**LMS PROJECT BY BEN MUNYASIA**

**What is LMS?**

A library management system is software that is designed to manage all the functions of a library. It helps librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.

**Technologies that will be used**

* Python framework, Django
* HTML
* CSS
* JS
* React (may be integrated)
* Database – sqlite (django’s default database)
* React-Native (mobile application)

**Functions of the LMS**

1. Shows available books and their respective shelves.
2. Fine calculation
3. Due date settings
4. Shows the number of copies of a certain book.
5. Online Public Access Catalogue, a digital catalogue that enables the user to search for books, journals, or any other material.
6. Sending an email or SMS to members with overdue books.
7. Request forms, sending a request for books that do not exist so the librarian can add them.

**USER INTERFACE**

**NB:** When I am referring to books it may be journals, articles, novels, etc.

**Admin’s section (Librarian)**

The admin must be able to see the following:

* The clients that have borrowed books.
* Books that are overdue and client’s detail.
* Number of copies that have remained on the shelves.
* See books requested by users.

The admin should be able to do the following with the system:

* CRUD operation on books.
* Find the book’s location i.e., which shelf number.

**User’s section**

The user must be able to see the following:

* See the books available.
* Books that can be borrowed physically.
* If the book they want is available or they are all off the shelf.

What can the user do with the system?

* Search for a book
* Read an online book
* Book a physical book.
* Edit user profile.

**Functional Requirements**

* Should have a filter and search functionality.
* Should be able to store clients’ details.

**1.** Two types of accounts:

- Librarian account.

- Client account.

2. Database to hold the following information:

- List of Books.

- List of clients.

- List of Librarians

- Books borrowed and the ones who borrowed them.

- Clients with fines.

3. A page for book acquisition.

4. A page for reading books online.

5. The database be able to perform CRUD.

6. Viewing the book's details should be possible.

7. a Total number of specific books should be known.

8. the Current number of books on shelves should be possible.

9. Books are divided according to their categories.

10. Search function.

11. Section in the librarian account to add a book.

12. Login and logout functions.

13. An admin section for adding, removing, and updating books

**Non-functional requirements**

1. Filtering booking according to categories.

2. Using icons and visuals to enable an intuitive understanding of a system.

3. Display summary detail of the book such as the name of the author, number of pages, it tags.

4. It has a mobile view.

**DATABASE DESIGN**

Which models will be used in Django?

* Books model, this will hold all the information of the books.
* User’s model, clients’ detail
* Fine model, clients with fines.
* Book’s Borrowed model, this will show which books were borrowed and borrowed them.
* News model holds any updates about the library.

**BOOKS MODELS**

|  |  |
| --- | --- |
| Name of Book |  |
| Author |  |
| Cover Image |  |
| Category | The genre of the books |
| Shelf Location |  |
| Serial Number | PK |
| Description |  |
| Access Mode i.e., it is a book that can be obtained physically or is it read online |  |

**CLIENTS MODEL**

|  |  |
| --- | --- |
| USERNAME |  |
| NAME |  |
| EMAIL |  |
| Profilepic |  |

**FINES MODEL**

|  |  |
| --- | --- |
| USERNAME/ID | Foreign key |
| BOOK’S SERIAL NUMBER | Foreign key |
| CHARGE |  |

**BORROWED** **MODEL**

|  |
| --- |
| Username/Id - Foreign key |
| Serial Number - Foreign key |
| Date Borrowed |
| Due Date |
| Returned Date (When the Book Was Returned) |

BOOKS’ MODEL NORMALIZATION

**BOOKS MODEL**

|  |  |
| --- | --- |
| NAME |  |
| SERIAL NUMBER | PK |
|  |  |

**BOOKS DETAILS MODEL**

|  |  |
| --- | --- |
| Author |  |
| Cover Image |  |
| Category |  |
| Shelf Location |  |
| Serial Number | **FK** |
| Description |  |
| Access Mode i.e., it is a book that can be obtained physically or is it read online |  |
| Amount |  |

**PAGES**

* Front page, has news regarding the lib site.
* Books List View page, shows all the books in the lib and has a search and filter function.
* Book view page, a page that shows information about the book selected by the user.
* User Profile

**BRAINSTORMING**

**How does the fine model function:**

* **What am thinking is it should be connected to the borrowed books model.**
* **This will essentially check whether the borrowed book is overdue if so the client is transferred to the borrowed books model.**

**What does the borrowed books model contain:**

* **Username**
* **Book title**
* **Date borrowed**
* **Issued by**

**Booking function**

1. **Name**
2. **Book title**

**IDEA OF IMPROVING ILMS**

Making an offline ILMS, which syncs with the online databases, in case there is no internet connection.