Dossier Projet - Développeur Web Full Stack

Introduction

Ce dossier projet présente en détail les compétences validées dans le cadre de ma formation de développeur web full stack à La Plateforme, en s'appuyant strictement sur les éléments du dossier professionnel et illustrées au travers des différents projets réalisés. Je suis MZE Abdoul-Hachim j'ai découvert l'informatique grâce à ma passion pour les jeux vidéo et ma curiosité qui m'a toujours poussé a cherché à comprendre ce qui m'entourent mais aussi les possibilité de projets réalisable grâce à l'informatique cela vient surtout de mon quotidien lors ce que j'étais enfant et que je me demandais mais comment ont -il fait pour les dessin animé bouge de cette façon et comment se fait -il qu'il parle et que je peux les comprendre aussi loin que je me souvienne c'est là mes premières expériences avec le numérique.

1. Développement Frontend

Qu'est-ce que le développement frontend ?

C’est le travail qui consiste à créer l’interface utilisateur d’un site web. Cela inclut :

- Le design (couleurs, boutons, menus…)
- L’organisation des éléments à l’écran
- Les animations et les interactions (clics, survols, transitions…)

Technologies utilisées (avec des exemples simples)

1. **HTML (HyperText Markup Langage).**
   C’est le squelette du site. Il sert à organiser le contenu.
   Exemple : un paragraphe et un titre ou une image est défini en HTML.

```
<h1>Bienvenue sur mon site</h1>
```

   L’image ci-dessus est une image contenant un titre HTML, il est reconnaissable grâce à la balise <h1></h1>

2. **CSS (Cascading style Sheets).**
   Sert à rendre le site joli. On s’en sert pour plus les couleurs, les tailles, les Espacements, etc.
   Exemple : faire un bouton bleu avec du texte blanc.

```
button {
  background-color: blue;
  color: white;
}
```

3. **JavaScript.**
   C’est ce qui permet de rendre la page interactive.
   Exemple : quand on clique sur un bouton, une image change ou une alerte s’affiche.

```
alert("Bienvenue !");
```

4. **Frameworks ou bibliothèques (comme React, Vue.js Bootstrap).**
   Ils facilitent et accélèrent le travail. Exemple : avec React, on peut créer des composants réutilisables comme une carte produit ou une barre de navigation.

1.1 Configuration de l'Environnement de Développement

Configuration de l'environnement de développement parle de mettre en place tout ce qu'il faut pour pouvoir créer un site ou une application sur ordinateur. Voici les concepts clés expliqués simplement :

1. **Environnement de développement.**
   C'est tout ce que tu installes sur ton ordinateur pour pouvoir ***écrire***, ***tester*** et ***faire fonctionner*** ton code.
   Exemple : Pour construire une maison, tu as besoin d'outils comme un marteau ou une scie. Pour coder (ex : Visual Studio Code, Cursor, Windsurf, Sublime Text, etc.).

2. **Installation des outils**.
   Il faut installer plusieurs outils :
   - Un éditeur de code (comme Visual Studio Code, Sublime Texte, Cursor, Windsurf, etc.).
   - Un **navigateur** (comme Chrome ou Firefox pour tester le site).
   - Des serveurs locaux ou outils comme LARAGON, XAMPP, WAMP, ou Docker pour faire tourner ton code comme s'il était en ligne.

3. **Gestion des dépendances.**
   Quand tu utilises des bibliothèques ou des outils exterieurs (comme React, Bootstrap ou Symfony), tu dois les installer et les gérer avec des outils comme :
   - Npm pour JavaScript
   - Composer pour PHP

4. **Comfiguration des fichiers**
   Tu dois parfois modifier ou créer des fichiers pour dire à ton projet comme il doit fonctionner :
   - .env pour les variables d'environnement (ex : mot de passe de la base de données)
   - Package.json ou composer.json pour lister les dépendances

   Exemple : Dans un fichier .env, tu écris : DB_PASSWORD=motdepasse. Cela permet de changer le mot de passe de la base de données sans toucher au code principal.

1.1.1 Installation et configuration de Node.js

Installation de node.js

Node.js est un outil qui permet d'exécuter du JavaScript en dehors du navigateur, directement sur ton ordinateur. C'est très utile pour créer des applications web côté serveur (backend).

Exemple d'installation :

- Aller sur https://nodejs.org/en
- Choisir une version à installer (souvent la version "LTS" = qui est stable).
- Tu installes comme un logiciel classique (suivant, suivant...).

Exemple concret : Imagine que tu veux créer un site où les gens peuvent publier des annonces. Tu auras besoin d'un serveur pour gérer ce serveur avec JavaScript.

Configuration de Node.js.

Une fois Node.js installé, tu peux :

- Vérifier qu'il fonctionne avec la commande node –v (ça affiche la version).
- Utiliser npm (le gestionnaire de paquets de Node.js) pour installer des modules (petits bouts de code utiles).

  Exemple de module :

  ```
  npm install express
  ```

  Cela installe Express, un outil populaire pour créer des serveurs web rapidement avec Node.js.

  Exemple : Si tu veux que ton site affiche une page d'acceuil quand quelqu'un Visite http://localhost:3000, Express peut t'aider à le faire très simplement.

Mise en place du framework React.js via Vite.

React.js est un outil (framework) utilisé pour créer des interfaces utilisateurs (comme les page web que tu vois dans ton navigateur). C'est très populaire pour créer des sites web interactifs.

Exemple : Imagine que tu construis un site comme YouTube. React t'aide à créer les différentes parties de la page, comme la barre de recherche, la liste des vidéos ou les commentaires, et à les faire réagir quand l'utilisateur clique ou tape quelque chose.

Vite, est un outil qui aide à démarrer plus rapidement un projet React. Il s'occupe de la configuration technique et permet que ton site se recharge instantanément pendant que tu travailles dessus.

Exemple :
Avant, on utilisait un outil plus lent (comme Webpack) pour démarrer un projet. Avec Vite c'est comme avoir une voiture de sport : tu démarres vite, et les changements s'affichent plus rapidement quand tu modifies ton code.

Configuration de Laragon pour le développement local.

1. **Laragon, c'est quoi ?**
   Laragon est un outil tout-en-un qui installe facilement ce qu'il faut pour faire tourner un site web en local (sur ton ordinateur). Il contient souvent :
   - Apache ou Nginx (pour simuler un serveur),
   - MySQL (pour gérer les bases de données),
   - PHP (le langage utilisé par beaucoup de site web),
   - Et parfois d'autres outils comme Node.js ou Composer.

Exemple : Tu veux créer un site en PHP avec une base de données ? Avec Laragon, tu lances l'application, et tout est prêt pour commencer à coder, comme si ton PC était un serveur web.

2. **Développement local**
   Cela veut dire tu crées et test ton site web sur ton propre ordinateur, avant de le mettre en ligne.
   Exemple : Tu construis une page d'inscription. Tu peux la tester avec Laragon pour voir si elle fonctionne bien, sans avoir besoin d'un hébergement en ligne.

3. **Configuration de Laragon**
   Cela signifie que tu règles certains paramètres pour que Laragon fonctionne comme tu veux. Par exemple :

- Choisir la version de PHP,
- Créer un "virtual host" (c'est comme une adresse personnalisée, du style mon-projet.test, au lieu de localhost),
- Définir le dossier où tu vas stocker tes projets.

Exemple : Tu veux que ton projet soit accessible sur http://monblog.test. Tu configures ça dans Laragon en quelques clics.

Utilisation des gestionnaires de paquets (npm, composer).

1. **C'est quoi un gestionnaire de paquets ?**

Un gestionnaire de paquets est un outil qui télécharge, installe et met à jour automatiquement des morceaux de code (appelé paquets ou bibliotheques) créés par d'autres développeurs. Ces paquets permettent de gagner du temps en évitant de tout créer soi-même.

2. **Npm : pour JavaScript/Node.js**
   - Npm signifie Node Package Manager.
   - Il est utilisé dans les projets JavaScript ou Node.js.
   - Exemple : Si tu fais web avec React, tu peux taper cette commande :

     ```
     npm install react
     ```

     Cela télécharge la bibliothèque React dans ton projet.
   - Tu peux aussi installer d'autres outils utiles, comme :

     ```
     npm install express
     ```

     Pour créer un serveur web facilement.

3. **Composer : pour PHP**
   - **Composer** est le gestionnaire de paquets pour les projets PHP.
   - Exemple : Tu veux utiliser un outil comme Symfony (un framework PHP). Tu tapes :

     ```
     composer require symfony/http-foundation
     ```

     Cela ajoute un composant Symfony pour gérer les requêtes HTTP.

4. **Pourquoi c'est important ?**
   - Tu gagnes du temps.
   - Tu utilises du code fiable et mise à jour par la communauté.
   - Tu peux organiser ton projet proprement (chaque paquet est listé dans un fichier, comme package.json pour npm ou composer.json pour Composer).

1.2 Conception d'Interfaces Utilisateur

La conception d'interfaces utilisateur consiste à imaginer et structurer l'apparence ainsi que l'organisation des éléments visibles et interactifs d'un site web ou d'une

application, dans le but d'offrir une expérience fluide, intuitive et accessible à l'utilisateur.

Dans un premier temps, il est essentiel d'analyser les besoins des utilisateurs finaux d'adapter l'interface à leurs attentes et usages. Par exemple, une application de livraison de repas devra permettre une navigation rapide et une prise de commande en quelques clics, ce qui implique une interface simple et efficace.

La création de maquettes (ou *wireframes*) est une étape clé. Elle permet de visualiser la disposition des éléments (boutons, menus, formulaires, etc.) avant le développement.

Ces maquettes servent de base pour les choix ergonomique et esthétiques.

L'ergonomie joue un rôle central : elle garantit une utilisation facile et logique d'interface. Les composants doivent être immédiatement identifiable et accessible, quel que soit le support utilisé (ordinateur, tablette, smartphone).

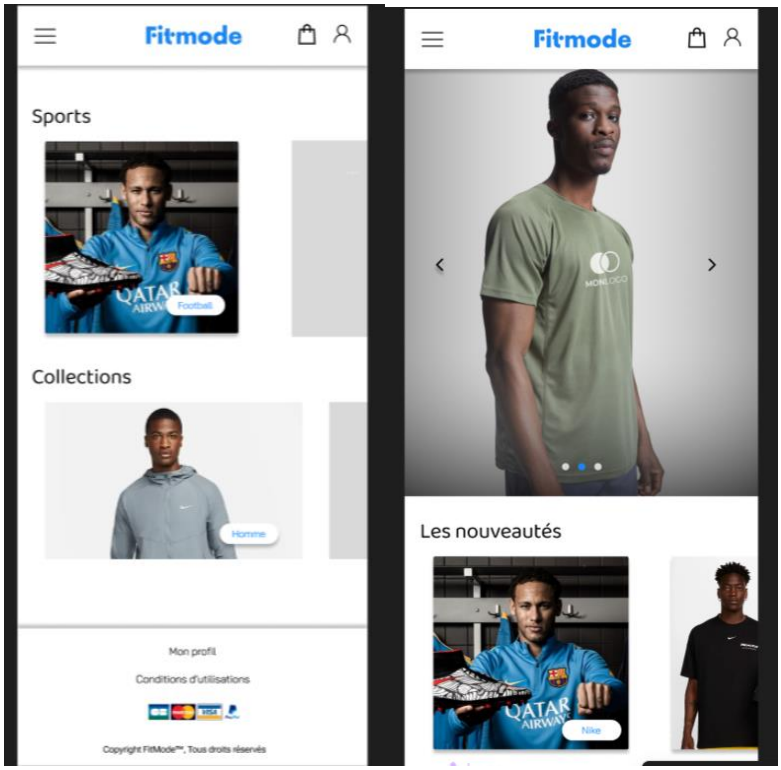
Le design graphique contribue également à l'efficacité de l'interface. Le choix des couleurs, des typographies et des icônes doit rester harmonieux et cohérent avec l'identité visuelle du projet. Une interface destinée à un jeune public utilisera par exemple des couleurs vives et de formes arrondies, tandis qu'un site professionnel privilégiera des tons sobres et une typographie claire.

Enfin, l'accessibilité doit être pris en compte dès la phase de conception. Cela inclut, l'utilisation de compatibilité avec une navigation clavier ou des lecteurs d'écran, afin de garantir l'accès au contenu pour tous les utilisateurs, y compris ceux en situation de handicap.

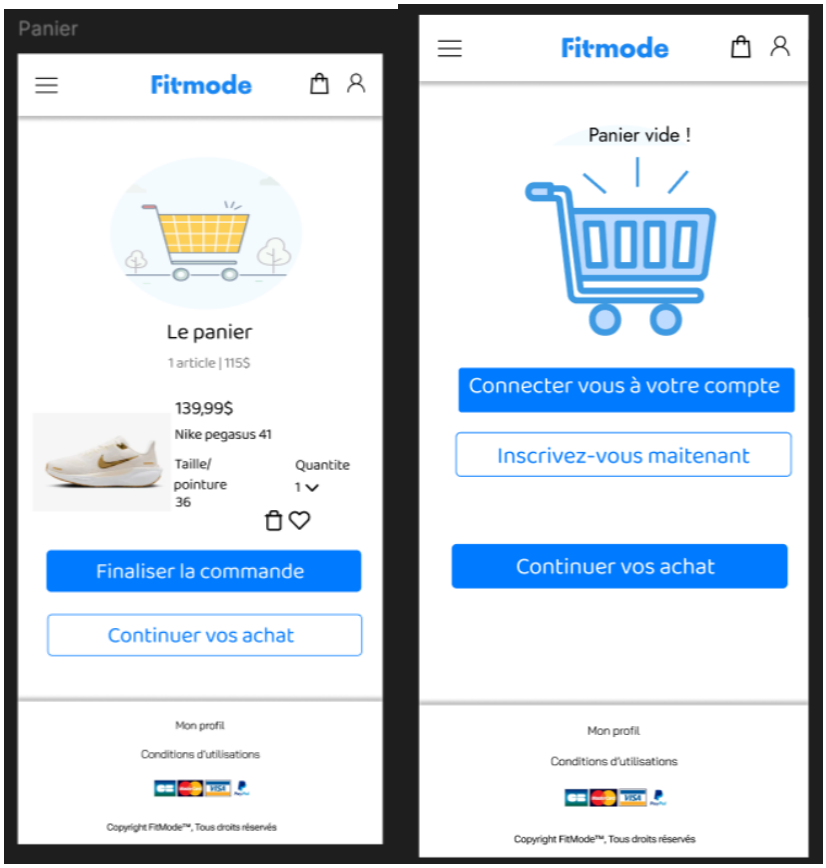
Exemples concrets de conception d'interfaces (illustré ci-dessus) :

Les images ci-après présentent les maquettes réalisées pour un site e-commerce dédié à la vente de vêtements. Elles illustrent la conception des pages clés (accueil, fiche produit, panier), avec un travail spécifique sur l'ergonomie, la navigation mobile, l'harmonie graphique (couleurs, typographie) et l'expérience utilisateur.

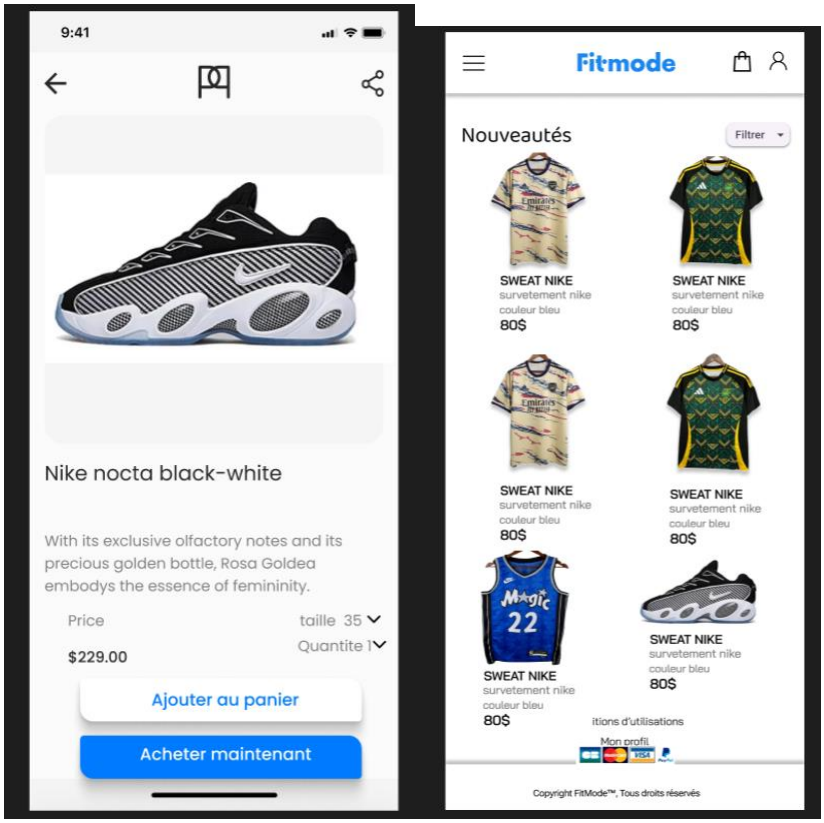
Accueil :



Panier :



Pages produits et détail produits :



1.3 Développement Frontend Statique

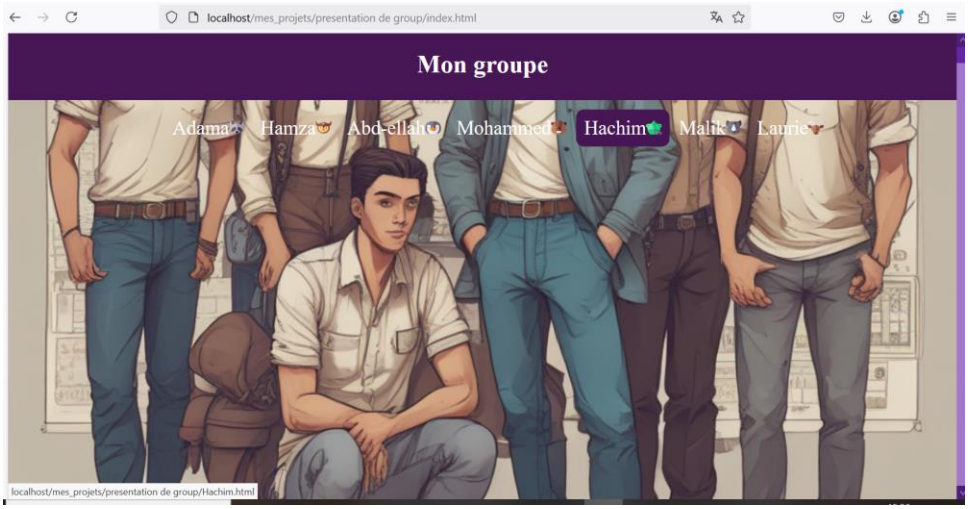
C’est la partie visible d’un site web, faite avec HTML, CSS et parfois un peu de JavaScript basique.

Comme une page d’accueil avec un titre, une image, et du texte fixe.

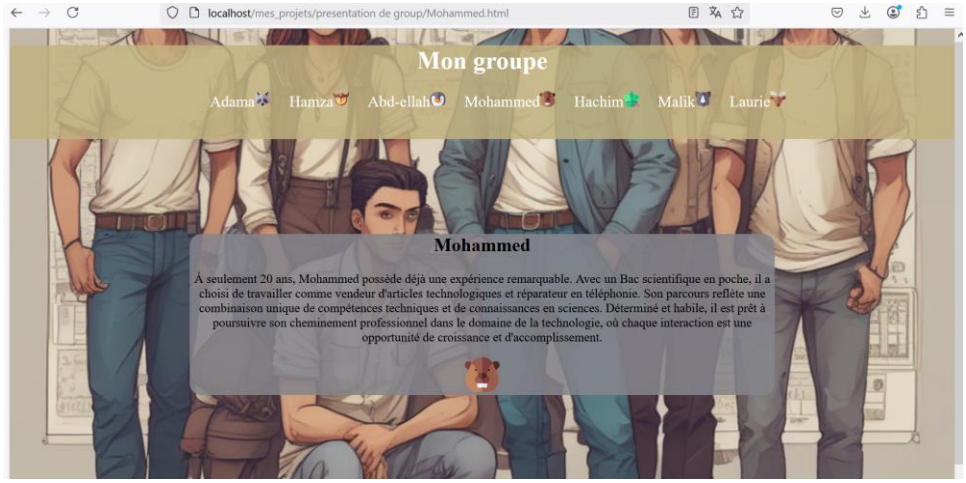
Maîtrise des technologies de base :

- **HTML et CSS.**
  Voici un projet Frontend statique que j’ai réalisé à titre d’exemple pour démontrer ce qu’est un projet frontend statique :

Voici une capture de la page d’accueil du projet statique réalisé en HTML et CSS.



Voici une capture d'écran de l'une des pages de présentation des membres du groupe :



Voici les captures d'écran du code qui démontrent comment les pages ci-dessus ont été réalisées :

- **Page d'accueil**
  HTML :

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>le groupe</title>
    <link rel="stylesheet" href="./assets/CSS/style.css">
</head>
<body>
    <header>
        <h1>Mon groupe</h1>
    </header>
    <div class="groupe">

        <ul>
            <li><a href="./Adama.html">Adama</a><img src="./assets/images/raton-laveur.png" alt=""></li>
            <li><a href="./Hamza.html">Hamza</a><img src="./assets/images/hibou.png" alt=""></li>
            <li><a href="./Abd-ellah.html">Abd-ellah</a><img src="./assets/images/manchot.png" alt=""></li>
            <li><a href="./Mohammed.html">Mohammed</a><img src="./assets/images/castor.png" alt=""></li>
            <li><a href="./Hachim.html">Hachim</a><img src="./assets/images/reptile.png" alt=""></li>
            <li><a href="./Malik.html">Malik</a><img src="./assets/images/rhinoceros.png" alt=""></li>
            <li><a href="./Laurie.html">Laurie</a><img src="./assets/images/cerf.png" alt=""></li>
        </ul>
    </div>

</body>
</html>
```

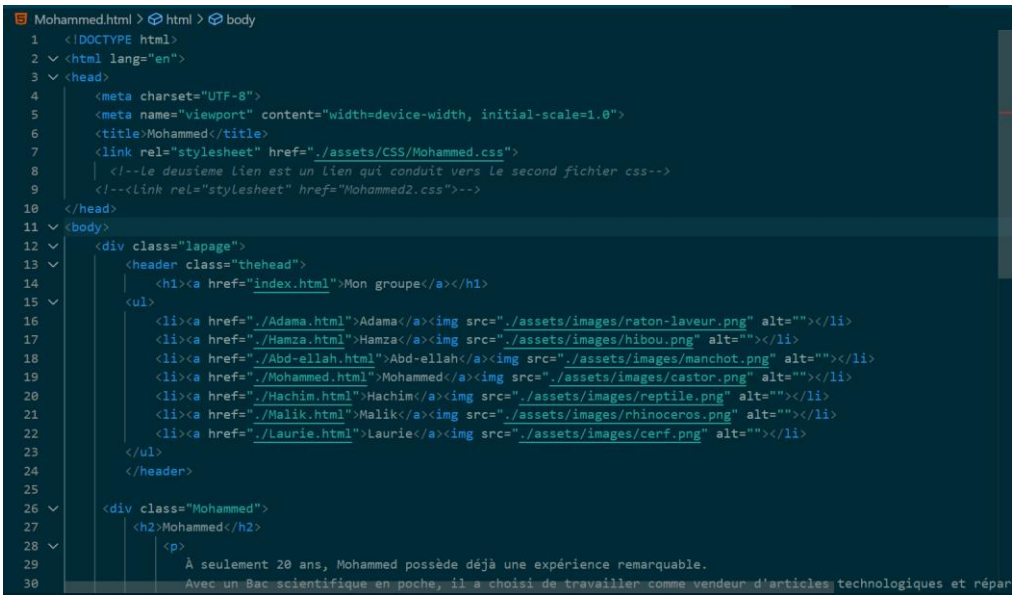
CSS :

```css
.groupe ul li {
    list-style-type: none;
    display: inline;
}
li:hover{
    color: rgb(255, 255, 255);
    background-color: rgb(70, 22, 85);
padding: 10px;
border-radius: 10px;

}
a{
    text-decoration: none;
}
ul li{
    padding: 8px;
    font-size: 25px;
}
h1{
    padding: 20px;
    text-align: center;
    background-color: rgb(70, 22, 85);
    background-size: cover;
    height: 45px;
    color: white;
}
.groupe img{
    width: 20px;
    height: 18px;
}
```

```css
a{
    color: white;
}
body{
    background-image: url("../images/DesignSansTitre.png");
    background-repeat: no-repeat;
    background-size: 100%;
    background-position: center;
    min-height: 100vh;
    margin: 0;
    padding: 0;
}
ul{
color: rgb(255, 255, 255);
text-align: center;
}
body{
    background-color: blueviolet;
}
```

- **Page membre de groupe**

HTML :



```
Mohammed.html > html > body
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mohammed</title>
    <link rel="stylesheet" href="./assets/CSS/Mohammed.css">
    <!--Le deusieme lien est un lien qui conduit vers le second fichier css-->
    <!--<link rel="stylesheet" href="Mohammed2.css">-->
</head>
<body>
    <div class="lapage">
        <header class="thehead">
            <h1><a href="index.html">Mon groupe</a></h1>
        <ul>
            <li><a href="./Adama.html">Adama</a><img src="./assets/images/raton-laveur.png" alt=""></li>
            <li><a href="./Hamza.html">Hamza</a><img src="./assets/images/hibou.png" alt=""></li>
            <li><a href="./Abd-ellah.html">Abd-ellah</a><img src="./assets/images/manchot.png" alt=""></li>
            <li><a href="./Mohammed.html">Mohammed</a><img src="./assets/images/castor.png" alt=""></li>
            <li><a href="./Hachim.html">Hachim</a><img src="./assets/images/reptile.png" alt=""></li>
            <li><a href="./Malik.html">Malik</a><img src="./assets/images/rhinoceros.png" alt=""></li>
            <li><a href="./Laurie.html">Laurie</a><img src="./assets/images/cerf.png" alt=""></li>
        </ul>
        </header>

    <div class="Mohammed">
        <h2>Mohammed</h2>
            <p>
                À seulement 20 ans, Mohammed possède déjà une expérience remarquable.
                Avec un Bac scientifique en poche, il a choisi de travailler comme vendeur d'articles technologiques et répar
```

```
Mohammed.html > html > body
<html lang="en">
<body>
    <div class="lapage">
        <header class="thehead">
        <ul>
            <li><a href="./Hachim.html">Hachim</a><img src="./assets/images/reptile.png" alt=""></li>
            <li><a href="./Malik.html">Malik</a><img src="./assets/images/rhinoceros.png" alt=""></li>
            <li><a href="./Laurie.html">Laurie</a><img src="./assets/images/cerf.png" alt=""></li>
        </ul>
        </header>

    <div class="Mohammed">
        <h2>Mohammed</h2>
            <p>
                À seulement 20 ans, Mohammed possède déjà une expérience remarquable.
                Avec un Bac scientifique en poche, il a choisi de travailler comme vendeur d'articles technologiques et réparateur en t
                Son parcours reflète une combinaison unique de compétences techniques et de connaissances en sciences. Déterminé et hab
                il est prêt à poursuivre son cheminement professionnel dans le domaine de la technologie,
                où chaque interaction est une opportunité de croissance et d'accomplissement.
            </p>
            <img src="./assets/images/castor.png" alt="">
        </div>
    </div>
</body>
</html>
```

CSS :

assets > CSS > Mohammed.css > body

```css
.lapage{
    text-align: center;
}
li{
    list-style-type: none;
    display: inline;
    padding: 10px;
    font-size: 20px;
}
.lapage img{
    width: 20px;
    height: 20px;
}
a{
    color: white;
}
a{
    text-decoration: none;
}
li:hover{
    background-color: rgb(133, 133, 139, 0.87);
    border-radius: 10px;
}
.Mohammed img{
    width: 45px;
    height: 45px;
}
.Mohammed{
    width: 750px;
    margin: auto;
```

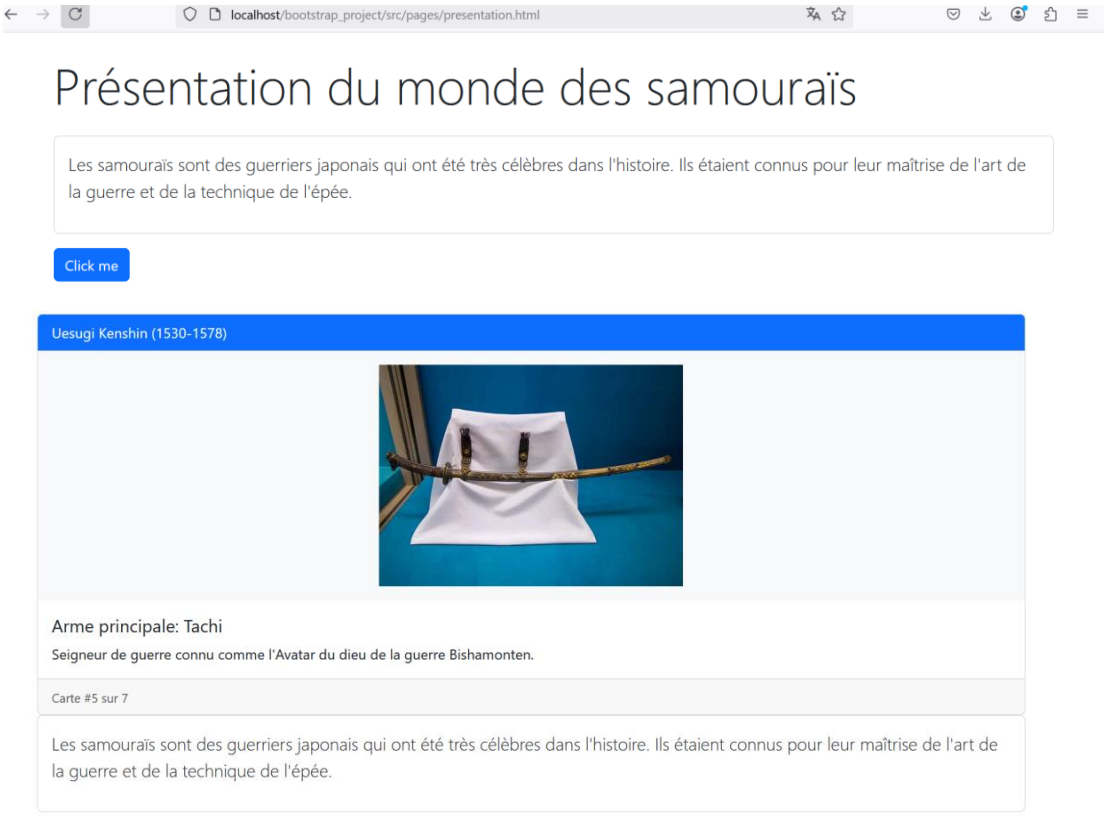
assets > CSS > Mohammed.css > body

```css
.Mohammed{
    width: 750px;
    margin: auto;
    background-color: rgba(133, 133, 139, 0.87);
    border-radius: 15px;
    margin-top: 10%;
}
body{
    background-image: url("../images/DesignSansTitre.png");
    background-repeat: no-repeat;
    background-size: 100%;
    background-position: center;
    min-height: 100vh;
    margin: 0;
    padding: 0;
}
.thehead{
    background-color: rgba(194, 178, 128, 0.842);
    border-radius: 5px;
    height: 120px;
}
```

- **L’utilisation des frameworks: Tailwind CSS, Bootstrap, Materialize**

1. **Bootstrap.**

Visuel des pages en Bootstrap CSS :



Pages JS :

src > js > presentation.js > document.addEventListener('DOMContentLoaded') callback > generateUniqueCard

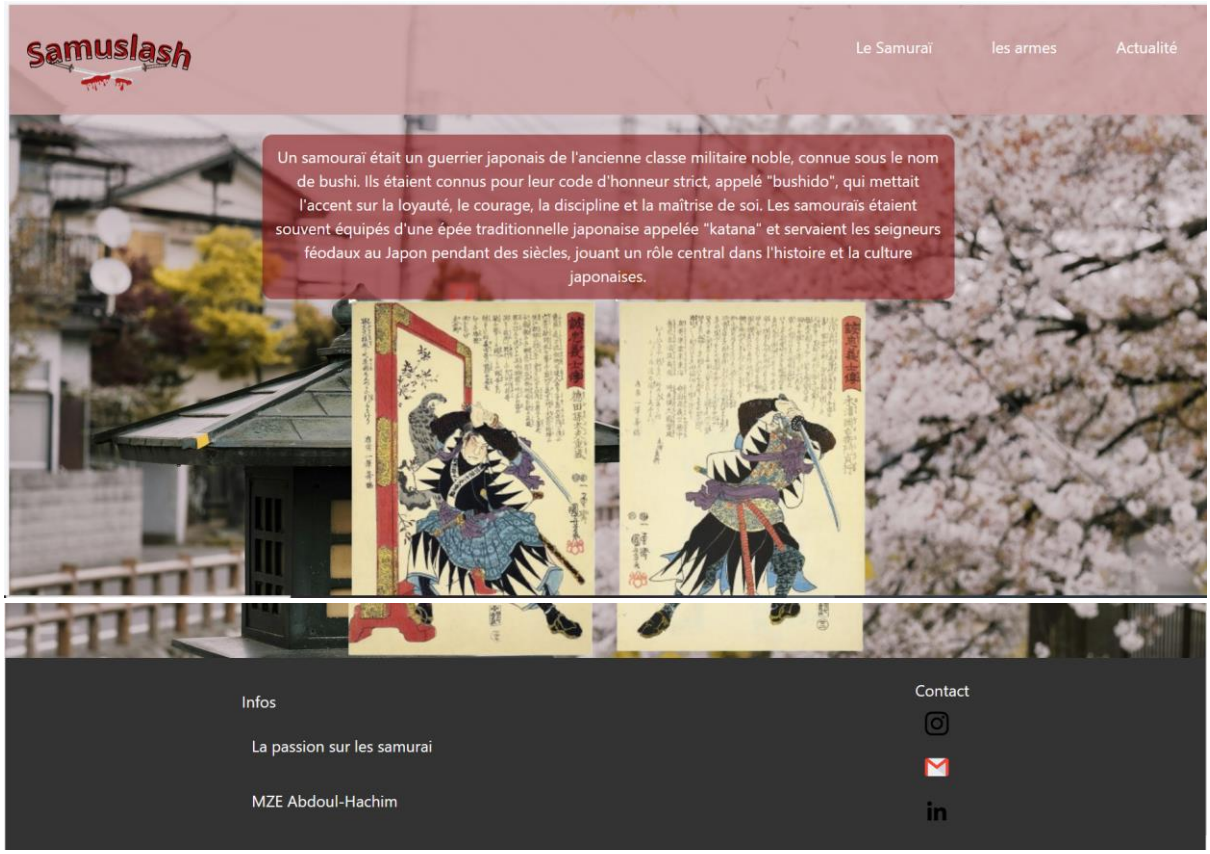
```js
document.addEventListener('DOMContentLoaded', () => {
    // Récupère le bouton et le conteneur où les cartes seront ajoutées
    const btnElement = document.getElementById('btn');
    const container = document.querySelector('.container');

    // Tableau de données pour générer des cartes uniques
    const samuraisData = [
        { nom: "Miyamoto Musashi", arme: "Deux sabres", période: "1584-1645", description: "Célèbre pour sa technique à deux sabres et
        { nom: "Oda Nobunaga", arme: "Katana", période: "1534-1582", description: "Grand stratège militaire qui a commencé l'unificati
        { nom: "Tokugawa Ieyasu", arme: "Nodachi", période: "1543-1616", description: "Fondateur du shogunat Tokugawa qui a gouverné l
        { nom: "Takeda Shingen", arme: "Naginata", période: "1521-1573", description: "Daimyo renommé pour ses tactiques militaires et
        { nom: "Uesugi Kenshin", arme: "Tachi", période: "1530-1578", description: "Seigneur de guerre connu comme l'Avatar du dieu de
        { nom: "Date Masamune", arme: "Katana", période: "1567-1636", description: "Surnommé 'Le Dragon à Un Œil', connu pour son cara
        { nom: "Sanada Yukimura", arme: "Lance", période: "1567-1615", description: "Considéré comme un guerrier héroïque et l'un des
    ];

    // Pour suivre les cartes déjà générées
    const generatedIndexes = new Set();

    // Fonction pour générer une carte unique
    const generateUniqueCard = () => {
        // Si toutes les cartes ont été générées, on réinitialise
        if (generatedIndexes.size === samuraisData.length) {
            alert("Toutes les cartes ont été générées. Réinitialisation.");
            generatedIndexes.clear();
        }

        // Choisir un index aléatoire qui n'a pas encore été utilisé
        let randomIndex;
        do {
            randomIndex = Math.floor(Math.random() * samuraisData.length);
        } while (generatedIndexes.has(randomIndex));

        // Marquer cet index comme utilisé
        generatedIndexes.add(randomIndex);

        // Créer la carte avec les données du samurai
        const samurai = samuraisData[randomIndex];
        const cardHTML = `
            <div class="card mt-3 unique-card-${randomIndex}">
                <div class="card-header bg-primary text-white">
                    ${samurai.nom} (${samurai.période})
                </div>
                <div class="text-center p-3 bg-light">
                    <img src="${samurai.image}" class="img-fluid" alt="${samurai.nom}"
                        style="max-height: 250px; width: auto;"
                        onerror="this.onerror=null; this.src='https://via.placeholder.com/300x200?text=Image+non+disponible'; console
                </div>
                <div class="card-body">
                    <h5 class="card-title">Arme principale: ${samurai.arme}</h5>
                    <p class="card-text">${samurai.description}</p>
                </div>
                <div class="card-footer">
                    <small class="text-muted">Carte #${randomIndex + 1} sur ${samuraisData.length}</small>
                </div>
            </div>
        `;

        // Insère la carte au début du conteneur (après le titre)
        const h1Element = container.querySelector('h1');
        h1Element.insertAdjacentHTML('afterend', cardHTML);

        // Log pour le débogage
        console.log(`Carte générée pour ${samurai.nom} avec l'image: ${samurai.image}`);
    };

    // Ajouter l'écouteur d'événement pour le clic sur le bouton
    btnElement.addEventListener('click', generateUniqueCard);
});
```

Page HTML :

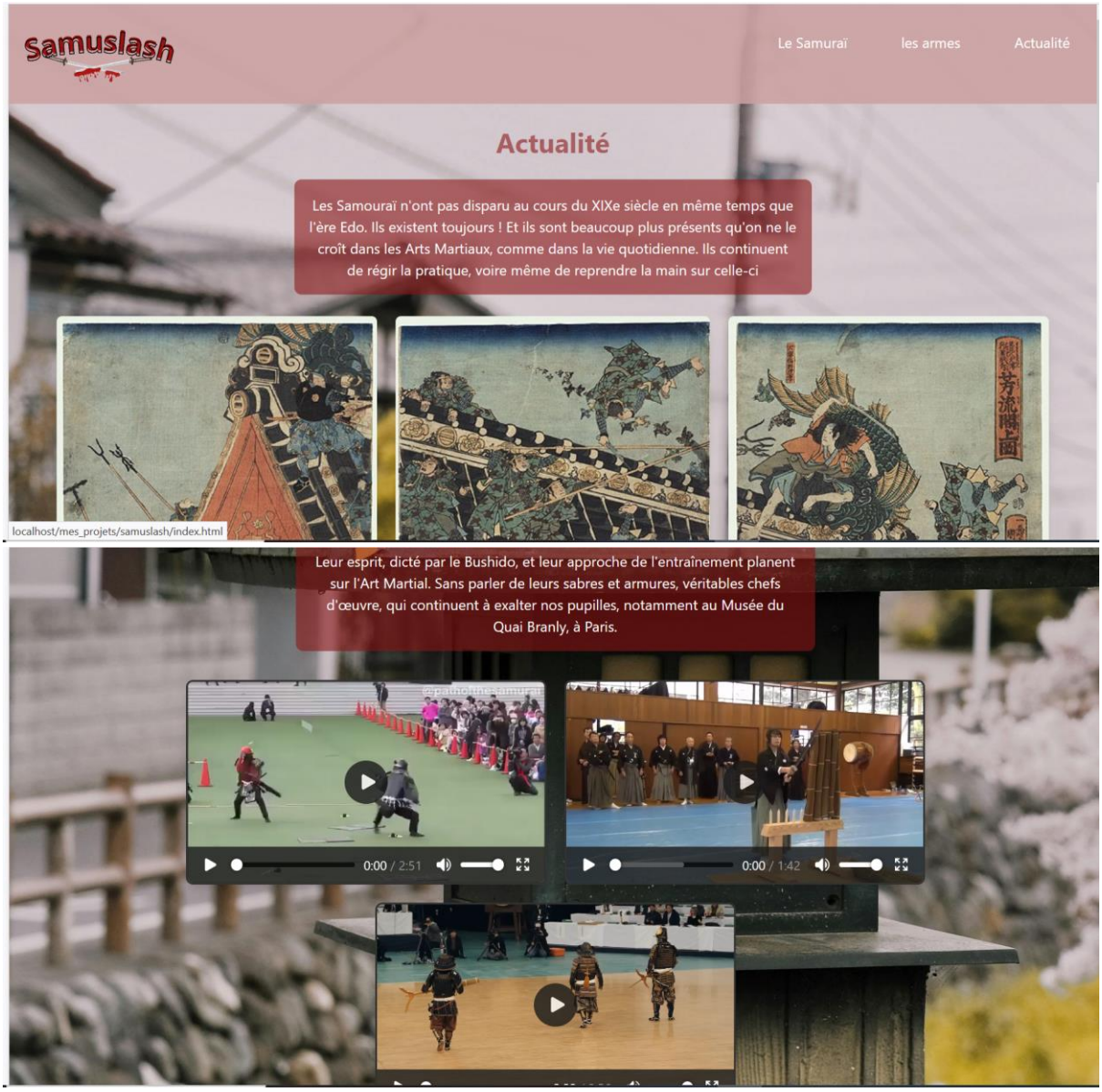
```
src > pages > presentation.html > html > body > div.container.mt-4 > div.card.mt-7
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>presentation du monde des samurai</title>
7       <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6u
8   </head>
9   <body>
10      <div class="container mt-4">
11          <h1 class="display-4 mb-4">Présentation du monde des samouraïs</h1>
12          <div class="card mt-7">
13              <div class="card-body">
14                  <p class="lead">
15                      Les samouraïs sont des guerriers japonais qui ont été très célèbres dans l'histoire.
16                      Ils étaient connus pour leur maîtrise de l'art de la guerre et de la technique de l'épée.
17                  </p>
18              </div>
19          </div>
20          <button id="btn" class="btn btn-primary mt-3">Click me</button>
21      </div>
22      <script src="../js/presentation.js"></script>
23  </body>
24  </html>
```

2. Tailwind CSS.

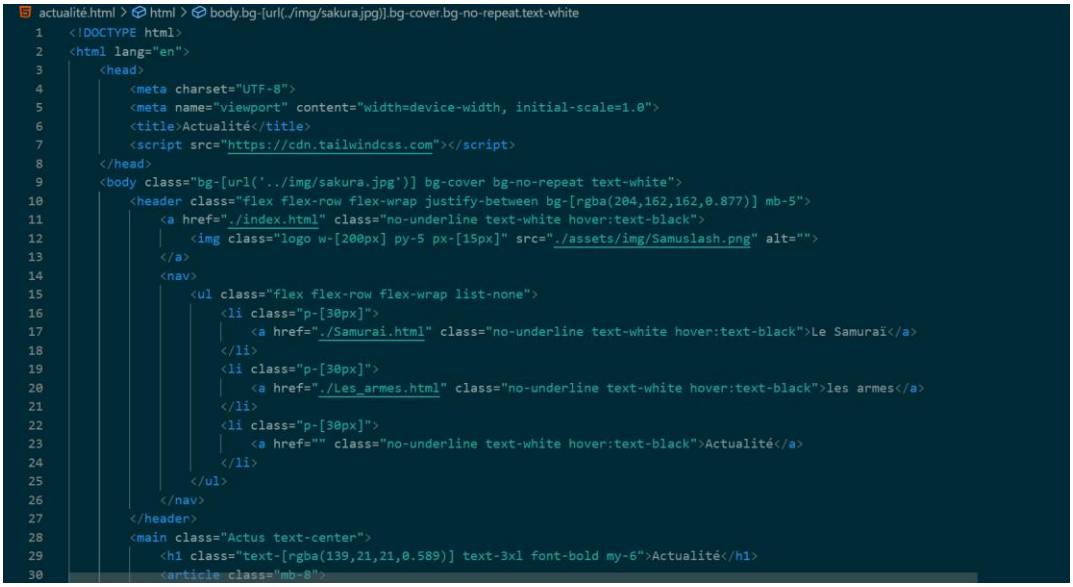
Page d'accueil en Tailwind CSS :



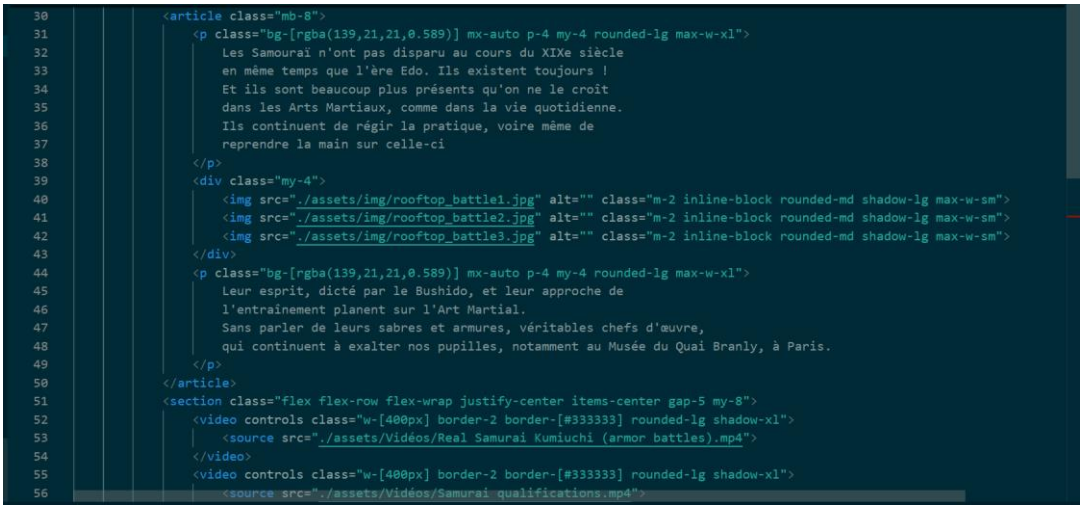
Page d'actualité en Tailwind CSS :



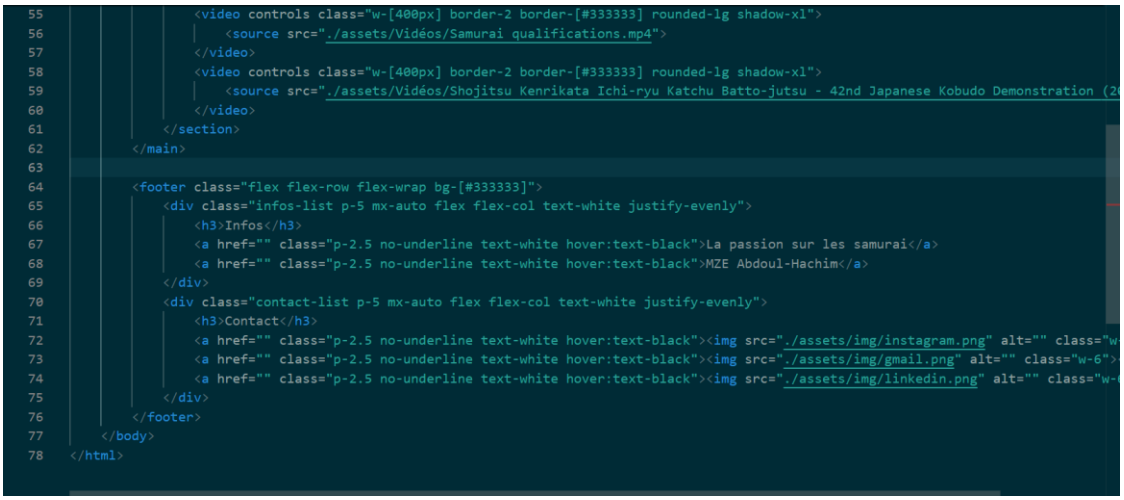
Code de la page d’actualité en Tailwind CSS :



```
actualité.html > html > body.bg-[url(../img/sakura.jpg)].bg-cover.bg-no-repeat.text-white
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Actualité</title>
        <script src="https://cdn.tailwindcss.com"></script>
    </head>
    <body class="bg-[url('../img/sakura.jpg')] bg-cover bg-no-repeat text-white">
        <header class="flex flex-row flex-wrap justify-between bg-[rgba(204,162,162,0.877)] mb-5">
            <a href="./index.html" class="no-underline text-white hover:text-black">
                <img class="logo w-[200px] py-5 px-[15px]" src="./assets/img/Samuslash.png" alt="">
            </a>
            <nav>
                <ul class="flex flex-row flex-wrap list-none">
                    <li class="p-[30px]">
                        <a href="./Samurai.html" class="no-underline text-white hover:text-black">Le Samuraï</a>
                    </li>
                    <li class="p-[30px]">
                        <a href="./Les_armes.html" class="no-underline text-white hover:text-black">les armes</a>
                    </li>
                    <li class="p-[30px]">
                        <a href="" class="no-underline text-white hover:text-black">Actualité</a>
                    </li>
                </ul>
            </nav>
        </header>
        <main class="Actus text-center">
            <h1 class="text-[rgba(139,21,21,0.589)] text-3xl font-bold my-6">Actualité</h1>
            <article class="mb-8">
```



```
            <article class="mb-8">
                <p class="bg-[rgba(139,21,21,0.589)] mx-auto p-4 my-4 rounded-lg max-w-xl">
                    Les Samouraï n'ont pas disparu au cours du XIXe siècle
                    en même temps que l'ère Edo. Ils existent toujours !
                    Et ils sont beaucoup plus présents qu'on ne le croît
                    dans les Arts Martiaux, comme dans la vie quotidienne.
                    Ils continuent de régir la pratique, voire même de
                    reprendre la main sur celle-ci
                </p>
                <div class="my-4">
                    <img src="./assets/img/rooftop_battle1.jpg" alt="" class="m-2 inline-block rounded-md shadow-lg max-w-sm">
                    <img src="./assets/img/rooftop_battle2.jpg" alt="" class="m-2 inline-block rounded-md shadow-lg max-w-sm">
                    <img src="./assets/img/rooftop_battle3.jpg" alt="" class="m-2 inline-block rounded-md shadow-lg max-w-sm">
                </div>
                <p class="bg-[rgba(139,21,21,0.589)] mx-auto p-4 my-4 rounded-lg max-w-xl">
                    Leur esprit, dicté par le Bushido, et leur approche de
                    l'entrainement planent sur l'Art Martial.
                    Sans parler de leurs sabres et armures, véritables chefs d'œuvre,
                    qui continuent à exalter nos pupilles, notamment au Musée du Quai Branly, à Paris.
                </p>
            </article>
            <section class="flex flex-row flex-wrap justify-center items-center gap-5 my-8">
                <video controls class="w-[400px] border-2 border-[#333333] rounded-lg shadow-xl">
                    <source src="./assets/Vidéos/Real Samurai Kumiuchi (armor battles).mp4">
                </video>
                <video controls class="w-[400px] border-2 border-[#333333] rounded-lg shadow-xl">
                    <source src="./assets/Vidéos/Samurai qualifications.mp4">
```



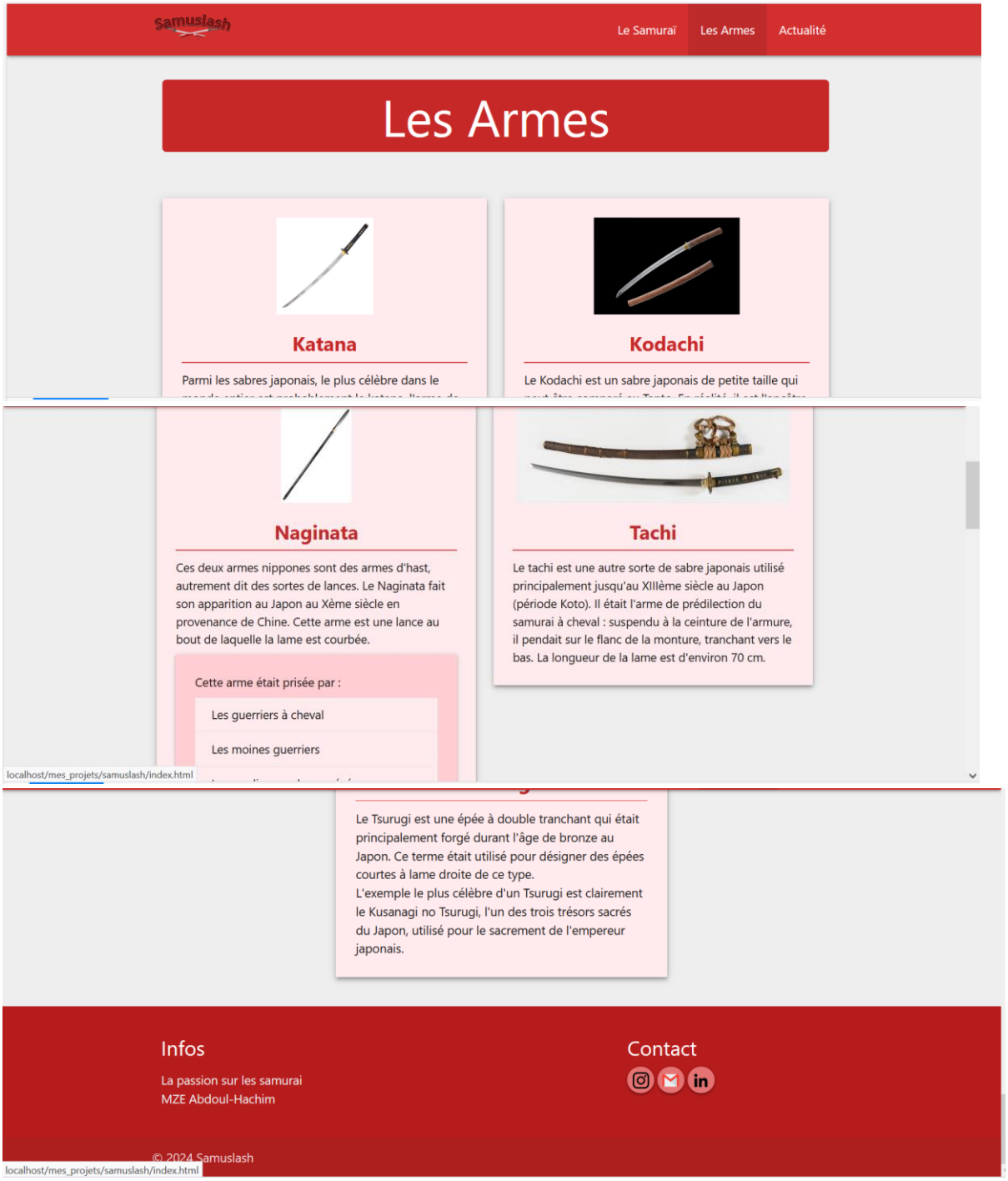
```
                <video controls class="w-[400px] border-2 border-[#333333] rounded-lg shadow-xl">
                    <source src="./assets/Vidéos/Samurai qualifications.mp4">
                </video>
                <video controls class="w-[400px] border-2 border-[#333333] rounded-lg shadow-xl">
                    <source src="./assets/Vidéos/Shojitsu Kenrikata Ichi-ryu Katchu Batto-jutsu - 42nd Japanese Kobudo Demonstration (20
                </video>
            </section>
        </main>

        <footer class="flex flex-row flex-wrap bg-[#333333]">
            <div class="infos-list p-5 mx-auto flex flex-col text-white justify-evenly">
                <h3>Infos</h3>
                <a href="" class="p-2.5 no-underline text-white hover:text-black">La passion sur les samurai</a>
                <a href="" class="p-2.5 no-underline text-white hover:text-black">MZE Abdoul-Hachim</a>
            </div>
            <div class="contact-list p-5 mx-auto flex flex-col text-white justify-evenly">
                <h3>Contact</h3>
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/instagram.png" alt="" class="w
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/gmail.png" alt="" class="w-6">
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/linkedin.png" alt="" class="w-
            </div>
        </footer>
    </body>
</html>
```

Code de la page d’accueil en Tailwind CSS :

```
index.html > html > body.bg-[url(./assets/img/sakura.jpg)].bg-cover.bg-no-repeat.text-white
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Samuslash</title>
        <script src="https://cdn.tailwindcss.com"></script>
    </head>
    <body class="bg-[url('./assets/img/sakura.jpg')] bg-cover bg-no-repeat text-white">
        <header class="flex flex-row flex-wrap justify-between bg-[rgba(204,162,162,0.877)] mb-5">
            <a href="./index.html" class="no-underline text-white hover:text-black">
                <img class="logo w-[200px] py-5 px-[15px]" src="./assets/img/Samuslash.png" alt="">
            </a>
            <nav>
                <ul class="flex flex-row flex-wrap list-none">
                    <li class="p-[30px]">
                        <a href="./Samurai.html" class="no-underline text-white hover:text-black">Le Samuraï</a>
                    </li>
                    <li class="p-[30px]">
                        <a href="./Les_armes.html" class="no-underline text-white hover:text-black">les armes</a>
                    </li>
                    <li class="p-[30px]">
                        <a href="./actualité.html" class="no-underline text-white hover:text-black">Actualité</a>
                    </li>
                </ul>
            </nav>
        </header>
        <main class="page-acceuil">
            <article class="present-text w-[700px] mx-auto pt-2.5 bg-[rgba(139,21,21,0.589)] p-[1%] rounded-[10px] text-center">
                <p>
                    Un samouraï était un guerrier japonais de l'ancienne
                    classe militaire noble, connue sous le nom de bushi. Ils étaient
                    connus pour leur code d'honneur strict, appelé "bushido", qui mettait l'accent
                    sur la loyauté, le courage, la discipline et la maîtrise de soi. Les samouraïs étaient souvent équipés
                    d'une épée traditionnelle japonaise appelée "katana" et servaient les seigneurs féodaux au Japon pendant des siècle
                    jouant un rôle central dans l'histoire et la culture japonaises.
                </p>

            </article>
            <div class="les-tableau flex flex-row justify-center gap-5 w-full">
                <div class="tableau-1">
                    <img src="./assets/img/tableau.png" alt="" class="max-w-full h-auto p-0 mx-auto block">
                </div>
                <div class="tableau-2">
                    <img src="./assets/img/tableau_de_samurai.png" alt="" class="max-w-full h-auto p-0 mx-auto block">
                </div>
            </div>
        </main>
        <footer class="flex flex-row flex-wrap bg-[#333333]">
            <div class="infos-list p-5 mx-auto flex flex-col text-white justify-evenly">
                <h3>Infos</h3>
                <a href="" class="p-2.5 no-underline text-white hover:text-black">La passion sur les samuraï</a>
                <a href="" class="p-2.5 no-underline text-white hover:text-black">MZE Abdoul-Hachim</a>
            </div>
            <div class="contact-list p-5 mx-auto flex flex-col text-white justify-evenly">
                <h3>Contact</h3>
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/instagram.png" alt="" class="w
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/gmail.png" alt="" class="w-6">
                <a href="" class="p-2.5 no-underline text-white hover:text-black"><img src="./assets/img/linkedin.png" alt="" class="w-
            </div>
        </footer>
    </body>
</html>
```

3. Materialize.

Ma page en Materialize CSS :



Code :

Les_armes.html > html > body.grey.lighten-3 > div.navbar-fixed > nav.red.darken-2

```html
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Les Armes</title>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    </head>
    <body class="grey lighten-3">
        <div class="navbar-fixed">
            <nav class="red darken-2">
                <div class="nav-wrapper container">
                    <a href="./index.html" class="brand-logo">
                        <img src="./assets/img/Samuslash.png" alt="Logo" style="height: 56px; padding: 8px 0;">
                    </a>
                    <ul class="right hide-on-med-and-down">
                        <li><a href="./Samurai.html">Le Samuraï</a></li>
                        <li class="active"><a href="#">Les Armes</a></li>
                        <li><a href="./actualité.html">Actualité</a></li>
                    </ul>
                </div>
            </nav>
        </div>

        <main class="container">
            <div class="row">
                <div class="col s12 center-align">
                    <h1 class="white-text red darken-3 center-align" style="padding: 10px; margin-top: 30px; margin-bottom: 30px; borde
                </div>
            </div>
            <div class="row">
                <div class="col s12 m6">
                    <div class="card z-depth-2">
                        <div class="card-content red lighten-5">
                            <div class="center-align">
                                <img src="./assets/img/katana-acier-damas-noir.jpg" alt="Katana" class="responsive-img" style="max-heig
                            </div>
                            <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Katana</h5>
                            <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                            <p>
                                Parmi les sabres japonais, le plus célèbre dans le monde entier est probablement le katana,
                                l'arme de prédilection du samouraï. Un katana est un sabre courbé à la lame assez longue (plus de 60 cm
                                et fine avec un seul tranchant. Il était porté glissé à la ceinture, le tranchant vers le haut.
                            </p>
                        </div>
                    </div>
                </div>
                <div class="col s12 m6">
                    <div class="card z-depth-2">
                        <div class="card-content red lighten-5">
                            <div class="center-align">
                                <img src="./assets/img/kodachi-authentique.webp" alt="Kodachi" class="responsive-img" style="max-height
                            </div>
                            <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Kodachi</h5>
                            <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
```

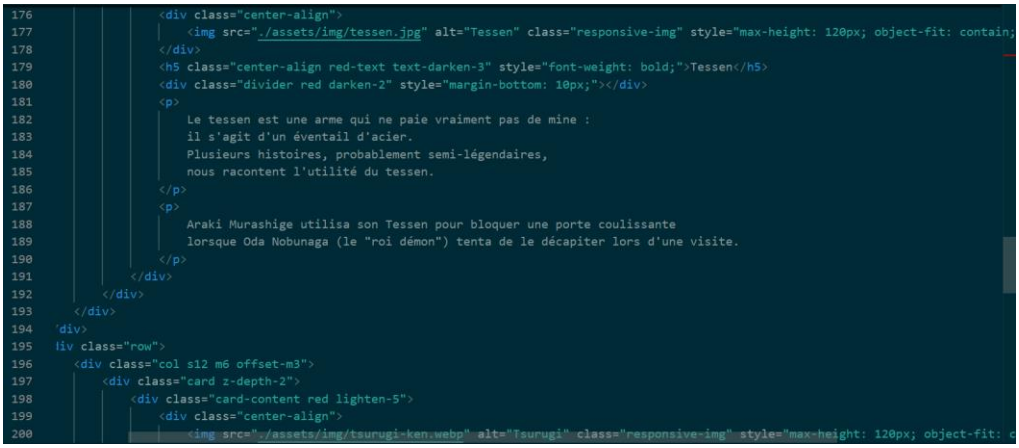
```
                        <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                        <p>
                            Le Kodachi est un sabre japonais de petite taille qui peut être comparé au Tanto.
                            En réalité, il est l'ancêtre du Wakizashi avec lequel il est souvent confondu : le Kodachi date de l'ère
                            tandis que l'autre n'apparaît que durant l'ère Muromachi.
                        </p>
                        <p>
                            Sa forme se rapproche un peu du Tachi et de ce fait il est plus fin et courbé
                            que le Wakizashi. Le Kodachi mesure entre 40 et 65 cm de long.
                        </p>
                    </div>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col s12 m6">
                <div class="card z-depth-2">
                    <div class="card-content red lighten-5">
                        <div class="center-align">
                            <img src="./assets/img/lance-yari--an-600025-1.webp" alt="Naginata" class="responsive-img" style="max-h
                        </div>
                        <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Naginata</h5>
                        <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                        <p>
                            Ces deux armes nippones sont des armes d'hast, autrement dit des sortes de lances.
```

```
            Le Naginata fait son apparition au Japon au Xème siècle en provenance de Chine.
            Cette arme est une lance au bout de laquelle la lame est courbée.
        </p>
        <div class="card-panel red lighten-4">
            <span>Cette arme était prisée par :</span>
            <ul class="collection">
                <li class="collection-item red lighten-5">Les guerriers à cheval</li>
                <li class="collection-item red lighten-5">Les moines guerriers</li>
                <li class="collection-item red lighten-5">Les gardiennes des gynécées</li>
                <li class="collection-item red lighten-5">Les femmes des seigneurs</li>
            </ul>
        </div>
    </div>
</div>
</div>
v class="col s12 m6">
<div class="card z-depth-2">
    <div class="card-content red lighten-5">
        <div class="center-align">
            <img src="./assets/img/Tachi_1.webp" alt="Tachi" class="responsive-img" style="max-height: 120px; object-fit: contain;">
        </div>
        <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Tachi</h5>
        <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
```

```
        <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
        <p>
            Le tachi est une autre sorte de sabre japonais utilisé principalement jusqu'au XIIIème siècle
            au Japon (période Koto). Il était l'arme de prédilection du samurai à cheval : suspendu à la ceinture de
            l'armure, il pendait sur le flanc de la monture, tranchant vers le bas. La longueur de la lame est d'environ 70 cm.
        </p>
    </div>
</div>
iv>

lass="row">
v class="col s12 m6">
<div class="card z-depth-2">
    <div class="card-content red lighten-5">
        <div class="center-align">
            <img src="./assets/img/Nodachi-removebg-preview.png" alt="Nodachi" class="responsive-img" style="max-height: 120px; objec
        </div>
        <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Nodachi</h5>
        <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
        <p>
            Nodachi désigne littéralement "sabre de champ". Il s'agit d'un très
            long sabre dont la lame fait en moyenne entre 1m et 1m50 et qui était utilisé
            par les bushi sur les champs de bataille.
        </p>
        <p>
```

```
                <p>
                    La légende raconte même qu'en l'utilisant bien, on pouvait couper en deux un cheval et
                    son cavalier d'un seul coup !
                </p>
            </div>
        </div>
    </div>
    <div class="col s12 m6">
        <div class="card z-depth-2">
            <div class="card-content red lighten-5">
                <div class="center-align">
                    <img src="./assets/img/Les Torimono Sandōgu.jpg" alt="Torimono Sandōgu" class="responsive-img" style="max-height: 120p
                </div>
                <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Torimono Sandōgu</h5>
                <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                <p>
                    Torimono Sandōgu désigne trois armes d'hast utilisées pendant
                    l'ère Edo par les forces de l'ordre. Ces trois types d'armes sont
                    le Sasumata (double fourche), le Sodegarami (pointes) et le Tsukubō
                    (en forme de T). Elles n'étaient pas conçues pour tuer, mais plutôt pour
                    attraper.
                </p>
            </div>
        </div>
    </div>
</div>
```

```
</div>
<div class="row">
    <div class="col s12 m6">
        <div class="card z-depth-2">
            <div class="card-content red lighten-5">
                <div class="center-align">
                    <img src="./assets/img/Tanto-Katana-Japonais-1555.webp" alt="Tanto" class="responsive-img" style="max-height: 120px;
                </div>
                <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Tanto</h5>
                <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                <p>
                    Le tanto est un couteau japonais de moins de 30 cm.
                    Il était utilisé par des samouraïs en tant que lame
                    perce-armure ou même par des femmes combattantes.
                    Il en existe deux types principaux : l'hamidachi
                    (avec une garde de petite dimension) et l'aikuchi
                    (pas de garde, poignée en peau de poisson).
                </p>
            </div>
        </div>
    </div>
    <div class="col s12 m6">
        <div class="card z-depth-2">
            <div class="card-content red lighten-5">
                <div class="center-align">
                    <img src="./assets/img/tessen.jpg" alt="Tessen" class="responsive-img" style="max-height: 120px; object-fit: contain;
```

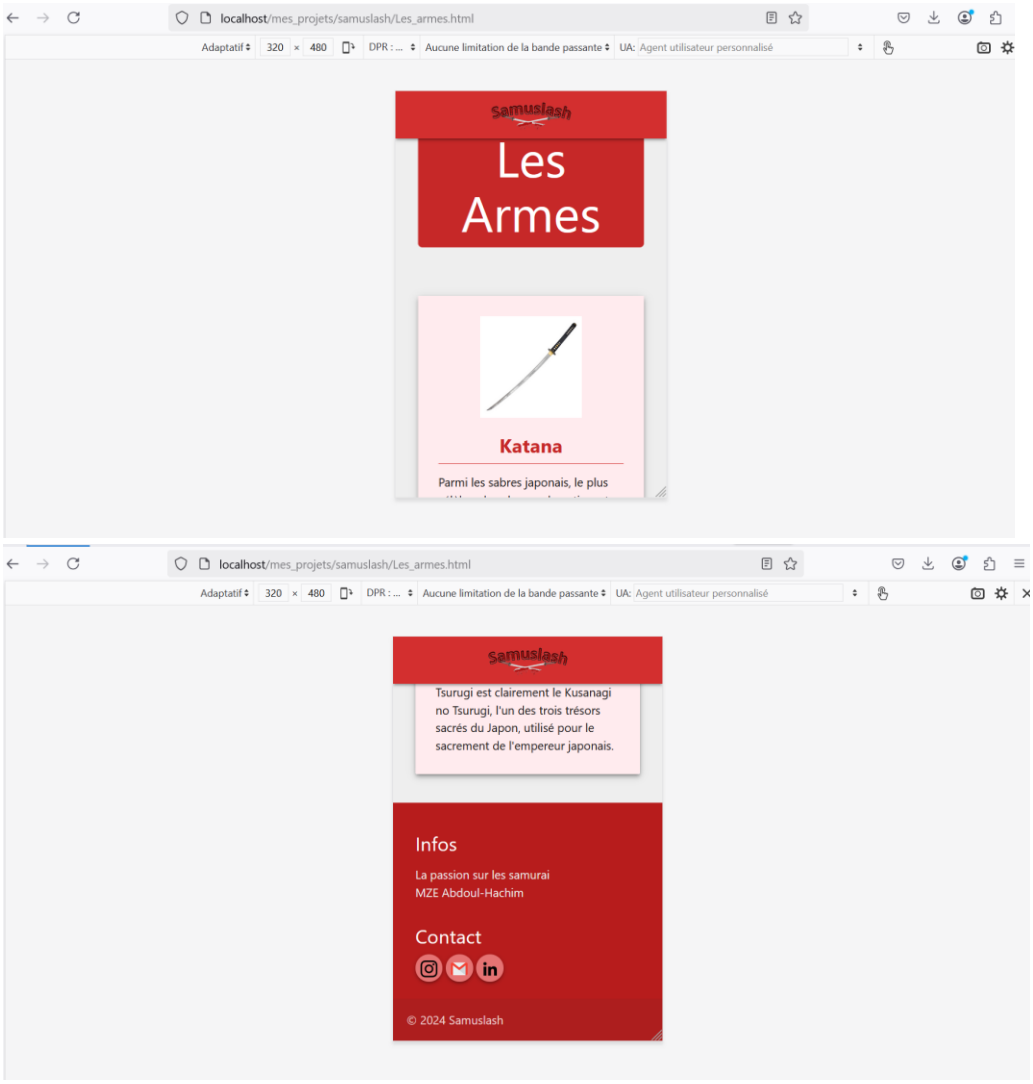


```
                <div class="center-align">
                    <img src="./assets/img/tessen.jpg" alt="Tessen" class="responsive-img" style="max-height: 120px; object-fit: contain;
                </div>
                <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Tessen</h5>
                <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                <p>
                    Le tessen est une arme qui ne paie vraiment pas de mine :
                    il s'agit d'un éventail d'acier.
                    Plusieurs histoires, probablement semi-légendaires,
                    nous racontent l'utilité du tessen.
                </p>
                <p>
                    Araki Murashige utilisa son Tessen pour bloquer une porte coulissante
                    lorsque Oda Nobunaga (le "roi démon") tenta de le décapiter lors d'une visite.
                </p>
            </div>
        </div>
    </div>
'div>
!iv class="row">
    <div class="col s12 m6 offset-m3">
        <div class="card z-depth-2">
            <div class="card-content red lighten-5">
                <div class="center-align">
                    <img src="./assets/img/tsurugi-ken.webp" alt="Tsurugi" class="responsive-img" style="max-height: 120px; object-fit: c
```

```
                    <img src="./assets/img/tsurugi-ken.webp" alt="Tsurugi" class="responsive-img" style="max-height: 120px; object-fit: co
                </div>
                <h5 class="center-align red-text text-darken-3" style="font-weight: bold;">Tsurugi</h5>
                <div class="divider red darken-2" style="margin-bottom: 10px;"></div>
                <p>
                    Le Tsurugi est une épée à double tranchant qui était principalement
                    forgé durant l'âge de bronze au Japon. Ce terme était utilisé pour désigner
                    des épées courtes à lame droite de ce type.
                </p>
                <p>
                    L'exemple le plus célèbre d'un Tsurugi est clairement le Kusanagi
                    no Tsurugi, l'un des trois trésors sacrés du Japon, utilisé pour le sacrement de l'empereur japonais.
                </p>
            </div>
        </div>
    </div>
</div>
iv>

 class="page-footer red darken-4">
v class="container">
<div class="row">
    <div class="col l6 s12">
        <h5 class="white-text">Infos</h5>
        <ul>
```

```
        <ul>
            <li><a class="grey-text text-lighten-3" href="#">La passion sur les samurai</a></li>
            <li><a class="grey-text text-lighten-3" href="#">MZE Abdoul-Hachim</a></li>
        </ul>
    </div>
    <div class="col l4 offset-l2 s12">
        <h5 class="white-text">Contact</h5>
        <div class="social-icons">
            <a href="#" class="btn-floating btn-small red lighten-2">
                <img src="./assets/img/instagram.png" alt="Instagram" style="width: 20px; height: 20px; margin-top: 7px;">
            </a>
            <a href="#" class="btn-floating btn-small red lighten-2">
                <img src="./assets/img/gmail.png" alt="Gmail" style="width: 20px; height: 20px; margin-top: 7px;">
            </a>
            <a href="#" class="btn-floating btn-small red lighten-2">
                <img src="./assets/img/linkedin.png" alt="LinkedIn" style="width: 20px; height: 20px; margin-top: 7px;">
            </a>
        </div>
    </div>
</div>
iv>
v class="footer-copyright">
<div class="container">
    © 2024 Samuslash
</div>
```

- **Création de layouts responsifs.**
  Voici les exemples pour les



1.4 Développement Frontend Dynamique

Le frontend dynamique signifie que la page web peut changer sans avoir besoin de la recharger entièrement.

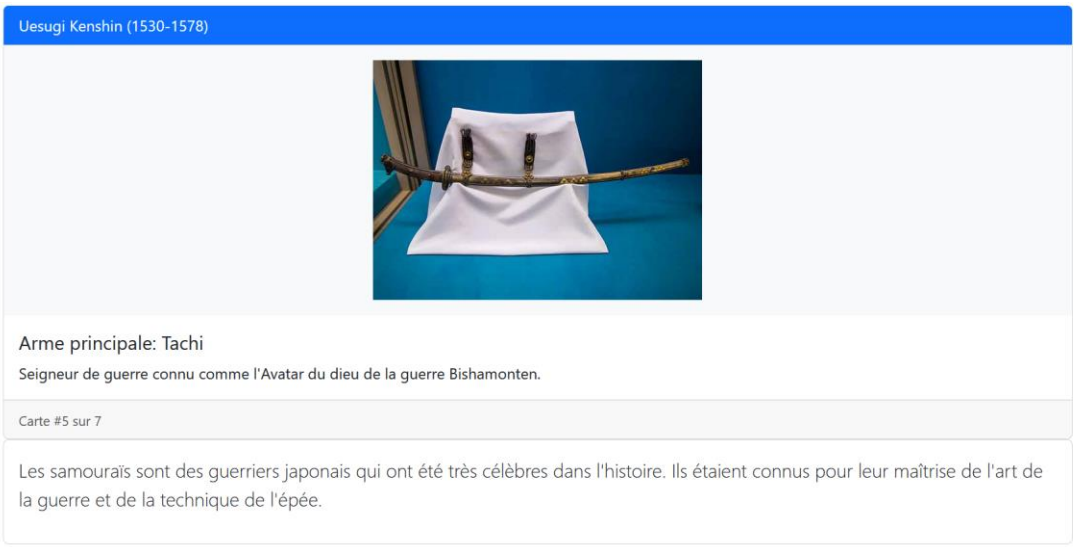
C'est l'inverse d'un site statique où le contenu est toujours le même.

Exemple :

Quand tu ouvres Facebook et que tu vois de nouveaux commentaires apparaître sans recharge la page, c'est du frontend dynamique.

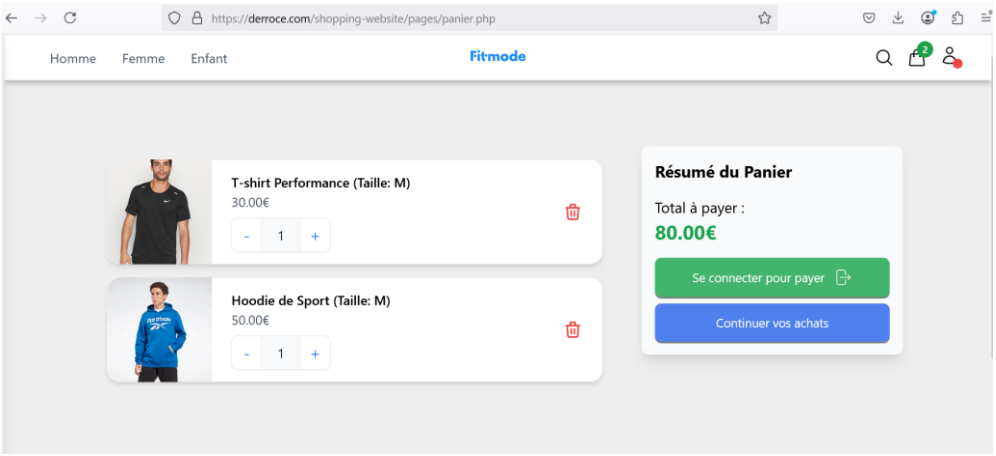
Développement avec les frameworks modernes :

- **React.js**
  Pour le projet dynamique côté client sans backend en React.js il s'agit d'un projet simple réalisé dans le but de démontrer le dynamisme d'une application. Le projet consiste à afficher une carte présentant des samurais lorsqu'un bouton est cliqué.
  Ici nous avons



- **Création de composants**
  L'exemple de composant créer que je peux donner est le formulaire de contact de mon portfolio.

- **Gestion de l'état des applications**
  L'état (ou state, ou state en anglais) représente les données en mémoire qui changent au fil de l'utilisation d'une application.
  Exemple : Le contenue d'un panier sur un site e-commerce.



- **Intégration d'APIs**

Pour l'intégrations d'API j'ai réalisé cela avec Stripe API une api de payement pour le projet e-commerce Stripe propose une variété de mode de payement que l'on peut intégrer au projet si-besoin tel que, Apple paye, google paye et les payements par carte bancaire.

2. Développement Backend

Le backend est la partie invisible d'un site web ou d'une application qui fait fonctionner tout ce qu'on ne voit pas. Il s'occupe de traiter les données, de gérer les comptes utilisateurs, de communiquer avec la base de données, et de répondre aux actions faites sur le site, comme une inscription ou une demande. Il joue aussi un rôle important dans la sécurité, en protégeant les informations sensibles comme les mots de passe. En résumé, le backend est le "cerveau" du site, qui travaille en coulisse pour que tout fonctionne Correctement.

Exemple : voici les captures des pages login.php et ad_product.PHP (connexion et ajout de produit qui sont tous deux gérer en backend,

```php
<?php
require 'db.php';

$name = $_POST['name'];
$price = $_POST['price'];
$description = $_POST['description'];

$stmt = $pdo->prepare("INSERT INTO products (name, price, description) VALUES (?, ?, ?)");
if ($stmt->execute([$name, $price, $description])) {
    echo "Produit ajouté.";
} else {
    echo "Erreur.";
}
```

```php
<?php
require 'db.php';

$email = $_POST['email'];
$password = $_POST['password'];

$stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");
$stmt->execute([$email]);
$user = $stmt->fetch();

if ($user && password_verify($password, $user['password'])) {
    $token = base64_encode($user['id'] . ':' . time());
    echo json_encode(['token' => $token]);
} else {
    echo "Identifiants invalides.";
}
```

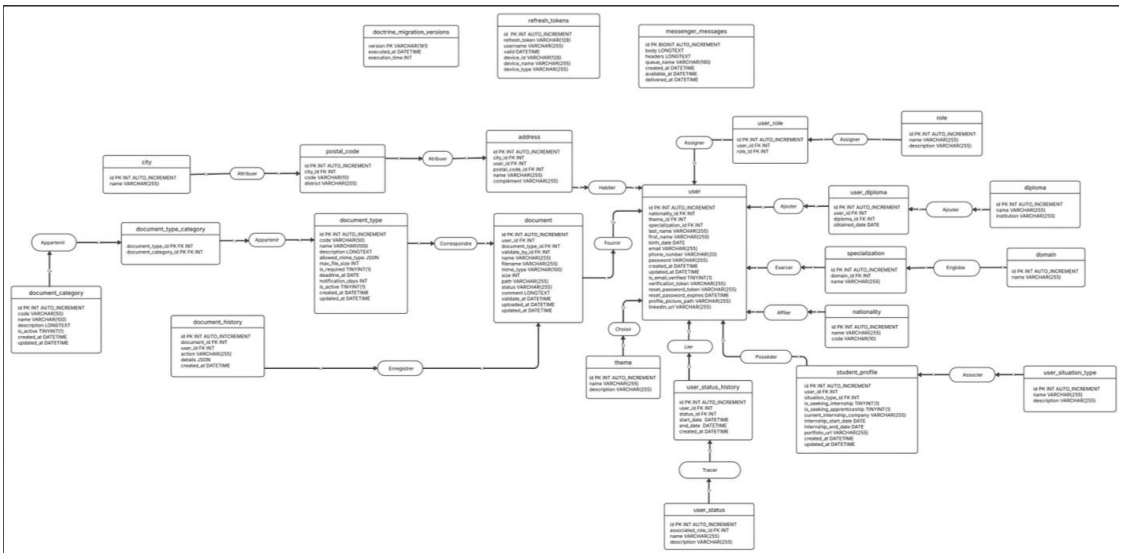
2.1 Bases de Données

Une base de données est un outil qui permet de stocker, organiser et retrouver facilement des informations. Elle est utilisée dans presque tous les sites web pour

garder des données comme les utilisateurs, les produits ou les commandes. Par exemple, sur un site e-commerce, la base de données enregistre qui a acheté quoi et quand.

Gestion des bases de données :

- ***Conception de schémas relationnels***



Le schéma ci-dessus est celui du projet a qui a pour but la gestion d'établissement ou d'entreprise pour le coup ce schéma vise les établissements scolaires, (projet : Big Projet) la méthode utilisée est la normalisation, la normalisation est un processus systématique appliqué à la conception des bases de données relationnelles, visant à organiser les données selon des règles précises appelées formes normales. Elle permet de structurer les relations de manière à réduire la redondance, éviter les anomalies (d'insertion, de mise à jour et de suppression) et garantir l'intégrité des données.

- ***Gestion des relations entre tables***

Voici un exemple de gestion de base de données sur un projet avec symfony.

```php
<?php

namespace App\Entity;

use App\Repository\AddressRepository;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Serializer\Annotation\Groups;

#[ORM\Entity(repositoryClass: AddressRepository::class)]
class Address
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    #[Groups(['address:read', 'user:read'])]
    private ?int $id = null;

    #[ORM\Column(length: 255)]
    #[Groups(['address:read', 'user:read'])]
    private ?string $name = null;

    #[ORM\Column(length: 255, nullable: true)]
    #[Groups(['address:read', 'user:read'])]
    private ?string $complement = null;

    #[ORM\ManyToOne(inversedBy: 'addresses')]
    #[ORM\JoinColumn(nullable: false)]
    #[Groups(['address:read'])]
    private ?City $city = null;

    #[ORM\ManyToOne(inversedBy: 'addresses')]
    #[ORM\JoinColumn(nullable: false)]
    #[Groups(['address:read', 'user:read'])]
    private ?PostalCode $postalCode = null;

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName(string $name): static
    {
        $this->name = $name;

        return $this;
    }

    public function getComplement(): ?string
    {
        return $this->complement;
    }
```

Dans un projet Symfony, la gestion des relations entre tables est assurée par l'ORM Doctrine. Celui-ci permet de modéliser les liens entre entités (représentant les tables de la base de données) directement dans le code à l'aide d'attributs PHP spécifiques.

Doctrine prend en charge les trois principaux types de relations :

- **OneToOne** (relation un à un),
- **OneToMany / ManyToOne** (relation un à plusieurs),
- **ManyToMany** (relation plusieurs à plusieurs).

La définition des relations s'effectue dans les entités à l'aide des attributs #[ORM\...]. Une fois les entités créées ou modifiées, des migrations sont générées via la console Symfony pour mettre à jour le schéma de la base de données.

Cette approche permet une gestion cohérente, centralisée et automatisée des relations, facilitant les interactions entre les objets PHP et les tables SQL tout en garantissant l'intégrité des données.

- ***Administration via phpMyAdmin***

Dans un projet Symfony, tu définis les informations de connexion à la base de données dans le fichier .env ou .env.local :
“DATABASE_URL="mysql://nom_utilisateur:mot_de_passe@127.0.0.1:3306/nom_de_la_base”.
Symfony va utiliser ces infos pour se connecter à la base.
PhpMyAdmin se connecte à cette même base via une interface graphique.
Symfony crée la structure, phpMyAdmin permet de la visualiser avec Symfony, tu peux créer les tables de la base à partir des entités PHP :
Exemple :

```php
#[ORM\Entity]
class Produit {
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 255)]
    private ?string $nom = null;

    #[ORM\Column]
    private ?float $prix = null;
}
```

Puis tu fais :

```
php bin/console make:migration
php bin/console doctrine:migrations:migrate
```

Symfony crée une table produit dans la base.

- ***Utilisation de MongoDB et Firebase***

  MongoDB

  MongoDB est une base de données. Imaginons un classeur rempli de fiches sur lesquelles on note des infos (comme des noms, des adresses, etc.). Mais au lieu d’utiliser des tableaux rigides comme dans Excel, MongoDB stocke les infos sous forme de documents (comme des fichier JSON). Cela permet d’avoir une structure flexible.
  Exemple :

Si l'on crée un site de recettes de cuisine, une recette peut contenir :

```
{
  "titre": "Pizza Margherita",
  "temps_preparation": "30 min",
  "ingredients": ["pâte", "tomate", "mozzarella"]
}
```

Et une autre recette peut avoir plus ou moins d'informations. MongoDB accepte ça facilement, contrairement à des bases classiques.

Firebase.

Firebase est un essemble d'outils proposés par Google pour aider les développeurs à créer des applications rapidement, surtout les apps mobiles et web.
Il comprend plusieurs services, mais voici deux très utilisés :

1. ***Firebase Authentification*** : permet d'ajouter facilement un système de connexion/inscription avec e-mail, Google, Facebook, etc.
   Exemple : si tu veux que les utilisateurs puissent se connecter à ton app sans coder tout le système de mot de passe, Firebase le fait pour toi.
2. ***Firebase Firestore*** : c'est aussi une base de données (comme MongoDB), mais hébergée par Google. Elle est temps réel, donc si quelqu'un modifie une donnée, tous les utilisateurs la voient immédiatement.

2.2 Développement Backend

Création d'APIs et sécurité :

- Développement avec Symfony et React.js
  Le développement avec Symfony et React.js est un projet de gestion pour les écoles, telles que la plateforme, mais qui peut être repris pour des entreprises ou d'autres structures.
  Voici une présentation rapide des fonctionnalités du projet avec Symfony et React.js tout d'abord le ce qu'il faut savoir sur le projet c'est que le backend était gérer par Symfony et le frontend par React.js et une base de données en MySQL, pour la première fonctionnalité il s'agit du système de connexion qui utilise l'architecture JWT (JSON Web Token) avec Symfony côté backend et React.js côté frontend.
  Composants de connexion
  - ConnectionModal.jsx : Modal de connexion avec validation
  - AuthForm.jsx : Formulaire d'authentification principal

- authService.js : Service principal gérant toute la logique d'auth

Pour la gestion côté backend voici des captures pour présenter comment tout cela a était traiter a commencer par le processus de connexion :

```
async login(email, password) {
    // 1. Nettoyage des données précédentes
    localStorage.clear();

    // 2. Appel API avec informations d'appareil
    const response = await apiService.post('/login_check', {
        username: email,
        password,
        device_id: deviceId,
        device_name: deviceName,
        device_type: deviceType
    });

    // 3. Stockage des tokens
    localStorage.setItem('token', response.token);
    localStorage.setItem('refresh_token', response.refresh_token);

    // 4. Décodage et stockage des infos utilisateur minimales
    const payload = decodeToken(response.token);
    const enhancedUser = {
        username: payload.username,
        roles: payload.roles,
        id: payload.id,
        // ...
    };
    localStorage.setItem('user', JSON.stringify(enhancedUser));
}
```

Puis la sécurité par JWT :

```
firewalls:
    login:
        pattern: ^/api/login
        stateless: true
        json_login:
            check_path: /api/login_check
            success_handler: lexik_jwt_authentication.handler.authentication_success
            failure_handler: lexik_jwt_authentication.handler.authentication_failure
    api:
        pattern: ^/api
        stateless: true
        jwt: ~
```

Et la gestion et la hiérarchie des rôles :

```php
public function getRoles(): array
{
    $roles = [];

    // Récupération des rôles depuis la table de liaison UserRole
    foreach ($this->userRoles as $userRole) {
        $roleName = $userRole->getRole()->getName();
        $roles[] = 'ROLE_' . strtoupper($roleName);
    }

    // Rôle par défaut si aucun rôle défini
    if (empty($roles)) {
        $roles[] = 'ROLE_GUEST';
    }

    return array_unique($roles);
}
```

```yaml
role_hierarchy:
    ROLE_ADMIN: ROLE_RECRUITER
    ROLE_SUPERADMIN: ROLE_ADMIN
```

- ***Tests avec Jest et Postman***

  La mise en place de tests est une étape essentielle dans le développement d'une application web. Elle permet de s'assurer que les fonctionnalités attendues sont bien respectées et que le code reste fiable dans le temps, notamment lors de futures évolutions. Deux outils complémentaires ont été utilisés dans le cadre du projet : Jest pour les tests unitaires, et Postman pour les tests des API.
  Tests avec Jest
  Jest est un framework de test JavaScript développé par Meta (anciennement Facebook), couramment utilisé pour effectuer des tests unitaires. Ces tests permettent de vérifier, de manière automatisée, que les fonctions ou composants développés renvoient bien les résultats attendus.
  Par exemple, si une fonction est censée additionner deux nombres, un test avec Jest permettra de valider qu'addition (2, 3) renvoie bien 5. Ce type de test est particulièrement utile pour détecter rapidement des erreurs logiques dans le code, et garantir sa stabilité en cas de modification future.

  Tests avec Postman
  Postman est un outil qui permet de tester et de documenter des API REST. Il est principalement utilisé pour envoyer des requêtes HTTP (GET, POST, PUT, DELETE) vers un serveur et analyser les réponses obtenues (statut HTTP, contenu, format, etc.).
  Dans le cadre du projet, Postman a été utilisé pour vérifier que chaque route de l'API renvoyait bien les données attendues, et que les différentes opérations (création, lecture, modification, suppression) fonctionnaient correctement,

notamment en simulant différents scénarios utilisateurs (données valides, erreurs, autorisations, etc.).

2.3 Déploiement

Le déploiement de l'application a été réalisé selon une procédure rigoureuse garantissant la stabilité, la sécurité et la maintenabilité du système en production.

Configuration des serveurs

Le serveur de production repose sur une infrastructure Linux (Ubuntu 22.04) avec une configuration web assurée par **Nginx** en tant que serveur HTTP inverse. Nginx a été choisi pour sa performance en matière de gestion des connexions simultanées, sa simplicité de configuration et son efficacité en proxy inverse.

Le fichier de configuration de Nginx comprend :

Le redémarrage automatique du service en cas d'erreur,

La redirection du trafic HTTP vers HTTPS via un certificat SSL (généré avec Let's Encrypt),

La gestion des en-têtes de sécurité (HSTS, X-Content-Type-Options, etc.),

Le routage vers le backend (Node.js/Express ou PHP/Laravel selon le stack du projet).

Dans le cas où Apache est utilisé (notamment pour les projets basés sur PHP), les modules nécessaires (mod_rewrite, mod_ssl) sont activés, et le `.htaccess` est configuré pour gérer les règles de réécriture et les protections de base.

Gestion des variables d'environnement

Les variables d'environnement sont stockées dans un fichier `.env`, non versionné (ajouté au `.gitignore`), afin de préserver la confidentialité des données sensibles (clés API, identifiants de base de données, secrets JWT, etc.). Ces variables sont chargées à l'exécution grâce à des bibliothèques telles que `dotenv` pour Node.js ou directement via `$_ENV` pour PHP. En environnement de production, ces variables sont définies dans le système ou injectées via les outils CI/CD.

Déploiement sur serveurs distants

Le déploiement s'effectue sur un **serveur distant VPS** (DigitalOcean, OVH ou équivalent) via un pipeline d'intégration continue (CI/CD) ou manuellement en SSH. Les étapes sont les suivantes :

- **Connexion SSH sécurisée** via clé publique/privée,
- **Pull depuis le dépôt Git** (GitHub/GitLab) sur le serveur distant,
- **Installation des dépendances** (`npm install`, `composer install`, etc.),
- **Migration et seed de la base de données** si nécessaire,
- **Build des assets front-end** (`npm run build`),
- **Redémarrage des services** via `pm2`, `systemd`, ou `Supervisor`.

Documentation des processus

L'ensemble du processus de déploiement est **documenté** dans un fichier `DEPLOY.md` inclus dans le dépôt du projet. Cette documentation décrit étape par étape :

Les prérequis serveur,

La configuration des outils,

Les scripts de déploiement utilisés,

Les commandes essentielles pour le redémarrage ou la surveillance des services.

Une attention particulière est portée à l'automatisation des tâches répétitives et à la traçabilité des mises en production (via des tags Git ou un changelog).

3. Projets Réalisés

Application des compétences dans des projets concrets :

- Fan site porté sur les samurai qui a pour nom Samuslash
- Le BigProjet un projet de gestion destiné aux établissement scolaire comme les entreprises.
- Fitemode un projet E-commerce destiné à la vente de vêtement de sport

Conclusion

Ce dossier reflète fidèlement les compétences acquises durant ma formation, démontrant ma capacité à intervenir sur l'ensemble du cycle de développement d'une application web moderne.