Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT ON

"Library Management System"

Submitted to

Department of Computer Application

Bhairahawa Multiple Campus

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

SUBMITTED BY:

Anil Barma

TU Registration No:6-2-52-192-2023

Under the Supervision of

Supervisior Name:"Sunil shah"

Tribhuvan university

Faculty of Humanities and Social Science

Bhairahawa Multiple Campus

Supervisor's recommendation

I hereby recommend that this project prepared under my supervision by Anil Barma 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

Teaching Assistant
Bhairahawa Multiple Campus

Tribhuvan university

Faculty of Humanities and Social Science

Bhairahawa Multiple Campus

LETTER OF APPROVAL

This is to certify that this project prepared by " Anil Barma " entitled "**Library Management System** " in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

<table>
<tr><td colspan="2">……..

Campus Chief
Bhairahawa Multiple Campus</td></tr>
<tr><td>……

Program Coordinator
Bhairahawa Multiple Campus</td><td>……

Supervisor
Bhairahawa Multiple Campus</td></tr>
<tr><td>……
Internal Examiner
Bhairahawa Multiple Campus</td><td>……
External Examiner
Tribhuvan University</td></tr>
</table>

ACKNOWLEDGEMENT

I am deeply grateful to several individuals whose support, suggestions, and guidance have played a key role in organizing and finalizing this project. Their continuous encouragement has been invaluable in bringing this project to completion, and I would like to take this opportunity to thank them all.

First and foremost, I would like to express my sincere thanks to my supervisor, for giving me the honor of working on this project, "**Library Management System**" whose guidance, insightful feedback, and constant encouragement have been crucial to the success of this project. Without his support, this project would not have reached its final form. I am also thankful for the many new things I learned throughout the process.

I am equally grateful to my family and friends for their endless support and motivation. Their innovative ideas and constant help have been a source of inspiration and encouragement throughout the development of the project. I am thankful for their belief in my abilities, which kept me going even during challenging times.

Finally, I would like to express my gratitude to everyone who has been part of this journey, offering help, guidance, and encouragement. Their assistance has played a crucial role in the successful completion of this project, and I am truly thankful for their support.

With heartfelt thanks.

STUDENT'S DECLARATION

We hereby declare that the project report entitled "**Library Management System**" submitted in partial fulfillment of requirement for the Bachelor's degree of Bachelor in Computer Application of Tribhuvan University is our original work and has not been submitted for the award of any other degree, diploma, fellowship or other similar title or prize.

……..

" Anil Barma "

ABSTRACT

The **Library Management System** is a web-based platform designed to manage and automate library operations efficiently. This system collects user details during the registration process, enabling members to search, view, and issue books based on their requirements. The online nature of the platform eliminates the need for manual record handling, providing a convenient and organized way to manage library resources.

The traditional library system relied heavily on manual processes, which were timeconsuming and prone to errors. Searching for available books, maintaining issue and return records, and managing member details required significant effort, often leading to delays and inconsistencies. Tracking book availability and overdue records manually made the system inefficient and difficult to manage as the library expanded. To overcome these challenges, an automated Library Management System was developed to improve accuracy and operational efficiency.

This project aims to digitize and streamline all aspects of library management, including maintaining a database of books, managing book issuance and returns, calculating fines for overdue books, and securely storing user information. The system provides real-time updates, allowing librarians to monitor book availability and transaction status instantly. It also enables users to search for books, check availability, and manage their issued books easily.

The Library Management System serves as a comprehensive solution that simplifies daily library operations. By automating routine tasks, it reduces administrative workload and minimizes human errors. Additionally, the system ensures secure data management,

Table of Contents

List of Abbreviations

CASE: Computer-Aided Software Engineering

CRUD: Create, Read, Update, Delete

CSS: Cascading Style Sheets

DFD: Data Flow Diagram

ER: Entity Relationship

HTML: Hypertext Markup Language

SQL: Structured Query Language

UI: User Interface

VS Code: Visual Studio Code

List of Figures

List of Tables

CHAPTER 1:Introduction

1.1 Introduction

The Library Management System is an online system developed to make it easier for users and librarians to manage library activities efficiently. Through this platform, users can search for available books, register on the system, and issue or reserve books according to their needs. The system collects essential user details to manage book transactions effectively .In the traditional manual system, managing book records, issue and return processes, and tracking book availability requires significant time and effort. Finding whether a particular book is available and maintaining accurate records becomes difficult, especially as the library grows. This project aims to automate these tasks, making library management simple and efficient.

1.2 Problem Statement

Managing a library involves various tasks such as maintaining book inventory, managing user records, tracking issued and returned books, and calculating fines for overdue books. Performing these operations manually is time-consuming, error-prone, and inefficient. A user-friendly, web-based Library Management System is required to automate and simplify these tasks.

1.3 Objectives

· To develop a user-friendly web-based Library Management System.

· To digitalize library operations in order to reduce manual workload, minimize errors, and improve efficiency.

· To automate and streamline key processes such as book inventory management, user management, book issuing and returning, and fine calculation.

1.4 Scope and Limitations

1.4.1 a) Scope

- **User Authentication:** Secure login and logout functionality ensures that only authorized users and administrators can access the system.
- **User-Friendly Design:** The system is designed with a simple and easy-to-use interface, making it accessible for both users and library staff.
- **Book Search and Management:** Allows users to search for available books, view book details, and issue or reserve books easily and securely.

1.4.2 b) Limitations

- **Limited Third-Party Integration:** Integration with external services such as online payment gateways or digital libraries is not fully implemented, which may limit advanced features.
- **Limited Language Support:** The initial version supports a limited number of programming or interface languages.
- **Profile Update Restriction:** Users cannot update their profile information after registration, which may reduce flexibility for making changes.
- **Single Admin Role:** The system supports only one administrator role, restricting the assignment of multiple admins or role-based permissions.

1.5 Report Organization

The first part of the report provides a brief introduction to the project. It outlines the purpose of the Library Management System, along with its scope and limitations. This section also presents the problem statement and defines the objectives of the project, such as simplifying library operations and providing efficient access to library resources for users while ensuring secure management for administrators.

The second part presents an overview of the project background. It explains key concepts related to library management systems, including user registration, book searching, issuing and returning processes, fine calculation, and logout mechanisms. This section also reviews existing library systems and highlights how the proposed system differs in terms of features, usability, and security. Additionally, it discusses the motivation behind developing the system.

The third part focuses on system planning and design. It explains the system requirements using diagrams such as use case diagrams and system flow diagrams. It also includes feasibility studies covering technical, economic, and operational aspects. This section describes the system design in detail, including database schema design, Entity Relationship (ER) diagrams, Data Flow Diagrams (DFD), and user interface design.

The fourth part discusses system implementation and testing. It lists the tools, programming languages, and database platforms used in the development of the system. It also explains the testing methods applied to ensure that the system functions correctly and meets user requirements.

The final part concludes the report by summarizing the outcomes of the project, lessons learned during development, and possible improvements. It also provides recommendations for future enhancements, such as adding advanced search features, online fine payment, and multi-admin support.

CHAPTER 2: Background Study and Literature Review

2.1 Background Study

When comparing the proposed **Library Management System** with existing traditional libraries and basic digital library systems used in many institutions in Nepal, several key differences can be identified. Many libraries still rely on manual record-keeping or partially computerized systems, where book records, issue details, and user information are maintained using registers or simple spreadsheets. These systems are slow, error-prone, and difficult to manage as the number of books and users increases.

Some existing library systems provide basic book search functionality but lack advanced features such as real-time availability status, user profiles, borrowing history, and automated fine calculation. In many cases, users must physically visit the library to check book availability or request book issuance, which consumes time and effort.

The proposed Library Management System improves upon these limitations by offering a user-friendly interface, faster access to information, and advanced search options based on book title, author, category, or availability. It allows users to view their issued books, track return dates, and check fines online. Administrators can easily manage book inventory, users, issue and return processes, and generate reports efficiently.

Overall, while existing library systems provide only basic services, the proposed system enhances usability, accuracy, and efficiency by automating key library operations and providing better tools for both users and administrators.

2.1.1 Literature Review

Previous studies on library management systems indicate that earlier library operations were largely dependent on manual processes. Although this approach was widely used, it had several significant drawbacks:

Data records were often inaccurate and inconsistent

Maintaining and generating reports was difficult and unreliable

The overall process was slow, tedious, and inefficient

Retrieving or correcting old records required considerable effort

There was a high risk of data loss, damage, or unauthorized access

2.1.2 Technological Frameworks

A substantial body of research has explored the technological foundations of web-based library management systems. According to Sharma et al. [1], modern library systems commonly utilize web development technologies such as HTML5, CSS3, JavaScript, and backend frameworks like PHP or Node.js, supported by relational databases such as MySQL. These technologies enable efficient data handling, real-time updates, and user interaction.

Further studies highlight the role of cloud computing and APIs in improving scalability, data accessibility, and system integration [2]. For example, Ogbiti and Aaron [3] developed a web-based library management system emphasizing automation of book inventory, user records, and transaction management. Similarly, Nasr et al. [4] proposed a digital library system with real-time data access, demonstrating how technology enhances operational efficiency and information availability.

2.1.3 User Experience and Usability

User experience (UX) plays a vital role in the successful adoption of web-based library management systems. Tuyambaze et al. [5] emphasized that intuitive interfaces and simple navigation significantly improve user engagement in information systems. In the context of libraries, features such as advanced book search, real-time availability status, borrowing history, and clear due-date notifications enhance user satisfaction.

Personalized user dashboards, easy access to issued books, and automated alerts for due dates or fines contribute to a better user experience. These usability improvements reduce dependency on library staff and allow users to manage their library activities independently and efficiently.

2.1.4 Security and Privacy

Security and privacy are critical concerns in web-based library management systems, as they store sensitive user information and transaction records. Luo et al. [6] highlighted the importance of secure authentication mechanisms and data protection strategies in information management systems. Common security measures include password hashing, role-based access control, and encrypted data transmission using SSL protocols.

2.1.5 Challenges and Future Directions

Despite their benefits, web-based library management systems face several challenges. These include limited technical infrastructure in some institutions, cybersecurity threats, resistance to adopting new technologies, and varying user expectations. Additionally, scalability and integration with digital libraries or online payment systems for fine collection remain areas for improvement.

CHAPTER 3: System Analysis and Design

3.1 System Analysis

System Analysis refers into the process of examining a system with the intent of improving it through better procedures and methods. It is the process of planning a new system to either replace or complement an existing system. It is, therefore, the process of gathering and interpreting facts, diagnosing problems and using the information to re- comment on improvements in the system.

System analysis is conducted with the following objectives in mind:

- Evaluate the system concept for feasibility.
- Perform economic and technical analysis.
- Allocate functions to hardware, software people, database and other system elements.

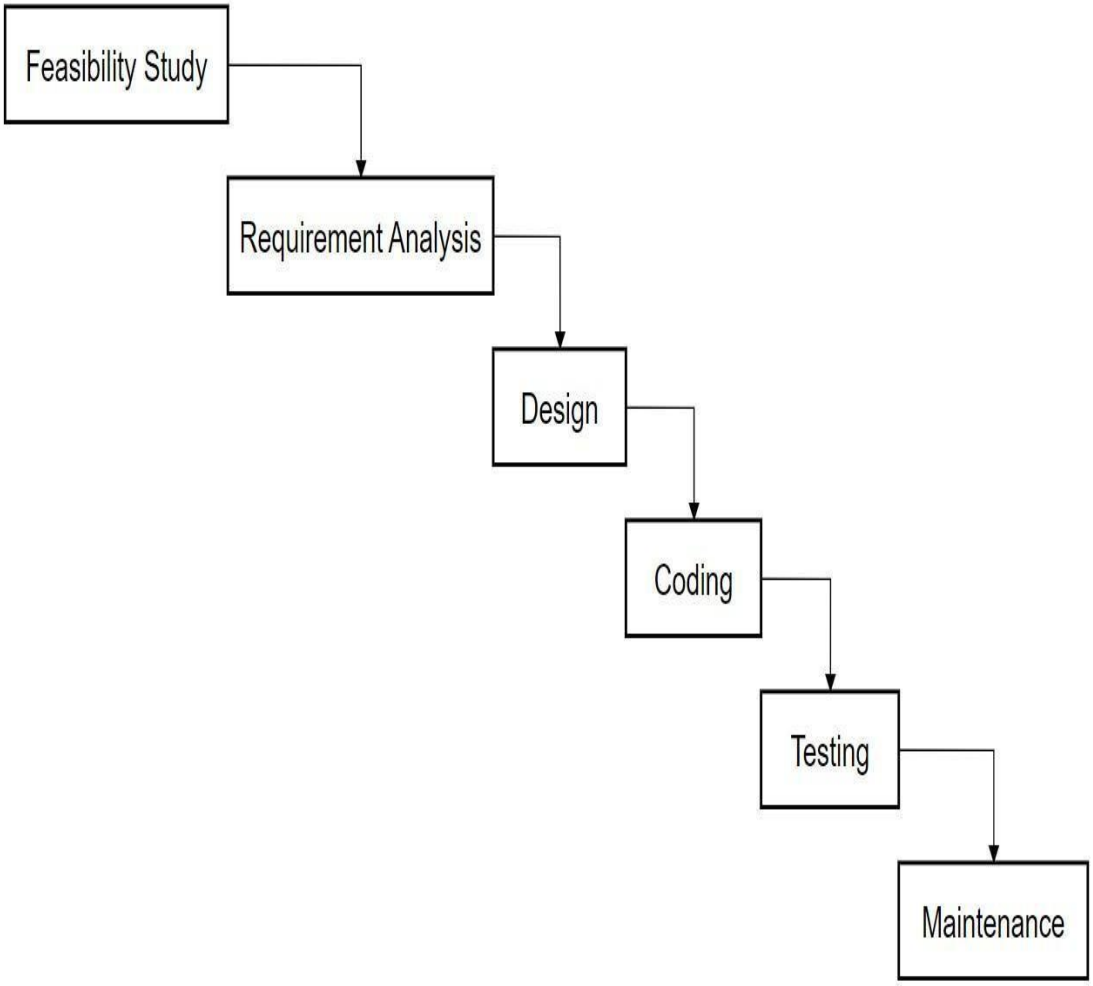

Figure3.1: Waterfall Model

3.1.1 Requirement Analysis

Requirement analysis involves refining and prioritizing the collected requirements to ensure they align with the project's objectives and constraints. This process includes:

I. Functional Requirements

- User Authentication: Users should be able to sign up, log in, and log out. •
  CRUD Operations: Users can create, read and Admin update, and delete.
- Admin Management: Admins can add books , Issue books, User's Management.
- User management : Registered users can view books,Books prices .

Use Case Diagram

A use case diagram is a graphical representation used in software engineering to illustrate the interactions between users (actors) and a system. It captures the functional requirements of a system, showing how different users engage with various functionalities within the system.

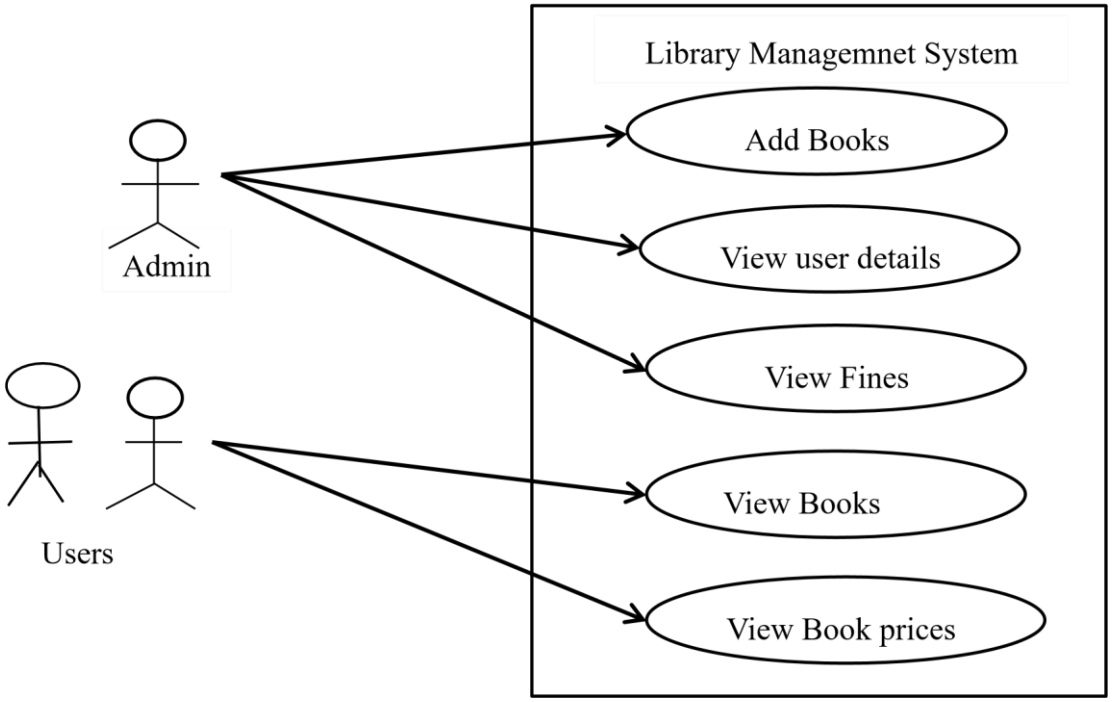

Figure CHAPTER 3: System Analysis and Design.1.1.1: Use Case Diagram

II. Non-Functional Requirements

Non-Functional Requirements describe the general properties of a system. They are among the most important things to define when building the specification for code snippet saving platform, as most have a direct impact on the experience and satisfaction of your users.

Here are some basic types of nonfunctional requirements:

- **Security:** It should be secure, allowing only the authorized users to access and manage their snippets.
- **Usability:** The user interface should be minimal and intuitive, allowing users to easily navigate to different components of the application.
- **Responsiveness**: The app should be fully functional on both desktop and mobile devices.
- **Performance**: The application should have fast response time.

3.1.2 Feasibility Analysis

Feasibility analysis of the Library Management System (LMS)is conducted to determine whether the proposed system is practical

- **Technical Feasibility**

The required technology stack (i.e., web technologies, database systems) is readily available and well-supported.

- **Operational Feasibility**

Skills, time management, and maintenance plans are feasible, with strategies in place for user adoption.

- **Economic Feasibility**

The web app does not require extra software and hardware i.e. it uses free and open-source technologies. So, there is no recurring cost than just the internet connection.

- **Schedule**

This includes the project schedule, and all the time allocated for its completion. The Gantt chart is as follow:

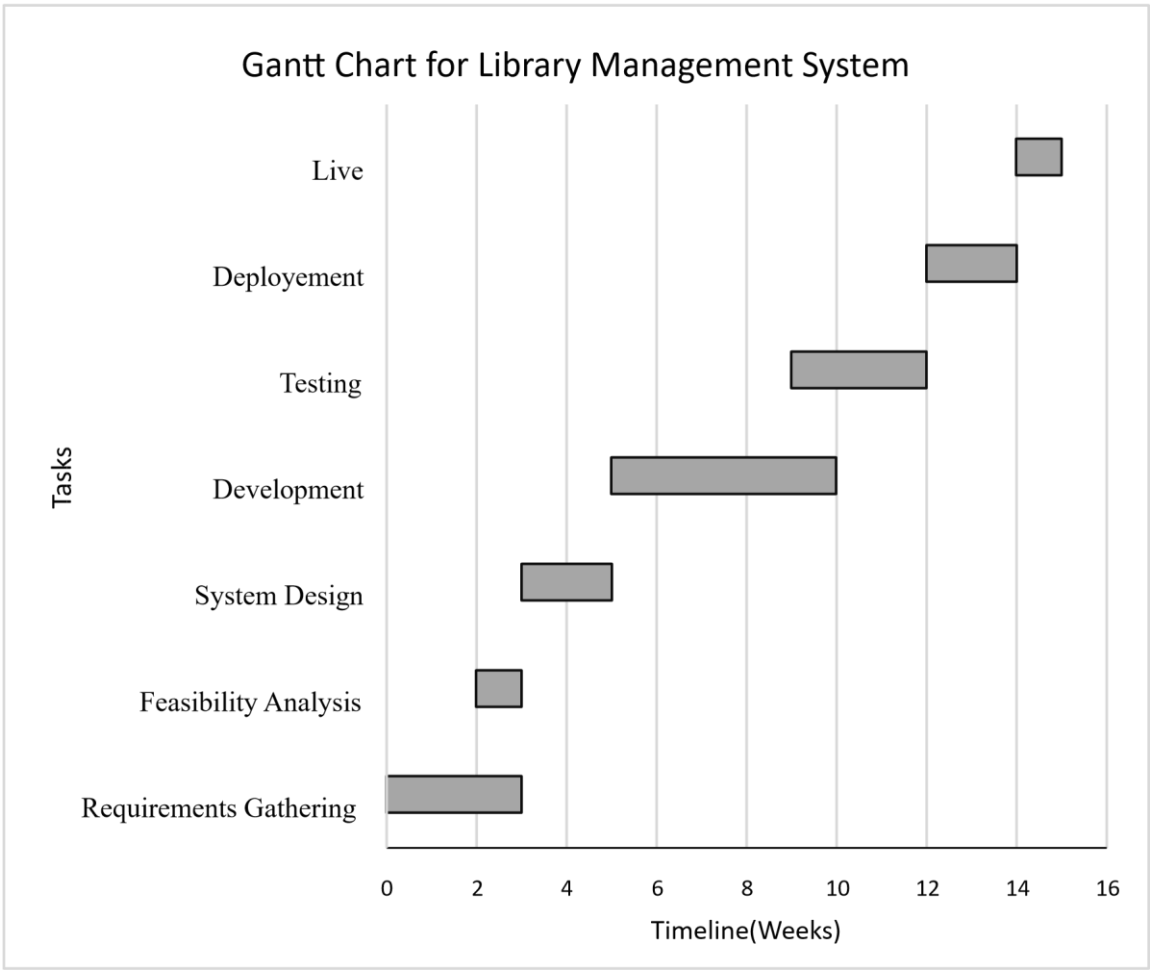

Figure CHAPTER 3: System Analysis and Design.1.2.4: Gantt Chart

3.1.3 Data Modelling (ER-diagram)

Data models define how data is connected to each other and how they are processed and stored in a system. There are different data models which are used to design and develop the database, but here we use ER (Entity Relationship) Model.

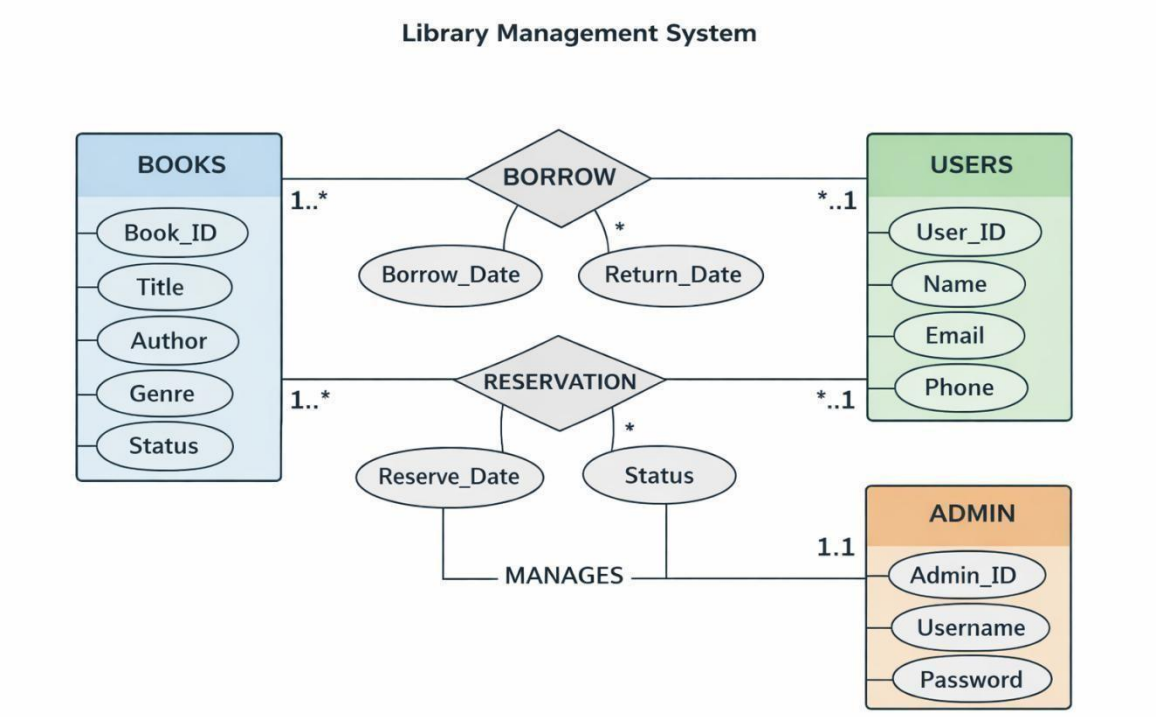

Figure CHAPTER 3: System Analysis and Design3.1.3: ER Diagram

The above ER diagram represents a library management system with six main entities.

- **ADMIN:** Represents the administrators of the system.

Attributes: ADMIN_ID (username), ADMIN_PASSWORD (password)

- **BOOK:** Represents the books available in the library.

Attributes: BOOK_ID (unique book ID), BOOK_NAME (book title), AUTHOR (author name), CATEGORY (book category), ISBN (book number), AVAILABLE (availability status), PRICE (book price)

- **USER:** Represents the registered library members.

Attributes: USER_ID (unique user ID), FNAME (first name), LNAME (last name), EMAIL (email address), PASSWORD (login password), GENDER (gender), PHONE (phone number), ADDRESS (address)

- **ISSUE (BORROW):** Represents the book issuing and returning process.

Attributes: ISSUE_ID (unique issue ID), ISSUE_DATE (book issue date), DUE_DATE (expected return date), RETURN_DATE (actual return date), BOOK_ID (foreign key), USER_ID (foreign key)

- **FINE :** Represents fine details for late book returns.

Attributes: FINE_ID (unique fine ID), AMOUNT (fine amount), ISSUE_ID (foreign key)

- **FEEDBACK:** Represents feedback provided by users.

Attributes: FEED_ID (unique feedback ID), COMMENT (user feedback), USER_ID (foreign key)

3.1.4 Process Modeling (DFD)

Process Modeling using **Data Flow Diagrams (DFDs)** visually represents the flow of data within the Library Management System. DFDs illustrate how data moves between users, system processes, and data stores. The system is represented at different levels of detail, where **Level 0** provides a high-level overview, **Level 1** breaks the system into major functional processes, and **Level 2** further explains each process in detail. These diagrams help in understanding, designing, and documenting the system clearly and effectively.

The **Level 0 DFD (Context Diagram)** presents a high-level view of the Library Management System. It shows the **User** interacting with the **Library Management System** to perform activities such as searching for books, issuing or returning books, and viewing results. The **Admin** interacts with the system to manage books, users, and transactions. At this level, internal processes are not shown; only the flow of data between external entities and the system is represented.

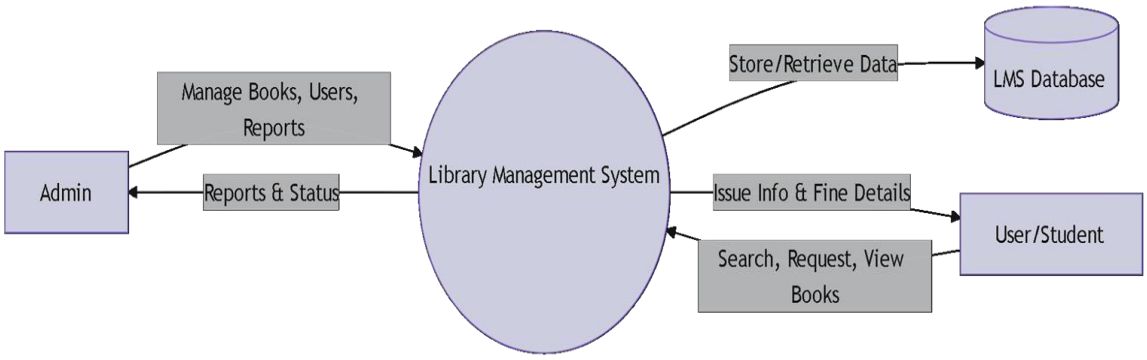

Figure CHAPTER 3: System Analysis and Design3.1.4.1: Level 0 DFD

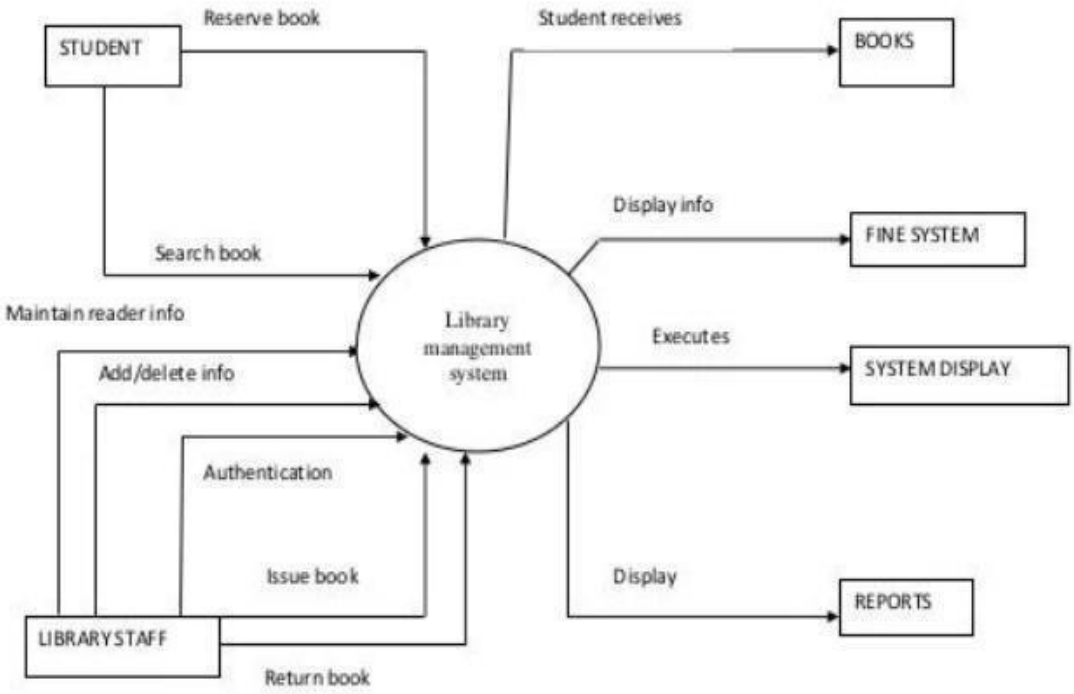

Figure CHAPTER 3: System Analysis and Design3.1.4.2: Level 1 DFD

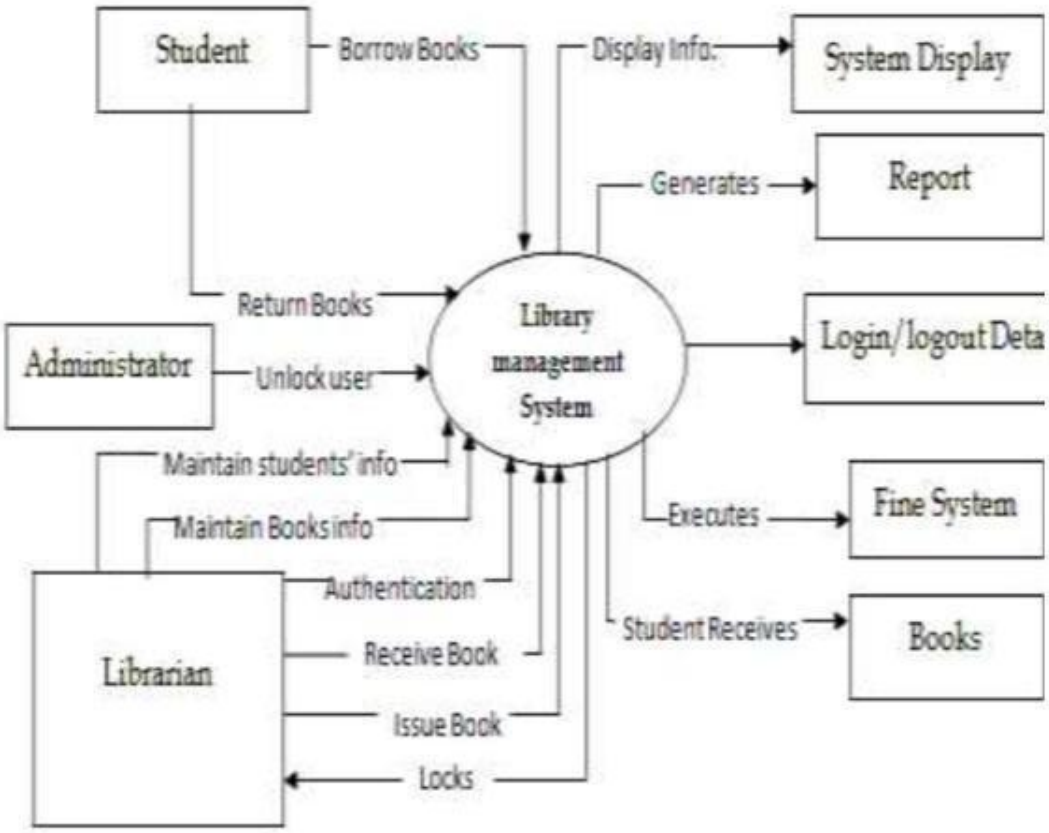

Figure CHAPTER 3: System Analysis and Design3.1.4.3: Level 2 DFD

3.2 System Design

A **Library Management System (LMS)** is a software system designed to manage and automate the daily operations of a library such as maintaining book records, managing users, and handling book issue and return processes. In system design, the LMS is generally built using a **3-tier architecture**, which consists of the presentation layer, business logic layer, and database layer. The presentation layer includes the user interface developed using technologies like HTML, CSS, and JavaScript, where users (admin or students) interact with the system. The business logic layer, typically implemented using a backend language such as PHP, processes user requests, applies validation rules, manages sessions, calculates fines, and controls access permissions. The database layer uses a relational database like MySQL to store structured data including books, users, authors, categories, and issued book records.

3.2.1 Architectural Design

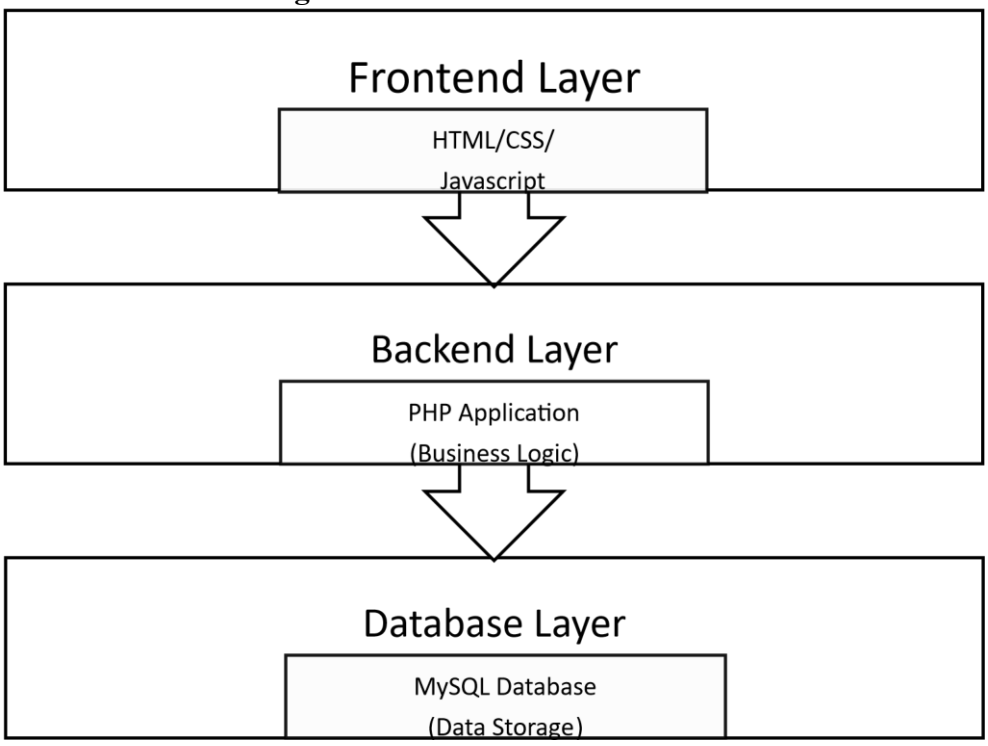

Figure CHAPTER 3: System Analysis and Design.2.1: Architectural Design

3.2.2 Database Schema Design

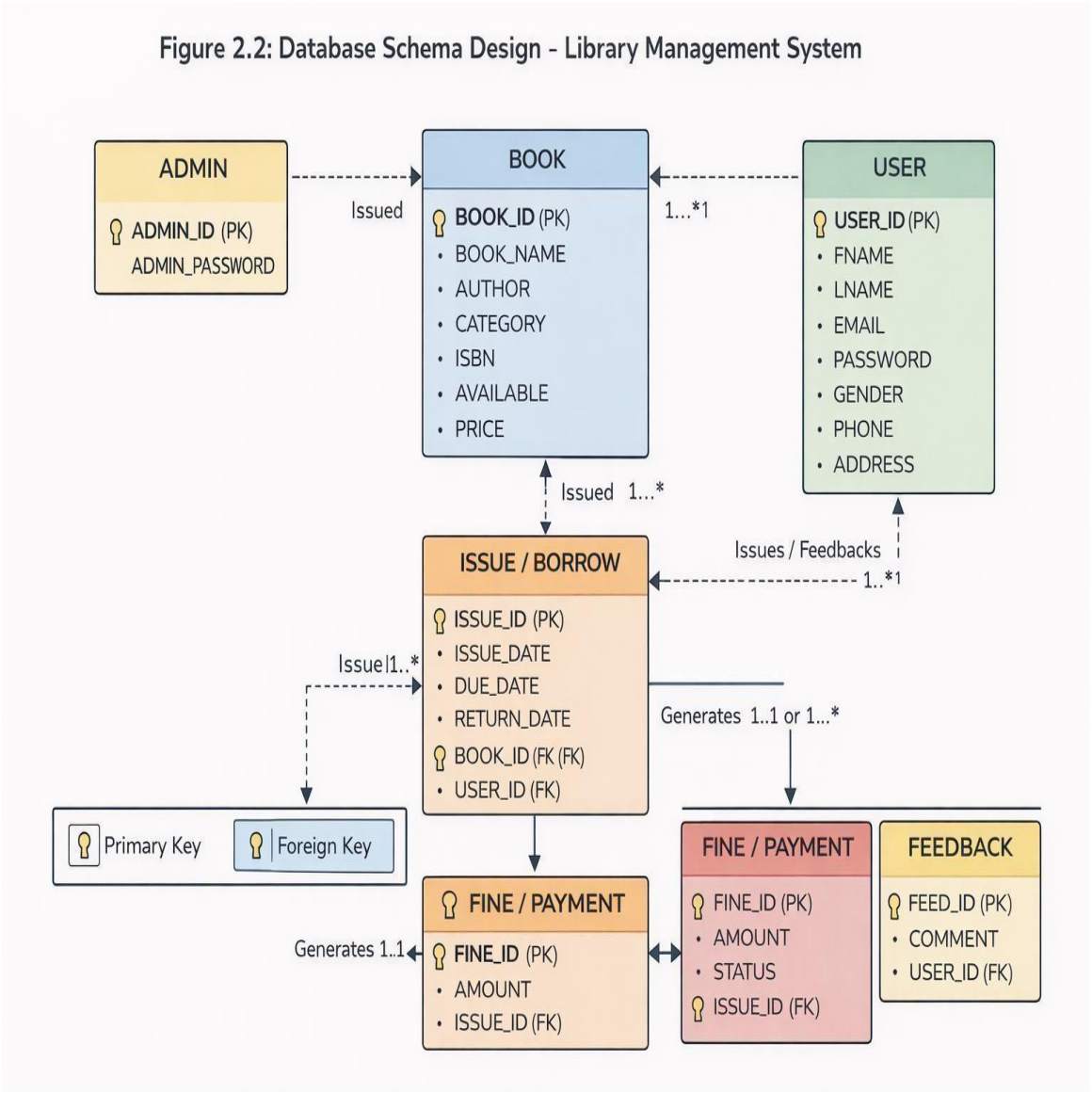

Figure CHAPTER 3: System Analysis and Design.2.2: Database Schema Design

3.2.3 Interface Design

The interface design is focused on providing a simple, interactive, and user-friendly experience for users and administrators of the Library Management System. The target users include students, teachers, and library staff who require an efficient way to search books, issue or return books, and manage library records online.

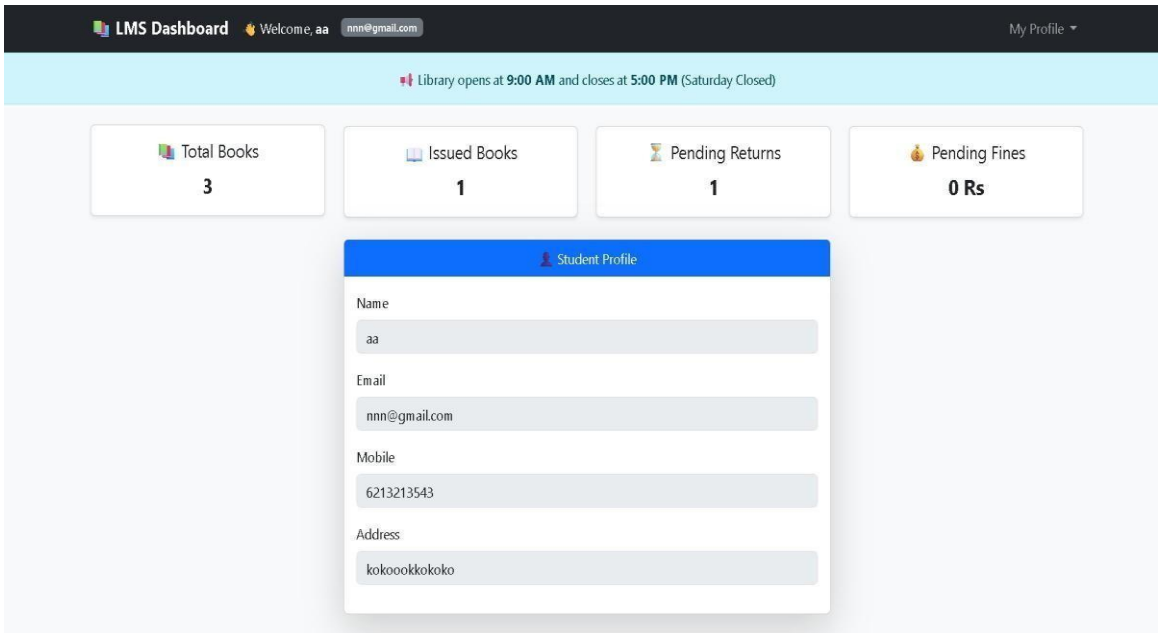

Figure CHAPTER 3: System Analysis and Design 3.2.3: Home Page

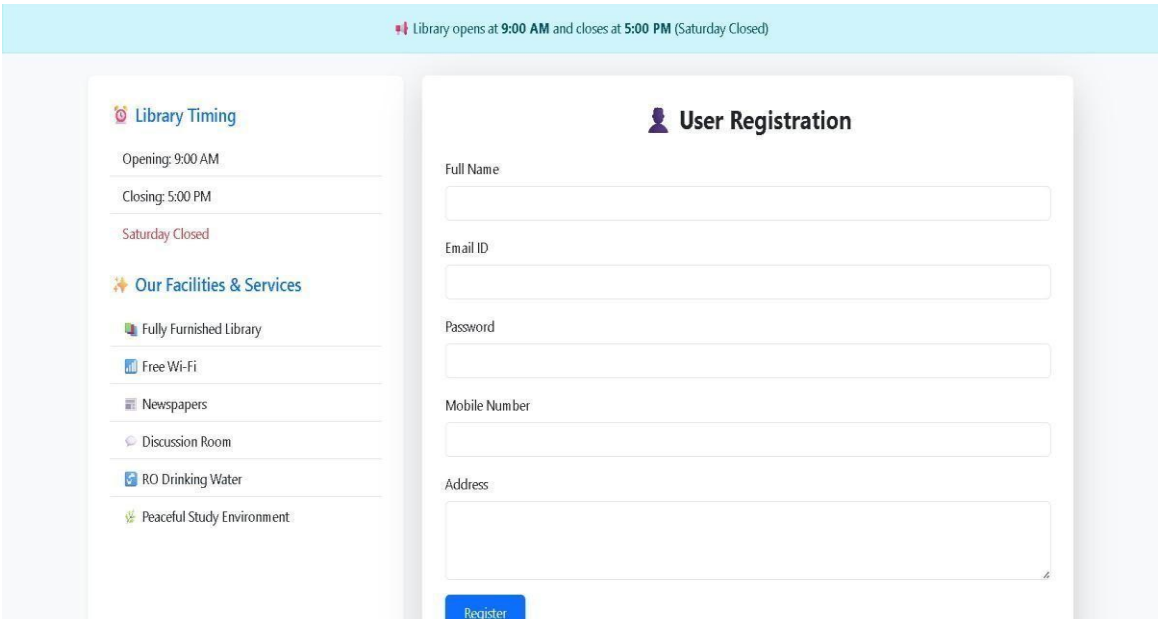

Figure CHAPTER 3: System Analysis and Design3.2.3.1: Register Page

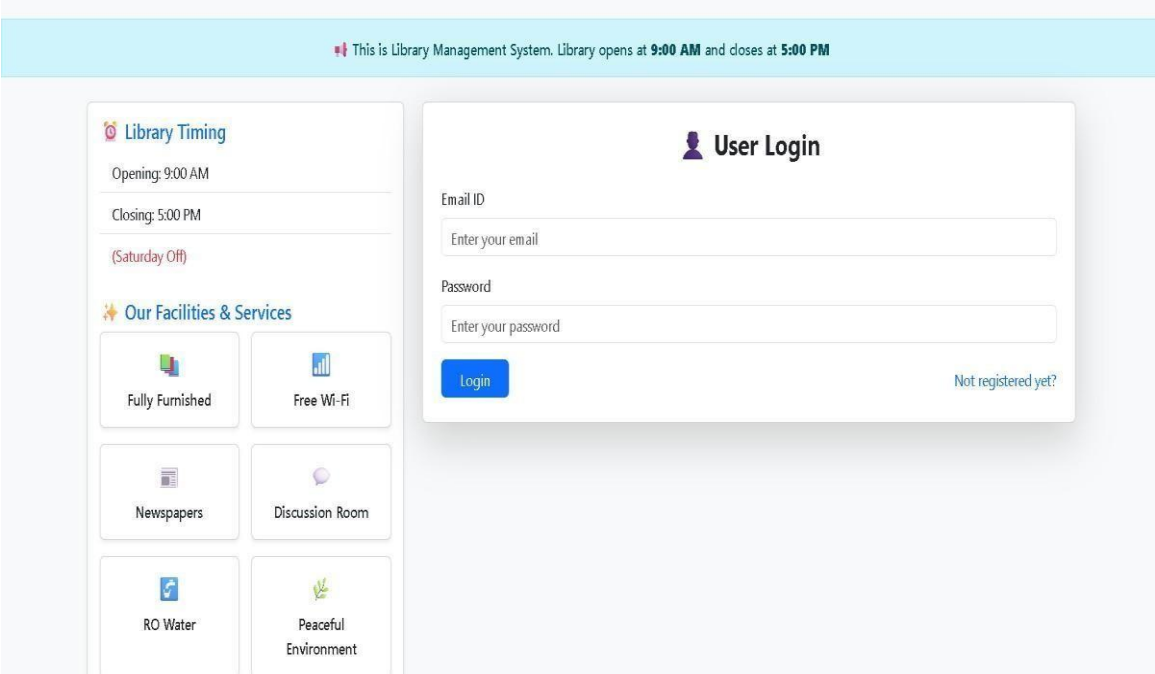

Figure CHAPTER 3: System Analysis and Design 3.2.3.2: Login Page

3.2.4 Physical DFD

A Physical DFD shows how data flows in a system, focusing on the physical implementation, such as hardware, software, databases, and user interactions. It emphasizes the how and where of data processing, unlike a logical DFD that focuses on the what.

- The **User** (student or library member) interacts with the system by performing actions such as Login/Register, View Book List, Issue Book, Return Book, and View Issue Status.

- These user requests are handled by the Frontend, which is developed using HTML, CSS, and JavaScript. The frontend provides the user interface and collects input from users.

- The frontend forwards the requests to the Backend, which is implemented using PHP. The backend processes the requests, validates user actions, manages book issues and returns, and applies business rules.

- The Backend communicates with the MySQL Database to store, retrieve, and update data, including user details, book records, issue/return transactions, and fine information.

- After processing the requests, the backend sends responses back to the frontend, which are then displayed to the user.

System Analysis and Design.2.4: Physical DFD

CHAPTER 4:Implementation & Testing

4.1 Implementation

In this phase, the theoretical design is transformed into a functional system. With the design phase complete, the focus now shifts to technical implementation. This involves coding the application based on the system's requirements and specifications.

4.1.1 Tools Used

We use different tools and techniques for the development of the system. We use HTML,CSS and JAVASCRIPT for the frontend and PHP for the backend development .We use for managing databases.

4.1.1.1 Programming Languages

PHP: Used for developing the backend with MySQL.

JavaScript: Used for building and logic in frontend

HTML & CSS: Used for structuring and styling the frontend.

4.1.1.2 Database Platforms

MySQL: Serves as the primary database for storing user data, books and related information.

4.1.2 Implementation Details of Modules

The implementation of design modules in a Library Management System (LMS) involves converting the planned system architecture into functional components using appropriate technologies. Each module is developed separately based on its responsibility and then integrated to form the complete system.

4.1.2.1Register Module:

The user needs to provide their first name, last name, email, phone number, password, confirm password for registration. These details will be stored in the database

4.1.2.2 Login Module:

For login user will input their email and password.

Admin will provide their admin id and password which will compared with a database content.

4.1.2.3 Book Issue Module:

Users can view the **list of available books** in the library.Book details such as title, author, category, and availability are managed by the admin.Users can select a book and submit a request to issue the book.The system updates the book status once the book is issued.

4.1.2.4 Fine / Payment Module:

Users can view fine details if a book is returned after the due date.Fine amount is calculated automatically by the system based on delay.Users can make the fine payment using available payment details (if implemented).After successful payment, the user receives a payment confirmation message.

4.1.2.5 Logout Module:

The system allows users to log out securely after completing their activities.The system also allows the admin to log out, ensuring secure access control.

4.2 Testing

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an undiscovered error. The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design are used.

4.2.1 Test Cases for Unit Testing

Unit Test focuses on verifying the correctness of individual components or units of code, such as a single function, method, or class. They are performed early in the development process to catch bugs at the foundational level and provide developers with immediate feedback on their code.

Table 4.2.1: Test Case for Login

Test cases no	Test Case	Expected results	Status
1	Logging into website	Email and password provided correct	Successful
2	Logging into website	Email incorrect	Unsuccessful
3	Logging into website	Password Incorrect	Unsuccessful

4	Logging into website	Any field left empty	Unsuccessful

Result Analysis for Login Test Cases

The login system was tested with different scenarios to ensure proper authentication and error handling. The observations are as follows:

a) Successful Login:

When the correct email and password were provided, the login was successful.This confirms that the system correctly authenticates valid user credentials.

b) **Unsuccessful Login Cases:**

- Incorrect Email: If the email was incorrect, the login attempt failed.
- Incorrect Password: If the password was incorrect, the login attempt failed.
- Empty Fields: If any field (email or password) was left blank, the system did not allow login.

These failed attempts confirm that the system has proper validation to prevent unauthorized access.

Table 4.2.1.1: Test Case for Signup

Test cases no	Test Case	Expected results	Status
1	Registration for new user	All details provided correctly	Successful
2	Registration for new user	Any one field is incorrect	Unsuccessful

3	Registration for new user	Any field left empty	Unsuccessful

Result Analysis for Signup Test Cases:

- If a new user provided all required details correctly, the registration was successful.
- If any field was left empty or incorrect, the system flagged the issue and prevented signup.
- This ensures that only valid users with complete information can register.

4.2.2 Test Cases for System Testing

System Test evaluates the entire system or application, including all integrated components and interactions. They are performed after integration and unit testing to confirm that the complete system works seamlessly and delivers the expected user experience.

Table 4.2.2: System Test Cases

Test Case No	Test Case	Test Data	Expected Outcome	Result
1	ADD books	Admin Added books	Books Added Successfully	Pass
2	Update Car Price	Admin Updated Books Prices	Price of Books Updated Successfully	Pass

3	Delete a User	Admin Deleted User	The user has been removed.	Pass
4	Issues	Admin isssue the Books	Book issues Successfully	Pass
5	Admin Login with Valid Credentials	Email: admin@example.com, Password: Abc123!	Admin is logged in successfully	Pass
6	Delete Book	Admin Deleted book	Book Deleted Successfully	Pass
7	Log in with valid credentials	Email: test@example.com, Password: StrongPass123!	User is logged in successfully.	Pass
8	Log in with invalid credentials	Email: test@example.com, Password: Wrong Pass	Error message "Invalid credentials".	Pass

4.3 Result Analysis for System Test Cases

The system test cases were executed to evaluate the functionality of the Library Management System. The following observations were recorded:

a) Book Management:

The system successfully allowed the admin to add, update, and delete book records, confirming that the book management module works correctly.

b) User Management:

The admin was able to delete user accounts successfully, indicating that the system correctly processes user management operations.

c) Book Issue and Return:

The admin was able to approve and manage book issue and return requests without any issues, ensuring that the transaction management module functions properly.

d) Login Functionality:

The system allowed authorized users (admin and regular users) to log in successfully. When incorrect login credentials were entered, the system displayed an “Invalid credentials” message, confirming that proper authentication and security checks are implemented.

All test cases were executed successfully, demonstrating that the Library Management System functions as intended. It effectively handles book management, user management, issue/return operations, and login authentication. The system is stable, reliable, and ready for deployment.

CHAPTER 5:Conclusion and Future Recommendations

5.1 Lesson Learnt

During the development of the Library Management System, several valuable lessons related to full-stack web development were learned, starting from system planning to deployment. The backend was developed using PHP, with a strong focus on user authentication, database connectivity, and data management. Implementing features such as book management, book issue and return, and fine handling improved understanding of how to manage data flow and user interactions efficiently.

On the frontend, HTML and CSS were used to design a clean, responsive, and user-friendly interface. The use of prebuilt components helped speed up development while still allowing customization according to system requirements.

Version control using GitHub helped track changes in the code and manage updates efficiently. Overall, the project enhanced practical knowledge of web technologies, database design, and system integration.

5.2 Conclusion

The Library Management System is designed to automate and simplify the process of managing library operations. This application allows users to register, search for books, issue and return books, and view their transaction history through an online platform. It provides a convenient and efficient way for users to access library services without relying on manual processes.

In the traditional manual system, managing book records, tracking issued books, and maintaining user details required significant time and effort. Checking book availability and updating records manually often resulted in errors and inefficiencies.

The main objective of this project was to digitize library management tasks, including maintaining book records, managing user information, tracking book issues and returns, and handling fines. The system serves as a comprehensive solution that streamlines daily library operations. By storing all data in a centralized database, librarians can easily access and manage information, leading to improved efficiency and better decision-making. Overall, the Library Management System is a reliable and effective solution that enhances library services and user experience.

5.3 Future Recommendations

Although the project successfully fulfills its core objectives, several enhancements can be made in the future to improve functionality and scalability:

Limited Language Support: The current version supports a limited set of programming technologies; additional technologies and frameworks can be integrated in future versions.

Profile Update Restriction: Users are not allowed to update their profile information after registration. Allowing profile updates would improve flexibility and user satisfaction.

Single Admin Role: The system currently supports only one admin role. Future improvements can include multiple admin roles with role-based access control.

Online Fine Payment: Integration of secure online payment gateways for fine payments can enhance usability.

Advanced Search and Reports: Adding advanced search filters and automated report generation can further improve system efficiency.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: State-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, May 2010, doi: 10.1007/s13174-010-0007-6.

[2] A. Kumar and R. Singh, “Development of a web-based library management system,” *International Journal of Computer Applications*, vol. 182, no. 45, pp. 12–18, Jan. 2023.

[3] S. Patel and M. Sharma, “Design and implementation of an automated library management system,” *International Journal of Advanced Research in Computer Science*, vol. 11, no. 2, pp. 45–52, 2022.

[4] T. Tuyambaze, E. Mkoba, B. Mgawe, and B. Mbunda, “Development of a cloudintegrated information system for smart library services,” Apr. 30, 2024, doi: 10.21203/rs.3.rs-4300397/v1.

[5] N. García-Moreno, P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil, “Building a blockchain-based decentralized library management system,” 2022 IEEE International Conference on Smart Systems.