Journal of Information Systems and Informatics
Vol. 7, No. 3, September 2025 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v7i3.1176 Published By DRPM-UBD

A Blockchain-Based Digital Library System Integrated with CryptoJS for Enhanced Security and Transparency

Abraham Eseoghene Evwiekpaefe[1]*, Darius Tienhua Chinyio[2], Fiyinfoluwa Ajakaiye[3], Paschal Obioma Aleke[4]

[1,2,3,4] Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria
Email: [1]aeevwiekpaefe@nda.edu.ng, [2]dtchinyio@nda.edu.ng, [3]fajakaiye@nda.edu.ng, [4]obynopascal2016@gmail.com

Abstract

In the context of digital library systems, blockchain presents a promising framework for enhancing the security, integrity, and transparency of operations such as book transactions, cataloging, and user authentication. Library systems face several challenges, including lack of transparency and security vulnerabilities. Previous research efforts have explored various centralized digital library management systems, but they often suffer from single points of failure and insufficient security measures. The methodology involves integrating blockchain technology using CryptoJS for advanced encryption and hashing, the backend was designed using PHP (Laravel), while the technologies used in the front end includes HTML, CSS and Javascript. The blockchain technology was implemented using Cryptojs which provides security by implementing AES encryption to safeguard user credentials and book transaction records, preventing unauthorized usage. The system was tested in a digital library environment and diverse user set, where results demonstrated enhanced data security and improved operational efficiency. The system is scalable and adaptable to academic, research, and public libraries, providing real-time verification of transactions and enhanced protection against unauthorized access. By combining blockchain's immutability with strong encryption and modern web technologies, the platform delivers a secure, transparent, and future-ready solution for digital library management with 88% effectiveness. Findings indicate that the proposed blockchain-integrated system not only resolves existing issues in digital library management, but also introduces new opportunities for innovation, including real-time transaction verification and improved trust among users.

Keywords: Blockchain, CryptoJs, Library System, Javascript, Database

1. INTRODUCTION

Personalized learning is increasingly recognized as an important aspect of modern education, allowing educational experiences to be channeled to the learning styles, Libraries have long been pivotal institutions for knowledge dissemination, offering access to a vast array of resources ranging from physical books to digital media. As libraries evolve to meet the demands of the digital age, they face numerous

challenges including data security, resource tracking, and user engagement. Conventional library management systems (LMS) often struggle with inefficiencies, vulnerability to data breaches, and limited transparency. These limitations hinder the ability of libraries to provide seamless and secure access to resources, thus necessitating a more robust solution.

Blockchain technology addresses challenges in conventional library systems by combining cryptographic hashing, distributed consensus, and smart contracts to create tamper-evident, transparent systems [1]. Unlike conventional relational databases (e.g., MySQL, PostgreSQL) or hybrid cloud solutions, blockchain's append-only ledger ensures data integrity without reliance on a central authority. Blockchain technology, initially developed as the backbone for cryptocurrencies like Bitcoin, has emerged as a transformative tool across various industries due to its decentralized, transparent, and immutable nature. By decentralizing data storage and utilizing cryptographic techniques, blockchain ensures that data is tamper-proof and verifiable. These characteristics make blockchain an ideal candidate for addressing the shortcomings of conventional LMS.

A blockchain-based library management system leverages these strengths to enhance security, transparency, and operational efficiency. Each transaction within the library, such as lending, returning, and reserving books, is recorded on a blockchain ledger, providing an immutable and transparent record of all activities. This decentralized approach eliminates single points of failure and reduces the risk of data breaches, as no single entity has control over the entire database [2]. Smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, play a crucial role in this system. They automate routine tasks such as tracking due dates, issuing fines, and updating inventory, thereby reducing administrative burdens and minimizing human errors [3]. This automation streamlines library operations, making them more efficient and reliable.

Furthermore, a blockchain-based LMS can incorporate a token-based incentive mechanism to promote user engagement and adherence to library policies. Users can earn tokens for activities such as timely book returns and participation in library events, which can be redeemed for various rewards. This gamification element encourages a more active and responsible user base, enhancing the overall library experience [4]. In this paper, we present the design and implementation of a blockchain-based library management system. Its architecture, components, and the integration of smart contracts and token-based incentives was explored. To validate the system's effectiveness, a case study was conducted in a university library setting, demonstrating significant improvements in operational efficiency, user satisfaction, and data security.

Recent advances in digital technologies have significantly transformed the landscape of Library Management Systems (LMS), yet many libraries continue to grapple with challenges such as inefficiencies, limited transparency, and data security concerns. Numerous studies have explored blockchain as a potential solution. [5] Applied blockchain to Nigerian university libraries to decentralize control and enhance access, although technical skill gaps among staff posed implementation challenges. [6] Improved access control and data security through a tamper-proof system, while [7]utilized immutable ledgers to improve transaction transparency albeit with high setup costs. [8] Introduced distributed ledgers and encryption for reliable record storage, although integration with legacy systems remained difficult.

Also, [9] focused on fraud prevention and verifiable record-keeping using blockchain, improving trust despite high costs. [10] Leveraged smart contracts for digital rights management, enhancing intellectual property protection but facing DRM integration issues. [11] Developed tamper-proof log storage, addressing unauthorized access but encountering scalability concerns. [12] Merged neural networks with blockchain for better resource allocation and data security, although implementation was complex. [13] Emphasized blockchain's value in addressing information sharing, preservation, and security, citing financial and training bardriers.

The study by [14], though largely theoretical, identified significant challenges in implementing blockchain-based library systems, namely, high costs and difficulties integrating with existing infrastructure. These issues highlight the need for a more simplified, cost-effective, and adaptable blockchain solution. While few practical implementations exist, [15] stands out. Their system improved data integrity and transparency using BigchainDB connected via API,but this approach introduced complex integration processes, scalability concerns, and user adoption challenges. Their study further revealed that the lack of intuitive system design hindered smooth transition for both users and administrators.

In response to these gaps, the current study proposes a more flexible and user-friendly approach by leveraging CryptoJS without relying on external APIs. This eliminates much of the integration complexity and enhances scalability across various LAN-based digital library systems. Moreover, adopting a LAN-based model significantly reduces implementation costs and technical barriers, making blockchain more accessible to academic libraries.

This solution implements a blockchain-integrated library management system that utilizes CryptoJS for AES-based encryption and hashing. The system aims to mitigate traditional challenges by securing digital book transactions, authenticating users, and maintaining a tamper-proof catalog. It is developed using PHP (Laravel) for the backend and HTML, CSS, and JavaScript for the frontend. The study

specifically investigates how the system enhances operational efficiency, strengthens security, and increases user trust compared to traditional and semi-decentralized digital library platforms. Therefore, the objective of the paper is to evaluate the applicability of the system in diverse library contexts; academic, public, and special libraries where digital integrity, trust, and transparency are paramount.

2. METHODS

This section presents the methodology for designing and implementing a blockchain-based library system operating within a Local Area Network (LAN). It highlights the architecture of this system which is designed to ensure easier integration of blockchain technology, security, transparency, and efficient management of library resources such as user records, book inventories, and transaction logs. The design of the system flow is as shown in Figure 1.

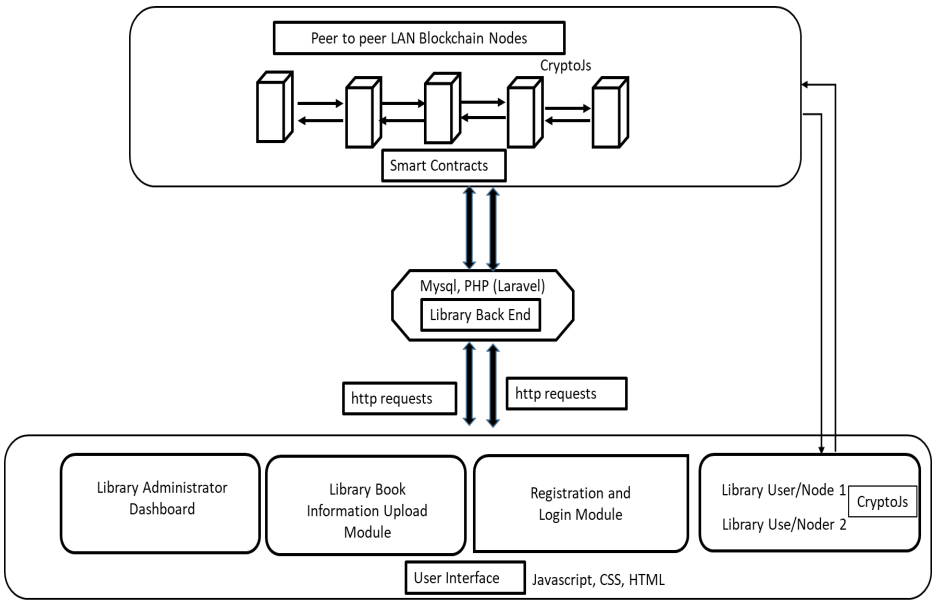

Figure 1. Schematics of the system process flow

2.1. System Components

The system component in this study consisted of many modules, described as follows.

1) User Interface: This is the interface where all users can register and login, the admin, students and institutional staff can see all the features of the system they can use.
2) Library Administrator: The library administrator module is the module designed for the administrator of the system to be able to upload books

and manage the library activities. Book borrowing and users activities on the system are authenticated via blockchain-based identity management. User credentials and activity are securely stored and managed on the blockchain. The admin adds new books, and approves book lending and returning.

3) Library User: This are the users of the library resources, ranges from students and staff of the institution.
4) Validated Consensus Response: The response from the blockchain after all the nodes have processed and confirmed the request from admin/user.
5) Library Backend: This is the component of the system responsible for sending admin/users request to the blockchain and also sends the blockchain response to the system user.
6) Blockchain Nodes: These are different devices or processing units in the blockchain that work together to ensure that only legitimate requests from within the system are processed.
7) Smart Contract: Smart contract is used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss.
8) User Satisfaction: Collect feedback from users regarding the usability and reliability of the system.

2.2. Blockchain Node Configuration

A private blockchain network is set up with multiple nodes to simulate a blockchain based library network. Each node represents a stakeholder (user node, admin, or archive server). Nodes are initialized using specialized Cryptojs functions. Nodes are responsible for: (1) Storing copies of the ledger, (2) Validating transactions, (3) Participating in consensus. The consensus algorithm is adopted due to its efficiency and suitability for permissioned environments like academic institutions.

2.3. Blockchain Node Architecture and Component Interaction

In this research, a private permissioned blockchain network was deployed to manage and secure digital library operations. The system consists of multiple blockchain nodes, each representing different stakeholders, such as the library server, librarian and student/client systems. Each node operates independently but participates in maintaining the distributed ledger through synchronized transactions. The core library components such as the catalog database, user registration, borrow/return logs, and digital resource access interact with the blockchain layer through smart contract interfaces. For instance, when a student borrows or accesses a resource, the action is validated by a smart contract and then written to the blockchain, ensuring traceability and immutability. These smart contracts define the rules for transaction validation and data handling.

2.4. Integration of CryptoJS with Blockchain Layer

The CryptoJS library was integrated on the client side to ensure validation of sensitive data and smooth implementation of the blockchain layer. User credentials, digital document hashes (library resources), and metadata are validated in the blockchain layer before being sent. Once validated, the data is packaged into http transaction request and submitted to the server. Before the data is sent to server, the blockchain section that receives the data confirms it via smart contracts and stores the hash on-chain. This combination of client-side encryption (CryptoJS) and on-chain storage ensures that no plaintext sensitive data resides on the blockchain, thus enhancing security and compliance with data protection regulations.

2.5. Node Management and Consensus Mechanism

To ensure robust management of the blockchain nodes, individual client/students systems were used to simulate a multi-node environment during development and testing. Each node runs an instance of the blockchain protocol, communicates over secure channels, and stores a synchronized copy of the ledger. The consensus mechanism adopted in this system is developed using CryptoJs functions, which is suited for permissioned blockchains. It enables all nodes to agree on the order of transactions, even when some nodes may be faulty or act maliciously. This method ensures low latency and high throughput, which is essential for real-time digital library operations such as lending, verification, and audit.

2.6Research Flow and Blockchain Integration

The research flow of this paper is shown in Figure 2 and detailed explanation of the steps in the figure as follow.

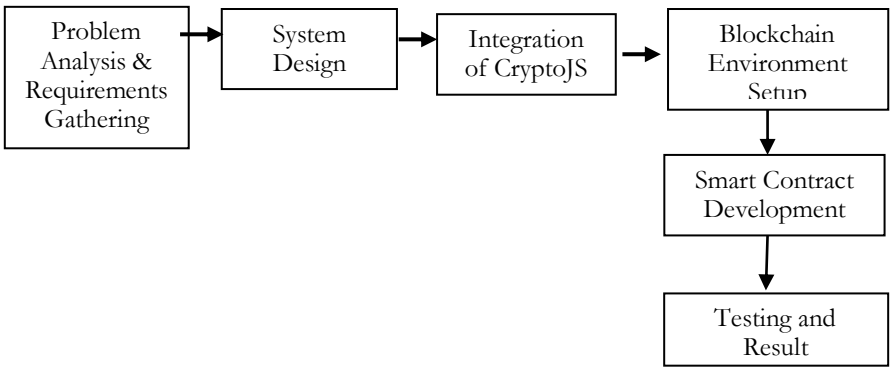

Figure 2. Research flow

Based on Figure 2 can be explained as follow.

1) Step 1 – Problem Analysis & Requirements Gathering

The first step involved analyzing limitations in traditional digital library systems: lack of tamper-proof logs, auditability, and secure user verification. Functional and non-functional requirements were defined in collaboration with librarians and academic IT staff.

2) Step 2 – System Design

A modular architecture was adopted. Frontend apps were developed in HTML, CSS and javascript, while the backend (Laravel) was designed to communicate with the blockchain layer. Smart contracts were defined using advanced features of CryptoJs.

3) Step 3 – Integration of CryptoJS

Before any sensitive data was transmitted, CryptoJS encrypted it on the client side. For instance, when a student uploads an academic certificate, the SHA256 hash of the file is calculated and encrypted, then submitted to the blockchain.

4) Step 4 – Blockchain Environment Setup

A multi-node private blockchain was deployed using CryptoJs framework and configured for consensus. Node identity, permission control, and peer discovery were handled using certificates and access lists.

5) Step 5 – Smart Contract Development

Smart contracts were written to handle student registration, book lending, resource access logs, and transcript issuance. These contracts enforce access control and data validation rules.

6) Step 6 – Testing and Result

Integration tests and user acceptance testing were run to validate the consistency of the ledger, effectiveness of the consensus protocol, and ability of the system to prevent unauthorized access.

2.7. System Requirements Specification

The **System Requirements Specification** in this study outlines both the hardware and software prerequisites necessary for the blockchain-based digital library system to function efficiently, as well as the steps required for its setup, usage, and testing. The class diagram, shown in Figure 3, illustrates the structure

of the system by depicting the primary database tables, their attributes, and the relationships between them. This diagram provides insight into how the system is organized and how various components, such as users, books, and transactions, interact with one another. Additionally, Figure 3 offers a detailed view of the system's static architecture, highlighting key relationships like inheritance, associations, and dependencies among the different classes.

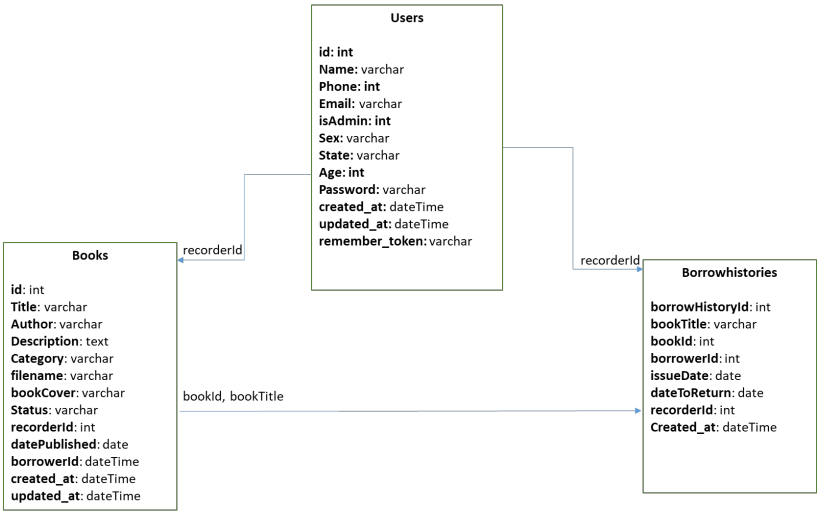

Figure 3. Class diagram showing main classes and entity relationships.

In addition, the system flow chart, as seen in Figure 4, visualizes the process flow within the blockchain-based library system. It outlines the key interactions between users and the system, including registration, login, and document issuance. The flow chart serves as an essential tool for understanding the sequence of steps in the system, ensuring that the workflow is efficient and that the blockchain framework operates seamlessly. For the hardware setup, the system requires a reliable computer or server capable of running the necessary software efficiently. Sufficient Random Access Memory (RAM) is essential for handling the processing demands, while a fast processor is necessary to manage the computational requirements of the blockchain network. Adequate storage space is needed to store blockchain data and related files, and a stable, high-speed Local Area Network (LAN) connection is vital to maintain uninterrupted communication within the blockchain network.

The software requirements for the system include using Windows (e.g., Windows 10 or 11) as the operating system. MySQL, a SQL database, is utilized for storing non-blockchain data, ensuring smooth integration with the blockchain. The back-end of the system is developed using Laravel, while Visual Studio Code serves as the Integrated Development Environment (IDE) for coding and project management. To execute server-side JavaScript code, Node.js is installed, and

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

front-end technologies such as HTML, CSS, and JavaScript are used to create the user interface. CryptoJS, a JavaScript library, is also used to implement cryptographic functions, such as AES, MD5, and SHA, to secure the data.

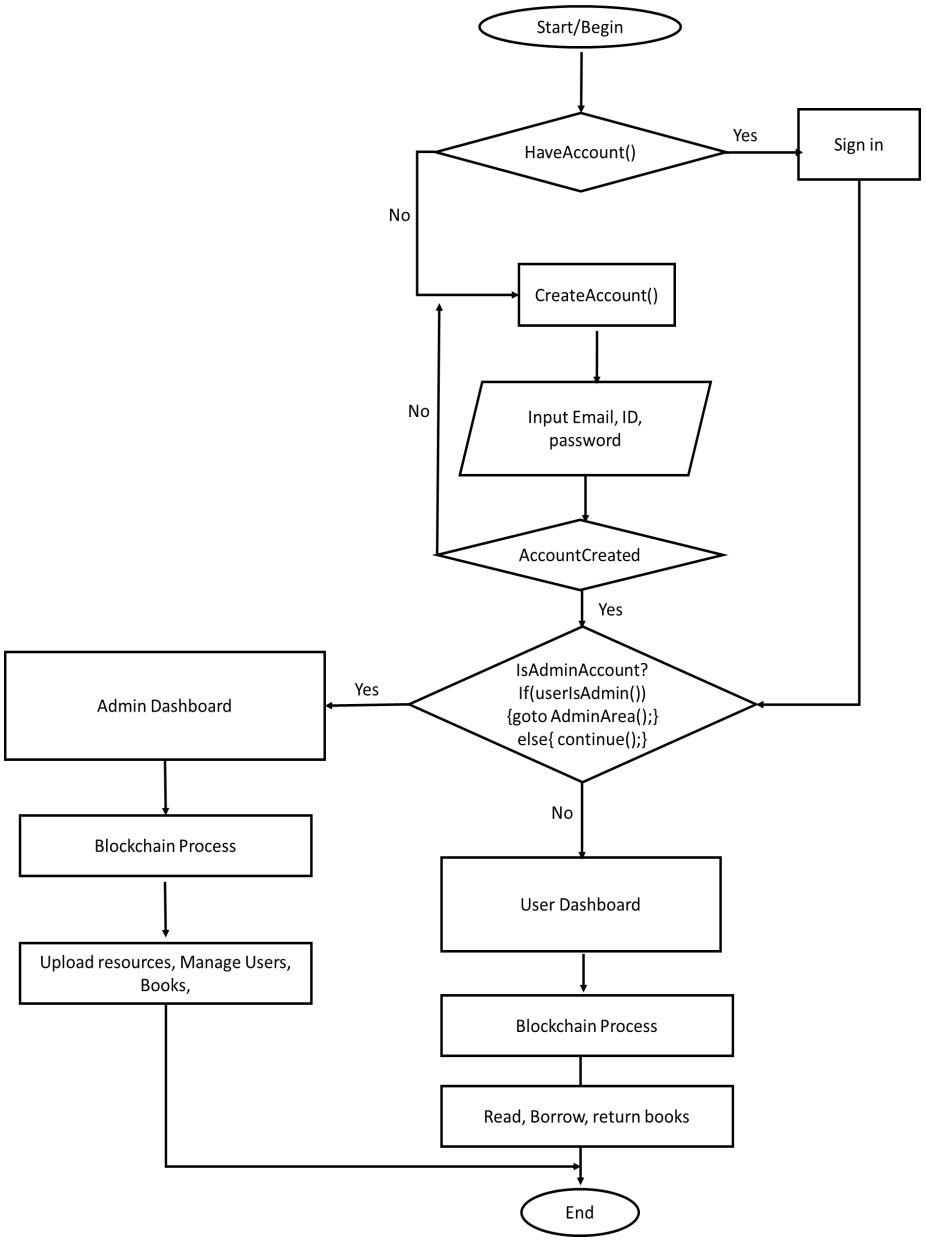

Figure 4. System flow chart for the blockchain-based academic library system

The setup procedure for the system involves several steps. First, the computer should be started, followed by setting up XAMPP on the server and activating the server. Then, the user can launch any installed browser and enter the address http://127.0.0.1:8000 in the address bar to access the system's login interface. Both students and system administrators will use the same login interface to authenticate their access. For the system usage procedure, users begin by entering http://localhost:8000 in their browser's address bar and logging in with their credentials. If the credentials are verified, they are directed to the homepage, where they can view available documents for borrowing or returning. There are two types of users: administrators and general users. Administrators can access the admin dashboard by entering http://localhost:8000/admin.

To ensure effective operation, training is provided to both administrators and users on how to navigate and utilize the system. This includes an introduction to the blockchain-based library system and an explanation of each module's functionalities. Users are required to install and open the application in their web browser before using the system. For testing, the system was deployed in a simulated academic library network environment. This setup replicated the operational conditions of a medium-sized university library. The system was hosted on a Local Area Network (LAN) with five interconnected user nodes, each representing a different library function, such as circulation, cataloging, and borrowing. This configuration ensured redundancy and fault tolerance, which are essential for maintaining stability and performance in the system. Finally, to assess system performance, a hybrid dataset was used. This dataset combined anonymized real-world book records from an academic library (2,000 entries) with synthetic user data (500 users), generated through Laravel's seeding feature. The synthetic data reflected realistic borrowing patterns, including book loans, returns, and reservations, allowing for a robust evaluation of the system's performance while maintaining the privacy of actual patrons.

3. RESULTS AND DISCUSSION

This chapter presents the results obtained from the implementation and evaluation of the Digital Library System integrated with Blockchain Technology and CryptoJS. The findings are analyzed based on system functionality, feature performance, and user acceptance. The discussion emphasizes how the integration of blockchain and cryptographic encryption enhances the efficiency, transparency, and security of the digital library system, ensuring a seamless and secure user experience.

3.1. System's Outputs

The system's outputs provide valuable insights into how its functionalities support both administrative and user operations. One of the primary outputs is the home

page, which serves as the main entry point for both users and admins. The homepage includes key sections such as About, Admin, and User areas, and functions as a secure and authenticated interface. Upon accessing the homepage, users and admins are required to log in. The admin login page ensures secure access through a unique username and password, preventing unauthorized access. This authentication process is critical in maintaining the integrity of the system, as it allows administrators to manage book records, handle user accounts, and oversee system activities. Figure 5 illustrates a screenshot of the login interface, providing a visual representation of the login process.

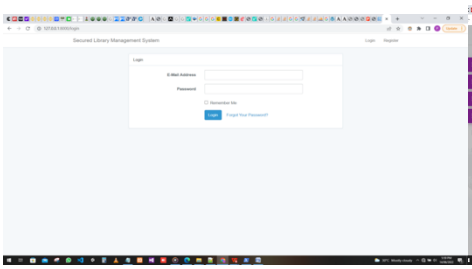

Figure 5. Login Page

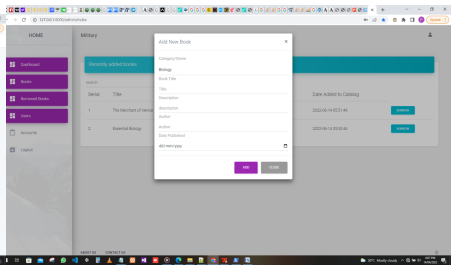

Figure 6. Book Page

One of the key functionalities of the system is the admin's ability to add books. Only authorized administrators have access to this feature, ensuring that the integrity of the library catalog is maintained. Regular users are restricted from adding or modifying book records. Once a book is added to the system, it becomes available for users to search, borrow, and read based on the library's policies. The process is further secured through CryptoJS encryption, which safeguards sensitive book records and prevents unauthorized access or data breaches. Figure 6 presents a screenshot of the "Add Books" page, showcasing the admin's interface for adding new books to the library catalog.

In addition to book management, the CryptoJS Blockchain Implementation plays a crucial role in securing library transactions. Blockchain technology, integrated with CryptoJS, ensures that each transaction—such as borrowing or returning books—is securely processed. Every transaction is hashed, and this hash is confirmed by the client node before the backend processing begins, ensuring that the transaction is both tamper-resistant and authentic. The use of CryptoJS functions makes it difficult to modify any transaction data without detection, providing a high level of data integrity. Figure 7 illustrates the blockchain blocks created during the system's operation, showing how transactions are recorded and verified on the blockchain.

The system database design plays a central role in organizing and storing essential information related to library operations. The database consists of several tables, each serving a specific purpose to ensure smooth interactions between users,

books, and transactions. The Books table stores critical details such as book title, author, ISBN, category, availability status, and the date the book was added to the system. The Users table contains user records, including user ID, name, email, username, password (encrypted using CryptoJS), and registration date. These tables are structured to allow for efficient data retrieval, updates, and security. Additionally, the Borrowing History table tracks all transactions related to book borrowing and returning, providing an audit trail for all actions taken within the system. Figure 8 offers a screenshot of the database design, displaying how the tables interact to manage the flow of information in the system. The results highlight the effectiveness of combining blockchain technology with CryptoJS encryption in securing and optimizing the digital library system. By enhancing data integrity, user authentication, and transaction security, the system demonstrates its potential to revolutionize the management of digital libraries, offering a transparent, efficient, and secure solution for users and administrators alike.

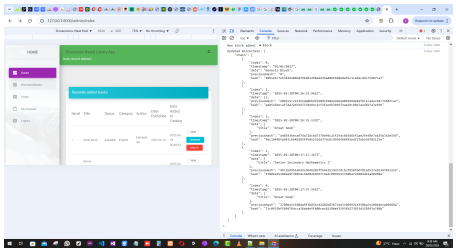

Figure 7. Blockchain blocks

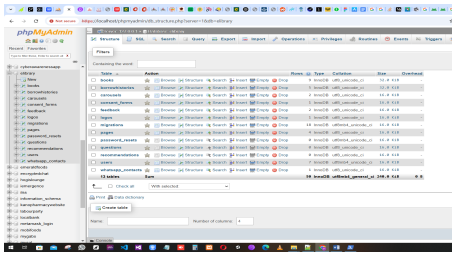

Figure 8. System database

3.2. System Testing

The User Acceptance Testing (UAT) was conducted to evaluate the system's performance from the user's perspective, ensuring that the system meets the expectations and requirements of its intended users. The system was designed to accommodate interactions from five distinct user groups: students, staff, and institution employees. Feedback was collected from all user groups via a survey conducted after they had engaged with the system. The results of the survey, which were based on practical usage and user experience, are summarized in Table 1.

Table 1. User acceptance survey

Feature Domain/SN	Evaluation Criteria	Score
User Registration		
1	Seamless user registration process	8
2	System provides clear error messages	8
Login Functionality		
3	Successful user login without issues	10
4	Informative error messages on login failure	8
Admin Control Features		
5	Admin can successfully add new books	9

Feature Domain/SN	Evaluation Criteria	Score
6	Admin has full control over book management	9
7	Admin can process book lending and returns	9
8	Admin has access to student and staff records	9
Book Borrowing and Returning		
9	Users can successfully borrow books	9
10	Users can return borrowed books smoothly	8
11	System maintains accurate borrowing history	8
Overall System Performance		
12	Users can navigate and utilize the system effectively	9
13	The interface is intuitive and user-friendly	9
14	The system is easy to operate	10
	Total	**123**

The total score for the survey was 123 out of a possible 140, resulting in an average system performance rating of 87.9% for the five users. This high rating reflects the system's user-friendliness, ease of use, and effective functionality, indicating that the system is performing well in terms of user expectations. Specifically, users found the login functionality, admin controls, and overall system interface to be intuitive, with minimal issues related to errors or navigation. The successful borrowing and returning of books, along with the maintenance of accurate borrowing history, were also positively rated, further demonstrating the system's capability to meet user needs efficiently. This feedback highlights the system's overall robustness, particularly in areas such as user interface design, book management, and transaction processing. However, some areas, such as the smoothness of returning borrowed books and maintaining an accurate borrowing history, received slightly lower scores, which may suggest areas for future improvement in system responsiveness and user experience.

3.3. Discussion

The integration of blockchain technology and CryptoJS into a digital library system represents an innovative approach to enhancing the security, transparency, and efficiency of library operations. Traditional digital library systems often rely on centralized databases where sensitive data, such as user credentials and transaction logs, are stored in plaintext or weakly encrypted. This exposes the system to risks like data breaches, unauthorized access, and other security vulnerabilities. By incorporating CryptoJS encryption, the system ensures that sensitive data, particularly user authentication credentials, are securely protected both at rest and during transmission. This approach adheres to best practices in data security, mitigating the risks of man-in-the-middle and brute-force attacks, which are prevalent in conventional systems.

From a functional perspective, blockchain's immutable ledger offers a significant improvement by introducing a tamper-evident mechanism for recording key transactions, such as book borrowing, returning, and user access history. This ensures that all actions are securely logged and cannot be altered or erased, providing a much higher level of accountability and auditability than traditional systems. In contrast, conventional systems often rely on centralized databases susceptible to log tampering or human error. The use of smart contracts within the blockchain further automates user privileges and access control, reducing the administrative workload and improving transaction efficiency. Unlike traditional role-based systems, smart contracts enforce access rules transparently and without the need for centralized oversight, offering more secure and automated management.

User feedback, gathered through User Acceptance Testing (UAT), indicates a high level of satisfaction with both the security features and the usability of the platform (as shown in Table 1). Respondents particularly appreciated the intuitive interface and the transparency provided by the blockchain ledger. However, some users required an initial acclimatization period to understand the blockchain-based interactions, particularly how transactions are validated and recorded. While this learning curve was steep for non-technical users, it was overcome with minimal support, suggesting that the system has strong usability potential for broader institutional adoption.

Despite its strengths, several limitations were identified. One key concern is the scalability of the system. As the number of users and transactions increases, the blockchain's performance may be affected by processing latency, especially in public or permissionless blockchain architectures. This can hinder real-time interactions, which are crucial in a dynamic library environment. Additionally, blockchain's immutability results in the accumulation of data over time, potentially leading to increased storage requirements. Unlike traditional databases that allow for archiving or purging data, the blockchain's continuous expansion may result in storage bloat unless off-chain storage strategies or pruning mechanisms are implemented.

Another challenge is the regulatory and compliance implications of storing user data on a decentralized ledger. While encryption addresses confidentiality concerns, strict data residency or privacy laws in certain jurisdictions may present obstacles to widespread adoption of blockchain-based systems. Furthermore, maintaining and upgrading smart contracts requires specialized expertise, and improper testing of these contracts could introduce significant risks to the system's stability and security.

Despite these challenges, the integration of blockchain and CryptoJS provides a notable improvement in the security and operational transparency of digital

libraries when compared to traditional models. By addressing vulnerabilities such as data tampering and unauthorized access, the system positions itself as a forward-thinking solution for academic and public libraries. Moving forward, efforts will focus on improving scalability through the implementation of Layer-2 solutions or permissioned blockchain architectures and enhancing user training to reduce onboarding friction, ensuring that the system becomes more accessible and efficient for a wider range of users.

4. CONCLUSION

This study presents the design and implementation of a digital library system that integrates blockchain technology and CryptoJS to enhance security, transparency, and operational efficiency. By combining blockchain's decentralized ledger with CryptoJS's cryptographic features, the system addresses key challenges such as data breaches, inefficient access control, and diminished user trust. Blockchain ensures there's no single point of failure, reinforcing the integrity of digital records, while CryptoJS provides secure storage and communication. The system adopts a LAN-based infrastructure, eliminating the need for costly third-party APIs, making it more accessible and cost-effective for academic institutions. Additionally, CryptoJS's encryption ensures compatibility across platforms. The design is scalable, supporting the management of large-scale library networks through modular blockchain node clusters and smart contracts to automate library operations based on dynamic rules. Future improvements include optimizing blockchain performance through better consensus algorithms, such as transitioning to Proof of Authority, and incorporating machine learning for predictive content delivery. This model offers a secure, scalable, and adaptable solution, particularly for institutions with limited resources or security concerns. With further enhancements, this system can transform academic information management.

REFERENCES

[1] N. Kshetri, "Globalization of innovations: A multilevel-multimethod framework to explain diffusion and adoption of the Internet," *Proc. University of Rhode Island*, Kingston, RI, 2024, doi: 10.23860/diss-1877.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," *in Proc. IEEE Big Data Congress*, 2017, pp. 557–564. doi: 10.1109/bigdatacongress.2017.85.

[3] A. Agbesi, A. Mensah, and S. Adjei, "Decentralized library management system using blockchain technology," *Int. J. Lib. Inf. Sci.*, vol. 17, no. 1, pp. 32–45, 2025. doi: 10.3389/jlis.2025.102.

[4] M. Swan, "Blockchain blueprint for a new economy," *J. Blockchain Econ.*, vol. 8, no. 4, pp. 25–39, 2015. doi: 10.2139/jbe.2015.11.

[5] T. Adebayo, J. Okoro, and M. Oduola, "Implementation of blockchain in Nigerian university libraries," *Afr. J. Lib. Inf. Sci.*, vol. 9, pp. 34–47, 2019.
[6] Y. Jeon and H. Kim, "Blockchain-based management of video surveillance systems," *J. Tech. Innov.*, vol. 11, no. 2, pp. 112–126, 2020. doi: 10.1016/j.jti.2020.11.006.
[7] E. Osagie, O. Eze, and S. Onoriode, "Blockchain for library management: A case study," *Lib. Manag. Today*, vol. 24, pp. 112–127, 2021.
[8] P. Ramachandran, K. Mehta, and N. Shah, "Blockchain technology for library systems," *J. Inf. Technol. Libr.*, vol. 30, pp. 165–182, 2021.
[9] A. Ojo, B. Adewale, and T. Lawal, "Blockchain in Nigerian academic libraries," *J. Niger. Lib. Inf. Sci.*, vol. 10, pp. 55–69, 2012.
[10] P. Talreja, R. Gupta, and A. Singh, "Blockchain for digital rights management," *J. Dig. Inf. Manag.*, vol. 17, pp. 83–97, 2019.
[11] R. Huang, "A blockchain-based framework for secure log storage," *in Proc. Int. Conf. Secur. Tech.*, 2023, pp. 210–218. doi: 10.1109/ICST.2023.2345679.
[12] R. Movassagh, A. Salehi, and M. Naseri, "Blockchain for library resource management," *J. Acad. Libr.*, vol. 26, pp. 378–392, 2021.
[13] R. Sharma and R. Batth, "Blockchain technologies in library services," *Emerald Insight J.*, vol. 36, pp. 401–418, 2023.
[14] M. A. S. Ekowati and D. Darsini, "Implementation of blockchain technology in UKTS library: Challenges, obstacles, and opportunities," *Asian J. Environ. Res.*, vol. 1, no. 2, Art. no. 2, Aug. 2024, doi: 10.69930/ajer.v1i2.89.
[15] D. Mathpal, N. Chandra, and D. Prashad, "Library management system using blockchain," *Govind Ballabh Pant Univ. Agric. Technol.*, 2023, pp. 1–9.
[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *J. Cryptography*, vol. 6, no. 1, pp. 1–20, 2009.
[17] European Union, "General data protection regulation (GDPR) – Legal text," *J. Eur. Data Protect.*, vol. 15, no. 4, pp. 33–55, 2025.