

# SAS HealProcess

## **Note de synthèse**

du 19 Mai au 20 juin 2025

Réalisé par AKILABANA Joseph

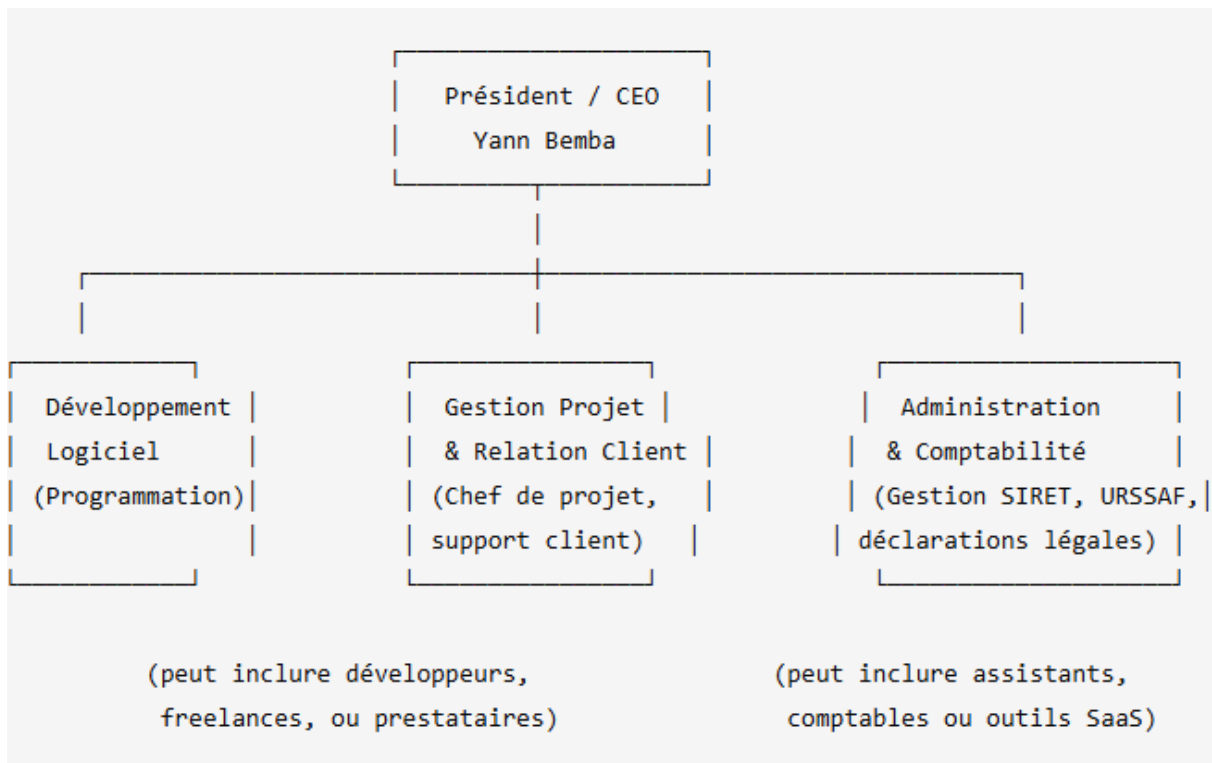
# Sommaire

Présentation.....	3
Organigramme.....	4
Projets (Portfolio Moderne).....	5
Introduction.....	5
Commandes utilisées et explication.....	6
Démarche de conception.....	7
Envoie de message non fonctionnel (résolu avec Formspree).....	7
Technologie et outils utilisés, problème rencontré et solution.....	8
README.md.....	11
Globals.css.....	14
Autonomie dans l'utilisation de Git et Github.....	16
Bilan.....	17
React en profondeur à partir de Zéro.....	18
Particularités de JSX.....	18
Les composants.....	18
Conclusion.....	20
Annexes.....	21
Annexe 1.....	21
Annexe 2.....	22
Glossaire.....	23

## Présentation

SAS HealProcess est une société française spécialisée dans le développement de solutions logicielles sur mesure, fondée le 23 avril 2021 par Yann Bemba. Installée à Mitry-Mory, elle intervient principalement dans le secteur de la programmation informatique, sous le code APE 6201Z. HealProcess accompagne les entreprises, en particulier les TPE, PME et professionnels indépendants, dans leur transformation numérique en leur proposant des solutions digitales adaptées à leurs besoins spécifiques. Forte d'une expertise technique solide et d'une approche orientée client, l'entreprise conçoit et développe des applications web et mobiles, automatise les processus métiers afin de réduire les tâches manuelles et améliorer la productivité, intègre des outils numériques en les synchronisant avec les systèmes déjà en place, et assure un accompagnement en architecture logicielle pour garantir performance, évolutivité des solutions déployées. Portée par des valeurs de proximité, de réactivité, de sécurité et d'innovation utile, HealProcess place l'utilisateur final au cœur de ses réflexions et conçoit des outils simples, fiables et adaptés à chaque environnement professionnel.

## Organigramme



Yann Bemba est non seulement le dirigeant de SAS HealProcess, mais également la personne qui s'occupe directement de moi. Il cumule plusieurs rôles opérationnels au sein de l'entreprise, ce qui lui permet de superviser chaque projet avec rigueur et implication. Les différents pôles de l'entreprise comme le développement logiciel, la gestion de projet, la relation client ou encore l'administration.

# Projets (Portfolio Moderne)

## Introduction

Dans le cadre de ma première année de BTS Services Informatiques aux Organisations (option SLAM), Lors de la première semaine de stage, Monsieur Yann Bamba me confie la tâche qui consiste à réaliser un projet : la création d'un **portfolio moderne et interactif pendant mon stage**. Ce projet avait pour but de **valoriser mes compétences acquises** tout au long de l'année et de **proposer une vitrine professionnelle** de mon parcours, mes projets, mes compétences, ma veille technologique etc.... .

Ce projet m'a permis de mobiliser de nombreuses connaissances acquises lors de ma formation, tout en me confrontant à des **problématiques techniques concrètes**, similaires à ce que l'on peut retrouver dans un environnement professionnel.

## Commandes utilisées et explication

```
137 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\akila\Desktop\portfolio\portfolio> npm run dev

> portfolio@0.1.0 dev
> next dev

⚠ Port 3000 is in use, using available port 3001 instead.
▲ Next.js 15.3.2
- Local:      http://localhost:3001
```

<code>npx create-next-app@latest</code>	Permet de créer le dossier en Next.js
<code>npm install react-icons</code>	permet d'ajouter rapidement des icônes diverses.
<code>npm install framer-motion</code>	permet de mettre en place des animations fluides.
<code>npm install @radix-ui/react-tabs</code>	permet aux onglets d'être accessibles et stylisés, je l' utilise particulièrement dans l'onglet BTS SIO.
<code>npm install lucide-react</code>	permet de mettre des icônes SVG modernes dans l'onglet contact.
<code>npm install autoprefixer postcss</code>	permet de faire en sorte que le CSS soit compatible avec tous les navigateurs.

## Démarche de conception

Le projet s'est déroulé en plusieurs phases :

- **Phase 1** : Maquettage (structure, choix de couleurs et typos, arborescence)
- **Phase 2** : Développement avec Next.js, création des composants
- **Phase 3** : Intégration des contenus (texte, projets, liens, images)
- **Phase 4** : Ajout des animations et effets dynamiques (Framer Motion, défilement fluide)
- **Phase 5** : Débogage, test responsive sur plusieurs tailles d'écran
- **Phase** : Déploiement du site sur Versel.

## Envoie de message non fonctionnel (résolu avec Formspree)

```
const res = await fetch('https://formspree.io/f/mgvkglen',
```

J'ai utilisé **Framer Motion** pour améliorer l'expérience utilisateur à travers des animations fluides, modernes et réactives, en envoyant un message, il stocke et l'envoie, il fait office de base de données en particulier dans l'onglet contact

<i>Nom du dossier</i>	<i>Son utilité</i>
node_modules/	Contient un bout de code pour le lancement du projet.
app/components	Permet d'organiser la structure du code en fonction des composants.
public/	Contient toutes les images qui s'affiche dans certaines pages.
env	Contient des paquets indispensables pour le déploiement de l'API.

## Technologie et outils utilisés, problème rencontré et solution

Je utilise Next.js (React), Tailwind CSS pour une personnalisation poussée, framer Motion (effets hover, transition d'apparition), **icônes** (Lucide React), Intégré avec Formspree pour gérer les envois de message et comme problème rencontré pendant le codage, il est beaucoup trop nombreux ça sera impossible pour moi de tout compte.

Il y a notamment le problème de structure de page dans le fichier Header.jsx et Page.js. ([voir Annexe 1](#))

Next.js est un **framework JavaScript basé sur React** qui permet de construire des applications web modernes, performantes et optimisées. Il est largement utilisé pour développer des sites web dynamiques, des blogs, des dashboards ou même des applications e-commerce.

```
import Header from '../components/Header';
import Hero from '../components/Hero';
// import Projects from '../components/Projects'; // supprimé
// import About from '../components/About'; // supprimé
// import Contact from '../components/Contact'; // supprimé
// import NotesDeSynthese from '../components/Notedessynte.jsx'; // supprimé
// import Veille from '../components/Veille.jsx'; // supprimé
// import Certification from '../components/Certification.jsx'; // supprimé
// import FormationDiplome from '../components/Diplome.jsx'; // supprimé
```

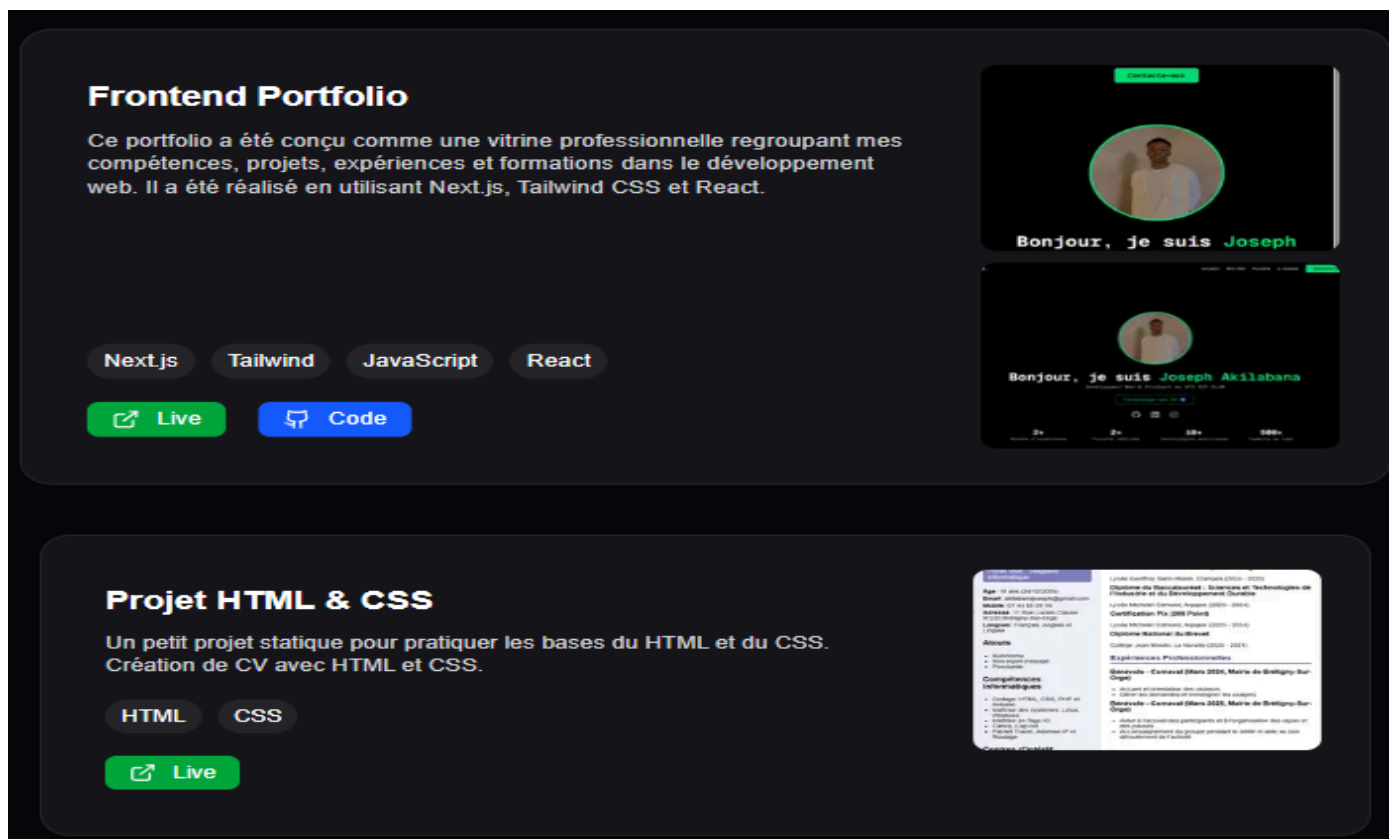
Au début je voulais juste que le code affiché le code du page d'accueil, le problème était que je ne savais pas où était le problème, j' essaye de regarder de vidéo pour m'aider je ne comprenais pas grand chose, à chaque fois que je lance localhost, dans ma page d'accueil il y a tout notamment le projets, contact, note de synthèse, veille technologique, certification et pour finir Diplôme.

il a fallu que je supprime tout cette page pour garder ma page d'accueil.

Les deuxième obstacles majeurs ont été liés à la **non-adaptabilité de certaines sections aux petits écrans** (mobiles, tablettes). Par exemple, certains composants comme les cartes projets ou le formulaire de



contact débordent de l'écran où s'affichent mal, ainsi sa compromettant l'expérience utilisateur mobile.



Voici les cartes projets qui s'adaptent parfaitement aux petits écrans (téléphone et tablette) actuellement.

Pour résoudre ce problème, j'ai adopté les Tailwind CSS (`sm:`, `md:`, `lg:`, etc.), permettant de définir des styles spécifiques selon la taille de l'écran. J'ai aussi structuré mes grilles (`grid`, `flex-col`, `flex-wrap`) pour rendre l'affichage plus fluide. Résultat : l'interface est désormais pleinement responsive et accessible quel que soit le support.

Quand j'ai voulu ajouter des animations dans mon portfolio avec Framer Motion (une librairie qui rend les éléments dynamiques et vivants), j'ai rencontré plusieurs problèmes. Certaines animations ne s'affichent pas du tout, D'autres étaient mal synchronisées et Les effets au survol ne fonctionnaient pas comme prévu.

J'ai pris le temps de comprendre chaque partie de l'animation notamment `transition={{ delay, duration, ease }}` : pour que l'animation démarre doucement, dure assez longtemps, et soit fluide. et pour finir le dernier problème que j'ai rencontré est que l'un des moments les plus frustrants a été lors de l'intégration du formulaire de contact. Initialement, l'envoi des messages ne fonctionnait pas : aucun e-mail n'était reçu et le formulaire ne renvoyait aucun feedback (c'est un retour d'information qu'on reçoit ou qu'on donne) à l'utilisateur. Pour corriger cela, j'ai exploré des solutions comme Formspree. J'ai créé un compte gratuit, mis en place un URL sécurisé (ex. <https://formspree.io/f/xabc123>) et intégré ce lien dans le formulaire HTML via l'attribut `action`. J'ai également ajouté des classes Tailwind pour styliser les champs et améliorer l'expérience utilisateur. Le tout fonctionne désormais parfaitement, avec confirmation visuelle post-envoi.

## README.md

This is a [Next.js](https://nextjs.org) project bootstrapped with [`create-next-app`](https://github.com/vercel/next.js/tree/canary/packages/create-next-app).

### ## Getting Started

First, run the development server:

```
``bash
npm run dev
# or
yarn dev
# or
pnpm dev
# or
bun dev
``
```

Open [http://localhost:3000](http://localhost:3000) with your browser to see the result.

You can start editing the page by modifying `app/page.js`. The page auto-updates as you edit the file.

This project uses

[`next/font`](https://nextjs.org/docs/app/building-your-application/optimizing/fonts) to automatically optimize and load [Geist](https://vercel.com/font), a new font family for Vercel.

### ## Learn More

To learn more about Next.js, take a look at the following resources:

- [Next.js Documentation](https://nextjs.org/docs) - learn about Next.js features and API.
- [Learn Next.js](https://nextjs.org/learn) - an interactive Next.js tutorial.

```
You can check out [the Next.js GitHub repository] (https://github.com/vercel/next.js) - your feedback and contributions are welcome!
```

## Deploy on Vercel

The easiest way to deploy your Next.js app is to use the [Vercel Platform] ([https://vercel.com/new?utm\\_medium=default-template&filter=next.js&utm\\_source=create-next-app&utm\\_campaign=create-next-app-readme](https://vercel.com/new?utm_medium=default-template&filter=next.js&utm_source=create-next-app&utm_campaign=create-next-app-readme)) from the creators of Next.js.

Check out our [Next.js deployment documentation] (<https://nextjs.org/docs/app/building-your-application/deploying>) for more details.

Le projet a commencé avec un outil automatique qui crée la base du site, ce qui m'a permis de voir comment un projet web est organisé. J'ai découvert notamment que le projet utilise un dossier spécial `app/` pour gérer les pages, ce qui est une nouvelle méthode dans Next.js. J'ai aussi vu que Next.js permet de charger les polices de manière efficace, ce qui améliore les performances du site.

Le but était de mettre le site en ligne sur la plateforme Vercel, qui est bien intégrée avec Next.js. Mais avant d'y arriver, j'ai rencontré plusieurs problèmes techniques. Par exemple, je ne pouvais pas lancer le projet sur mon ordinateur parce qu'un dossier important appelé `.next` manquait. J'ai compris qu'il fallait d'abord lancer une commande pour construire le projet avant de pouvoir l'ouvrir. Ensuite, il y avait deux dossiers `node_modules` dans le projet, ce qui créait des conflits et empêchait l'installation des modules nécessaires. J'ai dû nettoyer ces dossiers pour que ça fonctionne.

J'ai aussi eu des erreurs liées aux permissions de certains fichiers qui empêchaient la suppression automatique de fichiers pendant l'installation. Pour résoudre ça, j'ai dû fermer certains programmes en cours et supprimer manuellement ces fichiers bloqués. J'ai aussi vu des avertissements de Git à propos des fins de ligne dans les fichiers, mais

ça n'a pas empêché le projet de fonctionner. Cela m'a fait comprendre que certains détails techniques peuvent parfois poser problème, même s'ils ne bloquent pas directement le travail.

Après avoir réglé tous ces problèmes, j'ai pu déployer le site sur Vercel. La connexion avec GitHub a permis que chaque mise à jour du code soit automatiquement envoyée en ligne, ce qui est très pratique pour suivre les changements. J'ai configuré la commande pour construire le site avant le déploiement, et j'ai aussi ajouté un fichier PDF dans le dossier public pour qu'il soit accessible depuis le site.

## Globals.css

```
@media (prefers-color-scheme: dark) {
  :root {
    --background: #0a0a0a;
    --foreground: #5c1010;
  }
}

body {
  background: var(--background);
  color: var(--foreground);
  font-family: Arial, Helvetica, sans-serif;
}

@keyframes spin-slow {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

.animate-spin-slow {
  animation: spin-slow 10s linear infinite;
}
```

Dans mon fichier `globals.css`, j'ai créé des styles qui s'appliquent partout sur mon site pour que l'apparence soit cohérente. J'ai essayé d'utiliser une règle spéciale appelée *media query* pour détecter si l'utilisateur préfère le mode sombre (un fond sombre au lieu d'un fond clair), mais ça n'a pas fonctionné comme prévu.

J'ai aussi défini deux couleurs principales sous forme de variables, une pour le fond (`--background`) et une pour le texte (`--foreground`). Cela me permet de changer facilement les couleurs partout dans le projet sans devoir modifier chaque élément un par un.

Pour la page entière (`body`), j'utilise ces couleurs pour avoir un fond sombre et un texte clair, avec une police simple et facile à lire comme Arial ou Helvetica.

Enfin, j'ai ajouté une petite animation que j'ai appelée `spin-slow`. Grâce à une règle spéciale, elle fait tourner un élément lentement, en faisant un tour complet en 10 secondes. Cette animation peut être appliquée à n'importe quel élément grâce à une classe appelée `.animate-spin-slow`.

Ce fichier me sert donc à créer une base propre, agréable et un peu dynamique pour tout le site. ([voir Annexe 1](#))

## Autonomie dans l'utilisation de Git et Github

```
Path: /d/workspace/cours_git/clone_mon_site_web (master)
$ echo "mon site web pour le cours git" >> README.md
Path: /d/workspace/cours_git/clone_mon_site_web (master)
$ cat README.md
mon site web pour le cours git
Path: /d/workspace/cours_git/clone_mon_site_web (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
Path: /d/workspace/cours_git/clone_mon_site_web (master)
$ git commit -m"#1: ajout du readme"
[master 71d33d2] #1: ajout du readme
1 file changed, 1 insertion(+)
create mode 100644 README.md
Path: /d/workspace/cours_git/clone_mon_site_web (master)
$ git log
```

Sur cette image, on voit une étape importante avec Git, l'outil que j'utilise pour gérer les versions de mon projet. J'ai commencé par créer un fichier appelé `README.md` où j'ai écrit une petite description de mon projet. Pour ça, j'ai utilisé une commande qui permet d'ajouter du texte dans un fichier ou de créer ce fichier s'il n'existe pas encore.

```
echo "mon site web pour le cours git" >> README.md
```

Le fichier `README.md` est très courant dans les projets avec Git, car il explique en quelques lignes ce que fait le projet ou comment l'utiliser. Ensuite, j'ai affiché le contenu du fichier avec une autre commande pour vérifier que mon texte était bien enregistré.

```
git commit -m "#1: ajout du readme"
```

Après ça, j'ai sauvegardé officiellement cette modification avec un *commit*. Le commit est comme une photo de mon projet à un instant donné. J'ai mis un message clair pour expliquer ce que j'avais changé, ce qui aide à suivre l'évolution du projet plus facilement. Git a aussi créé automatiquement un identifiant unique pour ce commit.

```
git log
```



## Bilan

Pendant ce cours, j'ai appris plusieurs commandes Git importantes et leur utilité. Pour commencer, j'ai installé et configuré Git en définissant mon nom d'utilisateur et mon adresse e-mail avec les commandes `git config --global user.name` et `git config --global user.email`. Ces informations permettent d'identifier qui fait les modifications dans le projet.

Ensuite, j'ai appris à enregistrer les changements faits dans les fichiers. La commande `git add` sert à préparer les fichiers modifiés en les ajoutant à une zone appelée « staging area ». Une fois que c'est fait, j'utilise la commande `git commit -m "message"` pour sauvegarder officiellement ces changements dans l'historique du projet, en ajoutant un message qui explique ce que j'ai fait.

Pour voir l'historique des modifications, j'utilise la commande `git log`. Si je veux voir précisément ce qui a été modifié dans un commit, j'utilise la commande `git show`. J'ai aussi découvert comment revenir à une version précédente du projet ou changer de branche avec la commande `git checkout`.

Dans un contexte de travail en équipe, il est important d'utiliser des dépôts distants comme GitHub, qui permettent de stocker le code en ligne. Pour récupérer un projet depuis GitHub, j'utilise `git clone`. Après avoir fait des modifications, je peux envoyer mes changements avec `git push`, et récupérer ceux des autres avec `git pull`.

Parfois, quand plusieurs personnes modifient les mêmes fichiers, il peut y avoir des conflits. J'ai appris à les résoudre avec les commandes `git merge` ou `git rebase`. Enfin, pour mieux organiser le travail, j'utilise les branches avec la commande `git branch`. Cela permet de développer de nouvelles fonctionnalités sans changer la version principale du projet.

# React en profondeur à partir de Zéro

## Particularités de JSX

j'ai appris ses particularités, notamment avec JSX, qui ressemble à du HTML mais qui a ses propres règles. Par exemple, dans React, on utilise `className` au lieu de `class` pour définir une classe CSS, et certaines balises doivent être auto-fermées comme `<input type='text' />`. Même si ça ressemble à du HTML, React a ses propres spécificités.

```
<div className="flex flex-col md:flex-row md:justify-between items-start gap-4 text-gray-300">
  <p className="flex items-center gap-2"><Phone size={18} className="text-green-400" /> 07 44 85 59 14</p>
  <p className="flex items-center gap-2"><Mail size={18} className="text-green-400" /> akilabana joseph@gmail.com</p>
  <p className="flex items-center gap-2"><MapPin size={18} className="text-green-400" /> 91220 Brétigny-Sur-Orge</p>
</div>
</section>
```

## Les composants

En React, les composants (ou components) sont très importants. Ce sont des petits blocs de code indépendants qui servent à construire chaque partie d'une application. Par exemple, un bouton, une carte, un menu ou même une page entière peuvent être des composants. Cette méthode rend le développement plus organisé, plus clair et facilite la réutilisation du code.

Dans mon projet de portfolio réalisé en première année de BTS SIO SLAM, j'ai utilisé cette méthode. Chaque partie du site, comme le menu, le formulaire de contact ou la liste de mes diplômes, est codée dans un fichier séparé sous forme de composant. Cela rend le projet plus facile à comprendre et à modifier.

Par exemple, j'ai un fichier `Header.jsx` pour la barre de navigation, `Contact.jsx` pour le formulaire avec ses champs et icônes, et `DiplomeFormation.jsx` pour afficher mes diplômes avec des effets visuels. Grâce aux composants, j'ai pu éviter de répéter du code et ajouter facilement du style avec Tailwind CSS ou des animations avec Framer Motion. ([voir Annexe 2](#))

Un des grands avantages de cette méthode est que le code est plus clair. Chaque fichier a une seule fonction précise. Le routage, c'est-à-dire la gestion des pages, est assuré par Next.js dans le dossier `/app`, et chaque page utilise les composants dont elle a besoin. Cela m'a permis de bien séparer la partie visible du site (l'interface utilisateur) de la navigation. En cas de problème ou pour améliorer quelque chose, je sais exactement où chercher dans mon code.

## Conclusion

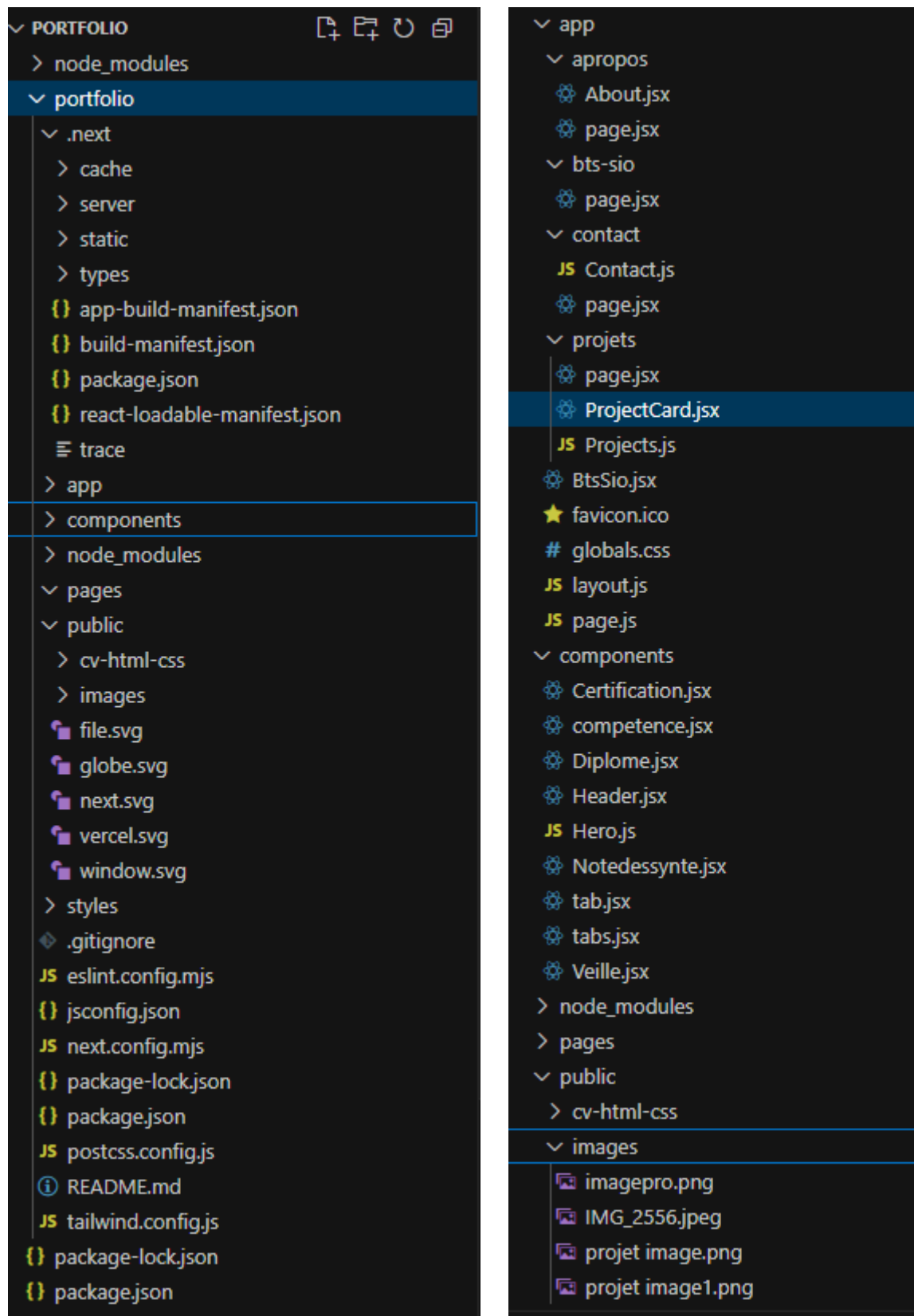
En conclusion, j'ai effectué 5 semaines de stage dans l'entreprise SAS healProcess. Grâce à ce stage, j'ai pu découvrir le monde professionnel dans un autre type d'organisation que dans une entreprise. Cela m'a permis de me rendre compte de comment est vraiment le milieu professionnel, quelles sont les exigences pour la création du site web et comment cette mise en place est procédée.

Effectuer ce stage permet d'acquérir une nouvelle compétence qui implique d'effectuer plusieurs tâches en même temps : c'est le fait d'être multitâche. Cette dernière m'a permis de m'y habituer.

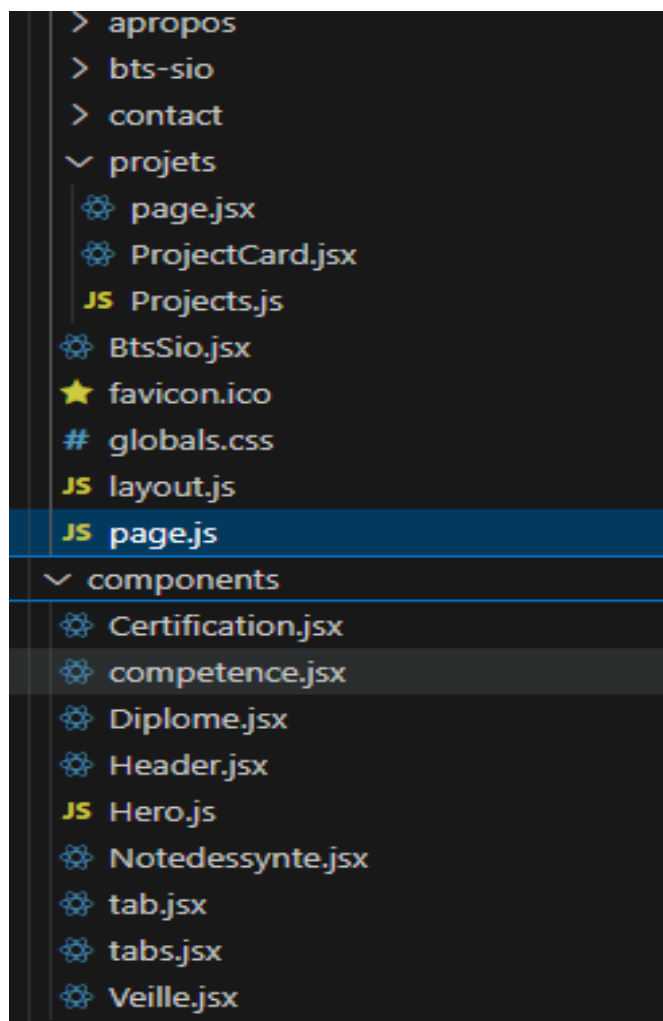
Je remercie notamment Monsieur Yann BEMBA de m'avoir encouragé et aidé dans ma Création de mon portfolio moderne (le fait de le moderniser avec acertinity ui) et également de m'avoir donné des conseils. Un grand merci à lui pour tout, du début et à la fin tous ainsi que les membres du personnel du SAS healProcess de m'avoir accueilli et félicité d'avoir obtenu une formation qui me plaît.

# Annexes

## Annexe 1



## Annexe 2



# Glossaire

*Ce dictionnaire contient un ensemble de termes techniques afin de mieux comprendre ses significations.*

Build	Étape de préparation avant de mettre le site en ligne. Elle optimise les fichiers du projet.
Media query	Règle CSS qui permet d'adapter le style d'un site selon la taille de l'écran ou le mode (sombre ou clair).
Animation (keyframes)	Ensemble de règles CSS qui permet de créer des effets animés (rotation, mouvement, etc.).
Vercel	Plateforme de déploiement utilisée pour mettre en ligne un site web développé avec Next.js. Permet une mise à jour automatique après chaque <code>git push</code> .
Framer Motion	Librairie React utilisée pour ajouter facilement des animations (comme des transitions douces, des effets de mouvement, etc.).
JSX	Syntaxe utilisée dans React qui ressemble à du HTML, mais avec des particularités propres à React (comme <code>className</code> au lieu de <code>class</code> ).