Frontend Developer Coding Assessment

The goal of this assessment is for us to take a look at how you write and organize your code. This is the most technical part of the Valence interview process, and we want to make sure that you are not under any stress and have plenty of time to complete this assessment.

Guidelines

- Take as much time as you need
- Use Google as you normally would
- Use of third party frameworks is encouraged
- Clearly define dependencies
- No need to write any unit tests, but you're welcome to add a few

To remove any potential bias, we will scrub any personally identifiable information from your submission before handing it over to an engineer for a code review.

Submission

When you are finished, please perform the following steps:

1. Remove any personally identifiable information from the project and file headers (such as the name of the author, provisioning information, etc).
2. Zip up the code, excluding any build folders or compiled artifacts.
3. Email your submission to your Valence recruiter.

Evaluation Criteria

We are looking at the following things:

- Does the app compile and run? Are all the base features implemented?
- How well is your code organized? (Hint: Break your code into meaningful functions and compose them in

the endpoint hook)
- Are your variable and class names descriptive? Do they make sense? (Tip: don't name things with a single character)
- Are you using proper encapsulation?
- Are you adhering to common code style guidelines, such as the Google JavaScript Style Guide?
- Did you implement any bonus features outlined below?
- Less code = more. Brevity is the soul of wit.

Project

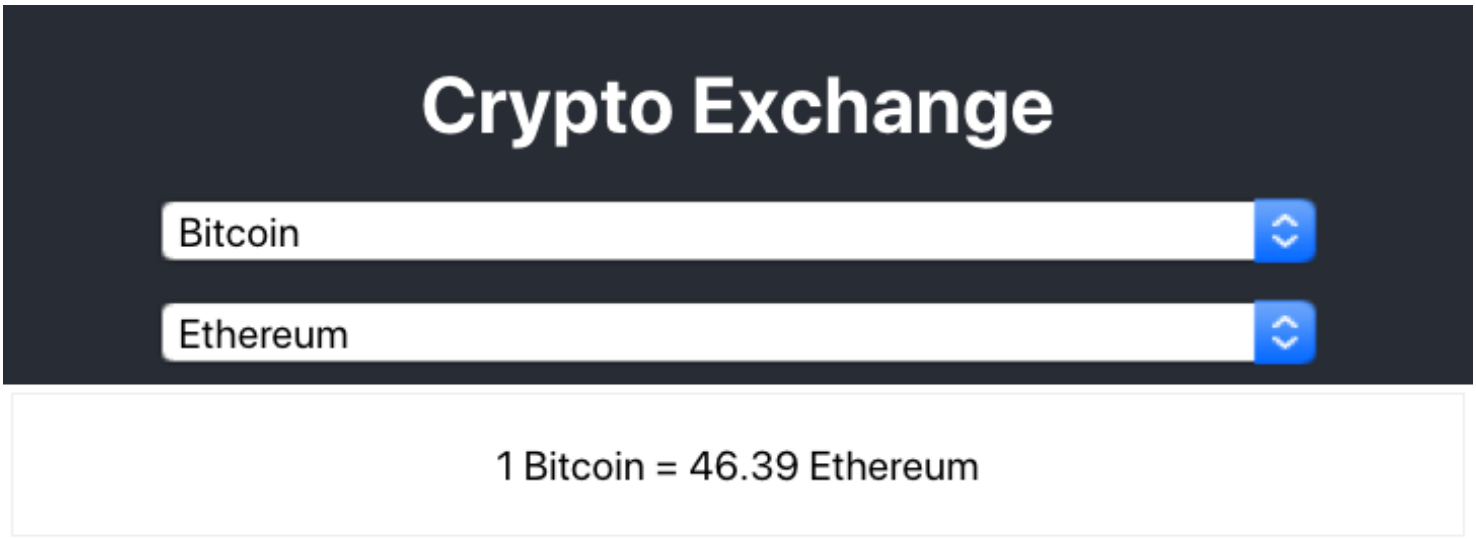
Using HTML, CSS, JavaScript, and a JS Framework of your choice, create a web app that uses the **CoinPaprika** API to compare the exhange rates between different Cryptocurrencies. This project is currently set up for React, but you are welcome to use another framework, see Acceptable JavaScript Frameworks below.

What to Do

Your mission, should you choose to accept it, is to develop a single page app that uses the CoinPaprika API to compare different cryptocurrencies. The minimum requirement is to display the exchange rate between the selected coins. You are welcome to compare more than 2 coins, or use other API endpoints to display more information.

1. **CoinPaprika** hosts over 2,000 coins - be sure to limit the coins to something manageable (e.g. `0 < rank < 10` via the `/coins` endpoint or `price > 10usd` via the `/tickers` endpoint)
2. It should be intuitive to select which coins to compare
3. It should be obvious which coin the information belongs to
4. Must look good on desktop and mobile
5. A coin shouldn't be able to be compared to itself
6. Be proud of the code you submit

Simple Example



Resources

[CoinPaprika API]
[Be mindful of the API rate-limit]

Notes

Your app must start with

1. `npm install`
2. `npm run start`
3. and navigating to `localhost:3000` Or something similar to these 3 steps

Bonus

Implement an additional feature that you find useful

Acceptable JavaScript Frameworks

- plain, vanilla JS
- angular
- react
- vue
- anything not on this list
- something new to you