Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

FinLedger AI: A Blockchain-Secured Organizational Finance Monitor with LLM-Powered Analysis

Hamza Madni

madnih474@gmail.com

University of Sahiwal

Research Article

Keywords: blockchain, SHA-256, financial ledger, tamper detection, large language models, role-based access control

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11684325/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

FinLedger AI: A Blockchain-Secured Organizational Finance Monitor with LLM-Powered Analysis

Hamza Madni

Department of Computer Science, University of Sahiwal, Sahiwal, Pakistan

2022-CS-10135@uosahiwal.edu.pk | madnih474@gmail.com

Preprint draft - prepared 10 Nov 2026 from the author's BS final-year project documentation. Not yet submitted.

Abstract

Most organizational finance systems share three persistent weaknesses: historical records can be altered without leaving a trace, users are granted broader access than their roles require, and the systems are reactive - they store and calculate but do not warn anyone when something looks wrong. This paper presents FinLedger AI, a web-based financial ledger that addresses all three problems in a single system. Built on the MERN stack (MongoDB, Express.js, React, Node.js), FinLedger AI seals every transaction into a custom blockchain in which each block is hashed with SHA-256 and linked to its predecessor, so any after-the-fact modification breaks the chain and is flagged immediately during verification. Access is separated by role: administrators see organization-wide data, while staff see only their own records. On top of this integrity layer, a large language model (Google Gemini) continuously analyses the ledger to produce budget alerts, spending forecasts, and plain-language financial summaries. Tamper-detection tests confirmed that altering even a single stored value breaks the hash chain and surfaces an alert. The system demonstrates that tamper-evident record keeping and automated financial intelligence can be combined in one practical application built entirely on an open-source stack.

Keywords: blockchain, SHA-256, financial ledger, tamper detection, large language models, MERN stack, role-based access control

1. Introduction

Managing organizational finances has always involved two problems that are usually treated separately: keeping records secure, and making sense of what those records mean. Most existing tools address one or the other, rarely both. Centralized databases handle storage well, but they give administrators the technical ability to change historical records without leaving any trace, which opens the door to fraud and makes auditing unreliable. Many platforms also separate roles poorly, so staff members can see organization-wide data that has nothing to do with their work. Finally, most systems are reactive rather than proactive: spotting a budget problem or an unusual pattern requires someone to go looking for it manually, so issues are often noticed too late.

FinLedger AI was built to tackle these problems together. Its design rests on four objectives:

1. Data immutability. A custom blockchain using SHA-256 hashing cryptographically seals every transaction; any change made to the database outside the normal flow is detected immediately.

2. Role-based access control. Administrators see the full picture across the organization; staff access only their own data.

3. AI-powered analysis. Google Gemini analyses transaction histories automatically, flags potential issues, generates forecasts, and produces financial summaries on demand.

4. Usability. A React and Tailwind CSS frontend presents complex financial data so it can be read at a glance without technical knowledge.

The system is deliberately scoped to digital ledger tracking within an organization: it does not connect to live banking APIs and does not process real currency transfers. That boundary was kept so that both layers - security and intelligence - could be built properly.

2. Related Work

Blockchain technology has been widely studied as a means of making financial records tamper-evident, beginning with Nakamoto's peer-to-peer electronic cash design [1] and developed further in analyses of blockchain as a new economic and organizational infrastructure [2]. Recent work has examined blockchain specifically for financial monitoring, noting both its integrity benefits and its practical challenges [3], and intelligent financial management systems that combine blockchain with connected devices [4]. In parallel, AI-driven financial risk assessment and anomaly detection have been shown to catch patterns that manual review misses [5], and predictive analytics is increasingly used to support corporate finance decisions [6]. FinLedger AI differs from most of this work in integration rather than in any single component: instead of treating the ledger, the integrity mechanism, and the analysis layer as separate products, it builds all three into one application aimed at small and medium organizations that cannot afford enterprise infrastructure.

3. System Design

FinLedger AI consists of two core subsystems on top of a standard web architecture. The backend (Node.js/Express.js) handles JWT-based authentication, exposes the REST API, and runs the cryptographic validation that keeps the blockchain intact. MongoDB (Atlas, via Mongoose) stores users, transactions, budgets, and blocks. The frontend provides role-specific dashboards, financial charts, an AI insights panel, and a Blockchain Explorer through which auditors can verify the integrity of the full ledger. Development followed an Agile process in iterative cycles.

3.1 Blockchain Ledger Subsystem

This subsystem is the security foundation of the platform. It accepts transaction data submitted by a user, seals each record into the chain, stores the resulting block, and runs sequential verification whenever an auditor requests it. It has three components:

- Transaction Controller - receives incoming API requests (JSON payloads carrying amount, category, and type) and passes the data through the pipeline.
- Hashing Engine - combines each new transaction with the previous block's hash and a timestamp to produce a unique SHA-256 fingerprint for the new block.
- Verification Engine - iterates through every stored block in order and checks whether the recorded hashes still match the actual data, returning a boolean integrity result.

Two design constraints shaped this subsystem. First, transactions cannot be edited once recorded - this is a deliberate requirement, not an oversight, because an edit would break the hash chain; corrections are instead made through a new compensating transaction. Second, block creation is synchronous and strictly sequential, because each block depends on the hash of the block immediately before it; parallel or out-of-order processing would produce incorrect hashes.

3.2 AI Intelligence Subsystem

This subsystem connects the ledger to Google Gemini, turning raw transaction data into automated insights. It assembles historical transaction data from the database, builds structured prompts from that data, sends them to the Gemini API, and returns: budget alerts when spending approaches or exceeds limits, spending forecasts, natural-language financial summaries, and conversational (chatbot) answers to questions about the organization's finances. Security and intelligence are therefore not separate add-ons in FinLedger AI: the blockchain guarantees that the data can be trusted, and the AI layer explains what the trusted data means.

3.3 Access Control and Security

Authentication uses JSON Web Tokens with bcrypt-hashed passwords. Role-based access control separates administrators (organization-wide visibility, budgets, verification) from staff (their own transactions and dashboards only), addressing the privacy and usability problems that arise when all users are treated the same.

4. Implementation

The system was implemented in approximately 3.8 KLOC across all modules using free-tier services throughout. The frontend uses React (Vite) with Tailwind CSS and Framer Motion for animated transitions; the backend uses Node.js and Express.js, with the Node.js Crypto module providing SHA-256 hashing; MongoDB Atlas with Mongoose serves as the data store. The interface was wireframed in Figma before development. Testing used Jest with MongoDB Memory Server for unit tests of the blockchain logic and Postman for REST API endpoint testing. Figure 1 shows the implemented staff dashboard, and Figure 2 shows the Blockchain Explorer.



Figure 1. FinLedger AI staff dashboard: monthly income and expenses, current balance, income-vs-expenses chart, and live budget status per category.

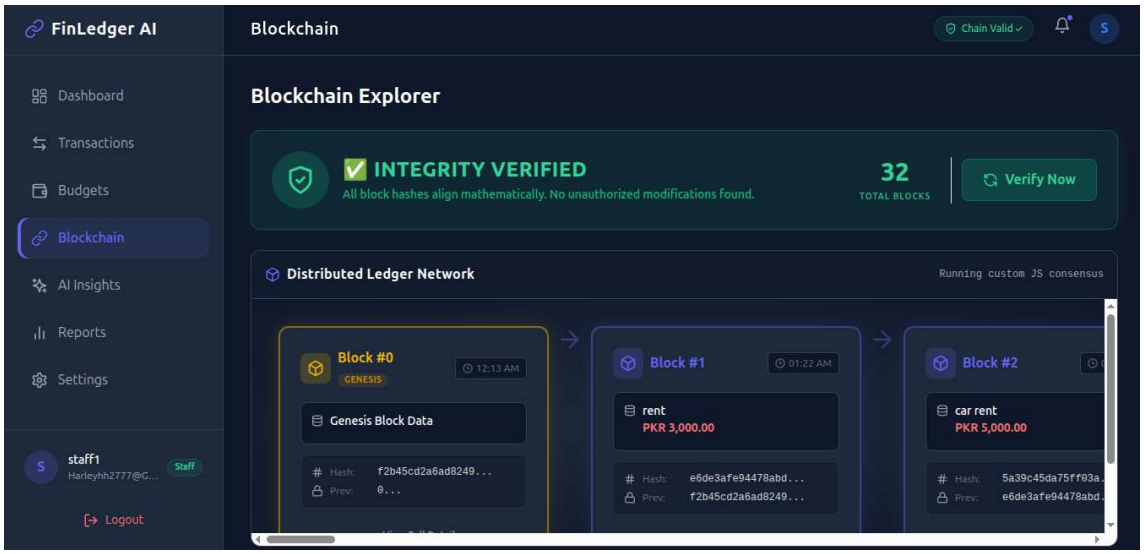


Figure 2. Blockchain Explorer: the chain reports Integrity Verified across 32 blocks; each block card shows its SHA-256 hash and the previous block's hash.

5. Evaluation

The system was evaluated in three ways. First, unit tests (Jest, with an in-memory MongoDB instance) exercised the blockchain logic: block creation, hash linking, and verification of both intact and modified chains. Second, REST API endpoints were tested with Postman for authentication, transaction submission, and role separation between administrator and staff accounts. Third, and most importantly, tamper-detection tests were run against the stored data: records in the database were altered directly, outside the normal application flow, and the verification engine was then executed. In every tamper test, altering a record - even a single value - broke the hash chain immediately and surfaced an alert without delay. All four objectives set out in Section 1 were met in the completed system: financial managers received a real-time dashboard view of budgets, auditors received a mathematically verifiable record of every transaction, and the AI layer produced alerts, forecasts, and summaries from the live ledger. Figure 3 shows the financial reports view generated from the same ledger data.



Figure 3. Financial Reports view: monthly totals and expense breakdown generated from the sealed ledger.

6. Limitations

Three limitations are acknowledged. First, full-chain verification is sequential: for a typical organizational ledger it is fast, but re-computing every block for an extremely large dataset (millions of records) introduces computational delay, because each block must be recalculated in order. Second, the immutability that protects the ledger also means genuine mistakes cannot be edited in place; they must be corrected with compensating transactions, which some organizations may find less convenient than direct editing. Third, the system does not connect to live banking APIs and processes no real currency transfers, so its forecasts and alerts are only as current as the data users enter into it.

7. Future Work

The documented limitations point to clear next steps: incremental or parallel verification strategies so that integrity checks stay fast on very large ledgers; optional integration with banking and payment APIs so that transactions are captured automatically rather than entered manually; and deployment studies with real organizations to measure how the AI alerts affect financial decision-making in practice.

8. Conclusion

FinLedger AI set out to answer a straightforward question: can blockchain-based security and AI-driven financial analysis be built into a single, practical web application? Based on the completed system and its tests, the answer is yes. The custom SHA-256 blockchain engine worked exactly as intended - every tamper test confirmed that the chain detects modification immediately. The Gemini integration added what the security layer alone could not: the ability to make the data useful in real time, explaining itself, flagging concerns, and looking ahead. Together, the two layers show that transparency and intelligence in financial management do not require expensive enterprise infrastructure; they can be built from scratch on an open-source stack.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Whitepaper, Bitcoin.org, 2008.

[2] M. Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, Sebastopol, CA, 2015.

[3] K. Jani, K. Joseph, and S. Varma, "Leveraging blockchain technology for enhancing financial monitoring: Main challenges and opportunities," International Journal of Advanced Research, ScienceDirect, Amsterdam, 2024, pp. 45-58.

[4] X. Wang, Y. Zhang, and L. Chen, "Optimization of intelligent financial management system based on blockchain and internet of things," Scientific Reports, Nature Publishing Group, London, 2025, pp. 112-125.

[5] J. Brown, R. Miller, and T. Davis, "AI-Driven Financial Risk Assessment and Anomaly Detection in Cross-Border Transactions," Annals of Applied Sciences, Academic Press, New York, 2024, pp. 201-215.

[6] L. Gupta and P. Sharma, "Predictive Analytics in Corporate Finance: Enhancing Decision Making with Machine Learning," Journal of FinTech Innovation, Springer, Berlin, 2023, pp. 15-28.