Tribhuvan University

Faculty of Humanities and Social Sciences

LIBRARY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted to

Department of Computer Application

National College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Amit KC - 04

July - 2023

Under the Supervision of

Anil Pandey

Tribhuvan University

Faculty of Humanities and Social Sciences

National College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Amit KC entitled "Library Management System" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Anil Pandey

Humanities and Social Science

National College

Lainchaur, Kathmandu

ABSTRACT

The Library Management System is a comprehensive solution aimed at modernizing and streamlining various aspects of library operations. Users benefit from features like seamless book browsing, requesting, and monitoring of issued books, ensuring a user-friendly experience. Concurrently, administrators can efficiently manage the book inventory, add new titles, and oversee user requests, empowering them with robust control over the library's resources. The system's primary goal is to enhance operational efficiency, alleviate manual workloads, and optimize user satisfaction within the library setting.

Additionally, the Library Management System enhances user experience by incorporating content-based filtering algorithms for personalized book recommendations based on individual preferences. This innovative feature provides book suggestions to each user, creating a more engaging and joyful reading experience. The system further reinforces responsible borrowing behavior through the introduction of a 10-day return period, emphasizing the importance of timely book returns. Overall, the Library Management System stands out as a valuable tool in promoting user-centricity and operational efficiency within the library setting.

Tribhuvan University

Faculty of Humanities and Social Sciences

National College

LETTER OF APPROVAL

This is to certify that this project prepared by Amit KC entitled "**Library Management System**" in partial fulfillment of the requirements for the degree of Bachelor's in computer application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

SIGNATURE of Supervisor	**SIGNATURE of HOD/Co-Ordinator**
Anil Pandey Humanities and Social Science National College Lainchaur, Kathmandu	Navaraj Heka Humanities and Social Science National College Lainchaur, Kathmandu

ACKNOWLEDGEMENT

We are very thankful to the National College for supporting and giving a platform for the project. We also thank our respected sir Nav Raj Heka (Academic Director) for supporting and managing us in our project. We would also thank all our respected teachers who supported this project. We also give thanks to all the expertise present and helped with our project. We also like to show our deep concentration towards all the teachers in National College for providing us with a favorable environment and providing with us great opportunity to learn about various kinds of IT equipment and tools and their working principle, design, control system related to our subject and permission to be familiar with the devices and equipment. We would also like to express our heart full thanks to all staffs of all departments and the staffs of Administration and site organization and thanks to all several working staff on these subject and special thanks for providing us with facilitated guidance during the project. Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Introduction

A Library Management System is a software application designed to revolutionize traditional library operations. This project aims to enhance the user experience by providing seamless book browsing, requesting, and monitoring features while, Administrators can manage the library's collection, efficiently managing the book inventory and overseeing user requests. The primary goal is to boost operational efficiency, reduce manual workloads, and ensure user satisfaction. The system goes beyond conventional library management, incorporating content-based filtering algorithms to offer personalized book recommendations based on individual preferences. Additionally, it promotes responsible borrowing behavior with a 10-day return period. The key is to put users first and make sure everything runs smoothly, creating a modern and accessible library space for everyone.

1.2 Problem Statement

In many traditional library settings, the reliance on manual processes and paperwork poses significant challenges, consuming valuable time and introducing the risk of errors. The intricate task of tracking borrowed books and effectively managing the library's collection amplifies these challenges. This project addresses these issues by aiming to create a user-friendly and efficient library system. The objective is to establish a seamless experience where users can effortlessly explore and borrow books, while administrators navigate library management with ease. Additionally, we've observed users forgetting to return books, prompting the introduction of a reminder system. Our ultimate goal is to transform the library into a hassle-free and enjoyable space for everyone, ensuring a more streamlined and error-resistant approach to library operations.

1.3 Objectives

The main objectives of the Library Management System are listed below:

- Enhance library efficiency by automating cataloging, issuance, and inventory control.
- Reduce manual workload for administrators.
- Improve user experience by simplifying book browsing and implementing personalized recommendations.
- Streamline administrative tasks, reducing manual efforts, and promote responsible borrowing and timely returns.

1.4 Scope and Limitation

1.4.1 Scope

The Scope of this project will be limited to the following:

- Users can access a user-friendly interface to browse, request, and monitor books in the library.
- The system facilitates efficient management for administrators, allowing them to handle book inventory, user requests, and new book additions.
- Users will receive personalized book recommendations through content-based filtering algorithms.
- The system incorporates a 10-day return period to encourage responsible borrowing behavior.

1.4.2 Limitation

While the Library Management System strives to enhance user experience, it is important to note that the system's effectiveness relies on network connectivity. Users without internet access may face limitations in utilizing certain features, such as real-time book recommendations and updates.

1.5 Development Methodology

We developed this system for the library based on the waterfall model. Regardless of this, it was necessary to mention and consider certain models including the following generic software development models:

- Agile Model: Agile is an iterative and incremental software development approach that prioritizes collaboration and adaptability. It emphasizes delivering functional software in short cycles called sprints. Agile development welcomes changing requirements, encourages customer feedback, and focuses on continuous improvement. Scrum and Kanban are popular Agile frameworks.
- Spiral Model: The Spiral Model combines elements of the Waterfall Model and iterative development. It involves multiple iterations, each building on the previous one, and incorporates risk assessment and mitigation at each stage. The model is ideal for large, complex projects where risks need constant evaluation, but it may be resource-intensive and time-consuming.
- V-Model (Verification and Validation Model): The V-Model is an extension of the Waterfall Model, where each development stage has a corresponding testing phase. It emphasizes the importance of testing at every step to ensure high-quality software. The model ensures that software requirements are thoroughly validated and that the final product meets the expected quality standards.
- Incremental Model: The Incremental Model divides the development process into smaller, manageable increments. Each increment represents a functional part of the software. Developers build and deliver these increments in successive iterations. This approach allows for faster delivery of essential functionalities, but it may require careful planning to integrate different increments effectively.
- RAD (Rapid Application Development): RAD is a fast-paced, iterative model that focuses on prototyping and quick feedback. It is well-suited for projects with rapidly changing requirements. RAD emphasizes user involvement and produces functional prototypes quickly to gather user feedback and refine the software incrementally.

The Waterfall Methodology for software development

The Library and Recommendation System will be built based on the waterfall model, which is a traditional project management approach that follows a sequential, linear progression through distinct phases of a project. The model is suitable for projects with well-defined requirements and minimal changes.

Phases of Waterfall Methodology:

- Requirements Gathering: The project requirements are identified and documented in detail.
- System Design: The system architecture, modules, and components are designed based on the requirements.
- Implementation: The system is developed and implemented according to the design specifications.
- Testing: The system is thoroughly tested to ensure its functionality, reliability, and adherence to requirements.
- Deployment: The fully tested system is deployed and made operational in the production environment.
- Maintenance: Ongoing maintenance and support activities are carried out to address issues and implement enhancements.

Why Waterfall Methodology?

- Clear Structure: The waterfall methodology provides a clear and well-defined structure for project execution, with distinct phases and deliverables. This ensures that each phase is completed before moving on to the next, promoting a systematic and organized approach.
- Predictability: The linear nature of the waterfall methodology allows for better predictability in terms of project timelines, costs, and scope. With requirements being defined upfront, there is greater clarity and visibility into the project's progress and outcomes.
- Documentation Emphasis: The waterfall methodology emphasizes documentation throughout the project lifecycle. Each phase generates documentation, including requirements specifications, design documents, test plans, and user manuals. This documentation aids in knowledge transfer, future maintenance, and traceability of decisions made during the project.
- Early Identification of Risks: Since the requirements are defined at the beginning, the waterfall methodology facilitates early identification of potential risks and challenges. This allows for better risk management and mitigation strategies to be implemented during the subsequent phases.
- Well-suited for Stable Requirements: The waterfall methodology is particularly suitable when project requirements are stable and well-understood. It works best when the scope is well-defined and unlikely to change significantly during the project.
- Client Visibility and Approval: The waterfall methodology encourages client involvement and provides opportunities for regular review and approval at the end of each phase. This allows clients to have visibility into the progress and provide feedback before proceeding to the next phase.
- Comprehensive Testing: The waterfall methodology incorporates testing as a dedicated phase, ensuring that the system is thoroughly tested before deployment. This helps identify defects and issues early in the project lifecycle, reducing the risk of major problems emerging at later stages.
- Well-suited for Large-Scale Projects: The waterfall methodology is often considered suitable for large-scale projects with complex requirements, as it provides a structured approach for managing and coordinating different aspects of the project.

This software methodology has been illustrated below in the figure:

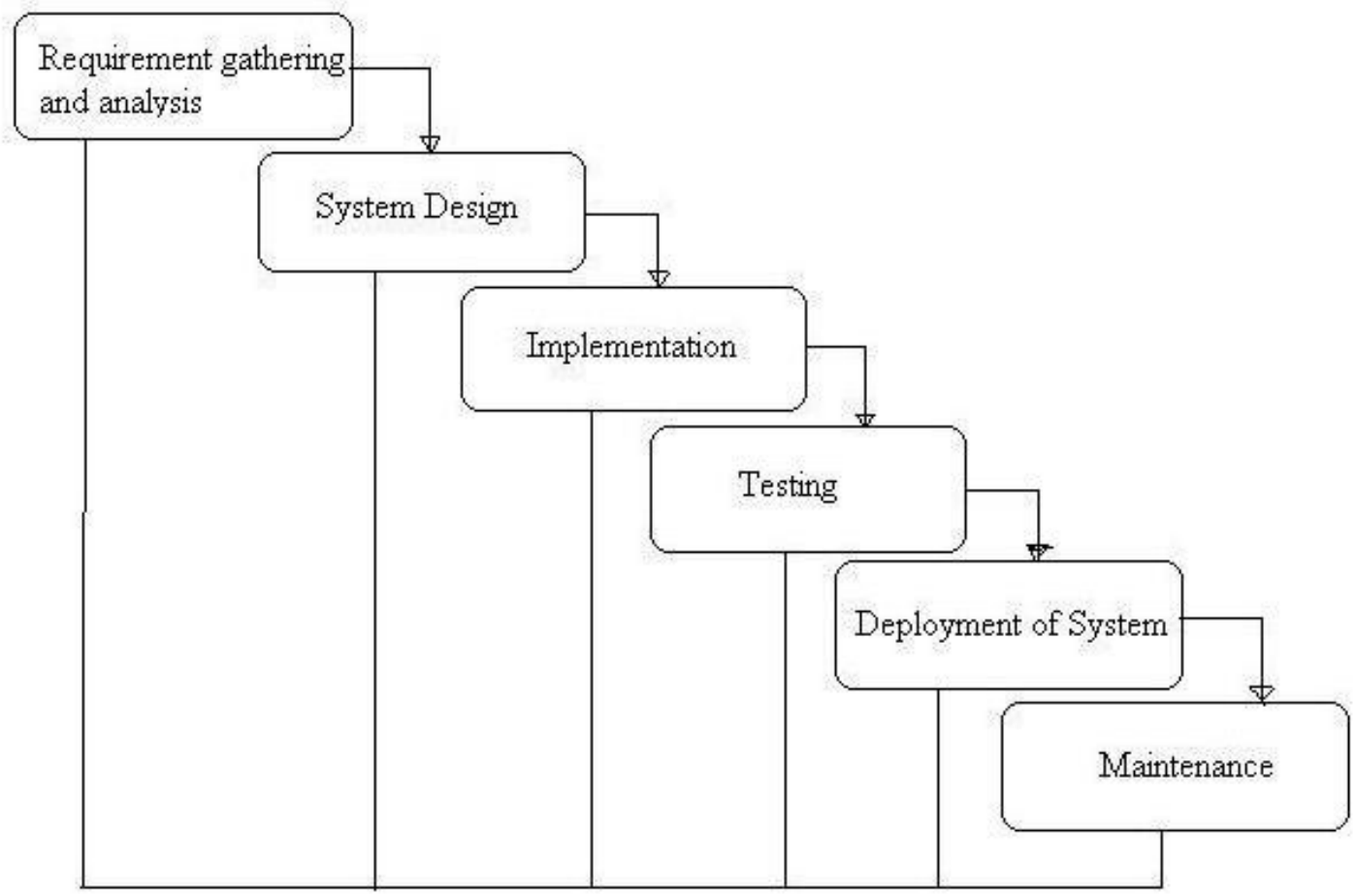


Figure 1 Waterfall Model Design

1.6 Report Organization

The structure of this report is designed to provide a logical and coherent flow of information for readers.

Chapter 1 Introduction: Highlights

Shows the background and overview, problem statement, objectives, scope and limitations of the project. Problem statement highlights the project for which the project was developed. This section is theoretical, and research based. The project objectives, scope and limitations are discussed.

Chapter 2 Background Study and Literature Review: Highlights

Defines and describes Background Study and literature review about existing systems along with their advantages and disadvantages.

Chapter 3 System Analysis and Design: Highlights

Presents the System Analysis and Design including Requirement Analysis and Feasibility Analysis. Requirement Analysis explains the Functional and Non-Functional Requirements of the project and Feasibility Analysis explains why/how the project is implemented.

Chapter 4 Implementation and Testing: Highlights

Presents the Implementation and Testing, it clarifies the system workflow, and provides an indication of how the system is implemented, and what tools/platforms have been used.

Chapter 5 Conclusion and Future Recommendations: Highlights

Presents the Conclusion and Recommendation, it marks an end to the documents by summing up the entire project and by opening a door for future research mentioning future enhancements that can be carried out on the system. This chapter summarizes the findings, discusses the project's implications, and suggests areas for future enhancements.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

A library is an organized collection of information sources that is made accessible to the people. The library usually contains the information physically or in a digitized format. In the olden period, access was usually in the library rooms as technology grew up access was made online. Michael Stern Hert is the founder of the first project for digital library-Project Gutenberg (established in 1971, was the world's first digital library) which makes electronic books freely available via the internet. This was originally set up to make 10,000 of the most consulted books available to the public at little or no cost. The library is a fast-growing organism. The ancient methods to maintain it are no longer dynamic and efficient. Since most of the system is digitized and online users can easily use the library features through internet access. Libraries can be divided into categories by several types, which are academic libraries, cooperate libraries, government libraries, private libraries, historical society libraries, digital libraries, slide libraries, architecture libraries, fine arts libraries medical libraries, etc.

Before starting the development on the Library Management System project, a comprehensive study of essential concepts, and project-specific terminologies was conducted such as:

- Cataloging, where books are systematically organized, and the logistics of book lending and returns
- User authentication to ensures secure access controls, while focusing on the efficient organization and maintenance of library resources
- Content-based filtering algorithms to offer personalized book recommendations based on user preferences.

This background study aimed to establish a solid understanding of the principles that influenced the project's design and development.

2.2 Literature Review

2.2.1 Introduction

A Library Management System (LMS) is a comprehensive software solution designed to automate and streamline various library operations. The system aims to enhance the efficiency and accessibility of library services, including cataloging, circulation, patron management, and inventory control. This literature review delves into the research and existing literature related to LMS to understand its evolution, key features, benefits, challenges, and future trends. In this section we are going to discuss the fundamentals of a similar existing system. Here we are going to compare ourselves with "Koha" which is one of the most popular Library Management Systems. Koha is the world's first free and open-source library system. The literature on Koha emphasizes its transformative journey from manual processes to a computerized library management system, including the incorporation of online reservation platforms. Scholars highlight numerous benefits, such as heightened efficiency, improved user services, refined resource management, and expanded accessibility for library users. Research explores user behaviors in online reservations, focusing on factors like book availability, convenience, user reviews, and trust. Crucially, user experience and interface design are pivotal for system success, with studies scrutinizing website design, usability, and personalization. The Koha system is intricately designed to assist library administrators in optimizing operations and elevating the overall user experience.

Some of the key features of Koha include:

1. **Cataloging and Reservations Management:** Koha empowers library staff to efficiently manage cataloging processes and handle reservations in real-time.
2. **Inventory Management:** Koha allows library staff to oversee inventory and supplies in real-time, ensuring that the library is well-stocked and ready to meet the needs of users.
3. **Borrowing and Invoicing:** The system provides automated borrowing and invoicing capabilities, streamlining processes for library staff to generate invoices and manage borrowing transactions.
4. **User Management:** User-friendly interface for managing users information, registrations, and user activities.
5. **Multi-Lingual Support:** Koha is Capable of supporting multiple languages for a diverse user base.

2.2.2 Evolution of Library Management Systems:

The concept of library management systems traces its roots back to the early 20th century when manual card catalog systems were used. As technology advanced, these manual systems were gradually replaced by computer-based systems in the late 20th century. The transition from stand-alone systems to web-based LMS has been transformative, enabling libraries to provide online services and access to a broader audience.

2.2.3 Key Features of Library Management Systems:

Various research studies have highlighted the essential features of an effective LMS, including:

- Cataloging and Metadata Management: The ability to efficiently manage and search through library resources using standardized metadata.
- Circulation Management: Facilitating check-in, check-out, and renewal processes, as well as handling overdue fines and holds.
- Patron Management: Maintaining patron records, tracking borrowing history, and providing personalized services.
- Online Public Access Catalog (OPAC): Allowing users to search and request resources online.
- Reporting and Analytics: Generating comprehensive reports to assess library usage and resource availability.

2.2.4 Benefits of Implementing Library Management Systems:

Research has consistently shown that the adoption of LMS provides numerous benefits to libraries, including:

- Improved Efficiency: Automation reduces manual tasks, leading to faster and more accurate library operations.
- Enhanced User Experience: Online access to resources and the ability to place requests remotely enhances patron satisfaction.
- Better Resource Management: LMS enables libraries to optimize collection management and resource allocation.
- Data-Driven Decision Making: Robust reporting features help library administrators make informed decisions.

2.2.5 Challenges in Implementing Library Management Systems:

While LMS offers significant advantages, its implementation may face certain challenges, such as:

- Initial Costs: Acquiring and setting up an LMS can be a substantial investment for libraries.
- Staff Training: Adequate training is essential to ensure staff can effectively utilize the system's features.
- Data Migration: Transferring data from legacy systems to the new LMS can be complex and time-consuming.
- User Acceptance: Resistance to change among library staff and users might hinder smooth adoption.

2.2.6 Future Trends in Library Management Systems:

Looking ahead, several trends are shaping the evolution of LMS:

- Integration of Artificial Intelligence (AI): AI-powered features like personalized recommendations and predictive analytics to enhance user experience.
- Mobile Accessibility: LMS will increasingly cater to mobile users through responsive interfaces and mobile apps.
- Linked Data and Semantic Web: Integration of linked data principles to improve resource discoverability and interoperability.
- Open Access and Collaboration: LMS will facilitate collaboration among libraries and open access initiatives for enhanced resource sharing.

2.2.7 Conclusion:

The literature reviewed here indicates that Library Management Systems have come a long way, revolutionizing library operations and services. By understanding the evolution, key features, benefits, challenges, and future trends of LMS, libraries can make informed decisions when selecting, implementing, and upgrading their systems.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

The system analysis phase serves as the bedrock for successful software development, providing a systematic approach to understanding, planning, and defining the scope of the Library Management System project. Adopting the Waterfall Model as our chosen methodology, the system analysis phase unfolds as a structured, sequential process, emphasizing a meticulous progression through each stage. This phase commences with a thorough exploration of existing library management systems, seeking inspiration and identifying unique features that can enhance the user experience and operational efficiency. The Waterfall Model's linear progression ensures a well-defined path, allowing for in-depth investigation and comprehensive insights before moving to subsequent phases.

3.1.1 Requirement Analysis

The requirement analysis phase, a key element within the Waterfall Model, was initiated with a thorough examination of existing library management systems. Insights gained from studying these systems identified unique features that inspired the design of the new Library Management System. Additionally, a critical observation was made regarding the complexity of previous systems, highlighting the need for simplicity in the user interface and overall functionality. Within the systematic framework of the Waterfall Model, the requirement analysis phase stands as a crucial juncture where the blueprint for the Library Management System takes shape. This phase follows a meticulous study of existing library systems, serving as the initial stepping stone in translating user expectations and operational needs into concrete project requirements.

This phase is not merely a data-gathering exercise but a strategic process that forms the bedrock for subsequent stages. It lays the groundwork for decision-making, allowing the project team to make informed choices about the system's scope, design, and functionality. The insights gathered during requirement analysis crystallize into a roadmap, guiding the development team toward a shared vision that prioritizes user satisfaction and operational efficiency.

I. Functional Requirements

- **User Sign-Up and Verification:** Users effortlessly create accounts using unique email addresses, ensuring security via email verification with OTP codes for activation, fostering a secure and personalized user experience.
- **Book Listing:** A user-friendly interface enables easy exploration of a diverse list of available books, ensuring accessibility and efficient discovery within the library's inventory.
- **Book Request:** Users submit specific book requests, contributing to a personalized library experience by catering to individual preferences and interests.
- **Issued Books Tracking:** Users efficiently track borrowed books with real-time updates and clear visibility into return deadlines, enhancing accountability and book management.
- **User Profile Maintenance:** A dynamic profile management system allows users to seamlessly view and update their information, fostering user control and customization.
- **User Password Reset:** A secure and straightforward password reset mechanism ensures hassle-free recovery for users, prioritizing account integrity and convenience.
- **Admin Book Management:** Administrators manage the book inventory effortlessly by adding new books and updating existing entries, ensuring the database reflects the evolving collection.

Use Case Diagram:

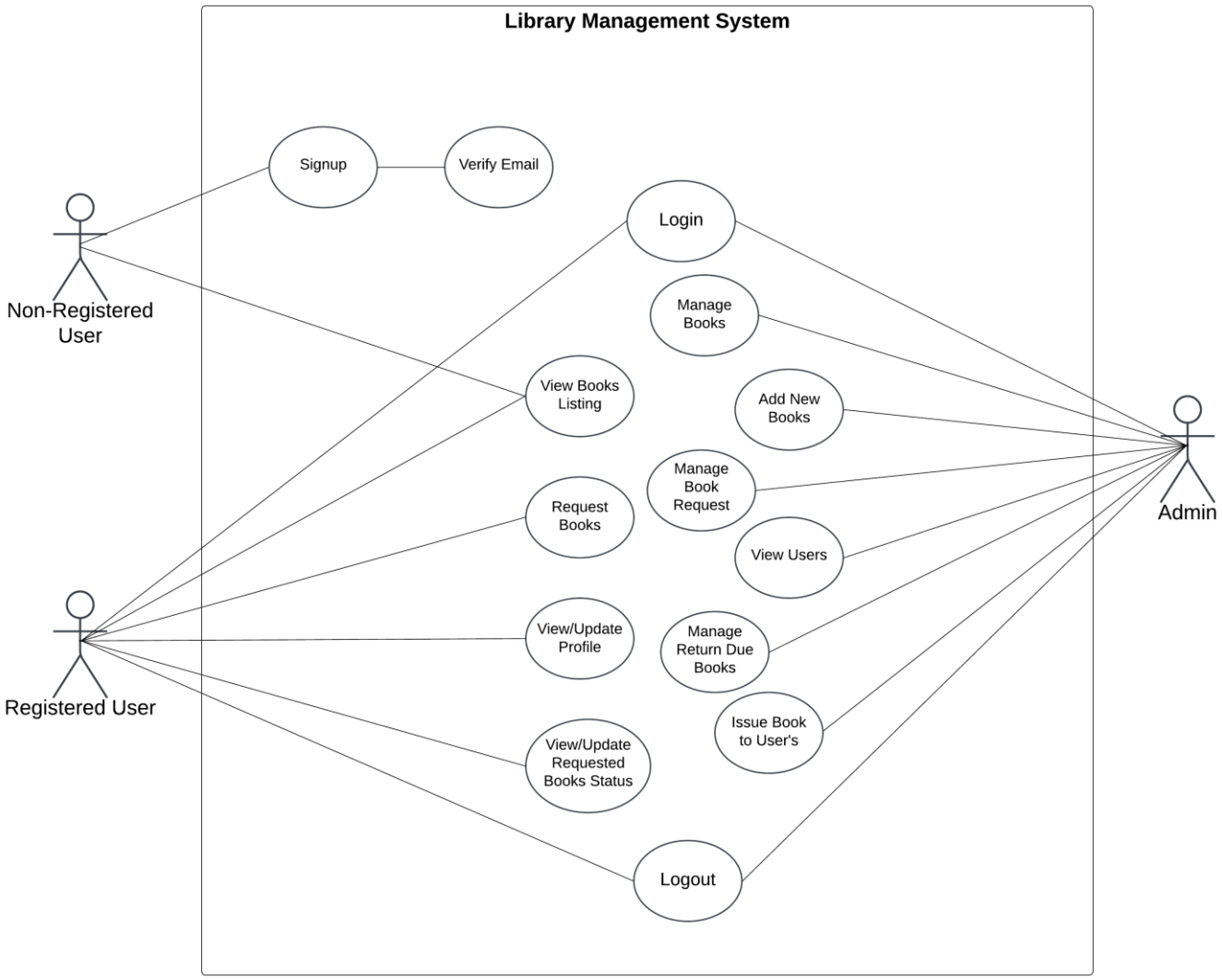


Figure 2 UseCase Diagram

Above use case diagram summarizes the detail of system and the users within the Library Management System. We can see the events and flow of events in above diagram. Above diagram summarizes how users will be interacting with the system with different events. Other events can be triggered anonymously too. Admin can do special event of managing overall system.

II. Non-Functional Requirements

- **Security:** Software security is an idea implemented to protect software against malicious attacks and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication, and availability.
- **Reliability:** Reliability is defined as the probability of failure-free software operation for a specified period in a specified environment. Software Reliability is hard to achieve because the complexity of software tends to be high.
- **Performance:** Performance is an indicator of how well a software system or component meets its requirements for timeliness. Timeliness is measured in terms of response time or throughput. The response time is the time required to respond to a request.
- **Maintainability:** The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment is known as maintainability.
- **Usability:** Usability refers to the quality of a user's experience when interacting with products or systems, including websites, software, devices, or applications. Usability is about effectiveness, efficiency, and the overall satisfaction of the user.
- **Delivery in time:** Software delivery is the process of deploying an application to the market. The software should be completed within the allotted time and delivered on time.

3.1.2 Feasibility Study

This project deals with different fields ranging from system-required technologies, budget and operational factors such as user-friendliness, legal, usability, etc. Conducting a feasibility study was imperative to evaluate the viability of implementing the Library Management System. The study encompassed three key aspects:

I. Technical Feasibility

The technical feasibility assessment focused on the system's compatibility with existing technologies and infrastructure. The study confirmed that the MERN (MongoDB, Express.js, React.js, Node.js) stack was well-suited for the project, ensuring seamless integration and scalability.

II. Operational Feasibility

This project aligned seamlessly with existing library processes and workflows. User-friendly interfaces for both administrators and users ensured smooth adoption and integration into day-to-day operations. Due to simplicity of project training requirements were minimal, further enhancing operational feasibility.

III. Economic Feasibility

The cost of developing and launching this project in the initial phase is moderate. The study revealed that the initial development costs were justified by the expected reduction in manual processes, decreased resource allocation, and improved resource utilization. Technologies required in this project like database and other server are easily available in the markets.

3.1.3 Data Modelling: Entity-Relationship (ER) Diagram

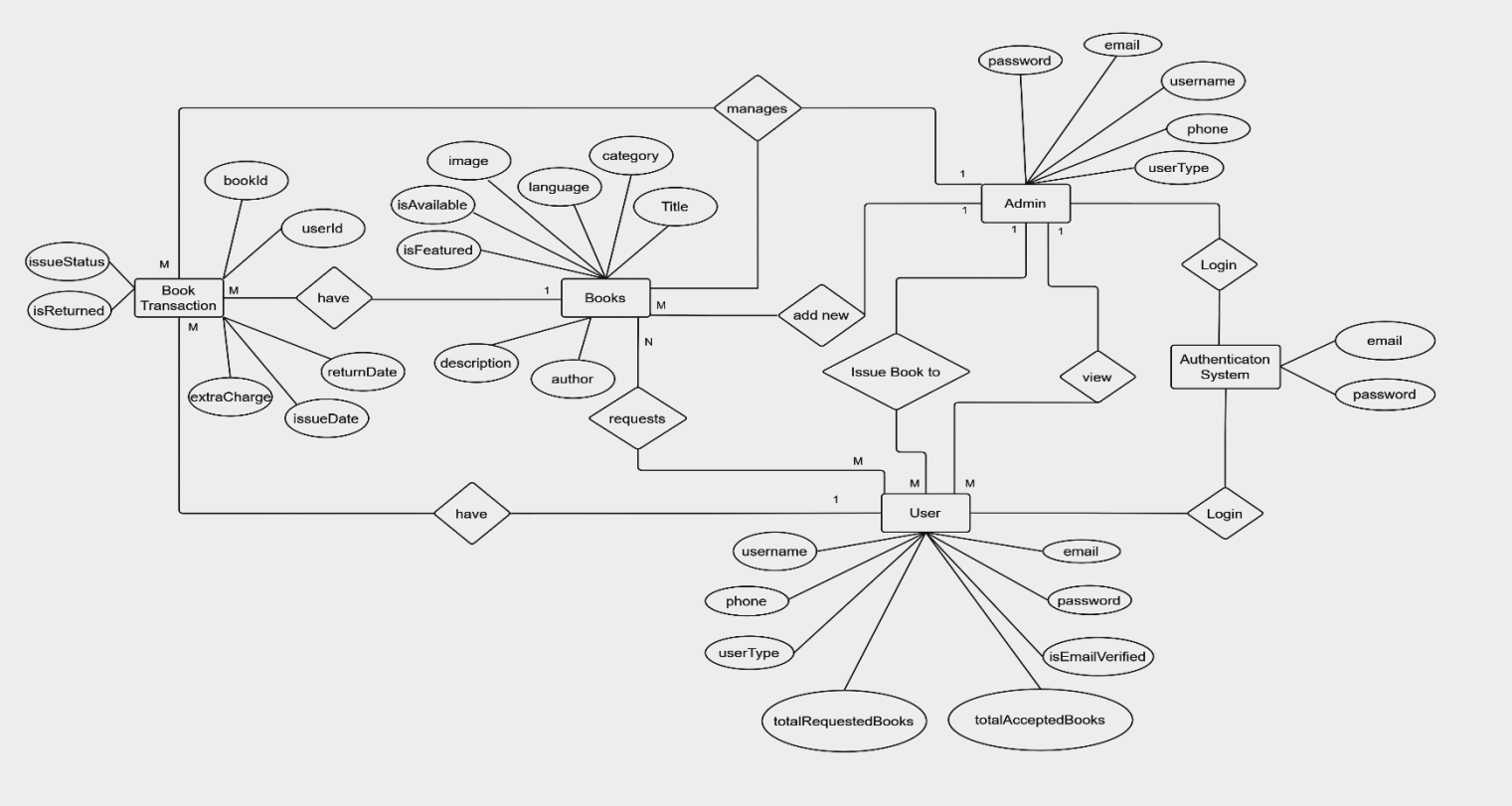


Figure 3 Entity Relationship Diagram

In the Above Entity-Relationship (ER) diagram for a Library Management System (LMS) have many entities representing key components of the system and relationships between them.

Entities:

- Books: Represents books in the library with attributes like BookID, Title, Author, etc.
- Book Transactions: Represents books transactions in the library that stores user's books requests, issued books data, etc.
- User: Represents individuals who interact with the library, with attributes like UserID, Username, etc.
- Admin: Represents administrators with similar attributes to users but distinguished by UserType capable of managing Books, Users, Books Transactions, etc.

Attributes:

- Book: BookID, Title, Author, etc.
- BookTransactions: BookID, UserID, IssueDate, ReturnDate, IssueStatus, IsReturned, ExtraCharge, etc.
- User: UserID, Username, etc.
- AdminUser: Similar to User attributes.

Relationships:

- Admin-Books Relationship: Admin can Manage Existing Books and Add new Books.
- Admin-BookTransaction: Admin can manage users book request, issue book to users, handle whole book transactions, etc.
- Admin-User: Indicates Admin can view and Manage Users
- User-Books: Users can view books listing and request books.
- Books-BookTransaction: All the Book Issuing and Request is Recorded on BookTransaction.

Cardinality:

- One-to-Many Relationships:
  i) One user can have multiple book transactions and view all book listings.
  ii) Admin can manage multiple book transactions, manage all existing books, add new books, etc.
  iii) Admin can issue books to many users and view all users, etc.
  iv) A Book can have multiple books transactions until it's out of Stock.
- One-to-N Relationships: One User can only request for N books i.e., 5Books.

3.1.4 Process Modelling: Data Flow Diagram (DFD)

A process model describes the flow of work or activities, usually in a graphic format, that contributes to accomplishing a specific goal. Process models are typically used to represent and analyze a series of activities that occur repeatedly and regularly. A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.

I. Context level DFD (0 Level)

The 0 Level DFD shows flow of data of the application. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled.

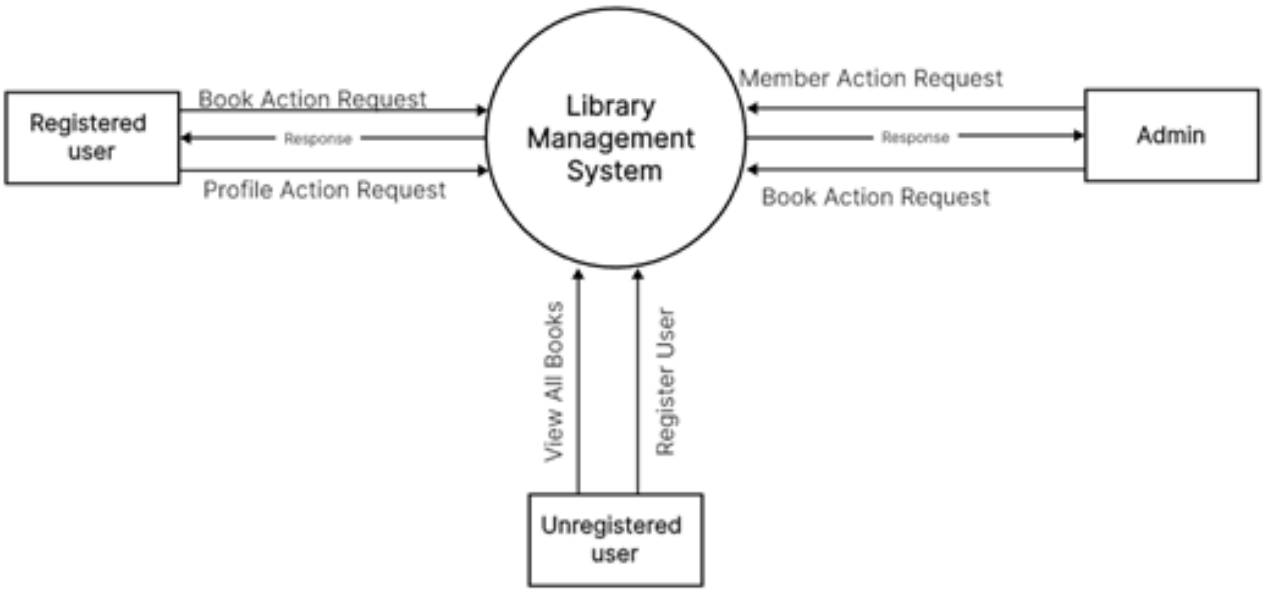


Figure 4 Context Level DFD

In the above Figure, we can see basic overview of library management system processing various operations. The Central part defines library management system process as its Core that handles all the operations of the System. Total of three Entities are connected to the System using parallel lines denoting request and response cycle. This simplified representation serves as the cornerstone for understanding the interplay of data in the broader context of the Library Management System.

II. DFD Level One

DFD Level one provides a more detailed breakout of pieces of the Context Level Diagram. This DFD describes the main functions carried out by the system, as we break down the high-level process of the context Diagram into its sub-processes.

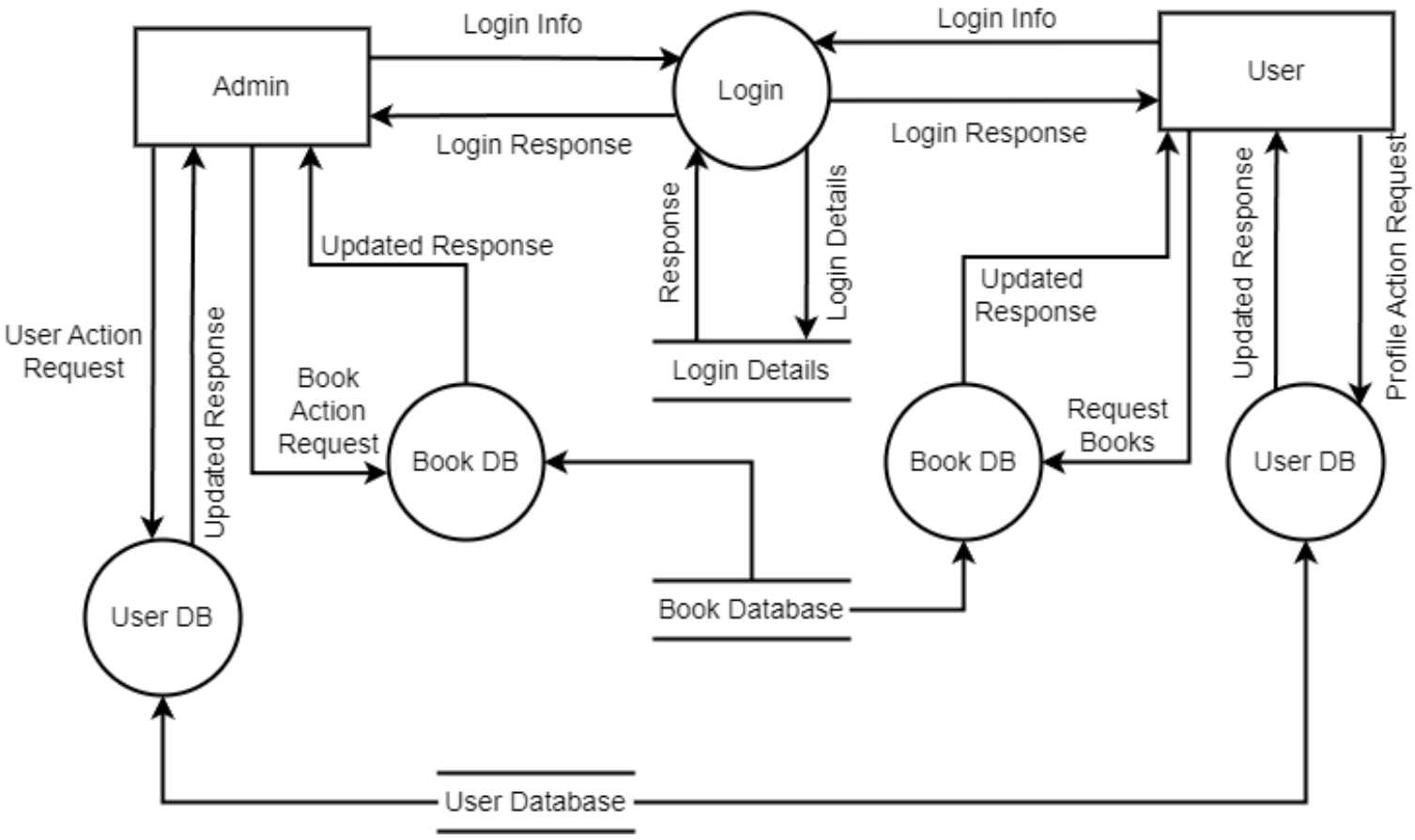


Figure 5 Level One DFD

In the above Figure, data flow between various entities, process and database models are shown. We have two major entities participating in request/response cycle i.e., Admin, User, etc. Login process represents the authentication process of the System. Book Database represents the Books Database and its Models and User Database represents the Users Database and Its Models. Book DB Process represents the operations related to books and User DB Process represents the operations related to Users details.

3.2 System Design

This section defines the basic views of the system. How the application flows to achieve the required objectives. It explains how the system works in simple language.

3.2.1 Architectural Design

Architectural design in software development is the process of defining the overall structure and organization of a system. It involves making high-level decisions on how different components will interact, ensuring scalability, maintainability, and meeting project requirements.

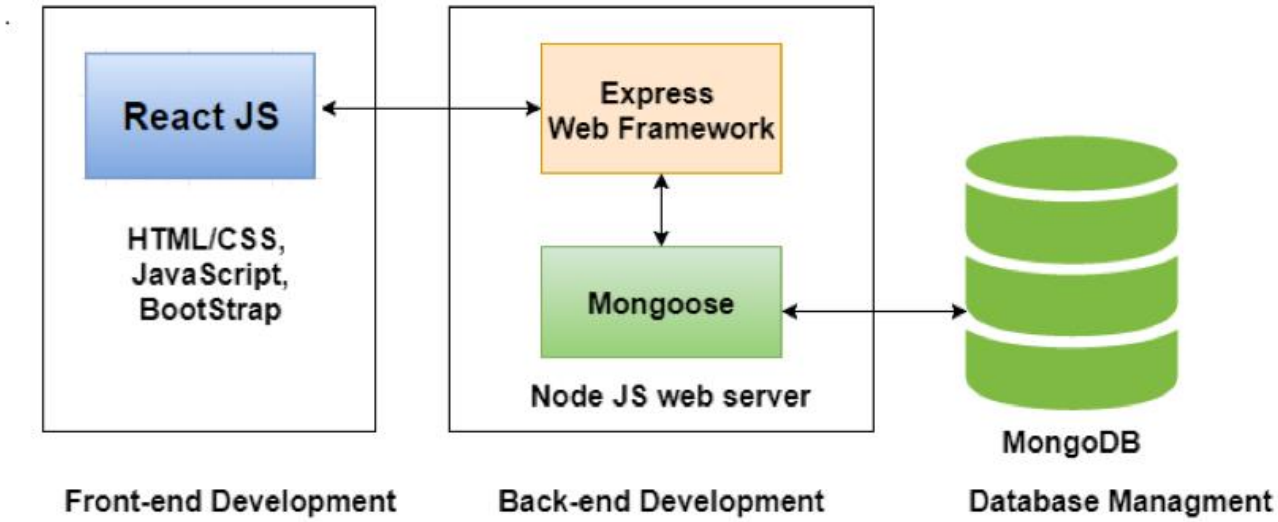


Figure 6 Architectural Design

In the above Figure, it represents a full-stack web development architecture of MERN (MongoDB, Express.js, React.js, Node.js) stack. MongoDB, a NoSQL database, stores data. Express.js facilitates server-side logic, handling HTTP requests and responses. React.js, a JavaScript library, manages the client-side user interface, allowing for dynamic and responsive web pages. Node.js acts as the server runtime, executing JavaScript on the server. Together, these technologies form a cohesive stack, enabling the development of robust and scalable web applications. MongoDB serves as the database layer, Express.js manages the server, React.js handles the front-end, and Node.js orchestrates the entire system, allowing for efficient communication between client and server components. This architecture fosters modularity, ease of development, and real-time data updates, making it popular for building modern web applications.

3.2.2 Database Schema Design

Database schema design is the process of defining the structure that organizes and represents data in a database system. It involves specifying tables, fields, relationships, and constraints to ensure efficient data storage, retrieval, and integrity.

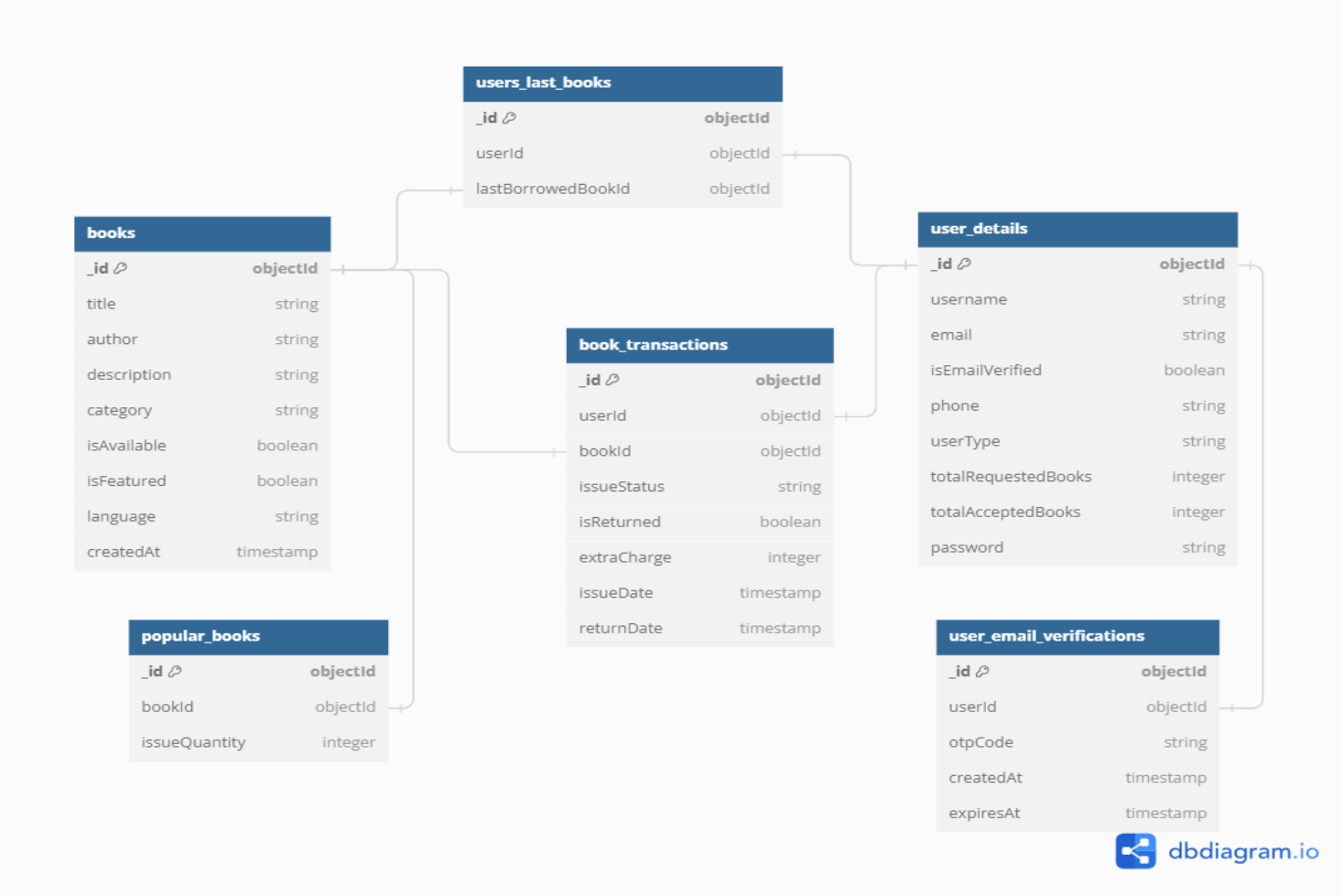


Figure 7 Database Schema Design

In the above Figure, each table has specified fields, data types, and relationships, providing a comprehensive overview of how data is stored and related within the MongoDB database. The books table contains information about each book. book_transactions table tracks transactions, connecting users and books, recording issue and return details, and handling extra charges. user_details table stores user information, including usernames, email, phone, user type, and password. user_email_verifications table manages user email verification with an OTP code and timestamps. The popular_books table identifies popular books based on the quantity issued. The users_last_books table keeps track of the last borrowed book for each user.

3.2.3 Interface Design (UI/UX)

Interface Design, often referred to as UI/UX design, involves creating user-friendly and visually appealing interfaces for software or websites. It focuses on enhancing user experience by optimizing layout, navigation, and visual elements, ensuring an intuitive and enjoyable interaction with the product. Good UI/UX design aims to make technology accessible, easy to use, and aesthetically pleasing for users.

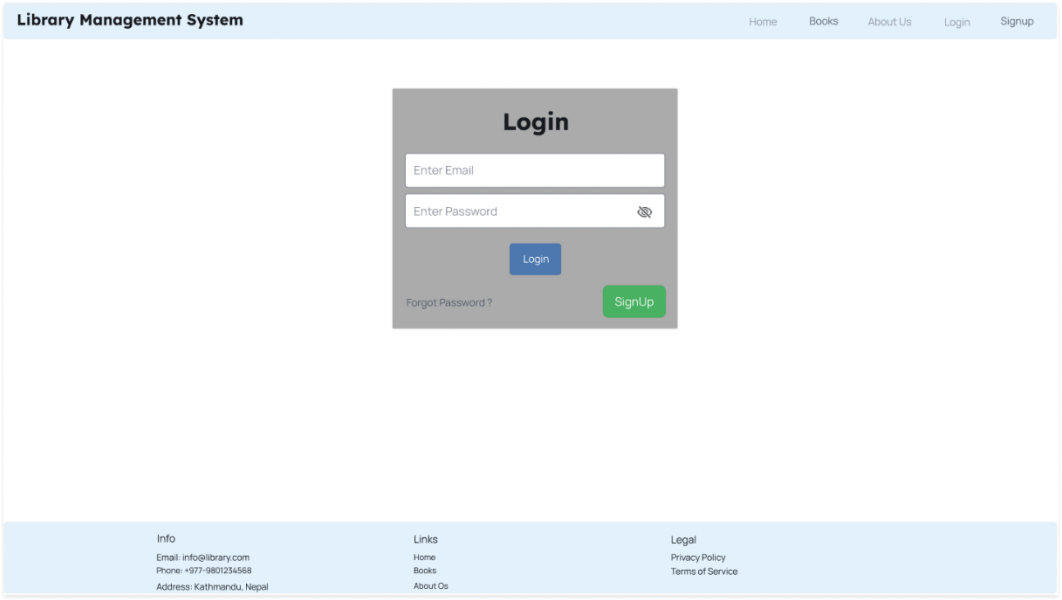


Figure 8 Login Page UI/UX

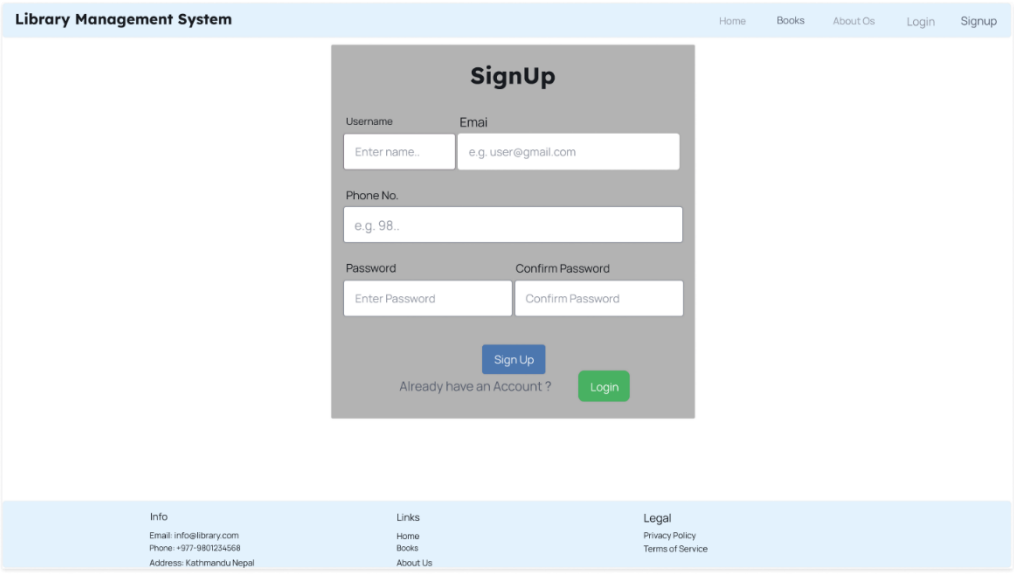


Figure 9 SignUp Page UI/UX

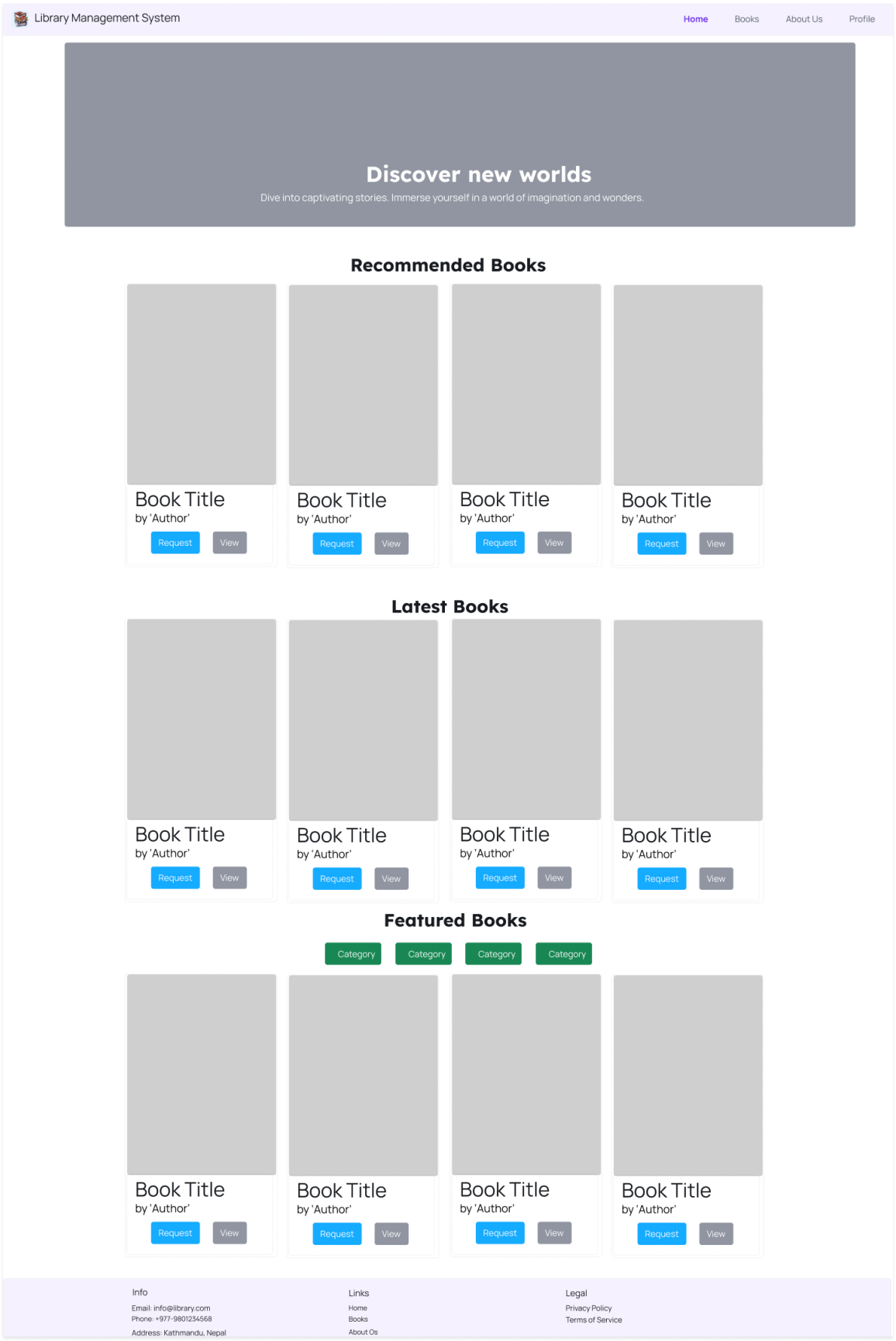


Figure 10 Homepage UI/UX

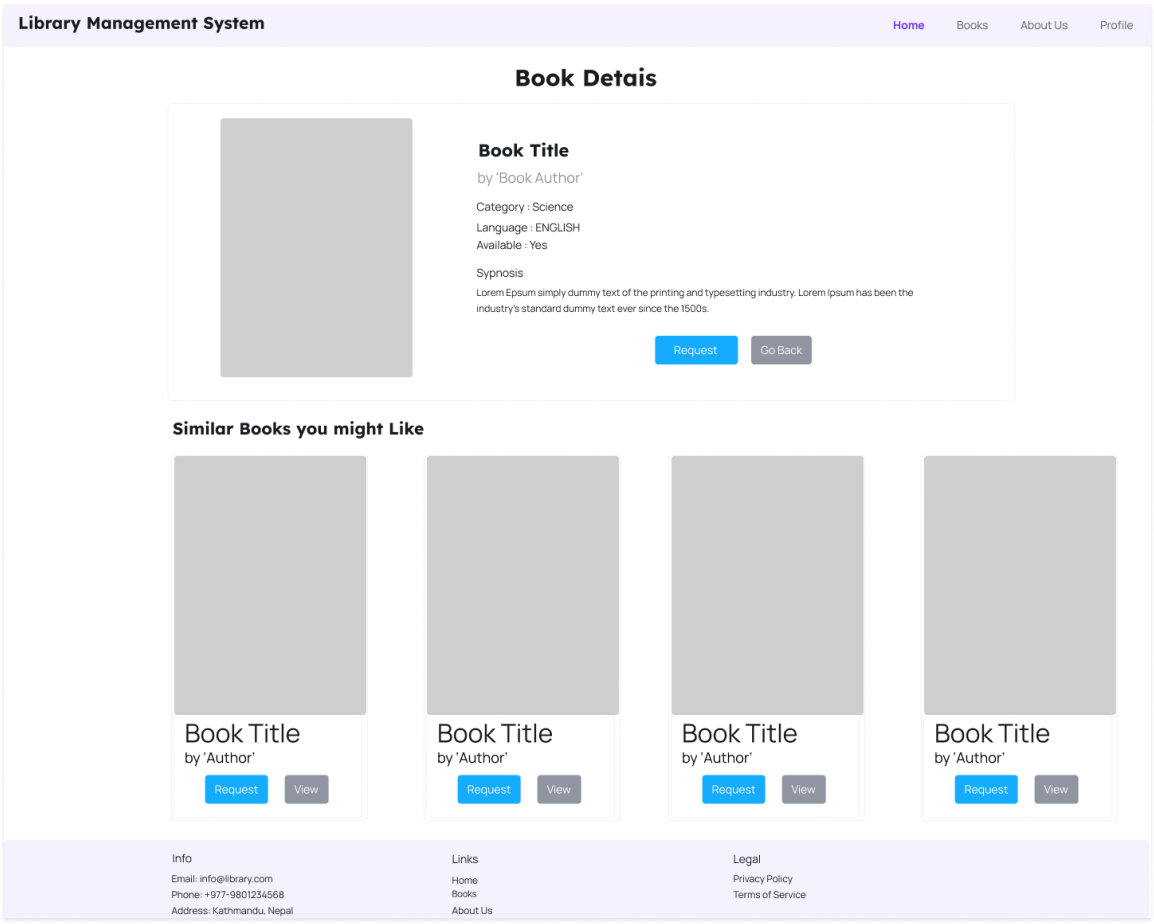


Figure 11 Book Details Page UI/UX

3.3 Algorithm Details

Content-Based Filtering is a recommendation algorithm that suggests items to users based on the characteristics of items they have previously interacted with or shown interest in. In the context of a library management system, it would recommend books to users based on the content features of the books they have read or liked. Features may include book categories, authors, language, and other relevant attributes.

Working Mechanism of Content-Based Algorithm:

- User Interaction Tracking: The system keeps track of books a user has recently read and their past reading history. This information is stored in the database.
- Feature Extraction: Relevant features of books are extracted, such as category, author, and language. These features become the basis for content-based recommendations.
- Similarity Calculation: The algorithm calculates the similarity between the recently read book and the user's past reading history based on common features.
- Recommendation Generation: Books with the highest similarity to the user's reading history are recommended. This ensures that the recommended books share characteristics with the user's preferences.
- Algorithm Activation: The content-based filtering algorithm comes into play after the user has been issued at least one book. This ensures that there is enough user history to generate meaningful recommendations.
- Dynamic Updates: As the user continues to read books, the system dynamically updates the user's profile and refines recommendations based on their evolving preferences.

In summary, content-based filtering algorithm tailors book recommendations to individual users by leveraging their past reading behavior. By focusing on relevant content features, it provides personalized suggestions that align with the user's taste and preferences in categories, authors, languages, and other book attributes.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

This describes how the system works and how it can be important for library management. So, at first, we did research about the necessary tools and sources required to be developed. Many tools and technologies that were utilized to develop the system are discussed in the proceeding chapter.

4.1.1 Tools Used

The various system tools that have been used in developing both the front end and backend of the project are being discussed in this chapter. The project utilizes JavaScript as the primary programming language. The CASE tools used for a project are based on the requirements of the project. Following types of CASE tools are used for this project:

- IDEs (Integrated Development Environments): Visual Studio Code, a lightweight and versatile code editor commonly used for JavaScript-based development.
- Version Control: Git version control system used for tracking changes in source code during development.
- Diagramming Tools: Draw.io and DbDiagram.io is used for creating database schema design, entity-relationship diagrams, and other visual representations of system architecture.

Table 1 Tools Used for Project

Tools	**Purpose**
React Js	Frontend Development
Express Js	Backend Development
Node Js	JavaScript Runtime
MongoDB	Database
VS Code IDE	Coding and Development
Postman	API Testing
Figma	Designing wireframe and UI
GitHub	Version Control

1. React.js (Frontend Development): React.js is a JavaScript library for building user interfaces, providing a component-based architecture for efficient and dynamic frontend development.

2. Express.js (Backend Development): Express.js is a minimal and flexible Node.js web application framework, ideal for building robust and scalable backend services. It simplifies the creation of APIs and server-side logic, seamlessly integrating with the frontend to deliver a complete web application.

3. Node.js (JavaScript Runtime): Node.js is a JavaScript runtime that enables server-side execution of JavaScript code. It powers the backend of the application, facilitating efficient handling of requests, data processing, and interactions with the database.

4. MongoDB (Database): MongoDB is a NoSQL database platform used for storing and retrieving data. Its flexible, document-oriented structure aligns well with JavaScript-based applications, providing scalability and ease of integration with the MERN stack.

5. VS Code IDE (Coding and Development): Visual Studio Code (VS Code) is a versatile and lightweight integrated development environment (IDE) widely favored for its rich features, extensions, and excellent support for JavaScript and React development. It streamlines coding and enhances developer productivity.

6. Postman (API Testing): Postman is a comprehensive API development and testing tool. It simplifies the testing and validation of API endpoints in the backend.

7. Figma (Designing wireframe and UI): Figma is a collaborative design tool used for creating wireframes and designing user interfaces. It ensures a visually appealing and user-friendly frontend design.

8. GitHub (Version Control): GitHub is a web-based platform for version control and collaboration. It enables developers to track changes, manage code repositories, and collaborate on projects. GitHub ensures versioning control for the entire codebase, promoting teamwork and code integrity.

4.1.2 Implementation details of Modules

This section elaborates on how the modules are implemented and what functions does they contribute to the system.

1. User Module

In this module, all the information of user's is stored i.e., username, email, isVerified, userType, etc. This module is used for user authentication and authorization based on the role of the user i.e., admin or client.

2. Books Module

In this module, all the information of Books is stored i.e., title, isFeatured, category, isAvailable, author, language, etc. along with their image's location in the server.

3. BookTransaction Module

In this module, all the information related to books transactions are stored such as, users book request, issued books, book return date, etc.

4. PopularBooks Module

When a User is issued a Book then the book is added to this module to track its total issue count. This data determines the popularity of the book in the system and is used to display the 'Popular Books' listing to users.

5. UserEmailVerification Module

When a user registers to the system a random OTP Code is generated and stored into this module along with the userID which represents the users unique ID. This data is used to verify the user's email.

6. UserLastBook Module

In this module, Which Book has the user recently issued is stored. This data is later used in content-based algorithm to further calculate the users' preferences and choices and recommended books are generated.

4.2 Testing

Based on the software requirement specification document, testing was performed to investigate and validate the behavior of a fully integrated software product. Before deploying an application or website, it must be thoroughly tested. As a result, this application was tested. Some of the types of testing that we did are described below.

4.2.1 Unit Testing

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method or property.

Table 2 Test Case for User Registration

S.N.	Input	Prompt	Expected Output	Actual Output	Result
1	Username: abc Email: xyz.com Phone: 12345 Password: abc123	Username length should be at least 5. Email must have @ symbol. Phone number must start from '9' and 10digits long. Password must have at least one Capital Letter, Number, Unique Symbol, etc.	Registration Complete	Registration Failed	Fail
2	Username: AmitKC Email: amit@gmail.com Phone: 9847612345 Password: Abc!123	Verify Email, OTP Code Sended to Email!	Registration Complete	Registration Successful	Pass

Table 3 Test case for User Login

S.N.	Input	Prompt	Expected Output	Actual Output	Result
1	Email: amit123@gmail.com Password: Abc!123	Invalid Username or Password	Login Success	Login Failed	Fail
2	Email: amit@gmail.com Password: Abc!123	Logged in Successfully	Login Success	Login Success	Pass

Table 4 Test case for Searching Books

S.N.	Input	Prompt	Expected Output	Actual Output	Result
1	Title: Palpasa Category: Novel Language: Nepali	Search Success	Searched Books Listing	Searched Books Listing	Pass

Table 5 Test case for Forgot Password

S.N.	Input	Prompt	Expected Output	Actual Output	Result
1	Email: abc@gmail.com Phone: 12345	Phone number must be 10digits long and start from '9'	Display Change Password Form	Invalid Email or Phone	Fail
2	Email: amit@gmail.com Phone: 9847612345	Credentials Validation Success	Display Change Password Form	Display Change Password Form	Pass

Table 6 Test case for Post Books

S.N.	Input	Prompt	Expected Output	Actual Output	Result
1	Title: abc Author: xyz Description: lorem Category: Novel Language: English Available: True Featured: True Image: None	Title and Author must be at least 5 characters long. Image Field is Required only Jpg, png, jpeg, etc. is supported.	Successfully added Book	Failed to Add Book	Fail
2	Title: TitleOne Author: AuthorOne Description: lorem Category: Novel Language: English Available: True Featured: True Image: Selected	Book added Successfully	Successfully added Book	Successfully added Book	Pass

4.2.2 System Testing

System Testing plays a crucial role in ensuring that the entire system functions as intended and meets the specified requirements. System testing is done to check whether the software or product meets the specified requirements or not.

User Registration and Email Verification

Upon accessing the Library Management System, user can view books listing but has to be logged in for requesting books. For requesting books, User initiates the process by filling out the signup form. The user provides essential details such as username, email, phone, and a secure password. Following the submission of the signup form, an email verification step is triggered.

- User fills out the signup form: Provides necessary details such as username, email, phone, and a secure password.
- Email verification process: User receives an email with a Otp code that expires in a minute. To complete the registration, the user has to enter the Otp code into the verification form and proceed to login.

User Login and Book Request

After successfully verifying their email, the user can log in to the system using their email and password. Once logged in, users have the capability to request books from the books listing. One user can only request for max of 5books or one user can be issued with 5books only. Logged in user can as well update their profile i.e., username, phone, password, etc. but user can't Update their Email.

- User login: Enters email and password login successfully.
- Book request process: User navigates to the book listing section and clicks the request button for requesting books up to a maximum limit of five.
- Content Based Algorithm: Once user is issued a Book, the algorithm comes into play and display recommended books listing to users based on users last issued books.

Admin Management of Book Requests

The administrator plays a crucial role in managing book requests and ensuring the orderly distribution of library resources. Each and every Admin request goes through middleware that authenticates the request. The admin oversees and processes the requests submitted by users.

- Admin views book requests: Accesses the admin dashboard to view pending book requests and Sees details of requested books and the respective users.
- Admin manages book requests: Approves or denies book requests based on availability. If approved, the admin updates the system to reflect the issuance of books to users.

Constraints on User Actions

Certain limitations are imposed on users to maintain order and fairness in the book request system.

- Email verification requirement: Only users with verified email addresses can log in. Unverified users are restricted from requesting books. If unverified user tries to login, then a prompt is displayed to signup or if user has already signed up then email verification process is started.
- Limit on requested books: Users are allowed to request a maximum of five books. This constraint ensures fair access to library resources and prevents misuse.

4.3 Work Schedule

A Gantt chart is a form of bar chart that shows the progress of a project. A Gantt chart, which is widely used in project management, is one of the most popular and useful methods for displaying activities against time. It can also be used to examine a project's start and finish dates in a single graph. Gantt charts were created in our project using Microsoft Excel, as seen in the picture below.

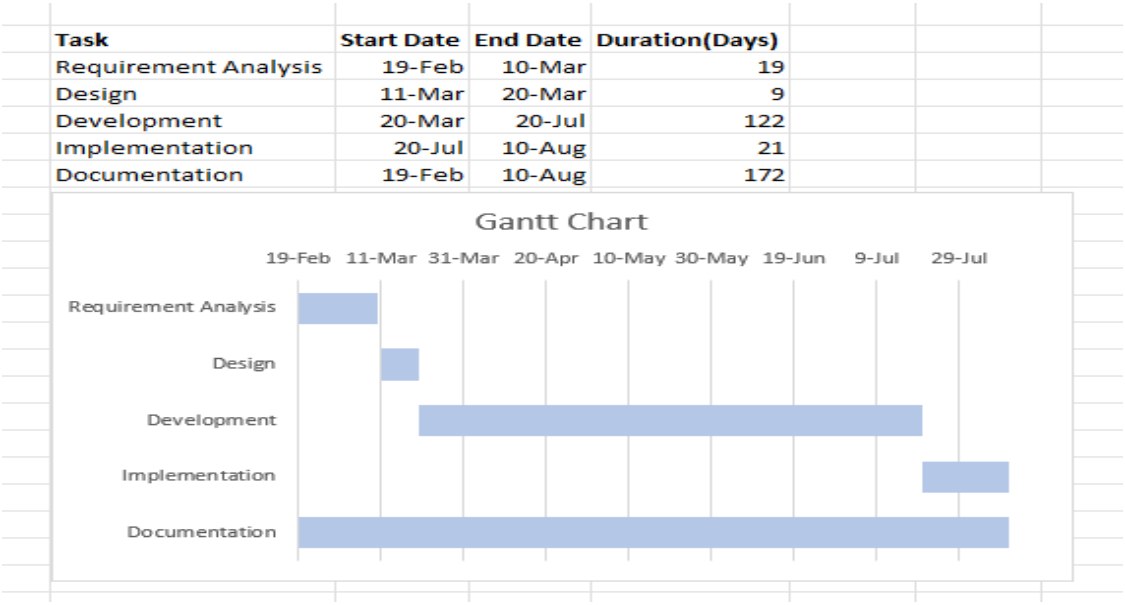


Figure 12 Gantt Chart

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Conclusion

The Library Management and Recommendation System is a simple UI based website, designed to provide a simple interface and to avoid the complex processes of the library system. This System plays a crucial role in enhancing the efficiency and effectiveness of library operations, as well as improving the overall user experience. The system automates various library management processes, such as inventory management, cataloging, and circulation, reducing the time and effort required to perform these tasks manually. This system was successfully implemented using ReactJs, NodeJs, ExpressJs, MongoDB and it was successfully capable of avoiding the complex processes of old library system. Towards the end of the project, it was discovered that the system benefits from several improvements as we felt and realized. Furthermore, enhancements can be made during future development.

5.2 Lesson Learnt

This Project helped gaining a significant technical insight into the intricacies of our chosen technologies, such as ReactJS, NodeJS, etc. navigating challenges and finding innovative solutions. Project management taught the value of realistic timelines and adaptive planning. The testing phase underscored the need for robust quality assurance practices and early bug detection. User experience considerations became pivotal, emphasizing iterative design processes and responsive interfaces. Scalability and future enhancements were on our radar, prompting us to build with flexibility in mind. Adaptability was key, as we navigated unforeseen challenges with agility. Knowledge transfer within the team emerged as a priority, ensuring that insights and best practices are documented for future reference. Embracing a culture of continuous improvement, we conducted retrospectives to refine our processes. Effective stakeholder communication proved crucial, underscoring the importance of regular updates and proactive expectation management.

5.3 Future Recommendation

There are some features that can be added in the future to improve the system furthermore. There is more to be done, thus this system can be enhanced gradually. Few things that can be seen as future recommendations are as follows:

- Upload new Books to System in Bulk
- Online payment system for fine
- Collaborative filtering recommendation

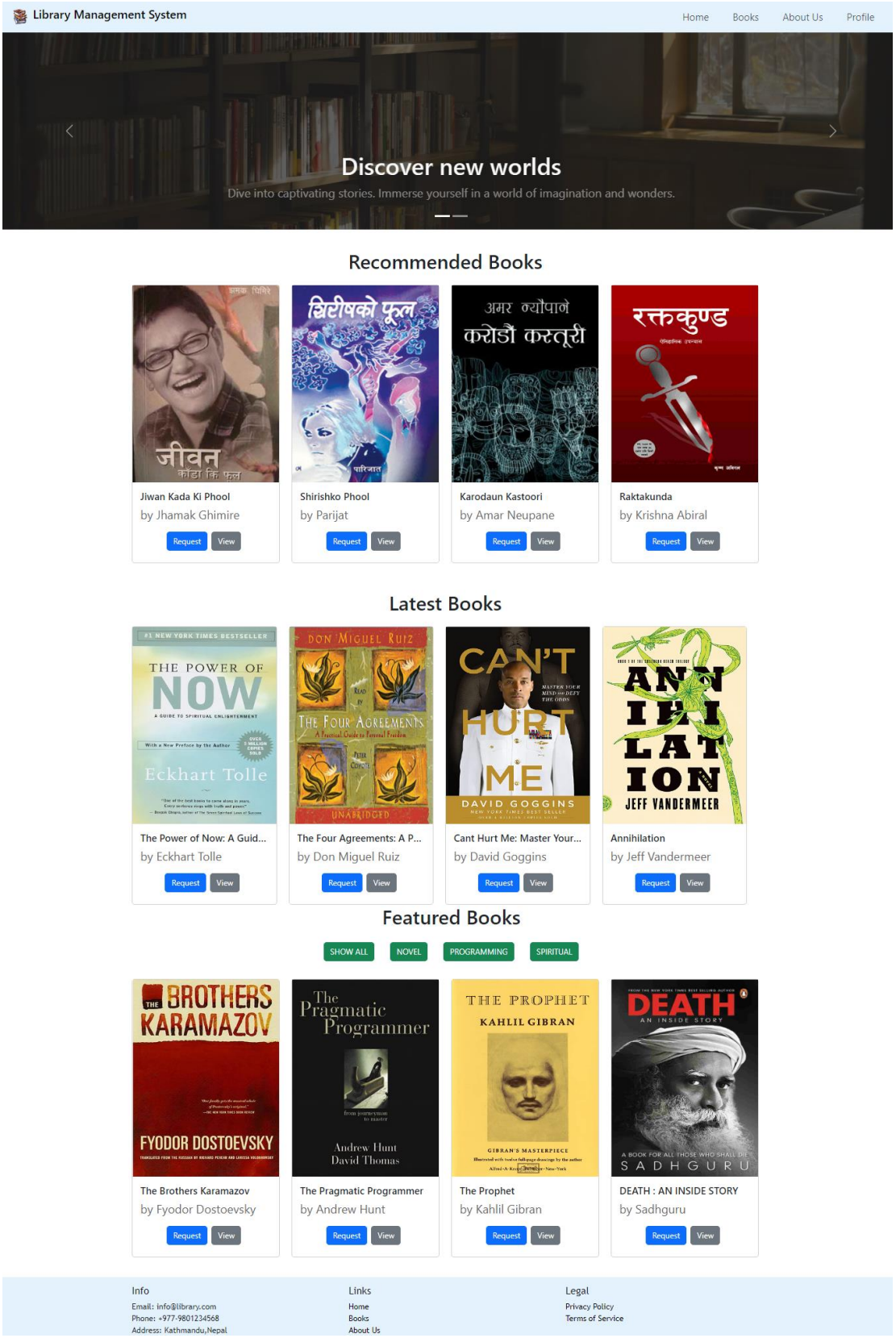
REFERENCES

Some of the information present here is acquired from the websites online

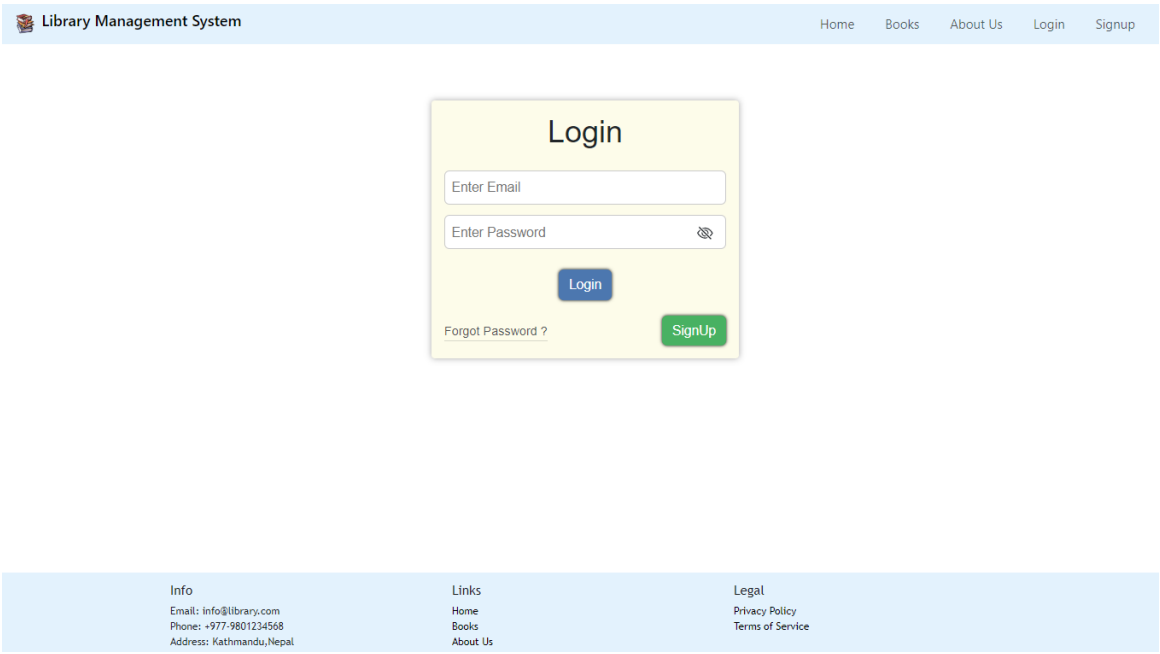
1. “Getting Started React.” https://legacy.reactjs.org/docs/getting-started.html (accessed Jun. 26, 2023).
2. “Node API Documentation.” https://nodejs.org/api/ (accessed Jun. 13, 2023).
3. “Mongoose Docs.” https://mongoosejs.com/docs/ (accessed Jul. 15, 2023).
4. “Nevon Projects.” https://nevonprojects.com/e-library-project/ (accessed Jul. 24, 2023).
5. “PHP Gurukul Projects.” https://phpgurukul.com/online-library-management-system/ (accessed Jul. 26, 2023).
6. “FreeCodeCamp MERN Authenticate.” https://www.freecodecamp.org/news/how-to-secure-your-mern-stack-application/ (accessed Aug. 26, 2023).
7. “Draw.io.” https://app.diagrams.net/ (accessed Aug. 26, 2023).
8. “DBDiagram.io.” https://dbdiagram.io/ (accessed Sep. 13, 2023).
9. “Postman API Testing.” https://learning.postman.com/docs/introduction/overview/ (accessed Sep. 26, 2023).
10. “Bootstrap Docs.” https://getbootstrap.com/ (accessed Jul. 12, 2023).

APPENDICIES

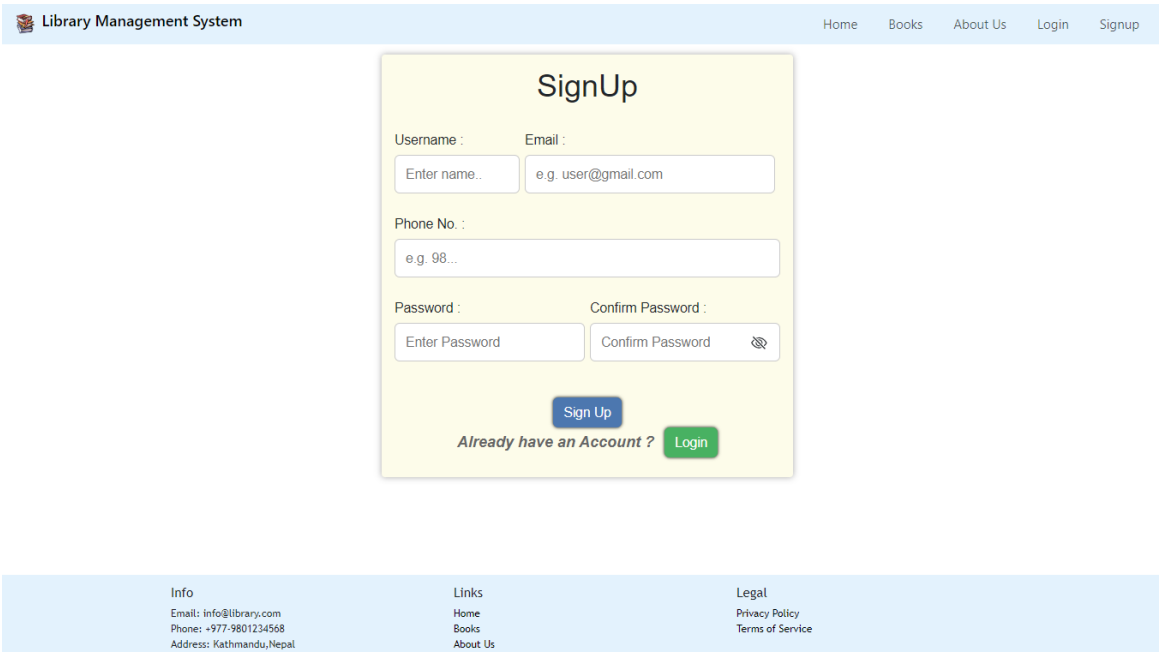
Home Page



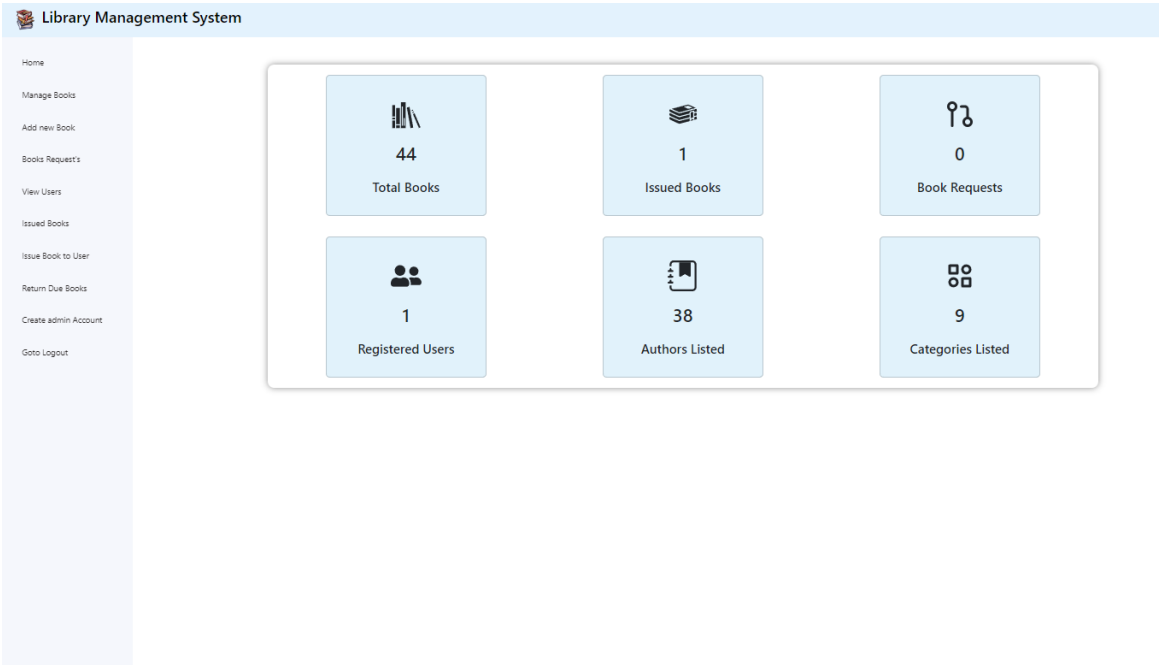
Login Page



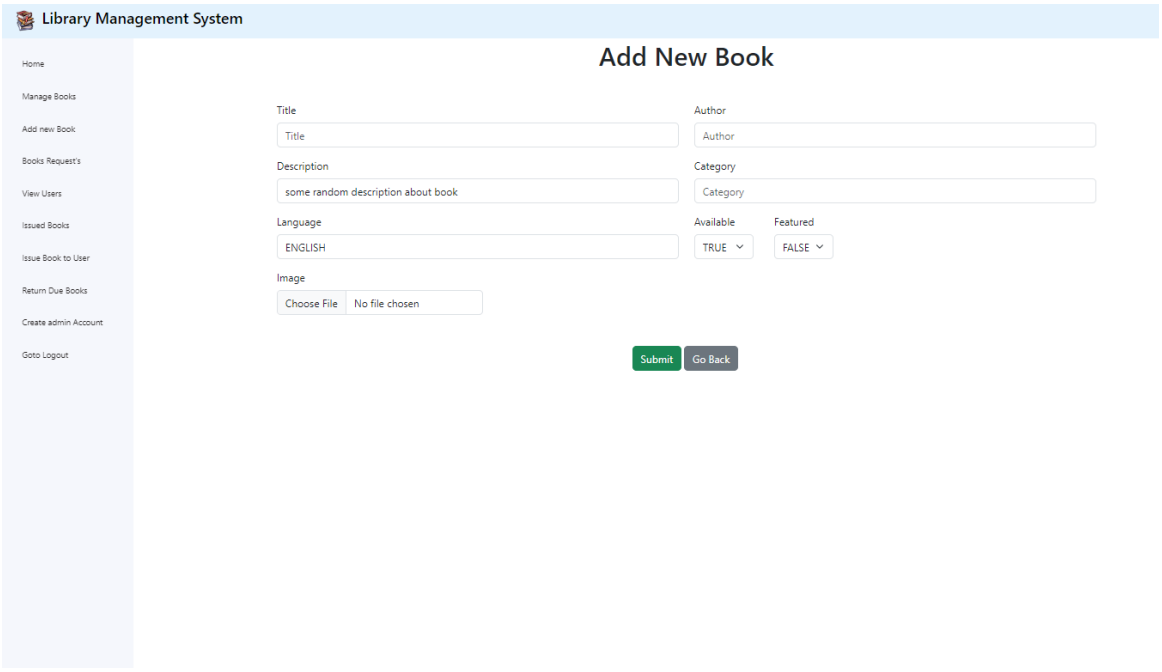
Signup Page



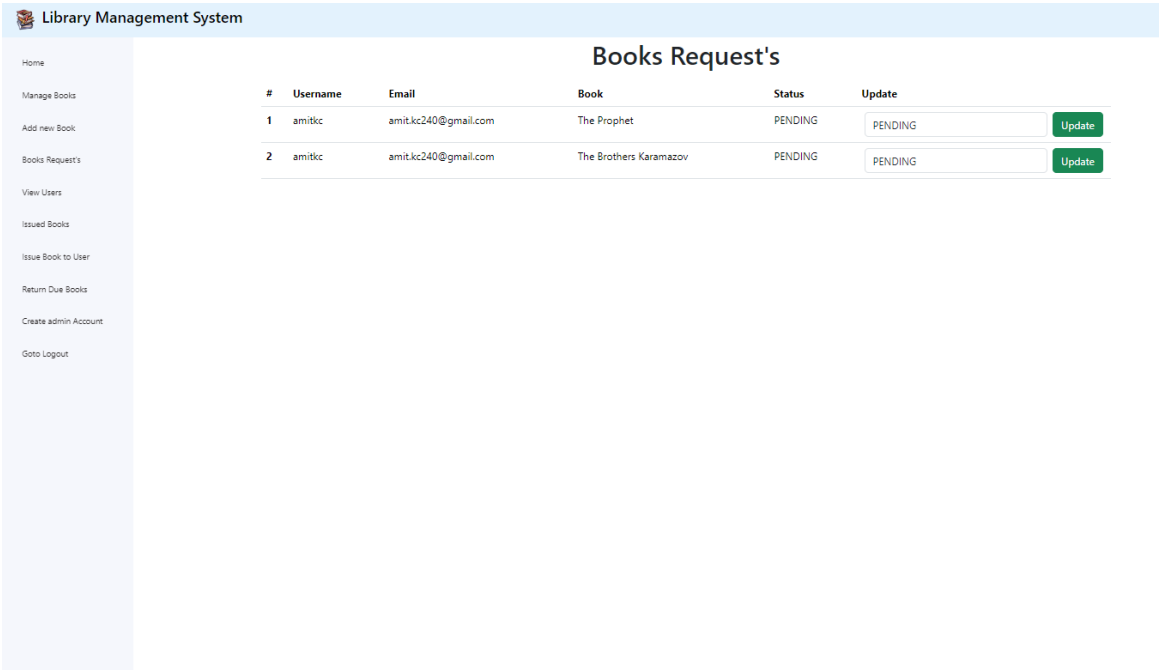
Admin Home Page



Admin Post Books



Admin Book Request's



Admin Manage Books