L1 – ASSESSMENT

SCENARIO: Bookshop Management system

EMPLOYEE NAME: PRABHAKARAN K S EMPLOYEE NUMBER: 12229

W3H TECHNIQUE:

Bookshop Management system	
WHAT?	**HOW?**
1. What are the modules required? User module, admin module and book module are the modules required. 2. What are the fields of user module? Userid, Username, Userphone, Usermailid, Useraadhar. 3. What are the functionalities of user? CRUD operations of the books. 4. What are the fields of admin module? Adminid, Adminname, Admin phone, Adminmailid, Adminaadhar. 5. What are the functionalities of admin? CRUD operations of the books. 6. What are the fields of book module? Bookid, Bookname, Bookauthor,Bookdateofbuy	**1. User Module:** CRUD Operations: Method –1: Add/View/Update/Delete of the books can be done by placing the buttons of those operations. Method –2: Add/View/Update/Delete of the books can be done by placing the operations as the drop-down box. Method-3: Add/View/Update/Delete of the books can be done by having the textbox to type which operation is required for the user. **2. Admin Module:** CRUD Operations: Method –1: Add/View/Update/Delete of the books can be done by having the textbox to type which operation is required for the user. Method –2: Add/View/Update/Delete of the books can be done by placing the operations as the drop-down box Method-3: Add/View/Update/Delete of the books can be done by placing the buttons of that operations

1.User Module: CRUD Operations: Method –2: Add/View/Update/Delete of the books can be done by placing the operations as the drop-down box. (It is because the CRUD operations views in drop-down box means the view will be nice for the user in choosing the operations) **2.Admin Module:** CRUD Operations: Method –3: Add/View/Update/Delete of the books can be done by placing the buttons of that operations (It is because the CRUD operations by the admin is the root of this system. So, if it places like buttons, it will be easy to choose and it is time efficient method)	**1. User Module:** CRUD Operations: Method-1: The CRUD operations of the books if we placed as buttons means it will not be attractive to the user. Method-3: Manually typing the CRUD operations which is required by the user is not a time efficient way **2. Admin Module:** CRUD Operations: Method-1: Typing the CRUD operations in the manual way means it is not a time efficient way. Method-2: There is a huge chance of error can be arising due to the choosing of the CRUD operations which is placed in the drop-down box.
WHY?	**WHY NOT?**

ALGORITHM:

Step-1: Start

Step-2: Login to the system

Step-3: View the list of books available

Step-4: Search the book by its name

Step-5: Click the book

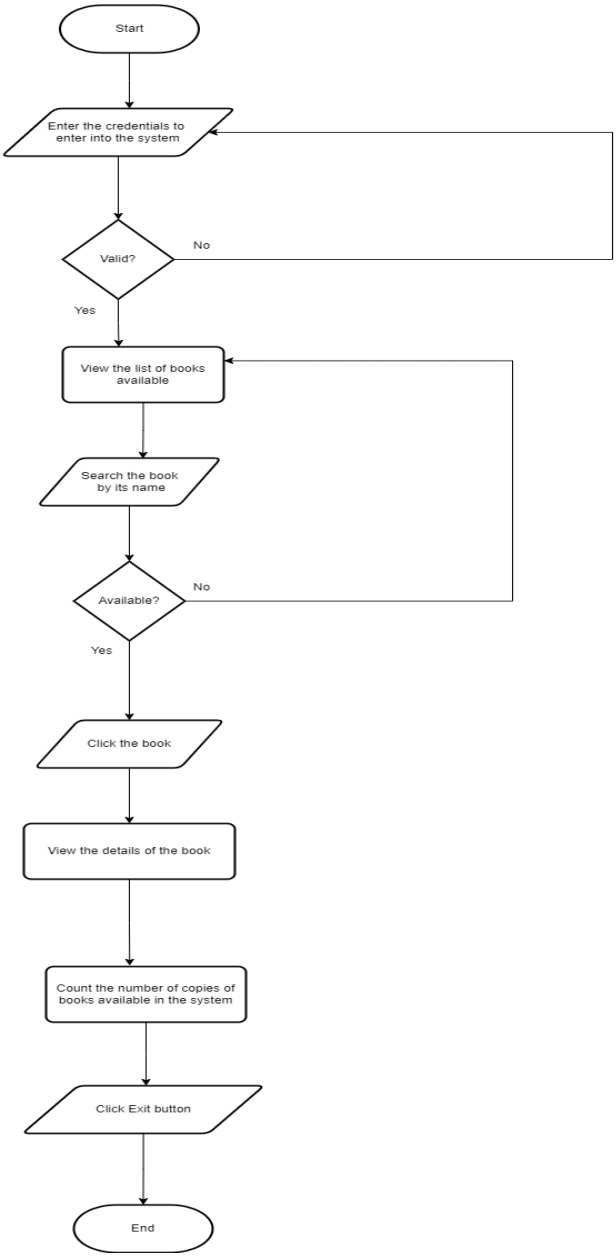
Step-6: View the details of the book

Step-7: Count the number of copies of the books in the system

Step-8: Click exit button.

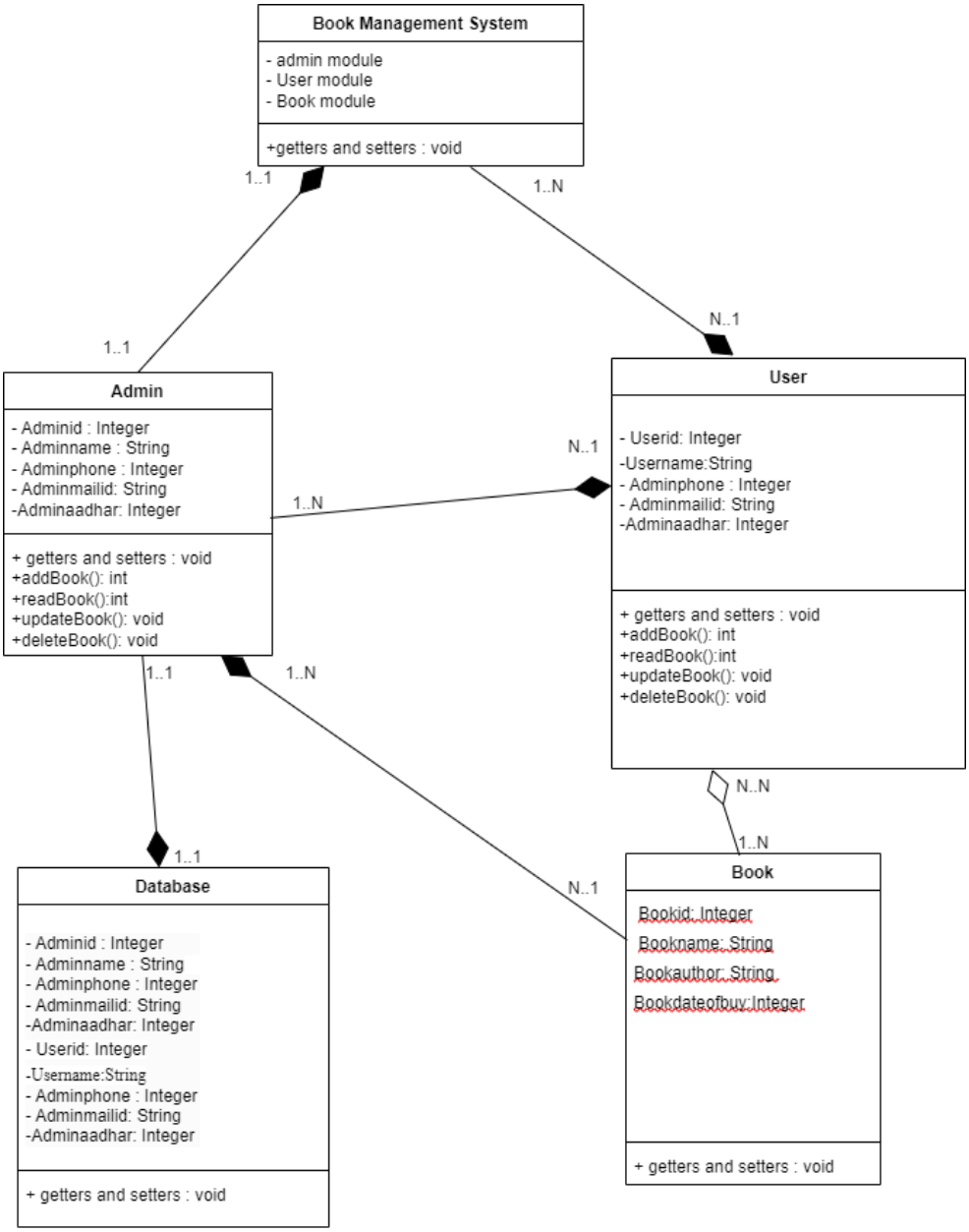
Step-9: End

FLOWCHART

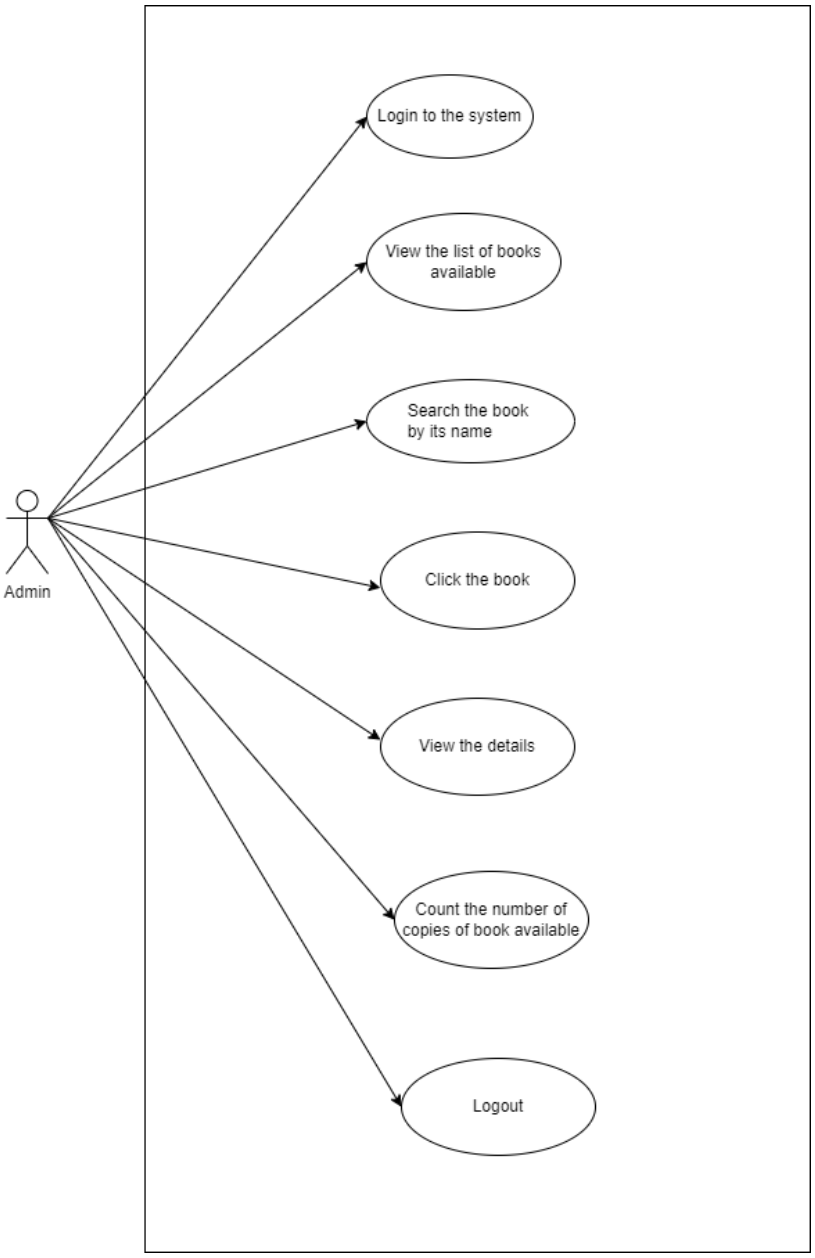

Start
Enter the credentials to enter into the system
Valid?
No
Yes
View the list of books available
Search the book by its name
Available?
No
Yes
Click the book
View the details of the book
Count the number of copies of books available in the system
Click Exit button
End

UML DIAGRAMS

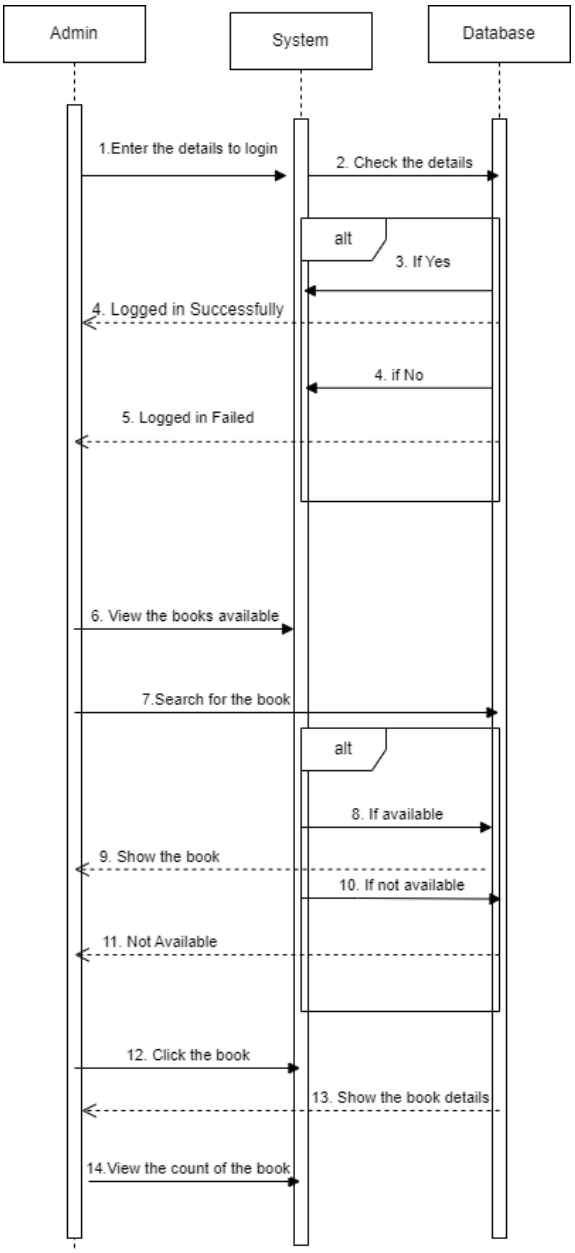
CLASS DIAGRAM



USECASE DIAGRAM:



SEQUENCE DIAGRAM:


Admin
System
Database
1.Enter the details to login
2. Check the details
alt
3. If Yes
4. Logged in Successfully
4. if No
5. Logged in Failed
6. View the books available
7.Search for the book
alt
8. If available
9. Show the book
10. If not available
11. Not Available
12. Click the book
13. Show the book details
14.View the count of the book

```
1  create database bms_db;
2  use bms_db;
3  create table books_details(Bookid int primary key, Bookname varchar(50), Bookauthor varchar(70), Bookdateofbuy varchar(50));
4  insert into  books_details(Bookid, Bookname, Bookauthor,Bookdateofbuy) values(1,' Thirukkural' ,'Thiruvalluvar','12th November'),(2,'Ponniyin sel
5  select * from books_details;
6
7
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Bookid	Bookname	Bookauthor	Bookdateofbuy
1	Thirukkural	Thiruvalluvar	12th November
2	Ponniyin selvan	Kalki	1st December
3	3 Wickets	Chetan Bagath	3rd July
4	rajaparvai	beem	2nd june
6	ven	syg	as
NULL	NULL	NULL	NULL

Result Grid | Form Editor

Books.bean

```
package com.bms.bean;

public class books {
int Bookid;
String Bookname;
String Bookauthor;
String Bookdateofbuy;
public books() {
super();
// TODO Auto-generated constructor stub
}

public int getBookid() {
return Bookid;
}

public void setBookid(int bookid) {
Bookid = bookid;
}

public String getBookname() {
```

```
return Bookname;
}

public void setBookname(String bookname) {
Bookname = bookname;
}

public String getBookauthor() {
return Bookauthor;
}

public void setBookauthor(String bookauthor) {
Bookauthor = bookauthor;
}

public String getBookdateofbuy() {
return Bookdateofbuy;
}

public void setBookdateofbuy(String bookdateofbuy) {
Bookdateofbuy = bookdateofbuy;
}

public books(int bookid, String bookname, String bookauthor, String
bookdateofbuy) {
super();
Bookid = bookid;
Bookname = bookname;
Bookauthor = bookauthor;
Bookdateofbuy = bookdateofbuy;
}


}
```

Controller

```
package com.bms.controller;

import java.io.IOException;
```

```
import java.io.PrintWriter;
import java.sql.SQLException;
import java.util.LinkedList;

import com.bms.DAO.bmsDAO;
import com.bms.bean.books;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class Controller extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
PrintWriter out = response.getWriter();
int n = 0;
String name = request.getParameter("action");
bmsDAO dao = new bmsDAO();
if (name.equalsIgnoreCase("Insert")) {
books b = new books(Integer.parseInt(request.getParameter("id")),
request.getParameter("name"),
request.getParameter("author"), request.getParameter("buys"));

try {
n = dao.doInsert(b);
} catch (SQLException s) {
s.printStackTrace();
}
if (n > 0) {
RequestDispatcher rd =
request.getRequestDispatcher("InsertedSuccess.jsp");
rd.forward(request, response);

} else {
RequestDispatcher rd =
request.getRequestDispatcher("InsertedFailure.jsp");
rd.forward(request, response);
```

```
}
}

else if (name.equalsIgnoreCase("Delete")) {
int id1 = Integer.parseInt(request.getParameter("id"));
try {
n = dao.doDelete(id1);
} catch (SQLException s) {
s.printStackTrace();
}
if (n > 0) {
RequestDispatcher rd1 =
request.getRequestDispatcher("DeleteSuccess.jsp");
rd1.forward(request, response);

} else {
RequestDispatcher rd1 =
request.getRequestDispatcher("DeleteFailure.jsp");
rd1.forward(request, response);

}

}


else if (name.equalsIgnoreCase("Update")) {
int id = Integer.parseInt(request.getParameter("id"));
books e = new books(Integer.parseInt(request.getParameter("Bookid")),
request.getParameter("Bookname"),
request.getParameter("Bookauthor"),
request.getParameter("Bookdateofbuy"));
try {
n = dao.doUpdate(e);

} catch (SQLException s) {
s.printStackTrace();
}
if (n > 0) {
```

```
RequestDispatcher rd =
request.getRequestDispatcher("UpdateSuccess.jsp");
rd.forward(request, response);

} else {
RequestDispatcher rd =
request.getRequestDispatcher("UpdateFailure.jsp");
rd.forward(request, response);

}
}
else if (name.equals("Find")) {
int id = Integer.parseInt(request.getParameter("Bookid"));

books e1 = new books(Integer.parseInt(request.getParameter("Bookid")),
request.getParameter("Bookname"),
request.getParameter("Bookauthor"),
request.getParameter("Bookdateofbuy"));

try {
e1 = dao.doFind(id);
} catch (SQLException e) {
e.printStackTrace();
}
if (e1 != null) {
RequestDispatcher rd =
request.getRequestDispatcher("FindSuccess.jsp");
request.setAttribute("res", e1);
rd.forward(request, response);

} else {
RequestDispatcher rd =
request.getRequestDispatcher("FindFailure.jsp");
rd.forward(request, response);

}
}
else if (name.equals("View")) {
```

```
RequestDispatcher rd =
request.getRequestDispatcher("ViewSuccess.jsp");
LinkedList<books> list = new LinkedList<>();
try {
list = dao.viewAll();
} catch (SQLException e) {
e.printStackTrace();
}
request.setAttribute("list", list);
rd.forward(request, response);

}

}

}
```

DAO

```
package com.bms.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.LinkedList;

import com.bms.bean.books;
import com.bms.util.bmsUtil;

public class bmsDAO {


public LinkedList viewAll() throws SQLException {
Connection con1=bmsUtil.getConnection();
LinkedList<books> lists=new LinkedList<books>();
String sql="select * from books_details";
PreparedStatement ps=con1.prepareStatement(sql);
ResultSet rs=ps.executeQuery();
```

```
while(rs.next()) {
books b1=new
books(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4));
lists.add(b1);
}
con1.close();
return lists;
}
public int doInsert(books b) throws SQLException {
Connection con = bmsUtil.getConnection();
String sql = "Insert into books_details values(?,?,?,?)";
PreparedStatement ps = con.prepareStatement(sql);
ps.setInt(1, b.getBookid());
ps.setString(2, b.getBookname());
ps.setString(3, b.getBookauthor());
ps.setString(4, b.getBookdateofbuy());

int n = ps.executeUpdate();
con.close();

return n;

}
public int doUpdate(books e) throws SQLException {
Connection con = bmsUtil.getConnection();
String sql = "Update books_details set
Bookname=?,Bookauthor=?,Bookdateofbuy=? where Bookid=?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, e.getBookname());
ps.setString(2, e.getBookauthor());
ps.setString(3, e.getBookdateofbuy());
ps.setInt(4, e.getBookid());

int n = ps.executeUpdate();
con.close();
return n;

}
public int doDelete(int id) throws SQLException {
```

```
Connection con = bmsUtil.getConnection();
String sql = "Delete from books_details where Bookid=?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setInt(1, id);
int n = ps.executeUpdate();
con.close();
return n;

}
public books doFind(int id) throws SQLException {
Connection con1=bmsUtil.getConnection();
String sql="select * from books_details where Bookid=?";
PreparedStatement ps= con1.prepareStatement(sql);
ps.setInt(1, id);
ResultSet rs=ps.executeQuery();
books w=null;
while(rs.next())
{
w=new
books(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4));

}
System.out.println(w);
con1.close();
return w;

}




}

Util
package com.bms.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```

```
public class bmsUtil {

public static Connection getConnection() throws SQLException {
final String url = "jdbc:mysql://localhost:3306/bms_db";
final String user = "root";
final String pass = "Prabha123#@!";
Connection con = DriverManager.getConnection(url,user,pass);
return con;
}

}
```

Homepage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Book Management System</title>
<frameset rows="10%,90%">
<frame src="Top.jsp" class="topview" name="f1"></frame>
<frameset cols="20%,80%">
<frame src="Links.jsp" name="f2"></frame>
<frame src="Screen.jsp" name="f3"></frame>
</frameset>
</frameset>

</frameset>
</head>
<body>



</body>
</html>
```

Top.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">

<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
}

}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
<title>Insert title here</title>
</head>
```

```
<body>
<center>
<h1>Book Management System</h1>
</center>


</body>
</html>
```

Links.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Links</title>

<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
}

}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}
```

```
@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<a href="insert.jsp" target="f3"><button>Insert</button></a>
<br>
<br>
<a href="delete.jsp" target="f3"><button>Delete</button></a>
<br>
<br>
<a href="update.jsp" target="f3"><button>Update</button></a>
<br>
<br>
<a href="find.jsp" target="f3">
<button>Find</button>
</a>
<br>
<br>
<a href="View.jsp" target="f3"><button>View All</button></a>

</body>
</html>
```

SCreen.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Book Management System</title>

<style>
```

```
body {
margin: 0;
padding: 0;
width: 100%;
height:70%;
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
display: flex;
justify-content: center;
align-items: center;

animation:ani 3s ease-in 1 forwards;

}


@keyframes ani{
0%{
opacity:0;
}
50%{
color:red;
}
100%{

opacity:1;
}

}


.content-container {
text-align: center;
}

h1, img {
position: relative;
z-index: 1;
color: white;
```

```
text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
max-width: 90%;
max-height: 90%;
margin: 0 auto;
padding: 20px;
}

img {


box-shadow: 5px 5px 10px rgba(0, 0, 0, 0);
animation: rise 20s ease forwards;
height:540px;
width: 1500px;
}

</style>
</head>
<body>

<div class="content-container">
<h1>Welcome to the Book Management System</h1>
<img
src="https://www.studentprojects.live/wp-
content/uploads/2022/02/Library-Management-System.jpg"
alt="Lake Bled">
</div>

</body>
</html>
```

Insert.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert</title>
```

```
<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
}

}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
<script>
function insertvalidate(){
var id = document.forms["Inserting"]["id"].value;
var name = document.forms["Inserting"]["name"].value;
var author = document.forms["Inserting"]["author"].value;
var buys = document.forms["Inserting"]["buys"].value;
if(name == ""){
alert("please fill the id !!!!");
event.preventDefault();
}
```

```
</script>
</head>
<body>

<form action="Controller" method="get" name="Inserting"
onsubmit="return insertvalidate();">
<center>
<label>Book Id:</label><br> <br> <input type="number"
name="id"><br> <br> <label>Book Name:</label><br>
<br> <input type="text" name="name"><br> <br>
<label>Book Author</label><br> <br> <input type="text"
name="author"><br> <br> <label>Book Date
Of Buy</label><br> <br> <input type="text" name="buys"><br>
<br> <input type="submit" name="action" value="Insert">
</center>
</form>

</body>
</html>
```

Update.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>

<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
```

```
}

}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<form action = "Controller" method="get">
<center>
<label>Book Id:</label><br><br>
<input type="number" name="Bookid"><br><br>
<label>Book Name:</label><br><br>
<input type="text" name="Bookname"><br><br>
<label>Book Author:</label><br><br>
<input type="text" name="Bookauthor"><br><br>
<label>Book Date Of Buy:</label><br><br>
<input type="text" name="Bookdateofbuy"><br><br>
<input type="submit" name="action" value="Update">
</center>
</form>

</body>
</html>
```

Find.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action ="Controller" method="get">
<center>
<label>Enter the Book id to find:</label><br><br>
<input type="number" name="Bookid" ><br><br>
<input type="submit" name="action" value="Find">
</center>
</form>

</body>
</html>
```

Delete.jsp

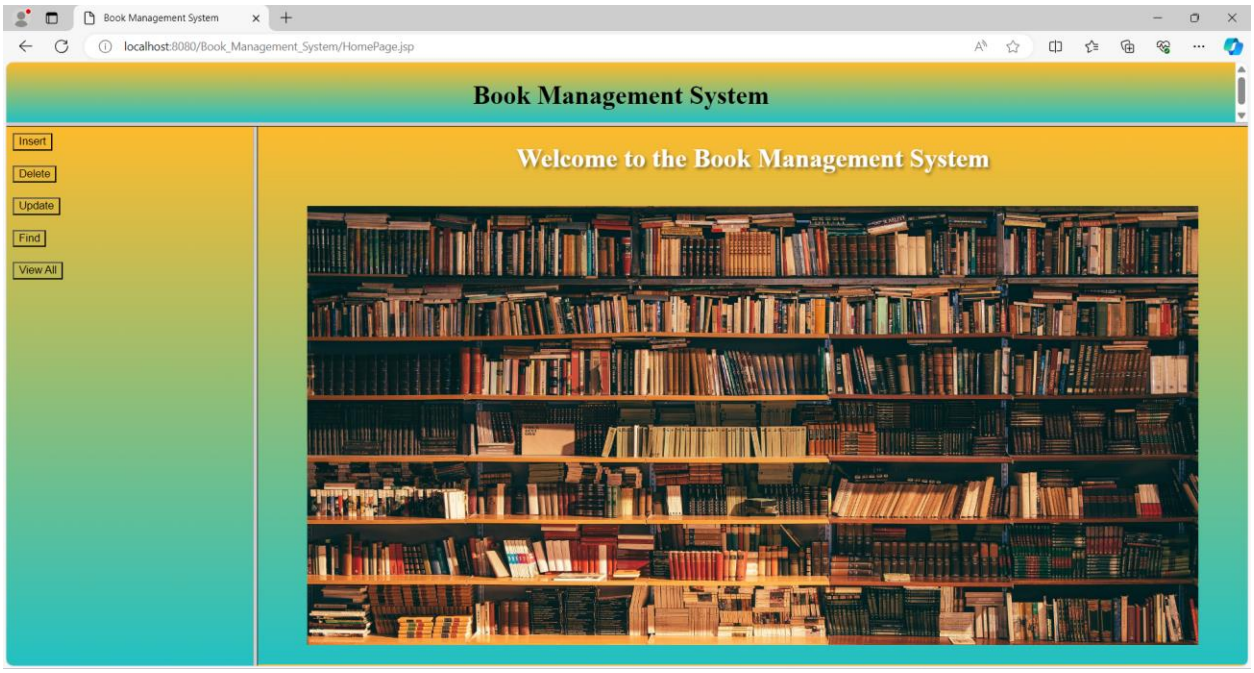
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>

<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
}
```

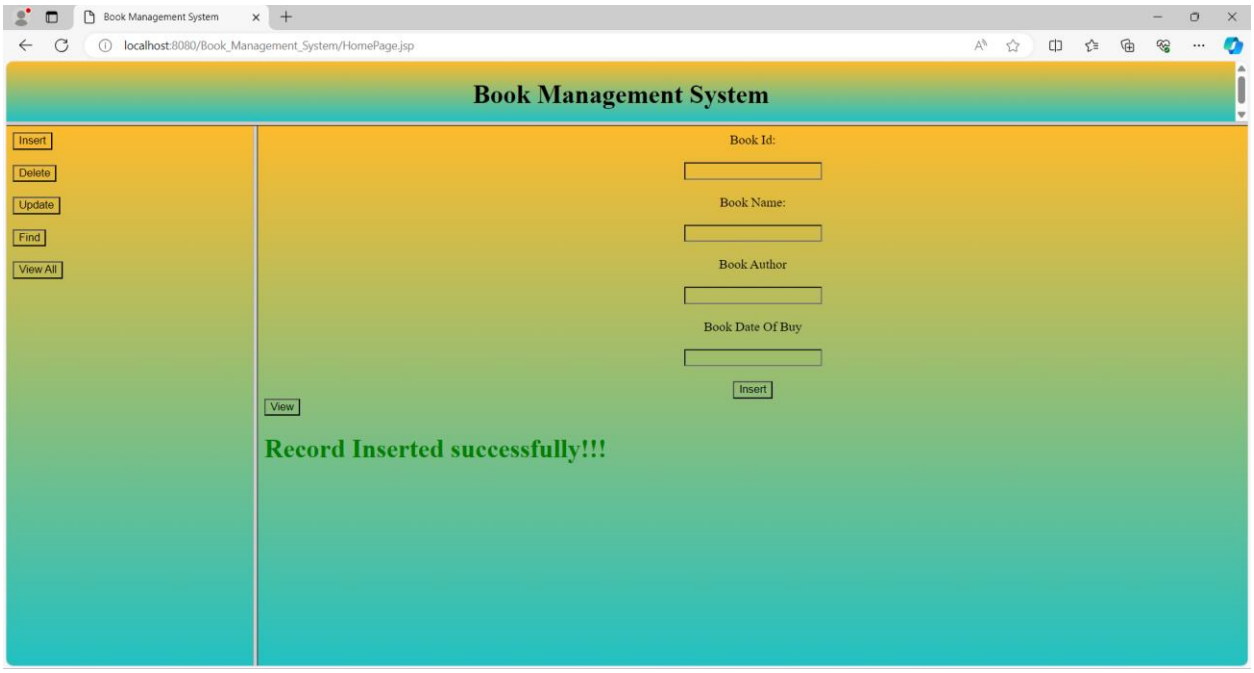
```
}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<form action = "Controller" method="get">
<center>

<label>Enter the Book id to delete:</label><br><br>
<input type="number" name="id" ><br><br>
<input type="submit" value="Delete" name="action">
</center>

</form>
</body>
</html>
```

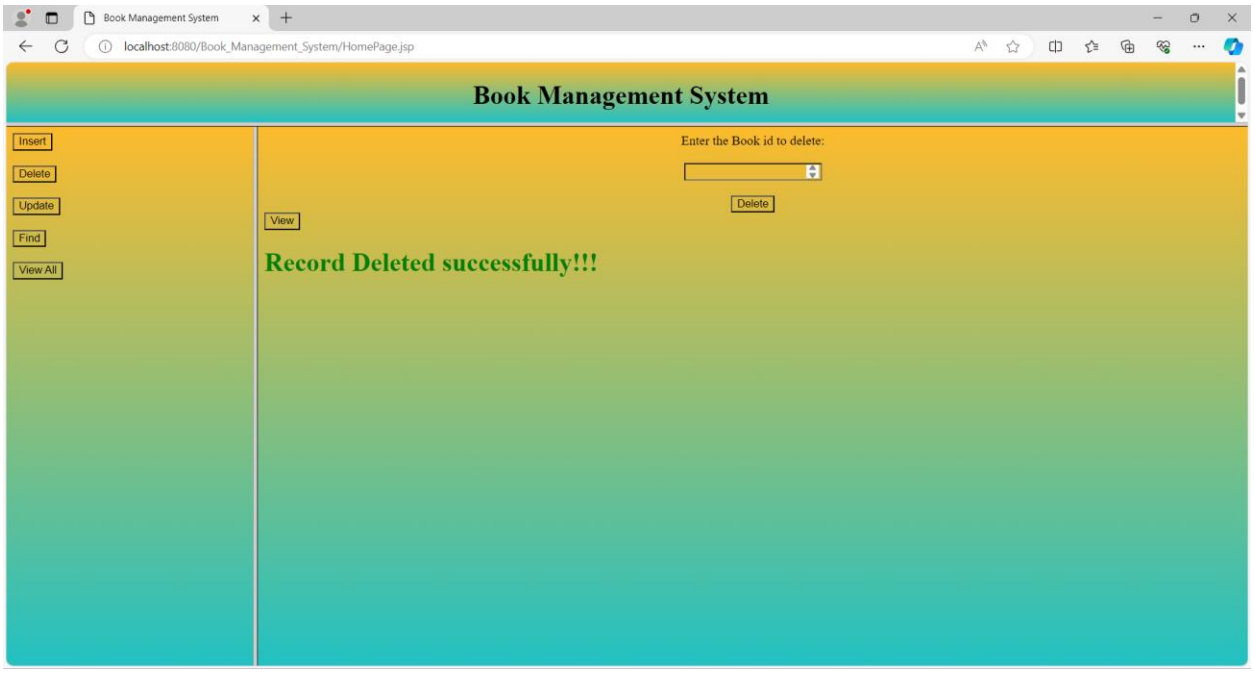
View.jsp

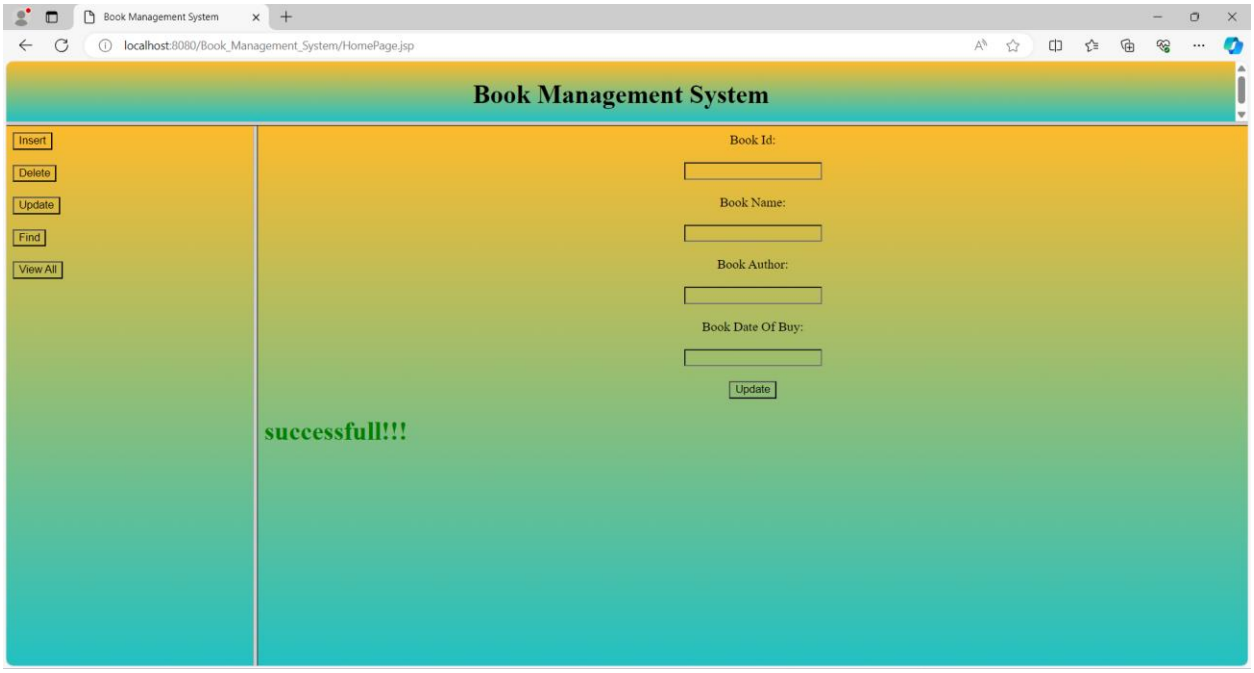
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
```

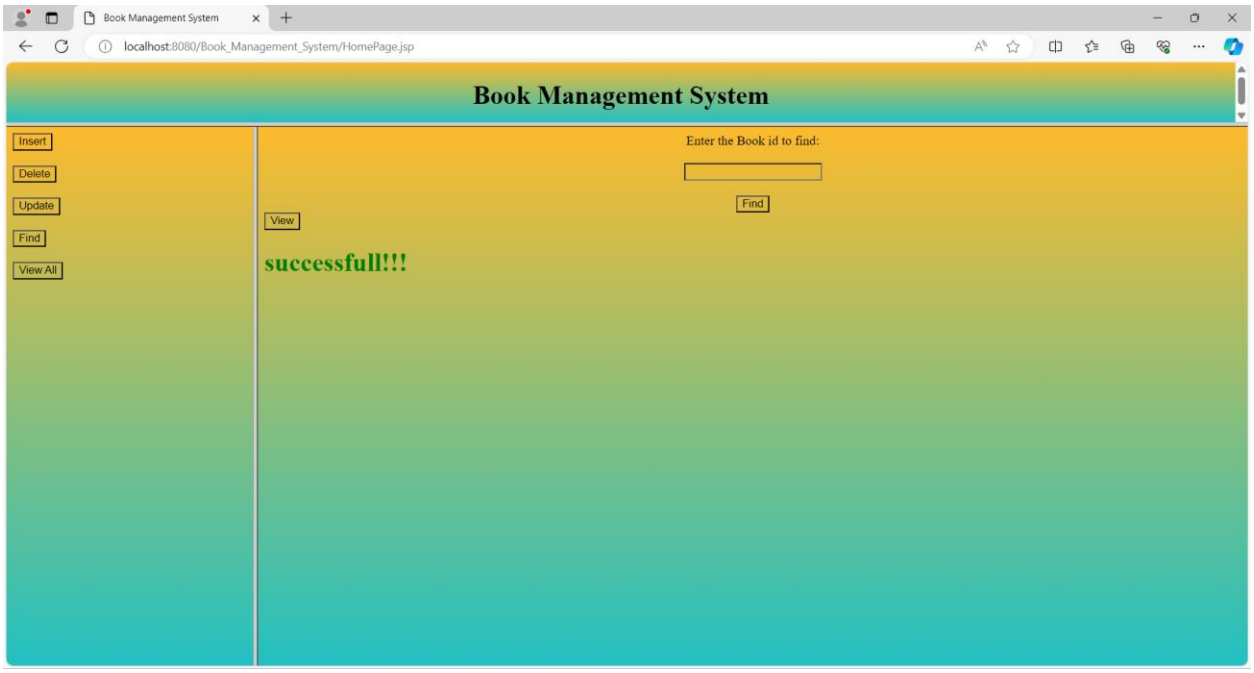
```
<style>


*{

background: rgb(34,193,195);
background: linear-gradient(0deg, rgba(34,193,195,1) 0%,
rgba(253,187,45,1) 100%);
background-attachment: fixed;
background-repeat: no-repeat;
}

}
button {
width: 35px;
height: 35px;
background: rgb(34,193,195);
animation-name: example;
animation-duration: 4s;
}

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<form action = "Controller" method="get" >
<input type="submit" name="action" value="View">
</form>

</body>
</html>
```

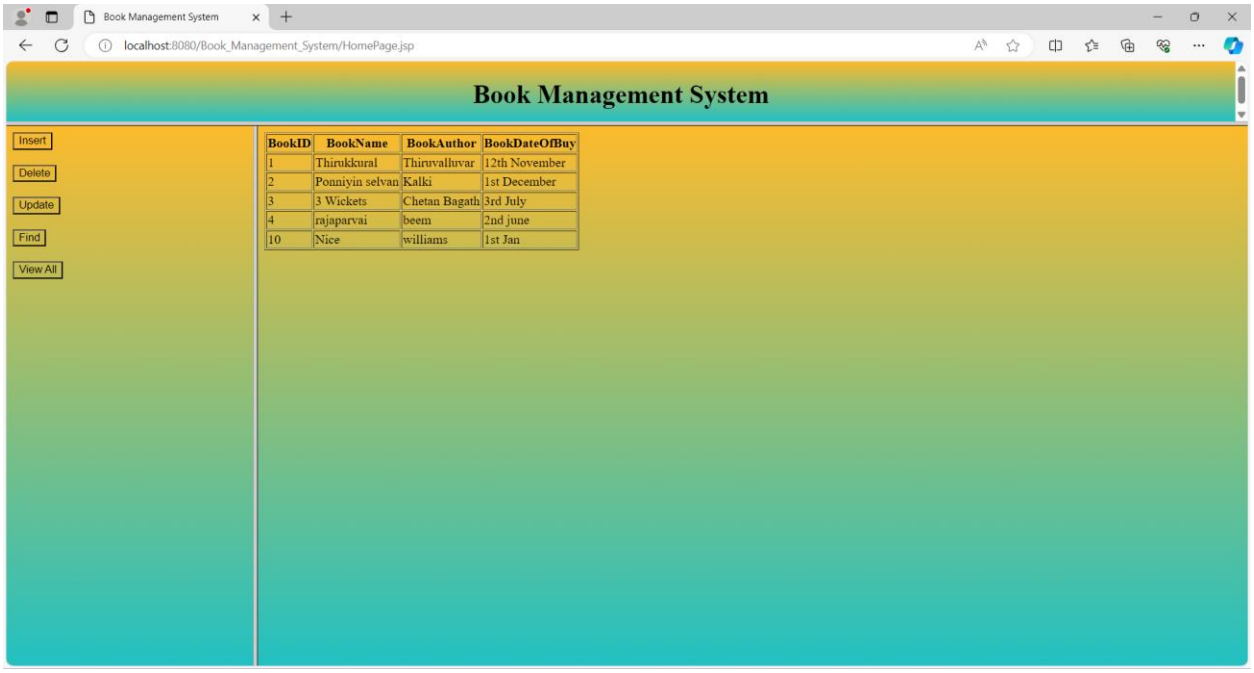

Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
Welcome to the Book Management System


Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
Book Id:
Book Name:
Book Author
Book Date Of Buy
Insert
View
Record Inserted successfully!!!


Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
Enter the Book id to delete:
Delete
View
Record Deleted successfully!!!


Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
Book Id:
Book Name:
Book Author:
Book Date Of Buy:
Update
successfull!!!


Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
Enter the Book id to find:
Find
View
successfull!!!


Book Management System
localhost:8080/Book_Management_System/HomePage.jsp
Book Management System
Insert
Delete
Update
Find
View All
BookID BookName BookAuthor BookDateOfBuy
1 Thirukkural Thiruvalluvar 12th November
2 Ponniyin selvan Kalki 1st December
3 3 Wickets Chetan Bagath 3rd July
4 rajaparvai beem 2nd june
10 Nice williams 1st Jan