



# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*

► BRAHAM

*Nom d'usage*

► BRAHAM

*Prénom*

► Mohamed Amin

*Adresse*

► 115 boulevard Jourdan, Paris 14

## Titre professionnel visé

Développeur Web et Web mobile

### MODALITE D'ACCES :

- 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.*



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

# DOSSIER PROFESSIONNEL (DP)

## Sommaire

### Exemples de pratique professionnelle

|   |    |    |
|---|----|----|
| <b>Activité n° 1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.</b> | p. | 6  |
| ▶ Compétence n° 1 : Maquetter une application .....   | p. | 6  |
| ▶ Compétence n° 2 : Réaliser une interface utilisateur web statique et adaptable .....  | p. | 10 |
| ▶ Compétence n° 3 : Développer une interface utilisateur web dynamique .....  | p. | 15 |
| ▶ Compétence n° 4 : Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce .....                      | p. | 20 |
| <b>Activité n° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.</b>  | p. | 25 |
| ▶ Compétence n° 5 : Créer une base de données .....   | p. | 25 |
| ▶ Compétence n° 6 : Développer les composants d'accès aux données .....   | p. | 36 |
| ▶ Compétence n° 7 : Développer la partie back-end d'une application web ou web mobile .   | p. | 41 |
| ▶ Compétence n° 8 : Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce .....           | p. | 54 |
| <b>Intitulé de l'activité-type n° 3</b>   | p. |    |
| ▶ Intitulé de l'exemple n° 1 .....  | p. |    |
| ▶ Intitulé de l'exemple n° 2 .....  | p. |    |
| ▶ Intitulé de l'exemple n° 3 .....  | p. |    |
| <b>Intitulé de l'activité-type n° 4</b>   | p. |    |
| ▶ Intitulé de l'exemple n° 1 .....  | p. |    |
| ▶ Intitulé de l'exemple n° 2 .....  | p. |    |
| ▶ Intitulé de l'exemple n° 3 .....  | p. |    |
| <b>Titres, diplômes, CQP, attestations de formation (facultatif)</b>  | p. |    |

# DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

p. 66

Documents illustrant la pratique professionnelle (*facultatif*)

p. 67

Annexes (*Si le RC le prévoit*)

p. 68

# **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

*Compétence n°1 ▶ Maquetter une application*

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

J'ai débuté par créer une **planche de tendance (Moodboard)**, avec le logiciel **Gimp** et **Adobe PhotoShop**, où l'on retrouve le logo, les couleurs, le slogan et quelques illustrations qui seront intégrées au projet.

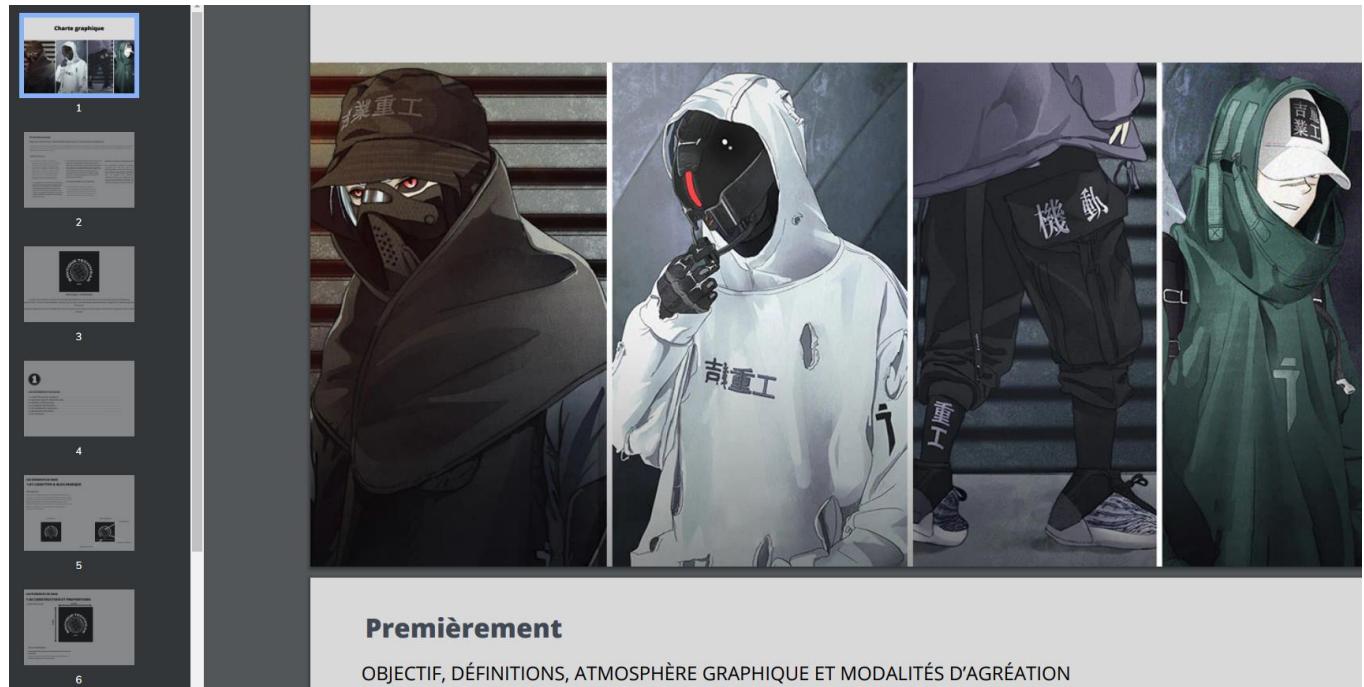
Ce qui va permettre aux clients(es) d'avoir un rendu visuel qui correspond à leurs attentes sur le contenu de leur projet.



Figure 1 : le MoodBoard

Ensuite j'ai créé une **charte graphique** sur Powerpoint ce qui consiste à détailler tous les points à respecter qui regroupe l'ensemble des codes définissant l'identité visuelle du projet.

# DOSSIER PROFESSIONNEL (DP)



## Premièrement

OBJECTIF, DÉFINITIONS, ATMOSPHÈRE GRAPHIQUE ET MODALITÉS D'AGRÉATION

*Figure 2 : La charte graphique*

L'objectif de mon projet est de créer un site E-commerce au tours de la niche du Techwear ( Streetwear Japonais ). Inspiré par la culture japonaise et les codes streetwear occidentaux. Boutique Techwear™ combine sur son shop le meilleur des deux mondes, en apportant aux vêtements de rue une touche unique, représentant la **nouvelle streetculture asiatique**.

À travers des collections originales et innovantes, des thèmes hauts en couleur allant du fluo aux effets réfléchissants, des pièces avant-gardistes et inspirantes, Boutique Techwear™ souhaite, en toute humilité, **marquer son empreinte dans l'univers du streetwear français**.

J'ai créé un slogan accrocheur, simple à retenir avec plusieurs dimensions et plusieurs couleurs, en commençant par le noir qui représente parfaitement la niche du streetwear, et la couleur blanche pour donner un coté minimaliste et simple, finalement la couleurs bleu marine pour ajouter des couleurs dans le site tout en respectant les codes de la niche.

Le logo correspond parfaitement à la niche et à l'arborescence du site grâce à son style qui se définit comme simple et futuriste.

Et puis j'ai créé la **maquette** de la page d'accueil du projet en me servant du logiciel **Figma** pour la mise en page, et pour redimensionner les images et les icônes, tout en respectant les codes définis dans la planche de tendance et la charte graphique.

# DOSSIER PROFESSIONNEL (DP)

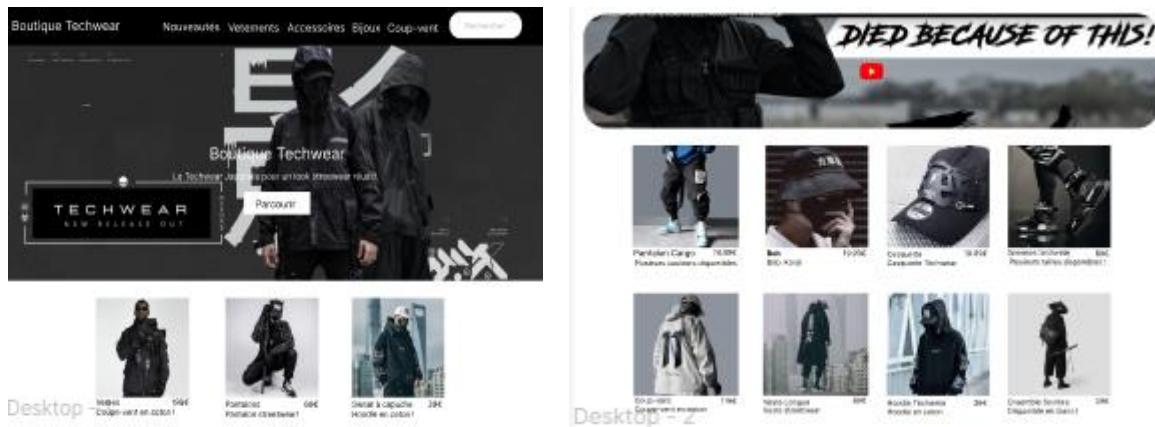


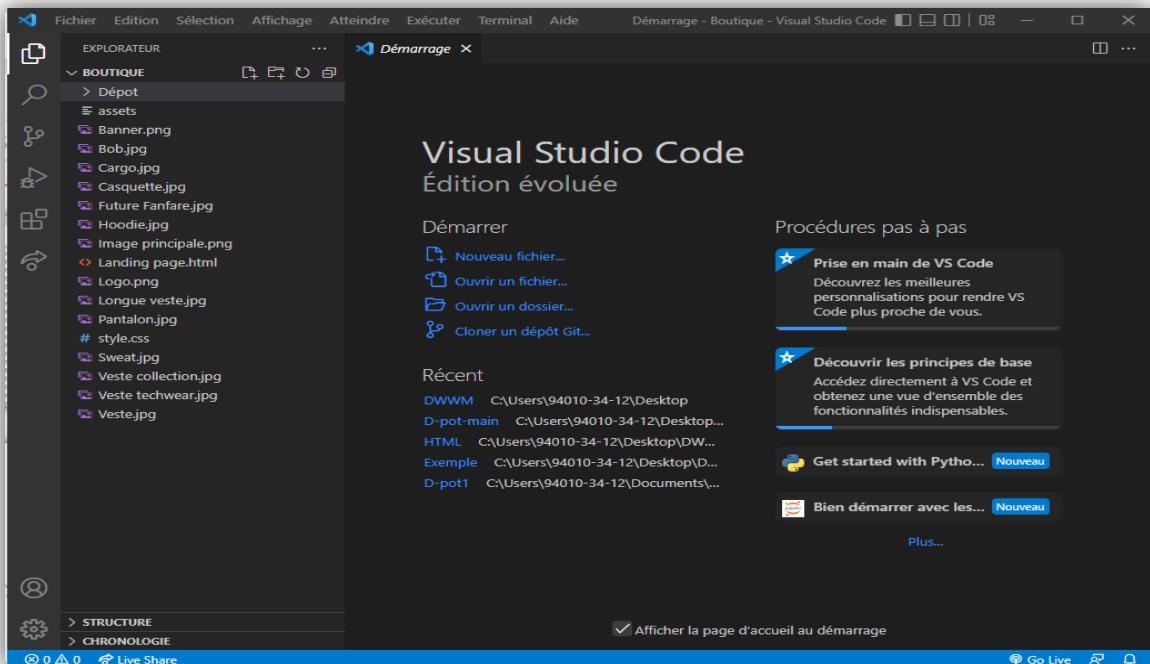
Figure 3 : Maquette de la page d'accueil de mon projet DoDoleGuide

Pour finir avec cette compétence j'ai commencé à coder mon projet en **HTML 5** (*HyperText Markup Language*) et en **CSS 3** (*Cascading Style Sheets*) avec l'éditeur de code **Visual Studio Code** et pour sauvegarder l'avancée de mon travail en créant des dépôts j'ai utilisé **Github Desktop**.

```
10  <!-- Barre de navigation -->
11  <nav>
12    <h1>Boutique Techwear</h1>
13    <div class="onglets">
14      <p class="link">Nouveautés</p>
15      <p class="link">Vêtements</p>
16      <p class="link">Accessoires</p>
17      <p class="link">Bijoux</p>
18      <p class="link">Coup-vent</p>
19      <form>
20        <input type="search" placeholder="Rechercher">
21      </form>
22      <p><i class="far fa-heart"></i></p>
23      <p><i class="fas fa-shopping-cart"></i></p>
24    </div>
25  </nav>
26  <!-- Fin de la barre de navigation -->
27
28  <!-- Header -->
29  <!-- Header -->
30  <header>
31    <h1>Boutique Techwear</h1>
32    <h3>Le Techwear Japonais pour un look streetwear réussi.</h3>
33    <button>Parcourir</button>
34  </header>
35
36  <!-- Fin du header -->
```

Figure 4 : Code HTML

# DOSSIER PROFESSIONNEL (DP)



```
# style.css  x
# style.css > ...
85 .cards{
86   display:flex;
87   flex-wrap:wrap;
88 }
89 .cards .card{
90   margin-right: 20px;
91   cursor:pointer;
92 }
93 .cards .card img{
94   width: 350px;
95 }
96 .cards .card .card-header{
97   display: flex;
98   flex-wrap:wrap;
99   justify-content: space-between;
100 }
101 .cards .card .card-body p{
102   margin-top: -10px;
103 }
104 /* Fin de toutes les cartes */
105
106 /* Video de présentation */
107 .main .video{
108   margin-top: 80px;
109   width: 90%;
110 }
111 .main .video iframe{
112   border:none;
113   border-radius: 10px;
114   width: 100%;
115   height: 400px;
116 }
117 .footer{
118   margin-top: 40px;
119   border-top: 5px solid #6f6f6f;
```

Figure 5 : Code CSS

Figure 6 : Dépôt de Visual Studio Code vers Github Desktop

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette première compétence :

Logiciel: Gimp,, Visual studio code, Canva.

Site web: Github desktop.

Application: Figma.

Langage: Html5, Css3.

## 3. Avec qui avez-vous travaillé ?

Pour les compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *23/09/2022* au : *29/09/2022*

## 5. Informations complémentaires (*facultatif*)

**Activité-type 1** Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

*Compétence n°2 ► Réalisation d'une interface utilisateur web statique et adaptable*

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

Pour cette deuxième compétence j'ai commencé par **maquetter** mon projet en format **mobile, tablette et desktop** toujours avec le logiciel **Figma**.

# DOSSIER PROFESSIONNEL (DP)

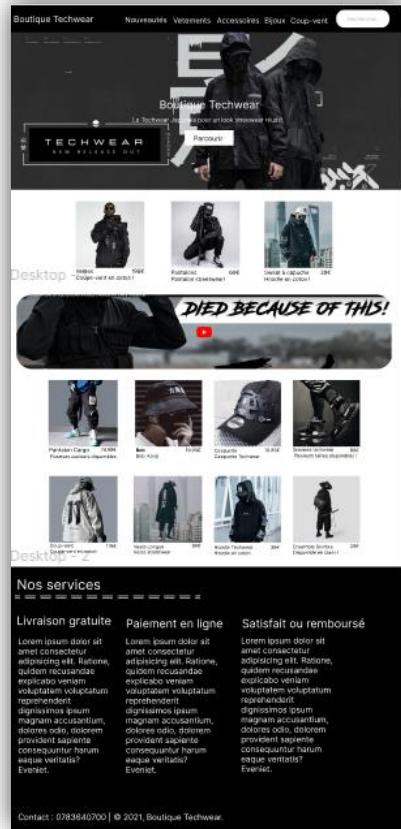


Figure 7 : La maquette au format Mobile, Tablette et Desktop

Et par la suite avec l'éditeur de code **Visual Studio Code**, j'ai continué le codage de la page d'accueil de mon projet, en y ajoutant les **média queries** pour le rendre adaptable sur tous les types de supports : Mobile, Tablette et Desktop.

# DOSSIER PROFESSIONNEL (DP)

```
<style>
/*MEDIA QUERIES*/
@media screen and (min-width: 320px) and (max-width: 374px) {
    /* API BOITE METEO PANORAMIQUE */
    .boitemeteo0,.boitemeteo6,.boitemeteo9 {
        border: 1px solid #f1f1f1;
        background: linear-gradient(45deg, #1069b659, #36a33269);
        border-radius: 25px 25px 22px 25px/25px 25px 25px;
        box-shadow: 0 .5rem 1rem rgba(0,0,0,.15)!important;
        display: grid;
        grid-template: 70px 30px 30px / repeat(7, 37px);
        height: 72px;
        margin-top: 76px;
        margin-bottom: 80px;
        margin-left: -75px;
        opacity: 0;
        position: relative;
        width: 264px;
    }
    .bloc-logo-info.pano {
        align-items: center;
        display: flex;
        grid-area: 1 / 1 / 2 / -1;
        justify-content: center;
    }
}

@media screen and (min-width: 768px) and (max-width: 991px) {
    h1 {
        margin-top: 350px;
        margin-bottom: 80px;
        font-size: 28px;
    }
    h2 {
        margin-top: 24px;
        font-size: 24px;
    }
    .col-12 {
        margin-bottom: 25px;
    }
    .pano {
        margin-bottom: 25px;
    }
    .espace {
        height: 5px;
        margin-bottom: 5px;
    }
}

@media screen and (min-width: 1024px) and (max-width: 1439px) {
    .container {
        max-width: 1018px;
    }
    h1 {
        margin-top: 400px;
        margin-bottom: 80px;
        font-size: 32px;
    }
    h2 {
        display: flex;
        position: relative;
        justify-content: center;
        margin-top: 24px;
        font-size: 25px;
    }
    .col-12 {
        margin-bottom: 22px;
    }
}
```

Figure 8 : Les média quéries

J'ai intégré du contenu multimédia, exemple une Vidéo provenant de "youtube" pour accentuer la visibilité de mon site par les utilisateurs.

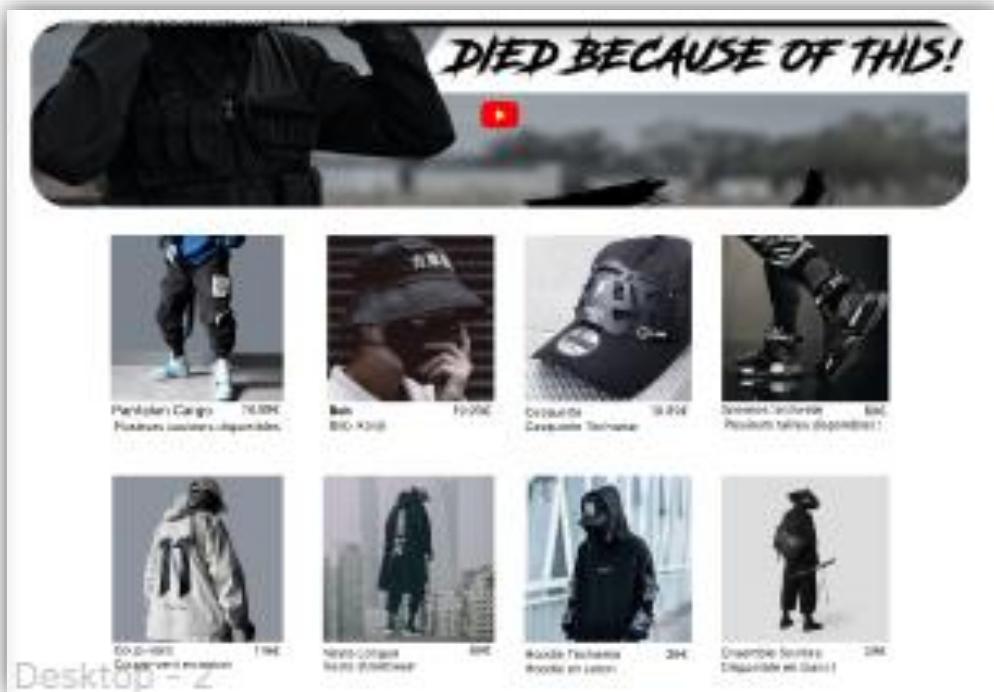


Figure 9 : Une vidéo youtube

# DOSSIER PROFESSIONNEL (DP)

J'ai aussi intégré un **menu hamburger** à mon bandeau de navigation (*nav-bar*), pour faciliter l'accès aux informations disponibles sur le site.

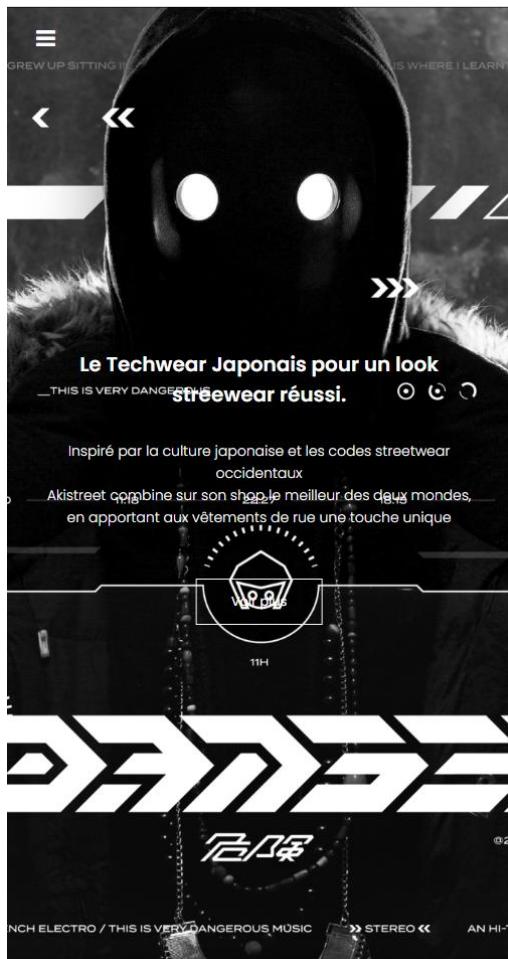


Figure 10 : Menu hamburger

J'ai également intégré des icônes cliquables dans mon bandeau de pied de page (*footer*), qui redirigent vers les **réseaux sociaux** du type : Twitter, Instagram et Facebook pour offrir une plus grande visibilité à ce projet sur le web.

## À propos de nous

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque aliquet turpis nulla, eleifend faucibus est sollicitudin ut. Maecenas ut venenatis ex, et dapibus purus.



Créé par ❤ Mohamed Amin Braham

Figure 11 : Icône réseaux sociaux

Puis j'ai créé des **personas** avec le logiciel **Figma**, correspondant aux profils d'utilisateurs qui pourraient se rendre sur le site.

# DOSSIER PROFESSIONNEL (DP)

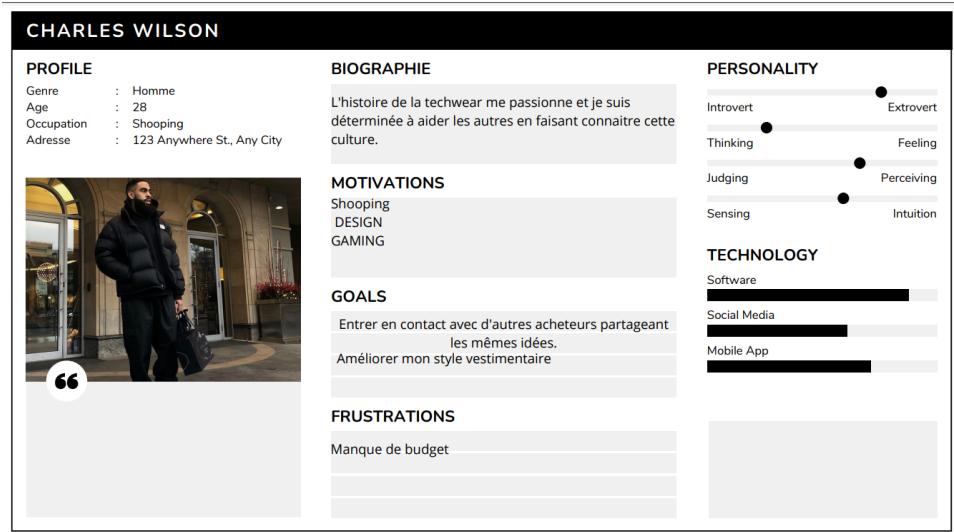


Figure 12 : Les personas

J'ai poursuivi le codage du site en intégrant du code **SCSS**. Avec **SASS** (*Syntactically awesome stylesheets*) on peut imiter la structure du html il faut commencer par intégrer un flux de travail de compilateur Sass à votre projet. Ce qui permet d'obtenir des goodies exemple : conception modulaire, fichiers inclus, mixins, héritage, et de pouvoir mieux se repérer dans le code en ayant moins de lignes en introduisant des variables.

# DOSSIER PROFESSIONNEL (DP)

```
nav{  
    display: flex;  
    padding: 2% 4%;  
    justify-content: space-between;  
    align-items: center;  
}  
nav img{  
    width: 135px;  
}  
.nav-links{  
    flex: 1;  
    text-align: right;  
}  
.nav-links ul li{  
    list-style: none;  
    display: inline-block;  
    padding: 8px 12px;  
    position: relative;  
}  
.nav-links ul li a{  
    color: ■#fff;  
    text-decoration: none;  
    font-size: 13px;  
}  
.nav-links ul li::after{  
    content: '';  
    width: 0%;  
    height: 2px;  
    background: □#2c2626;  
    display: block;  
    margin: auto;  
    transition: .5s;  
}  
.nav-links ul li:hover::after{  
    width: 100%;  
}  
nav .fa{  
    display: none;  
}
```

Figure 13 : Code Scss

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette deuxième compétence:

Logiciel: Visual studio code.

Site web: Github desktop.

Application: Figma.

Langage: Html5, Css, JavaScript

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 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

Compétence n°3 ▶ Développer une interface utilisateur web dynamique

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

Pour la première partie de cette compétence(3A), toujours à partir de l'éditeur de code **Visual Studio Code** j'ai utilisé **vue.js** qui est un framework qui permet de simplifier l'action du développeur web en lui donnant accès à une multitude de codes déjà prêts à être intégrés à son projet pour la navigation et les éléments interactifs de celui-ci.

This screenshot shows a Visual Studio Code interface with several tabs open. The left sidebar displays a tree view of a Vue.js project structure, including 'AKSIRET-VUE', 'public', 'src', 'router', 'store', 'views', 'components', 'jsontata', 'assets', and 'index.html'. The 'index.html' tab is active, showing the source code for a basic HTML page with meta tags, a title, and a body containing a note about JavaScript being required for certain features. The bottom status bar shows tabs for 'PROBLÈMES', 'SORTIE', 'CONSOLE DE DÉBOGAGE', and 'TERMINAL'. The terminal tab shows output related to a development build. A small note at the bottom of the terminal says 'Note that the development build is not optimized. To create a production build, run npm run build.'

Figure 14 : Page d'accueil de la documentation, et, un exemple de code pour une Nav-bar avec menu-hamburger

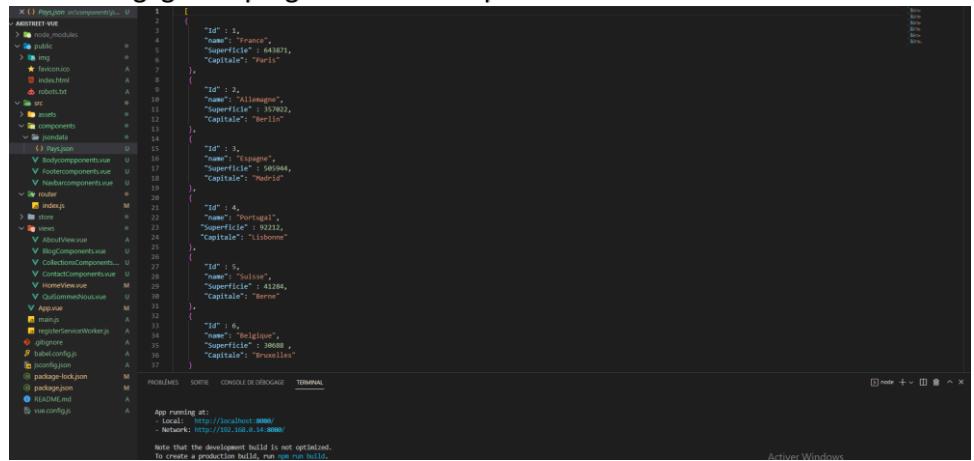
Figure 15 : Un formulaire de contact interactif

# DOSSIER PROFESSIONNEL (DP)

```
<template>
  <div class="header">
    <nav class="navbar dark-mode" role="navigation">
      <div class="navbar_logo"></div>
      <i class="fa fa-close" onclick="hideMenu()"></i>
      <ul class="navbar_links">
        <li class="navbar_link first"><a href="#">Akastreet</a></li>
        <li class="navbar_link second"><a href="#">Section2</a><router-link to="QuiSommesNous">Qui sommes-nous</router-link></a></li>
        <li class="navbar_link third"><a href="#"><router-link to="/CollectionsComponents">Collections</router-link></a></li>
        <li class="navbar_link fourth"><a href="#"><router-link to="/BlogComponents">Blog</router-link></a></li>
        <li class="navbar_link fifth"><a href="#"><router-link to="/ContactComponents">Contact</router-link></a></li>
      </ul>
      <i class="fa fa-bars" onclick="showMenu()"></i>
    </nav>
  <main class="main-content" role="main">
    <div id="section2"><ncr></div>
    <div class="text-box">
      <h1>Le Techwear Japonais pour un look streetwear réussi.</h1>
      <p>Inspiré par la culture japonaise et les codes streetwear occidentaux.  
Akastreet combine sur son shop le meilleur des deux mondes, en apportant aux vêtements de rue une touche de style et de qualité japonaise.</p>
      <a href="contact.html" class="hero-btn">Voir plus...</a>
    </div>
  </main>
</template>
<script>
export default {
  name: 'Navbarcomponents',
}
</script>
```

Figure 16 : Le vue.js

Ensuite pour la seconde partie de cette compétence(3B), toujours à partir de l'éditeur de code **Visual Studio Code**, j'ai repris mon projet précédent en y intégrant du **Json (JavaScript Object Notation)** qui permet de stocker, lire et partager tous les langages de programmations disponibles.



```
[{"id": 1, "name": "France", "superficie": 643871, "capitale": "Paris"}, {"id": 2, "name": "Allemagne", "superficie": 357002, "capitale": "Berlin"}, {"id": 3, "name": "Pays-Bas", "superficie": 360944, "capitale": "Amsterdam"}, {"id": 4, "name": "Espagne", "superficie": 499558, "capitale": "Madrid"}, {"id": 5, "name": "Portugal", "superficie": 92212, "capitale": "Lisbonne"}, {"id": 6, "name": "Italie", "superficie": 301338, "capitale": "Rome"}, {"id": 7, "name": "Suisse", "superficie": 41284, "capitale": "Berne"}, {"id": 8, "name": "Belgique", "superficie": 30688, "capitale": "Bruxelles"}, {"id": 9, "name": "Irlande", "superficie": 70273, "capitale": "Dublin"}]
```

Figure 17 : Vue.js

Voici le rendu de ma page d'accueil avec mon projet **Vue.js**.

# DOSSIER PROFESSIONNEL (DP)

## Pourquoi nous ?

... sit amet, consectetur adipiscing elit.

### Qualité irréprochable

... sit amet, consectetur adipiscing elit. ...

### Livraison sous 3 jours

... sit amet, consectetur adipiscing elit. ...

### Achetez une histoire

... sit amet, consectetur adipiscing elit. ...

## Nos collections

... sit amet, consectetur adipiscing elit.



Figure 19 : Visuel de mon projet Vue.js-Argon

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette troisième compétence :

Documentation: Site de Bootstrap, vue.js

Logiciels: Visual studio code.

Site web: Github desktop.

Application: Figma.

Framework: Bootstrap, Vue.js.

Langage : Html5, Scss3, Javascrip, Json.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **Formation AFPA**

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

Période d'exercice ► Du : **05/10/2020** au : **11/06/2021**

# DOSSIER PROFESSIONNEL (DP)

## 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

*Compétence n°4 ▶ Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce*

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

Pour cette quatrième compétence j'ai utilisé **Wordpress** qui se trouve dans la catégorie des **Cms (Content Management Système)**.

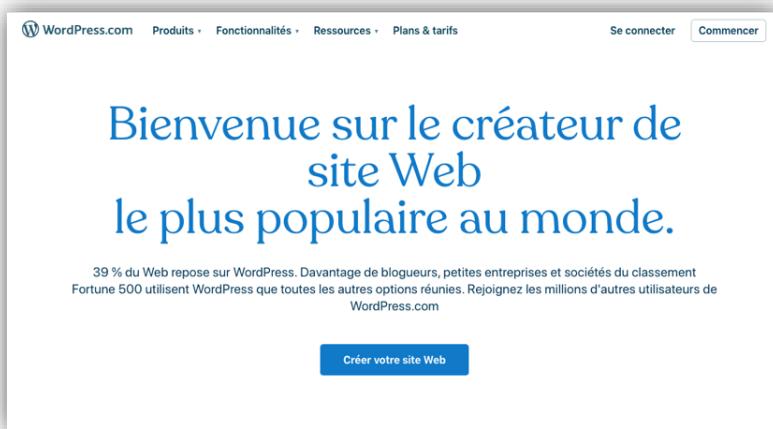


Figure 20 : Page d'accueil WordPress

J'ai donc commencé par installer (**Mamp (Mac)**, **Xampp (Universel)**, ou **Wamp(Windows)**) qui me permet de créer une **base de données** en interprétant des scripts, et d'avoir accès au **PhpmyAdmin** pour l'administration Web des bases **MySql** (*système de gestion de base de données relationnelle open source*) et de pouvoir développer et exécuter des sites Web dynamiques, et de mettre en place un serveur Web local, un serveur FTP ou un serveur de messagerie électronique.

# DOSSIER PROFESSIONNEL (DP)



Figure 21 : Mamp, Xampp et Wamp

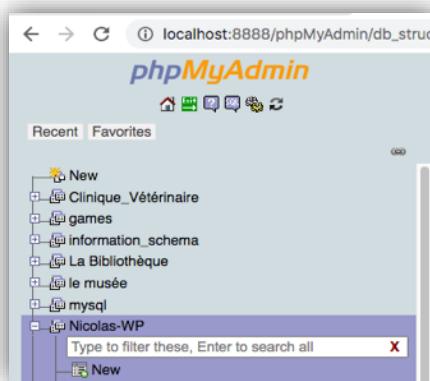


Figure 22 : Crédation d'une base de données

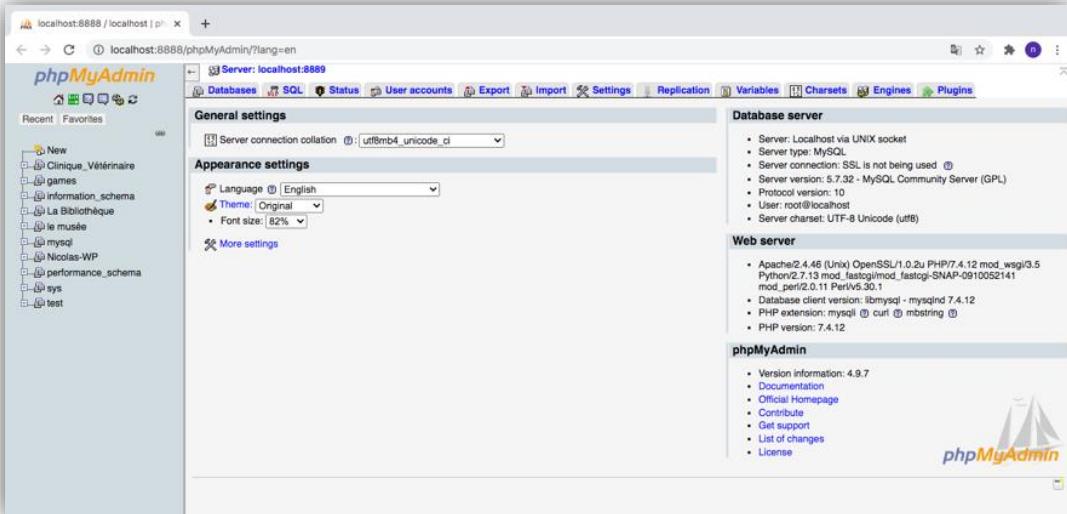


Figure 23 : Page du server local Php My Admin

# DOSSIER PROFESSIONNEL (DP)

Après avoir créée une base de données j'ai démarré mon **localhost** sur le **webserver Apache**, ce qui m'a permis d'accéder à Wordpress et tous ses outils.

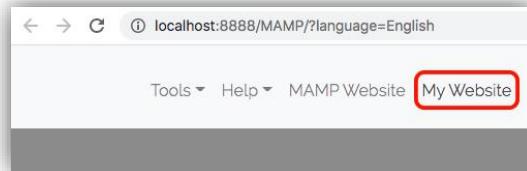


Figure 24 : Lancement de la base de données

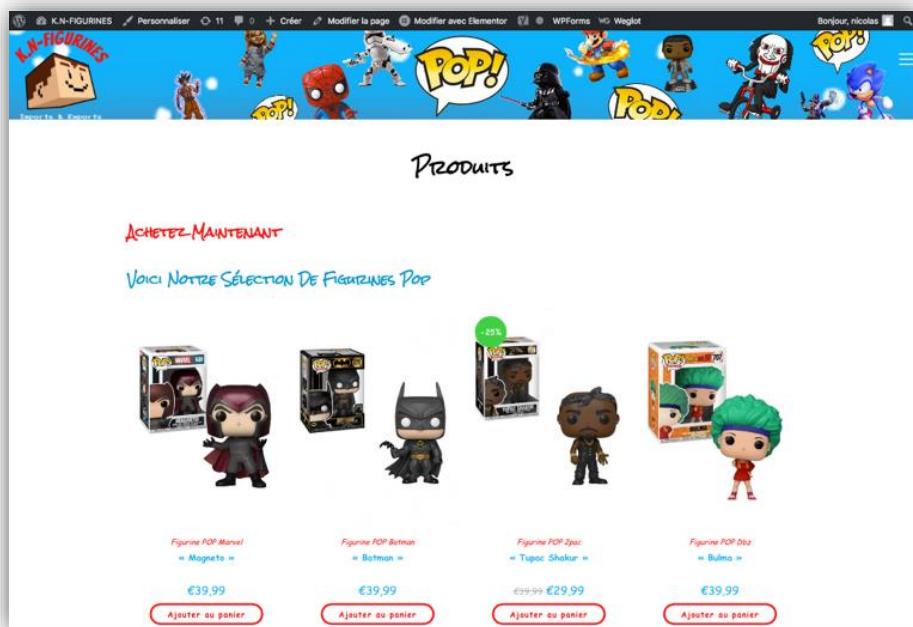


Figure 25 : Mon projet WordPress

# DOSSIER PROFESSIONNEL (DP)



Figure 26 : Tableau de bord de mon projet WordPress

J'ai poursuivi en créant un **thème enfant** à partir du **thème parent OcéanWP** ce qui me permet de ne pas subir une mise à jour du thème parent ce qui engendrerait des modifications irréversibles sur mon projet, et, qui serait une perte de temps.

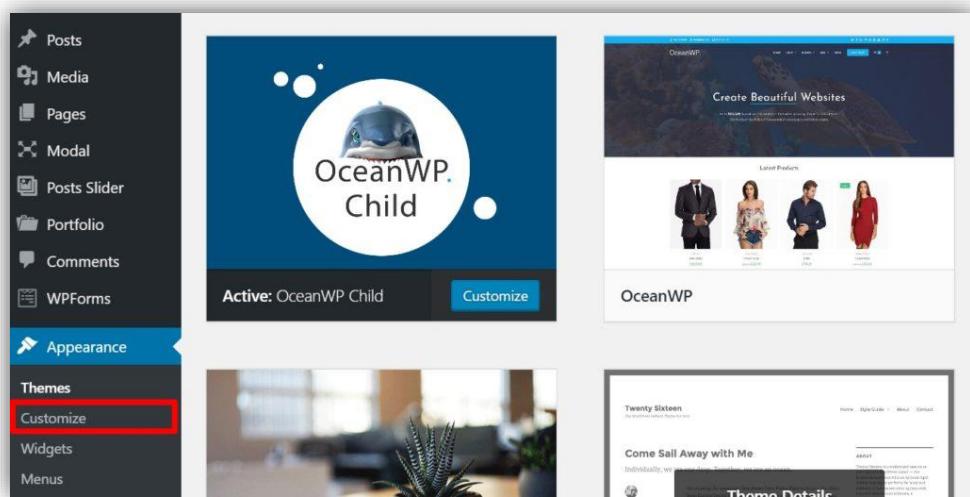


Figure 27 : Crédit du thème enfant

# DOSSIER PROFESSIONNEL (DP)

A la suite je me suis lancé dans la création de mon interface Ecommerce en installant des plugins :

- **WooCommerce** qui permet de faciliter le commerce en ligne en apportant des outils indispensables.
- **Weglot** qui permet la traduction de mon site web dans toutes les langues.
- **Wpforme** qui permet de créer des formulaires de contact ou autre....



Figure 28 : Les différents plugins utilisés dans mon projet

Et je me suis servi de nombreux autres plugins déjà intégrés dans Wordpress du type **OcéanWP/Elementor** qui permet de modifier pratiquement tout ce que l'on souhaite sur notre interface d'une manière simplifiée.

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette quatrième compétence :

Documentation: Site de WordPress.

Logiciel: Visual studio code, Mamp, Mysql.

Site web: Github desktop.

Cms: WordPress.

Application: PhpMyAdmin.

Framework: Bootstrap, Argon Vue.js.

Bibliothèque: Jquery.

Languge: Html5, Scss3.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 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.

Compétence n°5 ▶ **Créer une base de données**

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

Pour cette première compétence de la partie **Back-end**, j'ai commencé par comprendre le fonctionnement d'un site web et savoir faire la différence entre la partie clients et la partie serveurs, j'ai également appris à différencier un site **Statique** d'un site **Dynamique**.

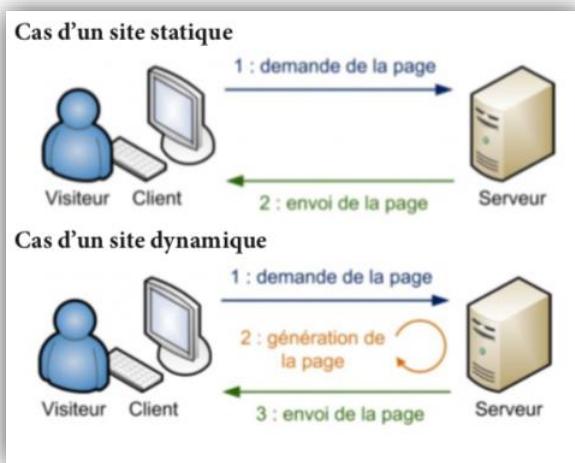


Figure 29 : La différence entre un site statique et un site dynamique

Sur un site Statique le serveur stocke des pages web et les envoie aux clients qui les demandent sans les modifier. Alors que sur un site Dynamique la page web est générée à chaque fois qu'un client la réclame, cela rend son contenu vivant il peut changer d'un instant à l'autre. Tout cela est possible grâce au langage **Php** et au Système de gestion de bases données relationnel (**SGBDr**)  **MySql** .

# DOSSIER PROFESSIONNEL (DP)

Ensute j'ai installé **Mamp** (mac), **Xampp** (universel) ou **Wamp** (windows) pour accéder à **MyPhpAdmin** ce qui m'a permis de créer une base de données pour faire fonctionner mon code Php.



Figure 30 : Mamp, Xampp et Wamp

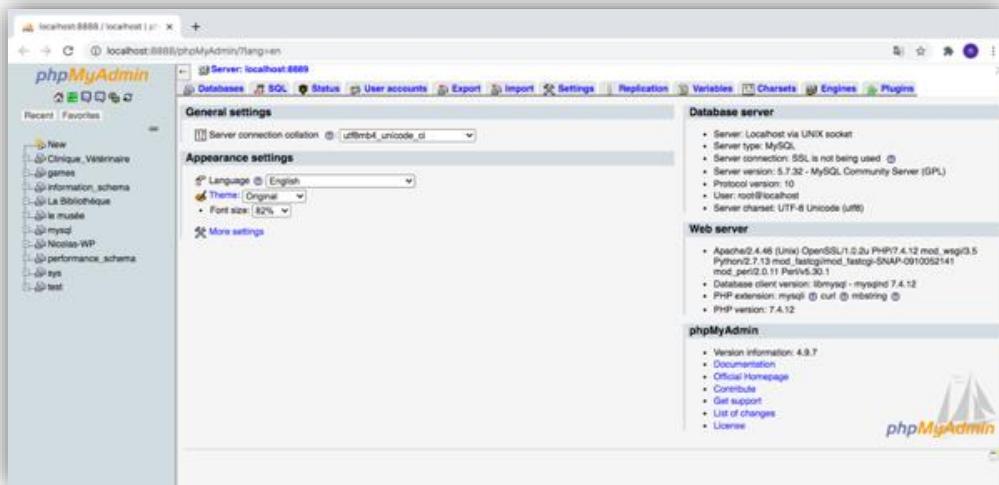


Figure 31 : Page serveur local Php My Admin

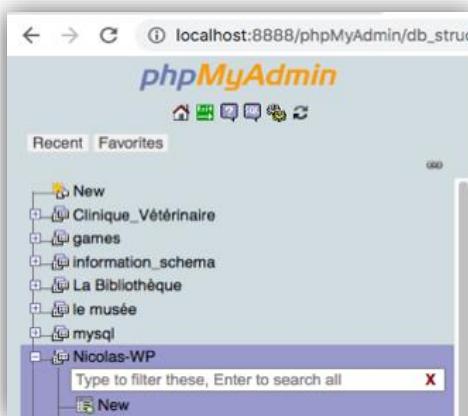


Figure 32 : Création base de données

# DOSSIER PROFESSIONNEL (DP)

The screenshot shows a code editor interface with a sidebar labeled "EXPLORATEUR" containing a tree view of files. Under "SEANCE1\_PHP-POO-LES\_BASSES", there are several "classes" files: Animal.class, Chien.class, classes3, classes5, classes7, classes9, and .gitattributes. Below these is index\_1.php, which is currently selected and has a blue bar at the bottom indicating it's the active file. The main pane displays the following PHP code:

```
index_1.php
1  ?php
2
3
4
5  include("classes/Animal.class");
6  $chien = new Animal();
7
8  $chien->setNom("Jump");
9  echo $chien->getNom();
```

Figure 33 : Code Php

A la suite j'ai appris à transmettre des données de page en page avec l'**Url** (*Uniform Ressource Locator*).

Imaginons que votre site s'appelle monsite.com et que vous avez une page PHP intitulée bonjour.php.  
Pour accéder à cette page, vous devez aller à l'URL suivante :  
<http://www.monsite.com/bonjour.php>

Si on veut maintenant envoyer des paramètres à la page PHP il suffit d'ajouter à cette URL après un point d'interrogation les informations suivantes telles que :  
<http://www.monsite.com/bonjour.php?nom=Dupont&prenom=Jean>

Figure 34 : La transmission de données avec l'URL

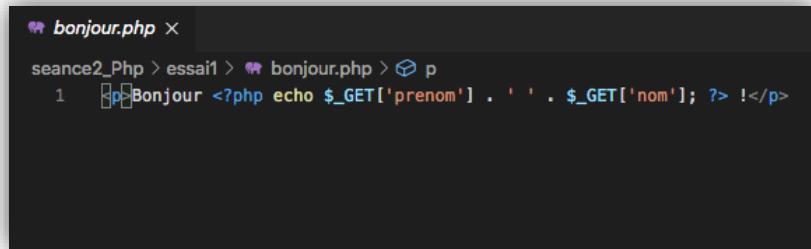
J'ai également appris à transmettre des données avec les formulaires et les méthodes **GET** : (*les données transiteront par l'URL*) et **POST** : (*les données ne transiteront pas par l'URL*) ainsi que Les variables **superglobales** et également les **Sessions** et les **Cookies**.

The screenshot shows a code editor interface with a sidebar labeled "EXPLORATEUR" containing a tree view of files. Under "seance2\_Php > essai1", there is an index.php file. The main pane displays the following PHP code:

```
index.php
1  ?php
2
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Document</title>
9  </head>
10 <body>
11 <a href="bonjour.php?nom=Dupont&prenom=Jean">Dis-moi bonjour !</a>
12 </body>
13 </html>
14
15
```

# DOSSIER PROFESSIONNEL (DP)

Figure 35 : HTML pour la méthode GET

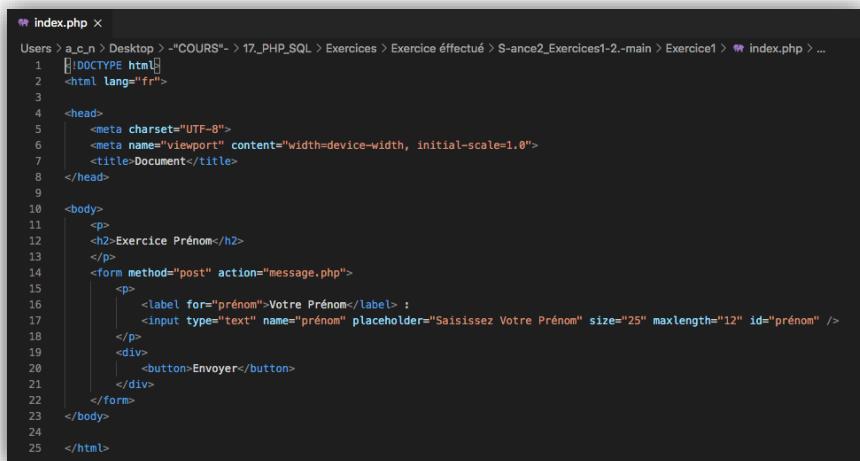


```
bonsjour.php x
seance2_Phps > essai1 > bonsjour.php > p
1   Bonjour <?php echo $_GET['prenom'] . ' ' . $_GET['nom']; ?> !</p>
```

Figure 36 : La méthode GET



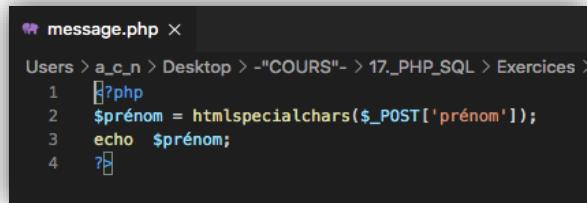
Figure 37 : L'écho du résultat de la méthode GET



```
index.php x
Users > a_c_n > Desktop > -"COURS"- > 17_PHP_SQL > Exercices > Exercice effectué > S-ance2_Exercices1-2.-main > Exercice1 > index.php > ...
1   !DOCTYPE html
2   <html lang="fr">
3
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8   </head>
9
10  <body>
11    <p>
12      <h2>Exercice Prénom</h2>
13    </p>
14    <form method="post" action="message.php">
15      <p>
16        <label for="prénom">Votre Prénom</label> :
17        <input type="text" name="prénom" placeholder="Saisissez Votre Prénom" size="25" maxlength="12" id="prénom" />
18      </p>
19      <div>
20        <button>Envoyer</button>
21      </div>
22    </form>
23  </body>
24
25  </html>
```

Figure 38 : HTML pour la méthode POST

# DOSSIER PROFESSIONNEL (DP)



```
?< message.php >
Users > a_c_n > Desktop > -"COURS"- > 17_PHP_SQL > Exercices >
1 <?php
2 $prenom = htmlspecialchars($_POST['prénom']);
3 echo $prenom;
4 ?>
```

Figure 39 : La méthode POST

## Exercice Prénom

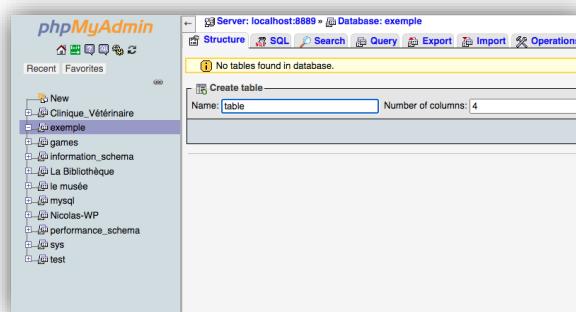
Votre Prénom :

Figure 40 : Résultat de la méthode POST

nicolas

Figure 41 : L'écho du résultat de la méthode POST

Et j'ai poursuivi mon apprentissage en apprenant à stocker des informations dans une base de données avec le langage **SQL** en créant des **Tables** et en modifiant leurs contenus.



# DOSSIER PROFESSIONNEL (DP)

Figure 42 : Création d'une table

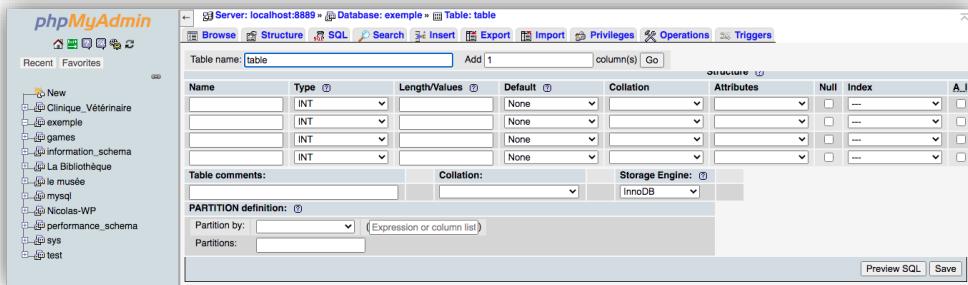


Figure 43 : La modification d'une table

J'ai aussi appris à lire des données en autorisant l'accès de PHP aux bases de données MySQL via le **PDO** : (*PHP Data Objects*). Ainsi pour les récupérer grâce aux **requêtes**, et les écrire avec la requête **INSERT INTO** qui permet d'ajouter une entrée.

```
?php
try
{
    $pdo_options [PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;

    $db = new PDO('mysql:host=localhost;dbname=games', 'root', 'root', $pdo_options);
```

Figure 44 : Connexion à MySQL via PDO

Votre première requête SQL  
Code : SQL  
SELECT \* FROM 'news'

```
$requete = $db->query('SELECT UPPER(nom) AS nom_maj FROM jeux_video');
```

Figure 45 : Les requêtes

# DOSSIER PROFESSIONNEL (DP)

## Ecrire des données

### INSERT : ajouter des données

La requête INSERT INTO permet d'ajouter une entrée

Code : SQL

```
INSERT INTO news(id, auteur, titre, contenu, dateAjout, dateModif) VALUES(",'Patrick', 'Ma première info', 'Bonjour le monde !', NOW(), NOW())
```

Figure 46 : La requête INSERT INTO

Et puis j'ai continué avec les **Fonctions Scalaires SQL**. Cela crée un "**champ virtuel**" qui n'existe que le temps de la requête, et j'ai poursuivi sur les différents types de dates que peut stocker MySQL et j'ai également travaillé sur les **jointures** entre les tables ce qui permet de pouvoir créer des **relations** entre elles et de les liées.

## Les fonctions SQL

### Les fonctions scalaires

Nous allons d'abord découvrir le mode d'emploi d'une fonction SQL de type scalaire : la fonction UPPER.

### Utiliser une fonction scalaire SQL

Code : SQL

```
SELECT UPPER(nom) FROM jeux_video
```

Figure 47 : Les fonctions scalaires SQL

## Les dates en SQL

### Les champs de type date

### Les différents types de dates

Voici les différents types de dates que peut stocker MySQL :

- DATE : stocke une date au format AAAA-MM-JJ (Année-Mois-Jour).
- TIME : stocke un moment de la journée au format HH:MM:SS (Heures:Minutes:Secondes).
- DATETIME : stocke une combinaison d'une date et d'un moment de la journée au format AAAA-MM-JJ HH:MM:SS. Ce type de champ est donc plus précis.
- TIMESTAMP : stocke une date et un moment de la journée sous le format AAAAMMJJHHMMSS.
- YEAR : stocke une année, soit au format AA, soit au format AAAA.

### Utilisation des champs de date en SQL

Si le champ est de type DATETIME, il faut indiquer précisément les heures, minutes et secondes :

Code : SQL

```
SELECT pseudo, message, date FROM minichat WHERE date = '2017-24-01' 14:18:45'
```

Pour récupérer la liste des messages postés entre le 24/01/2017 et le 27/01/2017 :

Code : SQL

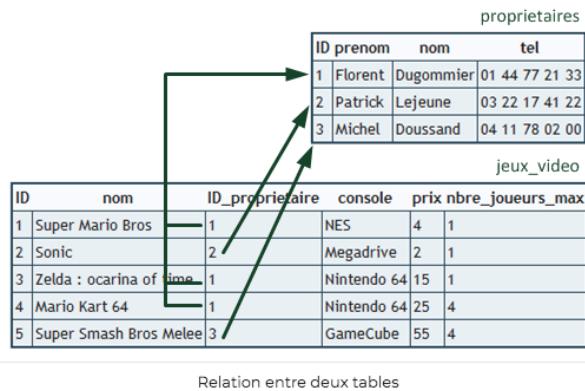
```
SELECT pseudo, message, date FROM minichat WHERE date BETWEEN '2017-24-01 00:00:00' AND '2017-27-01 00:00:00'
```

Figure 48 : Les différents types de dates que peut stocker MySQL

# DOSSIER PROFESSIONNEL (DP)

Le nouveau champ `IDProprietaire` est de type `INT`. Il permet de faire référence à une entrée précise de la table `Proprietaires`.

On peut maintenant considérer que les tables sont reliées à travers ces ID de propriétaires, comme le suggère la figure suivante.



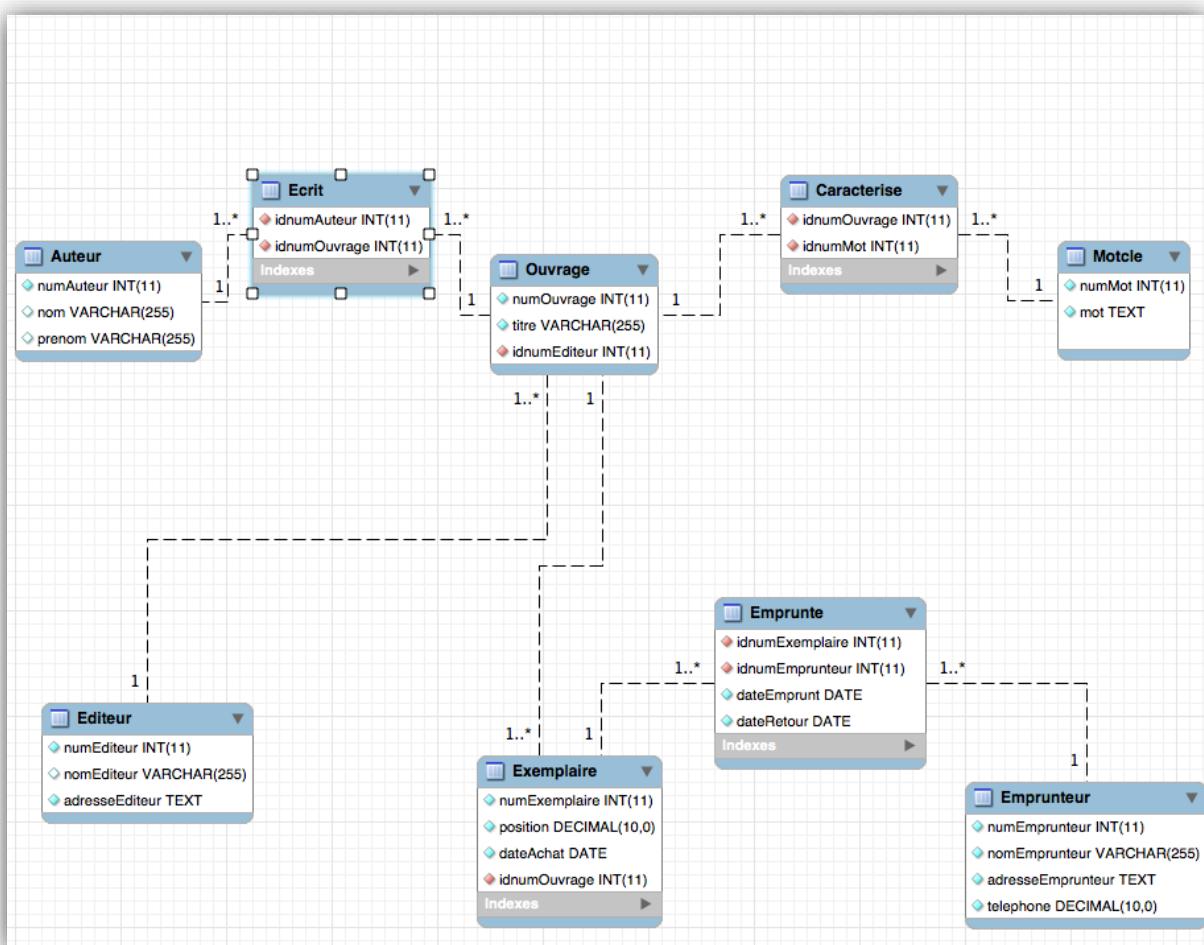
| Table        | Action   | Rows | Type   | Collation       | Size      | Overhead |
|--------------|--|------|--------|-----------------|-----------|----------|
| Animal       | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Chat         | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Chien        | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Dossier      | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Effectuer    | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Garde        | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Horaire      | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Medicament   | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Prescrire    | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Proprietaire | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Race_chat    | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Race_chien   | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Veterinaire  | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| Visite       | Browse  Structure  Search  Insert  Empty  Drop | 0    | InnoDB | utf8_general_ci | 16.0 Kib  | -        |
| 14 tables    | Sum  | 0    | InnoDB | utf8_general_ci | 224.0 Kib | 0 B      |

Figure 49 : Les jointures entre les tables

Figure 50 : La base de données

# DOSSIER PROFESSIONNEL (DP)

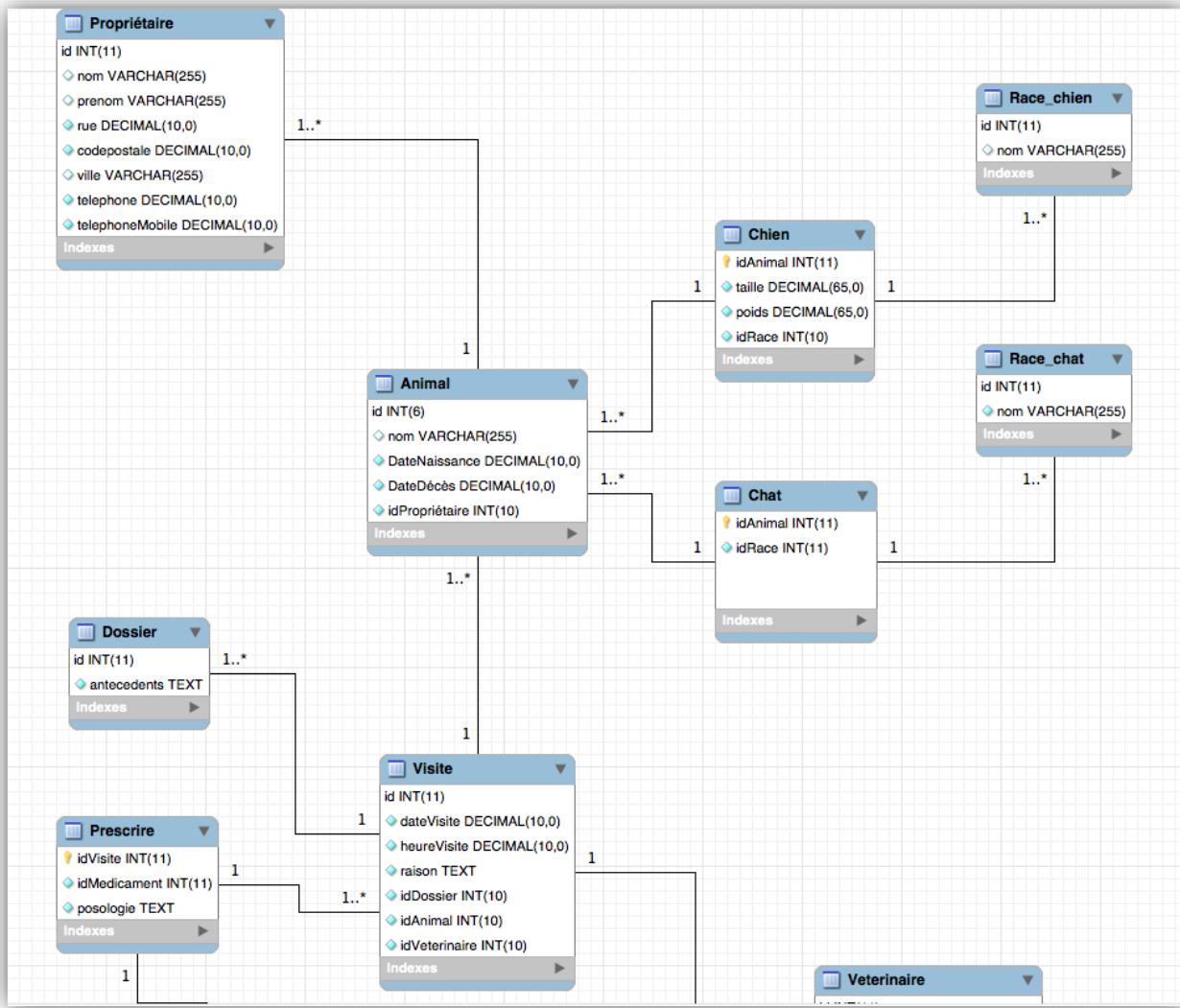
J'ai poursuivi avec le langage de modélisation UML (*Langage de Modélisation Unifié*). Il permet, grâce à un ensemble de diagrammes très explicites, de représenter l'architecture et le fonctionnement des systèmes informatiques complexes en tenant compte des relations entre les concepts utilisés et l'implémentation qui en découle.



Pour terminer cette compétence j'ai créé deux bases de données avec Php MySql, les thèmes des deux projets était : (*la Bibliothèque et le Musée*). Pour lesquelles je me suis occupé de la création des **diagrammes** avec le **MCD** (*Modèle Conceptuel de Données*) en y ajoutant les **contraintes d'intégrité** souhaitées dans les exercices tout cela créé via **MySQL WorkBench**. Le **MCD** permet d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

# DOSSIER PROFESSIONNEL (DP)

Figure 51 : Le modèle MCD



Pour l'exercices demandés dans cette compétence j'ai commencé par créer une base de données qui correspond à la gestion d'une clinique vétérinaire ensuite je me suis servi du **MLD** (*modèle logique de données*) qui est la représentation des données d'un système d'information. Les données sont représentées en prenant en compte le modèle technologique qui sera utilisée pour leur gestion.

Figure 52 : Le modèle MLD

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette cinquième compétence :

Logiciel: Visual studio code, Mamp, Mysql, Mysql-WorkBench.

Site web: Github desktop.

Application: PhpMyAdmin.

Langage: Html5, Css3, Sql, Uml.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

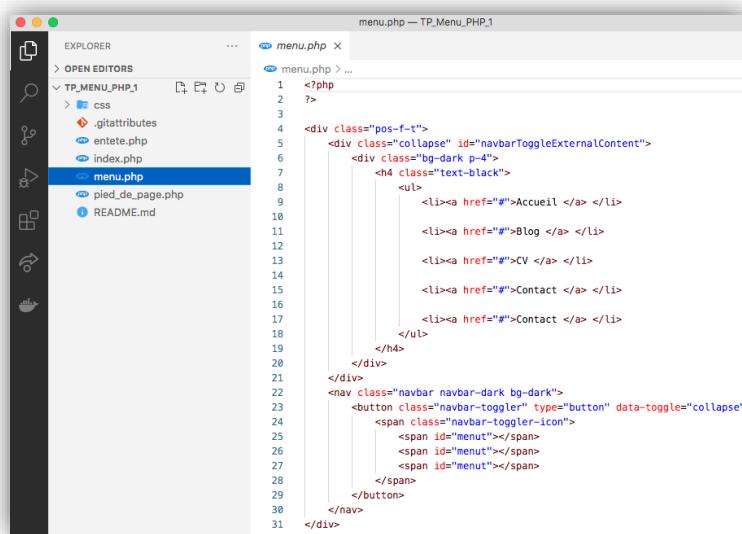
## Activité-type 2

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

Compétence n°6 ▶ Développer les composants d'accès aux données

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

Pour cette sixième compétence J'ai commencé par réaliser la page d'accueil d'un blog en PHP.



```
menu.php — TP_Menu_PHP_1
menu.php ...
1 <?php
2 >
3
4 <div class="pos-f-t">
5   <div class="collapse" id="navbarToggleExternalContent">
6     <div class="bg-dark p-4">
7       <h4 class="text-black">
8         <ul>
9           <li><a href="#">Accueil </a> </li>
10          <li><a href="#">Blog </a> </li>
11          <li><a href="#">CV </a> </li>
12          <li><a href="#">Contact </a> </li>
13        </ul>
14      </h4>
15    </div>
16  </div>
17  <nav class="navbar navbar-dark bg-dark">
18    <button class="navbar-toggler" type="button" data-toggle="collapse">
19      <span class="navbar-toggler-icon">
20        <span id="menut"></span>
21        <span id="menut"></span>
22        <span id="menut"></span>
23      </span>
24    </button>
25  </nav>
26
27
28
29
30
31 </div>
```



Figure 53 : Le code Php et La page d'accueil de mon blog en Php

# DOSSIER PROFESSIONNEL (DP)

J'ai continué sur la réalisation d'un blog avec Le design pattern **MVC** (*Modèle Views Controller*) administré **POO** (*Programmation Orientée Objet*).

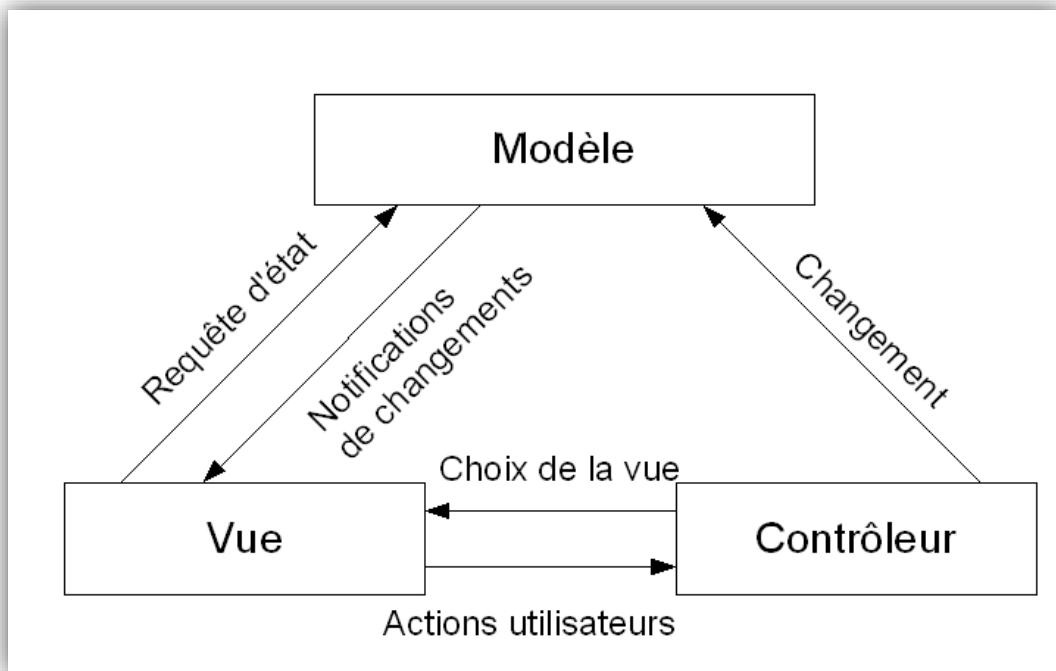


Figure 54 : Le modèle MVC

# DOSSIER PROFESSIONNEL (DP)

Dans le Modèle **MVC** on peut y intégrer le **CRUD** : (*create, read, update, delete*) qui nous permet de pouvoir : Créer – Lire – Mettre à jour et supprimer. Ce sont les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

Figure 55 : Le **CRUD**

Voilà ce qui se passe dans la base de données au moment de l'ajout, la modification, la mise à jour ou la suppression de données grâce au **CRUD** intégré.

|  | + Options | ↓ T → | ↓      | id | title                        | body  | image         | createdDate         | modifiedDate        |
|--|-----------|-------|--------|----|------------------------------|---|---------------|---------------------|---------------------|
|  | Edit      | Copy  | Delete | 11 | Phytoplancton bioluminescent | <p><span style="color: #ffff00;">La&ampnbspbiolumine... | phyto.jpg     | 2018-02-03 16:08:55 | NULL                |
|  | Edit      | Copy  | Delete | 24 | Les Fonds Marins             | <p><span style="color: #ffff00;">Les&ampnbsp<strong>... | fondmarin.jpg | 2018-02-03 18:33:05 | 2021-02-17 10:38:34 |
|  | Edit      | Copy  | Delete | 40 | La Tortue Marine             | <p><span style="color: #ffff00;">Les&ampnbsptortues ... | tortue.jpg    | 2018-02-07 13:41:04 | 2021-03-08 09:18:18 |

Figure 56 : La base de données au moment de sa modification

Je termine cette compétence en présentant mon Blog **MVC** personnalisé.

Voici le Dashboard (*Tableau de bord*), C'est un outil d'évaluation constitué de plusieurs indicateurs de performance à des moments donnés ou sur des périodes données.

| Titre                        | Date                  | Action                           |
|------------------------------|-----------------------|----------------------------------|
| La Tortue Marine             | Le 07/02/2018 à 13:41 | <b>MODIFIER</b> <b>SUPPRIMER</b> |
| Les Fonds Marins             | Le 03/02/2018 à 18:33 | <b>MODIFIER</b> <b>SUPPRIMER</b> |
| Phytoplancton bioluminescent | Le 03/02/2018 à 16:08 | <b>MODIFIER</b> <b>SUPPRIMER</b> |

# DOSSIER PROFESSIONNEL (DP)

Figure 57 : Le dashboard(tableau de bord)

The screenshot shows the 'Tableau de bord' (Dashboard) page for 'Les Profondeurs'. At the top, there are four colored boxes with statistics: Publications (3), Commentaires (2), Utilisateurs (1), and Signalements (0). Below this, a section titled 'Commentaires non lus' (Unread comments) displays a table with columns for 'NON SIGNALÉS' (Article, Aperçu des commentaires, Actions) and 'SIGNALÉS' (Actions). The table content indicates 'Aucun commentaire à valider' (No comments to validate).

| NON SIGNALÉS                | SIGNALÉS                |         |
|-----------------------------|-------------------------|---------|
| Article                     | Aperçu des commentaires | Actions |
| Aucun commentaire à valider |                         |         |

# DOSSIER PROFESSIONNEL (DP)

The screenshot shows a digital notebook interface with a blue header bar. The title 'Aller Simple Au Fond Des Océans' is displayed prominently. Below the title, there are two entries:

- La Tortue Marine** (Le 07/02/2018 à 13:41): An image of a sea turtle swimming over coral reefs.
- Les Fonds Marins** (Le 03/02/2018 à 18:33): An image of a vibrant coral reef with various tropical fish.

At the bottom of the screen, there are copyright and legal notices: 'Copyright © 2021 Mrc.Nicolas' and 'Mentions légales'.

The screenshot shows a blog post titled 'La Tortue Marine'. The post includes a small image of a baby sea turtle on a beach and a large, detailed illustration of a sea turtle's head and mouth.

**Text from the post:**

Les tortues marines sont les tortues de la super-famille des Cheloniidae. Elles sont présentes dans tous les océans du monde à l'exception de l'océan Arctique.

La tortue luth est la plus grande, mesurant 2 m de long, de 1 m à 1,5 m de large et pesant 600 kg. La taille moyenne des autres espèces va de 50 cm à 1 m et est proportionnellement moins large.

Ces espèces sont toutes vulnérables ou menacées. Elles font localement l'objet de protection ou de plan de restauration, mais la pollution, le braconnage et les prises accidentelles par engins de pêche restent des causes préoccupantes de recul de populations, déjà très rélictuelles.

Les tortues marines, comme les autres tortues (*Testudines*) sont des reptiles ectothermes disposant d'un système respiratoire pulmonaire, qui correspondait à une adaptation complète à la vie terrestre. En supplément des reins, des glandes lacrymales près de leurs yeux sont capables d'expulser les excédents de sel de leur circulation sanguine. Ces glandes lacrymales transformées en « glandes à sel » expliquent notamment les larmes des femelles lorsqu'elles viennent pondre. Leur métabolisme est capable de supporter des fortes concentrations de CO<sub>2</sub> lors des plongées profondes. Elles peuvent extraire 35 à 52 % de l'oxygène de leur respiration. Leur vision est très développée. Leur œil est protégé par trois paupières, leur ouïe est particulièrement fine même si elles n'ont pas d'oreilles externes, mais une oreille interne, sous une plaque auditive. Leur odorat est en revanche assez peu performant. Les tortues marines n'ont qu'un seul orifice excréteur, qui sert également pour la reproduction.

Le sexe des tortues n'est pas apparent, aussi c'est d'après leurs caractères sexuels secondaires que l'on peut le déterminer. On définit par puberté la période où ceux-ci apparaissent. Le plus

Figure 58 : Mon blog MVC

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette sixième compétence :

Logiciel: Visual studio code, Mamp, Mysql.

Site web: Github desktop.

Application: PhpMyAdmin.

Langage: Html5, Css3, Php.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 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.

Compétence n°7 ▶ Développer la partie back-end d'une application web ou web mobile

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

Pour cette septième compétence j'ai commencé par lancer l'éditeur de code **Visual Studio Code**, ouvrir le terminal et y installer **Symfony** : (*Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web*). En exécutant cette commande illustrée ci-dessous.

```
$ curl -sS https://get.symfony.com/cli/installer | bash
...
1
2
3
```

Figure 59 : Installation de Symfony

A la suite j'ai également installé **Composer** : (*Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin*). En exécutant les commandes illustrées ci-dessous.

Pour installer rapidement Composer dans le répertoire courant, exécutez le script suivant dans votre terminal. Pour automatiser l'installation, utilisez le guide d'installation de Composer par programmation .

```
php -r "copie ('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file ('sha384', 'composer-setup.php') === '756890a4488ce9024fc62c56153228907f1545c22f'
php composer-setup.php
php -r "unlink ('composer-setup.php');"
```

# DOSSIER PROFESSIONNEL (DP)

Figure 60 : Installation de composer

Après avoir vérifié les versions de **Symfony**, **Composer** et **Php**, J'ai poursuivi sur la création du projet **Symfony** en tapant dans le terminal la commande : (*composer create-project symfony/website-skeleton*).

Ce qui nous permet d'avoir cette structure illustrée ci-dessous.

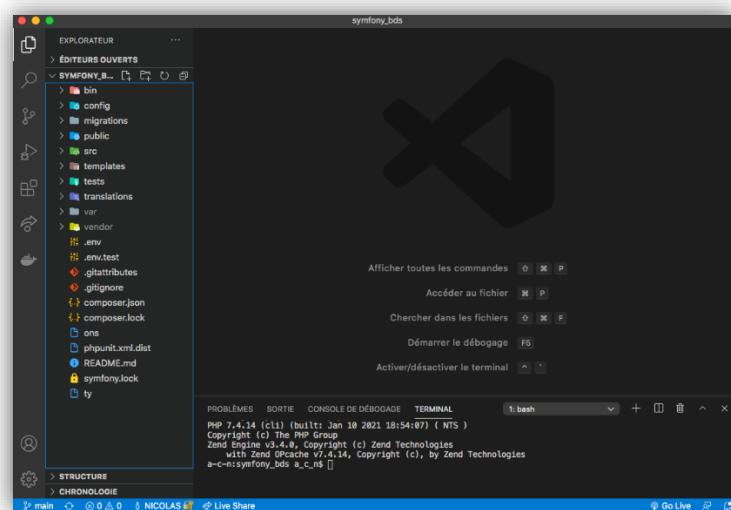


Figure 61 : La structure de **Symfony** obtenue dans *Visual studio code*

Ensuite je lance mon projet **Symfony** en tapant en ligne de commande : **symfony server -d**.

```
A-C-N:~ a_c_n$ symfony server -d
[WARNING] run "symfony server:ca:install" first if you want to run the web server with TLS support, or use "--no-tls" to avoid this warning

[OK] Web server listening
http://127.0.0.1:8000

Stream the logs via symfony server:log
A-C-N:~ a_c_n$
```

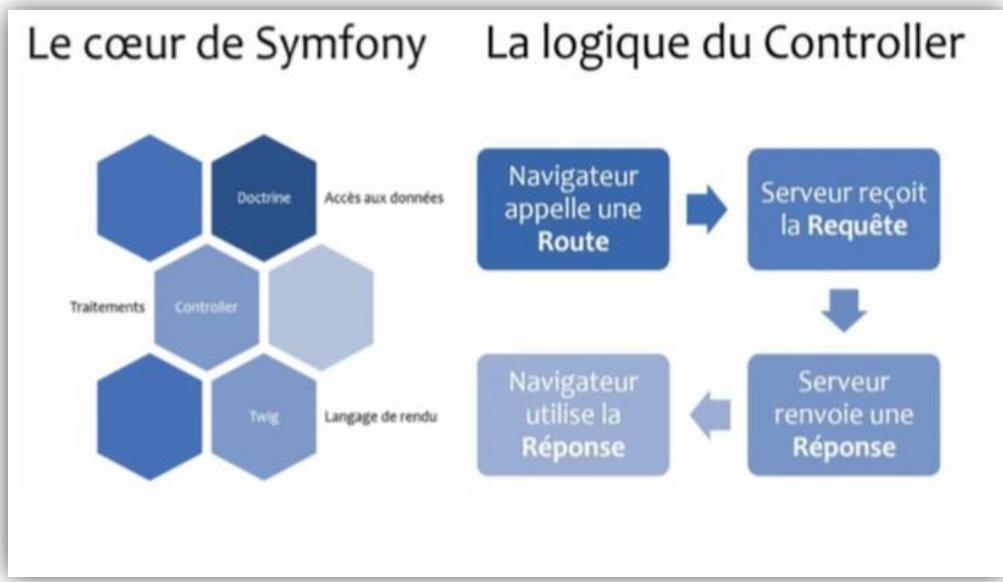
# DOSSIER PROFESSIONNEL (DP)



Figure 62 : Le démarrage du projet Symfony

Je continue sur la création d'une application web en créant une première page, pour ça, il faut créer un **controller** avec ces **routes** en installant le **maker-bundle** en tapant en ligne de commande : (*composer require symfony/maker-bundle --dev*) et (*composer require doctrine/annotations*). Ensuite je tape en ligne de commande : (*php bin/console make:controller*) pour la création du controller et je relance le serveur.

Le Controller permet de créer des fonctionnalités dans symfony voici un exemple de son fonctionnement ci-dessous ainsi que la définition de ces routes.



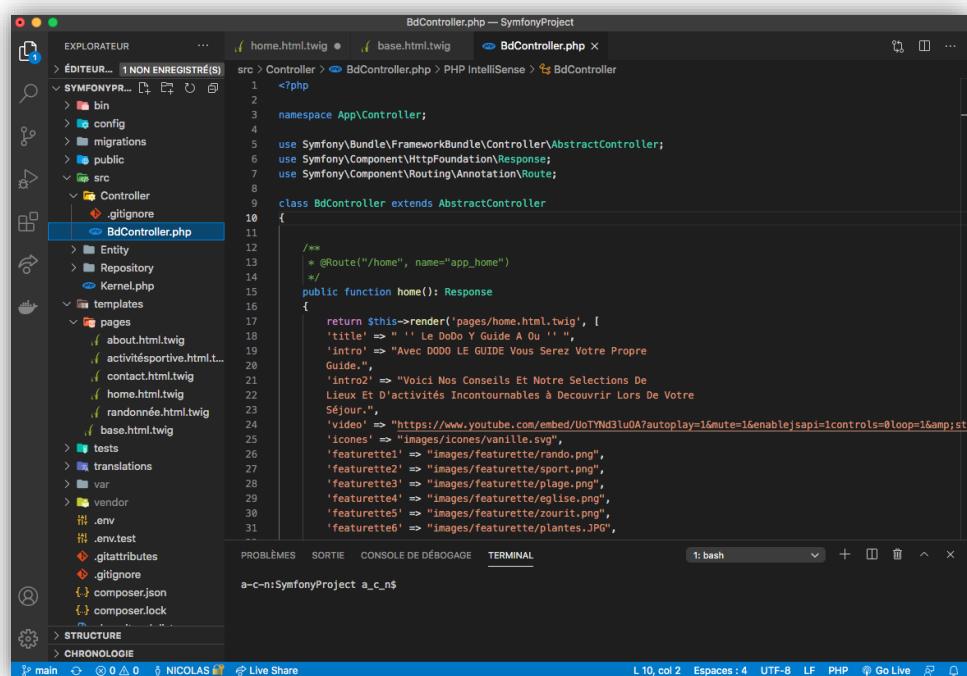
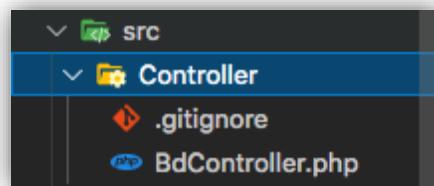
# DOSSIER PROFESSIONNEL (DP)

## Les routes

Une route est une correspondance entre une URL et une fonction. Celle-ci se trouve dans le controller. Une route peut être configurée par une annotation (mais aussi dans un fichier yaml).

Figure 63 : Le controller et ces routes

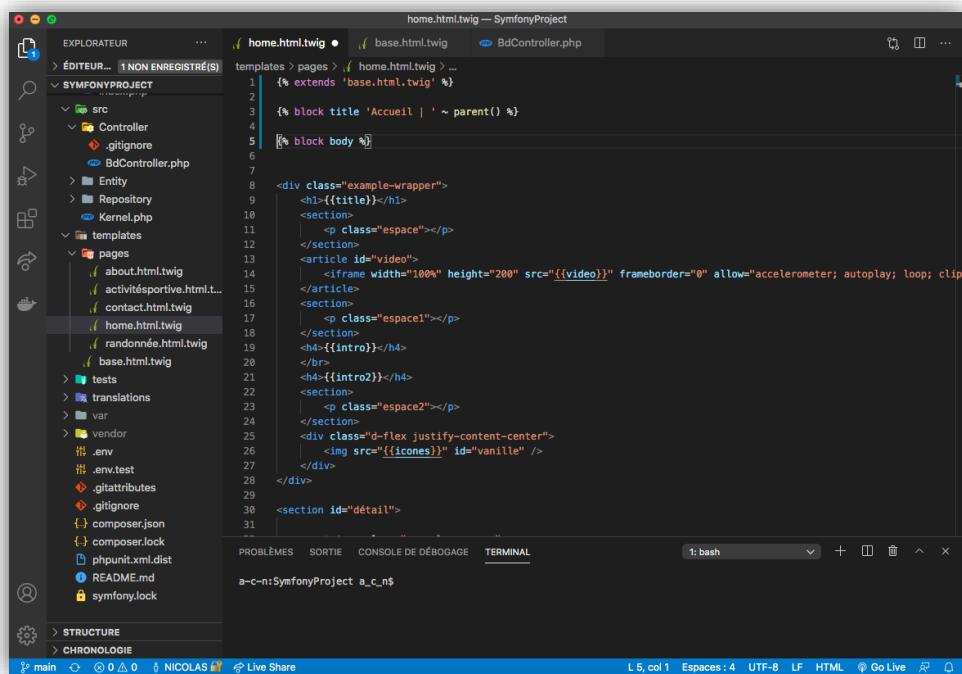
Voici mon controller créé.



```
BdController.php — SymfonyProject
src > Controller > BdController.php > PHP Intellisense > BdController
1  <?php
2
3  namespace App\Controller;
4
5  use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
6  use Symfony\Component\HttpFoundation\Response;
7  use Symfony\Component\Routing\Annotation\Route;
8
9  class BdController extends AbstractController
10 {
11
12     /**
13      * @Route("/home", name="app_home")
14      */
15     public function home(): Response
16     {
17         return $this->render('pages/home.html.twig', [
18             'title' => " La Dodo Y Guide A Ou ''",
19             'intro' => "Avec DODO LE GUIDE Vous Serez Votre Propre
20             Guide.",
21             'intro2' => "Voici Nos Conseils Et Notre Selections De
22             Lieux Et D'activités Incontournables à Découvrir Lors De Votre
23             Séjour.",
24             'video' => "https://www.youtube.com/embed/UoYnd3lu0A?autoplay=1&mute=1&enablejsapi=1&loop=1&amp;start=10",
25             'icons' => "images/icones/vanille.svg",
26             'featurette1' => "images/featurette/rando.png",
27             'featurette2' => "images/featurette/sport.png",
28             'featurette3' => "images/featurette/plage.png",
29             'featurette4' => "images/featurette/eglise.png",
30             'featurette5' => "images/featurette/zourit.png",
31             'featurette6' => "images/featurette/plantes.JPG",
32         ]);
33     }
34 }
```

Figure 64 : Le controller créé

# DOSSIER PROFESSIONNEL (DP)



The screenshot shows the Visual Studio Code interface with the file `home.html.twig` open in the editor. The code is a Twig template for a homepage. It extends the `base.html.twig` layout and defines a `body` block. The body contains sections for the title, a video player, and two sections for introduction and details, each with its own heading and paragraph. The code uses Bootstrap classes like `d-flex justify-content-center` and `img` tags for images. The sidebar shows the project structure with files like `BdController.php`, `Entity`, `Repository`, `Kernel.php`, and various `.html.twig` templates. The bottom status bar shows the terminal is in bash mode.

```
% extends 'base.html.twig'

{block title 'Accueil' | ~ parent()}

{block body}

<div class="example-wrapper">
    <h1>{{title}}</h1>
    <section>
        <p class="espace1"></p>
    </section>
    <article id="video">
        <iframe width="100%" height="200" src="{{video}}" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" clip-path="inset(0 0 100% 100%)"></iframe>
    </article>
    <section>
        <p class="espace1"></p>
    </section>
    <h4>{{intro}}</h4>
    <br>
    <h4>{{intro2}}</h4>
    <section>
        <p class="espace2"></p>
    </section>
    <div class="d-flex justify-content-center">
        
    </div>
</div>

<section id="détail">
```

Figure 65 : Code de la première page de mon application web sur VisualStudioCode avec Symfony

# DOSSIER PROFESSIONNEL (DP)

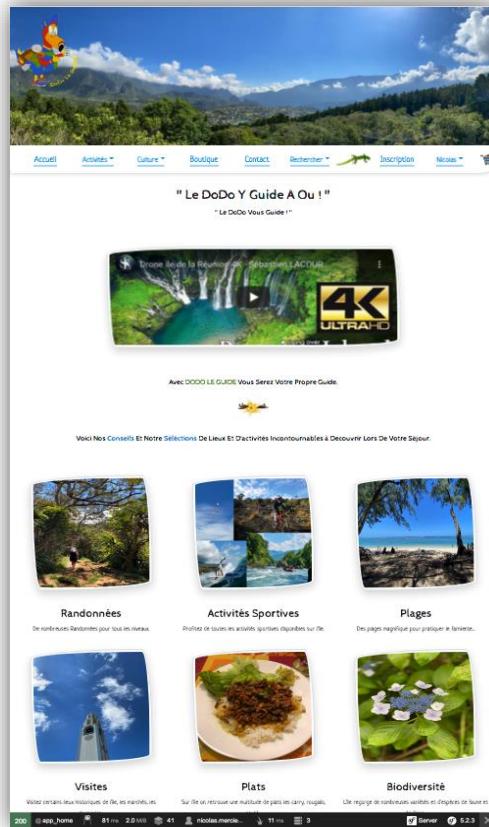


Figure 66 : Résultat sur le navigateur du code ci-dessus

Symfony propose un langage de **templates** puissant, appelé **Twig**.

A screenshot of the Twig template engine documentation page. The header includes the SymfonyHub logo and the text "SymfonyCloud: The best way to host your Symfony project". The main title is "Twig" with the subtitle "The flexible, fast, and secure template engine for PHP". Below the title is a green decorative bar with a plant icon. The text "Twig is a modern template engine for PHP" is followed by a bulleted list: "Fast: Twig compiles templates down to plain optimized PHP code. The overhead compared to regular PHP code was reduced to the very minimum.", "Secure: Twig has a sandbox mode to evaluate untrusted template code. This allows Twig to be used as a template language for applications where users may modify the template design.", and "Flexible: Twig is powered by a flexible lexer and parser. This allows the developer to define its own custom tags and filters, and create its own DSL.". A section titled "Why yet another template engine?" discusses the evolution of PHP as a template engine and highlights the conciseness of Twig's syntax. A code snippet shows how a complex PHP block is simplified in Twig:

```
<?php echo $var ?>
<?php echo htmlspecialchars($var, ENT_QUOTES, 'UTF-8') ?>
```

# DOSSIER PROFESSIONNEL (DP)



Figure 67 : Page d'accueil Twig et sa documentation

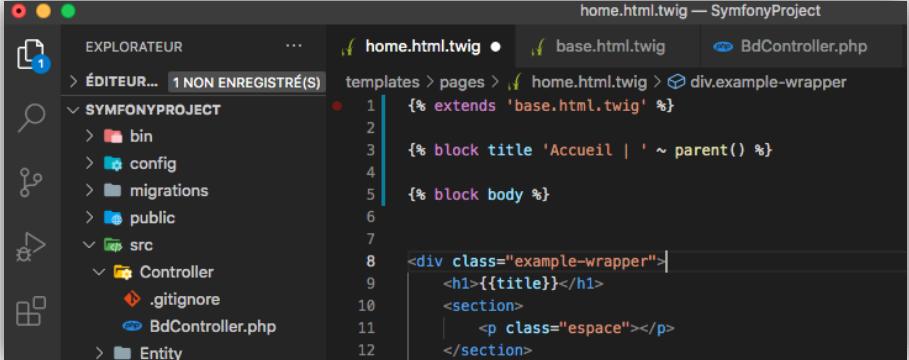
The screenshot shows a code editor window for a Twig template named "home.html.twig". The code is as follows:

```
home.html.twig • base.html.twig BdController.php
templates > pages > home.html.twig > div.example-wrapper
1  {% extends 'base.html.twig' %}
2
3  {% block title 'Accueil | ' ~ parent() %}
4
5  {% block body %}
6
7
8  <div class="example-wrapper">
9    <h1>{{title}}</h1>
10   <section>
11     <p class="espace"></p>
12   </section>
13   <article id="video">
14     <iframe width="100%" height="200" src="{{video}}" frameborder="0" allow="accelerometer; autoplay; loop; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
15   </article>
16   <section>
17     <p class="espace1"></p>
18   </section>
19   <h4>{{intro}}</h4>
20   <br>
21   <h4>{{intro2}}</h4>
22   <section>
23     <p class="espace2"></p>
24   </section>
25   <div class="d-flex justify-content-center">
26     
27   </div>
28 </div>
29
```

Figure 68 : L'écriture Twig dans VisualStudioCode

# DOSSIER PROFESSIONNEL (DP)

Les principales fonctionnalité sont qu'elles peuvent afficher le contenu d'une **variable** exemple avec : L'interpolation `{{ var }}` on utilise les **doubles-accolades** pour afficher le contenu d'une variable.



The screenshot shows the Symfony Project Explorer with the 'src' directory expanded, containing 'Controller' and 'Entity'. Inside 'Controller', there is a file named 'BdController.php'. In the main editor area, the file 'home.html.twig' is open, showing the following code:

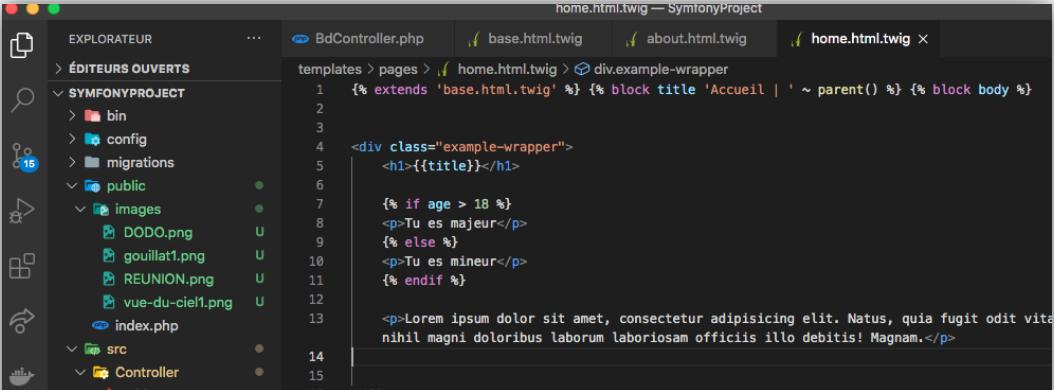
```
home.html.twig — SymfonyProject
templates > pages > home.html.twig > div.example-wrapper
1  {% extends 'base.html.twig' %}
2
3  {% block title 'Accueil | ' ~ parent() %}
4
5  {% block body %}
6
7
8  <div class="example-wrapper">
9    <h1>{{title}}</h1>
10   <section>
11     <p class="espace"></p>
12   </section>
```

Figure 69 : Fonctionnalité : afficher le contenu d'une variable



Figure 70 : Résultat de la fonctionnalité de l'exemple ci-dessus

On peut aussi appeler une structure de contrôle avec la commande : `{% cmd %}` et pour se faire on utilise l'**accolade pourcentage** pour appeler.

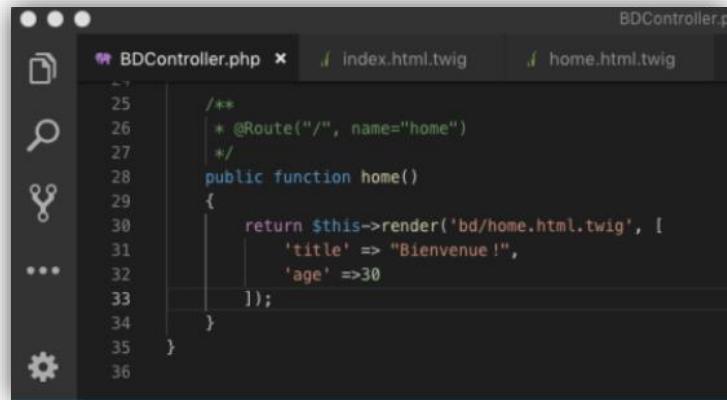


The screenshot shows the Symfony Project Explorer with the 'public' directory expanded, containing 'images' and 'index.php'. Inside 'images', there are several files including 'DODO.png', 'gouillat1.png', 'REUNION.png', and 'vue-du-ciel1.png'. In the main editor area, the file 'home.html.twig' is open, showing the following code:

```
home.html.twig — SymfonyProject
templates > pages > home.html.twig > div.example-wrapper
1  {% extends 'base.html.twig' %} {% block title 'Accueil | ' ~ parent() %} {% block body %}
2
3
4  <div class="example-wrapper">
5    <h1>{{title}}</h1>
6
7    {% if age > 18 %}
8      <p>Tu es majeur</p>
9    {% else %}
10      <p>Tu es mineur</p>
11    {% endif %}
12
13    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Natus, quia fugit odit vita
14    nihil magni doloribus laborum laboriosam officiis illo debitis! Magnam.</p>
15
16  </div>
```

Figure 71 : Fonctionnalité appelle d'une structure de contrôle avec les accolades pourcentage

# DOSSIER PROFESSIONNEL (DP)



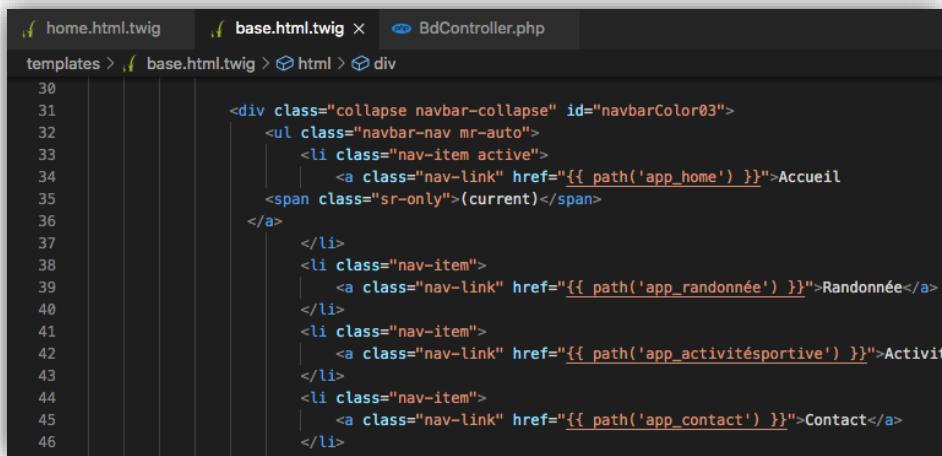
```
BDController.php x index.html.twig home.html.twig
25 /**
26  * @Route("/", name="home")
27 */
28 public function home()
29 {
30     return $this->render('bd/home.html.twig', [
31         'title' => "Bienvenue !",
32         'age' => 30
33     ]);
34 }
35
36
```

Figure 72 : Le controller qui contient la route pour traiter le code de la page .Twig et le renvoyer sur la page home du navigateur



Figure 73 : Résultat de la fonctionnalité de l'exemple ci-dessus

Il existe aussi la fonction **Path**, twig prend 2 paramètres : 1 obligatoire le nom de la route et 1 optionnel. J'utilise cette fonction pour créer un lien de cette manière : {{ path() }}.



```
home.html.twig x base.html.twig x BdController.php
templates > base.html.twig > html > div
30
31     <div class="collapse navbar-collapse" id="navbarColor03">
32         <ul class="navbar-nav mr-auto">
33             <li class="nav-item active">
34                 <a class="nav-link" href="{{ path('app_home') }}>Accueil
35                 <span class="sr-only">(current)</span>
36             </a>
37             </li>
38             <li class="nav-item">
39                 <a class="nav-link" href="{{ path('app_randonnée') }}>Randonnée</a>
40             </li>
41             <li class="nav-item">
42                 <a class="nav-link" href="{{ path('app_activitésportive') }}>Activité
43             </li>
44             <li class="nav-item">
45                 <a class="nav-link" href="{{ path('app_contact') }}>Contact</a>
46             </li>
```

# DOSSIER PROFESSIONNEL (DP)

Figure 74 : Fonctionnalité Path



Figure 75 : Résultat sur le navigateur de la fonctionnalité Path

**Symfony** utilise également **Doctrine**, c'est un **ORM**: (*Object Relational Mapping*).

L'ORM est une classe qui permet de manipuler les tables d'une base de données comme des objets (*Entities*) et leurs relations.

**Doctrine** ajoute des fonctionnalités à  **PDO**. (*PHP Data Objects*).

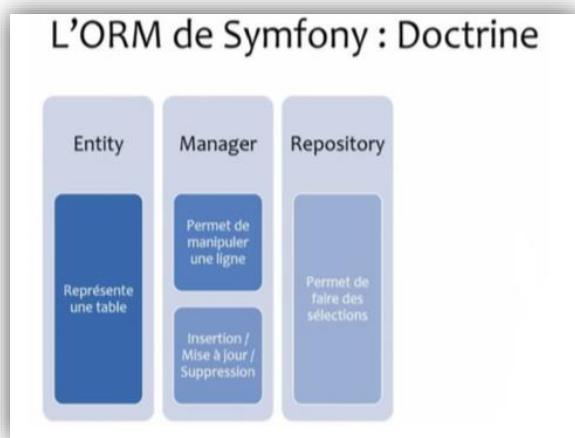


Figure 76 : L'ORM de Symfony

Je poursuis avec Les **migrations** dans **Symfony** qui permettent de faire passer une base de données d'un état A à un état B.

# DOSSIER PROFESSIONNEL (DP)

## Les migrations dans Symfony

### Migration #1

- Je créé 2 tables

### Migration #2

- Je modifie les champs d'une table
- J'en créé une autre
- J'en supprime une autre

### Migration #3

- Je supprime un champ d'une table
- J'ajoute une relation entre deux tables

Figure 77 : Les migrations

On peut parler aussi des **fixtures** ou données de tests, qui permettent au développeur d'initialiser les données de la base de données dans un état connu.

## Les Fixtures

Créer des jeux de fausses données

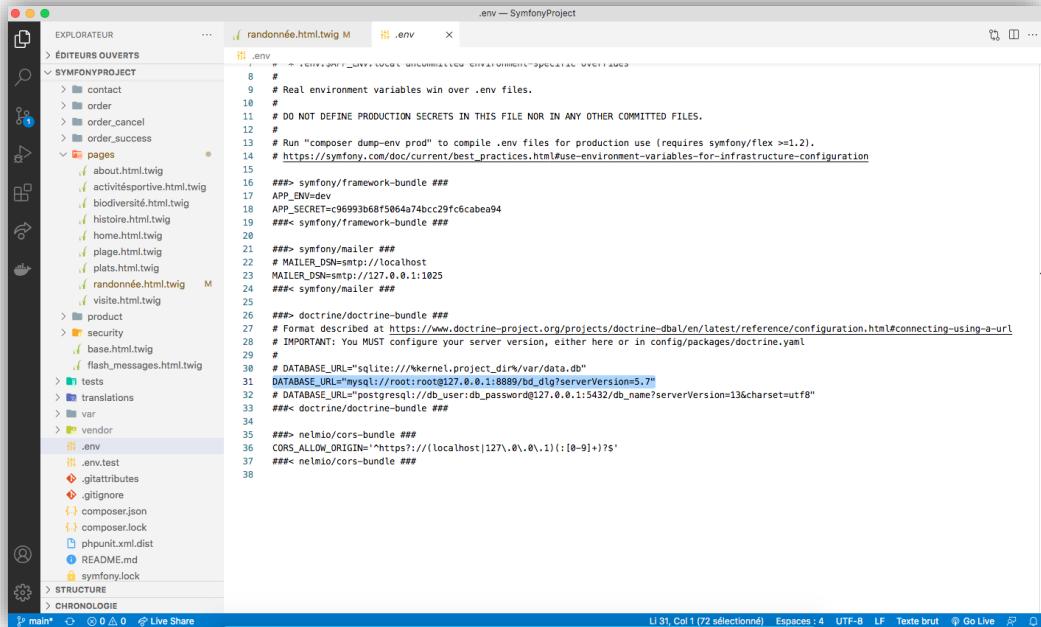
Exécutable à souhait

Réutilisable par les autres

Figure 78 : Les fixtures

Pour relier ma base de donnée à mon projet **Symfony** avec l'**ORM Doctrine**, j'ouvre le **fichier .env** comme dans l'exemple ci-dessous et je modifie la ligne 31 en y ajoutant le nom de ma base de donnée et en vérifiant le port du serveur MySql qui est le **8889**.

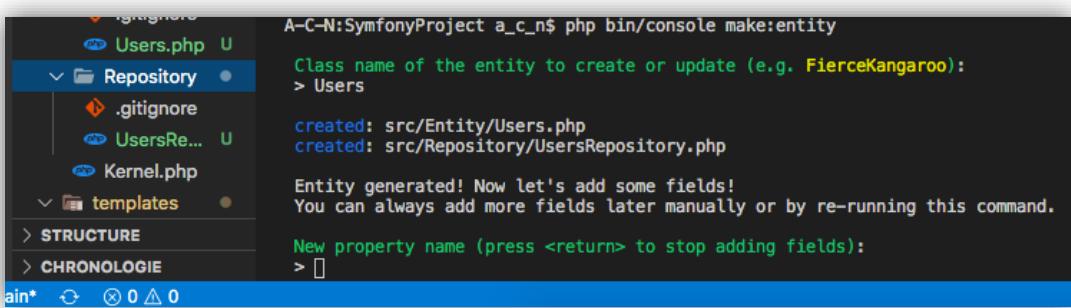
# DOSSIER PROFESSIONNEL (DP)



```
.env — SymfonyProject
.
.
.
# Real environment variables win over .env files.
#
# DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
#
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
# https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-configuration
#
# symfony/framework-bundle ####
APP_ENV=dev
APP_SECRET=96993b68f5064a74bcc29fc6cabea94
# symfony/framework-bundle ####
#
#> symfony/mailer ####
MAILER_DSN=smtp://localhost
MAILER_DSN=smtplib://127.0.0.1:1025
# symfony/mailer ####
#
#> doctrine/doctrine-bundle ####
# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
#
# DATABASE_URL="sqlite:///{kernel.project_dir}/var/data.db"
# DATABASE_URL="mysql://root:root@127.0.0.1:8889/_db_dig7?serverVersion=5.7"
# DATABASE_URL="pgsql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13&charset=utf8"
#> doctrine/doctrine-bundle ####
#
#> nelmio/cors-bundle ####
CORS_ALLOW_ORIGIN='https://(localhost|127\.\d{1,2}\.\d{1,2}\.\d{1,2})(:[0-9]+)?'
#< nelmio/cors-bundle ####
```

Figure 79 : Modification du fichier .env

Je peux désormais créer des **entités** avec la commande :(*php bin/console make :entity*.)



```
A-C-N:SymfonyProject a_c_n$ php bin/console make:entity
Class name of the entity to create or update (e.g. FierceKangaroo):
> Users
created: src/Entity/Users.php
created: src/Repository/UsersRepository.php
Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.
New property name (press <return> to stop adding fields):
> 
```

Figure 80 : Crédit d'une class (Users) qui représente les utilisateurs du site

La ligne de commande a créé deux nouveaux fichiers une **entité Users** (qui représente la table des Utilisateurs inscrit sur le site) et un **repository UsersRepository** (qui permet de faire des sélections sur les données de la table des utilisateurs).

On continue avec un exemple de relation entre deux entités (Produits et Auteur) avec la commande : (*php bin/console make :entity Produit*.)

# DOSSIER PROFESSIONNEL (DP)

The figure consists of three vertically stacked screenshots from a terminal window, showing the process of creating relationships between entities in a Symfony project.

**Screenshot 1:** The user runs the command `php bin/console make:entity Produit`. The response indicates that the entity already exists and prompts for adding new fields. The user adds a field named "auteur" of type "relation".

```
A-C-N:SymfonyProject a_c_n$ php bin/console make:entity Produit
Your entity already exists! So let's add some new fields!
New property name (press <return> to stop adding fields):
> auteur
Field type (enter ? to see all types) [string]:
> relation
```

**Screenshot 2:** The user is prompted to choose a related entity class. They select "Auteur". A table lists relationship types: ManyToOne, OneToMany, ManyToMany, and OneToOne, each with a detailed description.

| Type       | Description   |
|------------|---|
| ManyToOne  | Each <code>Produit</code> relates to (has) <code>one Auteur</code> .<br>Each <code>Auteur</code> can relate to (can have) <code>many Produit</code> objects                           |
| OneToMany  | Each <code>Produit</code> can relate to (can have) <code>many Auteur</code> objects.<br>Each <code>Auteur</code> relates to (has) <code>one Produit</code>                            |
| ManyToMany | Each <code>Produit</code> can relate to (can have) <code>many Auteur</code> objects.<br>Each <code>Auteur</code> can also relate to (can also have) <code>many Produit</code> objects |
| OneToOne   | Each <code>Produit</code> relates to (has) <code>exactly one Auteur</code> .<br>Each <code>Auteur</code> also relates to (has) <code>exactly one Produit</code> .                     |

**Screenshot 3:** The user confirms the relationship type as `ManyToOne`. They are asked if the `Produit.auteur` property can be null. They answer `no`. The user is then prompted to add a new property `products` to the `Auteur` entity. They are asked if they want to activate `orphanRemoval`. They answer `yes`. The user is then asked if they want to automatically delete orphaned `Produit` objects. They answer `no`. Finally, they are asked if they want to add another property, which they respond by pressing `return`. The process concludes with a "Success!" message.

```
What class should this entity be related to?:
> Auteur
What type of relationship is this?
-----

| Type       | Description                                                                                                                                                                           |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ManyToOne  | Each <code>Produit</code> relates to (has) <code>one Auteur</code> .<br>Each <code>Auteur</code> can relate to (can have) <code>many Produit</code> objects                           |
| OneToMany  | Each <code>Produit</code> can relate to (can have) <code>many Auteur</code> objects.<br>Each <code>Auteur</code> relates to (has) <code>one Produit</code>                            |
| ManyToMany | Each <code>Produit</code> can relate to (can have) <code>many Auteur</code> objects.<br>Each <code>Auteur</code> can also relate to (can also have) <code>many Produit</code> objects |
| OneToOne   | Each <code>Produit</code> relates to (has) <code>exactly one Auteur</code> .<br>Each <code>Auteur</code> also relates to (has) <code>exactly one Produit</code> .                     |

Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:
> ManyToOne
Is the Produit.auteur property allowed to be null (nullable)? (yes/no) [yes]:
> no
Do you want to add a new property to Auteur so that you can access/update Produit objects from it - e.g. $auteur->getProduits():
>
A new property will also be added to the Auteur class so that you can access the related Produit objects from it.
New field name inside Auteur [products]:
>
Do you want to activate orphanRemoval on your relationship?
A Produit is "orphaned" when it is removed from its related Auteur.
e.g. $auteur->removeProduit($produit)
NOTE: If a Produit may *change from one Auteur to another, answer "no".
Do you want to automatically delete orphaned App\Entity\Produit objects (orphanRemoval)? (yes/no) [no]:
> yes
updated: src/Entity/Produit.php
updated: src/Entity/Auteur.php
Add another property? Enter the property name (or press <return> to stop adding fields):
>
Success!
```

Figure 81 : Création des relations entre entités

Ensuite on effectue une migration vers la base données avec la commande : (`php bin/console make:migration`), qui permet de créer une migration SQL à partir des entités présentes. Et pour faire une mise à jour on se sert de la commande `php bin/console doctrine:migrations:migrate`, qui permet de lancer les scripts de migrations afin de mettre à jour la base de données.

# DOSSIER PROFESSIONNEL (DP)

```
Next: When you're ready, create a migration with php bin/console make:migration
A-C-N:SymfonyProject a_c_n$ php bin/console make:migration

Success!

Next: Review the new migration "migrations/Version20210302131936.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html
A-C-N:SymfonyProject a_c_n$
```

Figure 82 : Les migrations

Ensuite je vérifie si la requête est celle que je souhaite bien exécuter pour modifier la base de données et je valide (yes).

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette septième compétence :

Documentation: Site de Symfony.

Logiciel: Visual studio code, Mamp, Mysql, Composer.

Site web: Github desktop.

Framework: Symfony5.

Application: PhpMyAdmin.

Langage: Html5, css3, Php, Sql, Twig.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 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.

*Compétence n°8 ▶ Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce*

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

Pour cette huitième compétence j'ai étudié la création d'un **plugin WordPress** pour créer un **livre d'Or**.

Les **extensions** ou **plugins** en anglais sont des fonctionnalités que l'on ajoute à un site WordPress pour réaliser une application web.

On crée un **plugin** pour plusieurs raisons : pour alléger le functions.php du **thème enfant**, pour faire suivre le code, réutiliser le code et également pour prévenir les conflits et incompatibilités.

Je débute par l'installation de **WordPress**.

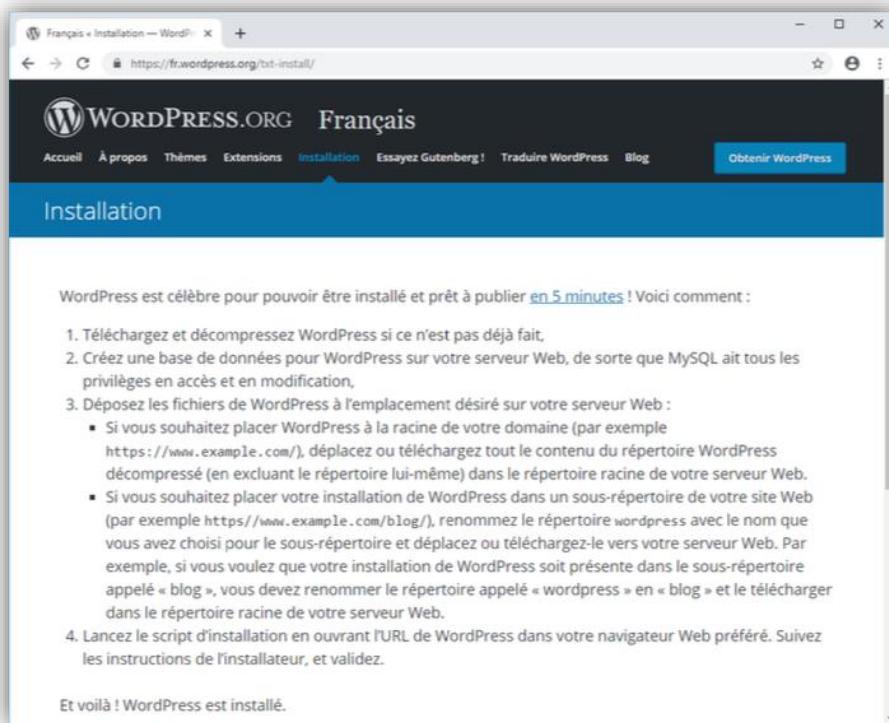


Figure 83 : Page de la documentation de l'installation de WordPress

# DOSSIER PROFESSIONNEL (DP)

A la suite je créer la **base de données**.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'Plugin\_Wp'. The left sidebar lists databases and tables. The main area displays a table of 12 database tables with columns for Action, Rows, Type, Collation, Size, and Overhead. The 'wp\_posts' table has 543 rows and a size of 1.8 MB.

| Table                 | Action                                    | Rows | Type   | Collation              | Size      | Overhead |
|-----------------------|---|------|--------|------------------------|-----------|----------|
| wp_commentmeta        | Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_520_ci | 48.0 KIB  | -        |
| wp_comments           | Browse Structure Search Insert Empty Drop | 1    | InnoDB | utf8mb4_unicode_520_ci | 96.0 KIB  | -        |
| wp_links              | Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_520_ci | 32.0 KIB  | -        |
| wp_options            | Browse Structure Search Insert Empty Drop | 191  | InnoDB | utf8mb4_unicode_520_ci | 1.1 MIB   | -        |
| wp_postmeta           | Browse Structure Search Insert Empty Drop | 218  | InnoDB | utf8mb4_unicode_520_ci | 80.0 KIB  | -        |
| wp_posts              | Browse Structure Search Insert Empty Drop | 94   | InnoDB | utf8mb4_unicode_520_ci | 224.0 KIB | -        |
| wp_termmeta           | Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_520_ci | 48.0 KIB  | -        |
| wp_terms              | Browse Structure Search Insert Empty Drop | 3    | InnoDB | utf8mb4_unicode_520_ci | 48.0 KIB  | -        |
| wp_term_relationships | Browse Structure Search Insert Empty Drop | 9    | InnoDB | utf8mb4_unicode_520_ci | 32.0 KIB  | -        |
| wp_term_taxonomy      | Browse Structure Search Insert Empty Drop | 3    | InnoDB | utf8mb4_unicode_520_ci | 48.0 KIB  | -        |
| wp_usermeta           | Browse Structure Search Insert Empty Drop | 23   | InnoDB | utf8mb4_unicode_520_ci | 48.0 KIB  | -        |
| wp_users              | Browse Structure Search Insert Empty Drop | 1    | InnoDB | utf8mb4_unicode_520_ci | 64.0 KIB  | -        |
| 12 tables             | Sum                                       | 543  | InnoDB | utf8_general_ci        | 1.8 MIB   | 0 B      |

Figure 84 : Base de données pour WordPress

Je place le dossier **WordPress1** dans le dossier (**htdocs**).

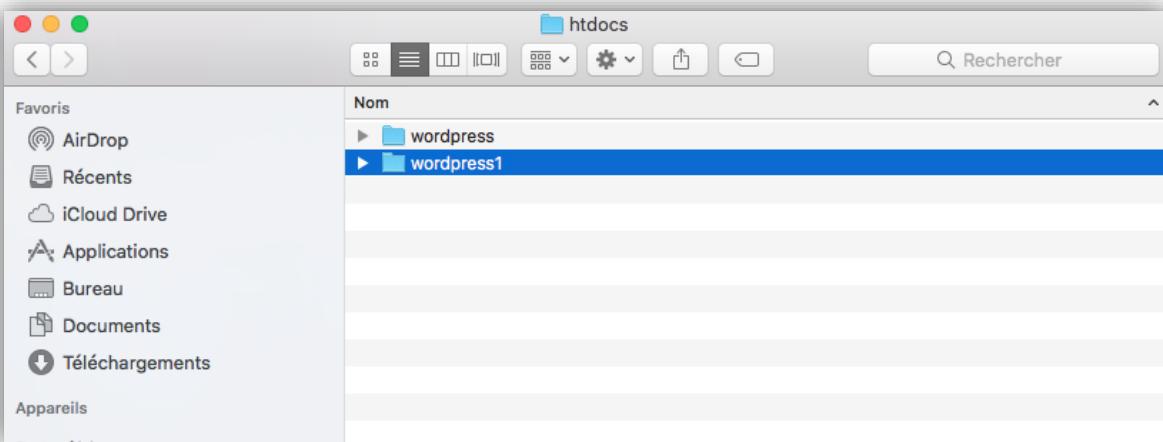


Figure 85 : Le Dossier du projet WordPress

# DOSSIER PROFESSIONNEL (DP)

Je lance le script d'installation en ouvrant l'URL de WordPress dans mon navigateur Web.

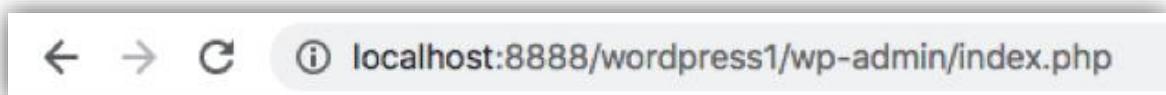
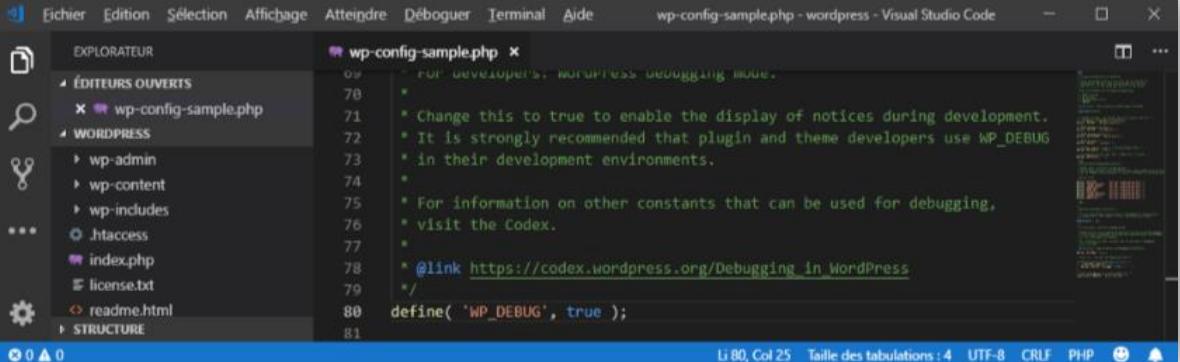


Figure 86 : Url de WordPress

A screenshot of the WordPress dashboard. The top navigation bar shows the URL "localhost:8888/wordpress1/wp-admin/index.php". The dashboard header says "Tableau de bord". On the left, there's a sidebar with various menu items like Accueil, Articles, Products, etc. The main content area has sections for "Bienvenue sur WordPress!", "Étapes suivantes", "Plus d'actions", "État de santé du site", "D'un coup d'œil", and "Brouillon rapide". The "Brouillon rapide" section includes fields for Titre and Contenu, and a "Enregistrer le brouillon" button. The "État de santé du site" section indicates potential improvements.

Figure 87 : Tableau de bord de WordPress

# DOSSIER PROFESSIONNEL (DP)



The screenshot shows a Visual Studio Code window with the title "wp-config-sample.php - wordpress - Visual Studio Code". The left sidebar displays a file tree with "wp-admin", "wp-content", "wp-includes", ".htaccess", "index.php", "license.txt", and "readme.html". The main editor pane contains the following code:

```
70 * Change this to true to enable the display of notices during development.
71 * It is strongly recommended that plugin and theme developers use WP_DEBUG
72 * in their development environments.
73 *
74 * For information on other constants that can be used for debugging,
75 * visit the Codex.
76 *
77 * @link https://codex.wordpress.org/Debugging_in_WordPress
78 */
79
80 define('WP_DEBUG', true);
```

Bottom status bar: Li 80, Col 25 Taille des tabulations : 4 UTF-8 CRLF PHP

Figure 88 : Script de l'installation généré par WordPress

Je génère le fichier **plugin.php**.

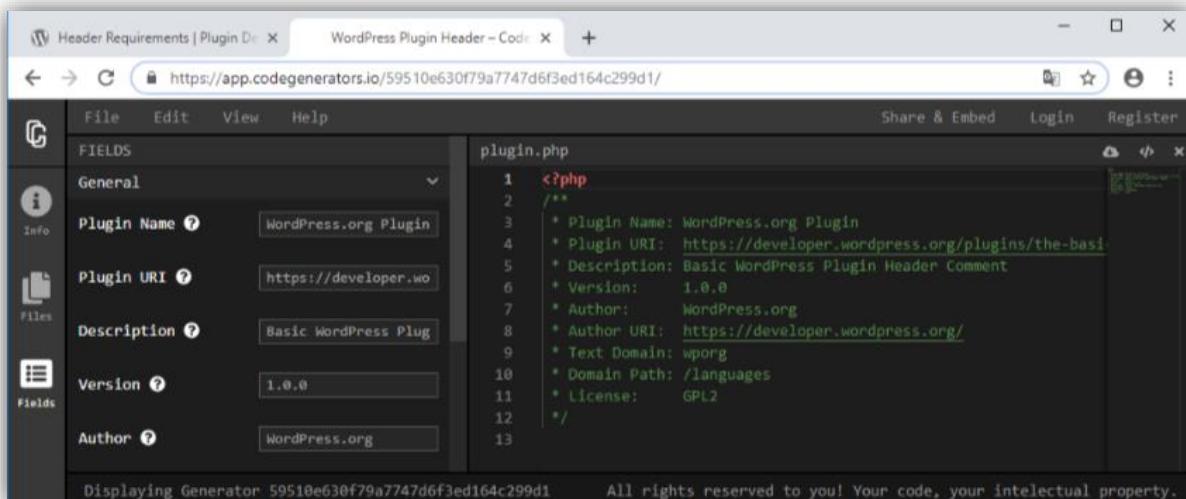


Figure 89 : Outil pour générer le fichier plugin.php

Je crée un nouveau fichier que je nomme **alecaddd-plugin.php** qui correspond à une fonctionnalité de l'application.

J'enregistre mon fichier et je le range dans un dossier que je nomme par exemple **mon-premier-plugin** qui lui-même se trouve dans le dossier **plugins**.

# DOSSIER PROFESSIONNEL (DP)

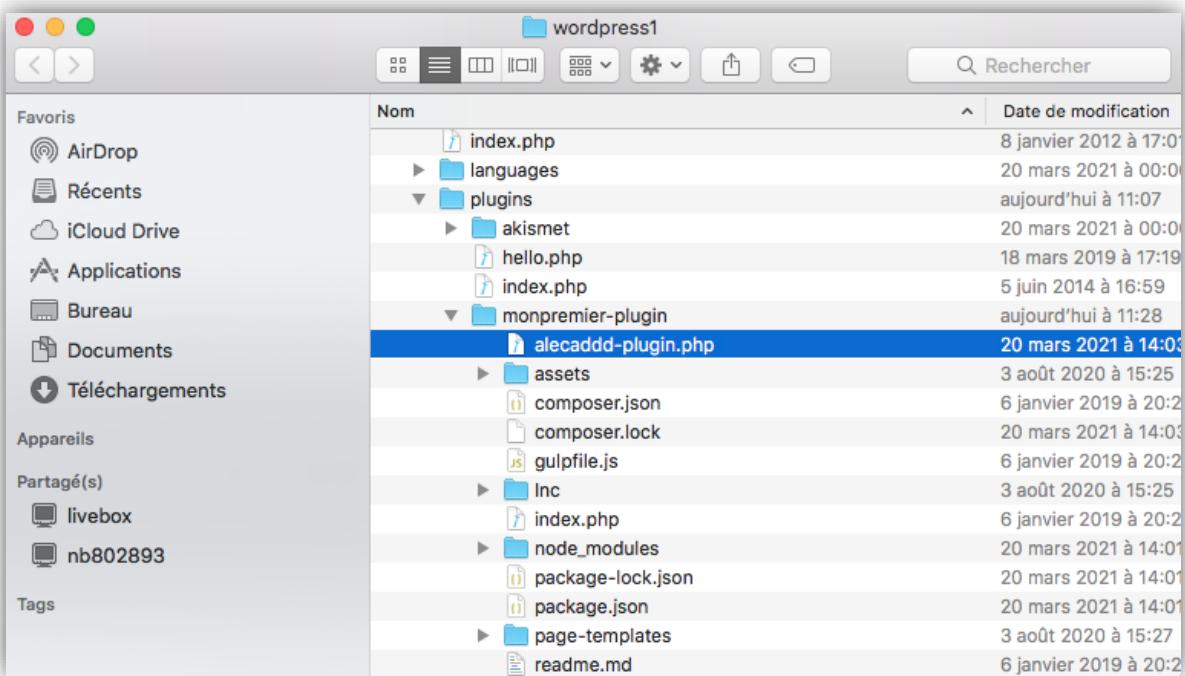


Figure 90 : Fichier alecaddd-plugin.php

Ensuite Je teste mon plugin et je vérifie s'il fonctionne bien une fois activé depuis le **back-office** de mon application **WordPress** en développement.

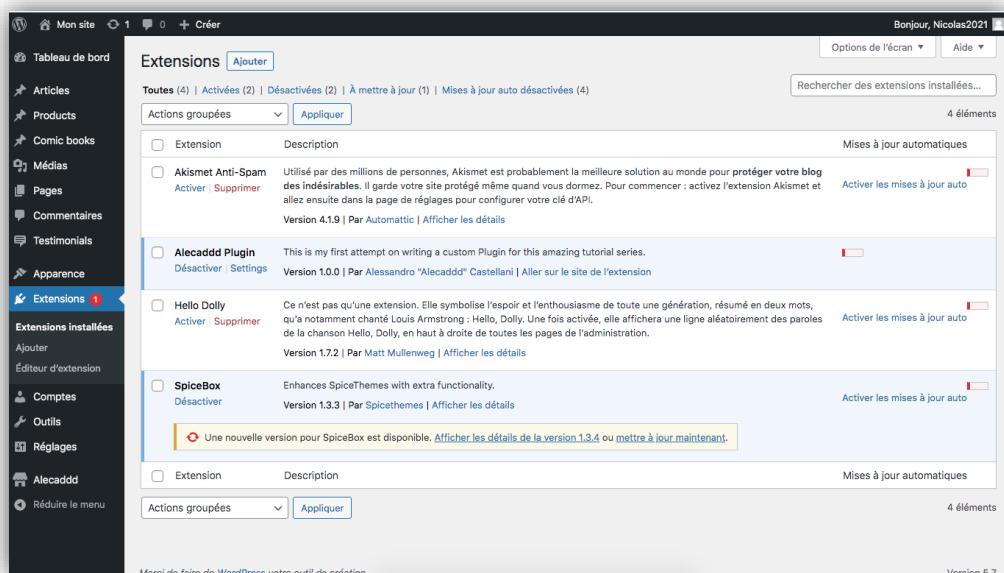


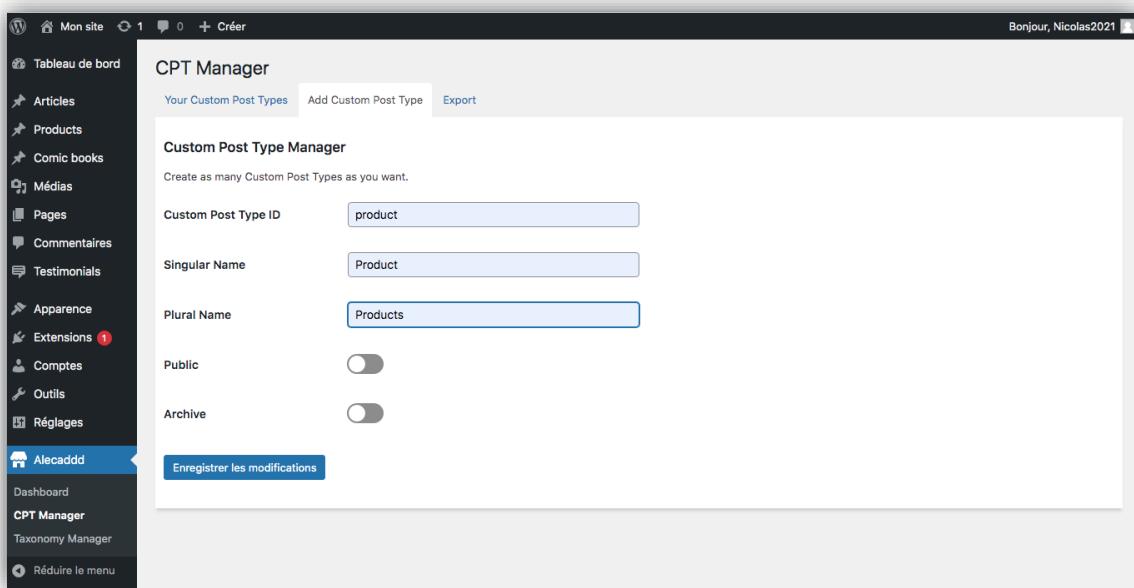
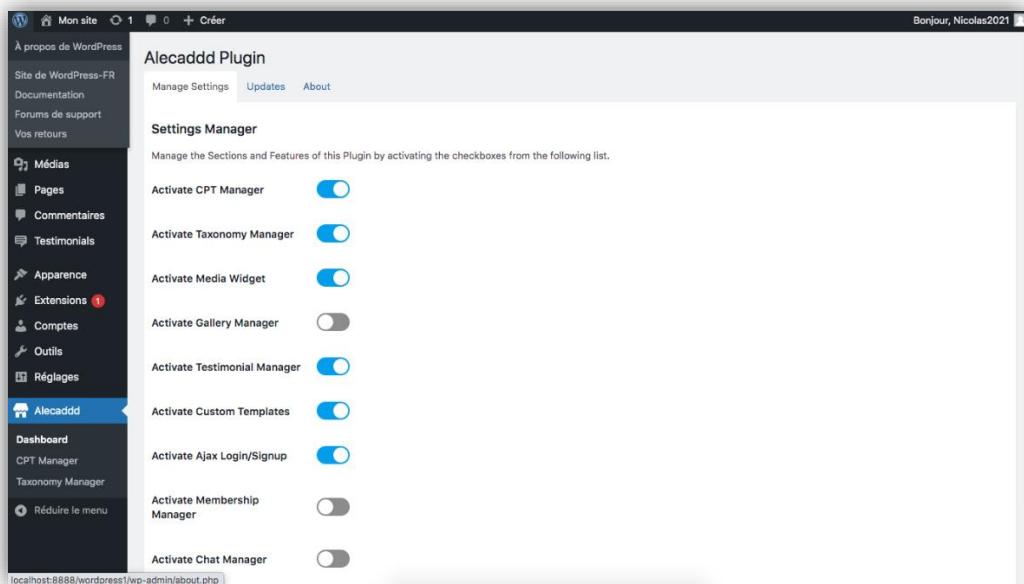
Figure 91 : Activation du plugin Alecaddd Plugin

# DOSSIER PROFESSIONNEL (DP)

Par la suite Je peux envoyer mon **plugin** dans mon application **WordPress** en production soit via **FTP**, soit via le **back-office** mais dans ce cas il doit être compressé.

A la suite je commence la configuration du plugin selon mes besoin pour mon projet.

Je créer les **Custom Post Types** qui vont me permettre de créer des types de contenus sur-mesure associant ou non une taxonomie particulière.



# DOSSIER PROFESSIONNEL (DP)

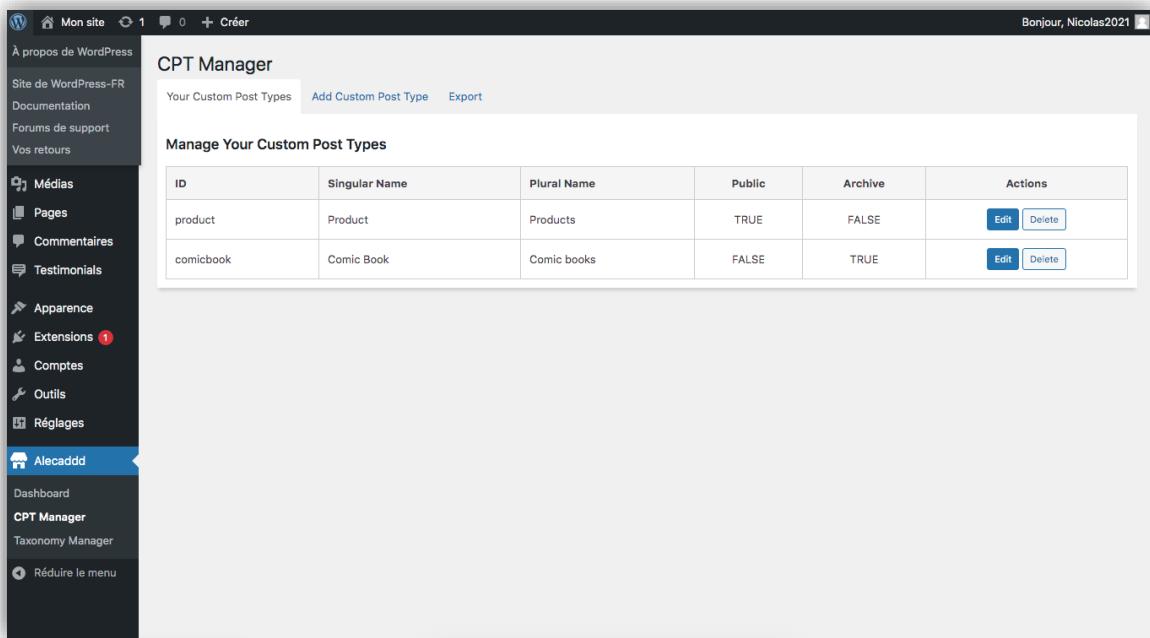


Figure 92 : Configuration du Plugin

Maintenant que mon **plugin** est configuré comme je le souhaite je peux y ajouter les articles que je souhaite voici un exemple :

# DOSSIER PROFESSIONNEL (DP)

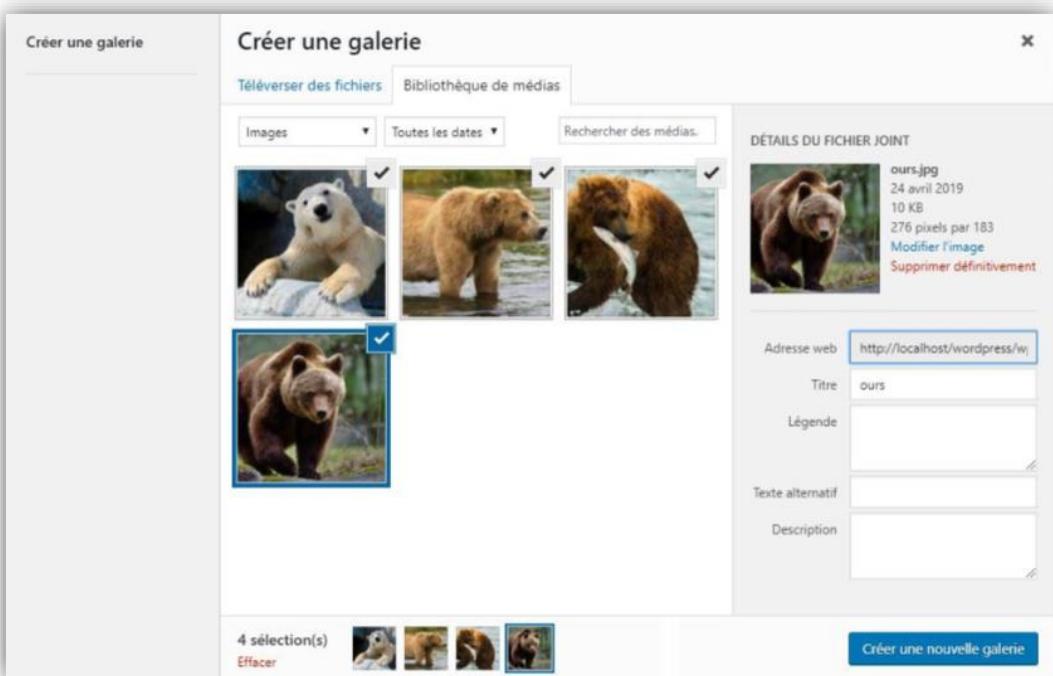


Figure 93 : Ajout des articles

Et je configure ma fonction **Testimonials** en y intégrant un article Média.

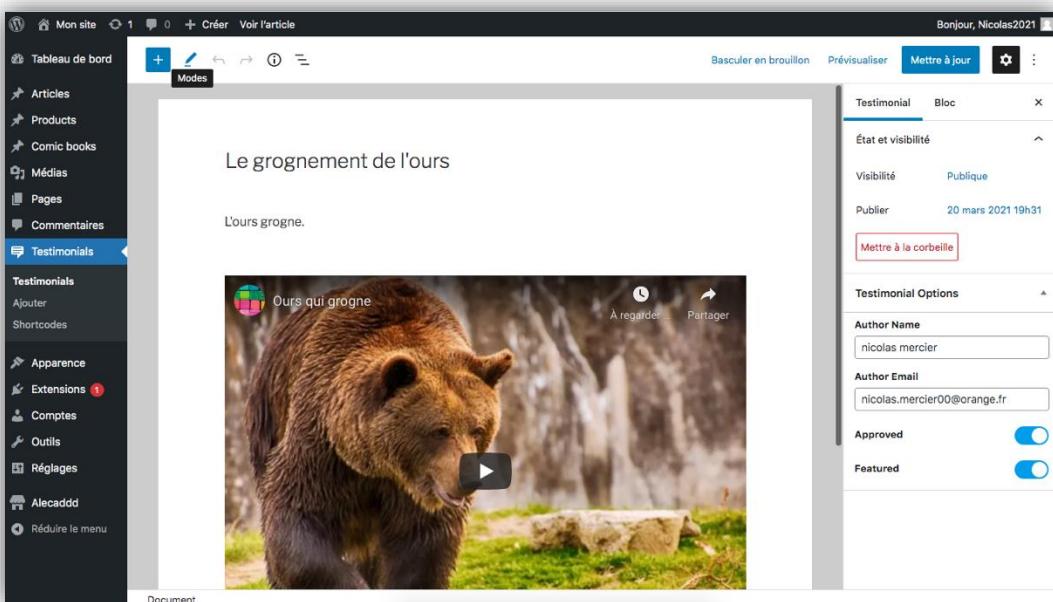


Figure 94: Ajout d'un article média

Ainsi qu'un article pour poster un témoignage à partir de mon **livre d'or**.

# DOSSIER PROFESSIONNEL (DP)

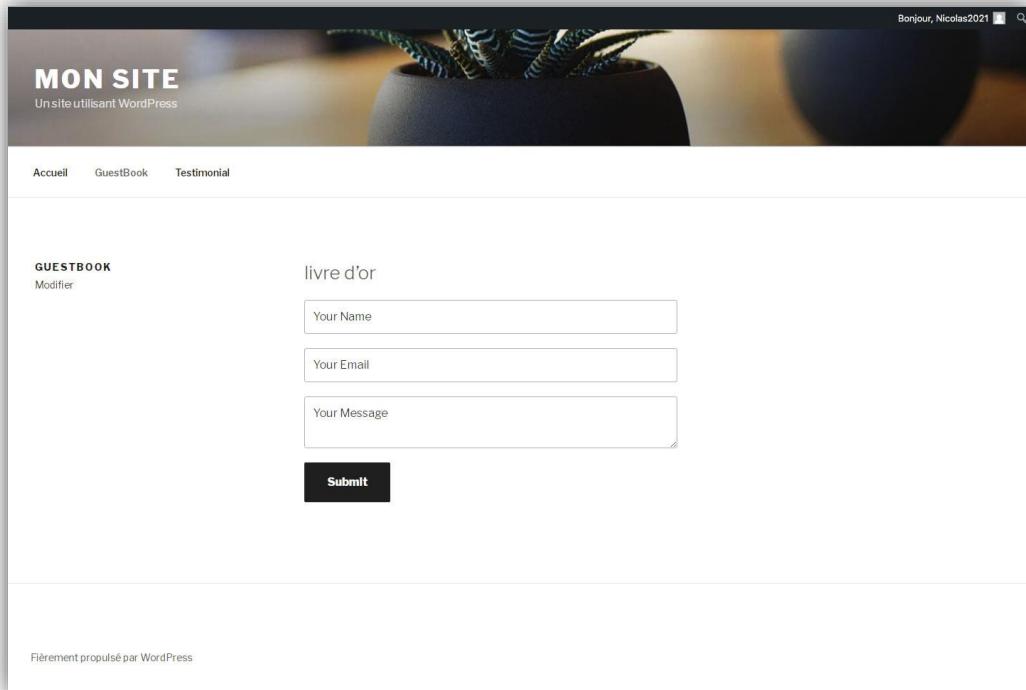


Figure 95 : Livre d'or

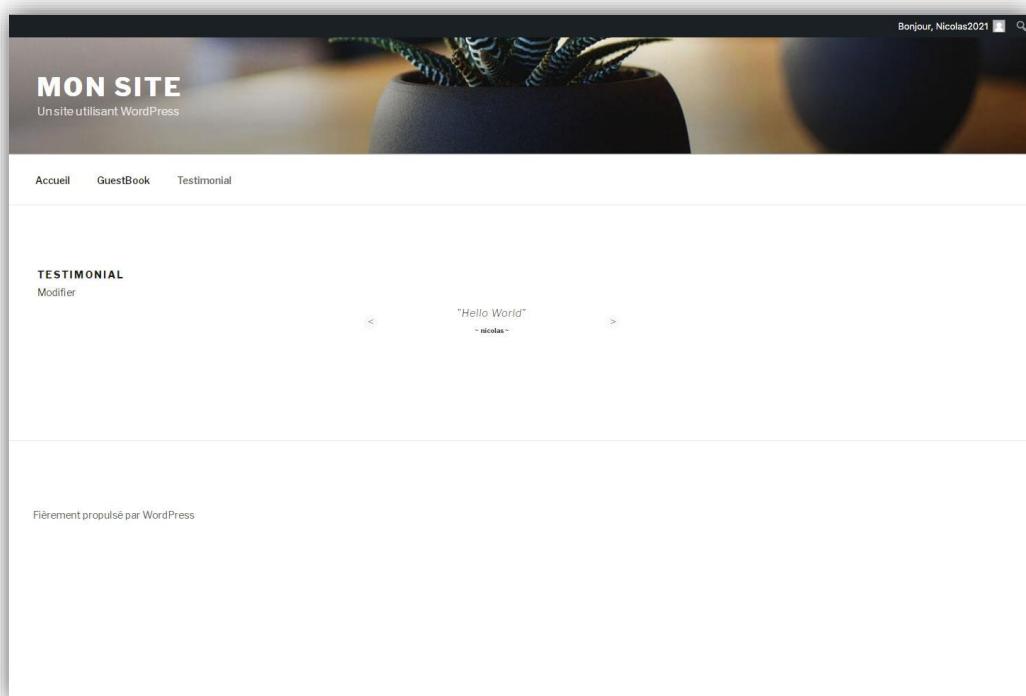


Figure 96: Poste d'un témoignage

Le résultat de ma page d'accueil de mon projet **Plugin WordPress**.

# DOSSIER PROFESSIONNEL (DP)

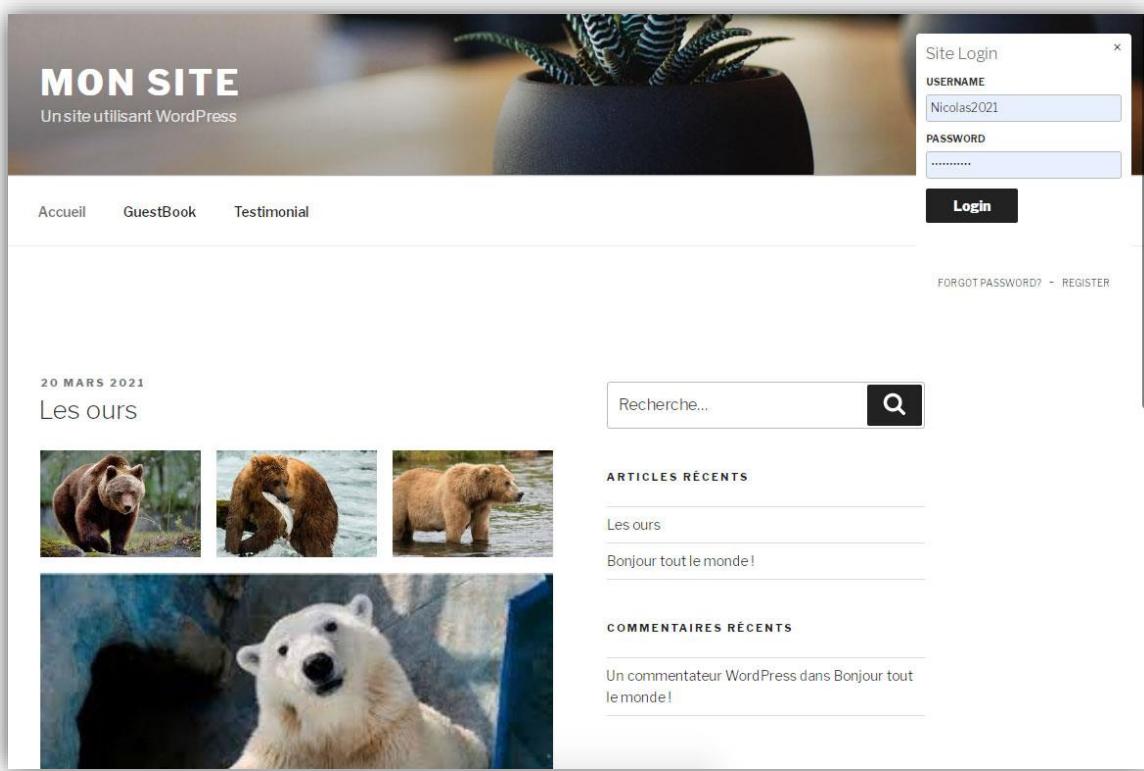


Figure 97 : Page d'accueil WordPress

## 2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisé pour cette huitième compétence :

Documentation: Site WordPress.

Logiciel: Visual studio code, Mamp, Mysql.

Site web: Github desktop.

Cms: WordPress.

Application: PhpMyAdmin.

Langage: Html5, Css3, Php.

## 3. Avec qui avez-vous travaillé ?

Pour les Compétences de 1 à 8 j'ai travaillé seul en me documentant sur internet

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Formation AFPA*

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

Période d'exercice ► Du : *05/10/2020* au : *11/06/2021*

## 5. Informations complémentaires ( *facultatif*)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 Cliquez ici pour entrer l'intitulé de l'activité

*Exemple n° 1 ▶ Cliquez ici pour entrer l'intitulé de l'exemple*

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

**2. Précisez les moyens utilisés :**

**3. Avec qui avez-vous travaillé ?**

**4. Contexte**

Nom de l'entreprise, organisme ou association ▶ *Cliquez ici pour taper du texte.*

Chantier, atelier, service ▶ *Cliquez ici pour taper du texte.*

Période d'exercice ▶ Du : *Cliquez ici* au : *Cliquez ici*

**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) [prénom et nom] Nicolas Mercier .....

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 Limeil-Brévannes ..... le 11/06/2021 .....

Pour faire valoir ce que de droit.

Signature :

## **DOSSIER PROFESSIONNEL (DP)**

## Documents illustrant la pratique professionnelle

(facultatif)

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

(*Si le RC le prévoit*)