React Task 1 - Components, state and props

The main goal of this task is to create app skeleton - set of some simple components for the app.

Please, follow next steps:

Steps:

Installation CRA (Create React App) and generation new project:

1. Install CRA globally

```
npm install create-react-app -g
```

2. Generate React project

```
create-react-app courses-app
```

3. Go to your project directory

```
cd courses-app
```

- Configure eslint and prettierSee instruction in the ESLINT_instruction
- Configure pre-commit hook.See instruction in the PRE-COMMIT_instruction
- 6. Run your project

```
npm run start
```

Project structure requirements

- 1. You should use **functions** components for your app.
- 2. Rename App.js to App.jsx.
- 3. App.jsx main app file.
- 4. Delete unnecessary files:
- src/App.test.js

- src/logo.svg
- 5. Delete default code in files:
- scr/App.jsx
- src/App.css

You can do like this:

```
function App() {
    return <div>React</div>;
}
export default App;
```

Result in the browder:

React

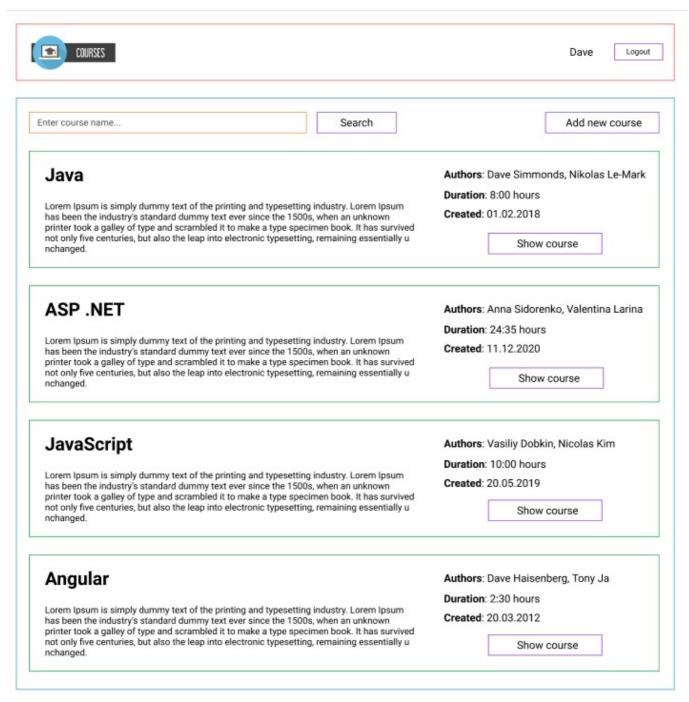
6. Create folders and jsx files for each component. Follow the architecture below:

```
__ SearchBar.jsx
            CourseCard
             |__ CourseCard.jsx
      _ // any components you want to add
-- common
    |--Button
   | |_ Button.jsx
   |__Input
   | |_ Input.jsx
   __ //any common components you want to add
-- helpers
     |--pipeDuration.js // a helper to format course duration
    |-- dateGeneratop.js // a helper to get current date in the correct format
    |__ // any helpers you want to add
|-- constants.js // file with mocked data
|_ ...
```

- The presented architecture is required for use, BUT you can add files and folders at your discretion.
 - 7. You can use any way to add styles to components (styled components, Bootstrap, ets).
- Please, keep each component's style file in their appropriate folders
 - 8. You can choose the design at your discretion.

It is important to keep the layout

(the arrangement of elements on the page)



App initial view

Criteria (20 points max)

Common

- [1 point] The architecture of the application **should be** the same as presented above.
- [1 point] Components are presented as function components (not class components).
- [1 point] Use .jsx extensions for files with jsx syntax.
- [1 point] All inputs and buttons should be reusable components.
- [2 points] eslint, prettier and pre-commit hook work correctly.

Searching

- [1 point] Implement searching functionality by title. Reset search result when searchbar is empty.
- [1 point] Implement searching functionality by id. Reset search result when searchbar is empty.

Courses

• [2 points] - Show list of courses.

CreatCourse Component

(**Add new course** in COMPONENTS)

- [1 point] Possibility to add a title.
- [1 point] Possibility to add a description.
- [1 point] Possibility to add a duration.
- [2 points] Show duration time in a format «hh:mm». Example: **122 min** should be showed as **02:02 hours**.
- [1 point] Add logic for creating new author.
- [1 point] Add logic for adding author to **Course authors**
- [1 point] Add logic for deleting author from **Course authors**
- [2 points] Add logic for saving course (new course should be presented in the courses list).

Please find a detailed description of components and functionality in the file COMPONENTS