

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By
ARCHANA B
38120010



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU

OCTOBER 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ARCHANA B** (Reg.no. 38120010) who carried out the project entitled "**Supermarket Billing System using Python**" under our supervision from Jan 2021 to April 2021.

Internal Guide

Dr. G. Mathivanan M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the Project Report entitled “**Supermarket Billing System using Python**” done by me under the guidance of **Dr. G. Mathivanan M.E., Ph.D.**, and is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering / Technology degree in Information Technology.

DATE: 23/01/2021

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr. R. SUBHASHINI M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. Mathivanan M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project

TRAINING CERTIFICATE



INT: RE20S041

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **ARCHANA B** student of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, BTECH/IT** has undergone the Internship Training through online in our concern entitled **PYTHON** from **12th Aug 2020** to **26th Aug 2020** in relevant departments related to their academic studies.

During the above period, the performance was good & we wish great success in all your future endeavours.

J. more
Technology drives the world. We are the people who are behind the technology.

Authorised Signature



No-31, 1st Floor, Alagesan Street, Tambaram West, Chennai-45.
Office : 044-42077204 | +91-8122266653 www.retechsolutions.in



07/03/2024

Archana B

has successfully completed

Programming for Everybody (Getting Started with Python)

an online non-credit course authorized by University of Michigan and offered through Coursera

A handwritten signature in black ink, appearing to read 'Charles Lawrence', followed by a horizontal line.

Charles Lawrence
Clinical Professor, School of Education
University of Michigan

**COURSE
CERTIFICATE**



Verify at coursera.org/verify/812N7J882V9586
Coursera has confirmed the identity of this individual and
their participation in the course.

ABSTRACT

SUPERMARKET BILLING SYSTEM USING PYTHON

The project is on Supermarket Billing System. Supermarket is the place where customers come to purchase their daily using products and pay for that. So, there is a need to calculate how many products are sold and to generate the bill for the customer.

In our project we have 2 users. First one is the administrator who will decide the taxes and commissions on the products and can see the report of any product. He is the one who will decide the products available for customers.

The second one is the customer or the billing manager who can purchase the items available or can make the bill for the customers. This project can also be used for online purchasing as the customer can access it easily.

CHAPTER NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 OBJECTIVE	11
	1.2 WORKFLOW OF THE PROJECT	11
2	AIM AND SCOPE	12
	2.1 AIM	12
	2.2 SCOPE	12
		12
3	SYSTEM ANALYSIS AND DESIGN	13
	3.1 USER INTERFACE	13
	3.2 FRAMEWORK	14
4	RESULT AND DISCUSSION	13
	4.1 SYSTEM REQUIREMENTS	13
	4.2 RESULT	14
5	CONCLUSION AND FUTURE WORK	17
	5.1 CONCLUSION	17
	A.SCREENSHOTS	21
	B.SOURCE CODE	23

FIGURE NO	FIGURE NAME	PAGE NO
3.1	DATA FLOW	12
4.1	OUTPUT	20

CHAPTER 1

INTRODUCTION

OBJECTIVE

To make software fast in processing, with good user interface so that user can change it and it should be used for a long time without error and maintenance.

WORKFLOW OF THE PROJECT

Work in the Supermarket will be done in the following way:

- The product will come in the store.
- Data entry operator will enter the information of the product in database.
- The Administrator will enter the taxes and commissions for each product.
- The customer will come and take the basket with him/her and choose the product and took it to the counter.
- The bill calculating operator will check the products with the bar code detecting machine then it will match with product-id then it will show its information and price and the bill will be calculated and total payment will be shown.
- Customer will pay for the products.
- All the products will be packed and delivered to the customer.

Work in the Supermarket will be done in the following way:

1. The product will come in the store.
2. The Administrator will enter the information of the product in database and price and discount available for each product.
3. The customer will come and take the basket with him/her and choose the product and take it to the counter.
4. The bill calculating operator will enter the product number then it will show its information and price and the bill will be calculated and total payment will shown.
5. Customer will pay for the products.
6. All the products will be packed and delivered to
7. the customer.

AIM AND SCOPE

AIM

Supermarket is the place where customers come to purchase their daily using products and pay for that. So, there is a need to calculate how many products are sold and to generate the bill for the customer.

EXISTING SYSTEM

A person entering shopping mall does not know what are the items located in the Fledge he/she wants to purchase in the shopping mall. He/ This is a time-consuming process and the searching process vexes the customer. Transaction is time consuming and he/she will have to exit.

PROPOSED SYSTEM

To overcome all these problems, we are developing a solution that is, instead of going to home and search items to buy we can search for the item in our mobile through wireless communication which makes your shopping easier and Automatically we can know the items offer on that particular product through the mobile. This reduces the burden on the customer.

FEASIBILITY STUDY

Many feasibility studies are dillusioning for both users and analysts. First the study often presupposes that when feasibility document is being prepared, The analyst is in a position to evaluate solutions.

Secondly, most studies tend to overlook the confusion inherent in the system development-the constraints and the assumed attitudes. If the feasibility study is to serve as a decision document for a project, it must answer 3 key questions.

Three key considerations involved in the feasibility analysis:

- Economical
- Technical
- Behavioral

SCOPE

Our project has a big scope to do.

We can:

- Calculate the bill.
- Give the bill to the customer.
- Store how many products are sold.
- Store products and their prices and with other information.

This application allows the users to search for the desired product in the shop so that it saves the time of the user instead of checking the shop. User can check all the information's like product cost, floor number, rack number and any concession on the products and many more details through their modules.

And they can pay the bill through their mobile itself by using the credit cards and debit cards. It minimizes the risk taken by the people to pay the bill. By using this application we can check the items present or not

from the current location and purchase the items using credit card, so we can reduce the time for shopping

FUNCTIONALITIES PROVIDED BY PYTHON ON SUPERMARKET BILLING SYSTEM

- Provides the searching facilities based on various factors, Such as Products, Purchasing, Stores, Customer.
- It tracks all the information of Payment, Sales, Stores etc.
- Manage the information of Payment
- Shows the information and description of the Products, Purchasing
- To increase efficiency of managing the Products, Payment
- It deals with monitoring the information and transactions of stores
- Manage the information of stores and integration of all records of Customer.
- Editing, adding and updating of Records is improved which results in proper resource management of Products data.

USER INTERFACE

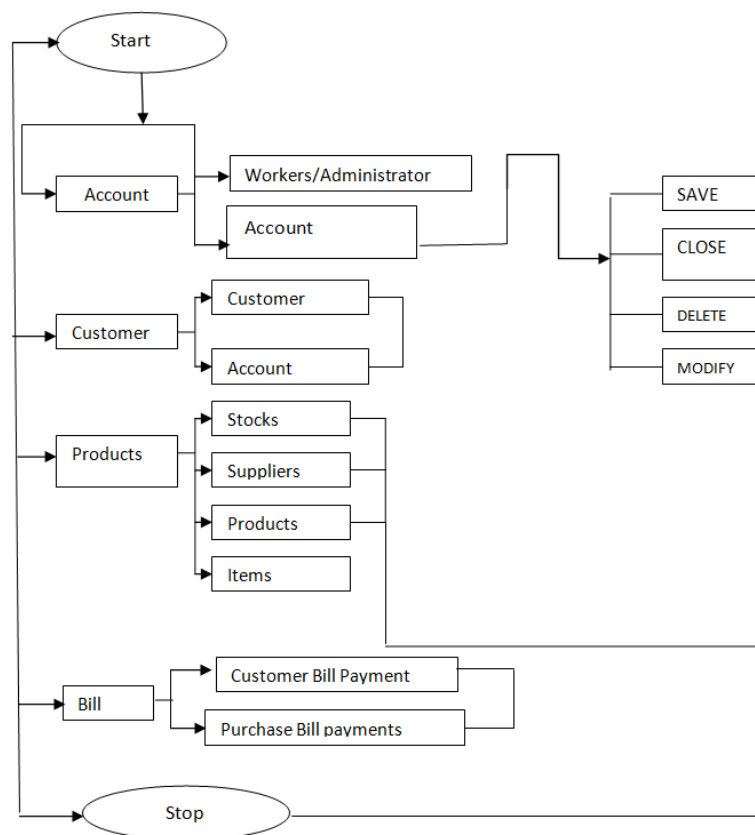
The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse

and the appearance of a desktop. It is also the way through which a user interacts with an application or a website

PYTHON WEB FRAMEWORK

A Web framework is a collection of packages or modules which allow developers to write Web applications or services without having to handle such low-level details as protocols, sockets or process/thread management. The Python web framework used in this project is Flask.

DATA FLOW



RESULTS AND DISCUSSION

SYSTEM REQUIREMENTS

- Operating System: Windows/Mac
- RAM: 4GB
- Processor: 64x 1.0Ghz processor
- ROM: 8GB

FUNCTIONAL REQUIREMENTS

In the billing application, the customers can view the details about the product from the implementation of the product search module.

- The new user can login into the system by registering into the Admin and the login into the system
- The customer can view the information about the offers on the desired products from the profile based adds module
- The user can pay the bill using credit card/debit card from the implementation of the bill payment module

NON-FUNCTIONAL REQUIREMENTS

Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given. It is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system.

This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the

system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

CONCLUSION

Integrated Shopping Application with mobile as domain is successfully implemented with all the features mentioned in the SRS.

The project to a huge extent simplifies the process of obtaining information whenever a RFID Readers enabled device request information.

The software thus developed has been implemented successfully which has been under observation for the past few days.

Various tests have been performed to scrutinize the validation of each data and the errors were stopped out and then finally cleared in a sophisticated manner.

Even though the software has many advantages, some limitations also exists which are meager and negligible. Those limitations need not to be considered since it does not affect the system as a whole.

An attempt is made to maintain maximum perfection in documenting the software in a simple, precise and in a self-explanatory manner

SOURCE CODE

```
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 System")
        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_name).
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.phone  
)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_no).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=11)
```

```
item1_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.nutella)  
.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=11)
```

```
item2_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.noodle  
s).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=11)
```

```
item3_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.lays).gr  
id(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=11)
```

```
item4_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.oreo).g  
rid(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=11)
```

```
item5_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.muffin).  
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=11)
```

```
item6_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.silk).gri  
d(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=11)
```

```
item7_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.namke  
en).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=11)
```

```
item8_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.atta).gr  
id(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=11)
```

```
item9_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.pasta).  
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=11)
```

```
item10_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.rice).  
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=11)
```

```
item11_entry=Entry(grocery,borderwidth=2,width=15,textvariable=self.oil).gr  
id(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).g  
rid(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).g  
rid(row=6,column=1,padx=10)
```

```
#=====beauty and  
hygiene=====
```

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

```
item15=Label(hygiene,text="Bathing Soap",font=("Arial  
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=0,column=0,pady=11)
```

```
item15_entry=Entry(hygiene,borderwidth=2,width=15,textvariable=self.soap).  
grid(row=0,column=1,padx=10)
```

```
item16=Label(hygiene,text="Shampoo(1ltr)",font=("Arial  
Black",11),bg="#E5B4F3",fg="#6C3483").grid(row=1,column=0,pady=11)
```

```
item16_entry=Entry(hygiene,borderwidth=2,width=15,textvariable=self.shamp  
oo).grid(row=1,column=1,padx=10)
```

```
item17=Label(hygiene,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.lotion).  
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.sanitiz  
er).grid(row=6,column=1,padx=10)
```

```
#=====billar  
ea=====
```

```
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,textvariable  
=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,textvariable  
=self.total_gro).grid(row=1,column=1,padx=10,pady=7)
```

```

total_hygiene=Label(billing_menu,text="Total Beauty & Hygiene
Price",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=2,column=0)

total_hygiene_entry=Entry(billing_menu,width=30,borderwidth=2,textvariable
=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,textvariable=
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,textvariable=
self.b).grid(row=1,column=3,padx=10,pady=7)

tax_hygiene=Label(billing_menu,text="Beauty & Hygiene
Tax",font=("Arial
Black",11),bg="#A569BD",fg="white").grid(row=2,column=2)

tax_hygiene_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",width=7,command=lambd
a:total(self)).grid(row=0,column=0,padx=16)
```

```
button_clear=Button(button_frame,text="Clear Field",font=("Arial
Black",15),pady=10,bg="#E5B4F3",fg="#6C3483",width=8,command=lambd
a:clear(self)).grid(row=0,column=1,padx=11,pady=6)
```

```
button_exit=Button(button_frame,text="Exit",font=("Arial
Black",15),pady=10,bg="#E5B4F3",fg="#6C3483",width=5,command=lambd
a:exit1(self)).grid(row=0,column=2,padx=11,pady=6)
```

```
button_print = Button(button_frame, text="Print", font=("Arial Black",
15), pady=10, bg="#E5B4F3",fg="#6C3483", command=lambda:
total(self)).grid(row=0, column=3, padx=8,pady=3)
```

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

OUTPUT

Billing System

Customer Details

Customer Name: ARCHANA Contact No.: 86466431368 Bill.No.: 6573

Snacks	Grocery	Beauty & Hygiene
Nutella Choco Spread: 4	Aashirvaad Atta(1kg): 3	Bathing Soap: 0
Noodles(1 Pack): 4	Pasta(1kg): 0	Shampoo(1ltr): 4
Lays(10Rs): 5	Basmathi Rice(1kg): 0	Body Lotion(1ltr): 7
Oreo(20Rs): 7	Sunflower Oil(1ltr): 4	Face Cream: 0
Chocolate Muffin: 9	Refined Sugar(1kg): 5	Shaving Foam: 6
Dairy Milk Silk(60Rs): 0	Daal(1kg): 0	Face Mask(1piece): 5
Namkeen(15Rs): 0	Tea Powder(1kg): 3	Hand Sanitizer(50ml): 0

Bill Area

Noodles	4	160
Lays	5	50
Oreo	7	140
Muffins	9	270
Atta	3	126
Oil	4	452
Sugar	5	275
Tea	3	1440
Shampoo	4	720
Lotion	7	3500
Foam	6	510
Mask	5	500
Total Snacks Tax : 55.0 Rs		
Total Grocery Tax : 22.93 Rs		
Total Beauty&Hygiene Tax : 523.0 Rs		
Total Bill Amount : 9223.93 Rs		

Billing Summary

Total Snacks Price	1100 Rs	Snacks Tax	55.0 Rs
Total Grocery Price	2293 Rs	Grocery Tax	22.93 Rs
Total Beauty & Hygiene Price	5230 Rs	Beauty & Hygiene Tax	523.0 Rs

Total Bill **Clear Field** **Exit** **Print**

Billing System

Customer Details

Customer Name: Contact No.: Bill.No.:

Snacks	Grocery	Beauty & Hygiene
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

Billing Summary

Total Snacks Price	0	Snacks Tax	0
Total Grocery Price	0	Grocery Tax	0
Total Beauty & Hygiene Price	0	Beauty & Hygiene Tax	0

Total Bill **Clear Field** **Exit**