



QUANTOX

SOFTWARE DEVELOPMENT & IT CONSULTING

Pre-Employment Test

CryptoTrack App

The goal is to make a SPA that will allow the user to track the values of cryptocurrencies and the value of the cryptocurrencies the user has.

Application details:

When entering the site, show a Loader until the API response is resolved. When the API response is resolved, a table of the 50 most valuable cryptocurrencies should be shown. One table row for each currency.

Every table row needs to contain the following:

1. Name of the cryptocurrency
2. Currency short name
3. Current \$ value of the currency
4. % of change in the last 24h – if positive color the text green, if negative color the text red

0.24 %

0.24 %

5. Input field where user should enter amount of the cryptocurrency he owns.
6. Submit button that's disabled if the user has not typed anything into the input field (4), when the user inputs something the button should become active, when the user clicks the button the next part (7) should be calculated and stored into localStorage. The user should be able to activate the submit button by pressing the enter key.
7. \$ value of the cryptocurrency amount the user has

| Name | Short name | \$ Value | Last 24h | Amount you own | \$ value of your coin |
|---------|------------|------------|----------|-------------------------------------|-----------------------|
| Bitcoin | BTC | \$ 7631.65 | 1.65 % | <input type="text" value="Submit"/> | \$ 0.00 |

Save data about how much of which currency the user has into localStorage. If there is values saved for that currency, fill the input field and the last column with the data from the local storage so that when the user comes next time, he will have all his data again.

Example of row when filled with data:

| | | | | | |
|----------|-----|------------|--------|---|------------|
| Ethereum | ETH | \$ 610.023 | 2.66 % | <input type="text" value="4"/> <input type="button" value="Submit"/> | \$ 2440.09 |
|----------|-----|------------|--------|---|------------|

- The data from the API should be refreshed every 60 seconds
- The background-color of each even row of the table should be grey, the odd ones should be white.
- Details page - Clicking on a name of some cryptocurrency should open a new page (not a modal) and on that page, all of the data that is available through the API about that currency should be shown in a format you prefer.
- Before the response from the API is resolved, show the same loader you used for the home page. If a user refreshes the page while on a details page of some currency, he should get the same page again, he shouldn't be returned to the home page.

Bonus points for:

- Unit testing
- Testing user input with regex
- Implementing pagination so the user has only 10 currencies and 5 pages
- Show how much \$ user has gained / lost since his last visit.

Technical requirements:

- We prefer native JS, but you can use frameworks if you want to
- Use a CSS preprocessor

- Use ES6
- Use promises
- If you are using React.js DON'T use Redux
- Commit often with descriptive commit messages
- Use ESLint
- Use Babel
- If you are using a framework, you can use boilerplates like create-react-app etc.

Link to the API: <https://coinmarketcap.com/api/>

A faint, semi-transparent watermark image of a person's hands resting on a computer keyboard is visible in the background.

Thank You For Your Application!

www.quantox.com