Fall 2023

Library Management System

Katyaini Manne
Governors State University

LIBRARY MANAGEMENT SYSTEM

M.Katyaini
By
B.Tech , JNTUH College Of
Engineering, 2020

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484
2023

ABSTRACT

As it is getting difficult to manage the library operations physically, there is a need of automating all those operations. We all know that there is lot of improvement in the technology, by making use of technology we want to build a dynamic web application with some intelligent elements. As this application is used by 3 users like librarian, student and staff. Every user is having their own operations like book entry, book search, view book, delete book, book issue, book return etc.

Apart from this we will be implementing data analysis and also inventory analysis. For example, there will be the data set which contains the information of book issue. We can retrieve the required data like what are the books borrowed by students, what are the books there in issue, is book available or not etc. By getting such information, library can able to keep such books in the library so that it can be useful to the students. This will be implemented as a feature to the librarian to know the information.

This application is implemented using visual studio 2022 version, with MVC architecture. We are using SQL server to store and retrieve the data.

Table of Content

Contents

1. Project Description 1
1.1 Competitive Information 1
1.2 Relationship to Other Applications/Projects 1
1.3 Assumptions and Dependencies 2
1.4 Future Enhancements 2
1.5 Definitions and Acronyms 3

2. Project Technical Description 3
2.1 Application Architecture 3
2.2 Application Information flows 4
2.3 Interactions with other Projects (if Any) 5
2.4 Interactions with other Applications 5
2.5 Capabilities 5
2.6 Risk Assessment and Management 6

3. ProjectRequirements 6
3.1 Identification of requirements 6
3.2 Operations, Administration, Maintenance and Provisioning (OAM&P) 7
3.3 Security and Fraud Prevention 7
3.4 Release and Transition Plan 8

4. Project Design Description 8

5. Project Internal/external Interface Impacts and Specification 19

6. Design Units Impacts 20
6.1 Functional Area A/Design Unit A 20
6.1.1 Functional Overview 20
6.1.2 Impacts 20
6.1.3 Requirements 20

7. Openissues 21

8. Acknowledgements 21

9. References 21

10. Appendices 22

Library management system 22

1. *Project Description*

The present situation around us is everything gets digitalized, and we are going to discuss in this system about Library Management Application includes three users they are: Admin, Librarian, User. Everyone in the application has their own roles and responsibilities, when we are going deep into the application Librarian and User will have registration and login process, but admin will have only login process, every user will have their own console after login into the system.

When admin gets login, he can manage all the details regarding the application including Add/Update/Delete Book, Member Management, Add/update/delete author details, add/update/delete publisher details, he can view registered librarian info, registered users' info from his console. Librarian will add/update/delete books when the new arrivals are in market, he can view user details, he can issue the books as per the user requirement so that there will be changes in the stock of available books. [1] User can login and view all books that are available in the library. Admin will update the user profile status, and he will track the information regarding all books and their stocks.

1.1 Competitive Information

Our program distinguishes out in the quickly changing field of digitalized library management systems due to its user-friendly interface and three separate user roles: Administrator, Librarian, and User. Admin powers allow you to manage members, add, update, and delete books, and keep an eye on publisher and author information. They also give you total control over the program. A comprehensive view of registered users and librarians is available through the admin panel. By overseeing the library's fluctuating inventory, librarians perform a crucial role. They may easily add, edit, or remove books, so the collection is kept up to date with new releases. Along with facilitating user interactions, librarians also keep track of current stock levels and provide books in response to requests from users.

After completing an easy-to-use login process, users are granted access to a comprehensive catalogue of books that are accessible within the library. The admin can also monitor user profiles.

Thus, our Library administration Application guarantees clear and effective library operations by integrating strong administrative control, collection administration by librarians, and a smooth user interface for visitors.

1.2 Relationship to Other Applications/Projects

Our Library Management Application fosters a collaborative and integrated library management experience by easily integrating with other apps and initiatives within the larger digital ecosystem. The architecture of the system facilitates data exchange and interoperability, strengthening its connections with other relevant applications:

(i) *Integrated Database Systems:* Our program may be connected to pre-existing database systems, including those in charge of larger organizational databases or educational establishments. This

(ii) connection makes it easier for users to share information and guarantees data integrity across several platforms [2].

(iii) *Educational Management Systems:* Information may move more quickly when there is cooperation with educational management systems. It is possible to synchronize faculty and student information, providing users who are a part of educational institutions and the library system with a seamless experience.

1.3 Assumptions and Dependencies

Assumptions:

(i) Consistent Internet Access: For the Library Management Application to function properly, each user must have a steady internet connection in order to provide smooth access to the system's functionalities.

(ii) Data Accuracy from External Sources: Integration with external databases, such author and publisher databases, relies on the accuracy and currentness of the data it receives. The dependability of the library system may be impacted by any disparities in outside data.

(iii) User Compliance: It is expected of users to abide by the policies and procedures of the system, which include providing correct information when registering and handling borrowed documents with care. The efficiency of the procedures used in library administration may be impacted by noncompliance.

Dependencies:

(i) Database Integration: To guarantee the correctness and completeness of the information, the program depends on smooth integration with external databases, such as author, publisher, and educational management systems.

(ii) User Training: Users must get sufficient training on the features and functions of the system, particularly librarians and administrators, for successful deployment. Sessions on user training are necessary for effective system use.

(iii) Ongoing Technical Support: Resolving technical problems, attending to user concerns, and guaranteeing the application's seamless functioning over time all depend on a solid technical support infrastructure.

1.4 Future Enhancements

The Library Management Application will soon be enhanced with extensive analytics for user preferences and trends in book popularity. Users' experiences can be improved overall by providing them with personalized book suggestions through integration with machine learning algorithms. The launch of a mobile application version will offer services and the library catalogue while on the go. Using barcoding or RFID technology can improve productivity by streamlining the return and checkout procedures for books. Integrating social media platforms might promote community involvement by enabling users to exchange evaluations and recommendations for books. Furthermore, experimenting with augmented reality (AR) and virtual reality (VR) technologies for interactive book displays and virtual library tours may modernize the user experience and make the library more immersive and interesting for visitors [3]. The aforementioned improvements are intended to improve the functionality, user experience, and technological innovation of the application within the dynamic field of digital library management.

1.5 Definitions and Acronyms

AR- Augmented Reality
VR- Virtual Reality
RFID- Radio-Frequency Identification

2. Project Technical Description

Modern technologies are used by the Library Management System to provide a reliable and effective solution. The backend, which was created with Visual Studio 2022, uses SQL Server 2019 for smooth data administration. The frontend offers a dynamic and responsive user experience. It is an ASP.NET Web App MVC version 6.0. This technological stack guarantees stability, security, and flexibility for an advanced library management experience with long-term support.

2.1 Application Architecture

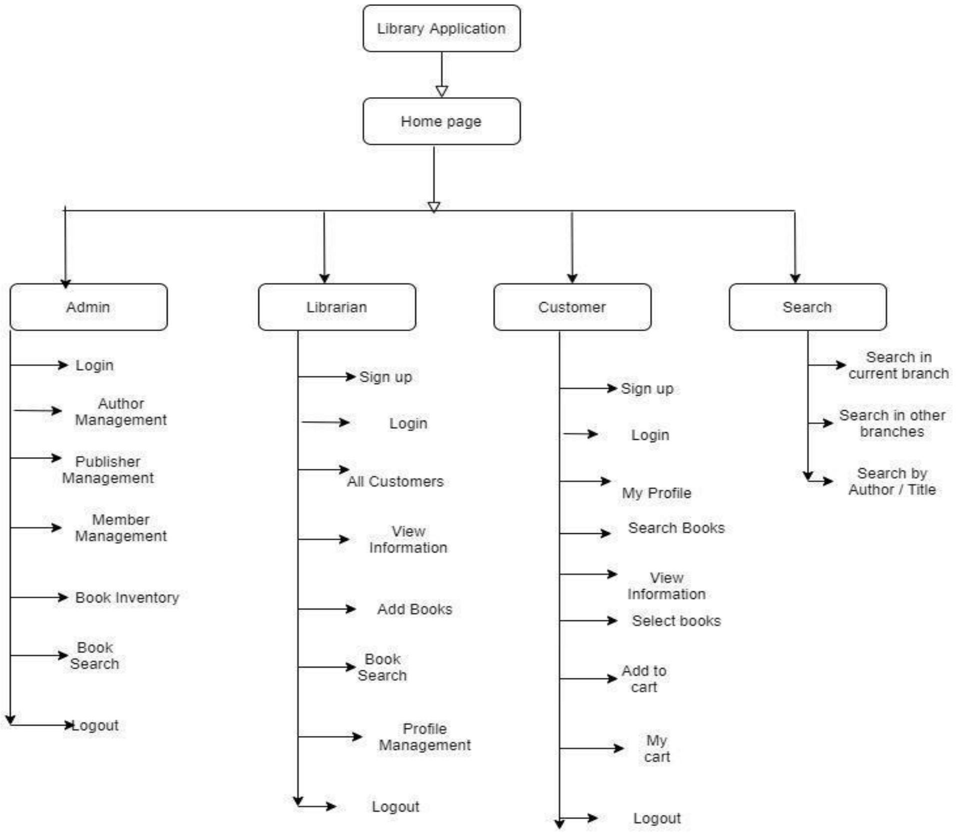
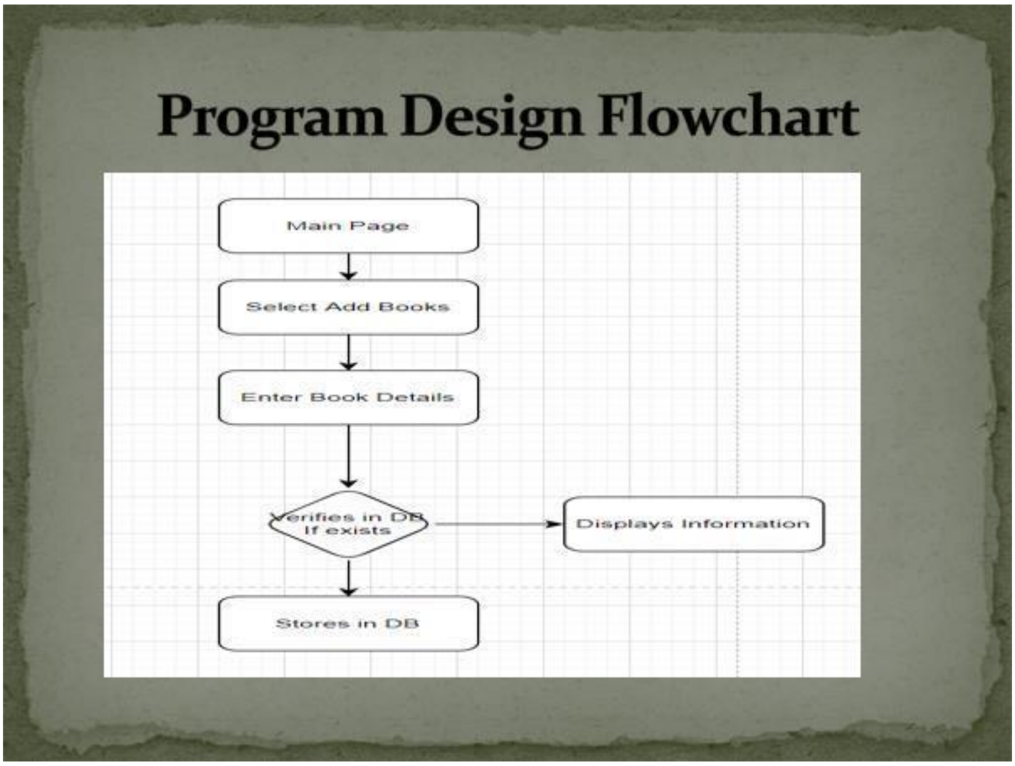


Figure 1: Application Architecture

2.2 Application Information flows



Program Design Flowchart

Figure 2: Flow Chart

2.3 Interactions with other Projects (if Any)

No Interactions with any other projects

2.4 Interactions with other Applications

No interaction with any other applications

2.5 Capabilities:

- User Authentication:

Enable secure login for Admin, Librarian, and User roles.

- Admin Capabilities:

Manage and update application details.
Add, update, or delete books.
Member management.
Author and publisher details management.
View registered librarian and user information.

- Librarian Capabilities:

Add, update, or delete books based on new arrivals.
View user details.
Issue books according to user requests.
Monitor and update stock levels.

- User Capabilities:

User-friendly login process.
View all available books in the library.

- Database Integration:

Integrate with external databases for author, publisher, and educational systems.
Interoperability:
Collaborate with other applications and projects in the digital ecosystem.

- Security Measures:

Implement robust security protocols to protect user data.

- Analytics and Reporting:

Provide insights into user behavior, popular book genres, and library usage.

- Mobile Access:

Develop a mobile application for on-the-go access to the library catalog.

- RFID/Barcoding:

Streamline book checkout and return processes with RFID or barcoding technology.

- Machine Learning Integration:

Offer personalized book recommendations to users.

- Social Platform Integration:

Allow users to share book recommendations and reviews on social platforms.

- Virtual/Augmented Reality Features:

Explore VR or AR for virtual library tours and interactive book displays.

- Technical Support:

Provide ongoing technical support for issue resolution.

- User Training:

Conduct training sessions for users, librarians, and administrators.

- Continuous Updates:

Ensure the system stays up-to-date with the latest technology and data.

- Single Sign-On Functionality:

Allow users to access multiple systems with a single set of credentials.

2.6 Risk Assessment and Management

- Technical Difficulties:

Risk: Possible problems with the system that might arise during upgrades or integration [6].
Mitigation: Have a rollback strategy in place in case there are unanticipated technical difficulties, and test everything thoroughly before implementing it.

- Security Concerns with Data:

Risk: Perils pertaining to the safety of confidential and user data.
Mitigation: Use strong encryption procedures, conduct frequent security audits, and keep up with emerging security threats.

- Adaptations to Regulations:

Risk: Modifications to privacy and data protection laws that might impact system compliance.
Mitigation: Keep abreast of regulatory changes, carry out routine compliance checks, and modify the system as necessary.

3. Project Requirements

3.1 Identification of requirements

Admin Module:

- ➢ Login
- ➢ Add/update/delete author
- ➢ Add/update/delete publisher
- ➢ Add/update/delete book
- ➢ Add/update/delete branch
- ➢ View/delete Member
- ➢ Add Journal/conference
- ➢ View Librarian Info
- ➢ Logout

Librarian Module:

- ➢ Login
- ➢ Add/update/delete book
- ➢ Book issue
- ➢ Book Return
- ➢ Logout

User module:

- Signup
- Login
- View all books
- View conferences/Journal
- Reserve/Return Books
- View/update profile
- Logout

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

Operations:
Goal: Assure the Library Management System's regular efficiency and operation.
Tasks:

- Track system efficiency.
- Keep an eye on user transactions and interactions.
- Assure correctness and integrity of the data.

Administration:
Goal: Oversee and control the several Library Management System components.
Tasks:

- Managing user accounts.
- Grant administrators, librarians, and users access privileges.
- Monitor the setups and configurations of the system.

Maintenance:
Goal: Maintain and improve the system's long-term performance.
Tasks:

- Update software components on a regular basis.
- Carry out system restores.
- Apply security updates and fixes.
- Respond quickly to technical concerns.

Getting ready:
Goal: Make it easier to configure and set up system resources.
Tasks:

- Introducing administrators and new users.
- setting up new software or hardware components.
- Controlling access credentials and licenses.

3.3 Security and Fraud Prevention

Strong security protocols are essential for the Library Management System. Several levels of protection are used to stop fraud and protect user information [7]. By using encryption techniques, data transmission is strengthened and sensitive information is shielded from unwanted access. Constant observation identifies irregularities and possible security problems, allowing for quick action to reduce hazards [4]. Frequent security audits evaluate system vulnerabilities and guarantee adherence to rules and industry standards. Unauthorized entrance is restricted via user authentication systems, such as secure login procedures and access controls.

The prevention of fraud is achieved by closely observing user behavior and spotting odd trends or inconsistencies. Proactive methods of reducing fraud, including anomaly detection algorithms, rely on early detection techniques. By emphasizing security recommended practices among users, education and awareness initiatives lessen the possibility of social engineering attacks. The Library Management System maintains a strong defense against fraudulent activities, unauthorized access, and data breaches by keeping up with emerging security risks and adjusting preventative measures accordingly. This fosters a secure environment that is safe for both administrators and users.

3.4 Release and Transition Plan

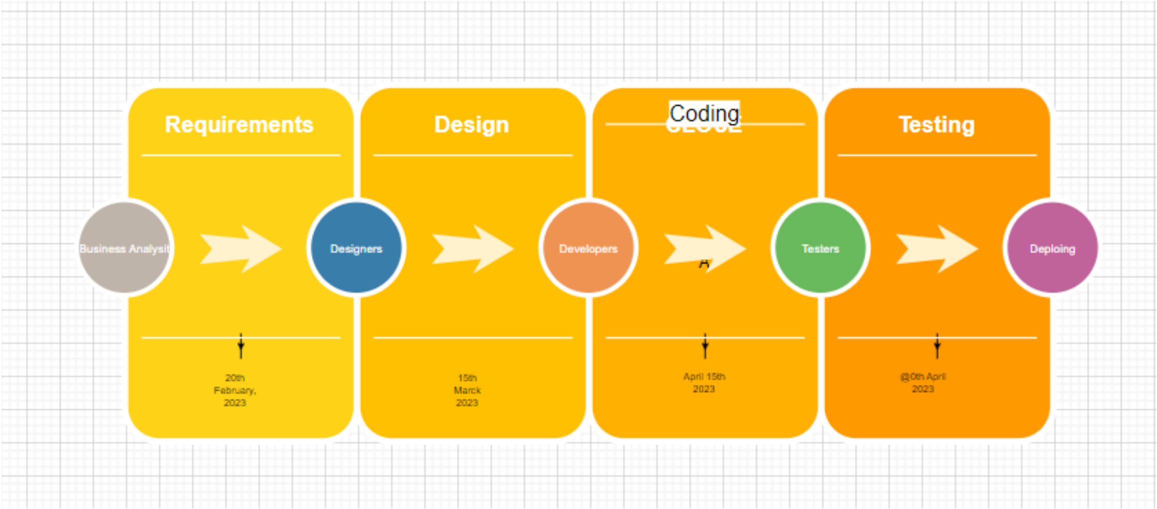


Figure 3: Design Plan chart

4. *Project Design Description*

In our Library Management Application, when we execute the application, it will show the main page which contains several options that users get registered and login within the application and every user will have some operations and communication between them including admin will add/update/delete book, those books will be viewed by both librarian and users. In the same manner Librarian will add new arrivals in market those details will be stored in the database and retrieved from the database [8].

a. Database Design:

Database is very important for our application because every user detail is stored and retrieved from the database, and we can also update or delete from database [9]. Each operation will have separate tables and they contain primary keys and foreign keys so that relation will be available between the tables. Authentication has been enhanced for user so that it would be helpful to improve more users to our application.

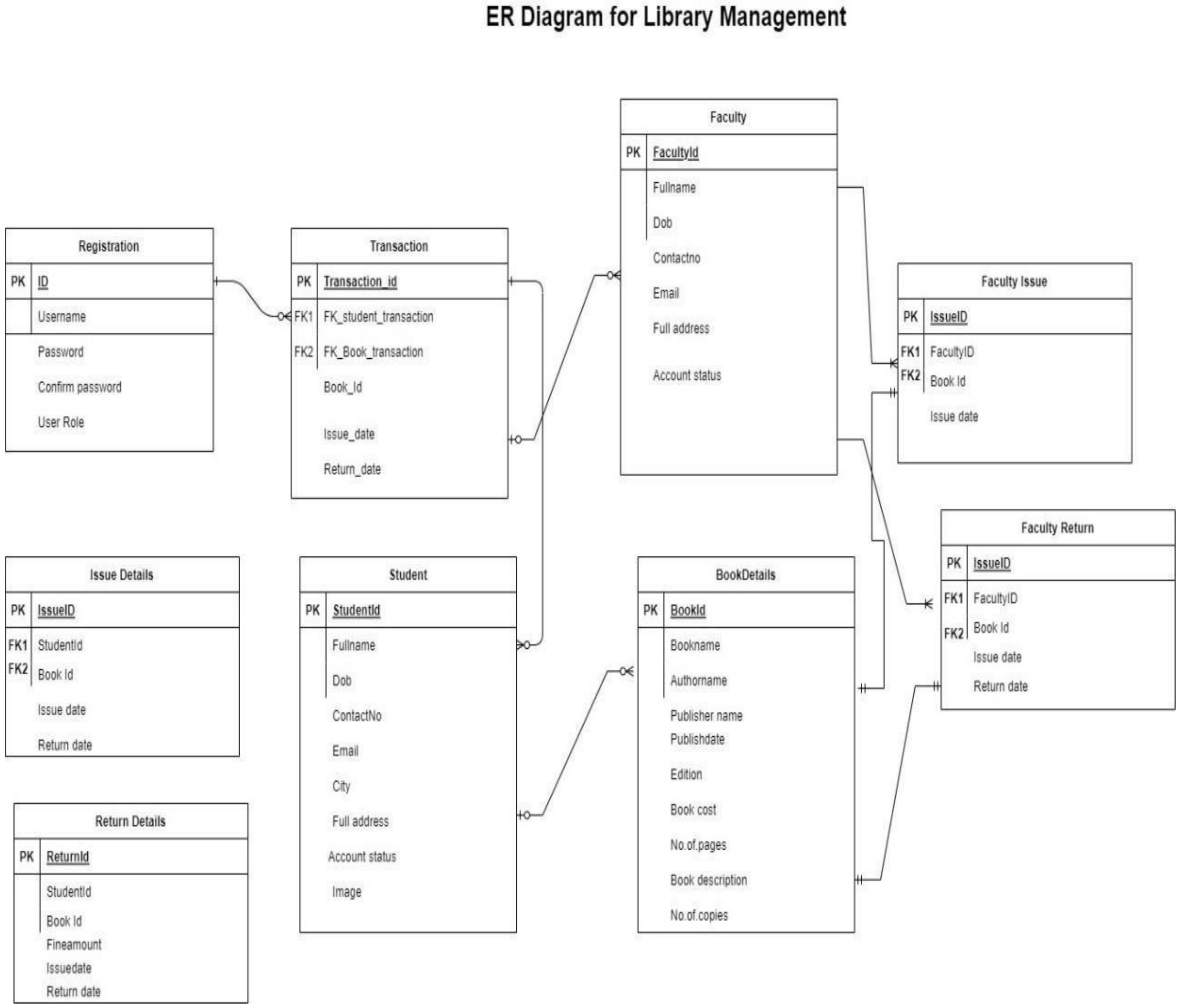


Figure 4: ER Diagram

Following are the user interfaces which presents the user operations that need to be performed by the application. Following user interface is the home page of the application.

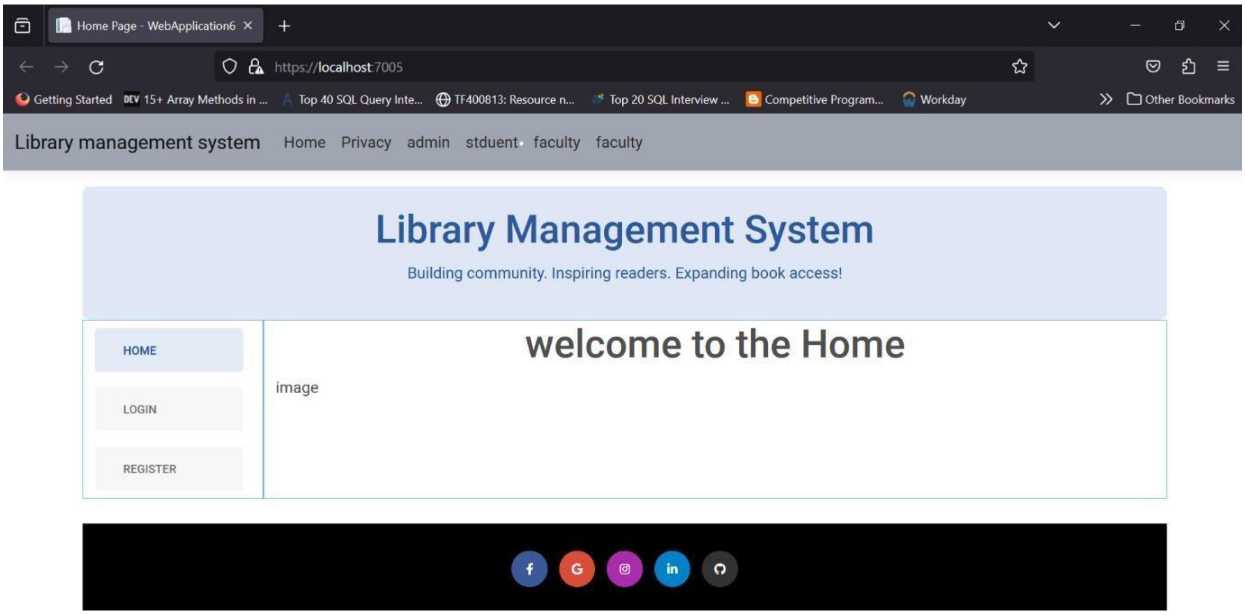


Figure 5: Dashboard

Following user interface is used to register with the application

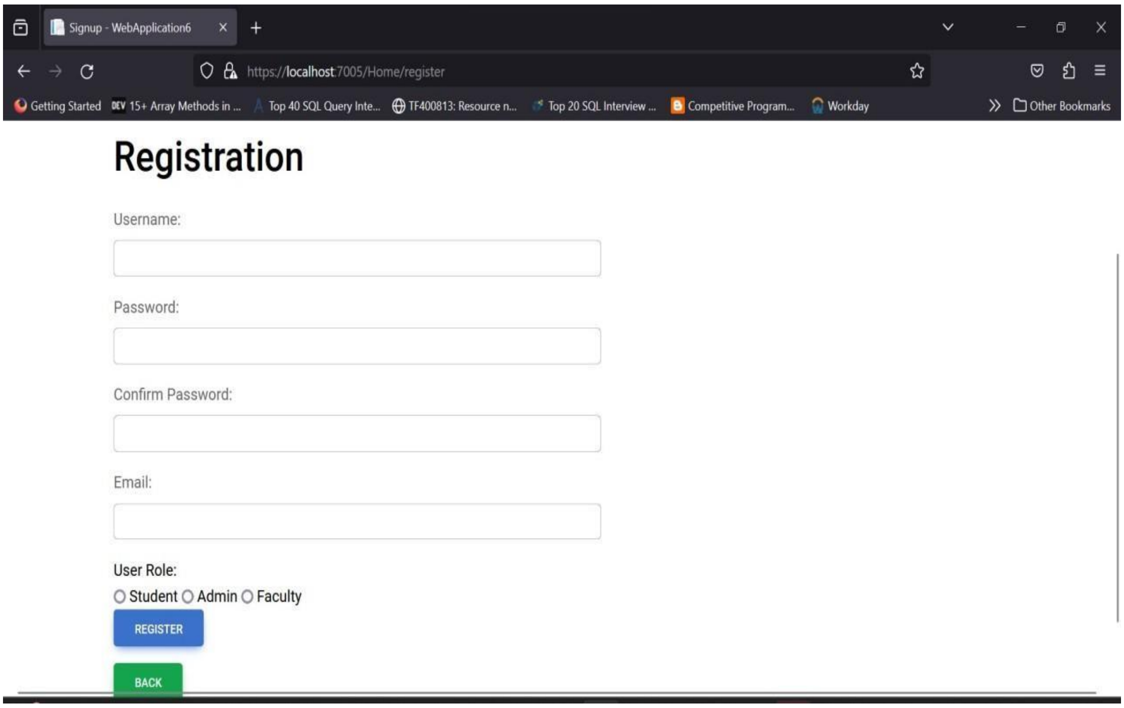
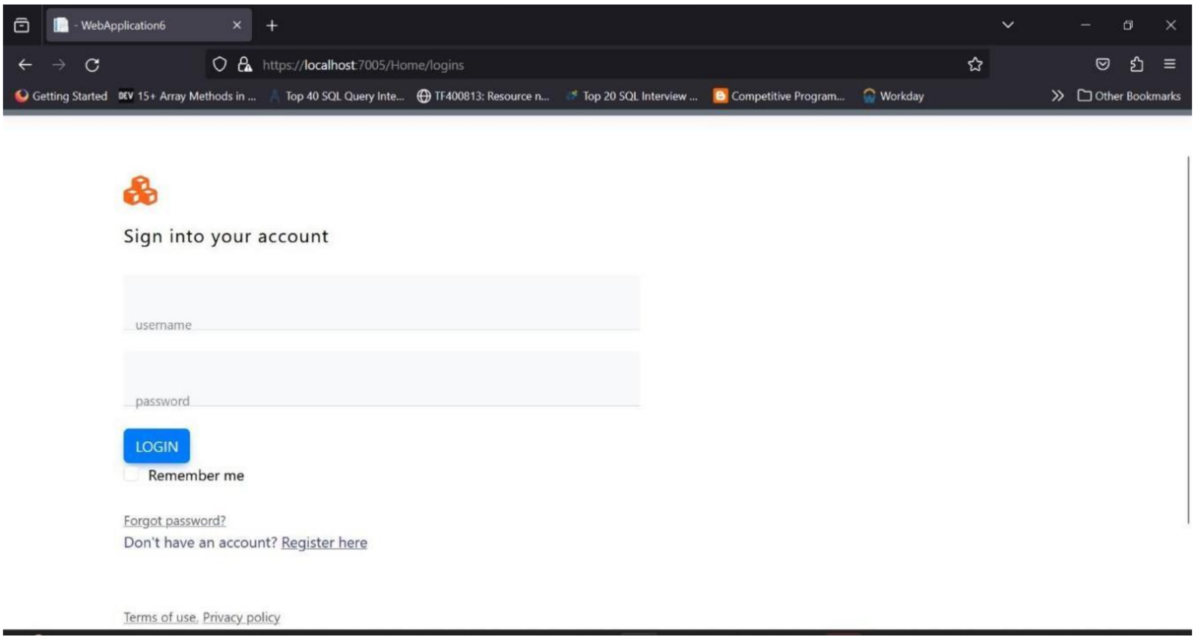


Figure 6: Registration form



Sign into your account

Don't have an account? Register here

Figure7: login page

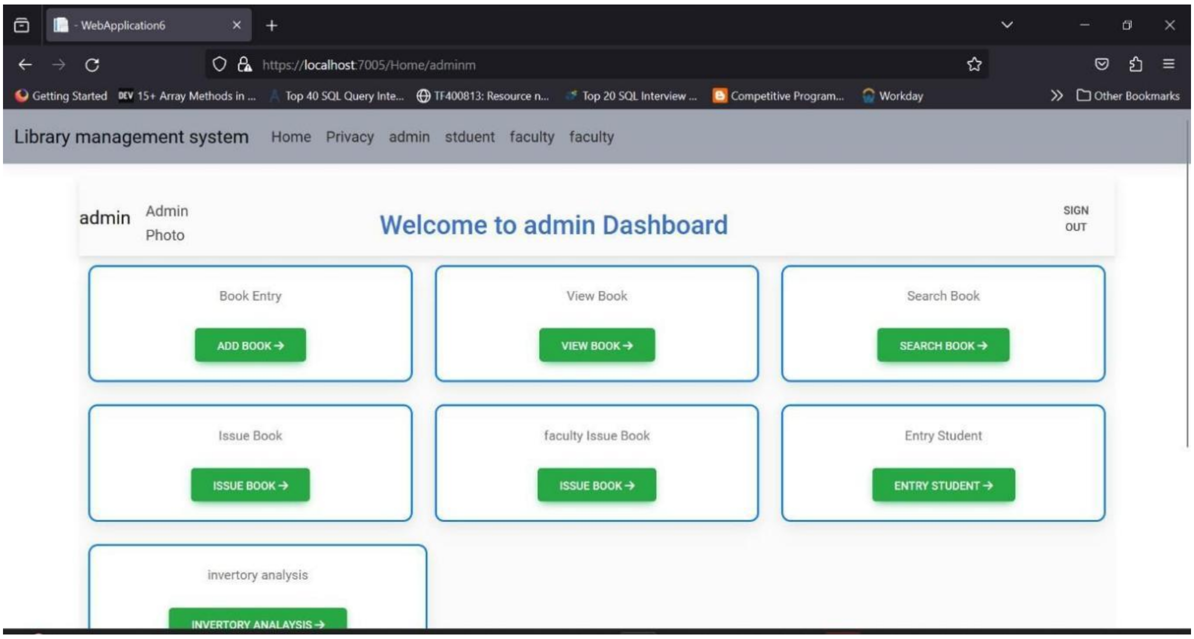
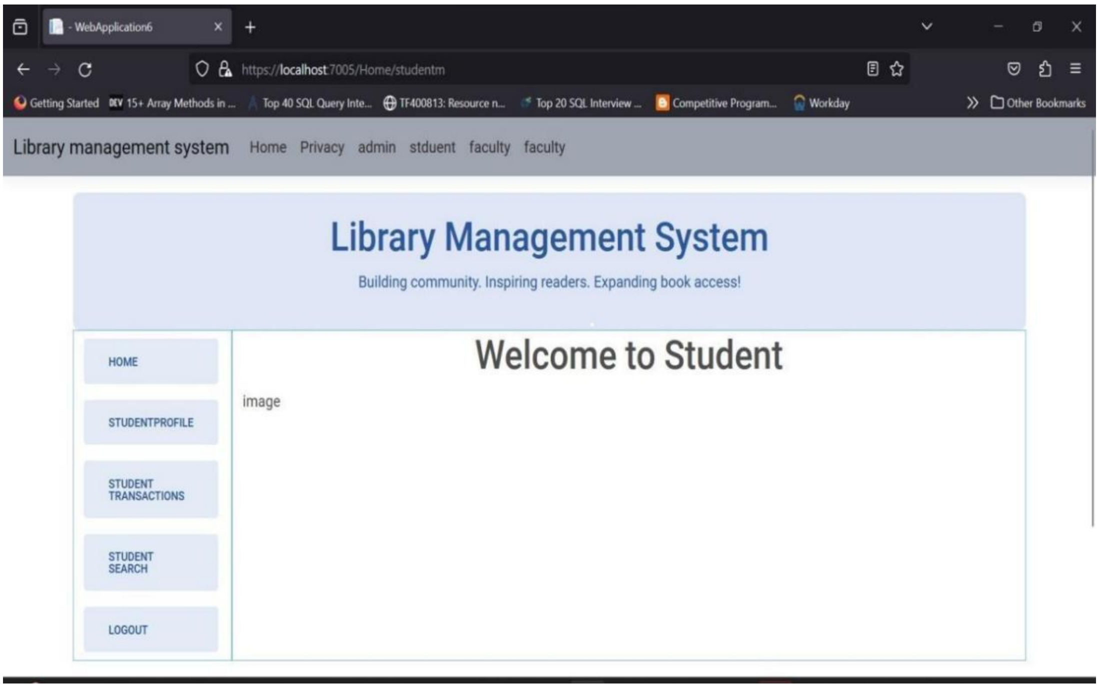


Figure- 8**:** Dashboard of Admin

- Above screenshot tell about the admin dashboard. when the admin logged in to the page and it describe about the admin operation.



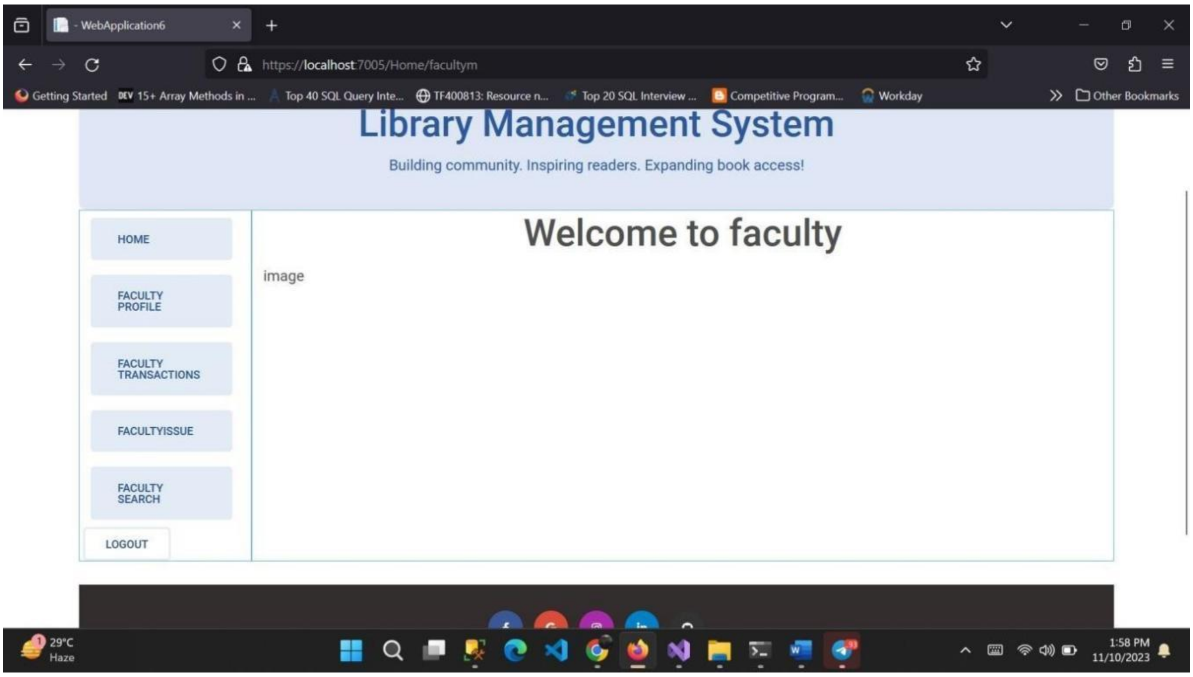
Library Management System

Building community. Inspiring readers. Expanding book access!

Welcome to Student

Figure 9: Student of Admin

- Above screenshot describes about the student home page which the student operations like view, search and transactions of the student.



Library Management System

Building community. Inspiring readers. Expanding book access!

Welcome to faculty

Figure 10: Dashboard of Faculty

- Above screenshot describe about the faculty page when the faculty has logged into the page and operation performed by the faculty like view ,search ,faculty issue of books and transactions.

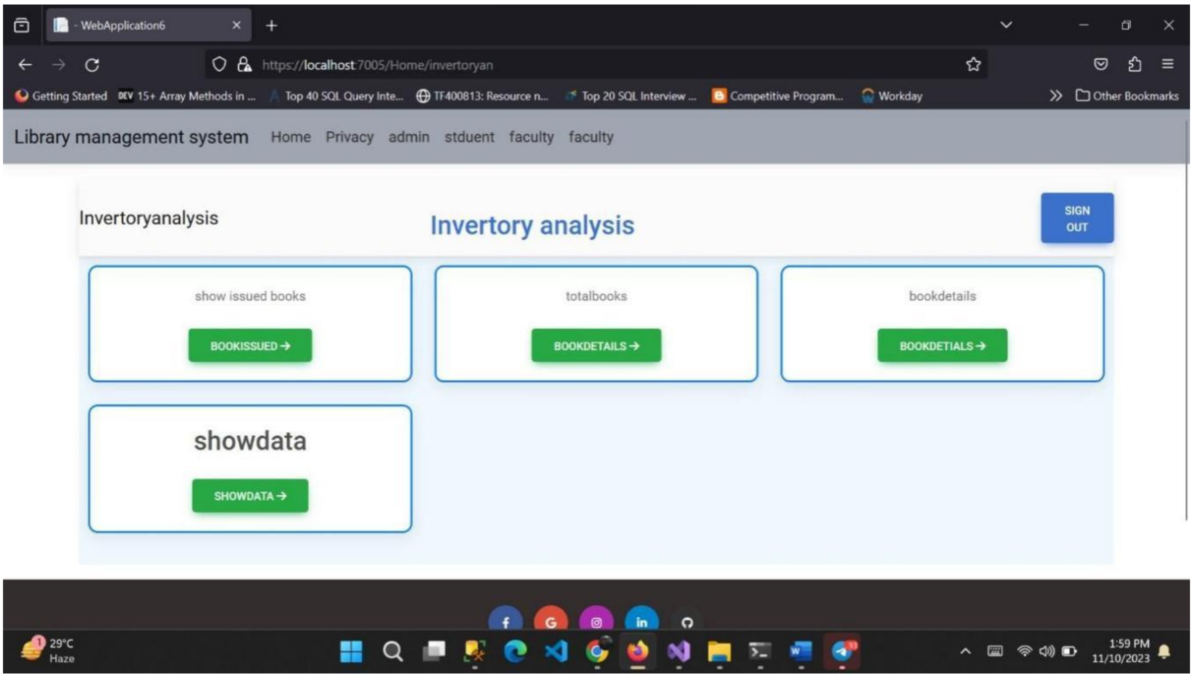
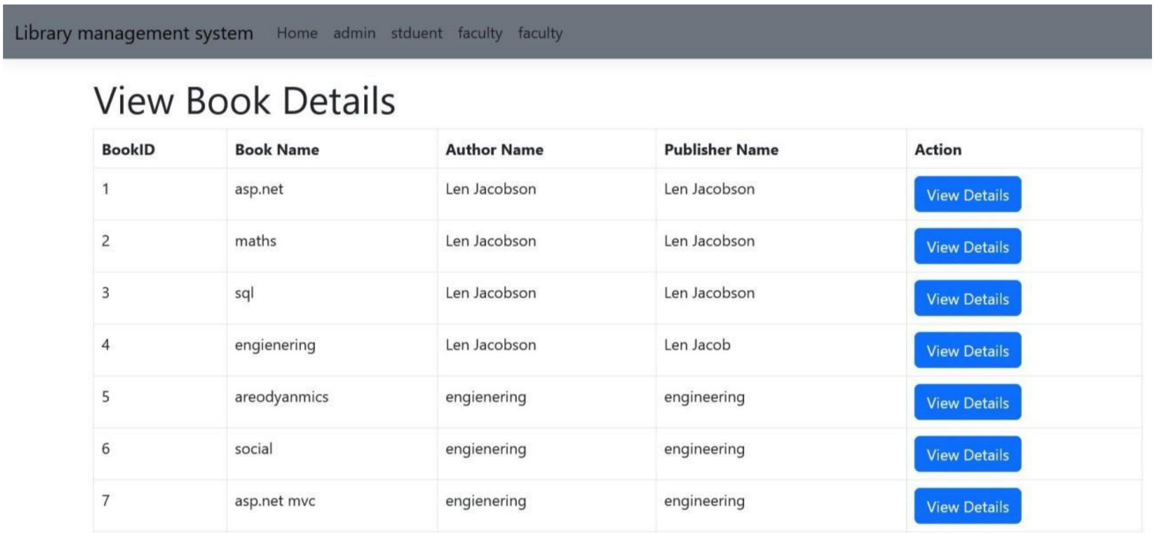


Figure 11**:** Inventory analysis

- ➢ Above screenshot describe about the inventory analysis where the analysis of the books and data of the admin ,student ,faculty should be shown.

View books

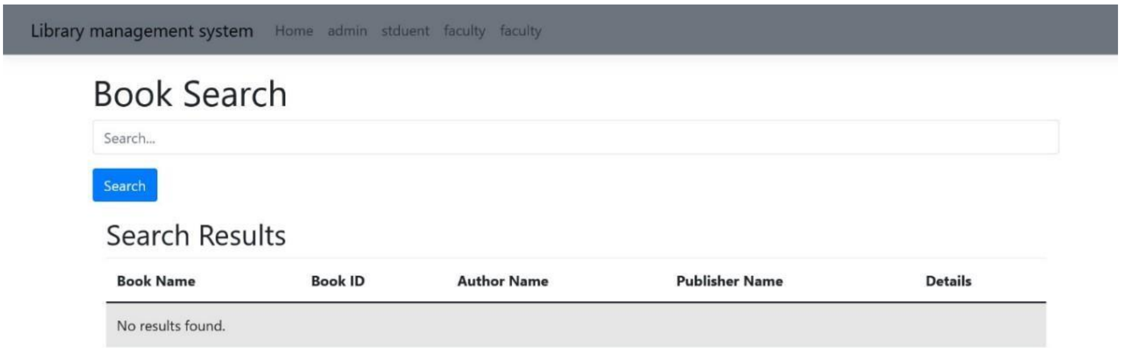


View Book Details

BookID	Book Name	Author Name	Publisher Name	Action
1	asp.net	Len Jacobson	Len Jacobson	View Details
2	maths	Len Jacobson	Len Jacobson	View Details
3	sql	Len Jacobson	Len Jacobson	View Details
4	engienering	Len Jacobson	Len Jacob	View Details
5	areodyanmics	engienering	engineering	View Details
6	social	engienering	engineering	View Details
7	asp.net mvc	engienering	engineering	View Details

Figure 12: View books

- ➢ Above screenshot describes to view details of the books and it also shows the information of a single book when we click view details button.



Book Name	Book ID	Author Name	Publisher Name	Details
No results found.				

Figure 13: Search book

- ➢ Above screenshot is to describe about the search of the each book from the stored book in the database.

Book issue

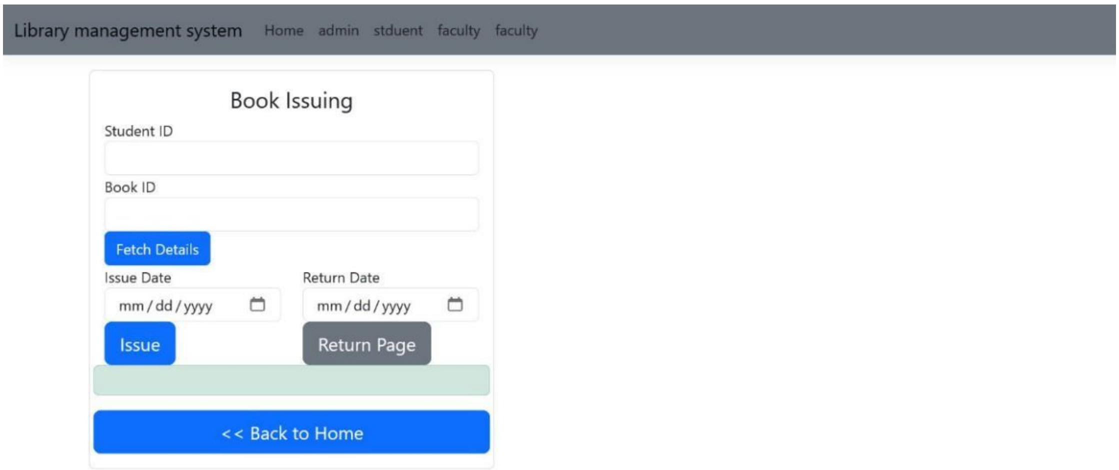


Figure 14: book Issue

- ➢ Above screenshot is use to describe about how the book issue operation to a student is done.

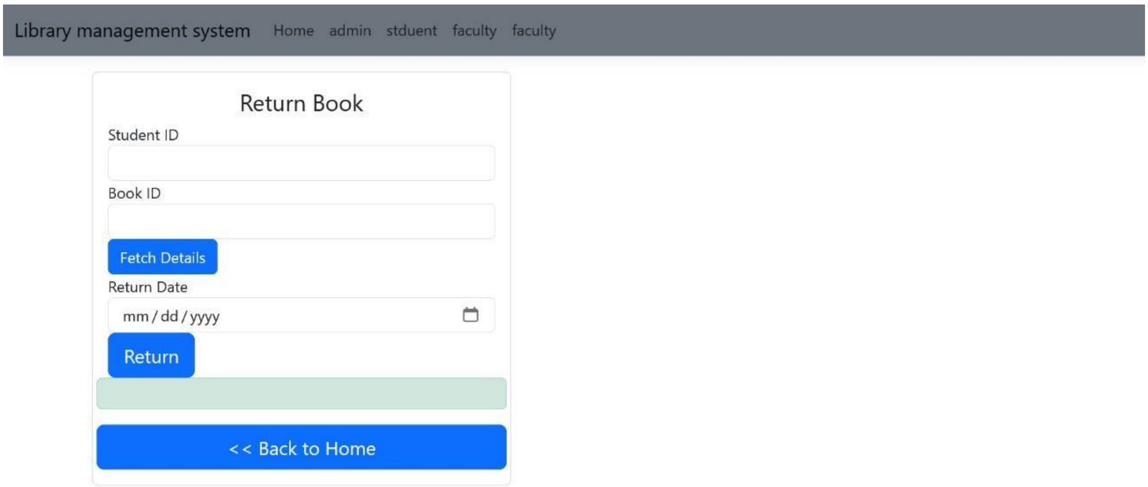


Figure 15: Book Return

- ➢ These screenshot describe about the return of the book which ever issued book to the student.

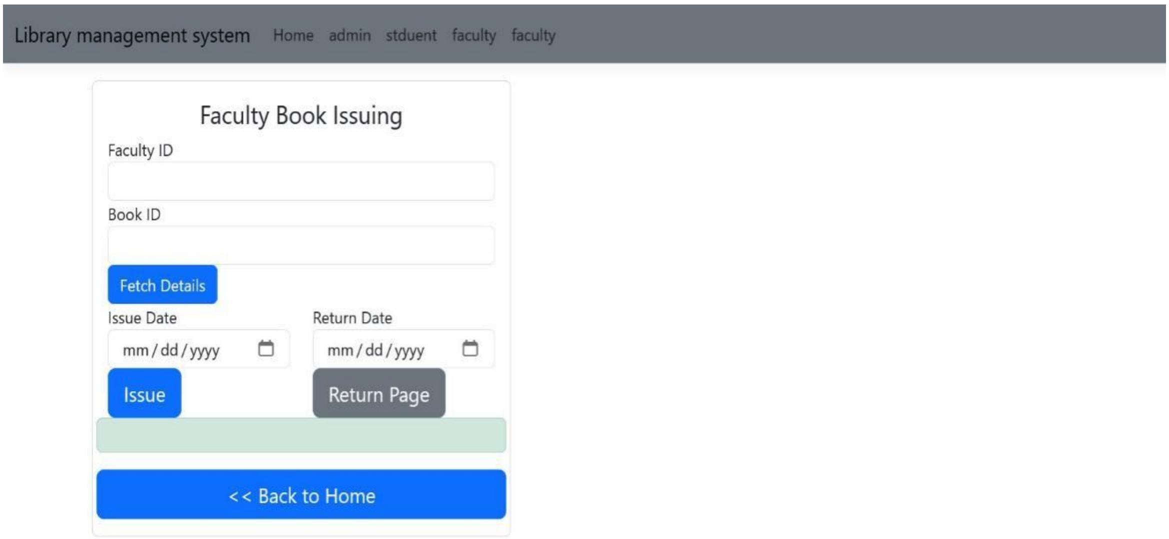


Figure 16: Faculty issue

- ➢ Above screenshot describes about the faculty issue for the faculty issue of the book and it is stored in the faculty transactions.

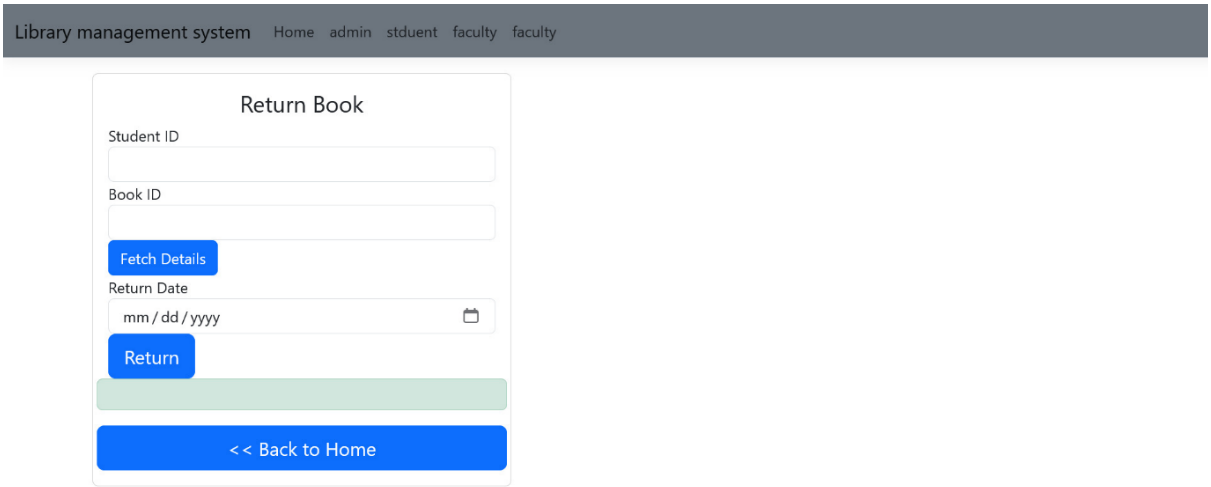


Figure 17: Faculty return

- ➢ Above screenshot describes the return of the book which ever issued to the faculty.

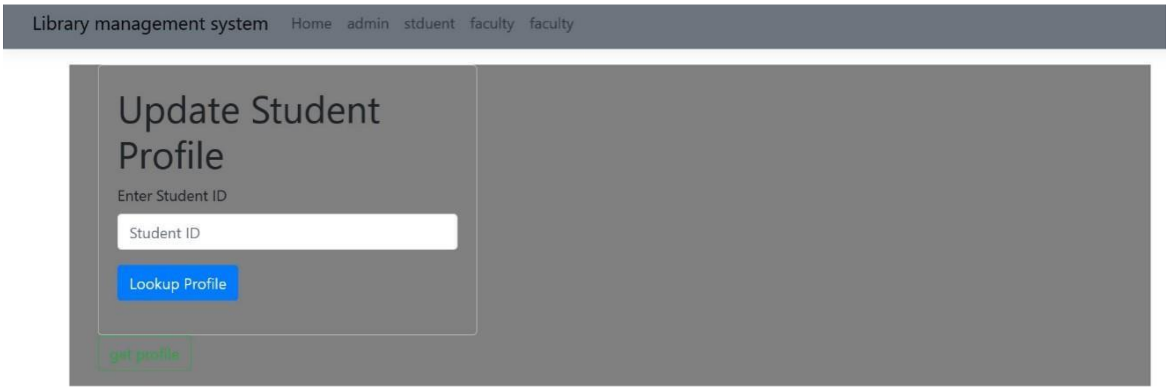
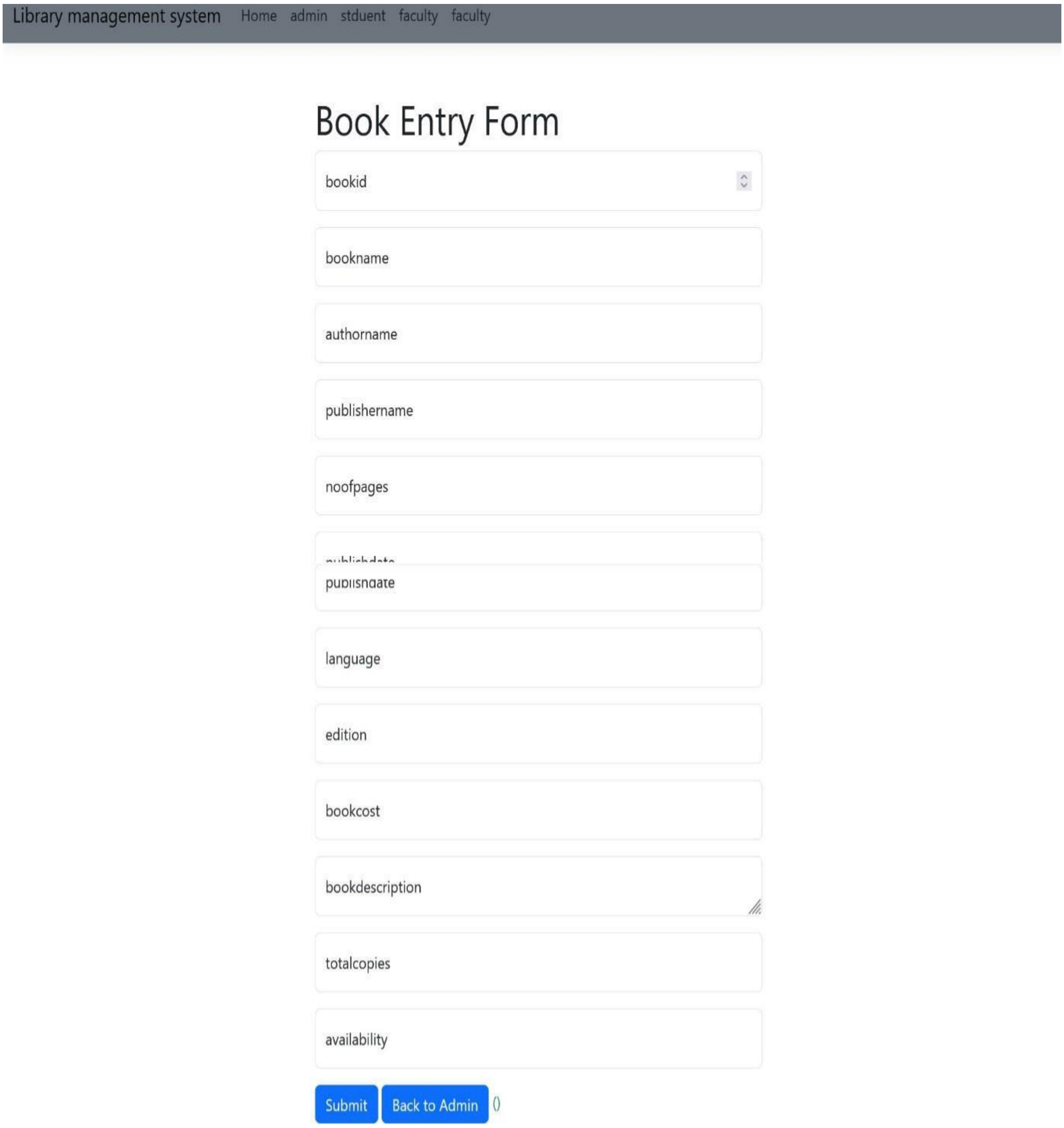


Figure 18: View student profile

- ➢ This screenshot shows the update of the student profile, user need to enter the student Id, that profile will be retrieved from the database and shown to the user.
- ➢ It shows the student data for each student.



Book Entry Form

bookid

bookname

authorname

publishername

noofpages

publishdate

language

edition

bookcost

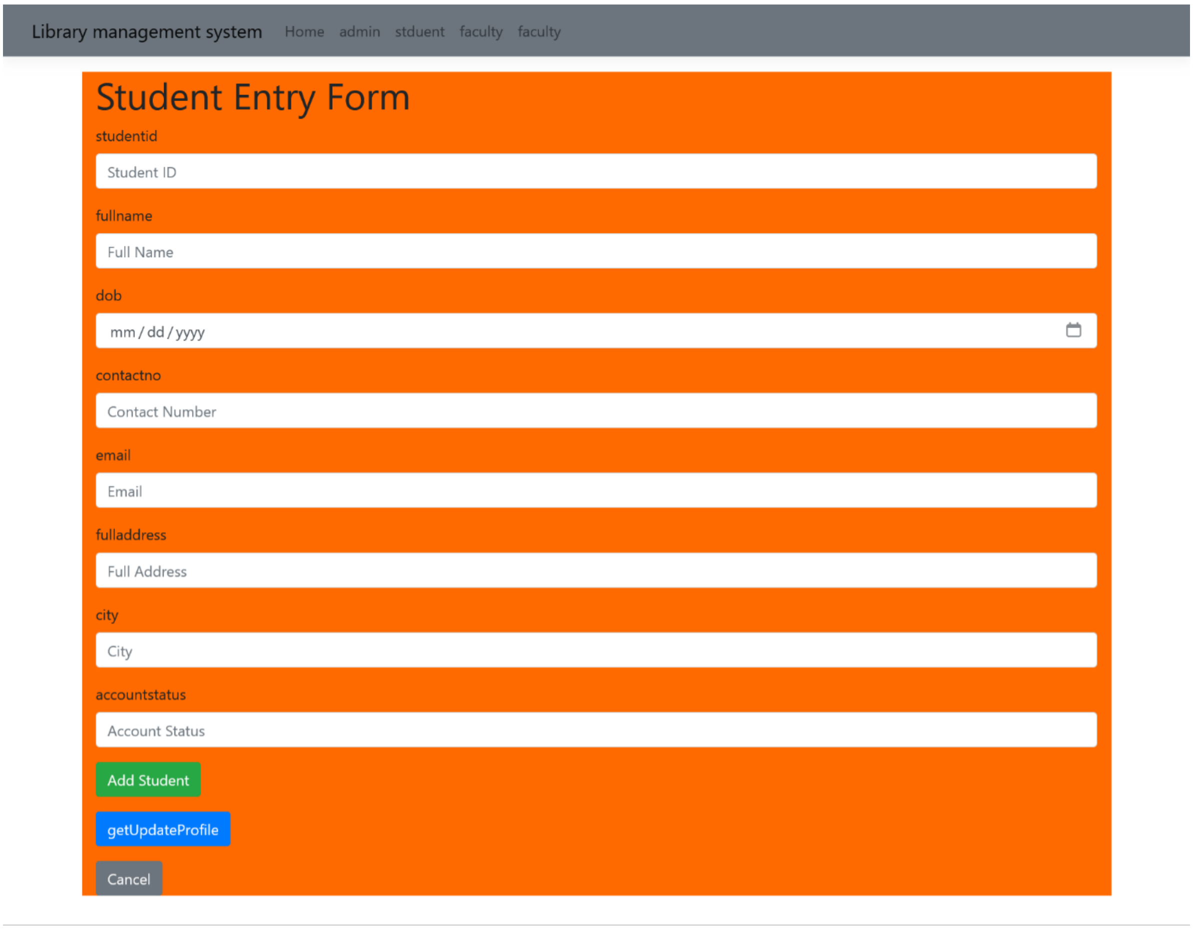
bookdescription

totalcopies

availability

Figure 19: Book entry

- This screenshot describes the entry of the book details which is stored in the books table of the database.



Library management system Home admin stduent faculty faculty

Student Entry Form

studentid

Student ID

fullname

Full Name

dob

mm / dd / yyyy

contactno

Contact Number

email

Email

fulladdress

Full Address

city

City

accountstatus

Account Status

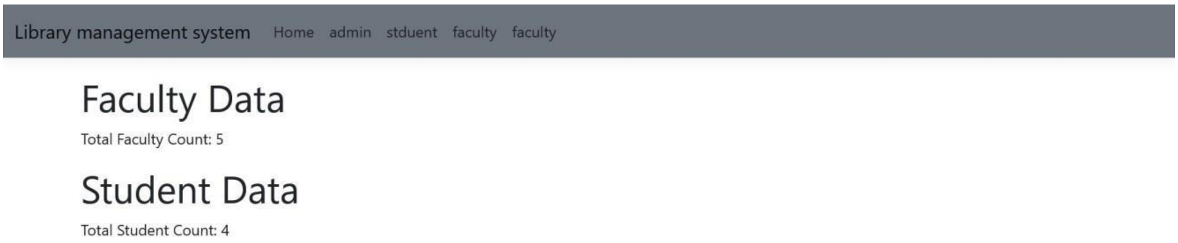
Add Student

getUpdateProfile

Cancel

Figure 20: Student profile

- ➢ This screenshot describes the adding of the student in the database of the student table.



Library management system Home admin stduent faculty faculty

Faculty Data

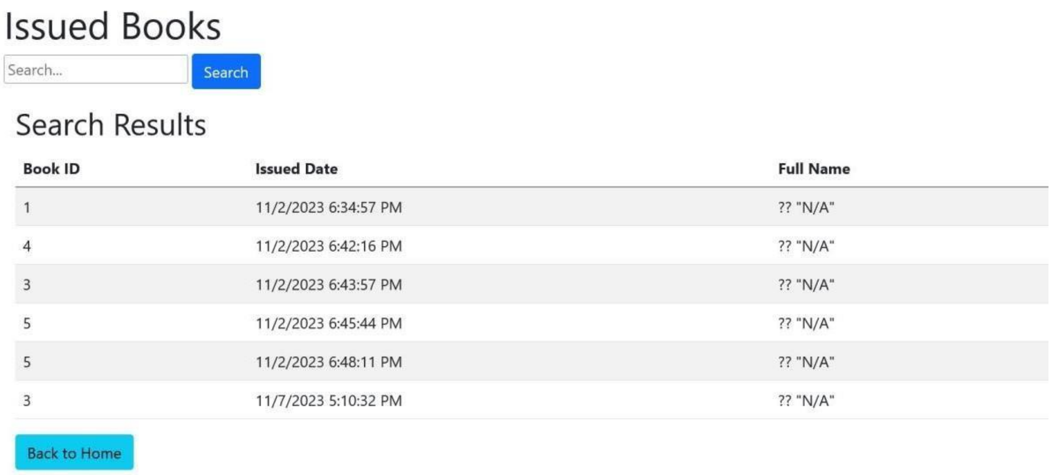
Total Faculty Count: 5

Student Data

Total Student Count: 4

Figure 21: Inventory analysis view data

- ➢ This screenshot describes about showing of the data of a student and faculty in the inventory analysis.



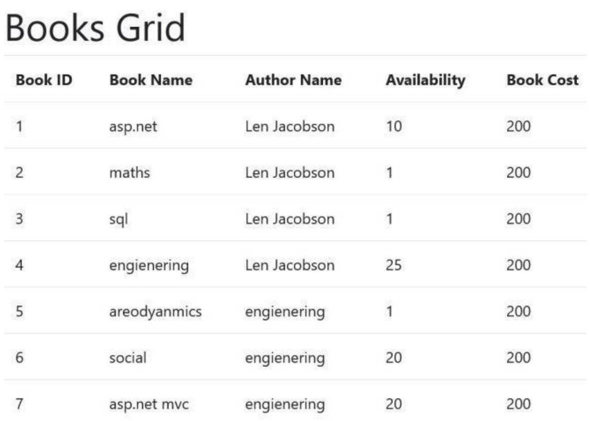
Issued Books

Search Results

Book ID	Issued Date	Full Name
1	11/2/2023 6:34:57 PM	?? "N/A"
4	11/2/2023 6:42:16 PM	?? "N/A"
3	11/2/2023 6:43:57 PM	?? "N/A"
5	11/2/2023 6:45:44 PM	?? "N/A"
5	11/2/2023 6:48:11 PM	?? "N/A"
3	11/7/2023 5:10:32 PM	?? "N/A"

Figure 22: Issued book in inventory analysis

- This screenshot describes the book issue details in the inventory analysis.

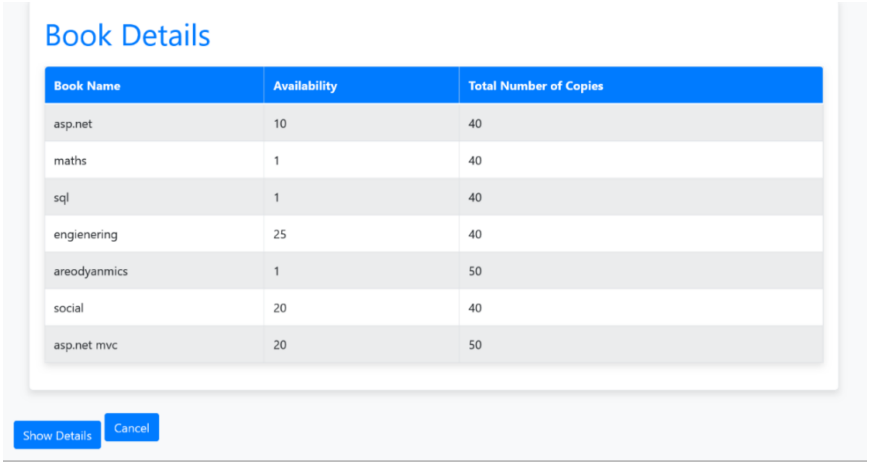


Books Grid

Book ID	Book Name	Author Name	Availability	Book Cost
1	asp.net	Len Jacobson	10	200
2	maths	Len Jacobson	1	200
3	sql	Len Jacobson	1	200
4	engienering	Len Jacobson	25	200
5	areodyanmics	engienering	1	200
6	social	engienering	20	200
7	asp.net mvc	engienering	20	200

Figure 23: Book details

- Above screenshot describes about the book grid view and total no of books, book cost for it and in the inventory analysis.

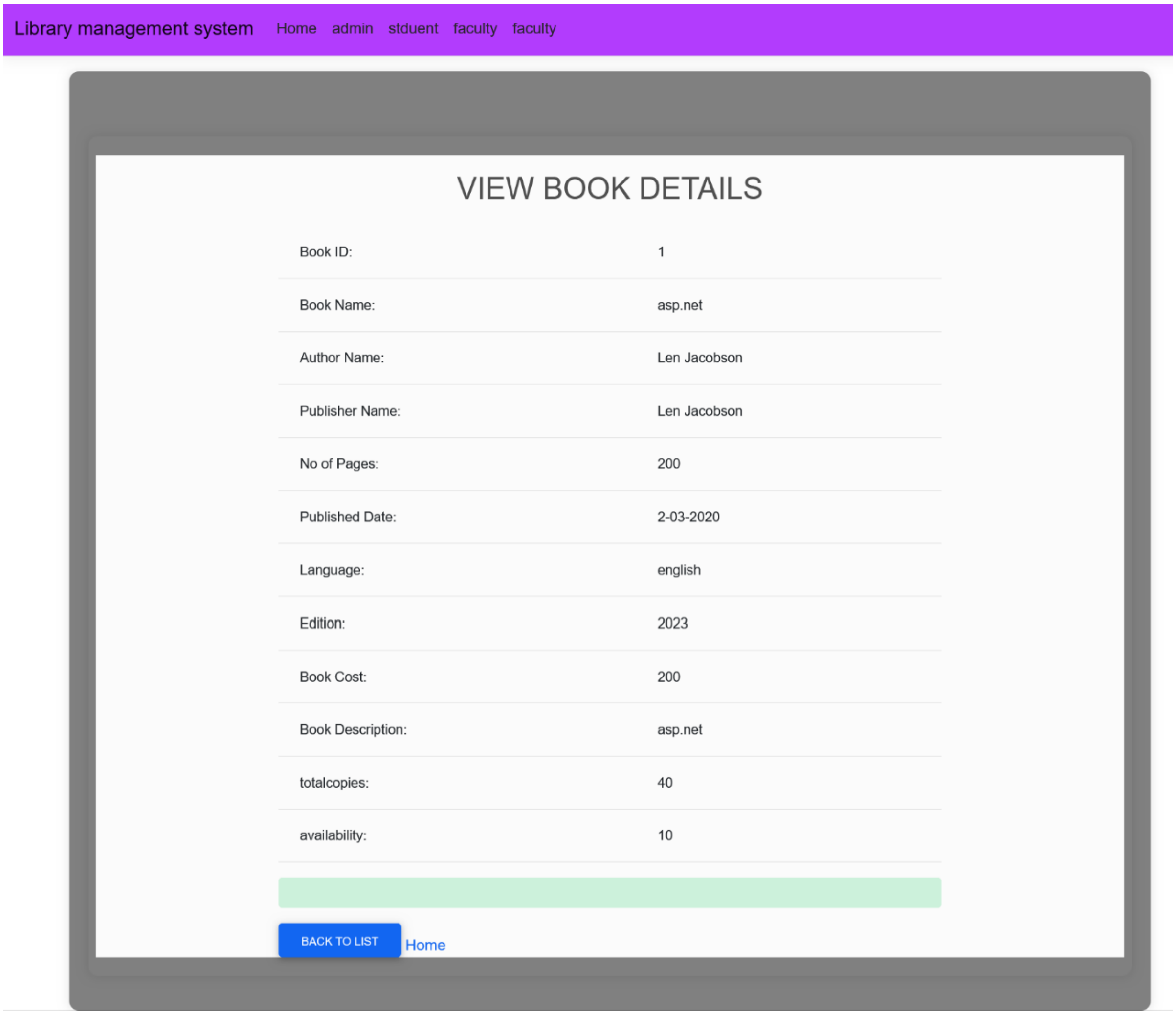


Book Details

Book Name	Availability	Total Number of Copies
asp.net	10	40
maths	1	40
sql	1	40
engienering	25	40
areodyanmics	1	50
social	20	40
asp.net mvc	20	50

Figure 24: Book details with the copies

- Above screenshot describes about the book details related to number of copies and availability of the books.



VIEW BOOK DETAILS

Book ID:	1
Book Name:	asp.net
Author Name:	Len Jacobson
Publisher Name:	Len Jacobson
No of Pages:	200
Published Date:	2-03-2020
Language:	english
Edition:	2023
Book Cost:	200
Book Description:	asp.net
totalcopies:	40
availability:	10

Figure 25: View details of books

- Above screenshot allows the user to view details of each book.

5. *Project Internal/external Interface Impacts and Specification*

Internal Interface

- Admin Console: The interface for all-inclusive application management.
- Librarian Console: A user interface that allows librarians to control user interactions and inventories.
- User Catalogue Interface: An easy-to-use interface that allows people to peruse the books that are offered.

External Interface:

- Technologies for Augmented and Virtual Reality: Interfaces that offer immersive experiences for prospective upgrades in the future.
- Platforms for Mobile Applications: Mobile interface that enables users to engage with the library while on the go.

- Barcoding and RFID Technology: Interfaces that make the process of returning and checking out books easier.

6. *Design Units Impacts*

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

Admin Module:

Authors and publishers may monitor author, publisher, book, and branch data by logging in to administer the system. They manage logouts, view librarian statistics, add conferences and journals, and manage member information.

Librarians module:

To manage book issuance and returns, handle logouts, and administrate books, librarians must log in.

User Module:

To view/update profiles, reserve/return books, reserve/return conferences, and explore available books and journals, users must first register or log in. With this modular strategy, optimized functionality customized to individual user roles is ensured.

6.1.2 Impacts

The streamlined administration, improved operations for librarians, and optimized user interactions are all achieved by this library management system [5]. It raises the total efficacy and functionality of the library by guaranteeing proper information administration, enhancing security protocols, and enabling customized user experiences.

6.1.3 Requirements

Admin Module:

- Login
- Add/update/delete author
- Add/update/delete publisher
- Add/update/delete book
- Add/update/delete branch
- View/delete Member
- Add Journal/conference
- View Librarian Info
- Logout

Librarian Module:

- Login
- Add/update/delete book
- Book issue
- Book Return
- Logout

User module:

- Signup
- Login
- View all books
- View conferences/Journal
- Reserve/Return Books
- View/update profile
- Logout

7. *Open issues*

All of the application's requirements have been met, and the project is now error-free. We had some technical difficulties during implementation, but we were able to resolve them by speaking with our supervisor.

8. *Acknowledgements*

We appreciate our supervisor's ongoing assistance and weekly project review. We extend our gratitude to everyone who contributed, both directly and indirectly, to the timely implementation of this application.

9. *References*

[1]. ADEBESIN, T., & Folorunsho, O. (2015). *Library Management System* (Doctoral dissertation, FEDERAL UNIVERSITY OYE EKITI).

[2]. Abdulrazaq, M. B., & Mustafa, O. M. (2017). Designing and Implementing of An Online Library Management System. *Science Journal of University of Zakho*, *5*(3), 278-284.

[3]. Brown, J. F., Nelson, J. L., & Wineburgh-Freed, M. (2013). Customized electronic resources management system for a multi-library university: Viewpoint from one library. In *Electronic Journal Management Systems* (pp. 89-102). Routledge.

[4]. Gasik, S. (2011). A model of project knowledge management. *Project Management Journal*, *42*(3), 23-44.

[5]. Lowry, C. B., & Troll, D. A. (2019). Carnegie Mellon University and University Microfilms International "Virtual Library Project". In *Serials to the Tenth Power* (pp. 143-169). Routledge.

[6]. Nalluri, S. R., & Gaddam, B. (2016). Mobile library services and technologies: A study. *International Journal of Research in Library Science*, *2*(2), 59-66.

[7] Hjørland, B. (2000). Library and information science: practice, theory, and philosophical basis. *Information processing & management*, *36*(3), 501-531.

[8] Dragon, A. C., & Leisner, T. (2021). The ABCs of implementing library marketing. In *Marketing and the Library* (pp. 33-47). Routledge.

[9] Bénaud, C. L., & Bordeianu, S. (1998). Outsourcing library operations in academic libraries: an overview of issues and outcomes.

10. ***Appendices***

Code:

Library management system

Login page code:

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@model WebApplication6.Models.login
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>@ViewBag.Title</title>
    <!-- Add your other CSS and JavaScript imports here -->

    <style>
        body {
            background-color: lightpink;
        }

        /* Additional styles for enhanced design */
        .card-body {
            border-radius: 10px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
            background-color: #fff;
        }

        .fa-cubes {
            color: #ff6219;
        }

        .btn-primary {
            background-color: #007bff;
            border-color: #007bff;
            transition: background-color 0.2s ease, border-color 0.2s ease;
        }

            .btn-primary:hover {
                background-color: #0056b3;
                border-color: #0056b3;
            }

        /* Add your existing styles here */
    </style>
        <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
rel="stylesheet" />
    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
rel="stylesheet" />
    <!-- MDB -->
    <link  href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"  rel="stylesheet"  />
    <link  href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"  rel="stylesheet"  />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
</head>
<body>
    <div class="col-md-6 col-lg-7 d-flex align-items-center">
        <div class="card-body p-4 p-lg-5 text-black">
            <div class="d-flex align-items-center mb-3 pb-1">
                <i class="fas fa-cubes fa-2x me-3"></i>
            </div>
            <h5 class="fw-normal mb-3 pb-3" style="letter-spacing: 1px;">Sign into your account</h5>
            @using (Html.BeginForm("logins", "Home", FormMethod.Post, new { @id = "form", @class =
"needs-validation", novalidate = "novalidate" }))
            {
                @Html.AntiForgeryToken()

                <div class="form-floating mb-3">
                    @Html.LabelFor(model => model.username, new { @class = "form-label" })
                    @Html.TextBoxFor(model => model.username, new { @class = "form-control form-
control-lg", @required = "required" })
                    <div class="invalid-feedback">
                        Please enter your username.
                    </div>
                </div>

                <div class="form-floating mb-3">
                    @Html.LabelFor(model => model.password, new { @class = "form-label" })
                    @Html.PasswordFor(model => model.password, new { @class = "form-control form-
control-lg", @required = "required" })
                    <div class="invalid-feedback">
                        Please enter your password.
                    </div>
                </div>

                <button type="submit" class="btn btn-primary" value="submit">Login</button>
                <div class="form-check mb-4">
                    <input class="form-check-input" type="checkbox" id="rememberMe" />
                    <label class="form-check-label" for="rememberMe">Remember me</label>
                </div>
            }
            <a class="small text-muted" href="/Home/forgotpassword">Forgot password?</a>
            <p class="mb-5 pb-lg-2" style="color: #393f81;">
                Don't have an account? <a href="register" style="color: #393f81;">Register here</a>
            </p>
            <a href="#!" class="small text-muted">Terms of use.</a>
            <a href="#!" class="small text-muted">Privacy policy</a>
            <a href="index" class="mt-3 btn btn-secondary btn-block">Back to Home</a>
            <div id="Message" class="mt-3"></div>
        </div>
    </div>
    <script>
        document.addEventListener('DOMContentLoaded', function () {
            document.getElementById('form').addEventListener('submit', function (event) {
                var password = document.getElementById('password').value;
                var confirmPassword = document.getElementById('confirmPassword').value;

                if (password !== confirmPassword) {
                    document.getElementById('Message').innerHTML = '<div class="alert alert-
danger">Passwords do not match.</div>';
                    event.preventDefault();
                }
            });
        });
    </script>
</body>
</html>
```

Regisrter page code:

```
@model WebApplication6.Models.registercs
@{
    ViewBag.Title = "Signup";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!doctype html>
<html>
<head>
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
rel="stylesheet" />
    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
rel="stylesheet" />
    <!-- MDB -->
    <link  href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"  rel="stylesheet"  />
</head>

<body>
    <div class="col-md-6 col-lg-7 d-flex align-items-center">
        <div class="card-body p-4 p-lg-5 text-black">
            <div class="d-flex align-items-center mb-3 pb-1">
                <i class="fas fa-cubes fa-2x me-3" style="color: #ff6219;"></i>
            </div>
            <h1 class="mb-4">Registration</h1>
            <div class="form-group">
                @using (Html.BeginForm("Register", "Home", FormMethod.Post, new { @class = "needs-
validation", novalidate = "novalidate" }))
                {
                    @Html.AntiForgeryToken() <!-- This generates the __RequestVerificationToken -->
                    @Html.ValidationSummary(true)
                    <div class="mb-3">
                        <label for="Username" class="form-label">Username:</label>
                        @Html.TextBoxFor(model => model.username, new { @class = "form-control",
required = "required" })
                        @Html.ValidationMessageFor(model => model.username)
                    </div>
                    <div class="mb-3">
                        <label for="Password" class="form-label">Password:</label>
                        @Html.PasswordFor(model => model.password, new { @class = "form-control",
required = "required" })
                        @Html.ValidationMessageFor(model => model.password)
                    </div>
                    <div class="mb-3">
                        <label for="ConfirmPassword" class="form-label">Confirm Password:</label>
                        @Html.PasswordFor(model => model.confirmpassword, new { @class = "form-
control", required = "required" })
                        @Html.ValidationMessageFor(model => model.confirmpassword)
                    </div>

                    <div class="mb-3">
                        <label for="Email" class="form-label">Email:</label>
                        @Html.TextBoxFor(model => model.email, new { @class = "form-control", required =
"required", type = "email", value = "sendemail" })
                        @Html.ValidationMessageFor(model => model.email)
                    </div>
                    <div>
                        <label>Student</label>
                        @Html.RadioButtonFor(m => m.userrole, "student")
                        <label>Admin</label>
                        @Html.RadioButtonFor(m => m.userrole, "admin")
                        <label>Faculty</label>
                        @Html.RadioButtonFor(m => m.userrole, "faculty")
                    </div>
                    <div class="mb-3">
                        <button type="submit" class="btn btn-primary">Register</button>
                    </div>
                    <a href="/Home" class="btn btn-success">Back</a>

                }
                <div>
                    <span style="color:@ViewBag.MessageColor">@ViewBag.Message</span>
                    @ViewBag.Message
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Admin page code:

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
          rel="stylesheet" />
    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
          rel="stylesheet" />
    <!-- MDB -->
    <link    href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"
```

```
          rel="stylesheet" />
    <link  href="https://cdn.jsdelivr.net/npm/mdb@5.1.2/dist/css/mdb.min.css"  rel="stylesheet">
    <script  src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    <link href="style1.css" rel="stylesheet" />
    <style>
        .custom-container {
            background-color: Tomato;
            padding: 10px; /* Add padding to create space between the cards and the container */
        }

        /* Custom card styles */
        .custom-card {
            border: 2px solid #007bff; /* Card border color */
            border-radius: 10px; /* Card border radius */
        }

            .custom-card .card-body {
                text-align: center;
                padding: 20px;
            }

            .custom-card .btn-success {
                background-color: #28a745; /* Button background color */
                color: #fff; /* Button text color */
            }

                .custom-card .btn-success:hover {
                    background-color: #218838; /* Button background color on hover */
                }

        h3 {
            text-align: center;
            padding-left: 200px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="bg-light">
            <nav class="navbar navbar-expand-lg navbar-light bg-light">
                <a class="navbar-brand" href="#">admindashboard</a>
                <!-- Add a profile photo image here -->
@*                  <img src="admin_profile.jpg" alt="Admin Photo" class="rounded-circle" style="width:
40px; height: 40px;" />
 *@                 <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">

                    <span  class="navbar-toggler-icon"></span>
                </button>
                <div class="container custom-containern text-center mt-4">
                    <div class="title">
                        <h3 class="text-primary">Welcome to admin Dashboard</h3>
                    </div>
                </div>

                <!-- Add your navigation links here -->
                <div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav ml-auto">
```

```
                        <div class="dropdown">
                            <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-
toggle="dropdown" aria-expanded="false">
                                profile button
                            </button>
                            <ul class="dropdown-menu">
                                <li><a class="dropdown-item" href="#">profile</a></li>
                                <li><a class="dropdown-item" href="#">settings</a></li>
                                <li><a class="dropdown-item" id="logoutButton" href="/home/index.cshtml">
logout</a></li>
                            </ul>
                    </ul>
                </div>
            </nav>
            <div class="container custom-container">
                <div class="row">
                    <!-- Card 1: Book Entry -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">Book Entry</div>
                                <div class="h3">
                                    <asp:Label ID="Label4" runat="server" Text="Book Entry"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/addbook">
                                    <span class="mr-2">Add Book</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>

                    <!-- Card 2: View Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">View Book</div>
                                <div class="h3">
                                    <asp:Label ID="Label3" runat="server" Text="View Book"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/viewbook">
                                    <span class="mr-2">View Book</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>

                    <!-- Card 3: Search Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">Search Book</div>
                                <div class="h3">
                                    <asp:Label ID="Label2" runat="server" Text="Search Book"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/search">
                                    <span class="mr-2">Search Book</span>
                                    <i class="fas fa-arrow-right"></i>
                                </a>
                            </div>
                        </div>
                    </div>

                    <!-- Card 4: Issue Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">Issue Book</div>
                                <div class="h3">
                                    <asp:Label ID="Label1" runat="server" Text="Issue Book"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/bookissue">
                                    <span class="mr-2">Issue Book</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>
                    <!-- Card 4: Issue Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">faculty Issue Book</div>
                                <div class="h3">
                                    <asp:Label ID="Label1" runat="server" Text="Issue Book"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/facultyt">
                                    <span class="mr-2">Issue Book</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>
                    <!-- Additional Card: Entry Student -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">Entry Student</div>
                                <div class="h3">
                                    <asp:Label ID="Label5" runat="server" Text="Entry Student"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/stprofile">
                                    <span class="mr-2">Entry Student</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>
                </div>
                <!-- Card 2: View Book -->
                <div class="col-lg-4 mb-4">
                    <div class="card custom-card">
                        <div class="card-body">
                            <div class="small text-muted">invertory analysis</div>
                            <div class="h3">
                                <asp:Label ID="Label3" runat="server" Text="invertory analysis"></asp:Label>
                            </div>
                            <a class="btn btn-success mt-3" href="/Home/invertoryan">
                                <span class="mr-2">invertory analaysis</span>
                            <i class="fas fa-arrow-right"></i>
                          </a>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
                <div>
                  <footer class="bg-dark text-center text-white">
                    <!-- Grid container -->
                     <div class="container p-4 pb-0">
                       <!-- Section: Social media -->
                      <section class="mb-4">
                        <!-- Facebook -->
                        <a class="btn text-white btn-floating m-1"
                          style="background-color: #3b5998;"
                          href="#!"
                          role="button">
                           <i class="fab fa-facebook-f"></i>
                        </a>
                        <!-- Google -->
                        <a class="btn text-white btn-floating m-1"
                          style="background-color: #dd4b39;"
                          href="#!"
                          role="button">
                           <i class="fab fa-google"></i>
                        </a>

                        <!-- Instagram -->
                        <a class="btn text-white btn-floating m-1"
                          style="background-color: #ac2bac;"
                          href="#!"
                          role="button">
                           <i class="fab fa-instagram"></i>
                        </a>

                        <!-- Linkedin -->
                        <a class="btn text-white btn-floating m-1"
                          style="background-color: #0082ca;"
                          href="#!"
                          role="button">
                           <i class="fab fa-linkedin-in"></i>
                        </a>
                        <!-- Github -->
                        <a class="btn text-white btn-floating m-1"
                          style="background-color: #333333;"
                          href="#!"
                          role="button">
                           <i class="fab fa-github"></i>
                         </a>
                      </section>
                      <!-- Section: Social media -->
                    </div>
                    <!-- Grid container -->
                    <!-- Copyright -->
                    <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
                      © 2020 Copyright:
                      <a class="text-white" href="https://mdbootstrap.com/">MDBootstrap.com</a>
                    </div>
                    <!-- Copyright -->
```

```
          </footer>
        </div>
    </form>

    <script>
        document.getElementById("logoutButton").addEventListener("click", function () {
            if (confirm("Are you sure you want to log out?")) {
                // If the user confirms the logout, redirect to the logout action
                window.location.href = '@Url.Action("Logout", "Home")'; // Replace with your actual URL
            }
        });
    </script>
    <div>
        <a href="/Home/index"><< Back to Home</a><br>
    </div>
</body>
</html>
```

Faculty page:

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
rel="stylesheet" />

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
rel="stylesheet" />

    <!-- MDB -->
    <link  href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"   rel="stylesheet"   />
    <link  href="https://cdn.jsdelivr.net/npm/mdb@5.1.2/dist/css/mdb.min.css"   rel="stylesheet">

    <!-- Custom Styles -->
    <style>

        body {
            background-color:orange; /* Set the background color for the entire body */
        }

        .nav-pills .nav-link {
            color: #007bff; /* Set blue color for nav-links */
        }

            .nav-pills .nav-link:hover {
                background-color: #007bff;
                color: #ffffff; /* Set white text color on hover */
            }
        .jumbotron {
            background-color: #007bff; /* Set a blue background color for the jumbotron */
            color: #ffffff; /* Set white text color */
            margin-bottom: 0;
            padding: 20px;
        }

        .footer {
            background-color: #343a40; /* Set a dark background color for the footer */
            color: #ffffff; /* Set white text color for the footer */
            padding: 20px;
        }

        .btn-logout {
            background-color: #dc3545; /* Set a red background color for the logout button */
            color: #ffffff; /* Set white text color for the logout button */
        }

        /* Add more custom styles as needed */
    </style>
</head>

<body>
    <div class="jumbotron text-center alert alert-primary">
        <h1>Library Management System</h1>
        <p>Building community. Inspiring readers. Expanding book access!</p>
    </div>
    <div class="container-fluid">
        <div class="row">
              <div class="col-sm-2 border border-info">
                  <ul class="nav nav-pills flex-column">
                    <li class="nav-item">
                        <a class="nav-link active" href="/home/index">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="/Home/facultyprofile">Faculty Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="/Home/facultyts">Faculty Transactions</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="/Home/facultyt">Faculty Issue</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="/Home/fsearch">Faculty Search</a>
                    </li>
                @*     <li class="nav-item">
                        <button id="logoutButton" class="btn btn-outline-secondary btn-logout">Logout</button>
                    </li> *@
                    <div class="dropdown">
                        <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">
                            Dropdown button
                        </button>
                        <ul class="dropdown-menu">
                            <li><a class="dropdown-item" href="#">profile</a></li>
                            <li><a class="dropdown-item" href="#">settings</a></li>
                            <li><a class="dropdown-item" id="logoutButton"href="#"> logout</a></li>
                        </ul>
                </ul>
                <hr class="d-sm-none">
            </div>
            <div class="col-sm-10 border border-info">
                <div>
                    <h1 style="text-align:center">Welcome to Faculty</h1>
                </div>

            </div>
        </div>
    </div>
    <script>
        document.getElementById("logoutButton").addEventListener("click", function () {
            if (confirm("Are you sure you want to log out?")) {
                // If the user confirms the logout, redirect to the logout action
                window.location.href = '@Url.Action("Logout", "Home")'; // Replace with your actual URL
            }
        });
    </script>
    <br />
    <div>
        <footer class="bg-dark text-center text-white">
            <div class="container p-4 pb-0">
                <section class="mb-4">
                    <a class="btn text-white btn-floating m-1" style="background-color: #3b5998;" href="#!"
                       role="button">
                        <i class="fab fa-facebook-f"></i>
                    </a>
                    <a class="btn text-white btn-floating m-1" style="background-color: #dd4b39;" href="#!"
                       role="button">
                        <i class="fab fa-google"></i>
                    </a>
                    <a class="btn text-white btn-floating m-1" style="background-color: #ac2bac;" href="#!"
                       role="button">
                        <i class="fab fa-instagram"></i>
                    </a>
                    <a class="btn text-white btn-floating m-1" style="background-color: #0082ca;" href="#!"
                       role="button">
                        <i class="fab fa-linkedin-in"></i>
                    </a>
                    <a class="btn text-white btn-floating m-1" style="background-color: #333333;" href="#!"
                       role="button">
                        <i class="fab fa-github"></i>
                    </a>
                </section>
            </div>
            <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
                © 2020 Copyright:
                <a class="text-white" href="https://mdbootstrap.com/">MDBootstrap.com</a>
            </div>
        </footer>
    </div>
</body>

</html>
```

Invertory Page:

@*

```
    For more information on enabling MVC for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860
```

```
*@
@{
}
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
          rel="stylesheet" />
    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
          rel="stylesheet" />
    <!-- MDB -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.4.1/mdb.min.css"
          rel="stylesheet" />
    <link   href="https://cdn.jsdelivr.net/npm/mdb@5.1.2/dist/css/mdb.min.css"   rel="stylesheet">
    <script   src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    <link href="style1.css" rel="stylesheet" />
    <style>
        .custom-container {
            background-color:aliceblue;
            padding: 10px; /* Add padding to create space between the cards and the container */
        }

        /* Custom card styles */
        .custom-card {
            border: 2px solid #007bff; /* Card border color */
            border-radius: 10px; /* Card border radius */
        }

            .custom-card .card-body {
                text-align: center;
                padding: 20px;
            }

            .custom-card .btn-success {
                background-color: #28a745; /* Button background color */
                color: #fff; /* Button text color */
            }

                .custom-card .btn-success:hover {
                    background-color: #218838; /* Button background color on hover */
                }

        h3 {
            text-align: center;
            padding-left: 200px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="bg-light">
            <nav class="navbar navbar-expand-lg navbar-light bg-gradient">
```

```
                <a class="navbar-brand" href="#">Inventoryanalysis</a>
                <!-- Add a profile photo image here -->
@*                  <img src="admin_profile.jpg" alt="Photo" class="rounded-circle" style="width: 40px;
height: 40px;" />
 *@                 <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">

                    <span  class="navbar-toggler-icon"></span>
                </button>
                <div class="container custom-containern text-center mt-4">
                    <div class="title">
                        <h3 class="text-primary"> Inventory analysis</h3>
                    </div>
                </div>

                <!-- Add your navigation links here -->
                <div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav ml-auto">
                        <li class="nav-item">
                            <!-- Bootstrap Sign Out Button -->
                            <a type="button" class="btn btn-light" id="logoutButton">Sign Out</a>
                        </li>
                    </ul>
                </div>
            </nav>
            <div class="container custom-container">
                <div class="row">
                    <!-- Card 1: Book Entry -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">show issued books</div>
                                <div class="h3">
                                    <asp:Label ID="Label4" runat="server" Text="show book issue"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/showissuedbook">
                                    <span class="mr-2">bookissued</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>

                    <!-- Card 2: View Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">totalbooks</div>
                                <div class="h3">
                                    <asp:Label ID="Label3" runat="server" Text="showbookdetails"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/bookdetails">
                                    <span class="mr-2">bookdetails</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>
                    <!-- Card 3: Search Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted">bookdetails</div>
                                <div class="h3">
                                    <asp:Label ID="Label2" runat="server" Text="show books"></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/getbook">
                                    <span class="mr-2">bookdetials</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>

                    <!-- Card 4: Issue Book -->
                    <div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted"></div>
                                <div class="h3">
                                    <asp:Label ID="Label1" runat="server"
Text="showdata">showdata</asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="/Home/showdata">
                                    <span class="mr-2">showdata</span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>
                        </div>
                    </div>
                    <!-- Card 4: Issue Book -->
                    <!--<div class="col-lg-4 mb-4">
                        <div class="card custom-card">
                            <div class="card-body">
                                <div class="small text-muted"></div>
                                <div class="h3">
                                    <asp:Label ID="Label1" runat="server" Text=""></asp:Label>
                                </div>
                                <a class="btn btn-success mt-3" href="">
                                    <span class="mr-2"></span>
                                    <i class="fas fa-arrow-right"></i>
                                 </a>
                             </div>-->
                </div>
            </div>
        </div>
        </div>
        </div>
        <div>
            <footer class="bg-dark text-center text-white">
                <!-- Grid container -->
                  <div class="container p-4 pb-0">
                      <!-- Section: Social media -->
                     <section class="mb-4">
                        <!-- Facebook -->
                        <a class="btn text-white btn-floating m-1"
                           style="background-color: #3b5998;"
                           href="#!"
```

```
              role="button">
               <i class="fab fa-facebook-f"></i>
            </a>
            <!-- Google -->
            <a class="btn text-white btn-floating m-1"
              style="background-color: #dd4b39;"
              href="#!"
              role="button">
               <i class="fab fa-google"></i>
            </a>

            <!-- Instagram -->
            <a class="btn text-white btn-floating m-1"
              style="background-color: #ac2bac;"
              href="#!"
              role="button">
               <i class="fab fa-instagram"></i>
            </a>

            <!-- Linkedin -->
            <a class="btn text-white btn-floating m-1"
              style="background-color: #0082ca;"
              href="#!"
              role="button">
               <i class="fab fa-linkedin-in"></i>
            </a>
            <!-- Github -->
            <a class="btn text-white btn-floating m-1"
              style="background-color: #333333;"
              href="#!"
              role="button">
               <i class="fab fa-github"></i>
            </a>
          </section>
          <!-- Section: Social media -->
        </div>
        <!-- Grid container -->
        <!-- Copyright -->
        <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
          © 2020 Copyright:
          <a class="text-white" href="https://mdbootstrap.com/">MDBootstrap.com</a>
        </div>
        <!-- Copyright -->
      </footer>
    </div>
  </form>
  <script>
    document.getElementById("logoutButton").addEventListener("click", function () {
      if (confirm("Are you sure you want to log out?")) {
        // If the user confirms the logout, redirect to the logout action
        window.location.href = '@Url.Action("Logout", "Home")'; // Replace with your actual URL
       }
    });
  </script>
  <div>
    <a href="/Home/index"><< Back to Home</a><br>
  </div>
</body>
</html>
```

Student page:

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Profile</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
/>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/mdb@5.1.2/dist/css/mdb.min.css" />
</head>
<body>
    <div class="container" style="background-color:aliceblue;">
        <div class="form-container">
            <h1>Student Entry Form</h1>

            <form method="post" asp-action="studentp">
                @Html.AntiForgeryToken()

                <form method="post" asp-action="studentp">
                    @Html.AntiForgeryToken()
                    <div class="form-group">
                        <label for="studentId">Enter Student ID:</label>
                        <input type="text" class="form-control" name="studentId" />
                    </div>
                    <button type="submit" class="btn btn-primary">View Profile</button>
                </form>
                <div class="container">
                    @if (Model != null)
                    {
                        <h1>Student Profile</h1>
                        <table class="table table-bordered">
                            <tr>
                                <th>Student ID</th>
                                <td>@Model.studentid</td>
                            </tr>
                            <tr>
                                <th>Full Name</th>
                                <td>@Model.fullname</td>
                            </tr>
                            <tr>
                                <th>Date of Birth</th>
                                <td>@Model.dob</td>
                            </tr>
                            <tr>
                                <th>Contact Number</th>
                                <td>@Model.contactno</td>
                            </tr>
                            <tr>
                                <th>Email</th>
                                <td>@Model.email</td>
                            </tr>
                            <tr>
                                <th>Full Address</th>
                                <td>@Model.fulladdress</td>
                            </tr>
                            <tr>
                                <th>City</th>
                                <td>@Model.city</td>
                            </tr>
                            <tr>
                                <th>Account Status</th>
                                <td>@Model.accountstatus</td>
                            </tr>
                        </table>
                    }
                    </div>
                    </form>
                </body>
                </html>
```

Add book page:

```
@*
    For more information on enabling MVC for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
}
@model WebApplication6.Models.bookentrym
@{
    ViewBag.message = "";
}
<!DOCTYPE html>
<html>
<head>
    <title>Book Entry Form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
/>
</head>
<body>
    <div class="container mt-5">
        <div class="row">
            <div class="col-md-6 offset-md-3">
                <h1>Book Entry Form</h1>
                <form method="post" asp-action="addbook" asp-controller="Home"
enctype="multipart/form-data">
                    <input type="hidden" />

                    <!---bookid--->
                    <div class="mb-3 form-floating">
                        <input asp-for="bookid" class="form-control" placeholder="" />
                        <label asp-for="bookid" for="txtbookid"></label>
                        <span asp-validation-for="bookid" class="text-danger"></span>
                    </div>

                    <!-- Book Name -->
                    <div class="mb-3 form-floating">
                        <input asp-for="bookname" class="form-control" placeholder="" />
                        <label asp-for="bookname" for="txtbookname"></label>
                        <span asp-validation-for="bookname" class="text-danger"></span>
                    </div>

                    <!-- Author Name -->
                    <div class="mb-3 form-floating">
                        <input asp-for="authorname" class="form-control" placeholder="" />
                        <label asp-for="authorname" for="txtauthorn"></label>
                        <span asp-validation-for="authorname" class="text-danger"></span>
                    </div>

                    <!-- Publisher Name -->
                    <div class="mb-3 form-floating">
                        <input asp-for="publishername" class="form-control" placeholder="" />
                        <label asp-for="publishername" for="txtpublishern"></label>
                        <span asp-validation-for="publishername" class="text-danger"></span>
                    </div>

                    <!-- Number of Pages -->
                    <div class="mb-3 form-floating">
                        <input asp-for="noofpages" class="form-control" placeholder="" />
                        <label asp-for="noofpages" for="txtpages"></label>
                        <span asp-validation-for="noofpages" class="text-danger"></span>
                    </div>

                    <!-- Published Date -->
                    <div class="mb-3 form-floating">
                        <input asp-for="publishdate" class="form-control" placeholder="" />
                        <label asp-for="publishdate" for="txtDatePicker"></label>
                        <span asp-validation-for="publishdate" class="text-danger"></span>
                    </div>

                    <!-- Language -->
                    <div class="mb-3 form-floating">
                        <input asp-for="language" class="form-control" placeholder="" />
                        <label asp-for="language" for="txtlanguage"></label>
                        <span asp-validation-for="language" class="text-danger"></span>
                    </div>

                    <!-- Edition -->
                    <div class="mb-3 form-floating">
                        <input asp-for="edition" class="form-control" placeholder="" />
                        <label asp-for="edition" for="txtedition"></label>
                        <span asp-validation-for="edition" class="text-danger"></span>
                    </div>

                    <!-- Book Cost -->
                    <div class="mb-3 form-floating">
                        <input asp-for="bookcost" class="form-control" placeholder="" />
                        <label asp-for="bookcost" for="txtbookc"></label>
                        <span asp-validation-for="bookcost" class="text-danger"></span>
                    </div>

                    <!-- Book Description -->
                    <div class="mb-3 form-floating">
                        <textarea asp-for="bookdescription" class="form-control" rows="4"
placeholder=""></textarea>
                        <label asp-for="bookdescription" for="txtdescription"></label>
                        <span asp-validation-for="bookdescription" class="text-danger"></span>
                    </div>

                    <!-- Actual Stock -->
                    <div class="mb-3 form-floating">
                        <input asp-for="totalcopies" class="form-control" placeholder="" />
                        <label asp-for="totalcopies" for="txtactuals"></label>
                        <span asp-validation-for="totalcopies" class="text-danger"></span>
                    </div>

                    <!-- Current Stock -->
                    <div class="mb-3 form-floating">
                        <input asp-for="availability" class="form-control" placeholder="" />
                        <label asp-for="availability" for="txtcurrents"></label>
```

```
                        <span asp-validation-for="availability" class="text-danger"></span>
                    </div>

                    <!-- Submit Button -->
                    <div class="mb-3">
                        <input type="submit" value="Submit" class="btn btn-primary" />
                        <input type="button" value="Back to Admin" class="btn btn-primary" id="backButton"
/>
                    </div>
                </form>
                <span class="text-success">(@ViewBag.Message)</span>
                <script>
                    document.getElementById('backButton').addEventListener('click', function () {
                        window.history.back(); // This will navigate back to the previous page
                    });
                </script>
            </div>
        </div>
    </div>
</body>
</html>
```