BLOCKCHAIN BASED PROPERTY MANAGEMENT SYSTEM

Project By

Paul Titus | Riken Kiri | Chetan Bhangare | Chrisitan George | Hasan Gandor

Project Overview

Objectives

- Revolutionize property management by leveraging blockchain technology for secure, transparent, and automated property transactions.
- Streamline property registration, transfer processes, and compliance through smart contracts.
- Eliminate intermediaries and ensure accuracy by integrating real-time data via MongoDB.

Key features

- Property Registration: Verified ownership and secure, transparent registration.
- Property Transfers: Multi-signature approval for secure ownership transfers.
- Real-Time Data: Use of MongoDB for real-time property metadata updates (e.g., market prices, zoning laws).

Tech Stack & Tools

- **Blockchain**
  Etherium (Sepolia Testnet)
- **Smart Contracts**
  Solidity (OpenZeppelin libraries)
- **Backend**
  Node.js , Express.js , MongoDB
- **Frontend**
  Vue.js , Web3.js , Ethers.js
- **Deployment & DevOps**
  Truffle , Docker , Alchemy , Chainlink

Smart Contract Design

Core Functionalities:

Property Registration & Allotment:

Authorities register verified properties with essential details to ensure secure and accurate records

VERIFIED

Transfer request & approval:

Property transfers require multi-signature approvals for legitimacy and compliance

Marketplace Integration:

Owners list properties for sale, and buyers purchase with automatic ownership transfer via ERC721 tokens

Blockchain Benefits

Security

Cryptographic algorithms protect data, ensuring only authorized users can interact

Trust & Accessbility

Decentralized records are accessible to all stakeholders, building trust in the property lifecycle

Transparency

Every transaction is recorded on an immutable ledger

Efficiency

Smart contracts automate processes like property transfers and approvals

Backend & Database Design

Framework:

- Express.js is used as the backend to handle API requests, routing, and business logic.
- It enables seamless communication between the frontend, blockchain, and MongoDB database, ensuring accurate property data processing.

Database:

- MongoDB stores off-chain property metadata, including property descriptions, images, and owner details.
- It supports fast and flexible querying, enabling features like property searches, status tracking, and historical data management.

Environemnt Management:

- dotenv is used to manage sensitive environment variables such as database connection strings, blockchain keys, etc

Frontend & User Interaction

Framework:

Built with Vue.js, using Web3.js or Ethers.js to interact with the Ethereum blockchain. This setup ensures a responsive and dynamic user experience

Wallet Integration:

MetaMask enables secure blockchain interactions, allowing users to connect their wallets, sign transactions, and manage property ownership directly from the app

UI Features:

Users can browse and search property listings, view details, and filter results. Role-based registration allows authorized users to add properties, initiate transfer requests, approve transactions, and track real-time status updates

Testing & Security

Comprehensive Testing Approach

- Validate role-based access control to ensure only authorized users can execute specific contract functions.
- Test Oracle data integrity to confirm the accuracy and reliability of external property data.
- Simulate reentrancy attacks and edge cases in property transfers to identify and mitigate security risks.

Security Measures

- Implement strict access control to ensure only designated roles (e.g., Approvers, Buyers, Sellers) can perform specific actions.
- Utilize reentrancy guards to safeguard contract functions from recursive exploit attempts.

Reliability & Robustness

- Ensuring accurate and tamper-proof property data retrieval using MongoDB for secure storage.
- Secure multi-signature approvals for property transfers to prevent unauthorized transactions.
- Implementing fallback mechanisms to enhance contract security and prevent failures.
- Conducting rigorous stress testing to validate system stability under high transaction loads.

Deployment & Dockerization

Deployment Workflow

- Smart contract deployed on Sepolia testnet for decentralized interactions
- Compilation: Converts Solidity code into EVM-executable bytecode
- Deployment: Migrates contract to Sepolia for on-chain execution
- Verification: Confirms deployment on Sepolia Etherscan

Docker Setup

- Containerization: Ensures a consistent environment for deployment
- Dockerfiles: Separate configurations for frontend (Vue.js) and backend (Express.js, MongoDB)
- Orchestration: Docker Compose manages MongoDB and services
- Single-Step Deployment: Runs all services in isolated containers efficiently

Achievement & Conclusion

Achievement:

- Developed a full-stack decentralized property management system
- Implemented secure, automated property transactions using blockchain
- Designed role-based access control for secure property transfers
- Utilized MongoDB for efficient off-chain property metadata storage

Challenges:

- Complex Role-Based Access Management: Ensuring seamless user authentication and role verification.
- Backend-Blockchain Synchronization: Keeping MongoDB in sync with blockchain events in real time.
- Web3 Integration Issues: Handling compatibility across different wallets and browsers.
- Testing and Debugging: Writing extensive test cases to cover edge cases and security vulnerabilities.
- Frontend-Backend Communication: Managing API calls efficiently while interacting with smart contracts.

Future Enhancements & Opportunities

AI-Driven Property Valuation:

- Leverage AI algorithms to predict property values based on historical market trends.
- Improve property valuation accuracy with real-time data and market sentiment analysis.

Decentralized Identity Verification:

- Use decentralized identifiers (DIDs) to create verifiable digital identities for all parties involved.
- Enhance security and reduce fraud risks by ensuring that identity verification is immutable and transparent.

Cross-Chain Interoperability:

- Enable seamless property transfers across different blockchain networks to increase platform flexibility.
- Expand the user base by providing access to a broader market through multi-chain capabilities.

LIVE DEMONSTRATION

THANK YOU

Any questions?