International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

PyCharm vault a Blockchain Based Transaction Ledger

Logu Ajtih Kumar R, Maadesh Raajan L A, Mohan Raj K, Navaneetha Krishnan K,

Mrs.Esther T

Department of Information Technology, Batchelor of Technology, Sri Shakthi Institute of Engineering and Technology (Autonomous) Coimbatore-641062

ABSTRACT

In an era where secure and tamper-proof record-keeping is critical, traditional transaction systems often face challenges such as unauthorized modifications, lack of transparency, and difficulties in auditing. The Centralized Blockchain Transaction Ledger is a technology-driven solution designed to address these issues by implementing a centralized blockchain architecture that ensures data integrity, security, and traceability. The system allows users to perform and record transactions via a React-based frontend, while a Flask backend manages OTP verification, block creation, and ledger updates. Each transaction is encapsulated in a block containing essential metadata, cryptographic hashes, and timestamps, which together form a continuous, verifiable chain. Additionally, the blockchain ledger is encrypted to prevent unauthorized access, providing an added layer of security. By integrating blockchain principles with user-friendly interfaces and OTP-based authentication, this project offers a reliable, centralized platform for secure transaction management. Ultimately, the system enhances trust, accountability, and transparency in record-keeping, providing a practical demonstration of blockchain technology's potential in real-world centralized applications..

INTRODUCTION

Blockchain is a type of digital ledger that records information in a way that makes it tamper proof and verifiable. In a blockchain, data is stored in blocks, and each block contains information about transactions, a timestamp, and a unique cryptographic hash of the previous block. This structure ensures that even the smallest change in a block's content will change its hash and invalidate all subsequent blocks. The primary benefit of this design is data integrity, as it makes unauthorized modifications easily detectable. In this project, we use a centralized blockchain, which means a single server manages the ledger. Unlike decentralized blockchains, which require consensus across multiple nodes and are more computationally intensive, centralized blockchains are faster and easier to manage. Despite being centralized, this system preserves the essential blockchain benefits such as immutability, transparency, and auditability. It allows us to implement and test blockchain concepts efficiently and serves as an educational and practical demonstration of secure transaction recording.

Objective:

1. **To develop a secure and tamper-proof transaction recording system**

The main goal is to ensure that all purchase and sale details are recorded in an immutable ledger, preventing unauthorized data modification or deletion.

2. **To implement a simplified blockchain mechanism for centralized storage**

The project focuses on using blockchain principles such as chaining and hashing to maintain integrity while keeping the data centralized within the shopkeeper's device.

3. **To integrate data encryption for enhanced privacy**

AES-based encryption is applied to ensure that all blockchain records and backups remain confidential and unreadable by unauthorized users.

4. **To include OTP-based verification for authenticated data entry**

A one-time password (OTP) system validates each transaction input, ensuring only verified shopkeepers can add new records to the blockchain.

5. **To provide a user-friendly interface using React and Flask**

The frontend, built with React, allows easy access to the blockchain data, while Flask manages backend operations and secure communication.

6. **To maintain automated backup of blockchain files in MongoDB**

Every update in the blockchain is backed up securely in MongoDB to prevent data loss in case of local file corruption or device issues.

7. **To validate blockchain integrity regularly**
The system periodically checks for tampering by recalculating hashes, ensuring that the stored data remains authentic and consistent.

LITERATURE SURVEY

Early research on digital ledger systems emphasized centralized databases and manual bookkeeping, which often suffered from data manipulation, loss, and unauthorized access. With the introduction of blockchain technology (*Nakamoto, 2008*), secure and immutable data recording became achievable through cryptographic hashing and linked block structures, ensuring transparency and traceability without third-party validation.

However, fully decentralized systems were found unsuitable for small-scale or local operations due to synchronization overhead and resource demands. Recent studies propose **centralized blockchain models** that retain blockchain's integrity and tamper detection features while improving efficiency and simplicity for single-node environments such as retail and institutional recordkeeping.

The integration of **AES encryption** for data confidentiality and **hash-based verification** for integrity has been widely recognized as an effective security mechanism in digital ledgers. Research further supports **OTP-based authentication** as a lightweight but powerful method for verifying transaction ownership and preventing unauthorized access.

Comparative analyses show that blockchain-inspired ledgers outperform traditional SQL systems in traceability and auditability while remaining scalable and cost-efficient. The inclusion of **MongoDB** for encrypted backup provides additional fault tolerance and data recovery capabilities.

In conclusion, literature indicates a strong trend toward **hybrid blockchain systems**—leveraging blockchain principles, cryptography, and centralized control—to achieve secure, transparent, and efficient record management for localized business and institutional use.

METHODOLOGY

The methodology for the **Centralized Blockchain Transaction Ledger** project begins with identifying the need for a secure, tamper-proof system to record and verify shop-level transactions. The design process focuses on integrating blockchain principles within a centralized framework, ensuring data integrity, immutability, and transparency without the complexity of a decentralized network. The system architecture combines a **Flask backend** for transaction processing, a **React frontend** for user interaction, and **MongoDB** for encrypted ledger backup and recovery.

Each transaction passes through verification steps involving **OTP authentication** and **AES encryption**, ensuring that only authorized entries are added to the blockchain. The blockchain is structured with cryptographic hashes to prevent unauthorized modifications and maintain a chronological record of transactions.

Testing and validation involve verifying block integrity, encryption correctness, and user accessibility through simulated transaction scenarios. The final stage includes optimizing performance, enhancing security, and ensuring scalability for small businesses or institutional applications.

Existing System

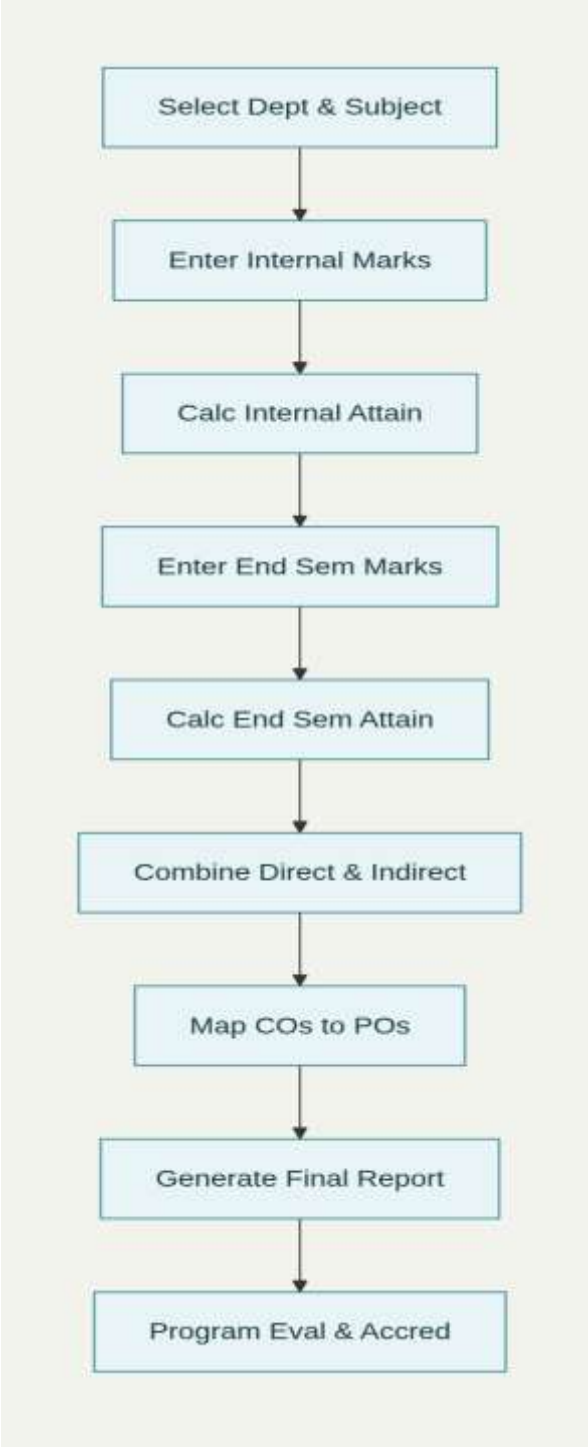
Existing transaction systems rely on traditional databases or manual recordkeeping, which are vulnerable to unauthorized edits, data loss, and human errors. Such systems often lack built-in verification or tamper detection, leading to low trust and auditability. Centralized databases without cryptographic integrity checks fail to ensure data immutability and secure traceability of transactions.

Disadvantages

1. Traditional ledgers are prone to tampering and accidental data loss.
2. Lack of cryptographic verification allows undetected alterations.
3. No secure backup or redundancy mechanisms.
4. Absence of automated authentication increases risk of unauthorized access.
5. Manual processes reduce efficiency and transparency in transaction validation.

Proposed system:

The proposed system automates Outcome-Based Education (OBE) by integrating internal assessments, assignments, and end-semester exams into a unified platform that calculates Course Outcome (CO) attainment automatically. It features dynamic CO–PO mapping to link course achievements with broader program goals, aiding accreditation and quality assurance. The system prioritizes usability with intuitive data entry interfaces for faculty and real-time performance reporting. It incorporates standard benchmarks and weighted calculations based on NBA guidelines to ensure consistent evaluation. Additionally, the system supports scalability across departments and courses, includes error-checking to improve data accuracy, and offers detailed analytics for continuous curriculum improvement, aligning academic goals with measurable outcomes.



SYSTEM REQUIREMENTS

1. Hardware Requirements

Processor: Intel Core i3 or higher

RAM: Minimum 4 GB (8 GB recommended for smoother development)

Storage: Minimum 200 MB of free space for blockchain files and dependencies

Display: Monitor with minimum 1024×768 resolution

Input Devices: Standard keyboard and mouse

2. Software Requirements

Operating System: Windows 10/11, Linux (Ubuntu), or macOS

Frontend Framework: React.js

Backend Framework: Flask (Python)

Database: MongoDB (for encrypted backup storage)

Programming Language: Python 3.9 or higher, JavaScript (ES6+)

Libraries and Modules:

- Flask, Flask-CORS (for backend routing)
- PyMongo (for MongoDB connection)
- PyCryptodome (for AES encryption/decryption)
- hashlib, base64, datetime (for hashing and encoding)
- React Router, Axios (for frontend routing and API calls)

Development Tools: Visual Studio Code / PyCharm / Node.js (v18+)

Browser: Chrome, Edge, or Firefox (latest versions)

MODULE DESCRIPTION

1. Transaction Entry Module

1. Allows shopkeepers to input transaction details such as batch ID, quantity, cost, and buyer information.
2. Each transaction is linked to a verified phone number via **OTP authentication** before being added to the blockchain.
3. Ensures data accuracy and prevents unauthorized or duplicate entries.

2. Blockchain Creation & Storage Module

1. Maintains a **centralized blockchain ledger** with each block containing index, timestamp, data, previous hash, and hash.
2. The first block (Genesis Block) initializes the ledger.
3. Each new transaction is stored as a block linked to the previous one, ensuring chronological order and tamper detection.
4. The complete blockchain is **encrypted using AES** and saved locally as blockchain.json.

3. Encryption and Backup Module

1. Employs **AES-128 (CBC mode)** encryption to secure blockchain data against unauthorized access.
2. The encrypted blockchain is also stored in **MongoDB**, providing redundancy and data recovery capability.
3. Any attempt to modify or decrypt data without the correct key results in failure, ensuring confidentiality and integrity.

4. OTP Authentication Module

1. Generates a random six-digit OTP sent (simulated) to the registered mobile number during transactions.
2. Ensures that only verified users can authorize new transactions or updates.
3. Adds an extra layer of **user authentication and security** to the centralized blockchain system.

5. Blockchain Validation Module

1. Periodically validates all stored blocks to detect any tampering or broken hash links.
2. Uses **SHA-256 hashing** to recalculate and verify block integrity.
3. Displays results through the React interface, confirming ledger authenticity.

6. Frontend Interface Module

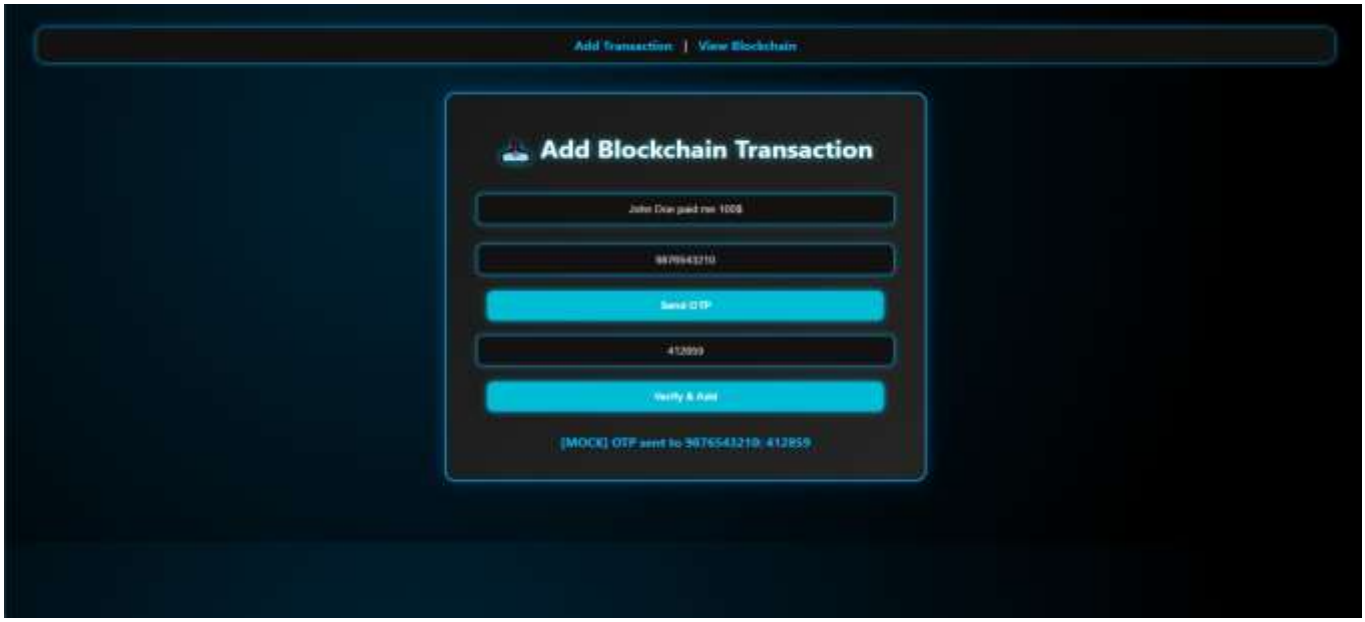
1. Developed using **React.js** with a user-friendly design and responsive layout.
2. Includes pages for **Login/Register**, **Add Produce (Add Block)**, and **View Blockchain**.
3. Uses **Axios** to send and retrieve data from Flask APIs in real-time.
4. Provides smooth navigation and clear visualization of transaction records.

7. Backup Retrieval & Restoration Module

1. Fetches encrypted blockchain backups from MongoDB if local files are lost or corrupted.
2. Automatically decrypts and restores the ledger to maintain continuity.
3. Ensures data persistence and reliability even in case of system failure.

MAINPAGE:

Page 1:



Page 2:

CONCLUSION

The Centralized Blockchain Transaction Ledger successfully demonstrates how blockchain principles such as immutability, transparency, and security can be applied in a controlled environment without decentralization. By integrating AES encryption, OTP verification, and MongoDB backups, the system ensures that transaction data remains protected, verifiable, and recoverable. The combination of Flask for backend logic and React for the frontend provides a seamless and user-friendly experience for shopkeepers. Through secure block creation and validation, the project highlights how cryptographic methods can safeguard digital records against tampering. Overall, this system provides a reliable, efficient, and scalable solution for maintaining transaction records in a centralized yet secure manner.

REFERENCES

1. Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). **An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends.** *2017 IEEE International Congress on Big Data (BigData Congress)*, 557–564. https://doi.org/10.1109/BigDataCongress.2017.85

2. Mougayar, W. (2016). **The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology.** Hoboken: Wiley.

3. Antonopoulos, A. M. (2017). **Mastering Bitcoin: Programming the Open Blockchain (2nd Edition).** O'Reilly Media.

4. Stallings, W. (2020). **Cryptography and Network Security: Principles and Practice (8th Edition).** Pearson.

5. Flask Documentation. (2025). *Flask Web Development Framework.* Available: https://flask.palletsprojects.com/

6. React Documentation. (2025). *React – A JavaScript Library for Building User Interfaces.* Available: https://reactjs.org/docs/getting-started.html

7. MongoDB Documentation. (2025). *MongoDB Manual.* Available: https://www.mongodb.com/docs/manual/