INDEX

SYNOPSIS

AIM: To create a supermarket billing program where the admin can bill out the details such as the item name, price, quantity, total price, etc about the items that are purchased by the customer.

OBJECTIVE: It is a Python MySql connectivity program. The billing program allows the admin to bill out the items that the customer buys from the store. It also displays all details including item name, cost, quantity, total cost, etc. As the admin, you can add, modify and delete items and the admin has a unique id and password which can be used in the future to access the main program.

ABOUT THE PROGRAM:
The program consists of a login page after logging in you are directed to the main window where you can ADD ITEM, MODIFY STOCK, REMOVE ITEM or continue billing out products that the customer intends to buy

- The **ADD ITEM** button on the main page leads you to a new window titled "Add item" in this window you can enter new items into your store. The item code can not be replicated and needs to be unique for each item. You are required to input item code, item name, MRP, quantity, and our price or the price you are selling the product.

- The **MODIFY STOCK** button on the main page leads you to a new window titled "Modify item" in this window you can choose any existing item from the store and update any of the values associated with it. The entry fields are automatically filled by clicking on the item shown above the entry fields. You can change the values you want to change and click on the "modify item" button at the bottom of the page to save the change you made.

- The **REMOVE ITEM** button on the main page leads you to a new window titled "Remove item" in this window you can remove any selected item from your store. You can select the intended item by clicking on the item shown in the window. The page also has an option to remove all items from the store.

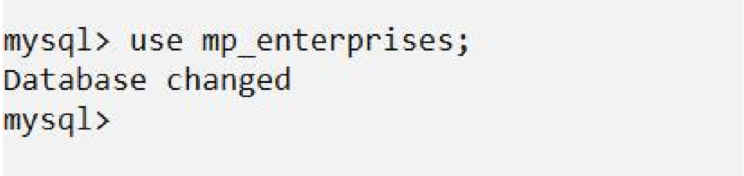
The program allows the admin to bill items the customer wants to buy.
At the end of the program, a bill will be displayed showing the customer name, the customer phone number, date and time, the mode of payment, the person who generated the bill, the item names, MRP, the quantity of the items, cost of the item, and total cost of the purchase by the customer to confirm the purchase.

Name of Database used:-mp_enterprises
Name of Tables used:-admin_account,purchase,stock

DATABASE & TABLE STRUCTURE

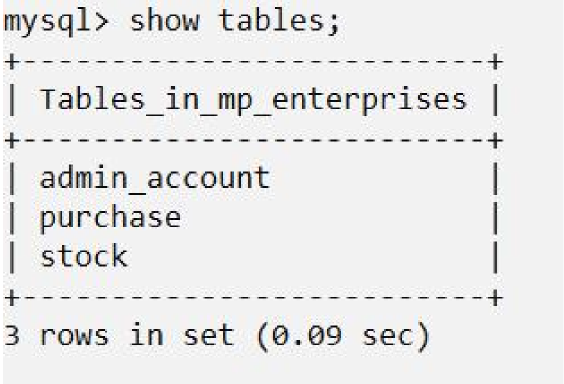
Name of Database used:-mp_enterprises
Name of Tables used:-admin_account,purchase,stock

Database:- mp_enterprises



```
mysql> use mp_enterprises;
Database changed
mysql>
```

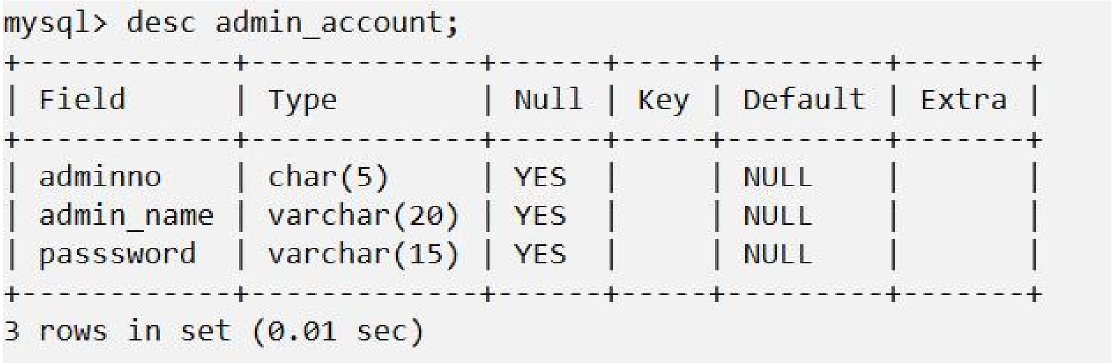
Tables:-admin_account,purchase,stock



```
mysql> show tables;
+-------------------------+
| Tables_in_mp_enterprises |
+-------------------------+
| admin_account           |
| purchase                |
| stock                   |
+-------------------------+
3 rows in set (0.09 sec)
```

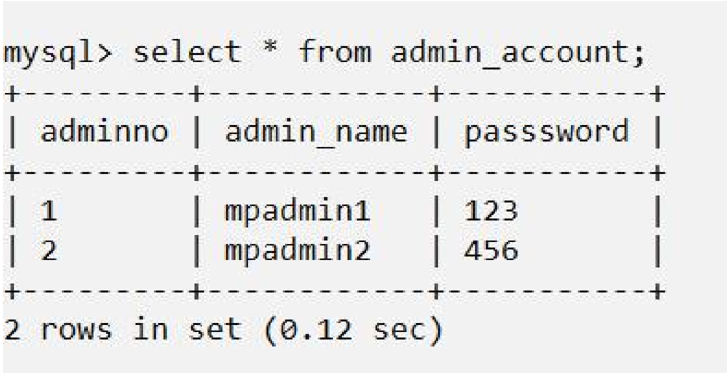
Admin account table

Structure of admin_account table:-



```
mysql> desc admin_account;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| adminno    | char(5)     | YES  |     | NULL    |       |
| admin_name | varchar(20) | YES  |     | NULL    |       |
| password   | varchar(15) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

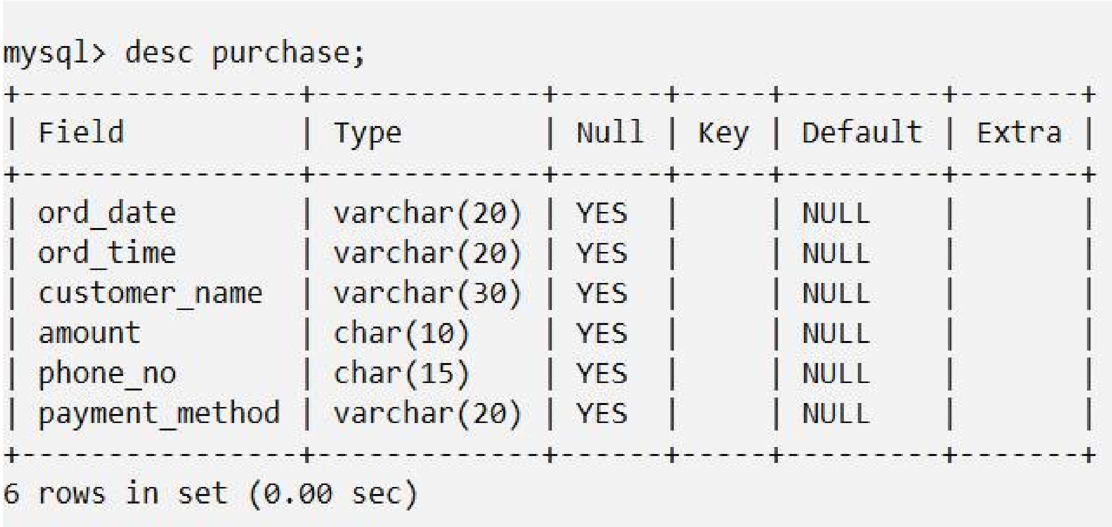
Records in admin_account:-



```
mysql> select * from admin_account;
+---------+------------+-----------+
| adminno | admin_name | passsword |
+---------+------------+-----------+
| 1       | mpadmin1   | 123       |
| 2       | mpadmin2   | 456       |
+---------+------------+-----------+
2 rows in set (0.12 sec)
```

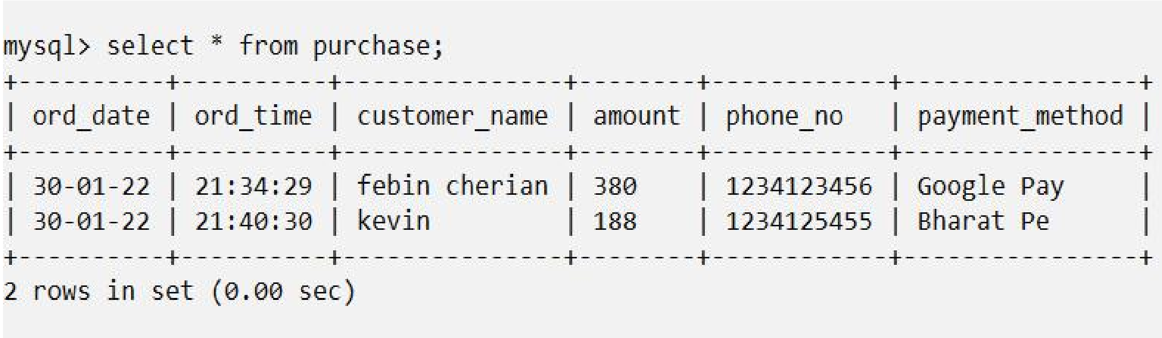
Purchase table

Structure of purchase table:-



```
mysql> desc purchase;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| ord_date       | varchar(20) | YES  |     | NULL    |       |
| ord_time       | varchar(20) | YES  |     | NULL    |       |
| customer_name  | varchar(30) | YES  |     | NULL    |       |
| amount         | char(10)    | YES  |     | NULL    |       |
| phone_no       | char(15)    | YES  |     | NULL    |       |
| payment_method | varchar(20) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

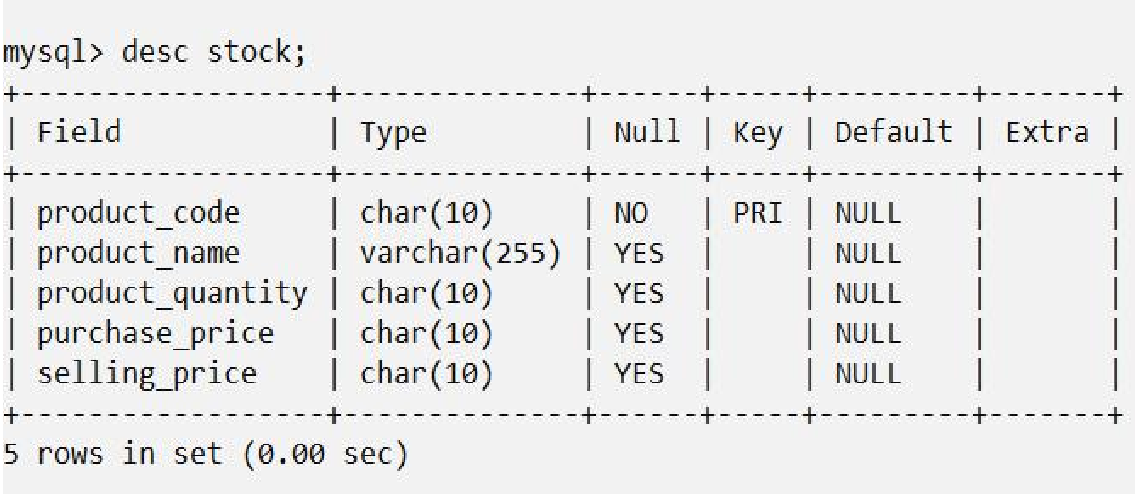
Records in purchase:-



```
mysql> select * from purchase;
+----------+----------+---------------+--------+------------+----------------+
| ord_date | ord_time | customer_name | amount | phone_no   | payment_method |
+----------+----------+---------------+--------+------------+----------------+
| 30-01-22 | 21:34:29 | febin cherian | 380    | 1234123456 | Google Pay     |
| 30-01-22 | 21:40:30 | kevin         | 188    | 1234125455 | Bharat Pe      |
+----------+----------+---------------+--------+------------+----------------+
2 rows in set (0.00 sec)
```

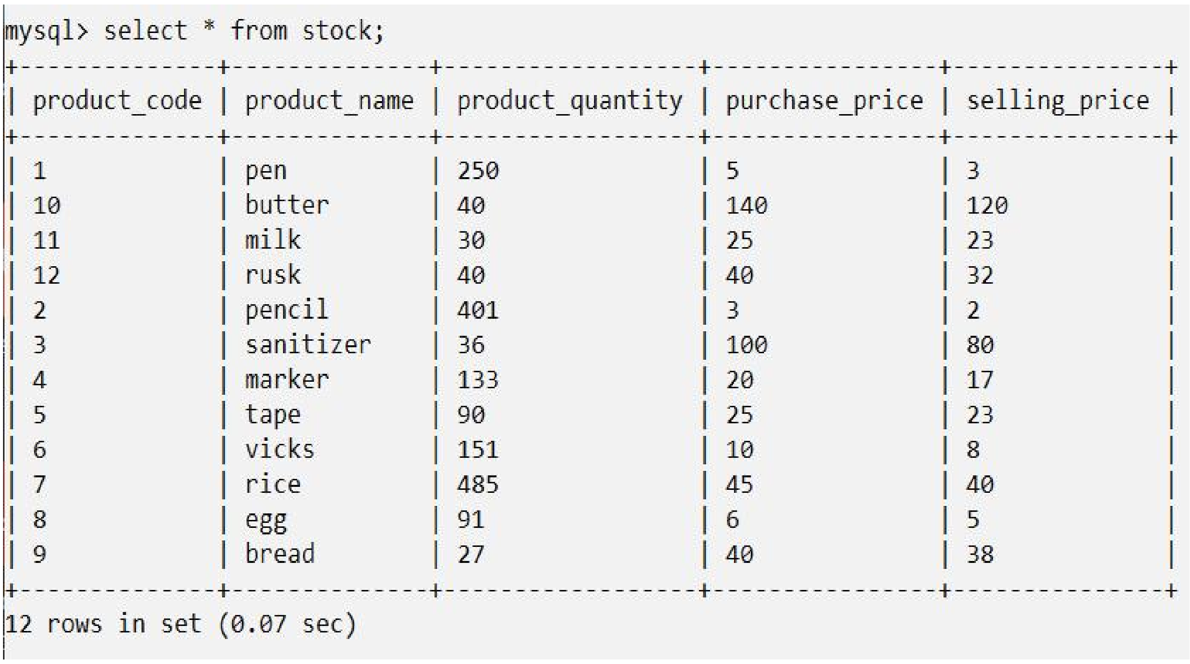
Stock table

Structure of stock table:-



```
mysql> desc stock;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| product_code     | char(10)     | NO   | PRI | NULL    |       |
| product_name     | varchar(255) | YES  |     | NULL    |       |
| product_quantity | char(10)     | YES  |     | NULL    |       |
| purchase_price   | char(10)     | YES  |     | NULL    |       |
| selling_price    | char(10)     | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Records in stock:-



```
mysql> select * from stock;
+--------------+--------------+------------------+----------------+---------------+
| product_code | product_name | product_quantity | purchase_price | selling_price |
+--------------+--------------+------------------+----------------+---------------+
| 1            | pen          | 250              | 5              | 3             |
| 10           | butter       | 40               | 140            | 120           |
| 11           | milk         | 30               | 25             | 23            |
| 12           | rusk         | 40               | 40             | 32            |
| 2            | pencil       | 401              | 3              | 2             |
| 3            | sanitizer    | 36               | 100            | 80            |
| 4            | marker       | 133              | 20             | 17            |
| 5            | tape         | 90               | 25             | 23            |
| 6            | vicks        | 151              | 10             | 8             |
| 7            | rice         | 485              | 45             | 40            |
| 8            | egg          | 91               | 6              | 5             |
| 9            | bread        | 27               | 40             | 38            |
+--------------+--------------+------------------+----------------+---------------+
12 rows in set (0.07 sec)
```

PROGRAM CODE

```
#importing needed modules
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
import mysql.connector
import datetime
from tkinter import filedialog

#Establishing connection with database and creating a cursor
mysqldb=mysql.connector.connect(host="localhost",user="root",passwd="youtube24/7",
                                database="mp_enterprises")
cursor=mysqldb.cursor()
GT=[]


def login():                                #accepting login credentials
    user=admin_id.get()
    passwd=admin_passwd.get()

    logindb(user,passwd)



def logindb(user,passwd):                   #processing login credentials
    if passwd:
        cursor.execute("""select passsword from admin_account
               where admin_name='{}'""".format(user))
        data=cursor.fetchall()
        if passwd!=data[0][0]:
            messagebox.showwarning("INVALID","Sorry Wrong Password or Admin id")
        else:
            main_program()




def main_program():                                         #main program function

    def saveall():                                          #function to record purchase and save bill
        cursor.execute("""insert into purchase
               values('{}','{}','{}',{},{},'{}')""".format(ndate,
                                                 ntime,
                                                 cname.get(),
                                                 grtotal.get(),
                                                 phoneno.get(),
                                                 payselected.get()))

        mysqldb.commit()
        text=filedialog.asksaveasfilename(initialdir=r"C:\Users\billing",title="SAVE BILL",
                                filetypes=(("text files","*.txt"),))


        text_file=open(text,'w')
        text_file.write(textarea.get(1.0,END))
        cname.set('')
        grtotal.set('')
        phoneno.set('')
        textarea.delete(1.0,END)
```

```
def intobill():                                                  #function to enter items into the bill
        if len(GT)==0:
            TOT=tamt.get()
            GT.append(TOT)
        else:
            TOT=tamt.get()
            GT.append(TOT)
        global GGTOT
        GGTOT=sum(GT)

        cursor.execute("""select product_quantity
               from stock where product_name='{}'""".format(nam.get()))

        data=cursor.fetchall()
        oldqty=data[0][0]
        newqty=int(oldqty)-int(qty1.get())

        cursor.execute("""update stock set product_quantity={}
               where product_name='{}'""".format(newqty,nam.get()))

        mysqldb.commit()



        if textarea.get(1.0,1.1)!=(''):
            textarea.insert(END,f"{nam.get()}")
            textarea.insert(END,f"\t{mrp1.get()}")
            textarea.insert(END,f"\t{opr.get()}")
            textarea.insert(END,f"\t{qty1.get()}")
            textarea.insert(END,f"\t{tamt.get()}\n")
        else:                                                                        #Bill header
            textarea.delete(1.0,END)
            textarea.insert(END,"                    MP ENTERPRISES\n")
            textarea.insert(END,f"\nDate:{ndate}                         Day:{day}\n")
            textarea.insert(END,f"Customer:{cname.get()}\n")
            textarea.insert(END,f"Ph:{phoneno.get()}\n")
            textarea.insert(END,f"Payment Type:{payselected.get()}\n")
            textarea.insert(END,"___________________________________\n")
            textarea.insert(END,"*******************************************************\n")
            textarea.insert(END," item")
            textarea.insert(END,"\tMRP")
            textarea.insert(END,"\tour")
            textarea.insert(END,"\tqty")
            textarea.insert(END,"\tTotal\n")
            textarea.insert(END,"                              price\n")
            textarea.insert(END,"*******************************************************\n")
            textarea.insert(END,f"{nam.get()}")
            textarea.insert(END,f"\t{mrp1.get()}")
            textarea.insert(END,f"\t{opr.get()}")
            textarea.insert(END,f"\t{qty1.get()}")
            textarea.insert(END,f"\t{tamt.get()}\n")



        cod.set('')
        nam.set('')
        mrp1.set('')
        qty1.set('')
        opr.set('')
        tamt.set('')
```

```
def finish():                                             # function to format bill ending
    textarea.insert(END,"*******************************************************\n")
    textarea.insert(END,f"\t\t     GRAND TOTAL: {GGTOT}\n")
    textarea.insert(END,f"\nCashier:{admin_id.get()}\t\t\tTime:{ntime}\n")
    textarea.insert(END,"*******************************************************\n")
    textarea.insert(END,"\t           THANK YOU\n\tFOR SHOPPING WITH US\n")
    textarea.insert(END,"_____________________________________\n")
    grtotal.set(GGTOT)


  def additm():                                           #function to add item into the program


    def adddata():
      cursor.execute("""insert into stock
          values({},"{}",{},{},{})""".format(itcodeety.get(),
                                  itnameety.get(),
                                  itqtyety.get(),
                                  itMRPety.get(),
                                  itoprcety.get()))
      mysqldb.commit()

      itcodeety.delete(0,END)
      itnameety.delete(0,END)
      itMRPety.delete(0,END)
      itqtyety.delete(0,END)
      itoprcety.delete(0,END)
      treedata.delete(*treedata.get_children())          #*=splat used for unpacking iterable objects

      data()


    def data():                                     #fuction to show records from stock
      cursor.execute("select * from stock")
      data=cursor.fetchall()

      global cnt
      cnt=0

      for x in data:
        if cnt%2==0:
          treedata.insert(parent='',index='end',iid=cnt,text='',
                  values=(x[0],x[1],x[3],x[2],x[4]))

        else:
          treedata.insert(parent='',index='end',iid=cnt,text='',
                  values=(x[0],x[1],x[3],x[2],x[4]))

        cnt+=1


    def selectrec(e):                                 #fucton used to autofill values
      itcodeety.delete(0,END)
      itnameety.delete(0,END)
      itMRPety.delete(0,END)
      itqtyety.delete(0,END)
      itoprcety.delete(0,END)
    selected=treedata.focus()
    values=treedata.item(selected, "values")
    itcodeety.insert(0,values[0])
    itnameety.insert(0,values[1])
    itMRPety.insert(0,values[2])
    itqtyety.insert(0,values[3])
    itoprcety.insert(0,values[4])



sub1=Toplevel()
sub1.title("ADD ITEM")
sub1.geometry("900x530")


style=ttk.Style()                                    #tkinter

style.theme_use("default")                                    #theme

style.configure("Treeview",bg="#D3D3D3",fg="black",
            rowheight=25,feildbackground="#D3D3D3")


treeframe=Frame(sub1)                                         #frame
treeframe.pack(pady=10)


treescroll=Scrollbar(treeframe)                              #scrollbar
treescroll.pack(side=RIGHT,fill=Y)
treedata=ttk.Treeview(treeframe,yscrollcommand=treescroll.set,selectmode="extended")
treedata.pack()


treescroll.config(command=treedata.yview)


treedata['columns']=("item code","item name","MRP","quantity","our price")

#structure of the treeview
treedata.column("#0",width=0,stretch=NO)              #needed to eliminate the first feild
treedata.column("item code",anchor=W,width=100)
treedata.column("item name",anchor=CENTER,width=250)
treedata.column("MRP",anchor=CENTER,width=100)
treedata.column("quantity",anchor=CENTER,width=120)
treedata.column("our price",anchor=CENTER,width=120)


treedata.heading("#0",text='',anchor=W)                 #needed to eliminate the first feild
treedata.heading("item code",text='item code',anchor=CENTER)
treedata.heading("item name",text='item name',anchor=CENTER)
treedata.heading("MRP",text='MRP',anchor=CENTER)
treedata.heading("quantity",text='quantity',anchor=CENTER)
treedata.heading("our price",text='our price',anchor=CENTER)

inputfrm=LabelFrame(sub1,text="INPUT")
inputfrm.pack(fill="x",expand="yes",padx=20)


itcodelbl=Label(inputfrm,text="item code")                  #entry box for item code
itcodelbl.grid(row=0,column=0,padx=10,pady=10)
itcodeety=Entry(inputfrm)
itcodeety.grid(row=0,column=1,padx=10,pady=10)
```

```
    itnamelbl=Label(inputfrm,text="item name")                    #entry box for item name
    itnamelbl.grid(row=0,column=2,padx=10,pady=10)
    itnameety=Entry(inputfrm)
    itnameety.grid(row=0,column=3,padx=10,pady=10)

    itMRPlbl=Label(inputfrm,text="MRP")                           #entry box for MRP
    itMRPlbl.grid(row=0,column=4,padx=10,pady=10)
    itMRPety=Entry(inputfrm)
    itMRPety.grid(row=0,column=5,padx=10,pady=10)

    itqtylbl=Label(inputfrm,text="quantity")                      #entry box for quantity
    itqtylbl.grid(row=1,column=0,padx=10,pady=10)
    itqtyety=Entry(inputfrm)
    itqtyety.grid(row=1,column=1,padx=10,pady=10)

    itoprclbl=Label(inputfrm,text="our price")                    #entry box for our price/store price
    itoprclbl.grid(row=1,column=2,padx=10,pady=10)
    itoprcety=Entry(inputfrm)
    itoprcety.grid(row=1,column=3,padx=10,pady=10)


    addbtn=Button(sub1,text="ADD NEW ITEM",command=adddata,bd=3,width=40)
    addbtn.pack(pady=40)

    treedata.bind("<ButtonRelease-1>",selectrec)                 #bind used to run function in a mouse click
    data()

    status=Label(sub1,text="Add item Page",bd=1,
              relief=SUNKEN,anchor=E).pack(side=BOTTOM,fill=X)

    sub1.mainloop()



def removeitm():                                        #remove item button


    def remdata():                                      #function to remove record according to item code
        cursor.execute("delete from stock where product_code={}".format(values[0]))
        mysqldb.commit()

        treedata.delete(*treedata.get_children())       # * splat used for unpacking
        data()

        messagebox.showinfo("DELETED!!","Your Item has been deleted!")


    def remalldata():                                    #function to remove all records
        resp=messagebox.askquestion("Are you sure?","Do you want to delete all items?")
        if resp=="yes":
            cursor.execute("delete from stock")
            mysqldb.commit()

            treedata.delete(*treedata.get_children())

            data()

     def selectrec(e):                                  #function used to find item code of selected item
        global values
        selected=treedata.focus()
        values=treedata.item(selected, "values")
```

```
def data():                                    #fuction to show records from stock
    cursor.execute("select * from stock")
    data=cursor.fetchall()

    global cnt
    cnt=0

    for x in data:
        if cnt%2==0:
            treedata.insert(parent='',index='end',iid=cnt,text='',values=(x[0],x[1],x[3],x[2],x[4]))

        else:
            treedata.insert(parent='',index='end',iid=cnt,text='',values=(x[0],x[1],x[3],x[2],x[4]))


        cnt+=1
sub1=Toplevel()
sub1.title("REMOVE ITEM")
sub1.geometry("800x430")

style=ttk.Style()                         #tkinter

style.theme_use("default")                #theme


style.configure("Treeview",bg="#D3D3D3",fg="black",
            rowheight=25,feildbackground="#D3D3D3")


treeframe=Frame(sub1)                     #frame
treeframe.pack(pady=10)

treescroll=Scrollbar(treeframe)           #scrollbar
treescroll.pack(side=RIGHT,fill=Y)


treedata=ttk.Treeview(treeframe,yscrollcommand=treescroll.set,selectmode="extended")
treedata.pack()


treescroll.config(command=treedata.yview)


treedata['columns']=("item code","item name","MRP","quantity","our price")

#structure of the treeview
treedata.column("#0",width=0,stretch=NO)
treedata.column("item code",anchor=W,width=100)
treedata.column("item name",anchor=CENTER,width=250)
treedata.column("MRP",anchor=CENTER,width=100)
treedata.column("quantity",anchor=CENTER,width=120)
treedata.column("our price",anchor=CENTER,width=120)


treedata.heading("#0",text='',anchor=W)
treedata.heading("item code",text='item code',anchor=CENTER)
treedata.heading("item name",text='item name',anchor=CENTER)
treedata.heading("MRP",text='MRP',anchor=CENTER)
treedata.heading("quantity",text='quantity',anchor=CENTER)
treedata.heading("our price",text='our price',anchor=CENTER)
```

```
    rembtn=Button(sub1,text="REMOVE SELECTED ITEM",command=remdata,bd=3,width=40)
    rembtn.pack(pady=15)                                   #remove button

    remallbtn=Button(sub1,text="REMOVE ALL ITEMS",command=remalldata,bd=3,width=20,bg="red")
    remallbtn.pack(pady=5,anchor=S)                        #remove all button

    treedata.bind("<ButtonRelease-1>",selectrec)       #bind used to run function in a mouse click

    data()

    status=Label(sub1,text="Remove item Page",bd=1,
              relief=SUNKEN,anchor=E).pack(side=BOTTOM,fill=X)

    sub1.mainloop()



def modifyitm():                                              #function to modify item

    def moddata():                                           #function to accept the updated values
        selected=treedata.focus()
        treedata.item(selected,text='',value=(itcodeety.get(),
                                    itnameety.get(),
                                    itMRPety.get(),
                                    itqtyety.get(),
                                    itoprcety.get(),))

        cursor.execute("""update stock set product_name='{x}',product_quantity={y},
               purchase_price={z},selling_price={a}
               where product_code={b}""".format(x=itnameety.get(),
                                     y=itqtyety.get(),
                                     z=itMRPety.get(),
                                     a=itoprcety.get(),
                                     b=itcodeety.get()))

        mysqldb.commit()

        itcodeety.delete(0,END)
        itnameety.delete(0,END)
        itMRPety.delete(0,END)
        itqtyety.delete(0,END)
        itoprcety.delete(0,END)


    def data():                                              #fuction to show records from stock
        cursor.execute("select * from stock")
        data=cursor.fetchall()

        global cnt
        cnt=0
        for x in data:
            if cnt%2==0:
                treedata.insert(parent='',index='end',iid=cnt,text='',
                        values=(x[0],x[1],x[3],x[2],x[4]))

            else:
                treedata.insert(parent='',index='end',iid=cnt,text='',
                        values=(x[0],x[1],x[3],x[2],x[4]))

            cnt+=1
```

```
def selectrec(e):                                        #fucton used to autofill values
    itcodeety.delete(0,END)
    itnameety.delete(0,END)
    itMRPety.delete(0,END)
    itqtyety.delete(0,END)
    itoprcety.delete(0,END)

    selected=treedata.focus()
    values=treedata.item(selected, "values")
    itcodeety.insert(0,values[0])
    itnameety.insert(0,values[1])
    itMRPety.insert(0,values[2])
    itqtyety.insert(0,values[3])
    itoprcety.insert(0,values[4])


sub1=Toplevel()
sub1.title("MODIFY ITEM")
sub1.geometry("900x530")


style=ttk.Style()                          #tkinter

style.theme_use("default")           #theme

style.configure("Treeview",bg="#D3D3D3",fg="black",
               rowheight=25,feildbackground="#D3D3D3")


treeframe=Frame(sub1)              #frame
treeframe.pack(pady=10)


treescroll=Scrollbar(treeframe)    #scrollbar
treescroll.pack(side=RIGHT,fill=Y)


treedata=ttk.Treeview(treeframe,yscrollcommand=treescroll.set,selectmode="extended")
treedata.pack()


treescroll.config(command=treedata.yview)


treedata['columns']=("item code","item name","MRP","quantity","our price")

# structure of the treeview
treedata.column("#0",width=0,stretch=NO)
treedata.column("item code",anchor=W,width=100)
treedata.column("item name",anchor=CENTER,width=250)
treedata.column("MRP",anchor=CENTER,width=100)
treedata.column("quantity",anchor=CENTER,width=120)
treedata.column("our price",anchor=CENTER,width=120)


treedata.heading("#0",text='',anchor=W)
treedata.heading("item code",text='item code',anchor=CENTER)
treedata.heading("item name",text='item name',anchor=CENTER)
treedata.heading("MRP",text='MRP',anchor=CENTER)
treedata.heading("quantity",text='quantity',anchor=CENTER)
treedata.heading("our price",text='our price',anchor=CENTER)
```

```
     inputfrm=LabelFrame(sub1,text="INPUT")
    inputfrm.pack(fill="x",expand="yes",padx=20)

    itcodelbl=Label(inputfrm,text="item code")                    #entry box for item code
    itcodelbl.grid(row=0,column=0,padx=10,pady=10)
    itcodeety=Entry(inputfrm)
    itcodeety.grid(row=0,column=1,padx=10,pady=10)

    itnamelbl=Label(inputfrm,text="item name")                    #entry box for item name
    itnamelbl.grid(row=0,column=2,padx=10,pady=10)
    itnameety=Entry(inputfrm)
    itnameety.grid(row=0,column=3,padx=10,pady=10)


    itMRPlbl=Label(inputfrm,text="MRP")                           #entry box for MRP
    itMRPlbl.grid(row=0,column=4,padx=10,pady=10)
    itMRPety=Entry(inputfrm)
    itMRPety.grid(row=0,column=5,padx=10,pady=10)


    itqtylbl=Label(inputfrm,text="quantity")                      #entry box for quantity
    itqtylbl.grid(row=1,column=0,padx=10,pady=10)
    itqtyety=Entry(inputfrm)
    itqtyety.grid(row=1,column=1,padx=10,pady=10)


    itoprclbl=Label(inputfrm,text="our price")                    #entry box for our price/store price
    itoprclbl.grid(row=1,column=2,padx=10,pady=10)
    itoprcety=Entry(inputfrm)
    itoprcety.grid(row=1,column=3,padx=10,pady=10)

    modbtn=Button(sub1,text="MODIFY ITEM",command=moddata,bd=3,width=30)
    modbtn.pack(pady=40)                                          #modify button

    treedata.bind("<ButtonRelease-1>",selectrec)                  #bind used to run function in a mouse click

    data()

    status=Label(sub1,text="Modify item Page",bd=1,
            relief=SUNKEN,anchor=E).pack(side=BOTTOM,fill=X)

    sub1.mainloop()


def totfitem():                              # function to calculate total price for a item
    totitmprc=int(qty1.get())*int(opr.get())
    tamt.set(int(totitmprc))


def newf():                                 #function to clear all values in the main page
    cname.set("")
    phoneno.set("")
    grtotal.set("")
    cod.set('')
    nam.set('')
    mrp1.set('')
    qty1.set('')
    opr.set('')
    tamt.set('')
    clear()
```

```
def comboclick(e):                    #function to auto fill all values regarding a chosen item
  cursor.execute("""select product_code,purchase_price,
      selling_price from stock where product_name='{}'""".format(nam.get()))

  data=cursor.fetchall()
  cod.set(data[0][0])
  mrp1.set(data[0][1])
  opr.set(data[0][2])


def clear():                          #function to clear the textarea
  textarea.delete(1.0,END)       # txtarea staring from=1.0


main=Toplevel()
main.title("billing software")
main.geometry("1245x645")


frmA=LabelFrame(main,text="DETAILS",relief="ridge",
          fg="black",bg="#edebe1",bd=5)   #code=hex colour code

frmA.place(x=0,y=0,relwidth=1)

#day,date and time
now=datetime.datetime.now()
ndate=now.strftime("%d-%m-%y")
ntime=now.strftime("%H:%M:%S")
day=now.strftime("%A")

#current day automatically set. can be changed through dropdown
dy=Label(frmA,text="DAY:-",font=("geordia bold",10),bg="#edebe1").grid(row=0,column=0)
dyselected=StringVar()
dyselected.set(day)
dydrop=OptionMenu(frmA,dyselected,"Monday","Tuesday","Wednesday",
            "Thursday","Friday","Saturday","Sunday").grid(row=0,column=1)

#current date automatically set. can be changed
date=Label(frmA,text="DATE:-",font=("geordia bold",10),bg="#edebe1").grid(row=1,column=0)
dateselected=StringVar()
dateselected.set(ndate)
datenow=Entry(frmA,textvariable=dateselected,width=8,
         font="american 14",relief=SUNKEN,bd=1).grid(row=1,column=1)

#mode of pay set to cash. Can be changed through dropdown
pay=Label(frmA,text="PAYMENT:-",font=("geordia bold",10),bg="#edebe1").grid(row=2,column=0)
payselected=StringVar()
payselected.set("Cash")
paydrop=OptionMenu(frmA,payselected,"Cash","Credit Card",
            "Debit Card","Google Pay","Bharat Pe").grid(row=2,column=1)


#space between date and name entries
labelspace=Label(frmA,text="       ",bg="#edebe1").grid(row=1,column=2)

#entry for customer name
customers_name=Label(frmA,text="CUSTOMERS:-",
             font=("geordia bold",10),bg="#edebe1").grid(row=0,column=3)

cname=StringVar()
name_entry=Entry(frmA,textvariable=cname,width=19,
          font="american 14",relief=SUNKEN,bd=2).grid(row=0,column=4)
```

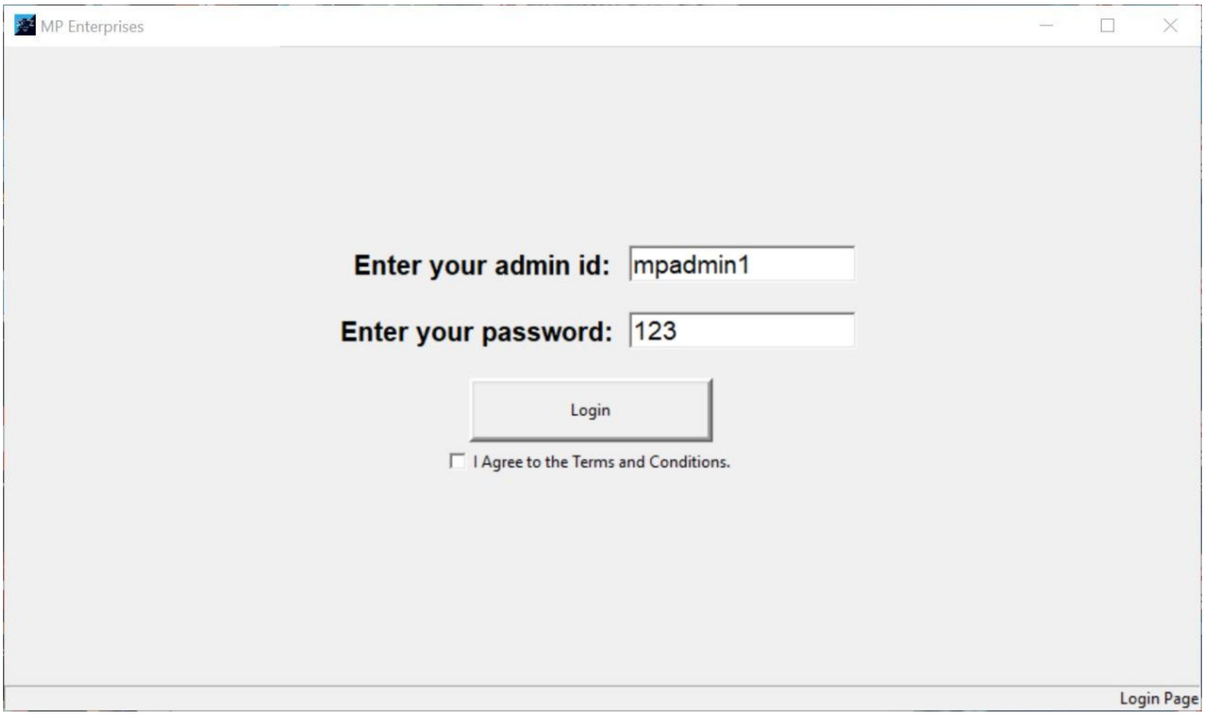
```
#entry for customer phone number
phone_no=Label(frmA,text="PHONE:-",font=("geordia bold",10),
            bg="#edebe1").grid(row=1,column=3)

phoneno=IntVar()
phoneno.set('')
phone_entry=Entry(frmA,textvariable=phoneno,width=19,
              font="american 14",relief=SUNKEN,bd=2).grid(row=1,column=4)

#button to clear all set values in the main page
new=Button(frmA,text="NEW",command=newf,
         padx=40,pady=7,bd=2).grid(row=2,column=3,rowspan=2,pady=10)


#button used to save the purchase and save the bill
save=Button(frmA,text="SAVE",command=saveall,
          padx=40,pady=7,bd=2).grid(row=2,column=4,rowspan=2,pady=10)


#distance between the buttons
labelspace1=Label(frmA,text="                    ",bg="#edebe1").grid(row=0,column=5)
labelspace2=Label(frmA,text="                    ",bg="#edebe1").grid(row=0,column=7)
labelspace3=Label(frmA,text="                    ",bg="#edebe1").grid(row=0,column=9)

#add item button
add_item=Button(frmA,text="ADD ITEM",command=additm,
             padx=33,pady=15,bd=2).grid(row=0,column=6,rowspan=2)

#remove item button
remove_item=Button(frmA,text="REMOVE ITEM",command=removeitm,
               padx=23,pady=15,bd=2).grid(row=0,column=8,rowspan=2)

#modify stock button
modify_stock=Button(frmA,text="MODIFY STOCK",command=modifyitm,
                padx=20,pady=15,bd=2).grid(row=0,column=10,rowspan=2)


frmB=LabelFrame(main, text="BILLING",relief="ridge",
             fg="black",bg="#edebe1",bd=5)   #code=hex colour code


frmB.place(x=0,y=138)

#entry for item code
code=Label(frmB,text="CODE",font=("geordia bold",10),
         bg="#c9cb8f",width=10).grid(row=0,column=0)
cod=IntVar()
cod.set('')
cd=Entry(frmB,textvariable=cod,width=10,font=('Arial',10,'bold')).grid(row=1,column=0)

#button to calculate the total of a specific item
totitm=Button(frmB,text="TOTAL FOR ITEM",padx=40,command=totfitem,
           pady=7,bd=2).grid(row=3,column=2,columnspan=2,pady=10)

#button to save item into the bill
sitmtobill=Button(frmB,text="SAVE ITEM INTO BILL",command=intobill,
              padx=40,pady=7,bd=2).grid(row=3,column=4,columnspan=4,pady=10)

#entry for item name
name=Label(frmB,text="NAME",font=("geordia bold",10),
         bg="#c9cb8f",width=30).grid(row=0,column=1,padx=5)
```

```
 nam=StringVar()
cursor.execute("select product_name from stock")
downitem=tuple(cursor.fetchall())
itnm=ttk.Combobox(frmB,textvariable=nam,width=33,font=('Arial',10,'bold'),values=downitem)
itnm.grid(row=1,column=1)

#entry for MRP
mrp=Label(frmB,text="MRP",font=("geordia bold",10),
         bg="#c9cb8f",width=13).grid(row=0,column=2)

mrp1=IntVar()
mrp1.set(")
itmr=Entry(frmB,textvariable=mrp1,width=15,font=('Arial',10,'bold')).grid(row=1,column=2)

#entry for quantity
qty=Label(frmB,text="QTY",font=("geordia bold",10),
         bg="#c9cb8f",width=11).grid(row=0,column=3,padx=5)

qty1=IntVar()
qty1.set(")
itqt=Entry(frmB,textvariable=qty1,width=13,font=('Arial',10,'bold'))
itqt.grid(row=1,column=3)

#entry for our price/store price
ouprice=Label(frmB,text="OUR PRICE",font=("geordia bold",10),
             bg="#c9cb8f",width=13).grid(row=0,column=4)

opr=IntVar()
opr.set(")
ourprice=Entry(frmB,textvariable=opr,width=14,font=('Arial',10,'bold')).grid(row=1,column=4)

#entry for total amount of the product
totamt=Label(frmB,text="TOTAL AMOUNT",font=("geordia bold",10),
            bg="#c9cb8f",width=15).grid(row=0,column=5,padx=5)

tamt=IntVar()
tamt.set(")
totalamt=Entry(frmB,textvariable=tamt,width=17,font=('Arial',10,'bold')).grid(row=1,column=5)


frmC=LabelFrame(main, text="",relief="ridge",fg="black",bg="#edebe1",bd=5)
frmC.place(x=817,y=139)

#Bill
bill_hed=Label(frmC,text='Bill',font=('arial',15,'bold'),
              bd=8,relief=GROOVE,width=32).pack(fill=X)

scroll=Scrollbar(frmC,orient=VERTICAL)
scroll.pack(side=RIGHT,fil=Y)
textarea=Text(frmC,font=('arial','14'),height=13,width=36,yscrollcommand=scroll.set)
textarea.pack(fill=BOTH,expand=1)
scroll.config(command=textarea.yview)
clr=Button(frmC,text="CLEAR",command=clear,pady=7,
          bd=4,width=15).pack(side=LEFT,expand=True)
#button to end the bill
fns=Button(frmC,text="FINISH",command=finish,pady=7,
          bd=4,width=15).pack(side=RIGHT,expand=True)


frmD=LabelFrame(main, text="TOTAL",relief="ridge",fg="black",
              bg="#edebe1",bd=5,width=50,height=99)
```

```
    frmD.place(x=0,y=525,relwidth=1)

    #entry box for grand total
    totamt=Label(frmD,text="TOTAL AMOUNT:-",font=("geordia bold",15),
                bg="#edebe1").place(x=850,y=25)

    grtotal=IntVar()
    grtotal.set('')
    totamtnow=Entry(frmD,textvariable=grtotal,width=15,font="american 14",
                  relief=SUNKEN,bd=3).place(x=1025,y=28)

    #entry for the name of the current admin using the program
    crtuser=Label(frmD,text="current user:-",font=("geordia bold",10),
                 bg="#edebe1").place(x=5,y=45)

    admin_id1=StringVar()
    admin_id1.set(admin_id.get())#.get() should be used
    crtusernow=Entry(frmD,textvariable=admin_id1,width=10,font="american 14",
                   relief=SUNKEN,bd=1).place(x=90,y=38)

    status=Label(main,text="Billing Page",bd=1,
                relief=SUNKEN,anchor=E).pack(side=BOTTOM,fill=X)

    itnm.bind("<<ComboboxSelected>>",comboclick)    #to auto fill the values when you click a product

    main.mainloop()

#Layout for login page
root=Tk()
root.title("MP Enterprises")
root.iconbitmap("c:/Users/user/Downloads/unnamed.ico")
root.geometry("900x500")

#entry box for admin id. admin id is set to be mpadmin1
admin_id=StringVar()
id=Label(root,text="Enter your admin id:",font=("geordia bold",15,"bold")).place(x=260,y=150)
admin_id.set("mpadmin1")
idtxt=Entry(root,width=15,textvariable=admin_id,font="american 14",
        relief=SUNKEN,bd=2).place(x=470,y=150)

#entry box for admin password. password is set to be 123
admin_passwd=StringVar()
passwd=Label(root,text="Enter your password:",font=("geordia bold",15,"bold")).place(x=250,y=200)
admin_passwd.set("123")
passwdtxt=Entry(root,width=15,textvariable=admin_passwd,font="american 14",
            relief=SUNKEN,bd=2).place(x=470,y=200)


trm_cond=Checkbutton(root,text="I Agree to the Terms and Conditions.").place(x=330,y=300)


status=Label(root,text="Login Page",bd=1,relief=SUNKEN,anchor=E).pack(side=BOTTOM,fill=X)

#Login button
loginbtton=Button(root,text="Login",command=login,padx=70,pady=10,bd=4).place(x=350,y=250)


root.mainloop()
```

SAMPLE OUTPUT

Login Page:-

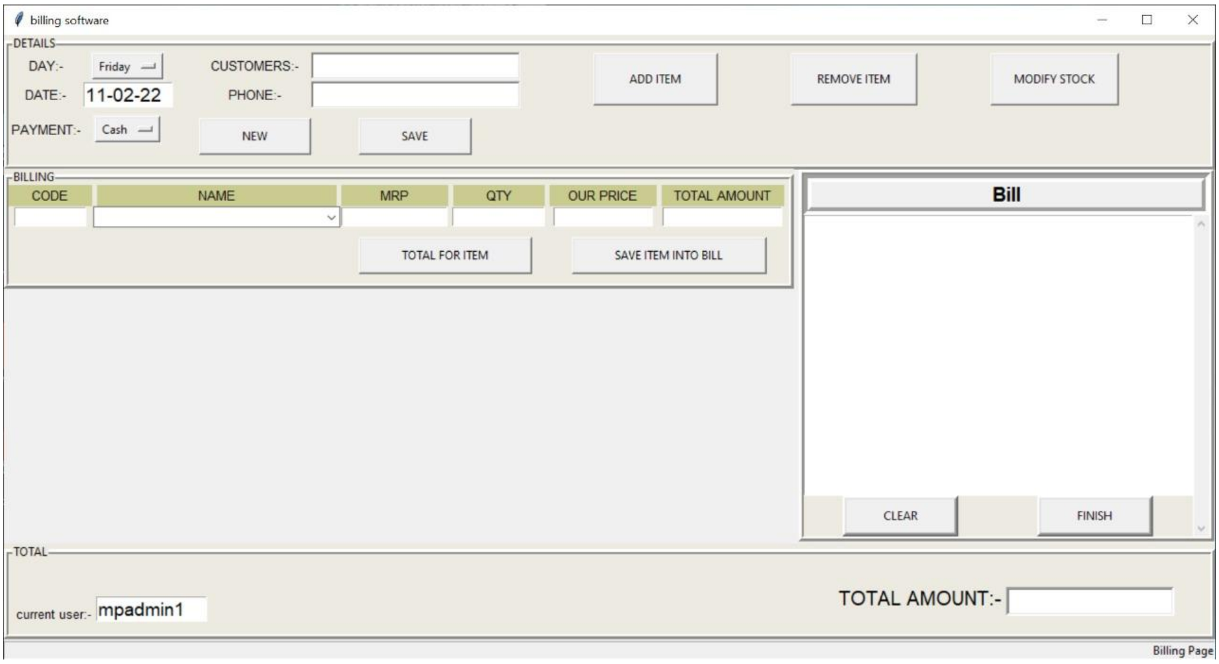


Enter your admin id: mpadmin1

Enter your password: 123

□ I Agree to the Terms and Conditions.

Main Page:-



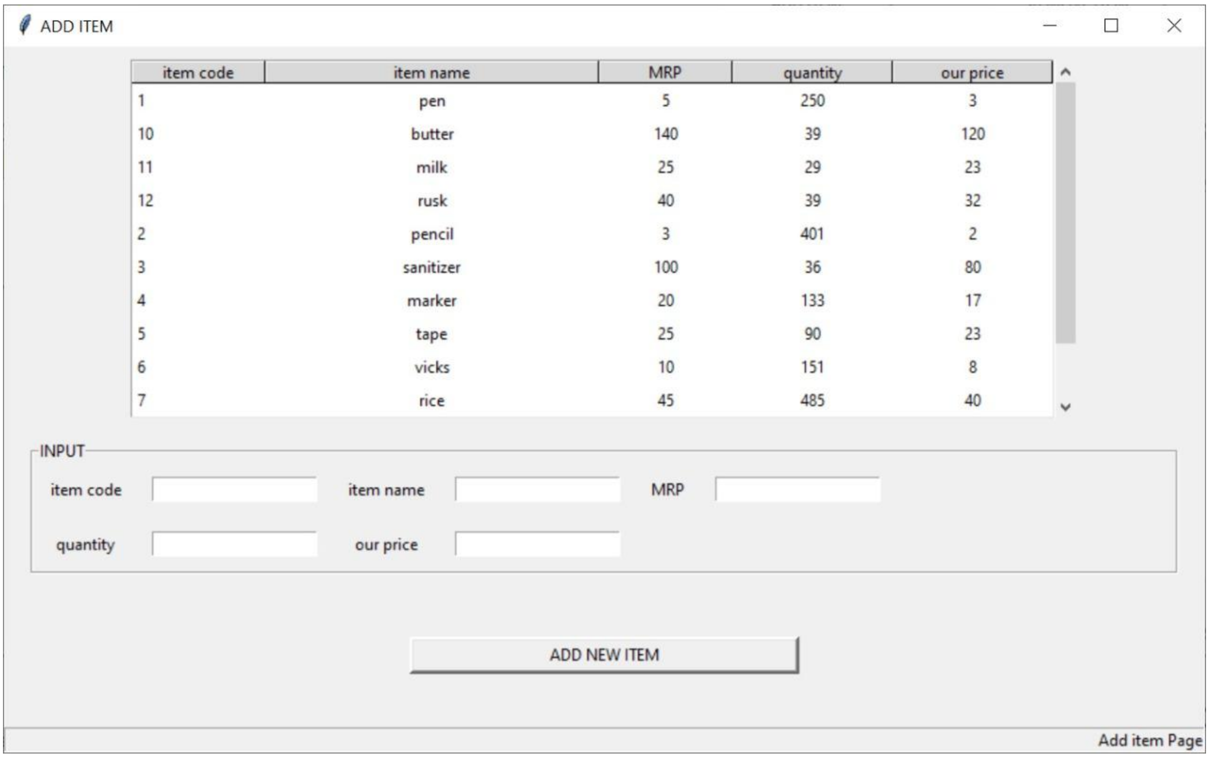
DAY:- Friday

DATE:- 11-02-22

PAYMENT:- Cash

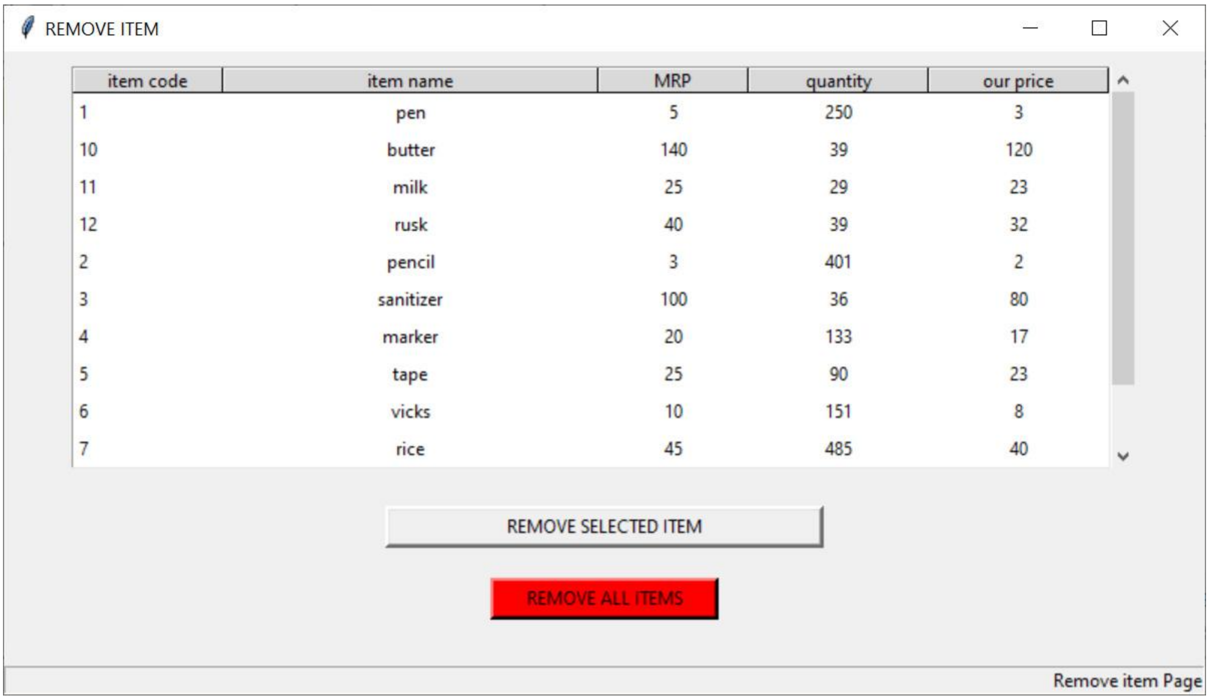
current user:- mpadmin1

Add Item Button:-



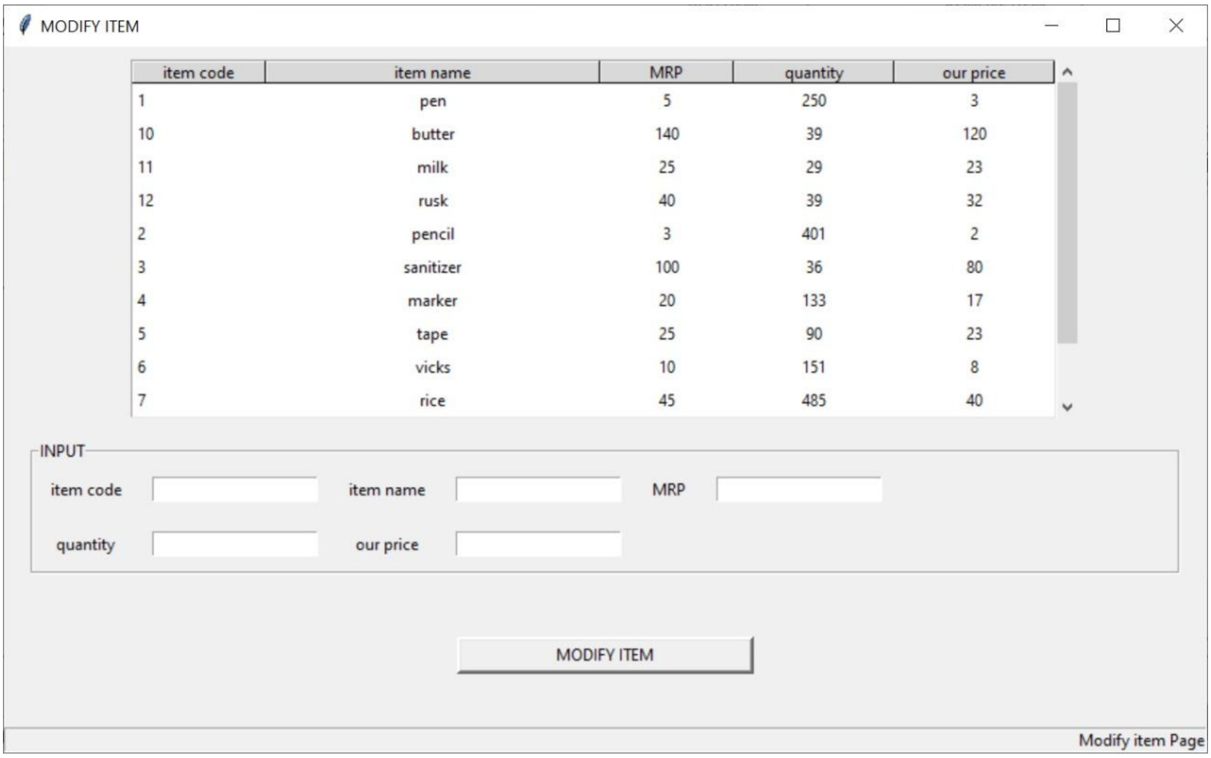
item code	item name	MRP	quantity	our price
1	pen	5	250	3
10	butter	140	39	120
11	milk	25	29	23
12	rusk	40	39	32
2	pencil	3	401	2
3	sanitizer	100	36	80
4	marker	20	133	17
5	tape	25	90	23
6	vicks	10	151	8
7	rice	45	485	40

Remove Item Button:-



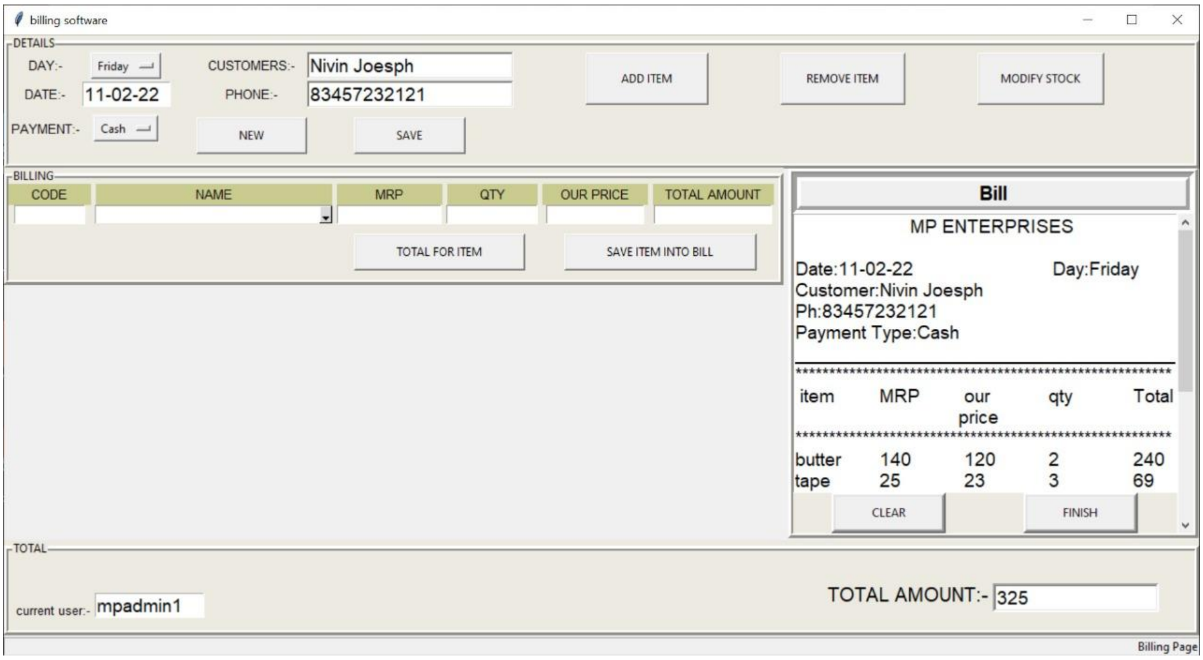
item code	item name	MRP	quantity	our price
1	pen	5	250	3
10	butter	140	39	120
11	milk	25	29	23
12	rusk	40	39	32
2	pencil	3	401	2
3	sanitizer	100	36	80
4	marker	20	133	17
5	tape	25	90	23
6	vicks	10	151	8
7	rice	45	485	40

Modify Stock Button:-



item code	item name	MRP	quantity	our price
1	pen	5	250	3
10	butter	140	39	120
11	milk	25	29	23
12	rusk	40	39	32
2	pencil	3	401	2
3	sanitizer	100	36	80
4	marker	20	133	17
5	tape	25	90	23
6	vicks	10	151	8
7	rice	45	485	40

Sample Entry:-



DAY:- Friday

DATE:- 11-02-22

CUSTOMERS:- Nivin Joesph

PHONE:- 83457232121

PAYMENT:- Cash

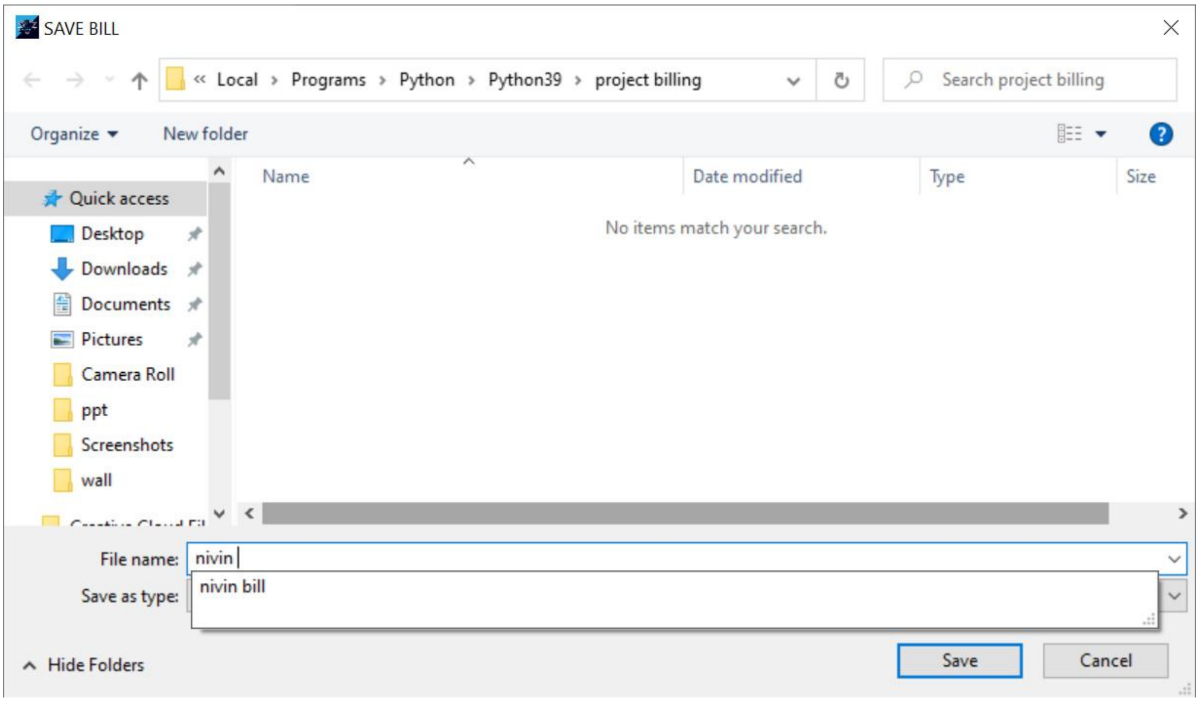
CODE	NAME	MRP	QTY	OUR PRICE	TOTAL AMOUNT

```
MP ENTERPRISES

Date:11-02-22                Day:Friday
Customer:Nivin Joesph
Ph:83457232121
Payment Type:Cash
___________________________________________
*******************************************
item    MRP     our     qty     Total
                price
*******************************************
butter  140     120     2       240
tape    25      23      3       69
```

current user:- mpadmin1

TOTAL AMOUNT:- 325

Saving Sample Entry:-



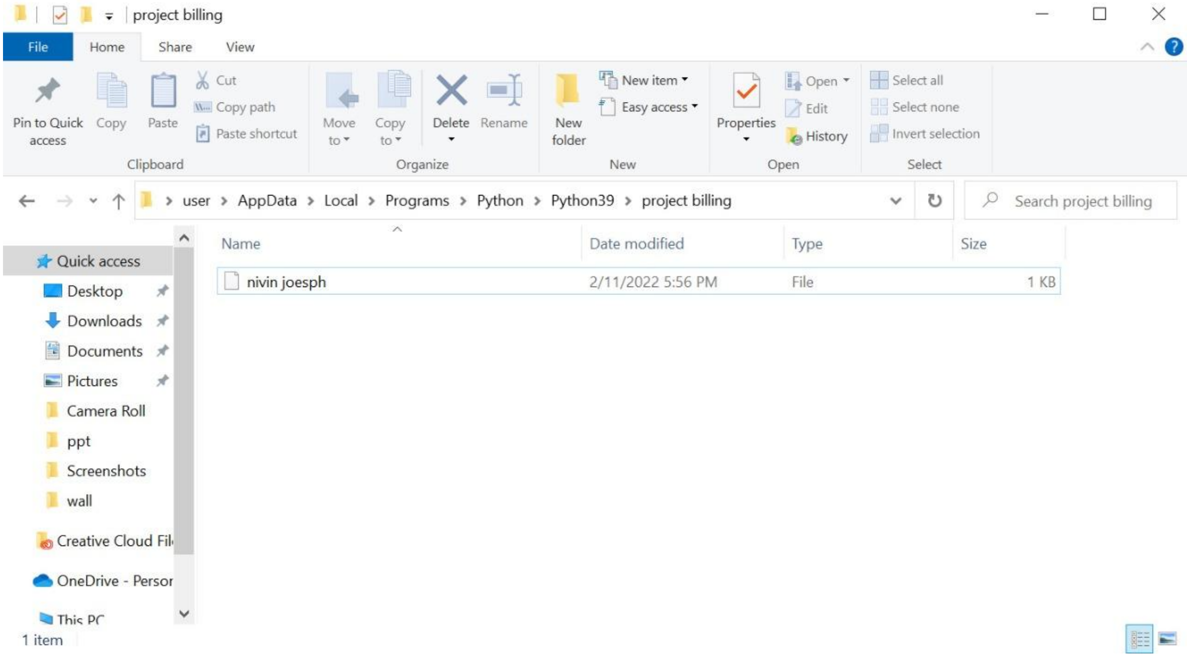
No items match your search.

File name: nivin

nivin bill

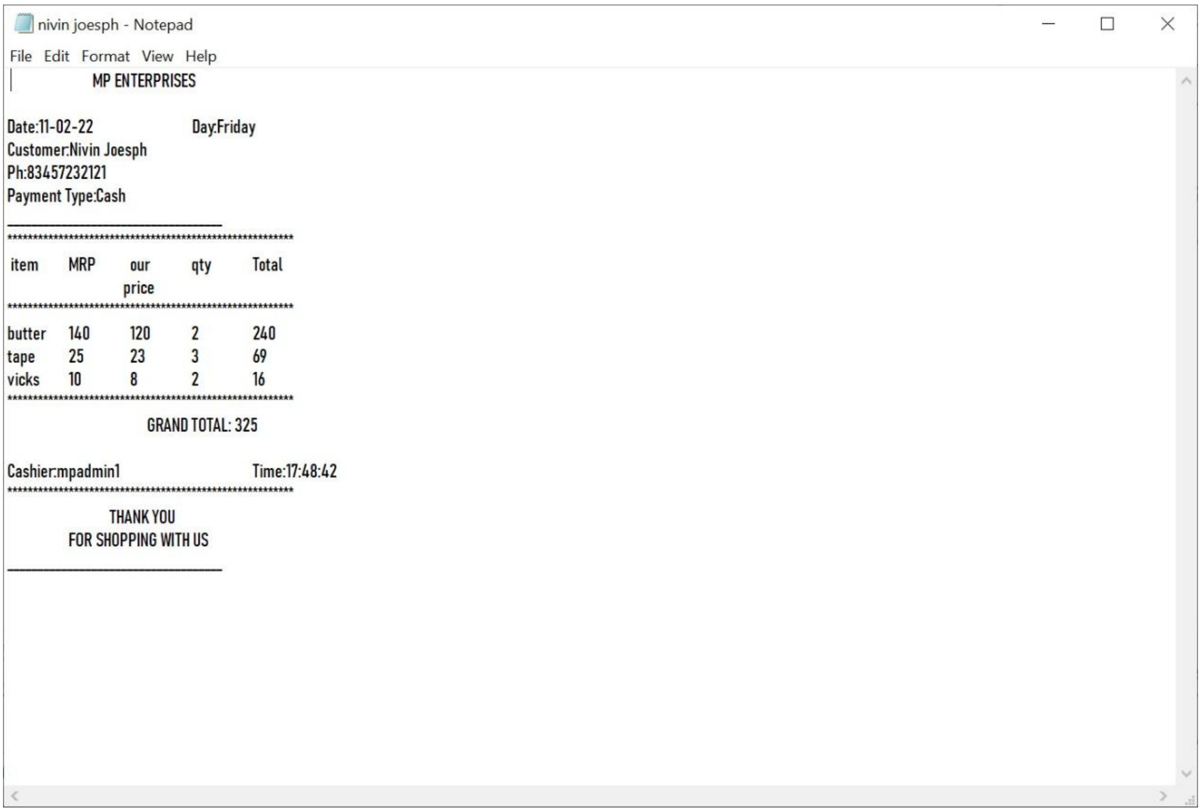
Save as type:

Searching Sample Entry File:-



Name	Date modified	Type	Size
nivin joesph	2/11/2022 5:56 PM	File	1 KB

Opening the Bill In Notepad (can be used to print the hard copy of the bill):-



```
            MP ENTERPRISES

Date:11-02-22              Day:Friday
Customer:Nivin Joesph
Ph:83457232121
Payment Type:Cash
________________________________________
************************************************************
 item    MRP     our      qty      Total
                 price
************************************************************
butter   140     120      2        240
tape     25      23       3        69
vicks    10      8        2        16
************************************************************
                     GRAND TOTAL: 325

Cashier:mpadmin1                   Time:17:48:42
************************************************************
                THANK YOU
           FOR SHOPPING WITH US
________________________________________
```

BIBLIOGRAPHY

1)Codemy.com → Youtube Channel

2)GeekforGeeks → Website

3)Stack Overflow → Website

4)Computer Science with Python by Sumita Arora → Book

5)Computer Science with Python by Preeti Arora → Book

6)Programcreek → Website

7)Kite → Website

8)Coderslegacy → Website

9)freeCodeCamp → Youtube Channel