

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ SAADOUN
Nom d'usage ▶ MORETTI
Prénom ▶ DAVID
Adresse ▶ 2 impasse des romarins 34410 Sérignan

Titre professionnel visé

Développeur web et web mobile

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	5
▶ Maquetter une application	p.	1-13
▶ Réaliser une interface web statique et adaptable	p.	14-19
▶ Développer une interface utilisateur web dynamique	p.	20-23
▶ Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce .	p.	24-25
Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	
▶ Créer une base de données	p.	26-30
▶ Développer les composants d'accès aux données	p.	31-35
▶ Développer la partie back-end d'une application web ou web mobile	p.	36-43
▶ Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce	p.	44-51
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	54
Déclaration sur l'honneur	p.	
Documents illustrant la pratique professionnelle (facultatif)	p.	
Annexes (Si le RC le prévoit)	p.	

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► Maquetter une application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Projet :

Dans le cadre de ce projet, John Doe un client m'a contacté pour réaliser son projet portfolio. N'ayant plus le temps de coder avec ses activités, et voulant tous de même réduire ses coûts au maximum, il a choisi de jeter son dévolu sur un développeur junior.

Après avoir eu un rapide échange par mail avec John Doe (qui se prénomme désormais « le client »), il a souhaité apporter quelques précisions à son projet.

Le client souhaite que son portfolio présente les caractéristiques communes des portfolios. Il souhaite pouvoir montrer ses projets, ses articles de blogs, son profil et également un formulaire de contact avec une map, prendre des couleurs définies avec le client, avoir une bonne accessibilité et de bonnes performances de SEO. Le client souhaite également que son site soit accessible sur les formats mobiles, tablette et desktop.

Users storie :

Suite à cela, pour valider les demandes du client, j'ai rédigé les users stories dans un tableau pour valider les actions que va pouvoir réaliser un visiteur sur le site, les users stories permettent de comprendre clairement les besoins des utilisateurs de manière concrète de manière concrète, et de s'assurer que les fonctionnalités développées répondent aux attentes des utilisateurs.

Users stories :

En tant que	Je souhaite	Afin
Utilisateur	Consulter une page d'accueil avec simple et élégante	d'avoir un aperçu du portfolio
Utilisateur	Voir des images des projets réalisés	d'en savoir plus sur chaque projet
Utilisateur	Naviguer facilement à travers le portfolio	de trouver des informations sur les projets et les compétences
Utilisateur	Lire une description bien détaillée de chaque projet	Afin de connaître le projet, son résumé du projet, les technologies utilisées
Utilisateur	Envoyer un message ou une demande de contact à travers un formulaire	de poser des questions ou engager une collaboration
Utilisateur	Naviguer sur mobile, tablette ou desktop	de pouvoir avoir une accessibilité de navigation sur plusieurs formats

Après un rapide échange de mails avec le client John Doe pour vérifier que les user stories étaient bien conformes à ses attentes, j'ai travaillé sur les wireframes du projet.

Wireframes :

On peut dire que les wireframes sont des schémas simplifiés de la structuration d'un site web, qui permettent de valider visuellement l'idée que se fait le client sur son projet. Les wireframes permettent également d'apporter des précisions sur certains éléments de la structuration d'une page, par exemple les images, les boutons cliquables, les navigations etc...

Le wireframe se doit d'être simple, généralement en nuance de gris, car il permet simplement de définir la structuration des pages et non pas de l'aspect visuel des pages.

Pour ce projet de John Doe j'ai donc réalisé pour chaque page du site les wireframes aux formats mobile, tablette et desktop. Pour coller au maximum à l'idée de design mobile first, j'ai choisi de toujours commencer par le wireframe de la page mobile, puis de continuer sur le format tablette et desktop.

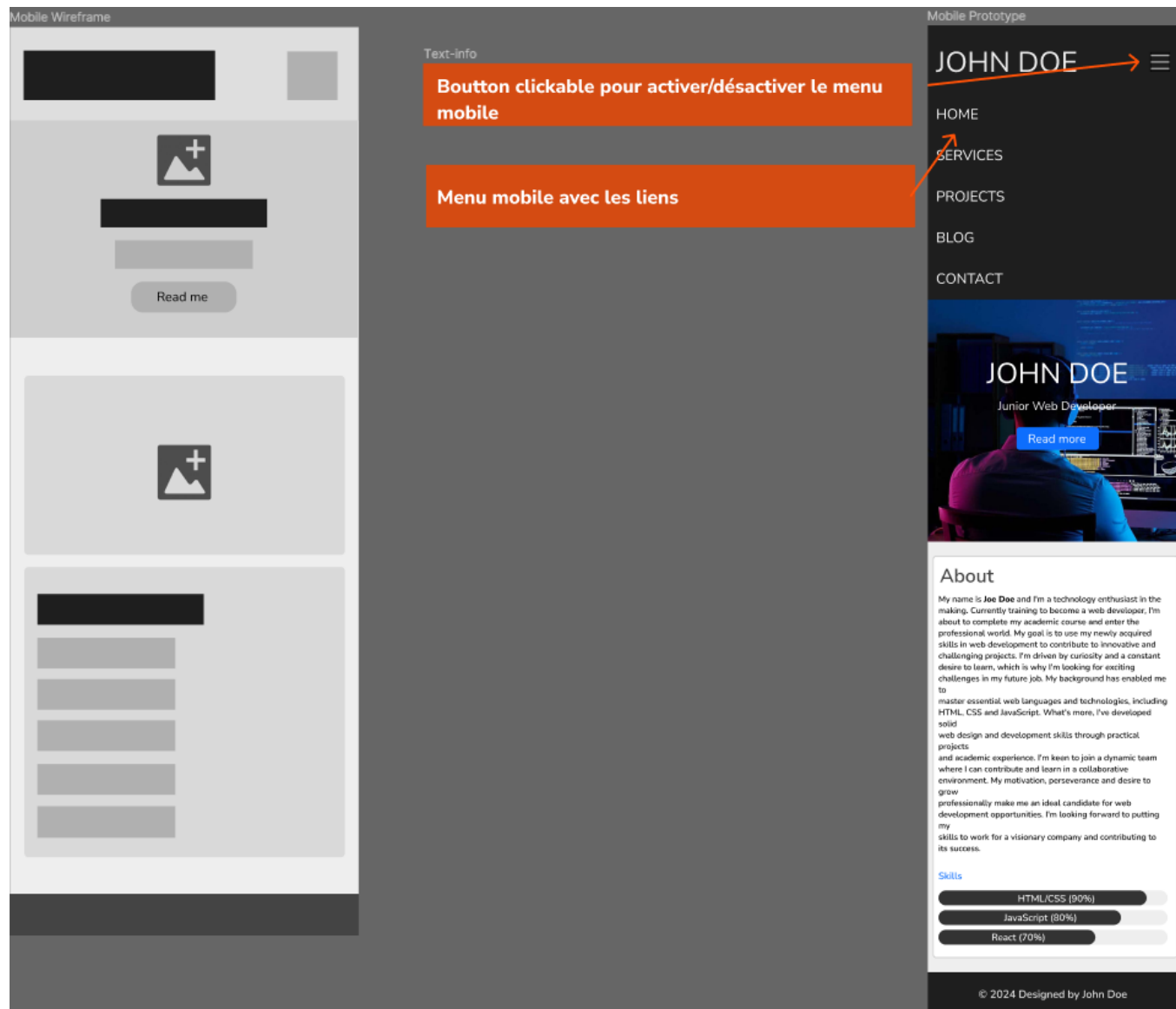
Wireframe générale de page « Home » :



On remarque sur cet exemple la représentation des trois formats spécifiés par le client. Je me suis basé sur le format mobile, puis j'ai décliné la structure de la page mobile aux formats tablette et desktop.

On remarque que les éléments ne changent pas, ils sont simplement réorganisés.

Wireframe de la page « Home » et Prototype - Format mobile :

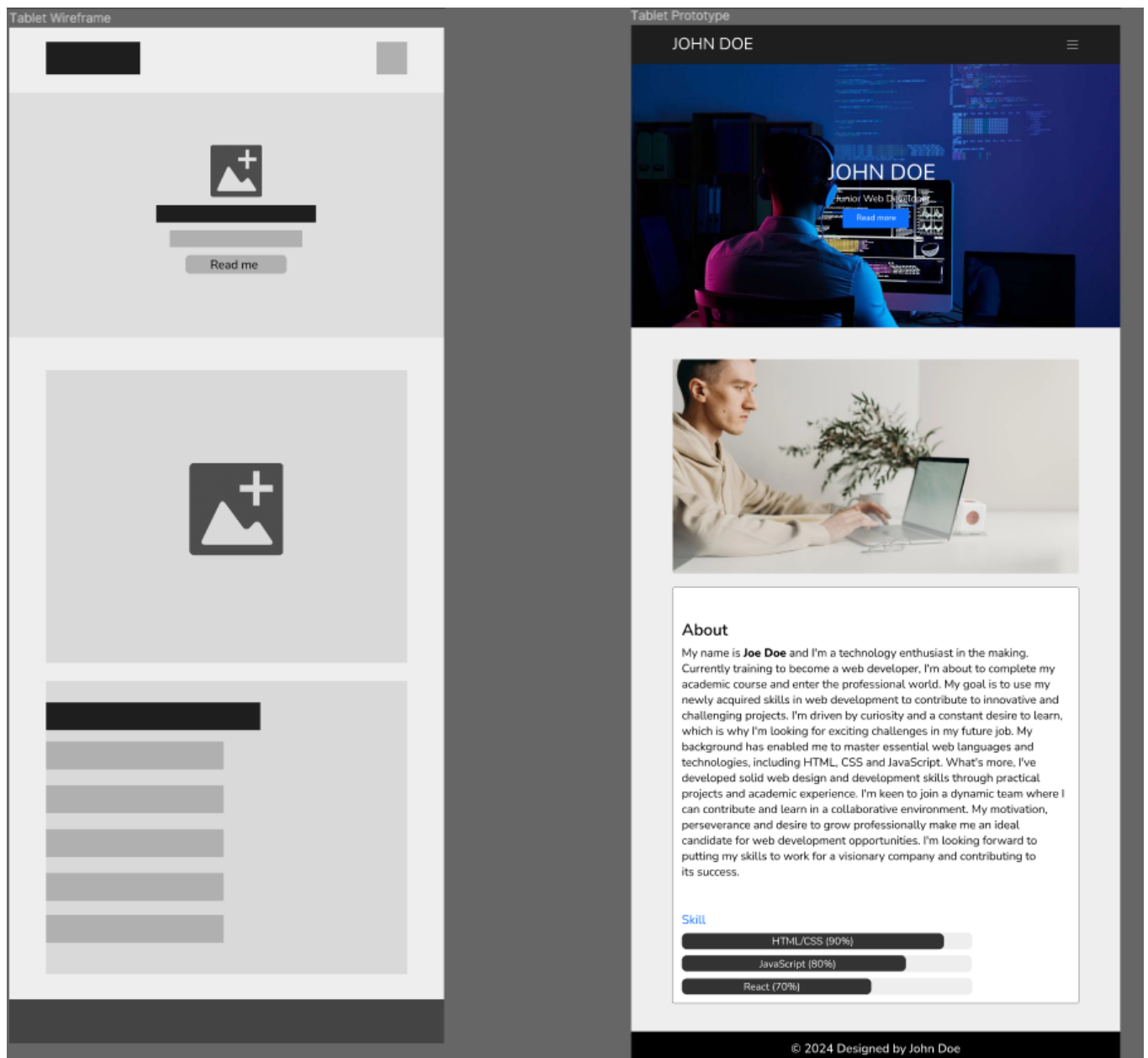


DOSSIER PROFESSIONNEL (DP)

On peut remarquer plusieurs choses sur ce wireframe et prototype :

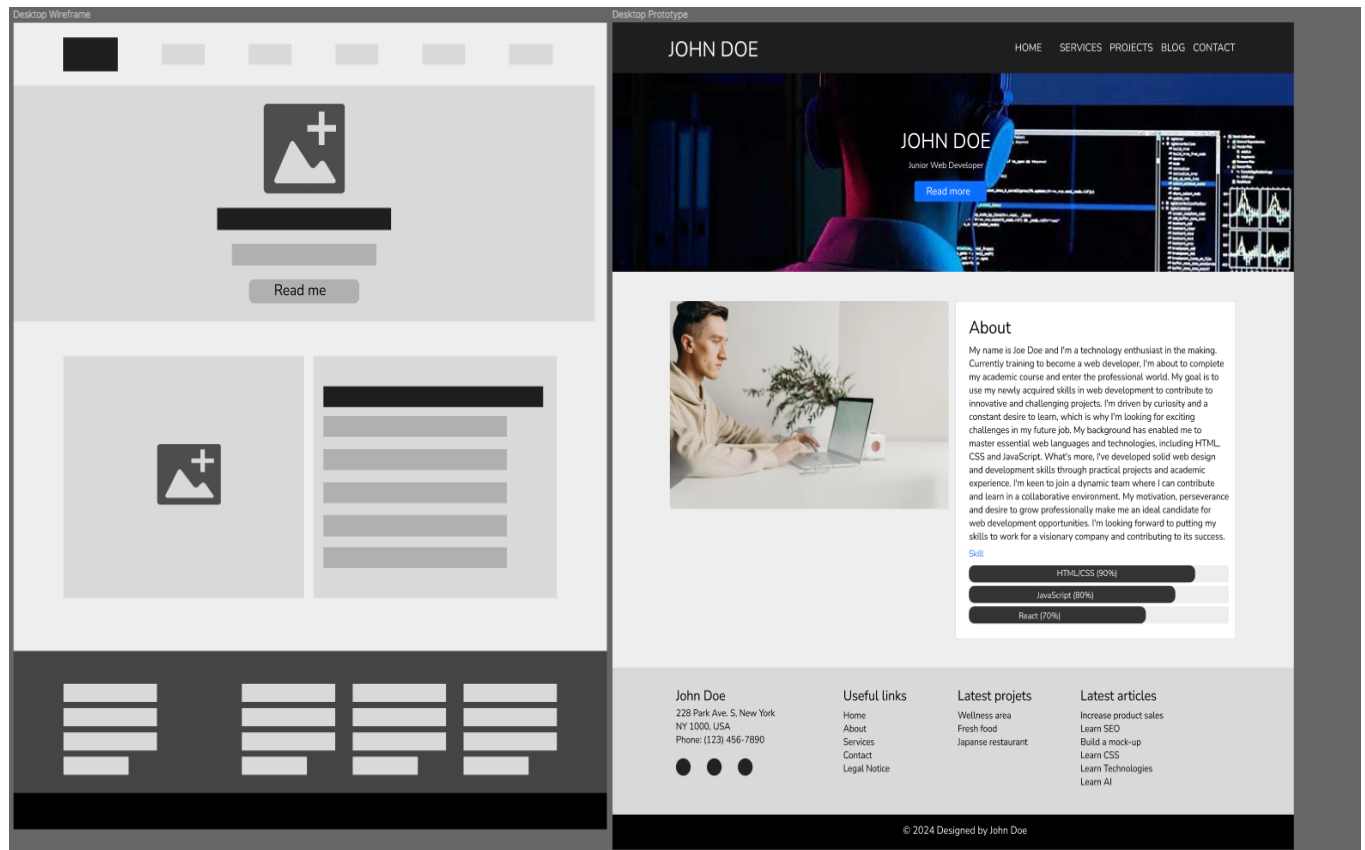
- L'utilisation de flèche représentative, il représente le menu mobile cliquable avec ouverture de plusieurs liens cliquables.
- J'ai choisi d'afficher sur la première page la représentation du menu mobile, ainsi que les événements qui permettent de l'afficher ou de le casher. Cette représentation du menu mobile n'apparaît pas sur les wireframes.

Wireframe de la page et prototype « Home » - format tablette :

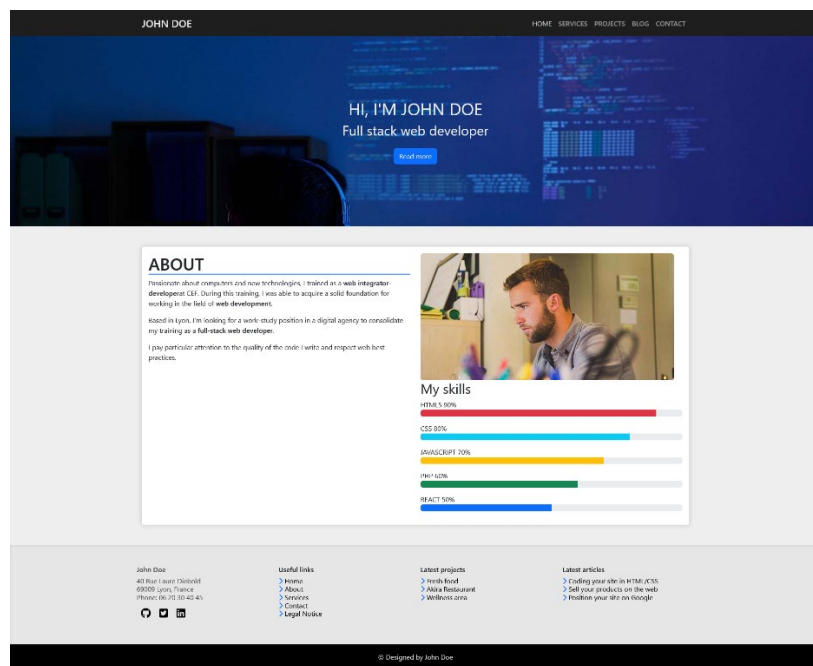


DOSSIER PROFESSIONNEL (DP)

Wireframe de la page « Home » et prototype - format desktop :

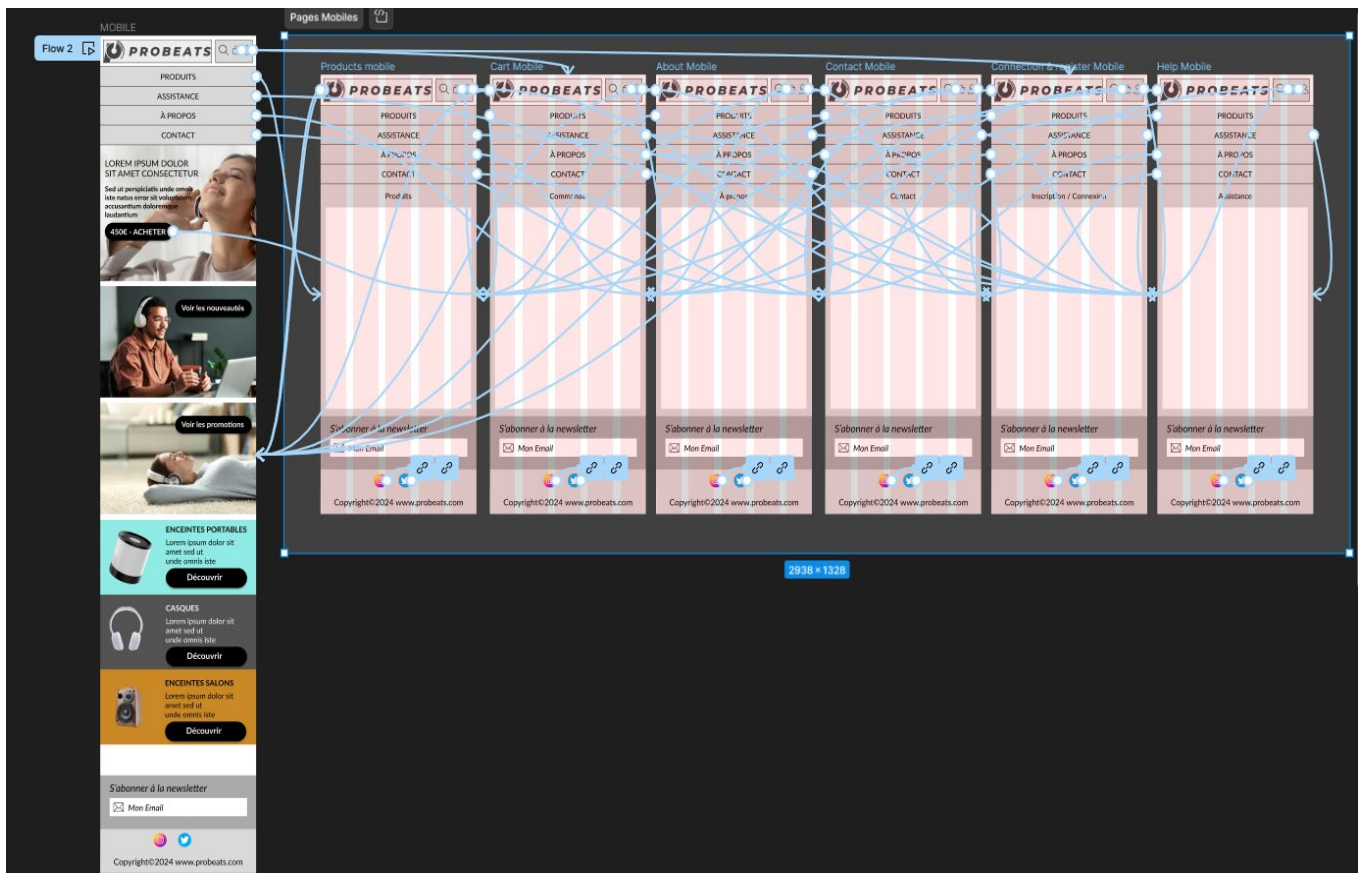
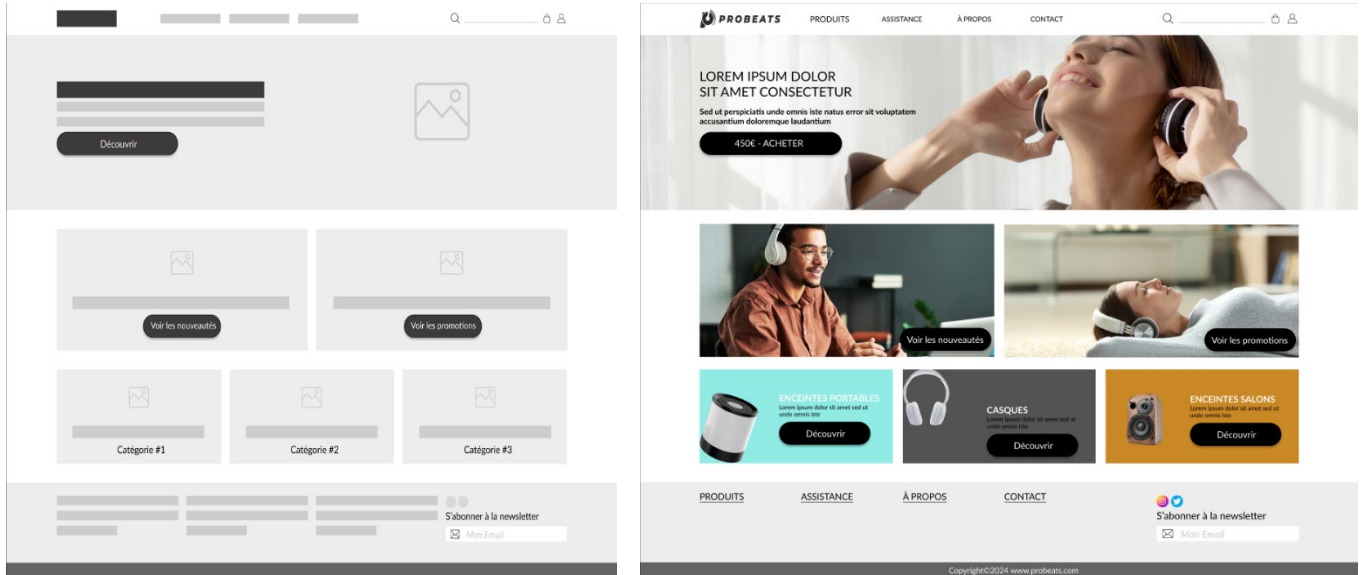


Nouvelles maquettes Prototype de la page « Home » - format desktop :



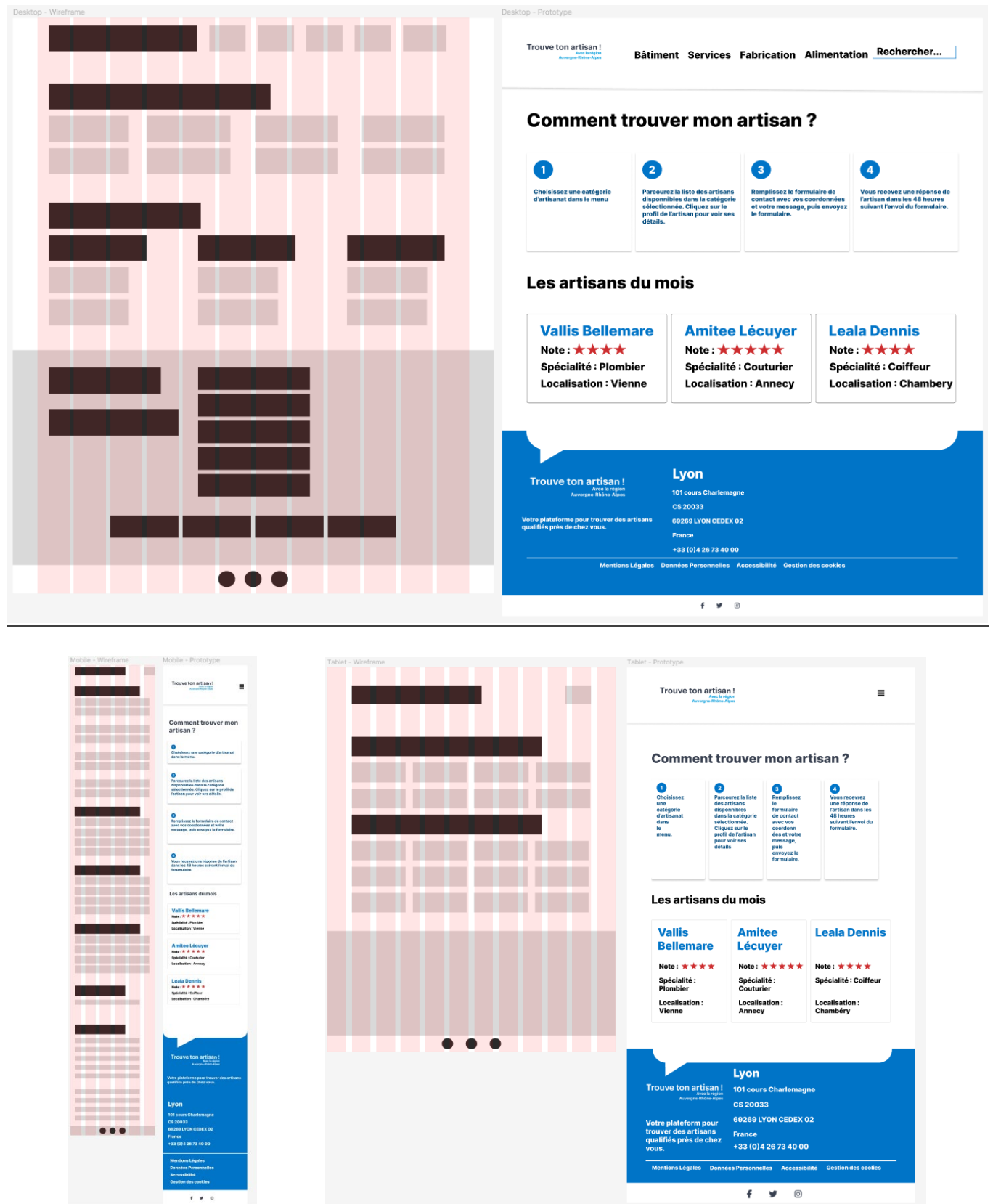
DOSSIER PROFESSIONNEL (DP)

Autres exemples de wireframes projet (PROBEATS) :



DOSSIER PROFESSIONNEL (DP)

Autres exemples de wireframes projet (Trouve ton artisan) :



DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé [Figma](#).

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : *19/02/2024* au : *24/02/2024*

5. Informations complémentaires (facultatif)

Le repo de ce projet est accessible à l'adresse suivante : <https://github.com/devlivepro/portfolio>
L'espace de travail Figma est disponible à cette adresse : [Lien](#)

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 2 ► Réaliser une interface statique et adaptable

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

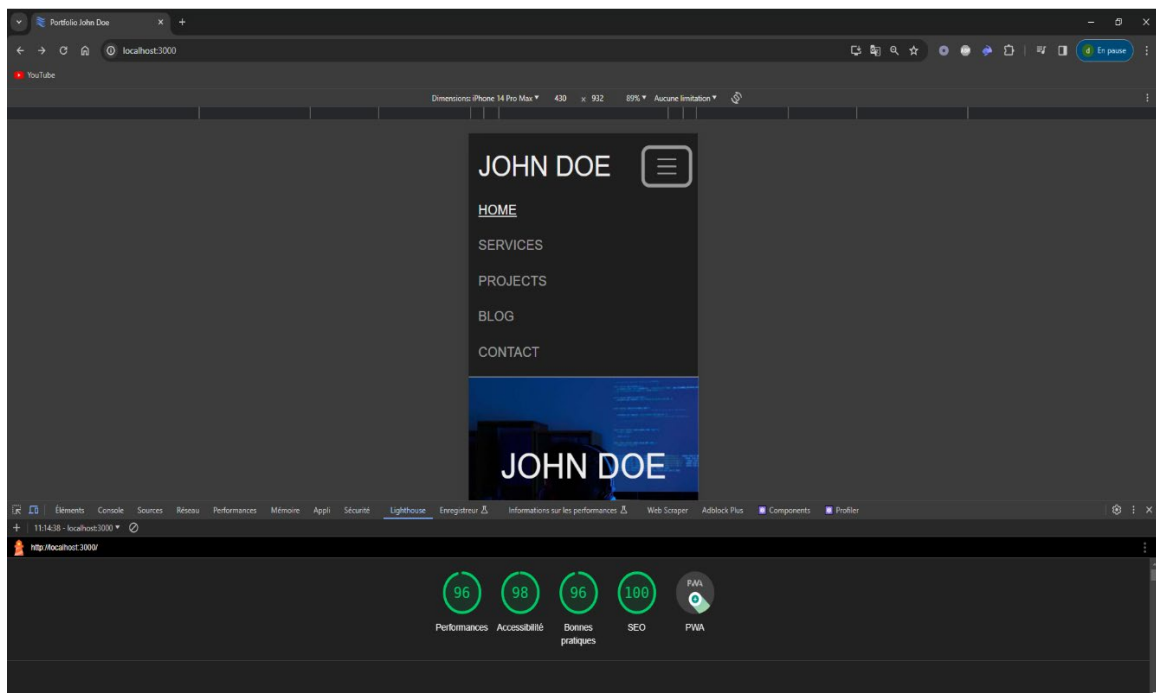
En reprenant les spécifications du projet (CP1), définies par les demandes du client, l'établissement des users stories et des wireframes, je suis passé à la réalisation du projet.

Ce projet définissait plusieurs objectifs. Le premier d'entre eux était d'être accessible et d'utiliser des bonnes pratiques de SEO. Le second objectif étant de réaliser un site responsive, accessible aux formats mobiles, tablettes et desktop.

La réalisation de ce projet se veut d'être le plus simple mais complet. J'ai choisi d'utiliser du HTML, du CSS avec comme framework Bootstrap. J'utilise cependant la librairie minify.css pour l'optimisation. Concernant l'utilisation de JavaScript, j'ai fait le choix d'en utiliser au minimum. Dans ce projet, Javascript est utilisé pour gérer l'ouverture et la fermeture du menu sur mobile et tablette avec un peu de structuration des articles.

Pour vérifier que le projet répondait bien à ses impératifs, j'ai choisi d'utiliser l'outil Lighthouse que j'installe au préalable sur google chrome comme module complémentaire, il s'intègre par la suite à la console de développement du navigateur.

DOSSIER PROFESSIONNEL (DP)



Structure des pages :

Pour structurer les pages du projets REACT, pour favoriser l’accessibilité et le SEO, j’ai choisi d’utiliser une bonne structuration pour la page index.html.

Voici par exemple la structure générale de la page d’accueil qui est coder en anglais :

```
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">
6     <meta name="description" content="Website portfolio john doe">
7     <meta name="theme-color" content="#000000">
8     <title>Portfolio John Doe</title>
9     <link rel="icon" href="/favicon.ico">
10    <link rel="apple-touch-icon" href="/favicon.ico">
11    <link rel="manifest" href="/manifest.json">
12    <link
13      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
14      rel="stylesheet"
15      integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhJY6hW+ALEwIH"
16      crossorigin="anonymous">
17  </head>
18
19  <body>
20    <noscript>You need to enable JavaScript to run this app.</noscript>
21    <div id="root"></div>
22    <script
23      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
24      integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"
25      crossorigin="anonymous"
26    ></script>
27  </body>
28 </html>
29
```

On remarque ici que le corps de la page est divisé en deux grandes parties, la partie *head* et la partie *body*, la partie intérieure du « div id root » contiendra tout le contenu des autres pages html.

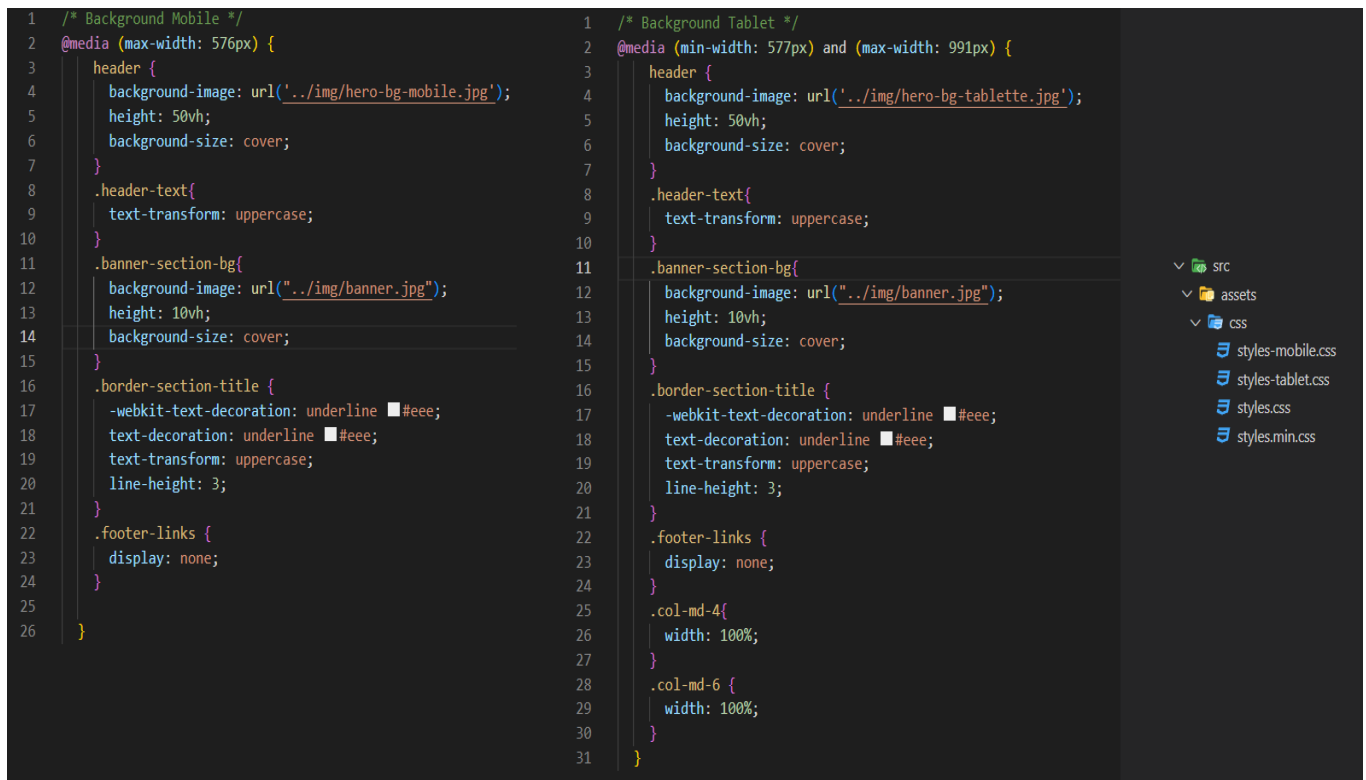
Exemple de la page « header.jsx » qui contiendra le header d'un component réutilisable :

```
src > components > Header.jsx > ...
1  import React from "react";
2
3  function Header() {
4    return (
5
6      // Bloc header
7      <header className="position-relative">
8        <div
9          className="position-absolute top-0 start-0 w-100 h-100"
10         style={{ opacity: 0.3 }}
11       ></div>
12
13       <div className="container h-100 d-flex flex-column justify-content-center align-items-center position-relative z-index-1">
14         <p class="text-center color-official4 size1">John Doe</p>
15         <p className="text-center color-official4 size2">
16           Junior Web Developer
17         </p>
18         <a href="#about" className="btn btn-primary btn-jd mt-3">
19           Read more
20         </a>
21       </div>
22     </header>
23
24     // End bloc header
25   );
26 }
27
28 export default Header;
29
```

On remarque dans la structure quatre éléments importants, un import react tous en haut, une fonction qui créera un header pour identifier les informations principales et un export header en bas de la page, pour un bon fonctionnement.

Responsive et media queries :

Pour suivre la demande d'avoir un styles responsive, et en suivant le mobile-first, j'ai utilisé des media queries. Pour avoir un site qui s'adapte sur trois formats différents comme le mobile, tablette et desktop. J'utilise [minifier](#) pour réduire la charge du « styles.css » en « styles.min.css ».



Le styles.css est adapté pour le desktop ayant une largeur minimum de 1024px (desktop).

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisé VSCode avec **Prettier** pour formater le code.
Pour vérifier l'accessibilité et les bonnes pratiques de SEO, j'ai utilisé l'outil **Lighthouse** de la console de développement de mon navigateur (Google Chrome). J'ai également utilisé la console pour simuler les media queries.
Je me suis référé au site MDN (Mozilla Developer Network) pour mieux utiliser les media queries.
Pour déployer le site, j'ai utilisé le système des **Github Pages**, puis j'ai utilisé l'outil **Google Search Console** pour permettre l'indexation du site.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : 27/02/2024 au : 03/03/2024

5. Informations complémentaires (facultatif)

Le repo du projet est accessible ici : <https://github.com/devlivepro/johndoe>

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 3 ► Développer une interface utilisateur web dynamique

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour ce projet, j'ai réalisé une application web qui se nomme portfolio. Ce site permet de visualiser le parcours d'un programmeur web junior. Cette application web à page unique (SPA) développée en utilisant la bibliothèque JavaScript React pour la création d'interfaces utilisateur dynamiques et réactives. Cette application se charge initialement avec une seule page HTML, puis utilise JavaScript pour charger dynamiquement le contenu au fur et à mesure que l'utilisateur navigue à travers l'application

Pour réaliser la partie front-end de l'application web, j'ai utilisé la librairie React en y ajoutant le module suivant :

- Gestion des routes : [React Router](#)

La page index.html sert de point d'entrée de notre application web. Elle est le premier élément chargé par le navigateur et contient l'ensemble de la structure de base de notre application. Dans cette page, nous référençons le fichier App.js, qui agit comme le cœur dynamique de notre application, offrant une interface utilisateur interactive et réactive.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="description" content="Portfolio Développeur WEB">
6     <link rel="icon" href="/favicon.ico">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Portfolio Développeur WEB Junior</title>
9   </head>
10
11  <body>
12    <!-- Le div qui contiendra tous le App.vue -->
13    <div id="app"></div>
14    <script type="module" src="/src/main.js"></script>
15  </body>
16 </html>
```

À l'intérieur de App.js, nous avons un élément <div> avec l'attribut id défini comme "app". Ce <div> est le bloc central de réception de notre application. C'est là que tout le contenu dynamique sera injecté et rendu par React. En d'autres termes, c'est le point focal autour duquel notre application se construit et se développe.

La structure de notre application repose sur la capacité de ce <div> à recevoir et à afficher les composants et les données appropriés en fonction de l'interaction de l'utilisateur et de l'état actuel de l'application. C'est là que les différents composants, les fonctionnalités et les vues de notre application prendront vie, créant ainsi une expérience utilisateur immersive et intuitive.

En résumé, le <div id="app"> de notre fichier index.html représente le noyau de notre application web, où la magie de React opère, permettant à notre application de prendre forme et d'offrir une expérience utilisateur riche et dynamique.

```
1  <script setup>
2  import { RouterLink, RouterView } from "vue-router";
3  import Header from "@components/Header.vue"; // Import components
4  import NavBar from "@components/NavBar.vue"; // Import components
5  import Footer from "@components/Footer.vue"; // Import components
6  </script>
7
8  <template>
9    <div id="app">
10      <!-- Barre de Navigation -->
11      <Header />
12
13      <NavBar />
14
15      <!-- Contenu de la page -->
16      <main>
17        <router-view></router-view>
18      </main>
19
20      <!-- Composant Footer -->
21      <Footer />
22    </div>
23  </template>
```

Pour créer le routeur dans l'application, j'utilise React Router et sa fonction « *createRouter* ». Dans un premier temps, nous définissons les routes principales de l'application. Ces routes seront ensuite rendues dans le composant App.

Le router :

```
1  import { createRouter, createWebHistory } from 'vue-router'
2  import HomeView from '../views/HomeView.vue'
3  import Dev from '../views/Dev.vue'; // Import de la nouvelle page
4  import Project from '../views/Project.vue'; // Import de la nouvelle page
5  import Contact from '../views/Contact.vue'; // Import de la nouvelle page
6  import Page404 from '../views/Page404.vue'; // Import de la nouvelle page 404
7
8  const router = createRouter({
9    history: createWebHistory(import.meta.env.BASE_URL),
10   routes: [
11     {
12       path: '/',
13       name: 'home',
14       component: HomeView
15     },
16     {
17       path: '/dev',
18       name: 'dev',
19       component: Dev
20     },
21     {
22       path: '/project',
23       name: 'project',
24       component: Project
25     },
26     {
27       path: '/contact',
28       name: 'contact',
29       component: Contact
30     },
31     {
32       path: '/*',
33       name: 'Page404',
34       component: Page404
35     }
36   ]
37 })
38
39 export default router
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour réaliser ce projet j'ai utilisé [Vite](#), ainsi que les documentations des différentes dépendances de mon projet, notamment les documentations de React Router.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : *02/02/2024* au : *05/02/2024*

5. Informations complémentaires (facultatif)

Le repo de ce projet est accessible à cette adresse : <https://github.com/devlivepro/portfolio>

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 4 ► Réaliser une interface utilisateur ou avec une solution de gestion de contenu ou e-commerce

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Projet :

Dans le cadre de ce projet, la société « La vie des plantes » est une entreprise française qui vend des kits de plantes à faire pousser des graines, elle souhaite mettre en place un site internet, avec une solution e-commerce pour pouvoir diversifier ses clients et augmenter son chiffre d'affaires.

Après avoir eu un rapide échange par mail avec la société « La vie des plantes » le client souhaite un site e-commerce voulant une solution rapide et tous de même réduire ses couts au maximum, le site fonctionnera sous Wordpress une solution gratuite CMS, un thème a été définie par le client qui se nomme « Flower Shop Lite » gratuit au téléchargement depuis le store WordPress.

Le client souhaite une personnalisation complète et souhaite que son site présente les caractéristiques communes des sites de ventes de fleurs, quatre menus, une présentation de l'entreprise, un formulaire de contact, une map google avec la localisation, une boutique avec produits mis en avant sur la page d'accueil, l'hébergement sera O2switch avec un nom de domaine définie avec le client « laviedesfleurs.fr », il souhaite m'être en ligne rapidement son site e-commerce.

User Storie :

Suites à cela, pour valider les demandes du client, j'ai rédigé les users stories dans un tableau pour valider les actions que va pouvoir réaliser un visiteur sur le site.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisé **Filezilla** pour le transfert de fichiers.

Pour la partie installation, j'ai utilisé **Wordpress** pour la partie CMS, modules, thèmes, et solution e-commerce.

Pour l'hébergement, j'ai utilisé **O2switch** qui représente l'hébergement.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : *12/03/2024* au : *14/03/2024*

5. Informations complémentaires (facultatif)

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 1 ► Crée une base de données

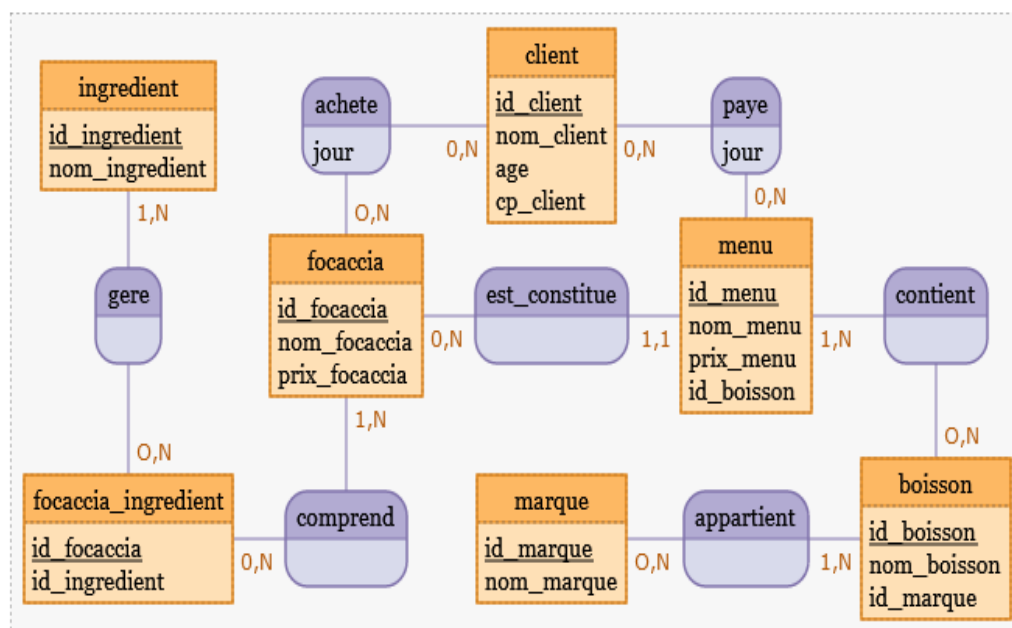
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai créé un petit gestionnaire MySQL pour la gestion des menus d'un restaurant nommé street-food italien « Tifosi ».

En premier lieu, il devait être possible pour un utilisateur de choisir son menu focaccia « pizza » et la boisson.

Pour mieux comprendre la structure de mon projet, j'ai donc créé un MCD « Modèle Conceptuel des Données ».

Le MCD :



On peut remarquer sur ce MCD les différentes cardinalités entre chaque tables, par exemple :

Client : un client est décrit par les attributs id_client (identifiant du client), nom_client (nom du client), age (âge du client) et cp_client (code postal du client).

Achete : relation entre un client et une focaccia, avec la spécification de la date (jour) à laquelle l'achat a été effectué. Un client peut acheter plusieurs focaccias (0N), et chaque achat est lié à un seul client (0N).

DOSSIER PROFESSIONNEL (DP)

Paye : Relation entre un client et un menu, avec la spécification de la date (jour) à laquelle le paiement a été effectué. Un client peut payer pour plusieurs menus (ON), et chaque paiement est associé à un seul client (ON).

Focaccia : une focaccia est décrite par les attributs id_focaccia (identifiant de la focaccia), nom_focaccia (nom de la focaccia) et prix_focaccia (prix de la focaccia).

Est Constitué : Relation entre un menu et des focaccias (11). Chaque menu est constitué de plusieurs focaccias (ON), et chaque focaccia peut être présente dans un seul menu (1).

Menu : un menu est décrit par les attributs id_menu (identifiant du menu), nom_menu (nom du menu), prix_menu (prix du menu) et id_boisson (identifiant de la boisson associée).

Contient : Relation entre un menu et des boissons (1N). Chaque menu contient plusieurs boissons (ON), et chaque boisson peut être présente dans un seul menu (ON).

Boisson : une boisson est décrite par les attributs id_boisson (identifiant de la boisson), nom_boisson (nom de la boisson) et id_marque (identifiant de la marque associée).

Appartient : relation entre une boisson et une marque (1N). Chaque boisson appartient à une seule marque (1), et chaque marque peut avoir plusieurs boissons (ON).

Marque : Une marque est décrite par les attributs id_marque (identifiant de la marque) et nom_marque (nom de la marque).

Focaccia_Ingredient : table d'association entre focaccias et ingrédients, décrivant les ingrédients utilisés dans chaque focaccia. Chaque entrée spécifie l'association entre un id_focaccia et un id_ingredient.

Comprend : relation entre une focaccia et des focaccia_ingredient (1N). Chaque focaccia comprend plusieurs ingrédients (ON), et chaque focaccia_ingredient est associé à une seule focaccia (1).

Ingredient : un ingrédient est décrit par les attributs id_ingredient (identifiant de l'ingrédient) et nom_ingredient (nom de l'ingrédient).

Gère : relation entre un focaccia_ingredient et un ingrédient (ON). Chaque focaccia_ingredient est associé à un ingrédient unique (1N), et chaque ingrédient peut être utilisé dans plusieurs focaccias (ON).

Après avoir défini la représentation visuelle de ma base de données avec le MCD, j'ai créé un MLD (modèle logique des données) pour représenter les relations entre les tables.

Le MLD :

- achete (#id_client, #id_focaccia, jour)
- boisson (id_boisson, nom_boisson, id_marque)
- client (id_client, nom_client, age, cp_client)
- focaccia (id_focaccia, nom_focaccia, prix_focaccia)
- focaccia_ingredient (id_focaccia, id_ingredient)
- ingredient (id_ingredient, nom_ingredient)
- marque (id_marque, nom_marque)
- menu (id_menu, nom_menu, prix_menu, id_boisson, #id_focaccia)
- paye (#id_client, #id_menu, jour)

J'ai ensuite créé un **dictionnaire des données** pour représenter les champs des tables.

Le dictionnaire des données :

Pour coller au plus près à la forme finale de ma base de données, j'ai choisi de traduire les champs des tables définis en français dans mon MCD vers l'anglais pour le dictionnaire des données.

Entité : Client

Field	Type	Details	Description
id_customer	Integer	Primary Key, Not Null	Customer's unique identifier.
name_customer	Varchar	Not Null	Customer name.
age	Integer	Not Null	Customer age.
cp_customer	Varchar	Not Null	Customer postal code.

Entité : Achete

Field	Type	Details	Description
id_purchase	Integer	Primary Key, Not Null	Unique customer identifier.
id_customer	Integer	Foreign Key, Not Null	Foreign key linked to the customer.
id_focaccia	Integer	Foreign Key, Not Null	Foreign key linked to focaccia.
day	Date	Not Null	Purchase date (day).

Entité : Menu

Field	Type	Details	Description
id_menu	Integer	Primary Key, Not Null	Unique menu identifier.
name_menu	Varchar	Not Null	Menu name.
price_menu	Decimal	Not Null	Menu price.
id_boisson	Integer	Foreign Key	Beverage-related foreign key.

Entité : Boisson

Field	Type	Details	Description
id_drink	Integer	Primary Key, Not Null	Unique drink identifier.
nom_drink	Varchar	Not Null	Beverage name.
id_brand	Integer	Foreign Key	Foreign key linked to the brand.

DOSSIER PROFESSIONNEL (DP)

Entité : Marque

Field	Type	Details	Description
id_brand	Integer	Primary Key, Not Null	Unique brand identifier.
nom_brand	Varchar	Not Null	Brand name.

Entité : Focaccia

Field	Type	Details	Description
id_focaccia	Integer	Primary Key, Not Null	Unique identifier for focaccia.
name_focaccia	Varchar	Not Null	Name of focaccia.
price_focaccia	Float	Not Null	Price of focaccia.

Entité : Ingredient

Field	Type	Details	Description
id_ingredient	Integer	Primary Key, Not Null	Unique ingredient identifier.
name_ingredient	Varchar	Not Null	Ingredient name.

Puis une fois le dictionnaire des données réalisé, je me suis servi de phpMyAdmin pour réaliser le MPD (modèle physique des données).

Le MPD :

Veillez noter : pour des raisons d'organisation, les extraits de code sont séparés.

```
26  --
27  -- Structure de la table `achete`
28  --
29
30  • CREATE TABLE `achete` (
31      `id_achat` int(11) NOT NULL,
32      `id_client` int(11) DEFAULT NULL,
33      `jour` date DEFAULT NULL,
34      `id_menu` int(11) NOT NULL
35  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
36
37  -----
38
39  --
40  -- Structure de la table `boisson`
41  --
42
43  • CREATE TABLE `boisson` (
44      `id_boisson` int(11) NOT NULL,
45      `nom_boisson` varchar(45) NOT NULL,
46      `id_marque` int(11) NOT NULL
47  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
48
49  --
50  -- Déchargement des données de la table `boisson`
51  --
52
53  • INSERT INTO `boisson` (`id_boisson`, `nom_boisson`, `id_marque`) VALUES
54      (1, 'Coca-cola zéro', 1),
55      (2, 'Coca-cola original', 1),
```

DOSSIER PROFESSIONNEL (DP)

```
459 -- Contraintes pour la table `achete`
460 --
461 • ALTER TABLE `achete`
462     ADD CONSTRAINT `achete_ibfk_1` FOREIGN KEY (`id_client`) REFERENCES `client` (`id_client`),
463     ADD CONSTRAINT `fk_client_achat` FOREIGN KEY (`id_client`) REFERENCES `client` (`id_client`),
464     ADD CONSTRAINT `fk_menu_achete` FOREIGN KEY (`id_menu`) REFERENCES `menu` (`id_menu`);
465
466 --
467 -- Contraintes pour la table `boisson`
468 --
469 • ALTER TABLE `boisson`
470     ADD CONSTRAINT `boisson_ibfk_1` FOREIGN KEY (`id_marque`) REFERENCES `marque` (`id_marque`),
471     ADD CONSTRAINT `fk_marque` FOREIGN KEY (`id_marque`) REFERENCES `marque` (`id_marque`);
472
473 --
474 -- Contraintes pour la table `focaccia_ingredient`
475 --
476 • ALTER TABLE `focaccia_ingredient`
477     ADD CONSTRAINT `fk_focaccia` FOREIGN KEY (`id_focaccia`) REFERENCES `focaccia` (`id_focaccia`),
478     ADD CONSTRAINT `fk_ingredient` FOREIGN KEY (`id_ingredient`) REFERENCES `ingredient` (`id_ingredient`),
479     ADD CONSTRAINT `focaccia_ingredient_ibfk_1` FOREIGN KEY (`id_focaccia`) REFERENCES `focaccia` (`id_focaccia`),
480     ADD CONSTRAINT `focaccia_ingredient_ibfk_2` FOREIGN KEY (`id_ingredient`) REFERENCES `ingredient` (`id_ingredient`);
481
```

2. Précisez les moyens utilisés :

Pour réaliser le MCD, j'ai utilisé [Mocodo](#), puis j'ai utilisé MySQL Workbench pour m'assister dans la réalisation du MPD.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► *Projet personnel réalisé en cours de formation.*

Période d'exercice ► Du : *17/04/2024* au : *19/04/2024*

5. Informations complémentaires (facultatif)

Le repo de ce projet complet est accessible à cette adresse : [LIEN](#)

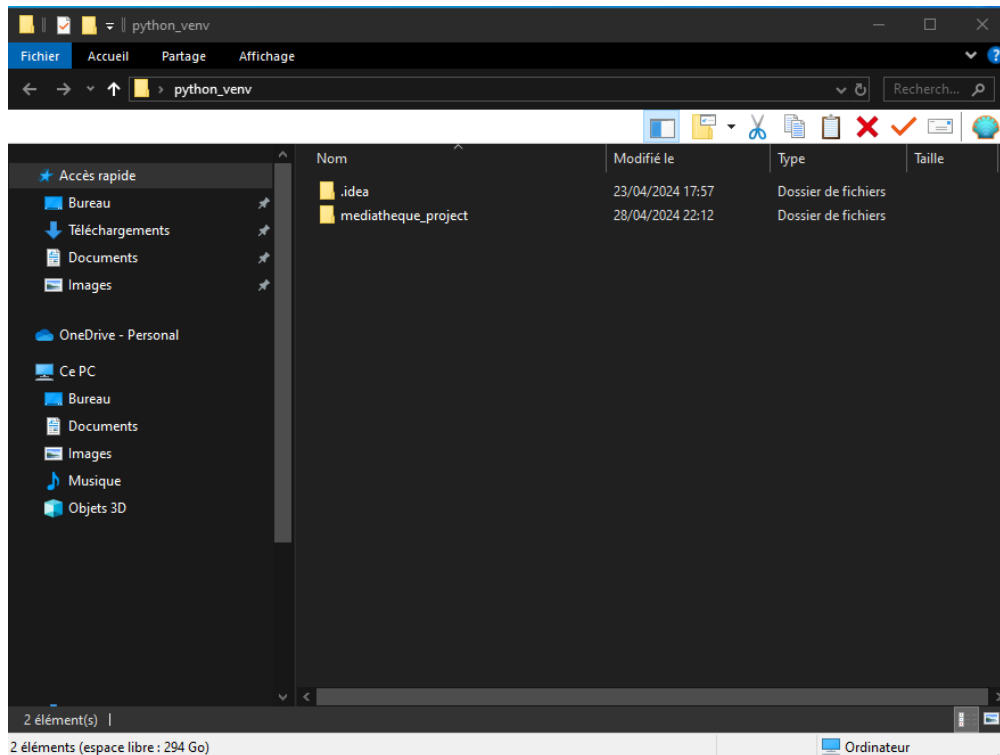
Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 2 ► Développer les composants d'accès aux données

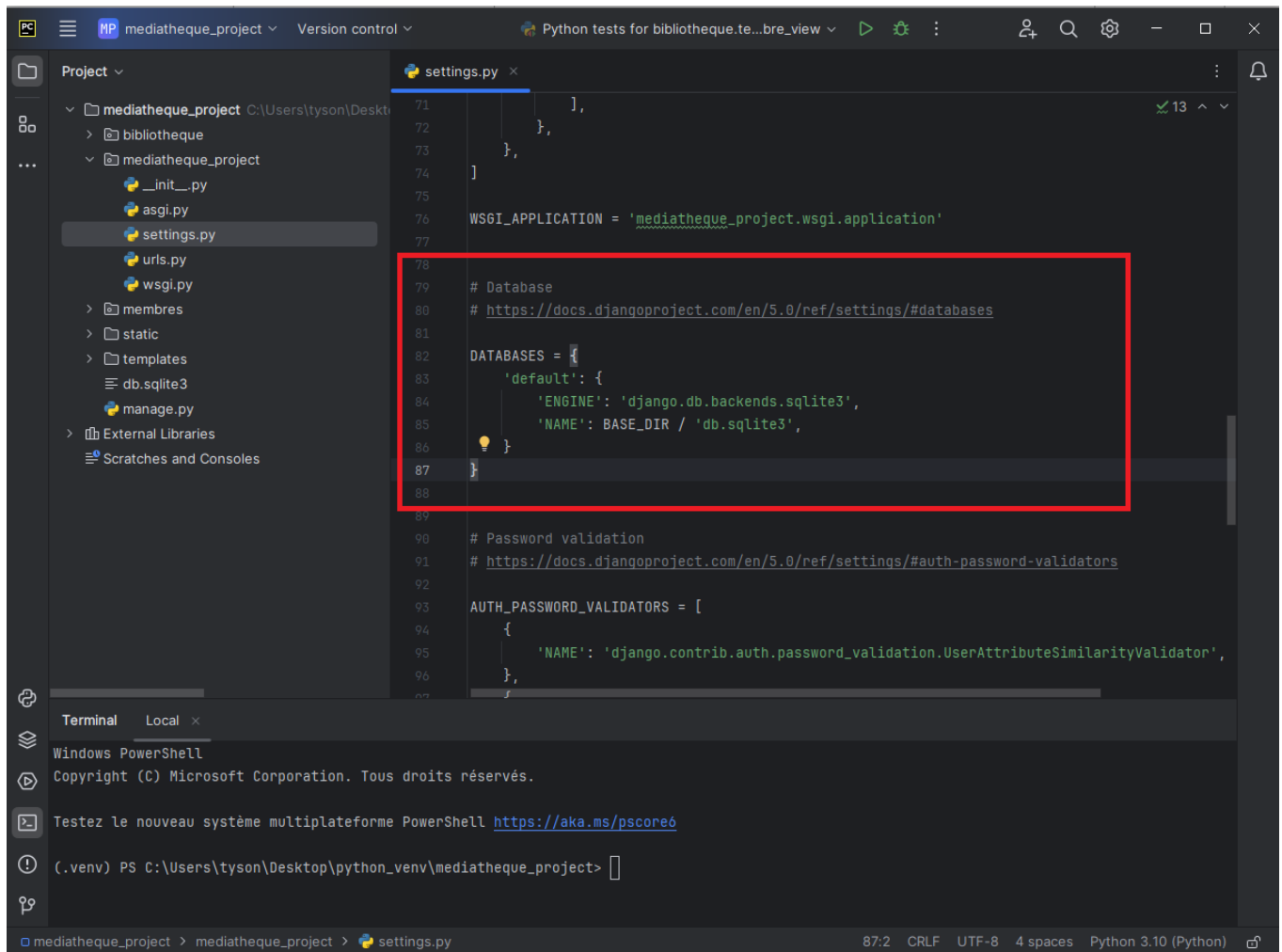
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Je commence par créer un environnement virtuel python qui se nommera `python_venv`, ensuite je crée le projet juste après le projet `mediatheque_project`.



Pour ce projet qui se nomme « mediaque_project » nous utiliseront python, le principe est de permettre à un membre de faire des emprunts de cd, dvd, livre, jeu de plateau avec des restrictions, pour la partie bibliothèque d'ajouter, supprimer, modifier en utilisant les fonctions « CRUD », l'accès à la base de données ce fait par le module `sqlite3` qui se connectera à la base de données est qui est inclus dans la bibliothèque standard, sans nécessiter l'installation de bibliothèques tierces, j'ai la possibilité d'utiliser `phpMyAdmin` pour une base de données externe, mais pour cet exemple je ne l'ai pas utilisé.

Connexion sqlite3:



The screenshot shows a code editor with the following content:

```
71     },
72     },
73     },
74 ]
75
76 WSGI_APPLICATION = 'mediatheque_project.wsgi.application'
77
78
79 # Database
80 # https://docs.djangoproject.com/en/5.0/ref/settings/#databases
81
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.sqlite3',
85         'NAME': BASE_DIR / 'db.sqlite3',
86     }
87 }
88
89
90 # Password validation
91 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators
92
93 AUTH_PASSWORD_VALIDATORS = [
94     {
95         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
96     },
97 ]
```

The code editor interface includes a sidebar with a project tree showing files like `__init__.py`, `asgi.py`, `settings.py`, `urls.py`, `wsgi.py`, `membres`, `static`, `templates`, `db.sqlite3`, and `manage.py`. The bottom panel shows a terminal window with the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

(.venv) PS C:\Users\tyson\Desktop\python_venv\mediatheque_project>
```

Pour utiliser PhpMyAdmin, je crée ma base de données correspondantes au champs spécifique de mon MCD que je crée au préalable au début de mon projet, j'installe le module :

➤ `pip install mysqlclient`

DOSSIER PROFESSIONNEL (DP)

Je change les données de connexion par :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'bibliotheque',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

Et je termine par les commandes :

- python manage.py makemigrations
- python manage.py migrate

Aperçu avec le logiciel « DB Browser for SQLite » :

The screenshot shows the DB Browser for SQLite application. The title bar indicates the file path: C:\Users\tyson\Desktop\python_venv\mediatheque_project\db.sqlite3. The interface includes a menu bar (Fichier, Édition, Vue, Outils, Aide) and a toolbar with buttons for creating a new database, opening an existing one, registering/modifying/canceling changes, opening a project, and attaching a database. Below the toolbar is a 'Structure de la Base de Données' section with buttons for creating tables, indexes, modifying tables, deleting tables, and printing. The main area displays a list of tables (20) and their corresponding SQL CREATE statements. The tables listed are: auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, bibliotheque_cd, bibliotheque_dvd, bibliotheque_emprunt, bibliotheque_jeudeplateau, bibliotheque_livre, bibliotheque_media, bibliotheque_membre, django_admin_log, django_content_type, django_migrations, django_session, membres_mediasconsulte, membres_membre, and sqlite_sequence. Each table entry is followed by its SQL definition, such as 'CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)'. The bottom of the interface shows sections for 'Index (18)', 'Vues (0)', and 'Déclencheurs (0)'.

Aperçu de l'administration Django :

The screenshot displays the Django administration interface in a web browser. The address bar shows the URL `127.0.0.1:8000/admin/`. The page header includes the text "Django administration" and a welcome message for the user "ADMIN" with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". The main content area is titled "Site administration" and is divided into two sections: "AUTHENTICATION AND AUTHORIZATION" and "BIBLIOTHEQUE".

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

BIBLIOTHEQUE

Cds	+ Add	Change
Dvds	+ Add	Change
Emprunts	+ Add	Change
Jeu de plateaus	+ Add	Change
Livres	+ Add	Change
Membres	+ Add	Change

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisé **PyCharm**.
Pour la partie base de données, j'ai utilisé « **DB Browser for SQLite** ».

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : *08/05/2024* au : *14/05/2024*

5. Informations complémentaires (facultatif)

Le repo de ce projet complet est accessible à cette adresse : [LIEN](#)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 3 ► Développer la partie back-end d'une application web ou web mobile

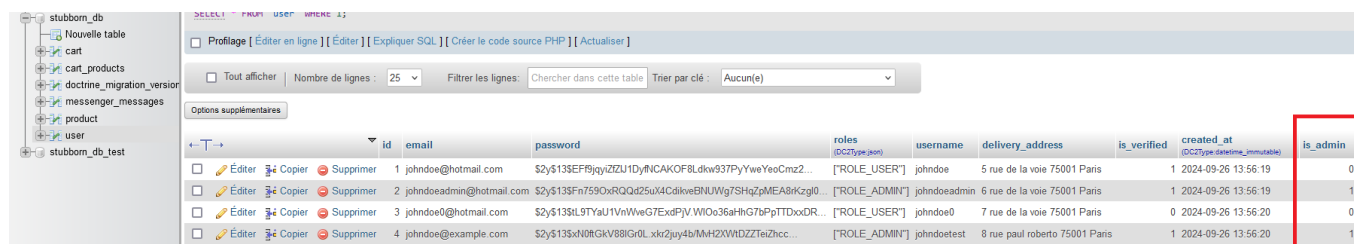
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans ce projet, j'ai développé une application web avec Symfony, qui est une plateforme de gestion, e-commerce qui se nommera : Stubborn

Je commence par installer mon projet avec toutes les dépenses nécessaires :

- symfony new stubborn --webapp
- composer install

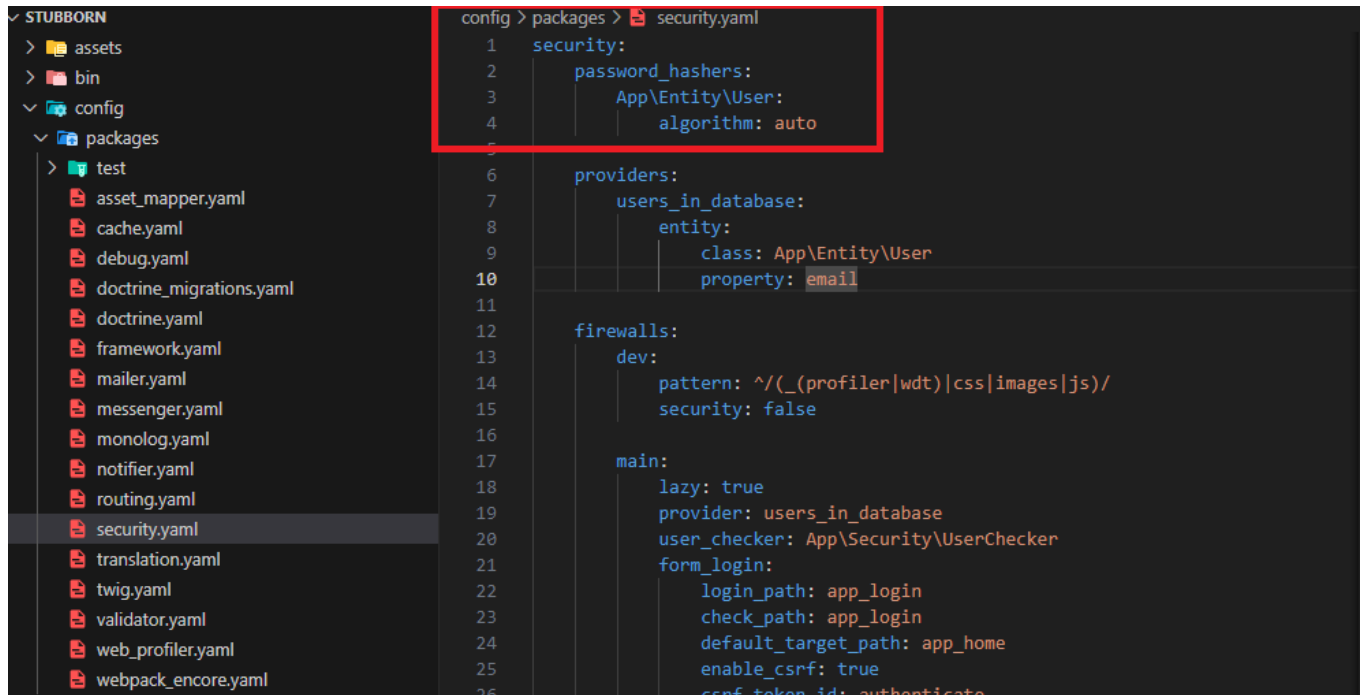
Pour la partie gestion des utilisateurs et des rôles, j'ai implémenté un système de gestion des utilisateurs avec un **is_admin** dans la base de données. Ce champ est un booléen (0 ou 1) qui détermine si un utilisateur est administrateur ou non :



	id	email	password	roles	username	delivery_address	is_verified	created_at	is_admin
<input type="checkbox"/>	1	johndoe@hotmail.com	\$2y\$13\$EF8jyZIZU1DyfnCAKOF8Ldkw937PyYweYeoCmz2...	[ROLE_USER]	johndoe	5 rue de la voie 75001 Paris	1	2024-09-26 13:56:19	0
<input type="checkbox"/>	2	johndoeadmin@hotmail.com	\$2y\$13\$F7590xRQd25uX4CdkweBNUVWg7SHgZpMEABkczg0...	[ROLE_ADMIN]	johndoeadmin	6 rue de la voie 75001 Paris	1	2024-09-26 13:56:19	1
<input type="checkbox"/>	3	johndoe0@hotmail.com	\$2y\$13\$IL9TYaU1VnWveGTEdPJv.WIOo36aHhG7bPpTDDoxDR...	[ROLE_USER]	johndoe0	7 rue de la voie 75001 Paris	0	2024-09-26 13:56:20	0
<input type="checkbox"/>	4	johndoe@example.com	\$2y\$13\$xn0tGKv88lGr0L_xkr2juy4bMhH2XWdZzTeizhcc...	[ROLE_ADMIN]	johndoe0test	8 rue paul roberto 75001 Paris	1	2024-09-26 13:56:20	1

Pour le rôle admin si la valeur de **is_admin** est 1, l'utilisateur obtient des droits supplémentaires pour accéder aux parties protégées de l'application, telle que l'administration.

Symfony utilise son composant Security pour gérer les rôles et l'accès, pour la **gestion des hashers de Mots de Passe**, voici une partie de mon code en rouge qui va nous intéresser :



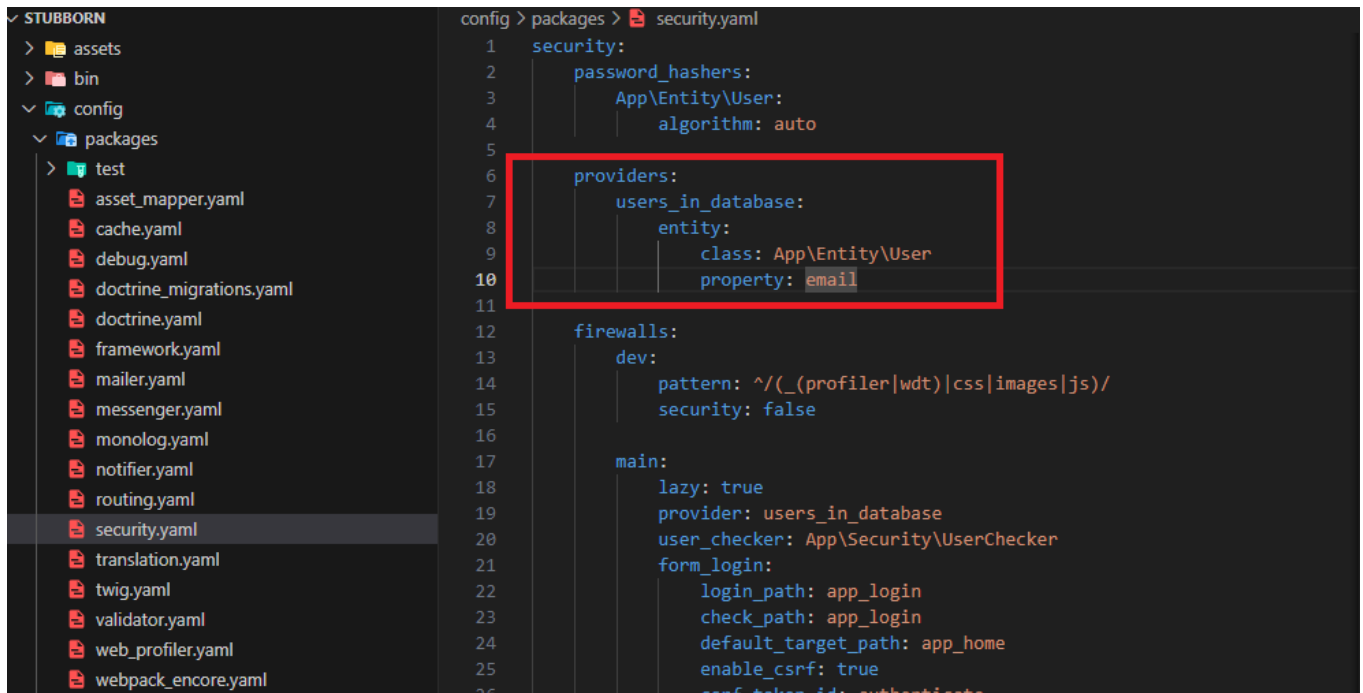
```
config > packages > security.yaml
1 security:
2     password_hashers:
3         App\Entity\User:
4             algorithm: auto
5
6     providers:
7         users_in_database:
8             entity:
9                 class: App\Entity\User
10                property: email
11
12     firewalls:
13         dev:
14             pattern: ^/(_(profiler|wdt)|css|images|js)/
15             security: false
16
17     main:
18         lazy: true
19         provider: users_in_database
20         user_checker: App\Security\UserChecker
21         form_login:
22             login_path: app_login
23             check_path: app_login
24             default_target_path: app_home
25             enable_csrf: true
26             csrf_token_id: authenticate
```

J'utilise la configuration par défaut de Symfony pour le hachage des mots de passe. Le paramètre **algorithm : auto** permet à Symfony de choisir automatiquement l'algorithme de hachage le plus sécurisé disponible comme (bcrypt ou argon2), garantissant que les mots de passe des utilisateurs sont sécurisés.

Cette configuration garantit une sécurité optimale des mots de passe, car l'algorithme de hachage utilisé est toujours à jour avec les dernières recommandations en matière de sécurité.

A noter aussi le plus important c'est que cela réduit les risques d'attaque de type brute-force ou rainbow table.

Pour la gestion des fournisseurs d'Utilisateurs, j'ai configuré Symfony pour récupérer les utilisateurs depuis la base de données en utilisant leur email comme identifiant. Cela signifie que lors de la connexion, Symfony va chercher l'utilisateur dans l'entité **User** en fonction de son adresse email.



```
1 security:
2     password_hashers:
3         App\Entity\User:
4             algorithm: auto
5
6     providers:
7         users_in_database:
8             entity:
9                 class: App\Entity\User
10                property: email
11
12     firewalls:
13         dev:
14             pattern: ^/(_(profiler|wdt)|css|images|js)/
15             security: false
16
17     main:
18         lazy: true
19         provider: users_in_database
20         user_checker: App\Security\UserChecker
21         form_login:
22             login_path: app_login
23             check_path: app_login
24             default_target_path: app_home
25             enable_csrf: true
26             csrf_token_id: authenticate
```

J'ai mis en place un système d'authentification sécurisé basé sur l'email, garantissant une gestion fiable et sécurisée des utilisateurs.

Ce mécanisme est flexible, car je peux changer le champ d'identification (par exemple, par **username**) si nécessaire.

Pour les firewalls et gestion de la connexion utilisateur, j'ai un formulaire de connexion sécurisé qui est géré via un formulaire (route **app_login**), et j'ai activé la **protection CSRF**. Cette protection est cruciale pour empêcher les attaques **Cross-Site Request Forgery**.

Pour la déconnexion, j'ai configuré un chemin pour la déconnexion sécurisée avec la route **app_logout**, ce qui permet de mettre fin à la session utilisateur de manière sécurisée.


La configuration d'une **déconnexion sécurisée** assure que les sessions utilisateurs sont correctement invalidées après la déconnexion, réduisant les risques de **vol de session**.

J'ai mis en place un contrôle d'accès par rôles (Access Control), des règles de contrôle d'accès qui restreignent l'accès à certaines parties de l'application. Ici, seules les personnes ayant le rôle **ROLE_USER** peuvent accéder aux pages commençant par **/products** ou **/product**.

J'ai mis un contrôle d'accès granulaire pour assurer que seuls les utilisateurs authentifiés peuvent accéder aux sections protégées.

DOSSIER PROFESSIONNEL (DP)

Pour la partie utilisateur connecté zone rouge, le menu admin et la page ne sont pas accessibles, test pour johndoe@hotmail.com :




La mode avec caractère

[Accueil](#) [Produits](#) [Panier](#) [Mon compte](#) [Se déconnecter](#)

Fourchette de prix

10 à 25 €


Filtrer



Blackbelt

29.9 €


Voir



BlueBelt

29.9 €


Voir



Street

34.5 €


Voir



Pokeball

45 €


Voir



PinkLady

29.9 €


Voir



Snow

32 €


Voir



Greyback

28.5 €


Voir



BlueCloud

45 €


Voir



BornInUsa

59.9 €

Voir



GreenSchool

42.2 €

Voir

[Retour à l'accueil](#)

Localisation

Nom de l'entreprise : Stubborn

Adresse : Piccadilly Circus, London W1J 0DA, Royaume-Uni

Téléphone : +33 1 23 45 67 89

Email : stubborn@blabla.com

Suivez-nous

Facebook

Twitter

LinkedIn

© 2024 Stubborn. Tous droits réservés.

DOSSIER PROFESSIONNEL-Version Traitement de texte – Version du 11/09/2017

Page 39

DOSSIER PROFESSIONNEL (DP)

Pour la partie admin connecté zone rouge, le menu admin et la page sont accessible, test pour johndoeadmin@hotmail.com :













La mode avec caractère

[Accueil](#) [Produits](#) [Panier](#) [Mon compte](#) [Admin](#) [Se déconnecter](#)

Bienvenue sur la page d'administration, johndoeadmin

Ajouter et gérer les produits

Image	Nom	Prix	Stock XS	Stock S	Stock M	Stock L	Stock XL	Mettre en avant	Action
<input type="text" value="Choisir un fichier"/> <input type="text" value="Aucun...r choisi"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	Ajouter
	Blackbelt	29.9 €	0	0	1	1	1	Oui	Supprimer Modifier
	BlueBelt	29.9 €	3	2	3	3	3	Non	Supprimer Modifier
	Street	34.5 €	3	3	2	3	3	Non	Supprimer Modifier
	Pokeball	45 €	3	3	3	3	3	Oui	Supprimer Modifier
	PinkLady	29.9 €	3	3	3	3	3	Non	Supprimer Modifier
	Snow	32 €	3	3	3	3	3	Non	Supprimer Modifier
	Greyback	28.5 €	2	3	3	3	3	Non	Supprimer Modifier
	BlueCloud	45 €	3	3	3	3	3	Non	Supprimer Modifier
	BornInUsa	59.9 €	3	3	3	3	3	Oui	Supprimer Modifier
	GreenSchool	42.2 €	2	3	3	3	3	Non	Supprimer Modifier

[Retour à l'accueil](#)

Localisation

Nom de l'entreprise : Stubborn
Adresse : Piccadilly Circus, London W1J 0DA, Royaume-Uni
Téléphone : +33 1 23 45 67 89
Email : stubborn@blabla.com

Suivez-nous

[Facebook](#)
[Twitter](#)
[Linkedin](#)

© 2024 Stubborn. Tous droits réservés.

J'ai mis en place des tests fonctionnels pour vérifier que les fonctionnalités essentielles de l'application fonctionnent correctement. Ces tests automatisés s'exécutent au lancement de l'application de l'application et affichent les résultats dans la console.

Test d'ajout de produit au panier (CartTest.php). Ce test valide l'ajout d'un produit au panier pour un utilisateur connecté :

- **Connexion de l'utilisateur** : Le test commence par simuler la connexion d'un utilisateur en récupérant un utilisateur de test à partir de la base de données (johndoe@example.com).
- **Ajout d'un produit au panier** : Le test simule l'ajout d'un produit (ID = 1) de taille « M » au panier via une requête POST vers /cart/add/1.
- **Vérification du panier** : Ensuite, le test suit la redirection vers la page du panier n'est pas vide et que le produit avec l'ID correct est ajouté.

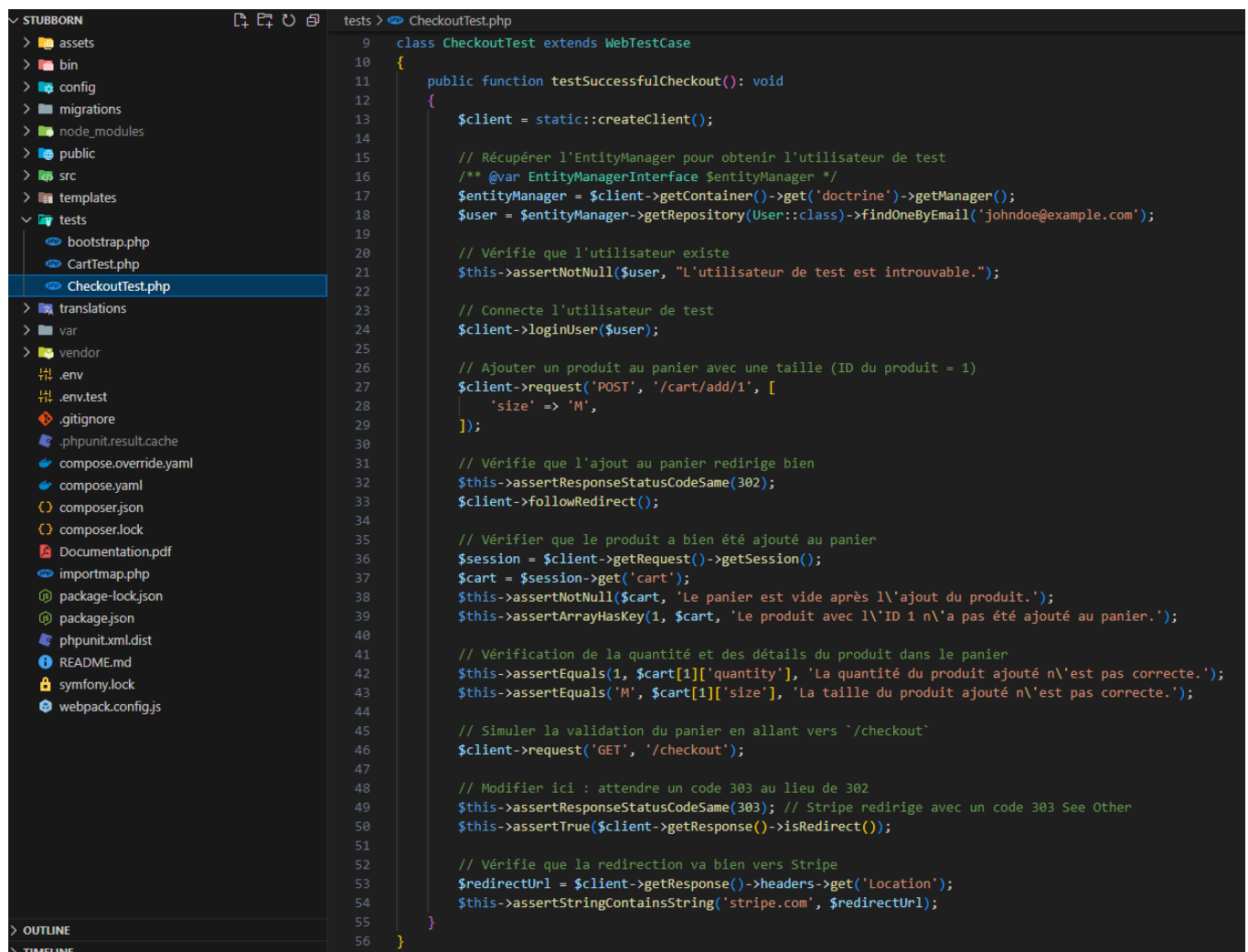
Test du processus d'achat (CheckoutTest.php). Ce test valide le processus de validation d'achat en simulant un achat pour un utilisateur connecté :

- **Ajout d'un produit au panier** : Similaire au premier test, il commence par ajouter un produit (ID = 1) au panier avec une taille spécifique (« M »).
- **Validation des détails du panier** : Le test vérifie que le produit a bien été ajouté avec les bonnes quantités et caractéristique (taille).
- **Validation du panier** : Ensuite, le test simule l'accès à la page /checkout pour démarrer le processus de validation de commande.
- **Vérification de la redirection vers Stripe** : Le test vérifie que la réponse redirige correctement vers Stripe pour le paiement, avec un code de réponse **303** (indiquant une redirection vers un autres service pour traiter le paiement).

Test du processus de validation d'achat. Pour garantir que les utilisateurs peuvent finaliser leurs achats en toute sécurité, j'ai mis en place un test fonctionnel qui simule un processus de validation de commande pour un utilisateur connecté :

- **Connexion utilisateur** : Le test se connecte avec un utilisateur de test déjà enregistré.
- **Ajout du produit au panier** : Il ajoute un produit (ID = 1, taille « M ») au panier via une requête POST.
- **Validation de la commande** : Le test vérifie que le produit est bien ajouté au panier avec les bonnes informations.
- **Validation de la commande** : Il simule l'accès à la page de validation de commande (`/checkout`).
- **Redirection vers Stripe** : Enfin, le test vérifie que la validation de la commande redirige bien l'utilisateur vers la plateforme Stripe pour le paiement.

Voici un extrait du code :

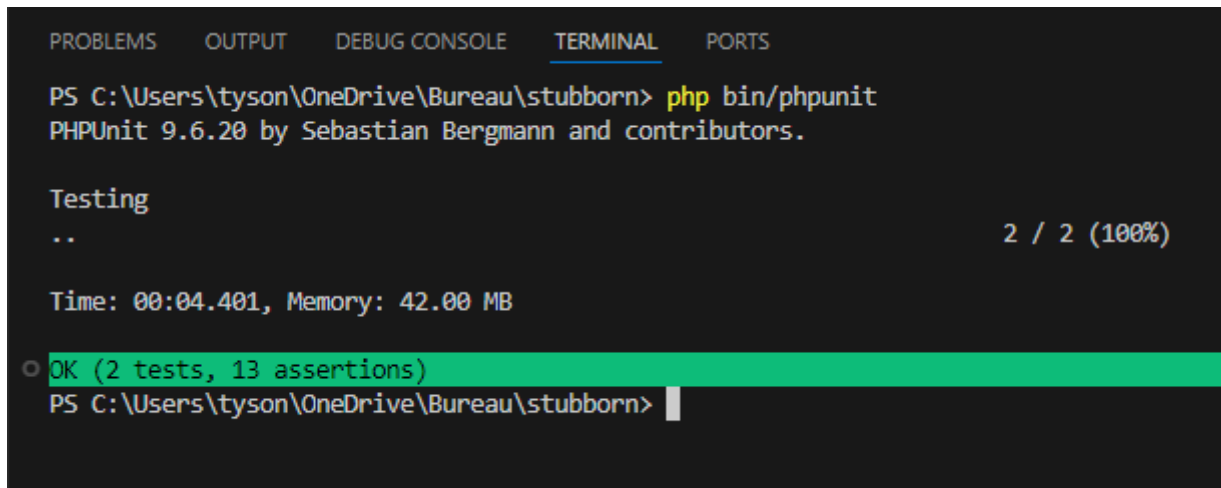


```
9 class CheckoutTest extends WebTestCase
10 {
11     public function testSuccessfulCheckout(): void
12     {
13         $client = static::createClient();
14
15         // Récupérer l'EntityManager pour obtenir l'utilisateur de test
16         /** @var EntityManagerInterface $entityManager */
17         $entityManager = $client->getContainer()->get('doctrine')->getManager();
18         $user = $entityManager->getRepository(User::class)->findOneByEmail('johndoe@example.com');
19
20         // Vérifie que l'utilisateur existe
21         $this->assertNotNull($user, "L'utilisateur de test est introuvable.");
22
23         // Connecte l'utilisateur de test
24         $client->loginUser($user);
25
26         // Ajouter un produit au panier avec une taille (ID du produit = 1)
27         $client->request('POST', '/cart/add/1', [
28             'size' => 'M',
29         ]);
30
31         // Vérifie que l'ajout au panier redirige bien
32         $this->assertResponseStatusCodeSame(302);
33         $client->followRedirect();
34
35         // Vérifier que le produit a bien été ajouté au panier
36         $session = $client->getRequest()->getSession();
37         $cart = $session->get('cart');
38         $this->assertNotNull($cart, 'Le panier est vide après l\'ajout du produit.');
39         $this->assertArrayHasKey(1, $cart, 'Le produit avec l\'ID 1 n\'a pas été ajouté au panier.');
40
41         // Vérification de la quantité et des détails du produit dans le panier
42         $this->assertEquals(1, $cart[1]['quantity'], 'La quantité du produit ajouté n\'est pas correcte.');
43         $this->assertEquals('M', $cart[1]['size'], 'La taille du produit ajouté n\'est pas correcte.');
44
45         // Simuler la validation du panier en allant vers '/checkout'
46         $client->request('GET', '/checkout');
47
48         // Modifier ici : attendre un code 303 au lieu de 302
49         $this->assertResponseStatusCodeSame(303); // Stripe redirige avec un code 303 See Other
50         $this->assertTrue($client->getResponse()->isRedirect());
51
52         // Vérifie que la redirection va bien vers Stripe
53         $redirectUrl = $client->getResponse()->headers->get('location');
54         $this->assertStringContainsString('stripe.com', $redirectUrl);
55     }
56 }
```

DOSSIER PROFESSIONNEL (DP)

Ces tests sont réalisés avec PHPUnit, un framework de test largement utilisé pour les projets PHP et Symfony.

PHPUnit permet d'automatiser les tests et d'exécuter facilement différents scénarios pour valider le bon fonctionnement de l'application. À chaque lancement de l'application, les tests s'exécutent et les résultats sont affichés dans la console.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\tyson\OneDrive\Bureau\stubborn> php bin/phpunit
PHPUnit 9.6.20 by Sebastian Bergmann and contributors.

Testing
..                                                    2 / 2 (100%)

Time: 00:04.401, Memory: 42.00 MB

OK (2 tests, 13 assertions)
PS C:\Users\tyson\OneDrive\Bureau\stubborn>
```

2. Précisez les moyens utilisés :

Framework : **Symfony**

Base de données : **Mysql**

Outils : **Doctrine ORM, Migration, Fixtures**

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► *École CEF « Centre Européen de formation »*

Chantier, atelier, service ► **Projet personnel réalisé en cours de formation.**

Période d'exercice ► Du : **22/09/2024** au : **27/09/2024**

5. Informations complémentaires (facultatif)

Le repo de ce projet complet est accessible à cette adresse : [LIEN](#)

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 4 ► Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Nom du projet : Knowledge Learning

Application Symfony de vente de formations en ligne (cursus et leçons), avec espace personnel, paiements, suivi et certifications.

Objectifs techniques :

- Authentification sécurisée avec token CSRF et rôle
- Paiement en ligne (Stripe)
- Génération automatique de factures et certificats PDF (Dompdf)
- Suivi des leçons et gestion des progrès

J'importe le projet depuis github.

Je crée la db : `php bin/console doctrine:database:create`

Je fais les migrations avec fixtures :

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

Je charge les données avec la commande :

```
php bin/console doctrine:fixtures:load
```

Back-end :

Partie utilisateur :

Avant d'accéder à cette page, nous avons une page register qui permettra de faire la première inscription avec un hashage de mots de passe pour la sécurisation optimale, un rôle prédéfini qui est ["ROLE_USER"], un token est générer pour la validation du compte avec l'adresse email.


Après validation de l'email et connexion, l'utilisateur accède à son espace personnel développé en Symfony. Cet espace lui permet d'acheter, suivre et valider des cursus ou leçons :

Boutique :

Musique

Cursus d'initiation à la guitare

Pack complet : **50 €**

 [Acheter le pack complet](#)

Leçons à l'unité

Découverte de
l'instrument **26 €**
[Acheter](#)

Les accords et les
gammes **26 €**
[Acheter](#)

DOSSIER PROFESSIONNEL (DP)

La boutique va permettre l'achat d'un pack cursus complet qui peut contenir une ou plusieurs formations ou la possibilité d'acheter des leçons à l'unité, si une leçon est achetée, le pack complet n'est plus disponible.

L'utilisateur doit impérativement valider son nom, prénom et adresse pour pouvoir commencer à acheter, une restriction à mis en place.

Formations :

- Fonctionnalité : Suivi des formations
- Liste des leçons achetées
- Progression affichée (pourcentage)
- Validation manuelle des leçons avec bouton "Valider cette leçon"
- Déblocage du certificat une fois toutes les leçons validées

Un aperçu de tous les achats des cursus achetées ou leçon individuelle, accès au suivi est ainsi validation de cursus ou leçon ci-dessous :

Musique

Cours d'initiation à la guitare

Pack complet : **50 €**


✔ Pack acheté

Léçons à l'unité

Découverte de l'instrument	26 € Inklus
Les accords et les gammes	26 € Inklus

Cours d'initiation au piano

Pack complet : **50 €**

 Acheter le pack complet

Léçons à l'unité


Découverte de l'instrument	26 € Acheter
Les accords et les gammes	26 € Acheter

Mes Résultats

Découverte de l'instrument

Prix : 26 €

● Leçon validée


Facture

Date d'achat : 11/05/2025

Client : david coco

Adresse : rue du paulhord

Formation : Cours d'initiation à la guitare

Montant : 50,00 €

Statut : Payé

Mes Certifications

Cursus d'initiation à la guitare

Formation : Cursus d'initiation à la guitare

Délivré le 11/05/2025

 [Télécharger le certificat](#)

Mes achats

Coursus d'initiation à la guitare

Thème : Musique
Prix : 50 €
Date : 11/05/2025 à 13:55
Statut : Payé

 [Télécharger la facture](#)



CERTIFICAT DE FORMATION

Ce certificat est octroyé à :

 nom

Adresse :
 rue du paillard

Pour avoir complété la formation :

Cours d'initiation à la guitare

2009 n° 11092021

Suivi - Coursus d'initiation à la guitare

Progression : 2 / 2 leçons validées

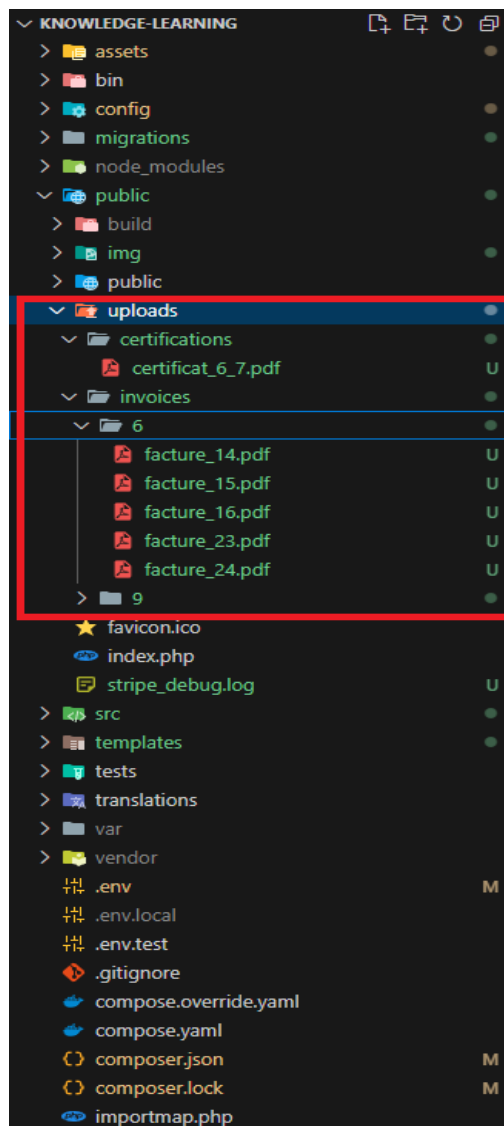
 Félicitations ! Vous avez terminé ce cursus.
Votre certification est disponible dans [Mes Certifications](#)

DOSSIER PROFESSIONNEL (DP)

Un fois que le cursus complet est validé, un certificat est généré avec dompdf, même utilisation pour une facture :

Génération de documents PDF (Dompdf) :

- Génération automatique de factures PDF après paiement
- Génération de certificats PDF après validation complète d'un cursus
- Templates HTML avec Dompdf, stockés dans uploads/
- Protection d'accès : seuls les utilisateurs propriétaires peuvent télécharger



DOSSIER PROFESSIONNEL (DP)


1 - Certificats de réussite

Mes Certifications

Cursus d'initiation à la guitare

Formation : Cursus d'initiation à la guitare

Délivré le 06/05/2025

 Télécharger le certificat



CERTIFICAT DE FORMATION

Ce certificat est décerné à :

david azerty

Adresse :

25 rue paulberg 75001

Pour avoir complété la formation :

Cursus d'initiation à la guitare

Délivré le 06/05/2025

2 - Factures PDF après achat

Mes achats


Cursus d'initiation à la guitare

Thème : Musique

Prix : 50 €

Date : 06/05/2025 à 17:23

Status : Payé

 Télécharger la facture



Facture

Date d'achat : 06/05/2025

Client : david azerty

Adresse : 25 rue paulberg 75001

Formation : Cursus d'initiation à la guitare

Montant : 50,00 €

Statut : Payé

DOSSIER PROFESSIONNEL (DP)

Connexion en tant qu'administrateur :

Un espace administrateur sécurisé a été mis en place pour gérer l'ensemble des contenus et utilisateurs du site. Accessible uniquement aux utilisateurs disposant du rôle ROLE_ADMIN, cet espace permet d'administrer la plateforme de formation.

Gestion des utilisateurs

+ Ajouter un utilisateur

Rechercher par utilisateur, nom, prénom ou email

ID	Nom d'utilisateur	Prénom	Nom	Email	Rôles	Actions
1	admin			admin@example.com	ROLE_ADMIN	Modifier Supprimer
2	student	david	coco	student@example.com	ROLE_USER	Modifier Supprimer

Affichage de tous les comptes avec les colonnes :

- ID utilisateur
- Nom d'utilisateur
- Prénom et nom
- Email
- Rôle (ROLE_USER ou ROLE_ADMIN)
- Ajout d'un nouvel utilisateur via un bouton d'action
- Recherche dynamique par nom, prénom ou email pour filtrer rapidement les comptes

Actions disponibles :

- Modifier les informations d'un utilisateur existant (nom, email, rôle...)
- Supprimer un compte si nécessaire

Sécurité :

- Pour la sécurité seuls les utilisateurs authentifiés avec le rôle ROLE_ADMIN ont accès à cette vue
- Les actions de suppression ou de modification sont sécurisées par des vérifications côté contrôleur

Modifier l'utilisateur

Nom d'utilisateur

admin

Prénom

Nom

Email

admin@example.com

Mot de passe

Confirmer le mot de passe

Rôle

Administrateur

Mettre à jour

Une interface a été développée dans la partie administrateur afin de permettre la consultation, la vérification et le téléchargement des commandes effectuées par les utilisateurs sur la plateforme.

Permettre à l'administrateur de suivre les ventes (formations ou leçons) en ligne, en facilitant :

- la lecture de l'historique des transactions
- l'accès aux factures PDF
- le contrôle du statut de paiement

Affichage d'un **tableau des commandes** avec :

- ID de la commande
- Nom + email de l'utilisateur
- Type de produit acheté (Cursus ou Leçon)
- Titre du produit
- Montant payé
- Date de la transaction
- Statut (Payé, En attente, etc.)
- Barre de recherche avec filtre dynamique par nom, email ou titre
- Bouton de téléchargement de la facture PDF associée à chaque commande
- Affichage responsive (mobile/desktop) avec Bootstrap + Twig

DOSSIER PROFESSIONNEL (DP)

Sécurité :

- Route protégée par rôle ROLE_ADMIN
- Les factures sont générées uniquement pour les commandes valides (via Stripe)
- Lien de téléchargement accessible uniquement à l'administrateur

Historique des commandes

Rechercher par utilisateur, email ou titre

ID	Utilisateur	Type	Nom	Montant	Date	Statut	Facture
7	david coco student@example.com	Cursus	Cursus d'initiation à la guitare	50 €	11/05/2025 13:55	Payé	Télécharger
8	david coco student@example.com	Cursus	Cursus d'initiation à la guitare	50 €	11/05/2025 14:07	Payé	Télécharger
9	david coco student@example.com	Cursus	Cursus d'initiation à la guitare	50 €	11/05/2025 14:16	Payé	Télécharger
10	david coco student@example.com	Cursus	Cursus d'initiation à la guitare	50 €	11/05/2025 14:28	Payé	Télécharger

Intégration de Stripe :

Objectif : Permettre aux utilisateurs d'acheter un cursus ou une leçon avec paiement sécurisé

Technologie : Stripe Checkout (API de paiement), Symfony, Webhooks

Tâches réalisées : Intégration de Stripe pour la gestion des paiements par carte bancaire

- Création dynamique de sessions Stripe avec les données produit
- Mise en place d'un webhook sécurisé pour la confirmation automatique
- Génération d'un Order uniquement après paiement validé
- Redirection vers l'application avec messages flash

Sécurité : Utilisation de STRIPE_SECRET_KEY et STRIPE_WEBHOOK_SECRET dans .env.local

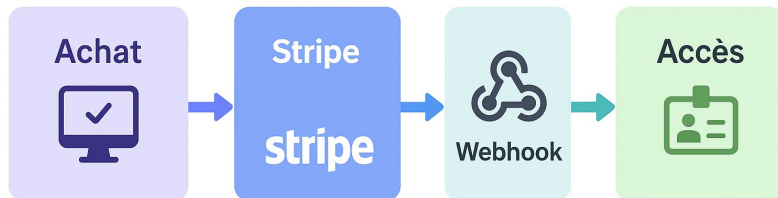
- Vérification de la signature du webhook pour empêcher les appels frauduleux

Compétences mobilisées : Développement back-end sécurisé

- Intégration d'API externe (Stripe)
- Gestion des flux asynchrones via webhook
- Encapsulation de la logique métier dans les contrôleurs Symfony

Résultat obtenu : Paiement en ligne fonctionnel et sécurisé

- Création automatique des commandes après validation
- Accès immédiat aux contenus achetés par l'utilisateur



Tests Unitaires :

Pour finaliser mon projet que tous fonctionnent bien, je vais utiliser PHPUnit pour effectuer des tests pour vérifier le bon fonctionnement, j'installe avec la commande :

```
composer require --dev symfony/phpunit-bridge
```

Je crée en parallèle une nouvelle base de données qui reprend les informations de celle de base « knowledge_learning », seul l'ajout du préfixe _test sera ajouter pour éviter d'effectuer des erreurs sur la base de données de base, donc on lance la création de « knowledge_learning_test » :

```
php bin/console doctrine:database:create --env=test
```

Par la suite je charge les données dans la db _test avec fixture :

```
php bin/console doctrine:fixtures:load --env=test
```

Pour terminer, je peux lancer les lignes de commandes tests suivantes :

```
php bin/phpunit tests/Controller/SecurityControllerTest.php
php bin/phpunit tests/Controller/StudentControllerTest.php
```

DOSSIER PROFESSIONNEL (DP)

Image des deux tests d'insertions réussies :

```
PS C:\Users\tyson\OneDrive\Bureau\knowledge-learning> php bin/phpunit tests/Controller/SecurityControllerTest.php
PHPUnit 9.6.21 by Sebastian Bergmann and contributors.

Testing App\Tests\Controller\SecurityControllerTest
..                                                    2 / 2 (100%)

Time: 00:00.907, Memory: 34.00 MB

OK (2 tests, 5 assertions)
PS C:\Users\tyson\OneDrive\Bureau\knowledge-learning> php bin/phpunit tests/Controller/StudentControllerTest.php
PHPUnit 9.6.21 by Sebastian Bergmann and contributors.

Testing App\Tests\Controller\StudentControllerTest
.....                                                5 / 5 (100%)

Time: 00:00.797, Memory: 40.00 MB

OK (5 tests, 13 assertions)
```

2. Précisez les moyens utilisés :

PHP 8, Symfony 7 (architecture MVC), MySQL / Doctrine ORM, Twig, Composer, Git / GitHub, Fixtures

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ► École CEF « Centre Européen de formation »

Chantier, atelier, service ► Projet personnel réalisé en cours de formation.

Période d'exercice ► Du : 01/05/2025 au : 10/05/2025

5. Informations complémentaires (facultatif)

Le repo de ce projet complet est accessible à cette adresse : [LIEN](#)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
No-Code pour les développeurs	Ecole O'clock	23/07/2024

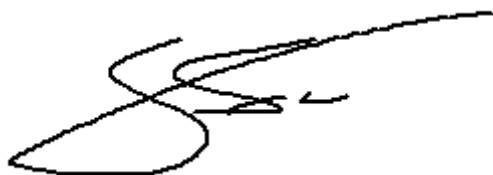
Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] DAVID SAADOUN ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à Sérignan le 16/05/2025

pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)