**Étape 1 : Structurer et styliser chaque composant avec styled-components**

Nous allons commencer par travailler sur les composants un à un pour ajouter des styles appropriés. On va se concentrer sur un design moderne, responsive et fluide.

**1. Header.jsx**

Le header doit donner une première bonne impression. Il doit être simple et épuré, mais aussi pratique avec des liens vers les différentes sections de ton portfolio.

**Header.jsx :**

jsx

Copier

import React from 'react';

import styled from 'styled-components';

const HeaderContainer = styled.header`

background-color: #2b2d42;

color: #fff;

padding: 20px 0;

text-align: center;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

`;

const Nav = styled.nav`

margin-top: 20px;

`;

const NavList = styled.ul`

list-style: none;

padding: 0;

display: flex;

justify-content: center;

gap: 30px;

`;

const NavItem = styled.li`

a {

text-decoration: none;

color: #fff;

font-size: 18px;

transition: color 0.3s;

&:hover {

color: #e63946;

}

}

`;

const Header = () => {

return (

<HeaderContainer>

<h1>Ton Nom</h1>

<p>Développeur Web | Intégrateur Front-End</p>

<Nav>

<NavList>

<NavItem><a href="#about">À propos</a></NavItem>

<NavItem><a href="#skills">Compétences</a></NavItem>

<NavItem><a href="#projects">Projets</a></NavItem>

<NavItem><a href="#contact">Contact</a></NavItem>

</NavList>

</Nav>

</HeaderContainer>

);

};

export default Header;

**Explication :**

* Nous avons créé un HeaderContainer qui contient l'en-tête principal avec une couleur sombre pour donner un effet moderne et professionnel.
* La navigation est centrée, avec un effet de survol sur les liens pour rendre l’interaction plus fluide.

**2. À propos de moi (About.jsx)**

Dans cette section, tu vas mettre en avant ton parcours et tes valeurs. Utilise un style épuré avec une photo de toi et une petite biographie.

**About.jsx :**

jsx

Copier

import React from 'react';

import styled from 'styled-components';

const AboutContainer = styled.section`

padding: 80px 20px;

background-color: #f1faee;

text-align: center;

`;

const AboutContent = styled.div`

max-width: 800px;

margin: 0 auto;

line-height: 1.6;

`;

const ImageContainer = styled.div`

margin-bottom: 30px;

`;

const ProfileImage = styled.img`

width: 150px;

height: 150px;

border-radius: 50%;

object-fit: cover;

`;

const About = () => {

return (

<AboutContainer id="about">

<h2>À propos de moi</h2>

<ImageContainer>

<ProfileImage src="path-to-your-photo.jpg" alt="Photo de profil" />

</ImageContainer>

<AboutContent>

<p>Je suis un développeur web passionné par l’intégration front-end. J'aime créer des expériences utilisateurs intuitives et esthétiques tout en respectant les standards du web moderne.</p>

<p>Mon expertise : HTML5, CSS3, JavaScript, React.js, et bien plus...</p>

</AboutContent>

</AboutContainer>

);

};

export default About;

**Explication :**

* Utilisation de l'élément ProfileImage pour afficher ta photo.
* La section est centrée, avec une couleur de fond claire et moderne (#f1faee).

**3. Compétences (Skills.jsx)**

La section des compétences doit être claire et facile à lire. Une bonne pratique est d'utiliser des icônes ou des barres de progression pour afficher tes compétences techniques.

**Skills.jsx :**

jsx

Copier

import React from 'react';

import styled from 'styled-components';

const SkillsContainer = styled.section`

padding: 80px 20px;

background-color: #edf2f4;

text-align: center;

`;

const SkillList = styled.ul`

list-style: none;

padding: 0;

display: flex;

justify-content: center;

flex-wrap: wrap;

gap: 30px;

`;

const SkillItem = styled.li`

background-color: #2b2d42;

color: #fff;

padding: 20px;

border-radius: 10px;

width: 120px;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

transition: transform 0.3s;

&:hover {

transform: translateY(-10px);

}

svg {

font-size: 40px;

margin-bottom: 10px;

}

`;

const Skills = () => {

return (

<SkillsContainer id="skills">

<h2>Compétences</h2>

<SkillList>

<SkillItem>

<svg> {/\* Insère ton icône ici \*/} </svg>

<p>React</p>

</SkillItem>

<SkillItem>

<svg> {/\* Insère ton icône ici \*/} </svg>

<p>HTML5</p>

</SkillItem>

<SkillItem>

<svg> {/\* Insère ton icône ici \*/} </svg>

<p>CSS3</p>

</SkillItem>

<SkillItem>

<svg> {/\* Insère ton icône ici \*/} </svg>

<p>JavaScript</p>

</SkillItem>

</SkillList>

</SkillsContainer>

);

};

export default Skills;

**Explication :**

* Les compétences sont affichées sous forme de cartes avec des icônes.
* Chaque carte a un effet de survol qui lui donne un petit mouvement, rendant l’interaction plus fluide.

**Étape 2 : Ajouter un système de navigation fluide**

On va maintenant rendre la navigation fluide avec des liens d'ancrage, pour que lorsque l'on clique sur une section, l'utilisateur soit redirigé directement vers celle-ci.

Tu peux utiliser l'attribut id dans chaque section (comme déjà fait pour #about, #skills, etc.) pour chaque composant afin d'améliorer l'expérience utilisateur.

**Étape 3 : Optimisation responsive**

La structure du design est déjà assez moderne, mais nous devons nous assurer que le portfolio est entièrement **responsive**.

Dans chaque composant, tu peux ajouter des règles CSS pour les petits écrans. Voici un exemple pour le Header.jsx pour que la navigation se transforme en une barre de navigation verticale sur mobile :

jsx

Copier

const NavList = styled.ul`

list-style: none;

padding: 0;

display: flex;

justify-content: center;

gap: 30px;

flex-wrap: wrap;

@media (max-width: 768px) {

flex-direction: column;

gap: 20px;

}

`;

**Étape 4 : Ajouter un Footer**

Le footer est une petite section à la fin de ton site où tu peux ajouter des informations supplémentaires comme un copyright, des liens vers tes réseaux sociaux, etc.

**Footer.jsx :**

jsx

Copier

import React from 'react';

import styled from 'styled-components';

const FooterContainer = styled.footer`

background-color: #2b2d42;

color: #fff;

padding: 20px;

text-align: center;

`;

const Footer = () => {

return (

<FooterContainer>

<p>© 2025 Ton Nom - Développeur Web</p>

</FooterContainer>

);

};

export default Footer;

**Étape 5 : Tester et déployer**

Une fois toutes les sections créées et stylisées, il est important de tester ton portfolio sur différents appareils et navigateurs pour s'assurer de sa **réactivité** et de sa **performance**.

Ensuite, tu pourras **déployer ton portfolio sur GitHub Pages** comme expliqué plus tôt, en utilisant gh-pages.

**Conclusion**

Nous avons abordé les sections principales du portfolio et utilisé styled-components pour créer un design moderne et réactif. Chaque composant est bien structuré, avec une interface épurée et professionnelle.

Si tu es prêt, continue à ajouter tes projets dans la section "Projets", et je serai heureux de t'aider pour cette étape, ou si tu as des questions concernant le déploiement ou la personnalisation du design. 😄