

LIBRARY

MANAGEMENT

SYSTEM

PROJECT REPORT
COMPUTER SCIENCE

NAME: ARYAN SINGH
BOARD ROLL NO.: 21663371
SCHOOL: DPS Indirapuram
SESSION: 2021 -22

CERTIFICATE

This is to certify that ARYAN SINGH of class XII-C has prepared the project on "LIBRARY MANAGAMENT SYSTEM". The project is result of his efforts and endeavours. This project is found worthy of acceptance as the final project report for the subject computer science of class XII.

He has prepared this project under my guidance.

Ms. Rinkoo Gupta
(Computer Science Teacher)
(DPS Indirapuram)

ACKNOWLEDGEMENT

I would like to express a deep sense of gratitude towards my computer science teacher Ms. Rinkoo Gupta ma'am for guiding me through the course of my project. She always evinced keen interest in my work and her constructive advice and constant motivation have been responsible for the successful completion of this project.

My sincere thanks goes to Ms. Sangeeta Hajela, our school principal for her coordination in extending every possible support possible in the success of this project.

I would like to thank all those who have helped directly or indirectly in the completion of the project.

Aryan Singh
XII -C

INDEX

S. No	Topic	Remarks
1	INTRODUCTION TO THE PROJECT	
2	MySQL TABLES USED AND THEIR STRUCTURES	
3	HARDWARE AND SOFTWARE REQUIREMENTS	
4	CODING	
5	OUTPUTS	
6	INPUT VALIDATION, RESTRICTIONS, AND ERROR HANDLING	
7	CONCLUSION	
8	FUTURE ENHANCEMENTS	
9	BIBLIOGRAPHY	

INTRODUCTION TO THE PROJECT

The project's aim is to create an interactive library application called "LIBRARY MANAGEMENT SYSTEM".

ABOUT THE GAME:

A user-menu pops up on starting the program, from where the user can access 2 modes: 'ADMIN' and 'STUDENT'. We used Python's MySQL Connector to establish a connection between the library application and the SQL database. The database contains tables that store the book, staff, and student details. It also securely stores the usernames and passwords of the staff.

After entering the Student mode, the user is directed to a page where they can either Issue a book, Return a book or View the book details.

1. **View Book List** - The user can view the genres, author, book codes, and the issued status of all the books currently available in the library.
2. **Issue Book** - The user can issue a book by entering their Admission Number and the Book Code (Book codes can be viewed through the View Button).
3. **Return Book** - The user can return their issued book by entering their Admission Number.

When the Admin button is clicked, a login window pops up where the staff member needs to enter the correct username and password to enter the Admin mode. If they want to change the password, they can do so by entering their old password followed by their new password.

1. **View Book Records** - The Admin can see the Date of Issue of all the books and the Date of Return if it has been returned.
2. **Book Details** - The Admin has the option to Add, Delete, and View Book Details.
3. **Student Details** - The Admin has the option to Add, Delete, and View Student Details.
4. **Staff Details** - The Admin has the option to Add, Delete, and View Staff Details.

WORKING DESCRIPTION:

The build-up of the user-menu section is as follows:

1. Student
 - 1.1. Issue Book
 - 1.2. Return Book
 - 1.3. View Book Details
 - 1.4. Go Back
2. Admin
 - 2.1. Login
 - 2.2. Book Details
 - 2.2.1. Add Book Details
 - 2.2.2. Delete Book Details
 - 2.2.3. View Book Details

- 2.2.4. Go Back
- 2.3. View Book Records
- 2.4. Student Details
 - 2.4.1. Add Student Details
 - 2.4.2. Delete Student Details
 - 2.4.3. View Student Details
 - 2.4.4. Go Back
- 2.5. Staff Details
 - 2.5.1. Add Staff Details
 - 2.5.2. Delete Staff Details
 - 2.5.3. View Staff Details
 - 2.5.4. Go Back
- 2.6. Go Back
- 3. Exit

All the points stated above have been made to incorporate the basic functions of SQL.

The program consists of separate files of code. These are:

- 1. Main_Window
- 2. Student
- 3. Issue
- 4. Return
- 5. Booklist
- 6. Admin
- 7. Password_Admin
- 8. Book_Records
- 9. Details_Book
- 10. Add_Book

11. Delete_Book
12. View_Book
13. Details_Student
14. Add_Student
15. Delete_Student
16. View_Student
17. Details_Staff
18. Add_Staff
19. Delete_Staff
20. View_Staff

MYSQL TABLES USED AND THEIR STRUCTURES

- Tables in database:

```
mysql> show tables;
+-----+
| Tables_in_cs_project |
+-----+
| book_records
| books
| password
| staff
| student
+-----+
5 rows in set (0.08 sec)
```

- Student Table:

```
mysql> select * from student;
+-----+-----+-----+-----+-----+
| name | Date_Of_Birth | admission_number | Class | Section |
+-----+-----+-----+-----+-----+
| Kanak Joshi | 2004-06-17 | I-08/5068 | 12 | A |
| Rohan Yadav | 2004-08-02 | I-09/5023 | 12 | A |
| Nivedita Sharma | 2005-04-19 | I-09/5787 | 11 | H |
| Shreya Jain | 2004-01-09 | I-09/5789 | 12 | E |
| Poulami Bhattacharjee | 2005-11-19 | I-09/5790 | 11 | B |
| Anisha Singh | 2005-10-29 | I-09/7249 | 11 | A |
| Sam Singh | 2006-06-12 | I-10/7866 | 10 | A |
| Vishal Jaiswal | 2005-04-11 | I-10/7896 | 11 | A |
| Salman Kapoor | 2003-07-27 | I-11/8906 | 12 | D |
| Aditya Kumar | 2004-05-23 | I-11/8976 | 12 | G |
| Anuj Singh | 2007-11-16 | I-12/8790 | 09 | C |
| Ramesh Tripathi | 2005-04-17 | I-13/9761 | 11 | I |
| Joel Joseph | 2005-12-11 | I-13/9799 | 11 | C |
| Jitendra Yadav | 2005-01-10 | I-13/9888 | 11 | D |
| Avi Gupta | 2004-02-05 | I-14/10100 | 12 | H |
| Ridhi Singh | 2004-12-20 | I-14/10111 | 12 | E |
| Aastha Bhat | 2004-12-22 | I-14/10122 | 12 | K |
| Sheena Bajaj | 2005-01-09 | I-14/10444 | 11 | A |
| Kritika Singh | 2004-11-09 | I-14/10467 | 12 | J |
| Arushi Kapkoti | 2004-07-27 | I-14/10554 | 12 | E |
| Simran Kaur | 2004-12-18 | I-14/10776 | 12 | C |
| Sandeep Kumar | 2006-09-12 | I-15/10281 | 10 | G |
| Rahul Modi | 2008-07-08 | I-15/10561 | 08 | C |
| Pankaj Sharma | 2008-09-30 | I-15/10781 | 11 | F |
| Aditya Kamal | 2004-03-23 | I-15/8889 | 12 | D |
| Shreya Chauhan | 2006-01-09 | I-15/9111 | 10 | E |
| Jasmine Fernandez | 2006-01-13 | I-15/9123 | 10 | C |
| Armaan Kohli | 2005-01-18 | I-15/9770 | 11 | F |
| Jayant Yadav | 2005-01-10 | I-15/9777 | 11 | E |
| Bhavya Jha | 2005-09-13 | I-15/9778 | 12 | A |
| Tony Kakkar | 2005-01-15 | I-15/9779 | 12 | G |
| Bhavdeep Singh | 2004-03-14 | I-15/9999 | 12 | F |
| Lakhshya Singh | 2004-07-20 | I-16/10059 | 12 | D |
| Aladdin Singh | 2004-07-23 | I-16/10089 | 12 | E |
| Raveena Kaur | 2004-10-20 | I-18/12229 | 12 | G |
| Ajey Nagar | 2004-01-10 | I-18/12620 | 12 | E |
| Ajit Singh | 2005-06-18 | I-18/12689 | 11 | I |
| Rashmi Pillai | 2009-02-28 | I-19/16789 | 07 | G |
| Mahesh Yadav | 2008-08-17 | I-19/16879 | 08 | A |
| Mona Kaur | 2009-12-16 | I-19/16889 | 07 | F |
+-----+
40 rows in set (0.13 sec)
```

• Books Table:

name_of_book	author	Book_Code	Genre	Status
A Beautiful Mind	Sylvia Nasar	BI0001	Biographies	Issued
The Lost City of Z	David Grann	BI0002	Biographies	Not Issued
The Minds of Billy Milligan	Daniel Keyes	BI0003	Biographies	Not Issued
Napolean: A Life	Andrwe Roberts	BI0004	Biographies	Not Issued
The Passage of Power	Robert Caro	BI0005	Biographies	Not Issued
Wings of Fire	Arun Tiwari	BI0006	Biographies	Not Issued
The Man who knew Infinity	Robert Kanigel	BI0007	Biographies	Not Issued
Vivekananda: A biography	Swami Nikhilananda	BI0008	Biographies	Not Issued
Eleven Gods and a Billion Indians	Boria Majumdar	BI0009	Biographies	Not Issued
I am Neil Armstrong	Brad Meltzer	BI0010	Biographies	Not Issued
Invaluable	Maya Grossman	EDU001	Educational	Not Issued
The Privileged Poor	Anthony Abraham Jack	EDU002	Educational	Not Issued
Thrive Through the Five	Dr Jill M Siler	EDU003	Educational	Not Issued
Origin of the Species	Charles Darwin	EDU004	Educational	Issued
A Brief History of Time	Stephen Hawking	EDU005	Educational	Not Issued
Cosmos	Carl Sagan	EDU006	Educational	Not Issued
A Short History of Nearly Everything	Bill Bryson	EDU007	Educational	Not Issued
Dark Matter	Blake Crouch	EDU008	Educational	Not Issued
The Making of the Atomic Bomb	Richard Rhodes	EDU009	Educational	Not Issued
The Structure of Scientific Revolutions	Thomas Kuhn	EDU010	Educational	Not Issued
The Adventure of Augie March	Saul Bellow	FIC001	Fiction	Not Issued
American Pastoral	Philip Roth	FIC002	Fiction	Not Issued
The Assistant	Bernard Malamud	FIC003	Fiction	Issued
Charlie and the Chocolate Factory	Roald Dahl	FIC004	Fiction	Not Issued
Nineteen Eighty-Four	George Orwell	FIC005	Fiction	Not Issued
The Lord of the Rings	JRR Tolkien	FIC006	Fiction	Issued
Enders Game	Orson Scott Card	FIC007	Fiction	Not Issued
The Dune Chronicles	Frank Herbert	FIC008	Fiction	Issued
The Kingkiller Chronicles	Patrick Rothfuss	FIC009	Fiction	Not Issued
Harry Potter and the Deathly Hallows	JK Rowling	FIC010	Fiction	Not Issued
The Bell Jar	Sylvia Plath	NOV001	Novel	Not Issued
The Great Gatsby	F Scott Fitzgerald	NOV002	Novel	Not Issued
Speak	Laurie Halse Anderson	NOV003	Novel	Not Issued
A Suitable Boy	Vikram Seth	NOV004	Novel	Not Issued
The Inheritance of Loss	Kiran Desai	NOV005	Novel	Not Issued
The White Tiger	Aravind Adiga	NOV006	Novel	Not Issued
A Fine Balance	Rohinton Mistry	NOV007	Novel	Not Issued
To Kill a Mockingbird	Harper Lee	NOV008	Novel	Not Issued
Ullyses	James Joyce	NOV009	Novel	Not Issued
Beloved	Toni Morrison	NOV010	Novel	Not Issued

40 rows in set (0.00 sec)

• Staff Table:

name	StaffId	Gender	address	phone_number
Akansha Singh	1	F	Z-420, National Colony	9384759708
Mridula Tiwari	2	F	F-397, Reserved Colony	9283748576
Suraj Tyagi	3	M	D-410, Gundanagar	9946529765
Lalu Prasad	4	M	E-4102, Shiva Colony	9284859673
Bir Gupta Anand	5	M	M-387, Shipra Rivera	9346847875

5 rows in set (0.12 sec)

- Book Records Table:

```
mysql> select * from book_records;
+-----+-----+-----+-----+-----+-----+-----+-----+
| name | admission_number | book_name | Book_Code | Date_of_issue | Date_of_return | Due_date | issue_time |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Kanak Joshi | I-08/5068 | A Beautiful Mind | BI0001 | 2021-11-06 | NULL | 2021-11-20 | 22:25:57 |
| Avi Gupta | I-14/10100 | The Minds of Billy Milligan | BI0003 | 2021-11-06 | 2021-11-06 | 2021-11-20 | 22:18:59 |
| Ridhi Singh | I-14/10111 | The Lord of the Rings | FIC006 | 2021-11-06 | NULL | 2021-11-20 | 22:23:35 |
| Arushi Kapkoti | I-14/10554 | Origin of the Species | EDU004 | 2021-11-06 | NULL | 2021-11-20 | 22:19:53 |
| Simran Kaur | I-14/10776 | The Assistant | FIC003 | 2021-11-06 | NULL | 2021-11-20 | 22:24:42 |
| Aditya Kamal | I-15/8889 | The Dune Chronicles | FIC008 | 2021-11-06 | NULL | 2021-11-20 | 22:23:00 |
| Aladdin Singh | I-16/10089 | The Bell Jar | NOV001 | 2021-11-06 | 2021-11-06 | 2021-11-20 | 22:21:04 |
+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

- Username-Password Table:

```
mysql> select * from password;
+-----+-----+
| user_name | password |
+-----+-----+
| 1 | wefrf |
| 2 | rf3ref |
| 3 | hello |
| 4 | hi |
| 5 | pass |
+-----+-----+
5 rows in set (0.00 sec)
```

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE -

- **Minimum:**
 - Intel Atom® processor or Intel® Core™ i3 processor
- **Recommended:**
 - Intel® Core™ i5 processor

SOFTWARE -

- **Minimum**-Windows 7 or later, macOS, and Linux
- **Recommended**-Windows 10, macOS, and Linux

The following Python libraries must be installed beforehand:

- Python 3
- Python pip
- Python Tkinter
- MySQL 8.0 Command Line Client
- MySQL Server 8
- MySQL Connector/Python 8

CODING

#Main Window:

```

from tkinter import *
from Student import *
from Password_Admin import *

window = Tk()

window.title("Library Management System")

window.state('zoomed')

window.geometry("1080x720+200+35")

#BACKGROUND IMAGE

bg = PhotoImage(file = "1.png")

img = Label(window, image = bg,borderwidth=0)

img.place(x = 0,y = 0)

#LABELS

lbl = Label(window, bg='#FFEBBC',fg='#7c5b07',text = "WELCOME TO TTM LIBRARY",
font = ("Comic Sans MS bold",34),padx=5,pady=5,borderwidth=10,relief='ridge')

lbl.pack(pady = 80)

from tkinter.font import Font

font = Font(family = "Comic Sans MS", size = 14)

window.option_add("*TCombobox*Listbox*Font", font)

#BUTTONS

btn1 = Button(window, text = "STUDENT", padx = 75, pady = 20,font=('Comic Sans MS',22),borderwidth=5,bg='#DEB887',fg='#7c5b07',command=student)

btn1.pack(pady=10)

btn2 = Button(window, text = "ADMIN", padx = 92, pady = 20,font=('Comic Sans MS',22),borderwidth=5,bg='#DEB887',fg='#7c5b07',command=password)

btn2.pack(pady=10)

btn3 = Button(window, text = "EXIT", padx = 107, pady = 20,font=('Comic Sans MS',22),borderwidth=5,bg='#DEB887',fg='#7c5b07',command=window.destroy)

btn3.pack(pady=10)

window.mainloop()

```

#Student:

```

from tkinter import *
from Issue import *
from Booklist import *
from Return import *

def student():

    root=Toplevel()
    root.title('Student')
    root.geometry('1920x1080')
    root.state('zoomed')

    #BACKGROUND IMAGE
    bg = PhotoImage(file = "snow.png")
    lbl1 = Label(root, image = bg,borderwidth=0)
    lbl1.place(x = 0,y = 0)

    #LABELS
    lbl = Label(root, bg='#4cbb17',fg='White',text = "STUDENT", font = ("Comic Sans MS bold",35),padx=20,pady=5,borderwidth=10,relief='ridge')
    lbl.pack(pady = 40)

    #BUTTONS
    btn1 = Button(root, text = "ISSUE BOOK", padx = 80, pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9ACD32',fg='white',command=issue)
    btn1.pack(pady=10)

    btn2 = Button(root, text = "RETURN BOOK", padx = 68, pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9ACD32',fg='white',command=return_book)
    btn2.pack(pady=10)

    btn3 = Button(root, text = "VIEW BOOK LIST", padx = 48, pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9ACD32',fg='white',command=book_list)
    btn3.pack(pady=10)

    btn4 = Button(root,text='GO BACK', padx = 109, pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9ACD32',fg='white',command=root.destroy)
    btn4.pack(pady=10)

    root.mainloop()

```

#Issue:

```

from tkinter import *
from datetime import *
import mysql.connector as con
import time
from tkinter import messagebox

def issue():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor(buffered=True)

r1=Toplevel()

r1.title('Issue')

r1.state('zoomed')

bg = PhotoImage(file = "newsunset.png")

img = Label(r1, image = bg,borderwidth=0)

img.place(x = 0,y = 0)

#LABELS

lb1=Label(r1,text='ISSUE BOOK',font = ("Comic Sans MS bold",40),
bg='#C71585',fg='White',padx=20,pady=10,borderwidth=3,relief='solid')

lb1.place(x=550,y=40)

lb2=Label(r1, bg='#FF00FF',fg='White',text = "ENTER BOOK CODE:", font =
("Comic Sans MS bold",20),padx=42,pady=5,width=20)

lb2.place(x=280,y=250)

lb3=Label(r1, bg='#FF00FF',fg='White',text = "ENTER ADMISSION NUMBER:", font
= ("Comic Sans MS bold",20),pady=5)

lb3.place(x=280,y=400)

#TEXT BOX

t1=Entry(r1,width=26,font = ("Comic Sans MS
bold",20),highlightcolor='blue',highlightthickness=4)

t1.place(y=250,x=760)

t2=Entry(r1,width=26,font = ("Comic Sans MS
bold",20),highlightcolor='blue',highlightthickness=4)

t2.place(y=400,x=760)

#FUNCTIONS

```

```
def issue_button():

    book_code=t1.get().strip()

    admn=t2.get().strip()

    if book_code=='' and admn=='':

        messagebox.showerror("ERROR","BOOK CODE AND ADMISSION NUMBER CANNOT BE EMPTY",parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

    elif book_code=='':

        messagebox.showerror("ERROR","BOOK CODE CANNOT BE EMPTY",parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

    elif admn=='':

        messagebox.showerror("ERROR","ADMISSION NUMBER CANNOT BE EMPTY",parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

    else:

        cur.execute('select Book_Code from books')

        flag1=False

        for i in cur.fetchall():

            if i[0]==book_code:

                flag1=True

                break

        cur.execute('select admission_number from student')

        flag2=False

        for i in cur.fetchall():

            if i[0]==admn:

                flag2=True

                break

        if flag1==False and flag2==False:
```

```

        messagebox.showerror("ERROR", "BOOK CODE AND ADMISSION NUMBER NOT
FOUND", parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

    elif flag1==False:

        messagebox.showerror("ERROR", "BOOK CODE NOT FOUND", parent=r1)

        t1.delete(0,END)

    elif flag2==False:

        messagebox.showerror("ERROR", "ADMISSION NUMBER NOT
FOUND", parent=r1)

        t2.delete(0,END)

    else:

        cur.execute('select date_of_return from book_records where
admission_number="{}"'.format(admn))

        l=cur.fetchone()

        if l!=None and l[0]==None:

            messagebox.showerror("ERROR", "YOU ALREADY HAVE ISSUED A
BOOK", parent=r1)

            t1.delete(0,END)

            t2.delete(0,END)

        else:

            cur.execute('select status from books where
book_code="{}"'.format(book_code))

            l1=cur.fetchone()

            if l1[0]=='Issued':

                messagebox.showerror("ERROR", "BOOK HAS ALREADY BEEN
ISSUED", parent=r1)

                t1.delete(0,END)

            else:

                cur.execute('select name from student where
admission_number="{}"'.format(admn))

                student_name=cur.fetchone()[0]

                cur.execute('select Name_of_book from books where
book_code="{}"'.format(book_code))

```

```

book_name=cur.fetchone()[0]

date_of_issue=date.today().strftime("%Y/%m/%d")

due_date=(date.today()+timedelta(days=14)).strftime('%Y/%m/%d')

issuetime=time.strftime('%H:%M:%S')

cur.execute('insert into
book_records(Name,Admission_number,book_name,book_code,date_of_issue,due_date,is
sue_time)
values ("{}","{}","{}","{}","{}","{}","{}")'.format(student_name,admn,book_name,b
ook_code,date_of_issue,due_date,issuetime))

cur.execute('update books set status="Issued" where
Book_code="{}"'.format(book_code))

mycon.commit()

due_date=datetime.strptime(due_date,'%Y/%m/%d')

messagebox.showinfo("SUCCESS", "BOOK ISSUED
SUCCESSFULLY\n\nDUE DATE OF RETURN IS
{}.".format(due_date.strftime('%d/%m/%Y')),parent=r1)

t1.delete(0,END)

t2.delete(0,END)

#BUTTONS

btn1=Button(r1,text='ISSUE',font = ("Comic Sans MS
bold",20),width=11,pady=5,borderwidth=3,
bg ='#C71585',fg='White',command=issue_button)

btn1.place(y=550,x=420)

btn2=Button(r1,text='GO BACK',font = ("Comic Sans MS
bold",20),width=11,pady=5,borderwidth=3,
bg ='#C71585',fg='White',command=r1.destroy)

btn2.place(y=550,x=840)

r1.mainloop()

```

#Return:

```

from tkinter import *
from datetime import *
import mysql.connector as con
from tkinter import messagebox

def return_book():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_proje
t')

```

```
cur=mycon.cursor()

r1=Toplevel()

r1.title('RETURN')

r1.state('zoomed')

#Background Image

bg = PhotoImage(file = "field.png")

img = Label(r1, image = bg,borderwidth=0)

img.place(x = 0,y = 0)

#LABELS

lb1=Label(r1,text='RETURN BOOK',font = ("Comic Sans MS bold",40),padx=20,pady=10,borderwidth=3,relief='solid',bg='orange',fg='black')

lb1.place(x=540,y=40)

lb2=Label(r1,fg='black',text = "ENTER ADMISSION NUMBER:", font = ("Comic Sans MS bold",20),width=25,pady=5)

lb2.place(x=270,y=330)

#TEXT BOX

t1=Entry(r1,width=25,font = ("Comic Sans MS bold",20),highlightcolor='blue',highlightthickness=4)

t1.place(x=820,y=330)

#FUNCTIONS

def return_button():

    admn=t1.get().strip()

    if admn=='':

        messagebox.showerror("ERROR", "ADMISSION NUMBER CANNOT BE EMPTY",parent=r1)

        t1.delete(0,END)

    else:

        cur.execute('select admission_number from student')

        for i in cur.fetchall():

            if i[0]==admn:

                flag3=False
```

```

        cur.execute('select date_of_return from book_records where
admission_number="{}"'.format(admn))

        l=cur.fetchall()

        for i in range(len(l)):

            if l==None or l[i][0]!=None:

                pass

            elif l[i][0]==None:

                flag3=True

                break

            if flag3==False:

                messagebox.showerror("ERROR","YOU HAVE NOT ISSUED ANY
BOOK",parent=r1)

                t1.delete(0,END)

            elif flag3==True:

                cur.execute('select book_code from book_records
where admission_number="{}" and date_of_return is Null'.format(admn))

                book_code=cur.fetchall()[0][0]

                date_of_return=date.today().strftime("%Y/%m/%d")

                cur.execute('update books set status="Not Issued"
where Book_code="{}"'.format(book_code))

                cur.execute('update book_records set
date_of_return="{}" where admission_number="{}" and date_of_return is
Null'.format(date_of_return,admn))

                mycon.commit()

                messagebox.showinfo("SUCCESS","BOOK RETURNED
SUCCESSFULLY",parent=r1)

                t1.delete(0,END)

            break

        else:

            messagebox.showerror("ERROR","ADMISSION NUMBER NOT
FOUND",parent=r1)

            t1.delete(0,END)

```

```
#BUTTONS

btn1=Button(r1,text='RETURN',font = ("Comic Sans MS bold",25),width=15,pady=5,borderwidth=3,bg='orange',fg='black',command=return_button)

btn1.place(y=600,x=340)

btn2=Button(r1,text='GO BACK',font = ("Comic Sans MS bold",25),width=15,pady=5,borderwidth=3,bg='orange',fg='black',command=r1.destroy)

btn2.place(y=600,x=870)

r1.mainloop()
```

#Booklist:

```
from tkinter import *
import mysql.connector as con
from tkinter import ttk
from Issue import *

def book_list():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1=Toplevel()

r1.title('BOOK LIST')

r1.state('zoomed')

r1.geometry('1080x720+200+35')

main_frame = Frame(r1,borderwidth=0,bg='#9EE2FF')

main_frame.pack(fill=BOTH, expand=1)

my_canvas = Canvas(main_frame,borderwidth=0,bg='#9EE2FF')

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL,
command=my_canvas.yview)

my_scrollbar.pack(side=RIGHT, fill=Y)

my_canvas.configure(yscrollcommand=my_scrollbar.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))
```

```

second_frame = Frame(my_canvas,borderwidth=0,bg="#9EE2FF")

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

#LABELS

l1=Label(second_frame,text='BOOK NAME',borderwidth=3,relief='solid',font =
("Comic Sans MS bold",20),bg='purple',fg='white',padx=100,pady=15)

l1.grid(row=0,column=0,padx=35,pady=10)

l2=Label(second_frame,text='AUTHOR',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",20),bg='purple',padx=70,pady=15,fg='white')

l2.grid(row=0,column=1,padx=35,pady=10)

l3=Label(second_frame,text='CODE',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",20),bg='purple',padx=30,pady=15,fg='white')

l3.grid(row=0,column=2,padx=35,pady=10)

l4=Label(second_frame,text='GENRE',borderwidth=3,relief='solid',bg='purple',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l4.grid(row=0,column=3,padx=35,pady=10)

l5=Label(second_frame,text='STATUS',borderwidth=3,relief='solid',bg='purple',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l5.grid(row=0,column=4,padx=35,pady=10)

cur.execute('select * from books')

books=cur.fetchall()

for i in range(len(books)):

l=Label(second_frame,text=books[i][0],wraplength=380,borderwidth=3,relief='solid',
', font = ("Comic Sans MS bold",15),fg='white',bg="#0ED53C',padx=5,pady=8,width=32)

l.grid(row=i+1,column=0,padx=10,pady=10)

l=Label(second_frame,text=books[i][1],borderwidth=3,relief='solid',wraplength=23
0, font = ("Comic Sans MS bold",15),fg='white',bg='red',padx=5,pady=8,width=20)

l.grid(row=i+1,column=1,padx=10,pady=10)

l=Label(second_frame,text=books[i][2],borderwidth=3,relief='solid', font =
("Comic Sans MS bold",15),fg='white',bg="#0123ff',padx=5,pady=8)

l.grid(row=i+1,column=2,padx=10,pady=10)

l=Label(second_frame,text=books[i][3],borderwidth=3,relief='solid', font =
("Comic Sans MS bold",15),fg='white',bg="#ff02ae',padx=5,pady=8,width=10)

```

```

l.grid(row=i+1,column=3,padx=10,pady=10)

if books[i][4]=='Not Issued':

    l=Button(second_frame,text=books[i][4],borderwidth=3,relief='solid',
font = ("Comic Sans MS bold",15),fg='white',bg='orange',padx=5,width=10,command=issue)

    l.grid(row=i+1,column=4,padx=10,pady=10)

else:

    l=Label(second_frame,text=books[i][4],borderwidth=3,relief='solid',
font = ("Comic Sans MS bold",15),fg='white',bg='orange',padx=6,pady=8,width=10)

    l.grid(row=i+1,column=4,padx=10,pady=10)

r1.mainloop()

```

#Admin:

```

from tkinter import *
from Details_Student import *
from Details_Staff import *
from Details_Book import *
from Book_Records import *

def admin():

    root1=Toplevel()

    root1.title('Admin')

    root1.geometry('1080x720+200+35')

    root1.state('zoomed')

    bg = PhotoImage(file = "snow.png")

    lbl1 = Label(root1, image = bg, borderwidth=0)

    lbl1.place(x = 0,y = 0)

#LABELS

    lbl = Label(root1, bg="#4cbb17",fg='White',text = "ADMIN", font = ("Comic Sans MS bold",40),pady=5,borderwidth=8,relief='ridge',width=15)

    lbl.place(x=530,y=40)

#BUTTONS

    btn1 = Button(root1, text = "BOOK DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9ACD32',fg='white',command=book_details,width=25)

    btn1.place(x=570,y=180)

```

```

btn2 = Button(root1, text = "VIEW BOOK RECORDS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#9ACD32', fg='white', command=book_records, width=25)

btn2.place(x=570,y=300)

btn3 = Button(root1, text = "STUDENT DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#9ACD32', fg='white', command=student_details, width=25)

btn3.place(x=570,y=420)

btn4 = Button(root1, text = "STAFF DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#9ACD32', fg='white', command=staff_details, width=25)

btn4.place(x=570,y=540)

btn5=Button(root1, text='GO BACK', pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#9ACD32', fg='white', command=root1.destroy, width=25)

btn5.place(x=570,y=660)

root1.mainloop()

```

#Password Admin:

```

from tkinter import *
from tkinter import ttk,messagebox
from Admin import *
import mysql.connector as con

def password():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1=Toplevel()

r1.title('LOGIN')

r1.geometry('960x340+300+170')

r1.focus()

#Background Image

global bg

bg = PhotoImage(file = "login.png")

lbl1 = Label(r1, image = bg, borderwidth=0)

lbl1.place(x = 0,y = 0)

#Labels

```

```
lb1=Label(r1, bg='#32E875',fg='White',text = "ENTER USERNAME:", font = ("Comic Sans MS bold",20),padx=20,pady=1,width=20)

lb1.place(x = 40, y = 30)

lb2=Label(r1, bg='#32E875',fg='White',text = "ENTER PASSWORD:", font = ("Comic Sans MS bold",20),padx=20,pady=1,width=20)

lb2.place(x = 40, y = 130)

#Textbox

t1=ttk.Entry(r1,width=24,font = ("Comic Sans MS bold",20))

t1.place(x = 520, y = 30)

t2=ttk.Entry(r1,width=24,font = ("Comic Sans MS bold",20),show='*')

t2.place(x = 520, y = 130)

#Login Function

def login():

    name=t1.get()

    password=t2.get()

    cur.execute('select * from password')

    l = cur.fetchall()

    for i in l:

        if i[0]==name and i[1]==password:

            r1.destroy()

            admin()

            break

    else:

        messagebox.showerror("ERROR","EITHER THE USERNAME OR THE PASSWORD IS WRONG",parent=r1)

    t1.delete(0,END)

    t2.delete(0,END)

def ch_password():

    r5=Toplevel()

    r5.title('CHANGE PASSWORD')

    r5.geometry('960x340+300+170')
```

```

r5.focus()

#Background Image

img = Label(r5, image = bg, borderwidth=0)

img.place(x = 0,y = 0)

#Labels

lb1=Label(r5, bg='#32E875',fg='White',text = "ENTER USERNAME:", font =
("Comic Sans MS bold",20),padx=20,pady=1,width=20)

lb1.place(x = 40, y =15)

lb2=Label(r5, bg='#32E875',fg='White',text = "ENTER OLD PASSWORD:", font
= ("Comic Sans MS bold",20),padx=20,pady=1,width=20)

lb2.place(x = 40, y = 85)

lb3=Label(r5, bg='#32E875',fg='White',text = "ENTER NEW PASSWORD:", font
= ("Comic Sans MS bold",20),padx=20,pady=1,width=20)

lb3.place(x = 40, y = 155)

#Textbox

t1=ttk.Entry(r5,width=24,font = ("Comic Sans MS bold",20))

t1.place(x = 520, y = 15)

t2=ttk.Entry(r5,width=24,font = ("Comic Sans MS bold",20),show='*')

t2.place(x = 520, y = 85)

t3=ttk.Entry(r5,width=24,font = ("Comic Sans MS bold",20),show='*')

t3.place(x = 520, y = 155)

def change_password():

    name=t1.get()

    old_password=t2.get()

    new_password=t3.get()

    cur.execute('select * from password')

    l = cur.fetchall()

    for i in l:

        if i[0] == name and i[1]==old_password:

            cur.execute('update password set password = "{}" where
user_name = "{}"'.format(new_password, name))

            mycon.commit()

```

```
    messagebox.showinfo("SUCCESSFUL","PASSWORD HAS BEEN UPDATED  
SUCCESSFULLY",parent=r5)  
  
    r5.destroy()  
  
    break  
  
else:  
  
    messagebox.showerror("ERROR","EITHER THE USERNAME OR THE OLD  
PASSWORD IS WRONG",parent=r5)  
  
    t1.delete(0,END)  
  
    t2.delete(0,END)  
  
    t3.delete(0,END)  
  
    btn1=Button(r5,text='CHANGE',font = ("Comic Sans MS  
bold",19),padx=20,pady=5,borderwidth=3,bg='#50CB93',fg='white',relief='solid',  
command = change_password)  
  
    btn1.place(x = 220, y = 235)  
  
    btn2=Button(r5,text='CANCEL',font = ("Comic Sans MS  
bold",19),padx=20,pady=5,borderwidth=3,bg='#50CB93',fg='white',relief='solid',  
command = r5.destroy)  
  
    btn2.place(x = 550, y = 235)  
  
#Buttons  
  
    btn1=Button(r1,text='CHANGE PASSWORD',font = ("Comic Sans MS  
bold",19),bg='#50CB93',fg='white',padx=20,pady=5,borderwidth=3,relief='solid',  
command = ch_password)  
  
    btn1.place(x = 70, y = 220)  
  
    btn2=Button(r1,text='LOGIN',font = ("Comic Sans MS  
bold",19),bg='#50CB93',fg='white',padx=20,pady=5,borderwidth=3,relief='solid',  
command = login)  
  
    btn2.place(x = 520, y = 220)  
  
    btn3=Button(r1,text='GO BACK',font = ("Comic Sans MS  
bold",19),bg='#50CB93',fg='white',padx=19,pady=5,borderwidth=3,relief='solid',  
command = r1.destroy)  
  
    btn3.place(x = 750, y = 220)  
  
    r1.mainloop()
```

#Book Records:

```

from tkinter import *
import mysql.connector as con
from tkinter import ttk

def book_records():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1=Tk()

r1.title('ISSUED BOOK RECORDS')

r1.state('zoomed')

r1.geometry('1920x1080')

main_frame = Frame(r1,borderwidth=0,bg='#9EE2FF')

main_frame.pack(fill=BOTH, expand=1)

my_canvas = Canvas(main_frame,borderwidth=0,bg='#9EE2FF')

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL,
command=my_canvas.yview)

my_scrollbar.pack(side=RIGHT, fill=Y)

my_scrollbar1= ttk.Scrollbar(my_canvas, orient=HORIZONTAL,
command=my_canvas.xview)

my_scrollbar1.pack(side=BOTTOM, fill=X)

my_canvas.configure(yscrollcommand=my_scrollbar.set,
xscrollcommand=my_scrollbar1.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))

second_frame = Frame(my_canvas,borderwidth=0,bg='#9EE2FF')

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

#LABELS

l1=Label(second_frame,text='NAME',borderwidth=3,relief='solid',font =
("Comic Sans MS bold",17),bg='purple',fg='white',pady=10,width=18)

l1.grid(row=0,column=0,padx=30,pady=10)

```

```

12=Label(second_frame,text='ADMISSION NUMBER',borderwidth=3,relief='solid',
font = ("Comic Sans MS bold",17),bg='purple',pady=10,padx=17,fg='white')

12.grid(row=0,column=1,padx=30,pady=10)

13=Label(second_frame,text='BOOK NAME',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",17),bg='purple',pady=10,fg='white',width=25)

13.grid(row=0,column=2,padx=30,pady=10)

14=Label(second_frame,text='BOOK
CODE',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",17),pady=8,fg='white',width=16)

14.grid(row=0,column=3,padx=20,pady=10)

15=Label(second_frame,text='DATE OF
ISSUE',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",17),pady=10,fg='white',width=16)

15.grid(row=0,column=4,padx=20,pady=10)

16=Label(second_frame,text='DATE OF
RETURN',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",17),pady=10,fg='white',width=16)

16.grid(row=0,column=5,padx=20,pady=10)

17=Label(second_frame,text='DUE
DATE',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",17),pady=10,fg='white',width=16)

17.grid(row=0,column=6,padx=20,pady=10)

18=Label(second_frame,text='ISSUE
TIME',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",17),pady=10,fg='white',width=16)

18.grid(row=0,column=7,padx=20,pady=10)

cur.execute('select * from book_records order by date_of_issue desc,
issue_time desc')

books=cur.fetchall()

for i in range(len(books)):

lb1=Label(second_frame,text=books[i][0],wraplength=240,borderwidth=3,relief='sol
id', font = ("Comic Sans MS
bold",14),fg='white',bg ='#54E346',padx=8,pady=5,width=20)

lb1.grid(row=i+1,column=0,padx=10,pady=10)

lb2=Label(second_frame,text=books[i][1],borderwidth=3,wraplength=160,relief='sol
id', font = ("Comic Sans MS
bold",14),fg='white',bg ='red',padx=5,pady=5,width=14)

```

```

lb2.grid(row=i+1, column=1, padx=10, pady=10)

lb3=Label(second_frame, text=books[i][2], borderwidth=3, wraplength=380, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#0123ff', padx=5, pady=5, width=32)

lb3.grid(row=i+1, column=2, padx=30, pady=10)

lb4=Label(second_frame, text=books[i][3], borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#ff02ae', padx=5, pady=5, width=7)

lb4.grid(row=i+1, column=3, padx=10, pady=10)

issue_date=books[i][4].strftime('%d/%m/%Y')

lb5=Label(second_frame, text=issue_date, borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#F98404', padx=5, pady=5, width=11)

lb5.grid(row=i+1, column=4, padx=10, pady=10)

if books[i][5]==None:

    lb6=Label(second_frame, text='-', borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#54E346', padx=5, pady=5, width=11)

    lb6.grid(row=i+1, column=5, padx=10, pady=10)

else:

    return_date=books[i][5].strftime('%d/%m/%Y')

lb6=Label(second_frame, text=return_date, borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#54E346', padx=5, pady=5, width=11)

lb6.grid(row=i+1, column=5, padx=10, pady=10)

due_date=books[i][6].strftime('%d/%m/%Y')

lb7=Label(second_frame, text=due_date, borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='red', padx=5, pady=5, width=11)

lb7.grid(row=i+1, column=6, padx=10, pady=10)

lb8=Label(second_frame, text=books[i][7], borderwidth=3, relief='solid', font = ("Comic Sans MS bold",14), fg='white', bg='#0123ff', padx=5, pady=5, width=11)

lb8.grid(row=i+1, column=7, padx=10, pady=10)

r1.mainloop()

```

#Details Books:

```

from tkinter import *
from Add_Book import *
from Delete_Book import *
from View_Book import *

def book_details():

    root=Toplevel()
    root.title('Book Details')
    root.geometry('1080x720+200+35')
    root.state('zoomed')

    #BACKGROUND IMAGE
    bg = PhotoImage(file = "Sunrise.png")

    lbl1 = Label(root, image = bg,borderwidth=0)
    lbl1.place(x = 0,y = 0)

    #LABELS
    lbl = Label(root, bg='#FB9300',fg='White',text = "BOOK DETAILS", font = ("Comic Sans MS bold",35),padx=20,width=15,pady=5,borderwidth=10,relief='ridge')
    lbl.place(x=530,y=70)

    #BUTTONS
    btn1 = Button(root, text = "ADD BOOK DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9B72AA',fg='white',width=25,command=book_add)
    btn1.place(x=570,y=220)

    btn2 = Button(root, text = "DELETE BOOK DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9B72AA',fg='white',width=25,command=book_delete)
    btn2.place(x=570,y=350)

    btn3 = Button(root, text = "VIEW BOOK DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9B72AA',fg='white',width=25,command=book_list)
    btn3.place(x=570,y=480)

    btn4 = Button(root,text='GO BACK', pady = 20,font=('Comic Sans MS',20),borderwidth=5,bg='#9B72AA',fg='white',command=root.destroy,width=25)
    btn4.place(x=570,y=610)

    root.mainloop()

```

#Add_Book:

```

from tkinter import *
import mysql.connector as con
from tkinter import ttk
from tkinter import messagebox

def book_add():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

root=Toplevel()
root.title('Add Books')
root.state('zoomed')

#BACKGROUND IMAGE

bg = PhotoImage(file = "scenery.png")

lbl1 = Label(root, image = bg,borderwidth=0)

lbl1.place(x = 0,y = 0)

#LABELS

lb1=Label(root,text='ADD BOOKS',font = ("Comic Sans MS bold",37),padx=20,pady=10,borderwidth=3,relief='solid',bg='#63DA3C',fg='white')

lb1.place(x=560,y=40)

lb2=Label(root, bg='#63DA3C',fg='White',text = "ENTER BOOK NAME:", font = ("Comic Sans MS bold",20),padx=30,pady=5,width=24)

lb2.place(x=200,y=260)

lb3=Label(root, bg='#63DA3C',fg='White',text = "ENTER AUTHOR:", font = ("Comic Sans MS bold",20),padx=30,pady=5,width=24)

lb3.place(x=200,y=340)

lb4=Label(root, bg='#63DA3C',fg='White',text = "ENTER GENRE:", font = ("Comic Sans MS bold",20),padx=30,pady=5,width=24)

lb4.place(x=200,y=420)

#TEXT BOX

t1=ttk.Entry(root,width=26,font = ("Comic Sans MS bold",20))

t1.place(x=750,y=260,height=50)

t2=ttk.Entry(root,width=26,font = ("Comic Sans MS bold",20))

```

```

t2.place(x=750,y=340,height=50)

t3=ttk.Combobox(root,height=10,values=['Genre','Novel','Biographies','Educationa
l','Fiction','Romance','Horror','Poetry','Fantasy','Crime','History','Narrative'
,'Mystery','Thriller','Suspense','Dystopian',"Children's","Inspirational",'Polit
ics','Spirituality','Health'],font = ("Comic Sans MS bold",20),state='readonly')

t3.current(0)

t3.place(x=750,y=420,height=52,width=423)

#FUNCTIONS

def add_button():

    book_name=t1.get().strip()

    book_author=t2.get().strip()

    book_genre=t3.get()

    if book_name=='' or book_author=='':

        messagebox.showerror("ERROR","NO FIELD CAN BE EMPTY",parent=root)

        t1.delete(0,END)

        t2.delete(0,END)

        t3.current(0)

    elif book_genre=='Genre':

        messagebox.showerror("ERROR","PLEASE SELECT A GENRE",parent=root)

        t1.delete(0,END)

        t2.delete(0,END)

        t3.current(0)

    else:

        cur.execute('select Name_of_Book,Author from books')

        l=cur.fetchall()

        flag=True

        if l!=None:

            for i in l:

                if i[0]==book_name and i[1]==book_author:

                    flag=False

```

```

        if flag==False:

            messagebox.showerror("ERROR", "THIS BOOK ALREADY
EXISTS", parent=root)

            t1.delete(0,END)
            t2.delete(0,END)
            t3.current(0)

        else:

            cur.execute('select max(book_code) from books where
genre="{}"'.format(book_genre))

            l=cur.fetchone()

            if l[0]!=None:

                book_code=l[0]

                digit=int(book_code[3:])

                digit+=1

                if digit>=100:

                    book_code=book_code[:3].upper()+str(digit)

                elif digit>=10:

                    book_code=book_code[:3]+'0'+str(digit)

                else:

                    book_code=book_code[:3]+'00'+str(digit)

            else:

                book_code=book_genre[:3].upper()+'001'

            cur.execute('insert into books(Name_of_Book, Author, Book_Code,
Genre, Status) values
("{}", "{}", "{}", "{}", "{}")'.format(book_name, book_author, book_code, book_genre, 'N
ot Issued'))

            mycon.commit()

            messagebox.showinfo("SUCCESS", "BOOK ADDED
SUCCESSFULLY", parent=root)

            t1.delete(0,END)
            t2.delete(0,END)
            t3.current(0)

#BUTTONS

```

```

btn1=Button(root,text='ADD',font = ("Comic Sans MS bold",24),bg='#63DA3C',fg='white',width=10,borderwidth=3,relief='solid',command=add_button)

btn1.place(x=390,y=580)

btn2=Button(root,text='GO BACK',font = ("Comic Sans MS bold",24),width=10,bg='#63DA3C',fg='white',borderwidth=3,relief='solid',command=root.destroy)

btn2.place(x=830,y=580)

root.mainloop()

```

#Delete Book:

```

from tkinter import *
from tkinter import ttk
import mysql.connector as con
from tkinter import messagebox

def book_delete():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

root=Toplevel()

root.title('Delete Books')

root.state('zoomed')

#BACKGROUND IMAGE

bg = PhotoImage(file = "Aurora.png")

lbl1 = Label(root, image = bg,borderwidth=0)

lbl1.place(x = 0,y = 0)

#LABELS

lb1=Label(root,text='DELETE BOOKS',font = ("Comic Sans MS bold",40),bg='#C1A7C8',fg='#041926',padx=20,pady=10,borderwidth=3,relief='solid')

lb1.place(x=540,y=60)

lb2=Label(root, bg='#4179A7',fg='White',text = "ENTER BOOK CODE:", font = ("Comic Sans MS bold",24),pady=5,width=25)

lb2.place(x = 210, y = 330)

#TEXT BOX

```

```

t1=ttk.Entry(root,width=24,font = ("Comic Sans MS bold",24))

t1.place(x = 800, y = 330,height=55)

#FUNCTIONS

def delete_button():

    book_code=t1.get().strip()

    if book_code=='':

        messagebox.showerror("ERROR","FIELD CANNOT BE EMPTY",parent=root)

        t1.delete(0,END)

    else:

        cur.execute('select status from books where
book_code="{}"'.format(book_code))

        b=cur.fetchone()

        if b!=None and b[0]=='Issued':

            messagebox.showerror("ERROR","BOOK IS CURRENTLY
ISSUED",parent=root)

            t1.delete(0,END)

        else:

            cur.execute('select book_code from books')

            flag2=False

            data=cur.fetchall()

            for i in data:

                if i[0]==book_code:

                    flag2=True

                    break

            if flag2==False:

                messagebox.showerror("ERROR","THIS BOOK CODE DOES NOT
EXISTS",parent=root)

                t1.delete(0,END)

            else:

                cur.execute('delete from books where
Book_Code="{}"'.format(book_code))

```

```

mycon.commit()

messagebox.showinfo("SUCCESS","BOOK DELETED
SUCCESSFULLY",parent=root)

t1.delete(0,END)

#BUTTONS

btn1=Button(root,text='DELETE',font = ("Comic Sans MS
bold",25),padx=40,fg='#041926',pady=5,bg='#C1A7C8',borderwidth=3,relief='solid',
command=delete_button)

btn1.place(y=570,x=420)

btn2=Button(root,text='GO BACK',font = ("Comic Sans MS
bold",25),padx=40,pady=5,fg='#041926',borderwidth=3,bg='#C1A7C8',relief='solid',
command=root.destroy)

btn2.place(y=570,x=850)

root.mainloop()

```

#View Book:

```

from tkinter import *
from tkinter import ttk
import mysql.connector as con

def book_list():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_proje
t')

cur=mycon.cursor()

r1=Toplevel()

r1.title('BOOK LIST')

r1.state('zoomed')

r1.geometry('1920x1080')

# Create A Main Frame

main_frame = Frame(r1,borderwidth=0,bg='#9EE2FF')

main_frame.pack(fill=BOTH, expand=1)

# Create A Canvas

my_canvas = Canvas(main_frame,borderwidth=0,bg='#9EE2FF')

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

# Add A Scrollbar To The Canvas

```

```

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL,
command=my_canvas.yview)

my_scrollbar.pack(side=RIGHT, fill=Y)

# Configure The Canvas

my_canvas.configure(yscrollcommand=my_scrollbar.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))

# Create ANOTHER Frame INSIDE the Canvas

second_frame = Frame(my_canvas,borderwidth=0,bg='#9EE2FF')

# Add that New frame To a Window In The Canvas

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

#LABELS

l1=Label(second_frame,text='BOOK NAME',borderwidth=3,relief='solid',font =
("Comic Sans MS bold",20),bg='purple',fg='white',padx=100,pady=15)

l1.grid(row=0,column=0,padx=35,pady=10)

l2=Label(second_frame,text='AUTHOR',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",20),bg='purple',padx=70,pady=15,fg='white')

l2.grid(row=0,column=1,padx=35,pady=10)

l3=Label(second_frame,text='CODE',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",20),bg='purple',padx=30,pady=15,fg='white')

l3.grid(row=0,column=2,padx=35,pady=10)

l4=Label(second_frame,text='GENRE',borderwidth=3,relief='solid',bg='purple',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l4.grid(row=0,column=3,padx=35,pady=10)

l5=Label(second_frame,text='STATUS',borderwidth=3,relief='solid',bg='purple',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l5.grid(row=0,column=4,padx=35,pady=10)

cur.execute('select * from books')

books=cur.fetchall()

for i in range(len(books)):

l=Label(second_frame,text=books[i][0],wraplength=380,borderwidth=3,relief='solid
', font = ("Comic Sans MS
bold",15),fg='white',bg='#0ED53C',padx=5,pady=5,width=32)

```

```

    l.grid(row=i+1, column=0, padx=10, pady=10)

l=Label(second_frame, text=books[i][1], borderwidth=3, wraplength=230, relief='solid',
', font = ("Comic Sans MS bold",15), fg='white', bg='red', padx=5, pady=5, width=20)

    l.grid(row=i+1, column=1, padx=10, pady=10)

l=Label(second_frame, text=books[i][2], borderwidth=3, relief='solid', font
= ("Comic Sans MS bold",15), fg='white', bg='#0123ff', padx=5, pady=5)

    l.grid(row=i+1, column=2, padx=10, pady=10)

l=Label(second_frame, text=books[i][3], borderwidth=3, relief='solid', font
= ("Comic Sans MS bold",15), fg='white', bg='#ff02ae', padx=5, pady=5, width=10)

    l.grid(row=i+1, column=3, padx=10, pady=10)

l=Label(second_frame, text=books[i][4], borderwidth=3, relief='solid', font
= ("Comic Sans MS bold",15), fg='white', bg='orange', padx=6, pady=5, width=10)

    l.grid(row=i+1, column=4, padx=10, pady=10)

r1.mainloop()

```

#Details Student:

```

from tkinter import *
from Add_Student import *
from View_Student import *
from Delete_Student import *

def student_details():

    t1 = Toplevel()

    t1.title("Student Details")

    t1.geometry('1080x720+200+35')

    t1.state('zoomed')

    bg = PhotoImage(file = "chakri.png")

    lbl1 = Label(t1, image = bg, borderwidth=0)

    lbl1.place(x = 0, y = 0)

#LABELS

    lbl = Label(t1, bg='orange', fg='White', text = "STUDENT DETAILS", font =
("Comic Sans MS bold",35), padx=20, pady=5, borderwidth=10, relief='ridge')

    lbl.pack(pady = 45)

#BUTTONS

```

```

btn1 = Button(t1, text = "ADD STUDENT DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#00bfff', fg='white', width=30, command = student_add)

btn1.pack(pady=13)

btn2 = Button(t1, text = "DELETE STUDENT DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#00bfff', fg='white', width=30, command = student_delete)

btn2.pack(pady=13)

btn3 = Button(t1, text = "VIEW STUDENT DETAILS", pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#00bfff', fg='white', width=30, command = student_view)

btn3.pack(pady=13)

btn4=Button(t1, text='GO BACK', pady = 20,font=('Comic Sans MS',20),borderwidth=5, bg ='#00bfff', fg='white', command=t1.destroy, width=30)

btn4.pack(pady=13)

t1.mainloop()

```

#Add Student:

```

from datetime import datetime
from tkinter import *
from tkinter import ttk,messagebox
import mysql.connector as con
from tkcalendar import DateEntry

def student_add():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1 = Toplevel()

r1.title("Add Student Details")

r1.state('zoomed')

#BACKGROUND IMAGE

bg = PhotoImage(file = "leaf.png")

img = Label(r1, image = bg,borderwidth=0)

img.place(x = 0,y = 0)

#LABELS

```

```

lb1=Label(r1,text='ADD STUDENT DETAILS',fg='White',font = ("Comic Sans MS bold",35),padx=20,pady=10,borderwidth=3,relief='solid', bg = '#000080')

lb1.place(x = 430, y = 55)

lb2=Label(r1, bg='#A2242F',fg='White',text = "ENTER NAME:", font = ("Comic Sans MS bold",17),padx=20,pady=5,width=26)

lb2.place(x = 280, y = 225)

lb3=Label(r1, bg='#A2242F',fg='White',text = "ENTER DATE OF BIRTH:", font = ("Comic Sans MS bold",17),padx=20,pady=5,width=26)

lb3.place(x = 280, y = 305)

lb4=Label(r1, bg='#A2242F',fg='White',text = "ENTER ADMISSION NUMBER:", font = ("Comic Sans MS bold",17),padx=20,pady=5,width=26)

lb4.place(x = 280, y = 385)

lb5=Label(r1, bg='#A2242F',fg='White',text = "ENTER CLASS:", font = ("Comic Sans MS bold",17),padx=20,pady=5,width=26)

lb5.place(x = 280, y = 465)

lb6=Label(r1, bg='#A2242F',fg='White',text = "ENTER SECTION:", font = ("Comic Sans MS bold",17),padx=20,pady=5,width=26)

lb6.place(x = 280, y = 545)

#TEXT BOX

t1=ttk.Entry(r1,width=30,font = ("Comic Sans MS bold",17))

t1.place(x = 780, y = 225)

cal = DateEntry(r1,date_pattern='dd/MM/yyyy',font = ("Comic Sans MS bold",18),state='readonly',background='darkblue',foreground='white')

cal.place(x = 780, y = 305,width=424)

t2=ttk.Entry(r1,width=30,font = ("Comic Sans MS bold",17))

t2.place(x = 780, y = 385)

dropdown=ttk.Combobox(r1,values=['Class','1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'], font = ("Comic Sans MS bold",17),state='readonly')

dropdown.current(0)

dropdown.place(x = 780, y = 465)

dropdown.config(width = 29, height = 6)

dropdown1=ttk.Combobox(r1,values=['Section','A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'], font = ("Comic Sans MS bold",17),state='readonly')

dropdown1.current(0)

```

```
dropdown1.place(x = 780, y = 545)
dropdown1.config(width = 29, height = 6)

def add_button():
    name = t1.get().strip()
    dob = cal.get()
    dob=datetime.strptime(dob, '%d/%m/%Y')
    dob=dob.strftime('%Y/%m/%d')
    admin_no = t2.get().strip()
    clas=dropdown.get()
    section = dropdown1.get()

    if name=='' or admin_no=='':
        messagebox.showerror("ERROR", "NO FIELD CAN BE EMPTY", parent=r1)
        t1.delete(0,END)
        t2.delete(0,END)
        dropdown.current(0)
        dropdown1.current(0)

    elif section=='Section':
        messagebox.showerror("ERROR", "PLEASE SELECT SECTION", parent=r1)
        t1.delete(0,END)
        t2.delete(0,END)
        dropdown.current(0)
        dropdown1.current(0)

    elif clas=='Class':
        messagebox.showerror("ERROR", "PLEASE SELECT CLASS", parent=r1)
        t1.delete(0,END)
        t2.delete(0,END)
        dropdown.current(0)
        dropdown1.current(0)

    else:
```

```

cur.execute('select Admission_Number from student')

l=cur.fetchall()

flag=True

for i in l:

    if i[0]==admin_no:

        flag=False

    if flag==False:

        messagebox.showerror("ERROR", "ADMISSION NUMBER ALREADY EXISTS", parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

    else:

        cur.execute('insert into student(Name, Date_of_Birth, Admission_Number, Class, Section)
values ("{}", "{}", "{}", "{}", "{}")'.format(name,dob,admin_no,clas,section))

        mycon.commit()

        messagebox.showinfo("SUCCESS", "STUDENT DETAIL ADDED SUCCESSFULLY", parent=r1)

        t1.delete(0,END)

        t2.delete(0,END)

        dropdown.current(0)

        dropdown1.current(0)

#BUTTONS

btn1=Button(r1,text='ADD',fg='White',font = ("Comic Sans MS bold",20), bg ='#000080',padx=20,pady=5,borderwidth=3,relief='solid',width=12,command = add_button)

btn1.place(y=675,x=410)

btn2=Button(r1,text='GO BACK',fg='White',font = ("Comic Sans MS bold",20), bg ='#000080',padx=20,pady=5,borderwidth=3,relief='solid',width=12,command=r1.destroy)

btn2.place(y=675,x=810)

r1.mainloop()

```

#Delete_Student:

```

from tkinter import *
import mysql.connector as con
from tkinter import messagebox

def student_delete():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1 = Toplevel()

r1.title("Delete Student Details")

r1.geometry('1080x720+200+35')

r1.state('zoomed')

bg = PhotoImage(file = "hogwarts.png")

img = Label(r1, image = bg,borderwidth=0)

img.place(x = 0,y = 0)

lb1=Label(r1,text='DELETE STUDENT DETAILS',font = ("Comic Sans MS bold",35),bg = "#E0B589",padx=20,pady=10,borderwidth=3,relief='solid')

lb1.place(x=400,y=60)

lb2=Label(r1, bg='#798EA4',fg='White',text = "ENTER ADMISSION NUMBER:", font = ("Comic Sans MS bold",20),width=26,pady=5)

lb2.place(x = 270, y = 330)

#TEXT BOX

t1=Entry(r1,width=26,font = ("Comic Sans MS bold",20),highlightcolor='blue',highlightthickness=4)

t1.place(x = 800, y = 330)

def delete_button():

admin_no = t1.get().strip()

if admin_no == "" :

    messagebox.showerror("ERROR","ADMISSION NUMBER CANNOT BE EMPTY",parent=r1)

    t1.delete(0,END)

else:

```

```

        cur.execute("select Admission_Number from student")

        flag = False

        data = cur.fetchall()

        for i in data:

            if i[0] == admin_no:

                flag = True

                break

        if flag == False:

            messagebox.showerror("ERROR", "ADMISSION NUMBER DOES NOT
EXIST", parent=r1)

            t1.delete(0,END)

        else:

            cur.execute('select date_of_return from book_records where
admission_number="{}"'.format(admin_no))

            l=cur.fetchall()

            flag=False

            if l!=[]:

                for i in l:

                    if i[0]==None:

                        flag=True

            if flag==True:

                messagebox.showerror("ERROR", "STUDENT HAS A PENDING ISSUED
BOOK", parent=r1)

                t1.delete(0,END)

            else:

                cur.execute("delete from student where Admission_Number =
'{}' ".format(admin_no))

                mycon.commit()

                messagebox.showinfo("SUCCESS", "STUDENT DETAIL DELETED
SUCCESSFULLY", parent=r1)

                t1.delete(0,END)

        btn1=Button(r1,text='DELETE',font = ("Comic Sans MS bold",20),bg =
"#E0B589",padx=20,pady=5,borderwidth=3,relief='solid', command = delete_
button)
    
```

```

btn1.place(y=570,x=480)

btn2=Button(r1,text='GO BACK',font = ("Comic Sans MS bold",20),bg =
"#E0B589", padx=20,pady=5,borderwidth=3,relief='solid',command=r1.destroy)

btn2.place(y=570,x=860)

r1.mainloop()

```

#View Student:

```

from tkinter import *
from tkinter import ttk
import mysql.connector as con

def student_view():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1 = Toplevel()

r1.title("View Student Details")

r1.geometry('1080x720+200+35')

r1.state('zoomed')

# Create A Main Frame

main_frame = Frame(r1,borderwidth=0,bg='#9EE2FF')

main_frame.pack(fill=BOTH, expand=1)

# Create A Canvas

my_canvas = Canvas(main_frame,borderwidth=0,bg='#9EE2FF')

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

# Add A Scrollbar To The Canvas

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL,
command=my_canvas.yview)

my_scrollbar.pack(side=RIGHT, fill=Y)

# Configure The Canvas

my_canvas.configure(yscrollcommand=my_scrollbar.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))

```

```

# Create ANOTHER Frame INSIDE the Canvas

second_frame = Frame(my_canvas,borderwidth=0,bg='#9EE2FF')

# Add that New frame To a Window In The Canvas

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

#LABELS

l1=Label(second_frame,text='NAME',borderwidth=3,relief='solid',font =
("Comic Sans MS bold",20),bg ='#d68438',fg='white',padx=100,pady=15)

l1.grid(row=0,column=0,padx=13,pady=10)

l2=Label(second_frame,text='DATE OF BIRTH',borderwidth=3,relief='solid',
font = ("Comic Sans MS bold",20),bg ='#f1b24b',padx=70,pady=15,fg='white')

l2.grid(row=0,column=1,padx=13,pady=10)

l3=Label(second_frame,text='ADMISSION NO.',borderwidth=3,relief='solid',
font = ("Comic Sans MS bold",20),bg ='#36846b',padx=30,pady=15,fg='white')

l3.grid(row=0,column=2,padx=13,pady=10)

l4=Label(second_frame,text='CLASS',borderwidth=3,relief='solid',bg ='#4bb39a',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l4.grid(row=0,column=3,padx=13,pady=10)

l5=Label(second_frame,text='SECTION',borderwidth=3,relief='solid',bg ='#d68438',
font = ("Comic Sans MS bold",20),padx=40,pady=15,fg='white')

l5.grid(row=0,column=4,padx=13,pady=10)

cur.execute('select * from student')

student=cur.fetchall()

for i in range(len(student)):

    for j in range(len(student[i])):

        if j == 0:

            label =

Label(second_frame,text=student[i][j],wraplength=240,borderwidth=3,relief='solid',
', font = ("Comic Sans MS bold",15),bg ='#d68438',padx=5,pady=5,width=20)

            label.grid(row=i+1,column=j,padx=10,pady=10)

        elif j == 1:

            label =

Label(second_frame,text=student[i][j],borderwidth=3,relief='solid', font =
("Comic Sans MS bold",15),bg ='#f1b24b',padx=5,pady=5,width=15)

```

```

        label.grid(row=i+1,column=j,padx=10,pady=10)

    elif j == 2:

        label =
Label(second_frame,text=student[i][j],wraplength=180,borderwidth=3,relief='solid',
', font = ("Comic Sans MS bold",15),bg ='#36846b',padx=5,pady=5,width=15)

        label.grid(row=i+1,column=j,padx=10,pady=10)

    elif j == 3:

        label =
Label(second_frame,text=student[i][j],borderwidth=3,relief='solid', font =
("Comic Sans MS bold",15),bg ='#4bb39a',padx=5,pady=5,width=5)

        label.grid(row=i+1,column=j,padx=10,pady=10)

    elif j == 4:

        label =
Label(second_frame,text=student[i][j],borderwidth=3,relief='solid', font =
("Comic Sans MS bold",15),bg ='#d68438',padx=5,pady=5,width=5)

        label.grid(row=i+1,column=j,padx=10,pady=10)

r1.mainloop()

```

#Details_Staff:

```

from tkinter import *
from Add_Staff import *
from Delete_Staff import *
from View_Staff import *

def staff_details():

    r1=Toplevel()

    r1.state('zoomed')

    r1.title('Staff Details')

    #BACKGROUND IMAGE

    bg = PhotoImage(file = "Waterfall.png")

    lbl1 = Label(r1, image = bg,borderwidth=0)

    lbl1.place(x = 0,y = 0)

    #LABELS

    lbl = Label(r1, bg ='#FB9300',fg='White',text = "STAFF DETAILS", font =
("Comic Sans MS bold",35),padx=20,width=16,pady=5,borderwidth=10,relief='ridge')

```

```

lbl.place(x=533,y=70)

#BUTTONS

btn1 = Button(r1, text = "ADD STAFF DETAILS", padx = 20, pady =
20,font=('Comic Sans
MS',20),borderwidth=5,bg='#81B214',fg='white',width=25,command=staff_add)

btn1.place(x=570,y=220)

btn2 = Button(r1, text = "DELETE STAFF DETAILS", padx = 20, pady =
20,font=('Comic Sans
MS',20),borderwidth=5,bg='#81B214',fg='white',width=25,command=staff_delete)

btn2.place(x=570,y=350)

btn3 = Button(r1, text = "VIEW STAFF DETAILS", padx = 20, pady =
20,font=('Comic Sans
MS',20),borderwidth=5,bg='#81B214',fg='white',width=25,command=staff_view)

btn3.place(x=570,y=480)

btn4 = Button(r1, text = "GO BACK", padx = 20, pady = 20,font=('Comic
Sans MS',20),borderwidth=5,bg='#81B214',fg='white',width=25,command=r1.destroy)

btn4.place(x=570,y=610)

r1.mainloop()

```

#Add_Staff:

```

from tkinter import *
import mysql.connector as con
from tkinter import ttk, messagebox

def staff_add():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_proje
t')

cur=mycon.cursor()

r1=Toplevel()

r1.title('Add Staff Details')

r1.state('zoomed')

#BACKGROUND IMAGE

bg = PhotoImage(file = "denmark.png")

lbl1 = Label(r1, image = bg,borderwidth=0)

lbl1.place(x = 0,y = 0)

```

```

#LABELS

lb1=Label(r1,text='ADD STAFF DETAILS',font = ("Comic Sans MS bold", 40),padx=20,pady=10,borderwidth=3,relief='solid', bg='#008000',fg='White')

lb1.place(x=450,y=50)

lb2=Label(r1, bg='#9ACD32',fg='White',text = "ENTER NAME:", font = ("Comic Sans MS bold",20),padx=20,pady=5,width=24)

lb2.place(x=270,y=250)

lb4=Label(r1, bg='#9ACD32',fg='White',text = "ENTER GENDER:", font = ("Comic Sans MS bold",20),padx=20,pady=5,width=24)

lb4.place(x=270,y=325)

lb5=Label(r1, bg='#9ACD32',fg='White',text = "ENTER ADDRESS:", font = ("Comic Sans MS bold",20),padx=20,pady=5,width=24)

lb5.place(x=270,y=400)

lb6=Label(r1, bg='#9ACD32',fg='White',text = "ENTER PHONE NUMBER:", font = ("Comic Sans MS bold",20),padx=20,pady=5,width=24)

lb6.place(x=270,y=475)

#TEXT BOXES

t1=ttk.Entry(r1,width=26,font = ("Comic Sans MS bold",20))

t1.place(x=780,y=250,height=50)

dropdown=ttk.Combobox(r1,values=['Gender','M','F'],font = ("Comic Sans MS bold",22),state='readonly')

dropdown.current(0)

dropdown.place(x=780,y=325,width=422,height=52)

t4=ttk.Entry(r1,width=26,font = ("Comic Sans MS bold",20))

t4.place(x=780,y=400,height=50)

t5=ttk.Entry(r1,width=26,font = ("Comic Sans MS bold",20))

t5.place(x=780,y=475,height=50)

def clear_fields():

    t1.delete(0,END)

    dropdown.current(0)

    t4.delete(0,END)

    t5.delete(0,END)

```

```

def add_button():

    staff_name=t1.get().strip()

    gender=dropdown.get()

    address=t4.get().strip()

    ph_no=t5.get().strip()

    if staff_name=='' or address=='' or ph_no=='':

        messagebox.showerror("ERROR", 'NO FIELD CAN BE
EMPTY',parent=r1)

    else:

        if ph_no.isdigit()==False:

            messagebox.showerror("ERROR", 'PHONE NUMBER CAN ONLY BE
DIGITS',parent=r1)

            t5.delete(0,END)

        elif len(ph_no)!=10:

            messagebox.showerror("ERROR", 'PHONE NUMBER CAN ONLY BE
10 DIGITS LONG',parent=r1)

            t5.delete(0,END)

        elif gender=='Gender':

            messagebox.showerror("ERROR", 'PLEASE SELECT A
GENDER',parent=r1)

        else:
            flag=True

            cur.execute('select phone_number from staff')

            l=cur.fetchall()

            if l!=None:

                for i in l:

                    if i[0]==ph_no:

                        flag=False

            if flag==False:

                messagebox.showerror("ERROR", 'STAFF WITH SAME PHONE
NUMBER EXISTS',parent=r1)

            t1.delete(0,END)

            dropdown.current(0)

```

```

        t4.delete(0,END)

        t5.delete(0,END)

    else:
        cur.execute('select max(staffid) from staff')

        l=cur.fetchone()

        if l[0] !=None:

            staff_id=(l[0] + 1)

        else:

            staff_id=1

        cur.execute('insert into staff
values ("{}", "{}", "{}", "{}", "{}")'.format(staff_name, staff_id, gender, address, ph_no)
)

        mycon.commit()

        messagebox.showinfo("SUCCESS", 'RECORD SUCCESSFULLY
ADDED', parent=r1)

        t1.delete(0,END)

        dropdown.current(0)

        t4.delete(0,END)

        t5.delete(0,END)

    btn1 = Button(r1, text = "ADD STAFF DETAILS", padx = 20, pady =
20,font=('Comic Sans
MS',15),borderwidth=3,relief='solid',bg='#008000',fg='white',width=17,command=ad
d_button)

    btn1.place(x=269,y=630)

    btn2 = Button(r1, text = "CLEAR", padx = 20, pady = 20,font=('Comic Sans
MS',15),borderwidth=3,relief='solid',bg='#008000',fg='white',width=15,command=cl
ear_fields)

    btn2.place(x=625,y=630)

    btn3 = Button(r1, text = "GO BACK", padx = 25, pady = 20,font=('Comic
Sans
MS',15),borderwidth=3,relief='solid',bg='#008000',fg='white',width=15,command=r1
.destroy)

    btn3.place(x=960,y=630)

    r1.mainloop()

```

#Delete_Staff:

```

from tkinter import *
import mysql.connector as con
from tkinter import ttk,messagebox

def staff_delete():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1=Toplevel()

r1.title('Delete Staff Details')

r1.state('zoomed')

#BACKGROUND IMAGE

bg = PhotoImage(file = "pinky.png")

lbl1 = Label(r1, image = bg,borderwidth=0)

lbl1.place(x = 0,y = 0)

#LABELS

lb1=Label(r1,text='DELETE STAFF DETAILS',font = ("Comic Sans MS bold",37),bg='#C71585',fg='white',padx=20,pady=10,borderwidth=3,relief='solid')

lb1.place(x=420,y=60)

lb2=Label(r1, bg='#FFB6C1',fg='White',text = "ENTER STAFF ID:", font = ("Comic Sans MS bold",24),pady=5,width=25)

lb2.place(x = 210, y = 330)

#TEXT BOXES

t1=ttk.Entry(r1,width=24,font = ("Comic Sans MS bold",24))

t1.place(x = 800, y = 330,height=55)

def delete_button():

    staff_id=t1.get().strip()

    if staff_id=='':

        messagebox.showerror("ERROR","STAFF ID CANNOT BE EMPTY",parent=r1)

    else:

        cur.execute('select staffid from staff')

```

```
l=cur.fetchall()

if l!=None:

    for i in l:

        if str(i[0])==staff_id:

            cur.execute('delete from staff where
staffid={}'.format(staff_id))

            mycon.commit()

            t1.delete(0,END)

            messagebox.showinfo("SUCCESS","RECORD SUCCESSFULLY
DELETED",parent=r1)

            break

    else:

        messagebox.showerror("ERROR","STAFF ID IS
INCORRECT",parent=r1)

        t1.delete(0,END)

    else:

        messagebox.showerror("ERROR","STAFF TABLE IS EMPTY",parent=r1)

        t1.delete(0,END)

btn1=Button(r1,text='DELETE',font = ("Comic Sans MS
bold",24),padx=40,fg='white',pady=5,bg='#C71585',borderwidth=3,relief='solid',co
mmand=delete_button)

btn1.place(y=570,x=420)

btn2=Button(r1,text='GO BACK',font = ("Comic Sans MS
bold",24),padx=40,pady=5,fg='white',borderwidth=3,bg='#C71585',relief='solid',co
mmand=r1.destroy)

btn2.place(y=570,x=850)

r1.mainloop()
```

#View_Staff:

```

from tkinter import *
import mysql.connector as con
from tkinter import ttk

def staff_view():

mycon=con.connect(host='localhost',user='root',passwd='ar12',database='cs_project')

cur=mycon.cursor()

r1=Toplevel()

r1.state('zoomed')

r1.title('View Staff Details')

# Create A Main Frame

main_frame = Frame(r1,borderwidth=0,bg='#9EE2FF')

main_frame.pack(fill=BOTH, expand=1)

# Create A Canvas

my_canvas = Canvas(main_frame,borderwidth=0,bg='#9EE2FF')

my_canvas.pack(side=LEFT, fill=BOTH, expand=1)

# Add A Scrollbar To The Canvas

my_scrollbar = ttk.Scrollbar(main_frame, orient=VERTICAL,
command=my_canvas.yview)

my_scrollbar.pack(side=RIGHT, fill=Y)

# Configure The Canvas

my_canvas.configure(yscrollcommand=my_scrollbar.set)

my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion =
my_canvas.bbox("all")))

# Create ANOTHER Frame INSIDE the Canvas

second_frame = Frame(my_canvas,borderwidth=0,bg='#9EE2FF')

# Add that New frame To a Window In The Canvas

my_canvas.create_window((0,0), window=second_frame, anchor="nw")

#LABELS

l1=Label(second_frame,text='NAME',borderwidth=3,relief='solid',font =
("Comic Sans MS bold",19),bg='purple',fg='white',width=17,pady=15)

```

```

11.grid(row=0,column=0,padx=22,pady=15)

12=Label(second_frame,text='STAFF ID',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",19),bg='purple',width=11,pady=15,fg='white')

12.grid(row=0,column=1,padx=27,pady=15)

13=Label(second_frame,text='GENDER',borderwidth=3,relief='solid', font =
("Comic Sans MS bold",19),bg='purple',width=8,pady=15,fg='white')

13.grid(row=0,column=2,padx=27,pady=15)

14=Label(second_frame,text='ADDRESS',borderwidth=3,relief='solid',bg='purple',
font = ("Comic Sans MS bold",19),padx=22,width=20,pady=15,fg='white')

14.grid(row=0,column=3,padx=22,pady=15)

15=Label(second_frame,text='CONTACT
NUMBER',borderwidth=3,relief='solid',bg='purple', font = ("Comic Sans MS
bold",19),padx=30,pady=15,fg='white')

15.grid(row=0,column=4,padx=22,pady=15)

cur.execute('select * from staff')

staff=cur.fetchall()

for i in range(len(staff)):

    for j in range(len(staff[i])):

        if j==0:

l=Label(second_frame,text=staff[i][j],borderwidth=3,wraplength=240,relief='solid
',font = ("Comic Sans MS bold",15),bg="#DC143C",fg='white',width=21,pady=15)

        l.grid(row=i+1,column=j,padx=25,pady=10)

    elif j==1:

l=Label(second_frame,text=staff[i][j],borderwidth=3,relief='solid',font =
("Comic Sans MS bold",15),bg="#00BFFF",fg='white',width=10,padx=5,pady=15)

        l.grid(row=i+1,column=j,padx=35,pady=10)

    elif j==2:

l=Label(second_frame,text=staff[i][j],borderwidth=3,relief='solid',font =
("Comic Sans MS bold",15),bg="#FF1493",fg='white',width=5,padx=5,pady=15)

        l.grid(row=i+1,column=j,padx=35,pady=10)

    elif j==3:

```

```
l=Label(second_frame,text=staff[i][j],borderwidth=3,wraplength=330,relief='solid'
',font = ("Comic Sans MS
bold",15),bg ='#40E0D0',fg='white',width=28,padx=5,pady=15)

l.grid(row=i+1,column=j,padx=35,pady=10)

else:

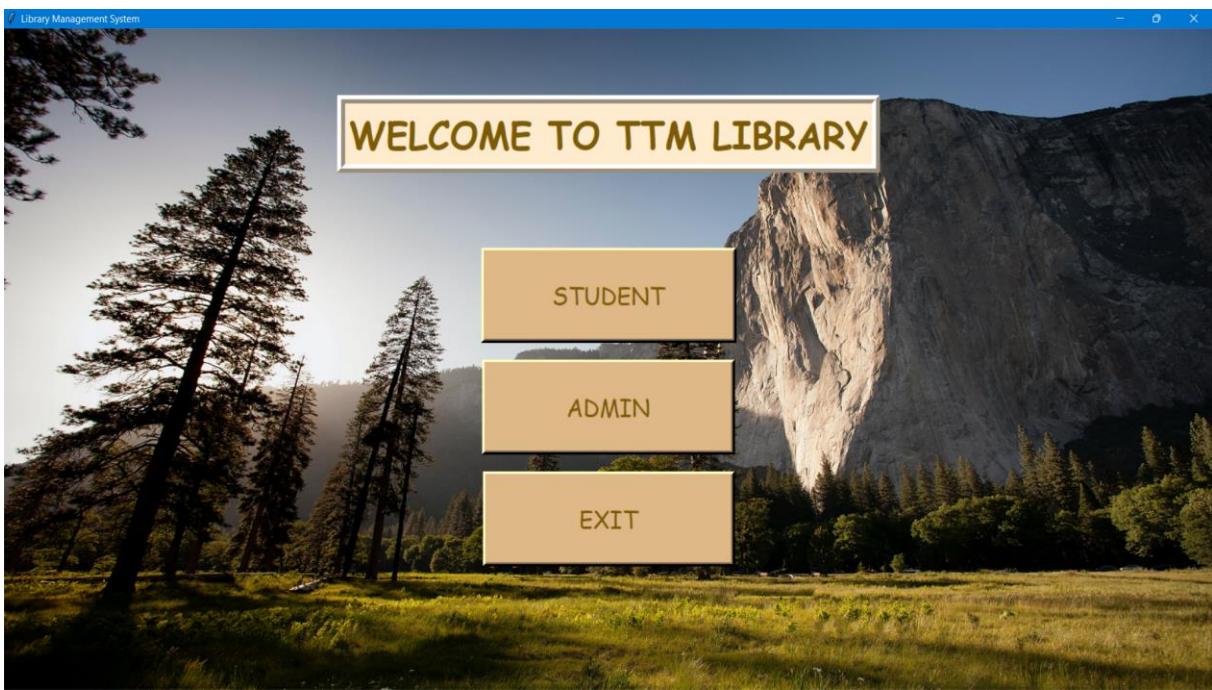
l=Label(second_frame,text=staff[i][j],borderwidth=3,relief='solid',font =
("Comic Sans MS bold",15),bg ='#008000',fg='white',width=15,padx=5,pady=15)

l.grid(row=i+1,column=j,padx=35,pady=10)

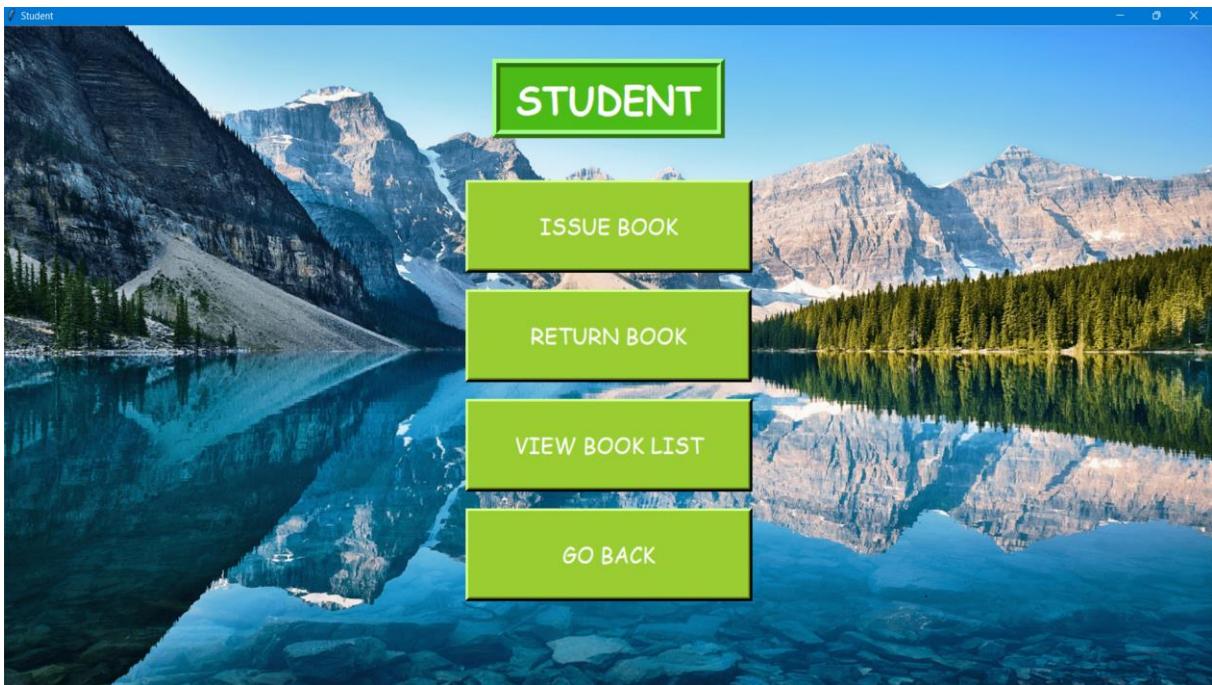
r1.mainloop()
```

OUTPUTS

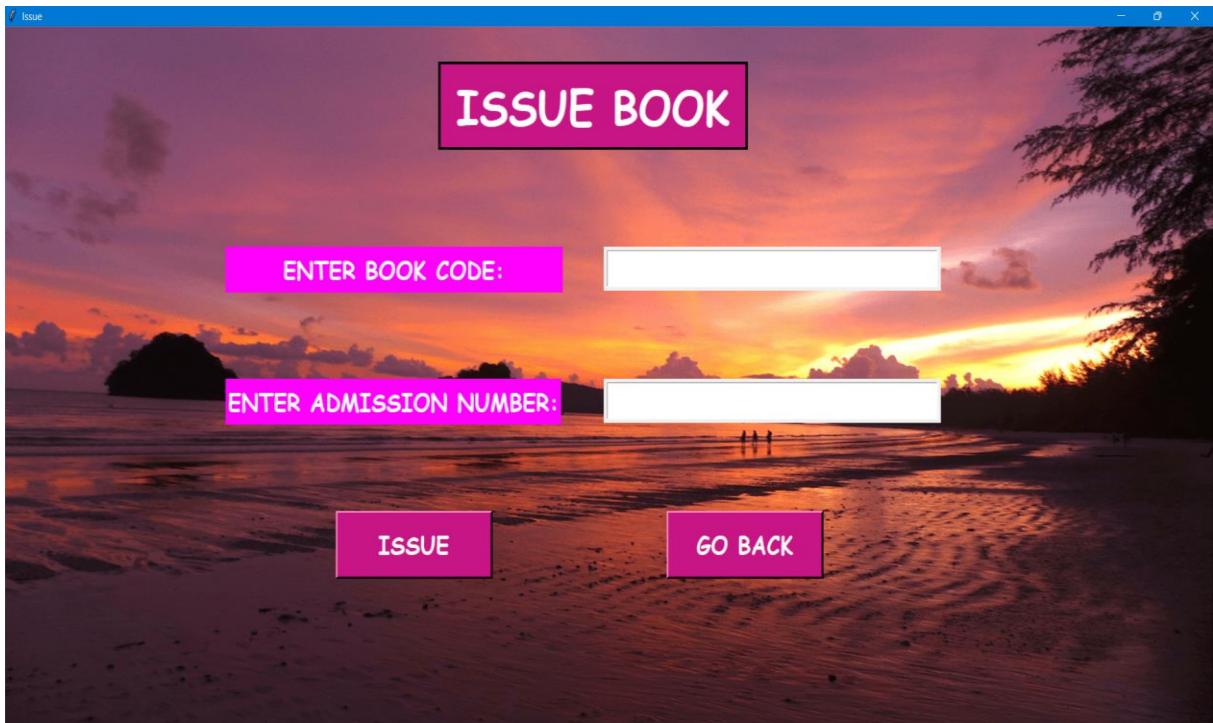
Main Window:



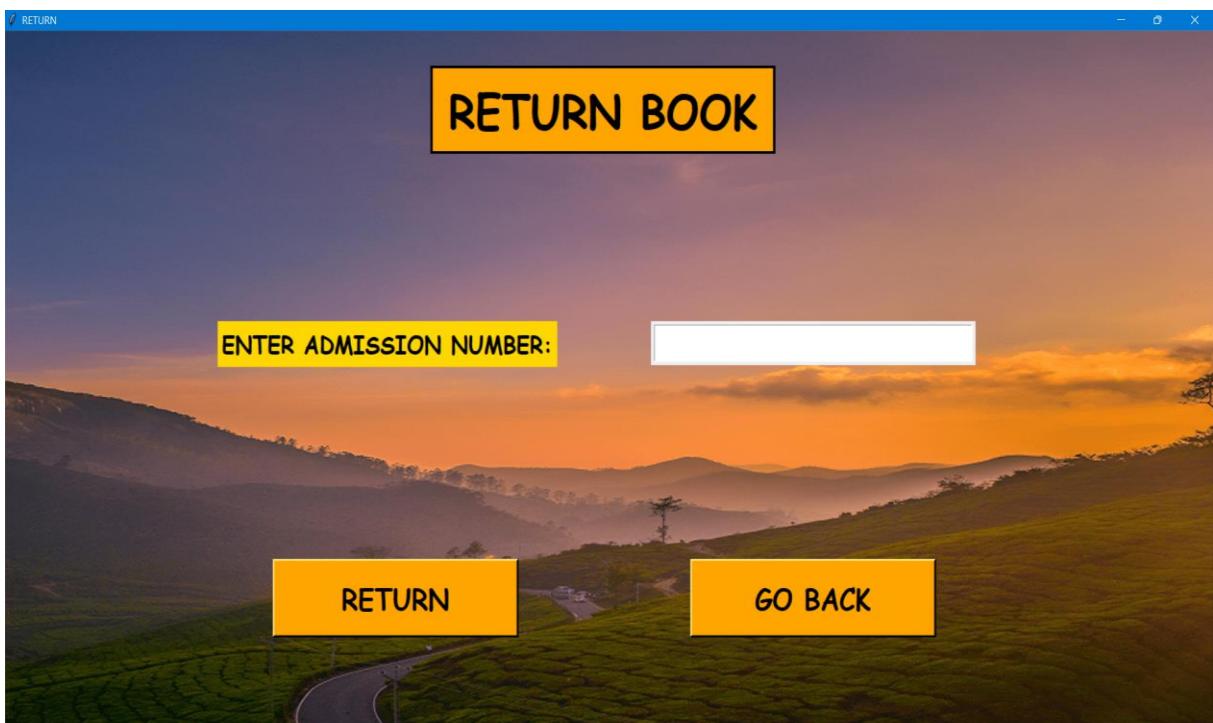
1. Student:



1.1. Issue Book:



1.2. Return Book:



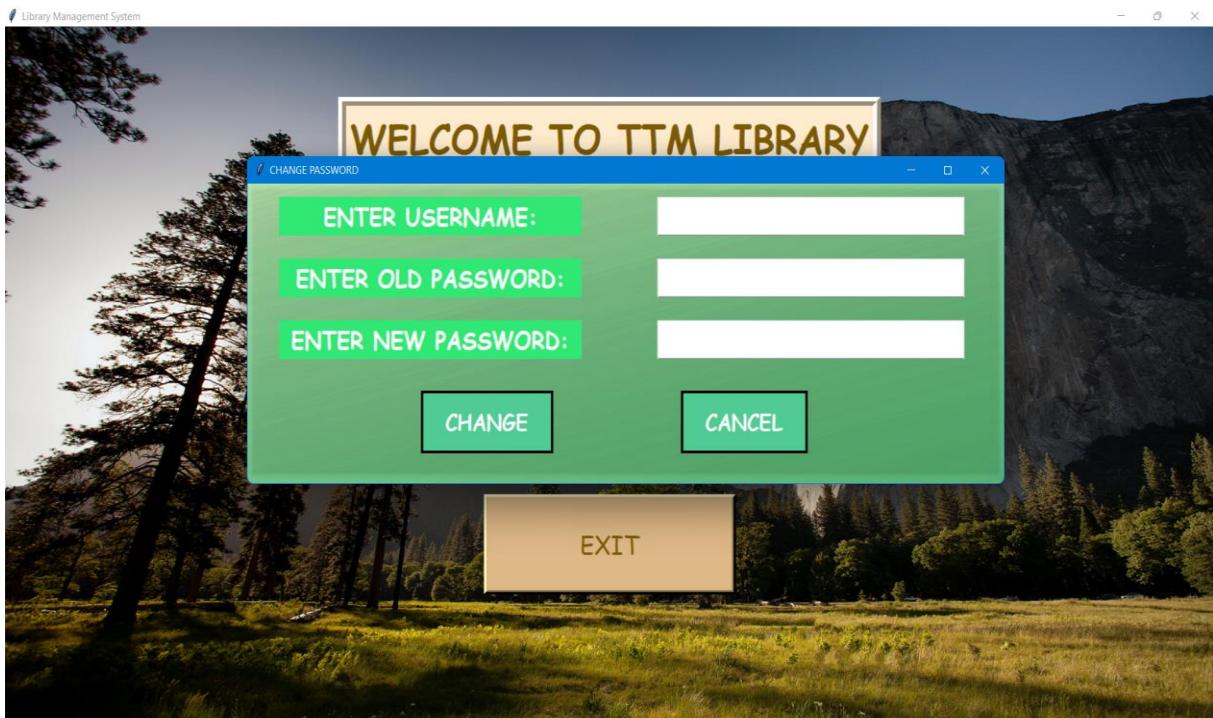
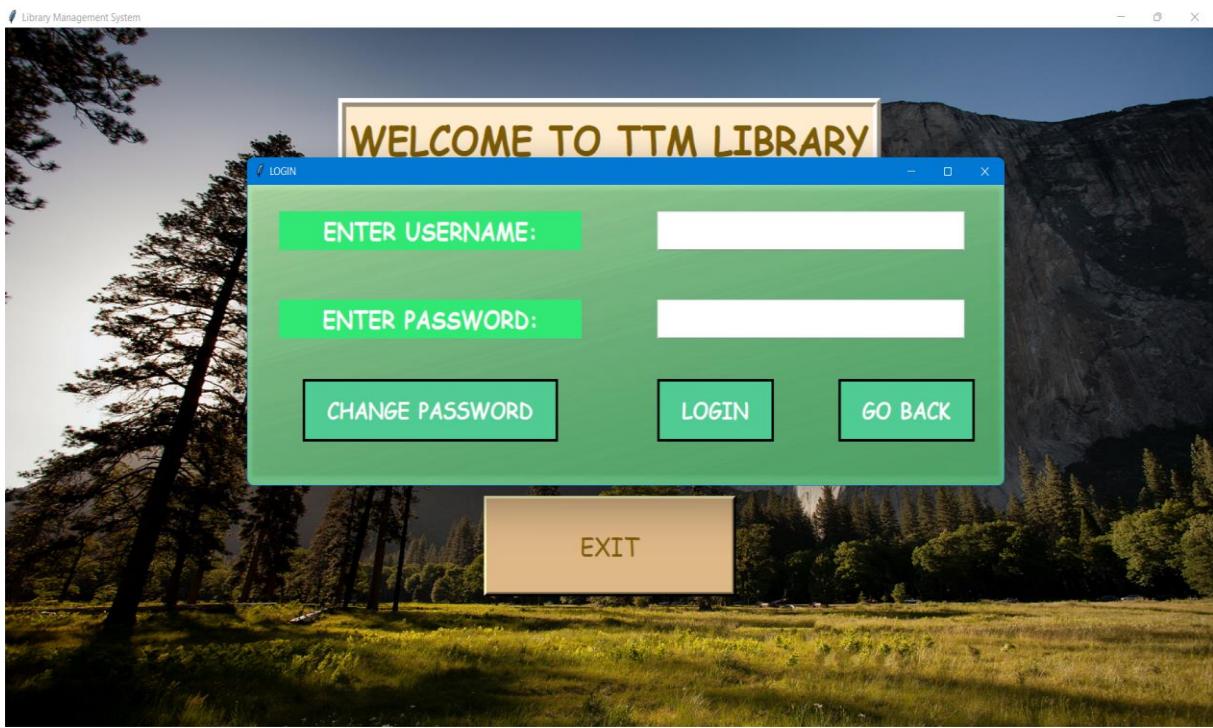
1.3. View Book Details:

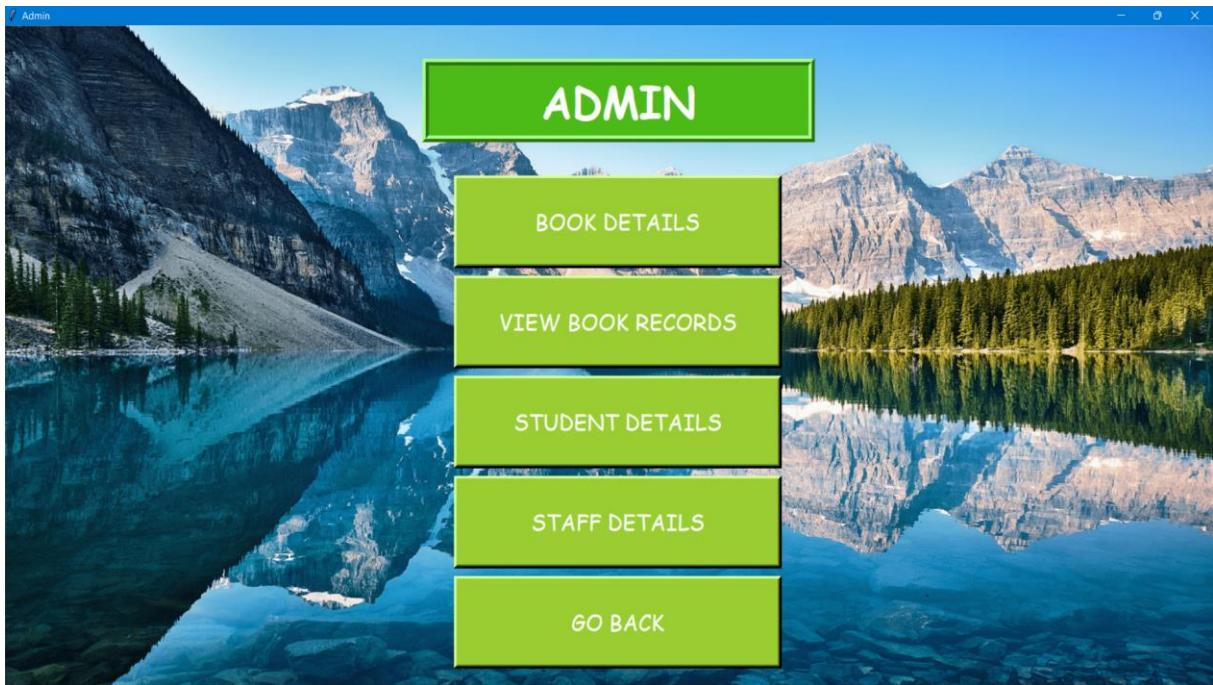
BOOK NAME	AUTHOR	CODE	GENRE	STATUS
A Beautiful Mind	Sylvia Nasar	BIO001	Biographies	Issued
The Lost City of Z	David Grann	BIO002	Biographies	Not Issued
The Minds of Billy Milligan	Daniel Keyes	BIO003	Biographies	Not Issued
Napolean: A Life	Andrwe Roberts	BIO004	Biographies	Not Issued
The Passage of Power	Robert Caro	BIO005	Biographies	Not Issued
Wings of Fire	Arun Tiwari	BIO006	Biographies	Not Issued
The Man who knew Infinity	Robert Kanigel	BIO007	Biographies	Not Issued
Vivekananda: A biography	Swami Nikhilananda	BIO008	Biographies	Not Issued
Eleven Gods and a Billion Indians	Boria Majumdar	BIO009	Biographies	Not Issued
I am Neil Armstrong	Brad Meltzer	BIO010	Biographies	Not Issued
Invaluable	Maya Grossman	EDU001	Educational	Not Issued
The Privileged Poor	Anthony Abraham Jack	EDU002	Educational	Not Issued
Thrive Through the Five	Dr Jill M Siler	EDU003	Educational	Not Issued
Origin of the Species	Charles Darwin	EDU004	Educational	Issued
A Brief History of Time	Stephen Hawking	EDU005	Educational	Not Issued
Cosmos	Carl Sagan	EDU006	Educational	Not Issued
A Short History of Nearly Everything	Bill Bryson	EDU007	Educational	Not Issued
Dark Matter	Blake Crouch	EDU008	Educational	Not Issued
The Making of the Atomic Bomb	Richard Rhodes	EDU009	Educational	Not Issued
The Structure of Scientific Revolutions	Thomas Kuhn	EDU010	Educational	Not Issued
The Adventure of Augie March	Saul Bellow	FIC001	Fiction	Not Issued
American Pastoral	Philip Roth	FIC002	Fiction	Not Issued
The Assistant	Bernard Malamud	FIC003	Fiction	Issued
Charlie and the Chocolate Factory	Roald Dahl	FIC004	Fiction	Not Issued
Nineteen Eighty-Four	George Orwell	FIC005	Fiction	Not Issued
The Lord of the Rings	JRR Tolkien	FIC006	Fiction	Issued
Enders Game	Orson Scott Card	FIC007	Fiction	Not Issued
The Dune Chronicles	Frank Herbert	FIC008	Fiction	Issued
The Kingkiller Chronicles	Patrick Rothfuss	FIC009	Fiction	Not Issued
Harry Potter and the Deathly Hallows	JK Rowling	FIC010	Fiction	Not Issued
The Bell Jar	Sylvia Plath	NOV001	Novel	Not Issued
The Great Gatsby	F Scott Fitzgerald	NOV002	Novel	Not Issued
Speak	Laurie Halse Anderson	NOV003	Novel	Not Issued
A Suitable Boy	Vikram Seth	NOV004	Novel	Not Issued
The Inheritance of Loss	Kiran Desai	NOV005	Novel	Not Issued
The White Tiger	Aravind Adiga	NOV006	Novel	Not Issued
A Fine Balance	Rohinton Mistry	NOV007	Novel	Not Issued
To Kill a Mockingbird	Harper Lee	NOV008	Novel	Not Issued
Ulysses	James Joyce	NOV009	Novel	Not Issued
Beloved	Toni Morrison	NOV010	Novel	Not Issued

1.4. Go Back

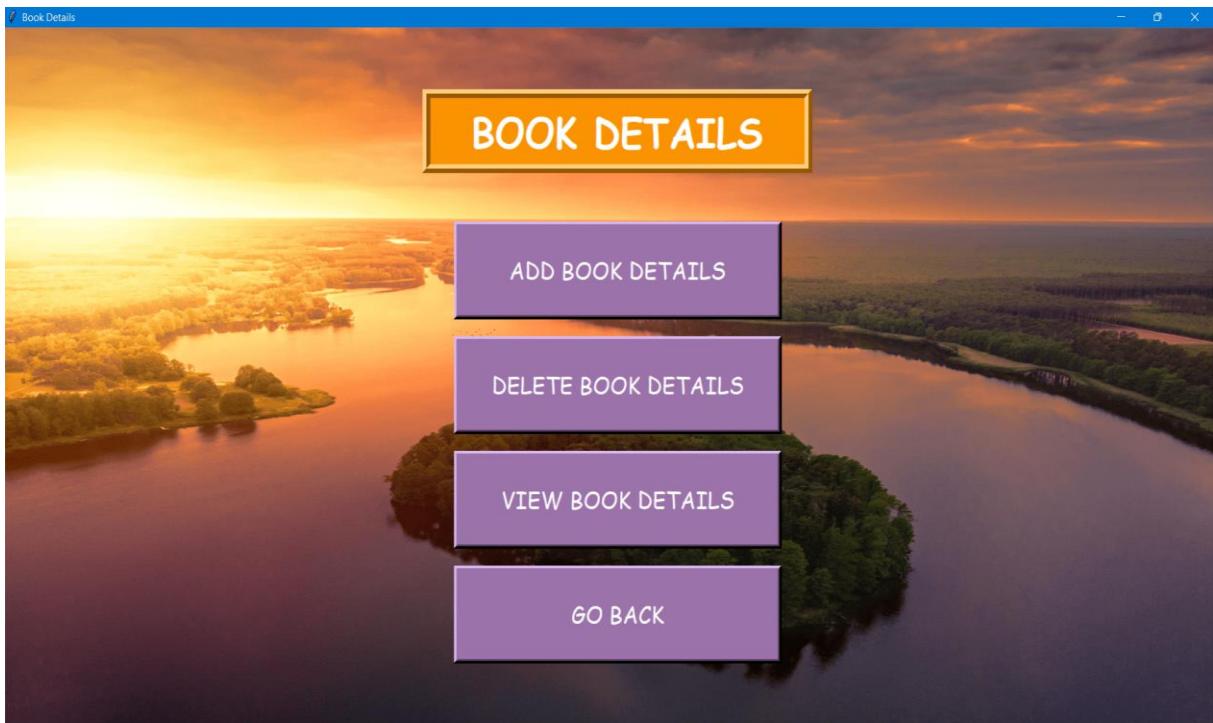
2. Admin

2.1. Login:

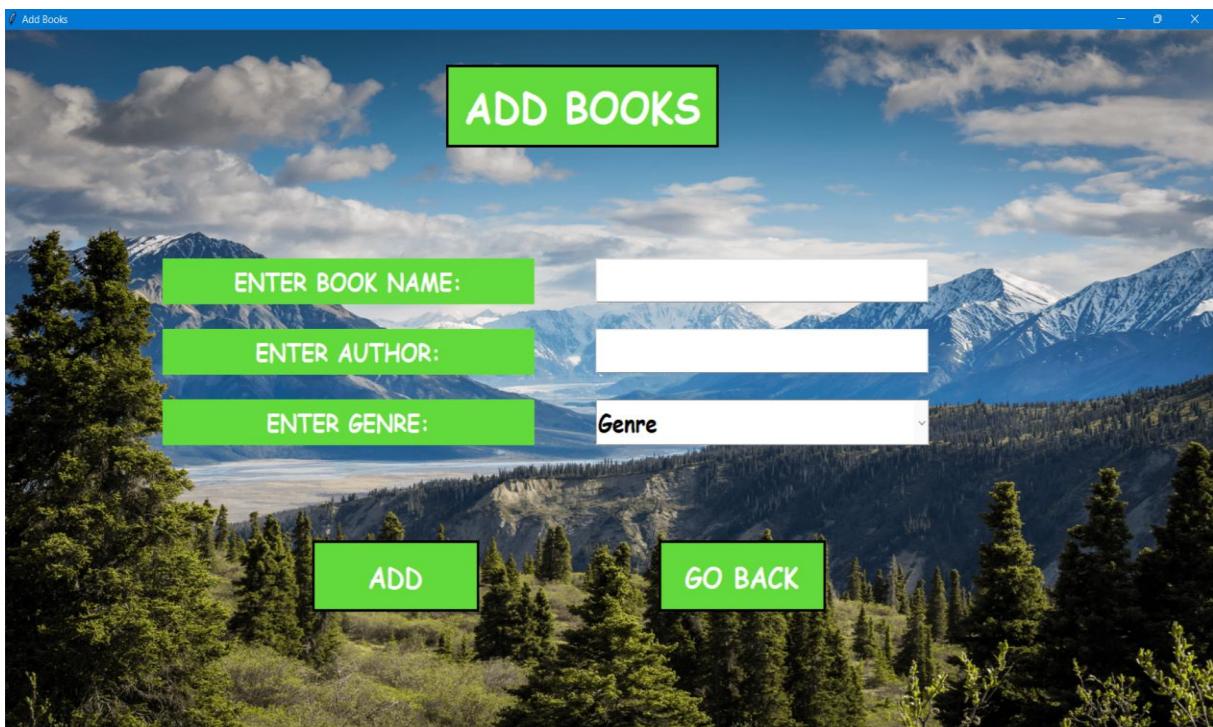




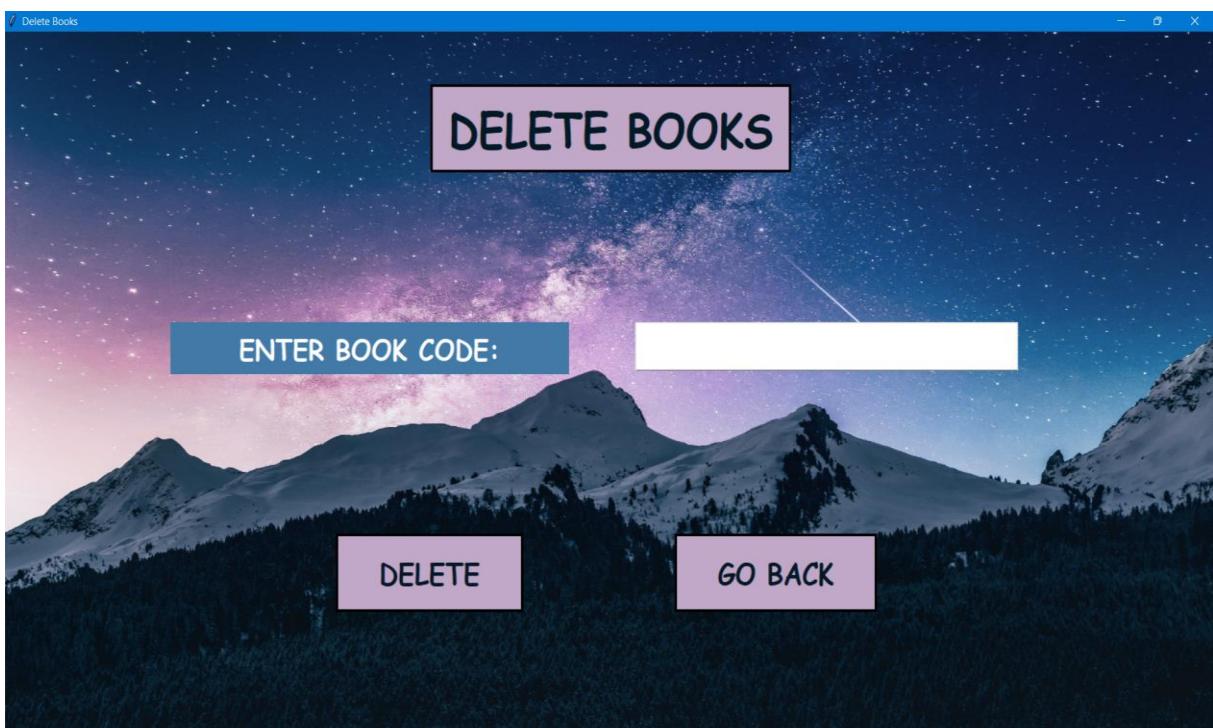
2.2. Book Details:



2.2.1. Add Book Details:



2.2.2. Delete Book Details:



2.2.3. View Book Details:

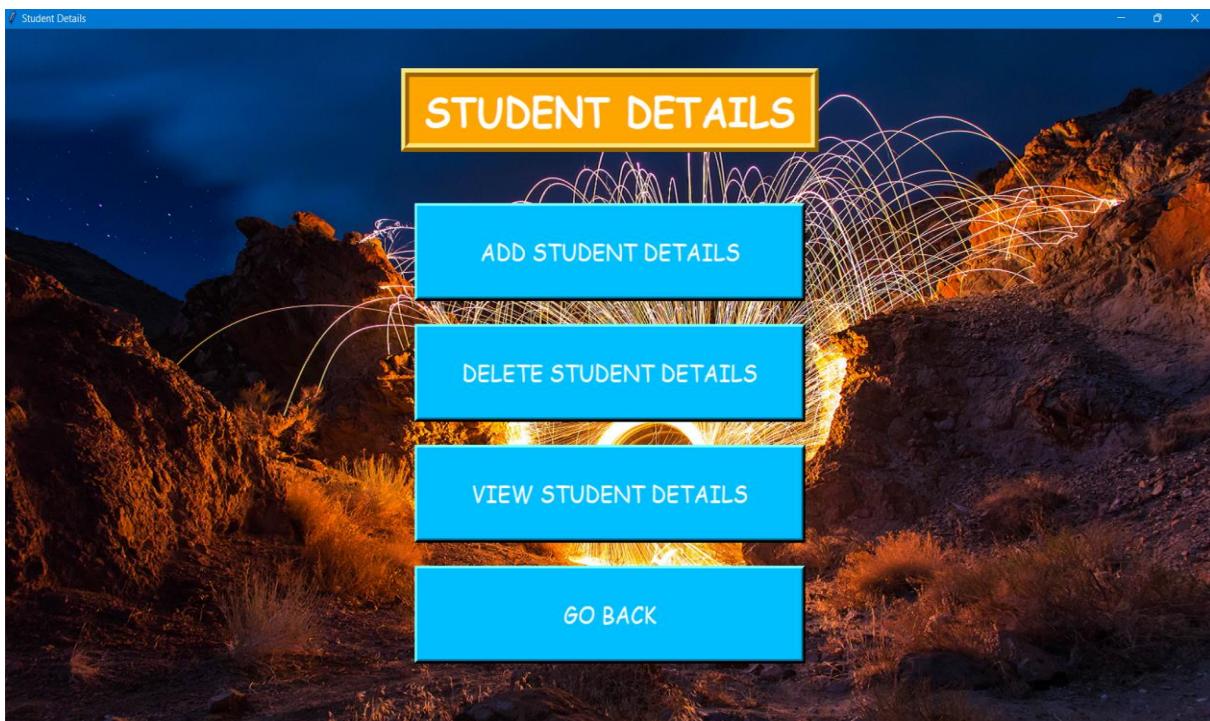
BOOK NAME	AUTHOR	CODE	GENRE	STATUS
A Beautiful Mind	Sylvia Nasar	BIO001	Biographies	Issued
The Lost City of Z	David Grann	BIO002	Biographies	Not Issued
The Minds of Billy Milligan	Daniel Keyes	BIO003	Biographies	Not Issued
Napoleon: A Life	Andrwe Roberts	BIO004	Biographies	Not Issued
The Passage of Power	Robert Caro	BIO005	Biographies	Not Issued
Wings of Fire	Arun Tiwari	BIO006	Biographies	Not Issued
The Man who knew Infinity	Robert Kanigel	BIO007	Biographies	Not Issued
Vivekananda: A biography	Swami Nikhilananda	BIO008	Biographies	Not Issued
Eleven Gods and a Billion Indians	Boria Majumdar	BIO009	Biographies	Not Issued
I am Neil Armstrong	Brad Meltzer	BIO010	Biographies	Not Issued
Invaluable	Maya Grossman	EDU001	Educational	Not Issued
The Privileged Poor	Anthony Abraham Jack	EDU002	Educational	Not Issued
Thrive Through the Five	Dr Jill M. Siler	EDU003	Educational	Not Issued
Origin of the Species	Charles Darwin	EDU004	Educational	Issued
A Brief History of Time	Stephen Hawking	EDU005	Educational	Not Issued
Cosmos	Carl Sagan	EDU006	Educational	Not Issued
A Short History of Nearly Everything	Bill Bryson	EDU007	Educational	Not Issued
Dark Matter	Blake Crouch	EDU008	Educational	Not Issued
The Making of the Atomic Bomb	Richard Rhodes	EDU009	Educational	Not Issued
The Structure of Scientific Revolutions	Thomas Kuhn	EDU010	Educational	Not Issued
The Adventure of Augie March	Saul Bellow	FIC001	Fiction	Not Issued
American Pastoral	Philip Roth	FIC002	Fiction	Not Issued
The Assistant	Bernard Malamud	FIC003	Fiction	Issued
Charlie and the Chocolate Factory	Roald Dahl	FIC004	Fiction	Not Issued
Nineteen Eighty-Four	George Orwell	FIC005	Fiction	Not Issued
The Lord of the Rings	JRR Tolkien	FIC006	Fiction	Issued
Enders Game	Orson Scott Card	FIC007	Fiction	Not Issued
The Dune Chronicles	Frank Herbert	FIC008	Fiction	Issued
The Kingkiller Chronicles	Patrick Rothfuss	FIC009	Fiction	Not Issued
Harry Potter and the Deathly Hallows	JK Rowling	FIC010	Fiction	Not Issued
The Bell Jar	Sylvia Plath	NOV001	Novel	Not Issued
The Great Gatsby	F Scott Fitzgerald	NOV002	Novel	Not Issued
Speak	Laurie Halse Anderson	NOV003	Novel	Not Issued
A Suitable Boy	Vikram Seth	NOV004	Novel	Not Issued
The Inheritance of Loss	Kiran Desai	NOV005	Novel	Not Issued
The White Tiger	Aravind Adiga	NOV006	Novel	Not Issued
A Fine Balance	Rahinton Mistry	NOV007	Novel	Not Issued
To Kill a Mockingbird	Harper Lee	NOV008	Novel	Not Issued
Ulysses	James Joyce	NOV009	Novel	Not Issued
Beloved	Toni Morrison	NOV010	Novel	Not Issued

2.2.4. Go Back

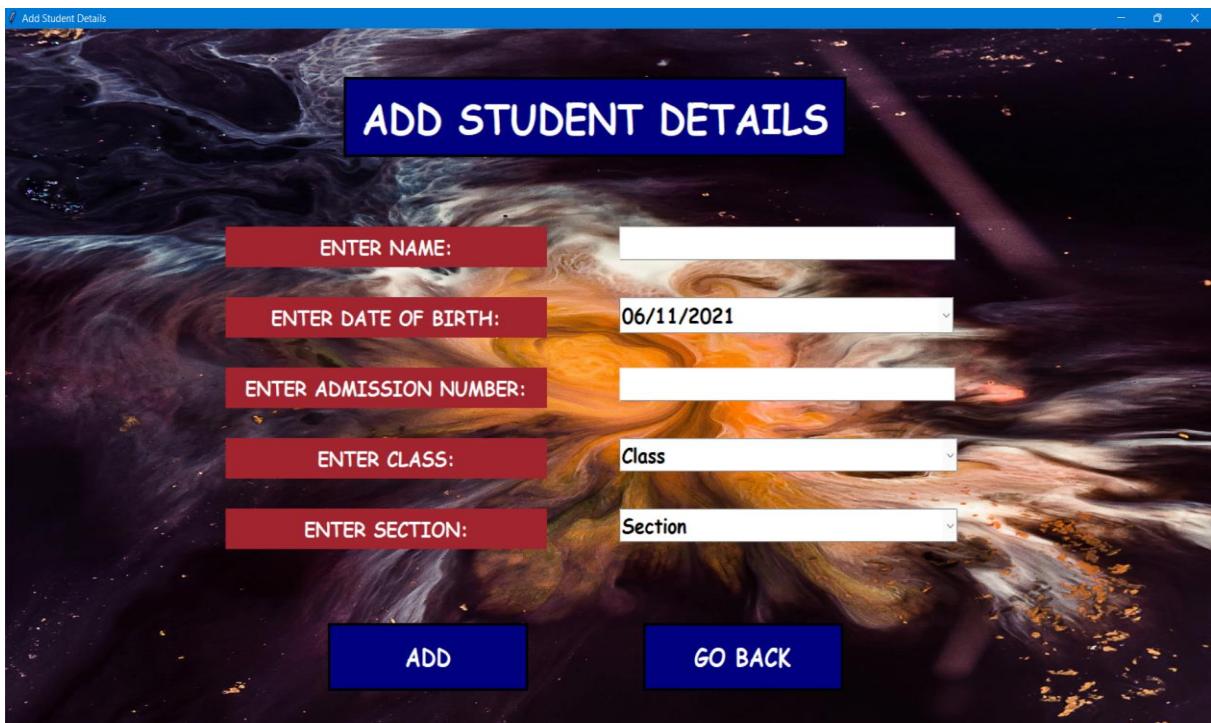
2.3. View Book Records:

NAME	ADMISSION NUMBER	BOOK NAME	BOOK CODE	DATE OF ISSUE	DATE OF RETURN	DUUE DATE	ISSUE TIME
Kanak Joshi	I-08/5068	A Beautiful Mind	BIO001	06/11/2021	-	20/11/2021	22:25:57
Simron Kaur	I-14/10776	The Assistant	FIC003	06/11/2021	-	20/11/2021	22:24:42
Ridhi Singh	I-14/10111	The Lord of the Rings	FIC006	06/11/2021	-	20/11/2021	22:23:35
Aditya Kamal	I-15/8889	The Dune Chronicles	FIC008	06/11/2021	-	20/11/2021	22:23:00
Aladdin Singh	I-16/10089	The Bell Jar	NOV001	06/11/2021	06/11/2021	20/11/2021	22:21:04
Anushi Kapoor	I-14/10554	Origin of the Species	Edu004	06/11/2021	-	20/11/2021	22:19:53
Avi Gupta	I-14/10100	The Minds of Billy Milligan	BIO003	06/11/2021	06/11/2021	20/11/2021	22:18:59

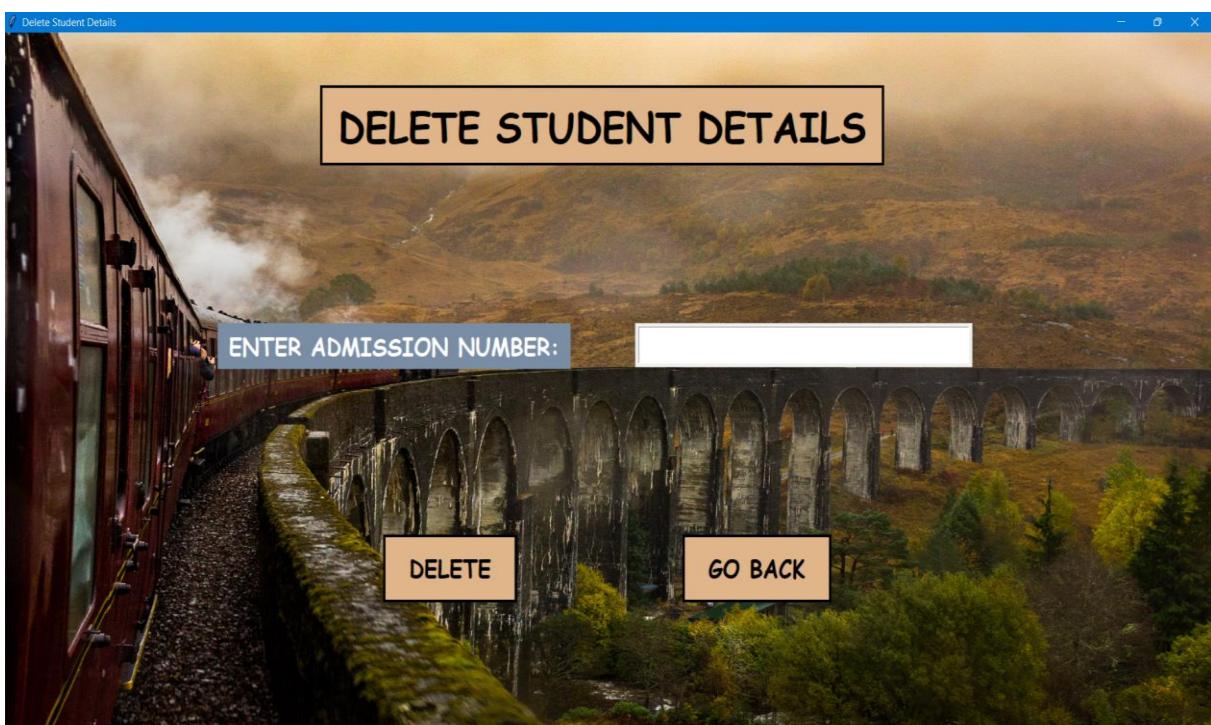
2.4. Student Details:



2.4.1. Add Student Details:



2.4.2. Delete Student Details:

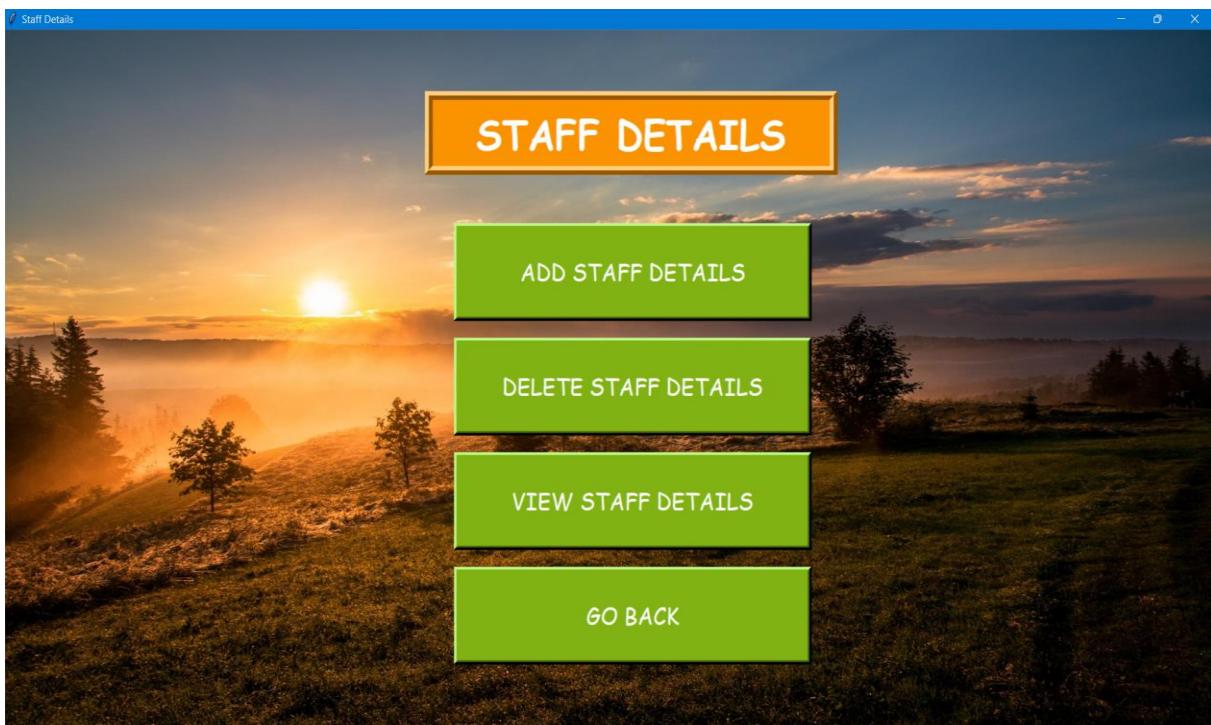


2.4.3. View Student Details:

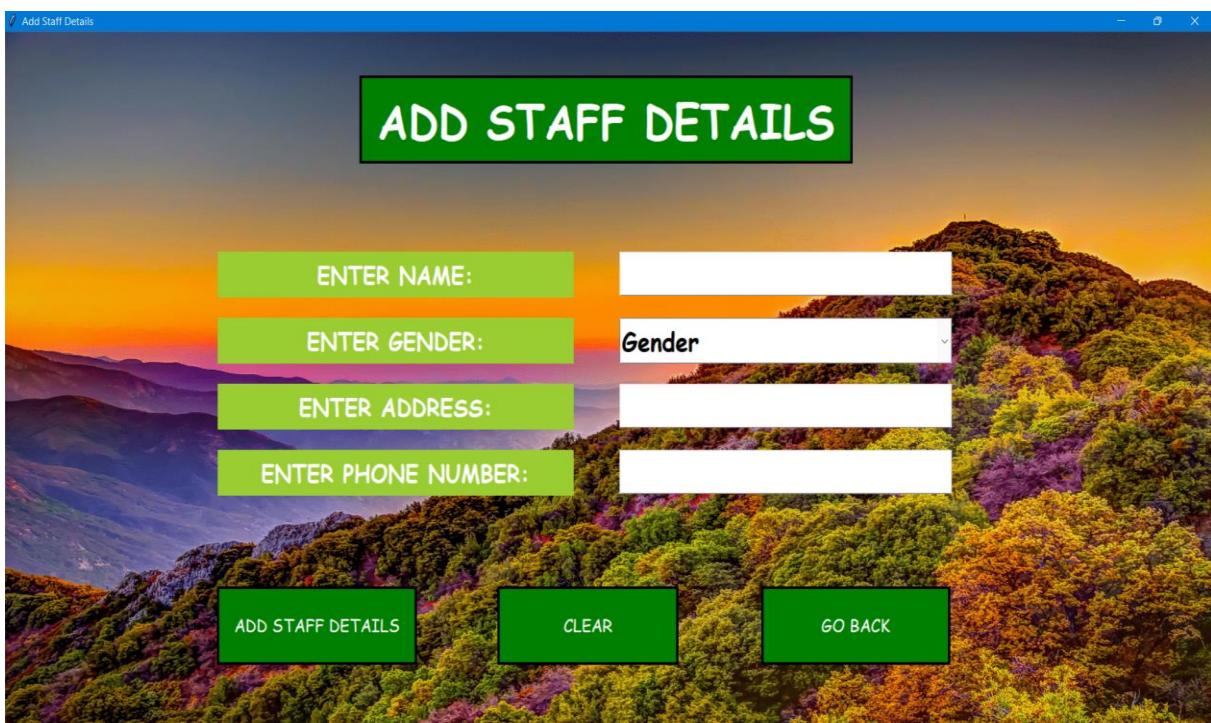
NAME	DATE OF BIRTH	ADMISSION NO.	CLASS	SECTION
Kanak Joshi	2004-06-17	I-08/5068	12	A
Rohan Yadav	2004-08-02	I-09/5023	12	A
Nivedita Sharma	2005-04-19	I-09/5787	11	H
Shreya Jain	2004-01-09	I-09/5789	12	E
Poulami Bhattacharjee	2005-11-19	I-09/5790	11	B
Anisha Singh	2005-10-29	I-09/7249	11	A
Sam Singh	2006-06-12	I-10/7866	10	A
Vishal Jaiswal	2005-04-11	I-10/7896	11	A
Salman Kapoor	2003-07-27	I-11/8906	12	D
Aditya Kumar	2004-05-23	I-11/8976	12	G
Anuj Singh	2007-11-16	I-12/8790	09	C
Ramesh Tripathi	2005-04-17	I-13/9761	11	I
Joel Joseph	2005-12-11	I-13/9799	11	C
Jitendra Yadav	2005-01-10	I-13/9888	11	D
Avi Gupta	2004-02-05	I-14/10100	12	H
Ridhi Singh	2004-12-20	I-14/10111	12	E
Aastha Bhat	2004-12-22	I-14/10122	12	K
Sheena Bajaj	2005-01-09	I-14/10444	11	A
Kritika Singh	2004-11-09	I-14/10467	12	J
Arushi Kapkoti	2004-07-27	I-14/10554	12	E
Simran Kaur	2004-12-18	I-14/10776	12	C
Sandeep Kumar	2006-09-12	I-15/10281	10	G
Rahul Modi	2008-07-08	I-15/10561	08	C
Pankaj Sharma	2008-09-30	I-15/10781	11	F
Aditya Kamal	2004-03-23	I-15/8889	12	D
Shreya Chauhan	2006-01-09	I-15/9111	10	E
Jasmine Fernandez	2006-01-13	I-15/9123	10	C
Armaan Kohli	2005-01-18	I-15/9770	11	F
Jayant Yadav	2005-01-10	I-15/9777	11	E
Bhavya Jha	2005-09-13	I-15/9778	12	A
Tony Kakkar	2005-01-15	I-15/9779	12	G
Bhavdeep Singh	2004-03-14	I-15/9999	12	F
Lakhshya Singh	2004-07-20	I-16/10059	12	D
Aladdin Singh	2004-07-23	I-16/10089	12	E
Raveena Kaur	2004-10-20	I-18/12229	12	G
Ajey Nagar	2004-01-10	I-18/12620	12	E
Ajit Singh	2005-06-18	I-18/12689	11	I
Rashmi Pillai	2009-02-28	I-19/16789	07	G
Mahesh Yadav	2008-08-17	I-19/16879	08	A
Mona Kaur	2009-12-16	I-19/16889	07	F

2.4.4. Go Back

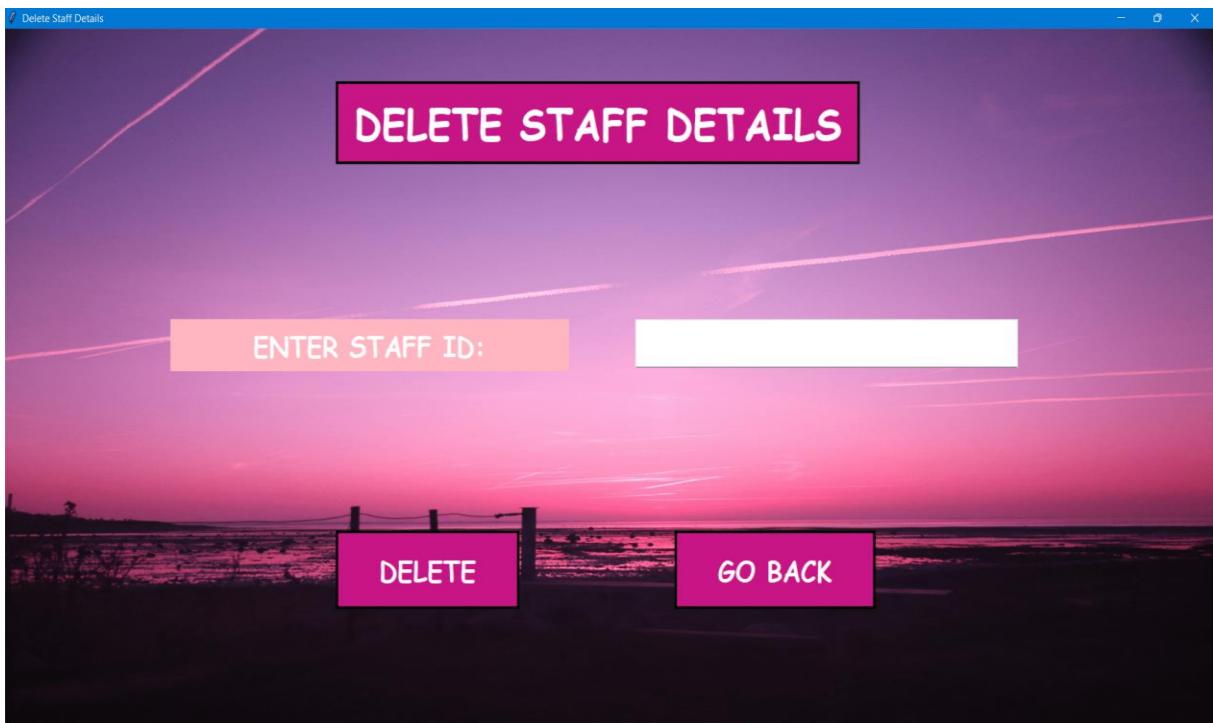
2.5. Staff Details:



2.5.1. Add Staff Details:



2.5.2. Delete Staff Details:



2.5.3. View Staff Details:

NAME	STAFF ID	GENDER	ADDRESS	CONTACT NUMBER
Akansha Singh	1	F	Z-420, National Colony	9384759708
Mridula Tiwari	2	F	F-397, Reserved Colony	9283748576
Suraj Tyagi	3	M	D-410, Gundanagar	9946529765
Lalu Prasad	4	M	E-4102, Shiva Colony	9284859673
Bir Gupta Anand	5	M	M-387, Shipra Rivera	9346847875

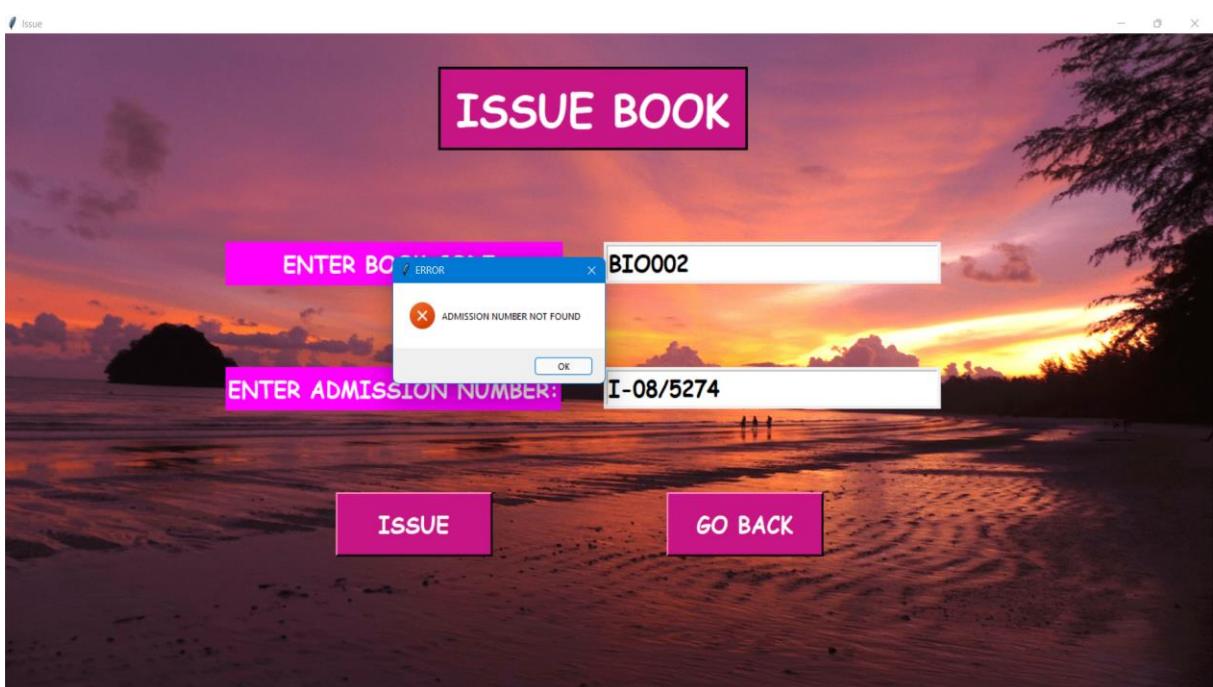
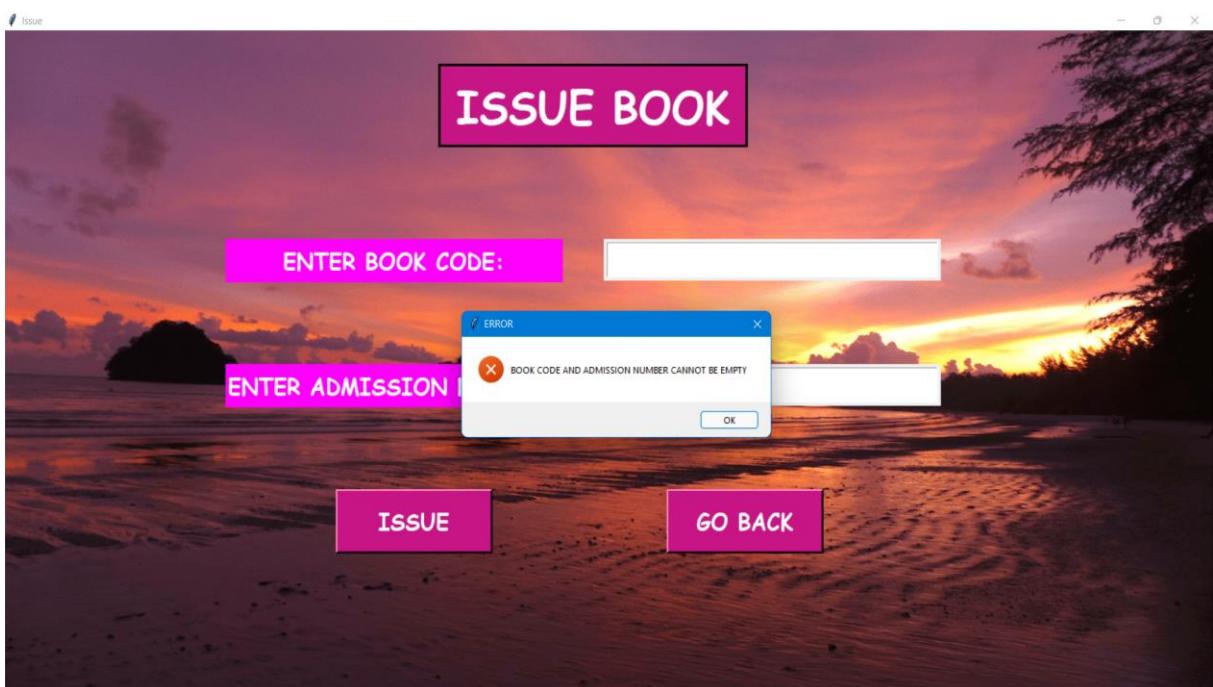
2.5.4. Go Back

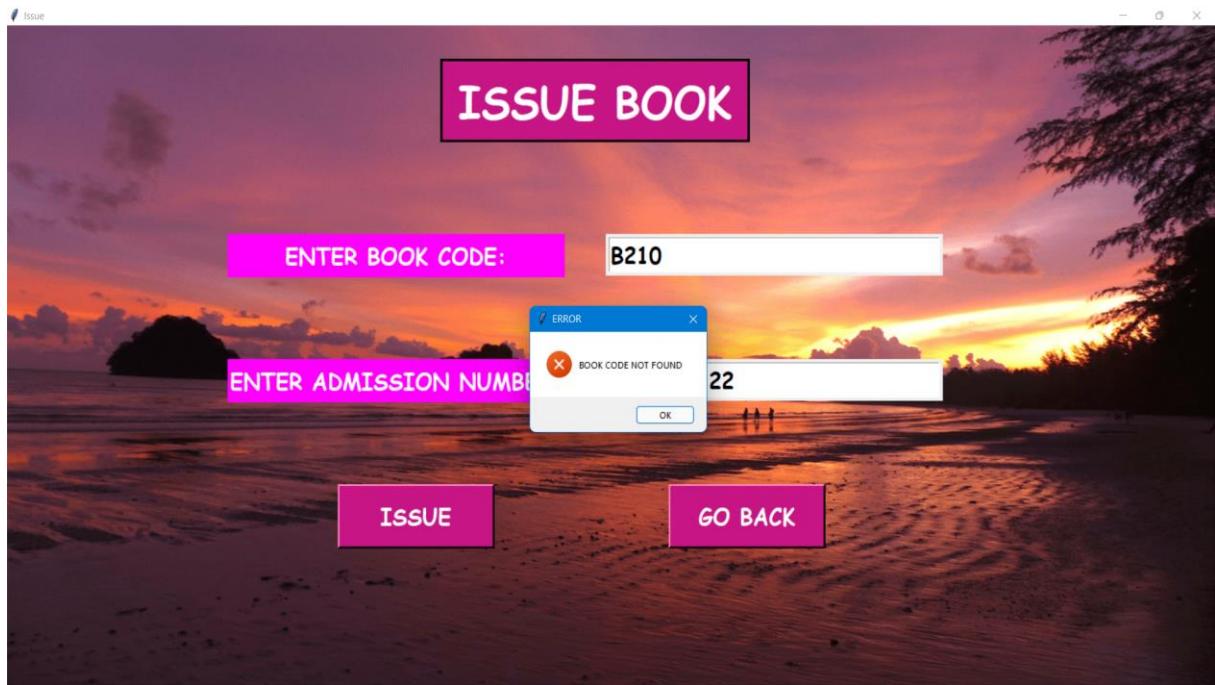
2.6. Go Back

3. Exit

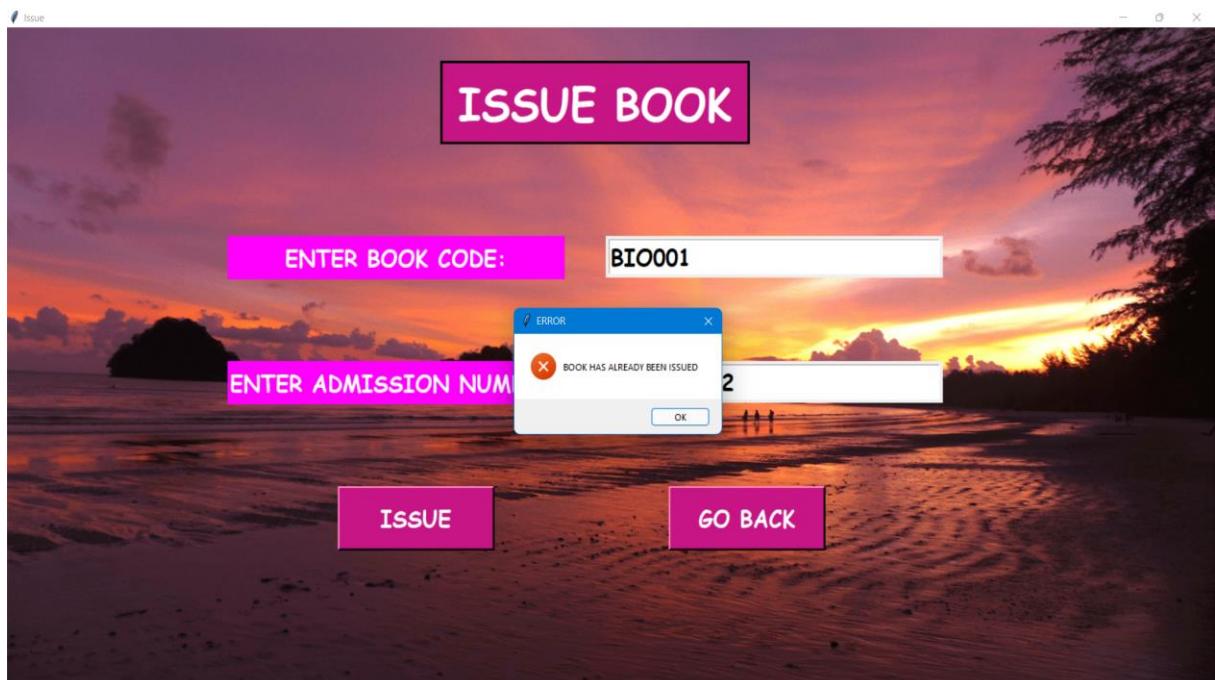
INPUT VALIDATION, RESTRICTIONS, AND ERROR HANDLING

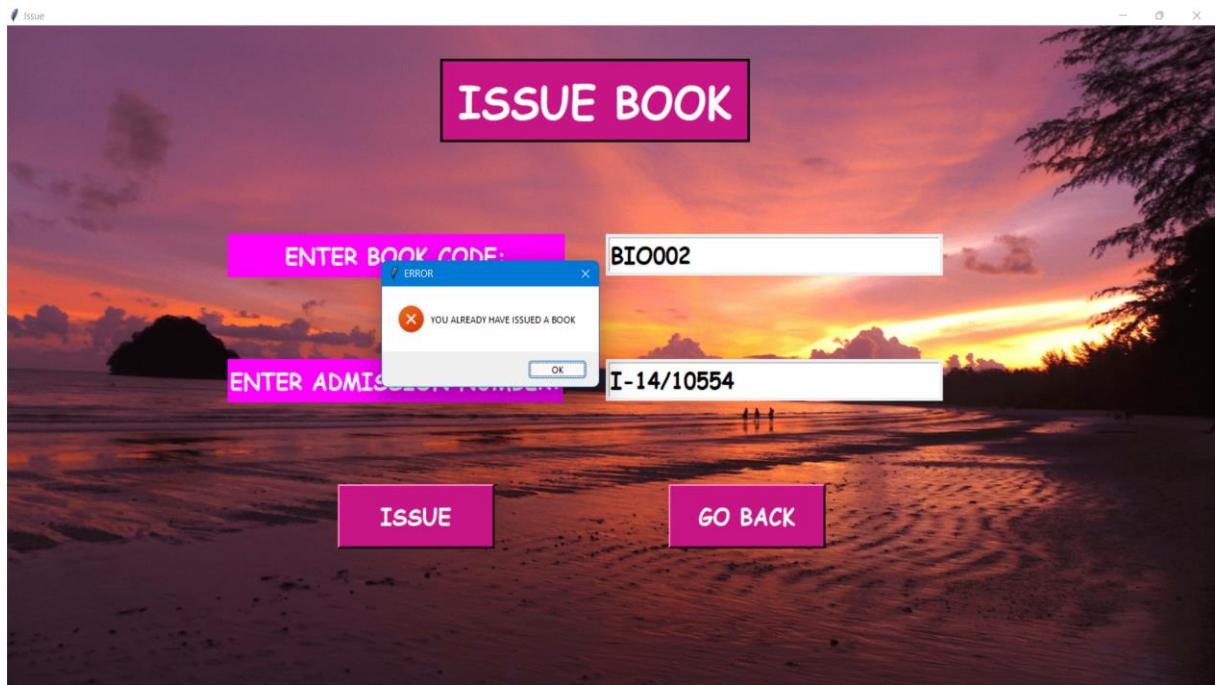
- In Issue Book, if we leave the text boxes empty or if we enter a book code or admission number which doesn't exist, it shows the following error message:



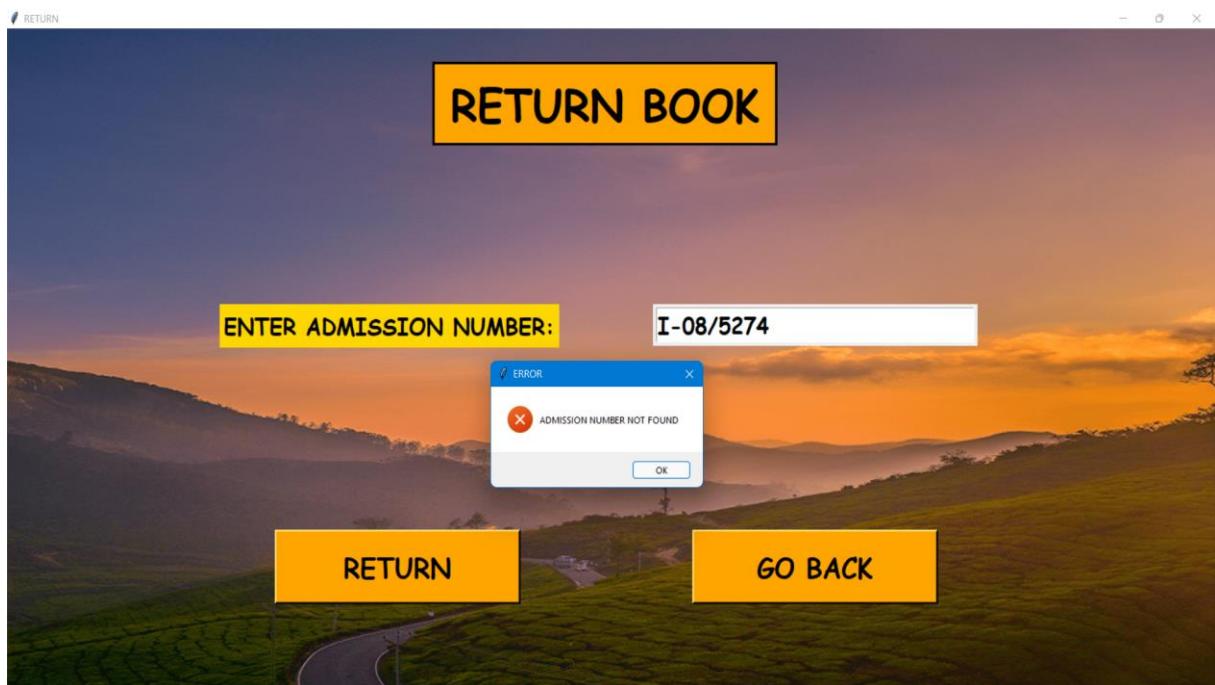


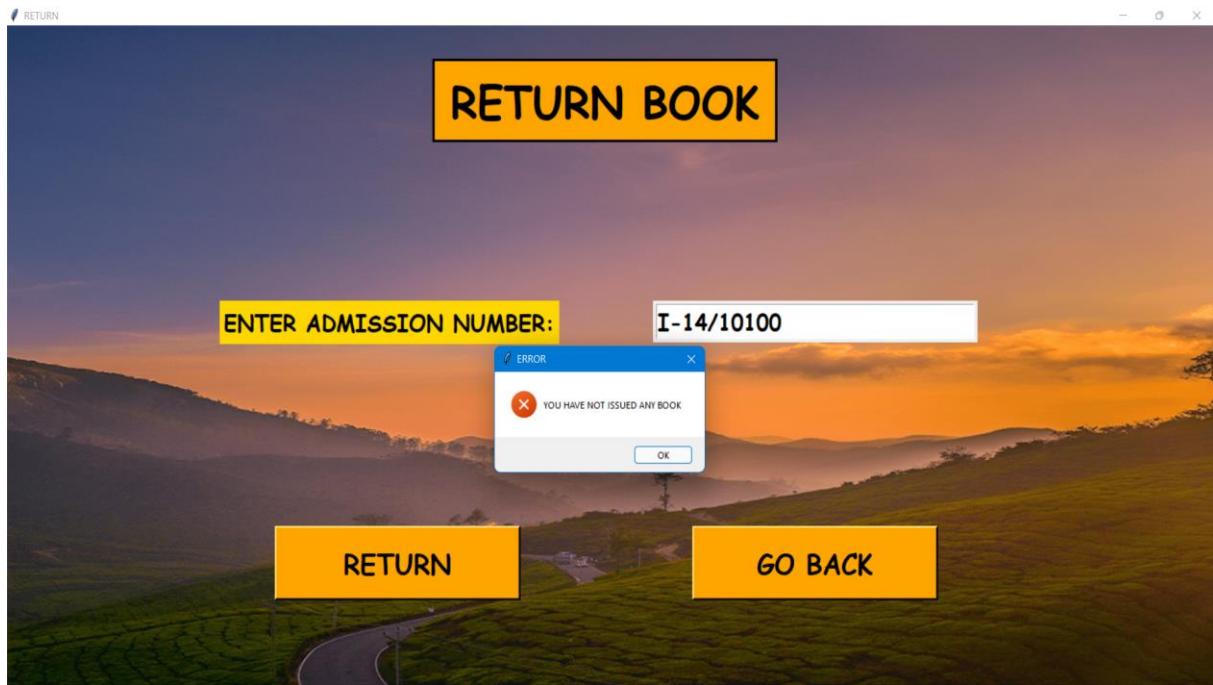
- In Issue Book, if we try to issue an already issued book or if we have already issued a book and try to issue another one, it will display the appropriate error message:



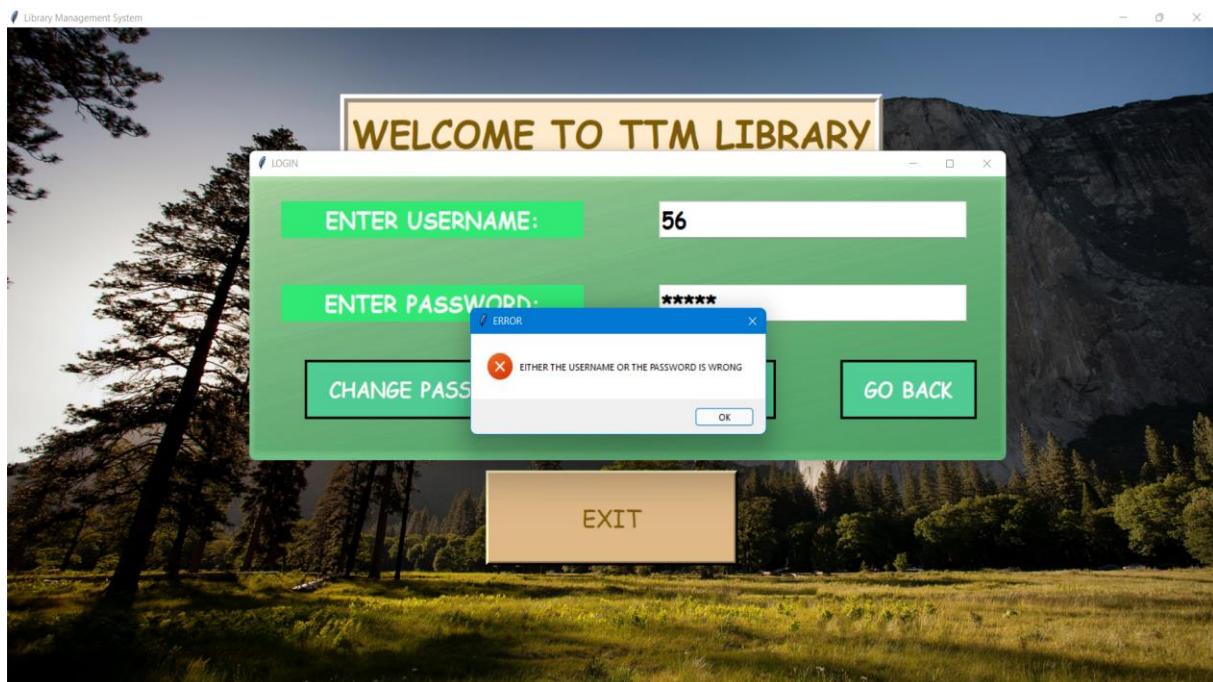


- In Return Book, if we enter an admission number which doesn't exist or if we have not issued any book, it displays the appropriate error message:

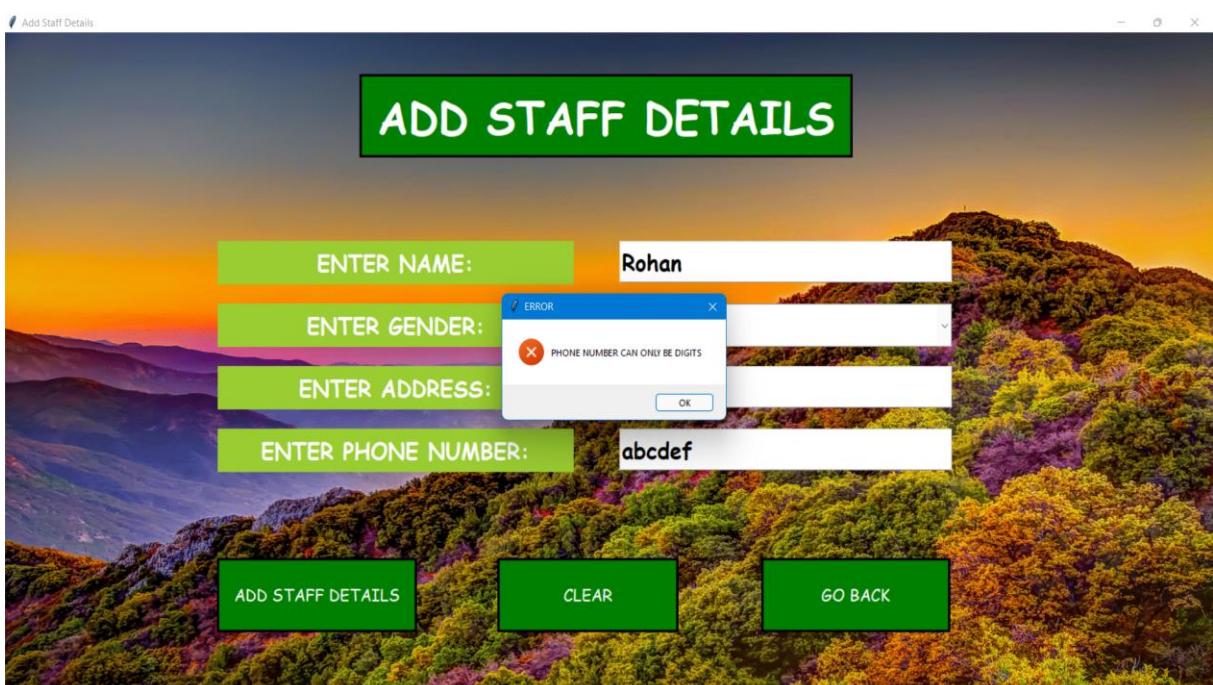
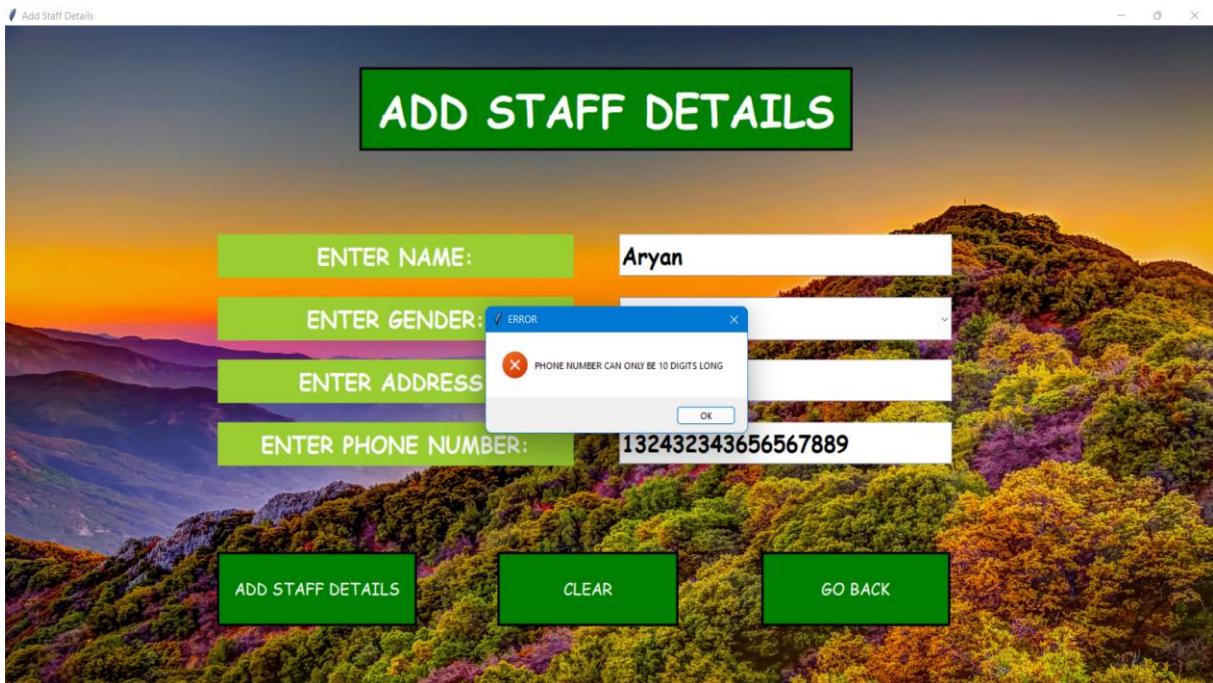




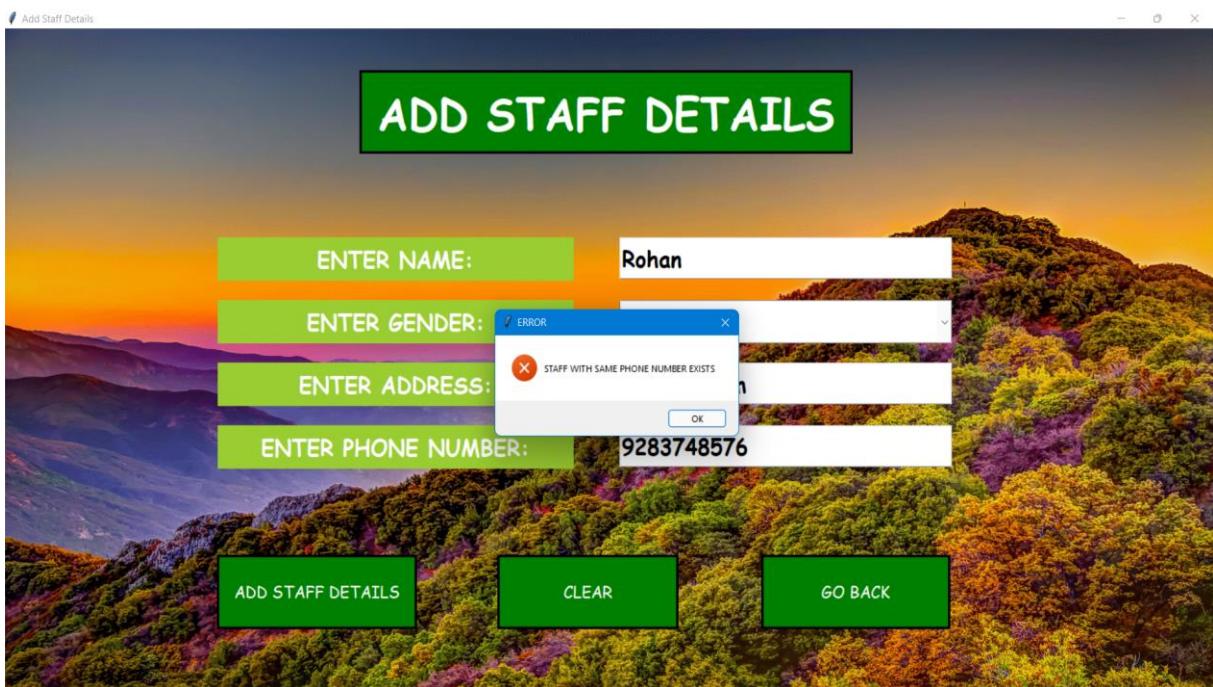
- When we try to access Admin Mode, if we enter a wrong username or password, then it will show the following error message:



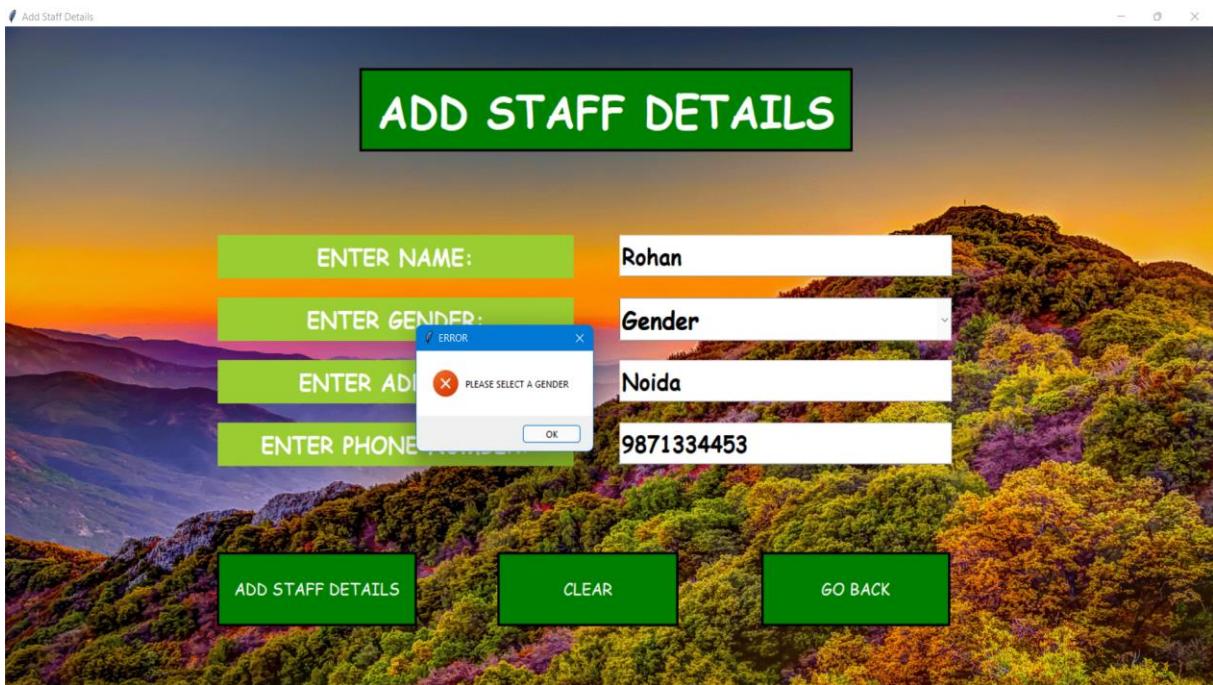
- In Add Staff Details, if we enter mobile number longer than 10 digits or if we don't use digits, then it will show the following error message:



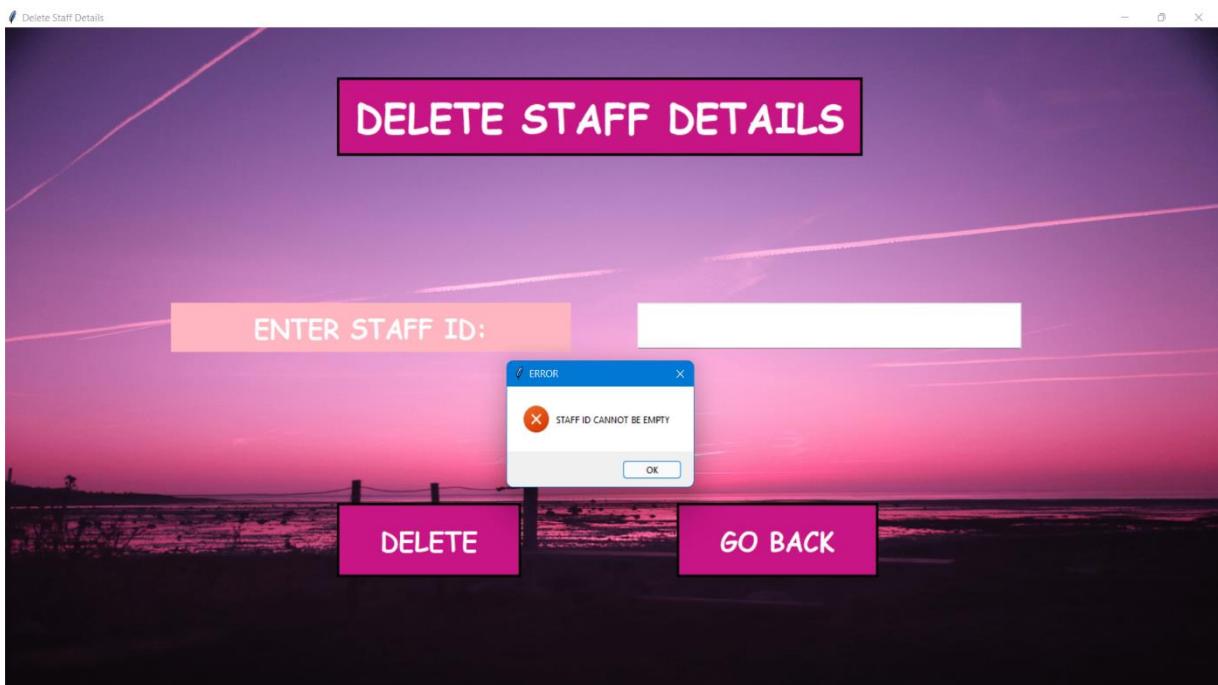
- If we try to add staff which has the same phone number as an already existing staff, it will display the appropriate error message:



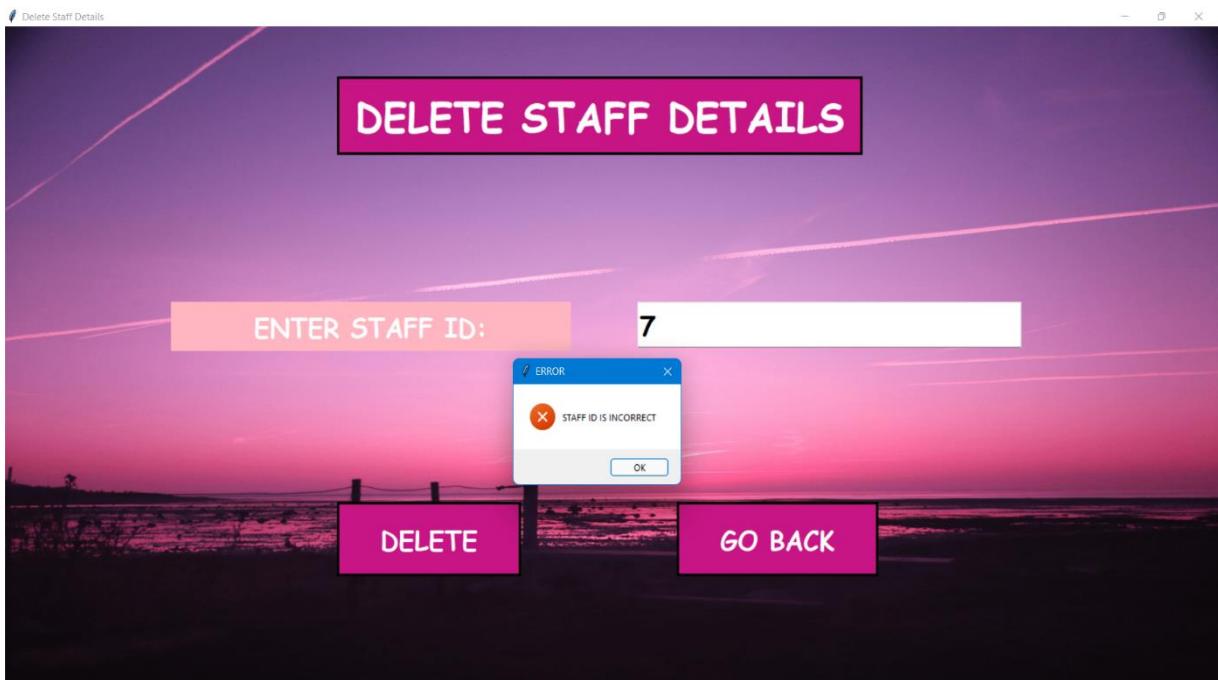
- In Add Staff Details, if we don't select any gender, then it will show the following error message:



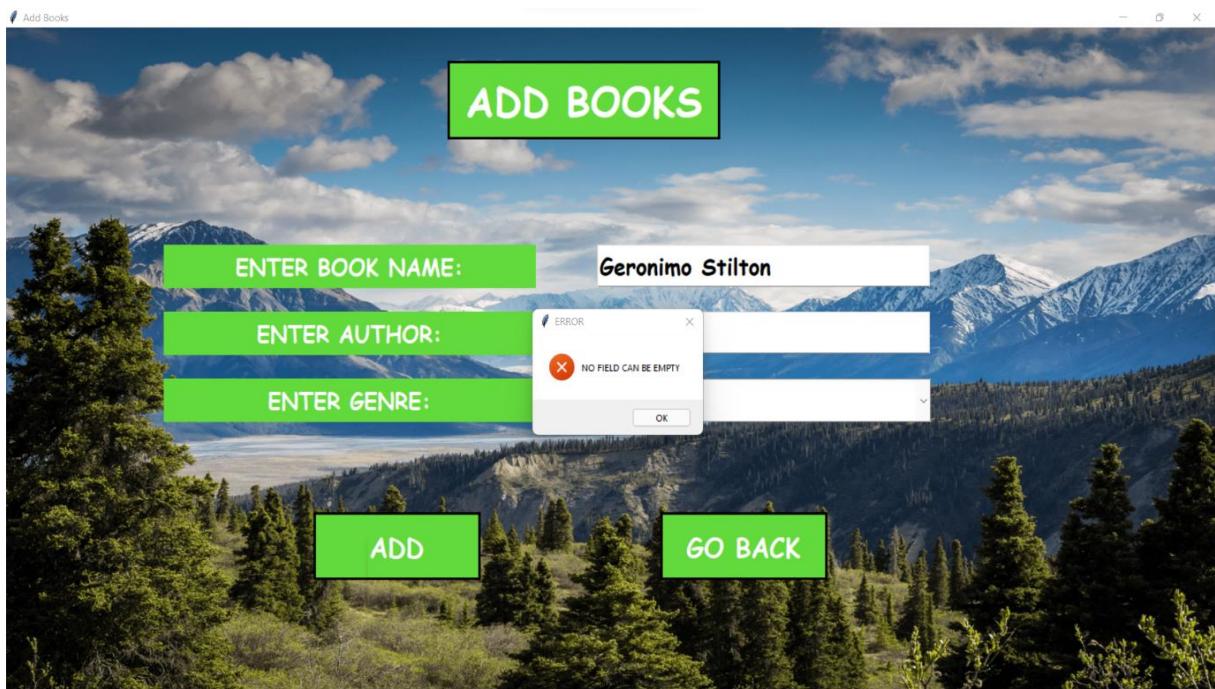
- In Delete Staff Details, if the text box is left blank, then it will show the following error message:



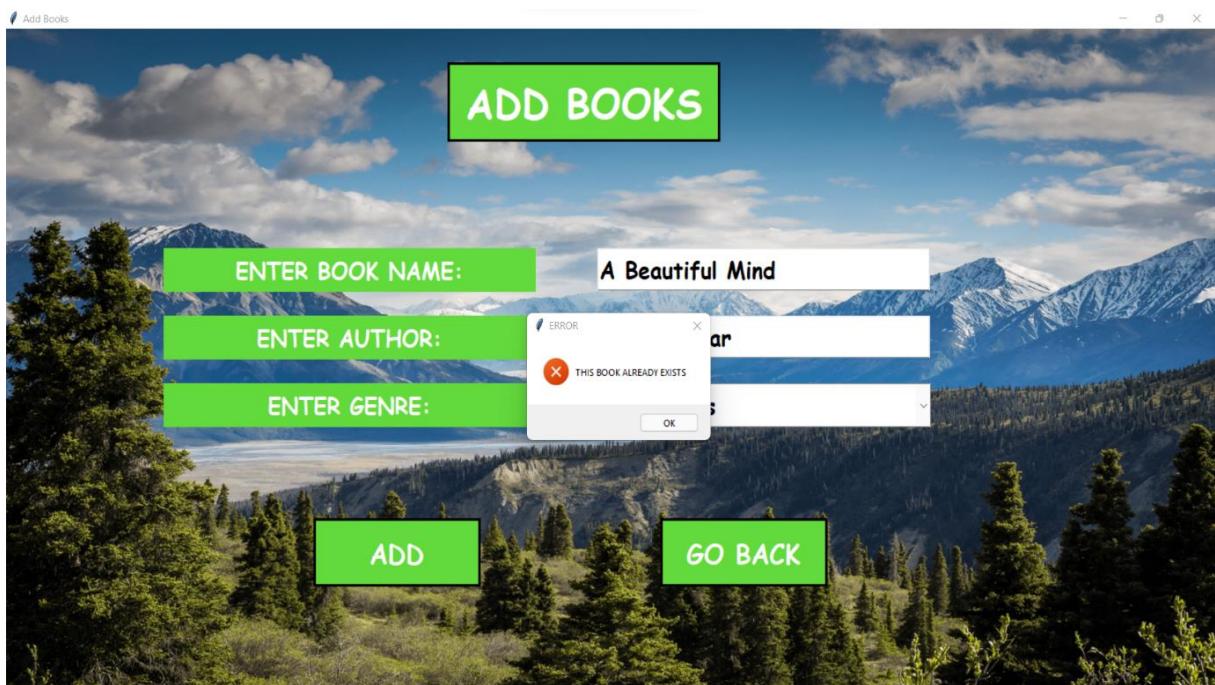
- In Delete Staff Details, if an unknown staff-ID is entered, then it will show the following error message:



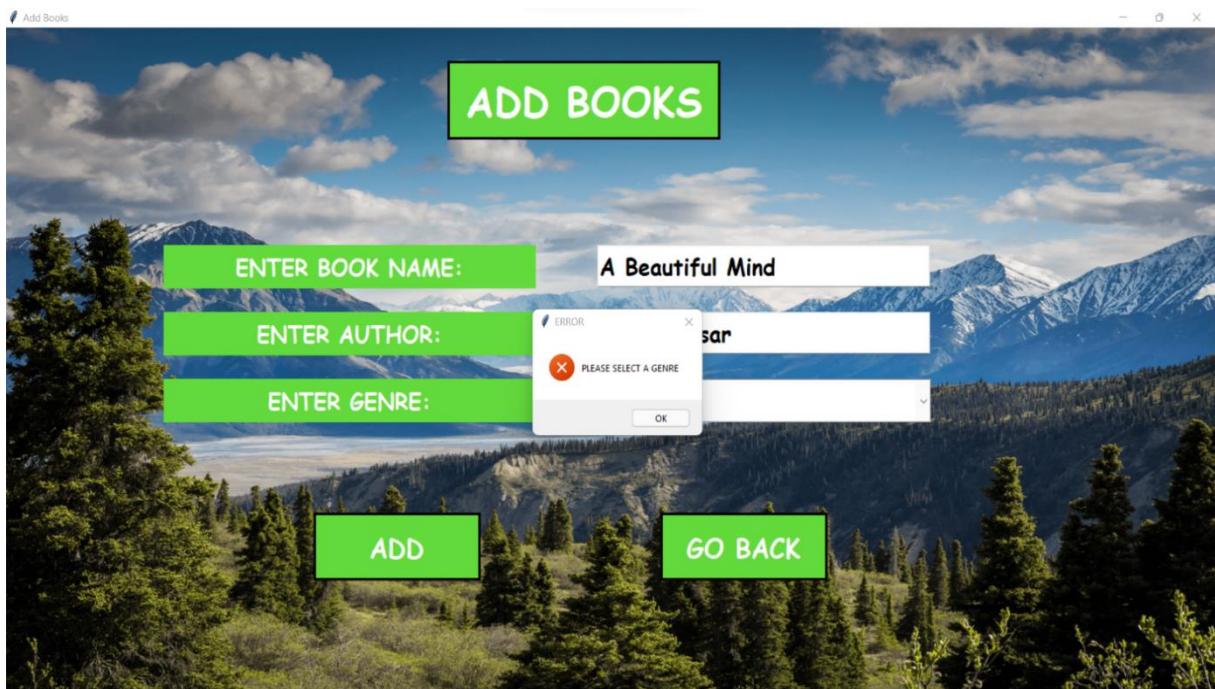
- In Add Book Details, if any box is left blank it will show the following error:



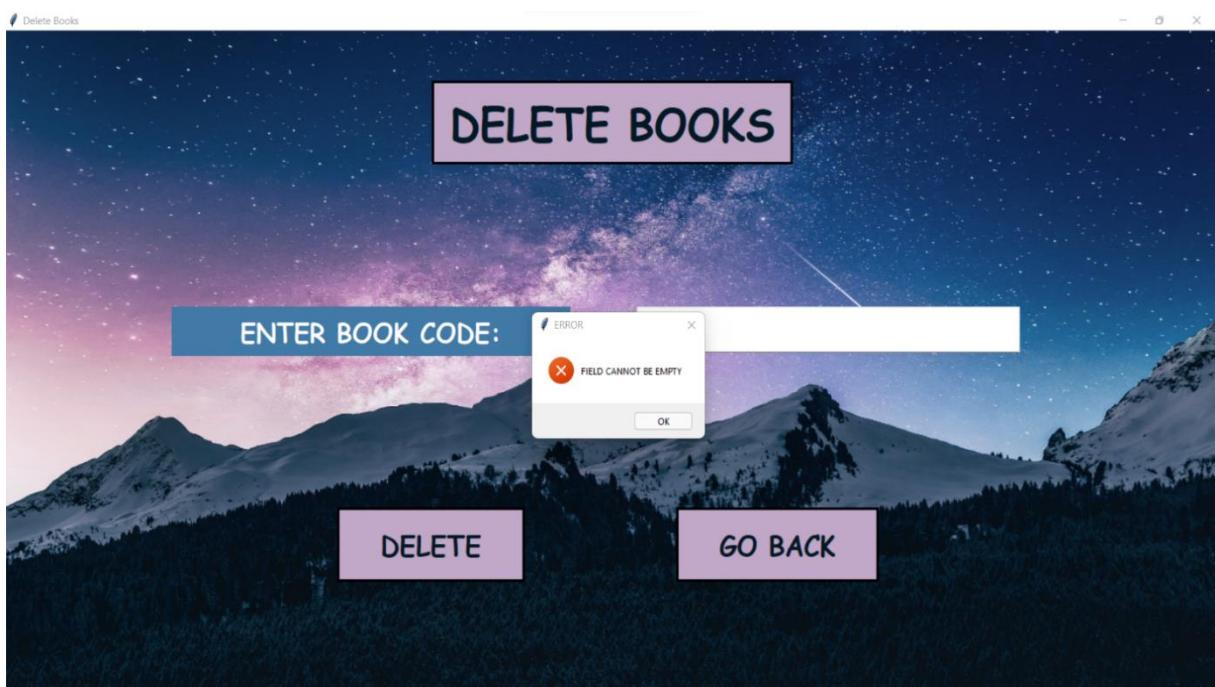
- In Add Book Details, if an already existing book is entered it will show the following error:



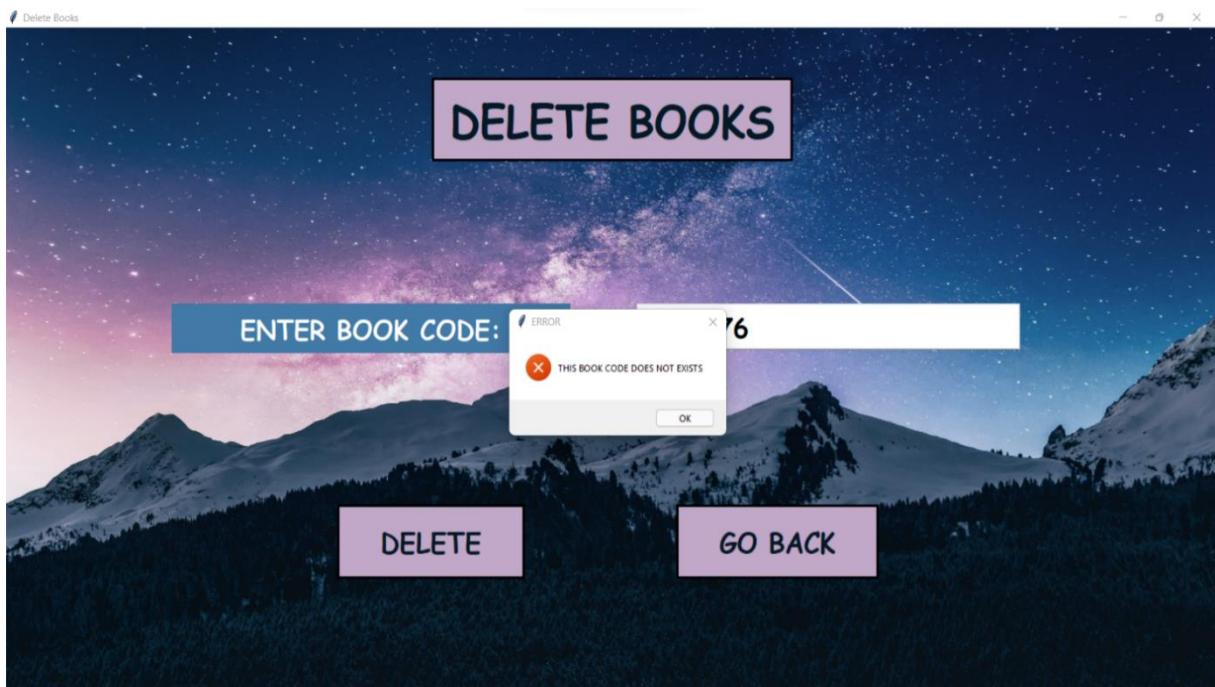
- In Add Book Details, if we enter book name and author, but do not choose genre it will show the following error:



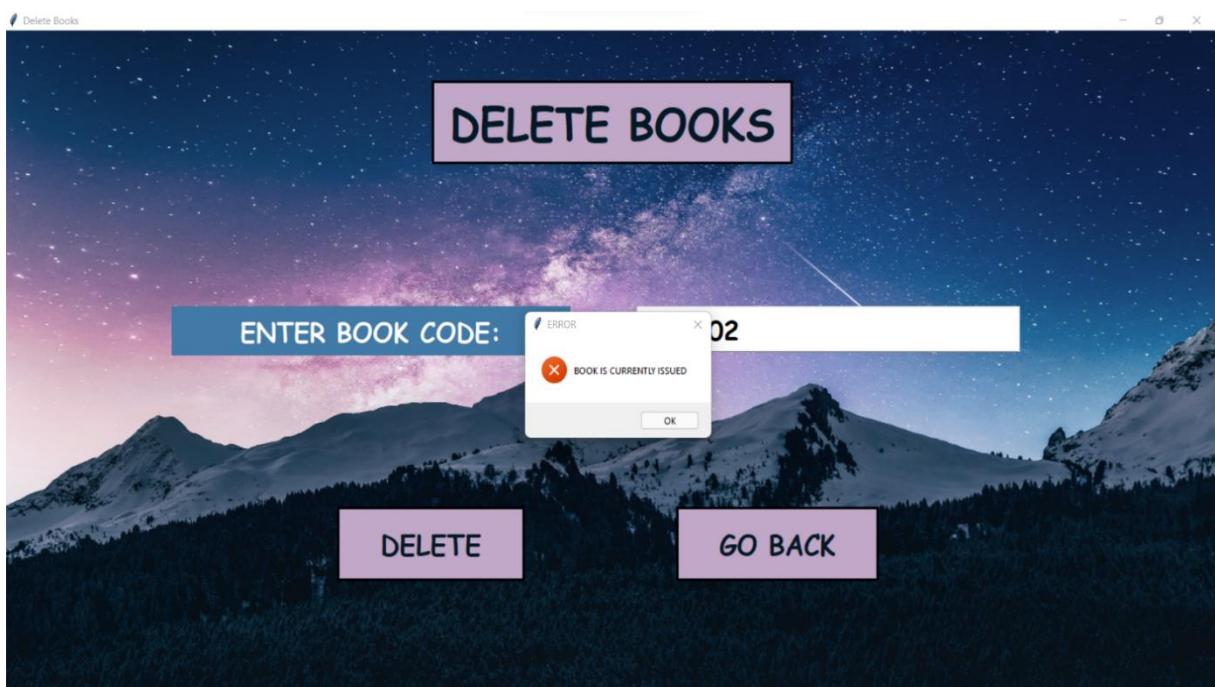
- In Delete Book Details, if box is left blank it will show the following error:



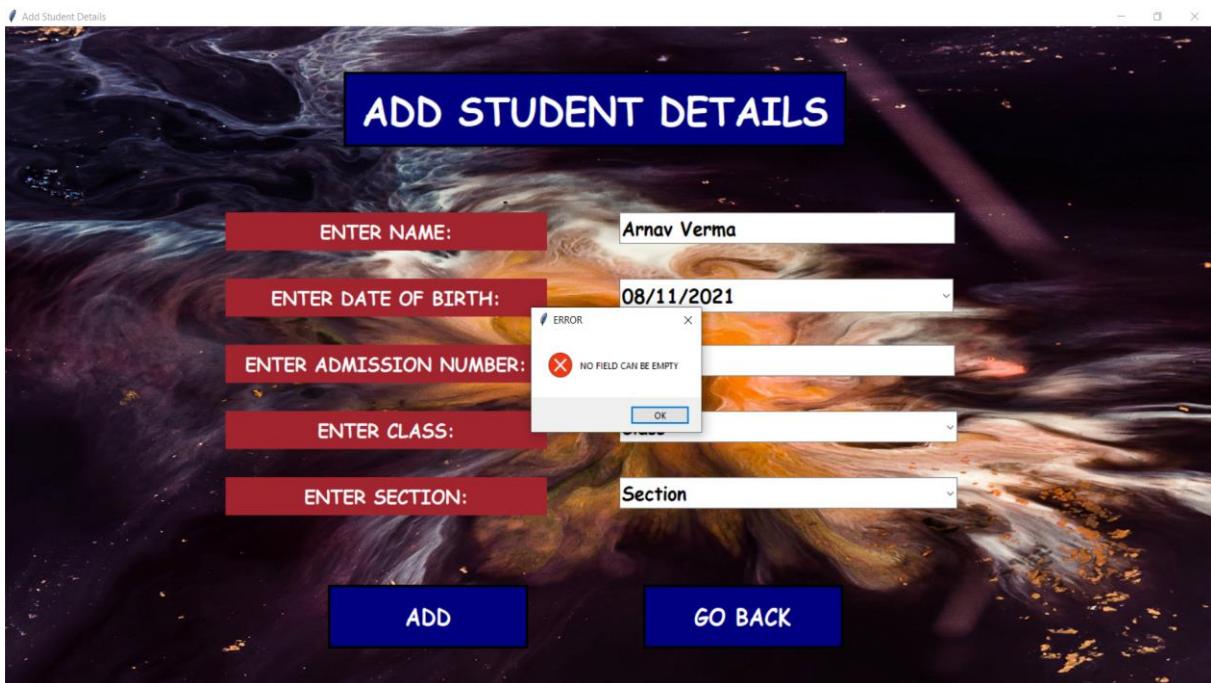
- In Delete Book Details, if an unknown book code is entered it will show the following error:



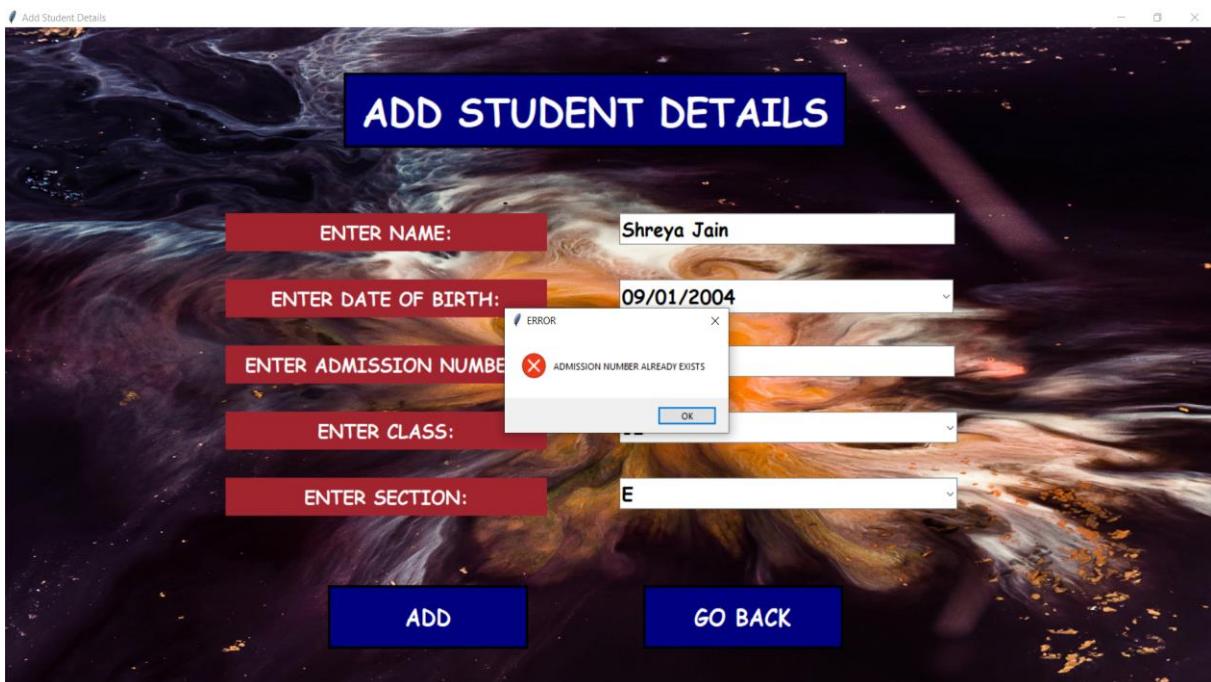
- In Delete Book Details, if a currently issued book code is entered it will show the following error:



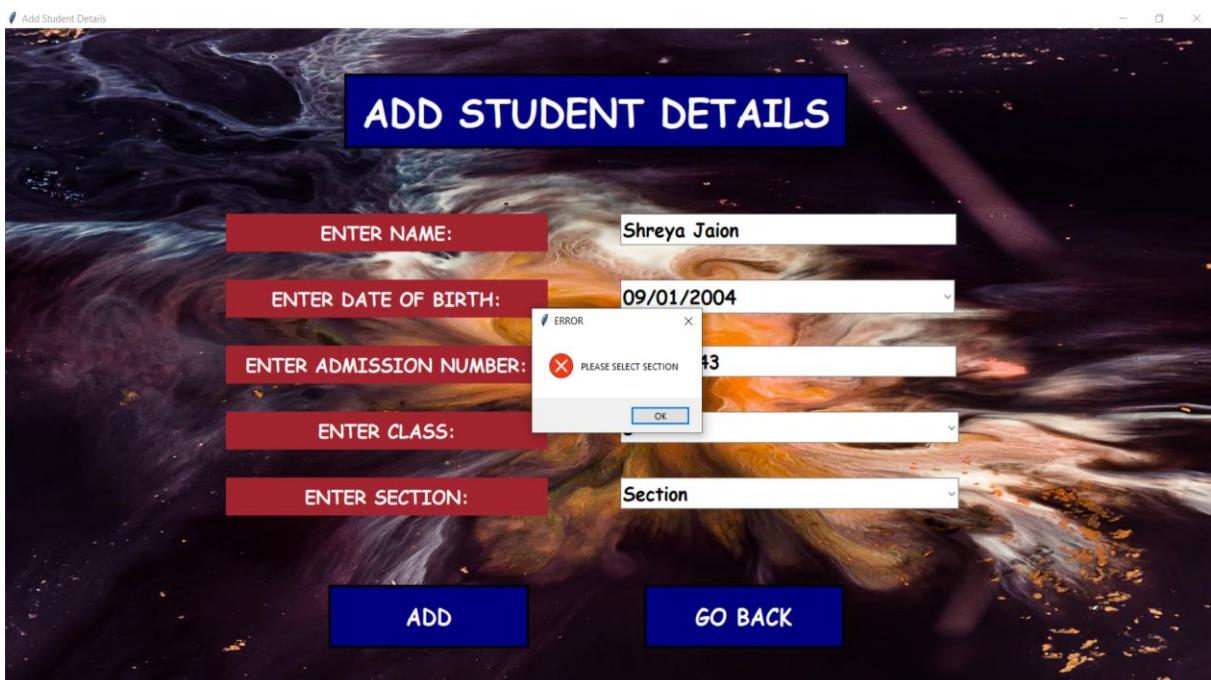
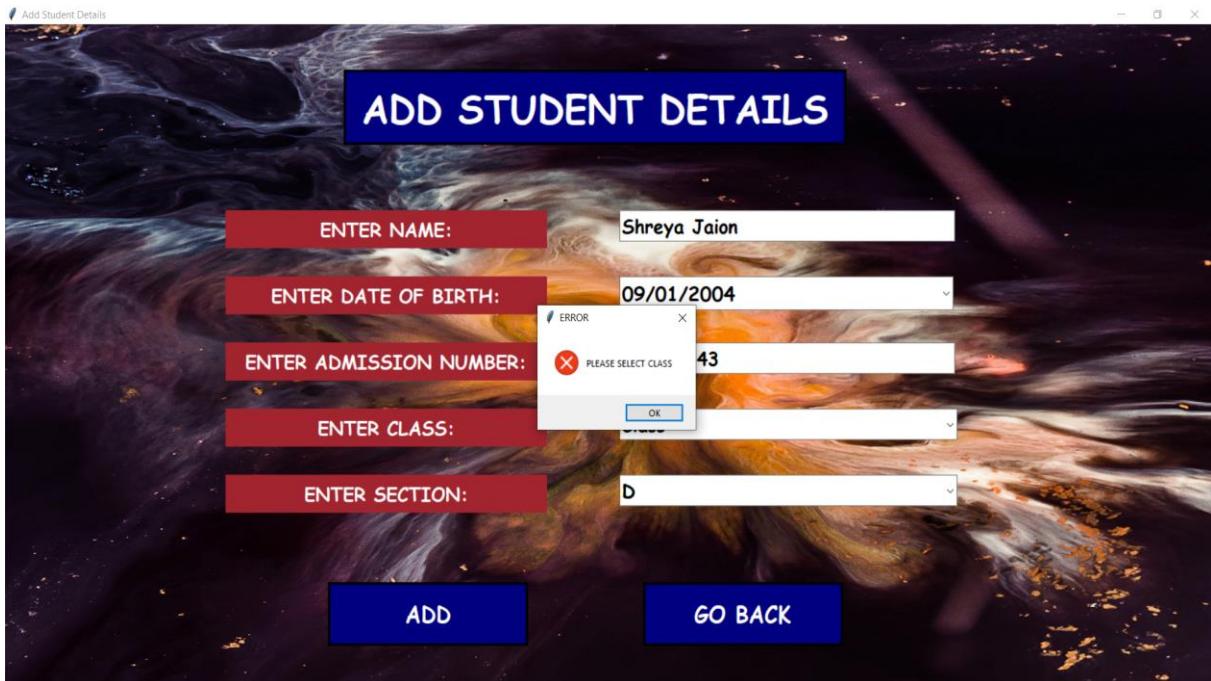
- In Add Student Details, if any box is left blank, it will show the following error:



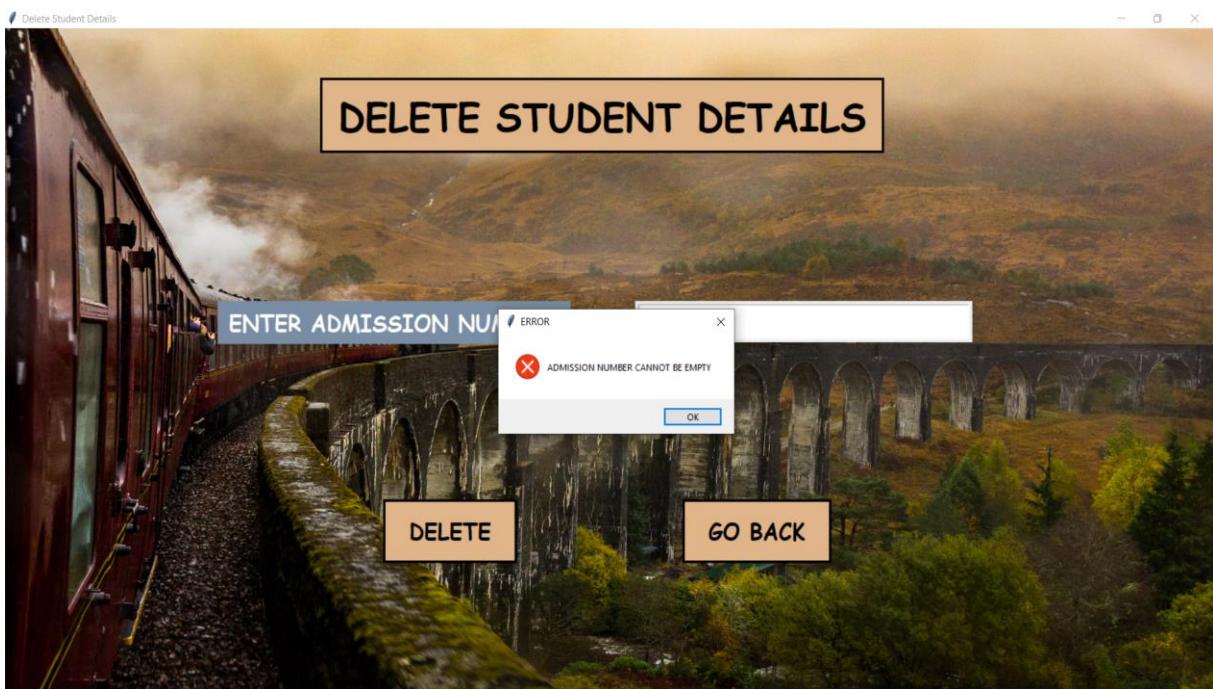
- In Add Student Details, if an already existing Admission Number is entered, it will show the following error:



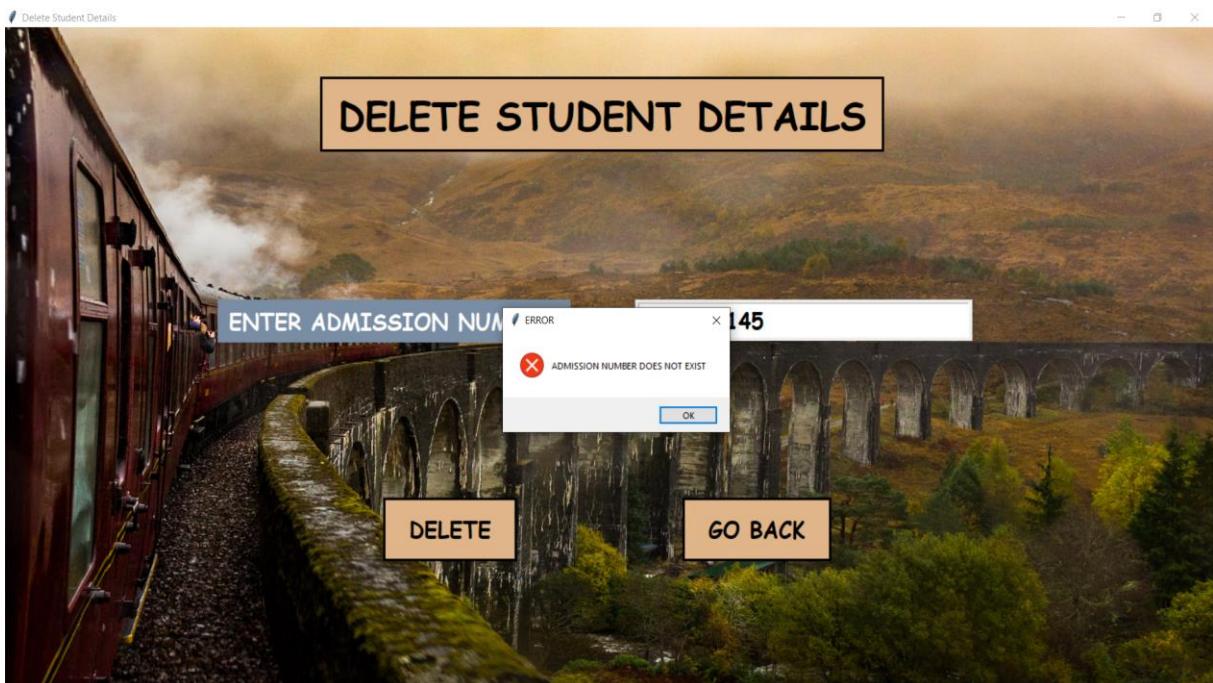
➤ In Add Student Details, if we enter the name and admission number but not the class or section, it shows the following error:



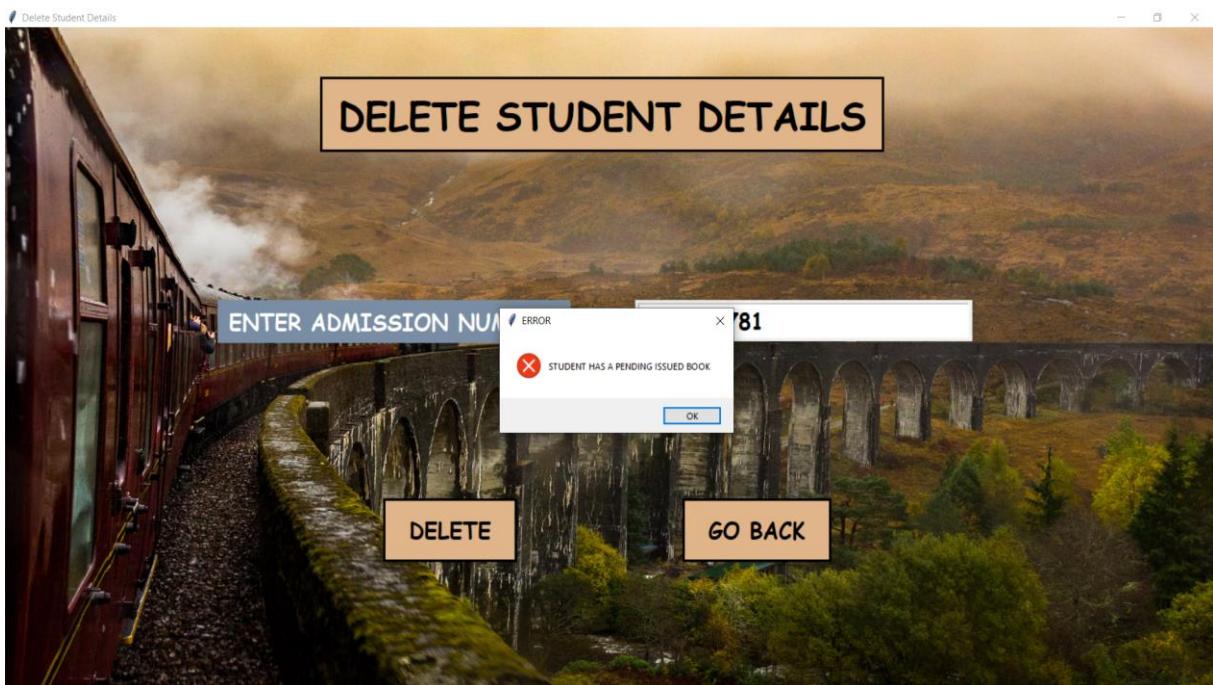
- In Delete Student Details, if box is left blank, it will show the following error:



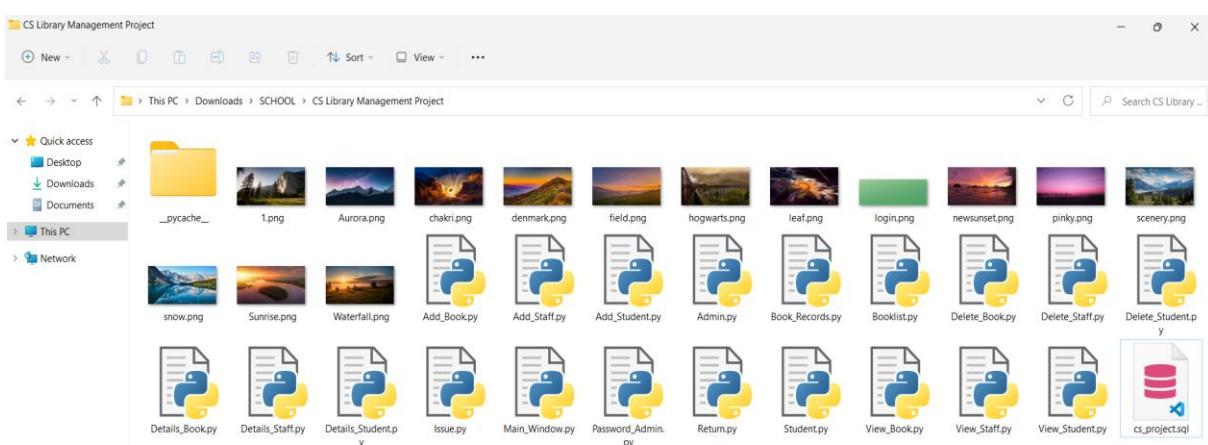
- In Delete Student Details, if an unknown admission number is entered, it will show the following error:



➤ In Delete Student Details, if the admin tries to delete a student that has currently issued a book and not returned it yet, it shows the following error:



All these files are stored in a single folder:



CONCLUSION

The entire project that has been displayed above, the thousands of lines of code written, and the tremendous amount of time and energy spent by the group on this project has successfully borne fruit and works quite well.

This project shows the working of SQL and python and provides an interactive application which allows people to issue and return books efficiently.

This project has enhanced our in-depth knowledge about the link between Python and MySQL. We have also learnt a lot about Tkinter and its usage in developing a program based on GUI (Graphical User Interface).

FURTHER IMPROVEMENTS

Currently, our Tkinter-based application works in multiple windows. In the future, we would like it to work in a single window with the help of a back button functionality.

Also, we would like to add a column to track the quantity of books for each book title listed in the database, to enable multiple issues.

BIBLIOGRAPHY

- <https://www.google.co.in>
- Computer Science with Python - Sumita Arora
- stackoverflow.com
- freeCodeCamp.org - YouTube Channel
- Unsplash.com