PROJECT NAME

" BILLING SOFTWARE OF KARYANA SHOP ;

- CREATOR
  Saad khan
- Date
  24/2/24

- INTRODUCTION
  Purpose : To automate the billing process for a store.

- Functionality : The billing software works by manging a database of items, calculating totals, applying discounts, and generating receipts.

- How to run :
  1. Double-click the software icon to launch it.
  2. User interfaxe overview : The software opens to a mainscreen with option for adding items, creating bills, and generating reports.
  3. Adding items : Click on the " add items button and enter the item details such as name , price and quantity and save it if you want.
  4. Creating a Bills : Click on the " create bill " button. Scan or select items from the database and add them to the bill and the software will automatically calculate the total amount.
  5. Exiting the software : click on the " exit button " or close the window to exit the software

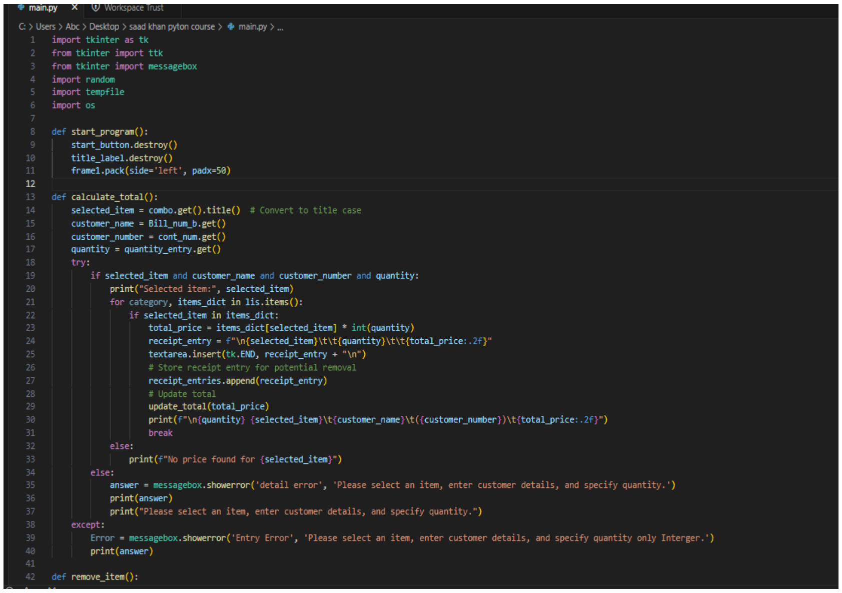
- Description :
  Our billing software is designed to streamline the checkout process for retail stores, specifically tailored for establishments that sell a variety of items including fruits and groceries. The software provides a user-friendly interface that allows cashiers to quickly and accurately generate bills for customers, ensuring a smooth and efficient shopping experience.
- Importance :
  Efficiency: Billing software speeds up the checkout process, reducing waiting times for customers and improving overall efficiency.

  Accuracy: Automated calculations reduce the risk of human errors in billing, ensuring that customers are charged correctly.

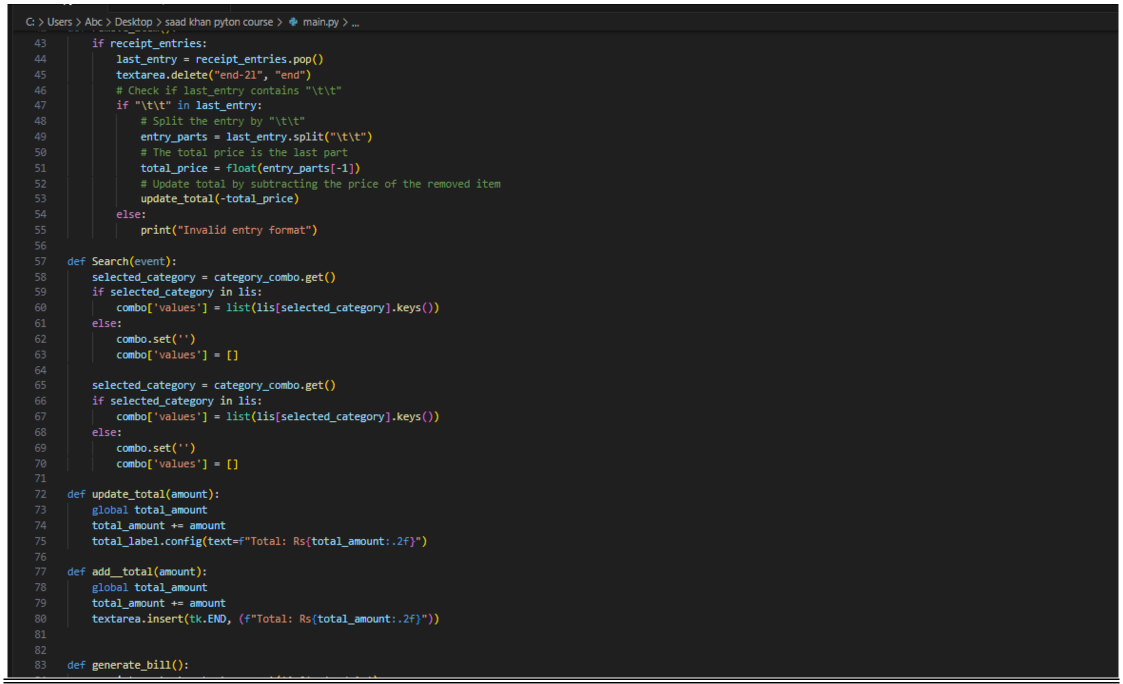
  Inventory Management: The software helps track inventory levels, allowing store owners to manage stock and avoid overstocking or stockouts.

Customer Service: Quick and accurate billing enhances the customer experience, leading to increased satisfaction and loyalty.

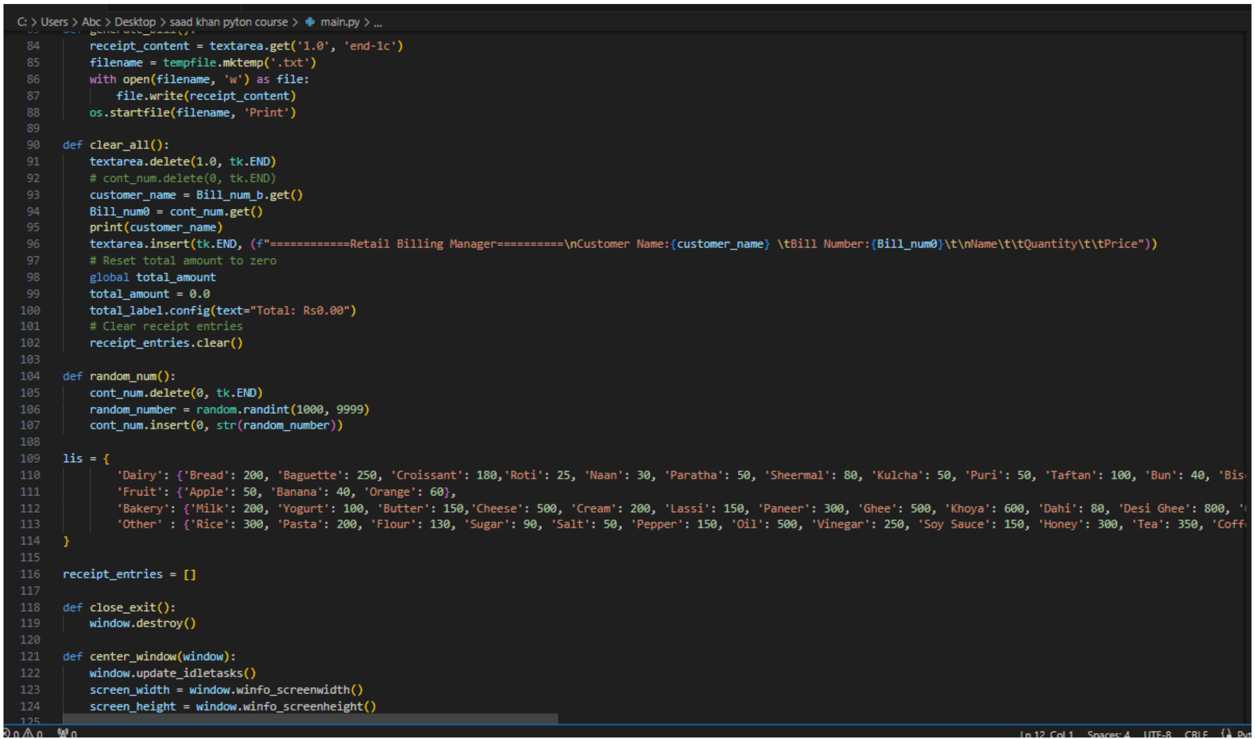
- <u>SCREENSHOTS:</u>


main.py
Workspace Trust
C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...

```python
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
import random
import tempfile
import os

def start_program():
    start_button.destroy()
    title_label.destroy()
    frame1.pack(side='left', padx=50)

def calculate_total():
    selected_item = combo.get().title()  # Convert to title case
    customer_name = Bill_num_b.get()
    customer_number = cont_num.get()
    quantity = quantity_entry.get()
    try:
        if selected_item and customer_name and customer_number and quantity:
            print("Selected item:", selected_item)
            for category, items_dict in lis.items():
                if selected_item in items_dict:
                    total_price = items_dict[selected_item] * int(quantity)
                    receipt_entry = f"\n{selected_item}\t\t{quantity}\t\t{total_price:.2f}"
                    textarea.insert(tk.END, receipt_entry + "\n")
                    # Store receipt entry for potential removal
                    receipt_entries.append(receipt_entry)
                    # Update total
                    update_total(total_price)
                    print(f"\n{quantity} {selected_item}\t{customer_name}\t({customer_number})\t{total_price:.2f}")
                    break
            else:
                print(f"No price found for {selected_item}")
        else:
            answer = messagebox.showerror('detail error', 'Please select an item, enter customer details, and specify quantity.')
            print(answer)
            print("Please select an item, enter customer details, and specify quantity.")
    except:
        Error = messagebox.showerror('Entry Error', 'Please select an item, enter customer details, and specify quantity only Interger.')
        print(answer)

def remove_item():
```

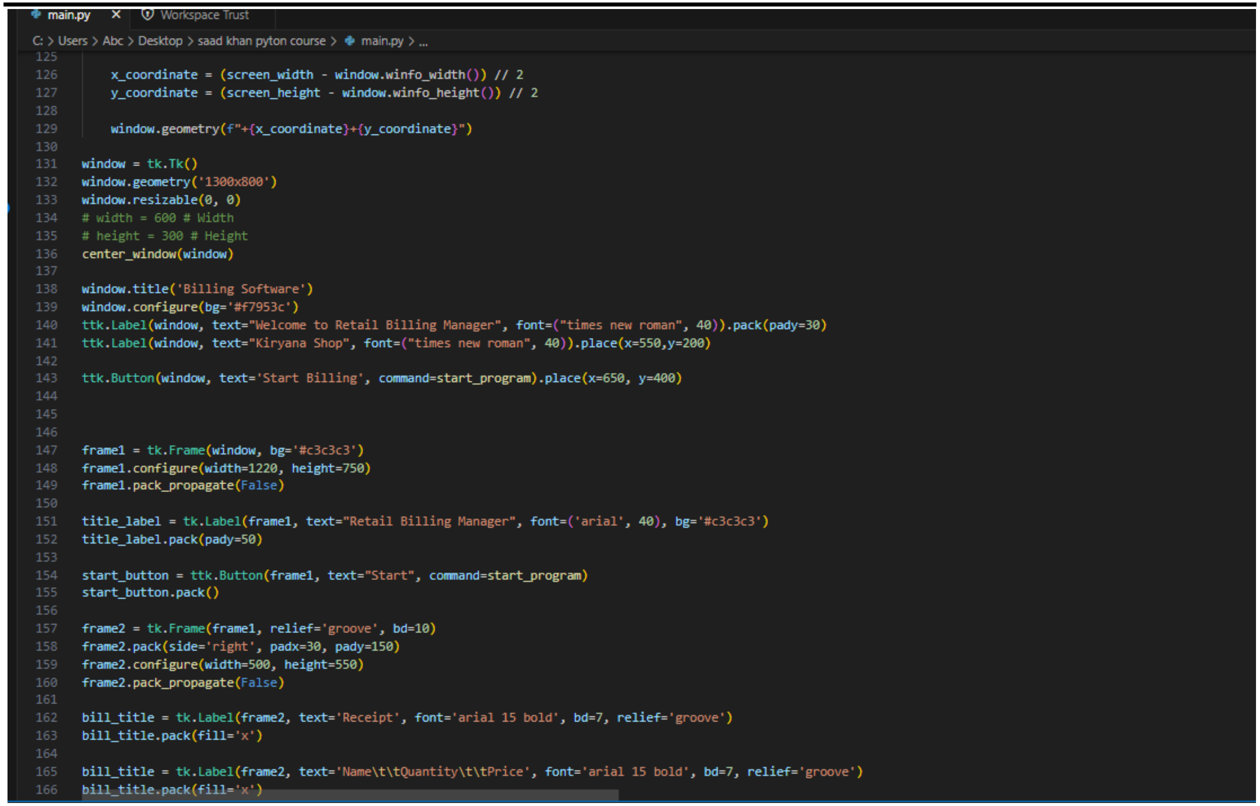

C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...

```python
    if receipt_entries:
        last_entry = receipt_entries.pop()
        textarea.delete("end-2l", "end")
        # Check if last_entry contains "\t\t"
        if "\t\t" in last_entry:
            # Split the entry by "\t\t"
            entry_parts = last_entry.split("\t\t")
            # The total price is the last part
            total_price = float(entry_parts[-1])
            # Update total by subtracting the price of the removed item
            update_total(-total_price)
        else:
            print("Invalid entry format")

def Search(event):
    selected_category = category_combo.get()
    if selected_category in lis:
        combo['values'] = list(lis[selected_category].keys())
    else:
        combo.set('')
        combo['values'] = []

    selected_category = category_combo.get()
    if selected_category in lis:
        combo['values'] = list(lis[selected_category].keys())
    else:
        combo.set('')
        combo['values'] = []

def update_total(amount):
    global total_amount
    total_amount += amount
    total_label.config(text=f"Total: Rs{total_amount:.2f}")

def add__total(amount):
    global total_amount
    total_amount += amount
    textarea.insert(tk.END, (f"Total: Rs{total_amount:.2f}"))


def generate_bill():
```



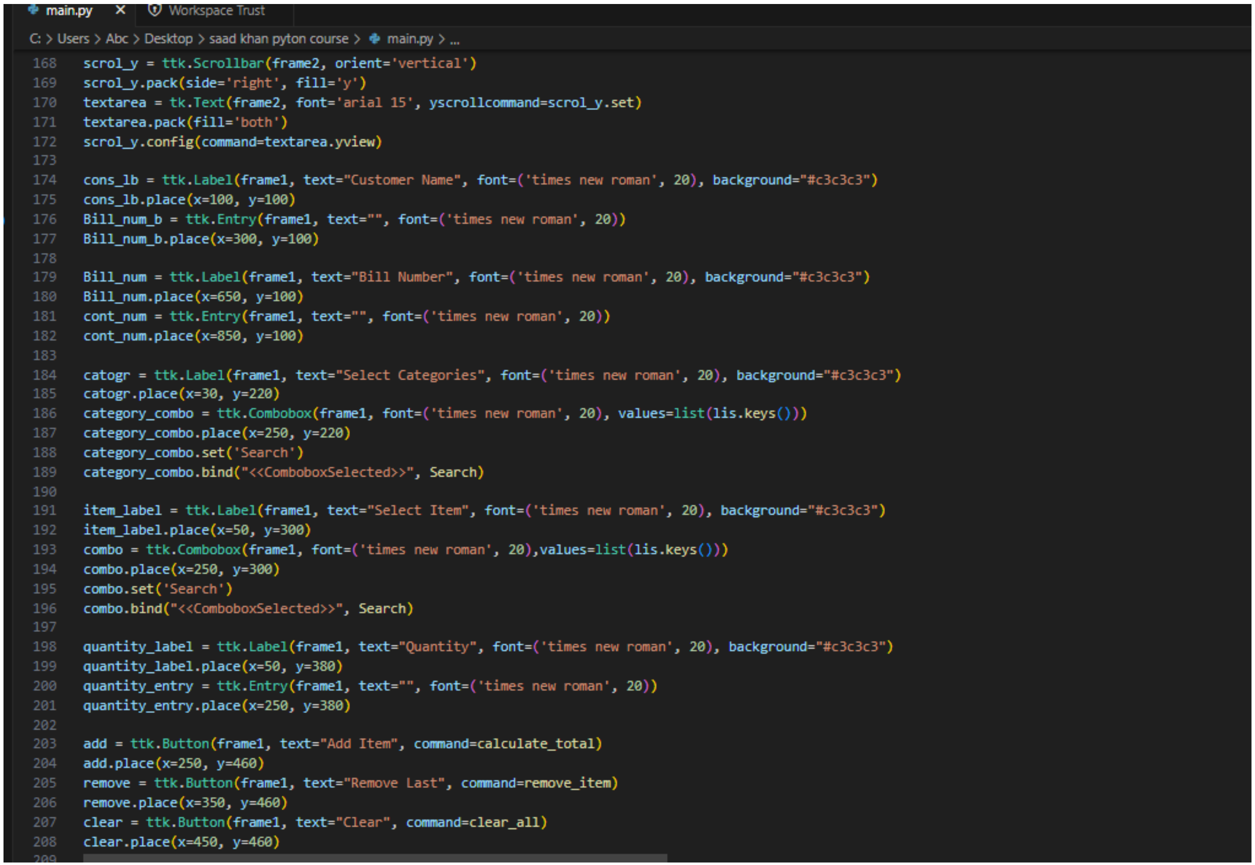
C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...
0 ⚠ 0 0
Ln 12, Col 1
Spaces: 4
UTF-8
CRLF

```python
    receipt_content = textarea.get('1.0', 'end-1c')
    filename = tempfile.mktemp('.txt')
    with open(filename, 'w') as file:
        file.write(receipt_content)
    os.startfile(filename, 'Print')

def clear_all():
    textarea.delete(1.0, tk.END)
    # cont_num.delete(0, tk.END)
    customer_name = Bill_num_b.get()
    Bill_num0 = cont_num.get()
    print(customer_name)
    textarea.insert(tk.END, (f"===========Retail Billing Manager==========\nCustomer Name:{customer_name} \tBill Number:{Bill_num0}\t\nName\t\tQuantity\t\tPrice"))
    # Reset total amount to zero
    global total_amount
    total_amount = 0.0
    total_label.config(text="Total: Rs0.00")
    # Clear receipt entries
    receipt_entries.clear()

def random_num():
    cont_num.delete(0, tk.END)
    random_number = random.randint(1000, 9999)
    cont_num.insert(0, str(random_number))

lis = {
        'Dairy': {'Bread': 200, 'Baguette': 250, 'Croissant': 180,'Roti': 25, 'Naan': 30, 'Paratha': 50, 'Sheermal': 80, 'Kulcha': 50, 'Puri': 50, 'Taftan': 100, 'Bun': 40, 'Bis
        'Fruit': {'Apple': 50, 'Banana': 40, 'Orange': 60},
        'Bakery': {'Milk': 200, 'Yogurt': 100, 'Butter': 150,'Cheese': 500, 'Cream': 200, 'Lassi': 150, 'Paneer': 300, 'Ghee': 500, 'Khoya': 600, 'Dahi': 80, 'Desi Ghee': 800, '
        'Other' : {'Rice': 300, 'Pasta': 200, 'Flour': 130, 'Sugar': 90, 'Salt': 50, 'Pepper': 150, 'Oil': 500, 'Vinegar': 250, 'Soy Sauce': 150, 'Honey': 300, 'Tea': 350, 'Coff
}

receipt_entries = []

def close_exit():
    window.destroy()

def center_window(window):
    window.update_idletasks()
    screen_width = window.winfo_screenwidth()
    screen_height = window.winfo_screenheight()
```

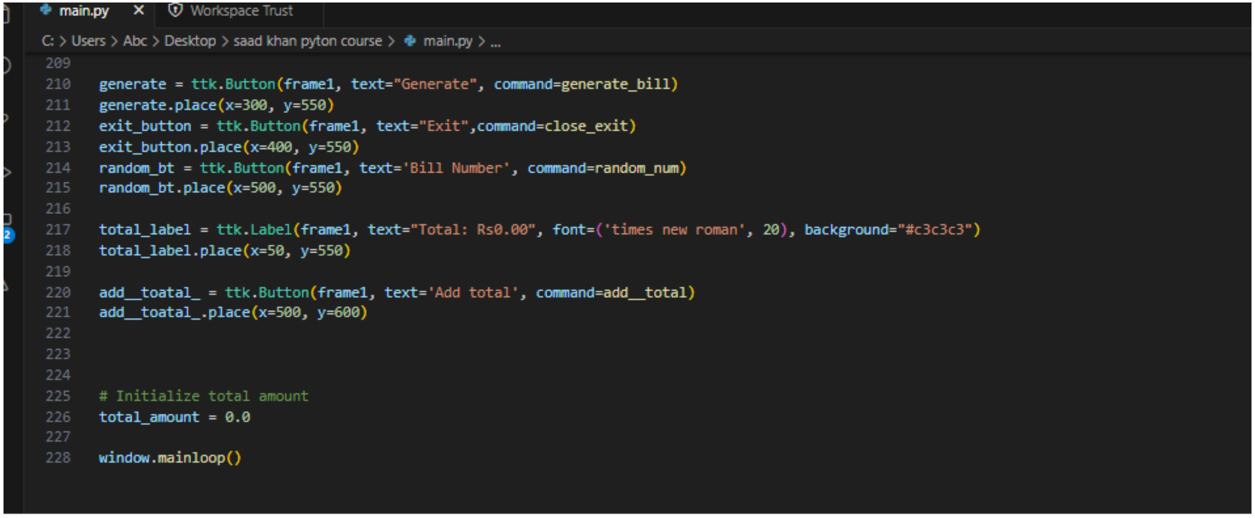


main.py
Workspace Trust
C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...

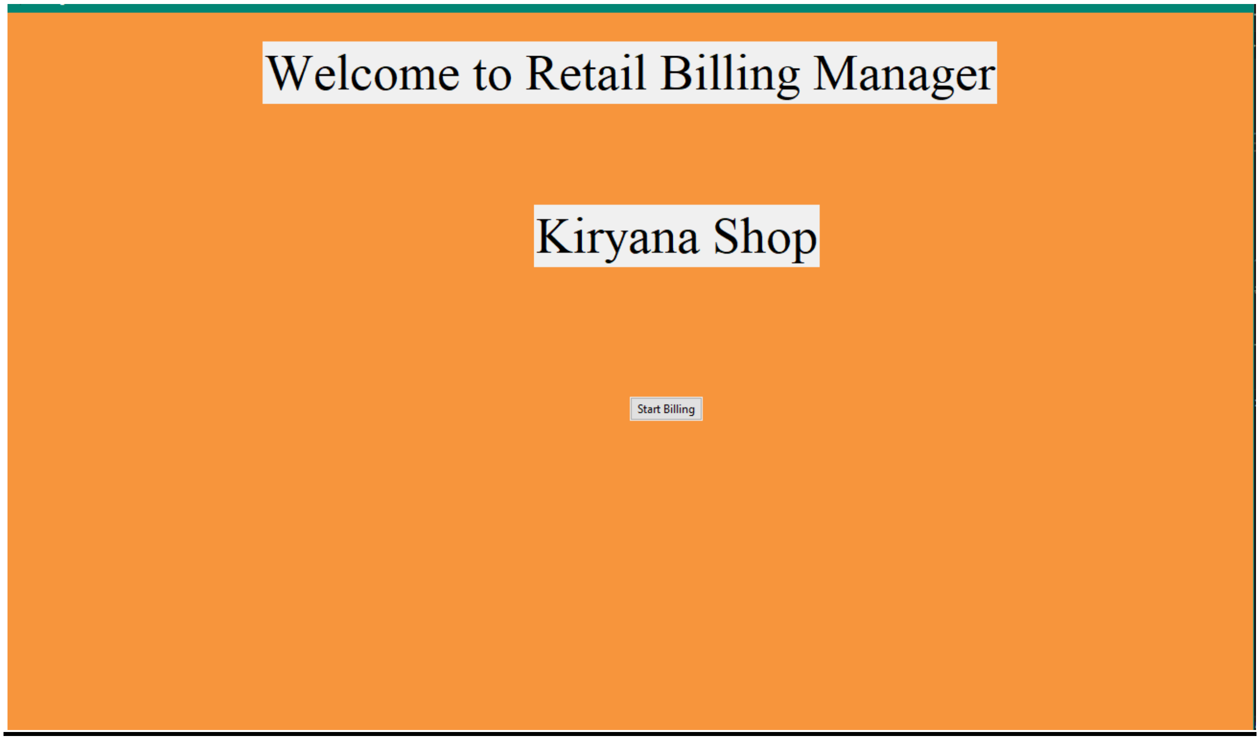
```python

    x_coordinate = (screen_width - window.winfo_width()) // 2
    y_coordinate = (screen_height - window.winfo_height()) // 2

    window.geometry(f"+{x_coordinate}+{y_coordinate}")

window = tk.Tk()
window.geometry('1300x800')
window.resizable(0, 0)
# width = 600 # Width
# height = 300 # Height
center_window(window)

window.title('Billing Software')
window.configure(bg='#f7953c')
ttk.Label(window, text="Welcome to Retail Billing Manager", font=("times new roman", 40)).pack(pady=30)
ttk.Label(window, text="Kiryana Shop", font=("times new roman", 40)).place(x=550,y=200)

ttk.Button(window, text='Start Billing', command=start_program).place(x=650, y=400)



frame1 = tk.Frame(window, bg='#c3c3c3')
frame1.configure(width=1220, height=750)
frame1.pack_propagate(False)

title_label = tk.Label(frame1, text="Retail Billing Manager", font=('arial', 40), bg='#c3c3c3')
title_label.pack(pady=50)

start_button = ttk.Button(frame1, text="Start", command=start_program)
start_button.pack()

frame2 = tk.Frame(frame1, relief='groove', bd=10)
frame2.pack(side='right', padx=30, pady=150)
frame2.configure(width=500, height=550)
frame2.pack_propagate(False)

bill_title = tk.Label(frame2, text='Receipt', font='arial 15 bold', bd=7, relief='groove')
bill_title.pack(fill='x')

bill_title = tk.Label(frame2, text='Name\t\tQuantity\t\tPrice', font='arial 15 bold', bd=7, relief='groove')
bill_title.pack(fill='x')
```

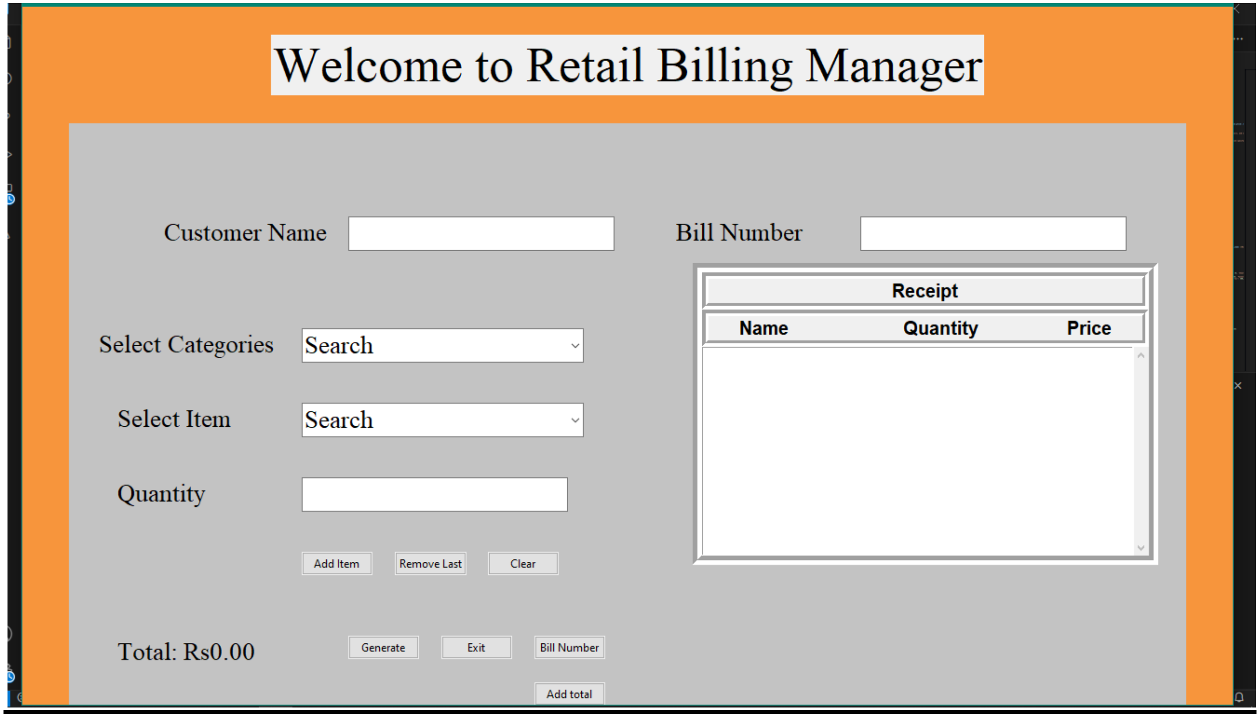

main.py
Workspace Trust
C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...

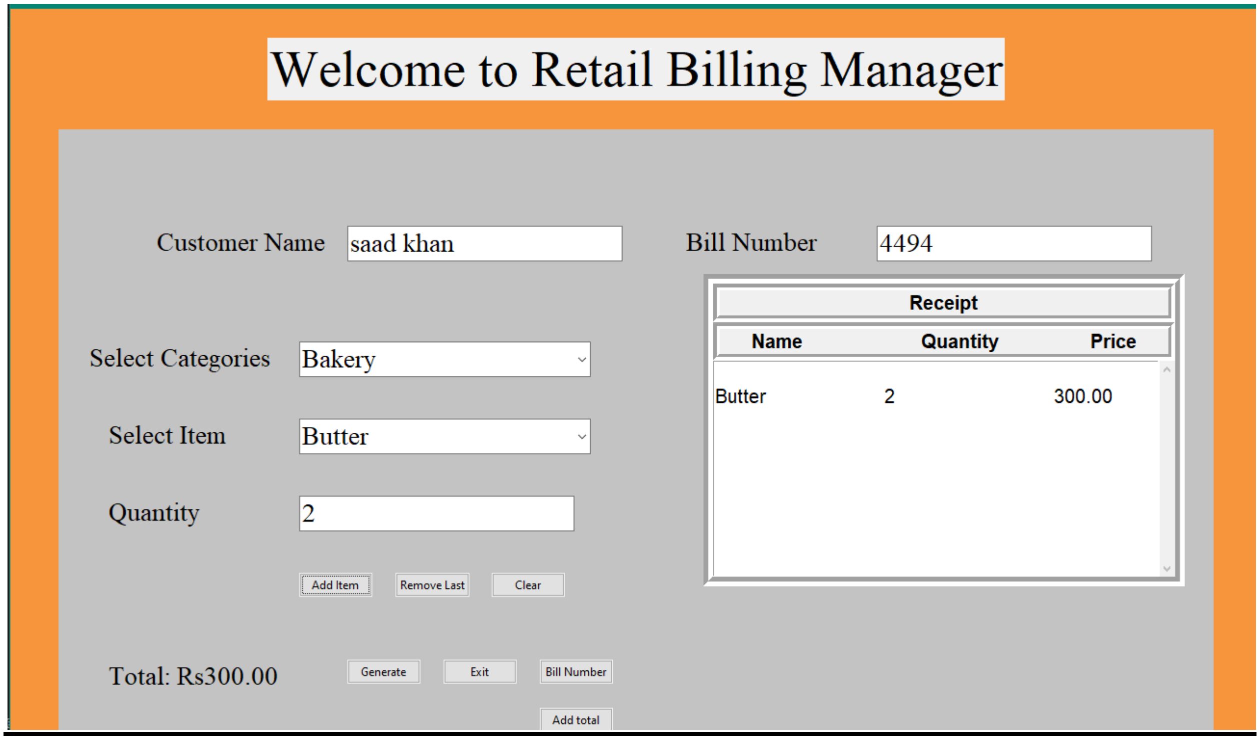
```
scrol_y = ttk.Scrollbar(frame2, orient='vertical')
scrol_y.pack(side='right', fill='y')
textarea = tk.Text(frame2, font='arial 15', yscrollcommand=scrol_y.set)
textarea.pack(fill='both')
scrol_y.config(command=textarea.yview)

cons_lb = ttk.Label(frame1, text="Customer Name", font=('times new roman', 20), background="#c3c3c3")
cons_lb.place(x=100, y=100)
Bill_num_b = ttk.Entry(frame1, text="", font=('times new roman', 20))
Bill_num_b.place(x=300, y=100)

Bill_num = ttk.Label(frame1, text="Bill Number", font=('times new roman', 20), background="#c3c3c3")
Bill_num.place(x=650, y=100)
cont_num = ttk.Entry(frame1, text="", font=('times new roman', 20))
cont_num.place(x=850, y=100)

catogr = ttk.Label(frame1, text="Select Categories", font=('times new roman', 20), background="#c3c3c3")
catogr.place(x=30, y=220)
category_combo = ttk.Combobox(frame1, font=('times new roman', 20), values=list(lis.keys()))
category_combo.place(x=250, y=220)
category_combo.set('Search')
category_combo.bind("<<ComboboxSelected>>", Search)

item_label = ttk.Label(frame1, text="Select Item", font=('times new roman', 20), background="#c3c3c3")
item_label.place(x=50, y=300)
combo = ttk.Combobox(frame1, font=('times new roman', 20),values=list(lis.keys()))
combo.place(x=250, y=300)
combo.set('Search')
combo.bind("<<ComboboxSelected>>", Search)

quantity_label = ttk.Label(frame1, text="Quantity", font=('times new roman', 20), background="#c3c3c3")
quantity_label.place(x=50, y=380)
quantity_entry = ttk.Entry(frame1, text="", font=('times new roman', 20))
quantity_entry.place(x=250, y=380)

add = ttk.Button(frame1, text="Add Item", command=calculate_total)
add.place(x=250, y=460)
remove = ttk.Button(frame1, text="Remove Last", command=remove_item)
remove.place(x=350, y=460)
clear = ttk.Button(frame1, text="Clear", command=clear_all)
clear.place(x=450, y=460)
```


main.py
Workspace Trust
C: > Users > Abc > Desktop > saad khan pyton course > main.py > ...

```

generate = ttk.Button(frame1, text="Generate", command=generate_bill)
generate.place(x=300, y=550)
exit_button = ttk.Button(frame1, text="Exit",command=close_exit)
exit_button.place(x=400, y=550)
random_bt = ttk.Button(frame1, text='Bill Number', command=random_num)
random_bt.place(x=500, y=550)

total_label = ttk.Label(frame1, text="Total: Rs0.00", font=('times new roman', 20), background="#c3c3c3")
total_label.place(x=50, y=550)

add__toatal_ = ttk.Button(frame1, text='Add total', command=add__total)
add__toatal_.place(x=500, y=600)



# Initialize total amount
total_amount = 0.0

window.mainloop()
```


Welcome to Retail Billing Manager
Kiryana Shop
Start Billing


Welcome to Retail Billing Manager
Customer Name
Bill Number
Receipt
Name
Quantity
Price
Select Categories
Search
Select Item
Search
Quantity
Add Item
Remove Last
Clear
Total: Rs0.00
Generate
Exit
Bill Number
Add total



Welcome to Retail Billing Manager

Customer Name: saad khan

Bill Number: 4494

Select Categories: Bakery

Select Item: Butter

Quantity: 2

Receipt		
Name	Quantity	Price
Butter	2	300.00

Total: Rs300.00

GITHUB LINK :

HTTPS://GITHUB.COM/SAADENTERPRENEOUR/-BANOQABIL_PYTHON_COURSE.-