PYTHON MINI PROJECT

Advanced Programming Lab [ICT 3166]

PROJECT TITLE: BILL GENERATION SYSTEM

Project done by: Group 7

Hemang Raj	200953240
Priyanshu Pranay	200953170
Gadiraju Lalithadithya	200953130

Project Submission Date: 15/11/2022

Table of Contents

Project Abstract

This mini project aims to generate a bill based on the items bought by the customer. Customer details, item details and other financial details are printed on the bill. This project is developed in PYTHON using concepts such as file-handling, string-handling. Front-end is developed with *tkinter* which is an in-built GUI in Python. Back-end is supported by python itself.

Project Introduction

User gets to select from a variety of electronic products of different brands, with desired quantity. Each range of product has a different tax based on its category. After adding items to the cart total is calculated by summing the gross amount with tax amount. The generated bill has a unique number which is used to distinguish the bill. The bill is saved in the system and can be printed if desired. Any bill which was generated earlier can be searched with the unique bill number. If the user is no longer interested in shopping or wants to reset the billing information, same can be done with clear button. Exit button closes the window.

Project Background

Front-end of the project is implemented using tkinter. Different frames are created to store customer information, product information, billing area. Items are added with the help of combo-box. Customer information is fed into the text field and later stored. As the items are added to the cart, gross amount increases as per the item rate. When all the items are added Generate Bill button sums up the gross amount with tax amount and displays in the billing area along with customer information. Bill can be saved for future reference or searching by using save button. Clear button clears the entire billing area.

Print button shows the bill in printing format.

Project Methodology

Project is created using GUI of python. All the images are added using respective python commands. Items are added in a drop-down manner using combo-box. A specific price and tax amount is associated with each product. So, when a product is added to cart its amount is also added. Quantity of item can be selected by the user. Generate bill button generates the final bill with all the customer details, bill number, items, quantity and total amount. Bill can be saved and searched later.

Project Implementation

```
import random
import os
import tempfile
from time import strftime
from tkinter import *
from tkinter import ttk
from PIL import Image, ImageTk
from tkinter import messagebox


class Billing_App():

    def __init__(self, root):
        self.root = root
        self.root.geometry("1530x830")
        self.root.title("Billing Software")

        self.c_name = StringVar()
        self.c_phone = StringVar()
        self.bill_no = StringVar()
        z = random.randint(1000, 9999)
        self.bill_no.set(z)
        self.c_email = StringVar()
        self.search_bill = StringVar()
        self.product = StringVar()
```

```
self.sub_total = StringVar()
self.tax_input = StringVar()
self.total = StringVar()
self.prices = IntVar()
self.qty = IntVar()

# Product Categories List
self.Category = [
   "Select option", "Mobiles & Accessories", "TV & Appliances", "Laptops & Accessories"]
self.SubCatMobile = ["Select option",
                     "Mobiles", "Earphones", "Cover", "Power Banks", "Speakers"]
self.mobiles = ["Select option", "Apple",
                "Samsung", "Oppo", "Mi", "Jio"]
self.price_mapple = 100000
self.price_samsung = 90000
self.price_oppo = 20000
self.price_mi = 25000
self.price_jio = 5000
self.earphones = ["Select option", "Boat",
                  "Boult", "Mivi", "Skull Candy", "Noise"]
self.price_boat = 1000
self.price_boult = 1200
self.price_mivi = 500
self.price_sc = 3000
self.price_noise = 2000
self.cover = ["Select option", "Avengers",
              "DC", "Transparent", "Blue", "Chequered"]
self.price_avengers = 500
```

```
        self.price_dc = 400
        self.price_trans = 100
        self.price_blue = 200
        self.price_chequ = 150
        self.speakers = ["Select option", "JBL",
                         "Bose", "Zebronics", "Musify", "Krisons"]
        self.price_jbl = 9000
        self.price_bose = 8000
        self.price_zeb = 5000
        self.price_musify = 2000
        self.price_kris = 3000
        self.pbank = ["Select option", "Callmate",
                      "Pomics", "Syska", "Ambrane", "PoMiFi"]
        self.price_callm = 6000
        self.price_pomics = 2000
        self.price_ambrane = 3000
        self.price_syska = 4000
        self.price_pomifi = 5000


        self.SubCatTV = ["Select option", "TV", "Microwave",
                         "Washing Machine", "Water Purifiers", "Trimmers"]
        self.tv = ["Select option", "Sony",
                   "Thompson", "Panasonic", "LG", "Kodak"]
        self.price_sony = 150000
        self.price_thom = 40000
        self.price_pana = 80000
        self.price_lg = 90000
        self.price_kodak = 25000
```

```
self.microwave = ["Select option",
                  "Bajaj", "IFB", "Prestige", "Usha", "Havells"]
self.price_bajaj = 20000
self.price_ifb = 15000
self.price_prestige = 10000
self.price_usha = 12000
self.price_havells = 25000
self.wm = ["Select option", "WhirlPool",
           "LifeLong", "Croma", "Videocon", "Godrej"]
self.price_whirlpool = 15000
self.price_ll = 14000
self.price_croma = 10000
self.price_videocon = 12000
self.price_godrej = 20000
self.trimmers = ["Select option",
                 "Philips", "Nova", "Misfit", "VGR", "Vega"]
self.price_philips = 6000
self.price_nova = 5500
self.price_misfit = 4000
self.price_vgr = 1200
self.price_vega = 6500
self.wp = ["Select option", "Kent", "AquaGuard",
           "Elixir", "BlueStar", "AquaFresh"]
self.price_kent = 9000
self.price_aquaguard = 6000
self.price_bluestar = 7000
self.price_elixir = 8000
self.price_aqfresh = 5000
```

```
self.SubCatLaptop = ["Select option", "Laptop",
                     "Mouse", "Keyboard", "PenDrive", "Printer"]
self.laptop = ["Select option", "Dell", "HP", "Acer", "Asus", "Lenovo"]
self.price_ldell = 90000
self.price_hp = 85000
self.price_acer = 70000
self.price_asus = 100000
self.price_lenovo = 50000
self.mouse = ["Select option", "Abronix",
              "LogiTech", "Portonics", "HyperX", "Razer"]
self.price_abronix = 200
self.price_ltech = 700
self.price_portonics = 400
self.price_hyper = 900
self.price_razer = 1200
self.keyboard = ["Select option", "Wipro",
                 "Corsair", "Zenith", "Gamdias", "Keychron"]
self.price_wipro = 1000
self.price_corsair = 1200
self.price_zenith = 1100
self.price_gamidas = 1500
self.price_keychron = 1700
self.pendrive = ["Select option",
                 "Toshiba", "SanDisk", "Strontium", "E-Rays", "Transcend"]
self.price_toshiba = 1500
self.price_sandisk = 1000
self.price_strontium = 800
```

```
self.price_eray = 500
self.price_transcend = 1200
self.printer = ["Select option", "Canon",
                "Epson", "AES", "Pixma", "DeskJet"]
self.price_canon = 11000
self.price_epson = 10000
self.price__aes = 15000
self.price_pixma = 8000
self.price_deskjet = 16000

# image 1
img = Image.open("./Img/image 1.png")
# Resampling.LANCZOS to resize
img = img.resize((500, 130), Image.Resampling.LANCZOS)
self.photoimg = ImageTk.PhotoImage(img)

label_img = Label(self.root, image=self.photoimg)
label_img.place(x=0, y=0, width=500, height=130)

# image 2
img0 = Image.open("./Img/center.jpg")
img0 = img0.resize((500, 130), Image.Resampling.LANCZOS)
self.photoimg0 = ImageTk.PhotoImage(img0)

label_img0 = Label(self.root, image=self.photoimg0)
label_img0.place(x=500, y=0, width=500, height=130)

# image 3
```

```
img1 = Image.open("./Img/image 2.jpg")
img1 = img1.resize((500, 130), Image.Resampling.LANCZOS)
self.photoimg1 = ImageTk.PhotoImage(img1)

label_img1 = Label(self.root, image=self.photoimg1)
label_img1.place(x=1000, y=0, width=500, height=130)

label_title = Label(self.root, text="Electronics Store Billing System", font=(
    "times new roman", 35, "bold",), bg="navy blue", fg="White")
label_title.place(x=0, y=130, width=1500, height=50)

def time():
    string = strftime('%H:%M:%S %p')
    lbl.config(text=string)
    lbl.after(1000, time)

lbl = Label(label_title, font=("times new roman", 16,
                               "bold"), bg="navy blue", fg="white")
lbl.place(x=0, y=(-15), width=120, height=50)
time()

# main frame to contain other frames
Main_Frame = Frame(self.root, bd=5, relief=RIDGE, bg="dark grey")
Main_Frame.place(x=0, y=180, width=1500, height=600)

# customer frame
cust_frame = LabelFrame(Main_Frame, text="CUSTOMER INFORMATION", font=(
    "arial", 12, "bold",), bg="light green", fg="Black")
```

```
cust_frame.place(x=10, y=5, width=350, height=140)

# name
self.lbl1_name = Label(cust_frame, text="Name", font=(
   "arial", 12, "bold",), bg="light green")
self.lbl1_name.grid(row=0, column=0, stick=W, padx=5, pady=2)

self.entry_name = ttk.Entry(
   cust_frame, textvariable=self.c_name, font=("arial", 12, "bold",), width=20,)
self.entry_name.grid(row=0, column=1)

# mobile
self.lbl1_mobile = Label(cust_frame, text="Mobile No.", font=(
   "arial", 12, "bold",), bg="light green",)
self.lbl1_mobile.grid(row=1, column=0, stick=W, padx=5, pady=2)

self.entry_mobile = ttk.Entry(
   cust_frame, textvariable=self.c_phone, font=("arial", 12, "bold",), width=20)
self.entry_mobile.grid(row=1, column=1)

# email
self.lbl1_email = Label(cust_frame, text="E-mail",
                  font=("arial", 12, "bold",), bg="light green",)
self.lbl1_email.grid(row=2, column=0, stick=W, padx=5, pady=2)

self.entry_email = ttk.Entry(
   cust_frame, textvariable=self.c_email, font=("arial", 12, "bold",), width=20)
self.entry_email.grid(row=2, column=1)
```

```
# product frame
product_frame = LabelFrame(Main_Frame, text="PRODUCT INFORMATION", font=(
    "arial", 12, "bold",), bg="mintcream", fg="Black")
product_frame.place(x=370, y=5, width=600, height=140)

# Category
self.lbl1_category = Label(product_frame, text="Category", font=(
    "arial", 12, "bold",), bg="mintcream",)
self.lbl1_category.grid(row=0, column=0, stick=W, padx=5, pady=2)

self.Combo_category = ttk.Combobox(product_frame, value=self.Category, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.Combo_category.current(0)
self.Combo_category.grid(row=0, column=1, sticky=W, padx=5, pady=2)
self.Combo_category.bind("<<ComboboxSelected>>", self.Categories)

# Sub Category
self.lbl1_subcategory = Label(product_frame, text="Subcategory", font=(
    "arial", 12, "bold",), bg="mintcream",)
self.lbl1_subcategory.grid(row=1, column=0, stick=W, padx=5, pady=2)

self.Combo_subcategory = ttk.Combobox(product_frame, value=[""], font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.Combo_subcategory.grid(row=1, column=1, sticky=W, padx=5, pady=2)
self.Combo_subcategory.bind("<<ComboboxSelected>>", self.Product_add)

# Product Name
```

```
self.lbl1_pcategory = Label(product_frame, text="Product Name", font=(
    "arial", 12, "bold",), bg="mintcream",)
self.lbl1_pcategory.grid(row=2, column=0, stick=W, padx=5, pady=2)

self.Combo_pcategory = ttk.Combobox(product_frame, textvariable=self.product, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.Combo_pcategory.grid(row=2, column=1, sticky=W, padx=5, pady=2)
self.Combo_pcategory.bind("<<ComboboxSelected>>", self.price)

# Price
self.lbl1_price = Label(product_frame, text="Price", font=(
    "arial", 12, "bold",), bg="mintcream",)
self.lbl1_price.grid(row=0, column=2, stick=W, padx=5, pady=2)

self.Combo_price = ttk.Combobox(product_frame, textvariable=self.prices, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.Combo_price.grid(row=0, column=3, sticky=W, padx=5, pady=2)

# Quantity
self.lbl1_qty = Label(product_frame, text="Quantity", font=(
    "arial", 12, "bold",), bg="mintcream")
self.lbl1_qty.grid(row=1, column=2, stick=W, padx=5, pady=2)

self.Combo_qty = ttk.Combobox(product_frame, textvariable=self.qty, font=(
    "arial", 10, "bold"), width=20)
self.Combo_qty.grid(row=1, column=3, sticky=W, padx=5, pady=2)

# middleframe
```

```
middle_frame = Frame(self.root, bd=5)
middle_frame.place(x=10, y=340, width=970, height=300)

# image 1 in middle
imgs = Image.open("./Img/i4.jpg")
# Resampling.LANCZOS to resize
imgs = imgs.resize((470, 250), Image.Resampling.LANCZOS)
self.photoimgd = ImageTk.PhotoImage(imgs)

label_imge = Label(self.root, image=self.photoimgd)
label_imge.place(x=14, y=360, width=470, height=250)

# image 2 in middle
img01 = Image.open("Img/i5.jpg")
img01 = img01.resize((470, 250), Image.Resampling.LANCZOS)
self.photoimg01 = ImageTk.PhotoImage(img01)

label_img01 = Label(self.root, image=self.photoimg01)
label_img01.place(x=502, y=360, width=470, height=250)

# searchbar
search_frame = Frame(Main_Frame, bd=2, bg="pink")
search_frame.place(x=1000, y=5, width=480, height=35)

self.lbill = Label(search_frame, font=(
    'calibri', 15, 'bold'), fg="black", bg="pink", text="Bill Number")
self.lbill.grid(row=0, column=0, sticky=W, padx=1, pady=2)
```

```
self.enty_subt12 = ttk.Entry(
    search_frame, textvariable=self.search_bill, font=("arial", 14, "bold"), width=20)
self.enty_subt12.grid(row=0, column=1, sticky=W, padx=30)

self.btnadd12 = Button(search_frame, command=self.find_bill, text="Search", font=(
    "arial", 12, "bold"), bg="blue", fg="white", cursor="target")
self.btnadd12.grid(row=0, column=4)

# billing area
rightlabelframe = LabelFrame(Main_Frame, text="Bill", font=(
    "arial", 12, "bold",), bg="pink", fg="black")
rightlabelframe.place(x=1000, y=45, width=480, height=410)

# scrollbar
scroll_y = Scrollbar(rightlabelframe, orient=VERTICAL)
self.textarea = Text(rightlabelframe, yscrollcommand=scroll_y.set,
                     bg="white", fg="blue", font=("times new roman", 12, "bold"))
scroll_y.pack(side=RIGHT, fill=Y)
scroll_y.config(command=self.textarea.yview)
self.textarea.pack(fill=BOTH, expand=1)

# total bill
bill_frame = LabelFrame(Main_Frame, text="BILL TOTAL", font=(
    "arial", 14, "bold",), bg="gold", fg="black")
bill_frame.place(x=0, y=460, width=1485, height=120)

# subtotal
self.lbl1_subt = Label(bill_frame, text="Sub Total", font=(
```

```
    "arial", 12, "bold",), bg="gold",)
self.lbl1_subt.grid(row=0, column=0, stick=W, padx=5, pady=2)

self.enty_subt = ttk.Entry(bill_frame, textvariable=self.sub_total, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.enty_subt.grid(row=0, column=1, sticky=W, padx=5, pady=2)

# tax
self.lbl1_subt1 = Label(bill_frame, text="TAX", font=(
    "arial", 12, "bold",), bg="gold",)
self.lbl1_subt1.grid(row=1, column=0, stick=W, padx=5, pady=2)

self.enty_subt1 = ttk.Entry(bill_frame, textvariable=self.tax_input, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.enty_subt1.grid(row=1, column=1, sticky=W, padx=5, pady=2)

# amount
self.lbl1_subt2 = Label(bill_frame, text="Total Amount", font=(
    "arial", 12, "bold",), bg="gold",)
self.lbl1_subt2.grid(row=2, column=0, stick=W, padx=5, pady=2)

self.enty_subt2 = ttk.Entry(bill_frame, textvariable=self.total, font=(
    "arial", 10, "bold"), width=20, state="readonly")
self.enty_subt2.grid(row=2, column=1, sticky=W, padx=5, pady=2)

# buttons
btn_frame = Frame(bill_frame, bd=2, bg="gold")
btn_frame.place(x=320, y=0)
```

```
        self.btnadd = Button(btn_frame, command=self.AddToCart, text="Add to Cart",
width=20, height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd.grid(row=0, column=0)

        self.btnadd1 = Button(btn_frame, command=self.GenBill, text="Generate Bill",
width=20, height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd1.grid(row=0, column=1)

        self.btnadd2 = Button(btn_frame, command=self.SaveBill, text="Save Bill", width=20,
height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd2.grid(row=0, column=2)

        self.btnadd3 = Button(btn_frame, command=self.iprint, text="Print", width=20,
height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd3.grid(row=0, column=3)

        self.btnadd4 = Button(btn_frame, command=self.clear, text="Clear", width=20,
height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd4.grid(row=0, column=4)

        self.btnadd5 = Button(btn_frame, command=self.root.destroy, text="Exit", width=20,
height=2, font=(
            "arial", 10, "bold"), bg="blue", fg="white", cursor="pirate")
        self.btnadd5.grid(row=0, column=5)
```

```
    self.welcome()


    self.l = []


def AddToCart(self):
    Tax = 1
    self.n = self.prices.get()
    self.m = self.qty.get()*self.n
    self.l.append(self.m)
    if self.product.get() == "":
        messagebox.showerror("ERROR", "Please Select a Product")
    else:
        self.textarea.insert(
            END, f"\n {self.product.get()}\t\t{self.qty.get()}\t\t{self.m}")
        self.sub_total.set(str('₹ %.2f' % (sum(self.l))))
        self.tax_input.set(
            str('₹ %.2f' % ((((sum(self.l))-(self.prices.get())*Tax)/100))))
        self.total.set(
            str('₹ %.2f' %
                (((sum(self.l))+((((sum(self.l))-(self.prices.get()))*Tax)/100)))))


def GenBill(self):
    if self.product.get() == "":
        messagebox.showerror("ERROR", "Please  Add to Cart a Product")
    else:
        text = self.textarea.get(10.0, (10.0+float(len(self.l))))
        self.welcome()
        self.textarea.insert(END, text)
```

```
        self.textarea.insert(
            END, f"\n\n ================================================")
        self.textarea.insert(
            END, f"\n Sub Amount: \t\t\t{self.sub_total.get()}")
        self.textarea.insert(
            END, f"\n Tax Amount: \t\t\t{self.tax_input.get()}")
        self.textarea.insert(
            END, f"\n Total Amount: \t\t\t{self.total.get()}")
        self.textarea.insert(
            END, f"\n ================================================")

    def SaveBill(self):
        op = messagebox.askyesno("Save Bill", "Do you want to save the Bill ?")
        if op > 0:
            self.bill_data = self.textarea.get(1.0, END)
            f1 = open("bills/"+str(self.bill_no.get()) +
                      ".txt", 'w', encoding="utf-8")
            f1.writelines(self.bill_data)
            op = messagebox.showinfo(
                "Saved", f"Bill No:{self.bill_no.get()} Saved Successfully")
            f1.close()

    def iprint(self):
        q = self.textarea.get(1.0, "end-1c")
        filename = tempfile.mktemp('.txt')
        open(filename, 'w', encoding="utf-8").writelines(q)
        os.startfile(filename, "print")
```

```
def find_bill(self):
    found = "no"
    for i in os.listdir("bills/"):
        if i.split('.')[0] == self.search_bill.get():
            f1 = open(f'bills/{i}', 'r', encoding="utf-8")
            self.textarea.delete(1.0, END)
            for d in f1:
                self.textarea.insert(END, d)
            f1.close()
            found = "yes"
    if found == 'no':
        messagebox.showerror("Error", "Invalid Bill")


def clear(self):
    self.textarea.delete(1.0, END)
    self.c_name.set("")
    self.c_email.set("")
    self.c_phone.set("")
    x = random.randint(1000, 9999)
    self.bill_no.set(str(x))
    self.search_bill.set("")
    self.product.set("")
    self.prices.set(0)
    self.qty.set(0)
    self.l = [0]
    self.total.set("")
    self.sub_total.set("")
    self.tax_input.set('')
```

```
    self.welcome()

def welcome(self):
    self.textarea.delete(1.0, END)
    self.textarea.insert(END, "\t\t WELCOME TO PROJECT MALL")
    self.textarea.insert(END, f"\n Bill Number:{self.bill_no.get()}")
    self.textarea.insert(END, f"\n Customer Name:{self.c_name.get()}")
    self.textarea.insert(END, f"\n Phone Number:{self.c_phone.get()}")
    self.textarea.insert(END, f"\n Customer E-Mail:{self.c_email.get()}")
    self.textarea.insert(
        END, f"\n==========================================")
    self.textarea.insert(END, f"\n Products\t\tQTY\t\tPrice")
    self.textarea.insert(
        END, f"\n==========================================\n")

def Categories(self, event=""):
    if self.Combo_category.get() == "Mobiles & Accessories":
        self.Combo_subcategory.config(value=self.SubCatMobile)
        self.Combo_subcategory.current(0)
    if self.Combo_category.get() == "TV & Appliances":
        self.Combo_subcategory.config(value=self.SubCatTV)
        self.Combo_subcategory.current(0)
    if self.Combo_category.get() == "Laptops & Accessories":
        self.Combo_subcategory.config(value=self.SubCatLaptop)
        self.Combo_subcategory.current(0)

def Product_add(self, event=""):
    if self.Combo_subcategory.get() == "Mobiles":
```

```
        self.Combo_pcategory.config(value=self.mobiles)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Earphones":
        self.Combo_pcategory.config(value=self.earphones)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Cover":
        self.Combo_pcategory.config(value=self.cover)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Speakers":
        self.Combo_pcategory.config(value=self.speakers)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Power Banks":
        self.Combo_pcategory.config(value=self.pbank)
        self.Combo_pcategory.current(0)

    if self.Combo_subcategory.get() == "TV":
        self.Combo_pcategory.config(value=self.tv)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Microwave":
        self.Combo_pcategory.config(value=self.microwave)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Washing Machine":
        self.Combo_pcategory.config(value=self.wm)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Trimmers":
        self.Combo_pcategory.config(value=self.trimmers)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Water Purifiers":
```

```
        self.Combo_pcategory.config(value=self.wp)
        self.Combo_pcategory.current(0)

    if self.Combo_subcategory.get() == "Laptop":
        self.Combo_pcategory.config(value=self.laptop)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Mouse":
        self.Combo_pcategory.config(value=self.mouse)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Keyboard":
        self.Combo_pcategory.config(value=self.keyboard)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "PenDrive":
        self.Combo_pcategory.config(value=self.pendrive)
        self.Combo_pcategory.current(0)
    if self.Combo_subcategory.get() == "Printer":
        self.Combo_pcategory.config(value=self.printer)
        self.Combo_pcategory.current(0)

def price(self, event=""):
    if self.Combo_pcategory.get() == "Apple":
        self.Combo_price.config(value=self.price_mapple)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Samsung":
        self.Combo_price.config(value=self.price_samsung)
        self.Combo_price.current(0)
        self.qty.set(1)
```

```
    if self.Combo_pcategory.get() == "Oppo":
        self.Combo_price.config(value=self.price_oppo)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Mi":
        self.Combo_price.config(value=self.price_mi)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Jio":
        self.Combo_price.config(value=self.price_jio)
        self.Combo_price.current(0)
        self.qty.set(1)


    if self.Combo_pcategory.get() == "Boat":
        self.Combo_price.config(value=self.price_boat)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Mivi":
        self.Combo_price.config(value=self.price_mivi)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Skull Candy":
        self.Combo_price.config(value=self.price_sc)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Boult":
        self.Combo_price.config(value=self.price_boult)
        self.Combo_price.current(0)
```

```
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Noise":
        self.Combo_price.config(value=self.price_noise)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Bajaj":
        self.Combo_price.config(value=self.price_bajaj)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "IFB":
        self.Combo_price.config(value=self.price_ifb)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Usha":
        self.Combo_price.config(value=self.price_usha)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Prestige":
        self.Combo_price.config(value=self.price_prestige)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Havells":
        self.Combo_price.config(value=self.price_havells)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "WhirlPool":
```

```
        self.Combo_price.config(value=self.price_whirlpool)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "LifeLong":
        self.Combo_price.config(value=self.price_ll)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Videocon":
        self.Combo_price.config(value=self.price_videocon)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Godrej":
        self.Combo_price.config(value=self.price_godrej)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Croma":
        self.Combo_price.config(value=self.price_croma)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Avengers":
        self.Combo_price.config(value=self.price_avengers)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "DC":
        self.Combo_price.config(value=self.price_dc)
        self.Combo_price.current(0)
        self.qty.set(1)
```

```
    if self.Combo_pcategory.get() == "Transparent":
        self.Combo_price.config(value=self.price_trans)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Blue":
        self.Combo_price.config(value=self.price_blue)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Chequered":
        self.Combo_price.config(value=self.price_chequ)
        self.Combo_price.current(0)
        self.qty.set(1)


    if self.Combo_pcategory.get() == "Wipro":
        self.Combo_price.config(value=self.price_wipro)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Corsair":
        self.Combo_price.config(value=self.price_corsair)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Zenith":
        self.Combo_price.config(value=self.price_zenith)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Gamdias":
        self.Combo_price.config(value=self.price_gamidas)
        self.Combo_price.current(0)
```

```
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Keychron":
        self.Combo_price.config(value=self.price_keychron)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Abronix":
        self.Combo_price.config(value=self.price_abronix)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "LogiTech":
        self.Combo_price.config(value=self.price_ltech)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Portonics":
        self.Combo_price.config(value=self.price_portonics)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "HyperX":
        self.Combo_price.config(value=self.price_hyper)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Razer":
        self.Combo_price.config(value=self.price_razer)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Sony":
```

```
        self.Combo_price.config(value=self.price_sony)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Kodak":
        self.Combo_price.config(value=self.price_kodak)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Thompson":
        self.Combo_price.config(value=self.price_thom)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Panasonic":
        self.Combo_price.config(value=self.price_pana)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "LG":
        self.Combo_price.config(value=self.price_lg)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Dell":
        self.Combo_price.config(value=self.price_ldell)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "HP":
        self.Combo_price.config(value=self.price_hp)
        self.Combo_price.current(0)
        self.qty.set(1)
```

```
    if self.Combo_pcategory.get() == "Acer":
        self.Combo_price.config(value=self.price_acer)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Asus":
        self.Combo_price.config(value=self.price_asus)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Lenovo":
        self.Combo_price.config(value=self.price_lenovo)
        self.Combo_price.current(0)
        self.qty.set(1)


    if self.Combo_pcategory.get() == "Philips":
        self.Combo_price.config(value=self.price_philips)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Nova":
        self.Combo_price.config(value=self.price_nova)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Misfit":
        self.Combo_price.config(value=self.price_misfit)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "VGR":
        self.Combo_price.config(value=self.price_vgr)
        self.Combo_price.current(0)
```

```
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Vega":
        self.Combo_price.config(value=self.price_vega)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Kent":
        self.Combo_price.config(value=self.price_kent)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "AquaGuard":
        self.Combo_price.config(value=self.price_aquaguard)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Elixir":
        self.Combo_price.config(value=self.price_elixir)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "BlueStar":
        self.Combo_price.config(value=self.price_bluestar)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "AquaFresh":
        self.Combo_price.config(value=self.price_aqfresh)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "JBL":
```

```
        self.Combo_price.config(value=self.price_jbl)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Bose":
        self.Combo_price.config(value=self.price_bose)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Zebronics":
        self.Combo_price.config(value=self.price_zeb)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Musify":
        self.Combo_price.config(value=self.price_musify)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Krisons":
        self.Combo_price.config(value=self.price_kris)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Callmate":
        self.Combo_price.config(value=self.price_callm)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Pomics":
        self.Combo_price.config(value=self.price_pomics)
        self.Combo_price.current(0)
        self.qty.set(1)
```

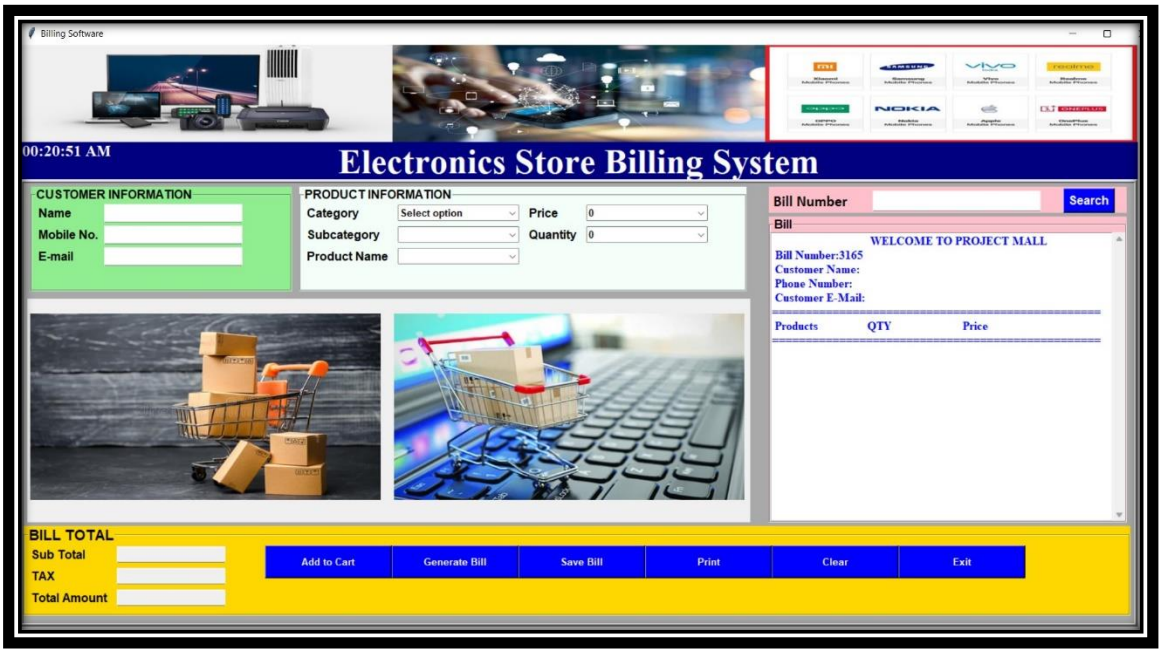
```
    if self.Combo_pcategory.get() == "Syska":
        self.Combo_price.config(value=self.price_syska)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Ambrane":
        self.Combo_price.config(value=self.price_ambrane)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "PoMiFi":
        self.Combo_price.config(value=self.price_pomifi)
        self.Combo_price.current(0)
        self.qty.set(1)


    if self.Combo_pcategory.get() == "Toshiba":
        self.Combo_price.config(value=self.price_toshiba)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "SanDisk":
        self.Combo_price.config(value=self.price_sandisk)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Strontium":
        self.Combo_price.config(value=self.price_strontium)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "E-Rays":
        self.Combo_price.config(value=self.price_eray)
        self.Combo_price.current(0)
```

```
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Transcend":
        self.Combo_price.config(value=self.price_transcend)
        self.Combo_price.current(0)
        self.qty.set(1)

    if self.Combo_pcategory.get() == "Canon":
        self.Combo_price.config(value=self.price_canon)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Epson":
        self.Combo_price.config(value=self.price_epson)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "AES":
        self.Combo_price.config(value=self.price_aes)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "Pixma":
        self.Combo_price.config(value=self.price_pixma)
        self.Combo_price.current(0)
        self.qty.set(1)
    if self.Combo_pcategory.get() == "DeskJet":
        self.Combo_price.config(value=self.price_deskjet)
        self.Combo_price.current(0)
        self.qty.set(1)
```

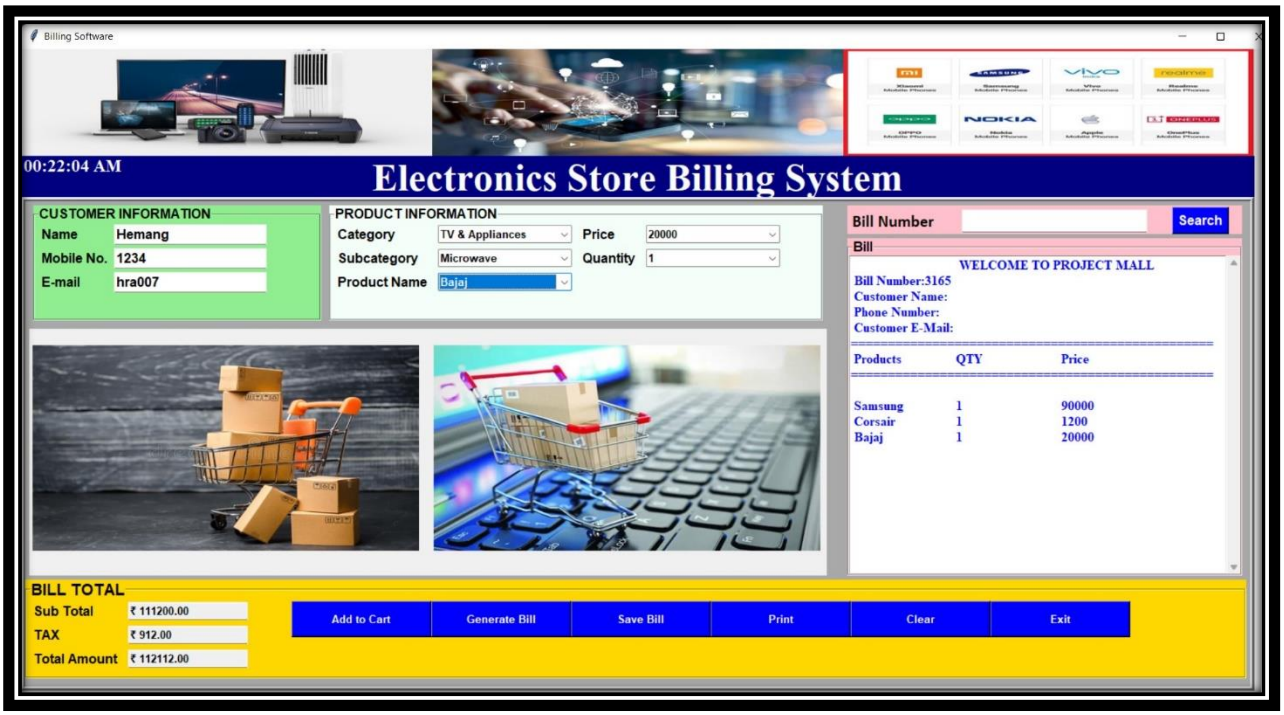
```
if __name__ == '__main__':
    root = Tk()
    obj = Billing_App(root)
    root.mainloop()
```

Results

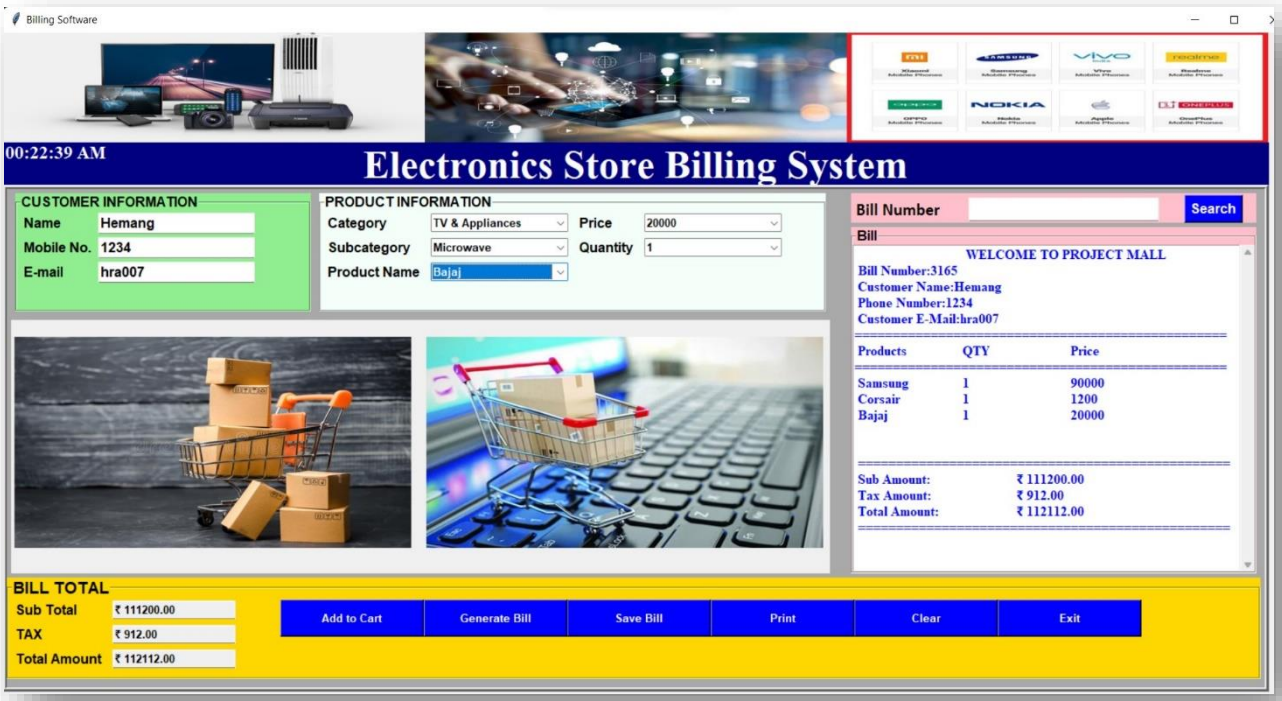
Before entering details:



After entering details:



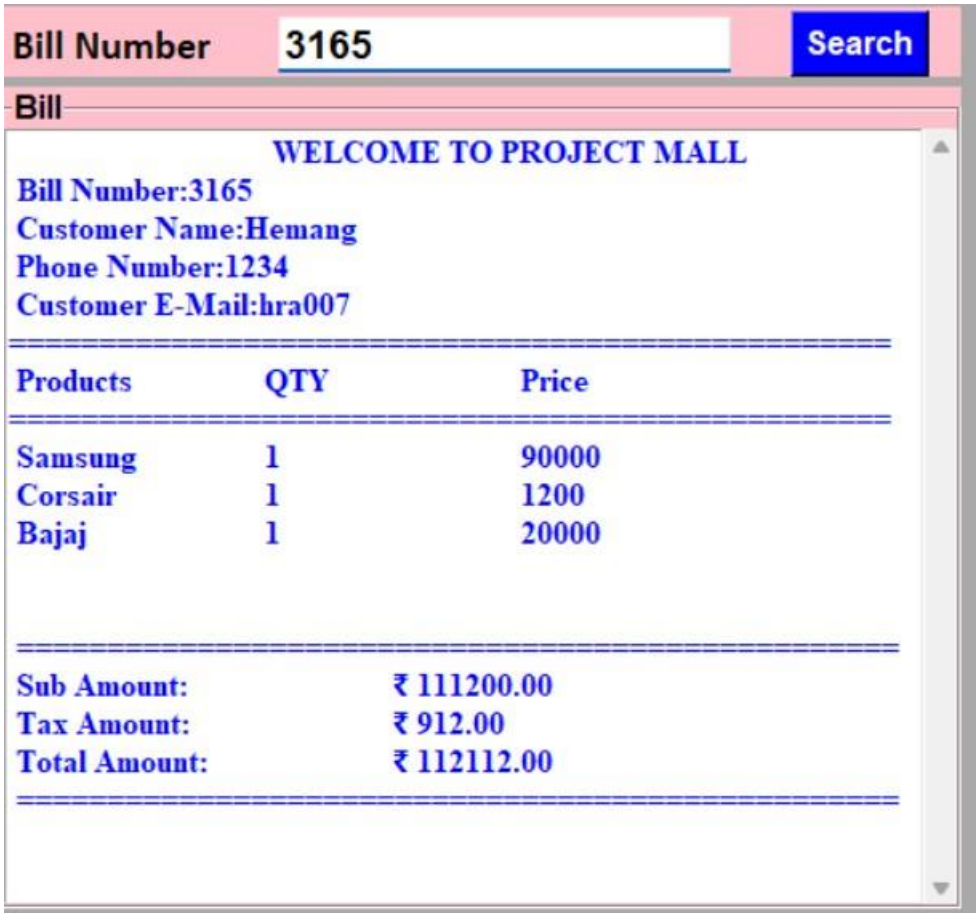
After generating bill:



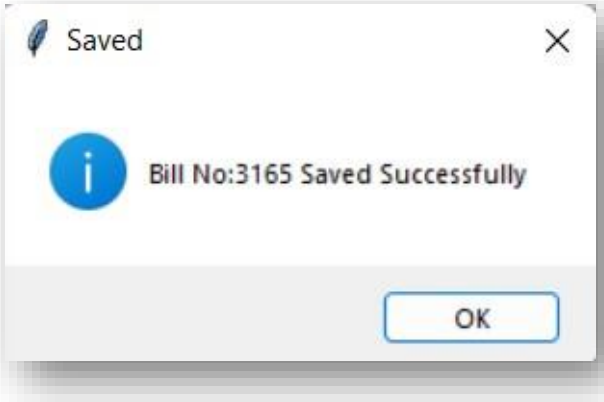
Printing The Bill

```
                    WELCOME TO PROJECT MALL
 Bill Number:3165
 Customer Name:Hemang
 Phone Number:1234
 Customer E-Mail:hra007
=============================================
===
 Products                  QTY                 Price
=============================================
===
 Samsung                   1                   90000
 Corsair                   1                   1200
 Bajaj          1                    20000



=============================================
===
 Sub Amount:                                   ₹ 111200.00
 Tax Amount:                                   ₹ 912.00
 Total Amount:                                 ₹ 112112.00

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

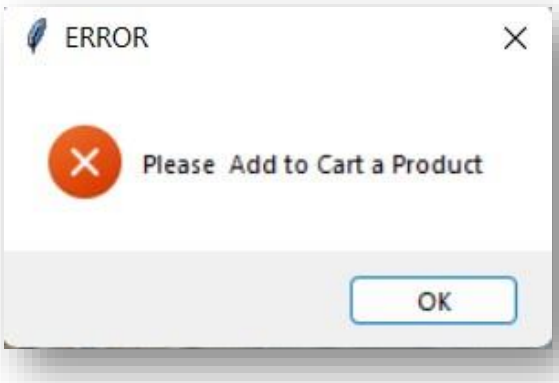
Searching for a bill:



Saving a bill:



Other Dialogue boxes:



Conclusions

The mini-project is developed successfully as per the specified conditions. The combination of front-end and back-end can be used to solve many such problems. The interface can be changed or modified as per the needs.

References

https://www.geeksforgeeks.org/

https://www.youtube.com

https://docs.python.org/

https://www.w3schools.com/python/

https://www.tutorialspoint.com/python/python_gui_programming.htm

https://www.javatpoint.com/python-tkinter