

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

▶ BARON

Nom d'usage

▶ BARON

Prénom

▶ VALERIE

Adresse

▶ 312 A boulevard des écureuils – 06210 Mandelieu-la-Napoule

Titre professionnel visé

Développeur Logiciel

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>

DOSSIER PROFESSIONNEL (DP)

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1 : Développer une application client-serveur p. 6

- *Création from scratch du projet Guider - Stage au Sictiam* p. 6
- *Développement Back-end du site Guider* p. 12

Intitulé de l'activité-type n° 2 : Développer une application web p. 17

- *Site Guider- développement du front-end* p. 17
- *Création d'un Thème WORDPRESS demandé par l'école Simplon.co* p. 24

Déclaration sur l'honneur p. 30

Annexes p. 31

DOSSIER PROFESSIONNEL (DP)

Abstract

The project I'm going to present you is born from a third prize won at the Hackathon for Tourism contest that took place in April 2018 at Sophia-Antipolis.

The main objectives were to create and enhance tourism in Alpes Maritimes's backyards.

I imagined with one colleague a website that could create links between local guides, wishing to tell the story of their own village and reveal the hidden heritage, and people who want to run from the beaches and the crowded coast.

The objective is also to attract new tourism toward local farms, local producers (as beekeepers).

Sictiam, the company who gave us this third prize, proposed us an internship in order to develop our idea.

First of all, this website makes a small presentation of the seven countries that divide our department, emphasizing geographical characteristics and local sights. Each country proposes a selection of local guides available for discovery, promenade or tasting experience. Each guide is bookable online. Each guide can manage his appointments.

The administrators of the website (us) keep the control on it. They recruit themselves the guides and create the personal file thanks to a panel control (CRUD) which is available after login.

The user can create a personal account, log in and eventually modify his personal data like email address, phone. He can also cancel the latest reservation.

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 DEVELOPPER UNE APPLICATION CLIENT-SERVEUR

Exemple n°1 ► Création from scratch du projet « Guider » - Stage au SICTIAM

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

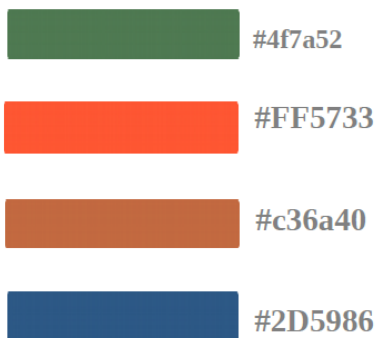
Le projet Guider est né du **Hackathon « Hack For Tourism »** qui a eu lieu en Avril 2018. Il s'agit de développer un site web destiné à mettre en relation des guides locaux et des clients afin d'accroître le tourisme dans l'arrière-pays. Le **Sictiam** nous a aidés à développer l'idée du projet grâce à sa connaissance de l'arrière-pays et des acteurs pouvant nous accompagner si nous décidions de lancer l'application.

Les étapes de création sont les suivantes :

1. Maquettage du site

La page de présentation de Guider donne accès à des informations sur les 7 arrière-pays qui composent le département 06. Chaque arrière-pays y est détaillé et des guides locaux sont proposés à l'utilisateur. L'utilisateur peut créer un compte et se connecter afin de réserver un guide ou plusieurs.

Guider – Charte graphique et Logo



Les couleurs sont tirées de l'image principale présente sur la page d'accueil du site. Ce sont des couleurs provençales en rapport avec le thème.

Le logo Guider



Les fonctionnalités pour l'utilisateur sont les suivantes :

- Accès à la page d'accueil
- Accès aux pages arrière-pays/thèmes
- Accès au formulaire de contact
- Création d'un compte utilisateur/connexion au compte utilisateur
- Accès aux fiches des guides
- Visualisation de la localisation des guides
- Réservation d'un guide
- Consultation/Modification des données personnelles
- Consultation des 3 dernières réservation/Annulation dernière réservation

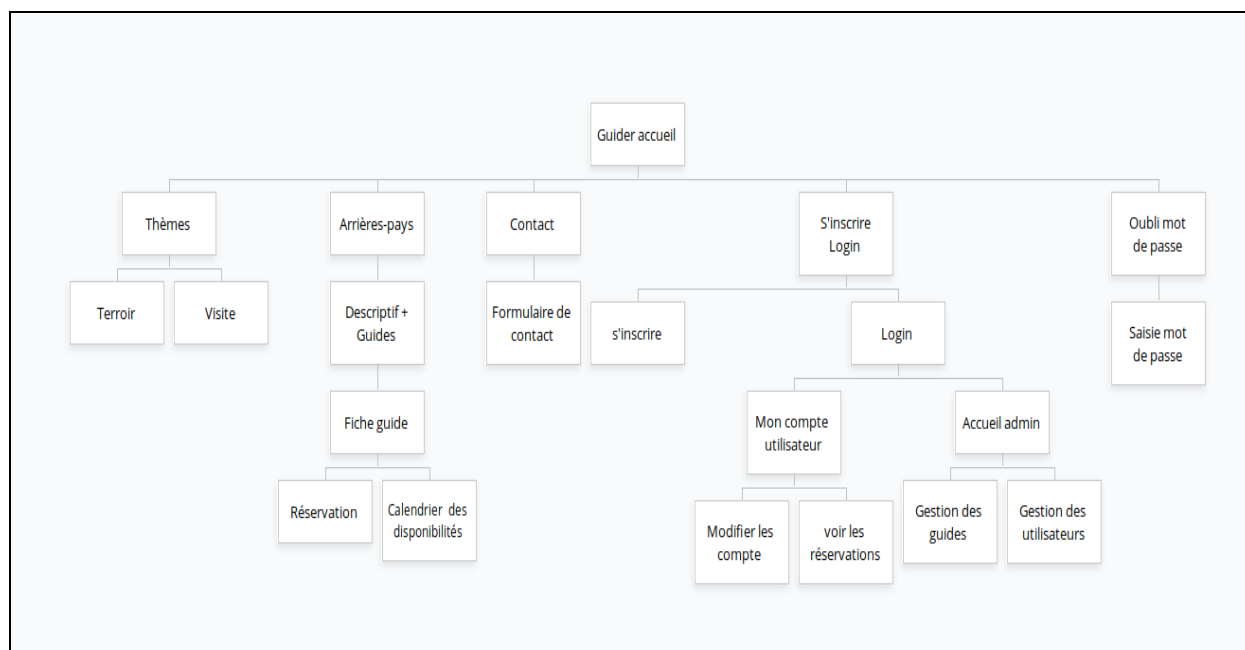
Les fonctionnalités pour les guides sont les suivantes :

- Accès à la page d'accueil
- Accès aux pages arrière-pays/thèmes
- Accès au formulaire de contact
- Connexion compte guide
- Consultation des réservations
- Annulation des réservations

Les fonctionnalités pour l'administrateur sont les suivantes :

- Accès à la page d'accueil
- Accès aux pages arrière-pays/thèmes
- Connexion en tant qu'administrateur
- Accès au panneau de gestion des guides
- Accès au panneau de gestion des utilisateurs

Architecture du site :



Les wireframes ont été créés avec Balsamiq 3 mis en annexe.

2. Conception de la base de données

Je suis partie des fonctionnalités attendues afin de déterminer les éléments dont nous avons besoin pour la base. J'ai construit le schéma suivant avant la création de la base comprenant 6 tables :

«password_reset_request»
-id: int(11)
-#users_id: int(11)
-date_requested: datetime
-token: varchar(255)
-confirmed_at: datetime

«users»
-users_id: int(11)
-users_first: varchar(255)
-users_last: varchar(255)
-users_phone: varchar(12)
-users_email: varchar(50)
-users_pwd: varchar(255)
-users_rank: int(11)

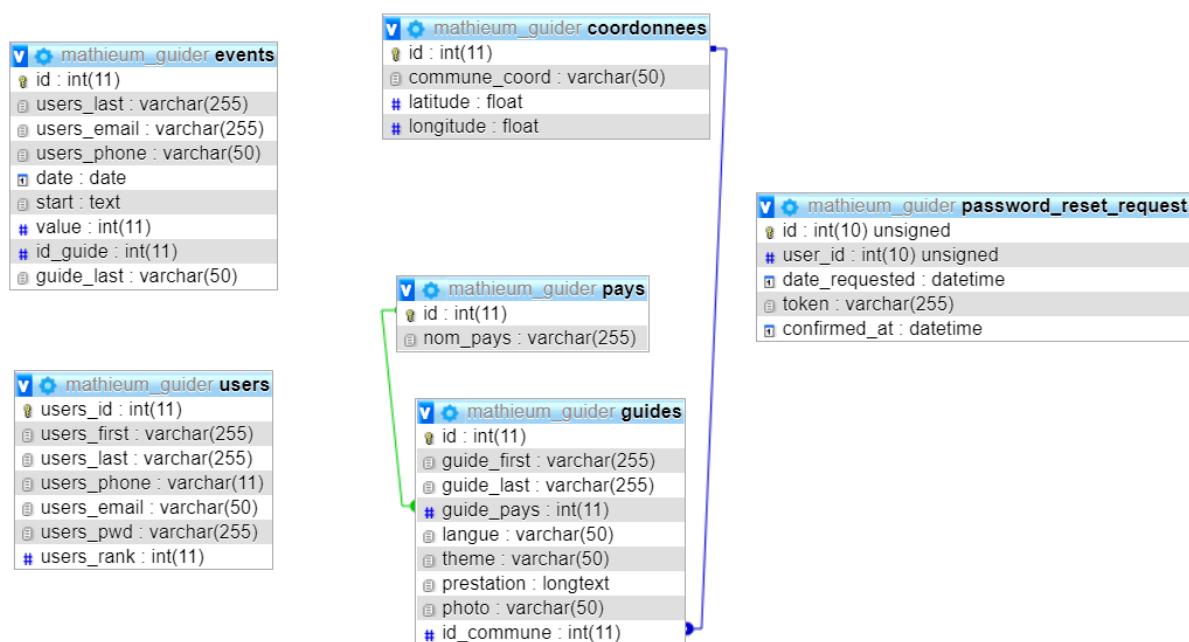
«pays»
-id: int(11)
-nom_pays: varchar(255)

«coordonnees»
-id: int(11)
-commune_coord: varchar(50)
-latitude: float
-longitude: float

«guides»
-id: int(11)
-guide_first: varchar(255)
-guide_last: varchar(255)
-#guide_pays: int(11)
-langue: varchar(50)
-theme: varchar(50)
-prestation: longtext
-photo: varchar(50)
-#id_commune: int(11)

«events»
-id: int(11)
-users_last: varchar(255)
-users_phone: varchar(12)
-users_email: varchar(50)
-date: datetime (default: current_timestamp)
-start: text
-value: int(11)
-#id_guide: int(11)
-guide_last: varchar(50)

Avec **PhpMyAdmin**, J'ai pu bénéficier d'un diagramme relationnel précis qui permet de visualiser les liaisons entre les tables, les clés primaires et étrangères grâce à l'outil « **concepteur** ». Toute modification ou suppression dans la structure des tables sera mise à jour automatiquement dans le « concepteur » :



La table « guides » contient toutes les données sur les guides recrutés ainsi que 2 **clés étrangères** permettant une relation avec les tables « pays » et « coordonnées ».

La table « pays » est une table intermédiaire, chaque pays pouvant regrouper plusieurs guides.

La table « coordonnées » a été créée avec les champs « latitude » et « longitude » qui serviront à positionner les guides et leur commune sur une carte de géolocalisation grâce à l'**API OpenstreetMap – Leaflet**.

La table « events » est utilisée par le **plugin FullCalendar.js** et répertorie toutes les réservations effectuées.

La table « users » possède un champ users_rank qui définit les droits d'accès aux fonctionnalités.

La table « password_reset_request » permettra d'enregistrer toute modification de mot de passe demandée par un utilisateur et gérer l'utilisation du **token** fourni dans le mail de modification.

3. Mise en place de la base de données

J'ai utilisé le serveur **mysql** (sous Lamp pour moi et Xamp pour mon collègue) pour l'intégration de la base de données et l'interface graphique **PhpMyAdmin**.

J'ai tout d'abord créé le script de création de la base de données en langage SQL que vous trouverez en Annexe. J'ai, dans un premier temps, utilisé la base en « **localhost** » afin de développer toutes les fonctionnalités du site. Ensuite, mon collègue et moi avons transféré le site sur un hébergement payant (02switch) et la **base de données distante** a été créée par l'import du fichier SQL via l'onglet « import » de l'interface PhpMyAdmin.

Nous avons autorisé tous les droits pour cet utilisateur grâce au mot clé sql **GRANT** et par la suite nous révoquerons une partie des droits d'accès afin de sécuriser la base de données et ce grâce au mot clé

REVOQUE (géré par le panel du site d'hébergement de mon collègue).

2. Précisez les moyens utilisés :

Je dispose d'un ordinateur portable, sous Linux Ubuntu avec une connexion internet.

Afin de réaliser ce projet, j'ai eu besoin de :

Papier et stylo pour les idées, liste d'idées et ébauche des schémas

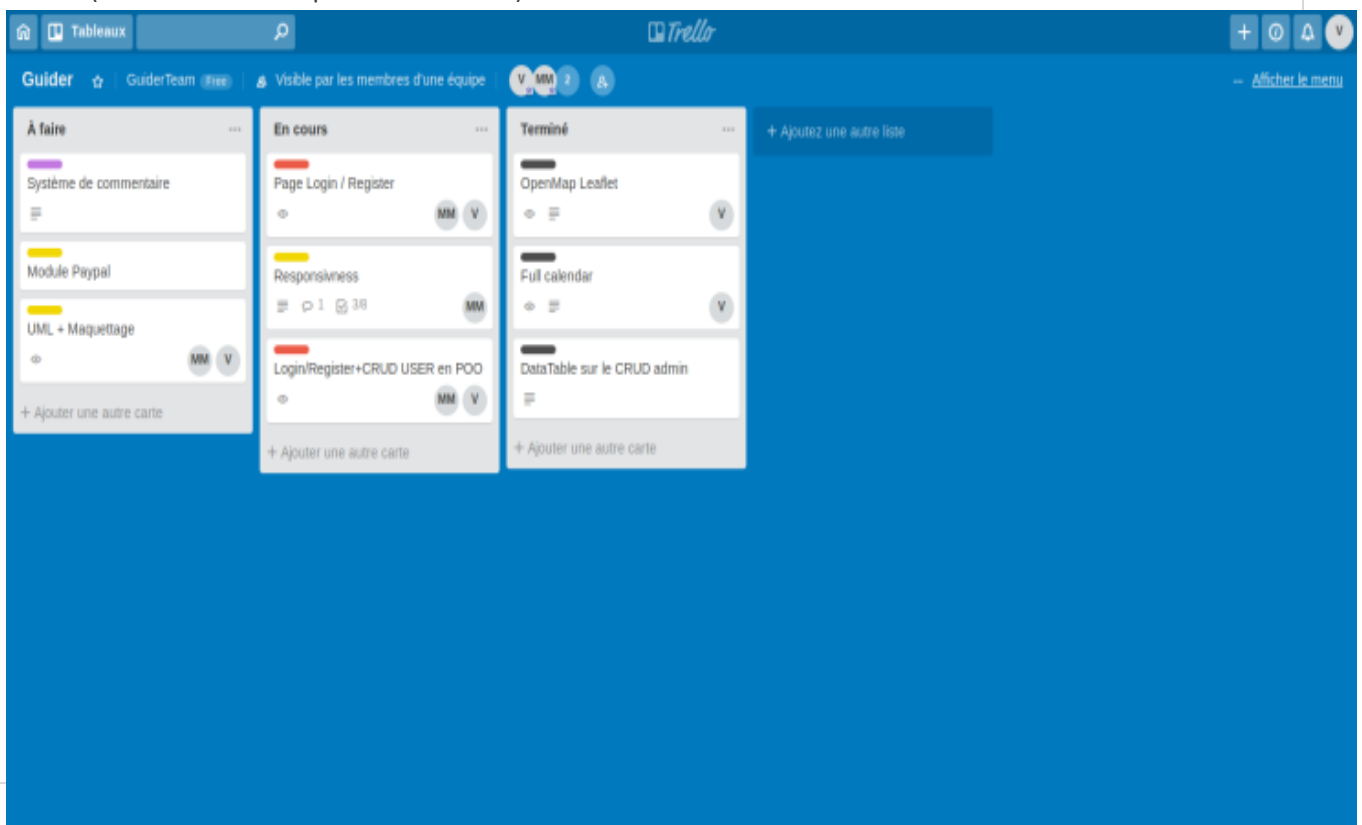
- l'éditeur de texte Sublime Text 3
- un serveur Lamp
- MySQL,
- PhpMyAdmin
- Un hébergement web incluant un panel de gestion (o2switch)

Les maquettes ont été conçues avec :

- Gloomaps (pour l'architecture)
- **Balsamiq 3** (pour les wireframes)
- **UMLet** (Schéma conception de la base de données)
- Photoshop (pour le Logo).

Le suivi de l'avancement a été réalisé avec :

Trello (méthode **Kanban** pour le workflow)



- **Versioning** : les nouveaux fichiers et les modifications ont été sauvegardés sur **Git** et partagés sur **Github**.

3. Avec qui avez-vous travaillé ?

J'ai réalisé cette construction de site avec mon collègue Mathieu Money. Nous avons pensé et fait évoluer le site en rajoutant des fonctionnalités au cours du temps.

Nous avons eu 2 réunions au cours du mois de stage avec le chef de projet, **Mr Clément Seite**, qui nous a demandé de rédiger un cahier des charges et de faire le **mockup** du site, d'utiliser le diagramme de **Gantt** afin de gérer l'évolution du projet et nous donner des deadlines.

Nous avons aussi été conseillés pour le lancement de l'activité, le but du **Sictiam** étant de nous faire connaître auprès des collectivités territoriales et des offices de tourisme.

4. Contexte

Nom de l'entreprise, organisme ou association ► **Accueil du SICTIAM– Ecole Simplon.co Cannes**

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du **04/06/2018** au **05/10/2018**

5. Informations complémentaires (facultatif)

L'organisme appelé **SICTIAM**, qui a récompensé d'un troisième prix l'idée de création de ce site, m'a accueillie avec un collègue dans ses locaux pendant un mois afin de démarrer dans la conception et le développement du projet en toute tranquillité.

Le chef de projet du SICTIAM nous a aiguillé sur la façon d'établir une stratégie, de toucher les bons interlocuteurs, afin de diffuser l'application d'un point de vue commercial et nous a également donné des conseils sur les éléments qui lui paraissaient indispensables pour attirer le maximum de touristes vers les guides locaux dans l'arrière-pays des Alpes Maritimes. Il a également effectué un test « d'ergonomie » sur la partie de l'application qui avait été créée en un mois et a validé notre approche du projet.

Activité-type 1 DEVELOPPER UNE APPLICATION CLIENT-SERVEUR

Exemple n°2 ► Développement Back-end du site Guider

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

Dans ce projet, le système de gestion de l'utilisateur a tout d'abord été construit en langage procédural puis a été modifié pour une utilisation plus simple et éviter la répétition des requêtes SQL de même type. Chaque fonctionnalité prévue lors de la conception du projet est traitée sur une nouvelle page. Le langage que nous avons utilisé est le PHP.

La Classe Database, qui utilise la classe native PDO, a été créée en **Poo** et nécessaire à la connexion à la base de données : Cette classe contient deux méthodes **statiques** de connexion et de déconnexion dont la valeur est stockée dans une variable `$connection`.

```
1 <?php
2
3 class Database
4 {
5     private static $dbHost = "localhost";
6     private static $dbName = "guider";
7     private static $dbUser = "root";
8     private static $dbUserpassword = "valgritim";
9
10    private static $connection = null;
11
12    public static function connect()
13    {
14        try{
15            //echo "try connexion\n";
16            self::$connection = new PDO("mysql:host=" . self::$dbHost . ";charset=utf8;dbname=" . self::$dbName, self::$dbUser, self::$dbUserpassword);
17
18        }
19        catch (PDOException $e)
20        {
21            {
22
23                die($e->getMessage());
24            }
25        }
26        return self::$connection;
27    }
28
29    public static function disconnect(){
30        self::$connection = null;
31    }
32 }
33
34 }
```

Vous trouverez en **annexe 1** une partie de la **Class « User »** mais voici ci-dessous un exemple de fonctionnalité associé à la classe : Création d'un nouvel utilisateur :

```

2
3  require './class/Database.php';
4      // va chercher le fichier database.php pour connexion
5  $db = Database::connect();
6
7  require('./includes/functions.php');
8
9  session_start();
10
11 //post button registration new user
12
13 if(isset($_POST['submitRegister']) && !empty($_POST['submitRegister']))){
14
15     $first = cleanInput($_POST['users_first']);
16     $last = cleanInput($_POST['users_last']);
17     $email = cleanInput($_POST['users_email']);
18     $phone= cleanInput($_POST['users_phone']);
19     $pwd = $_POST['users_pwd'];
20
21     // hashage du password pour sécurité
22
23     $passHash = password_hash($pwd, PASSWORD_DEFAULT);
24
25
26     $user = new User($email,$passHash,$id='', $first,$last,$phone);
27     if($user->exist($db)){
28
29         $user = new User($email,$passHash,$id='', $first,$last,$phone);
30         if($user->exist($db)){
31
32             $_SESSION['errors']['user_exist'] = "Cet utilisateur existe déjà";
33
34             header('Location:login-register.php');
35         } else {
36             $regInput = '/^[a-z éèêñçîï]{2,30})$/i';
37             $regPhone = '/^[+](\d){1}(([\d+() -.]){5,16})([+](\d){1})$/';
38
39             if ((isset($first) && !empty($first)) && (isset($last) && !empty($last))){
40
41                 if(preg_match($regInput, $first) && preg_match($regInput, $last)){
42
43                     if(isset($phone) && !empty($phone)){
44
45                         if(preg_match($regPhone, $phone)){
46
47                             if(filter_var($email, FILTER_VALIDATE_EMAIL)){
48
49                                 if(isset($pwd) && !empty($pwd) && strlen($pwd) >= 8 && strlen($pwd) <= 16){
50
51                                     // ALL verification has been passed, we created a new DB entry
52                                     try{
53
54                                         $user->add($db);
55                                         header('Location:login-register.php');
56                                         //echo "OK en DB <a href='login-register.php'> Retout a La page Login</a>";
57                                     }
58                                     catch(CustomException $e){
59
60                                         die("<br /><strong>Error ! Details: </strong><br />" . $e->getMessage());
61                                     }
62
63                                 }else{
64                                     header('Location:login-register.php'); //work
65                                     //echo "Le pass ou confPass est vide ou n'existe pas ou ne correspond pas au filtre <a href='login-reg
66                                 }
67                             }else{
68                                 header('Location:login-register.php'); //work
69                                 //echo "Le champs email ne correspond pas au filtre <a href='login-register.php'> Retout a La page Login</c
70                             }
71                         }
72                     }
73                 }
74             }
75         }
76     }
77 }

```

Un traitement **POST** est effectué quand le bouton de validation est cliqué. Les champs du formulaire passent tout d'abord dans une fonction de « nettoyage » définie dans un fichier `function.php` (qui permet de supprimer les caractères invisibles en début et fin de chaîne, de supprimer également les anti slashes et convertir les caractères spéciaux en entités HTML).

Un nouvel objet « user » est créé avant vérification des champs vides ou erronés suivant le **diagramme d'activité** suivant :



La **connexion à la base de données** et les **exceptions personnalisées** ont été codées en POO.

L'**exception personnalisée** nous permet de générer des messages d'erreurs personnalisés pour un débogage plus facile des erreurs. Code ci-dessous :

```

<code>
</code>

```

J'ai utilisé une **requête préparée** (fonction de **PDO**). Les **valeurs des paramètres** sont renseignés grâce à la **méthode PDO `bindValue()`** plutôt que d'utiliser la **méthode `bindParam()`** qui n'accepte pas de **méthode de classe** en paramètre. Puis la requête est envoyée au serveur.

Le code **lèvera une exception** dans le cas où la méthode **PDO execute()** renvoie la valeur **FALSE** indiquant que la requête ne peut pas s'exécuter correctement.

La requête ne comprend pas le champ ``users_id`` car celui-ci est une **clef primaire en auto incrémentation** dans la base de données (donc création automatique) et n'est donc pas nécessaire lors de la création.

```
public function add($bdd){
    $addUser = $bdd->prepare('INSERT INTO `users`(`users_first`, `users_last`, `users_phone`, `users_email`, `users_pwd`, `users_rank`) VALUES ( :first, :last, :phone, :mail, :pass, :rank)');
    $addUser->bindValue(':first', $this->getFirst(), PDO::PARAM_STR);
    $addUser->bindValue(':last', $this->getLast(), PDO::PARAM_STR);
    $addUser->bindValue(':phone', $this->getPhone(), PDO::PARAM_STR);
    $addUser->bindValue(':mail', $this->getMail(), PDO::PARAM_STR);
    $addUser->bindValue(':pass', $this->getPass(), PDO::PARAM_STR);
    $addUser->bindValue(':rank', $this->getRank(), PDO::PARAM_INT);

    $exec = $addUser->execute();

    if(!$exec){
        throw new CustomException("An error occurred, please check your query in User.php -> add()");
    }
}
```

La **Class « User »** est également utilisée dans le **CRUD** (Create, Read, Update, Delete) que j'ai créé pour l'administration du site. L'administrateur a seulement pouvoir sur la modification et la suppression des données utilisateur. L'utilisateur gère lui-même la création de son compte et la modification des data dans la partie « mon compte » à laquelle il a accès quand il est connecté.

L'administrateur a également la gestion complète des guides locaux dans un CRUD prévu à cet effet.

Screen du rendu fonctionnel du **CRUD** utilisateur et du CRUD guides:

Guides

Utilisateurs

Liste des utilisateurs

Show 10 entries
 Search:

Nom	Prénom	Email	Téléphone	Droits	Action
Baron	Valérie	valerie.baron67@gmail.com	0609637613	3	Modifier Supprimer
Delcros	Alexis	ale.delcros1142@gmail.com	0613118432	1	Modifier Supprimer

Guides

Utilisateurs

Liste des guides + Ajouter

Show 10 entries
 Search:

Prénom	Nom	Village	Thème	Pays	Action
Adrien	Resparelli	Saint-Cézaire	Promenade	APGrassois	Voir Modifier Supprimer
Alain	Labat	Saint-Dalmas le Sauvage	Culture	Tinée	Voir Modifier Supprimer
Albert	Baron	Tourette-du-Château	Visite	Estéron	Voir Modifier Supprimer
Arnaud	Vialle	Guillaumes	Visite	Var	Voir Modifier Supprimer
Clémence	Antaro	Touët-sur-Var	Visite	Var	Voir Modifier Supprimer
Dominique	Facciola	Valdeblore	Balade	Tinée	Voir Modifier Supprimer
Eric	Charles	Saint Vallier de Thiey	Dégustation	APGrassois	Voir Modifier Supprimer
Jean	Capitan	Lucéram	Visite	Pailhon	Voir Modifier Supprimer
Jeannette	Boutoux	Saint Dalmas de Tende	Visite	Roya	Voir Modifier Supprimer
Joe	Demailly	Touët de l'Escarène	Dégustation	Pailhon	Voir Modifier Supprimer

Showing 1 to 10 of 21 entries
 Previous
 1
 2
 3
 Next

2. Précisez les moyens utilisés :

Même configuration que précédemment.

Mise en place des **algorithmes** sur papier ainsi que la structure de la classe user.

IDE utilisée : SublimeText 3

Versioning : Git, couplé avec **GitHub**

Documentation utilisée pour la **syntaxe** et les propriétés de fonction PHP et PDO: **documentation officielle anglaise** (<https://php.net>) et <https://apprendre-php.com/tutoriels.html> en français.

- Avancement du projet sur **Trello** (méthode **Kanban** pour le workflow).

3. Avec qui avez-vous travaillé ?

Le développement a été réalisé de concert avec mon collègue Mathieu Money de façon à ce nous puissions être capable de comprendre et reproduire le code écrit par l'autre.

4. Contexte

Nom de l'entreprise, organisme ou association ► **Projet chef d'œuvre – Ecole Simplon.co Cannes**

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du **04/06/2018** au **05/10/2018**

5. Informations complémentaires (facultatif)

Le back-end a été développé après la période de stage durant laquelle nous avons plus travaillé sur la conception et le front-end. J'ai effectué une présentation au **Sictiam**, un mois après la fin du stage afin de leur montrer le travail réalisé.

Activité-type 2 DEVELOPPER UNE APPLICATION WEB

Exemple n°1 ► Site Guider- développement du front-end

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

Le site Guider a été développé en **HTML5**, **CSS3** (et son framework **Bootstrap 4.1**), **Javascript ES5** (et son framework **jQuery**) ainsi qu'en **PHP 7.1**. Les **conventions de nommage** (**camelCase**) et les normes **W3C** ainsi que l'**accessibilité** ont été respectés dans la mesure du possible.

Le résultat du test de validation **W3C** de la page d'accueil, test réalisé sur <https://validator.w3c.org> est le suivant :

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://guider.devcanes.com/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<https://guider.devcanes.com/>

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

- Warning** Consider using the `h1` element as a top-level heading only (all `h1` elements are treated as top-level headings by many screen readers and other tools).
From line 189, column 34; to line 189, column 37
`tle-font"><h1>Découv`
- Warning** Consider using the `h1` element as a top-level heading only (all `h1` elements are treated as top-level headings by many screen readers and other tools).
From line 436, column 35; to line 436, column 38
`tle-font"><h1>Trouve`

Document checking completed.

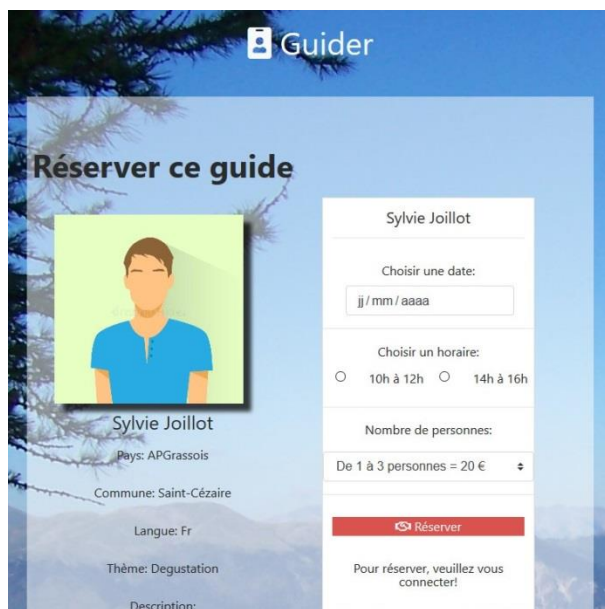
Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 572 milliseconds.

Les deux **avertissements** résultent d'une contrainte graphique. Dans la maquette de conception, nous avons décidé de mettre le **h1** dans un carré destiné à séparer les sections et présenter la section à laquelle elle appartient. Il est donc contenu à l'intérieur d'une sous-partie de la section (`<div>`) : ceci génère l'avertissement.



De la même façon, j'ai dû faire un compromis entre le contraste des **normes d'accessibilité** et le design avec la couleur de fond des sections et la couleur de leur texte. Je dois encore mettre en place un système **TTS (Text To Speech)** afin de permettre aux malvoyants de profiter pleinement de nos services et améliorer l'accessibilité du site.

Le site est construit en **Bootstrap** pour la responsivité, des **@Media queries** ont été rajoutées quand cela a été nécessaire. Ci-dessous le rendu d'une page en format tablette et en format mobile :



Toutes les pages **dynamiques** en Front-end se reposent sur :

- le framework **jQuery** avec sa librairie **jQueryUI** (par exemple : système d'onglet dans la partie admin, animation de disparition de texte dans la partie formulaire d'enregistrement/connexion) Ex : animation de la partie réservation du compte Utilisateur : ajout de la fonctionnalité « **accordion** » par le script suivant :

```
$( function() { $( "#accordion" ).accordion({collapsible: true}); } );
```

Le résultat est le suivant: avant ouverture en slide-down/après ouverture



Les données de réservation sont injectées dans le HTML par requête SQL :

```
<div id="accordion" class="bg-primary text-light mb-2 text-center">
<?php

    require 'class/Database.php';

    $users_last = $_SESSION['user']['LAST'];

    $db = Database::connect();

    $statement = $db->prepare('SELECT * FROM events WHERE users_last=? ORDER BY date DESC LIMIT 3');
    $statement->execute(array($users_last));

    while($item = $statement->fetch()){
        echo '
        <h3>Réservation du ' . date ( 'd/m/Y' , strtotime($item['date'])) . ' </h3>
        <div>
            <ul class="list-group list-group-flush">
                <li class="list-group-item"><strong>Guide réservé: </strong> ' . $item['guide_first'] . ' ' . $item['guide_last'] .
                </li>;
                <li class="list-group-item">Jour de la visite: ' . date ( 'd/m/Y' , strtotime($item['date'])) . ' </li>;
                <li class="list-group-item">Heure de la visite: ' . $item['start'] . ' </li>
            </ul>
        </div>
        ';
    }

    Database::disconnect();
?>
```

-Le plugin **jQuery Datatable** (pour l’affichage des tableaux CRUD), Cdn utilisé et rajout du script suivant dans la partie Html: `<script>$(document).ready(function () {$('#tabs').tabs(); $('#table_id1,#table_id2').DataTable(); }); </script>`

- l’API **Leaflet : Open Street Map** (localisation des guides par pays), API en open source contrairement à Google-Maps qui est devenu payant.

- le plugin **jQuery Validate** (validation dynamique et en temps réel des formulaires)

- et **FullCalendarJs** utilisé pour l’affichage des réservations déjà effectuées pour un guide donné sur sa fiche personnelle de prestation.

L’application est **maintenable**, et sécurisé à la fois pour la société Guider et pour les utilisateurs. Notre code a été organisé de façon logique et les noms de fichiers sont parlants. Je n’ai pas utilisé l’architecture **MVC (Model View Controller)** que n’avions pas encore abordé en cours au moment de la création du site.

Illustration de l’organisation des fichiers en **Annexe 2**

Comment ai-je utilisé et intégré l’API **Leaflet** au site ?

Les *informations de localisation des guides* sont stockées en base de données (latitude et longitude de leur commune) Elles sont récupérées lors de la génération en **PHP** de l’affichage des ‘fiches’ de guide par la requête SQL suivante :

Elles sont ensuite récupérées dans le script **Javascript** et rentrées dans une variable de **type array** afin de pouvoir générer leur affichage sur la carte.

```
function countryListGuides($db,$id){
    // j'ai besoin de 2 paramètres: La variable de connexion à la bd et l'id du pays
    // que je vais récupérer avec le Get Url sur la page pays choisie.

    $statement = $db->prepare('SELECT guides.id, guides.guide_first, guides.guide_last,
        coordonnees.commune_coord AS commune, guides.langue, guides.prestation, guides.
        theme, guides.photo, pays.nom_pays AS pays , coordonnees.latitude, coordonnees.
        longitude FROM guides, coordonnees, pays WHERE guides.id_commune = coordonnees.
        id AND guides.guide_pays = pays.id AND guides.guide_pays = :id');

    $statement->bindValue(":id", $id, PDO::PARAM_INT);
    $statement->execute();

    while ($item = $statement->fetch())
    // prend la première ligne du statement et le met dans le html, s'exécute tant qu'il y a
    // des lignes
    {
        echo '<div class="card js-marker" data-lat = "' . $item['latitude'] . '"
            data-lng = "' . $item['longitude'] . '" data-text = "' . $item['guide_first']
            . ' ' . $item['guide_last'] . '">
            
            <div class="card-body">
            <h4 class="card-title"><strong>' . $item['guide_first'] . '</strong>' . ' ' . ' <
            strong>' . $item['guide_last'] . '</strong>' . '</h4>
            <p class="card-text"><strong>Thème: </strong>' . $item['theme'] . ' <br><strong>
            Vallée: </strong>' . $item['pays'] . ' <br><strong>Village: </strong>' . $
            item['commune'] . ' <br><strong>Langues parlées: </strong>' . $item['langue']
            . '</p>
            </div>
            <div class="card-footer" id="ensavoirplus">
            <a class="btn p-2 rounded" href="includes/pays/view_guide.php?id=' . $item['id'] .
            '">En savoir +</a>
            </div>
            </div>';
    }
}
```

Ici, le passage des données de la base de données au **Javascript** n'a pas nécessité de passage par des langages tel qu'XML ou JSON.

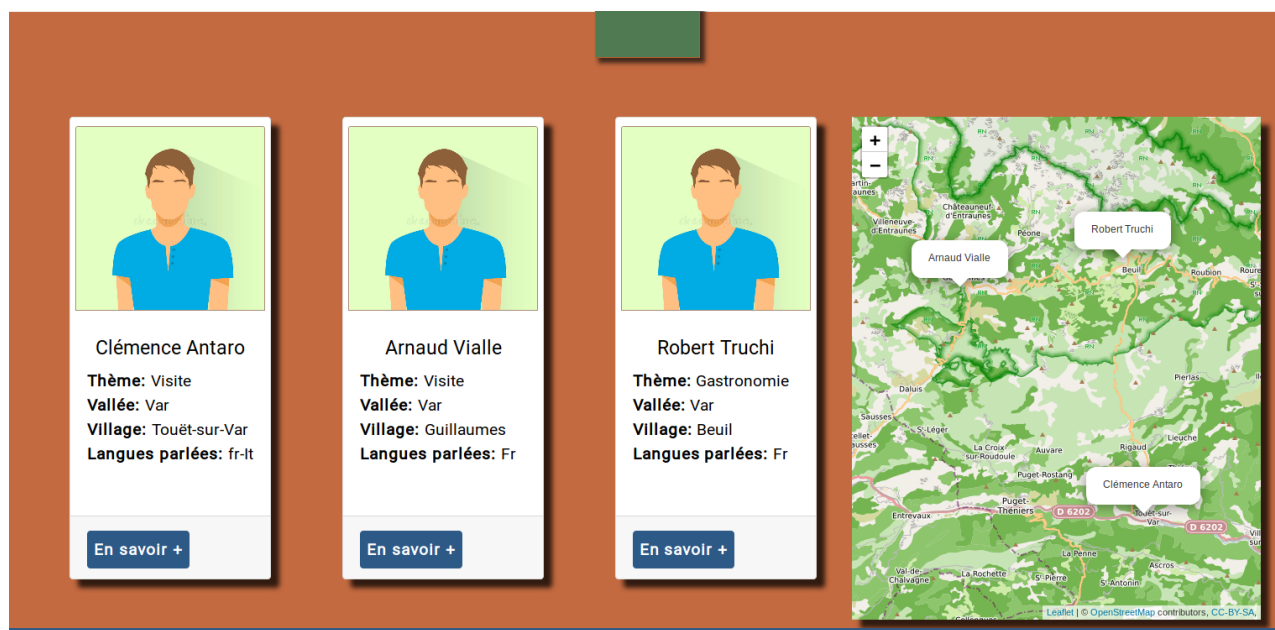
```
const initMap = async function(){
    // // variable servant à afficher la map dans ma div id 'mapid', set view determine l
    // // de zoom choisi

    let map = new leafletMap()
    await map.load($map)
    Array.from(document.querySelectorAll('.js-marker')).forEach((card) => {
        map.addMarker(card.dataset.lat, card.dataset.lng, card.dataset.text)
    })

    map.center()
}

if ($map !== null){
    initMap()
}
```

e résultat obtenu est le suivant :



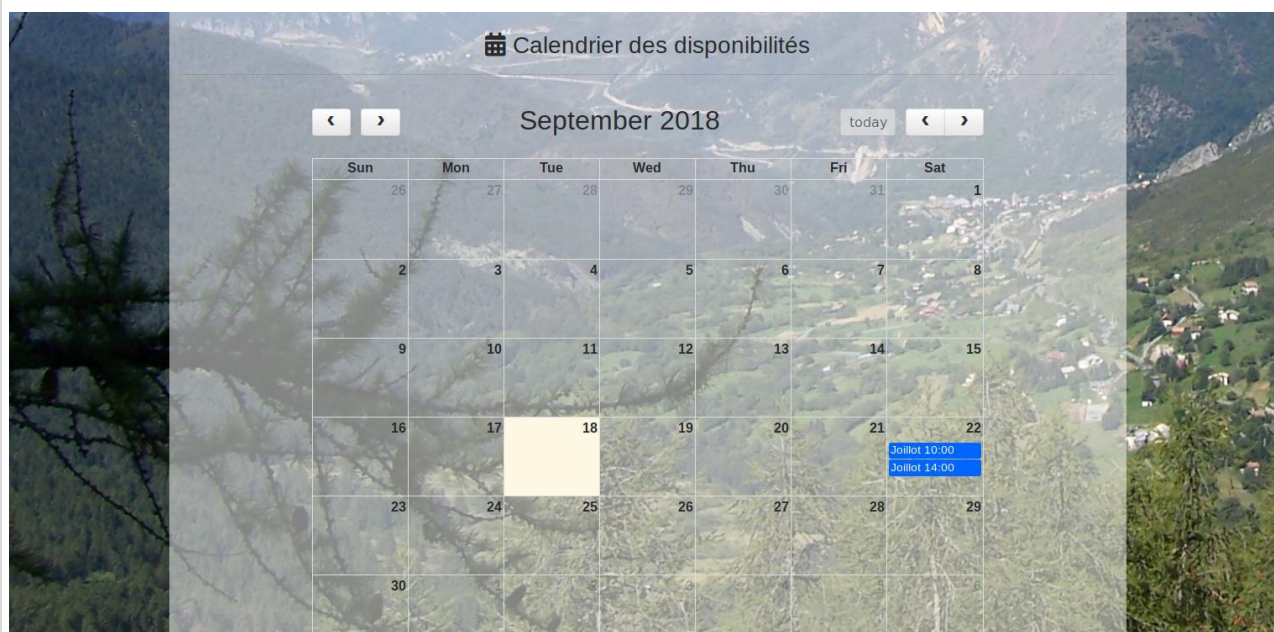
Comment avons-nous utilisé et intégré l'API FullCalendar à notre site ?

```
1 <?php
2
3 session_start();
4
5 require '../class/Database.php';
6
7
8
9 $id = $_SESSION['guide'];
10
11 $data = array();
12
13 // prepare and execute statement query on guide events
14 $db = Database::connect();
15 $statement = $db->prepare('SELECT * FROM events WHERE id_guide = ?');
16 $statement->execute(array($id));
17 $result = $statement->fetchAll();
18
19
20 foreach($result as $row)
21 {
22     // datas that will be implemented into calendar
23     $data[] = array(
24         'id' => $row['id'],
25         'title' => $row['guide_last'] . ' ' . $row['start'],
26         'start' => $row['date']
27     );
28
29 };
30
31 }
32 // echo "<pre>";
33 // $statement->debugDumpParams();
34 // echo "</pre>";
35
36
37 // // conversion data into string to be displayed into event calendar
38 // }
39 echo json_encode($data);
40 // function cleaning datas
41
42
```

Les informations de réservation des guides sont stockées en base de données dans la table « events » contenant le nom, l'e-mail et le téléphone de l'utilisateur, la date et l'heure du rendez-vous, l'id et le nom du guide.

Les données sont récupérées grâce à une **requête SQL** . Le résultat sera stocké dans une variable contenant un **tableau associatif** nommé \$data. Les données seront converties au format JSON par la fonction **json_encode()** et seront récupérées et intégrées par l'API directement dans le layout du calendrier.

Rendu dans FullCalendarJS :



2. Précisez les moyens utilisés :

La configuration machine est la même que cité précédemment.

-IDE utilisée : **SublimeText3**.

J'ai utilisé la **documentation anglaise officielle** de : **PHP** (php.net) et **LeafLet** (https://wiki.openstreetmap.org/wiki/Main_Page).

- Documentation **FullCalendar**,

- Documentation LeafLet OpenStreetMap et suivi d'un tutoriel pour l'installation (Grafikart)

3. Avec qui avez-vous travaillé ?

Tout le développement a été réalisé avec Mathieu Money étant donné que nous étions dans l'optique de montée en compétence, nous avons toujours pris le temps de faire des feed-back sur ce que nous codions afin que les deux personnes puissent bénéficier des connaissances de l'autre.

4. Contexte

Nom de l'entreprise, organisme ou association ► **Sictiam - Ecole Simplon.co Cannes**

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du **04/06/2018** au **05/10/2018**

5. Informations complémentaires (*facultatif*)

La partie front-end a été élaborée durant la période de stage au Sictiam et des fonctionnalités, rendant le site plus dynamique, ont été rajoutées durant la période de projet pédagogique.

Activité-type 2 DEVELOPPER UNE APPLICATION WEB

Exemple n°2 ► Création d'un Thème WORDPRESS demandé par l'école Simplon.co pour une boutique vétérinaire

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

Afin d'utiliser le **CMS Wordpress** en local sur ma machine, j'ai téléchargé et installé le fichier compressé dans var/www/html. J'ai ensuite suivi toutes les instructions indiquées dans le manuel d'installation WP et remplis les écrans d'installation.

Le thème devra être intégré au dossier wp-content /themes/nom du thème créé.

Les fichiers qui ont été ajoutés dans le dossier WP et qui sont nécessaires à la construction du thème sont les suivants :

Nom	Modifié le	Type	Taille
css	20/09/2018 19:52	Dossier de fichiers	
img	20/09/2018 19:52	Dossier de fichiers	
js	20/09/2018 19:52	Dossier de fichiers	
woocommerce	20/09/2018 19:52	Dossier de fichiers	
class-wp-bootstrap-navwalker.php	21/04/2018 17:17	Fichier PHP	21 Ko
footer.php	06/09/2018 14:22	Fichier PHP	1 Ko
functions.php	06/09/2018 14:38	Fichier PHP	3 Ko
header.php	06/09/2018 14:22	Fichier PHP	2 Ko
index.php	18/08/2018 18:57	Fichier PHP	2 Ko
page.php	18/08/2018 18:52	Fichier PHP	1 Ko
screenshot.png	17/08/2018 14:23	Fichier PNG	587 Ko
style.css	06/09/2018 16:01	Document de feui...	3 Ko

Le fichier **CSS** contient la feuille de style **Bootstrap**, le fichier image contient les images qui ont été utilisées dans le thème, le fichier **JS** contient le fichier front.html qui matérialise la page d'accueil du site ainsi que le code Javascript compris dans le framework Bootstrap.

WP reconnaitra mon thème à partir du moment où les fichiers **index.php** et **style.css** existent. Le fichier style doit comporter les éléments suivants pour être pris en compte :

```
/*
Theme Name: WP Bootstrap
Theme URI: https://valeriebaron.website
Author: ValDeMand'
Author URI: https://valeriebaron.website
Description: Responsive bootstrap theme design for e-commerce
website Version: 1.1
License: GNU General Public
License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Text Domain: valeriebaron
*/
```

Le fichier index.php est construit comme suit :


```

<?php get_header(); ?>
<!-- enf of navbar part -->
<main role="main">
    <?php woocommerce_content(); ?>

    <!-- Main jumbotron for a primary marketing message or call to action -->
    <div class="jumbotron">
        <div class="container">
            <h2 class="display-3">Bienvenue chez <?php bloginfo('name'); ?></h2>
            <p><?php bloginfo('description'); ?></p>
            <p><a class="btn btn-primary btn-lg" href="#" role="button">En savoir plus</a></p>
        </div>
    </div>
<!-- end of Jumbotron part -->
<section class="boxes">

    <div class="container">
        <div class="row">
            <div class="col-md-4">
                <div class="box">
                    <i class="fas fa-utensils"></i>
                    <h2>Heading</h2>
                    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. </p>
                    <a class="btn btn-primary" href="#" role="button">View details</a>
                </div>
            </div>
            <div class="col-md-4">
                <div class="box">
                    <i class="fas fa-piggy-bank"></i>
                    <h2>Heading</h2>
                    <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. </p>
                    <a class="btn btn-primary" href="#" role="button">View details &raquo;</a>
                </div>
            </div>
            <aside class="cart col-md-4 p-3 "><?php if(is_active_sidebar('sidebar-1')){dynamic_sidebar( 'sidebar-1' );} ?></aside>
        </div>
    </div> <!-- end of container -->
</section>

</main>
<hr>
<?php get_footer(); ?>

```

Les liens nécessaires à l'inclusion du header sont créés par la ligne de code `<?php wp_head(); ?>` et du footer par la ligne de code `<?php wp_footer(); ?>`.

Pour plus de style, j'ai téléchargé la **WP Bootstrap Navwalker** qui est une barre de navigation facile à installer. Le menu pourra être créé sous **Apparence -> Menu** et être associé au menu principal.

Les informations habituellement « hardcodées » qui concernent le titre du site et sa description sont modifiées dans le fichier `index.php` pour `<div class="container"> <h2 class="display-3">Bienvenue chez <?php bloginfo('name'); ?></h2>` ce qui permettra à l'utilisateur de modifier le titre et la description dans la console d'administration sous **Réglages -> Général**.

Le thème peut déjà être activé à ce stade si une image de la page d'accueil a été chargée. On se positionne sur le menu **Apparence->Thème**.

Tous les liens vers les CDN sont appelés dans un fichier **functions.php** qui sert à ajouter de nouvelles fonctionnalités et à en modifier d'autres déjà présentes dans WP.

Fichier **functions.php** en **Annexe 3**.

Le fichier **page.php** définit la façon dont les pages sont affichées :

```
<?php
/**
 * The template for displaying all pages.
 *
 * This is the template that displays all pages by default.
 * Please note that this is the WordPress construct of pages
 * and that other 'pages' on your WordPress site will use a
 * different template.
 *
 * @package VW Hair Salon
 */

get_header(); ?>

<?php do_action( 'wpbootstrap_page_top' ); ?>

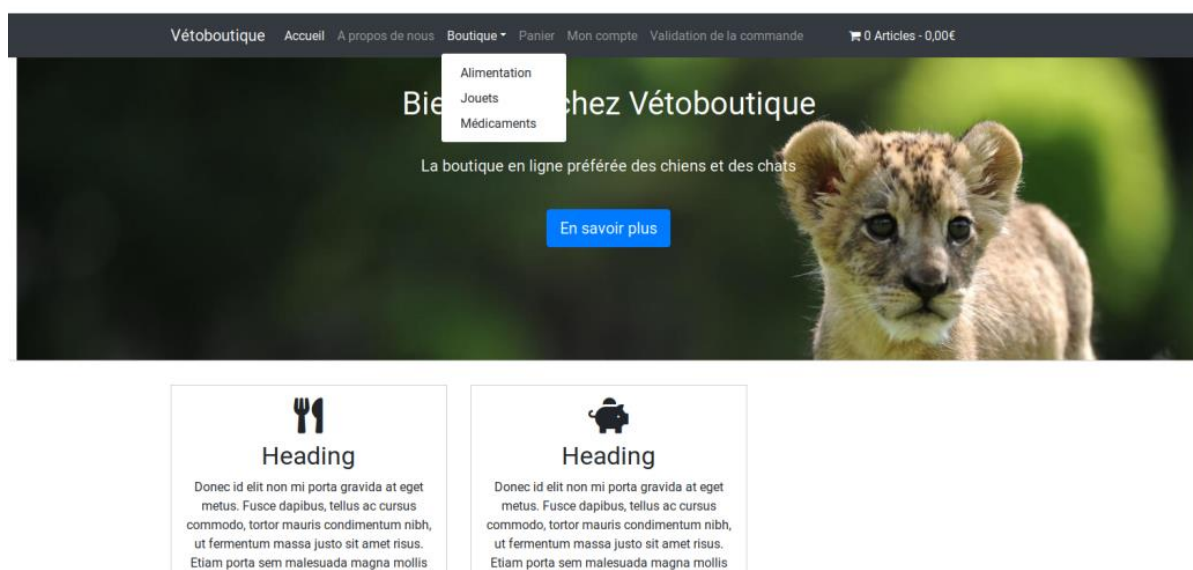
<div id="content-vw" class="container">
    <div class="middle-align">
        <?php while ( have_posts() ) : the_post(); ?>
            
            <h1><?php the_title();?></h1>
            <?php the_content();?>
        <?php endwhile; // end of the loop. ?>
    <?php
        //If comments are open or we have at least one comment, load up the comment template
        if ( comments_open() || '0' != get_comments_number() )
            comments_template();
    ?>
</div>
<div class="clear"></div>
</div>

<?php do_action( 'wpbootstrap_page_bottom' ); ?>

<?php get_footer(); ?>
```

La **boucle WP** est utilisée pour afficher des messages. Elle peut être utilisée pour afficher un seul article ou tous les messages publiés sur une seule page.


Au final, le thème WP que j'ai créé est le suivant :



Le site est alimenté par une base de données qui contient les produits vétérinaires proposés à la vente.
Le téléchargement du **plugin WOOCOMMERCE** a permis de créer la boutique en ligne.

Vétoboutique Accueil À propos de nous Boutique Panier Mon compte Validation de la commande 1 Article - 7,75€

Panier

Produit	Prix	Quantité	Total
 Crocodile en latex	7,75€	1	7,75€

Code promo [Appliquer le code promo](#) [Mettre à jour le panier](#)

Total panier


Sous-total	7,75€
Expédition	Forfait: 10,00€ Calculer les frais d'expédition
Total	17,75€

[Valider la commande](#)


© 2018 - Vétoboutique
[Back to top](#)

Un exemple de produit vendu en ligne est annexé en **Annexe 4**.

Le téléchargement du **plugin** de paiement **STRIPE** a été également intégré, il permet donc le paiement en ligne à partir du compte utilisateur.

Carte de paiement (Stripe) 

Payez avec votre carte bancaire avec Stripe. **MODE TEST ACTIVÉ.** En mode test, vous pouvez utiliser le numéro de carte 4242424242424242 avec n'importe quel cryptogramme visuel et une date d'expiration valide ou consulter la [documentation Test Stripe](#) pour obtenir plus de numéros de carte.

Numéro de carte * 

Date d'expiration * Cryptogramme visuel *

☐ Paiement à la livraison

Vos données personnelles seront utilisées pour le traitement de votre commande, vous accompagner au cours de votre visite du site web, et pour d'autres raisons décrites dans notre [politique de confidentialité](#).

☐ J'ai lu et j'accepte les [conditions générales](#) *

[Commander](#)

© 2018 - Vétoboutique

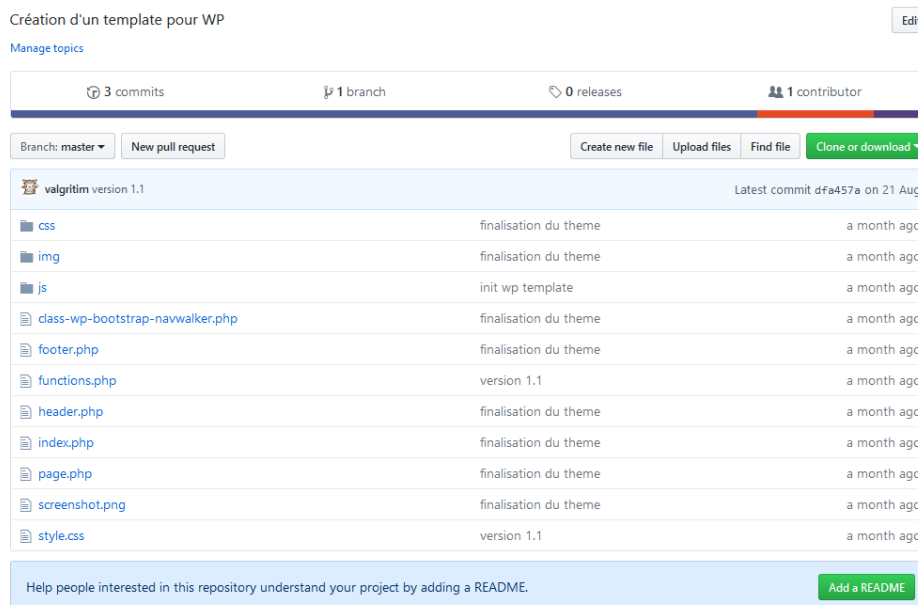
Pour l'utiliser il faut d'abord créer son compte **STRIPE** et rentrer des données obligatoires. Un **certificat SSL** est proposé à la vente, mais il peut être aussi proposé par certains hébergeurs mutualisés (par ex. 1 & 1). Je n'ai pas de certificat étant donné que le site n'est pas destiné à être mis en ligne.

Je suis passée directement à l'installation et l'activation de l'extension via **WooCommerce**. Pour lier WooCommerce et **STRIPE** nous avons besoin des clés API pour que les deux applications puissent communiquer entre elles. 4 clés sont fournies sur le compte **STRIPE** créé, j'ai utilisé les clés nécessaires au **mode test** (Elles sont copiées/collées dans WooCommerce).

Il est possible de procéder à un **paiement factice** en ajoutant un produit dans le panier et valider la commande. Le paiement par carte bancaire est immédiatement disponible et les données bancaires de

simulations saisies sont les suivantes : 42424242424242 Visa. Si tout est opérationnel, **STRIPE** indique que le paiement a bien été effectué et la commande bascule dans les commandes du compte utilisateur.

Au fur et à mesure de l'avancée du projet, les informations ont été « pushé » sur Git et Github :



2. Précisez les moyens utilisés :

Je dispose d'un ordinateur portable, sous Linux Ubuntu avec une connexion internet.

Afin de réaliser ce projet, j'ai eu besoin de :

- Papier et stylo pour les idées, liste d'idées et ébauche des schémas
- L'éditeur de texte Sublime Text 3
- un serveur Lamp
- MySQL,
- PhpMyAdmin
- Le CMS Wordpress
- L'outil de versioning Git

Des tutoriels ont été utilisés pour la création du projet : cours sur Udemy, le site hostinger.fr/tutoriels.

3. Avec qui avez-vous travaillé ?

Le travail a été réalisé en solo et sous forme de **SPRINT** de 2 jours (en méthode agile). Le formateur est allé récupérer le travail effectué sur Github grâce au lien qui lui a été fourni.

4. Contexte

Nom de l'entreprise, organisme ou association ▶	<i>Evaluation en cours de formation – Ecole Simplon.co Cannes</i>		
Chantier, atelier, service ▶	Cliquez ici pour taper du texte.		
Période d'exercice ▶	Du	16/08/2018	au 17/08/2018

5. Informations complémentaires (facultatif)

Domage que nous n'ayons pas eu plus de temps pour affiner le rendu du module WooCommerce.

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] *Valérie Baron*..... ,
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 *Mandelieu-la-Napoule*..... le *10/10/2018*.....
pour faire valoir ce que de droit.

Signature :

ANNEXE 1

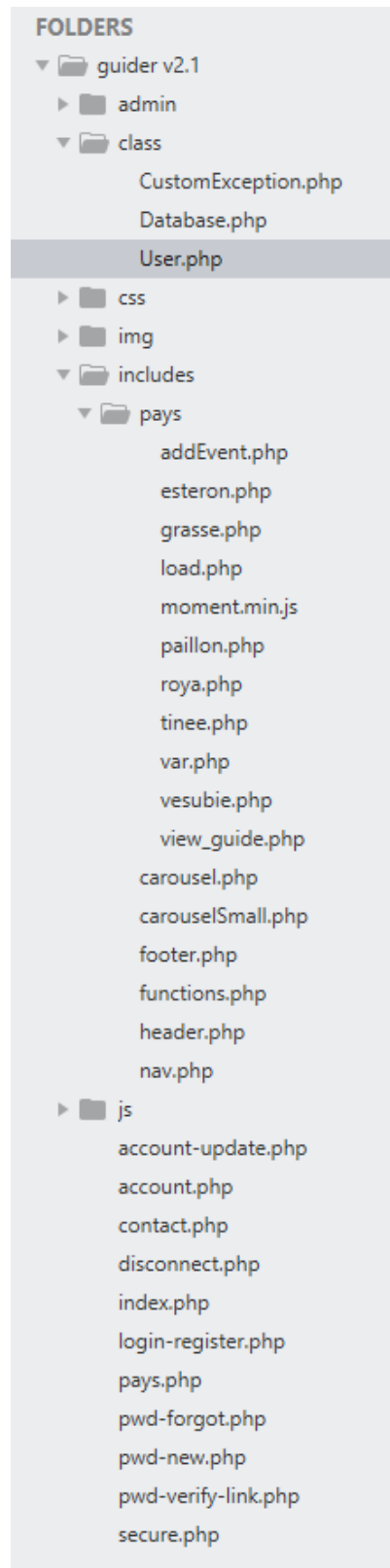
```
1 <?php
2 /**
3  *
4  * Class User
5  * @attributes : mail,password,[id,[first,[last,[phone,[rank]]]]]
6  *
7  *
8  */
9 class User
10 {
11     private $_id, $_first, $_last, $_phone, $_mail, $_password, $_rank;
12
13     function __construct($mail,$password,$id='', $first='', $last='', $phone='', $rank=1)
14     {
15         $this->setMail($mail);
16         $this->setPass($password);
17         $this->setId($id);
18         $this->setFirst($first);
19         $this->setLast($last);
20         $this->setPhone($phone);
21         $this->setRank($rank);
22     }
23
24     // Accesseurs/Mutateurs
25     public function getId(){
26         return $this->_id;
27     }
28
29     public function getFirst(){
30         return $this->_first;
31     }
32
33     public function getLast(){
34         return $this->_last;
35     }
36
37     public function getPhone(){
38         return $this->_phone;
39     }
40
41     public function getMail(){
42         return $this->_mail;
43     }
44
45     private function getPass(){
46         return $this->_password;
47     }
48     private function getRank(){
49         return $this->_rank;
50     }
51
52     public function setId($id){
53         $this->_id = $id;
```

```

79 //Methods
80
81 // Class method to authenticate a user when sign-in.
82 public function authenticate($bdd){
83     // We check if we have any errors generated during a previous attempt and unset those
84     // error if necessary.
85     if (isset($_SESSION['errors']['user_LogError']) && !empty($_SESSION['errors']['
86         user_LogError'])) {
87         unset($_SESSION['errors']['user_LogError']);
88     }
89
90     // We select into the database all entries which match with the email that user has
91     // entered.
92     $auth = $bdd->prepare("SELECT * FROM `users` WHERE `users_email` = :mail");
93     $auth->bindValue(':mail', $this->getMail());
94     $exec = $auth->execute();
95
96     // If our query sends an error (code 400), we throw a custom exception.
97     if (!$exec)
98         throw new CustomException("An error occurred, please check your query in User.php
99         -> authenticate()");
100
101     // We count our results.
102     $result = $auth->rowCount();
103
104     // If we have only one result, we try to create a session else we redirect to the
105     // login page.
106     if ($result == 1) {
107         $user = $auth->fetch();
108
109         // We check if the password that user has entered match with which is into the
110         // database.
111         $verifPassword = password_verify($this->getPass(), $user['users_pwd']);
112         // If it's true, we create the session. Otherwise, we redirect the user to the
113         // login page.
114         if ($verifPassword) {
115             $_SESSION['user']['ID'] = $user['users_id'];
116             $_SESSION['user']['MAIL'] = $user['users_email'];
117             $_SESSION['user']['FIRST'] = $user['users_first'];
118             $_SESSION['user']['LAST'] = $user['users_last'];
119             $_SESSION['user']['PHONE'] = $user['users_phone'];
120             $_SESSION['user']['RANK'] = $user['users_rank'];
121             $_SESSION['user']['PASS'] = $user['users_pwd'];
122             $_SESSION['user']['CONNECT'] = Date("Y-m-d-H-i-s");
123             $file = '../logs/connections.txt';
124             $newline = "\nConnexion : ".$_SESSION['user']['CONNECT']." - ".$_SESSION['user']
125                 ['ID']." ".$_SESSION['user']['MAIL']." Statut : OK ";
126             file_put_contents($file, $newline, FILE_APPEND | LOCK_EX);
127             // file put content: utilisé à la place de file open, file write et file close
128             header('Location: index.php');
129         } else {

```


ANNEXE 2




ANNEXE 3

```
1 <?php
2 // Define constant version
3 define('VERSION','0.0.4');
4
5 // chargement des scripts
6
7 function wpbootstrap_scripts(){
8
9     //wp_enqueue_style('wpbootstrap_custom', get_template_directory_uri() . '/style.css',
10     array(), '1.0.0', 'all');
11
12     //Register bootstrap CDN:
13     wp_register_style('wpbootstrapBoot','https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/
14     bootstrap.min.css',array(), VERSION,'all');
15     //Enqueue Bootstrap CDN
16     wp_enqueue_style('wpbootstrapBoot');
17     //Register FontAwesome 5
18     wp_register_style('wpbootstrapFa','https://use.fontawesome.com/releases/v5.2.0/css/all.css',
19     array('wpbootstrapBoot'), VERSION,'all');
20     //Enqueue FontAwesome
21     wp_enqueue_style('wpbootstrapFa');
22     //Register Custom CSS
23     wp_register_style('wpbootstrapStyle',get_template_directory_uri().'/style.css',array('
24     wpbootstrapFa'),VERSION,'all');
25     //Enqueue Custom CSS
26     wp_enqueue_style('wpbootstrapStyle');
27     // Custom fonts
28     wp_enqueue_style('wpbootstrapFont','https://fonts.googleapis.com/css?family=Roboto',array('
29     wpbootstrapStyle'),VERSION,'all');
30
31     //Register and enqueue Scripts
32     /*
33     Enqueue Script last param : $in_footer, default value FALSE
34     */
35     wp_enqueue_script('wpbootstrapPopper-js', 'https://cdnjs.cloudflare.com/ajax/libs/
36     popper.js/1.14.3/umd/popper.min.js', array('jquery'), VERSION, true);
37     wp_enqueue_script('wpbootstrapBstrap-js', 'https://stackpath.bootstrapcdn.com/
38     bootstrap/4.1.3/js/bootstrap.min.js', array('wpbootstrapPopper-js'), VERSION, true);
39
40 }
41 //hook (crochet pour appeler les fonctions callback)
42 add_action('wp_enqueue_scripts', 'wpbootstrap_scripts');
43
44 function wpbootstrapAdmScripts(){
45     wp_enqueue_style('wpbootstrapBstrap', 'https://stackpath.bootstrapcdn.com/
46     bootstrap/4.1.3/css/bootstrap.min.css', array('jquery'), VERSION, 'all');
47     wp_enqueue_style('wpbootstrapAdmFa', 'https://use.fontawesome.com/releases/v5.2.0/css/
48     all.css', array('jquery'), VERSION, 'all');
49     wp_enqueue_script('wpbootstrapAdmPopper-js', 'https://cdnjs.cloudflare.com/ajax/libs/
```



ANNEXE 4



Véto Boutique

Accueil

Alimentation

Jouets


Médicaments

Offres spéciales

S'inscrire


Se connecter

Votre panier




Crocodile caoutchouc
jouet pour chien
6 €
Quantité en stock: 10

Commander



Cochons en corde
jouet pour chat
5 €
Quantité en stock: 10

Commander



Singe en tissu
jouet pour chien
8 €
Quantité en stock: 10

Commander

Administrateur

35