



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

- SERDJEBI

Nom d'usage

- SERDJEBI

Prénom

- THOMAS

Adresse

- 3 Rue Pointe à Pitre 13006 Marseille

Titre professionnel visé

Titre RNCP - Développeur Web & Web Mobile

MODALITÉ D'ACCÈS :

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

DOSSIER PROFESSIONNEL (DP)

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.

DOSSIER PROFESSIONNEL^(DP)

 <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
- Boutique en ligne	p. p.	5
- Création d'un site e-commerce via le CMS WordPress	p. p.	10
- O'Clock : site web avec une horloge, une alarme, un minuteur et un chronomètre	p. p.	14
Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	
- Camping	p. p.	22
- Autocomplétion	p. p.	28
- Boutique en ligne - parcours d'achat utilisateur	p. p.	35
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	39
Déclaration sur l'honneur	p.	40
Documents illustrant la pratique professionnelle (facultatif)	p.	41
Annexes (Si le RC le prévoit)	p.	42

DOSSIER PROFESSIONNEL (DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

Exemple n°1 - Boutique en ligne

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

Présentation du projet

Dans le cadre de ma formation j'ai été amené à réaliser une boutique en ligne fictive. Le thème de cette boutique est la vente de briquets au détail et en gros. Le site devait comporter :

- une page d'index attractive mettant en avant les produits phares (le plus populaire, les nouveautés, et les produits les mieux notés)
- une page "Produits" comportant tous les produits de la boutique en ligne, ainsi que les catégories et sous catégories, permettant un tri d'affichage des produits dans la page
- un espace client complet permettant à l'utilisateur de s'inscrire, se connecter, consulter ses informations, modifier ses informations, consulter ses commandes
- un espace administrateur complet permettant à l'administrateur de créer, consulter, modifier, supprimer les produits, les catégories et sous catégories, les commandes, consulter les utilisateurs.

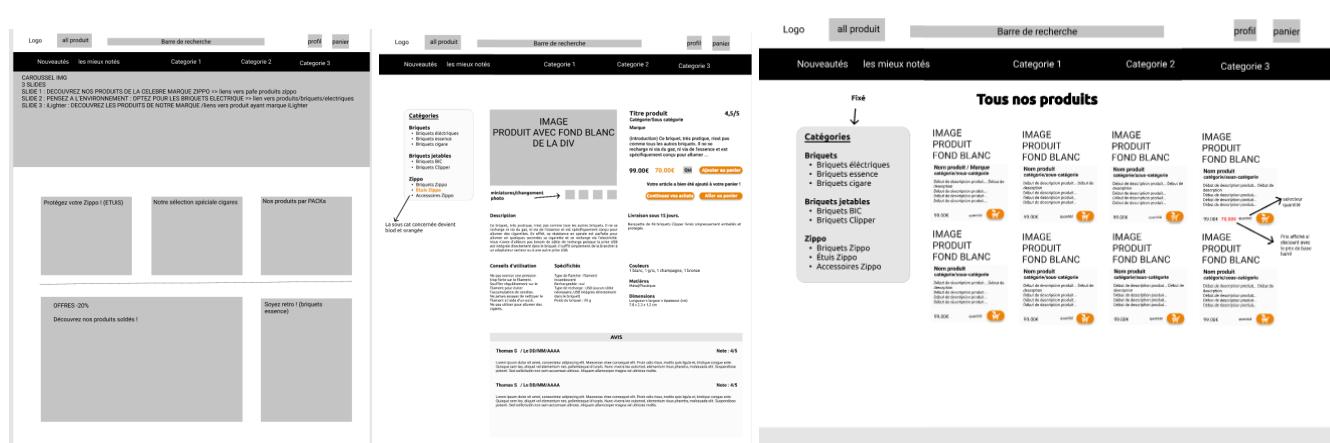
La partie front de ce projet est dynamique. Ce projet a été réalisé en groupe de 3 personnes.

Il a été réalisé avec les langages suivants : **HTML / CSS / PHP / JAVASCRIPT**.

Tâches effectuées

Maquettage du site

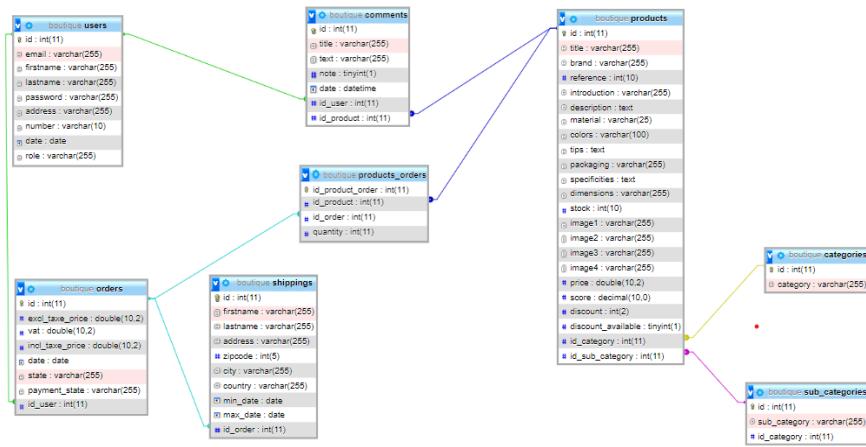
J'ai d'abord réalisé les maquettes (wireframes) basse et haute fidélité du parcours utilisateur afin d'avoir une vue d'ensemble des pages qui lui seront visibles, en définissant les couleurs principales utilisées, c'est-à-dire le blanc pour le fond, le noir pour le texte, le gris pour les blocs et l'orange pour les boutons.



DOSSIER PROFESSIONNEL (DP)

Modélisation et création de la base de données

Par la suite, j'ai modélisé la base de données. Pour cela, j'ai utilisé le site Ludichart, un environnement de travail en ligne permettant de créer des diagrammes et notamment de modéliser des bases de données. J'ai d'abord conçu le Modèle Conceptuel de Données dans un premier temps puis dans un second temps le Modèle Logique de Données. J'ai ensuite créé la base de données sur PHPMYADMIN, un système en ligne de gestion de base de données. La base de données a été créée en engine = innodb afin de maintenir l'intégrité référentielle de la base de données dont vous pouvez voir une vue dans le concepteur de PHPMYADMIN ci dessous :



Création de la page Produits

J'ai développé la page Produits présentant tous les produits ainsi qu'un bloc affichant les catégories et sous catégories. Le contenu de cette page est généré dynamiquement via le PHP et des requêtes SQL. En effet, la page inclut un fichier de traitement permettant d'appeler les fonctions de la classe Article(produits) et de retourner à la page des arrays(tableaux) contenant les informations des produits et des catégories nécessaires à l'affichage. Grâce à des boucles Foreach qui parcouruent ces tableaux, l'affichage est généré avec un code raccourci.

```
<div class="products_list">
<?php foreach ($all_products as $product) {
    echo "
<div class='product'>
    <div class='box_img'>
        <a href='produit?id=' . $product['id'] . '>'>
            <img class='p_image' src='".$product['image1']."'></a>
    </div>
    <div class='box_texte'>
        <div>$product['title'] . " / ". $product['brand'] . "</div>
        <div class='p_category'>".$product['category'] . " / ". $product['sub_category'] . "</div>
        <div class='p_intro'>".$product['introduction'] . "</div>
        <div class='price_cant'>
            <div>". $product['price'] ."€</div>
            <?if($product['stock'] > 0) { echo "
                <div><a href='addtocart?id=' . $product['id'] . '>'><img class='icone' src='/boutique-en-ligne/view/assets/add-to-basket.PNG' alt='add_to_cart_icon'></a></div>
            <?} else { echo "
                <div>Produit indisponible</div>
            <?
            <?
        </div>
    </div>
</div> ";
}
?>
</div>
```

DOSSIER PROFESSIONNEL (DP)

Le style de la page a été géré avec du CSS et des MediaQueries pour le responsive et également la méthode de mise en page flexbox pour le placement des différents blocs. Le tri du contenu se fait via un clic sur une catégorie ou une sous catégorie, permettant de filtrer les produits affichés également via les fonctions PHP de la classe Article contenant des requêtes SQL. Le fichier de traitement inclus dans la page récupère en GET dans l'URL le paramètre de la catégorie ou sous catégorie choisie afin de renvoyer à la page les objets et informations que nous souhaitions y afficher. Chaque bloc produit comporte un lien qui amène vers la page Produit avec l'id du produit choisi en paramètre dans l'URL.

Création de la page Produit

J'ai développé la page Produit affichant tous les détails d'un produit et permettant de l'ajouter à son panier. Le contenu de cette page est généré dynamiquement via la fonction de la classe Article contenant une requête SQL appelée dans le fichier de traitement inclus dans la page. Ce fichier récupère en GET dans l'URL le paramètre id du produit choisi et appelle la fonction qui permet ainsi de renvoyer à la page un tableau contenant les informations du produit choisi. L'affichage des blocs, le responsive et le style de la page ont été générés à l'aide de Bootstrap 5, permettant une adaptation à tous les écrans.

Briquet arc électrique Gentleman X3 couleurs
Les briquets arc électrique Gentleman sont efficaces et simples d'utilisation. Il suffit de les recharger en les branchant à une prise via un câble USB (fourni) sur un ordinateur ou sur un adaptateur secteur (non fourni). Vous profitez ainsi de trois briquets électriques avec un double arc électrique grâce à un allumage électronique qu'il est possible d'utiliser dans n'importe quelle condition.
10.90 €
AJOUTER AU PANIER
CONTINUER VOS ACHATS

DOSSIER PROFESSIONNEL (DP)

Création de l'espace compte de l'utilisateur et sécurisation des formulaires

J'ai développé les pages concernant le compte utilisateur (inscription, connexion, profil, mes commandes, modification des informations). Les formulaires ont été générés en HTML/CSS, le responsive de chaque page géré à l'aide des media queries afin de les adapter à tous les écrans. Les informations de l'utilisateur et de ses commandes, ainsi que le formulaire de modification du profil avec les champs pré-remplis avec les informations déjà enregistrées sont également générées à l'aide de fonctions des classes correspondantes appelées dans les fichiers de traitement des pages, renvoyant les tableaux contenant les informations nécessaires à la page. La page mon compte affiche les informations générales du profil, la dernière commande passée, ainsi que des liens pour accéder aux pages de modification des informations, du mot de passe, et la consultation de la dernière commande en détail ou de toutes les commandes.

Les formulaires sont en méthode POST et les champs sont sécurisés à l'aide de fonctions PHP htmlspecialchars() et htmlentities() permettant de transformer les caractères éligibles en éléments HTML et de protéger le site contre les injections SQL et les attaques XSS Cross Site Scripting. Ces fonctions sont utilisées pour chacune des informations rentrées de chaque champ qui sont récupérées dans le fichier de traitement inclus dans les pages.

The screenshot shows the iLighter website's user profile editing interface. At the top, there is a navigation bar with the logo 'iLighter', a search icon, and links for 'Compte', 'Déconnexion', and 'Admin'. Below the navigation, there are two main sections: 'Mon Compte' (My Account) on the left and 'Modifier les informations' (Edit Information) on the right. The 'Mon Compte' section displays a welcome message 'Bonjour admin, bienvenue sur votre espace personnel', a profile picture of a silver lighter, and two small boxes: 'Mon profil' (Profile) and 'Mes commandes' (Orders). The 'Mon profil' box contains the user's name 'admin', email 'admin@gmail.com', and ID '102030405'. The 'Mes commandes' box shows the 'Dernière commande' (Last order) with details: 'Commande numéro 140' (Order number 140), 'du 2022-04-13' (on 2022-04-13), and '13.08 euros' (13.08 euros). Below these sections are links to 'Modifier mes informations' and 'Modifier mon mot de passe'. The 'Modifier les informations' section has input fields for 'Prénom' (First name) containing 'admin', 'Nom' (Name) containing 'admin', 'Mobile' (Mobile) containing '102030405', 'Adresse' (Address) containing 'admin', 'Email' (Email) containing 'admin@gmail.com', and a 'Confirmez avec votre mot de passe' (Confirm with your password) field containing '.....'. A 'Sauvegarder' (Save) button is at the bottom right.

Création du parcours utilisateur du panier jusqu'à la confirmation de la commande

Cette partie est détaillée dans une autre section du dossier à la page 35.

Ce projet m'a permis d'acquérir les compétences de l'activité type 1 du référentiel suivantes :

- ✓ Maquetter une application
- ✓ Réaliser une interface utilisateur web statique et adaptable
- ✓ Réaliser une interface utilisateur web dynamique

Ce projet m'a permis d'acquérir les compétences de l'activité type 2 du référentiel suivantes :

- ✓ Créer une base de données
- ✓ Développer la partie back-end d'une application web ou web mobile.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

J'ai utilisé :

- Visual Studio Code pour la création et l'édition de code.
- Trello pour la planification des tâches, leur répartition entre les membres du groupe et leur suivi
- Lucidchart pour la modélisation de la base de données
- PHPMYADMIN pour la création et la gestion de la base de données
- GitKraken et GitHub pour le travail en équipe et le versionning.
- Google pour les recherches
- De la documentation en anglais : StarOverFlow.com
- De la documentation en français : OpenClassroom.com
- HTML
- CSS
- PHP
- Bootstrap 5 pour la structure de certaines pages et le responsive design
- Freelogodesign pour la création du logo
- Media queries pour le responsive design

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet avec Valentin Mathieu et Stéphane Mathieu.

4. Contexte

Nom de l'entreprise, organisme ou association - ***La Plateforme_***

Chantier, atelier, service - ***Boutique en ligne***

Période d'exercice - Du : ***10/02/2022*** au : ***10/03/2022***

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

Exemple n° 2 - *Création d'un site e-commerce via le CMS WordPress*

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

Présentation du projet

Le projet création d'un site e-commerce WordPress avait pour objectif la création d'un site fictif pour une entreprise spécialisée dans le conseil et la vente. J'ai choisi en thème la vente de montres d'exception. La boutique fictive porte le nom de "Sablier". Ce projet devait contenir une page d'accueil avec les articles mis en avant, une page produits avec les produits affichés, une page notre histoire, un formulaire de contact et un espace utilisateur complet (inscription, connexion, profil, modification du profil, commandes).

Tâches effectuées

Installation et configuration de WordPress

Dans un premier temps, j'ai dû me documenter sur le fonctionnement de WordPress, via un tutoriel Youtube orienté e-commerce. J'ai installé WordPress sur l'hébergeur Plesk. J'ai ensuite choisi le thème Astra comme conseillé dans le tutoriel. C'est un thème gratuit, personnalisable et un des plus populaires. Pour le bon fonctionnement du site, j'ai dû également ajouter des plugins à WordPress :

- Elementor, qui permet de construire ses pages WordPress
- WooCommerce, qui est une solution e-commerce open source.

L'installation des plugins sur WordPress se fait comme suit :

- panneau de configuration -> extensions -> recherche du plugin -> installation -> activation.

Création des catégories et des produits

Dans la création des produits et du contenu, j'ai procédé de la manière suivante :

- choix des produits sur des sites de montres divers, récupération des photos et des informations de chaque produit
- création des catégories de produits dans WooCommerce
- création des produits dans WooCommerce : nom, prix, catégorie, description, matériaux, mécanisme, réductions à date programmée, couleurs, stock disponible

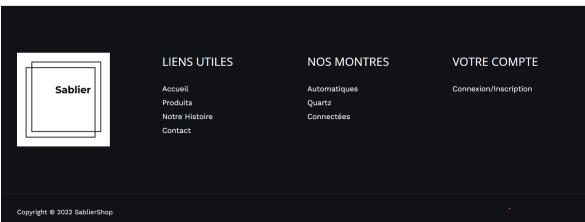
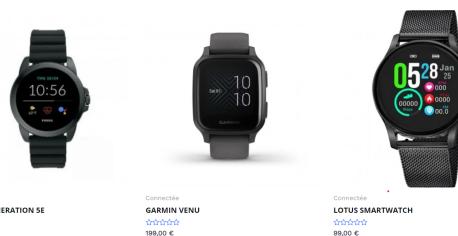
Création de la page d'accueil affichant les produits phares

Cette page est constituée d'une bannière en haut de la page d'accueil présentant une montre automatique d'exception accompagnée du slogan "Passez à l'automatique" suivi du nom du modèle et d'un bouton "BUY NOW" qui permet d'être redirigé directement sur la page du produit phare. En seconde partie de page, une section "Sablier s'adapte au numérique" présentant les produits les plus modernes appartenant à la catégorie "Montre connectée", avec un lien sur chaque image redirigeant vers la page du produit concerné. Le header est sous forme de menu qui se réduit en bouton "hamburger" quand la largeur de l'écran est en dessous d'un certain nombre de pixels ce qui rend la page responsive.

DOSSIER PROFESSIONNEL (DP)



SABLIER S'ADAPTE AU NUMERIQUE

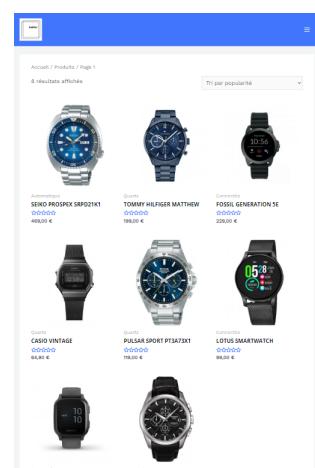
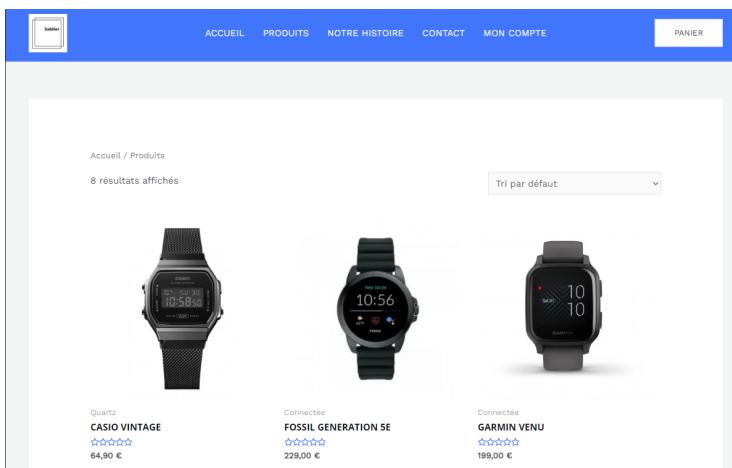


SABLIER S'ADAPTE AU NUMERIQUE



Personnalisation de la page Produits avec Elementor

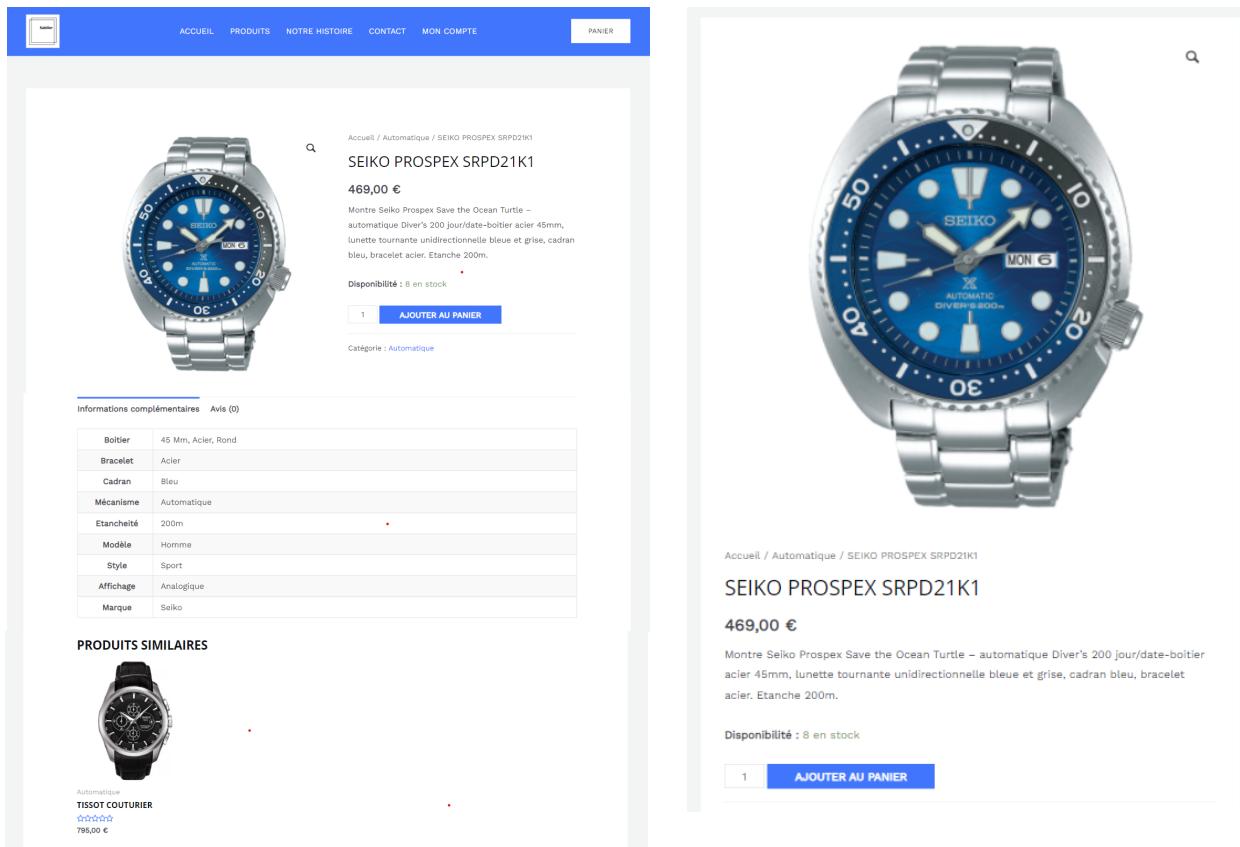
La page Produits est pré-fabriquée dans le thème Astra, mais elle a été personnalisée avec Elementor. Elle affiche le nom, la catégorie, la note, l'image, le prix de chaque produit, chaque bloc produit ayant un lien cliquable permettant d'aller sur la page du produit concerné. Une barre d'option de tri permettant de trier les produits est également disponible en haut de la page.



DOSSIER PROFESSIONNEL (DP)

Personnalisation de la page Produit avec Elementor

La page Produit est aussi pré-fabriquée dans le thème Astra, mais elle a été personnalisée avec Elementor. Elle affiche le nom, la catégorie, la note, l'image, le prix de chaque produit. J'ai décidé d'y afficher également la description produit. Un bouton ajouter au panier avec la quantité permet d'ajouter le produit au panier avec la quantité choisie. Les informations complémentaires sont affichées dans un tableau dans la seconde partie de la page. En bas de page, les produits similaires, et notamment ceux qui appartiennent à la même catégorie sont proposés.



Création et personnalisation de la page Notre Histoire avec Elementor

La page Notre Histoire résume l'histoire de la boutique ainsi que son processus commercial d'achat, de conseil et de vente de montres d'exceptions. Elle s'organise en différentes parties : une bannière dans le header avec une image et un titre général "QUI SOMMES NOUS", ensuite, une première section "SABLIER UNE HISTOIRE QUI DURE, NOTRE HISTOIRE" suivie de trois blocs de texte présentant les étapes clés de l'histoire de Sablier. Une seconde section, NOTRE PROCESSUS, suivie de quatre blocs de texte présentant les étapes clés du processus de prospection, d'achat, de conseil et de vente. Enfin, un footer qui reprend les liens importants disponibles dans le site web. La page est entièrement responsive.

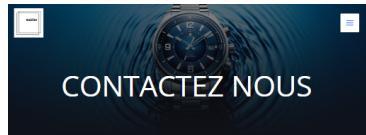
DOSSIER PROFESSIONNEL (DP)

Création et personnalisation de la page Contact avec Elementor

La page Contact permet aux utilisateurs de contacter les propriétaires de la boutique via email. Un formulaire est en effet disponible sur la page avec les champs suivants :

- nom
- prénom
- email
- message

Le message renseigné par l'utilisateur sera donc envoyé à l'email de l'administrateur paramétré de la boutique. La page dispose aussi d'une section INFOS avec les horaires, l'adresse et les coordonnées téléphoniques et email de la boutique. La page est entièrement responsive.



Création et personnalisation de l'espace client avec WooCommerce et Elementor

J'ai créé et paramétré l'espace client grâce au plugin WooCommerce. Cet espace client permet aux utilisateurs de s'inscrire, se connecter, consulter leurs commandes, ajouter des moyens de paiement, ajouter, modifier et supprimer leurs informations personnelles. La structure des pages a été personnalisée avec le plugin Elementor. Tous les formulaires y sont sécurisés.

Création du panier et simulation des commandes

Le panier et la simulation des commandes sont gérés avec le plugin d'extension WooCommerce Stripe Gateway qui gère les paiements par carte bancaire. L'administrateur du site a accès à toutes les commandes et aux différents récapitulatifs dans l'espace administrateur.

Le site conçu via WordPress avec les plugins Elementor pour la personnalisation de l'affichage des pages et WooCommerce pour tout ce qui est relatif à la gestion commerciale est entièrement responsive et s'adapte à toutes les tailles d'écrans.

Ce projet m'a permis d'acquérir les compétences de l'activité type 1 du référentiel suivantes :

- ✓ Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce
- ✓ Réaliser une interface utilisateur web statique et adaptable
- ✓ Réaliser une interface utilisateur web dynamique

2. Précisez les moyens utilisés :

J'ai utilisé :

- Google pour les recherches
- Youtube pour suivre un tutoriel d'installation et de paramétrage de WordPress et de ses plugins WooCommerce et Elementor
- Astra en tant que thème WordPress
- Elementor pour personnaliser l'affichage des pages
- WooCommerce pour paramétrier tout ce qui est relatif à la gestion commerciale
- FreeLogoDesign pour la création du logo

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur la conception de ce site internet.

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *La Plateforme_*

Chantier, atelier, service ➤ *Mon CMS*

Période d'exercice ➤ Du : *27/01/2022* au : *07/02/2022*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

Exemple n° 3 - O'Clock

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

Présentation du projet

Le projet O'Clock avait pour objectif de créer un site sur lequel tourne une horloge, ainsi que trois dispositifs : une alarme, un chronomètre et un minuteur, le tout en développé en Javascript. Ce site porte le nom de HighTimer.

Tâches effectuées

Processus de création du site : créations des différents fichiers et organisation

J'ai d'abord commencé par la création des différentes pages PHP et de leur structure en HTML :

- une page d'accueil présentant brièvement le site
- une page pour le chronomètre
- une page pour le minuteur
- une page pour l'alarme

Le header du site web a été créé dans un fichier à part qui est inséré à l'aide de la fonction PHP require(). Le header contient un logo sous lequel se trouve l'horloge qui affiche l'heure, les minutes et les secondes en temps réel, ainsi qu'une barre de navigation avec trois liens qui permettent d'accéder au chronomètre, au minuteur ou à l'alarme.



Le projet étant majoritairement développé en Javascript, j'ai créé un fichier script.js différent pour chaque fonctionnalité : un pour l'horloge, un pour l'alarme, un pour le minuteur et un pour le chronomètre, qui seront inclus dans les balises <head></head> de chaque page, sauf pour le script de l'horloge qui le sera sur toutes les pages étant donné qu'elle est contenue dans le header.

L'HORLOGE

Développement du header

J'ai d'abord construit le header en utilisant le modèle Flexbox : le header se constitue donc du logo et de trois liens vers les différents dispositifs du site : l'alarme, le minuteur et le chronomètre. J'ai créé une div avec l'id "display" dans le header à l'endroit où je souhaitais voir mon horloge être affichée.

DOSSIER PROFESSIONNEL (DP)

Développement du script de l'horloge

Récupération de l'élément html <div> ayant l'id display

Dans le script dédié à l'horloge, j'ai récupéré l'élément div ayant l'id display à l'aide de méthode javascript getElementById() dans une variable constante appelée display.

Création de la fonction updateTime()

J'ai créé une fonction updateTime(). Dans cette fonction, la date actuelle est stockée dans une variable date au format new Date(), à partir de laquelle je récupère dans différentes variables les heures, les minutes et les secondes à l'instant T, et qui va venir insérer en tant que texte dans la div ayant l'id display les heures, les minutes et les secondes à l'aide de la propriété javascript node.innerText.

Appel de la fonction updateTime()

Enfin à l'aide de la méthode javascript setInterval(), j'ai pu paramétriser l'appel de la fonction updateTime() toutes les 1000 millisecondes, soit toutes les secondes, ce qui permet d'afficher l'heure en temps réel à chaque seconde dans le header.

Le script de l'horloge est inclus dans l'élément html des métadonnées <head></head> de toutes les pages étant donné qu'il s'exécute dans le header.

L'ALARME

Le projet demande que l'utilisateur puisse paramétriser plusieurs alarmes qui seront affichées dans une liste avec le temps restant jusqu'à ce que l'heure choisie soit atteinte et marquée "passée" lorsque l'heure aura été atteinte. Chaque alarme peut également avoir un nom qui va s'afficher dans une fenêtre d'alerte au moment où l'alarme est paramétrée ainsi que dans la liste. Lorsque l'heure est atteinte, une alerte sonore doit se faire entendre.

Développement de la page structure HTML de la page de l'alarme

J'ai d'abord construit deux inputs dans la page PHP: un de type time pour la saisie de l'heure de l'alarme et un autre de type text pour le titre de l'alarme. Ensuite j'ai créé deux boutons : Set Alarm qui permet de paramétriser les alarmes ainsi qu'un bouton Clear Alarm qui permet d'arrêter le son de l'alarme. Enfin j'ai créé une liste uniquement avec les balises HTML . Chacun des éléments cités a un id qui lui est propre.

```
<!-- HORLOGE + ALARM -->
<div id="alarm">

    <input type="time" name="alarm" id="alarm-input">
    <input type="text" name="alarm-text" id="alarm-text" placeholder="Alarm name ">

    <div class="controls">
        <button class="button set-alarm" id="set-alarm">Set alarm</button>
        <button class="button clear-alarm" id="clear-alarm">Clear alarm</button>
    </div>

</div>

<div><ul id="alarm-list"></ul></div>
```

DOSSIER PROFESSIONNEL (DP)

Développement du script de l'alarme

Récupération des éléments HTML et créations des variables nécessaires

Dans le script et à l'aide de la méthode Javascript prototype.getElementById() je récupère chacun des éléments HTML nécessaire dans une variable qui lui est propre. J'ai également créé une constante audio = new Audio() qui permet de stocker un son dans une variable, son qui sera déclenché à l'heure de l'alarme, ainsi que deux variables alarmTime et alarmTimeOut qui prennent la valeur null et permettent respectivement de stocker l'heure de l'alarme et le temps qui doit s'écouler avant son déclenchement.

Création de l'évènement alarmInput

Avec la méthode javascript prototype.addEventListener('change', (event) => {}) lorsque la valeur de l'input de l'heure de l'alarme change, alors la variable alarmTime prend sa valeur, c'est-à-dire l'heure paramétrée. Cette variable sera utilisée ultérieurement.

Création de la fonction alarmTimer()

Cette fonction permet de modifier en temps réel l'affichage du temps restant avant le déclenchement de l'alarme dans la liste des alarmes. En effet, grâce à plusieurs conditions IF et ELSE IF dans cette fonction, les secondes, les minutes et les heures sont décrémentées et leur valeur est ramenée à 59 lorsque nécessaire. Enfin, les nouvelles valeurs sont injectées dans la page HTML grâce à la propriété javascript innerText.

Création de l'évènement setAlarm

Avec la méthode javascript prototype.addEventListener('click', (event) => {}) le paramétrage de l'alarme se déclenche lorsque l'input de l'alarme est rempli et que l'utilisateur clique sur le bouton Set Alarm. Pour ce faire plusieurs étapes se succèdent dans l'événement :

- récupération de l'heure de l'alarme choisie
- récupération de la date et heure du moment de paramétrage de l'alarme
- création des éléments de liste de l'alarme grâce à la méthode javascript document.createElement() et attribution d'une classe pour le style grâce à la méthode Element.setAttribute() qui affichera une alarme non passée en blanc.
- création de plusieurs variables de type string (texte) concaténées qui seront injectées dans la liste des alarmes
- gestion de l'heure de l'alarme paramétrée : si l'heure choisie est inférieure à l'heure du moment où on paramètre l'alarme, alors on modifie la variable date/heure de l'alarme pour qu'elle soit déclenchée le lendemain à l'heure choisie, et dans le cas inverse, l'alarme sera déclenchée dans la journée. La variable de texte de l'alarme sera aussi modifiée en fonction pour afficher "demain à XX:XX" ou aujourd'hui à XX:XX".
- injection de l'élément de liste dans la liste des alarmes
- création d'une variable timeout pour connaître le temps en heures, minutes et secondes avant le déclenchement de l'alarme
- création et calcul des variables heures, minutes et secondes restantes grâce à la méthode Math.Trunc()
- création d'une nouvelle variable de type string qui concatène les différentes variables de type string à injecter dans la liste
- création d'une condition IF qui viendra modifier le texte à injecter dans la liste HTML des alarmes et modifier l'affichage en "passée" lorsque l'alarme est passée et modifier également la classe de l'élément qui affichera une alarme passée en rouge.
- paramétrage du déclenchement de l'alarme et de l'audio grâce à la méthode setTimeout()
- appel de la fonction alarmTimer() toutes les secondes grâce à la méthode setInterval() pour que les temps restants avant le déclenchement de l'alarme se mettent à jour en temps réel
- création de l'alerte qui se déclenche lorsqu'une alarme est paramétrée

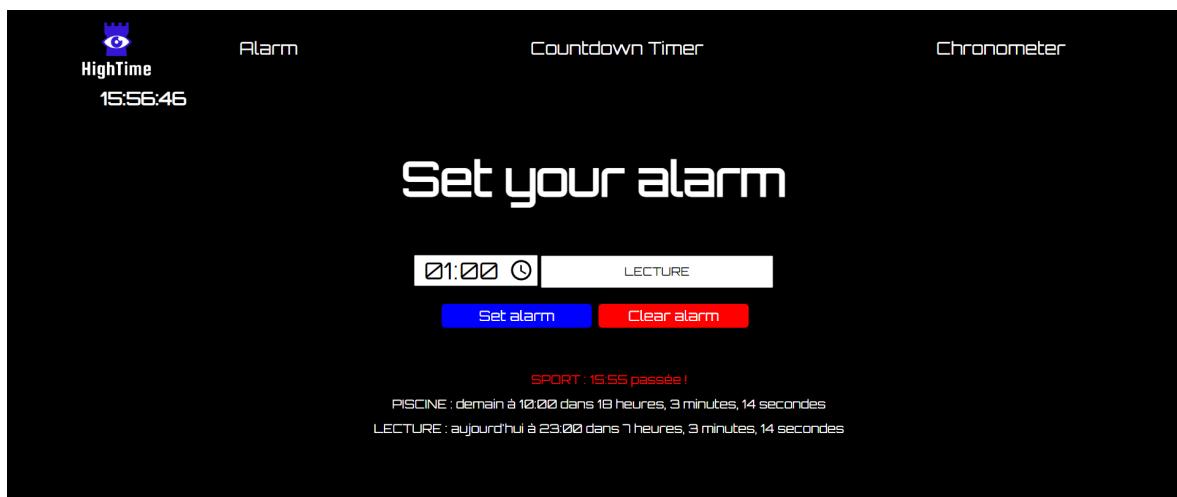
DOSSIER PROFESSIONNEL (DP)

Création de l'évènement clearAlarm

Avec la méthode javascript element.addEventListener('click', (event)=>{}), lorsque l'utilisateur clique sur le bouton clearAlarm, le son déclenché par l'alarme est stoppé.

Stylisation de l'alarme

Avec les classes attribuées grâce à la méthode setAttribute(), j'ai stylisé les affichages dans la liste des alarmes : blanc lorsqu'elles ne sont pas encore passées, rouge lorsqu'elles sont passées. J'ai utilisé la méthode flexbox pour la disposition des div et les media queries pour le responsive. Le fichier de style est inclus dans les balises des métadonnées de la page de l'alarme.



LE CHRONOMÈTRE

Le sujet impose un chronomètre avec un bouton qui permet de démarrer et stopper le chronomètre, un bouton tour qui permet d'ajouter le temps afficher dans une liste temps et un bouton reset pour remettre le chronomètre à 0. Pour cela, j'ai d'abord créé la page HTML avec tous les éléments nécessaires à l'exécution du script. Chaque élément à un id qui lui est propre.

```
<div id="chronometer">
  <div id="chrono"><span id="hours">00</span>:<span id="minutes">00</span>:<span id="seconds">00</span></div>
  <div id="buttons">
    <button class="button-blue" id="start-and-stop">Start</button>
    <button class="button" id="round">Tour</button> .
    <button class="button" id="reset">Reset</button>
  </div>
  <div>
    <ul id="round-list">

    </ul>
```

DOSSIER PROFESSIONNEL (DP)

Développement du script du chronomètre

Récupération des éléments HTML et créations des variables nécessaires

Dans le script, je récupère l'ensemble des éléments HTML grâce à la méthode prototype.getElementById() ainsi que les valeurs dans des variables et j'y ai créé de nouvelles variables nécessaires à l'exécution du script.

Création des fonctions

J'ai ensuite créé les différentes fonctions nécessaires à l'exécution du script:

- une fonction timer(), qui vient incrémenter les heures, les minutes, les secondes selon des conditions IF et ELSE IF et sera appelée à chaque seconde grâce à la méthode setInterval() et en adapter les affichages
- une fonction stopTimer(), qui vient arrêter l'appel de la fonction timer() et arrêtera de faire défiler le temps du chronomètre
- une fonction resetTimer(), qui lorsqu'elle est appelée appellera également la fonction stopTimer() et viendra remettre les valeurs des heures minutes et secondes à 0.
- une fonction roundListing(), qui va créer les éléments grâce à la méthode document.createElement() nécessaires à la création de la liste des temps de tour et les injecter dans la page HTML avec la propriété .innerText.
- une fonction startOrStop(), qui va gérer la transformation du bouton Start en bouton Stop et aussi changer la classe attribuée à ce bouton pour en changer la couleur.

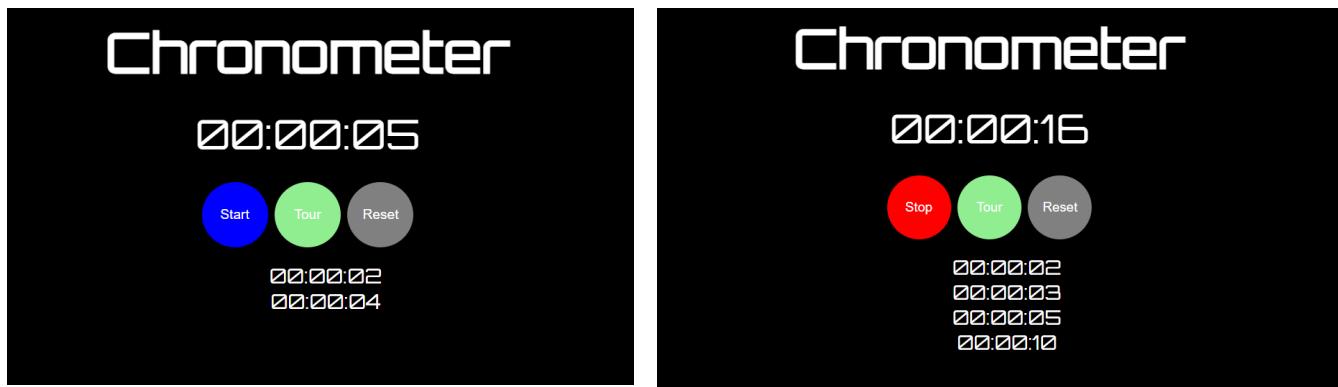
Création des événements

J'ai enfin créé les événements nécessaires à l'exécution du script :

- un premier événement déclenché lorsqu'on clique sur le bouton Start/Stop et qui va appeler la fonction startOrStop() qui change le bouton Start en Stop ou inversement, et qui va également appeler selon le cas soit la fonction timer(), soit la fonction stopTimer() qui vont venir déclencher ou arrêter le chronomètre.
- un second événement lorsqu'on clique sur le bouton reset, qui va appeler la fonction resetTimer et remettre le chronomètre à 0 et la fonction startOrStop qui va réinitialiser le bouton en Start et effacer la liste des tours.
- un dernier événement lorsqu'on clique sur le bouton tour, qui va appeler la fonction roundListing() et créer la liste des tours.

Stylisation de la page

Avec les classes attribuées grâce à la méthode setAttribute(), j'ai stylisé la couleur du bouton Start/Stop qui est bleu lorsque l'affichage est Start et rouge lorsque l'affichage est Stop. J'ai utilisé la méthode flexbox pour la disposition des div et les media queries pour le responsive. Le fichier de style est inclus dans les balises des métadonnées de la page du chronomètre.



DOSSIER PROFESSIONNEL (DP)

LE MINUTEUR

Le sujet impose qu'il soit possible d'augmenter le temps du minuteur grâce à des flèches ou de rentrer le temps dans un input. Lorsque le minuteur arrive à 0, une alerte doit apparaître et indiquer que le temps est écoulé. Pour ce faire, j'ai d'abord construit la structure du minuteur dans la page HTML, avec tous les éléments ayant chacun un id qui lui est propre.

```
<!-- MINUTEUR -->


<p id="hour-label" class="label">Hours</p><p id="min-label" class="label">Minutes</p><p id="sec-label" class="label">Seconds</p>
    <input id="hour" type="number" max="99" min="0" value="0" class="time"><input id="minute" type="number" max="60" min="0" value="0" class="time"><input id="second" type="number" max="60" min="0" value="0" class="time">
    <button id="startOrStop" class="btn">Start</button>
    <button id="reset" class="btn">Reset</button>


```

Développement du script du minuteur

Récupération des éléments HTML et créations des variables nécessaires

Dans le script, je récupère l'ensemble des éléments HTML grâce à la méthode prototype.getElementById() ainsi que les valeurs dans des variables et j'ai aussi créé de nouvelles variables nécessaires à l'exécution du script.

Création des fonctions

J'ai ensuite créé les différentes fonctions nécessaires à l'exécution du script :

- une fonction timer(), qui vient décrementer les valeurs des heures minutes et secondes en les ramenant à 59 lorsque nécessaire, et appelle la fonction stopTimer() si les trois ont atteint 0, et déclenche l'alerte "Time is Over" dans la page HTML.
- une fonction stopTimer(), qui vient arrêter l'appel de la fonction timer() et donc arrêter le défilement du temps du minuteur.
- une fonction startOrStopDisplay() qui gère le changement du bouton Start en bouton Stop et inversement ainsi que la classe du bouton grâce à la méthode setAttribute().

Création des événements

Enfin, j'ai créé les évènements nécessaires à l'exécution du script :

- un premier évènement lorsqu'on clique sur le bouton start ou stop, qui vient appeler la fonction timer() toutes les secondes afin que le temps du minuteur défile en temps réel, et appelle également la fonction startOrStopDisplay() pour le changement du bouton start en stop ou inversement.
- un second évènement lorsqu'on clique sur le bouton reset qui remet les valeurs des heures minutes et secondes à 0 et appelle la fonction stopTimer() qui permet d'arrêter l'appel de la fonction timer() et donc le défilement du temps du minuteur.

Stylisation de la page

Avec les classes attribuées grâce à la méthode setAttribute(), j'ai stylisé la couleur du bouton Start/Stop qui est bleu lorsque l'affichage est "Start" et rouge lorsque l'affichage est "Stop". J'ai utilisé la méthode flexbox pour la disposition des div et les media queries pour le responsive. Le fichier de style est inclus dans les balises des métadonnées de la page du minuteur.

DOSSIER PROFESSIONNEL (DP)

Ce projet m'a permis d'acquérir les compétences de l'activité type 1 du référentiel suivantes :

- ✓ Réaliser une interface utilisateur web statique et adaptable
- ✓ Réaliser une interface utilisateur web dynamique

i

2. Précisez les moyens utilisés :

J'ai utilisé :

- Visual Studio Code pour la création et l'édition de code.
- GitKraken et GitHub pour le versionning.
- Google pour les recherche en anglais
- Youtube pour suivre des tutoriels en anglais de création d'une horloge
- De la documentation en anglais : StarOverFlow.com
- De la documentation et forums et en français : OpenClassroom.com
- HTML / CSS
- JAVASCRIPT
- Freelogodesign pour la création du logo
- Media queries pour le responsive design
- Plesk pour l'hébergement du site web

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur la conception de ce site internet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ ***La Plateforme_***

Chantier, atelier, service ▶ ***O'CLOCK***

Période d'exercice ▶ Du : ***16/05/2022*** au : ***21/05/2022***

5. Informations complémentaires (facultatif)

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° 1 - Camping

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

Présentation du projet

Le projet Camping consistait à la création d'un site permettant aux utilisateurs de réserver un séjour au sein d'un camping. Le camping dispose de trois aires différentes, le lac, le maquis, la plage, et chacun des lieux ayant 4 emplacements. Les clients souhaitant réserver peuvent venir s'y installer en camping-car, occupant alors deux emplacements, ou en tente, occupant alors un emplacement. Le sujet impose un tarif fixe par emplacement occupé par jour. Des options (électricités, activités, accès au club) étaient également disponibles, facturées par jour, avec un tarif fixé par le sujet.

Le site devait obligatoirement comporter :

- une page d'accueil générale
- une page présentant le camping
- une page permettant de réserver, uniquement accessible aux utilisateurs connectés
- une page affichant les tarifs
- un espace utilisateur complet (inscription, connexion, mon compte, modifications des informations, mes réservations)
- un panel administrateur permettant à l'administrateur de gérer l'ensemble du site et de ses informations (recherche d'une réservation, détail d'une réservation, planning, modification d'une réservation, modification des tarifs...)

Tâches effectuées

Conception et création de la base de données

Dans un premier temps, j'ai conçu la base de données. Après réalisation du modèle conceptuel de données et du modèle logique de données, la base de données a été créée sur PHPMYADMIN. D'abord il a fallu créer la base de données, lui donner un nom (camping) et paramétrer l'interclassement (utf8_general_ci). Ensuite, il a fallu créer chacune des tables directement sur PHPMYADMIN en choisissant le typage pour chaque champ de la table.

La base de données s'organise comme suit :

- une table users comportant les champs id (clé primaire), login, email, firstname, lastname, password, ainsi qu'un champ booléen admin (0 ou 1), permettant de filtrer les droits d'accès des utilisateurs au panel administrateur.
- une table equipments comportant les champs id (clé primaire), type (tente ou camping car), size (soit le nombre d'emplacements occupés selon que le client choisi respectivement la tente ou le camping car) ainsi qu'un champ tarif.
- une table locations comportant les champs id (clé primaire), le nom du lieu (la plage, le maquis ou les pins) ainsi que le champ spaces, concernant le nombre d'emplacements disponible de base soit 4.
- une table options avec les champs id (clé primaire), name (le nom de l'option) et rate (le tarif de l'option par jour)
- une table reservations avec les champs id (clé primaire), arrival (date d'arrivée), departure (date de départ), length (durée du séjour), option_borne, option_discolcub, option_activites, en booléen (0 ou 1) afin de savoir si l'option est activée sur le séjour, rate(prix du séjour), id_user(clé étrangère liée à l'id du client qui réserve le séjour), id_location (clé étrangère liée au lieu de location choisi par le client) et id_equipment (clé étrangère liée à l'installation choisie par le client, camping car ou tente).

DOSSIER PROFESSIONNEL (DP)

Création des différents dossiers en reprenant un modèle de développement MODEL VIEW CONTROLLER

Dans un second temps, j'ai créé les différents dossiers afin de répartir les fichiers de notre site internet :

- un dossier model, contenant le fichiers PHP de connexion à la base de données ainsi que les classes qui contiennent les méthodes et fonctions permettant de lancer des requêtes SQL qui permettent de créer, modifier, afficher ou supprimer des informations de la base de données
- un dossier controller, contenant les fichiers PHP intervenant dans la logique et concernant les actions à effectuer par l'utilisateur et/ou l'administrateur, la gestion des erreurs, et le pont entre les fichiers de classe et l'affichage
- un dossier view, comportant les pages HTML et PHP visibles à l'utilisateur et/ou à l'administrateur

Ainsi, les fichiers du dossier view avec des formulaires ayant une méthode POST ou des pages ayant des paramètres GET dans l'URL envoient des informations aux fichiers du dossier controller qui traite les informations et prends des décisions d'actions en utilisant les méthodes des fichiers du dossier model qui permet d'interagir avec la base de données, et inversement, les interactions avec la base de données dans les fichiers du dossiers model sont envoyées aux fichiers du dossier controller qui après l'analyse logique renvoient alors les informations à l'affichage des fichiers du dossier view.

Développement du fichier PHP d'accès aux données

Dans un troisième temps, j'ai développé le fichier d'accès à la base de données. Pour cela, j'ai créé dans le dossier model le fichier bdd.PHP, qui permet la connexion à la base de données en PHP Data Objects. Ce fichier sera ensuite inséré en require dans le constructeur de chaque classe, permettant à chaque classe d'accéder et d'interagir avec la base de données.

```
1 <?php
2
3 $bdd = new PDO('mysql:host=localhost;dbname=camping', 'root', '');
4
5 try {
6     $bdd = new PDO('mysql:host=localhost;dbname=camping', 'root', '');
7 }
8
9
10 catch (PDOException $e) {
11     echo $e->getMessage();
12     die();
13 }
14
15 ?>
```

Création et développements des fichiers de classes PHP

Ensuite, j'ai créé dans le sous dossier classes du dossier model, les différents fichiers de classe PHP permettant l'interaction avec les différentes tables de la base de données, ainsi que le fichier de classe permettant la création d'un planning.

Les différentes classes développées sont :

- la classe user, comportant toutes les méthodes qui vont permettre de créer, consulter, modifier, supprimer les informations utilisateur
- la classe réservations, comportant toutes les méthodes qui vont permettre de créer, consulter, modifier, supprimer les informations des réservations
- la classe month, comportant toutes les méthodes qui vont permettre de créer un planning mensuel des réservations accessible aux administrateurs
- la classe admin, comportant une méthode qui permet de gérer les droits d'accès aux pages administrateur
- la classe equipments, comportant les méthodes permettant de récupérer le nombre d'emplacement occupés par le client en fonction de l'équipement choisi pour s'y installer
- la classe locations, comportant les méthodes permettant de vérifier les espaces disponibles sur chaque lieu du camping
- la classe rates, comportant les méthodes permettant d'afficher et/ou modifier les tarifs

DOSSIER PROFESSIONNEL (DP)

```
// FONCTION REGISTER -----  
  
public function Register($login, $email, $firstname, $lastname, $password) {  
  
    $data = [  
        'login'=>$login,  
        'email'=>$email,  
        'firstname'=>$firstname,  
        'lastname'=>$lastname,  
        'password'=>md5($password),  
        'admin'=>'0',  
    ];  
  
    $register = $this->connexion->prepare("INSERT into users (login, email, firstname, lastname, password, admin) VALUES (:login, :email, :firstname, :lastname, :password, :admin)");  
    $register->execute($data);  
  
}
```

Toutes les méthodes de classes et requêtes SQL ont été développées en Programmation orientée objet.

Création et développements des pages PHP du dossier VIEW et de leurs fichiers de traitement PHP CONTROLLERS associés

Pour poursuivre, j'ai créé dans le dossier view les pages du site visibles aux utilisateurs clients ou administrateurs. La structure de ces pages est construite avec du HTML et leur style avec du CSS. A la création de chaque page, nous avons également créé dans le dossier controller le fichier de traitement PHP qui récupère/voie les informations de la base de données à afficher en appelant les méthodes définies dans les différentes classes, et qui effectue aussi la gestion des erreurs ainsi que le lancement des fonctions de création, de modification, d'affichage ou de suppression des éléments de la base de données. Les fichiers controllers PHP sont appelés via la fonction PHP require() en amont de la page dans les balises <?php ?>.

```
<?php  
session_start();  
  
require('../controller/bookingController.php')  
  
?>
```

Dans chaque fichier du dossier controller, les classes utilisées sont inclus avec la fonction require(). La gestion des erreurs se fait à l'aide d'une variable \$valid qui prend la valeur true en amont du controller. A chaque erreur détectée en utilisant des conditions IF et/ou ELSE IF, la variable \$valid devient false, et une variable d'erreur est générée, permettant son affichage sur la page si elle existe via la fonction PHP isset(). Si aucune erreur n'est détectée à la fin du fichier controller, alors \$valid reste true on peut lancer la fonction de la classe associée. Dans ces fichiers, on peut également récupérer en paramètre GET dans l'URL une information qui sera stockée dans une variable sécurisée avec htmlspecialchars() et permettra de rendre les affichages dynamiques en fonction de ce paramètre, comme pour les informations de l'utilisateur dans son espace mon compte par exemple ou les informations d'une réservation.

```
require('../model/classes/class_user.php');  
  
$valid = (boolean) true;
```

```
if (empty($login)) {  
    $valid = false;  
    $err_login = "Renseignez votre login.";  
}  
  
elseif ( strlen($login)<6 || strlen($login)>20) {  
    $valid = false;  
    $err_login = "Le login doit contenir entre 6 et 20 caractères.";  
}
```

DOSSIER PROFESSIONNEL (DP)

```
if ($valid==true) {  
  
    $create_user = new User();  
    $create_user->Register("$login", "$email", "$firstname", "$lastname", "$password");  
    $form_register = 0;  
}
```

Création et développements du planning des réservations

Afin de construire le planning mensuel des réservations, j'ai suivi un tutoriel en anglais sur YouTube que j'ai adapté au sujet qui m'était donné. En effet, les affichages du planning s'adaptent en fonction du lieu choisi. Aussi, on peut changer le mois avec les boutons précédents et suivants. Pour chaque réservation, on retrouve le numéro de réservation (son id) ainsi que le nom de la personne ayant réservé. Un lien cliquable sur le texte renvoie vers la page affichant le détail de la réservation. Comme le camping comporte trois lieux, quand on clique sur le nom d'un lieu au dessus du planning, cela rajoute le nom du lieu en paramètre GET de l'URL et vient afficher le planning de réservation du lieu demandé, c'est-à-dire la Plage, les Pins ou le Maquis.

The screenshot displays a monthly calendar titled "Planning mensuel" for February 2022. The calendar grid shows days from Monday 31 to Sunday 06. Specific dates like February 20 (Sunday) are highlighted in yellow and linked to reservation details. A sidebar on the right shows a detailed view for reservation number 197, listing Thomas Serdjebi, dates from 2022-02-20 to 2022-02-26, and a total cost of 120 €.

Gestion des accès aux pages en fonction de l'utilisateur

L'accès aux pages dédiées aux utilisateurs connectés telles que les pages pour effectuer une réservation, consulter son compte et ses réservations est géré dans le fichier XXXXcontroller.php de chaque page. Lorsqu'un utilisateur se connecte, une variable de session `$_SESSION['id']` prenant la valeur de l'id de l'utilisateur est créée, ainsi qu'une variable de `$_SESSION['admin']` prenant la valeur du champ "admin" (0 ou 1). Grâce à ces variables de session, et selon les différents cas, les liens de la barre de navigation dans le header s'adaptent dans différents cas :

- si aucun utilisateur n'est connecté et donc qu'il n'existe aucune variable de session, l'utilisateur ne pourra accéder qu'aux pages d'accueils et de présentation du camping ou sera redirigé vers la page de connexion/inscription s'il tente d'aller dans l'onglet réservations
- si un utilisateur est connecté mais pas en tant qu'administrateur, il pourra en plus accéder à son espace mon compte ainsi qu'au formulaire de réservation
- si l'utilisateur connecté est un administrateur, il pourra accéder en plus au planning mensuel des réservations, à la page de recherche de réservations, à la consultation des détails de chaque réservations ainsi qu'aux formulaires de modifications des tarifs.

DOSSIER PROFESSIONNEL (DP)

De la même façon, les accès aux pages dédiées à l'administrateur telles que le planning des réservations, la gestion des tarifs, la recherche et l'accès aux détails des réservations, sont gérées grâce à la méthode définie dans la classe admin, méthode appelée dans le fichier XXXXController.PHP des pages concernées. La méthode vient vérifier la valeur de la session \$_SESSION['admin'] de l'utilisateur. Dans le cas où, un utilisateur autre qu'un administrateur tente d'accéder à ces pages, il se voit redirigé vers la page de connexion.

```
1 <?php
2
3     class Admin {
4
5         public function CheckAdmin () {
6
7
8             if (!isset($_SESSION['login']) || $_SESSION['admin'] =='0') {
9                 header('Location: compte.php');
10            }
11        }
12    }
13 ?>
```

```
$check_admin = new Admin;
$check_admin->CheckAdmin();
```

Sécurisation des formulaires et des paramètres GET dans l'URL

Enfin, je me suis occupé de mettre en place les sécurités sur les formulaires accessibles à l'utilisateur. En effet, tout ce qui est saisi par l'utilisateur sur le site web doit être contrôlé afin d'éviter les attaques ou les tentatives de piratages et de prise de contrôle du site et d'accès à la base de données. D'abord en utilisant la fonction PHP htmlspecialchars() sur tous les champs de formulaire et variables récupérant un paramètre GET de l'URL qui convertit tous les caractères éligibles en entités HTML contre les injections SQL et XSS Cross Site Scripting. Puis, j'ai ajouté des vérifications du typage de chaque champ rempli par l'utilisateur. Par exemple, pour un nom et un prénom, j'ai créé un regex puis utilisé la fonction preg_match() pour vérifier qu'ils ne contiennent ni chiffres, ni caractères spéciaux.

Ce projet m'a permis d'acquérir les compétences de l'activité type 2 du référentiel suivantes :

- ✓ Créer une base de données
- ✓ Développer les composants d'accès aux données
- ✓ Développer la partie back-end d'une application web ou web-mobile.

2. Précisez les moyens utilisés :

J'ai utilisé :

- Visual Studio Code pour la création et l'édition de code.
- GitKraken et GitHub pour le travail en équipe et le versionning.
- PHPMYADMIN pour la gestion de la base de données
- Google pour les recherches en anglais
- Youtube pour suivre des tutoriels de création de planning mensuel
- De la documentation en anglais : StarOverFlow.com / forums en français : OpenClassroom.com
- HTML / CSS
- PHP
- Freelogodesign pour la création du logo
- Media queries pour le responsive design
- Plesk pour l'hébergement du site web

DOSSIER PROFESSIONNEL^(DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé en binôme avec Valentin Mathieu.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ ***La Plateforme_***

Chantier, atelier, service ▶ ***Camping***

Période d'exercice ▶ Du : ***28/12/2021*** au : ***07/01/2022***

5. Informations complémentaires (*facultatif*)

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° 2 - Autocomplétion

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

Présentation du projet

Le but du projet était de créer un site moteur de recherche sur le thème de notre choix. Les pages exigées étaient :

- une page d'accueil semblable à un moteur de recherche
- une page affichant les résultats de la recherche
- une page affichant le détail d'un résultat de recherche
- une barre de recherche contenue dans le header des pages
- les suggestions de la barre de recherche devait être en deux parties : en premier, les résultats qui correspondent exactement à la recherche de l'utilisateur et commence donc par sa saisie, et en second les résultats qui contiennent les résultats de la recherche de l'utilisateur.

Tâches effectuées

Choix du thème et récupération des informations

Dans un premier temps, j'ai choisi le thème des araignées et choisi une liste d'une vingtaine d'araignées. J'ai construit un tableau excel avec l'ensemble des informations que j'ai sélectionnées pour chaque araignée : son nom commun, son nom latin, sa région d'habitation, sa mortalité (oui/non) ainsi qu'une image. J'ai stocké les informations dans un tableau excel et les images dans un serveur en ligne.

Création de la base de données

La base de données ne nécessitait qu'une table. J'ai donc créé ma base de données sur PHPMYADMIN avec un interclassement en utf8_general_ci, ainsi que la table spiders avec les informations que j'avais récupérées, ainsi que le lien vers l'image de chaque araignée.

Création de la classe Spider et des fonctions associées

J'ai créé un fichier de classe spider.php dans le constructeur de la classe, j'ai développé l'accès à la base de données en PHP Data Objects. Ensuite j'ai créé différentes fonctions :

- une fonction renvoyant sous forme d'array les informations d'une araignée avec en paramètre l'id de l'araignée concernée.
- une fonction renvoyant sous forme d'un array à double entrée (une première entrée start pour les araignées dont le nom commun commence par la saisie de l'utilisateur, et une seconde entrée contain pour les araignées dont le nom commun contient la saisie de l'utilisateur). Dans cette fonction, deux requêtes SQL différentes et développées en programmation orientée objet ont été créées pour chacun des cas et viennent insérer dans l'array les résultats des requêtes dans leur entrée respective. Enfin dans cette fonction, j'ai créé une boucle for dans une boucle foreach afin de parcourir les deux entrées de l'array (start et contain), comparer les résultats et supprimer les doublons éventuels de résultats contenus à la fois dans l'entrée contain et l'entrée start et ne conserver que ceux dans l'entrée start.

DOSSIER PROFESSIONNEL (DP)

Création de la barre de recherche avec suggestion

Pour la barre de recherche, je l'ai développée sous forme d'un formulaire en méthode GET avec un input qui permet l'utilisateur de saisir sa recherche. Pour la page d'accueil, cette barre se trouve dans le <body> de la page tandis que pour les autres elle se situe dans le header. J'ai également créé une div pour l'affichage des suggestions de la recherche sous la barre de recherche.

```
<form method="GET">
    <div class="row">
        <div class="col s12">
            <div class="row">
                <div class="input-field col s12">
                    <i class="material-icons prefix"></i>
                    <input type="text" id="autocomplete-input" placeholder="Chercher une araignée" class="autocomplete" name="search" autocomplete="off" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        <label for="autocomplete-input"></label>
                    </div>
                </div>
            </div>
        </div>
    </form>
    .
    <div id="header_dropdown">

    </div>
```

Développement du fichier JSON

Afin de récupérer le tableau des suggestions de recherche dans le script, j'ai créé un fichier ou j'inclus le contenu de la classe spider avec un require. J'y récupère la saisie de l'utilisateur envoyée en GET dans l'URL qui sera le paramètre de la fonction retournant les résultats de suggestions. Enfin, j'appelle cette fonction dans le fichier et je l'affiche au format json grâce à un echo json_encode(). Ce fichier affiche donc au format json les résultats de suggestions.

```
3  require_once('spiders.php');
4
5  $word = htmlspecialchars($_GET['search']);
6
7  $autocomplete = new Spider();
8  @$results = $autocomplete->research($word);
9
10
11 echo json_encode($results);
12
```

Développement du script de la barre de recherche avec suggestion par autocomplétion

Je récupère d'abord les éléments HTML grâce à la méthode javascript document.getElementById().

Création des évènements

Ensuite, j'ai créé un premier événement déclenché lorsque la valeur de l'élément input change avec la méthode javascript element.addEventListener('input', function() {}). Dans cet évènement, plusieurs cas de figures sont gérés par le code :

- le cas où l'input est vide, et donc on supprime l'ensemble des résultats de suggestions de l'autocomplétion
- le cas où la valeur de l'input est différente de 0 et on récupère alors le fichier PHP affichant le tableau des suggestions au format json grâce à la méthode fetch en indiquant le nom du fichier avec un paramètre search en GET dans l'URL ayant la valeur de l'input saisi par l'utilisateur. Ainsi, j'ai pu traiter le fichier json en plusieurs étapes :

DOSSIER PROFESSIONNEL (DP)

1. récupération des éléments de la liste de suggestions par leur nom de classe avec la méthode javascript document.getElementByClassName()
2. suppression de la liste de suggestion dès que l'input de la barre de recherche est vidé ou modifié
3. récupération et stockage de l'élément HTML (div ayant l'id header_dropdown) dans lequel la liste est générée grâce à la méthode javascript document.getElementById()
4. création d'une première div pour les suggestions qui commencent par la recherche de l'utilisateur avec la méthode javascript document.createElement('div')
5. attribution à cette première div d'un id avec la méthode javascript element.setAttribute()
6. injection de cette div dans la div header_dropdown grâce à la méthode javascript node.appendChild()
7. parcours du tableau ayant la première entrée "start" à l'aide du boucle for, qui va venir créer le lien vers le résultat de recherche avec la méthode document.createElement('a') puis attribution du texte avec la propriété element.innerText, attribution du lien avec la propriété element.href, et attribution de la class dropdown-item avec la propriété element.classList.add()
8. injection dans la page HTML de la première liste grâce à la propriété element.appendChild()

J'ai à nouveau suivi les étapes 4 à 8 pour la création de la seconde partie de la liste avec les suggestions qui contiennent la saisie de l'utilisateur et le parcours de l'entrée "contain" du tableau.

Enfin, j'ai créé un évènement lorsque l'utilisateur appuie sur la touche Entrée de son clavier grâce à la méthode javascript document.addEventListener('keypress', function() {}) : cela le mène aux résultats de recherches de ce qui a été saisi. En effet, la saisie de l'utilisateur sera alors en paramètre GET de l'URL de la page des résultats de recherche. Ce script est inclus dans les balises <head></head> des pages du site web.



Développement de la page des résultats de recherche

En premier lieu, j'ai construit la structure de ma page en utilisant un tableau de Bootstrap 5 qui est entièrement responsive et où les noms des colonnes sont directement saisis dans les balises de tête de colonnes du tableau. Ensuite, j'ai créé un fichier de traitement PHP de la page où j'inclus mon fichier de classe et j'appelle ma fonction permettant de récupérer les résultats de recherche via le mot de recherche saisi par l'utilisateur en paramètre GET dans l'URL et qui sera donc le paramètre d'exécution de la fonction. Le paramètre GET est sécurisé avec la fonction PHP htmlspecialchars() afin de contrer les attaques de type XSS Cross Site Scripting et injections SQL. Après avoir récupéré le tableau des résultats et à l'aide de boucles foreach sur chaque entrée du tableau (start et contain), j'ai mis en forme le placement de chaque information dans ma page HTML. Pour chaque araignée présente dans le tableau, un lien contenu sur chaque élément de la ligne renvoie vers la page détaillant les informations sur l'araignée choisie.

```
1 <?php
2
3 require('spiders.php');
4
5 //Récupère les résultats de recherche en deux array
6
7 $word = htmlspecialchars($_GET['search']);
8 $searching = new Spider();
9 $results = $searching->research($word);
10
11
12 ?>
```

DOSSIER PROFESSIONNEL (DP)

```
<?php foreach($results['start'] as $spider) {
    if ($spider['mortelle'] == 1 ) { $spider['mortelle'] = 'Oui' ;} else {$spider['mortelle'] = 'Non' ;}
    echo "
<tr>
    <td>
        <a href='element.php?id='".$spider['id']."'><img src='".$spider['image']."' class='img-fluid img-thumbnail' alt='Sheep'></a>
    </td>
    <td><a href='element.php?id='".$spider['id']."'>".$spider['nom_commun']. "</a></td>
    <td><a href='element.php?id='".$spider['id']."'>".$spider['nom_latin']. "</a></td>
    <td>".$spider['region']. "</td>
    <td>".$spider['mortelle']. "</td>
</tr>
</a>" ;
}

?>

<?php foreach($results['contain'] as $spider) {
    if ($spider['mortelle'] == 1 ) { $spider['mortelle'] = 'Oui' ;} else {$spider['mortelle'] = 'Non' ;}
    echo "
<tr>
    <td class='w-25'>
        <a href='element.php?id='".$spider['id']."'><img src='".$spider['image']."' class='img-fluid img-thumbnail' alt='Sheep'></a>
    </td>
    <td><a href='element.php?id='".$spider['id']."'>".$spider['nom_commun']. "</a></td>
    <td><a href='element.php?id='".$spider['id']."'>".$spider['nom_latin']. "</a></td>
    <td>".$spider['region']. "</td>
    <td>".$spider['mortelle']. "</td>
</tr>" ;
}

?>
```

SPIDER CHASER

Chercher une araignée

Résultats de votre recherche

Nom Commun	Nom Latin	Région	Mortelle
	Mygale Goliath	Theraphosa blondi	Forêts tropicales
	Mygale australienne	Atrax robustus	Australie
	Veuve noire d'Amérique	Latrodectus mactans	Amérique du Nord
	Araignée chameau	Solifugae	Moyen-Orient

Ce site Web a été conçu par :
Thomas Serdjebi
Etudiant à l'école du numérique La Plateforme_
Retrouvez le projet sur : [GitHub](#).

La Plateforme



DOSSIER PROFESSIONNEL (DP)

Développement de la page element avec les détails sur chaque araignée

En premier lieu, j'ai construit la structure de ma page en utilisant Bootstrap 5 ce qui la rend entièrement responsive et où j'ai adapté le contenu. Ensuite, j'ai créé un fichier traitement de la page où j'inclus la classe spider et appelle la fonction qui retourne un tableau avec toutes les informations d'une araignée, ayant en paramètre l'id de l'araignée qui est récupérée en paramètre GET['id'] dans l'URL. Afin de pallier aux attaques de types XSS Cross Site Scripting, j'ai utilisé la fonction htmlspecialchars sur la récupération de l'id de l'araignée en paramètre GET dans l'URL. Ainsi, j'ai pu afficher chaque information du tableau retourné dans la page HTML.

```
1 <?php
2
3 require('spiders.php');
4
5 //Récupère les résultats
6
7 $id = htmlspecialchars($_GET['id']);
8 $infos = new Spider();
9 $spider= $infos->getOne($id);
10
11 if($spider[0]['mortelle'] == 0 ) {
12     $spider[0]['mortelle'] = 'Non';
13 } else {
14     $spider[0]['mortelle'] = 'Oui';
15 }
16
17
18 ?>
```

```
<div class="container mt-5">
    <div class="row">
        <div class="col-lg-8">
            <!-- Post content-->
            <article>
                <!-- Post header-->
                <header class="mb-4">
                    <!-- Post title-->
                    <h1 class="fw-bolder mb-1"><?php echo $spider[0]['nom_commun']?></h1>
                    <!-- Post meta content-->
                    <div class="text-muted fst-italic mb-2"><?php echo $spider[0]['nom_latin']?></div>
                <!-- Preview image figure-->
                <figure class="mb-4"></figure>
                <!-- Post content-->
                <section class="mb-5">
                    <p class="fs-5 mb-4"><?php echo $spider[0]['description']?></p>
                </section>
            </article>
        </div>
    </div>
</div>
```

SPIDER CHASER

Chercher une araignée



Araignée-banane

Phoneutria nigriventer

Aussi appelée araignée-banane à cause du fait qu'elle vit majoritairement dans les régimes de bananes, cette espèce vivant au Brésil est connue comme étant l'une des plus dangereuses au monde ! Leur morsure, pouvant abattre une personne de 80 kilos, est plutôt violente et provoque des vertiges, des problèmes cardiaques et de longues et douloureuses érections chez les hommes.

Ce site Web a été conçu par :
Thomas Serdjebi
Etudiant à l'école du numérique La Plateforme...
Retrouvez le projet sur : [GitHub](#)

La Plateforme



DOSSIER PROFESSIONNEL (DP)

Création du header avec la barre de recherche

Le header est construit avec un logo et la barre de recherche avec suggestion par autocomplétion. Le header est inclus dans chaque page avec un require.

The screenshot shows a search interface titled "SPIDER CHASER". At the top right is a search bar with the placeholder "RE". Below it is a section titled "Résultats de votre recherche". A sidebar on the right lists suggestions: "Recluse brune", "Araignée tigre", "Veuve noire d'Amérique", "Tarentule asiatique", "Veuve noire d'Europe", and "Araignée redback". The main content area displays three search results in a table format:

Nom Commun	Nom Latin	Région	Mortelle
	Loxocephala reclusa	Amérique du Nord	Oui
	Haploclima Schmidtii	Sud de la Chine, Vietnam	Oui
	Poecilothelia	Sri Lanka	Non

Les pages ont été stylisées à l'aide de Bootstrap 5 et CSS.

Ce projet m'a permis d'acquérir les compétences de l'activité type 2 du référentiel suivantes :

- ✓ Créer une base de données
- ✓ Développer les composants d'accès aux données
- ✓ Développer la partie back-end d'une application web ou web-mobile.

Ce projet m'a permis d'acquérir les compétences de l'activité type 1 du référentiel suivantes :

- ✓ Réaliser une interface utilisateur web statique et adaptable
- ✓ Développer une interface utilisateur web dynamique

2. Précisez les moyens utilisés :

J'ai utilisé :

- Visual Studio Code pour la création et l'édition de code.
- GitKraken et GitHub pour le versionning.
- PHPMYADMIN pour la gestion de la base de données
- Google pour les recherches en anglais
- Youtube pour suivre des tutoriels de création de barre de recherche avec autocompletion en anglais
- De la documentation en anglais : StarOverFlow.com / Forums en français : OpenClassroom.com
- HTML / CSS
- Javascript
- PHP
- Media queries pour le responsive design
- Bootstrap 5 pour certaines pages.
- Plesk pour l'hébergement du site web

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ ***La Plateforme_***

Chantier, atelier, service ▶ ***Autocompletion***

Période d'exercice ▶ Du : ***25/04/2022*** au : ***28/04/2022***

5. Informations complémentaires (*facultatif*)

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 - *Boutique en ligne - parcours d'achat utilisateur*

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

Présentation du projet

Dans la continuité du projet Boutique en Ligne présenté dès la page 5, j'ai également réalisé le système de panier de la boutique jusqu'à la commande. Pour cela je me suis documenté sur divers sites comme Stackoverflow.com en anglais ou le site web du développeur Graphikart.

Tâches effectuées

Développement de la classe Cart et des fonctions et méthodes associées

J'ai d'abord créé une classe Cart, qui hérite de la classe Model permettant la connexion à la base de données. J'y ai développé plusieurs fonctions :

- une fonction permettant de récupérer les infos du produit sélectionné à ajouter au panier.
- une fonction qui ajoute le produit au panier. Le panier est en fait une variable de session \$_SESSION['cart'] avec une entrée supplémentaire sous la forme de \$_SESSION['cart'][product_id] avec l'id du produit choisi qui prend la valeur de la quantité du produit ajouté au panier. Si le produit existe dans le panier alors ce produit est ajouté avec une quantité de 1 au panier, sinon, le nombre du produit présent dans le panier est incrémenté
- une fonction qui retourne un tableau avec toutes les informations des produits contenus dans le panier
- une fonction permettant de supprimer un produit du panier
- une fonction permettant de supprimer le panier
- une fonction permettant de calculer le prix total du panier
- une fonction permettant de recalculer le prix total du panier après modification des quantités des produits
- une fonction permettant de compter le nombre de produits présents dans le panier

Développement de la classe Order et des fonctions et méthodes associées

J'ai aussi créé une classe Order(commande) héritant de la classe Model permettant la connexion à la base de données. J'y ai développé les fonctions suivantes :

- une fonction qui permet de créer une commande
- une fonction qui retourne un tableau affichant l'ensemble des informations d'une commande et des produits liés grâce à des INNER JOIN
- une fonction qui permet de supprimer une commande
- une fonction qui permet de valider une commande et passer son statut de "en attente de confirmation" à "confirmé"
- une fonction qui permet de changer le statut du paiement de la commande de "en attente de paiement" à "payé".
- une fonction qui permet d'afficher les commandes d'un utilisateur

DOSSIER PROFESSIONNEL (DP)

Développement du système de panier

Lorsqu'un utilisateur ajoute un produit au panier, il est systématiquement redirigé dans l'immédiat sur la page produits. Cependant, le bouton ajouter au panier renvoie en fait vers la page addtocart(ajouter au panier)avec en paramètre GET l'id du produit choisi dans l'URL. J'ai donc créé un fichier de traitement de la page addtocart qui vient récupérer l'id en GET présent dans l'URL en la sécurisant avec la fonction htmlspecialchars() et qui l'utilise en paramètre de la fonction permettant de l'ajouter au panier. Ce fichier de traitement vérifie également le stock disponible en appelant une fonction de vérification du stock définie dans la classe propre aux produits, avant de lancer l'ajout au panier.

L'utilisateur peut ensuite accéder à son panier grâce au bouton disponible dans le header. La page panier a également un fichier de traitement PHP qui appelle la fonction qui permet d'afficher l'ensemble des informations des produits contenus dans le panier. Aussi le fichier de traitement de cette page permet d'appeler les différentes fonctions citées ci-dessus (suppression du panier, suppression d'un article, modification des quantités et donc du prix du panier, comptage du nombre de produits présents dans le panier), lorsque l'utilisateur clique sur les boutons associés à ces fonctions.

The screenshot shows the iLighter website's shopping cart page. At the top, there is a navigation bar with the logo 'iLighter', a search icon, and links for 'Connexion', 'Inscription', and a shopping cart icon showing '6'. Below the navigation is a black header bar with 'Nos produits' and 'Nouveautés' links. The main content area is titled 'VOTRE PANIER' (Your Cart). It lists two items: 'Brûquet à résistance USB intégré X4 couleurs' and 'Brûquet à résistance USB slim X3 couleurs', each with a quantity input field (set to 4 and 2 respectively), a price of 35.60€ and 21.80€, and a delete icon. A button at the bottom of this list says 'Actualisez votre panier pour enregistrer les modifications de quantités !' (Update your cart to save changes to quantities!). To the right of this button is a 'Actualiser le panier' (Update cart) button. Below the cart list, there is a summary table with rows for 'Sous total HT' (57.40€), 'Livraison HT' (7.90€), 'Total HT' (65.30€), 'TVA 20%' (13.06€), and 'Total TTC' (78.36€). At the bottom of the page are two buttons: 'CONTINUER VOS ACHATS' (Continue Shopping) and 'VALIDER LE PANIER' (Validate Cart). The footer contains the iLighter logo, contact information (tel: +33 4 42 00 00 00, email: contact@ilighter.com), navigation links (Accueil, Bûquets, Bûquets jetables, Zippo), and social media links for Facebook and Instagram.

DOSSIER PROFESSIONNEL (DP)

Développement du système de transformation du panier en commande

Si l'utilisateur valide son panier, il transite sur une page qui ne lui sera pas visible qui vient valider le panier et le transformer en commande. La page cartvalidation (validation du panier) inclut un fichier de traitement. Ce fichier de traitement appelle deux classes : Cart(panier) et Order(commande) et vient insérer en base de données dans la table de liaison products_orders entre la table products(produits)et orders(commandes) les différentes informations pour chaque produits tel que son id, l'id de la commande qui lui est liée ainsi que la quantité associée, et insérer dans la table orders(commande) les informations liées à la commande tel que son numéro, son prix HT, son prix TTC, sa date, son état de confirmation (ici en attente de confirmation), l'état du paiement et l'id de l'utilisateur ayant passé la commande. une variable de session \$_SESSION['order'] est créée et prend la valeur de l'id de la commande.

Développement du système d'enregistrement des informations de livraison et sécurisation du formulaire

L'utilisateur est donc directement redirigé vers le formulaire de livraison qui inclut un fichier de traitement qui permet de gérer les erreurs, sécuriser les données saisie par l'utilisateur à l'aide des fonctions PHP htmlspecialchars(), htmlentities() et des filtres regex associés à la fonction PHP preg_match(), créer les variables d'erreurs qui seront affichées si elles ont été générées, et insérer les informations de livraison dans la table shippings(livraisons) contenant le nom, le prénom, l'adresse, la ville, le code postal, la date minimum et maximum de livraison ainsi que le numéro de commande associé.

Développement du système de validation et modification de commande

L'utilisateur est ensuite dirigé vers une page qui récapitule l'ensemble des informations de sa commande qui est toujours en statut en attente de confirmation et vient demander confirmation ou modification à l'utilisateur. Il y retrouve ses informations de livraison ainsi que les informations de la commande. Cette page inclut un fichier de traitement. Ce fichier de traitement inclut les classes orders(commandes) et shippings(shippings). Si l'utilisateur choisit de modifier sa commande, ses informations de livraisons et sa commande sont supprimées de la base de données et il est redirigé vers son panier qui est resté identique. S'il décide de seulement changer ses informations de livraisons, ce sont uniquement ces dernières qui sont supprimées de la table shippings dans la base de données et il est redirigé vers le formulaire de livraison. S'il décide de valider sa commande, le statut de sa commande en base de données passe de en attente de confirmation à confirmé. Il est ensuite redirigé vers la page de paiement.

Confirmez votre commande

Vérifiez votre commande et vos informations de livraison avant de passer au paiement.

Votre commande					
Articles	PU	Qté	Montant HT	TVA	Montant TTC
Briquet à résistance USB intégré X4 couleurs	8.90	5	44.5	8.9	53.4
Briquet à résistance USB slim X3 couleurs	10.90	3	32.7	6.54	39.24
Briquet Pierre Zorro Antik Dragon	29.00	1	29	5.8	34.8
TOTAL			106.20 €	21.24 €	127.44 €

[Modifier la commande](#)

Vos informations de livraison

Destinataire
Nom : Serdjebi
Prénom : Thomas

Luzia.eu
3 Rue Pointe à Pitre
13006 MARSEILLE
FRANCE

[Modifier la livraison](#)

[Valider et passer au paiement](#)

DOSSIER PROFESSIONNEL (DP)

Développement du système de paiement

La page de paiement est mise en place avec Stripe qui est une plateforme de gestion de paiement. Cette page inclut donc un fichier de traitement qui récupère l'id de la commande enregistré dans une variable de session et appelle la fonction permettant de récupérer les informations de la commande et notamment son prix pour procéder au paiement. Lorsque le paiement est confirmé, l'utilisateur transite sur une page indexpayment qui lui est invisible. En transitant sur cette page, le fichier de traitement qui y est inclus enclenche la suppression du panier qui est désormais obsolète et redirige l'utilisateur vers la page qui récapitule à nouveau ses informations de livraison et de commande. Le fichier de traitement de cette page appelle les fonctions nécessaires aux affichages des informations et vient changer le statut du paiement de la commande de "en attente de paiement" à "payé".

Ce projet m'a permis d'acquérir les compétences de l'activité type 2 du référentiel suivantes :

- ✓ Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

2. Précisez les moyens utilisés :

J'ai utilisé :

- Visual Studio Code pour la création et l'édition de code.
- GitKraken et GitHub pour le versionning.
- PHPMYADMIN pour la gestion de la base de données
- Google pour les recherches en anglais
- Youtube pour suivre des tutoriels de création d'un système de panier
- De la documentation en anglais : StarOverFlow.com / forums et en français : OpenClassroom.com
- HTML / CSS
- PHP
- Media queries pour le responsive design
- Bootstrap 5 pour certaines pages.

3. Avec qui avez-vous travaillé ?

J'ai travaillé en binôme avec Valentin Mathieu et Stéphane Mathieu.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La Plateforme_**

Chantier, atelier, service ▶ **Boutique en Ligne**

Période d'exercice ▶ Du : **10/02/2022** au : **10/03/2022**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL^(DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) Serdjebi Thomas

,

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 Marseille

le 20/06/2022

pour faire valoir ce que de droit.

Signature :



DOSSIER PROFESSIONNEL^(DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)