**INTRODUCTION**

The main objective of the python project on pizza management is to manage the details of customer, pizza details including editing, adding , updating of records which results in proper resource management of customer data. It manages the information of pizza, it deals with monitoring the information and transactions of pizza it shows the information and description of the customer.

It manages all the information about customer, order status, payments. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the customer details and order status.

THE MAIN MENU has the following options:

* **LOGIN AS ADMIN**
* **LOGIN AS CUSTOMER**
* **EXIT**

Login as admin contains following operations:

1. Add product
2. List product
3. Remove product
4. Find product
5. Modify product
6. Stock entry
7. Stock remain

Login as customer contains following operations:

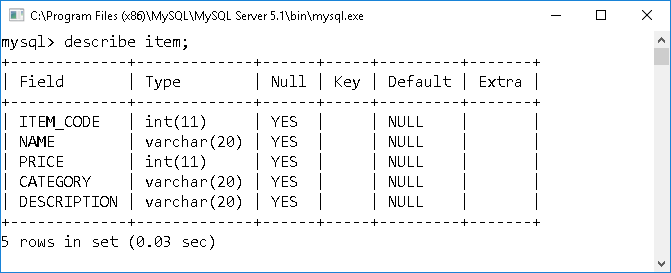
1. Order of Customer
2. Final Bill Amount

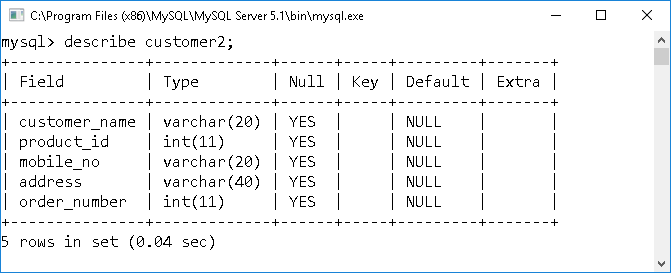
**MODULES IMPORTED**

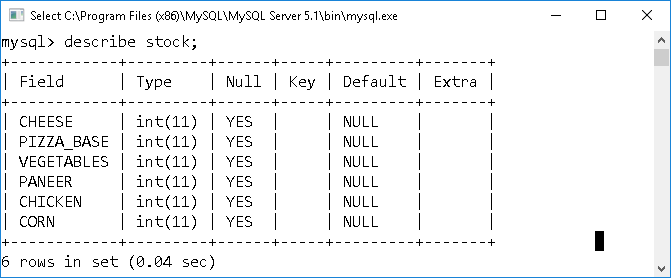
|  |  |  |
| --- | --- | --- |
| S.NO | MODULE NAME | USE |
| 1 | **TKINTER MODULE** | **Tkinter  is the standard GUI library for Python.**  **Tkinter  provides a powerful object oriented interface to the Tk GUI toolkit.** |
| 2 | **MYSQL.CONNECTOR**  **MODULE** | **MySQL Connector allows**  **to compress the data**  **stream between Python and**  **MySQL database server using protocol compression.** |
| 3 | **DATE TIME MODULE** | **The date time module supplies classes for manipulating dates and times. It displays the current date and for an instances.** |

**SCHEMA OF DATABASE**

|  |  |  |
| --- | --- | --- |
| SNO | TABLE | DESCRIPTION |
| 1 | **ITEM** | **It store the information of pizza available in the menu list .the contains price, category, and item code of the following pizza.** |
| 2 | **CUSTOMER** | **It stores the necessary information of customer like order no. , product code, mobile no. , address and customer name.** |
| 3 | **STOCK** | **It gives the current status of items available in stock.** |







**USER DEFINED FUNCTIONS**

|  |  |
| --- | --- |
| FUNCTION | DESCRIPTION |
| def confirm() | **Details are entered by customer to confirm their order.** |
| def Disp() | **Getting the product values to the database and message box of confirmation of order has been displayed.** |
| defmainpage() | **First page of the program has been displayed.** |
| def page2() |  |
| def menu() |  |
| def List Item() |  |
| def find() |  |
| def add() |  |
| def deleteItem() |  |
| def updateItem() |  |
| def entry() |  |
| def remain() |  |
| defpurchaseord() |  |
| defProductManagement() |  |
| def menu1() |  |
| def main1() |  |
| def stockchange() |  |

**SOURCE**

**CODE**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

con=mysql.connector.connect(

host="localhost",

user="root",

password="tiger",

database="system22")

def confirm(m):

m.destroy()

m=Tk()

m.geometry("500x400")

m.configure(background="red")

m.title("CUSTOMER RECORD")

c1=IntVar()

l=Label(m,text="CUSTOMER RECORD")

lbl1=Label(m,text="Enter Customer Name ")

lbl2=Label(m,text="Enter product code ")

lbl3=Label(m,text="Enter Mobile Number ")

lbl4=Label(m,text="Enter Address ")

t1=Entry(m,width=20)

t2=Entry(m,width=20)

t3=Entry(m,width=20)

t4=Entry(m,width=20)

t1.place(x=300,y=50)

t2.place(x=300,y=100)

t3.place(x=300,y=150)

t4.place(x=300,y=200)

l.place(x=180,y=20)

lbl1.place(x=50,y=50)

lbl2.place(x=50,y=100)

lbl3.place(x=50,y=150)

lbl4.place(x=50,y=200)

per=IntVar()

check=Checkbutton(m,text="CONFIRM ORDER",variable=c1)

check.place(x=200,y=250)

cursor1=con.cursor()

sql1='select count(order\_number)from customer2'

cursor1.execute(sql1)

data=cursor1.fetchall()

data1=data[0][0]

def disp(data1):

cursor1=con.cursor()

g1=t1.get()

g2=t2.get()

g3=t3.get()

g4=t4.get()

data2=int(data1)+1

sql="insert into customer2 values('"+g1+"',"+str(g2)+","+str(g3)+",'"+(g4)+"',"+str(data2)+")"

cursor1.execute(sql)

con.commit()

cursor1.execute("select \* from customer2")

record=cursor1.fetchall()

z=c1.get()

if z==1:

messagebox.showinfo("customer data","YOUR ORDER CONFIRMED")

purchaseOrder(g1,g2,g3,g4,m)

else:

messagebox.showinfo("customer data","YOUR ORDER IS NOT CONFIRM")

b=Button(m,text="SUBMIT",width=20,font="Gulim",command=lambda:disp(data1))

b.place(x=150,y=300)

m.mainloop()

def mainpage(r):

r.destroy()

n=Tk()

n.geometry("700x700")

n.configure(background="black")

var=IntVar()

var1=IntVar()

var2=IntVar()

var3=IntVar()

var4=IntVar()

l=Label(n,text="VEGETARIAN PIZZAS",font=("algerian",20),bg="black",fg="white")

l.place(x=200,y=0)

F1=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\pdt1.png")

L1=Label(n,image=F1)

L1.place(x=33,y=120)

F2=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\pdt2.png")

L2=Label(n,image=F2)

L2.place(x=250,y=120)

F3=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\pdt3.png")

L3=Label(n,image=F3)

L3.place(x=467,y=120)

F4=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\Fresh\_Veggie.png")

L4=Label(n,image=F4)

L4.place(x=150,y=355)

F5=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\veggie.png")

L5=Label(n,image=F5)

L5.place(x=360,y=355)

l6=Label(n,text="REMEMBER YOUR PRODUCT CODE",font=("algerian",15),bg="blue",fg="white")

l6.place(x=190,y=70)

but=Button(n,text="SUBMIT",width=20,command=lambda:confirm(n))

but.place(x=280,y=610)

n.mainloop()

def page2(r):

r.destroy()

m=Tk()

m.geometry("700x700")

m.configure(background="black")

var=IntVar()

var1=IntVar()

var2=IntVar()

var3=IntVar()

var4=IntVar()

l=Label(m,text="NON-VEGETARIAN PIZZAS",font=("algerian",20),bg="black",fg="red")

l.place(x=200,y=0)

F1=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\pdt6.png")

L1=Label(m,image=F1)

L1.place(x=33,y=94)

F2=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\dominator.png")

L2=Label(m,image=F2)

L2.place(x=467,y=94)

F3=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\fiesta.png")

L3=Label(m,image=F3)

L3.place(x=250,y=94)

F4=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\supreme.png")

L4=Label(m,image=F4)

L4.place(x=150,y=380)

F5=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\PHOTOS\\bbq.png")

L5=Label(m,image=F5)

L5.place(x=360,y=380)

l6=Label(m,text="REMEMBER YOUR PRODUCT CODE",font=("algerian",15),bg="blue",fg="white")

l6.place(x=200,y=50)

but=Button(m,text="SUBMIT",width=20,command=lambda:confirm(m))

but.place(x=260,y=650)

m.mainloop()

def menu():

r=Tk()

r.geometry("650x500")

pic1=PhotoImage(file="FRESH2.png")

L1= Label(r,image=pic1).pack()

Button(r,text="VEGETARIAN",width=25,height=4,fg="white",bg="green",command=lambda:mainpage(r)).place(x=100,y=410)

Button(r,text="NONVEGETARIAN",width=25,height=4,fg="white",bg="red",command=lambda:page2(r)).place(x=300,y=410)

r.mainloop()

#Item Management

def listItem():

cursor1=con.cursor()

sql="select \* from ITEM"

cursor1.execute(sql)

record=cursor1.fetchall()

cursor1.close()

c=1

for i in record:

print("\t\t\tRecord No :",c)

print("\t\t\tProduct Id :",i[0],"\n\t\t\tName :", i[1],"\n\t\t\tPrice :", i[2],"\n\t\t\tCategory :", i[3],"\n\t\t\tDescription :",i[4])

print("\t","="\*50)

c=c+1

cursor1.close()

def find(d):

cursor1=con.cursor()

sql="select \* from ITEM where ITEM\_CODE="+str(d)

cursor1.execute(sql)

record=cursor1.fetchall()

for i in record:

print("\t\t\tProduct Id :",i[0],"\n\t\t\tName :", i[1],"\n\t\t\tPrice :", i[2],"\n\t\t\tCategory :", i[3],"\n\t\t\tDescription :",i[4])

print("\t","="\*50)

cursor1.close()

def getPrice(d):

cursor1=con.cursor()

sql="select Price from ITEM where ITEM\_CODE="+str(d)

cursor1.execute(sql)

record=cursor1.fetchall()

for i in record:

return i[0]

#print(i,i[0])

cursor1.close()

def add():

cursor1=con.cursor()

cont='Y'

while cont=='Y':

code=input("\t\t\tENTER ITEM CODE :")

name=input("\t\t\tENTER ITEM NAME :")

price=int(input("\t\t\tENTER PRICE :"))

category=input("\t\t\tENTER CATEGORY :")

description=input("\t\t\tENTER DESCRIPTION :")

sql1="insert into item values('"+str(code)+ "','"+str(name)+"','"+str(price)+"', '"+category+"', '"+description+"')"

cursor1.execute(sql1)

con.commit()

print("\t","-"\*50)

cont=input("\t\tDO YOU WANT TO ENTER MORE RECORDS :")

print("\t","-"\*50)

cursor1.close()

def deleteItem():

d=input("\t\tENTER ITEM\_CODE TO BE DELETED :")

find(d)

cursor1=con.cursor()

sql="delete from item where ITEM\_CODE ="+str(d)

cursor1.execute(sql)

con.commit()

print("\t","\*"\*50)

print("\t\t\tDATA DELETED")

print("\t","\*"\*50)

def updateItem():

d=input("\t\t ENTER ITEM\_CODE TO BE MODIFIED :")

print("\t","\*"\*50)

find(d)

cursor1=con.cursor()

print("\t\tEnter New Record details ")

name=input("\t\tENTER ITEM NAME :")

price=int(input("\t\tENTER PRICE :"))

category=input("\t\tENTER CATEGORY :")

description=input("\t\tENTER DESCRIPTION :")

sql="update item set price="+str(price)+", name='"+str(name)+"',category='"+category+"',description='"+description+"' where ITEM\_CODE="+str(d)

cursor1.execute(sql)

con.commit()

cursor1.close()

print("\t","\*"\*50)

print("\t\t\tDATA UPDATED")

print("\t","\*"\*50)

def entry():

cursor1=con.cursor()

print("\t","-"\*50)

print("\t\t\tSTOCK ENTRY")

print("\t","-"\*50)

cheese=int(input("\t Enter the quantity of cheeses(in kg) :"))

pizza\_base=int(input("\t Enter the number of pizza base :"))

vegetables=int(input("\t Enter the quantity of vegetables(in kg):"))

paneer=int(input("\t Enter the amt of paneer :"))

chicken=int(input("\t Enter the amt of chicken :"))

corn=int(input("\t Enter the amt of corn :"))

sql1="insert into stock values('"+str(cheese)+ "','"+str(pizza\_base)+"','"+str(vegetables)+"','"+str(paneer)+"','"+str(chicken)+"','"+str(corn)+"')"

cursor1.execute(sql1)

print("\t","-"\*50)

con.commit()

cursor1.close()

def remain():

cursor1=con.cursor()

sql="select \* from stock"

cursor1.execute(sql)

record=cursor1.fetchall()

print("\t","\*"\*50)

print("\t CHEESE PIZZA\_BASE VEGETABLE PANEER CHICKEN CORN")

for i in record:

print("\t ",i[0]," ",i[1]," ",i[2]," ",i[3]," ",i[4]," ",i[5])

print("\t","\*"\*50)

con.commit()

cursor1.close()

#Customer Management

def findCustomer(d):

cursor1=con.cursor()

sql="select \* from customer where mob\_no="+str(d)

cursor1.execute(sql)

record=cursor1.fetchall()

for i in record:

print("Customer Name ",i[1],"\nMobile no ", i[0], "\nAddress ",i[2])

print("="\*10)

return True

return False

cursor1.close()

#Puchase Order

def purchaseOrder(name,product,number,address,m):

m.destroy()

cursor1=con.cursor()

cont='y'

#number=int(input("\t\t\tMOBILE NUMBER :"))

#cursor1=con.cursor()

sql1='select count(order\_number) from customer2'

cursor1.execute(sql1)

data=cursor1.fetchall()

data1=data[0][0]

order\_number=int(data1)

print("\t\t\tYOUR ORDER NO.:",order\_number)

l=[]

cur1=con.cursor()

#product=int(input("Enter the product\_code:"))

cursor1=con.cursor()

sql="select price from item where ITEM\_CODE="+str(product)

cursor1.execute(sql)

amt=cursor1.fetchall()

gst=0.12\*amt[0][0]+amt[0][0]

total=gst+amt[0][0]

print("\t","\*"\*50)

print("\t\t\t BILL")

print("\t","\*"\*50)

print("\t\tORDER\_NO :",order\_number)

print("\t\tPRODUCT\_CODE :",product)

print("\t\tCUSTOMER\_NAME :",name)

print("\t\tCUSTOMER\_MOBILENO.:",number)

#print("\t\tGST(%) : 10%")

print("\t","\*"\*50)

print("\t\tFINAL BILL AMOUNT :",total,"RS/-")

print("\t","\*"\*50)

stockchange(product)

#cursor1.close()

def productManagement():

while True:

print("\t\t\t1. Add Product ")

print("\t\t\t2. List Product")

print("\t\t\t3. Remove Product")

print("\t\t\t4. Find ")

print("\t\t\t5. Modify Record ")

print("\t\t\t6. Stock Entry ")

print("\t\t\t7. Stock Remain")

print("\t\t\t8. Back")

print("\t","\*"\*50)

ch=int(input("\t\t Enter Your Choice: "))

print("\t","-"\*50)

if(ch==8):

break

elif ch==1:

add()

elif ch==2:

listItem()

elif ch==3:

deleteItem()

elif ch==4:

d=input("\t\t\tENTER ITEM\_CODE :")

find(d)

elif ch==5:

updateItem()

elif ch==6:

entry()

elif ch==7:

remain()

def menu1():

r=Tk()

#r.destroy()

r.geometry("650x500")

pic1=PhotoImage(file="\\\srlab\\tcdata\\system19\\Documents\\Himani XII C\\himani 12thc\\FINAL.png")

L1= Label(r,image=pic1).pack()

Button(r,text="CLICK HERE TO PROCEED",width=25,height=2,fg="white",bg="red").place(x=0,y=450)

r.mainloop()

menu1()

def main1():

while True:

print("\t","\*"\*50)

print("\t\t WELCOME TO PIZZA CASTLE")

print("\t","\*"\*50)

print("\t\t\t1.LOGIN AS ADMIN ")

print("\t\t\t2.LOGIN AS CUSTOMER ")

print("\t\t\t0.Exit ")

print("\t","\*"\*50)

ch=int(input("\t\t Enter Your Choice: "))

print("\t","\*"\*50)

if ch==1:

productManagement()

elif ch==2:

menu()

break

elif ch==0:

break

def stockchange(product):

cursor1=con.cursor()

if int(product)==101:

s="update stock set cheese=cheese-1"

cursor1.execute(s)

s1="update stock set pizza\_base=pizza\_base-1"

cursor1.execute(s1)

s2="update stock set vegetables=vegetables-1"

cursor1.execute(s2)

con.commit()

elif int(product)==102:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set corn=corn-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==103:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set paneer=paneer-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==104:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

con.commit()

elif int(product)==105:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set corn=corn-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==201:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set chicken=chicken-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==202:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set chicken=chicken-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==203:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set chicken=chicken-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==204:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set vegetables=vegetables-1"

s3="update stock set chicken=chicken-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

cursor1.execute(s3)

con.commit()

elif int(product)==205:

s="update stock set cheese=cheese-1"

s1="update stock set pizza\_base=pizza\_base-1"

s2="update stock set

vegetables=vegetables-1"

s3="update stock set chicken=chicken-1"

cursor1.execute(s)

cursor1.execute(s1)

cursor1.execute(s2)

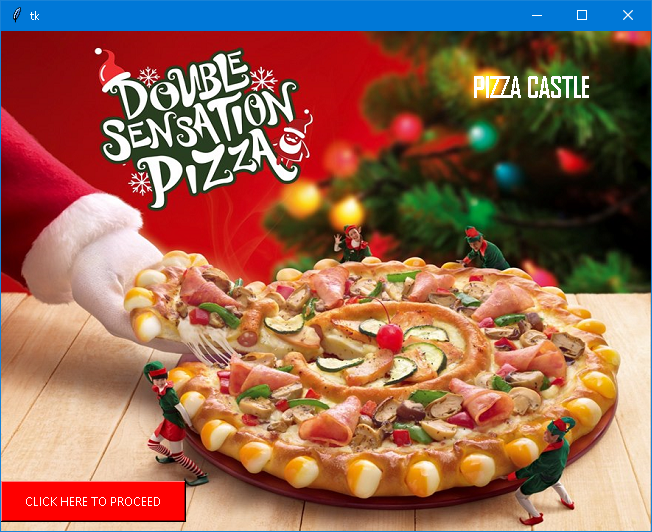
cursor1.execute(s3)

con.commit()

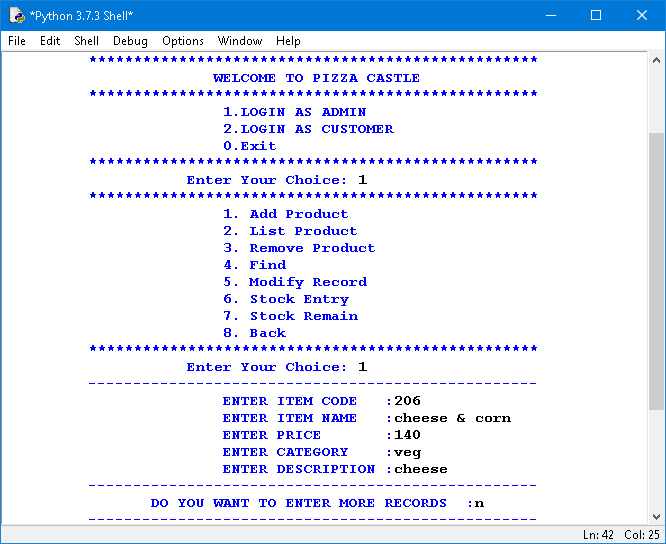
main1()

**SCREEENSHOTS**

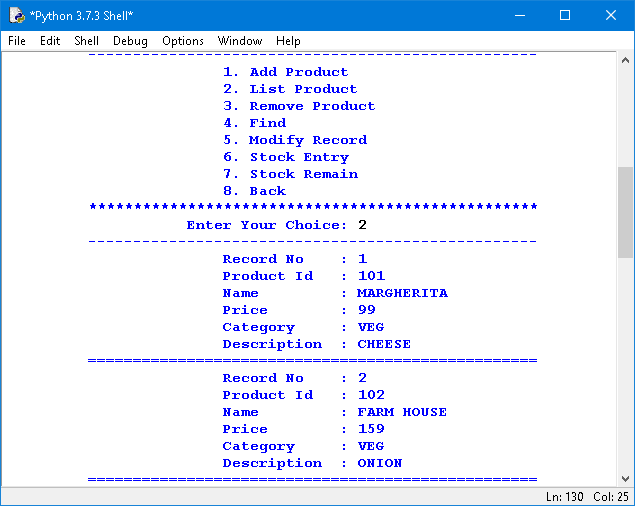
**HOME PAGE**

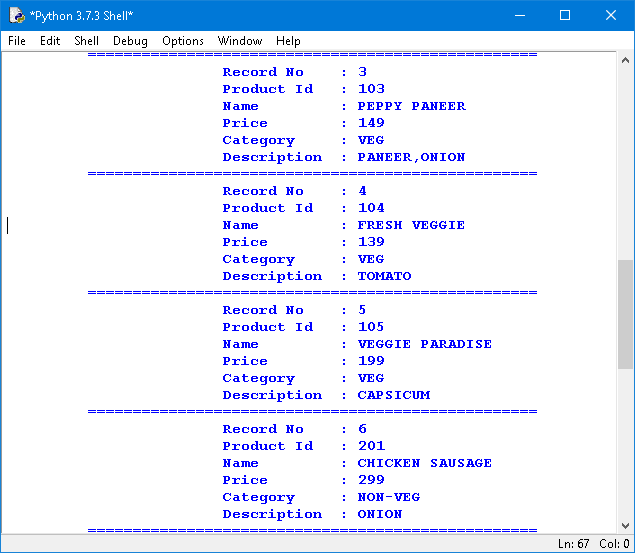
****

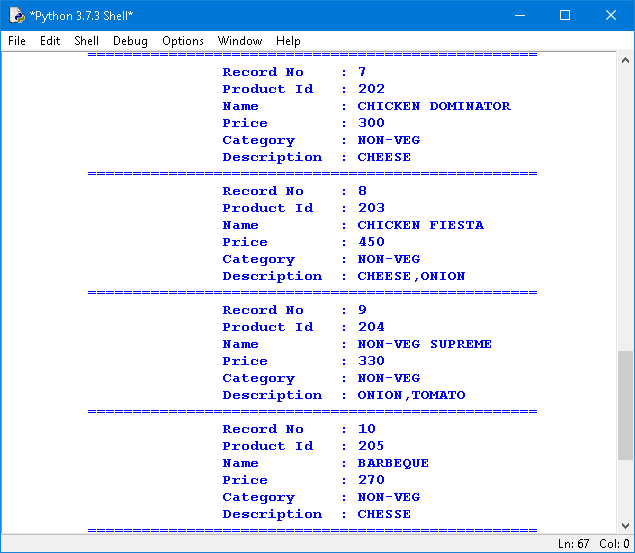
**LOGIN AS ADMINISTRATOR - ADDING NEW PIZZA**

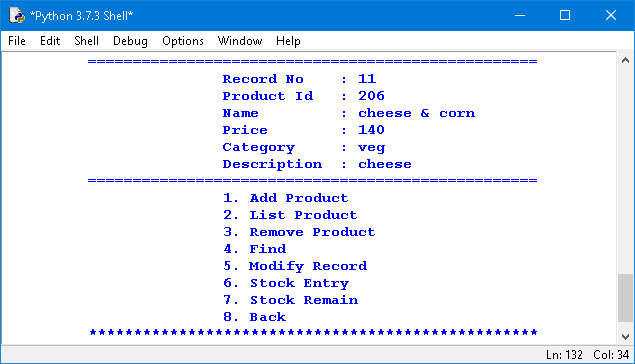
****

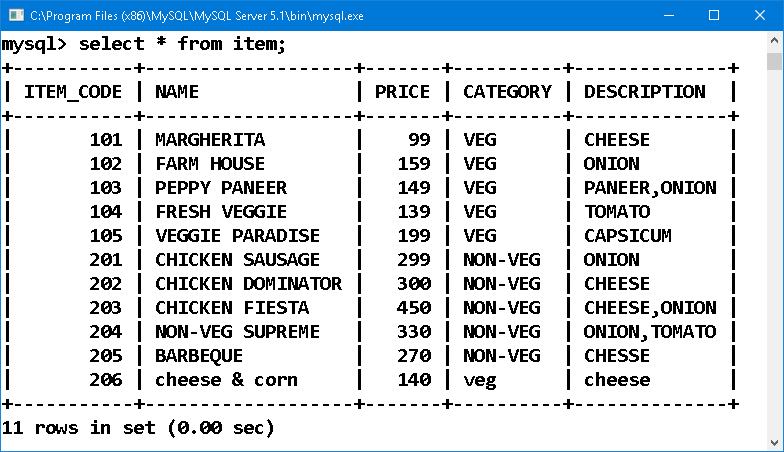
**LOGIN AS ADMINISTRATOR –LISTING PIZZA**

****

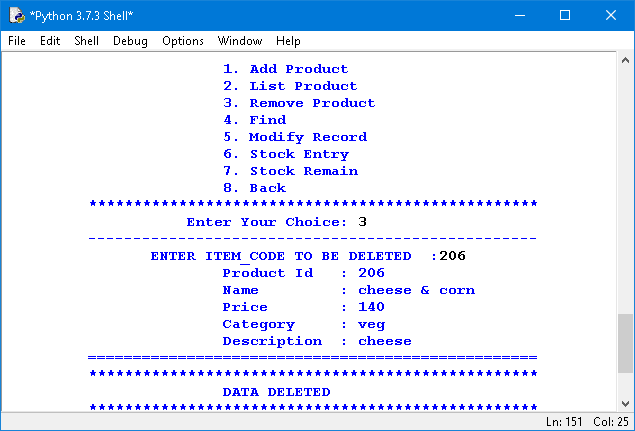
****

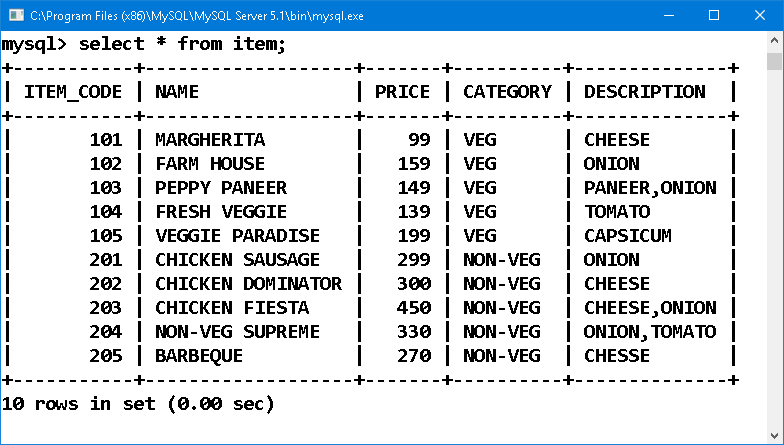
****

****

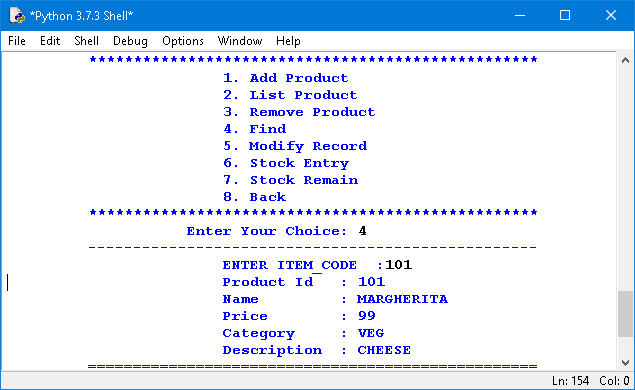
****

**LOGIN AS ADMINISTRATOR-DELETION OF PIZZA**

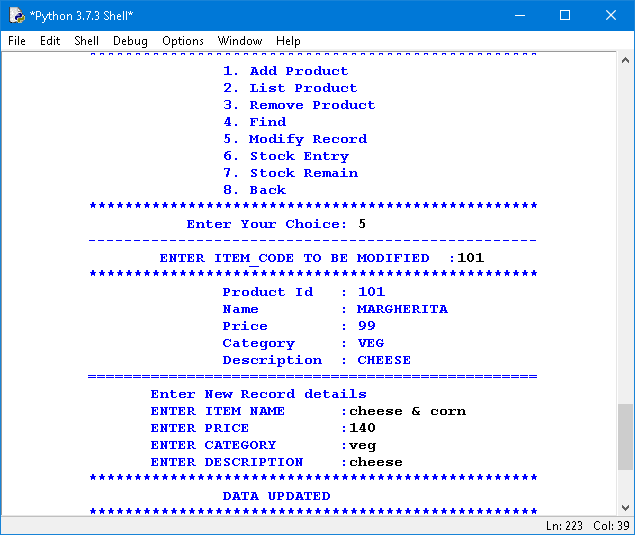
****

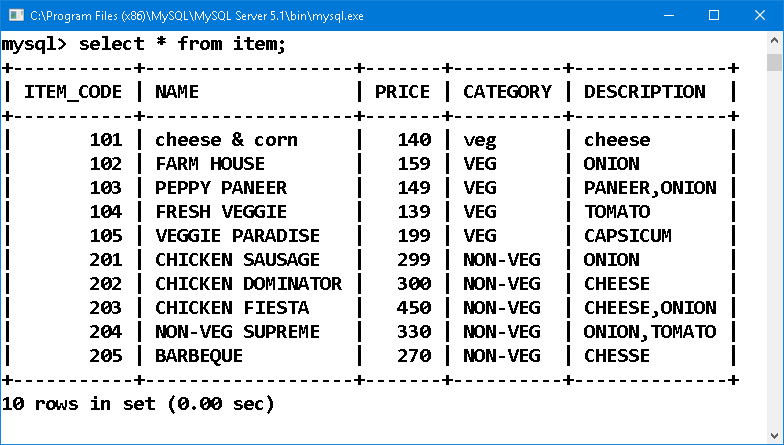
****

**LOGIN AS ADMINISTRATOR-FINDIND PIZZA**

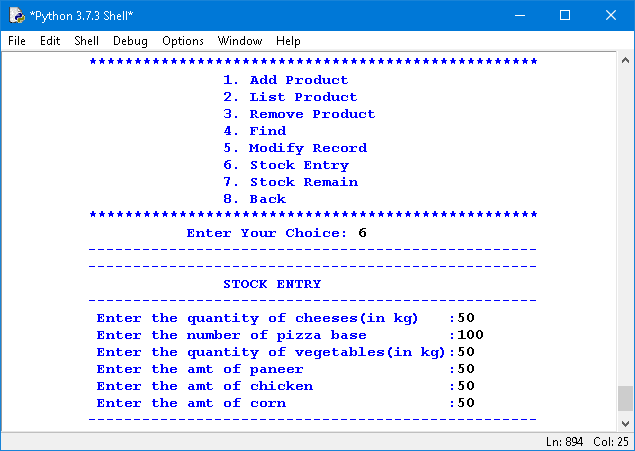
****

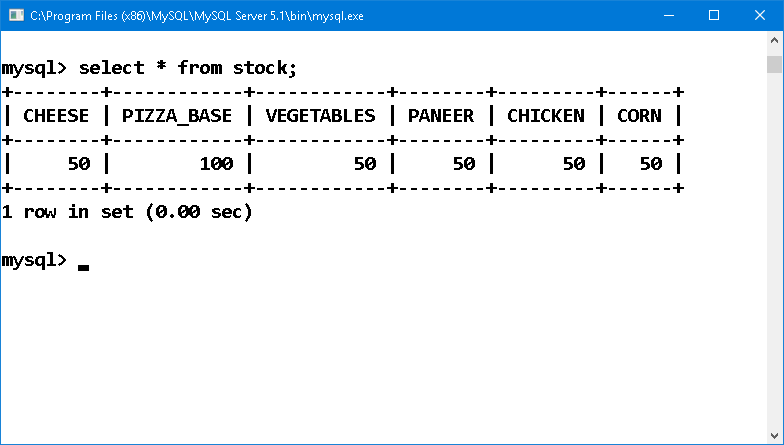
**LOGIN AS ADMINISTRATOR-UPDATING PIZZA**

****

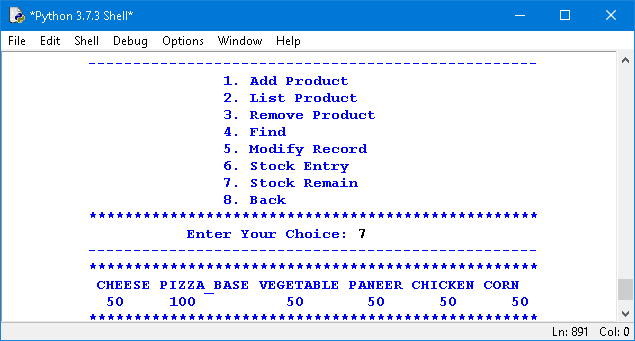
****

**LOGIN AS ADMINISTRATOR –ENTERING STOCK DATA**

****

****

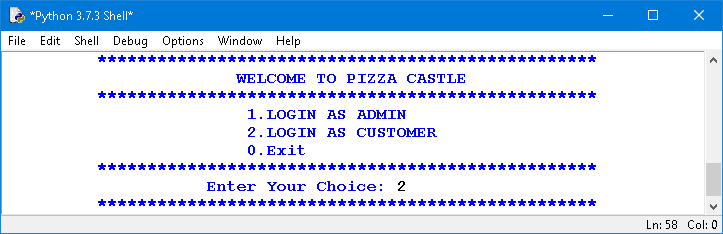
**LOGIN AS ADMINISTRATOR –STOCK REMAINING**

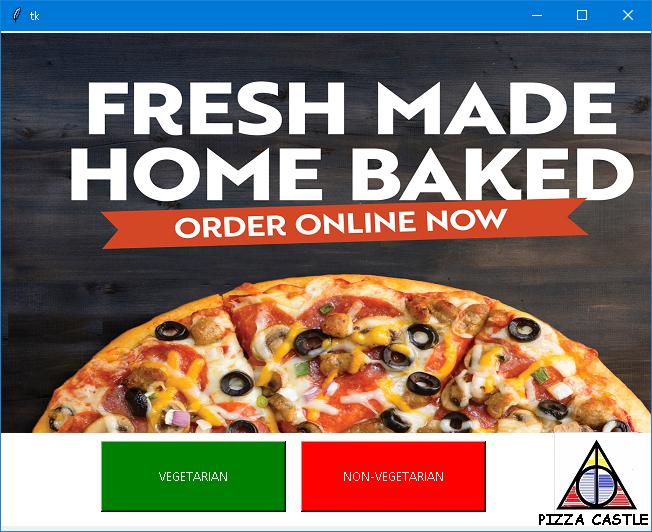
****

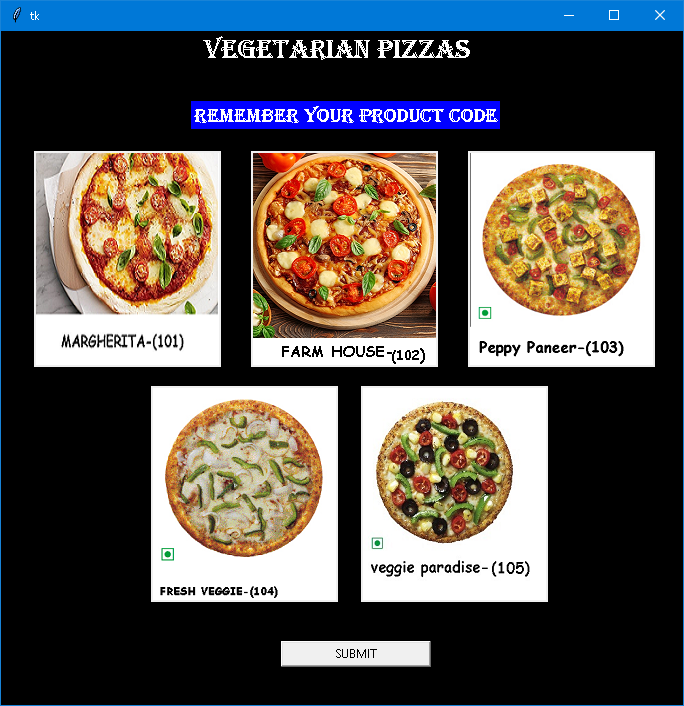
**LOGIN AS ADMINISTRATOR –BACK**

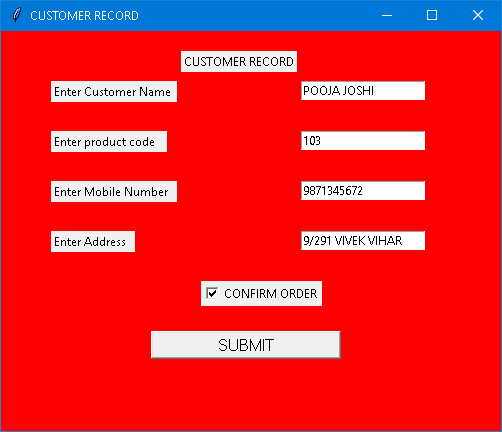
****

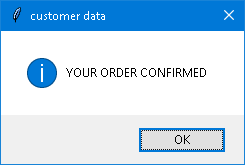
**LOGIN AS CUSTOMER –PURCHASING ORDER**

****

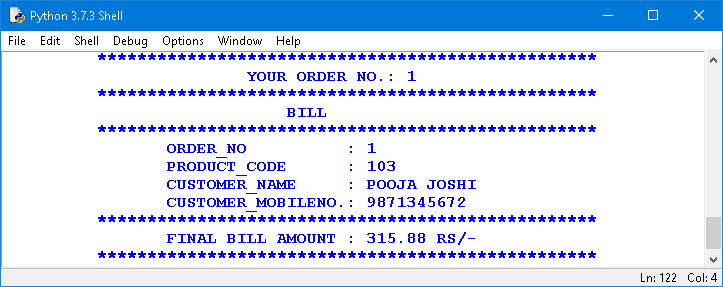
****

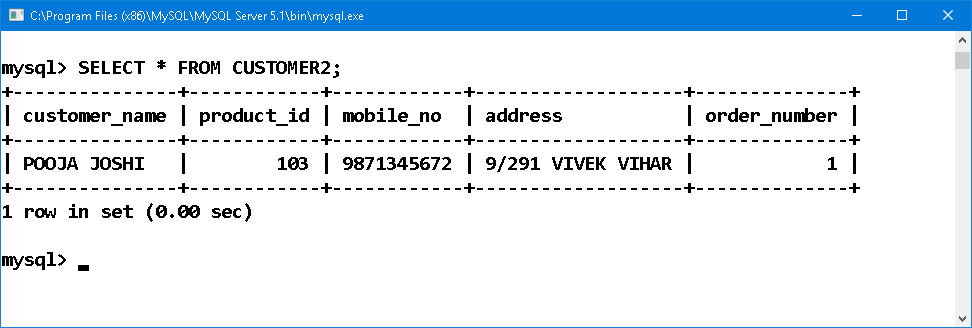
****

****

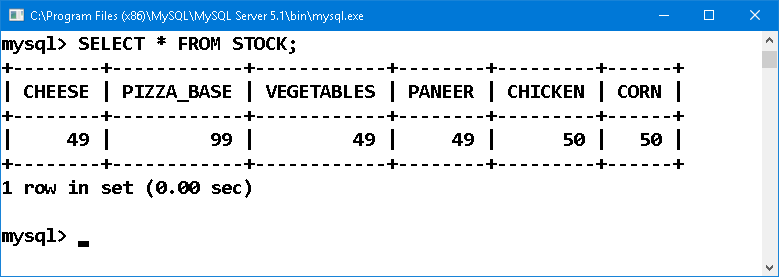
****

BILL OF CUSTOMER

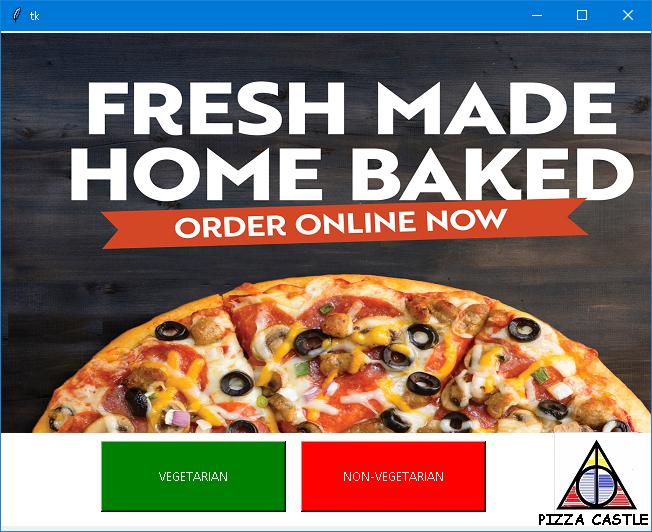
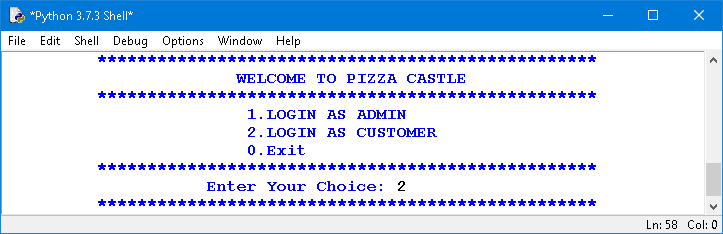


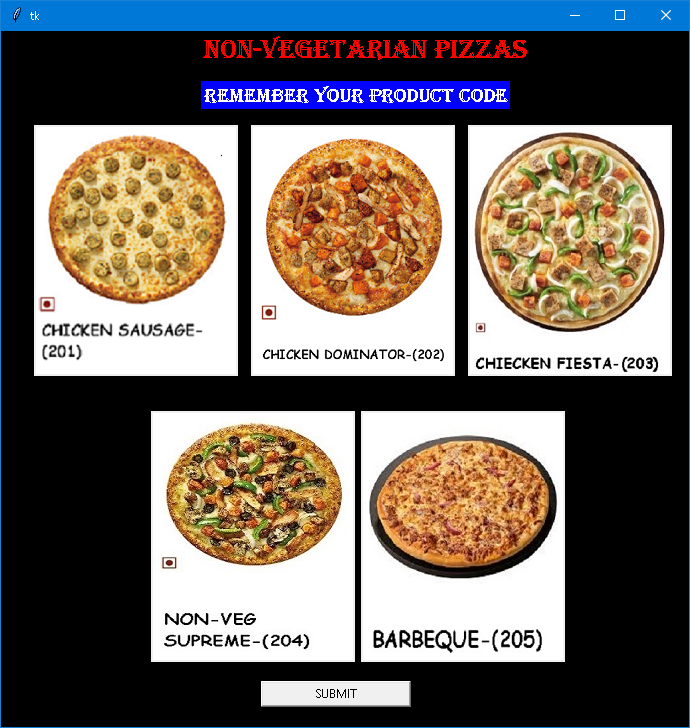


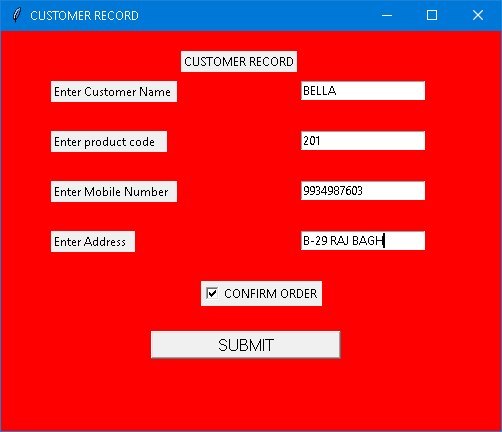
STOCK REMAINING

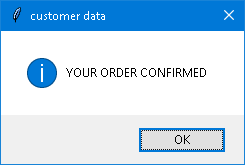


**LOGIN AS CUSTOMER – PURCHAISING ORDER**

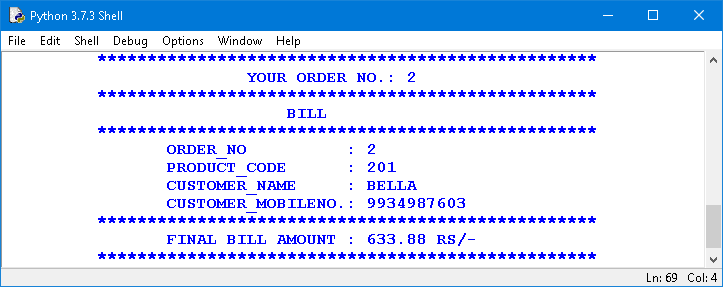
****

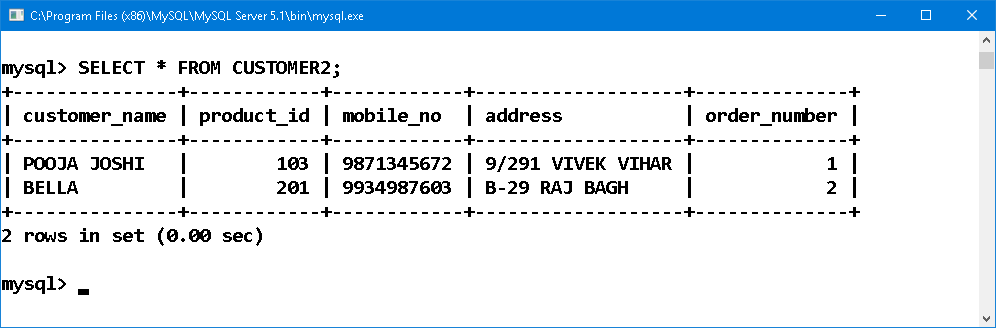
****

****

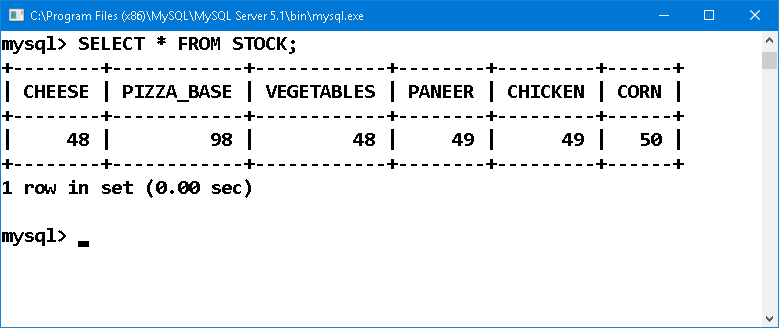
****

**BILL OF CUSTOMER**

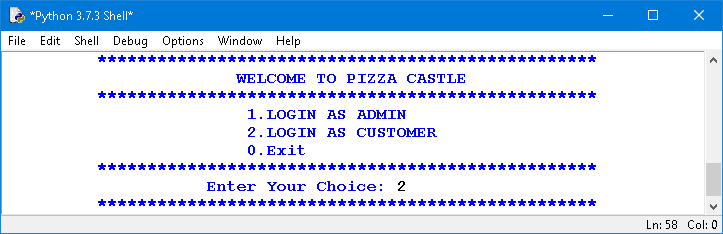
****

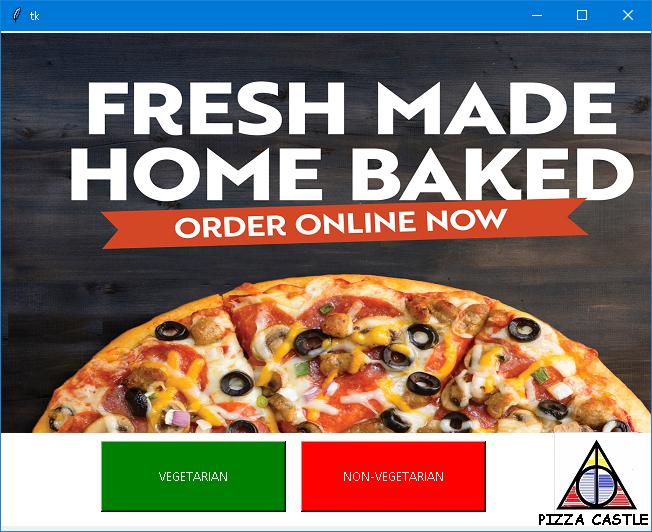
****

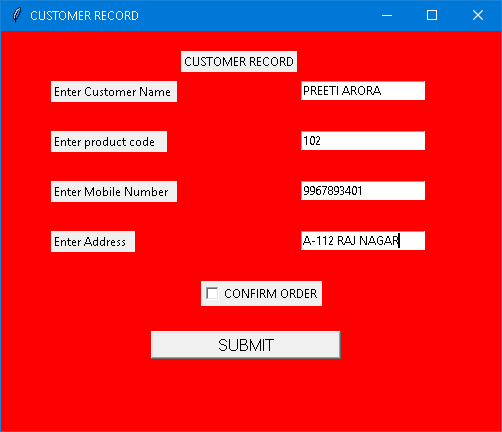
**STOCK REMAINING**

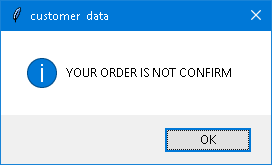
****

**LOGIN AS ADMINISTRATOR-PURCHAISING ORDER**

****

****

****

****

**BIBLOGRAPHY**

* **SUMITA ARORA**
* **PREETI ARORA**
* **WWW.PIZZAHUT.COM**