



**A**  
**PROJECT REPORT**  
**On**  
**Library Management System**  
**Using Rest API**

**By**  
**(Batch: 2022-7469)**

<b>Pooja Yadav</b>	<b>EBEON0522606589</b>
<b>Bushra Khan</b>	<b>EBEON0522606347</b>
<b>Priyanka Deore</b>	<b>EBEON0522601610</b>
<b>Dimpal Raghuvanshi</b>	<b>EBEON0522604637</b>
<b>Saloni More</b>	<b>EBEON0522608243</b>

**Under the Supervision of**  
**MS. Pooja Mehta**



**India's leading Workforce Development Platform**

## INDEX

SR. NO	CONTENT	PAGE NO.
1	INTRODUCTION	4
2	EXISTING SYSTEM	5
3	PROPOSED SYSTEM	6
4	SOFTWARE AND HARDWARE REQUIREMENTS	8
5	ANNOTATION	10
6	UML DIAGRAMS	12
7	IMPLEMENTATION	19
8	CONCLUSION	23
9	FUTURE SCOPE	24
10	REFERENCES	25

## ABSTRACT

Library Management System is a system which maintains the information about the books present in the library, their authors, the members of library to whom books are issued, library staff and all. This is very difficult to organize manually. maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of a Library becomes much simple. The Library Management has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced.

Hence in the existing system for Library Management System, the performance evaluation system and the maintenance are done manually. The proposed system will maintain all the information in a standard database and will be able to generate reports when necessary.

## 1. INTRODUCTION

A library is a collection of sources of information and similar resources , made accessible to a defined community for references and borrowing. It provides physical and digital access to material, and may be physical building or room, or a virtual space, or both. Library is collection of books, periodicals, newspapers, manuscripts, films, documents, CDs, e-books, audiobooks. databases, and other formats.

Owing to the advancement of technology, organization of a Library becomes much simple.

The Library Management has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced.

### 1.1 OBJECTIVES

- To select, evaluate and acquire library materials in varied formats to meet and respond to the needs of our diverse community.
- To maintain and preserve books, materials and resources with historical, cultural, social, economic and archival value, and other related materials in an organized collection to provide members of the community these materials and enriched their personal and professional lives.
- The main objective of this project is to keep track of all the information about library activities, their complete details, and total number of transactions done in the library.
- It saves time of management system.

## 2. Existing System

In early days, libraries were managed manually. It required lot of time to record or to retrieve the details. The employees who have to record the details must perform their job very carefully. Even a small mistake would create a lot of problems. Security of information is very less. Report generations of all the information was very tough task. Maintenance of Library catalogue and arrangement of the books to the catalogue was very complex task. In addition to its maintenance of member details, issue dates and return dates etc. manually is a complex task. All the operations must be performed in perfect manner for the maintenance of the library without any degradation which may finally result in the failure of the entire system.

### 3. PROPOSED SYSTEM

To overcome the inconveniences as mentioned in the existing system, a Library Management System is proposed.

The proposed system contains the following features:

- The students will register them through Online
- Individually each member will have his account through which he can access the information he needs.
- Book details like authors, number of copies totally maintained by library, present available number of books, reference books, non-reference books etc. all this information can be made handy.
- Regarding the members designation, number of books was issued.
- Issue dates and returns of each member is maintained separately and fine charged if there is any delay in returning the book.
- Administrator can add, update, delete the books.
- Time consuming is low, gives accurate results, reliability can be improved with the help of security.

The proposed system has divided into further modules:

#### 3.1 User/ Member Module

**This module is further divided into various sub-modules describing the user in a better way:**

##### **New User Register:**

To sign up a new user to this system.

##### **Student Login:**

So as to confirm that only an authenticated user is using the project.

##### **Search Book:**

The user can search book based on book id, book name, or by author name.

##### **Issue Book:**

To help the user get the required books issued.

**Return Book:**

To return the book before the last date without fine, or after the specified time duration with a late fine.

**3.2 Admin Module**

It is to be operated by the admin with a unique id and password. The admin is the person who decides authentication and authorization for all the different users of the application.

**It further can be subdivided as:**

- Register user.
- Issue Book.
- Maintain books in a stack, which means record the availability at a regular time interval.

**3.3 Book Module**

These are the basic building block of this system as well as any library. In other words, the main purpose of any library and the cause to develop systems like this.

**Book-Name:**

The name of the book which is almost unique in some way.

**Author Name:**

The one who has written the book. As sometimes the book's series becomes more popular by the author's name rather than the book name.

**Book Price:**

The market value of the book is also required to maintain in the record, as sometimes it is needed to arrange and sort based on this, secondly, it is also required for compensation in case of loss or damage, as fine charges.

**Book Department Name:**

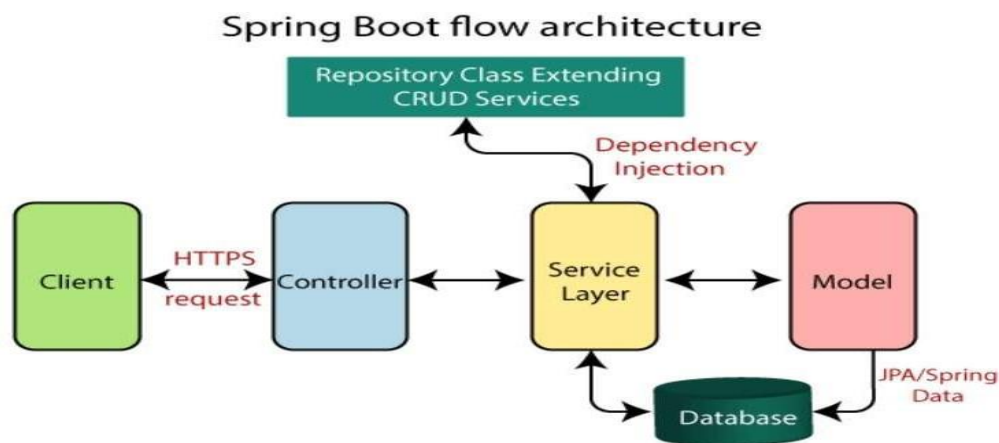
The name of the department in which the book is almost unique in some way.

## 4. SOFTWARE AND HARDWARE REQUIREMENTS

### 4.1 SOFTWARE REQUIREMENTS

- **Spring Boot**

Spring Boot is a java framework used for develop standalone application. Need strong knowledge in OOPS & Java concept then only working with spring framework. Mostly all applications are developed by spring boot. Because it was very secure no one hack the information. Spring Boot is an opensource Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.



**Fig. Spring Boot flow architecture.**

- **HTML**

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.





- **CSS**

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colours are to be used, different displays for different devices and screen sizes, and much more!

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the **style** attribute inside HTML elements
- **Internal** - by using a **<style>** element in the **<head>** section
- **External** - by using a **<link>** element to link to an external CSS file

## 4.2 HARDWARE REQUIREMENTS

- Intel core i5 2nd generation is used as a processor because it is fast than other processors and provide reliable and stable and we can run our pc for longtime. By using this processor, we can keep on developing our project without any worries.
- Ram 2 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing.

## 5. ANNOTATION

- **@Autowired:**

Spring provides annotation-based auto-wiring by providing @Autowired annotation. It is used to auto wire spring bean on setter methods, instance variable, and constructor. When we use @Autowired annotation, the spring container auto-wires the bean by matching data-type.

- **@RequestMapping:**

It is used to map the web requests. It has many optional elements like consumes, header, method, name, params, path, produces, and value.

- **@RestController:**

It is a convenience annotation for creating Restful controllers. It is a specialization of @Component and is autodetected through classpath scanning.

- **@GetMapping:**

It maps the HTTP GET requests on the specific handler method. It is used to create a web service endpoint that fetches. It is used instead of using: @RequestMapping(method = RequestMethod.GET).

- **@PostMapping:**

It maps the HTTP POST requests on the specific handler method. It is used to create a web service endpoint that creates. It is used instead of using: @RequestMapping(method = RequestMethod.POST)

- **@PutMapping:**

It maps the HTTP PUT requests on the specific handler method. It is used to create a web service endpoint that creates or updates. It is used instead of using: @RequestMapping(method = RequestMethod.PUT)

- **@DeleteMapping:**

It maps the HTTP DELETE requests on the specific handler method. It is used to create a web service endpoint that deletes a resource. It is used instead of using: @RequestMapping(method = RequestMethod.DELETE)

- **@PatchMapping:**

It maps the HTTP PATCH requests on the specific handler method. It is used instead



of using: `@RequestMapping(method = RequestMethod.PATCH)`

- **@NotNull:**

It is actually, an explicit contract declaring that: A method should not return null. Variables (fields, localvariables, and parameters) cannot hold a null value.

- **@SpringBootApplication:**

This annotation is used to mark a configuration class that declares one or more `@Bean` methods and also triggers auto-configuration and component scanning.

- **@Id:**

This annotation is inherited from `javax.persistence.Id`, indicating the member field below is the primary key of the current entity.

- **@Generated:**

The Generated annotation is used to mark source code that has been generated.

- **@ManyToOne:**

This mapping means that many instances of this entity are mapped to one instance of another entity – many items in one cart.

- **@GeneratedValue:**

This annotation specifies the generation strategies for the values of primary keys

- **@Column:**

This annotation is used for Adding the column the name in the table of a particular MySQL database.

- **@JoinTable:**

This specifies the mapping of associations. It is applied to the owning side of an association.

- **@Service:**

It is also used at class level. It tells the spring that class contains the business logic.

- **@Entity:**

This annotation defines that a class can be mapped to a table. And that is it, it is just a marker, like for example `Serializable` interface.

- **@JoinColumn:**

Specifies a column for joining an entity association or element collection.

## 6. UML DIAGRAM

### 6.1 USE CASE DIAGRAM

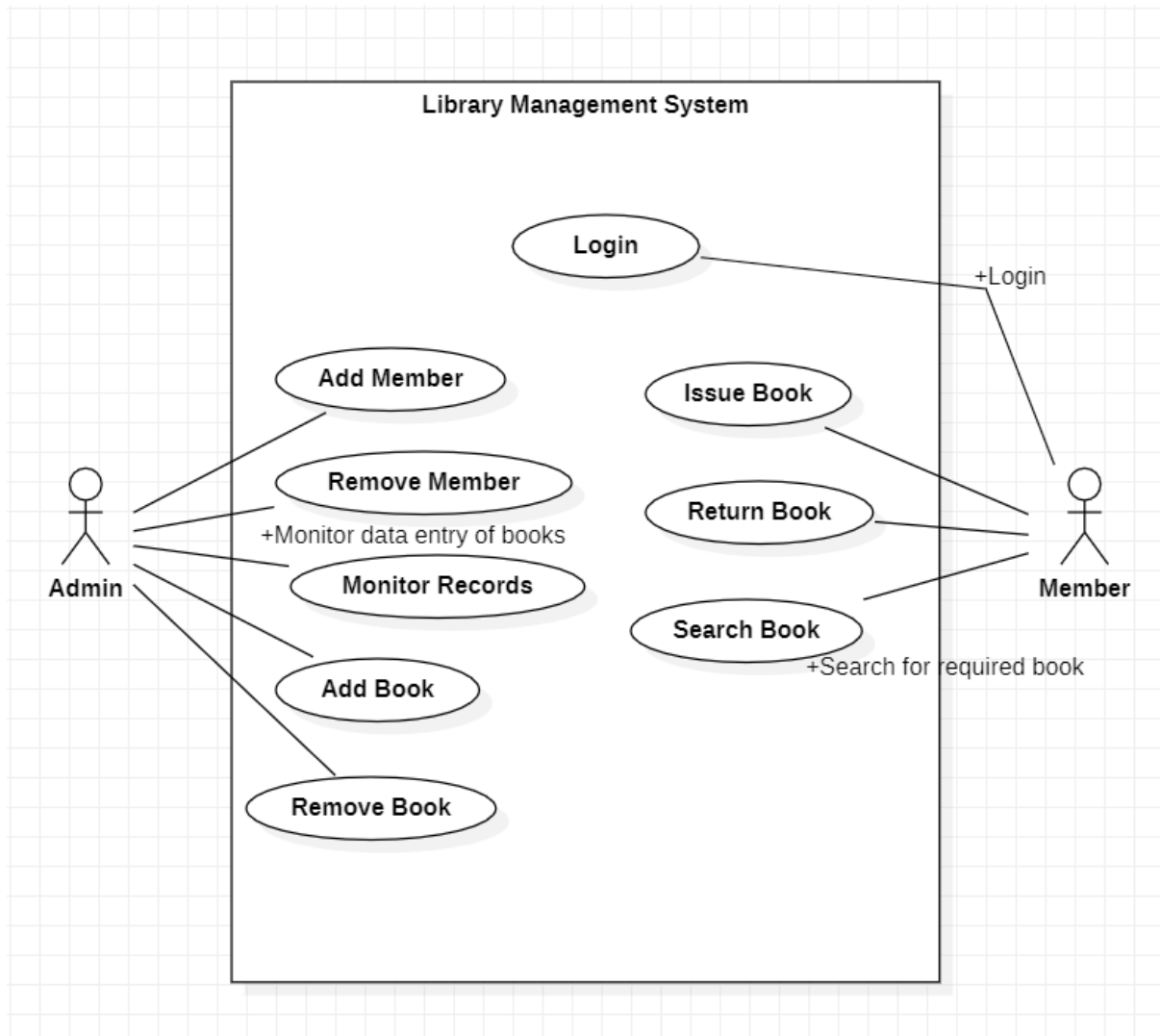


Fig. Use case diagram for Library management System

#### ➤ USECASE DESCRIPTION

- LOGIN



### **1. Brief Description:**

This use case describes how an actor login into the 'Library Management System'.

### **2. Actors:**

Admin, Member/User

### **3. Flow of events:**

#### **a. Basic Flow:**

This use case starts when the actor wishes to log into the 'Library Management System'.

The system requests the actor to enter the username and password.

The actor enters his/her username and password.

The system validates the username and password and then the actor is logged into the system.

#### **b. Alternative flows:**

If in the basic flow, the actor enters an invalid name or password, the system displays an error message.

The actor can use to either return to the beginning of the basic flow or cancel the login at the point where use case ends.

## **• ADD/REMOVE\_MEMBER**

### **1.Brief Description:**

This use case describes how an actor adds or deletes the user's record in the system.

### **2. Actors:**

Admin

### **3.Flow of events:**

#### **a. Basic Flow:**

This use case starts when the actor(Admin) has to add or remove users/members within the 'Library Management System'.

The actor fills the details of the member.

The member is then added to the system now the member got registered for further uses.

The actor can delete a member by filling the appropriate details.

**b. Alternative flows:**

If in the basic flow the actor fills wrong details while deleting a member then the system displays an error message.

The actor can either return to the beginning of the basic flow or can cancel the deletion of the member from the system.

- **ISSUE BOOK**

**1. Brief Description:**

This use case describes how the admin issues book when requested by the member.

**2. Actors:**

Admin, Member

**3. Flow of events:****a. Basic Flow:**

If a member wants to borrow a book it is important that the staff should login to the system.

If login is successful the admin should enter the member id to be searched.

If the member search is successful the admin should enter the book id.

If the book is available then it can be borrowed.

**b. Alternative flows:**

If the login fails then the admin should re- register themselves.

If the member search is unsuccessful then the staff should re-register the student.

If the book search is unsuccessful and book data is not found then admin must enter the book in requisition report.

- **RETURN BOOK**

**1. Brief Description:**

This use case describes how the return book procedure carried out when requested by the member.

**2. Actors:**



Admin, Members

### **3. Flow of events:**

#### **a. Basic Flow:**

Member gives the book to be returned to the admin

Admin checks if the book is returned on time.

Admin update the book records.

#### **b. Alternative flows:**

In the basic flow the admin checks if the book is returned on time if it is not on the time then he/she generates slip of calculated fine.  
The member submits the fine.

## **• MAINTAIN MEMBER RECORD**

### **1. Brief Description:**

This use case describes how the actor maintains the record of members which includes edit or view the member's data.

#### **2. Actors:**

Admin

### **3. Flow of events:**

#### **a. Basic Flow:**

Staff member login to the system and selects the menu option to change the data of specific member.

Enter the name of categories that he/she want to change.

The system save the change in membership record and update previous record.

To view the record of a member the actor selects from the menu option.

LMS presents the record of members.

#### **b. Alternative Flows**

If the password is incorrect then a message is printed on the screen and admin is returned to



the beginning.

If the name of changed to be category is not among the existing categories a message is printed on the screen and the actor is returned to the menu screen.

.

- **SEARCH A BOOK**

- 1. Brief Description:**

This use case describes how the actor can search for a particular book.

- 2. Actors:**

Member  
Admin

- 3. Flow of events:**

- a. Basic Flow:**

Member or staff enters the book name or ISBN or author name and presses search  
If the search is successful then that book is displayed on the screen.

- **ADD/DELETE BOOK**

- 1. Brief Description:**

This use case describes how the actor adds or removes the books.

- 2. Actors:**

Admin

- 3. Flow of events:**

- a. Basic flow:**

The admin login to the system.

If login is successful then to add a book the admin must search for the book.

If the book is not found then it is checked in the requisition list.

If the book is not currently available and is part of the requisition list it can be





added to the book database.

To remove a book it is again searched in the library system.

If it is found it should be checked in borrower record.

If it is not in the borrowed list it can be removed.

- **MAINTAIN BOOK RECORD**

### **1. Brief Description:**

This use case describes how the actor maintains the book record in the database of the system which includes added, issued or returned books record

### **2. Actors:**

Admin

### **3. Flow of events:**

#### **a. Basic flow:**

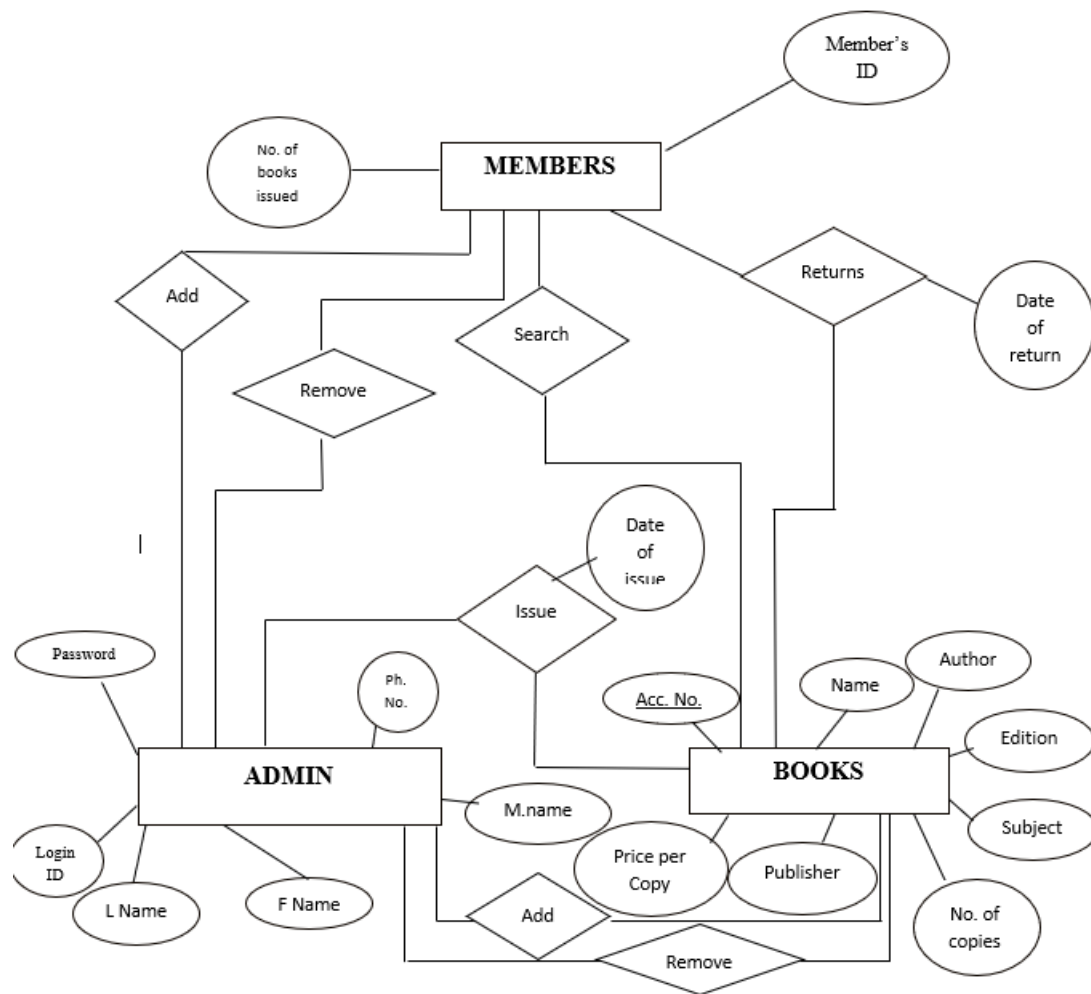
The admin selects the menu option to add record of different books. He/ She enters the name , Author's name, edition etc. or he/she can use a barcode.

To enter the record of issued book the admin goes to menu option.

LMS presents the menu for maintaining the issued books record which contains two options

- a. Add issued books record
- b. Edit issued books record.

## 6.2 E-R DIAGRAM



**Fig. E-R Diagram for Library Management System**



## 7. IMPLEMENTATION

The screenshot shows a web browser window with the address bar displaying "localhost:8080/userlogin". The page has a dark navigation bar at the top with the text "LBM-System" and several menu items: "Home", "About", "ContactUs", "BookServices", and "UserServices". There is also a "Logout" link and a search bar with a "Search" button. The main content area is white and features the heading "Login Here" in blue. Below this heading are two input fields: "Username:" and "Password:". A blue "submit" button is positioned below the password field. To the right of the submit button is a link that says "Forget Password?". At the bottom of the page, there is a green horizontal line followed by the copyright notice: "Copyright © 2019 All Rights Reserved by Chinna Alupuru :SmartLibrarySystem - India".

localhost:8080/userlogin

Postman

userlogin

LBM-System

Home About ContactUs BookServices UserServices

Logout Search Search

Login Here

Username:

Password:

submit

Forget Password?

Copyright © 2019 All Rights Reserved by Chinna Alupuru :SmartLibrarySystem - India



Smart LBM-System

Home About ContactUs BookServices UserServices Logout

Search Search

## Invalid Username & Password

### Login Here

Username:

Password:

submit

[Forget Password?](#)

Copyright © 2019 All Rights Reserved by Chinna Alupuru :[SmartLibrarySystem - India](#)

localhost:8080/contactus

Smart LBM-System

Home About ContactUs BookServices UserServices Logout

Search Search

Drop

Add/Remove Book

Search BookDetails

Smart Library Management Book System

Show List of BarrowBooks

Total History

Your Name \*

Your Email \*

Your Phone Number \*

Send Message

Copyright © 2019 All Rights Reserved by Chinna Alupuru :[SmartLibrarySystem - India](#)



localhost:8088/add-member x +

localhost:8088/add-member

## ADD NEW MEMBER HERE

Enter Member Name:	Enter Member Address:
<input type="text" value="bhusr"/>	<input type="text" value="address"/>
Enter Member Phone:	Enter MemberDepartment:
<input type="text" value="1234567890"/>	<input type="text" value="Computer"/>
Gender:	Age:
<input type="text" value="male/female"/>	<input type="text" value="21"/>

[ADD MEMBER](#)

localhost:8088/add-book x +

localhost:8088/add-book

Smart LBM-System

[Home](#) [About](#) [ContactUs](#) [BookServices](#) [UserServices](#) [Logout](#)

[Click For Back](#)

## ADD NEW BOOK HERE

Enter Book Name:	Enter Book Author Names:
<input type="text" value="c"/>	<input type="text" value="I"/>
Enter Book Price:	Enter BookDepartment:
<input type="text" value="100.25"/>	<input type="text" value="Department"/>

[ADD BOOK](#)



localhost:8088/show-listof-all-bo x Postman x +

localhost:8088/show-listof-all-books

Smart LBM-System Home About ContactUs BookServices UserServices Logout Search Search

### List of Books

Book Number	Book Name	Author	Date of Entered		
1	C	DenniesRich	19-04-2019	Edit	Delete
2	Java	James	19-04-2019	Edit	Delete
3	Python	Rossum	19-04-2019	Edit	Delete

Add New Book

Copyright © 2019 All Rights Reserved by Chinna Alupuru :SmartLibrarySystem - India



## CONCLUSION

This project provides a computerized version of library management system which will benefit the students as well as the staff of the library. It makes entire process online where user can search books, admin can generate reports and do book transactions. It also has a facility for user login where user/member can login and can see status of books issued as well request for book or give some suggestions.



## **FUTURE SCOPE**

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/html/html_intro.asp)
- [http://www.w3schools.com/js/js\\_datatypes.asp](http://www.w3schools.com/js/js_datatypes.asp)