

Project Report on **Cafe Billing System**

Priyanshu Kumar

CSE 3rd Sem, UCEM

11th March, 2022

Submitted to Head of Department CSE Mr. Jeetesh Srivastava



Table of Content

1. Table of Content
2. Abstract
3. Introduction
4. Objective & Features of the project
5. System design of the project
6. User Interface Design
7. Tools and Platform - Software Requirements
8. Tools and Platform - Hardware Requirements
9. Code
10. Conclusion of the project
11. Limitation of the project
12. Bibliography and references



Abstract of the Project

The purpose of Cafe billing system is to automate the existing manual system by the help of computerized equipment and full fledged computer software, fulfilling their requirements so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.



Introduction

The “Cafe Billing System” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by the existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data.

It also provides error messages while entering Invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user friendly, Cafe Billing System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather than concentrating on the record keeping thus it will help organizations in better utilization of resources.



Objective & Features

The main objective of the project on Cafe Billing System is to manage the details of sales, unit of commodity, bill, bill taxes etc. It manages all the information about Cafe Billings System sales purchases cases the project is totally built at the administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Cafe building system.

- It can hold numerous amount of bills
- It can search bills from your repository
- It can save new bills in your repository
- It can show you existing bills
- It can hold upto 18 articles to be sold on cafe
- It can show total values of 3 types of articles
- It can show the individual total of 3 types of articles
- It can show the tax @ 12
- It can show the service tax @ 6
- It can handle discount if any



System design of the project

The project mainly uses file handling to perform basic operations like to save a bill or retrieve a bill the source code of cafe management system is around 350 lines, to enhance the efficiency of the system

This Mini project is built with graphics to make it more intuitive with simple file handling operations the key features in Cafe Billing System are

- Enter a customer details
 - Name
 - Phone Number
- Enter the commodity purchased by the customer

And that is all, now it will total the amount and generate an invoice which can be saved or discarded according to the need.

1. Enter Customer details

It asks details of the customer like Name and Phone number to use it for future references.

2. Enter the billing items

Now it will ask the items purchased by the customer under 3 categories

- Beverages
- Meal
- Desert

Which can easily be modified to be used by other shops.

Page 10 of 10

—

—

—

Cafe Severney Billing Portal

—

—

Cafe Severney

—

—

Customer Details

Customer Name

Priyanshu Kumar

Phone No.

91 79057 96473

Bill No.

Search

Beverages

Meal

Desert

Billing Area

Tea

1

Sandwich

1

Brownie

0

Article

QTY

Price

Coffee

0

Burger

0

Chocolate

0

Tea

1

15

Espresso

0

Soup

0

Puddings

0

Water

1

10

Softdrink

0

Noodles

0

Waffle

0

Sanwich

1

35

Smoothie

0

Momos

0

Swiss Roll

1

45

Water

1

0

Fries

0

Swiss Roll

1

45

Apple Pie

0

Number of Beverages:

2

Number of Meals:

1

Number of Deserts:

1

Total Quantity:

4

Taxable Value:

Rs. 105.0

Tax:

12.6

Service Tax:

6.3

Discount:

12.39

Bill Menu

Total Beverages Cost

Rs. 25.0

Tax @ 12

Rs. 12.6

Total Meal Cost

Rs. 35.0

Service Tax @ 6

Rs. 6.3

Total Desert Cost

Rs. 45.0

Discount @ 10

Rs. 12.39

Total

Generate Bill

Save/Print

Cafe Severney Billing Portal

Cafe Severney

Customer Details

Customer Name
Phone No.
Bill No.

Beverages

Tea
Coffee
Espresso
Softdrink
Smoothie
Water

Meal

Sandwich
Burger
Soup
Noodles
Momos
Fries

Desert

Brownie
Chocolate
Swiss Roll
Apple Pie

Billing Area

Article	QTY	Price
Tea	1	15
Water	1	10
Sanwich	1	35
Swiss Roll	1	45
Number of Beverages:		2
Number of Meals:		1
Number of Deserts:		1
Total Quantity:		4
Taxable Value:		Rs. 105.0
Tax:		12.6
Service Tax:		6.3
Discount:		12.39

Bill Menu

Total Beverages Cost
Tax @ 12

Total Meal Cost
Service Tax @ 6

Total Desert Cost
Discount @ 10

Save Bill

Do you want to save the Bill?

Yes

No

Cafe Severney Billing Portal

Cafe Severney

Customer Details

Customer Name
Phone No.
Bill No.

Beverages

Tea
Coffee
Espresso
Softdrink
Smoothie
Water

Meal

Sandwich
Burger
Soup
Noodles
Momos
Fries

Desert

Brownie
Chocolate
Daddings
Apple Pie

Billing Area

Cafe Severney
Tax Invoice

Bill Number: 878
Customer Name: Priyanshu Kumar
Phone Number: 7905796473

Bill Description

Article	QTY	Price
Swiss Roll	2	90
Number of Beverages:		0
Number of Meals:		0
Number of Deserts:		2
Total Quantity:		2
Taxable Value:		Rs. 90.0

Bill Menu

Total Beverages Cost
Tax @ 12

Total Meal Cost
Service Tax @ 6

Total Desert Cost
Discount @ 10

Save

Bill Number : 878 saved successfully!

OK

7



Tools and Platform - Software Requirements

Language Used

- Python 3.9.7

Environment Used

- Visual Studio Code

Libraries Used

- tkinter
- turtle
- OS
- textwrap
- posixpath



Tools and Platform - Hardware Requirements

Minimum System Requirement Windows

- Windows 7
- 2 GB RAM
- 64 GB Secondary Storage

Minimum System Requirement Mac

- Catalina
- 1 GB RAM
- 128 GB Secondary Storage

Code

```
from email import message
import math, random
from posixpath import split
from tkinter import messagebox
from cgitb import text
from textwrap import fill
from tkinter import*
from tkinter import font
from turtle import bgcolor, color, onclick, onkeypress, width
import os

from setuptools import Command

class Bill_App:
    def __init__(self, root):
        self.root = root
        self.root.geometry("1350x712+0+0")
        self.root.title("Cafe Severney Billing Portal")
        bg_color = "#074463"
        title = Label(self.root, text="Cafe Severney", bd=12, fg="white",
relief=GROOVE, bg=bg_color, font=(
    "Times new roman", 30, "bold"), pady=3).pack(fill=X)

#=====Variable=====#
#=====Beverages Variables=====#
        self.tea=IntVar()
        self.coffee=IntVar()
        self.espresso=IntVar()
        self.softdrink=IntVar()
        self.smoothie=IntVar()
        self.water=IntVar()
#=====Meal Variable=====#
        self.sandwich=IntVar()
        self.burger=IntVar()
        self.soup=IntVar()
        self.noodles=IntVar()
        self.momos=IntVar()
        self.fries=IntVar()
#=====Desert Variable=====#
        self.brownie=IntVar()
        self.chocolate=IntVar()
        self.puddings=IntVar()
        self.waffle=IntVar()
        self.roll=IntVar()
        self.applepie=IntVar()
#=====Total Prices and Taxes=====#
        self.beverages_cost=StringVar()
```

```

self.meal_cost=StringVar()
self.desert_cost=StringVar()

self.tax=IntVar()
self.service_tax=IntVar()
self.discount=IntVar()

#=====Customer Variables=====#
self.c_name=StringVar()
self.c_phone=StringVar()
self.bill_no=StringVar()
x=random.randint(100,999)
self.bill_no.set(str(x))
self.search_bill=StringVar()

#=====Customer Detail Frame=====#
F1=LabelFrame(self.root,border=10,relief=GROOVE,text="Customer
Details",font=("times new roman",15,"bold"),fg="gold",bg=bg_color)
F1.place(x=0,y=80,relwidth=1)

cname_lbl=Label(F1,text="Customer Name",bg=bg_color,fg="white",font=("times
new roman",18,"bold")).grid(row=0,column=0,padx=20,pady=5)

cname_txt=Entry(F1,textvariable=self.c_name,width=22,font=("arial",15),bd=4,relief
=SUNKEN).grid(row=0,column=1,padx=5,pady=10)

cphn_lbl=Label(F1,text="Phone No.",bg=bg_color,fg="white",font=("times new
roman",18,"bold")).grid(row=0,column=2,padx=20,pady=5)

cphn_txt=Entry(F1,textvariable=self.c_phone,width=18,font=("arial",15),bd=4,relief
=SUNKEN).grid(row=0,column=3,padx=5,pady=10)

cbill_lbl=Label(F1,text="Bill No.",bg=bg_color,fg="white",font=("times new
roman",18,"bold")).grid(row=0,column=4,padx=20,pady=5)

cbill_txt=Entry(F1,textvariable=self.search_bill,width=15,font=("arial",15),bd=4,r
elief=SUNKEN).grid(row=0,column=5,padx=5,pady=10)

billbtn=Button(F1,text="Search",command=self.find_bill,width=10,border=7,font="Ari
al 12 bold").grid(row=0,column=6,pady=12,padx=35)

#=====Beverages=====#

F2=LabelFrame(self.root,border=10,relief=GROOVE,text="Beverages",font=("times new
roman",15,"bold"),fg="gold",bg=bg_color)
F2.place(x=6,y=188,width=325,height=380)

tea_lbl=Label(F2,text="Tea",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=0,column=0,padx=10,pady=10
,sticky="w")

tea_txt=Entry(F2,textvariable=self.tea,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=0,column=1,padx=10,pady=10)

```

```

        coffee_lbl=Label(F2,text="Coffee",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=1,column=0,padx=10,pady=10
,sticky="w")
        coffee_txt=Entry(F2,textvariable=self.coffee,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=1,column=1,padx=10,pady=10)

        espresso_lbl=Label(F2,text="Espresso",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=2,column=0,padx=10,pady=10
,sticky="w")
        espresso_txt=Entry(F2,textvariable=self.espresso,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=2,column=1,padx=10,pady=10)

        softdrink_lbl=Label(F2,text="Softdrink",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=3,column=0,padx=10,pady=10
,sticky="w")
        softdrink_txt=Entry(F2,textvariable=self.softdrink,width=10,font=("times
new roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=3,column=1,padx=10,pady=10)

        smoothie_lbl=Label(F2,text="Smoothie",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=4,column=0,padx=10,pady=10
,sticky="w")
        smoothie_txt=Entry(F2,textvariable=self.smoothie,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=4,column=1,padx=10,pady=10)

        water_lbl=Label(F2,text="Water",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=5,column=0,padx=10,pady=10
,sticky="w")
        water_txt=Entry(F2,textvariable=self.water,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=5,column=1,padx=10,pady=10)

        #=====Meal=====#
        F3=LabelFrame(self.root,border=10,relief=GROOVE,text="Meal",font=("times
new roman",15,"bold"),fg="gold",bg=bg_color)
        F3.place(x=341,y=188,width=325,height=380)

        sandwich_lbl=Label(F3,text="Sandwich",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=0,column=0,padx=10,pady=10
,sticky="w")
        sandwich_txt=Entry(F3,textvariable=self.sandwich,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=0,column=1,padx=10,pady=10)

        burger_lbl=Label(F3,text="Burger",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=1,column=0,padx=10,pady=10
,sticky="w")
        burger_txt=Entry(F3,textvariable=self.burger,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=1,column=1,padx=10,pady=10)

        soup_lbl=Label(F3,text="Soup",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=2,column=0,padx=10,pady=10
,sticky="w")
        soup_txt=Entry(F3,textvariable=self.soup,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=2,column=1,padx=10,pady=10)

        noodles_lbl=Label(F3,text="Noodles",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=3,column=0,padx=10,pady=10
,sticky="w")

```

```

noodles_txt=Entry(F3,textvariable=self.noodles,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=3,column=1,padx=10,pady=10)

momos_lbl=Label(F3,text="Momos",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=4,column=0,padx=10,pady=10
,sticky="w")
momos_txt=Entry(F3,textvariable=self.momos,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=4,column=1,padx=10,pady=10)

fries_lbl=Label(F3,text="Fries",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=5,column=0,padx=10,pady=10
,sticky="w")
fries_txt=Entry(F3,textvariable=self.fries,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=5,column=1,padx=10,pady=10)

#=====Desert=====#
F4=LabelFrame(self.root,border=10,relief=GROOVE,text="Desert",font=("times
new roman",15,"bold"),fg="gold",bg=bg_color)
F4.place(x=675,y=188,width=325,height=380)

brownie_lbl=Label(F4,text="Brownie",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=0,column=0,padx=10,pady=10
,sticky="w")
brownie_txt=Entry(F4,textvariable=self.brownie,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=0,column=1,padx=10,pady=10)

chocolate_lbl=Label(F4,text="Chocolate",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=1,column=0,padx=10,pady=10
,sticky="w")
chocolate_txt=Entry(F4,textvariable=self.chocolate,width=10,font=("times
new roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=1,column=1,padx=10,pady=10)

puddings_lbl=Label(F4,text="Puddings",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=2,column=0,padx=10,pady=10
,sticky="w")
puddings_txt=Entry(F4,textvariable=self.puddings,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=2,column=1,padx=10,pady=10)

waffle_lbl=Label(F4,text="Waffle",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=3,column=0,padx=10,pady=10
,sticky="w")
waffle_txt=Entry(F4,textvariable=self.waffle,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=3,column=1,padx=10,pady=10)

roll_lbl=Label(F4,text="Swiss Roll",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=4,column=0,padx=10,pady=10
,sticky="w")
roll_txt=Entry(F4,textvariable=self.roll,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=4,column=1,padx=10,pady=10)

applepie_lbl=Label(F4,text="Apple Pie",font=("times new
roman",16,"bold"),bg=bg_color,fg="lightgreen").grid(row=5,column=0,padx=10,pady=10
,sticky="w")
applepie_txt=Entry(F4,textvariable=self.applepie,width=10,font=("times new
roman",16,"bold"),bd=5,relief=SUNKEN).grid(row=5,column=1,padx=10,pady=10)

```

```

#=====Bill AREA=====#
F5=Frame(self.root,bd=10,relief=GROOVE)
F5.place (x=1008,y=188,width=335,height=380)
bill_title = Label(F5,text="Billing Area",font ="arial 15
bold",bd=7,relief=GROOVE).pack(fill=X)
scrol_y=Scrollbar(F5,orient=VERTICAL)
self.txtarea=Text(F5,yscrollcommand=scrol_y.set)
scrol_y.pack(side=RIGHT,fill=Y)
scrol_y.config(command=self.txtarea.yview)
self.txtarea.pack(fill=BOTH)

#=====Button Frame=====#
F6=LabelFrame(self.root,border=10,relief=GROOVE,text="Bill
Menu",font=("times new roman",15,"bold"),fg="gold",bg=bg_color)
F6.place(x=0,y=574,relwidth=1,height=138)

m1_lbl=Label(F6,text="Total Beverages
Cost",bg=bg_color,fg="white",font=("times new
roman",14,"bold")).grid(row=0,column=0,padx=20,pady=2,sticky="w")
m1_txt=Entry(F6,textvariable=self.beverages_cost,width=18,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=0,column=1, padx=10,pady=1)

m2_lbl=Label(F6,text="Total Meal Cost",bg=bg_color,fg="white",font=("times
new roman",14,"bold")).grid(row=1,column=0,padx=20,pady=2,sticky="w")
m2_txt=Entry(F6,textvariable=self.meal_cost,width=18,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=1,column=1, padx=10,pady=1)

m3_lbl=Label(F6,text="Total Desert
Cost",bg=bg_color,fg="white",font=("times new
roman",14,"bold")).grid(row=2,column=0,padx=20,pady=2,sticky="w")
m3_txt=Entry(F6,width=18,textvariable=self.desert_cost,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=2,column=1, padx=10,pady=1)

c1_lbl=Label(F6,text="Tax @ 12",bg=bg_color,fg="white",font=("times new
roman",14,"bold")).grid(row=0,column=2,padx=20,pady=2,sticky="w")
c1_txt=Entry(F6,textvariable=self.tax,width=18,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=0,column=3, padx=10,pady=1)

c2_lbl=Label(F6,text="Service Tax @ 6",bg=bg_color,fg="white",font=("times
new roman",14,"bold")).grid(row=1,column=2,padx=20,pady=2,sticky="w")
c2_txt=Entry(F6,textvariable=self.service_tax,width=18,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=1,column=3, padx=10,pady=1)

c3_lbl=Label(F6,text="Discount @ 10",bg=bg_color,fg="white",font=("times
new roman",14,"bold")).grid(row=2,column=2,padx=20,pady=2,sticky="w")
c3_txt=Entry(F6,textvariable=self.discount,width=18,font="arial 10
bold",border=3,relief=SUNKEN).grid(row=2,column=3, padx=10,pady=1)

btn_F=Frame(F6,bd=3,relief=GROOVE)
btn_F.place(x=758,width=560,height=97)

total_btn=Button(btn_F,command=self.total,text="Total",bg="cadetblue",fg="black",p
ady=5,padx=10,width=14,bd=3,height=3,font="arial 12
bold").grid(row=0,column=0,padx=7,pady=5)

```

```

total_btn=Button(btn_F,command=self.bill_area,text="Generate
Bill",bg="cadetblue",fg="black",pady=5,padx=10,width=14,bd=3,height=3,font="arial
12 bold").grid(row=0,column=1,padx=7,pady=5)

total_btn=Button(btn_F,text="Save/Print",command=self.save_bill,bg="cadetblue",fg=
"black",pady=5,width=14,padx=10,bd=3,height=3,font="arial 12
bold").grid(row=0,column=2,padx=7,pady=5)
self.welcome_bill()

def total(self):
    self.total_beverages_price=float(
        (self.tea.get()*15)+
        (self.coffee.get()*25)+
        (self.espresso.get()*35)+
        (self.softdrink.get()*30)+
        (self.smoothie.get()*45)+
        (self.water.get()*10)
    )
    self.beverages_cost.set("Rs. "+str(self.total_beverages_price))

    self.total_meal_price=float(
        (self.sandwich.get()*35)+
        (self.burger.get()*40)+
        (self.soup.get()*60)+
        (self.noodles.get()*70)+
        (self.momos.get()*7)+
        (self.fries.get()*40)
    )
    self.meal_cost.set("Rs. "+str(self.total_meal_price))

    self.total_desert_price=float(
        (self.brownie.get()*40)+
        (self.chocolate.get()*50)+
        (self.puddings.get()*35)+
        (self.waffle.get()*30)+
        (self.roll.get()*45)+
        (self.applepie.get()*60)
    )
    self.desert_cost.set("Rs. "+str(self.total_desert_price))

self.z=(self.total_beverages_price+self.total_desert_price+self.total_meal_price)*
.12

self.y=(self.total_beverages_price+self.total_desert_price+self.total_meal_price)*
.06

    self.tax.set("Rs. "+str(round(self.z,2)))
    self.service_tax.set("Rs. "+str(round(self.y,2)))

self.dis=((self.z+self.y+self.total_beverages_price+self.total_desert_price+self.t
otal_meal_price)*.1)
    self.discount.set("Rs. "+str(round(self.dis,2)))

def welcome_bill(self):
    self.txtarea.delete('1.0',END)

```



```

self.txtarea.insert(END,f"                                Cafe Severney\n                                Tax
Invoice\n-----\nBill Number:
{self.bill_no.get()}\nCustomer Name:    {self.c_name.get()}\nPhone Number:
{self.c_phone.get()}\n-----\n                                Bill
Description\n=====Article                                QTY
Price=====Article                                QTY\n")
def bill_area(self):
    if self.c_name.get()==" " or self.c_phone.get()==" " :
        messagebox.showerror("Error","Customer Details are Missing")
    else:
        self.welcome_bill()
        if self.tea.get()!=0:
            self.txtarea.insert(END,f"Tea                                {self.tea.get() }
{self.tea.get()*15}\n")

            if self.coffee.get()!=0:
                self.txtarea.insert(END,f"Coffee                                {self.coffee.get() }
{self.coffee.get()*25}\n")

                if self.espresso.get()!=0:
                    self.txtarea.insert(END,f"Espresso                                {self.espresso.get() }
{self.espresso.get()*35}\n")

                    if self.softdrink.get()!=0:
                        self.txtarea.insert(END,f"Softdrink                                {self.softdrink.get() }
{self.softdrink.get()*30}\n")

                        if self.smoothie.get()!=0:
                            self.txtarea.insert(END,f"Smoothie                                {self.smoothie.get() }
{self.smoothie.get()*45}\n")

                            if self.water.get()!=0:
                                self.txtarea.insert(END,f"Water                                {self.water.get() }
{self.water.get()*10}\n")

                                if self.sandwich.get()!=0:
                                    self.txtarea.insert(END,f"Sanwich                                {self.sandwich.get() }
{self.sandwich.get()*35}\n")

                                    if self.burger.get()!=0:
                                        self.txtarea.insert(END,f"Burger                                {self.burger.get() }
{self.burger.get()*70}\n")

                                        if self.soup.get()!=0:
                                            self.txtarea.insert(END,f"Soup                                {self.soup.get() }
{self.soup.get()*60}\n")

                                            if self.noodles.get()!=0:
                                                self.txtarea.insert(END,f"Noodles                                {self.noodles.get() }
{self.noodles.get()*70}\n")

                                                if self.momos.get()!=0:
                                                    self.txtarea.insert(END,f"Momo                                {self.momos.get() }
{self.momos.get()*7}\n")

                                                    if self.fries.get()!=0:

```

```

        self.txtarea.insert(END,f"Fries                                {self.fries.get() }
{self.fries.get()*40}\n")

        if self.brownie.get()!=0:
            self.txtarea.insert(END,f"Brownie                            {self.brownie.get() }
{self.brownie.get()*40}\n")

        if self.chocolate.get()!=0:
            self.txtarea.insert(END,f"Chocolate                        {self.chocolate.get() }
{self.chocolate.get()*50}\n")

        if self.puddings.get()!=0:
            self.txtarea.insert(END,f"Puddings                            {self.puddings.get() }
{self.puddings.get()*35}\n")

        if self.waffle.get()!=0:
            self.txtarea.insert(END,f"Waffle                                {self.waffle.get() }
{self.waffle.get()*30}\n")

        if self.roll.get()!=0:
            self.txtarea.insert(END,f"Swiss Roll                            {self.roll.get() }
{self.roll.get()*45}\n")

        if self.applepie.get()!=0:
            self.txtarea.insert(END,f"Apple Pie                            {self.applepie.get() }
{self.applepie.get()*60}\n")
            self.txtarea.insert(END,"=====")
            self.txtarea.insert(END,f"Number of Beverages:
{self.tea.get()+self.coffee.get()+self.espresso.get()+self.softdrink.get()+self.sm
oothie.get()+self.water.get()}\n")
            self.txtarea.insert(END,f"Number of Meals:
{(self.sandwich.get())+(self.burger.get())+(self.soup.get())+(self.noodles.get()*7
0)+(self.momos.get())+(self.fries.get())}\n")
            self.txtarea.insert(END,f"Number of Deserts:
{(self.brownie.get())+(self.chocolate.get())+(self.puddings.get())+(self.waffle.ge
t())+(self.roll.get())+(self.applepie.get()*60)}\n")

        self.txtarea.insert(END,f"\nTotal Quantity:
{self.tea.get()+self.coffee.get()+self.espresso.get()+self.softdrink.get()+self.sm
oothie.get()+self.water.get()+self.sandwich.get()+self.burger.get()+self.soup
.get()+self.noodles.get()*70)+(self.momos.get())+(self.fries.get())+(self.browni
e.get())+(self.chocolate.get())+(self.puddings.get())+(self.waffle.get())+(self.ro
ll.get())+(self.applepie.get()*60)}\n")
        self.txtarea.insert(END,"-----")
        self.txtarea.insert(END,f"Taxable Value:                Rs.
{round(((self.z*100)/12),2)}\n")
        self.txtarea.insert(END,f"Tax:
{round((self.z),2)}\n")
        self.txtarea.insert(END,f"Service Tax:
{round((self.y),2)}\n")
        self.txtarea.insert(END,f"Discount:
{round((self.dis),2)}\n")
        self.txtarea.insert(END,"=====")

        self.total_bill=(((self.z*100)/12)+self.y+self.z)-self.dis

```

```

        self.txtarea.insert(END,f"Total Bill:                Rs.
{round((self.total_bill),2)}\n")

def save_bill(self):
    op=messagebox.askyesno("Save Bill","Do you want to save the Bill?")
    if op>0:
        self.bill_data=self.txtarea.get('1.0',END)
        fl=open("bill/"+str(self.bill_no.get())+".txt","w")
        fl.write(self.bill_data)
        fl.close()
        messagebox.showinfo("Saved",f"Bill Number : {self.bill_no.get()} saved
successfully!")
    else:
        return
def find_bill(self):
    present ="no"
    for i in os.listdir("bill/"):
        if i.split('.')[0]==self.search_bill.get():
            fl=open(f"bill/{i}","r")
            self.txtarea.delete('1.0',END)
            for d in fl:
                self.txtarea.insert(END,d)
            fl.close()
            present="yes"
    if present=="no":
        messagebox.showerror("Error","Invalid Bill Number")

root = Tk()
obj = Bill_App(root)
root.mainloop()

```



Limitation

- It needs to be restarted to each time to generate a fresh entry
- While printing the indentation might get misaligned
- It is not resizable



Conclusion

The project Cafe Billing System is for computerizing the working in a cafe. It is a great improvement over the manual system. The computerizing of the system has sped up the process. The Cafe Billing System was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average cafe and is capable to provide easy and effective storage of information related to customers that come up to the hospitals.



Bibliography

- <https://stackoverflow.com/>
- https://www.tutorialspoint.com/python/python_gui_programming.htm#:~:text=Tkinter%20is%20the%20standard%20GUI,Tkinter%20is%20an%20easy%20task.
- <https://www.w3schools.com/python/>
- https://www.youtube.com/watch?v=_uQrJ0TkZlc
- <https://www.geeksforgeeks.org/python-gui-tkinter/>

