International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

Decentralized Asset Management System Using Blockchain

Ms Sameera Begum[1], Jashwanthi.K[2], Lalitha.K[3], Sai Vaishnavi.B[4]

[1]Assistant Professor; Department Of Computer Science And Engineering (AI & ML) Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3,4]B.Tech Students; Department Of Computer Science And Engineering (AI & ML) Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; jashwanthikalluri2203@gmail.com , Vaishhnavii.2703@gmail.com , lallikonatham@gmail.com

Abstract

In the modern digital economy, users increasingly exchange assets through online platforms, including digital art, ownership certificates, real-estate records, and supply-chain goods. Most conventional trading systems depend on centralized authorities or third-party intermediaries, which can introduce issues such as fraud, unauthorized data modification, excessive service charges, delayed settlement, and limited transparency. To overcome these limitations, this paper presents a blockchain-based decentralized asset management system for secure and transparent asset trading. The proposed platform employs Ethereum smart contracts to automate transactions, validate ownership, and permanently store transaction records on an immutable distributed ledger. Asset owners can register and list digital or physical assets, while buyers can complete purchases using Ether without relying on intermediaries. The decentralized architecture minimizes operational costs, strengthens trust between participants, and prevents tampering of ownership data. In addition, the system supports traceability and real-time verification of transferred assets. Potential application areas include digital art marketplaces, real-estate tokenization, supply-chain management, and peer-to-peer ownership exchange. Experimental analysis indicates that the proposed framework offers improved security, transparency, and efficiency compared with traditional centralized asset trading models.

Keywords— *Blockchain, Decentralized Asset Management, Smart Contracts, Ethereum, Asset Trading, Transparency, Security, Tokenization*.

Introduction

In the rapidly expanding digital economy, asset trading has evolved beyond conventional financial exchanges to include digital artwork, real-estate ownership records, intellectual property, and supply-chain resources. Most existing systems depend on centralized organizations or third-party intermediaries such as brokers, agents, and online platforms. Although these models offer convenience, they frequently suffer from problems such as fraud, unauthorized modification of records, delayed settlements, excessive transaction charges, and lack of transparency. These limitations reduce trust among users and create inefficiencies in asset management.

Blockchain technology has emerged as a promising solution to overcome these challenges. A blockchain is a distributed ledger that stores records in a transparent, secure, and immutable manner. Once data is recorded, altering it becomes extremely difficult without network consensus. When combined with smart contracts, blockchain enables automated execution of transactions based on predefined conditions. This eliminates the need for intermediaries while increasing speed, trust, and reliability.

The proposed work introduces a blockchain-based decentralized asset management system built on Ethereum. The platform enables secure asset registration, ownership verification, listing, purchasing, and transfer. Sellers can upload and manage their assets, while buyers can purchase them using Ether. Each transaction is permanently stored on the blockchain, ensuring transparency, traceability, and authenticity. The system demonstrates how blockchain can modernize traditional asset trading processes.

Existing System

Traditional asset trading systems mainly rely on centralized databases and authority-controlled platforms. Ownership details are typically maintained by government departments, brokers, agencies, or private organizations. Transactions often involve several manual processes such as document verification, payment approval, legal validation, and ownership updates. These procedures consume time and increase complexity.

Such systems also face serious security and operational risks. Centralized databases create single points of failure, making them vulnerable to cyberattacks, insider manipulation, and accidental data loss. Since users cannot independently verify records, they must rely entirely on third parties. Hidden charges, processing delays, and limited access to transaction history further reduce user confidence. As a result, conventional asset management methods are often expensive, slow, and less secure.

Proposed System

The proposed system is a blockchain-enabled decentralized asset management platform developed

using Ethereum smart contracts. It is designed to facilitate secure, transparent, and automated trading of various assets such as digital property, land records, collectibles, and supply-chain items. Unlike traditional approaches, the system removes the requirement for brokers or centralized authorities during transactions.

The platform supports multiple user roles, including administrator, seller, and buyer. Sellers can register and upload asset details, while administrators or authorized entities verify ownership authenticity. Buyers can inspect verified assets and complete purchases using Ether. Once payment is confirmed, ownership is automatically transferred through smart contracts and recorded permanently on the blockchain. This creates a reliable and tamper-resistant asset marketplace.

Literature Survey

Functional Requirements

Satoshi Nakamoto introduced blockchain as a decentralized peer-to-peer ledger secured by cryptographic consensus. This innovation showed that trust can be achieved without central institutions and laid the foundation for decentralized asset trading systems.

Vitalik Buterin extended blockchain capabilities through the creation of Ethereum, which supports programmable smart contracts. These contracts enable self-executing agreements and form the technological core of the proposed asset transfer mechanism.

William Mougayar discussed the transformative impact of blockchain on trust, governance, and digital business models. His work supports the elimination of intermediaries and reduction of transaction overhead in decentralized marketplaces.

Gavin Wood explained the Ethereum Virtual Machine, gas model, and contract execution process. These concepts are essential for understanding how decentralized applications securely process transactions.

Michael Crosby and co-authors explored blockchain applications in supply chains, healthcare, identity management, and real-estate systems. Their findings validate blockchain as an effective solution for secure asset lifecycle management.

Konstantinos Christidis emphasized the role of blockchain and smart contracts in trusted automation across distributed environments. This directly supports the project objective of eliminating manual intervention.

Zibin Zheng analyzed blockchain architecture, consensus methods, and scalability challenges. These insights are useful for improving the performance and reliability of decentralized asset platforms.

Sara Saberi studied blockchain-enabled traceability in supply chains. Their research supports the use of immutable ledgers for verifying asset movement and preventing counterfeit transactions.

Methodology

The methodology of the proposed blockchain-based decentralized asset management system focuses on designing a secure, transparent, and efficient platform for trading both digital and physical assets. The development process includes requirement analysis, smart contract design, user role management, transaction automation, and deployment on the Ethereum network. The system architecture is divided into functional modules such as administrator control, seller operations, buyer operations, and blockchain transaction management. Smart contracts are used to validate ownership, process payments, and record transactions permanently on the distributed ledger.

The platform follows a decentralized workflow where sellers register and submit asset information, administrators verify asset authenticity, and buyers purchase approved assets using Ether. Once payment is confirmed, ownership transfer is executed automatically through smart contracts. This methodology eliminates intermediaries, reduces fraud, and ensures transparency throughout the asset lifecycle.

Design

Design is an important phase of system development in which the overall structure, workflow, and interaction between components are planned before implementation begins. It provides a clear understanding of how the proposed platform will function and how different modules will communicate with each other. A well-defined design helps developers reduce complexity, improve efficiency, and ensure that all requirements are properly addressed.

In the proposed blockchain-based decentralized asset management system, the design focuses on secure asset handling, automated transactions through smart contracts, and seamless interaction between users and the blockchain network. It illustrates how sellers list assets, how administrators verify ownership, and how buyers complete purchases using Ether. The design also explains how blockchain records are updated after each successful transaction. Diagrams are commonly used to represent these workflows in a simple and understandable manner.

Software Architecture

The software architecture of the proposed system follows a layered model consisting of the user interface layer, application layer, blockchain layer, and storage layer. Each layer performs specific tasks and works together to provide a complete decentralized trading environment.

The user interface layer is developed using web technologies such as React and allows

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

administrators, sellers, and buyers to interact with the system through a browser. Users can register, log in, list assets, browse available assets, and perform purchases using a connected wallet.

The application layer handles business logic, user requests, wallet communication, and frontend-backend coordination. It connects the user interface with blockchain services using libraries such as Web3.js or Ethers.js.

The blockchain layer consists of Ethereum smart contracts that manage asset registration, ownership verification, payment processing, and ownership transfer. Once deployed, these contracts execute automatically without manual intervention.

The storage layer is used for storing large files such as ownership documents, images, or certificates. These files may be stored off-chain, while their hash values are recorded on-chain to ensure integrity and authenticity.

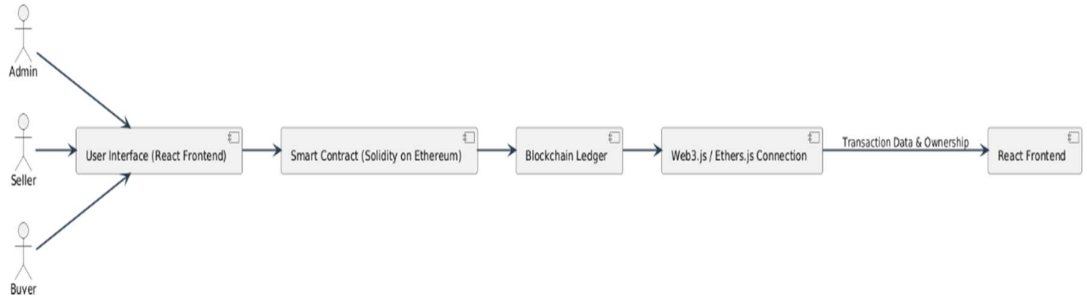


Fig.1; represents the software architecture of the system.

Technical Architecture

The technical architecture describes the technologies and infrastructure used to implement the proposed decentralized platform. It explains how frontend tools, blockchain networks, wallets, and storage systems are integrated to deliver secure asset transactions.

The frontend is built using React or similar frameworks to create a responsive and user-friendly interface. Users connect their wallets such as MetaMask to authenticate and authorize transactions.

The backend development environment uses tools such as Node.js for application services and APIs. Smart contracts are written in Solidity and deployed using Hardhat or Remix IDE.

For testing purposes, local blockchain networks such as Ganache can be used, while production deployment can occur on public or private Ethereum networks. Off-chain document storage may use distributed storage platforms such as IPFS.

The architecture ensures secure communication between the frontend, wallet, blockchain network, and storage services. It supports transparency, tamper resistance, and efficient transaction execution.

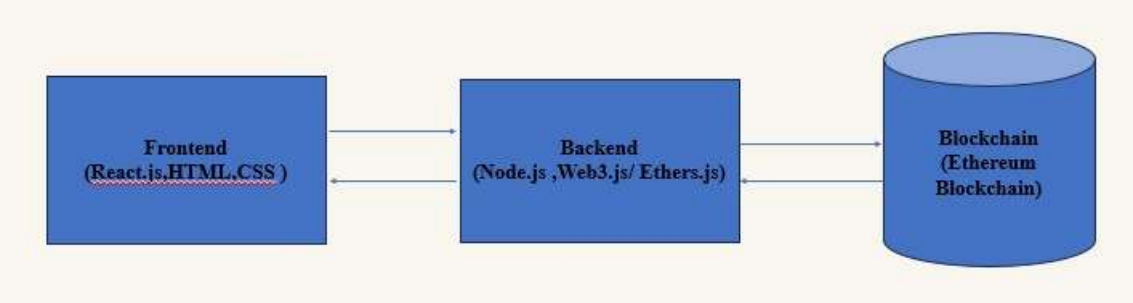


Fig.2; represents the technical architecture of the proposed system.

Implementation

Development Tools and Technologies

The implementation of the Decentralized Asset Management System is carried out using modern blockchain and web development technologies to ensure security, transparency, and tamper-proof asset management. The development environment includes Node.js, npm, and Visual Studio Code. These tools are used for configuring the project environment, managing libraries, and writing application code efficiently.

Smart contract development is performed using Solidity, which is the standard programming language for Ethereum-based contracts. The contracts are developed, compiled, tested, and deployed using Remix IDE and Hardhat. These platforms simplify blockchain development and debugging.

For blockchain integration, the system uses Ethereum as the decentralized ledger. Libraries such as Web3.js or Ethers.js are used to connect the

frontend with smart contracts. MetaMask is used for wallet authentication and transaction approval.

The frontend interface is developed using React to provide an interactive and responsive user experience. Users can view asset details, connect wallets, manage transactions, and access ownership records through the interface.

For decentralized file storage, the project uses IPFS and Pinata. Asset-related files such as ownership documents and images are stored securely on IPFS, while Pinata helps upload and manage these files reliably.

Pseudo Code

The core operations of the system are represented through structured pseudo code. During user registration, the system checks whether the user already exists in the database or blockchain records. If not, the user details are stored and registration is completed successfully.

For user authentication, the system connects to MetaMask and verifies the wallet address. If the wallet is valid, access is granted; otherwise, authentication is denied.

When an administrator adds an asset, the system first checks whether the logged-in user has administrative privileges. If authorized, the asset details are verified and stored on the blockchain. Unauthorized users are denied access.

When a buyer requests an asset, the system validates the buyer role, allows asset selection, and sends a purchase request to the seller. Once the seller accepts, the buyer transfers Ether. The smart contract verifies payment, transfers ownership to the buyer, records the transaction on the blockchain, and confirms successful completion.

For transaction verification, the system retrieves transaction details from the blockchain, displays ownership history, and confirms asset authenticity.

Implementation Steps

The practical implementation begins by creating a Hardhat project and launching a local blockchain network using the command npx hardhat node. This creates multiple test accounts and simulates an Ethereum environment for development and testing.

Next, the smart contract source file is opened in Remix IDE. After verifying the code, the contract is compiled using the Solidity compiler available in Remix. Once compilation is successful, the deployment environment is changed to a custom local network.

The smart contract is then deployed through the deploy and verify option. After successful deployment, the generated contract address is copied for integration with the frontend application.

The frontend project is opened in Visual Studio Code. The deployed contract address is inserted into the configuration file such as App.js. The project is then started using npm commands. Once the web page loads, users can click the Connect Wallet button and log in through MetaMask. After successful wallet connection, the decentralized asset management platform becomes fully operational.

Implementation Code

The system implementation is divided into multiple components. Smart contract logic is written in Solidity to manage assets, ownership, and transactions. Blockchain scripts may be written in Python or JavaScript for deployment and automation. The backend services are implemented using Node.js with Express.js to manage APIs and server-side communication. The frontend interface is developed using React for user interaction.

Testing

Software testing is a critical process used to evaluate whether the Decentralized Asset Management System meets its functional and security requirements. Since the project involves ownership transfer and financial transactions, even small errors can create serious consequences such as incorrect ownership updates, unauthorized access, or monetary loss. Therefore, thorough testing is necessary to ensure the platform is secure, accurate, reliable, and efficient.

Testing improves transparency and user trust by validating that all transactions are correctly executed. It also ensures data integrity by confirming that blockchain records remain consistent and tamper-proof. In addition, testing enhances overall system performance and confirms that all modules operate smoothly under real-world conditions.

Dimensions of Testing

Several dimensions are considered during system testing. The first dimension includes application layers such as frontend, backend, and blockchain components. These layers must communicate properly for secure asset management.

The second dimension involves testing scale, including unit testing, module testing, integration testing, and full system testing. This ensures both individual components and the complete system function correctly.

The third dimension focuses on testing types such as functional testing, performance testing, security testing, and usability testing. These tests validate system behavior, speed, safety, and user experience.

The fourth dimension is methodology, which includes both manual and automated testing. Manual testing verifies real user workflows, while automation increases efficiency for repetitive tasks.

Additional dimensions include data validation, transaction integrity, wallet authentication, and prevention of unauthorized ownership transfers.

Stages of Testing

Unit testing is used to verify individual modules such as registration, login, asset listing, and asset transfer independently. Each function is checked using valid and invalid inputs.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Integration testing ensures that multiple modules work together correctly. For example, an asset listed from the frontend must be correctly processed by the backend and stored on the blockchain.

System testing evaluates the complete platform, including user registration, wallet connection, asset management, transactions, and ownership transfer.

Acceptance testing confirms that the developed system satisfies user expectations and is ready for deployment in practical environments.

Performance testing measures transaction speed, response time, and system stability when many users perform simultaneous actions.

Types of Testing

Black box testing evaluates the application based only on inputs and outputs without examining internal code logic. Features such as registration, wallet login, asset addition, purchasing, and transaction display are validated through this method.

White box testing focuses on internal program structure and code execution. It includes testing smart contracts, backend APIs, validation conditions, and blockchain transaction logic. This helps identify logical errors, uncovered branches, and security vulnerabilities.

Test Cases

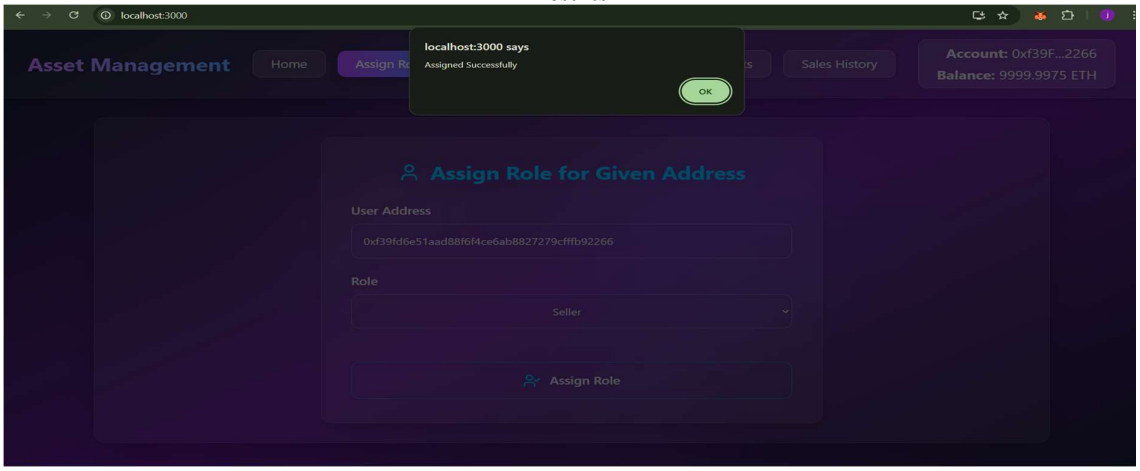
Multiple test cases were executed to validate system functionality. Wallet connection testing confirmed that when a user connects MetaMask, the wallet address and ETH balance are displayed correctly. Smart contract connection testing verified that the application successfully communicates with the deployed contract.

Administrative test cases confirmed that the admin can assign user roles such as buyer or seller and view all registered users. Seller-side testing validated successful asset listing and viewing of personal assets.

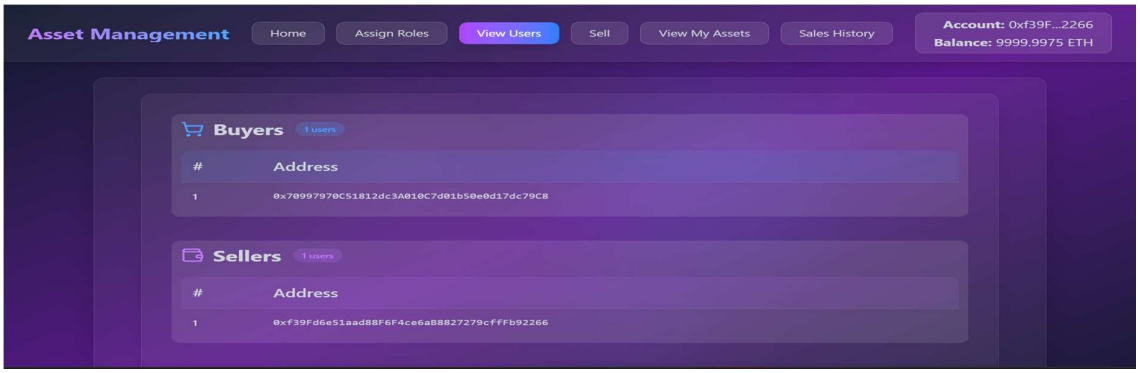
Buyer-side testing confirmed that available marketplace assets are displayed properly and purchases are completed successfully with payment confirmation and ownership transfer. Sales history was also verified to ensure previous transactions are shown accurately.

Security testing confirmed that unauthorized users attempting admin-only actions receive access denied messages. All executed test cases passed successfully, indicating that the system is functionally stable, secure, and ready for practical use.

Results

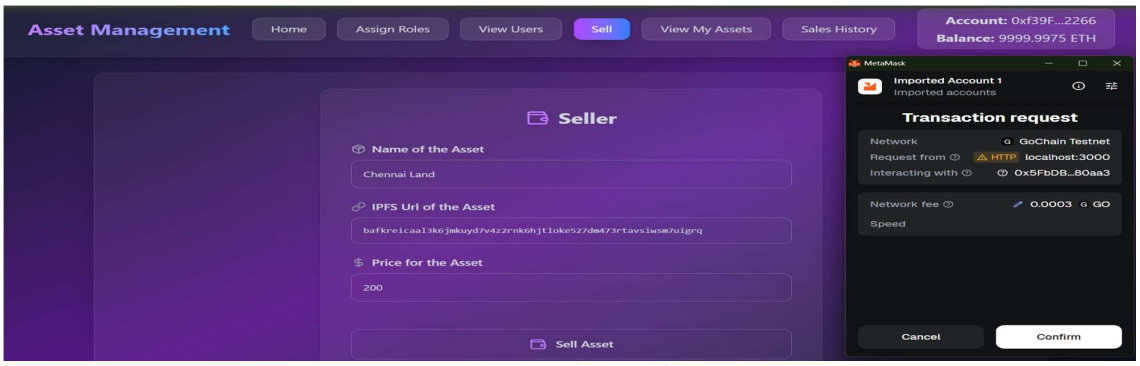


Roles assigning : Seller and Buyer

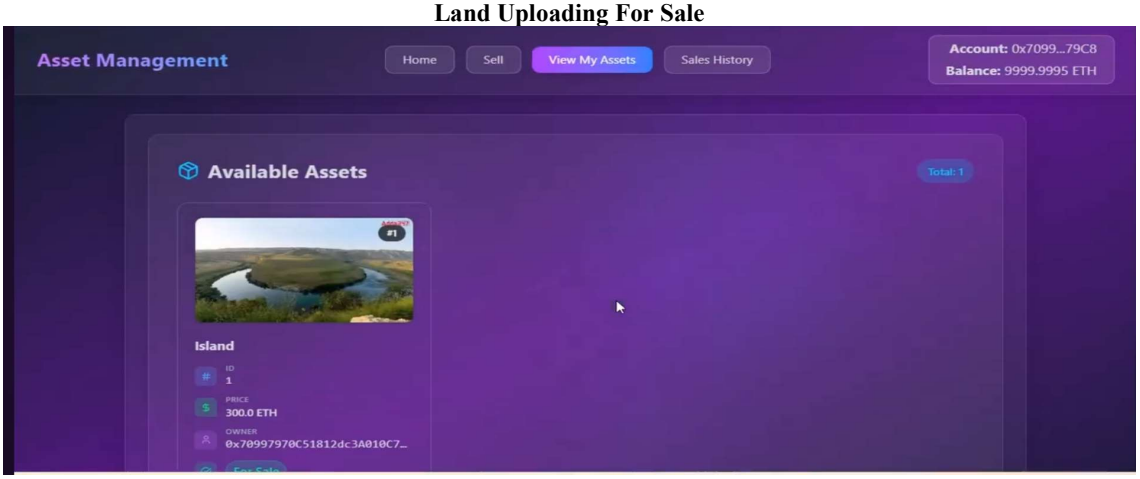


Users List

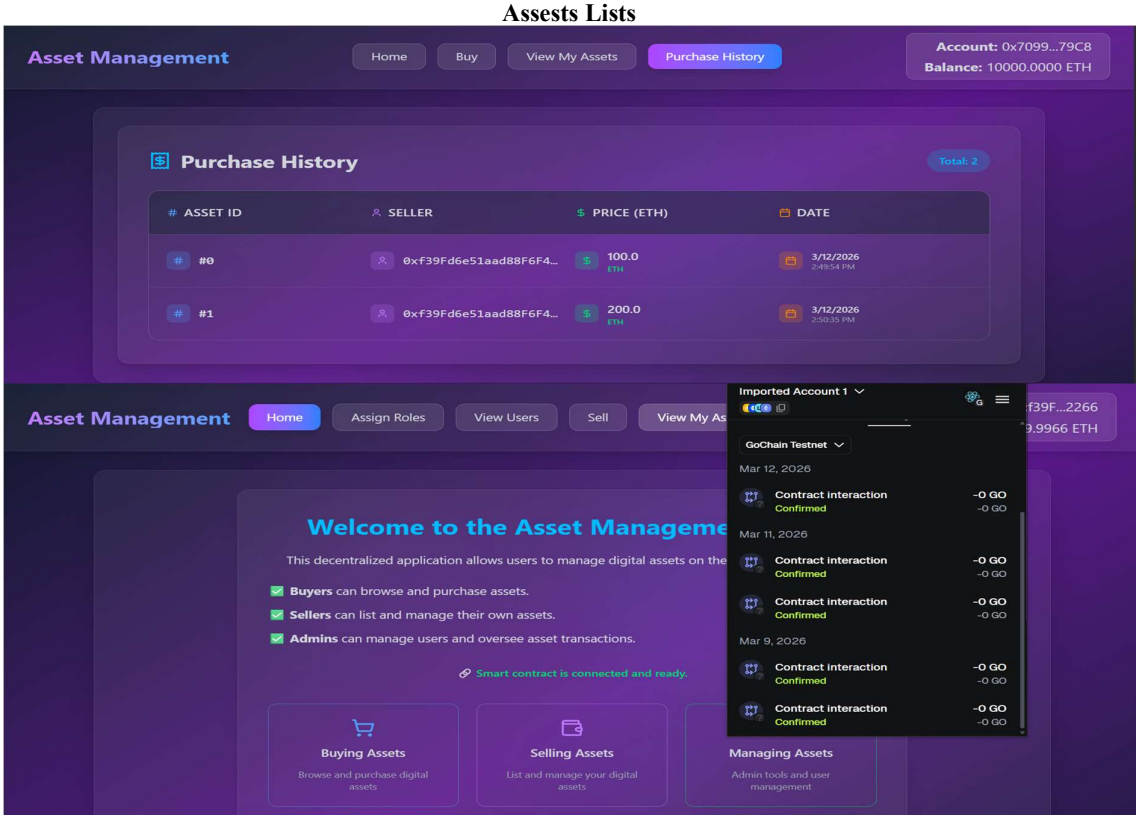
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



Land Uploading For Sale



Assests Lists



Admin can manage interactions

Conclusion

This research presented a blockchain-enabled decentralized asset management framework designed to improve the security, transparency, and efficiency of ownership transactions. Conventional asset management systems often rely on centralized authorities, resulting in higher operational costs, delayed verification processes, and vulnerability to data manipulation. The proposed system addresses these limitations through the use of smart contracts that automate asset registration, ownership transfer, and transaction validation on a distributed ledger.

The implemented AssetManager smart contract ensures that every ownership modification is permanently recorded, traceable, and resistant to unauthorized changes. The decentralized model minimizes dependence on intermediaries while enhancing trust among participants. Experimental analysis indicates that blockchain technology can significantly streamline asset handling processes and provide reliable proof of ownership.

Future Scope

Future enhancements of the proposed system may focus on integrating decentralized identity verification mechanisms to strengthen user authentication and compliance. Cross-chain interoperability can be introduced to support multiple blockchain platforms and reduce transaction costs. Integration with government or enterprise registration databases may further improve legal validity and practical adoption.

Additional improvements may include mobile-based decentralized applications for easier user access, AI-assisted fraud detection models for suspicious transaction monitoring, and advanced analytics dashboards for asset performance insights. These enhancements can expand the usability and real-world deployment potential of blockchain-based asset management solutions.

References

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

[2] V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” Ethereum Whitepaper, 2013.

[3] W. Mougayar, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. Hoboken, NJ, USA: Wiley, 2016.

[4] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” Ethereum Yellow Paper, 2014.

[5] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[6] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in *Proceedings of IEEE International Congress on Big Data*, 2017, pp. 557–564.

[7] Ethereum Foundation, “Ethereum Smart Contract Documentation,” 2023.

[8] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*. Sebastopol, CA, USA: O’Reilly Media, 2018.

[9] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, “Blockchain Technology: Beyond Bitcoin,” *Applied Innovation Review*, vol. 2, pp. 6–19, 2016.

[10] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” arXiv preprint arXiv:1407.3561, 2014.