Crypto Student Internship

Week 2

Task

1. Build a blockchain explorer UI

This week, we are going to build the frontend part of the blockchain explorer application we will build during this internship.

Your task is to implement the following functionalities of the application focusing just on the UI part, without any backend or blockchain logic.

All addresses and Ether holdings of each address are just fake and virtual for now.

Login page

This page should allow users to log in to the application to be able to use it. It should be the root URL of your web application. Users should not be able to access any of the other routes and application parts before logging in.

By the end of the internship, we will implement Metamask login so the user will authenticate himself using his wallet. Since we are not implementing any of the blockchain related logic at the moment, we will implement the most primitive login.

Your login page should consist of the following:

App logo
Come up with an application logo. Whether it is any crypto related sign like Ethereum logo or anything you want to use, place it on top of the login page.

Address input
Input text field where user can input his Ethereum wallet address. Try to find out how to validate the input after the “Login” button click and make sure the user entered a valid Ethereum address. Validation is not mandatory, but it is definitely a plus.

Login button
Login button should allow the user to actually log in to the system. Once logged in, the user should be issued a cookie and he should not be able to access the login page anymore (by going to the root URL) until he is logged out.

User dashboard

After the user is logged in, he lands on the dashboard page on /dashboard URL.

Here, he should be presented with three options:

Explore
Clicking this option should open up a new page on /explore URL where the user will be presented with an input field and “Explore” button.

User should be able to enter an Ethereum address in the input field and after clicking the “Explore” button, he should be presented with the amount of Ether the address is holding (up to 8 decimal places).

There should also be a “Back” button returning the user to the dashboard.

Mint
Clicking this option should open up a modal where the user will be presented with the amount of Ether his address is holding, an input field and “Mint” button.

Input field should accept numbers only, up to 8 decimal places. After clicking the “Mint” button, the value of Ether on the address the user is logged in with should be increased by the amount entered in the input field.

There should also be a “Back” button returning the user to the dashboard.

Transfer
Clicking this option should open up a modal where the user will be presented with the amount of Ether his address is holding, two input fields and a “Transfer” button.

User should be able to enter an Ethereum address in the first input field, the amount of Ether he wants to transfer to that address (number up to 8 decimal places)

After clicking the “Transfer” button, he should either be shown a success message if the entered amount is less or equal to the current user address Ether holdings or the error message if the entered amount is greater than the current user address Ether holdings.

If the transfer is successful, address holdings should update.

There should also be a “Back” button returning the user to the dashboard.

Additional notes

Keep all the addresses in the local storage. When a user explores a new address, it should be added to the local storage if it does not exist and its Ether holdings should be zero.

All Ether holdings on any of the addresses are just virtual for now. We are not connecting to the Ethereum network at any point. Minting new Ether for now just means we will increase the value of Ether for the specific address in our local storage.

After minting new Ether to the current user’s address or transferring them to another address, all balances should be updated in the local storage.

If a user tries to transfer Ether to the address that has not been explored before and therefore it does not exist in the local storage, it should be added at that moment and its Ether holdings should be equal to the transferred amount.

Pay attention to the UI and your CSS, try your best to make the app look nice.

Good luck!