Indira Gandhi Delhi Technical University for Women

(Established by Govt. of Delhi vide Act 09 of 2012)

Kashmere Gate, Delhi – 110006

PROGRAMMING WITH PYTHON

PROJECT REPORT ON LIBRARY MANAGEMENT SYSTEM

Submitted To:-

Dr.Nonita ma'am

Submitted By:-

Neha Dhiman

Enrollment no.:-04004092023

MCA 1st SEM

TABLE OF CONTENTS

INTRODUCTION

This chapter gives an overview about the aim and objectives of the system.

1.1 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issuing.
- A view column to view the list of available books.
- Adding a book in library.
- Column to delete a book.

1.2 BACKGROUND OF PROJECT

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can add new books, issue books, and delete books.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

SYSTEM ANALYSIS

In this chapter, we will discuss and analyse about the developing process of Library Management System including software requirement specification (SRS) and software tools used.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.1.1 GENERAL DESCRIPTION

PRODUCT DESCRIPTION:

Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and timesaving

PROBLEM STATEMENT:

The problem occurred before having computerized system includes:

- *File lost* :-When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.
- *File damaged:-* When a computerized system is not there, file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage the files.
- *Difficult to search record:-* When there is no computerized system there is always a difficulty in searching of records if the records are large in number .
- *Space consuming:-* After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

- *Cost consuming:-* As there is no computerized system the to add each record paper will be needed which will increase the cost for the management of library.

PROCESSOR	**INTEL CORE PROCESSOR OR BETTER PERFORMANCE**
OPERATING SYSTEM	**WINDOWS 10 ,WINDOWS 11, UBUNTU**
MEMORY	**4GB RAM OR MORE**
HARD DISK SPACE	**MINIMUM 4 GB FOR DATABASE USAGE FOR FUTURE**
DATABASE	**MY SQL**

2.2 SOFTWARE TOOLS USED

The whole Project is divided in two parts the front end and the back end.

2.3.1 Front end

The front end is designed using Tkinter Library.

Tkinter Library:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python tkinter is the fastest and easiest way to create GUI applications. Creating a GUI using tkinter is an easy task.

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows and There are a number of widgets which you can put in your tkinter application.

2.3.2 Back end

The Back end of the server is designed using mysql server.

Mysql:

MYSQL- MySQL("My S-Q-L", officially, but also called "My Sequel") is (as ofJuly 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation .MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

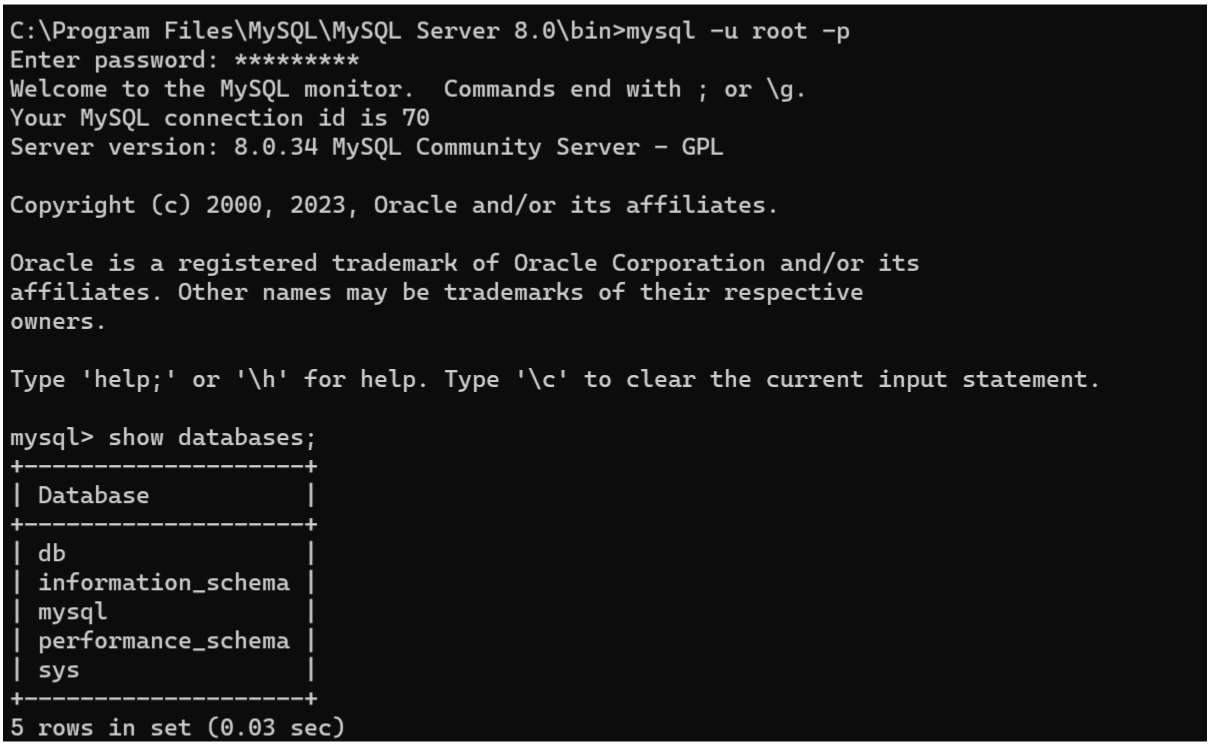
SYSTEM DESIGN

Through this project user(admin) will be able to perform the following task:-

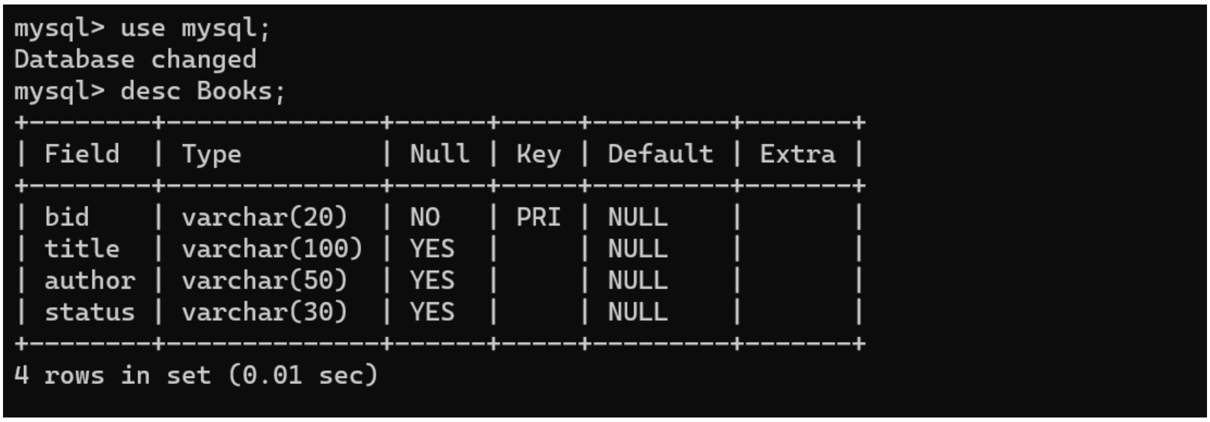
1. **Add Book:**
   If any student wants to donate/add a book in the library ,user can do that using add book feature .For this, user have to provide the following information like(book id, book name, book author, status) and by clicking on submit button the information will be stored successfully stored in the database.
2. **Issue Book:**
   User will enter the name of the student , book id of the book they want to issue. The status of the availability of the book will be checked through the book id .If it is available then can be issued otherwise user will get the message of non-availability of the book.
3. **Return Book:**
   User will provide the book id. By Clicking on the return book button the status of the book will get updated and the book will get returned.
4. **View Book:**
   To view the list of all the books present in the library. User can select view book option.
5. **Delete Book:**
   User have to provide book id in order to delete any book.

3.1 TABLE DESIGN

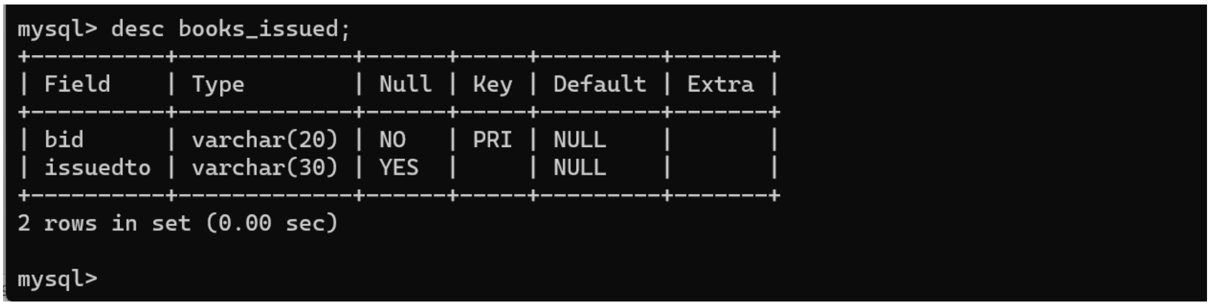
Tables to maintain information.



```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 70
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| db                 |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.03 sec)
```

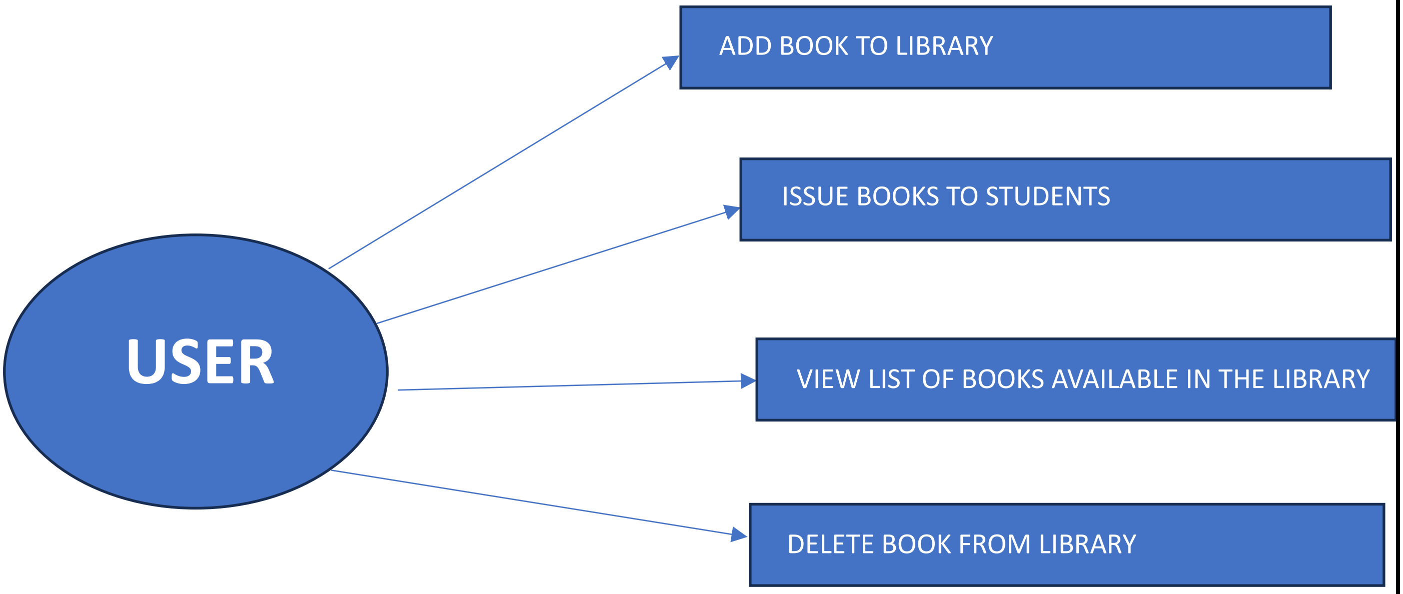


```
mysql> use mysql;
Database changed
mysql> desc Books;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| bid    | varchar(20)  | NO   | PRI | NULL    |       |
| title  | varchar(100) | YES  |     | NULL    |       |
| author | varchar(50)  | YES  |     | NULL    |       |
| status | varchar(30)  | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```



```
mysql> desc books_issued;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| bid      | varchar(20) | NO   | PRI | NULL    |       |
| issuedto | varchar(30) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql>
```

3.2 USE CASE DIAGRAM FOR ADMIN(USER)



SYSTEM IMPLEMENTATION

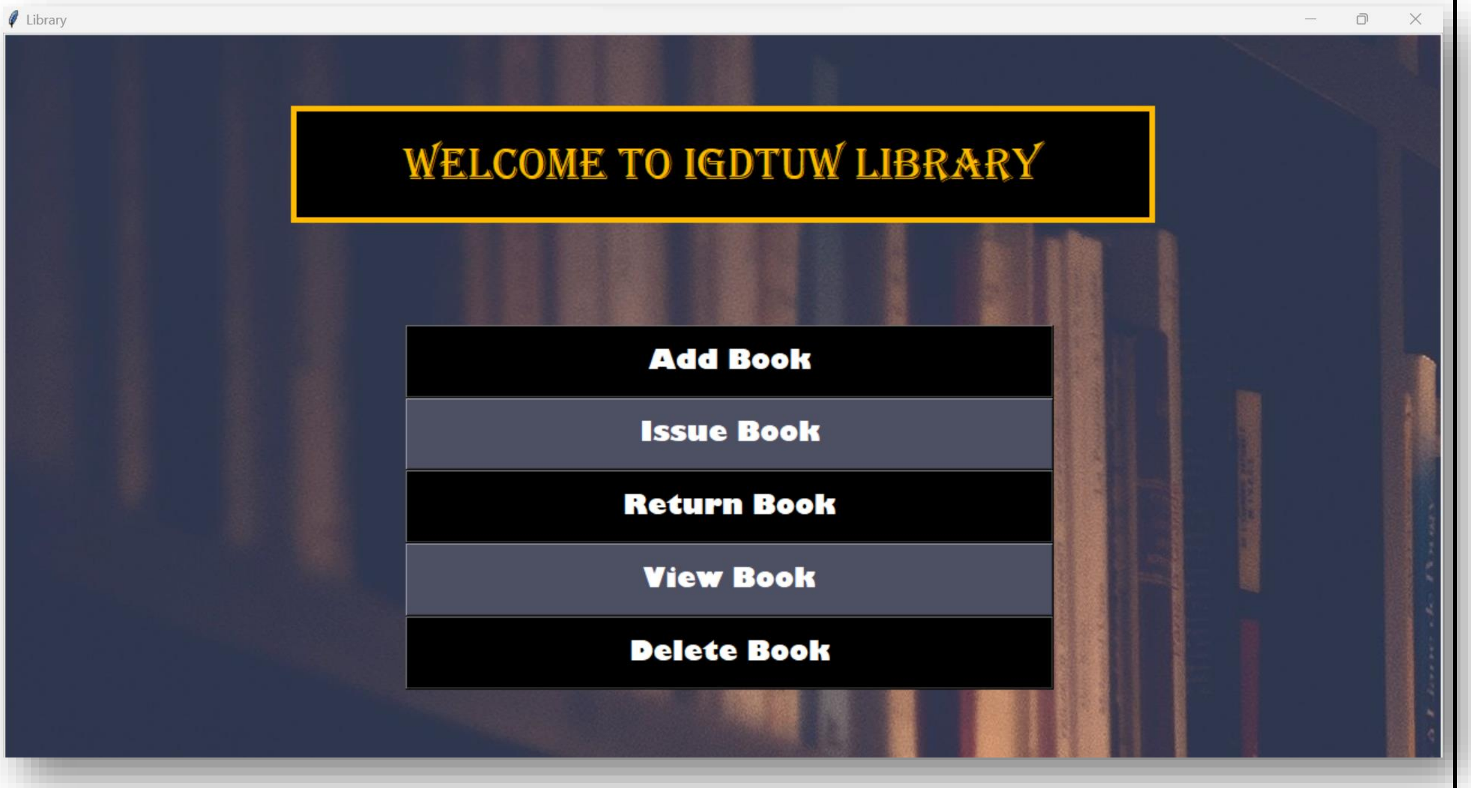
->Window.py/main

Code:-

```
from tkinter import *
from tkinter import messagebox
import pymysql
from ViewBooks import *
from IssueBook import*
from ReturnBook import*
from AddBook import*


mypass="Neha@2001"
mydatabase="mysql"

con=pymysql.connect(host="localhost",user="root",password=mypass,database=mydatabase)

cur=con.cursor()


root = Tk()
root.title("Library")
root.geometry("800x400")
bg=PhotoImage(file="lib.png")
Canvas1 = Canvas(root,width=400,height=200)
Canvas1.pack(fill="both",expand=True)
Canvas1.create_image(0,0,image=bg,anchor="nw")


#HEADING FRAMES
headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
headingFrame1.place(relx=0.2,rely=0.1,relwidth=0.6,relheight=0.16)
headingLabel = Label(headingFrame1, text="Welcome To IGDTUW Library",
bg='black', fg='#FFBB00', font=('Algerian',30))
```

```
headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

#ADDING BUTTONS
btn1=Button(root,text="Add Book",bg='black',
        fg='white',font=('Gill Sans Ultra Bold',18),command=addBook)
btn1.place(relx=0.28,rely=0.4,relwidth=0.45,relheight=0.1)


btn2=Button(root,text="Issue Book",bg='#4D4F63',
        fg='white',font=('Gill Sans Ultra Bold',18),command=issueBook)
btn2.place(relx=0.28,rely=0.5,relwidth=0.45,relheight=0.1)


btn3=Button(root,text="Return Book",bg='black',
        fg='white',font=('Gill Sans Ultra Bold',18),command=returnBook)
btn3.place(relx=0.28,rely=0.6,relwidth=0.45,relheight=0.1)


btn4=Button(root,text="View Book",bg='#4D4F63',
        fg='white',font=('Gill Sans Ultra Bold',18),command=View)
btn4.place(relx=0.28,rely=0.7,relwidth=0.45,relheight=0.1)


btn5=Button(root,text="Delete Book",bg='black',
        fg='white',font=('Gill Sans Ultra Bold',18))
btn5.place(relx=0.28,rely=0.8,relwidth=0.45,relheight=0.1)


root.mainloop()
```

Output:-



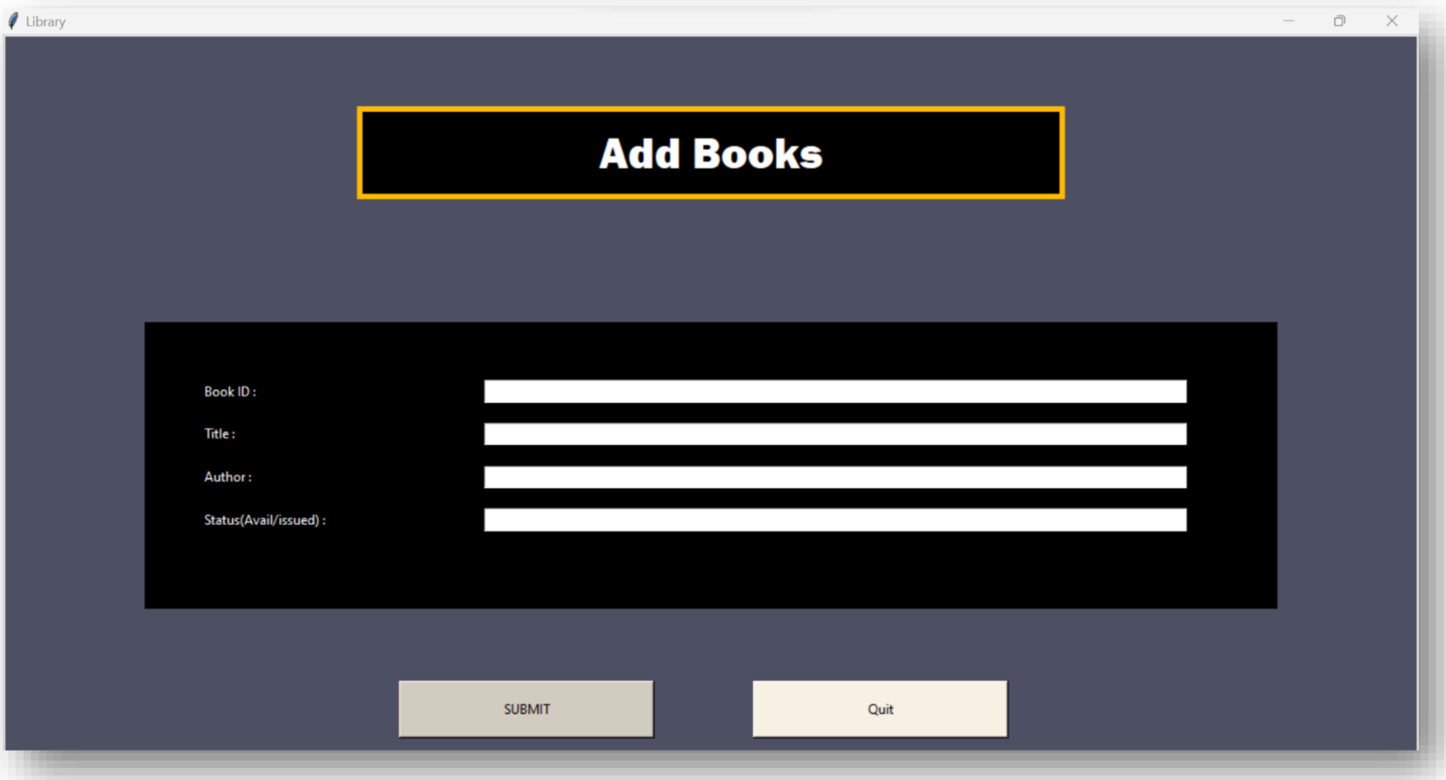
->AddBook.py

Code:-

```
from tkinter import *
from tkinter import messagebox
import pymysql

def bookRegister():

    bid = bookInfo1.get()
    title = bookInfo2.get()
    author = bookInfo3.get()
    status = bookInfo4.get()
    status = status.lower()

    insertBooks = "insert into "+bookTable+"
values('"+bid+"','"+title+"','"+author+"','"+status+"')"
    try:
        cur.execute(insertBooks)
        con.commit()
        messagebox.showinfo('Success',"Book added successfully")
    except:
        messagebox.showinfo("Error","Can't add data into Database")

    print(bid)
    print(title)
    print(author)
    print(status)


    root.destroy()

def addBook():

    global
bookInfo1,bookInfo2,bookInfo3,bookInfo4,Canvas1,con,cur,bookTable,root

    root = Tk()
    root.title("Library")
    root.geometry("600x500")


    mypass = "Neha@2001"
    mydatabase="mysql"

    con =
pymysql.connect(host="localhost",user="root",password=mypass,database=my
database)
    cur = con.cursor()


    bookTable = "books"

    Canvas1 = Canvas(root)

    Canvas1.config(bg="#4D4F63")
    Canvas1.pack(expand=True,fill=BOTH)

    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingLabel = Label(headingFrame1, text="Add Books", bg='black',
fg='white',font=('Franklin Gothic Heavy',30))
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)


    labelFrame = Frame(root,bg='black')
    labelFrame.place(relx=0.1,rely=0.4,relwidth=0.8,relheight=0.4)

    # Book ID
    lb1 = Label(labelFrame,text="Book ID : ", bg='black', fg='white')
    lb1.place(relx=0.05,rely=0.2, relheight=0.08)

    bookInfo1 = Entry(labelFrame)
    bookInfo1.place(relx=0.3,rely=0.2, relwidth=0.62, relheight=0.08)

    # Title
    lb2 = Label(labelFrame,text="Title : ", bg='black', fg='white')
    lb2.place(relx=0.05,rely=0.35, relheight=0.08)

    bookInfo2 = Entry(labelFrame)
    bookInfo2.place(relx=0.3,rely=0.35, relwidth=0.62, relheight=0.08)

    # Book Author
    lb3 = Label(labelFrame,text="Author : ", bg='black', fg='white')
    lb3.place(relx=0.05,rely=0.50, relheight=0.08)

    bookInfo3 = Entry(labelFrame)
    bookInfo3.place(relx=0.3,rely=0.50, relwidth=0.62, relheight=0.08)

    # Book Status
    lb4 = Label(labelFrame,text="Status(Avail/issued) : ", bg='black', fg='white')
    lb4.place(relx=0.05,rely=0.65, relheight=0.08)

    bookInfo4 = Entry(labelFrame)
    bookInfo4.place(relx=0.3,rely=0.65, relwidth=0.62, relheight=0.08)

    #Submit Button
```

```
   SubmitBtn = Button(root,text="SUBMIT",bg='#d1ccc0',
fg='black',command=bookRegister)
   SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)

   quitBtn = Button(root,text="Quit",bg='#f7f1e3', fg='black',
command=root.destroy)
   quitBtn.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

   root.mainloop()
```

Output:-

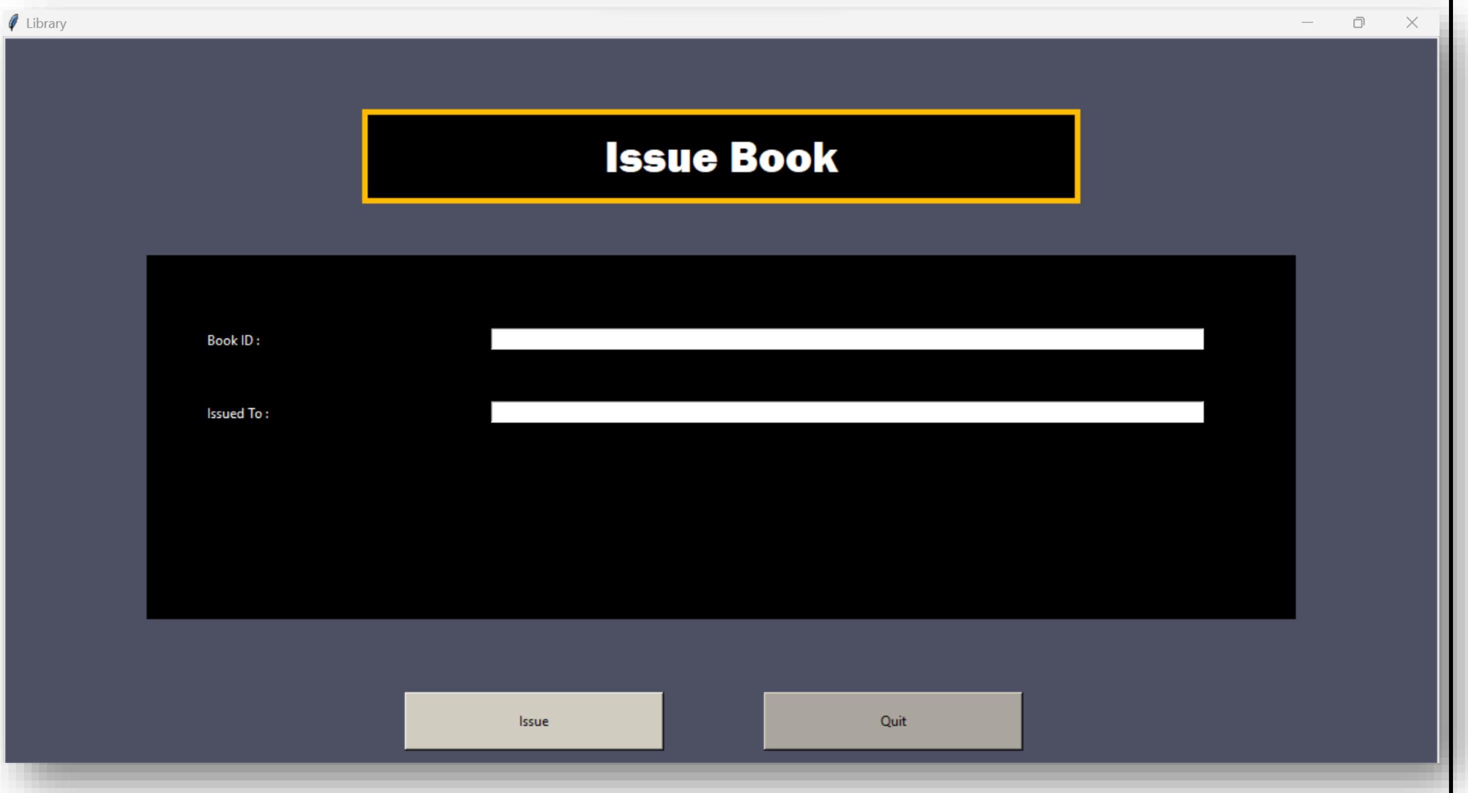


->IssueBook.py
Code:-

```
from tkinter import *
from tkinter import messagebox
import pymysql


mypass = "Neha@2001"
```

```
mydatabase="mysql"

con=pymysql.connect(host="localhost",user="root",password=mypass,databas
e=mydatabase)
cur = con.cursor()


issueTable = "books_issued"
bookTable = "books"


allBid = []

def issue():

    global issueBtn,labelFrame,lb1,inf1,inf2,quitBtn,root,Canvas1,status

    bid = inf1.get()
    issueto = inf2.get()

    issueBtn.destroy()
    labelFrame.destroy()
    lb1.destroy()
    inf1.destroy()
    inf2.destroy()


    extractBid = "select bid from "+bookTable
    try:
        cur.execute(extractBid)
        con.commit()
        for i in cur:
            allBid.append(i[0])

        if bid in allBid:
            checkAvail = "select status from "+bookTable+" where bid = '"+bid+"'"
            cur.execute(checkAvail)
            con.commit()
            for i in cur:
                check = i[0]
            if check == 'Avail' or 'avail':
                status = True
            else:
                status = False

        else:
            messagebox.showinfo("Error","Book ID not present")
    except:
        messagebox.showinfo("Error","Can't fetch Book IDs")

    issueSql = "insert into "+issueTable+" values ('"+bid+"','"+issueto+"')"
    show = "select * from "+issueTable

    updateStatus = "update "+bookTable+" set status = 'issued' where bid =
'"+bid+"'"
    try:
        if bid in allBid and status == True:
            cur.execute(issueSql)
            con.commit()
            cur.execute(updateStatus)
            con.commit()
            messagebox.showinfo('Success',"Book Issued Successfully")
            root.destroy()
        else:
            allBid.clear()
            messagebox.showinfo('Message',"Book Already Issued")
            root.destroy()
            return
    except:
        messagebox.showinfo("Search Error","The value entered is wrong, Try
again")

    print(bid)
    print(issueto)

    allBid.clear()

def issueBook():
    global issueBtn,labelFrame,lb1,inf1,inf2,quitBtn,root,Canvas1,status

    root = Tk()
    root.title("Library")
    root.geometry("600x500")

    Canvas1 = Canvas(root)
    Canvas1.config(bg="#4D4F63")
    Canvas1.pack(expand=True,fill=BOTH)

    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingLabel = Label(headingFrame1, text="Issue Book", bg='black',
 fg='white', font=('Franklin Gothic Heavy',30))
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

    labelFrame = Frame(root,bg='black')
    labelFrame.place(relx=0.1,rely=0.3,relwidth=0.8,relheight=0.5)

    # Book ID
    lb1 = Label(labelFrame,text="Book ID : ", bg='black', fg='white')
    lb1.place(relx=0.05,rely=0.2)

    inf1 = Entry(labelFrame)
    inf1.place(relx=0.3,rely=0.2, relwidth=0.62)


    # Issued To Student name
    lb2 = Label(labelFrame,text="Issued To : ", bg='black', fg='white')
    lb2.place(relx=0.05,rely=0.4)

    inf2 = Entry(labelFrame)
    inf2.place(relx=0.3,rely=0.4, relwidth=0.62)


    #Issue Button
    issueBtn = Button(root,text="Issue",bg='#d1ccc0', fg='black',command=issue)
    issueBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)
    quitBtn = Button(root,text="Quit",bg='#aaa69d', fg='black', command=root.destroy)
    quitBtn.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

    root.mainloop()
```

Output:-

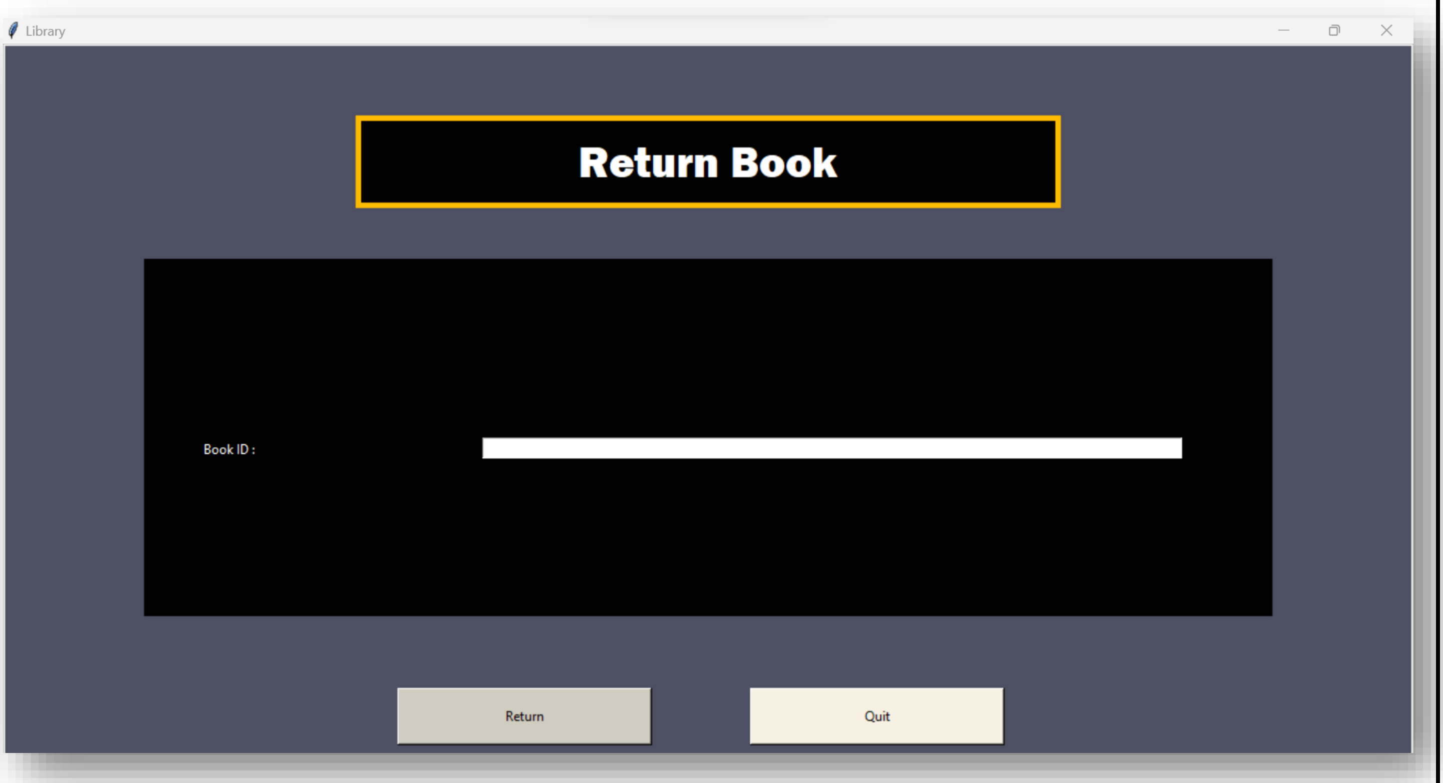


->ReturnBook.py
Code:-

```
from tkinter import *
from tkinter import messagebox
import pymysql


mypass = "Neha@2001"
mydatabase="mysql"
```

```
con =
pymysql.connect(host="localhost",user="root",password=mypass,database=my
database)
cur = con.cursor()


issueTable = "books_issued"
bookTable = "books"


allBid = []

def returnn():

    global SubmitBtn,labelFrame,lb1,bookInfo1,quitBtn,root,Canvas1,status

    bid = bookInfo1.get()

    extractBid = "select bid from "+issueTable
    try:
        cur.execute(extractBid)
        con.commit()
        for i in cur:
            allBid.append(i[0])

        if bid in allBid:
            checkAvail = "select status from "+bookTable+" where bid = '"+bid+"'"
            cur.execute(checkAvail)
            con.commit()
            for i in cur:
                check = i[0]

            if check == 'issued':
                status = True
            else:
                status = False

        else:
            messagebox.showinfo("Error","Book ID not present")
    except:
        messagebox.showinfo("Error","Can't fetch Book IDs")


    issueSql = "delete from "+issueTable+" where bid = '"+bid+"'"

    print(bid in allBid)
    print(status)
    updateStatus = "update "+bookTable+" set status = 'avail' where bid =
'"+bid+"'"
    try:
        if bid in allBid and status == True:
            cur.execute(issueSql)
            con.commit()
            cur.execute(updateStatus)
            con.commit()
            messagebox.showinfo('Success',"Book Returned Successfully")
        else:
            allBid.clear()
            messagebox.showinfo('Message',"Please check the book ID")
            root.destroy()
            return
    except:
        messagebox.showinfo("Search Error","The value entered is wrong, Try
again")


    allBid.clear()
    root.destroy()

def returnBook():

    global bookInfo1,SubmitBtn,quitBtn,Canvas1,con,cur,root,labelFrame, lb1

    root = Tk()
    root.title("Library")
    root.geometry("600x500")


    Canvas1 = Canvas(root)
    Canvas1.config(bg="#4D4F63")
    Canvas1.pack(expand=True,fill=BOTH)

    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingLabel = Label(headingFrame1, text="Return Book", bg='black',
fg='white',font=('Franklin Gothic Heavy',30) )
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

    labelFrame = Frame(root,bg='black')
    labelFrame.place(relx=0.1,rely=0.3,relwidth=0.8,relheight=0.5)

    # Book ID to Delete
    lb1 = Label(labelFrame,text="Book ID : ", bg='black', fg='white')
    lb1.place(relx=0.05,rely=0.5)

    bookInfo1 = Entry(labelFrame)
    bookInfo1.place(relx=0.3,rely=0.5, relwidth=0.62)

    #Submit Button
    SubmitBtn = Button(root,text="Return",bg='#d1ccc0',
fg='black',command=returnn)
    SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)

    quitBtn = Button(root,text="Quit",bg='#f7f1e3', fg='black',
command=root.destroy)
    quitBtn.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

    root.mainloop()
```

Output:-



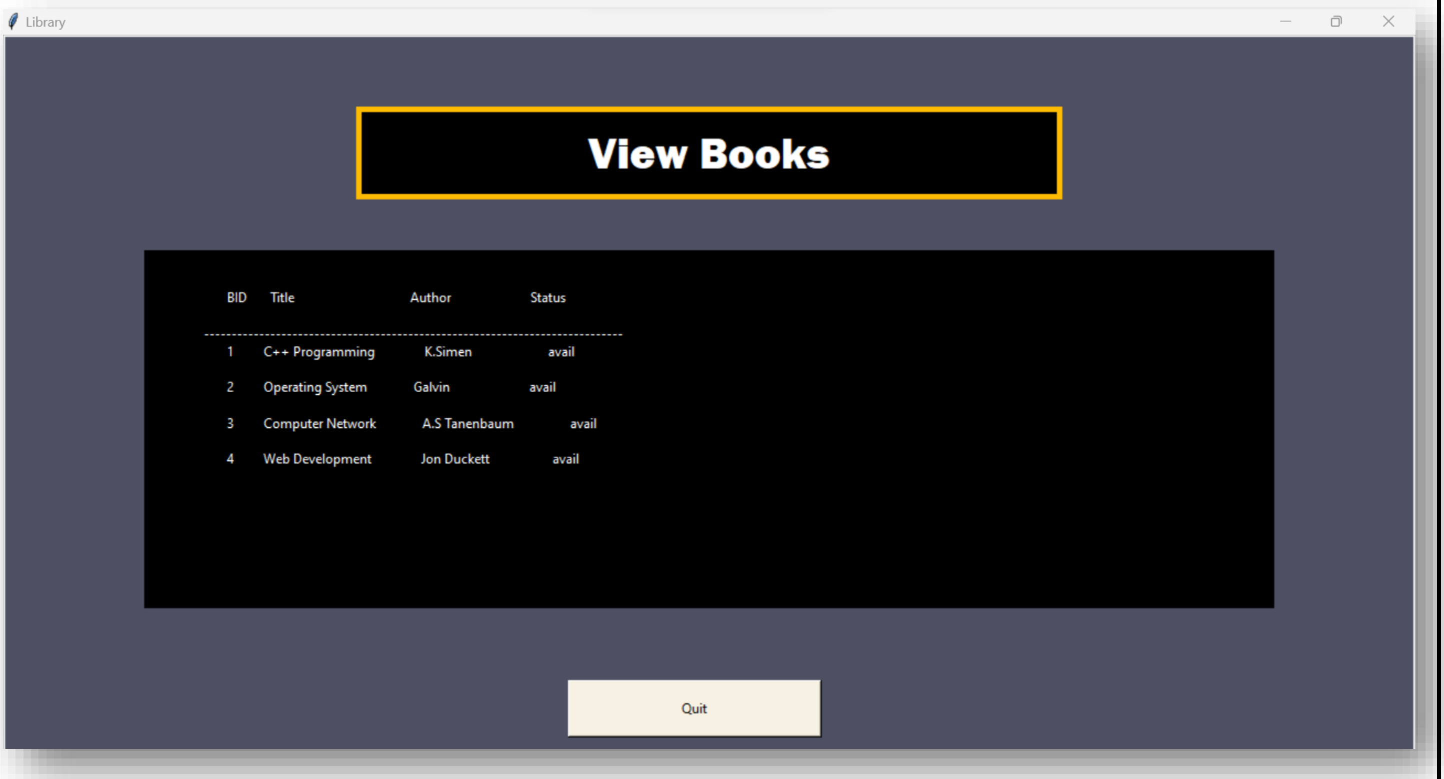
->ViewBook.py

Code:-

```
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox
import pymysql


mypass = "Neha@2001"
mydatabase="mysql"

con = pymysql.connect(host="localhost",user="root",password=mypass,database=mydatabase)
cur = con.cursor()


bookTable = "books"
```

```
def View():

    root = Tk()
    root.title("Library")
    root.minsize(width=400,height=400)
    root.geometry("600x500")


    Canvas1 = Canvas(root)
    Canvas1.config(bg="#4D4F63")
    Canvas1.pack(expand=True,fill=BOTH)


    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)


headingLabel = Label(headingFrame1, text="View Books", bg='black',
fg='white', font=('Franklin Gothic Heavy',30))
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)


    labelFrame = Frame(root,bg='black')
    labelFrame.place(relx=0.1,rely=0.3,relwidth=0.8,relheight=0.5)
    y = 0.25

    Label(labelFrame, text="%-10s%-40s%-30s%-
20s"%('BID','Title','Author','Status'),bg='black',fg='white').place(relx=0.07,rely=
0.1)
    Label(labelFrame, text="---------------------------------------------------
-------",bg='black',fg='white').place(relx=0.05,rely=0.2)
    getBooks = "select * from "+bookTable
    try:
        cur.execute(getBooks)
        con.commit()
        for i in cur:
            Label(labelFrame, text="%-10s%-30s%-30s%-
20s"%(i[0],i[1],i[2],i[3]),bg='black',fg='white').place(relx=0.07,rely=y)
            y += 0.1
    except:
        messagebox.showinfo("Failed to fetch files from database")

    quitBtn = Button(root,text="Quit",bg='#f7f1e3', fg='black',
command=root.destroy)
    quitBtn.place(relx=0.4,rely=0.9, relwidth=0.18,relheight=0.08)

    root.mainloop()
```

Output:-



```
BID   Title                  Author            Status
------------------------------------------------------------------
  1   C++ Programming          K.Simen            avail
  2   Operating System       Galvin            avail
  3   Computer Network         A.S Tanenbaum          avail
  4   Web Development         Jon Duckett          avail
```

->Delete.py

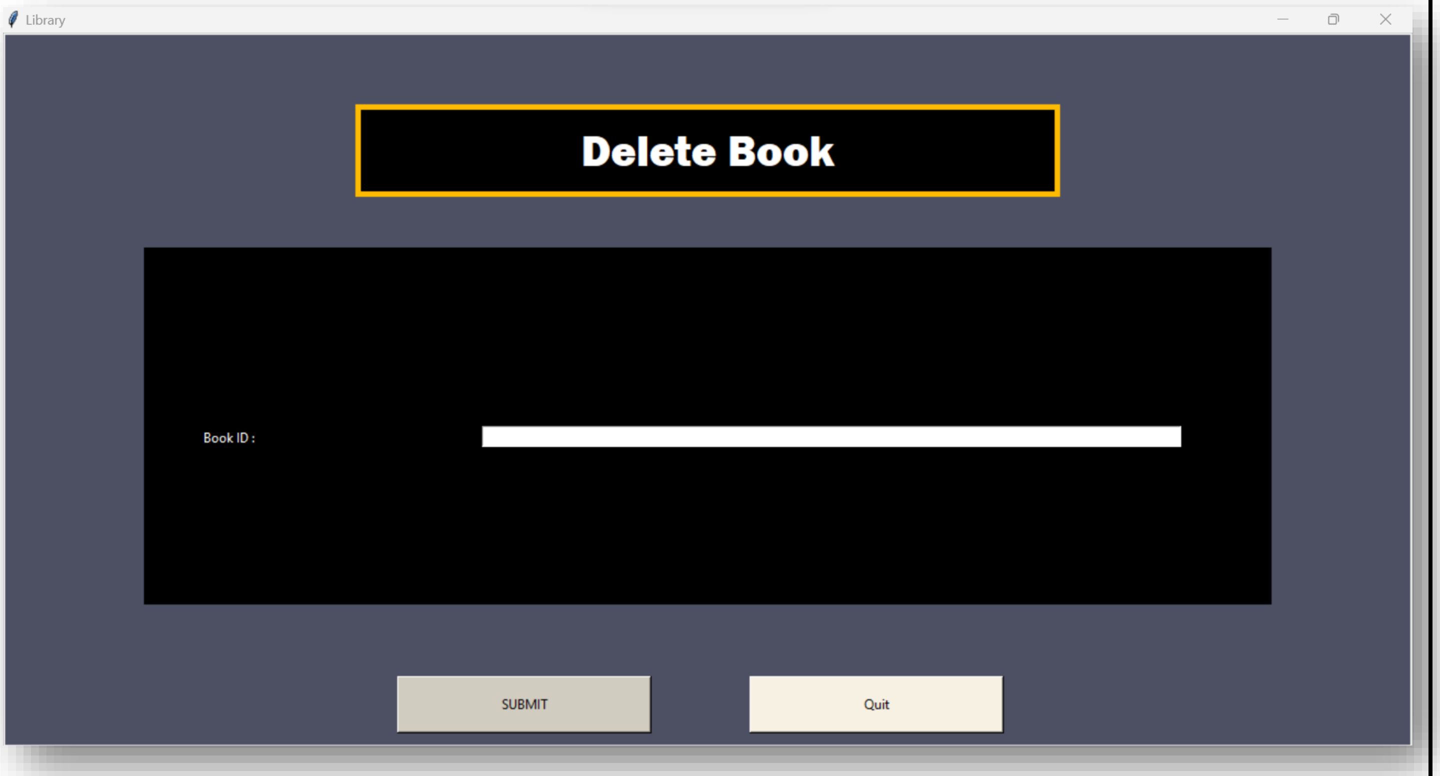
Code:-

```
from tkinter import *
from tkinter import messagebox
import pymysql


mypass = "Neha@2001"
mydatabase="mysql"
```

```
con =
pymysql.connect(host="localhost",user="root",password=mypass,database=my
database)
cur = con.cursor()


issueTable = "books_issued"
bookTable = "books"


def deleteBook():

    bid = bookInfo1.get()



    deleteSql = "delete from "+bookTable+" where bid = '"+bid+"'"
    deleteIssue = "delete from "+issueTable+" where bid = '"+bid+"'"
    try:
        cur.execute(deleteSql)
        con.commit()
        cur.execute(deleteIssue)
        con.commit()
        messagebox.showinfo('Success',"Book Record Deleted Successfully")
    except:
        messagebox.showinfo("Please check Book ID")


    print(bid)

    bookInfo1.delete(0, END)
    root.destroy()


def delete():

    global
bookInfo1,bookInfo2,bookInfo3,bookInfo4,Canvas1,con,cur,bookTable,root
```

```
  root = Tk()
    root.title("Library")
    root.minsize(width=400,height=400)
    root.geometry("600x500")


    Canvas1 = Canvas(root)
    Canvas1.config(bg="#4D4F63")
    Canvas1.pack(expand=True,fill=BOTH)


    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingLabel = Label(headingFrame1, text="Delete Book", bg='black', fg='white', font=('Franklin Gothic Heavy',30))
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

    labelFrame = Frame(root,bg='black')
    labelFrame.place(relx=0.1,rely=0.3,relwidth=0.8,relheight=0.5)


    lb2 = Label(labelFrame,text="Book ID : ", bg='black', fg='white')
    lb2.place(relx=0.05,rely=0.5)

    bookInfo1 = Entry(labelFrame)
    bookInfo1.place(relx=0.3,rely=0.5, relwidth=0.62)


    SubmitBtn = Button(root,text="SUBMIT",bg='#d1ccc0', fg='black',command=deleteBook)
    SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)


  quitBtn = Button(root,text="Quit",bg='#f7f1e3', fg='black', command=root.destroy)
    quitBtn.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

    root.mainloop()
```

Output:-



CONCLUSION AND FUTURE SCOPE

This website provides a computerized version of library management system which will benefit the students as well as the staff of the library. It makes entire process online where student can search books, staff can generate reports and do book transactions

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility , a feature Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfils each users need in the best way possible.

REFERENCES

- https://www.javatpoint.com/dbms-sql-command
- https://www.w3schools.com/sql/sql_syntax.asp
- https://www.geeksforgeeks.org/python-gui-tkinter/
- https://www.tutorialspoint.com/python/python_gui_programming.htm