



WEALTH HEALTH

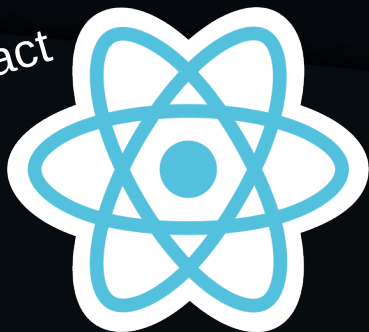
Faire passer une librairie jQuery vers React

P14 formation développeur front-
end OpenClassrooms

Bénédicte HÉRAULT

Faire passer une librairie jQuery vers React

- Nouvelle version du site :
- Créer nos propres composants React à la place des plugins jQuery tiers
 - Convertir un composant jQuery en composant React
 - Utilisation possible de composants npm pour les autres pluggins
 - Convertir tout le site en React



jQuery

- Publier le composant converti sur npm
- Réduire la dette technologique
- Possibilité de moderniser le design du site

Contraintes :

React :



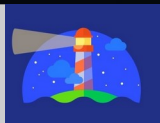
- Ensemble du projet
- Un plugin converti (pour débiter)
- Bibliothèques utilisables



npm :

- Publier plugin converti
- Utilisation dans le projet

Lighthouse :



- Mesurer les performances des 2 versions
- Rapport Lighthouse en pdf

Design :

- cohérence



Doc fournie:

Repository ancienne version



Issues

4 Open	0 Closed
[HRnet] Dropdown menus are slow and inconsistent	bug
#4 opened on 12 Jul 2020 by mja2128	
[HRnet] The modal dialog is very difficult to style	bug
#3 opened on 12 Jul 2020 by mja2128	
[HRnet] Current Employees table is very slow to load new data	bug
#2 opened on 12 Jul 2020 by mja2128	
[HRnet] Date picker is slow, sometimes unresponsive	bug
#1 opened on 12 Jul 2020 by mja2128	

Liens repositories plugins jQuery



Logo de WealthHealth



WEALTH HEALTH

Choix techniques :

React :

- npx create-react app
- Redux pour la gestion du state
- Redux toolkit

npm :

- publication du composant
- Plutôt que package Github
- Recherche de composants

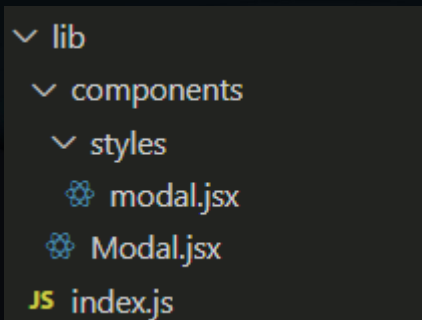
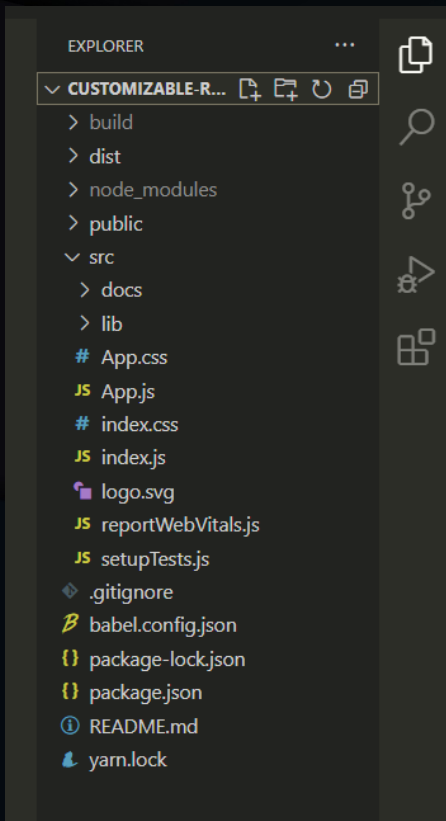
Autres outils :

- Lighthouse
- Styled components
- Netlify

Design :

- Canva
- <https://coolors.co/>

Réalisation plugin 1/3 :



```
src > lib > components > Modal.jsx > ...
1  import React from "react";
2  import { ThemeProvider } from "styled-components";
3
4  import { ModalContainer, ModalMessage, ModalButton } from "../styles/modal";
5
6  export default function Modal(props) {
7    return (
8      <ThemeProvider theme={props.theme}>
9        <ModalContainer>
10          <ModalMessage>
11            <p>{props.text}</p>
12            <ModalButton onClick={props.close}>X</ModalButton>
13          </ModalMessage>
14        </ModalContainer>
15      </ThemeProvider>
16    );
17  }
```



```
JS Appjs M X
src > JS Appjs > ...
1 import { useState } from "react";
2 import "../App.css";
3 import Modal from "../lib/components/Modal";
4
5 function App() {
6   // Set here your own colors or keep my theme if you like it
7   const myTheme = {
8     containerBg: "rgba(146, 146, 96, 1)",
9     messageBg: "#444444",
10    messageHoverBg: "#aebf8e",
11    messageHoverText: "black",
12    borderColor: "#d5db99",
13    buttonBg: "#d5db99",
14    buttonHoverBg: "#444444",
15    buttonHoverText: "white",
16  };
17
18  const [openModal, setOpenModal] = useState(false);
19  const onOpenModal = () => setOpenModal(true);
20  const onCloseModal = () => setOpenModal(false);
21  const handleSubmit = (e) => {
22    e.preventDefault();
23    onOpenModal();
24  };
25
26  return (
27    <div className="App">
28      <form onSubmit={handleSubmit}>
29        <button type="submit" className="submit">
30          Clic to open Modal
31        </button>
32        <div className="back">
33          <a href="https://www.npmjs.com/package/customizable-react-modal-by-lazez">
34            Back to NPM
35          </a>
36        </div>
37      </form>
38
39      {openModal && (
40        <Modal
41          theme={myTheme}
42          close={onCloseModal}
43          text="Write here whatever you want!"
44        </>
45      )}
46    </div>
47  );
48 }
49
50 export default App;
```

Réalisation plugin 2/3 :

```
src > lib > components > styles > modal.jsx > ...
1 import styled from "styled-components";
2
3 export const ModalContainer = styled.div`
4   width: 100%;
5   height: 100vh;
6   margin: 0;
7   position: -webkit-sticky;
8   position: sticky;
9   bottom: 0;
10  left: 0;
11  background-color: ${(props) => props.theme.containerBg};
12  padding-bottom: 20rem;
13 `;
```



HRnet
early manage human resources

Réalisation plugin 3/3 :

npm

Search packages

Search

customizable-react-modal-by-lazez

0.3.6 • Public • Published 7 days ago

Readme

Explore

0 Dependencies

0 Dependents

10 Versions

Settings

A modal React component, customize it with your favorite colors!

About

Simple component for React project. Keep the default theme or chose your colors simply.

Installation

You can install this component with npm or yarn:

npm i customizable-react-modal-by-lazez

or

yarn add customizable-react-modal-by-lazez

caution

You'll perhaps have to do npm start or yarn start again so that it works in your project (it'll be opened in another port then)

NPM link

customizable react modal by lazez on NPM

Install

> npm i customizable-react-modal-by-lazez

Repository

github.com/LazezBZH/customizable-rea...

Homepage

github.com/LazezBZH/customizable-rea...

Weekly Downloads

199

Version

0.3.6

License

MIT

Unpacked Size

7.96 kB

Total Files

9

Issues

0

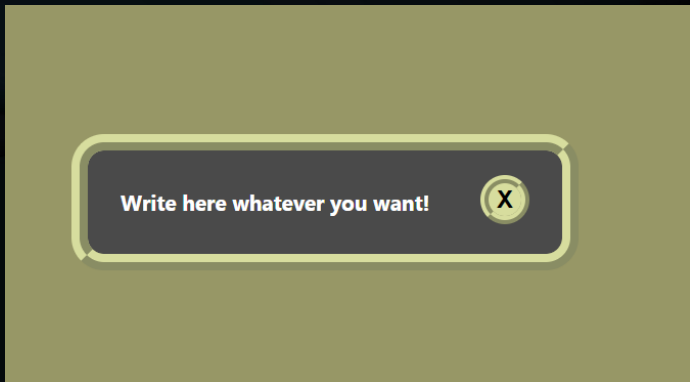
Pull Requests

0

Last publish

7 days ago

Collaborators



Réalisation App 1/4 :

✓ src

- > assets
- > data
- > ReactComponents
- > Redux
- > utils

✓ Redux

✓ Slice

⚙ employeeSlice.jsx

✓ store

⚙ store.jsx

✓ ReactComponents

- > App
- > components
- > pages

✓ ReactComponents

> App

✓ components

> Banner

> Footer

✓ Form

> DatePicker

> Dropdown

> Input

Form.css

⚙ Form.jsx

> Header

> Table

✓ pages

> Employees

> Error

> Home



Réalisation App 2/4 :

App.jsx

src > ReactComponents > App > App.jsx > ...

```
1  import { Route, Routes } from "react-router-dom";
2  import React, { Suspense, lazy } from "react";
3  import Header from "../components/Header/Header";
4
5  import "../App.css";
6
7  const Home = lazy(() => import("../pages/Home/Home"));
8  const Employees = lazy(() => import("../pages/Employees/Employees"));
9  const Error = lazy(() => import("../pages/Error/Error"));
10
11 function App() {
12   return (
13     <>
14       <Suspense fallback={<div>Loading...</div>}>
15         <Header />
16         <Routes>
17           <Route index element={<Home />} />
18           <Route path="employees" element={<Employees />} />
19           <Route path="*" element={<Error />} />
20         </Routes>
21       </Suspense>
22     </>
23   );
24 }
25
26 export default App;
```

App.jsx

Réalisation App 3/4 :

```
2
3  const initialState = {
4    employee: {
5      firstName: "",
6      lastName: "",
7      startDate: "",
8      department: "",
9      birthDate: "",
0      street: "",
1      city: "",
2      State: "",
3      zipCode: "",
4    },
5  };
```

```
src > Redux > Slice > employeeSlice.jsx > employeeSlice
1  import { createSlice } from "@reduxjs/toolkit";
2
```

```
export const { add } = employeeSlice.actions;
export default employeeSlice.reducer;
```

Actions et reducers créés via toolkit

Réalisation App 4/4 :

```
export default function Home() {  
  return (  
    <div className="home">  
      <Banner />  
      <div className="all-form">  
        <div className="form-header">  
          <div className="form-name">HRnet</div>  
          <div className="form-link">  
            <Link to="/employees">View Current Employees &#x21AA;</Link>  
          </div>  
        </div>  
        <Form />  
      </div>  
      <Footer />  
    </div>  
  );  
}
```

Séparation du code en petits composants

```
return (  
  <>  
    <form onSubmit={handleSubmit}>  
      <div className="form-description">Create Employee</div>  
      <Input  
        type="text" ...  
        setInput={setFirstName}  
      />  
      <Input  
        type="text" ...  
        setInput={setLastName}  
      />  
      <DatePicker  
        labelTitle="Birth Date:" ...  
      />  
      <DatePicker  
        labelTitle="Start Date:" ...  
      />  
      <div className="adress">  
        <Input  
          type="text" ...  
          setInput={setStreet}  
        />  
        <Input  
          type="text" ...  
          setInput={setCity}  
        />  
        <Dropdown  
          name="state" ...  
          datas={dataStates}  
        />  
      </div>  
    </form>  
  </>  
)
```



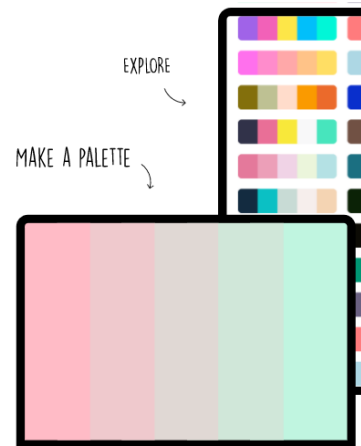
<https://colors.co/>

The super fast color schemes generator!

Create the perfect palette or get inspired by
thousands of beautiful color schemes.

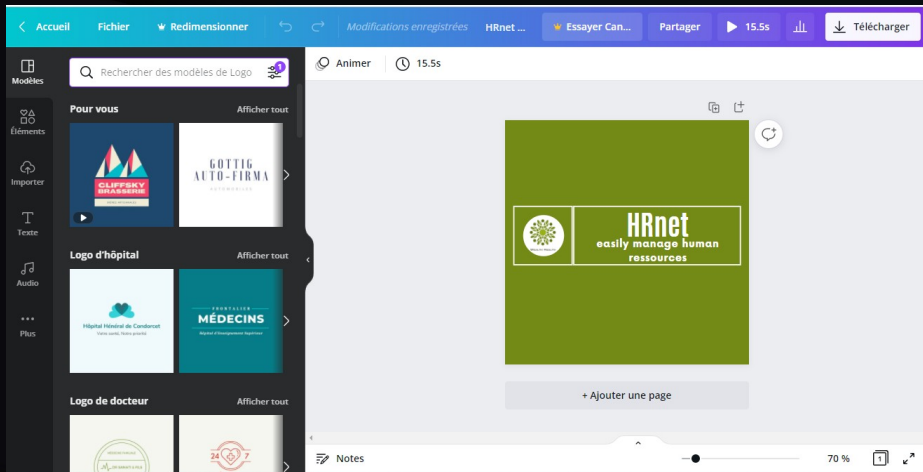
Start the generator!

Explore trending palettes



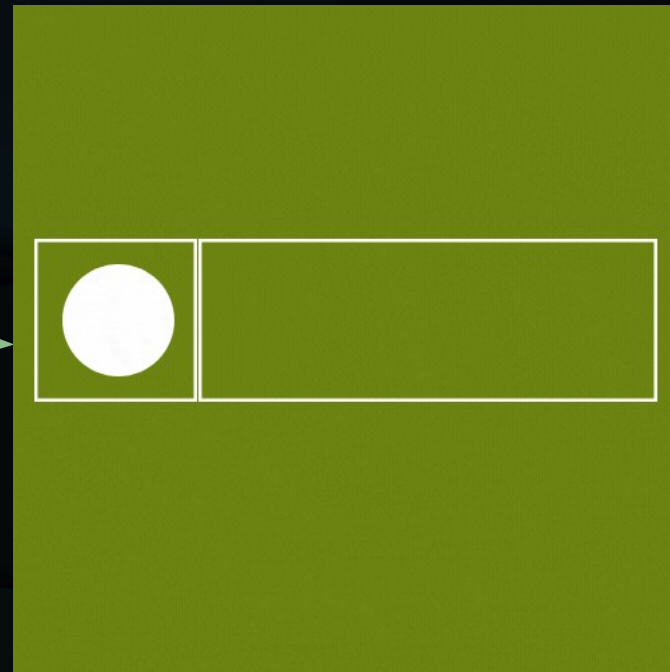
Génération d'une palette de couleurs

Design 2/3 :



<https://www.canva.com/>

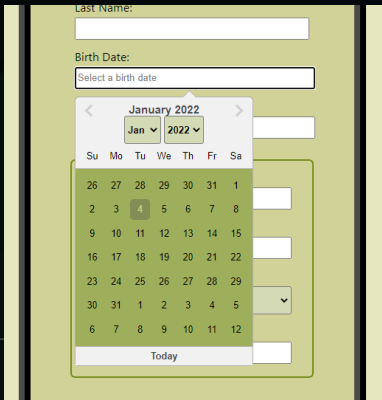
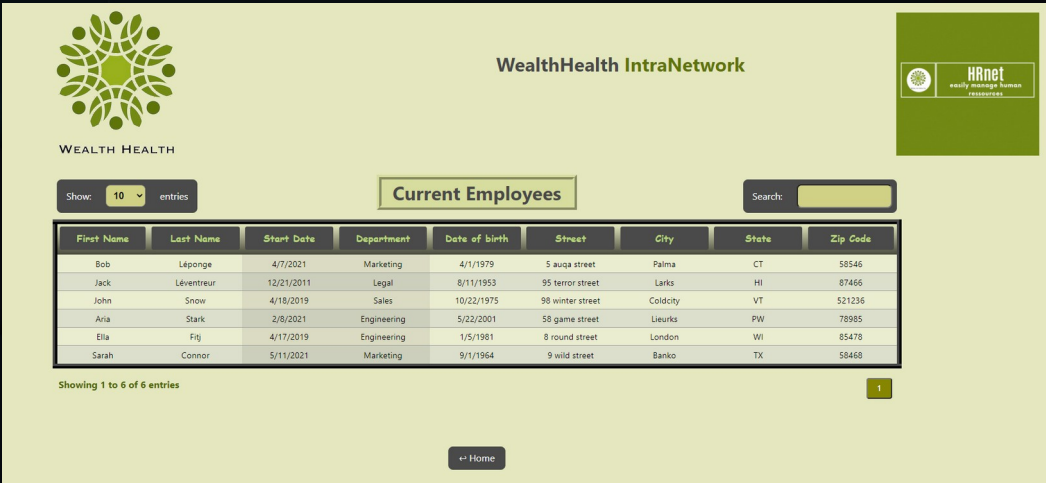
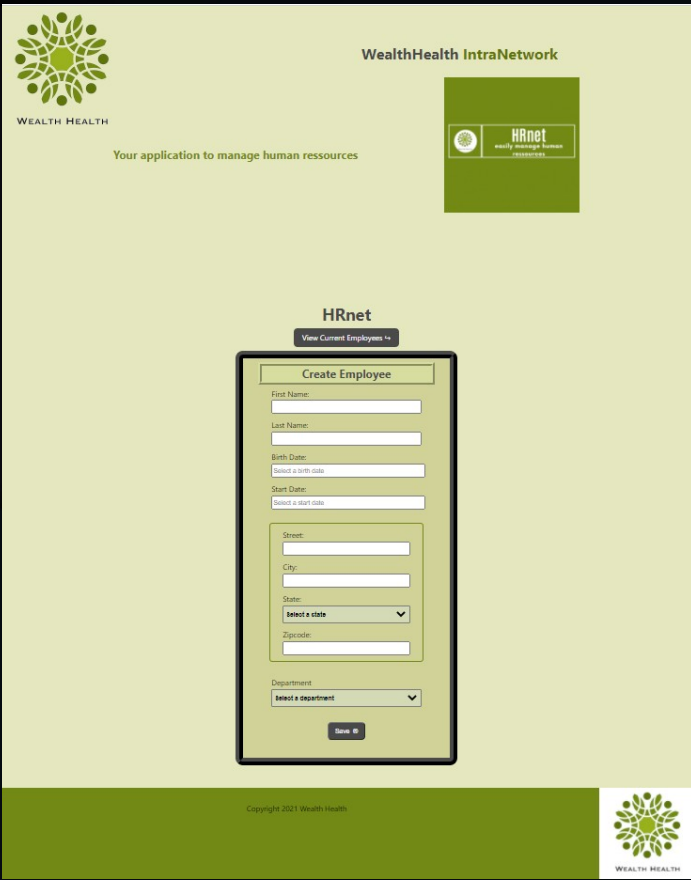
Création d'un logo pour l'appli





Design 3/3 :

Rendu

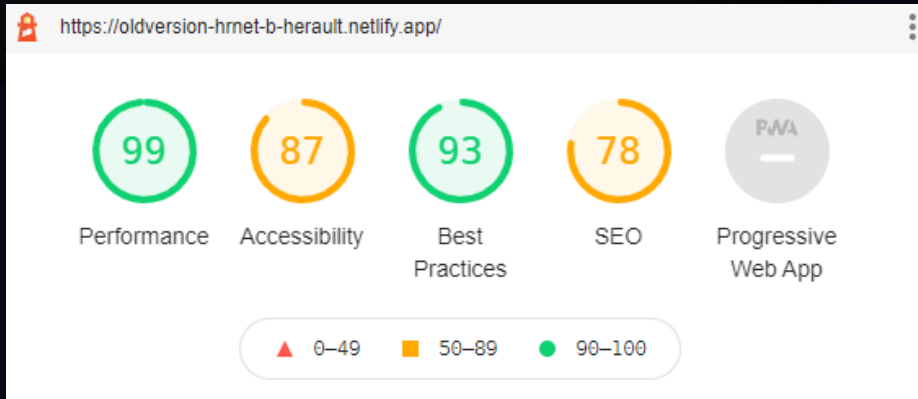


Performances 1/2 :

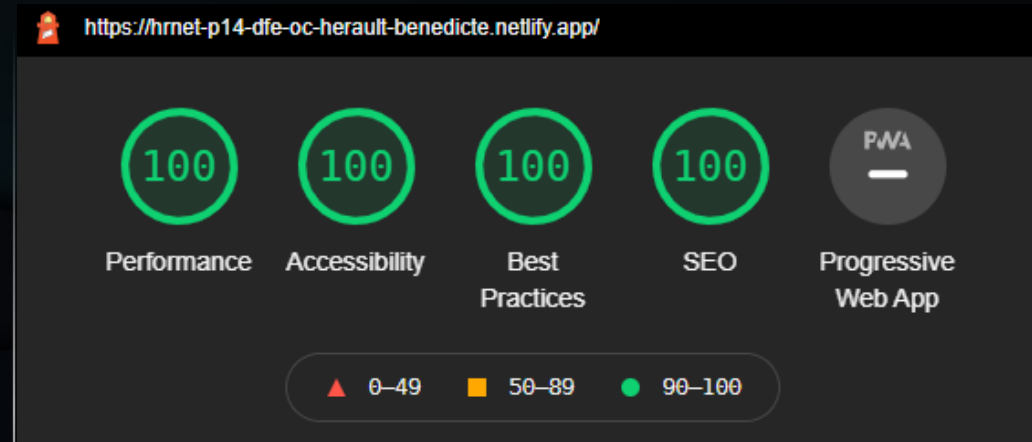
Page Home

Rapports disponibles en json

Ancienne version



Lighthouse



Nouvelle version



Performances 2/2 :

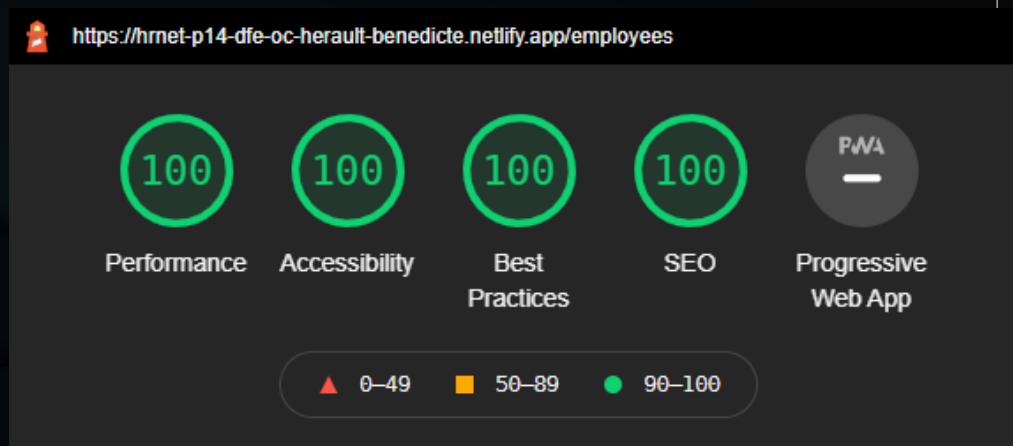
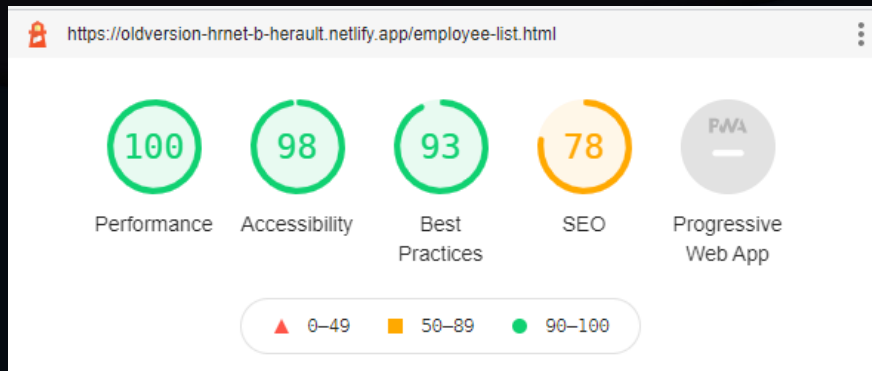
Page Employees
Rapports disponibles en json

Ancienne version

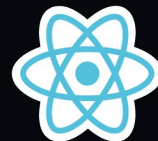
jQuery



Lighthouse



Nouvelle version





Your application to manage human resources

