Final Report

Restaurant Bill Management System with GST Billing

As a Field work for Course

Python Programming (INT 213)

By

Sr No.	Registration No.	Name	Roll No.	Total Marks	Marks Obtained	Signature
1	12007298	YUVRAJ PANDEY	RK20BGA18	30		

Transforming Education Transforming India

Submitted To

Dr. Sagar Pande,

Professor

Lovely Professional University

Jalandhar, Punjab, India.

RESTAURANT BILL MANAGEMENT SYSTEM

16th NOVEMBER 2021

ABSTRACT: -

A simple project based on Restaurant/Cafe Billing System which uses Python Language with Tkinter Library for GUI. Following Python with Tkinter Library project contains the least, but important features. It has features that will allow all the users to interact in a way that the restaurant manager interacts with their customers regarding their billing payments. This system as well as the python application's concept is all clear, it's the same as real-life scenarios and well-implemented on it.

ACKNOWLEDGEMENT: -

I would like to thank my Professor Dr. Sagar Pande for his teaching and guidance on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

Contents

INTRODUCTION: -

Context

This project has been done as part of my course for the B. Tech (CSE) at Lovely Professional University. Supervised by Dr. Sagar Pande, I have two months to fulfill the requirements in order to succeed the module.

Moving on, this restaurant/cafe system project in Python focuses mainly on dealing with customer’s payment details with their respective food orders and amounts. Also, the system allows the selection of food and drink items for calculation and entering the quantities. In an overview of this app, the system user has to enter a certain quantity of particular food and drink item and generate the total cost. In addition, the system generates the total bill amount with GST. Besides, the system also generates a bill with a reference number. Additionally, the system also contains a mini calculator where the user can perform simple mathematics for calculation too. So, with it, this simple project can perform all the important tasks for calculations of the total bill amount of the customer.

Last but not least, a clean and simple GUI is presented with simple color combinations for a greater user experience while using this restaurant billing system project in Python. For its UI elements, a standard GUI library; Tkinter is on board. Presenting a new restaurant/cafe billing system in Python project which includes a user panel that contains all the essential features to follow up, and a knowledgeable resource for learning purposes.

LIBRARIES: -

Python GUI – tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications.
Importing tkinter is same as importing any other module in the Python code.

import tkinter

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1.Tk(screenName=None, baseName=None, className='Tk', useTk=1): To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

m=tkinter.Tk() where m is the name of the main window object

2.mainloop(): There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

m.mainloop()

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1.pack () method: It organizes the widgets in blocks before placing in the parent widget.
2.grid () method: It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3.place () method: It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:
1.Button: To add a button in your application, this widget is used.
The general syntax is:
w=Button(master, option=value)
master is the parameter used to represent the parent window. There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas.

Some of them are listed below.

activebackground: to set the background color when button is under the cursor.
activeforeground: to set the foreground color when button is under the cursor.
bg: to set he normal background color.
command: to call a function.
font: to set the font on the button label.
image: to set the image on the button.
width: to set the width of the button.
height: to set the height of the button.

2.Canvas: It is used to draw pictures and other complex layout like graphics, text and widgets. The general syntax is:
w = Canvas(master, option=value)
master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas.
Some of them are listed below.

bd: to set the border width in pixels.
bg: to set the normal background color.
cursor: to set the cursor used in the canvas.
highlightcolor: to set the color shown in the focus highlight.
width: to set the width of the widget.
height: to set the height of the widget.

3.Frame: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets. The general syntax is:
w = Frame(master, option=value)
master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas.
Some of them are listed below.

highlightcolor: To set the color of the focus highlight when widget has to be focused.
bd: to set the border width in pixels.
bg: to set the normal background color.
cursor: to set the cursor used.
width: to set the width of the widget.
height: to set the height of the widget.

4.Label: It refers to the display box where you can put any text or image which can be updated any time as per the code. The general syntax is:
w=Label(master, option=value)
master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas.
Some of them are listed below.

bg: to set the normal background color.
command: to call a function.
font: to set the font on the button label.
image: to set the image on the button.
width: to set the width of the button.
Height: to set the height of the button.

5. Menu: It is used to create all kinds of menus used by the application. The general syntax is:
w = Menu(master, option=value)
master is the parameter used to represent the parent window.
There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas.
Some of them are listed below.

title: To set the title of the widget.
activebackground: to set the background color when widget is under the cursor.
activeforeground: to set the foreground color when widget is under the cursor.
bg: to set he normal background color.
command: to call a function.
font: to set the font on the button label.
image: to set the image on the widget.

random — Generate pseudo-random numbers:

This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement. To use functions defined in the module, we need to import the module first. Here's how:

import random

On the real line, there are functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, the von Mises distribution is available.
Almost all module functions depend on the basic function random(), which generates a random float uniformly in the semi-open range [0.0, 1.0). Python uses the Mersenne Twister as the core generator. It produces 53-bit precision floats and has a period of 2**19937-1. The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for

cryptographic purposes.
The functions supplied by this module are actually bound methods of a hidden instance of the random.Random class. You can instantiate your own instances of Random to get generators that don’t share state.
Class Random can also be subclassed if you want to use a different basic generator of your own devising: in that case, override the random(), seed(), getstate(), and setstate() methods. Optionally, a new generator can supply a getrandbits() method — this allows randrange() to produce selections over an arbitrarily large range.

time — Time access and conversions

Python has a module named time to handle time-related tasks. To use functions defined in the module, we need to import the module first. Here's how:

```
import time
```

Here are commonly used time-related functions.

Python time.time()
The time() function returns the number of seconds passed since epoch.
For Unix system, January 1, 1970, 00:00:00 at UTC is epoch (the point where time begins).

```
import time
seconds = time.time()
print("Seconds since epoch =", seconds)
```

Python time.ctime()
The time.ctime() function takes seconds passed since epoch as an argument and returns a string representing local time.

```
import time
local_time = time.ctime(seconds)
print("Local time:", local_time)
```

If you run the program, the output will be something like:

```
Local time: Thu Dec 27 15:49:29 2018
```

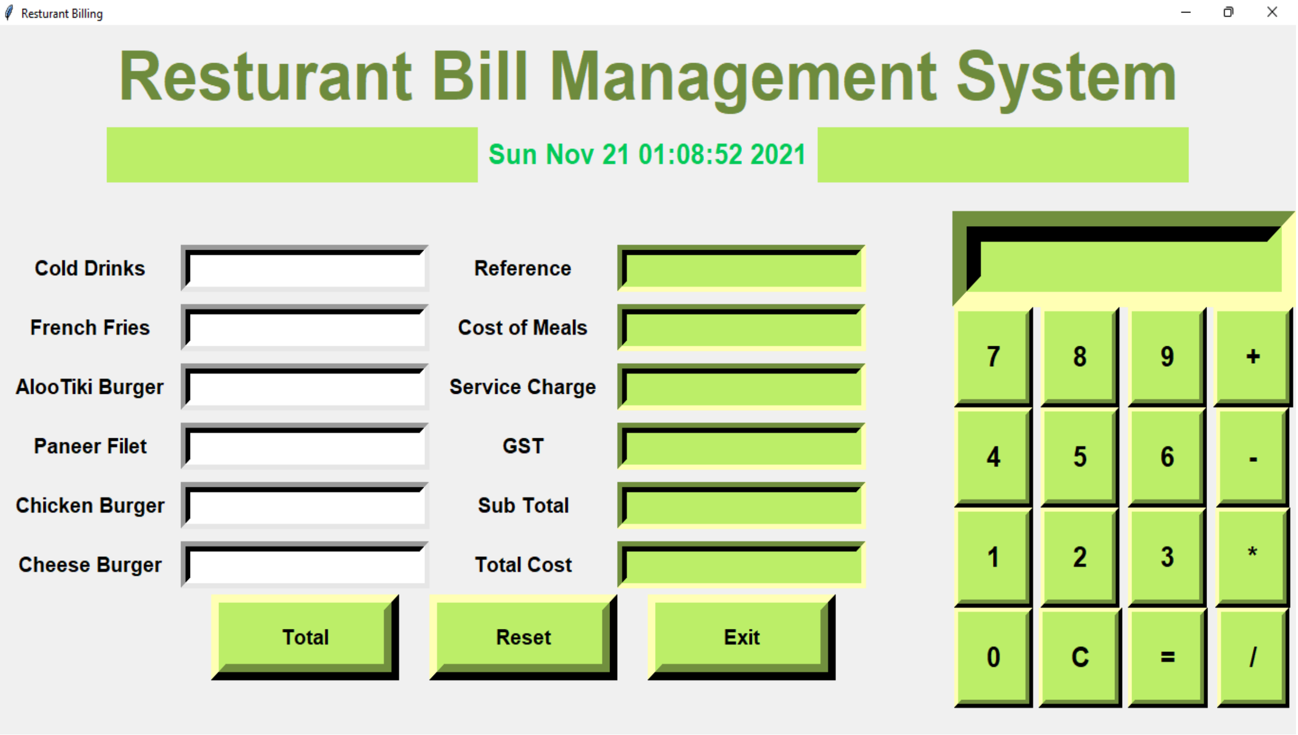
Python time.sleep()

The sleep() function suspends (delays) execution of the current thread for the given number of seconds.

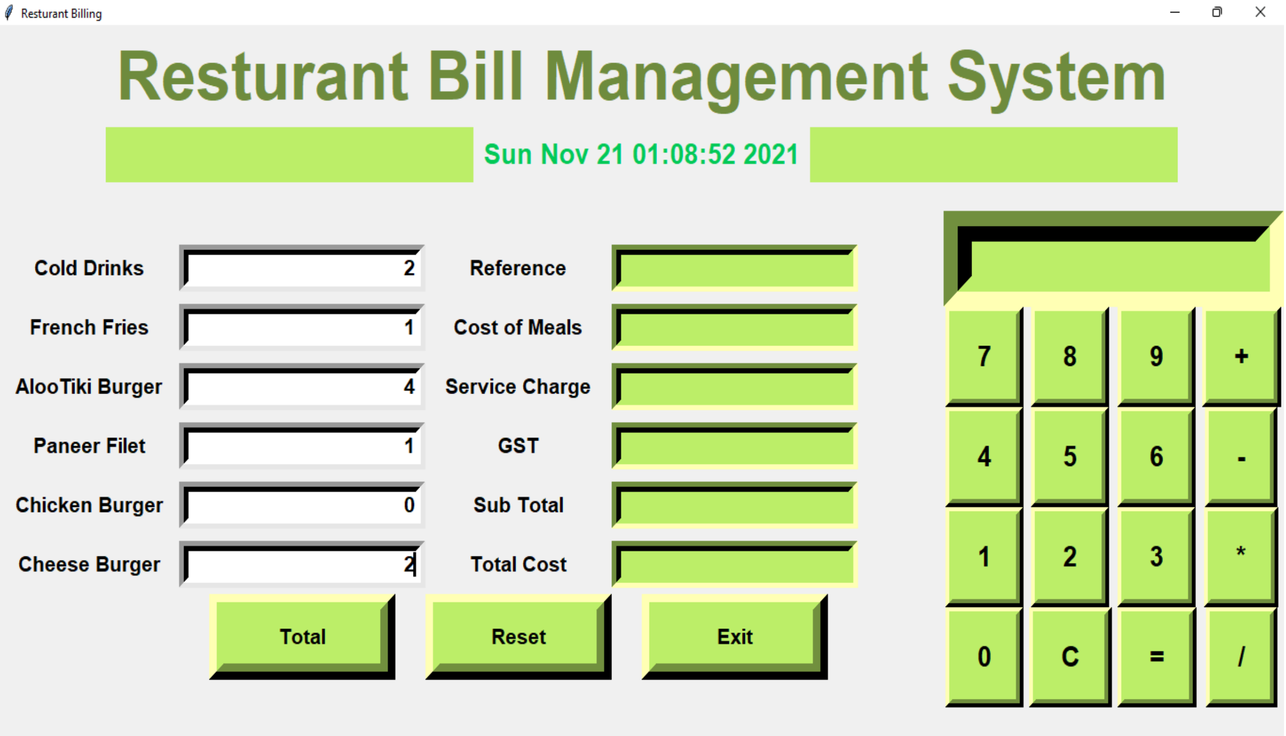
```
import time
print("This is printed immediately.")
time.sleep(2.4)
print("This is printed after 2.4 seconds.")
```

SCREENSHOTS: -

1. Main Page:

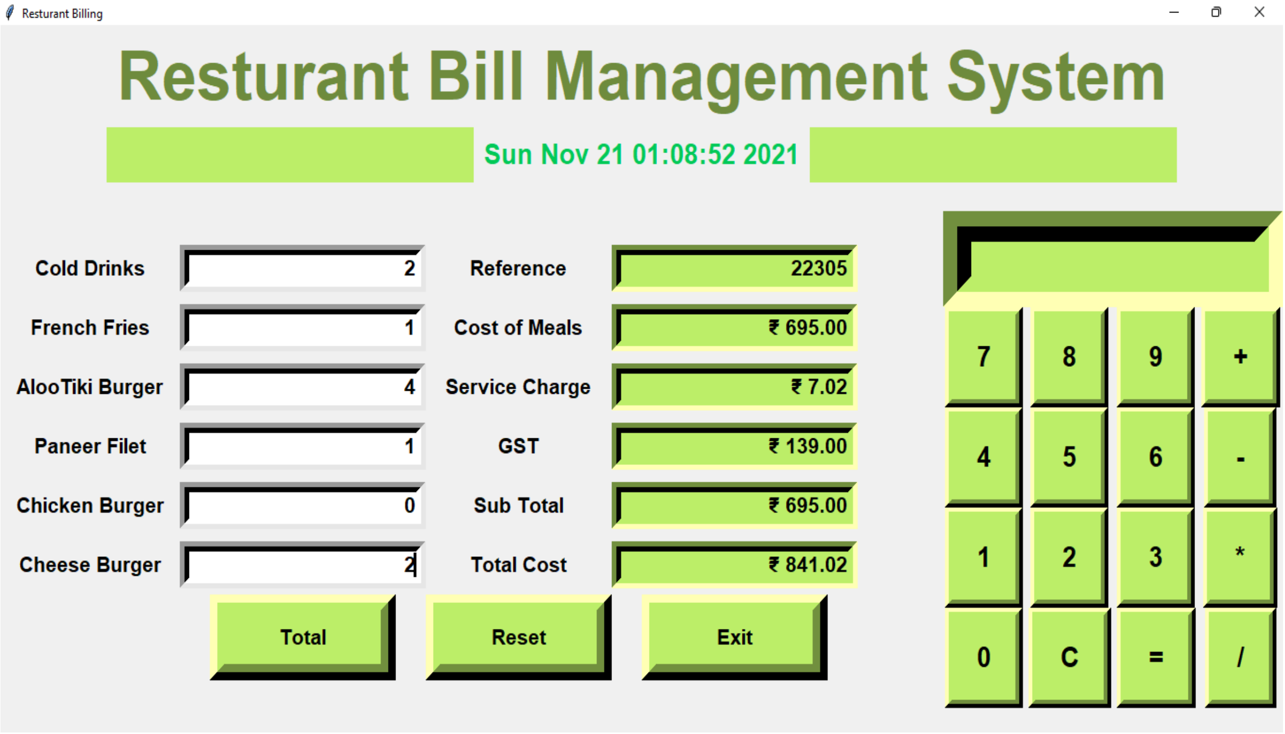


2.After entering quantity of items:



Cold Drinks	2
French Fries	1
AlooTiki Burger	4
Paneer Filet	1
Chicken Burger	0
Cheese Burger	2

3.After clicking Total button for generating bill:



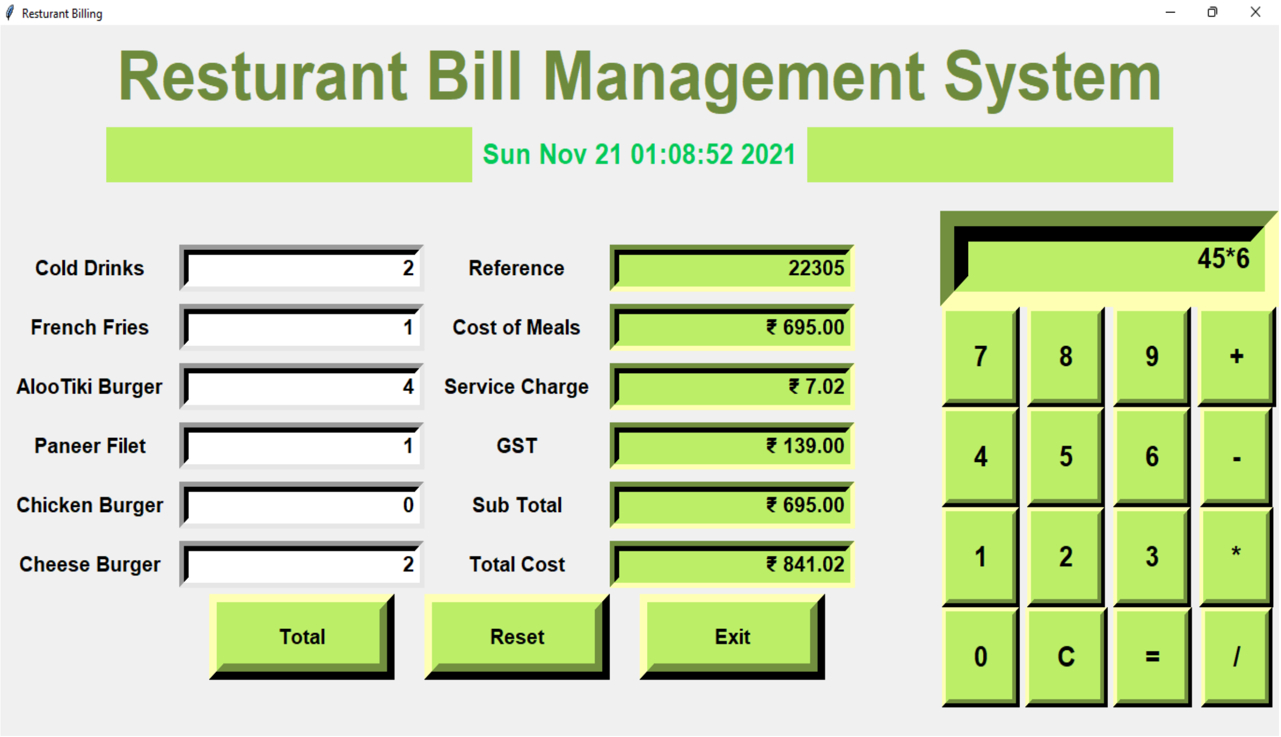
Resturant Billing

Resturant Bill Management System

Sun Nov 21 01:08:52 2021

Cold Drinks	2	Reference	22305
French Fries	1	Cost of Meals	₹ 695.00
AlooTiki Burger	4	Service Charge	₹ 7.02
Paneer Filet	1	GST	₹ 139.00
Chicken Burger	0	Sub Total	₹ 695.00
Cheese Burger	2	Total Cost	₹ 841.02

4.Performing calculation on calculator:



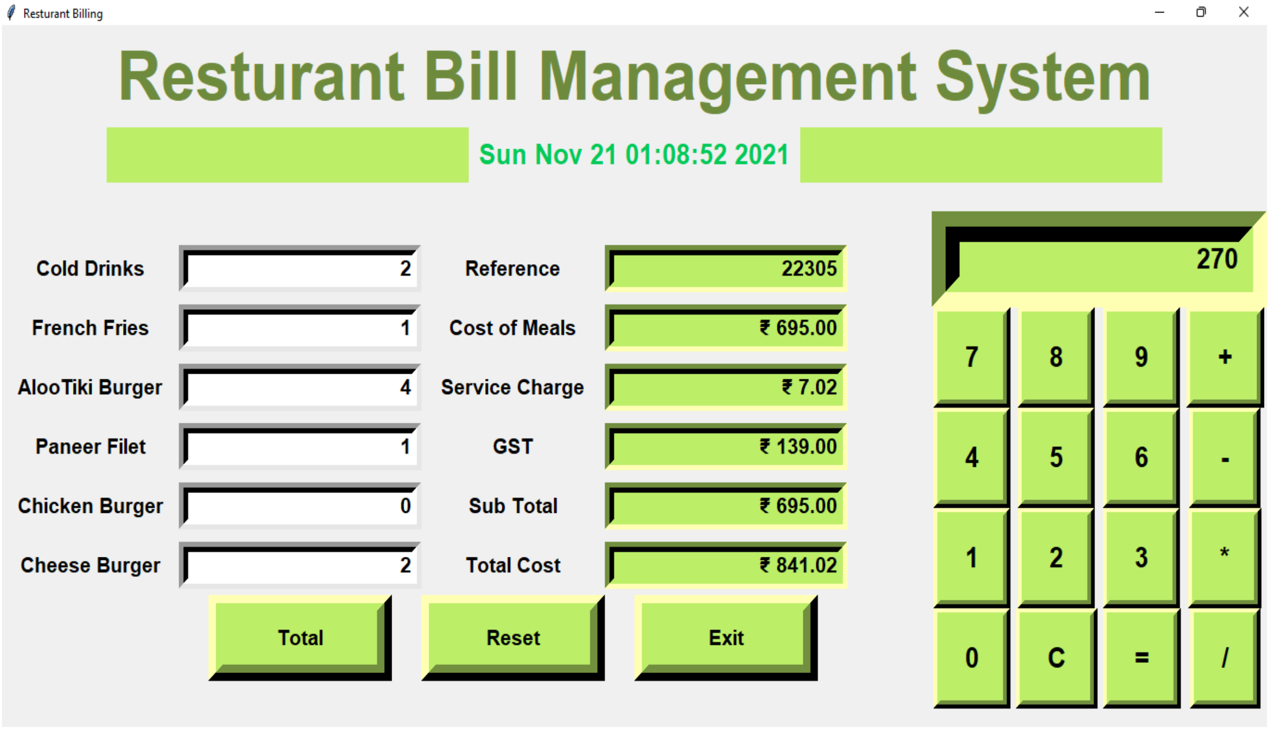
Resturant Billing

Resturant Bill Management System

Sun Nov 21 01:08:52 2021

45*6

Cold Drinks	2	Reference	22305
French Fries	1	Cost of Meals	₹ 695.00
AlooTiki Burger	4	Service Charge	₹ 7.02
Paneer Filet	1	GST	₹ 139.00
Chicken Burger	0	Sub Total	₹ 695.00
Cheese Burger	2	Total Cost	₹ 841.02



Resturant Billing

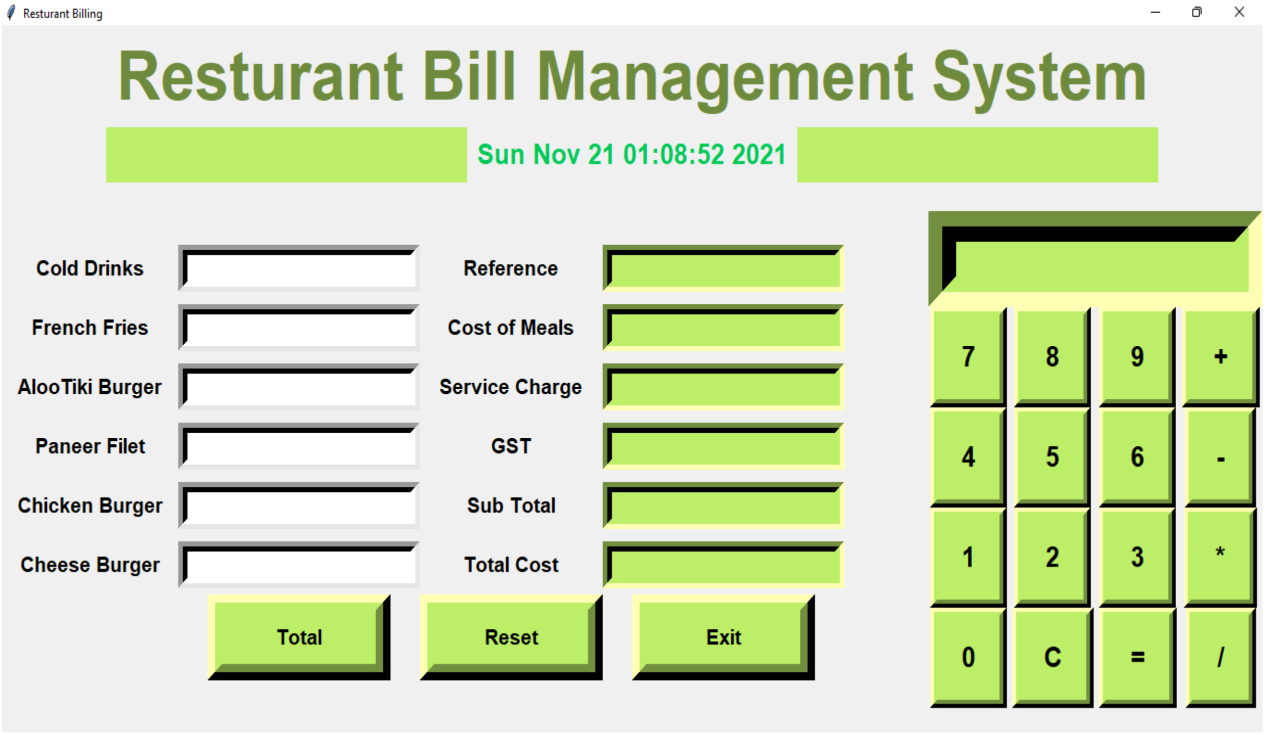
Resturant Bill Management System

Sun Nov 21 01:08:52 2021

Cold Drinks	2	Reference	22305
French Fries	1	Cost of Meals	₹ 695.00
AlooTiki Burger	4	Service Charge	₹ 7.02
Paneer Filet	1	GST	₹ 139.00
Chicken Burger	0	Sub Total	₹ 695.00
Cheese Burger	2	Total Cost	₹ 841.02

270

5.After clicking Reset button for next billing:

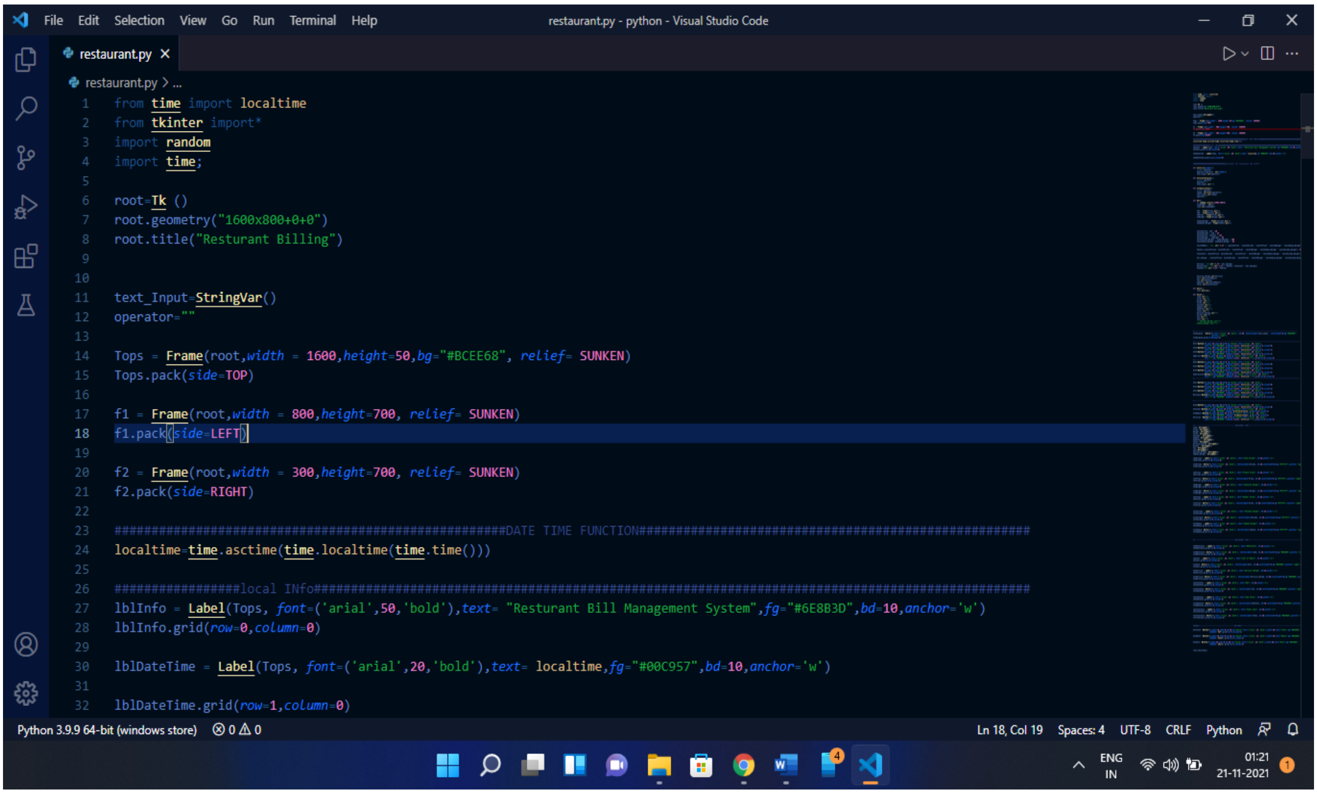


Resturant Billing

Resturant Bill Management System

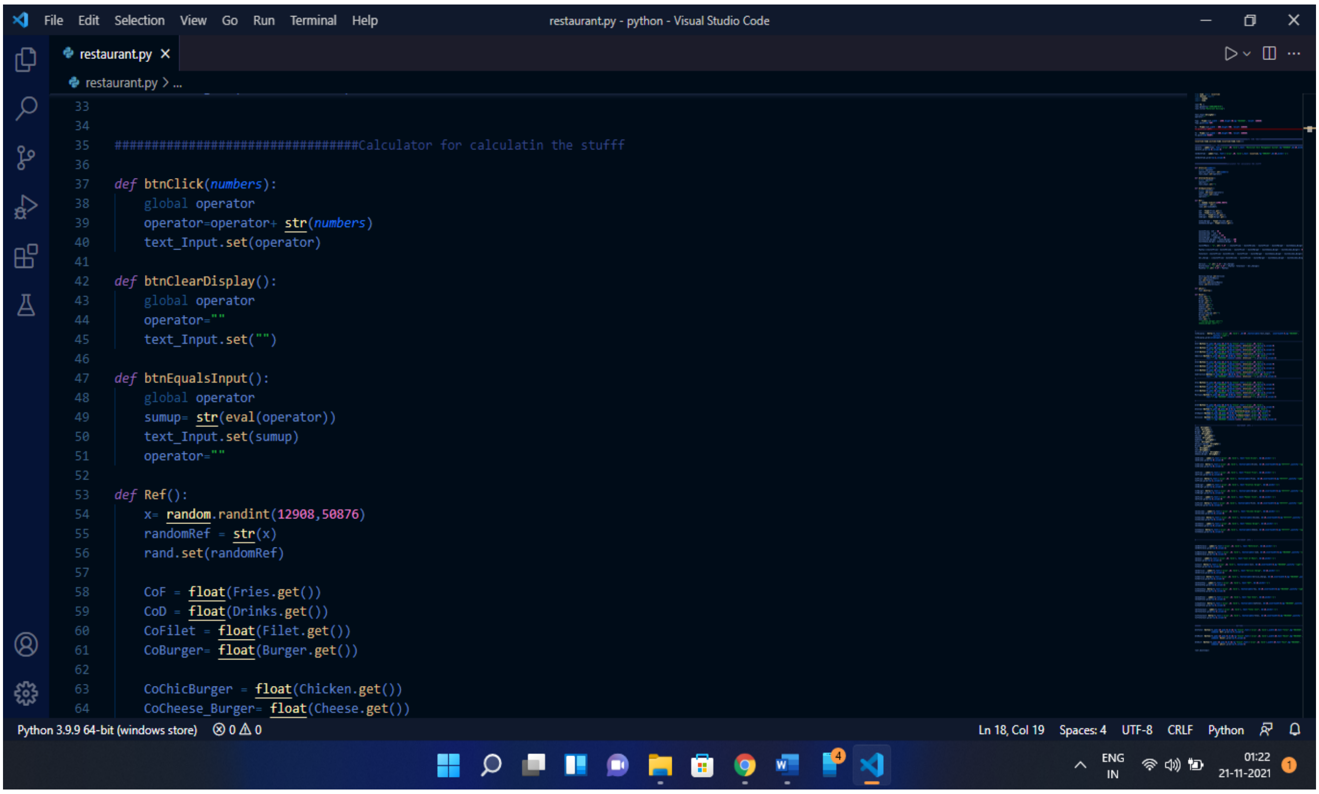
Sun Nov 21 01:08:52 2021

SOURCE CODE: -



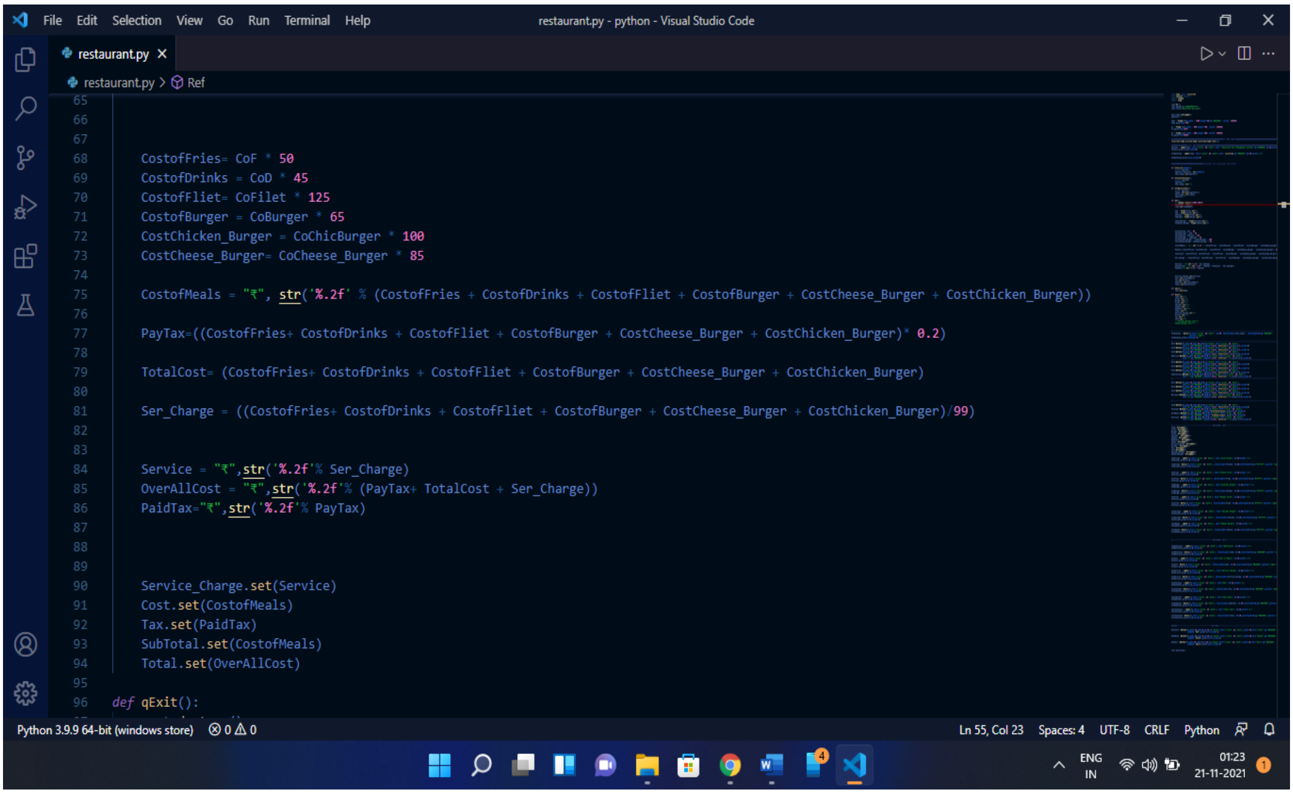
```python
from time import localtime
from tkinter import*
import random
import time;

root=Tk ()
root.geometry("1600x800+0+0")
root.title("Resturant Billing")


text_Input=StringVar()
operator=""

Tops = Frame(root,width = 1600,height=50,bg="#BCEE68", relief= SUNKEN)
Tops.pack(side=TOP)

f1 = Frame(root,width = 800,height=700, relief= SUNKEN)
f1.pack(side=LEFT)

f2 = Frame(root,width = 300,height=700, relief= SUNKEN)
f2.pack(side=RIGHT)

#####################################################DATE TIME FUNCTION#####################################################
localtime=time.asctime(time.localtime(time.time()))

#################local INfo#################################################################################################
lblInfo = Label(Tops, font=('arial',50,'bold'),text= "Resturant Bill Management System",fg="#6E8B3D",bd=10,anchor='w')
lblInfo.grid(row=0,column=0)

lblDateTime = Label(Tops, font=('arial',20,'bold'),text= localtime,fg="#00C957",bd=10,anchor='w')

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

File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > ...
Python 3.9.9 64-bit (windows store)
0 0
Ln 18, Col 19
Spaces: 4
UTF-8
CRLF
Python
ENG IN
01:21
21-11-2021

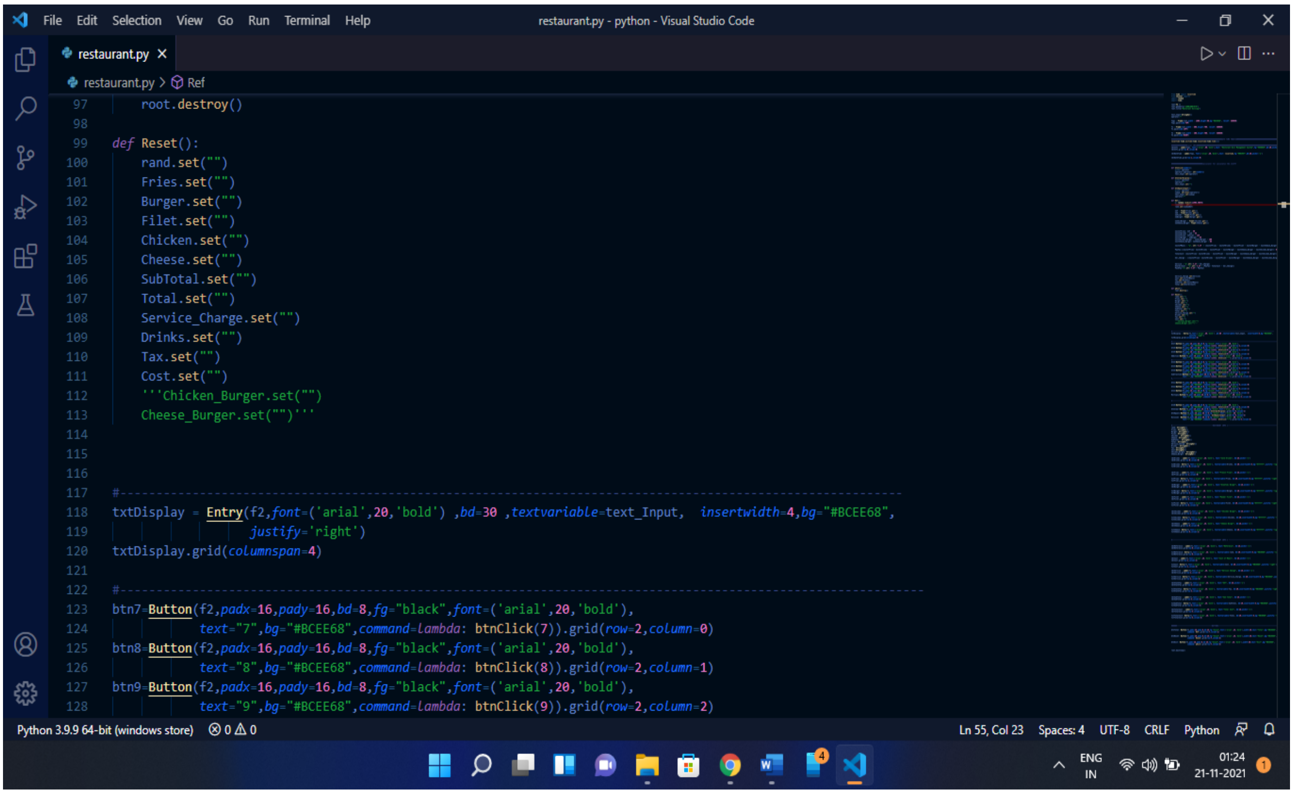


```python


#################################Calculator for calculatin the stufff

def btnClick(numbers):
    global operator
    operator=operator+ str(numbers)
    text_Input.set(operator)

def btnClearDisplay():
    global operator
    operator=""
    text_Input.set("")

def btnEqualsInput():
    global operator
    sumup= str(eval(operator))
    text_Input.set(sumup)
    operator=""

def Ref():
    x= random.randint(12908,50876)
    randomRef = str(x)
    rand.set(randomRef)

    CoF = float(Fries.get())
    CoD = float(Drinks.get())
    CoFilet = float(Filet.get())
    CoBurger= float(Burger.get())

    CoChicBurger = float(Chicken.get())
    CoCheese_Burger= float(Cheese.get())
```

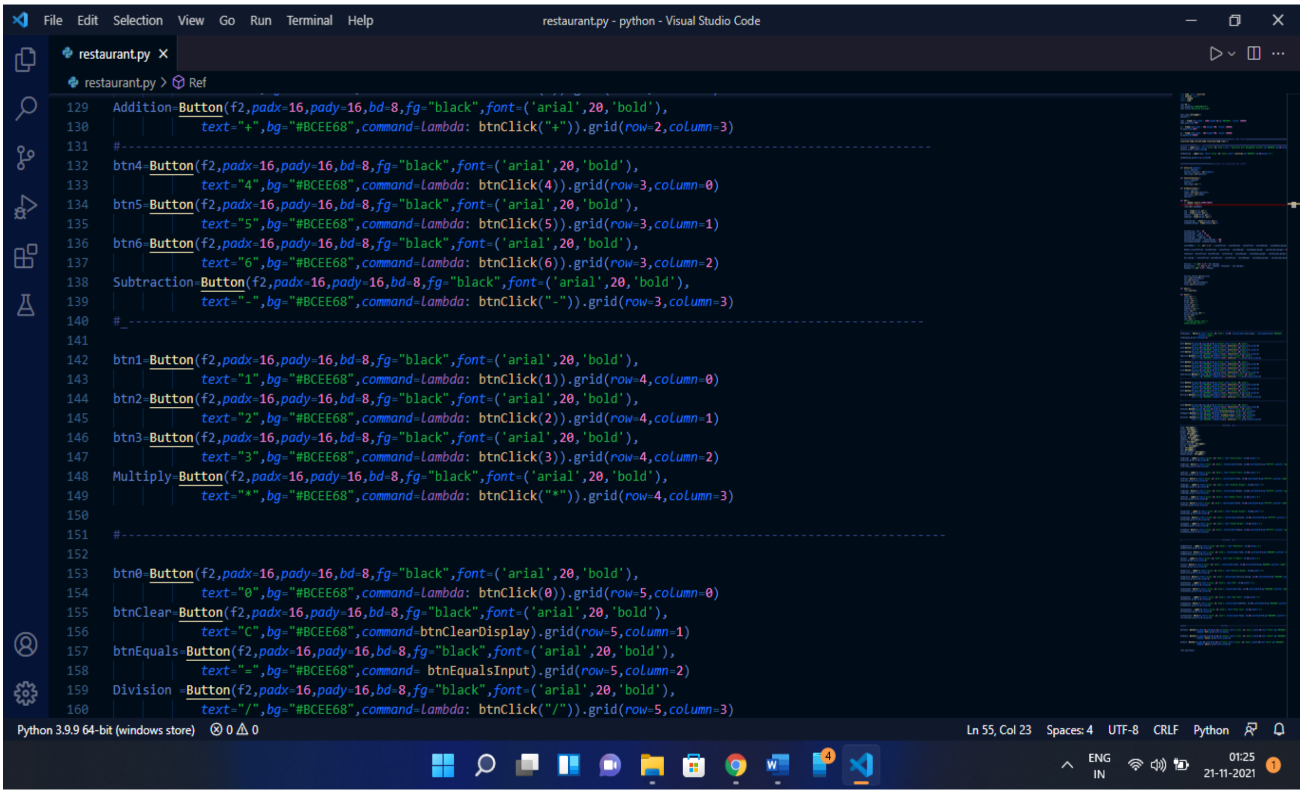
File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > ...
Python 3.9.9 64-bit (windows store)
0 0
Ln 18, Col 19
Spaces: 4
UTF-8
CRLF
Python
ENG IN
01:22
21-11-2021


File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store) ⊗ 0 ⚠ 0
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:23
21-11-2021

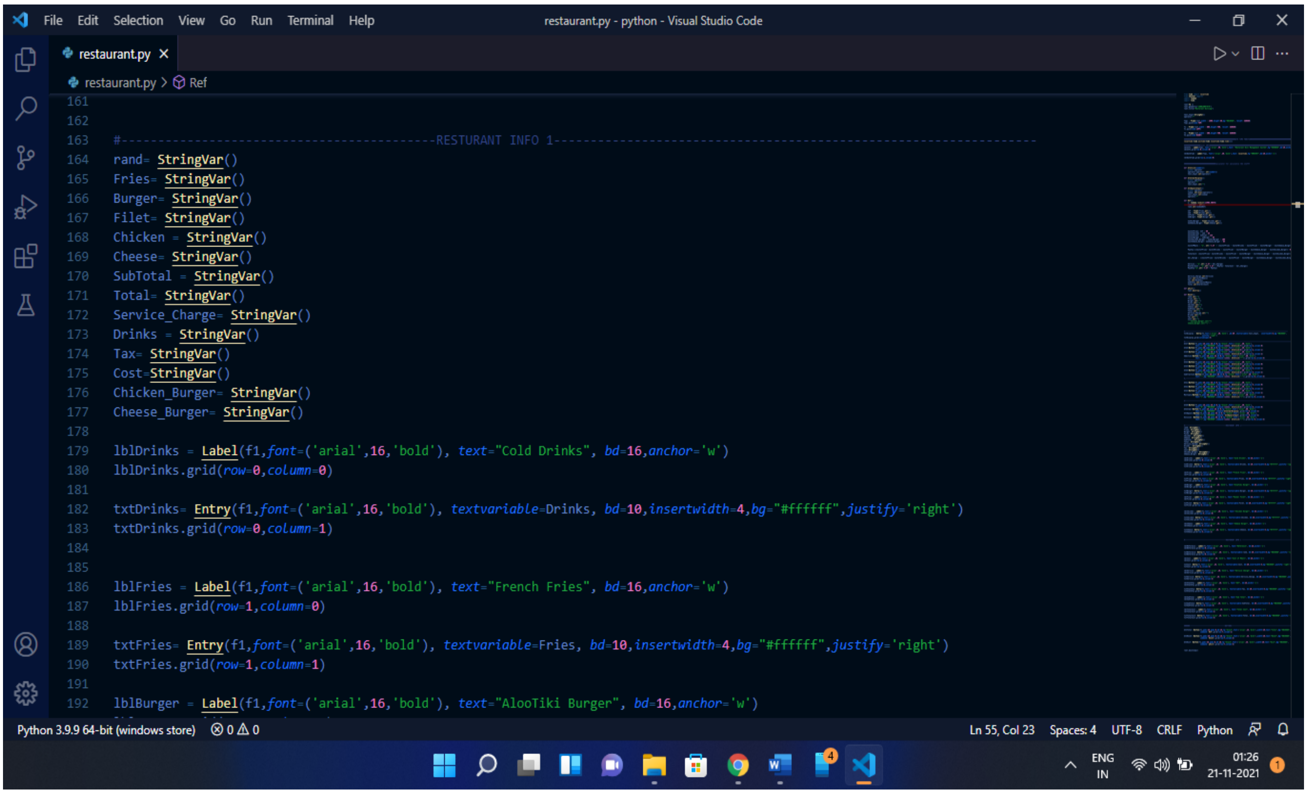
```python



    CostofFries= CoF * 50
    CostofDrinks = CoD * 45
    CostofFliet= CoFilet * 125
    CostofBurger = CoBurger * 65
    CostChicken_Burger = CoChicBurger * 100
    CostCheese_Burger= CoCheese_Burger * 85

    CostofMeals = "₹", str('%.2f' % (CostofFries + CostofDrinks + CostofFliet + CostofBurger + CostCheese_Burger + CostChicken_Burger))

    PayTax=((CostofFries+ CostofDrinks + CostofFliet + CostofBurger + CostCheese_Burger + CostChicken_Burger)* 0.2)

    TotalCost= (CostofFries+ CostofDrinks + CostofFliet + CostofBurger + CostCheese_Burger + CostChicken_Burger)

    Ser_Charge = ((CostofFries+ CostofDrinks + CostofFliet + CostofBurger + CostCheese_Burger + CostChicken_Burger)/99)


    Service = "₹",str('%.2f'% Ser_Charge)
    OverAllCost = "₹",str('%.2f'% (PayTax+ TotalCost + Ser_Charge))
    PaidTax="₹",str('%.2f'% PayTax)



    Service_Charge.set(Service)
    Cost.set(CostofMeals)
    Tax.set(PaidTax)
    SubTotal.set(CostofMeals)
    Total.set(OverAllCost)

def qExit():
```


File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store) ⊗ 0 ⚠ 0
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:24
21-11-2021

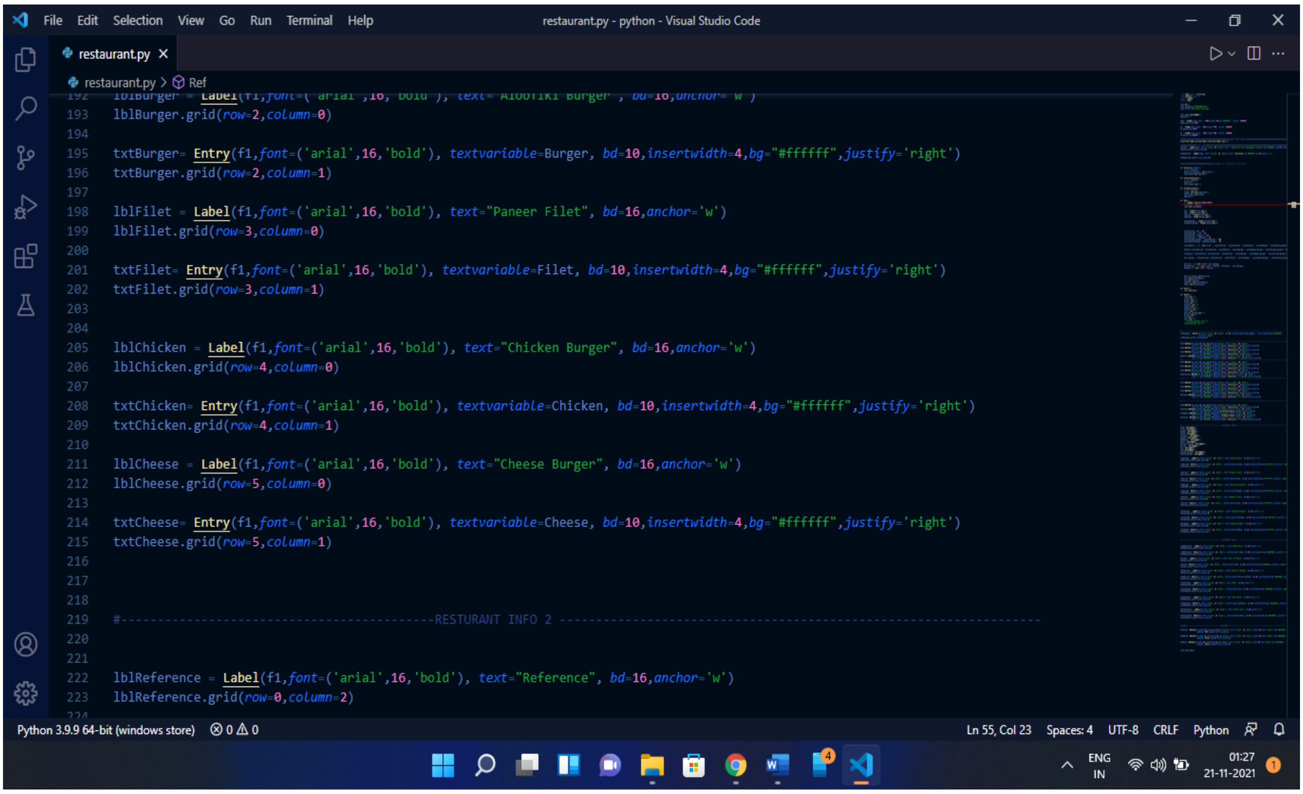
```python
    root.destroy()

def Reset():
    rand.set("")
    Fries.set("")
    Burger.set("")
    Filet.set("")
    Chicken.set("")
    Cheese.set("")
    SubTotal.set("")
    Total.set("")
    Service_Charge.set("")
    Drinks.set("")
    Tax.set("")
    Cost.set("")
    '''Chicken_Burger.set("")
    Cheese_Burger.set("")'''



#---------------------------------------------------------------------------------------------------------------
txtDisplay = Entry(f2,font=('arial',20,'bold') ,bd=30 ,textvariable=text_Input,  insertwidth=4,bg="#BCEE68",
                    justify='right')
txtDisplay.grid(columnspan=4)

#------------------------------------------------------------------------------------------------------------------
btn7=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="7",bg="#BCEE68",command=lambda: btnClick(7)).grid(row=2,column=0)
btn8=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="8",bg="#BCEE68",command=lambda: btnClick(8)).grid(row=2,column=1)
btn9=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="9",bg="#BCEE68",command=lambda: btnClick(9)).grid(row=2,column=2)
```


File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store) ⊗ 0 ⚠ 0
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:25 21-11-2021

```python
Addition=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="+",bg="#BCEE68",command=lambda: btnClick("+")).grid(row=2,column=3)
#-------------------------------------------------------------------------------------------
btn4=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="4",bg="#BCEE68",command=lambda: btnClick(4)).grid(row=3,column=0)
btn5=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="5",bg="#BCEE68",command=lambda: btnClick(5)).grid(row=3,column=1)
btn6=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="6",bg="#BCEE68",command=lambda: btnClick(6)).grid(row=3,column=2)
Subtraction=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="-",bg="#BCEE68",command=lambda: btnClick("-")).grid(row=3,column=3)
#_------------------------------------------------------------------------------------------

btn1=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="1",bg="#BCEE68",command=lambda: btnClick(1)).grid(row=4,column=0)
btn2=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="2",bg="#BCEE68",command=lambda: btnClick(2)).grid(row=4,column=1)
btn3=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="3",bg="#BCEE68",command=lambda: btnClick(3)).grid(row=4,column=2)
Multiply=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="*",bg="#BCEE68",command=lambda: btnClick("*")).grid(row=4,column=3)

#---------------------------------------------------------------------------------------------

btn0=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="0",bg="#BCEE68",command=lambda: btnClick(0)).grid(row=5,column=0)
btnClear=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="C",bg="#BCEE68",command=btnClearDisplay).grid(row=5,column=1)
btnEquals=Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="=",bg="#BCEE68",command= btnEqualsInput).grid(row=5,column=2)
Division =Button(f2,padx=16,pady=16,bd=8,fg="black",font=('arial',20,'bold'),
            text="/",bg="#BCEE68",command=lambda: btnClick("/")).grid(row=5,column=3)
```

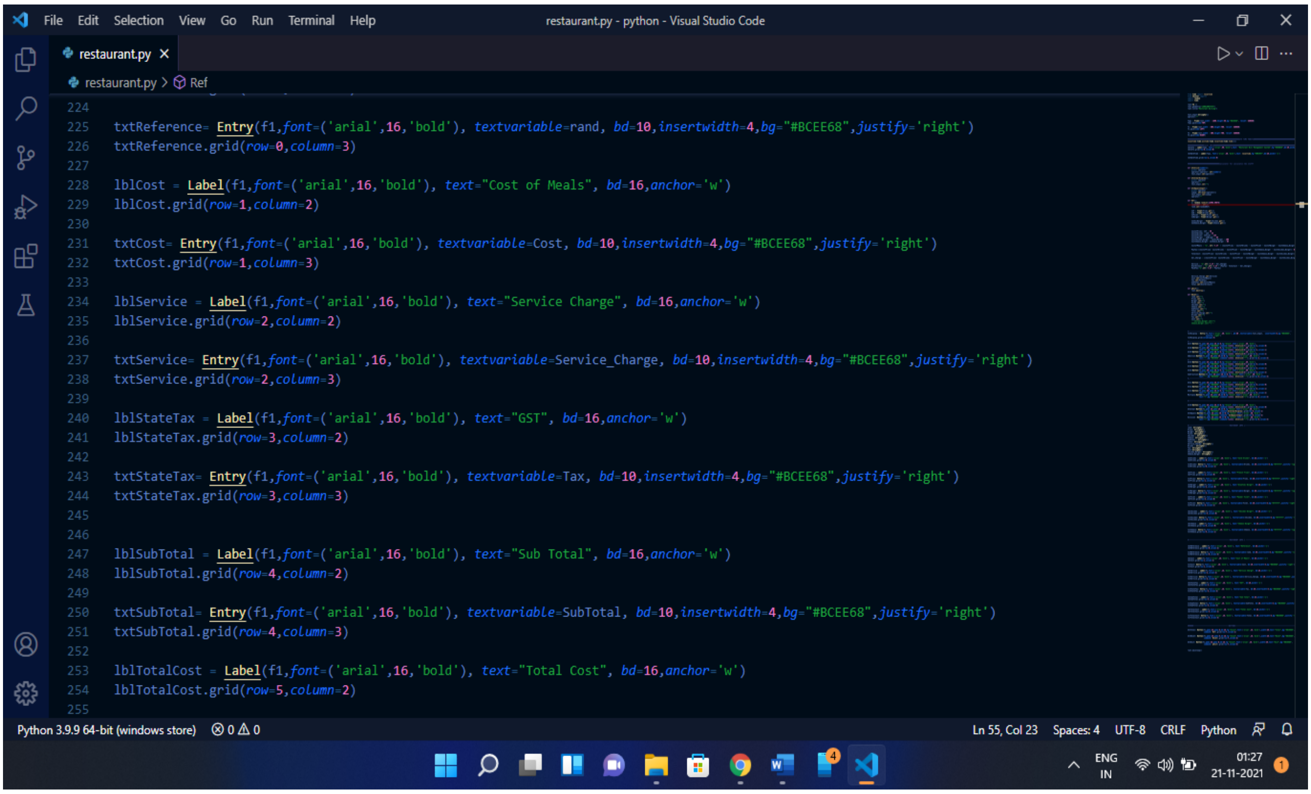

File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store) ⊗ 0 ⚠ 0
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:26 21-11-2021

```python


#----------------------------------------RESTURANT INFO 1--------------------------------------------------
rand= StringVar()
Fries= StringVar()
Burger= StringVar()
Filet= StringVar()
Chicken = StringVar()
Cheese= StringVar()
SubTotal = StringVar()
Total= StringVar()
Service_Charge= StringVar()
Drinks = StringVar()
Tax= StringVar()
Cost=StringVar()
Chicken_Burger= StringVar()
Cheese_Burger= StringVar()

lblDrinks = Label(f1,font=('arial',16,'bold'), text="Cold Drinks", bd=16,anchor='w')
lblDrinks.grid(row=0,column=0)

txtDrinks= Entry(f1,font=('arial',16,'bold'), textvariable=Drinks, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtDrinks.grid(row=0,column=1)


lblFries = Label(f1,font=('arial',16,'bold'), text="French Fries", bd=16,anchor='w')
lblFries.grid(row=1,column=0)

txtFries= Entry(f1,font=('arial',16,'bold'), textvariable=Fries, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtFries.grid(row=1,column=1)

lblBurger = Label(f1,font=('arial',16,'bold'), text="AlooTiki Burger", bd=16,anchor='w')
```



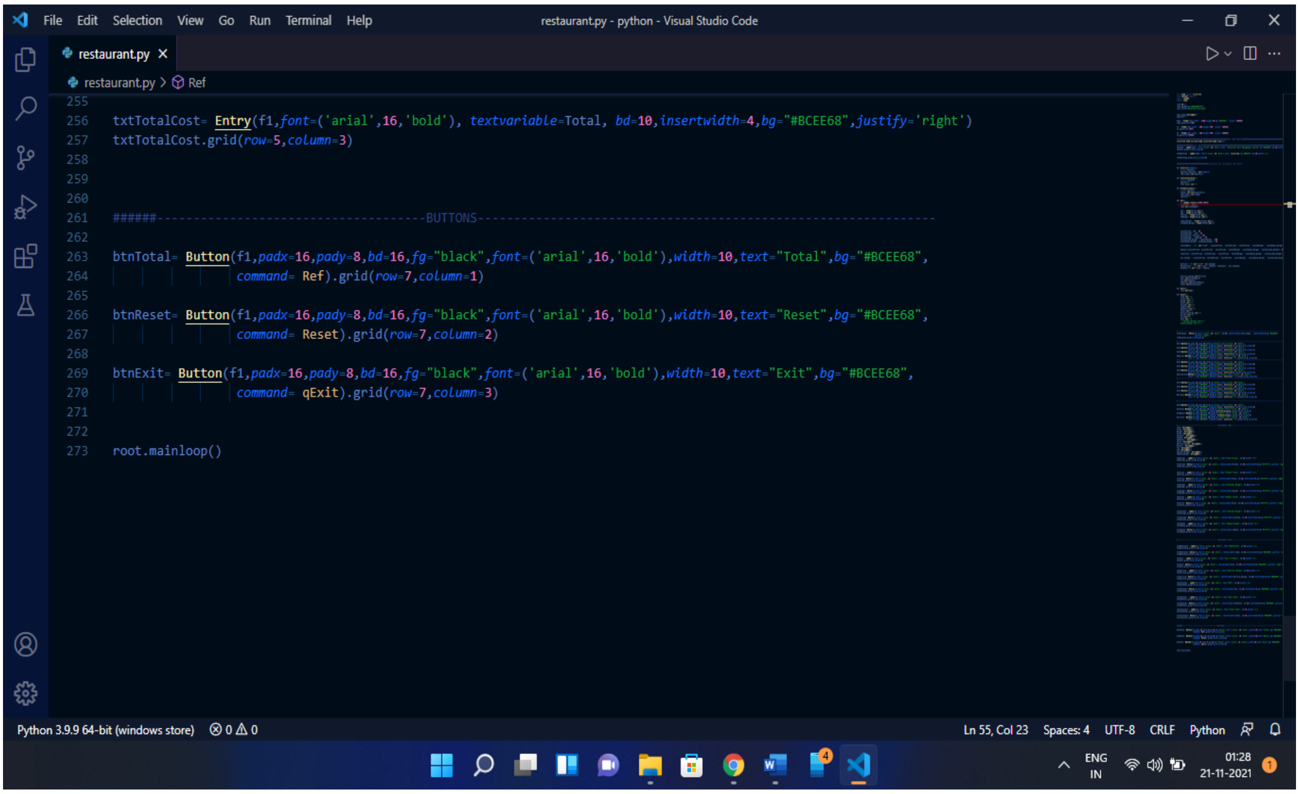
File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store)
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:27
21-11-2021

```python
lblBurger = Label(f1,font=('arial',16,'bold'), text="AlooTiki Burger", bd=16,anchor='w')
lblBurger.grid(row=2,column=0)

txtBurger= Entry(f1,font=('arial',16,'bold'), textvariable=Burger, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtBurger.grid(row=2,column=1)

lblFilet = Label(f1,font=('arial',16,'bold'), text="Paneer Filet", bd=16,anchor='w')
lblFilet.grid(row=3,column=0)

txtFilet= Entry(f1,font=('arial',16,'bold'), textvariable=Filet, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtFilet.grid(row=3,column=1)


lblChicken = Label(f1,font=('arial',16,'bold'), text="Chicken Burger", bd=16,anchor='w')
lblChicken.grid(row=4,column=0)

txtChicken= Entry(f1,font=('arial',16,'bold'), textvariable=Chicken, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtChicken.grid(row=4,column=1)

lblCheese = Label(f1,font=('arial',16,'bold'), text="Cheese Burger", bd=16,anchor='w')
lblCheese.grid(row=5,column=0)

txtCheese= Entry(f1,font=('arial',16,'bold'), textvariable=Cheese, bd=10,insertwidth=4,bg="#ffffff",justify='right')
txtCheese.grid(row=5,column=1)



#-------------------------------------------RESTURANT INFO 2 -----------------------------------------------------------------


lblReference = Label(f1,font=('arial',16,'bold'), text="Reference", bd=16,anchor='w')
lblReference.grid(row=0,column=2)
```



File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store)
Ln 55, Col 23 Spaces: 4 UTF-8 CRLF Python
ENG IN
01:27
21-11-2021

```python

txtReference= Entry(f1,font=('arial',16,'bold'), textvariable=rand, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtReference.grid(row=0,column=3)

lblCost = Label(f1,font=('arial',16,'bold'), text="Cost of Meals", bd=16,anchor='w')
lblCost.grid(row=1,column=2)

txtCost= Entry(f1,font=('arial',16,'bold'), textvariable=Cost, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtCost.grid(row=1,column=3)

lblService = Label(f1,font=('arial',16,'bold'), text="Service Charge", bd=16,anchor='w')
lblService.grid(row=2,column=2)

txtService= Entry(f1,font=('arial',16,'bold'), textvariable=Service_Charge, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtService.grid(row=2,column=3)

lblStateTax = Label(f1,font=('arial',16,'bold'), text="GST", bd=16,anchor='w')
lblStateTax.grid(row=3,column=2)

txtStateTax= Entry(f1,font=('arial',16,'bold'), textvariable=Tax, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtStateTax.grid(row=3,column=3)


lblSubTotal = Label(f1,font=('arial',16,'bold'), text="Sub Total", bd=16,anchor='w')
lblSubTotal.grid(row=4,column=2)

txtSubTotal= Entry(f1,font=('arial',16,'bold'), textvariable=SubTotal, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtSubTotal.grid(row=4,column=3)

lblTotalCost = Label(f1,font=('arial',16,'bold'), text="Total Cost", bd=16,anchor='w')
lblTotalCost.grid(row=5,column=2)

```



File Edit Selection View Go Run Terminal Help
restaurant.py - python - Visual Studio Code
restaurant.py
restaurant.py > Ref
Python 3.9.9 64-bit (windows store)
⊗ 0 ⚠ 0
Ln 55, Col 23
Spaces: 4
UTF-8
CRLF
Python
ENG
IN
01:28
21-11-2021

```python

txtTotalCost= Entry(f1,font=('arial',16,'bold'), textvariable=Total, bd=10,insertwidth=4,bg="#BCEE68",justify='right')
txtTotalCost.grid(row=5,column=3)



######-----------------------------------BUTTONS-------------------------------------------------------------

btnTotal= Button(f1,padx=16,pady=8,bd=16,fg="black",font=('arial',16,'bold'),width=10,text="Total",bg="#BCEE68",
                 command= Ref).grid(row=7,column=1)

btnReset= Button(f1,padx=16,pady=8,bd=16,fg="black",font=('arial',16,'bold'),width=10,text="Reset",bg="#BCEE68",
                 command= Reset).grid(row=7,column=2)

btnExit= Button(f1,padx=16,pady=8,bd=16,fg="black",font=('arial',16,'bold'),width=10,text="Exit",bg="#BCEE68",
                 command= qExit).grid(row=7,column=3)


root.mainloop()
```

CONCLUSION: -

It is my hope that this document will be of huge help with understanding of my little project as I have used a different approach which has proved beneficial for me and easy for us to understand the vast ocean. I have put all my effort in this project with best of my knowledge and I will work for betterment.

REFERENCES: -

1.Geeksforgeeks: https://www.geeksforgeeks.org/

2.W3schools: https://www.w3schools.com/

3.Stackoverflow: https://stackoverflow.com/

4.Youtube: https://www.youtube.com/