ABOUT THE PROJECT

A **library management system** is a software that is designed to manage all the functions of a library. It helps librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.

This system completely automates all your library's activities. The best way to maintain, organize, and handle countless books systematically is to implement a library management system software.

A library management system is used to maintain library records. It tracks the records of the number of books in the library, how many books are issued, or how many books have been returned, etc.

You can find books in an instant, issue/reissue books quickly, and manage all the data efficiently and orderly using this system. The purpose of a library management system is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.

HARDWARE AND SOFTWARE REQUIREMENTS

1. Operating System: Windows 7 and above

2. RAM: 4GB and above

3. Python 2.7.18 version and above

with Tkinter and mysql.connector module

4. MySQL 8.0.11 version and above

5. Microsoft Visual C++ Redistributable for Visual Studio 2015

INSTALLATION PROCEDURE

1. **Open MySQL Command Line Client** and enter the first code (given on Page: 8) in it.

2. **Open Python IDLE** and stepwise enter the

Code given in this document (Code: 2 to 7) (Page: 10 to 36) in it.

3. **Save the required files** with proper names where Python is installed.

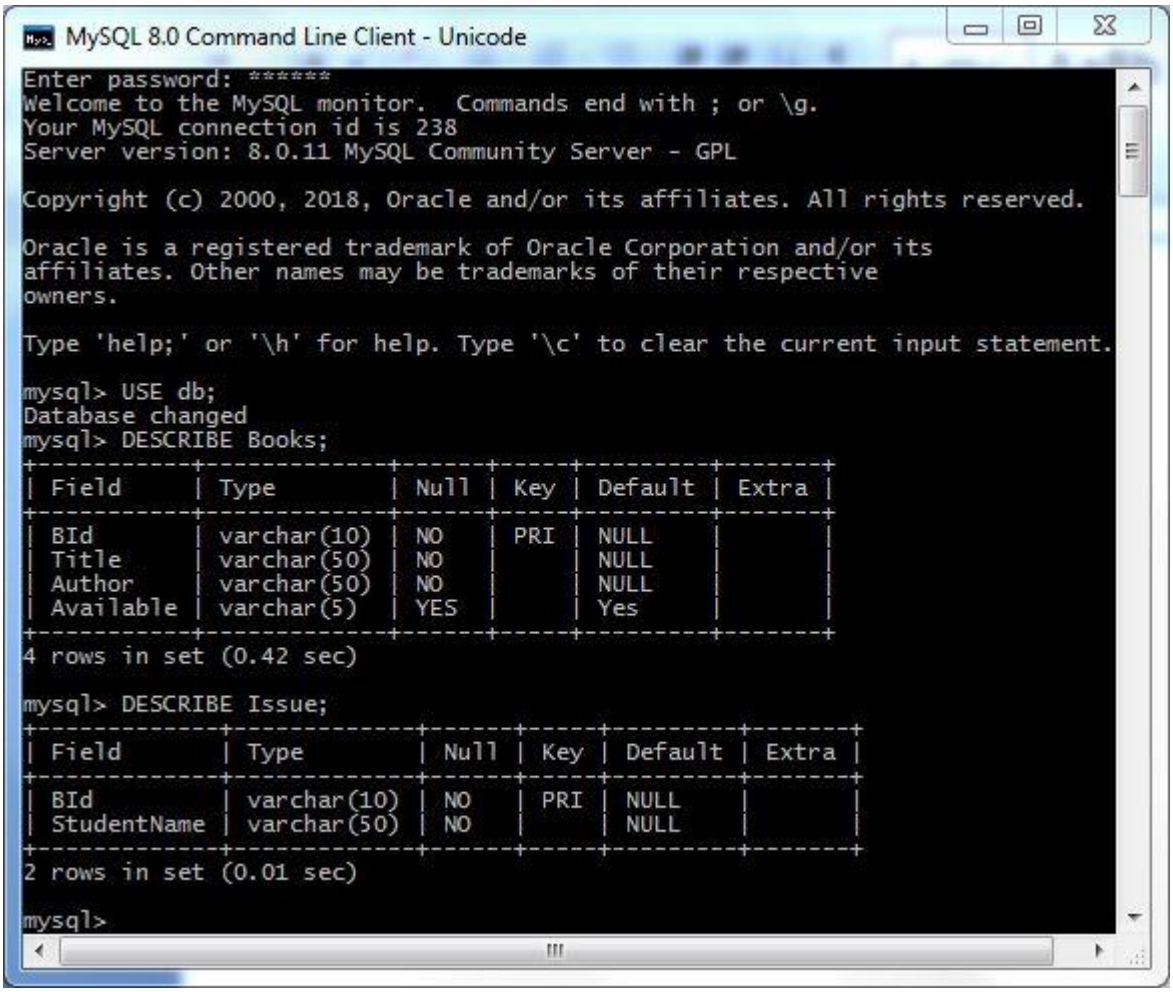
4. **Open home.py** and **click on F5**.

SOURCE CODE:

1. Run this code in MySQL Command Line Client

```
CREATE DATABASE db;

USE db;


CREATE TABLE Books (

BId varchar(10) PRIMARY KEY,

Title varchar(50) NOT NULL UNIQUE,

Author varchar(50) NOT NULL,

Available varchar(5) DEFAULT‘YES’);


CREATE TABLE Issue (

BId varchar(10) PRIMARY KEY,

Student Name varchar(50) NOT NULL,

FOREIGN KEY(BId) REFERENCES Books(BId) );


DESCRIBE Books;

DESCRIBE Issue;
```

Output: (Sample)



2. (Make a Python File add.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import tkMessageBox

import mysql.connector


def add_db():

    window=Tk()


    global id

    global title

    global author


    bid=id.get()

    btitle=title.get()

    bauthor=author.get()
```

```
db=mysql.connector.connect(host="localhost",user="root",passwo
rd="jai123",database="db",auth_plugin='mysql_native_password')

cursor=db.cursor()


print(bid,end='--')

print(btitle,end='--')

print(bauthor,end='--')

print("Add")


sqlquery="INSERT INTO Books VALUES('" + bid
+"','"+btitle+"','"+bauthor+"','YES');"

print(sqlquery)


try:

   cursor.execute(sqlquery)

   db.commit()

   tkMessageBox.showinfo("Success","Book Added Successfully")

except:

   tkMessageBox.showinfo("Error","Cannot add given book data
   into Database")
```

```
    window.destroy()


def addBooks():

    global id
    global title
    global author

    window=Tk()
    window.title("Library Management System")

    window.minsize(width=280,height=200)
    window.maxsize(width=280,height=200)

    window.iconbitmap("library44.ico")

    greet=Label(window,font=('arial', 30, 'bold'),text="Add Books")
    greet.grid(row = 0,columnspan = 3)
```

```
L=Label(window,font=('arial', 15, 'bold'),text="Enter Book ID:")
L.grid(row=2,column=1)

L=Label(window,font=('arial', 15, 'bold'),text="   ")
L.grid(row=2,column=2)

id=Entry(window,width=5,font=('arial', 15, 'bold'))
id.grid(row=2,column=3)

L=Label(window,font=('arial', 15, 'bold'),text="Enter Title:")
L.grid(row=4,column=1)

L=Label(window,font=('arial', 15, 'bold'),text="   ")
L.grid(row=4,column=2)

title=Entry(window,width=5,font=('arial', 15, 'bold'))
title.grid(row=4,column=3)
```

```
L=Label(window,font=('arial', 15, 'bold'),text="Enter Author:")

L.grid(row=6,column=1)


L=Label(window,font=('arial', 15, 'bold'),text="   ")

L.grid(row=6,column=2)


author=Entry(window,width=5,font=('arial', 15, 'bold'))

author.grid(row=6,column=3)


submitbtn=Button(window,text="Submit",command=add_db,bg="
DodgerBlue2",fg="white",font=('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)


print("Add")

pass
```

3. (Make a Python File delete.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import tkMessageBox

import mysql.connector


def delete_db():

    window=Tk()


    window.iconbitmap("library44.ico")


    global id


    bid=id.get()
```

```
db=mysql.connector.connect(host="localhost",user="root",passwo
rd='jai123',database='db',auth_plugin="mysql_native_password")

cursor=db.cursor()


print(bid,end='--')

print("Delete")


cursor.execute("SELECT Available FROM Books")

a=cursor.fetchall()

print(a)

for i in a:

   if i==(u'YES',):

      cursor.execute("SELECT BId FROM Books")

      a=cursor.fetchall()

      print(a)

      if int(bid)<10:

         if tuple(bid) in a:

            sqlquery="DELETE FROM Books WHERE BId="+(bid)

            print(sqlquery)
```

```
            cursor.execute(sqlquery)
            db.commit()
            tkMessageBox.showinfo("Success","Book Deleted
            Successfully")
        else:
            tkMessageBox.showinfo("Error","Book with given id does
            not exist")
    else:
        t1=(bid,)
        if t1 in a:
            sqlquery="DELETE FROM Books WHERE BId="+(bid)
            print(sqlquery)
            cursor.execute(sqlquery)
            db.commit()
            tkMessageBox.showinfo("Success","Book Deleted
            Successfully")
        else:
            tkMessageBox.showinfo("Error","Book with given id does
            not exist")
```

```
        else:
            tkMessageBox.showinfo("Error","Book is currently not
            available")

    window.destroy()

def deleteBooks():

    global id

    window=Tk()
    window.title("Library Management System")

    window.iconbitmap("library44.ico")

    window.minsize(width=325,height=135)
    window.maxsize(width=325,height=135)
```

```
greet=Label(window,font=('arial',30,'bold'),text="Delete Books")

greet.grid(row=0,columnspan=3)


L=Label(window,font=('arial',15,'bold'),text="Enter Book ID:")

L.grid(row=2,column=1)


L=Label(window,font=('arial',15,'bold'),text="    ")

L.grid(row=2,column=2)


id=Entry(window,width=5,font=('arial',15,'bold'))

id.grid(row=2,column=3)



submitbtn=Button(window,text="Submit",command=delete_db,bg
="DodgerBlue2",fg="white",font=('arial',15,'bold'))

submitbtn.grid(row=8,columnspan=3)


print("Delete")

pass
```

4. (Make a Python File issue.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import tkMessageBox

import mysql.connector


def issue_db():


    window=Tk()

    window.iconbitmap("library44.ico")


    global id

    global StudentName


    bid=id.get()

    bStudentName=StudentName.get()
```

```
db=mysql.connector.connect(host="localhost",user="root",passwo
rd="jai123",database="db",auth_plugin="mysql_native_password"
)

cursor=db.cursor()


print(bid,end='--')

print(bStudentName,end='--')

print("Issue")


try:

    checkavailability="SELECT * FROM Books WHERE
    Available='YES';"

    print(checkavailability)

    cursor.execute(checkavailability)


    flag=0


    for i in cursor:

        print(i[0])

        if(i[0]==bid):
```

```
        flag=1
        break;

if flag==1:
    updatequery="UPDATE Books SET available='NO' WHERE
    bid='"+bid +"';"
    print(updatequery)
    cursor.execute(updatequery)
    db.commit()

     sqlquery="INSERT INTO Issue VALUES('" + bid +"','"
    +bStudentName+"' );"
    print(sqlquery)

    cursor.execute(sqlquery)
    db.commit()

    tkMessageBox.showinfo("Success","Book issued Successfully")
else:
```

```
            tkMessageBox.showinfo("Error","Required Book is not
            available")
    except:
        messagebox.showinfo("Error","Cannot issue given book ")

def issueBooks():

    global id
    global StudentName

    window=Tk()
    window.title("Library Management System")

    window.minsize(width=302,height=165)
    window.maxsize(width=302,height=165)

    window.iconbitmap("library44.ico")

    greet=Label(window,font=('arial',30,'bold'),text="Issue Books")
```

```
greet.grid(row=0,columnspan=3)


L=Label(window,font=('arial',15,'bold'),text="Enter Book ID: ")

L.grid(row=2,column=1)


L=Label(window,font=('arial',15,'bold'),text="    ")

L.grid(row=2,column=2)


id=Entry(window,width=5,font=('arial',15,'bold'))

id.grid(row=2,column=3)


L=Label(window,font=('arial',15,'bold'),text="Enter Student Name:
")

L.grid(row=4,column=1)


L=Label(window,font=('arial',15,'bold'),text="    ")

L.grid(row=4,column=2)


StudentName=Entry(window,width=5,font=('arial',15,'bold'))
```

```
StudentName.grid(row=4,column=3)


submitbtn=Button(window,text="Submit",command=issue_db,bg=
"DodgerBlue2",fg="white",font=('arial',15,'bold'))

submitbtn.grid(row=8,columnspan=3)


print("Issue")

pass
```

5. (Make a Python File Return.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import tkMessageBox

import mysql.connector


def return_db():


    window=Tk()

    window.iconbitmap("library44.ico")


    global id


    bid=id.get()


    db=mysql.connector.connect(host="localhost",user="root",passwo
    rd="jai123",database="db",auth_plugin="mysql_native_password"
    )
```

```
cursor=db.cursor()


print(bid,end='--')
print("Return")


try:
    checkavailability="SELECT * FROM Books WHERE
    Available='NO';"
    print(checkavailability)
    cursor.execute(checkavailability)


    flag=0


    for i in cursor:
        print(i[0])
        if(i[0]==bid):
            flag=1
            break;
```

```
    if flag==1:
        updatequery="UPDATE Books SET Available='YES' WHERE
        BId='"+bid +"';"
        print(updatequery)
        cursor.execute(updatequery)
        db.commit()

        sqlquery="DELETE FROM Issue WHERE BId='" + bid +"';"
        print(sqlquery)

        cursor.execute(sqlquery)
        db.commit()

        tkMessageBox.showinfo("Success","Book Returned
        Successfully")
    else:
        tkMessageBox.showinfo("Error","Invalid Book ID")
except:
    tkMessagBox.showinfo("Error","Cannot return given book ")
```

```
def returnBooks():

    global id

    window=Tk()
    window.title("Library Management System")

    window.iconbitmap("library44.ico")

    window.minsize(width=330,height=135)
    window.maxsize(width=330,height=135)

    greet=Label(window,font=('arial', 30, 'bold'),text="Return Books")
    greet.grid(row = 0,columnspan = 3)

    L=Label(window,font=('arial', 15, 'bold'),text="Enter Book ID:")
    L.grid(row = 2, column = 1)

    L=Label(window,font=('arial', 15, 'bold'),text="   ")
```

```
L.grid(row=2,column=2)


id=Entry(window,width=5,font=('arial', 15, 'bold'))

id.grid(row=2,column=3)


submitbtn=Button(window,text="Submit",command=return_db,bg
="DodgerBlue2",fg="white",font=('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)


print("Return")

pass
```

6. (Make a Python File view.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import tkMessageBox

import mysql.connector


def viewBooks():


    global id


    window=Tk()

    window.title("Library Management System")


    window.minsize(width=440,height=100)

    window.maxsize(width=440,height=700)


    window.iconbitmap("library44.ico")
```

```
greet=Label(window,font=('arial',30,'bold'),text="View Books")

greet.grid(row=0,columnspan=4)



db=mysql.connector.connect(host="localhost",user="root",passwo
rd="jai123",database="db",auth_plugin="mysql_native_password"
)

cursor=db.cursor()


sqlquery="SELECT * FROM Books;"

print(sqlquery)


try:

    cursor.execute(sqlquery)

    L=Label(window,font=('arial',20),text="%-10s%-10s%-10s%-
    10s"%('BId','Title','Author','Available'))

    L.grid(row = 1,columnspan = 4)
```

```
    x=4
    for i in cursor:
        L=Label(window,font=('arial',15),text="%-10s%-15s%-15s%-
        10s"%(i[0],i[1],i[2],i[3]))
        L.grid(row=x,columnspan=4)
        x+=1

except:
    tkMessageBox.showinfo("Error","Cannot Fetch data")

print("View")
pass
```

7. (Make a Python File home.py and enter the following code in it.)

NOTE: THIS CODE WILL WORK ONLY FOR PYTHON 2.7.18 USERS. IF YOU ARE A PYTHON 3+ USER THEN YOU HAVE TO MODIFY IT YOURSELF.

```
from __future__ import print_function

from Tkinter import *

import mysql.connector


from add import *

from delete import *

from issue import *

from Return import *

from view import *

from PIL import Image,ImageTk


db=mysql.connector.connect(host="localhost",user="root",password="jai123",database="db",auth_plugin="mysql_native_password")

cursor=db.cursor()


window=Tk()
```

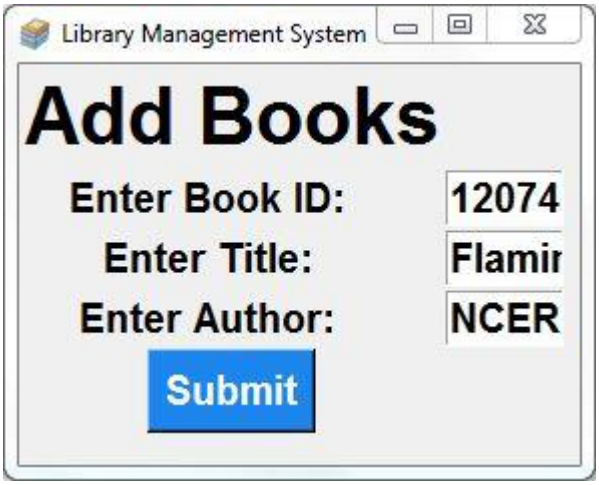
```
window.geometry("385x365")

window.minsize(width=385,height=365)

window.maxsize(width=385,height=365)

window.title("Library Management System")


img=ImageTk.PhotoImage(Image.open("capture1.jpg"))

lab=Label(image=img).place(x=0,y=0)


window.iconbitmap("library44.ico")


greet=Label(window,font=('arial',30,'bold'),text="Welcome to
Library!",bg="white",fg="black")

greet.grid(row=0,columnspan=3)


addbtn=Button(window,text="Add
Books",command=addBooks,bg="DodgerBlue2",fg="white",font=('ari
al',20,'bold'))

addbtn.grid(row=3,columnspan=3)

deletebtn=Button(window,text="Delete
Books",command=deleteBooks,bg="DodgerBlue2",fg="white",font=('
arial',20,'bold'))
```

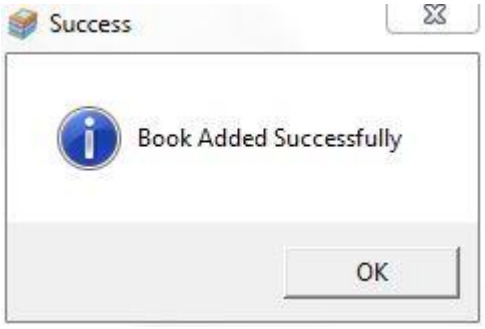
```
deletebtn.grid(row=5,columnspan=3)


issuebtn=Button(window,text="Issue
Books",command=issueBooks,bg="DodgerBlue2",fg="white",font=('a
rial',20,'bold'))

issuebtn.grid(row=7,columnspan=3)


returnbtn=Button(window,text="Return
Books",command=returnBooks,bg="DodgerBlue2",fg="white",font=('
arial',20,'bold'))

returnbtn.grid(row=9,columnspan=3)


viewbtn=Button(window,text="View
Books",command=viewBooks,bg="DodgerBlue2",fg="white",font=('a
rial',20,'bold'))

viewbtn.grid(row=11,columnspan=3)


greet=Label(window,font=('arial', 15, 'bold'),text="Thank You")

greet.grid(row=13,columnspan=3)


window.mainloop()
```

Output:

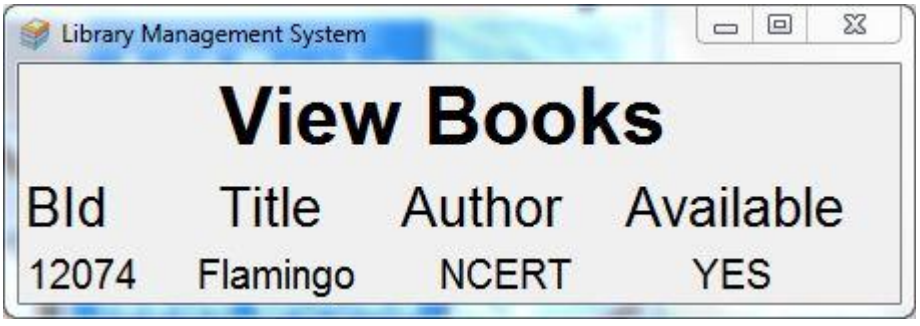

Library Management System
Welcome to Library!
Add Books
Delete Books
Issue Books
Return Books
View Books
Thank You

Add Books

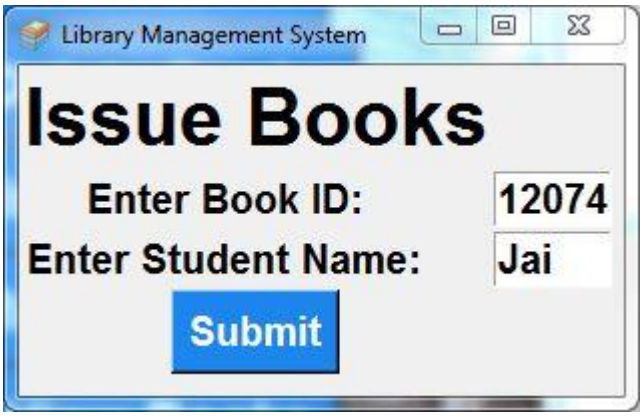


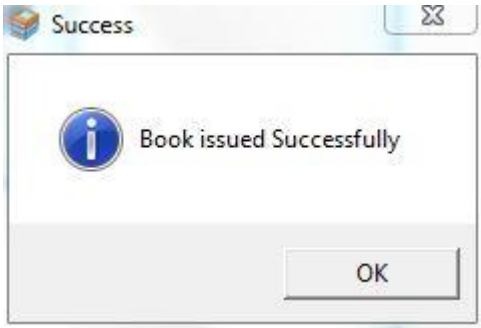


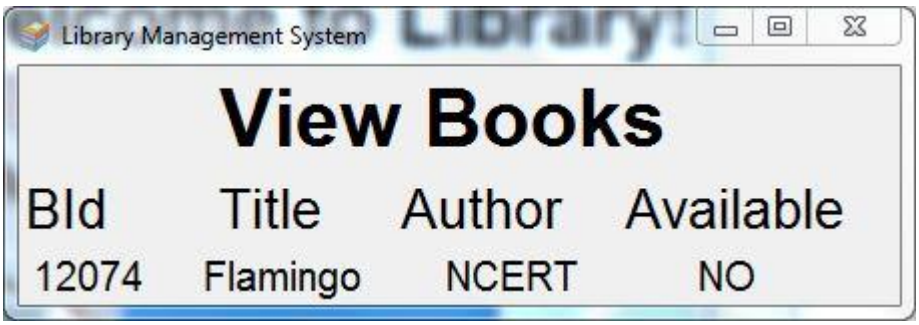
View Books



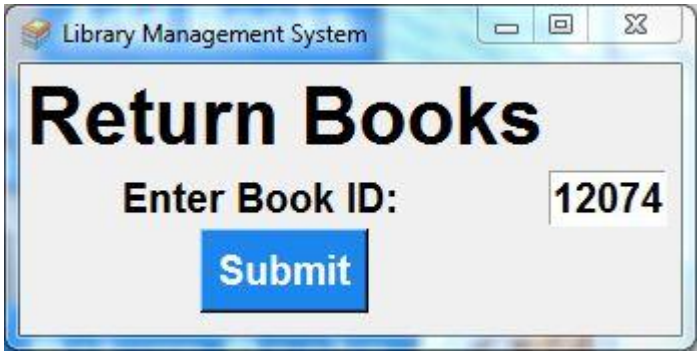
Issue Books

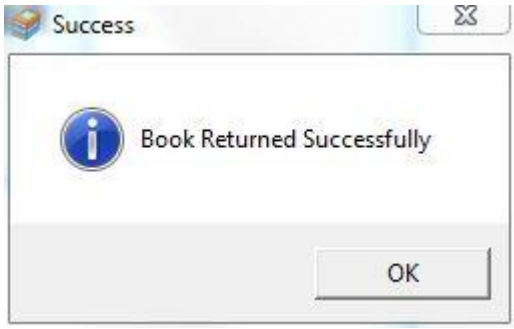

Library Management System
Issue Books
Enter Book ID:
12074
Enter Student Name:
Jai
Submit

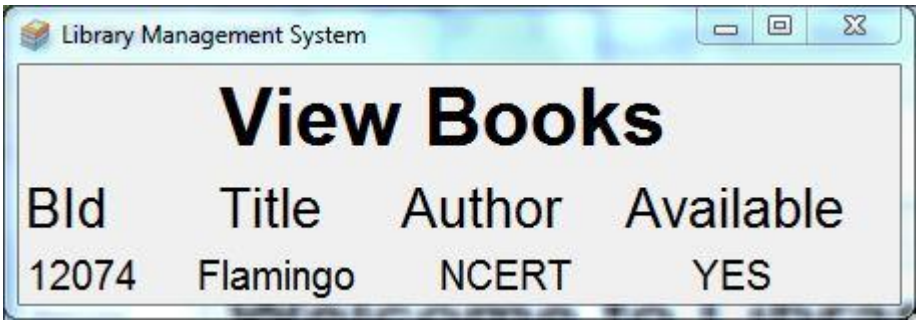

Success
Book issued Successfully
OK


Library Management System
View Books
BId
Title
Author
Available
12074
Flamingo
NCERT
NO

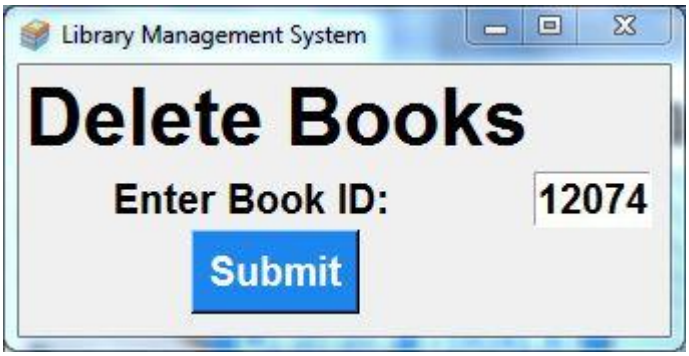
Return Books

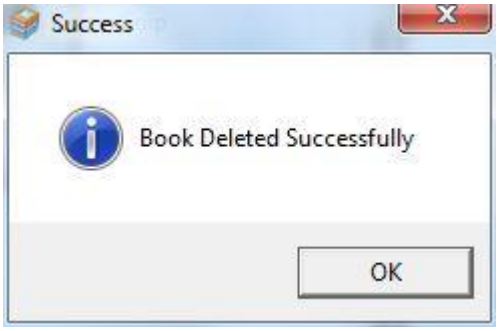


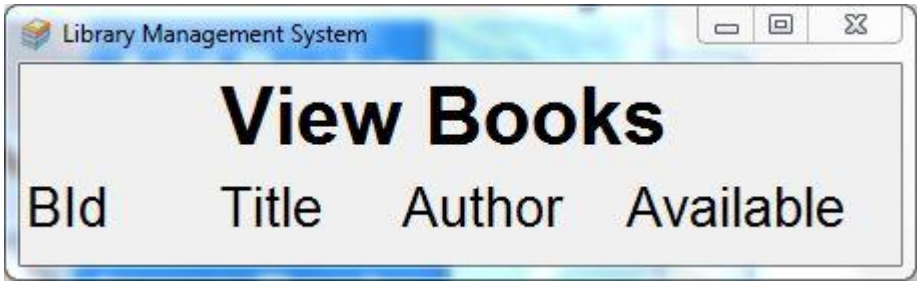




Delete Books







LIMITATIONS

1. Upto a certain limit books can be viewed using View Books option but after that limit has been crossed books can’t be viewed. (details of these books can be viewed using MySQL Queries)

2. Books with longer names can be added but it becomes difficult to view them using View Books option. (details of these books can be viewed using MySQL Queries)

ENHANCEMENTS

1. A vertical scroll bar can be added. It will enable the user to scroll the content up or down. Furthermore, when navigating a long piece of content, the scrollbar will help to rapidly move to a specific point within that content.

WEBLIOGRAPHY

1. https://docs.python.org/2.7/library/tk.html

2. Used the idea of various projects using Tkinter module available on internet

BIBLIOGRAPHY

1. Computer Science Textbook For Class 12
Published by NCERT

2. Computer Science with Python Textbook For Class 12 (Examination 2022-2023)
Publisher : Dhanpat Rai & Co. (Pvt.) Ltd.
Author : Sumita Arora