DELHI PUBLIC SCHOOL – BOPAL, AHMEDABAD

CERTIFICATE

This is to certify that ______________________ student of Class XI ____ has successfully completed project on the________________________topic under the guidance of _____________________-Subject Teacher, during the academic year 2021-2022, for fulfilment of Practical Examination 2021-2022.

Subject: Computer Science (083)

Subject Teacher **Examiner**

Date: ___________

ACKNOWLEDGEMENT

I tried to apply my best of knowledge and experience gained during study and class work. I would like to express my special thanks to my Computer Science Teacher Ms. Malvika Sharma for providing me with the opportunity to do this project on Library Management System and also provided support in completing my project.

I also feel indebted to my friend Pritish Saraf, Vedica Mahendian for valuable suggestions during the project work

I would also like to express my gratitude to our principal ma’am Ms. Sabina Sahwney for providing me with valuable time and all the facilities that were required

Dhruv Jha
XI-Y

INDEX

1.	Acknowledgement
2.	Introduction
3.	Hardware/Software Requirements
4.	Modules used in the Project
5.	Database Design
6.	Source Code
7.	Output Screen
8.	Conclusion

1. Introduction
2. Hardware/Software Requirements
    2.1. Hardware Requirements
    2.2. Software Requirements
    2.3. Built-in Python Libraries used
3. Modules used in the Project
    3.1. FILE STRUCTURE
4. Database Design
5. Source Code
    5.1. main.py
    5.2. AddBook.py
    5.3. DeleteBook.py
    5.4. IssueBook.py
    5.5. ReturnBook.py
    5.6. ViewBooks.py
6. I/O Screen
7. User Manual
8. Conclusion
9. Bibliography/ Reference

Introduction

This is a Python Project on Library Management System created using Tkinter module for GUI window and using MySQL for database connectivity, the purpose of this software is to develop the Management Information System (MIS)

A MIS mainly consists of a computerized database, inter-related tables capable of producing different reports relevant to the user. An application program is tied with the database for easy access and interface to the database for easy access and interface to database. Using an Application program in the front-end, we can store, retrieve and manage all information in the proper way.

The scope of my Library Management System project is to maintain and keep a track of books which are present in the library and the books issued to students.

To Provide a User Friendly, User Interface (GUI) based the application uses Tkinter which is an in-built python module for making GUI applications

To Run this Project Locally on your own device, read the User Manual first

Hardware/Software Requirements

Hardware Requirements

- Minimum
  - Processor: Intel Atom or Intel i3 or Ryzen 3
  - RAM: 2GB
  - OS: Windows 7, Linux, macOS
- Recommended
  - Processor: Intel i5+ or Ryzen 5+
  - RAM: 4GB+
  - OS: Windows 8/10/11, Linux, macOS

Software Requirements

- Python 3.5.x+
- MySQL 5.0+

Built-in Python Libraries used

- **Tkinter-** The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit.
- **Pillow(PIL)-** The Python Imaging Library adds image processing capabilities to your Python interpreter.
  This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.
- **PyMySQL-** The PyMySQL library is used as a connector for MySQL databases and Python

Modules used in the Project

bookRegister():- Function is used to add book into Database
addBook():- Function creates a GUI window for user to add book details

deleteBook():- Function is used to delete book from Database
delete():- Function creates a GUI window for user to delete book from list

issue():- Function changes the status of book to issued in the Database
issueBook():- Function creates a GUI window to let user issue a book

returnn():- Function changes the status of book to available in the Database
returnBook():- Function creates a GUI window to let user return a book

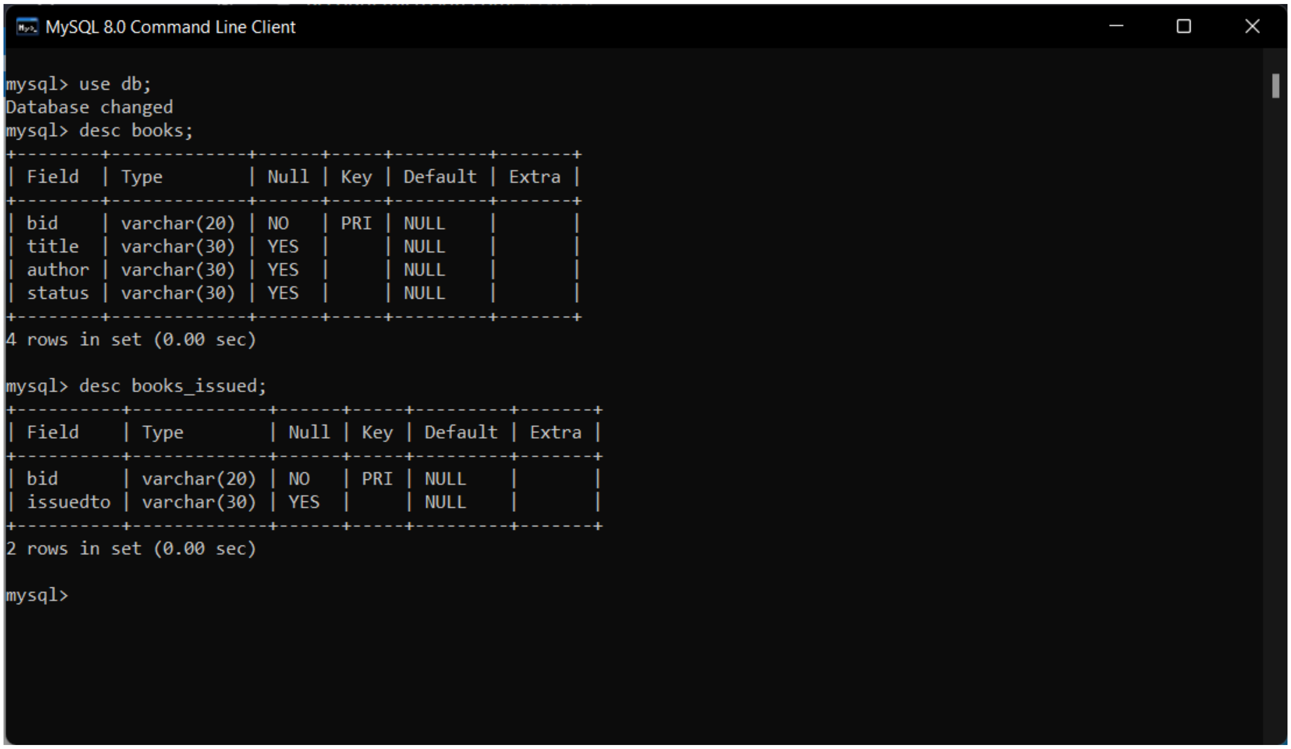
View():- Function creates a GUI window to view the list of books with status

FILE STRUCTURE

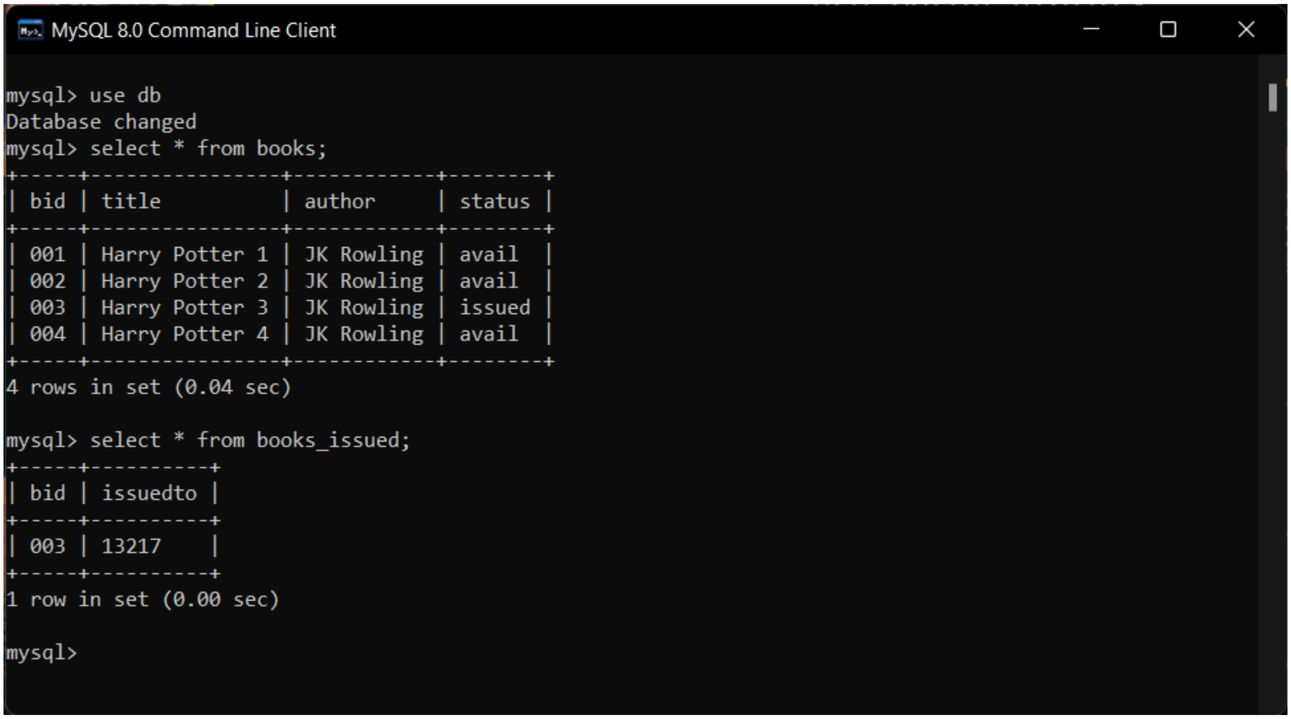
```
.
└── Lib Mgmt Sys/
    ├── AddBook.py
    ├── DeleteBook.py
    ├── IssueBook.py
    ├── lib.jpg
    ├── main.py
    ├── requirements.txt
    ├── ReturnBook.py
    └── ViewBooks.py
```

Database Design

Database name: db
Table name: books & books_issued



```
mysql> use db;
Database changed
mysql> desc books;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bid    | varchar(20) | NO   | PRI | NULL    |       |
| title  | varchar(30) | YES  |     | NULL    |       |
| author | varchar(30) | YES  |     | NULL    |       |
| status | varchar(30) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 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>
```



```
mysql> use db
Database changed
mysql> select * from books;
+-----+----------------+------------+--------+
| bid | title          | author     | status |
+-----+----------------+------------+--------+
| 001 | Harry Potter 1 | JK Rowling | avail  |
| 002 | Harry Potter 2 | JK Rowling | avail  |
| 003 | Harry Potter 3 | JK Rowling | issued |
| 004 | Harry Potter 4 | JK Rowling | avail  |
+-----+----------------+------------+--------+
4 rows in set (0.04 sec)

mysql> select * from books_issued;
+-----+----------+
| bid | issuedto |
+-----+----------+
| 003 | 13217    |
+-----+----------+
1 row in set (0.00 sec)

mysql>
```

Source Code

5.1. main.py

```python
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 *


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

same=True
n=2
# Adding a background image
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(600,340,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 DPS BOPAL 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()
```

5.2. AddBook.py

```
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 = "12345"
    mydatabase="db"

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

5.3. DeleteBook.py

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

mypass = "12345"
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)

    # Book ID to Delete
    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)

    #Submit Button
    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()
```

5.4. IssueBook.py

```
mypass = "12345"
mydatabase="db"
con = pymysql.connect(host="localhost",user="root",
password=mypass,database=mydatabase)
cur = con.cursor()
issueTable = "books_issued"
bookTable = "books"
allBid = []  #To store all the Book ID's
```

5.4. IssueBook.py

```python
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
```

5.4. IssueBook.py

```
    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)
```

5.4. IssueBook.py

```
    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()
```

5.5. ReturnBook.py

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

mypass = "12345"
mydatabase="db"
con = pymysql.connect(host="localhost",user="root",
password=mypass,database=mydatabase)
cur = con.cursor()
```

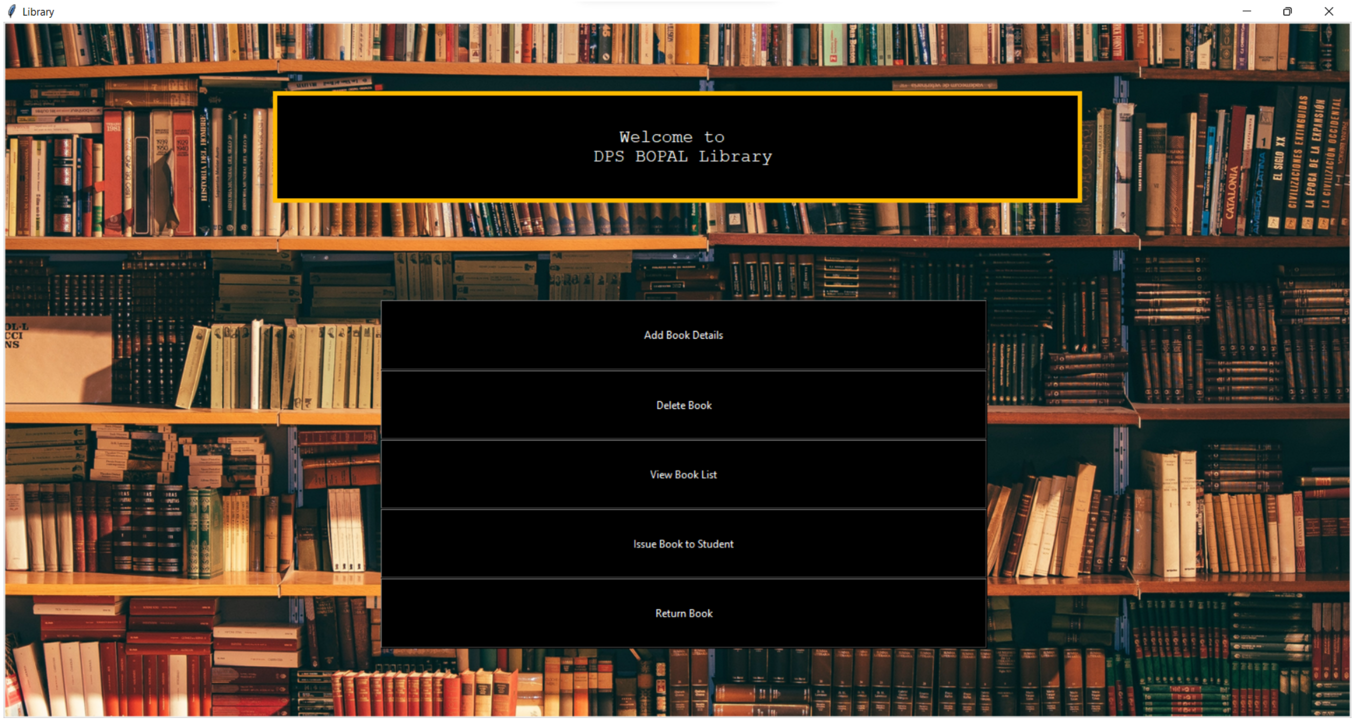
```
issueTable = "books_issued"
bookTable = "books"
allBid = []  #To store all the Book ID's

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)

    # 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()
```

5.6. ViewBooks.py

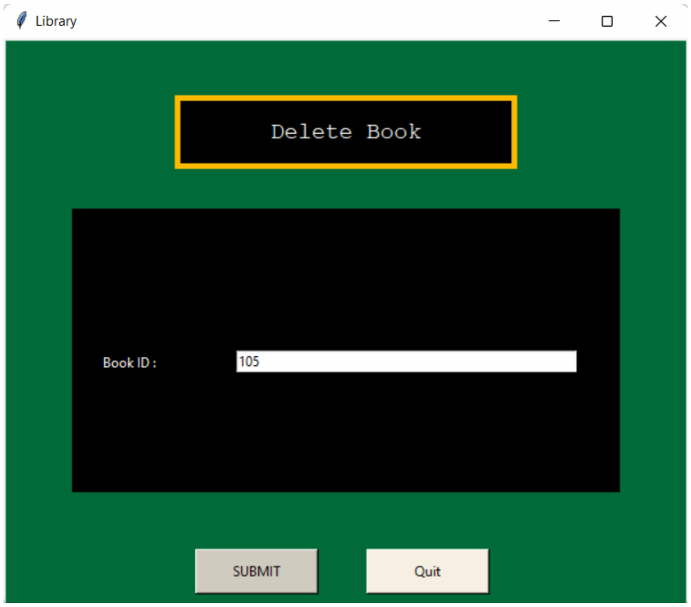
```
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox
import pymysql

mypass = "12345"
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("700x600")

    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%-60s%-40s%-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%-40s%-40s%-10s"%(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()
```

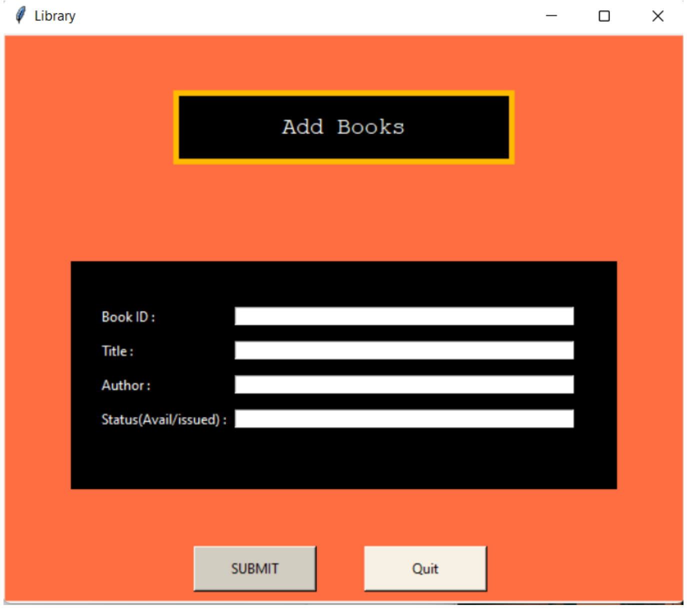
Output Screen



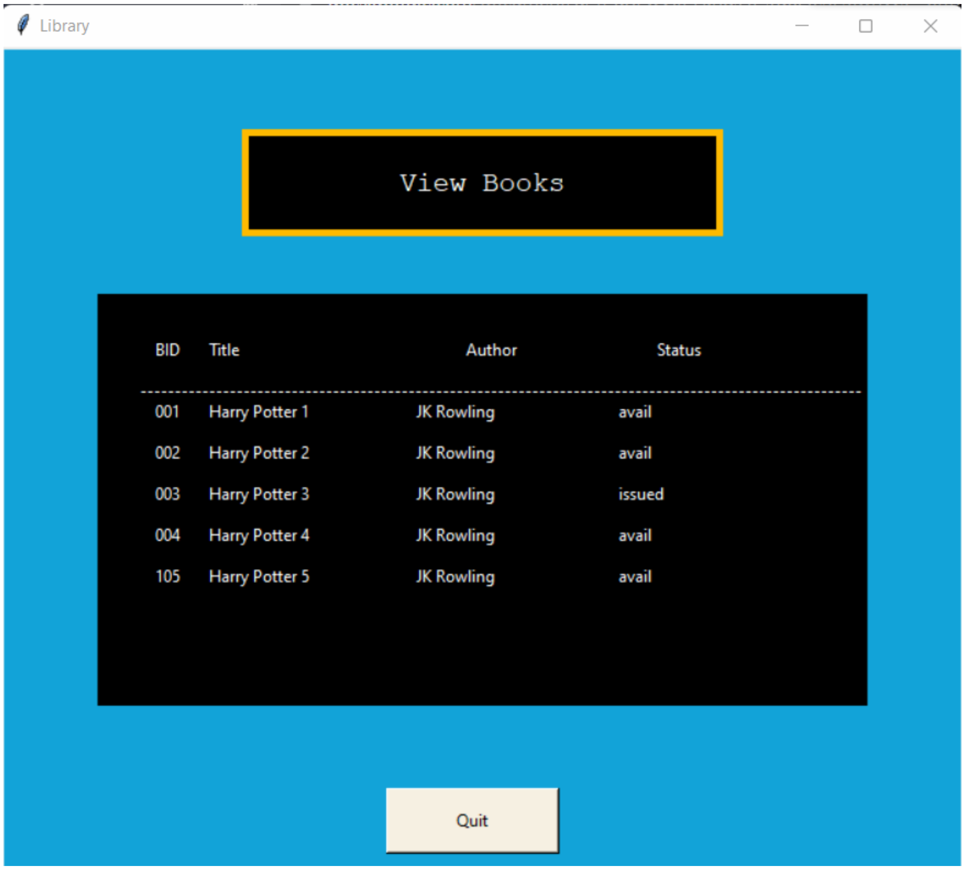
Welcome to
DPS BOPAL Library



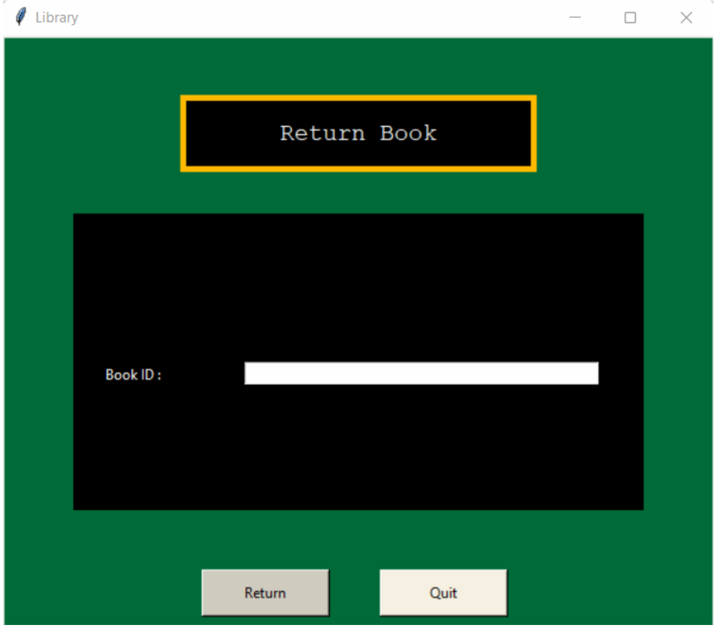
Book ID : 105

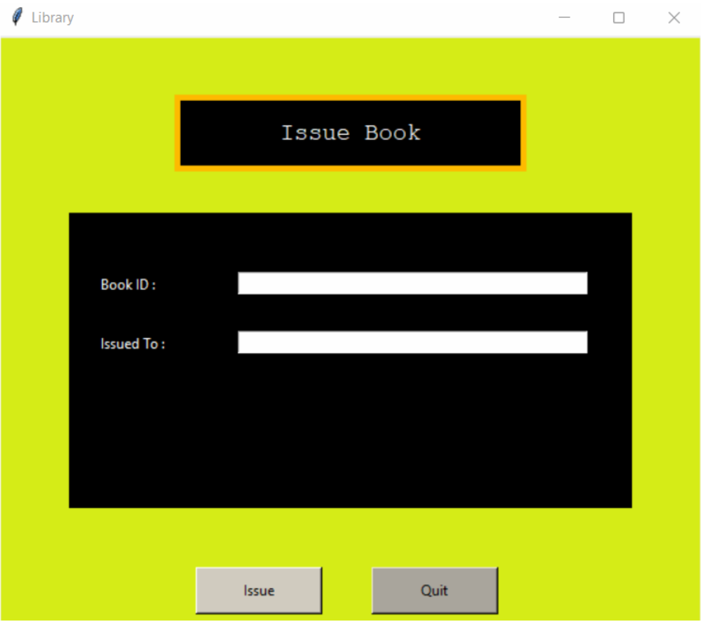


Book ID :
Title :
Author :
Status(Avail/issued) :



BID	Title	Author	Status
001	Harry Potter 1	JK Rowling	avail
002	Harry Potter 2	JK Rowling	avail
003	Harry Potter 3	JK Rowling	issued
004	Harry Potter 4	JK Rowling	avail
105	Harry Potter 5	JK Rowling	avail





User Manual

1. Download the *Lib Mgmt Sys.zip* and extract all the files.
2. Open a terminal in the directory where you have extracted the files
3. Setup
   3.1. Run the command `pip install -r requirements.txt`
   3.2. Open MySQL and run the following commands

```sql
create database db;
use db;
create table books(bid varchar(20) primary key, title varchar(30), author varchar(30), status varchar(30));
create table books_issued(bid varchar(20) primary key, issuedto varchar(30));
```

4. Open *main.py* and run the file

Conclusion

This is a python project which can be used to fulfill basic library needs, it is an efficient way to manage a library. It saves time and is hassle-free from the traditional way of maintaining records in a register. The application has functionalities which include adding, removing, issuing a book from/to a library.

Bibliography/ Reference

Library Image: Unsplash- Alfonso Morales
Python.org
Tkinter Docs