DELHI PUBLIC SCHOOL HARNI

ACADEMIC SESSION 2021-22

Café Management System

Name of the Student: Vaibhav Shekar

Class: XII - A

Roll No:

CERTIFICATE

This is to certify that Vaibhav Shekar of class XII has successfully completed his/her project titled **"Café Management System"** for class XII AISSCE practical Examination of CBSE for the year 2021-2022. The aforesaid Project Work has been submitted to the Informatics Practices Department of Delhi Public School Harni

Internal Examiner **External Examiner**

School Stamp

ACKNOWLEDGEMENT

Acknowledgement is the most beautiful page in any Project's starting pages. More than a convention, it always

I would like to sincerely thank ***Mrs. Preeti Jha***, our Informatics Practices Teacher for providing me valuable information and guiding me with the relevant topics which facilitated smooth completion of my project. It was her priceless guidance, constant encouragement, constructive comments, sympathetic approach and immense motivation, which has sustained my efforts during all stages of this Project. I would also thank my ***Parents*** for being a source of encouragement and providing time and freedom to develop this Software Project.

DECLARATION

This is to declare that to the best of my knowledge this project is a bona fide work of Vaibhav Shekar.

He has worked sincerely on the project titled

"**Café Management System**"

under my supervision and guidance in the school Computer Lab and otherwise.

I certify that the aforementioned work is authentic work of Vaibhav Shekar.

Internal Examiner

INDEX

1. Aim and Introduction 1

2. Certificate 2

3. Acknowledgements 3

4. Declaration 4

5. Purpose Of The Project 7

6. System Implementations 7

7. Software Coding 9

   a. Start Up Page 9

   b. Login window (E:/Cafe Management/cafe with login.py) 10

      i. Valid Authentication 11

      ii. Invalid Authentication 13

INDEX

c. Billing Window('E:\Cafe Management\CAFE FINAL 1.py') 15

d. Search Bill Window('E:\Cafe Management\sf1.py') - 45

8. Known Limitations 55

9. Bibliography 55

Purpose Of the Project

The purpose of this Software is to develop a Café Management System in order to automate and ease the billing process of a Café.

A Café Management mainly consist of an electronic database, which is a collection of logically-related various files(CSV) and tables(MySQL) for various purposes such as verification of the billing employee, taking customer's order, saving a customer's bill, searching for a bill and viewing it, etc. Here, Python-Tkinter based Programming Interface connected to the Relational Database Management Software (RDBMS) and CSV files facilitates easy access to everything. Using the Interactive Graphical User Interface (GUI), we can login to the program, take customer orders, generate bills, save the bills, and search the bills on the basis of customer's name and contact number.

The proposed Software System is expected to perform the following tasks:

- **To provide a user-friendly and a GUI-based integrated environment for Café's Billing operations.**
- **To maintain all the past bill records, take an order, and search for a bill record.**
- **To ensure employee authenticity before the main billing window opens.**

System Implementations

The proposed Software Project has successfully been developed using the following software:

- **Python 3.10.1**
- **Tkinter – Python interface**
- **Server version: 5.1.33-community MySQL Community Server (GPL)**
- **Microsoft Excel 2019**

Following are the operating systems with respective hardware configuration on which the aforesaid Project has been mutually developed and tested with no abnormalities:

- **Microsoft Windows 10 Pro 64-bit operating system, x64-based processor over Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.19 GHz, 7.91 GB of RAM, and graphic processor Intel(R) HD Graphics 5500**
- **Microsoft Windows 10 Home Single Language 64-bit operating system, x64-based processor over Intel(R) Core(TM) i5-1034G1 CPU @ 1.00GHz 1.19 GHz, 7.75 GB of RAM, and graphic processor Intel(R) UHD Graphics 5500.**

This Document File was created using: Microsoft Office Word 2019 (Home and Student Edition 2019)

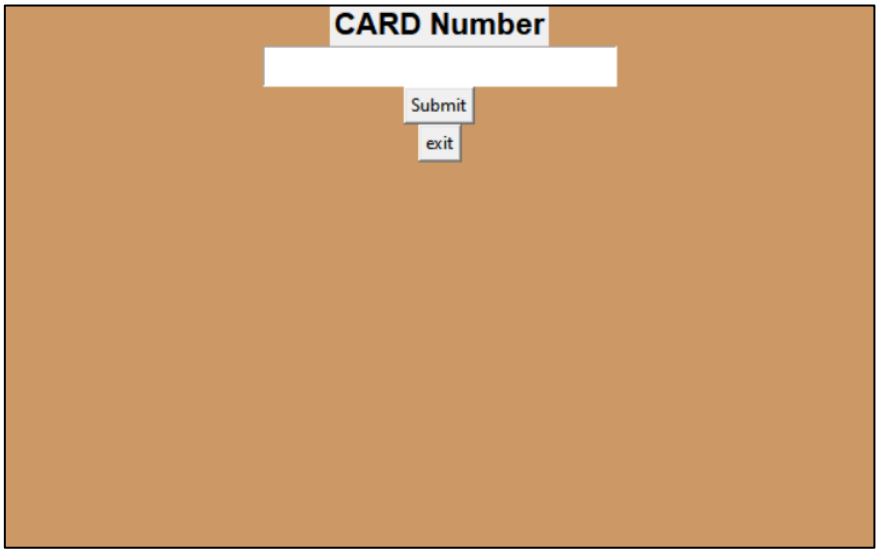
Software Coding

(a) Start Up Page

'E:\Cafe Management\logo.png'

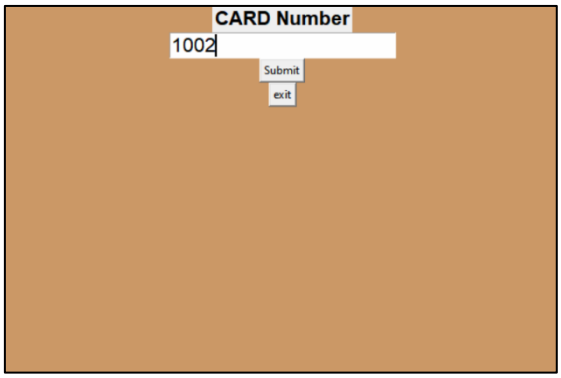
```
from tkinter import *

startup_root = Tk()

startup_root.attributes('-fullscreen', True)

img = PhotoImage(file='E:\Cafe Management\logo.png')

stlbl = Label(startup_root, image = img).pack()

def main():

   startup_root.destroy()

   execfile('E:/Cafe Management/cafe with login.py') #pg-10

startup_root.after(4000,main)

mainloop()
```

(b) Login window (E:/Cafe Management/cafe with login.py)




```
from tkinter import *

from datetime import datetime

import tkinter as tk

import mysql.connector as contr

import pandas as pd

import csv

r1=tk.Tk()

r1.geometry("600x400")

r1.configure(bg='#CC9966')

#r1.attributes('-fullscreen', True)

card_var=StringVar()
```

(b.1) Login window – Valid Authentication



VALID AUTHENTICATION

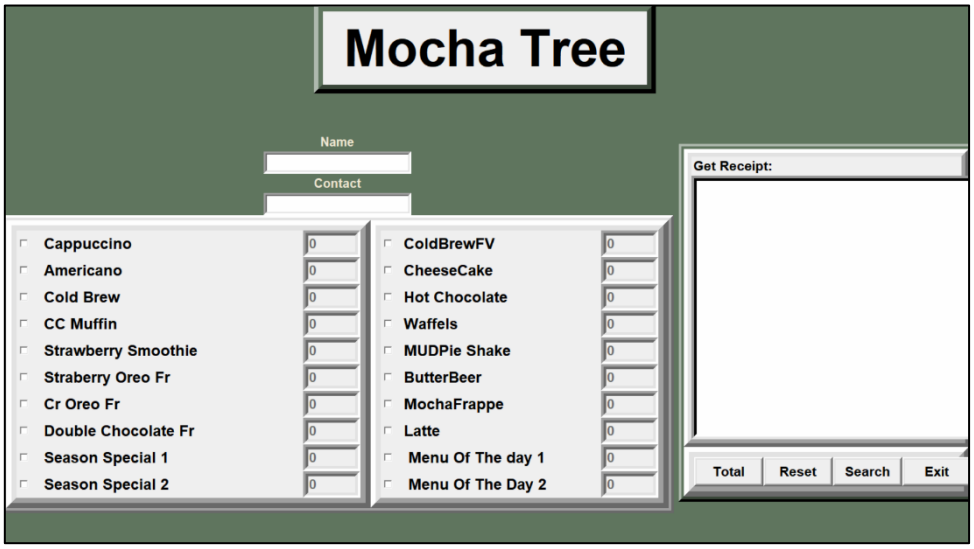
```
def submit():

   cardno=card_var.get()

   timestamp = str(datetime.now())

   data = [cardno,timestamp]

df1 = pd.read_csv('E:\Cafe Management\saved cards.csv',usecols
= ['card number'])


#/////////checking if the entered card no is valid/////////
```

card number	names associated
1001	Rohan
1002	Vipul
1003	Roshan
1004	Mohan
1005	Vanshika
1006	Ritu

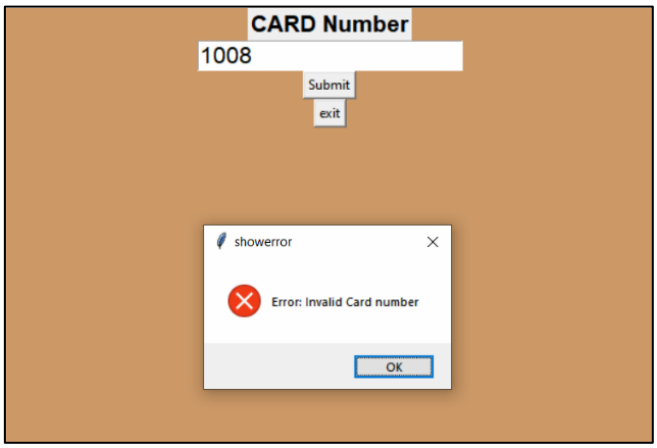
'E:\Cafe Management\saved cards.csv
Valid Card Numbers

```
lst = df1['card number'].tolist()

intcdno = int(cardno)
```

```
if (intcdno in lst):
   df = pd.read_csv('E:\Cafe Management\punch.csv',usecols =
   ['card number','punch details',])
   df2 = pd.DataFrame({'card number':[cardno],'punch
   details':[timestamp]})
       df3 = df.append(df2,ignore_index = True)
   df3.to_csv('E:\Cafe Management\punch.csv',index=False)
   card_var.set("")
   r1.destroy()
   execfile('E:\Cafe Management\CAFE FINAL 1.py')#pg-15
```

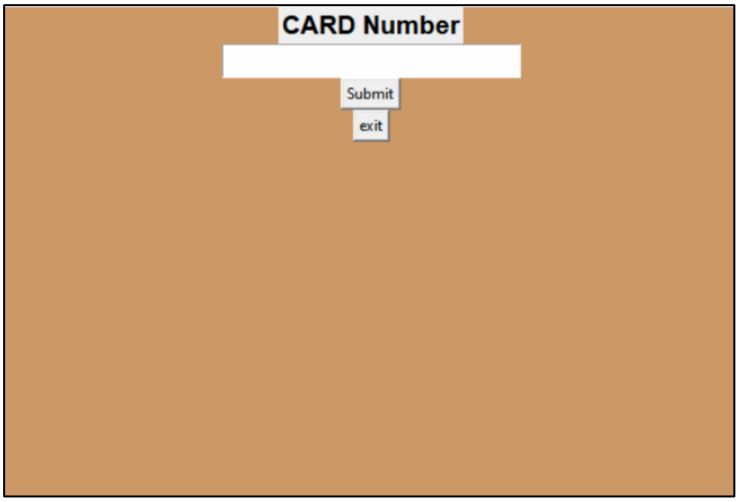


(b.2) Login window – Invalid Authentication




```
# ////////////////// error for wrong card number ////////////

else:
        messagebox.showerror("showerror", "Error: Invalid Card
number")
        card_var.set("")

# //// pressing the exit button will exit the program //////

def ext():
    r1.destroy()
```

```
# ////////////////// code for main window //////////////////
```




```
name_label = tk.Label(r1, text = 'CARD Number',
font=('calibre',16, 'bold')).pack()

name_entry = tk.Entry(r1,textvariable = card_var,
font=('calibre',16,'normal')).pack()

sub_btn=tk.Button(r1,text = 'Submit', command = submit).pack()

ext_btn = tk.Button(r1, text = 'exit', command = ext).pack()

r1.mainloop()
```

(C) Billing Window ('E:\Cafe Management\CAFE FINAL 1.py')


```
# ///////////////importing the required libraries ////////////

from tkinter import*

import random

import time;

import datetime

import pandas as pd

import mysql.connector
```

```
mydb =
mysql.connector.connect(host="localhost",user="root",passwd="1
23456",database="Cafe_Management")

mycursor=mydb.cursor()


# ////////////////// backend main window //////////////////////


root= Tk()

root.title("Mocha Tree")

root.configure(background='#5F755E')

root.attributes('-fullscreen', True)

Tops = Frame(root, width=1350,height=100,bd=14,
relief="raise")

Tops.pack(side=TOP)

customerName = StringVar()

customerName.set("")

customerContact = StringVar()

customerContact.set("")

f1 = Frame(root, width=700,height=750,bd=8) #relief = 'raise'

f1.pack(side=LEFT)

f2 = Frame(root, width=340,height=750,bd=8, relief="raise")

f2.pack(side=RIGHT)

customer_name_label = Label(f1, text="Name", font=("arial",
15, "bold"),bg = "#5F755E", fg="#F3EAD4")
```

```
customer_name_label.pack(side=TOP)

customer_name_entry = Entry(f1,width=20,font="arial
15",bd=5,textvariable=customerName)

customer_name_entry.pack(side = TOP)

customer_contact_label = Label(f1, text="Contact",
font=("arial", 15, "bold"),bg = "#5F755E", fg="#F3EAD4")

customer_contact_label.pack(side =TOP)

customer_contact_entry = Entry(f1,width=20,font="arial
15",bd=5,textvariable=customerContact)

customer_contact_entry.pack(side=TOP)

ft2 = Frame(f2, width=340,height=450,bd=12, relief="raise")

ft2.pack(side=TOP)

fb2 = Frame(f2, width=340,height=250,bd=16, relief="raise")

fb2.pack(side=BOTTOM)

f1a = Frame(f1, width=700,height=430,bd=8, relief="raise")

f1a.pack(side=TOP)

f1aa = Frame(f1a, width = 300, height=630,bd=16,
relief="raise")

f1aa.pack(side=LEFT)

f1ab = Frame(f1a, width = 300, height=630,bd=16,
relief="raise")

f1ab.pack(side=RIGHT)

Tops.configure(background='#5F755E')

f1.configure(background='#5F755E')
```

```
f2.configure(background='#5F755E')

lblInfo = Label(Tops, font=('arial',60,'bold'), text= " Mocha
Tree ", bd=10)

lblInfo.grid(row=0,column=0)


#////////////////////// VARIABLES ////////////////////////


var1= IntVar()

var2= IntVar()

var3= IntVar()

var4= IntVar()

var5= IntVar()

var6= IntVar()

var7= IntVar()

var8= IntVar()

var9= IntVar()

var10= IntVar()

var11= IntVar()

var12= IntVar()

var13= IntVar()

var14= IntVar()

var15= IntVar()

var16= IntVar()
```

```
var17 = IntVar()

var18 = IntVar()

var19 = IntVar()

var20 = IntVar()


DateofOrder=StringVar()

Receipt_Ref=StringVar()

PaidTax=StringVar()

SubTotal=StringVar()

TotalCost=StringVar()

ServiceCharge=StringVar()

menulist=["Cappuccino","Americano","Cold_Brew","CC
Muffin","Strawberry Smoothie","Straberry Oreo","Cr Oreo Fr",\

          "Double Chocolate Fr","Season Special 1","Season
Special 2","ColdBrewFV","CheeseCake","Hot Chocolate",\

              "Waffels","MUDPie
Shake","ButterBeer","MochaFrappe","Latte","Menu Of The day
1","Menu Of The day 2"]

pricelist=[125,120,110,70,140,150,160,155,120,85,90,150,145,18
0,135,125,140,140,150,150]

bilist=[]

billdf = pd.DataFrame(index=menulist)

x1=random.randint(10908, 500876)
```

```
E_cappuccino=StringVar()

E_americano=StringVar()

E_Cold_Brew=StringVar()

E_CCMff=StringVar()

E_Sr_Smoothie=StringVar()

E_SOF=StringVar()

E_COF=StringVar()

E_DCF=StringVar()


E_Cold_Brew_FV=StringVar()

E_CheeseCake=StringVar()

E_HotChoc=StringVar()

E_Waffels=StringVar()

E_MDShake=StringVar()

E_ButterBeer=StringVar()

E_MochaFrappe=StringVar()

E_Latte=StringVar()


E_Menu_Of_Day1 = StringVar()

E_Menu_Of_Day2 = StringVar()

E_Season_Special_1 = StringVar()

E_Season_Special_2 = StringVar()
```

```
E_cappuccino.set("0")

E_americano.set("0")

E_Cold_Brew.set("0")

E_CCMff.set("0")

E_Sr_Smoothie.set("0")

E_SOF.set("0")

E_COF.set("0")

E_DCF.set("0")


E_Cold_Brew_FV.set("0")

E_CheeseCake.set("0")

E_HotChoc.set("0")

E_Waffels.set("0")

E_MDShake.set("0")

E_ButterBeer.set("0")

E_MochaFrappe.set("0")

E_Latte.set("0")


E_Menu_Of_Day1.set('0')

E_Menu_Of_Day2.set('0')

E_Season_Special_1.set('0')

E_Season_Special_2.set('0')

bi ="E:/Cafe Management/bill records/"
```

```
DateofOrder.set(time.strftime("%d/%m/%y"))


# //////// Printing the total cost of items////////////


def CostofItems():
    if (len(customerName.get()) == 0):
        messagebox.showerror("showerror", "Error: Invalid
Customer details")
        customerName.set("")
    elif (len(customerContact.get()) == 0):
        messagebox.showerror("showerror", "Error: Invalid
Customer Details")
        customerContact.set("")
    Cost1=125
    Cost2=120
    Cost3=110
    Cost4=70
    Cost5=140
    Cost6=150
    Cost7=160
    Cost8=155
    Cost9=120
    Cost10=85
```

```
Cost11=90

Cost12=150

Cost13=145

Cost14=180

Cost15=135

Cost16=125

Cost17 = 140

Cost18 = 140

Cost19 = 150

Cost20 = 150

Item1=float(E_cappuccino.get())

Item2=float(E_americano.get())

Item3=float(E_Cold_Brew.get())

Item4=float(E_CCMff.get())

Item5=float(E_Sr_Smoothie.get())

Item6=float(E_SOF.get())

Item7=float(E_COF.get())

Item8=float(E_DCF.get())

Item9=float(E_Cold_Brew_FV.get())

Item10=float(E_CheeseCake.get())

Item11=float(E_HotChoc.get())

Item12=float(E_Waffels.get())

Item13=float(E_MDShake.get())
```

```
    Item14=float(E_ButterBeer.get())
    Item15=float(E_MochaFrappe.get())
    Item16=float(E_Latte.get())
    Item17 = float(E_Menu_Of_Day1.get())
    Item18 = float(E_Menu_Of_Day2.get())
    Item19 = float(E_Season_Special_1.get())
    Item20 = float(E_Season_Special_2.get())
    binm = str(customerName.get())
    bilc = str(customerContact.get())
    b = ""
    bv = str(Receipt_Ref.get())
    if bv==b:
        bv=str(x1)
    b1 = """INSERT INTO custdet VALUES(%s,%s,%s)"""
    b2 = (binm,bilc,bv)
    mycursor.execute(b1,b2)
    mydb.commit()
    b = bi + bv + ".csv"
   orderlist=
[Item1,Item2,Item3,Item4,Item5,Item6,Item7,Item8,Item9,Item10,
Item11,Item12,Item13,Item14,Item15,Item16,Item17,Item18,Item19
,Item20]
    ST=0
```

```
for i in range(20):
        for j in range(20):
            if i==j:
                bilist.append(pricelist[i]*orderlist[j])
                ST+= (pricelist[i]*orderlist[j])
    billdf["Quantity"]=orderlist
    billdf["Rate"]=pricelist
    billdf["IndividualCost"]=bilist
    billdf.to_csv(b)
    Price = (Item1 * Cost1) + (Item2 * Cost2) + (Item3 *
Cost3) + (Item4 * Cost4) + (Item5 * Cost5) + (Item6 * Cost6) +
(Item7 * Cost7) + (Item8 * Cost8) + (Item17 * Cost17) +
(Item18 * Cost18) + (Item19 * Cost19) + (Item20 * Cost20)+
(Item9 * Cost9) + (Item10 * Cost10) + (Item11 * Cost11) +
(Item12 * Cost12) + (Item13 * Cost13) + (Item14 * Cost14) +
(Item15 * Cost15) + (Item16 * Cost16)
    SC= "Rs" , str(0.04 * Price)
    ServiceCharge.set(SC)
    SubTotalofITEMS = "Rs" , str(Price+(0.04 * Price))
    SubTotal.set(SubTotalofITEMS)
    Tax= "Rs" , str(0.11 * Price)
    PaidTax.set(Tax)
    TC = "Rs" , str(Price + (0.04 * Price) + (0.11 * Price))
    TotalCost.set(TC)
```

```
    a1 = int(E_cappuccino.get())
    a2 = int(E_americano.get())
    a3 = int(E_Cold_Brew.get())
    a4 = int(E_CCMff.get())
    a5 = int(E_Sr_Smoothie.get())
    a6 = int(E_SOF.get())
    a7 =int(E_COF.get())
    a8 =int(E_DCF.get())
    a9 =int(E_Cold_Brew_FV.get())
    a10 =int(E_CheeseCake.get())
    a11 =int(E_HotChoc.get())
    a12 =int(E_Waffels.get())
    a13 =int(E_MDShake.get())
    a14 =int(E_ButterBeer.get())
    a15 =int(E_MochaFrappe.get())
    a16 =int(E_Latte.get())
    a17 =int(E_Menu_Of_Day1.get())
    a18 =int(E_Menu_Of_Day2.get())
    a19 =int(E_Season_Special_1.get())
    a20 =int(E_Season_Special_2.get())
    txtReceipt.delete("1.0",END)
    txtReceipt.insert(END,'Receipt Ref:\t\t\t'+ bv + '\t\t'+
DateofOrder.get()+"\n")
```

```
txtReceipt.insert(END,'=====================================
========\n'+ 'Items\t\t'+'\tRate\t\t'+"Count of Items \n"+ '--
------------------------------------------------------------
--------------------\n')

    while a1 > 0:

        txtReceipt.insert(END,'Cappuccino:\t\t\t'+ str(Cost1)
+'\t\t\t' + E_cappuccino.get()+"\n")

        break

    while a2 > 0:

        txtReceipt.insert(END,'Americano:\t\t\t'+ str(Cost2)
+'\t\t\t' + E_americano.get()+"\n")

        break

    while a3 > 0:

        txtReceipt.insert(END,'Cold Brew:\t\t\t'+ str(Cost3)
+'\t\t\t' + E_Cold_Brew.get()+"\n")

        break

    while a4 > 0:

        txtReceipt.insert(END,'Chocochip Muffin:\t\t\t'+
str(Cost4) +'\t\t\t' + E_CCMff.get()+"\n")

        break

    while a5 > 0:

        txtReceipt.insert(END,'Strawberry Smoothie:\t\t\t'+
str(Cost5) +'\t\t\t' + E_Sr_Smoothie.get()+"\n")

        break
```

```
 while a6 > 0:
        txtReceipt.insert(END,'Sr oreo Frappe:\t\t\t'+
str(Cost6) +'\t\t\t' + E_SOF.get()+"\n")
        break

    while a7 > 0:
        txtReceipt.insert(END,'Cr Oreo Frappe:\t\t\t'+
str(Cost7) +'\t\t\t' + E_COF.get()+"\n")
        break
    while a8 > 0:
        txtReceipt.insert(END,'Double Ch Fr:\t\t\t'+
str(Cost8) +'\t\t\t' + E_DCF.get()+"\n")
        break
    while a9 > 0:
        txtReceipt.insert(END,'ColdBrewFV:\t\t\t'+ str(Cost9)
+'\t\t\t' + E_Cold_Brew_FV.get()+"\n")
        break
    while a10 > 0:
        txtReceipt.insert(END,'CheeseCake:\t\t\t'+ str(Cost10)
+'\t\t\t' + E_CheeseCake.get()+"\n")
        break
    while a11 > 0:
        txtReceipt.insert(END,'Hot Chocolate:\t\t\t'+
str(Cost11) +'\t\t\t' + E_HotChoc.get()+"\n")
        break
```

```
    while a12 > 0:
        txtReceipt.insert(END,'Waffels:\t\t\t'+ str(Cost12)
+'\t\t\t' +E_Waffels.get()+"\n")
        break

    while a13 > 0:
        txtReceipt.insert(END,'MUDPieShake:\t\t\t'+
str(Cost13) +'\t\t\t' + E_MDShake.get()+"\n")
        break
    while a14 > 0:
        txtReceipt.insert(END,'ButterBeer:\t\t\t'+ str(Cost14)
+'\t\t\t' + E_ButterBeer.get()+"\n")
        break
    while a15 > 0:
        txtReceipt.insert(END,'MochaFrappe:\t\t\t'+
str(Cost15) +'\t\t\t' + E_MochaFrappe.get()+"\n")
        break
    while a16 > 0:
        txtReceipt.insert(END,'Latte:\t\t\t'+ str(Cost16)
+'\t\t\t' + E_Latte.get()+"\n")
        break
    while a17 > 0:
        txtReceipt.insert(END,'Menu_Of_Day_1:\t\t\t'+
str(Cost17) +'\t\t\t' + E_Menu_Of_Day1.get()+"\n")
        break
```

```
    while a18 > 0:
        txtReceipt.insert(END,'Menu_Of_Day_2:\t\t\t'+
str(Cost18) +'\t\t\t' + E_Menu_Of_Day2.get()+"\n")
        break

    while a19 > 0:
        txtReceipt.insert(END,'Season_Special_1:\t\t\t'+
str(Cost19) +'\t\t\t' + E_Season_Special_1.get()+"\n")
        break
    while a20 > 0:
        txtReceipt.insert(END,'Season_Special_2:\t\t\t' +
str(Cost20) +'\t\t\t' + E_Season_Special_2.get()+"\n")
        break

txtReceipt.insert(END,'\n=====================================
==========\n' + 'Tax Paid:\t\t' + PaidTax.get() +"\n")
    txtReceipt.insert(END,'Service Charge:\t\t'+
ServiceCharge.get()+'\nSubTotal:\t\t' +SubTotal.get() +"\n")

txtReceipt.insert(END,'=====================================
========\n'+ 'Total Cost:\t\t' + TotalCost.get()
+"\n=============================================\n")
```

```
# //////// reset button:  resetting the values to zero ///////
```


```
def Reset():
    PaidTax.set("")
    SubTotal.set("")
    TotalCost.set("")
    txtReceipt.delete("1.0",END)
    x = random.randint(10908, 500876)
    randomRef = str(x)
    Receipt_Ref.set(randomRef)


    E_cappuccino.set("0")
    E_americano.set("0")
    E_Cold_Brew.set("0")
    E_CCMff.set("0")
    E_Sr_Smoothie.set("0")
```

```
E_SOF.set("0")

E_COF.set("0")

E_DCF.set("0")

E_Cold_Brew_FV.set("0")

E_CheeseCake.set("0")

E_HotChoc.set("0")

E_Waffels.set("0")

E_MDShake.set("0")

E_ButterBeer.set("0")

E_MochaFrappe.set("0")

E_Latte.set("0")

E_Menu_Of_Day1.set('0')

E_Menu_Of_Day2.set('0')

E_Season_Special_1.set('0')

E_Season_Special_2.set('0')

customerName.set("")

customerContact.set("")


var1.set(0)

var2.set(0)

var3.set(0)

var4.set(0)

var5.set(0)
```

```
var6.set(0)

var7.set(0)

var8.set(0)

var9.set(0)

var10.set(0)

var11.set(0)

var12.set(0)

var13.set(0)

var14.set(0)

var15.set(0)

var16.set(0)

var17.set(0)

var18.set(0)

var19.set(0)

var20.set(0)


txtcappuccino.configure(state=DISABLED)

txtamericano.configure(state=DISABLED)

txtCold_Brew.configure(state=DISABLED)

txtCCMff.configure(state=DISABLED)

txtSr_Smoothie.configure(state=DISABLED)

txtSOF.configure(state=DISABLED)

txtCOF.configure(state=DISABLED)
```

```
    txtDCF.configure(state=DISABLED)
    txtCold_Brew_FV.configure(state=DISABLED)
    txtCheeseCake.configure(state=DISABLED)
    txtHotChoc.configure(state=DISABLED)
    txtWaffels.configure(state=DISABLED)
    txtMDShake.configure(state=DISABLED)
    txtButterBeer.configure(state=DISABLED)
    txtMochaFrappe.configure(state=DISABLED)
    txtLatte.configure(state=DISABLED)
    txtMen1.configure(state=DISABLED)
    txtMen2.configure(state=DISABLED)
    txtSp1.configure(state=DISABLED)
    txtSp2.configure(state=DISABLED)

# ////////////////////// search button ////////////////////////

def srchb():
    execfile("E:\Cafe Management\sf1.py")#pg-45
```

```
# ////////////////// exit button //////////////////////////
```


```
def goback():
    qExit=messagebox.askyesno("Quit System","Do you want to quit?")
    if qExit > 0:
        root.destroy()
        return
```

```
#///// placing menu item check buttons and other widgets /////
```


```
Capp = Checkbutton(f1aa, text="  Cappuccino\t\t",
variable=var1, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=0,s
ticky=W)
```

```
Americano = Checkbutton(f1aa, text="  Americano\t\t",
variable=var2, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=1,s
ticky=W)
```

```
Cold_Brew = Checkbutton(f1aa, text="  Cold Brew\t\t",
variable=var3, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=2,s
ticky=W)
```

```
Cold_Brew_FV = Checkbutton(f1aa, text="  CC Muffin\t\t",
variable=var4, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=3,s
ticky=W)


Sr_Smoothie = Checkbutton(f1aa, text="  Strawberry
Smoothie\t\t", variable=var5, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=4,s
ticky=W)


SOF = Checkbutton(f1aa, text="  Straberry Oreo Fr\t\t",
variable=var6, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=5,s
ticky=W)


COF = Checkbutton(f1aa, text="  Cr Oreo Fr\t\t",
variable=var7, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=6,s
ticky=W)

DOC = Checkbutton(f1aa, text="  Double Chocolate Fr\t\t",
variable=var8, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=7,s
ticky=W)


MOD1 = Checkbutton(f1aa, text="  Season Special 1\t\t",
variable=var19, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=8,s
ticky=W)
```

```
MOD2 = Checkbutton(f1aa, text="  Season Special 2\t\t",
variable=var20, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=9,s
ticky=W)


ColdBrewFV = Checkbutton(f1ab, text=" ColdBrewFV \t\t",
variable=var9, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=0,s
ticky=W)


CheeseCake = Checkbutton(f1ab, text=" CheeseCake \t\t",
variable=var10, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=1,s
ticky=W)


Hot_Chocolate = Checkbutton(f1ab, text=" Hot Chocolate \t\t",
variable=var11, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=2,s
ticky=W)

Waffels = Checkbutton(f1ab, text=" Waffels \t\t",
variable=var12, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=3,s
ticky=W)


MUDPieShake = Checkbutton(f1ab, text=" MUDPie Shake \t\t",
variable=var13, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=4,s
ticky=W)
```

```
ButterBeer = Checkbutton(f1ab, text=" ButterBeer \t\t",
variable=var14, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=5,s
ticky=W)


MochaFrappe = Checkbutton(f1ab, text=" MochaFrappe \t\t",
variable=var15, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=6,s
ticky=W)


Latte = Checkbutton(f1ab, text=" Latte \t\t", variable=var16,
onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=7,s
ticky=W)


SPP1 = Checkbutton(f1ab, text="  Menu Of The day 1 \t",
variable=var17, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=8,s
ticky=W)

SPP2 = Checkbutton(f1ab, text="  Menu Of The Day 2 \t",
variable=var18, onvalue=1, offvalue=0,
font=('arial',18,'bold'),command=chkbutton_value).grid(row=9,s
ticky=W)


txtcappuccino =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_cappuccino, state=DISABLED)
```

```
txtcappuccino.grid(row=0,column=1)


txtamericano =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_americano, state=DISABLED)

txtamericano.grid(row=1,column=1)


txtCold_Brew =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Cold_Brew,state=DISABLED)

txtCold_Brew.grid(row=2,column=1)


txtCCMff =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_CCMff,state=DISABLED)

txtCCMff.grid(row=3,column=1)


txtSr_Smoothie =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Sr_Smoothie,state=DISABLED)

txtSr_Smoothie.grid(row=4,column=1)


txtSOF =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_SOF,state=DISABLED)

txtSOF.grid(row=5,column=1)
```

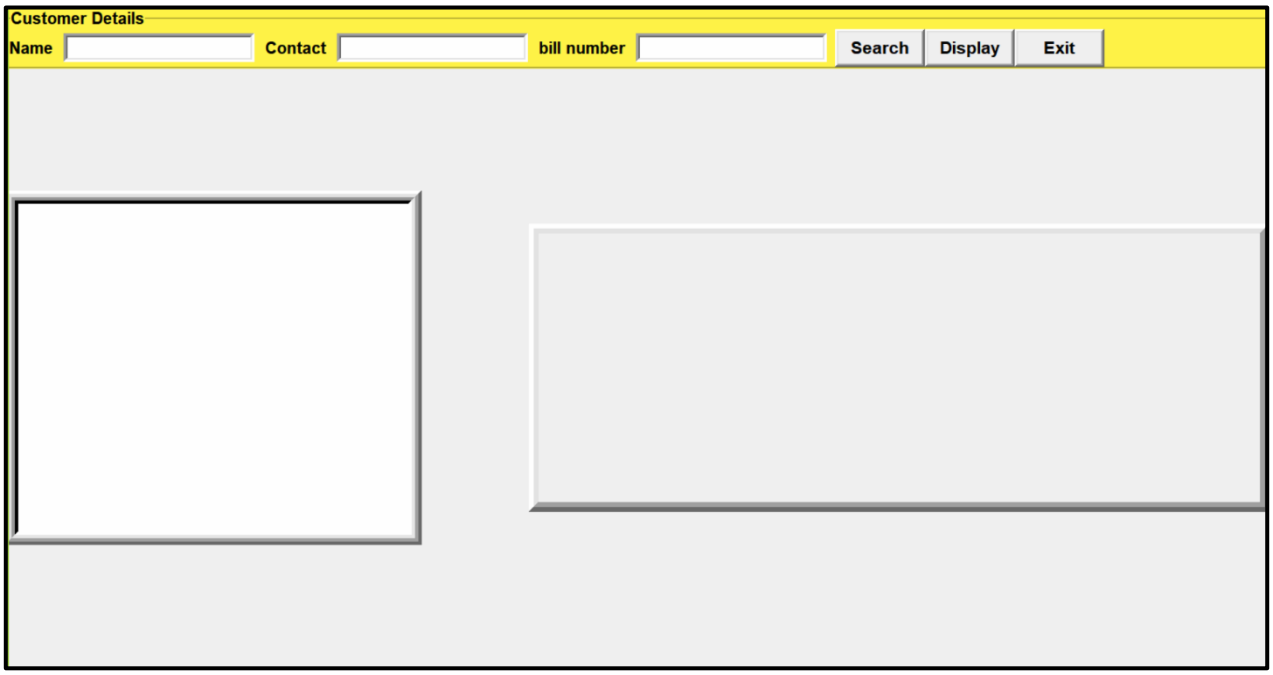
```
txtCOF =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_COF,state=DISABLED)

txtCOF.grid(row=6,column=1)


txtDCF =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_DCF,state=DISABLED)

txtDCF.grid(row=7,column=1)


txtSp1 =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Season_Special_1,state=DISABLED)

txtSp1.grid(row=8,column=1)


txtSp2 =
Entry(f1aa,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Season_Special_2,state=DISABLED)

txtSp2.grid(row=9,column=1)


txtCold_Brew_FV =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Cold_Brew_FV,state=DISABLED)

txtCold_Brew_FV.grid(row=0,column=1)
```

```
txtCheeseCake =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_CheeseCake,state=DISABLED)

txtCheeseCake.grid(row=1,column=1)


txtHotChoc =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_HotChoc,state=DISABLED)

txtHotChoc.grid(row=2,column=1)


txtWaffels =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Waffels,state=DISABLED)

txtWaffels.grid(row=3,column=1)


txtMDShake =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_MDShake,state=DISABLED)

txtMDShake.grid(row=4,column=1)


txtButterBeer =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_ButterBeer,state=DISABLED)

txtButterBeer.grid(row=5,column=1)
```

```
txtMochaFrappe =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_MochaFrappe,state=DISABLED)

txtMochaFrappe.grid(row=6,column=1)


txtLatte =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Latte,state=DISABLED)

txtLatte.grid(row=7,column=1)


txtMen1 =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Menu_Of_Day1,state=DISABLED)

txtMen1.grid(row=8,column=1)


txtMen2 =
Entry(f1ab,font=('arial',16,'bold'),bd=8,width=6,justify='left
',textvariable=E_Menu_Of_Day2,state=DISABLED)

txtMen2.grid(row=9,column=1)


# /////// placing receipt box /////////////////

lblReceipt = Label(ft2,font=('arial',16,'bold'),text="Get
Receipt:",bd=2,anchor='w')


lblReceipt.grid(row=0,column=0,sticky=W)
```

```
txtReceipt =
Text(ft2,font=('arial',11,'bold'),bd=8,width=59,height=22,bg="
white")

txtReceipt.grid(row=1,column=0)


# ////////////////////////// other buttons ////////////////////


btnTotal=Button(fb2,padx=16,pady=1,bd=4,fg="black",font=('aria
l',16,'bold'),width=5,text="Total",command=CostofItems).grid(r
ow=0,column=0)


btnReset=Button(fb2,padx=16,pady=1,bd=4,fg="black",font=('aria
l',16,'bold'),width=5,text="Reset",command=Reset).grid(row=0,c
olumn=2)


btnSearch=Button(fb2,padx=16,pady=1,bd=4,fg="black",font=('ari
al',16,'bold'),width=5,text="Search",command=srchb).grid(row=0
,column=3)


btnExit=Button(fb2,padx=16,pady=1,bd=4,fg="black",font=('arial
',16,'bold'),width=5,text="Exit",command=goback).grid(row=0,co
lumn=4)


root.mainloop()
```

(D) Search Bill Window ('E:\Cafe Management\sf1.py')



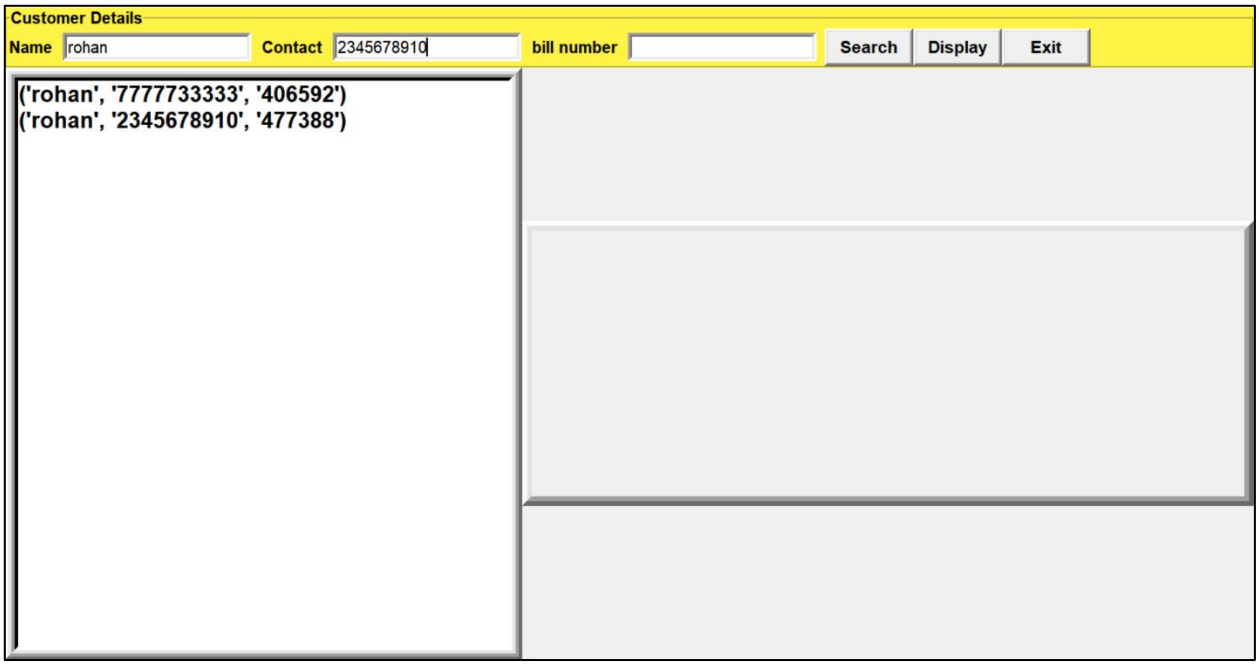

```
# ////////////// importing required libraries ///////////////


from tkinter import*

import csv

import pandas as pd

import mysql.connector

mydb =
mysql.connector.connect(host="localhost",user="root",passwd="1
23456",database="Cafe_Management")

mycursor=mydb.cursor()
```

```
# /////////////// main search window code backend ///////////

root1= Tk()

root1.geometry("1600x8000")


# ////////////////////// setting variables ////////////////////


ntbs = StringVar()

ntbs.set("")

ctbs = StringVar()

ctbs.set("")

btbs = StringVar()

btbs.set("")


#////////// front end code and required entry boxes ////////


customer_frame = LabelFrame(root1,text="Customer
Details",font=("arial", 15, "bold"),bg="#fff346",
relief=GROOVE)

customer_frame.pack(side=TOP, fill="x")


customer_name_label = Label(customer_frame, text="Name",
font=("arial", 15, "bold"),bg = "#fff346", fg="black")

customer_name_label.grid(row = 0, column = 0)
```

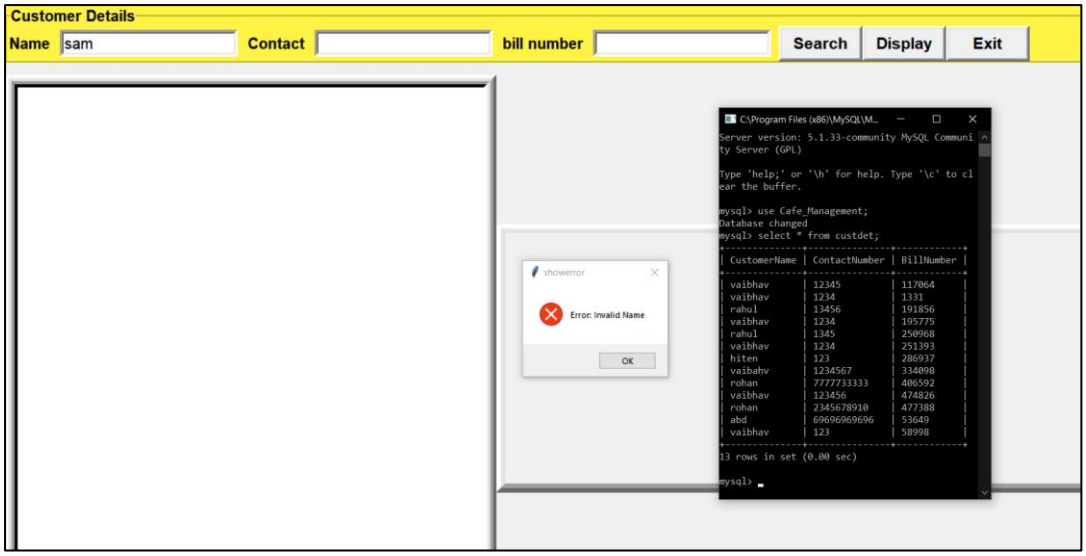
```
customer_name_entry =
Entry(customer_frame,width=20,font="arial
15",bd=5,textvariable=ntbs)

customer_name_entry.grid(row = 0, column=1,padx=10)


customer_contact_label = Label(customer_frame, text="Contact",
font=("arial", 15, "bold"),bg = "#fff346", fg="black")

customer_contact_label.grid(row = 0, column = 2)


customer_contact_entry =
Entry(customer_frame,width=20,font="arial
15",bd=5,textvariable=ctbs)

customer_contact_entry.grid(row = 0, column=3,padx=10)


bill_number_label = Label(customer_frame, text="bill number",
font=("arial", 15, "bold"),bg = "#fff346", fg="black")

bill_number_label.grid(row = 0, column = 4)


bill_number_entry = Entry(customer_frame,width=20,font="arial
15",bd=5,textvariable=btbs)

bill_number_entry.grid(row = 0, column=5,padx=10)



# ////////////// creating additional frames to display bill
records and necessary data ///////////////
```

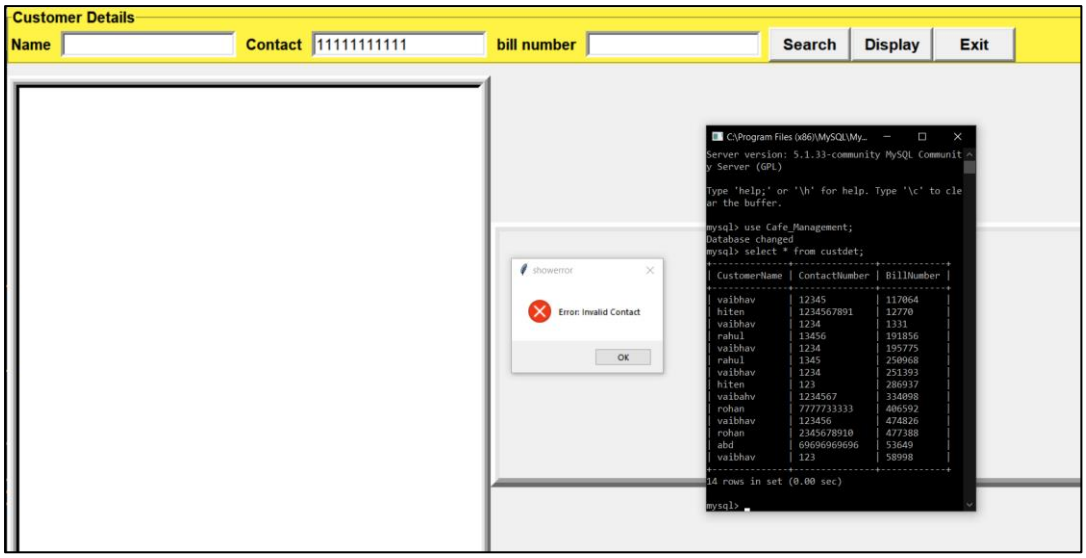
```
ft1.pack(side=RIGHT)

f2 = Frame(root1, width=940,height=650,bd=8, relief="raise")

f2.pack(side=LEFT)

txtsrch =
Text(f2,font=('arial',20,'bold'),bd=8,width=59,height=22,bg="w
hite")

txtsrch.pack()


'''

The further flow of the code is explained as follows:
When the bill is generated, a csv file with the bill number as
the file name. At the same time the name, contact number and
the bill no is stored in a sql table with bill number as the
primary key. So even if a customer orders again, his name will
be associated with different bill no. every time.

We enter either name of the customer or contact number in the
search window and press search, it searches the sql table for
the name or contact number, and display the found records in
the output box.

Now we can take the bill number from the above reference and
put it in the Bill number field. It will search the required
folder for the entered bill number as the file name.

'''
```

```
# ////////////// search button ////////////////
```



CustomerName	ContactNumber	BillNumber
vaibhav	12345	117064
vaibhav	1234	1331
rahul	13456	191856
vaibhav	1234	195775
rahul	1345	250968
vaibhav	1234	251393
hiten	123	286937
vaibahv	1234567	334098
rohan	7777733333	406592
vaibhav	123456	474826
rohan	2345678910	477388
vaibhav	123	58998

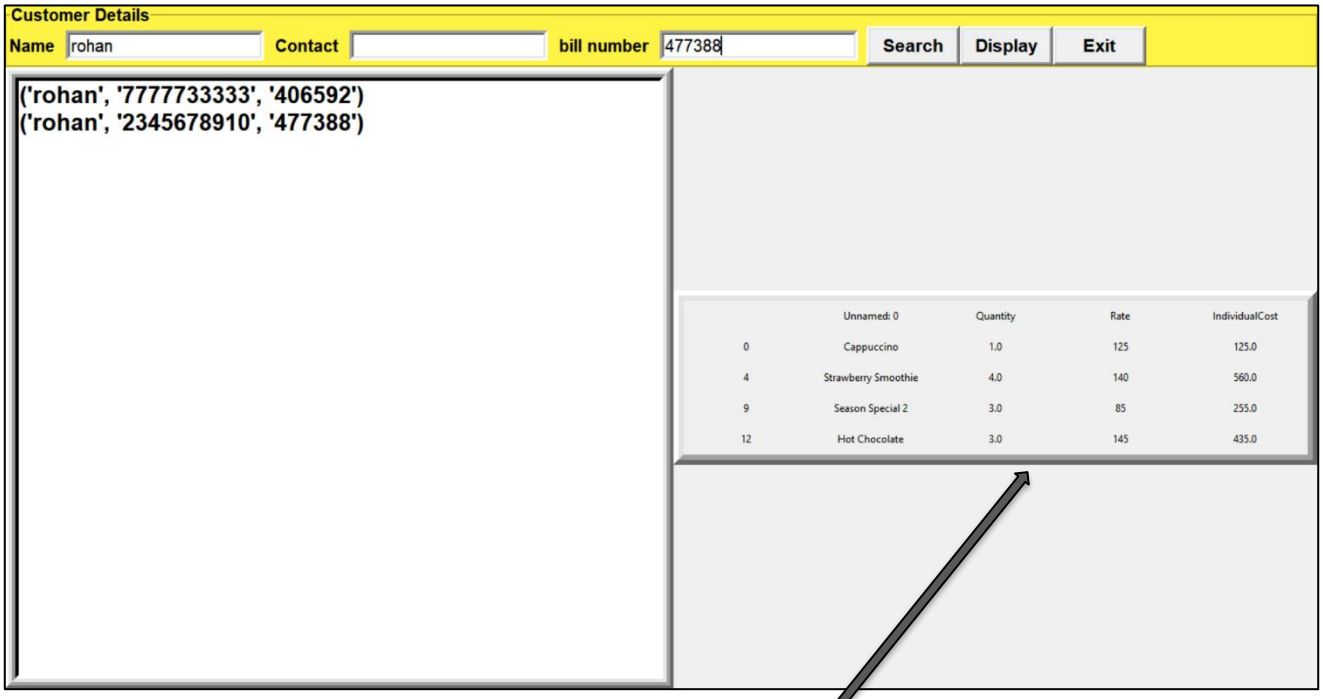





```
def sbn():
    w1=str(ntbs.get())
    w2 = str(ctbs.get())
    w4 = """SELECT * FROM custdet WHERE CustomerName=(%s) OR
ContactNumber=(%s)"""
    mycursor.execute(w4,(w1,w2))
    for val in mycursor:
```

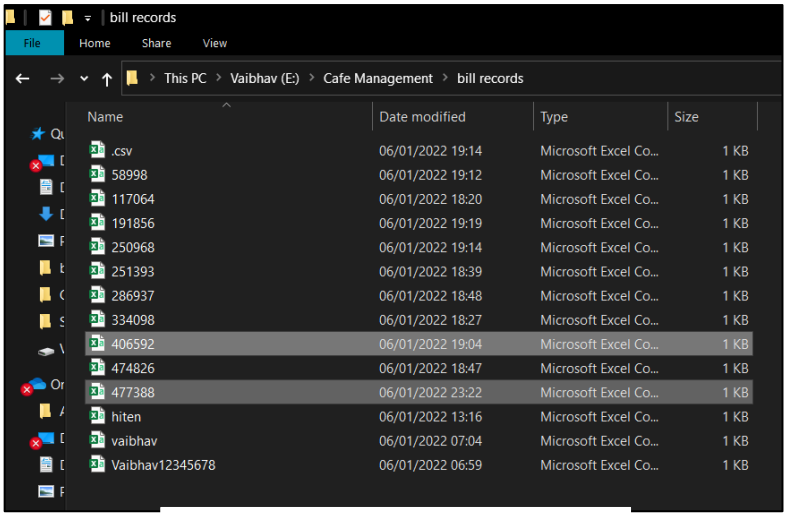
```
        srchval = str(val)

        txtsrch.insert(END,srchval + "\n")
```

```
# ////////////////////// display button ////////////////////////
```



desired bill data

	Unnamed: 0	Quantity	Rate	IndividualCost
0	Cappuccino	1.0	125	125.0
4	Strawberry Smoothie	4.0	140	560.0
9	Season Special 2	3.0	85	255.0
12	Hot Chocolate	3.0	145	435.0



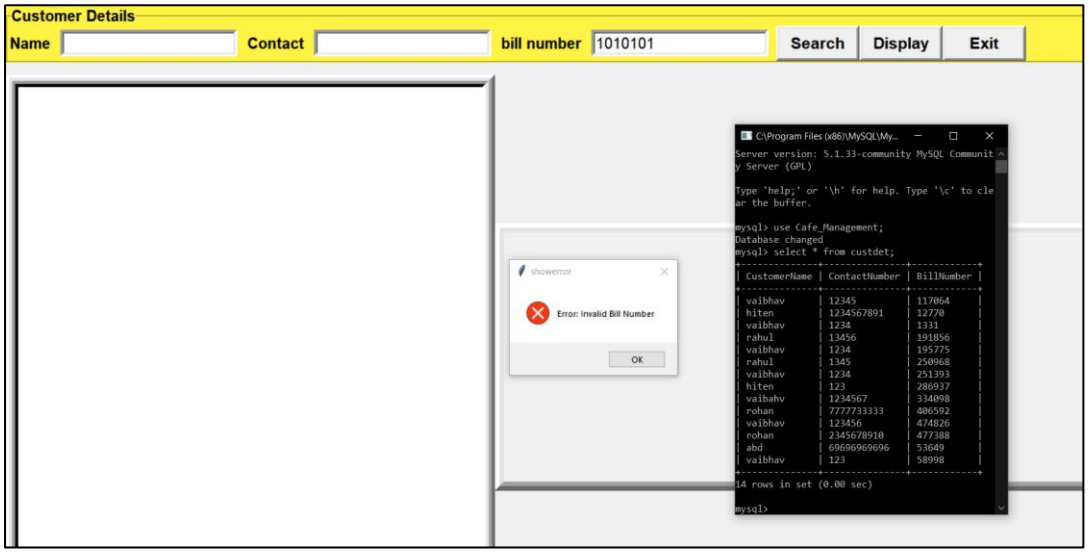
Every Bill is saved in a folder as CSV file

Unnamed: 0	Quantity	Rate	IndividualCost
Cappuccino	1	125	125
Strawberry Smoothie	4	140	560
Season Special 2	3	85	255
Hot Chocolate	3	145	435

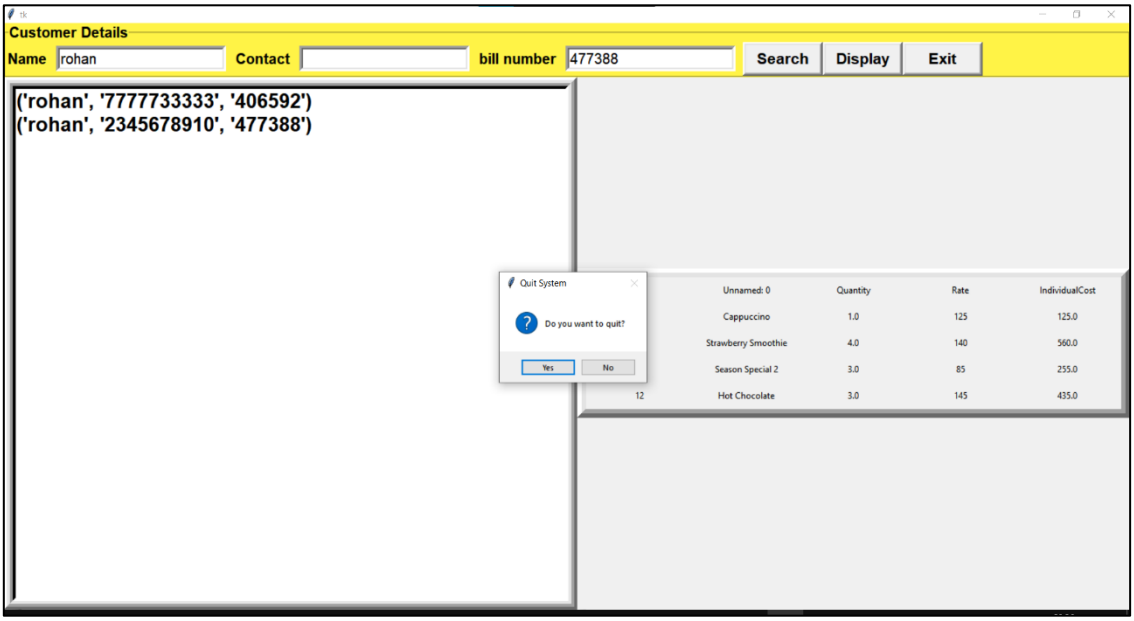
CSV record of the BILL

```
s="E:/Cafe Management/bill records/"

def sbtn():

    s1=str(btbs.get())

    s2=""

    s2=s+s1+".csv"

    bild = pd.read_csv(s2)

    dftbu = bild[bild["Quantity"]>0]

    dftbu.to_csv(s2)

    with open(s2, newline = "") as file:

       reader = csv.reader(file)
```

```
        # r and c tell us where to grid the labels
        r = 0
        for col in reader:
            c = 0
            for row in col:

                label = Label(ft1, width = 20, height = 2, text =
row)
                label.grid(row = r, column = c)
                c += 1
            r += 1
```

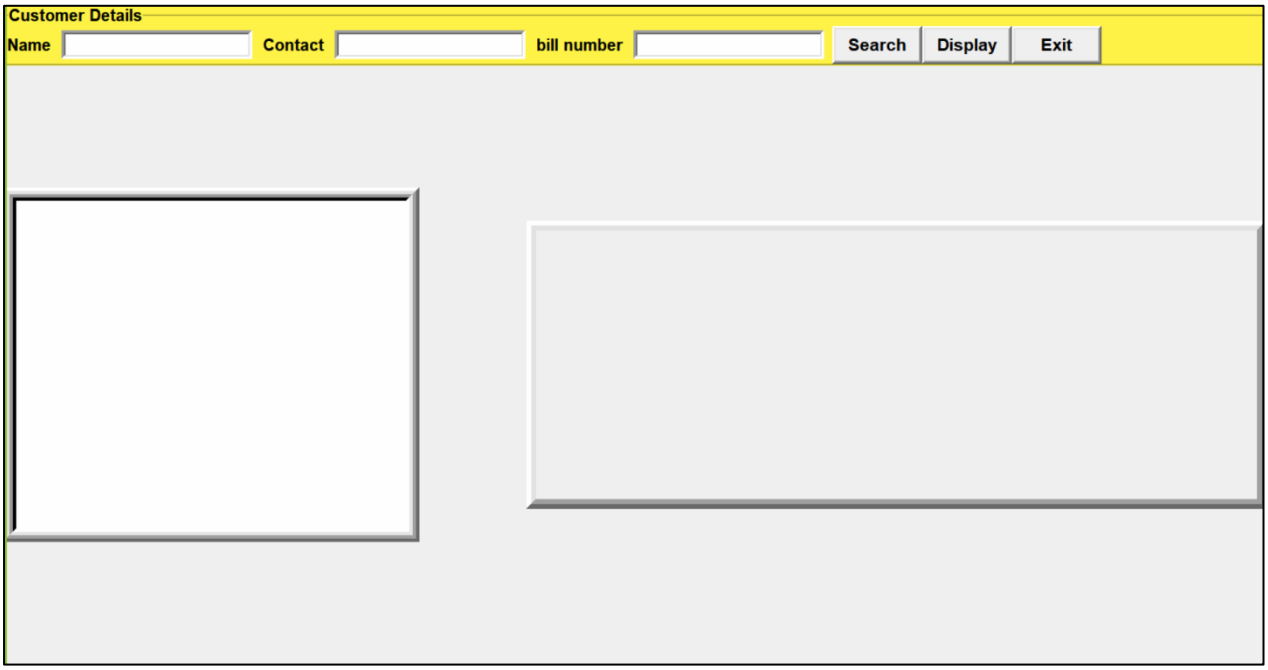


```
# ////////////////// exit window button //////////////////////
```




```
def ext():
    qexit=messagebox.askyesno("Quit System","Do you want to quit?")
    if qexit > 0:
        root1.destroy()
        return
```

```
# /////////////// search, exit and display button and
remaining window alignment//////////
```




```
Searchbtn=Button(customer_frame,padx=16,pady=1,bd=4,fg="black"
,font=('arial',16,'bold'),width=5,text="Search",command=sbn).g
rid(row=0,column=6)


displbtn=Button(customer_frame,padx=16,pady=1,bd=4,fg="black",
font=('arial',16,'bold'),width=5,text="Display",command=sbtn).
grid(row=0,column=7)


extbtn=Button(customer_frame,padx=16,pady=1,bd=4,fg="black",fo
nt=('arial',16,'bold'),width=5,text="Exit",command=ext).grid(r
ow=0,column=8)


root1.mainloop()
```

Now, we finally conclude the programming of the aforesaid software.

We will now discuss about few limitations encountered by the Software Project.

Known Limitations

Following are few limitations/drawbacks that couldn't be removed from this Software Project. So far, we could detect only of them, but there could be many:

- None of the frames are resizable
- The backend index on the menu items selected on the bill display screen are shown.
- None of the menu items are editable
- The search fields in the respective frames are not as user-friendly as in a search engine, say, *Google*.

Bibliography

This Software Project has been made possible due to our concentrated efforts, experiences through laboratory lectures, and our teachers as said before.

We also acknowledge the following internet sources that assisted us throughout the process.

- *GitHub: Where the world builds software · GitHub*

  https://github.com/

- *Python Programming Language – GeeksforGeeks*

  https://www.geeksforgeeks.org/python-programming-language/

- *Stack Overflow - Where Developers Learn, Share, & Build ...*

  https://stackoverflow.com/

- ***Logo Maker | Make a Free Logo|***

  https://www.logomaker.com/home-return

- ***Canva - https://www.canva.com/***

- ***Informatics Textbook for class XI (Sumita Arora)***

- ***Informatics Textbook for class XII (Sumita Arora)***

****