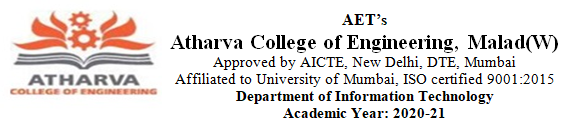
****

Online Stationary

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Rohit Gehlot Roll No.33(SEIT 1)

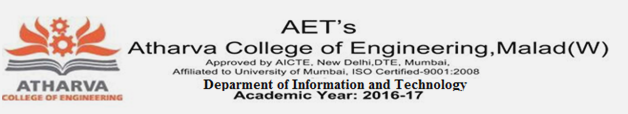
Ayush Yelne Roll No.63(SEIT 2)

Pranav Satam Roll No.38(SEIT 2)

Naima Dalwai Roll No.75(TEIT 2)

Supervisor:

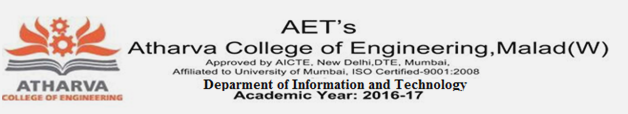
Prof. Odilia Gonsalves



Department of Information Technology

Atharva College of Engineering

Year: 2020-21



**ATHARVA COLLEGE OF ENGINEERING**

**MALAD (W), MUMBAI 400 095**

**YEAR: 2020-2021**

CERTIFICATE

*This is to certify that*

**Ayush Yelne(Roll No.63)TEIT-2**

*have submitted the Mini Project report for the requirements of the Bachelor of Engineering in Information Technology satisfactorily*

on

**“Online Stationary”**

*As prescribed by the* ***University of Mumbai*** *Under the guidance of*

**PROJECT GUIDE H.O.D. PRINCIPAL**

**INTERNAL EXAMINER COLLEGE SEAL EXTERNAL EXAMINER**

Approval for T.E. Mini Project Report

This project report entitled (***Online Stationary***) by (**Ayush Yelne**) is approved for the degree of ***Bachelor of Engineering in Information Technology***

**Examiners:**

* 1. -----------------------------------

(Internal Examiner Name & Sign)

* 1. -----------------------------------

(External Examiner Name & Sign)

Date: 29/10/2021

Place: Mumbai

**Contents**

**1.** **Introduction**

**1.1** **Introduction**

**1.2** **Motivation**

**1.3 Problem Statement**

**1.4 Objectives**

**2.** **Literature Survey**

**3.** **Proposed System**

**3.1 Algorithm and Process Design**

**3.2 Software Requirement Specification**

**3.3 Implementation Code**

**3.4**  **Output Screenshots**

**4.**  **Conclusion**

**5. Future work**

* + 1. **INTRODUCTION**
  1. **Introduction**
* The ‘Billing System’ is a Desktop-based application which is used to generate bills of orders received in super market.
* This application keeps tracks of all bills generated till date in database.
* It contains login system for employees for making bills.
* It has a feature of accessing any bill with its bill number from database.
  1. **Motivation**
* Earlier, people use to make bills on papers and there was chance of human error and also waste of paper and that increases work load.
* Now using this application we can help them to get their work done fast with less errors.
* Many people print 2copies of bill,1 for customer and 1 for their shop record. Now with this app it stores all the bills record in database.
* So, we got the motivation to create such app that can help people and save the natural resources.
  1. **Problem Statement**

Nowadays, billing process is done either manually or using barcode readers which are highly time consuming. Moreover, barcodes are susceptible to damage and failure rate in barcodes are relatively high.

In order to overcome these disadvantages, the billing process has to be automated.

An automated and intuitive billing solution makes things easier in such a situation.

Current system requires a lot of manual paper work i.e. printing bill, keeping them safe in file with each record and also there is always a chance of human error in calculations and bills getting lost.

So there is need of application that can overcome all these problems and help people reduce work

* 1. **Objectives**
* The main objective of Billing system is to manage details of bills, reduce human error.
* Provide a database that can store all information.
* Develop a system that will lessen process delay in terms of releasing receipts and customer bill.
* Save time of employee and customer.

* + 1. **LITERATURE SURVEY**
* In local grocery shop, there are very less who use the billing system. They make bills on paper and there is chance of error in calculation and also this process takes a lot of time.
* It also cause a rush at grocery store which is dangerous in ongoing pandemic.
* Instead they can make use of such application that can make bills, store them in database and also errorless and also it saves time.

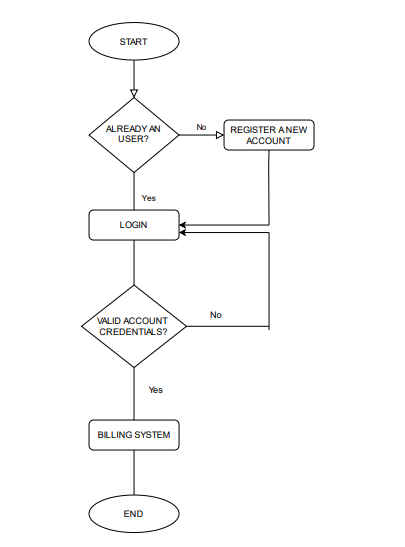


* + 1. **PROPOSED SYSTEM**

**3.1 Algorithm and Process Design**

* First is admin registration, then login window.
* Login details are stored in database.
* After successful login, Billing window appears.
* Enter details of customers and then select products.
* Then click on generate bill and it will also save the bill amount with customer name in database.

**Flowchart**



**3.2 Software requirement specification**

* Visual studio code
* Python Version 3.9
* Tkinter
* Phpmyadmin
* Other GUI Software
  1. **Implementation Code**

**Register:**

from tkinter import \*

from PIL import Image, ImageTk

from tkinter import ttk, messagebox

import getpass

import pymysql

class Register:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Registeration Window")

self.root.geometry("1181x700+100+50")

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

self.bg = ImageTk.PhotoImage(file="grocery.jfif")

self.bg\_image = Label(self.root, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)

frame1 = Frame(self.root, bg="white")

frame1.place(x=250, y=100, width=700, height=500)

title = Label(frame1, text="REGISTER HERE", font=("times new roman", 20, "bold"), bg="white", fg="green").place(

x=50, y=30)

f\_name = Label(frame1, text="First Name", font=("times new roman", 15, "bold"), bg="white", fg="black").place(

x=50, y=100)

self.txt\_fname = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_fname.place(x=50, y=130, width=250)

l\_name = Label(frame1, text="Last Name", font=("times new roman", 15, "bold"), bg="white", fg="black").place(

x=370, y=100)

self.txt\_lname = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_lname.place(x=370, y=130, width=250)

contact = Label(frame1, text="Contact No.", font=("times new roman", 15, "bold"), bg="white", fg="black").place(

x=50, y=170)

self.txt\_contact = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_contact.place(x=50, y=200, width=250)

email = Label(frame1, text="Email", font=("times new roman", 15, "bold"), bg="white", fg="black").place(x=370,

y=170)

self.txt\_email = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_email.place(x=370, y=200, width=250)

gender = Label(frame1, text="Gender", font=("times new roman", 15, "bold"), bg="white", fg="black").place(x=50,

y=240)

self.txt\_gender = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_gender.place(x=50, y=270, width=250)

address = Label(frame1, text="City", font=("times new roman", 15, "bold"), bg="white", fg="black").place(x=370,

y=240)

self.txt\_address = Entry(frame1, font=("times new roman", 15), bg="lightgray")

self.txt\_address.place(x=370, y=270, width=250)

passwd = Label(frame1, text="Password", font=("times new roman", 15, "bold"), bg="white", fg="black").place(

x=50, y=310)

self.txt\_password = Entry(frame1, show='\*', font=("times new roman", 15), bg="lightgray")

self.txt\_password.place(x=50, y=340, width=250)

confirm\_password = Label(frame1, text="Confirm Password", font=("times new roman", 15, "bold"), bg="white",

fg="black").place(x=370, y=310)

self.txt\_confirmpassword = Entry(frame1, show='\*', font=("times new roman", 15), bg="lightgray")

self.txt\_confirmpassword.place(x=370, y=340, width=250)

self.var\_chk = IntVar()

chk = Checkbutton(frame1, text="I agree the terms and conditions", variable=self.var\_chk, onvalue=1, offvalue=0,

bg="white", font=("times new roman", 12)).place(x=50, y=380)

btn\_register = Button(frame1, cursor="hand2", command=self.register\_data, text="REGISTER",

font=("times new roman", 20, "bold"), fg="white", bg="#B00857").place(x=150, y=420,

width=180,

height=40)

btn\_login = Button(frame1, command=self.login\_window, cursor="hand2", text="SIGN IN",

font=("times new roman", 20, "bold"), fg="white", bg="#B00857").place(x=400, y=420,

width=180,

height=40)

def login\_window(self):

self.root.destroy()

import login

def register\_data(self):

if self.txt\_fname.get() == "" or self.txt\_lname.get() == "" or self.txt\_contact.get() =="" or self.txt\_email == "" or self.txt\_gender.get() == "" or self.txt\_address.get() == "" or self.txt\_password.get() == "" or self.txt\_confirmpassword.get() == "":

messagebox.showerror("Error", "All fields are required", parent=self.root)

elif self.txt\_password.get() != self.txt\_confirmpassword.get():

messagebox.showerror("Error", "Password doesn't match", parent=self.root)

elif self.var\_chk.get() == 0:

messagebox.showerror("Error", "Please agree to our terms and conditions", parent=self.root)

else:

try:

conn = pymysql.connect(host="localhost", user="root", password="", database="grocery")

cur = conn.cursor()

cur.execute(

"insert into register (f\_name, l\_name, contact, email, gender, city, password) values (%s, %s, %s, %s, %s, %s, %s)",

(self.txt\_fname.get(),

self.txt\_lname.get(),

self.txt\_contact.get(),

self.txt\_email.get(),

self.txt\_gender.get(),

self.txt\_address.get(),

self.txt\_password.get()

))

conn.commit()

conn.close()

messagebox.showinfo("Success", "Registration Successful", parent=self.root)

except Exception as es:

messagebox.showinfo("Error", f"Error due to {str(es)}", parent=self.root)

root = Tk()

obj = Register(root)

root.mainloop()

**Login:**

from tkinter import \*

from PIL import Image, ImageTk, ImageDraw

from datetime import \*

import time

from math import \*

import pymysql

from tkinter import messagebox

class Login\_window:

def \_\_init\_\_(self,root):

self.root = root

self.root.title("Clock")

self.root.geometry("1350x700")

self.root.config(bg="#021e2f")

left\_lbl = Label(self.root, bg="#08A3D2", bd=0)

left\_lbl.place(x=0, y=0, relheight=1, width=600)

right\_lbl = Label(self.root, bg="#031F3C", bd=0)

right\_lbl.place(x=600, y=0, relheight=1, relwidth=1)

login\_frame = Frame(self.root, bg="white")

login\_frame.place(x=250, y=100, width=800,height=500)

title = Label(login\_frame, text="LOGIN HERE", font=("times new roman", 30, "bold"), bg="white", fg="#08A3D2").place(x=250, y=50)

username = Label(login\_frame, text="USERNAME", font=("times new roman", 18, "bold"), bg="white", fg="gray").place(x=250, y=150)

self.username\_txt = Entry(login\_frame, font=("times new roman", 15), bg="lightgray")

self.username\_txt.place(x=250, y=180, width=350, height=35)

passwd = Label(login\_frame, text="PASSWORD", font=("times new roman", 18, "bold"), bg="white", fg="gray").place(x=250, y=250)

self.passwd = Entry(login\_frame, show='\*', font=("times new roman", 15), bg="lightgray")

self.passwd.place(x=250, y=280, width=350, height=35)

btn\_reg = Button(login\_frame, command=self.register\_window, cursor="hand2", text="Register New Account?", font=("times new roman",14), bg="white", bd=0, fg="#B00857").place(x=250,y=320)

btn\_login = Button(login\_frame, command=self.login\_data, cursor="hand2", text="Login", font=("times new roman", 20, "bold"), fg="white", bg="#B00857").place(x=250, y=380, width=180, height=40)

self.lbl = Label(self.root, bg="#081923", bd=0)

self.lbl.place(x=90, y=120, height=450, width=350)

self.working()

def register\_window(self):

self.root.destroy()

import register

def login\_data(self):

if self.username\_txt.get() == "" or self.passwd.get() =="":

messagebox.showerror("Error", "All fields are required", parent=self.root)

else:

try:

conn = pymysql.connect(host="localhost", user="root", password="", database="grocery")

cur = conn.cursor()

cur.execute("select \* from register where email=%s and password=%s", (self.username\_txt.get(), self.passwd.get()))

row = cur.fetchone()

if row == None:

messagebox.showerror("Error", "Invalid user or password", parent=self.root)

else:

messagebox.showinfo("Success", "Welcome", parent=self.root)

self.root.destroy()

import billingsystem

conn.close()

except Exception as es:

messagebox.showerror("Error", f"Error due to: {str(es)}", parent=self.root)

def clock\_image(self, hr, min\_, sec\_):

clock = Image.new("RGB", (400, 400), (8, 258, 35))

draw = ImageDraw.Draw(clock)

bg = Image.open("c.jfif")

bg = bg.resize((300, 300), Image.ANTIALIAS)

clock.paste(bg, (50, 50))

origin = 200, 200

draw.line((origin, 200+50\*sin(radians(hr)), 200-50\*cos(radians(hr))), fill="#DF005E", width=4)

draw.line((origin, 200+80\*sin(radians(min\_)), 200-80\*cos(radians(min\_))), fill="white", width=3)

draw.line((origin, 200+100\*sin(radians(sec\_)), 200-100\*cos(radians(sec\_))), fill="yellow", width=2)

draw.ellipse((195, 195, 210, 210), fill="#1AD5D5")

clock.save("clock\_new.png")

def working(self):

h = datetime.now().time().hour

m = datetime.now().time().minute

s = datetime.now().time().second

hr = (h/12)\*360

min\_ = (m/60)\*360

sec\_ = (s/60)\*360

self.clock\_image(hr, min\_, sec\_)

self.img = ImageTk.PhotoImage(file="clock\_new.png")

self.lbl.config(image=self.img)

self.lbl.after(200, self.working)

root = Tk()

obj = Login\_window(root)

root.mainloop()

**Billing page:**

from tkinter import \*

import random

import os

import pymysql

import time

import sys

from tkinter import messagebox

import time as tm

class Bill\_App:

def \_\_init\_\_(self, root):

self.root = root

self.root.geometry("1350x700+0+0")

self.root.configure(bg="#FFFF00")

self.root.title("Billing System")

title = Label(self.root, text="Anand Super Market", bd=12, relief=RIDGE, font=("Arial Black", 20), bg="#A569BD", fg="white").pack(fill=X)

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()

def time\_update():

currentt = tm.strftime ("%I:%M:%S %p")

clock.configure(text=currentt)

clock.after(1000,time\_update)

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)

clock\_label1=Label(details,text="Time:",font=("Arial Black",14),bg="#A569BD",fg="white").grid(row=0,column=6,padx=8)

clock=Label(details,font=("Calibri",14),bg="#A569BD",fg="white")

clock.grid(row=0,column=7,padx=10)

time\_update()

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.noodles).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).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=11)

item4\_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.oreo).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=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).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=11)

item7\_entry=Entry(snacks,borderwidth=2,width=15,textvariable=self.namkeen).grid(row=6,column=1,padx=10)

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).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=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).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)

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.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.sanitizer).grid(row=6,column=1,padx=10)

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=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\_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,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\_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"

try:

conn = pymysql.connect(host="localhost", user="root", password="", database="grocery")

cur = conn.cursor()

cur.execute(

"insert into customer ( C\_name,C\_contact,Bill\_no,Amount) values (%s, %s, %s, %s)",

(self.c\_name.get(),

self.phone.get(),

self.bill\_no.get(),\*

str(self.total\_all\_bill),

))

conn.commit()

conn.close()

except Exception as es:

messagebox.showinfo("Error", f"Error due to {str(es)}", parent=self.root)

billarea(self)

def intro(self):

self.txtarea.delete(1.0,END)

self.txtarea.insert(END,"\t ANAND SUPER MARKET\n\tPhone-No.739275410")

self.txtarea.insert(END,f"\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"------------------------------------\n")

self.txtarea.insert(END,f"Total Bill Amount : {self.total\_all\_bil}\n")

self.txtarea.insert(END,f"------------------------------------\n")

self.txtarea.insert(END,f"Thankyou for shopping with us!!\n")

def clear(self):

rand = random.randint(1000,9999)

self.bill\_no.set(rand)

self.txtarea.delete(13.0,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.phone.set(0)

def exit1(self):

self.root.destroy()

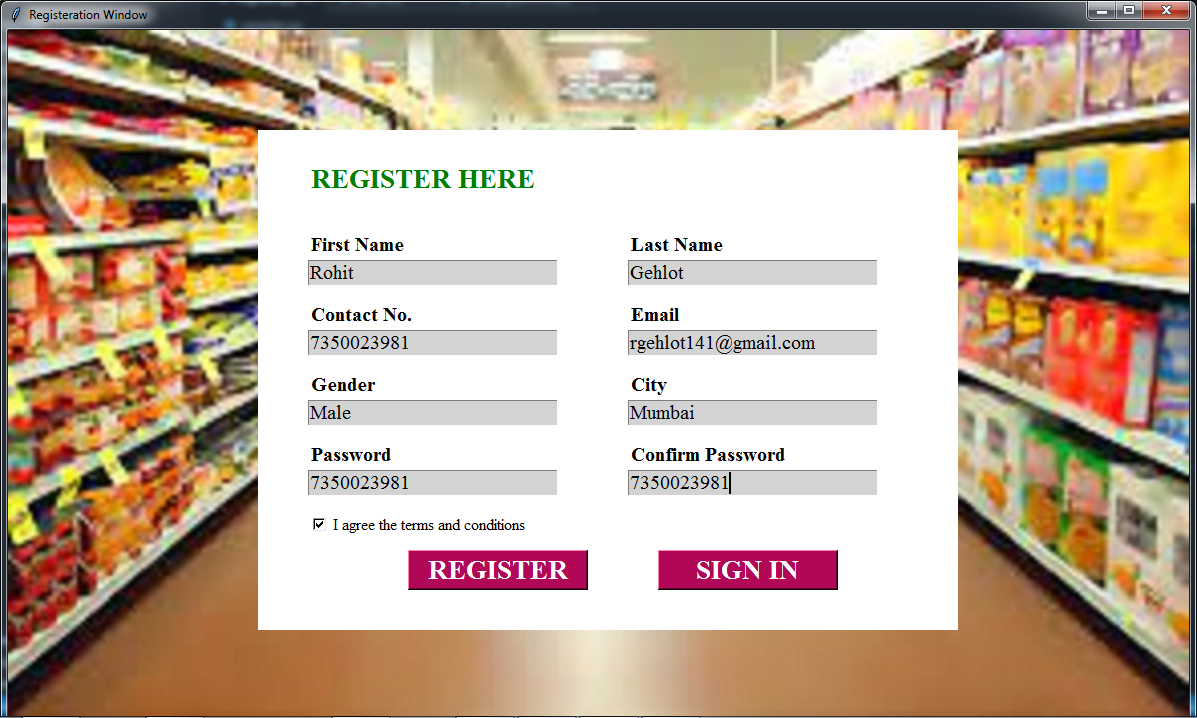
root=Tk()

obj=Bill\_App(root)

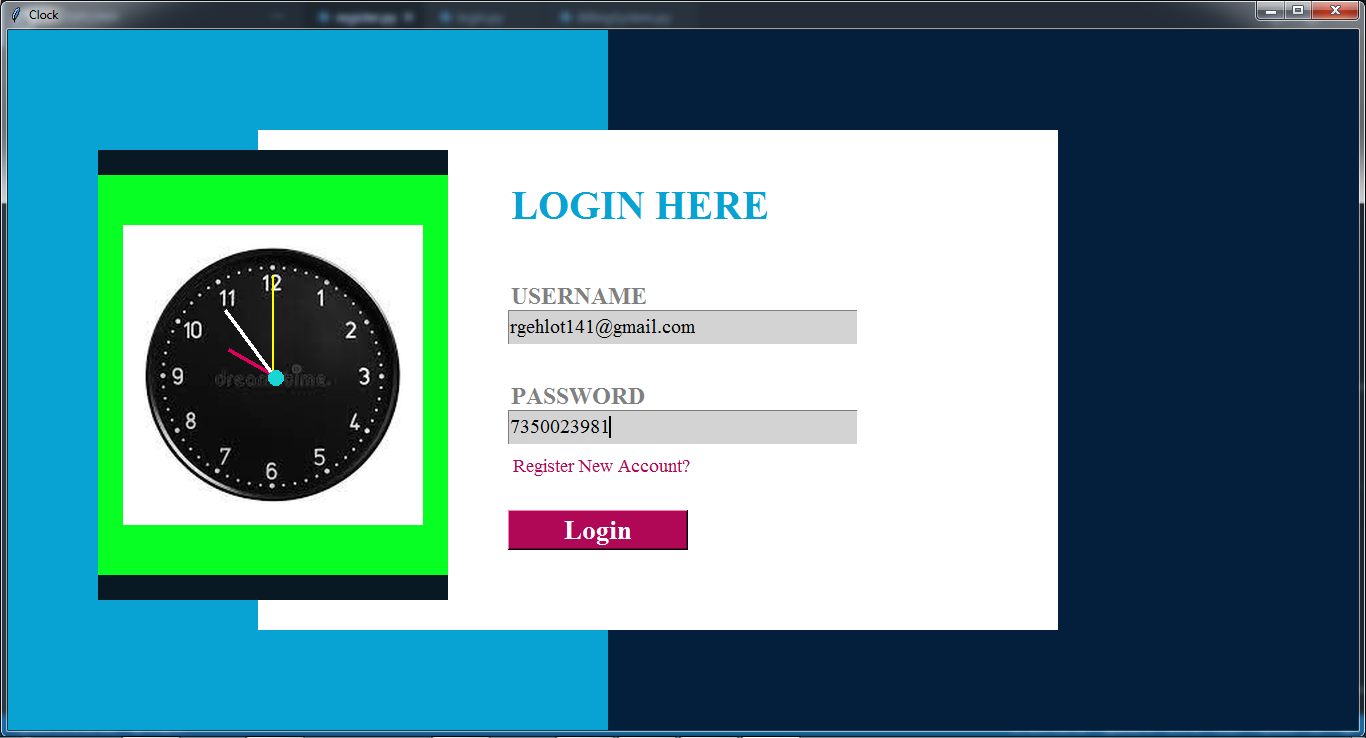
root.mainloop()

* 1. **Output Screenshots**

Register page:



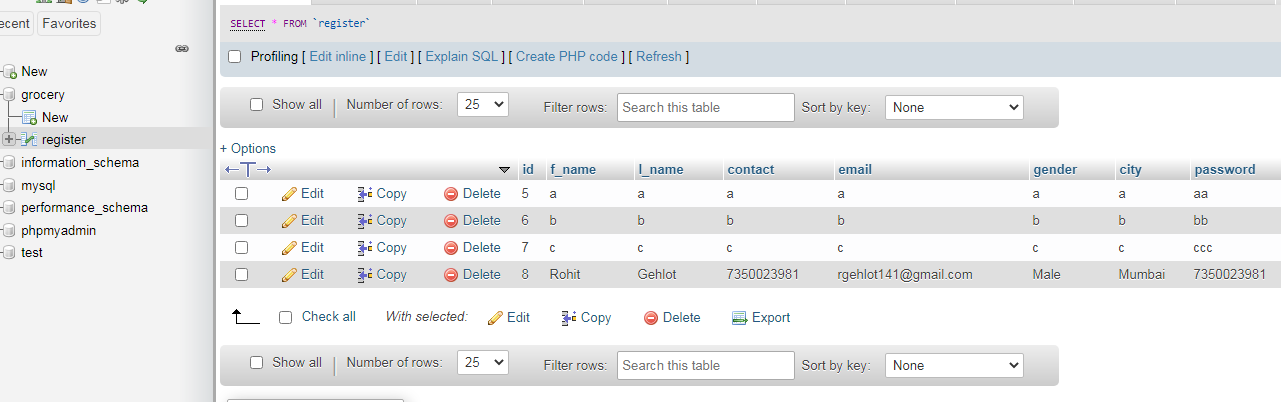
Login Page:



Billing Page:



Database:



* + 1. **CONCLUSION**
* Thus, we created a Billing System that create bills and store them in database for later use.
* We created this application using python and php database.
* Thus it can be concluded that efficient system is must and we attempted to create such efficient system and serve the community.
  + 1. **FUTURE WORK**
* In future, we would be adding online payment options such as upi, credit card, debit card, net banking.
* More products will be added in groceries section.
* Discount on online payment will be given as an option.
* We will try to make a friendly, easy to use application.

**References**

**Website**

* https://1library.net/document/zlm1n56y-survey-technologies-used-billing-supermarkets.html
* https://www.academia.edu/34485398/Tips\_For\_Simplified\_Invoice\_Management\_for\_businesses

**Book,**

Python GUI programming with Tkinter by ALLEN D MOORE