**PUSTAKBOOK**

**(LIBRARY MANAGEMENT SYSTEM)**

**A SYNOPSIS SUBMITTED**

**FOR**

**LABORATORY OF**

**COMPUTER SCIENCE (083)**

**IN**

**CLASS XII-SCIENCE**

**BY**

**SNEHASISH MANDAL (ROLL NO-30)**

**DOMINIC MURMU (ROLL NO-14)**

**UNDER THE SUPERVISION OF**

**MR. GOVIND PRASAD ARYA**

**PGT- COMPUTER SCIENCE**

**KENDRIYA VIDYALAYA COOCHBEHAR**

**WEST BENGAL**

****

**TO THE**

**KOLKATA REGION, KENDRIYA VIDYALAYA SANGATHAN**

**MAY, 2022**

**CERTIFICATE**

This is to certified that Snehasish Mandal (Roll No-12) and Domnic Murmu   
(Roll No-00),have carried out the project work presented in this report entitled “**PUSTAKBOOK (LIBRARY MANAGEMENT SYSTEM)”** for laboratory of computer science (083) in class XII-Science, from Kendriya Vidyalaya Coochbehar, West Bengal under my supervision. The report embodies results of original work, and studies/implementation is carried out by the students themselves.

Date: **PGT- Computer Science**

**Mr. Govind Prasad Arya Kendriya VidyalayaCoochbehar**

**West Bengal**

**REPORT APPROVAL SHEET**

This is certified that the Project entitled “**PUSTAKBOOK**

**(LIBRARY MANAGEMENT SYSTEM)”** is approved for laboratory of computer science (083) in class XII-Science, from Kendriya Vidyalaya Coochbehar, West Bengal.

Name and Signature of project group members-

1. Snehasish Mandal (Roll No-30) \_\_\_\_\_\_\_\_\_\_\_\_\_
2. Dominic Murmu (Roll No-14) \_\_\_\_\_\_\_\_\_\_\_\_

(Mr. Govind Prasad Arya)

**Name & Signature of Project Supervisor**

**Name & Signature of Internal Examiner**

**Name & Signature of External Examiner**

**ABSTRACT**

PustAKbook is a Library Management System Software. Library is place where all kind of books are available. Library Management System is considered to be the current trend and boom to every Library administration and management. An LMS helps to control different complex workflows and processes which are inherent to the normal working of any Library. The LMS acts as the frontiers for various Library stakeholders to access, capture, report, and manage every single information happening in the functioning of your Library. This system contains list of all the books and can be accessed by remote users concurrently from any where in the campus.

Client sends requests, on receiving the request the server processes it and extracts the data from database and sends the result back to the client. We got so many errors while making this software. Librarian can modify database. The phase of understanding the concept and idea has taken a lot of time to complete. Users can search for books and renewal books online. They can view the issue and return dates of any book and due they have to pay. This system generates reports that can be used in analyzing the Library performance. Thus the management can take appropriate steps to improve the facilities At last, we conclude that the project was a different experience for us but we have done it successfully overcoming all the defaults.

**ACKNOWLEDGEMENTS**

It is a great pleasure to express our profound sense of gratitude and reverence to our project supervisor **Mr. Govind Prasad Arya, PGT- Computer Science, Kendriya Vidyalaya Coochbehar, West Bengal.** He was always a source of encouragement and inspiration, and constantly guided us for the accomplishment of this task with meticulous care. We owe to him the most, to have had the opportunity to accomplish the work under his guidance.We are extremely grateful and highly obliged to him. We have a special bond of kinship with him. His sincere, affable attitude and helping nature will always remain a pleasant part of our memory.

On a moral personal note, our deepest appreciation and gratitude to our beloved parents, who have been an inspiration and have provided us with unrelenting encouragement and support.

**Snehasish Mandal**

**Dominic Murmu**

**TABLE OF CONTENTS**

**Page No.**

COVER PAGE i

CERTIFICATE ii

REPORT APPROVAL SHEET iii

ABSTRACT iv

ACKNOWLEDGEMENTS v

**CHAPTER 1: INTRODUCTION & REQUIREMENT ANALYSIS 1-2**

1.1 WHAT IS A LIBRARY MANAGEMENT SYSTEM? 1-2

1.2 SOFTWARE REQUIREMENT SPECIFICATION (SRS) 2

**CHAPTER 2: DESIGN DOCUMENT 3-5**

2.1 DATABASE DESIGNING 3-4

2.2 GUI DESIGNIG 4-5

**CHAPTER 3: PROJECT IMPLEMENTATION** **6-43**

3.1 IMPLEMENTATION OF ALL REQUIRED MODULES CODE: 6

3.2 IMPLEMENTATION OF LOGIN WINDOW DETAILS 9

3.3 MAKING MENU OF ADD , DELETE,VIEW BOOK LIST, 13

SEARCH, ISSUE BOOKS TO STUDENT,RETURN BOOK

**CHAPTER 4: PROJECT TESTING 44-49**

4.1 TEST CASES IN CUSTOMER DETAIL 44

4.2 TEST CASES IN ADDING BOOK DETAILS 44

4.3 TEST CASES IN DELETE BOOK DETAILS 45

4.4 TEST CASES IN SEARCH BOOK DETAILS 47

4.5 TEST CASES IN ISSUING BOOKS TO STUDENTS 48

4.6 TEST CASES IN RETURNING BOOKS TO STUDENTS 49

**CHAPTER 5: CONCLUSION AND FUTURE ENHANCEMENT 50-52**

**REFERENCES** 52

**CHAPTER 1**

**INTRODUCTION & REQUIREMENT ANALYSIS**

The library management system is the important part of a making a digitised and automated library . It automates numerous daily operations and enables smooth interactions of the users. Developing the library management system software is a great opportunity to create the distinct, efficient and fast delivering library books and books storing data model. Implementation of library management system project helps to store all the kinds of records, provide coordination and user communication, proper calculations , proper record management , improve day-to-day operations, arrange the issue and return of book in management services. This beneficial decision covers the needs of the students , staff , teachers , common people and simplifies their interactions. It has become the usual approach to manage all the data and books present in a Vidyalaya, school , universities and etc. The library management system software is a great solution for any Vidyalaya, school , universities and most important our present public libraries in our area . Our present modern information system makes use of computers for the execution, each of them connected through an optimized network. 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 . To reduce this type of burdens Library management system came into existence

* 1. **WHAT IS A LIBRARY MANAGEMENT SYSTEM?**

Online Library Management System is a system which maintains the information about the books present in the library, their authors, the members of library to whom books are issued, library staff and all. This is very difficult to organize manually. Maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of an Online Library becomes much simple. The Online Library Management has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced.

* 1. **SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

1. Configuration Form- Register as Student or Admin,Personal ID & Pass
2. Books list- Name of the books available,No of bboks issued with name , Books name and ID, availability.
3. Basic Model of Library Management- number of Books , Admin Login,Student Login .
4. Search bar/query window for:
   * + - 1. Books Available/Issueed
         2. Student’s Details
         3. Books Category

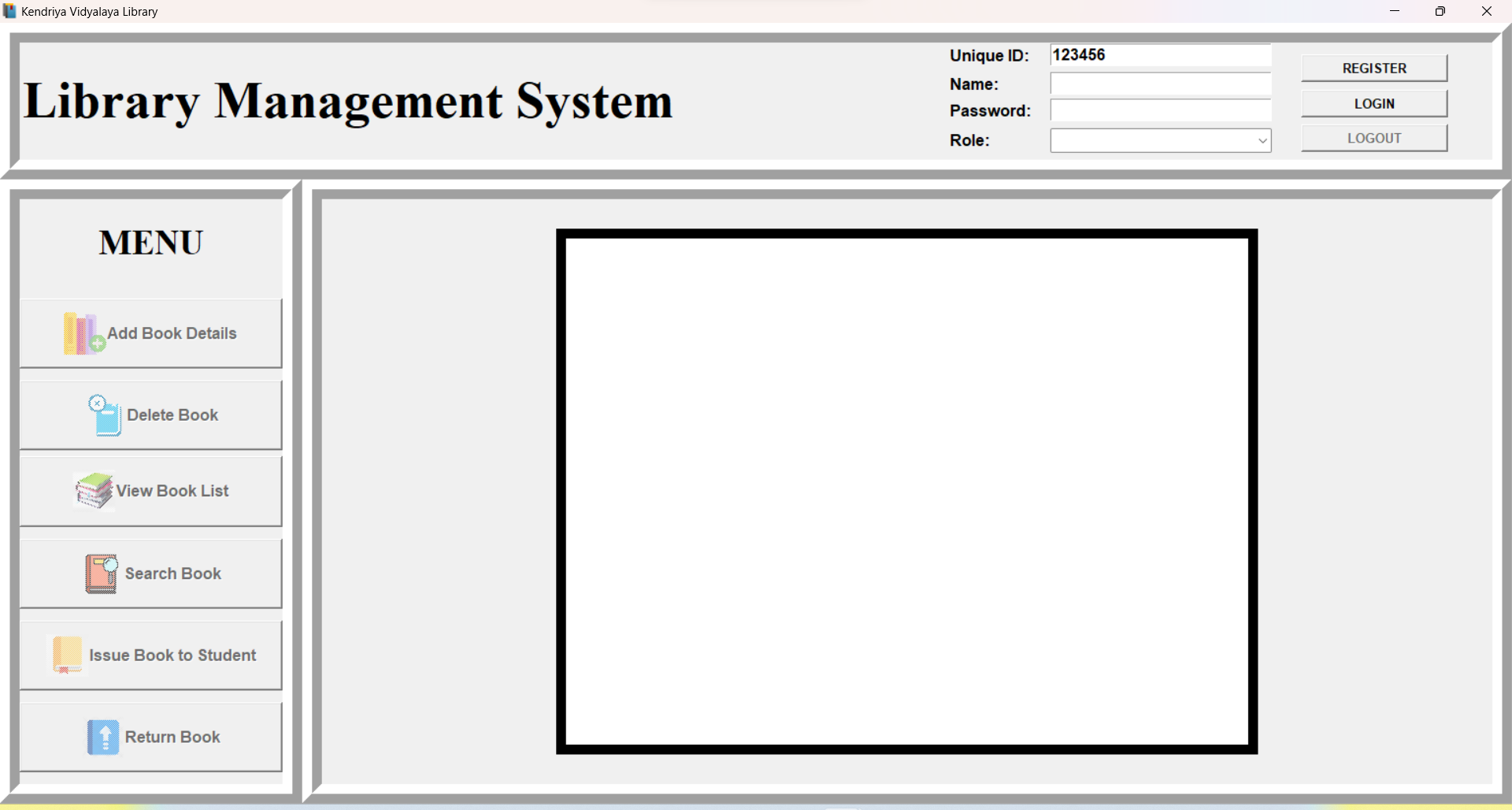
**CHAPTER 2**

**DESIGN DOCUMENT**

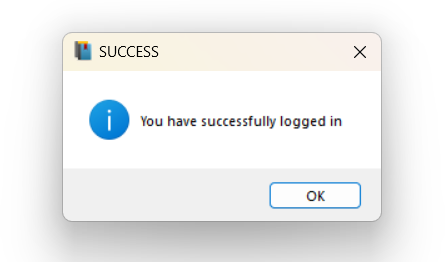
* 1. **DATABASE DESIGNING**
     1. ADMIN DETAILS

1. UNIQUE ID (INTEGERS) – Not Null, PRIMARY KEY
2. Name (MEDIUM TEXT) – Not Null
3. PASSWORD(INTEGERS) – Not Null
4. ROLE- ADMIN/STUDENT
   * 1. STUDENT DETAILS
5. UNIQUE ID (INTEGERS) – Not Null, PRIMARY KEY
6. Name (MEDIUM TEXT) – Not Null
7. PASSWORD(INTEGERS) – Not Null
8. ROLE- ADMIN/STUDENT
   * 1. ADDING BOOK DETAILS
     2. BOOK ID (INTEGERS) – Not Null, PRIMARY KEY
     3. TITLE OF BOOK (VARCHAR) – Not Null,
     4. SUBJECT (VARCHAR) – Not Null,PRIMARY KEY
     5. AUTHOR(VARCHAR) – Not Null
     6. DELETE BOOK DETAILS
     7. BOOK ID (INTEGERS) – Not Null, PRIMARY KEY
     8. VIEW BOOK LIST
     9. SHOWS ALL THE BOOKS OF THE LIBRARY
     10. SEARCH BOOKS
     + ACCORDING TO SUBJECT
     1. ISSUE BOOK TO STUDENT
9. BOOK ID (INTEGERS) – Not Null, PRIMARY KEY
10. ISSUED TO STUDENT ID(INTEGERS) – Not Null, PRIMARY KEY
11. ISSUED BY ADMIN ID (INTEGERS) – Not Null, PRIMARY KEY
12. VIEW ISSUED BOOKS & ISSUE BOOK
    * 1. RETURN BOOK
13. ISSUED TO STUDENT ID(INTEGERS) – Not Null, PRIMARY KEY
14. BOOK ID (INTEGERS) – Not Null, PRIMARY KEY
15. Return Button
    * 1. If student logins using his student id then he can only access too VIEW BOOKS AND SEARCH BOOKS WINDOW

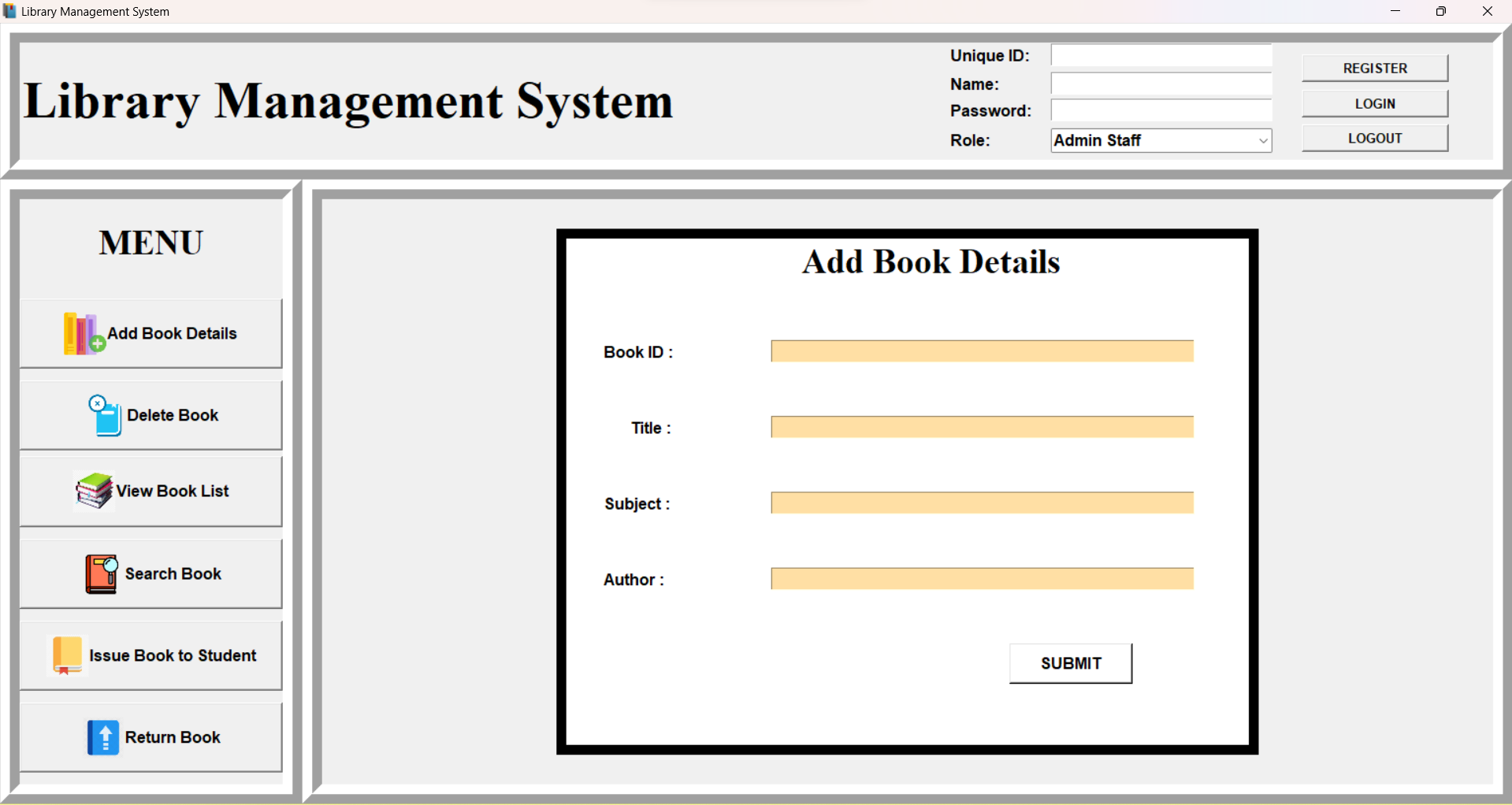
**2.2 GUI DESIGNING**

****

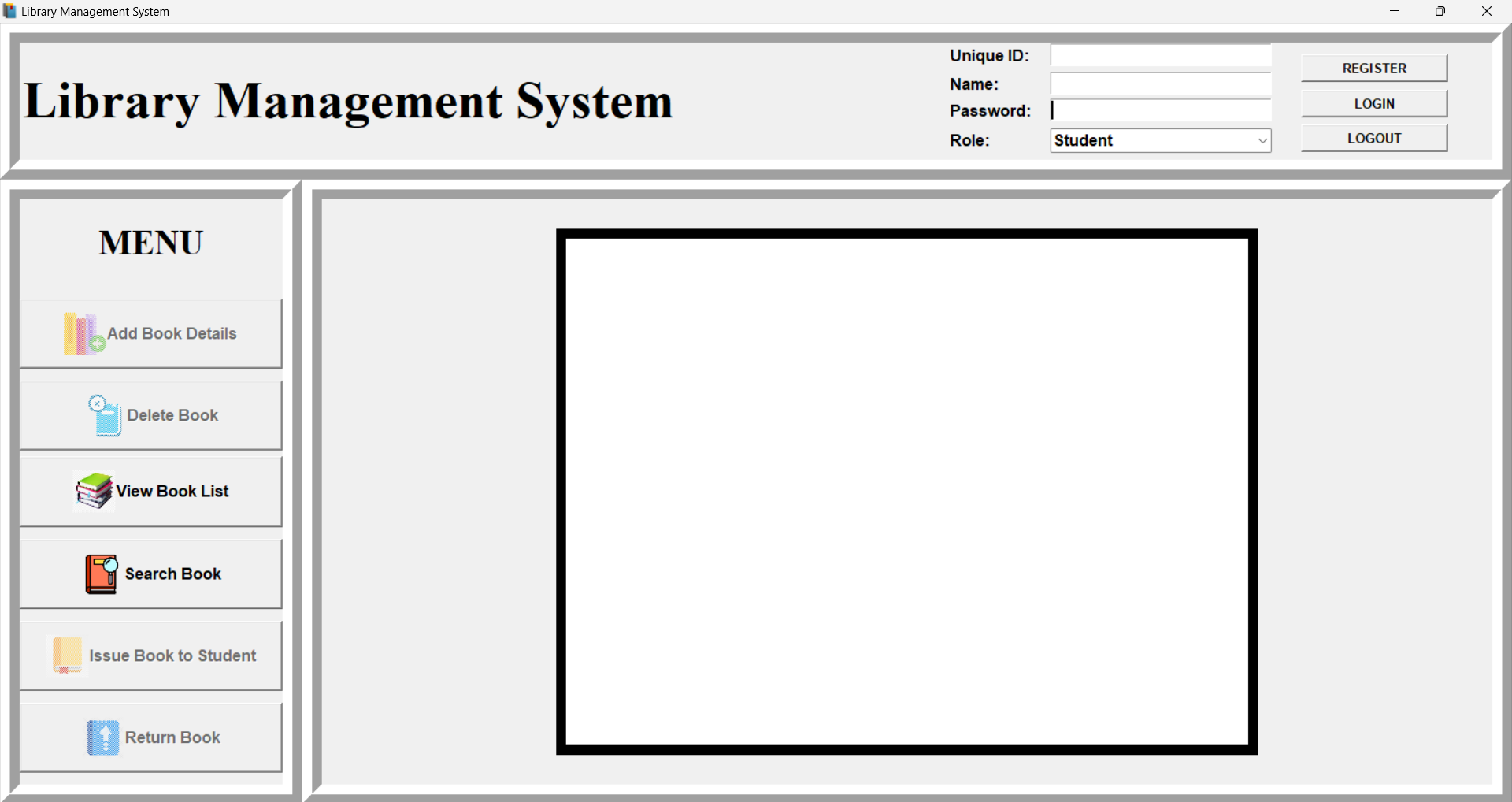
**Fig.2.1 Representation of the GUI Main Window**

****

**Fig.2.3 GUI Admin Login Window**

****

**Fig.2.3 Representation of the GUI Add Book Details Window**

**Fig.2.3 Representation of the GUI Student Login Window**

**CHAPTER 3**

**PROJECT IMPLEMENTATION**

**3.1 IMPLEMENTATION OF ALL REQUIRED MODULES CODE:**

from tkinter import \*

from tkinter import ttk

import sqlite3

from tkinter import messagebox

import os

**CREATING A TKINTER WINDOW:**

#Table Names

empTable = "empdetail" #Employee Table

stuTable = "studetail" #Student Table

bookTable = "books" # Book Table

issueTable = "issuedetail" #Issue Table

root = Tk()

root.title("Library Management System")

root.state("zoomed")

root.geometry("1920x1080")

root.iconbitmap(r'Icon.ico')

state1='disabled'

state2='disabled'

state3='disabled'

allRoll = [] #List To store all Roll Numbers

allEmpId = [] #List To store all Employee IDs

allBid = [] #List To store all Book IDs

num=0

**CONNECTION OF PYTHON TO SQLITE :**

#Creating Directory for database

path=os.environ["userprofile"]

try:

os.mkdir(path+"\\Documents\\Library Database")

except FileExistsError:

pass

con = sqlite3.connect(path+"\\Documents\\Library Database\mydatabase.db")

cur = con.cursor()

#Creating Tables

con.execute("create table if not exists empdetail (empid varchar(20) primary key,name varchar(30),password varchar(30));")

con.execute("create table if not exists studetail (rollno varchar(20) primary key,name varchar(30),password varchar(30));")

con.execute("create table if not exists books (bid varchar(20) primary key,title varchar(30),subject varchar(30),author varchar(30),status varchar(30) not null default 'Available');")

con.execute("create table if not exists issuedetail (bid varchar(20) primary key,issueto varchar(30),issueby varchar(30));")

def logout():

global num,state1,state2,state3

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

state1="disabled"

state2="disabled"

state3="disabled"

Menu()

messagebox.showinfo("Logged out", "You have Successfully logged out")

* 1. **IMPLEMENTATION OF LOGIN WINDOW DETAILS**

def gettingDetails():

Id = ent1.get()

name = ent2.get()

password = ent3.get()

role=ent4.get()

if role=="Admin Staff":

try:

if (type(int(Id)) == int):

pass

else:

messagebox.showinfo("Invalid Value","Unique ID should be an integer")

return

except:

messagebox.showinfo("Invalid Value","Unique ID should be an integer")

return

sql = "insert into "+empTable+" values ('"+Id+"','"+name+"','"+password+"')"

try:

cur.execute(sql)

con.commit()

messagebox.showinfo("Success", "Successfully registered")

except:

messagebox.showinfo("Error inserting","Cannot add data to Database")

else:

try:

if (type(int(Id)) == int):

pass

else:

messagebox.showinfo("Invalid Value","Unique ID should be an integer")

return

except:

messagebox.showinfo("Invalid Value","Unique ID should be an integer")

return

sql = "insert into "+stuTable+" values ('"+Id+"','"+name+"','"+password+"')"

try:

cur.execute(sql)

con.commit()

messagebox.showinfo("Success", "Successfully registered")

except:

messagebox.showinfo("Error inserting","Cannot add data to Database")

ent1.delete(0, END)

ent2.delete(0, END)

ent3.delete(0, END)

ent4.delete(0, END)

def gettingLoginDetails():

global role,state1,state2,btn1,btn2,btn3,btn4,btn5,btn6,state3

Id = ent1.get()

name = ent2.get()

password = ent3.get()

role=ent4.get()

if role=='Admin Staff':

sqlLoginID = "select empid from "+empTable+" where password = '"+password+"'"

sqlName = "select name from "+empTable+" where password = '"+password+"'"

try:

cur.execute(sqlLoginID)

for i in cur:

getLoginID = i[0]

cur.execute(sqlName)

for i in cur:

getName = i[0]

if(getLoginID == Id and getName == name):

messagebox.showinfo("SUCCESS","You have successfully logged in")

btn1.destroy()

btn2.destroy()

btn3.destroy()

btn4.destroy()

btn5.destroy()

btn6.destroy()

state1='normal'

state2='normal'

state3='normal'

Menu()

else:

messagebox.showerror("Failure","Can't log in, check your credentials")

except:

messagebox.showinfo("FAILED","Please check your credentials")

else:

sqlLoginID = "select rollno from "+stuTable+" where password = '"+password+"'"

sqlName = "select name from "+stuTable+" where password = '"+password+"'"

try:

cur.execute(sqlLoginID)

for i in cur:

getLoginID = i[0]

cur.execute(sqlName)

for i in cur:

getName = i[0]

if(getLoginID == Id and getName == name):

btn1.destroy()

btn2.destroy()

btn3.destroy()

btn4.destroy()

btn5.destroy()

btn6.destroy()

state2='normal'

state3='normal'

Menu()

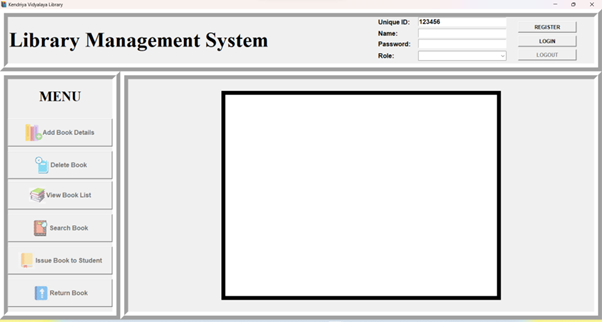
messagebox.showinfo("SUCCESS","You have successfully logged in")

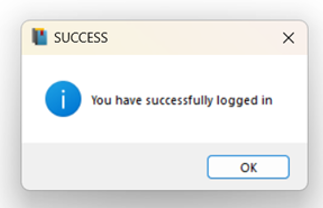
else:

messagebox.showerror("Failure","Can't log in, check your credentials")

except:

messagebox.showinfo("FAILED","Please check your credentials")

 **GUI WINDOW OF THE ABOVE CODE:**



ent1.delete(0, END)

ent2.delete(0, END)

ent3.delete(0, END)

* 1. **MAKING MENU OF ADD,DELETE,VIEW BOOK LIST,SEARCH, ISSUE BOOKS TO STUDENT,RETURN BOOK**

3.3.1 CODE:

def Menu():

global state1,state2,btn1,btn2,btn3,btn4,btn5,btn6,photoimage1, photoimage2, photoimage3, photoimage4, photoimage5, photoimage6

photo1 = PhotoImage(file =r"icons\Add book.png")

photoimage1 = photo1.subsample(3, 3)

btn1 = Button(moduleFrame,text="Add Book Details",font=("arial",12,'bold'),image=photoimage1,compound = LEFT, command=addBooks, state=state1)

btn1.place(relx=0,rely=0.17, relwidth=1,relheight=0.12)

photo2 = PhotoImage(file =r"icons\delete book.png")

photoimage2 = photo2.subsample(3,3)

btn2 = Button(moduleFrame,text="Delete Book",font=("arial",12,'bold'),image=photoimage2, compound= LEFT, command=delete,state=state1)

btn2.place(relx=0,rely=0.31, relwidth=1,relheight=0.12)

photo3 = PhotoImage(file =r"icons\view book.png")

photoimage3 = photo3.subsample(3,3)

btn3 = Button(moduleFrame,text="View Book List",font=("arial",12,'bold'),image=photoimage3, compound= LEFT, command=View,state=state2)

btn3.place(relx=0, rely=0.44, relwidth=1,relheight=0.12)

photo4 = PhotoImage(file =r"icons\search book.png")

photoimage4 = photo4.subsample(3,3)

btn4 = Button(moduleFrame,text="Search Book",font=("arial",12,'bold'),image=photoimage4, compound= LEFT, command=searchBook,state=state2)

btn4.place(relx=0,rely=0.58, relwidth=1,relheight=0.12)

photo5 = PhotoImage(file =r"icons\issue book.png")

photoimage5 = photo5.subsample(3,3)

btn5 = Button(moduleFrame,text="Issue Book to Student",font=("arial",12,'bold'),image=photoimage5, compound = LEFT, command = issueBook,state=state1)

btn5.place(relx=0,rely=0.72, relwidth=1,relheight=0.12)

photo6 = PhotoImage(file =r"icons\return book.png")

photoimage6 = photo6.subsample(3,3)

btn6 = Button(moduleFrame, text="Return Book",font=("arial",12,'bold'), image=photoimage6, compound = LEFT, command=ReturnBook,state=state1)

btn6.place(relx=0,rely=0.86, relwidth=1,relheight=0.12)

logoutBtn=Button(headingFrame, text="LOGOUT", font=('arial',10,'bold'), command=logout, state=state3)

logoutBtn.place(relx=0.87, rely=0.7, relwidth=0.1)

########################## Add Boook ########################

def bookRegister():

global en1,en2,en3,en4

bid = en1.get()

title = en2.get()

title=title.title()

subject = en3.get()

subject=subject.title()

author = en4.get()

author=author.title()

insertBooks = "insert into "+bookTable+" (bid,title,subject,author) values('"+bid+"','"+title+"','"+subject+"','"+author+"')"

if bid=="" or title=="" or subject=="" or author=="":

messagebox.showinfo("Error", "Please fill all the details")

else:

try:

cur.execute(insertBooks)

con.commit()

messagebox.showinfo("Sucess","Book added")

except:

messagebox.showinfo("Error","Can't add data into Database")

en1.delete(0, END)

en2.delete(0, END)

en3.delete(0, END)

en4.delete(0, END)

def addBooks():

global en1,en2,en3,en4,lb1,lb2,lb3,lb4,SubmitBtn,lb,num

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

lb=Label(displayFrame,text='Add Book Details',font=("Times New Roman",26,'bold'),bg='white')

lb.place(relx=0.34,rely=0)

# Book ID

lb1 = Label(displayFrame,text="Book ID : ",font=("arial",12,'bold'),bg='white')

lb1.place(relx=0.05,rely=0.2)

en1 = Entry(displayFrame,font=("arial",12,'bold'),bg='#FFDFA4')

en1.place(relx=0.3,rely=0.2, relwidth=0.62)

# Title

lb2 = Label(displayFrame,text="Title : ",font=("arial",12,'bold'),bg='white')

lb2.place(relx=0.09,rely=0.35)

en2 = Entry(displayFrame,font=("arial",12,'bold'),bg='#FFDFA4')

en2.place(relx=0.3,rely=0.35, relwidth=0.62)

# Book Subject

lb3 = Label(displayFrame,text="Subject :",font=("arial",12,'bold'),bg='white')

lb3.place(relx=0.05,rely=0.5)

en3 = Entry(displayFrame, font=("arial",12,'bold'), bg='#FFDFA4')

en3.place(relx=0.3,rely=0.5, relwidth=0.62)

# Book Author

lb4 = Label(displayFrame,text="Author : ",font=("arial",12,'bold'),bg='white')

lb4.place(relx=0.05,rely=0.65)

en4 = Entry(displayFrame,font=("arial",12,'bold'), bg='#FFDFA4')

en4.place(relx=0.3,rely=0.65, relwidth=0.62)

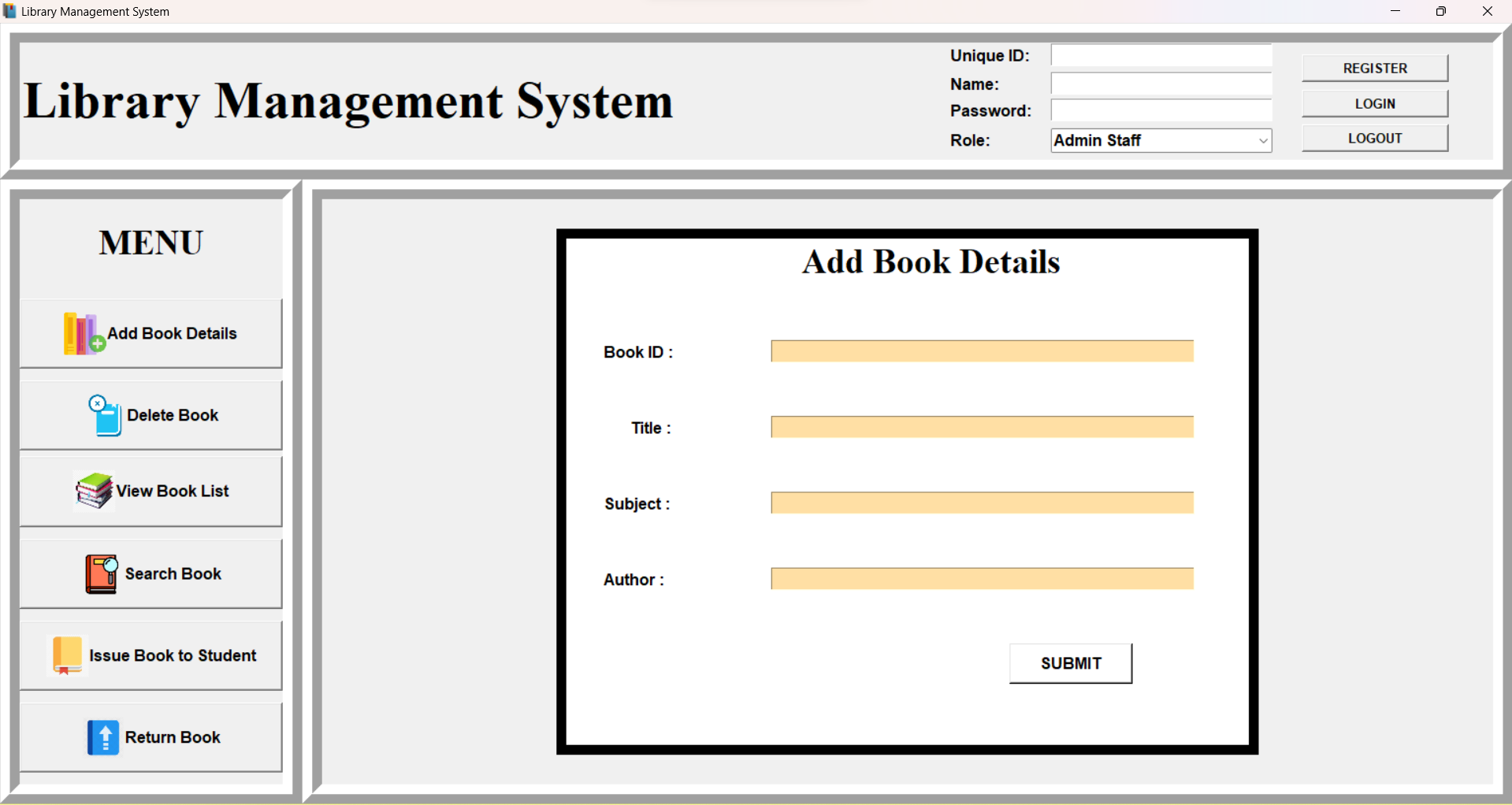
#Submit Button

SubmitBtn = Button(displayFrame,text="SUBMIT",font=("arial",12,'bold'),bg='white',command=bookRegister)

SubmitBtn.place(relx=0.65,rely=0.8, relwidth=0.18,relheight=0.08)

num=1

* + 1. **GUI WINDOW OF THE ABOVE CODE:**

****

######################## Delete BOOK #########################

def deleteBook():

bid = en1.get()

deleteSql = "delete from "+bookTable+" where bid = '"+bid+"'"

if bid=='':

messagebox.showinfo("","Please enter Book ID")

else:

try:

cur.execute(deleteSql)

con.commit()

messagebox.showinfo("Success","Book Deleted Successfully")

except:

messagebox.showinfo("Check Credentials","Please check Book ID")

en1.delete(0, END)

def delete():

global en1, lb1, SubmitBtn,lb,num

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

lb = Label(displayFrame,text='Delete Book',font=("Times New Roman",26,'bold'),bg='white')

lb.place(relx=0.37,rely=0)

# Book ID to Delete

lb1 = Label(displayFrame,text="Book ID : ",font=("arial",12,'bold'), bg='white')

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

en1 = Entry(displayFrame,font=("arial",12,'bold'),bg='#FFDFA4')

en1.place(relx=0.3,rely=0.5, relwidth=0.62)

#Submit Button

SubmitBtn = Button(displayFrame,text="SUBMIT", font=("arial",12,'bold'),bg='white',command=deleteBook)

SubmitBtn.place(relx=0.6,rely=0.75, relwidth=0.18,relheight=0.08)

num=2

######################## View Issued Books #########################

def displayissuedbooks():

global scroll\_y,issue\_table,num

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

show= "select issuedetail.bid, books.title, issuedetail.issueto, issuedetail.issueby from " +bookTable+ " inner join issuedetail on issuedetail.bid=books.bid"

scroll\_y=Scrollbar(displayFrame,orient=VERTICAL)

issue\_table=ttk.Treeview(displayFrame,columns=("bid","bname","issuedto","issueby"),yscrollcommand=scroll\_y.set)

scroll\_y.pack(side=RIGHT, fill=Y)

scroll\_y.config(command=issue\_table.yview)

issue\_table.heading("bid",text="Book ID")

issue\_table.heading("bname",text="Title")

issue\_table.heading("issuedto",text="Issued To")

issue\_table.heading("issueby",text="Issued By")

issue\_table["show"]="headings"

issue\_table.column("bid",width=50)

issue\_table.column("bname",width=50)

issue\_table.column("issuedto",width=50)

issue\_table.column("issueby",width=50)

issue\_table.pack(fill=BOTH,expand=1)

try:

cur.execute(show)

con.commit()

for i in cur:

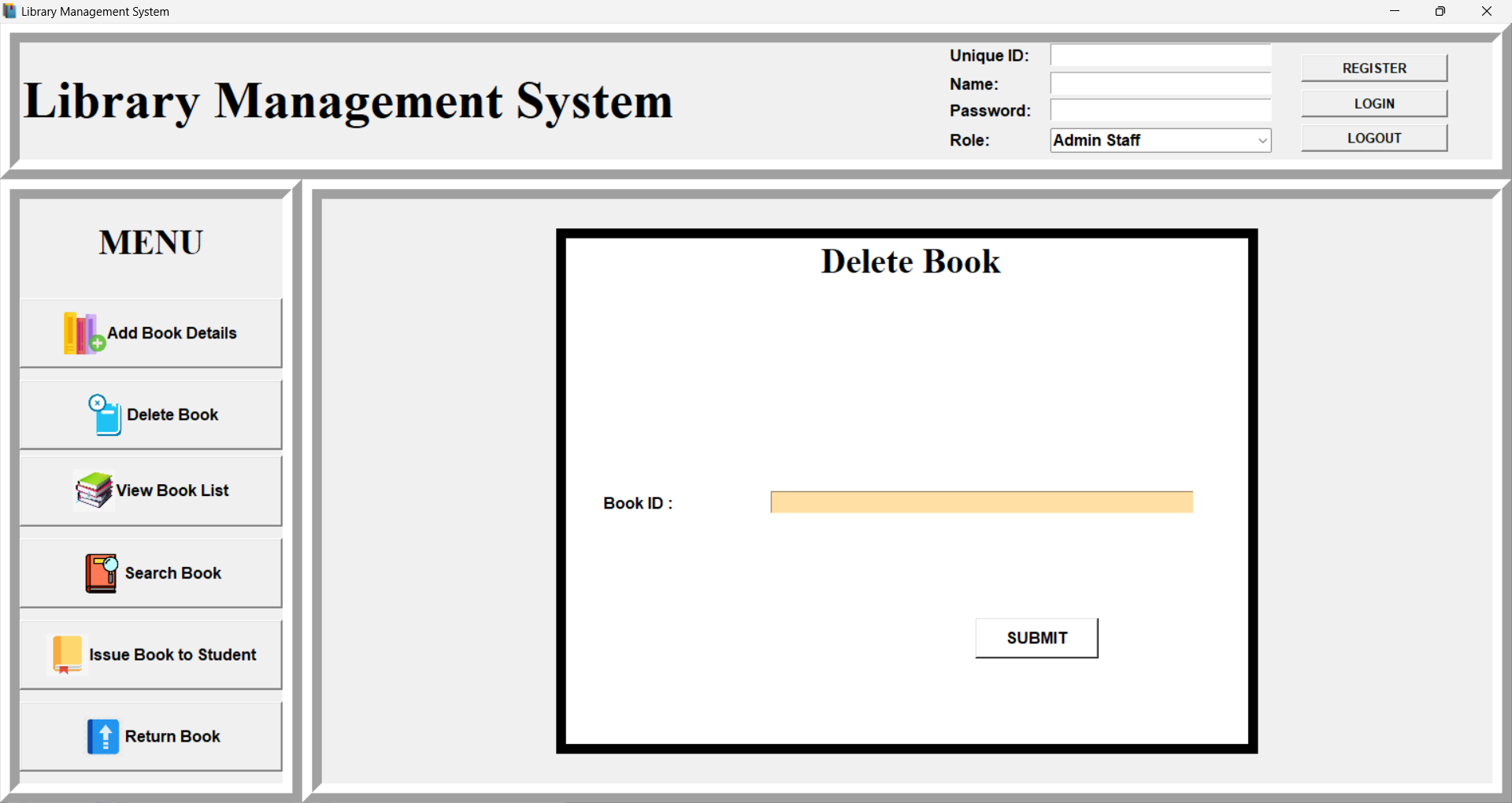
issue\_table.insert('',END,values=i)

except:

messagebox.showinfo("Error","Failed to fetch data")

num=4

**3.3.3 GUI WINDOW OF THE ABOVE CODE:**



####################### Issue BOOK############################

def issue():

#global lb1,lb2,lb3,en1,en2,en3,status,scroll\_y,issue\_table,num

bid = en1.get()

issueto = en2.get()

issueby = en3.get()

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 == 'Available':

status = True

else:

status = False

else:

messagebox.showinfo("Error","Book ID not present")

except:

messagebox.showinfo("Error","Can't fetch Book IDs")

extractRollno = "select rollno from "+stuTable

try:

cur.execute(extractRollno)

con.commit()

for i in cur:

allRoll.append(i[0])

if issueto in allRoll:

pass

else:

messagebox.showinfo("Error","Roll No not present")

except:

messagebox.showinfo("Error","Can't fetch Roll No")

extractEmpID = "select empid from "+empTable

try:

cur.execute(extractEmpID)

con.commit()

for i in cur:

allEmpId.append(i[0])

if issueby in allEmpId:

pass

else:

messagebox.showinfo("Error","Emp ID not present")

except:

messagebox.showinfo("Error","Can't fetch Emp IDs")

issueSql = "insert into "+issueTable+" values ('"+bid+"','"+issueto+"','"+issueby+"')"

updateStatus = "update "+bookTable+" set status = 'issued' where bid = '"+bid+"'"

try:

if bid in allBid and issueto in allRoll and issueby in allEmpId and status == True:

cur.execute(issueSql)

con.commit()

cur.execute(updateStatus)

con.commit()

messagebox.showinfo("Success"," Issued Book successfully")

else:

allBid.clear()

allEmpId.clear()

allRoll.clear()

return

except:

messagebox.showinfo("Search Error","The value entered is wrong, Try again")

en1.delete(0,END)

en2.delete(0,END)

en3.delete(0,END)

allBid.clear()

allEmpId.clear()

allRoll.clear()

def issueBook():

global en1,en2,en3,issueBtn,lb1,lb2,lb3,en1,en2,en3,lb,num,issuedBooks

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

lb = Label(displayFrame,text="Issue Book to Student", font=('Times New Roman',26,'bold'), bg='white')

lb.place(relx=0.27,rely=0)

# Book ID

lb1 = Label(displayFrame,text="Book ID : ",font=('arial', 12,'bold'), bg='white')

lb1.place(relx=0.05,rely=0.2)

en1 = Entry(displayFrame,font=('arial', 12,'bold'),bg='#FFDFA4')

en1.place(relx=0.4,rely=0.2, relwidth=0.55)

# Issued To Roll Number

lb2 = Label(displayFrame,text="Issued To(Student Unique ID) : ",font=('arial', 12,'bold'),bg='white')

lb2.place(relx=0.05,rely=0.4)

en2 = Entry(displayFrame,font=('arial', 12,'bold'),bg='#FFDFA4')

en2.place(relx=0.4,rely=0.4, relwidth=0.55)

# Issued By Employee Number

lb3 = Label(displayFrame,text="Issued By(Admin Unique ID) : ",font=('arial', 12,'bold'), bg='white')

lb3.place(relx=0.05,rely=0.6)

en3 = Entry(displayFrame,font=('arial', 12,'bold'),bg='#FFDFA4')

en3.place(relx=0.4,rely=0.6, relwidth=0.55)

#Issue Button

issueBtn = Button(displayFrame,text="Issue",bg='white',font=("arial",12,'bold'), command=issue)

issueBtn.place(relx=0.7,rely=0.75, relwidth=0.18,relheight=0.08)

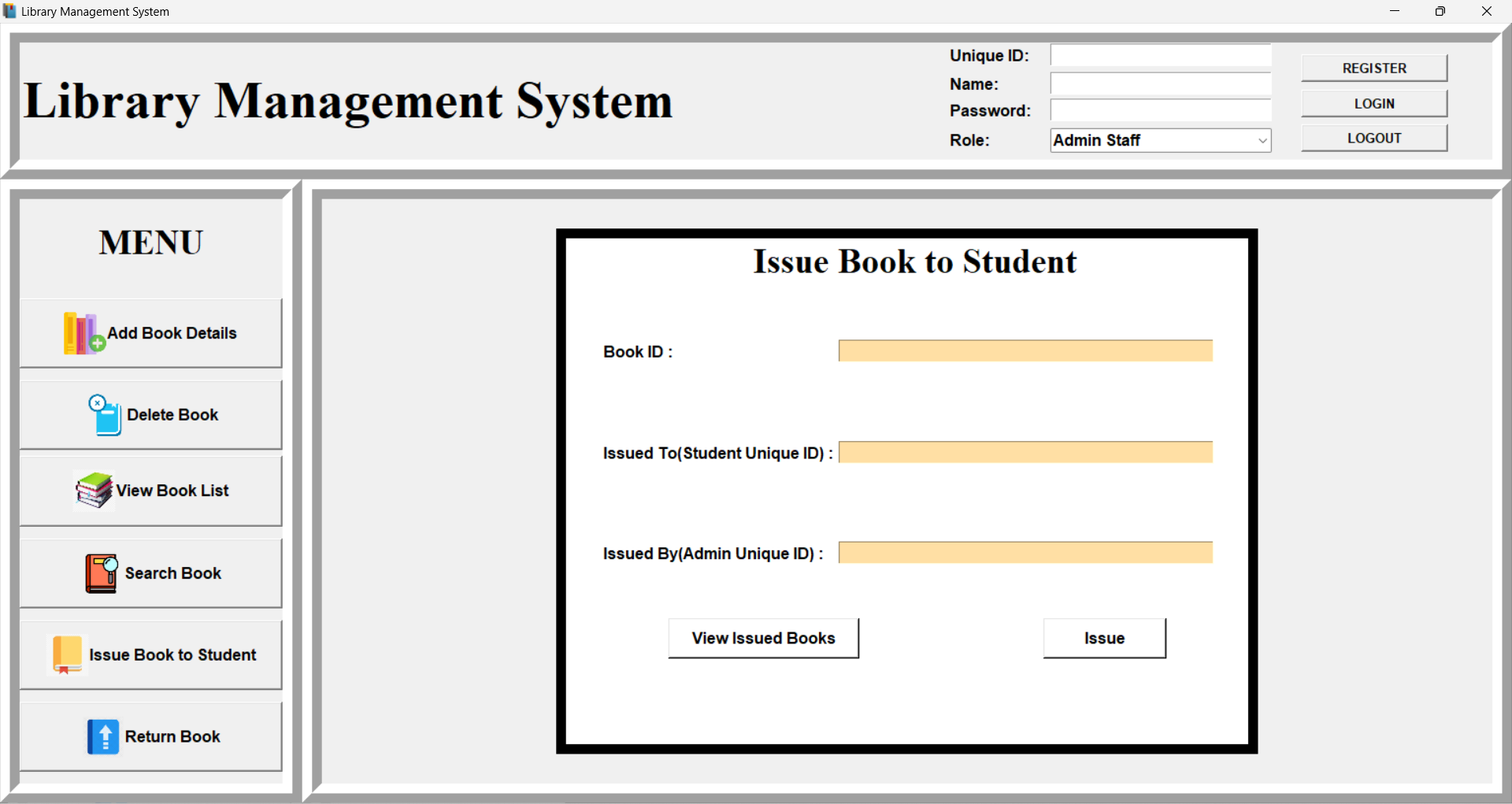
#View Issued Books

issuedBooks= Button(displayFrame,text="View Issued Books",bg='white',font=("arial",12,'bold'), command=displayissuedbooks)

issuedBooks.place(relx=0.15,rely=0.75, relwidth=0.28,relheight=0.08)

num=3

**3.3.4 GUI WINDOW OF THE ABOVE CODE:**



############### Return Book #####################

def Return():

global ReturnBtn,labelFrame,lb1,en1,en2,status

bookid = en1.get()

Returnby = en2.get()

extractBid = "select bid from "+bookTable

try:

cur.execute(extractBid)

con.commit()

for i in cur:

allBid.append(i[0])

if bookid in allBid:

checkAvail = "select status from "+bookTable+" where bid = '"+bookid+"'"

cur.execute(checkAvail)

con.commit()

for i in cur:

check = i[0]

if check == 'Available':

status = True

else:

status = False

else:

messagebox.showinfo("Error","Book ID not present in database")

except:

messagebox.showinfo("Error","Can't fetch Book IDs")

extractRollno = "select rollno from "+stuTable

try:

cur.execute(extractRollno)

con.commit()

for i in cur:

allRoll.append(i[0])

if Returnby in allRoll:

pass

else:

messagebox.showinfo("Error","Student ID not present")

except:

messagebox.showinfo("Error","Can't fetch Student ID")

ReturnSql = "delete from "+issueTable+" where bid="+bookid

updateStatus = "update "+bookTable+" set status ='Available' where bid="+bookid

try:

if bookid in allBid and Returnby in allRoll and status == False:

cur.execute(ReturnSql)

con.commit()

cur.execute(updateStatus)

con.commit()

messagebox.showinfo("Successs","Book returned successfully")

else:

allBid.clear()

allRoll.clear()

except:

messagebox.showinfo("Search Error","The value entered is wrong, Try again")

en1.delete(0, END)

en2.delete(0, END)

allBid.clear()

allRoll.clear()

def ReturnBook():

global en1,en2,ReturnBtn,lb1,lb2,headingLabel,num

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

headingLabel = Label(displayFrame, text="RETURN BOOK",font=('Times New Roman',26,'bold'),bg='white')

headingLabel.place(relx=0.35,rely=0.05)

# Book ID

lb1 = Label(displayFrame,text="Book ID : ",font=('arial',12,'bold'),bg='white')

lb1.place(relx=0.05,rely=0.3)

en1 = Entry(displayFrame,font=('arial',12,'bold'),bg='#FFDFA4')

en1.place(relx=0.3,rely=0.3, relwidth=0.62)

# Student ID

lb2 = Label(displayFrame,text="Student ID : ",font=('arial',12,'bold'),bg='white')

lb2.place(relx=0.05,rely=0.6)

en2 = Entry(displayFrame,font=('arial',12,'bold'),bg='#FFDFA4')

en2.place(relx=0.3,rely=0.6, relwidth=0.62)

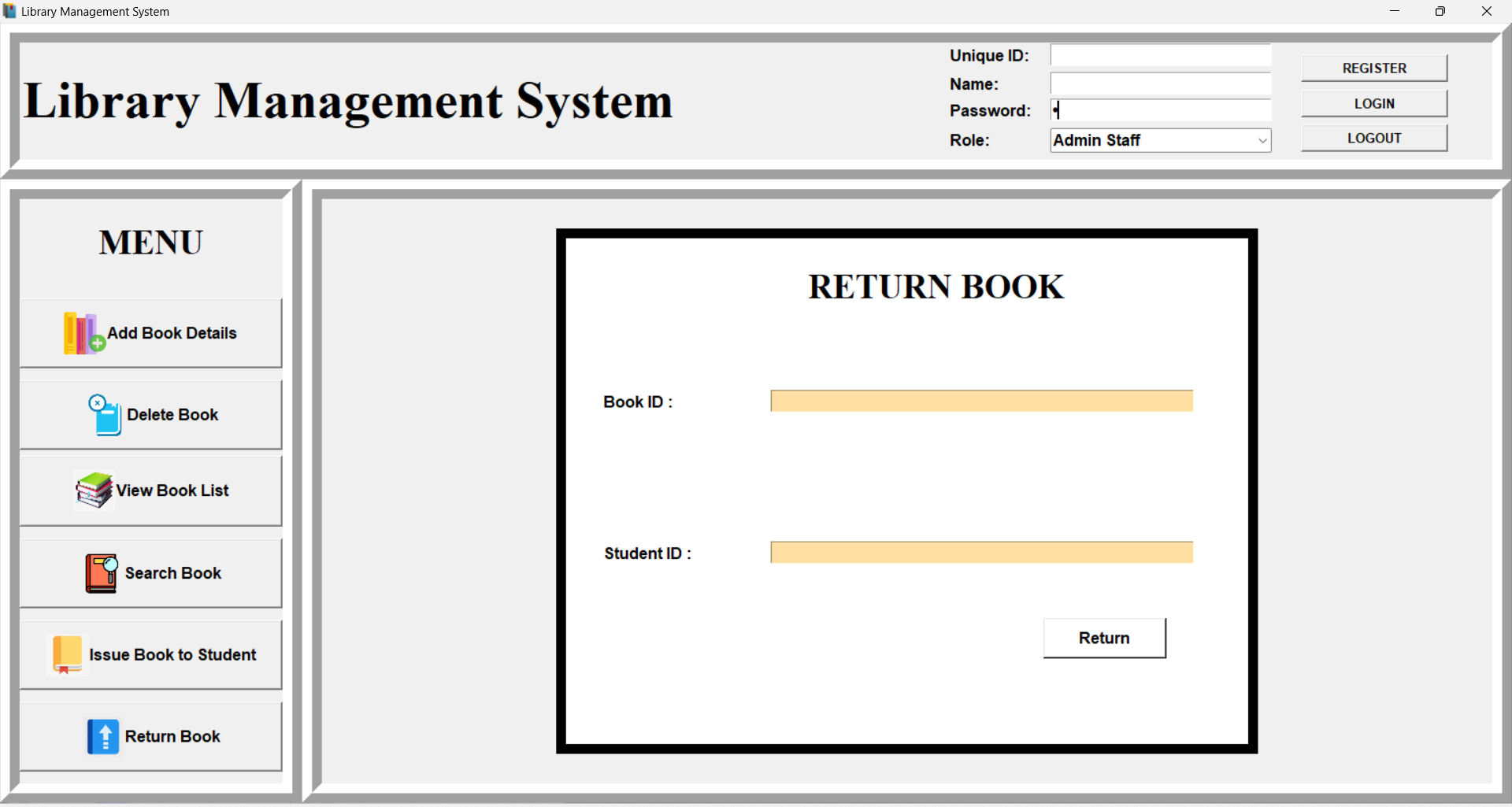
#Return Button

ReturnBtn = Button(displayFrame,text="Return",font=('arial',12,'bold'),bg='white',command=Return)

ReturnBtn.place(relx=0.7,rely=0.75, relwidth=0.18,relheight=0.08)

num=5

**3.3.5 GUI WINDOW OF THE ABOVE CODE:**



############# Search Book ##############################

def search():

global SearchBtn,labelFrame,lb1,en1,scroll\_y, books\_table,num

sub = en1.get()

sub=sub.title()

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

scroll\_y=Scrollbar(displayFrame,orient=VERTICAL)

books\_table=ttk.Treeview(displayFrame,columns=("bid","title","subject","author","status"),yscrollcommand=scroll\_y.set)

scroll\_y.pack(side=RIGHT, fill=Y)

scroll\_y.config(command=books\_table.yview)

books\_table.heading("bid",text="Book ID")

books\_table.heading("title",text="Title")

books\_table.heading("subject",text="Subject")

books\_table.heading("author",text="Author")

books\_table.heading("status",text="Status")

books\_table["show"]="headings"

books\_table.column("bid",width=50)

books\_table.column("title",width=50)

books\_table.column("subject",width=50)

books\_table.column("author",width=50)

books\_table.column("status",width=50)

books\_table.pack(fill=BOTH,expand=1)

num=7

searchSql = "select \* from "+bookTable+" where subject = '"+sub+"'"

try:

cur.execute(searchSql)

con.commit()

rows=cur.fetchall()

for row in rows:

books\_table.insert('',END,values=row)

except:

messagebox.showinfo("Search Error","The value entered is wrong, Try again")

def searchBook():

global en1,SearchBtn,lb1, headingLabel,num

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

headingLabel = Label(displayFrame, text="Search Book",font=("Times New Roman",26,'bold'), bg='white')

headingLabel.place(relx=0.35,rely=0.05)

# Book ID to Delete

lb1 = Label(displayFrame,text="Enter Subject : ",font=("arial",12,'bold'), bg='white')

lb1.place(relx=0.05,rely=0.4)

en1 = Entry(displayFrame,font=("arial",12,'bold'),bg='#FFDFA4')

en1.place(relx=0.3,rely=0.4, relwidth=0.62)

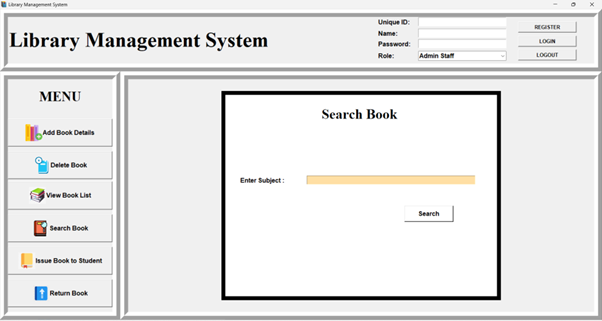
#Submit Button

SearchBtn = Button(displayFrame,text="Search",font=('arial',12,'bold'),bg='white',command=search)

SearchBtn.place(relx=0.66,rely=0.55, relwidth=0.18,relheight=0.08)

num=6

**3.3.6 GUI WINDOW OF THE ABOVE CODE**



####################### View Books############################

def View():

global scroll\_y, books\_table,num,headingLabel

if num==1:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

lb4.destroy()

en3.destroy()

en4.destroy()

SubmitBtn.destroy()

elif num==2:

lb.destroy()

lb1.destroy()

en1.destroy()

SubmitBtn.destroy()

elif num==3:

lb.destroy()

lb1.destroy()

en1.destroy()

lb2.destroy()

en2.destroy()

lb3.destroy()

en3.destroy()

issueBtn.destroy()

issuedBooks.destroy()

elif num==4:

scroll\_y.destroy()

issue\_table.destroy()

elif num==5:

en1.destroy()

en2.destroy()

ReturnBtn.destroy()

lb1.destroy()

lb2.destroy()

headingLabel.destroy()

elif num==6:

en1.destroy()

lb1.destroy()

headingLabel.destroy()

SearchBtn.destroy()

elif num==7 or num==8:

scroll\_y.destroy()

books\_table.destroy()

else:

pass

scroll\_y=Scrollbar(displayFrame,orient=VERTICAL)

books\_table=ttk.Treeview(displayFrame,columns=("bid","title","subject","author","status"),yscrollcommand=scroll\_y.set)

scroll\_y.pack(side=RIGHT, fill=Y)

scroll\_y.config(command=books\_table.yview)

books\_table.heading("bid",text="Book ID")

books\_table.heading("title",text="Title")

books\_table.heading("subject",text="Subject")

books\_table.heading("author",text="Author")

books\_table.heading("status",text="Status")

books\_table["show"]="headings"

books\_table.column("bid",width=50)

books\_table.column("title",width=50)

books\_table.column("subject",width=50)

books\_table.column("author",width=50)

books\_table.column("status",width=50)

books\_table.pack(fill=BOTH,expand=1)

num=8

getBooks = "select \* from "+bookTable

try:

cur.execute(getBooks)

con.commit()

rows=cur.fetchall()

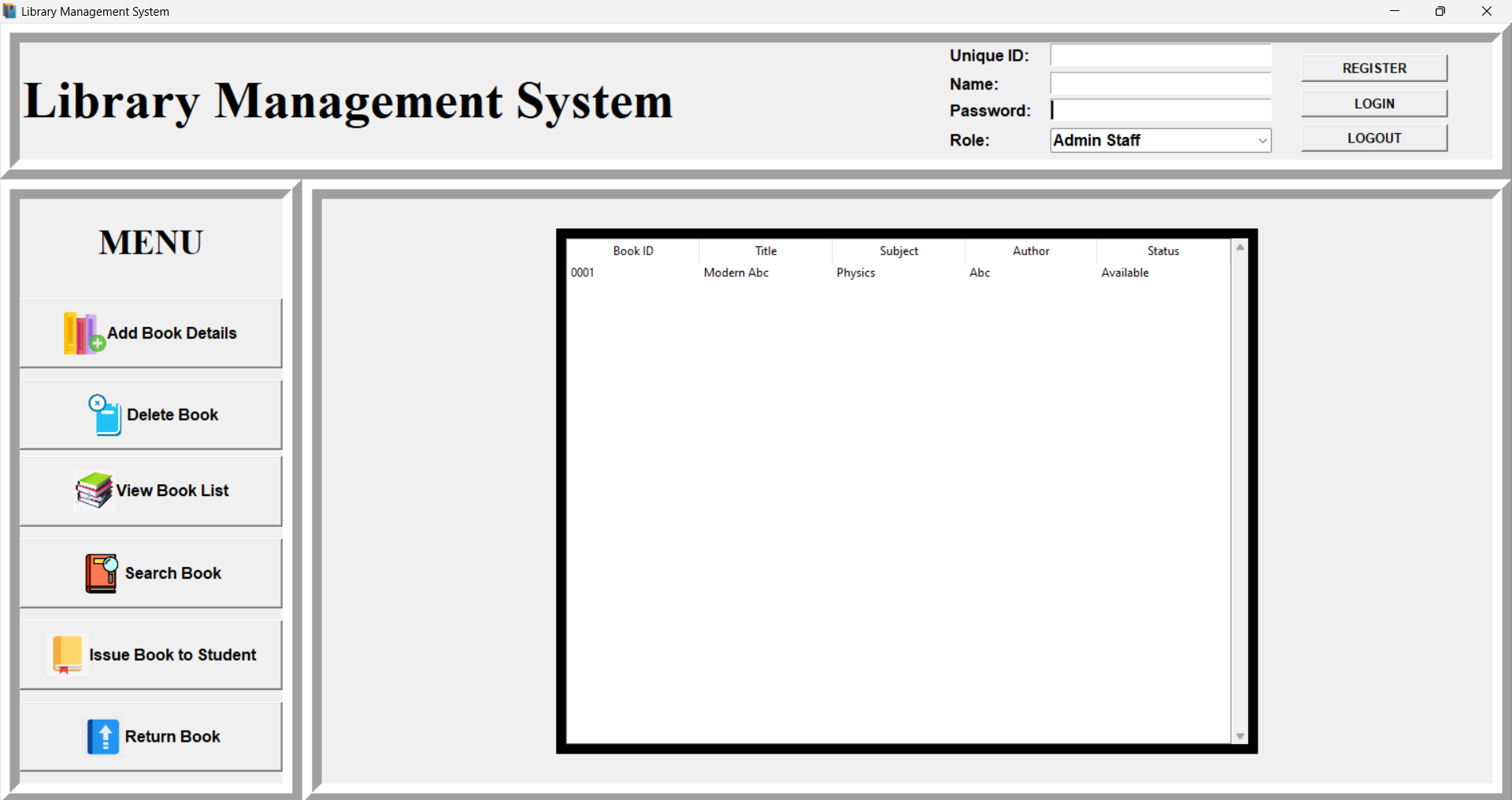
for row in rows:

books\_table.insert('',END,values=row)

except:

messagebox.showinfo("Error","Can't fetch data from database")

**3.3.7 GUI WINDOW OF THE ABOVE CODE:**



################## Frames #################################

headingFrame = Frame(root,bd=20, relief=RIDGE)

headingFrame.place(relx=0,rely=0,relwidth=1,relheight=0.2)

heading=Label(headingFrame,font=('Times New ROman', 40, 'bold'), text="Library Management System")

heading.place(relx=0, rely=0.2)

moduleFrame = Frame(root,bd=20, relief=RIDGE)

moduleFrame.place(relx=0,rely=0.2,relwidth=0.2,relheight=0.8)

headingLabel = Label(moduleFrame, text="MENU",font=("Times New Roman",26,'bold'))

headingLabel.place(relx=0,rely=0, relwidth=1, relheight=0.15)

dFrame= Frame(root,bd=20, relief=RIDGE)

dFrame.place(relx=0.2,rely=0.2,relwidth=0.8,relheight=0.8)

displayFrame=Frame(dFrame, bd=10, relief=SOLID, bg='white')

displayFrame.place(relx=0.2,rely=0.05, relwidth=0.6, relheight=0.9)

###################### Interface for login and registration ##############

# ID

lb1=Label(headingFrame,text='Unique ID:',font=('arial', 12,'bold'))

lb1.place(relx=0.63, rely=0.01)

ent1=Entry(headingFrame, font=('arial', 12,'bold'))

ent1.place(relx=0.7, rely=0.01, relwidth=0.15)

#Name

lb2=Label(headingFrame,text='Name:',font=('arial', 12,'bold'))

lb2.place(relx=0.63, rely=0.25)

ent2=Entry(headingFrame, font=('arial', 12,'bold'))

ent2.place(relx=0.7, rely=0.25, relwidth=0.15)

#Password

lb3=Label(headingFrame,text='Password:',font=('arial', 12,'bold'))

lb3.place(relx=0.63, rely=0.48)

ent3=Entry(headingFrame, font=('arial', 12,'bold'),show="\u2022")

ent3.place(relx=0.7, rely=0.48, relwidth=0.15)

#Role Combobox

lb4=Label(headingFrame,text='Role:',font=('arial', 12,'bold'))

lb4.place(relx=0.63, rely=0.73)

ent4=ttk.Combobox(headingFrame, font=('arial', 12, 'bold'), state='readonly', width=23)

ent4['value']=('','Admin Staff', 'Student')

ent4.current(0)

ent4.place(relx=0.7,rely=0.73,relwidth=0.15)

loginBtn = Button(headingFrame,text="LOGIN", font=("arial",10,'bold'),command=gettingLoginDetails)

loginBtn.place(relx=0.87,rely=0.4, relwidth=0.1)

regBtn=Button(headingFrame,text="REGISTER", font=("arial",10,'bold'),command=gettingDetails)

regBtn.place(relx=0.87,rely=0.1, relwidth=0.1)

Menu()

root.mainloop()

**3.3.8 GUI WINDOW OF THE ABOVE CODE:**

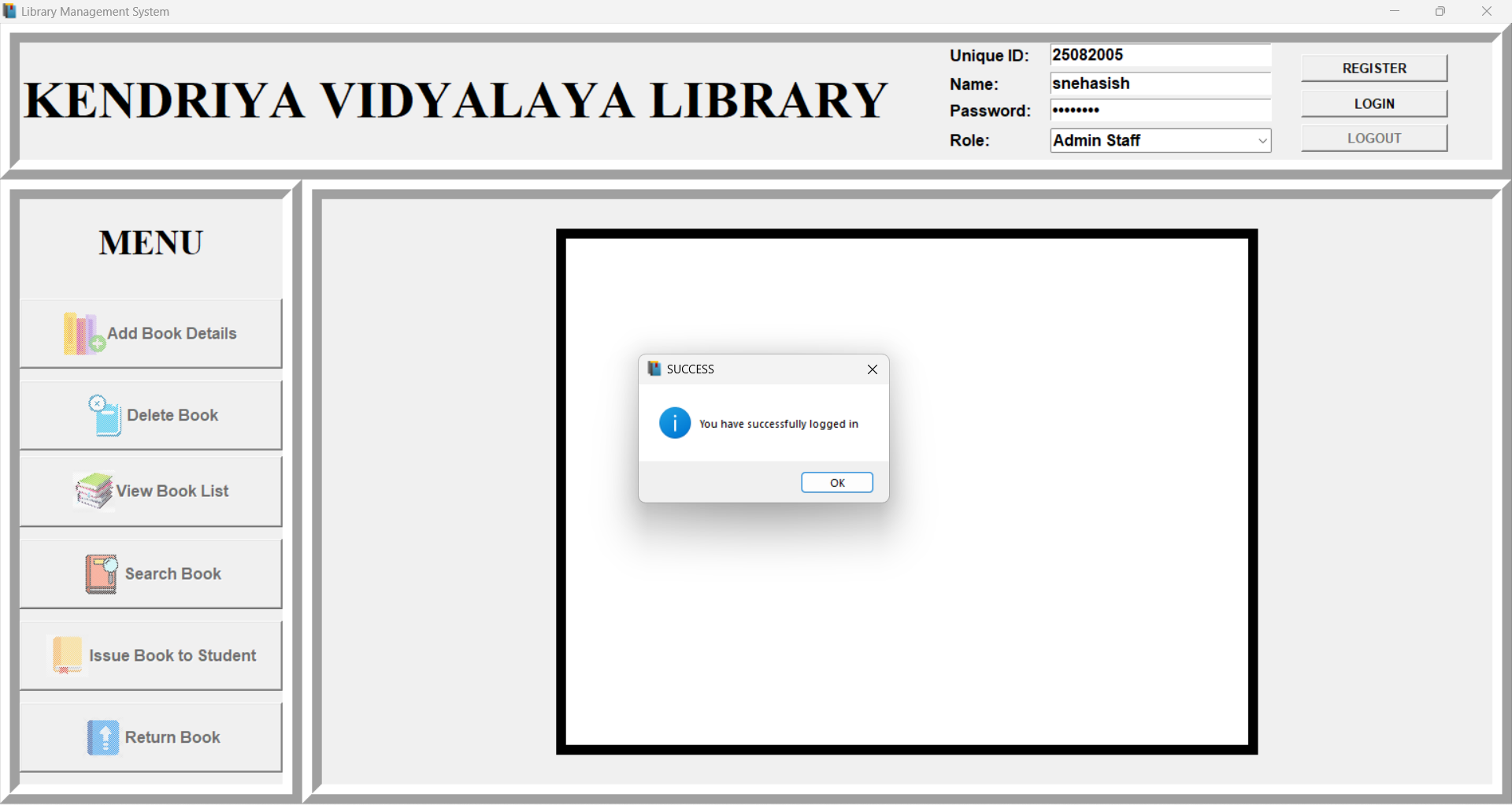


**CHAPTER 4**

**PROJECT TESTING**

**4.1 TEST CASES IN CUSTOMER DETAIL**

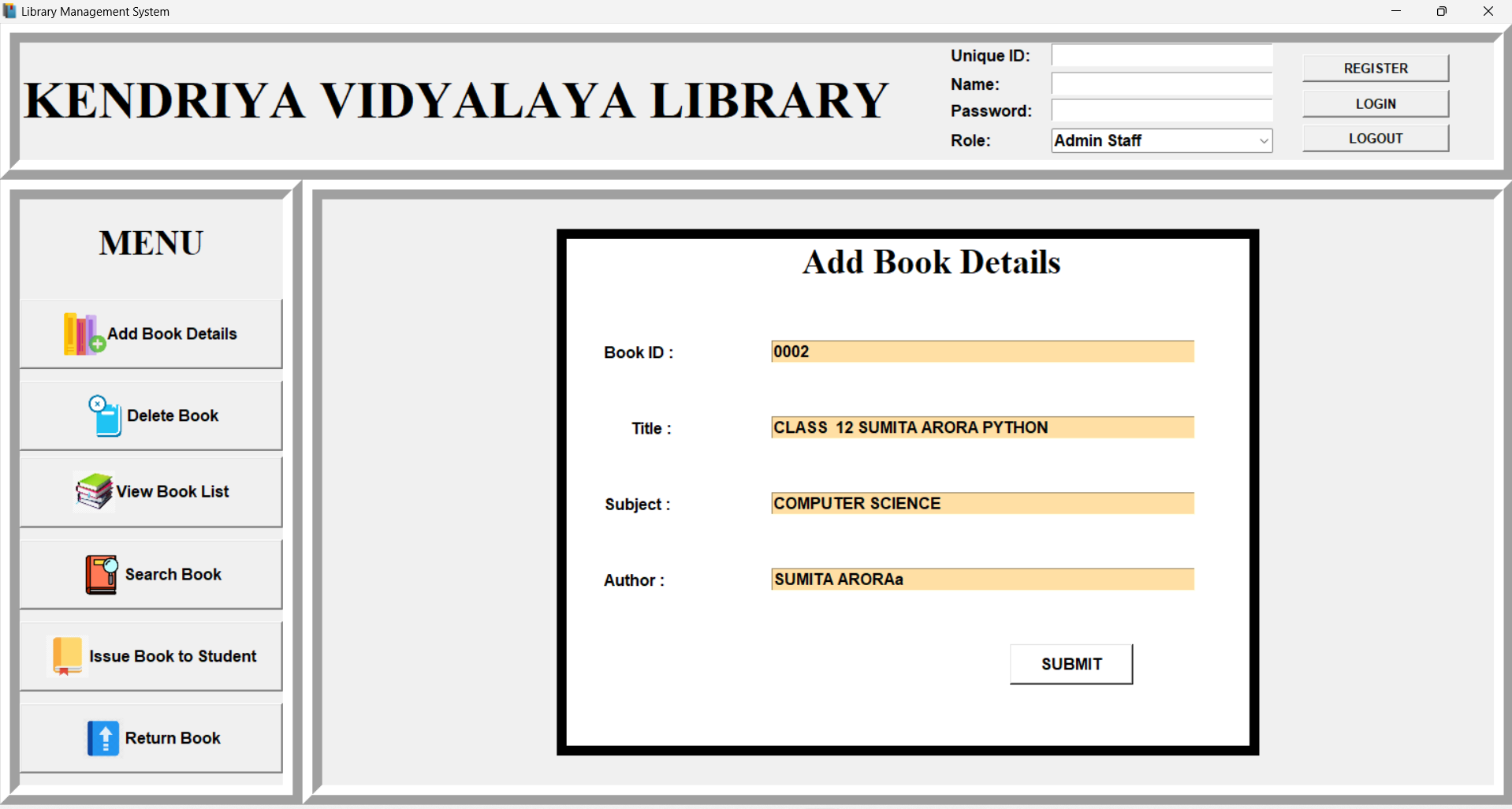
4.1.1 ADMIN ID LOGIN

****

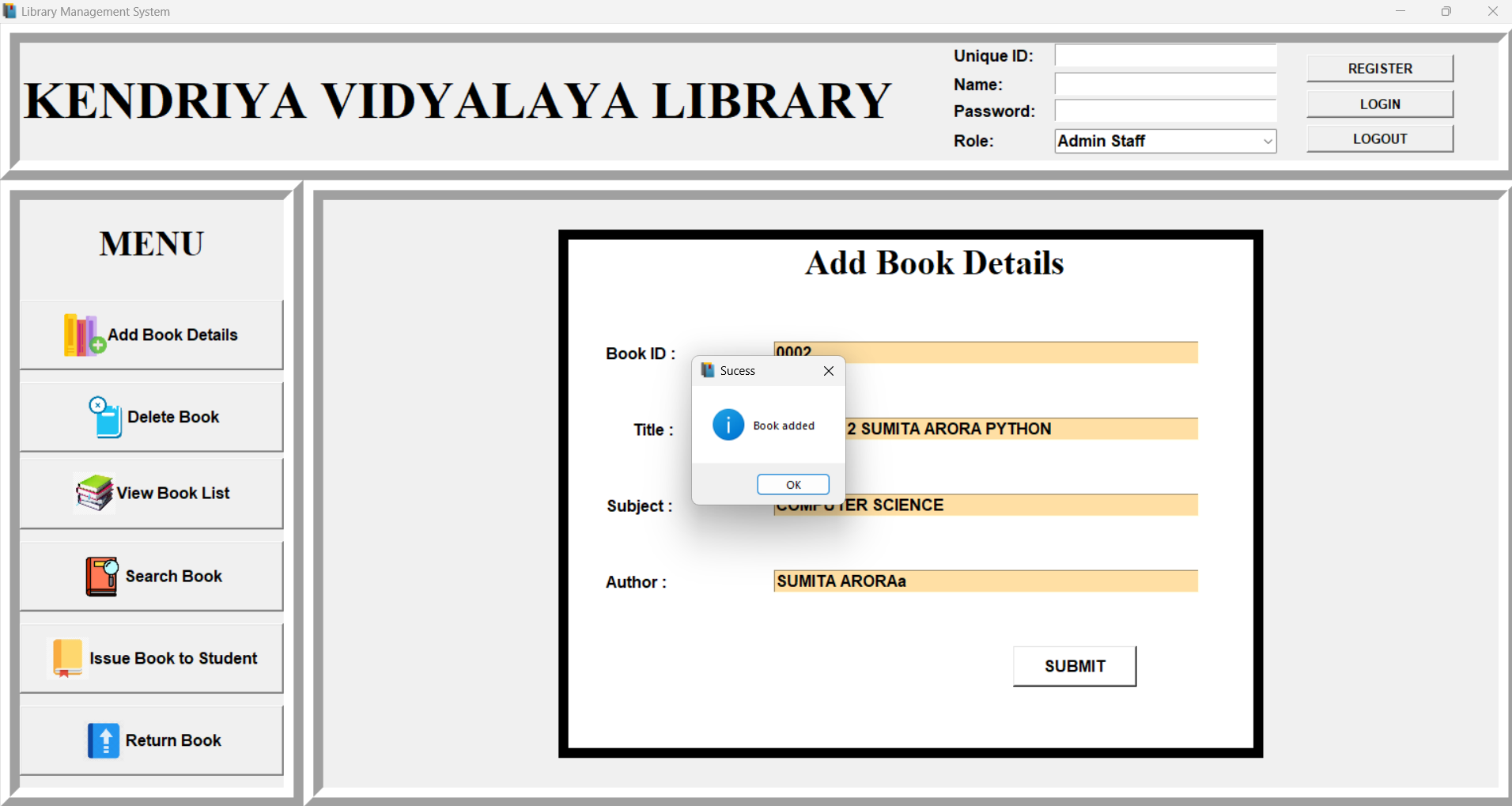
**Fig.4.1 Entering Login ID as Admin to access the menu,message displayed ‘You have successfully Logged in”**

**4.2 TEST CASES IN ADDING BOOK DETAILS**

4.2.1 ADDING BOOK RECORD

****

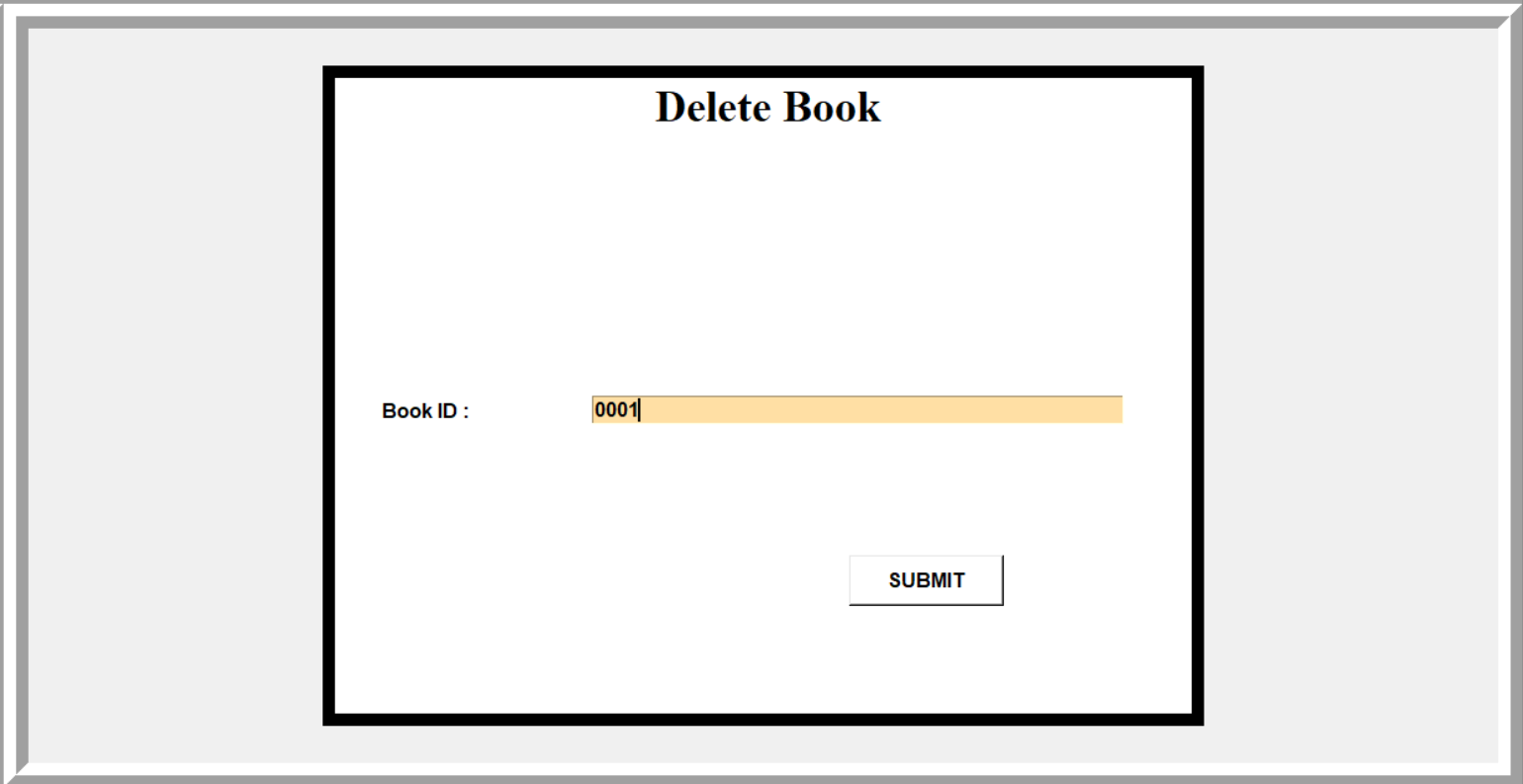
**Fig.4.2 Entering Data- Before Clicking the ‘SUBMIT’ Button**

****

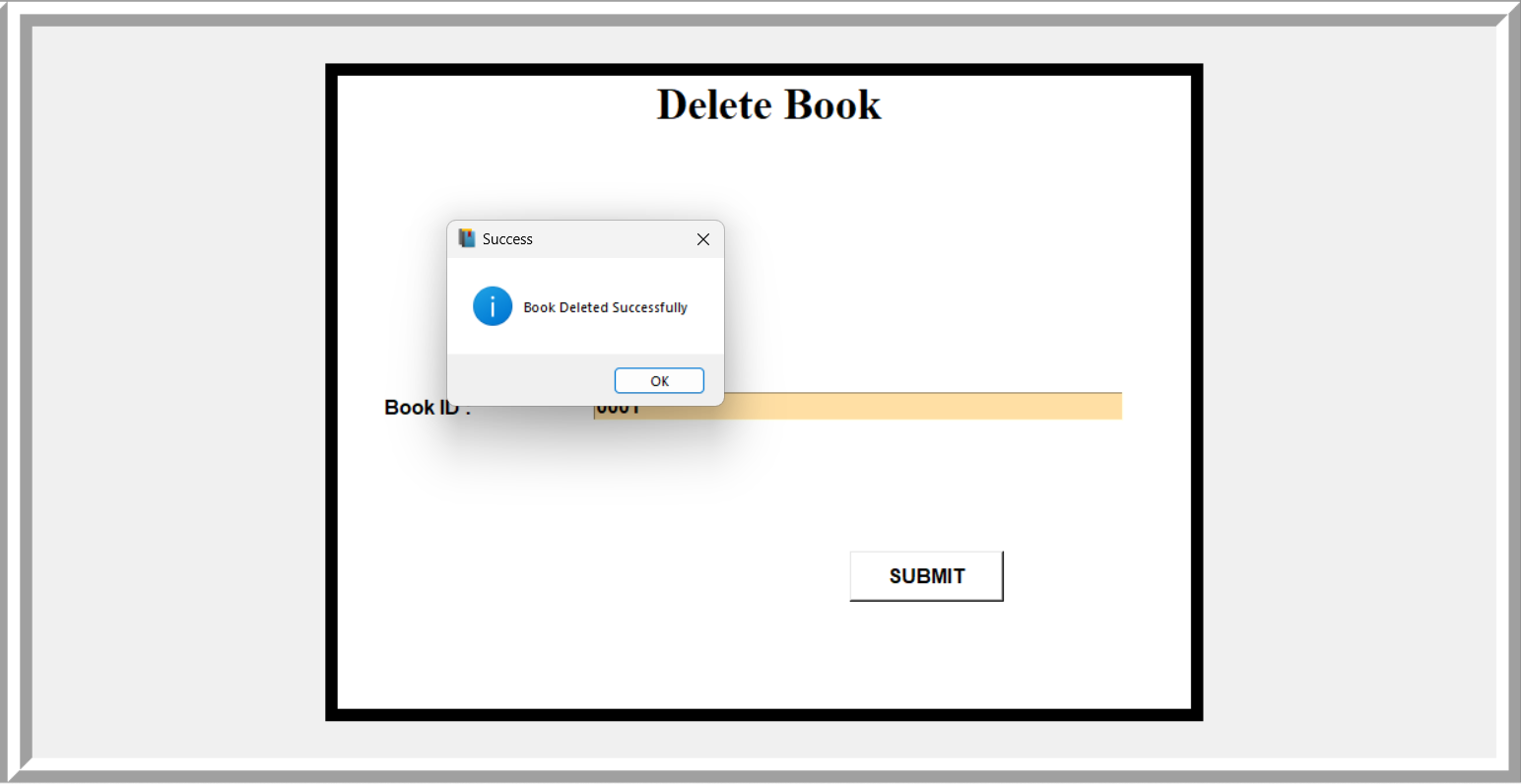
**Fig.4.3 Message Displayed After Clicking the ‘SUBMIT’ Button**

**4.3 TEST CASES IN DELETE BOOK DETAILS**

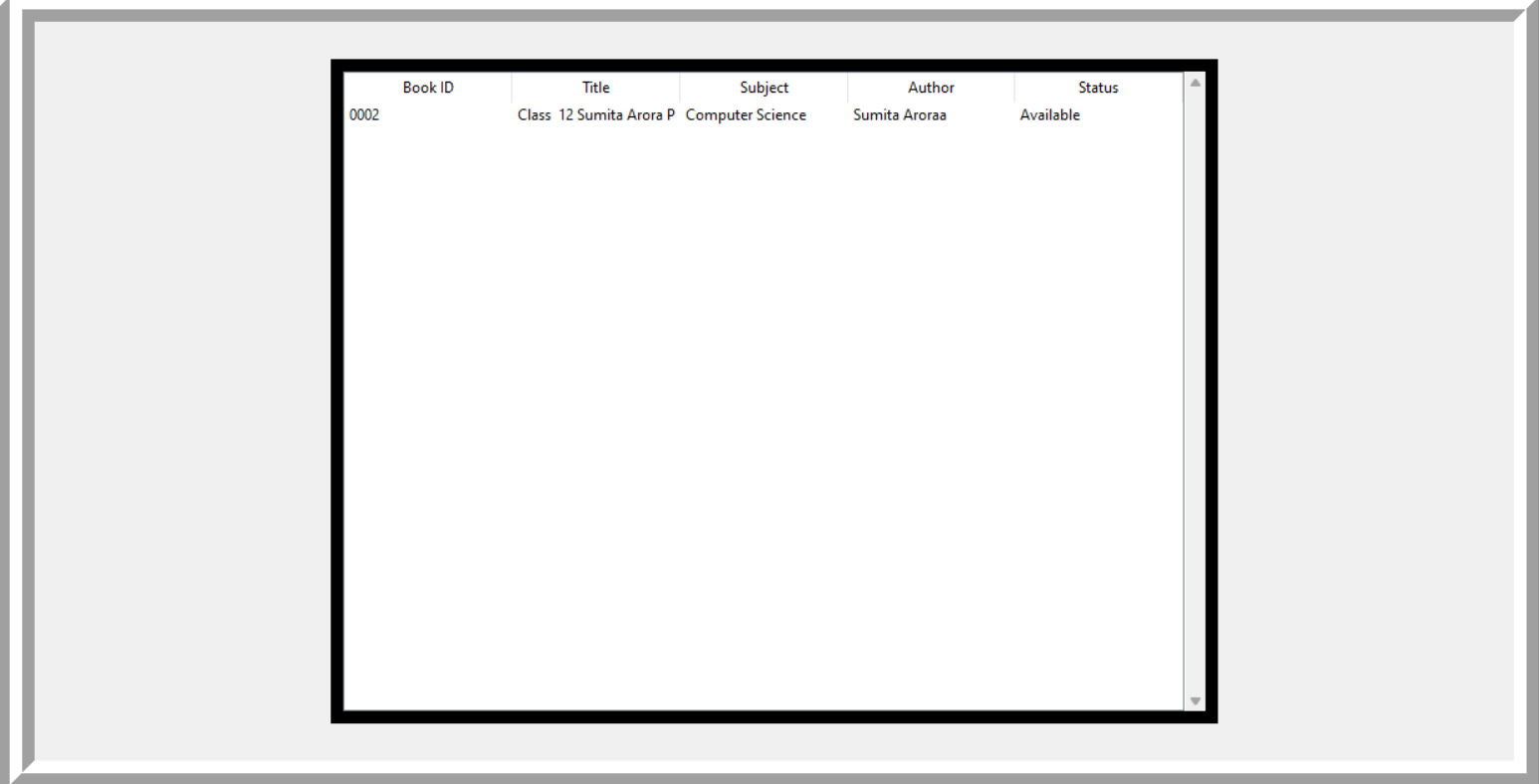
4.3.1 DELETING BOOK RECORD



**Fig.4.4 Entering Book ID Before Clicking the ‘SUBMIT’ Button**

****

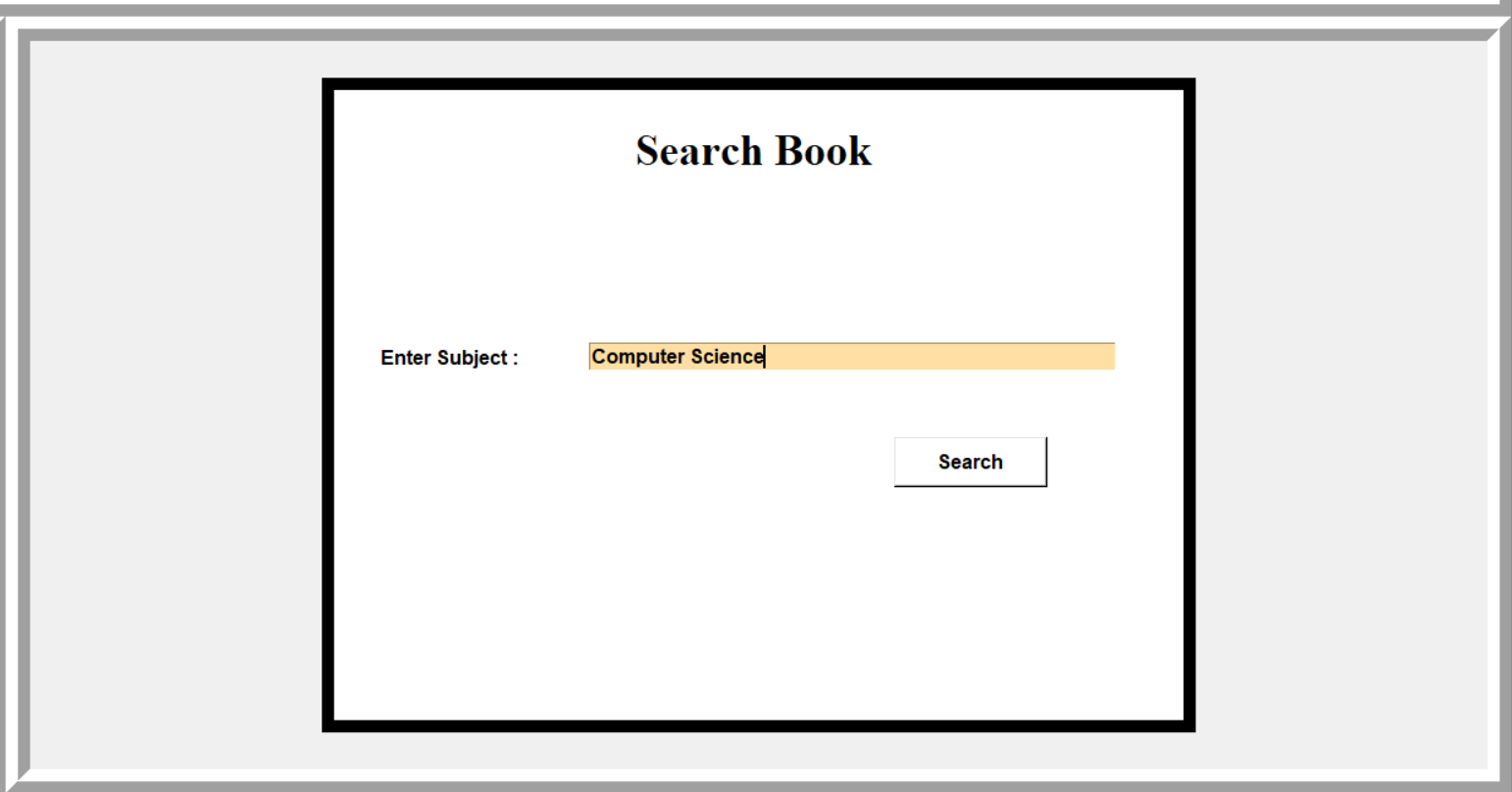
**Fig.4.3 Message Displayed After Clicking the ‘SUBMIT’ Button**

****

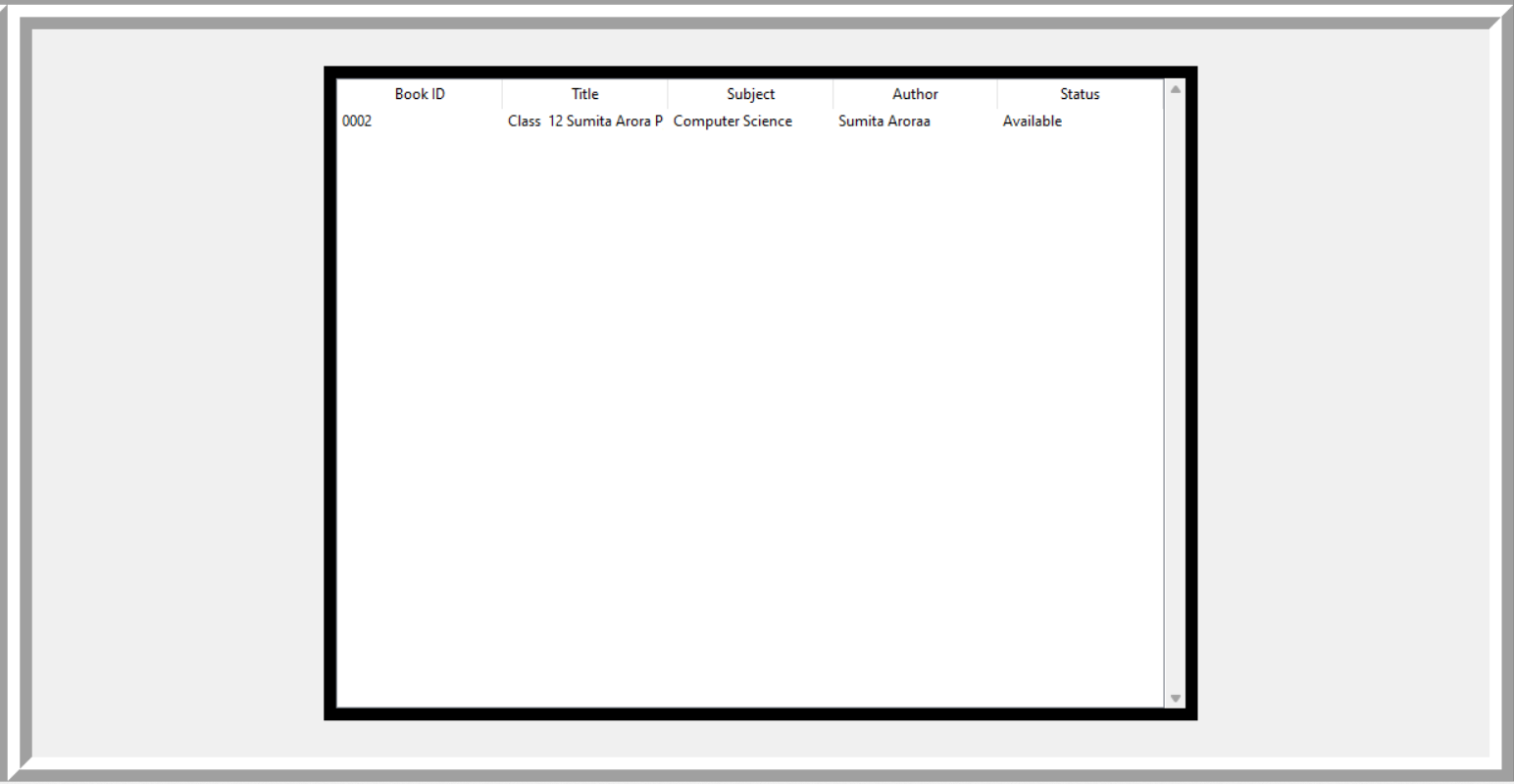
**Fig.4.4 View book list**

**4.4 TEST CASES IN SEARCH BOOK DETAILS**

**4.4.1** SEARCHING BOOK



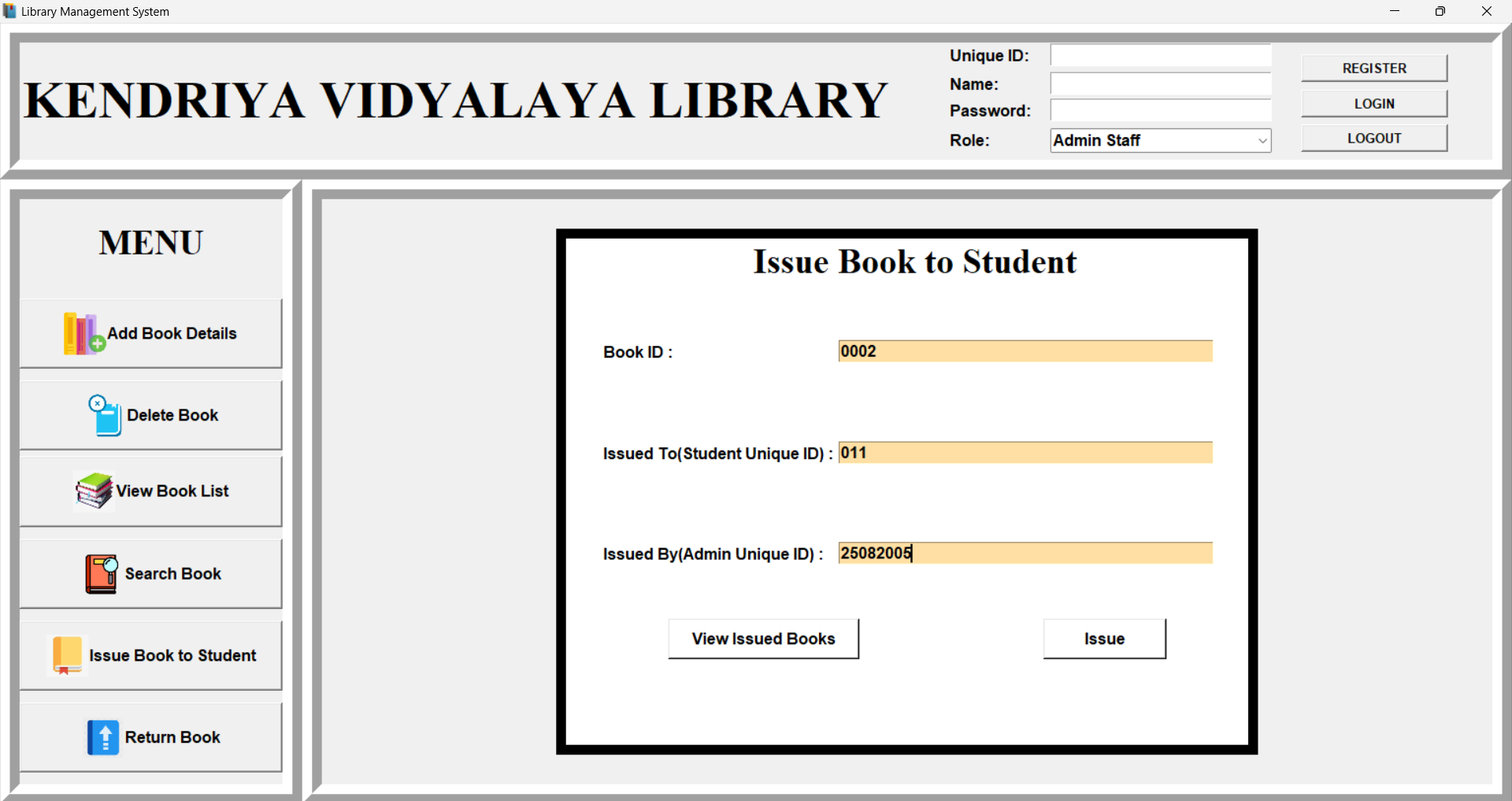
**Fig.4.5 Entering Subject name to show all the Books**



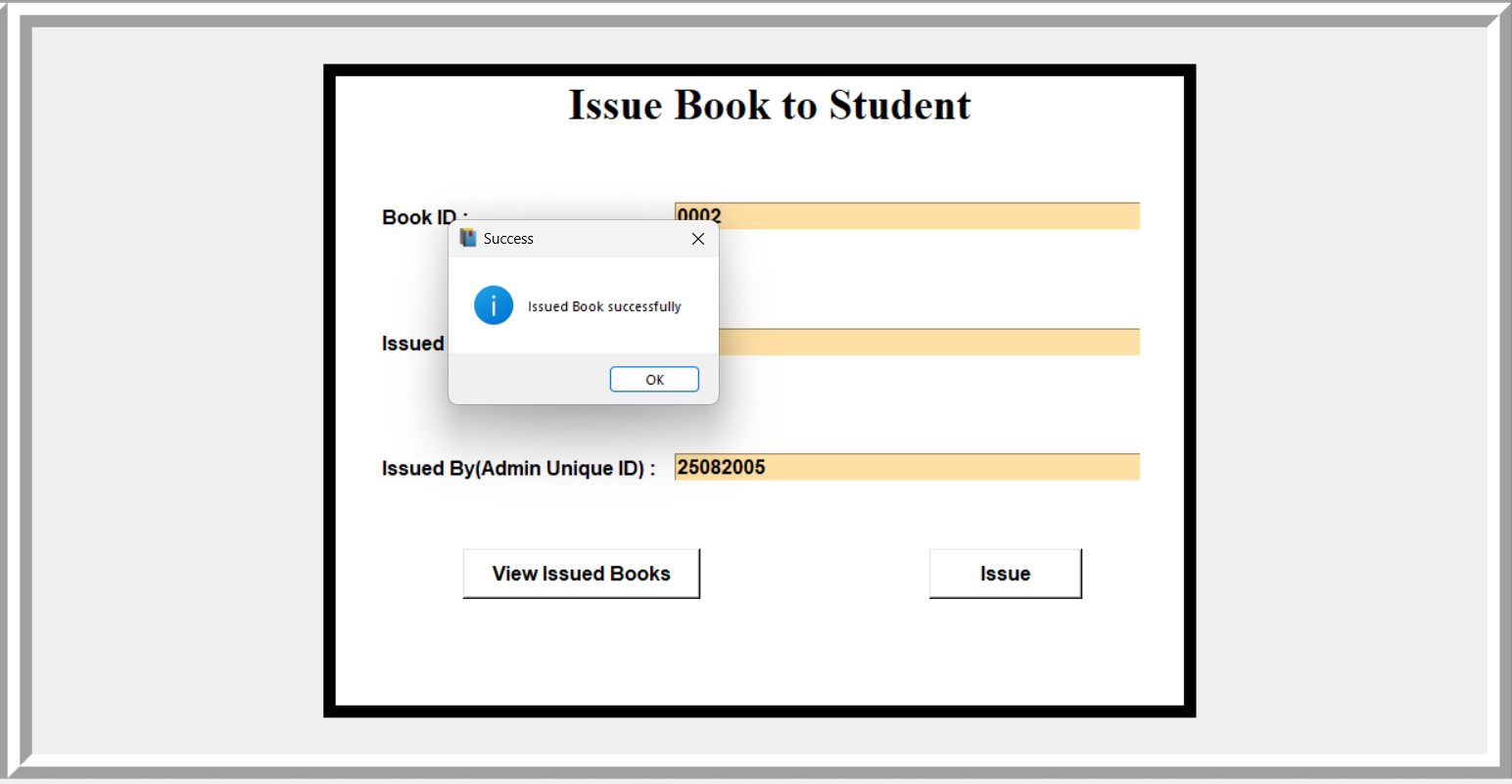
**Fig.4.6 Books Displayed After Clicking the ‘SUBMIT’ Button**

**4.5 TEST CASES IN ISSUING BOOKS TO STUDENTS**

**4.5.1** ISSUING BOOK



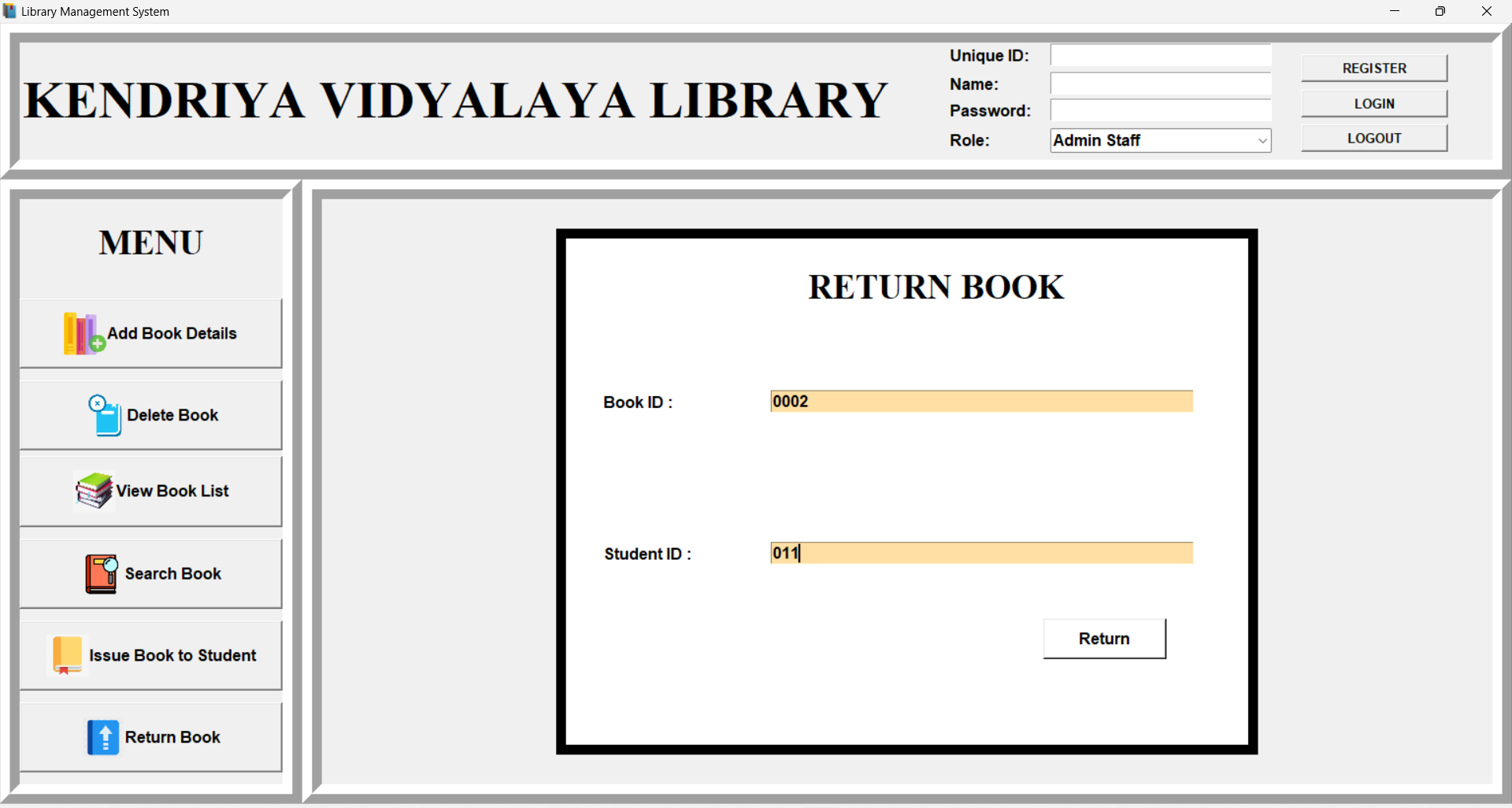
**Fig.4.7 Entering Required Details before issuing.**

****

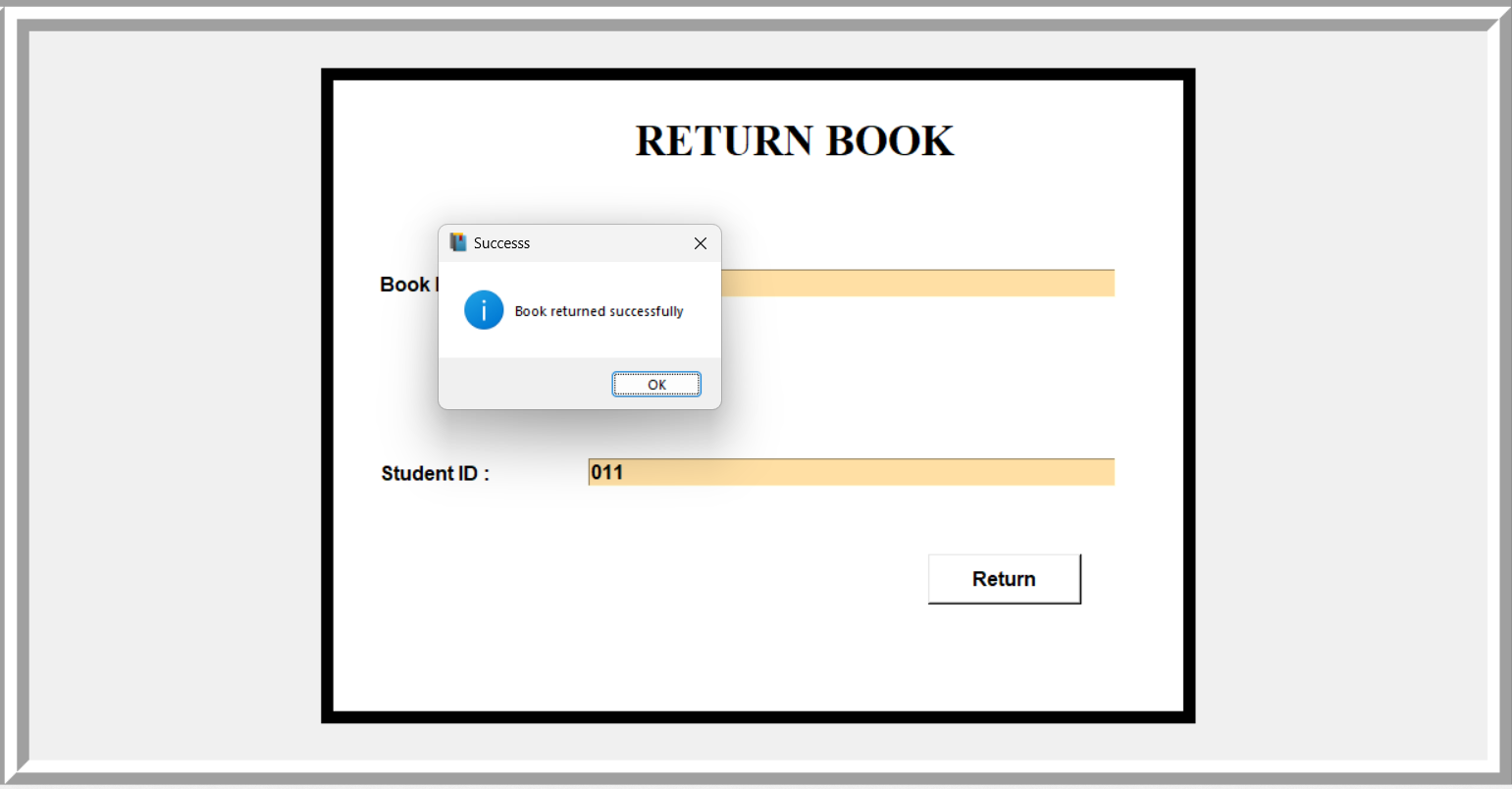
**Fig.4.6 Issued Book Displayed After Clicking the ‘ISSUE’ Button**

**4.6 TEST CASES IN RETURNING BOOKS TO STUDENTS**

**4.6.1** RETURNING BOOK



**Fig.4.8 Entering Required Details before Returning .**

****

**Fig.4.9 Book Returned Displayed After Clicking the ‘RETURN’ Button**

**CHAPTER 5**

**CONCLUSION & FUTURE ENHANCEMENT**

**5.1 CONCLUSION**

In conclusion, a library management system made using Python can provide a wide range of features and benefits for both library staff and patrons. Some of the key features that can be included in such a system are:

* Automation of routine tasks such as book checkouts and returns, which can save time and reduce errors.
* Search functionality that allows patrons to easily find books and other resources by keyword, author, or other criteria.
* Ability to generate reports and statistics on library usage, such as the most popular books or the busiest times of day.
* User-friendly interface that makes it easy for patrons to navigate and access the library's resources.
* Integration with other systems such as databases, which can enable the library to manage large amounts of data and provide more accurate information.

In addition to these features, a library management system made using Python can also provide significant benefits for library staff. These include:

* Improved efficiency and accuracy in managing the library's resources.
* Ability to access and analyze data in real-time, which can help staff make informed decisions about library operations.
* Increased productivity and reduced workload for staff, as many routine tasks are automated.
* Enhanced patron service, as staff can quickly and easily access information about library resources and assist patrons in finding what they need.

Overall, a library management system made using Python can be a powerful tool for improving the efficiency and effectiveness of any library. The system can provide significant benefits for both library staff and patrons, making it an excellent choice for managing a library's resources.

**5.2 FUTURE SCOPE**

The future scope for a library management system made using Python can be quite broad, as the technology continues to advance and new features and capabilities are added. Some potential areas of future development include:

* Integration with new technologies such as artificial intelligence and machine learning, which can enable the system to better understand and respond to user needs. For example, a library management system that uses AI to recommend books or resources based on a patron's reading history.
* Development of mobile and web-based interfaces that allow patrons to access the library's resources from anywhere, at any time.
* Increased use of data analytics to gain insights into library usage and performance, and make more informed decisions about library operations.
* Development of more advanced security features to protect sensitive user data and ensure the integrity of the library's resources.
* Integration with other systems such as e-commerce platforms, which can enable patrons to purchase or download digital versions of books and other resources directly from the library management system.

In addition to these areas of development Further I am going to work on code optimization and I am going to make our Library management System more efficient. As technology continues to advance, it will open up new possibilities for library management system.

**References**

[1] Computer Science with Python book, Sultan Chand & Sons(P) Ltd. Publication, Author: Preeti Arora, Class 12, CBSE Board, 2021 Edition

[2] Sanchaya Education Private Ltd., GeeksforGeeks Online Education Platform, Founder: Sandeep Jain, Website: www.geeksforgeeks.org

[3] Python Crash Course eBook, 2nd Edition, No Starch Press Publication, Author: Eric Matthes, Published on 21 May 2019

[4] Tutorials Point Online Learning Platform, Maintained by Tutorial Point India Limited, Founder: Mohtashim, Website: www.tutorialspoint.com

[5] Programming Python: Powerful Object-Oriented Programming, 6th Edition, O'Reilly Publication, Author: Mark Lutz, Published on 1 January 2018

[6] Online tutorials and guides