****

**DOCUMENTATION**

**LIBRARY MANAGEMENT SYSTEM Batch No:10**

**Ch. Ani Siva Rama Krishna**

**121710319008**

**B19-08**

**ABSTRACT:**

The Library Management System is an application for assisting a librarian in managing a book library in university. The system would provide a basic set of features to add/update members, add/update books, and manage check in specifications for the systems based on the client’s statement of need.

Library management system is a typical management Information system (MIS), its Development includes the establishment and maintenance of back-end database and front-end application development aspects. For the former require the establishment of data consistency and integrity of the strong data security and good libraries. As for the latter requires the application fully functional, easy to use and so on.

**INTRODUCTION:**

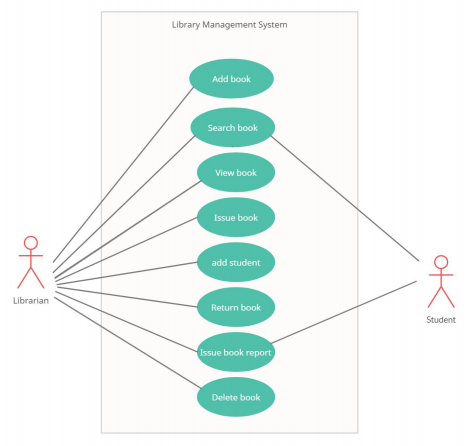
The project is titled as Library management system.

Management software for monitoring and controlling the transactions in a library .The project “Library Management System” is developed in Vscode IDE, which mainly focuses on basic operations in a library like adding new books, and updating new information, searching books and members and returning books.

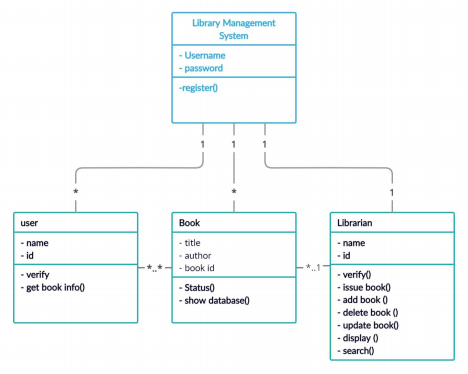
This project of “LIBRARY MANAGEMENT SYSTEM” gives us the complete information about the library. We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records.

**SYSTEM ARCHITECTURE:**

**USE CASE DIAGRAM:**

****

**CLASS DIAGRAM:**

****

**TOOLS:**

The IDE which are used in LIBRARY MANAGEMENT SYSTEM are IDE: vscode, Language used is python, Database used is my sql

**ABOUT TOOLS:**

**VSCODE:**

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

This source code editor can be used with a variety of programming languages.

**Python:**

Programming language we used in this project is python, using microsoft python editor which makes it an excellent python editor. First of all you have to install a Python interpreter after

installing, activate it using the python: select interpreter command. If vscode doesn’t locate the interpreter automatically refer to the environment manually specify an interpreter.

**MYSQL:**

MySQL is an open-source relational database management system (RDBMS). ... MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL. We used Mysql database to store the information of books, librarian, Authors.

**CODING:**

**MAIN PAGE:**

from tkinter import \*

from PIL import ImageTk,Image

import pymysql

from tkinter import messagebox

from AddBook import \*

from DeleteBook import \*

from ViewBooks import \*

from IssueBook import \*

from ReturnBook import \*

mypass = "root"

mydatabase="db"

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

root = Tk()

root.title("Library")

root.minsize(width=400,height=400)

root.geometry("600x500")

same=True

n=0.25

background\_image =Image.open("lib.jpg")

[imageSizeWidth, imageSizeHeight] = background\_image.size

newImageSizeWidth = int(imageSizeWidth\*n)

if same:

newImageSizeHeight = int(imageSizeHeight\*n)

else:

newImageSizeHeight = int(imageSizeHeight/n)

background\_image =

background\_image.resize((newImageSizeWidth,newImageSizeHeight),Image.ANTIALIAS) img = ImageTk.PhotoImage(background\_image)

Canvas1 = Canvas(root)

Canvas1.create\_image(300,340,image = img)

Canvas1.config(bg="white",width = newImageSizeWidth, height = newImageSizeHeight) Canvas1.pack(expand=True,fill=BOTH)

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 \n DataFlair Library", bg='black', fg='white', font=('Courier',15))

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

btn1 = Button(root,text="Add Book Details",bg='black', fg='white', command=addBook) btn1.place(relx=0.28,rely=0.4, relwidth=0.45,relheight=0.1)

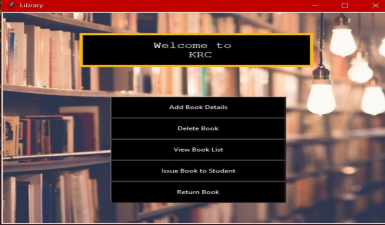
btn2 = Button(root,text="Delete Book",bg='black', fg='white', command=delete) btn2.place(relx=0.28,rely=0.5, relwidth=0.45,relheight=0.1)

btn3 = Button(root,text="View Book List",bg='black', fg='white', command=View) btn3.place(relx=0.28,rely=0.6, relwidth=0.45,relheight=0.1)

btn4 = Button(root,text="Issue Book to Student",bg='black', fg='white', command = issueBook) btn4.place(relx=0.28,rely=0.7, relwidth=0.45,relheight=0.1)

btn5 = Button(root,text="Return Book",bg='black', fg='white', command = returnBook) btn5.place(relx=0.28,rely=0.8, relwidth=0.45,relheight=0.1)

root.mainloop()



**ADD BOOK:**

from tkinter import \*

from PIL import ImageTk,Image

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.minsize(width=400,height=400)

root.geometry("600x500")

mypass = "root"

mydatabase="db"

con =

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

bookTable = "books" # Book Table

Canvas1 = Canvas(root)

Canvas1.config(bg="#ff6e40")

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=('Courier',15))

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)

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)

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)

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)

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

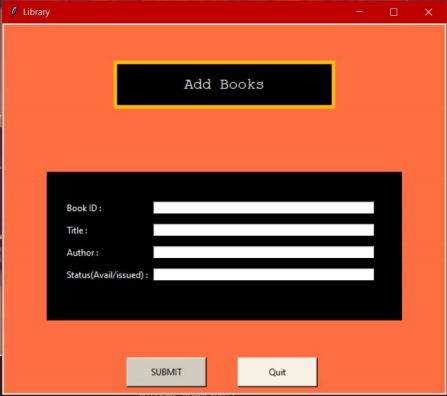
bookInfo4 = Entry(labelFrame)

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

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()



**DELETE BOOK:**

from tkinter import \*

from PIL import ImageTk,Image

from tkinter import messagebox

import pymysql

mypass = "root"

mydatabase="db"

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

issueTable = "books\_issued"

bookTable = "books" #Book Table

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="#006B38")

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=('Courier',15))

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)

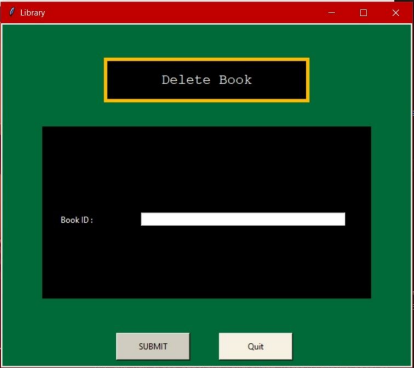
bookInfo1 = Entry(labelFrame)

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()

**VIEW BOOK LIST:**

from tkinter import \*

from PIL import ImageTk,Image

from tkinter import messagebox

import pymysql

mypass = "root"

mydatabase="db"

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="#12a4d9")

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=('Courier',15))

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,

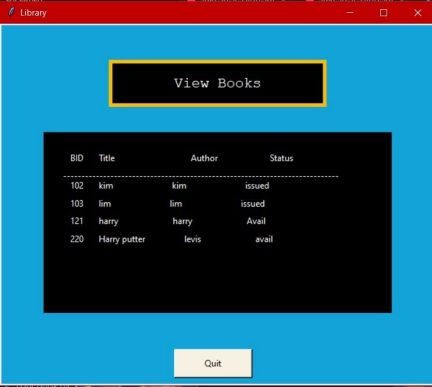
text="%-10s%-30s%-30s%-20s"%(i[0],i[1],i[2],i[3]),bg='black',fg='white').place(relx=0.07,rely=y) y += 0.1

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()



**ISSUE BOOK TO STUDENT:**

from tkinter import \*

from PIL import ImageTk,Image

from tkinter import messagebox

import pymysql

mypass = "root"

mydatabase="db"

con = pymysql.connect(host="localhost",user="root",password=mypass,database=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':

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.minsize(width=400,height=400)

root.geometry("600x500")

Canvas1 = Canvas(root)

Canvas1.config(bg="#D6ED17")

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=('Courier',15))

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)

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)

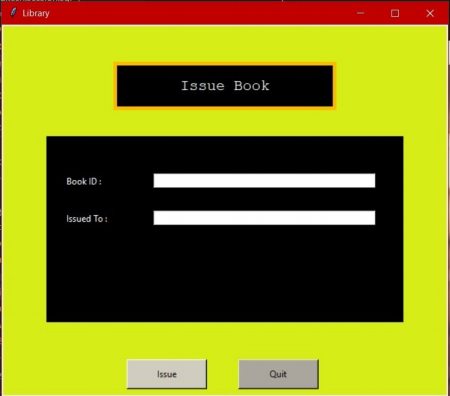
inf2 = Entry(labelFrame)

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

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()



**RETURN BOOK:**

from tkinter import \*

from PIL import ImageTk,Image

from tkinter import messagebox

import pymysql

mypass = "root"

mydatabase="db"

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

]

issueTable = "books\_issued" #Issue Table

bookTable = "books" #Book Table

allBid = [] #List To store all Book IDs

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.minsize(width=400,height=400)

root.geometry("600x500")

Canvas1 = Canvas(root)

Canvas1.config(bg="#006B38")

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=('Courier',15))

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)

lb1 = Label(labelFrame,text="Book ID : ", bg='black', fg='white')

lb1.place(relx=0.05,rely=0.5)

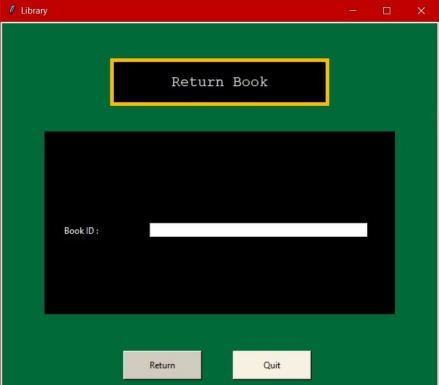
bookInfo1 = Entry(labelFrame)

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

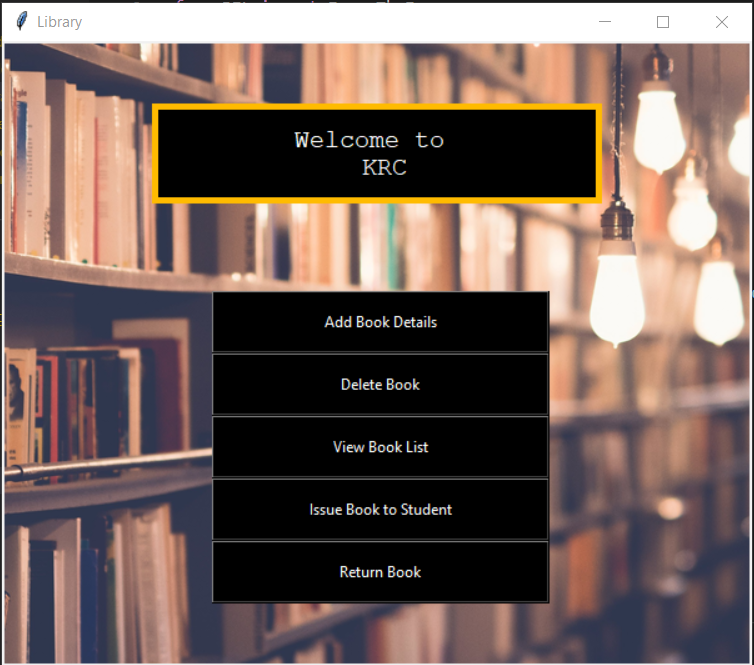
SubmitBtn = Button(root,text="Return",bg='#d1ccc0', fg='black',command=returnn) jjjjjSubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)

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

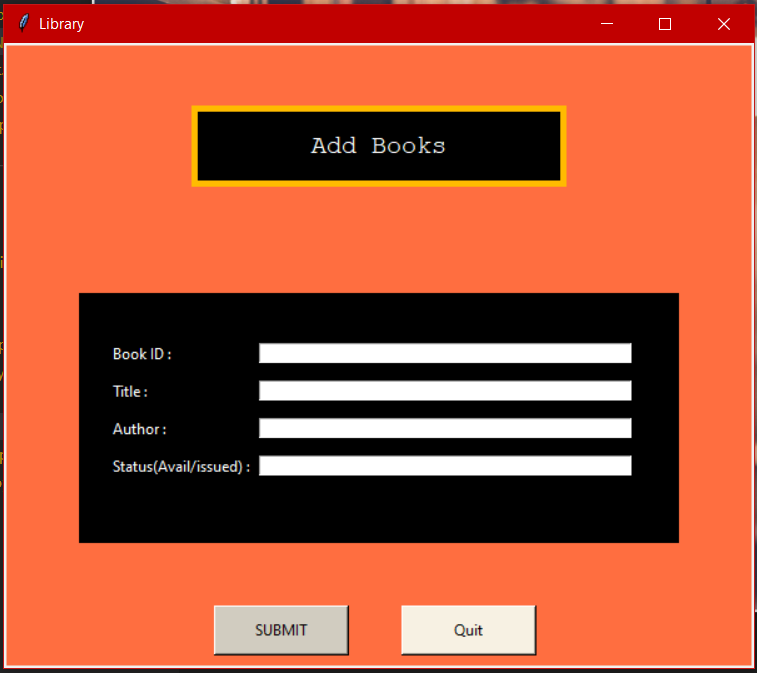
root.mainloop()

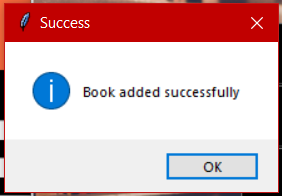


**OUTPUT:**

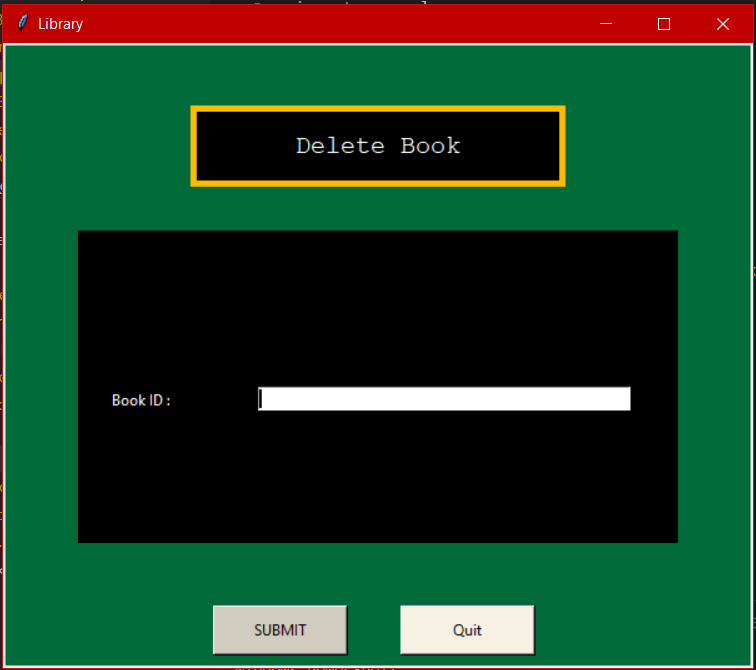
****

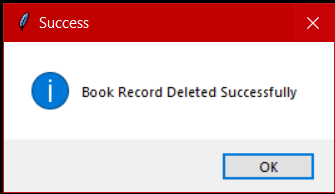
**Add book:**

****

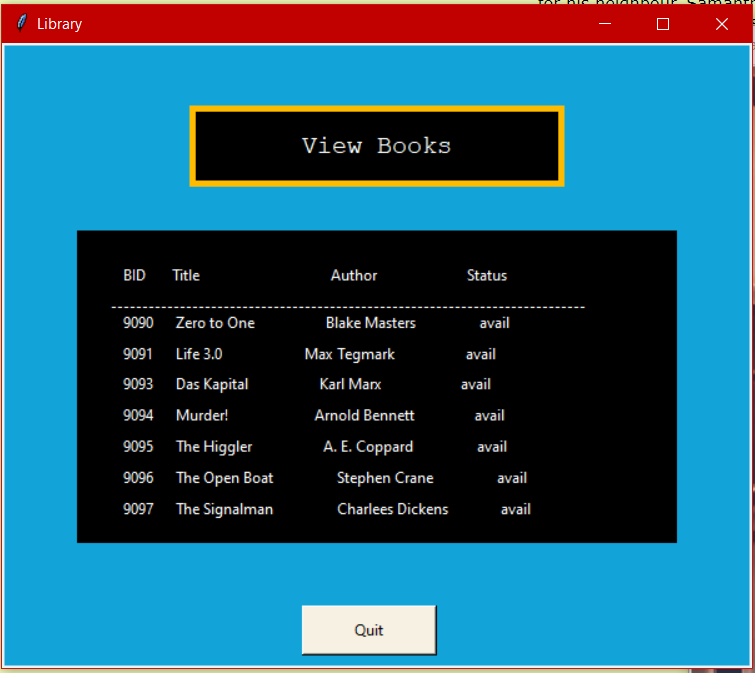
****

**Delete book:**

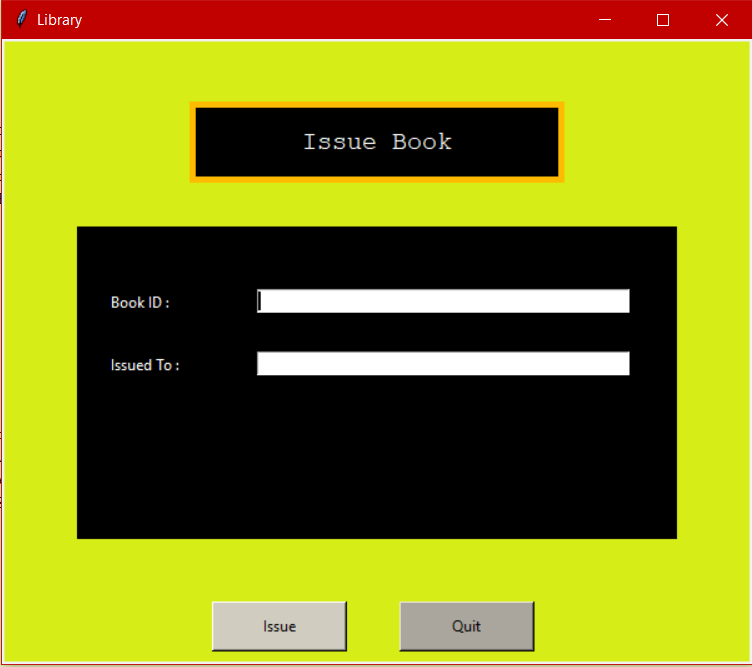
****

****

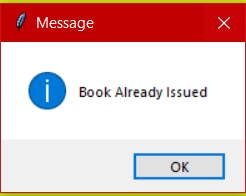
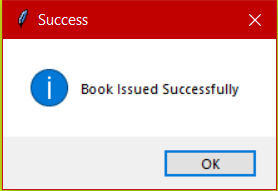
**View book list:**

****

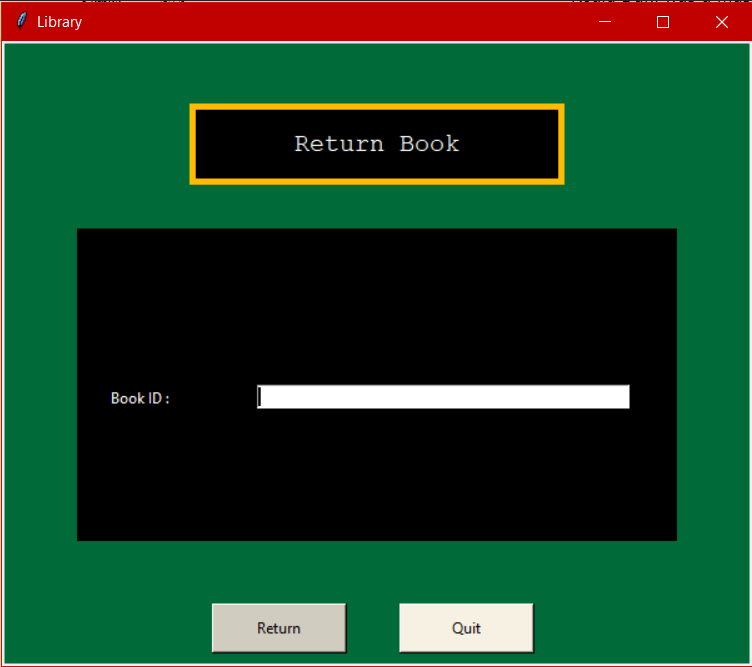
**Issuing book:**

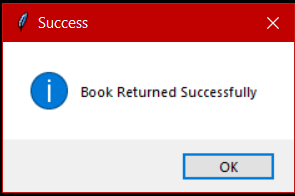
****

**If already issued:**

****

**Return book:**

****

****

**CONCLUSION:**

The project Library management system is for computerising the work in a library. The software takes care of all the requirements of a library and is capable of providing easy and effective storage of information related to books and users.

The library management system ultimately helps in smooth functioning of the library. It helps librarians to add books, delete books, issue books, and view book lists.