

```
from tkinter import *
import random
import time
from tkinter import ttk
from tkinter import messagebox
import mysql.connector as mysql
from tkinter import *
```

```
def system():
    root = Tk()
    root.geometry("1300x800")
    root.title("Restaurant Billing System")
```

```
def Database():
    global connectn, cursor
    connectn = mysql.connect(database="Restaurant",user="root",password="root")
    cursor = connectn.cursor()
```

```
    cursor.execute(
        "CREATE TABLE IF NOT EXISTS Restaurantrecords(ordno text,piz text,bur text,ice text,
dr text, ct text,sb text,tax text,sr text,tot text)")
```

```
orderno = StringVar()
pizza = StringVar()
burger = StringVar()
icecream = StringVar()
drinks = StringVar()
cost = StringVar()
subtotal = StringVar()
tax = StringVar()
service = StringVar()
total = StringVar()
```

```
def tottal():
```

```
    order = (orderno.get())
    pi = float(pizza.get())
    bu = float(burger.get())
    ice = float(icecream.get())
    dr = float(drinks.get())
```

```
costpi = pi * 14
costbu = bu * 4
costice = ice * 6
costdr = dr * 2
```

```
costofmeal = (costpi + costbu + costice + costdr)
ptax = ((costpi + costbu + costice + costdr) * 0.18)
sub = (costpi + costbu + costice + costdr)
ser = ((costpi + costbu + costice + costdr) / 99)
paidtax = str(ptax)
Service = str(ser)
overall = str(ptax + ser + sub)
```

```
cost.set(costofmeal)
tax.set(ptax)
subtotal.set(sub)
service.set(ser)
total.set(overall)
```

```
def reset():
    ordeno.set("")
    pizza.set("")
    burger.set("")
    icecream.set("")
    drinks.set("")
    cost.set("")
    subtotal.set("")
    tax.set("")
    service.set("")
    total.set("")
```

```
def exit():
    root.destroy()
```

```
topframe = Frame(root, width=1300, height=50)
topframe.pack(side=TOP)
```

```
leftframe = Frame(root, width=800, height=700)
leftframe.pack(side=RIGHT)
```

```

rightframe = Frame(root, width=500, height=700)
rightframe.pack(side=LEFT)

##### display data #####
def DisplayData():
    Database()
    my_tree.delete(*my_tree.get_children())
    cursor = connectn.execute("SELECT * FROM Restaurantrecords")
    fetch = cursor.fetchall()
    for data in fetch:
        my_tree.insert("", 'end', values=(data))
    cursor.close()
    connectn.close()

style = ttk.Style()
style.configure("Treeview",
                foreground="black",
                rowheight=40,
                fieldbackground="white"
                )
style.map('Treeview',
         background=[('selected', 'lightblue')])

##### Creating table #####
my_tree = ttk.Treeview(rightframe)
my_tree['columns'] = ("ordno", "piz", "bur", "ice", "dr", "ct", "sb", "tax", "sr", "tot")

##### creating for table #####
horizontal_bar = ttk.Scrollbar(rightframe, orient="horizontal")
horizontal_bar.configure(command=my_tree.xview)
my_tree.configure(xscrollcommand=horizontal_bar.set)
horizontal_bar.pack(fill=X, side=BOTTOM)

vertical_bar = ttk.Scrollbar(rightframe, orient="vertical")
vertical_bar.configure(command=my_tree.yview)
my_tree.configure(yscrollcommand=vertical_bar.set)
vertical_bar.pack(fill=Y, side=RIGHT)

# defining column for table
my_tree.column("#0", width=0, minwidth=0)
my_tree.column("ordno", anchor=CENTER, width=80, minwidth=25)
my_tree.column("piz", anchor=CENTER, width=60, minwidth=25)
my_tree.column("bur", anchor=CENTER, width=50, minwidth=25)

```

```

my_tree.column("ice", anchor=CENTER, width=80, minwidth=25)
my_tree.column("dr", anchor=CENTER, width=50, minwidth=25)
my_tree.column("ct", anchor=CENTER, width=50, minwidth=25)
my_tree.column("sb", anchor=CENTER, width=100, minwidth=25)
my_tree.column("tax", anchor=CENTER, width=50, minwidth=25)
my_tree.column("sr", anchor=CENTER, width=100, minwidth=25)
my_tree.column("tot", anchor=CENTER, width=50, minwidth=25)

```

defining headings for table

```

my_tree.heading("ordno", text="Order No", anchor=CENTER)
my_tree.heading("piz", text="Pizza", anchor=CENTER)
my_tree.heading("bur", text="Burger", anchor=CENTER)
my_tree.heading("ice", text="Ice cream", anchor=CENTER)
my_tree.heading("dr", text="Drinks", anchor=CENTER)
my_tree.heading("ct", text="Cost", anchor=CENTER)
my_tree.heading("sb", text="Subtotal", anchor=CENTER)
my_tree.heading("tax", text="Tax", anchor=CENTER)
my_tree.heading("sr", text="Service", anchor=CENTER)
my_tree.heading("tot", text="Total", anchor=CENTER)

```

```
my_tree.pack()
```

```
DisplayData()
```

defining add function to add record

```
def add():
```

```
    Database()
```

```
    # getting data
```

```
    orders = orderno.get()
```

```
    pizzas = pizza.get()
```

```
    burgers = burger.get()
```

```
    ices = icecream.get()
```

```
    drinkss = drinks.get()
```

```
    costs = cost.get()
```

```
    subtotals = subtotal.get()
```

```
    taxes = tax.get()
```

```
    services = service.get()
```

```
    totals = total.get()
```

```
    if orders == "" or pizzas == "" or burgers == "" or ices == "" or drinkss == "" or costs == "" or
    subtotals == "" or taxes == "" or services == "" or totals == "":
```

```
        messagebox.showinfo("Warning", "Please fill the empty field!!!")
```

```
    else:
```

```
        connectn.execute(
```

```
            'INSERT INTO Restaurantrecords (ordno, piz, bur , ice ,dr ,ct ,sb ,tax, sr, tot) VALUES
            (?, ?, ?, ?, ?, ?, ?, ?, ?)',
```

```

        (orders, pizzas, burgers, ices, drinkss, costs, subtotals, taxes, services, totals));
    connectn.commit()
    messagebox.showinfo("Message", "Stored successfully")
# refresh table data
DisplayData()
connectn.close()

# defining function to access data from sqlite database
def DisplayData():
    Database()
    my_tree.delete(*my_tree.get_children())
    cursor = connectn.execute("SELECT * FROM Restaurantrecords")
    fetch = cursor.fetchall()
    for data in fetch:
        my_tree.insert("", 'end', values=(data))
    cursor.close()
    connectn.close()

# defining function to delete record
def Delete():
    # open database
    Database()
    if not my_tree.selection():
        messagebox.showwarning("Warning", "Select data to delete")
    else:
        result = messagebox.askquestion('Confirm', 'Are you sure you want to delete this
record?',
                                     icon="warning")
        if result == 'yes':
            curItem = my_tree.focus()
            contents = (my_tree.item(curItem))
            selecteditem = contents['values']
            my_tree.delete(curItem)
            cursor = connectn.execute("DELETE FROM Restaurantrecords WHERE ordno= %d" %
selecteditem[0])
            connectn.commit()
            cursor.close()
            connectn.close()

# Time
localtime = time.asctime(time.localtime(time.time()))
# Top part
main_lbl = Label(topframe, font=('Arial', 25, 'bold'), text="Restaurant Billing System",
fg="Blue",

```

```

        anchor=W)
main_lbl.grid(row=0, column=0)
main_lbl = Label(topframe, font=('Arial', 15,), text=localtime, fg="red", anchor=W)
main_lbl.grid(row=1, column=0)

#### Labels
# items
ordlbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Order No.", fg="black", bd=5,
anchor=W).grid(row=1,
column=0)
ordtxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=orderno).grid(row=1, column=1)
# Pizza
pizlbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Pizza", fg="black", bd=5,
anchor=W).grid(row=2,
column=0)
piztxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=pizza).grid(row=2, column=1)
# burger
burlbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Burger", fg="black", bd=5,
anchor=W).grid(row=3,
column=0)
burtxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=burger).grid(row=3, column=1)

# icecream
icelbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Ice Cream", fg="black", bd=5,
anchor=W).grid(row=4,
column=0)
icetxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=icecream).grid(row=4, column=1)
# drinks
drinklbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Drinks", fg="black", bd=5,
anchor=W).grid(row=5,
column=0)
drinkttx = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=drinks).grid(row=5, column=1)
# cost
costlbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Cost", bd=5, anchor=W).grid(row=6,
column=0)
costtxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
textvariable=cost).grid(row=6, column=1)
# subtotal
sublbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Subtotal", bd=5,

```

```

anchor=W).grid(row=7, column=0)
    subtxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
                    textvariable=subtotal).grid(row=7, column=1)
    # tax
    taxlbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Tax", bd=5, anchor=W).grid(row=8,
column=0)
    taxtxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
                    textvariable=tax).grid(row=8, column=1)
    # service
    servicelbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Service", bd=5,
anchor=W).grid(row=9,
                    column=0)
    servicetxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
                        textvariable=service).grid(row=9, column=1)
    # total
    totallbl = Label(leftframe, font=('Arial', 16, 'bold'), text="Total", bd=5, anchor=W).grid(row=10,
                    column=0)
    totaltxt = Entry(leftframe, font=('Arial', 16, 'bold'), bd=6, insertwidth=4, justify='right',
                        textvariable=total).grid(row=10, column=1)
    # ---button--

    totbtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Total", bg="Lightgrey", fg="black",
bd=3, padx=5, pady=5,
                    width=6, command=tottal).grid(row=6, column=3)

    resetbtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Reset", bg="lightgrey", fg="black",
bd=3, padx=5,
                    pady=5, width=6, command=reset).grid(row=4, column=3)

    exitbtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Exit", bg="lightgrey", fg="black",
bd=3, padx=5,
                    pady=5, width=12, command=exit).grid(row=6, column=2)

    addbtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Add", bg="lightgrey", fg="black",
bd=3, padx=5, pady=5,
                    width=6, command=add).grid(row=2, column=3)

    deletebtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Delete Record", bg="lightgrey",
fg="black", bd=3,
                    padx=5, pady=5, width=12, command=Delete).grid(row=4, column=2)

##### feedback form #####

```

```

def feedbackk():
    feed = Tk()
    feed.geometry("600x500")
    feed.title("Submit Feedback form")

    connectn = sqlite3.connect("Restaurant.db")
    cursor = connectn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS FEEDBACK(n text,eid text,feedback5
text,com text)")

    name = StringVar()
    email = StringVar()
    comments = StringVar()

def submit():
    n = name.get()
    eid = email.get()
    com = txt.get('1.0', END)
    feedback1 = ""
    feedback2 = ""
    feedback3 = ""
    feedback4 = ""
    if (checkvar1.get() == "1"):
        feedback1 = "Excellent"
    if (checkvar2.get() == "1"):
        feedback2 = "Good"
    if (checkvar3.get() == "1"):
        feedback2 = "Average"
    if (checkvar4.get() == "1"):
        feedback2 = "Poor"
    feedback5 = feedback1 + " " + feedback2 + " " + feedback3 + " " + feedback4
    conn = sqlite3.connect("Restaurant.db")
    cursor = conn.cursor()
    cursor.execute("INSERT INTO FEEDBACK VALUES ('" + n + "','" + eid + "','" + com + "','"
+ feedback5 + "')")
    messagebox.showinfo("message", "data inserted !")
    feed.destroy()

def cancel():
    feed.destroy()

lb1 = Label(feed, font=("Calisto MT", 15, "bold"), text="Thanks for Visiting!",

```



```

fg="black").pack(side=TOP)
    lbl2 = Label(feed, font=("calisto MT", 15), text="We're glad you chose us ! Please tell us
how it was!",
                fg="black").pack(side=TOP)

    nameLbl = Label(feed, font=('vardana', 15), text="Name:-", fg="black", bd=10,
anchor=W).place(x=10, y=150)
    nameTxt = Entry(feed, font=('vardana', 15), bd=6, insertwidth=2, bg="white", justify='right',
textvariable=name).place(x=15, y=185)

    emailLbl = Label(feed, font=('vardana', 15), text="Email:-", fg="black", bd=10,
anchor=W).place(x=280, y=150)
    emailTxt = Entry(feed, font=('vardana', 15), bd=6, insertwidth=2, bg="white", justify='right',
textvariable=email).place(x=285, y=185)

    ratelbl = Label(feed, font=('vardana', 15), text="How would you rate us?", fg="black",
bd=10, anchor=W).place(
    x=10, y=215)
    checkvar1 = StringVar()
    checkvar2 = StringVar()
    checkvar3 = StringVar()
    checkvar4 = StringVar()
    c1 = Checkbutton(feed, font=('Arial', 10, "bold"), text="Excellent", bg="white",
variable=checkvar1)
    c1.deselect()
    c1.place(x=15, y=265)
    c2 = Checkbutton(feed, font=('Arial', 10, "bold"), text="Good", bg="white",
variable=checkvar2, )
    c2.deselect()
    c2.place(x=120, y=265)
    c3 = Checkbutton(feed, font=('Arial', 10, "bold"), text=" Average", bg="white",
variable=checkvar3, )
    c3.deselect()
    c3.place(x=220, y=265)
    c4 = Checkbutton(feed, font=('Arial', 10, "bold"), text=" Poor ", bg="white",
variable=checkvar4, )
    c4.deselect()
    c4.place(x=320, y=265)

    commentslbl = Label(feed, font=('Arial', 15), text="Comments", fg="black", bd=10,
anchor=W).place(x=10, y=300)
    txt = Text(feed, width=50, height=5)
    txt.place(x=15, y=335)

```

```

submit = Button(feed, font=("Arial", 14), text="Submit", fg="black", bg="green", bd=2,
command=submit).place(
    x=145, y=430)
cancel = Button(feed, font=("Arial", 14), text="Cancel", fg="black", bg="red", bd=2,
command=cancel).place(
    x=245, y=430)
feed.mainloop()

feedbtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Feedback Form", fg="black",
bg="lightgrey", bd=3, padx=10,
    pady=10, width=10, command=feedbackk).grid(row=8, column=2, columnspan=1)

def menu():
    roott = Tk()
    roott.title("Price Menu")
    roott.geometry("300x300")
    lblinfo = Label(roott, font=("Arial", 20, "bold"), text="ITEM LIST", fg="black", bd=10)
    lblinfo.grid(row=0, column=0)
    lblprice = Label(roott, font=("Arial", 20, "bold"), text="Prices", fg="black", bd=10)
    lblprice.grid(row=0, column=3)
    lblpizza = Label(roott, font=("Arial", 20, "bold"), text="Pizza", fg="Blue", bd=10)
    lblpizza.grid(row=1, column=0)
    lblpricep = Label(roott, font=("Arial", 20, "bold"), text="14$", fg="blue", bd=10)
    lblpricep.grid(row=1, column=3)
    lblburger = Label(roott, font=("Arial", 20, "bold"), text="Burger", fg="Blue", bd=10)
    lblburger.grid(row=3, column=0)
    lblpriceb = Label(roott, font=("Arial", 20, "bold"), text="4$", fg="blue", bd=10)
    lblpriceb.grid(row=3, column=3)
    lbliccream = Label(roott, font=("Arial", 20, "bold"), text="Ice-Cream", fg="Blue", bd=10)
    lbliccream.grid(row=4, column=0)
    lblpricei = Label(roott, font=("Arial", 20, "bold"), text="6$", fg="blue", bd=10)
    lblpricei.grid(row=4, column=3)
    lbldrinks = Label(roott, font=("Arial", 20, "bold"), text="Drinks", fg="Blue", bd=10)
    lbldrinks.grid(row=5, column=0)
    lblpriced = Label(roott, font=("Arial", 20, "bold"), text="2$", fg="blue", bd=10)
    lblpriced.grid(row=5, column=3)
    roott.mainloop()

menubtn = Button(leftframe, font=('Arial', 14, 'bold'), text="Menu", bg="lightgrey", fg="black",
bd=3, padx=5,
    pady=6, width=12, command=menu).grid(row=2, column=2)

```

```
root.mainloop()
```

```
system()
```