**Name:** IRADUKUNDA Benjamin

**Reg No:** 221018380

**Projet: LIBRARY MANAGEMENT SYSTEM**

**PART 1: SYSTEM ENGENIRING**

**2. Introduction:**

A library management system is a software that is designed to manage all the functions of a library. It helps librarians to maintain the database of new books and the books that are borrowed by members along with their due dates. This system completely automates all your Library’s activities.

**3. Problem description:**

Among the main challenges bedeviling use of Library management systems in the libraries are problems with lack of software systems that help them in LMS and inadequacy of staff with ICT skills.

The biggest challenge public libraries are facing;

* **Funding**

Some libraries have the problem with funding the money for building the library management system because they think that is very expensive and their owner of the libraries they are worried about is going to take them a lot of money and workers to use that system. But, they are wrong.

* **ICT Skills, Tools And Technology**

Nowadays, in a highly technological society, There is higher technological in different sectors, also libraries have the problem of lack  of skilled workers in ICT to make efficiency and effective in there workplace

The lack of internet this is the best challenge that limit different companies to use the software in there work like in management some they use Books in recording some transactions and they workflow per day.

The other issue is the problems of buying the tools to use that softwares even if they have done for building that softwares

* **Lack of good planning**

Some libraries fail in planning before implementing their libraries.

* **Limited knowledge**

Owners have little knowledge of financial management in their libraries, whether in public or private libraries, and feel that using paper is enough. not the only thing that leads to a lack of bookkeeping information and a lack of information on libraries or are they following their libraries

**4.Proposed solution:**

Build software **“LIBRARY MANAGEMENT SYSTEM”** which will help the libraries to maintain and keep their Books/CD etc, and also help Members to request book and other activities of Library

**5.General Objective of Library Management System**

**The Library Management system** helps in maintaining data of books issued to learners and books available in the library. This helps librarians spot any particular book at any given time in the library.

**Features of Library Management System Are As Follows :**

* Provide searching facilities based on various factors. Such as address, issues, Books
* The transactions are executed in off-line mode, hence online data for address, member capture and modifications is not possible
* Shows the process of borrowing Book
* Shows the information and description of the Books
* To know the availability of Books
* Get easily information about Library

**6.Specific Objectives Of Library Management System**

**Library management systems** enable system administrators to keep an eye on the library department’s functioning and also enable librarians and users to maximize time and efficiency. By using this kind of system, the management would be able to understand the work outline and fineness of different librarians as well. They also get to know how well-maintained the record of issued books and fine collection is, apart from this management is also able to track the income from fines due to late submissions of books.

The librarian and the management will also get reports with various productive data. There can be reports on borrowing and lending, which can be engendered in different formats. Such reports can help with ease in the decision making procedure. It also keeps the record of all suppliers and bookbinders.

**Specific Features of Library Management System Are As Follows :**

* It tracks all the information of members
* Provide filter reports on issues, Books and transactions
* Provide Monthly reports so that they can be easy in management and decision making
* Editing, adding and updating of records is improved which result in proper resource management of address data.
* Integrating of records of members
* Increase efficiency of managing
* You can easily export PDF for the address, librarian, Books

**7. FUNDAMENTAL REQUIREMENTS**

The key requirements that need to be offered by the library management system can be classified into **functional and non-functional requirements.**

**A Library Management System** is software that provides the ability to find books, manage books, track borrowed books, managing fines and bills all in one place. It helps the librarian manage the books and books borrowed by members and automates most of the library activities. It increases efficiency and reduces the cost needed for maintaining a library and saves time and effort for both the user and the librarians.

**7.1.Functional Requirements**

1. Allow the librarian to add and remove new members.
2. Allow the user to search for the books based on title, publication date, author, etc and find their location in the library
3. Users can request, reserve, or renew a book.
4. Librarians can add and manage the books.
5. The system should notify the users and librarian about the overdue books
6. The system calculates the fine for overdue books on their return.

### 8. Non - Functional Requirements

**Usability**

Usability is the main non-functional requirement for a library management system. The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

**Accuracy**

Accuracy is another important non-functional requirement for the library management system. The data stored about the books and the fines calculated should be correct, consistent, and reliable.

**Availability**

The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

**Maintainability**

The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

**9. Feasibility Study**

Considering the requirements, a full scale feasibility study was undertaken for testing the possibility of computerization of the Library management system. The feasibility study was carried out under the following three areas :

* Technical Feasibility,
* Social Feasibility
* Economic Feasibility

1. **Technical feasibility**

The system may not be complicated to install either in taking much space/ storage to the computer or and to use, the system can be running in the many computers that may not lead to too much cost.

This shows that technically it is possible to get the required hardware, software and a person having complete understanding of the requirements to develop the software package.

Before starting developing software we must first make sure that the software we are going to build will work in the library’s computer

**2. Social Feasibility**

Generally computerization brings fear of unemployment among the existing staff. The automation of Libraries will not create unemployment problems but it will create new job positions, like system manager, system analyst, programmers and programme maintenance staff, data entry and problems they have in the existing system. At the time of implementation, each employee will be consulted and accordingly, in-house education and training programmers will be arranged for the existing staff member of the library for 8-1 0 days duration. However, staff members who are about to retire within 1-2 years, may be reluctant to undergo training in computer use of computers due to ignorance, fear and hostility, such persons can be utilized for manual professional activities, as an alternative to avoid redundancy payment.

**3.  Economic Feasibility**

The economic feasibility study was carried out under following areas:

* Cost of operations of existing system
* Cost of operations of the proposed system
* Cost of development of the proposed system
* Benefits of the proposed system.

Cost of operations of the existing system : Financial records, like annual budget, payrolls, etc. were scrutinized and the librarian and each section incharge in BAMUL was interviewed to determine the manpower cost, material cost, operating cost, equipment cost, overhead cost, etc. for calculation of unit cost of each operations in Library.

The total operational expenditure of the library on manpower, material overhead, etc

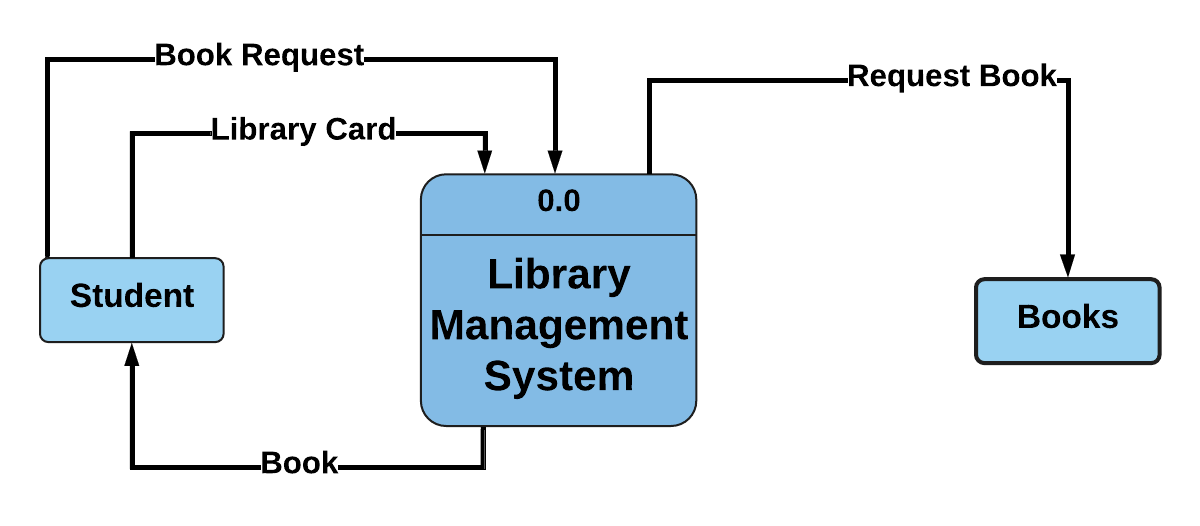
Cost of operation of the proposed system: The computerized Library system requires manpower, materials, equipment, overhead cost, etc. Also, it requires data preparation and conversion cost, maintenance cost, site preparation cost, etc

**CONCLUSION**

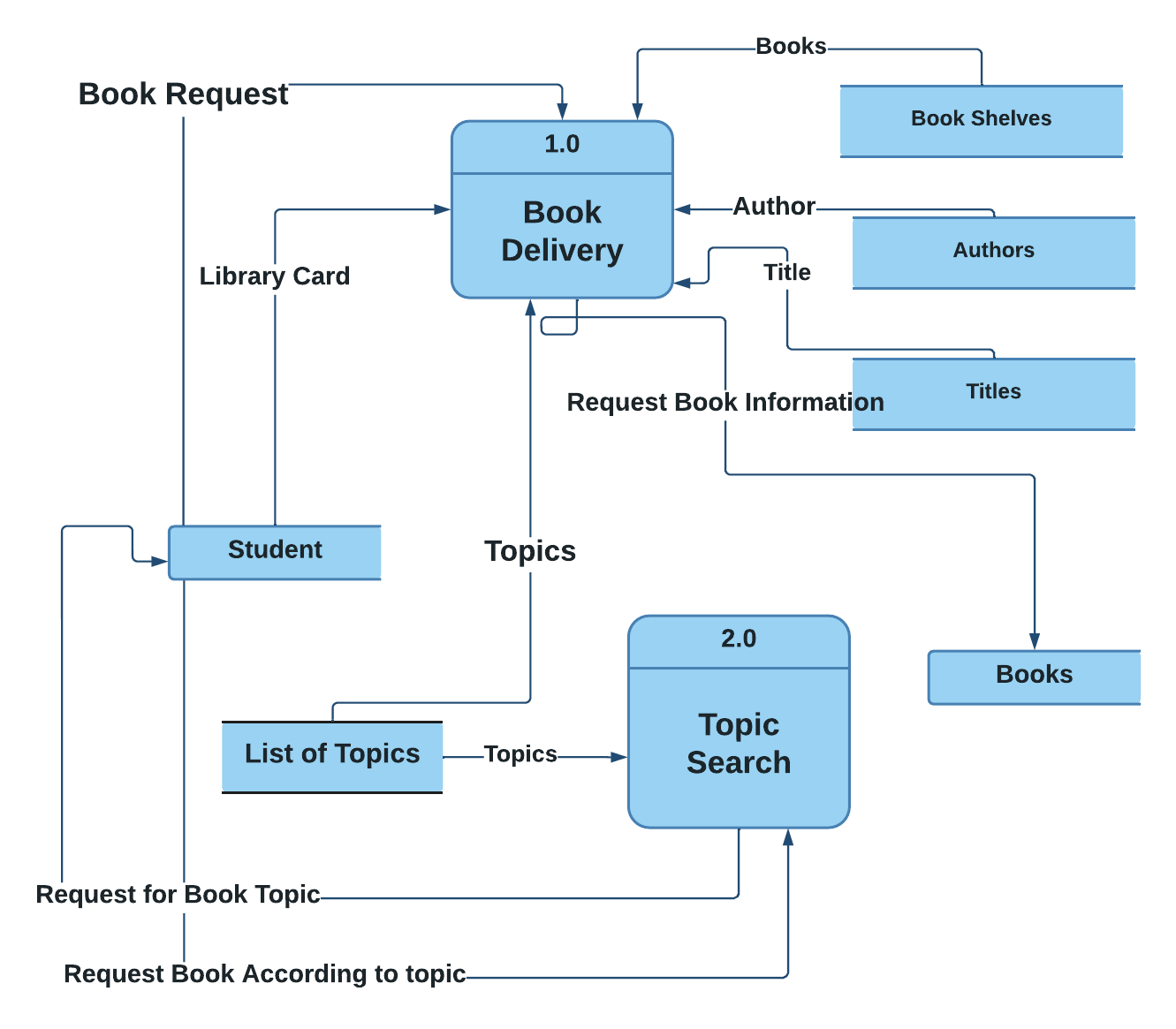
 Library automation is the need of the day. However, one has to think of the investment, its technical and social aspects. Technically it is possible to get the required hardware, software and person having complete understanding of the requirements to develop software packages. Automation will create new job opportunities hth higher scales and relieve the existing professional staff from their routine clerical activities to enable them to perform intellectual professional duties. The reorganization of professional staff will lead to job satisfaction. The social cost of the proposed system will be only the training cost of library personnel for 8-10 days. It shows that library automation in LMS is feasible socially. The operational cost of the proposed system is less than that of the existing system which includes manpower, material equipment and overhead cost. At the same time, the system will be able to generate financial resources to make the system cost beneficial along with many intangible benefits, hence economically library automation is feasible.

**10.DATA FLOW DIAGRAM OF LIBRARY MANAGEMENT SYSTEM**

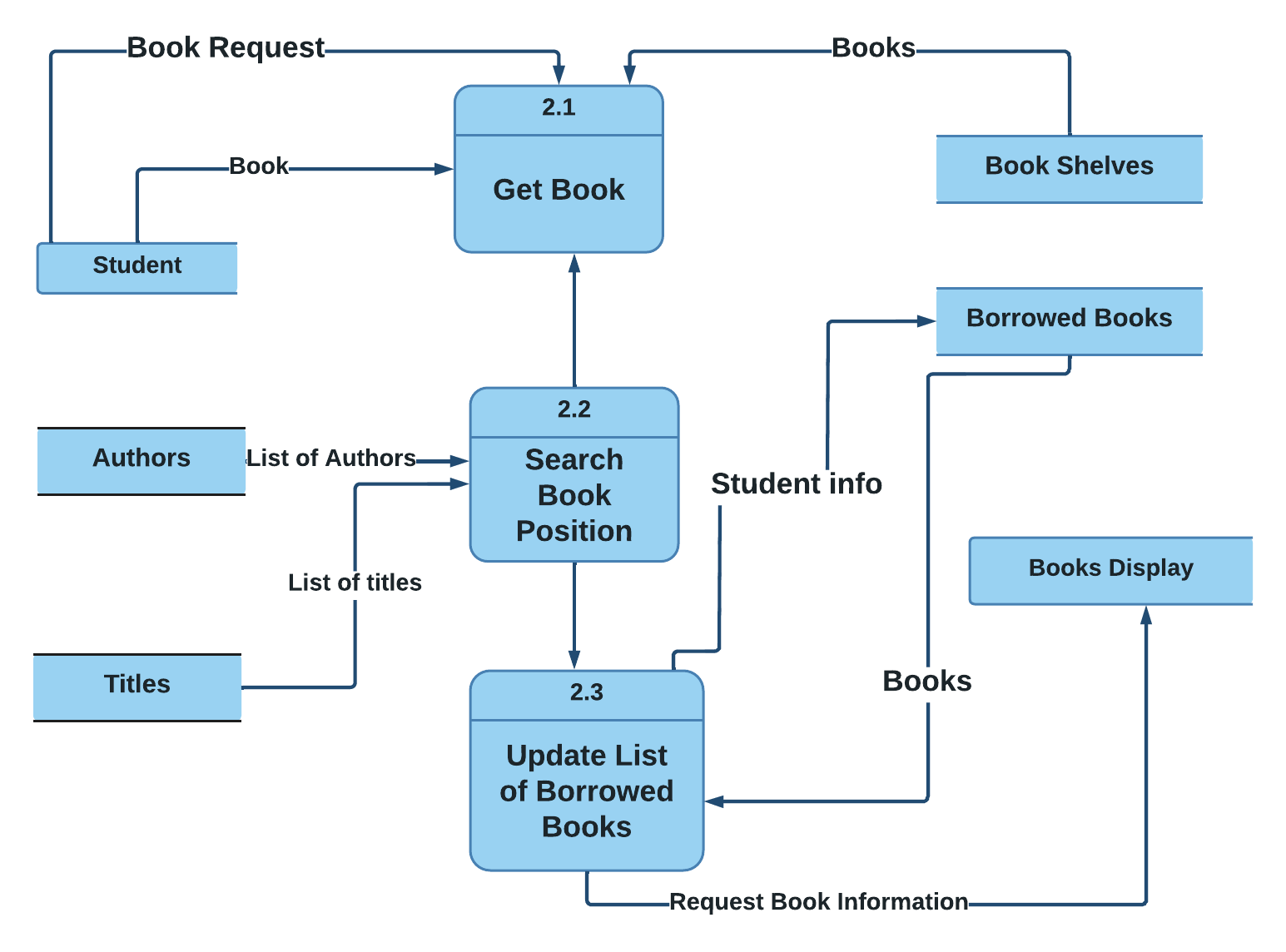
**10.1. Data Flow Diagram Level 0**

****

**10.2 Data Flow Diagram Level 1**

****

**10.3 Data Flow Diagram Level 2**

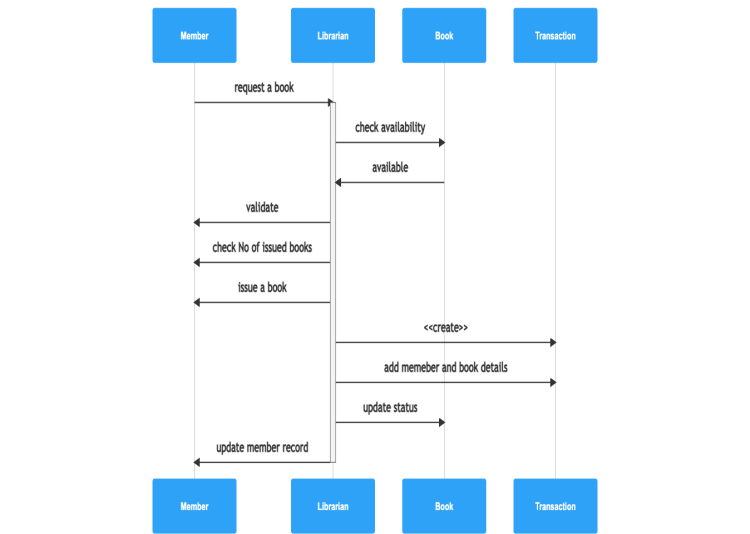
****

**11. User Case Diagram**

****

**​​**

**12. Sequence Diagram**

****

**PART 2: DATABASE**

**SECTION 1:**

1. Describe all the entities and their corresponding attributes that are in your database.

* BOOK TABLE

|  |
| --- |
| Book\_id PRIMARY KEY that will hold the ID of whole row records |
| bookTitle Book title |
|  |
| BookAuth Auth of the book |
| BookPublisher publisher of the Book |
| BookCopies |
| Book\_source |
| Book\_cost the price of the book |
| Book\_remarks |

* STUDENTS/ MEMBERS TABLE

|  |
| --- |
| student\_ID PRIMARY KEY |
| fname |
| lname |
| gender |
| age |
| contact\_add |
| stundent\_email |
| student\_pass |

* STAFF

|  |
| --- |
| staff\_ID PRIMARY KEY |
| Fname |
| Lname |
| Gender |
| Age |
| Contact\_add |
| Staff\_email |
| Satff\_pass |

* REPORTS

|  |
| --- |
| reportID PRIMARY KEY |
| reportTransaction\_id FOREIGN KEY |
| reportbook\_ID FOREIGN KEY |
| reportbookTitle |
| reportCopieReturned |
| reportReturnedDate |
| reportRemarks |
| reportNumberOfDays |
| reportPenalty |
| reportReceived |

|  |
| --- |
| Borrowers\_id PRIMARY KEY |
| Book\_id # FK Foreign Key |
| stud\_ID # Foreign Key |
| Staff\_id # Foreign Key |
| staffName |
| studentNOcopies |
| ReleaseDate |
| DueDate |

* TRANSACTION

| transaction\_ID PRIMARY KEY |
| --- |
| transaction\_name |
| Borrowing\_id FOREIGN KEY |
| student\_ID FOREIGN KEY |
| transactionFee |
| transaction\_date |

2. Create a LDM of your entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| Book\_id PK | Book Id number | int | 11 |
| bookTitle | Book Title | varchar | 30 |
| bookEdition | Book Edition | varchar | 30 |
| BookAuth | Auth Of the Book | varchar | 30 |
| BookPublisher | Publisher of the book | varchar | 30 |
| BookCopies | Number of copies | int | 11 |
| Book\_source | Source of the book | varchar | 30 |
| Book\_cost | Cost of the book | int | 11 |
| Book\_remarks | Status of the book | varchar | 30 |

Table Name : **Borrowing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| borrowers\_id(PK | Borrowers Id number | int | 11 |
| Book\_id FK | Book ID number | int | 11 |
| stud\_ID | Student ID number | int | 11 |
| staff\_id | Staff ID number | int | 11 |
| staffName | Name of the staff | varchar | 30 |
| studentNOcopies | Number of books to be borrowed | int | 11 |
| ReleaseDate | Date of the publication that was released | date | 30 |
| DueDate | Due date of the book returned | date | 30 |

Table Name: **tbclearedrecords**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| clearID(PK | Specific ID for cleared record | int | 11 |
| Borowwer\_id | Book ID number | int | 11 |
| book\_ID | Book id | int | 11 |
| bookTitle | Book title | varchar | 30 |
| studID | Student id | int | 11 |
| studName | Staff name | varchar | 30 |
| studentCopies | Student number of copies | int | 11 |
| releaseDate | Due date of the book returned | Date | 11 |
| DueDate | Due date of the book returned | Date | 11 |

Table Name: **tblreport**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| reportID(PK | Report IOD | int | 11 |
| reportTransaction\_id | Transaction ID | int | 11 |
| reportbook\_ID | Book id | int | 11 |
| reportbookTitle | Book title | varchar | 30 |
| reportCopieReturned | Book number of copies | varchar | 30 |
| reportReturnedDate | Date to be returned | Date | 11 |
| reportRemarks | Remarks of the book | varchar | 30 |
| reportNumberOfDays | Number of days to be borrowed | int | 11 |
| reportPenalty | A penalty of the student | varchar | 30 |
| reportReceived | Receive | varchar | 30 |

Table Name: StundentTable

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| student\_ID(PK | Student ID | int | 11 |
| fname | First name | varchar | 30 |
| lname | Last name | varchar | 30 |
| gender | gender | varchar | 30 |
| age | age | int | 11 |
| contact\_add | Phone number | int | 11 |
| stundent\_email | Student email | varchar | 30 |
| student\_pass | Student password | varchar | 30 |

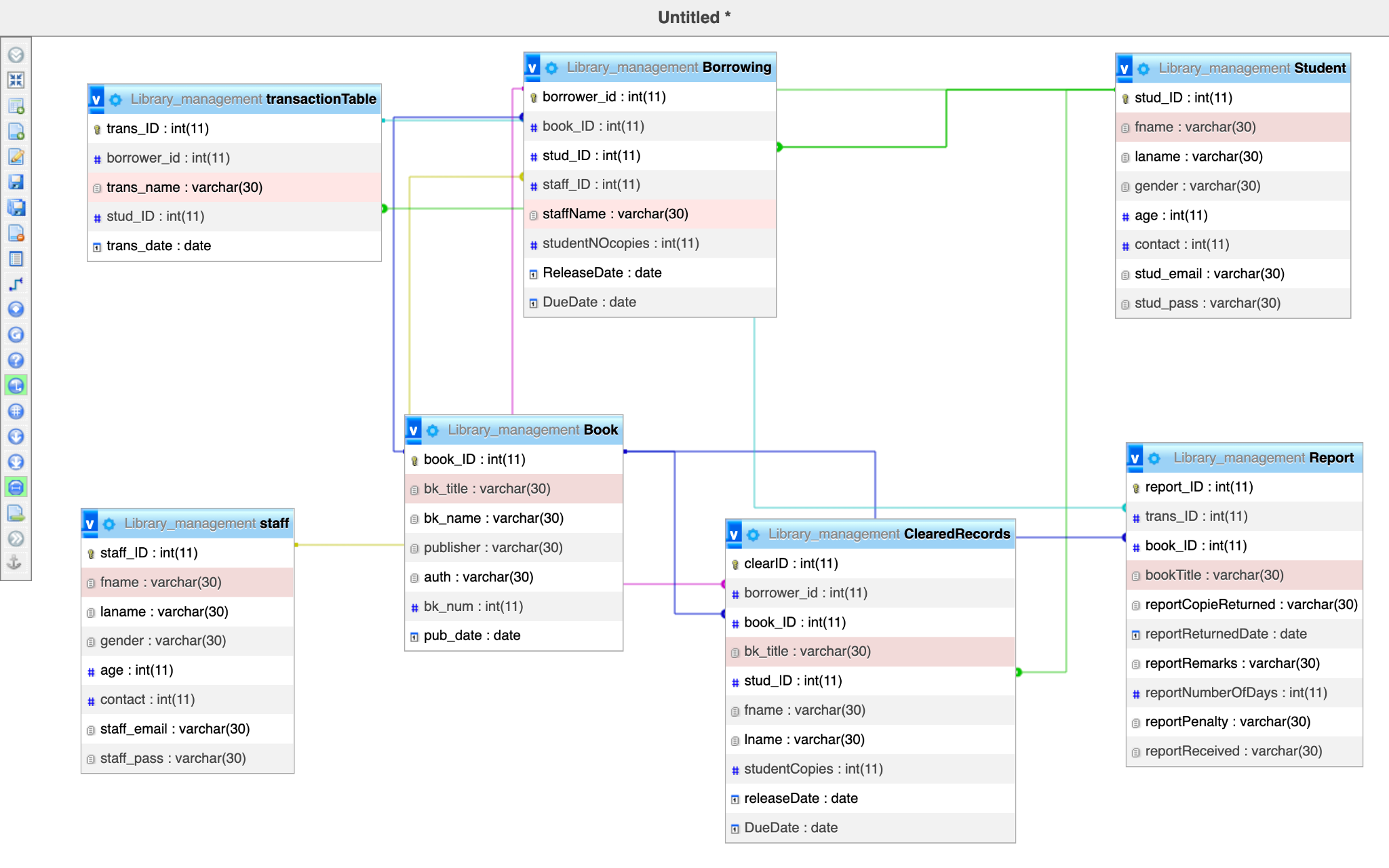
Table Name: staffTable

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| staff\_ID(PK | Staff ID | int | 11 |
| fname | First name | varchar | 30 |
| lname | Last name | varchar | 30 |
| gender | gender | varchar | 30 |
| age | age | int | 11 |
| contact\_add | Phone number | int | 11 |
| staff\_email | Staff email | varchar | 30 |
| satff\_pass | Staff password | varchar | 30 |

Table Name: transactionTable

| **Field** | **Description** | **Type** | **Length** |
| --- | --- | --- | --- |
| transaction\_ID(PK | Transaction ID | int | 11 |
| transaction\_name | Transaction name | varchar | 30 |
| Borrowing\_id (FK | Borrowing id | int | 11 |
| student\_ID(FK | Student id | int | 11 |
| transactionFee | fees | int | 11 |
| transaction\_date | Date | int | 11 |

3. Create an ERD



**SECTION II SQL:**

1. Create the database of your system

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) [DATABASE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) Library\_management;

2. Write queries to create all the tables and relationship of your system

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Student (stud\_ID int (11) PRIMARY KEY, fname varchar (30), laname varchar (30), gender varchar (30), age int (11), contact int (11), stud\_email varchar (30), stud\_pass varchar (30));

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Book (book\_ID int (11) PRIMARY KEY, bk\_title varchar (30), bk\_name varchar (30), publisher varchar (30), auth varchar (30), bk\_num int (11), pub\_date date);

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) staff (staff\_ID int (11) PRIMARY KEY, fname varchar (30), laname varchar (30), gender varchar (30), age int (11), contact int (11), staff\_email varchar (30), staff\_pass varchar (30));

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) transactionTable (trans\_ID int (11), trans\_name varchar (30), stud\_ID int (11), PRIMARY KEY (trans\_ID), FOREIGN KEY (stud\_ID) REFERENCES Student (stud\_ID) );

//create Borrowing Table

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Borrowing (borrower\_id int (11), book\_ID int (11), stud\_ID int (11), staff\_ID int (11), staffName varchar (30), studentNOcopies int (11), ReleaseDate date, DueDate date, PRIMARY KEY (borrower\_id), FOREIGN KEY (stud\_ID) REFERENCES Student (stud\_ID) );

//add CONSTRAINTS

[ALTER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) Borrowing ADD CONSTRAINT FK\_BorrowBook FOREIGN KEY (staff\_ID) REFERENCES staff(staff\_ID);

[ALTER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) Borrowing ADD CONSTRAINT FK\_BorrowANDBook FOREIGN KEY (book\_ID) REFERENCES Book(book\_ID);

[ALTER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) Borrowing ADD CONSTRAINT FK\_BorrowANDStudent FOREIGN KEY (stud\_ID) REFERENCES Student(stud\_ID);

// CREATE REPORT TABLE

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Report (report\_ID int (11), trans\_ID int (11), book\_ID int (11), bookTitle varchar (30), reportCopieReturned varchar (30), reportReturnedDate date, reportRemarks varchar (30), reportNumberOfDays int (11), reportPenalty varchar (30), reportReceived varchar (30), PRIMARY KEY (report\_ID), FOREIGN KEY (trans\_ID) REFERENCES transactionTable (trans\_ID));

//ADD CONTRAINTS

[ALTER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) Report ADD CONSTRAINT FK\_BookIDinReport FOREIGN KEY (book\_ID) REFERENCES Book(book\_ID);

// CREATE CLEARED RECORD TABLE

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) ClearedRecords (clearID int (11), borrower\_id int (11), book\_ID int (11), bk\_title varchar (30), stud\_ID int (11), fname varchar (30), lname varchar (30), studentCopies int (11), releaseDate date, DueDate date , PRIMARY KEY (clearID), FOREIGN KEY (borrower\_id) REFERENCES Borrowing (Borrower\_id) );

// ADD CONTRAINST

[ALTER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) ClearedRecords ADD CONSTRAINT FK\_BookIDinCleardRecord FOREIGN KEY (book\_ID) REFERENCES Book(book\_ID);

[TABLE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) ClearedRecords ADD CONSTRAINT FK\_StudentIDinCleardRecord FOREIGN KEY (stud\_ID) REFERENCES Student(stud\_ID);

3. write queries to insert data into your tables.

//insert into Book Table

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Book [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (1, 'Rich dad Poor dad', 'Rich dad Poor dad', 'Ben Iraa', 'Nicolas', 1234, 2022-07-11);

//insert into student

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Student [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (12, 'Benjamin', 'IRADUKUNDA', 'male', 21, '781975442', 'beniraa@gmail.com', 87654321);

//insert into staff

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO staff [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (222, 'Ingabire', 'Rosette', 'female', 21, 83883772, 'rosita@gmail.com', 9876272)

//insert into Borrowing

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Borrowing [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (22, 2, 12, '222', 'manager', 1234, 2022-07-20, 2022-07-23);

//insert into Reports

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Report [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (32, NULL, 2, 'crazy me', 123, 2022-07-23, 'returned', 12, 'none', 'yes');

//insert into TransactionTable

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO transactionTable [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (88, 'penality fees', 12, 2022-07-30);

//insert into CleardRecords

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO ClearedRecords [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (44, 22, 2, 'Rich dad Poor dad', 12, 'Benjamin', 'IRADUKUNDA', 1234, 2022-07-20, 2022-07-23);

4. Write queries to display all the information in your tables.

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Book;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Borrowing;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM ClearedRecords;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Report;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM staff;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Student;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM transactionTable;

5. Write a query to update information in any of the two tables of your system.

//Update Information in table Report

[UPDATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) Report [SET](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) `trans\_ID` = '88' WHERE Report.report\_ID = 32;

//Update information in table Student

[UPDATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) Student [SET](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) stud\_email = 'beniraa5@gmail.com' WHERE Student.stud\_ID = 12;

**Section III**

1. Create a view to insert data into your tables.

//BookView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) Book\_view AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) book\_ID, bk\_title, bk\_name, publisher, auth, bk\_num, pub\_date FROM Book;

//insert into BookView

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Book\_view [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (3, 'mathematics', 'Algorithm', 'Ben Iraa', 'Nicolas', 1234, 2022-07-19);

//BorrowingView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) BorrowingView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Borrowing;

//insert into BorrowingView

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO BorrowingView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (332, 3, 12, 222, 'manager', 1234, 2022-07-11, 2022-07-27);

//ClearedRecordView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) ClearedRecordsView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM ClearedRecords;

//insert into ClearedRecordView

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO BorrowingView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (34, 3, 12, 222, 'manager', 1234, 2022-07-14, 2022-07-20);

//create ReportVeiw

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) ReportView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Report;

//insert

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO ReportView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (234, 88, 3, crazy me, 123, 2022-07-22, 'returned', 12, 'none', 'no');

//create staffVeiw

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) staffView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html)\* FROM staff;

//insert

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO staffView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (221, 'Benjamin', 'IRADUKUNDA', 'male', 21, 83883772, 'beniraa@gmail.com', 9876272);

//create studentVeiw

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) StudentView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Student;

//insert

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO StudentView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (13, 'Nshuti', 'Blaise', 'male', 21, 838283772, 'nshuti@gmail.com', 87654321);

//create transactionTableView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) transactionTableView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM transactionTable;

//insert into a view

[INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO transactionTableView [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) (89, 332, 'penalty fees', 12, 2022-07-29);

2. Create a view to display all the information in your tables.

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Book\_view;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM BorrowingView;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM ClearedRecordsView;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM ReportView;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM staffView;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM StudentView;

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM transactionTableView;

3. Create a view to update information in any of the two tables of your system.

//Update staffView

[UPDATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) staffView [SET](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) fname= 'Ben' WHERE staff\_ID= 221;

//Update BookView

[UPDATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) Book\_view [SET](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) bk\_title='English' WHERE book\_ID =2;

4. Create a view to delete data in any two of your tables according to any simple condition of your choice.

//StaffView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) staffView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html)\* FROM staff;

//delete

DELETE FROM staffView WHERE staffView.staff\_ID = 221

//StudentView

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) StudentView AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Student;

//delete

DELETE FROM StudentView WHERE `StudentView`.`stud\_ID` = 12

5. In your database, create one view of your choice that considers sub-query.

//BookView To retrieve minimum BookNumber

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) Book\_view AS [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) book\_ID, bk\_title, bk\_name, publisher, auth, bk\_num, pub\_date FROM Book;

//subquery

[SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) bk\_num, book\_ID FROM Book\_view WHERE bk\_num [IN](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ( [SELECT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [MIN](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_min)(bk\_num) );

**Section IV**

1. Create a stored procedure to insert data into your tables.

//Book

CREATE PROCEDURE `BookProcedure`(IN `book\_ID` INT(11), IN `bk\_title` VARCHAR(30), IN `bk\_name` VARCHAR(30), IN `publisher` VARCHAR(11), IN `auth` VARCHAR(10), IN `bk\_num` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO book VALUES (88, 'Hello', 'world','Benjamin', 123456);

//Borrowings

CREATE PROCEDURE `BorrowingProcedure`(IN `borrower\_id` INT(11), IN `book\_ID` INT(11), IN `stud\_ID` INT(11), IN `staffName` VARCHAR(10), IN `studeentNoCopies` INT(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO borrowing VALUES (001,0022,0111,'Benjamin Iradukunda',12345);

//ClearedRecords

CREATE PROCEDURE `ClearedRecordsProceduer`(IN `Cleard\_ID` INT(11), IN `borrower\_id` INT(11), IN `book\_id` INT(11), IN `bk\_title` VARCHAR(11), IN `stud\_ID` INT(11), IN `fname` INT(11), IN `lastname` VARCHAR(31), IN `studentNoCopies` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO clearedrecords VALUES (0011,00011,00022,'Jules',000011,'Benjamin','IRADUKUNDA', 99911);

//Report

CREATE PROCEDURE `ReportProcedure`(IN `report\_ID` INT(11), IN `trans\_id` INT(11), IN `book\_id` INT(11), IN `returnedCopies` INT(11), IN `reportRemarks` VARCHAR(10), IN `numbersOfdays` INT(11), IN `penalities` VARCHAR(31), IN `reportRecieved` VARCHAR(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO clearedrecords VALUES (00020,00222,00202,2222,'Late', 12, 'yes', 'Payment');

//Staff

CREATE PROCEDURE `staffProcedure`(IN `staff\_ID` INT(11), IN `fname` VARCHAR(30), IN `lname` VARCHAR(30), IN `gender` VARCHAR(30), IN `age` INT(11), IN `cantact` INT(11), IN `staff\_email ` VARCHAR(31), IN `staff\_pass` VARCHAR(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT into staff VALUES (00011, 'Benny','crispe', 'male', 0987654321,'beniraa50@gmail',0987654321 );

//Student

CREATE PROCEDURE `studentProcedure`(IN `student\_ID` INT(11), IN `firstname` INT(30), IN `lastname` INT(30), IN `gender` INT(10), IN `age` INT(10), IN `contact` INT(11), IN `student\_email` VARCHAR(31), IN `student\_pass` VARCHAR(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO student VALUES (0001,'IRANKUNDA', 'Benjamin','male', 12, 098765432,'[iradukunda@gmail.com](mailto:iradukunda@gmail.com),');

2. Create a stored procedure to display all the information in your tables.

CREATE PROCEDURE `displayProcedurer`(IN `book\_id` INT(11), IN `bk\_title` VARCHAR(11), IN `bk\_name` VARCHAR(11), IN `publisher` VARCHAR(11), IN `auth` VARCHAR(11), IN `bk\_num` VARCHAR(11), IN `pub\_date` DATE) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT \* FROM book

3. Create a stored procedure to update information in any of the two tables of your system.

CREATE PROCEDURE `displayBorrowing`(IN `book\_id` INT(11), IN `Borrower\_id` INT(11), IN `student\_id` INT(11), IN `staff\_id` INT(11), IN `staffName` VARCHAR(11), IN `NumberOfCpoies` INT(11), IN `DueDate` DATE) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER select \* FROM borrowing;

4. Create a stored procedure to delete data in any two of your tables according to any simple condition of your choice.

CREATE PROCEDURE `DisplayReport`(IN `report\_id` INT, IN `trans\_id` INT, IN `book\_id` INT, IN `bk\_title` INT, IN `ReturnedCopies` INT, IN `ReportPenality` INT) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT \* from report;

5. In your database, a stored procedure view of your choice that considers sub-query.

**Section V**

1. Create after insert triggers for any two tables of your choice.

//Book Table Creating Trigger

DELIMITER

$$

USE Library\_management

$$

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-trigger.html) [TRIGGER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-trigger.html) Book AFTER [INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) ON Book FOR EACH ROW

BEGIN [INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO Book [VALUES](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values)(NEW.book\_ID, NEW.bk\_title, New.bk\_name, NEW.publisher, NEW.auth, NEW.bk\_num, NEW.pub\_date, NOW()); END;

//Transaction Table

DELIMITER

$$

USE Library\_management

$$

CREATE

TRIGGER transactionTable

AFTER INSERT ON transactionTable

FOR EACH ROW

BEGIN

INSERT INTO transactionTable VALUES(NEW.trans\_ID, NEW.borrower\_id, New.trans\_name, NEW.stud\_ID, NEW.trans\_date, NOW());

END$$

2. Create after update triggers for any two tables of your choice.

//Staff Table Update

DELIMITER

$$

USE Library\_management

$$

CREATE

TRIGGER StaffUPDATE

AFTER UPDATE

ON staff FOR EACH ROW

BEGIN

INSERT into staff VALUES (user(), CONCAT('Update staff Record ',

OLD.staff\_ID,' Previous Class :',OLD.fname,' Ben',

NEW.age));

END

$$

//Student Table Update

$$

USE Library\_management

$$

CREATE

TRIGGER StudentUPDATE

AFTER UPDATE

ON Student FOR EACH ROW

BEGIN

INSERT into Student VALUES (user(), CONCAT('Update Student Record ',

OLD.stud\_ID,' Previous Borrowing :',OLD.gender,' female',

NEW.age));

END

$$

3. Create after delete triggers for any two tables of your choice.

//Student Table Delete Student Record

USE Library\_management

DELIMITER

$$

CREATE TRIGGER StudentDelete

AFTER DELETE ON Student FOR EACH ROW

-- Edit trigger body code below this line. Do not edit lines above this one

BEGIN

INSERT into Student VALUES (user(), CONCAT('Delete Student Record ',

OLD.stud\_ID,' Name :',OLD.fname, '-> Deleted on ', NOW()));

END;

$$

//TransactionRecordsTable Delete Transaction records

USE Library\_management

DELIMITER

$$

CREATE TRIGGER transactionTableDeleteRECORDS

AFTER DELETE ON transactionTable FOR EACH ROW

BEGIN

INSERT INTO transactionTable VALUES (user(), CONCAT('Delete transaction Record ',

OLD.trans\_ID,' Name of transaction:',OLD.trans\_name, '-> Deleted on ', NOW()));

END;

$$

**Section VI**

1. Create a user with your names as username and your student number as password and grantee all privileges to the created user.

CREATE [USER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'BenjaminIRADUKUNDA'@'localhost' IDENTIFIED BY '221018380';

2. Create a user with your "names\_semi" as username and your student number as password and give him insert, update and delete privileges to the created user.

//creating user

CREATE [USER](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'IRADUKUNDA\_semi'@'localhost' IDENTIFIED BY '221018380';

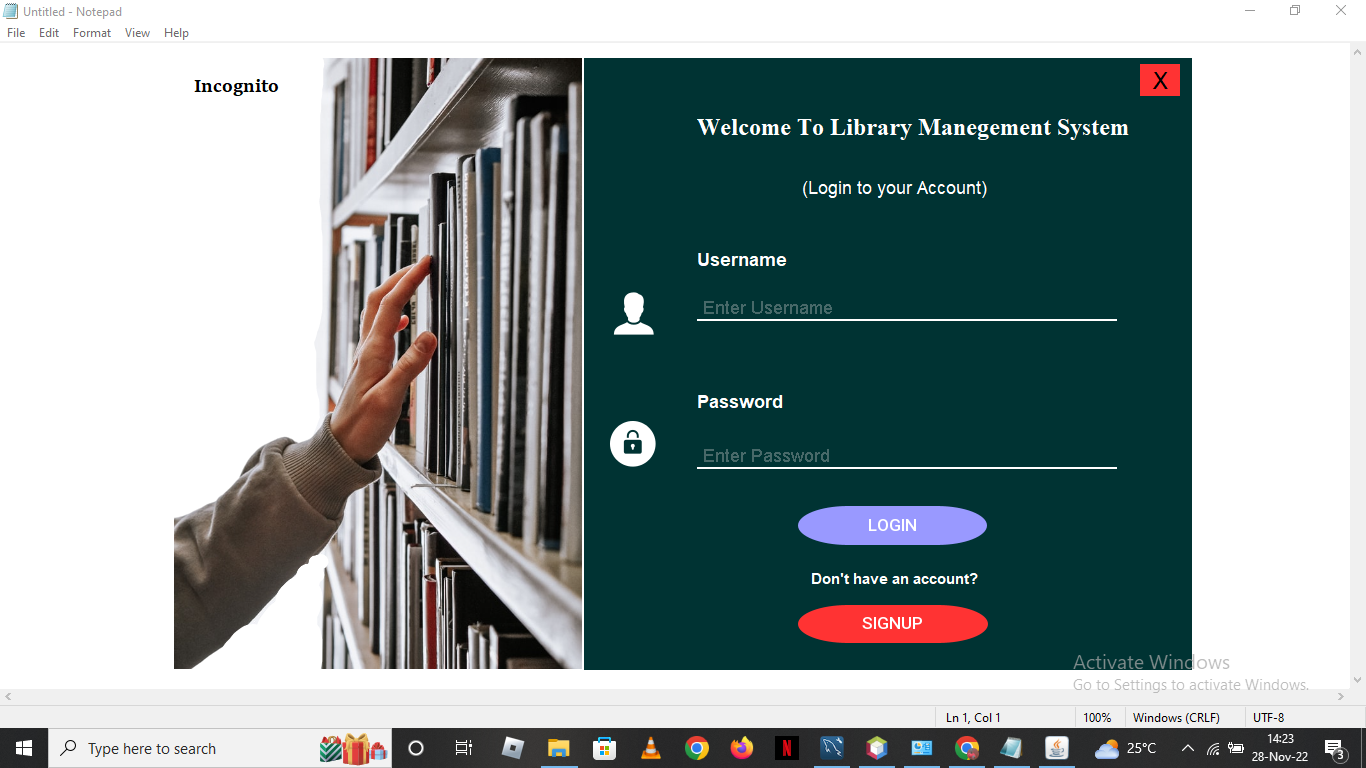
//Grant Privileges

GRANT [INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html), [UPDATE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html), [DELETE](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) ON \*.\* TO 'IRADUKUNDA\_semi'@'localhost';

3. Revoke insert privileges to the last user you created.

REVOKE [INSERT](http://localhost:8080/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) ON \*.\* FROM 'IRADUKUNDA\_semi'@'localhost';

PART 3: JAVA APPLICATION [Screenshots].

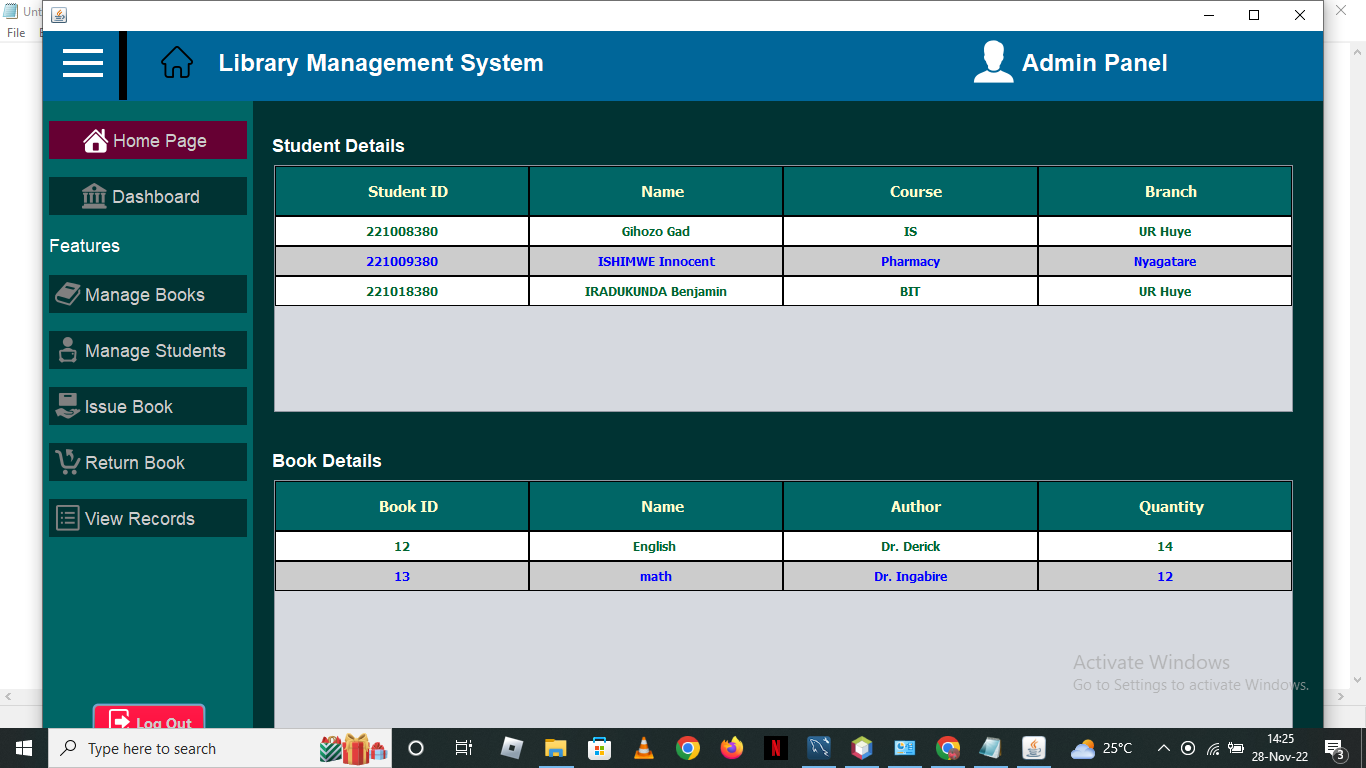


**Login page:** Allows the user to log in the system with the credential that is stored in the database

Username: BenIraa password: 1234



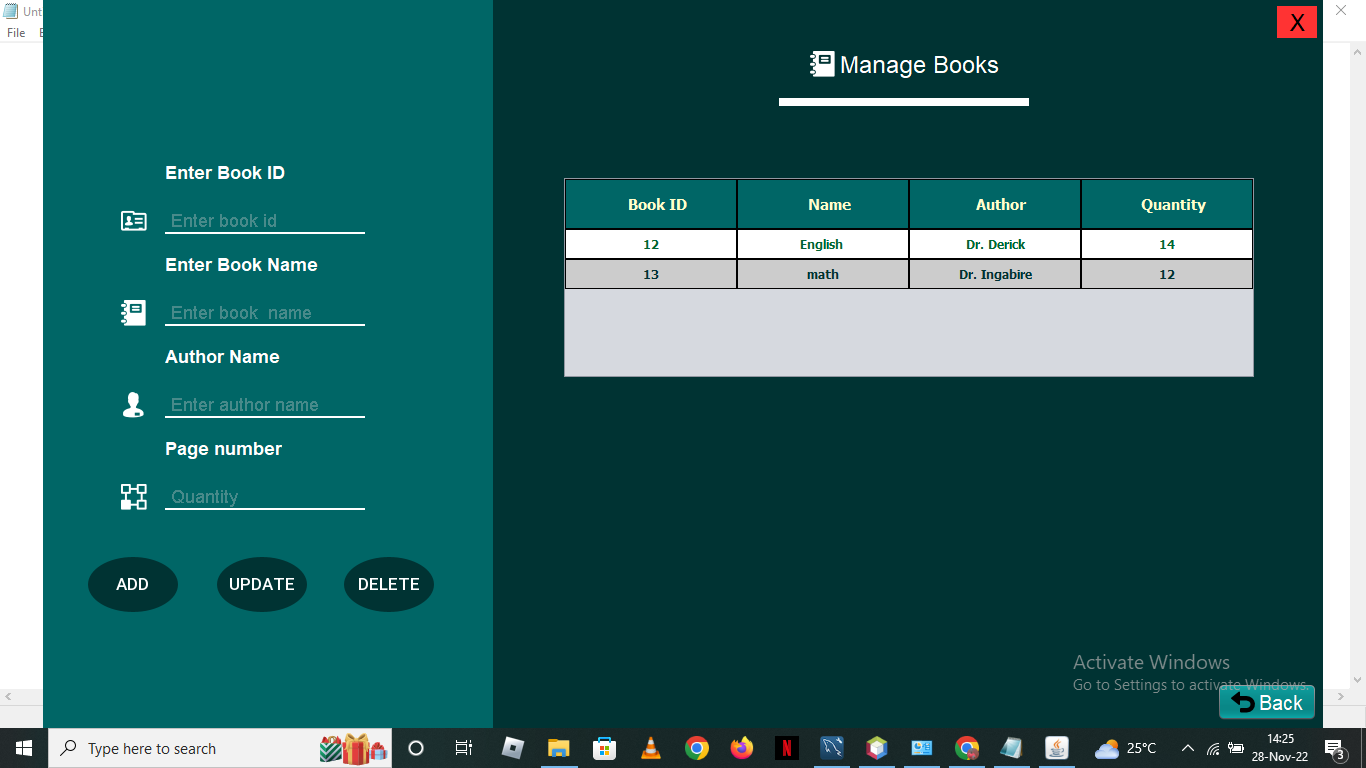
**Sign up page:** New user can create account with his or her information (This user can access the system as Admin)



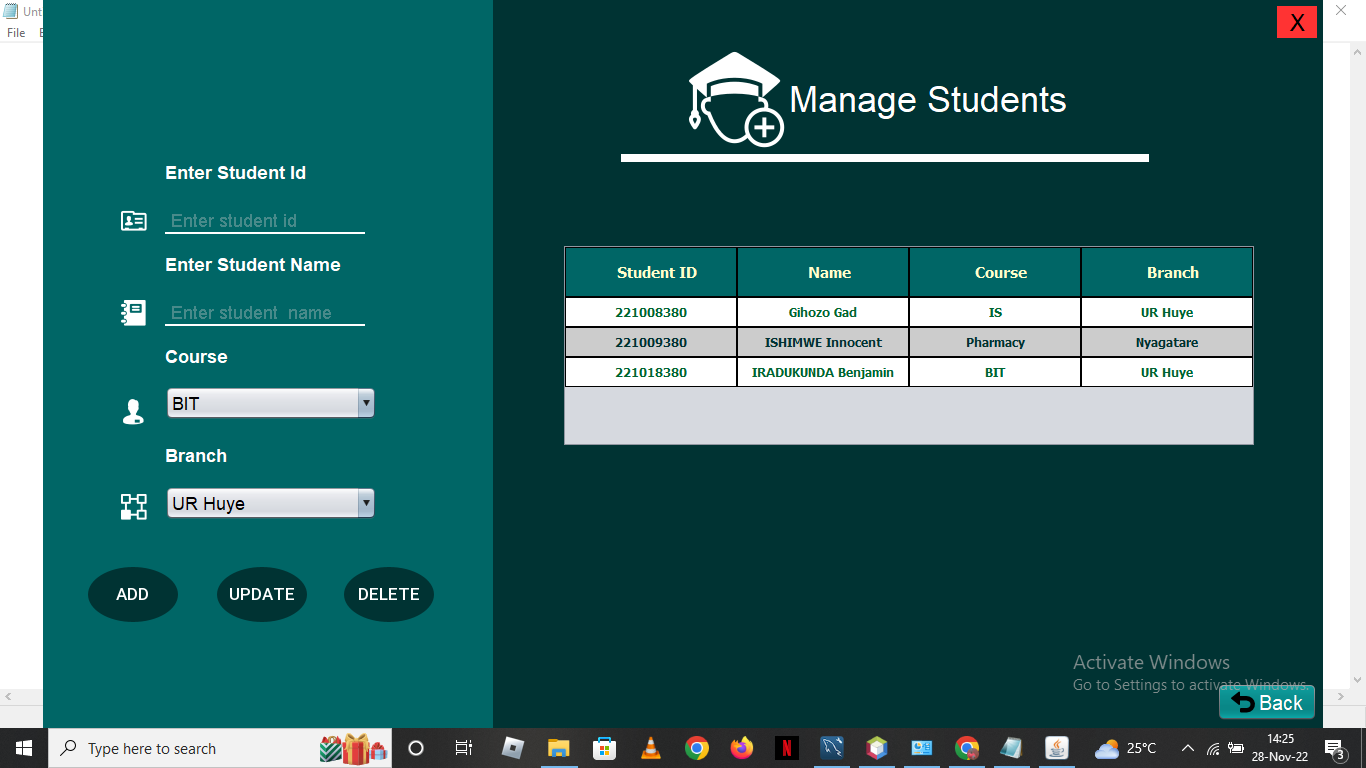
**Landing page:** As picture shows there are 3 parts one is nav bar, side bar and content side,

In the content side includes two tables one for student or members which stored in our system

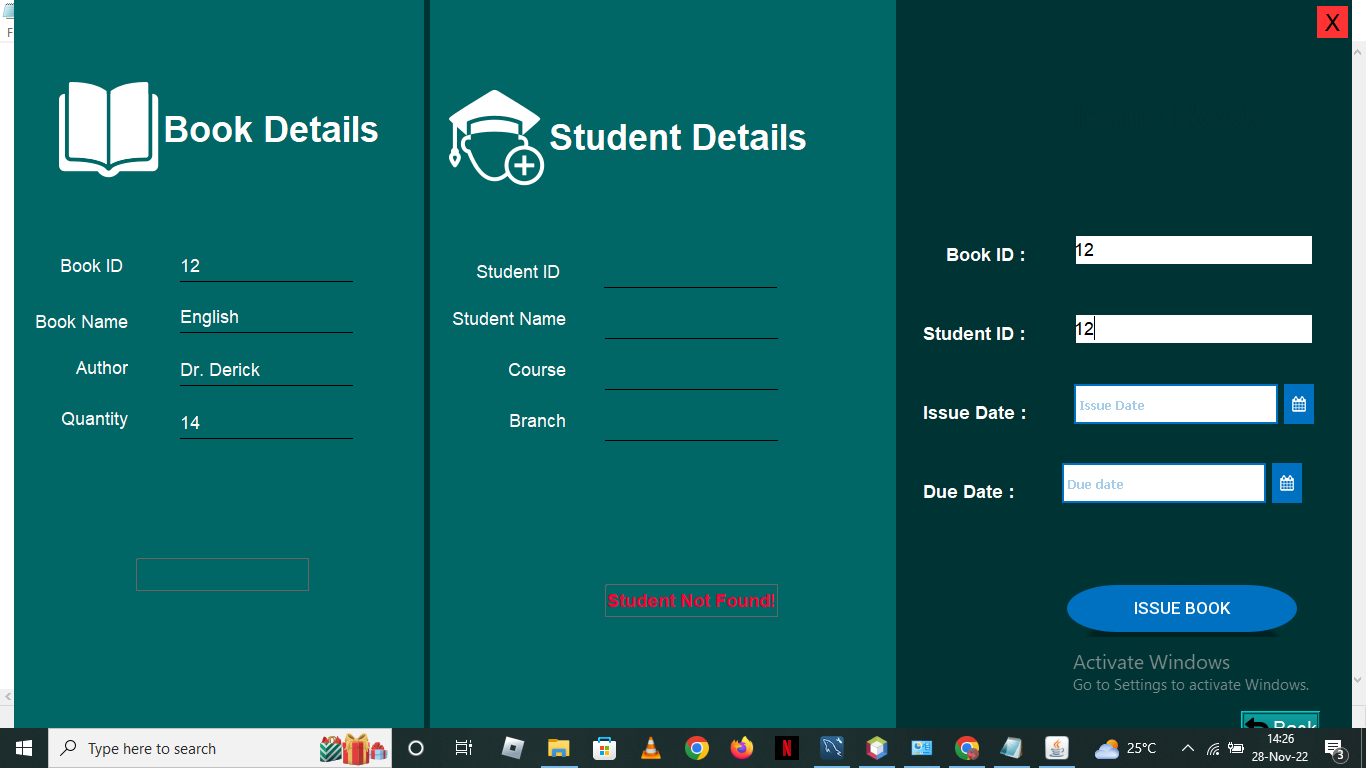
Other table for books details we have in the system



**Manage Books:** Allow admin to add, update, delete book



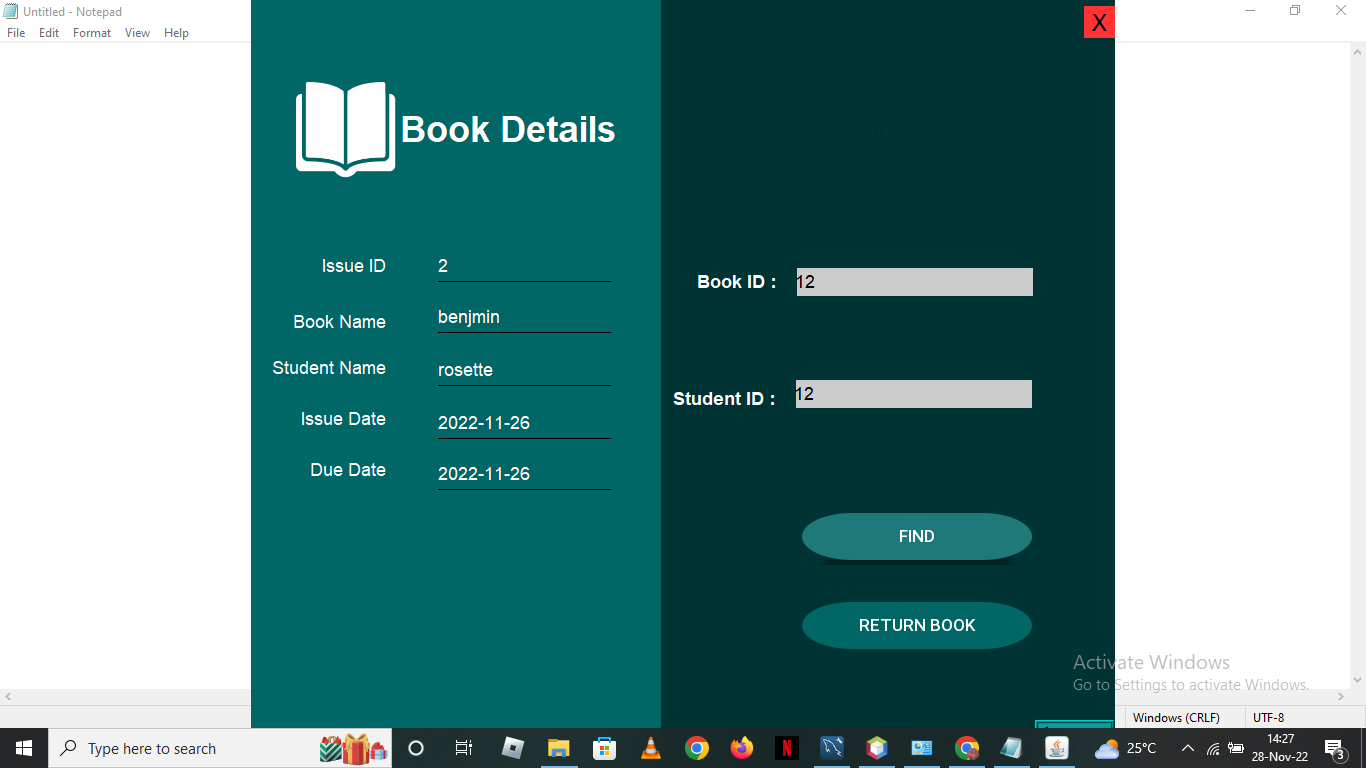
**Manage student:** Allows admin to add, update, and delete student / members stored in our system



**Issue book:** has two parts

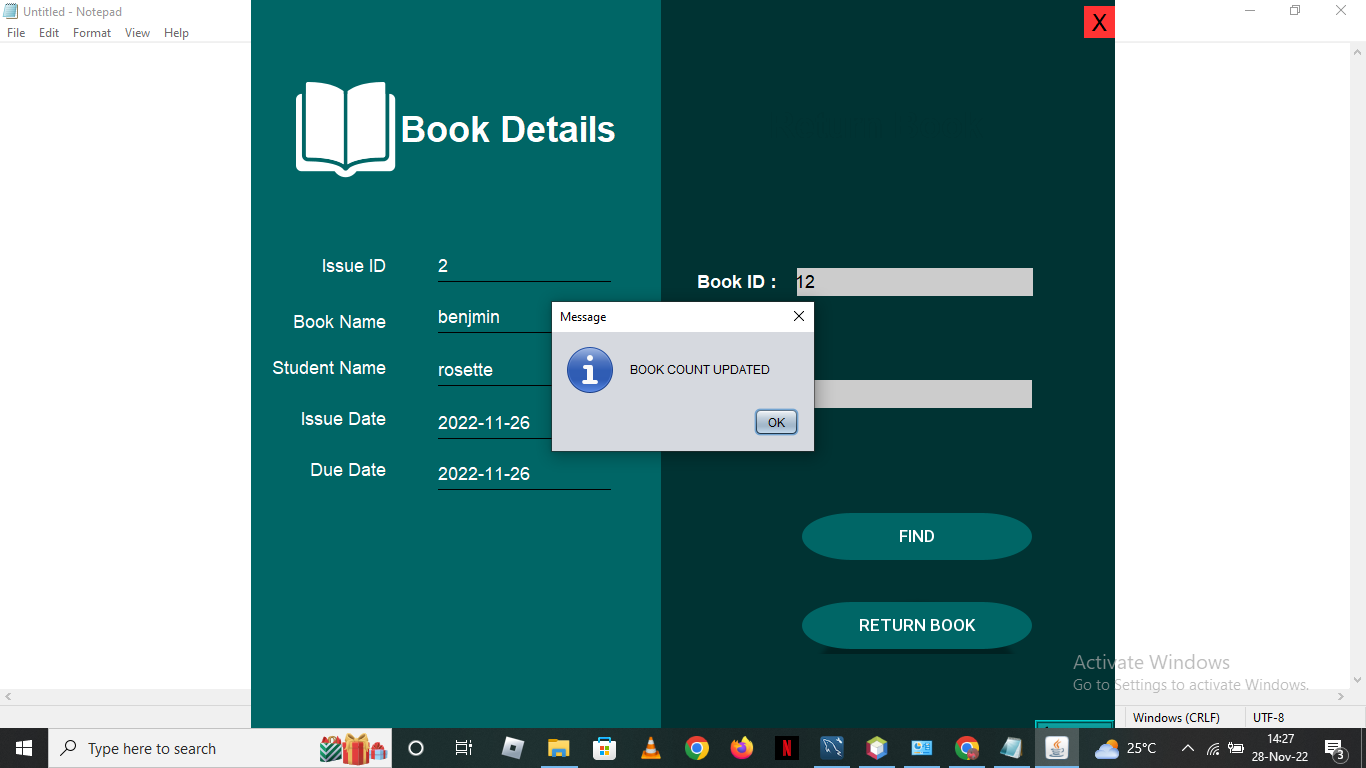
1. **Book Details:** This allow admin to find the book that is stored in the database using the input of the **Book ID:** and if he clicks in the other field system retrieve the data of the book according to ID if there is no Book with that ID at the bottom there is buttons which retrieve the message
2. **Student Details:** This allow admin to find the student that is stored in the database using the input of the **Student ID:** and if he clicks in the other field system retrieve the data of the book according to ID if there is no Student with that ID at the bottom there is buttons which retrieve the message

Admin using that two IDs to assign book to student and the dates : Borrowed at And the date to be Returned!

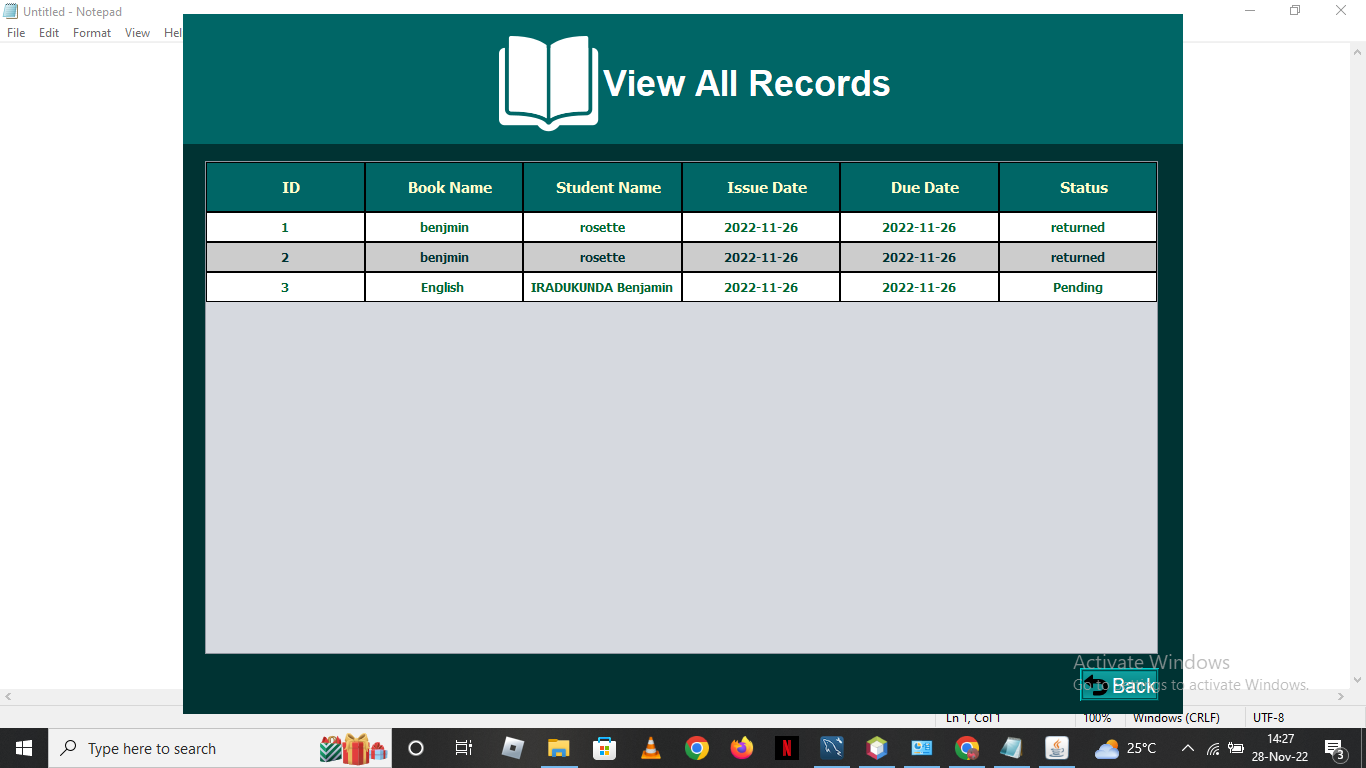


**Return Book:**  That two fields admin use that to find Book ID of the book which is Borrowed and the Student ID who borrowed the book after that he click on Find after that the data retrieved on the left side includes all information of the book and someone who was assigned the book

Then Click on RUTURN BOOK button to reassign book to student



And the message was to count the times that book was borrowed in order to know the favorite books



**RECORDS:** Admins use this page to know the information of the books was returned or not!

To Run This: I included the .txl File in the folder use that guide to Run Java Application.

**Conclusion:**