MEMBRANE

Senior Frontend Engineer - Code Evaluation

The goal of this project is to create a web application that connects to Ethereum blockchain to retrieve token balances. The coding language should be Javascript, and you're free to implement the solution however you see fit but you should focus on state management.

Problem Specification

- Use React or NextJs frameworks to set up the project.
- You can design a simple functional UI which should allow the user to:
  - log in using Metamask using a header button
  - the button should be replaced by user address once logged
  - the screen should be practical and simple, with a token list and their user balances. It should also display the total balance in USD.
  - a plus: use Ant Design, Material UI, or your favorite library to design the components. Consider responsiveness and implementation of sass
- The tokens to list balances should be ETH, USDC and USDT. Show both current balance and underlying balance on Compound protocol. You could read balances from Compound using Compound API or reading the smart contract directly, but in this challenge we ask you to explicitly read the smart contract. Reference to documentation. Use Ethers.js library for web3 connectivity.
- The UX flow should be the following:
  - The first time the user logs into the webapp, it needs to login. Login only admits metamask login. Which means, the user is identified by Metamask wallet connected address.
  - When the user logs in, if he is not connected to Ethereum blockchain, a prompt modal should ask him to change the network.
  - Once the user is logged in on Ethereum blockchain, the UI should display token balances and Compound underlying balances, only for ETH, USDC and USDT.
  - The user should be able to close connections at any time.
- Store data using one of the following state management libraries: redux, mobx or mobx state tree. (Plus points using mobx state tree).
- Show total balance in USD using market prices from a third party public API like coinmarketcap, bittrex, bitfinex, etc.
- Submit your solution to a private github repository. Keep the organization of branches and commits as if you were working in a team. Give access to users:
  - daniellenarduzziDB
  - diegodhh
  - fedecaccia
  - francomangussi
  - juanmavillarrazadb
  - liccampanavd
  - plarrea