|  |
| --- |
| **LIBRARY MANAGEMENT SYSTEM**  **21CSC203P– ADVANCED PROGRAMMING PRACTICE**  **Project Report**  *Submitted by*  **Archisman Hes [Reg. No.: RA2211003010273]**  **B.Tech. CSE - Core**  **Saloni Bhardwaj [Reg. No.: RA2211003010268]**  **B.Tech. CSE – Core**  **Shovik Banerjee [Reg. No.: RA2211003010270] B.Tech. CSE - Core**  **SRMIST-01.jpg**  **SCHOOL OF COMPUTING**  **COLLEGE OF ENGINEERING AND TECHNOLOGY**  **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  **(Under Section 3 of UGC Act, 1956)**  S.R.M. NAGAR, KATTANKULATHUR – 603 203  KANCHEEPURAM DISTRICT  **December 2022** |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Bonafide Certificate | 3 |
| 2 | Acknowledgement | 4 |
| 3 | Problem statement | 5 |
| 4 | Abstract | 5 |
| 5 | Methodology / Procedure | 7 |
| 6 | Architecture and design | 8 |
| 7 | Flow Chart | 12 |
| 8 | Source code | 20 |
| 9 | Results/Outputs | 41 |
| 10 | Conclusion | 48 |
| 11 | Bibliography | 49 |

Logo, company name

Description automatically generated

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

**BONAFIDE CERTIFICATE**

Certified that this Course Project Report titled **“LIBRARY MANAGEMENT SYSTEM”** is the Bonafide work done by **ARCHISMAN HES [RA2211003010273], SALONI BHARDWAJ [RA2211003010268] And SHOVIK BANERJEE [RA2211003010270]** who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

|  |  |
| --- | --- |
| **SIGNATURE**  **Ajanthaa Lakkshmanan**  ASSISTANT PROFESSOR  DEPARTMENT OF COMPUTING TECHNOLOGIES  SRM Institute of Science and Technology | **HEAD OF THE DEPARTMENT**  **Dr. M. Pushpalatha**  PROFESSOR & HEAD  DEPARTMENT OF COMPUTING TECHNOLOGIES  SRM Institute of Science and Technology |

# ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty Ajanthaa **Lakkshmanan , Ast. Professor , Department of Computing Technologies,** for his/herassistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. M. Pushpalatha PROFESSOR & HEAD**

**DEPARTMENT OF COMPUTING TECHNOLOGIES and** my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

**PROBLEM STATEMENT**

SRM's central library requires an effective system for managing the records of all the books in the library as well as the students currently enrolled in the institute.

**ABSTRACT**

* A library management system is the most proficient and easy to use system for managing all the processes involved in a Library in the most effective ways.
* This system will reduce all the manual work and the whole process can be managed just through single clicks and edits.
* There will be no headache and doubtfulness of storing the data securely and searching the records of any individual afterward.
* Any book seeker can rent a book just by signing in with their details, and return it with the date of returning.
* The staff can also facilitate themselves with some extra authorizations and privileges.
* Only, one person is required to take care of the whole system, without any chances of mistakes.

**Advantage of the library management system:**

* It reduces the manual paperwork through it and gives proper information of books has been recorded automatically.
* Librarian can update the information of books and manage availability and arriver record of the books.
* It saves human efforts and time.

With the help of library management software, the customer can easily search and find the books.

The library management system is nowadays essential for schools, colleges, private libraries, and other organizations. They can use this software as the purpose of books issuing and returning for renewal. Moreover, this software helps the librarian to maintain information about book issuing and returning the book before the last date.

Library management system are useful for such organization:

- School libraries

- Collage libraries

- Private libraries

- Reference libraries

- National libraries

- Public libraries

- Modern Public libraries

- Public leading libraries

**METHODOLOGY**

Proposed system is an automated Library Management System intended for SHANNEN SCHOOL, part of final practical assessment for Computer Science. Through this software user can

* **Add new student**
* **Add new book**
* **Issue book to student**
* **Return book from student**
* **View all students**
* **View all books**
* **View issued books**

All the manual difficulties in managing the Library will be rectified by implementing this application.

**FEASIBILITY ANALYSIS**

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happen in any organization by the development of a system. This impact can either be positive or negative. When the positive dominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as TECHNICAL FEASIBILITY and ECONOMIC FEASIBILITY.

* TECHNICAL FEASIBILITY

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of this software as well as maintenance is available in the school. We will be utilizing the resources which are already available.

* ECONOMIC FEASIBILITY

As this software is built for the final submission for practical assignment for Computer Science, economic feasibility study is not required at all. If the same is accepted by the school for use in the school library, then only one thing is to be done is making an environment for the development with an effective supervision. If it is done so, we can attain maximum usability of the corresponding resources.

**ARCHITECTURE AND DESIGN**

* Problem definition analyzed in depth.
* Database modelling done.
* Feasibility study made
* Table structure prepared
* Database created in SQL server
* E – R diagram prepared.
* Sample data collected and stored in table.
* Flow chart prepared
* Coding for every module prepared
* Module tested in stand alone mode
* Integrated all modules
* Integration testing done
* Bug fixed
* Prepared all the files for distribution.

LIBRARY MANAGEMENT (Master Business Process)

Transactions

Sub Process 2

Sub Process 1

Issue

Return

Books

Student

Student :

* + Collecting the student registration form
  + Checking for fees payment
  + Checking for Principle’s approval
  + Once accepted, student details are entered and unique student ID is given to student for future use.
  + Student enrolled in system. Transactions can now begin.

Book :

* + Book received physically in library
  + Details checked with invoices
  + Principle’s approval sought
  + Book details entered in system. Transactions can now begin

Transactions :

Issue Book :

* + Check student ID
  + Check book ID whether available or not.
  + Issue book to student, if available.

Return Book

* + Check book ID
  + Return book to library.

**E – R Diagram**

Title

Author

Available

Book ID

Book

SID

Borrowed by

Sname

Student

Issue

Sclass

**Database modelling**

library

1

1

books

***bookID***

title

author

available

student

***sID***

sname

sclass

issue

sID

bID

∞

∞

**RDBMS** – MySQL 8.0

**Database Name** – library

**Table**: student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default |
| sid | Varchar (5) | NO | PRI | NULL |
| sname | Varchar (50) | NO |  | NULL |
| sclass | Varchar (4) | NO |  | NULL |

**Table:** books

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default |
| bid | Varchar (10) | NO | PRI | NULL |
| title | Varchar (50) | NO |  | NULL |
| author | Varchar (50) | NO |  | NULL |
| available | Varchar (5) | YES |  | YES |

**Table**: issue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default |
| bid | Varchar (10) | NO |  | NULL |
| Sid | Varchar (10) | NO |  | NULL |

**FLOW CHART**

Main Menu

**F**

A

Delete Book

Add Book

Add Student

**G**

**B**

Delete Student

**H**

**C**

View Student

Issue Book

**I**

**D**

View Issued Books

Return Book

**E**

View Book

Add book:

A

Enter bid, title, author, available

Is Data Valid?

NO

Error message

YES

Write data in book table

MAIN MENU

Success message

Delete book:

Enter book ID

B

Book found?

NO

Error Message

YES

Delete book details from book table

MAIN MENU

Success Message

Issue book:

C

Enter bID, sID

Book available?

NO

YES

Student valid?

NO

YES

Error message

Issue book to student. Data stored in issue table

MAIN MENU

Success message

Return book:

D

Enter bID

BookID found in issue table?

NO

Error message bID

YES

Take return. Data deleted from issue table. Field available=”YES” in book table

MAIN MENU

Success message

View All Books:

Retrieve all records from book table

MAIN MENU

Show all data

E

Add student:

F

Enter sid, sname, sclass

Is Data Valid?

NO

Error message

YES

Write data in student table

MAIN MENU

Success message

Delete Student:

Enter student ID

G

Error Message

Student found?

NO

YES

Delete student details from student table

MAIN MENU

Success Message

View All Students:

Retrieve all records from student table

MAIN MENU

Show all data

H

View Issued Books:

G

Retrieve all records from issue table

Join issue table with student and book table

Display all details of issued books

MAIN MENU

**SOURCE CODE**

**HOME.PY (Main Menu)**

from tkinter import \*

import mysql.connector

from add import \*

from delete import \*

from issue import \*

from Return import \*

from view import \*

from add\_student import \*

from delete\_student import \*

from view\_student import \*

from view\_issued import \*

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004',database='library')

cursor = db.cursor()

window=Tk()

window.geometry("650x500")

#bg=photoimage(file="background.jpg")

window.title("SRM Central Library Management System")

#label1=label(root,image=bg)

#label1.place(x=0,y=0)

greet = Label(window, font = (‘times new roman’, 30, 'bold'), fg="red", text = "Welcome to SRM Central Library")

greet.grid(row = 1,columnspan = 3)

addbtn=Button(window,text="Add Books",command=addBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

addbtn.grid(row=3,column=0)

deletebtn=Button(window,text="Delete Books",command=deleteBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

deletebtn.grid(row=5,column=0)

issuebtn=Button(window,text="Issue Books",command=issueBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

issuebtn.grid(row=7,column=0)

returnbtn=Button(window,text="Return Books",command=returnBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

returnbtn.grid(row=9,column=0)

viewbtn=Button(window,text="View Books",command=viewBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

viewbtn.grid(row=11,column=0)

addstudbtn=Button(window,text="Add Students",command=addStudents,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

addstudbtn.grid(row=3, column=2)

delstudbtn=Button(window,text="Delete Students",command=deleteStudents,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

delstudbtn.grid(row=5, column=2)

viewstudbtn=Button(window,text="View Students",command=viewStudents,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

viewstudbtn.grid(row=7, column=2)

viewissuedbtn=Button(window,text="View Issued Books",command=viewIssuedBooks,bg="DodgerBlue2",fg="white",font = ('arial', 20, 'bold'))

viewissuedbtn.grid(row=9, column=2)

greet1 = Label(window, font = ('arial', 15, 'bold'), fg='red',text = "")

greet1.grid(row = 13,columnspan = 3)

greet2 = Label(window, font = ('arial', 15, 'bold'), fg='red',text = " ")

greet2.grid(row = 15,columnspan = 3)

greet3 = Label(window, font = ('arial', 15, 'bold'), fg='red',text = "")

greet3.grid(row = 17,columnspan = 3)

greet4 = Label(window, font = ('arial', 15, 'bold'), fg='red',text = "developed by : ARCHISMAN HES")

greet4.grid(row = 19,columnspan = 3)

window.mainloop()

**ADD.PY (Addition of new books)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def add\_db():

global id

global title

global author

bid=id.get()

btitle=title.get()

bauthor=author.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004’,database='library')

cursor = db.cursor()

print(bid,end='--------')

print(btitle,end='------------------------------------')

print(bauthor,end='-----------------------------------')

print("add")

sqlquery= "insert into books values('" + bid +"','"+btitle+"','"+bauthor+"','YES');"

print(sqlquery)

try:

if bid=="" or btitle=="":

messagebox.showinfo("Error","Book ID / Title can not be blank")

else:

cursor.execute(sqlquery)

db.commit()

messagebox.showinfo('Success',"Book added Successfully")

except:

messagebox.showinfo("Error","Duplicate / Invalid Book ID.... Cannot add into Database")

window.destroy()

def addBooks():

global id

global title

global author

window=Tk()

window.title(SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'), text = "Add Books")

greet.grid(row = 0,columnspan = 10)

#----------id-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Book id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=10,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

#----------title-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Title: ")

L.grid(row = 4, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 4, column = 2)

title=Entry(window,width=30,font =('arial', 15, 'bold'))

title.grid(row=4,column=3)

#----------author-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Author: ")

L.grid(row = 6, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 6, column = 2)

author=Entry(window,width=30,font =('arial', 15, 'bold'))

author.grid(row=6,column=3)

submitbtn=Button(window,text="Submit",command=add\_db,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)

print("add")

pass

**DELETE.PY (Deletion of books)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def delete\_db():

global id

bid=id.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004’,database='library')

cursor = db.cursor()

print(bid,end='--')

print("delete")

try:

sqlquery="select \* from books where bid='"+bid+"';"

print(sqlquery)

cursor.execute(sqlquery)

cursor.close

flag=0

for i in cursor:

if (i[0]==bid):

flag=1

print(flag)

print (bid)

break;

if flag==1:

delquery1="delete from books where bid='"+bid+"';"

print(delquery1)

cursor.execute(delquery1)

db.commit()

messagebox.showinfo('Success',"Student removed Successfully")

else:

messagebox.showinfo("Error","Student not found")

except:

messagebox.showinfo("Error","Student with given id does not exist")

#window.destroy()

def deleteBooks():

global id

window=Tk()

window.geometry("500x300")

window.title('SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'), text = "Delete Books")

greet.grid(row = 0,columnspan = 3)

#----------id-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Book id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=5,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

submitbtn=Button(window,text="Submit",command=delete\_db,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)

print("delete")

pass

**ISSUE.PY (Issue book to Student)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def issue\_db():

global id

global Sid

bid=id.get()

bStudentID=Sid.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004’

,database='library')

cursor = db.cursor()

print(bid,end='--')

print(bStudentID,end='--')

print("issue")

try:

checkavailability="select \* from books where available='YES' AND bid='"+bid+"';"

print(checkavailability)

cursor.execute(checkavailability)

flag=0

for i in cursor:

print(i[0])

if(i[0]==bid):

flag=1

break;

checkstudent="select \* from student where sid='"+bStudentID+"';"

print(checkstudent)

cursor.execute(checkstudent)

tag=0

for i in cursor:

print(i[0])

if(i[0]==bStudentID):

tag=1

break;

#if tag==0:

# messagebox.showinfo ("Error"."Student not found")

# break;

# if flag==0:

# messagebox.showinfo ("Error","Book not available")

# break;

if flag==1 and tag==1:

updatequery="update books set available='NO' where bid='"+bid +"';"

print(updatequery)

cursor.execute(updatequery)

db.commit()

addquery="insert into issue values('"+bid+"','"+bStudentID+"');"

print(addquery)

cursor.execute(addquery)

db.commit()

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

else:

messagebox.showinfo("Error","INVALID book ID ... ") if flag==0 else messagebox.showinfo("Error","Invalid Student")

except:

messagebox.showinfo("Error","Cannot issue given book ")

def issueBooks():

global id

global Sid

window=Tk()

window.geometry("700x200")

window.title('SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'),fg='red', text = " Issue Books")

greet.grid(row = 0,columnspan = 1000)

#----------id-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Book id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=5,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

#----------Student ID-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Student ID: ")

L.grid(row = 4, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 4, column = 2)

Sid=Entry(window,width=5,font =('arial', 15, 'bold'))

Sid.grid(row=4,column=3)

submitbtn=Button(window,text="Submit",command=issue\_db,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)

print("issue")

pass

**RETURN.PY (Return book from Student)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def return\_db():

global id

bid=id.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004’,database='library')

cursor = db.cursor()

print(bid,end='--------')

print("return")

try:

checkavailability=" select \* from books where available='NO' AND bid='"+bid+"';"

print(checkavailability)

cursor.execute(checkavailability)

flag=0

for i in cursor:

print(i[0])

if(i[0]==bid):

flag=1

break;

if flag==1:

updatequery="update books set available='YES' where bid='"+bid +"';"

print(updatequery)

cursor.execute(updatequery)

db.commit()

sqlquery= "delete from issue where bid='" + bid +"';"

print(sqlquery)

cursor.execute(sqlquery)

db.commit()

messagebox.showinfo('Success',"Book returned Successfully")

else:

messagebox.showinfo("Error","Invalid Book id")

except:

messagebox.showinfo("Error","Cannot return given book ")

def returnBooks():

global id

window=Tk()

window.geometry=("800x600")

window.title(“SRM Central Library Management System”)

greet = Label(window, font = ('arial', 30, 'bold'), fg="red",text = " Return Books")

greet.grid(row = 0,columnspan = 223)

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Book id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=5,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

submitbtn=Button(window,text="Submit",command=return\_db,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)

print("return")

window.destroy

pass

**VIEW.PY (View total List of Books)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def viewBooks():

global id

window=Tk()

window.geometry=("1000x400")

window.title(‘SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'),fg="red", text = " View Books")

greet.grid(row = 0,columnspan = 3)

db = mysql.connector.connect(host ="localhost",user = "root",password = ‘archis2004’,database='library')

cursor = db.cursor()

sqlquery= "select \* from books ;"

print(sqlquery)

try:

cursor.execute(sqlquery)

L = Label(window, font = ('arial', 20), text = "%-10s%-30s%-30s%-30s"%('BID','Title','Author','Available'))

L.grid(row = 1,column= 0)

L = Label(window, font = ('arial', 20), text = "-----------------------------------------------------------------------------------")

L.grid(row = 2,column = 0)

x=4

for i in cursor:

L = Label(window, font = ('arial', 15), text = "%-10s%-30s%-30s%-30s"%(i[0],i[1],i[2],i[3]))

L.grid(row = x,column = 0)

x+=1

except:

messagebox.showinfo("Error","Cannot Fetch data.")

print("view")

pass

**ADD\_STUDENT.PY (Add new student)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def add\_students():

global sid

global sname

global sclass

sid=id.get()

sname=name.get()

sclass=clas.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = 'archis2004’,database='library')

cursor = db.cursor()

print(sid,end='--')

print(sname,end='--')

print(sclass,end='--')

print("add")

sqlquery= "insert into student values('" + sid +"','"+sname+"','"+sclass+"');"

print(sqlquery)

try:

if sid=="" or sname=="":

messagebox.showinfo("Error","Student ID / NAme can not be blank")

else:

cursor.execute(sqlquery)

db.commit()

messagebox.showinfo('Success',"Student added Successfully")

except:

messagebox.showinfo("Error","Duplicate / Invalid Student ID.... Cannot add into Database")

window.destroy()

def addStudents():

global id

global name

global clas

window=Tk()

window.geometry=("900x400")

window.title(‘SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'), fg="red", text = "Add Students")

greet.grid(row = 0,columnspan = 10)

#----------id-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Student id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=8,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

#----------title-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Name: ")

L.grid(row = 4, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 4, column = 2)

name=Entry(window,width=30,font =('arial', 15, 'bold'))

name.grid(row=4,column=3)

#----------author-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Class: ")

L.grid(row = 6, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 6, column = 2)

clas=Entry(window,width=8,font =('arial', 15, 'bold'))

clas.grid(row=6,column=3)

submitbtn=Button(window,text="Submit",command=add\_students,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=9,columnspan=3)

print("add")

pass

**DELETE\_STUDENT.PY (Remove student)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def delete\_students():

global id

sid=id.get()

db = mysql.connector.connect(host ="localhost",user = "root",password = 'archis2004',database='library')

cursor = db.cursor()

print(sid,end='--')

print("delete")

try:

sqlquery="select \* from student where sid='"+sid+"';"

print(sqlquery)

cursor.execute(sqlquery)

cursor.close

flag=0

for i in cursor:

if (i[0]==sid):

flag=1

print(flag)

print (sid)

break;

if flag==1:

delquery1="delete from student where sid='"+sid+"';"

print(delquery1)

cursor.execute(delquery1)

db.commit()

messagebox.showinfo('Success',"Student removed Successfully")

else:

messagebox.showinfo("Error","Student not found")

except:

messagebox.showinfo("Error","Student with given id does not exist")

#root.destroy()

def deleteStudents():

global id

window=Tk()

window.geometry("500x300")

window.title(‘SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'), fg="red", text = " Delete Student")

greet.grid(row = 0,columnspan = 3)

#----------id-------------------

L = Label(window, font = ('arial', 15, 'bold'), text = "Enter Student id: ")

L.grid(row = 2, column = 1)

L = Label(window, font = ('arial', 15, 'bold'), text = " ")

L.grid(row = 2, column = 2)

id=Entry(window,width=5,font =('arial', 15, 'bold'))

id.grid(row=2,column=3)

submitbtn=Button(window,text="Submit",command=delete\_students,bg="DodgerBlue2",fg="white",font = ('arial', 15, 'bold'))

submitbtn.grid(row=8,columnspan=3)

print("delete")

pass

**VIEW\_STUDENT.PY (View total list of students)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def viewStudents():

global id

window=Tk()

window.geometry=("1000x400")

window.title(‘SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'),fg="red", text = " View Students")

greet.grid(row = 0,columnspan = 3)

db = mysql.connector.connect(host ="localhost",user = "root",password = 'archis2004',database='library')

cursor = db.cursor()

sqlquery= "select \* from student;"

print(sqlquery)

try:

cursor.execute(sqlquery)

L = Label(window, font = ('arial', 20), text = "%-10s%-30s%-30s"%(' SID',' Name',' Class'))

L.grid(row = 1,columnspan = 4)

L = Label(window, font = ('arial', 20), text = "-----------------------------------------------")

L.grid(row = 2,columnspan = 4)

x=4

for i in cursor:

L = Label(window, font = ('arial', 15), text = "%-10s%-30s%-30s"%(i[0],i[1],i[2]))

L.grid(row = x,columnspan = 4)

x+=1

except:

messagebox.showinfo("Error","Cannot Fetch data.")

print("view")

pass

**VIEW\_ISSUED.PY (View Issued bokks list to students)**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

def viewIssuedBooks():

global id

window=Tk()

window.geometry=("1000x400")

window.title(‘SRM Central Library Management System')

greet = Label(window, font = ('arial', 30, 'bold'),fg="red", text = " View Issued Books")

greet.grid(row = 0,columnspan = 3)

db = mysql.connector.connect(host ="localhost",user = "root",password = 'archis2004',database='library')

cursor = db.cursor()

sqlquery= "select \* from issue inner join student on issue.sid=student.sid inner join books on issue.bid=books.bid;"

print(sqlquery)

try:

cursor.execute(sqlquery)

L = Label(window, font = ('arial', 20), text = "%-10s%-30s%-30s%-30s%-30s"%('BID','Book Title','Author','Student Name', 'Class'))

L.grid(row = 1,column = 0)

L = Label(window, font = ('arial', 20), text = "------------------------------------------------------------------------------------------------------------------------------")

L.grid(row = 2,column = 0)

x=4

for i in cursor:

L = Label(window, font = ('arial', 20), text = "%-10s%-30s%-30s%-30s%-30s"%(i[0],i[6],i[7],i[3],i[4]))

L.grid(row = x,column = 0)

x+=1

except:

messagebox.showinfo("Error","Cannot Fetch data.")

print("view")

pass

**OUTPUT**

**PREREQUISITS:**

1. MySQL server – in my PC I have installed MySQL 8.0 with all the connectors
2. PYTHON – installed Python 3.10
3. PATH – Edited all the environment paths containing MySQL, PYTHON and my working directory

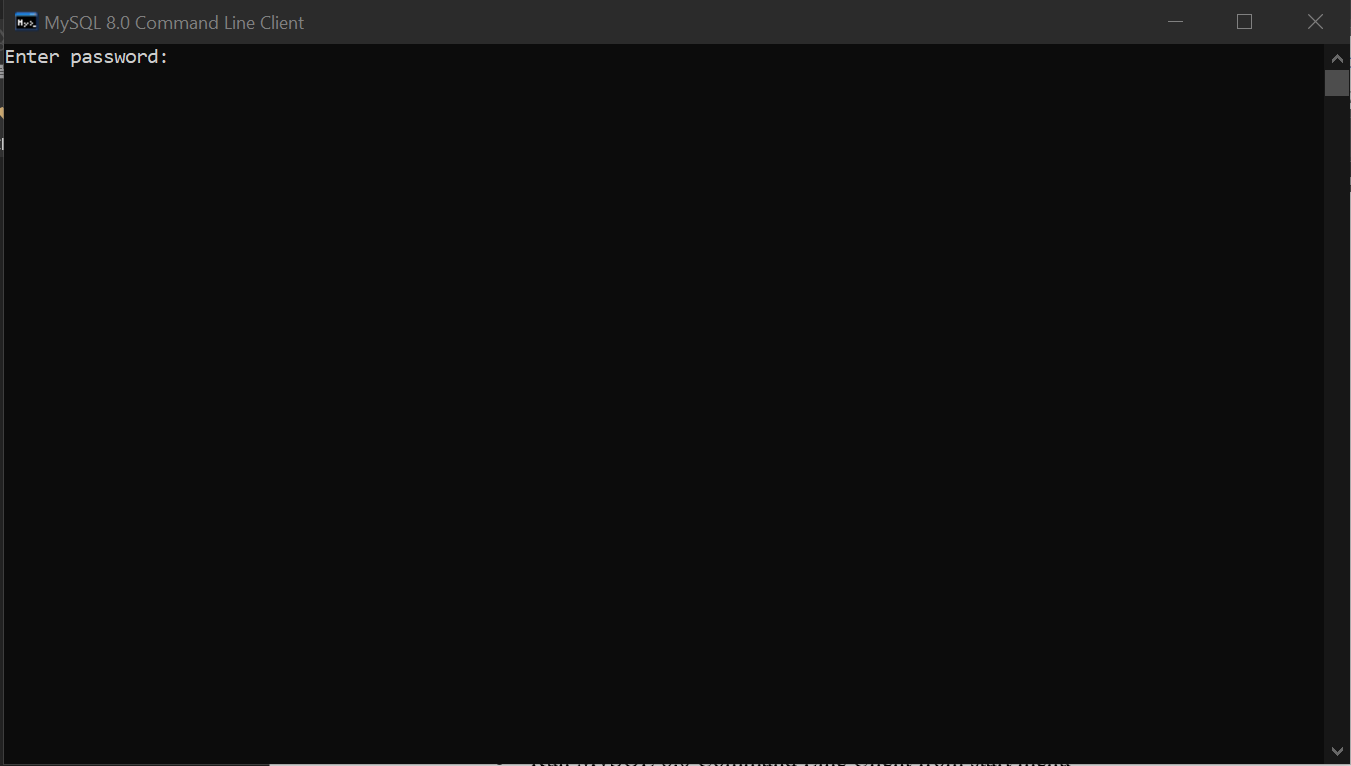
**INSTALLING MYSQL CONNECTOR FOR PYTHON:**

Run <<Command Prompt>> from windows explorer or from search bar

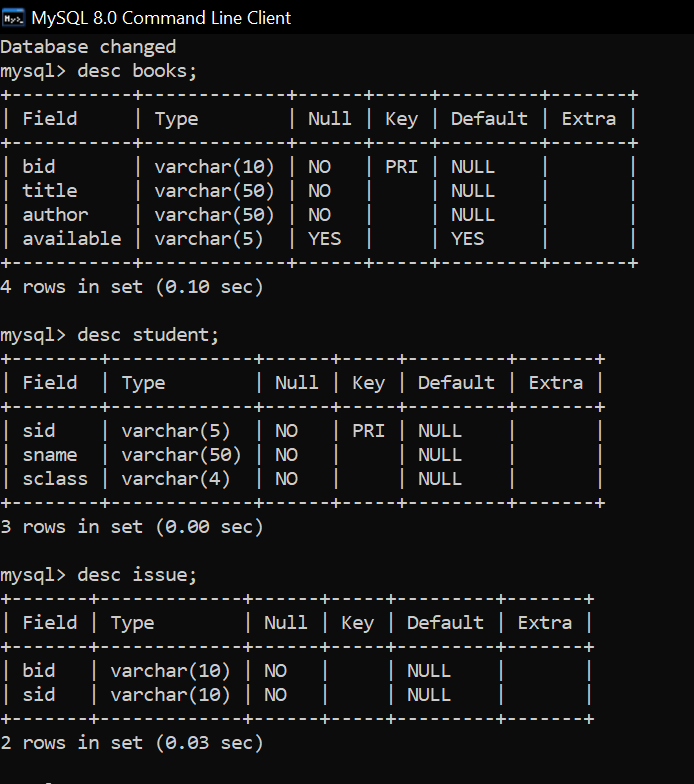
C:> pip install mysql-connector-python

**CREATING DATABASE IN MYSQL:**

* Run MySQL 8.0 Command Line Client from start menu



* Enter password for root
* Enter the following commands in mysql prompt to create database and tables:
  + create database library;
  + use library
  + create table books (bid varchar(10) PRIMARY KEY NOT NULL,title varchar(50) NOT NULL, author varchar(50) NOT NULL, available vartchar(5));
  + create student (sid varchar(5) PRIMARY KEY NOT NULL, sname varchar (50) NOT NULL, sclass varchar(4) NOT NULL);
  + create table issue (bid varchar(10) NOT NULL, sid varchar(10) NOT NULL);
  + structures of the tables are as follows:

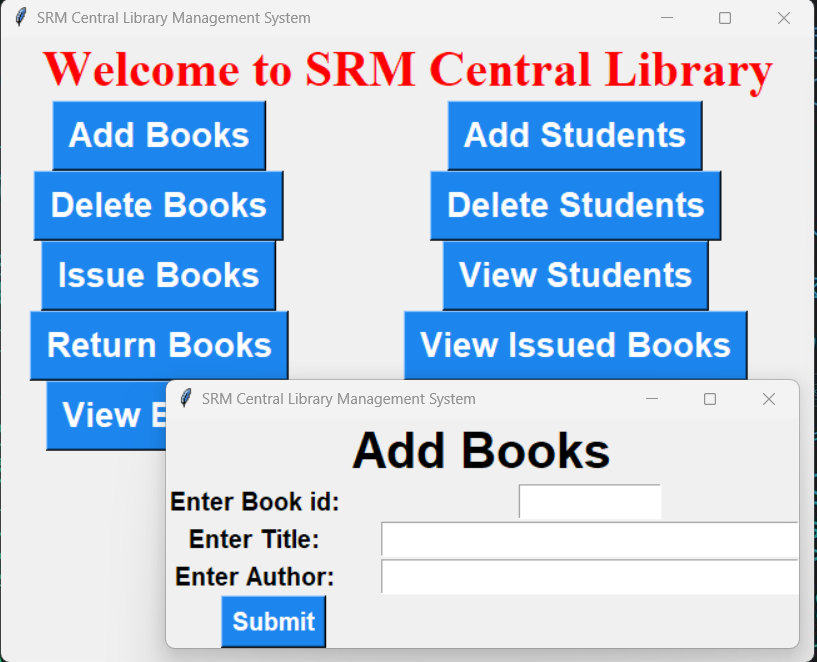


This is to be done only for first time.

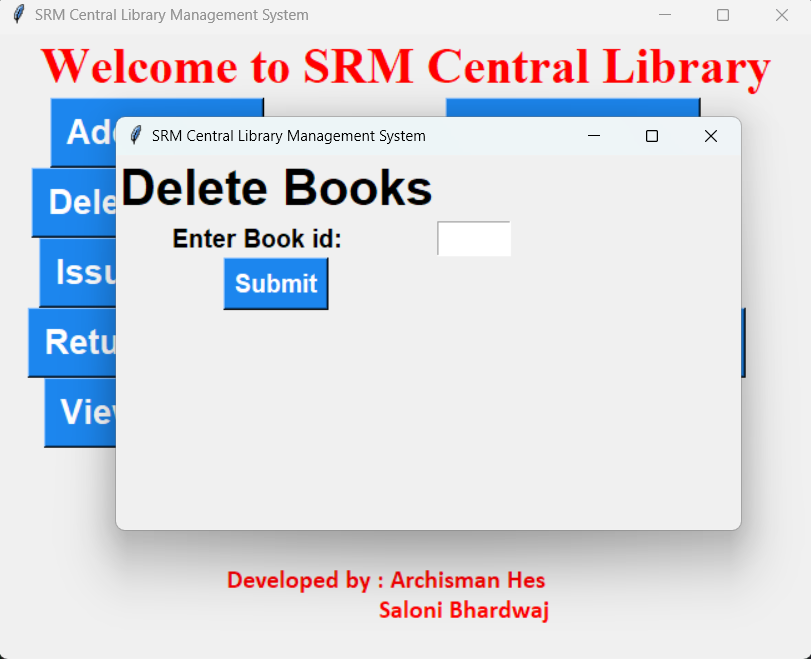
**MAIN MENU:**

****

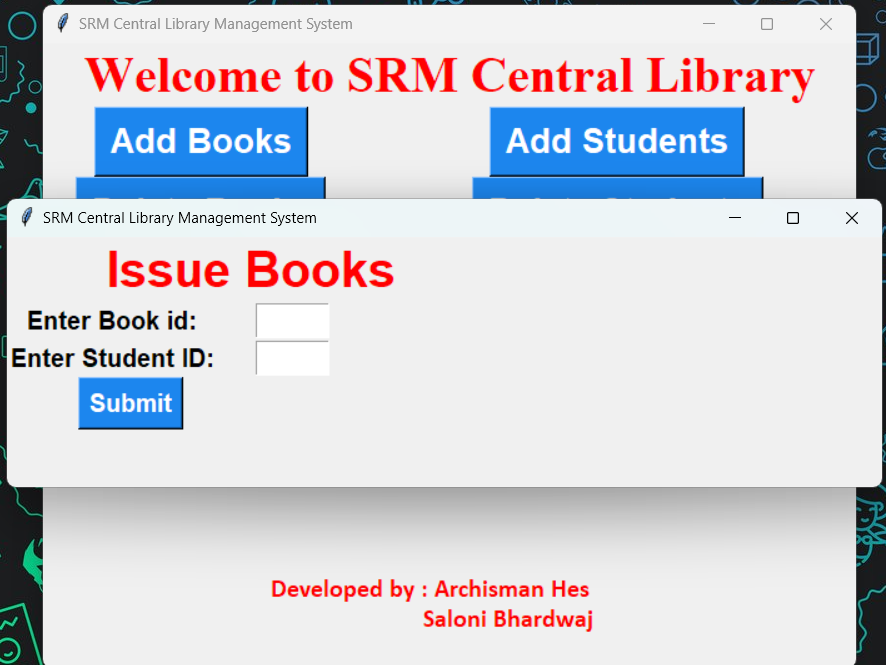
**ADD BOOKS:**

****

**DELETE BOOKS:**

****

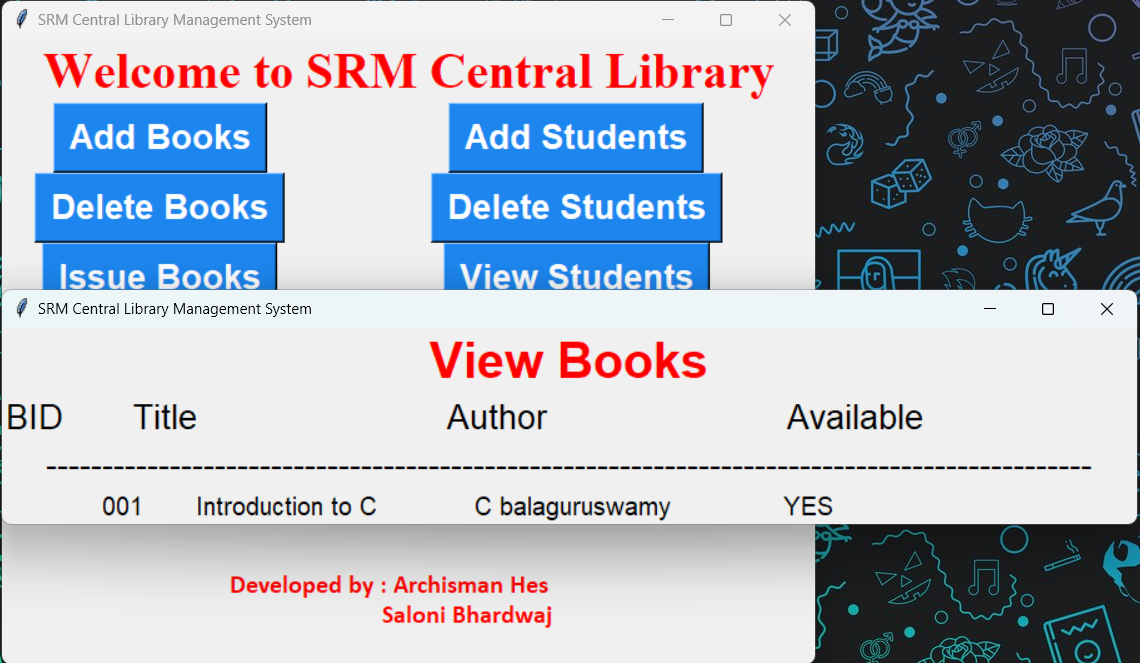
**ISSUE BOOKS:**

****

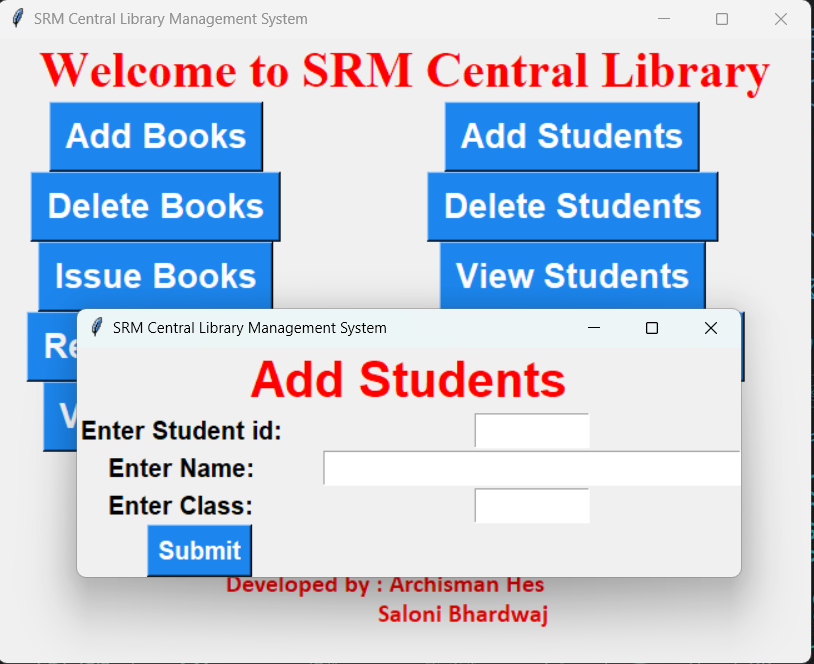
**RETURN BOOKS:**

****

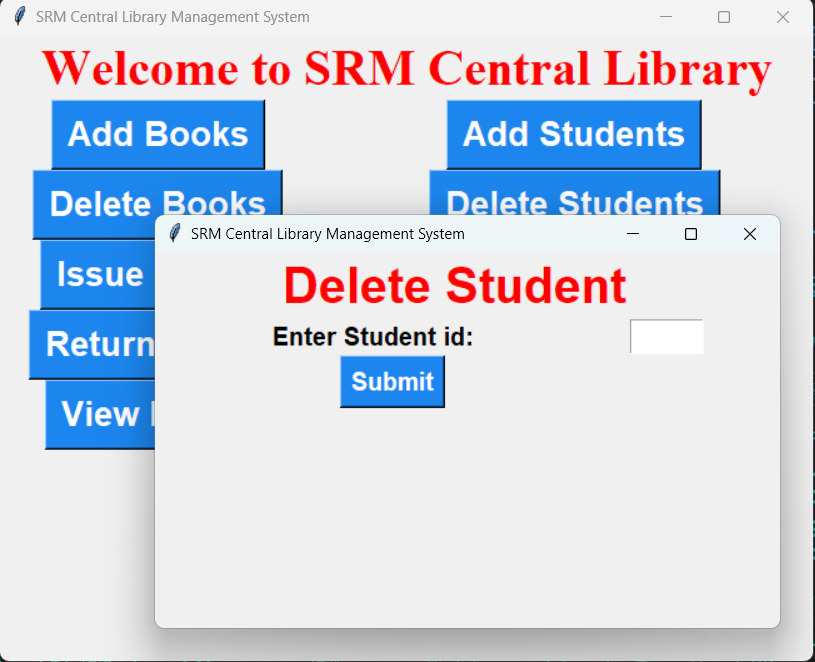
**VIEW BOOKS:**

****

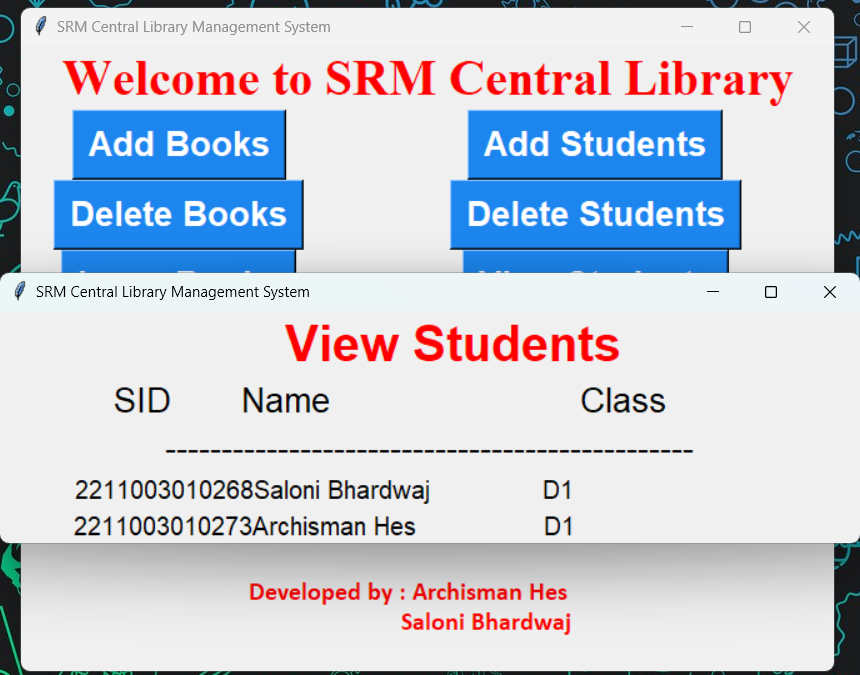
**ADD STUDENT:**

****

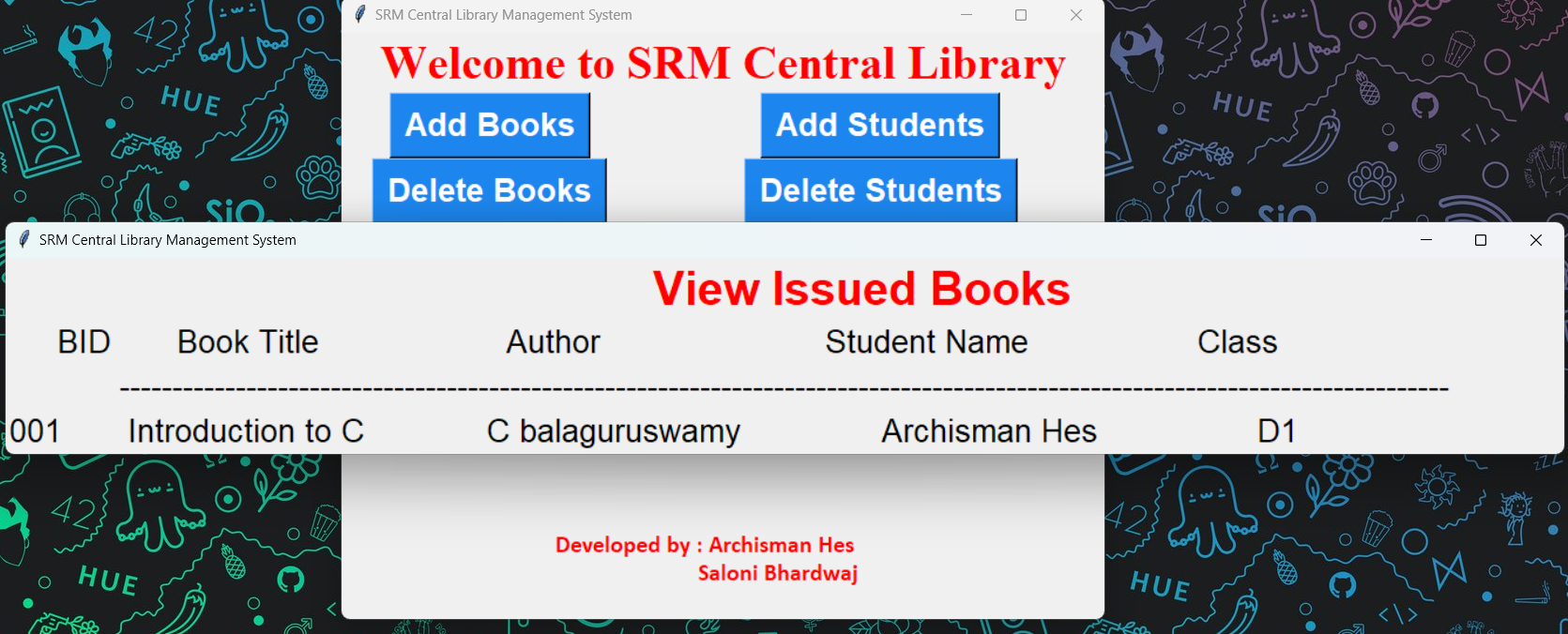
**DELETE STUDENT:**

****

**VIEW STUDENT:**

****

**VIEW ISSUED BOOKS:**

****

**CONCLUSION**

**ADVANTAGES**

By implementing this software, the user will be equipped with the following advantages :

* Use of REGISTERS will be stopped, like
  + Book register
  + Issue register
  + New book purchase register
  + Student register
* Searching time for any book, whether it is in library or it is already issued will be minimized.
* No unauthorized person can take book from the library. As every student has an unique ID, system will allow only to registered student to issue any book available.
* As the database is in MySQL 8.0 server, the application can be used by as many users as required with a little change in user authentication. This application is presently developed in single user environment.

**DRAWBACKS**

* Due to very limited time and with my limited knowledge in Python and MySQL, the project is completed in very basic manner.
* There is ample scope to modify the project as per real time requirement.
* The software is not multi user compatible.
* It is not safe and secure. Anyone can access the data.
* Since I am in beginner level of Python programming, GUIs are not so attractive. It can be done more attractive after acquiring more knowledge in Python – Tkinter.
* No option is present if more than one copy of a book is present.
* Finally I took the help of my father’s server where MySQL was already installed. I just made my database. Faced a lot of problem in installing the MySQL server in my laptop.

**BIBLIOGRAPHY**

1. Complete reference PYTHON by Maria C Brown
2. Python Crash Course by Eric Matthews
3. Learning Python, 5th edition by Mark Lutz
4. SQL, PL/SQL by Ivan Bayross
5. [www.w3schools.com](http://www.w3schools.com)
6. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
7. [www.tutorialspoint.com](http://www.tutorialspoint.com)
8. Different tutorial videos on Python, Tkinter and SQL in YouTube
9. <https://dev.mysql.com>
10. [www.mysql.com](http://www.mysql.com)