Library Management System

- **Objective:** Develop a Library Management System to with features like book cataloging, borrower management, due date tracking and fine calculation.

- **Scope:** Application has been developed for the Purpose of issuing books to Students of that University as well as any institute. Also, Teaching staff should be able to access those books. Also, whenever there is any update regarding books or if there are any new books added in library, then Student from that Institute should get notified.

- **Technology:** Core Java, SQL, JDBC etc.

- **Tools Used:** VS Code, MYSQL etc.

- **Functionality**

1. **Add/View/Delete/Update Books in Library.**
2. **Add the Department Details.**
3. **Add/View/Delete/Update Students.**
4. **Display Department-Wise Students.**
5. **Count Category-wise Books.**
6. **Display Category-wise Books.**
7. **Search any Particular Book.**
8. **Issue Book to Student.**

9. **Display Student wise Books.**

10. **Display List of Most issued Books (Most read Books).**

11. **Show the Remaining Stock after Book Issue.**

12. **Update the Book Status with Return Date.**

13. **Sending Update regarding Book details to Students by Email.**

- **Tables Required**

1. **Department Table**

Sr. no	Field	Data Type	Size	Constraints
1.	Dept_id	Int	5	Primary key auto_increment
2.	Name	varchar	30	Not Null

- **Dept_id:** Holds the Department Id which will be Unique and acts as a Primary Key which can be used as foreign key in any other table.
- **Name:** Holds the Name of the Department to indicate Particular Student belongs to which Department will be Not Null.

2. **Students Table**

Sr no	Field	Data Type	Size	Constraints
1.	Sid	Int	5	Primary Key auto_increment
2.	Name	Varchar	100	Not Null
3.	Email	Varchar	100	Not Null Unique
4.	Contact	Varchar	100	Not Null Unique
5.	Dept_id	Int	5	Foreign key on delete on update cascade

- **Sid:** Holds the Unique Student Id of that Student and will be Primary key.
- **Name:** Store the Name of the Student.
- **Email:** Stores the Email Id of a Student which will be Unique and necessary to use it as email will be needed for sending Updates to Student on their Email Id.
- **Contact:** We will be Storing the Contact details of a student here.
- **Dept_id:** Department Id will be foreign key from Department table which shows relation between both tables.

3. **Book Table**

Sr. No	Field	Data Type	Size	Constraints
1.	Bid	Int	5	Primary Key auto_increment
2.	Name	Varchar	100	Not Null
3.	Category	Varchar	100	Not Null
4.	Author	Varchar	100	Not Null
5.	Publication	Varchar	100	Not Null
6.	Language	Varchar	50	Not Null

- **Bid:** This will be the unique Book Id which is Unique for Books in Library.
- **Name:** here we will store the Name of the Book.
- **Category:** Here we will be storing that book belongs to which Category.
- **Author:** we will mention Author of Book here.
- **Publication:** Holds publication of the Book.
- **Language:** Here we will be Storing the Language of the Book.

4. **Book Issue Table**

Sr. No	Fields	Data Type	Size	Constraints
1.	Issueid	Int	5	Primary Key auto_increment
1.	Bid	Int	5	Foreign key on delete cascade on update cascade
2.	Sid	Int	5	Foreign key on delete cascade on update cascade
3.	Status	Varchar	20	Null
4.	Issuedate	Date	-	Not Null
5.	Returndate	Date	-	Null

- **Issueid:** here Book Issue ID which will be Unique will be Stored.
- **Bid:** Here we are storing Book Id which is Primary key from Book Table and will be foreign key here.
- **Sid:** Here Student id will be primary key from Students table and here it is foreign key.
- **Status:** This column will hold the Status of the book whether it is issued or not.

- **issuedate:** Here we will be storing the date when the Book is issue to the student.
- **Returndate:** Here we will be storing the date when Student need to return the Book.

- **Classes and Methods Used**

1. **ClientApplication:** This is Class which contains main Method. Also, When we create Service class we create it's Object in Client Application and call it's Method here as well as Object of Model class is created in ClientApplication only and call the Setter and Getter Methods here.

2. **Model Class:** We have 4 different Model classes where we are creating POJO class and defining Setter and getter methods.

    a. **Bookmodel:** Here we are writing the Getter and Setter methods to fetch and pass the date inside Book Table.
    b. **Studentmodel:** Here we are writing the Getter and Setter methods to fetch and pass the date inside Students Table.
    c. **Bookissuemodel:** Here we are writing the Getter and Setter methods to fetch and pass the date inside Bookissue Table.
    d. **Department:** Here we are writing the Getter and Setter methods to fetch and pass the date inside Department Table.

3. **Repository Class:** We have 6 different Repositories where we are writin database Logic which are as follows.

    a. **Bookrepository:** Here we are writing all the database logics to perform Book Operations.
    b. **Studentrepository:** Here we are writing all the database logics to perform Student Operations.

c. **Departmentrepository:** Here we are writing all the database logics to perform Department Operations.
d. **Bookissuerepository:** Here we are writing all the database logics to issue Book to Students and for other functionality.
e. **Studentbookrepository:** Here we are writing all the database logics to show Student wise Books and other Functionality.
f. **Studentdeptrepository:** Here we are writing all the database logics to display Department wise Student and some other functionalities.

4. **Service Class:** Here we are writing the Business Logics. Means here we will be creating object of repository.

    a. **Bookservice:** Here we are calling the methods written in Bookrepository by creating its Object.
    b. **Studentsservice:** Here we are calling the methods written in Studentsrepository by creating its object.
    c. **Bookissueservice:** Here we are calling the methods written in Bookissuerepository by creating its object.
    d. **Studentdeptservice:** Here we are calling the methods written in Studentdeptrepository by creating its object.
    e. **Studentbookservice:** Here we are calling the methods written in Studentbookrepository by creating its object.
    f. **Departmentservice:** Here we are calling the methods written in Departmentrepository by creating its object.

5. **DBConfig:** In this class, we are loading the Properties File where we have written DB credentials also we are establishing DB connection here.

6. **DBHelper:** In this class, we have created reference of Connection, ResultSet, PreparedStatement and inherit that class in every repository where we are writing DB logics.

7. **PathHelper:** Here we are accessing the path where we have Store Properties File and calling its method in DBConfig to load the Properties File.

8. **EmailSender:** This class is used to write the Logics for sending the Email. For that we need to use Email API nae as Jakarta which we need to import in our Project. Also, when we are executing the Functionality of Email sending then we need to call the Method present in EmailSender.