from tkinter import \*

from tkinter import Tk

from tkinter import messagebox

from tkinter import font

from turtle import bgcolor

from PIL import Image

from PIL import ImageTk

from calendar import \*

import smtplib

import sqlite3

import time

import datetime

from datetime import datetime

from datetime import timedelta

from datetime import date

db=sqlite3.connect('admin.db')

dbstore=sqlite3.connect('StoreBooks.db')

dbstudents=sqlite3.connect('StudentsData.db')

root = Tk()

root.title("Library Management System")

root.iconbitmap('filename.ico')

root.geometry("900x500+50+100")

root.resizable(0, 0)

class main:

    def login(self):

        self.var1 = self.e1.get()

        self.var2 = self.e2.get()

        cursor=db.cursor()

        cursor.execute("SELECT \* FROM UserLogin WHERE UserID='"+self.var1+"' and Password='"+self.var2+"'")

        db.commit()

        self.ab = cursor.fetchone()

        if self.ab!=None:

            self.under\_fm=Frame(root,height=500,width=900,bg='#fff')

            self.under\_fm.place(x=0,y=0)

            self.fm2=Frame(root,bg='#012727',height=80,width=900)

            self.fm2.place(x=0,y=0)

            self.lbb=Label(self.fm2,bg='#012727')

            self.lbb.place(x=15,y=5)

            self.ig=PhotoImage(file='filename.png')

            self.lbb.config(image=self.ig)

            self.lb3=Label(self.fm2,text='DASHBOARD',fg='White',bg='#012727',font=('times new roman',30,'bold'))

            self.lb3.place(x=325,y=17)

            self.name=Label(root,text="Name : ",bg='#fff',fg="black",font=('Calibri',12,'bold'))

            self.name.place(x=5,y=83)

            self.name1=Label(root,text=self.ab[0],fg='black',bg='#fff',font=('Calibri',12,'bold'))

            self.name1.place(x=60,y=83)

            self.today=date.today()

            self.dat=Label(root,text='Date : ',bg='#fff',fg='black',font=('Calibri',12,'bold'))

            self.dat.place(x=750,y=83)

            self.dat2 = Label(root, text=self.today, bg='#fff', fg='black', font=('Calibri', 12, 'bold'))

            self.dat2.place(x=800, y=83)

            self.cur()

        else:

            messagebox.showerror('Library System', 'Your ID or Password is invalid!')

    def cur(self):

        self.fm3=Frame(root,bg='#fff',width=900,height=390)

        self.fm3.place(x=0,y=110)

        def clock():

            h = str(time.strftime("%H"))

            m = str(time.strftime("%M"))

            s = str(time.strftime("%S"))

            if int(h) >=12 and int(m) >=0:

                self.lb7\_hr.config(text="PM")

            self.lb1\_hr.config(text=h)

            self.lb3\_hr.config(text=m)

            self.lb5\_hr.config(text=s)

            self.lb1\_hr.after(200, clock)

        self.lb1\_hr = Label(self.fm3, text='12', font=('times new roman', 20, 'bold'), bg='#581845', fg='white')

        self.lb1\_hr.place(x=607, y=0, width=60, height=30)

        self.lb3\_hr = Label(self.fm3, text='05', font=('times new roman', 20, 'bold'), bg='#581845', fg='white')

        self.lb3\_hr.place(x=677, y=0, width=60, height=30)

        self.lb5\_hr = Label(self.fm3, text='37', font=('times new roman', 20, 'bold'), bg='#581845', fg='white')

        self.lb5\_hr.place(x=747, y=0, width=60, height=30)

        self.lb7\_hr = Label(self.fm3, text='AM', font=('times new roman', 17, 'bold'), bg='#581845', fg='white')

        self.lb7\_hr.place(x=817, y=0, width=60, height=30)

        clock()

        self.canvas8 = Canvas(self.fm3, bg='black', width=400, height=300)

        self.canvas8.place(x=475, y=40)

        self.photo9=PhotoImage(file="filename.png")

        self.canvas8.create\_image(0,0,image=self.photo9,anchor=NW)

        self.bt1=Button(self.fm3,text='  Add Books',fg='#fff',bg='#581845',font=('Candara',15,'bold'),width=170,height=0,bd=7,relief='flat',command=self.addbook,cursor='hand2',activebackground='black',activeforeground='#581845')

        self.bt1.place(x=40,y=40)

        self.logo = PhotoImage(file='bt1.png')

        self.bt1.config(image=self.logo, compound=LEFT)

        self.small\_logo = self.logo.subsample(1,1)

        self.bt1.config(image=self.small\_logo)

        self.bt2 = Button(self.fm3, text='  Issue Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0, bd=7,relief='flat',command=self.issuebook,cursor='hand2',activebackground='black',activeforeground='#581845')

        self.bt2.place(x=250, y=40)

        self.log = PhotoImage(file='bt2.png')

        self.bt2.config(image=self.log, compound=LEFT)

        self.small\_log = self.log.subsample(1, 1)

        self.bt2.config(image=self.small\_log)

        self.bt3 = Button(self.fm3, text='  Edit Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0,bd=7,relief='flat',cursor='hand2',command=self.edit,activebackground='black',activeforeground='#581845')

        self.bt3.place(x=40, y=120)

        self.logb = PhotoImage(file='bt3.png')

        self.bt3.config(image=self.logb, compound=LEFT)

        self.small\_logb = self.logb.subsample(1, 1)

        self.bt3.config(image=self.small\_logb)

        self.bt4 = Button(self.fm3, text='  Return Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0,bd=7,relief='flat',cursor='hand2',command=self.returnbook,activebackground='black',activeforeground='#581845')

        self.bt4.place(x=250, y=120)

        self.log4 = PhotoImage(file='bt4.png')

        self.bt4.config(image=self.log4, compound=LEFT)

        self.small\_log4 = self.log4.subsample(1, 1)

        self.bt4.config(image=self.small\_log4)

        self.bt5 = Button(self.fm3, text=' Delete Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0,bd=7,relief='flat',cursor='hand2',command=self.delete,activebackground='black',activeforeground='#581845')

        self.bt5.place(x=40, y=200)

        self.log5 = PhotoImage(file='bt5.png')

        self.bt5.config(image=self.log5, compound=LEFT)

        self.small\_log5 = self.log5.subsample(1, 1)

        self.bt5.config(image=self.small\_log5)

        self.bt6 = Button(self.fm3, text=' Show Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0,bd=7, relief='flat',cursor='hand2',command=self.show,activebackground='black',activeforeground='#581845')

        self.bt6.place(x=40, y=280)

        self.log6 = PhotoImage(file='bt6.png')

        self.bt6.config(image=self.log6, compound=LEFT)

        self.small\_log6 = self.log6.subsample(1, 1)

        self.bt6.config(image=self.small\_log6)

        self.bt7 = Button(self.fm3, text='  Search Books', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0,bd=7, relief='flat',cursor='hand2',command=self.search,activebackground='black',activeforeground='#581845')

        self.bt7.place(x=250, y=200)

        self.log7 = PhotoImage(file='bt7.png')

        self.bt7.config(image=self.log7, compound=LEFT)

        self.small\_log7 = self.log7.subsample(1, 1)

        self.bt7.config(image=self.small\_log7)

        try:

            self.bt8 = Button(self.fm3, text='  Log Out', fg='#fff', bg='#581845', font=('Candara', 15, 'bold'),width=170,height=0, bd=7, relief='flat',cursor='hand2',command=self.code,activebackground='black',activeforeground='#581845')

            self.bt8.place(x=250, y=280)

            self.log8 = PhotoImage(file='bt8.png')

            self.bt8.config(image=self.log8, compound=LEFT)

            self.small\_log8 = self.log8.subsample(1, 1)

            self.bt8.config(image=self.small\_log8)

        except:

            self.bt9 = Tk.Button(self.fm3, text="Ram", bg='#a40000', font=('Candara', 15, 'bold'), width=150,height=0)

            self.bt9.place(x=40, y=350)

            self.log9 = PhotoImage(file='bt8.png')

            self.bt9.config(image=self.log9, compound=LEFT)

            self.small\_log9 = self.log9.subsample(3, 3)

            self.bt9.config(image=self.small\_log9)

    def addbook(self):

        class temp(main):

            def book(self):

                self.fm=Frame(root,bg='#ffe8ec',width=900,height=390)

                self.fm.place(x=0,y=110)

                self.fm1=Frame(self.fm,bg='#ffe8ec',width=500,height=360,bd=5,relief='flat')

                self.fm1.place(x=200,y=15)

                self.backbt = Button(self.fm, width=60, bg='#ffe8ec', bd=0, relief='flat',command=self.cur,activeforeground='black',activebackground='#ffe8ec')

                self.backbt.place(x=2, y=7)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2, 2)

                self.backbt.config(image=self.small\_log)

                self.fll=Frame(self.fm1,width=150,height=40,bg='#ff6690')

                self.fll.place(x=150,y=15)

                self.ll=Label(self.fll,text='ADD BOOKS',fg='#fff',bg='#ff6690',font=('Canara',12,'bold'),width=15)

                self.ll.config(height=5)

                self.ll.place(x=0,y=8)

                self.lb=Label(self.fm1,text='ID',fg='black',bg='#ffe8ec',font=('times new roman',11,'bold'))

                self.lb.place(x=70,y=90)

                self.lb2 = Label(self.fm1, text='Title', fg='black', bg='#ffe8ec', font=('times new roman', 11, 'bold'))

                self.lb2.place(x=70, y=130)

                self.lb3 = Label(self.fm1, text='Author', fg='black', bg='#ffe8ec', font=('times new roman', 11, 'bold'))

                self.lb3.place(x=70, y=170)

                self.lb4= Label(self.fm1, text='Edition', fg='black', bg='#ffe8ec', font=('times new roman', 11, 'bold'))

                self.lb4.place(x=70, y=210)

                self.lb5 = Label(self.fm1, text='Price', fg='black', bg='#ffe8ec', font=('times new roman', 11, 'bold'))

                self.lb5.place(x=70, y=250)

                self.ee1=Entry(self.fm1,width=25,bd=4,relief='groove',font=('Calibri',11,'bold'))

                self.ee1.place(x=180,y=88)

                self.ee2=Entry(self.fm1,width=25,bd=4,relief='groove',font=('Calibri',11,'bold'))

                self.ee2.place(x=180,y=130)

                self.ee3=Entry(self.fm1,width=25,bd=4,relief='groove',font=('Calibri',11,'bold'))

                self.ee3.place(x=180,y=170)

                self.ee4=Entry(self.fm1,width=25,bd=4,relief='groove',font=('Calibri',11,'bold'))

                self.ee4.place(x=180,y=210)

                self.ee5=Entry(self.fm1,width=25,bd=4,relief='groove',font=('Calibri',11,'bold'))

                self.ee5.place(x=180,y=250)

                self.bt=Button(self.fm1,text='SUBMIT',width=8,fg='white',bg='#ff6690',font=('Canara',12,'bold'),bd=3,relief='flat',command=self.submit1,activebackground='black',activeforeground='#ff6690')

                self.bt.place(x=70,y=300)

            def submit1(self):

                try:

                    self.id=self.ee1.get()

                    self.ttl=self.ee2.get()

                    self.aut=self.ee3.get()

                    self.edi=self.ee4.get()

                    self.pri=self.ee5.get()

                    if(self.id and self.ttl and self.aut and self.edi and self.pri):

                            cursor=dbstore.cursor()

                            cursor.execute("INSERT INTO Books(BookID,Title,Author,Edition,Price) values(?,?,?,?,?)",(self.id, self.ttl,self.aut,self.edi,self.pri))

                            dbstore.commit()

                            messagebox.showinfo("Success","Book has been added to the library succesfully")

                            self.clear()

                    else:

                            messagebox.showerror("Error", "Enter Valid Details")

                except Exception as e:

                    messagebox.showerror("Error", "Enter Valid Details")

            def clear(self):

                self.ee1.delete(0,END)

                self.ee2.delete(0,END)

                self.ee3.delete(0,END)

                self.ee4.delete(0,END)

                self.ee5.delete(0,END)

        obj=temp()

        obj.book()

    def issuebook(self):

        class test(main):

            max=0

            n = 1

            def issue(self):

                self.f = Frame(root, bg='#ffe8ec', width=900, height=390)

                self.f.place(x=0, y=110)

                self.fmi=Canvas(self.f,bg='#ffe8ec',width=900,height=390,bd=0,relief='flat')

                self.fmi.place(x=0,y=0)

                self.fc=Frame(self.fmi,bg='#ffe8ec',width=338,height=230,bd=4,relief='flat')

                self.fc.place(x=70,y=20)

                self.ffbll=Frame(self.fc,bg='#00203f', bd=2,relief='flat', width=210,height=40)

                self.ffbll.place(x=50,y=0)

                self.lc=Label(self.ffbll,text='STUDENT  INFORMATION',bg='#00203f',fg='#adefd1',font=('Arial',12,'bold'))

                self.lc.place(x=0,y=6)

                self.lb = Label(self.fc, text='ERP ID', bg='#ffe8ec', fg='black', font=('times new roman', 11, 'bold'))

                self.lb.place(x=15, y=90)

                self.em2 = Entry(self.fc, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

                self.em2.place(x=105, y=90)

                self.bt = Button(self.fc, text='SUBMIT', width=8, bg='#00203f', fg='#adefd1', font=('Canara', 12, 'bold'),bd=5,relief='flat',command=self.check, activeforeground='#00203f',activebackground='#adefd1')

                self.bt.place(x=15,y=160)

                self.backbt = Button(self.fmi,width=60, bg='#ffe8ec',activebackground='#ffe8ec',bd=0, relief='flat',command=self.issueback)

                self.backbt.place(x=5, y=5)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2, 2)

                self.backbt.config(image=self.small\_log)

            def check(self):

                self.b=self.em2.get()

                cursor=dbstudents.cursor()

                cursor.execute("SELECT \* FROM Students WHERE ERP='"+self.b+"'")

                self.var=cursor.fetchone()

                if self.var!=None:

                    self.fmii=Canvas(self.f,bg='#ffe8ec',width=338,height=90,bd=0,relief='flat')

                    self.fmii.place(x=70,y=255)

                    self.lb1=Label(self.fmii,text='Name :',fg='black',bg ='#ffe8ec',font=('Calibri',12,'bold'))

                    self.lb1.place(x=5,y=5)

                    self.lb2 = Label(self.fmii, text=self.var[1],fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb2.place(x=70, y=5)

                    self.lb3 = Label(self.fmii, text='Course :',fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb3.place(x=5, y=25)

                    self.lb4 = Label(self.fmii, text=self.var[2],fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb4.place(x=70, y=25)

                    self.lb5 = Label(self.fmii, text='Year :', fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb5.place(x=5, y=45)

                    self.lb6 = Label(self.fmii, text=self.var[3], fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb6.place(x=70, y=45)

                    self.lb7 = Label(self.fmii, text='Contact :', fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb7.place(x=5, y=65)

                    self.lb8 = Label(self.fmii, text=self.var[6],fg='black',bg ='#ffe8ec', font=('Calibri', 12, 'bold'))

                    self.lb8.place(x=70, y=65)

                    self.fr=Frame(self.fmi,bg='#ffe8ec',bd=5,relief='flat',width=338,height=250)

                    self.fr.place(x=420,y=20)

                    self.ff=Frame(self.fr,bg='#adefd1',bd=2,relief='flat',width=140,height=40)

                    self.ff.place(x=80,y=0)

                    self.lb=Label(self.ff,text='ISSUE BOOK',bg='#adefd1',fg='#00203f',font=('Arial',12,'bold'))

                    self.lb.place(x=13,y=5)

                    self.tt=Label(self.fr,text='BOOK ID',bg='#ffe8ec',fg='#00203f',font=('times new roman',11,'bold'))

                    self.tt.place(x=30,y=90)

                    self.e1 = Entry(self.fr, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

                    self.e1.place(x=130, y=90)

                    self.bt1 = Button(self.fr, text='SUBMIT', width=8, bg='#adefd1', fg='#00203f', font=('Canara', 12,'bold'),bd=5,relief='flat',command=self.data,activeforeground='#adefd1',activebackground='#00203f')

                    self.bt1.place(x=15, y=160)

                else:

                    messagebox.showwarning('Warning','This student is not registered !')

                    self.em2.delete(0,END)

            def issueback(self):

                try:

                    self.boot.destroy()

                    self.cur()

                except Exception as e:

                    self.cur()

            repeat=0

            def data(self):

                self.b=self.em2.get()

                cursor=dbstudents.cursor()

                cursor.execute("SELECT \* FROM Students WHERE ERP='"+self.b+"'")

                self.var=cursor.fetchone()

                self.flag=0

                if(int(self.var[11])>=3):

                    try:

                        self.boot.destroy()

                        messagebox.showerror("Unable to process request","You exceed the limit of Books per student!")

                        self.flag=1

                        self.cur()

                    except Exception as e:

                        messagebox.showerror("Unable to process request","You exceed the limit of Books per student!")

                        self.flag=1

                        self.cur()

                self.vva=self.e1.get()

                cursor=dbstore.cursor()

                cursor.execute("SELECT \* FROM Books WHERE BookID='"+self.vva+"'")

                dbstore.commit()

                self.value=cursor.fetchone()

                if self.value!=None:

                    if(self.flag!=1):

                        self.boot=Tk()

                        self.boot.title("Issue Books")

                        self.boot.iconbitmap("filename.ico")

                        self.boot.configure(bg='#ffe8ec')

                        self.boot.geometry("370x450+880+30")

                        self.boot.resizable(0,0)

                        test.repeat=1

                        self.lb=Label(self.boot,text='Title:',bg='#ffe8ec',fg='black',font=('Calibri',12,'bold'))

                        self.lb.place(x=30,y=30)

                        self.lbn = Label(self.boot, text=self.value[1], bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                        self.lbn.place(x=120,y=30)

                        self.lb = Label(self.boot, text='Author:', bg='#ffe8ec', fg='black', font=('Calibri', 12,'bold'))

                        self.lb.place(x=30, y=60)

                        self.lbn = Label(self.boot, text=self.value[2], bg='#ffe8ec', fg='black', font=('Calibri', 12,'bold'))

                        self.lbn.place(x=120, y=60)

                        self.lb = Label(self.boot, text='Edition:', bg='#ffe8ec', fg='black', font=('Calibri', 12,'bold'))

                        self.lb.place(x=30, y=90)

                        self.lbn = Label(self.boot, text=self.value[3], bg='#ffe8ec', fg='black', font=('Calibri', 12,'bold'))

                        self.lbn.place(x=120, y=90)

                        self.label = Label(self.fr, text='ADD MORE BOOKS ', bg='#ffe8ec', fg='black', font=('times new romman', 11,'bold'))

                        self.label.place(x=15, y=220)

                        self.it1=Radiobutton(self.fr,text='YES',bg='#ffe8ec',variable='radio',value=1,command=self.yes)

                        self.it1.place(x=170,y=220)

                        self.it2 = Radiobutton(self.fr, text='NO',bg='#ffe8ec', variable='radio', value=2,command=self.no)

                        self.it2.place(x=240, y=220)

                        self.button1 = Button(self.boot, text='ISSUE', bg='#adefd1', fg='#00203f', width=10, height=0,font=('Canara', 11, 'bold'), activebackground='#00203f',activeforeground='#adefd1',command=self.issued)

                        self.button1.place(x=30, y=400)

                        self.btn = Button(self.boot, text='SEND MAIL', bg='#adefd1', fg='#00203f', width=10, height=0,font=('Canara', 11, 'bold'),activebackground='#00203f',activeforeground='#adefd1', command=self.mail)

                        self.btn.place(x=160, y=400)

                        self.x = date.today()

                        self.cal = Calendar(self.boot, selectmode="day", bg='black',year=2023,month=5,day=10)

                        self.cal.place(x=20,y=150)

                        btn1 = Button(self.boot, text="CONFIRM DATE",command=self.get\_data,  bg='#343148',font=('Canara', 11,'bold'),fg='#d7c49e',activebackground='black', activeforeground='#d7c49e', relief='flat')

                        btn1.place(x=90,y=350)

                        self.boot.mainloop()

                else:

                    messagebox.showerror('Book Not Found','No such book exists!')

                    self.e1.delete(0,END)

            def get\_data(self):

                self.datecon=self.cal.selection\_get()

            def yes(self):

                self.n=self.n+1

                self.bt1 = Button(self.fr, text='SUBMIT', width=8, bg='#adefd1', fg='#00203f', font=('Canara', 12,'bold'),bd=5,relief='flat',command=self.data,activeforeground='#adefd1',activebackground='#00203f',state=ACTIVE)

                self.bt1.place(x=15, y=160)

                self.e1.delete(0, END)

                self.e2.delete(0, END)

                self.max=self.max-1

            def no(self):

                self.bt1 = Button(self.fr, text='SUBMIT', width=8, bg='#adefd1', fg='#00203f', font=('Canara', 12,'bold'),bd=5,relief='flat',command=self.data,activeforeground='#adefd1',activebackground='#00203f',state=DISABLED)

                self.bt1.place(x=15, y=160)

            def issued(self):

                self.datecon=self.cal.selection\_get()

                self.ac=self.e1.get()

                cursor=dbstore.cursor()

                cursor.execute("UPDATE Books SET Issue='Issued', ID='"+self.b+"' WHERE BookID='"+self.ac+"'")

                dbstore.commit()

                if self.n<=3:

                    book=dbstudents.cursor()

                    self.erpid1=self.em2.get()

                    book.execute("SELECT \* FROM Students WHERE ERP='"+self.erpid1+"'")

                    self.issuevar=book.fetchone()

                    self.sum=self.issuevar[11]+1

                    book.execute("UPDATE Students SET NoBook='"+str(self.sum)+"' WHERE ERP='"+self.b+"' ")

                    dbstudents.commit()

                comm=dbstudents.cursor()

                comm.execute("UPDATE Students SET FromDate='"+str(self.x)+"', ToDate='"+str(self.datecon)+"' , SubmitDate='' WHERE ERP='"+self.b+"'")

                dbstudents.commit()

                messagebox.showinfo('Library Management System', 'YOUR BOOK HAS BEEN ISSUED')

                self.boot.destroy()

                self.e1.delete(0, END)

            def mail(self):

                self.erpid=self.em2.get()

                cursor=dbstudents.cursor()

                cursor.execute("SELECT \* FROM Students WHERE ERP='"+self.erpid+"'")

                self.var=cursor.fetchone()

                sender = "libraryauthority@gmail.com"

                reciever =self.var[5]

                with open("pass.txt",'r') as file:

                        password=file.read()

                message = """FROM: LIBRARY DEPARTMENT

                          TO : Library Issued Books Department

                          Subject: Hello Student! Your book has been Issued"""

                try:

                    server = smtplib.SMTP\_SSL("smtp.gmail.com", 465)

                    server.login(sender, password)

                    server.sendmail(sender, reciever, message)

                    print("ok")

                    messagebox.showinfo("Library System","Send mail Successfully !")

                except Exception as e:

                    pass

        obissue=test()

        obissue.issue()

    def edit(self):

        class editing(main):

            def edbooks(self):

                self.ffm=Frame(root,bg='#ffe8ec',width=900,height=390)

                self.ffm.place(x=0,y=110)

                self.fm1 = Frame(self.ffm, bg='#ffe8ec', width=500, height=200, bd=5, relief='flat')

                self.fm1.place(x=150, y=30)

                self.ed = Frame(self.fm1, bg='#1c1c1b', bd=0, relief='flat', width=160, height=35)

                self.ed.place(x=170,y=0)

                self.lab = Label(self.ed, text='EDIT BOOK DETAILS', bg='#1c1c1b', fg='#ce4a7e', font=('Calibri', 12,'bold'))

                self.lab.place(x=9, y=5)

                self.label3=Label(self.fm1,text='Book ID',bg='#ffe8ec',fg='black',font=('Times New Roman',11,'bold'))

                self.label3.place(x=85,y=65)

                self.entry=Entry(self.fm1,width=30,bd=4,relief='groove',font=('Calibri',8,'bold'))

                self.entry.place(x=188,y=65)

                self.button7 = Button(self.fm1, text='SEARCH', bg='#1c1c1b', fg='#ce4a7e', width=8,font=('Calibri', 12, 'bold'),command=self.searchedit ,relief='flat',activebackground='#ce4a7e',activeforeground='#1c1c1b')

                self.button7.place(x=85,y=125)

                self.backbt = Button(self.ffm, width=60, bg='#ffe8ec',activebackground='#ffe8ec',bd=0, relief='flat', command=self.cur)

                self.backbt.place(x=0, y=0)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2, 2)

                self.backbt.config(image=self.small\_log)

            def searchedit(self):

                self.datas=self.entry.get()

                cursor=dbstore.cursor()

                cursor.execute("SELECT \* FROM Books WHERE BookID = '"+self.datas+"'" )

                dbstore.commit()

                self.val=cursor.fetchone()

                if self.val!=None:

                    self.edcat=Tk()

                    self.edcat.title("Library System")

                    self.edcat.geometry("300x360+600+230")

                    self.edcat.configure(bg='#ffe8ec')

                    self.edcat.iconbitmap("filename.ico")

                    self.fc=Frame(self.edcat,bg='#1c1c1b',width=90,height=30)

                    self.fc.place(x=80,y=10)

                    self.lab=Label(self.fc,bg='#1c1c1b',fg='#ce4a7e',text='EDIT BOOK',font=('Calibri',12,'bold'))

                    self.lab.place(x=3,y=3)

                    self.labid = Label(self.edcat, bg='#ffe8ec', fg='black', text='Book ID:', font=('Calibri', 12,'bold'))

self.labid.place(x=30, y=60)

                    self.labti = Label(self.edcat, bg='#ffe8ec', fg='black', text='Title:', font=('Calibri', 12, 'bold'))

                    self.labti.place(x=30, y=100)

                    self.labaut = Label(self.edcat, bg='#ffe8ec', fg='black', text='Author:', font=('Calibri', 12,'bold'))

                    self.labaut.place(x=30, y=140)

                    self.labed = Label(self.edcat, bg='#ffe8ec', fg='black', text='Edition:', font=('Calibri', 12,'bold'))

                    self.labed.place(x=30, y=180)

                    self.labpr = Label(self.edcat, bg='#ffe8ec', fg='black', text='Price:', font=('Calibri', 12,'bold'))

                    self.labpr.place(x=30, y=220)

                    self.en1=Entry(self.edcat,width=20,bd=4,relief='groove',font=('Times New Roman',9,'bold'))

                    self.en1.place(x=110,y=60)

                    self.en2 = Entry(self.edcat, width=20, bd=4, relief='groove',font=('Times New Roman',9,'bold'))

                    self.en2.place(x=110, y=100)

                    self.en3 = Entry(self.edcat, width=20, bd=4, relief='groove',font=('Times New Roman',9,'bold'))

                    self.en3.place(x=110, y=140)

                    self.en4 = Entry(self.edcat, width=20, bd=4, relief='groove',font=('Times New Roman',9,'bold'))

                    self.en4.place(x=110, y=180)

                    self.en5 = Entry(self.edcat, width=20, bd=4, relief='groove',font=('Times New Roman',9,'bold'))

                    self.en5.place(x=110, y=220)

                    self.butt = Button(self.edcat, text='SUBMIT', bg='#1c1c1b', fg='#ce4a7e', width=8,font=('Calibri', 12, 'bold'),command=self.savedit,relief='flat')

                    self.butt.place(x=30, y=273)

                    self.en1.insert(0, self.val[0])

                    self.en2.insert(0, self.val[1])

                    self.en3.insert(0, self.val[2])

                    self.en4.insert(0, self.val[3])

                    self.en5.insert(0, self.val[4])

                    self.edcat.mainloop()

                else:

                    messagebox.showerror('Invalid Entry',"This Book doesn't exists!")

                    self.entry.delete(0,END)

            def savedit(self):

                self.id = self.en1.get()

                self.ti = self.en2.get()

                self.au = self.en3.get()

                self.ed = self.en4.get()

                self.pi = self.en5.get()

                if(self.id and self.ti and self.au and self.ed and self.pi):

                    cursor= dbstore.cursor()

                    cursor.execute("UPDATE Books SET BookID='"+self.id+"', Title='"+self.ti+"',Author='"+self.au+"',Edition='"+self.ed+"',Price='"+self.pi+"' WHERE BookID='"+self.datas+"'")

                    dbstore.commit()

                    messagebox.showinfo('Changes Saved','Data has been updated successfully!')

                    self.edcat.destroy()

                    self.entry.delete(0,END)

                else:

                    messagebox.showerror('Error','Enter Valid Details')

                    self.entry.delete(0,END)

        obj=editing()

        obj.edbooks()

    def returnbook(self):

        class retu(main):

            def \_\_init\_\_(self):

                self.frame=Frame(root,bd=0,relief='flat',bg='#ffe8ec',width=900,height=390)

                self.frame.place(x=0,y=110)

                self.f1 = Frame(self.frame, bg='#ffe8ec', width=500, height=200, bd=5, relief='flat')

                self.f1.place(x=200, y=15)

                self.ed = Frame(self.f1, bg='#581845', bd=0, relief='flat', width=130, height=35)

                self.ed.place(x=170, y=0)

                self.lac = Label(self.ed, text='RETURN BOOKS ', bg='#581845', fg='#fff', font=('Calibri', 12, 'bold'))

                self.lac.place(x=10, y=5)

                self.label8 = Label(self.f1, text='ERP ID', bg='#ffe8ec', fg='black', font=('Times New Roman', 11, 'bold'))

                self.label8.place(x=85, y=65)

                self.entry4 = Entry(self.f1, width=30, bd=4, relief='groove', font=('Calibri', 8, 'bold'))

                self.entry4.place(x=188, y=65)

                self.button9 = Button(self.f1, text='RETURN', bg='#581845', fg='#fff', width=8, height=0,font=('Calibri', 12, 'bold'),command=self.retbook,activebackground="#000",activeforeground="#581845")

                self.button9.place(x=85, y=120)

                self.backbt = Button(self.frame, width=60, bg='#ffe8ec', activebackground='#ffe8ec',bd=0, relief='flat', command=self.cur)

                self.backbt.place(x=0, y=0)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2,2)

                self.backbt.config(image=self.small\_log)

            def retsucc(self):

                self.entry4.delete(0,END)

                cursor1 = dbstudents.cursor()

                cursor1.execute("UPDATE Students SET FromDate='',ToDate='',Charge='"+str(self.charge)+"' WHERE ERP='"+self.entry+"'")

                dbstudents.commit()

                messagebox.showinfo("Success","Charges Updated and Books Returned Succesfully")

                self.tom.destroy()

            def retbook(self):

                self.charge=0

                self.entry=self.entry4.get()

                cursor=dbstudents.cursor()

                cursor.execute("SELECT \* FROM Students WHERE ERP='"+self.entry+"'")

                dbstudents.commit()

                self.data=cursor.fetchone()

                if self.data!=None:

                    if(int(self.data[11])>=1):

                        self.get\_date = date.today()

                        cursor = dbstudents.cursor()

                        cursor.execute("UPDATE Students SET NoBook = 0, SubmitDate='" + str(self.get\_date) + "' WHERE ERP='" + self.entry + "'")

                        dbstudents.commit()

                        cursor=dbstore.cursor()

                        cursor.execute("UPDATE Books SET Issue='', ID='' WHERE ID='"+self.entry+"'")

                        dbstore.commit()

                        from datetime import datetime

                        cursor=dbstudents.cursor()

                        cursor.execute("SELECT \* FROM Students WHERE ERP='"+self.entry+"'")

                        dbstudents.commit()

                        self.var=cursor.fetchone()

                        if self.var!=None:

                            self.a=self.var[8]

                            self.b=self.var[9]

                            formatStr='%Y-%m-%d'

                            delta1=datetime.strptime(self.a,formatStr)

                            delta2=datetime.strptime(self.b, formatStr)

                            delta=delta2-delta1

                            chm=delta.days

                            if chm<=0:

                                messagebox.showinfo("Success","Books returned successfully")

                                self.entry4.delete(0,END)

                            else:

                                self.tom=Tk()

                                self.tom.geometry("300x150+300+258")

                                self.tom.iconbitmap("filename.ico")

                                self.tom.title("Library System")

                                self.tom.resizable(0,0)

                                self.tom.configure(bg="#ffe8ec")

                                self.lb=Label(self.tom,text="Name of Student: ",bg="#ffe8ec",fg="black",font=('Calibri',11,'bold'))

                                self.lb.place(x=5,y=20)

                                self.lb2=Label(self.tom,text=self.var[1],bg="#ffe8ec",fg="black",font=('Calibri',11,'bold'))

                                self.lb2.place(x=130,y=20)

                                self.charge=(5\*chm)+int(self.var[10])

                                self.lb3=Label(self.tom,text="Fine Charge: ",bg="#ffe8ec",fg="black",font=('Calibri',11,'bold'))

                                self.lb3.place(x=5,y=55)

                                self.lc2 = Label(self.tom, text=self.charge, bg="#ffe8ec", fg="black", font=('Calibri',11,'bold'))

                                self.lc2.place(x=130, y=55)

                                self.lc3 = Label(self.tom, text='Rs.', bg="#ffe8ec", fg="black",font=('Calibri', 11, 'bold'))

                                self.lc3.place(x=150, y=55)

                                self.tombtn = Button(self.tom,text='SUBMIT', background='#581845',foreground='white',font=('Calibri',12,'bold'),width=8,activebackground='black',activeforeground='#581845',relief='flat',command=self.retsucc)

                                self.tombtn.place(x=5,y=90)

                                self.tom.mainloop()

                            cursor1 = dbstudents.cursor()

                            cursor1.execute("UPDATE Students SET FromDate='',ToDate='',Charge='"+str(self.charge)+"' WHERE ERP='"+self.entry+"'")

                            dbstudents.commit()

                    else:

                        messagebox.showwarning("No Books Found","This student does not have any book issued!")

                        self.entry4.delete(0,END)

                else:

                    messagebox.showerror("Invalid ERP ID","This student doesn't exist!")

                    self.entry4.delete(0,END)

        object=retu()

    def delete(self):

        class dele(main):

            def deletebooks(self):

                self.ff = Frame(root, bg='#ffe8ec', width=900, height=390)

                self.ff.place(x=0, y=110)

                self.f1 = Frame(self.ff, bg='#ffe8ec', width=500, height=200, bd=5, relief='flat')

                self.f1.place(x=200, y=15)

                self.ed = Frame(self.f1, bg='#7ea310', bd=0, relief='flat', width=120, height=30)

                self.ed.place(x=150, y=0)

                self.lac = Label(self.ed, text='DELETE BOOKS ', bg='#7ea310', fg='#213502', font=('Calibri', 12,'bold'))

                self.lac.place(x=7, y=3)

                self.label8 = Label(self.f1, text='Book ID', bg='#ffe8ec', fg='black', font=('times new roman', 11, 'bold'))

                self.label8.place(x=85, y=65)

                self.entry4 = Entry(self.f1, width=30, bd=4, relief='groove', font=('Calibri', 8, 'bold'))

                self.entry4.place(x=188, y=65)

                self.button9 = Button(self.f1, text='DELETE', bg='#7ea310', fg='#213502', width=8,font=('Calibri', 12, 'bold'),command=self.deldata,relief='flat',activebackground='black',activeforeground='#7ea310')

                self.button9.place(x=85, y=120)

                self.backbt = Button(self.ff,width=60, bg='#ffe8ec',activebackground='#ffe8ec',bd=0, relief='flat', command=self.cur)

                self.backbt.place(x=0, y=0)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2,2)

                self.backbt.config(image=self.small\_log)

            def deldata(self):

                self.a=self.entry4.get()

                cursor=dbstore.cursor()

                cursorv=dbstore.cursor()

                cursorv.execute("SELECT \* FROM BOOKS WHERE BookID='"+self.a+"'")

                dbstore.commit()

                self.validation=cursorv.fetchone()

                if(self.validation!=None):

                    cursor.execute("DELETE FROM Books WHERE BookID='"+self.a+"'")

                    dbstore.commit()

                    messagebox.showinfo('Succesful','The book is successfully removed from the store!')

                    self.entry4.delete(0,END)

                else:

                    messagebox.showerror('Invalid Operation','This book does not exist!')

                    self.entry4.delete(0,END)

        occ=dele()

        occ.deletebooks()

    def search(self):

        class demt(main):

            def delmdata(self):

                self.fc = Frame(root, bg='#ffe8ec', width=900, height=390)

                self.fc.place(x=0, y=110)

                self.fc1 = Frame(self.fc, bg='#ffe8ec', width=500, height=200, bd=5, relief='flat')

                self.fc1.place(x=200, y=15)

                self.edm = Frame(self.fc1, bg='#b76e79', bd=0, relief='flat', width=130, height=35)

                self.edm.place(x=140, y=0)

                self.lac = Label(self.edm, text='SEARCH BOOKS ', bg='#b76e79', fg='#fff', font=('Calibri', 12, 'bold'))

                self.lac.place(x=8, y=5)

                self.label8 = Label(self.fc1, text='Book ID', bg='#ffe8ec', fg='black', font=('Times New Roman', 11, 'bold'))

                self.label8.place(x=85, y=65)

                self.entryl= Entry(self.fc1, width=30, bd=4, relief='groove', font=('Calibri', 8, 'bold'))

                self.entryl.place(x=188, y=65)

                self.butto = Button(self.fc1, text='SEARCH', bg='#b76e79', fg='#fff', width=8,font=('Calibri', 12, 'bold'),command=self.srch,relief='flat',activebackground='black',activeforeground='#b76e79')

                self.butto.place(x=85, y=120)

                self.backbt = Button(self.fc,width=60, bg='#ffe8ec',activebackground='#ffe8ec',bd=0, relief='flat', command=self.cur)

                self.backbt.place(x=0, y=0)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(2, 2)

                self.backbt.config(image=self.small\_log)

            def srch(self):

                self.emp=self.entryl.get()

                cursor=dbstore.cursor()

                cursor.execute("SELECT \* FROM Books WHERE BookID='"+self.emp+"'")

                dbstore.commit()

                self.srval=cursor.fetchone()

                if self.srval!=None:

                    self.top=Tk()

                    self.top.title("Library System")

                    self.top.iconbitmap("filename.ico")

                    self.top.geometry("400x200+335+250")

                    self.top.resizable(0, 0)

                    self.top.configure(bg='#ffe8ec')

                    self.frm=Frame(self.top,bg='#b76e79',width=100,height=35)

                    self.frm.place(x=100,y=10)

                    self.mnlb=Label(self.frm,bg='#b76e79',fg='#fff',text="AVAILABLE",font=('Calibri',12,'bold'))

                    self.mnlb.place(x=9,y=5)

                    self.lb1 = Label(self.top, text='Title: ', bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                    self.lb1.place(x=85,y=70)

                    self.lb2=Label(self.top,text=self.srval[1],bg='#ffe8ec', fg='black',font=('Calibri',12,'bold'))

                    self.lb2.place(x=165,y=70)

                    self.lb3 = Label(self.top, text='Author: ', bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                    self.lb3.place(x=85, y=110)

                    self.lb4 = Label(self.top, text=self.srval[2], bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                    self.lb4.place(x=165, y=110)

                    self.lb5 = Label(self.top, text='Edition: ',bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                    self.lb5.place(x=85, y=150)

                    self.lb6 = Label(self.top, text=self.srval[3], bg='#ffe8ec', fg='black', font=('Calibri', 12, 'bold'))

                    self.lb6.place(x=165, y=150)

                    self.entryl.delete(0,END)

                else:

                    messagebox.showwarning('Invalid Data','This book does not exists!')

                    self.entryl.delete(0,END)

        object=demt()

        object.delmdata()

    def show(self):

        class test(main):

            def \_\_init\_\_(self):

                self.fc = Frame(root, bg='#ffe8ec', width=900, height=390)

                self.fc.place(x=0, y=110)

                self.popframe=Frame(self.fc,width=180,height=30,bg='#edb40d')

                self.popframe.place(x=360,y=0)

                self.lbn=Label(self.popframe,bg='#edb40d',text='BOOKS INFORMATION',fg='#fff',font=('Calibri',12,'bold'))

                self.lbn.place(x=8,y=4)

                self.backbt = Button(self.fc,width=30, bg='#ffe8ec',activebackground='#ffe8ec',bd=0, relief='flat', command=self.cur)

                self.backbt.place(x=0, y=0)

                self.log = PhotoImage(file='filename.png')

                self.backbt.config(image=self.log, compound=LEFT)

                self.small\_log = self.log.subsample(3, 3)

                self.backbt.config(image=self.small\_log)

                self.table\_frame=Frame(self.fc,bg='#ffe8ec',bd=1,relief='flat')

                self.table\_frame.place(x=0,y=30,width=900,height=360)

                self.scroll\_x=Scrollbar(self.table\_frame,orient=HORIZONTAL)

                self.scroll\_y=Scrollbar(self.table\_frame,orient=VERTICAL)

                self.book\_table=Tk.Treeview(self.table\_frame,columns=("Book ID","Title","Author","Edition","Price"),xscrollcommand=self.scroll\_x.set,yscrollcommand=self.scroll\_y.set)

                self.scroll\_x.pack(side=BOTTOM,fill=X)

                self.scroll\_y.pack(side=RIGHT, fill=Y)

                self.scroll\_x.config(command=self.book\_table.xview)

                self.scroll\_y.config(command=self.book\_table.yview)

                self.book\_table.heading("Book ID",text="Book ID")

                self.book\_table.heading("Title", text="Title")

                self.book\_table.heading("Author", text="Author")

                self.book\_table.heading("Edition", text="Edition")

                self.book\_table.heading("Price", text="Price")

                self.book\_table['show']='headings'

                self.book\_table.column("Book ID",width=200)

                self.book\_table.column("Title", width=200)

                self.book\_table.column("Author", width=200)

                self.book\_table.column("Edition", width=120)

                self.book\_table.column("Price", width=110)

                self.book\_table.pack(fill=BOTH,expand=1)

                self.fetch\_data()

            def fetch\_data(self):

                cursor=dbstore.cursor()

                cursor.execute("SELECT \* FROM Books")

                self.rows=cursor.fetchall()

                if len(self.rows)!=0:

                        for self.row in self.rows:

                                self.book\_table.insert('',END,values=self.row)

                dbstore.commit()

        oc=test()

    def mainclear(self):

        self.e1.delete(0,END)

        self.e2.delete(0,END)

    def code(self):

            self.fm=Frame(root,height=500,width=900,bg='white')

            self.fm.place(x=0,y=0)

            self.canvas=Canvas(self.fm,height=500,width=900,bg='#000000')

            self.canvas.place(x=0,y=0)

            self.photo=PhotoImage(file=r"pathtoimage\filename.png")

            self.canvas.create\_image(0,0,image=self.photo,anchor=NW)

            self.fm1=Frame(self.canvas,height=260,width=300,bg='#000000',bd=3,relief='sunken')

            self.fm1.place(x=300,y=120)

            self.b1=Label(self.fm1,text='User ID',bg='black',font=('Arial',10,'bold'),fg='white')

            self.b1.place(x=20,y=42)

            self.e1=Entry(self.fm1,width=22,font=('arial',9,'bold'),bd=4,relief='groove')

            self.e1.place(x=100,y=40)

            self.lb2=Label(self.fm1,text='Password',bg='black',font=('Arial',10,'bold'),fg='white')

            self.lb2.place(x=20,y=102)

            self.e2=Entry(self.fm1,width=22,show='\*',font=('arial',9,'bold'),bd=4,relief='groove')

            self.e2.place(x=100,y=100)

            self.btn1=Button(self.fm1,text='  Login',fg='black',bg='yellow',width=100,font=('Arial',11,'bold'),activebackground='black',activeforeground='yellow',command=self.login,bd=3,relief='flat',cursor='hand2')

            self.btn1.place(x=25,y=160)

            self.logo = PhotoImage(file=r"pathtoimage.png")

            self.btn1.config(image=self.logo, compound=LEFT)

            self.small\_logo = self.logo.subsample(1, 1)

            self.btn1.config(image=self.small\_logo)

            self.btn2=Button(self.fm1,text='  Clear',fg='black',bg='yellow',width=100,font=('Arial',11,'bold'),activebackground='black',activeforeground='yellow',bd=3,relief='flat',cursor='hand2',command=self.mainclear)

            self.btn2.place(x=155,y=160)

            self.log = PhotoImage(file=r"pathtoimage.png")

            self.btn2.config(image=self.log, compound=LEFT)

            self.small\_log = self.log.subsample(1, 1)

            self.btn2.config(image=self.small\_log)

            self.forgot=Label(self.fm1,text='Forgot Password?',fg='White',bg='#000000',activeforeground='black',font=('cursive',9,'bold'))

            self.forgot.place(x=80,y=220)

            self.forgot.bind("<Button>",self.mouseClick)

            root.mainloop()

    def mouseClick(self,event):

        self.rog=Tk()

        self.rog.title("Change password")

        self.rog.geometry("400x300+300+210")

        self.rog.iconbitmap("filename.ico")

        self.rog.resizable(0,0)

        self.rog.configure(bg='#000')

        self.framerog=Frame(self.rog,width=160,height=30,bg="#d6ed17")

        self.framerog.place(x=95,y=15)

        self.label=Label(self.framerog,text="SET NEW PASSWORD",bg='#d6ed17',fg='#606060',font=("Calibri",12,'bold'))

        self.label.place(x=5,y=4)

        self.user=Label(self.rog,text='User ID',bg='#000',fg='white',font=("Times New Roman",11,'bold'))

        self.user.place(x=40,y=95)

        self.user = Label(self.rog, text='New Password',bg='#000', fg='white', font=("Times New Roman", 11, 'bold'))

        self.user.place(x=40, y=170)

        self.ef1 = Entry(self.rog, width=24, font=('Calibri', 8, 'bold'), bd=4, relief='groove')

        self.ef1.place(x=170, y=95)

        self.ef2 = Entry(self.rog, width=24, font=('Calibri', 8, 'bold'), bd=4, relief='groove')

        self.ef2.place(x=170, y=170)

        self.btn1 = Button(self.rog, text='SUBMIT', fg='#606060', bg='#d6ed17', width=8, font=('Calibri', 12, 'bold'),activebackground='black', activeforeground='#d6ed17',bd=3, relief='flat',cursor='hand2',command=self.chan\_pas)

        self.btn1.place(x=40, y=240)

    def chan\_pas(self):

        self.a=self.ef1.get()

        self.b=self.ef2.get()

        import sqlite3

        conn=sqlite3.connect('admin.db')

        cursor=conn.cursor()

        cursor.execute("SELECT \* FROM UserLogin WHERE UserID='"+self.a+"'")

        conn.commit()

        self.data=cursor.fetchone()

        if self.data!=None:

                cursor = conn.cursor()

                cursor.execute("UPDATE UserLogin SET Password='" + self.b + "' WHERE UserID='" + self.a + "'")

                conn.commit()

                messagebox.showinfo("SUCCESSFUL","Your Password is changed")

                self.rog.destroy()

        else:

                messagebox.showerror("ERROR", "UserID doesn't exist")

                self.rog.destroy()

        self.rog.mainloop()

obj=main()

obj.code()