ANSWER SHEET

WEM411

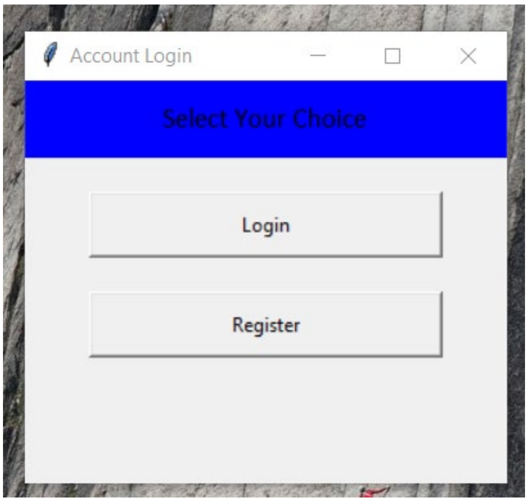
JUNE 14, 2019

CHRIS GRUNDLING`

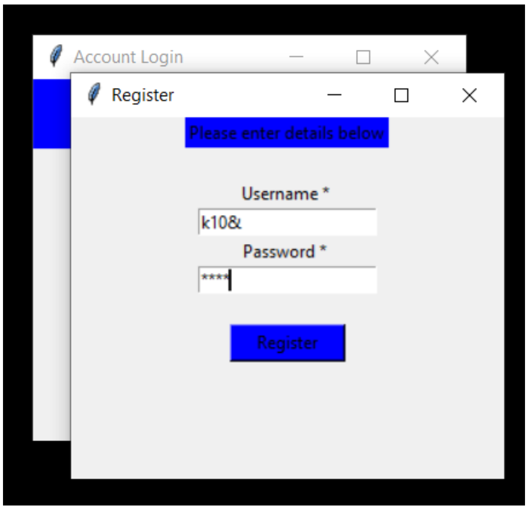
5485

Table of Contents

Login Account Window:



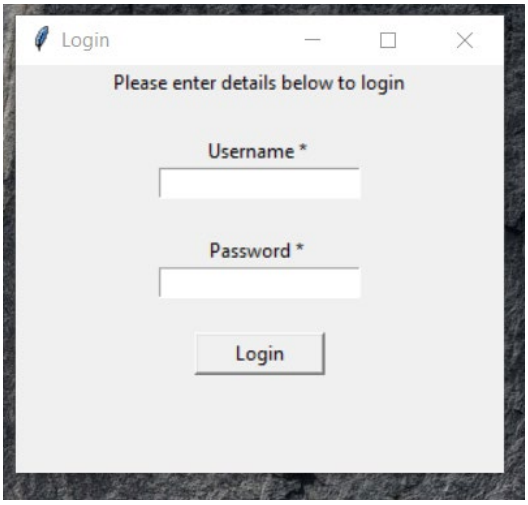
Register Window:



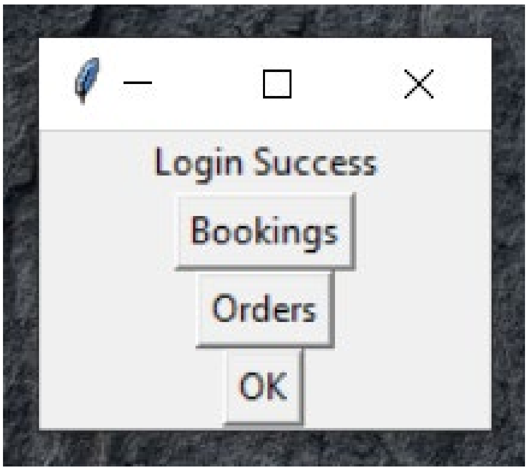
Username * k10&

Password * ****

Login Window:



Successful Login Window:



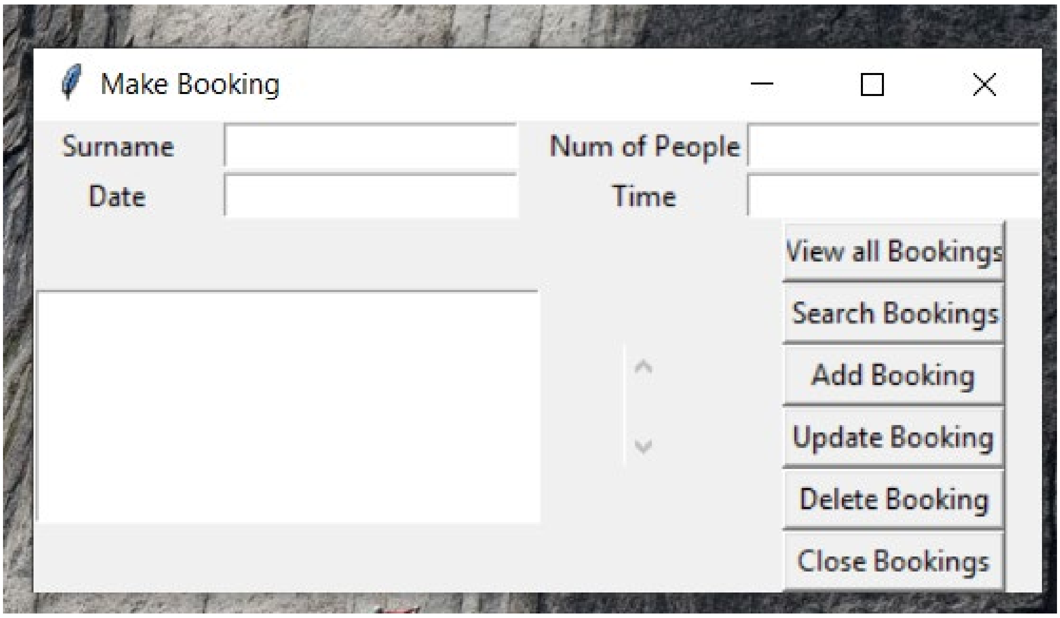
Login Success

All Login_window.py Code:

```
import tkinter as tk
from tkinter import *
from tkinter import ttk
import os

#=================registration Window==========================================

def register():
    global register_screen
    register_screen = Toplevel(main_screen)
    register_screen.title("Register")
    register_screen.geometry("300x250")

    global username
    global password
    global username_entry
    global password_entry
    username = StringVar()
    password = StringVar()

    Label(register_screen, text="Please enter details below", bg="blue").pack()
    Label(register_screen, text="").pack()
    username_lable = Label(register_screen, text="Username * ")
    username_lable.pack()
    username_entry = Entry(register_screen, textvariable=username)
    username_entry.pack()
    password_lable = Label(register_screen, text="Password * ")
    password_lable.pack()
    password_entry = Entry(register_screen, textvariable=password, show='*')
    password_entry.pack()
    Label(register_screen, text="").pack()
    Button(register_screen, text="Register", width=10, height=1, bg="blue", command =
register_user).pack()


# ======================================login Window
==========================================================

def login():
    global login_screen
    flag = False
    while flag == False:
        try:
            login_screen = Toplevel(main_screen)
            login_screen.title("Login")
            login_screen.geometry("300x250")
            Label(login_screen, text="Please enter details below to login").pack()
            Label(login_screen, text="").pack()

            global username_verify
            global password_verify

            username_verify = StringVar()
            password_verify = StringVar()

            global username_login_entry
            global password_login_entry

            Label(login_screen, text="Username * ").pack()
            username_login_entry = Entry(login_screen, textvariable=username_verify)
            username_login_entry.pack()
            Label(login_screen, text="").pack()
            Label(login_screen, text="Password * ").pack()
            password_login_entry = Entry(login_screen, textvariable=password_verify,
show= '*')
            password_login_entry.pack()
            Label(login_screen, text="").pack()
            Button(login_screen, text="Login", width=10, height=1, command =
login_verify).pack()
            flag = True
        except ValueError:
                print("Please enter numbers only !")


# ============================= register
method======================================================

def register_user():

    username_info = username.get()
    password_info = password.get()

    file = open(username_info, "w")
    file.write(username_info + "\n")
    file.write(password_info)
    file.close()

    username_entry.delete(0, END)
    password_entry.delete(0, END)

    Label(register_screen, text="Registration Success", fg="green", font=("calibri",
11)).pack()

# ========================login
method=====================================================================

def login_verify():
    username1 = username_verify.get()
    password1 = password_verify.get()
    username_login_entry.delete(0, END)
    password_login_entry.delete(0, END)

    list_of_files = os.listdir()
    if username1 in list_of_files:
        file1 = open(username1, "r")
        verify = file1.read().splitlines()
        if password1 in verify:
            login_sucess()

        else:
            password_not_recognised()

    else:
        user_not_found()

# ========================login success popup
======================================================

def login_sucess():
    global login_success_screen
    login_success_screen = Toplevel(login_screen)
    login_success_screen.title("Success")
    login_success_screen.geometry("150x100")
    Label(login_success_screen, text="Login Success").pack()
    Button(login_success_screen, text="Bookings", command=Make_Booking).pack()
    Button(login_success_screen, text="Orders", command=Make_Orders).pack()
    Button(login_success_screen, text="OK", command=delete_login_success).pack()


# ==============================login invalid password
popup================================================

def password_not_recognised():
    global password_not_recog_screen
    password_not_recog_screen = Toplevel(login_screen)
    password_not_recog_screen.title("Success")
    password_not_recog_screen.geometry("150x100")
    Label(password_not_recog_screen, text="Invalid Password ").pack()
    Button(password_not_recog_screen, text="OK",
command=delete_password_not_recognised).pack()

# ===================================user not found pop
up=====================================================

def user_not_found():
    global user_not_found_screen
    user_not_found_screen = Toplevel(login_screen)
    user_not_found_screen.title("Success")
    user_not_found_screen.geometry("150x100")
    Label(user_not_found_screen, text="User Not Found").pack()
    Button(user_not_found_screen, text="OK",
command=delete_user_not_found_screen).pack()

# ====================================Deleting
popups=============================================================

def delete_login_success():
    login_success_screen.destroy()


def delete_password_not_recognised():
    password_not_recog_screen.destroy()


def delete_user_not_found_screen():
    user_not_found_screen.destroy()
```

```
def Make_Booking():
    main_screen.destroy()
    import Booking_Front

def Make_Orders():
    main_screen.destroy()
    import Make_Orders


# =========================================Main
window==========================================================

def main_account_screen():
    global main_screen
    main_screen = Tk()
    main_screen.geometry("300x250")
    main_screen.title("Account Login")
    Label(text="Select Your Choice", bg="blue", width="300", height="2",
font=("Calibri", 13)).pack()
    Label(text="").pack()
    Button(text="Login", height="2", width="30", command = login).pack()
    Label(text="").pack()
    Button(text="Register", height="2", width="30", command=register).pack()

    main_screen.mainloop()


main_account_screen()
```

Bookings Window:



Bookings_Front.py Code:

```
from tkinter import *
import tkinter as tk
from tkinter import *
from tkinter import ttk
import os
import Booking_Back
```

```
#===================================getting selected row=============================
def get_selected_row(event):
    global selected_tuple
    index=list1.curselection()[0]
    selected_tuple=list1.get(index)
    e1.delete(0,END)
    e1.insert(END,selected_tuple[1])
    e2.delete(0,END)
    e2.insert(END,selected_tuple[2])
    e3.delete(0,END)
    e3.insert(END,selected_tuple[3])
    e4.delete(0,END)
    e4.insert(END,selected_tuple[4])
#====================================View
method=============================================
def view_command():
    list1.delete(0,END)
    for row in Booking_Back.view():
        list1.insert(END,row)
#====================================Search
method=============================================
def search_command():
    list1.delete(0,END)
    for row in
Booking_Back.search(surname_text.get(),table_num_text.get(),date_text.get(),time_text.
get()):
        list1.insert(END,row)
#====================================Add
method=============================================
def add_command():

Booking_Back.insert(surname_text.get(),table_num_text.get(),date_text.get(),time_text.
get())
    list1.delete(0,END)

list1.insert(END,(surname_text.get(),table_num_text.get(),date_text.get(),time_text.ge
t()))
#====================================Delete
method=============================================
def delete_command():
    Booking_Back.delete(selected_tuple[0])
#====================================Update
method=============================================
def update_command():

Booking_Back.update(selected_tuple[0],surname_text.get(),table_num_text.get(),date_tex
t.get(),time_text.get())
#====================================close
method=============================================
def Clo_Open():
    window.destroy()
    import Login_window

#====================================Generation of
window=============================================
window=Tk()

window.wm_title("Make Booking") #Window title
#====================================lables==========================================
===
l1=Label(window,text="Surname")
l1.grid(row=0,column=0)
```

```
l2=Label(window,text="Num of People")
l2.grid(row=0,column=2)

l3=Label(window,text="Date")
l3.grid(row=1,column=0)

l4=Label(window,text="Time")
l4.grid(row=1,column=2)
#====================================entry
boxes==============================================
surname_text=StringVar()
e1=Entry(window,textvariable=surname_text)
e1.grid(row=0,column=1)

table_num_text=StringVar()
e2=Entry(window,textvariable=table_num_text)
e2.grid(row=0,column=3)

date_text=StringVar()
e3=Entry(window,textvariable=date_text)
e3.grid(row=1,column=1)

time_text=StringVar()
e4=Entry(window,textvariable=time_text)
e4.grid(row=1,column=3)
#====================================list
box==============================================
list1=Listbox(window, height=6,width=35)
list1.grid(row=2,column=0,rowspan=6,columnspan=2)
#====================================scrollbar========================================
======
sb1=Scrollbar(window)
sb1.grid(row=2,column=2,rowspan=6)

list1.configure(yscrollcommand=sb1.set)
sb1.configure(command=list1.yview)

list1.bind('<<ListboxSelect>>',get_selected_row)
#====================================Buttons==========================================
====
b1=Button(window,text="View all Bookings", width=12,command=view_command)
b1.grid(row=2,column=3)

b2=Button(window,text="Search Bookings", width=12,command=search_command)
b2.grid(row=3,column=3)

b3=Button(window,text="Add Booking", width=12,command=add_command)
b3.grid(row=4,column=3)

b4=Button(window,text="Update Booking", width=12,command=update_command)
b4.grid(row=5,column=3)

b5=Button(window,text="Delete Booking", width=12,command=delete_command)
b5.grid(row=6,column=3)

b6=Button(window,text="Close Bookings", width=12,command=Clo_Open)
b6.grid(row=7,column=3)


window.mainloop()
```

Bookings_Back.py Code:

```
import sqlite3
#====================================Connect
method==============================================
def connect():
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS book (id INTEGER PRIMARY KEY, surname
text, table_num integer, date integer, time integer)")
    conn.commit()
    conn.close()
 #====================================Insert
method==============================================
def insert(surname,table_num,date,time):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("INSERT INTO book VALUES
(NULL,?,?,?,?)",(surname,table_num,date,time))
    conn.commit()
    conn.close()
    view()
#====================================View
method==============================================
def view():
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book")
    rows=cur.fetchall()
    conn.close()
    return rows
#====================================Search
method==============================================
def search(surname="",table_num="",date="",time=""):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book WHERE surname=? OR table_num=? OR date=? OR
time=?", (surname,table_num,date,time))
    rows=cur.fetchall()
    conn.close()
    return rows
#====================================Dellete
method==============================================
def delete(id):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("DELETE FROM book WHERE id=?",(id,))
    conn.commit()
    conn.close()
#====================================Update
method==============================================
def update(id,surname,table_num,date,time):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("UPDATE book SET surname=?, table_num=?, date=?, time=? WHERE
id=?",(surname,table_num,date,time,id))
    conn.commit()
    conn.close()

connect()
```

Make Orders Window

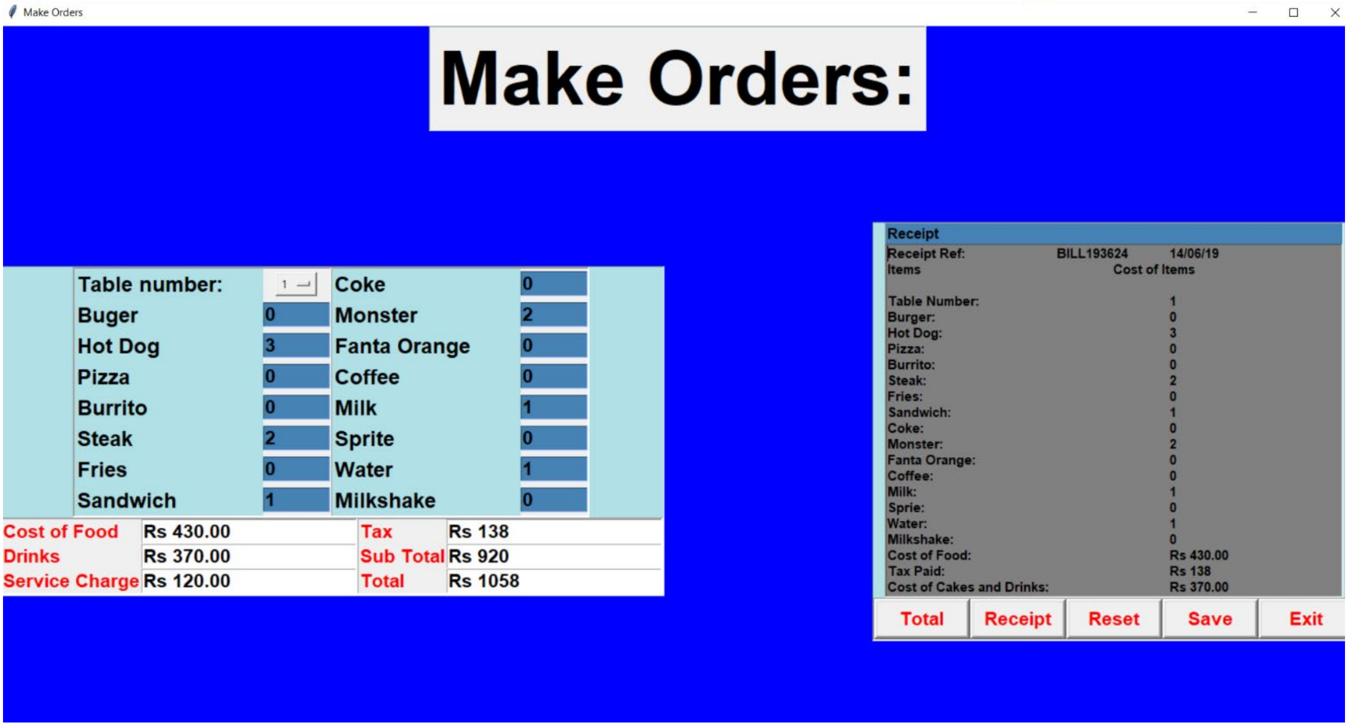


Table number:	1	Coke	0
Buger	0	Monster	2
Hot Dog	3	Fanta Orange	0
Pizza	0	Coffee	0
Burrito	0	Milk	1
Steak	2	Sprite	0
Fries	0	Water	1
Sandwich	1	Milkshake	0

Cost of Food	Rs 430.00	Tax	Rs 138
Drinks	Rs 370.00	Sub Total	Rs 920
Service Charge	Rs 120.00	Total	Rs 1058

```
Receipt
Receipt Ref:                BILL193624    14/06/19
Items                               Cost of Items

Table Number:                          1
Burger:                                0
Hot Dog:                               3
Pizza:                                 0
Burrito:                               0
Steak:                                 2
Fries:                                 0
Sandwich:                              1
Coke:                                  0
Monster:                               2
Fanta Orange:                          0
Coffee:                                0
Milk:                                  1
Sprie:                                 0
Water:                                 1
Milkshake:                             0
Cost of Food:                          Rs 430.00
Tax Paid:                              Rs 138
Cost of Cakes and Drinks:              Rs 370.00
```

Make_Orders.py Code:

```
from tkinter import *

import random

import time

from tkinter import messagebox

from tkinter import ttk


root = Tk()


root.geometry("1600x800+0+0")      #Size Of Window


root.title("Make Orders")     #Title Of Window


root.configure(background='blue') #Background of Window

#==================================================Size of sections in window========================================================

Tops = Frame(root, width=1250, height=90, bd=1, relief="sunken")

Tops.pack(side=TOP)
```

```
f1 = Frame(root, width=750, height=750, bd=1, relief="sunken")
f1.pack(side=LEFT)

f2 = Frame(root, width=400, height=950, bd=1, relief="sunken")
f2.pack(side=RIGHT)

f1a = Frame(f1, width=900, height=330, bd=1, relief="sunken")
f1a.pack(side=TOP)

f2a = Frame(f1, width=850, height=520, bd=1, relief="sunken")
f2a.pack(fill=BOTH, side=BOTTOM)

ft2 = Frame(f2, width=440, height=450, bd=1, relief="sunken")
ft2.pack(side=TOP)

fb2 = Frame(f2, width=440, height=750, bd=1, relief="sunken")
fb2.pack(side=BOTTOM)

f1aa = Frame(f1a, width=400, height=330, bd=1, relief="sunken")
f1aa.pack(side=LEFT)

f1ab = Frame(f1a, width=400, height=330, bd=1, relief="sunken")
f1ab.pack(side=RIGHT)

f2aa = Frame(f2a, width=450, height=330, bd=1, relief="sunken")
f2aa.pack(side=LEFT)

f2ab = Frame(f2a, width=450, height=330, bd=1, relief="sunken")
f2ab.pack(side=RIGHT)
```

```
# =========================== Background color
================================================

Tops.configure(background='powder blue')

f1.configure(background='powder blue')

f2.configure(background='powder blue')

# ===============================Main Lable================================

lblInfo = Label(Tops, font=('sans', 64, 'bold'), text="Make Orders:", bd=9)

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



# ================================= Cost Method
===================================

def CostofItems():

    flag = False

    while flag == False:

        try:

            Item1 = float(E_Table_Num.get())

            Item2 = float(E_Burger.get())

            Item3 = float(E_Hot_Dog.get())

            Item4 = float(E_Pizza.get())

            Item5 = float(E_Burrito.get())

            Item6 = float(E_Steak.get())

            Item7 = float(E_Fries.get())

            Item8 = float(E_Sandwich.get())


            Item9 = float(E_Coke.get())

            Item10 = float(E_Monster.get())

            Item11 = float(E_Fanta_Orange.get())
            Item12 = float(E_Coffee.get())
            Item13 = float(E_Milk.get())
            Item14 = float(E_Sprite.get())
            Item15 = float(E_Water.get())
            Item16 = float(E_Milkshake.get())

            PriceofFood = (Item2 * 50) + (Item3 * 23) + (Item4 *25) + (Item5 * 65) + (Item6 * 95) +\
                    (Item7 * 20) + (Item8 * 35)

            PriceofCakesandDrinks =(Item9 * 12) + (Item10 * 14) + (Item11 * 12) + (Item12 *25) + (Item13
* 10) + \
                    (Item14 * 12) + (Item15 * 5) + (Item16 * 25)

            FoodPrice = "Rs "+ str('%.2f' % PriceofFood)
            DrinksPrice = "Rs "+ str("%.2f" % PriceofDrinks)
            CostofFood.set(FoodPrice)
            CostofDrinks.set(DrinksPrice)
            SC =  ((PriceofFood + PriceofDrinks)*0.15)
            print("Rs " + str("%.2f" % SC))
            ServiceCharge.set("Rs "+ str("%.2f" % SC))

            SubTotalofITEMS = "Rs "+ str(round(PriceofFood + PriceofDrinks + SC))
            SubTotal.set(SubTotalofITEMS)

            Tax = "Rs "+ str(round((PriceofFood + PriceofDrinks + SC)*0.15))
            Tax_Flo = float(round(PriceofDrinks + PriceofFood + SC)*0.15)
            PaidTax.set(Tax)
            TT = (PriceofFood + PriceofDrinks + SC)
            TC = "Rs "+ str(round((PriceofFood + PriceofDrinks + SC) + Tax_Flo))
            print((TC))
            TotalCost.set(TC)
```

```
            flag = True
        except :
                print("Please enter numbers only !")
                Reset()


# ===================================== Exit Window
===============================================



def qExit():


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


# ============================ Save Files  =========================================
def Save():
    file = open("myfile.txt", "x")
    file = open("myfile.txt", "w")
    file.write('Receipt Ref:\t\t\t'+ Receipt_Ref.get() + '\t\t' + DateofOrder.get()+"\n")
    file.write('Items\t\t\t\t' + "Cost of Items \n\n")
    file.write('Table Number:\t\t\t\t\t' + E_Table_Num.get() + "\n")
    file.write('Lunch Meal: \t\t\t\t\t' + E_Burger.get() + "\n")
    file.write('Burger: \t\t\t\t\t' + E_Hot_Dog.get() + "\n")
    file.write('Pizza: \t\t\t\t\t' + E_Pizza.get() + "\n")
    file.write('Cheese Burger: \t\t\t\t\t' + E_Burrito.get() + "\n")
    file.write('Mac Nuggets: \t\t\t\t\t' + E_Steak.get() + "\n")
    file.write('Mac Puff: \t\t\t\t\t' + E_Fries.get() + "\n")
    file.write('Chicken Wings: \t\t\t\t\t' + E_Sandwich.get() + "\n")
    file.write('Coffee Cake: \t\t\t\t\t' + E_Coke.get() + "\n")
    file.write('Red Valet Cake: \t\t\t\t\t' + E_Monster.get() + "\n")
    file.write('Black Forest Cake: \t\t\t\t\t' + E_Fanta_Orange.get() + "\n")
    file.write('Boston Cream Cake: \t\t\t\t\t' + E_Coffee.get() + "\n")
    file.write('Latte: \t\t\t\t\t' + E_Milk.get() + "\n")
    file.write('Coke: \t\t\t\t\t' + E_Sprite.get() + "\n")
    file.write('Pepsi: \t\t\t\t\t' + E_Water.get() + "\n")
    file.write('Cappuccino: \t\t\t\t\t' + E_Milkshake.get() + "\n")
    file.write('Cost of Food: \t\t\t\t\t' + CostofFood.get() + "\t\t\tTax Paid:\t\t\t\t\t" + PaidTax.get() +
"\n")
    file.write('Cost of Drinks: \t\t\t\t\t' + CostofDrinks.get() + "\t\t\tSub Total:\t\t\t\t\t" +
                  SubTotal.get() + "\n")
    file.write('Service Charge: \t\t\t\t\t' + ServiceCharge.get() + "\t\t\tTotal Cost:\t\t\t\t\t" +
TotalCost.get() + "\n")
    file.close()


# ===================================== Reset Method
========================================
def Reset():


    PaidTax.set("")
    SubTotal.set("")
    TotalCost.set("")
    CostofFood.set("")
    CostofDrinks.set("")
    ServiceCharge.set("")
    txtReceipt.delete("1.0", END)


    E_Table_Num.set("1")
    E_Burger.set("0")
    E_Hot_Dog.set("0")
```

```
E_Pizza.set("0")
E_Burrito.set("0")
E_Steak.set("0")
E_Fries.set("0")
E_Sandwich.set("0")

E_Coke.set("0")
E_Monster.set("0")
E_Fanta_Orange.set("0")
E_Coffee.set("0")
E_Milk.set("0")
E_Sprite.set("0")
E_Water.set("0")
E_Milkshake.set("0")

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")
```

```
# ================================== Receipt Method =====================

def Receipt():
    txtReceipt.delete("1.0", END)
    x = random.randint(1000, 500890)
    randomRef = str(x)
    Receipt_Ref.set("BILL" + randomRef)
    txtReceipt.insert(END, 'Receipt Ref:\t\t\t'+ Receipt_Ref.get() + '\t\t' + DateofOrder.get()+"\n")
    txtReceipt.insert(END, 'Items\t\t\t\t' + "Cost of Items \n\n")
    txtReceipt.insert(END, 'Table Number:\t\t\t\t\t' + E_Table_Num.get() + "\n")
    txtReceipt.insert(END, 'Burger: \t\t\t\t\t' + E_Burger.get() + "\n")
    txtReceipt.insert(END, 'Hot Dog: \t\t\t\t\t' + E_Hot_Dog.get() + "\n")
    txtReceipt.insert(END, 'Pizza: \t\t\t\t\t' + E_Pizza.get() + "\n")
    txtReceipt.insert(END, 'Burrito: \t\t\t\t\t' + E_Burrito.get() + "\n")
    txtReceipt.insert(END, 'Steak: \t\t\t\t\t' + E_Steak.get() + "\n")
    txtReceipt.insert(END, 'Fries: \t\t\t\t\t' + E_Fries.get() + "\n")
    txtReceipt.insert(END, 'Sandwich: \t\t\t\t\t' + E_Sandwich.get() + "\n")
    txtReceipt.insert(END, 'Coke: \t\t\t\t\t' + E_Coke.get() + "\n")
    txtReceipt.insert(END, 'Monster: \t\t\t\t\t' + E_Monster.get() + "\n")
    txtReceipt.insert(END, 'Fanta Orange: \t\t\t\t\t' + E_Fanta_Orange.get() + "\n")
    txtReceipt.insert(END, 'Coffee: \t\t\t\t\t' + E_Coffee.get() + "\n")
    txtReceipt.insert(END, 'Milk: \t\t\t\t\t' + E_Milk.get() + "\n")
    txtReceipt.insert(END, 'Sprie: \t\t\t\t\t' + E_Sprite.get() + "\n")
    txtReceipt.insert(END, 'Water: \t\t\t\t\t' + E_Water.get() + "\n")
    txtReceipt.insert(END, 'Milkshake: \t\t\t\t\t' + E_Milkshake.get() + "\n")
    txtReceipt.insert(END, 'Cost of Food: \t\t\t\t\t' + CostofFood.get() + "\t\t\tTax Paid:\t\t\t\t\t" +
PaidTax.get() + "\n")
    txtReceipt.insert(END, 'Cost of Drinks: \t\t\t\t\t' + CostofDrinks.get() + "\t\t\tSub Total:\t\t\t\t\t" +
                      SubTotal.get() + "\n")
    txtReceipt.insert(END, 'Service Charge: \t\t\t\t\t' + ServiceCharge.get() + "\t\t\tTotal
Cost:\t\t\t\t\t" + TotalCost.get() + "\n")




# ============================== Inisiolizing 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()


DateofOrder = StringVar()

Receipt_Ref = StringVar()

PaidTax = StringVar()

SubTotal = StringVar()

TotalCost = StringVar()

CostofDrinks = StringVar()
```

```
CostofFood = StringVar()
ServiceCharge = StringVar()

E_Table_Num = StringVar()
E_Burger = StringVar()
E_Hot_Dog = StringVar()
E_Pizza = StringVar()
E_Burrito = StringVar()
E_Steak = StringVar()
E_Fries = StringVar()
E_Sandwich = StringVar()


E_Coke = StringVar()
E_Monster = StringVar()
E_Fanta_Orange = StringVar()
E_Coffee = StringVar()
E_Milk = StringVar()
E_Sprite = StringVar()
E_Water = StringVar()
E_Milkshake = StringVar()

E_Table_Num.set("0")
E_Burger.set("0")
E_Hot_Dog.set("0")
E_Pizza.set("0")
E_Burrito.set("0")
E_Steak.set("0")
E_Fries.set("0")
E_Sandwich.set("0")
```

```
E_Coke.set("0")

E_Monster.set("0")

E_Fanta_Orange.set("0")

E_Coffee.set("0")

E_Milk.set("0")

E_Sprite.set("0")

E_Water.set("0")

E_Milkshake.set("0")


# =========================Declaring
Date==========================================


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


# ============= Food  lables ===================================================


Table_Num = Label(f1aa, bg="powder blue", font=('sans',18, 'bold'), text="Table number:
\t").grid(row=0, sticky=W)


Burger = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Buger          \t").grid(row=1,
sticky=W)


Hot_Dog = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Hot Dog
\t").grid(row=2, sticky=W)


Pizza = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Pizza          \t").grid(row=3,
sticky=W)


Burrito = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Burrito
\t").grid(row=4, sticky=W)


Steak = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Steak          \t").grid(row=5,
sticky=W)
```

```
Fries = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Fries            \t").grid(row=6, sticky=W)

Sandwich = Label(f1aa, bg="powder blue", font=('sans', 18, 'bold'), text="Sandwich \t").grid(row=7, sticky=W)

# ======================== Drinks  lables ==============================================

Coke = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Coke         \t").grid(row=0, sticky=W)

Monster = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Monster      \t").grid(row=1, sticky=W)

Fanta_Orange = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Fanta Orange \t").grid(row=2, sticky=W)

Coffee = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Coffee      \t").grid(row=3, sticky=W)

Milk = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Milk       \t").grid(row=4, sticky=W)

Sprite = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Sprite      \t").grid(row=5, sticky=W)

Water = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Water         \t").grid(row=6, sticky=W)

Milkshake = Label(f1ab, bg="powder blue", font=('sans', 18, 'bold'), text="Milkshake \t").grid(row=7, sticky=W)
```

```
# ======================================  Table Number assignment
====================================================


# Create a Tkinter variable

tkvar = StringVar(root)


tkvar.set('1') # set the default option

from Login_window.register import username_entery


if username_entry == Chris:

   choices = { '1','2','3'}

   txtFries = OptionMenu( f1aa, tkvar, *choices)

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

elif username_entry == Joana:

    choices = {'4','5', '6'}

    txtFries = OptionMenu( f1aa, tkvar, *choices)

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

else:

    choices = {'7','8', '9'}

    txtFries = OptionMenu( f1aa, tkvar, *choices)

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


# ====================================== Food Widgets
====================================================


txtBurger = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
textvariable=E_Burger,

                  state=NORMAL)

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

txtHot_Dog = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
textvariable=E_Hot_Dog,

                   state=NORMAL)
```

```
txtHot_Dog.grid(row=2, column=1)

txtPizza = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
textvariable=E_Pizza,

                  state=NORMAL)

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

txtBurrito = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
textvariable=E_Burrito,

                 state=NORMAL)

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

txtSteak = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
textvariable=E_Steak

                    , state=NORMAL)

txtSteak.grid(row=5, column=1)

txtFries = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',

                     textvariable=E_Fries, state=NORMAL)

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

txtSandwich = Entry(f1aa, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',

                     textvariable=E_Sandwich, state=NORMAL)

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


# ===================================== Drink Widgets
==================================================


txtCoke = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',

                  textvariable=E_Coke, state=NORMAL)

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

txtMonster = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',

                    textvariable=E_Monster, state=NORMAL)

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

txtFanta_Orange = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',

                       textvariable=E_Fanta_Orange, state=NORMAL)

txtFanta_Orange.grid(row=2, column=1)
```

```
txtCoffee = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
                          textvariable=E_Coffee, state=NORMAL)
txtCoffee.grid(row=3, column=1)
txtMilk = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
                            textvariable=E_Milk, state=NORMAL)
txtMilk.grid(row=4, column=1)
txtSprite = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
                             textvariable=E_Sprite, state=NORMAL)
txtSprite.grid(row=5, column=1)
txtWater = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
                          textvariable=E_Water, state=NORMAL)
txtWater.grid(row=6, column=1)
txtMilkshake = Entry(f1ab, font=('sans', 16, 'bold'), bd=1, bg="steel blue", width=6, justify='left',
                        textvariable=E_Milkshake, state=NORMAL)
txtMilkshake.grid(row=7, column=1)


# =========================== Receipt Box
====================================================


lblReceipt = Label(ft2, font=('sans', 12, 'bold'), text="Receipt
", bd=1, bg="steel blue", anchor='w')
lblReceipt.grid(row=0, column=0, sticky=W)
txtReceipt = Text(ft2, width=59, height=22, bg="grey", bd=1, font=('sans', 11, 'bold'))
txtReceipt.grid(row=1, column=0, sticky=W+E)


# ======================= left side outputs
==================================================


lblCostofFood = Label(f2aa, font=('sans', 16, 'bold'), fg="red", text="Cost of Food", bd=1)
lblCostofFood.grid(row=2, column=0, sticky=W)
txtCostofFood = Entry(f2aa, font=('sans', 16, 'bold'), bd=1, justify="left", textvariable=CostofFood)
txtCostofFood.grid(row=2, column=1)
```

```
lblCostofDrinks = Label(f2aa, font=('sans', 16, 'bold'), fg="red", text="Drinks", bd=1)

lblCostofDrinks.grid(row=3, column=0, sticky=W)

txtCostofDrinks = Entry(f2aa, font=('sans', 16, 'bold'), bd=1, justify="left", textvariable=CostofDrinks)

txtCostofDrinks.grid(row=3, column=1, sticky=W)


lblServiceCharge = Label(f2aa, font=('sans', 16, 'bold'), fg="red", text="Service Charge", bd=1)

lblServiceCharge.grid(row=4, column=0, sticky=W)

txtServiceCharge = Entry(f2aa, font=('sans', 16, 'bold'), bd=1, justify="left", textvariable=ServiceCharge)

txtServiceCharge.grid(row=4, column=1, sticky=W)


# ===================Right side outputs===================================================


lblPaidTax = Label(f2ab, font=('sans', 16, 'bold'),fg="red", text="Tax", bd=1)

lblPaidTax.grid(row=2, column=0, sticky=W)

txtPaidTax = Entry(f2ab, font=('sans', 16, 'bold'),  bd=1, justify="left", textvariable=PaidTax)

txtPaidTax.grid(row=2, column=1, sticky=W)


lblSubTotal = Label(f2ab, font=('sans', 16, 'bold'), fg="red", text="Sub Total", bd=1)

lblSubTotal.grid(row=3, column=0, sticky=W)

txtSubTotal = Entry(f2ab, font=('sans', 16, 'bold'), bd=1, justify="left", textvariable=SubTotal)

txtSubTotal.grid(row=3, column=1, sticky=W)


lblTotalCost = Label(f2ab, font=('sans', 16, 'bold'), fg="red", text="Total", bd=1)

lblTotalCost.grid(row=4, column=0, sticky=W)

txtTotalCost = Entry(f2ab, font=('sans', 16, 'bold'), bd=1, justify="left", textvariable=TotalCost)

txtTotalCost.grid(row=4, column=1, sticky=W)



# ==========Buttons===================================================
btnTotal = Button(fb2, padx=16, pady=1, bd=4, fg="red", font=('sans', 16, 'bold'), width=5,
                text="Total", command=CostofItems).grid(row=0, column=0, sticky=W+E+N+S)
btnReceipt = Button(fb2, padx=16, pady=1, bd=4, fg="red", font=('sans', 16, 'bold'), width=5,
                  text="Receipt", command=Receipt).grid(row=0, column=1, sticky=W+E+N+S)
btnReset = Button(fb2, padx=16, pady=1, bd=4, fg="red", font=('sans', 16, 'bold'), width=5,
                text="Reset", command=Reset).grid(row=0, column=2, sticky=W+E+N+S)
btnSave = Button(fb2, padx=16, pady=1, bd=4, fg="red", font=('sans', 16, 'bold'), width=5,
                text="Save", command=Save).grid(row=0, column=3, sticky=W+E+N+S)
btnExit = Button(fb2, padx=16, pady=1, bd=4, fg="red", font=('sans', 16, 'bold'), width=5,
               text="Exit", command=qExit).grid(row=0, column=4, sticky=W+E+N+S)




root.mainloop()
```