Table of Contents

Technical Documentation for NFT Marketplace on Cardano Blockchain

Project Overview

This document outlines the technical architecture and implementation details for developing an NFT (Non-Fungible Token) marketplace on the Cardano blockchain. The marketplace will enable users to mint, buy, and sell NFTs seamlessly. The design is based on the provided Figma prototype, and the development will utilize the following technology stack:

- Frontend: React, Tailwind CSS, Redux
- Backend: Node.js, Express, MongoDB
- Blockchain Integration: Mesh SDK Core, Mesh Contract, Lucid.js

System Architecture

The system is structured into three primary components:

1. **Frontend**: A React-based single-page application (SPA) styled with Tailwind CSS, managing state using Redux.
2. **Backend**: An Express server running on Node.js, interfacing with a MongoDB database for data persistence.
3. **Blockchain Integration**: Interaction with the Cardano blockchain facilitated through Mesh SDK and Lucid.js.

Component Interaction

- **User Interface**: Users interact with the React frontend to perform actions such as browsing NFTs, connecting wallets, and initiating transactions.
- **State Management**: Redux manages the application state, ensuring a predictable data flow.
- **API Communication**: The frontend communicates with the backend via RESTful APIs to fetch and persist data.
- **Blockchain Operations**: The backend utilizes Mesh SDK and Lucid.js to interact with the Cardano blockchain for minting, buying, and selling NFTs.

Detailed Component Descriptions

Frontend

- **React**: Facilitates the development of a dynamic and responsive user interface.
- **Tailwind CSS**: Provides utility-first CSS for rapid and consistent styling.
- **Redux**: Manages global state, ensuring consistency across components.

Key Features

- **Wallet Integration**: Users can connect their Cardano wallets to the application.
- **NFT Browsing**: Display available NFTs with details such as images, descriptions, and prices.
- **Transaction Initiation**: Interfaces for users to mint new NFTs or purchase existing ones.

Backend

- **Node.js**: Provides a scalable runtime environment for the server.
- **Express**: Simplifies the creation of RESTful APIs to handle client requests.
- **MongoDB**: Stores user data, NFT metadata, and transaction histories.

API Endpoints

- **User Management**: Endpoints for user authentication and profile management.
- **NFT Operations**: Endpoints to create, retrieve, update, and delete NFT records.
- **Transaction Handling**: Endpoints to initiate and verify blockchain transactions.

Blockchain Integration

- **Mesh SDK Core**: Offers APIs for wallet integration, transaction building, and smart contract interactions on the Cardano blockchain.
- **Mesh Contract**: Provides pre-built smart contracts for common use cases, including NFT marketplaces.
- **Lucid.js**: A library for creating and signing transactions on Cardano, facilitating seamless blockchain interactions.

Blockchain Operations

- **Minting NFTs**: Utilize Mesh SDK to create new NFTs on the Cardano blockchain.
- **Buying/Selling NFTs**: Implement smart contracts to handle the transfer of NFTs between users securely.
- **Transaction Monitoring**: Use Lucid.js to monitor and confirm the status of blockchain transactions.

Development Workflow

- Environment Setup:
  - Install Node.js and npm.
  - Set up a MongoDB database.
  - Initialize the React project with Tailwind CSS and Redux.
- Frontend Development:
  - Develop React components based on the Figma design.
  - Implement Redux for state management.
  - Integrate wallet connection features using Mesh SDK.
- Backend Development:
  - Set up the Express server with necessary middleware.
  - Define API routes for user and NFT operations.
  - Implement database schemas and models in MongoDB.
- Blockchain Integration:
  - Configure Mesh SDK and Lucid.js for Cardano interactions.
  - Develop functions to mint NFTs and handle transactions.
  - Deploy and test smart contracts using Mesh Contract.
- Testing and Deployment:
  - Conduct unit and integration tests for all components.
  - Deploy the backend server and database to a cloud platform.
  - Host the frontend application and ensure proper environment configurations.

Security Considerations

- Authentication: Implement robust user authentication mechanisms to protect user accounts.
- Data Validation: Validate all inputs on both frontend and backend to prevent injection attacks.

- Blockchain Security: Ensure smart contracts are audited and tested to prevent vulnerabilities.

Resources and References

- Mesh SDK Documentation: Comprehensive guides and API references for integrating with the Cardano blockchain. Link to Mesh sdk docs
- Lucid.js Repository: Source code and documentation for Lucid.js, a library for Cardano transactions. Link to Lucid Cardano docs
- Figma Design: Link to Figma Design

This documentation serves as a foundational guide for developing the NFT marketplace, detailing the architecture, components, and development workflow necessary to build a functional and secure platform on the Cardano blockchain.