A

Major Project On

E-Library SYSTEM

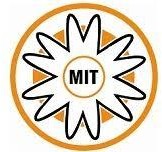
Submitted

In partial fulfilment

For the award of the Degree of

**Bachelor of Technology**

**in Department of COMPUTER SCIENCE Engineering**



MODI INSTITUTE OF TECHNOLOGY

Submitted To: Submitted By:

Mr. MONAJ AMERIA Ashish Kumar

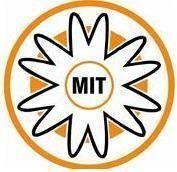
Designation:HOD Prafull Parashar

**Department of COMPUTER SCIENCE Engineering**

Modi Institute of Technology, Kota Rajasthan Technical University , Kota

**MODI INSTITUTE OF TECHNOLOGY, KOTA**

**Session 2020-2021**



**Certificate**

***This is to certify that the report entitled “E-Library System” has been***

***submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in “Computer Engineering” by following student of final year B.Tech. (ComputerEngineering).***

Ashish Kumar (17EMICS008)

Prafull Parashar (17EMICS034)

**Supervisor: HOD:**

Mr. Monaj Ameria Mr. Monaj Ameria

(CSE Department) (CSE Department)

**ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to all those people who have been directly and indirectly with me during the completion of this project/seminar/IT ACT seminar.

I pay thank to Manoj Ameria who has given guidance and a light to me during this training. His versatile knowledge about “title name” has eased me in the critical times during the span of this major project/seminar/IT ACT Seminar.

I acknowledge here out debt to those who contributed significantly to one or more steps. I take full responsibility for any remaining sins of omission and commission.

Ashish Kumar

Prafull Parashar

Btech. VIII Year

(ComputerScience Engineering)

**PREFACE**

Bachelor of Technology in Computer Science & Engineering is the Rajasthan Technical University course (Approved by AICTE) having duration of 4 years. As a prerequisite of the syllabus every student on this course has to make a report on Training in order to complete his studies successfully. And it is required to submit the report on the completion of it.

The main objective of this report is to create awareness regarding the application of theories in the practical world of Computer Science & Engineering and to give a practical exposure of the real world to the student.

I, therefore, submit this report on *“E-Library system”*, which was undertaken at MIT, KOTA. I feel great pleasure to present this report.

Certificate...................................................................................................i Acknowledgement..................................................................................... ii Preface.......................................................................................................iii

**Table of content :**

Objectives of the project:

1. Existing System:
2. Proposed System :
3. Project Modules:
4. Users of the System
5. Functional Requirements
6. Hardware configuration
7. Software Requirements/ Technologies
8. Project Architecture
9. Functional Diagram :
10. Tables Used :
    * 1. Ebook
      2. E\_Librarian
      3. E\_IssuedBook
11. JDBC Connection With DB
12. E-Library Project Algorithm/Code
13. Admin module Code;
14. Librarian Module code :
15. E-Libraray ScreenShot :
16. SYSTEM TESTING
17. Unit testing
18. integration testing
19. Test For the admin module
20. Test for Librarian login module
21. CONCLUSION & FUTURE SCOPE
22. REFERENCES

**Objectives of the project**:

The main objective of this project is to provide a complete automated Library by digitizing its each and every functionality. Starting from the book-keeping, issuing of books, fine generation, advance booking and report generation all will be accomplished under one single project. The project will increase the accuracy by making the whole issuing and returning of books speedier, manageable, less time consuming and more efficient. It will certainly reduce the manual work used in preparing reports and data retrieval..

**Existing System**:

The Existing system has a lot of redundancies and inconsistencies. A lot of work force is deployed to drive the library efficiently. All the data management is done over papers. It becomes very difficult to respond at once how many copies of a particular book are there or is the book available with the library or not. The Book issuing, returning and advance booking also involves ambiguity at many levels.

**Proposed System :**

The project will be a web based project with a database server responsible for maintaining every single detail of the Library. It has a very user friendly interface which can easily be operated by any non-technical person. It will make the Library management more efficient, responsive and less time consuming with high level of accuracy. Any person can inquire about the books with a single mouse click. Issuing of books, returning of books and fine generation will become simple and easy

**Project Modules:**

A library management software where admin can add/view/delete librarian and librarian can add/view books, issue, view issued books and return books.

**Users of the System**

Admin : Act as Super user. He have control on all the librarian user.

Librarian : Have the authority to control the Book details.

**Functional Requirements**

The Proposed software will be having the following modules:

1. **Admin**: Admin module will have the full control and permissions to add a new librarian, edit the existing librarian or delete any of the librarian. He will also be responsible for addition and modification of any book entry. Admin can view the librarian detail. Also this module contain login/logout feature.
2. **Librarian**: The librarian can see his information added by the admin. Each and every librarian will be identified by a unique id. He can only edit his information other .he can add a new book details, edit the existing detail of book, view the book details, and delete a book details. He can also view issued book and return book. This module also have login/logout feature.

**Hardware configuration**

**Servers**: We are using Tomcat server to run our application. Also we are using Oracle as database server.

**Terminals**: These are simply the desktops where the GUI will be running.

Processor Pentium –IV and above

RAM 128 MD SD EAM

Monitor 800\* 600 resolution

Hard disk 10 GB

Floppy drive 1.44 MB

CD drive LG 52X

Key board Standard 102 keys

Mouse 3 buttons, scrollable

**Software Requirements/ Technologies**

Operating system Windows XP Professional/ 2000

Coding Language J2EE and JSP

Frontend (GUI) HTML5/CSS3, BootStrap

Backend Oracle 8i

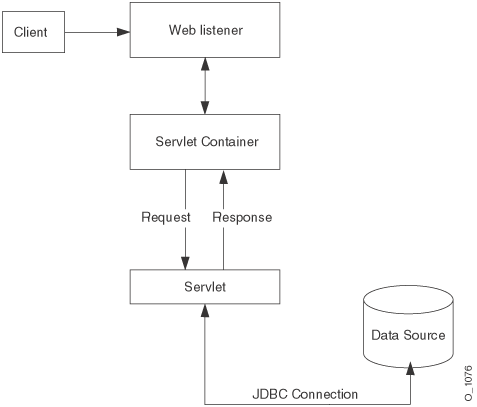
Server Apache Tomcat 5.0

**Project Architecture :**

Execution of Servlets basically involves six basic steps:

1. The clients send the request to the webserver.
2. The weserver receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generates the response in the form of output.
5. The servlet sends the response back to the webserver.
6. The web server sends the response back to the client and the client browser displays it on the screen

Below is the client server architecture of our project .



Client send a request to sever. Web listener listen the request and send a HTTp request to Servlet container , who talks to servlet. Then in the servlet , we execute the business operation as well as it communicatw with data source for JDBC operation . then a HTTP response will get generated and this response will be send to client .

**Functional Diagram :**

Login System :

( Admin login )

( Librarian login )

|  |
| --- |
| Add librarian |
| View librarian |
| Edit librarian |
| Delete Librarian |
| Logout |
| Login |

Admin Home :

Librarian home

|  |
| --- |
| Issue a Book |
| Add a Book details |
| View Book details |
| Delete book details |
| Edit Book details |
| View issued book |
| View return Book |
| Login/logout |

**Tables Used :**

We have created below table in our application

1. Ebook : Ebook table has the column as callno, name, author, publisher, quantity, issued.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CallNo | Name | Author | publisher | Quantity | Isssued |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**CREATE** **TABLE** "E\_BOOK"

( "CALLNO" VARCHAR2(4000),

"NAME" VARCHAR2(4000),

"AUTHOR" VARCHAR2(4000),

"PUBLISHER" VARCHAR2(4000),

"QUANTITY" NUMBER,

"ISSUED" NUMBER,

**CONSTRAINT** "E\_BOOK\_PK" **PRIMARY** **KEY** ("CALLNO") ENABLE

)

1. E\_Librarian : Elibrarian table has the column as id, name, passcode, email, mobile.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Name | Passcode | Email | Mobile |
|  |  |  |  |  |
|  |  |  |  |  |

**CREATE** **TABLE** "E\_LIBRARIAN"

( "ID" NUMBER,

"NAME" VARCHAR2(4000),

"PASSWORD" VARCHAR2(4000),

"EMAIL" VARCHAR2(4000),

"MOBILE" NUMBER,

**CONSTRAINT** "E\_LIBRARIAN\_PK" **PRIMARY** **KEY** ("ID") ENABLE

)

/

Note: ID must be generated through sequence **in** E\_LIBRARIAN **table**.

1. E\_IssuedBook : E\_IssuedBook table has the column as callNo,StudentId, Student name, Student Date, issueddate, and returnstatus.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| callNo | StudentName | StudentDate | issueDate | Studentid | Status |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**CREATE** **TABLE** "E\_ISSUEBOOK"

( "CALLNO" VARCHAR2(4000) **NOT** **NULL** ENABLE,

"STUDENTID" VARCHAR2(4000) **NOT** **NULL** ENABLE,

"STUDENTNAME" VARCHAR2(4000),

"STUDENTMOBILE" NUMBER,

"ISSUEDDATE" **DATE**,

"RETURNSTATUS" VARCHAR2(4000)

)

/

**JDBC Connection With DB** :

To connect java application with the oracle database, we have used below steps.

1. Driver class: The driver class for the oracle database is oracle.jdbc.driver.OracleDriver.
2. Connection URL: The connection URL for the oracle10G database is jdbc:oracle:thin:@localhost:1521:xe where jdbc is the API, oracle is the database, thin is the driver, localhost is the server name on which oracle is running, 1521 is the port number and XE is the Oracle service name
3. Username: The default username for the oracle database is system.
4. Password: It is the password given by the user at the time of installing the oracle database.

Connection class:

**public** **class** DB {

**public** **static** Connection getCon(){

Connection con=**null**;

**try**{

Class.*forName*("oracle.jdbc.driver.OracleDriver");

con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe","system","system");

}**catch**(Exception e){System.***out***.println(e);}

**return** con;

}

}

**E-Library Project Algorithm/Code**

**Admin module Code;**

1. Add Librarian method:
2. **public** **static** **int** save(LibrarianBean bean){
3. **int** status=0;
4. **try**{
5. Connection con=DB.*getCon*();
6. PreparedStatement ps=con.prepareStatement("insert into e\_librarian(name,email,password,mobile) values(?,?,?,?)");
7. ps.setString(1,bean.getName());
8. ps.setString(2,bean.getEmail());
9. ps.setString(3,bean.getPassword());
10. ps.setLong(4,bean.getMobile());
11. status=ps.executeUpdate();
12. con.close();
14. }**catch**(Exception e){System.***out***.println(e);}
16. **return** status;

}

1. Update Librarian

**public** **static** **int** update(LibrarianBean bean){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("update e\_librarian set name=?,email=?,password=?,mobile=? where id=?");

ps.setString(1,bean.getName());

ps.setString(2,bean.getEmail());

ps.setString(3,bean.getPassword());

ps.setLong(4,bean.getMobile());

ps.setInt(5,bean.getId());

status=ps.executeUpdate();

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. View All Librarian details

**public** **static** List<LibrarianBean> view(){

List<LibrarianBean> list=**new** ArrayList<LibrarianBean>();

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_librarian");

ResultSet rs=ps.executeQuery();

**while**(rs.next()){

LibrarianBean bean=**new** LibrarianBean();

bean.setId(rs.getInt("id"));

bean.setName(rs.getString("name"));

bean.setEmail(rs.getString("email"));

bean.setPassword(rs.getString("password"));

bean.setMobile(rs.getLong("mobile"));

list.add(bean);

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** list;

}

1. View A particular Librarian Details

**public** **static** LibrarianBean viewById(**int** id){

LibrarianBean bean=**new** LibrarianBean();

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_librarian where id=?");

ps.setInt(1,id);

ResultSet rs=ps.executeQuery();

**if**(rs.next()){

bean.setId(rs.getInt(1));

bean.setName(rs.getString(2));

bean.setPassword(rs.getString(3));

bean.setEmail(rs.getString(4));

bean.setMobile(rs.getLong(5));

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** bean;

}

1. Delete A particular Librarian details

**public** **static** **int** delete(**int** id){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("delete from e\_librarian where id=?");

ps.setInt(1,id);

status=ps.executeUpdate();

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. Authenicate Feature

**public** **static** **boolean** authenticate(String email,String password){

**boolean** status=**false**;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_librarian where email=? and password=?");

ps.setString(1,email);

ps.setString(2,password);

ResultSet rs=ps.executeQuery();

**if**(rs.next()){

status=**true**;

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

**Librarian Module code :**

1 Add Book details

**public** **static** **int** save(BookBean bean){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("insert into e\_book values(?,?,?,?,?,?)");

ps.setString(1,bean.getCallno());

ps.setString(2,bean.getName());

ps.setString(3,bean.getAuthor());

ps.setString(4,bean.getPublisher());

ps.setInt(5,bean.getQuantity());

ps.setInt(6,0);

status=ps.executeUpdate();

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. View Books Details

**public** **static** List<BookBean> view(){

List<BookBean> list=**new** ArrayList<BookBean>();

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_book");

ResultSet rs=ps.executeQuery();

**while**(rs.next()){

BookBean bean=**new** BookBean();

bean.setCallno(rs.getString("callno"));

bean.setName(rs.getString("name"));

bean.setAuthor(rs.getString("author"));

bean.setPublisher(rs.getString("publisher"));

bean.setQuantity(rs.getInt("quantity"));

bean.setIssued(rs.getInt("issued"));

list.add(bean);

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** list;

}

1. Delete Book details

**public** **static** **int** delete(String callno){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("delete from e\_book where callno=?");

ps.setString(1,callno);

status=ps.executeUpdate();

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. issued books

**public** **static** **int** getIssued(String callno){

**int** issued=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_book where callno=?");

ps.setString(1,callno);

ResultSet rs=ps.executeQuery();

**if**(rs.next()){

issued=rs.getInt("issued");

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** issued;

}

1. check Issued Book

**public** **static** **boolean** checkIssue(String callno){

**boolean** status=**false**;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_book where callno=? and quantity>issued");

ps.setString(1,callno);

ResultSet rs=ps.executeQuery();

**if**(rs.next()){

status=**true**;

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. Issued a book

**public** **static** **int** issueBook(IssueBookBean bean){

String callno=bean.getCallno();

**boolean** checkstatus=*checkIssue*(callno);

System.***out***.println("Check status: "+checkstatus);

**if**(checkstatus){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("insert into e\_issuebook values(?,?,?,?,?,?)");

ps.setString(1,bean.getCallno());

ps.setString(2,bean.getStudentid());

ps.setString(3,bean.getStudentname());

ps.setLong(4,bean.getStudentmobile());

java.sql.Date currentDate=**new** java.sql.Date(System.*currentTimeMillis*());

ps.setDate(5,currentDate);

ps.setString(6,"no");

status=ps.executeUpdate();

**if**(status>0){

PreparedStatement ps2=con.prepareStatement("update e\_book set issued=? where callno=?");

ps2.setInt(1,*getIssued*(callno)+1);

ps2.setString(2,callno);

status=ps2.executeUpdate();

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}**else**{

**return** 0;

}

}

1. Return a Book

**public** **static** **int** returnBook(String callno,**int** studentid){

**int** status=0;

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("update e\_issuebook set returnstatus='yes' where callno=? and studentid=?");

ps.setString(1,callno);

ps.setInt(2,studentid);

status=ps.executeUpdate();

**if**(status>0){

PreparedStatement ps2=con.prepareStatement("update e\_book set issued=? where callno=?");

ps2.setInt(1,*getIssued*(callno)-1);

ps2.setString(2,callno);

status=ps2.executeUpdate();

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

1. View List Of issued Book details

**public** **static** List<IssueBookBean> viewIssuedBooks(){

List<IssueBookBean> list=**new** ArrayList<IssueBookBean>();

**try**{

Connection con=DB.*getCon*();

PreparedStatement ps=con.prepareStatement("select \* from e\_issuebook order by issueddate desc");

ResultSet rs=ps.executeQuery();

**while**(rs.next()){

IssueBookBean bean=**new** IssueBookBean();

bean.setCallno(rs.getString("callno"));

bean.setStudentid(rs.getString("studentid"));

bean.setStudentname(rs.getString("studentname"));

bean.setStudentmobile(rs.getLong("studentmobile"));

bean.setIssueddate(rs.getDate("issueddate"));

bean.setReturnstatus(rs.getString("returnstatus"));

list.add(bean);

}

con.close();

}**catch**(Exception e){System.***out***.println(e);}

**return** list;

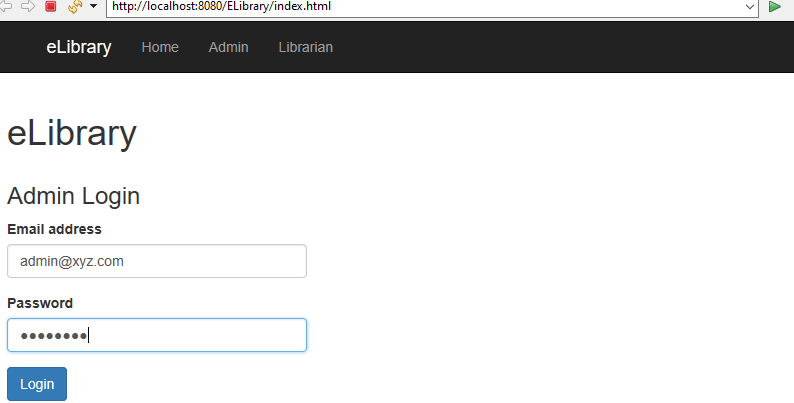
}

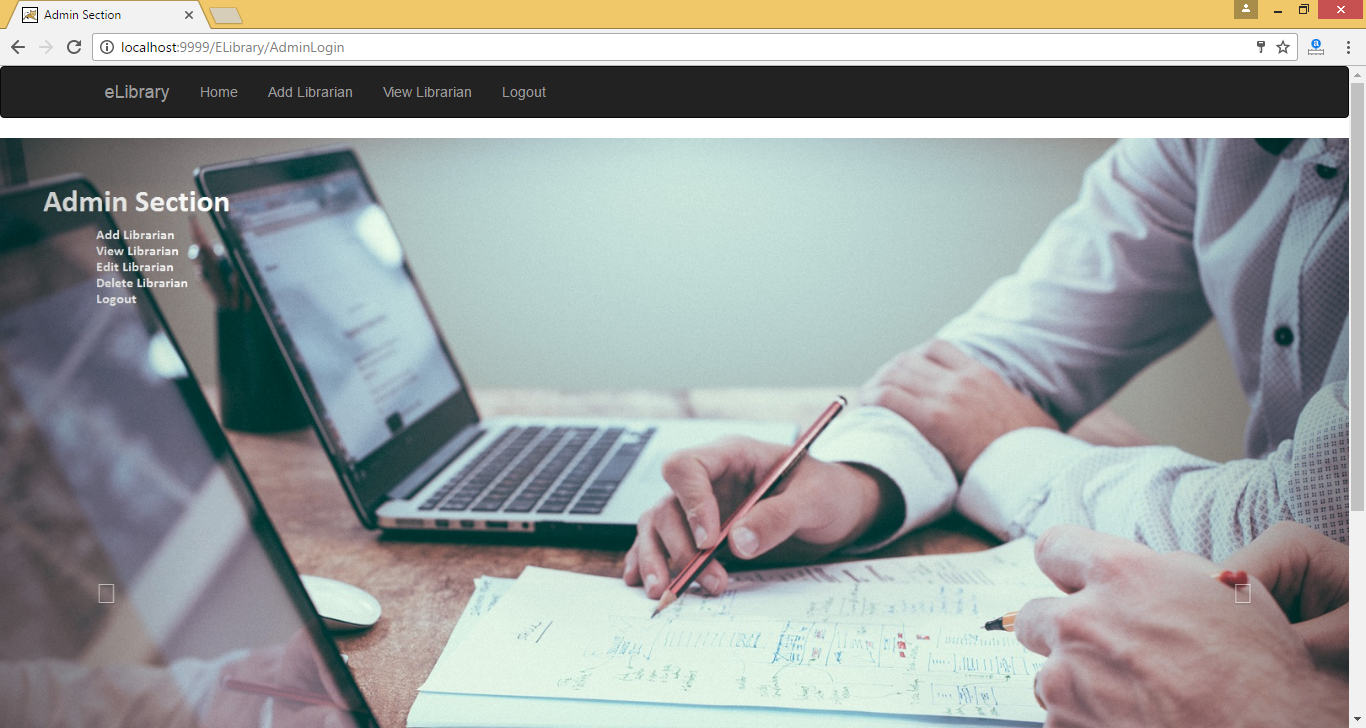
**E-Libraray ScreenShot :**

Enter email: [admin@xyz.com](mailto:admin@xyz.com)

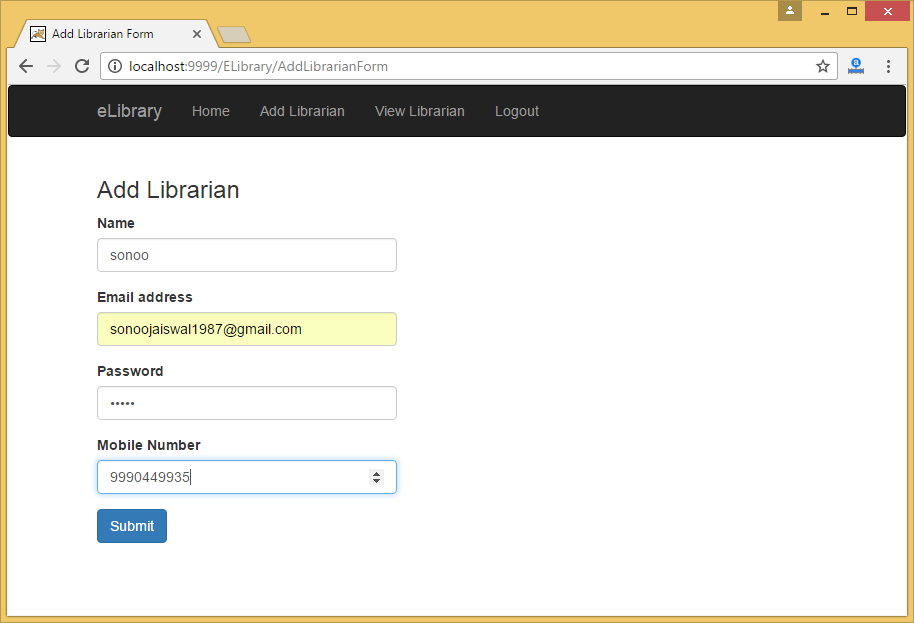
Enter password: admin123

Now login as admin.

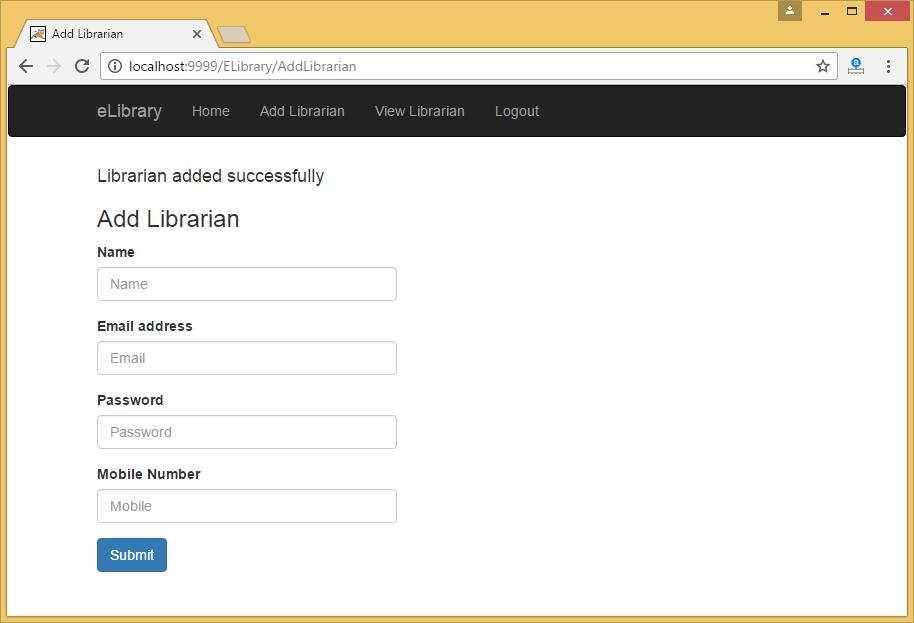




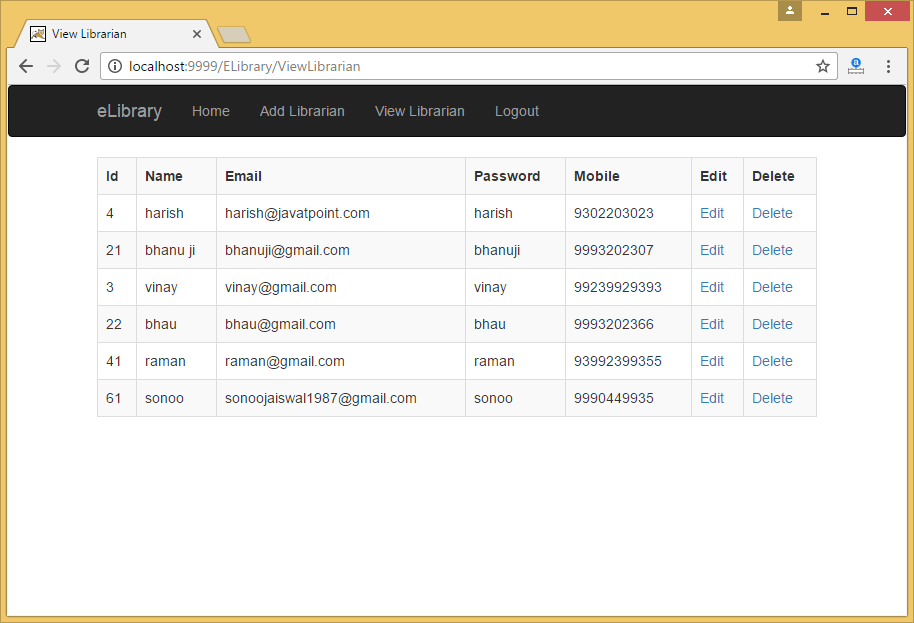
Click on Add Librarian:



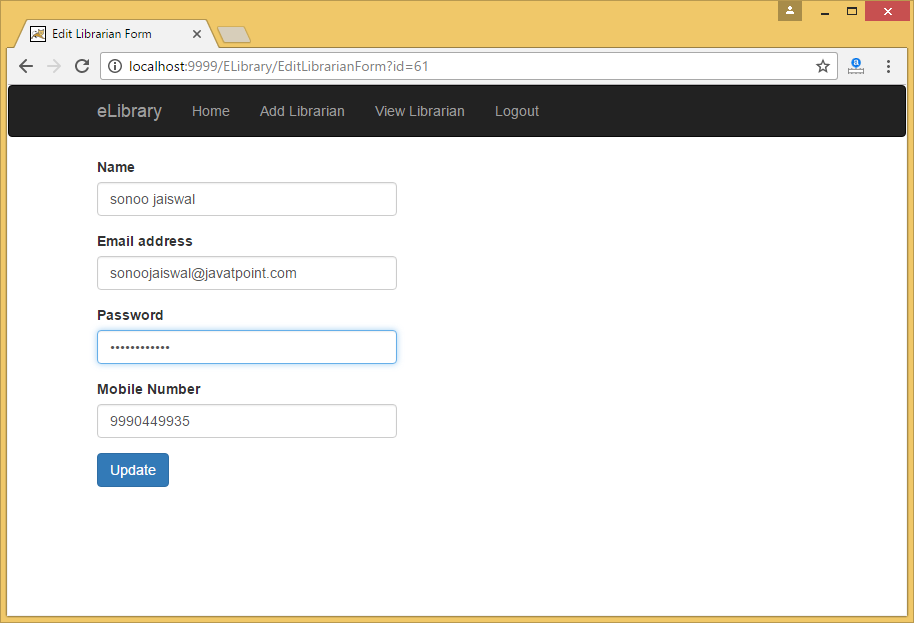
After filling form click on submit button.



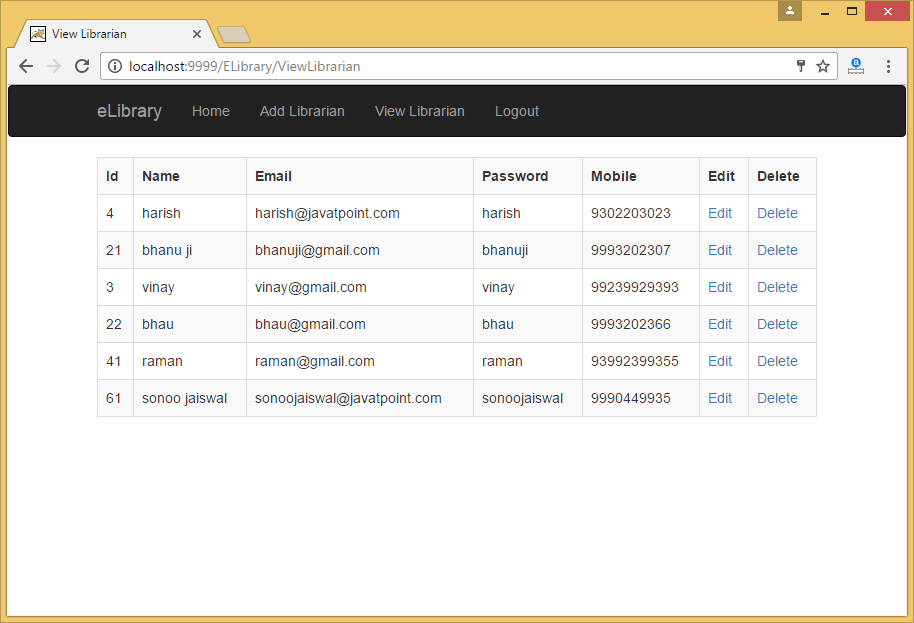
Now click on View Librarian:



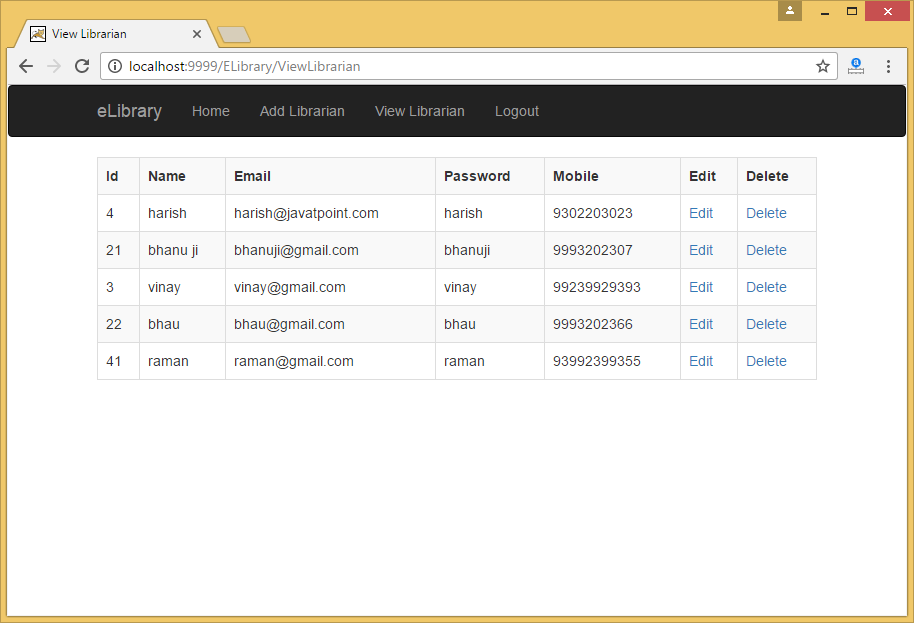
Now click on edit link:



After changing the information, click on update button.

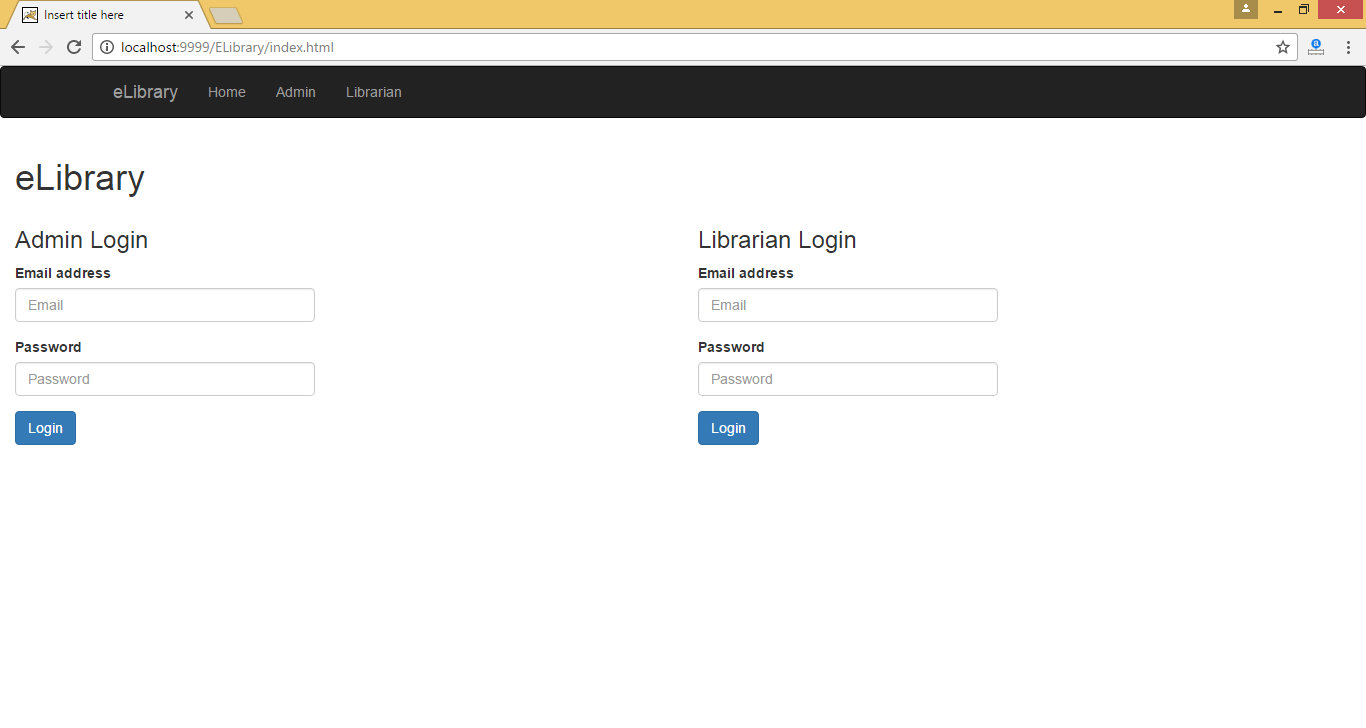


Now click on delete link:

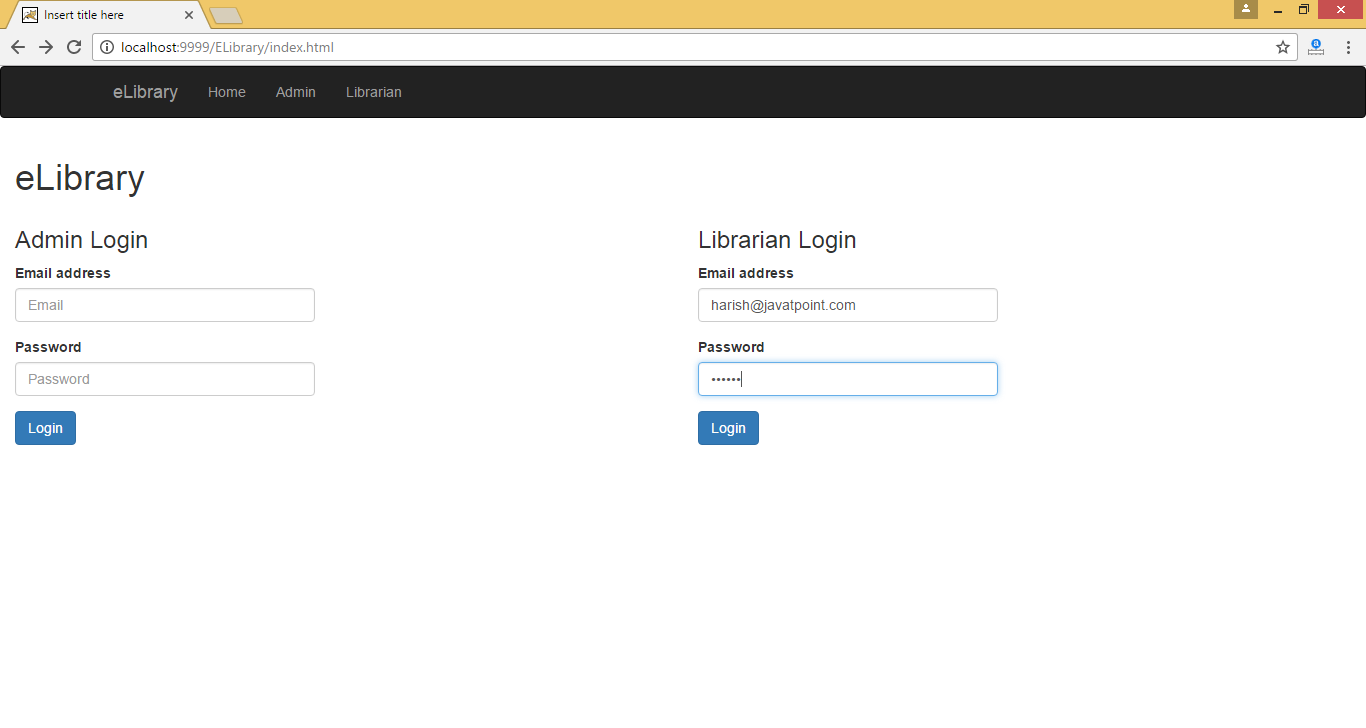


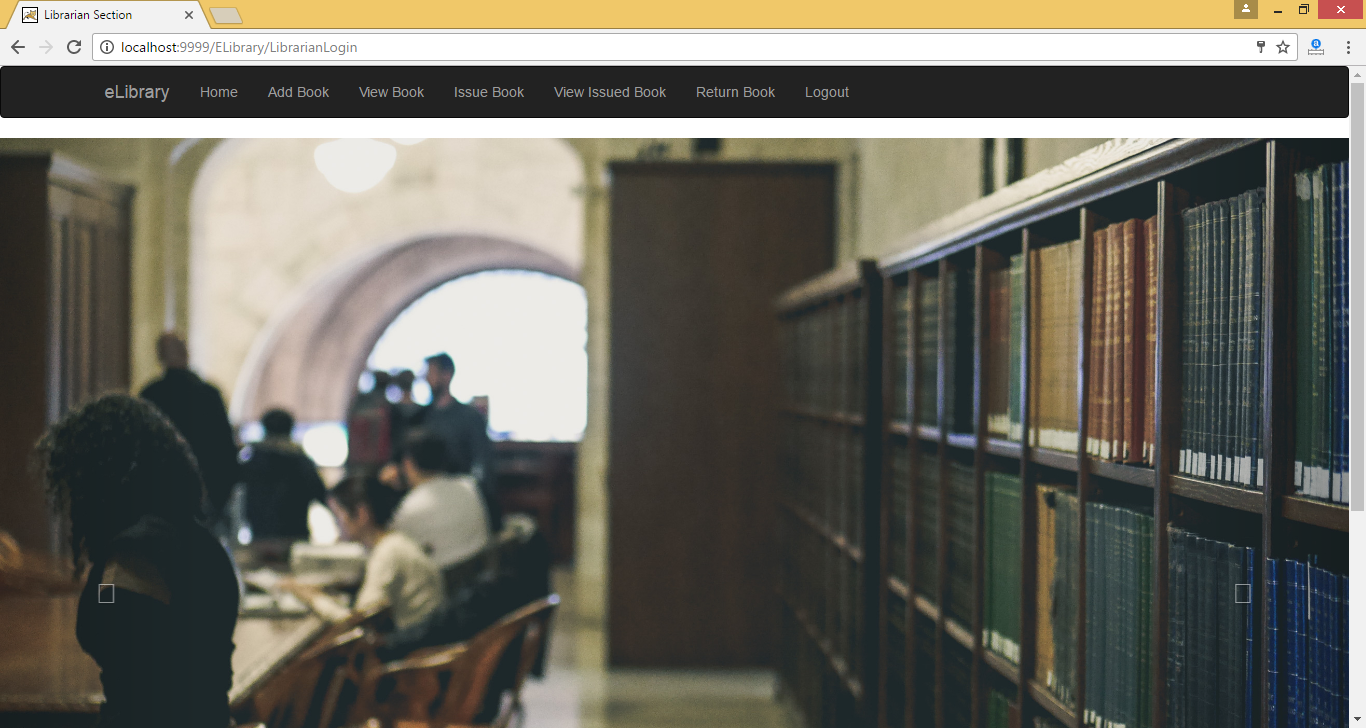
Now you can see that record is deleted.

Now click on logout button.

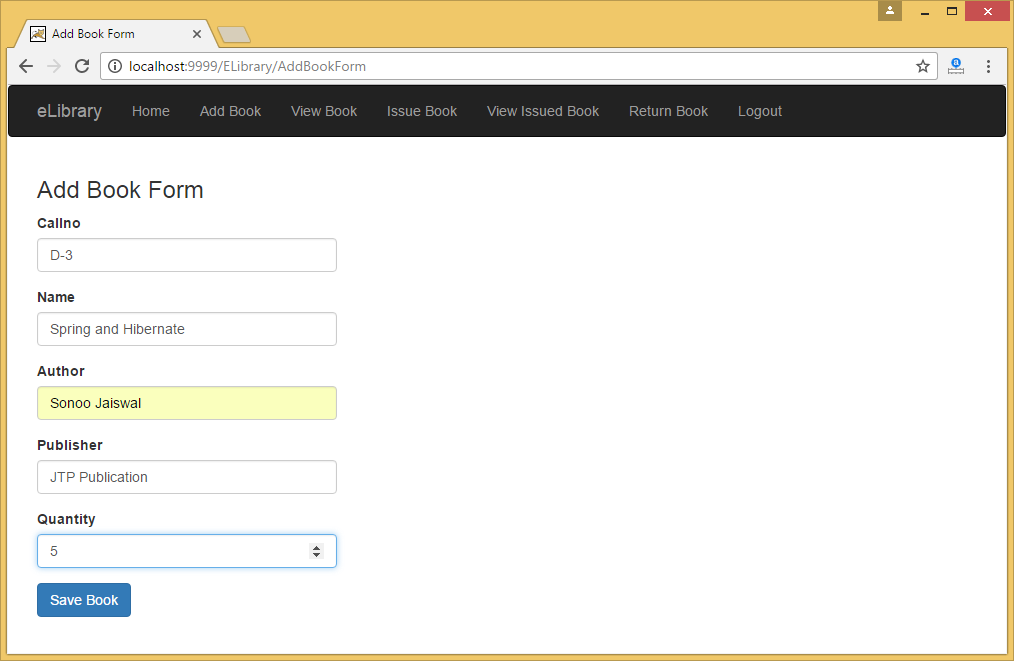


Now login through librarian section.

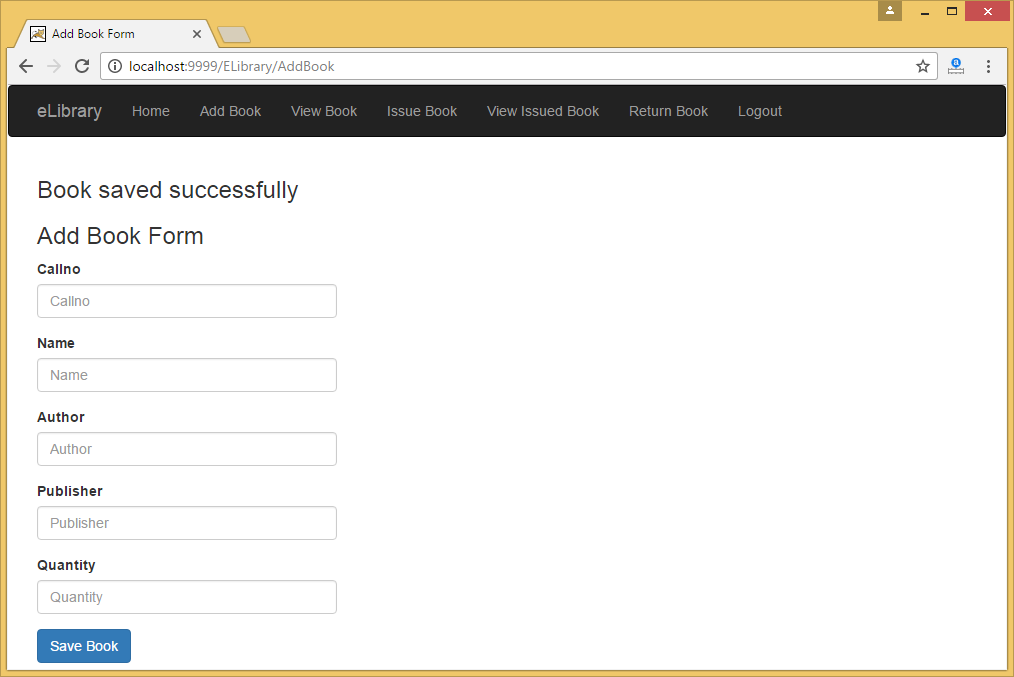




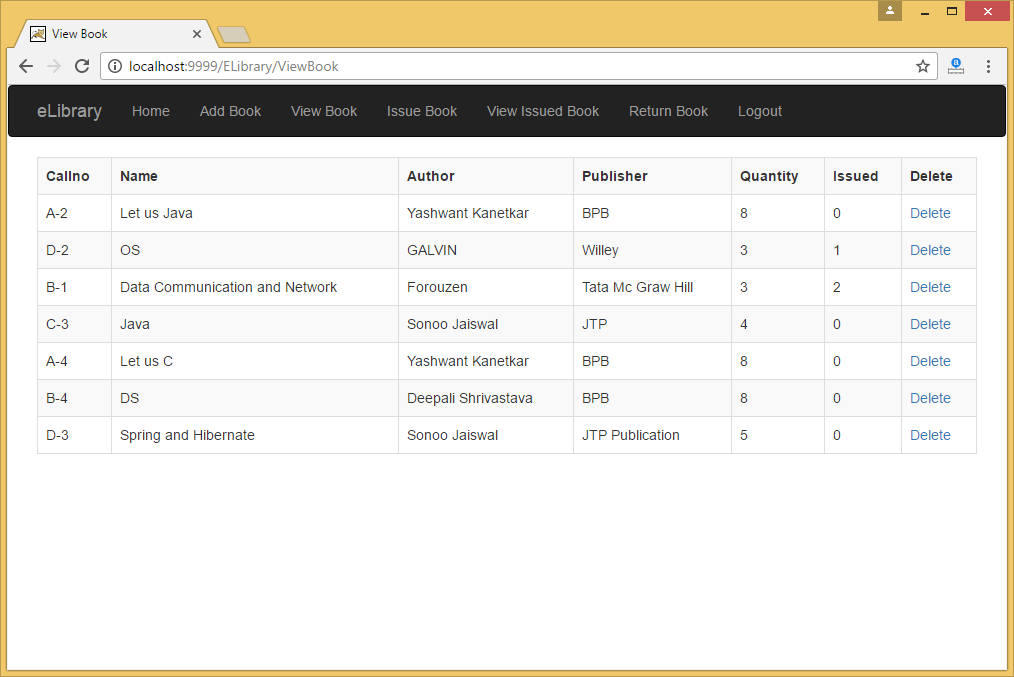
Now click on add book:



After filling the form, click on Save Book button.

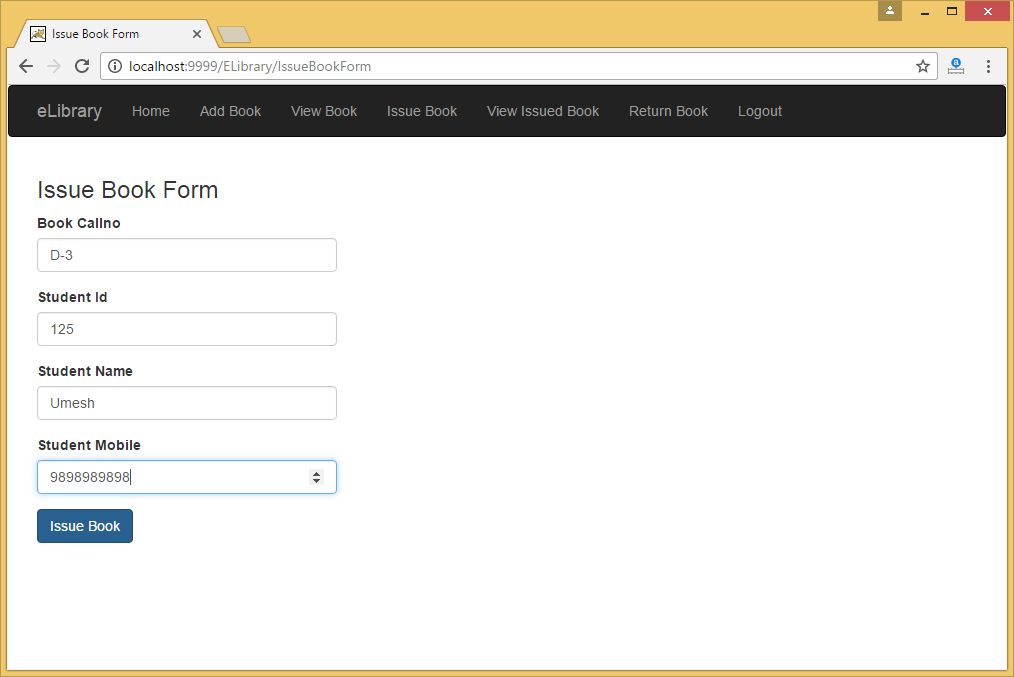


Now click on View Book link:



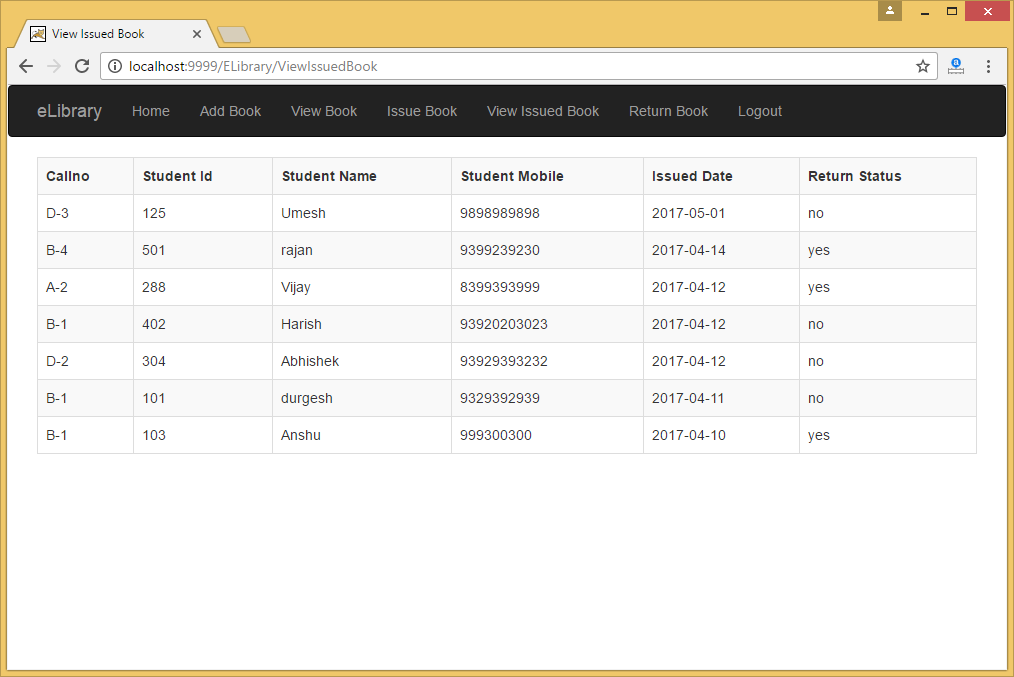
Now you can see that book is not issued to anyone, so it is set to 0.

Now click on Issue Book:



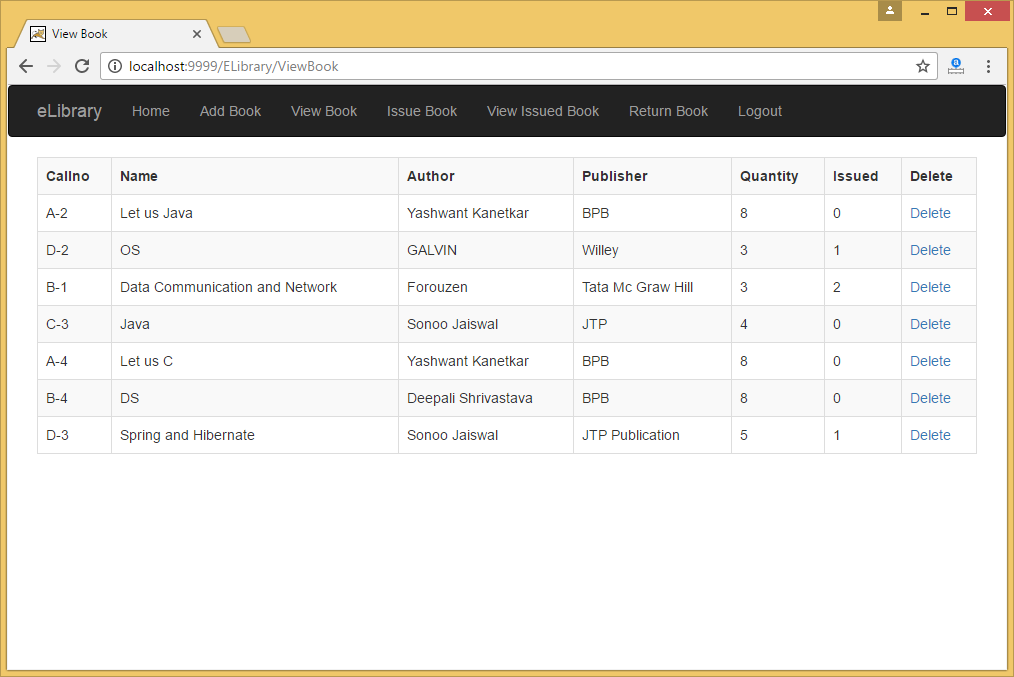
After filling the form, click on Issue Book button.

Now click on the View Issued Book link:



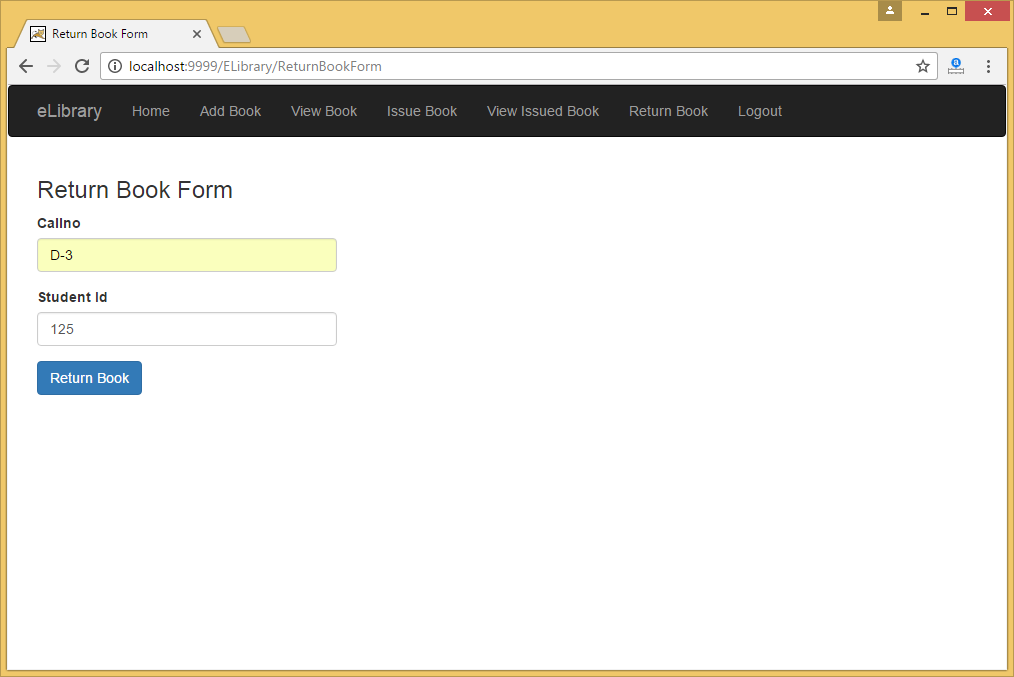
Now you can see that D-3 callno is listed in the table and its return status is no.

Again click on the View Book link:

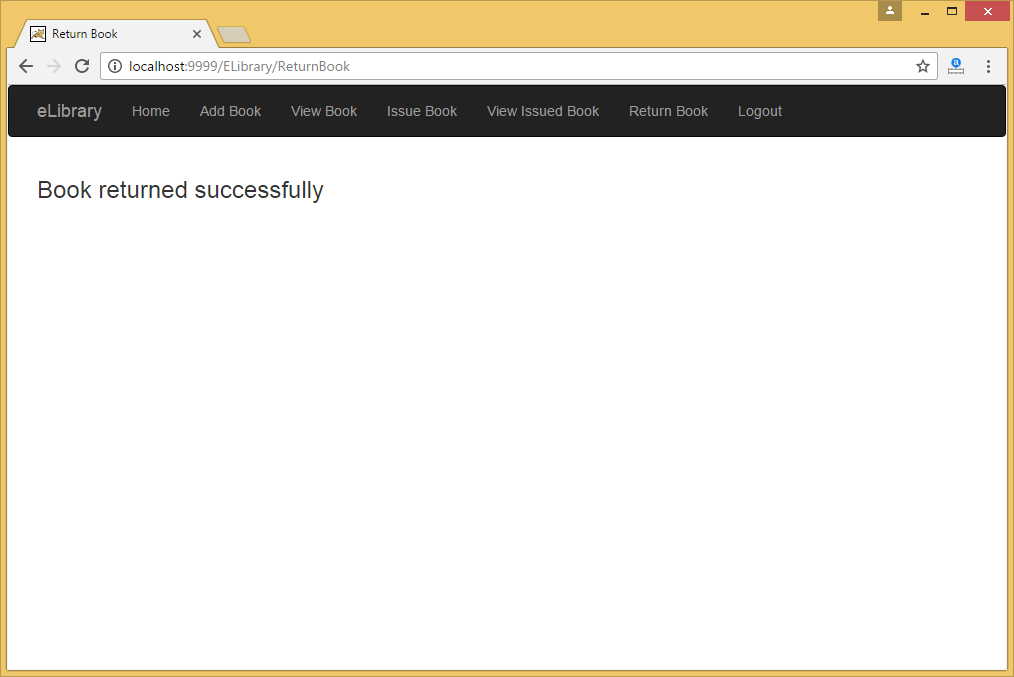


Here, you can see that issued column displays 1. If no. of issued books are equal to quantity, you will not be able to issue book.

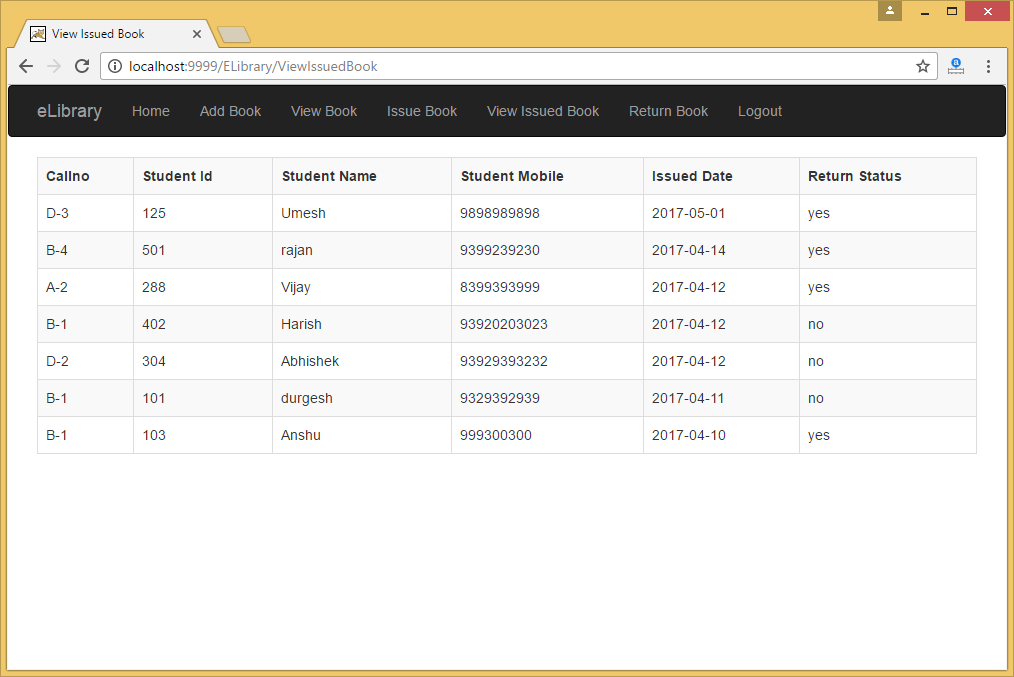
Now click on the Return Book link:



After filling correct callno and student id, click on Return Book button.

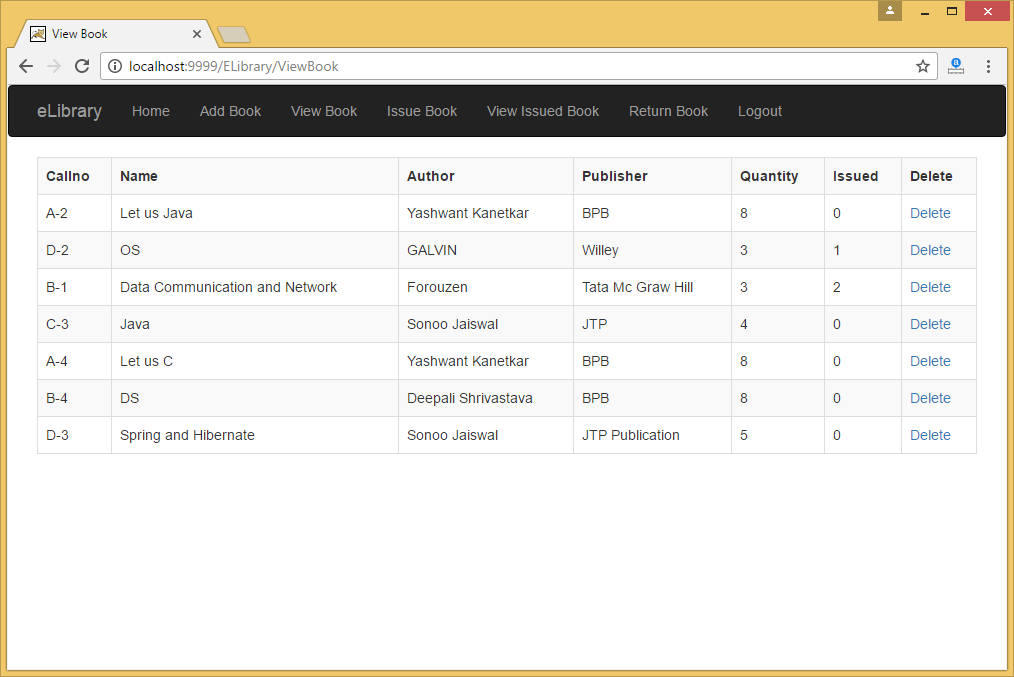


Now click on View Issued Book link again:



Here, you can see that return status is set to yes.

Now click on the View Book link again:



Here, you can see that issued no. is set to 0.

Now, click on logout.

**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. OurProject went through two levels of testing

1.Unit testing

2.integration testing

**UNIT TESTING**

Unit testing is undertaken when a module has been created and succesfully reviewed .In order to test a single module we need to provide a complete environment ie 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

**1. Test For the admin module**

**A. Testing admin login form**- This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password

**B Librarian account addition**- In this section the admin can verify librarian details from librarian info and then only add librarian details to main library database it contains add and delete buttons if user click add button data will be added to librarian database and if he clicks delete button the librarian data will be deleted

**C Book Addition**- Librarian can enter details of book and can add the details to the main book table also he can view the books requests .

**2. Test for Librarian login module**

**A Test for Librarian login Form**-This form is used for log in of Librarian .In this we enter the username and password if all these are correct Librarian login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.

**B Test for account creation**- This form is used for new account creation when Librarian does not fill the form completely it asks again to fill the whole form when he fill the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.

**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.

**CONCLUSION & FUTURE SCOPE**

This website provides a computerized version of library management system which will benefit the admin as well as the staff of the library.

It makes entire process online where librarian can search books, staff can generate reports and do book transactions. It also has a facility for librarian login where librarian can login and can see status of books issued as well request for book or give some suggestions. It has a facility of admin’s login where admin can add librarian details 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 librarian 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.Udemy.com/css/css\_background.asp

<http://www.w3schools.com/js/js_datatypes.asp>

https://www.javatpoint.com/