

Projet



Développez un tableau de bord
d'analytics avec React

Contexte

SportSee
Pages utilisateurs
Développeur Front-end
Sessions sportives
Graphiques
Nouvelle version

Coaching sportif

Adaptation projet

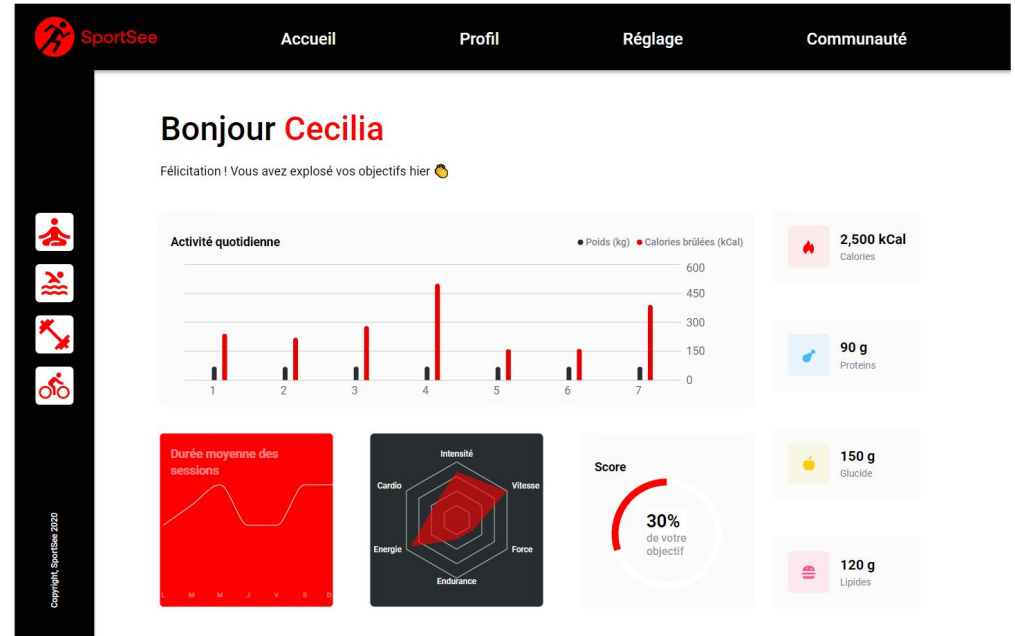
Pour s'imprégner du projet, documents et outils à disposition :

- Slack, mail d'informations sur la mission et conseils techniques
- Maquette Figma :
<https://www.figma.com/file/BMomGVZqLZb81lMDShpLu/UI-design-Sportify-FR?node-id=1%3A2>
- Kanban :
<https://www.notion.so/Tableau-de-bord-SportSee-6686aa4b5f44417881a4884c9af5669e>
- Backend du projet :
<https://github.com/OpenClassrooms-Student-Center/P9-front-end-dashboard>

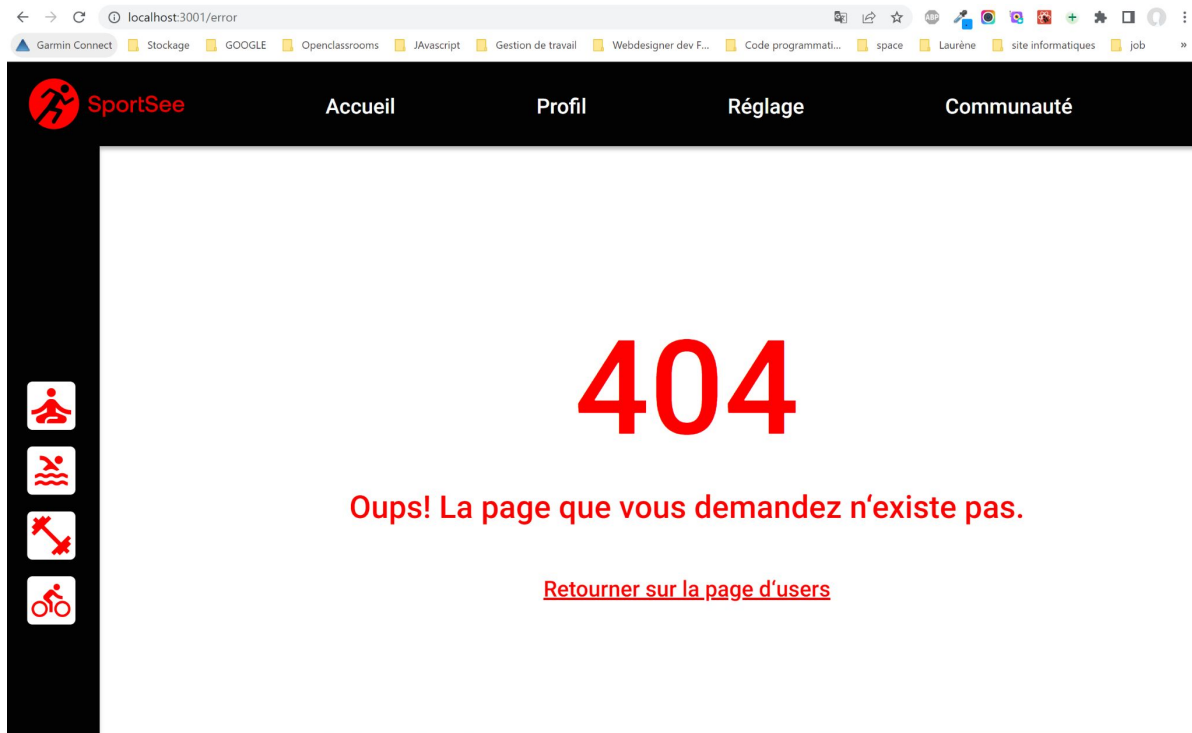
Projet codé

Projet codé

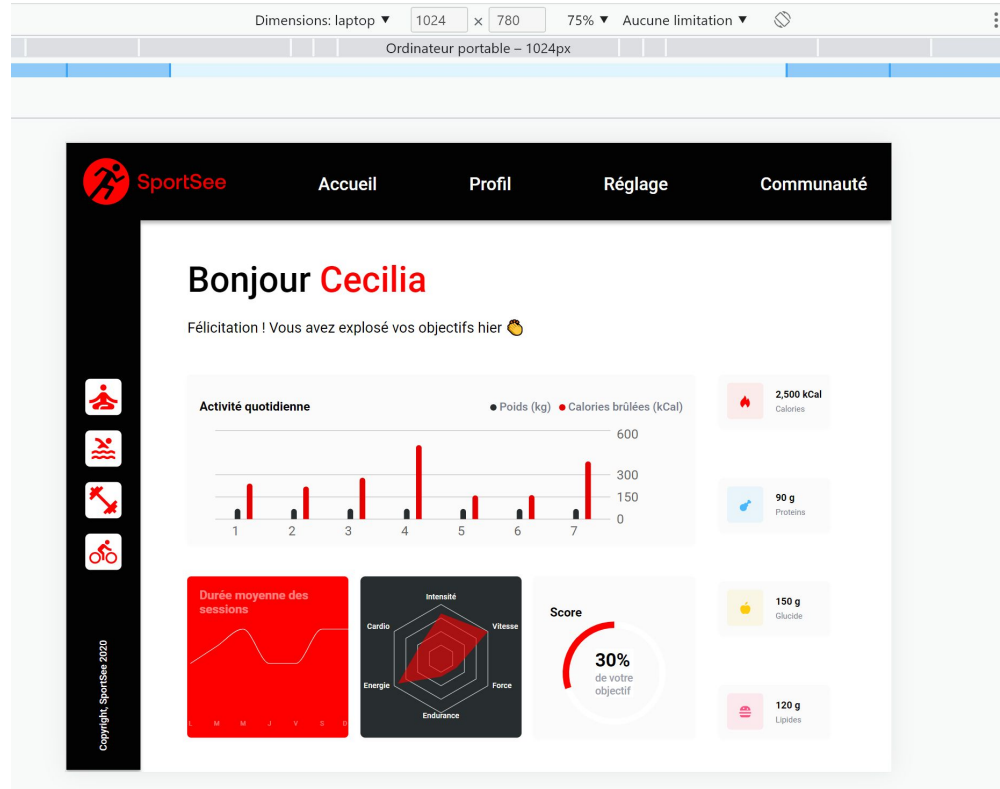
- Projet codé et conforme à la maquette Figma
- Comprend :
 - informations utilisateurs
 - graphiques
 - informations caloriques
 - erreur 404



Erreur 404



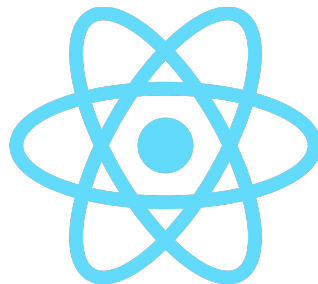
Responsive



Rappel : React

React.js

- Bibliothèque JS
- React génère un DOM virtuel avec les instructions du JSX, distinct du DOM du navigateur
- Architecture de code
- Création des UI à partir de composants :
 - modulaires
 - flexibles
 - réutilisables

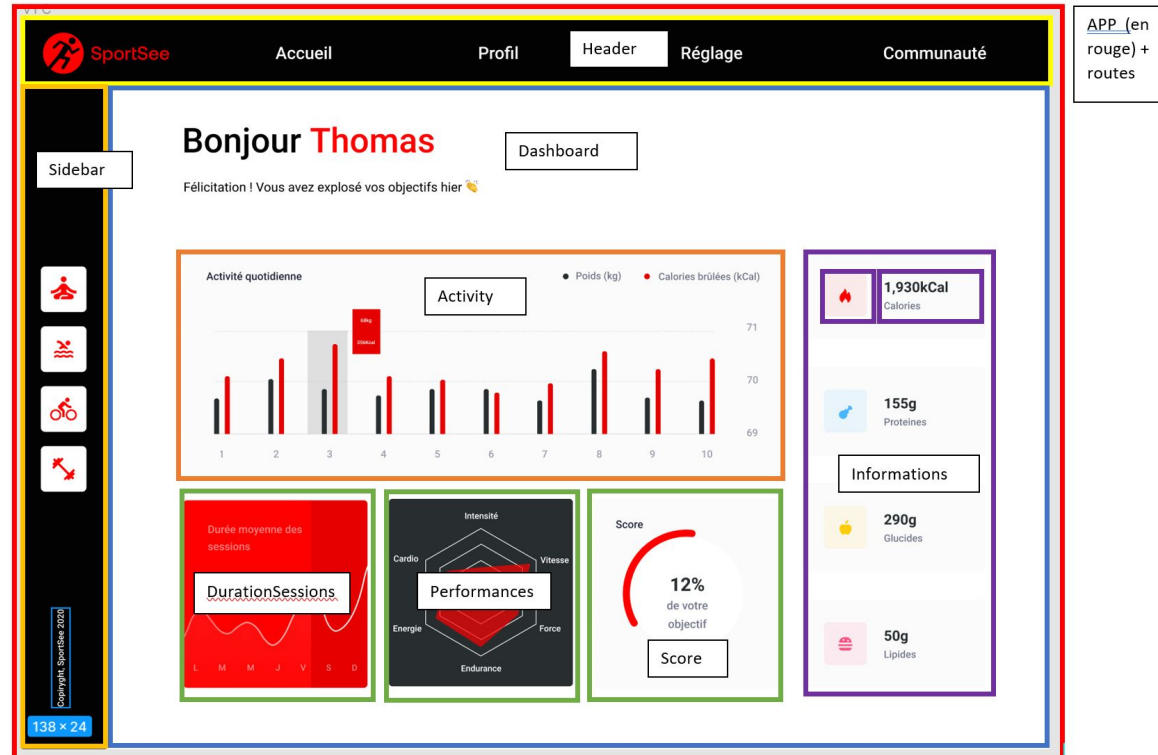


Projet approfondi

Composition

Composition

DOC WORD



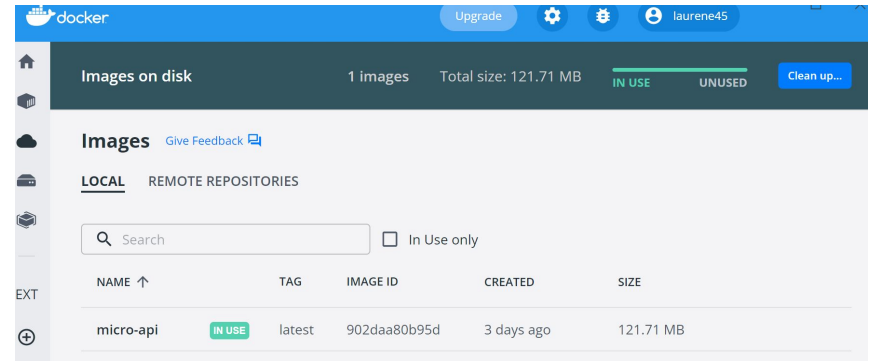
APP (en rouge) + routes

Docker

Docker



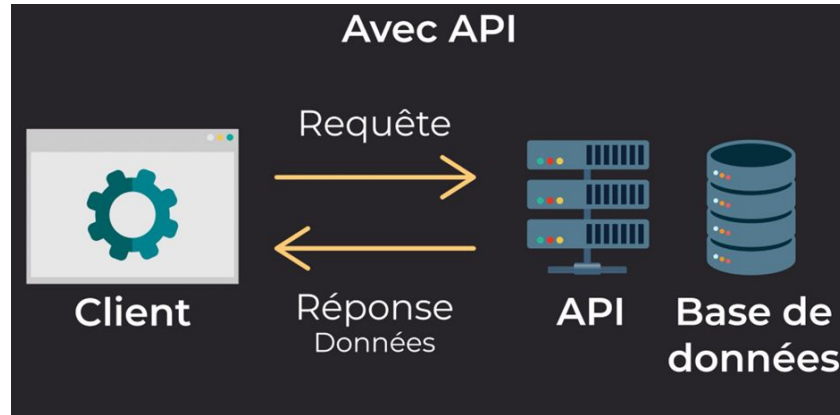
- Logiciel libre Open Source qui permet d'automatiser le déploiement d'applications.
- Plateforme de virtualisation par container qui va permettre de concevoir, tester et déployer des applis dans n'importe quel environnement en s'assurant que le code s'exécutera automatiquement.
- Installation via : <https://github.com/OpenClassrooms-Student-Center/P9-front-end-dashboard>
- Avantages :
 - Simple à installer
 - Améliore la productivité des dev.
 - Migration facile des applications
 - Meilleure gestion des applications



API

API : Application programming interface

- Intermédiaire entre 2 logiciels qui permet de communiquer entre eux,
- l'API transmet des messages à travers un système de requêtes et de réponses.
- Le client va demander à l'API une information, celle-ci va aller chercher cette information dans la BDD puis la renvoyer au client dans un second temps.



API REST

- API qui suit un ensemble de règles architecturales qui permet aux logiciels de communiquer entre eux sur un réseau/même appareil.
 - API REST peut créer des services web.
 - Services web RESTfull : utilise des méthodes HTTP pour récupérer et publier des données dans un format JSON, entre client et un serveur.
-
- Fonctionnement sur la base d'une relation client/serveur : une interface uniforme basée sur des ressources et des collections.
 - Sans état : le serveur ne maintient pas de connexions ou de sessions entre les appels.

Call API : Axios

- Bibliothèque qui permet de gérer les appels asynchrones
- Faire des requêtes HTTP vers des ressources externes
- Utiliser les promesses (objet 'promise') pour nous retourner un résultat.
- Transforme automatiquement des données JSON.
- Axios : <https://github.com/axios/axios>
- Base URL : localhost:3000
- Chaque fonction à l'ajout d'un

endpoint pour que l'URL soit complète.

```
import axios from 'axios';

axios.defaults.baseURL = 'http://localhost:3000';

const getData = (url) => axios.get(url).then((response) => response.data.data);

// ID user
export const userId = 18;

export const API = {
  /** @function get user info from api
   *
   * @param {number} userId
   * @returns {Object} (user info)
   */
  getUser: (userId) => {
    const url = `/user/${userId}`;
    return getData(url);
  },
}
```

Context

Hook: useContext()

- useContext() est un hook qui permet de se “brancher” depuis un composant enfant qui a été wrappé par un Provider, donc d’accéder simplement au state partagé.
- Etape :
 - création du context
 - utilisation de state
 - requête asynchrone retourne une promesse
 - Provider dans le return
 - rappeler le provider dans app
 - définir le context dans le composant

```
// Create Context
export const UserContext = createContext({ userInfo: null });

export const UserProvider = ({ children }) => {
  const [userInfo, setUserInfo] = useState(null);
  const [loading, setIsLoading] = useState(true);
  const [userActivity, setUserActivity] = useState(null);
  const [userDurationSession, setUserDurationSession] = useState(null);
  const [userPerformance, setUserPerformance] = useState(null);
  const [userScore, setUserScore] = useState(null);
  const [error, setError] = useState(null);

  // @ fonctions au chargement des données users + graphiques
  const loadUserInfo = () => {
    return API.getUser(userId);
  };

  const loadActivity = () => {
    return API.getUserActivity(userId);
  };

  const loadDurationSession = () => {
    return API.getUserAverageSessions(userId);
  };

  const loadPerformance = () => {
    return API.getUserPerformance(userId);
  };
};
```

Graphiques

Graphiques

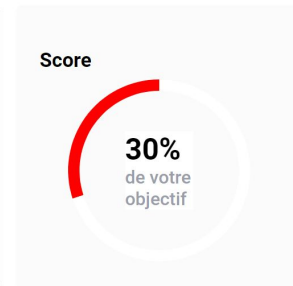
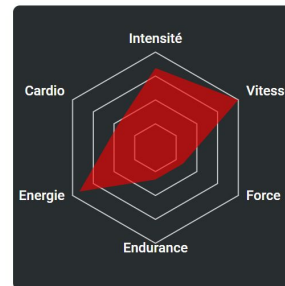
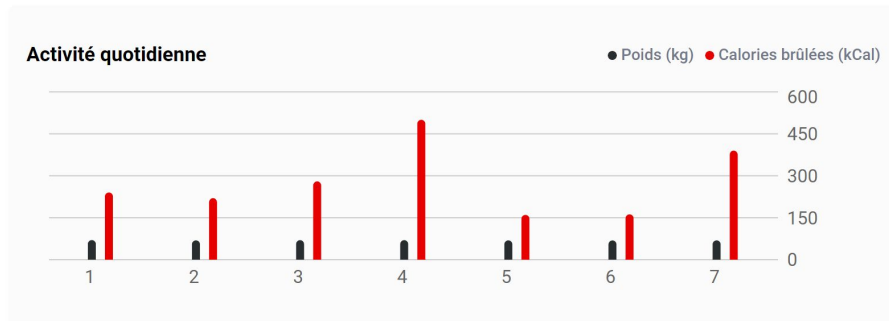
- Utilisation de la bibliothèques :

<Recharts />

- Recharts : <https://recharts.org/en-US/guide>
 - Doc qui facilite la compréhension de Recharts
 - Présentation des graphiques possibles
 - Présentation du code construit pour la réalisation des graphiques

Graphiques

- 4 graphiques présentés :
 - Activités : BarChart
 - Durée de session : LineChart
 - Performances : RadarChart
 - Score : PieChart
- Construction des graphiques :
 - Informations générales du graphiques
 - Les axes X et Y
 - Les tooltips
 - Les légendes
 - Css dans le graphique



JSDoc

JSDoc

- JSDoc est un langage de balisage utilisé pour documenter les codes sources Javascript
- <https://jsdoc.app/about-getting-started.html>
- Projet : fonctions et méthodes documentées

```
export const API = {  
  
  /** @function get user info from api  
   *  
   * @param {number} userId  
   * @returns {Object} (user info)  
   */  
  
  getUser: (userId) => {  
    const url = `/user/${userId}`;  
    return getData(url);  
  },  
}
```

PropTypes

PropTypes

- Vérification des données reçues afin qu'elles soient valides
- Précisions sur le type de props utilisées
- Correction des erreurs dans la console
- Installation PropTypes: `npm install --save prop-types`

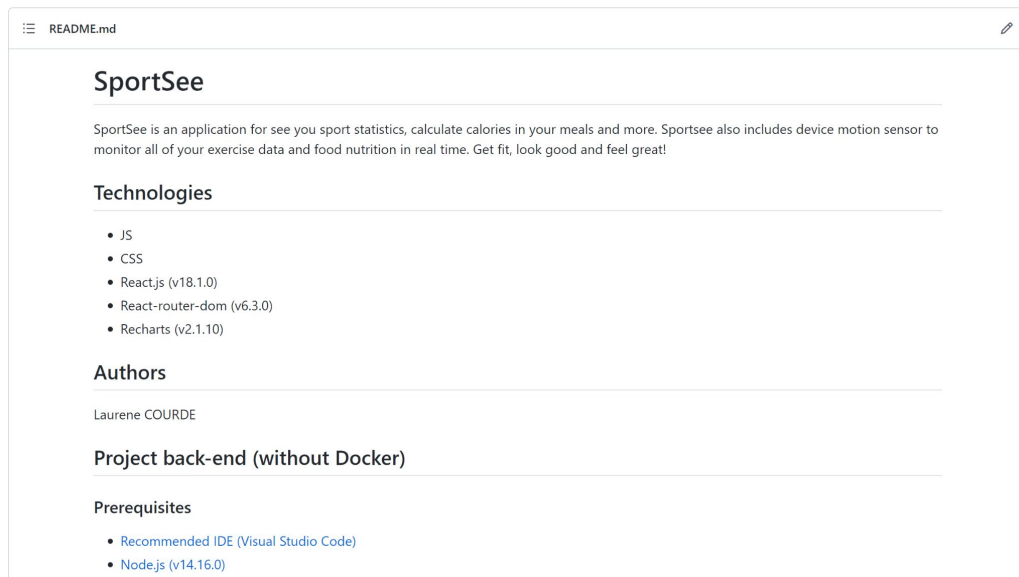
```
2 import PropTypes from 'prop-types';
```

```
Activity.propTypes = {  
  |   userId: PropTypes.number,  
};  
  
CustomTooltip.propTypes = {  
  |   payload: PropTypes.array,  
  |   active: PropTypes.bool,  
};
```

Readme.md

Readme.md

- 1er fichier qu'un développeur doit consulter avant d'entamer un projet.
- md : Extension en Markdown
- Comprend les informations :
 - Description du projet
 - Technologies
 - Install projet Back-end
 - Install projet Front-end



Repository

Github

Lien vers le repo Github :

https://github.com/Laurene45/LaureneCourde_12_19052022-/tree/main/sportsee

The screenshot shows the GitHub interface for the repository 'Laurene45 / LaureneCourde_12_19052022-'. The repository is public and has 1 Unwatch, 0 Fork, and 0 Star. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The current view is the 'main' branch, specifically the 'LaureneCourde_12_19052022- / sportsee / src /' directory. A recent commit by Laurene45 is shown, titled 'modification de la données data en données dans tooltip', with a commit hash of 2015b68 and a timestamp of 2 minutes ago. Below the commit, a table lists the files and folders in the directory, including assets, components, css, data, pages, sass, utils, Api.jsx, App.jsx, index.css, index.jsx, and logo.svg, along with their respective commit messages and timestamps.

File/Folder	Commit Message	Timestamp
..		
assets	ajout fichiers Src + 1er graphique activity	20 days ago
components	modification de la données data en données dans tooltip	2 minutes ago
css	ajout fichiers Src + 1er graphique activity	20 days ago
data	ajout des données mockées Data	2 days ago
pages	suppression des commentaires inutiles et corrections des syntaxes de...	2 days ago
sass	modification paddin main-container responsive laptop	2 days ago
utils	précision commentaire promesse asynchrone	21 hours ago
Api.jsx	suppression des commentaires inutiles et corrections des syntaxes de...	2 days ago
App.jsx	mise à jour des fichiers apiContext et graphique dashboard sur le con...	8 days ago
index.css	ajout react app sportsee + header / lateralbar + config react	last month
index.jsx	ajout fichiers Src + 1er graphique activity	20 days ago
logo.svg	ajout react app sportsee + header / lateralbar + config react	last month

Avez-vous des questions ?

