Project Report

Date	21 .10.2023
Team ID	NM2023TMID10632
Project Name	Block-chain powered library management

1. Introduction:

Project overview:

"Block chain-Powered Library Management" revolutionizes traditional library systems by harnessing Ethereum smart contracts for transparent and secure book data management. This cutting-edge approach ensures the integrity of library operations in a decentralized environment. Libraries, historical repositories of knowledge, can now seamlessly transition to a digital age with immutable and transparent book records stored on the block chain. This system introduces a structured database where each book is represented by a smart contract, containing essential details such as title, author, ISBN, and ownership history. Users can query book information, and authorized personnel can efficiently add new books or transfer ownership with a single, secure transaction. By eliminating centralized intermediaries and enabling end-to-end verification, this system empowers libraries with unprecedented data transparency, security, and efficiency. Patrons can trust the accuracy of book details, while librarians can streamline operations and maintain an unforgeable history of book ownership changes. "Block chain-Powered Library Management" is the future of library administration, enhancing accessibility and trust in an ever-evolving digital landscape.

Purpose:

The purpose of a block chain-powered library management system is to enhance traditional library operations by leveraging the unique features and benefits of blockchain technology. Here are the primary purposes:

- Security-- Block chain ensures the security and integrity of library records. All transactions, including book checkouts, returns, and due dates, are securely recorded and tamper-proof. This prevents unauthorized alterations and enhances the overall data security of the library.
- Transparency—Block chain offers a transparent and publicly accessible ledger.
- Efficiency-- Smart contracts automate many library processes, reducing administrative overhead.
- Digital Integration--A block chain-powered library management system can seamlessly manage digital resources, such as e-books.
- Interlibrary Collaboration—Block chain facilitates secure interlibrary loans and resource sharing.
- Data Privacy—Block chain can protect patron privacy by allowing them to use block chain-based IDs for library services.

- Immutable Records-- Records stored on the block chain are permanent and cannot be altered or deleted. This ensures the long-term preservation of library data and history.
- Decentralization—Block chain operates on a decentralized network of nodes, reducing the risk of a single point of failure. This ensures the availability and reliability of library services.
- Innovation—Block chain opens up opportunities for innovative library services.

2. Literature survey:

Existing problem:

- Scalability
- User Adoption
- Privacy Concerns
- Cost and Energy Consumption
- Interoperability
- Data Recovery
- Technical Expertise
- Legal and Regulatory Challenges
- Smart Contracts

References :

- Application of Blockchain in Libraries and Information Centers

IM Omame, JC Alex-Nmecha... and Organization Systems in Library, 2021-igi-global.com

- Block chain in libraries Meth-2019-scholarworks.sjsu.edu
- Off the chain: Blockchain technology-An information organization system L. Dolan, B Kavanaugh, K Korinek... - Technical services..., 2019 - Taylor & Francis
  Block chain for Information Security and Privacy

UP Rao, PK Shukla, C Trivedi, S Gupta, ZS Shibeshi - 2021 - books.google.com

- Block chain and Cryptocurrencies: Legal and Ethical Considerations

N Mason, MN Halgamuge, K Aiyar... Use Cases on

Block chain, 2021-igi-global.com

- Application of Blockchain in Libraries and Information Centers

Omame, JC Alex-Nmecha and Organization Systems in Library, 2021-igi-global.com

- Understanding user acceptance of block chain- based smart locker

JW Lian, CT Chen, LF Shen, HM Chen - The Electronic Library,2020 - emerald.com

- The library as a facilitator: how bitcoin and chain technology can aid nations J Nicholson - The Serials Librarian, 2017 - Taylor & Francis

Problem statement definition:

Libraries play a crucial role in providing access to knowledge and resources for communities and institutions. Efficient library management is essential for ensuring that books and digital materials are cataloged, tracked, and accessible to library patrons. However, traditional library management systems face several challenges, including issues related to data integrity, security, and interoperability. Block chain technology has the potential to address these challenges and revolutionize library management systems.

The aim of this project is to develop a block chain-powered library management system that addresses the following key challenges are;

Data Integrity and Security

Interoperability

Patron Privacy

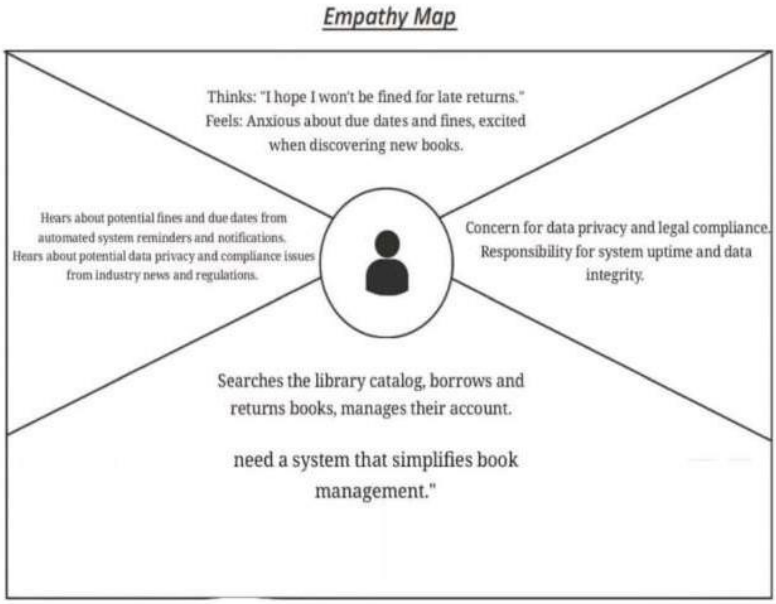
Decentralization and Redundancy

User Experience

Scalability and Cost

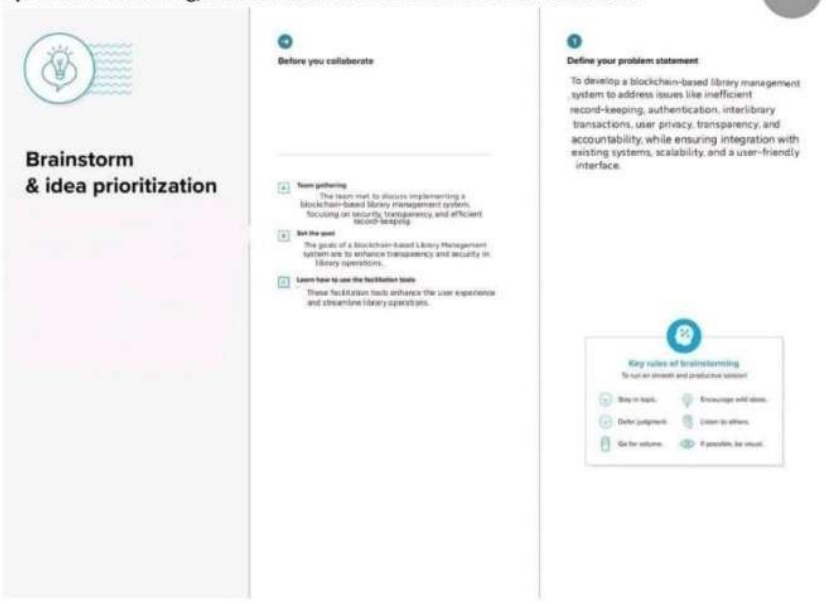
3. Ideation and proposed solution:

Empathy map

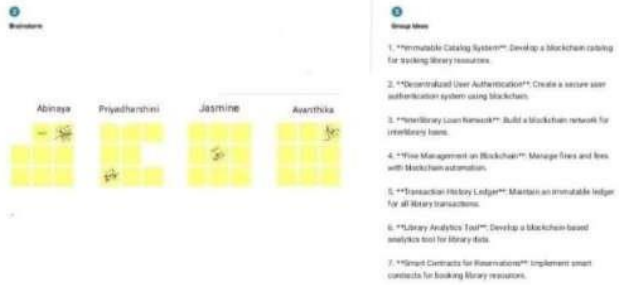


Brainstorming:

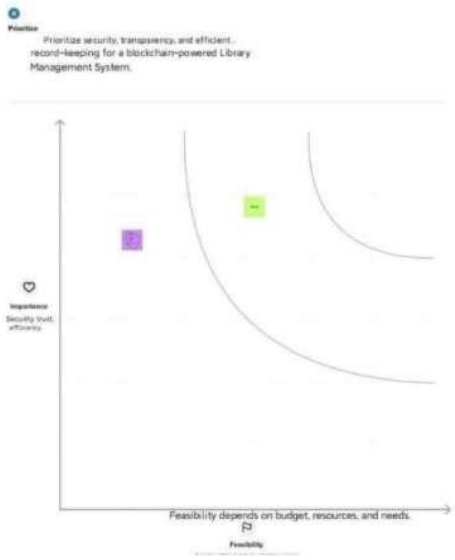
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



4. Requirement analysis:

Function requirements:

1. User Registration and Authentication:

- Users should be able to register on the platform securely.

- Authenticate users with unique identifiers (e.g., public keys) and secure login methods (e.g., multi-factor authentication).

2. Digital Identity Management:
    - Create and manage digital identities for library users, including students, staff, and patrons.
    - Associate digital identities with real-world library card or user IDs.
3. Catalog Management:
    - Maintain a comprehensive digital catalog of library resources, including books, journals, digital media, and other materials.
    - Enable easy search, retrieval, and filtering of resources.
4. Transaction History:
    - Record and timestamp all library transactions, such as borrowing, returning, and renewing materials.
    - Ensure the immutability of transaction records through blockchain.
5. Smart Contracts:
    - Implement smart contracts for automating routine library operations, such as overdue fines, reservations, and interlibrary loans.
    - Execute these contracts based on predefined rules and triggers.
6. Inventory Management:
    - Keep track of the availability, location, and status of each library resource.
    - Provide real-time updates on resource availability to users.
7. Borrowing and Returning:
    - Allow users to borrow and return library materials using a secure and transparent process.
    - Update the blockchain ledger with the status of each item.
8. Digital Ownership and Licensing:
    - Define and enforce digital ownership rights for library materials.
    - Enable licensing and copyright management for e-books, journals, and other digital content.
9. Interlibrary Loans:
    - Facilitate interlibrary loans using blockchain to ensure transparency and traceability of borrowed items.
10. User Profiles:
    - Enable users to manage their profiles, including preferences, borrowing history, and notifications.
    - Allow users to view their digital checkouts and holds.
11. Privacy and Data Security:
    - Implement strong encryption and data protection measures to safeguard user information.
    - Provide privacy controls for users to manage the visibility of their transaction history.
12. Notifications and Alerts:
    - Send automated notifications for due dates, reservation availability, and other library-related updates.

Non functional requirements:

Non-functional requirements are critical aspects of a blockchain-powered library management system that focus on the system's characteristics, performance, and quality attributes. Here are some non-functional requirements for such a system:

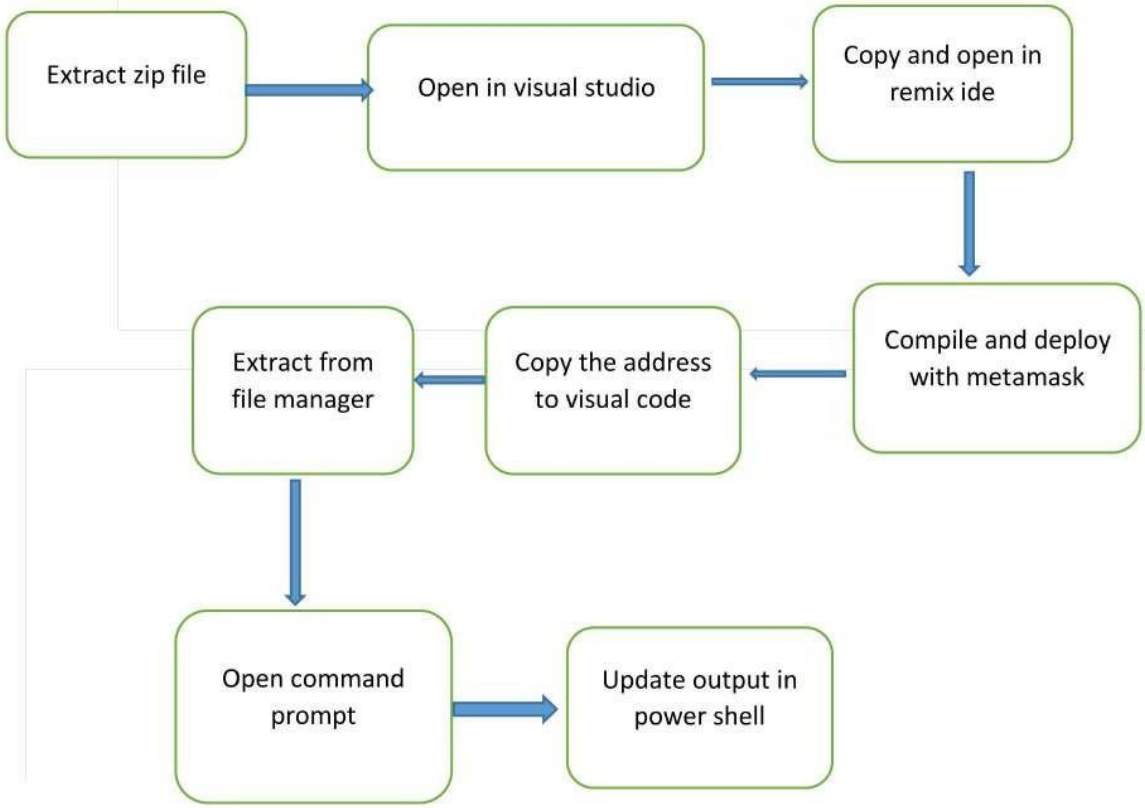
1. **Security**:
    - **Data Security**: Ensure that user data, transaction history, and resource information are stored securely on the blockchain to prevent unauthorized access or data breaches.
    - **Cryptography**: Implement strong cryptographic techniques to protect user identities, transactions, and sensitive information.
    - **Consensus Algorithm Security**: Choose a secure consensus algorithm to prevent malicious attacks on the blockchain network.
2. **Scalability**:
    - Ensure the system can handle a growing number of library users, resources, and transactions without compromising performance or response times.
3. **Performance**:
    - **Transaction Speed**: Define acceptable transaction processing times to ensure that borrowing, returning, and other library operations are swift.
    - **Throughput**: Specify the system's ability to handle a high volume of concurrent transactions efficiently.
    - **Latency**: Determine acceptable latency for user interactions, such as searching the catalog or checking out materials.
4. **Availability**:
    - Define the system's availability requirements to ensure it is accessible to users 24/7 with minimal downtime.
    - Implement failover mechanisms and redundancy to guarantee high availability.
5. **Reliability**:
    - Set reliability targets to minimize the risk of system failures, data corruption, or other disruptions.
    - Implement backup and recovery strategies to restore the system in case of failure.
6. **Compliance and Regulations**:
    - Ensure that the system adheres to relevant regulations and industry standards, such as data protection laws and library management guidelines.
7. **Usability**:
    - Define user interface and user experience (UI/UX) requirements to ensure the system is user-friendly, accessible, and intuitive for library staff and patrons.
8. **Interoperability**:
    - Specify how the blockchain-powered library management system will integrate with existing library software, hardware, and third-party services.
9. **Auditability**:

- Enable a comprehensive audit trail that logs all system activities and interactions on the blockchain, ensuring transparency and accountability.

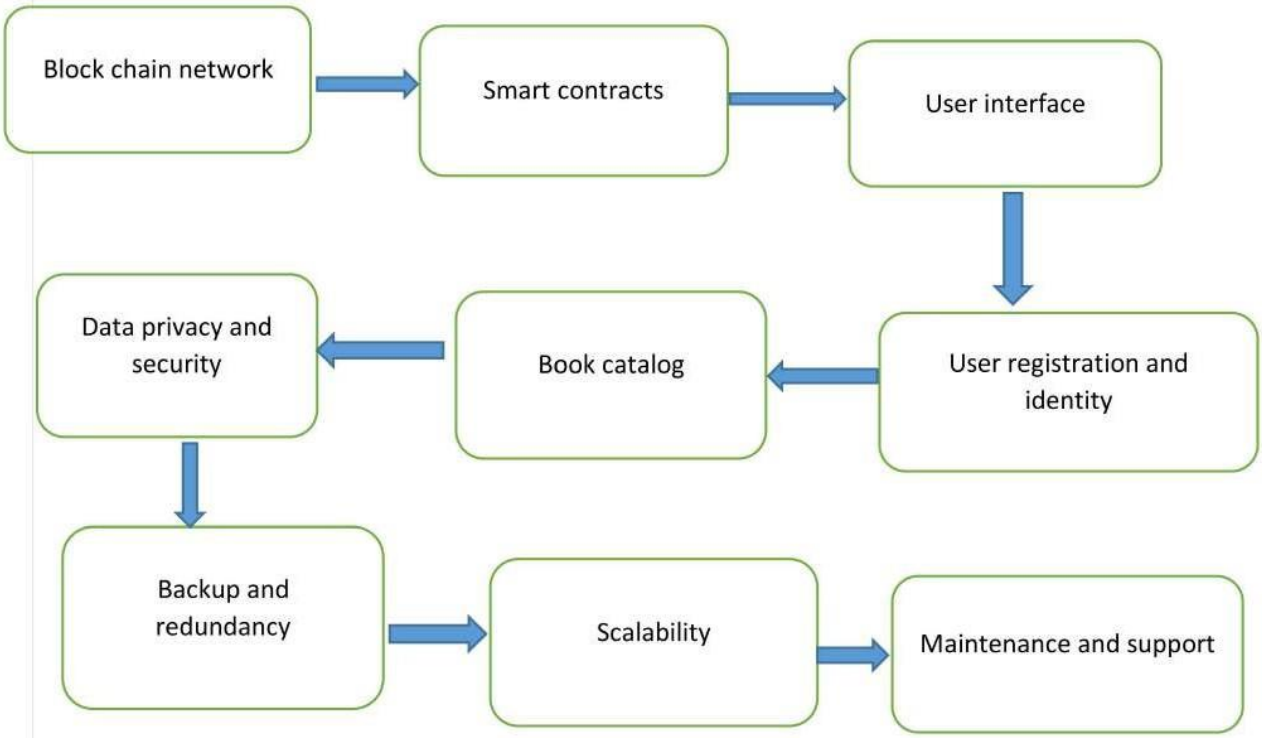
10. **Adaptability**:
    - Design the system to accommodate future changes, such as updates to blockchain technology, regulatory requirements, or library processes.
11. **Resource Efficiency**:
    - Minimize resource consumption, such as computational power and storage space, to keep operational costs low and eco-friendly.
12. **Blockchain Network Latency**:
    - Consider the latency of the underlying blockchain network and its impact on transaction confirmation times.
13. **Data Migration and Portability**:
    - Plan for the migration of data to new blockchain versions or platforms while ensuring data integrity and continuity.
14. **Backup and Disaster Recovery**:
    - Implement backup and disaster recovery procedures to protect against data loss or system failures.

5. Project design:

Data flow diagrams and user stories:

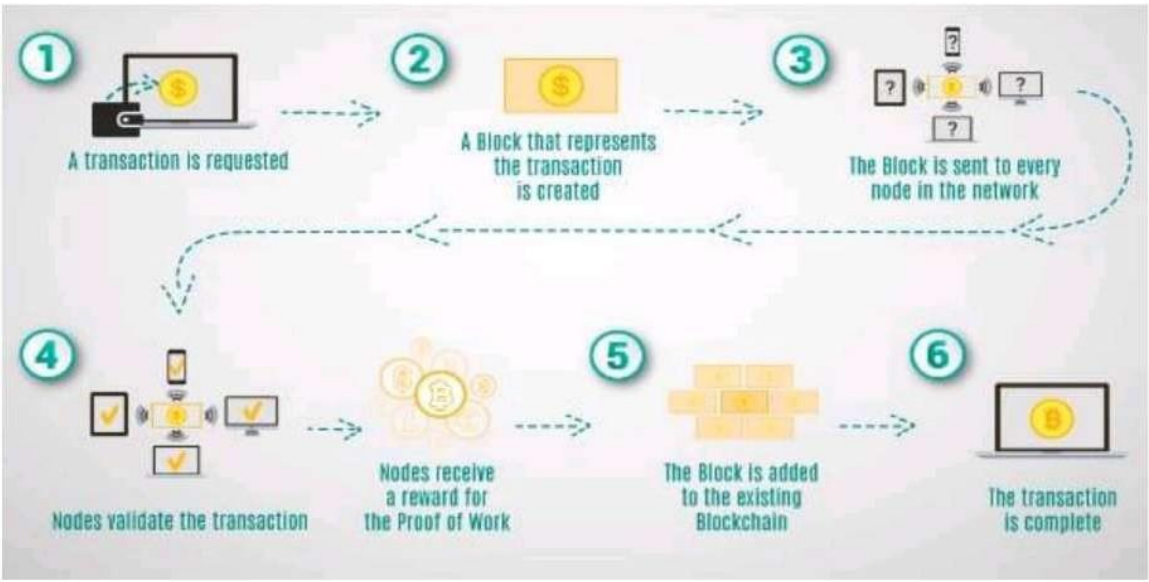


Solution architecture:



6. Project planning and scheduling:

Technical architecture:



Sprint planning and estimation:

Tools required are installed before deadlines. The tools are meta mask, visual studio and node JS

Remix code were built

Using file connector.js

All requirements were built before deadlines

Sprint delivery schedule:

Team members do all steps in developing the project

Each member are separated to do different works

Works are done on date fixed

7. Coding and solutioning:

Feature1

SPDX License Identifier: This is a comment indicating the SPDX license identifier for the contract. In this case, it's specified as MIT, which is a permissive open-source license.

Solidity Version Pragma: The contract specifies the Solidity version to be used, in this case, version 0.8.0. This pragma ensures that the contract is compiled using the specified version of the Solidity compiler.

Feature 2:

Owner Management:

The contract has an owner variable that represents the owner's Ethereum address. The owner is set to the address that deploys the contract in the constructor.

There's a only Owner modifier that restricts certain functions to be callable only by the owner. This modifier checks if the sender of the transaction is the owner before allowing the function to proceed.

Book Struct - The contract defines a Book struct to represent book information. It contains fields for the book's title, author, and the current owner's Ethereum address.

8. Performance testing:

Performance metrics:

Transaction Speed: Measure the time it takes to record and validate transactions. Block chain systems should provide faster and more efficient transactions compared to traditional library management systems.

Data Security: Assess the level of data security and integrity provided by the block chain. Block chain's immutability and encryption can enhance data protection.

Decentralization: Evaluate the degree of decentralization achieved. A well-implemented block chain system should reduce the reliance on a single central authority.

Cost Efficiency: Analyze the cost savings achieved through block chain, including reduced administrative costs and improved resource allocation.

Accessibility: Measure the accessibility of library resources and services for users. Block chain can provide improved accessibility through decentralized, online access.

Privacy and Permissions: Assess the control over user privacy and permissions. Block chain can enable fine-grained control over who can access and modify library resources.

Interoperability: Check how well the block chain system integrates with other library systems and external services.

Scalability: Measure the system's ability to scale with the growing needs of the library. Block chain should support increased transactions and data.

User Experience: Gather feedback from library patrons and staff regarding their experience with the block chain-powered system.

Auditing and Reporting: Examine the system's capabilities for generating reports and audits. Block chain's transparency can simplify this process.

Smart Contracts: Evaluate the use of smart contracts for automating tasks such as resource checkouts, returns, and overdue fines.

Energy Efficiency: Consider the energy consumption of the block chain network, as this can impact both operational costs and environmental sustainability.

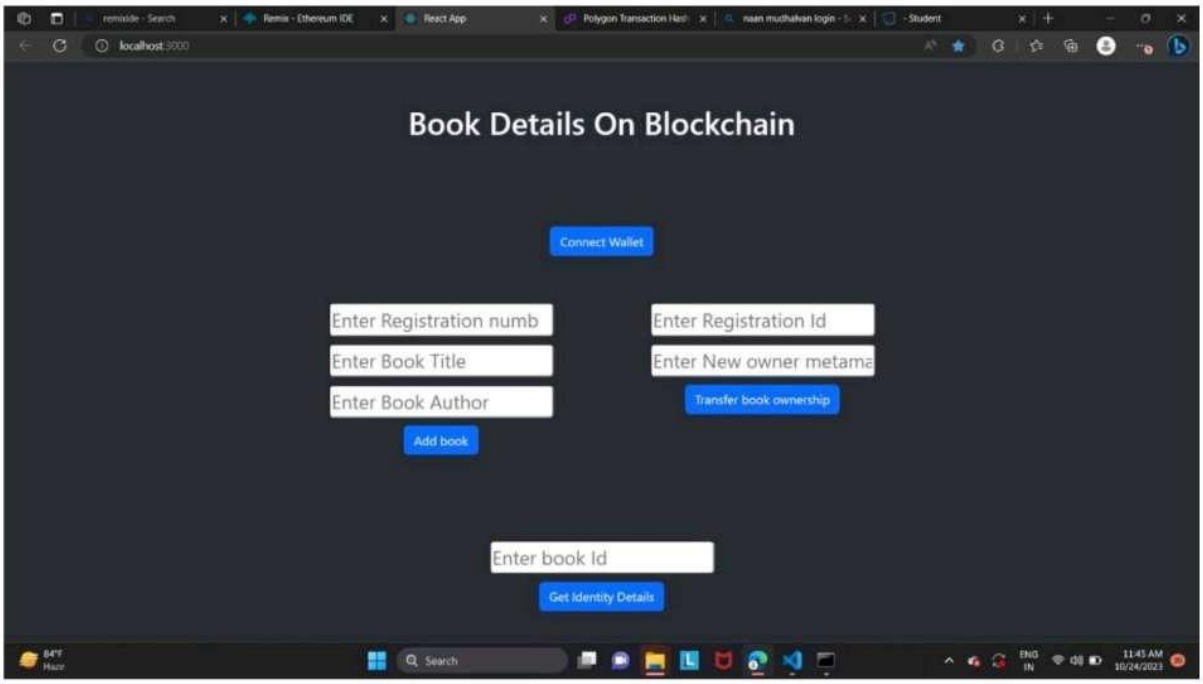
Adoption and Integration: Monitor how well the technology is adopted by the library staff and integrated into their workflows.

Maintenance and Support: Assess the availability of maintenance and support for the block chain system, including software updates and troubleshooting.

User Feedback and Satisfaction: Regularly collect feedback from library users and staff to gauge their satisfaction with the block chain-powered library management system.

Long-Term Sustainability: Ensure the block chain system's sustainability and its ability to adapt to future technological developments.

9. Results:



10. Advantages and Disadvantages:

Advantages:

- Enhanced Security: Prevents unauthorized book removal or tampering of records.
- Efficiency: Automates many library processes, reducing administrative workload.
- Transparency: Enables patrons to trust library operations and book availability.
- Interlibrary Collaboration: Streamlines interlibrary cooperation.

- Digital Integration: Manages e-books and digital resources effectively.

Disadvantages:

- Initial Setup: Transitioning from traditional systems to block chain can be complex.
- User Adoption: Patrons and staff may require training to adapt to the new system.
- Scalability: Ensuring the block chain can handle the library's future growth.

11. Conclusion:

A block chain-powered library management system promises a future where library operations are more secure, efficient, and transparent. It enhances the overall library experience for patrons and staff while paving the way for innovative services in the digital age.

Block chain implementation can be complex and costly, with potential scalability issues and energy consumption concerns. User adoption and regulatory challenges may also pose hurdles. The success of block chain in library management depends on a careful evaluation of these advantages and disadvantages, as well as a commitment to address these challenges effectively.

As block chain technology continues to mature and gain broader acceptance, libraries exploring its adoption should stay informed about best practices, advancements in block chain solutions, and the evolving regulatory landscape. With thoughtful planning and a focus on user needs and security, block chain-powered library management systems have the potential to usher in a new era of efficiency, transparency, and accessibility in library services.

12.Future scope:

- Block chain can facilitate increased transparency alongside reducing the costs and risks associated with supply chain management.
- Better end-to-end traceability of materials in the supply chain for compliance with corporate standards.
- Limitations on product counterfeiting and fraud.
- Improved transparency and visibility into outsourced contract management with better control over compliance.
- Lower paperwork and reduction in administrative costs.

13.Appendix:

Source code:

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

 contract BookRegistry {

    address public owner;

  constructor() {

        owner = msg.sender;

    }

    modifier onlyOwner() {

        require(msg.sender == owner, "Only the owner can perform this action");

        _;

    }

    struct Book {

        string title;

        string author;

        address currentOwner;

    }


    mapping(uint256 => Book) public books;

    uint256 public bookCount;


    event BookAdded(uint256 indexed bookId, string title, string author, address indexed
owner);
```

```
    event OwnershipTransferred(uint256 indexed bookId, address indexed previousOwner,
address indexed newOwner);



    function addBook(uint256 registration, string memory _title, string memory _author)
external onlyOwner {



        books[registration] = Book(_title, _author, owner);

        bookCount++;

        emit BookAdded(registration, _title, _author, owner);

    }



    function transferOwnership(uint256 registrationId, address _newOwner) external {

        require(_newOwner != address(0), "Invalid address");

        require(_newOwner != books[registrationId].currentOwner, "The new owner is the
same as the current owner");

        require(msg.sender == books[registrationId].currentOwner, "Only the current owner
can transfer ownership");



        address previousOwner = books[registrationId].currentOwner;

        books[registrationId].currentOwner = _newOwner;



        emit OwnershipTransferred(registrationId, previousOwner, _newOwner);

    }
```

```
    function getBookDetails(uint256 registrationId) external view returns (string memory,
string memory, address) {



        Book memory book = books[registrationId];

        return (book.title, book.author, book.currentOwner);

    }

}
```

Github and project demo link:

https://youtu.be/2t4kvnQFVfE?si=RORqxXQOkTfgnRsX

GitHub link:

https://github.com/Avanthikabaskaran/Blockchain-Powered-Library-Management.git