

Project Report: Library Management System

1. Introduction

1.1 Project Overview

The Library Management System is a web-based application developed using Java, JSP, Servlets, JDBC, and Oracle SQL. It provides efficient management of library operations, including book management, user authentication, and transaction handling. The system offers an admin panel for managing books, viewing records, and logging out securely, as well as a user module for registration, login, and book requests. Passwords are securely hashed using BCrypt for enhanced security.

1.2 Objectives

- Efficient Book Management: Add, delete, update, and view books.
- User Management: Register, login, request books, and view requests.
- Secure Authentication: Use BCrypt hashing for password security.
- Seamless Navigation: Intuitive and user-friendly interface.
- Database Management: Efficient data storage and retrieval with Oracle SQL.

1.3 Scope

- For Librarians/Admins:
 - Manage book records.
 - Approve/reject book requests.
 - Perform CRUD (Create, Read, Update, Delete) operations.
 - Secure authentication with session management.
- For Users:
 - Register and login.
 - Browse and search for books.
 - Request books.
 - View their book request history.

2. System Requirements

2.1 Functional Requirements

- User Authentication: Admins and users log in with valid credentials (passwords hashed using BCrypt).
- Book Management:
 - Add, delete, update, and view books.
 - Display book details in a structured format.
- Book Requests:
 - Users can request books.
 - Admins can approve or reject requests.
- Logout Functionality: Securely end the session.
- Navigation: User-friendly sidebar navigation.

2.2 Non-Functional Requirements

- Security: Prevent back navigation after logout.
- Performance: Fast data retrieval and rendering.
- Scalability: Future-ready architecture for additional functionalities.
- Responsiveness: Optimized UI for different screen sizes.

2.3 Operating Environment

- Server-Side: Apache Tomcat (Servlet Container).
- Database: Oracle SQL.
- Development Tools: Eclipse IDE.
- Browser Compatibility: HTML5, CSS3, JavaScript support.

3. System Architecture

3.1 Overview

The system follows the Model-View-Controller (MVC) architecture:

- Model: Manages data interactions with Oracle SQL using JDBC.
- View: JSP files render the interface.

- Controller: Java Servlets handle requests and responses.

3.2 MVC Flow

1. User Interaction: Admins and users log in via JSP form.
 2. Servlet Handling: Form data is sent to appropriate servlets.
 3. Password Hashing: BCrypt verifies the password against the hashed value.
 4. Database Interaction: JDBC validates credentials and processes book and request operations.
 5. Session Management: On success, the admin/user panel is displayed; otherwise, an error message is shown.
 6. Book and Request Management: CRUD operations and book requests are handled by dedicated servlets.
-

4. Database Design

4.1 Database Tables

Books Table

```
CREATE TABLE book (
    bookid VARCHAR2(6) PRIMARY KEY,
    bookname VARCHAR2(30) NOT NULL,
    author VARCHAR2(45) NOT NULL,
    price NUMBER NOT NULL,
    quantity NUMBER NOT NULL
);
```

desc Book;

Name	Null?	Type
------	-------	------

BOOKID	NOT NULL	VARCHAR2(6)
--------	----------	-------------

BOOKNAME	VARCHAR2(30)
----------	--------------

AUTHOR	VARCHAR2(45)
PRICE	NUMBER
QUANTITY	NUMBER

Admin Table

```
CREATE TABLE admin (
    email VARCHAR2(100) PRIMARY KEY,
    name VARCHAR2(50) NOT NULL,
    phone varchar2(10) NOT NULL
    DOB date,
    password VARCHAR2(255) NOT NULL -- BCrypt hashed password
);
```

```
desc Admin;
```

Name	Null?	Type
NAME		VARCHAR2(50)
EMAIL		VARCHAR2(50)
PHONE		VARCHAR2(10)
DOB		DATE
PASSWORD		VARCHAR2(60)

User Table

```
CREATE TABLE library_user (
    user_id VARCHAR2(15) PRIMARY KEY,
    name VARCHAR2(50) NOT NULL,
    email VARCHAR2(100) UNIQUE NOT NULL,
    password VARCHAR2(255) NOT NULL, -- Store hashed password
    role VARCHAR2(10) NOT NULL CHECK (role IN ('USER', 'ADMIN')) -- Role-based access
);
```

```
desc library_user;
```

Name	Null?	Type
USER_ID	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	VARCHAR2(50)
EMAIL	NOT NULL	VARCHAR2(100)
PASSWORD	NOT NULL	VARCHAR2(255)
ROLE	NOT NULL	VARCHAR2(10)

Book Request Table

```
CREATE TABLE book_request (
    request_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    bookname VARCHAR2(50) NOT NULL,
    author VARCHAR2(50) NOT NULL,
    requested_by VARCHAR2(50) NOT NULL,
```

```

status VARCHAR2(20) DEFAULT 'PENDING' CHECK (status IN ('PENDING', 'APPROVED',
'REJECTED')),

request_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);

desc book_request;
-----
```

Name	Null?	Type
REQUEST_ID	NOT NULL	NUMBER
BOOKNAME	NOT NULL	VARCHAR2(50)
AUTHOR	NOT NULL	VARCHAR2(50)
REQUESTED_BY	NOT NULL	VARCHAR2(50)
STATUS		VARCHAR2(20)
REQUEST_DATE		TIMESTAMP(6)

5. Implementation Details

5.1 Project Structure

📁 Java Resources:

src/main/java

 └ 📁 bean_pojo → Contains JavaBeans for Admin, User, Book, and Book Requests.

- | └ AdminBean.java
- | └ BookBean.java
- | └ UserBean.java
- | └ RequestBookBean.java

|-  controller → Handles HTTP requests and responses.

- | |- AddBook.java
- | |- AdminLogin.java
- | |- AdminRegister.java
- | |- DeleteBook.java
- | |- Logout.java
- | |- UpdateBook.java
- | |- ViewBook.java
- | |- UserRegister.java
- | |- UserLogin.java
- | |- UserLogout.java
- | |- RequestBook.java
- | |- UserBookRequests.java
- | |- ViewUserRequests.java

|-  dao → Handles DB operations using JDBC.

- | |- AddBookDAO.java
- | |- AdminLoginDAO.java
- | |- AdminRegisterDAO.java
- | |- DBconnection.java
- | |- DBinfo.java
- | |- DeleteBookDAO.java
- | |- UpdateBookDAO.java
- | |- ViewBookDAO.java
- | |- UserDAO.java
- | |- UserBookDAO.java
- | |- RequestBookDAO.java

|-  service → Password hashing service.

```
| └ PasswordHashing.java  
└ Web Content:  
  src/main/webapp  
    ├ META-INF  
    ├ WEB-INF  
    | └ lib  
    |   ├ bcrypt-0.4.jar      → Password hashing library  
    |   └ ojdbc11.jar        → Oracle JDBC driver  
    | └ web.xml              → Servlet configuration  
    ├ JSP Files             → Pages for admin and user functionalities  
    └ CSS Files             → Styling the frontend
```

6. User Roles and Access Levels

6.1 Admin

- Privileges:
 - Add, delete, update, and view books.
 - Approve or reject book requests.
 - Access the admin panel.
- Limitations:
 - No access to user functionalities.

6.2 User

- Privileges:
 - Register and log in.
 - Request books.
 - View request history.
- Limitations:

- No access to admin functionalities.
-

7. Testing and Validation

7.1 Testing Strategies

- **Unit Testing:**
 - Tested individual components such as servlets, DAO classes, and JavaBeans.
 - Verified password hashing and validation logic.
 - **Integration Testing:**
 - End-to-end testing of the entire flow:
 - Registration, Login, Book Management, and Logout.
 - Verified the correct flow of data between the **JSP pages, servlets**, and the **Oracle database**.
 - **User Acceptance Testing:**
 - Ensured that the UI behaves as expected with **SweetAlert** notifications.
 - Validated error handling, such as duplicate email registration and invalid credentials.
 - Ensured no **back navigation** after logout.
 - **Performance Testing:**
 - Evaluated the system's responsiveness under different loads.
 - Verified SQL query execution time for large datasets.
-

7.2 Test Cases

📌 User Registration

Test Case	Input	Expected Output	Actual Output	Status
Register new user	Valid email, password	Registration successful	Registration success	<input checked="" type="checkbox"/> Passed

Test Case	Input	Expected Output	Actual Output	Status
Register with duplicate	Existing email	SweetAlert: "User exists"	User exists alert	Passed
Invalid Email Format	invalid@	Validation error message	Validation message	Passed
Empty fields	Empty form submission	SweetAlert: "All fields required"	Validation message	Passed

The screenshot shows a web browser window titled "Admin Registration Form" with the URL "localhost:8585/LibraryManagement/Registration.jsp". The page displays a registration form with the following fields:

- Name:** A text input field with placeholder text "Enter your name".
- Email:** A text input field with placeholder text "Enter your email".
- Phone:** A text input field with placeholder text "Enter your phone number".
- DOB:** A date input field with placeholder text "dd-mm-yyyy".
- Password:** A text input field with placeholder text "Create a strong password".

A large red validation error message "Validation message" is displayed prominently at the top of the form area. At the bottom of the form, there is a purple "Register" button and a link "Already have an account? Login Here".

Registration

Name: Akash Kumar

Email: kumarakash09@gmail.com

Phone: 7778880001

DOB: 01-01-2001

Password: *****

Register

Already have an account? [Login Here](#)

Registration

Name: Akash Kumar

Email: kumarakash09@gmail.com

Please include an '@' in the email address. 'kumarakash09@gmail.com' is missing an '@'.

Phone: 7778880001

DOB: 01-01-2002

Password: ...

Register

Already have an account? [Login Here](#)

The screenshot shows a web browser window titled "Admin Registration Form" with the URL "localhost:8585/LibraryManagement/AdminRegistration". The main content is a registration form titled "Registration". The form fields are:

- Name:** Enter your name
- Email:** Enter your email
- Phone:** Enter your phone number
- DOB:** dd-mm-yyyy
- Password:** Create a strong password

A purple "Register" button is at the bottom. A green success message box in the top right corner says "Registration Successful!".

The screenshot shows a web browser window titled "Admin Registration Form" with the URL "localhost:8585/LibraryManagement/AdminRegistration". The main content is a registration form titled "Registration". The form fields are populated with sample data:

- Name:** Raj Rao
- Email:** kumarakash09@gmail.com
- Phone:** 7778880001
- DOB:** 02-02-2002
- Password:** ...

A purple "Register" button is at the bottom. A small link "Already have an account? Login Here" is at the bottom of the form.

Admin Registration Form

localhost:8585/LibraryManagement/AdminRegistration

Registration

Name:

Email:

Phone:

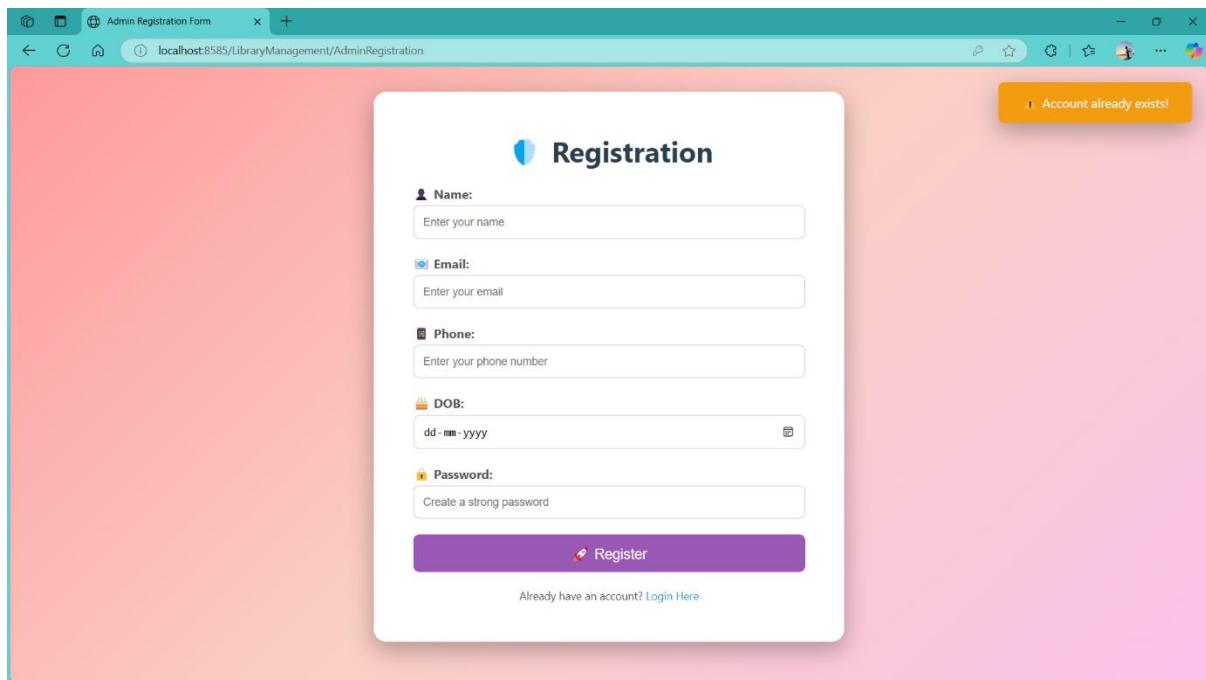
DOB:

Password:

Register

Already have an account? [Login Here](#)

Account already exists!



User Registration

localhost:9505/LibraryManagement/UserRegistration.jsp

User Registration

User ID (Max 15 chars):

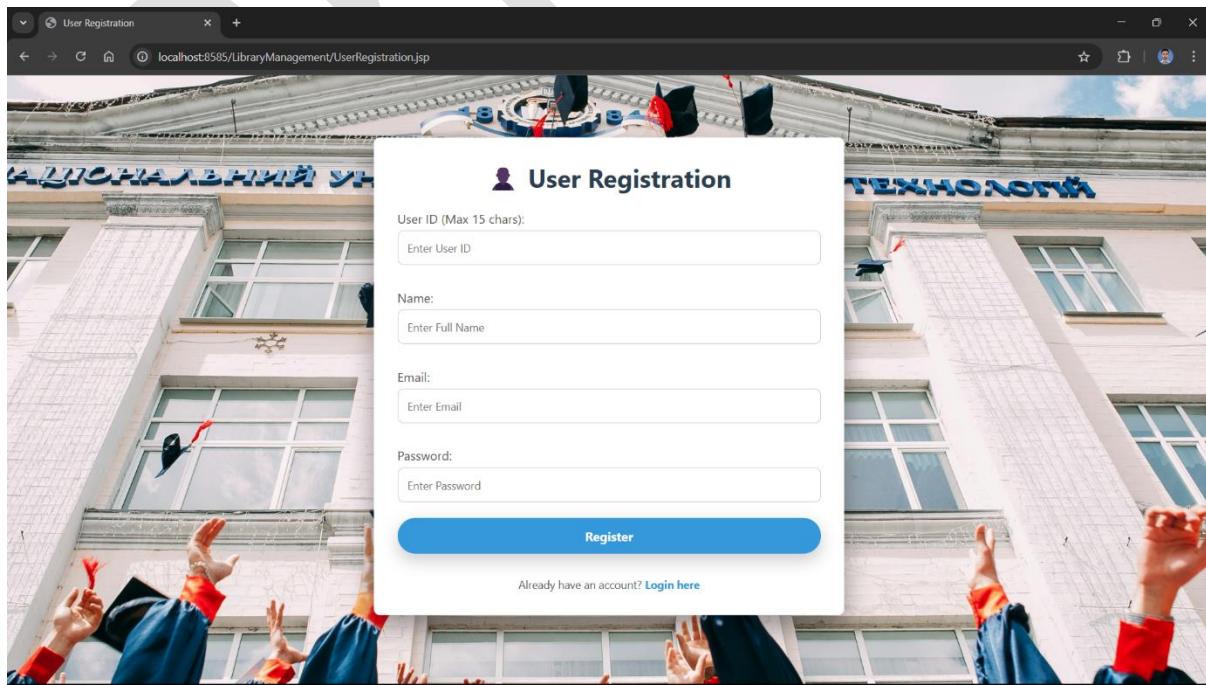
Name:

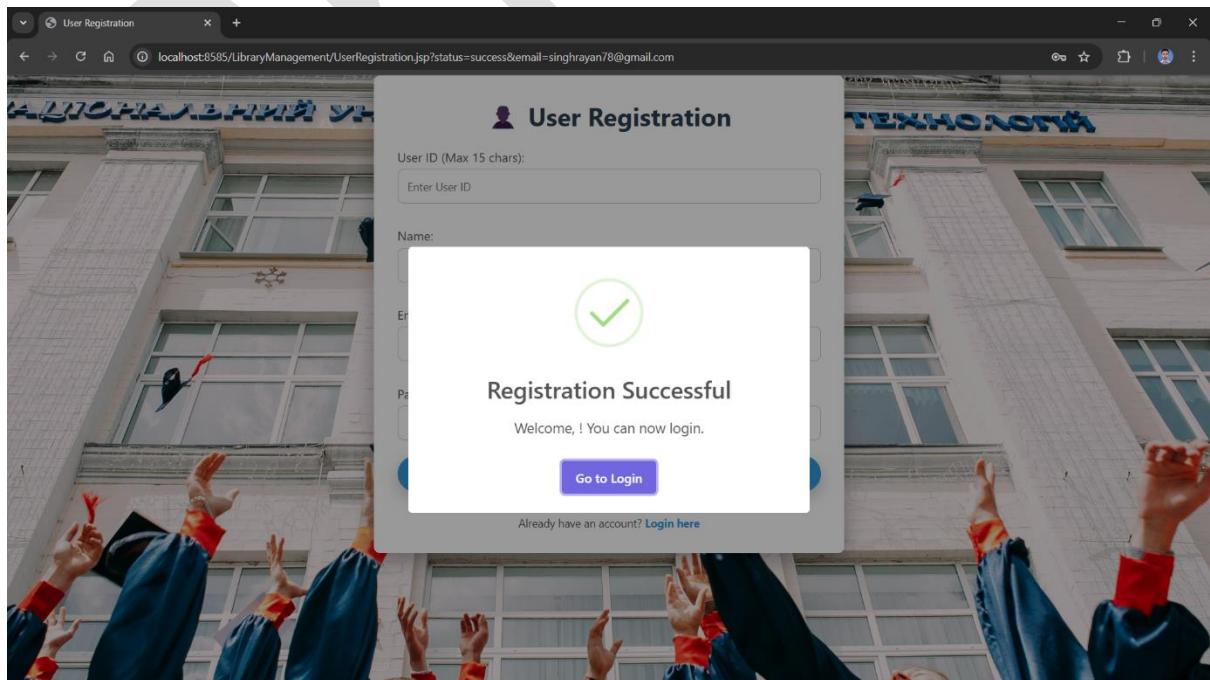
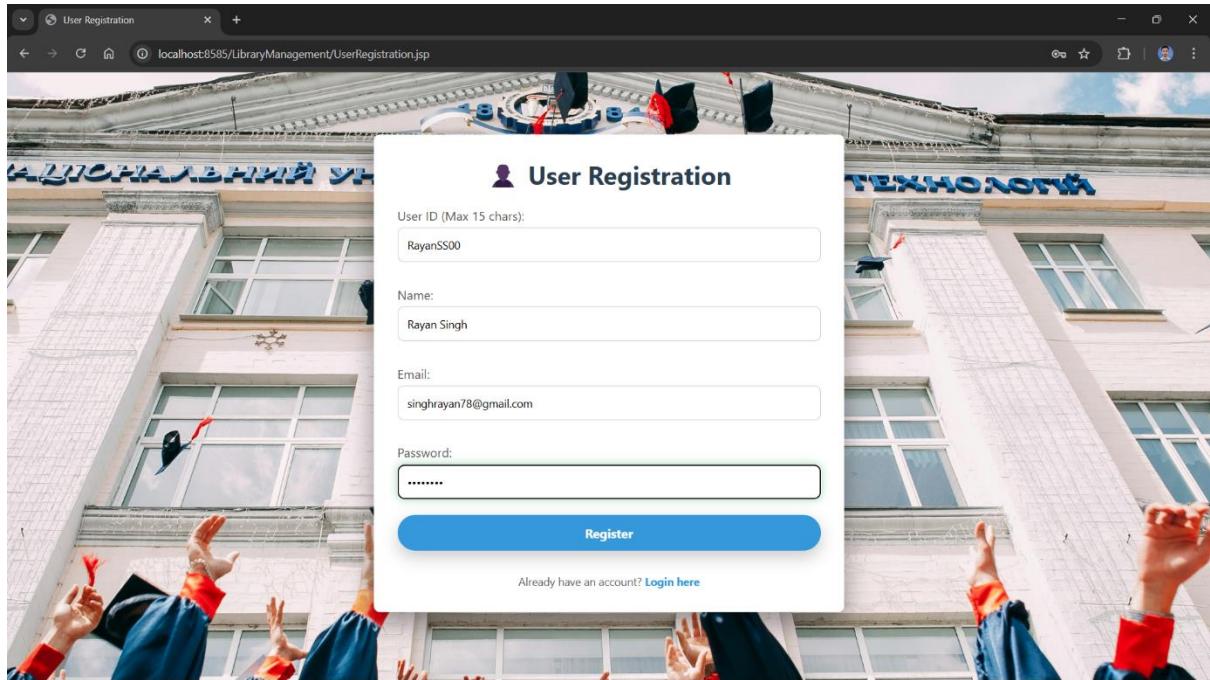
Email:

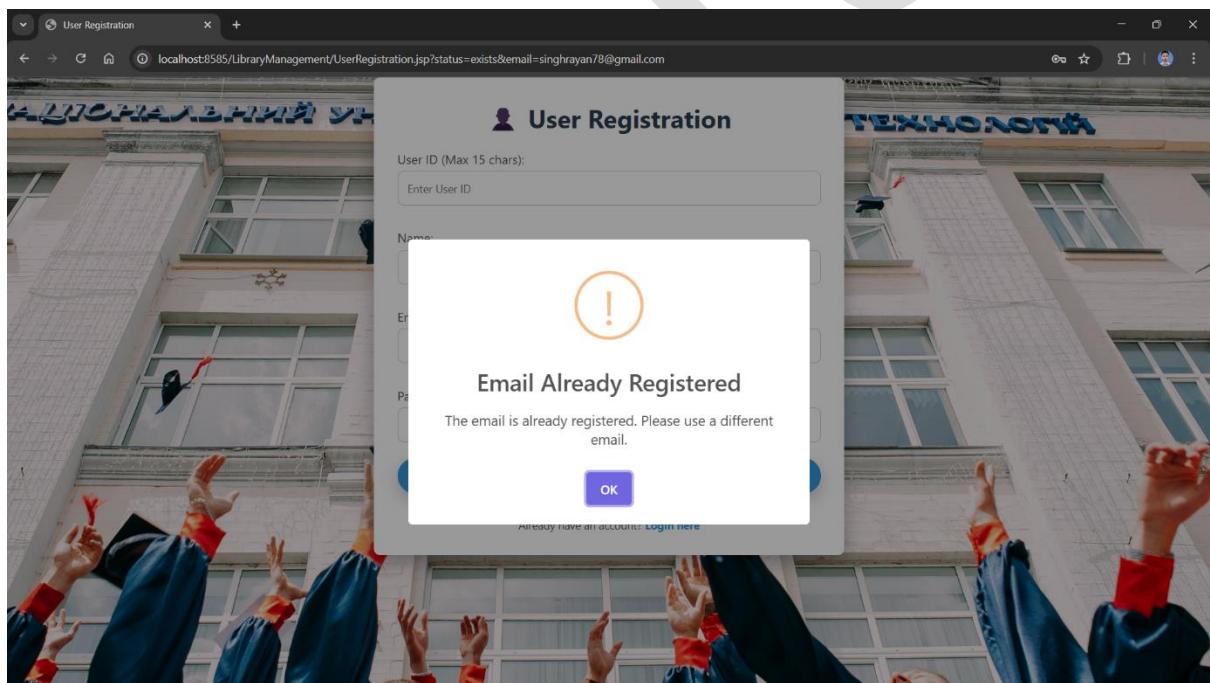
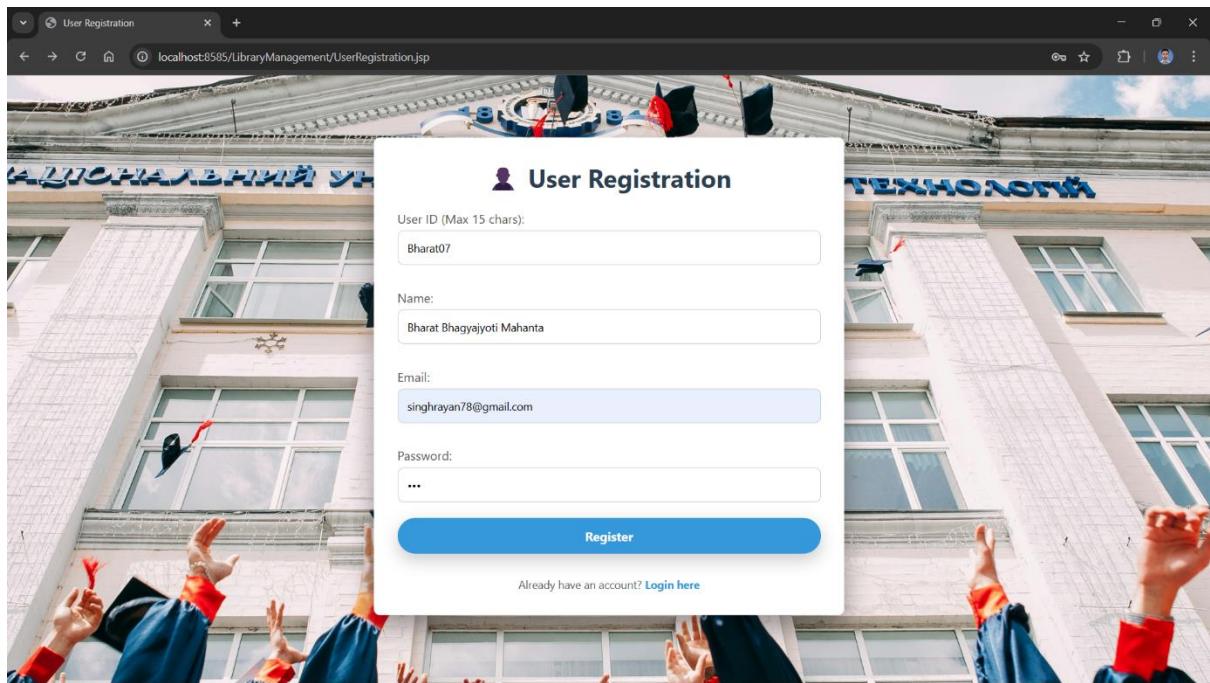
Password:

Register

Already have an account? [Login here](#)



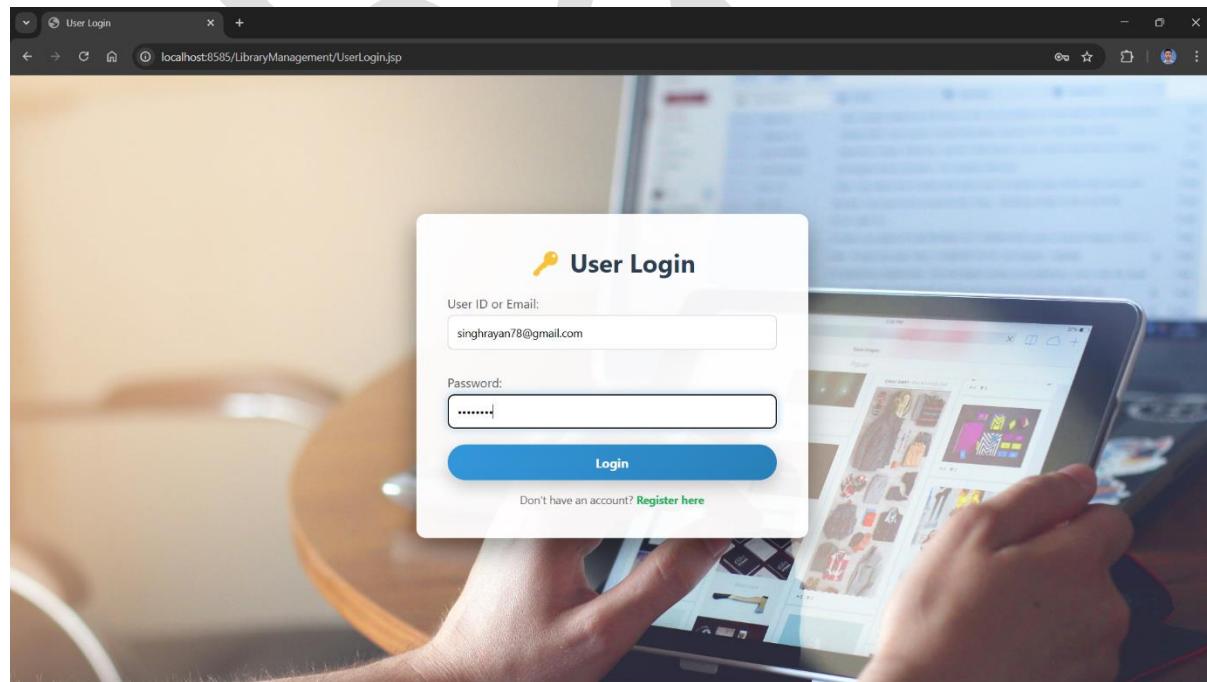


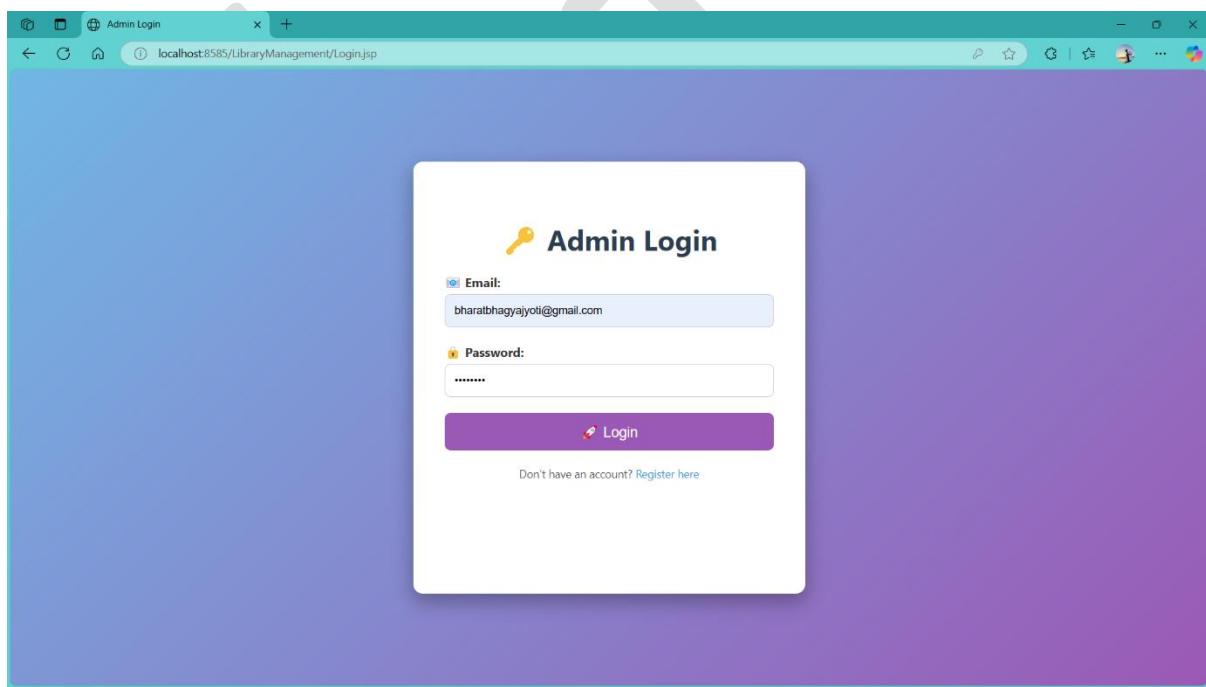
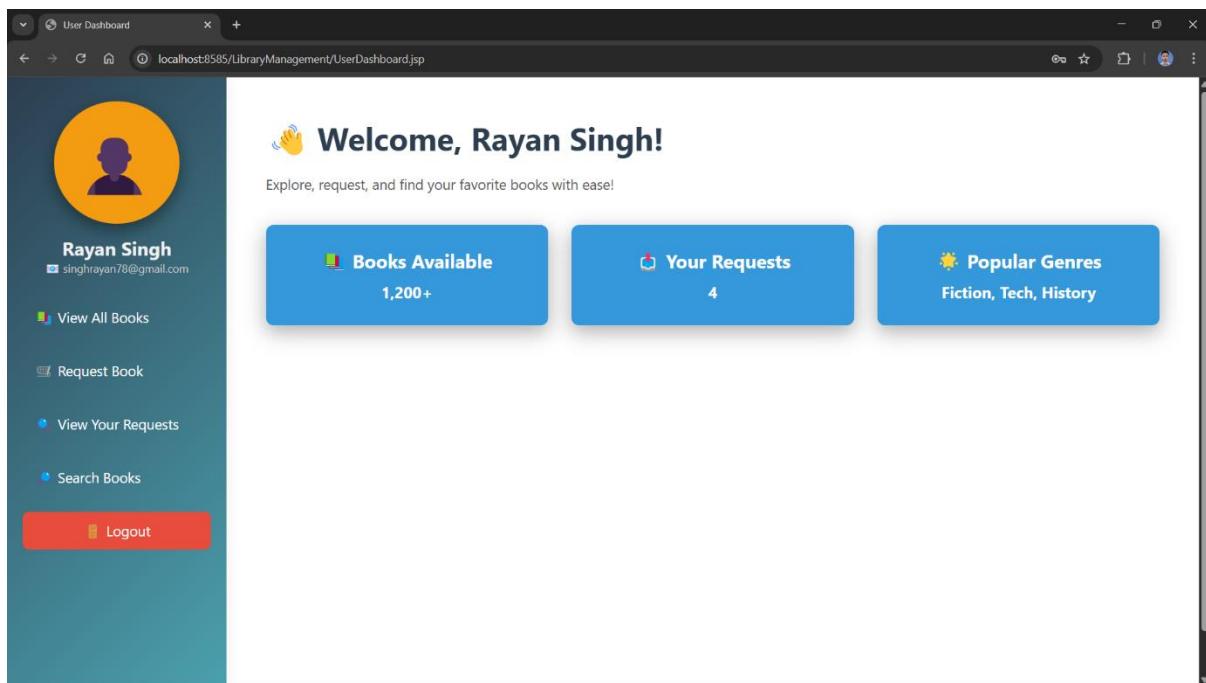


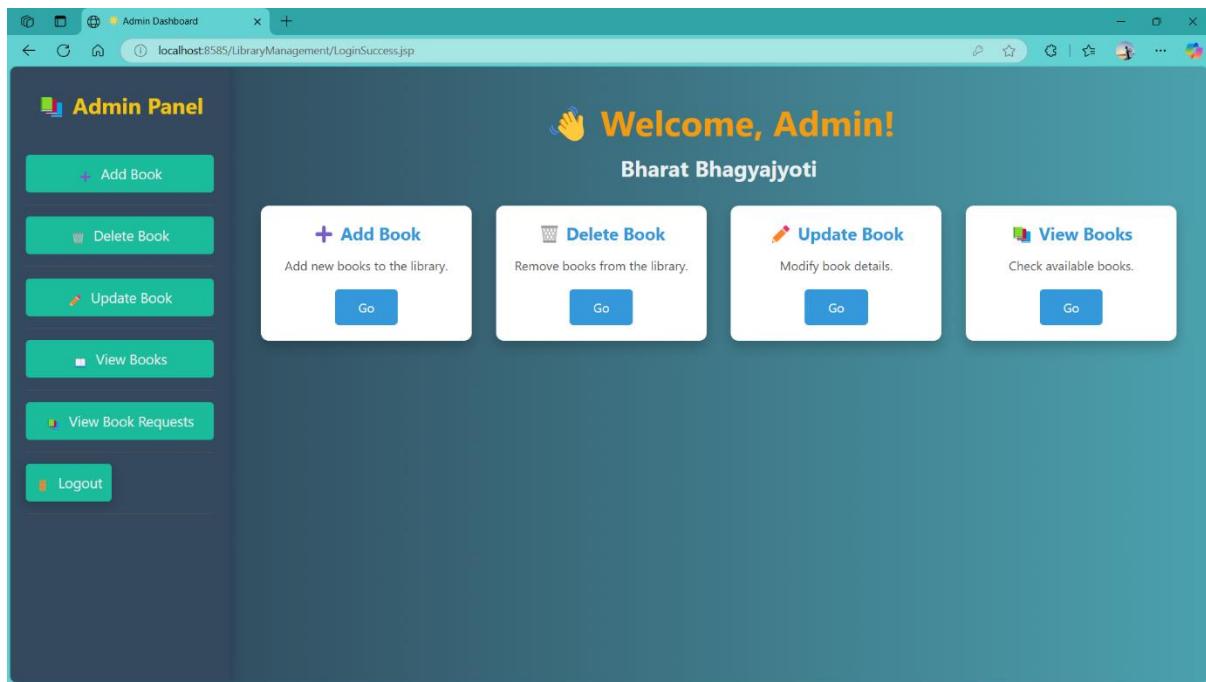


User Login

Test Case	Input	Expected Output	Actual Output	Status
Valid credentials	Correct email & password	Login successful, dashboard shows	Redirect to dashboard	✓ Passed
Invalid password	Wrong password	SweetAlert: "Invalid credentials"	Invalid credentials	✓ Passed
Non-existing email	Random email	SweetAlert: "User not found"	User not found alert	✓ Passed
Empty fields	Empty form submission	Validation message	Validation message	✓ Passed







📌 Book Management

Test Case	Action	Expected Output	Actual Output	Status
Add book	Enter book details	Book added successfully	Book added alert	<input checked="" type="checkbox"/> Passed
Update book	Modify existing book	Book updated successfully	Book updated alert	<input checked="" type="checkbox"/> Passed
Delete book	Select and delete book	Book deleted successfully	Book deleted alert	<input checked="" type="checkbox"/> Passed
Invalid book ID	Non-existent book ID	SweetAlert: "Book not found"	Book not found alert	<input checked="" type="checkbox"/> Passed
Check Book requests	Accept the request and add the book otherwise reject	Requests page will be updated	Book Added / Not Added	<input checked="" type="checkbox"/> Passed

Add Book Form

localhost:8585/LibraryManagement/AddBook.jsp

Book Details Form

Book ID (Length : 6 chars):
SPJ09A

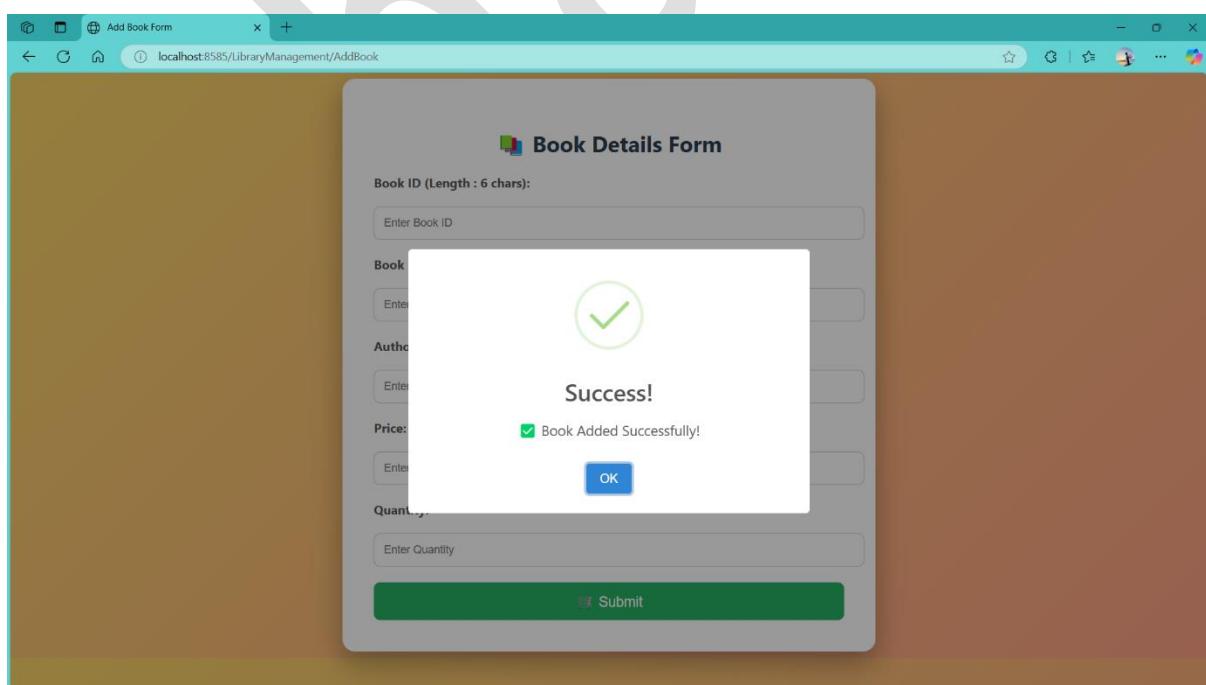
Book Name:
Spring - Java

Author:
Rod Johnson

Price:
769

Quantity:
10

Submit





A screenshot of a web browser window titled "Update Book". The URL in the address bar is "localhost:8585/LibraryManagement/UpdateBook.jsp". The main content is a form titled "Update Book Details" with the following fields:

- Book ID (Max Length : 6 chars): BBM999
- Book Name: You Will Be...
- Author: Bharat Bhagajyoti
- Price: Enter Price
- Quantity: Enter Quantity

At the bottom are two buttons: a green "Update" button with a checkmark icon and a red "Reset" button with a circular arrow icon. Below the buttons is a link "Back to Book List".



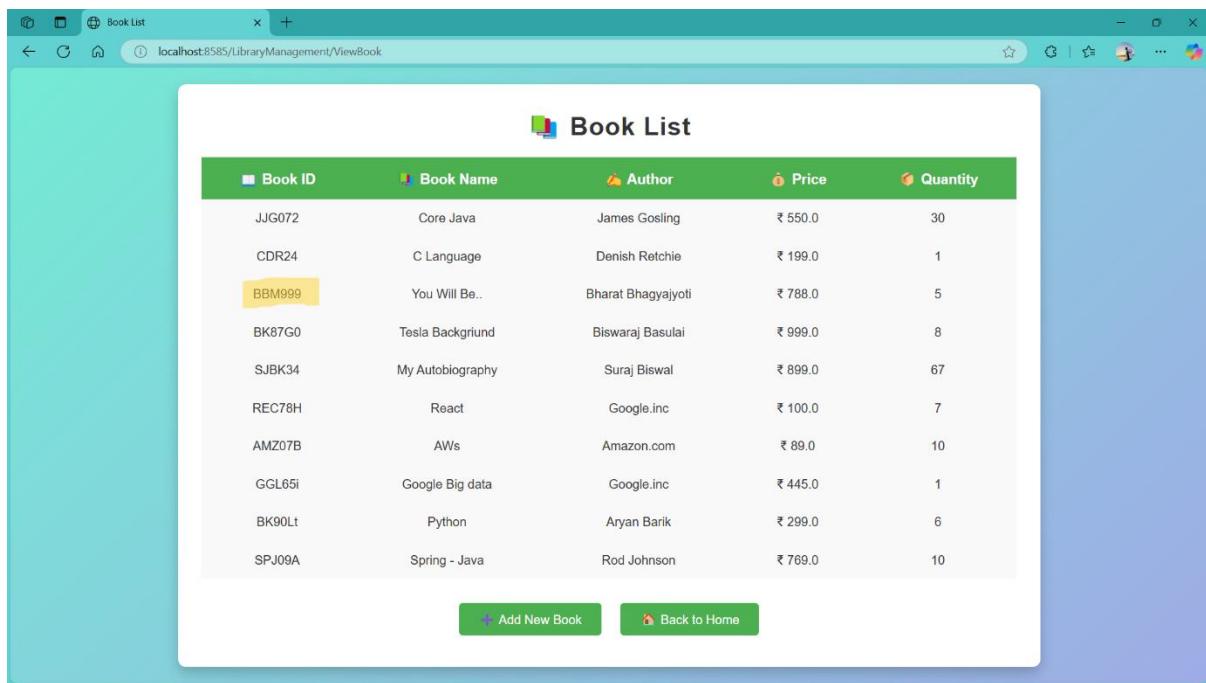
A screenshot of a web browser window titled "Update Book". The URL in the address bar is "localhost:8585/LibraryManagement/UpdateBook.jsp". The main content is a form titled "Update Book Details" with the following message at the top:

✓ Book Updated Successfully!

The rest of the form fields are identical to the one above:

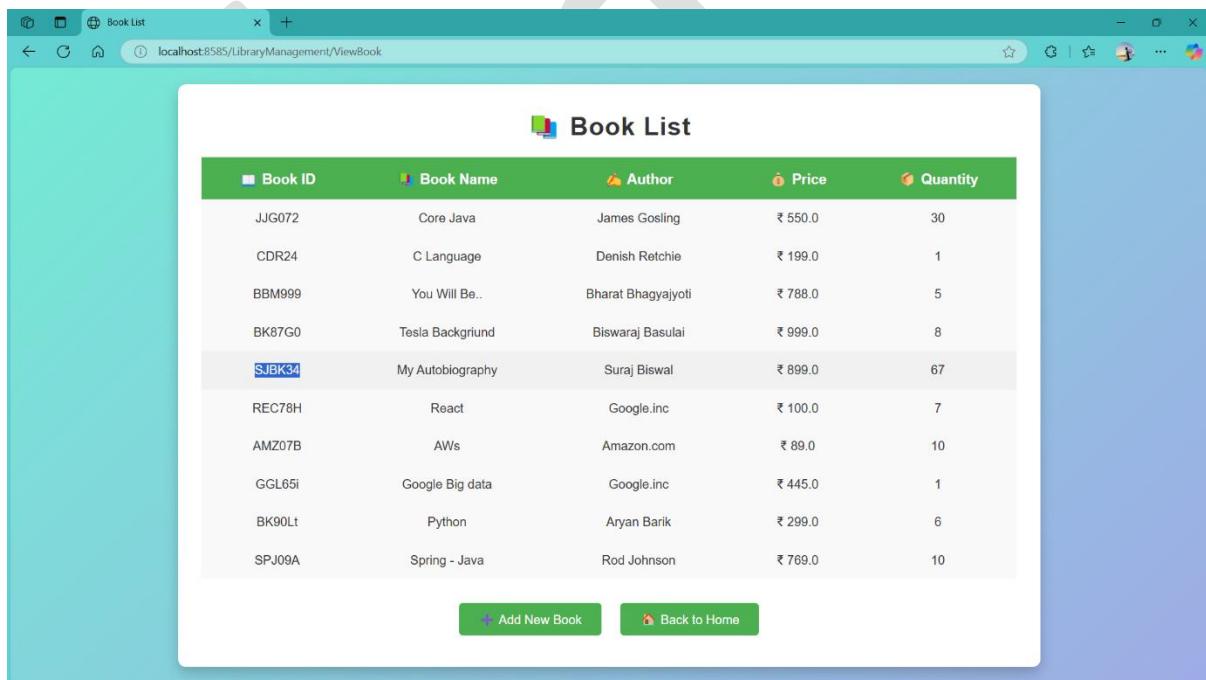
- Book ID (Max Length : 6 chars): Enter Book ID
- Book Name: Enter Book Name
- Author: Enter Author Name
- Price: Enter Price
- Quantity: Enter Quantity

At the bottom are two buttons: a green "Update" button with a checkmark icon and a red "Reset" button with a circular arrow icon. Below the buttons is a link "Back to Book List".



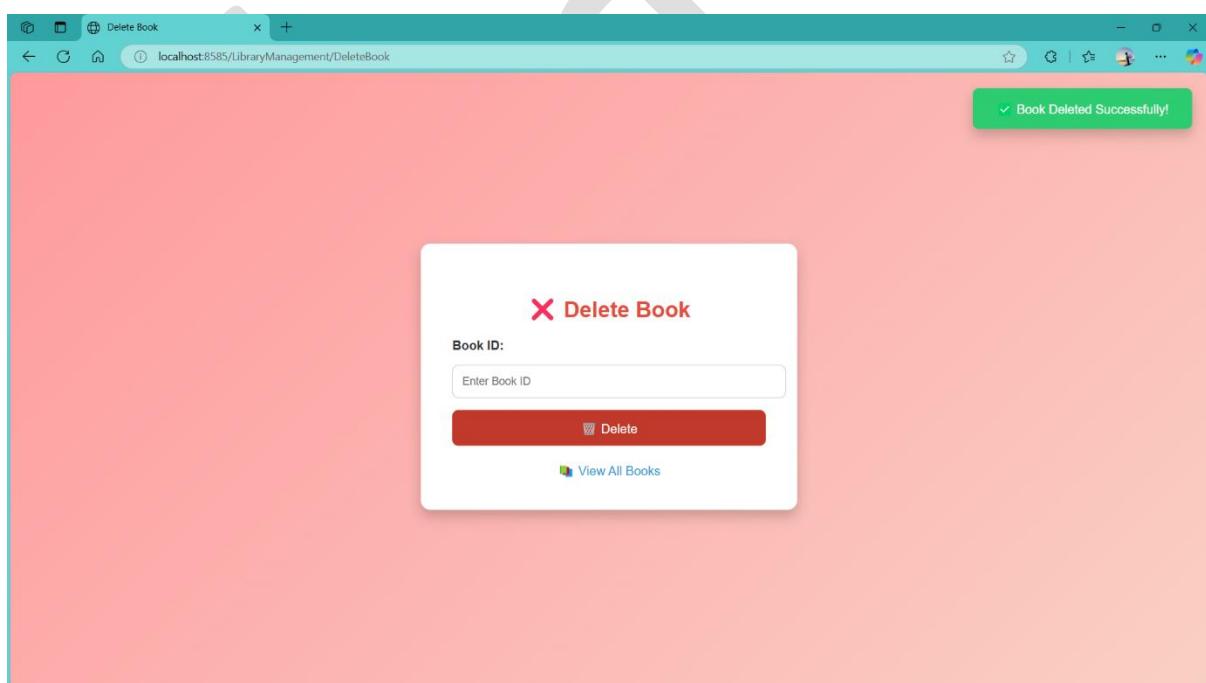
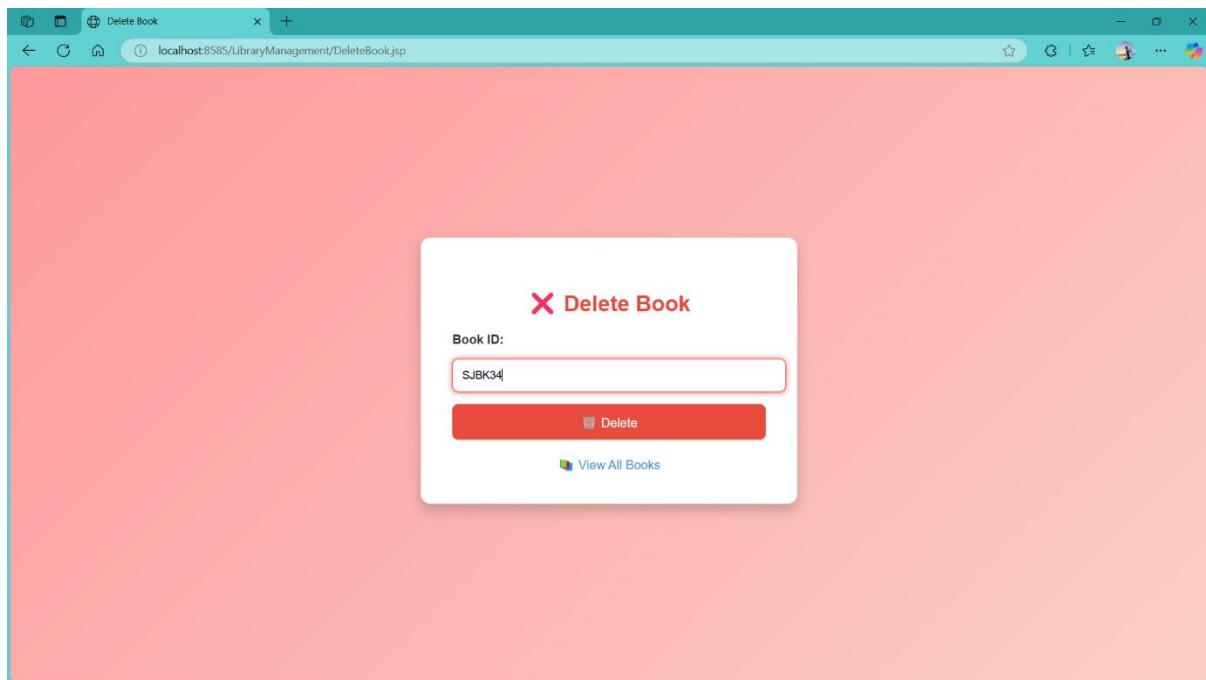
Book ID	Book Name	Author	Price	Quantity
JJJG072	Core Java	James Gosling	₹ 550.0	30
CDR24	C Language	Denish Retchie	₹ 199.0	1
BBM999	You Will Be..	Bharat Bhagyajyoti	₹ 788.0	5
BK87G0	Tesla Backgrund	Biswaraj Basulai	₹ 999.0	8
SJBK34	My Autobiography	Suraj Biswal	₹ 899.0	67
REC78H	React	Google.inc	₹ 100.0	7
AMZ07B	AWS	Amazon.com	₹ 89.0	10
GGL65i	Google Big data	Google.inc	₹ 445.0	1
BK90Lt	Python	Aryan Barik	₹ 299.0	6
SPJ09A	Spring - Java	Rod Johnson	₹ 769.0	10

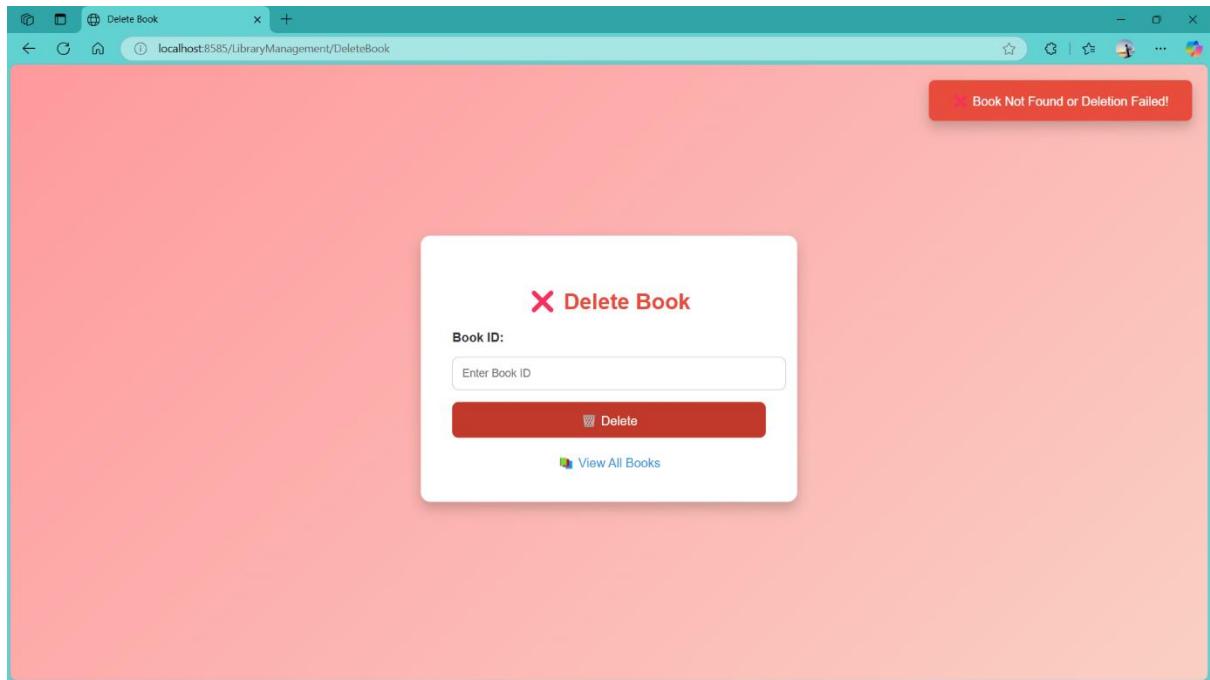
[Add New Book](#) [Back to Home](#)



Book ID	Book Name	Author	Price	Quantity
JJJG072	Core Java	James Gosling	₹ 550.0	30
CDR24	C Language	Denish Retchie	₹ 199.0	1
BBM999	You Will Be..	Bharat Bhagyajyoti	₹ 788.0	5
BK87G0	Tesla Backgrund	Biswaraj Basulai	₹ 999.0	8
SJBK34	My Autobiography	Suraj Biswal	₹ 899.0	67
REC78H	React	Google.inc	₹ 100.0	7
AMZ07B	AWS	Amazon.com	₹ 89.0	10
GGL65i	Google Big data	Google.inc	₹ 445.0	1
BK90Lt	Python	Aryan Barik	₹ 299.0	6
SPJ09A	Spring - Java	Rod Johnson	₹ 769.0	10

[Add New Book](#) [Back to Home](#)





BACK TO HOME

Book Requests

Book Name	Author	Requested By	Status	Request Date	Action
History Of India	ASI	RayanSS00	PENDING	2025-03-28 16:00:49	<button>REMOVE</button> <button>✓ APPROVE</button>
Elements Of Mathematics	CHSE, Odisha	RayanSS00	PENDING	2025-03-28 15:59:26	<button>REMOVE</button> <button>✓ APPROVE</button>

BACK TO HOME

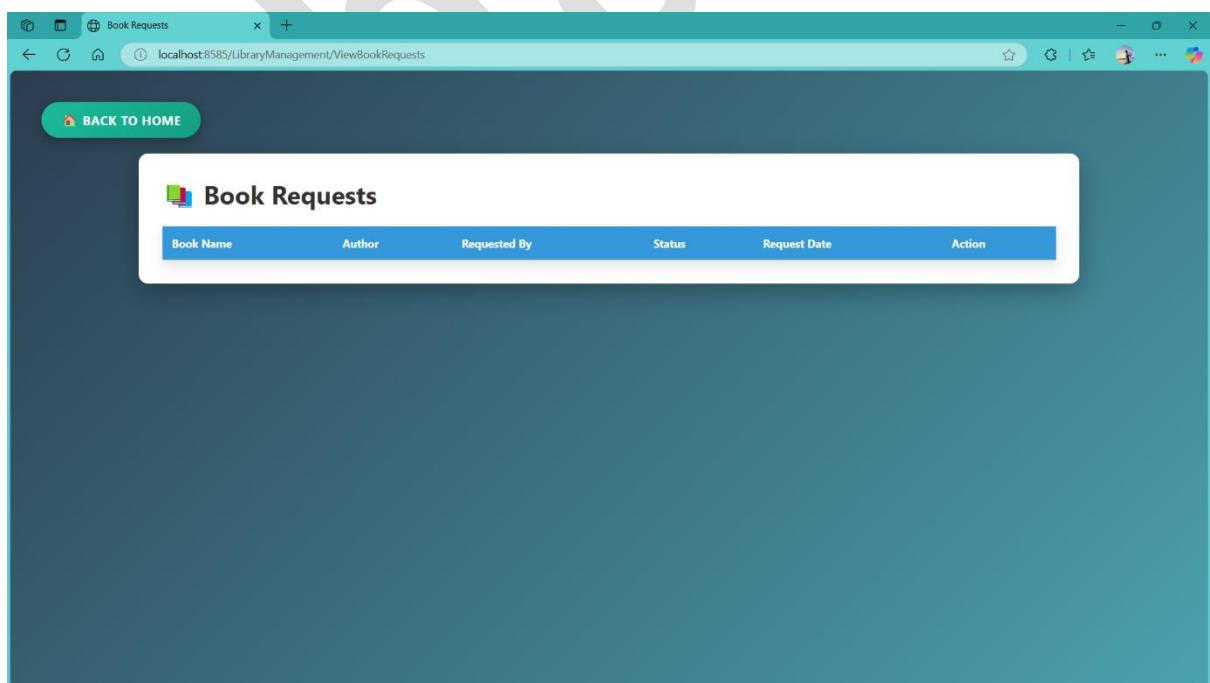
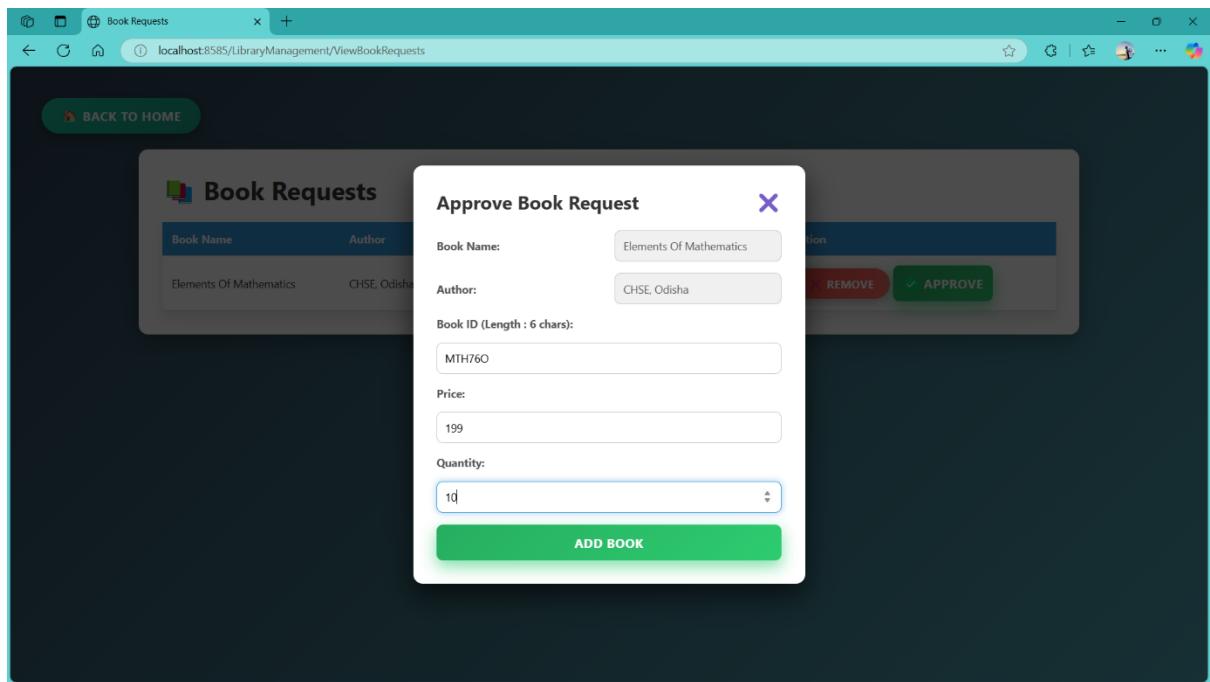
Book Requests

Book Name	Author	Requested By	Status	Request Date	Action
History Of India	ASI	RayanSS00	PENDING	2025-03-28 16:00:49	<button>REMOVE</button> <button>APPROVE</button>
Elements Of Mathematics	CHSE, Odisha	RayanSS00	PENDING	2025-03-28 15:59:26	<button>REMOVE</button> <button>APPROVE</button>

BACK TO HOME

Book Requests

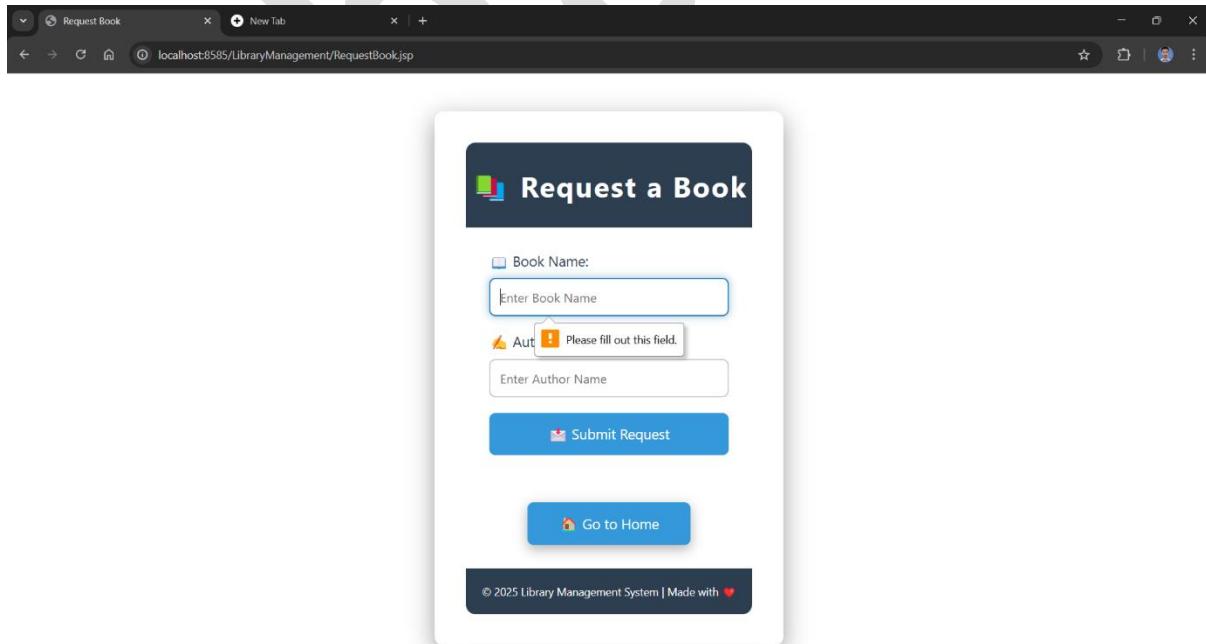
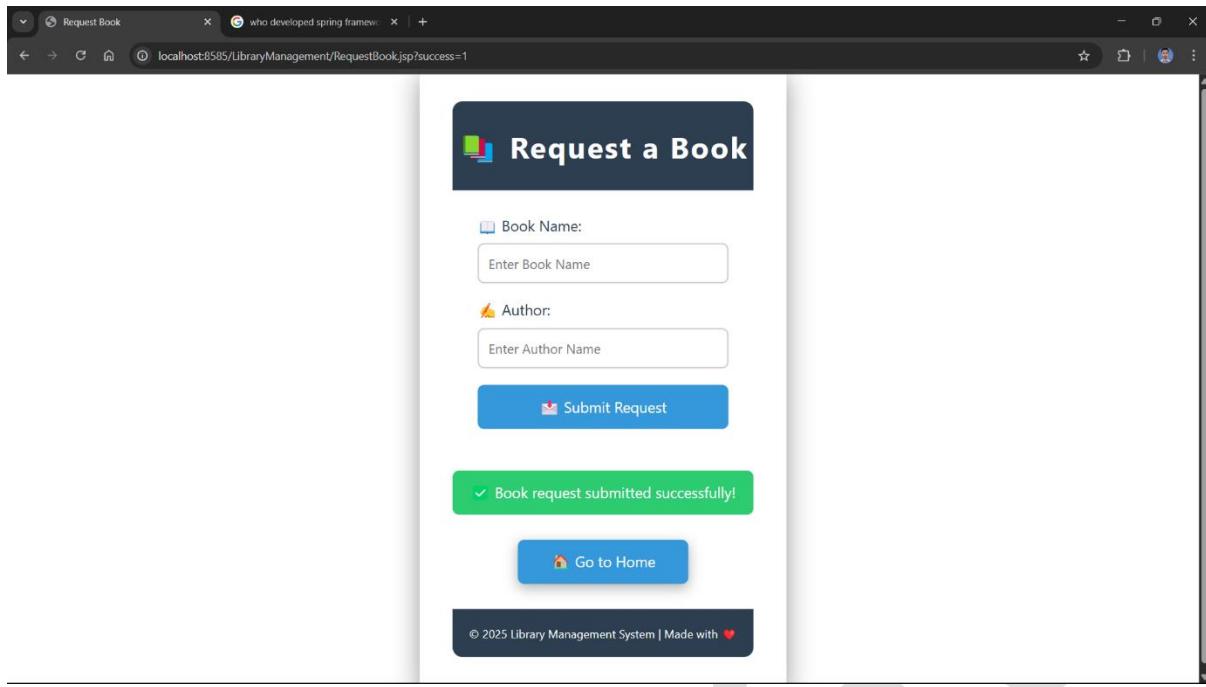
Book Name	Author	Requested By	Status	Request Date	Action
Elements Of Mathematics	CHSE, Odisha	RayanSS00	PENDING	2025-03-28 15:59:26	<button>REMOVE</button> <button>APPROVE</button>



📌 Book Requests

Test Case	Action	Expected Output	Actual Output	Status
Submit book request	Enter book name & author	SweetAlert: "Request submitted"	Request submitted alert	✓ Passed
Duplicate request	Request same book twice	SweetAlert: "Request already exists"	Duplicate alert	✓ Passed
Empty request form	Submit with empty fields	Validation error message	Validation message	✓ Passed

The screenshot shows a web browser window with the title 'Request Book'. The address bar indicates the URL is 'localhost:8085/LibraryManagement/RequestBook.jsp'. The main content is a 'Request a Book' form. It features two input fields: 'Book Name:' containing 'Elements Of Mathematics' and 'Author:' containing 'CHSE, Odisha'. Below these fields is a blue 'Submit Request' button. At the bottom of the form is a blue 'Go to Home' button. The footer of the page contains the text '© 2025 Library Management System | Made with ❤️'.



📌 View Book Requests

Test Case	Action	Expected Output	Actual Output	Status
View all book requests	Click on "My Requests"	Displays list of submitted requests	List displayed	<input checked="" type="checkbox"/> Passed
No book requests	View without submitting any	"No requests found" message	No requests alert	<input checked="" type="checkbox"/> Passed
Multiple requests	Submit multiple requests	Displays all submitted requests	All requests displayed	<input checked="" type="checkbox"/> Passed

The screenshot shows a web browser window titled "Request Book". The URL in the address bar is "localhost:8585/LibraryManagement/RequestBook.jsp?success=1". The main content area has a dark header with the text "Request a Book" and a small logo. Below the header are two input fields: "Book Name:" with placeholder "Enter Book Name" and "Author:" with placeholder "Enter Author Name". A blue button labeled "Submit Request" is below these fields. A green success message box is centered, containing the text "✓ Book request submitted successfully!". Below this message is a blue button labeled "Go to Home". At the bottom of the page, a dark footer bar contains the text "© 2025 Library Management System | Made with ❤️".

Request Book

localhost:8585/LibraryManagement/RequestBook.jsp

Request a Book

Book Name: Physics

Author: SS Aurora

Submit Request

Go to Home

© 2025 Library Management System | Made with ❤️

My Book Requests

Request ID	Book Name	Author	Status	Request Date
44	Physics	SS Aurora	PENDING	2025-03-28 16:12:25.463644

Back to Dashboard

© 2025 Library Management System | Made with ❤️

The screenshot shows a web browser window with the title "Request Book". The URL is "localhost:8085/LibraryManagement/RequestBook.jsp". The main content is a form titled "Request a Book" with the following fields:

- Book Name:** An input field containing "Enter Book Name" with a validation message: "Aut. Please fill out this field."
- Author Name:** An input field containing "Enter Author Name".
- Submit Request:** A blue button with a mail icon.
- Go to Home:** A blue button with a house icon.

At the bottom, there is a footer with the text "© 2025 Library Management System | Made with ❤️".

📌 Search Books

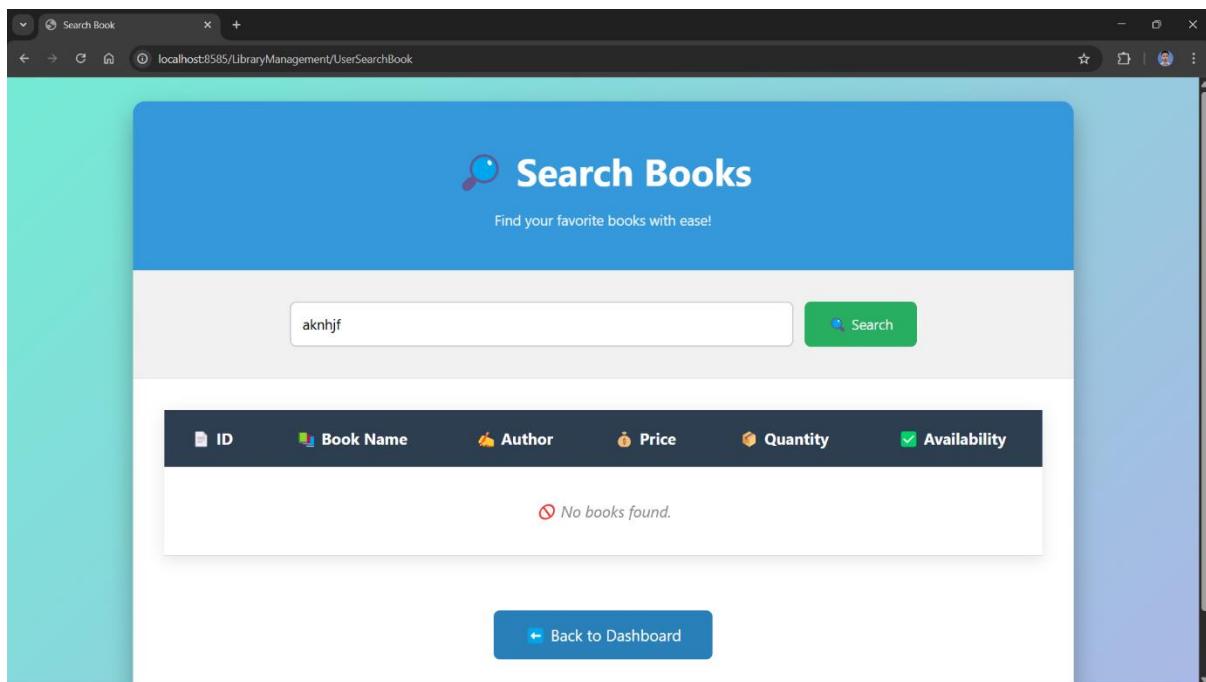
Test Case	Action	Expected Output	Actual Output	Status
Valid book name search	Enter existing book name	Displays matching books	Matching books displayed	✓ Passed
Invalid book name search	Enter non-existent book name	"No book found" message	No book found alert	✓ Passed
Empty search field	Click search without input	Validation message	Validation message	✓ Passed

A screenshot of a web browser window titled "Search Book". The URL is "localhost:8505/LibraryManagement/UserSearchBook". The main content area has a blue header with the title "Search Books" and a magnifying glass icon. Below it is a sub-header "Find your favorite books with ease!". There is a search bar with the input "Java" and a green "Search" button. Below the search bar is a table with columns: ID, Book Name, Author, Price, Quantity, and Availability. The table contains two rows of data:

ID	Book Name	Author	Price	Quantity	Availability
JJG072	Core Java	James Gosling	₹550.0	30	AVAILABLE
SPJ09A	Spring - Java	Rod Johnson	₹769.0	10	AVAILABLE

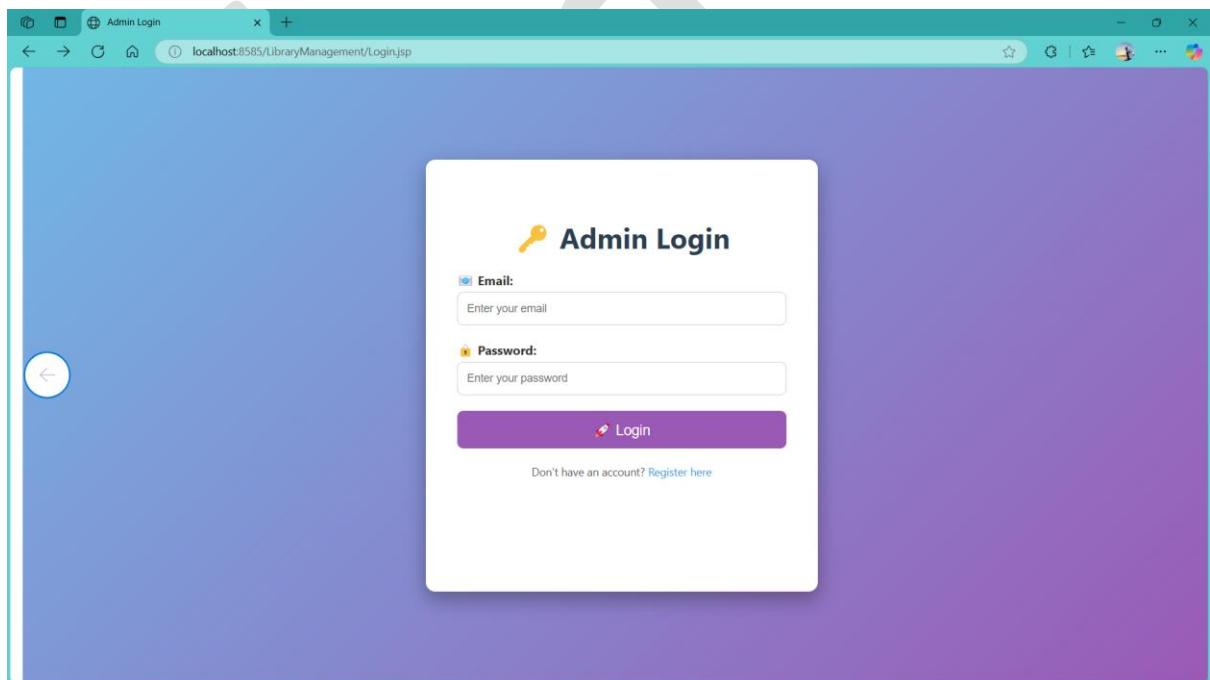
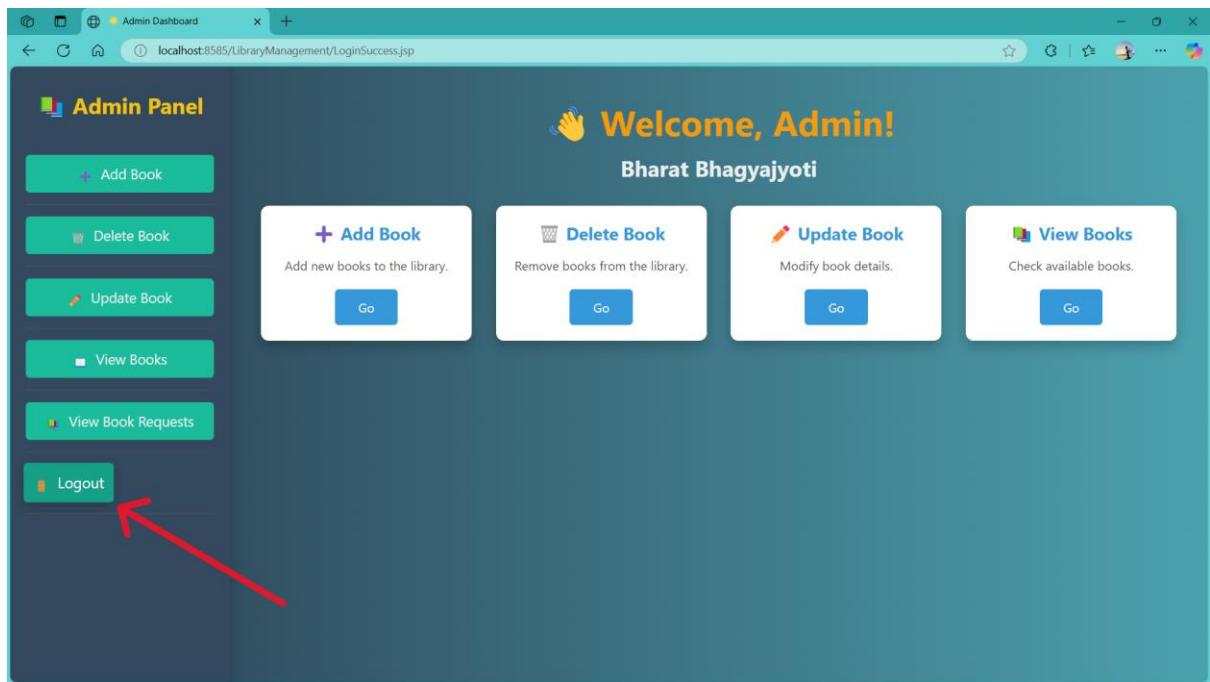
At the bottom of the page is a blue "Back to Dashboard" button.

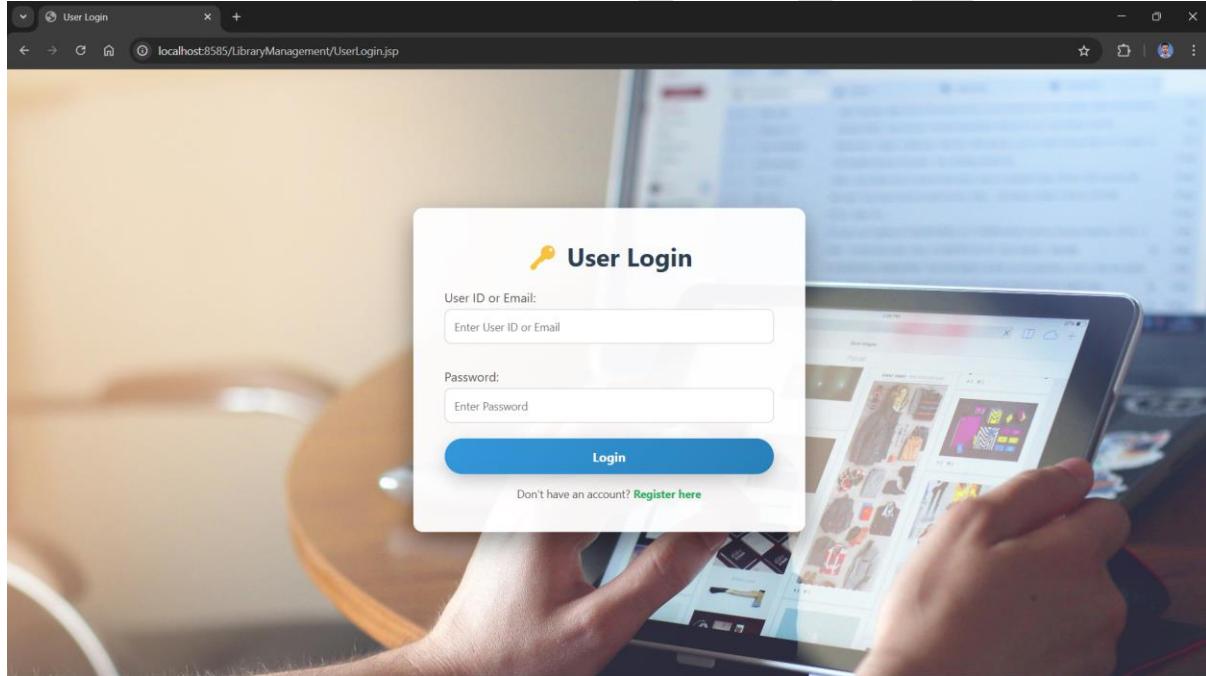
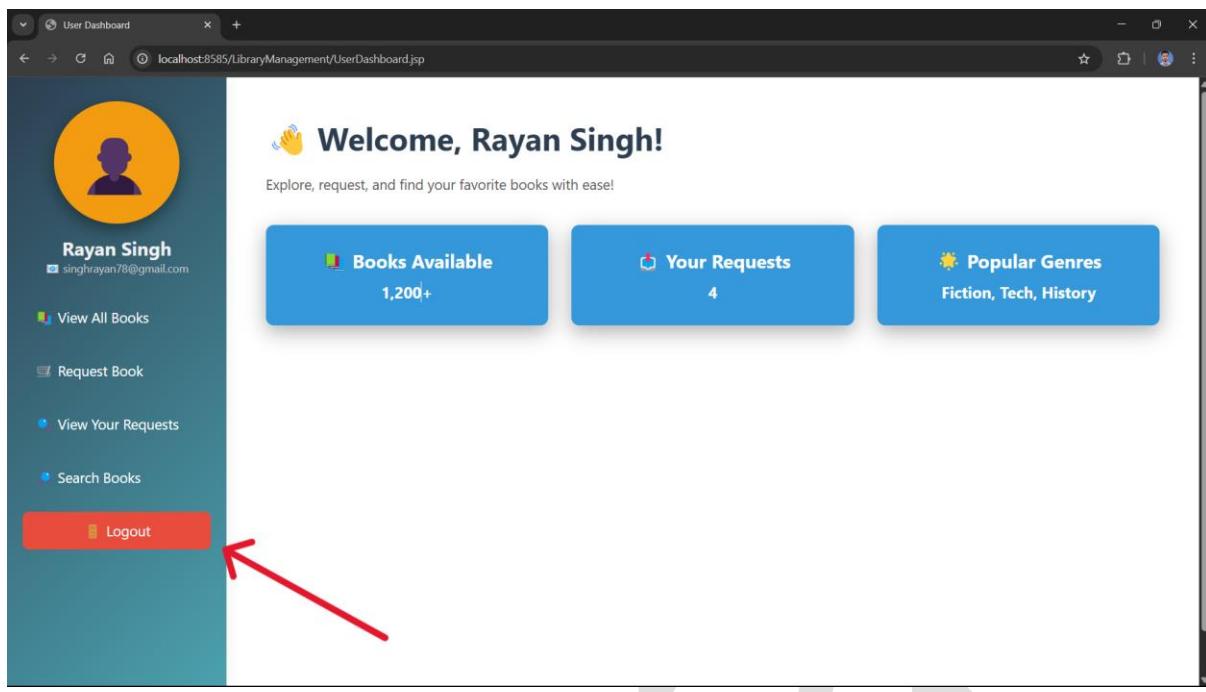
A screenshot of a web browser window titled "Search Book". The URL is "localhost:8505/LibraryManagement/SearchBook.jsp". The main content area has a blue header with the title "Search Books" and a magnifying glass icon. Below it is a sub-header "Find your favorite books with ease!". There is a search bar with the placeholder "Enter book name..." and a green "Search" button. A red error message box says "Please fill out this field." Below the search bar is a table with columns: ID, Book Name, Author, Price, Quantity, and Availability. The table is empty and displays the message "No books found." At the bottom of the page is a blue "Back to Dashboard" button.



📌 Logout and Session Management

Test Case	Action	Expected Output	Actual Output	Status
Valid logout action	Click logout	Redirect to login page	Redirect successful	✓ Passed
Back navigation after logout	Press back button after logout	Redirect to login page	Redirect successful	✓ Passed
Session timeout	Inactive session timeout	Auto logout, redirect to login	Redirect successful	✓ Passed





7.3 Test Results

- All **core functionalities** were validated successfully.
- **Edge cases** (e.g., duplicate emails, SQL errors) were handled properly.
- **Performance** was consistent with minimal latency during CRUD operations.
- **Session handling** and **navigation** were smooth and secure.

8. Conclusion

The Library Management System provides efficient management of library operations, enabling admins to perform CRUD operations and users to request books with ease. It follows the MVC architecture, uses Oracle SQL for database management, and employs BCrypt hashing for secure authentication. The system is secure, responsive, and ready for future enhancements.