Course App Development Project

CS 315: Principles Of Database Systems
Submitted By : Shreyasi Prasad
Roll No. : 190824

KitabGhar

12th April 2022

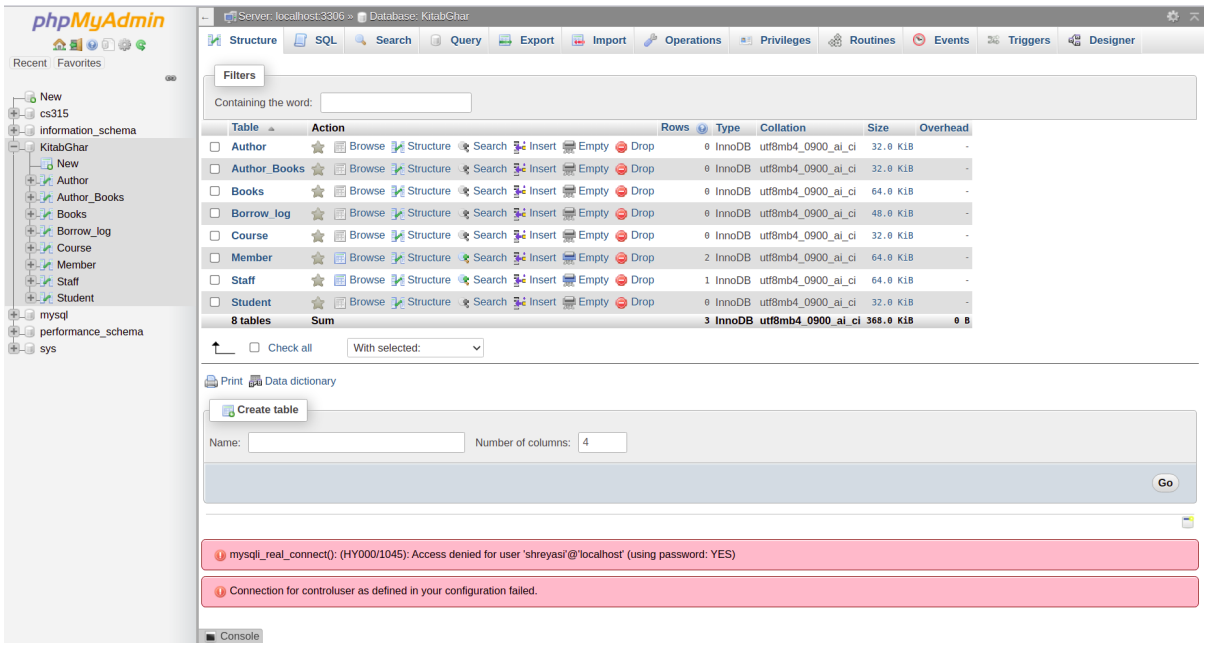
AIM AND OBJECTIVE

KitabGhar is a library management system developed for schools and colleges. The app will manage the database in a library and perform primary functions in a library like issue, return, check availability and position of books and show and clear dues of members like students and teachers that access the library.

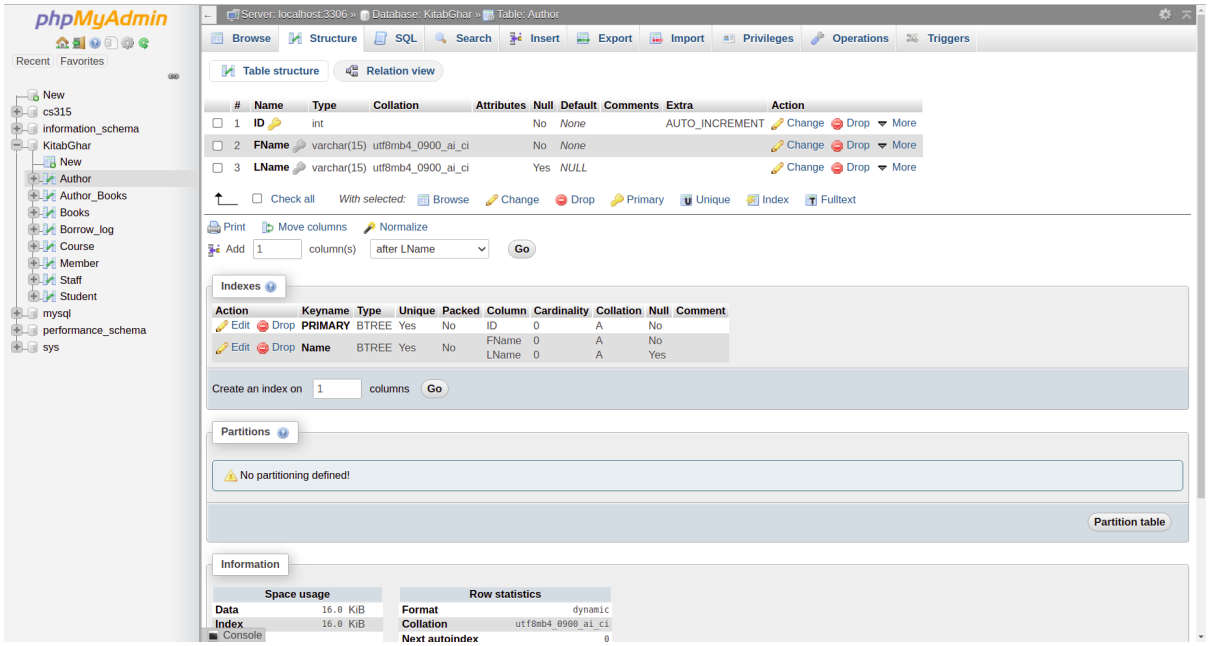
DESIGN

Table Design:

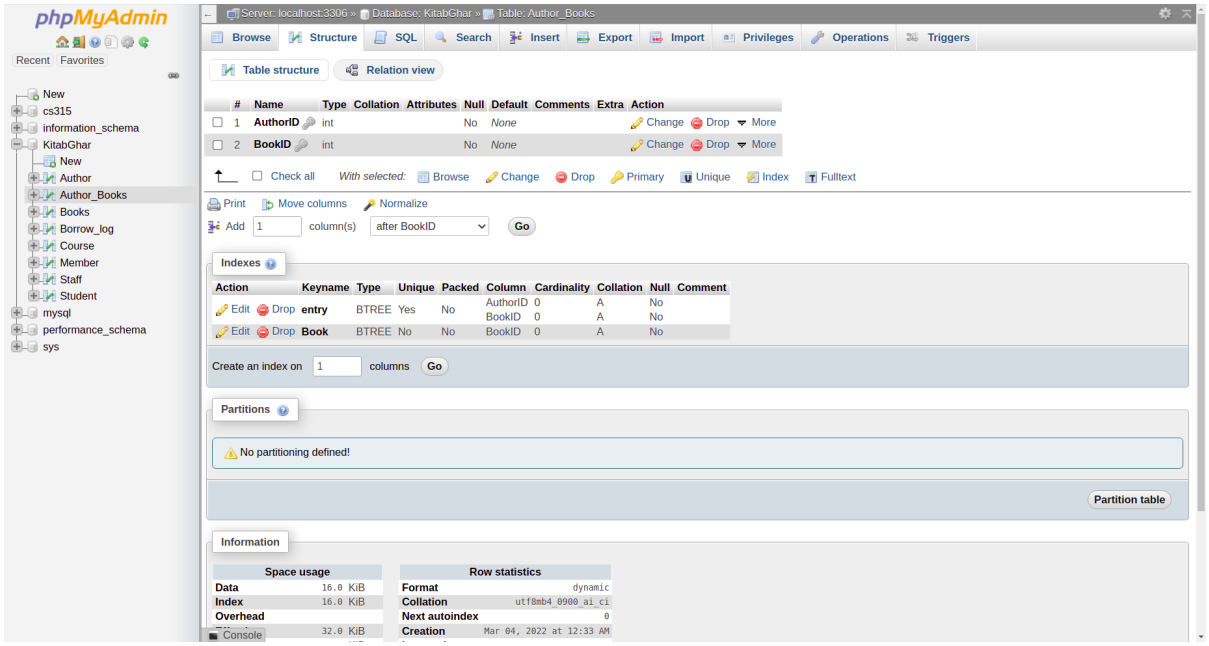
1. Database design:



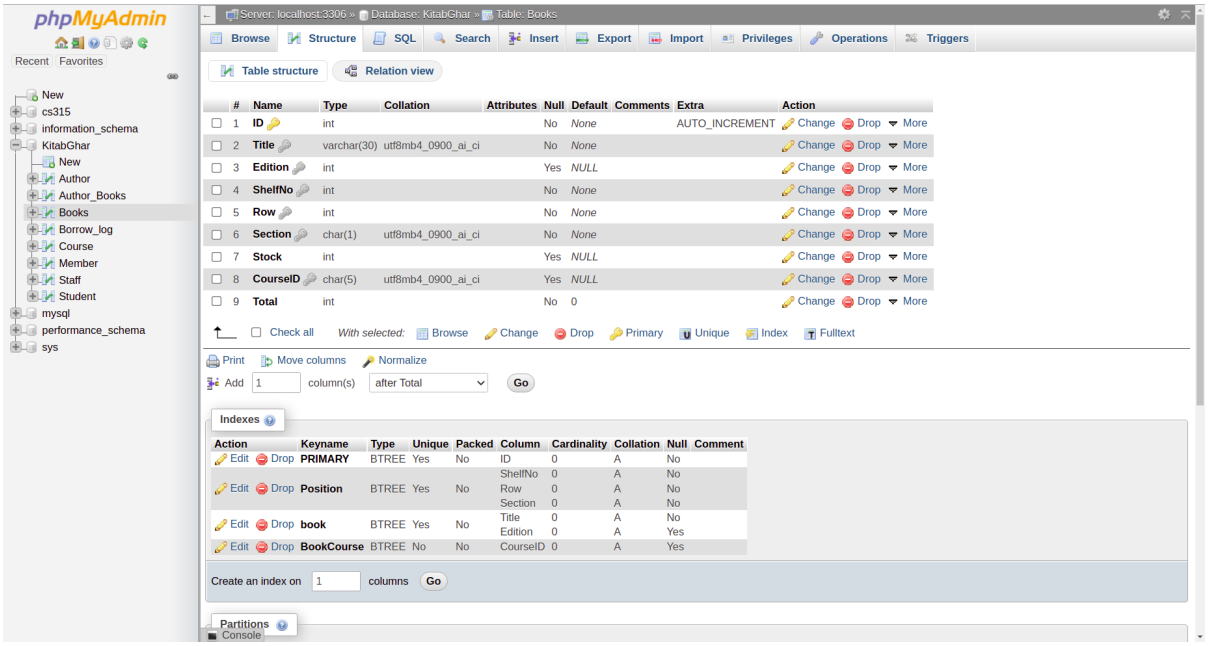
2. Author Table design:



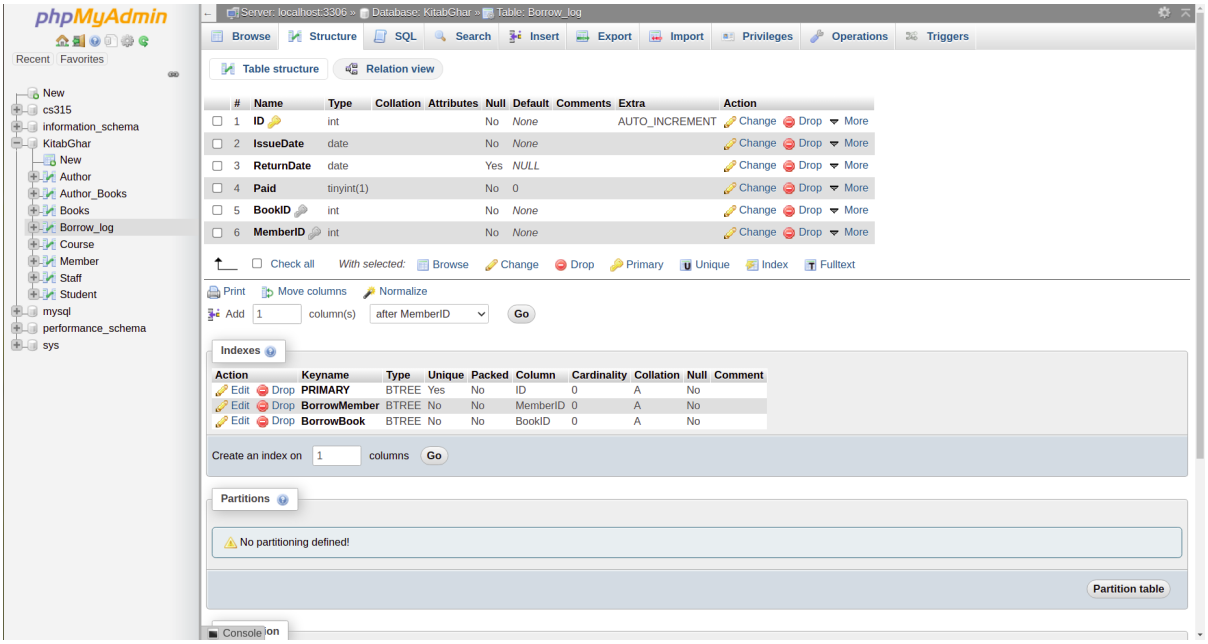
3. Author_Books Table design:



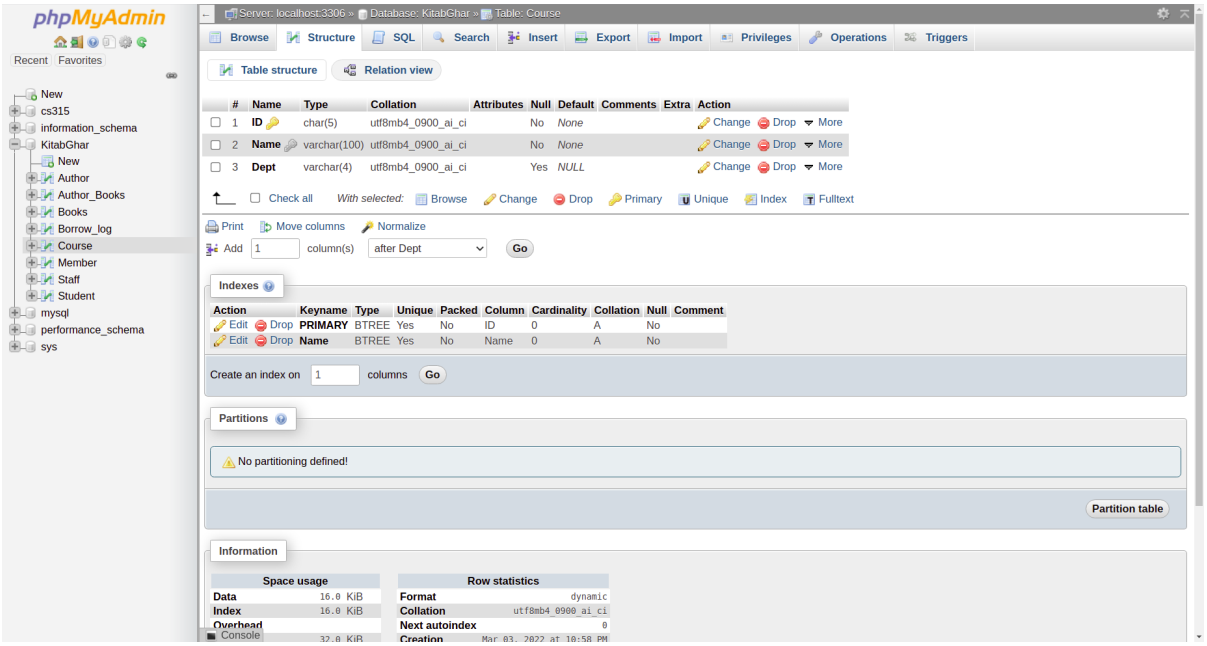
4. Books Table design:



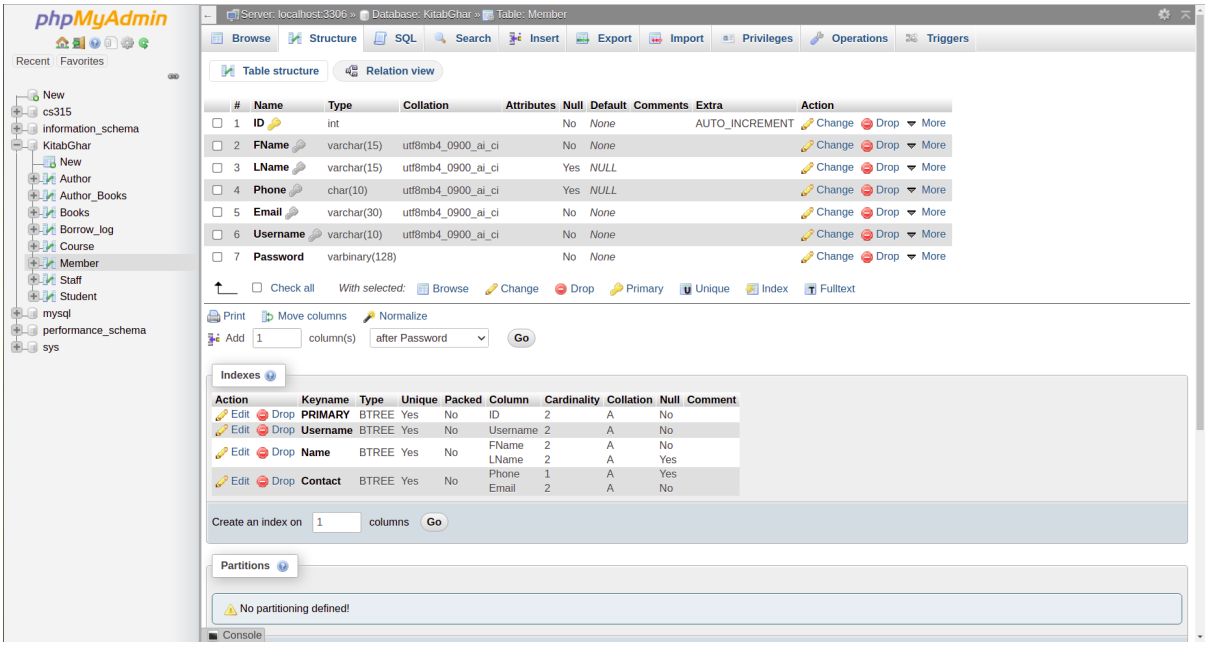
5. Borrow_log Table design:



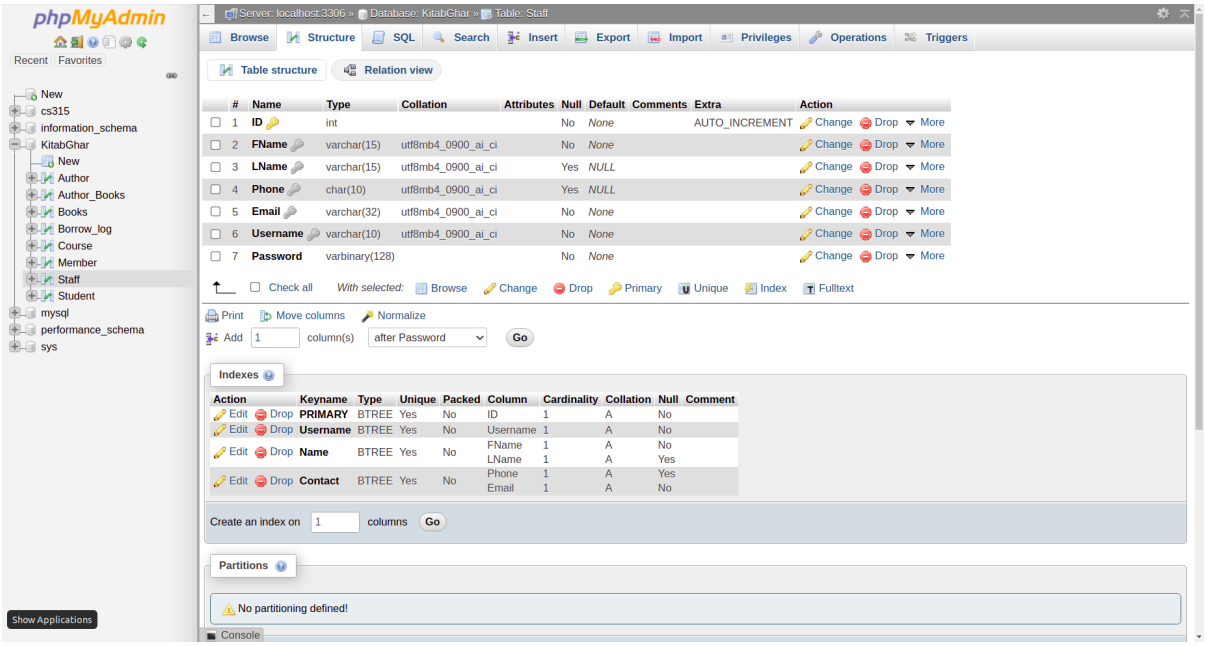
6. Course Table design:



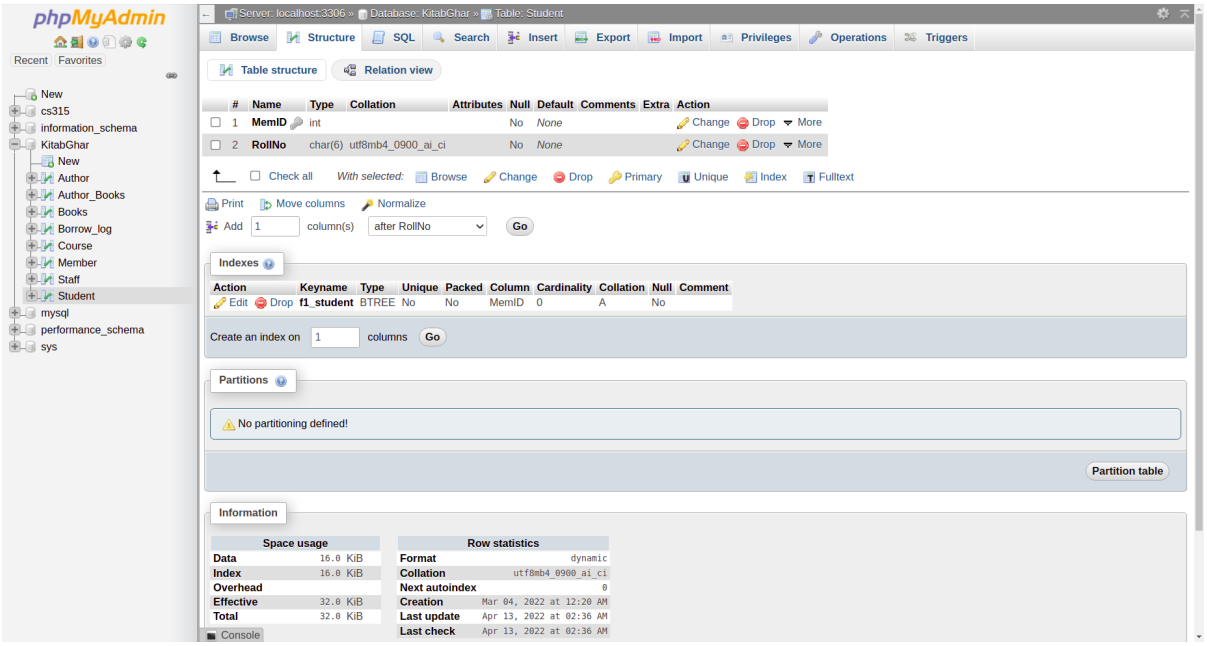
7. Member Table design:



8. Staff Table design:

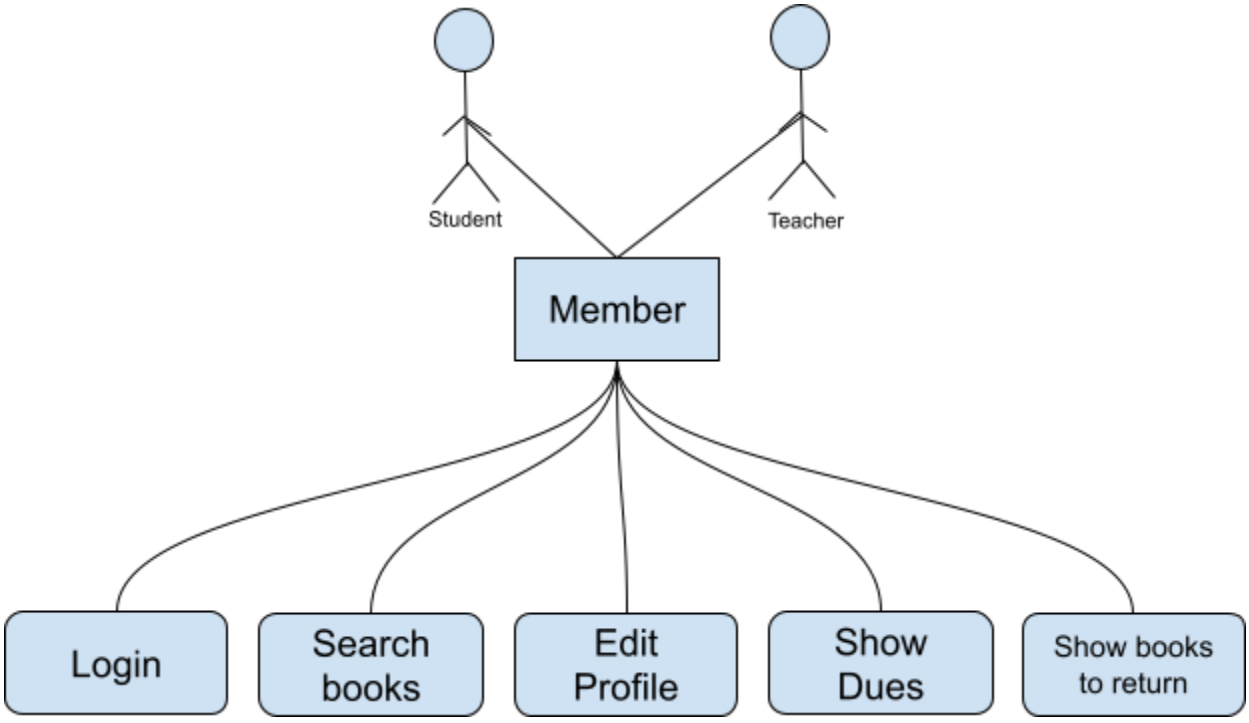


9. Student Table design:

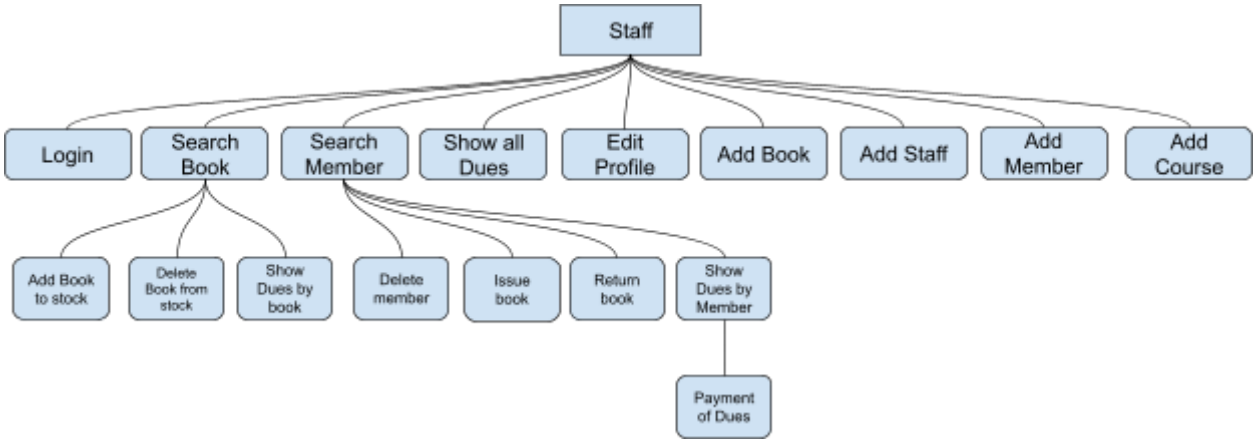


Use case diagram :

For Member:



For Staff:



IMPLEMENTATION

In making this app, I used Python3 for the backend and MySQL for Database management. The app is to be run on the terminal using the binary produced using PyInstaller. The basic requirement to run this app is any system that supports Python3 and MySQL.

FUNCTIONALITY

1. Login:

   Requires username, password and mode of login(member or staff), authenticates from the database and creates a person object of appropriate type.

2. Edit Profile:

   Requires to authenticate password before editing information. Can change Last name, password and phone number.

3. Search Book:

   Requires the parameter by which we need to search and the value of the parameter. We can search books by title, author name, course and department.

4. Search Member:

   Requires the username to be searched and displays the user with the given username. This function can only be used by staff.

5. Add new staff:

Requires first name, email id, username and a password to create a new staff account. Checks if the username is already present. This can only be done by an existing staff.

6. Add new member:

   Requires first name, email id, username and a password to create a new member account. Checks if the username is already present. This can only be done by an existing staff.

7. Add new course:

   Requires course name, course id and department to add a course. Checks if the course is already present. This can only be done by an existing staff.

8. Add new book:

   Requires book name, edition, position, course, number of books and name of authors to add new book. Checks if the book of given name and edition is already present, position is not occupied, course is already present, . This can only be done by an existing staff.

9. Delete member:

   Requires member id. Checks if the member is present. This can only be done by staff.

10. Add book to stock:

    Add books in the stock of books already present. Requires the number of books to be added. This can only be done by staff.

11. Delete book from stock:

    Delete books from the stock of books present. Requires the number of books to be added. This can only be done by staff.

12. Show Dues:

    Members can see their personal dues. Staff can see the list of dues of a member or dues to be cleared by on a book or both. Requirement is the id of parameters used to see the dues(bookid for books and memid for members).

13. Fine payment:

    Fine payment should be done by a member to the staff then staff will update on the system using this function. This function can only be used by the staff and need the id of the borrow-log.

14. Issue Book:

    This function can be used only by staff. Inputs required are member ID and book ID to issue a book to the member.

15. Return Book:

    This function can only be used by the staff when a member returns a book to the library. This requires the borrow-log ID.

16. Show books to return:

    This function is used by the member to see the books they have to return and their due dates.

FUTURE SCOPE

The future scope of this app is to add many more functionalities like

1. Members can request books and that can get approved by the staff.
2. Members can pay the fine which can be validated by the staff.
3. Front end can be more robust and user friendly.