

# FINAL REPORT



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

**PROJECT :BOOK MANAGEMENT SYSTEM**

**SECTION:BG**

UNDER THE GUIDANCE OF

**SHIVANI BHARDWAJ**

**Submitted By:**

<b>SAI SUDHAKAR</b>	<b>----- K21BGB45</b>
<b>UTKARSH</b>	<b>----- K21BGB70</b>
<b>SINGH</b>	
<b>ESWAR SAI</b>	<b>-----K21BGA20</b>

# **INTRODUCTION**

## **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 reading.
- A search column to search availability of books.
- Facility to download required book.
- Video tutorial for students.
- An Admin login page where admin can add books, videos or page sources.open link for Learning Websites

## **OBJECTIVE:**

E-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, videos and Page sources.

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.

# DESIGN

## PYTHON 3.10-

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

## TKINTER :

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the TK GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps. Import the Tkinter module. Create the GUI application main window Add one or more of the above-mentioned widgets to the GUI

## Common attributes used

- a. Fg is a common attribute to color the text as in the project the foreground of the text is white.
- b. Bg is a common attribute called background, already clear from the name it is used for giving color to the background of the text. In the project the bg or background of the text in the button is black. and the color of the window is grey.
- c. Command is used at various times. It is used in the buttons. The functions are called using this command attribute.
- d. Font it is used to beautify as well as magnify the text. In the project, the font used is

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS VISTA ,WINDOWS7, UBUNTU
MEMORY	1GB RAM OR MORE
HARD DISK SPACE	MINIMUM 3 GB FOR DATABASE USAGE FOR FUTURE
DATABASE	MY SQL

### SYSTEM ANALYSIS:

In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification (SRS) and comparison between existing and proposed system . The functional and non functional requirements are included

in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

# SOFTWARE REQUIREMENT SPECIFICATION

## 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 time- saving

### 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. Some times 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

## **FUNCTIONAL REQUIREMENTS:**

### **1. NORMAL USER**

#### **USER LOGIN**

##### Description of feature

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system .The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

##### Functional requirements

- user id is provided when they register
- The system must only allow user with valid id and password to enter the system
- The system performs authorization process which decides what user level can access to.
- The user must be able to logout after they finished using system.

#### **1.1 REGISTER NEW USER**

##### Description of feature

This feature can be performed by all users to register new user to create account.

##### Functional requirements

- System must be able to verify information
- System must be able to delete information if information is wrong

## **REGISTER NEW BOOK:**

### **Description of feature**

This feature allows to add new books to the library

### **Functional requirements**

- System must be able to verify information
  - System must be able to enter number of copies into table.
  - System must be able to not allow two books having same book id.
- 

## **REQUIREMENTS:**

### **Functional Requirements:**

**Book Entry:** In this module we can store the details of the books.

**Register Student:** In this module we can keep the details of the new student.

**Book Issue:** This module is used to keep the track of book issue details.

**Book Return:** This module enables to keep a track of returned books

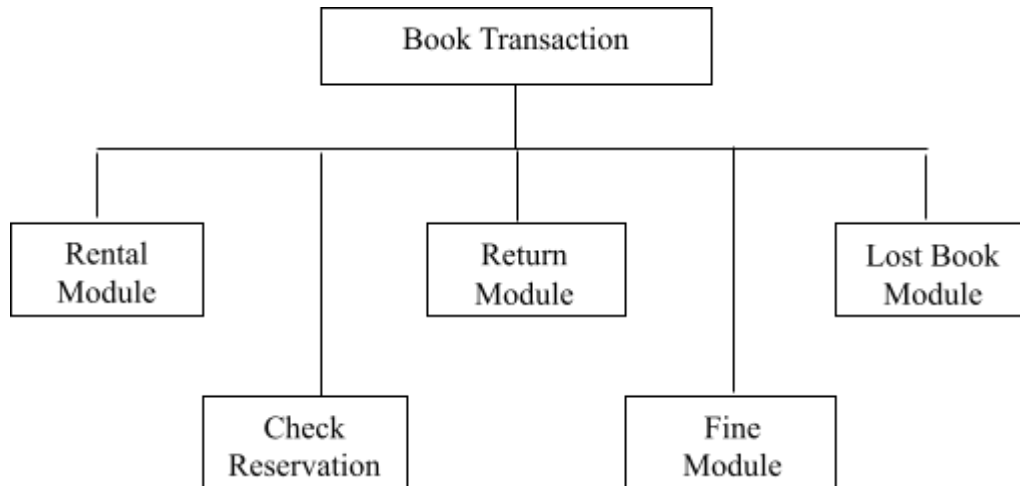
**LABEL:-** We used label for text like frame and headings, etc  
and also we use for entry

**BUTTON:-** We used button for signup, signin, register, login and more etc

**ENTRY:-**We used entry for widgets for reading the books, author, newpages and more etc.....



# Book Transaction Module

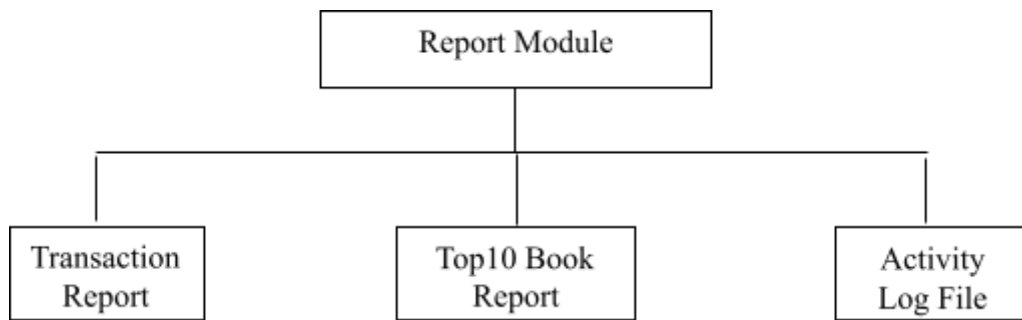


Book Transaction module is a main module in Library Management System. When member wants to borrow books, return books or they want to register lost book, it is all under book Transaction module. This module can be accessed by normal user or admin user. When member wants to borrow a book, librarian needs to scan in their member id.

After that, librarian will scan their book's barcode id. If the book is under reservation, the book is not available to rent.

For return module, librarian just needs to scan the book's barcode id, and confirm the rental detail with user.

# Report Module

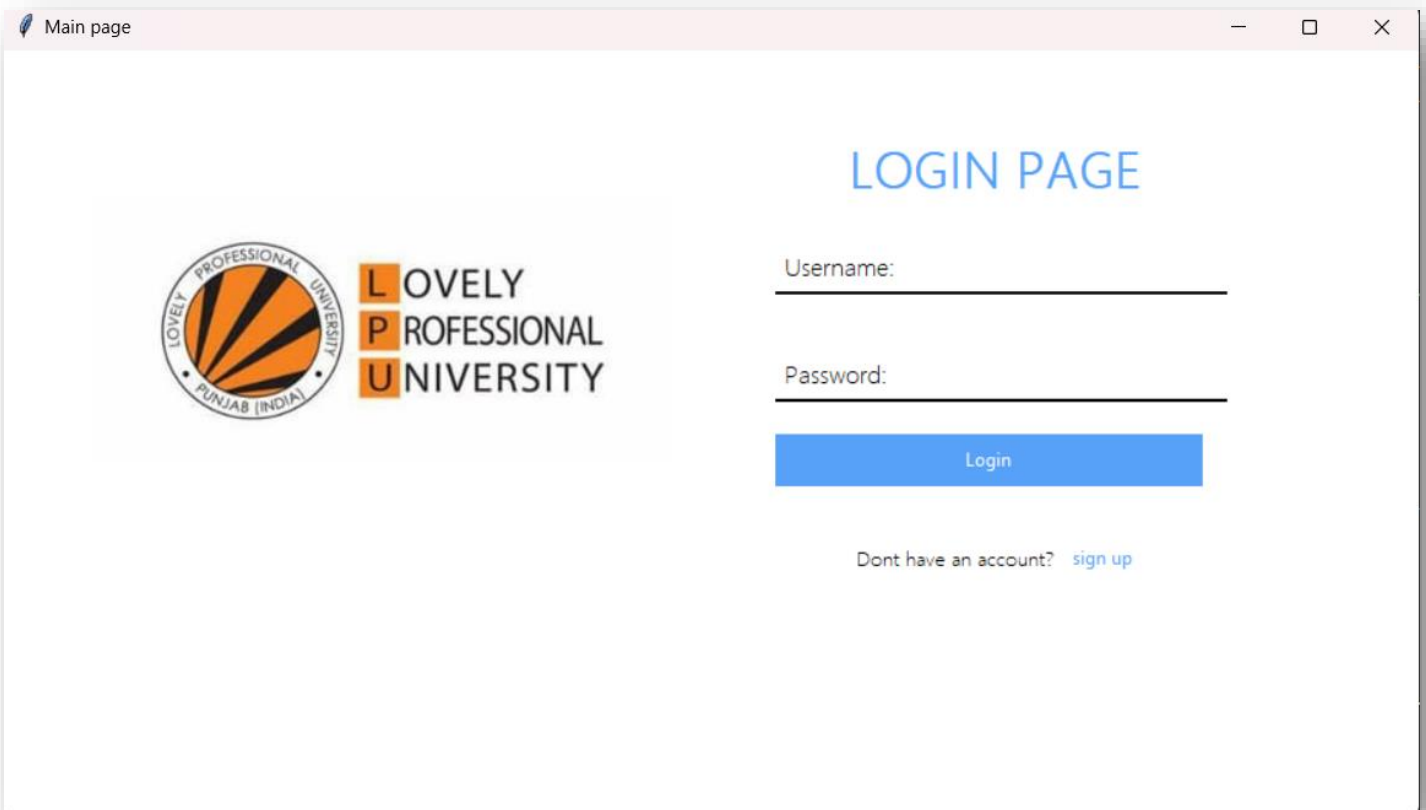


Report module is the main module for admin user. It is because normal user is not allowed to view the report. The report divided into 3 types. First one is transaction report which can let admin views the book transaction happen on particular date such as rental report and return report.

Top10 Report is the top rental rate's book. Admin can filter the information based on book's category and also filter by date in type of daily, monthly and yearly.

Activity Log File is a log which records every process in the Library Management System such as login / logout activity, register new book, new member or edit information or a member. All the activity done by every user will be record so that when system crash, admin or system admin are able to check the activity that may crash the system.

# Screen shots of gui



## Snapshots of the coding :

```
File Edit Selection View Go Run Terminal Help
login.py - Visual Studio Code - Insiders
available_books.py database.py avail_book.py login.py X register.py returnbook1.py ADD
D:\> python project > login.py > sign in
1 from tkinter import * #####this for window page##
2 from tkinter import messagebox
3 from register1 import*
4 from returnbook1 import*
5 import ast
6
7 window = Tk()
8 window.title("Main page")
9 window.geometry("925x500+300+200")
10 window.configure(bg='ffff')
11
12
13 #-----sign in -----
14 def signin():
15     Username = user.get()
16     Password = code.get()
17
18     if (Username == "abc" and Password == "1234"):
19
20
21
22     frame =Frame(window,width=350,height=390,bg='ffff')
23     frame.place(x=480,y=50)
24
25     heading =Label(frame,text='BMS',bg="white",fg="#57a1f8",font=("Microsoft Yahei UI Light",23,'bold'))
26     heading.place(x=130,y=50)
27
28
29
30     #####BUTTONS#####
31     Button(frame,width=39,pady=7,text="RETURN BOOK",bg="#57a1f8",fg="white",border=0,command=ret).place(x=25,y=100)
32     Button(frame,width=39,pady=7,text="AVAILABLE BOOKS",bg="#57a1f8",fg="white",border=0).place(x=25,y=150)
33     Button(frame,width=39,pady=7,text="ISSUE BOOK",bg="#57a1f8",fg="white",border=0).place(x=25,y=200)
34     Button(frame,width=39,pady=7,text="ADD BOOKS",bg="#57a1f8",fg="white",border=0).place(x=25,y=250)
35     #####
36
37 #-----register-----
38
39
40
41
42
43
44
45
46
47
48
49
50
```

```
File Edit Selection View Go Run Terminal Help login.py - Visual Studio Code - Insiders
available books.py database.py avail_book.py login.py X register1.py returnbook1.py ADD Debug login.py 2 V

D:\> python project > login.py > signin
51
52 elif Password != '1234':
53     messagebox.showerror('Invalid', 'Invalid password')
54
55 elif Username != 'admin':
56     messagebox.showerror('Invalid', 'Invalid username')
57
58
59 img = PhotoImage(file='LOGO1234.png')
60 Label(window, image=img, border=0, bg='white').place(x=50, y=50)
61
62 frame = Frame(window, width=350, height=390, bg='ffff')
63 frame.place(x=480, y=50)
64
65 heading = Label(frame, text='LOGIN PAGE', bg='white', fg='#57a1f8', font=('Microsoft Yahei UI Light', 23, 'bold'))
66 heading.place(x=70, y=5)
67
68 #####-----USERNAME-----#####
69
70
71 def on_enter(e):
72     user.delete(0, 'end')
73
74
75 def on_leave(e):
76     name = user.get()
77     if name == '':
78         user.insert(0, 'Username')
79
80
81 user = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft Yahei UI Light', 11))
82 user.place(x=30, y=80)
83 user.insert(0, 'Username:')
84 user.bind("<FocusIn>", on_enter)
85 user.bind("<FocusOut>", on_leave)
86 Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)
87
88
89 #####-----PASSWORD-----#####
90
91
92 def on_enter(e):
93     code.delete(0, 'end')
94
95
96 def on_leave(e):
97     name = code.get()
98     if name == '':
99         code.insert(0, 'Password')
```

```
File Edit Selection View Go Run Terminal Help login.py - Visual Studio Code - Insiders
available books.py database.py avail_book.py login.py X reg Debug login.py 2 V :py

D:\> python project > login.py > signin
101
102 code = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft Yahei UI Light', 11))
103 code.place(x=30, y=150)
104 code.insert(0, 'Password:')
105 code.bind("<FocusIn>", on_enter)
106 code.bind("<FocusOut>", on_leave)
107 Frame(frame, width=295, height=2, bg='black').place(x=25, y=177)
108
109
110
111 #####-----BUTTON FOR window-----#####
112 Button(frame, width=39, pady=7, text='Login', bg='#57a1f8', fg='white', border=0, command=signin).place(x=25, y=200)
113 label = Label(frame, text="Don't have an account?", fg="black", bg="white", font=("Microsoft Yahei UI Light", 9))
114 label.place(x=75, y=270)
115 #####-----BUTTON FOR SIGN UP -----#####
116 sign_up = Button(frame, width=6, text='sign up', border=0, bg="white", cursor='hand2', fg='#57a1f8', command=sign)
117 sign_up.place(x=215, y=270)
118
119 window.mainloop()
120
```

## LOGIN PAGE



LOVELY  
PROFESSIONAL  
UNIVERSITY

Dont have an account? [sign up](#)

invalid



invalid username and password

## BMS



LOVELY  
PROFESSIONAL  
UNIVERSITY

Book management system -Issue Book

## Issue Book

**Enter Book Details**

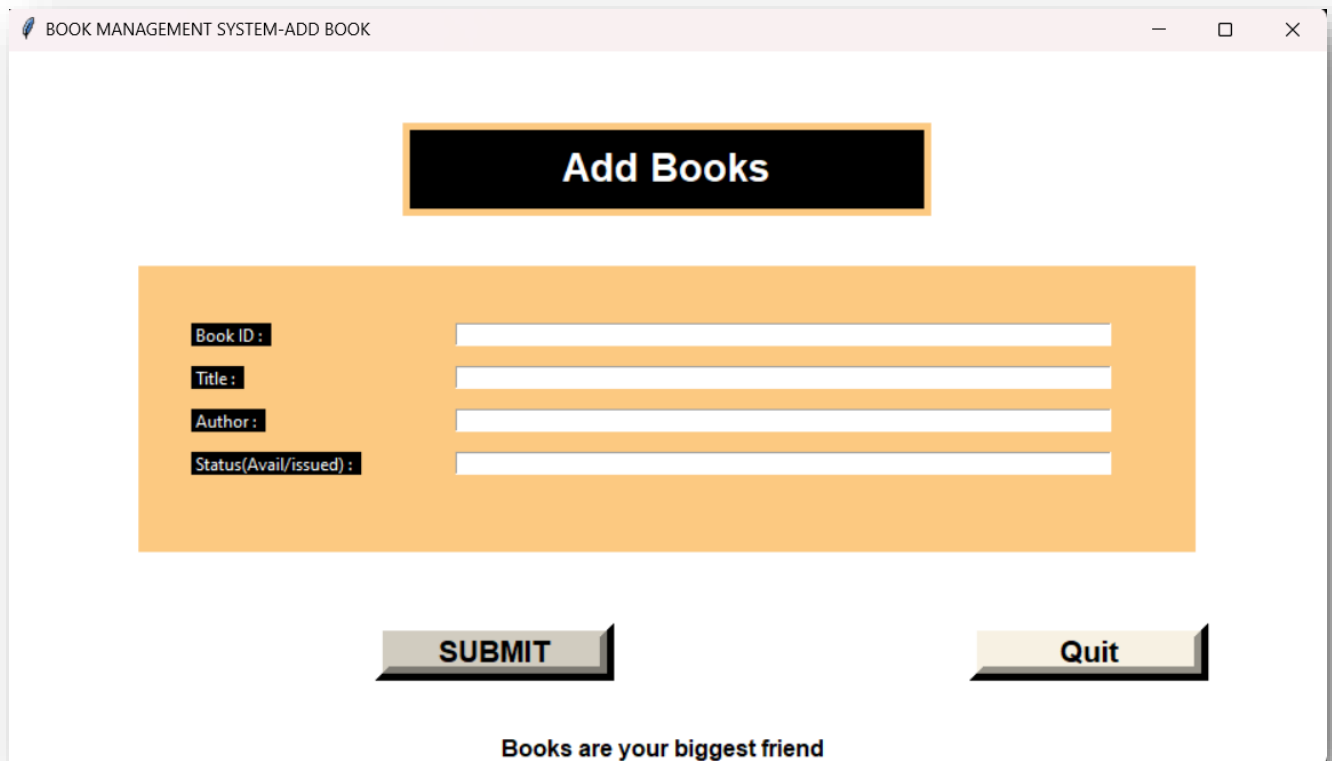
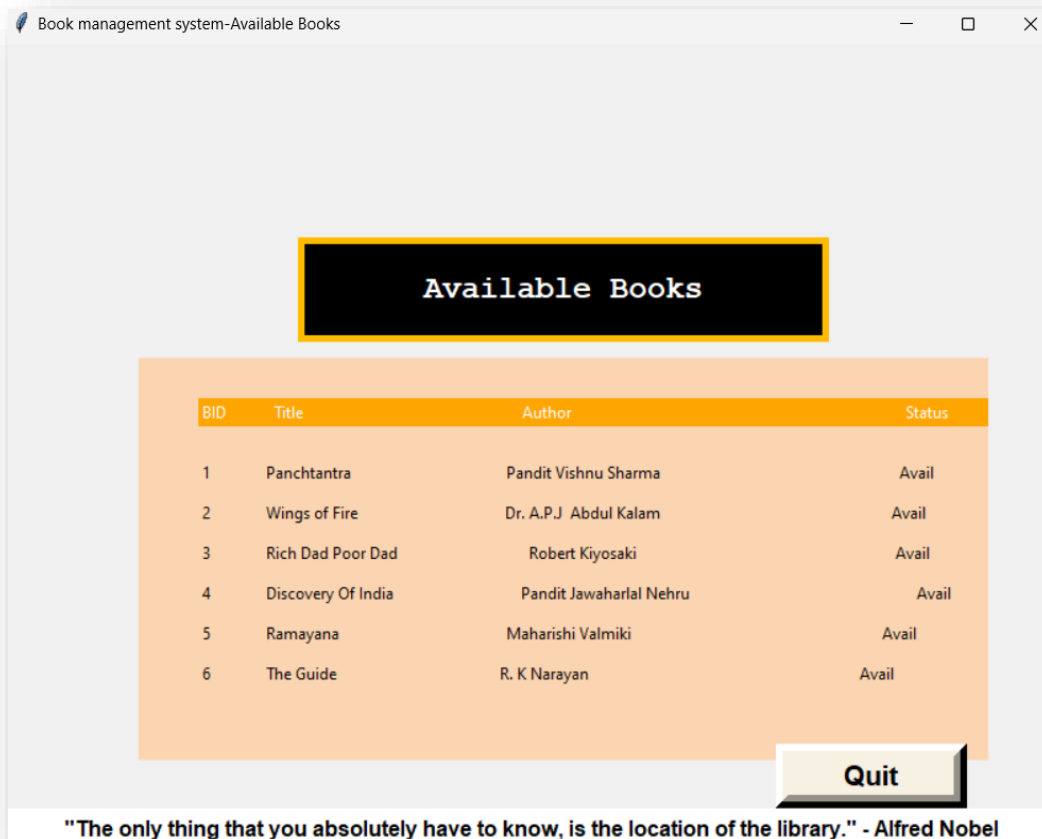
Book ID	<input type="text"/>
Date of Issue	<input type="text"/>

Book management system -Return Book

## Return book !

**Enter Book Details**

Book ID	<input type="text"/>
Fine (if any)	<input type="text"/>



## CONCLUSION:

- It will take care of the current book detail at any point of time.
- The book issue, book return will update the current book details automatically show that user will get the update current book details.
- The main purpose of this project is to reduce the manual work.
- This software is capable of managing book issues, returns, and calculating/managing fines.
- Generating various reports for record keeping according to end user requirements.

### **Work Done BY Sai Sudhakar (Roll No: B45)**

- Written code for the main page (login page ,register ,bms page)
- Linking pages with gui interface
- Written the codes how to input
- Created the buttons to linking

### **Work Done BY Utkarsh (Roll No:B70 )**

- Helping in the code of the labelling of the gui
- Given some descriptions in the coding part
- Report file created



## **Work Done BY Eswar (Roll No: A20)**

- Helped to created the available book , return book, issue book
- Designed the page with data saving
- Imported images and linked the page

## **Conclusion**

This software is successfully done to maintain a book shop's transaction.

The project is completed that follow incremental model

## **Reference**

- YOUTUBE
- GOOGLE

<https://www.google.com/search?q=book+management+system+project>

<https://itsourcecode.com/free-projects/python-projects/book-store-management>

## **GitHub Link for project:-**

[12105038/BOOK-MANAGEMENT-SYSTEM: BOOK MANAGEMENT SYSTEM IN GUI INTERFACE USING PYTHON \(github.com\)](https://github.com/12105038/BOOK-MANAGEMENT-SYSTEM)

# **Thank You!!!!**