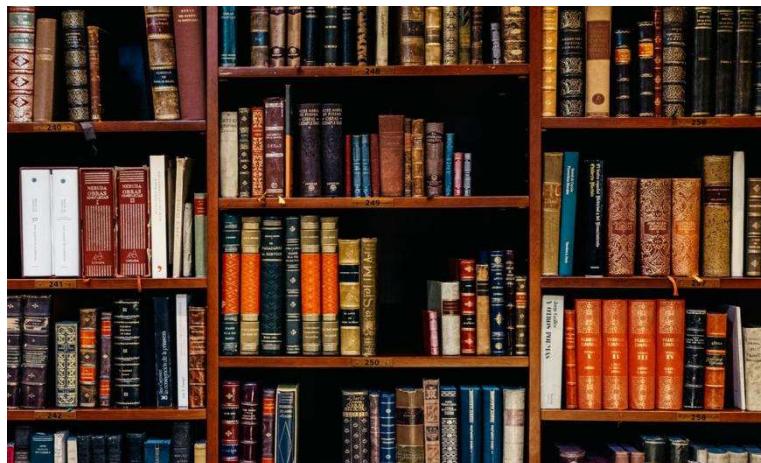


Project

# Computer Science



# Library Management System

By: Mukundh J

Class: XII C

Reg no:



***SSVM World School***  
*Singanallur, Coimbatore*  
***Department of Computer Science***

**Bonafide Certificate**

I hereby certify that this project entitled, **LIBRARY MANAGEMENT SYSTEM** is a bonafide work done by **MUKUNDH.J**

Register No: \_\_\_\_\_

In the department of Computer science of ***SSVM WORLD SCHOOL*** under my supervision during the academic year 2020-2021.

Date:\_\_\_\_\_

*Signature of the teacher-in-charge*

*Signature of the Principal*

*Submitted for practical examination held in the school centre on*

\_\_\_\_\_  
Signature of  
Internal Examiner

*School seal*

\_\_\_\_\_  
Signature of  
External Examiner

## **ACKNOWLEDGEMENT**

The successful completion of any task would be incomplete without crediting the names of those who helped to make it possible. I hereby take this opportunity to express my gratitude, gratefulness and respect in few words to all those who backed me in completion of this project. I express my special thanks to my Correspondent, **Mrs. Manimekalai Mohan**, my Principal **Mrs. Nimutha Pramod** and the teachers of the Computer science department, **Mr. Mujibullah** and **Mrs. Nirmala** who gave me ample help and support to pursue and complete a project on this topic.

Secondly I would like to thank CBSE for giving me such an opportunity without which I would not have been able to do such a project. I would also like to thank my parents and friends who helped me in my research and execution and it is with their supportiveness that I could complete this venture within the limited time frame.

My hearty thanks to the aforementioned.

<i>Sno.</i>	<i>Index</i>	<i>Page no.</i>
1.	<i>INTRODUCTION</i>	1
2.	<i>OBJECTIVE AND SCOPE OF THE PROJECT</i>	2
3.	<i>THEORETICAL BACKGROUND</i>	3
	<i>    3.1.WHAT IS A DATABASE</i>	
	<i>    3.2.WHAT IS Python IDLE</i>	4
	<i>    3.3.FEATURES OF Python</i>	5
	<i>    3.4 WHAT IS MYSQL</i>	8
	<i>    3.5THE MAIN FEATURES OF MYSQL</i>	
4.	<i>PROBLEM DEFINITIONS &amp;ANALYSIS</i>	10
5.	<i>SYSTEM IMPLEMENTATION</i>	11
	<i>    5.1.HARDWARE USED</i>	
	<i>    5.2.SOFTWARE USED</i>	
6.	<i>SYSTEM DESIGN &amp;DEVELOPMENT</i>	12
	<i>    6.1.DATABASE DESIGN</i>	
	<i>    6.2.TABLE DESIGN</i>	
7.	<i>SOURCE CODE</i>	13
8.	<i>USER MANUAL</i>	61
	<i>    8.1.HOW TO INSTALL SOFTWARE</i>	
	<i>    8.2.WORKING WITH SOFTWARE PROJECT</i>	
9.	<i>BIBLIOGRAPHY</i>	63

## **INTRODUCTION**

The purpose of this project, **Library Management System**, is to evolve a source code that provides a user friendly interface and focuses on developing a computerized system to maintain the daily work of a library that is developed to automate the functionalities necessary for a library management application. It mainly brings forth the usage of GUI programming in the daily use over the network. The program when made to work over the network can prove to be an ultimate way of interaction between the user and the software.

A MIS (Back-end) primarily consists of a computerized database, that stores a collection of inter-related tables for a specific purpose, capable of producing different reports that are particular to the user. An application program (Front-end) is tied with the database for easy access and as user interface to the database. Using a front-end, we can store, retrieve and manage all information in the most optimum and suitable manner.

This software, being simple in design and working, does not require much training to users, it can be used as the right tool for a library management system.

For the coding and design of the Project, Python IDLE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. My SQL, a powerful, open source RDBMS, is used as back-end as per requirement and suitability of the CBSE curriculum of Computer science course.

## **OBJECTIVE AND SCOPE OF THE PROJECT**

The objective of this Python project on Library Management System is to manage the details of booking and updating book, student, admin information.

1. The project is built at both administrative end and non-administrative end.
2. The purpose of the project is to build an application program to reduce the manual work for managing a library and display of other required details.
3. It stores all information about students, admin, fine, issued books, date of issue/submission and available quantity of a book.

### **The software system is expected to do the following functionality:**

1. To create a student or an admin account with appropriate details accordingly for the selected particular choice by the user
2. To renew password.
3. View the book status, book details
4. Update the database (Deletion can also be done if required)
5. Edit, issue, add books
6. View the books, its history and current status.
7. Search per requirement.

# THEORETICAL BACKGROUND

## WHAT IS DATABASE ?

Database is a systematic collection of data. It supports storage and manipulation of data, makes data management easy.

## What is RDBMS?

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

## What is a table?

The data in an RDBMS is stored in database objects which are called as **tables**. This table is basically a collection of related data entries and it consists of numerous columns and rows.

Example of a table:

All the information are stored in the form of tables.

Field	Type	Null	Key	Default	Extra
Id	int(4)	YES		NULL	
Name	varchar(30)	YES		NULL	
dateofjoin	datetime	YES		NULL	
Username	varchar(20)	YES		NULL	
Password	varchar(20)	YES		NULL	

## ABOUT PYTHON IDLE

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability.

The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has the programming easy and fun to do.

The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming.

It has a comprehensive and large standard library that has automatic memory management and dynamic features.

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default.

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

According to the included README, its main features are:

- Multi-window text editor with syntax highlighting, autocompletion, smart indent and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

Python has topped the charts in the recent years over other programming languages like C, C++ and Java and is widely used by the programmers. The language has

undergone a drastic change since its release 25 years ago as many add-on features are introduced. The Python 1.0 had the module system of Modula-3 and interacted with Amoeba Operating System with varied functioning tools. Python 2.0 introduced in the year 2000 had features of garbage collector and Unicode Support. Python 3.0 introduced in the year 2008 had a constructive design that avoids duplicate modules and constructs. With the added features, now the companies are using Python 3.5.

Some of the characteristics of Python that attract big companies to use this programming language are:

- **Interactive**
- **Interpreted**
- **Modular**
- **Dynamic**
- **Object-oriented**
- **Portable**
- **High level**
- **Extensible in C++ & C**

Some of the advantages of Python are:

- ◆ **Extensive Support Libraries**
- ◆ **Integration Feature**
- ◆ **Improved Programmer's Productivity**

## Limitations or Disadvantages of Python:

- **Difficulty in Using Other Languages**
- **Weak in Mobile Computing**
- **Gets Slow in Speed**
- **Run-time Errors**
- **Under developed Database Access Layers**

## ABOUT TKINTER IN PYTHON

The **Tkinter** module (“Tk interface”) is the standard Python interface to the Tk GUI toolkit from [Scriptics](#) (formerly developed by Sun Labs).

Both Tk and Tkinter are available on most Unix platforms, as well as on Windows and Macintosh systems. Starting with the 8.0 release, Tk offers native look and feel on all platforms.

Tkinter consists of a number of modules. The Tk interface is provided by a binary extension module named `_tkinter`. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

The public interface is provided through a number of Python modules. The most important interface module is the **Tkinter** module itself. To use Tkinter, all you need to do is to import the **Tkinter** module:

```
import Tkinter
```

Or, more often:

```
from Tkinter import *
```

The Tkinter module only exports widget classes and associated constants, so you can safely use the **from-in** form in most cases. If you prefer not to, but still want to save some typing, you can use **import-as**:

```
import Tkinter as Tk
```

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

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

## ABOUT MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- 1.Own specific environments MySQL is released under an open-source license. So you have nothing to pay to use it.
2. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySQL uses a standard form of the well-known SQL data language.
4. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
5. MySQL works very quickly and works well even with large data sets.
6. MySQL is very friendly to PHP, the most appreciated language for web development.
7. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
8. MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their.

In this project MySQL is used as an back end.

MySQL in Python can be used by:

1. Connecting to the database.
2. Creating an object for your database.
3. Executing the **SQL** query.
4. Fetching records from the result.
5. Informing the Database if you make any changes in the table

## **PROBLEM DEFINITION AND ANALYSIS**

In the scenario where data is maintained through registers, a computerized way to maintain and manage a library is much needed.

The reasons that contribute to its need:

- ❖ Less efficient man power.
- ❖ Time consumption.
- ❖ The large number of registers, requirement of storage spaces.
- ❖ Difficulty in accessing and retrieve written data.
- ❖ It is less economical.
- ❖ Loss of paper.
- ❖ Cheaper.

The pros of Library Management System :

- ❖ Records can be accessed/modified easily at one stroke.
- ❖ Data is retrieved easily.
- ❖ Data is more secure.
- ❖ Time consumption is a lot lesser.
- ❖ Paper is no more a necessity.

## **SYSTEM IMPLEMENTATION**

### **THE HARDWARE USED :**

- I5 6<sup>th</sup> generation
- 8GB RAM
- 1 TB HDD

### **THE SOFTWARE USED :**

- Microsoft windows 10 as operating system
- Python idle 3.7.5 as Front end
- MySQL as Back end server with database for testing
- WPS office for documentation

# SYSTEM DESIGN & DEVELOPMENT

## *Table Design:*

The database named LibraryManagement, contains 4 tables. The tables are to minimize the redundancies of data. The tables and their structure are given below:

Tables_in_librarymanagement
adetail
bookdetail
issuedbooks
studetail

TABLE NAME: adetail

Field	Type	Null	Key	Default	Extra
Id	int(4)	YES		NULL	
Name	varchar(30)	YES		NULL	
dateofjoin	datetime	YES		NULL	
Username	varchar(20)	YES		NULL	
Password	varchar(20)	YES		NULL	

TABLE NAME: bookdetail

Field	Type	Null	Key	Default	Extra
bookid	varchar(6)	YES		NULL	
Name	varchar(30)	YES		NULL	
quantity	int(11)	YES		NULL	
availableqty	int(11)	YES		NULL	
Author	varchar(25)	YES		NULL	
doa	varchar(50)	YES		NULL	

### TABLE NAME: studetail

Field	Type	Null	Key	Default	Extra
Adno	int(6)	YES		NULL	
Name	varchar(30)	YES		NULL	
dob	date	YES		NULL	
Class	varchar(3)	YES		NULL	
Username	varchar(20)	YES		NULL	
Password	varchar(20)	YES		NULL	
fine	int(11)	YES		NULL	

### TABLE NAME: issuedbooks

Field	Type	Null	Key	Default	Extra
adno	int(11)	YES		NULL	
Name	varchar(50)	YES		NULL	
Class	varchar(50)	YES		NULL	
bookid	varchar(10)	YES		NULL	
bookname	varchar(50)	YES		NULL	
D_O_I	varchar(30)	YES		NULL	
Due_Date	varchar(50)	YES		NULL	
D_O_S	varchar(50)	YES		NULL	

# Source Code

## Library.py

```
from tkinter import *
from tkinter import messagebox as mbox
from tkinter import simpledialog,ttk
import mysql.connector as sq
import datetime as datym
import os
import random
import pyttsx3
global con,cur

schoolcode='_123THeZxen2o2*L!bR4rY.-!'
con=sq.connect(host='localhost',user='root',passwd='123')
cur=con.cursor()
cur.execute('create database if not exists LibraryManagement')
cur.execute('use LibraryManagement')
cur.execute('create table if not exists bookdetail(bookid varchar(6) ,Name varchar(30),quantity int,availableqtyint,Author varchar(25),doa varchar(50))'#set the primary keys
cur.execute('create table if not exists issuedbooks(adno int, Name varchar(50),Class varchar(50),bookid varchar(10),bookname varchar(50), D_O_I varchar(30),Due_Date varchar(50),D_O_S varchar(50))')
cur.execute('create table if not exists studetail(Adno int(6) ,Name varchar(30),dob date,Class varchar(3),Username varchar(20),Password varchar(20),fine int)' ) #set the primary keys
cur.execute('create table if not exists adetail(Id int(4) ,Name varchar(30),datymofjoin varchar(30),Username varchar(20),Password varchar(20))')
con.commit()
#-----speak
function-----
engine=pyttsx3.init('sapi5')
voices=engine.getProperty('voices')
engine.setProperty('voice',voices[0].id)
def speak(text):
```

```

engine.say(text)
engine.runAndWait()
#-----VIEW HISTORY
-----
def searchh():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    p=s1.get()
    if p!="":
        try:
            search2.delete(0,END)
            sql1=f'select * from issuedbooks where adno ="{p}"'
        cur.execute(sql1)
        r=cur.fetchall()
        total=len(r)
        labl.configure(text=total)
        for i in tree.get_children():
            tree.delete(i)
            for i in r:
                tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
        con.close()
    except:
        mbox.showinfo('ALERT!','Unable to search book.')

```

```

def delete_infoh(*event):

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    if tree.focus():
        c=tree.focus()
        f=tree.item(c)
        f=f['values']
        p1=str(f[0])
        p2 = str(f[3])

```

```

s = f'delete from issued where adno = "{p1}" and bookid ="{p2}"'
cur.execute(s)
con.commit()
    for i in tree.get_children():
tree.delete(i)
cur.execute('select * from issuedbooks')
    r=cur.fetchall()
    total=len(r)
labl.configure(text=total)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))

def reh():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
cur.execute('select * from issuedbooks')
    r=cur.fetchall()
    total=len(r)
labl.configure(text=total)
    for i in tree.get_children():
tree.delete(i)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
def viewbook_backh():
viewform.destroy()
admin_page()

def history():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    global labl,s1,search2,tree,viewform
try:
    root6.destroy()
except:
    pass

```

```

viewform=Tk()
#-----treeview styling-----
style = ttk.Style(viewform)

style.theme_use("clam")
style.configure("Treeview", background="black", fieldbackground="#D1A684",
foreground="black")
#-----


viewform.geometry('1200x650')
viewform.title('BOOK HISTORY')
s1=StringVar()

TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)
TopViewForm.pack(side=TOP, fill=X)
LeftViewForm = Frame(viewform, width=300,bg='#FE858D')
LeftViewForm.pack(side=LEFT, fill=Y)
MidViewForm = Frame(viewform, width=600)
MidViewForm.pack(side=RIGHT)

text = Label(TopViewForm, text="History", bg='#37CBBE',fg='white',font=('times',
18,'italic'), width=600)
text.pack(fill=X)

txtsearch = Label(LeftViewForm, text="BOOK ID",bg='#FE858D',fg='white',font=('times',
20,'bold','italic')).place(x=70,y=50)
search2 = Entry(LeftViewForm, font=(15),bd=3,textvariable=s1 ,width=20)
search2.place(x=50,y=100)
photo = PhotoImage(file = "search.png")
search =
Button(LeftViewForm,image=photo,bd=0,bg='#FE858D',height=40,width=100,command=search_
chh).place(x=70,y=150)

photo3 = PhotoImage(file = "delete.png")
delete =
Button(LeftViewForm,image=photo3,bd=0,bg='#FE858D',height=40,width=100,command=del_
ete_infoh).place(x=70,y=310)

photo4 = PhotoImage(file = "refresh.png")
delete =
Button(LeftViewForm,image=photo4,bd=0,bg='#FE858D',height=40,width=100,command=re_
h).place(x=60,y=390)

```

```

scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)
scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)
    tree = ttk.Treeview(MidViewForm, columns=('1','2','3','4','5','6','7','8'), height=100,
yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
    tree['columns']=('1','2','3','4','5','6','7','8')
    tree['show']='headings'
tree.heading('1',text='Admission Number')
tree.heading('2',text='Name')
tree.heading('3',text='Class')
tree.heading('4',text='BookId')
tree.heading('5',text='Book Name')
tree.heading('6',text='D.O.I')
tree.heading('7',text='Due Date')
tree.heading('8',text='D.O.S')
tree.column('8',width=280,anchor='center')
tree.column('7',width=280,anchor='center')
tree.column('6',width=280,anchor='center')
tree.column('5',width=280,anchor='center')
tree.column('4',width=280,anchor='center')
tree.column('3',width=240,anchor='center')
tree.column('2',width=240,anchor='center')
tree.column('1',width=170,anchor='center')
tree.pack()
cur.execute('select * from issuedbooks')
r=cur.fetchall()
for i in r:
    tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
total=len(r)
photo44 = PhotoImage(file = "back1.png")
search=
Button(LeftViewForm,image=photo44,height=40,width=70,bg='#FE858D',bd=0,command=vie
wbook_backh).place(x=80,y=460)

```

```

Label(viewform,text='TOTAL RESULTS FOUND :
',bg='#FE858D',fg='white',font=('times',14,'italic')).place(x=40,y=575)
labl=Label(viewform,text=total,bg='#FE858D',fg='white',font=('times',20,'bold'))
labl.place(x=140,y=600)
viewform.mainloop()

#----- VIEW ISSUED
BOOKS-----
def searchts():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    p=s1.get()
    if p!="":
        try:
            search2.delete(0,END)
            sql1=f'select * from issuedbooks where adno ="{p}" and D_O_S = "'
    cur.execute(sql1)
    r=cur.fetchall()
    total=len(r)
    labl.configure(text=total)
        for i in tree.get_children():
            tree.delete(i)
            for i in r:
                tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
    con.close()
    except:
mbox.showinfo('ALERT!','Unable to search book.')

def submit_infots(*event):

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    if tree.focus():
        c=tree.focus()
        f=tree.item(c)

```

```

f=f['values']
p1=str(f[0])
p2 = str(f[3])
today = datym.date.today()
today_str = today.strftime("%Y-%m-%d")
s=[f'updateissuedbooks set D_O_S = "{today_str}" where adno="{p1}" and
bookid="{p2}"',f"updatebookdetail set availableqty=availableqty+1 where bookid='{p2}'"]
for j in s:
    cur.execute(j)
con.commit()

s = f'selectDue_date from issuedbooks where adno="{p1}" and bookid="{p2}"'
cur.execute(s)
duedate = cur.fetchone()[0]
fdueDate = datym.date.fromisoformat(duedate)
day = (today - fdueDate).days
if day > 0:
    mbox.showinfo('FINE ',f"Total Fine amount to be paid Rs.{day*10}")
    s = f'updatestudetail set fine = fine + {day * 10} where Adno="{p1}"'
    cur.execute(a)
    con.commit()

try:
    for i in tree.get_children():
tree.delete(i)
cur.execute('select * from issuedbooks where D_O_S = "")')
r=cur.fetchall()
total=len(r)
labl.configure(text=total)
for i in r:
    tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))

except:
    pass

def rets():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
cur.execute('select * from issuedbooks where D_O_S = "")')

```

```

r=cur.fetchall()
total=len(r)
labl.configure(text=total)
for i in tree.get_children():
tree.delete(i)
for i in r:
    tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
def viewbook_backts():
viewform.destroy()
admin_page()

def to_submit():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
global labl,s1,search2,tree,viewform
try:
    root6.destroy()
except:
    pass
viewform=Tk()
#-----
style = ttk.Style(viewform)

style.theme_use("clam")
style.configure("Treeview", background="black", fieldbackground="#D1A684",
foreground="black")
#-----

viewform.geometry('1200x650')
viewform.title('TO SUBMIT DETAILS')
s1=StringVar()
TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)
TopViewForm.pack(side=TOP, fill=X)
LeftViewForm = Frame(viewform, width=300,bg='#FE858D')
LeftViewForm.pack(side=LEFT, fill=Y)
MidViewForm = Frame(viewform, width=600)

```

```

MidViewForm.pack(side=RIGHT)

    text = Label(TopViewForm, text="Submission Details",
bg ='#37CBBE',fg='white',font=('times', 18,'italic'), width=600)
text.pack(fill=X)

txtsearch = Label(LeftViewForm, text="BOOK ID",bg ='#FE858D',fg='white',font=('times',
20,'bold','italic')).place(x=70,y=50)

    search2 = Entry(LeftViewForm, font=(15),bd=3,textvariable=s1 ,width=20)
    search2.place(x=50,y=100)
    photo = PhotoImage(file = "search.png")
    search =
Button(LeftViewForm,image=photo,bd=0,bg ='#FE858D',height=40,width=100,command=sear
chts).place(x=70,y=150)

    photo3 = PhotoImage(file = "sub2.png")
    delete =
Button(LeftViewForm,image=photo3,bd=0,bg ='#FE858D',height=40,width=100,command=su
bmit_infots).place(x=70,y=310)

    photo4 = PhotoImage(file = "refresh.png")
    delete =
Button(LeftViewForm,image=photo4,bd=0,bg ='#FE858D',height=40,width=100,command=ret
s).place(x=60,y=390)

scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)
scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)

    tree = ttk.Treeview(MidViewForm, columns=('1','2','3','4','5','6','7','8'), height=100,
yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)

    tree['columns']=('1','2','3','4','5','6','7','8')
    tree['show']='headings'
tree.heading('1',text='Admission Number')
tree.heading('2',text='Name')
tree.heading('3',text='Class')
tree.heading('4',text='BookId')
tree.heading('5',text='Book Name')
tree.heading('6',text='D.O.I')
tree.heading('7',text='Due Date')

```

```

tree.heading('8',text='D.O.S')
tree.column('8',width=280,anchor='center')
tree.column('7',width=280,anchor='center')
tree.column('6',width=280,anchor='center')
tree.column('5',width=280,anchor='center')
tree.column('4',width=280,anchor='center')
tree.column('3',width=240,anchor='center')
tree.column('2',width=240,anchor='center')
tree.column('1',width=170,anchor='center')
tree.pack()
cur.execute('select * from issuedbooks where D_O_S = "")')
r=cur.fetchall()
for i in r:
    tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
total=len(r)
photo44 = PhotoImage(file = "back1.png")
search =
Button(LeftViewForm,image=photo44,height=40,width=70,bg='#FE858D',bd=0,command=vie
wbook_backts).place(x=80,y=460)
Label(viewform,text='TOTAL RESULTS FOUND :
',bg='#FE858D',fg='white',font=('times',14,'italic')).place(x=40,y=575)
labl=Label(viewform,text=total,bg='#FE858D',fg='white',font=('times',20,'bold'))
labl.place(x=140,y=600)
viewform.mainloop()

#-----
def search111():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
p=s1.get()
if p!="":
    try:
        search2.delete(0,END)
        sql1=f'select * from studetail where Adno ="{p}"'
        cur.execute(sql1)
        r=cur.fetchall()

```

```

total=len(r)
lbl.configure(text=total)
    for i in tree.get_children():
tree.delete(i)
    for i in r:
tree.insert("",'end',values=(i[0],i[1],i[2],i[3]))
con.close()
except:
mbox.showinfo('ALERT!','Unable to search book.')

def delete_info11(*event):
con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
speak('Are you sure?')
if mbox.askquestion("Confirm Resubmission", "Are you sure?")=='yes':
    if tree.focus():
        c=tree.focus()
        f=tree.item(c)
        f=f['values']
        p1=str(f[0])
        s=f'delete from studetail where Adno ={p1}"'
cur.execute(s)
con.commit()
    for i in tree.get_children():
tree.delete(i)
cur.execute('select * from studetail')
r=cur.fetchall()
total=len(r)
lbl.configure(text=total)
    for i in r:
tree.insert("",'end',values=(i[0],i[1],i[2],i[3]))
else:
    pass
def re11():

```

```

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
cur.execute('select * from studetail')
r=cur.fetchall()
total=len(r)
labl.configure(text=total)
for i in tree.get_children():
    tree.delete(i)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3]))
def viewbook_back1():
    viewform.destroy()
    admin_page()

def pay_fine():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
if tree.focus():
    c=tree.focus()
    f=tree.item(c)
    f=f['values']
    p1=str(f[0])
    s=f'updatestudetail set fine = 0 where adno = "{p1}"'
    cur.execute(s)
    con.commit()
    for i in tree.get_children():
        tree.delete(i)
    cur.execute('select * from studetail')
    r=cur.fetchall()
    total=len(r)
    labl.configure(text=total)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3]))

def student_details():

```

```

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
global labl,s1,search2,tree,viewform
try:
    root6.destroy()
except:
    pass
viewform=Tk()
#-----
style = ttk.Style(viewform)

style.theme_use("clam")
style.configure("Treeview", background="black", fieldbackground="#D1A684",
foreground="black")
#-----

viewform.geometry('1200x650')
viewform.title('STUDENT DETAILS')
s1=StringVar()

TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)
TopViewForm.pack(side=TOP, fill=X)
LeftViewForm = Frame(viewform, width=300,bg='#FE858D')
LeftViewForm.pack(side=LEFT, fill=Y)
MidViewForm = Frame(viewform, width=600)
MidViewForm.pack(side=RIGHT)

text = Label(TopViewForm, text="Student Details", bg="#37CBBE",fg='white',font=('times',
18,'italic'), width=600)
text.pack(fill=X)

txtsearch = Label(LeftViewForm, text="BOOK ID",bg="#FE858D",fg='white',font=('times',
20,'bold','italic')).place(x=70,y=50)

search2 = Entry(LeftViewForm, font=(15),bd=3,textvariable=s1 ,width=20)
search2.place(x=50,y=100)

photo = PhotoImage(file = "search.png")

search =
Button(LeftViewForm,image=photo,bd=0,bg="#FE858D",height=40,width=100,command=search111).place(x=70,y=150)

```

```

photo3 = PhotoImage(file = "delete.png")
delete =
Button(LeftViewForm,image=photo3,bd=0,bg='#FE858D',height=40,width=100,command=del
ete_info11).place(x=70,y=290)
delpic=PhotoImage(file = "payfyn.png")
delete =
Button(LeftViewForm,image=delpic,bd=0,bg='#FE858D',height=40,width=100,command=pay
_fine).place(x=70,y=350)
photo4 = PhotoImage(file = "refresh.png")
delete =
Button(LeftViewForm,image=photo4,bd=0,bg='#FE858D',height=40,width=100,command=re1
1).place(x=70,y=410)
scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)
scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)
tree = ttk.Treeview(MidViewForm, columns=('1','2','3','4','5'), height=100,
yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
tree['columns']=('1','2','3','4','5')
tree['show']='headings'
tree.heading('1',text='Admission Number')
tree.heading('2',text='Name')
tree.heading('3',text='D.O.B')
tree.heading('4',text='Class')
tree.heading('5',text='Fine')
tree.column('4',width=280,anchor='center')
tree.column('3',width=240,anchor='center')
tree.column('2',width=240,anchor='center')
tree.column('1',width=170,anchor='center')
tree.column('5',width=170,anchor='center')
tree.pack()
cur.execute('select * from studetail')
r=cur.fetchall()
for i in r:
tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[6]))

```

```

total=len(r)
photo44 = PhotoImage(file = "back1.png")
search =
Button(LeftViewForm,image=photo44,height=40,width=70,bg='#FE858D',bd=0,command=vie
wbook_back1).place(x=80,y=470)
Label(viewform,text='TOTAL RESULTS FOUND :
',bg='#FE858D',fg='white',font=('times',14,'italic')).place(x=40,y=575)
labl=Label(viewform,text=total,bg='#FE858D',fg='white',font=('times',20,'bold'))
labl.place(x=140,y=600)
viewform.mainloop()
##-----#
#-----#admin login -----#
def search11():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
p=s1.get()
if p!="":
try:
    search2.delete(0,END)
    sql1=f'select * from bookdetail where bookid ="{p}"'
cur.execute(sql1)
r=cur.fetchall()
total=len(r)
labl.configure(text=total)
for i in tree.get_children():
tree.delete(i)
for i in r:
tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5]))
con.close()
except:
mbox.showinfo('ALERT!','Unable to search book.')

def delete_info1(*event):

```

```

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
if tree.focus():
    c=tree.focus()
    f=tree.item(c)
    f=f['values']
    p1=str(f[0])
    s=f'delete from bookdetail where bookid ={p1}"'
    print(s)
    cur.execute(s)
    con.commit()
    for i in tree.get_children():
        tree.delete(i)
    cur.execute('select * from bookdetail')
    r=cur.fetchall()
    total=len(r)
    labl.configure(text=total)
    for i in r:
        tree.insert("","",end,values=(i[0],i[1],i[2],i[3],i[4],i[5]))
def re1():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
cur.execute('select * from bookdetail')
r=cur.fetchall()
total=len(r)
labl.configure(text=total)
for i in tree.get_children():
    tree.delete(i)
    for i in r:
        tree.insert("","",end,values=(i[0],i[1],i[2],i[3],i[4],i[5]))
def viewbook_back():
    viewform1.destroy()
admin_page()
def view_book():

```

```

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
global labl,s1,search2,tree,viewform1
try:
    root6.destroy()
except:
    pass
viewform1=Tk()
#-----
style = ttk.Style(viewform1)

style.theme_use("clam")
style.configure("Treeview", background="black", fieldbackground="#D1A684",
foreground="black")
#-----

viewform1.geometry('1200x650')
viewform1.title('VIEW BOOK')
s1=StringVar()
Topviewform1 = Frame(viewform1, width=600, bd=1, relief=SOLID)
Topviewform1.pack(side=TOP, fill=X)
Leftviewform1 = Frame(viewform1, width=300,bg='#FE858D')
Leftviewform1.pack(side=LEFT, fill=Y)
Midviewform1 = Frame(viewform1, width=600)
Midviewform1.pack(side=RIGHT)
text = Label(Topviewform1, text="Book Details", bg='#37CBBE',fg='white',font=('times',
18,'italic'), width=600)
text.pack(fill=X)
txtsearch = Label(Leftviewform1, text="BOOK ID",bg='#FE858D',fg='white',font=('times',
20,'bold','italic')).place(x=70,y=50)
search2 = Entry(Leftviewform1, font=(15),bd=3,textvariable=s1 ,width=20)
search2.place(x=50,y=100)
photo = PhotoImage(file = "search.png")
search =
Button(Leftviewform1,image=photo,bd=0,bg='#FE858D',height=40,width=100,command=sea
rch11).place(x=70,y=150)

```

```

photo3 = PhotoImage(file = "delete.png")
delete =
Button(Leftviewform1,image=photo3,bd=0,bg ='#FE858D',height=40,width=100,command=de
lete_info1).place(x=70,y=310)
photo4 = PhotoImage(file = "refresh.png")
delete =
Button(Leftviewform1,image=photo4,bd=0,bg ='#FE858D',height=40,width=100,command=re
1).place(x=60,y=390)
scrollbarx = Scrollbar(Midviewform1, orient=HORIZONTAL)
scrollbary = Scrollbar(Midviewform1, orient=VERTICAL)
tree = ttk.Treeview(Midviewform1, columns=('1','2','3','4','5','6'), height=100,
yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
tree['columns']=('1','2','3','4','5','6')
tree['show']='headings'
tree.heading('1',text='Book Id')
tree.heading('2',text='Book Name')
tree.heading('3',text='Supplied Quantity')
tree.heading('4',text='Avilable Quantity')
tree.heading('5',text='Author')
tree.heading('6',text='D.O.A')
tree.column('6',width=280,anchor='center')
tree.column('5',width=280,anchor='center')
tree.column('4',width=280,anchor='center')
tree.column('3',width=240,anchor='center')
tree.column('2',width=240,anchor='center')
tree.column('1',width=170,anchor='center')
tree.pack()
cur.execute('select * from bookdetail')
r=cur.fetchall()
for i in r:
tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5]))
total=len(r)
photo44 = PhotoImage(file = "back1.png")

```

```

search =
Button(Leftviewform1,image=photo44,height=40,width=70,bg="#FE858D",bd=0,command=vie
wbook_back).place(x=80,y=460)
    Label(viewform1,text='TOTAL RESULTS FOUND :
',bg="#FE858D",fg='white',font=('times',14,'italic')).place(x=40,y=575)
    labl=Label(viewform1,text=total,bg="#FE858D",fg='white',font=('times',20,'bold'))
labl.place(x=140,y=600)
    viewform1.mainloop()

def submitbook():
    try:
st = f"insert into bookdetail
values('{bookid.get()}','{bookname.get()}',{bookqty.get()},{bookqty.get()},{bookdealer.get()},{datym.date.today()})"
cur.execute(st)
con.commit()
mbox.showinfo("SUCCESS","Book Registered")
    except:
mbox.showinfo("ERROR","Failed to register book. Try again")
def add_bookback():
    try :
        root7.destroy()
admin_page()
except:pass
def add_book():
    global root7,bookid,bookname,bookqty,bookdealer
    try:
        root6.destroy()
    except:
        pass
    root7 = Tk()
bookid = StringVar()
bookname = StringVar()
bookqty = StringVar()
    bookdealer = StringVar()
    root7.geometry("1000x700")
    pic=PhotoImage(file='zencred2.png')

```

```

Label(root7,image=pic).place(relwidth=1,relheight=1)
Label(root7,text = "Book id",bg='black',fg='white').place(x=200,y=200)

b1 = Entry(root7,textvariable = bookid)
b1.place(x=200,y=220)

Label(root7,text = "Book Name",bg='black',fg='white').place(x=200,y=240)

b2 = Entry(root7,textvariable = bookname)
b2.place(x=200,y=260)

Label(root7,text = "Book Quantity",bg='black',fg='white').place(x=200,y=280)

b3 = Entry(root7,textvariable = bookqty)
b3.place(x=200,y=300)

Label(root7,text = "Book Author",bg='black',fg='white').place(x=200,y=320)

b4 = Entry(root7,textvariable = bookdealer)
b4.place(x=200,y=340)
pic2=PhotoImage(file='subbutton.png')
button1 = Button(root7,image=pic2,width=120,height=25,bg='black',command =
submitbook)
button1.place(x=200,y=380)
pic3=PhotoImage(file='back.png')
button2 =
Button(root7,image=pic3,width=120,height=25,bg='black',command=add_bookback)
#,command
button2.place(x=200,y=420)
root7.mainloop()

def addm_book():
admno = addno.get()
bookid = bid.get()
if admno and bookid:
st = f'selectName,Class from studetail where Adno = {admno}'
cur.execute(st)

```

```

stdname = cur.fetchone()
st = f"selectName,availableqty from bookdetail where bookid = '{bookid}'"
cur.execute(st)
    values = cur.fetchone()
    if values and stdname:
        if values[-1] >0:
            today = datym.date.today()
            DD = datym.timedelta(days=10)
            earlier = today + DD
earlier_str = earlier.strftime("%Y-%m-%d")
today_str = today.strftime("%Y-%m-%d")
st = f"insert into
issuedbooksvalues({admno},{stdname[0]},{stdname[1]},{bookid},{values[0]},{today_str},{{
earlier_str}},)"
cur.execute(st)
con.commit()
st = f"updatebookdetail set availableqty=availableqty-1 where bookid='{bookid}'"
cur.execute(st)
con.commit()
mbox.showinfo("SUCCESS",f"Booknammed {values[0]} issued to {stdname[0]}¥n Due date for
submission {earlier_str}")

else:
mbox.showinfo("ERROR","Book Currently Not Available")
else:
mbox.showinfo("ERROR","Enter Correct Admission number or Book Id")
else:
mbox.showinfo("Enter","Fill All Parameters.")
def issue_booksback():
try :
    root8.destroy()
admin_page()
except:pass
def issue_books():
try:
    root6.destroy()
except:

```

```

    pass
global root8,addno,bid
root8 = Tk()
root8.geometry('1000x700')
addno = StringVar()
bid = StringVar()
pic=PhotoImage(file='zenew.png')
Label(root8,image=pic).place(relwidth=1,relheight=1)
Label(root8,text="Admission
Number:",bg='black',fg='white',font=('times',18,'italic')).place(x=410,y=300)
b1 = Entry(root8,textvariable = addno)
b1.place(x=450,y=330)

Label(root8,text="Book
Id:",bg='black',fg='white',font=('times',18,'italic')).place(x=450,y=380)
b2 = Entry(root8,textvariable = bid)
b2.place(x=450,y=410)
pic2=PhotoImage(file='subutton.png')
b3 = Button(root8,image=pic2,command= addm_book,width=120,height=25,bg='black')
pic3=PhotoImage(file='back.png')
button2 =
Button(root8,image=pic3,width=120,height=25,bg='black',command=issue_booksback)
#,command
button2.place(x=450,y=500)
b3.place(x=450,y=450)
root8.mainloop()
#-----#-----#-----#-----#-----#-----#-----#-----##

def adminlogout():
try:
    root6.destroy()
    admin()
except:
    pass
def admin_page():
global root6
try:

```

```

root1.destroy()
except:
    pass
root6 = Tk()
root6.geometry("1000x700")
pic=PhotoImage(file='zenauth.png')
Label(root6,image=pic).place(relwidth=1,relheight=1)
b1=PhotoImage(file='adbook.png')
bb1=Button(root6,image=b1,height=25,width=150,command = add_book)
bb1.place(x=460,y=250)
b2=PhotoImage(file='bkdet.png')
bb2=Button(root6,image=b2,height=25,width=150, command = view_book)
bb2.place(x=460,y=290)
b3=PhotoImage(file='issue.png')
bb3=Button(root6,image=b3,height=25,width=150, command = issue_books)
bb3.place(x=460,y=330)

ty=PhotoImage(file='submibook.png')
bb4=Button(root6,image=ty, height=25,width=150,command= to_submit)
bb4.place(x=460,y=370)

b5=PhotoImage(file='histry.png')
bb5=Button(root6,image=b5,height=25,width=150,bg='white',command = history)
bb5.place(x=460,y=410)

b4=PhotoImage(file='studet.png')
bb6=Button(root6,image=b4,height=25,width=150,command = student_details)
bb6.place(x=460,y=450)

pic3=PhotoImage(file='logout.png')
button2 =
Button(root6,image=pic3,width=160,height=25,bg='black',command=adminlogout)
button2.place(x=460,y=490)
root6.mainloop()

```

```

def admin_login():
adnusn = adnvalue.get()
adnpwd = adnvalue2.get()
cur.execute("select * from adetail")
adndetails = cur.fetchall()
for i in adndetails:
    if adnusn == i[-2] and adnpwd == i[-1]:
admin_page()
else:
mbox.showinfo("ERROR!","Incorrect Password or Username.")

def admin():
    global root1,adnvalue,adnvalue2
    try:
root.destroy()
except:
    pass
root1= Tk()
root1.geometry("1000x700")
adnvalue=StringVar()
adnvalue2=StringVar()
bg=PhotoImage(file='wlecome.png')
Label(root1,image=bg).place(relwidth=1,relheight=1)
bck=PhotoImage(file='backbtn.png')

Backbtn=Button(root1,bg='#2A382B',image=bck,height=30,width=80,command=admin_back)
Backbtn.grid(row=1,column=1)

username=Entry(root1,width=30,textvariable=adnvalue).place(x=450,y=300)
password=Entry(root1,width=30,textvariable=adnvalue2,show='•').place(x=450,y=325)
login=PhotoImage(file='login.png')
loginbtn=Button(root1,image=login,bg='black',height=25,width=175,command =
admin_login)
loginbtn.place(x=450,y=360)

```

```

signin=PhotoImage(file='signin.png')

signinbtn=Button(root1,image=signin,bg='black',height=25,width=175,command=enter_code)
signinbtn.place(x=450,y=450)

signup=PhotoImage(file='signin.png')
signupbtn=Button(root1,image=signup,bg='black',height=25,width=175, command =
studregister)
signupbtn.place(x=450,y=525)

Label(root1,text='New admin?',fg='white',bg='#37322F').place(x=450,y=425)
Label(root1,text='New student?',fg='white',bg='#37322F').place(x=450,y=500)
root1.mainloop()

def admin_valid():
    code = passvalue.get()
    if code == schoolcode:
        admin_registration()
    else:
        mbox.showinfo("ERROR!","Incorrect school code")

def enter_codeback():
    root3.destroy()
    admin()

def enter_code():
    global root3,passvalue
    try:
        root1.destroy()
    except:
        pass

    root3 =Tk()
    root3.geometry("1000x700")
    passvalue = StringVar()
    bg=PhotoImage(file='zenauth.png')
    Label(root3,image=bg).place(relwidth=1,relheight=1)
    l1=Label(root3,text="Enter school code",bg='#1C1C1A',fg='white')
    l1.place(x=650,y= 250)

```

```

e1 = Entry(root3,textvariable = passvalue)
e1.place(x=650,y=300)
b1 = Button(root3,text="OK",bg='#252525',fg='white', command= admin_valid)
b1.place(x=650,y= 350)
b2 = Button(root3,text="BACK",bg='#191919',fg='white',command =
enter_codeback)##command
b2.place(x=680,y= 350)
root3.mainloop()

def submit_values():
    global cur,con
adnom=adno.get()
nam=lastvalue.get()
uname=uservalue.get()
pwd=pwdvalue.get()
conpwd=conpwdvalue.get()
doj=datym.datetime.now()

    if pwd!=conpwd:
mbox.showinfo('Alert','Password Does Not Match!')
elif len(uname)>20 or len(pwd)>20:
mbox.showinfo('Alert!','Maximum limit is 20 characters')
elif len(uname)<5 or len(pwd)<5:
mbox.showinfo('Alert!','Minimum limit is 5 characters')
    else:
cur.execute(f"insert into adetail values({adnom},{nam},{doj},{uname},{pwd})")
con.commit()
mbox.showinfo("SUCCESS","Admin registered. Please login with your login credentials.")

def adminreg_back():
    root4.destroy()
    admin()

def admin_registration():
    global root4,adno,lastvalue,uservalue,pwdvalue,conpwdvalue
    try:
        root3.destroy()
    except:
        pass

```

```

root4 = Tk()
root4.geometry("1000x700")
adno=StringVar()
lastvalue=StringVar()
uservalue=StringVar()
pwdvalue=StringVar()
conpwdvalue=StringVar()

def intrback():
    root2.destroy()
newroot()

bg=PhotoImage(file='zenregis.png')
Label(root4,image=bg).place(relwidth=1,relheight=1)
Label(root4,text='Name:',bg='#39322C',fg='white').place(x=540,y=200)
frstname=Entry(root4,width=30,textvariable=lastvalue).place(x=650,y=200)

Label(root4,text='Adno:',bg='#39322C',fg='white').place(x=540,y=250)
adno1=Entry(root4,width=30,textvariable=adno).place(x=650,y=250)

Label(root4,text='Username:',bg='#38302D',fg='white').place(x=540,y=300)
username=Entry(root4,width=30,textvariable=uservalue).place(x=650,y=300)

Label(root4,text='Password:',bg='#332B28',fg='white').place(x=540,y=325)
passwd=Entry(root4,width=30,textvariable=pwdvalue,show='•').place(x=650,y=325)

Label(root4,text='Confirm password:',bg='#342D27',fg='white').place(x=540,y=350)
confirmPWD=Entry(root4,width=30,textvariable=conpwdvalue,show='•').place(x=650,y=350)

sub=PhotoImage(file='subbutton.png')
submit1=Button(root4,image=sub,bg='black',height=50,width=175,command =
submit_values)
submit1.place(x=650,y=450)

back=PhotoImage(file='backbtn.png')
intr=Button(root4,bg='#16120F',image=back,height=30,width=80, command =
adminreg_back)##command

```

```

intr.grid(row=1,column=1)
    root4.mainloop()

def studentreg_back():
    root2.destroy()
main_window()

def admin_back():
    root1.destroy()
main_window()

def validate_resetpwd():
usr = rusr.get()
pwd= rpwd.get()
cpwd = rcpwd.get()
notp = rotp.get()

    if usr and pwd and cpwd and notp:
        if int(notp) == int(otp):
            s = f'select * from studetail where Username = "{usr}"'
cur.execute(s)

        r = cur.fetchone()
        if r:
            if pwd == cpwd:
                s = f'updatestudetail set Password = "{pwd}" where Username = "{usr}"'
cur.execute(s)
con.commit()

mbox.showinfo('SUCCUESS','Password updated successfully')
        else:
            mbox.showinfo('ERROR','Passwords do not match!')
        else:
            mbox.showinfo('ERROR','Invalid Username')
        else:
            mbox.showinfo('ERROR','Enter correct OTP')
        else:
            mbox.showinfo('ERROR','Enter All values')

def otps(otp):
mbox.showinfo('OTP',f"Your OTP for password reset is {otp}")

```

```
def speakagain():
    global c
    speak(c)
    c=""

def forgotpassword():
    global root10,otp,rotp,rusr,rpwd,rcpwd,c
    try:
        root2.destroy()
    except:
        pass
    otp = random.randint(1000,9999)

    root10=Tk()

    root10.geometry('1000x700')

    rusr=StringVar()
    rpwd = StringVar()
    rcpwd= StringVar()
    rotp = StringVar()

    pic=PhotoImage(file='zenauth2.png')
    Label(root10,image=pic).place(relwidth=1,relheight=1)

    e = Entry(root10,textvariable = rusr)
    e.place(x=520,y=240)

    en = Entry(root10,textvariable = rpwd,show='•')
    en.place(x=520,y=275)

    e = Entry(root10,textvariable = rcpwd,show='•')
    e.place(x=610,y=310)

    e = Entry(root10,textvariable = rotp)
    e.place(x=460,y=360)
```

```

sub3=PhotoImage(file='sub3.png')
bu =
Button(root10,image=sub3,bg='white',height=25,width=150,command=validate_resetpwd)
bu.place(x=320,y=500)
    sub2=PhotoImage(file='backw.png')
bu = Button(root10,image=sub2,bg='white',height=25,width=150,command=pwdback)
bu.place(x=500,y=500)

##    Label(root10,text=f'YOUR OPT IS:{otp}',fg='red').grid(row = 6,columnspan = 2)
c=f'YOUR O..T..P.. IS.. {otp}'
speak(c)
sub=PhotoImage(file='rpotp.png')
spa = Button(root10,image=sub,bg='white',height=25,width=150,command=speakagain)
spa.place(x=680,y=500)
##    otps(otp)
root10.mainloop()
def pwdback():
    root10.destroy()
    student()

def searchhis():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
cur=con.cursor()
p=s1.get()
if p!="":
    try:
        search2.delete(0,END)
        sql1=f'select * from issuedbooks where bookid ="{p}" and adno="{val[0]}" '
cur.execute(sql1)
r=cur.fetchall()
total=len(r)
labl.configure(text=total)
    for i in tree.get_children():
tree.delete(i)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))

```

```

con.close()
    except:
mbox.showinfo('ALERT!','Unable to search book.')

def rehis():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
cur.execute(f'select * from issuedbooks where adno="{val[0]}")'
    r=cur.fetchall()
    total=len(r)
labl.configure(text=total)
    for i in tree.get_children():
tree.delete(i)
    for i in r:
        tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))


def view_history():

con=sq.connect(host='localhost',user='root',passwd='123',database="librarymanagement")
    cur=con.cursor()
    global labl,s1,search2,tree,viewform
try:
    root11.destroy()
except:
    pass
viewform=Tk()
    #-----
    style = ttk.Style(viewform)

style.theme_use("clam")
style.configure("Treeview", background="black", fieldbackground="#D1A684",
foreground="black")
    #-----
```

```

viewform.geometry('1200x650')
viewform.title('TO SUBMIT DETAILS')
s1=StringVar()
TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)
TopViewForm.pack(side=TOP, fill=X)
LeftViewForm = Frame(viewform, width=300,bg='#FE858D')
LeftViewForm.pack(side=LEFT, fill=Y)
MidViewForm = Frame(viewform, width=600)
MidViewForm.pack(side=RIGHT)
text = Label(TopViewForm, text="Student Details", bg='#37CBBE',fg='white',font=('times', 18,'italic'), width=600)
text.pack(fill=X)
txtsearch = Label(LeftViewForm, text="BOOK ID",bg='#FE858D',fg='white',font=('times', 20,'bold','italic')).place(x=70,y=50)
search2 = Entry(LeftViewForm, font=(15),bd=3,textvariable=s1 ,width=20)
search2.place(x=50,y=100)
photo = PhotoImage(file = "search.png")
search =
Button(LeftViewForm,image=photo,bd=0,bg='#FE858D',height=40,width=100,command=search).place(x=70,y=150)

photo4 = PhotoImage(file = "refresh.png")
delete =
Button(LeftViewForm,image=photo4,bd=0,bg='#FE858D',height=40,width=100,command=refresh).place(x=60,y=390)
scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)
scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)
tree = ttk.Treeview(MidViewForm, columns=('1','2','3','4','5','6','7','8'), height=100,
yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
tree['columns']=('1','2','3','4','5','6','7','8')
tree['show']='headings'
tree.heading('1',text='Admission Number')

```

```

tree.heading('2',text='Name')
tree.heading('3',text='Class')
tree.heading('4',text='BookId')
tree.heading('5',text='Book Name')
tree.heading('6',text='D.O.I')
tree.heading('7',text='Due Date')
tree.heading('8',text='D.O.S')
tree.column('8',width=280,anchor='center')
tree.column('7',width=280,anchor='center')
tree.column('6',width=280,anchor='center')
tree.column('5',width=280,anchor='center')
tree.column('4',width=280,anchor='center')
tree.column('3',width=240,anchor='center')
tree.column('2',width=240,anchor='center')
tree.column('1',width=170,anchor='center')
tree.pack()
cur.execute(f'select * from issuedbooks where adno="{val[0]}")')
r=cur.fetchall()
for i in r:
    tree.insert("",'end',values=(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]))
cur.execute(f'select fine from studetail where Adno="{val[0]}")')
total_fine = cur.fetchone()
photo44 = PhotoImage(file = "back1.png")
search =
Button(LeftViewForm,image=photo44,height=40,width=70,bg='#FE858D',bd=0,command=history_back).place(x=80,y=460)
Label(viewform,text='TOTAL FINE :',
      bg='#FE858D',fg='white',font=('times',14,'italic')).place(x=40,y=575)
labl=Label(viewform,text=total_fine,bg='#FE858D',fg='white',font=('times',20,'bold'))
labl.place(x=140,y=600)
viewform.mainloop()

def history_back():
    viewform.destroy()
    student_page()
def hlogout():
    root11.destroy()

```

```

student()
def student_page():
    global root11,val
    try:
        root2.destroy()
    except:
        pass
    root11 =Tk()
    root11.geometry('1000x700')
    s = f'select * from studetail where Username = "{studentusr}"'
    cur.execute(s)
    val = cur.fetchone()
    bg=PhotoImage(file='woood.png')
    Label(root11,image=bg).place(relwidth=1,relheight=1)
    Label(root11,text= 'STUDENT
DETAILS',fg='white',font=('times',30,'italic'),bg='#95512B').place(x =350, y=180)
    Label(root11,text= '=*24,fg='white',font=('times',16,'italic'),bg='#95512B').place(x =350,
y=220)
    Label(root11,text= 'Addmission
No:',fg='white',bg='#95512B',font=('times',16,'italic')).place(x =350, y=250)
    Label(root11,text= f'{val[0]}',fg='white',bg='#95512B',font=('times',16,'italic')).place(x
=550, y=250)
    Label(root11,text= 'Name:',fg='white',bg='#95512B',font=('times',16,'italic')).place(x =350,
y=300)
    Label(root11,text= f'{val[1]}',fg='white',bg='#95512B',font=('times',16,'italic')).place(x
=550, y=300)
    Label(root11,text= 'D.O.B:',fg='white',bg='#95512B',font=('times',16,'italic')).place(x =350,
y=350)
    Label(root11,text= f'{val[2]}',fg='white',bg='#95512B',font=('times',16,'italic')).place(x
=550, y=350)
    Label(root11,text= 'Class:',fg='white',bg='#95512B',font=('times',16,'italic')).place(x =350,
y=400)
    Label(root11,text= f'{val[3]}',fg='white',bg='#95512B',font=('times',16,'italic')).place(x
=550, y=400)
    Label(root11,text= 'Fine
amount:',fg='white',bg='#95512B',font=('times',16,'italic')).place(x =350, y=450)

```

```

Label(root11,text= f'{val[6]}',fg='white',bg='#95512B',font=('times',16,'italic')).place(x=550, y=450)

pic2=PhotoImage(file='hstrvw.png')
b = Button(root11,image=pic2,width=165,height=25,bg='black',command = view_history)
b.place(x=400,y=550)
pic=PhotoImage(file='logout.png')
b1 = Button(root11,image=pic,width=175,height=25,bg='black',command = hlogout)
b1.place(x=400,y=600)
root11.mainloop()

def validate_student_login():
    global studentusr
    studentusr = susername.get()
    pwd = spwd.get()
    s = f"select * from studetail where Username='{studentusr}' and Password = '{pwd}'"
    cur.execute(s)
    result = cur.fetchone()
    if result:
        student_page()
    else:
        mbox.showinfo("ERROR",'Invalid Login Credentials')

#-----#-----#-----#-----##

def student():
    global root2,susername,spwd
    try:
        root.destroy()
    except:
        pass
    root2=Tk()
    root2.title('Student Page')
    root2.geometry('1000x700')

```

```

susername=StringVar()
spwd=StringVar()
bg=PhotoImage(file='wlecome.png')
Label(root2,image=bg).place(relwidth=1,relheight=1)

username=Entry(root2,width=30,textvariable=susername).place(x=450,y=300)
password=Entry(root2,width=30,textvariable=spwd,show='•').place(x=450,y=325)

bck=PhotoImage(file='backbtn.png')

Backbtn=Button(root2,bg='#2A382B',image=bck,height=30,width=80,command=studentreg_
back)
Backbtn.grid(row=1,column=1)

login=PhotoImage(file='login.png')
loginbtn=Button(root2,image=login,bg='black',height=25,width=175,command =
validate_student_login)
loginbtn.place(x=450,y=360)
fpwd=PhotoImage(file='frgtpwd.png')
pwdresetbutton = Button(root2,image=fpwd,bg='#4C98BE',height=30,width=160,command =
forgotpassword)
pwdresetbutton.place(x = 460,y = 410)

root2.mainloop()

def submitstud():
    global cur
    global con
    idstud=stdvalue.get()
    nam=stdlastvalue.get()
    dob=stddobvalue.get()

```

```

classec=stdclassvalue.get()
uname=stduservalue.get()
pwd=stdpwdvalue.get()
conpwd=stdconpwdvalue.get()

if pwd!=conpwd:
    mbox.showinfo('Alert','Password Does Not Match!')
elif len(uname)>20 or len(pwd)>20:
    mbox.showinfo('Alert!','Maximum limit is 20 characters')
elif len(uname)<5 or len(pwd)<5:
    mbox.showinfo('Alert!','Minimum limit is 5 characters')
else:
    st='insert into studetail
values({}, "{}", "{}", "{}", "{}", "{}", 0)'.format(idstud,nam,dob,classec,uname,pwd)
    cur.execute(st)
    con.commit()
    mbox.showinfo("success","Student registration successful!")

def student_back():
    root5.destroy()
    admin()

def studregister():
    global
root5,stdvalue,stdlastvalue,stddobvalue,stduservalue,stdclassvalue,stdpwdvalue,stdconpwdval
ue
    try:
        root1.destroy()
    except:
        pass
    root5=Tk()
    root5.title(' Page')
    root5.geometry('1000x700')

    stdvalue=StringVar()
    stdlastvalue=StringVar()
    stddobvalue=StringVar()

```

```

stduservalue=StringVar()
stdclassvalue=StringVar()
stdpwdvalue=StringVar()
stdconpwdvalue=StringVar()

bg=PhotoImage(file='zenregis.png')
Label(root5,image=bg).place(relwidth=1,relheight=1)

Label(root5,text='Admission Number:',bg='#4B4138',fg='white').place(x=540,y=200)
frstname=Entry(root5,width=30,textvariable=stdvalue).place(x=650,y=200)

Label(root5,text='Name:',bg='#39322C',fg='white').place(x=540,y=225)

lastname=frstname=Entry(root5,width=30,textvariable=stdlastvalue).place(x=650,y=225)

Label(root5,text='Dob(YYYY-MM-DD):',bg='#413A33',fg='white').place(x=540,y=250)
DOB=Entry(root5,width=30,textvariable=stddobvalue).place(x=650,y=250)

Label(root5,text='Class:',bg='#342D27',fg='white').place(x=540,y=275)
classs=Entry(root5,width=30,textvariable=stdclassvalue).place(x=650,y=275)

Label(root5,text='Username:',bg='#38302D',fg='white').place(x=540,y=300)
username=Entry(root5,width=30,textvariable=stduservalue).place(x=650,y=300)

Label(root5,text='Password:',bg='#332B28',fg='white').place(x=540,y=325)
passwd=Entry(root5,width=30,textvariable=stdpwdvalue,show='•').place(x=650,y=325)

Label(root5,text='Confirm password:',bg='#342D27',fg='white').place(x=540,y=350)
confirmpwd=Entry(root5,width=30,textvariable=stdconpwdvalue,show='•').place(x=650,y=350)

sub=PhotoImage(file='subbutton.png')

submit=Button(root5,image=sub,bg='black',height=20,width=175,command=submitstud)
submit.place(x=650,y=450)

```

```
back=PhotoImage(file='backbtn.png')

intr=Button(root5,bg='#16120F',image=back,command=student_back,height=30,width=80)
intr.grid(row=1,column=1)

root5.mainloop()

def main_window():

    global root
    root=Tk()
    root.title('Enter as')
    root.geometry('1000x700')

    pic1=PhotoImage(file='zenbg.png')
    Label(root,image=pic1).place(relwidth=1,relheight=1)

    btpic1=PhotoImage(file='admin.png')
    b1=Button(root,text='ADMIN',padx=30,pady=15,bg='#372515',image=btpic1,
    command=admin)
    b1.place(x=460,y=300)

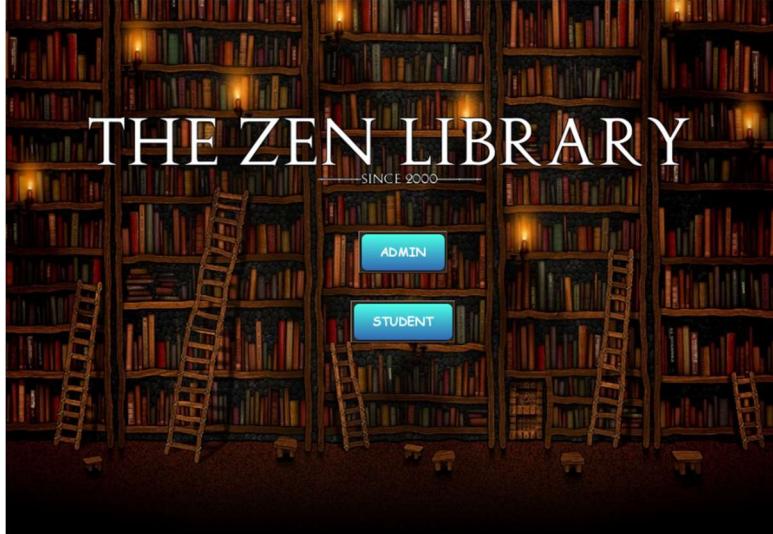
    btpic=PhotoImage(file='student.png')
    b2=Button(root,text='STUDENT',padx=30,pady=15,bg='#372515',image=btpic,command
    = student)
    b2.place(x=450,y=390)

    root.mainloop()

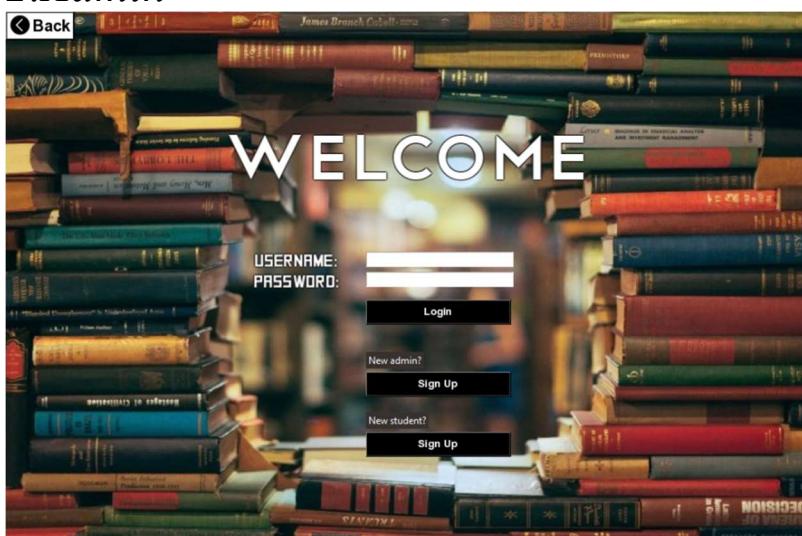
if __name__ == "__main__":
    main_window()
```

# Outputs

## 1. Front page



## 2. Admin



## 3. Admin → admin signup



#### 4. Admin → admin signup→admin registration

Back

# THE ZEN LIBRARY

SINCE 2000

## REGISTRATION

**SUCCESS**

Admin registered. Please login with your login credentials.

Name: Mukundh  
Adno: 2001  
Username: mukundh  
Password: \*\*\*\*\*  
Confirm password: \*\*\*\*\*

OK

Submit

#### 5. Admin → student registration

Back

# THE ZEN LIBRARY

SINCE 2000

## REGISTRATION

Admission Number:  
Name:  
Dob(YYYY-MM-DD)  
Class:  
Username:  
Password:  
Confirm password:

Submit

#### 6. Admin→Login

Back

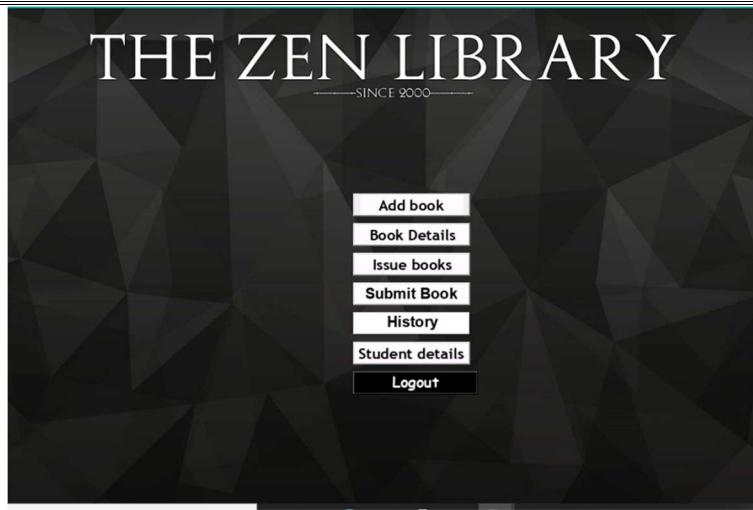
# WELCOME

USERNAME: mukundh  
PASSWORD: \*\*\*\*\*

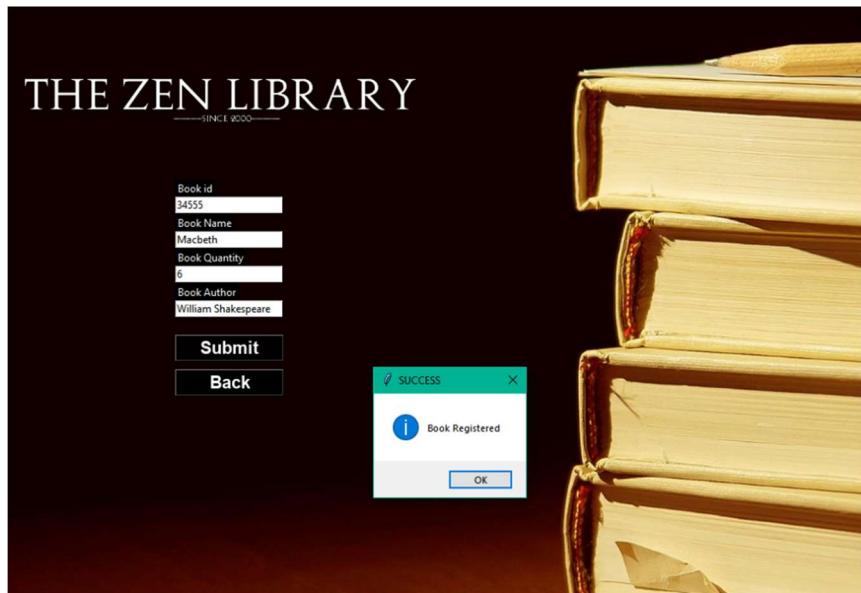
Login

New admin?  
Sign Up

New student?  
Sign Up



7. Admin → Login → Add book



8. Admin → Login → Book details → (search 34555)

Book Details			
Book Id	Book Name	Supplied Quantity	Available Quantity
1	harry potter	10	9
2	rli	100	99
3	cloudy	10	9
34555	Macbeth	6	6

BOOK ID: 34555

search delete Refresh back

TOTAL RESULTS FOUND: 4

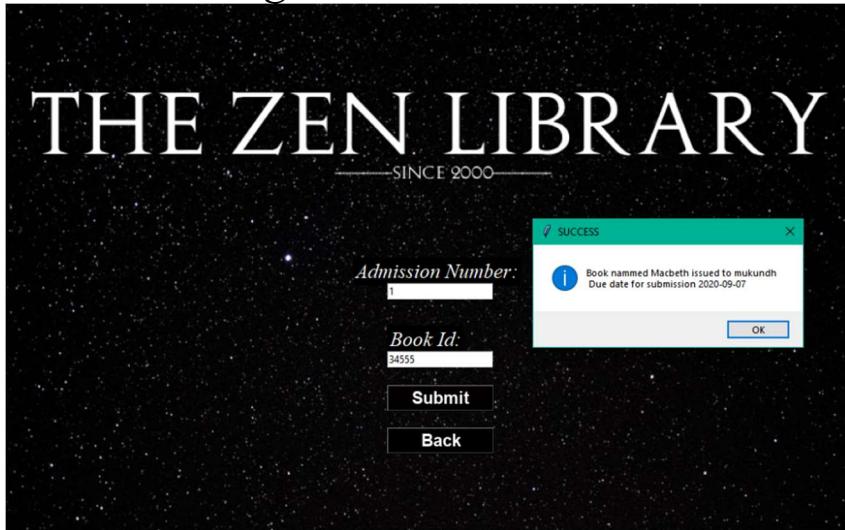
Book Details			
Book Id	Book Name	Supplied Quantity	Available Quantity
34555	Macbeth	6	5

BOOK ID:

search delete Refresh back

TOTAL RESULTS FOUND: 1

## 9.Admin→Login→Issue Books



## 10.Admin→Login→Submission Details

Submission Details				Submission Details				D.O.S
Admission Number	Name	Class	BookId	Book Name	D.O.I	Due Date		
1	mukundh	12C	34555	Aecheth	2020-08-29	2020-09-07		
3760	Victor	12d	34555	Aecheth	2020-08-28	2020-09-07		
9860	Sameer	12a	34555	Aecheth	2020-08-28	2020-09-07		

## 11.Admin→Login→Submission Details→(Submit sameer)

Submission Details				
Admission Number	Name	Class	BookId	
1	mukundh	12C	34555	
3760	Victor	12d	34555	

## 12. Admin→Login→History

History				History			
Admission Number	Name	Class	BookId	Book Name	D.O.I	Due Date	D.O.S
2020	mukundh	12	3	Soudy	2020-09-23	2020-09-02	2020-08-23
1234	qverty	12	3	Soudy	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	3	Soudy	2020-09-23	2020-09-02	2020-08-23
1234	qverty	12	3	Soudy	2020-09-23	2020-09-02	2020-08-23
2020	qverty	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	2	rl	2020-09-23	2020-08-02	2020-08-23
2020	mukundh	12	3	rl	2020-09-23	2020-08-02	2020-08-23
1234	qverty	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
2020	mukundh	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
1234	qverty	12	1	ry potter	2020-09-23	2020-09-02	2020-08-23
1	mukundh	12C	34555	sceth	2020-09-28	2020-09-07	2020-08-23
3760	Victor	12d	34555	sceth	2020-09-28	2020-09-07	2020-08-23
9860	Sameer	12a	34555	sceth	2020-09-28	2020-09-07	2020-08-28
9960	Sameer	12a	34555	sceth	2020-09-28	2020-09-07	2020-08-28

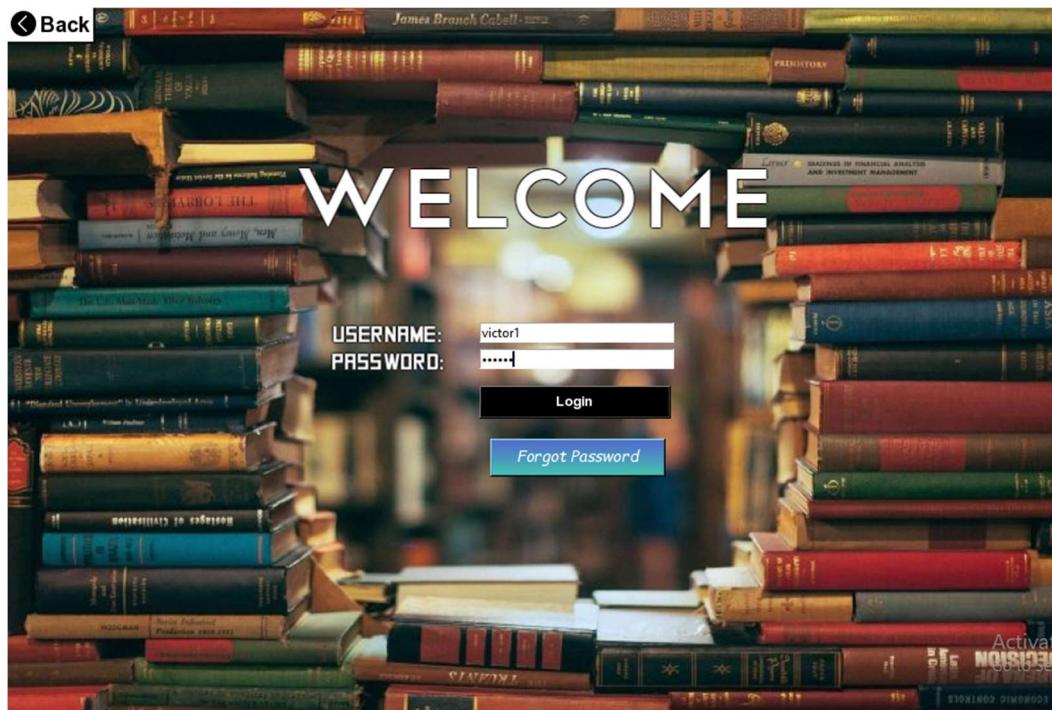
## 13. Admin→Login→Student Details

Student Details					
Admission Number	Name	D.O.B	Class	Fine	
1	mukundh	2003-03-17	12C	300	
3760	Victor	2003-04-17	12d	0	
9860	Sameer	2003-08-27	12a	60	

## 14. Admin→Login→Submission Details→(Pay fine -mukundh)

Student Details				
Name	D.O.B	Class	Fine	
mukundh	2003-03-17	12C	0	
Victor	2003-04-17	12d	0	
Sameer	2003-08-27	12a	60	

## 15.Student



## 16.Student→Login

The image shows a student details page titled "STUDENT DETAILS" on a wooden-paneled background. The page displays the following information:

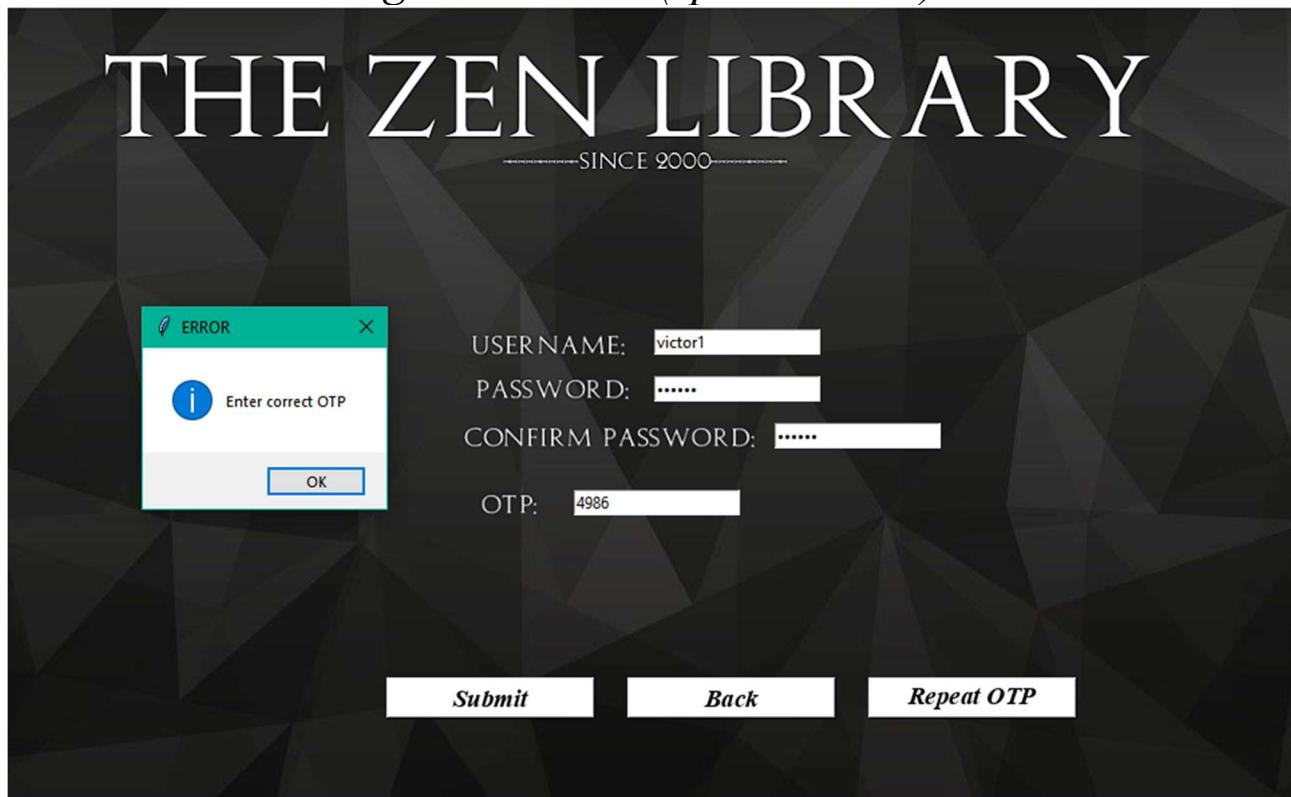
<i>Addmission No.:</i>	3760
<i>Name:</i>	Victor
<i>D.O.B.:</i>	2003-04-17
<i>Class:</i>	12d
<i>Fine amount:</i>	0

At the bottom of the page are two buttons: a purple "View History" button and a black "Logout" button.

## 17. Student→Login→View History

Student Details				Student Details			
BOOK ID	Admission Number	Name	Class	BookId	Book Name	D.O.I	Due Date
	3760	Victor	12d	34555	Macbeth	2020-08-28	2020-09-07
<input type="text"/>							
<input type="button" value="search"/>							
<input type="button" value="Refresh"/>							
<input type="button" value="back"/>							
TOTAL FINE :	0						

## 18. Student→Forgot Password (speaks 4985 )



# **USER MANUAL**

## **About the application:**

This application, Library Management System, was created with the intention to make library management easier through a graphical user interface in Python known as tkinter, it also displays various other details about the students, books that contribute.

## **Requirements for executing the project:**

1. Python 3.7 (64-bit) with libraries: tkinter, mysql.connector, pyttsx3
2. Mysql 8.0 Command Line Client

## **What can be done using this application:**

The Library Management application has both student and admin terminal.

- In admin terminal:

It is possible to delete a student from database with confirmation from the GUI, update submission status, update fine status, search for a specific students' details or create a student id. It is also possible to view the history of issued books by a person or in all, issue a book, add a book and display student details, book details, and status of fine and submission.

- In student terminal:

Has permissions to register a student on username -password basis providing valid and necessary credentials.

Displays student details i.e., the books issued previously, pending submission, last date and fine pending. Shows the history of books issued and has limited permissions, change password made simpler.

### Current student info:

Username: victor1  
Password: jeyjey

### Current Admin info:

Username: Mukundh  
Password: mukundh

## CONCLUSION:

The purpose of the project is to develop a program which manages all the information on booking, display, editing and deleting of various details in an Library management system this also helps in reducing the manual work managing all these information, improve the efficiency on bookings and management and provide a user friendly interface .

## Further enhancements:

- ❖ Reminders, Updates via SMS, Mail.
- ❖ Integration with AI.
- ❖ Payments through Net Banking, UPI.
- ❖ Book Reservations
- ❖ Speech recognition, Chat-bot service.
- ❖ Audio Books, Podcast subscriptions.
- ❖ Automated Book recommendations.

## **BIBLIOGRAPHY**

### Text Books:

- ✓ COMPUTER SCIENCE with Python XI-by SUMITA ARORA  
(2019-20 edition, Dhanpat rai & co publications)
- ✓ COMPUTER SCIENCE with Python XII-by SUMITA ARORA  
(2020-21 edition, Dhanpat rai & co publications)

### Web sites:

- ✓ <https://engineering.eckovation.com/forum/t/what-is-tkinter-in-Python-how-it-can-be-used-and-why/146>
- ✓ [https://www.tutorialspoint.com/Python/Python\\_gui\\_programming.htm](https://www.tutorialspoint.com/Python/Python_gui_programming.htm)
- ✓ <https://pynative.com/Python-mysql-database-connection>  
<https://www.geeksforgeeks.org › Python-gui-tkinter>

### Video links:

- ✓ <https://www.youtube.com/watch?v=4k9CphTdnWE&t=1249s>
- ✓ <https://www.youtube.com/watch?v=ZGf2e-1J-Io&list=PLUqe82V0MEeYUEFqf48ljFgDLBte4f0GF>