A PROJECT REPORT

On

Billing Software using Python

Submitted by

Dilshad Ahmad (10800220064)
Anmol Kumar Gupta (10800220041)
Meghal Pandey (10800220009)
Adrita Pathak (10800220080)

Submitted to Asansol Engineering College in partial fulfilment of the requirements for the degree of

Bachelor of Technology

(Information Technology)

Under the Guidance
of

Mr. Biplab Kumar Mondal
(Assistant Professor)

INFORMATION TECHNOLOGY
Asansol Engineering College
Asansol

Affiliated to

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

2024

ASANSOL ENGINEERING COLLEGE
Kanyapur, Vivekananda Sarani, Asansol, Paschim Bardhaman, West Bengal - 713305
Phone: 225-3057, 225-2108 Telefax: (0341) 225-6334
E-mail: principal.aecwb@gmail.com Website: www.aecwb.edu.in

CERTIFICATE

Certified that this project report on**" Billing Software using Python"** is the bonafide work of **"Dilshad Ahmad (10800220064), Anmol Kumar Gupta (10800220041), Meghal Pandey (10800220009), Adrita Pathak (10800220080)"** who carried out the project work under my supervision.

Mr. Biplab Kumar Mondal
Assistant Professor
Information Technology

Dr. Anup Kumar Mukhopadhyay
HoD, Information Technology

Information Technology
Asansol Engineering College
Asansol

ACKNOWLEDGEMENT

It is our great privilege to express my profound and sincere gratitude to our Project Supervisor **Mr. Biplab Kumar Mondal**, Assistant Professor for providing me with very cooperative and precious guidance at every stage of the present project work being carried out under his supervision. His valuable advice and instructions in carrying out the present study have been a very rewarding and pleasurable experience that has greatly benefitted us throughout our work.

We would also like to pay our heartiest thanks and gratitude to **Dr. Anup Kumar Mukhopadhyay**, HOD, and all the faculty members of the Information Technology, Asansol Engineering College for various suggestions being provided in attaining success in our work.

We would like to express our earnest thanks to our colleagues along with all technical staff of the Information Technology, Asansol Engineering College for their valuable assistance being provided during our project work.

Finally, we would like to express our deep sense of gratitude to our parents for their constant motivation and support throughout our work.

……
Dilshad Ahmad

……
Anmol Kumar Gupta

……
Meghal Pandey

……
Adrita Pathak

Date: __ /___/____ **4th Year**
Place: Asansol **Information Technology**

CONTENT

LIST OF FIGURES

CHAPTER 1: Project Synopsis

The Grocery shops is enlarging rapidly and their owners are keen to improve every section of their business. Though much attention is paid to digitalizing the billing management, but not many shop owners realize the importance of applying digital billing software in their system. The customer's experience at your shop includes the billing and payment experiences too. Billing software provides some exclusive features that ease up the payment services. It upgrades the billing process and uplift the customer's experience. It enables customers to pay bills more easily. The software can generate detailed bills that eliminate the need to calculate bills separately when the guests wish to know total tax amount in different products. Apart from billing, the software enables you to organize a number of processes. It makes your system more effective and helps you provide faster and easy services to the customers. So many times, customers leave unhappy due to improper billing. When the crowd is vast in the store, it might take you some time to generate manual bills that may leave your customers unsatisfied. This is where the automated billing system can be used. It generates digital bills automatically and allows customers to make quick payments.

- Firstly, Customer’s name and phone number is taken to generate bill.
- Bill no. will be generated automatically.
- Product quantity is mentioned to calculate the total amount.
- Total button will sum up the bill and display it in the respective product section along with total tax.
- Generate bill button will generate the bill along with company name.

CHAPTER 2: Introduction

Billing System Using Python can be very useful within a business environment. Instead of doing manual work for making up a bill at store, which gets tiring and time consuming, you can generate a bill including tax and service charges in just few clicks. When making up a bill manually at a Restaurant may contain some human errors like adding wrong items into the bill or summing up their total also may end up wrong, it also sometimes results into a Bad Impression towards the Store from a Customer. Ideally, user should be able to generate bill without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, Billing System Using Python.

The Billing System Using Python is very useful to small business or grocery stores. This helps the owner to fasten the process which is bug free and easy to use. It also has a calculator to case the use of the user. This project firstly has the list and then adds up the selected items by customer and sums up the total of all items adds tax and service charges and displays total. To perform any other operation like division, multiplication. etc.

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

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

CHAPTER 3: Project Details

Billing System Project is a simple console application designed to demonstrate the practical use of python and its features as wells as to generate an application which can be used in any grocery store, shops, cafes etc. for billing to the customer. There is always a need of a system that will perform easy billing calculation in a grocery store. This system will reduce the manual operation required to maintain all the bills. And also generates bill receipt with unique bill number.

3.1 System Requirements:

Here we are including the software's and hardware's used for developing the project and implementing the project.

A. Software Requirements
1. Python 3.9 or VS Code
2. Notepad
3. Any OS

B. Hardware Requirements
1. 2 GB RAM or above
2. Intel i3 Processor or above
3. 32 Bit System or above

3.2 Proposed System:

Since many grocery shops make bills for their customers manually with a pen paper. This sometimes results into an error of total or wrong items added or some items missing in bill or extra items added. This may end up by building up a bad impression of customer towards the Store. So, to overcome this problem we've come up with this helpful project named Billing System Using Python. We all love going to cafes or restaurants but when it takes time for them to make a bill or if they Make wrong bill then it's time consuming. So, to avoid all such chaos our project will help in All possible terms.

3.3 Definitions and Theories:

The Bill Management System helps the Store manager to manage the store more effectively and efficiently by computerizing product ordering, billing and inventory control. There is always a need of a system that will perform easy billing calculation in a grocery store. This system will reduce the manual operation required to maintain all the bills. And also generates bill receipt with unique bill number.

3.4 Data Flow Diagram:

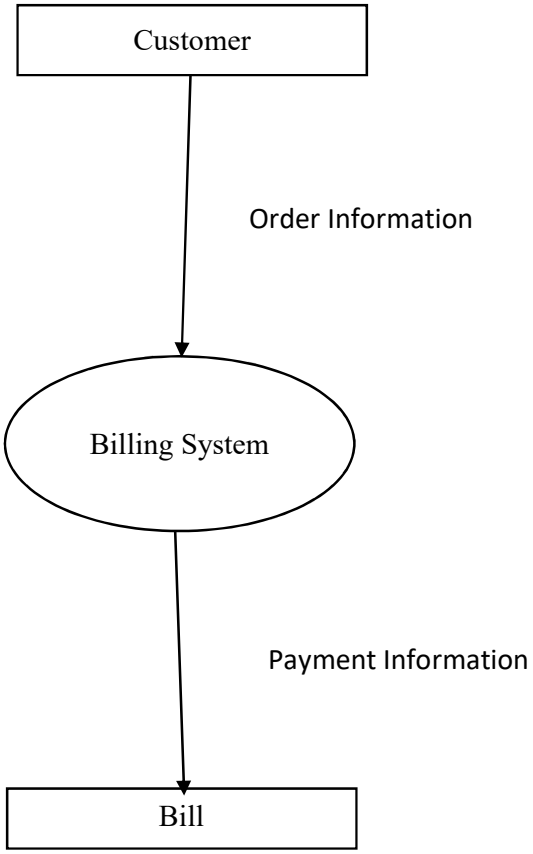

Fig 1: Level Zero DFD of Billing System

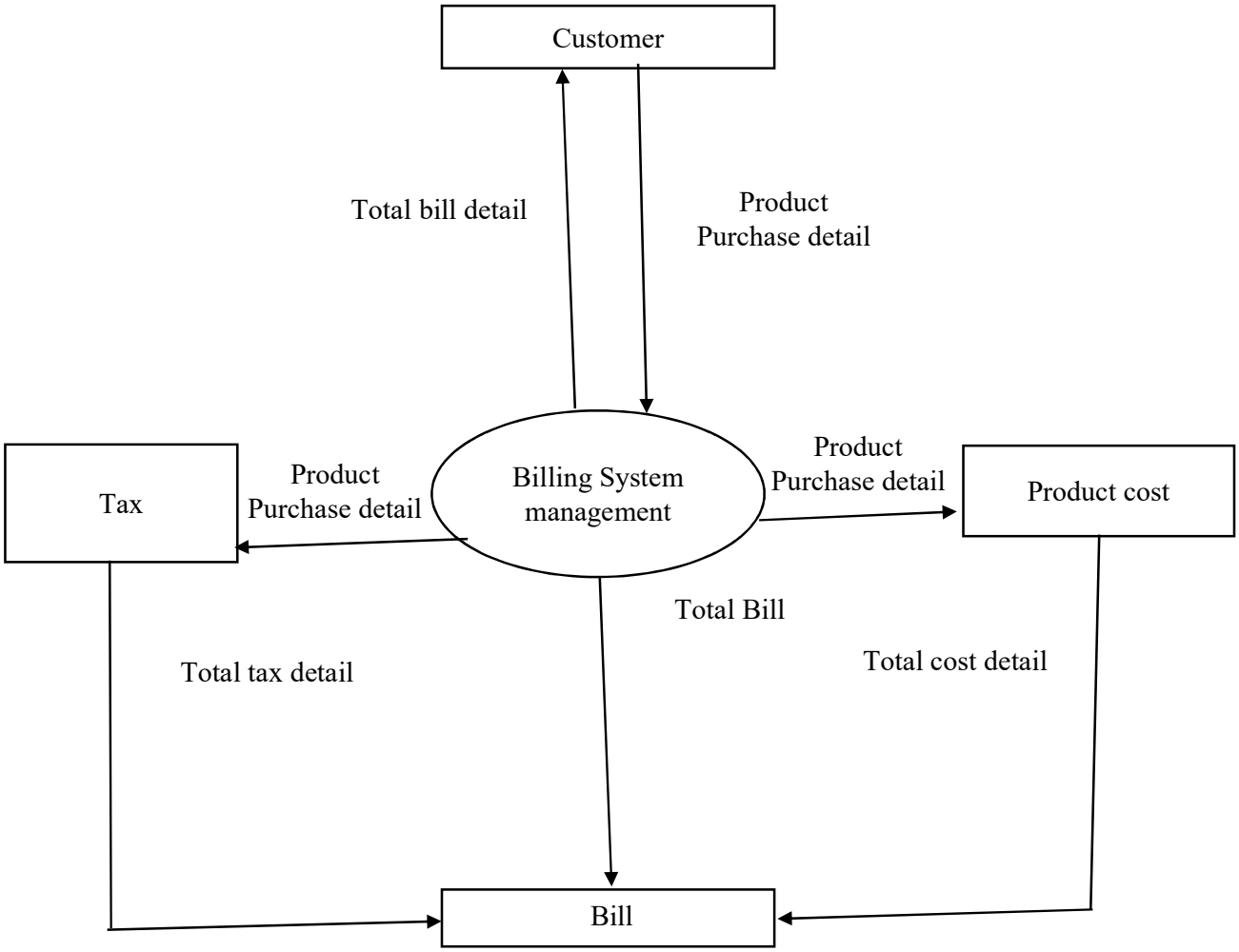

Fig 2: Level One DFD of Billing System

3.5 Outcomes of the project

As we already said, businesses use billing systems to generate automated bill to their customers or partners and receive payments. Consequently, the billing system helps companies to improve performance and reduce errors by automating document preparation and other routine tasks. One of the most important benefits of billing software is that it makes your payment and calculation processing easier. Other than that, it allows you to collect details of regular customer and help you with the tax process. The software will generate bill along with the tax included in particular items.

Automated billing systems improve the overall customer experience by providing timely and accurate invoices. Customers appreciate the convenience of receiving clear, itemized bills and having multiple payment options. The software can also send automated reminders for due or overdue payments, reducing the likelihood of missed payments and improving the business's cash flow.

As businesses grow, their billing needs become more complex. Billing software is scalable and can handle increased transaction volumes without compromising performance. This scalability ensures that the billing process remains smooth and efficient, even as the business expands its customer base and service offerings.

3.6 Work Flow of the Project

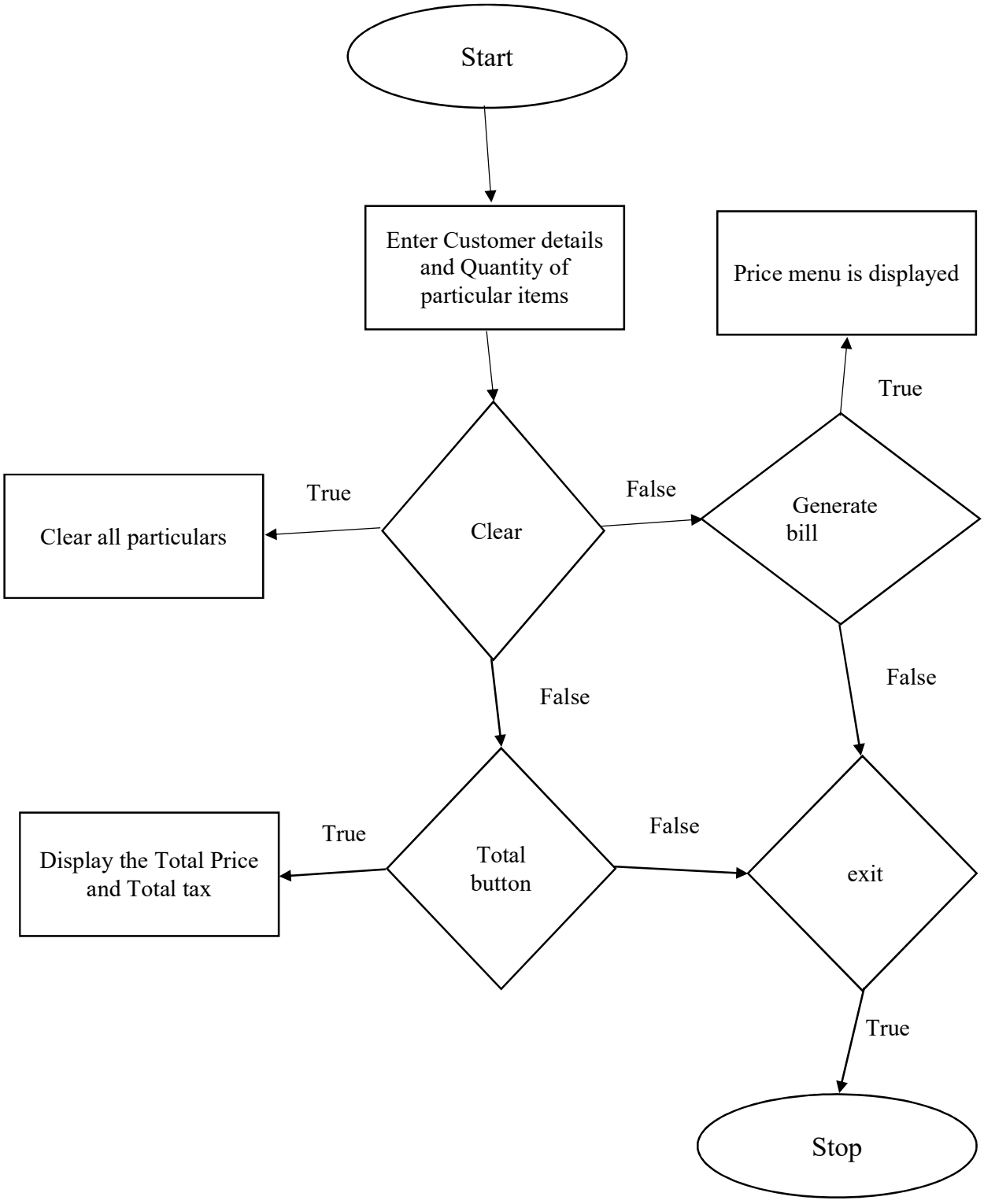

Fig 3: Flow Diagram

3.7 Application Interface

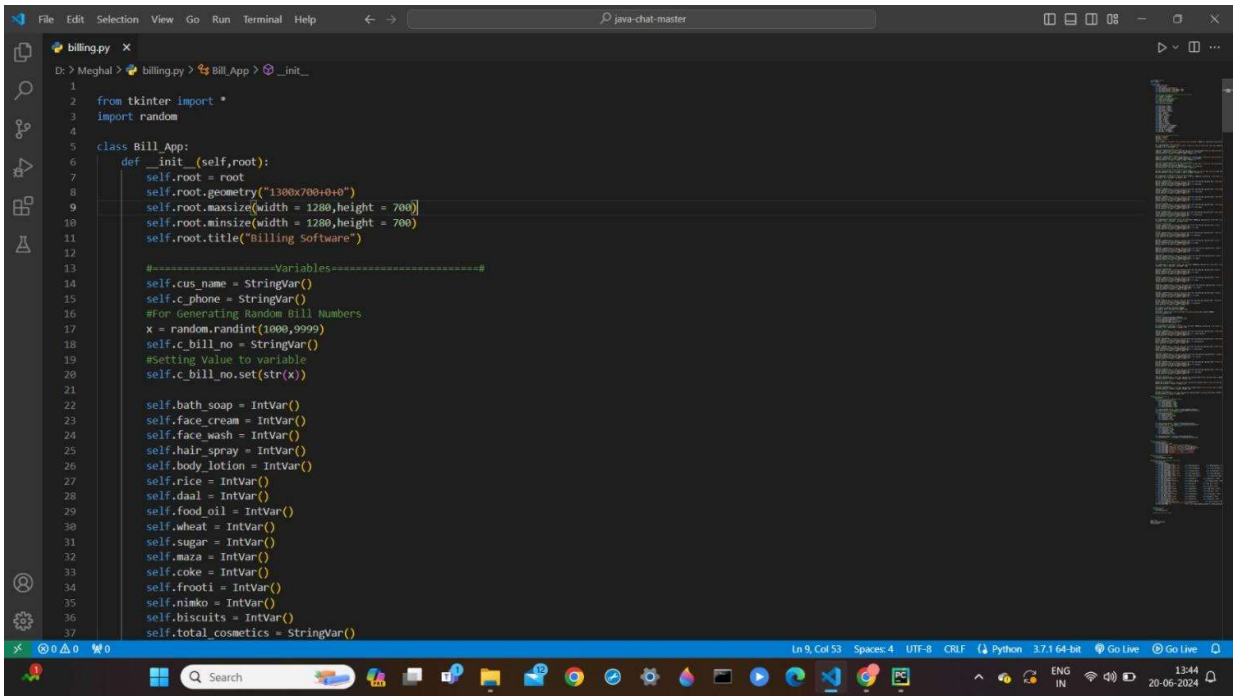

Fig 4: Importing Library

```
from tkinter import *
import random

class Bill_App:
def _init_(self,root):
self.root = root
self.root.geometry("1300x700+0+0")
self.root.maxsize(width = 1280,height = 700)
self.root.minsize(width = 1280,height = 700)
self.root.title("Billing Software")

#====================Variables=======================#
self.cus_name = StringVar()
self.c_phone = StringVar()
#For Generating Random Bill Numbers
x = random.randint(1000,9999)
self.c_bill_no = StringVar()
#Seting Value to variable
self.c_bill_no.set(str(x))

self.bath_soap = IntVar()
self.face_cream = IntVar()
self.face_wash = IntVar()
self.hair_spray = IntVar()
self.body_lotion = IntVar()
```

```
self.rice = IntVar()
self.daal = IntVar()
self.food_oil = IntVar()
self.wheat = IntVar()
self.sugar = IntVar()
self.maza = IntVar()
self.coke = IntVar()
self.frooti = IntVar()
self.nimko = IntVar()
self.biscuits = IntVar()
self.total_cosmetics = StringVar()
self.total_grocery = StringVar()
self.total_other = StringVar()
self.tax_cos = StringVar()
self.tax_groc = StringVar()
self.tax_other = StringVar()


#=====================================
bg_color = "#074463"
fg_color = "white"
lbl_color = 'white'
#Title of App
title = Label(self.root,text = "Billing Software",bd = 12,relief = GROOVE,fg = fg_color,bg = bg_color,font=("times new roman",30,"bold"),pady = 3).pack(fill = X)
```

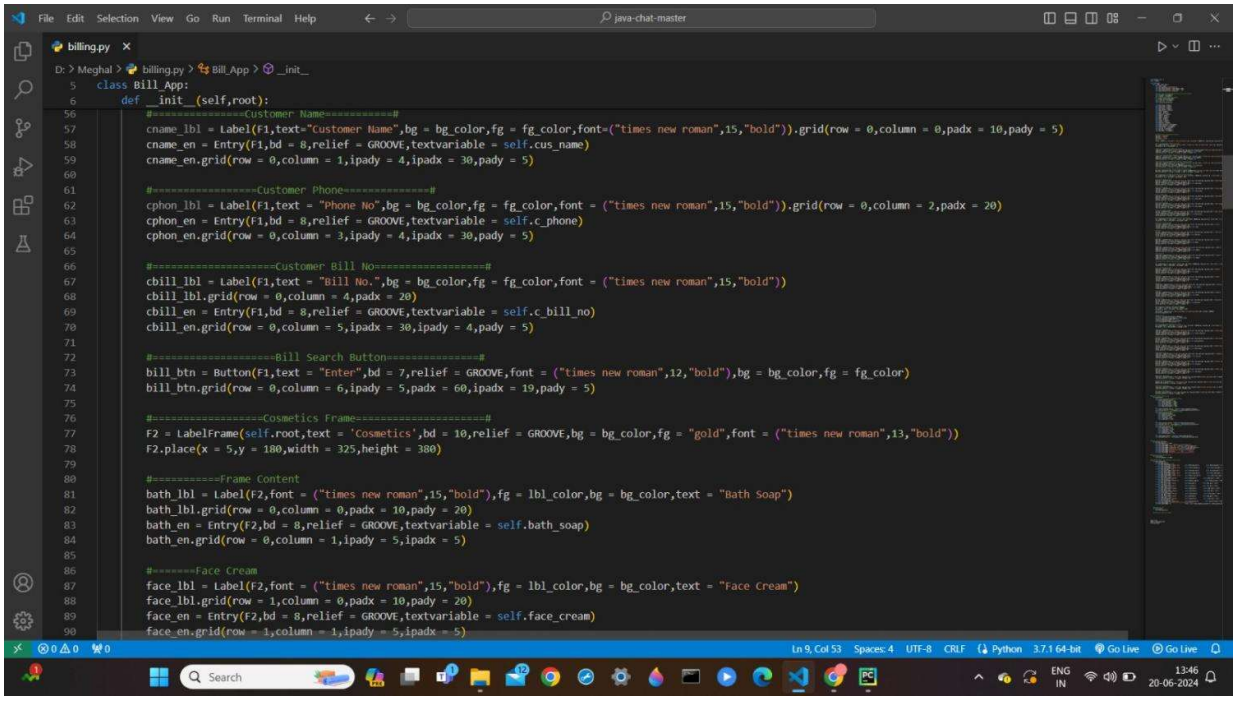

Fig 5: Label

```
#==========Customers Frame==========#
F1 = LabelFrame(text = "Customer Details",font = ("time new roman",12,"bold"),fg = "gold",bg = bg_color,relief = GROOVE,bd = 10)
F1.place(x = 0,y = 80,relwidth = 1)
```

```
#==============Customer Name==========#
cname_lbl = Label(F1,text="Customer Name",bg = bg_color,fg = fg_color,font=("times new roman",15,"bold")).grid(row = 0,column = 0,padx = 10,pady = 5)
cname_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.cus_name)
cname_en.grid(row = 0,column = 1,ipady = 4,ipadx = 30,pady = 5)

#================Customer Phone============#
cphon_lbl = Label(F1,text = "Phone No",bg = bg_color,fg = fg_color,font = ("times new roman",15,"bold")).grid(row = 0,column = 2,padx = 20)
cphon_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.c_phone)
cphon_en.grid(row = 0,column = 3,ipady = 4,ipadx = 30,pady = 5)

#==================Customer Bill No================#
cbill_lbl = Label(F1,text = "Bill No.",bg = bg_color,fg = fg_color,font = ("times new roman",15,"bold"))
cbill_lbl.grid(row = 0,column = 4,padx = 20)
cbill_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.c_bill_no)
cbill_en.grid(row = 0,column = 5,ipadx = 30,ipady = 4,pady = 5)

#==================Bill Search Button==============#
bill_btn = Button(F1,text = "Enter",bd = 7,relief = GROOVE,font = ("times new roman",12,"bold"),bg = bg_color,fg = fg_color)
bill_btn.grid(row = 0,column = 6,ipady = 5,padx = 60,ipadx = 19,pady = 5)

#==================Cosmetics Frame====================#
F2 = LabelFrame(self.root,text = 'Cosmetics',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))
F2.place(x = 5,y = 180,width = 325,height = 380)

#==========Frame Content
bath_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Bath Soap")
bath_lbl.grid(row = 0,column = 0,padx = 10,pady = 20)
bath_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.bath_soap)
bath_en.grid(row = 0,column = 1,ipady = 5,ipadx = 5)

#=======Face Cream
face_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Face Cream")
face_lbl.grid(row = 1,column = 0,padx = 10,pady = 20)
face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.face_cream)
face_en.grid(row = 1,column = 1,ipady = 5,ipadx = 5)

#========Face Wash
wash_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Face Wash")
wash_lbl.grid(row = 2,column = 0,padx = 10,pady = 20)
wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.face_wash)
wash_en.grid(row = 2,column = 1,ipady = 5,ipadx = 5)

#========Hair Spray
hair_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Hair Spray")
hair_lbl.grid(row = 3,column = 0,padx = 10,pady = 20)
hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.hair_spray)
hair_en.grid(row = 3,column = 1,ipady = 5,ipadx = 5)

#==========Body Lotion
lot_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Body Lotion")
lot_lbl.grid(row = 4,column = 0,padx = 10,pady = 20)
```

```
lot_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.body_lotion)
lot_en.grid(row = 4,column = 1,ipady = 5,ipadx = 5)

#==================Grocery Frame=====================#
F2 = LabelFrame(self.root,text = 'Grocery',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times
new roman",13,"bold"))
F2.place(x = 330,y = 180,width = 325,height = 380)

#==========Frame Content
rice_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Rice")
rice_lbl.grid(row = 0,column = 0,padx = 10,pady = 20)
rice_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.rice)
rice_en.grid(row = 0,column = 1,ipady = 5,ipadx = 5)

#=======
oil_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Food Oil")
oil_lbl.grid(row = 1,column = 0,padx = 10,pady = 20)
oil_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.food_oil)
oil_en.grid(row = 1,column = 1,ipady = 5,ipadx = 5)

#=======
daal_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Daal")
daal_lbl.grid(row = 2,column = 0,padx = 10,pady = 20)
daal_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.daal)
daal_en.grid(row = 2,column = 1,ipady = 5,ipadx = 5)

#========
wheat_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Wheat")
wheat_lbl.grid(row = 3,column = 0,padx = 10,pady = 20)
wheat_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.wheat)
wheat_en.grid(row = 3,column = 1,ipady = 5,ipadx = 5)

#===========
sugar_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Sugar")
sugar_lbl.grid(row = 4,column = 0,padx = 10,pady = 20)
sugar_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.sugar)
sugar_en.grid(row = 4,column = 1,ipady = 5,ipadx = 5)

#==================Other Stuff====================#

F2 = LabelFrame(self.root,text = 'Others',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times
new roman",13,"bold"))
F2.place(x = 655,y = 180,width = 325,height = 380)

#==========Frame Content
maza_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Maaza")
maza_lbl.grid(row = 0,column = 0,padx = 10,pady = 20)
maza_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.maza)
maza_en.grid(row = 0,column = 1,ipady = 5,ipadx = 5)

#=======
cock_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Coke")
cock_lbl.grid(row = 1,column = 0,padx = 10,pady = 20)
cock_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.coke)
cock_en.grid(row = 1,column = 1,ipady = 5,ipadx = 5)
```

```
#=======
frooti_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Frooti")
frooti_lbl.grid(row = 2,column = 0,padx = 10,pady = 20)
frooti_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.frooti)
frooti_en.grid(row = 2,column = 1,ipady = 5,ipadx = 5)

#========
cold_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Nimkos")
cold_lbl.grid(row = 3,column = 0,padx = 10,pady = 20)
cold_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.nimko)
cold_en.grid(row = 3,column = 1,ipady = 5,ipadx = 5)

#============
bis_lbl = Label(F2,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Biscuits")
bis_lbl.grid(row = 4,column = 0,padx = 10,pady = 20)
bis_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.biscuits)
bis_en.grid(row = 4,column = 1,ipady = 5,ipadx = 5)

#==================Bill Aera================#
F3 = Label(self.root,bd = 10,relief = GROOVE)
F3.place(x = 960,y = 180,width = 325,height = 380)
#===========
bill_title = Label(F3,text = "Bill Area",font = ("Lucida",13,"bold"),bd= 7,relief = GROOVE)
bill_title.pack(fill = X)

#============
scroll_y = Scrollbar(F3,orient = VERTICAL)
self.txt = Text(F3,yscrollcommand = scroll_y.set)
scroll_y.pack(side = RIGHT,fill = Y)
scroll_y.config(command = self.txt.yview)
self.txt.pack(fill = BOTH,expand = 1)

#==========Buttons Frame============#
F4 = LabelFrame(self.root,text = 'Bill Menu',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times
new roman",13,"bold"))
F4.place(x = 0,y = 560,relwidth = 1,height = 145)

#==================
cosm_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Total
Cosmetics")
cosm_lbl.grid(row = 0,column = 0,padx = 10,pady = 0)
cosm_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.total_cosmetics)
cosm_en.grid(row = 0,column = 1,ipady = 2,ipadx = 5)

#==================
gro_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Total Grocery")
gro_lbl.grid(row = 1,column = 0,padx = 10,pady = 5)
gro_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.total_grocery)
gro_en.grid(row = 1,column = 1,ipady = 2,ipadx = 5)

#===============
oth_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Others Total")
oth_lbl.grid(row = 2,column = 0,padx = 10,pady = 5)
oth_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.total_other)
oth_en.grid(row = 2,column = 1,ipady = 2,ipadx = 5)
```

```
#================
cosmt_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Cosmetics Tax")
cosmt_lbl.grid(row = 0,column = 2,padx = 30,pady = 0)
cosmt_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.tax_cos)
cosmt_en.grid(row = 0,column = 3,ipady = 2,ipadx = 5)

#================
grot_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Grocery    Tax")
grot_lbl.grid(row = 1,column = 2,padx = 30,pady = 5)
grot_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.tax_groc)
grot_en.grid(row = 1,column = 3,ipady = 2,ipadx = 5)

#================
otht_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Others Tax")
otht_lbl.grid(row = 2,column = 2,padx = 10,pady = 5)
otht_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.tax_other)
otht_en.grid(row = 2,column = 3,ipady = 2,ipadx = 5)

#==================
total_btn = Button(F4,text = "Total",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.total)
total_btn.grid(row = 1,column = 4,ipadx = 20,padx = 30)

#=======================
genbill_btn = Button(F4,text = "Generate Bill",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.bill_area)
genbill_btn.grid(row = 1,column = 5,ipadx = 20)

#==================
clear_btn = Button(F4,text = "Clear",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.clear)
clear_btn.grid(row = 1,column = 6,ipadx = 20,padx = 30)

#====================
exit_btn = Button(F4,text = "Exit",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.exit)
exit_btn.grid(row = 1,column = 7,ipadx = 20)

#Function to get total prices
def total(self):
#================Total Cosmetics Prices
self.total_cosmetics_prices = (
(self.bath_soap.get() * 40)+
(self.face_cream.get() * 140)+
(self.face_wash.get() * 240)+
(self.hair_spray.get() * 340)+
(self.body_lotion.get() * 260)
)
self.total_cosmetics.set("Rs. "+str(self.total_cosmetics_prices))
self.tax_cos.set("Rs. "+str(round(self.total_cosmetics_prices*0.05)))
#==================Total Grocery Prices
self.total_grocery_prices = (
(self.wheat.get()*100)+
(self.food_oil.get() * 180)+
(self.daal.get() * 80)+
```

```
(self.rice.get() *80)+
(self.sugar.get() * 170)

)
self.total_grocery.set("Rs. "+str(self.total_grocery_prices))
self.tax_groc.set("Rs. "+str(round(self.total_grocery_prices*0.05)))
#=====================Total Other Prices
self.total_other_prices = (
(self.maza.get() * 20)+
(self.frooti.get() * 50)+
(self.coke.get() * 60)+
(self.nimko.get() * 20)+
(self.biscuits.get() * 20)
)
self.total_other.set("Rs. "+str(self.total_other_prices))
self.tax_other.set("Rs. "+str(round(self.total_other_prices*0.05)))


#Function For Text Area
def welcome_soft(self):
self.txt.delete('1.0',END)
self.txt.insert(END,"       Welcome To Iconic Retail\n")
self.txt.insert(END,f"\nBill No. : {str(self.c_bill_no.get())}")
self.txt.insert(END,f"\nCustomer Name : {str(self.cus_name.get())}")
self.txt.insert(END,f"\nPhone No. : {str(self.c_phone.get())}")
self.txt.insert(END,"\n=======================================")
self.txt.insert(END,"\nProduct          Qty         Price")
self.txt.insert(END,"\n=======================================")

#Function to clear the bill area
def clear(self):
self.txt.delete('1.0',END)
```

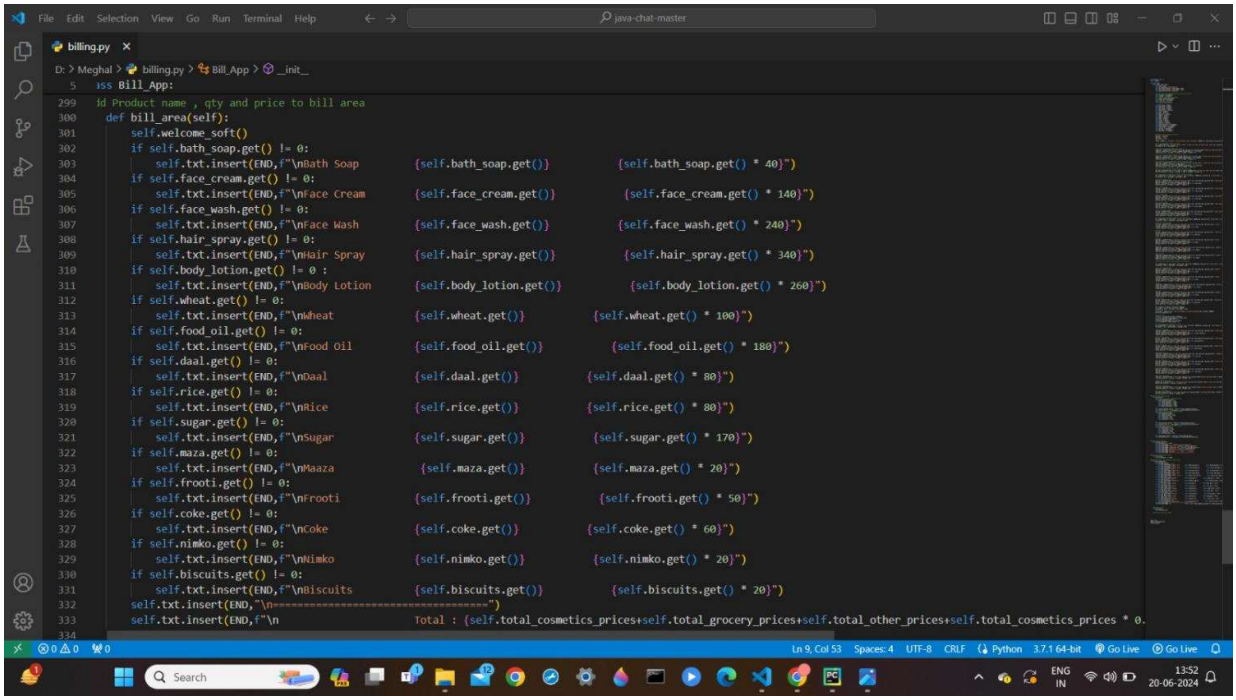

Fig 6: Bill Area

```
#Add Product name , qty and price to bill area
def bill_area(self):
self.welcome_soft()
if self.bath_soap.get() != 0:
self.txt.insert(END,f"\nBath Soap        {self.bath_soap.get()}          {self.bath_soap.get() * 40}")
if self.face_cream.get() != 0:
self.txt.insert(END,f"\nFace Cream       {self.face_cream.get()}         {self.face_cream.get() * 140}")
if self.face_wash.get() != 0:
self.txt.insert(END,f"\nFace Wash        {self.face_wash.get()}          {self.face_wash.get() * 240}")
if self.hair_spray.get() != 0:
self.txt.insert(END,f"\nHair Spray       {self.hair_spray.get()}         {self.hair_spray.get() * 340}")
if self.body_lotion.get() != 0 :
self.txt.insert(END,f"\nBody Lotion      {self.body_lotion.get()}        {self.body_lotion.get() * 260}")
if self.wheat.get() != 0:
self.txt.insert(END,f"\nWheat           {self.wheat.get()}          {self.wheat.get() * 100}")
if self.food_oil.get() != 0:
self.txt.insert(END,f"\nFood Oil         {self.food_oil.get()}          {self.food_oil.get() * 180}")
if self.daal.get() != 0:
self.txt.insert(END,f"\nDaal            {self.daal.get()}          {self.daal.get() * 80}")
if self.rice.get() != 0:
self.txt.insert(END,f"\nRice            {self.rice.get()}          {self.rice.get() * 80}")
if self.sugar.get() != 0:
self.txt.insert(END,f"\nSugar           {self.sugar.get()}          {self.sugar.get() * 170}")
if self.maza.get() != 0:
self.txt.insert(END,f"\nMaaza           {self.maza.get()}          {self.maza.get() * 20}")
if self.frooti.get() != 0:
self.txt.insert(END,f"\nFrooti          {self.frooti.get()}          {self.frooti.get() * 50}")
if self.coke.get() != 0:
self.txt.insert(END,f"\nCoke            {self.coke.get()}          {self.coke.get() * 60}")
if self.nimko.get() != 0:
self.txt.insert(END,f"\nNimko           {self.nimko.get()}          {self.nimko.get() * 20}")
if self.biscuits.get() != 0:
self.txt.insert(END,f"\nBiscuits         {self.biscuits.get()}          {self.biscuits.get() * 20}")
self.txt.insert(END,"\n==================================")
self.txt.insert(END,f"\n                                                                                    Total     :
{self.total_cosmetics_prices+self.total_grocery_prices+self.total_other_prices+self.total_cosmetics_prices     *
0.05+self.total_grocery_prices * 0.05+self.total_other_prices * 0.05}")


#Function to exit
def exit(self):
self.root.destroy()

#Function To Clear All Fields




root = Tk()
object = Bill_App(root)
root.mainloop()
```

Outputs:

Step 1: Enter customer details, click Enter. Then enter the quantity of the products. Click Total to display bill menu.

Fig 7: Customer Detail Input

Step 2: Click Generate bill to generate the bill with bill no. and Customer detail.

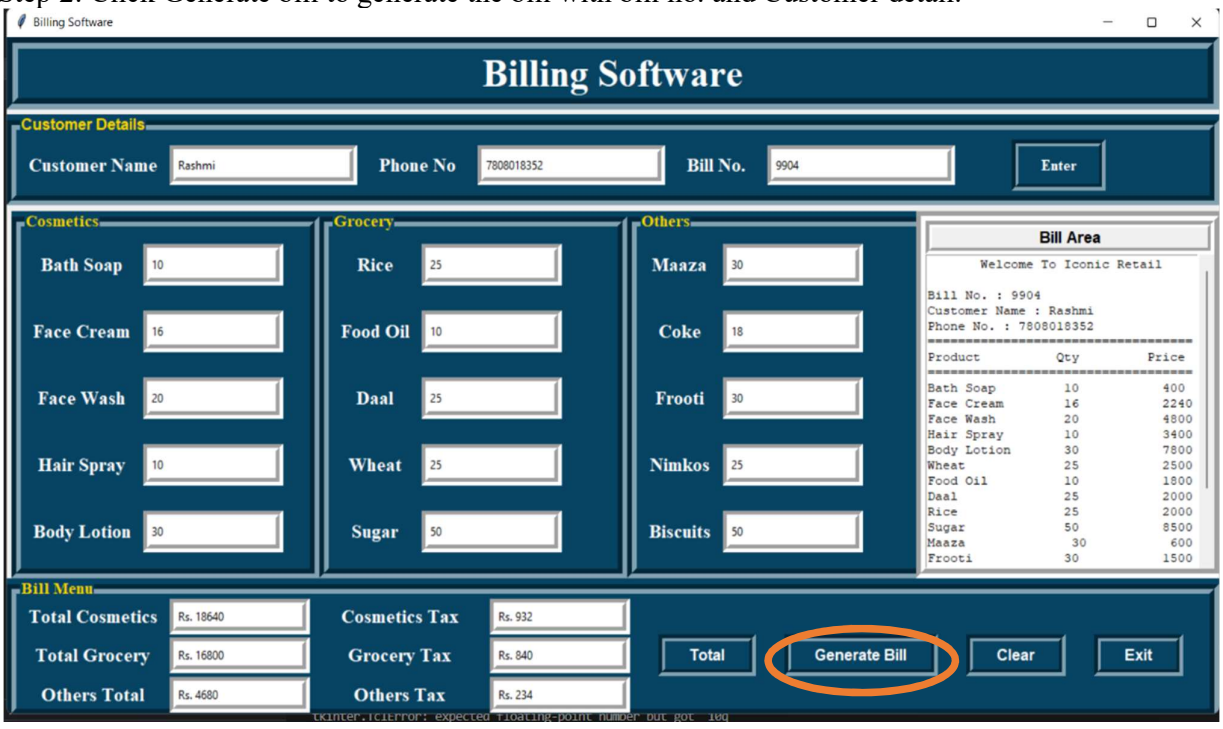

Fig 8 : Generate Bill

Step 3: Scroll down the bill.

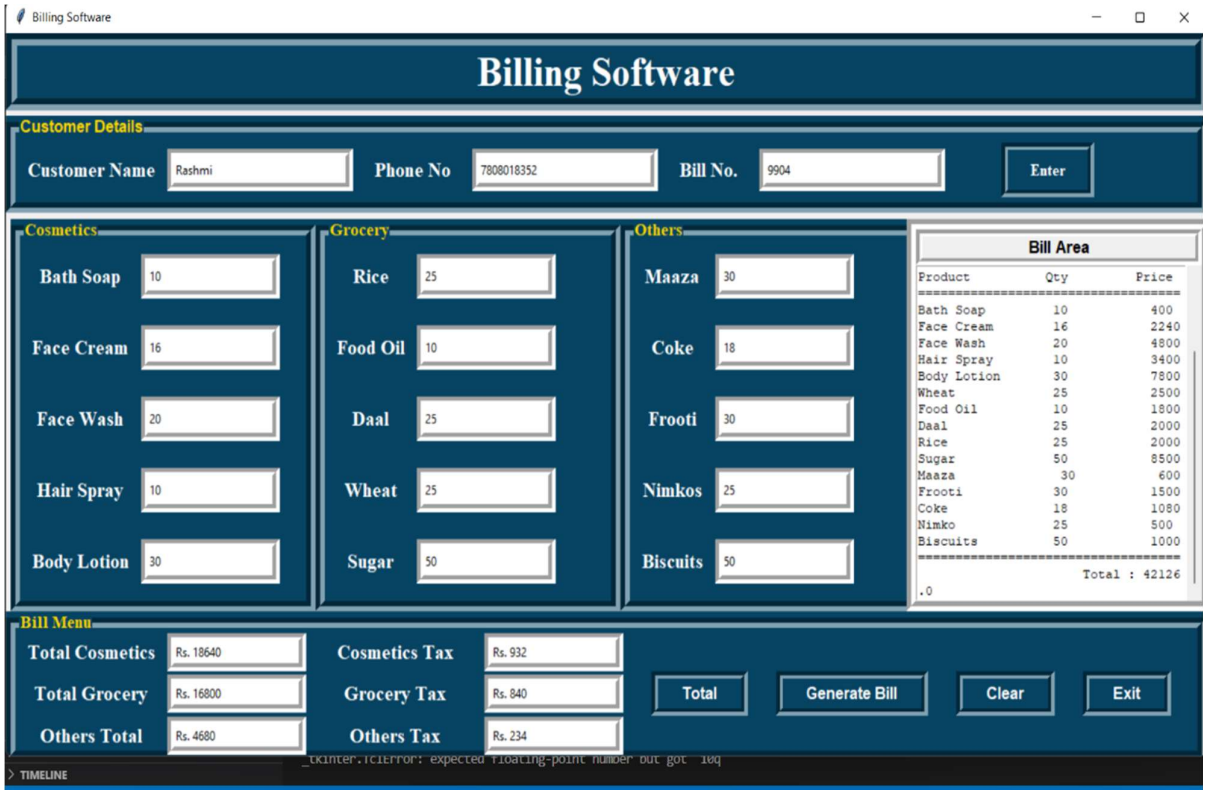

Fig 9 : Final and Print Bill

CHAPTER 4 : Conclusion and Future Scope

Future Scope

In future, this application can be updated with some more items. Many other latest features will be added. Project will surely be enhanced with respect to looks and appearance and also as per user requirements. Many more functionalities will be added. Some enhancement can also be done with calculator. For now, this application generates the bill but with respect to future application it will be enhanced that it will also print a bill. It can also be used on a large scale. Many more modification can do with menu or prices or tax as well. It will be easy to use and bug free to all future or upcoming users. This can also be enhanced in future as per customer requirements. Many more features can be added. This will surely help users instead of making a bill manually.

Conclusion

The urge for the digital bill management systems is increasing day by day. Billing System Using Python is a perfect solution for this. Through this the ease of access and flexibility of the day to day works in the store is made simpler. The features such as bill number, CGST and SGST make this software user friendly. Both the management side and worker site can manage the data easily using such a system. It is very good and reliable system which can be in corporate to the chain of stores so can easily maintained and addressed.

CHAPTER 5: Reference

- **https://www.learnpython.org/**
- **https://docs.python.org/3/library/**
- **https://www.javatpoint.com/**
- **https://www.w3schools.com/**
- **https://www.geeksforgeeks.org**
- **https://hackr.io**
- **https://www.youtube.com**