Blockchain Asset Manager

Securing Digital Asset Ownership Using Blockchain

Abstract

The Blockchain Asset Manager is a decentralized web application that enables users to register, view, and verify ownership of digital assets using blockchain technology. The system leverages Ethereum smart contracts and IPFS for decentralized storage, ensuring assets are tamper-proof, easily auditable, and securely linked to user wallets.

Introduction

Digital assets, such as artworks, academic records, and certificates, are susceptible to theft, forgery, and tampering. Traditional storage methods lack transparency and are centralized, making them vulnerable to manipulation. Blockchain offers a decentralized alternative by storing asset data on an immutable ledger. This project aims to build a user-friendly platform where users can:

- Mint new digital assets.
- Associate them with a content hash or CID (e.g., from Pinata/IPFS).
- Securely manage asset metadata/description.
- Verify ownership through Ethereum wallets (e.g., MetaMask).

The overarching goal is to demonstrate how blockchain can democratise asset control and prevent digital fraud.

Methodology

Design Decisions

- **Ethereum Blockchain** was chosen for its mature tooling and smart contract support.
- **React + Vite** for the modern frontend.
- **IPFS via Pinata** allows users to upload assets and store a CID as proof.
- **Ethers.js** is used to interface with the blockchain from the frontend.

System Architecture

- **Frontend**: React app with wallet integration via MetaMask.
- **Smart Contract**: Solidity-based ***AssetRegistry*** deployed on Ethereum testnet.

- **Storage**: Users upload assets to Pinata and store only the CID on-chain.
- **Interactions**: All minting and querying is done through wallet-signed transactions.

Implementation

Smart Contract: AssetRegistry.sol

- mintAsset(...): Registers an asset tied to a wallet.
- getHolderTokens(...): Fetches token IDs for a wallet.
- getAssetInfo(...): Returns full asset metadata.
- Events are emitted for UI updates and blockchain logs.

Frontend UI (React + Vite)

Mint Asset Page
Users input:

- Title
- Metadata
- Storage ID (Pinata CID)
- Content Hash

Asset List
Displays assets owned by the connected wallet.

Asset Details Page
Includes all metadata and timestamps, and an update feature.

Smart Contract Interaction (ethers.js)

```
const contract = new ethers.Contract(address, abi, signer);
await contract.mintAsset("Certificate", "Issued 2024", "Qm123...", "hash123...");
```

How to Use

1. **Upload your asset** (PDF, image, etc.) to Pinata
2. **Copy the CID**
3. **Open the Asset Registry**:
   🔗https://assetregistry.onrender.com/
4. **Connect MetaMask**
5. **Mint your asset** by providing a title, metadata, CID, and optional hash.

GitHub Repository

🔗 https://github.com/GChukwudi/digital_asset_manager