SportSee

Build an analytics dashboard with React

PRESENTATION

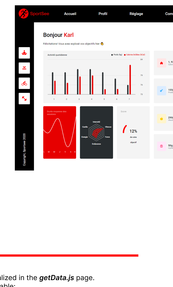
SportSee is a startup dedicated to sports coaching. In full growth, the company will today launch a new version of the user profile page. The goal is to redo the profile page with React.

PROJECT

The project is developed as an application, under the React environment, using the ***create-react-app*** command.
React, **React DOM** as well as **Axios** (for data recovery) will be used.
SASS will be used for CSS

only the **Desktop** version was made, according to the following structure:

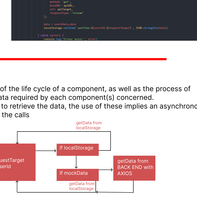
- **app** : functions
- **assets** : medias
- **components**
- **pages**
- **styles** : scss + css



API REQUESTS

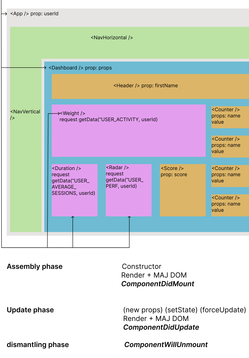
API requests are centralized in the ***getData.js*** page.
The module is configurable:

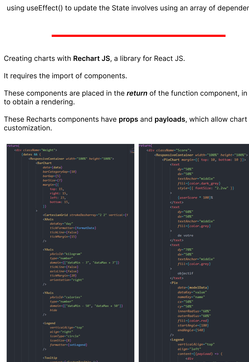
- **.env** - this file contains the launch port of the port, the URL containing the BACK-END, the access to the mocked data as well as the request routes for the BACK-END API
- **localStorage** - the recovered data is then stored locally, in order to limit network requests
- **@return** - an object of the form *(data: données)*



life cycle and asynchronicity

Representation of the life cycle of a component, as well as the process of retrieving the data required by each component(s) concerned.
Axios allows us to retrieve the data, the use of these implies an asynchronous management of the calls





Assembly phase

Constructor
Render + MAJ DOM
ComponentDidMount

Update phase

(new props) (setState) (forceUpdate)
Render + MAJ DOM
ComponentDidUpdate

dismantling phase

ComponentWillUnmount

using useEffect() to update the State involves using an array of dependencies

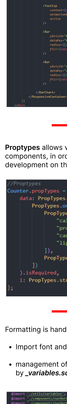
graphics

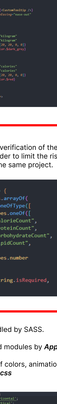
Creating charts with **Rechart JS**, a library for React JS.

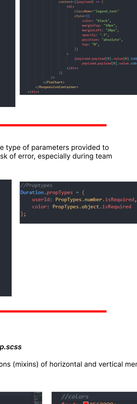
It requires the import of components.

These components are placed in the ***return*** of the function component, in order to obtain a rendering.

These Recharts components have **props** and **payloads**, which allow chart customization.







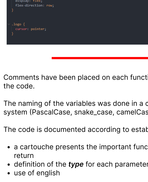
Proptypes

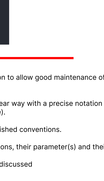
Proptypes allows verification of the type of parameters provided to components, in order to limit the risk of error, especially during team development on the same project.

SASS

Formatting is handled by SASS.

- Import font and modules by ***App.scss***
- management of colors, animations (mixins) of horizontal and vertical menus by ***_variables.scss***





Documentation and Readme

Comments have been placed on each function to allow good maintenance of the code.

The naming of the variables was done in a clear way with a precise notation system (PascalCase, snake_case, camelCase).

The code is documented according to established conventions.

- a cartouche presents the important functions, their parameter(s) and their return
- definition of the ***type*** for each parameter discussed
- use of english