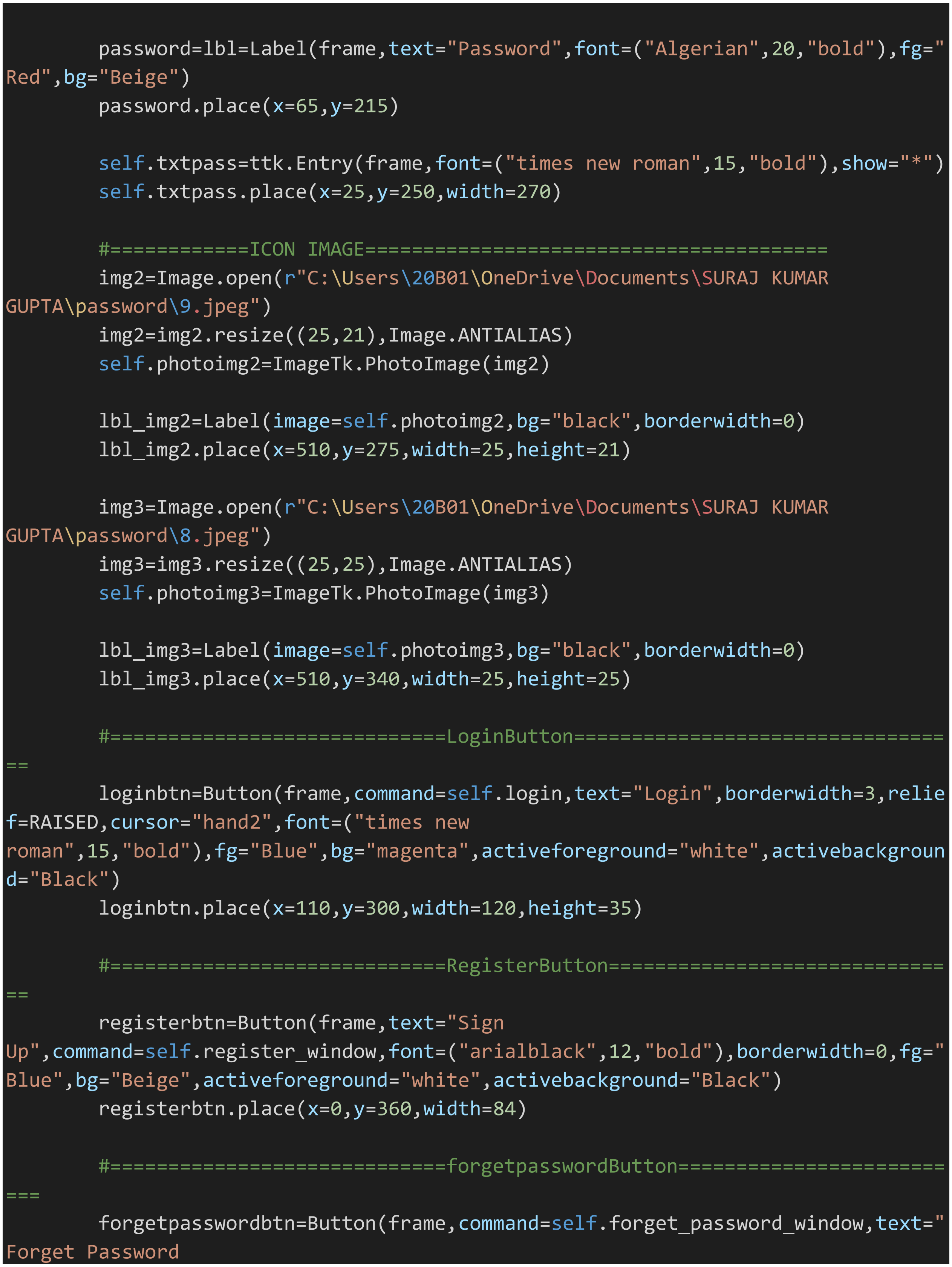
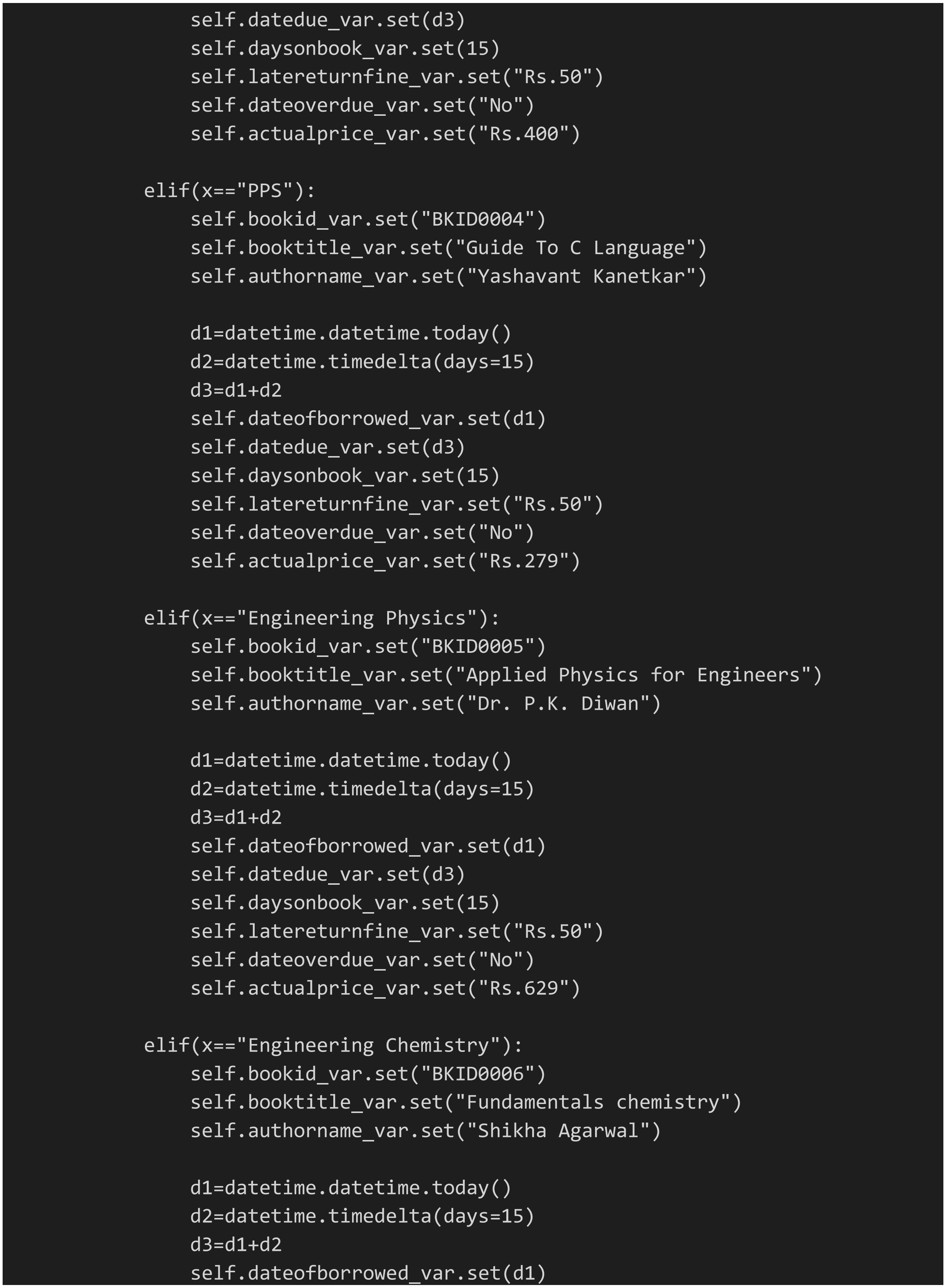
```
from tkinter import *
from tkinter import ttk
from PIL import Image,ImageTk   #pip install pillow
from tkinter import messagebox
import mysql.connector
import tkinter
import datetime


def main():
    win=Tk()
    app=Login_Window(win)
    win.mainloop()


class Login_Window:
    def __init__(self,root):
        self.root=root
        self.root.title("Login")
        self.root.geometry("1270x700+0+0")

        self.bg=ImageTk.PhotoImage(file=r"C:\Users\20B01\OneDrive\Documents\SURAJ
KUMAR GUPTA\password\14.jpg")

        lbl_bg=Label(self.root,image=self.bg)
        lbl_bg.place(x=0,y=0,width=1270,height=700)

        frame=Frame(self.root,bg="Beige")
        frame.place(x=480,y=120,width=340,height=450)

        img1=Image.open(r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR
GUPTA\password\6.jpg")
        img1=img1.resize((340,260),Image.ANTIALIAS)
        self.photoimg1=ImageTk.PhotoImage(img1)
        lbl_img1=Label(image=self.photoimg1,bg="Beige",borderwidth=0)
        lbl_img1.place(x=480,y=110,width=340,height=146)

        #==============================label1=====================================
==
        username=lbl=Label(frame,text="Username",font=("Algerian",20,"bold"),fg="
Red",bg="Beige")
        username.place(x=65,y=145)

        self.txtuser=ttk.Entry(frame,font=("times new roman",15,"bold"))
        self.txtuser.place(x=25,y=180,width=270)
```

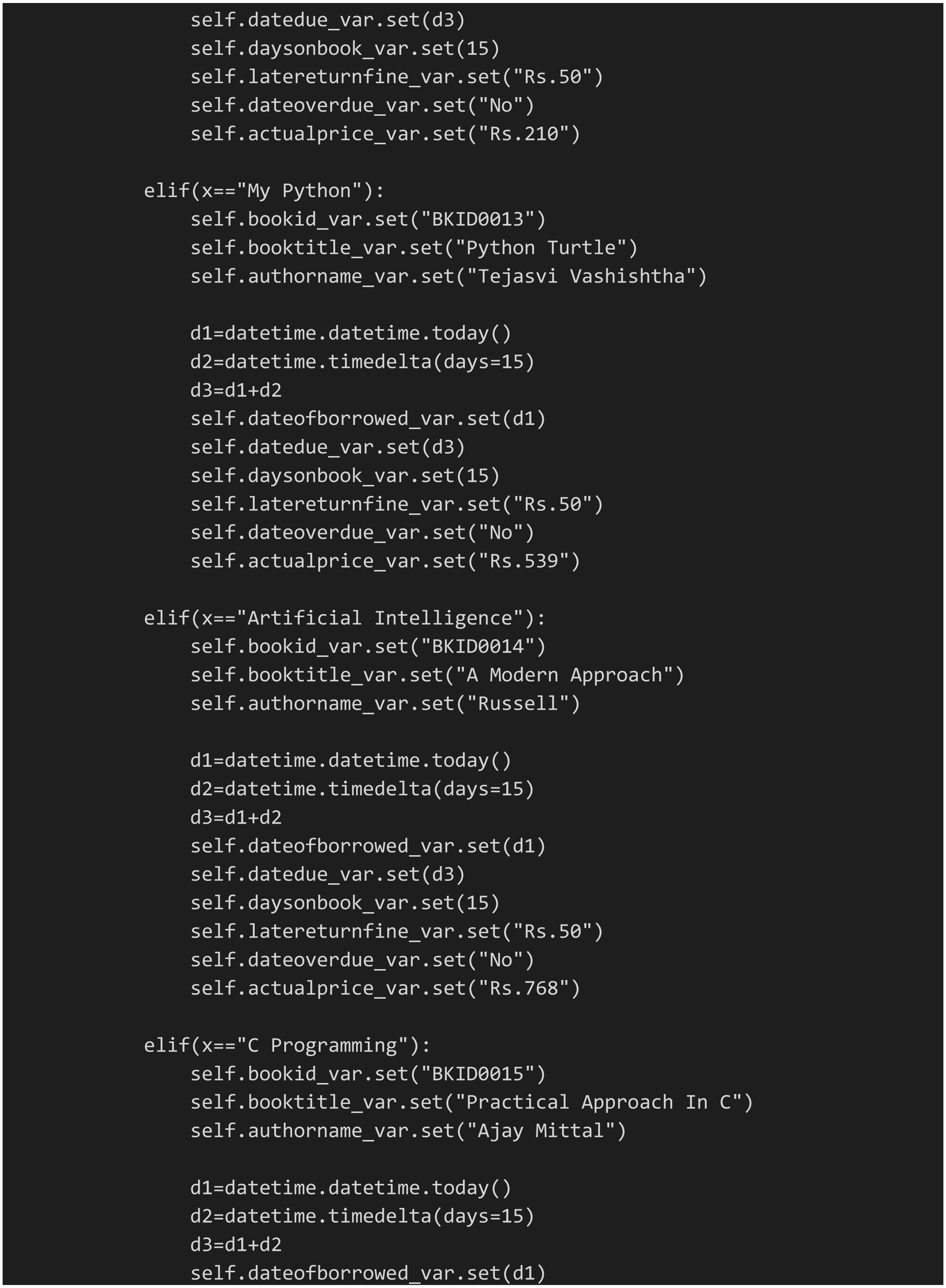


```
        password=lbl=Label(frame,text="Password",font=("Algerian",20,"bold"),fg="Red",bg="Beige")
        password.place(x=65,y=215)

        self.txtpass=ttk.Entry(frame,font=("times new roman",15,"bold"),show="*")
        self.txtpass.place(x=25,y=250,width=270)

        #===========ICON IMAGE=========================================
        img2=Image.open(r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR GUPTA\password\9.jpeg")
        img2=img2.resize((25,21),Image.ANTIALIAS)
        self.photoimg2=ImageTk.PhotoImage(img2)

        lbl_img2=Label(image=self.photoimg2,bg="black",borderwidth=0)
        lbl_img2.place(x=510,y=275,width=25,height=21)

        img3=Image.open(r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR GUPTA\password\8.jpeg")
        img3=img3.resize((25,25),Image.ANTIALIAS)
        self.photoimg3=ImageTk.PhotoImage(img3)

        lbl_img3=Label(image=self.photoimg3,bg="black",borderwidth=0)
        lbl_img3.place(x=510,y=340,width=25,height=25)

        #=============================LoginButton===================================
        loginbtn=Button(frame,command=self.login,text="Login",borderwidth=3,relief=RAISED,cursor="hand2",font=("times new roman",15,"bold"),fg="Blue",bg="magenta",activeforeground="white",activebackground="Black")
        loginbtn.place(x=110,y=300,width=120,height=35)

        #=============================RegisterButton==============================
        registerbtn=Button(frame,text="Sign Up",command=self.register_window,font=("arialblack",12,"bold"),borderwidth=0,fg="Blue",bg="Beige",activeforeground="white",activebackground="Black")
        registerbtn.place(x=0,y=360,width=84)

        #=============================forgetpasswordButton========================
        forgetpasswordbtn=Button(frame,command=self.forget_password_window,text="Forget Password
?",font=("arialblack",12,"bold"),borderwidth=0,fg="Blue",bg="Beige",activeforegro
und="white",activebackground="black")
        forgetpassworddbtn.place(x=10,y=390,width=150)

    def register_window(self):
        self.new_window=Toplevel(self.root)
        self.app=Register(self.new_window)


    def login(self):
        if self.txtuser.get()=="" or self.txtpass.get()=="":
            messagebox.showerror("Error","Fill the required details.")
        else:
            conn=mysql.connector.connect(host="localhost",username="root",passwor
d="Suraj22#",database="sys")
            my_cursor=conn.cursor()
            my_cursor.execute("Select * from register where email=%s and
password=%s",(

  self.txtuser.get(),

  self.txtpass.get()
                                                                                 )
)
            row=my_cursor.fetchone()
            if row==None:
                messagebox.showerror("Error","Invalid Username and Password")
            else:
                open_main=messagebox.askyesno("YesNo","Access only admin")
                if open_main>0:
                    self.new_window=Toplevel(self.root)
                    self.app=LibraryManagementSystem(self.new_window)
                else:
                    if not open_main:
                        return

            conn.commit()
            conn.close()

#============================Reset
Password========================================================
    def reset_pass(self):

        if self.combo_security_q.get()=="Select":
            messagebox.showerror("Error","Select security
Question",parent=self.root2)
        elif self.txt_secans.get()=="":
            messagebox.showerror("Error","Please enter the
answer",parent=self.root2)
        elif self.txt_newpass.get()=="":
            messagebox.showerror("Error","Please enter the new
password",parent=self.root2)
        else:
            conn=mysql.connector.connect(host="localhost",username="root",passwor
d="Suraj22#",database="sys")
            my_cursor=conn.cursor()
            query=("Select * from login where email=%s and Securityq=%s and
SecurityA=%s")
            value=(self.txtuser.get(),self.combo_security_q.get(),self.txt_securi
tyA.get())
            my_cursor.execute(query,value)
            row=my_cursor.fetchone()
            if row==None:
                messagebox.showerror("Error","Please enter correct
Answer",parent=self.root2)
            else:
                query=("update login set password=%s where email=%s")
                value=(self.txt_newpass.get(),self.txtuser.get())
                my_cursor.execute(query,value)

                conn.commit()
                conn.close()
                messagebox.showinfo("Info","Your Password has been reset
successfully.",parent=self.root2)
                self.root2.destroy()

    #=========================Forget
Password========================================
    def forget_password_window(self):
        if self.txtuser.get()=="":
            messagebox.showerror("Error","Please enter the username to reset the
password.")
        else:
            conn=mysql.connector.connect(host="localhost",username="root",passwor
d="Suraj22#",database="sys")
            my_cursor=conn.cursor()
            query=("Select * from register where email=%s")
            value=(self.txtuser.get(),)
            my_cursor.execute(query,value)
            row=my_cursor.fetchone()

            if row==None:
                messagebox.showerror("My Error","Please enter the valid 
username")
            else:
                conn.close()
                self.root2=Toplevel()
                self.root2.title("Forget Password")
                self.root2.geometry("340x450+610+170")

                l=Label(self.root2,text="Forgot Password",font=("times new 
roman",20,"bold"),fg="white",bg="black")
                l.place(x=50,y=10,relwidth=1)

                security_q=Label(self.root2,text="Select Security Question 
",font=("times new roman",15,"bold"),fg="black")
                security_q.place(x=30,y=80)

                self.combo_security_q=ttk.Combobox(self.root2,font=("times new 
roman",15,"bold"),state="readonly")
                self.combo_security_q["values"]=("Select","Your Birth 
place","Your pet name","Your favourite game")
                self.combo_security_q.place(x=30,y=120,width=200)
                self.combo_security_q.current(0)

                securityA=Label(self.root2,text="Security Answer",font=("times 
new roman",15,"bold"),fg="black")
                securityA.place(x=30,y=160)

                self.txt_securityA=ttk.Entry(self.root2,font=("times new 
roman",15,"bold"))
                self.txt_securityA.place(x=30,y=190,width=200)

                newpass=Label(self.root2,text="New Password",font=("times new 
roman",15,"bold"),fg="black")
                newpass.place(x=30,y=230)

                self.txt_newpass=ttk.Entry(self.root2,font=("times new 
roman",15,"bold"),show="*")
                self.txt_newpass.place(x=30,y=260,width=200)

                resetbtn=Button(self.root2,command=self.reset_pass,text="Reset 
Password",font=("times new roman",15,"bold"),fg="white",bg="red")
                resetbtn.place(x=80,y=310,width=160,height=40)
```

```
class Register:
    def __init__(self,root):
        self.root=root
        self.root.title("Register")
        self.root.geometry("1270x700+0+0")


        #==================Variables==================================================
        self.var_fname=StringVar()
        self.var_lname=StringVar()
        self.var_contact=StringVar()
        self.var_email=StringVar()
        self.var_securityq=StringVar()
        self.var_securityA=StringVar()
        self.var_password=StringVar()
        self.var_conpass=StringVar()
        self.var_check=IntVar()


        #====================bg image======================================
        self.bg=ImageTk.PhotoImage(file=r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR GUPTA\password\5.jpg")
        lbl_bg=Label(self.root,image=self.bg)
        lbl_bg.place(x=0,y=0,relwidth=1,relheight=1)

        #====================left image======================================
        self.bg1=ImageTk.PhotoImage(file=r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR GUPTA\password\11.jpeg")
        lbl_left=Label(self.root,image=self.bg1)
        lbl_left.place(x=60,y=100,width=340,height=450)

        #===============main Frame=======================================
        frame=Frame(self.root,bg="lightyellow")
        frame.place(x=400,y=100,width=580,height=450)

        register_lbl=Label(frame,text="REGISTER HERE",font=("Algerian",16,"bold"),fg="DarkGreen",bg="lightyellow")
        register_lbl.place(x=20,y=20)

        #=======================lable and entry===========================

        #==================row1==========================================
        fname=Label(frame,text="First Name",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        fname.place(x=30,y=60)

        fname_entry=ttk.Entry(frame,textvariable=self.var_fname,font=("times new 
roman",15,"bold"))
        fname_entry.place(x=30,y=90,width=200)

        lname=Label(frame,text="Last Name",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        lname.place(x=310,y=60)

        self.txt_lname=ttk.Entry(frame,textvariable=self.var_lname,font=("times 
in roman",15))
        self.txt_lname.place(x=310,y=90,width=200)

        #==================row2========================================
        contact=Label(frame,text="Contact No.",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        contact.place(x=30,y=130)
        self.txt_contact=ttk.Entry(frame,textvariable=self.var_contact,font=("tim
es in roman",15))
        self.txt_contact.place(x=30,y=160,width=200)

        email=Label(frame,text="E-mail",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        email.place(x=310,y=130)
        email_entry=ttk.Entry(frame,textvariable=self.var_email,font=("times new 
roman",15,"bold"))
        email_entry.place(x=310,y=160,width=200)

        security_q=Label(frame,text="Select Security Question ",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        security_q.place(x=30,y=200)
        self.combo_security_q=ttk.Combobox(frame,textvariable=self.var_securityq,
font=("times new roman",15,"bold"),state="readonly")
        self.combo_security_q["values"]=("Select","Your Birth place","Your pet 
name","Your favourite game")
        self.combo_security_q.place(x=30,y=230,width=200)
        self.combo_security_q.current(0)

        securityA=Label(frame,text="Security Answer",font=("times new 
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        securityA.place(x=310,y=200)
        securityA_entry=ttk.Entry(frame,textvariable=self.var_securityA,font=("ti
mes new roman",15,"bold"))
        securityA_entry.place(x=310,y=230,width=200)

        password=Label(frame,text="Password ",font=("times new
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        password.place(x=30,y=270)
        password_entry=ttk.Entry(frame,textvariable=self.var_password,font=("time
s new roman",15,"bold"),show="*")
        password_entry.place(x=30,y=300,width=200)

        conpass=Label(frame,text="Confirm Password",font=("times new
roman",15,"bold"),fg="Maroon",bg="lightyellow")
        conpass.place(x=310,y=270)
        conpass_entry=ttk.Entry(frame,textvariable=self.var_conpass,font=("times
new roman",15,"bold"),show="*")
        conpass_entry.place(x=310,y=300,width=200)

        #====================check button==================================
        checkbtn=Checkbutton(frame,variable=self.var_check,text="I agree to the
terms and condition.",font=("times new
roman",12,"bold"),fg="Maroon",onvalue=1,offvalue=0)
        checkbtn.place(x=30,y=340)

        #=========================BUTTONS========================================
===================================
        img=Image.open(r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR
GUPTA\password\12.jpeg")
        img=img.resize((180,40),Image.ANTIALIAS)
        self.photoimage=ImageTk.PhotoImage(img)
        b1=Button(frame,image=self.photoimage,command=self.register_data,borderwi
dth=0,cursor="hand2",font=("times new roman",15,"bold"))
        b1.place(x=30,y=370,width=180)

        img1=Image.open(r"C:\Users\20B01\OneDrive\Documents\SURAJ KUMAR
GUPTA\password\13.jpeg")
        img1=img1.resize((180,42),Image.ANTIALIAS)
        self.photoimg1=ImageTk.PhotoImage(img1)
        b1=Button(frame,image=self.photoimg1,command=self.return_login,borderwidt
h=0,cursor="hand2",font=("times new roman",15,"bold"))
        b1.place(x=290,y=370,width=180)


        #=================================Fuction
declaration========================================
    def register_data(self):
        if self.var_fname.get()=="" or self.var_email.get()=="" or
self.var_securityq.get()=="Select":
            messagebox.showerror("Error","Fill the required details.")
        elif self.var_password.get()!=self.var_conpass.get():
            messagebox.showerror("Error","Passord and Confirm Password must be
same.")
        elif self.var_check.get()==0:
            messagebox.showerror("Error","Please agree the terms and condition")
        else:
            conn=mysql.connector.connect(host="localhost",username="root",passwor
d="Suraj22#",database="sys")
            my_cursor=conn.cursor()
            query=("Select * from register where email=%s")
            value=(self.var_email.get(),)
            my_cursor.execute(query,value)
            row=my_cursor.fetchone()
            if row!=None:
                messagebox.showerror("Error","User already exist, Please try
another email")
            else:
                my_cursor.execute("insert into register
values(%s,%s,%s,%s,%s,%s,%s)",(

  self.var_fname.get(),

  self.var_lname.get(),

  self.var_contact.get(),

  self.var_email.get(),

  self.var_securityq.get(),

  self.var_securityA.get(),

  self.var_password.get()



 ))
            conn.commit()
            conn.close()
            messagebox.showinfo("Success","Register successfully")
    def return_login(self):
        self.root.destroy()

class LibraryManagementSystem:
    def __init__(self,root):
        self.root=root
        self.root.title("Library Management System")
        self.root.geometry("1270x700+0+0")

        #=============================Variables=================================

        self.member_var=StringVar()
        self.idNo_var=StringVar()
        self.firstname_var=StringVar()
        self.lastname_var=StringVar()
        self.address_var=StringVar()
        self.postcode_var=StringVar()
        self.mobileNo_var=StringVar()
        self.bookid_var=StringVar()
        self.booktitle_var=StringVar()
        self.authorname_var=StringVar()
        self.dateofborrowed_var=StringVar()
        self.daysonbook_var=StringVar()
        self.datedue_var=StringVar()
        self.latereturnfine_var=StringVar()
        self.dateoverdue_var=StringVar()
        self.actualprice_var=StringVar()


        lbltitle=Label(self.root,text="LIBRARY MANAGEMENT
SYSTEM",bg="Aquamarine",fg="red",bd=10,relief=RIDGE,font=("Algerian",50,"bold"),p
adx=2,pady=6)
        lbltitle.pack(side=TOP,fill=X)

        frame=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="light blue")
        frame.place(x=0,y=105,width=1270,height=325)

        #=================================Data Frame
Left===================================

        DataFrameLeft=LabelFrame(frame,text="Library Membership
Information",bg="light yellow",fg="maroon",bd=12,relief=RIDGE,font=("times new
roman",12,"bold"))
        DataFrameLeft.place(x=0,y=4,width=740,height=290)
        lblMember=Label(DataFrameLeft,text=" Member Type",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblMember.grid(row=0,column=0,sticky=W)

        comMember=ttk.Combobox(DataFrameLeft,textvariable=
self.member_var,font=("times new roman",10,"bold"),width=24,state="readonly")
        comMember["value"]=("Admin staff","Student","Lecturer")
        comMember.grid(row=0,column=1)

        lblID_No=Label(DataFrameLeft,text=" ID No:",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=4)
        lblID_No.grid(row=1,column=0,sticky=W)
        txtID_No=Entry(DataFrameLeft,textvariable=self.idNo_var,font=("arial",10,
"bold"),width=27)
        txtID_No.grid(row=1,column=1)

        lblFirstName=Label(DataFrameLeft,text=" First Name",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblFirstName.grid(row=2,column=0,sticky=W)
        txtFirstName=Entry(DataFrameLeft,textvariable=self.firstname_var,font=("a
rial",10,"bold"),width=27)
        txtFirstName.grid(row=2,column=1)

        lblLastName=Label(DataFrameLeft,text=" Last Name",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblLastName.grid(row=3,column=0,sticky=W)
        txtLastName=Entry(DataFrameLeft,textvariable=self.lastname_var,font=("ari
al",10,"bold"),width=27)
        txtLastName.grid(row=3,column=1)

        lblAddress=Label(DataFrameLeft,text=" Address",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblAddress.grid(row=4,column=0,sticky=W)
        txtAddress=Entry(DataFrameLeft,textvariable=self.address_var,font=("arial
",10,"bold"),width=27)
        txtAddress.grid(row=4,column=1)

        lblPostCode=Label(DataFrameLeft,text=" Post Code",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblPostCode.grid(row=5,column=0,sticky=W)
        txtPostCode=Entry(DataFrameLeft,textvariable=self.postcode_var,font=("ari
al",10,"bold"),width=27)
        txtPostCode.grid(row=5,column=1)
        lblMobile_No=Label(DataFrameLeft,text=" Mobile No.",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblMobile_No.grid(row=6,column=0,sticky=W)
        txtMobile_No=Entry(DataFrameLeft,textvariable=
self.mobileNo_var,font=("arial",10,"bold"),width=27)
        txtMobile_No.grid(row=6,column=1)

        lblBookId=Label(DataFrameLeft,text=" Book Id:",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2)
        lblBookId.grid(row=7,column=0,sticky=W)
        txtBookId=Entry(DataFrameLeft,textvariable=self.bookid_var,font=("arial",
10,"bold"),width=27)
        txtBookId.grid(row=7,column=1)

        lblBookTitle=Label(DataFrameLeft,text="       Book Title",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblBookTitle.grid(row=0,column=2,sticky=W)
        txtBookTitle=Entry(DataFrameLeft,textvariable=self.booktitle_var,font=("a
rial",10,"bold"),width=27)
        txtBookTitle.grid(row=0,column=3)

        lblAuthorName=Label(DataFrameLeft,text="       Author Name",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblAuthorName.grid(row=1,column=2,sticky=W)
        txtAuthorName=Entry(DataFrameLeft,textvariable=
self.authorname_var,font=("arial",10,"bold"),width=27)
        txtAuthorName.grid(row=1,column=3)

        lblDateBorrowed=Label(DataFrameLeft,text="       Date Borrowed",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblDateBorrowed.grid(row=2,column=2,sticky=W)
        txtDateBorrowed=Entry(DataFrameLeft,textvariable=self.dateofborrowed_var,
font=("arial",10,"bold"),width=27)
        txtDateBorrowed.grid(row=2,column=3)

        lblDateDue=Label(DataFrameLeft,text="       Date Due",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblDateDue.grid(row=3,column=2,sticky=W)
        txtDateDue=Entry(DataFrameLeft,textvariable=self.datedue_var,font=("arial
",10,"bold"),width=27)
        txtDateDue.grid(row=3,column=3)

        lblDaysOnBook=Label(DataFrameLeft,text="       Days on Book",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2)
        lblDaysOnBook.grid(row=4,column=2,sticky=W)
        txtDaysOnBook=Entry(DataFrameLeft,textvariable=
self.daysonbook_var,font=("arial",10,"bold"),width=27)
        txtDaysOnBook.grid(row=4,column=3)

        lblLateReturnFine=Label(DataFrameLeft,text="       Late Return
Fine",bg="light yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblLateReturnFine.grid(row=5,column=2,sticky=W)
        txtLateReturnFine=Entry(DataFrameLeft,textvariable=self.latereturnfine_va
r,font=("arial",10,"bold"),width=27)
        txtLateReturnFine.grid(row=5,column=3)

        lblDateOverDue=Label(DataFrameLeft,text="       Date Over Fine",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblDateOverDue.grid(row=6,column=2,sticky=W)
        txtDateOverFine=Entry(DataFrameLeft,textvariable=
self.dateoverdue_var,font=("arial",10,"bold"),width=27)
        txtDateOverFine.grid(row=6,column=3)

        lblActualPrice=Label(DataFrameLeft,text="       Actual Price",bg="light
yellow",fg="red",font=("arial",10,"bold"),padx=2,pady=6)
        lblActualPrice.grid(row=7,column=2,sticky=W)
        txtActualPrice=Entry(DataFrameLeft,textvariable=self.actualprice_var,font
=("arial",10,"bold"),width=27)
        txtActualPrice.grid(row=7,column=3)

        #==============================Data Frame
Right=================================

        DataFrameRight=LabelFrame(frame,text="Book Details",bg="light
yellow",fg="Maroon",bd=12,relief=RIDGE,font=("times new roman",12,"bold"))
        DataFrameRight.place(x=760,y=5,width=470,height=290)

        self.txtBox=Text(DataFrameRight,font=("arial",10,"bold"),width=34,height=
15,padx=2,pady=6)
        self.txtBox.grid(row=0,column=2)

        listScrollbar=Scrollbar(DataFrameRight)
        listScrollbar.grid(row=0,column=1,sticky="ns")

        listBooks=['Discrete Mathematics','Engineering Mathematics','Python
Programming','PPS','Technical Communication','Electrical Technology',
        'Machine Technology','Advance Python','Engineering
Chemistry','Engineering Physics','COA','Machine Python','C
Programming','Artificial Intelligence','My Python',
        'Three Men in a Boat','Gullivers Travels','General Knowledge','Aptitude 
Learning','Logical Reasoning',]

        def SelectBook(event=""):
            value=str(listBox.get(listBox.curselection()))
            x=value
            if(x=="Discrete Mathematics"):
               self.bookid_var.set("BKID0001")
               self.booktitle_var.set("Discrete Logic")
               self.authorname_var.set("Harish Mittal")

               d1=datetime.datetime.today()
               d2=datetime.timedelta(days=15)
               d3=d1+d2
               self.dateofborrowed_var.set(d1)
               self.datedue_var.set(d3)
               self.daysonbook_var.set(15)
               self.latereturnfine_var.set("Rs.50")
               self.dateoverdue_var.set("No")
               self.actualprice_var.set("Rs.466")

            elif(x=="Engineering Mathematics"):
                self.bookid_var.set("BKID0002")
                self.booktitle_var.set("Basic Maths")
                self.authorname_var.set("Manish Goyal")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.489")

            elif(x=="Python Programming"):
                self.bookid_var.set("BKID0003")
                self.booktitle_var.set("Learning To Python")
                self.authorname_var.set("Reema Thareja")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
```

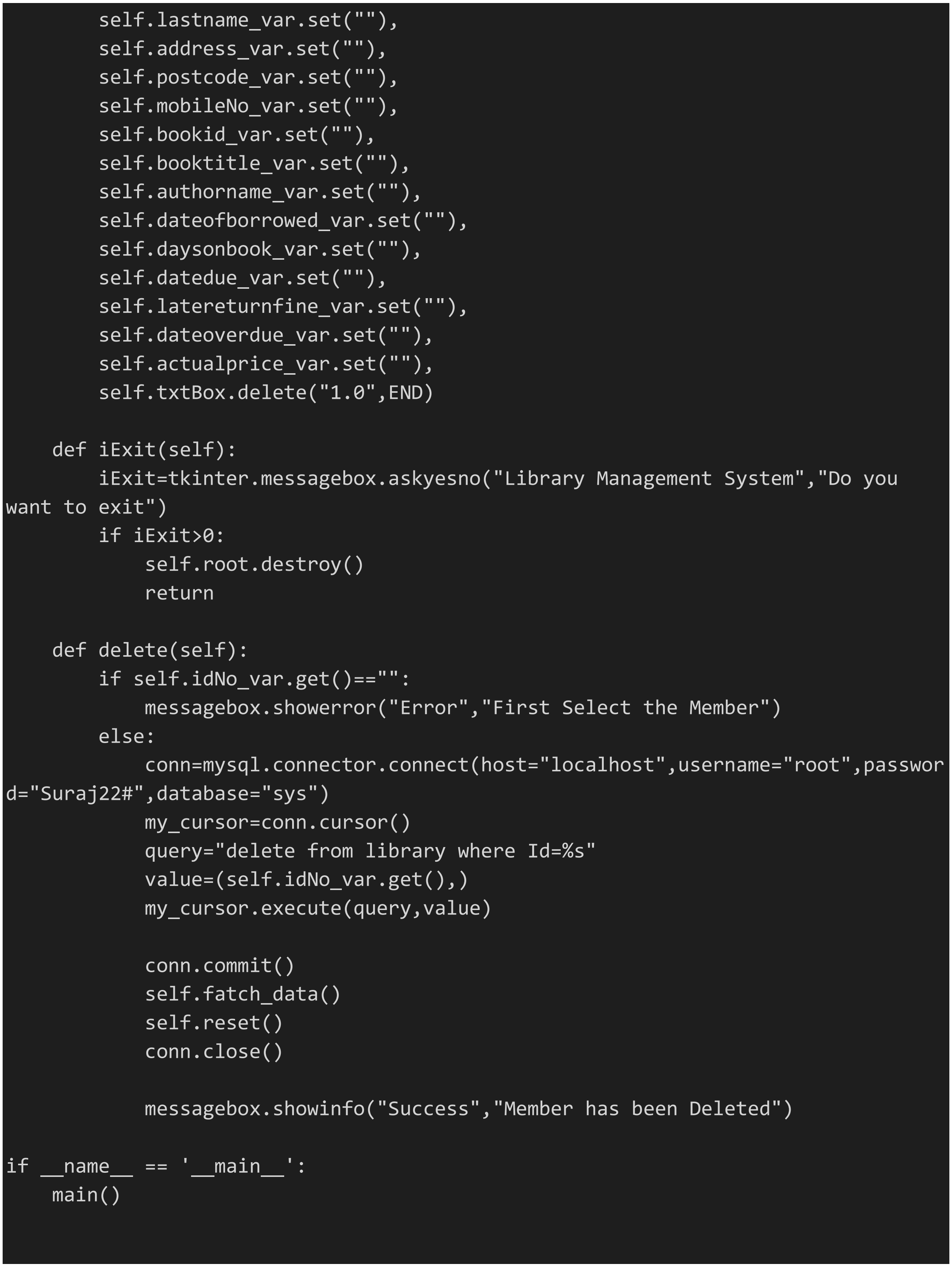


```
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.400")

        elif(x=="PPS"):
            self.bookid_var.set("BKID0004")
            self.booktitle_var.set("Guide To C Language")
            self.authorname_var.set("Yashavant Kanetkar")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.279")

        elif(x=="Engineering Physics"):
            self.bookid_var.set("BKID0005")
            self.booktitle_var.set("Applied Physics for Engineers")
            self.authorname_var.set("Dr. P.K. Diwan")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.629")

        elif(x=="Engineering Chemistry"):
            self.bookid_var.set("BKID0006")
            self.booktitle_var.set("Fundamentals chemistry")
            self.authorname_var.set("Shikha Agarwal")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.847")

        elif(x=="Advance Python"):
            self.bookid_var.set("BKID0007")
            self.booktitle_var.set("Learning With Python")
            self.authorname_var.set("Allen Downey")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.314")

        elif(x=="Machine Technology"):
            self.bookid_var.set("BKID0008")
            self.booktitle_var.set("Machine Design")
            self.authorname_var.set("Robert C. Juvinall")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.875")

        elif(x=="COA"):
            self.bookid_var.set("BKID0009")
            self.booktitle_var.set("Computer System")
            self.authorname_var.set("Mano M Morris")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.721")

        elif(x=="Electrical Technology"):
            self.bookid_var.set("BKID0010")
            self.booktitle_var.set("Electrical Fundamentals")
            self.authorname_var.set("SP Bali")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.289")

        elif(x=="Three Men in a Boat"):
            self.bookid_var.set("BKID0011")
            self.booktitle_var.set("Novel")
            self.authorname_var.set("Jerome K Jerome")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
            self.datedue_var.set(d3)
            self.daysonbook_var.set(15)
            self.latereturnfine_var.set("Rs.50")
            self.dateoverdue_var.set("No")
            self.actualprice_var.set("Rs.99")

        elif(x=="Gullivers Travels"):
            self.bookid_var.set("BKID0012")
            self.booktitle_var.set("Penguin Classics")
            self.authorname_var.set("Jonathan Swift")

            d1=datetime.datetime.today()
            d2=datetime.timedelta(days=15)
            d3=d1+d2
            self.dateofborrowed_var.set(d1)
```



```
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.210")

            elif(x=="My Python"):
                self.bookid_var.set("BKID0013")
                self.booktitle_var.set("Python Turtle")
                self.authorname_var.set("Tejasvi Vashishtha")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.539")

            elif(x=="Artificial Intelligence"):
                self.bookid_var.set("BKID0014")
                self.booktitle_var.set("A Modern Approach")
                self.authorname_var.set("Russell")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.768")

            elif(x=="C Programming"):
                self.bookid_var.set("BKID0015")
                self.booktitle_var.set("Practical Approach In C")
                self.authorname_var.set("Ajay Mittal")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.537")

            elif(x=="Technical Communication"):
                self.bookid_var.set("BKID0016")
                self.booktitle_var.set("Principle and practice")
                self.authorname_var.set("Meenakshi Raman")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.520")

            elif(x=="Aptitude Learning"):
                self.bookid_var.set("BKID0017")
                self.booktitle_var.set("Analysis of Aptitude")
                self.authorname_var.set("Richard E.Snow")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.9613")

            elif(x=="Logical Reasoning"):
                self.bookid_var.set("BKID0018")
                self.booktitle_var.set("Analytical Reasoning")
                self.authorname_var.set("Peeyush Bhardwaj")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
```

```
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.275")

            elif(x=="General Knowledge"):
                self.bookid_var.set("BKID0019")
                self.booktitle_var.set("Quicker Knowledge")
                self.authorname_var.set("Babita Kumari")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.390")

            elif(x=="Machine Python"):
                self.bookid_var.set("BKID0020")
                self.booktitle_var.set("Intro to Machine Learning")
                self.authorname_var.set("Andreas Muller")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.1200")

            elif(x=="Technical Communication"):
                self.bookid_var.set("BKID0021")
                self.booktitle_var.set("Principle and practice")
                self.authorname_var.set("Meenakshi Raman")

                d1=datetime.datetime.today()
                d2=datetime.timedelta(days=15)
                d3=d1+d2
                self.dateofborrowed_var.set(d1)
                self.datedue_var.set(d3)
                self.daysonbook_var.set(15)
                self.latereturnfine_var.set("Rs.50")
                self.dateoverdue_var.set("No")
                self.actualprice_var.set("Rs.520")

        listBox=Listbox(DataFrameRight,font=("arial",10,"bold"),width=22,height=1
5)
        listBox.bind("<<ListboxSelect>>",SelectBook)
        listBox.grid(row=0,column=0,padx=4)
        listScrollbar.config(command=listBox.yview)

        for item in listBooks:
            listBox.insert(END,item)

        # Buttons Frame

        Framebutton=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="light blue")
        Framebutton.place(x=0,y=430,width=1270,height=70)

        btnAddData=Button(Framebutton,command=self.adda_data,text="Add
Data",font=("arial",10,"bold"),width=22,bg="Sky Blue",fg="Black")
        btnAddData.grid(row=0,column=0)

        btnAddData=Button(Framebutton,command=self.showData,text="Show
Data",font=("arial",10,"bold"),width=22,bg="Yellow",fg="Black")
        btnAddData.grid(row=0,column=1)

        btnAddData=Button(Framebutton,command=self.update,text="Update",font=("ar
ial",10,"bold"),width=22,bg="Green",fg="White")
        btnAddData.grid(row=0,column=2)

        btnAddData=Button(Framebutton,command=self.delete,text="Delete",font=("ar
ial",10,"bold"),width=22,bg="Red",fg="White")
        btnAddData.grid(row=0,column=3)

        btnAddData=Button(Framebutton,command=self.reset,text="Reset",font=("aria
l",10,"bold"),width=22,bg="Blue",fg="White")
        btnAddData.grid(row=0,column=4)

        btnAddData=Button(Framebutton,command=self.iExit,text="Exit",font=("arial
",10,"bold"),width=22,bg="Magenta",fg="White")
        btnAddData.grid(row=0,column=5)
        #===================================Information 
Frame=================================

        FrameDetails=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="light blue")
        FrameDetails.place(x=0,y=500,width=1270,height=135)

        Table_frame=Frame(FrameDetails,bd=6,relief=RIDGE,bg="light blue")
        Table_frame.place(x=0,y=2,width=1220,height=110)

        xscroll=ttk.Scrollbar(Table_frame,orient=HORIZONTAL)
        yscroll=ttk.Scrollbar(Table_frame,orient=VERTICAL)
        self.library_table=ttk.Treeview(Table_frame,column=("membertype","idno","
firstname","lastname","address","postcode","mobileno",
                                                              "bookid","booktitle"
,"author","dateofborrowed","daysonbook","datedue","latereturnfine",
                                                              "dateoverdue","actua
lprice"),xscrollcommand=xscroll.set,yscrollcommand=yscroll.set)

        xscroll.pack(side=BOTTOM,fill=X)
        yscroll.pack(side=RIGHT,fill=Y)

        xscroll.config(command=self.library_table.xview)
        yscroll.config(command=self.library_table.yview)

        self.library_table.heading("membertype",text="Member Type")
        self.library_table.heading("idno",text="ID No.")
        self.library_table.heading("firstname",text="First Name")
        self.library_table.heading("lastname",text="Last Name")
        self.library_table.heading("address",text="Address")
        self.library_table.heading("postcode",text="Post Code")
        self.library_table.heading("mobileno",text="Mobile No.")
        self.library_table.heading("bookid",text="Book ID")
        self.library_table.heading("booktitle",text="Book Title")
        self.library_table.heading("author",text="Author 
Name")
        self.library_table.heading("dateofborrowed",text="Date Of Borrowed")
        self.library_table.heading("daysonbook",text="Days On Book")
        self.library_table.heading("datedue",text="Date Due")
        self.library_table.heading("latereturnfine",text="Late Return Fine")
        self.library_table.heading("dateoverdue",text="Date Over Due")
        self.library_table.heading("actualprice",text="Actual Price")

        self.library_table["show"]="headings"
        self.library_table.pack(fill=BOTH,expand=1)
        self.library_table.column("membertype",width=100)
        self.library_table.column("idno",width=100)
        self.library_table.column("firstname",width=100)
        self.library_table.column("lastname",width=100)
        self.library_table.column("address",width=100)
        self.library_table.column("postcode",width=100)
        self.library_table.column("mobileno",width=100)
        self.library_table.column("bookid",width=100)
        self.library_table.column("booktitle",width=100)
        self.library_table.column("author",width=100)
        self.library_table.column("dateofborrowed",width=100)
        self.library_table.column("daysonbook",width=100)
        self.library_table.column("datedue",width=100)
        self.library_table.column("latereturnfine",width=100)
        self.library_table.column("dateoverdue",width=100)
        self.library_table.column("actualprice",width=100)

        self.fatch_data()
        self.library_table.bind("<ButtonRelease-1>",self.get_cursor)

    def adda_data(self):
        conn=mysql.connector.connect(host="localhost",username="root",password="S
uraj22#",database="sys")
        my_cursor=conn.cursor()
        my_cursor.execute("insert into library 
values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(

                        self.member_var.get(),


                        self.idNo_var.get(),

                        self.firstname_var.get(),

                        self.lastname_var.get(),

                        self.address_var.get(),

                        self.postcode_var.get(),

                        self.mobileNo_var.get(),

                        self.bookid_var.get(),

                        self.booktitle_var.get(),
                        self.authorname_var.get(),

                        self.dateofborrowed_var.get(),

                        self.daysonbook_var.get(),

                        self.datedue_var.get(),

                        self.latereturnfine_var.get(),

                        self.dateoverdue_var.get(),

                        self.actualprice_var.get()

                       ))

        conn.commit()
        self.fatch_data()
        conn.close()

        messagebox.showinfo("Success","Member has been inserted successfully")

    def update(self):
        conn=mysql.connector.connect(host="localhost",username="root",password="S
uraj22#",database="sys")
        my_cursor=conn.cursor()
        my_cursor.execute("Update library set
Member=%s,FirstName=%s,LastName=%s,Address=%s,Postcode=%s,Mobile=%s,Bookid=%s,Boo
ktitle=%s,AuthorName=%s,DateOfBorrowed=%s,DaysonBook=%s,Datedue=%s,Latereturnfine
=%s,Dateoverdue=%s,ActualPrice=%s where Id=%s",(

                        self.member_var.get(),

                        self.firstname_var.get(),

                        self.lastname_var.get(),

                        self.address_var.get(),

                        self.postcode_var.get(),

                        self.mobileNo_var.get(),

                        self.bookid_var.get(),
                        self.booktitle_var.get(),

                        self.authorname_var.get(),

                        self.dateofborrowed_var.get(),

                        self.datedue_var.get(),

                        self.daysonbook_var.get(),

                        self.latereturnfine_var.get(),

                        self.dateoverdue_var.get(),

                        self.actualprice_var.get(),

                        self.idNo_var.get(),

       ))
        conn.commit()
        self.fatch_data()
        self.reset()
        conn.close()

        messagebox.showinfo("Success","Member has been Updated")


    def fatch_data(self):
        conn=mysql.connector.connect(host="localhost",username="root",password="S
uraj22#",database="sys")
        my_cursor=conn.cursor()
        my_cursor.execute("select * from library")
        rows=my_cursor.fetchall()

        if len(rows)!=0:
            self.library_table.delete(*self.library_table.get_children())
            for i in rows:
                self.library_table.insert("",END,values=i)
                conn.commit()
            conn.close()
    def get_cursor(self,event=""):
            cursor_rows=self.library_table.focus()
            content=self.library_table.item(cursor_rows)
            row=content['values']
            self.member_var.set(row[0]),
            self.idNo_var.set(row[1]),
            self.firstname_var.set(row[2]),
            self.lastname_var.set(row[3]),
            self.address_var.set(row[4]),
            self.postcode_var.set(row[5]),
            self.mobileNo_var.set(row[6]),
            self.bookid_var.set(row[7]),
            self.booktitle_var.set(row[8]),
            self.authorname_var.set(row[9]),
            self.dateofborrowed_var.set(row[10]),
            self.daysonbook_var.set(row[11]),
            self.datedue_var.set(row[12]),
            self.latereturnfine_var.set(row[13]),
            self.dateoverdue_var.set(row[14]),
            self.actualprice_var.set(row[15])

    def showData(self):
        self.txtBox.insert(END,"Member Type\t\t"+ self.member_var.get()+"\n")
        self.txtBox.insert(END,"ID No:\t\t"+ self.idNo_var.get()+"\n")
        self.txtBox.insert(END,"FirstName\t\t"+ self.firstname_var.get()+"\n")
        self.txtBox.insert(END,"LastName\t\t"+ self.lastname_var.get()+"\n")
        self.txtBox.insert(END,"Address\t\t"+ self.address_var.get()+"\n")
        self.txtBox.insert(END,"Post Code\t\t"+ self.postcode_var.get()+"\n")
        self.txtBox.insert(END,"Mobile No.\t\t"+ self.mobileNo_var.get()+"\n")
        self.txtBox.insert(END,"Book ID\t\t"+ self.bookid_var.get()+"\n")
        self.txtBox.insert(END,"Book Title\t\t"+ self.booktitle_var.get()+"\n")
        self.txtBox.insert(END,"Author Name\t\t"+ self.authorname_var.get()+"\n")
        self.txtBox.insert(END,"Date Of Borrowed\t\t"+
self.dateofborrowed_var.get()+"\n")
        self.txtBox.insert(END,"Days On Book\t\t"+
self.daysonbook_var.get()+"\n")
        self.txtBox.insert(END,"Date Due\t\t"+ self.datedue_var.get()+"\n")
        self.txtBox.insert(END,"Late Return Fine\t\t"+
self.latereturnfine_var.get()+"\n")
        self.txtBox.insert(END,"Date Over Due\t\t"+
self.dateoverdue_var.get()+"\n")
        self.txtBox.insert(END,"Actual Price\t\t"+
self.actualprice_var.get()+"\n")

    def reset(self):
        self.member_var.set(""),
        self.idNo_var.set(""),
        self.firstname_var.set(""),
```



```
        self.lastname_var.set(""),
        self.address_var.set(""),
        self.postcode_var.set(""),
        self.mobileNo_var.set(""),
        self.bookid_var.set(""),
        self.booktitle_var.set(""),
        self.authorname_var.set(""),
        self.dateofborrowed_var.set(""),
        self.daysonbook_var.set(""),
        self.datedue_var.set(""),
        self.latereturnfine_var.set(""),
        self.dateoverdue_var.set(""),
        self.actualprice_var.set(""),
        self.txtBox.delete("1.0",END)

    def iExit(self):
        iExit=tkinter.messagebox.askyesno("Library Management System","Do you
want to exit")
        if iExit>0:
            self.root.destroy()
            return

    def delete(self):
        if self.idNo_var.get()=="":
            messagebox.showerror("Error","First Select the Member")
        else:
            conn=mysql.connector.connect(host="localhost",username="root",passwor
d="Suraj22#",database="sys")
            my_cursor=conn.cursor()
            query="delete from library where Id=%s"
            value=(self.idNo_var.get(),)
            my_cursor.execute(query,value)

            conn.commit()
            self.fatch_data()
            self.reset()
            conn.close()

            messagebox.showinfo("Success","Member has been Deleted")

if __name__ == '__main__':
    main()
```