A WEB BASED LIBRARY MANAGEMENT SYSTEM

BY
DEBABRATA ROY CHINMOY
ID: 201-15-13942

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering.

Supervised By
Mr. Saiful Islam
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By
Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
JULY 2024

APPROVAL

This Project titled "A Web Based Library Management System", submitted by Debabrata Roy Chinmoy and to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 15/07/2024.

BOARD OF EXAMINERS

Dr. Shek Rashed Haider Noori — **Board Chairman**
Professor & Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mr. Mohammad Jahangir Alam — **Internal Examiner 1**
Senior Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mr. Golam Rabbany — **Internal Examiner 2**
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Arshad Ali — **External Examiner**
Professor
Department of CSE
Hajee Mohammad Danesh Science and Technology University

DECLARATION

I hereby declare that this project has been done by us under the supervision of **Mr. Saiful Islam, Assistant Professor, Department of CSE**, Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Mr. Saiful Islam
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Raja Tariqul Hasan Tuser
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Debabrata Roy

Debabrata Roy Chinmoy
ID: 201-15-13942
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty ALLAH for His divine blessing in making it possible to complete the final year project/internship successfully.

I am really grateful and wish my profound indebtedness to **Mr. Saiful Islam, Assistant Professor, Department of CSE** Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of Web Application" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express our heartiest gratitude to **Dr. Sheak Rashed Haider Noori, Head Department of CSE**, for his kind help to finish this project and also to other faculty members and the staff of the CSE department of Daffodil International University.

I would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

The library management system project represents a technological innovation aimed at optimizing the management of educational resources within academic institutions. Developed using the MERN stack, the system offers a comprehensive solution for students, librarians, and administrative staff. Users can seamlessly issue, renew, and return books, while librarians benefit from real-time updates for student and staff management, as well as efficient book inventory control. The integration of an automated fine system ensures accountability, imposing penalties for overdue book returns. The MERN stack's robustness ensures scalability, security, and optimal performance. The design prioritizes flexibility for future technological advancements and accessibility across devices. Integration with external systems enhances the educational experience, and mobile responsiveness accommodates users on various platforms. Usability testing and user feedback mechanisms contribute to continuous improvement, while cross-browser compatibility and consistent branding elements enhance the system's visual appeal. This project signifies a commitment to creating a modern, user-centric, and technologically advanced library management system

TABLE OF CONTENTS

CONTENTS	PAGE NO
Board of examiners	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
List of Figures	viii
List of Table	x
CHAPTER 1: INTRODUCTION	**1-8**
1.1 Introduction	1
1.2 Motivation	2
1.4 Expected Outcome	4
1.5 Project Management and Finance	5
1.6 Report Layout	7
CHAPTER 2: BACKGROUND	**9-12**
2.1 Preliminary and Terminology	9
2.2 Related Works	9
2.3 Comparative Analysis	10
2.4 Scope of the Problem	11
CHAPTER 3: REQUIREMENT SPECIFICATION	**13-25**
3.1 Business Process Model	13
3.2 Requirement Collection and Analysis	14
3.3 Use Case Modeling and Description	15
3.4 Logical Data Model	24
3.5 Design Requirements	25
CHAPTER 4: DESIGN SPECIFICATION	**26-38**
4.1 Back-end Design	26
4.2 Front-end Design	35
4.3 Interaction Design and User Experience	37
4.4 Implementation Requirements:	37
CHAPTER 5: IMPLEMENTATION AND TESTING	**39-50**
5.1 Implementation of Database	39
5.2 Implementation of Front-end Design	43
5.3 Testing Implementation	45
5.4 Test Result and Report	47
CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	**51-52**
6.1 Impact on Society	51

6.2 Impact on Environment 51
6.3 Ethical Aspects 52
6.4 Sustainability Plan 52
CHAPTER 7: CONCLUSION AND FUTURE SCOPE 53-54
7.1 Discussion and Conclusion: 53
7.2 Scope for Further Development 54
REFERENCES 55

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1.1: Business Processing Model of Library Management System	13
Figure 3.3.1: Use Case Diagram of Library Management System	15
Figure 3.4.1: Logical Data Model	24
Figure 4.1.1: Admin login page	26
Figure 4.1.2: Admin profile page	27
Figure 4.1.3: Book list page	27
Figure 4.1.4: Add new book page	28
Figure 4.1.5: Manage student page	28
Figure 4.1.6: Book issue request Page	29
Figure 4.1.7: Book recommendation Page	29
Figure 4.1.8: All issue book page	30
Figure 4.1.9: Today issue book page	30
Figure 4.1.10: Today return book page	31
Figure 4.1.11: Add employee page	31
Figure 4.1.12: Add service data page	32
Figure 4.1.13: Student or User registration page	32
Figure 4.1.14: Student profile page	33
Figure 4.1.15: All available book in library page	33
Figure 4.1.16: Recommendation of books page	34
Figure 4.1.17: My issue book page	34
Figure 4.2.1: Home page	35

Figure 4.2.2: About page	35
Figure 4.2.3: Catalog	36
Figure 4.2.4: Services page	36
Figure 4.2.5: Contact Page	37
Figure 5.1.1: All database table	39
Figure 5.1.2: Book list table	40
Figure 5.1.3: Book issue record table	40
Figure 5.1.4: Return book record	41
Figure 5.1.5: Service table	41
Figure 5.1.6: Student table	42
Figure 5.1.7: Transaction table	42
Figure 5.1.8: User table	43
Figure 5.2.1: Some code of front-end using react.js framework	43
Figure 5.2.2: Some code of CSS for front-end design	44
Figure 5.2.3: Some code of Dashboard using react.js framework	44
Figure 5.2.4: Some code of All Student using react.js framework	45

LIST OF TABLE

TABLES	PAGE NO
Table 3.3.1: Login Use Case of Library Management System	16
Table 3.3.2: View Catalog Case of Library Management System	17
Table 3.3.3: Request Book Use Case of Library Management System	17
Table 3.3.4: All Books Use Case of Library Management System	18
Table 3.3.5: Send Recommendation Of Books Use Case of Library Management System	18
Table 3.3.6: Currently Issued Books Use Case of Library Management System	19
Table 3.3.7: See Detail with Penalty Report Use Case of Library Management System	19
Table 3.3.8: Return Books Use Case of Library Management System	20
Table 3.3.9: Renew Books Use Case of Library Management System	20
Table 3.3.10: AdminUse Case of Library Management System	21
Table 3.3.11: Add Books Use Case of Library Management System	21
Table 3.3.12: Issue Request Use Case of Library Management System	22
Table 3.3.13: Receive Recommendation of Books Use Case of Library Management System	22
Table 3.3.14: Manage Student Use Case of Library Management System	23
Table 3.3.15: Add Employee Use Case of Library Management System	23

CHAPTER 1

INTRODUCTION

1.1 Introduction

The objective of the library management system project is to offer a streamlined and user-friendly solution for overseeing library operations. This project facilitates an understanding of the practical aspects of working in real-life situations. This study on library administration aims to enhance students' efficiency in adapting to the evolving technological landscape. Gaining knowledge about the operational and functional mechanism of the library is beneficial for me. It provides a positive and enriching experience for the pupils. The system is specifically engineered to optimize a range of functions, including organizing books, maintaining member records, monitoring the circulation of borrowed, returned, and reissued books, and producing reports. The main goal of this project is to mechanize the manual procedures associated with library management, thereby enhancing precision, minimizing paperwork, and improving overall effectiveness.

This project report provides an in-depth overview of the creation and execution of a library management system utilizing the MERN stack. By utilizing the capabilities of the MERN (MongoDB, Express.js, React.js, Node.js) stack, this project combines a strong collection of technologies to develop a scalable, efficient, and user-friendly solution.

This report will include an analysis of the project's objectives, scope, methodology, and the technologies employed. Additionally, we will present a comprehensive analysis of the system's architecture, database structure, and the principal features and functionalities that have been incorporated. In addition, we will provide a detailed overview of the development process, the obstacles encountered, and the tactics utilized to surmount them. In addition, we will analyze the design of the system's user interface, highlighting the significance of user experience and accessibility. We will showcase pictures and provide detailed descriptions of the several screens and menus, emphasizing the user-friendly navigation and the simplicity of operation for both library staff and members. The study will also address the execution of crucial modules, including book management, member management, circulation management, and reporting. We

will elucidate the fundamental algorithms and reasoning employed in each module and deliberate on their incorporation into the comprehensive system. In addition, we will discuss the security procedures in place to safeguard sensitive data, such as authentication and authorization mechanisms. Additionally, we will address the system's capacity to handle increased demands and its potential for future improvements and extensions.

1.2 Motivation

The development of a library management system is not just about building software, but also a solution that streamlines the operations of a critical institution within academic settings. This project allows you to bridge the gap between theoretical knowledge and practical application, showcasing your skills in software development, database management, and system design. The system has the potential to simplify complex processes, reduce administrative burdens, and create a more efficient and organized library environment. The convenience the system brings to students, such as easy book issuance, renewal, and return, and to librarians, who benefit from real-time updates and automated processes. By incorporating features like automatic fines for overdue books, the system contributes to the establishment of fair and efficient policies within educational institutions. A well-designed library management system can positively impact educational institutions, contributing to a better learning experience for students and leaving a lasting mark on the academic community. Throughout the development process, focus on user-centric design, ensuring that the technology is embraced by both librarians and students, fostering a positive and effective user experience. The motivation for implementing a MERN stack Library Management System lies in creating an advanced, efficient, and user-friendly solution to modernize library operations. The stack aims to enhance scalability, real-time communication, and accessibility, facilitating rapid development, cross-platform compatibility, and seamless integration of front-end and back-end components. The objective is to improve efficiency, user experience, and data management while benefiting from the active developer community and ensuring cost-effective, secure, and scalable library services.

1.3 Objective

The objectives of your library management system project can be outlined to achieve specific goals and functionalities. Here are the objectives for our project:

- Automated Book Transactions: Enable students to easily issue, renew, and return books through a user-friendly interface, reducing manual paperwork and enhancing user experience.
- Real-Time Updates: Implement a system that provides real-time updates for librarians, ensuring that the library database is always current and accurate.
- Student Management: Create a comprehensive student management system to track borrower information, borrowing history, and contact details for effective communication.
- Book Management: Facilitate efficient book management, allowing librarians to add new books to the system, delete outdated or damaged books, and update book information as needed.
- Staff Management: Develop a staff management module that allows librarians to add new staff members, update their information, and manage their roles within the library system.
- Fine Calculation: Implement an automated fine calculation system to penalize students who fail to return books within the stipulated time, promoting timely returns.
- User-Friendly Interface: Design an intuitive and user-friendly interface for both students and librarians to ensure ease of use and efficient navigation through the system.
- Accessibility: Ensure that the system is accessible to users with varying levels of technical expertise, making it inclusive for all members of the academic community.
- Security Measures: Implement security features to safeguard sensitive information, ensuring the confidentiality and integrity of student and staff data.
- Reporting and Analytics: Incorporate reporting and analytics tools to provide librarians with insights into book circulation, popular titles, and other relevant data for informed decision-making.
- Notification System: Develop a notification system that alerts students about due dates, renewals, and fines, enhancing communication and reducing instances of overdue books.
- Scalability: Design the system to be scalable, allowing for future updates, expansions, and integration with additional features or technologies.

- Documentation: Create comprehensive documentation for the system, including user manuals and technical documentation, to facilitate understanding and troubleshooting.

1.4 Expected Outcome

The expected outcome of your library management system project is a fully functional and efficient software solution that addresses the needs of both librarians and students. The system should seamlessly integrate into the existing library infrastructure and provide a range of features and benefits. Here's a description of the expected outcomes:

User-Friendly Interface: The system should have an intuitive and user-friendly interface that makes it easy for both librarians and students to navigate and perform necessary tasks.

Efficient Book Transactions: Students should be able to issue, renew, and return books with minimal effort, reducing the time and paperwork traditionally associated with these transactions.

Real-Time Updates: Librarians should have access to real-time updates, ensuring that the library database is consistently accurate and up-to-date.

Comprehensive Student Management: The system should provide librarians with a comprehensive view of student information, including borrowing history, contact details, and any outstanding fines.

Effective Book Management: Librarians should be able to efficiently manage the library's book inventory, including adding new books, removing outdated ones, and updating book information as needed.

Automated Fine Calculation: Fines for overdue books should be calculated automatically, ensuring fair and consistent enforcement of library policies.

Security Measures: The system should include robust security measures to protect sensitive information, ensuring the confidentiality and integrity of student and staff data.

Reporting and Analytics: Librarians should have access to reporting and analytics tools that provide insights into book circulation, popular titles, and other relevant data for informed decision-making.

Adherence to Policies: The system should align with existing library policies and procedures, promoting consistency and adherence to institutional guidelines.

Scalability: The system should be designed to be scalable, allowing for future updates, expansions, and integration with additional features or technologies.

The successful implementation of these outcomes will result in a modern, efficient, and well-integrated library management system that enhances the overall library experience for both librarians and students.

1.5 Project Management and Finance

Managing a project involves various aspects, including project management and finance. Here's a brief overview of these two critical components for your library management system project:

Project Management:

Project Scope: Clearly define the scope of the project, outlining the specific features and functionalities your library management system will include.

Resource Allocation: Identify and allocate resources such as human resources (team members, stakeholders), technology, and any external services or tools required for the project. In this project we use Mongodb, Express.js, React.js, Node.js.

Task Assignment: Assign specific tasks and responsibilities to team members based on their skills and expertise. Clearly communicate expectations and deadlines.

Risk Management: Identify potential risks and develop a risk management plan. This includes anticipating challenges, outlining mitigation strategies, and having contingency plans in place.

Quality Assurance: Implement quality assurance measures throughout the project to ensure that the final product meets the specified requirements and standards.

Testing and Debugging: Plan for thorough testing of the system, including unit testing, integration testing, and user acceptance testing. Allow time for debugging and fixing any issues that arise during testing.

Documentation: Keep comprehensive documentation throughout the project, including design documents, code documentation, and user manuals. This facilitates understanding and future maintenance.

Deployment Plan: Develop a deployment plan that outlines how the library management system will be rolled out, including any training sessions for users and staff.

Finance:

Budgeting: Create a detailed budget that includes estimated costs for personnel, technology, software licenses, hardware, and any other resources required for the project.

Cost Monitoring: Monitor project costs regularly to ensure that expenses align with the budget. Address any deviations promptly and efficiently.

Resource Optimization: Optimize resource utilization to minimize unnecessary costs. This involves effective time management, avoiding scope creep, and ensuring efficient use of technology.

Return on Investment (ROI): Consider the potential return on investment for the library management system. Assess how the system will bring value to the institution and justify the project costs.

Funding Sources: Identify and secure funding sources for the project. This may involve collaboration with the educational institution, grants, or other financial support mechanisms.

Financial Reporting: Develop a system for financial reporting, providing stakeholders with clear insights into how project funds are being used and the progress of the project.

Post-Implementation Review: After the project is completed, conduct a post-implementation review to assess the actual costs against the budget and identify lessons learned for future projects. By effectively managing both the project and its finances, you increase the likelihood of delivering a successful library management system that meets the specified requirements while staying within budget and timeline constraints.

1.6 Report Layout

This report consists of 6 chapters. These are:

Introduction

- Introduction
- Objectives
- Motivation
- Expected Outcome
- Project Management
- Report Layout

Background

- Preliminaries / terminologies
- Related Works
- Comparative Analysis
- Scope of the Problem
- Challenges

Software Requirement Specification

- Business Process Model
- Requirement Collection and Analysis
- Use Case Model

- Login Data Model
- Design Requirement

Design Specification

- Front-end Design
- Back-end Design
- Interaction Design and User Experience
- Implementation Requirements

Implementation and Testing

- Implementation of Database
- Front-End Interaction
- Testing Implementation
- Test Results and Reports

Impact on Social Environment

- Impact on Society
- Impact on Environment
- Ethical Aspect
- Sustainability Plan

Conclusion and Future Scope

- Conclusion
- Future Scope

CHAPTER 2

BACKGROUND

2.1 Preliminary and Terminology

A library management system is a web application that automates and manages various library tasks, including book circulation, cataloging, and student management. It involves user roles such as librarians, students, and staff. Librarians manage the overall library system, while students interact with the system to borrow, return, and renew books. Staff have limited access compared to librarians.

Book management involves cataloging, book status, fine system, borrowing and returning, member ID, and account balance. Library users can check out, renew, or check in, and their borrowing history and account details are recorded. Students can track their fines and account balance using unique identifiers, while system administrators can backup and restore data to prevent loss in case of system failures. User authentication is also necessary to protect against unauthorized access.These concepts form the foundation for a library management system project, ensuring a common understanding of key terms and functionalities among developers and end-users. By implementing these concepts, libraries can streamline their operations and improve user experience.

2.2 Related Works

Openlibrary.org: Openlibrary.org offers a wide array of books that can be read or borrowed from this digital library. Furthermore, users can effortlessly retrieve any book by utilizing the search tool. In order to access and borrow books, users are required to register and log in, as well as make a payment. A modifiable library catalog can be accessed at this accessible library.

Amazon Books: This online library system is primarily utilized for commercial endeavors. It is a network of autonomous bookstores that is owned by the e-commerce giant Amazon. They possess a total of 24 shops. Essentially, they distribute publications authored by a diverse range of writers. The user is required to purchase the physical edition of the book from Amazon.

DIU library: The DIU Library is the official website of Daffodil International University, offering comprehensive information on several aspects such as Admission, Academics, Campus, Research, International Affairs, as well as the History and Achievements of the university. Within the Campus, there is a specific section that contains various details about several facilities.

Koha: Koha is the initial open-source software system for library automation. With a global presence, its progress is guided by an expanding community of users who work together to accomplish their technological objectives. Koha's range of features is always developing and growing in order to suit the requirements of its user community.

DU Library: The website of the Dhaka University Library features a comprehensive catalog search for books, DVDs, CDs, journals, theses, manuscripts, and music. The website additionally provides hyperlinks to the university's homepage, DU e-archives, daily newspapers, Koha Wiki Home, and Koha Community.

2.3 Comparative Analysis

A comparative analysis of a web-based library management system project involves evaluating its features, functionalities, and performance against existing or similar systems. Key aspects to consider include user interface and user experience, technology stack, database model, real-time updates, scalability, security, and learning curve.The MERN stack (MongoDB, Express.js, React.js, and Node.js) is a popular choice for this project due to its unified JavaScript environment for front-end and back-end development. MongoDB offers flexibility in handling unstructured data, while the React.js library facilitates a responsive and interactive user interface. However, the learning curve for developers unfamiliar with JavaScript or the MERN stack may be a potential consideration.

An alternative technology stack, such as LAMP (Linux, Apache, MySQL, PHP/Python/Perl), offers widespread usage and community support but may require separate languages for front-end and back-end development. The flexible schema of MongoDB allows for easy modification of data structure, but it may not be suitable for complex transactions requiring ACID properties.Real-time updates are enabled by technologies like WebSocket and Socket.io, making it suitable for collaborative environments. Scalability is supported by the MERN stack, but proper architecture planning is required for optimal scalability. Security measures are crucial, and alternatives like Ruby on Rails with PostgreSQL offer built-in security features.In conclusion, the choice of the MERN stack aligns with modern development practices, flexibility, and scalability. However, each technology stack and alternative has its unique advantages and considerations, emphasizing the importance of aligning choices with project requirements and team expertise.

2.4 Scope of the Problem

The platform we are developing is a web-based application, ensuring a user-friendly and accessible platform accessible from any location. This platform is different from standalone software or Android applications, as it only requires an internet connection for access. We will address the manual processes used in library tasks, such as book cataloging, member registration, and inventory management, and the inefficiencies associated with these processes. We will also address the lack of a centralized system for managing library-related information, highlighting the need for real-time access and data management. We will also address the challenges faced by library staff and members in accessing and interacting with library services, emphasizing the importance of a system that enhances accessibility, convenience, and satisfaction. Defining the scope helps manage expectations, avoid scope creep, and deliver a solution that meets identified needs within specified boundaries.

2.5 Challenges

A library management system project involves several challenges, including understanding user requirements, integrating with existing systems, ensuring data security and privacy, scalability, user adoption, balancing customization vs. standardization, frequent system changes, technical complexity, budget constraints, and testing and quality assurance. Understanding the diverse needs of librarians, students, and other stakeholders is crucial, and conducting thorough interviews, surveys, and workshops can help refine requirements. Integrating the new system with existing systems, using standardized data formats, and collaborating closely with IT teams can be challenging. Data security and privacy are also critical, and robust encryption, access controls, and regular security audits are essential.Scalability is crucial for the growing library collection and user base, and designing the system with scalability in mind and using appropriate technologies can help address potential bottlenecks. User adoption can be met with resistance, so providing training sessions, creating user-friendly interfaces, and involving end-users in the development process can help. Balancing customization with standardization can be challenging, but identifying core functionalities that need standardization and providing configurable options can help.Frequent system changes can disrupt the development process, so establishing a clear change management process, conducting regular stakeholder meetings, and documenting changes systematically can minimize disruptions. Lastly, ensuring the system's reliability and identifying and fixing bugs can be time-consuming, so implementing comprehensive testing strategies and adopting agile development methodologies can help overcome these challenges.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Model

The business process model is a method for displaying a particular system's workflow. It is possible to visualize the data flow of this web application using a business process model. To utilize the service, a user or reader must first complete out the sign-up form on our website and register. He must log in using his username and password after signing up. A user can access all of our website's features after successfully logging in. A user may perform a search for the book they want to borrow.

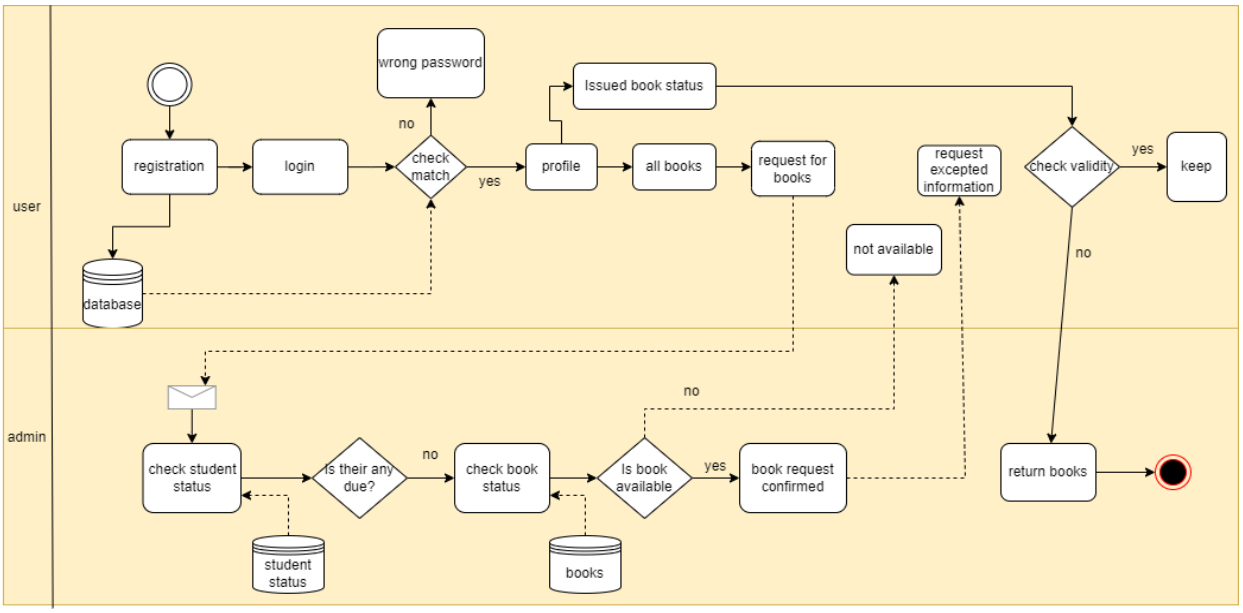

Figure 3.1.1: Business Process Model of Library Management System

3.2 Requirement Collection and Analysis

Requirement Collection: The process began by engaging with stakeholders, including students, librarians, and administrative staff, to gather insights into their expectations and challenges. Meetings, interviews, and surveys were conducted to collect requirements. Key areas of focus included:

User Needs: Understanding the requirements and expectations of students, librarians, and administrative staff in terms of functionality, accessibility, and ease of use.

Functional Requirements: Identifying the essential features such as book transactions, student management, staff administration, and notification systems.

Technical Requirements: Determining the technology stack, infrastructure, and compatibility requirements for seamless integration and operation.

Security and Privacy Requirements: Ensuring the implementation of robust security measures to protect user data and maintain privacy.

Compliance Requirements: Identifying legal and regulatory requirements, including intellectual property rights, data protection laws, and other relevant standards.

Requirement Analysis:

Upon collection, the gathered requirements underwent a thorough analysis to determine their feasibility, relevance, and potential impact on the system. This phase involved:

Prioritization: Prioritizing requirements based on their importance and impact on the system's functionality and usability.

Feasibility Study: Assessing the technical and operational feasibility of implementing various features and functionalities.

Risk Analysis: Identifying potential risks associated with specific requirements and developing mitigation strategies.

Alignment with Goals: Ensuring that the identified requirements aligned with the overall goals and objectives of the library management system project.

User Experience Analysis: Evaluating how the proposed features and functionalities contribute to a positive and intuitive user experience.

3.3 Use Case Modeling and Description

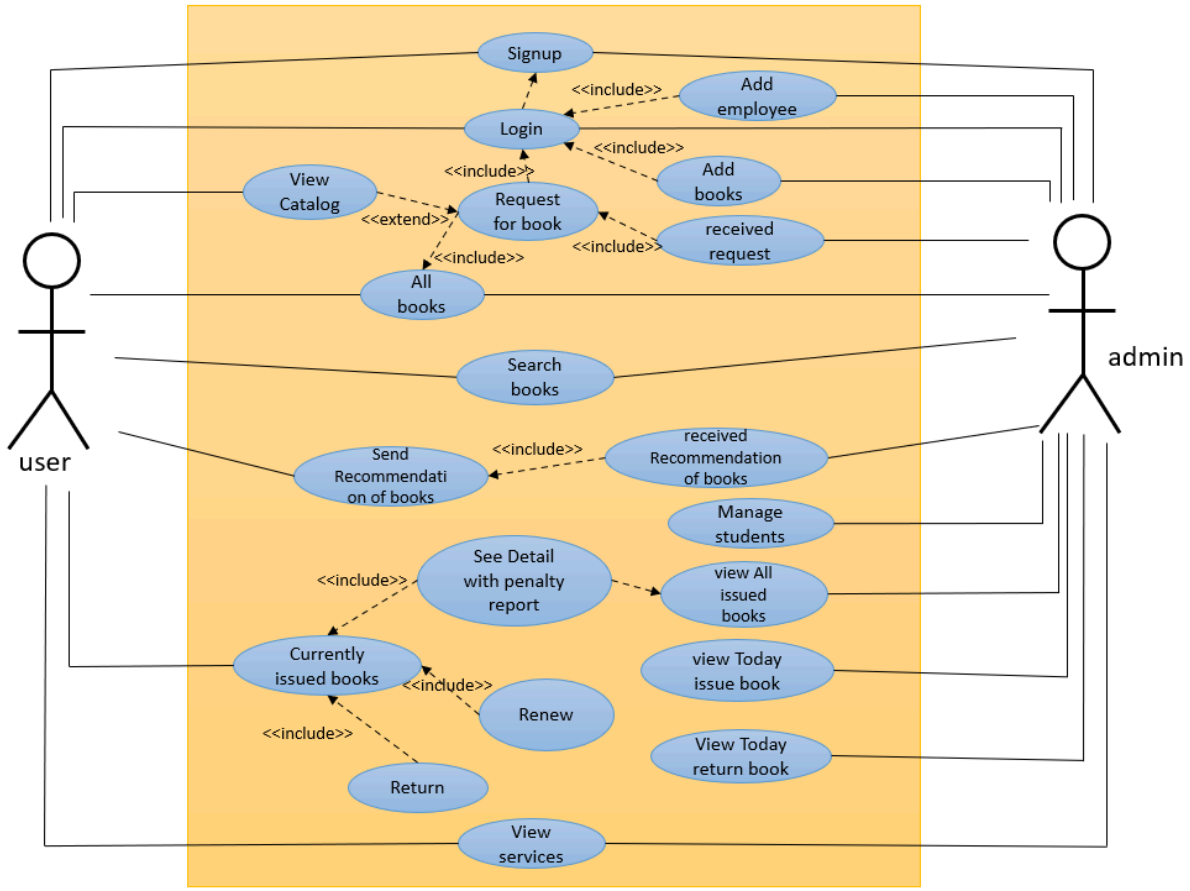

Figure 3.3.1: Use case diagram of Library Management System

The use case diagram consists of two characters: user and administrator. Users can access services, libraries, and book catalogs without a login. To access the website, users must complete the signup form and log in. Users can search for books to borrow or issue, while administrators can add new books, manage employees, and manage students. Users can search for books for seven days and send an issue request to the admin. The admin receives the request and can accept or reject it. After accepting the request, users can view the book in detail on the currently issued book page. Users can also return books, and if they don't return them in time, a small fine will be added. Overall, the user-agent relationship is crucial for efficient and effective book management.

Use Case Name	LogIn
Prerequisite	Registration
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Enter Registration Button
	Fill Require Information
	Submit Data
	Now LogIn
Exceptional Path	Registration Form
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.1: Login Use Case of Library Management System

The Login Use Case of Library Management System shows how to login into the system step by step.

Use Case Name	View Catalog
Prerequisite	None
Actor	User
Internal Path	Enter Website
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.2: View Catalog Use Case of Library Management System

The View Catalog Use Case of Library Management System shows the collection of books in the library moreover it shows how it works.

Use Case Name	Request Book
Prerequisite	Registration or LogIn
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to All Book Page
	Search Your Book
	Click Issue Button
Exceptional Path	From Catalog
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.3: Request Book Use Case of Library Management System

The Request Book Use Case shows how to request for a book using the system and what process should follow for a request for a book.

Use Case Name	All books

Prerequisite	Registration or Login
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to All Book page
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.4: All books Use Case of Library Management System

If users want to see the book collections of the library, they must have to login in the system. Otherwise they would not be able to see the collection of books. And this use case shows this functionality.

Use Case Name	Send Recommendation Of Books
Prerequisite	Registration of Login
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to Book Recommendation Page
	Fill Require Information
	Submit Data
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.5: Send Recommendation Of Books Use Case of Library Management System

A user, how to send a recommendation of a necessary book to the librarian, all the processes are discussed in the above table.

Use Case Name	Currently Issued Books
Prerequisite	Registration Login
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to Currently Issue Book Page
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.6: Currently Issued Books Use Case of Library Management System

When a user requests a book to the librarian and the librarian accepts the request, then the user can see the book in the currently issued book page. This table shows all that process for perceiving all issued books.

Use Case Name	See Detail with Penalty Report
Prerequisite	Registration or Login
Actor	User
Internal Path	Enter Website
	Click Login Button
	Go to Currently Issue Book Page
	Click Detail Button
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.7: See Detail with Penalty Report Use Case of Library Management System

If a user will not return the issued book in a given dateline, then there will be an extra charge for each day. The user can perceive the penalty fee. To see the penalty fee a user must have followed the above procedure.

Use Case Name	Return Books
Prerequisite	Registration
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to Currently Issue Book Page
	Click Detail Button
	Click Return Button
Exceptional Path	Registration Form
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.8: return Books Use Case of Library Management System

How to return a book to the librarian, all that processes are indicated in the above table.

Use Case Name	Renew Books
Prerequisite	Registration
Actor	User
Internal Path	Enter Website
	Click LogIn Button
	Go to Currently Issue Book Page
	Click Detail Button
	Click Renew Button
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy

	Volunteered by Mr. Saiful Islam

Table 3.3.9: Renew Books Use Case of Library Management System

If a user wants to increase the given deadline for issued books, he can renew or reissue the book for a certain period.

Use Case Name	LogIn
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Click LogIn Button
	Fill Require Information
	Submit Data
Exceptional Path	Pre Registered
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.10: Admin Login Use Case of Library Management System

For admin login, admin has to give his id and password for login. Because admin is manually registered by the system controller.

Use Case Name	Add Books
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Click Add Books Button
	Fill Require Information
	Submit Data

Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.11: Add Books Use Case of Library Management System

If there are some new collections of books coming to the library then the admin has to update the book list. He can add books in the library records following the above process.

Use Case Name	Issue Request
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Click Issue request Button
	Except or Reject
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.12: Issue Request Use Case of Library Management System

When a user requests a book to the admin, then the request comes through this issue request page to the admin. Admin has two options for the request that are accept and reject.

Use Case Name	Receive Recommendation of Books
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Go to Book Recommendation Page

	Accept or Delete
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.13: Receive Recommendation of Books Use Case of Library Management System

This is also the same as Issue request. Admin can accept or delete the request from users.

Use Case Name	Manage Students
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Go to Manage Student Page
	Action: Removed
Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.14: Manage Students Use Case of Library Management System

Use Case Name	Add Employee
Prerequisite	Admin Login
Actor	Admin
Internal Path	Enter Website
	Go to Book Recommendation Page
	Fill Require Information
	Submit Data

Exceptional Path	None
Notes	Written by Debabrata Roy Chinmoy Volunteered by Mr. Saiful Islam

Table 3.3.15: Add Employee Use Case of Library Management System

Admin can add librarians. For adding librarian admin has to enter the add employee page. After that he has to give input employee name, employee password and employee id and submit the information. After all that processes the employee add will be completed.

3.4 Logical Data Model

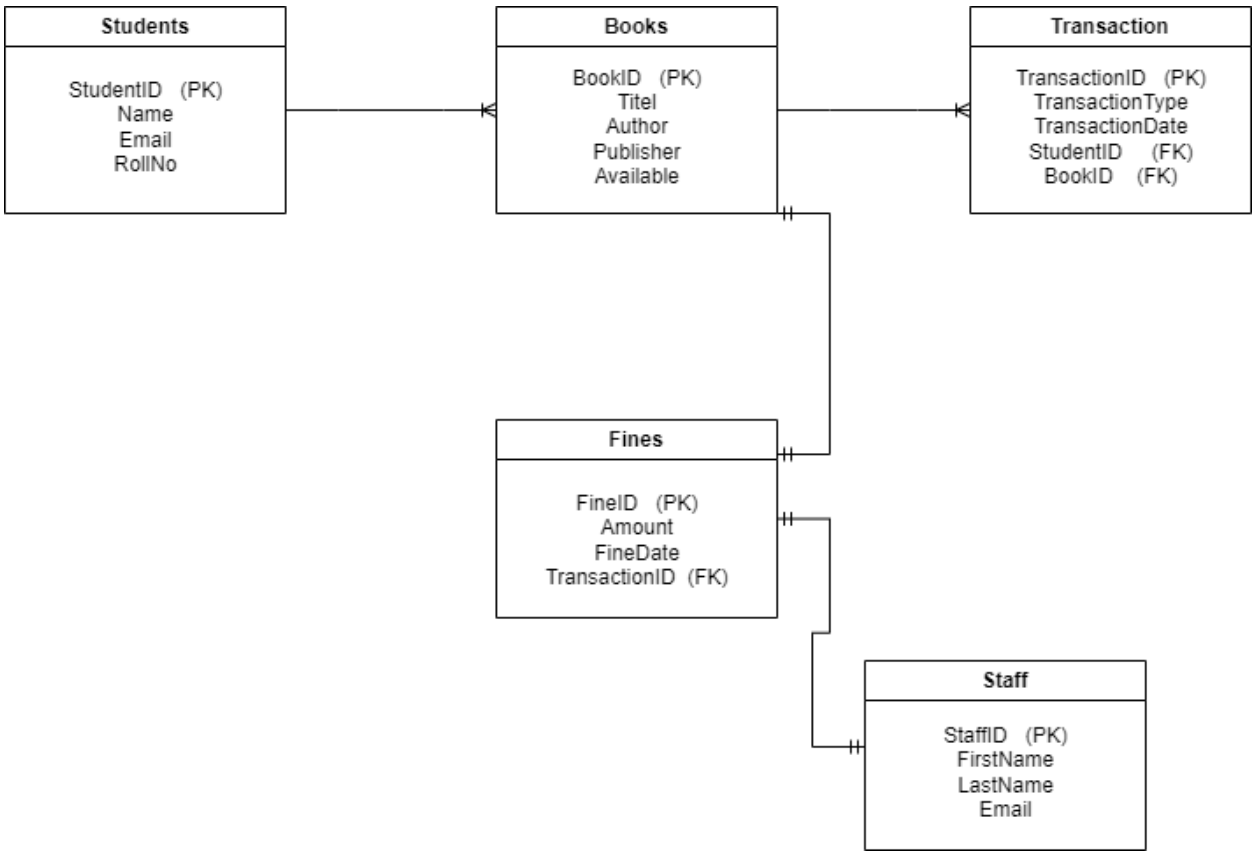

Figure 3.4.1: Logical Data Model of Library Management System

3.5 Design Requirements

The design requirements for the library management system project revolve around creating a seamless, user-friendly, and efficient system. The user interface design should prioritize intuitiveness, ensuring that students and librarians can easily issue, renew, and return books. Real-time updates for librarians should cover student management, book addition, deletion, and staff management. A key feature is the automated fine system, imposing penalties if students fail to return books within the specified timeframe. The MERN stack integration should facilitate scalability, security, and optimal system performance. Flexibility in design allows for future technological adaptations, while accessibility features ensure a positive user experience across devices. Integrating with external systems, such as Learning Management Systems, enhances the educational experience. The system's mobile responsiveness accommodates users accessing it through smartphones and tablets. User feedback mechanisms, a robust notification system, and usability testing contribute to continuous improvement. Cross-browser compatibility and consistent branding elements enhance the system's visual appeal and accessibility. Overall, the design requirements aim to create a comprehensive and adaptable library management system using the MERN stack.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Back-end Design

The entire project's back-end design is created using HTML, JavaScript, React, and bootstrap. React code is used to design the entire back-end.React is a JavaScript library for creating declarative, effective, and adaptable user interfaces. It enables you to put together complex user interfaces from discrete, small pieces of code.The admin will have complete control over the website's features from the back end.

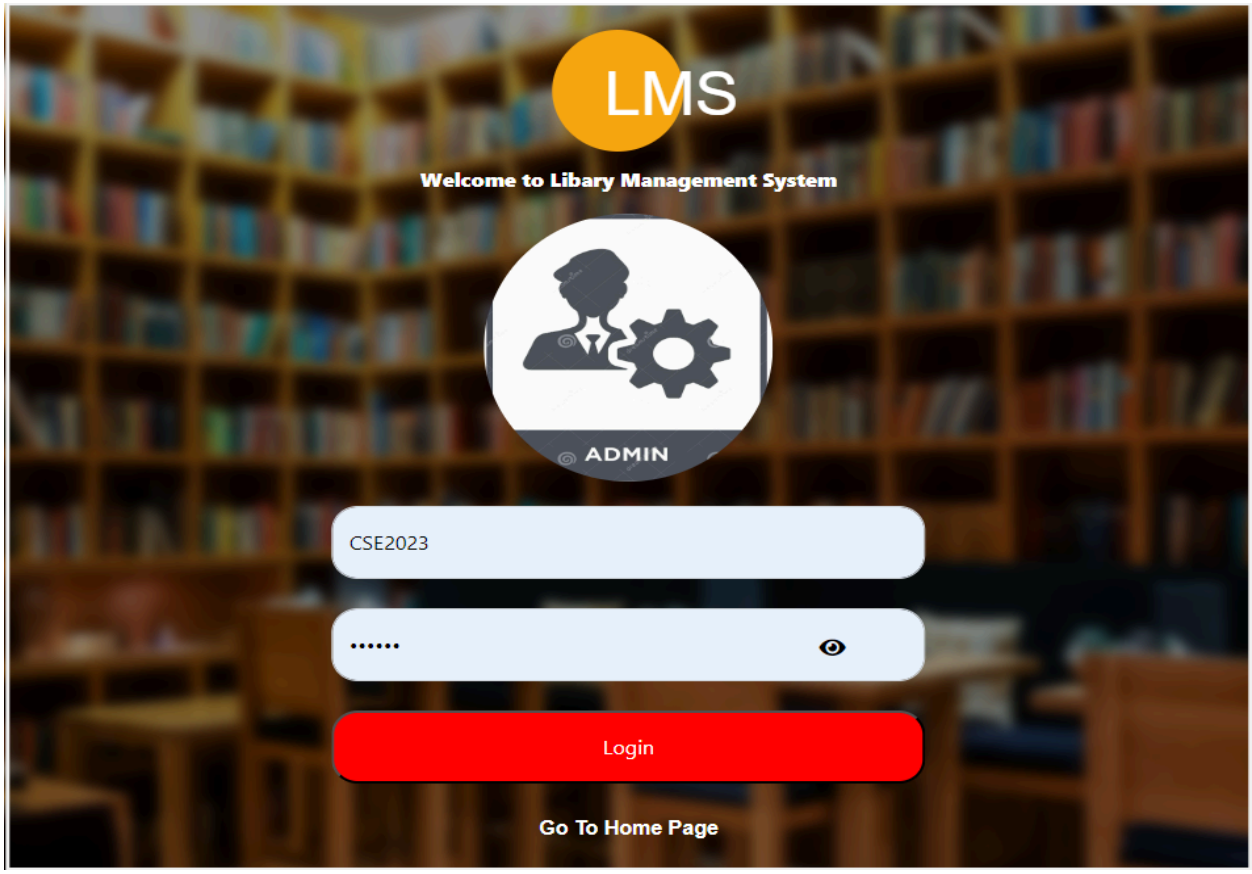

Figure 4.1.1: Admin Login page of Library management system

Figure 4.1.1 shows the admin login page. Admin have to input id number and password for login.

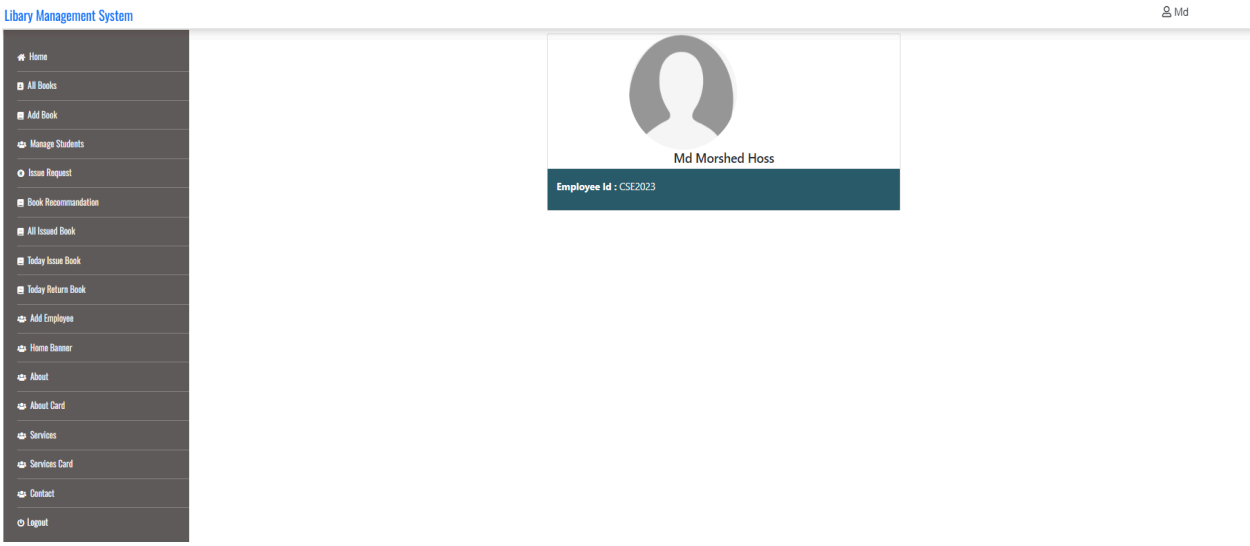

Figure 4.1.2: Admin profile page of Library Management System

Figure 4.1.2 shows the admin profile details. Admin will control the whole system of all books, add books, manage students, issue requests, book recommendations, all issued books, today issue books, today return books, add employee, home banner, about, about card, services, services card, contact etc.

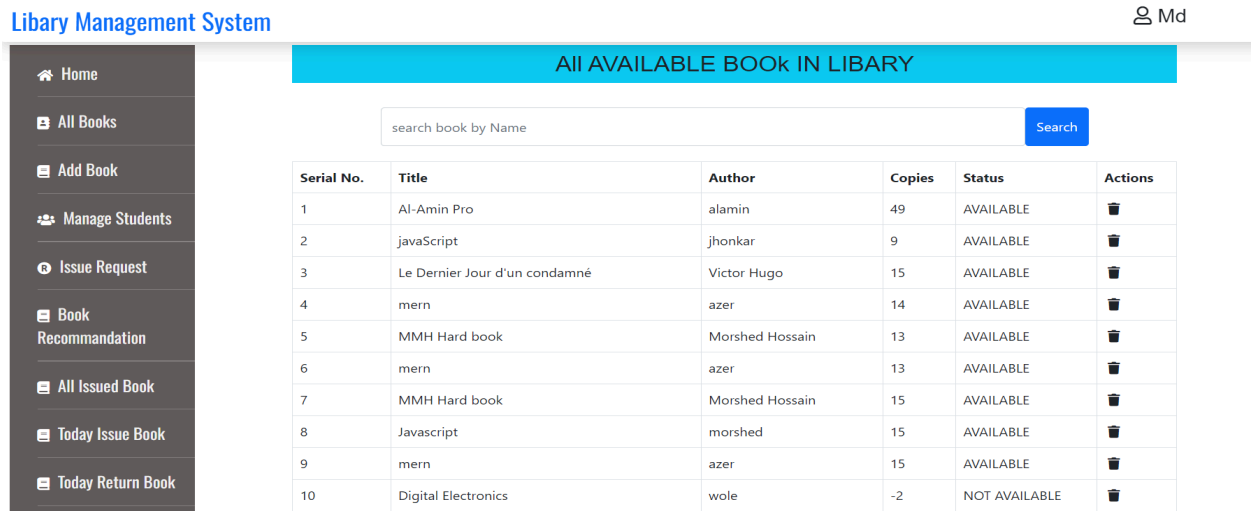

Figure 4.1.3: Book list page of Library Management System

Figure 4.1.3 shows the book list and books information which are available in the library. Admin can delete any type of book.

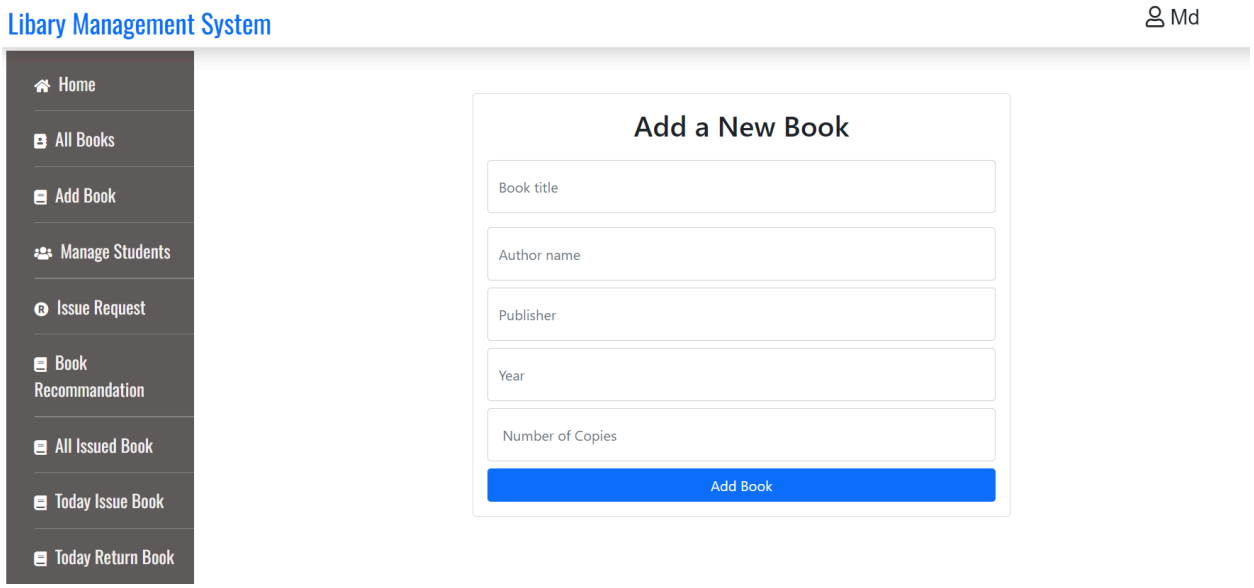

Figure 4.1.4: Add new book page of Library Management System

In Figure 4.1.4 admin can add a new book. For new book addition admin must input book title, author name, publisher, year and number of books copies.

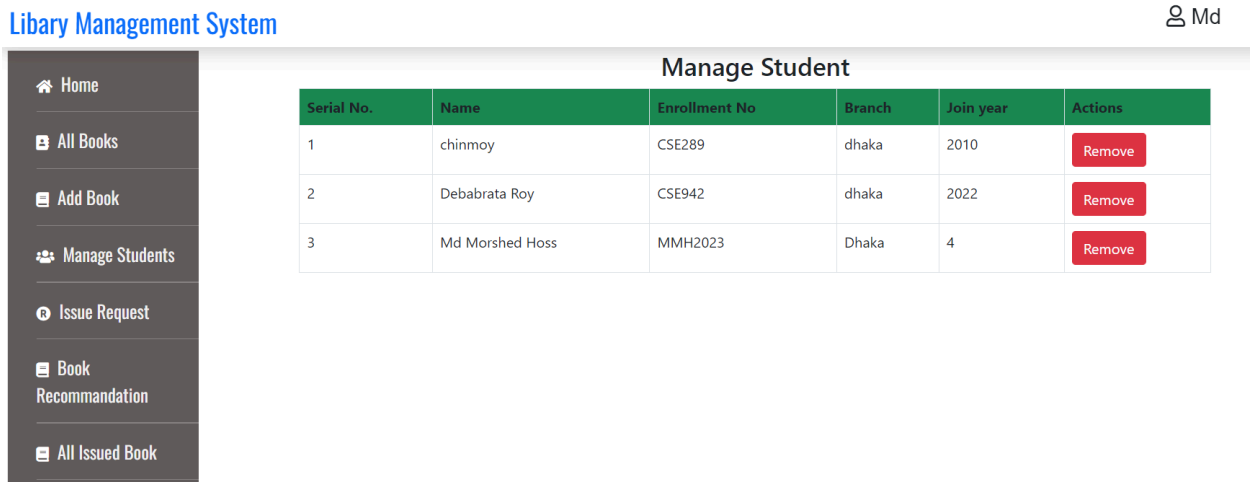

Figure 4.1.5: Manage student page of Library Management System

In this page admin can remove students from the record.

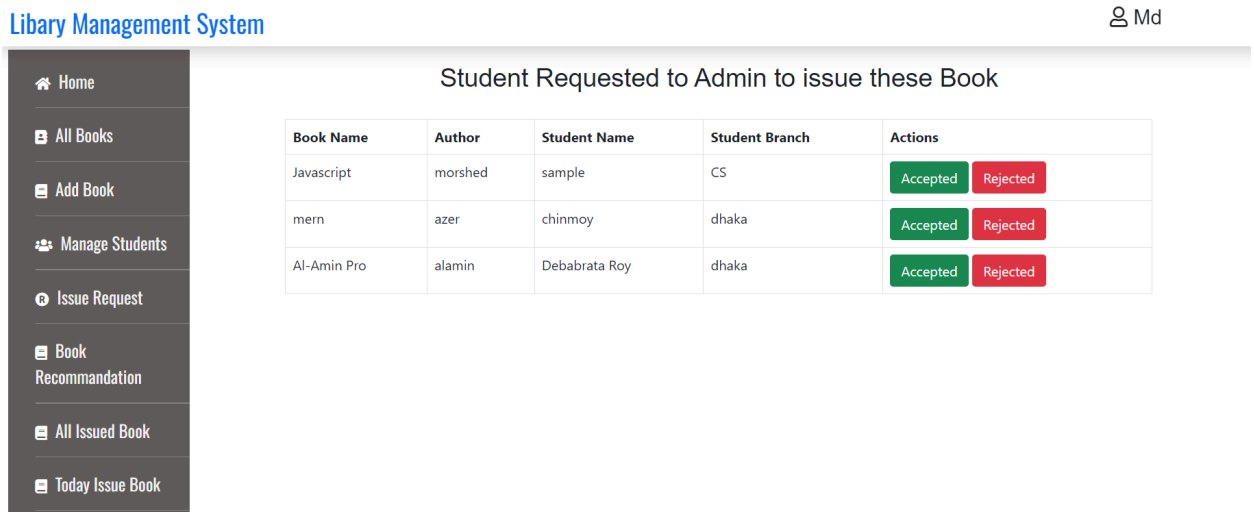

Figure 4.1.6: Book Issue request Page of Library Management System

In this figure the admin accepts or rejects book issue requests from students.

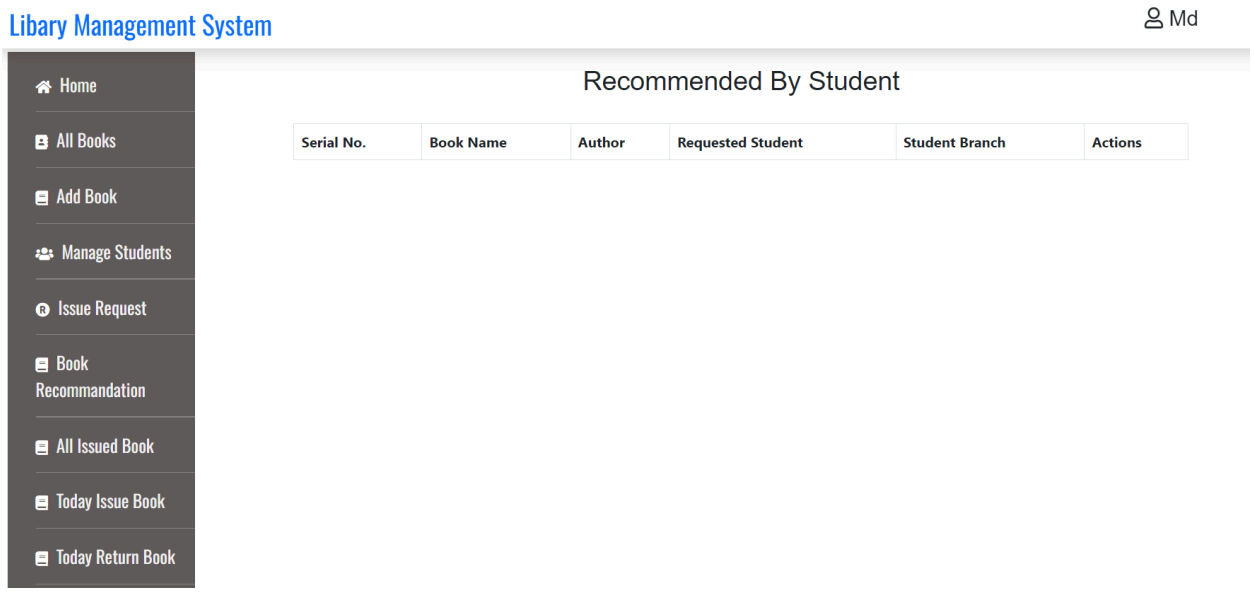

Figure 4.1.7: Book recommendation page of Library Management System

Sometimes the library does not have all the books required by the students, so the students suggest particular books to the librarian. And this suggested book can be viewed by the admin on this page.

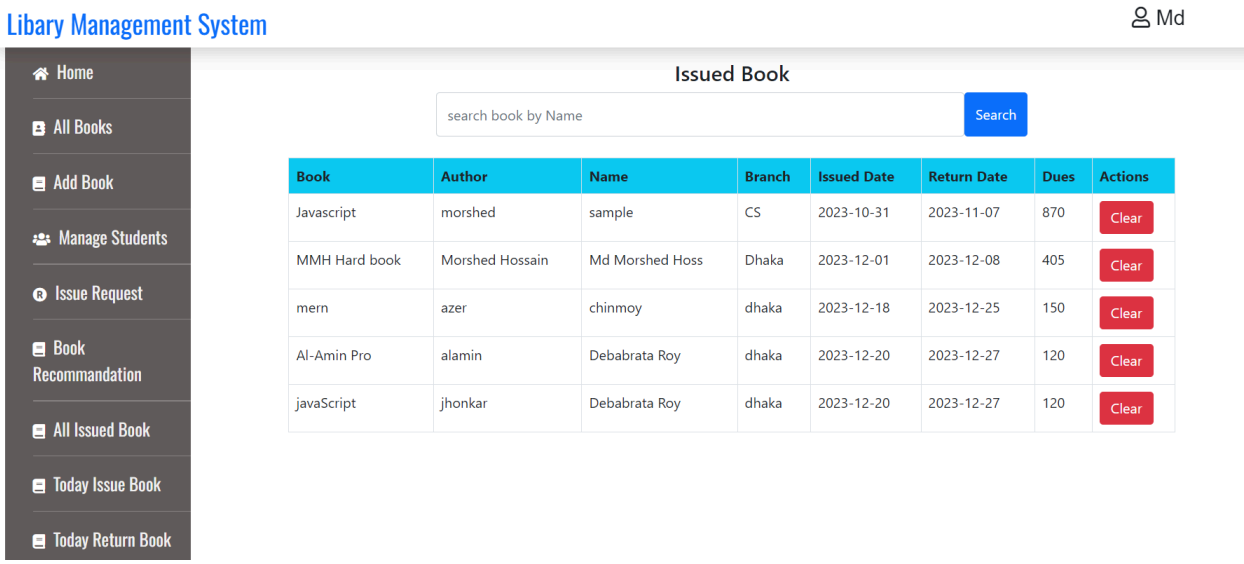

Figure 4.1.8: All issued book page of Library Management System

In this figure admin can see students book issue information and fine. Here the admin can clear the book issue record of a student.

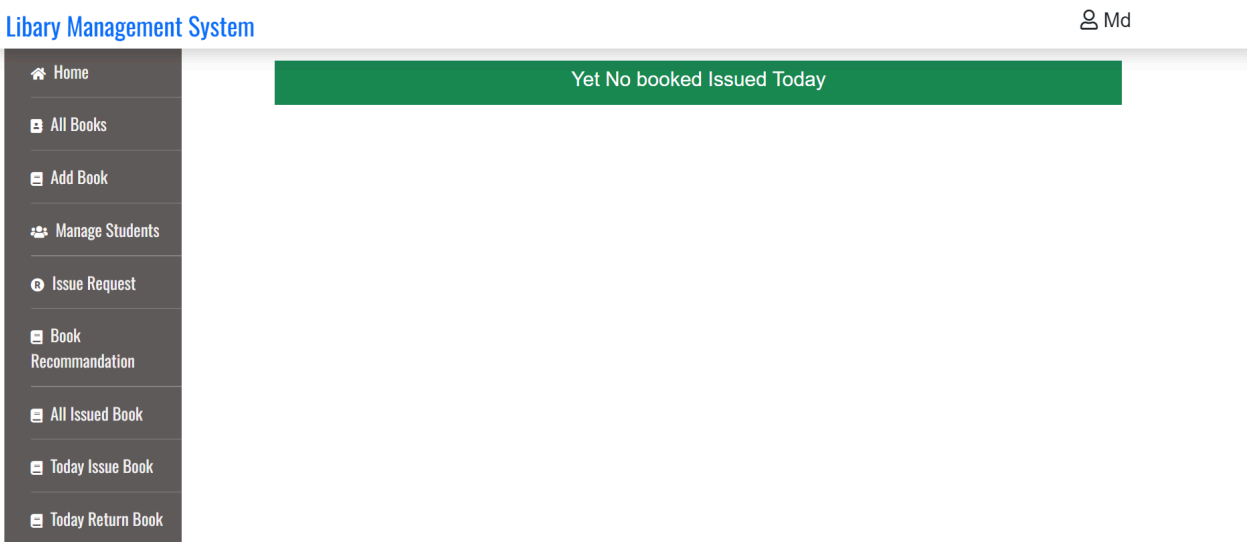

Figure 4.1.9: Today issue book page of Library Management System

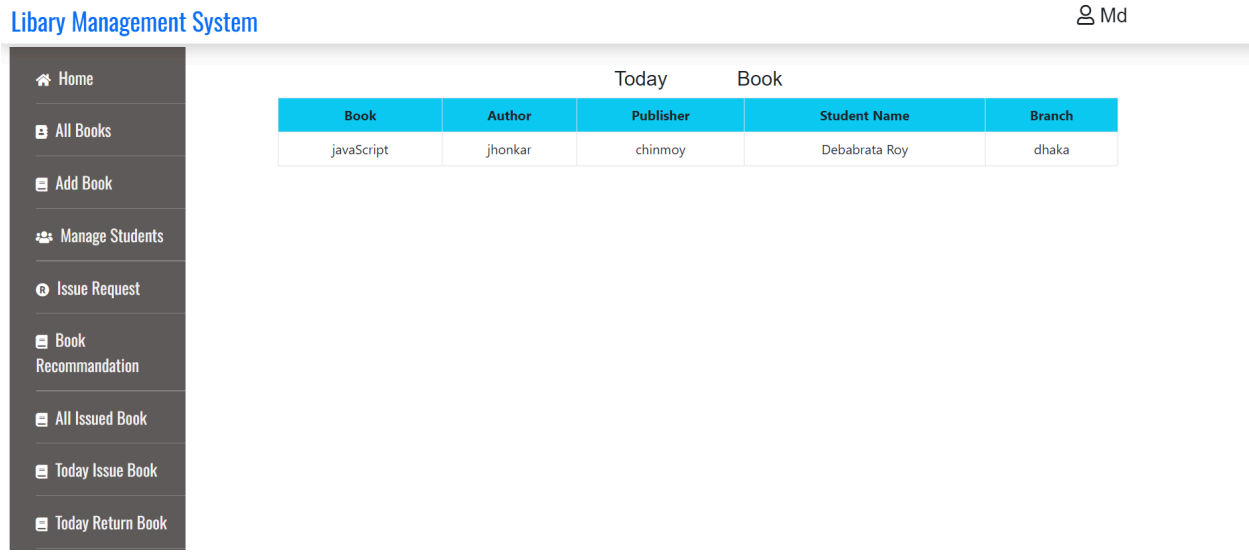

Figure 4.1.10: today return books page of Library Management System

In this figure admin can see which students are returning books today.

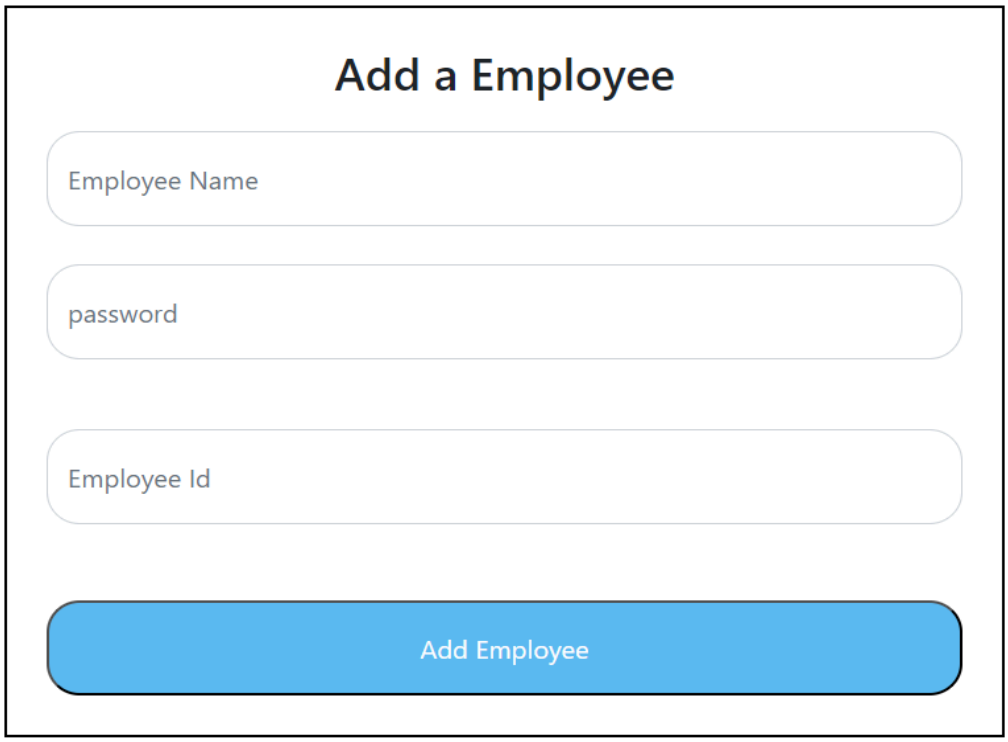

Figure 4.1.11: add employee page of Library Management System

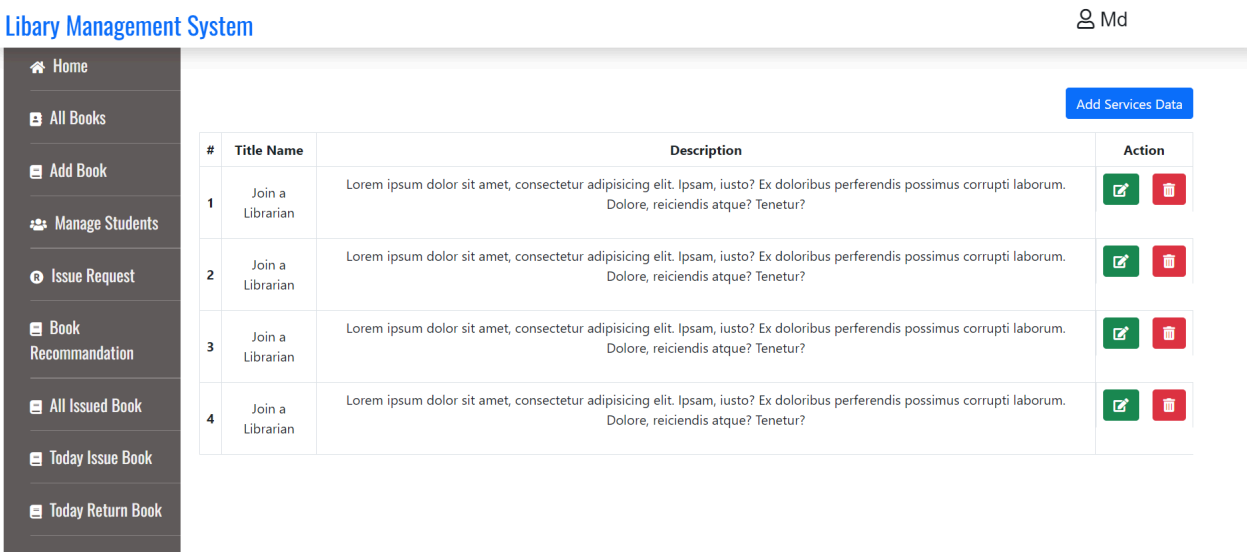

Figure 4.1.12: Add service data page of Library Management System

In this page admin can customize service information which will show in the frontend.

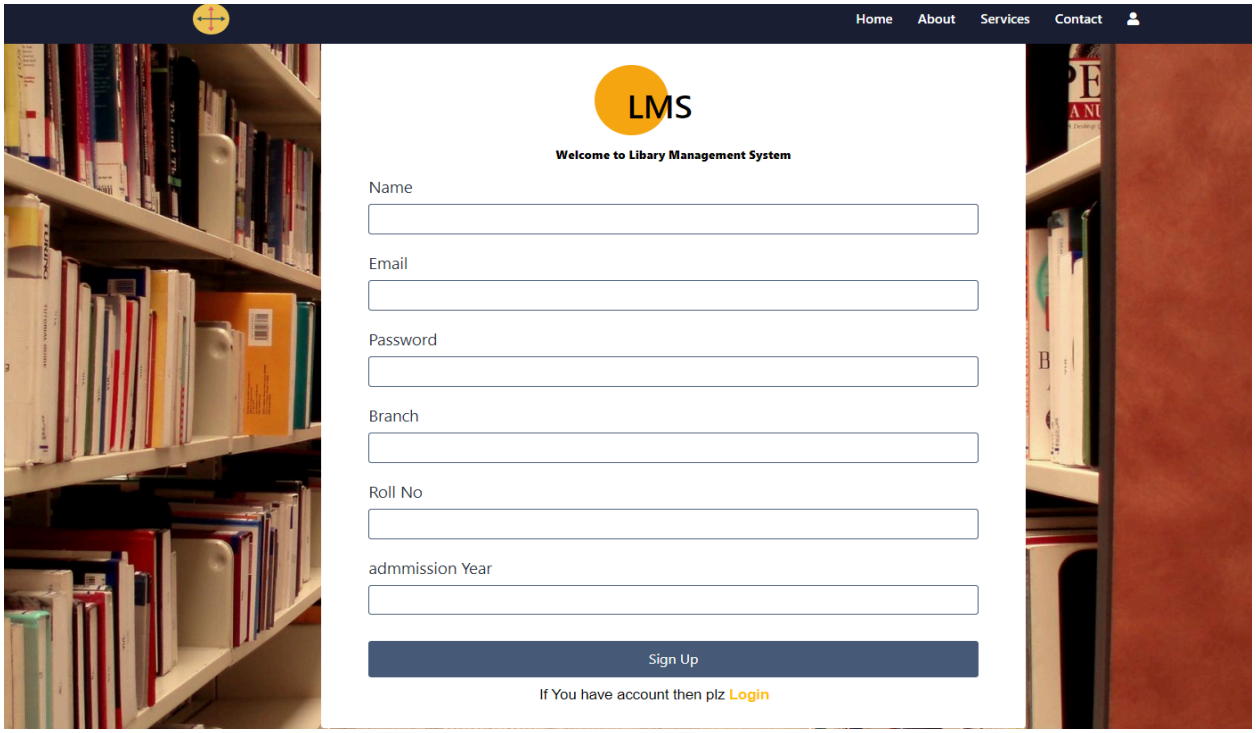

Figure 4.1.13: Student or User registration page of Library Management System

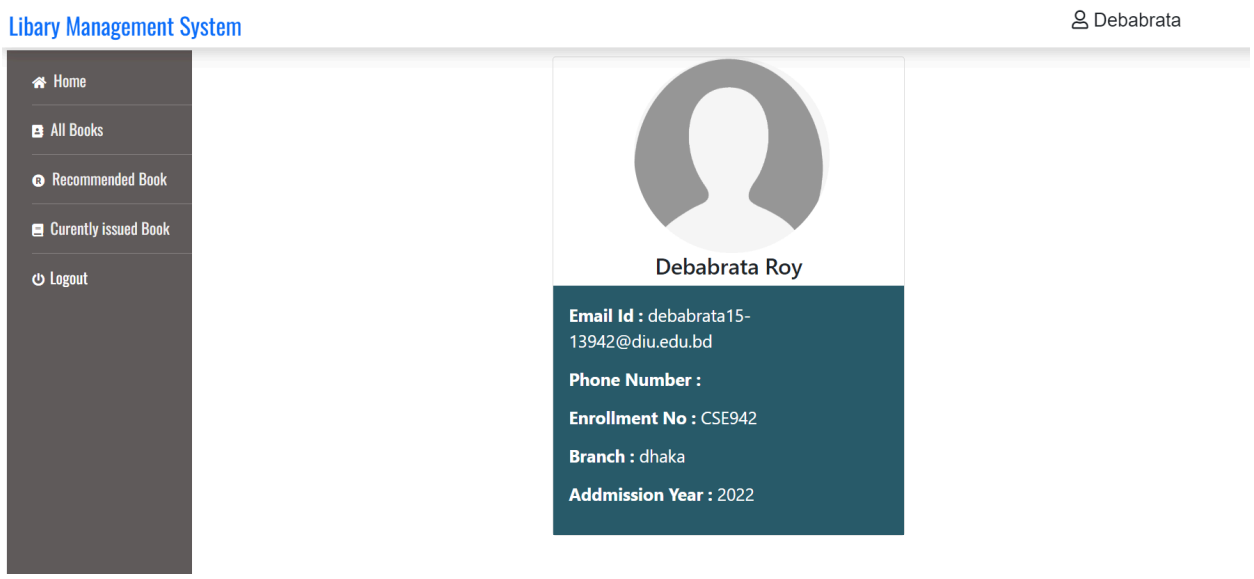

Figure 4.1.14: Student profile page of Library Management System

The figure shows that students can see all books, and students can recommend books to the administrator. And students can also see the issue book status.

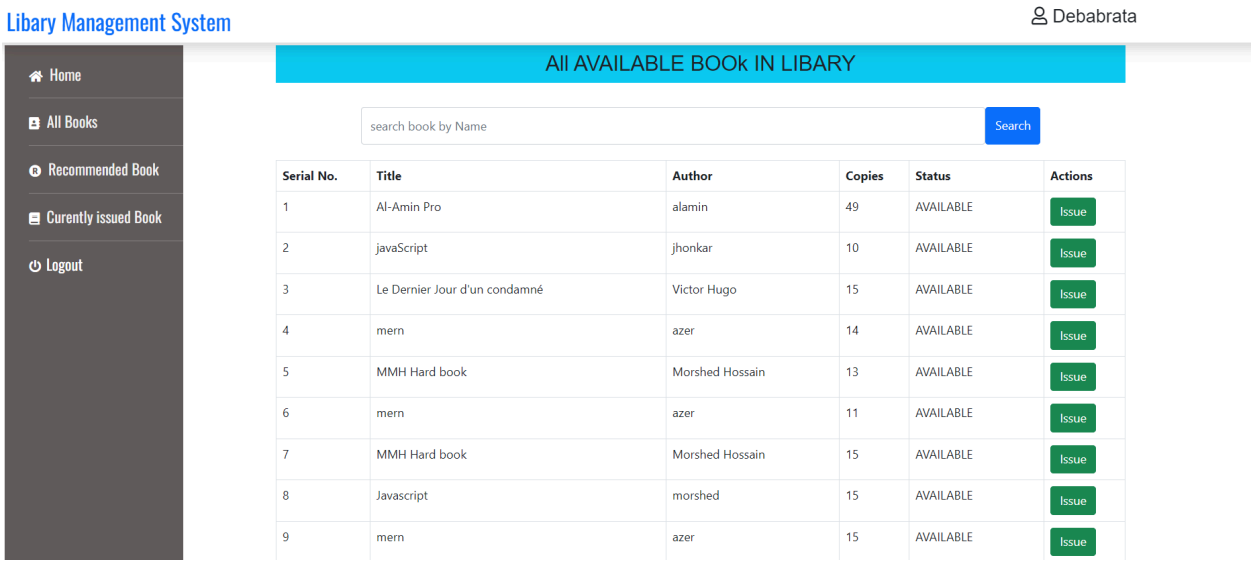

Figure 4.1.15: All available book in library page of Library Management System

In this page students can search books and can see the book information like copies of books. And they can issue books from here.

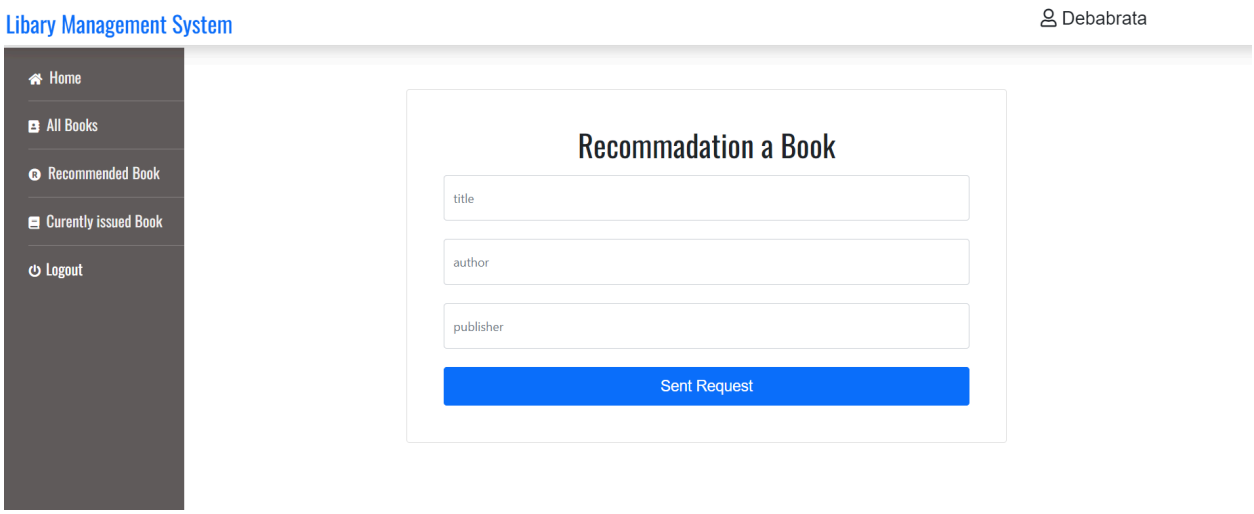

Figure 4.1.16: Recommendation of a book page of Library Management System

For sending a recommendation of a book, students have to fill this form.

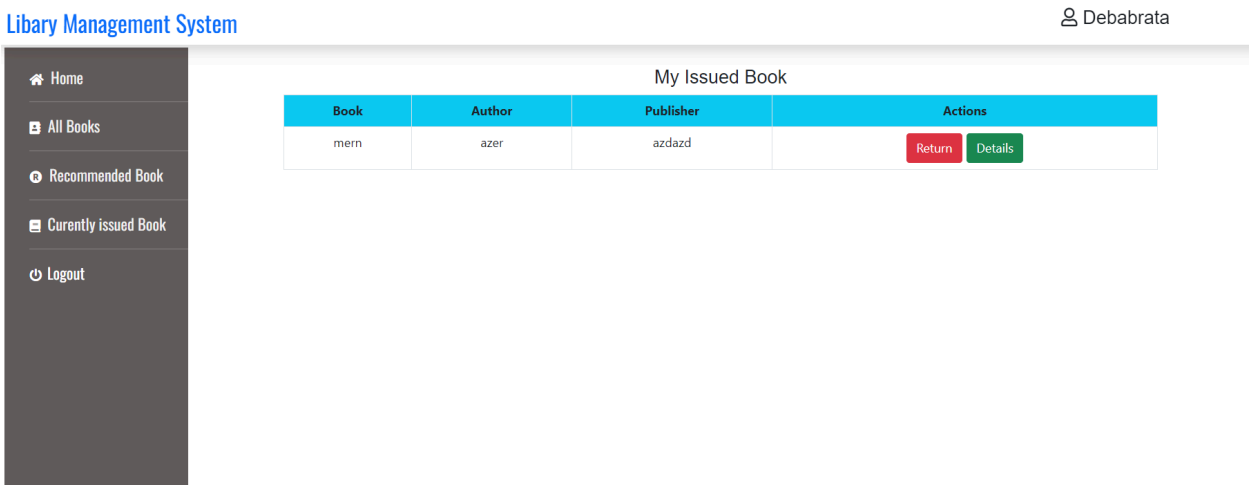

Figure 4.1.17: My issue book page of Library Management System

Students can see detail from this page by clicking the detail button, they can also return books by clicking the return button.

4.2 Front-end Design

I have used react js to implement the front-end. Along with basic HTML, CSS and bootstrap has been used to make the page more attractive. Also json, and various libraries have been used. Various icons, fonts are used to make the website look more beautiful and dynamic so that a user can use the application with ease.

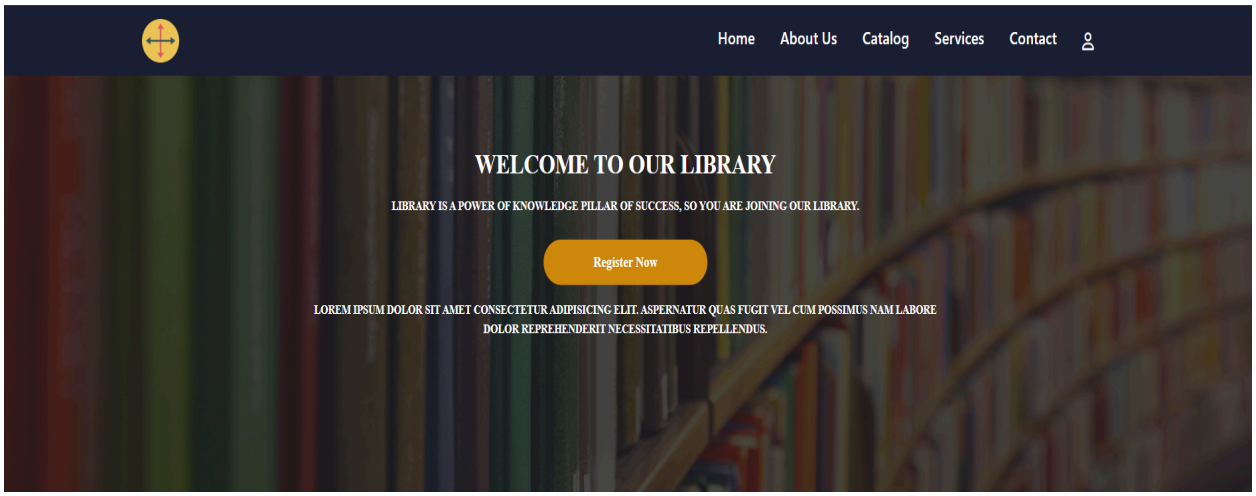

Figure 4.2.1: Home page of Library Management System

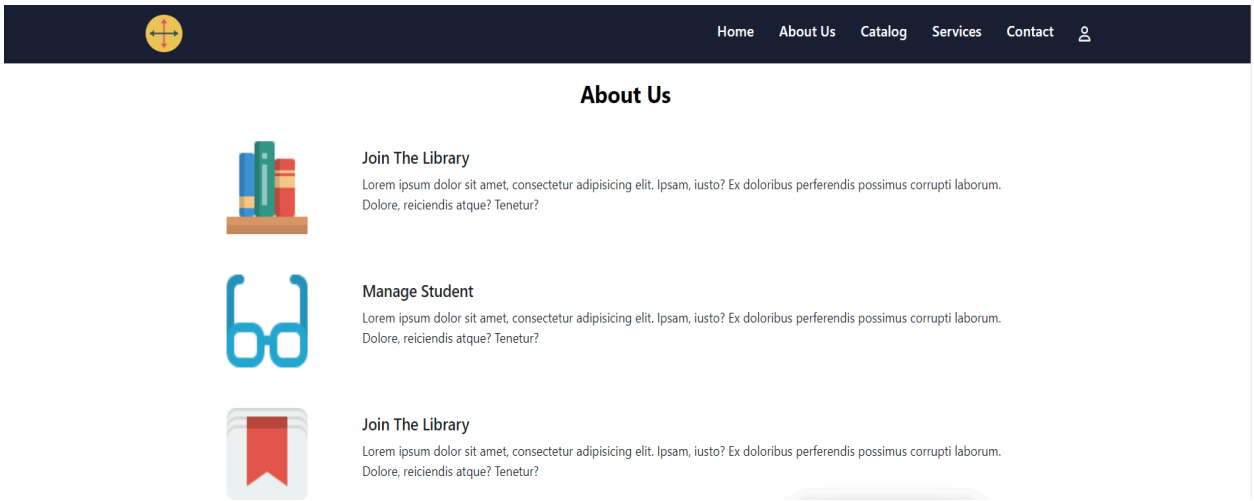

Figure 4.2.2: About page of Library Management System

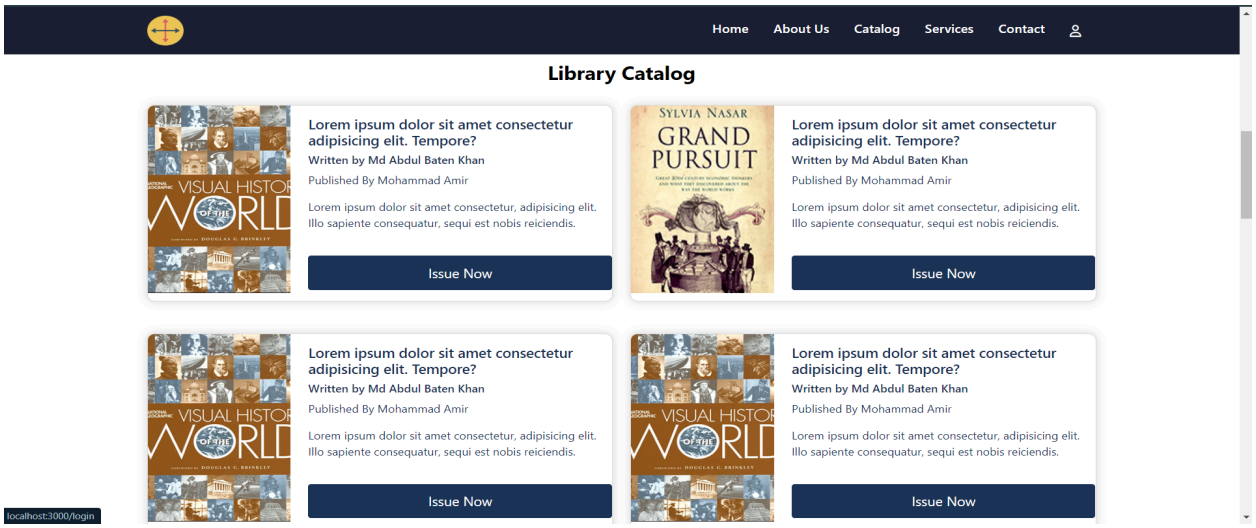

Figure 4.2.3: About page of Library Management System

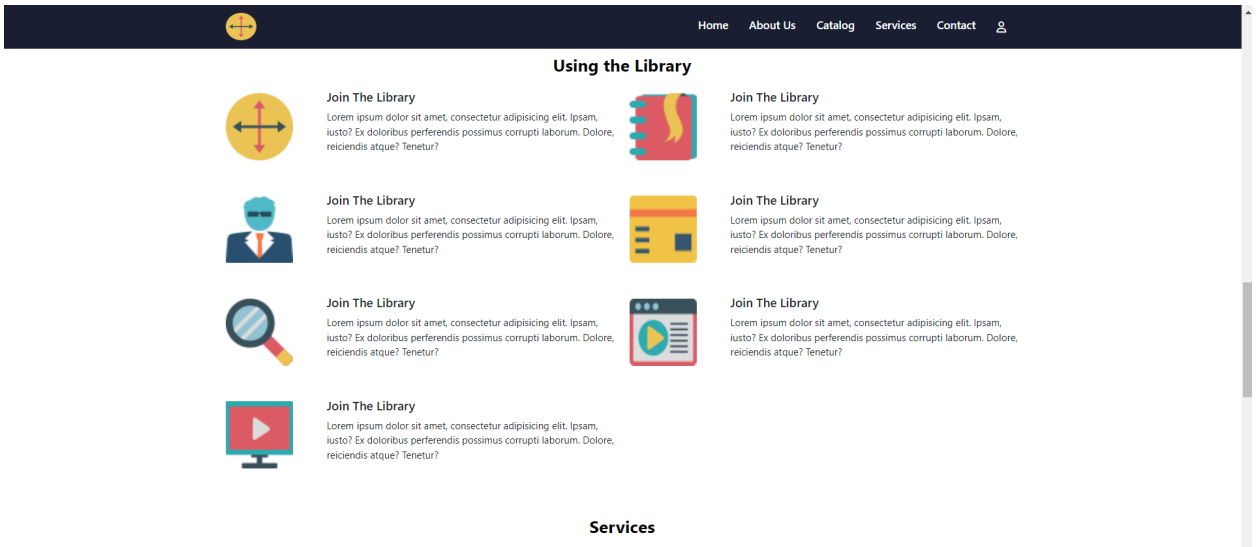

Figure 4.2.4: Service page of Library Management System

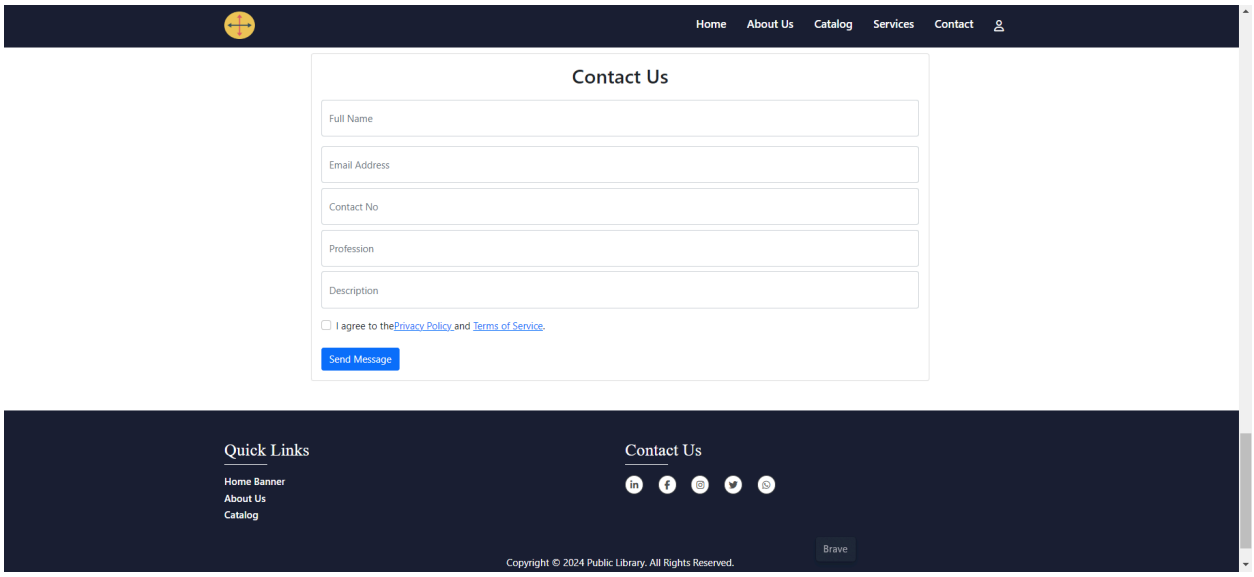

Figure 4.2.5: Contact page of Library Management System

4.3 Interaction Design and User Experience

The MERN stack-based library management system offers a user-friendly interface for librarians and students. It features a secure user authentication process, a visually appealing dashboard, and an intuitive interface for inventory management. The system simplifies book addition, updating, and deletion, while student management is streamlined. Students can easily navigate the system for book issuing, renewal, or returning, with real-time updates and an automatic fine calculation system. The system also includes a robust notification feature for due dates and fines, enhancing user engagement. The system is responsive across devices and is designed to be accessible to users with different abilities. This inclusive design ensures a well-managed and user-friendly library environment, benefiting both librarians and students.

4.4 Implementation Requirements:

The implementation of the library management system using the MERN stack requires careful consideration across multiple layers.

Frontend (MERN):

The project aims to create a user-friendly interface for students and librarians, integrate a secure authentication system, create a centralized dashboard for users, manage books efficiently, and provide features for student management. It also aims to streamline book transactions, provide real-time status updates for students, and implement an automated system for overdue books, ensuring smooth and efficient processes. The system will also provide notifications for students.

Backend (MERN):

The text outlines the process of setting up a Node.js server for backend operations, designing RESTful API endpoints for CRUD operations, managing MongoDB schemas, developing scheduled tasks for automatic book fine calculation, and implementing real-time communication using technologies like WebSocket for instant updates for all users.

Database (MongoDB):

The document outlines the implementation of efficient document schemas for various data types, including books, students, transactions, and fines, to enhance query performance.

Deployment**:**Choose a reliable hosting service and configure domain settings for a user-friendly URL for application deployment. Implement horizontal scalability strategies for potential user and data growth.

Testing:The process involves developing and running unit tests for individual components using frameworks like Jest and React Testing Library, and conducting integration tests for seamless communication between frontend and backend components.

Security:

Data Encryption: Implement encryption for sensitive data, particularly user credentials.

Authorization: Ensure that only authorized users can perform specific actions through proper access controls.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

In the implementation of the library management system using the MERN stack, the database structure is crucial for efficient data management. The MongoDB database is employed for its flexibility with JSON-like documents. Here is the database structure:

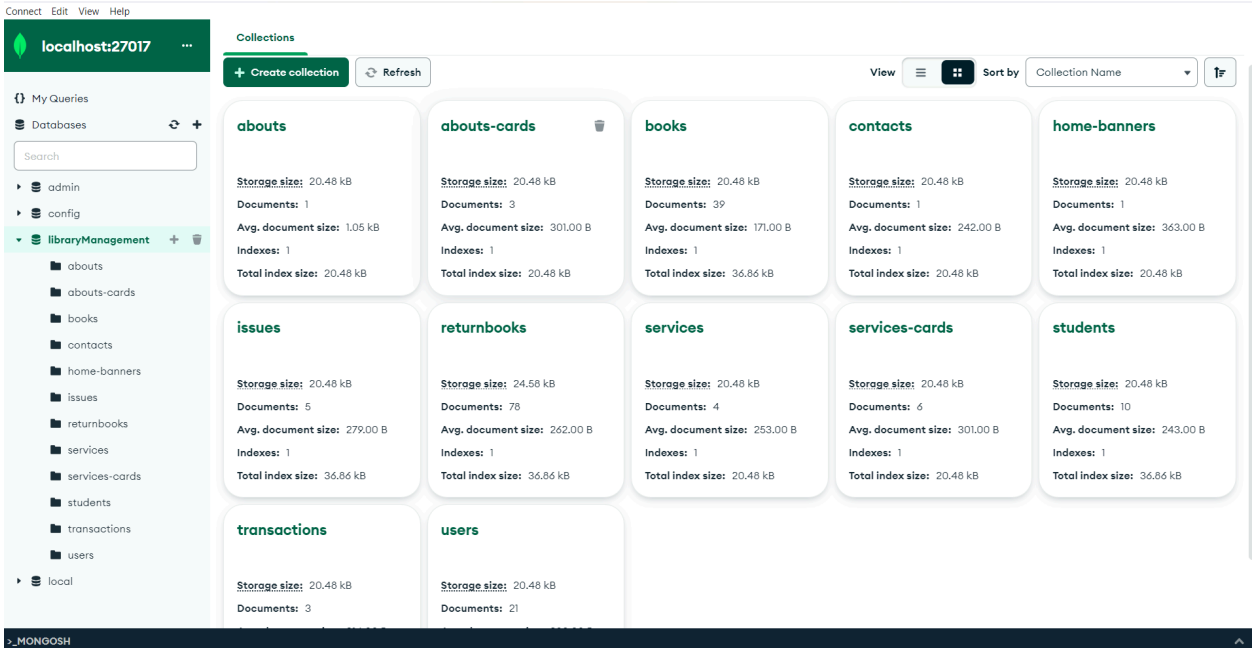

Figure 5.1.1: All database table of Library Management System

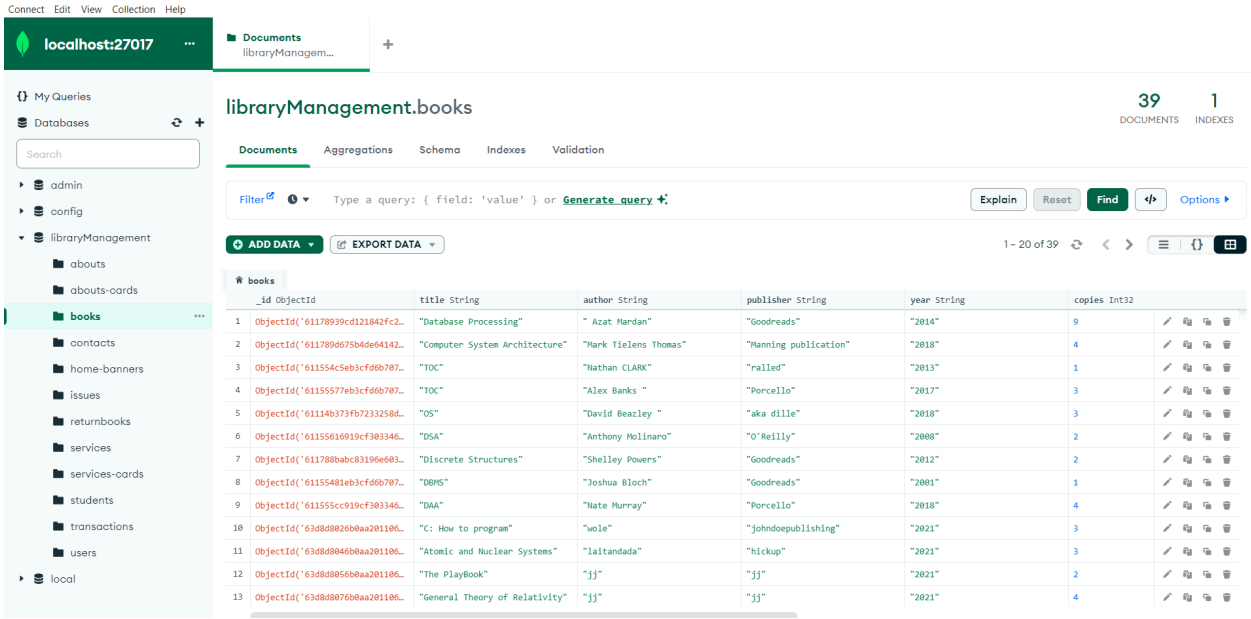

Figure 5.1.2: Book list table of Library Management System

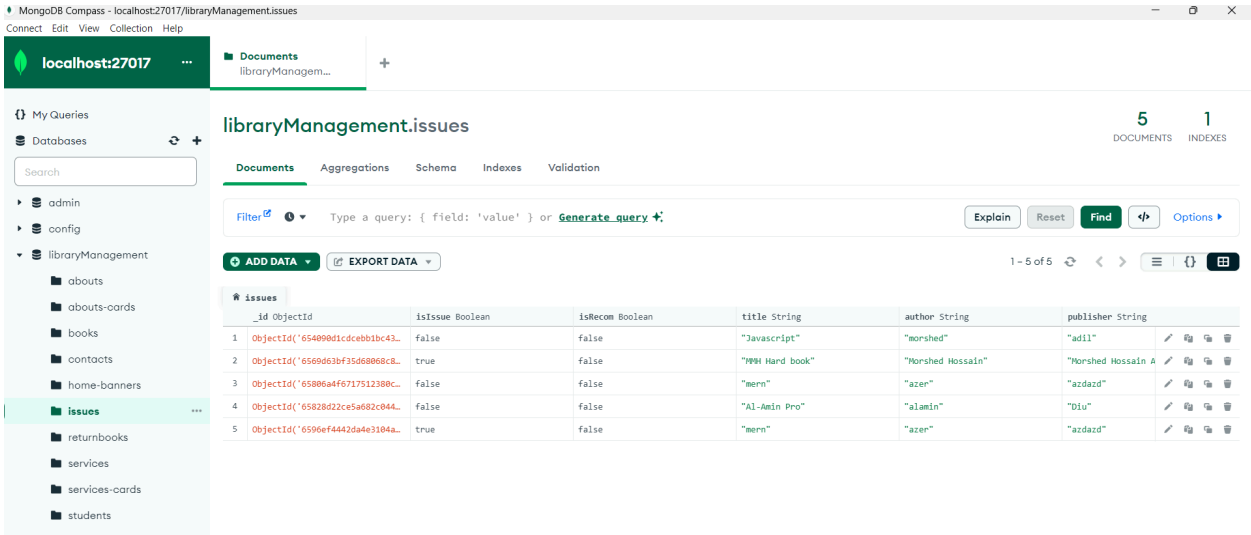

Figure 5.1.3: Book issue record table of Library Management System

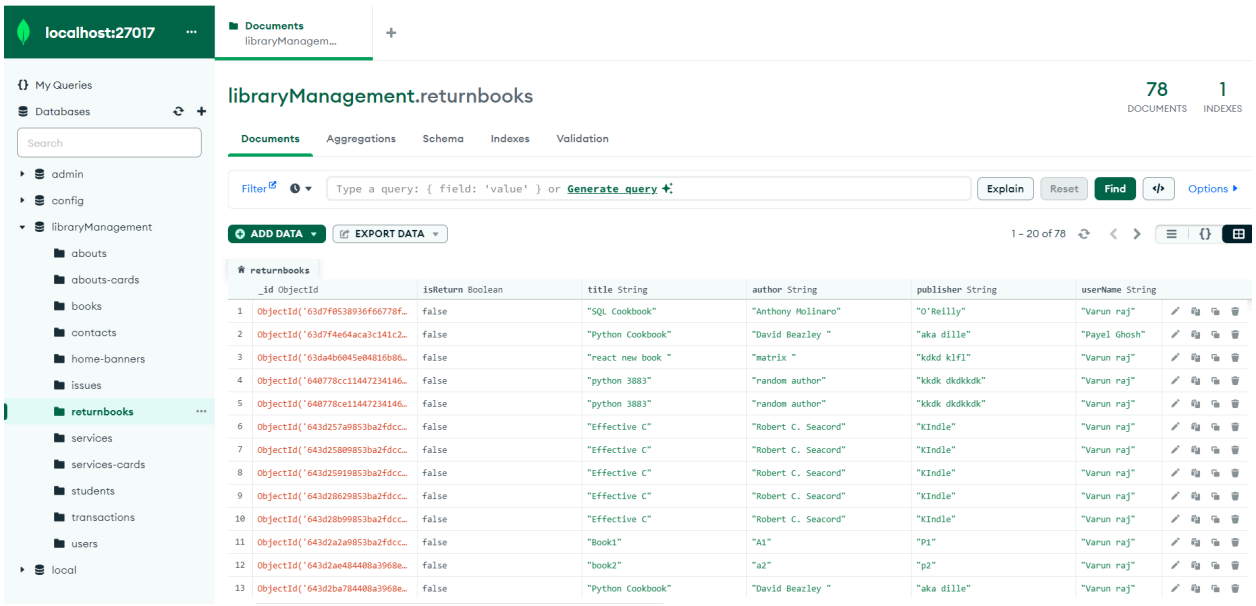

Figure 5.1.4: Return book record table of Library Management System

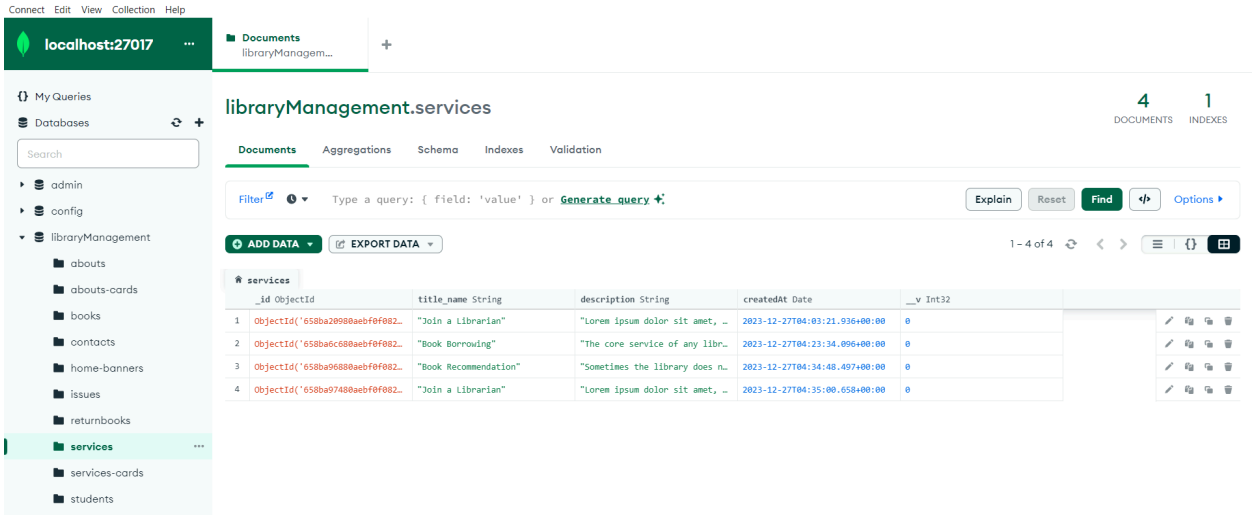

Figure 5.1.5: Service table of Library Management System

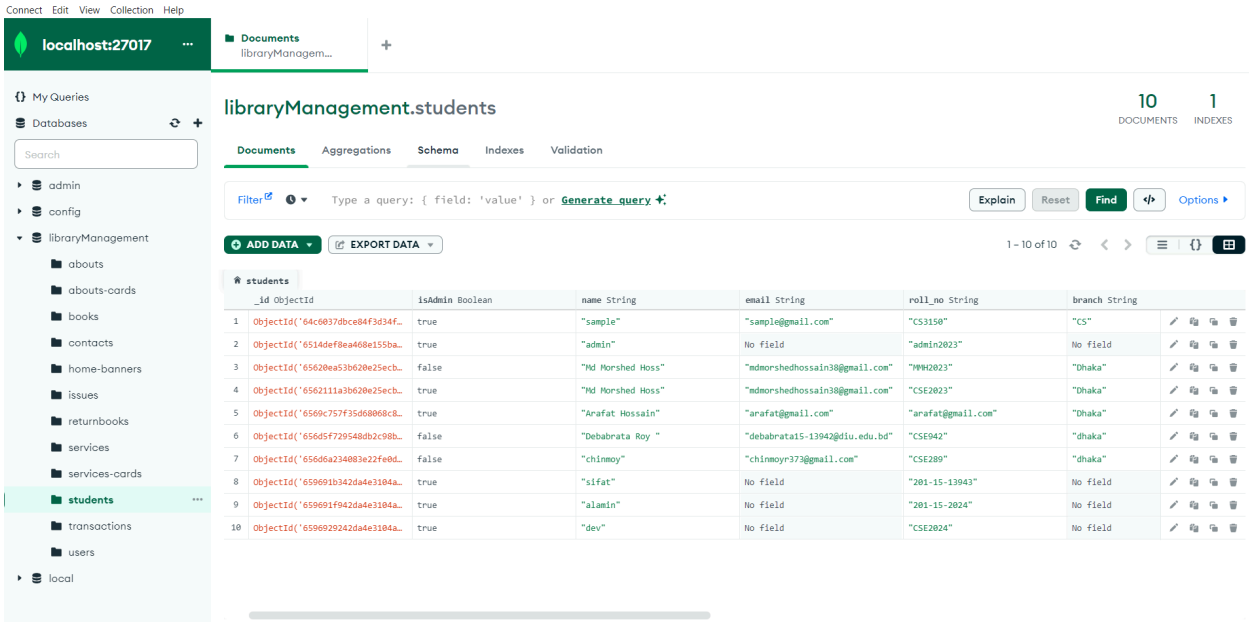

Figure 5.1.6: Student table of Library Management System

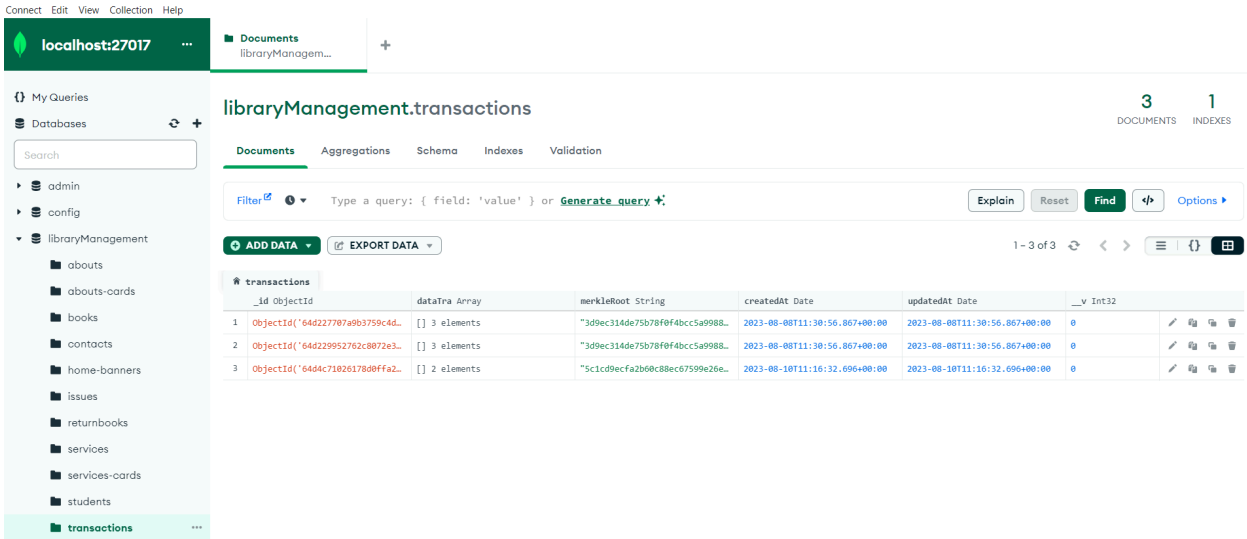

Figure 5.1.7: Transaction table of Library Management System

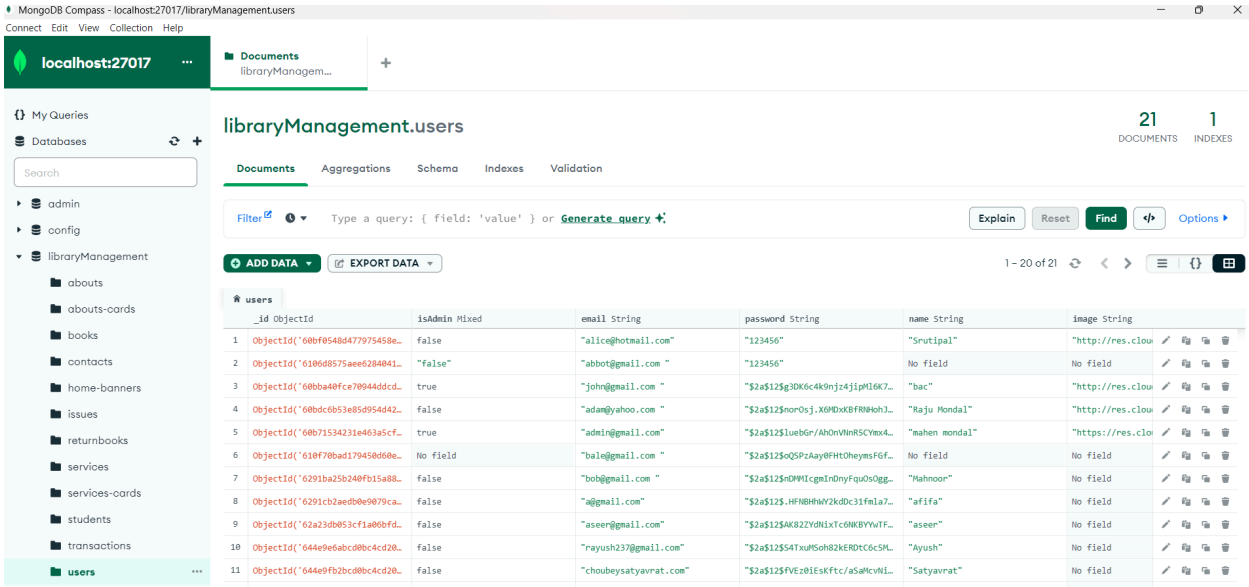

Figure 5.1.8: Users table of Library Management System

5.2 Implementation of Front-end Design

The frontend implementation focuses on creating a user-friendly, responsive, and efficient interface for both students and librarians. The design aims to enhance the overall user experience and streamline the various functionalities offered by the library management system.

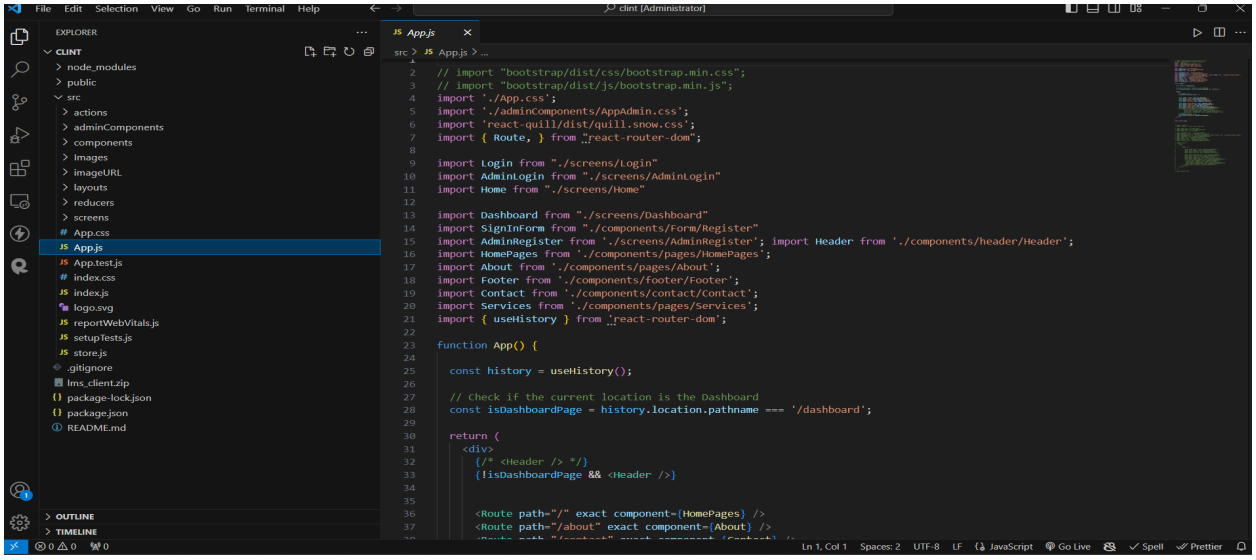

Figure 5.2.1: Some code of front-end using react.js framework

```
.bg__home__banner {
  position: relative;
  background: rgba(38, 38, 38, 0.65);
}

.bg__home__banner .banner__video {
  position: relative;
  top: 0;
  left: 0;
  width: 100%;
  height: 500px;
  object-fit: cover;
  z-index: -1;
}

.home__banner__content {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.banner__title {
  text-align: center;
  color: #fff;
  font-weight: 600;
  text-transform: uppercase;
  font-family: 'Montserrat';
  font-size: 28px;
}

.banner__desc {
  color: #fff;
  font-size: 13px;
```

Figure 5.2.2: Some code of CSS for front-end design

```
import React from "react";
import Sidebar from "../components/Sidebar"
import { Switch, Route, Link } from "react-router-dom";
import AddBook from "./AddBook"
import AllBook from "./AllBook"
import AllStudent from "./AllStudent"
import IssueRequest from "./IssueRequest"
import UserHome from "../components/UserHome"
import UserIssuedBook from "./UserIssuedBook"
import Recom_Book from "./Recom_Book"
import RecomdationBook from "./Recommdation"
import Messages from "./Messages"
import AllIssuedBook from "./AllissuedBook"
import Navbar from "../components/Navbar"
import IssueReturn from "./Issue_Return";
import ReturnBook from "./ReturnBook"
import Addemployee from "./Add_Employee"
import ContactSetup from '../adminComponents/contactSetup/ContactSetup';
import EditContactSetup from '../adminComponents/contactSetup/EditContactSetup';
import AboutSetup from '../adminComponents/aboutSetup/AboutSetup';
import AddAboutSetup from '../adminComponents/aboutSetup/AddAboutSetup';
import EditAboutSetup from '../adminComponents/aboutSetup/EditAboutSetup';
import AboutCardSetup from '../adminComponents/aboutCardSetup/AboutCardSetup';
import AddAboutCardSetup from '../adminComponents/aboutCardSetup/AddAboutCardSetup';
import EditAboutCardSetup from '../adminComponents/aboutCardSetup/EditAboutCardSetup';
import ServicesCardSetup from '../adminComponents/servicesCardSetup/ServicesCardSetup';
import AddServicesCardSetup from '../adminComponents/servicesCardSetup/AddServicesCardSetup';
import EditServicesCardSetup from '../adminComponents/servicesCardSetup/EditServicesCardSetup';
import ServicesSetup from '../adminComponents/servicesSetup/ServicesSetup';
import AddServicesSetup from '../adminComponents/servicesSetup/AddServicesSetup';
import EditServicesSetup from '../adminComponents/servicesSetup/EditServicesSetup';
import HomeBannerSetup from '../adminComponents/homeBannerSetup/HomeBannerSetup';
import AddHomeBannerSetup from '../adminComponents/homeBannerSetup/AddHomeBannerSetup';
import EditHomeBannerSetup from '../adminComponents/homeBannerSetup/EditHomeBannerSetup';

const Dashboard = () => {
    return (
```

Figure 5.2.3: Some code of dashboard using react.js framework

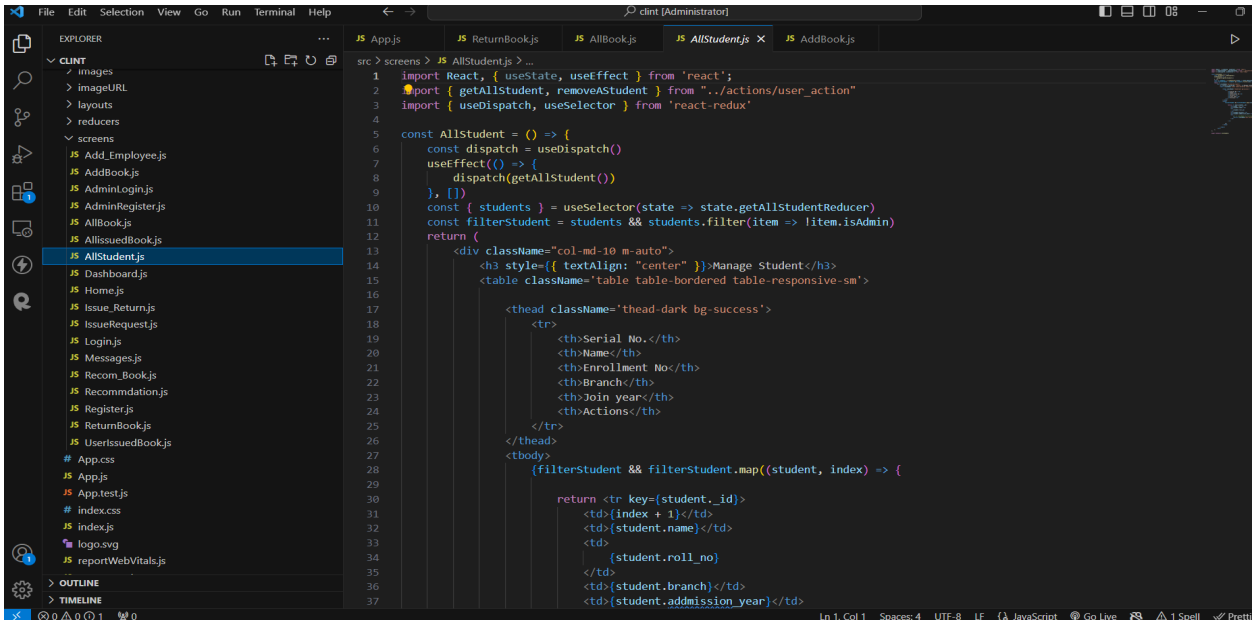

Figure 5.2.4: Some code of all student using react.js framework

5.3 Testing Implementation

Test Type	Description	Expected Outcome
Unit Testing	Testing individual functions and methods in isolation.	Successful execution of each function with expected results.
Integration Testing	Testing interactions between different components.	Seamless communication between integrated components.
End-to-End Testing	Testing the entire application flow from start to finish.	Successful completion of all steps in the workflow without errors.

Login Functionality	Verifying the login process for students and librarians.	Users can log in successfully with valid credentials.
Dashboard Accuracy	Testing the accuracy of information displayed on the dashboard.	Dashboard reflects correct and up-to-date data for each user.
Book Addition	Testing the ability to add new books to the system.	New books are added to the catalog with accurate details.
Book Deletion	Testing the ability to remove books from the catalog.	Books are successfully deleted without any system errors.
Student Registration	Verifying the process of registering new students.	New student records are created accurately with the provided information.
Student Update	Testing the ability to update student information.	Changes to student details are saved correctly and reflected in the system.
Automatic Fines	Testing the automatic calculation of fines for overdue books.	Fines are accurately calculated based on due dates.

Responsive Design	Testing the application's responsiveness across devices.	Consistent and user-friendly experience on desktop, tablet, and mobile.
Security Testing	Testing the application for potential security vulnerabilities.	No security vulnerabilities found; sensitive data is handled securely.
Performance Testing	Testing the application's performance under various loads.	System response times meet acceptable standards under stress conditions.

5.4 Test Result and Report

Test Type	Description	Expected Outcome	Test Result	Issues Identified	Resolution/Comments

Unit Testing	Testing individual functions and methods in isolation.	Successful execution of each function with expected results.	Passed	None	N/A
Login Functionality	Verifying the login process for students and librarians.	Users can log in successfully with valid credentials.	Passed	None	N/A
Dashboard Accuracy	Testing the accuracy of information displayed on the dashboard.	Dashboard reflects correct and up-to-date data for each user.	Passed	None	N/A
Book Addition	Testing the ability to add new books to the system.	New books are added to the catalog with accurate details.	Passed	None	N/A

Book Deletion	Testing the ability to remove books from the catalog.	Books are successfully deleted without any system errors.	Passed	None	N/A
Student Registration	Verifying the process of registering new students.	New student records are created accurately with the provided information.	Passed	None	N/A
Student Update	Testing the ability to update student information.	Changes to student details are saved correctly and reflected in the system.	Passed	None	N/A
Transaction Processing	Testing the issuance, renewal, and return of books by students.	Book transactions are processed accurately with proper updates.	Passed	None	N/A

Automatic Fines	Testing the automatic calculation of fines for overdue books.	Fines are accurately calculated based on due dates.	Passed	None	N/A
Responsive Design	Testing the application's responsivenes s across devices.	Consistent and user-friendly experience on desktop, tablet, and mobile.	Passed	None	N/A
Security Testing	Testing the application for potential security vulnerabilities .	No security vulnerabilities found; sensitive data is handled securely.	Passed	None	N/A
Performance Testing	Testing the application's performance under various loads.	System response times meet acceptable standards under stress conditions.	Passed	None	N/A

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Society

The MERN stack library management system can significantly improve access to knowledge, resource management, administrative processes, and information. It facilitates easy access to a wide range of books and educational resources, fostering a culture of continuous learning and academic development. The system also streamlines administrative tasks, allowing librarians to focus on quality services. Real-time updates on book availability, due dates, and fines promote responsible borrowing behavior. The system encourages digital literacy among students and staff, enhancing technological skills. The user experience is enhanced by a user-friendly interface and responsive design, making library services accessible to individuals with diverse technology backgrounds. The system creates optimized learning environments, contributing to intellectual growth. Furthermore, the MERN stack system prepares students and staff for future technological trends, aligning with the evolving needs of the digital era.

6.2 Impact on Environment

The MERN stack library management system project has positive environmental impacts, including reduced paper consumption, energy efficiency, and a minimized carbon footprint. It eliminates the need for paper-based records and transactions, conserving trees and reducing the environmental impact of paper production. The system also offers cloud-based hosting solutions, which can be more energy-efficient than on-premises servers, reducing the carbon footprint associated with hosting and server maintenance. Remote access to library resources reduces the need for physical travel, promoting a more sustainable approach to education. The digitization of library resources promotes long-term resource efficiency, eliminating the need for physical printing and distribution, reducing the environmental impact of printed materials. The transition to a digital and paperless environment helps mitigate the environmental impact of waste disposal and recycling processes. Adopting a digital library system aligns with broader trends in sustainable and eco-friendly practices, encouraging other educational institutions to adopt digital solutions.

6.3 Ethical Aspects

The MERN stack library management system requires ethical considerations for user privacy, data security, fair access, transparency in fines and penalties, user consent, accuracy and integrity of information, responsibility in system design, sustainability and environmental impact, and intellectual property rights. These aspects include implementing robust security measures, providing fair and inclusive access, ensuring transparency in fines and penalties, ensuring user consent, ensuring accurate and reliable information, avoiding biases in algorithms, minimizing the environmental impact, and adhering to copyright laws and licensing agreements. By addressing these ethical aspects, the system can ensure a user-friendly and user-friendly experience, reducing potential biases and promoting ethical practices in the library management system.

6.4 Sustainability Plan

The project involves several key aspects, including user privacy, environmental impact, adaptability, ethical considerations, cost management, intellectual property compliance, continuous improvement, disaster recovery and contingency planning, legal and regulatory compliance, and a dynamic sustainability plan. These aspects are crucial for ensuring the project's success and minimizing its environmental impact. Regular security assessments, encryption protocols, and user data anonymization are essential. The system should be adaptable to technological advancements and scalable, with regular updates to technology stacks and modular design principles. Ethical considerations should be integrated into the system's governance, and the project should be cost-effective, ensuring financial sustainability. A culture of continuous improvement and legal and regulatory compliance is also essential. The sustainability plan should be dynamic, evolving with the project and addressing changing circumstances, technological advancements, and stakeholder feedback.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion:

The library management system using the MERN stack has revolutionized library operations and user experiences. It streamlines processes like book transactions, student management, and staff administration, enhancing efficiency and automation. The system's user-friendly interface and responsive design create a positive learning environment. The MERN stack represents a commitment to leveraging modern technologies, ensuring its relevance and adaptability to future trends. Data security and privacy are prioritized, with robust measures in place to safeguard user data. The project has successfully embraced technological innovation, modernizing library operations and enhancing the overall educational experience. The positive impacts include improved access to knowledge, efficient resource management, and promotion of digital literacy. Ethical considerations, such as user privacy, fairness, and transparency, have been carefully addressed, ensuring responsible and accountable system operation. To maintain the system's long-term success, it is crucial to maintain a dynamic sustainability plan, adapting to technological changes and evolving user needs. Continuous engagement with the user community, adherence to ethical guidelines, and regular security updates will contribute to the system's long-term success. This project represents the intersection of technology, education, and ethical considerations in creating a more efficient and user-centric learning environment.

7.2 Scope for Further Development

The library management system using the MERN stack has been successfully implemented, laying the groundwork for future enhancements. The project's scope can be expanded to meet technological trends and user needs. Potential areas for further development include mobile application development, advanced analytics and reporting, integration with Learning Management Systems (LMS), augmented reality features, enhanced collaboration tools, voice-activated commands, and expanding resource types. Develop dedicated mobile applications for Android and iOS platforms to enhance accessibility and manage library activities. Implement advanced analytics tools to gather insights into library usage patterns, popular resources, and user behaviors. Explore integration possibilities with existing LMS platforms to provide a holistic educational experience for students. Additionally, develop augmented reality features to enhance the physical library experience, such as interactive navigation and virtual tours. Enhance collaboration tools to enable students and staff to share resources, create study groups, and engage in research. Implement voice-activated commands to enhance accessibility and support a broader range of resource types, such as multimedia resources, digital archives, and online courses.

REFERENCES

[1] Learn about Wikipedia, available at << http:// www.tutorialrepublic.com/twitter-bootstrap-tutorial/>>, last accessed on 06-06-2020 at 10:00 PM.

[2] Learn about Express.js, available at << http:// www. Express.js //>>, last accessed on 20-06-2020 at 10:00 PM.

[3] Learn about HTML, CSS, JS, available at << http:// www.w3schools.com//>>, last accessed on 06-06-2020 at 7:00 PM.

[4] Learn about Node.js, available at << http:// www. nodejs.org/en/about//>>, last accessed on 10-07-2021 at 9:00 PM.

[5] Learn about React.js, available at << http:// www. react.dev/reference/react //>>, last accessed on 16-12-2021 at 10:00 AM.

[6] Learn about MongoDB, available at << http:// www.mongodb.com/docs/guides //>>, last accessed on 10-4-2022 at 10:00 PM.

[7] Learn about Koha, available at <<https://koha-community.org///>>, last accessed on 06-06-2023 at 10:00 PM.

[8] Learn about DIU Library, available at <<https://library.daffodilvarsity.edu.bd/>>, last accessed on 10-06-2023 at 10:00 PM.

[9] Learn about DU Library, available at <<https://www.du.ac.bd/index.php/offices/LIB>>, last accessed on 20-06-2023 at 10:00 PM.

[10] Learn about Amazon books, available at <<https://www.amazon.com/books-used-books-textbooks/>>, last accessed on 1-07-2023 at 10:00 PM.

[11] Learn about Visual Studio, available at <<https://code.visualstudio.com//>>, last accessed on 11-08-2023 at 10:00 PM.

Turnitin Originality Report

14/07/24

Processed on: 29-Jun-2024 13:32 +06
ID: 2410164699
Word Count: 9141
Submitted: 1

A WEB BASED LIBRARY MANAGEMENT SYSTEM By Debabrata Roy Chinmoy 201-15-13942

Similarity Index	Similarity by Source	
21%	Internet Sources:	19%
	Publications:	3%
	Student Papers:	14%

3% match (Internet from 10-Nov-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/10604/23374.pdf?isAllowed=y&sequence=1

2% match (student papers from 21-Jan-2024)
Submitted to Daffodil International University on 2024-01-21

1% match (Internet from 26-Oct-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8253/211-25-929%20%2825%25%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 26-Oct-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8209/181-15-1868%288%25%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 14-Feb-2022)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7129/171-15-9197%20%2817%25%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 21-Jul-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/10027/22662.pdf?isAllowed=y&sequence=1

1% match (Internet from 03-Oct-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4391/161-15-707.161-15-684.pdf?isAllowed=y&sequence=1

1% match (student papers from 28-Jun-2024)
Submitted to Daffodil International University on 2024-06-28

1% match (student papers from 09-Jul-2023)
Submitted to Daffodil International University on 2023-07-09

1% match (Chen, Xue, Dongdai Zhou, and Liuguo Wu. "Theory of automatic transformation from Computation Independent Model to Platform Independent Model for MDA", Engineering Technology and Applications, 2014.)
Chen, Xue, Dongdai Zhou, and Liuguo Wu. "Theory of automatic transformation from Computation Independent Model to Platform Independent Model for MDA", Engineering Technology and Applications, 2014.

< 1% match (Internet from 06-May-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/9099/21381.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 05-Aug-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/10497/23156.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 28-Jun-2024)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/12742/26571.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 22-Jul-2023)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/10345/23045.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 26-Oct-2022)