## **Requirements:**

* Creat-react-app - as starter template
* High-level mockups for tech task  
  <https://drive.google.com/file/d/1c23I1p57iTinRdsLrWleEHYSxb43Cq95/view?usp=sharing>
* Result of the test task - link to <https://codesandbox.io/> (100% working application, we can click on everything, responsive layouts, no errors in the console or on the screen, where we can check code structure, structure of folders, as example - <https://codesandbox.io/s/y2lrywpk21>)
* Currency API <https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5>
* On each 5th api request - imitate server error (create counter, store it in localStorage). After - reset value in localStortage. On error case - hide table, show error message.
* Create currency converter: two input fields and appropriate dropdown with currencies(USD, EUR, UAH, BTC), calculate button, currency for calculations - get actual (can be changed in the table) value from cell with currency  
  Example - in the mockups (<https://drive.google.com/file/d/1c23I1p57iTinRdsLrWleEHYSxb43Cq95/view?usp=sharing> two dropdowns with currencies). Values in the input can be swapped (please see arrow, functionality the same as in google translate for example).
* Height of layout application - full screen height, without vertical scroll header, footer - 100px height, all another vertical space - content.

## **Technology stack:**

* React - required
* State management - local state, context, Redux/Effector
* Ui framework - feel free to choose any
* Typescript is a big advantage

## **Nice to have:**

* Fancy UI (any framework - bootstrap, material ui, etc )
* Error handling (imitate and catch server errors, handle min/max values in the inputs)
* Good-structured code, react best practices are welcome
* React hooks
* Responsive layout
* Add fancy styles for head, footer, table and whole layout
* List of available packages - react, redux, effector, typescript, any ui/css framework for styling, any test framework.