***A Project report***

**On**

***Online library Management System***

Submitted in partial fulfillment of the requirements of the award of the   
degree of Bachelor of Computer Application

****

Guide Submitted by

Dr. Pramod Joshi Kamaljeet Singh Rana

170610330054

Submitted to

Faculty of Computer Science & Applications

Amrapali Institute of Applied Sciences, Haldwani

DECLARATION

I hereby declare that the project work entitled “Online Library Management system” is being submitted to Faculty of Computer Science and Application is the authentic record of my own project work done under the mentor Dr. Pramod Joshi FCSA, Amrapali Group of Institutes, Haldwani .

Kamaljeet Singh Rana

170610330054

Amrapali Group of Institutes, Haldwani .

CERTIFICATE

This is to certify that this project entitled ” Online Library Management system” Submitted in partial fulfillment of the degree of Bachelor of Computer Application to the Dr. Pramod Joshi, Associate Professor, Amrapali group of institute, done by Kamaljeet Singh Rana, Roll No. 170610330054 is an authentic work carried out by her at Amrapali group of institute under my mentorship. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

CERTIFICATE

This is to certify that the project titled “Online library management system” is the bonafide work carried out by kamaljeet singh rana student of Bachelor of Computer Applications, Amrapali Institute of Applied Sciences, Haldwani affiliated to Kumaun University, Nainital, India, during the academic year 2019-20, and that the project has not formed on the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Signature of the Guide**

ACKNOWLEDGEMENT

I take the opportunity to express my gratitude and thanks to all the people who directly or indirectly helped and guided me throughout the project. It is my privilege to express my sincerest regards to my mentor Dr. Pramod Joshi Sir for their valuable inputs, able guidance, encouragement, whole-hearted co-operation and constructive criticism throughout the duration of my project. I take this opportunity to thank all our lecturers who have directly or indirectly helped my project.

CONTENTS

|  |
| --- |
| Title Page |
| Declaration of the Student |
| Certificate of the Guide  1 Introduction |

1.1Overview

2 The overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communication Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirement

2.2 Product Function

2.3 User Characteristics

2.4 Constraints

3 Specific Requirements

3.1 External Interfaces

3.2 Functions

3.3 Performance Requirement

3.4 Logical database requirement

3.5 Software system attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.5 Portability

4 Requirement analysis

4.1 Study of old model

4.2 Software tool used

4.3 Cost and benefit analysis

4.4 Requirement Gathering

5 Design

5.1 Data flow diagram

5.2 ER diagram

5.3 Use Case diagram

6 System implementation

6.1 Input Output Screen

6.2 Process Involved

6.3 Module description

7 System Testing

7.1 Unit testing

7.2 Integrated testing

8. Project legacy

8.1 Future Scope

8.2 References

INTRODUCTION

In any organization, it's too hard to maintain library since they have students from 1st standard up to the graduation or post-graduation level, if students wants to take any particular books from library, it's very hard to search manually. To solve this, Online Library Management system has taken into role, developed and designed that allows access to the library wherein the user can view availability of the books and also keep a check on the stock. The automated library system has been created with the intention of making libraries more organized and to make library stocking and accessing more simplified and structured. It uses technology in terms of check-in and check-out, fine management and the status of a book. It has become an essential tool for large libraries which have a huge number of books since stock checking and procurement is a lot simpler using the various technological options and a structured approach.

The library software is one of the most effective ways of organizing a library and using technological options to make management of resources more efficient. The library software can be used by both big and small colleges and schools for a better management of their books and more importantly for decreasing manual labor and time.

The system utilizes user authentication, displaying only information necessary for an individual’s duties. Additionally, each sub-system has authentication to allow authorized users to create or update information in that subsystem. All data is stored securely on SQL servers managed by the college administrator and ensures highest possible level of security.

1.1purpose:

1. The online library management system is developed for managing the library online.

2. This portal will be most useful for librarian as well as for library users for viewing different records of library system.

3. Student login page where student can find books issued by him/her and date of return

4. A search column to search availability of books

5. The data once entered will be able to edited/deleted as per required when there will be Vast entries of data.

Scope

* The system “Online library management system” can be used by organization for better running experience for library.
* There are no banners for advertisement on the site.
* This system would be able to run only in Windows platform.
* Supported only by My SQL database.
* There will be no security of data.
* Supportive language is only English

1.2 Definition, Acronyms and Abbreviations:

ADMIN: Administrator.

RAM: Random access memory.

ROM: Read only memory.

GUI: Graphics user interface.

GB: Giga Byte.

SRS: Software Requirement and Specification.

DFD: Data Flow Diagram.

E-RD: Entity Relationship Diagram.

E-Mail: Electronic Mail.

OS: Operating System.

DBMS: Database Management System.

PHP: Hypertext preprocessor.

SQL: Structured Query Language.

WWW: World Wide Web.

HTML: Hyper Text Markup Language.

CSS: Cascade Style Sheet.

JS: Java Script.

CI: Code Igniter

HDD: Hard Disk Drive.

DD: Data Dictionary.

OVERVIEW

Online library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login .It also has a facility of admin login through which the admin can monitor the whole system .. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can see the available books in library. Managing fine on student can be done with ease. The system can be back up with reduces the risk of losing library information.

2. THE OVERALL DESCRIPTION

The library plays a vital role in student life so providing the information of library through online medium can help student to find books easily.

The online library management system can be used in different organization, for management of books, magazines and journals. The online library management system has been designed to make it simpler to manage libraries, saving manual labor and time. The online library management system helps in recording issue and return, calculation of fines and procurement of books. The online library management system helps greatly in reducing the time taken record books manually.

2.1 Product Perspective

The software is a self-contained and an independent product with proper user interface.

The application uses the centralized databases which contains all the information.

2.1.1 System Interface

Apache will be used as web server. The user inputs data via the web server

Using HTML forms. The actual program that will perform the operations is

written in PHP.

2.1.2 Interfaces

The new system shall provide a very intuitive and simple interface to the user and the administrator, so that the user can easily navigate through pages, issued books, adding new author, status of users.

.

2.1.3 Hardware Interfaces

a) Server side the web application will be hosted on a web server which is

listening on the web standard port, port 80.

b) Client side Monitor screen – the software shall display information to

the user via the monitor screen Mouse – the software shall interact

with the movement of the mouse and the mouse buttons. The mouse

shall activate areas for data input, command buttons and select options

from menus. Keyboard – the software shall interact with the keystrokes

of the keyboard. The keyboard will input data into the active area of the

database

(1) Processor: Intel Core I 3  
 (2) RAM: 2 GB   
 (3) Hard Disc: 20 GB

2.1.4 Software Interfaces

Server side- An Apache web server will accept all requests from the client

and forward it. Accordingly a database will be hosted centrally using

My SQL.

Client side- An OS which is capable of running a modern web browser

which supports JavaScript and HTML5.

* + 1. Front End Tool: PHP
    2. Back End Tool: My SQL
    3. Development Tools: XAMPP Server
    4. Browser: Any browser
    5. DocumentationTool:MicrosoftOffice2010

2.1.5 Communication Interfaces

The HTTP or HTTPS protocol(s) will be used to facilitate communication

between the Client & Server.

2.1.6 Memory Constraints

Memory constraints will come into play when the size of My SQL grows

to a considerable size**.**

2.1.7 Operations

The product shall have operations to protect the database from being

corrupted or accidentally altered during a system failure.

2.1.8 Site Adaptation Requirements

The component will be adapted to the overarching system at the conclusion

of the system creation.

2.2 Product Function

**Admin:**

* On visiting the online library management system the admin logins through admin login module.
* On successfully login into system the admin is displayed dashboard where information is displayed like total users, total books, and all time book issued.

Some links are also displayed change password, profile and menu button.

* Change pass word will take user to change password screen.
* Profile will take user to profile screen.
* On clicking side menu button different operation can be accessed like manage author, manage book, add new book, add new author, issue new book, manage users, view issued book, fine, manage issued book.
* Manage author will take user to manage author screen.
* Manage book will take user to manage book screen.
* Add new book will take user to add new book screen.
* Add new author will take user to add new author screen.
* Issue new book will take user to issue new book screen.
* Manage user will take user to manage user screen.
* View issued book will display the issued books.
* Fine will show the fine and return and add fine.
* Manage issued book will take user to manage issued book screen.
* From, any screen user can switch to home screen and log out.

**Student:**

* On visiting the online library management system, a Login screen is displayed. If the user is a new user, he/she will click the sign-up button on the screen; otherwise will Login with the specified details.
* On successfully filling all information correctly the user is registered with system. A user can exit any time during login/sign-up.
* Now, user has logged in and a dashboard is displayed with some information of total book issued, book not returned yet, fine, menu button, profile, setting.
* Profile take the users to user profile screen and user can see the user information and can edit user detail.
* Setting takes the user to change password screen.
* On clicking on side menu button different operation can be accessed like books issued, available book.
* Issued book takes the user to issued book screen
* Available book takes the user to available book screen
* Dashboard and logout operation can be accessed from any screen

2.3 User Characteristics

* Education level: At least user should be comfortable with

English language.

* ­­­­­­­­­­­­­­­­­­Technical expertise: Should be comfortable using

general-purpose applications on a computer.

2.4 Constraints:

* Sign-in and password are used for the identification of user.

1. SPECIFIC REQUIREMENTS

To overcome the disadvantage of current library system we are going to purpose an application name as “Online library management system”. Some of the features that are included into our system are:

1. Availability of books can be viewed by the user.
2. Display book detail.
3. Display author detail.
4. Display issued book.
5. Issuing new book.
6. Only authenticate person can enter into the system.
7. Managing the users.
8. Managing the book.
9. Managing fine.
10. Book transaction record

3.1 External Interface:

The following screen will be provided:

**Login Screen:**

This module is for the log in of the existing user. This will be first screen that will be displayed. It will allow user to access different screens based upon the user’s role. Various fields available on this screen will be

* User ID
* Password
* Role : user

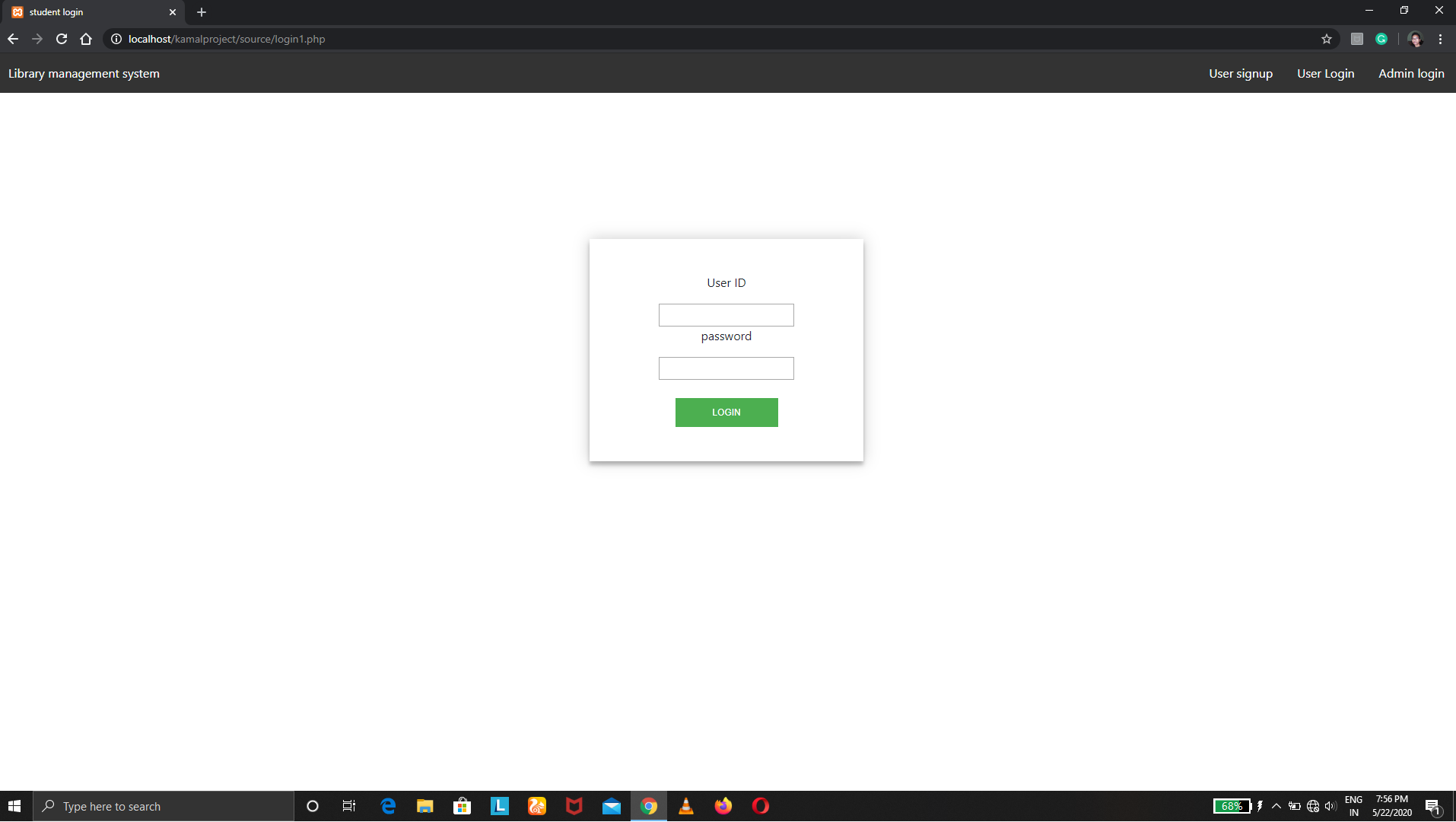
****

Figure 1: Login page

**Admin login Screen:**

This module is for the admin. This screen will be displayed when an admin wants to login system. It will allow different screen to admin. Various fields available in this screen will be:

* Admin ID
* Password
* Confirm password
* Role: admin

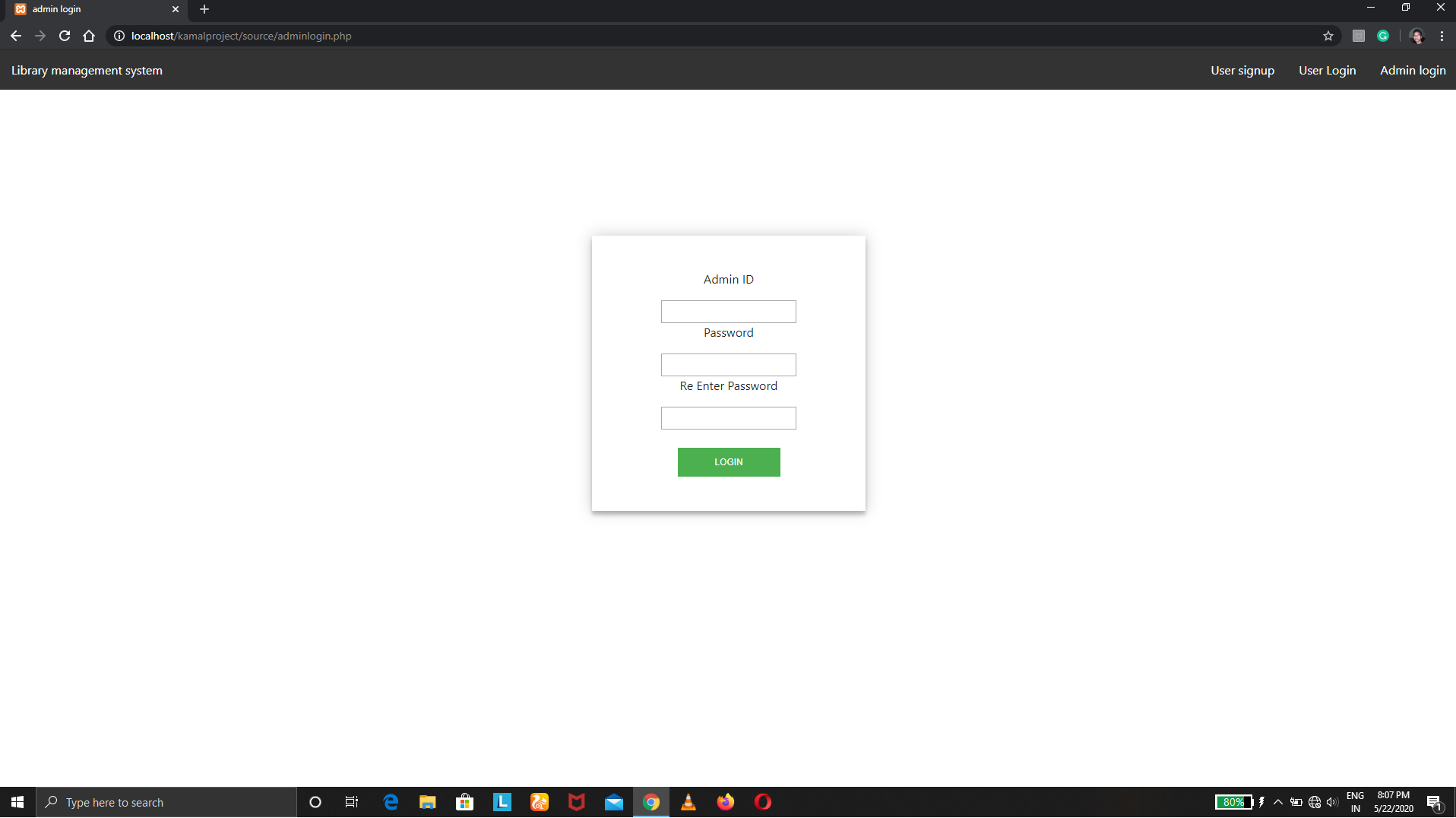


Figure 2: Admin Login page

**Sign-up screen:**

This module is for the signing up for the new users. Its fields are:

* Name
* Email
* User id
* Password
* Confirm password

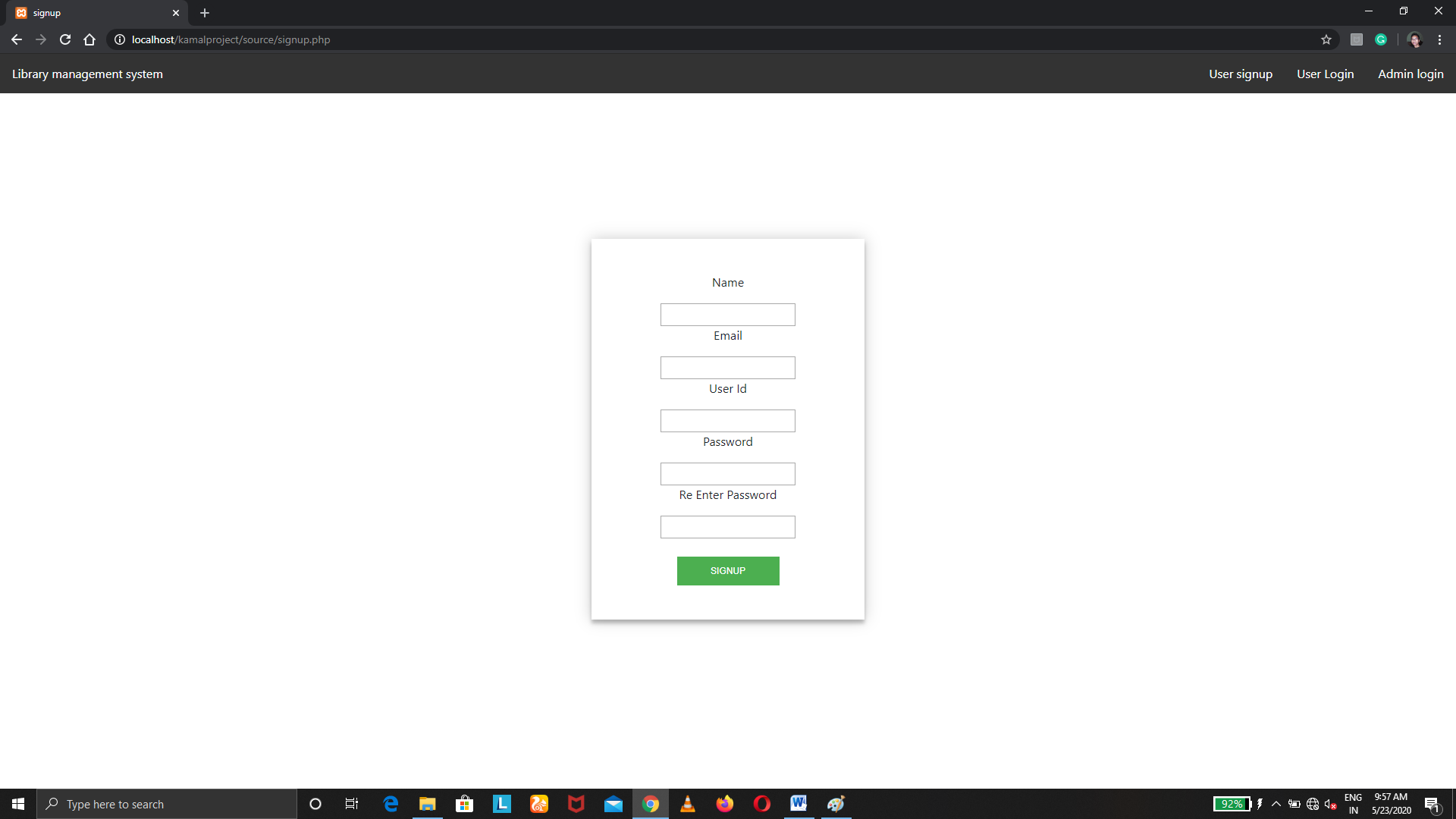


Figure 3: signup page

* 1. Functions:
  2. Performance requirement

Basically there are four types of users who deal with the system. The users are listed below

* Admin
* User
  1. Logical Database Requirement

The following information will be placed in a database:

1. **User info:** user name, roll number, email, user id, etc**.**
2. **Admin info:** admin name, admin id, admin email, etc.
3. **Book info:** book name, book id etc.
4. **Fine info:** fine, fine id.
5. **Author info:** author name, author id.
   1. Software System Attributes
      1. Availability

The “online library management system” software will be available for user only by their login details. During this time, it shall be operational for as long as is possible. Administrators will have 24-hour access to the system but admin also must have to login to the system.

3.5.2 Reliability

The entire system should be available 99.9%. Redundancy in clustered hardware will be used to obtain web servers availability.

The system shall not leave any cookies on the client’s computer containing the user’s password. The system shall not leave any cookies on the client’s computer containing any of the user’s confidential information.

Data Storage The client’s web browser shall never display a client’s password. It shall always be echoed with special characters representing typed characters. The client’s web browser shall never display a client’s password after retrieving from the database. It shall always be shown with Boolean values true or false. The software system’s back-end servers shall never display a client’s password. The client’s password may be reset but never shown. The system’s back-end servers shall only be accessible to authenticated administrators. The system’s back-end databases shall be encrypted.

Maintainability System administrators will be given an administrative interface to manage books, transaction of books and users. Additionally, user interfaces will allow the configuration of news processing, and the application “look and feel”. Hardware maintenance on the server infrastructure will be maintained by the developer.

3.5.3 Security

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.

2. The user's IP will be logged. .

3.5.4 Maintainability

My SQL is used for maintaining the database and the Apache server takes care of the site

In case of a failure, a re-initialization of the program is recommended.

3.5.5 Portability

The application is Windows-based and should be compatible with other systems.

Apache PHP and My SQL programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web.

Objectives

The objective of our application is to provide an effective and efficient way which enables student and other user to access the library information online. Admin manages the different aspects of library like issuing book returning book adding new book managing old books managing the information of old books, adding new author managing old authors managing library user etc. The primary objective of application is to provide effective and smart system to organizations and library users to view issued books and available books to the users.

Scope

The system “Online library management system” can be used in any organization like School, Colleges who wants to make their library system online and their students and users can access the library online. This system makes the library system smart and reduces the labor work. By considering these entire one can make a smart organization.

1. There are no banners for advertisement on the site.

2. This system would be able to run only in windows platform.

3. Supported only by My SQL database.

4. There will be no security of data.

5. Supportive language is only English

HARDWARE CONFIGURATION

This system is developed on the following hardware configuration

1. Intel i5 8th generation processor
2. 8GB RAM
3. Hard disk 1 TB
4. Microsoft compatible key board
5. Scroll mouse

DESCRIOTION OF SOFTWARE

This system is developed using the following software

1. Operating system: Windows 10
2. Technology: PHP, HTML, JAVASCRIPT
3. Development Tools: XAMPP Server
4. Back End :My SQL

Overview of the PHP Scripting language

The **PHP Hypertext Preprocessor (PHP)** is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

Overview of Html

**HTML** stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text.

**Overview of java script**

**JavaScript** is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

**METHODOLOGY ADOPTED**

1. **REQUIREMENT ANALYSIS:**
   1. **Study of the existing system**

**In order to understand the requirement of system we can take example of previous systems which are running on different organization. All those organizations has a system which require lots of labor work all the transaction details are written in registers which are hard to maintain and take backup. The user student cannot view the issued books and finding books by users is also difficult. Managing books, managing transaction, managing library members is also not an easy task for the librarian. Taking membership in old system takes so much time.**

* 1. Analysis of problem and weakness of existing system

After study of existing system we found following problem and weaknesses in the system. Following are the problem and weaknesses of the current system:

1. Users cannot easily find availability of books
2. Manually work
3. Checking the list of issued books is not easy
4. Book management is not easy

1. REQUIREMENT GATHERING:

Gather requirements of new system

To overcome the disadvantage of current library system we are going to propose an application name as “online library management system”. Some of the features into the systems are:

* Available books can be viewed by users.
* Display issued of the books.
* Only authenticate person can enter into the system.
* Add new books
* Add new author
* Managing author.
* Managing book.
* Manage transaction of books

Hardware and Software to be used

Minimum Hardware Requirement

1. Intel dual core / amd processor
2. RAM 2GB
3. Hard Disc 30GB

Minimum Software Requirement

1. Front End Tool: PHP
2. Back End Tool: MY SQL
3. Development Tool: Xampp Server
4. Browser: Any browser
5. Documentation Tool: Microsoft Word 2010
6. Design Tool: CSS

Maintenance

MySQL is used for maintain the database and Apache Server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

Evaluation

The software manages everything including maintenance, data, security, updates, etc.., and no to hire exclusive staff for the above functions.

COST AND BENEFIT ANALYSIS

It is a cost-effective solution with cheaper software licenses which are completely subscription based.

Data flow diagram

0 LEVEL DFD

View issued book

User

View books

Admin

1 LEVEL DFD

Database

Database

Manage author

Database

v

v

Database

v

v

Transaction of book

View available book

Manage users

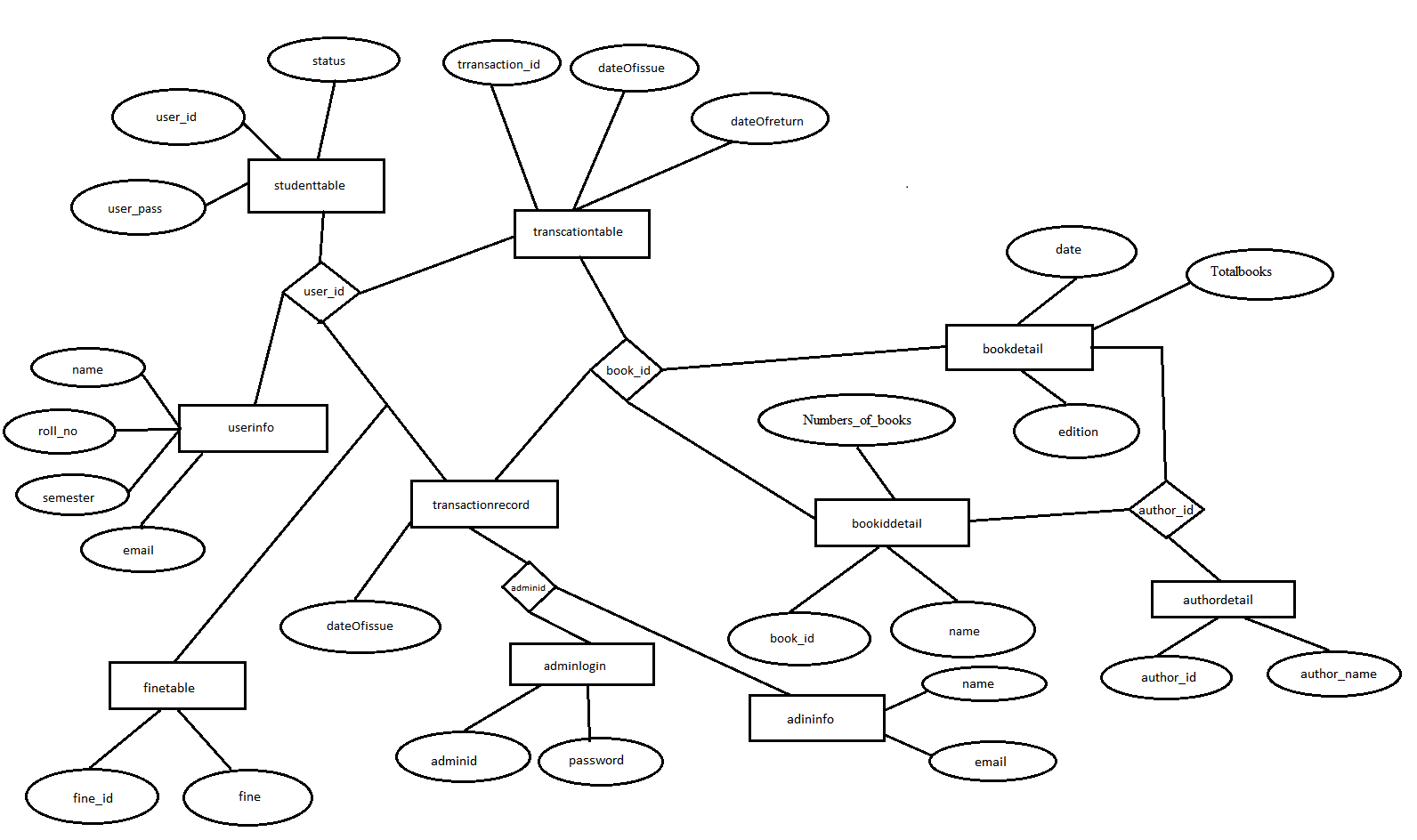
View Issued book

Manage

Books

Admin

User



DATA DICTIONARY-

Userinfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | roll\_no | Int | 11 | None | Auto increment |
| 2 | User\_id | Varchar | 40 | None |  |
| 3 | Name | Varchar | 40 | None |  |
| 4 | Course | Varchar | 40 | None |  |
| 5 | Semester | Int | 11 | None |  |
| 6 | Email | Varchar | 40 | None |  |
| 7 | Status | Int | 11 | Yes |  |

Admininfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | s.no | Int | 11 | None | Auto increment |
| 2 | Adminid | Varchar | 40 | None |  |
| 3 | Name | Varchar | 40 | None |  |
| 4 | Email | Varchar | 40 | None |  |

Adminlogin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | Adminid | Varchar | 40 | None |  |
| 2 | Password | Varchar | 40 | None |  |

Studenttable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | User\_id | Varchar | 40 | None |  |
| 2 | User\_pass | Varchar | 40 | None |  |
| 3 | Status | Int | 11 | Yes |  |

Bookiddetail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | Book\_id | Int | 10 | None |  |
| 2 | Name | Varchar | 40 | None |  |
| 3 | Author\_id | Int | 11 | None |  |
| 4 | totalbooks | Int | 10 | Yes |  |

Bookdetail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | s.no | Int | 11 | None | Auto increment |
| 2 | Book\_id | Int | 10 | None |  |
| 3 | Author\_id | Int | 11 |  |  |
| 4 | Date | Date |  | current\_timestamp() |  |
| 5 | Edition | Varchar | 40 | None |  |
| 6 | Numbers\_of\_books | Int | 10 | Yes |  |

Finetable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | Fine\_id | Int | 11 | None | Auto increment |
| 2 | User\_id | Varchar | 40 | None |  |
| 3 | fine | int | 11 | None |  |

Authordetail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | Author\_id | Int | 11 | None | Auto increment |
| 2 | Author\_name | Varchar | 40 | None |  |

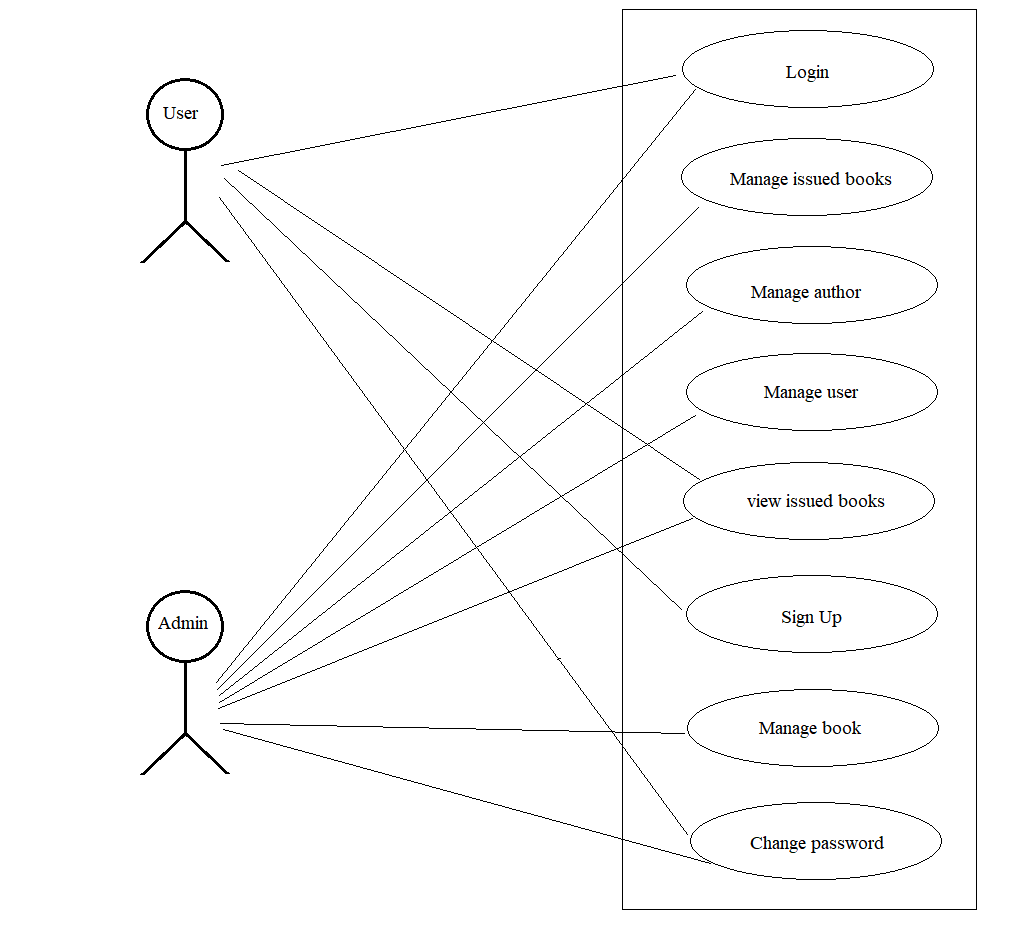
Transactionrecord

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | transactionRecordId | Int | 11 | None | Auto increment |
| 2 | User\_id | Varchar | 40 | None |  |
| 3 | Book\_id | Int | 11 | None |  |
| 4 | dateOfIssue | Date |  | None |  |
| 5 | adminId | Varchar | 40 | None |  |

Transactiontable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Data Type** | **Field Size** | **Default** | **Extra** |
| 1 | Transaction\_id | Int | 11 | None | Auto increment |
| 2 | User\_id | Varchar | 40 | None |  |
| 3 | Book\_id | Int | 11 | None |  |
| 4 | dateOfIssue | Date |  | current\_timestamp() |  |
| 5 | dateOfReturn | Date |  | None |  |

**USE CASE DIAGRAM FOR ADMIN AND USER**



1. Login

**Actor:** The following actors participate in this use case :

Admin, Users.

**Basic flow:** This case starts when the actor Login to the Online management System.

1. The system ask the actor to enter his/her user id and password.
2. The actor enter user id and password.
3. The system checks the user id and password and logs the actor into the system

**Alternative Flow:**

* If invalid name and password

If actor enters an invalid name, password then the system shows invalid message and the basic flow cancel the login. The actor returns to login page

**Pre – Conditions:** All users must have a user account. Account can be created using signup module, prior to executing the use case.

**Post – Condition:** If the use case was successful, the actor is logged into the system. If not, the system state is not changed.

1. Manage issued books

This use case allows actor with role as admin or librarian to maintain the issued books.

This includes returning the book.

**Actor:** The following actor will participate in this use case:

Admin

**Basic Flow:** This use case starts when admin wants to manage issued books and wants perform some actions like return a book.

1. Login
2. Once the admin gets the requested information, the following sub flow is executed.
   * Return the issued book
   * Check the fine on student

**Alternative Flow:**

1. No record found

**Pre – Condition:** The admin must be logged onto system before use case begins.

**Post– Condition:** If the use case is successful, the book is returned to the system. Otherwise the system is unchanged

**3.** ManageBook

This use case allow actor with role as admin or librarian to manage book. This include add new book update details of book delete book.

**Actor:** The following actor will participate in this use case:

Admin

**Basic Flow:** This use case starts when admin wants to add, update, delete book details from the system.

1. Login
2. Once the admin gets requested information, one of the following is executed

* Add new book detail ,book author, book name, book edition
* Update book detail, book author, book name, edition
* Delete the book

**Alternate Flow:**

1. Book already exists
2. Update cancelled
3. Delete cancelled

**Pre – Condition:** The admin must be logged onto system before use case begins.

**Post– Condition:** If the use case is successful, the book is added, updated deleted to the system. Otherwise the system is unchanged

**4.** Manage user

This use case allow actor with role as admin or librarian to manage users of the system. This include changing the password of the and changing their status as active or inactive.

**Actor:** The following actor will participate in this use case:

Admin

**Basic Flow:** This use case starts when admin wants to change, invoke user authority from the system.

1. Login
2. Once the admin gets requested information, one of the following is executed

* Set user status as active or inactive
* Change password of users

**Alternate Flow:**

1. User not found
2. Update password cancelled

**Pre – Condition:** The admin must be logged onto system before use case begins.

**Post– Condition:** If the use case is successful, the password changed, set status to the system. Otherwise the system is unchanged

**5.** View issued books

This use case allow actor with role as admin and user to view the issued book from the system according to the use.

**Actor:** The following actor will participate in this use case:

Admin, user

**Basic Flow:** This use case starts when admin or users wants to view issued from the system.

1. Login
2. View the information of issued book

**Alternate Flow:**

1. No record found

**Pre – Condition:** The admin or user must be logged onto system before use case begins.

**Post– Condition:** If the use case is successful, the issued books are shown. Otherwise the system is unchanged

**6.** Manage Author

This use case allow actor with role as admin to manage the author from the system.

**Actor:** The following actor will participate in this use case:

Admin

**Basic Flow:** This use case starts when admin or users wants to view issued from the system.

1. Login
2. Once the admin gets requested information, one of the following is executed

* Add new author
* Update author detail
* Delete author

**Alternate Flow:**

1. Author already exists
2. Update cancelled
3. Delete cancelled

**Pre – Condition:** The admin must be logged onto system before use case begins.

**Post– Condition:** If the use case is successful, the author is added, updated, deleted. Otherwise the system is unchanged

**7.** Sign Up

**Actor:** The following actor will participate in this use case:

User

**Basic flow:** This case starts when the actor or user is new to the online management System.

1. The actor fills the signup form and provide the information like name, email, user id, password to the system
2. The user gets registered with system

**Alternate Flow:**

1. User already registered

**Pre – Condition:** The user must have email and user id for system.

**Post– Condition:** If the use case is successful, the user is successfully added to the system. Otherwise the system is unchanged

**8.** Change Password

This use case allow actor to change the password of the actor.

**Actor:** The following actor will participate in this use case:

Admin, user

**Basic flow:** This use case starts when admin or users wants to change the password.

1. Login
2. Once the admin and gets the required information the following action can be taken

* The user can change its own password
* The admin can change the user password if asked
* The admin can change its own password

**Alternate Flow:**

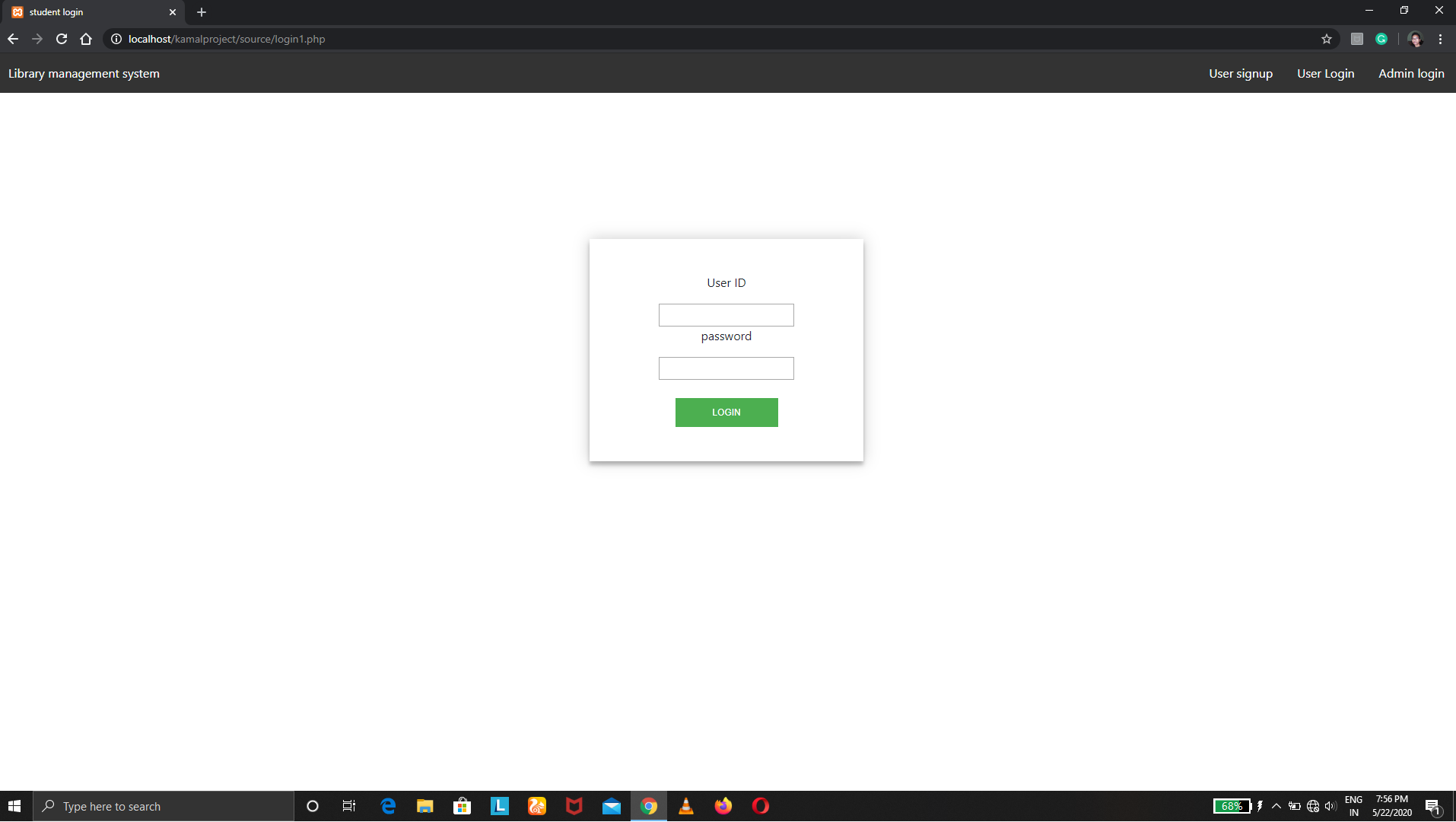
1. Password not changed

**Pre – Condition:** The admin and user must be logged onto system before use case begins.

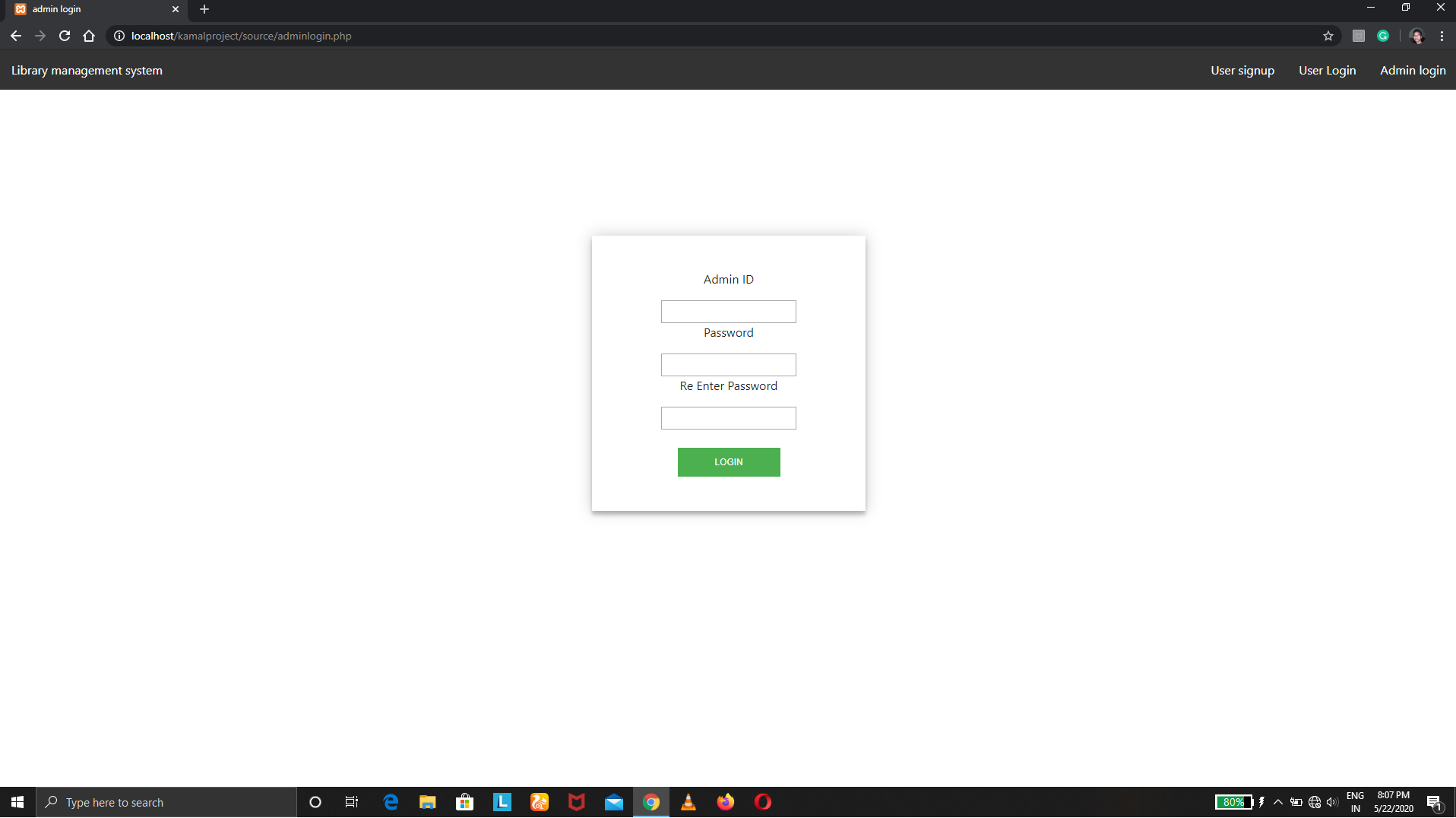
**Post– Condition:** If the use case is successful, the password of user or password of admin changed. Otherwise the system is unchanged.

**Input and Output Screen Design**

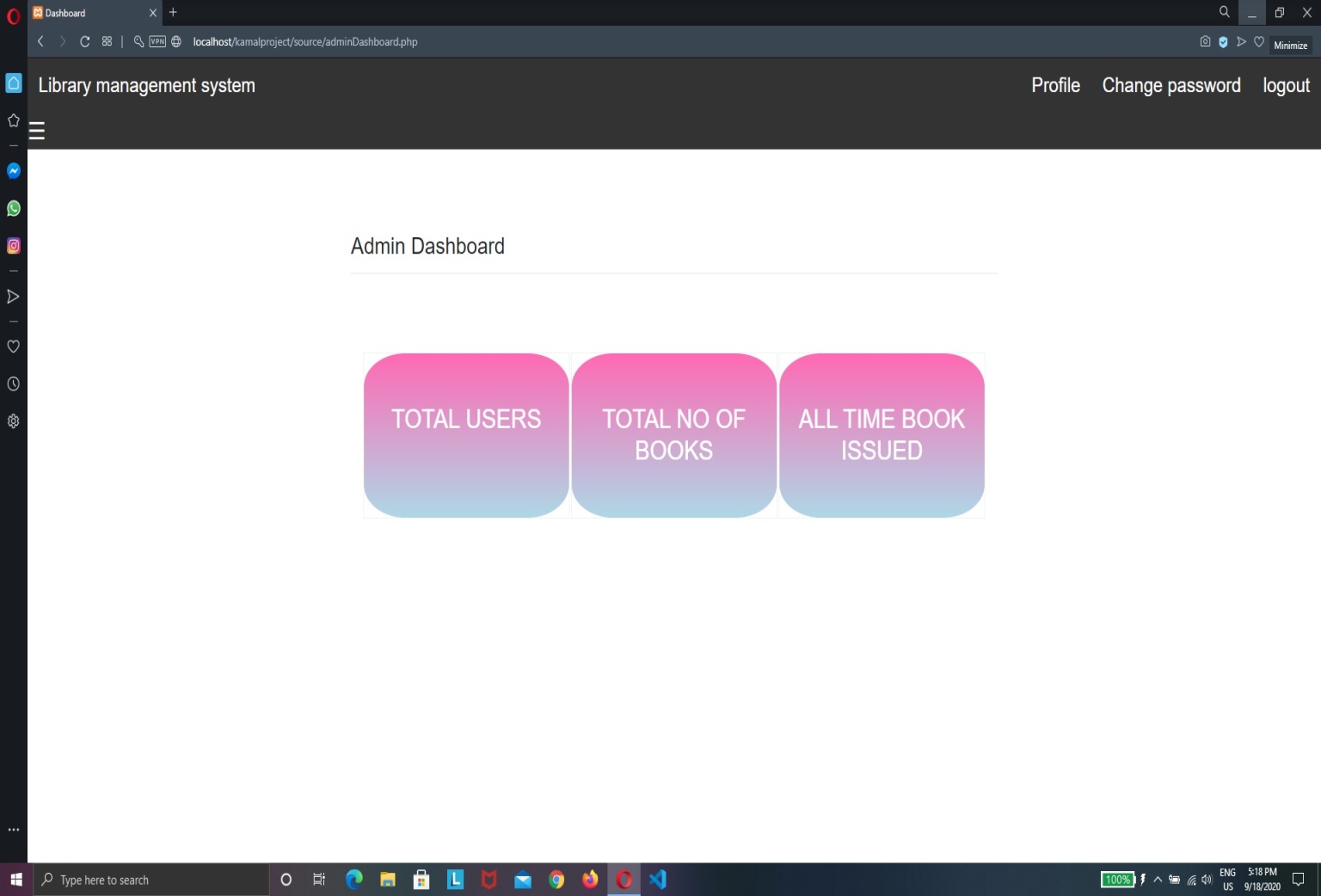
Login



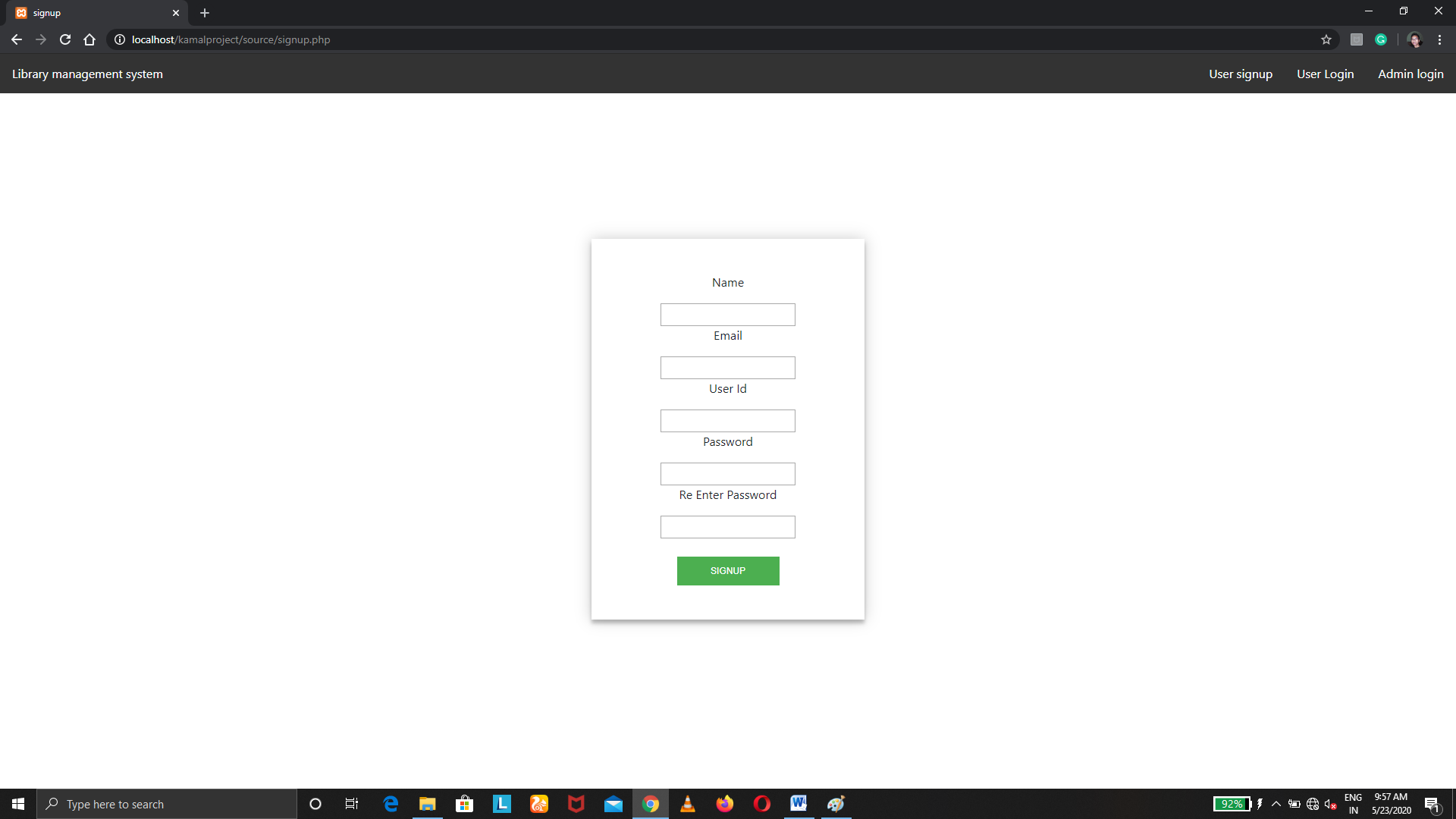
Admin login



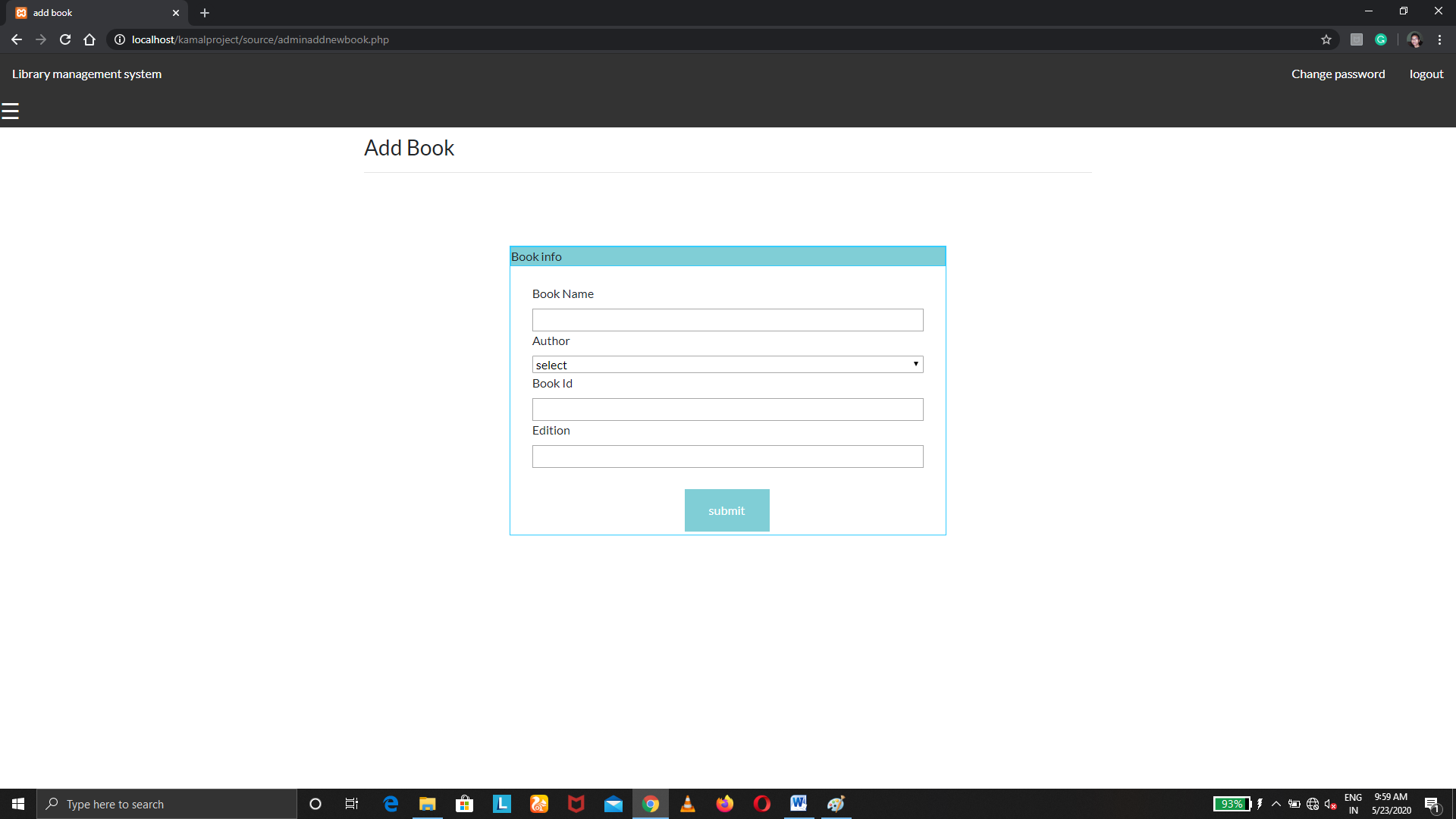
Admin Dashboard



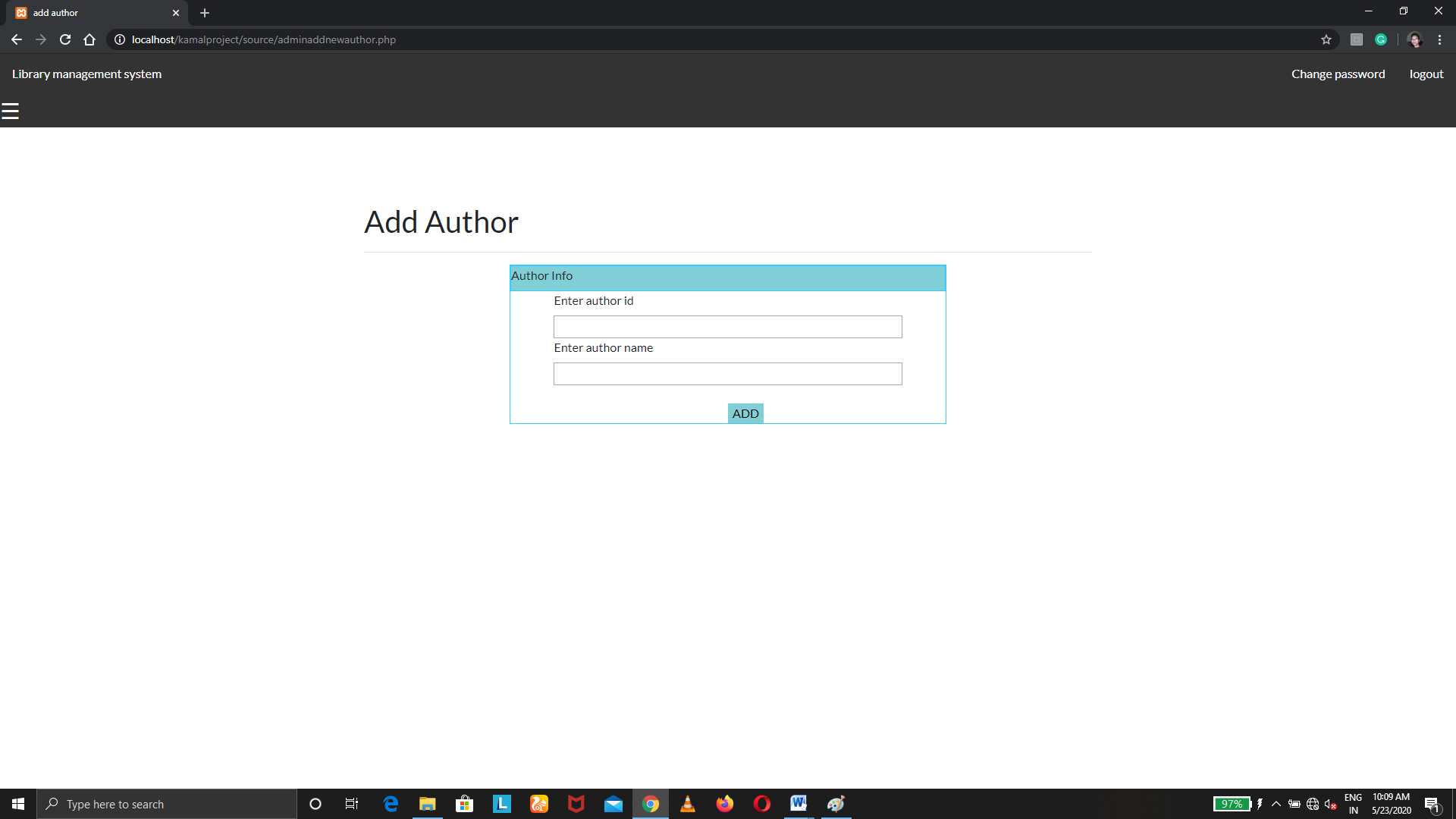
Signup



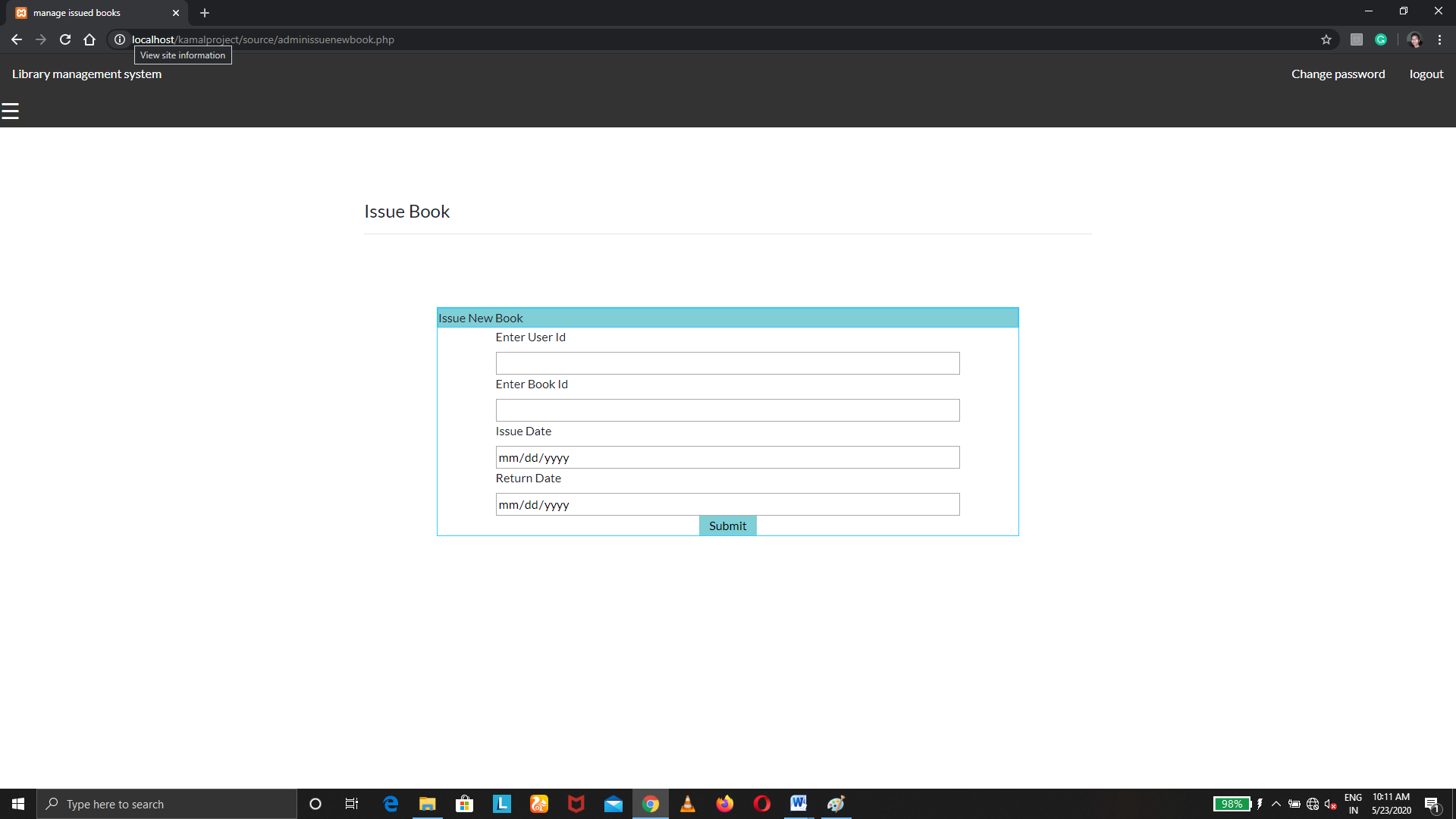
Add Book



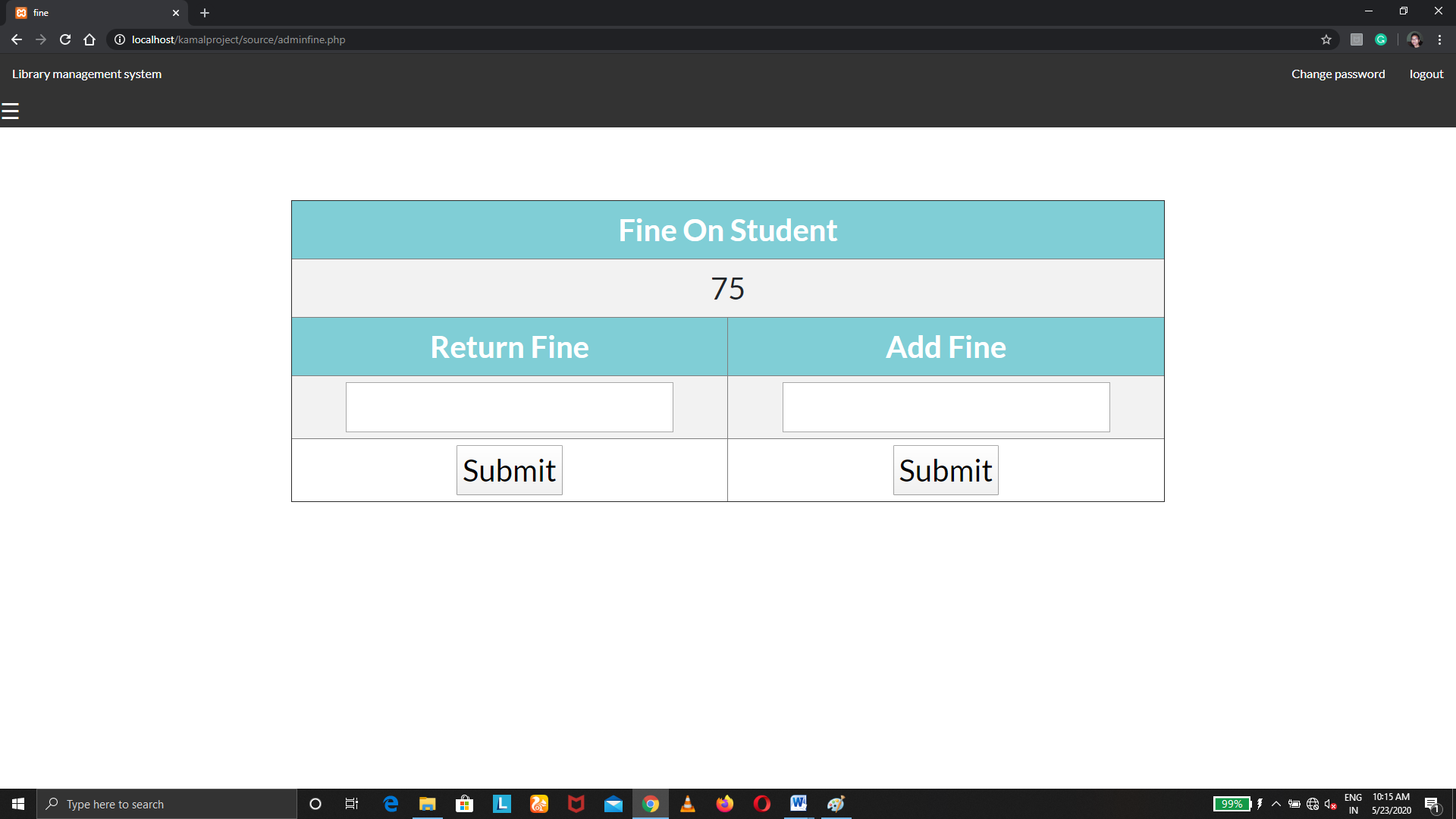
Add Author



Issue Book



Fine



PROCESS INVOLVED

Basically there are two types of users who deal with the system. The following two users are

1. Admin
2. User

The characteristics of each user are explained below:

1. Admin

* Admin manage books
* Admin manage author
* Admin manage users
* Admin manage fine
* Admin issue the book
* Admin return the book
* Admin can change the password of user
* Admin can view the issued book
* Admin add new book
* Admin add new author
* Admin update the book information
* Admin update the author information

1. User

* User can view the issued book
* User can see total fine on him/her
* User can view the available books
* User can view profile
* User can update profile
* User can change password

MODULE DESCRIPTION

Online library management system is divided into following module

1. Book management module
2. Author management module
3. Transaction module
4. Fine module
5. User management module
6. Admin module
7. User module
8. Book management module:

In book management module the admin or librarian has the following has the following function

Book management

View book

Delete book

Edit book

Add new book

1. Add new book: Here admin can add new book by providing book name, book id, edition and by selecting author.
2. Edit book: Here admin can change book name, edition and author.
3. Delete book: Here admin can delete the book.
4. View Book: Here user can view the books.
5. Author management module

In author management module admin or librarian has the following function

Author management

Delete author

Edit author

Add new Author

1. Add new author: Here the admin can add new author to the system.
2. Edit author info: Here admin can update the information of the author.
3. Delete author: Here admin can delete the author.

3. Transaction module

In transaction module admin or librarian has the following function

Transaction

Issue book

View issued book

Return book

1. Issue book: Here admin issue book.
2. Return book: Here admin return book.
3. View issued book: Here admin and user view issued book.
4. Fine Module

In fine module admin or librarian can perform the following function

Fine

View fine

Return fine

Add new fine

1. Add new fine: Here admin can add new fine.
2. Return fine: Here admin can return fine.
3. View fine: Here admin can view fine.
4. User Management Module

In user management admin or librarian has the following function

User Management

Change password

User status

1. User status: Here admin can set user status.
2. Change password: Here admin can change the password of user.
3. Admin module

In admin module, admin or librarian has the following function

Admin

Admin profile

Update information

Change password

1. Admin profile: Here admin can view the admin profile information.
2. Change password: Here admin can change password.
3. Update information: Here admin can change admin information.
4. User module

In user module user has the following function

User

Update profile

Update profile

User profile

Available book

View issued book

1. View book issued: Here user can view issued book.
2. Available book: Here user can view available book.
3. User profile: Here User can view user information.
4. Update profile information: Here user can update information.
5. Change password: Here user can change password.

SYSTEM TESTING

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not.

Our Project went through two levels of testing

1. Unit testing

2. Integration testing

1. UNIT TESTING

Unit testing is undertaken when a module has been created and successfully reviewed .In order to test a single module we need to provide a complete environment i.e. besides the module we would require

The procedures belonging to other modules that the module under test calls

 Non local data structures that module accesses

A procedure to call the functions of the module under test with appropriate parameters

TEST CASES

1. Login

TEST CASE ID: T1

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the user will be entered into the system. |
| INPUTS: User Id, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to online library management system  2. enter user id  3. enter password  4.click login | User id= kamal  Password = kamal | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to online library management system  2. enter user id  3. enter password  4.click login | User d id=Test  Password = 12345 | User should not Login into application | As Expected | Pass |

1. Sign Up

TEST CASE ID: T2

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to register user into the system | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: The user will successfully registered with the system |
| INPUTS: Name, Email, User Id, Password, Confirm Password | IF FAILS, ANY POSSIBLE REASON: If any of the field left blank, or user already registered with the system, password and confirm password do not match |
| EXPECTED OUTPUTS: User get registered with the system. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor will successfully registered with system. | ANY SUGGESTION: By entering valid information actor can successfully register. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Sign up with valid Data | 1. Go to online library management system  2. click on user sign up  3. enter name  4.enter email  5. enter user id  6.enter password  7. confirm password  8.click on sign up | Name bharat  Email [bharat@gmail.com](mailto:bharat@gmail.com)  User id bharat123  Password bharat  Confirm password bharat | User should register into application | As Expected | Pass |
| TU02 | Check User sign up with invalid Data | 1. Go to online library management system  2. click on user sign up  3. enter name  4.enter email  5. enter user id  6.enter password  7. confirm password  8.click on sign up | Name bharat  Email [bharat@gmail.com](mailto:bharat@gmail.com)  User id bharat123  Password bharat  Confirm password 1234 | User should not register into application | As Expected | Pass |

1. Add new book

TEST CASE ID: T3

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to add new book into the system | EXECUTION HISTORY: The author is already registered with the system |
| PRECONITION: User must be login with the system before this use case begin | RESULT: the book will get added |
| INPUTS: Name, select author, book id, edition. | IF FAILS, ANY POSSIBLE REASON: If book is already registered, field are left blank, book id is already exists. |
| EXPECTED OUTPUTS: Book get added to the system | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the book will add with system. | ANY SUGGESTION: By entering valid information actor can add book. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Adding new book to the system | 1. Go to online library management system  2. Click on add new book.  3. enter book name  4.slelect author  5.Enter book id  6. Enter book edition  7.click on submit | Book name : C  Author : author1  Book id: 1  Edition 1 | Book added to the system | As Expected | Pass |
| TU02 | Adding new book to the system with invalid details | 1. Go to online library management system  2. Click on add new book.  3. enter book name  4.slelect author  5.Enter book id  6. Enter book edition  7.click on submit | Book name : C  Author : author1  Book id: 1  Edition | Book not added to the system | As Expected | Pass |

1. Add new author

TEST CASE ID: T4

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to add new author into the system | EXECUTION HISTORY: |
| PRECONITION: User must be login with the system before this use case begin | RESULT: the author will get added |
| INPUTS: Admin name, author id | IF FAILS, ANY POSSIBLE REASON: If author is already registered, field are left blank, author id already exists. |
| EXPECTED OUTPUTS: Author get added to the system | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the author will add with system. | ANY SUGGESTION: By entering valid information actor can add author. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Adding new author to the system | 1. Go to online library management system  2. Click on add author.  3. enter author name  4.enter author id  5.click on submit | Author name: author2  author id: 2 | Author registered with system | As Expected | Pass |
| TU02 | Adding new author to the system with invalid details | 1. Go to online library management system  2. Click on add author.  3. enter author name  4.enter author id  5. click on submit | Author name: author2  author id: | Author not added tot system | As Expected | Pass |

1. Issue new Book

TEST CASE ID: T5

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to issue book to users from system | EXECUTION HISTORY: |
| PRECONITION: User must be login with the system before this use case begin | RESULT: the book will get issued |
| INPUTS: User id, Book id, issue date, return date. | IF FAILS, ANY POSSIBLE REASON: If user id is invalid, book id is invalid |
| EXPECTED OUTPUTS: Book get issued | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the book will get issued. | ANY SUGGESTION: By entering valid information, actor can issue book. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Issuing book to the user form the system | 1. Go to online library management system  2. Click on issue book  3. Enter user id  4. Enter book Id  5. select issue date  6.select return date  7.Click on submit | User id kamal  Book id 1  Issue date 26-05-2020  Return date 31-05-2020 | Book issued to user | As Expected | Pass |
| TU02 | Issuing book to the user form the system with invalid details | 1. Go to online library management system  2. Click on issue book  3. Enter user id  4. Enter book Id  5. select issue date  6.select return date  7.Click on submit | User id 61  Book id 6  Issue date 26-05-2020  Return date 31-05-2020 | Book not issued to user | As Expected | Pass |

1. Edit book detail

TEST CASE ID: T6

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to Edit book details | EXECUTION HISTORY: The book is already added to the system |
| PRECONITION: User must be login with the system before this use case begin | RESULT: the book detail will get updated |
| INPUTS: name, author, edition | IF FAILS, ANY POSSIBLE REASON: if wrong information is entered |
| EXPECTED OUTPUTS: book details updated | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the book detail will get updated. | ANY SUGGESTION: By entering valid information, actor can edit update book detail. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Updating book detail | 1. Go to online library management system  2. Click on  Manage book  3. Select book click on update detail  4. Enter book name  5. select author  6. Enter book Edition | Name C  Select author author3  Edition 3 | Book details updated | As Expected | Pass |
| TU02 | Updating book detail with wrong detail | 1. Go to online library management system  2. Click on  Manage book  3. Select book click on update detail  4. Enter book name  5. select author  6. Enter book Edition | Name C  Select author author3  Edition | Book details not updated | As Expected | Pass |

1. Change password

TEST CASE ID: T7

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to change password | EXECUTION HISTORY: |
| PRECONITION: User must be login with the system before this use case begin | RESULT: the password will change |
| INPUTS: old password, new password, confirm new password | IF FAILS, ANY POSSIBLE REASON: if wrong password is entered |
| EXPECTED OUTPUTS: password changed | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the password changes. | ANY SUGGESTION: By entering valid information, actor can update password. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Updating password | 1. Go to online library management system  2. Click on change password  3. enter old password  4. enter new password  5. confirm new entered password  6. click on submit | Old password kamal  Enter new password 1234  Confirm password 1234 | Password changed | As Expected | Pass |
| TU02 | Updating password with wrong old password | 1. Go to online library management system  2. Click on change password  3. enter old password  4. enter new password  5. confirm new entered password  6. click on submit | Old password abcd  Enter new password 1234  Confirm password 1234 | Password not changed | As Expected | Pass |

2. INTEGRATION TESTING

In this type of testing we test various integration of the project module by providing the input .The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

Future scope

This website provides a computerized version of library management system which will benefit the students as well as the other users of the library.

It makes entire process online where student can search books. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of teacher’s login where teachers can add lectures notes and also give necessary suggestion to library and also add info about workshops or events happening in our college or nearby college in the online notice board.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility , a feature

Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfills each users need in the best way possible.

References

http://www.w3schools.com/html/html\_intro.asp

http://www.w3schools.com/css/css\_background.asp

http://www.w3schools.com/js/js\_datatypes.asp

http://www.w3schools.com/sql/sql\_insert.asp

http://www.w3schools.com/sql/sql\_update.asp

http://www.w3schools.com/php/php\_forms.asp