\*7+"| Phone \n"

output += "-"\*(35)

for i in a :

output +="\n\t" + i[0]+" "\*((14-len(a[0])))+str(i[1])

print(output)

c = input(" \n Enter phone number :")

viewpayment(c)

categories = ["Men's Fashion", "Women's Fashion","Kid's Fashion"]

ctg = ["mf", "wf","kf"]

db = mysql.connector.connect(user="root", passwd="area51", host="localhost", database="shop")

cur = db.cursor()

final = {}

**MODULE 3 : ITEM DETAILS**

**querie\_data.py**

import mysql.connector

import data1

def strike(text):

s = ""

for c in text:

a = u'\u0336' + c

s += a

return s

def view(tkey):

c = 1

for i in tkey:

print()

print(" Items in", categories[i], " are:")

print()

tname = ctg[i]

e = "select item,price,discount from " + tname

cur.execute(e)

a = (cur.fetchall())

for x in a:

item = x[0].capitalize()

price = x[1]

discount = x[2]

if str(discount) != "0":

op = strike(str(price))

statement = str(item + "Rs." + op + "(-" + str(discount) + "%) Rs.") + str(int(price) + int(price)-int(price) \* int(discount) / 100)

print("\t",item, "Rs.", op, "(-", discount, "%) Rs.", int(price) - int(price) \* int(discount) / 100, " "\*(38-len(str(statement))),end="")

else:

statement = item + ": Rs"+ str(price)

#print(str(statement),len(statement),len(str(statement)))

print( "\t", item, ": Rs", price, " "\*(38 - len(str(statement))), end = " ")

if c%2 == 0:

print()

c=c+1

print()

def sort(tkey, order, check):

for i in tkey:

tname = ctg[i]

print()

print("Item's in ",categories[i],"are:")

print()

if check:

e = "select item,price - price\*discount/100 from " + tname + " order by price-price\*discount/100 " + order

cur.execute(e)

a = (cur.fetchall())

c = 1

for x in a:

print("\t\t", x[0], ": Rs.", x[1], end="")

if c%3 ==0:

print()

c=c+1

print()

else:

e = "select item,price,discount from " + tname + " order by discount " + order

cur.execute(e)

a = (cur.fetchall())

c=1

for x in a:

if x[2] != 0:

statement = str( x[0]+ " "+ str(x[2])+ "%off"+ " :Rs")+str(int(x[1])\*int(x[2])/100)

print("\t\t", x[0], " ", x[2], "%off", " :Rs",int(x[1])- int(x[1])\*int(x[2])/100, " "\*(25-len(statement)), end="")

else:

statement = str(x[0]+ " :Rs")+str(int(x[1]))

print("\t\t", x[0], " :Rs", int(x[1])," "\*(25-len(statement)), end="")

if c%2 == 0:

print()

c = c+1

print()

def itemcategory(item, tkey = [0,1,2]):

category = []

pcategory = []

for i in tkey:

tname = ctg[i]

printname = categories[i]

e = "select \* from " + tname + " where item='" + item + "'"

cur.execute(e)

a = cur.fetchall()

if len(a) != 0:

category.append(tname)

pcategory.append(printname)

if len(category) > 1:

print("Choose category:")

print("\t", pcategory)

cat = input(" \t Enter : ")

cat = cat.strip()

cat = cat.lower()

while cat not in pcategory:

print(cat, pcategory)

print("Item does not exist.")

return None, None, False

else:

for i in range(len(pcategory)):

if cat == pcategory[i]:

cat = category[i]

break

elif len(category) == 0:

print("Item does not exist")

return None, None, False

else:

cat = category[0]

itemdetail(item, cat)

return item, cat, True

def itemdetail(item, cat):

e = "select \* from " + cat + " where item='" + item+"'"

cur.execute(e)

a = cur.fetchall()

for i in a:

for j in range(len(ctg)):

if ctg[j] == i[1][:2]:

j = categories[j]

break

if i[3] != 0:

print(i[0], " (", j, ") ", " - Rs ", i[2]- i[2] \* i[3] / 100)

else:

print(i[0], " (", j, ") ", " - Rs ", i[2])

data1.createdb()

data1.insertvalues()

ctg = ["mf", "wf", "kf"]

db = mysql.connector.connect(user="root", passwd="area51", host="localhost", database="shop")

cur = db.cursor()

categories = ["men's fashion", "women's fashion", "kid's fashion"]

**MODULE 4 : CART DETAILS**

**cart\_data.py**

import mysql.connector

import csv

import datetime

itemlist = []

ctglist = []

cart = {}

def add(user,ctgofitem,item):

qty = int(input("Enter quantity : "))

if user in cart:

c = 0

for i in cart[user]:

if i[0] == ctgofitem and i[1] == item:

qty = i[2]+1

cart[user].pop(c)

c = c+1

if item in cart[user]:

cart[user].append([ctgofitem,item,qty])

else:

cart[user].append([ctgofitem, item, qty])

else:

cart[user]=[[ctgofitem,item,qty]]

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

print("\nItem Added! Items in "+name+"'s cart are :")

x = cart[user]

for i in x: # x- nested list i- list

for k in range(len(ctg)):

if ctg[k] == i[0]:

cat = categories[k]

print("\t",i[1], "(", cat, ")", i[2])

print()

def view(user):

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

if user in cart:

print("Items in "+name+"'s cart are :")

x = cart[user]

for i in x:

print(i[1]," (",i[2],")")

else:

print("Your cart is empty!")

def delete(item,user):

if len(cart) == 0:

print("Cart empty! No item to delete.")

else:

c=0

for i in cart[user]:

i[1] = i[1].capitalize()

if str(i[1]) == str(item) :

if i[2]>1:

c= i[2]

while c >= i[2]:

c = int(input("Quantity of items you want to reduce: "))

if c == i[2]:

cart[user].pop(c)

break

else:

i[2] = i[2]-c

break

else:

cart[user].pop(c)

break

c = c+1

if cart[user] == []:

cart.pop(user)

break

else:

print("Item not in cart")

view(user)

def bill(user):

if user in cart:

x = cart[user] #nested list

data =[]

for i in x: #list

y=[]

e = "select \* from "+str(i[0])+" where item ='"+str(i[1])+"'"

cur.execute(e)

a = cur.fetchall()

item = a[0][0]

for k in range(len(ctg)):

if ctg[k] == i[0]:

catgry = categories[k]

break

hp = a[0][2]

discount = a[0][3]

qty = i[2]

if discount == 0:

fp = hp\*qty

else:

fp = (hp-hp\*discount/100) \* qty

y = [item,catgry,hp,discount,qty,fp]

data.append(y)

t,a,s = tableprint(data,user)

return t,a,s

else:

print("Cart empty! :'(") ## $-P $$

return None,None,None

def tableprint(data,user):

headers = ["Item", "Category", "Price", "Discount %", "Quantity", "Final Price"]

total = 0

hspaces = [8,13,9,11,9,12]

dspaces = [12,19,9,12,10,12]

output =""

for i in range(len(headers)):

output += "|"+" "\*(hspaces[i]-len(headers[i])) + headers[i] + " "\*(hspaces[i]-len(headers[i]))

output += "\n"+"-"\*(82)

for i in data:

output += "\n"

for j in range(len(i)):

no = (dspaces[j]-len(str(i[j])))//2+1

output +=" "\*no+ str(i[j]) + " "\*no

total += i[5]

e = "select name from user where phone ='"+str(user)+"'"

cur.execute(e)

a = cur.fetchall()

name = a[0][0]

print("\n User : ",name)

print(output)

state = input("Deliver to : \n\tState : ")

while check(state) == False:

state = input("\tState : ")

city = input("City : ")

pin = input("Pin : ")

t = time(state)

print("\n Distance + handling fee = Rs.", 10 + 5\*t)

total += 10 + 5\*t

print("\n\tTotal payable amount is ",total)

address = state + ","+ city + " "+ pin

return(total,address,state)

def time(state):

with open("distances.csv","r") as file:

data = csv.reader(file)

for row in data:

if row[0] == state:

time = row[1]

break

else:

time = 0

time = float(time)//105+2

if time > 10:

time = time-10

return time

def check(state):

with open("distances.csv","r") as file:

data = csv.reader(file)

for i in data:

if i[0] == state:

break

else:

print("Sorry we do not deliver in entered location!")

return False

def rtd():

e = "select name, phone from user"

cur.execute(e)

a = cur.fetchall()

output = " "\*7 + "Name " + " "\*7+"| Phone \n"

output += "-"\*(35)

for i in a :

output +="\n\t" + i[0]+" "\*((14-len(a[0])))+str(i[1])

print(output)

c = input(" \n Enter phone number :")

viewpayment(c)

categories = ["Men's Fashion", "Women's Fashion","Kid's Fashion"]

ctg = ["mf", "wf","kf"]

db = mysql.connector.connect(user="root", passwd="area51", host="localhost", database="shop")

cur = db.cursor()

final = {}

**MODULE 5 : CHECK OUT**

**check\_out.py**

import mysql.connector

import datetime

import cart\_data as cdata

import csv

def time(state):

with open("distances.csv","r") as file:

data = csv.reader(file)

for row in data:

if row[0] == state:

time = row[1]

break

else:

time = 0

time = float(time)//105+2

if time > 10:

time = time-10

return time

def payment(total,user,ad,state):

try :

x = cdata.cart[user]

user = str(user)

try :

e = "create table "+"a"+user+"(item varchar(35),category varchar(40),price int(4),quantity int(4),do \

date,dd date,state varchar(30),address varchar(100))"

cur.execute(e)

except mysql.connector.errors.ProgrammingError:

pass

db.commit()

do = datetime.date.today()

t = cdata.time(state)

dd = do + datetime.timedelta(days = t)

for i in x: # list

y = []

e = "select \* from " + str(i[0]) + " where item ='" + str(i[1]) + "'"

cur.execute(e)

a = cur.fetchall()

item = a[0][0]

catgry = a[0][1][:2:]

hp = a[0][2]

discount = a[0][3]

qty = i[2]

if discount == 0:

fp = hp \* qty

else:

fp =( hp - hp \* discount / 100) \* qty + 10 + 5\*t

cur.execute("insert into "+ "a"+user + "(item,category,price,quantity,do,dd,state,address) values \

(%s,%s,%s,%s,%s,%s,%s,%s )",(item,catgry,fp,qty,do,dd,state,str(ad)))

print("\nOrder placed on ", do,"\n\tDelivery by ",dd)

cdata.cart.pop(int(user))

print("\tPayment of Rs.",total," done ! \n\t\tThank you for shopping with us")

db.commit()

except KeyError:

pass

categories = ["Men's Fashion", "Women's Fashion","Kid's Fashion"]

ctg = ["mf", "wf","kf"]

db = mysql.connector.connect(user="root", passwd="area51", host="localhost", database="shop")

cur = db.cursor()

final = {}

**MODULE 6 : VIEW PAST ORDERS**

**Past\_order.py**

import mysql.connector

def viewpayment(user):

try:

e = "select \* from "+ "a"+str(user)

cur.execute(e)

a = cur.fetchall()

if len(a) == 0:

print("No orders!")

else:

data = []

for i in a:

data.append(i)

headers = ["Item", "Category" ," Price ", "Quantity", "Date Of Order" , "Date Of Delivery","Deliver to ","Address of user"]

hspaces = [8, 13, 6, 11, 14, 17,15,20]

dspaces = [15, 20, 10, 10, 19, 20,20,32]

output = ""

for i in range(len(headers)):

output += "|" + " " \* (hspaces[i] - len(headers[i])) + headers[i] + " " \* (hspaces[i] - len(headers[i]))

output += "\n" + "-" \* (130)

for row in data:

row = list(row)

output += "\n"

for j in range(len(row)):

if str(row[j]) in ctg:

for p in range(3):

if ctg[p] == str(row[j]):

row[j] = categories[j]

output += str(row[j]) + (dspaces[j] - len(str(row[j])))\*" "

print(output)

except mysql.connector.errors.ProgrammingError:

print("No orders")

categories = ["Men's Fashion", "Women's Fashion","Kid's Fashion"]

ctg = ["mf", "wf","kf"]

db = mysql.connector.connect(user="root", passwd="area51", host="localhost", database="shop")

cur = db.cursor()

final = {}

**MODULE 7 : CREATION OF TABLES**

**data1.py**

import mysql.connector

import csv

def category():

ctg = ["mf", "wf","kf"]

return ctg

def createdb():

try:

cur.execute("create database shop")

cur.execute("use shop")

except mysql.connector.errors.DatabaseError:

cur.execute("use shop")

def createtable(c):

try:

e = "create table " + c + "(Item varchar(35),code varchar(10) UNIQUE,Price int(10)," \

"discount int(15))"

cur.execute(e)

except mysql.connector.errors.ProgrammingError:

pass

def takedata(c):

d = []

name = c + ".csv"

with open(name, 'r') as file:

data = csv.reader(file)

for row in data:

d.append(row)

item = row[0]

code = row[1]

price = row[2]

discount = row[3]

try:

e = "insert into " + c + "(Item,code,price,discount) " \

"values ('{}','{}',{},{})".format(item, code, price, discount)

cur.execute(e)

db.commit()

except mysql.connector.errors.IntegrityError:

pass

def insertvalues():

ctg = category()

for c in ctg:

createtable(c)

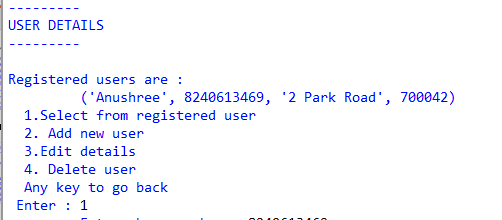
takedata(c)

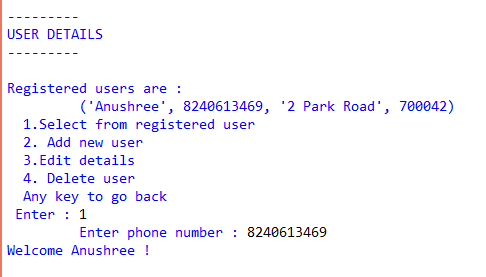
db = mysql.connector.connect(user="root", passwd="area51", host="localhost")

cur = db.cursor()

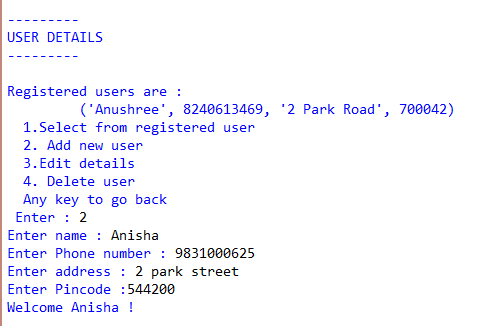
**OUTPUTS**

**Module** **1** **:** **Home**

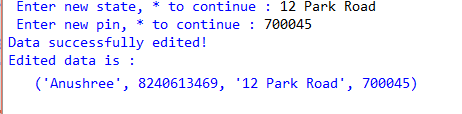
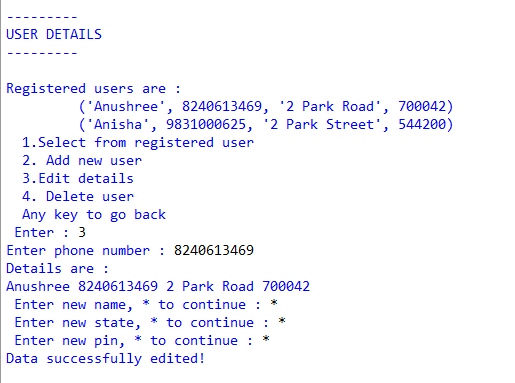
**Module** **2** **:** **User** **Profile**

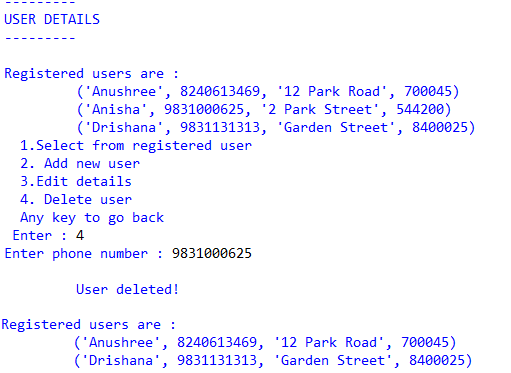
1.Selecting user

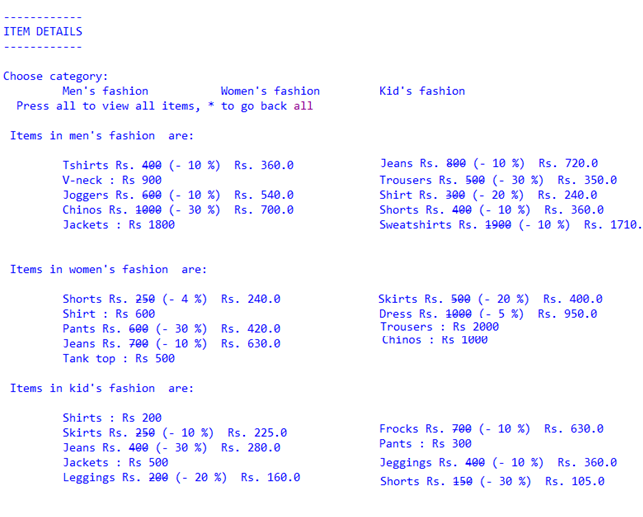
2.Adding new user



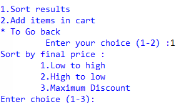
3.Updating user profile



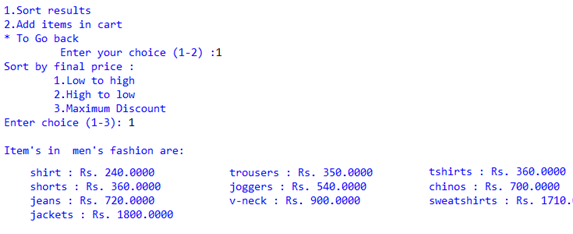
4.Deleting user profile

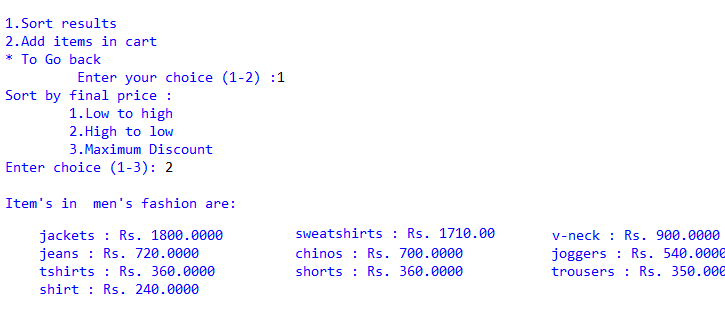
 **Module** **3** **:** **Item** **Details**

1.Displaying Items

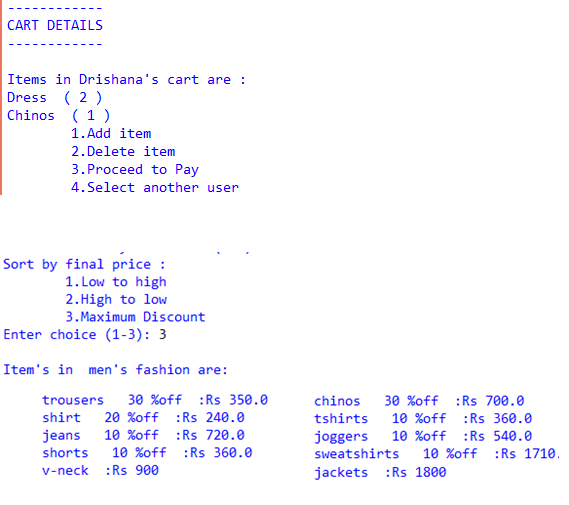
2.Sorting items

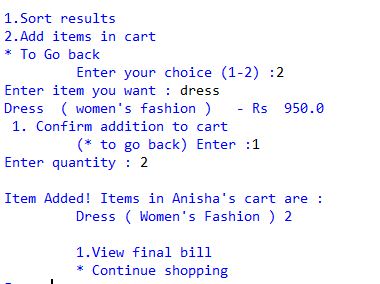
a) Sorting men’s fashion by price low to high

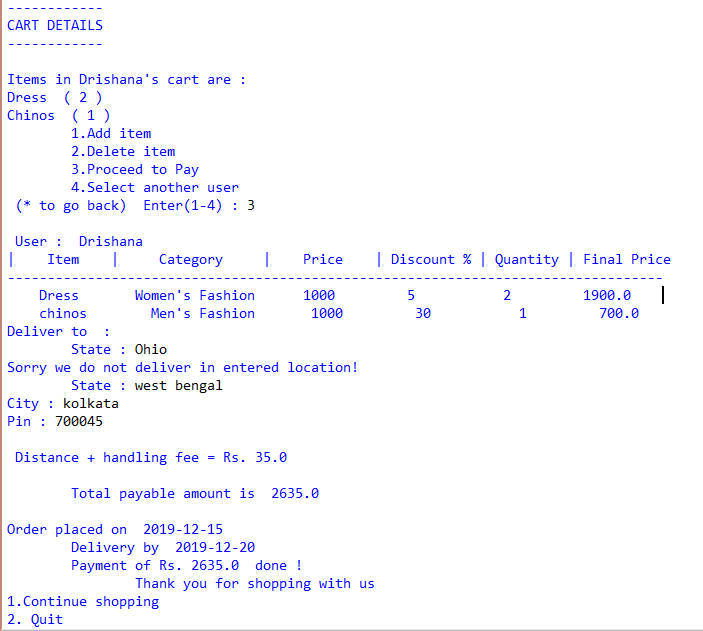


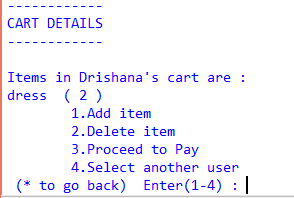
b) Sorting men’s fashion by price high to low

c) Sorting men’s fashion by maximum discount

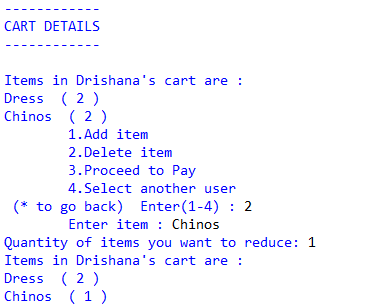


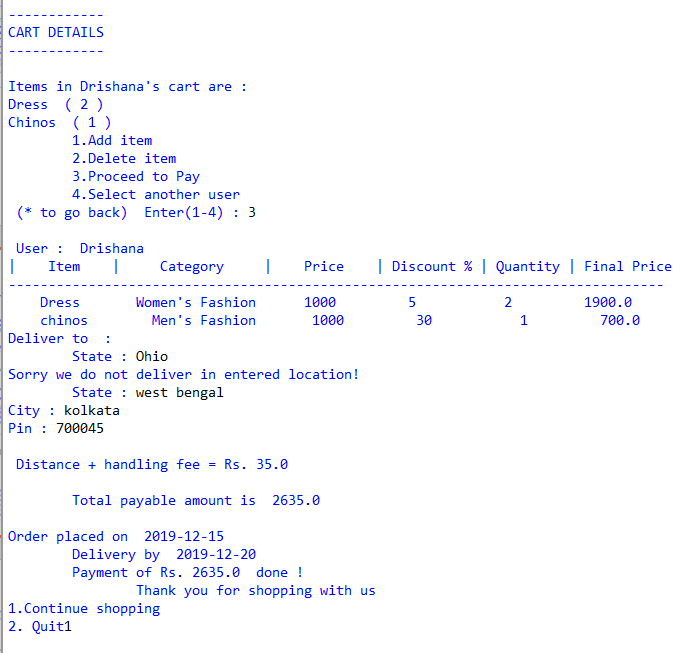
 3.Addition to cart

 4.Check out

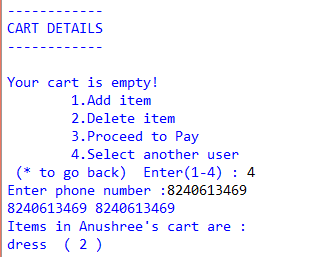
**Module** **4** **:** **Cart** **Details**

1.Addition to cart

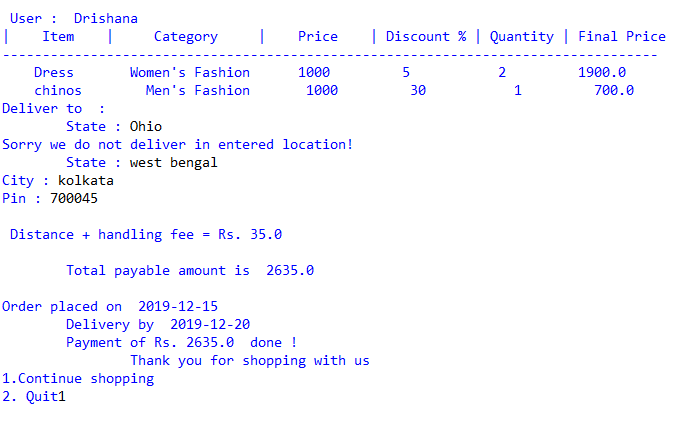
2.Delete from cart

 3. Check out

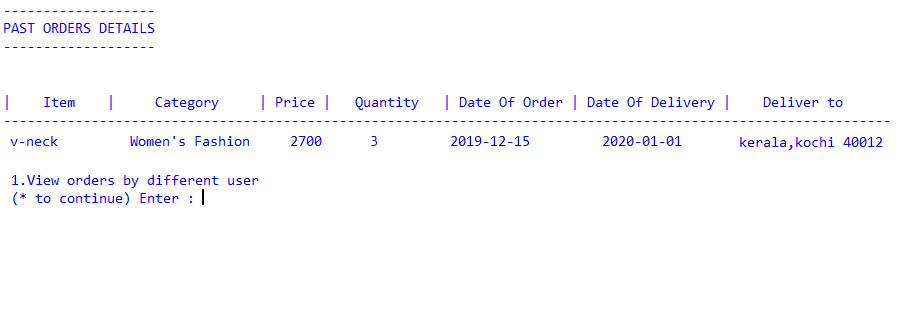
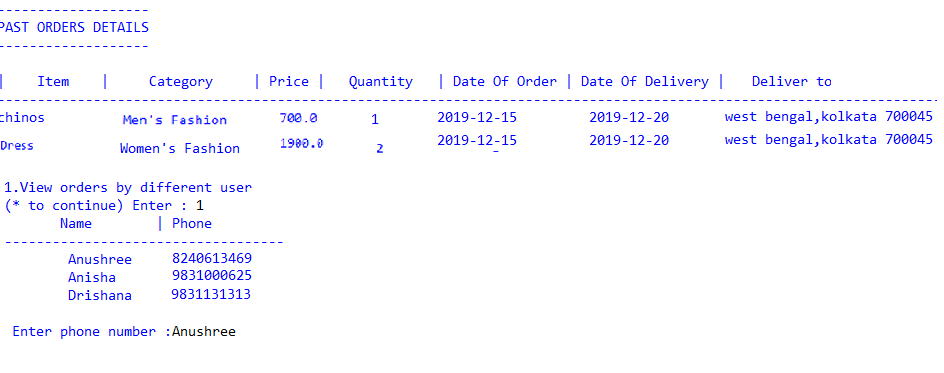
4. Select another user



**Module** **5** **:** **Check** **Out**



**Module** **6** **:** **View** **past** **orders**



**BIBLIOGRAPHY**

The project has been completed successfully with the help of the following sources:

1. Computer Science with Python, Textbook for class XI, Sumita Arora Publisher: Dhanpat Rai & Co.

2. Computer Science with Python, Textbook for class XII, Sumita Arora Publisher: Dhanpat Rai & Co.