**Project Title:**

Library Management System

**Purpose Of the Project:**

This project is to create a library management system for students. Through this software one can issue/return a book to the library and the librarian can update/delete the record of books kept in the library.

Features:

This project is mainly had two parts.

1. Library Members Part

Feature for Member.

* 1. Each Library members should have a Member ID, which is Unique. By this Unique ID member can take a book from Library and can Return to the Library.
  2. Maximum two books can be issued to a member.

1. Libarian Part

Feature for Librarian

* 1. Libarian will have a user credential without this credential he can’t perform any job.
  2. Libarian can Add/Delete/Update Books in Library and Print a list of Books.
  3. Libarian can Add/Delete/Update Member in Library and Print a list of Members.

**Team members:**

Ananya Mondal (2105693)

**List of Data Structures will used**

1. **Back End:**

MySQL database.

1. **Front End:**

Python Programming Languagewill be used to develop the software.

**DFD For Member**

Library Member

Issue

Return

Return Complete

Update Stock

Issue Book

Enter Book

Match

Enter Member Id

Check Availability

Invalid Member

Enter Book

Not Match

Match

Enter Member Id

**Operations to be perform on Book:**

**Insertion:** We are using insertion operation to add books in data base book master table.

**Deletion:** We use deletion operation to delete books in data base book master table.

**List or print all the Books:** There is also an option where all the details of the books are printed. Name of the book, Author Name, available quantity of the books that can be issued and total number of books that library has.

**Search**: There is an option to search a particular book in the library.

**Operations to be perform on Member:**

**Insertion:** We are using insertion operation to add member in data base member master.

**Deletion:** We use deletion operation to delete member from data base member master.

**Updating:** The librarian can update the member’s name and other details.

**Search:** The librarian can search the member to see the details.

**List or print all the Member:** There is also an option where all the details of the member are printed.

**Operations to be perform on Library Manage:**

**Issue a Book:** At First, we need to search a member based on Member Id from member list. If Member is found then we will search for a book from book details. If book found and number of books is available and member have not consumed two books. Then book will be issued. We need to update members record as well as we need to update book master.

**Return a Book:** To return a book we will accept member id, which need to be searched issue details if found need to accept which book need to return that needs to be searched and also need to check that book is allocated to that member. Then we need to update available quantity of the book and update the member node to release the book.

ER Diagram

Member

Issue/Return

Book

Member and Book is in one-to-one relationship.

|  |  |  |
| --- | --- | --- |
| **Member Entity and Its Properties** | **Books Entity and Its Properties** | **Issue/Return Properties** |
| Roll No (PrimaryKey) | Book Name | Rollno (ForeignKey) |
| Name | Author Name | Bookid (ForeignKey) |
|  | Total number of Copies | Issue Date Time |
|  | Available Copies to Issue | Schedule Return Date |
|  | BookId (PrimaryKey) | Issueid (PrimaryKey) |
|  |  |  |

**DATABASE Script**

**CREATE DATABASE `newdb`; /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;**

**CREATE TABLE `book` (**

**`bookid` int NOT NULL,**

**`bookname` varchar(150) DEFAULT NULL,**

**`AuthorName` varchar(150) DEFAULT NULL,**

**`noofBook` int DEFAULT '0',**

**`Available` int DEFAULT '0',**

**PRIMARY KEY (`bookid`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;**

**CREATE TABLE `issue` (**

**`issueid` int NOT NULL AUTO\_INCREMENT,**

**`rollno` int DEFAULT NULL,**

**`bookid` int DEFAULT NULL,**

**`issuedate` datetime DEFAULT NULL,**

**`returndate` datetime DEFAULT NULL,**

**PRIMARY KEY (`issueid`)**

**) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;**

**CREATE TABLE `students` (**

**`roll` int NOT NULL AUTO\_INCREMENT,**

**`name` varchar(150) DEFAULT NULL,**

**PRIMARY KEY (`roll`)**

**) ENGINE=InnoDB AUTO\_INCREMENT=2105694 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;**

● Input data and output generated

1) Add book

 Input data Enter name of the book:

Enter quantity of book:

 Output generated Book added successfully...

2) Delete book

 Input data Enter name of the book:

 Output generated Book deleted successfully...

2) Update book

 Input data Enter name of book:

Enter quantity of book to add more:

 Output generated Successfully updated...

4) Print book details

 Output generated All the details of book will be printed

7) Isuue book by user

 Input data Enter your university ID:

Enter name of book:

 Output generated Book issued successfully!!

Current date time:

Due date and time:

8) Return book:

 Input data Enter your university ID:

Enter name of book:

 Output generated Book returned successfully!!

Current date time:

Return date and time:

Project Plan:

**Ananya Mondal (2105693):**

Database Design in MySQL.

Crating MySQL connection with Python

Creating Book Master, Student Master Interface in Python

Creating Interface to Issue and Return Books

Code Repository: