Market View (Front End)

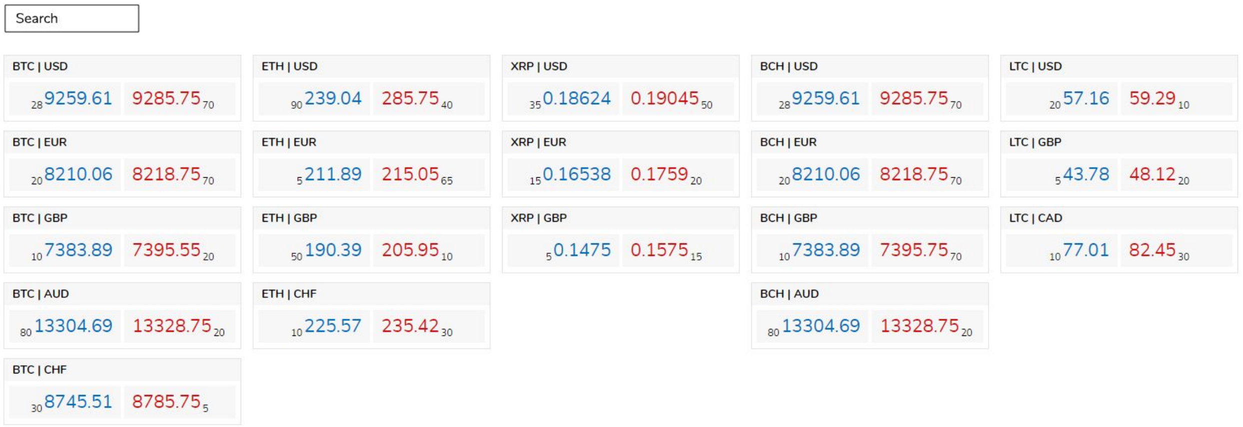
As we mainly work with trading markets, the technical task is to create a Whiteboard of trading instruments called Market View.

Technical requirements:

1. You can use Angular, React or Vue.js.
2. You can use **npm/yarn** package manager to manage dependencies.
3. Optionally, you can use any **CSS** framework (**Tailwind CSS**, **Material UI**, **Bootstrap**, etc.)
4. Having **unit tests** using Jasmine, Jest or any other testing framework, as an advantage.
5. Application can support **Google Chrome** browser.
6. Deliver the code either in **Github** or as a **zip** file.
7. Make sure to document your project in a simple **README.md** file so it would be easier for us to set up, run and test your application.

Business requirements:

1. Market view wireframe:



Search

BTC | USD
28 9259.61 9285.75 70

BTC | EUR
20 8210.06 8218.75 70

BTC | GBP
10 7383.89 7395.55 20

BTC | AUD
80 13304.69 13328.75 20

BTC | CHF
30 8745.51 8785.75 5

ETH | USD
90 239.04 285.75 40

ETH | EUR
5 211.89 215.05 65

ETH | GBP
50 190.39 205.95 10

ETH | CHF
10 225.57 235.42 30

XRP | USD
35 0.18624 0.19045 50

XRP | EUR
15 0.16538 0.1759 20

XRP | GBP
5 0.1475 0.1575 15

BCH | USD
28 9259.61 9285.75 70

BCH | EUR
20 8210.06 8218.75 70

BCH | GBP
10 7383.89 7395.75 70

BCH | AUD
80 13304.69 13328.75 20

LTC | USD
20 57.16 59.29 10

LTC | GBP
5 43.78 48.12 20

LTC | CAD
10 77.01 82.45 30

2. The client application should be a **single page application** with 2 components:
   a. The market view which should show a list of instrument tiles grouped by left currency. Each instrument tile should contain:
      i. Currency pair
      ii. Bid side info (price and amount) on the left side of instrument
      iii. Offer side info (price and amount) on the right side of instrument
   b. Search input field on the top of the screen. It should filter instruments by currency pair (full or partial match).
3. There should be real-time updates logic making prices/amounts to change on the fly. The price/amount of any instrument (in the Market View) should be updated once a second with a different value.

4. Instruments will be provided in an **instruments.json** file and should be fetched using HTTP request.

Guidelines and Tips

- Implement best practices on how you see fit.
- Using Typescript is an advantage.
- Design the page in such a way that it is user-friendly. Layout-wise, device-wise, etc.
- The minimum business requirements are defined in step 1 above. But feel free to add components to make the application easier to use (charts, help links, tooltips, etc.).
- Be resourceful. The internet is at your disposal.
- The more you put in the more we can appreciate your time and effort.
- Bring out your inner Da Vinci! Be creative, innovative and original. Make this your masterpiece.