

Test Specification

****Language****: TypeScript (*.ts, *.tsx), SCSS (*.scss)

****Rules****:

- Start using `create-react-app`
- Adding additional libraries is forbidden
- `function` keyword is forbidden
- `redux` is forbidden
- Everything has to be typed as a result
 - Known types have to be omitted (`const str = 'test';` > `const str:

string = 'test';`)

- `any` and `unknown` type is forbidden
- `object` type is forbidden - objects has to be described by eg.: `{
someProp: string; }`

- Code have to be splitted into components
 - You have to use at least once function component and class component

****correctly****

- Styles have to be written in `*.scss` files and imported in `*.ts(x)` files

source

- Styles have to be written by programmer - not imported from external

- Styles cannot be written inline (in JS/TS files)

CSS variables and custom `data-*` attributes

****Result application****:

- Have to have two forms
 - First form allows add new color to database
 - Form have to be submitted by `onSubmit` event
 - Colors will be saved in LocalStorage in some way
 - Color have to be checked, parsed before adding
 - Color have to be in HEX RGB form
 - Writting non valid characters have to be blocked
 - Writting too many characters have to be blocked
 - Writting `#` (except first character) have to be blocked
 - Second form allows filter colors displayed below
 - After every change list of colors are changed
 - Displayed colors have to meet ALL selected by user conditions,

possible conditions:

- `Red > 50%` means Red in RGB is higher that 127
- `Green > 50%` means Green in RGB is higher that 127
- `Blue > 50%` means Blue in RGB is higher that 127
- `Saturation > 50%` means Saturation in HSL is higher than 50%
- List of colors have to contains all colors:
 - Listed in predefined, default list:
 - You cannot remove them, so "x" should be hidden
 - Listed in LocalStorage, added by user:
 - They should be removable
 - Colors have to be named using HEX RGB form, with uppercase
 - Colors have to be represented by colored rectangle above name
 - Colors have to be ordered in such way:
 - Higher Red value first; if the same then:
 - Higher Green value first; if the same then:
 - Higher Blue value first; if the same then on the same position