Product Eng Screening

Take Home Task

Objective

The goal of this assignment is to assess your frontend development skills, particularly with React, your ability to interact with APIs, your understanding of component design, and your product sense in translating requirements into a working application. You will build a simplified interface that allows users to select crypto tokens and see their value relative to a USD amount and another crypto token.

Scenario

Imagine you are building a feature for a crypto platform where users can explore potential token swaps. They need a simple tool to select two tokens (a source and a target) and input a USD amount to see the approximate equivalent amounts in those tokens.

Details

Core Task:

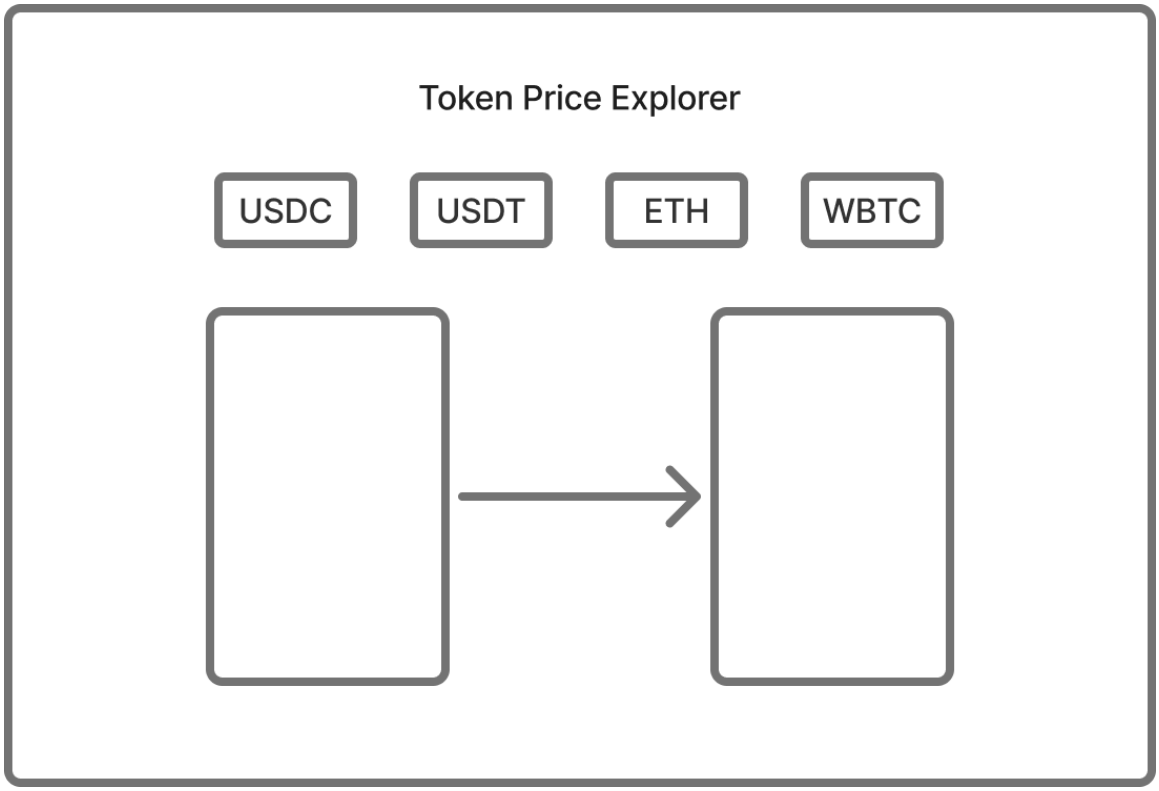
Build a single-page web application using React that implements the Token Swap Interface.

Technical Requirements:

1. **Framework:** You **must** use **React**.
2. **Auxiliary Libraries:** You are free to use other libraries or meta-frameworks as you see fit.
3. **API Interaction:** You will need to fetch data from a provided API package.
4. **Environment:** Assume a modern web browser environment.

Design Guidance:

- Please use the provided wireframe as a visual guide for the layout and core components.
- You can be creative on the design for this task and demonstrate UX design skills



Token Info API

We will use the npm package `@funkit/api-base`

```
import {
  getAssetErc20ByChainAndSymbol, // erc20 is a term for crypto token
  getAssetPriceInfo
} from '@funkit/api-base'

// ask AI about what is chainId!
const tokenInfo = await getAssetErc20ByChainAndSymbol({
    chainId: '1',
    symbol: 'USDC',
```

```
    apiKey: 'api-key'
})
// ask AI about what is token addr/chain id
const price = await getAsssetPriceInfo({
  chainId: '1',
  assetTokenAddress: '0xeeeee.....',
  apiKey: 'api-key'
})
```

The code above is only for demonstration, it may not work and you may need extra investigation to incorporate it into the app. You also need investigate the return type/parametr

You can use this dev api key `Z9SZaOwpmE40KX61mUKWm5hrpGh7WHVkaTvQJpQk`

Notable Tokens to support

Your app can include, but not limited to, these tokens. How to maintain/fetch the token list is at your discretion.

Token Name	Chain Id
USDC	1
USDT	137
ETH	8453
WBTC	1

Submission

1. **Deployment:** Deploy your application using Vercel (preferred, free with a GitHub account) or a similar platform (Netlify, GitHub Pages) with a public accessible URL.
2. **Source Code:** Provide a link to your source code repository (e.g., GitHub, GitLab). Ensure the repository is public or grant access.
3. **README:** Include a README.md file in your repository. You can write notable features/instructions for your app, like
    - Instructions on how to set up and run the project locally.

- A link to the deployed application.
- Any assumptions you made or design choices you want to highlight.
- Brief notes on any libraries you chose to use and why.

4. **Deadline:** Please submit your assignment within a week.

Handling Ambiguity

This description intentionally leaves some details open to interpretation (e.g., specific UI interactions for token selection, error handling details, loading states). Use your product and engineering judgment to make sensible decisions. Please document any significant assumptions you make in your project's README file.

AI Usage Policy

You are welcome to use AI tools extensively during this assignment. However, the final submission should represent your own understanding, best practices, and coding style. Be prepared to explain your code, design choices, and how you used AI assistance. The quality, correctness, and maintainability of the final code are your responsibility.