BRAUDE
College of Engineering, Karmiel

HD Wallet - Final Blockchain Project

Blockchains and Cryptocurrencies (61914)

Presenting:

Tomer Meidan - 204750459

Roman Milman - 320942808

Link - https://github.com/TomerMeidan/Blockchain-Project

Summary

The project is a toolkit for managing Bitcoin and Ethereum wallets, with features for creating wallets, managing private and public keys, and interacting with the respective blockchains for transaction-related activities. The backend services facilitate these functionalities, while the frontend, though not detailed in this summary, provides a user interface for these operations, making it accessible for users to manage their cryptocurrency assets securely and efficiently.

Here’s a summary of the features and subjects the project covers.

Bitcoin Wallet Features

1. Utilizes `bip39` for mnemonic (seed phrase) generation, which is crucial for creating a new wallet securely.
2. Implements `tiny-secp256k1` and `BIP32Factory` from `bip32` for cryptographic functions and hierarchical deterministic (HD) wallet creation.
3. Employs `bitcoinjs-lib` and `bitcore-lib` for constructing, signing, and broadcasting Bitcoin transactions. There’s also an integration with the BlockCypher API (as indicated by the `block_cypher_api` environment variable), likely for querying blockchain data and transaction broadcasting.

Ethereum Wallet Features

1. Similar to the BTC wallet functionality, it provides means to generate new wallets using a randomly generated seed phrase. It leverages `ethers.js`, a popular library in Ethereum development for wallet creation and interaction with the Ethereum blockchain.
2. Configured to interact with both Ethereum mainnet and testnets (specifically Sepolia, as indicated by the API URLs), using API keys for Alchemy, a blockchain infrastructure provider. This suggests capabilities for sending transactions, querying network data, and possibly smart contract interaction on Ethereum.

Frontend Setup

Step 1: Navigate to the Frontend Directory

Open your terminal and navigate to the frontend directory of the project.

```
cd frontend
```

Step 2: Install Dependencies

Within the frontend directory, install the necessary packages using npm.

```
npm i
```

Step 3: Configure Environment Variables

Create a `.env` file in the frontend directory and set the `VITE_BACKEND_URL` variable to point to your backend server.

```
VITE_BACKEND_URL=http://localhost:3000
```

Step 4: Start the Frontend Application

Run the following command to start the React application.

```
npm run dev
```

Backend Setup

Step 1: Navigate to the Backend Directory

Open your terminal and navigate to the backend directory of the project.

```
cd backend
```

Step 2: Install Dependencies

Within the backend directory, install the necessary packages using npm.

```
npm i
```

Step 3: Configure Environment Variables

Create a `.env` file in the backend directory and set the `alchemy_api_key` and `block_cypher_api` variables with your respective API keys.

```
alchemy_api_key=your_alchemy_api_key_here
block_cypher_api=your_block_cypher_api_key_here
```

Step 4: Start the Backend Application

Run the following command to start the backend server.

```
npm run dev
```

Note:

For mainnet transaction changes, you need to modify the `backend/ethWallet.js` file. Specifically, change `ethTestNet` to `ethMainNet` on lines 57 and 72.