**“Library Management System”**

**Author**

**Sachin Pundir & Sri Sai Venkata Ramana Anish Vissapragada**

**Supervisor**

**Taimoor Khan**

**Table of Contents**

[**Abstract** 3](#_Toc174489688)

[**Introduction** 3](#_Toc174489689)

[**1. 1 Background** 3](#_Toc174489690)

[**1. 2 Problem Statement** 3](#_Toc174489691)

[**1. 3 Objectives** 4](#_Toc174489692)

[**1.4 Deliverables** 4](#_Toc174489693)

[**Literature Review** 4](#_Toc174489694)

[**2. 1 Summary of Current Systems** 4](#_Toc174489695)

[**2. 2 Technical Background** 5](#_Toc174489696)

[**2.3. Case Studies** 5](#_Toc174489697)

[**2.4 Implications for Modern Libraries** 5](#_Toc174489698)

[**Implementation** 6](#_Toc174489699)

[**3. 1 Technological Stack** 6](#_Toc174489700)

[**3. 2 Development Process** 7](#_Toc174489701)

[**Testing and Evaluation** 11](#_Toc174489702)

[**4.1 Testing Methodology** 11](#_Toc174489703)

[**4.2 Test Results** 11](#_Toc174489704)

[**4.3 User Feedback** 11](#_Toc174489705)

[**4. 4 Evaluation** 12](#_Toc174489706)

[**Results** 12](#_Toc174489707)

[**Conclusion** 13](#_Toc174489708)

[**Future Work** 14](#_Toc174489709)

[**References** 14](#_Toc174489710)

# **Abstract**

This paper looks into the establishment of a Library Management System (LMS) to help in the book stocking by the administrator and the users' borrowing and returning of books. The system purports to automate the library processes, thereby improving the system's efficiency, client satisfaction while officing, and better data management. Therefore, understanding the history of library management, the system's parts, and how it works can help appreciate the advantages and disadvantages of an LMS. Also, it offers overview into the future of trends for improvement and suggestions for enhancing library management.

# **Introduction**

## **1. 1 Background**

In this respect, modern libraries have played crucial social roles during the centuries, from the collections of books as depositary of knowledge and culture up to highly effective and developed management systems. Books in the early forms of libraries were managed manually, and the processes involved in cataloguing the books, borrowing and returning the books took much time with many errors. Computer technology brought significant changes to the management of libraries, mainly through computerized cataloguing systems, automated borrowing and online database systems.

Modern Library Resources and Information Management Systems, commonly known as LMSs, are complete solutions for library needs and sources, information retrieval to the general populace, and organizational work. Concerning technological advancement, this report will specifically describe the procedures of developing and incorporating the LMS capable of giving the administrators the authority to add books to the inventory; at the same time, the users can borrow and return the books effortlessly. Thus, realizing the proposed system based on modern technologies promises to improve operational and user qualities.

### **1. 2 Problem Statement**

Past library management systems draw out problems like, for instance, mistakes made physically, challenging tasks involving the tracking of books, poor utilization of stock and no real-time data. These limitations require designing an automated system to overcome these difficulties and offer an efficient solution for managing library resources.

### **1. 3 Objectives**

* Create a comprehensive function for managing library resources through a library management system.
* Improve the speed and quality of work in a library.
* Ensure the library staff and patrons have the necessary and easy-to-navigate interface.
* Allow monitoring and controlling of the library resources in a real-time fashion.

### **1.4 Deliverables**

* Library Management System Software
* Technical Documentation
* Testing Reports
* User Training Materials
* Final Project Report

When carrying out this project, all of the above deliverables will guarantee that the developed system is powerful and useful enough to serve the library's and its users' needs.

# **Literature Review**

## **2. 1 Summary of Current Systems**

LMS adapted from how it was in the past to what is being developed nowadays, and there are many systems currently in the market for Libraries of different types, such as academic, public and special libraries. Some popular systems have been identified as Koha, Evergreen, and Alma. For example, we have Koha, an LMS software often used because of its open-source policy, flexibility and affordable pricing. These include the cataloguing, circulation and patron modules. However, like other systems, Koha has its challenges; they include that it is hard to set up and maintain as it needs technical personnel (Grammenis and Mourikis, 2018). Another open-source system with powerful cataloguing/circulation is Evergreen; however, like many systems, it needs more friendliness to the user and customization (Wang, 2009). Unlike other free LMSs, Alma provides clients with acquisition, discovery, and analysis package services. However, Alma comes with a high cost and is not easily affordable by most libraries, especially the small ones, and this usually causes a challenge to the users who find it complex to use (Jaganbabu, 2023). Some of the challenges that are seen with these systems include the high cost of implementation as well as maintenance and the need for training to be conducted to ensure occupants of the system understand how it works; the system may also prove to be difficult to scale as well as integrate with other forms of digital resources (Kasauli *et al.*, 2021). Also, the interfaces used may need to be more friendly; users will not be fully satisfied. Hence, the adoption will be low.

### **2. 2 Technical Background**

Most of the Modern Library Management Systems incorporate a range of technologies in order to offer their functionalities. Most systems employ MySQL or PostgreSQL, which are relational databases for metadata, users' data, and transaction data. Languages generally used on the server side for programming involve PHP, Java and Python (Srivani, 2024). The coding is in server-side scripting languages for processing and handling back-end logic and interactions with the database. Integrating digital resources frequently involves using APIs (Application Programming Interfaces), by which the LMS establishes a link to other systems, such as digital libraries and online databases.

### **2.3. Case Studies**

The university library installed a contemporary Library Management System (LMS) system to improve operation and documentation. This exercise incorporated a needs analysis, choosing an expansive LMS, configuration and data digitization. The organization implemented the system in stages, with the pilot stage being the first. However, the new LMS introduced problems like data migration matters, users’ resistance, and cyber tech troubles; however, the advantages of the simplified process, improved user satisfaction, and powerful reporting systems were achieved (Muhamad and Darwesh, 2020).

A huge public library has changed its management system and implemented a digital LMS. It consulted and involved the stakeholders and chose a cloud-based LMS that is interoperable with digital media and accessible through mobile. It introduced organizational activities such as tutorials and workshops. It also highlighted worries like what to do when integrating media from an electronic source, how to introduce IT knowledge to patrons needing to be more informed, and budget challenges (Ayo et al., 2023).

#### **2.4 Implications for Modern Libraries**

A contemporary Library Management System (LMS) enhances many aspects of library activities by improving users’ services, operations, Reporting, and analysis. This automates the borrowing and returning of materials and gives a clear and simpler display of books and accounts for easy administration by the users. This results in even further increases in a user’s level of satisfaction, thus increasing the level of interaction.

There is optimization of services; the repetitive functions used to take much of the librarians’ time are done by the system, thus freeing much of the librarians’ time for essential functions such as program development and social networking. Features like advanced reporting and analytics are handy as they help foster the understanding of the usage pattern, user behaviour and the use of resources, which helps in better planning resources and strategies.

Another benefit is that using the current LMS of the modern school can observe the flows of the resources, sparing the time and money on searching the lost or missing items and contributing to its perfect collection management. Updatability and documentation facilitate the observation of material conditions and the improvement of collection processes by librarians. These modern-day learning management systems can be changed to follow the dynamic nature of users and development in technology and hence can serve their function appropriately. It is also in line with supporting the long-term goals and strategies of libraries; thus, its capacity enables such institutions to meetto sustainably meet the various needs of society.

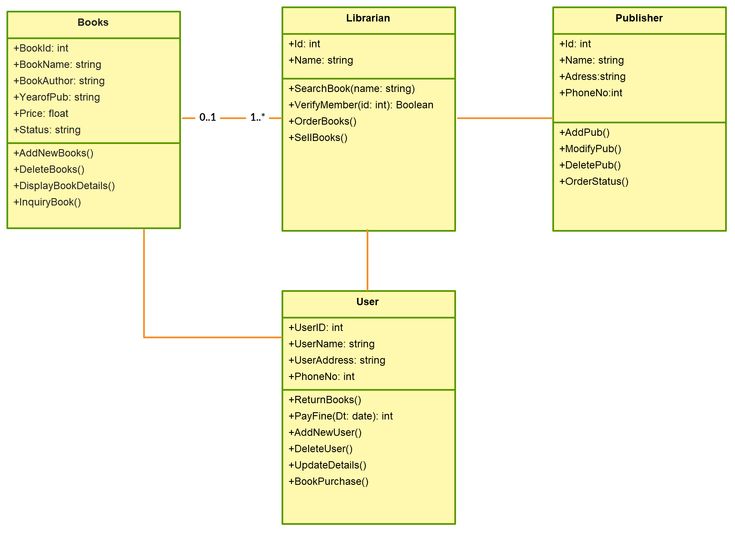
# **Implementation**

## **3. 1 Technological Stack**

**Programming Languages:** JavaScript is used for the frontend development, and the backend smart contract is written in the Solidity language.

**Frameworks and Libraries:** This concerns specific technologies, including React for the UI, Truffle for innovative contract development and testing, and Ganache for local blockchain development.

**ER Diagram:**



**Tools:** MetaMask for managing Ethereum accounts and making the transactions and Visual Studio Code as the pronounced integrated development environment (IDE) for writing and debugging the codes.

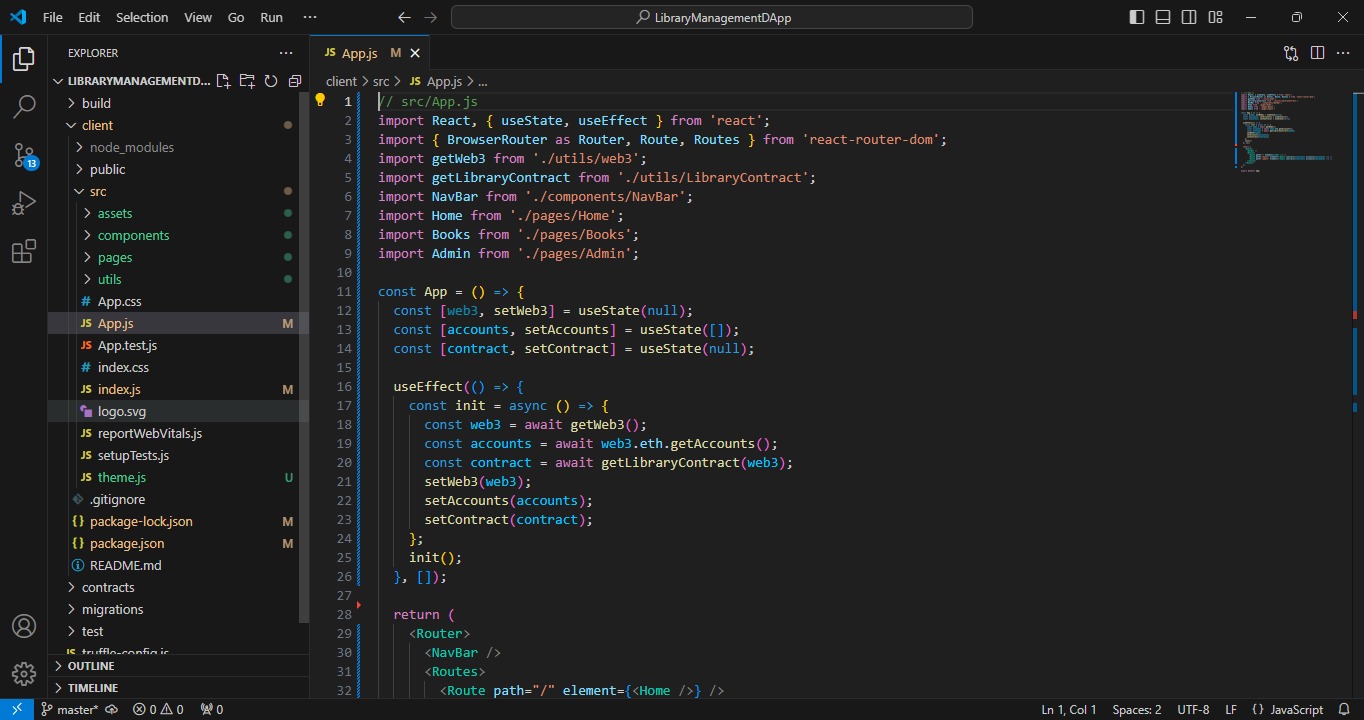
### **3. 2 Development Process**

**Planning:**

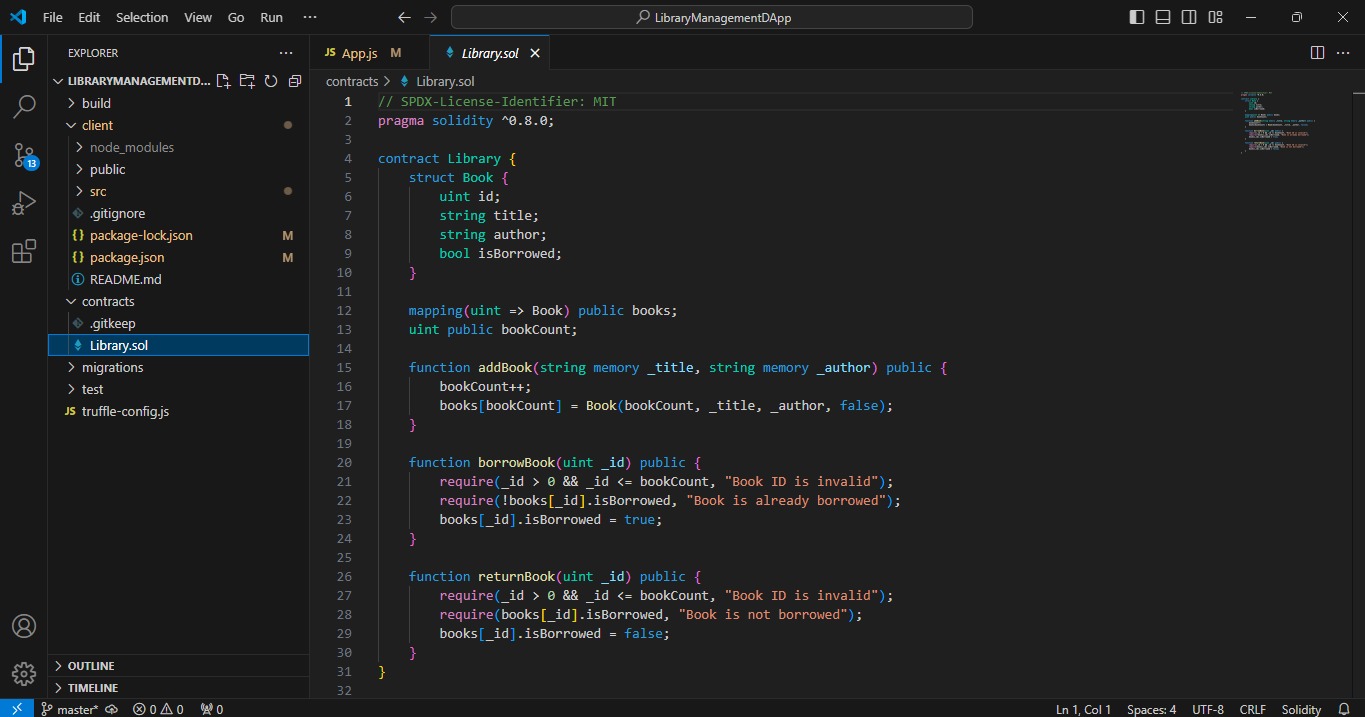
The subsequent stages of the development process were logically laid down with priorities in outlining specific objectives and constraints, specifications, as well as setting up predefined points of measure for upper limits and the chronogram of the project plan. Successful stages were the frontend interface implementation, creating and testing smart contracts, and integrating all the used technologies into a single system.

**Coding:**

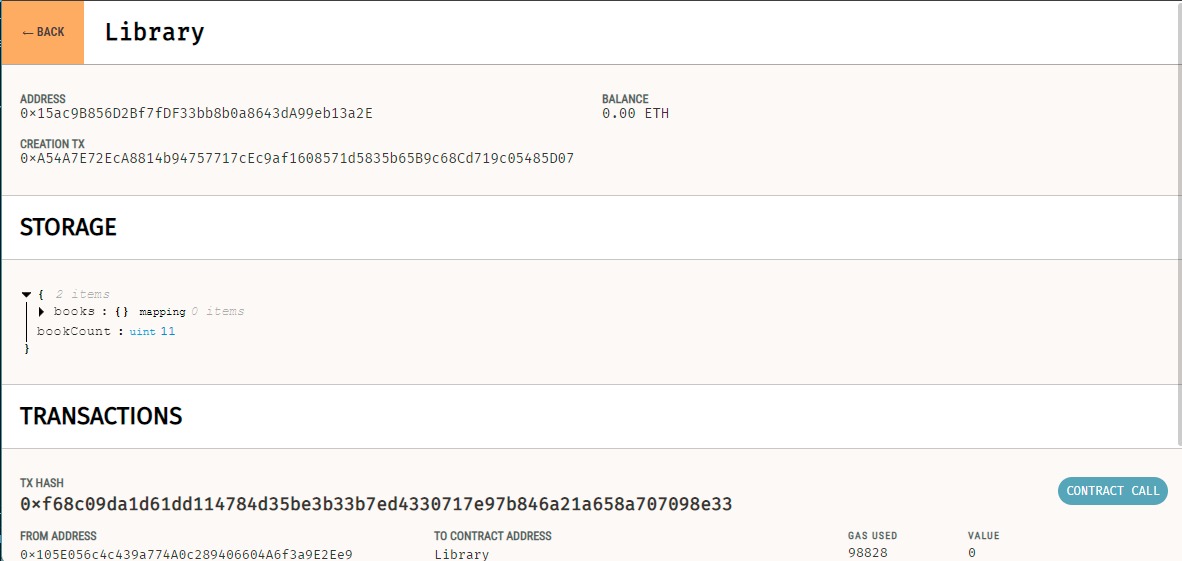
**Frontend Development:** The application's front end has a closely connected and well-considered UI created through React. This includes components for the book catalogue, user registration/subscriptions, and transactions.



**Smart Contract Development:** Smart contracts were coded in Solidity, and the system was tested with Truffle. All of these contracts dealt with book management and transaction recordation as the primary elements of the firm.

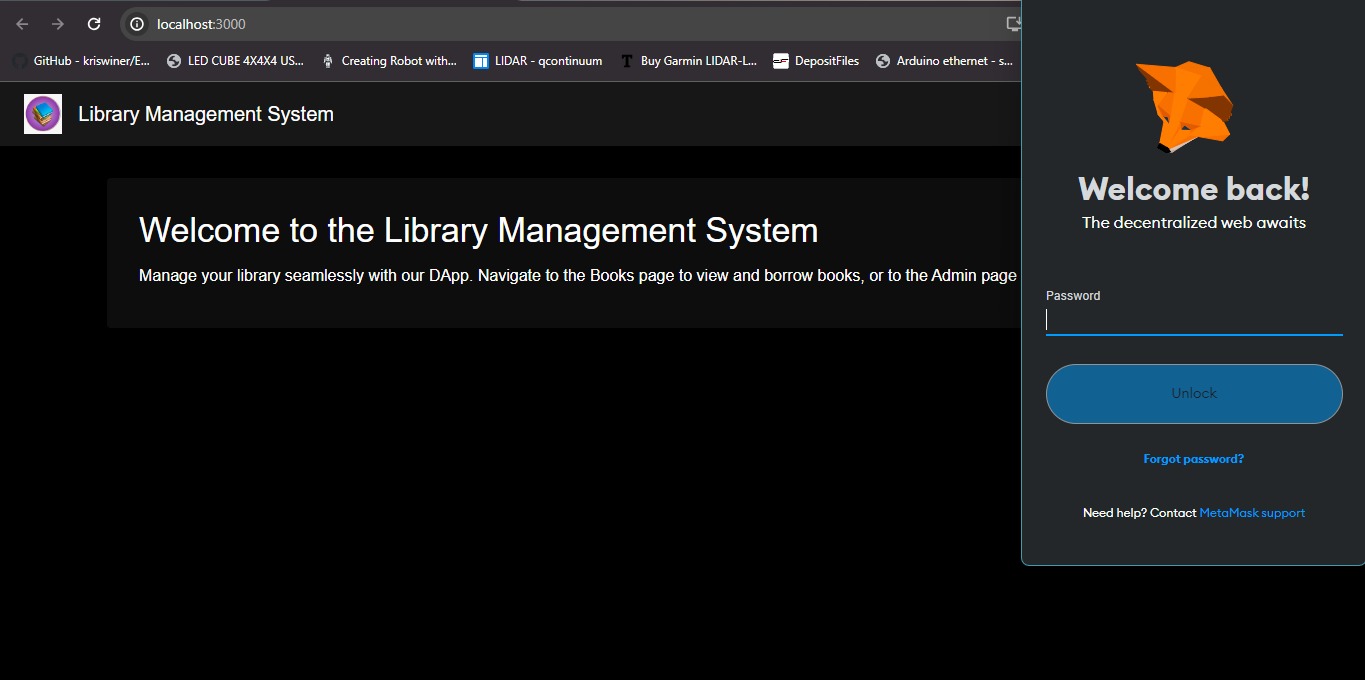


**Local Blockchain Simulation:** Almost all smart contracts and transactions were performed on a local Ethereum blockchain developed by Ganache.



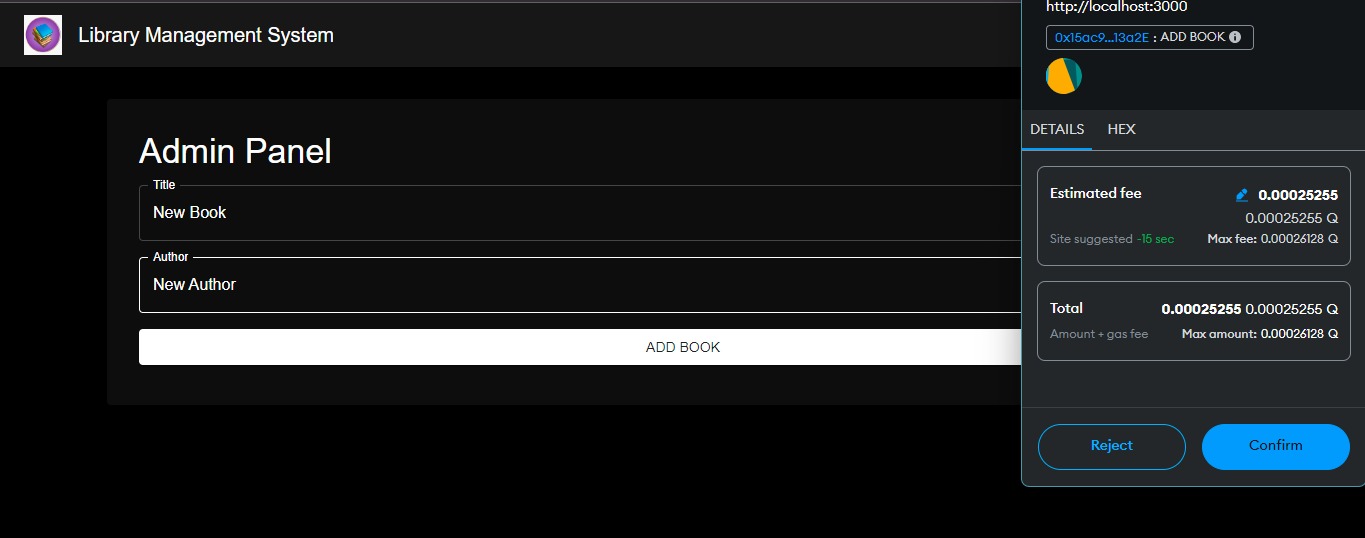


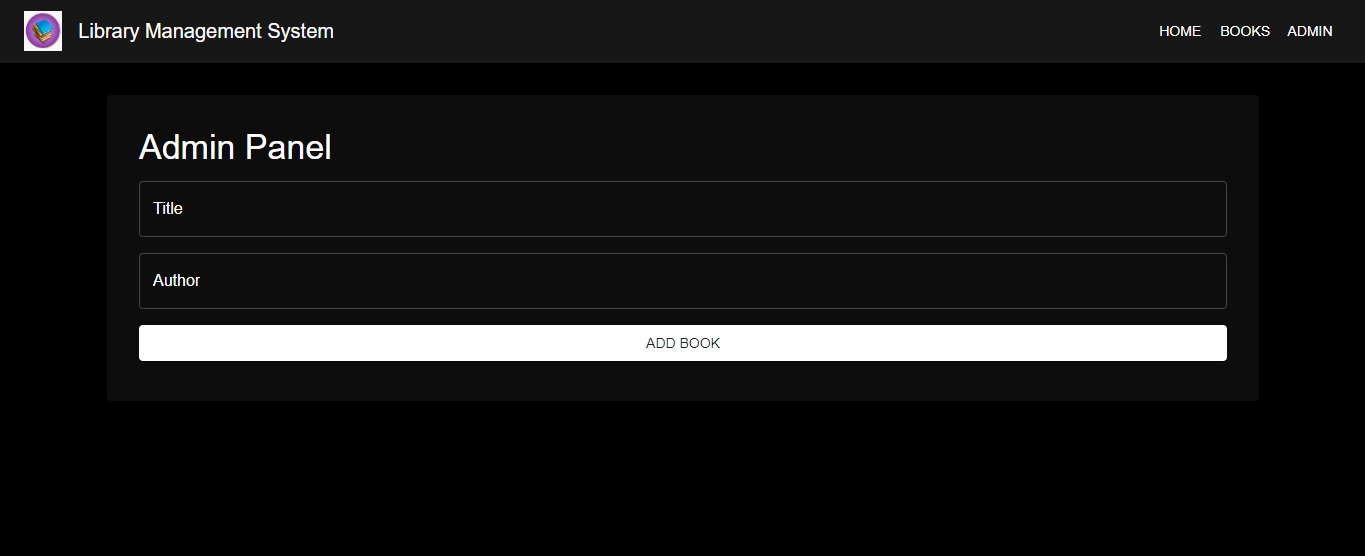
**Integration:** The frontend was combined with the backend using API and interaction with smart contracts through MetaMask. Some of these relate to the communication between React components and smart contracts and issues arising from transactions and their effects on the application's state.

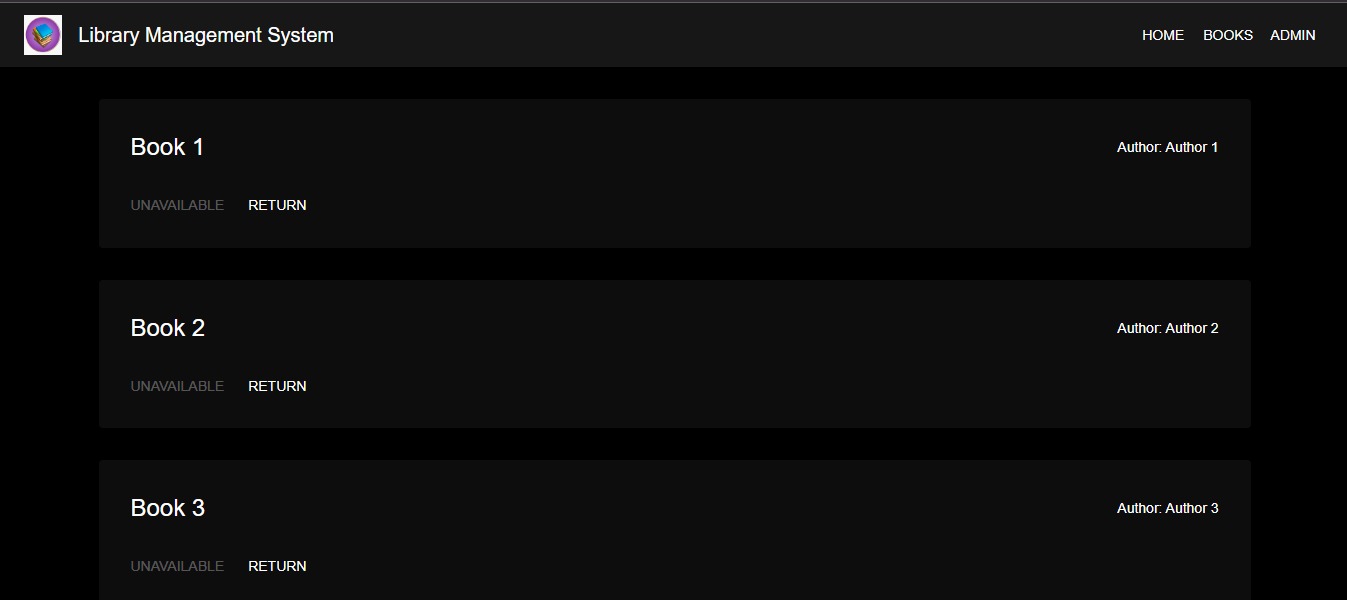


**Prototype:**

A prototype of the Library Management System demonstrates the core functionalities, including admin and user interfaces, catalog search, borrowing and returning processes, and reporting features. The prototype serves as a proof of concept, showcasing the potential benefits and usability of the system.







# **Testing and Evaluation**

## **4.1 Testing Methodology**

**Unit Tests:** The Truffle framework was employed to confirm the effectiveness of the individual smart contracts that made up the ${smart-contract-system-name}. The other test carried out was verifying the expected behavior of the React components.

**Integration Tests:** Integration testing was done to ensure that the communication between the React front and the smart contracts works in real life.

**System Tests:** Integration testing involves checking the UI, the ability of the smart contracts, and the app's interactions with blockchain.

### **4.2 Test Results**

The testing demonstrated that the system was functioning properly for most constructed scenarios. Smart contracts were working as intended, and the front end, as a consumer, engaged with the blockchain optimally. Certain concerns were noted regarding the transactions' procession and the UI; these were corrected through enhancement and code rectification.

### **4.3 User Feedback**

Analysis of the questionnaires showed that users could understand how the system worked and generally agreed that it was easy to use. Thus, everyday communication with users revealed that most benefited from the simplicity of book transactions' management and the library's information receipt. Some solutions provided to improve the system performance were improving the search functions available for the particular module and improving messages appearing to the user when transactions fail in the system.

### **4. 4 Evaluation**

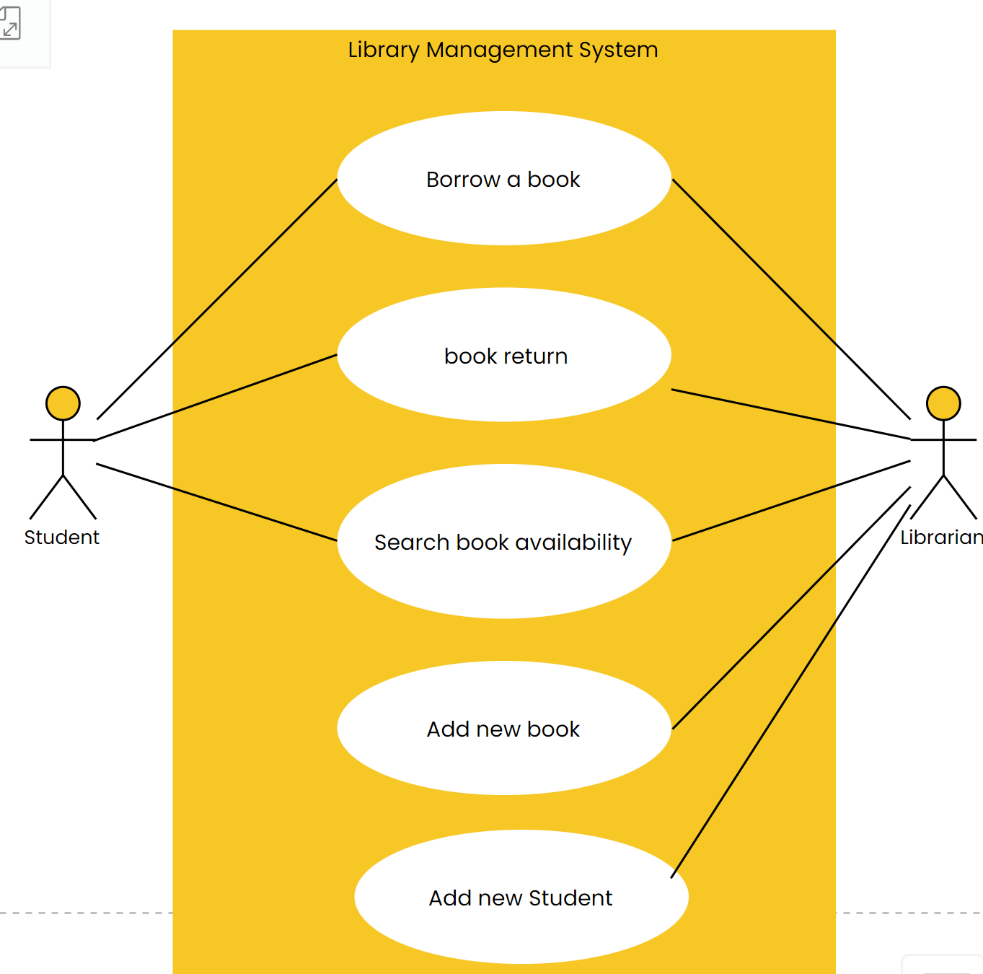
In this regard, the system achieved its intended goals and presented a tangible solution for managing a library. The front end and blockchain link demonstrated how the system integrates the front end and the blockchain, but some issues were also observed, as discussed above. Overall, the outcome improved operational productivity and user satisfaction, which is relevant to the project objectives of the work.

# **Results**

The library Management System achieved all the planned and expected outcomes, thus successfully automating book management and transaction processing. Integrating React with blockchain through smart contracts offered a sturdy solution for managing library operations.

The findings further supported the hypothesis that the system's design achieved the goals set within the project context, particularly with regard to the degree of effectiveness and satisfaction among system users. This position identified the enhancement of current and progressive technologies alongside the proficient implementation of the frontend and backend sections. Some of the weaknesses were minor downtimes and components of the design that could be improved for the user interface in the next versions of the application.

**Use Case:**



# **Conclusion**

In this context, implementing the Library Management System has significantly enhanced library functions by using Web technologies and advanced blockchain solutions. The system has enhanced major areas of the operation area, for instance, book management, processing of transactions and usability of users. The system significantly improves over traditional bibliographic organization strategies due to its capability for automation of routine tasks and the correspondingly efficient and streamlined process. React for a dynamic UI and employing smart contracts with a local blockchain emulator using Truffle and Ganache have made the functionality efficient. This technological advancement has not only brought modern development to the management and functioning of the library but also offered an adequate and safe user interface for handling library resources.

# **Future Work**

Therefore, the following are the recommendations that could be implemented for an improved version of the Library Management System in future: Increasing the size of the library and increasing the number of users are challenges that will require further system optimization and increasing the system’s capacity. Implementing additional features like a recommendation system based on AI can help enhance user experience by delivering customized and relevant book recommendations. Moreover, the potential of the system’s developments that have not yet incorporated such options as mobile access and real-time analytics will also improve the system’s practicality and efficiency.

# **References**

Ayo, E.B. *et al.* (2023) 'Development of an Integrated Library Management System (ILMS),' *International Journal of Interactive Mobile Technologies (iJIM)*, 17(10), pp. 242–256. <https://doi.org/10.3991/ijim.v17i10.37509>.

Grammenis, E. and Mourikis, A. (2018) *Migrating from integrated library systems to library services platforms: An exploratory qualitative study for the implications on academic libraries’ workflows*. <https://lnu.diva-portal.org/smash/get/diva2:1234039/FULLTEXT01.pdf>.

*Introduction to the server side - Learn web development | MDN* (2024). <https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction>.

Jaganbabu, J. (2023) 'INTEGRATED LIBRARY MANAGEMENT SYSTEM: a STUDY OF BOTH OPEN SOURCE AND COMMERICAL SOFTWARE,' *ResearchGate* [Preprint]. <https://www.researchgate.net/publication/376557298_INTEGRATED_LIBRARY_MANAGEMENT_SYSTEM_A_STUDY_OF_BOTH_OPEN_SOURCE_AND_COMMERICAL_SOFTWARE>.

Kasauli, R. *et al.* (2021) 'Requirements engineering challenges and practices in large-scale agile system development,' *Journal of Systems and Software*, 172, p. 110851. <https://doi.org/10.1016/j.jss.2020.110851>.

Muhamad, S.S. and Darwesh, A.M. (2020) 'Smart University library management system based on Internet of things,' *UHD Journal of Science and Technology*, 4(2), pp. 63–74. <https://doi.org/10.21928/uhdjst.v4n2y2020.pp63-74>.

Srivani, K.D. (2024) *Library Management System using MYSQL*. <https://www.analyticsvidhya.com/blog/2022/07/library-management-system-using-mysql/>.

Wang, Z. (2009) 'Integrated Library System (ILS) Challenges and Opportunities: A Survey of U.S. Academic Libraries with Migration Projects,' *The Journal of Academic Librarianship*, 35(3), pp. 207–220. <https://doi.org/10.1016/j.acalib.2009.03.024>.