



# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ GRECO  
*Nom d'usage* ▶  
*Prénom* ▶ Benjamin  
*Adresse* ▶ 40 rue des Clématites 34000 Montpellier

## Titre professionnel visé

*Développeur Web et Web Mobile*

### MODALITÉ D'ACCÈS :

- ☒ 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 :**

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.

du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.

des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation

de l'entretien final (dans le cadre de la session titre).

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

### **Ce dossier comporte :**

pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;

un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;

une déclaration sur l'honneur à compléter et à signer ;

des documents illustrant la pratique professionnelle du candidat (facultatif)

des annexes, si nécessaire.

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



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

## Sommaire

### Exemples de pratique professionnelle

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

p. 5

- ▶ CP 1 Maquetter une application. p. 5
- ▶ CP 2 Réaliser une interface statique et adaptable. p. 9
- ▶ CP 3 Développer une interface utilisateur web dynamique. p. 16

#### Intitulé de l'activité-type n° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.

p.

- ▶ CP 5 Créer une base de données. p. 23
- ▶ CP 6 Développer les composants d'accès aux données. p. 26
- ▶ CP 7 Développer la partie back-end d'une application web ou web mobile. p. 30

**Titres, diplômes, CQP, attestations de formation** *(facultatif)*

p. 36

**Déclaration sur l'honneur**

p. 37

**Documents illustrant la pratique professionnelle** *(facultatif)*

p. 38

**Annexes** *(Si le RC le prévoit)*

p. 39

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1

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

### CP 1 ► Maquetter une application.

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

Pour réaliser cette première compétence, je vais travailler sur la préparation d'un projet que j'ai dû réaliser au cours de ma formation, un site en Responsive Web Design qui regroupera des recettes de cuisine. Nous avons eu 3 heures pour répondre à ce mini cahier des charges, wireframes et user stories incluses. Ici, il s'agit d'un MVP (minimum viable product), et donc les fonctionnalités à implémenter dans un premier temps sont assez peu nombreuses.

Voici la liste des fonctionnalités attendues sur cette première version :

En haut de la page :

- il faut afficher un logo (au clic sur ce logo, on revient sur la page d'accueil)
- le menu doit permettre de naviguer sur les pages principales : accueil, recettes de saison, entrées, plats, desserts
- 

Sur la page d'accueil :

- on doit retrouver un édito (un petit paragraphe susceptible de changer de façon hebdomadaire)
- on doit trouver un lien vers une recette au hasard. A chaque fois qu'on rafraîchit la page, on affiche une recette différente
- il faut afficher une liste des 5 recettes les plus récentes
- il faut afficher une liste des 5 recettes les mieux notées
- 

Dans le pied de page :

- on veut afficher des liens vers la page de contact, les mentions légales et vers le formulaire permettant de proposer une recette, ainsi qu'une citation résumant le principe du site et pouvant être utile au référencement (SEO)
- 

Sur la page de recette :

- on doit voir la liste des ingrédients séparée du reste de la recette
- on doit voir les étapes de la recette identifiée notamment par un numéro

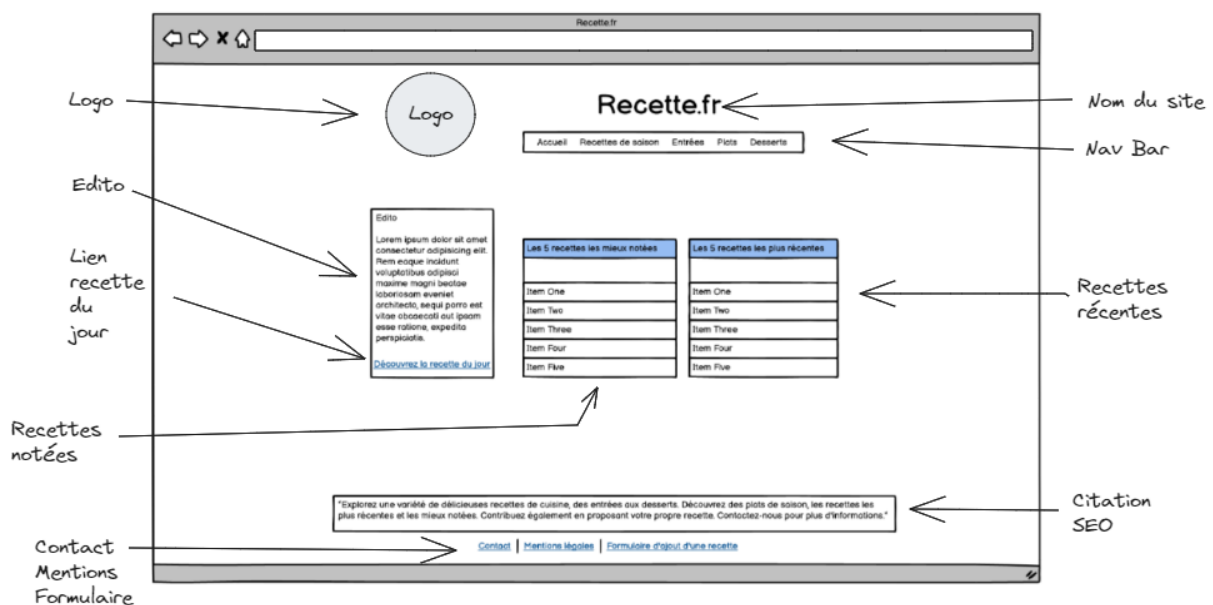
User stories :

## User stories

Fiche récap : <https://kourou.oclock.io/ressources/fiche-recap/user-stories/>

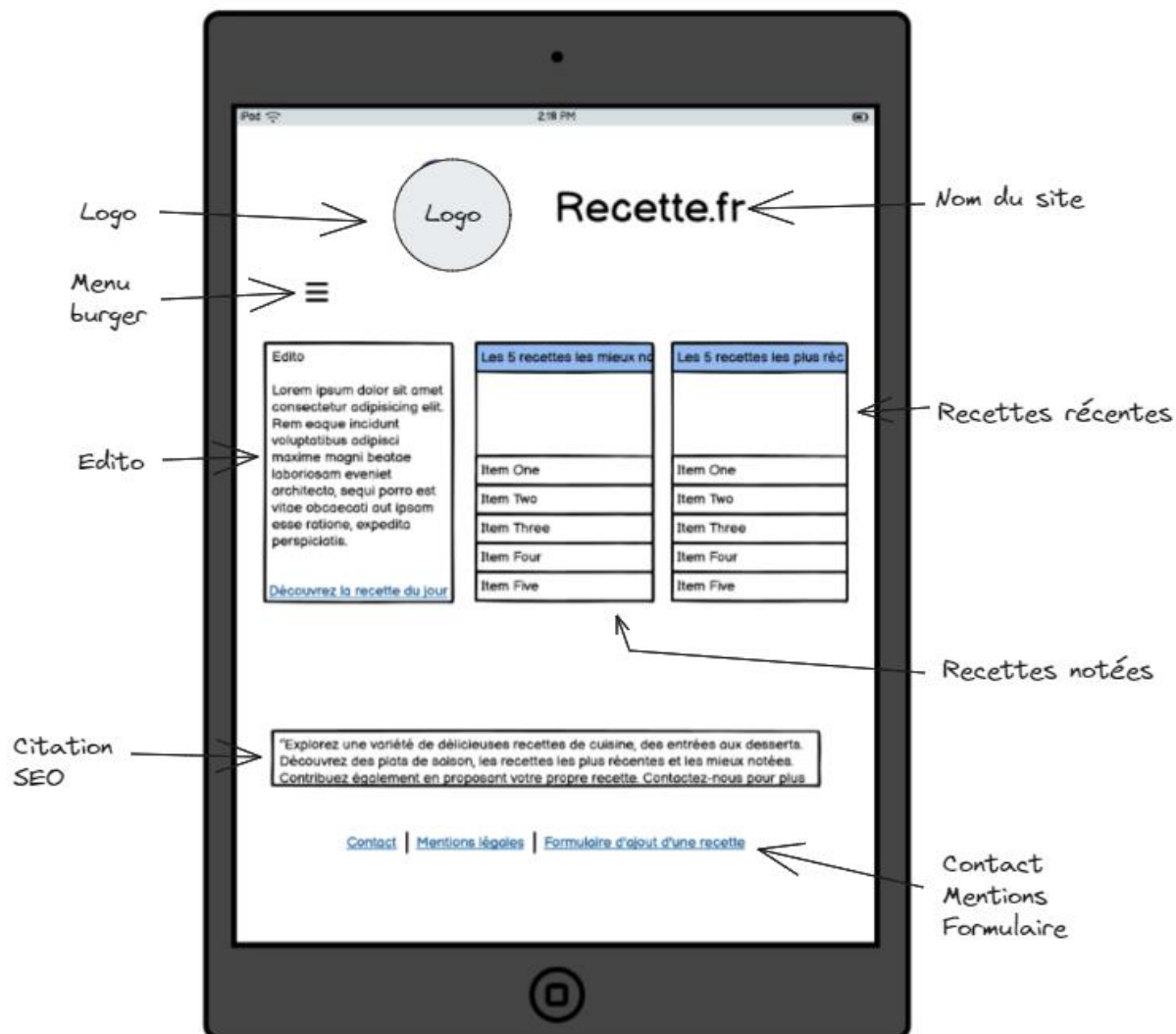
En tant que	Je veux	Afin de (si besoin/nécessaire)
Visiteur	Pouvoir accéder aux recettes de saison	Consulter les différentes recettes
Visiteur	Pouvoir accéder aux entrées	Consulter les différentes entrées
Visiteur	Pouvoir accéder aux plats	Consulter les différents plats
Visiteur	Pouvoir accéder aux desserts	Consulter les différents desserts
Visiteur	Pouvoir accéder au formulaire de contact	Contacter les responsables du site
Visiteur	Pouvoir accéder aux mentions légales	Consulter les mentions légales
Visiteur	Pouvoir accéder à la liste des ingrédients d'une recette	Réaliser celle-ci dans ma cuisine
Visiteur	Pouvoir visualiser les étapes d'une recette	Suivre les instructions de préparation
Visiteur	Pouvoir rechercher des recettes en fonction de certains critères	Trouver des recettes correspondant à mes besoins/envies
Visiteur	Pouvoir noter et donner mon avis sur une recette	Partager mon expérience avec la communauté
Visiteur	Pouvoir sauvegarder des recettes dans une liste de favoris	Retrouver facilement mes recettes préférées
Utilisateur enregistré	Pouvoir ajouter mes propres recettes au site	Les partager avec la communauté
Utilisateur enregistré	Pouvoir modifier ou supprimer mes recettes publiées	Mettre à jour ou retirer mes recettes si nécessaire
Administrateur	Pouvoir gérer les recettes, y compris l'approbation des recettes proposées	Maintenir la qualité et la pertinence du contenu
Administrateur	Pouvoir gérer les utilisateurs, y compris la création, modification et suppression de comptes	Assurer la sécurité et la gestion du site

Version desktop :

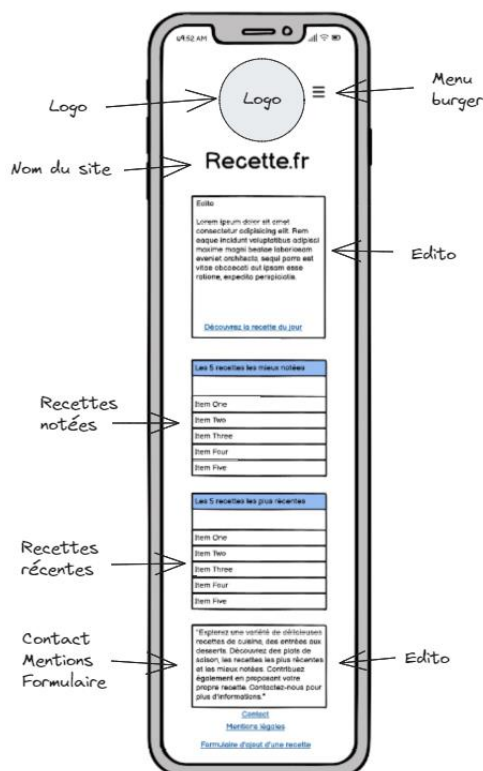


# DOSSIER PROFESSIONNEL (DP)

Version tablette :



## Version mobile :



## 2. Précisez les moyens utilisés :

Balsamiq et Excalidraw pour les wireframes et Markdown dans VsCode pour les User stories

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ecole O'Clock*

Chantier, atelier, service ► *Formation*

Période d'exercice ► Du : *27 mars 2023* au : *10 septembre 2023*

## 5. Informations complémentaires (facultatif)

<https://balsamiq.com/>

<https://excalidraw.com/>



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

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

Bootstrap est un framework de développement web open source largement utilisé qui simplifie la création de sites web responsives. Bootstrap inclue une grille flexible qui facilite la mise en page des éléments de la page.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <meta name="description" content="Explorez une variété de délicieuses recettes de cuisine, des entrées aux des
9   Découvrez des plats de saison, les recettes les plus récentes et les mieux notées. Contribuez également en pro
10  votre propre recette. Contactez nous pour plus d'informations.">
11   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"
12   integrity="sha384-9ndCyUa1bzAi2lUXqAb1Zv3DxUB9O6BhF6q9QKh8nq9P7j8UwYkavkX6iW6DvlVM" crossorigin="anonymous"
13   <title>Recette</title>
14 </head>
15
16 <body>
17   <header>
18     <div class="container">
19       <div class="row align-items-center">
20         <div class="col-6 text-center">
21           
22         </div>
23         <div class="col-12 col-md-6 text-md-start text-center">
24           <h1>Recette </h1>
25         </div>
26         <nav class="navbar navbar-expand-lg navbar-light">
27           <div class="container-fluid">
28             <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#na
29             aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
30               <span class="navbar-toggler-icon"></span>
31             </button>
32             <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
33               <ul class="navbar-nav mx-auto">
34                 <li class="nav-item">
35                   <a class="nav-link" href="#">Accueil</a>
36                 </li>
37                 <li class="nav-item">
38                   <a class="nav-link" href="#">Recettes de saison</a>
39                 </li>
40                 <li class="nav-item">
41                   <a class="nav-link" href="#">Entrées</a>
42                 </li>
43                 <li class="nav-item">
44                   <a class="nav-link" href="#">Plats</a>
45                 </li>
46                 <li class="nav-item">
47                   <a class="nav-link" href="#">Desserts</a>
48                 </li>
49               </ul>
50             </div>
51           </div>
52         </nav>
53       </div>
54     </div>
55   </header>

```

J'inclue la feuille de style CSS de Bootstrap dans le code HTML, cela signifie que les styles prédéfinis par Bootstrap seront appliqués à la page. J'ajoute un élément meta avec la propriété name définie sur "description" et le contenu défini sur le texte de votre citation qui sera associé au footer pour améliorer le SEO. J'utilise la classe "container" pour envelopper le contenu principal du header, ce qui crée une mise en page réactive qui s'ajuste automatiquement à différentes tailles d'écran. De plus, j'utilise des classes de grille Bootstrap comme "col-6" et "col-md-6" pour définir la largeur des colonnes en fonction de la largeur de l'écran, les classes "navbar", "navbar-expand-lg", "navbar-light" pour configurer la barre de navigation. La barre de navigation est également équipée d'un bouton de basculement (navbar-toggler) qui s'affiche sur les écrans plus petits pour permettre la navigation.

Main :

Pour le Main , j'utilise à nouveau la classe "container" pour envelopper le contenu principale et ensuite la classe "row" pour aligner horizontalement mes éléments.

Les classes "col-md-4" et "col-lg-2" sont utilisées pour définir la largeur des colonnes dans la grille Bootstrap. Ces classes indiquent que la première colonne occupe 4 colonnes sur les écrans de taille moyenne (md) et 2 colonnes sur les écrans de grande taille (lg).

```
53
54
55 <main>
56   <div class="container text-center mt-5">
57     <div class="row">
58       <div class="col-md-4 col-lg-2 border">
59         <h4>Edito hebdomadaire</h4>
60         <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugiat, voluptatum reiciendis. Tenetur ipsum rem repellendus.
61         <a href="#">Lien vers une recette au hasard</a>
62       </div>
63       <div class="col-md-4">
64         <table class="table table-bordered">
65           <thead>
66             <tr>
67               <th>Les 5 recettes les plus récentes</th>
68             </tr>
69           </thead>
70           <tbody>
71             <tr>
72               <td><a href="#">Recette 1</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
73             </tr>
74             <tr>
75               <td><a href="#">Recette 2</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
76             </tr>
77             <tr>
78               <td><a href="#">Recette 3</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
79             </tr>
80             <tr>
81               <td><a href="#">Recette 4</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
82             </tr>
83             <tr>
84               <td><a href="#">Recette 5</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
85             </tr>
86           </tbody>
87         </table>
88       </div>
89       <div class="col-md-4">
90         <table class="table table-bordered">
91           <thead>
92             <tr>
93               <th>Les 5 recettes les mieux notées</th>
94             </tr>
95           </thead>
96           <tbody>
97             <tr>
98               <td><a href="#">Recette A</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
99             </tr>
100            <tr>
101              <td><a href="#">Recette B</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
102            </tr>
103            <tr>
104              <td><a href="#">Recette C</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
105            </tr>
106            <tr>
107              <td><a href="#">Recette D</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
108            </tr>
109            <tr>
110              <td><a href="#">Recette E</a><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></td>
111            </tr>
112          </tbody>
113        </table>
114      </div>
115    </div>
116  </div>
117 </main>
```

# DOSSIER PROFESSIONNEL (DP)

```

123 <footer>
124   <div class="container">
125     <div>
126       <div class="text-center col-md-10 mt-5 mb-5">
127         <p itemprop="description">Explorez une variété de délicieuses recettes
128           de cuisine, des entrées aux desserts. Découvrez des plats de saison
129           les recettes les plus récentes et les mieux notées.
130           Contribuez également en proposant votre propre recette.
131           Contactez nous pour plus d'informations.
132       </div>
133     </div>
134     <div class="col-md-10 d-flex justify-content-center">
135       <a href="#" class="card-link">Contact</a>
136     </div>
137     <div class="col-md-10 d-flex justify-content-center">
138       <a href="#" class="card-link">Mentions légales</a>
139     </div>
140     <div class="col-md-10 d-flex justify-content-center">
141       <a href="#" class="card-link">Formulaire d'ajout d'une recette</a>
142     </div>
143   </div>
144 </footer>
145 </body>
146 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
147   integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM800ewa9r"
148 </script>
149 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js"
150   integrity="sha384-fbb0QedDUMZSKreZpsb1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bG8q4leYeM/fS"
151 </script>
152 </html>
153

```

Footer :

Ici je réutilise "container" pour envelopper le pied de page.

Les classes "col-md-10" sont utilisées pour définir la largeur des colonnes à l'intérieur du pied de page. Elles indiquent que le contenu à l'intérieur de ces colonnes doit occuper 10 colonnes sur les écrans de taille moyenne (md).

J'utilise également l'attribut "itemprop" pour indiquer aux moteurs de recherche que le paragraphe est une citation importante pour améliorer le référencement du site.

Voici les six "breakpoints" qu'utilise Bootstrap :

## Available breakpoints #

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
Extra small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are also representative of a subset of common device sizes and viewport dimensions—they don't specifically target every use case or device. Instead, the ranges provide a strong and consistent foundation to build on for nearly any device.

Voici le rendu du code :

Version ≥992px



Version ≥768px

# DOSSIER PROFESSIONNEL (DP)



Recette.fr



## Édito hebdomadaire

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.

Fugiat, voluptatum  
reiciendis. Tenetur ipsum  
rem repellendus deserunt  
temporibus voluptas eum  
esse. Facere saepe,  
quisquam beatae nostrum  
cumque unde mollitia, eos  
a animi natus voluptate  
dignissimos corrupti sunt in  
blanditiis recusandae culpa,  
excepturi perferendis  
voluptas. Enim, recusandae  
facilis?

[Lien vers une recette au  
hasard](#)

## Les 5 recettes les plus récentes

### [Recette 1](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette 2](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette 3](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette 4](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette 5](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

## Les 5 recettes les mieux notées

### [Recette A](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette B](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette C](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette D](#)

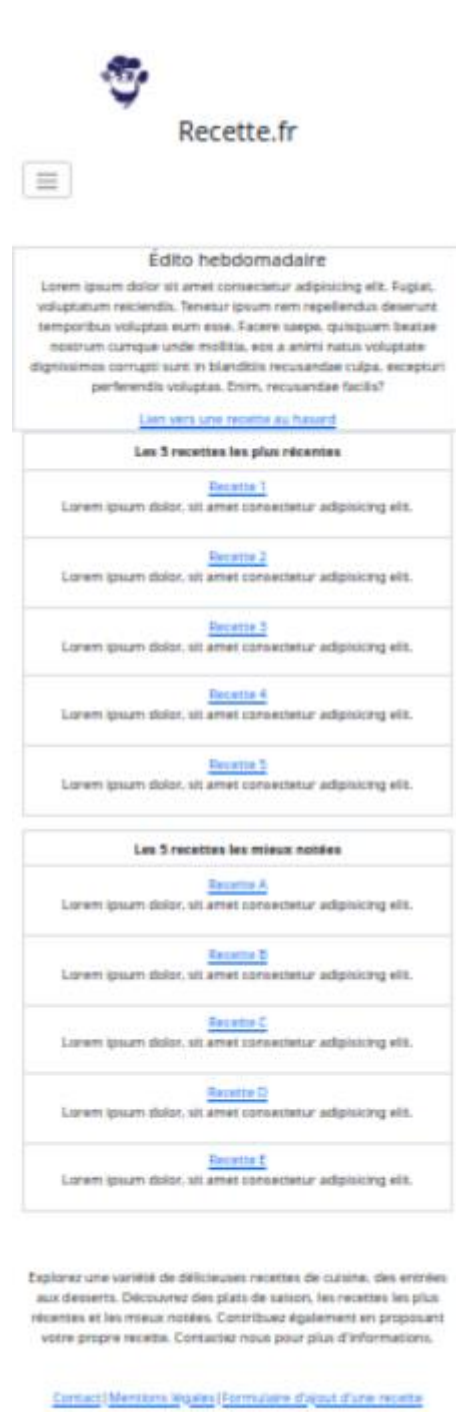
Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

### [Recette E](#)

Lorem ipsum dolor, sit  
amet consectetur  
adipisicing elit.

Explorez une variété de délicieuses recettes de cuisine, des entrées aux  
desserts. Découvrez des plats de saison, les recettes les plus récentes et les  
mieux notées. Contribuez également en proposant votre propre recette.  
Contactez nous pour plus d'informations.

[Contact](#) | [Mentions légales](#) | [Formulaire d'ajout d'une recette](#)





# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Linux, VsCode, HTML et Bootstrap

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ecole O'Clock*

Chantier, atelier, service ► *Formation*

Période d'exercice ► Du : *27 mars 2023* au : *10 septembre 2023*

## 5. Informations complémentaires (facultatif)

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://getbootstrap.com/docs/5.3/layout/breakpoints/#available-breakpoints>



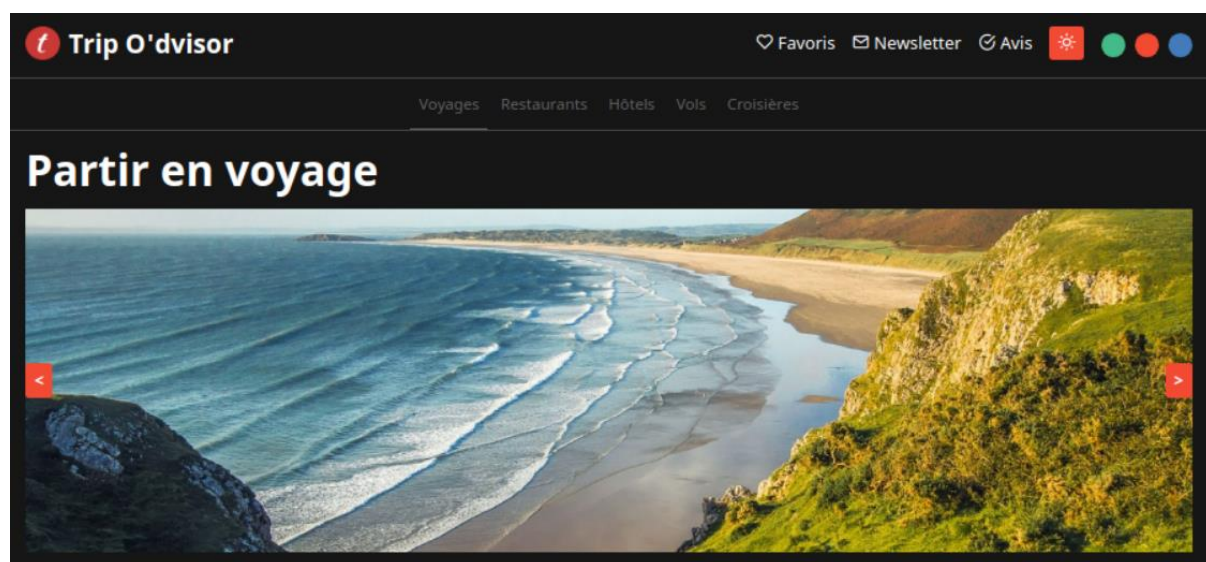
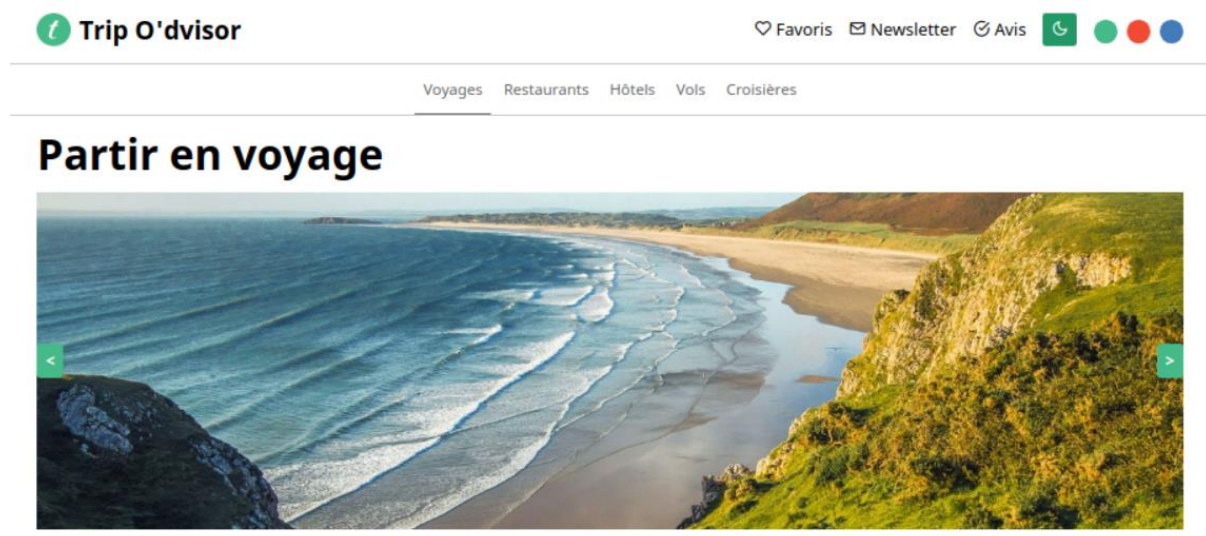
## Activité-type 1

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

CP 3 ► Développer une interface utilisateur web dynamique.

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

Pour cette compétence, je vais utiliser un projet réalisé pendant ma formation. L'objectif était de réaliser un site d'avis semblable à TripAdvisor. Nous l'avons nommé TripOdvisor. Sur ce projet j'ai inclus une fonction "darkmode" ainsi que des thèmes de couleurs pour les icônes.



HTML :



# DOSSIER PROFESSIONNEL (DP)

Dans le HTML on retrouve les boutons "theme-switch" "theme-green" "theme-red" "theme-blue", qui vont permettre par la suite de passer en "darkmode" et de changer le thème des icônes.

```

37 <header class="header">
38   <nav>
39     <div class="header_part">
40       <div class="container">
41         <div class="header_flex">
42           <a class="logo" href="">
43             
44             <span class="logo_title">Trip O'dvisor</span>
45           </a>
46           <div class="menu">
47             <a class="menu_item" href=""><i class="icon-heart"></i> Favoris</a>
48             <a id="newsletter_btn" class="menu_item" href=""><i class="icon-mail"></i> Newsletter</a>
49             <a class="menu_item" href=""><i class="icon-check-circle"></i> Avis</a>
50             <button id="theme-switch" class="btn" type="button" aria-label="Changer le thème"><i
51               class="icon-moon"></i></button>
52             <div class="menu_color-switcher">
53               <button id="theme-green" class="theme-button" type="button"
54                 aria-label="Changer la couleur du thème"><i data-feather="pen-tool"></i></button>
55               <button id="theme-red" class="theme-button" type="button"
56                 aria-label="Changer la couleur du thème"></button>
57               <button id="theme-blue" class="theme-button" type="button"
58                 aria-label="Changer la couleur du thème"></button>
59             </div>
60           </div>
61         </div>
62       </div>
63     </div>
64     <div class="header_part">
65       <div class="container">
66         <div class="submenu">
67           <a class="submenu_item submenu_item--current" href="">Voyages</a>
68           <a class="submenu_item" href="">Restaurants</a>
69           <a class="submenu_item" href="">Hôtels</a>
70           <a class="submenu_item" href="">Vols</a>
71           <a class="submenu_item" href="">Croisières</a>
72         </div>
73       </div>
74     </div>
75   </nav>
76 </header>

```

## JavaScript :

Le module est initialisé avec la fonction `init()`. Il commence par récupérer les préférences de thème depuis le stockage local. Si un thème sombre est défini dans le stockage local et que la classe "theme-dark" n'est pas présente sur le `<body>`, la fonction `handleSwitchThemeClick()` est appelée pour activer le thème sombre.

Il permet aux utilisateurs de basculer entre les thèmes clairs et sombre en cliquant sur un bouton ayant l'ID "theme-switch". Lorsque ce bouton est cliqué, la fonction `handleSwitchThemeClick()` est appelée pour changer le thème en modifiant la classe CSS de l'élément `<body>` et en mettant à jour les préférences de thème dans le stockage local.

Il permet également aux utilisateurs de choisir parmi plusieurs couleurs de thème en cliquant sur des boutons ayant la classe "theme-button". Lorsqu'un bouton de couleur est cliqué, la fonction handleSwitchColor() est appelée pour appliquer la couleur sélectionnée au <body> et supprimer les autres couleurs de thème.

```
1  const themeModule = {
2
3    init: function () {
4      themeModule.preferences = {};
5      const defaultTheme = localStorage.getItem('theme')
6      if(defaultTheme === 'dark' && document.body.classList.contains('theme-dark') === false){
7        themeModule.handleSwitchThemeClick();
8      }
9
10     const preferencesAsString = localStorage.getItem('preferences');
11     console.log(preferencesAsString)
12     themeModule.preferences = JSON.parse(preferencesAsString);
13     console.log(themeModule.preferences);
14     if (themeModule.preferences.theme === "dark" && document.body.classList.contains('theme-dark') === false) {
15       themeModule.handleSwitchThemeClick();
16     }
17
18     const btnSwitch = document.querySelector('#theme-switch');
19     //? alternative avec getElementById
20     btnSwitch.addEventListener('click', themeModule.handleSwitchThemeClick)
21
22     const colorThemeButtons = document.querySelectorAll('.theme-button');
23     console.log(colorThemeButtons)
24     for (const colorButton of colorThemeButtons) {
25       colorButton.addEventListener('click', themeModule.handleSwitchColor);
26     }
27   },
28
29   handleSwitchThemeClick: function (event) {
30     const body = document.body;
31     if (body.classList.contains('theme-dark')) {
32       body.classList.remove('theme-dark');
33       themeModule.preferences.theme = 'light';
34       const preferencesAsString = JSON.stringify(themeModule.preferences);
35       localStorage.setItem('preferences', preferencesAsString);
36     } else {
37       body.classList.add('theme-dark');
38       themeModule.preferences.theme = 'dark';
39       const preferencesAsString = JSON.stringify(themeModule.preferences);
40       localStorage.setItem('preferences', preferencesAsString);
41     }
42   },
43 }
```

Pour compléter cette compétence professionnelle, je vais maintenant utiliser un autre projet que j'ai développé en privé, il s'agit d'une application météo qui utilise 3 API afin de récupérer l'adresse IP de l'utilisateur, de géo localiser l'adresse IP en question et de définir la météo à l'endroit géo localisé. Pour ce faire, j'ai utilisé VsCode, HTML, CSS et javascript. Pour plus de simplicité de compréhension, j'ai commenté le code Javascript en français dans le code.

Les 3 API sont :

<https://www.ipify.org/> pour récupérer l'adresse IP de l'utilisateur

<https://ip-api.com/> pour géo localiser l'adresse IP

<https://openweathermap.org/> pour définir la météo à l'endroit géo localiser.

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

Javascript :

```

1  // Définition d'une correspondance entre les conditions météorologiques et les icônes
2  const weatherIcons = {
3    "Rain": "wi wi-day-rain",
4    "Clouds": "wi wi-day-cloudy",
5    "Clear": "wi wi-day-sunny",
6    "Snow": "wi wi-day-snow",
7    "Mist": "wi wi-day-fog",
8    "Drizzle": "wi wi-day-sleet",
9  }
10
11 // Fonction pour mettre en majuscule la première lettre d'une chaîne de caractères
12 function capitalize(str) {
13   return str[0].toUpperCase() + str.slice(1);
14 }
15
16 // Fonction principale
17 async function main(withIp = true) {
18   let ip;
19   let ville;
20
21   // Vérifie si nous devons obtenir l'adresse IP de l'utilisateur
22   if (withIp) {
23     // Récupère l'adresse IP de l'utilisateur en utilisant une API
24     await fetch('https://api.ipify.org?format=json')
25       .then(resultat => resultat.json())
26       .then(json => {
27         ip = json.ip;
28         // Utilise l'adresse IP pour obtenir la ville de l'utilisateur en utilisant une autre API
29         return fetch(`http://ip-api.com/json/${ip}`);
30       })
31       .then(resultat => resultat.json())
32       .then(json => {
33         ville = json.city;
34         // Utilise la ville pour obtenir les données météorologiques en utilisant l'API OpenWeatherMap
35         return fetch(`http://api.openweathermap.org/data/2.5/weather?q=${ville}&appid=8e602b9ea28ed4f9f8fc97a5f6d`);
36       })
37       .then(resultat => resultat.json())
38       .then(meteoData => {
39         // Affiche les données météorologiques dans la console
40         console.log(meteoData);
41         // Appel de la fonction displayWeatherInfos pour afficher les données dans l'interface utilisateur
42         displayWeatherInfos(meteoData);
43       });
44   } else {
45     // Utilise la valeur saisie par l'utilisateur pour obtenir les données météorologiques
46     const villeInput = document.querySelector('#ville');
47     ville = villeInput.textContent;
48   }

```

```

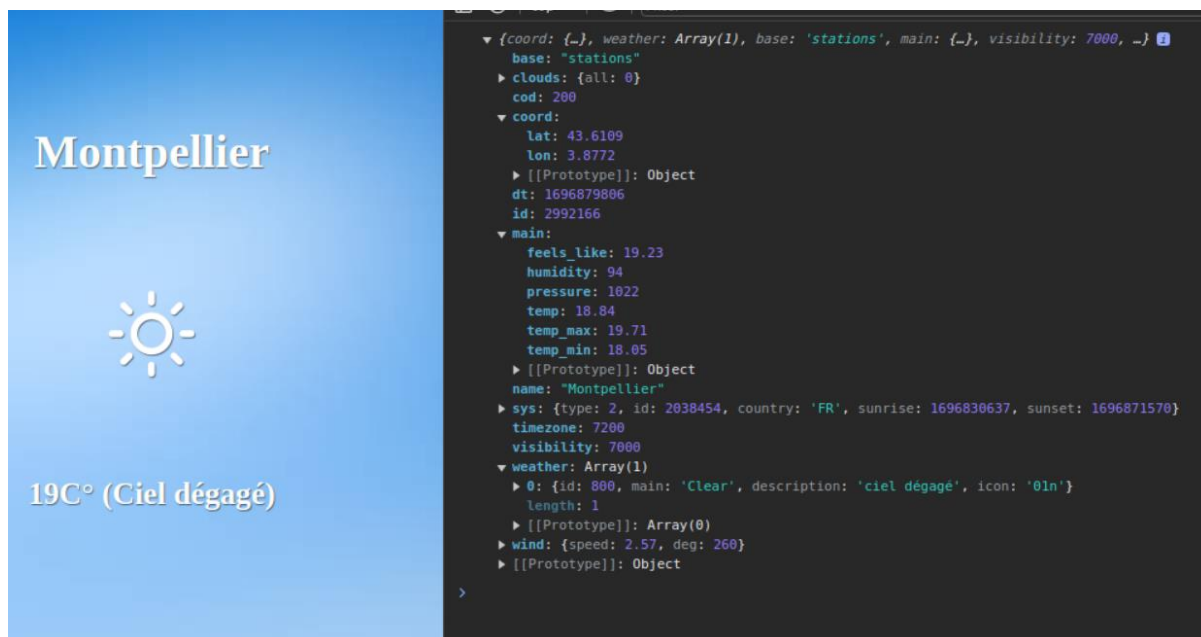
49 // Utilise la ville pour obtenir les données météorologiques en utilisant l'API OpenWeatherMap
50 fetch('http://api.openweathermap.org/data/2.5/weather?q=${ville}&appid=8e602b9ea28ed4f9f8fc97a5f6d1105c&lang=fr&units=met
51   .then(resultat => resultat.json())
52   .then(meteoData => {
53     // Affiche les données météorologiques dans la console
54     console.log(meteoData);
55     // Appel de la fonction displayWeatherInfos pour afficher les données dans l'interface utilisateur
56     displayWeatherInfos(meteoData);
57   })
58   .catch(error => {
59     console.error("Erreur lors de la récupération des données météo : ", error);
60   });
61 }
62 }
63
64 // Fonction pour afficher les données météorologiques dans l'interface utilisateur
65 function displayWeatherInfos(data) {
66   const name = data.name;
67   const temperature = data.main.temp;
68   const conditions = data.weather[0].main;
69   const description = data.weather[0].description;
70
71   // Met à jour les éléments HTML avec les données météorologiques
72   document.querySelector('#ville').textContent = name;
73   document.querySelector('#temperature').textContent = Math.round(temperature);
74   document.querySelector('#conditions').textContent = capitalize(description);
75
76   // Met à jour l'icône météo en fonction des conditions météorologiques
77   document.querySelector('i.wi').className = weatherIcons[conditions];
78
79   // Met à jour la classe CSS du corps de la page en fonction des conditions météorologiques
80   document.body.className = conditions.toLowerCase();
81 }
82
83 // Récupère l'élément HTML de la ville
84 const villeInput = document.querySelector('#ville');
85
86 // Ajoute un gestionnaire d'événements pour permettre l'édition de la ville au clic
87 villeInput.addEventListener('click', () => {
88   villeInput.contentEditable = true;
89 });
90
91 // Ajoute un gestionnaire d'événements pour détecter la touche Entrée pour rechercher la météo
92 villeInput.addEventListener('keydown', (e) => {
93   if (e.keyCode === 13) {
94     e.preventDefault();
95     villeInput.contentEditable = false;
96     // Appel de la fonction main sans obtenir l'IP
97     main(false);
98   }
99 });
100
101 // Appel initial de la fonction main pour obtenir l'IP
102 main(true);
103

```

Dans cette application, j'utilise des fonctions asynchrones en conjonction avec la méthode "fetch" pour communiquer avec des API. Les fonctions asynchrones permettent à l'application de poursuivre son exécution tout en attendant la résolution des promesses renvoyées par les API. Pour accomplir cela, j'utilise le mot-clé "await" pour attendre la résolution des promesses avant de continuer l'exécution du script.



Ici je récupère bien la météo de Montpellier :



Ensuite, en ajoutant un gestionnaire d'évènement, et en utilisant l'attribut HTML "contentEditable" on peut rentrer le nom d'une autre ville pour aller chercher une autre météo :



## 2. Précisez les moyens utilisés :

Linux, VsCode, HTML, CSS, JavaScript

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ces deux projets mais une correction de l'exercice TripAdvisor a été effectuée en cours.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ecole O'Clock*

Chantier, atelier, service ► *Formation*

Période d'exercice ► Du : *27 mars 2023* au : *10 septembre 2023*

## 5. Informations complémentaires (facultatif)

## Activité-type 2

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

CP 5 ► Créer une base de données.

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

Pendant ma formation j'ai eu à concevoir un site qui ressemble quelque peu au site Allociné, une plateforme qui regroupe des films, avec l'affiche, le synopsis, les critiques, le casting ,etc .. Cette plateforme se nomme Oflix, les utilisateurs du site peuvent créer un compte, se faire une liste de films favoris et laisser des critiques.

Pour créer la base de données, j'ai utilisé la méthode **Merise** qui consiste à d'abord identifier les besoins du site et de concevoir ensuite le modèle conceptuel, le modèle logique et le modèle physique de données.

Pour cette plateforme j'ai d'abord définis les entités dont j'aurais besoin, qui seront par la suite les tables dans la base de données, ainsi que leurs propriétés : -

- Film (titre, durée, synopsis, sommaire, affiche, type)
- Genre (nom)
- Saison (numéro de saison, numéro d'épisode)
- Casting (rôle, ordre d'apparition)
- Personne (prénom, nom)
- Critique (pseudo, email, contenu, avis, réaction)
- Utilisateur (email, rôle, mot de passe)

Ensuite j'établie les **relations**, les **cardinalités**, les quantités minimum et maximum d'occurrences entre chaque table et ce dans les deux sens :

La relation entre film et casting : -

- Un film peut avoir au minimum 0 casting et au maximum plusieurs casting => 0,N
- Un casting, dès qu'il est créé, peut avoir au minimum et au maximum un film => 1,1

ici je suis sur une relation **Many to One**.

La relation entre film et critique :

- Un film peut avoir au minimum 0 critique et au maximum plusieurs critiques => 0,N
- Une critique, dès qu'elle est créée, peut avoir au minimum et au maximum un film => 1,1

Ici je suis sur une relation **One to Many**.

La relation entre film et saison :

- Un film peut avoir au minimum 0 saison et au maximum plusieurs saisons => 0,N
- Une saison, dès qu'elle est créée, peut avoir au minimum et au maximum un film => 1,1

Ici je suis également sur une relation **One to Many**.

La relation entre film et genre :

- Un film peut avoir au minimum 0 genre et au maximum plusieurs genres => 0,N
- Un genre peut avoir au minimum 0 film et au maximum plusieurs films => 0,N

Je suis ici sur une relation **Many to Many**.

La relation entre casting et personne :

- Un casting peut avoir au minimum une et au maximum une personne => 1,1
- Une personne peut appartenir à au minimum 0 et au maximum plusieurs casting => 0,N

Je suis ici sur une relation **Many to One**

La relation entre utilisateur et critique :

- Un utilisateur peut écrire au minimum 0 et au maximum plusieurs critiques => 0,N
- Une critique peut être écrite par au minimum et au maximum un utilisateur => 1,1

Je suis ici sur une relation **One to Many**

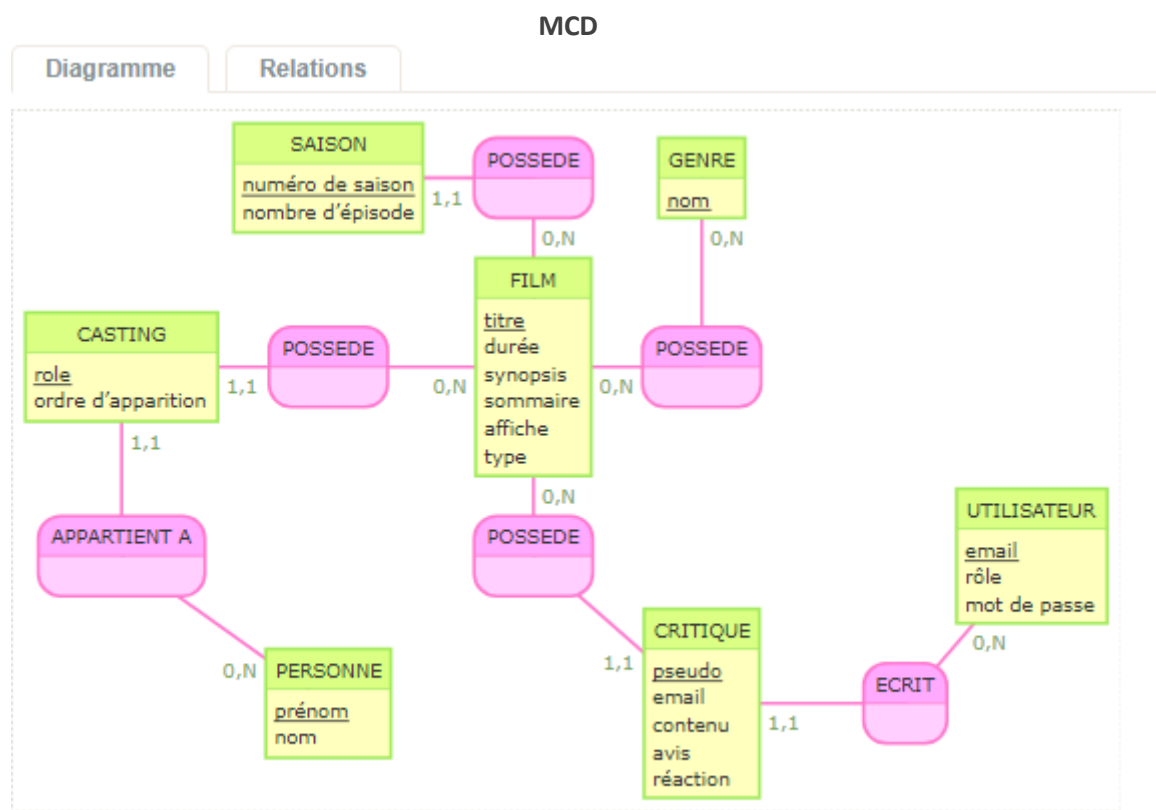
Une fois cet exercice terminé, je me suis servi de <https://www.mocodo.net> afin de réaliser le MCD

```
Films
:
SAISON : numéro de saison, nombre d'épisode
POSSEDE4, 0N FILM, 11 SAISON
GENRE : nom
:
:
:

CASTING : role, ordre d'apparition
POSSEDE2, 01 FILM, 11 CASTING
FILM : titre, durée, synopsis, sommaire, affiche, type
POSSEDE, 0N GENRE, 0N FILM
:
:
:

APPARTIENT A, 1N CASTING, 0N PERSONNE
:
POSSEDE3, 0N FILM, 11 CRITIQUE
:
:
:
UTILISATEUR : email, rôle, mot de passe
:
|
PERSONNE : prénom, nom
:
CRITIQUE : pseudo, email, contenu, avis, réaction
:
ECRIT, 0N UTILISATEUR, 11 CRITIQUE
```





## 2. Précisez les moyens utilisés :

J'ai utilisé l'application web Mocodo

## 3. Avec qui avez-vous travaillé ?

Ce projet est celui avec lequel j'ai fait ma spécialisation Symfony avec l'école. J'ai travaillé seul sur ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Ecole O'Clock*

Chantier, atelier, service ▶ *Formation*

Période d'exercice ▶ Du : *27 mars 2023* au : *11 septembre 2023*

## 5. Informations complémentaires (facultatif)

## Activité-type 2

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

CP 6 ► Développer les composants d'accès aux données.

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

Je continue avec le projet Oflix utilisé pour la CP 5.

Pour créer et communiquer avec la base de données en suivant la documentation de Symfony, j'ai installé l'ORM (Object Relational Mapping) de Symfony : **Doctrine**

Doctrine permet de communiquer avec la base de données en utilisant des objets et des requêtes orientées objet plutôt que des requêtes SQL directes.

Voici la documentation de Symfony pour installer Doctrine :

#### Installing Doctrine

First, install Doctrine support via the `orm` [Symfony pack](#), as well as the `MakerBundle`, which will help generate some code:

```
$ composer require symfony/orm-pack
$ composer require --dev symfony/maker-bundle
```

#### Configuring the Database

The database connection information is stored as an environment variable called `DATABASE_URL`. For development, you can find and customize this inside `.env`:

```
# .env (or override DATABASE_URL in .env.local to avoid committing your changes)

# customize this line!
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7"

# to use mariadb:
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=mariadb"

# to use sqlite:
# DATABASE_URL="sqlite:///kernel.project_dir/var/app.db"

# to use postgresql:
# DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=

# to use oracle:
# DATABASE_URL="oci8://db_user:db_password@127.0.0.1:1521/db_name"
```

Ensuite, après configuration du fichier **.env.local**, je crée la base de données via le terminal de commande avec la commande suivante : **php bin/console doctrine :database :create**

J'ai ensuite créé les entités, propriétés et relations également avec Doctrine via la commande suivant : **php bin/console make :entity**

Voici un exemple de code dans lequel on établit la relation ManyToMany entre Movie et Genre :

```
> student@teleporter:~/var/www/html/tp/symfo-oflix-journee-15-GrecoBen$ php bin/console make:entity

Class name of the entity to create or update (e.g. DeliciousPopsicle):
> Movie

Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):
> Genre

Field type (enter ? to see all types) [string]:
> relation

What class should this entity be related to?:
> Genre

What type of relationship is this?
-----
Type      Description
-----
ManyToOne Each Movie relates to (has) one Genre.
          Each Genre can relate to (can have) many Movie objects

OneToMany Each Movie can relate to (can have) many Genre objects.
          Each Genre relates to (has) one Movie

ManyToMany Each Movie can relate to (can have) many Genre objects.
          Each Genre can also relate to (can also have) many Movie objects

OneToOne   Each Movie relates to (has) exactly one Genre.
          Each Genre also relates to (has) exactly one Movie.
-----

Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:
> ManyToMany

Do you want to add a new property to Genre so that you can access/update Movie objects from it - e.g. $genre->getMovies()? (yes/no) [yes]:
> █
```

Ici en exemple, on trouve le code de l'entité Casting avec ses deux relations ManyToOne :

```
/**
 * @ORM\Entity(repositoryClass=CastingRepository::class)
 */
class Casting
{
    /**
     * @ORM\Id
     * @ORM\GeneratedValue
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $role;

    /**
     * @ORM\Column(type="integer")
     */
    private $creditOrder;

    /**
     * @ORM\ManyToOne(targetEntity=Movie::class, inversedBy="castings")
     * @ORM\JoinColumn(nullable=false)
     */
    private $movie;

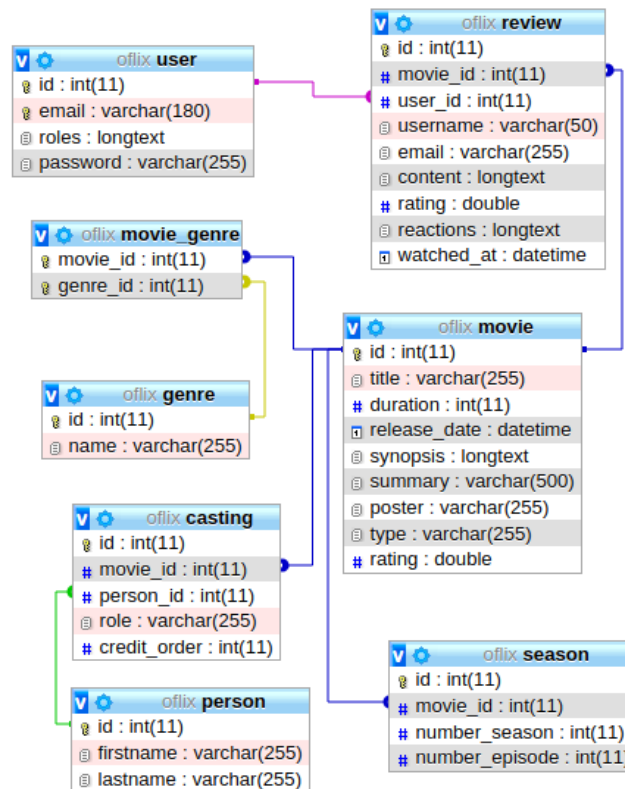
    /**
     * @ORM\ManyToOne(targetEntity=Person::class, inversedBy="castings")
     * @ORM\JoinColumn(nullable=false)
     */
    private $person;

    public function getId(): ?int
```

Doctrine propose un langage de requête appelé DQL, similaire au SQL, mais qui fonctionne avec des objets et des entités plutôt qu'avec des tables et des colonnes

Une fois la base de données créée, j'ai récupéré le MPD sur **phpmyadmin** :

MPD



Dans ce MPD, on constate la présence de la table movie\_genre, créée automatiquement par Doctrine en raison de la présence d'une relation ManyToMany dans les 2 sens entre Movie et Genre

## 2. Précisez les moyens utilisés :

Les technos utilisées pour ce projet sont **Linux**, **Symfony** (Php), **Doctrine** pour l'ORM (Object Relation Mapping), **MySQL** en SGBD et **PhpMyAdmin** en outil d'administration de base de données.



# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Toujours dans un contexte de formation, j'ai travaillé seul sur ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ecole O'Clock*

Chantier, atelier, service ► *Formation*

Période d'exercice ► Du : *27 mars 2023* au : *11 septembre 2023*

## 5. Informations complémentaires (facultatif)

Lien vers la page Doctrine : <https://symfony.com/doc/current/doctrine.html>

## Activité-type 2

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

CP 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 :

Toujours dans le projet Oflix, il était demandé de créer un back-office pour les admin avec un CRUD pour les films. Ce CRUD permet à l'admin connecté de créer, éditer, modifier ou supprimer un film.

Voici le MovieController du Backoffice qui gère le CRUD Movie. Dans ce CRUD j'ai défini le nom des routes, j'injecte la classe Request comme argument de la méthode (add, show, edit, delete) pour accéder aux paramètres des requêtes comme POST et GET et récupérer l'ID du film quand il est spécifié dans la route, ensuite je rends la vue à Twig pour l'affichage :

```
15  /**
16  * @isGranted("BACK_OFFICE_ACCESS", message="Rentres chez toi !")
17  */
18  class MovieController extends AbstractController
19  {
20      /**
21       * @Route("/admin", name="app_back_movie_list", methods={"GET"})
22       */
23       public function list(MovieRepository $movieRepository): Response
24       {
25
26           return $this->render('back/movie/list.html.twig', [
27               'movies' => $movieRepository->findAll(),
28           ]);
29       }
30
31       /**
32       * @Route("/admin/film/ajouter", name="app_back_movie_add", methods={"GET", "POST"})
33       */
34       public function add(Request $request, MovieRepository $movieRepository, MyMailerService $mailer): Response
35       {
36           $movie = new Movie();
37           $form = $this->createForm(MovieType::class, $movie);
38           $form->handleRequest($request);
39
40           if ($form->isSubmitted() && $form->isValid()) {
41               $movieRepository->add($movie, true);
42
43               $mailer->send(
44                   $movie->getTitle(),
45                   "emails/create_movie.html.twig",
46                   ["movie" => $movie],
47                   "greco.benjamin@orange.fr"
48               );
49
50               return $this->redirectToRoute('app_back_movie_list', [], Response::HTTP_SEE_OTHER);
51           }
52
53           return $this->renderForm('back/movie/new.html.twig', [
54               'movie' => $movie,
55               'form' => $form,
56           ]);
57       }
58   }
```

```
59  /**
60  * @Route("/admin/film/{id}", name="app_back_movie_show", methods={"GET"})
61  */
62  public function show(Movie $movie): Response
63  {
64      return $this->render('back/movie/show.html.twig', [
65          'movie' => $movie,
66      ]);
67  }
68
69  /**
70  * @Route("/admin/film/modifier/{id}", name="app_back_movie_edit", methods={"GET", "POST"})
71  */
72  public function edit(Request $request, Movie $movie, MovieRepository $movieRepository): Response
73  {
74      $form = $this->createForm(MovieType::class, $movie);
75      $form->handleRequest($request);
76
77      if ($form->isSubmitted() && $form->isValid()) {
78          $movieRepository->add($movie, true);
79
80          return $this->redirectToRoute('app_back_movie_list', [], Response::HTTP_SEE_OTHER);
81      }
82
83      return $this->renderForm('back/movie/edit.html.twig', [
84          'movie' => $movie,
85          'form' => $form,
86      ]);
87  }
88
89  /**
90  * @Route("/admin/film/supprimer/{id}", name="app_back_movie_delete", methods={"POST"})
91  */
92  public function delete(Request $request, Movie $movie, MovieRepository $movieRepository): Response
93  {
94      if ($this->isCsrfTokenValid('delete' . $movie->getId(), $request->request->get('_token'))) {
95          $movieRepository->remove($movie, true);
96      }
97
98      return $this->redirectToRoute('app_back_movie_list', [], Response::HTTP_SEE_OTHER);
99  }
```

Dans la fonction « delete » j'ai rajouter une sécurité avec la méthode native de Symfony « isCsrfTokenValid ». Cela consiste à insérer un token CSRF dans un «input hidden » dans le html du formulaire qui est renvoyé dans l'objet Request à la soumission afin d'éviter de supprimer des données depuis l'url du navigateur.

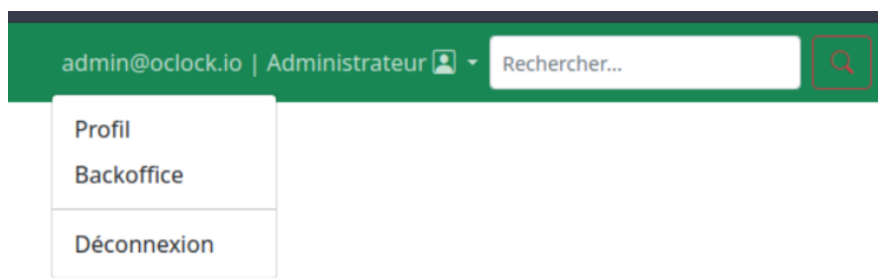
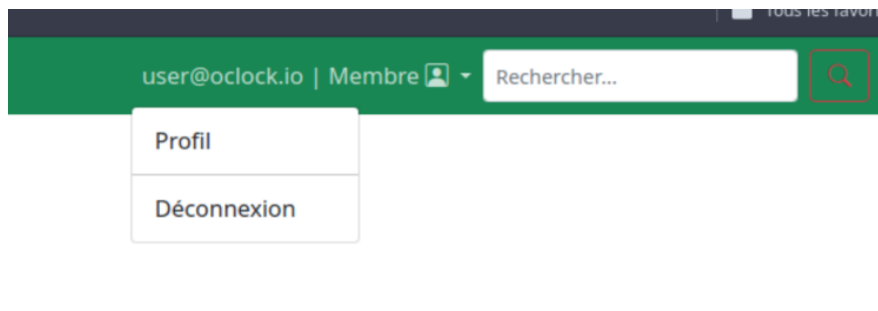
Après l'installation du bundle de sécurité « security-bundle » de Symfony, j'ai configuré le Access Control pour définir les rôles User, Manager et Admin et sécuriser les routes du Backoffice qui commence par /admin dans le fichier de configuration security.yaml. Les routes commençant par /admin ne sont accessible que par les admins désormais.

```

59 # Easy way to control access for large sections of your site
60 # Note: Only the *first* access control that matches will be used
61 access_control:
62 # It is important here to start from the most specific to the most general because once a regex is validated, the script stops
63
64 # * Here, the route to add a review is restricted to non-users (anonymous)
65 - { path: ^/critique/ajouter/\d+, roles: ROLE_USER }
66
67 # * Here, I put the admin role first because the restrictions are more specific
68 # * we will restrict access to creation
69 - { path: ^/admin.*ajouter, roles: ROLE_ADMIN }
70 # * Here, I put the admin role first because the restrictions are more specific
71 # * we will restrict access to editing
72 - { path: ^/admin.*modifier/\d+, roles: ROLE_ADMIN }
73
74 # * All routes in the back-office ending with an id and being in POST will be blocked unless you are an admin (deletion management)
75 - { path: ^/admin.*\/\d+, roles: ROLE_ADMIN, methods: POST }
76
77 # * All the rest of the back office is accessible only to managers (or admins thanks to the hierarchy)
78 - { path: ^/admin, roles: ROLE_MANAGER }
79 # To save lines in our ACL and make the application more logical, we hierarchize roles
80 role_hierarchy:
81 # Admin has the rights of the manager
82 | ROLE_ADMIN: ROLE_MANAGER
83 # The manager has the rights of the user (by default, all users are set as 'user' during creation in Symfony)
84 | ROLE_MANAGER: ROLE_USER

```

Pour accéder au back-office, il se suffit de se connecter sur le login général de la plateforme avec les droits admin, un onglet Backoffice apparaît alors dans le menu déroulant admin.



Cette fonctionnalité est assurée par une simple vérification d'utilisateur dans la NavBar avec la fonction « if\_is\_granted » :



# DOSSIER PROFESSIONNEL (DP)

```

57     {% if is_granted("IS_AUTHENTICATED_FULLY") %}
58     <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
59       <li class="nav-item dropdown">
60         <a
61           class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown">
62           <!-- Bootstrap CSS -->
63           <span>{{app.user.email}} |
64           {% if "ROLE_ADMIN" in app.user.roles %}
65             Administrateur
66           {% elseif "ROLE_MANAGER" in app.user.roles %}
67             Manager
68           {% else %}
69             Membre
70           {% endif %}
71           </span>
72           <i class="bi bi-person-square"></i>
73         </a>
74         <ul class="dropdown-menu">
75           <li>
76             <a class="dropdown-item" href="#">Profil</a>
77           </li>
78           {% if is_granted("ROLE_MANAGER") %}
79             {% if is_granted("BACK_OFFICE_ACCESS") %}
80               <li>
81                 <a class="dropdown-item" href="{{path('app_back_movie_list')}}">Backoffice</a>
82               </li>
83             {% else %}
84               <li>
85                 <span class="dropdown-item">Rentres chez toi !</span>
86               </li>
87             {% endif %}
88           {% endif %}
89           <li>
90             <hr class="dropdown-divider">
91           </li>
92           <li>
93             <a class="dropdown-item" href="{{path('app_security_logout')}}">Déconnexion</a>
94           </li>
95         </ul>
96       </li>
97     </ul>
98   {% else %}
99     <a href="{{path('app_security_login')}}" class="btn btn-danger me-2">Connexion</a>
100   {% endif %}
101
102
  
```

Voici le formulaire de création d'un film créé avec le FormBuilder de Symfony :

```
namespace App\Form;

use App\Entity\Genre;
use App\Entity\Movie;
use Symfony\Bridge\Doctrine\Form\Type\EntityType;
use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\Extension\Core\Type\ChoiceType;
use Symfony\Component\Form\Extension\Core\Type\DateType;
use Symfony\Component\Form\Extension\Core\Type\IntegerType;
use Symfony\Component\Form\Extension\Core\Type\TextareaType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\Form\Extension\Core\Type\UrlType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolver;

class MovieType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options): void
    {
        $builder
            ->add('title', TextType::class, [
                "label" => "Titre du film"
            ])
            ->add('duration', IntegerType::class, [
                "label" => "Durée du film",
                "help" => "En minutes"
            ])
            ->add('synopsis', TextareaType::class, [
                "label" => "Synopsis"
            ])
            ->add('summary', TextareaType::class, [
                "label" => "Résumé"
            ])
            ->add('poster', UrlType::class, [
                "label" => "Image",
                "help" => "Url absolu"
            ])
            ->add('type', ChoiceType::class, [
                "label" => "Type de média",
                "choices" => [
                    "Film" => "film",
                    "Série" => "série"
                ]
            ])
            ->add('genres', EntityType::class, [
                "class" => Genre::class,
                "label" => "Genres",
                "expanded" => true,
                "multiple" => true,
                "choice_label" => "name"
            ])
            ->add('releaseDate', DateType::class, [
                "label" => "Date de sortie",
                "input" => "datetime_immutable",
                "widget" => "single_text",
                "row_attr" => ['class' => 'col-2']
            ])
        ;
    }
}
```



# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Les technos utilisées pour ce projet sont **Linux**, **Symfony** (Php), **Doctrine** pour l'ORM (Object Relation Mapping), **MySQL** en SGBD, **PhpMyAdmin** et **Twig**

## 3. Avec qui avez-vous travaillé ?

Toujours dans un contexte de formation, j'ai travaillé seul sur ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ecole O'Clock*

Chantier, atelier, service ► *Formation*

Période d'exercice ► Du : *27 mars 2023* au : *11 septembre 2023*

## 5. Informations complémentaires (facultatif)

<https://symfony.com/doc/current/security.html>

<https://symfony.com/doc/current/forms.html>

<https://symfony.com/doc/current/templates.html>

## 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 Benjamin Greco,  
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je  
suis l'auteur des réalisations jointes.

Fait à Montpellier  
pour faire valoir ce que de droit.

le 05/10/2023

Signature :

## Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.



# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

*(Si le RC le prévoit)*