ARAVALi

COLLEGE OF ENGINEERING & MANAGEMENT

creating achievers

PROJECT REPORT

ON

<u>“Billing system”</u>

Submitted To:
Ms.Antim

Submitted By
Anuj Kumar Jha (22CSE19)
Santosh Singh (22CSE97)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ARAVALI COLLEGE OF ENGINEERING ANDMANAGEMENT
FARIDABAD – 121002

ACKNOWLEDGEMENT

This project would not have taken shape, without the guidance provided by **Ms. Antim**, my Trainer who helped in the modules of our project and resolved all the technical as well as other problems related to the project and, for always providing us with a helping hand whenever we faced any bottlenecks, in spite of being quite busy with their hectic schedules.

We would also like to thank our project supervisor Ms.Antim who gave me the opportunity and provided us all the academic and conceptual support for our project.

Above all we wish to express our heartfelt gratitude to **Ms. Sakshi**, H.O.D, CSE DEPARTMENT whose support has greatly boosted our self-confidence and will go a long way on helping us to reach further milestones and greater heights

ABSTRACT

The Billing System is an essential tool for businesses to streamline their billing processes efficiently. This project aims to develop a billing management system using Python programming language. It provides a user-friendly interface for generating invoices, managing customer information, tracking payments, and generating reports. In which we use Python programming language. VS code is used as a code editor in this project.

TABLE OF CONTENTS

Introduction

Billing management is a crucial aspect of any business, regardless of its size or industry. Traditional manual billing processes are often time-consuming, error-prone, and inefficient. Automation of billing processes using software systems can significantly improve accuracy, reduce human error, and enhance productivity. This project introduces a Billing Management System developed using Python to address these challenges.

1.1 PROBLEM STATEMENT:

Traditional billing processes involve manual data entry, which is prone to errors and consumes a lot of time. Additionally, managing customer information, tracking payments, and generating reports manually can be tedious and inefficient. There is a need for an automated billing management system to streamline these processes and improve overall efficiency.

1.2 OBJECTIVE OF PROPOSED SYSTEM:

The main objectives of the proposed Billing System are as follows:

- Automate the billing process to reduce manual effort and minimize errors.
- Provide a user-friendly interface for generating invoices, managing customer information, and tracking payments.
- Enable easy customization of invoices to suit the specific needs of different businesses.
- Implement security features to protect sensitive customer data.
- Generate various reports such as sales reports, payment reports, and overdue invoices for better decision-making.

1.3 SCOPE OF THE PROPOSED PROJECT

Our project's scope revolves around creating a user-friendly billing system using Python, with a focus on simplicity and practicality. Here's what we aim to cover:

- **Customer Management**: Add, edit, and delete customer information.
- **Invoice Generation**: Generate customized invoices with relevant details.
- **Payment Tracking:** Track payments received and pending from customers.
- **Report Generation:** Generate various reports for analysis and decision-making.
- **Security:** Implement authentication and authorization mechanisms to secure the

system.

1.4 FEASIBILITY STUDY:

Before initiating the project, a feasibility study was conducted to assess the technical, operational, and economic feasibility of the proposed Billing Management System.

- **Technical Feasibility:** Python is a widely used programming language with extensive libraries and frameworks suitable for developing the proposed system. The required technical expertise is readily available, making the project technically feasible.

- **Operational Feasibility:** The system will streamline billing processes, improve accuracy, and enhance productivity. User training will be provided to ensure smooth adoption of the system, making it operationally feasible.

- **Economic Feasibility:** The cost of developing and maintaining the Billing Management System is within budget constraints. The benefits of improved efficiency and reduced errors outweigh the costs, making the project economically feasible.

SYSTEM ANALYSIS

2.1 USER INTERFACE

Back-end software: Python programming Language

2.2 HARDWARE REQUIREMENTS

Following are the hardware used:

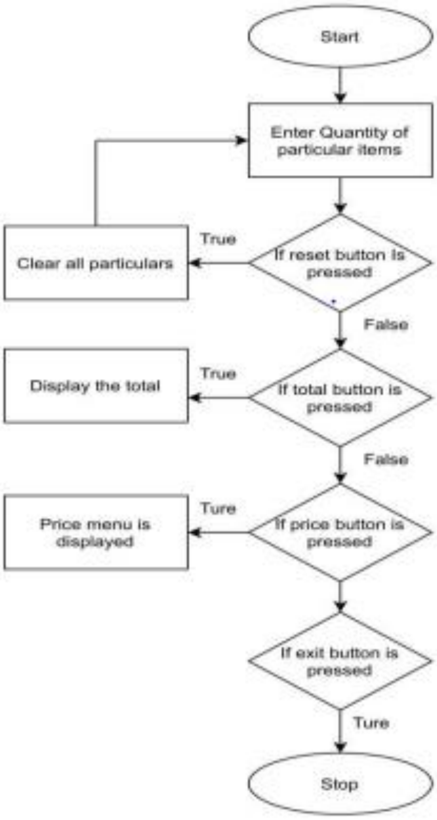
1. Server Requirements (if deployed on a local server or private network):
   - Processor: Minimum Intel i5 or equivalent
   - RAM: 8GB
   - Storage: 500GB SSD or higher for faster data retrieval
   - Network: High-speed internet connection for remote access and online features
2. Client/End-User Machine:
   - Processor: Intel i3 or higher
   - RAM: 4GB
   - Storage: 250GB HDD or SSD for client-side applications
   - Display: Any display supporting a resolution of 1366x768 or higher

2.3 SOFTWARE REQUIREMENTS

Following are the software used:

- Operating system – Windows
- Simulation Tools – Visual Studio Code
- Python – Version 3.9.6

2.4Data Flow Diagram (DFD):

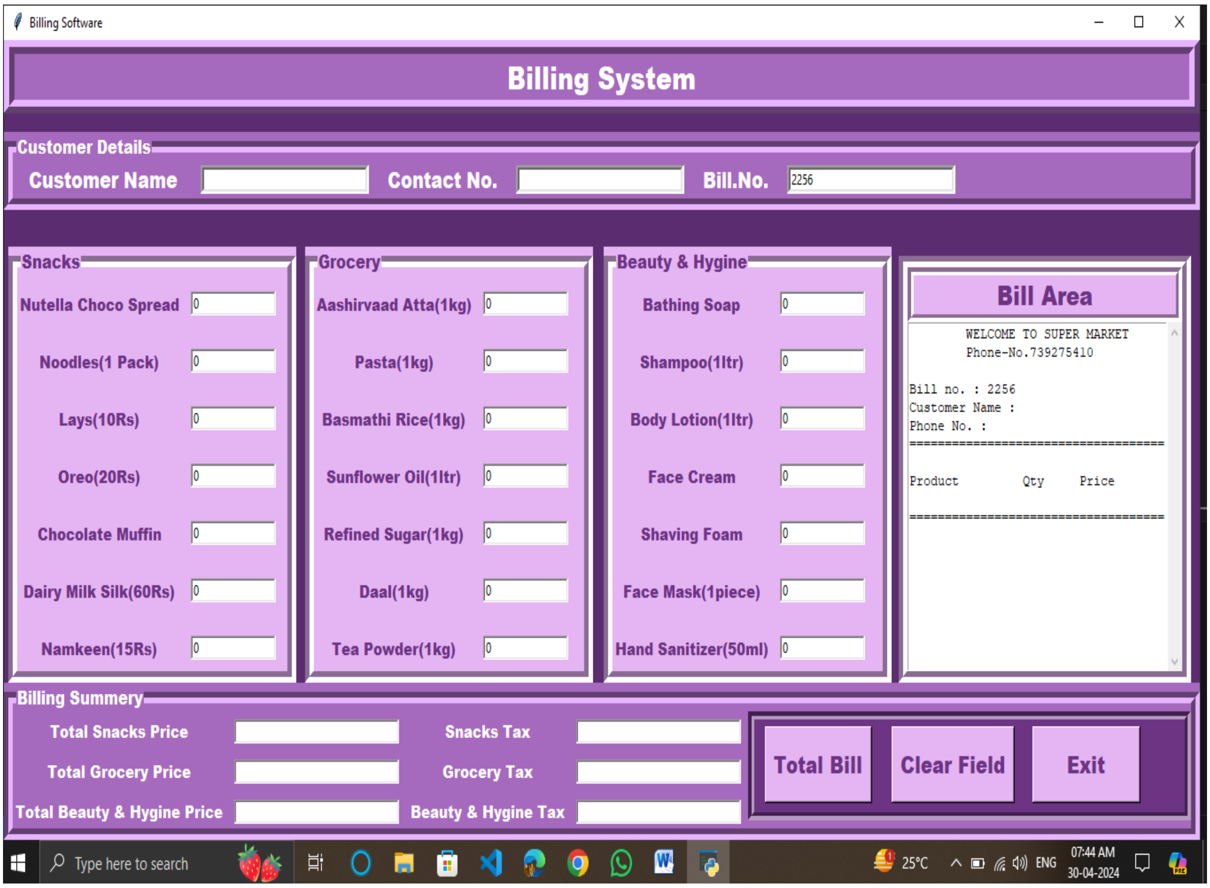

Start
Enter Quantity of particular items
True
Clear all particulars
If reset button is pressed
False
Display the total
True
If total button is pressed
False
Price menu is displayed
Ture
If price button is pressed
If exit button is pressed
Ture
Stop

SYSTEM DESIGN

3.1System Module:

- Before entering the data:

(Fig:1)



Billing Software

Billing System

Customer Details

Customer Name

Contact No.

Bill.No. 2256

Snacks		Grocery		Beauty & Hygine	
Nutella Choco Spread	0	Aashirvaad Atta(1kg)	0	Bathing Soap	0
Noodles(1 Pack)	0	Pasta(1kg)	0	Shampoo(1ltr)	0
Lays(10Rs)	0	Basmathi Rice(1kg)	0	Body Lotion(1ltr)	0
Oreo(20Rs)	0	Sunflower Oil(1ltr)	0	Face Cream	0
Chocolate Muffin	0	Refined Sugar(1kg)	0	Shaving Foam	0
Dairy Milk Silk(60Rs)	0	Daal(1kg)	0	Face Mask(1piece)	0
Namkeen(15Rs)	0	Tea Powder(1kg)	0	Hand Sanitizer(50ml)	0

Bill Area

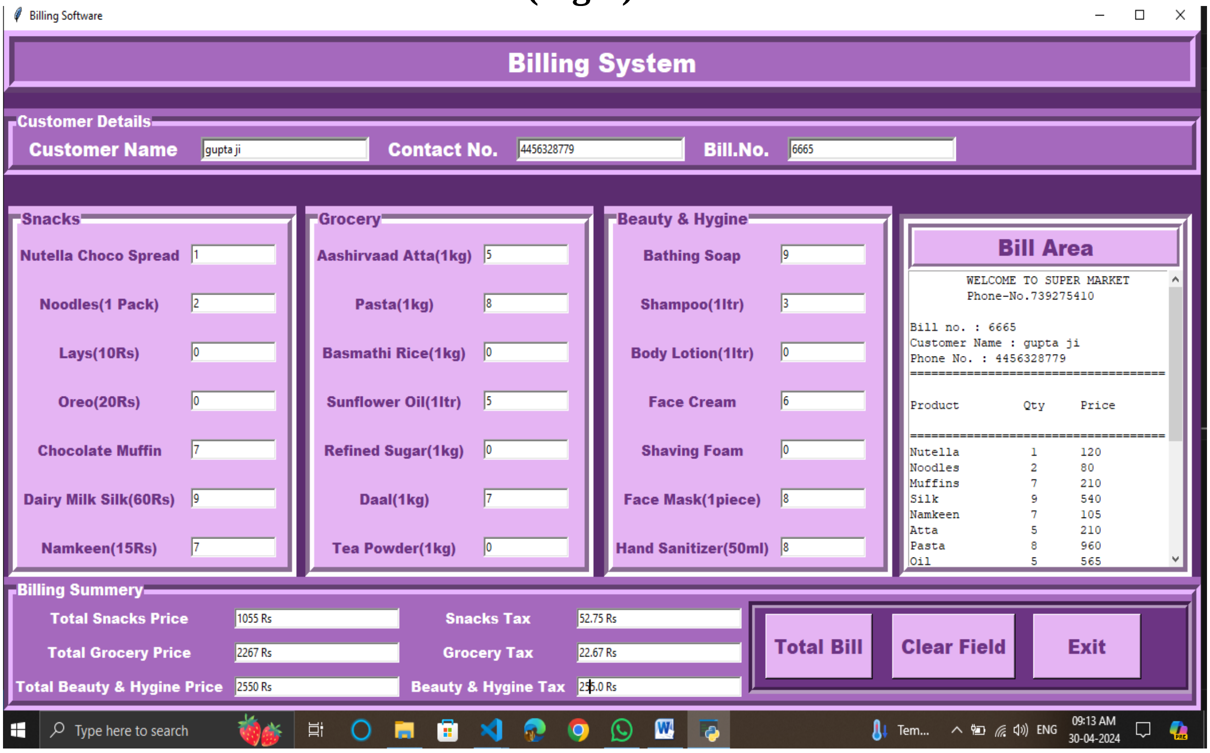
```
        WELCOME TO SUPER MARKET
        Phone-No.739275410

Bill no. : 2256
Customer Name :
Phone No. :
=====================================
Product          Qty     Price

=====================================
```

Billing Summery

Total Snacks Price		Snacks Tax	
Total Grocery Price		Grocery Tax	
Total Beauty & Hygine Price		Beauty & Hygine Tax	

- After entering the data result is :

(Fig:2)



Billing System

Customer Details

Customer Name	gupta ji	Contact No.	4456328779	Bill.No.	6665

Snacks		Grocery		Beauty & Hygine	
Nutella Choco Spread	1	Aashirvaad Atta(1kg)	5	Bathing Soap	9
Noodles(1 Pack)	2	Pasta(1kg)	8	Shampoo(1ltr)	3
Lays(10Rs)	0	Basmathi Rice(1kg)	0	Body Lotion(1ltr)	0
Oreo(20Rs)	0	Sunflower Oil(1ltr)	5	Face Cream	6
Chocolate Muffin	7	Refined Sugar(1kg)	0	Shaving Foam	0
Dairy Milk Silk(60Rs)	9	Daal(1kg)	7	Face Mask(1piece)	8
Namkeen(15Rs)	7	Tea Powder(1kg)	0	Hand Sanitizer(50ml)	8

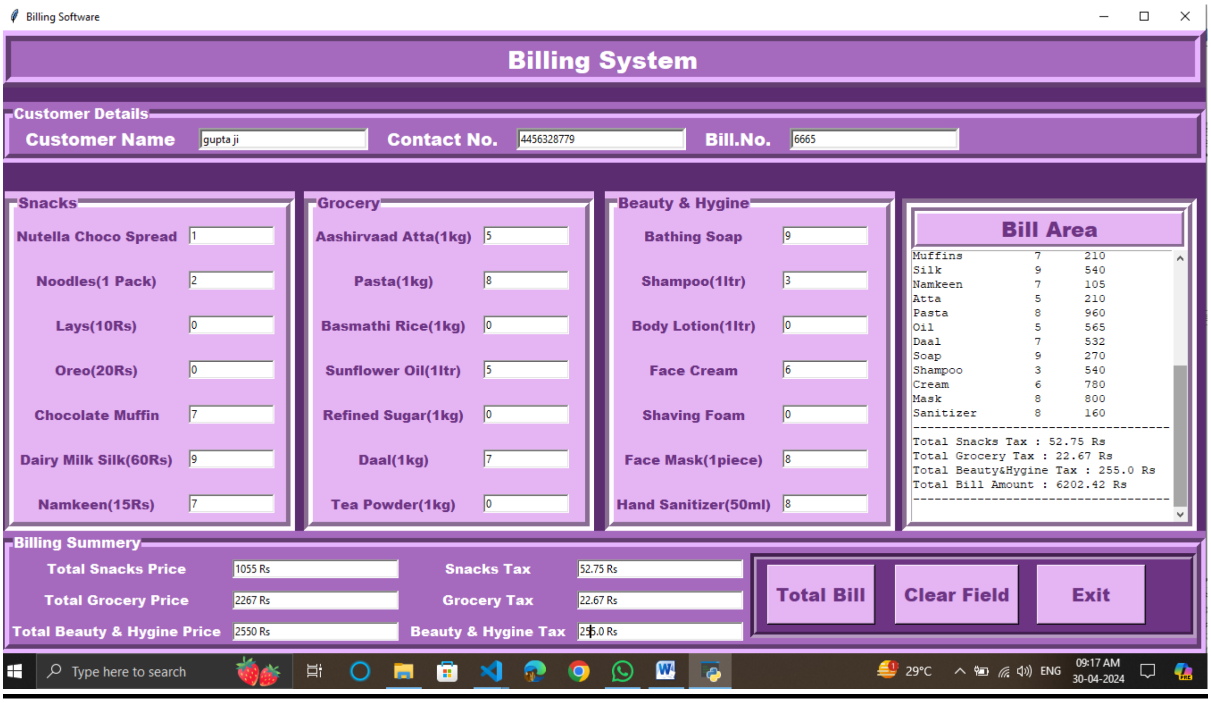
Bill Area

```
        WELCOME TO SUPER MARKET
        Phone-No.739275410

Bill no. : 6665
Customer Name : gupta ji
Phone No. : 4456328779
====================================


Product         Qty     Price


====================================
Nutella          1      120
Noodles          2      80
Muffins          7      210
Silk             9      540
Namkeen          7      105
Atta             5      210
Pasta            8      960
Oil              5      565
```

Billing Summery

Total Snacks Price	1055 Rs	Snacks Tax	52.75 Rs
Total Grocery Price	2267 Rs	Grocery Tax	22.67 Rs
Total Beauty & Hygine Price	2550 Rs	Beauty & Hygine Tax	255.0 Rs

(Fig:3)



Billing System

Customer Details

Customer Name	gupta ji	Contact No.	4456328779	Bill.No.	6665

Snacks		Grocery		Beauty & Hygine	
Nutella Choco Spread	1	Aashirvaad Atta(1kg)	5	Bathing Soap	9
Noodles(1 Pack)	2	Pasta(1kg)	8	Shampoo(1ltr)	3
Lays(10Rs)	0	Basmathi Rice(1kg)	0	Body Lotion(1ltr)	0
Oreo(20Rs)	0	Sunflower Oil(1ltr)	5	Face Cream	6
Chocolate Muffin	7	Refined Sugar(1kg)	0	Shaving Foam	0
Dairy Milk Silk(60Rs)	9	Daal(1kg)	7	Face Mask(1piece)	8
Namkeen(15Rs)	7	Tea Powder(1kg)	0	Hand Sanitizer(50ml)	8

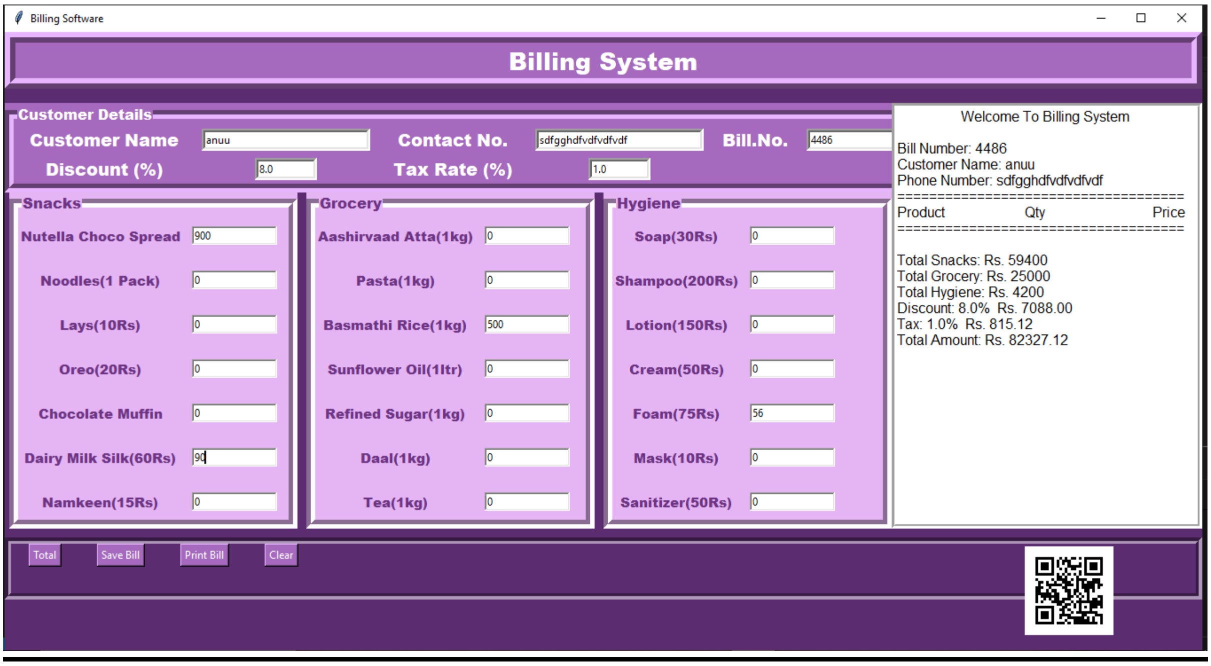
Bill Area

```
Muffins          7      210
Silk             9      540
Namkeen          7      105
Atta             5      210
Pasta            8      960
Oil              5      565
Daal             7      532
Soap             9      270
Shampoo          3      540
Cream            6      780
Mask             8      800
Sanitizer        8      160
------------------------------------
Total Snacks Tax : 52.75 Rs
Total Grocery Tax : 22.67 Rs
Total Beauty&Hygine Tax : 255.0 Rs
Total Bill Amount : 6202.42 Rs
------------------------------------
```

Billing Summery

Total Snacks Price	1055 Rs	Snacks Tax	52.75 Rs
Total Grocery Price	2267 Rs	Grocery Tax	22.67 Rs
Total Beauty & Hygine Price	2550 Rs	Beauty & Hygine Tax	255.0 Rs

(Fig:4)

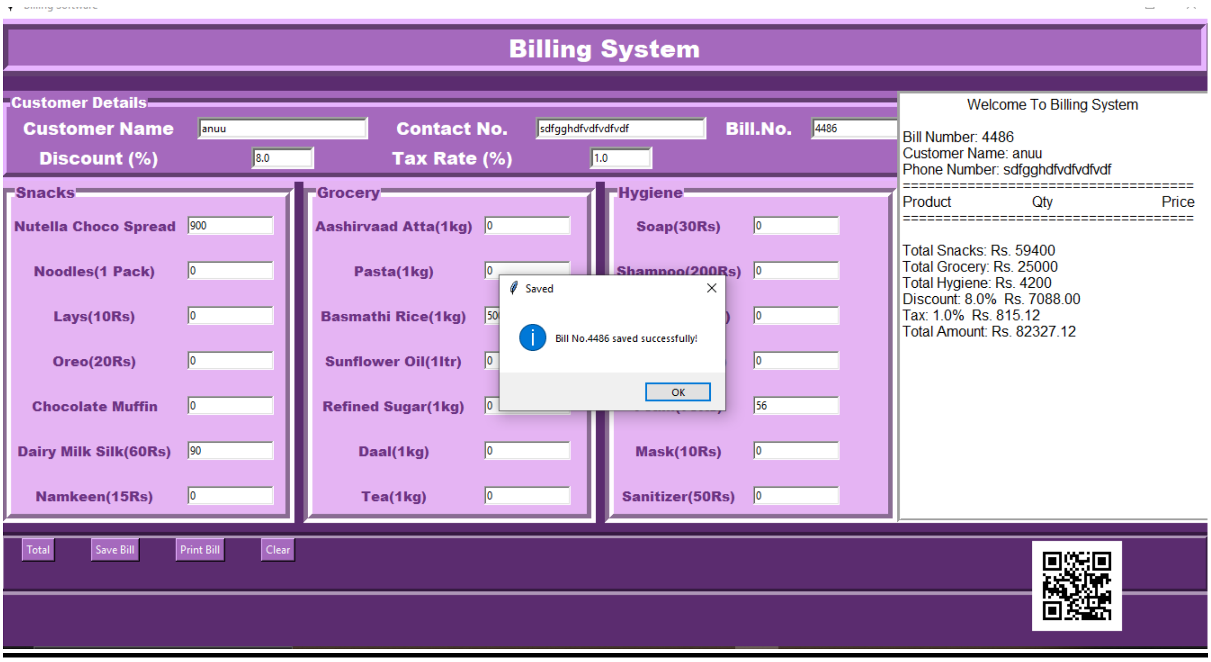


Customer Name	anuu	Contact No.	sdfgghdfvdfvdfvdf	Bill.No.	4486
Discount (%)	8.0	Tax Rate (%)	1.0		

Snacks		Grocery		Hygiene	
Nutella Choco Spread	900	Aashirvaad Atta(1kg)	0	Soap(30Rs)	0
Noodles(1 Pack)	0	Pasta(1kg)	0	Shampoo(200Rs)	0
Lays(10Rs)	0	Basmathi Rice(1kg)	500	Lotion(150Rs)	0
Oreo(20Rs)	0	Sunflower Oil(1ltr)	0	Cream(50Rs)	0
Chocolate Muffin	0	Refined Sugar(1kg)	0	Foam(75Rs)	56
Dairy Milk Silk(60Rs)	90	Daal(1kg)	0	Mask(10Rs)	0
Namkeen(15Rs)	0	Tea(1kg)	0	Sanitizer(50Rs)	0

```
          Welcome To Billing System

Bill Number: 4486
Customer Name: anuu
Phone Number: sdfgghdfvdfvdfvdf
=====================================
Product             Qty          Price
=====================================

Total Snacks: Rs. 59400
Total Grocery: Rs. 25000
Total Hygiene: Rs. 4200
Discount: 8.0%  Rs. 7088.00
Tax: 1.0%  Rs. 815.12
Total Amount: Rs. 82327.12
```

(Fig:5)



Bill No.4486 saved successfully!

Customer Name	anuu	Contact No.	sdfgghdfvdfvdfvdf	Bill.No.	4486
Discount (%)	8.0	Tax Rate (%)	1.0		

Snacks		Grocery		Hygiene	
Nutella Choco Spread	900	Aashirvaad Atta(1kg)	0	Soap(30Rs)	0
Noodles(1 Pack)	0	Pasta(1kg)	0	Shampoo(200Rs)	0
Lays(10Rs)	0	Basmathi Rice(1kg)	50		0
Oreo(20Rs)	0	Sunflower Oil(1ltr)	0		0
Chocolate Muffin	0	Refined Sugar(1kg)	0		56
Dairy Milk Silk(60Rs)	90	Daal(1kg)	0	Mask(10Rs)	0
Namkeen(15Rs)	0	Tea(1kg)	0	Sanitizer(50Rs)	0

```
          Welcome To Billing System

Bill Number: 4486
Customer Name: anuu
Phone Number: sdfgghdfvdfvdfvdf
=====================================
Product             Qty          Price
=====================================

Total Snacks: Rs. 59400
Total Grocery: Rs. 25000
Total Hygiene: Rs. 4200
Discount: 8.0%  Rs. 7088.00
Tax: 1.0%  Rs. 815.12
Total Amount: Rs. 82327.12
```

SYSTEM IMPLEMENTATION

4.1System Coding

```
from tkinter import *
import random
import os
import sys
from tkinter import messagebox
class Bill_App:
def __init__(self,root):
self.root=root
self.root.geometry("1350x700+0+0")
self.root.configure(bg="#5B2C6F")
self.root.title("Billing Software")
title=Label(self.root,text="Billing
System",bd=12,relief=RIDGE,font=("Arial
Black",20),bg="#A569BD",fg="white").pack(fill=X)

# variables
self.nutella=IntVar()
self.noodles=IntVar()
self.lays=IntVar()
self.oreo=IntVar()
self.muffin=IntVar()
self.silk=IntVar()
self.namkeen=IntVar()
self.atta=IntVar()
self.pasta=IntVar()
self.rice=IntVar()
self.oil=IntVar()
self.sugar=IntVar()
self.dal=IntVar()
self.tea=IntVar()
self.soap=IntVar()
self.shampoo=IntVar()
self.lotion=IntVar()
self.cream=IntVar()
self.foam=IntVar()
self.mask=IntVar()
self.sanitizer=IntVar()
self.total_sna=StringVar()
self.total_gro=StringVar()
self.total_hyg=StringVar()
self.a=StringVar()
self.b=StringVar()
self.c=StringVar()
self.c_name=StringVar()
self.bill_no=StringVar()
x=random.randint(1000,9999)
self.bill_no.set(str(x))
self.phone=StringVar()
#=customer details label frame=
details=LabelFrame(self.root,text="Customer Details",font=("Arial
Black",12),bg="#A569BD",fg="white",relief=GROOVE,bd=10)
details.place(x=0,y=80,relwidth=1)
cust_name=Label(details,text="Customer Name",font=("Arial
Black",14),bg="#A569BD",fg="white").grid(row=0,column=0,padx=15)

cust_entry=Entry(details,borderwidth=4,width=30,textvariable=self.c_na
me).grid(row=0,column=1,padx=8)

contact_name=Label(details,text="Contact No.",font=("Arial
Black",14),bg="#A569BD",fg="white").grid(row=0,column=2,padx=10)

contact_entry=Entry(details,borderwidth=4,width=30,textvariable=self.p
hone).grid(row=0,column=3,padx=8)

bill_name=Label(details,text="Bill.No.",font=("Arial
Black",14),bg="#A569BD",fg="white").grid(row=0,column=4,padx=10)

bill_entry=Entry(details,borderwidth=4,width=30,textvariable=self.bill_n
o).grid(row=0,column=5,padx=8)

#=snacks label frame==
snacks=LabelFrame(self.root,text="Snacks",font=("Arial
Black",12),bg="#E5B4F3",fg="#6C3483",relief=GROOVE,bd=10)
snacks.place(x=5,y=180,height=380,width=325)

item1=Label(snacks,text="Nutella Choco Spread",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=0,column=0,pady=1
1)
item1_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.nut
ella).grid(row=0,column=1,padx=10)

item2=Label(snacks,text="Noodles(1 Pack)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=1,column=0,pady=1
1)
item2_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.noo
dles).grid(row=1,column=1,padx=10)

item3=Label(snacks,text="Lays(10Rs)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=2,column=0,pady=1
1)
item3_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.lay
s).grid(row=2,column=1,padx=10)

item4=Label(snacks,text="Oreo(20Rs)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=3,column=0,pady=1
1)
item4_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.ore
o).grid(row=3,column=1,padx=10)

item5=Label(snacks,text="Chocolate Muffin",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=4,column=0,pady=1
1)
item5_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.mu
ffin).grid(row=4,column=1,padx=10)

item6=Label(snacks,text="Dairy Milk Silk(60Rs)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=5,column=0,pady=1
```

```
1)
item6_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.silk
).grid(row=5,column=1,padx=10)

item7=Label(snacks,text="Namkeen(15Rs)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=6,column=0,pady=1
1)
item7_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.na
mkeen).grid(row=6,column=1,padx=10)

#=GROCERY=
grocery=LabelFrame(self.root,text="Grocery",font=("Arial
Black",12),relief=GROOVE,bd=10,bg="#E5B4F3",fg="#6C3483")
grocery.place(x=340,y=180,height=380,width=325)

item8=Label(grocery,text="Aashirvaad Atta(1kg)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=0,column=0,pady=1
1)
item8_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.att
a).grid(row=0,column=1,padx=10)

item9=Label(grocery,text="Pasta(1kg)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=1,column=0,pady=1
1)
item9_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.pa
sta).grid(row=1,column=1,padx=10)

item10=Label(grocery,text="Basmathi Rice(1kg)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=2,column=0,pady=1
1)
item10_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.ri
ce).grid(row=2,column=1,padx=10)

item11=Label(grocery,text="Sunflower Oil(1ltr)",font=("Arial
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=3,column=0,pady=1
1)
item11_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.o
il).grid(row=3,column=1,padx=10)
```

```
item12=Label(grocery,text="Refined Sugar(1kg)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=4,column=0,pady=11)
item12_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.sugar).grid(row=4,column=1,padx=10)

item13=Label(grocery,text="Daal(1kg)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=5,column=0,pady=11)
item13_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.dal).grid(row=5,column=1,padx=10)

item14=Label(grocery,text="Tea Powder(1kg)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=6,column=0,pady=11)
item14_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.tea).grid(row=6,column=1,padx=10)

#=beauty and hygine=
hygine=LabelFrame(self.root,text="Beauty & Hygine",font=("Arial Black",12),relief=GROOVE,bd=10,bg="#E5B4F3",fg="#6C3483")
hygine.place(x=677,y=180,height=380,width=325)

item15=Label(hygine,text="Bathing Soap",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=0,column=0,pady=11)
item15_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.soap).grid(row=0,column=1,padx=10)

item16=Label(hygine,text="Shampoo(1ltr)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=1,column=0,pady=11)
item16_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.shampoo).grid(row=1,column=1,padx=10)

item17=Label(hygine,text="Body Lotion(1ltr)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=2,column=0,pady=11)
item17_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.lo
tion).grid(row=2,column=1,padx=10)

item18=Label(hygine,text="Face Cream",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=3,column=0,pady=11)
item18_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.cream).grid(row=3,column=1,padx=10)

item19=Label(hygine,text="Shaving Foam",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=4,column=0,pady=11)
item19_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.foam).grid(row=4,column=1,padx=10)

item20=Label(hygine,text="Face Mask(1piece)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=5,column=0,pady=11)
item20_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.mask).grid(row=5,column=1,padx=10)

item21=Label(hygine,text="Hand Sanitizer(50ml)",font=("Arial Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=6,column=0,pady=11)
item21_entry=Entry(hygine,borderwidth=2,width=15,textvariable=self.sanitizer).grid(row=6,column=1,padx=10)

#=billarea=
billarea=Frame(self.root,bd=10,relief=GROOVE,bg="#E5B4F3")
billarea.place(x=1010,y=188,width=330,height=372)

bill_title=Label(billarea,text="Bill Area",font=("Arial Black",17),bd=7,relief=GROOVE,bg="#E5B4F3",fg="#6C3483").pack(fill=X)

scrol_y=Scrollbar(billarea,orient=VERTICAL)
self.txtarea=Text(billarea,yscrollcommand=scrol_y.set)
scrol_y.pack(side=RIGHT,fill=Y)
scrol_y.config(command=self.txtarea.yview)
```

```
self.txtarea.pack(fill=BOTH,expand=1)

#=billing menu

billing_menu=LabelFrame(self.root,text="Billing Summery",font=("Arial
Black",12),relief=GROOVE,bd=10,bg="#A569BD",fg="white")
billing_menu.place(x=0,y=560,relwidth=1,height=137)

total_snacks=Label(billing_menu,text="Total Snacks Price",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=0,column=0)
total_snacks_entry=Entry(billing_menu,width=30,borderwidth=2,textvari
able=self.total_sna).grid(row=0,column=1,padx=10,pady=7)

total_grocery=Label(billing_menu,text="Total Grocery
Price",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=1,column=0)
total_grocery_entry=Entry(billing_menu,width=30,borderwidth=2,textvar
iable=self.total_gro).grid(row=1,column=1,padx=10,pady=7)


total_hygine=Label(billing_menu,text="Total Beauty & Hygine
Price",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=2,column=0)
total_hygine_entry=Entry(billing_menu,width=30,borderwidth=2,textvari
able=self.total_hyg).grid(row=2,column=1,padx=10,pady=7)

tax_snacks=Label(billing_menu,text="Snacks Tax",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=0,column=2)
tax_snacks_entry=Entry(billing_menu,width=30,borderwidth=2,textvaria
ble=self.a).grid(row=0,column=3,padx=10,pady=7)

tax_grocery=Label(billing_menu,text="Grocery Tax",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=1,column=2)
tax_grocery_entry=Entry(billing_menu,width=30,borderwidth=2,textvari
able=self.b).grid(row=1,column=3,padx=10,pady=7)


tax_hygine=Label(billing_menu,text="Beauty & Hygine
Tax",font=("Arial Black",11),bg="#A569BD",fg="white").grid(row=2,column=2)
tax_hygine_entry=Entry(billing_menu,width=30,borderwidth=2,textvariable=self.c).grid(row=2,column=3,padx=10,pady=7)

button_frame=Frame(billing_menu,bd=7,relief=GROOVE,bg="#6C3483")
button_frame.place(x=830,width=500,height=95)

button_total=Button(button_frame,text="Total Bill",font=("Arial Black",15),pady=10,bg="#E5B4F3",fg="#6C3483",command=lambda:total(self)).grid(row=0,column=0,padx=12)
button_clear=Button(button_frame,text="Clear Field",font=("Arial Black",15),pady=10,bg="#E5B4F3",fg="#6C3483",command=lambda:clear(self)).grid(row=0,column=1,padx=10,pady=6)
button_exit=Button(button_frame,text="Exit",font=("Arial Black",15),pady=10,bg="#E5B4F3",fg="#6C3483",width=8,command=lambda:exit1(self)).grid(row=0,column=2,padx=10,pady=6)
intro(self)


def total(self):
if (self.c_name.get=="" or self.phone.get()==""):
messagebox.showerror("Error", "Fill the complete Customer Details!!")
self.nu=self.nutella.get()*120
self.no=self.noodles.get()*40
self.la=self.lays.get()*10
self.ore=self.oreo.get()*20
self.mu=self.muffin.get()*30
self.si=self.silk.get()*60
self.na=self.namkeen.get()*15
total_snacks_price=(
self.nu+
self.no+
self.la+
self.ore+
self.mu+
self.si+
self.na)
self.total_sna.set(str(total_snacks_price)+" Rs")
self.a.set(str(round(total_snacks_price*0.05,3))+" Rs")

self.at=self.atta.get()*42
self.pa=self.pasta.get()*120
self.oi=self.oil.get()*113
self.ri=self.rice.get()*160
self.su=self.sugar.get()*55
self.te=self.tea.get()*480
self.da=self.dal.get()*76
total_grocery_price=(
self.at+
self.pa+
self.oi+
self.ri+
self.su+
self.te+
self.da)

self.total_gro.set(str(total_grocery_price)+" Rs")
self.b.set(str(round(total_grocery_price*0.01,3))+" Rs")

self.so=self.soap.get()*30
self.sh=self.shampoo.get()*180
self.cr=self.cream.get()*130
self.lo=self.lotion.get()*500
self.fo=self.foam.get()*85
self.ma=self.mask.get()*100
self.sa=self.sanitizer.get()*20

total_hygine_price=(
self.so+
self.sh+
self.cr+
self.lo+
self.fo+
self.ma+
self.sa)

self.total_hyg.set(str(total_hygine_price)+" Rs")
self.c.set(str(round(total_hygine_price*0.10,3))+" Rs")
self.total_all_bill=(total_snacks_price+
total_grocery_price+
total_hygine_price+
(round(total_grocery_price*0.01,3))+
(round(total_hygine_price*0.10,3))+
(round(total_snacks_price*0.05,3)))
self.total_all_bil=str(self.total_all_bill)+" Rs"
billarea(self)
def intro(self):
self.txtarea.delete(1.0,END)
self.txtarea.insert(END,"\tWELCOME TO SUPER MARKET\n\tPhone-No.739275410")
self.txtarea.insert(END,f"\n\nBill no. : {self.bill_no.get()}")
self.txtarea.insert(END,f"\nCustomer Name : {self.c_name.get()}")
self.txtarea.insert(END,f"\nPhone No. : {self.phone.get()}")
self.txtarea.insert(END,"\n=======================================\n")
self.txtarea.insert(END,"\nProduct\t\tQty\tPrice\n")
self.txtarea.insert(END,"\n=======================================\n")
def billarea(self):
intro(self)
if self.nutella.get()!=0:
self.txtarea.insert(END,f"Nutella\t\t {self.nutella.get()}\t{self.nu}\n")
if self.noodles.get()!=0:
self.txtarea.insert(END,f"Noodles\t\t {self.noodles.get()}\t{self.no}\n")
if self.lays.get()!=0:
self.txtarea.insert(END,f"Lays\t\t {self.lays.get()}\t{self.la}\n")
if self.oreo.get()!=0:
self.txtarea.insert(END,f"Oreo\t\t {self.oreo.get()}\t{self.ore}\n")
if self.muffin.get()!=0:
```

```
self.txtarea.insert(END,f"Muffins\t\t {self.muffin.get()}\t{self.mu}\n")
if self.silk.get()!=0:
self.txtarea.insert(END,f"Silk\t\t {self.silk.get()}\t{self.si}\n")
if self.namkeen.get()!=0:
self.txtarea.insert(END,f"Namkeen\t\t {self.namkeen.get()}\t{self.na}\n")
if self.atta.get()!=0:
self.txtarea.insert(END,f"Atta\t\t {self.atta.get()}\t{self.at}\n")
if self.pasta.get()!=0:
self.txtarea.insert(END,f"Pasta\t\t {self.pasta.get()}\t{self.pa}\n")
if self.rice.get()!=0:
self.txtarea.insert(END,f"Rice\t\t {self.rice.get()}\t{self.ri}\n")
if self.oil.get()!=0:
self.txtarea.insert(END,f"Oil\t\t {self.oil.get()}\t{self.oi}\n")
if self.sugar.get()!=0:
self.txtarea.insert(END,f"Sugar\t\t {self.sugar.get()}\t{self.su}\n")
if self.dal.get()!=0:
self.txtarea.insert(END,f"Daal\t\t {self.dal.get()}\t{self.da}\n")
if self.tea.get()!=0:
self.txtarea.insert(END,f"Tea\t\t {self.tea.get()}\t{self.te}\n")
if self.soap.get()!=0:
self.txtarea.insert(END,f"Soap\t\t {self.soap.get()}\t{self.so}\n")
if self.shampoo.get()!=0:
self.txtarea.insert(END,f"Shampoo\t\t {self.shampoo.get()}\t{self.sh}\n")
if self.lotion.get()!=0:
self.txtarea.insert(END,f"Lotion\t\t {self.lotion.get()}\t{self.lo}\n")
if self.cream.get()!=0:
self.txtarea.insert(END,f"Cream\t\t {self.cream.get()}\t{self.cr}\n")
if self.foam.get()!=0:
self.txtarea.insert(END,f"Foam\t\t {self.foam.get()}\t{self.fo}\n")
if self.mask.get()!=0:
self.txtarea.insert(END,f"Mask\t\t {self.mask.get()}\t{self.ma}\n")
if self.sanitizer.get()!=0:
self.txtarea.insert(END,f"Sanitizer\t\t {self.sanitizer.get()}\t{self.sa}\n")

self.txtarea.insert(END,f"-----------------------------------\n")
if self.a.get()!="0.0 Rs":
```

```
self.txtarea.insert(END,f"Total Snacks Tax : {self.a.get()}\n")
if self.b.get()!="0.0 Rs":
self.txtarea.insert(END,f"Total Grocery Tax : {self.b.get()}\n")
if self.c.get()!="0.0 Rs":
self.txtarea.insert(END,f"Total Beauty&Hygine Tax : {self.c.get()}\n")
self.txtarea.insert(END,f"Total Bill Amount : {self.total_all_bil}\n")
self.txtarea.insert(END,f"--------------------------------------\n")
def clear(self):
self.txtarea.delete(1.0,END)
self.nutella.set(0)
self.noodles.set(0)
self.lays.set(0)
self.oreo.set(0)
self.muffin.set(0)
self.silk.set(0)
self.namkeen.set(0)
self.atta.set(0)
self.pasta.set(0)
self.rice.set(0)
self.oil.set(0)
self.sugar.set(0)
self.dal.set(0)
self.tea.set(0)
self.soap.set(0)
self.shampoo.set(0)
self.lotion.set(0)
self.cream.set(0)
self.foam.set(0)
self.mask.set(0)
self.sanitizer.set(0)
self.total_sna.set(0)
self.total_gro.set(0)
self.total_hyg.set(0)
self.a.set(0)
self.b.set(0)
self.c.set(0)
self.c_name.set(0)
self.bill_no.set(0)
self.bill_no.set(0)
self.phone.set(0)
def exit1(self):
self.root.destroy()

# Apply discount and tax
    discount_amount = (self.discount.get() / 100) * (total_sna + total_gro + total_hyg)
    total_after_discount = total_sna + total_gro + total_hyg - discount_amount
    tax_amount = (self.tax_rate.get() / 100) * total_after_discount
    grand_total = total_after_discount + tax_amount

    # Save totals for display in bill
    self.total_sna.set(f"Rs. {total_sna:.2f}")
    self.total_gro.set(f"Rs. {total_gro:.2f}")
    self.total_hyg.set(f"Rs. {total_hyg:.2f}")

    # Display bill
    self.welcome_bill()
    self.txtarea.insert(END, f"\nTotal Snacks: Rs. {total_sna}")
    self.txtarea.insert(END, f"\nTotal Grocery: Rs. {total_gro}")
    self.txtarea.insert(END, f"\nTotal Hygiene: Rs. {total_hyg}")
    self.txtarea.insert(END, f"\nDiscount: {self.discount.get()}% Rs. {discount_amount:.2f}")
    self.txtarea.insert(END, f"\nTax: {self.tax_rate.get()}% Rs. {tax_amount:.2f}")
    self.txtarea.insert(END, f"\nTotal Amount: Rs. {grand_total:.2f}")

    # Generate QR code for payment
    qr_data = f"Total Amount: Rs. {grand_total:.2f}"
    qr_code = qrcode.make(qr_data)
    qr_code = qr_code.resize((100, 100), Image.LANCZOS)
    qr_code_image = ImageTk.PhotoImage(qr_code)
    self.qr_image_label.configure(image=qr_code_image)
```

```
        self.qr_image_label.image = qr_code_image

    def welcome_bill(self):
        self.txtarea.delete("1.0", END)
        self.txtarea.insert(END, "\tWelcome To Billing System\n")
        self.txtarea.insert(END, f"\nBill Number: {self.bill_no.get()}")
        self.txtarea.insert(END, f"\nCustomer Name: {self.c_name.get()}")
        self.txtarea.insert(END, f"\nPhone Number: {self.phone.get()}")
        self.txtarea.insert(END,
f"\n=======================================")
        self.txtarea.insert(END, f"\nProduct\t\tQty\t\tPrice")
        self.txtarea.insert(END,
f"\n=======================================\n")

    def save_bill(self):
        bill_data = self.txtarea.get("1.0", END)
        filename = f"Bills/{self.bill_no.get()}.txt"
        os.makedirs("Bills", exist_ok=True)
        with open(filename, "w") as f:
            f.write(bill_data)
        messagebox.showinfo("Saved", f"Bill No.{self.bill_no.get()} saved
successfully!")

    def print_bill(self):
        import tempfile
        import os
        import platform

        bill_data = self.txtarea.get("1.0", END)  # Get the text from the bill
area
        if not bill_data.strip():
            messagebox.showerror("Error", "No bill to print!")
            return

        # Create a temporary file to store the bill content
        with tempfile.NamedTemporaryFile(delete=False, suffix=".txt") as
temp_file:
```

```
            temp_file.write(bill_data.encode("utf-8"))
            temp_file_path = temp_file.name

        # Send the file to the printer
        try:
            if platform.system() == "Windows":
                # Windows printing
                os.startfile(temp_file_path, "print")
            elif platform.system() == "Linux":
                # Linux printing using lpr
                os.system(f"lp {temp_file_path}")
            elif platform.system() == "Darwin":
                # macOS printing
                os.system(f"lpr {temp_file_path}")
            else:
                messagebox.showerror("Unsupported OS", "Printing is not supported on this operating system!")
        except Exception as e:
            messagebox.showerror("Error", f"Failed to print the bill: {e}")
        finally:
            # Clean up the temporary file after printing
            os.remove(temp_file_path)


    # Function to find a bill
    def find_bill(self):
        bill_file = f"bills/{self.search_bill.get()}.txt"
        try:
            with open(bill_file, "r") as file:
                self.textarea.delete(1.0, END)
                self.textarea.insert(END, file.read())
        except FileNotFoundError:
            self.textarea.insert(END, "\nBill Not Found!")


    def clear(self):
        self.txtarea.delete("1.0", END)
        for var in [self.nutella, self.noodles, self.lays, self.oreo, self.muffin,
    self.silk, self.namkeen,
            self.atta, self.pasta, self.rice, self.oil, self.sugar, self.dal,
    self.tea, self.soap,
            self.shampoo, self.lotion, self.cream, self.foam, self.mask,
    self.sanitizer]:
          var.set(0)
        self.total_sna.set("")
        self.total_gro.set("")
        self.total_hyg.set("")
        self.c_name.set("")
        self.phone.set("")
        self.bill_no.set(str(random.randint(1000, 9999)))
        self.discount.set(0.0)
        self.tax_rate.set(0.0)
        self.qr_image_label.config(image="")


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

SUMMARY AND CONCLUSIONS

5.1 LIMITATIONS OF THE SYSTEM

While the billing management system developed in this project offers several advantages, it also has certain limitations:

- **Scalability:** The system may face challenges in handling a large volume of transactions and data due to the limitations of SQLite database.

- **Lack of advanced features:** Compared to enterprise-level billing systems, this project may lack certain advanced features such as integration with external payment gateways, inventory management, and tax calculation.

- **Security concerns:** As SQLite is a file-based database system, there may be security concerns regarding data integrity and protection against unauthorized access.

- **Limited platform compatibility**: The GUI developed using Tkinter may not provide an optimal user experience on all platforms, particularly mobile devices.

5.2 CONCLUSION

The urge for the digital restaurant management systems is increasing day by date. Restaurant 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 restaurant 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 hotels so can easily maintained and addressed

5.3 Future Enhancement

To address the limitations mentioned above and improve the functionality of the billing management system, the following enhancements can be considered:

- Migration to a more robust database management system such as MySQL or PostgreSQL to improve scalability and security.

- Integration with external APIs for payment processing and inventory management.

- Implementation of advanced reporting features including graphical representations of sales data.

- Development of a web-based interface using frameworks like Django or Flask for better platform compatibility

REFERENCES

- **Online Learning Platforms:**
  Platforms like Google often have courses related to voice recognition and Python programming.
  Ex – Udemy.com, techmosphere.in
- **YouTube Tutorials:**
  Many creators share step-by-step tutorials on building voice assistants. Search for relevant videos on platforms like YouTube.
  Ex – codewithharry, Python with Bro Code.