**Library Management system**

# Purpose

* Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.
* The site helps the student save time to search for the book he wants. If it exists, he will reserve it through the site instead of going to the library and searching for and that will save him time and effort
* Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.
* We will use Recommendation system seeks to predict the rating or preference a user would give to an item given his old item ratings or preferences.
* **Recommender System** refers to a **system** that is capable of predicting the future preference of a set of items for a user, and recommend the top items. One key reason why we need a **recommender system** in modern society is that people have too much options to use from due to the prevalence of Internet.

# Product Scope

**The objectives of this project are:**

- To offer a web application service which can assist the reader to find a book which may user login to site to find it.

- The software keeps track of all information about the books and their complete details.

- The system contains database where all the information will be stored safely.

-arrange, organized and preserve the information.

# ­­­­­­languages

-the application will run on all browsers, in addition to MYSQL as database , laravel as development framework .

# Overall Description

## **Product Functions:**-

***Login:*** *enables user to login our website.*

***Register:*** *user/admin can register by his personal detail.*

***Search:*** *enables user to search for certain books.*

***Book:*** *enables user to book a book for borrow.*

***Delete:*** *enables admin to delete book or category­­­­­­.*

***Add:*** *enables admin for add book or category.*

***Borrow:*** *enables admin to complete user’ booking*

*For borrow.*

## User Classes and Characteristics:

**User:** anyone interested in reading.

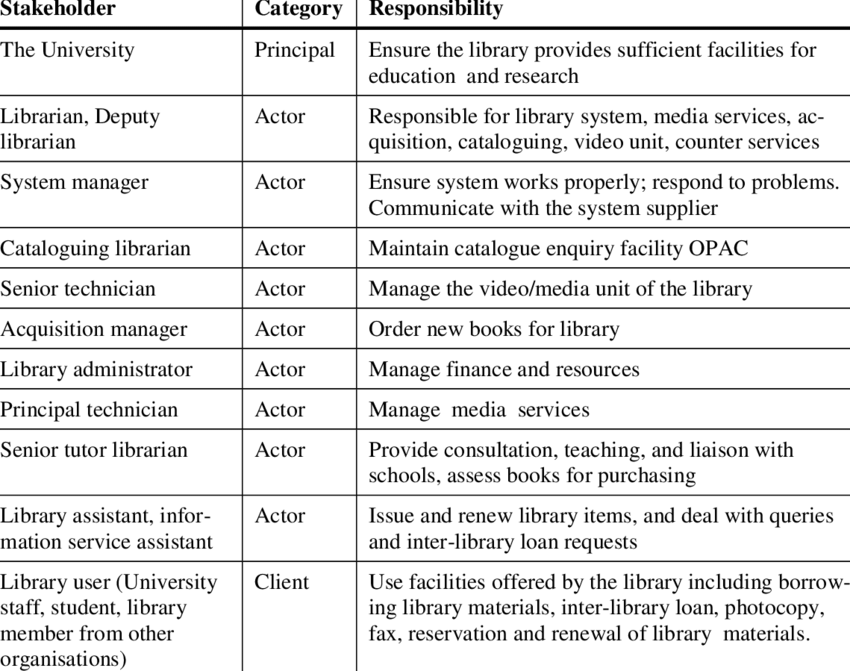
Can be **faculty's** student, teaching staff members, or employees.

**Librarian:** person that manage website (admin).

## Operating Environment:

• Windows 2000  
• Windows XP  
• Windows Vista  
• Windows 7  
• Windows 8  
• Windows 10

# Stakeholders



# Design and Implementation Constraints:

- Student user: can’t borrow more than 3 books.

- Teaching staff members' user: can’t borrow more than 5 books.

- Employee user: can’t borrow more than 2 books.

-User can renew the borrowing process for one time only.

-the maximum duration of borrow for student is a 7 days.

-the maximum duration of borrow for Teaching staff members is a 14 days.

-the maximum duration of borrow for Employee is a 14 days.

-user can't borrow the book if number of book's copies doesn't allow.

-user can't borrow :

* References in general: such as: dictionaries, manuals, knowledge.
* University messages except for Teaching staff members and for only 2 days.
* Audio-visual materials and CDs
* Scientific sessions
* Single-copy books
* United Nations Publications.

-user can’t book a book without login.

-website is developed in laravel framework , using visual studio code IDE ,bootstrap 4 Libraries and MYSQL database.

## User Documentation:

**for Register in website**

The user enters data :

-first name

-last name

-username

-user E-mail

-user password

-confirm password

The application verify the E-mail to valid E-mail by sending unique code in verified message.

-user enter the code as input

* For login to website

The user enters data

-his E-mail

-his Password

Application check from DB e-mail and password entered.

* For Search about book

-user enter his prefer book name

-select the book

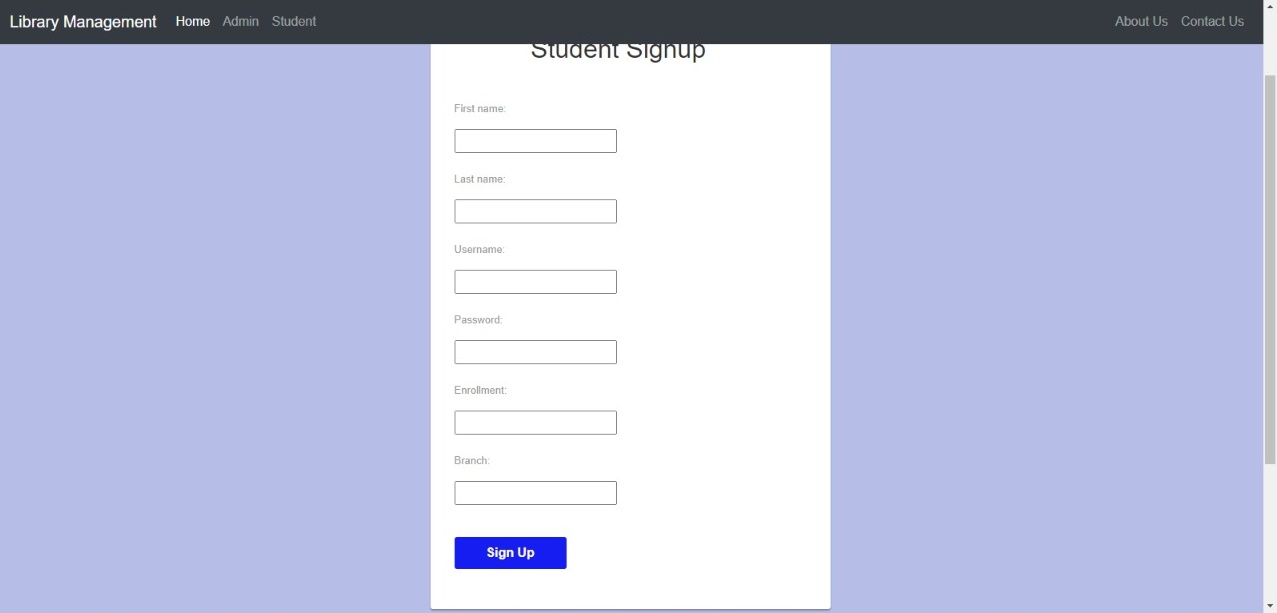
* The user can borrow book by press Book button.

**External Interface Requirements**

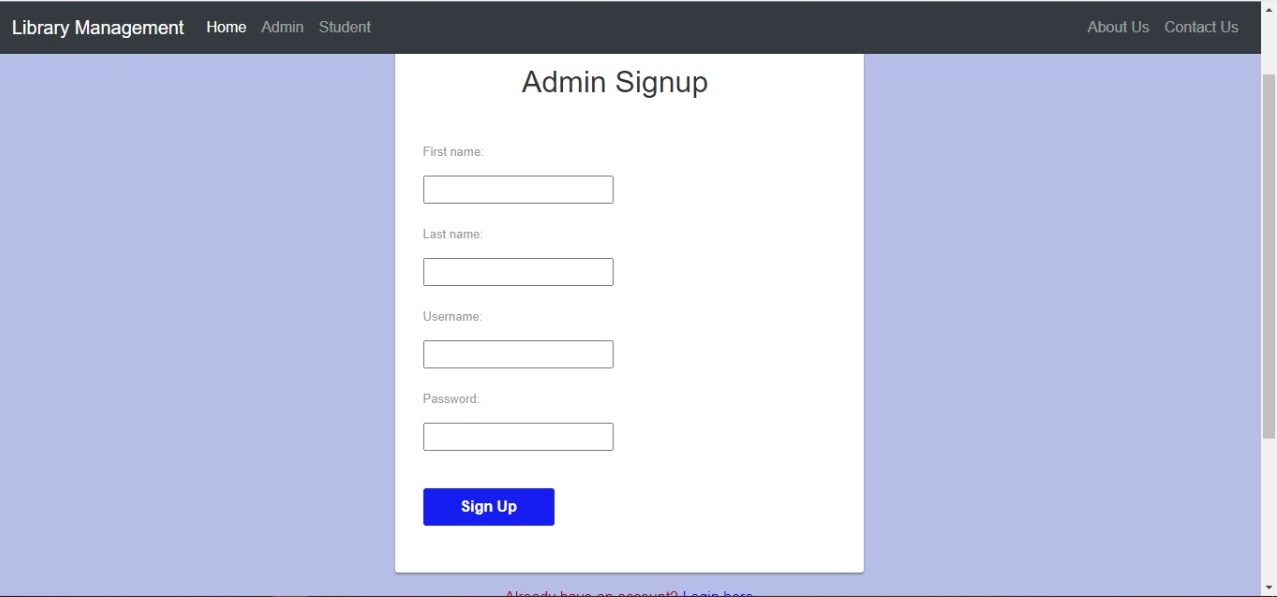
## User Interfaces:

# Register

# -*student register*

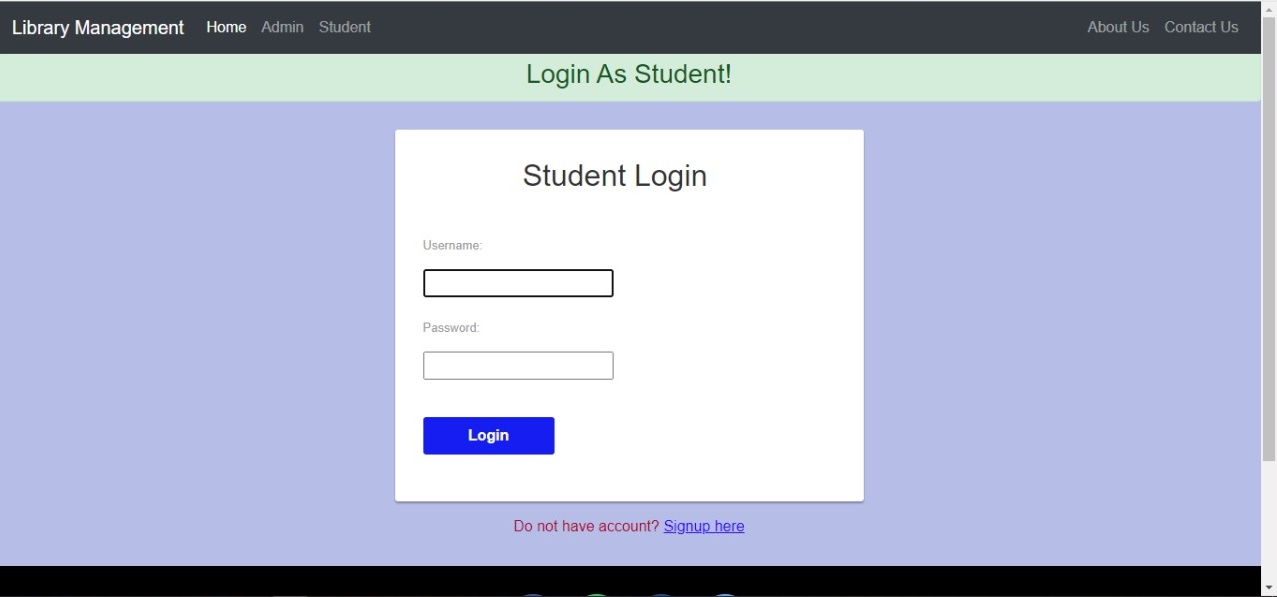


# -*Admin register*

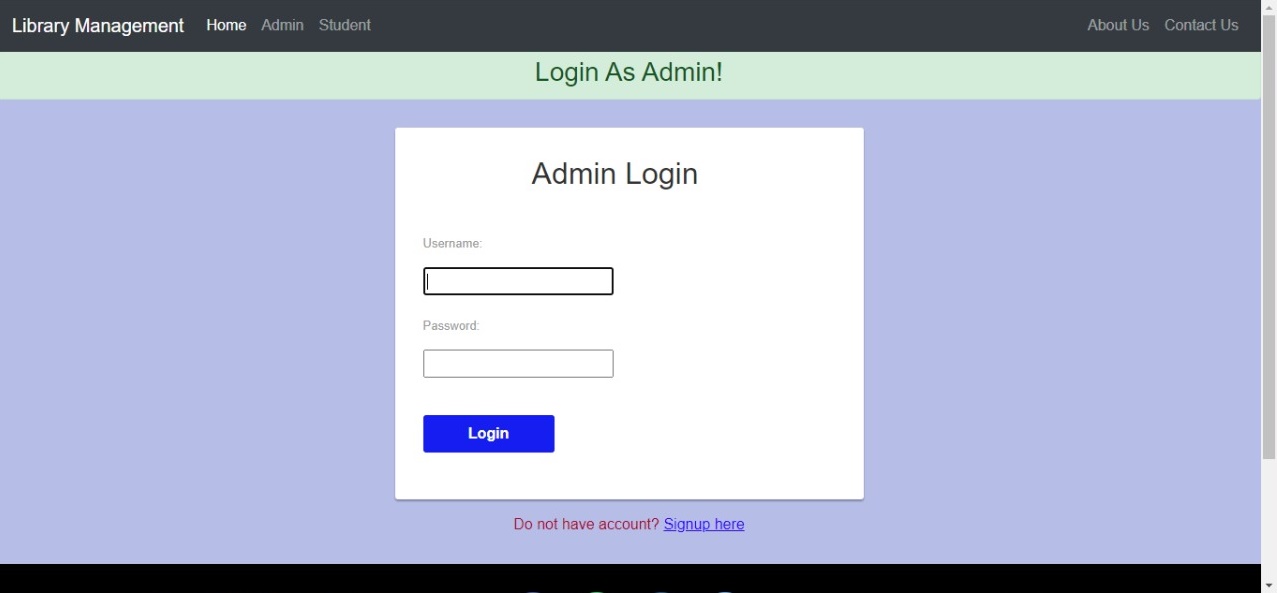


Login

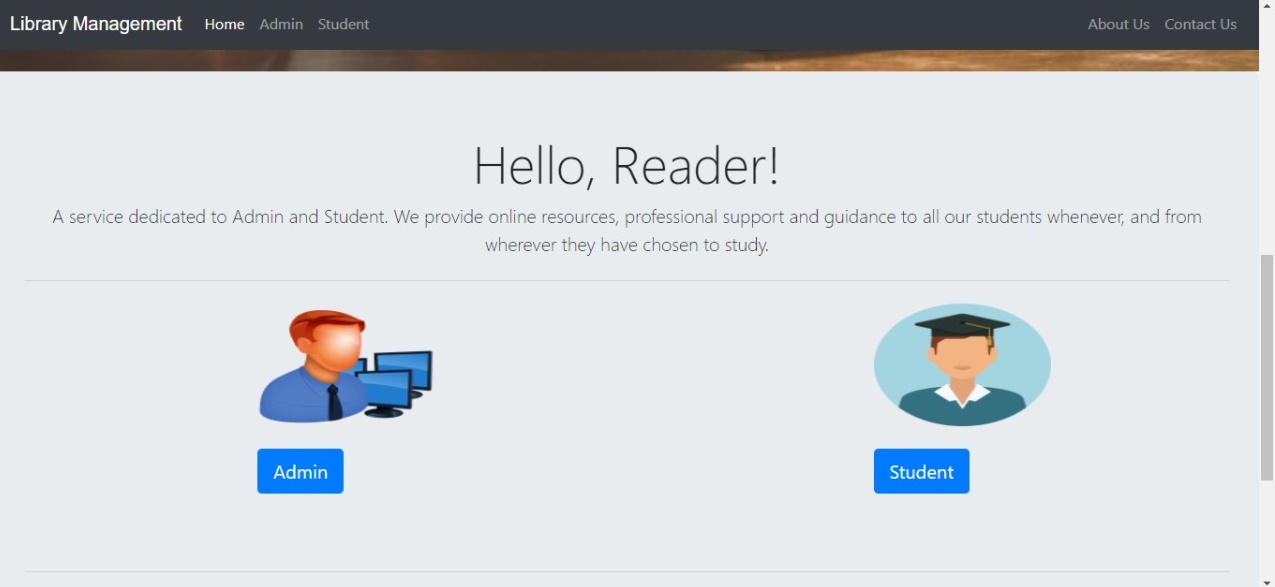
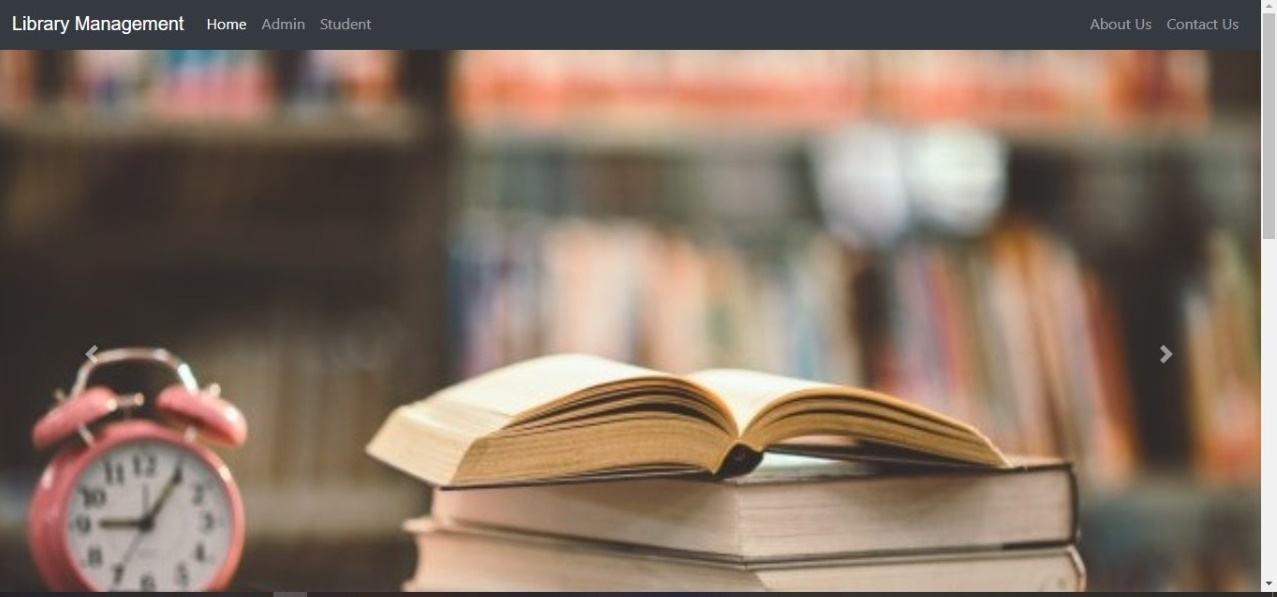
# *-student login*



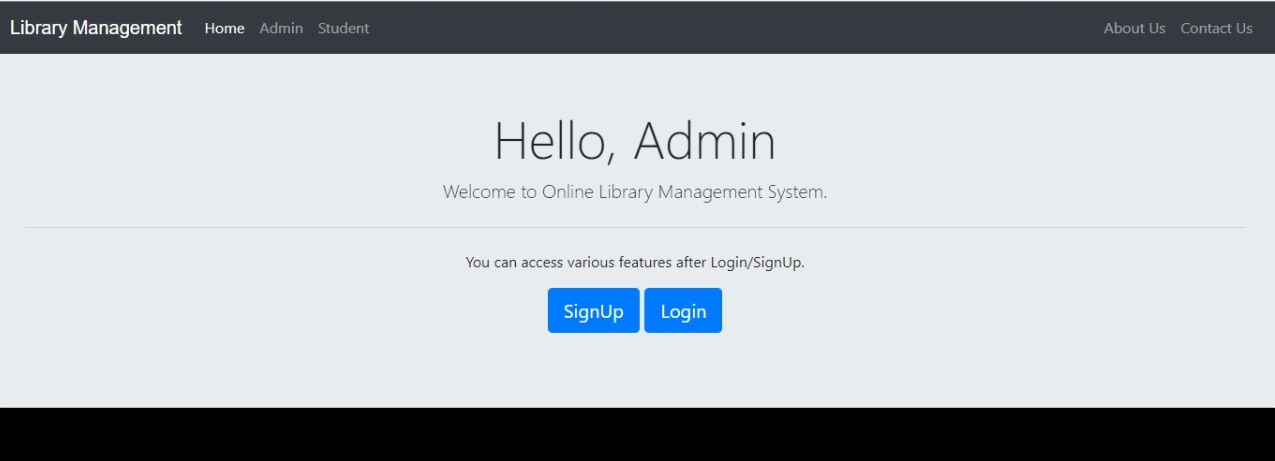
# -*Admin login*

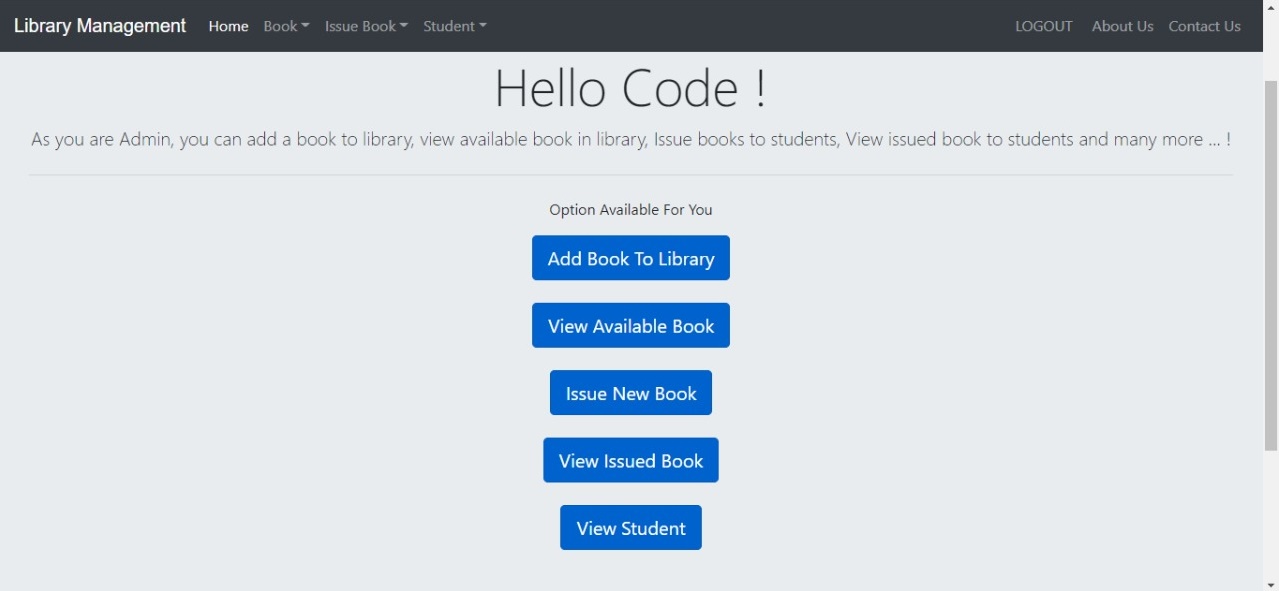
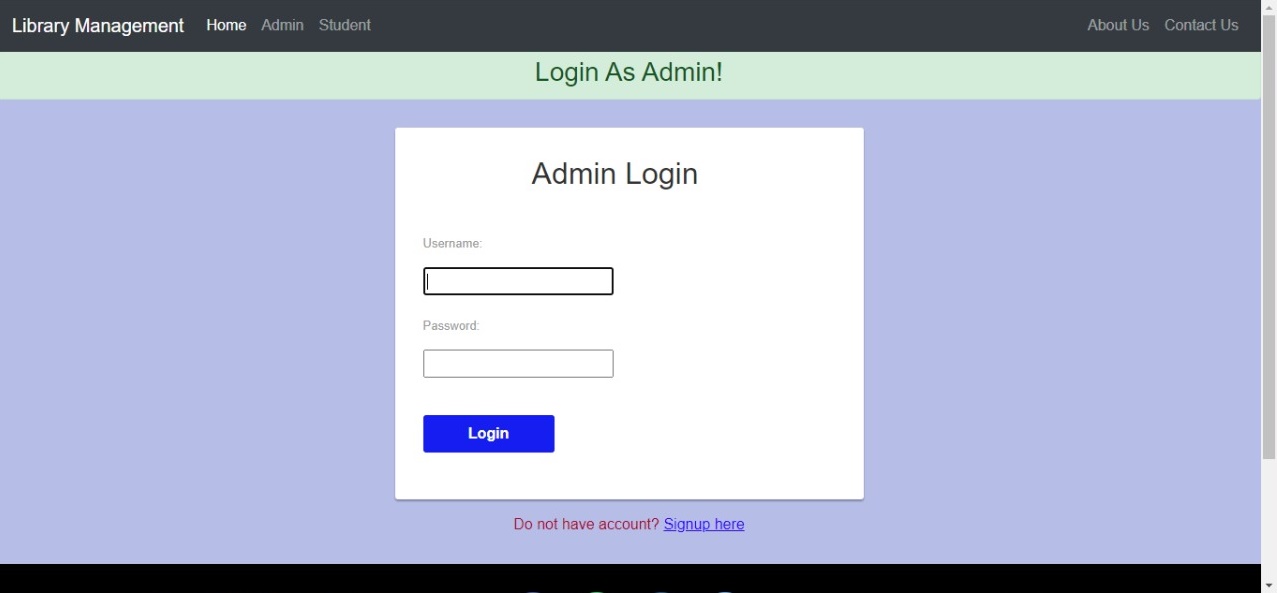


# -Home page

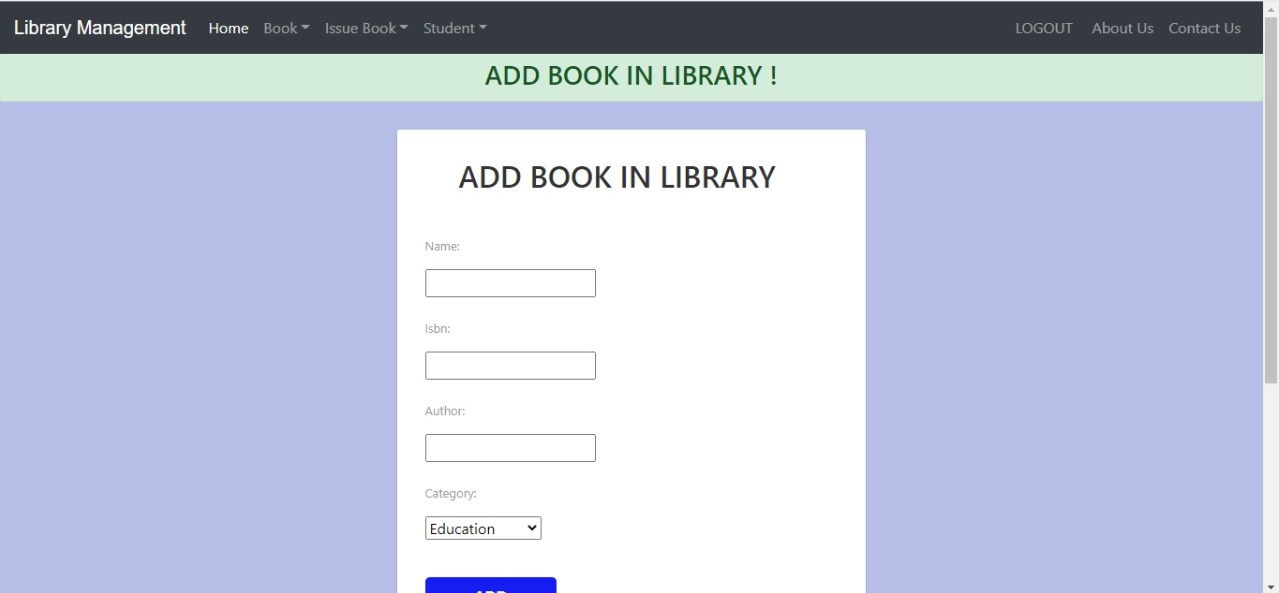


# -Admin pages

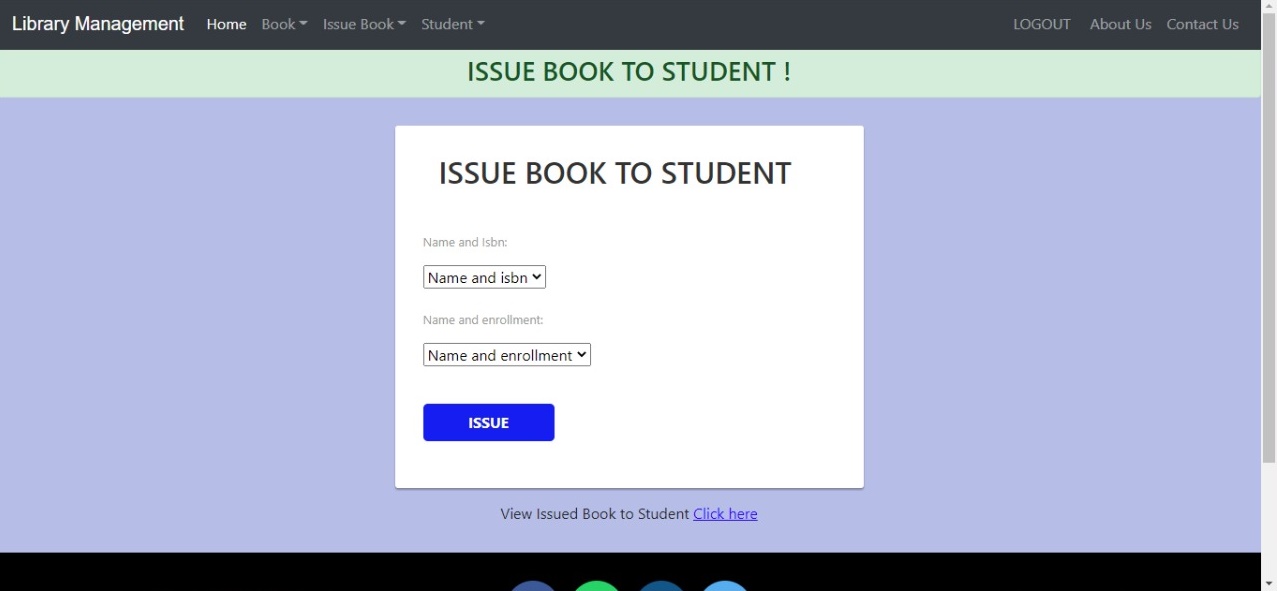




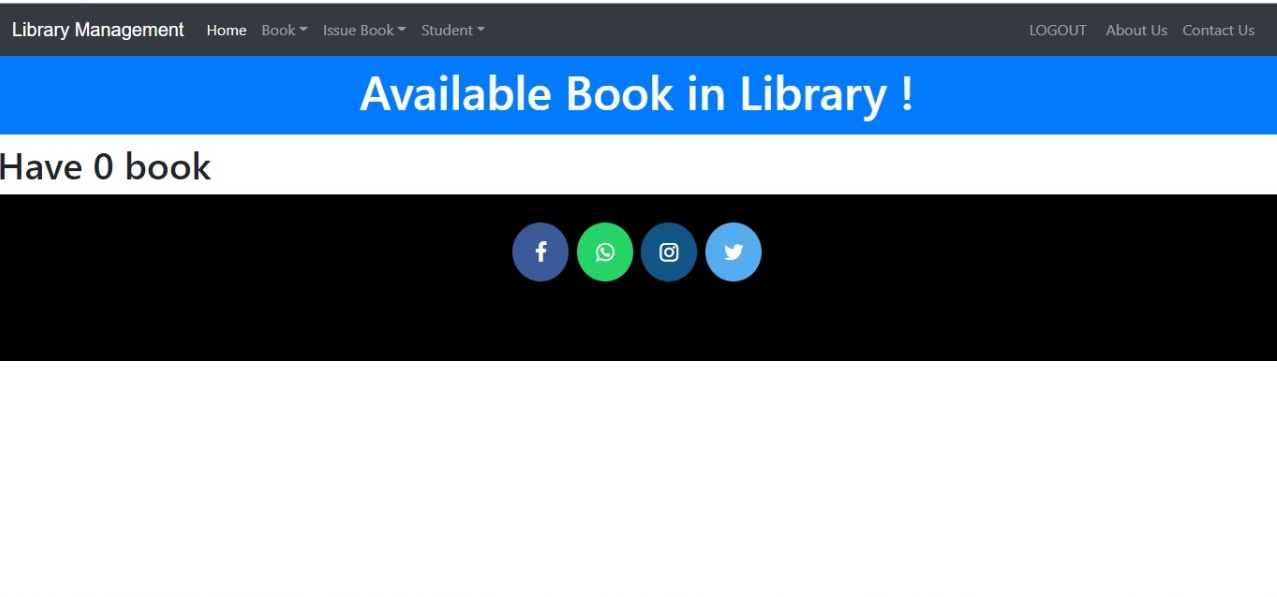
# - ADD BOOK



# -ISSU BOOK



# -AVAILABLE BOOKS



# *System Feature*

# Requirements:

# -Functional Requirements

**I. user requirements**

-user can register with his personal details.

-user can log in system after his register

-user can select his category from(student ,education staff member, employee)

- User searches for certain book.

-user select the book that he searched about.

-user can borrow a book.

-user can give his rating for his books.

-user can request book in waiting list.

-admin receive user’s order .

-admin check user’s order .

-admin can add a book.

-admin can delete a book.

- admin can update book’s details.

-admin can review book.

-admin can check waiting list.

**II. System requirements**

- The system should record following user details into member  
database: Name, Email, Password, Address.

- The system shall send verification message to email.

- The system should verify the user email & password against the  
member database when logging in.

- After login, user should be directed to Home screen.

- The system should enable User to check for availability of a book by search for it.

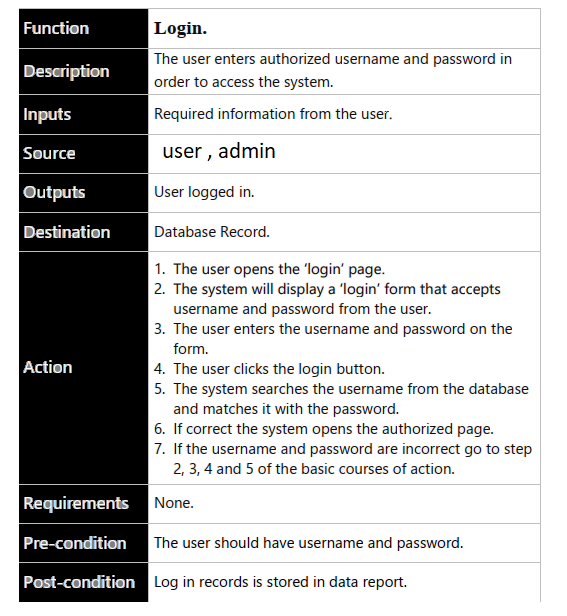
- The system should enable u­ser to issue a book.  
- the system should display all information about books.

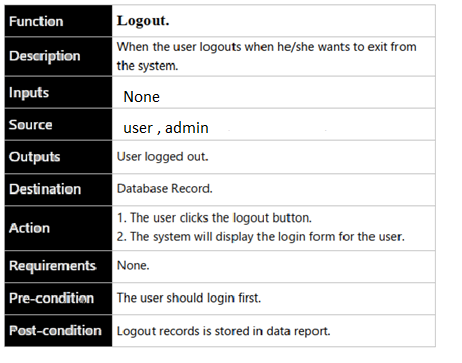
-the system enable user to give his rating for books.

-the system send notifications to user for his order in waiting list .

-the system should display recommendation list for user that logged in system.

# Use Case

******

******

|  |  |
| --- | --- |
| Function | update a Book |
| Description | The Admin update a book . |
| Inputs | -book name |
| Source | Admin. |
| Outputs | The book has been updated, And the database successfully updated. |
| Destination | Database Record. |
| Action | 1. The admin opens the update a book page. 2. The user clicks ‘update’ button. 3. The system will display a update form that contain the following: (Book name, author name…. ). 4. admin fills the form. 5. Admin click update button. 6. The system checks all the form fields have filled out. 7. If the form filled correctly the system display successfully updated message. 8. If the form is not filled correctly go back to step 3, 4, 5 and 6 of basic course of action. |
| Requirements | None. |
| Pre-condition | There should be a book to be updated. |
| Post-condition | updated record is updated in database. |

## Use case diagram

## D:\fourth term\R\Capture.PNG

## data flow Diagram

# DFD level 0.JPG

# Sequence diagram

# Login-

# Capture.JPG

# Register:

# SequenceRegister.PNG

# Search for Book

# search.png

# Borrow a book

# Issue a Book.png

# Return a book

# 

# ADD a Book

# 

# Delete

# 

# Class Diagram

# state diagram:

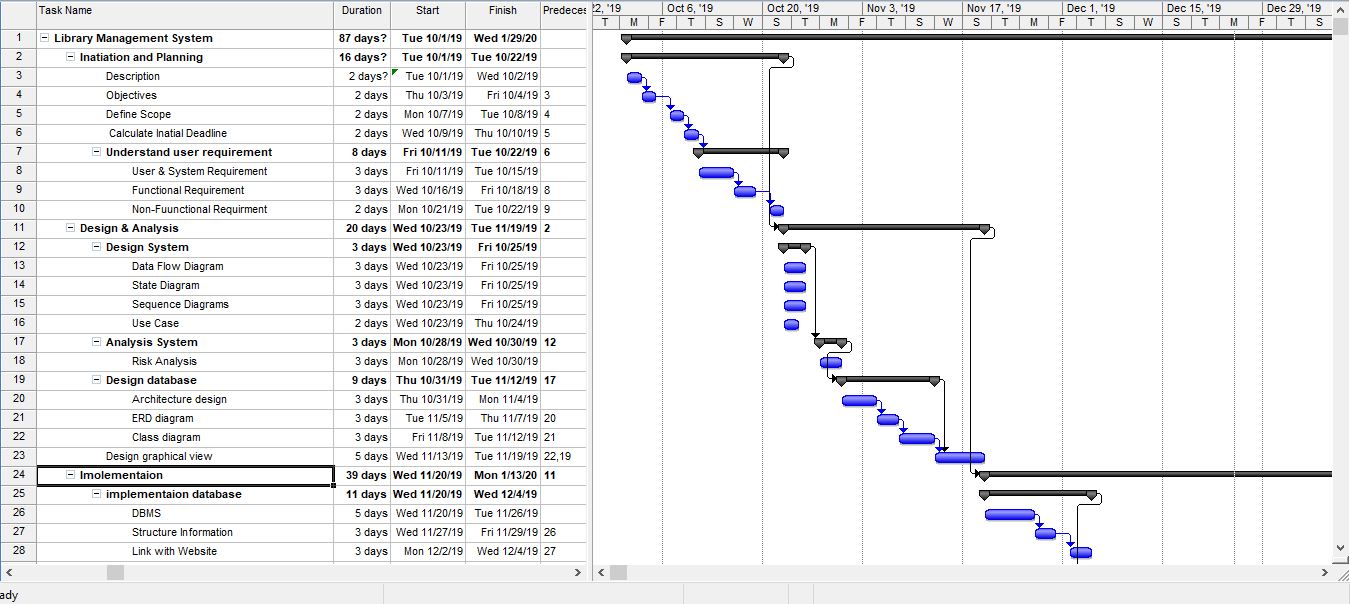
# F:\Capture33333333.JPG

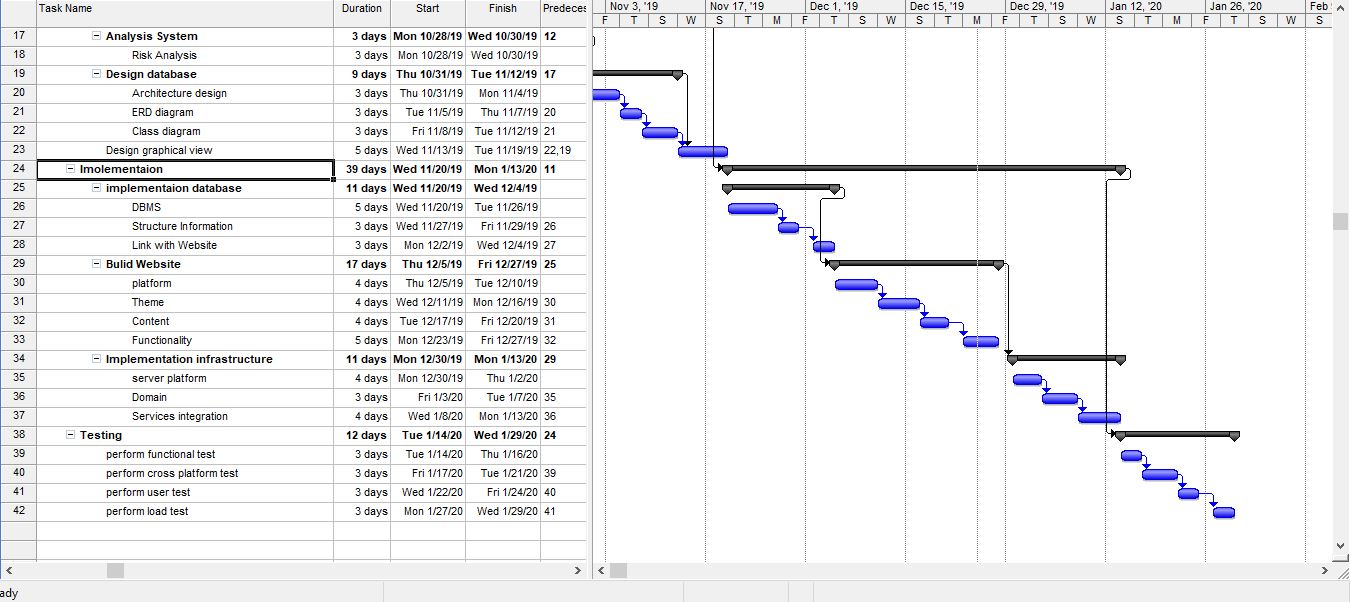
# F:\Captur11111111111111.JPG

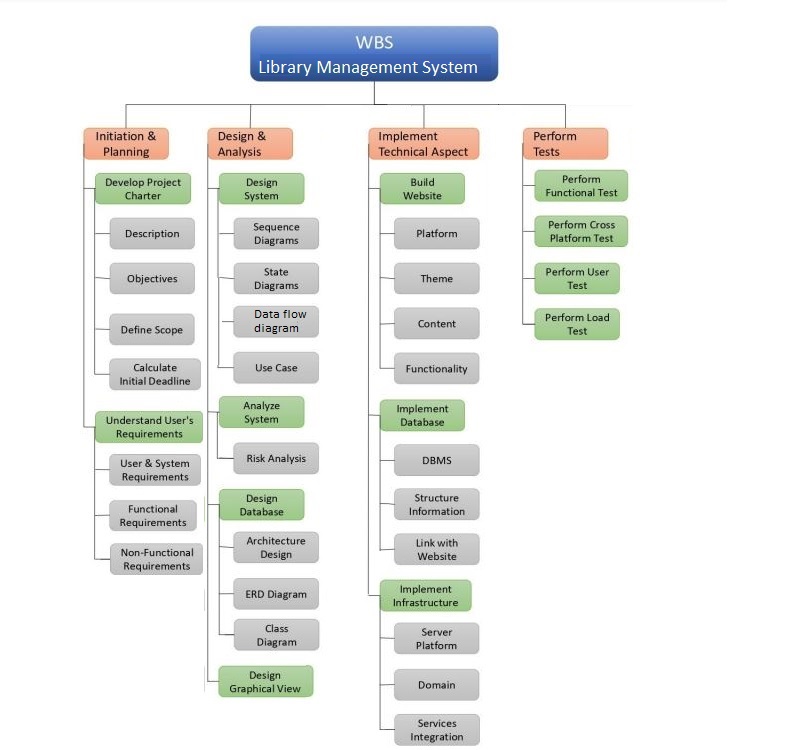
# F:\Capture2222222222.JPG

# Work plan

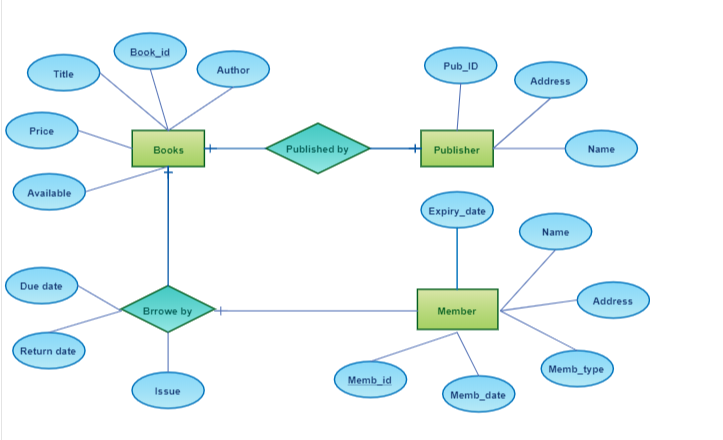
**Gantt Chart**

****



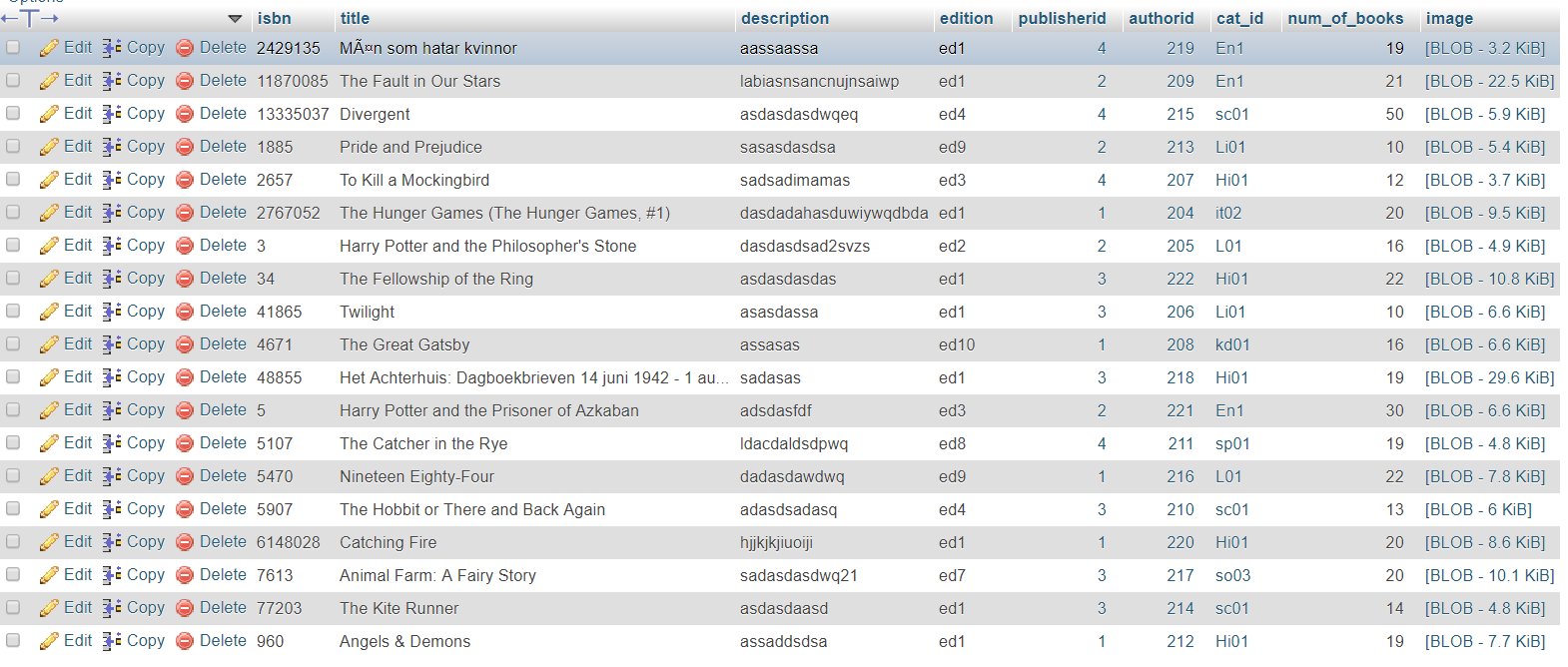
WBS

# Erd diargram:

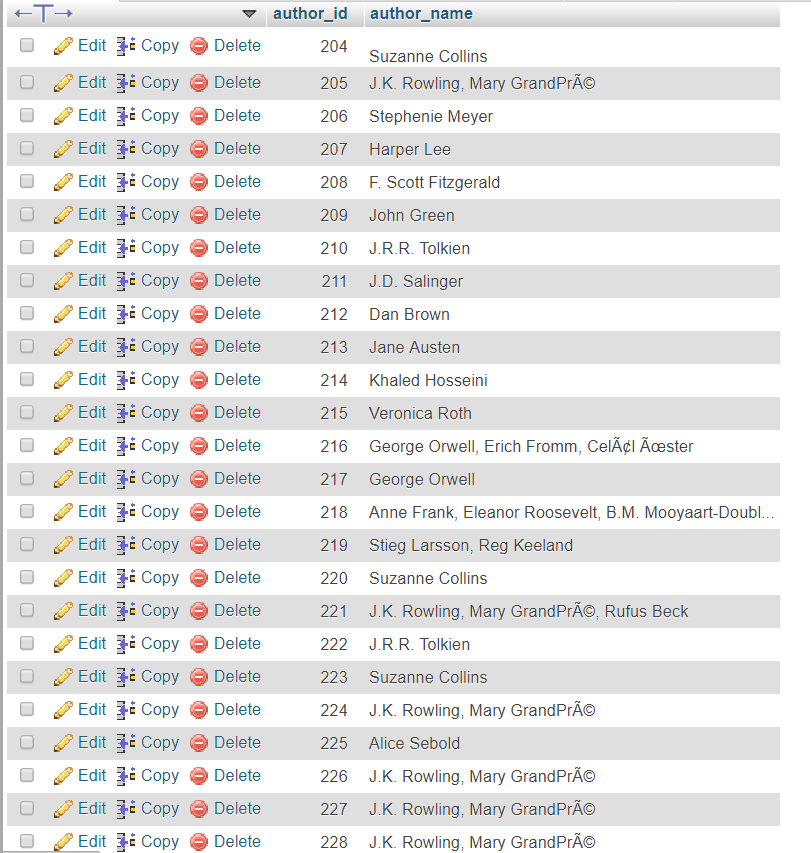
****

# Database Tables:

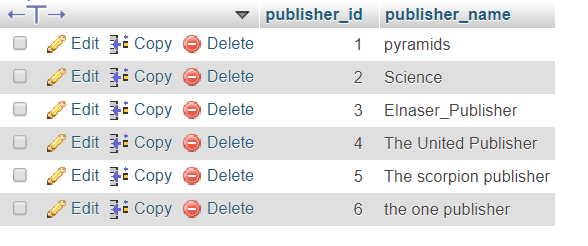
# book:

****

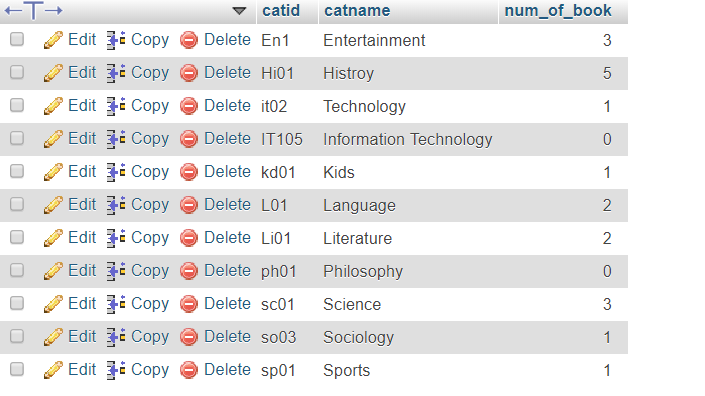
Author

****

**\***publisher

****

**\***category**:**

****

Other Nonfunctional Requirements:

**Performance Requirements:**

**\*Scalability**

•The system must upgrade when some updates will be available.   
•The system must operate correctly when the number of recorded information is being increased.

**Efficiency**

•The system gives appropriate output based the list of inputs.  
•The system operates in the shortest time with the least amount of  
resources.

**Safety Requirements**

To ensure that no one of Website’s users loses any data while using our website (due to a  
crash or a bug of some kind) the developer team updates Website regularly. There is a  
bug tracker available where users can report any bugs they have encountered so that  
the developers can fix it in the next release.

**Security:**

•The authenticate users should have a privilege to access the database.  
•The system does not allow unauthorized users to login.

**Software Quality Attributes:**

**Availability:**•The system must operate 99% at working time**.**•We must make sure that the system be accessible for all authorized staff and not accessibility by anyone else.  
**Flexibility:**•The system must support commonly usable browsers.  
  
**Speed:**

•The system must have quick response time.  
•The system should respond users request within maximum of 2 seconds interval.  
•The system must be simple to retrieve comprehensive information.

**Usability:**•The system should be easy to use by Staff.  
•And our system gives direct input on how real users use the system.  
**Portability:**•The system must be designed for plate form independent.  
•The system supports every operating system.  
**Reliability:**•The system that is to be developed is real not an ideal.  
•The system does not allow unauthorized users to login.

**Maintainability:**

•Software must be developed to meet changes

•Accept the new requirement.  
•Allow changing the format.