

Documentation

PROJET : MON PORTFOLIO

AUTEUR : FOY ORIANE

Sommaire

1. Introduction	3
1.1. Contexte.	3
1.2. Objectifs.	3
1.3. Technologies utilisées.....	3
2. Cahier des charges	3
2.1. Besoins fonctionnels.	3
2.2. Contraintes techniques.....	3
3. Conception du site	4
3.1. Maquette interface.	4
3.2. Charte graphique	5
3.3. Arborescence du site.	7
4. Documentation technique.....	8
4.1. Structure HTML	8
4.2. Mise en forme CSS.....	11
4.3. JavaScript	13
Fonctions pour la transition du fond:.....	13
Fonction pour l'animation de particule :	21
4.4 Responsive.	22
5. Tests et validation	24
5.1. Tests de compatibilité.	24
5.2 Tests de responsive design.....	25
6. Documentation utilisateur	25
6.1. Navigation.....	25
6.2 M'envoyer un message.....	27

6.3 Télécharger mon CV.....	27
6.3. Mise à jour du contenu.	28
7.Conclusion	29
7.1 Bilan.	29
7.2 Améliorations possibles.....	29

1. Introduction

1.1. Contexte.

Dans le cadre de ma formation en BTS Services Informatiques aux Organisations, option Solutions Logicielles et Applications Métiers (SLAM), j'ai réalisé un portfolio personnel.

Outre le simple devoir, ce projet s'inscrit dans une démarche professionnelle : il me permet de mettre en avant mes compétences techniques, mes projets et ma progression en développement web.

En effet, le portfolio constitue un outil de communication important pour tout développeur, car il offre une vitrine claire de son savoir-faire et de sa créativité.

1.2. Objectifs.

L'objectif de ce projet est de créer un site web personnel me permettant de me présenter, de mettre en avant mes projets et de démontrer mes compétences techniques.

Il s'agit également de concevoir mon premier site en utilisant les langages HTML, CSS et JavaScript, tout en développant une présence en ligne pouvant m'aider à saisir de futures opportunités professionnelles.

1.3. Technologies utilisées.

Langages : HTML, CSS, JavaScript.

Conception interface : Figma, Photoshop.

IDE: PhpStorm (JetBrains).

2. Cahier des charges

2.1. Besoins fonctionnels.

- Le site doit présenter mon profil et mes compétences.
- Le visiteur doit pouvoir consulter mes projets.
- Un formulaire de contact doit permettre de m'écrire facilement.
- Toutes les pages doivent être accessibles depuis les autres.

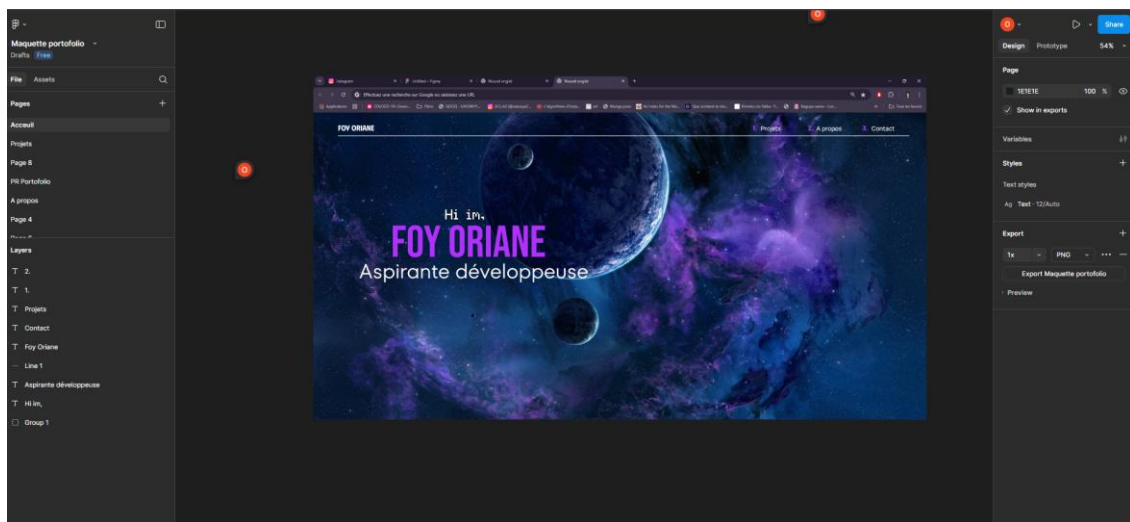
2.2. Contraintes techniques

- Site statique (sans base de données) -> Hébergé sur GitHub Pages.
- Compatible sur ordinateurs, tablettes et smartphones.

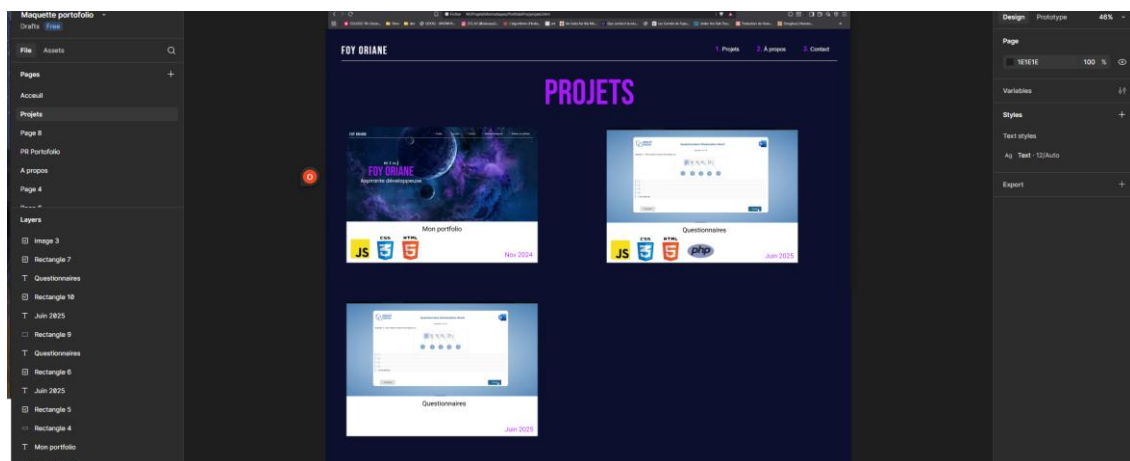
3. Conception du site

3.1. Maquette interface.

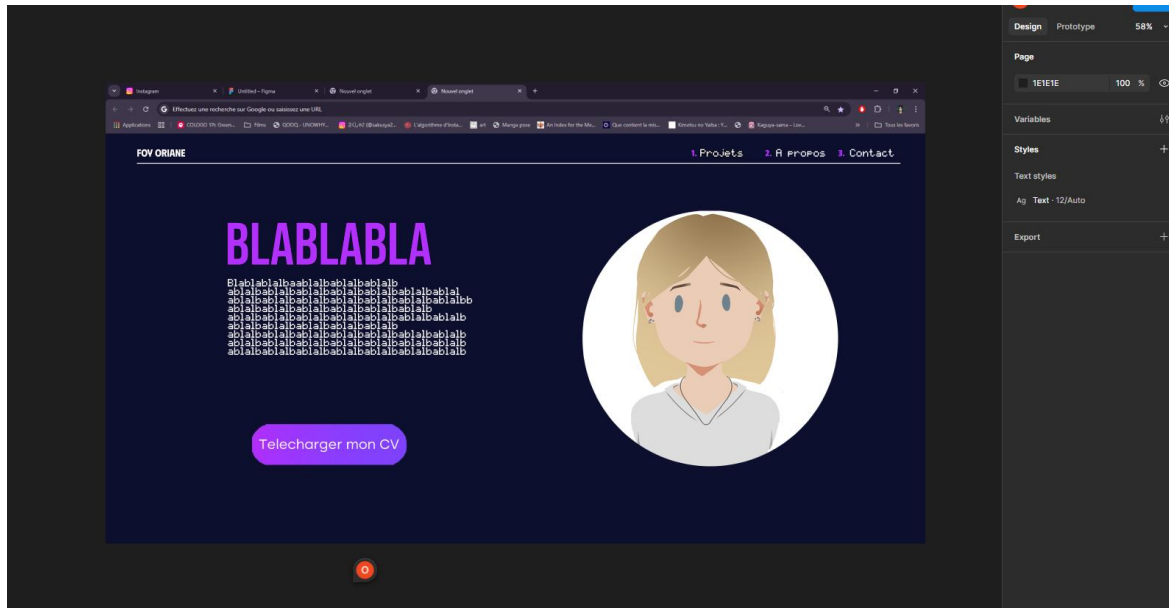
Chaque page a été préalablement conçue sur Figma, ce qui a permis de centraliser la réflexion sur l'interface en un seul temps, afin d'optimiser l'efficacité et la cohérence lors du développement.



Maquette de la page « index »



Maquette de la page « projets »



Maquette de la page « à propos »

Dans le cadre de ma présentation sur la page « À propos », j’ai également mis à profit mes compétences en infographie pour réaliser un avatar illustrant la page. Celui-ci a été créé avec Photoshop à l’aide d’une tablette graphique.

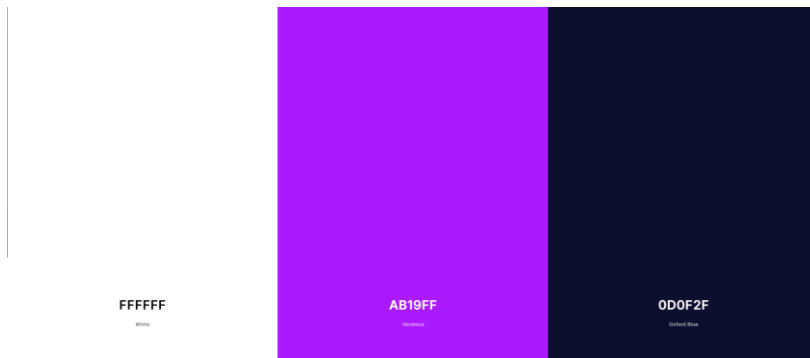
3.2. Charte graphique

Pour garantir la cohérence et l’harmonie graphique du site, j’ai établi des normes visuelles sous la forme d’une charte graphique :

Un nombre limité de couleurs a été choisi, harmonieuses entre elles, de différentes “valeurs” et respectant un thème précis, ici celui de l’espace, en accord avec l’image de fond sélectionnée pour la page d’accueil.



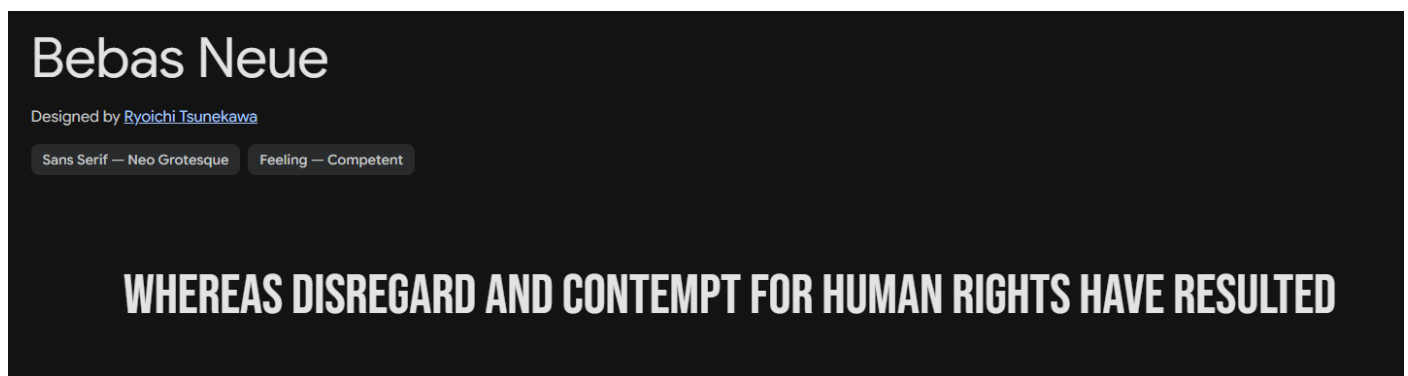
Code déclarant les variables couleurs



Palette de couleurs du site

Le choix d'un nombre limité de polices selon l'importance du texte dans la page (titres, paragraphes...) :

Exemples :



Police Bebas Neue pour les titres



Police Montserrat pour les paragraphes

3.3. Arborescence du site.

```
/
└─ PortfolioPro
    ├── .gitattributes
    ├── apropos.html
    ├── competences.html
    ├── contact.html
    ├── index.html
    ├── projets.html
    ├── styles
    │   ├── cssIndex.css
    │   ├── cssPages.css
    │   └── style.css
    ├── script
    │   ├── carrousel.js
    │   ├── particle.js
    │   └── script.js
    ├── projets
    │   ├── location_voiture.html
    │   ├── portfolio.html
    │   └── questionnaires.html
    ├── documents
    │   ├── Location
    │   └── dolfi
    ├── .idea
    │   ├── .gitignore
    │   ├── modules.xml
    │   ├── php.xml
    │   ├── portofolio-oriane.iml
    │   ├── vcs.xml
    │   └── workspace.xml
```

4. Documentation technique

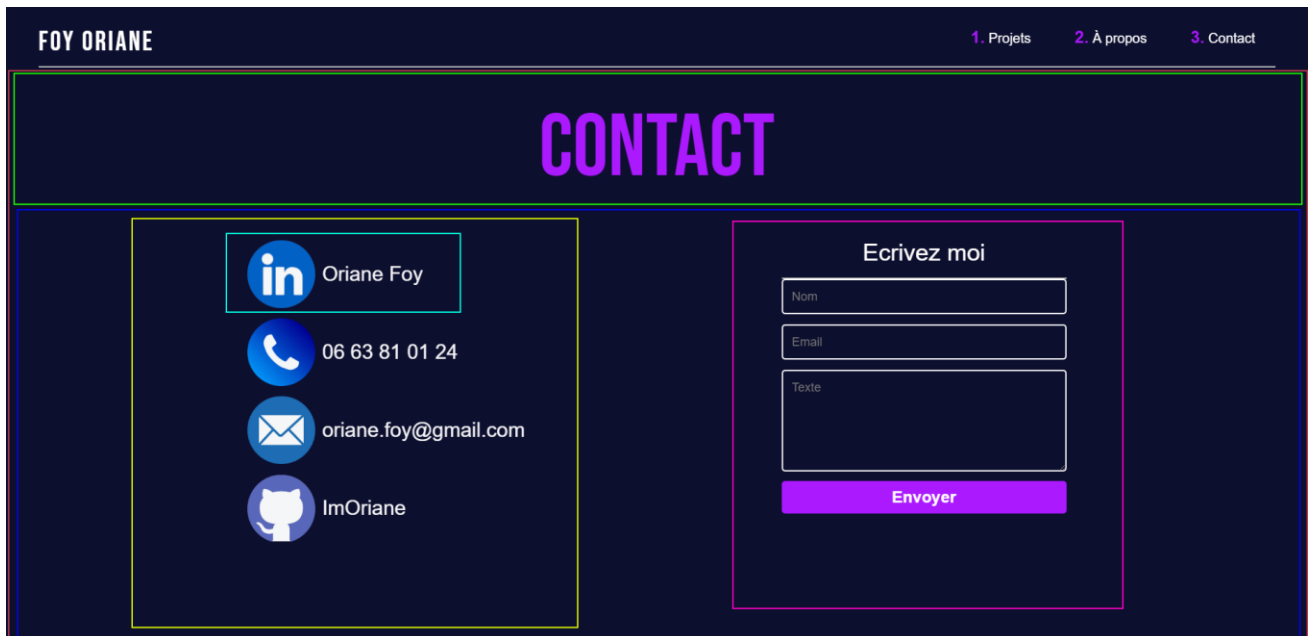
4.1. Structure HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Portfolio Oriane</title>
7      <link rel="shortcut icon" type="png" href="documents/favicon.png"/> // Le favicon
8      <link rel="stylesheet" href="styles/style.css" /> // Appel du css
9      <link rel="stylesheet" href="styles/cssPages.css" /> // Appel du css
10
11     // Toutes les références de style comme les polices...
12     .....
13
14     <script src="script/script.js"></script> // Appel du script JS
15 </head>
16 <body>
17 <header>
18     <div class="bg" id="bg"></div> // La div pour la couleur/image du fond présente dans toutes les pages
19
20     <div class="credit left"><a href="index.html">FOY ORIANE</a></div>
21     <nav class="right">
22         .....
23         // HTML de la barre de navigation présente sur toutes les pages
24         .....
25     </nav>
26     <hr/>
27 </header>
28
29 <main>
30     .....
31     // Ensemble de div formant le contenu unique à la page
32     .....
33 </main>
34
35 <script>
36     fadeBackground(document.referrer, "contact.html"); //Appel de la fonction js pour gérer le fondu de background.
37 </script>
38 </body>
39 </html>
40
```

Exemple structure type

Le contenu du main est composé de plusieurs div pour permettre la mise en forme de la page :

Exemple :



Structure HTML page « contact »

```

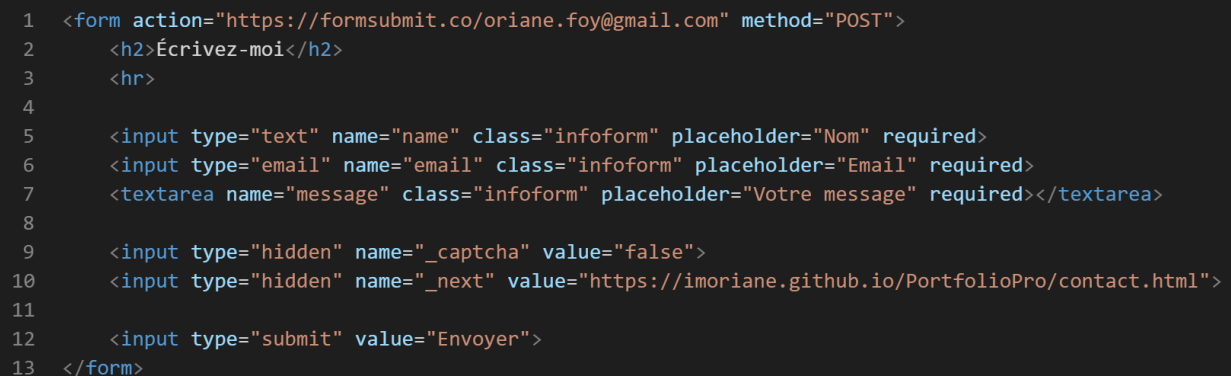
1  <main>
2    <div class="titrecontact"> <h1>Contact</h1></div>
3
4  <div class="contact-page">
5
6
7    <div class="listecontact">
8      <div>
9         Oriane Foy
10      </div>
11      <div>
12         06 63 81 01 24
13      </div>
14      <div>
15         oriane.foy@gmail.com
16      </div>
17      <div>
18         ImOriane
19      </div>
20    </div>
21
22    <form action="https://formsubmit.co/oriane.foy@gmail.com" method="POST">
23      <h2>Écrivez-moi</h2>
24      <hr>
25
26      <input type="text" name="name" class="infoform" placeholder="Nom" required>
27      <input type="email" name="email" class="infoform" placeholder="Email" required>
28      <textarea name="message" class="infoform" placeholder="Votre message" required></textarea>
29
30      <input type="hidden" name="_captcha" value="false">
31      <input type="hidden" name="_next" value="https://imoriane.github.io/PortfolioPro/contact.html">
32
33      <input type="submit" value="Envoyer">
34    </form>
35
36
37  </div>
38 </main>

```

Le formulaire présent sur la page `contact.html` permet d'envoyer un message directement à l'adresse mail du propriétaire du site, sans nécessiter de traitement côté serveur.

L'envoi est géré via le service [FormSubmit](#), qui agit comme un intermédiaire entre le formulaire HTML et la boîte mail de destination.

Structure du formulaire :



```
1 <form action="https://formsubmit.co/oriane.foy@gmail.com" method="POST">
2   <h2>Écrivez-moi</h2>
3   <hr>
4
5   <input type="text" name="name" class="infoform" placeholder="Nom" required>
6   <input type="email" name="email" class="infoform" placeholder="Email" required>
7   <textarea name="message" class="infoform" placeholder="Votre message" required></textarea>
8
9   <input type="hidden" name="_captcha" value="false">
10  <input type="hidden" name="_next" value="https://imoriane.github.io/PortfolioPro/contact.html">
11
12  <input type="submit" value="Envoyer">
13 </form>
```

Div «form» de la page «contact.html»

Fonctionnement :

- L'attribut `action` contient l'URL de FormSubmit suivie de l'adresse e-mail de réception.
- Les champs `name`, `email` et `message` recueillent les informations saisies par l'utilisateur.
- Le champ caché `_captcha` désactive la vérification automatique de type *captcha* proposée par le service.
- Le champ caché `_next` redirige l'utilisateur vers la page contact après l'envoi du message.
- L'envoi du formulaire est effectué via la méthode POST.

Ainsi, aucun script PHP n'est nécessaire, et le message est directement transmis par FormSubmit au destinataire spécifié.

4.2. Mise en forme CSS.

Le design du site repose sur plusieurs feuilles de style afin de séparer les différentes parties du projet et de faciliter la maintenance du code :

- style.css : fichier principal contenant les règles globales du site (marges, polices, couleurs, positionnement des grandes sections).
- cssIndex.css : feuille de style dédiée au fond de la page d'accueil, gérant notamment les animations du fond et ce dernier.
- cssPages.css : Gère principalement l'animation de transition du fond (effet de "fade out") entre la page d'accueil et les autres pages du site.

Cette animation permet de rendre la navigation plus fluide : pendant la transition, l'image de fond de l'accueil reste visible sur les autres pages le temps que le fondu s'effectue.

Sur la feuille de style principale (style.css), les propriétés CSS sont organisées de manière structurée afin de faciliter la lecture et la maintenance du code.

Les règles sont classées par catégories, chacune correspondant à un type de mise en forme ou à une partie du site :

- Catégorie "Animations" : regroupe l'ensemble des animations utilisées
- Catégorie "Classique" : contient les styles de base, triés par page (accueil, à propos, projets, contact).
- Catégorie "Responsive" : rassemble les règles d'adaptation mobile, également triées par page, pour garantir une lecture fluide sur tous les écrans.

Cette organisation permet d'avoir une feuille de style claire, hiérarchisée et facile à maintenir, tout en améliorant la lisibilité du code.

Exemple animation :

```
1  @keyframes slide_gauche {
2      0% { /* L'objet appelé par l'animation commence invisible et plus loin sur l'axe X que son origine. */
3          transform: translateX(500px);
4          opacity: 0;
5      }
6      17% { /*Transition: commence à rendre l'objet visible. La propriété translateX continue de bouger jusqu'à atteindre
7              automatiquement 0px à 100%.*/
8          opacity: 0.2;
9      }
10     24% {
11         opacity: 0.4;
12     }
13     46% {
14         opacity: 0.6;
15     }
16     100% { /*État final de l'animation : l'objet est désormais à sa position prédéfinie sur l'axe X et complètement visible.*/
17         transform: translateX(0px);
18         opacity: 1;
19     }
20 }
```

Animation slide_gauch

Exemple classique :

```
1  /* Contenu page d'accueil */
2  .content {
3    padding: 9% 0 0 5%;
4  }
5  .hero {
6    width: 40%;
7    text-align: center;
8  }
9  .hero h1 {
10   font-family: "Bebas Neue", sans-serif;
11   font-weight: 400;
12   font-size: 130px;
13   color: var(--couleur_principale);
14   opacity: 0;
15   -webkit-animation: 1s slide_gauche 1s forwards;
16   -moz-animation: 1s slide_gauche 1s forwards;
17   -o-animation: 1s slide_gauche 1s forwards;
18   animation: 1s slide_gauche 1s forwards; /* appel de l'animation declarer en dessous (les 3 lignes au dessus sont juste utilisées sur les anciens navigateurs)*/
19 }
20
21 .hero h2 {
22   width: 140px;
23   font-family: "Pixelify Sans", sans-serif;
24   font-size: 40px;
25   color: var(--couleur_secondaire);
26   overflow: hidden; /
27   border-right: 0.15em solid orange; /* The typewriter cursor */
28   white-space: nowrap; /* Keeps the content on a single line */
29   margin: 0 auto; /* Gives that scrolling effect as the typing happens */
30   letter-spacing: 0.1em; /* Adjust as needed */
31   animation: typing 1s steps(30, end), blink-caret 0.75s step-end infinite;
32 }
33
34 .hero h3 {
35   font-family: "Poppins", sans-serif;
36   font-weight: 400;
37   font-size: 50px;
38   color: var(--couleur_secondaire);
39   opacity: 0;
40   animation: 1s apparition 1s forwards;
41 }
42
43 /*****
```

Code de la page principale

Exemple responsive :

```
1  /* ===== MOBILE===== */
2
3  /* ===== INDEX ===== */
4
5  @media screen and (max-width: 1024px) {
6    .hero h1 {
7      font-size: 90px;
8    }
9    .hero h2 {
10     font-size: 35px;
11   }
12   .hero h3 {
13     font-size: 40px;
14   }
15 }
16
```


Code responsive page principale

4.3. JavaScript

Le site utilise JavaScript pour la gestion de certaines fonctionnalités interactives :

- La transition du fond d'écran de la page d'accueil (passage d'une image de fond vers un fond uni pour améliorer la lisibilité, puis retour à l'image initiale lors du retour sur la page).
- L'animation de particules (via la bibliothèque Particles.js), affichées sous forme de constellations sur la page d'accueil.


Fonctions pour la transition du fond :



```
1 function urlPrevious() {  
2     var p = sessionStorage.getItem('url_prev');  
3     p = p ? p : '';  
4     sessionStorage.setItem('url_prev', location.href);  
5     return p;  
6 }
```

Code de la fonction urlPrevious

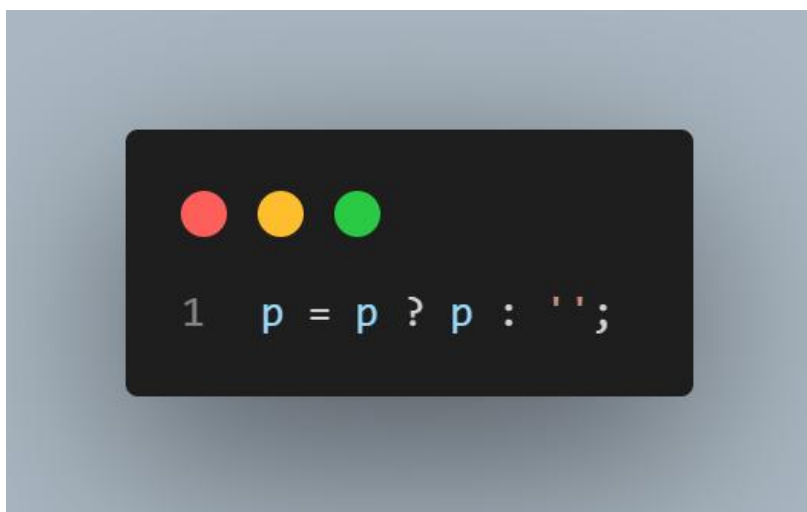
La fonction urlPrevious() ne prend aucun argument et renvoie une variable p correspondant à une URL.



```
1 var p = sessionStorage.getItem('url_prev');
```

Partie du code de la fonction urlPrevious

La variable `p` reçoit la valeur de l'URL de la dernière page visitée (exemple : <file:///M:/ProjetsInformatiques/PortfolioPro/index.html>).



Partie du code de la fonction `urlPrevious`


Si `sessionStorage.getItem` ne contient aucune valeur (par exemple lors de la première visite sur le site), `p` est alors remplacée par une chaîne de caractères vide.



Partie du code de la fonction `urlPrevious`

`sessionStorage` enregistre l'URL de la page actuelle, ce qui permet d'identifier ultérieurement la page précédemment visitée.

Ainsi, la fonction `urlPrevious()` renvoie soit l'URL de la page précédente, soit une chaîne vide s'il s'agit de la première page visitée.



```

1  function fadeBackground (pageActuel) {
2      pagePrecedente=urlPrevious();
3      if (pagePrecedente.includes("index.html")) {
4          if (pageActuel !== "index.html") {
5              document.querySelector("#bg").classList.add("bgfadeOut");
6          }
7      } else {
8          if (pageActuel === "index.html") {
9              document.querySelector("#bg").classList.add("bgfadeIn");
10         }
11     }
12 }


```

Code de la fonction

La fonction `fadeBackground(pageActuel)` prend un argument correspondant à l'URL simplifiée de la page actuellement affichée.

Elle utilise deux variables :

- `pageActuel`, passée en argument via l'appel de la fonction à la fin de chaque page.
- `pagePrecedente`, obtenue grâce à la fonction `urlPrevious()`.



```

1  <script>
2      fadeBackground("index.html");
3  </script>

```

Code d'appel de la fonction `fadeBackground`

CAS 1 :

```
1  if (pagePrecedente.includes("index.html")) {  
2      if (pageActuel !== "index.html") {  
3          document.querySelector("#bg").classList.add("bgfadeOut");  
4      }  
}
```

Partie du code de fadeBackground

Si pagePrecedente contient la chaîne de caractères "index.html" (indiquant que la page précédente était la page d'accueil) et que pageActuel est différente de "index.html", une transition du fond est appliquée.

La div ayant pour identifiant #bg est sélectionnée par document.querySelector("#bg"). Une nouvelle classe CSS nommée bgfadeOut est ajoutée à cet élément. Le CSS associé à cette classe déclenche une animation de fondu permettant la disparition progressive de l'image de fond au profit d'un fond uni défini dans le style global du document.

```
1  <div class="bg" id="bg"></div>
```

Classe «bg» tirée de la structure HTML du site


```

1  .bg {
2    background-image: url("../documents/fond.jpg");
3    position: fixed;
4    width: 100%;
5    min-height: 100%;
6    opacity: 0;
7    z-index: -1;
8    background-size: cover;
9    background-position: center;
10   background-attachment: fixed; /* L'img background ne bouge pas au scroll/zomm... */
11 }

```

CSS de la classe « bg » tiré de cssPage.css

L'élément <div> possédant la classe bg se voit ajouter dynamiquement une classe supplémentaire, bgfadeOut, via l'instruction suivante :

```

1  document.querySelector("#bg").classList.add("bgfadeOut");

```

Partie du code de la fonction fadeBackground

La classe bgfadeOut est associée à une animation CSS permettant de faire disparaître progressivement l'image de fond de l'élément .bg. L'image de fond est définie dans le fichier cssPage.css, tandis que la couleur de fond unie est spécifiée dans les styles de la balise <html>.



```
1  html {  
2    background-color: #0d0f2f;  
3  }
```

CSS de la balise « html »



```
1  .bgfadeOut {  
2    -webkit-animation: bgfadeOut 1s forwards;  
3    -moz-animation: bgfadeOut 1s forwards;  
4    -o-animation: bgfadeOut 1s forwards;  
5    animation: bgfadeOut 1s forwards;  
6  }
```

CSS de la classe « bgfadeOut »



```
1  @keyframes bgfadeOut {  
2    0% {  
3      opacity: 1;  
4    }  
5    100% {  
6      opacity: 0;  
7    }  
8  }
```

Animation bgfadeOut

Dans le fichier `cssPage.css`, la classe `.bg` possède une opacité définie à 0 par défaut sur toutes les pages autres que `index.html`, empêchant ainsi l'affichage de l'image de fond. L'ajout de la classe `bgfadeOut` déclenche une animation qui rétablit temporairement l'opacité à 1, rendant l'image de fond visible de manière transitoire.

CAS 2 :

Dans le cas où le scénario précédent (Cas 1) ne s'applique pas — c'est-à-dire lorsque la page précédente n'est pas `index.html`, ou qu'elle correspond également à la page actuelle —, la logique suivante est appliquée :

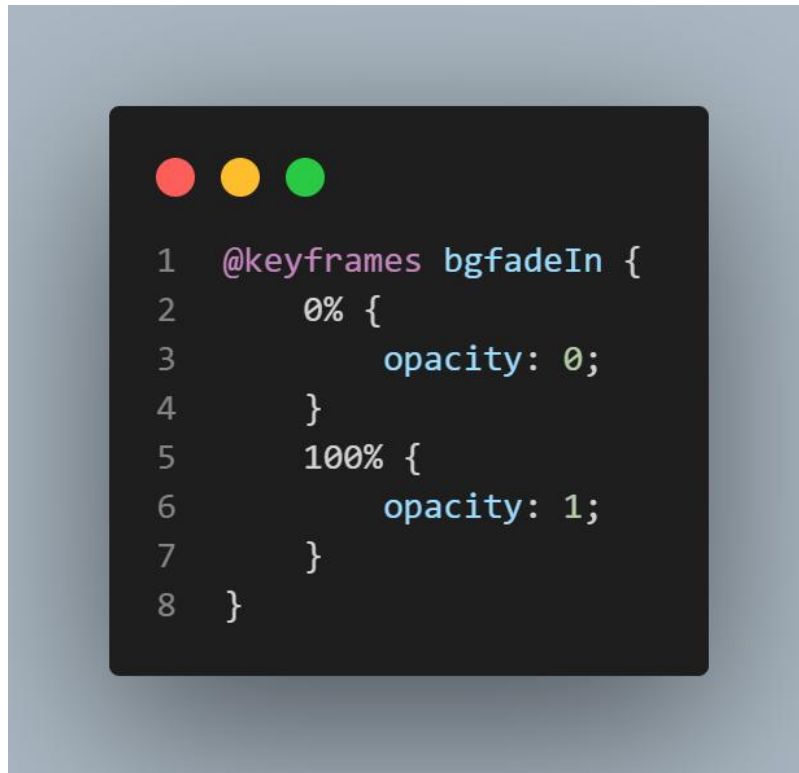
Si la page actuelle (`pageActuel`) correspond à `index.html`, cela indique une navigation depuis une autre page. Dans ce cas, la fonction sélectionne l'élément ayant pour identifiant `bg` et lui ajoute la classe CSS `bgfadeIn`, afin d'assurer la réapparition progressive de l'image de fond.

```
1 } else {  
2     if (pageActuel === "index.html") {  
3         document.querySelector("#bg").classList.add("bgfadeIn");  
4     }
```

Partie du code de la fonction `fadeBackground`

```
1 .bgfadeIn {  
2     -webkit-animation: bgfadeIn 1s forwards;  
3     -moz-animation: bgfadeIn 1s forwards;  
4     -o-animation: bgfadeIn 1s forwards;  
5     animation: bgfadeIn 1s forwards;  
6 }  
7
```

CSS de la classe « `bgfadeIn` »



Animation « bgFadeIn »

Si la page actuelle correspond à index.html et qu'il s'agit de la première page visitée (valeur de pagePrecedente égale à null ou chaîne vide), la classe bgfadeIn est ajoutée à l'élément bg. En revanche, si la première page chargée est différente de index.html, la variable pagePrecedente reste vide mais aucune condition n'est remplie ; le fond conserve alors son état par défaut, c'est-à-dire un fond uni.

Fonction pour l'animation de particule :

```
1 document.addEventListener('DOMContentLoaded', () => {
2   particlesJS('particles-js', {
3     particles: {
4       number: { value: 80, density: { enable: true, value_area: 800 } },
5       color: { value: 'ffffff' },
6       shape: { type: 'circle' },
7       opacity: { value: 0.5 },
8       size: { value: 5, random: true },
9       line_linked: { enable: true, distance: 150, color: 'ffffff', opacity: 0.4, width: 1 },
10      move: { enable: true, speed: 6, direction: 'none', out_mode: 'out' }
11    },
12    interactivity: {
13      detect_on: 'canvas',
14      events: { onhover: { enable: true, mode: 'repulse' }, onclick: { enable: true, mode: 'push' } },
15      modes: { grab: { distance: 400, line_linked: { opacity: 1 } }, repulse: { distance: 200 } }
16    },
17    retina_detect: true
18  });
19 });
```

Fonction javascript pour particleJS

```
1 #particles-js {
2   position: fixed;
3   top: 0;
4   left: 0;
5   width: 100%;
6   height: 100%;
7   z-index: 0;
8   pointer-events: none;
9 }
10
```

Classe CSS pour particles-js tirée de cssIndex.css

```
1 <script src="script/particle.js"></script>
2 <script src="https://cdn.jsdelivr.net/npm/particles.js@2.0.0/particles.min.js"></script>
3 <script defer src="scripts/particles-config.js"></script>
```

Appel des différents scripts pour particles.js dans le head de « index.html »

Utilisations de la bibliothèque JavaScript particles.js

Source: <https://github.com/VincentGarreau/particles.js>

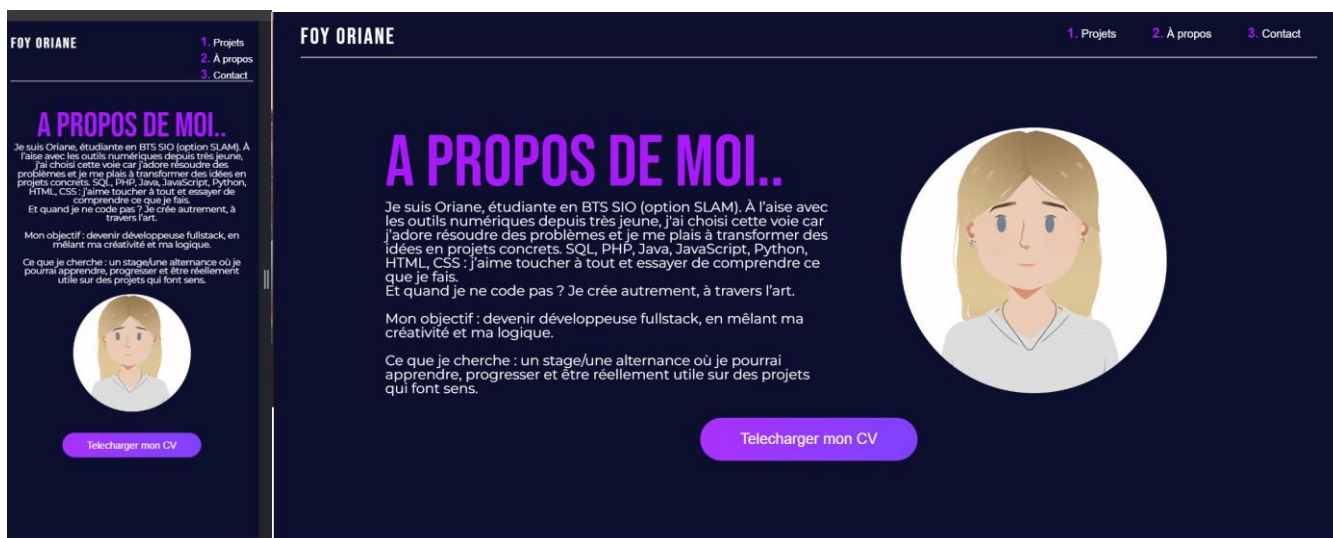
4.4 Responsive.

Le site est conçu pour garantir une expérience utilisateur cohérente et adaptée sur l'ensemble des types d'écrans : ordinateurs, tablettes et smartphones.

Pour cela, plusieurs techniques ont été mises en place :

- Media queries : permettent d'ajuster le style des éléments selon la largeur de l'écran.
- Flexbox et Grid : assurent un positionnement flexible et une mise en page fluide.
- Images et contenus adaptatifs : les images sont redimensionnées automatiquement pour éviter qu'elles débordent ou se coupent.

Exemple :



Page « à propos »

```

1  @media screen and (max-width: 768px) {
2    .presentation {
3      flex-direction: column;
4      text-align: center;
5      padding: 0 20px;
6    }
7
8    .present h1 {
9      font-size: 70px;
10   }
11   .col{
12     padding-left: 0px;
13   }
14   .present h2 {
15     font-size: 18px;
16   }
17
18   .avatar img {
19     width: 50%;
20     margin-top: 20px;
21   }
22
23   .myButton {
24     font-size: 20px;
25     padding: 15px 50px;
26   }
27 }
28 }

```

CSS pour la page « à propos » pour les écrans de moins de 768 px

```

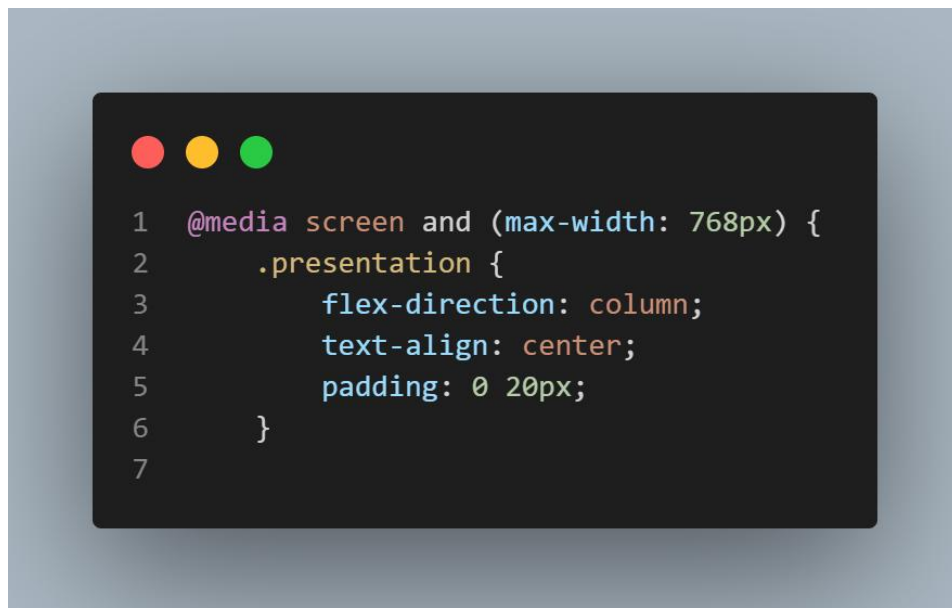
1  nav ul {
2    display: flex;
3    flex-direction: column;
4    gap: 8px;
5  }

```

CSS pour les différents onglets de la page de navigation pour les écrans de moins de 768px

Cette règle Media Query applique des styles spécifiques lorsque la largeur de la fenêtre d’affichage est inférieure ou égale à 768 px.

Dans cet exemple :



CSS pour la classe « presentation » pour les écrans de moins de 768px

L’élément nav ul est réorganisé en colonne afin d’adapter l’affichage de la barre de navigation sur les écrans de petite taille (liens « À propos », « Contact », etc.).

La classe .presentation utilise la propriété flex-direction: column; issue du module Flexbox, ce qui permet de réorganiser les éléments verticalement lorsque l’espace horizontal devient insuffisant.

5. Tests et validation

5.1. Tests de compatibilité.

Navigateurs testés :

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Brave

Résultats :

- L'ensemble des pages s'affiche correctement et les mises en page restent cohérentes.
- Les animations et les transitions fonctionnent sur tous les navigateurs testés.
- Les liens de navigation sont opérationnels
- La responsive se comporte correctement sur petits écrans.

5.2 Tests de responsive design.

Ecrans testés :

- iPhone SE
- iPhone XR
- iPhone 12 Pro
- iPhone 14 Pro Max
- Pixel 7
- Samsung Galaxy S8+
- Samsung Galaxy S20 Ultra
- iPad Mini
- iPad Air
- iPad Pro
- Surface Pro 7
- Surface Duo
- Galaxy Z Fold 5
- Asus Zenbook Fold
- Samsung Galaxy A51/71
- Nest Hub
- Nest Hub Max
- MSI 23.6" LED - MAG 242C

Résultats :

- L'ensemble des pages s'affiche correctement et les mises en page restent cohérentes.
- Les liens de navigation sont opérationnels

6.Documentation utilisateur

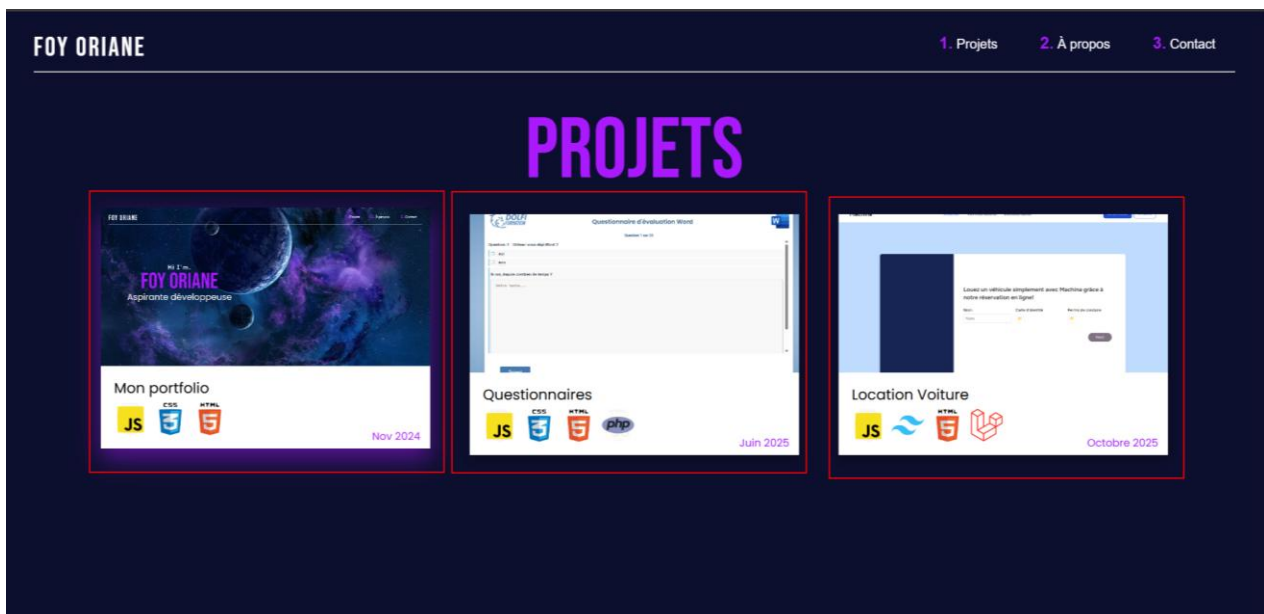
6.1. Navigation

La navigation au sein du site s'effectue à l'aide d'éléments graphiques intégrés à l'interface :

- Les pages principales sont accessibles via la **barre de navigation** : chaque lien porte le nom de la page correspondante.
- Les pages dédiées aux **projets** sont accessibles depuis la section “Projets”. L'utilisateur sélectionne le projet souhaité en cliquant sur le bloc correspondant.



Page « Index »



Page « Projets »

6.2 M'envoyer un message.

Pour m'envoyer un message directement depuis mon portfolio, il faut se placer sur la page "contact", ensuite, il faut remplir les informations requises dans le formulaire "Ecrivez-moi" à droite de l'écran sur ordinateur ou en bas sur téléphone.

Une fois les informations rentrées, il suffit de cliquer sur envoyer pour me transmettre votre message ainsi que votre nom et mail.

The screenshot shows the 'CONTACT' page of the 'FOY ORIANE' portfolio. The page has a dark blue background. At the top left is the logo 'FOY ORIANE'. At the top right are navigation links: '1. Projets', '2. À propos', and '3. Contact' (which is underlined). The main heading 'CONTACT' is in large, bold, light blue letters. On the left side, there are four social media icons with corresponding text: LinkedIn 'Oriane Foy', Phone '06 63 81 01 24', Email 'oriane.foy@gmail.com', and GitHub 'ImOriane'. On the right side, there is a white-bordered box containing the form 'Ecrivez moi'. The form has three input fields: 'Nom', 'Email', and 'Texte'. Below these fields is a red 'Envoyer' button.

Page « Contact »

6.3 Télécharger mon CV

Le téléchargement du CV s'effectue depuis la page "À propos".

Un bouton intitulé "Télécharger mon CV", situé en bas de la page, permet d'obtenir une copie du document au format PDF.

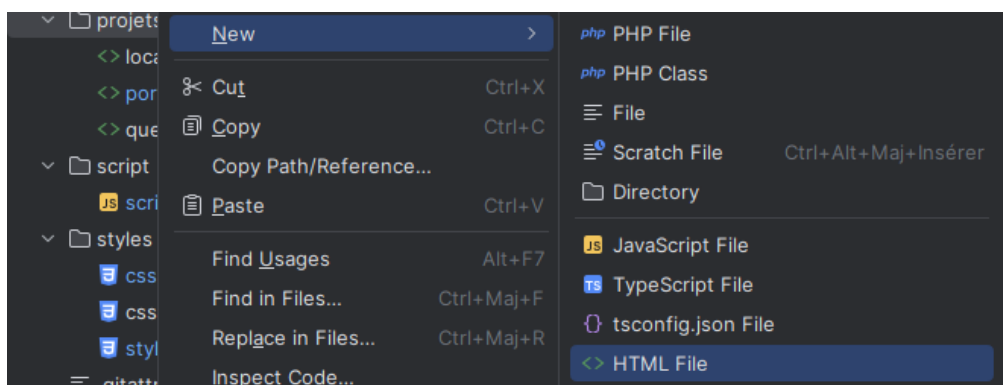
The screenshot shows the 'À PROPOS DE MOI..' page of the 'FOY ORIANE' portfolio. The page has a dark blue background. At the top left is the logo 'FOY ORIANE'. At the top right are navigation links: '1. Projets', '2. À propos' (which is underlined), and '3. Contact'. The main heading 'À PROPOS DE MOI..' is in large, bold, light blue letters. Below the heading, there is a paragraph of text in white: 'Je suis Oriane, étudiante en BTS SIO (option SLAM). À l'aise avec les outils numériques depuis très jeune, j'ai choisi cette voie car j'adore résoudre des problèmes et je me plais à transformer des idées en projets concrets. SQL, PHP, Java, JavaScript, Python, HTML, CSS : j'aime toucher à tout et essayer de comprendre ce que je fais. Et quand je ne code pas ? Je crée autrement, à travers l'art. Mon objectif : devenir développeuse fullstack, en mêlant ma créativité et ma logique. Ce que je cherche : un stage/une alternance où je pourrai apprendre, progresser et être réellement utile sur des projets qui font sens.' To the right of the text is a circular profile picture of a woman with blonde hair. At the bottom center, there is a red button labeled 'Télécharger mon CV'.

Page « A propos »

6.3. Mise à jour du contenu.

Ajout d'une nouvelle page :

- Créer un fichier .html dans le répertoire principal pour une page générale, ou dans le dossier projets pour une page de projet.
- Reproduire la structure HTML de base (voir section 4.1 — Structure HTML) et adapter le contenu de la balise <main> selon les besoins.
- Ajouter toutes les ressources utilisées (images, documents, etc.) dans le sous-dossier documents.



Manipulation pour créer un fichier html sur PhpStorm

Ajout d'une nouvelle compétence :

Pour ajouter une nouvelle compétence, il est nécessaire de modifier le fichier competences.html.

Procédure :

1. Ouvrir le fichier competences.html dans un éditeur de code.
2. Repérer la section <div class="carousel-track"> : c'est à cet emplacement que sont répertoriées toutes les compétences sous forme de cartes.
3. Ajouter un nouveau bloc suivant le modèle ci-dessous :



Div «carousel-card» de la page competences.html

4. Remplacer les valeurs :
nomImage.png : nom du fichier image correspondant à la compétence.
Nom de la compétence : nom de la technologie, outil ou langage ajouté.
Catégorie : type de compétence (ex. : *Langage, Framework, Outil, OS...*).
5. Placer l'image correspondante dans le répertoire documents/.
Le nom du fichier doit correspondre à celui indiqué dans l'attribut src.
6. Enregistrer le fichier puis actualiser la page Mes compétences pour vérifier l'ajout.

7.Conclusion

7.1 Bilan.

La création de ce portfolio m'a permis de m'exercer aux langages HTML, CSS et JavaScript, ainsi qu'à comprendre comment structurer son code.

La phase de conception sur Figma a été d'un grand secours pour réfléchir sur l'interface et faciliter le développement, grâce aux maquettes précises de chaque page.

L'organisation du CSS et la gestion des animations m'ont permis de m'améliorer et d'avoir un site fluide et lisible, tandis que la partie du responsive m'a amenée à comprendre davantage comment gérer l'adaptation du site sur différents périphériques.

Ce projet est surtout un outil professionnel concret, me permettant de présenter mes compétences et réalisations à des potentiels recruteurs.

7.2 Améliorations possibles.

Plusieurs améliorations pourraient être envisagées pour enrichir le portfolio et le rendre encore plus interactif et ergonomique :

- Système de filtrage des projets : il pourrait être intéressant d'ajouter des filtres permettant de trier les projets par langage ou technologie utilisée (HTML, CSS, JavaScript, PHP, etc.). Quand j'aurai plus de projets à présenter, il permettrait aux recruteurs de cibler directement ce qui les intéresseraient potentiellement.
- Optimisation mobile avancée : bien que le site soit responsive, certains ajustements supplémentaires pourraient améliorer encore l'expérience sur smartphones ou tablette.