LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

B-TECH(CSE)

Project Report

On

RESTAURANT MANAGEMENT WITH GST BILLING

SUBMITTED BY :

NAME OF STUDENT - MADHUR YADAV, AYUSH PARASHAR

REGISTRATION NUMBER - 12008190, 1200758

LOVELY PROFESSIONAL UNIVERSITY

ACKNOWLEDGMENT

We would like to express our gratitude towards all the people who have contributed their precious time and effort to help us while making this report . Without them it would not have been possible for us to understand the concept of representing and the ways of making the report file and complete the project.

We would like to thank our faculty, Department of Computer Science and Engineering for their guidance, support, motivation and encouragement through out the period this work was carried out.

Their readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

We are grateful to Head of School , Head of Faculty and all the faculty concerned toward this report of Python Programming (INT 213) for providing necessary guidance for making the same.

LOVELY PROFESSIONAL UNIVERSITY

ABSTRACT

The project "Restaurant Management With GST Billing" is implemented to reduce the manual work and enhances the accuracy and precision of work in a restaurant.

The businesses in restaurants are increasing rapidly. At the same time, challenges in managing its operations tasks arises. The best way to optimize these activities is growing the business digitaly. Today's generation encourages high-tech services especially over the Internet.

RESTAURANT MANAGEMENT WITH GST BILLING is online application for restaurant management. Hence the project is developed proficiently to help restaurant. This system wake to provide service facility to restaurant and owners automate their BILLING OPERATIONS.

This system reduce unnacessary calculations.As per the new rule Goods and Service Tax (GST)by Government of India levy on both AC and non-AC restaurants to 5%. Every restaurant charges 5% GST breakup as 2.5% State GST (SGST) and 2.5 % Central GST (CGST) without any other service tax and any other VAT charges..

We have developed a RESTAURANT MANAGEMENT WITH GST BILLING with inbuilt GST calculation in the bill, as well as it shows the amount in rupees how much GST is applied.

RESTAURANT MANAGEMENT WITH GST BILLING project fully developed in python language which is currently demanded in market using python programming.

LOVELY PROFESSIONAL UNIVERSITY

INDEX

CHAPTER 1

INTRODUCTION TO PYTHON PROGRAMMING

✓ PYTHON

Python Language Introduction

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

✓ History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python graphical user interfaces (GUIs)

- **Tkinter** – Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.

- **wxPython** – This is an open-source Python interface for wxWindows http://wxpython.org.
- **JPython** – JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

There are many other interfaces available, which you can find them on the net.

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
1	**Button** The Button widget is used to display buttons in your application.
2	**Canvas** The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	**Checkbutton** The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.

4	**Entry** The Entry widget is used to display a single-line text field for accepting values from a user.
5	**Frame** The Frame widget is used as a container widget to organize other widgets.
6	**Label** The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	**Listbox** The Listbox widget is used to provide a list of options to a user.
8	**Menubutton** The Menubutton widget is used to display menus in your application.
9	**Menu** The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	**Message** The Message widget is used to display multiline text fields for accepting values from a user.

11	**Radiobutton** The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	**Scale** The Scale widget is used to provide a slider widget.
13	**Scrollbar** The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	**Text** The Text widget is used to display text in multiple lines.
15	**Toplevel** The Toplevel widget is used to provide a separate window container.
16	**Spinbox** The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	**PanedWindow** A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	**LabelFrame**

	A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	**tkMessageBox** This module is used to display message boxes in your applications.

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

- The *pack()* Method – This geometry manager organizes widgets in blocks before placing them in the parent widget.
- The *grid()* Method – This geometry manager organizes widgets in a table-like structure in the parent widget.
- The *place()* Method – This geometry manager organizes widgets by placing them in a specific position in the parent widget.

LOVELY PROFESSIONAL UNIVERSITY

CHAPTER-2

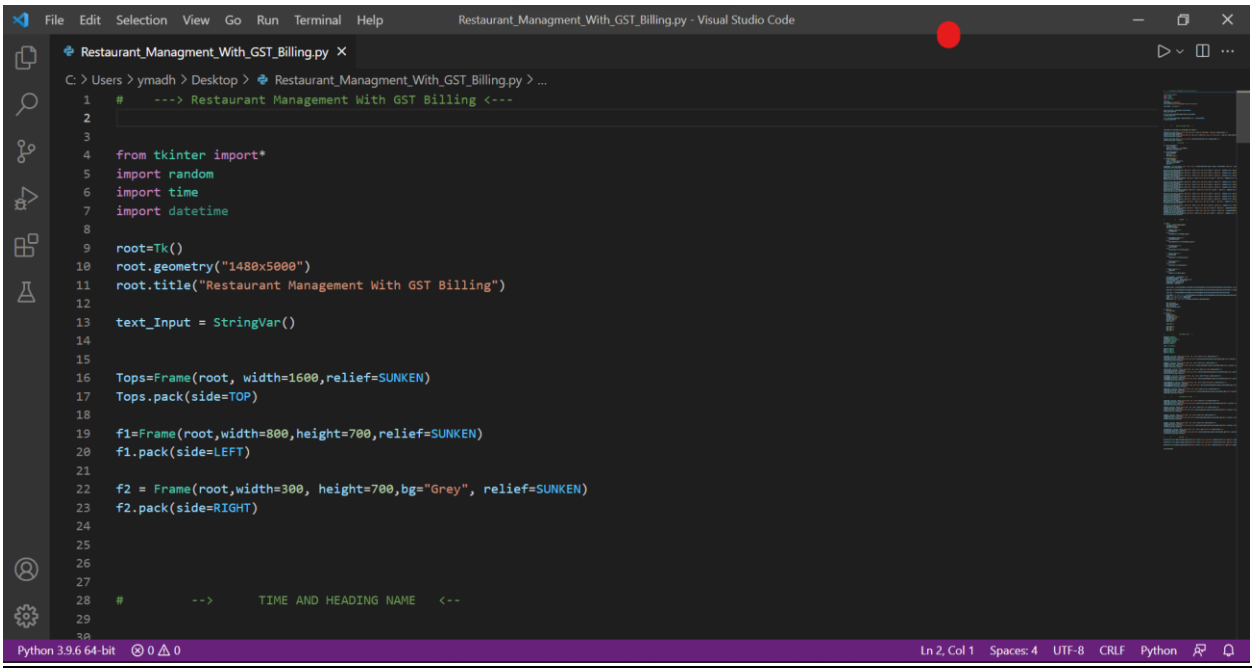
IMPLEMENTATION OF PROGRAM

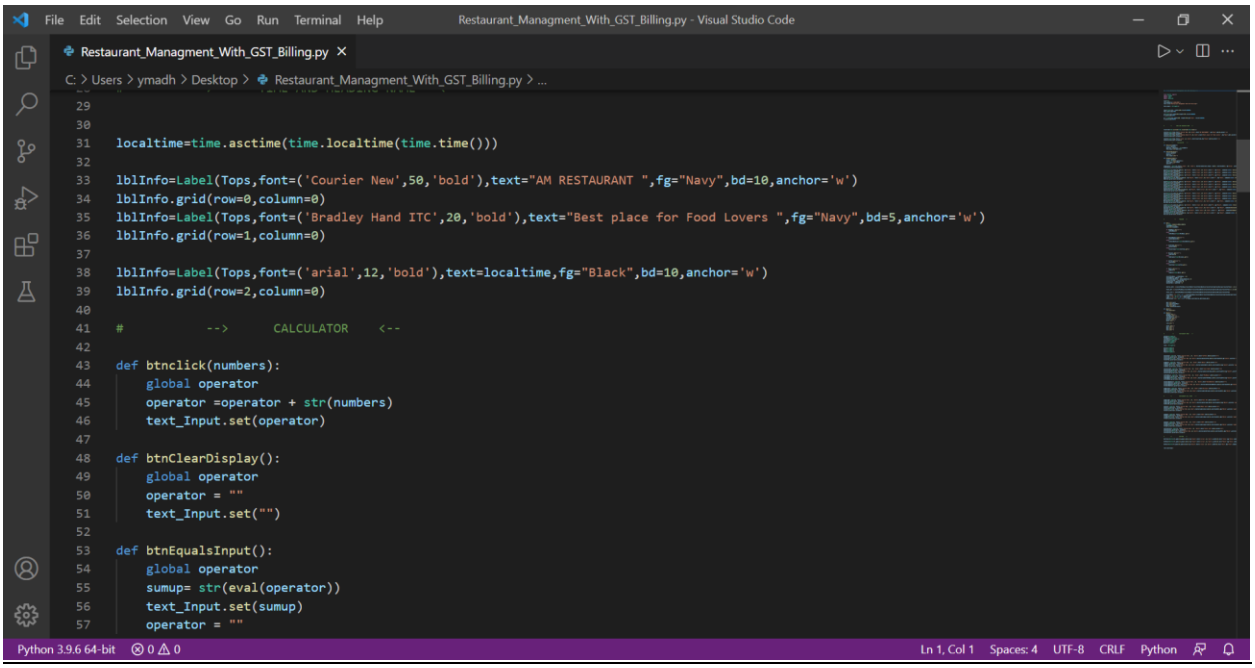
Language : Python

Compiler : VS CODE

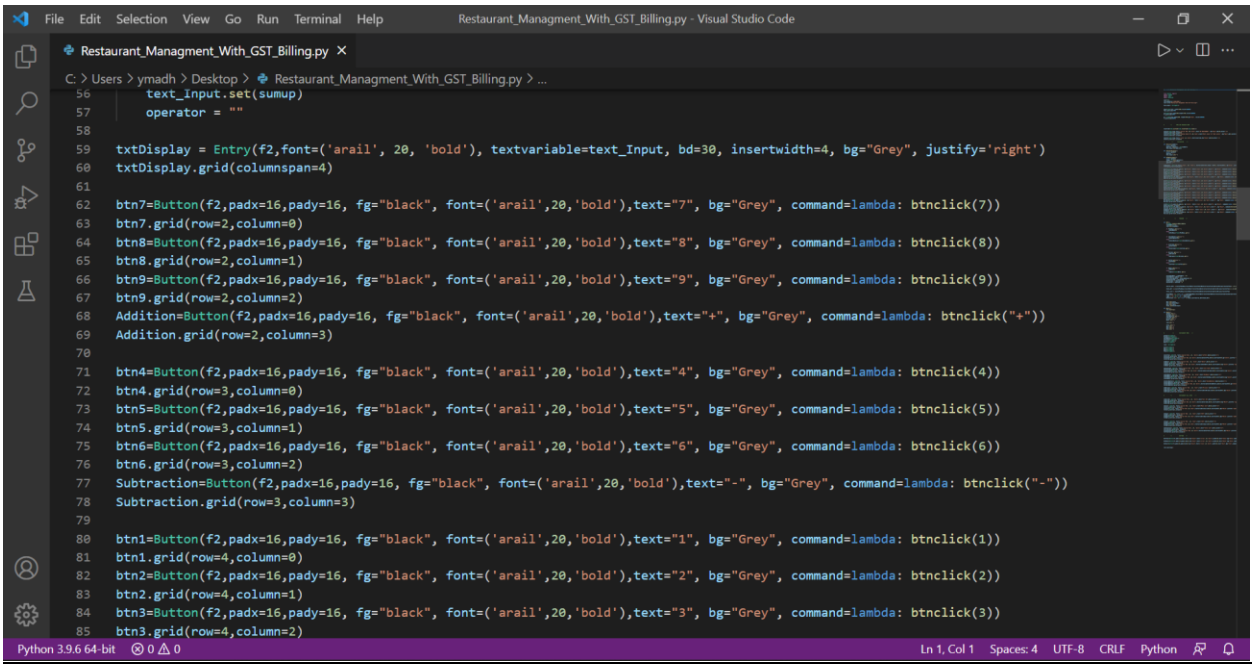
Technologies used : Python 3.6. , Python Tkinter GUI

CODE OF PROJECT

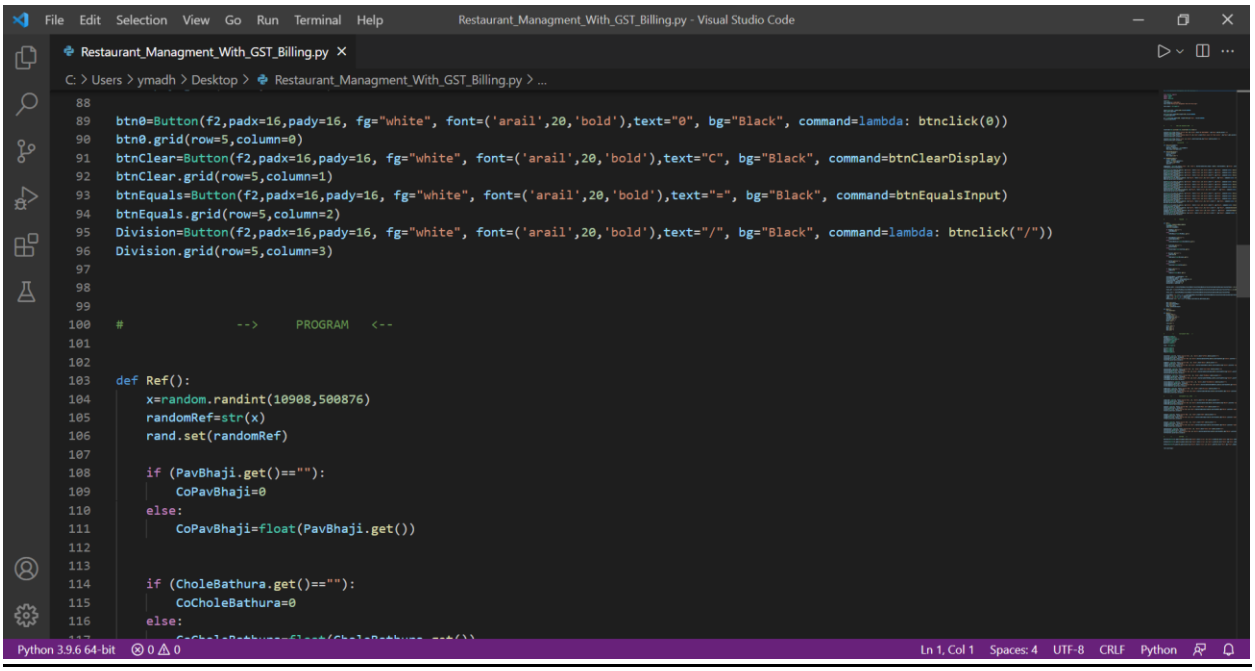




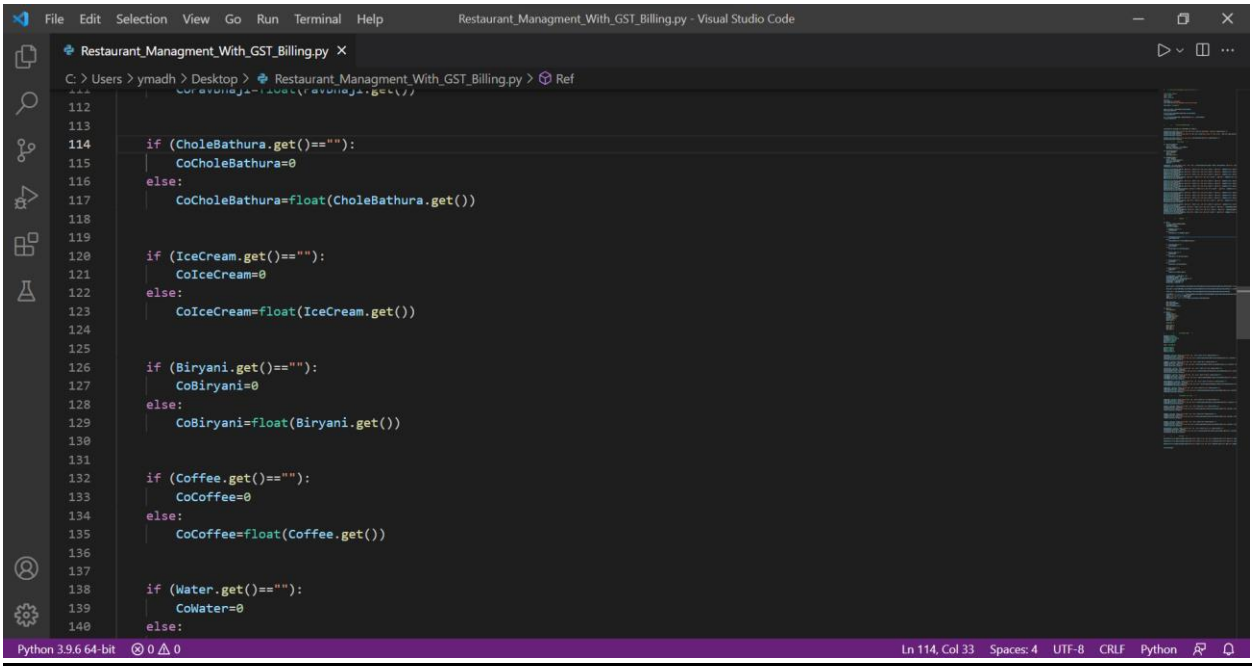
```
localtime=time.asctime(time.localtime(time.time()))

lblInfo=Label(Tops,font=('Courier New',50,'bold'),text="AM RESTAURANT ",fg="Navy",bd=10,anchor='w')
lblInfo.grid(row=0,column=0)
lblInfo=Label(Tops,font=('Bradley Hand ITC',20,'bold'),text="Best place for Food Lovers ",fg="Navy",bd=5,anchor='w')
lblInfo.grid(row=1,column=0)

lblInfo=Label(Tops,font=('arial',12,'bold'),text=localtime,fg="Black",bd=10,anchor='w')
lblInfo.grid(row=2,column=0)

#           -->      CALCULATOR    <--

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 = ""
```



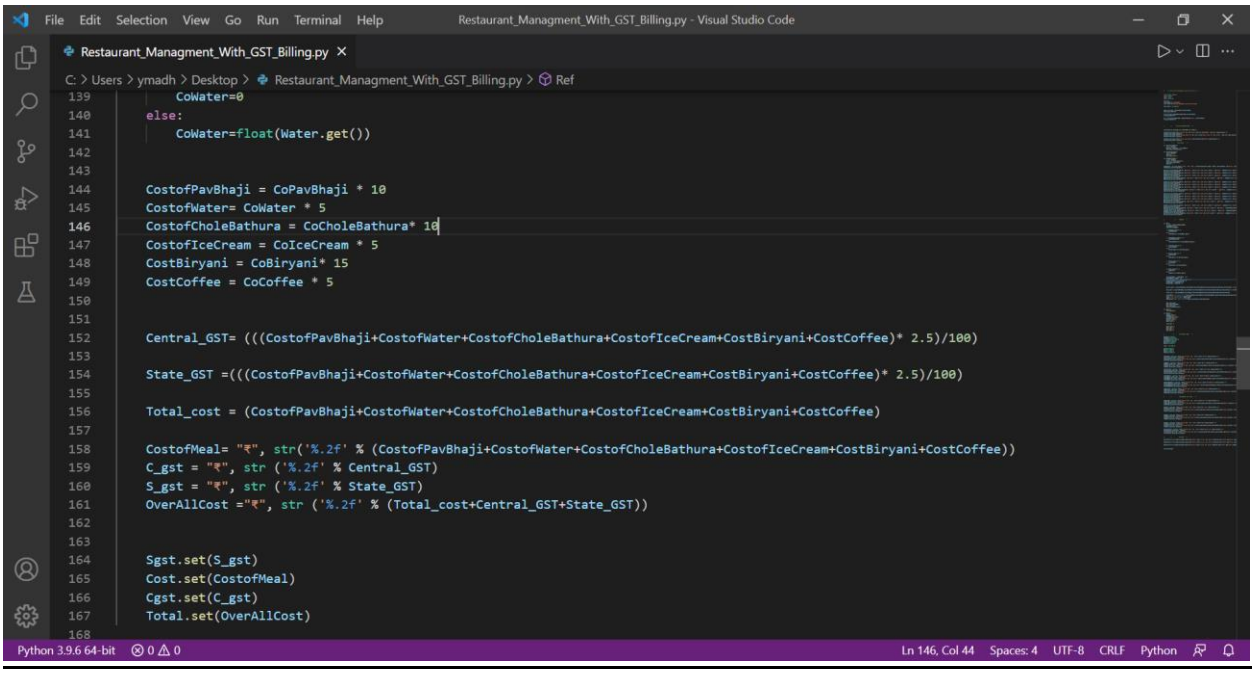
```
    text_Input.set(sumup)
    operator = ""

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

btn7=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="7", bg="Grey", command=lambda: btnclick(7))
btn7.grid(row=2,column=0)
btn8=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="8", bg="Grey", command=lambda: btnclick(8))
btn8.grid(row=2,column=1)
btn9=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="9", bg="Grey", command=lambda: btnclick(9))
btn9.grid(row=2,column=2)
Addition=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="+", bg="Grey", command=lambda: btnclick("+"))
Addition.grid(row=2,column=3)

btn4=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="4", bg="Grey", command=lambda: btnclick(4))
btn4.grid(row=3,column=0)
btn5=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="5", bg="Grey", command=lambda: btnclick(5))
btn5.grid(row=3,column=1)
btn6=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="6", bg="Grey", command=lambda: btnclick(6))
btn6.grid(row=3,column=2)
Subtraction=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="-", bg="Grey", command=lambda: btnclick("-"))
Subtraction.grid(row=3,column=3)

btn1=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="1", bg="Grey", command=lambda: btnclick(1))
btn1.grid(row=4,column=0)
btn2=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="2", bg="Grey", command=lambda: btnclick(2))
btn2.grid(row=4,column=1)
btn3=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="3", bg="Grey", command=lambda: btnclick(3))
btn3.grid(row=4,column=2)
```



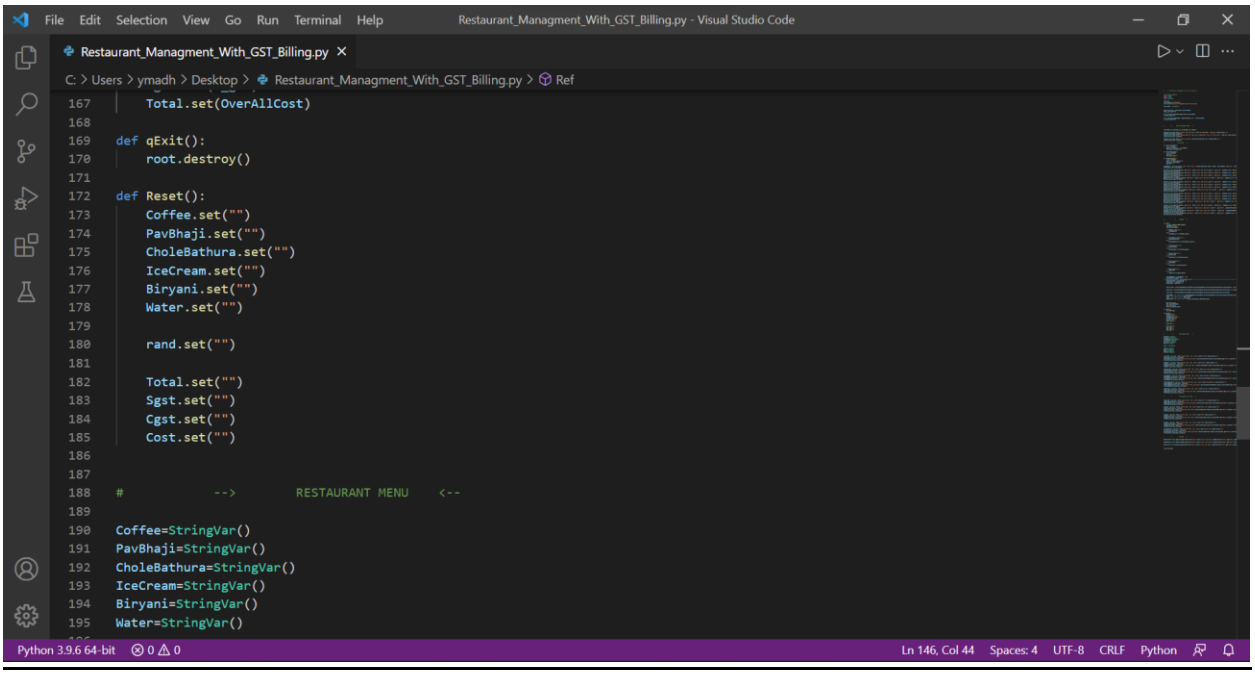
```
btn0=Button(f2,padx=16,pady=16, fg="white", font=('arail',20,'bold'),text="0", bg="Black", command=lambda: btnclick(0))
btn0.grid(row=5,column=0)
btnClear=Button(f2,padx=16,pady=16, fg="white", font=('arail',20,'bold'),text="C", bg="Black", command=btnClearDisplay)
btnClear.grid(row=5,column=1)
btnEquals=Button(f2,padx=16,pady=16, fg="white", font=('arail',20,'bold'),text="=", bg="Black", command=btnEqualsInput)
btnEquals.grid(row=5,column=2)
Division=Button(f2,padx=16,pady=16, fg="white", font=('arail',20,'bold'),text="/", bg="Black", command=lambda: btnclick("/"))
Division.grid(row=5,column=3)



#               -->     PROGRAM   <--


def Ref():
    x=random.randint(10908,500876)
    randomRef=str(x)
    rand.set(randomRef)

    if (PavBhaji.get()==""):
        CoPavBhaji=0
    else:
        CoPavBhaji=float(PavBhaji.get())


    if (CholeBathura.get()==""):
        CoCholeBathura=0
    else:
```



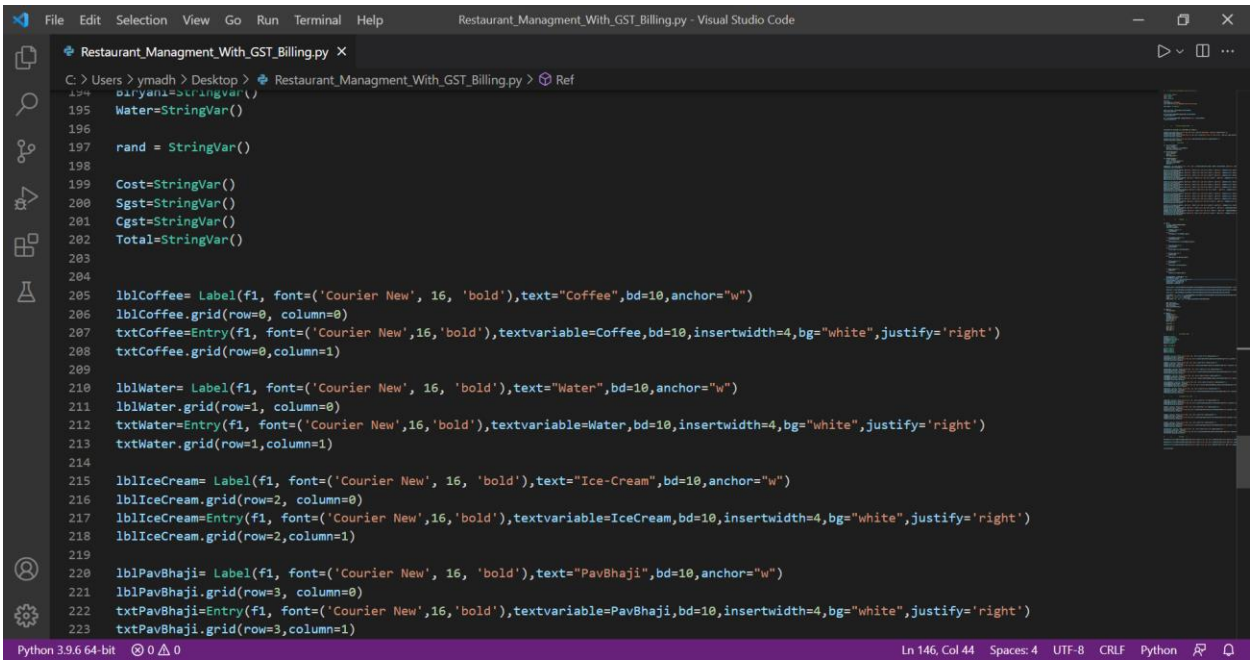
```
    if (CholeBathura.get()==""):
        CoCholeBathura=0
    else:
        CoCholeBathura=float(CholeBathura.get())


    if (IceCream.get()==""):
        CoIceCream=0
    else:
        CoIceCream=float(IceCream.get())


    if (Biryani.get()==""):
        CoBiryani=0
    else:
        CoBiryani=float(Biryani.get())


    if (Coffee.get()==""):
        CoCoffee=0
    else:
        CoCoffee=float(Coffee.get())


    if (Water.get()==""):
        CoWater=0
    else:
```

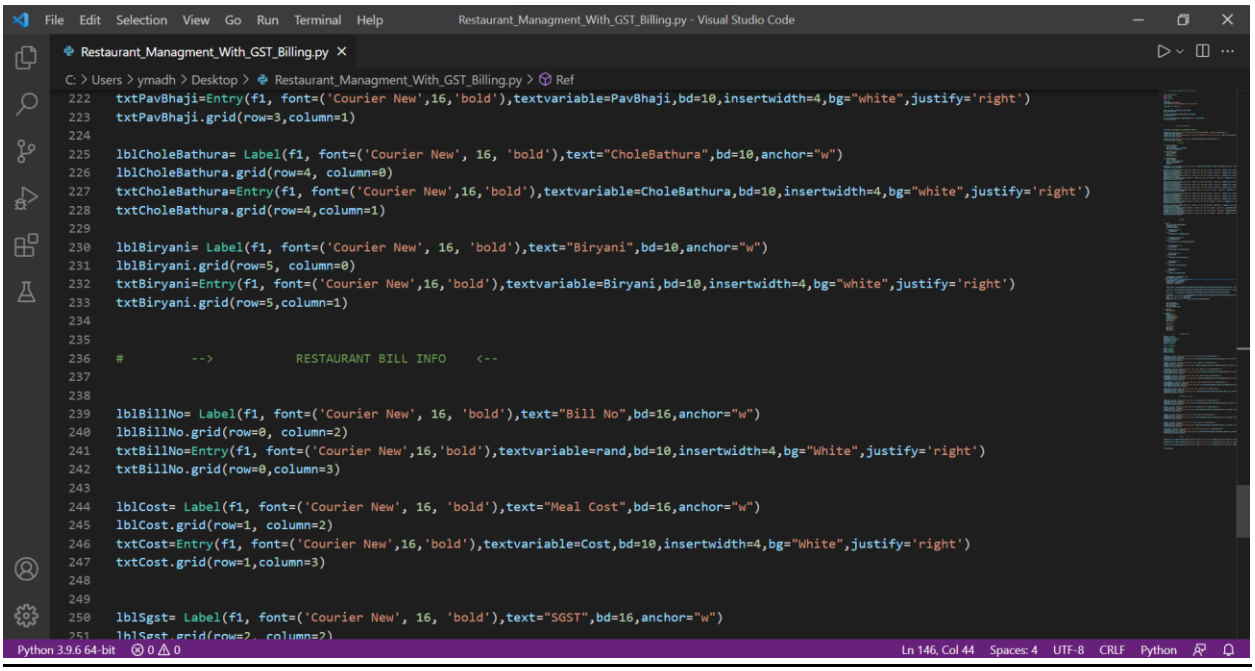


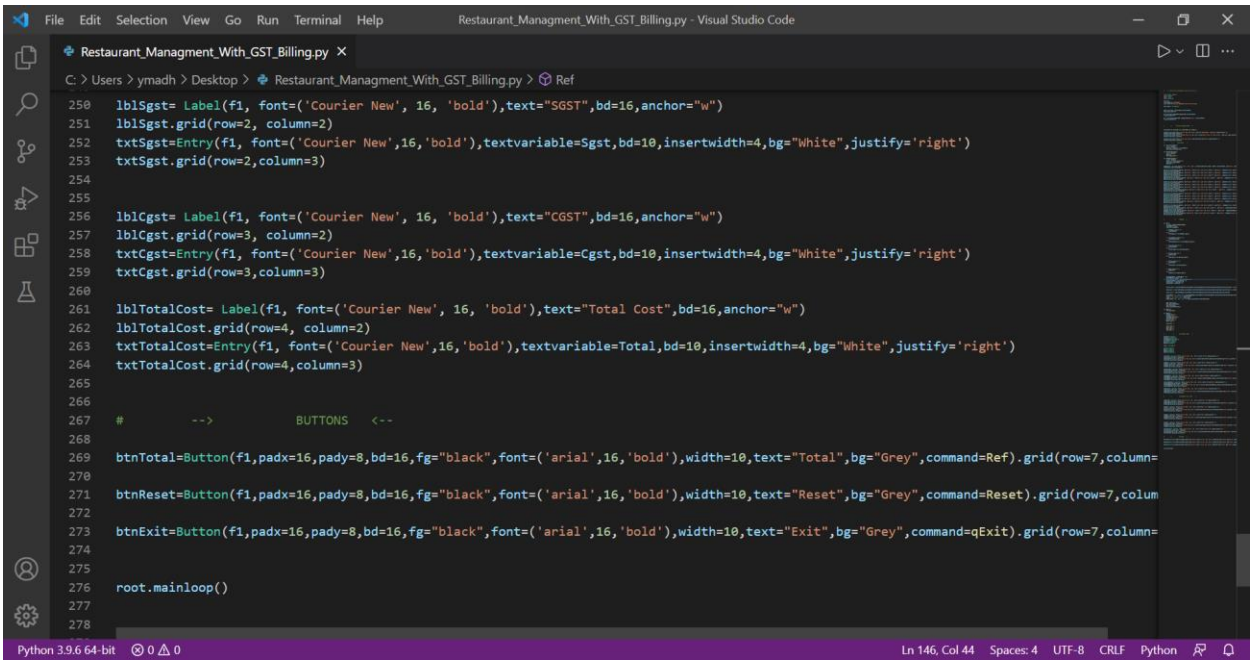
```
        CoWater=0
    else:
        CoWater=float(Water.get())


    CostofPavBhaji = CoPavBhaji * 10
    CostofWater= CoWater * 5
    CostofCholeBathura = CoCholeBathura* 10
    CostofIceCream = CoIceCream * 5
    CostBiryani = CoBiryani* 15
    CostCoffee = CoCoffee * 5


    Central_GST= (((CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBiryani+CostCoffee)* 2.5)/100)

    State_GST =(((CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBiryani+CostCoffee)* 2.5)/100)

    Total_cost = (CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBiryani+CostCoffee)

    CostofMeal= "₹", str('%.2f' % (CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBiryani+CostCoffee))
    C_gst = "₹", str ('%.2f' % Central_GST)
    S_gst = "₹", str ('%.2f' % State_GST)
    OverAllCost ="₹", str ('%.2f' % (Total_cost+Central_GST+State_GST))


    Sgst.set(S_gst)
    Cost.set(CostofMeal)
    Cgst.set(C_gst)
    Total.set(OverAllCost)
```



```
    Total.set(OverAllCost)

def qExit():
    root.destroy()

def Reset():
    Coffee.set("")
    PavBhaji.set("")
    CholeBathura.set("")
    IceCream.set("")
    Biryani.set("")
    Water.set("")

    rand.set("")

    Total.set("")
    Sgst.set("")
    Cgst.set("")
    Cost.set("")


#           -->        RESTAURANT MENU     <--

Coffee=StringVar()
PavBhaji=StringVar()
CholeBathura=StringVar()
IceCream=StringVar()
Biryani=StringVar()
Water=StringVar()
```


File Edit Selection View Go Run Terminal Help
Restaurant_Managment_With_GST_Billing.py - Visual Studio Code
Restaurant_Managment_With_GST_Billing.py
C: > Users > ymadh > Desktop > Restaurant_Managment_With_GST_Billing.py > Ref
BiryaniStringVar()
Water=StringVar()
rand = StringVar()
Cost=StringVar()
Sgst=StringVar()
Cgst=StringVar()
Total=StringVar()
lblCoffee= Label(f1, font=('Courier New', 16, 'bold'),text="Coffee",bd=10,anchor="w")
lblCoffee.grid(row=0, column=0)
txtCoffee=Entry(f1, font=('Courier New',16,'bold'),textvariable=Coffee,bd=10,insertwidth=4,bg="white",justify='right')
txtCoffee.grid(row=0,column=1)
lblWater= Label(f1, font=('Courier New', 16, 'bold'),text="Water",bd=10,anchor="w")
lblWater.grid(row=1, column=0)
txtWater=Entry(f1, font=('Courier New',16,'bold'),textvariable=Water,bd=10,insertwidth=4,bg="white",justify='right')
txtWater.grid(row=1,column=1)
lblIceCream= Label(f1, font=('Courier New', 16, 'bold'),text="Ice-Cream",bd=10,anchor="w")
lblIceCream.grid(row=2, column=0)
lblIceCream=Entry(f1, font=('Courier New',16,'bold'),textvariable=IceCream,bd=10,insertwidth=4,bg="white",justify='right')
lblIceCream.grid(row=2,column=1)
lblPavBhaji= Label(f1, font=('Courier New', 16, 'bold'),text="PavBhaji",bd=10,anchor="w")
lblPavBhaji.grid(row=3, column=0)
txtPavBhaji=Entry(f1, font=('Courier New',16,'bold'),textvariable=PavBhaji,bd=10,insertwidth=4,bg="white",justify='right')
txtPavBhaji.grid(row=3,column=1)
Python 3.9.6 64-bit
Ln 146, Col 44 Spaces: 4 UTF-8 CRLF Python


File Edit Selection View Go Run Terminal Help
Restaurant_Managment_With_GST_Billing.py - Visual Studio Code
Restaurant_Managment_With_GST_Billing.py
C: > Users > ymadh > Desktop > Restaurant_Managment_With_GST_Billing.py > Ref
txtPavBhaji=Entry(f1, font=('Courier New',16,'bold'),textvariable=PavBhaji,bd=10,insertwidth=4,bg="white",justify='right')
txtPavBhaji.grid(row=3,column=1)
lblCholeBathura= Label(f1, font=('Courier New', 16, 'bold'),text="CholeBathura",bd=10,anchor="w")
lblCholeBathura.grid(row=4, column=0)
txtCholeBathura=Entry(f1, font=('Courier New',16,'bold'),textvariable=CholeBathura,bd=10,insertwidth=4,bg="white",justify='right')
txtCholeBathura.grid(row=4,column=1)
lblBiryani= Label(f1, font=('Courier New', 16, 'bold'),text="Biryani",bd=10,anchor="w")
lblBiryani.grid(row=5, column=0)
txtBiryani=Entry(f1, font=('Courier New',16,'bold'),textvariable=Biryani,bd=10,insertwidth=4,bg="white",justify='right')
txtBiryani.grid(row=5,column=1)
# --> RESTAURANT BILL INFO <--
lblBillNo= Label(f1, font=('Courier New', 16, 'bold'),text="Bill No",bd=16,anchor="w")
lblBillNo.grid(row=0, column=2)
txtBillNo=Entry(f1, font=('Courier New',16,'bold'),textvariable=rand,bd=10,insertwidth=4,bg="White",justify='right')
txtBillNo.grid(row=0,column=3)
lblCost= Label(f1, font=('Courier New', 16, 'bold'),text="Meal Cost",bd=16,anchor="w")
lblCost.grid(row=1, column=2)
txtCost=Entry(f1, font=('Courier New',16,'bold'),textvariable=Cost,bd=10,insertwidth=4,bg="White",justify='right')
txtCost.grid(row=1,column=3)
lblSgst= Label(f1, font=('Courier New', 16, 'bold'),text="SGST",bd=16,anchor="w")
Python 3.9.6 64-bit
Ln 146, Col 44 Spaces: 4 UTF-8 CRLF Python



IN TEXT FORM:

 # ---> Restaurant Management With GST Billing <---

from tkinter import*

import random

import time

import datetime

root=Tk()

root.geometry("1480x5000")

root.title("Restaurant Management With GST Billing")

text_Input = StringVar()

```
Tops=Frame(root, width=1600,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,bg="Grey", relief=SUNKEN)

f2.pack(side=RIGHT)

#        -->      TIME AND HEADING NAME    <--

localtime=time.asctime(time.localtime(time.time()))

lblInfo=Label(Tops,font=('Courier New',50,'bold'),text="AM RESTAURANT
",fg="Navy",bd=10,anchor='w')

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

lblInfo=Label(Tops,font=('Bradley Hand ITC',20,'bold'),text="Best place for
Food Lovers ",fg="Navy",bd=5,anchor='w')

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

lblInfo=Label(Tops,font=('arial',12,'bold'),text=localtime,fg="Black",bd=10,an
chor='w')

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


#        -->      CALCULATOR     <--

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 = ""


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


btn7=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="7", bg="Grey", command=lambda: btnclick(7))
btn7.grid(row=2,column=0)
btn8=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="8", bg="Grey", command=lambda: btnclick(8))
btn8.grid(row=2,column=1)
btn9=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="9", bg="Grey", command=lambda: btnclick(9))
btn9.grid(row=2,column=2)
```

```
Addition=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="+", bg="Grey", command=lambda: btnclick("+"))

Addition.grid(row=2,column=3)


btn4=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="4", bg="Grey", command=lambda: btnclick(4))

btn4.grid(row=3,column=0)

btn5=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="5", bg="Grey", command=lambda: btnclick(5))

btn5.grid(row=3,column=1)

btn6=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="6", bg="Grey", command=lambda: btnclick(6))

btn6.grid(row=3,column=2)

Subtraction=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="-", bg="Grey", command=lambda: btnclick("-"))

Subtraction.grid(row=3,column=3)


btn1=Button(f2,padx=16,pady=16, fg="black", font=('arail',20,'bold'),text="1",
bg="Grey", command=lambda: btnclick(1))

btn1.grid(row=4,column=0)

btn2=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="2", bg="Grey", command=lambda: btnclick(2))

btn2.grid(row=4,column=1)

btn3=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="3", bg="Grey", command=lambda: btnclick(3))

btn3.grid(row=4,column=2)

Multiply=Button(f2,padx=16,pady=16, fg="black",
font=('arail',20,'bold'),text="*", bg="Grey", command=lambda: btnclick("*"))
```

```
Multiply.grid(row=4,column=3)


btn0=Button(f2,padx=16,pady=16, fg="white",
font=('arail',20,'bold'),text="0", bg="Black", command=lambda: btnclick(0))

btn0.grid(row=5,column=0)

btnClear=Button(f2,padx=16,pady=16, fg="white",
font=('arail',20,'bold'),text="C", bg="Black", command=btnClearDisplay)

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

btnEquals=Button(f2,padx=16,pady=16, fg="white",
font=('arail',20,'bold'),text="=", bg="Black", command=btnEqualsInput)

btnEquals.grid(row=5,column=2)

Division=Button(f2,padx=16,pady=16, fg="white",
font=('arail',20,'bold'),text="/", bg="Black", command=lambda: btnclick("/"))

Division.grid(row=5,column=3)


#           -->    PROGRAM   <--

def Ref():

   x=random.randint(10908,500876)

   randomRef=str(x)

   rand.set(randomRef)


   if (PavBhaji.get()==""):

      CoPavBhaji=0

   else:

      CoPavBhaji=float(PavBhaji.get())

   if (CholeBathura.get()==""):
```

```
    CoCholeBathura=0
else:
    CoCholeBathura=float(CholeBathura.get())
if (IceCream.get()==""):
    CoIceCream=0
else:
    CoIceCream=float(IceCream.get())
if (Biryani.get()==""):
    CoBiryani=0
else:
    CoBiryani=float(Biryani.get())

 if (Coffee.get()==""):
    CoCoffee=0
else:
    CoCoffee=float(Coffee.get())

if (Water.get()==""):
    CoWater=0
else:
    CoWater=float(Water.get())

CostofPavBhaji = CoPavBhaji * 10
CostofWater= CoWater * 5
```

```
    CostofCholeBathura = CoCholeBathura* 10

    CostofIceCream = CoIceCream * 5

    CostBiryani = CoBiryani* 15

    CostCoffee = CoCoffee * 5



    Central_GST=
(((CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBir
yani+CostCoffee)* 2.5)/100)


    State_GST
=(((CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBi
ryani+CostCoffee)* 2.5)/100)


    Total_cost =
(CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBirya
ni+CostCoffee)


    CostofMeal= "₹", str('%.2f' %
(CostofPavBhaji+CostofWater+CostofCholeBathura+CostofIceCream+CostBirya
ni+CostCoffee))

    C_gst = "₹", str ('%.2f' % Central_GST)

    S_gst = "₹", str ('%.2f' % State_GST)

    OverAllCost ="₹", str ('%.2f' % (Total_cost+Central_GST+State_GST))



    Sgst.set(S_gst)
```

```
    Cost.set(CostofMeal)
    Cgst.set(C_gst)
    Total.set(OverAllCost)

def qExit():
    root.destroy()

def Reset():
    Coffee.set("")
    PavBhaji.set("")
    CholeBathura.set("")
    IceCream.set("")
    Biryani.set("")
    Water.set("")

    rand.set("")

    Total.set("")
    Sgst.set("")
    Cgst.set("")
    Cost.set("")

#          -->       RESTAURANT MENU    <--
```

```
Coffee=StringVar()

PavBhaji=StringVar()

CholeBathura=StringVar()

IceCream=StringVar()

Biryani=StringVar()

Water=StringVar()


rand = StringVar()


Cost=StringVar()

Sgst=StringVar()

Cgst=StringVar()

Total=StringVar()



lblCoffee= Label(f1, font=('Courier New', 16,
'bold'),text="Coffee",bd=10,anchor="w")

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

txtCoffee=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Coffee,bd=10,insertwidth=4,bg="white",justify='
right')

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


lblWater= Label(f1, font=('Courier New', 16,
'bold'),text="Water",bd=10,anchor="w")
```

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

txtWater=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Water,bd=10,insertwidth=4,bg="white",justify='r
ight')

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


lblIceCream= Label(f1, font=('Courier New', 16, 'bold'),text="Ice-
Cream",bd=10,anchor="w")

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

lblIceCream=Entry(f1, font=('Courier
New',16,'bold'),textvariable=IceCream,bd=10,insertwidth=4,bg="white",justif
y='right')

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


lblPavBhaji= Label(f1, font=('Courier New', 16,
'bold'),text="PavBhaji",bd=10,anchor="w")

lblPavBhaji.grid(row=3, column=0)

txtPavBhaji=Entry(f1, font=('Courier
New',16,'bold'),textvariable=PavBhaji,bd=10,insertwidth=4,bg="white",justify
='right')

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


lblCholeBathura= Label(f1, font=('Courier New', 16,
'bold'),text="CholeBathura",bd=10,anchor="w")

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

txtCholeBathura=Entry(f1, font=('Courier
New',16,'bold'),textvariable=CholeBathura,bd=10,insertwidth=4,bg="white",ju
stify='right')
```

```
txtCholeBathura.grid(row=4,column=1)


lblBiryani= Label(f1, font=('Courier New', 16,
'bold'),text="Biryani",bd=10,anchor="w")

lblBiryani.grid(row=5, column=0)

txtBiryani=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Biryani,bd=10,insertwidth=4,bg="white",justify=
'right')

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



#       -->         RESTAURANT BILL INFO    <--



lblBillNo= Label(f1, font=('Courier New', 16, 'bold'),text="Bill
No",bd=16,anchor="w")

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

txtBillNo=Entry(f1, font=('Courier
New',16,'bold'),textvariable=rand,bd=10,insertwidth=4,bg="White",justify='ri
ght')

txtBillNo.grid(row=0,column=3)


lblCost= Label(f1, font=('Courier New', 16, 'bold'),text="Meal
Cost",bd=16,anchor="w")

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

txtCost=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Cost,bd=10,insertwidth=4,bg="White",justify='ri
ght')
```

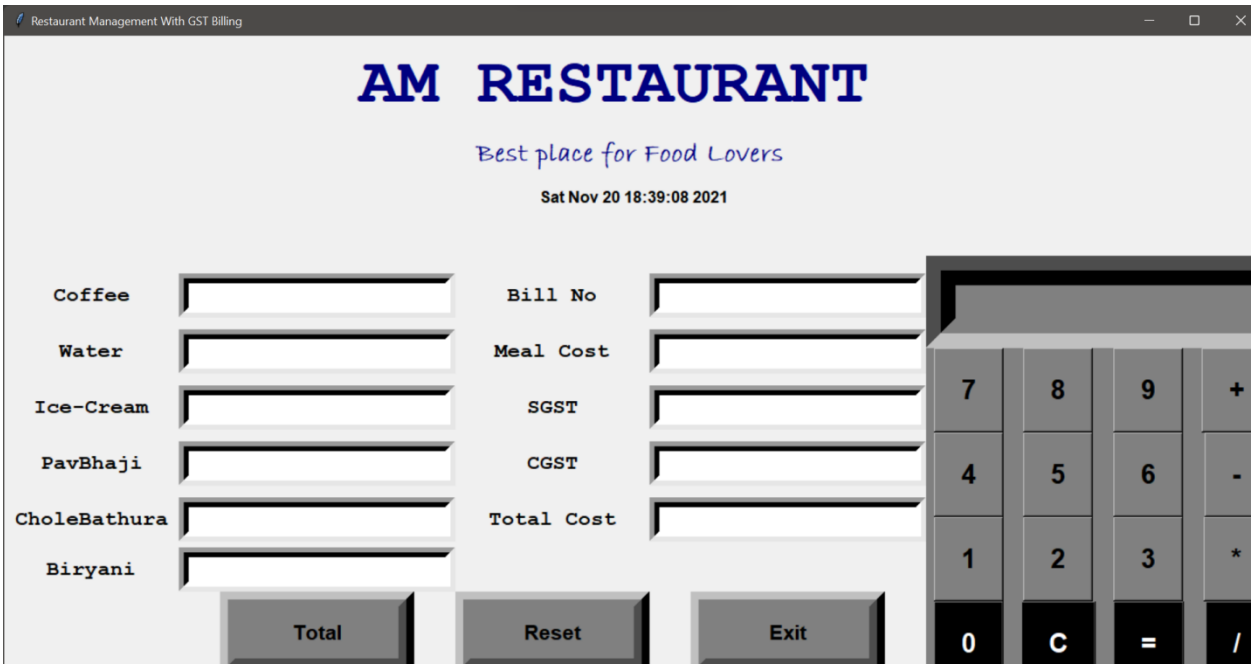
```
txtCost.grid(row=1,column=3)



lblSgst= Label(f1, font=('Courier New', 16,
'bold'),text="SGST",bd=16,anchor="w")

lblSgst.grid(row=2, column=2)

txtSgst=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Sgst,bd=10,insertwidth=4,bg="White",justify='ri
ght')

txtSgst.grid(row=2,column=3)



lblCgst= Label(f1, font=('Courier New', 16,
'bold'),text="CGST",bd=16,anchor="w")

lblCgst.grid(row=3, column=2)

txtCgst=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Cgst,bd=10,insertwidth=4,bg="White",justify='ri
ght')

txtCgst.grid(row=3,column=3)


lblTotalCost= Label(f1, font=('Courier New', 16, 'bold'),text="Total
Cost",bd=16,anchor="w")

lblTotalCost.grid(row=4, column=2)

txtTotalCost=Entry(f1, font=('Courier
New',16,'bold'),textvariable=Total,bd=10,insertwidth=4,bg="White",justify='r
ight')

txtTotalCost.grid(row=4,column=3)
```

```
#       -->         BUTTONS    <--


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


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



root.mainloop()
```

CHAPTER-3

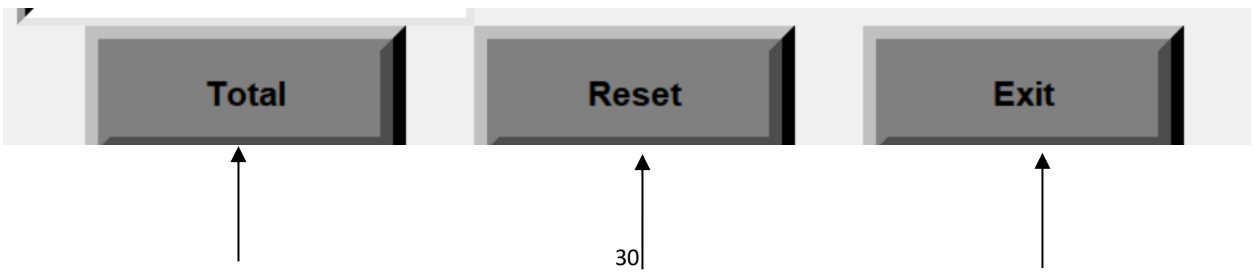
APLLICATION INTERFACE

INTERFACE:



SOME FEATURES:

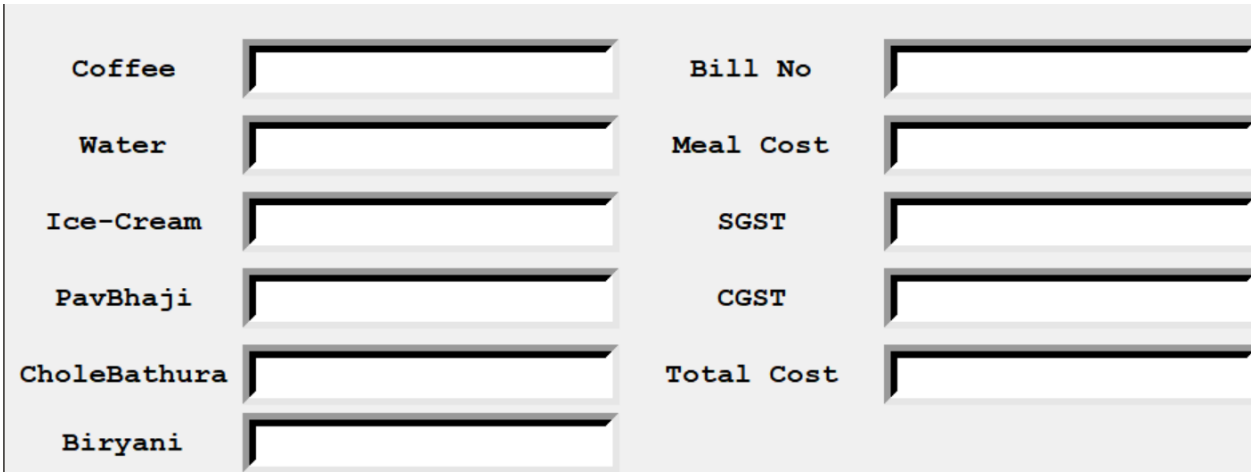
1) BUTTONS :



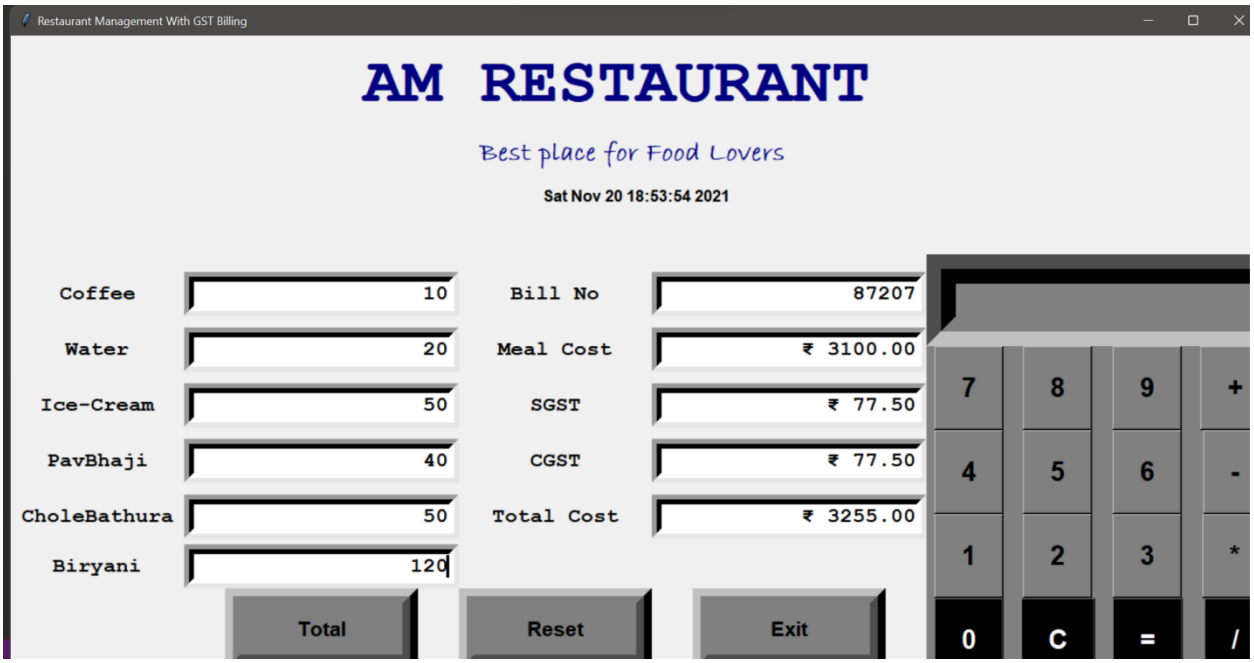
Total button gives the calculation of

Bill	Reset button resets all the values	Exit button closes the window

2) MENU ITEM AND BILL GRENRATION :



Let’s insert some value and check the apllication :



3) Calculator


20+40
7
8
9
+
4
5
6
-
1
2
3
*
0
C
=
/


60
7
8
9
+
4
5
6
-
1
2
3
*
0
C
=
/

CHAPTER-4

CONCLUSION

The coding of Restaurant Managment Syste was challenging with many errors arising. Many systems had to be written numerous ways before a final working solution was found.

This project has really been faithful and informative. It has made us learn and understand the many trivial concepts of Python Language. As we have used python Tkinter as a GUI it provides various controls, such as buttons, labels and text boxes to build a user friendly application.

The fast growing use of internet confirms the good future and scope of the proposed project.

Finally it has taught us a valuable lifelong lesson about the improvements and working and interacting in a group.

LOVELY PROFESSIONAL UNIVERSITY

REFERENCES

Link

www.python.org

https://en.wikipedia.org/wiki/Python_(programming_language)

https://www.geeksforgeeks.org/python-gui-tkinter/#:~:text=Out%20of%20all%20the%20GUI,tkinter%20is%20an%20easy%20task.

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

https://www.tutorialspoint.com/python/python_gui_programming.htm