



Billing System project

Minor Project Report (S305(a))

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR IN COMPUTER APPLICATION (B.C.A)



PALE KUMAR (1800085011104)

HIMANSHU JADOUN (1800085011051)

MANISH PRATAP SINGH (1800085011079)

UNDER THE SUPERVISION OF

DR. MONIKA VARSHNEY

DEPARTMENT OF COMPUTER APPLICATION

DHARMA SAMAJ COLLEGE NEAR

ACHAL TANK, ALIGARH 2021

ACKNOWLEDGMENT

Keep away from people who try to belittle your ambitions. Small people always do that, but the really great make you feel that you too, can become great.

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

First of all, I would like to thanks **Dr. Monika Varshney, (D.S. College, Aligarh)** for creating opportunities to undertake me in the esteemed organization.

Special thanks to **Dr. Monika Varshney** Project Manager for all the help and guidance extended to me by him in every stage during my Training. His inspiring suggestion and timely guidance enabled me to perceive the various aspects of the project in a new light.

I would also thank to my parents & Project mate for guiding and encouraging me throughout the duration of the project.

Pale Kumar

Himanshu Jadoun

Manish Pratap Singh

Billing System project

(using Python)

INDEX

S.No	Topics	Page
1	Introduction.	
2	Objectives.	
3	Project Category.	
4	Tools/Platform.	
5	Hardware & Software Requirement.	
6	Future scope and further enhancement of the project.	
7	Source Code.	
8	Output.	

Introduction

The computer has brought revolution in every sphere of human life, whether it is business, education field, governance, medical science etc. The computer has reduced the human work load, businesses are going global and everything is available at the click of mouse. The concept of e-shopping has been introduced and we can buy the products online and make payments through credit or debit cards.

Presently I proposing the system "General Store Billing System". The general stores issue their client handwritten bills and they enter details in manual registers. And maintain MS Excel file for product rate. So the proposed system will computerised their manual bill generation system.

As stated above the general stores presently uses manual bills and hand written record to maintains their product list, customer list, and keep the invoice, there is lot of duplicate work, and chance of mistake. When the product prices are changed they need to update each and every hand written record.

There is no security; anybody can access any report and sensitive data, also there are no reports to find out the sales volume, stock list, and summary report. This Billing system is used to overcome the entire problem which the client is

facing currently, and making complete atomization of manual billing system.

Employee can submit bill of various amount. An email will be sent to the concerned people to let them know about the status of the bill. The main propose of this bill management system project is developing a system that automate the bill submission and bill approval task. In big organization bill submission is very tiresome work and maintaining the record of bill is very difficult and time consuming. In present system, user have to work manually to maintain bill records and it is very difficult know.

In present manual system, submitting the bills to their corresponding managers is a time consuming process and we have to maintain records manually. Some times in manual process, there is a possibility to get errors. To overcome all these problem, we have developed this bill management system.

Objectives

- Customer, Products, Billing Generation: Automate the current manual bill generation system and maintain the searchable customer, products database and product invoice, maintain the data security, user rights.
- Report Generation: A Report Generation system will be developed for the user of Billing and Invoicing System. This MIS system will have both details and summary type reports for analysis the sales volume, sales trend, available stock
- To develop a system for the management of sales, Purchase and stock maintenance processes that will be performed with a click of mouse button.
- To develop a system that has a good management of data along with integrity and minimizing redundancy.
- To develop a system that will be user friendly in all possible ways.
- To develop a system that provides easier work than existing system for the user.
- To develop a secure system that can be accessed only by authorised users.

Project Category:

DESKTOP APPLICATION

Tools/Platform:

1	Frontend	-Visual Studio Code(PYTHON). - PyCharm IDE.
2	Backend	-OS Module(File Handling)

Hardware and Software Requirement:

Hardware:

1	Processor	1.6 GHz or Faster
2	RAM	1GB or Higher
3	Hard Disk	40 GB or Higher

Software Requirement:

1	Operating System	-Window 7 or Higher
2	Frontend Software	-Visual Studio Code(PYTHON). - PyCharm IDE
3	Backend Software	-OS Module(File Handling)

Future scope and further enhancement of the project:

Software development life cycle process specifies a method of developing the software. Each software development projects starts with some needs and ends with some software that satisfies those needs. A software development life cycle specifies the set of activities that should be performed to go from user needs to final products. There are different models of SDLC process and each model specifies the activities and the order in which they should be performed. Depending on the nature of project, a suitable model is chosen and the entire process of software requirement analysis, design, coding, testing and maintenance is preformed accordingly.

An initial investigation culminates in a proposal that determines whether a system is feasible or not. It determines its workability, impact on the organization, ability to meet user needs, and effective user resources. The objective of feasibility study is not solve the problem but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, cost and benefits are estimated with greater accuracy at this stage. This is a bridge in between the User Requirements and the output that he can avail under a set of given constraints, inputs and outputs

Working

Billing System project is written in Python. The project file contains a python script (Billingsoftware.py).

This is a simple GUI based application which is very easy to understand and use. It uses Tkinter module for

the GUI. Talking about the application, the user just has to select among the Grocery, Cosmetic and Cold Drinks items, enter the quantity and click on the total button to view the total price.

The user can view the total receipt of their items which displays receipt number and number of their grocery/cosmetic/cold drinks items with the total amount. There's also an extra calculator feature for the users.

Here, the total bill of the customer includes tax and service charges too. The design is so simple that the user won't find any difficulties while working on it.

In order to run the project, you must have installed Python and on your PC. This is a simple food billing system application for beginners. Food Billing System in Python with source code is free to download. Use for education purpose only! For the project demo, have a look at the video below.

This is a Program to Generate a .txt Invoice with all the needed functionality such as

1. Invoice Generated in For.txt mat.

Here we are getting data from a UI for developing which i have used Tkinter there are fields such as

1. Departmental Store Name
2. Customer Name
3. Contact Number
4. Product Name
5. Product Amount
6. Service Tax

After filling all the fields and submitting it. It will directly store all the data in the database and create a .txt Invoice file and stores it in the available INVOICE folder and also sends a Email to the customer with the .txt Invoice file attached to it.

To start with you need to set up the Database with columns such as

After setting up the database run the server.py file

Source Code:

```
from tkinter import*

import tkinter.messagebox as tmsg

import os

import math,random

class Billing:

    def __init__(self,root):

        self.root=root

        self.root.geometry("1200x1200")

        self.root.title("Billing software")

        Heading=Label(self.root,text="Billing  Software",font=("times  new
roman",40,"bold"),bg="dark blue",border=9,relief=GROOVE,fg='yellow')

        Heading.place(relwidth=1)

        self.root.config(bg="black")

        # -----Variable Declaration-----#

        self.CustomerName1 =StringVar()

        self.Contact_No_1 =StringVar()

        self.Bill_No_1 = StringVar()

        x=random.randint(1000,9999)

        self.Bill_No_1.set(x)
```

```
self.Search_bill= StringVar()
```

```
# -----Variable Declaration F2-----#
```

```
self.Bath_Soap_1 = IntVar()
```

```
self.Face_Cream_1 =IntVar()
```

```
self.Face_Wash_1 =IntVar()
```

```
self.Hair_Spray_1 =IntVar()
```

```
self.Hair_Gel_1 =IntVar()
```

```
self.Body_Losan_1 =IntVar()
```

```
# -----Variable Declaration F3-----#
```

```
self.Rice_1 =IntVar()
```

```
self.Wheat_1 =IntVar()
```

```
self.Sugar_1 =IntVar()
```

```
self.Tea_1 =IntVar()
```

```
self.Daal_1 =IntVar()
```

```
self.Food_oil_1 =IntVar()
```

```
# -----Variable Declaration F4-----#
```

```
self.Maza_1 =IntVar()
```

```
self.Coca_1=IntVar()
```

```
self.Thumbs_up_1 =IntVar()
```

```

self.Fruti_1 =IntVar()

self.Sprite_1 =IntVar()

self.Limca_1 =IntVar()

# -----Variable Declaration F5-----#

self.Total_Cosmatic_1=IntVar()

self.Total_Grocery_1 = IntVar()

self.Total_colddrink_1 = IntVar()

self.Cosmatic_Tax_1=IntVar()

self.Grocery_Tax_1 = IntVar()

self.colddink_Tax_1 = IntVar()

#-----Frame_1-----#

F1=Frame(self.root,relief=GROOVE,bg="dark blue",border=7)

F1.place(x=0,y=90,relwidth=1)

Customer_Details=Label(F1,text="Customer      Details",font=("times
new roman",15,"bold"),bg="dark blue",fg='yellow')

Customer_Details.place(x=-1,y=-16)

Customer_Name=Label(F1,text="Customer Name",font=("times new
roman",17,"bold"),bg="dark blue",fg='white')

Customer_Name.grid(row=0,column=0,pady=20,padx=16)

```

```
C1=Entry(F1, font=("times new roman", 15, "bold"),relief=GROOVE,border=4,width=23,textvariable=self.CustomerName1)
```

```
C1.grid(row=0, column=1)
```

```
Contact_No=Label(F1,text="Contact No.",font=("times new roman",17,"bold"),bg="dark blue",fg='white')
```

```
Contact_No.grid(row=0,column=2,pady=20,padx=16)
```

```
C_1=Entry(F1, font=("times new roman", 15, "bold"),relief=GROOVE,border=4,width=23,textvariable=self.Contact_No_1)
```

```
C_1.grid(row=0, column=3)
```

```
Bill_No= Label(F1, text="Bill No.", font=("times new roman", 17, "bold"), bg="dark blue", fg='white')
```

```
Bill_No.grid(row=0, column=4, pady=20, padx=16)
```

```
B1 = Entry(F1, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=23,textvariable=self.Bill_No_1)
```

```
B1.grid(row=0, column=5,padx=16)
```

```
Search_butt=Button(F1,text="Search",font=("times new roman",15,"bold"),width=10,command=self.Search)
```

```
Search_butt.grid(row=0,column=6,padx=16)
```

```
# -----Frame_2-----#
```

```

F2 = Frame(self.root, relief=GROOVE, bg="dark blue", border=7)

F2.place(x=0, y=200, width=300, height=350)

Costmatic=Label(F2, text="Costmatic", font=("times new roman", 15,
"bold"), bg="dark blue",fg='yellow')

Costmatic.place(x=-1, y=-16)

Bath_Soap=Label(F2,text="Bath          Soap",fg='white',bg="dark
blue",font=("times new roman", 14, "bold"))

Bath_Soap.grid(row=0,column=0,pady=15,padx=16)

B_S1  =  Entry(F2,  font=("times  new  roman",  15,  "bold"),
relief=GROOVE, border=4,width=10,textvariable=self.Bath_Soap_1)

B_S1.grid(row=0, column=1)

Face_Cream = Label(F2, text="Face Cream", fg='white', bg="dark
blue", font=("times new roman", 14, "bold"))

Face_Cream.grid(row=1, column=0, pady=15, padx=16)

F_C1  =  Entry(F2,  font=("times  new  roman",  15,  "bold"),
relief=GROOVE, border=4,width=10,textvariable=self.Face_Cream_1)

F_C1.grid(row=1, column=1)

Face_Wash = Label(F2, text="Face Wash", fg='white', bg="dark blue",
font=("times new roman", 14, "bold"))

Face_Wash.grid(row=3, column=0, pady=10, padx=16)

F_W1  =  Entry(F2,  font=("times  new  roman",  15,  "bold"),
relief=GROOVE, border=4,width=10,textvariable=self.Face_Wash_1)

```



```
F_W1.grid(row=3, column=1)
```

```
Hair_Spray = Label(F2, text="Hair Spray", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Hair_Spray.grid(row=4, column=0, pady=15, padx=16)
```

```
H_S1= Entry(F2, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4,width=10,textvariable=self.Hair_Spray_1)
```

```
H_S1.grid(row=4, column=1)
```

```
Hair_Gel = Label(F2, text="Hair Gel", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Hair_Gel.grid(row=5, column=0,pady=15, padx=16)
```

```
H_G1 = Entry(F2, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4,width=10,textvariable=self.Hair_Gel_1)
```

```
H_G1.grid(row=5, column=1)
```

```
Body_Losan = Label(F2, text="Body Losan", fg='white', bg="dark  
blue", font=("times new roman", 14, "bold"))
```

```
Body_Losan.grid(row=6, column=0, pady=15, padx=16)
```

```
B_L1 = Entry(F2, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4,width=10,textvariable=self.Body_Losan_1)
```

```
B_L1.grid(row=6, column=1)
```

```
# -----Frame_3-----#
```

```
F3 = Frame(self.root, relief=GROOVE, bg="dark blue", border=7)
```

```
F3.place(x=310, y=200, width=300, height=350)
```

```
Grocery = Label(F3, text="Grocery", font=("times new roman", 15, "bold"), bg="dark blue", fg='yellow')
```

```
Grocery.place(x=-1, y=-16)
```

```
Rice = Label(F3, text="Rice", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Rice.grid(row=0, column=0, pady=15, padx=16)
```

```
R1= Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Rice_1)
```

```
R1.grid(row=0, column=1)
```

```
Food_oil = Label(F3, text="Food oil", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Food_oil.grid(row=1, column=0, pady=15, padx=16)
```

```
F_O1 = Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Food_oil_1)
```

```
F_O1.grid(row=1, column=1)
```

```
Daal = Label(F3, text="Daal", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Daal.grid(row=3, column=0, pady=10, padx=16)
```

```
D1 = Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Daal_1)
```

```
D1.grid(row=3, column=1)
```

```
Wheat = Label(F3, text="Wheat", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Wheat.grid(row=4, column=0, pady=15, padx=16)
```

```
W1 = Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE,  
border=4, width=10, textvariable=self.Wheat_1)
```

```
W1.grid(row=4, column=1)
```

```
Sugar = Label(F3, text="Sugar", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Sugar.grid(row=5, column=0, pady=15, padx=16)
```

```
S1 = Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE,  
border=4, width=10, textvariable=self.Sugar_1)
```

```
S1.grid(row=5, column=1)
```

```
Tea = Label(F3, text="Biscuits", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Tea.grid(row=6, column=0, pady=15, padx=16)
```

```
T1 = Entry(F3, font=("times new roman", 15, "bold"), relief=GROOVE,  
border=4, width=10, textvariable=self.Tea_1)
```

```
T1.grid(row=6, column=1)
```

```
# -----Frame_4-----#
```

```
F4 = Frame(self.root, relief=GROOVE, bg="dark blue", border=7)
```

```
F4.place(x=620, y=200, width=300, height=350)
```

```
Cold_Drink= Label(F4, text="Cold Drink", font=("times new roman",  
15, "bold"), bg="dark blue", fg='yellow')
```

```
Cold_Drink.place(x=-1, y=-16)
```

```
Maza= Label(F4, text="Maza", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Maza.grid(row=0, column=0, pady=15, padx=16)
```

```
M1 = Entry(F4, font=("times new roman", 15, "bold"), relief=GROOVE,  
border=4, width=10, textvariable=self.Maza_1)
```

```
M1.grid(row=0, column=1)
```

```
Coca= Label(F4, text="Cock", fg='white', bg="dark blue",  
font=("times new roman", 14, "bold"))
```

```
Coca.grid(row=1, column=0, pady=15, padx=16)
```

```
Co_1 = Entry(F4, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10, textvariable=self.Coca_1)
```

```
Co_1.grid(row=1, column=1)
```

```
Fruti = Label(F4, text="Fruti", fg='white', bg="dark blue", font=("times  
new roman", 14, "bold"))
```

```
Fruti.grid(row=3, column=0, pady=10, padx=16)
```

```
F1 = Entry(F4, font=("times new roman", 15, "bold"), relief=GROOVE,  
border=4, width=10, textvariable=self.Fruti_1)
```

```
F1.grid(row=3, column=1)
```

```
Thumbs_up = Label(F4, text="Thumbs_up", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Thumbs_up.grid(row=4, column=0, pady=15, padx=16)
```

```
T_u1 = Entry(F4, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Thumbs_up_1)
```

```
T_u1.grid(row=4, column=1)
```

```
Sprite = Label(F4, text="Sprite", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Sprite.grid(row=5, column=0, pady=15, padx=16)
```

```
Sp1 = Entry(F4, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Sprite_1)
```

```
Sp1.grid(row=5, column=1)
```

```
Limca = Label(F4, text="Limca", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Limca.grid(row=6, column=0, pady=15, padx=16)
```

```
L1 = Entry(F4, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10, textvariable=self.Limca_1)
```

```
L1.grid(row=6, column=1)
```

```
# -----Frame_5-----#
```

```
F4 = Frame(self.root, relief=GROOVE, bg="white", border=7)
```

```
F4.place(x=930, y=195, width=420, height=360)
```

```
Heading_2= Label(F4, text="Bill Detail", font=("times new roman", 20, "bold"), bg="dark blue",border=9, relief=GROOVE, fg='yellow')
```

```
Heading_2.pack(fill=X)
```

```
Scroll_y=Scrollbar(F4,orient=VERTICAL)
```

```
Scroll_y.pack(side=RIGHT,fill=Y)
```

```
self.Text_1= Text(F4,yscrollcommand=Scroll_y.set)
```

```
self.Text_1.pack()
```

```
Scroll_y.config(command=self.Text_1.yview)
```

```
# -----Frame_6-----#
```

```
F6 = Frame(self.root, relief=GROOVE, bg="dark blue", border=7)
```

```
F6.place(x=0, y=560, relwidth=1, height=200)
```

```
Billing_Menu= Label(F6, text="Billing Menu", font=("times new roman", 15, "bold"), bg="dark blue",fg='yellow')
```

```
Billing_Menu.place(x=-1, y=-16)
```

```
Total_Cosmatic= Label(F6, text="Total Cosmatic Price", fg='white', bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Total_Cosmatic.grid(row=0, column=0,pady=16)
```

```
T_C1 = Entry(F6, font=("times new roman", 15, "bold"), relief=GROOVE, border=4, width=10,textvariable=self.Total_Cosmatic_1)
```

```
T_C1.grid(row=0, column=1)
```

```
Costmatic_Tax=Label(F6, text="Costmatic Tax ", fg='white', bg="dark  
blue", font=("times new roman", 14, "bold"))
```

```
Costmatic_Tax.grid(row=0, column=2)
```

```
C_T1 = Entry(F6, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10,textvariable=self.Cosmatic_Tax_1)
```

```
C_T1.grid(row=0, column=3)
```

```
Total_Grocery = Label(F6, text="Total Grocery Price", fg='white',  
bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Total_Grocery.grid(row=1, column=0,pady=10)
```

```
T_G1 = Entry(F6, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10,textvariable=self.Total_Grocery_1)
```

```
T_G1.grid(row=1, column=1)
```

```
Grocery_Tax = Label(F6, text="Grocery Tax", fg='white', bg="dark  
blue",font=("times new roman", 14, "bold"))
```

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

```
G_T1 = Entry(F6, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10,textvariable=self.Grocery_Tax_1)
```

```
G_T1.grid(row=1, column=3)
```

```
Total_Cold_Drink = Label(F6, text="Total Cold Drink Price", fg='white',  
bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Total_Cold_Drink.grid(row=2, column=0,pady=10)
```

```
T_CD1 = Entry(F6, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10, textvariable=self.Total_colddrink_1)
```

```
T_CD1.grid(row=2, column=1)
```

```
Cold_Drink_Tax = Label(F6, text="Cold Drink Tax", fg='white',  
bg="dark blue", font=("times new roman", 14, "bold"))
```

```
Cold_Drink_Tax.grid(row=2, column=2, padx=16, pady=10)
```

```
C_DT1 = Entry(F6, font=("times new roman", 15, "bold"),  
relief=GROOVE, border=4, width=10, textvariable=self.colddrink_Tax_1)
```

```
C_DT1.grid(row=2, column=3)
```

```
# -----Frame-6_1-----#
```

```
F6_1=Frame(F6, relief=GROOVE, bg="white", border=14)
```

```
F6_1.place(x=630,y=5,width=690, height=130 )
```

```
Total_butt=Button(F6_1, text="Total", height=3, width=18, bg="dark  
blue", font=("times new roman", 10, "bold"), fg='yellow',  
border=6, command=self.Total)
```

```
Total_butt.grid(row=0, column=0, padx=7, pady=20)
```

```
Bill_Generate_butt = Button(F6_1, text="Bill_Generate", height=3,  
width=18, bg="dark blue", font=("times new roman", 10, "bold"), fg='yellow',  
border=6, command=self.Bill_Generate)
```

```
Bill_Generate_butt.grid(row=0, column=1, padx=14, pady=20)
```



```
Clear_butt = Button(F6_1, text="Clear", height=3, width=18,bg="dark
blue",font=("times      new      roman",      10,      "bold"),fg='yellow',
border=6,command=self.Clear)
```

```
Clear_butt.grid(row=0, column=2, padx=14, pady=20)
```

```
Exit_butt = Button(F6_1, text="Exit", height=3, width=18,bg="dark
blue",font=("times      new      roman",      10,      "bold"),fg='yellow',
border=6,command=exit)
```

```
Exit_butt.grid(row=0, column=3, padx=10, pady=20)
```

```
self.Welcome_bill()
```

```
# print(x)
```

```
#-----Function Declaration-----#
```

```
def Clear(self):
```

```
# -----Variable Declaration-----#
```

```
self.CustomerName1.set("")
```

```
self.Contact_No_1.set("")
```

```
self.Bill_No_1.set("")
```

```
# -----Variable Declaration F2-----#
```

```
self.Bath_Soap_1.set("0")
```

```
self.Face_Cream_1.set("0")
```

```
self.Face_Wash_1.set("0")
```

```
self.Hair_Spray_1.set("0")

self.Hair_Gel_1.set("0")

self.Body_Losan_1.set("0")

# -----Variable Declaration F3-----#

self.Rice_1.set("0")

self.Wheat_1.set("0")

self.Sugar_1.set("0")

self.Tea_1.set("0")

self.Daal_1.set("0")

self.Food_oil_1.set("0")

# -----Variable Declaration F4-----#

self.Maza_1.set("0")

self.Coca_1.set("0")

self.Thumbs_up_1.set("0")

self.Fruti_1.set("0")

self.Sprite_1.set("0")

self.Limca_1.set("0")

# -----Variable Declaration F5-----#

self.Total_Cosmatic_1.set("")

self.Total_Grocery_1.set("")
```

```
self.Total_colddrink_1.set("")
```

```
self.Cosmatic_Tax_1.set("")
```

```
self.Grocery_Tax_1.set("")
```

```
self.colddrink_Tax_1.set("")
```

```
self.Welcome_bill()
```

```
def Total(self):
```

```
    self.c_b_s=(self.Bath_Soap_1.get()*20)
```

```
    self.c_f_c=(self.Face_Cream_1.get() * 110)
```

```
    self.c_f_w=(self.Face_Wash_1.get() * 65)
```

```
    self.c_h_s= (self.Hair_Spray_1.get() * 200)
```

```
    self.c_h_g=(self.Hair_Gel_1.get() * 70)
```

```
    self.c_b_l=(self.Body_Losan_1.get() * 40)
```

```
    a=(    self.c_b_s+
```

```
        self.c_f_c+
```

```
        self.c_f_w+
```

```
        self.c_h_s+
```

```
        self.c_h_g+
```

```
        self.c_b_l
```

```
    )
```

```
    self.Total_Cosmatic_1.set(f"Rs {str(a)}")
```

```
self.g_r=( self.Rice_1.get()*40)
self.g_d= (self.Daal_1.get()*35)
self.g_s= (self.Sugar_1.get()*50)
self.g_t=(self.Tea_1.get()*30)
self.g_w= (self.Wheat_1.get()*30)
self.g_f_o=(self.Food_oil_1.get()*30)
b=( self.g_r+
self.g_d+
self.g_s+
self.g_t+
self.g_w+
self.g_f_o)
self.Total_Grocery_1.set(f"Rs {str(b)}")
self.c_m=(self.Maza_1.get()*12)
self.c_t=(self.Thumbs_up_1.get()*12)
self.c_f= (self.Fruti_1.get()*12)
self.c_c=(self.Coca_1.get()*12)
self.c_s=(self.Sprite_1.get()*12)
self.c_l= (self.Limca_1.get()*12)
c=( self.c_m+
```

```

self.c_t+

self.c_f+

self.c_c+

self.c_s+

self.c_l)

self.Total_colddrink_1.set(f"Rs {str(c)}")

#-----Calculate_Cosmatic_tax-----#

self.c_tax=round(a*(10/100))

self.Cosmatic_Tax_1.set(f"Rs {self.c_tax}")

self.g_tax=round(b*(10/100))

self.Grocery_Tax_1.set(f"Rs {self.g_tax}")

self.cd_tax=round(c*(5/100))

self.coldrink_Tax_1.set(f"Rs {self.cd_tax}")

self.Total= float(self.c_tax+self.g_tax+self.cd_tax+c+a+b)

def Welcome_bill(self):

    self.Text_1.delete(1.0,END)

    self.Text_1.insert(END,"Wt*** Goswami Departmental Store***Wn")

    self.Text_1.insert(END, f"Bill No.: {self.Bill_No_1.get()} Wn")

    self.Text_1.insert(END,          f"Customer          Name          :

{self.CustomerName1.get()}Wn")

```

```

self.Text_1.insert(END, f"Contact No. :{self.Contact_No_1.get()}Wn")

self.Text_1.insert(END, f"---*****---
Wn")

self.Text_1.insert(END, f"ProductWtWtWtQuantityWtWtPriceWn")

self.Text_1.insert(END, f"---*****---
Wn")

```

```

def Bill_Generate(self):

    self.Welcome_bill()

    if self.CustomerName1.get()==" and self.Contact_No_1.get()=="":

        tmsg.showerror("Error","Fill Customer Name and Contact No")

    else:

        self.bill_area()

    a=tmsg.askquestion("Billing","Do you want to save this bill?")

    # print (a)

    if a=="yes":

        f=open("files/"+str(self.Bill_No_1.get())+".txt","w")

        f.write(

            self.Text_1.get(1.0,END)

        )

        f.close()

```

```

def bill_area(self):

    self.Welcome_bill()

    # -----Cosmatic-----#

    if self.Bath_Soap_1.get() != 0:

        self.Text_1.insert(END, f"Bath
Soap{self.Bath_Soap_1.get()}{self.c_b_s}\n")

    if self.Face_Cream_1.get() != 0:

        self.Text_1.insert(END, f"Face
Cream{self.Face_Cream_1.get()}{self.c_f_c}\n")

    if self.Face_Wash_1.get() != 0:

        self.Text_1.insert(END, f"Face Wash{self.Face_Wash_1.get()
}{self.c_f_w}\n")

    if self.Hair_Spray_1.get() != 0:

        self.Text_1.insert(END, f"Hair
Spray{self.Hair_Spray_1.get()}{self.c_h_s}\n")

    if self.Hair_Gel_1.get() != 0:

        self.Text_1.insert(END, f"Hair Gel{self.Hair_Gel_1.get()
}{self.c_h_g}\n")

    if self.Body_Losan_1.get() != 0:

        self.Text_1.insert(END, f"Body Losan{self.Body_Losan_1.get()
}{self.c_b_l}\n")

    # -----Grocery-----#

```

```

        if self.Rice_1.get() != 0:

            self.Text_1.insert(END,
f"Rice{self.Rice_1.get()}{self.g_r}\n")

            if self.Wheat_1.get() != 0:

                self.Text_1.insert(END,
f"Wheat{self.Wheat_1.get()}{self.g_w}\n")

                if self.Daal_1.get() != 0:

                    self.Text_1.insert(END, f"Daal {self.Daal_1.get()}{self.g_d}\n")

                    if self.Food_oil_1.get() != 0:

                        self.Text_1.insert(END, f"Food oil{self.Food_oil_1.get()}{self.g_f_o}\n")

                        if self.Sugar_1.get() != 0:

                            self.Text_1.insert(END, f"Sugar{self.Sugar_1.get()}{self.g_s}\n")

                            if self.Tea_1.get() != 0:

                                self.Text_1.insert(END, f"Tea{self.Tea_1.get()}{self.g_t}\n")

                                # -----Coldrink-----#

                                if self.Maza_1.get() != 0:

                                    self.Text_1.insert(END,
f"Maza{self.Maza_1.get()}{self.c_m}\n")

```



```

        if self.Coca_1.get() != 0:

            self.Text_1.insert(END,
f"CocaWtWtWt{self.Coca_1.get()}WtWt{self.c_c}Wn")

        if self.Fruti_1.get() != 0:

            self.Text_1.insert(END,          f"Fruti          WtWtWt{self.Fruti_1.get()}WtWt{self.c_f}Wn")

        if self.Thumbs_up_1.get() != 0:

            self.Text_1.insert(END,                                f"Thumbs
upWtWtWt{self.Thumbs_up_1.get()}WtWt{self.c_t}Wn")

        if self.Sprite_1.get() != 0:

            self.Text_1.insert(END,                                f"SpriteWtWtWt{self.Sprite_1.get()}WtWt{self.c_s}Wn")

        if self.Limca_1.get() != 0:

            self.Text_1.insert(END,                                f"LimcaWtWtWt{self.Limca_1.get()}WtWt{self.c_l}Wn")

        # -----Total Tax-----#

        self.Text_1.insert(END, f"---*****_--
Wn")

        if self.c_tax != 0:

            self.Text_1.insert(END, f"Cosmatic TaxWtWtWtWtWt{self.c_tax}Wn")

        if self.g_tax != 0:

            self.Text_1.insert(END, f"Grocery TaxWtWtWtWtWt{self.g_tax}Wn")

```

```

        if self.cd_tax != 0:

            self.Text_1.insert(END, f"Cold Drink
Tax{self.cd_tax}Wn")

            self.Text_1.insert(END, f"---*****_--
Wn")

            self.Text_1.insert(END, f"Total{self.Total}Wn")

def Search(self):

    present="no"

    files=os.listdir("files/")

    if len(files)>0:

        for i in files:

            # print(i)

            if i.split(".")[0]==self.Bill_No_1.get():

                # print("yes")

                present="yes"

                f1=open(f"files/{i}", "r")

                self.Text_1.delete(1.0,END)

                for j in f1:

                    self.Text_1.insert(END,j)

                f1.close()

            if present=="no":

```

```
tmsg.showerror("Error","File doesn't Existed.")  
  
root=Tk()  
  
B1=Billing(root)  
  
root.mainloop()
```

Output screenshots:

Billing Software

Customer Details

Customer Name
pale
Contact No.
6968695465
Bill No.
1122
Search

Cosmatic

Bath Soap
6

Face Cream
6

Face Wash
6

Hair Spray
6

Hair Gel
6

Body Lusan
6

Grocery

Rice
7

Food oil
7

Daal
7

Wheat
7

Sugar
7

Biscuits
7

Cold Drink

Maza
8

coca cola
8

Fruiti
8

Thumbs up
8

Sprite
8

Limca
8

Bill Detail

*** SUPER MARKET ***
Bill No. 1
Customer Name :
Contact No. :
Product Quantity Price

Billing Menu

Total Cosmatic Price
Costmatic Tax

Total Grocery Price
Grocery Tax

Total Cold Drink Price
Cold Drink Tax

Total
Bill, Cosmatic
Clear
Exit

Billing Menu

Total Cosmatic Price
Rs 1515
Costmatic Tax
Rs 152

Total Grocery Price
Rs 1075
Grocery Tax
Rs 108

Total Cold Drink Price
Rs 288
Cold Drink Tax
Rs 14

Total

Bill Detail

*** SUPER MARKET***

Bill No.: 1122
Customer Name : pale
Contact No. :6968695465

Product	Quantity	Price

Bath Soap	6	120
Face Cream	6	660
Face Wash	6	390
Hair Spray	6	1200
Hair Gel	6	420
Body Losan	6	240
Rice	7	280
Wheat	7	210
Daal	7	245
Food oil	7	210
Sugar	7	350

Bill No.

8888

Search

Bill Detail

*** Goswami Departmental Store***

Bill No.: 8888
Customer Name : manish
Contact No. :8941943770

Product	Quantity	Price

Bath Soap	1	20
Face Cream	1	110
Face Wash	1	65
Hair Spray	1	200
Hair Gel	1	70
Body Losan	1	40
Rice	2	80
Wheat	2	60
Daal	2	70
Food oil	2	60
Sugar	2	100



5555 - Notepad

File Edit Format View Help

*** SUPER MARKET***

Bill No.: 5555

Customer Name : pratap

Contact No. :9785858875

Product	Quantity		Pric
Bath Soap	3		60
Face Cream	3		330
Face Wash	3		195
Hair Spray	3		600
Hair Gel	3		210
Body Losan	3		120
Rice	5	200	
Wheat	5	150	
Daal	5	175	
Food oil	5		150
Sugar	5	250	
Tea	5	150	
Maza	4	48	
Coca	4	48	
Fruti	4	48	
Thumbs up	4		48
Sprite	4	48	
Limca	4	48	

Cosmatic Tax			152
Grocery Tax			108
Cold Drink Tax			14

Total			3152.0

