



Chongqing University of Technology

# 《Advance Java》

## Integrated Application Project

### LibrarySSH Management System

Academic Term: 2020

Class: 6201703L1

Student ID: 62017010084

Name: Md Anower(an hao ming)

Teacher: Liu Yahui

Score:

## Table of Contents

<b>1 System Overview .....</b>	<b>3</b>
<b>2 Database Design .....</b>	<b>7</b>
2.1 Users Database .....	7
2.2 Student Database .....	8
2.3 Book Database .....	8
2.4 Book Category Database .....	9
<b>3 Specific Function Description .....</b>	<b>9</b>
3.1 User Login .....	9
3.2 User Registration.....	12
3.3 Home Page .....	13
3.4 User Manage .....	15
3.4.1 User List.....	15
3.5 Student Mange .....	18
3.5.1 Student List .....	18
3.6 Book Manage .....	24
3.6.1 Book Category List .....	24
3.6.2 Book List.....	31
3.6.3 Book Search .....	39
<b>4 Summary .....</b>	<b>40</b>

## 1 System Overview

In this modern era of the internet, almost all of us rely on web-based applications from small to big tasks. Well, LibrarySSH management system is one of the most popular use-cases considered by the professionals while building applications in Java. In this report, I tried to show how to create a library management system project using jsp, framework. LibrarySSH management system project report – The main objective of the Library Management system project is discipline of the planning, organizing and managing the library tasks. Our project aims at making the task of library easy. LibrarySSH Management is entering the records of new book and retrieving the details of book available in the library. We can issue book to the library member and maintain their records and can also checks how many book are issued and stock available in the library. In the project we can maintain the late fine of library member who return the issued book after the due date. This LibrarySSH management system project in JAVA and MS SQL server. To execute the below project, you will need the following business Requirements:

- MySQL Community Server
- MySQL JDBC Connector
- Java
- Eclipse IDE
- Sqljdbc4.jar

The Sqljdbc4.jar is used to display the data in a table format. So, once you create a project in Eclipse IDE, you have to import the Sqljdbc4.jar and JDBC connector JAR into the project. To do that, **right-click on the project**, choose **Build Path > Configure Build Path**. In the dialog box, which opens up, choose **Add External JARs**, and add the JAR files. Once added, click on **Apply and Close**. This is the main module in the proposed project. The administrator can read and write information about any member. The administrator can also update, create and delete the record of membership as per requirement and implementation plans. Now, for this particular project, I have considered three tables, which are:

**User Manage:** - This table consists of the columns

{UID, Username, and Password}

**Book Manage:** - The book's table consists of the columns

{BID, Book name, Picture, Category ID, Description, Remark}

**Book Category:** - The book's table consists of the columns

{Book Category ID, Book Category Name, Remark}

**Student Manage:** - This table consists of the columns

{Student ID, Student No, Name, Age, and Gender}

We can consider much future scope to this application. The following are some of there. Online use of the librarySSH can be good feature for the Library Management

system. Advanced fine payment system can be added. Inventory system can be used to maintain the books of the library. Alright, so now that the Initial set is done, and I have told you the schema of tables, let us get started. In this diagram, the user has been shown more specifically as a student who will be the first login to the system, to get access to the application. After getting the authentication and being authorized, the user will use the system with ease and security. There is a database maintained for storing the records of books that are available in the stock, books that have been issued to some user, then the return date of each issued book. After logging in the user will search for the books in the library using the subject code, book code, and therefore access to the rack number becomes easy, so ultimately using the rack number user finds the book, if it is available or not. Once the book has been found, if wanted the book, the user can get it issued easily with their unique college or university id. Now since the book has been issued it needs to be returned after a defined duration of time as a part of the system procedure. At a regular interval of time user can check if there is any fine charged on him/her or not. The librarian will also monitor all these activities at its end. And along with this, there is some additional work for the librarian to be done like, adding new books in the rack, updating the book availability status time to time, checking the user's identity is valid or not For a particular time period. There may be some cases at times when for example say

a student has been suspended due to some disciplinary action and therefore has been blocked from using any service provided by the university normally. In this case, it becomes the duty of the librarian to check the identity is valid to issue books or not. Also similar is the case when a student has already passed out from the university. Library Management System is an application refer to other librarySSH system and it is suitable to use by small and medium size library. It is use by librarian and library admin to manage the library using a computerized system. The system was developed and designed to help librarian record every book transaction so that the problem such as file missing or record missing will not happened again. Barcode reader is equipped in this system so that users can enjoy the convenience without need to key in the barcode of the book themselves. It is convenience and time saving as the users can direct scan in the book's barcode id when the members borrows few books in one time. Book and member maintenance module also included in Library Management System. Users can register or edit the member or book in the system. With this computerized maintenance, library will not lost the book record or member record which always happen when no computerized system bring used. In addition, report module is also included in LibrarySSH Management System. If user's position is Admin, the user is able to view different kind of report. First Type of report are rental and return report, user can check the rental, return transaction which

happen on particular day. Besides that, user can check the Top10 books which borrow by the member in a day, month or year based on category. Moreover, activity logs report also.

## 2 Database Design

Database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques. The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

Database Name: LibraryNewDB

### 2.1 Users Database

Table 2-1 Users

Table Name	Users
------------	-------

Column	Data Type	Allowable null Value	Constant Condition	Introduce
Id	int		Pk	Login Account
User Name	Nvarchar(20)			Login User Name
Password	Nvarchar(20)			User Password

## 2.2 Student Database

Table 2-2 Student

Table Name	Student			
Column	Data Type	Allowable null Value	Constant condition	Introduce
Id	Int		Pk	Student ID
No	Nvarchar(20)			Student No
Name	Nvarchar(20)			Student Name
Age	Int	✓		Student Age
Gender	Nvarchar(20)	✓		Male/ Female

## 2.3 Book Database

Table 2-3 Book

Table Name	Book			
Column	Data Type	Allowable null value	Constant Condition	Introduce



Id	Int		Pk	Book ID
Book Name	Nvarchar(20)			Name Of Book
Picture	Nvarchar(200)	✓		Image Of Book
Category ID	int	✓	FK	Id of Category
Remark	Nvarchar(20)	✓		Remark

## 2.4 Book Category Database

Table 2-4 Book Category

Table Name	Book Category			
Column	Data Type	Allowable null Value	Constraint condition	Introduce
Id	int		Pk	Category ID
Category Name	Nvarchar(20)			Name Of Category
Remark	Nvarchar(20)	✓		Remark

## 3 Specific Function Description

### 3.1 User Login

Logging in is usually used to enter a specific page, website or application, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site. Logging out may be performed explicitly by the user taking some actions, such as entering the appropriate command or clicking a website link labeled as such. It can also be done implicitly, such as by the user powering off his or her workstation,

closing a web browser window, leaving a website, or not refreshing a webpage within a defined period. In the case of websites that use cookies to track sessions, when the user logs out, session-only cookies from that site will usually be deleted from the user's computer. In addition, the server invalidates any associations with the session, thereby making any session-handle in the user's cookie store useless. This feature comes in handy if the user is using a public computer or a computer that is using a public wireless connection. As a security precaution, one should not rely on implicit means of logging out of a system, especially not on a public computer; instead, one should explicitly log out and wait for the confirmation that this request has taken place. In the log in page the students has already registered by giving there information in registration page. Here just input the unique name which is username and the password. After the login all thing can access the logged in student.

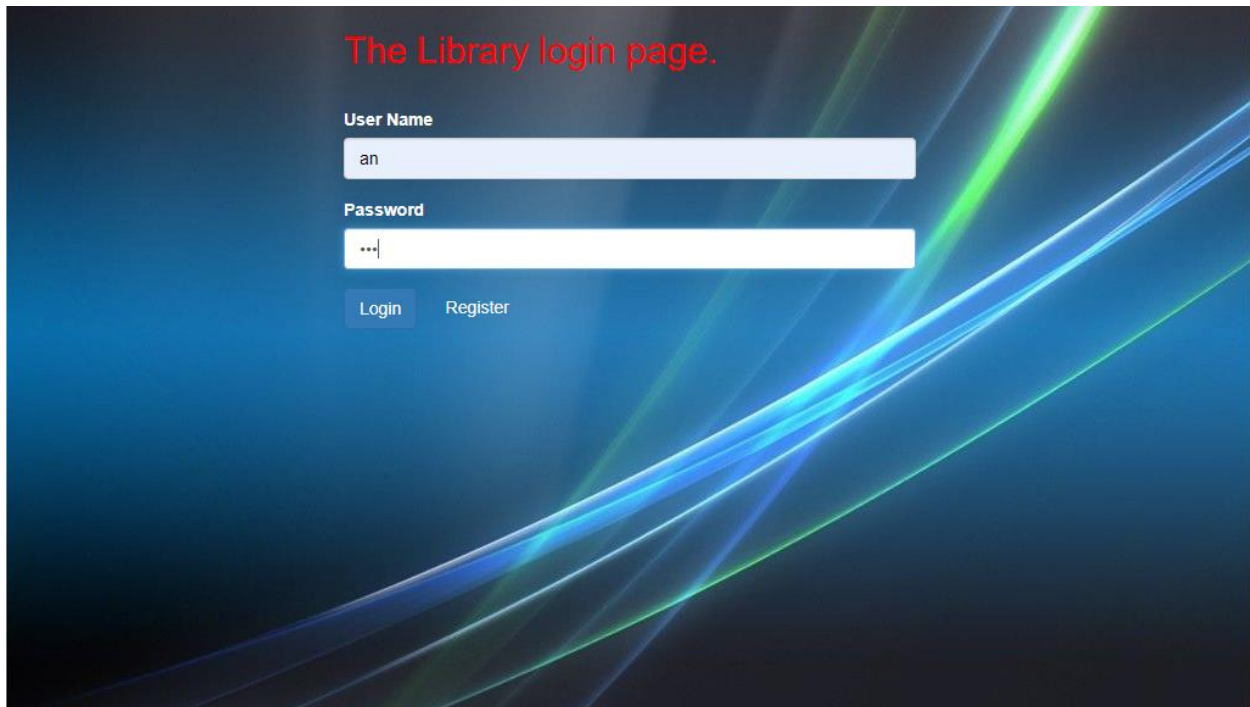


Figure 3-1 Log in page

```
<body>
  <div class="container">
    <h2>The Library login page.</h2> <br>
    <form method="post" action="userAction_login">
      <div class="form-group">
        <label for="userName">User Name</label>
        <input type="text" class="form-control" id="userName"
name="userName" >
      </div>
      <div class="form-group">
        <label for="password">Password</label>
        <input type="password" class="form-control"
id="password" name="password">
      </div>
```

```
<button type="submit" class="btn btn-primary">Login</button>  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <a href="register.jsp">Register</a>  
  
</form>  
  
${msg }  
  
</div>  
  
</body>
```

### 3.2 User Registration

This is a small project for LibrarySSH Management System. First of all the students have to registration for access the library where will have Username, password, real name, age, gender, and address. By the way if the student have already registered then he should log in by providing username, password.

## User register page.

**User Name**

**Password**

**Confirm Password**

Figure 3-2 User Register page

```

<body>
  <div class="container">
    <h2>User register page.</h2> <br>
    <form method="post" action="userAction_register">
      <div class="form-group">
        <label for="userName">User Name</label>
        <input type="text" class="form-control" id="userName"
name="userName" >
      </div>
      <div class="form-group">
        <label for="password">Password</label>
        <input type="password" class="form-control "
id="password" name="password">
      </div>
      <div class="form-group">
        <label for="repeatPassword">Confirm Password</label>
        <input type="password" class="form-control "
id="repeatPassword" name="repeatPassword">
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
    ${msg }
  </div>
</body>

```

### 3.3 Home Page

A library is organized for use and maintained by a public body, an institution, a corporation, or a private individual. Public and institutional collections and services may be intended for use by people who choose not to—or cannot afford

to—purchase an extensive collection themselves, who need material no individual can reasonably be expected to have, or who require professional assistance with their research. In addition to providing materials, libraries also provide the services of librarians who are experts at finding and organizing information and at interpreting information needs. Libraries often provide quiet areas for studying, and they also often offer common areas to facilitate group study and collaboration. Libraries often provide public facilities for access to their electronic resources and the Internet. Modern libraries are increasingly being redefined as places to get unrestricted access to information in many formats and from many sources. They are extending services beyond the physical walls of a building, by providing material accessible by electronic means, and by providing the assistance of librarians in navigating and analyzing very large amounts of information with a variety of digital resources. Libraries are increasingly becoming community hubs where programs are delivered and people engage in lifelong learning.

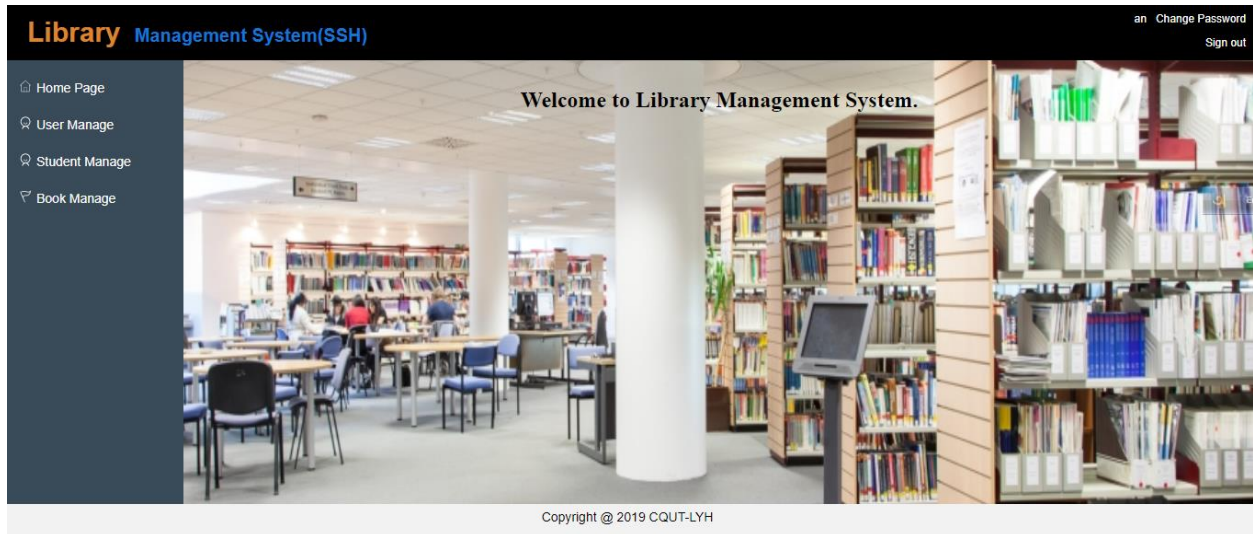


Figure 3-2 User Register page

### 3.4 User Manage

#### 3.4.1 User List

The User List are administration pages in the User and User talk namespaces that are useful for organizing and aiding the work users do on Library Management System. As well as facilitating interaction and sharing between users. User pages are mainly for interpersonal discussion, notices, testing and drafts and, if desired, limited autobiographical and personal content. User pages are available to Library Management System users personally for purposes compatible with the Library Management System project and acceptable to the community.

UserList Manage / User List

KeyWord:  search add User

Id	UserName	Password	
1	an	123	<button>edit</button> <button>delete</button>
2	an	123	<button>edit</button> <button>delete</button>
3	suzana	9450	<button>edit</button> <button>delete</button>
4	moyna	12345	<button>edit</button> <button>delete</button>
5	shakira	623423	<button>edit</button> <button>delete</button>
6	anower	anower33	<button>edit</button> <button>delete</button>

Figure 3-4-1: User List

```

<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">UserList Manage
    </a></li>
      <li class="breadcrumb-item active" aria-current="page">User
List</li>
    </ol>
  </nav>
  <div class="container">
    KeyWord:<input type="text" id="keyword" name="keyword" />
    <button class="btn btn-info" onclick="search();">search</button>
    <button class="btn btn-success" id="add_hw"
onclick="add();">add User</button>
  </div>
  <div class="container">
    <table id="infoTable" class="table table-striped table-
bordered table-condensed table-hover datatable">

```



```

<thead>
  <tr>
    <th scope="col">Id</th>
    <th scope="col">UserName</th>
    <th scope="col">Password</th>
  </thead>
  <tbody>
    <s:iterator var="item" status="st" value="#list">
      <tr>
        <td><s:property value="#item.Id"/></td>
        <td><s:property value="#item.UserName"/></td>
        <td><s:property value="#item.Password"/></td>
        <td>
          <button class="btn btn-xs btn-info"
onclick="edit('<s:property value="#item.Id"/>')">edit</button>
          <button class="btn btn-xs btn-danger"
onclick="del('<s:property value="#item.Id"/>');">delete</button>
        </td>
      </tr>
    </s:iterator>
  </tbody>
</table>
</div>
<script type="text/javascript">
  function search(){
    var keyWord = $('#keyword').val();
    window.location.href = "userAction_query?keyWord=" +
keyWord;
  }

  function del(id){

```

```

        if(confirm("Are you sure to delete the record")){
            window.location.href = "userAction_delete?id=" +
id;
        }
    }

    function add(){
        window.location.href = "userAction_showAdd";
    }

    function edit(id){
        window.location.href = "userEdit.jsp?id=" + id;
    }
</script>
</body>

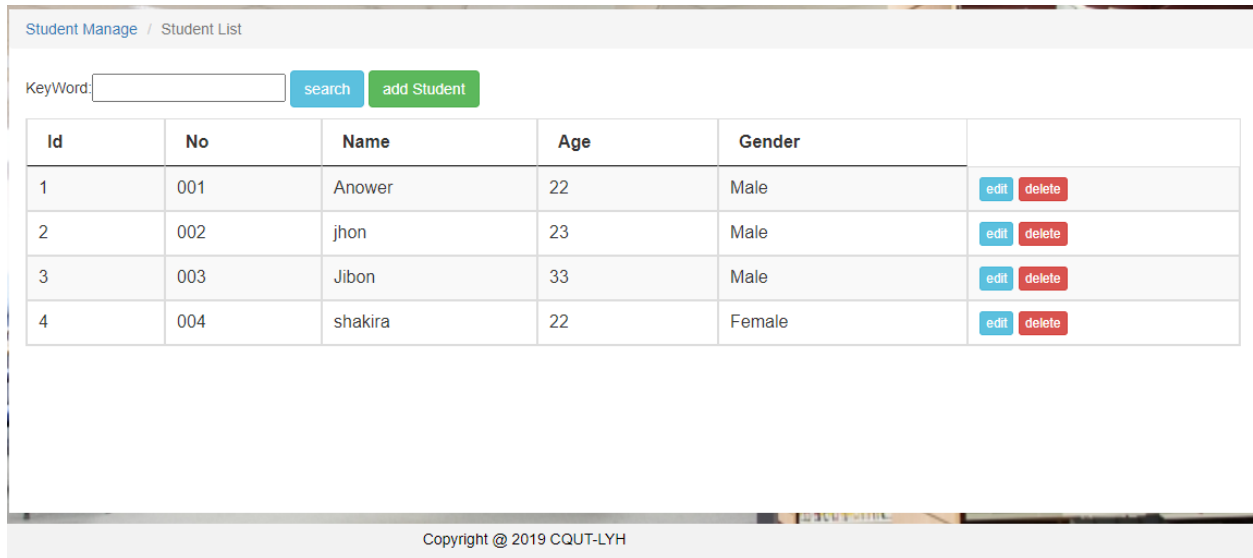
```

### 3.5 Student Mange

#### 3.5.1 Student List

A student is primarily a person enrolled in a school or other educational institution who attends classes in a course to attain the appropriate level of mastery of a subject under the guidance of an instructor and who devotes time outside class to do whatever activities the instructor assigns that are necessary either for class preparation or to submit evidence of progress towards that mastery. In the broader sense, a student is anyone who applies themselves to the intensive intellectual

engagement with some matter necessary to master it as part of some practical affair in which such mastery is basic or decisive.



Id	No	Name	Age	Gender	
1	001	Anower	22	Male	<a href="#">edit</a> <a href="#">delete</a>
2	002	jhon	23	Male	<a href="#">edit</a> <a href="#">delete</a>
3	003	Jibon	33	Male	<a href="#">edit</a> <a href="#">delete</a>
4	004	shakira	22	Female	<a href="#">edit</a> <a href="#">delete</a>

Copyright @ 2019 CQUT-LYH

Figure 3-5-1: Student List

```
<script type="text/javascript">
    function search(){
        var keyWord = $('#keyword').val();
        window.location.href = "studentAction_query?keyWord="
+ keyWord;
    }

    function del(id){
        if(confirm("Are you sure to delete the record?")){
            window.location.href = "studentAction_delete?id=" +id;
        }
    }

    function add(){
        window.location.href = "studentAdd.jsp";
    }

    function edit(id){
        window.location.href = "studentEdit.jsp?id="+id;
    }
</script>
```

### 3.5.1.1 Add Student

The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database. In the add student function the user can add student manually.

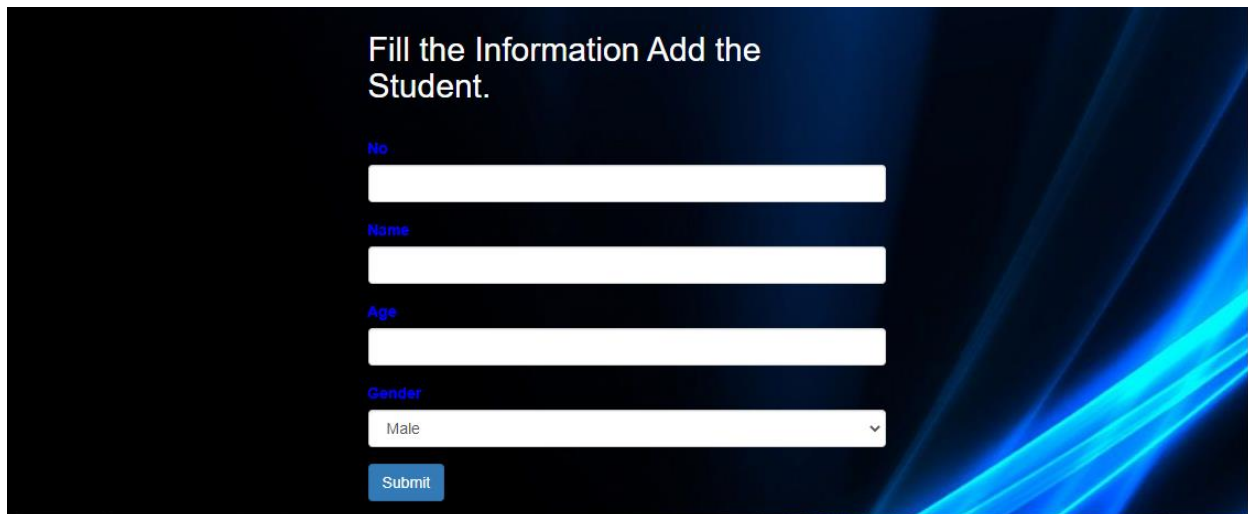


Figure 3-5-1-1: Add Student

```
<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">Student Manage
    </a></li>
```

```

        <li class="breadcrumb-item" aria-current="page">Student
List</li>
        <li class="breadcrumb-item active" aria-current="page">Add
Student</li>
    </ol>
</nav>
<div class="container">
    <h2>Fill the Information Add the Student.</h2> <br>
    <form method="post" action="studentAction_add"
enctype="multipart/form-data">

        <div class="form-group">
            <label for="No">No</label>
            <input type="text" class="form-control" id="No"
name="no">
        </div>

        <div class="form-group">
            <label for="Name">Name</label>
            <input type="text" class="form-control" id="Name"
name="name">
        </div>

        <div class="form-group">
            <label for="Age">Age</label>
            <input type="text" class="form-control" id="Age"
name="age">
        </div>

        <div class="form-group">
            <label for="Gender">Gender</label>

```

```

<select class="form-control" id="Gender" name="gender">
  <option value="Male">Male</option>
  <option value="Female">Female</option>
</select>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
</body>

```

### 3.5.1.2 Edit Student

When students edit in the Library, they show the world what they know, and they show themselves that they know. They're making a statement. They've learned and they tell the world: "I know this." That's the difference between writing for an assignment and writing for an audience. Students make the leap from passive learning to an active expression of knowledge. They rephrase and revise their understanding as they work.

The screenshot displays the 'Edit Student' interface of a Library Management System. On the left, a sidebar lists navigation options: Home Page, User Manage, Student Manage, Student List, and Book Manage. The main area features a form with the title 'This is edit Student page.' The form includes input fields for 'Id' (containing '1'), 'No', 'Name', and 'Age'. A 'Gender' dropdown menu is set to 'Male'. A blue 'Submit' button is at the bottom of the form. The footer indicates 'Copyright @ 2019 CQUT-LYH'.

Figure 3-5-1-2: Edit Student Information

```

<body>
    <%
        String id = request.getParameter("id");
    %>
    <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="#">Student Manage
        </a></li>
            <li class="breadcrumb-item" aria-current="page">Student
List</li>
            <li class="breadcrumb-item active" aria-current="page">Edit
Student</li>
        </ol>
    </nav>
    <div class="container">
        <h2>This is edit Student page.</h2> <br>
        <form method="post" action="studentAction_edit"
enctype="multipart/form-data">
            <div class="form-group">
                <label for="id">Id</label>
                <input type="text" class="form-control" id="id"
name="id" value="<%=id %>" readonly="readonly">
            </div>
            <div class="form-group">
                <label for="No">No</label>
                <input type="text" class="form-control" id="No"
name="no">
            </div>

            <div class="form-group">
                <label for="Name">Name</label>
                <input type="text" class="form-control" id="Name"
name="name">
            </div>

            <div class="form-group">
                <label for="Age">Age</label>
                <input type="text" class="form-control" id="Age"
name="age">
            </div>

```

```

    <div class="form-group">
      <label for="Gender">Gender</label>
      <select class="form-control" id="Gender" name="gender">
        <option value="Male">Male</option>
        <option value="Female">Female</option>
      </select>
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</div>

</body>

```

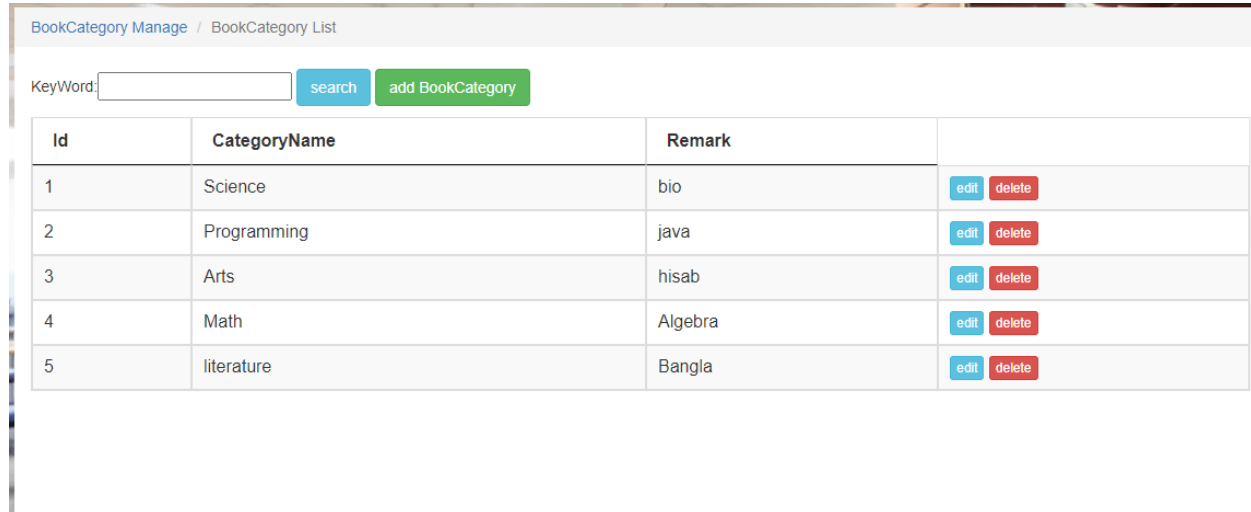
## 3.6 Book Manage

### 3.6.1 Book Category List

This is a category for different types of books, as distinct and separate from the Category: Books by genre. Types of books are defined by the format, method of manufacture, or appearance of the books. Genre refers to the content of the writing in the book. This classification system for Library books is based on Books Outline of knowledge. If you want guidance categorizing a book or subcategory, please refer to that outline. All books from the Book: namespace (i.e. community books) should be placed under at least one of these categories. Do not categorize books from the User. Genres of books simply specify what categories or sub-categories a book belongs to. If you are a regular book reader, you probably already know about some book genres that you have read or that you love reading. In this post, you will learn



about the two main types of books and their sub-categories which we commonly refer to as genres of books or book genres.



Id	CategoryName	Remark	
1	Science	bio	<a href="#">edit</a> <a href="#">delete</a>
2	Programming	java	<a href="#">edit</a> <a href="#">delete</a>
3	Arts	hisab	<a href="#">edit</a> <a href="#">delete</a>
4	Math	Algebra	<a href="#">edit</a> <a href="#">delete</a>
5	literature	Bangla	<a href="#">edit</a> <a href="#">delete</a>

Figure 3-6-1: Book Category List

```
<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">BookList Manage
    </a></li>
      <li class="breadcrumb-item active" aria-
current="page">BookList List</li>
    </ol>
  </nav>
  <div class="container">
    Keyword:<input type="text" id="keyword" name="keyword" />
    <button class="btn btn-info" onclick="search();">search</button>
    <button class="btn btn-success" id="add_hw"
onclick="add();">add Book</button>
  </div>
  <div class="container">
    <table id="infoTable" class="table table-striped table-
bordered table-condensed table-hover datatable">
      <thead>
        <tr>
          <th scope="col">Id</th>
```

```

        <th scope="col">BookName</th>
        <th scope="col">PicturePath</th>
        <th scope="col">Picture</th>
        <th scope="col">CategoryId</th>
        <th scope="col">Description</th>
        <th scope="col">Remark</th>
    </thead>
    <tbody>
    <s:iterator var="item" status="st" value="#list">
        <tr>
            <td><s:property value="#item.Id"/></td>
            <td><s:property value="#item.BookName"/></td>
            <td>"></td>
            <td><s:property value="#item.Picture"/></td>
            <td><s:property
value="#item.Category.CategoryName"/></td>
            <td><s:property value="#item.Description"/></td>
            <td><s:property value="#item.Remark"/></td>
            <td>
                <button class="btn btn-xs btn-info"
onclick="edit('<s:property value="#item.Id"/>')">edit</button>
                <button class="btn btn-xs btn-danger"
onclick="del('<s:property value="#item.Id"/>');">delete</button>
            </td>
        </tr>
    </s:iterator>
    </tbody>
</table>
</div>
<script type="text/javascript">
    function search(){
        var keyWord = $('#keyword').val();
        window.location.href = "bookAction_query?keyWord=" +
keyWord;
    }

    function del(id){
        if(confirm("Are you sure to delete the record?")){

```

```

        window.location.href = "bookAction_delete?id=" +
id;
    }
}

function add(){
    window.location.href = "bookAction_showAdd";
}

function edit(id){
    window.location.href = "bookAction_showEdit?id=" + id;
}
</script>
</body>

```

#### 3.6.1.1 Add Book Category

Categories are normally found at the bottom of an article page. Clicking a category name brings up a category page listing the articles (or other pages) that have been added to that particular category. There may also be a section listing the *subcategories* of that category. The sub categorization feature makes it possible to organize categories into tree-like structures to aid navigation. The term *category* does refer to both the title of a category page—the *category page name*—and the category itself. Keeping this in mind while reading about categorization, plus learning a category page layout is a worthwhile investment in research techniques. (See also the search box parameter "in category".) The layout of a category page is mostly text, but see about displaying category trees below. So

the feature Add Book Category, May be in the library has been collected new Book Category in this time add the Book Category and upload the book.

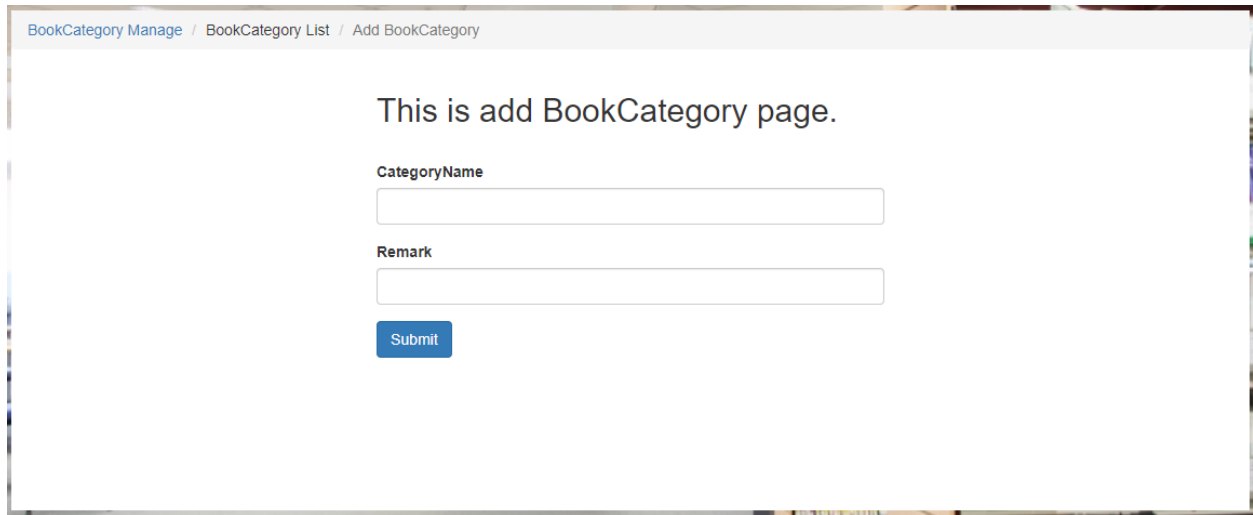


Figure 3-6-1-1: Add Book Category

```
<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">BookCategory Manage
    </a></li>
      <li class="breadcrumb-item" aria-current="page">BookCategory
List</li>
      <li class="breadcrumb-item active" aria-current="page">Add
BookCategory</li>
    </ol>
  </nav>
  <div class="container">
    <h2>This is add BookCategory page.</h2> <br>
    <form method="post" action="bookCategoryAction_add">
      <div class="form-group">
        <label for="CategoryName">CategoryName</label>
        <input type="text" class="form-control"
id="CategoryName" name="categoryName" >
      </div>
```

```

        <div class="form-group">
            <label for="Remark">Remark</label>
            <input type="text" class="form-control" id="Remark"
name="remark">
        </div>

        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
</div>

</body>

```

### 3.6.1.2 Edit Book Category

This is a category for different types of books, as distinct and separate from the Category: Books by genre. Types of books are defined by the format, method of manufacture, or appearance of the books. Genre refers to the content of the writing in the book. This classification system for Library books is based on Books Outline of knowledge. If you want guidance categorizing a book or subcategory, please refer to that outline. Maybe the collection of book is out of stock all books are borrowed that time need to edit the category that there is no more this type of book this is Edit book category.

BookCategory Manage / BookCategory List / Edit BookCategory

This is edit BookCategory page.

Id

CategoryName

Remark

Figure 3-6-1-2: Edit Book Category

```

<body>
    <%
        String id = request.getParameter("id");
    %>
    <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="#">BookCategory Manage
            </a></li>
            <li class="breadcrumb-item" aria-current="page">BookCategory
List</li>
            <li class="breadcrumb-item active" aria-current="page">Edit
BookCategory</li>
        </ol>
    </nav>
    <div class="container">
        <h2>This is edit BookCategory page.</h2> <br>
        <form method="post" action="bookCategoryAction_edit">
            <div class="form-group">
                <label for="Id">Id</label>
                <input type="text" class="form-control" id="Id"
name="id" value="<%=id %>" readonly="readonly">
            </div>

```

```

    <div class="form-group">
        <label for="CategoryName">CategoryName</label>
        <input type="text" class="form-control"
id="CategoryName" name="categoryName" >
    </div>
    <div class="form-group">
        <label for="Remark">Remark</label>
        <input type="text" class="form-control" id="Remark"
name="remark">
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>

</body>

```

### 3.6.2 Book List

Many publishers have lists of 100 best books, defined by their own criteria. This article enumerates some lists of "100 best" books for which there are fuller articles. Among them, Science Fiction: The 100 Best Novels (Xanadu, 1985) and Modern Fantasy: The 100 Best Novels (Grafton, 1988) are collections of 100 short essays by a single author, David Pringle, with moderately long critical introductory chapters also by Pringle. For publisher Xanadu, Science Fiction was the first of four "100 Best" books published from 1985 to 1988. The sequels covered crime and mystery, horror, and fantasy. so the Book list is the best way to describe the list of book.

BookList Manage / BookList List

KeyWord:



Id	BookName	PicturePath	Picture	CategoryId	Description	Remark	
1	Java Programming		upload\java-programming.jpg	Programming	Core Java tutorial: This tutorial will help you learn Java Programming in a simple and effective manner so you won't face any difficulty learning Java.	java	<input type="button" value="edit"/> <input type="button" value="delete"/>
2	Math Algebra		upload\math.jpg	Math	Product details Paperback Publisher: (March 1, 2017) ASIN: B0791BKTHW Shipping Information: View shipping rates and policies Customer Reviews: Be the first to write a review	math	<input type="button" value="edit"/> <input type="button" value="delete"/>

Figure 3-6-2: Book List

```

<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">BookList Manage
    </a></li>
      <li class="breadcrumb-item active" aria-
current="page">BookList List</li>
    </ol>
  </nav>
  <div class="container">
    Keyword:<input type="text" id="keyword" name="keyword" />
    <button class="btn btn-info" onclick="search();">search</button>
    <button class="btn btn-success" id="add_hw"
onclick="add();">add Book</button>
  </div>
  <div class="container">
    <table id="infoTable" class="table table-striped table-
bordered table-condensed table-hover datatable">
      <thead>
        <tr>
          <th scope="col">Id</th>
          <th scope="col">BookName</th>
          <th scope="col">PicturePath</th>

```



```

        <th scope="col">Picture</th>
        <th scope="col">CategoryId</th>
        <th scope="col">Description</th>
        <th scope="col">Remark</th>
    </thead>
    <tbody>
    <s:iterator var="item" status="st" value="#list">
        <tr>
            <td><s:property value="#item.Id"/></td>
            <td><s:property value="#item.BookName"/></td>
            <td>"></td>
            <td><s:property value="#item.Picture"/></td>
            <td><s:property
value="#item.Category.CategoryName"/></td>
            <td><s:property value="#item.Description"/></td>
            <td><s:property value="#item.Remark"/></td>
            <td>
                <button class="btn btn-xs btn-info"
onclick="edit('<s:property value="#item.Id"/>')">edit</button>
                <button class="btn btn-xs btn-danger"
onclick="del('<s:property value="#item.Id"/>');">delete</button>
            </td>
        </tr>
    </s:iterator>
    </tbody>
</table>
</div>
<script type="text/javascript">
    function search(){
        var keyWord = $('#keyword').val();
        window.location.href = "bookAction_query?keyWord=" +
keyWord;
    }

    function del(id){
        if(confirm("Are you sure to delete the record?")){
            window.location.href = "bookAction_delete?id=" +
id;
        }
    }

```

```

    }

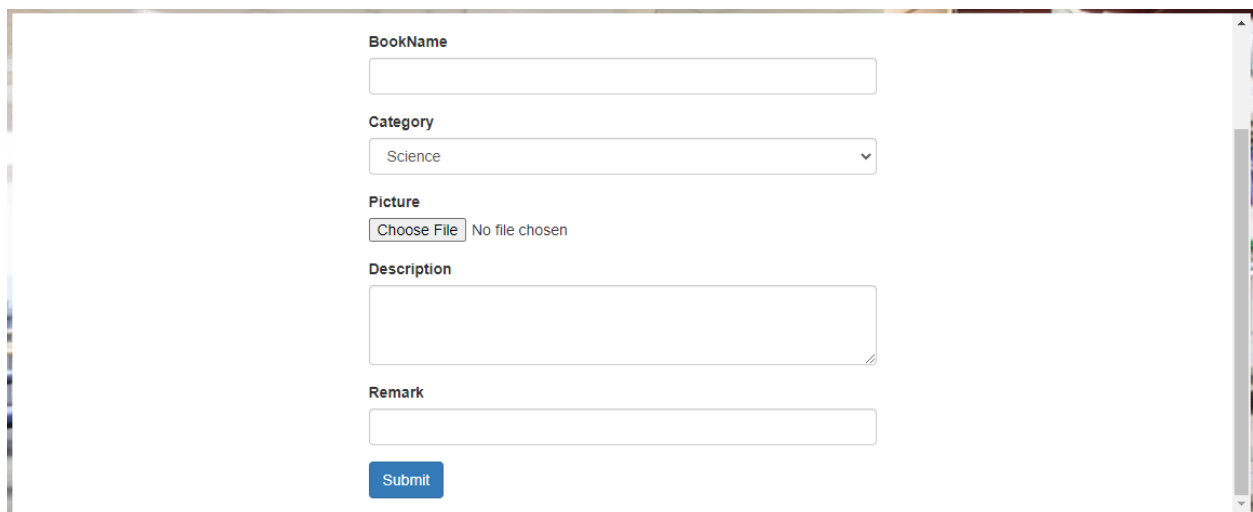
    function add(){
        window.location.href = "bookAction_showAdd";
    }

    function edit(id){
        window.location.href = "bookAction_showEdit?id=" + id;
    }
</script>

```

### 3.6.2.1 Add Book

Add Book is like Add Category we need add category when we buy many books as same type that time we need add a category but if we buy a specific collection of book that the category of books are already have in our library in this time we just need to add Book in particular category .



The screenshot shows a web form titled 'Add Book'. It has the following elements:

- BookName**: A text input field.
- Category**: A dropdown menu with 'Science' selected.
- Picture**: A file upload section with a 'Choose File' button and the text 'No file chosen'.
- Description**: A text area for entering details.
- Remark**: A text input field for additional notes.
- Submit**: A blue button to submit the form.

Figure 3-6-2-1: Add Book

```
<body>
```

```

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">BookCategory Manage
  </a></li>
    <li class="breadcrumb-item" aria-current="page">BookCategory
List</li>
    <li class="breadcrumb-item active" aria-current="page">Add
BookCategory</li>
  </ol>
</nav>
<div class="container">
<h2>This is add BookAdd page.</h2> <br>
<form method="post" action="bookAction_add"
enctype="multipart/form-data">
  <div class="form-group">
    <label for="BookName">BookName</label>
    <input type="text" class="form-control" id="bookName"
name="bookName" >
  </div>

  <div class="form-group">
    <label for="Category">Category</label>
    <select class="form-control" id="categoryId"
name="categoryId">
      <s:iterator var="item" status="st"
value="#listCategory">
        <option value='<s:property value="#item.Id"/>'>
          <s:property value="#item.CategoryName"/>
        </option>
      </s:iterator>
    </select>
  </div>

  <div class="form-group">
    <label for="Picture">Picture</label>
    <s:file name="file" label="upload file"></s:file>
  </div>

  <div class="form-group">
    <label for="Description">Description</label>

```

```

        <textarea class="form-control" id="description"
name="description" rows="3"></textarea>
    </div>

    <div class="form-group">
        <label for="Remark">Remark</label>
        <input type="text" class="form-control" id="Remark"
name="remark">
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>

</body>

```

#### 3.6.2.2 Edit Book

In the book publishing industry, editors may organize anthologies and other compilations, produce definitive editions of a classic author's works (scholarly editor), and organize and manage contributions to a multi-author book (symposium editor or volume editor). Obtaining manuscripts or recruiting authors is the role of an acquisitions editor or a commissioning editor in a publishing house.<sup>[4]</sup> Finding marketable ideas and presenting them to appropriate authors are the responsibilities of a sponsoring editor. Copy editors correct spelling, grammar and align writings to house style. Changes to the publishing industry since the 1980s have resulted in nearly all copy editing of book manuscripts being outsourced to freelance copy editors. Finally the main reason is editing book is changing anything the book.

The screenshot shows a web form for editing a book. The form is titled 'Edit Book' and is located within a browser window. The form fields are as follows:

- Id:** A text input field containing the value '1'.
- BookName:** A text input field.
- Category:** A dropdown menu with 'Science' selected.
- Picture:** A file upload button labeled 'Choose File' with the text 'No file chosen' next to it.
- Description:** A text area.
- Remark:** A text input field.

Figure 3-6-2-2: Edit Book

```
<body>
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">Book Manage
    </a></li>
      <li class="breadcrumb-item" aria-current="page">Book
List</li>
      <li class="breadcrumb-item active" aria-current="page">Edit
Book</li>
    </ol>
  </nav>
  <div class="container">
    <h2>This is edit Book page.</h2> <br>
    <form method="post" action="bookAction_edit"
enctype="multipart/form-data">
      <div class="form-group">
        <label for="id">Id</label>
        <input type="text" class="form-control" id="id"
name="id" value="{id}" readonly="readonly">
      </div>
      <div class="form-group">
        <label for="BookName">BookName</label>
```

```

        <input type="text" class="form-control" id="bookName"
name="bookName" >
    </div>

    <div class="form-group">
        <label for="Category">Category</label>
        <select class="form-control" id="categoryId"
name="categoryId">
            <s:iterator var="item" status="st"
value="#listCategory">
                <option value='<s:property value="#item.Id"/>'>
                    <s:property value="#item.CategoryName"/>
                </option>
            </s:iterator>
        </select>
    </div>

    <div class="form-group">
        <label for="Picture">Picture</label>
        <s:file name="file" label="upload file"></s:file>
    </div>

    <div class="form-group">
        <label for="Description">Description</label>
        <textarea class="form-control" id="description"
name="description" rows="3"></textarea>
    </div>

    <div class="form-group">
        <label for="Remark">Remark</label>
        <input type="text" class="form-control" id="Remark"
name="remark">
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>

</body>

```

### 3.6.3 Book Search

Online search is the process of interactively searching for and retrieving requested information via a computer from databases that are online. Interactive searches became possible in the 1980s with the advent of faster databases and smart terminals. In contrast, computerized batch searching was prevalent in the 1960s and 1970s. Today, searches through web search engines constitute the majority of online searches. From that time very place all of the people use the search option. In the online library the search option is the best option for find a specific book, specific category etc. The search option of find a specific book is very useful.

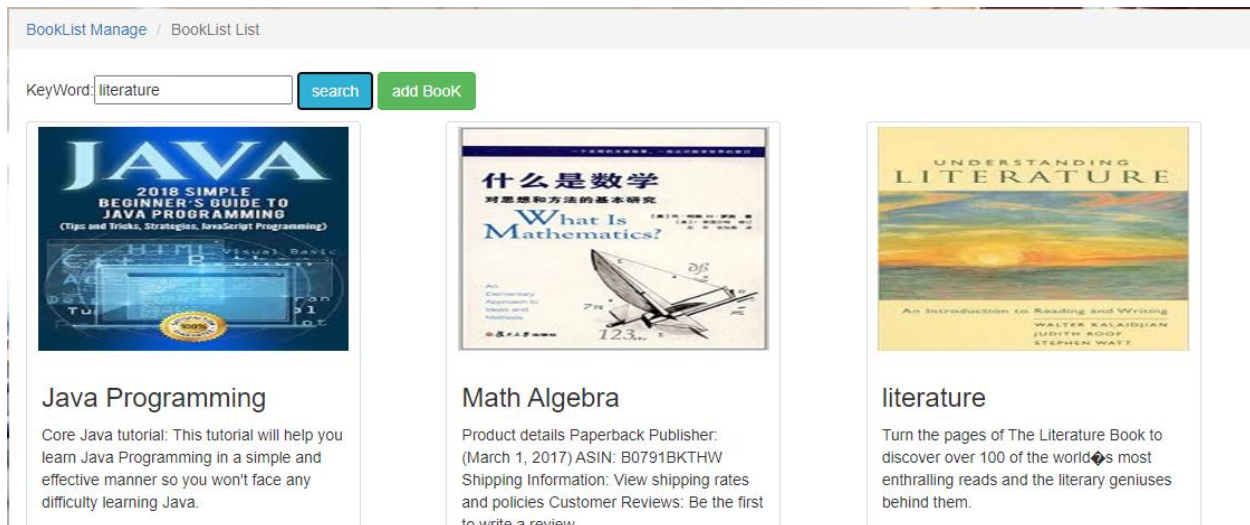


Figure 3-6-3: Search Book

```
function search(){  
    var keyWord = $('#keyword').val();  
    window.location.href = "bookAction_query?keyWord=" +  
keyWord;  
}
```

## 4 Summary

A LibrarySSH Management System (LMS) is a tool to help any libraries which are still using the old way to manage their library. The old way like searching for a book using manual work is hassle, fast report generation is not possible, information about issue/return of the books are not properly maintained, no central database can be created as information is not available in database. But by using the LMS, user can overcome all the problems mentioned above. This system can manage all the happenings of the library. Book transactions including book searching, availability of the book, details and appearance of the book, personal book borrowing history and etc. can be very easily handled by this system. This system is suitable for small to big libraries including medical and legal libraries, colleges, schools, universities, corporate houses and other academic resource centers. However, I would like to focus on LMS for colleges or universities.