FACULTY OF SCIENCE AND ENGINEERING

MASTER 2 NETWORKING DEVOPS

DAAR

Project 2 – Collectible Card Game

BM Bui-Xuan and Guillaume Hivert

Created By:
Alaoui Belghiti Hanaa - 21410228
Sokhna Khadijatou BA – 21220680
Mekkhishan ARULANANTHAM – 21318636

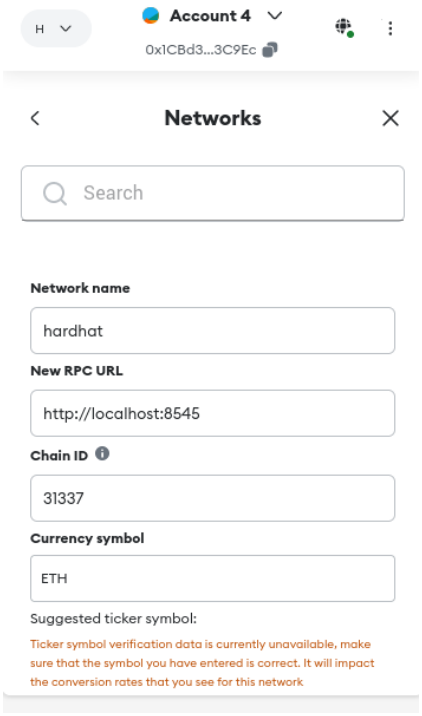
Contents

I. Introduction:

The Pokémon NFT Project bridges the world of Pokémon trading cards with blockchain technology. Built on Ethereum, this decentralized application (dApp) enables users to mint, trade, and collect unique digital Pokémon cards as NFTs (Non-Fungible Tokens). By combining elements of blockchain, smart contracts, and user-friendly frontend integration, this project provides a platform for Pokémon card collectors and traders, creating a seamless, interactive digital experience.

II. MetaMask Integration:

One of the main features of our project is the integration with MetaMask. We started by setting up a localhost at **localhost:8545** on MetaMask to ensure secure and seamless user connections.



We used public users to test the interactions, ones listed after starting the hardhat node:

Steps for Integration:

- **Connection Setup:** Users can connect their MetaMask wallet upon visiting the site. This is achieved by checking for the **window.ethereum** variable, which is injected by MetaMask. If available, it confirms that MetaMask is installed and accessible by the application.
- **Transaction Signing:** All transactions, such as minting and trading, are securely signed using MetaMask. MetaMask injects the **window.ethereum** object into the browser, enabling the application to request transaction signing from the user securely.
- **Account Change Detection:** The project successfully detects when users change their accounts, ensuring a smooth experience. This is done by setting up an event listener on the **window.ethereum** object to listen for account changes. When a change is detected, the application updates the state and triggers a re-render to reflect the new account information.

Our code listens for changes using an **useEffect** hook, which adds an event listener for **accountsChanged** events. When an account change is detected, the application updates the account state and toggles a state variable to trigger a re-render, ensuring that the UI reflects the current account information.

III. Smart Contracts:

The backbone of the project lies in the **smart contracts** developed in Solidity, which manage the minting, transfer, and ownership of the Pokémon NFTs. These contracts were deployed on the Hardhat network, allowing for efficient development and testing.

Key Functions:

- **`mintCard(address to, string memory tokenURI):`** This function allows users to mint a new Pokémon card NFT, assigning a unique `tokenURI` that points to the card's metadata (image, name, and attributes).
- **`transferCard(address from, address to, uint256 tokenId):`** This function facilitates the secure transfer of Pokémon cards between users by specifying the `tokenId` and the destination address. The smart contract validates the ownership before processing transfers, ensuring only authorized transfers.

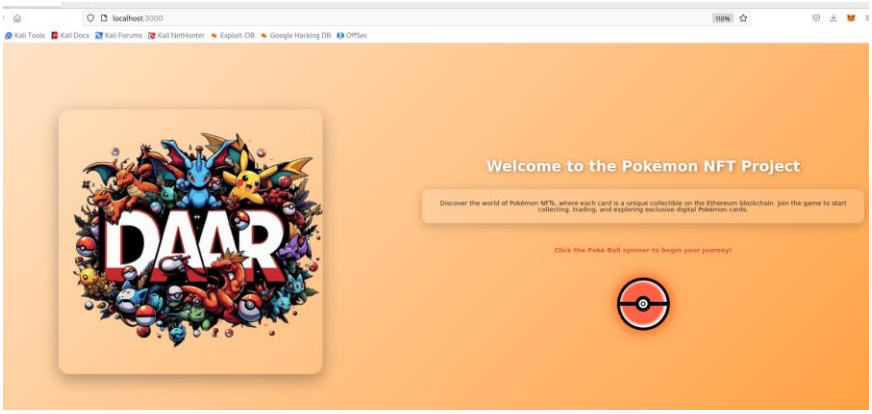
These functions are fundamental to the application's logic, making each card unique and transferable on the Ethereum blockchain.

IV. Frontend Development:

Our frontend is built using React, providing a dynamic and interactive user experience. Users can view their cards, mint new ones, and explore the marketplace.

Main Features:

Home Page: A captivating home page with animations, guiding users to discover the project. A Pokémon spinner animation was added for an engaging user experience.



Et:

• **Card Viewing:** Displays users' Pokémon cards, retrieving attributes from the smart contract and metadata storage.

• **Minting:** An interface for creating new cards, with form input and MetaMask transaction confirmation.

• **Marketplace**: A dedicated section for buying and selling Pokémon cards.

• **Booster Packs:** Organized in a “binder” format, where users explore packs and their cards.

Let's Go Pink Theme: Inspired by the original project, we used a vibrant "Let's Go Pink" theme to enhance the visual appeal.

V. Backend Development:

The backend, developed using Node.js and Express, handles server-side logic and database interactions. We used MongoDB to store metadata related to each Pokémon card.

Key Endpoints:

• `POST /mint`: Mints a new Pokémon card, interacting with the smart contract and saving card metadata to MongoDB.

• `GET /cards`: Retrieves a user’s card collection from MongoDB, displaying each card’s metadata.

• `POST /trade`: Facilitates trading by checking card ownership, updating MongoDB, and ensuring blockchain-based transfer between accounts.

VI. API Integration:

The **Pokémon TCG API** provides real Pokémon card data, including images and set details, which are used to create NFTs.

Integration Steps:

- **Set Selection**: Specific sets were chosen from the API to match real Pokémon card collections.
- **Card Count**: Maintains correct card counts for each set.
- **API Modification**: Enhanced API calls to fetch card names, images, and values, storing data in JSON format for NFT mapping.

This integration enhances the dApp’s realism by representing actual Pokémon cards in the NFT realm.

VII. Challenges and Solutions:

While developing the Pokémon NFT Project, several complex challenges emerged, along with key areas for future improvement to enhance user experience and project functionality.

1. Current Challenges:

- **Smart Contract Security**: Ensuring secure interactions to prevent unauthorized access was a primary concern. Rigorous testing, code audits, and best practices in Solidity helped optimize the security of critical functions like minting and transferring cards.
- **Frontend-Backend Sync**: Achieving smooth interaction between the frontend and backend, especially with blockchain transactions, required careful React state management and real-time UI feedback.
- **User Interface Design**: Creating an intuitive and engaging UI to simplify blockchain interactions required adding tooltips, animations, and user-friendly layouts, making the app accessible to both beginners and seasoned collectors.

2. Advanced Challenges and Future Enhancements

To take the project further, several ambitious features and enhancements could greatly expand its functionality and user appeal:

- **Collection Management**: Enhancing the frontend with a "binder" feature to help users manage their Pokémon collections would improve organization. This would mimic real-world card binders, making collection browsing and management more interactive and visually engaging.
- **Multichain Codebase**: Developing a **multichain-capable contract** to allow cross-chain interactions would enable broader reach, allowing NFTs to exist and be traded across various blockchains. Implementing a **bridge** between chains could facilitate this, making it possible to seamlessly move assets between networks.
- **Game Engine for Battles**: Integrating a **game engine** to enable card battles, where users could battle using Pokémon cards with specific energy points or attributes, would add an entirely new dimension. This would also involve designing rules and card attributes to support strategic gameplay, enhancing engagement.
- **NFT Marketplace**: Creating an in-app **marketplace** would allow users to exchange cards with each other, fostering a community-driven economy. This marketplace could be configured to support auctions, direct trades, and listings for rarer Pokémon cards, further enriching the dApp ecosystem.
- **Tournament Mode and Promotional Cards**: Establishing a **tournament system** could promote competitiveness, where users compete in organized battles or trading tournaments. Winners could be rewarded with **promotional cards** that would serve as exclusive collectibles, adding value and incentivizing participation.

VIII. Conclusion and Future Work

The Pokémon NFT Project successfully brings together blockchain technology and the excitement of Pokémon card collecting into a single dApp. The planned future updates—like a collection binder, multichain compatibility, card battles, and a user-driven marketplace—will continue to enrich the user experience and grow the platform's community.

IX. Appendix:

Link to Code:

https://github.com/xorya1/BitcoinCards

X. Resources:

- Solidity Documentation:

https://docs.soliditylang.org/en/latest/contracts.html

- MetaMask Integration Guide:

https://docs.metamask.io/wallet/reference/provider-api/

- Pokémon TCG API Documentation:

https://pokemontcg.io/