RentalNode — Blockchain-Based Rental Management System

VARSHINI S
23MID0261
MTECH Integrated (Data science)
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
varshini.s2023@vitstudent.ac.in

SAKHANAA P
23MID0377
MTECH Integrated (Data science)
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
sakhanaa.p2023@vitstudent.ac.in

MELSHIA ATHISAYAM
23MID0361
MTECH Integrated (Data science)
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
melshia.a2023@vitstudent.ac.in

***Abstract*— The property rental ecosystem often suffers from fake agreements, payment delays, and unreliable communication between tenants and landlords. RentalNode introduces a blockchain-driven decentralized rental management framework designed to enhance transparency, integrity, and automation in rental processes. The platform integrates Ethereum smart contracts for handling digital agreements and payments, MetaMask for wallet-based authentication, and IPFS (Lighthouse) for decentralized document storage. Using this system, users can sign agreements, make rent payments, and submit maintenance requests without intermediaries. All transactions are recorded immutably on the blockchain, ensuring a secure, verifiable, and trustworthy rental experience.**

***Keywords*— Blockchain, Smart Contracts, IPFS, Rental Management, Ethereum, MetaMask, Decentralization.**

I. INTRODUCTION

Conventional rental processes rely heavily on manual documentation and centralized databases that can be manipulated or compromised. Dependence on brokers or third parties increases both cost and turnaround time. Furthermore, disputes over payments, maintenance, and authenticity of agreements are common.

Blockchain technology provides a decentralized, tamper-resistant alternative to these systems. Smart contracts facilitate automated, self-executing rental agreements, reducing the need for human intervention while preserving transparency. RentalNode applies this principle to create a platform where tenants and landlords can manage agreements, rent payments, and maintenance activities securely through a distributed network.

II. LITERATURE REVIEW

Several researchers have highlighted blockchain's utility in real estate and rental management.

Zheng et al. [1] analysed blockchain's effectiveness in preventing fraud and promoting transparency in property transactions. Kshetri [2] discussed how immutable ledgers can enhance trust and reliability in property records. Kumar and Singh [3] proposed a smart-contract-based rental framework automating agreement creation and payment flow. Li et al. [4] explored the integration of IPFS with blockchain to preserve document authenticity and accessibility.

Building on these studies, RentalNode merges blockchain automation with decentralized storage to design a rental management solution that emphasizes efficiency, accountability, and user confidence.

III. METHODOLOGY

A. System Architecture

The proposed system contains the following core components:

1. *Frontend (Web DApp):* Built with HTML, CSS, and JavaScript; interacts with the blockchain through Ethers.js.

2. *Smart Contracts (Solidity):* Manage user accounts, rental terms, payments, and feedback securely on Ethereum.
3. *Blockchain Network (Hardhat Local):* Used for development, compilation, and contract deployment.
4. *MetaMask*: Enables wallet-based login and transaction authentication.
5. *IPFS (Lighthouse):* Stores documents such as agreements and maintenance proofs securely off-chain while preserving on-chain hashes.

TABLE I. TECHNOLOGY STACK OVERVIEW

Layer	Technology	Purpose
Frontend	HTML, CSS, JavaScript	User Interface
Middleware	Ethers.js, Hardhat	Blockchain communication
Blockchain Layer	Ethereum (Hardhat, Localhost)	Smart contract execution
Storage Layer	IPFS (Lighthouse)	Decentralized file storage
Authentication	MetaMask	Secure wallet-based login

B. Smart Contract Workflow

1. *User Registration*: Tenants and landlords register via MetaMask, creating an on-chain identity.
2. *Agreement Creation*: Landlords define terms such as rent, deposit, and duration.
3. *Agreement Signing*: Tenants digitally sign contracts through wallet authorization.
4. *Rent Payment*: Rent is paid in ETH and recorded through blockchain transactions.
5. *Document Storage*: All uploaded documents are stored on IPFS, while reference hashes remain immutable on-chain.

IV. IMPLEMENTATION

Development was carried out using the Hardhat framework for compiling and deploying Solidity contracts. MetaMask provided wallet connectivity for blockchain interaction, and Lighthouse IPFS ensured persistent, decentralized file storage.

The user interface includes several functional modules:

- *Authentication Module*: Handles blockchain login and wallet linkage.
- *Rental Agreement Module*: Manages creation, signing, and tracking of agreements.
- *Payment Module*: Facilitates rent transfers between users.
- *Maintenance Module*: Logs and stores maintenance requests using IPFS.
- *Profile Module*: Displays user data synchronized with blockchain state.

TABLE II. SMART CONTRACT FUNCTIONALITIES

Function Name	Purpose	Access Level
createOrUpdateUser()	Registers or updates user details.	Public
createAgreement()	Creates new rental agreement by landlord.	Landlord Only
signAgreement()	Signs agreement from tenant's wallet.	Tenant Only
getUser()	Fetches user profile data.	Public
getAgreement()	Retrieves rental agreement details.	Public

V. RESULTS AND DISCUSSION

Testing on the Hardhat local blockchain demonstrated successful end-to-end interaction between tenants and landlords. Both parties were able to create profiles, execute agreements, and complete payments without centralized oversight.

Key outcomes:

- Agreements were recorded immutably on the blockchain.
- Rent transactions were transparent and traceable.
- Maintenance requests were stored on IPFS and accessible through content hashes.
- Event logs allowed full transaction auditing directly from the blockchain.

The decentralized approach improved data security and reduced the risk of tampering. Integration with MetaMask offered users a seamless, wallet-based experience.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

RentalNode provides a working prototype for a blockchain-enabled rental management system. By combining smart contracts and decentralized storage, the solution enhances trust, automation, and accountability in rental operations. The immutability of blockchain ensures that agreements and payments remain verifiable throughout the lifecycle of the rental process.

B. Future Work

Planned enhancements include:

- Deployment on public testnets such as Sepolia or Polygon Mumbai.
- Integration of Chainlink Keepers for automated payment reminders.
- Enabling stablecoin payments (DAI, USDT) for real-world usability.
- Incorporating decentralized identity (DID) for verification and access control.
- Developing a mobile interface with push notifications and simplified interaction.

ACKNOWLEDGMENT

The authors extend appreciation to the faculty of VIT Vellore for their continuous guidance and encouragement throughout the project. Gratitude is also expressed to mentors, peers, and the open source blockchain community for their technical advice and contributions during the design and implementation of RentalNode.

REFERENCES

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “Blockchain Challenges and Opportunities: A Survey,” *International Journal of Web and Grid Services*, 2018.
[2] N. Kshetri, “Blockchain’s Roles in Strengthening Cybersecurity and Protecting Privacy,” *Telecommunications Policy*, 2020.
[3] A. Kumar and R. Singh, “Smart Contracts for Rental Systems on Blockchain,” *IEEE Access*, 2021.
[4] Y. Li, J. Wang, and D. Zhang, “Decentralized Storage Systems with Blockchain and IPFS Integration,” *ACM Digital Library*, 2022.