Holy Grace Academy, Mala

Computer Project

**Topic: Library Management System**

**Name :**

**Class :**

**Reg. No. :**

**Roll No. :**

**Academic Year :**

**ACKNOWLEDGEMENT**

I solemnly take the opportunity to thank all the helping hands who made me to complete this project. First of all, I thank the Almighty for keeping me hale and healthy in order to successfully complete my work.

I wish to express my sincere gratitude to Mr. Jose Joseph Alumkal, Principal of Holy Grace Academy, for permitting me to carry out the project and for the facilities he has provided for the fulfilment of this project work.

I am greatly indebted to Mrs. Vandhana, Teacher in Computer Science who gave me immense support and guidance throughout the completion of this project.

Last but not least, I express my heartiest thanks to my lovable parents and friends for their prayers, suggestions and encouragement for the successful completion of the project.

Sincerely,

Rahul A B

**CONTENT**

1. **Introduction**
2. **Hardware And Software Reequipment’s**
3. **Source Code**
4. **Sample Output**
5. **Conclusion**
6. **Bibliography**

**INTRODUCTION**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido Van Rossum and first released in 1991,Python’s design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale objects.

This project is done using python programming language. This program is titled Library Database. As the name suggests, this program is a simple version of a library book management where book can be added ,update, search etc. The work interface uses GUI tkinter module available in python. The program also uses Structured Query Language connectivity to store and retrieve user data.

**HARDWARE & REQUIRMENTS**

RAM : 2GB(Minimum)

4GB(Recommended)

Operating System : 32-bit x86

64-bit x 64(Recommended)

Hard Disk : Minimum 250MB Free Memory

Processor : Dual Core 2.80 GHz or Greater

Screen Resolution : 1366 x 768 (Optional)

Graphics Card : Minimum 64 MB

Platform : Windows 7/8/10 with SP1

Python Version : Python 3.0 or Greater

MySQL Version : MySQL 5.0 or Greater

**SOURCE CODE**

**CODE**

from tkinter import \*

from tkinter import ttk

import tkinter.messagebox

import mysql.connector as mysql

import random

import time

class DataEnrtyForm:

    def \_\_init\_\_(self,root):

        self.root=root

        self.root.title("Data Entry Form")

        self.root.geometry("1350x750+0+0")

        self.root.configure(background="gainsboro")

        try:

            sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@")

            mycursor=sqlcon.cursor()

            q6="create database librarydatabase"

            mycursor.execute(q6)

            sqlcon.close()

            sqlconn=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

            mycursor=sqlconn.cursor()

            q7="create table book(Book\_Id varchar(10) primary key,Book\_Name varchar(100),Author\_Name varchar(100),Book\_Category varchar(20),Book\_Copies int(10),Book\_Language varchar(20),DOR date,Publishers varchar(50),Cost int(10),GST varchar(10),Total int(10))"

            mycursor.execute(q7)

            sqlconn.close()

            tkinter.messagebox.showinfo("Project","New Database And Table Created Successfully")

        except:

            tkinter.messagebox.showinfo("Project","Either Database Or Table Already Exists")

            pass

#=======Tables Variables And Setting================================

        Book\_ID1=IntVar()

        Book\_Name1=StringVar()

        Author\_Name1=StringVar()

        Book\_Category1=StringVar()

        Cpoies1=StringVar()

        Book\_Language1=StringVar()

        DOR1=StringVar()

        Publisher1=StringVar()

        Cost1=IntVar()

        GST1=StringVar()

        Total1=IntVar()

        Search1=StringVar()

        Book\_Name1.set("")

        Author\_Name1.set("")

        Book\_Category1.set("")

        Cpoies1.set("")

        Book\_Language1.set("")

        DOR1.set(time.strftime("%Y/%m/%d"))

        Publisher1.set("")

        Cost1.set("")

        GST1.set("")

        Total1.set("")

#=======Frame Work============================================

        MainFrame=Frame(self.root,bd=10,width=1340,height=700,relief=RIDGE)

        MainFrame.grid()

        TopFrame1=Frame(MainFrame,bd=5,width=1340,height=200,relief=RIDGE,bg="cadet blue")

        TopFrame1.grid(row=0,column=0)

        TopFrame2=Frame(MainFrame,bd=5,width=1340,height=50,relief=RIDGE,bg="cadet blue")

        TopFrame2.grid(row=1,column=0)

        TopFrame3=Frame(MainFrame,bd=5,width=1340,height=300,relief=RIDGE,bg="cadet blue")

        TopFrame3.grid(row=2,column=0)

        InnerTopFrame1=Frame(TopFrame1,bd=5,width=1330,height=180,relief=RIDGE)

        InnerTopFrame1.grid()

        InnerTopFrame2=Frame(TopFrame2,bd=5,width=1330,height=48,relief=RIDGE)

        InnerTopFrame2.grid()

        InnerTopFrame3=Frame(TopFrame3,bd=5,width=1330,height=280,relief=RIDGE)

        InnerTopFrame3.grid()

#=======Definintion============================================

        def iReset():

            Book\_ID1.set("")

            Book\_Name1.set("")

            Author\_Name1.set("")

            Book\_Category1.set("")

            Cpoies1.set("")

            Book\_Language1.set("")

            DOR1.set(time.strftime("%Y/%m/%d"))

            Publisher1.set("")

            Cost1.set("")

            GST1.set("")

            Total1.set("")

            Search1.set("")

        def iExit():

            iExit=tkinter.messagebox.askyesno("Project","Are You Sure You Want to Close?")

            if iExit>0:

                root.destroy()

                return

        def iDisplay():

            sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

            mycursor=sqlcon.cursor()

            q2="select \* from book"

            mycursor.execute(q2)

            result=mycursor.fetchall()

            if len(result)!=0:

                tree\_records.delete(\*tree\_records.get\_children())

                for row in result:

                    tree\_records.insert("",END,values=row)

                    sqlcon.commit()

                sqlcon.close()

        def addData():

            if Book\_ID1.get()==0 or Book\_Name1.get()=="":

                tkinter.messagebox.showerror("Project","No Entry Found In Necessary Columns?")

            else:

                sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

                mycursor=sqlcon.cursor()

                q1="insert into book values('{0}','{1}','{2}','{3}',{4},'{5}','{6}','{7}',{8},'{9}',{10})".format(Book\_ID1.get(),Book\_Name1.get(),Author\_Name1.get(),Book\_Category1.get(),Cpoies1.get(),Book\_Language1.get(),DOR1.get(),Publisher1.get(),Cost1.get(),GST1.get(),Total1.get())

                mycursor.execute(q1)

                sqlcon.commit()

                sqlcon.close()

                iDisplay()

                tkinter.messagebox.showinfo("Project","Record Entered Successfully")

                iReset()

        def TraineeInfo(ev):

            viewInfo=tree\_records.focus()

            learnerData=tree\_records.item(viewInfo)

            row=learnerData['values']

            Book\_ID1.set(row[0])

            Book\_Name1.set(row[1])

            Author\_Name1.set(row[2])

            Book\_Category1.set(row[3])

            Cpoies1.set(row[4])

            Book\_Language1.set(row[5])

            DOR1.set(row[6])

            Publisher1.set(row[7])

            Cost1.set(row[8])

            GST1.set(row[9])

            Total1.set(row[10])

        def iUpdate():

            sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

            mycursor=sqlcon.cursor()

            q3="update book set Book\_Name='{}',Author\_Name='{}',Book\_Language='{}',Book\_Copies={},Publishers='{}' where Book\_ID={}".format(Book\_Name1.get(),Author\_Name1.get(),Book\_Language1.get(),Cpoies1.get(),Publisher1.get(),Book\_ID1.get())

            mycursor.execute(q3)

            sqlcon.commit()

            sqlcon.close()

            iDisplay()

            tkinter.messagebox.showinfo("Project","Record Updated Successfully")

        def iDelete():

            sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

            mycursor=sqlcon.cursor()

            q3="delete from book where Book\_ID={}".format(Book\_ID1.get())

            mycursor.execute(q3)

            sqlcon.commit()

            sqlcon.close()

            iDisplay()

            tkinter.messagebox.showinfo("Project","Record Successfully Removed")

            iReset()

        def iSearch():

            try:

                sqlcon=mysql.connect(host="localhost",user="root",password="123rahulAB@",database="librarydatabase")

                mycursor=sqlcon.cursor()

                q5="select \* from book where Book\_ID='{}'".format(Search1.get())

                mycursor.execute(q5)

                row=mycursor.fetchall()

                x=row[0]

                Book\_ID1.set(x[0])

                Book\_Name1.set(x[1])

                Author\_Name1.set(x[2])

                Book\_Category1.set(x[3])

                Cpoies1.set(x[4])

                Book\_Language1.set(x[5])

                DOR1.set(x[6])

                Publisher1.set(x[7])

                Cost1.set(x[8])

                GST1.set(x[9])

                Total1.set(x[10])

                sqlcon.commit()

                tkinter.messagebox.showinfo("Project","Record Found")

            except:

                tkinter.messagebox.showinfo("Project","No Record Found")

                iReset()

            Search1.set("")

#=======Label & Entry==========================================

        self.lblReferenceID=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Book ID",bd=10)

        self.lblReferenceID.grid(row=0,column=0,stick=W)

        self.txtReferenceID=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Book\_ID1)

        self.txtReferenceID.grid(row=0,column=1)

        self.lblReferenceName=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Book Name",bd=10)

        self.lblReferenceName.grid(row=1,column=0,stick=W)

        self.txtReferenceName=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Book\_Name1)

        self.txtReferenceName.grid(row=1,column=1)

        self.lblReferenceAuthor=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Author Name",bd=10)

        self.lblReferenceAuthor.grid(row=2,column=0,stick=W)

        self.txtReferenceAuthor=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Author\_Name1)

        self.txtReferenceAuthor.grid(row=2,column=1)

        self.lblReferenceCategory=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Book Category",bd=10)

        self.lblReferenceCategory.grid(row=3,column=0,stick=W)

        self.txtReferenceCategory=ttk.Combobox(InnerTopFrame1,font=('arial',12,'bold'),width=29,textvariable=Book\_Category1)

        self.txtReferenceCategory['values']=['','Nonfictional','Fictional','Food','History','Politics','Reference','Self-Help','Others']

        self.txtReferenceCategory.current(0)

        self.txtReferenceCategory.grid(row=3,column=1)

        self.lblReferenceCopy=Label(InnerTopFrame1,font=('arial',12,'bold'),text="No. of copies of the book",bd=10)

        self.lblReferenceCopy.grid(row=0,column=3,stick=W)

        self.txtReferenceCopy=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Cpoies1)

        self.txtReferenceCopy.grid(row=0,column=4)

        self.lblReferenceLanguage=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Language Of the Book",bd=10)

        self.lblReferenceLanguage.grid(row=1,column=3,stick=W)

        self.txtReferenceLanguage=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Book\_Language1)

        self.txtReferenceLanguage.grid(row=1,column=4)

        self.lblReferenceRelease=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Date Of Release",bd=10)

        self.lblReferenceRelease.grid(row=2,column=3,stick=W)

        self.txtReferenceRelease=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=DOR1)

        self.txtReferenceRelease.grid(row=2,column=4)

        self.lblReferencePublishers=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Publishers Of the Book",bd=10)

        self.lblReferencePublishers.grid(row=3,column=3,stick=W)

        self.txtReferencePublishers=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=30,justify="left",textvariable=Publisher1)

        self.txtReferencePublishers.grid(row=3,column=4)

        self.lblReferenceCost=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Cost Of the Book",bd=10)

        self.lblReferenceCost.grid(row=0,column=5,stick=W)

        self.txtReferenceCost=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=20,justify="left",textvariable=Cost1)

        self.txtReferenceCost.grid(row=0,column=6)

        self.lblReferenceGST=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Gst For the book",bd=10)

        self.lblReferenceGST.grid(row=1,column=5,stick=W)

        self.txtReferenceGST=ttk.Combobox(InnerTopFrame1,font=('arial',12,'bold'),width=20,textvariable=GST1)

        self.txtReferenceGST['values']=['','5%','10%','15%','Others']

        self.txtReferenceGST.current(0)

        self.txtReferenceGST.grid(row=1,column=6)

        self.lblReferenceTotal=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Total Cost Of the Book",bd=10)

        self.lblReferenceTotal.grid(row=2,column=5,stick=W)

        self.txtReferenceTotal=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=21,justify="left",textvariable=Total1)

        self.txtReferenceTotal.grid(row=2,column=6)

        self.lblReferenceSearch=Label(InnerTopFrame1,font=('arial',12,'bold'),text="Book ID Search",bd=10)

        self.lblReferenceSearch.grid(row=3,column=5,stick=W)

        self.txtReferenceSearch=Entry(InnerTopFrame1,font=('arial',12,'bold'),bd=5,width=21,justify="left",textvariable=Search1)

        self.txtReferenceSearch.grid(row=3,column=6)

#=======Tables================================================

        scroll\_x=Scrollbar(InnerTopFrame3,orient=HORIZONTAL)

        scroll\_x=Scrollbar(InnerTopFrame3,orient=VERTICAL)

        tree\_records=ttk.Treeview(InnerTopFrame3,height=20,columns=("Book\_ID","Book\_Name","Author\_Name","Book\_Category","Copies","Book\_Language","DOR","Publishers","Cost","GST","Total"),xscrollcommand=scroll\_x.set, yscrollcommand=scroll\_x.set)

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

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

        tree\_records.heading("Book\_ID",text="Book ID")

        tree\_records.heading("Book\_Name",text="Book Name")

        tree\_records.heading("Author\_Name",text="Author Name")

        tree\_records.heading("Book\_Category",text="Book Category")

        tree\_records.heading("Copies",text="Copies Of Book")

        tree\_records.heading("Book\_Language",text="Language Written")

        tree\_records.heading("DOR",text="Date Of Book Release")

        tree\_records.heading("Publishers",text="Publishers")

        tree\_records.heading("Cost",text="Cost Of Book")

        tree\_records.heading("GST",text="GST")

        tree\_records.heading("Total",text="Total Cost")

        tree\_records['show']='headings'

        tree\_records.column("Book\_ID",width=70)

        tree\_records.column("Book\_Name",width=120)

        tree\_records.column("Author\_Name",width=120)

        tree\_records.column("Book\_Category",width=120)

        tree\_records.column("Copies",width=120)

        tree\_records.column("Book\_Language",width=120)

        tree\_records.column("DOR",width=150)

        tree\_records.column("Publishers",width=150)

        tree\_records.column("Cost",width=120)

        tree\_records.column("GST",width=120)

        tree\_records.column("Total",width=120)

        tree\_records.pack(fill=BOTH,expand=1)

        tree\_records.bind("<ButtonRelease-1>",TraineeInfo)

        iDisplay()

#=======Button================================================

        self.btnAddNew=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Add Data",command=addData)

        self.btnAddNew.grid(row=0,column=0,padx=3)

        self.btnDisplay=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Display",command=iDisplay)

        self.btnDisplay.grid(row=0,column=1,padx=3)

        self.btnUpdate=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Update",command=iUpdate)

        self.btnUpdate.grid(row=0,column=2,padx=3)

        self.btnSearch=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Search",command=iSearch)

        self.btnSearch.grid(row=0,column=3,padx=3)

        self.btnDelete=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Delete",command=iDelete)

        self.btnDelete.grid(row=0,column=4,padx=3)

        self.btnReset=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Reset",command=iReset)

        self.btnReset.grid(row=0,column=5,padx=3)

        self.btnExit=Button(InnerTopFrame2,pady=1,bd=4,font=('arial',16,'bold'),width=13,text="Exit",command=iExit)

        self.btnExit.grid(row=0,column=6,padx=3)

#=======Tables================================================

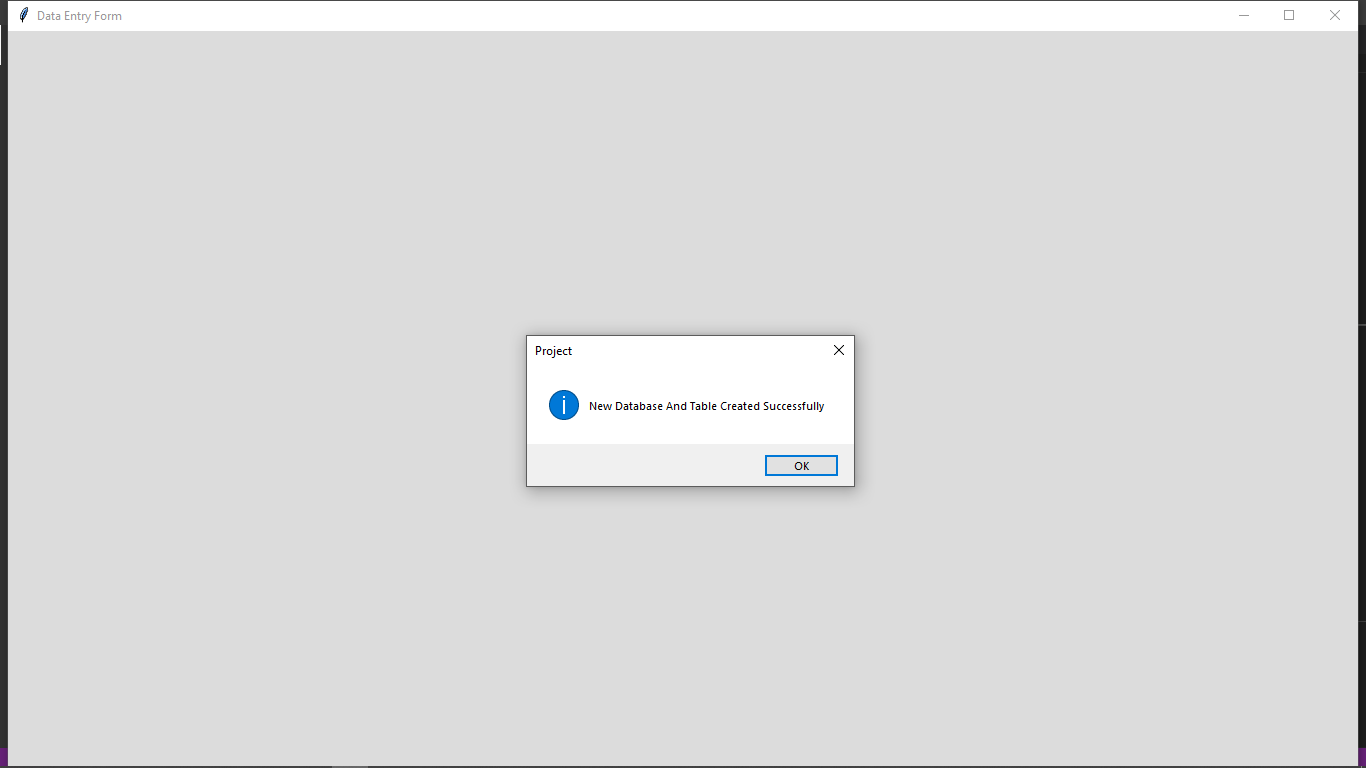
if \_\_name\_\_=='\_\_main\_\_':

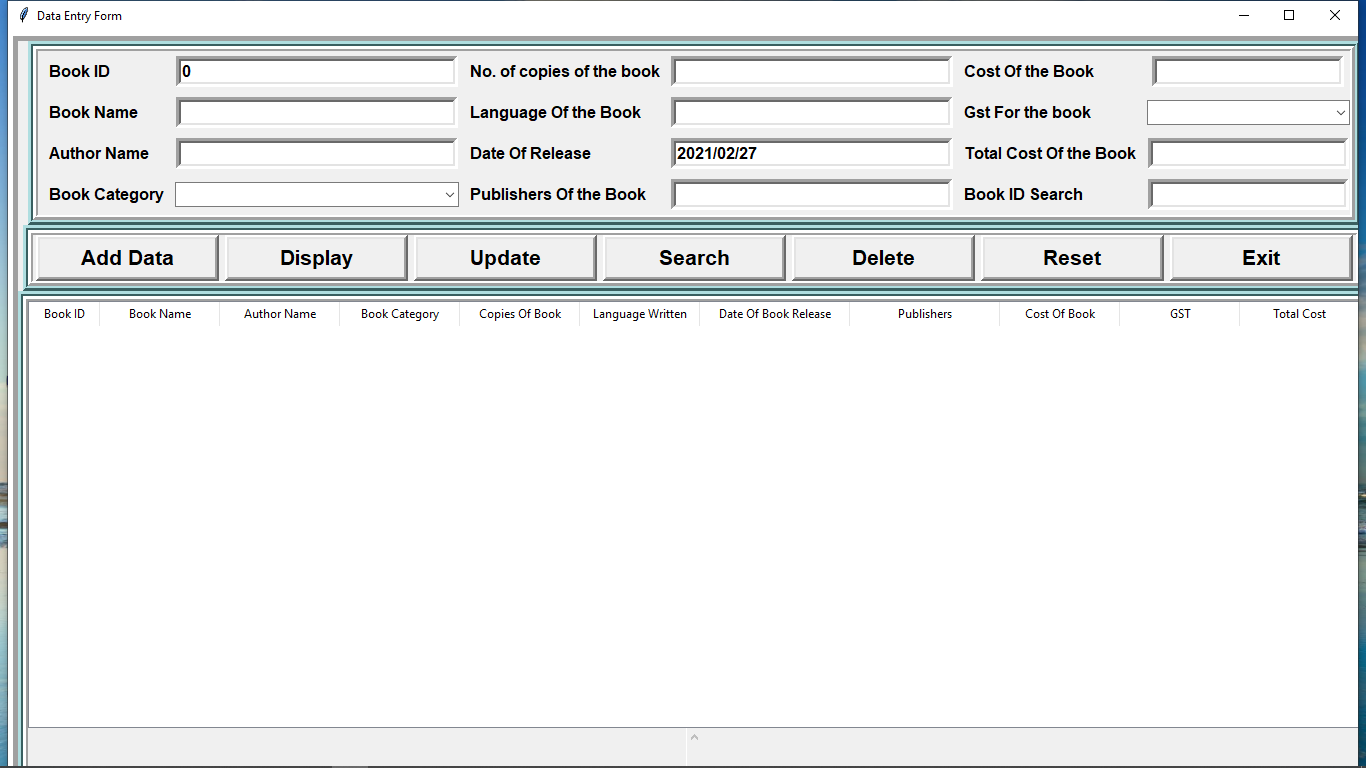
    root=Tk()

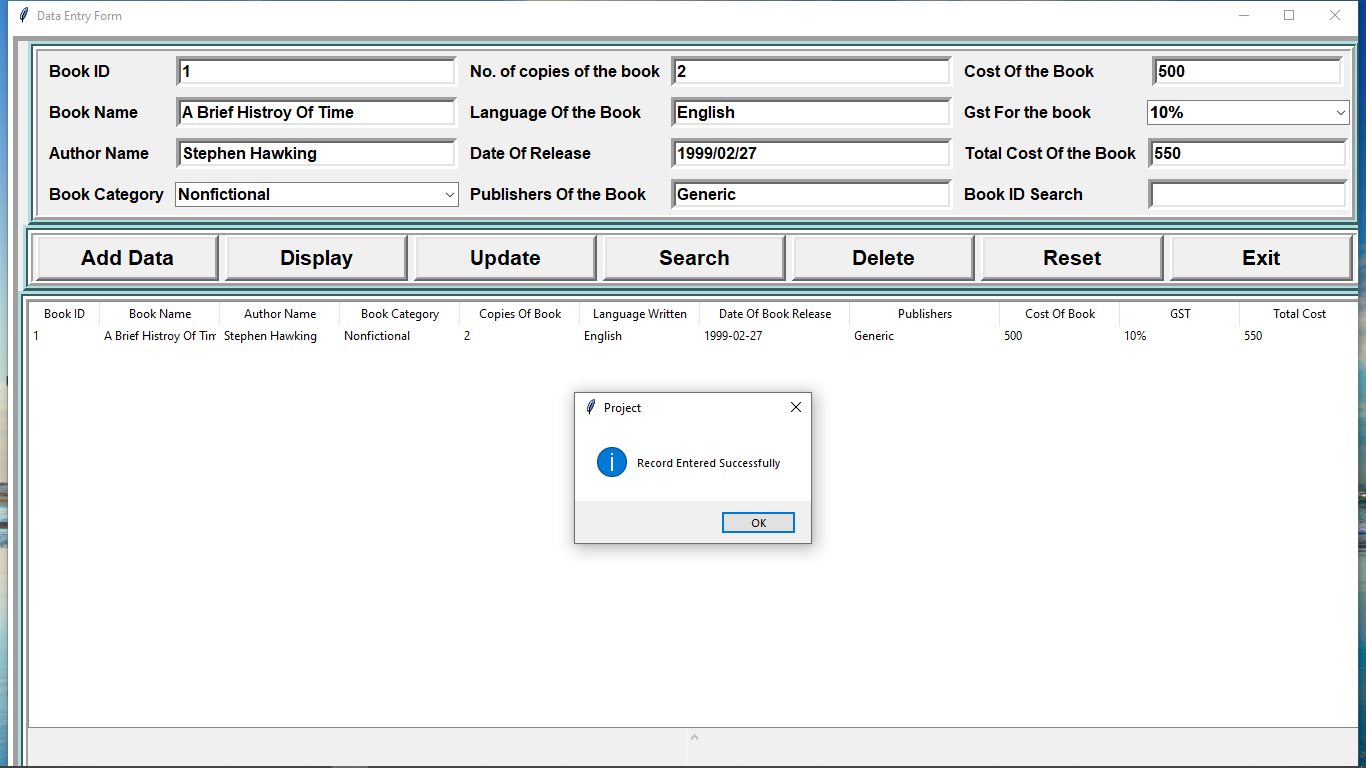
    applicatiopn=DataEnrtyForm(root)

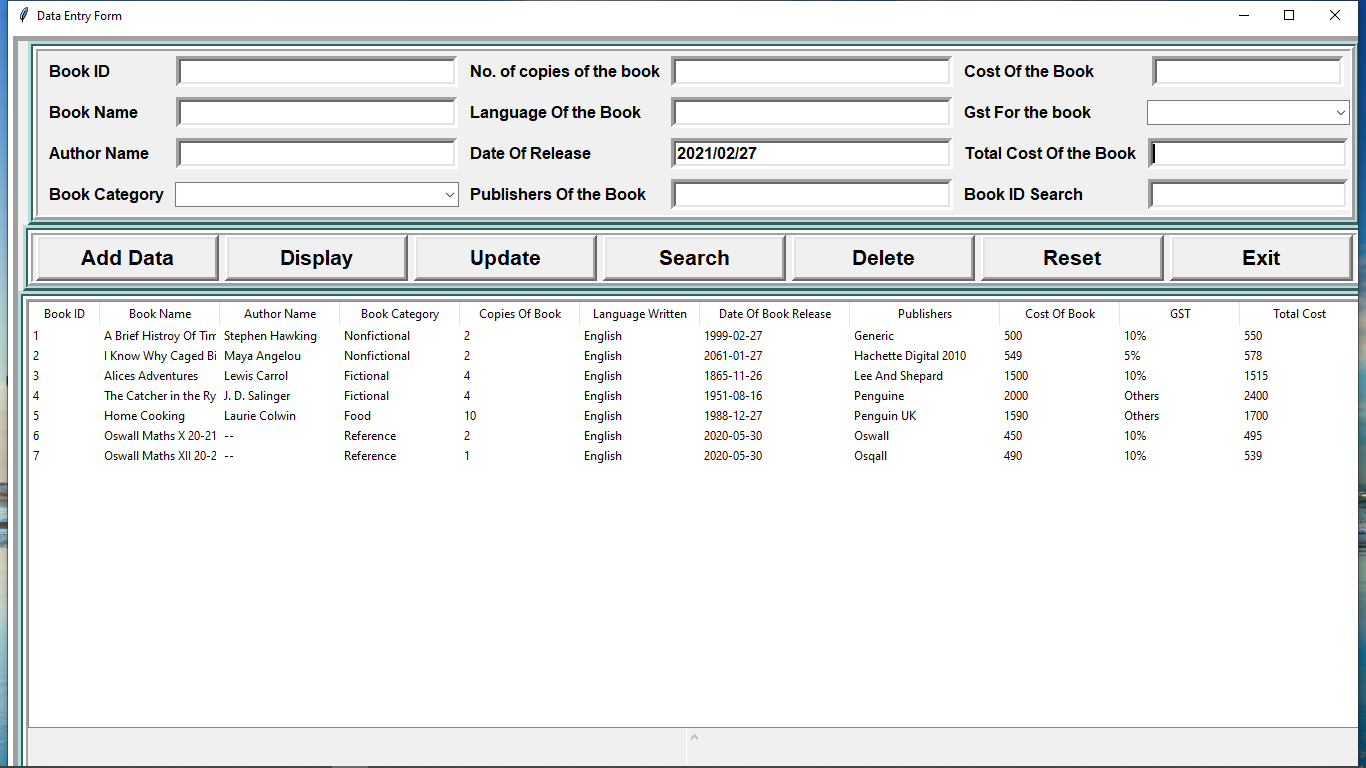
    root.mainloop()

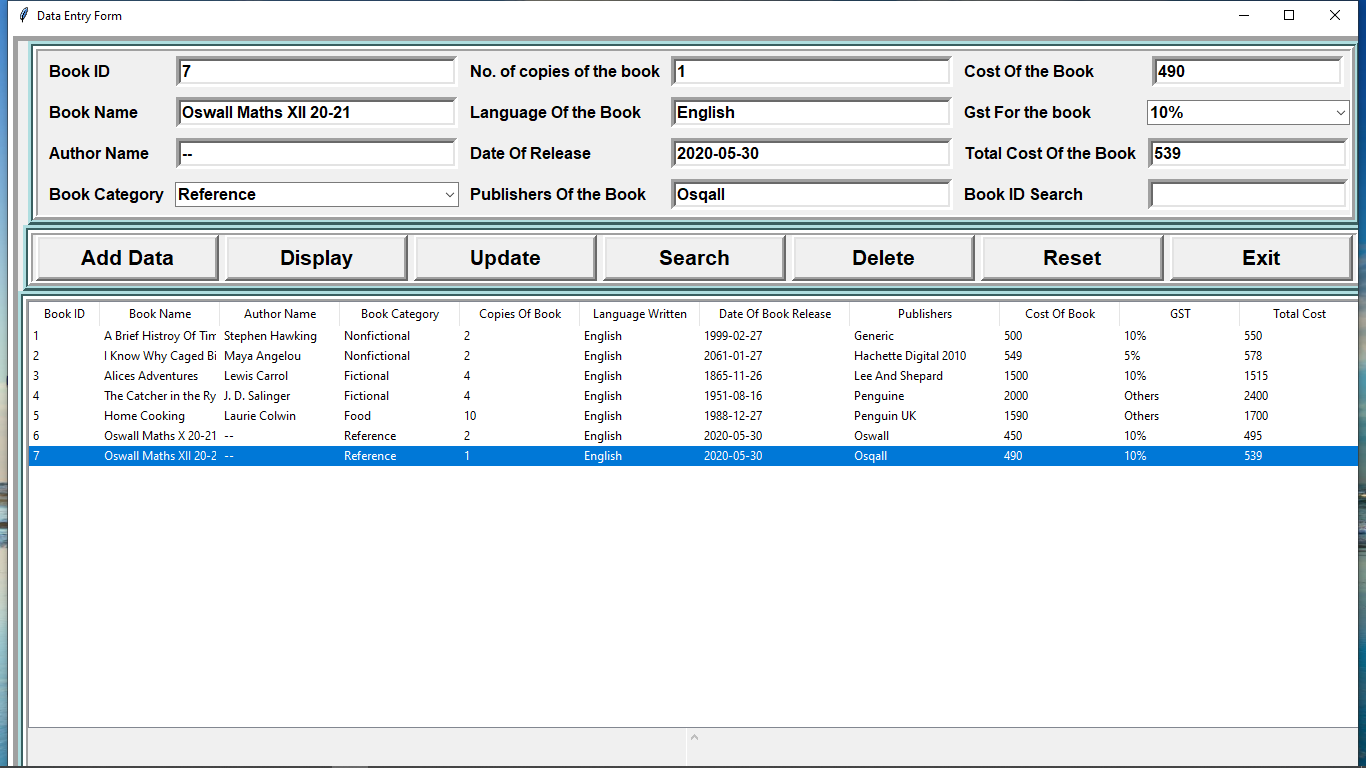
**OUTPUT**

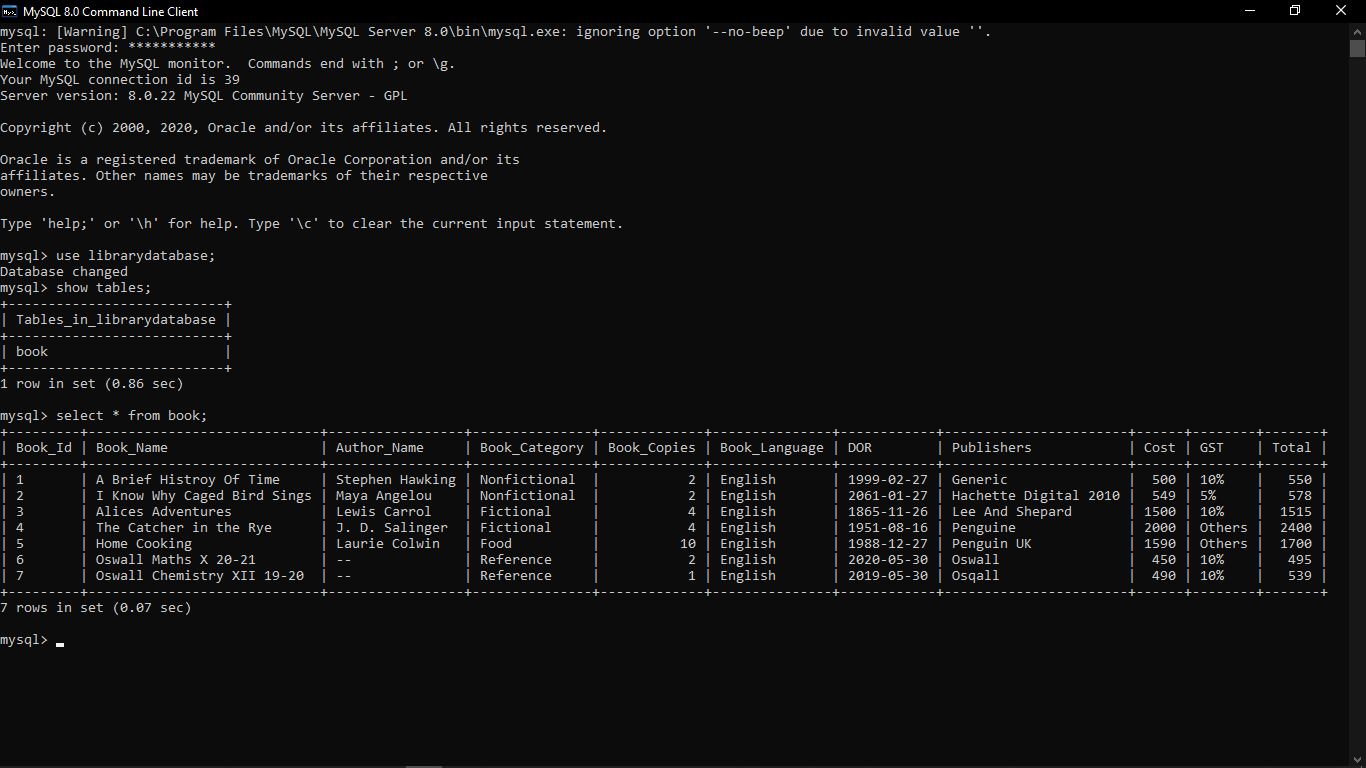
1.While opening up checks where the database as well as the table exists else create both database and table

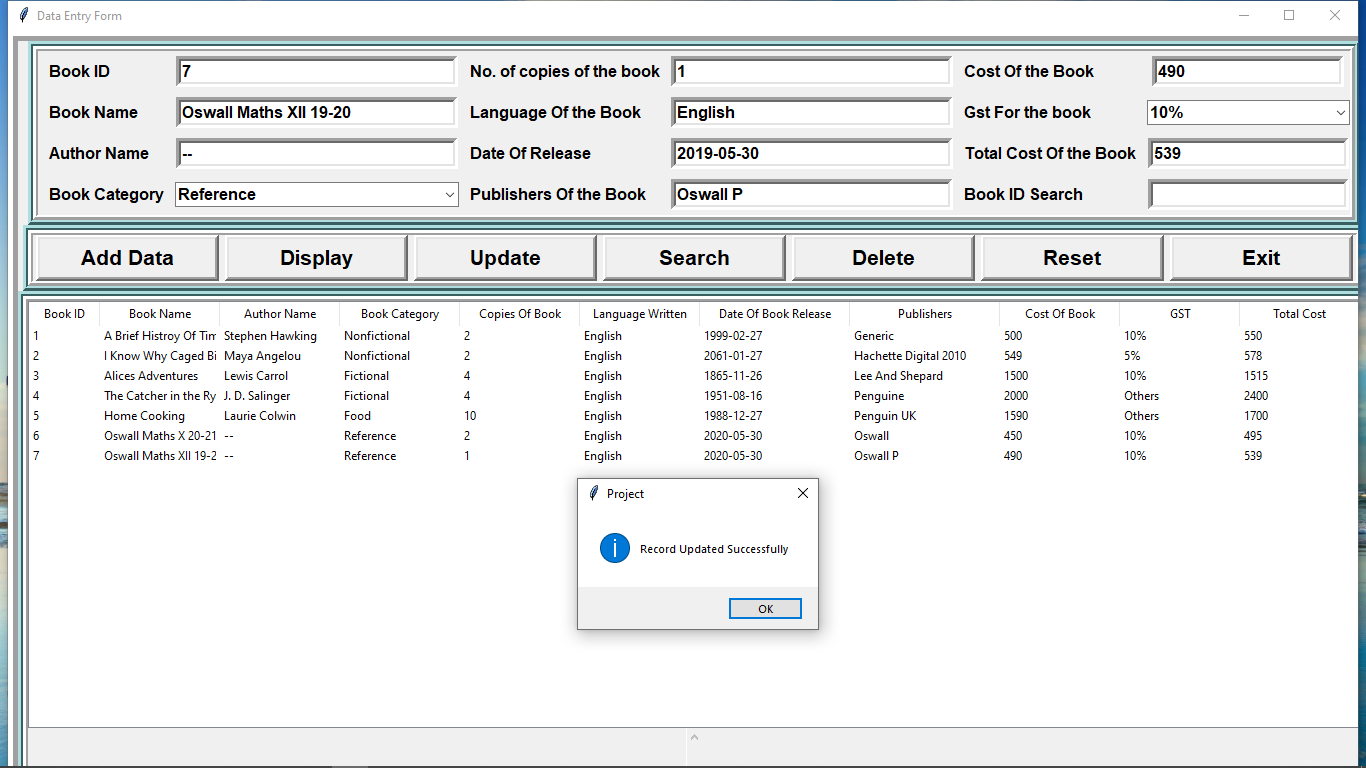
2.Opening Up Screen

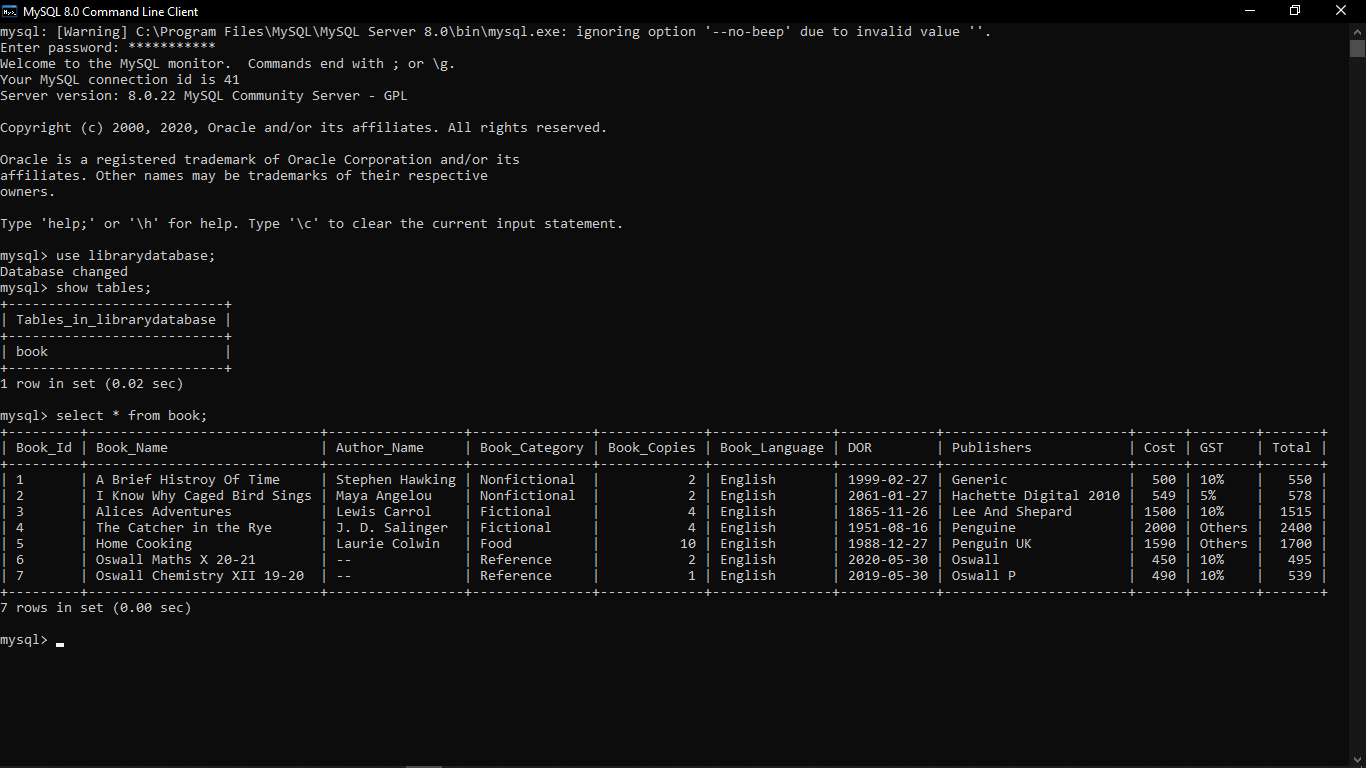
3.After Adding Data

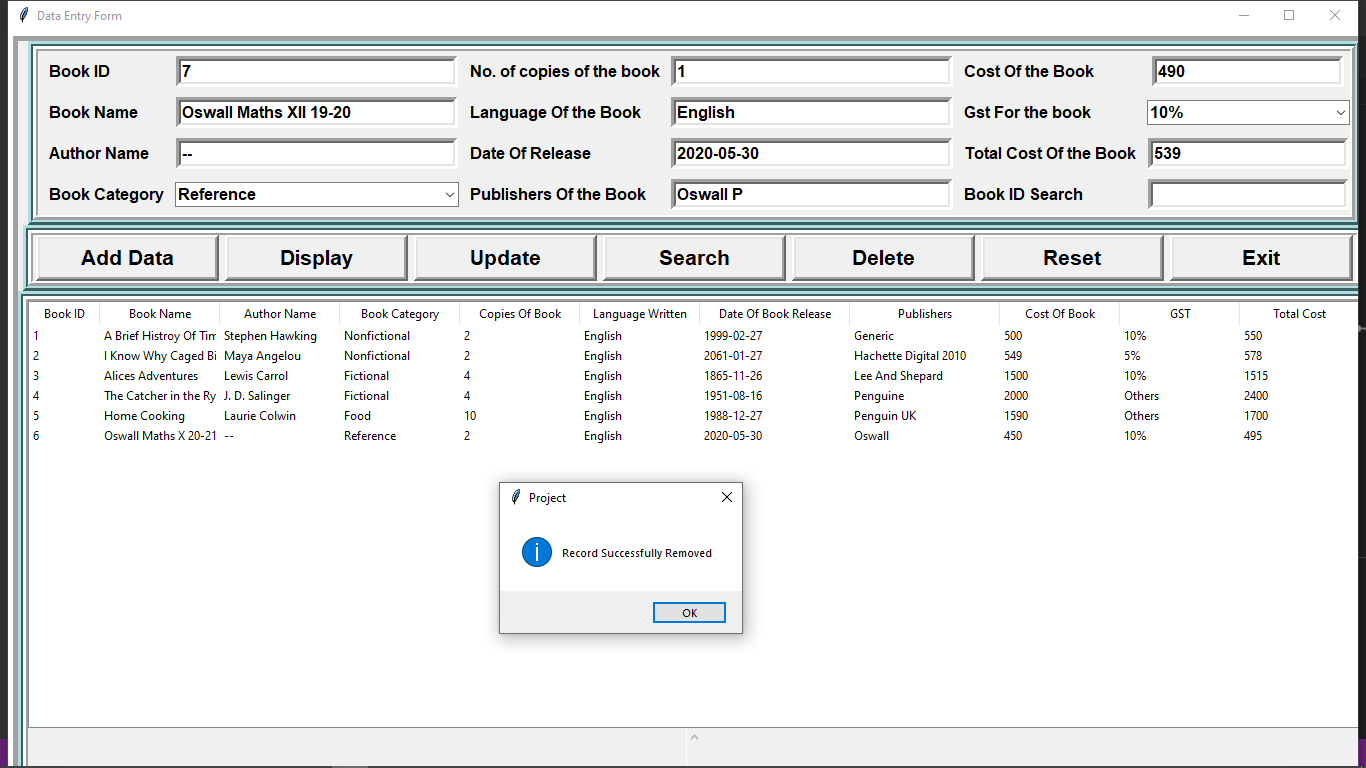
4.After adding 6 more entry

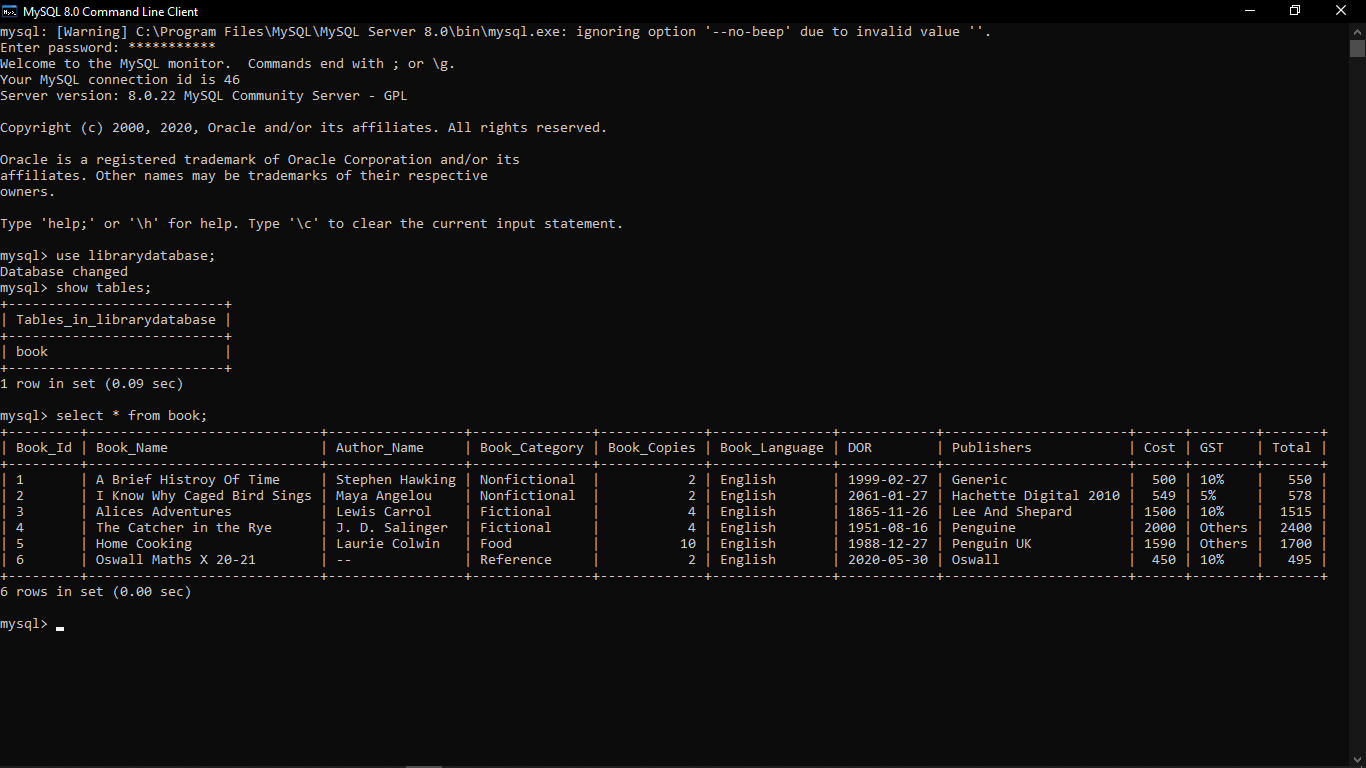
5.By clicking on the below table we can select the data

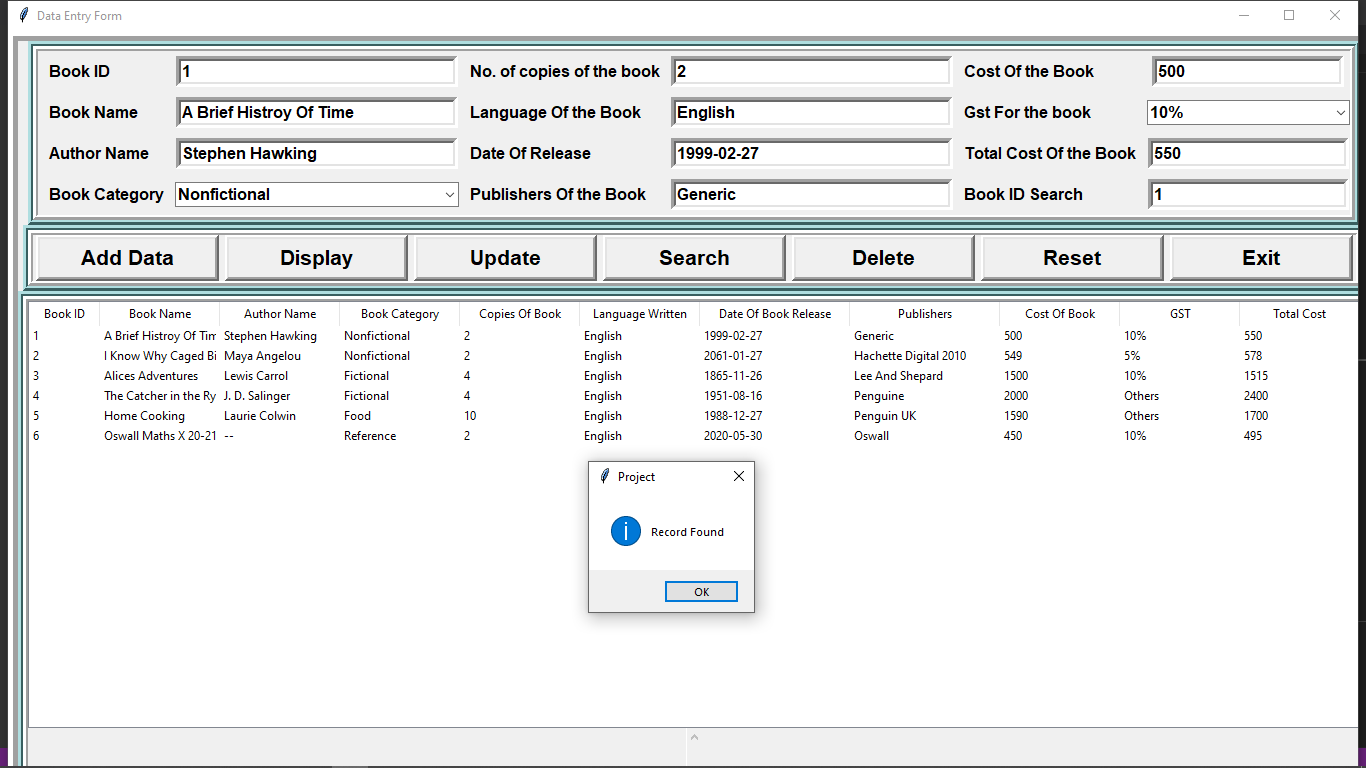
6.MySQL view of database

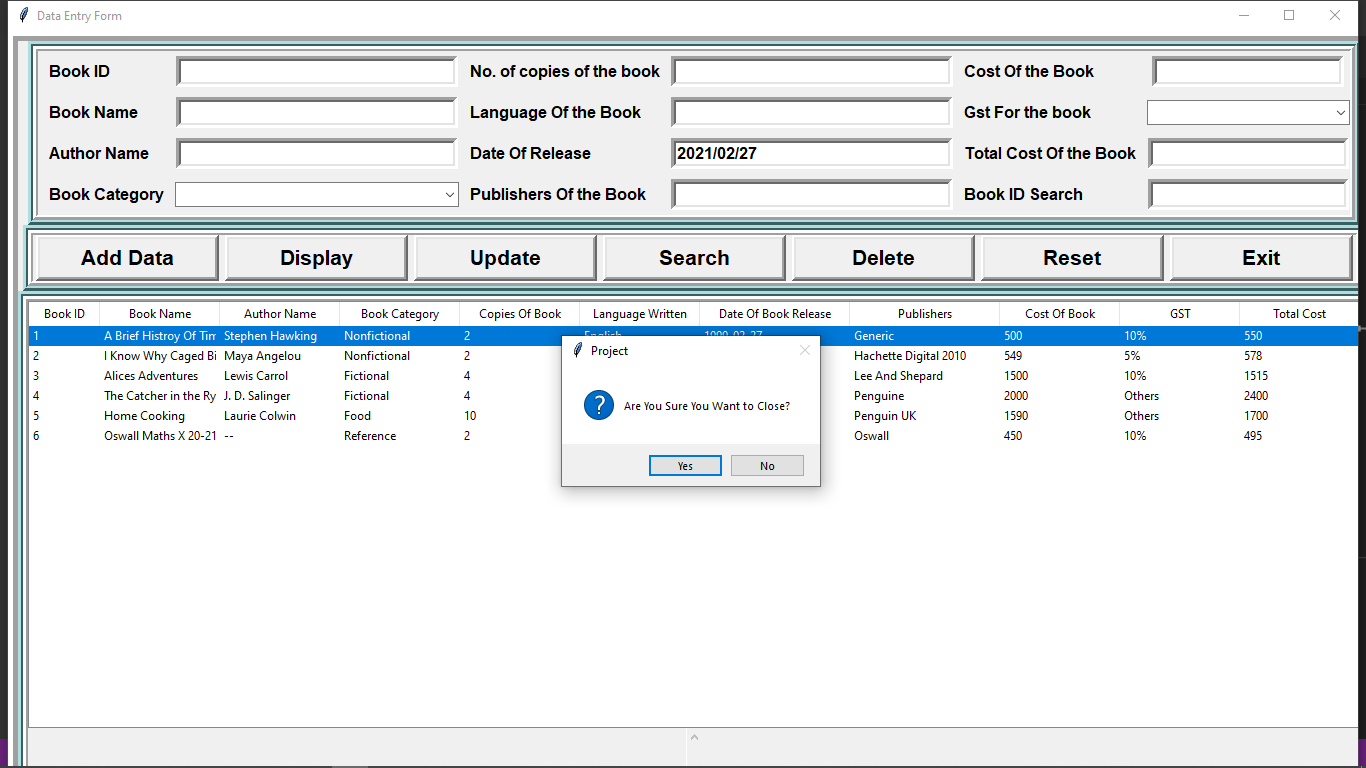
7.Updating the name of publisher from Owqal to Oswall P

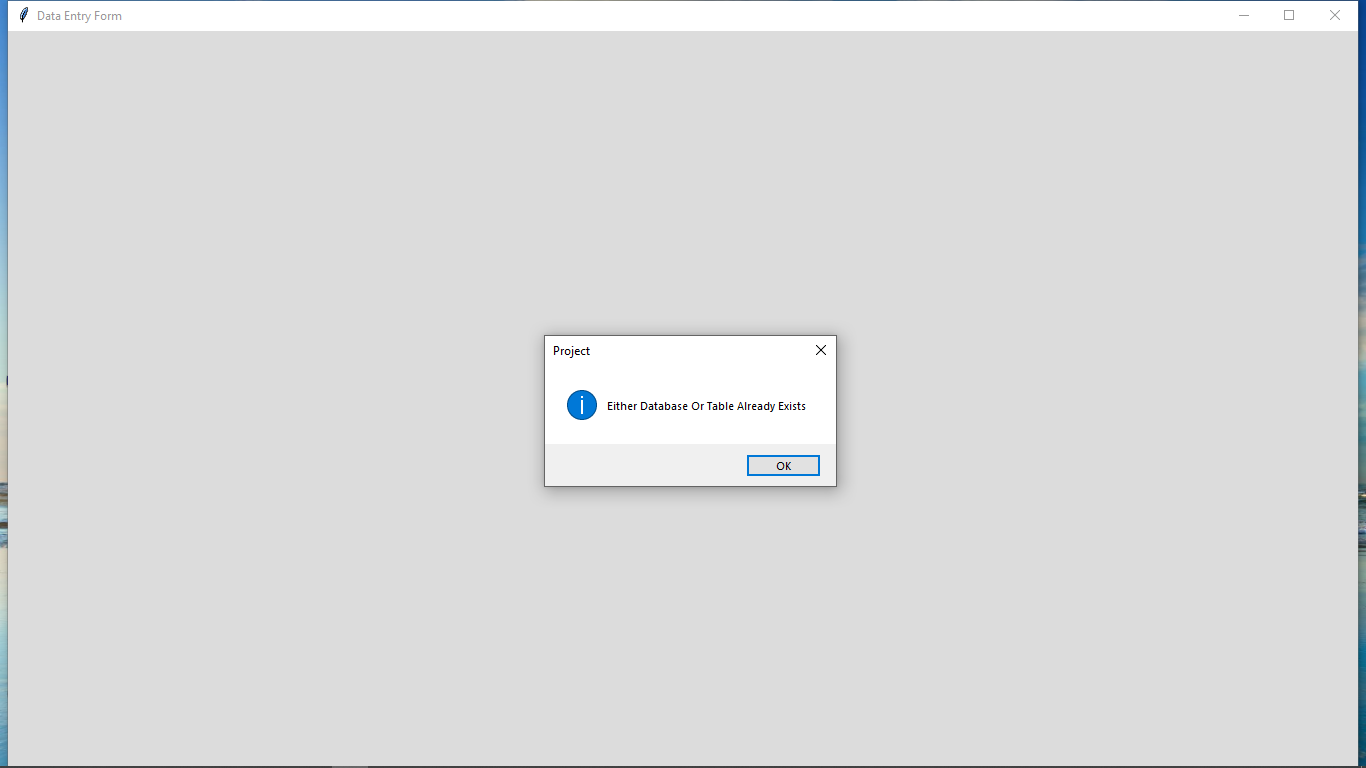
8.MySQL View after updating Owqal to Oswall P

9.Removal of data entry 7

10.MySQL View After deletion of data entry 7

11.Searching Data with Book ID:1

12.After using exit button confirmation

13.If file already exist

**CONCLUTION**

This program Is an operative and efficient prototype of creating, writing, displaying and updating library data. The program interface is simple and suits any type of users. It comes with every type of service that you need,

This project is programmed with Python because of its resourcefulness and flexibility. The program syntax is simple and even a novel user can comprehend the program without difficulty. In addition, the program offers simple formatting options and new options can be added easily.

**BIBLIOGRAPHY**

1. **Reference Text:**
   1. CBSE Class XII Computer Science with Python by Preeti Arora.
2. **Reference Websites:**
   1. [www.google.com](http://www.google.com)
   2. [www.wikipedia.com](http://www.wikipedia.com)
   3. [www.w3schools.com](http://www.w3schools.com)
   4. [www.codehelper.com](http://www.codehelper.com)