

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ JACOBIN  
*Nom d'usage* ▶  
*Prénom* ▶ GWLADYS  
*Adresse* ▶ 6 rue André Messenger Hall 5 – 75018 Paris

## Titre professionnel visé

RNCP 37674 Développeur web / 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 :

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

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

### Ce dossier comporte :

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

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

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

## Sommaire

### Exemples de pratique professionnelle

#### Intitulé de l'activité-type n° 1

Développer la partie front-end d'une application web ou web mobile sécurisée

p. 5

► Installer et configurer son environnement de travail en fonction du projet web ou web mobile

p.

► Réaliser des interfaces utilisateur statiques web ou web mobile

► Développer la partie dynamique des interfaces utilisateur web ou web mobile

p.

#### Intitulé de l'activité-type n° 2

Développer la partie back-end d'une application web ou web mobile sécurisée

p.

► Mettre en place une base de données relationnelle

p.

► Développer des composants d'accès aux données SQL et NoSQL

p.

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

p.

**Déclaration sur l'honneur**

p.

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

p.

**Annexes** *(Si le RC le prévoit)*

p.

---

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type

Développer la partie front-end d'une application web ou web mobile sécurisée

**Exemple n°1** ► Installer et configurer son environnement de travail en fonction du projet web ou web mobile

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

Au début de notre chantier, nous avons installé et configuré notre poste de travail. Cela nous a permis d'avoir un environnement complet et prêt à l'emploi pour suivre la formation. Cette étape était importante car elle nous a aidés à découvrir et à utiliser les différents outils dont nous aurons besoin pendant notre apprentissage.



J'ai installé Local **WP**, ce qui m'a permis de créer et de gérer plusieurs sites Web **WordPress** sur mon ordinateur, sans avoir besoin de me connecter à Internet.

J'ai pu créer plusieurs sites Web **WordPress** sur mon ordinateur et basculer facilement entre eux.

En plus des fonctionnalités de base, Local **WP** inclut également des extensions utiles telles que des outils SEO, des formulaires, des palettes de couleurs, des thèmes, ect...

J'ai également créé un blog de Promo sur le thème de la musique appelé "**Quatorze**" *Influences Musicales*.

Les apprenants de la formation ont contribué à l'alimentation du blog avec des articles riches en informations, des découvertes, des sorties et des playlists. Nous avons également cotisé pour pouvoir héberger le blog sur "**Hostinger**".

Le liens de notre blog de Promo : <https://quatorze.org/>



J'ai installé **Visual Studio Code** (VS Code) comme éditeur de code (IDE) principal pour améliorer ma productivité.

---

**VS Code** est un éditeur open source léger qui prend en charge un large éventail de langages et de frameworks de programmation.

J'aime beaucoup son interface intuitive et ses fonctionnalités puissantes, telles que l'intégration Git intégrée, le débogage et la coloration syntaxique.

De plus, la vaste bibliothèque d'extensions de VS Code me permet de personnaliser mon éditeur pour l'adapter à mes besoins spécifiques, comme l'ajout d'extraits de code, de linters et d'outils d'auto-complétion.

Exemples d'extensions :

- ◆ **HTML Class Suggestions** - Suggère des classes CSS lorsque vous tapez dans les balises HTML pour améliorer votre flux de travail de développement.
- ◆ **CSS Flexbox Cheatsheet** - Fournit un aperçu rapide des propriétés et valeurs Flexbox pour vous aider à créer des mises en page CSS plus facilement.
- ◆ **Indent-Rainbow** - Affiche les niveaux d'indentation dans des couleurs différentes pour améliorer la lisibilité du code.
- ◆ **Material Icon Theme** - Un ensemble d'icônes matérielles pour les fichiers et les dossiers dans Visual Studio Code.
- ◆ **IntelliSense for CSS class names in HTML** - Fournit des suggestions de noms de classes CSS lorsque vous tapez dans les balises HTML.
- ◆ **Longsight Light Theme** - Un thème clair et lisible pour Visual Studio Code.
- ◆ **One Dark Pro** - Un thème sombre et élégant pour Visual Studio Code.
- ◆ **PHP DocBlocker** - Génère automatiquement des commentaires DocBlock pour les fonctions et les méthodes PHP.
- ◆ **PHP getters and setters for Visual Studio Code** - Génère automatiquement des getters et des setters pour les propriétés d'une classe PHP.
- ◆ **PHP code intelligence for Visual Studio Code** - Fournit une analyse de code et des suggestions de complétion pour le langage PHP.
- ◆ **PHP Namespace Resolver** - Fournit une résolution automatique des espaces de noms pour les fichiers PHP.
- ◆ **MongoDB for VS Code** - Une extension pour connecter Visual Studio Code à une instance MongoDB et exécuter des commandes MongoDB directement dans l'éditeur.
- ◆ **Zoom Bar README** - Ajoute une barre de zoom à Visual Studio Code pour un contrôle plus facile de la taille du texte.

Grâce à **VS Code**, j'ai pu considérablement améliorer mon expérience de développement, ce qui me permet d'écrire facilement du code plus propre et plus efficace.



J'ai installé **GitHub**, une plateforme en ligne , pour héberger et partager mon code source lorsque je travaille sur des projets de programmation avec Visual Studio Code. Cette intégration m'a apporté de nombreux avantages :

- ◆ Garder une trace de l'historique de mon code
- ◆ Suivre les modifications apportées et revenir à des versions antérieures si nécessaire
- ◆ Mon code est stocké en ligne sur les serveurs de **GitHub**, je peux y accéder depuis n'importe quel ordinateur et de synchroniser mes modifications
- ◆ Éviter de perdre mon travail en cas de problème local
- ◆ Partager mon code avec d'autres développeurs, travailler ensemble sur un même projet

J'ai également un compte github : **Gwladys75**, dans lequel j'ai pu y stocker mes cours et mes projets au cours de la formation

**Les principales commandes Git que j'ai utilisées pour créer un nouveau dépôt "new repository" GitHub depuis Visual Studio Code :**

## ...or create a new repository on the command line

```
echo "# eval-front" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Gwladys75/eval-front.git
git push -u origin main
```

**Les principales commandes Git que j'ai utilisées pour pousser « pusher » un dépôt existant depuis la ligne de commande de Visual Studio Code :**

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/Gwladys75/eval-front.git
git branch -M main
git push -u origin main
```



Chaque fois qu'un fichier est modifié, Git enregistre ces modifications dans un "commit". Un commit contient un message expliquant le changement, qui l'a fait et à quel moment.

Grâce à Git, les développeurs peuvent facilement revenir en arrière à une ancienne version d'un fichier. Ils peuvent aussi combiner différentes versions modifiées en parallèle.

Git est un système de gestion de versions décentralisé qui permet de suivre les modifications apportées à des fichiers au fil du temps.

Les principales fonctionnalités de Git :

- ◆ La création de "commits" qui enregistrent des changements avec un message descriptif
- ◆ La possibilité de revenir à des versions antérieures des fichiers
- ◆ La gestion de branches parallèles de développement





J'ai installé **Laragon** Full, un environnement de développement complet pour les développeurs web.

Le logiciel contient les dernières versions de PHP(8.3.4), Apache(2.4.54) et MySQL(8.0.30)

- ◆ PHP le langage de programmation côté serveur, me permet de créer des sites web dynamiques et interactifs
- ◆ Apache le serveur web open source, me permet d'héberger et de servir ces sites web localement
- ◆ MySQL le système de gestion de bases de données relationnelles, il me donne la possibilité de stocker et de manipuler efficacement les données

Avec **Laragon**, j'ai tout ce qu'il faut pour développer des sites web facilement



J'ai installé MongoDB, une base de données qui stocke les informations sous forme de documents.

Ces documents ressemblent à des objets JavaScript avec des paires clé-valeur. Au lieu d'avoir des tables comme dans les bases de données traditionnelles, MongoDB utilise des collections pour regrouper les documents similaires.

Les principaux éléments de **MongoDB** sont :

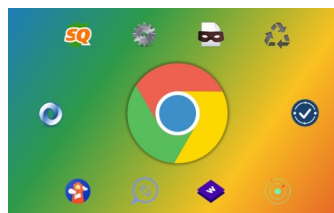
- ◆ **Documents** : Ce sont les données elles-mêmes, stockées au format JSON.

- ◆ **Collections** : Un groupe de documents du même type, comme une table dans une base de données classique.
- ◆ **Base de données** : Un regroupement de collections liées.



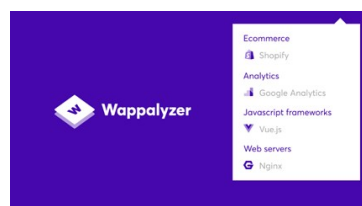
J'ai également installé **mongosh**, qui est un outil en ligne de commande pour interagir avec **MongoDB**. Avec **mongosh**, j'ai pu, modifier ou supprimer des données dans les bases de données **MongoDB** en tapant des commandes.

Exemple : `db.users.deleteOne({name: "John"})` => supprimera le premier **document** de la collection **"users"** où le champ **"name"** est égal à **"John"**.



Les extensions **Google Chrome** peuvent aider les développeurs Web à gagner du temps, à améliorer leur flux de travail et à fournir des fonctionnalités supplémentaires.

J'ai installé les extensions suivantes pour faciliter mon travail de développement web :

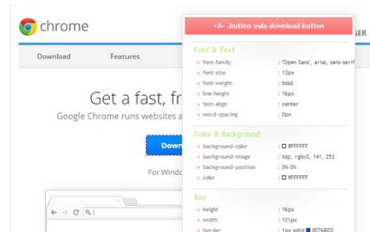


**Wappalyzer** permet d'identifier les différentes technologies utilisées sur les sites web, y compris les CMS, les frameworks de développement, les outils d'analyses et bien d'autres.

## DOSSIER PROFESSIONNEL (DP)



**GoFullPage** est particulièrement utile pour créer des « prototypes et wireframes », car il vous permet de capturer des captures d'écran pleine page de la fenêtre actuelle du navigateur. Cette extension utilise une technologie de capture d'écran avancée pour gérer des pages complexes, y compris des éléments de défilement internes et des iframes intégrés.



**CSS Viewer** affiche les propriétés CSS d'un élément de page Web lorsque vous le survolez, ce qui facilite l'inspection et l'identification des propriétés CSS clés.



**Font Ninja** aide les concepteurs et les développeurs à identifier et à essayer les polices de n'importe quel site Web. Il permet une reconnaissance précise des polices. En survolant le texte, vous pouvez voir le nom de la police et les propriétés CSS

**42. Précisez les moyens utilisés :**

Documentations, tutoriels , Google Chrome

### 3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur, un formateur externe, et seule

#### 4. Contexte

Nom de l'entreprise, organisme ou association **Colombbus**

Chantier, atelier, service **chantier d'insertion développeur web & web mobile.**

Période d'exercice Du : **1** au : **16/08/2024**

#### 5. Informations complémentaires (facultatif)

**Activité-type** 1 Développer la partie front-end d'une application web ou web

mobile sécurisée

**Exemple n° 2** ► Réaliser des interfaces utilisateur statiques web ou web mobile

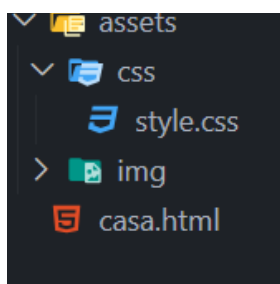
## 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Lors de la formation, j'ai travaillé en groupe sur une maquette (TP) appelée "**La Casa**"  
Via ce TP, j'ai appris à découper, maquetter mon code et apprendre à utiliser Bootstrap via les liens CDN (fichiers CSS et JavaScript externes) et les media queries.

Avant tout nous avons créer l'arborescence des dossiers comme ceci :

**assets** => contenant le **dossier css** (qui contient le fichier **style.css**(cascade feuille de style)), et le **dossier img** (qui contient les **images**)

A la racine nous avons créer le fichier **casa.html** qui contiendra tout notre **html** et nos liens css, bootstrap...



Puis nous avons réaliser le HTML (HTML 5) et le CSS (CSS 3) pour cette maquette

Pour le HTML : (Appeler arbre HTML)

Nous avons appris le découpage du code qui se distingue en plusieurs sections :

### 1. Le **DOCTYPE** ! :

Le doctype est une instruction qui indique au navigateur la version du langage HTML utilisée dans le document. C'est la première ligne d'un document HTML

### 2. La balise **<head>** contient les métadonnées du document, telles que le titre de la page, les

feuilles de style CSS, les liens bootstrap et d'autres informations non visibles pour l'utilisateur

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Ephesis&display=swap" rel="stylesheet">
  <!-- les icones depuis bootstrap -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
  <!-- le lien css -->
  <link rel="stylesheet" href="assets/css/style.css">
  <title>La casa - Accueil</title>
</head>
```

3. balise **<header>** peut contenir des éléments de titre, un logo, la barre de navigation , qui se trouve dans la balise **<body>** => La balise définit le corps du document HTML = le contenu visible de la page web.

```
<body>
  <header>
    <div id="navigation">
      <h1>La Casa</h1>
      <nav>
        <a href="#">Buy</a>
        <a href="#">Rent</a>
        <a href="#">Sell</a>
        <a href="#">About</a>
        <a id="contact" href="#">Contact</a>
      </nav>
    </div>
    <div id="presentation">
      <h2>Find your dream home</h2>
      <p>apartments - Houses - Mansions</p>
    </div>
  </header>
```

4. La balise **<main>** représente le contenu principal unique du document. Contient la plupart du contenu visible pour l'utilisateur. Il ne doit y avoir qu'un seul élément **<main>** par document.

5. Les balises sémantiques comme `<article>`, `<section>` permettent de structurer le contenu de manière logique et sémantique, facilitant le référencement.

```
<main class="container">
  <section class="row">
    <article class="col-4">
      <div class="annonce">
        
        <div class="prix">
          1000 €
        </div>
        <div class="texte">
          <h2>Lorem ipsum dolar sit</h2>
          <p> 2 kitchens, 2 beds, 2 bath...(288ftsq)</p>
        </div>
      </div>
    </article>
```

6. La balise `<footer>` représente un pied de page ou des informations de fin pour le contenu. Il contient des informations diverses ; comme des liens pour les reseaux sociaux (leurs logo) le copyright, etc

```
<footer>
  <div class="container">
    <div class="row">
      <div class="col-2">
        <p><a href="#">About</a></p>
        <p><a href="#">Support</a></p>
        <p><a href="#">Contact</a></p>
        <p><a href="#">Terms</a></p>
        <p><a href="#">Policy</a></p>
      </div>
      <br>
      <p>
        <a href="#"><i class="bi bi-facebook"></i></a>
        <a href="#"><i class="bi bi-twitter"></i></a>
        <a href="#"><i class="bi bi-google"></i></a>
        <a href="#"><i class="bi bi-skype"></i></a>
      </p>
    </div>
  </div>
  <hr>
  <p class="copyright">
    Copyright &copy; Colombbus - all rights reserved.
  </p>
</div>
</footer>
</body>
```

Nous avons aussi utilisé les **media queries**, ce qui nous a permis de changer le design de la maquette pour différentes tailles d'écran.

J'ai défini plusieurs points d'arrêt (**breakpoints**) pour cibler quelques formats spécifiques :

- ◆ Petit format (s) : Pour les écrans de smartphones, avec une largeur maximale de 767px.

```
@media screen and (max-width:767px) {  
  .col-4 {  
    width: 100%;  
  }  
  
  #navigation {  
    flex-direction: column;  
    height: 130px;  
  }  
  
  h1 {  
    margin: 0 auto;  
  }  
  
  .col-2 {  
    width: 100%;  
    text-align: center;  
  }  
  
  .col-6 {  
    width: 100%;  
  }  
}
```



# DOSSIER PROFESSIONNEL (DP)

- ◆ **Format moyen (md)** : Pour les tablettes et petits ordinateurs portables, avec une largeur comprise entre 768px et 992px.

```
/* *****Media queries***** */
@media screen and (max-width:991px) and (min-width:768px) {
    .col-4 {
        width: 50%;
    }

    #navigation {
        flex-direction: column;
        height: 130px;
    }

    h1 {
        margin: 0 auto;
    }

    .col-2 {
        width: 30%;
    }
}
```

## 2. Précisez les moyens utilisés :

Visual Studio Code, HTML5, CSS 3, Bootstrap 5, W3C Validator, W3SCHOOL, MDN

## 3. Avec qui avez-vous travaillé ?

Avec le formateur et seule

#### 4. Contexte

Nom de l'entreprise, organisme ou association *colombbus*

Chantier, atelier, service *chantier d'insertion développeur web & web mobile.*

Période d'exercice Du : au :

#### 5. Informations complémentaires *(facultatif)*

## Activité-type 1

Développer la partie front-end d'une application web ou web mobile sécurisée

**Exemple n° 4** ► Développer la partie dynamique des interfaces utilisateur web ou web mobile

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

Dans le cadre de ma formation, j'ai passé une évaluation Front-end, dans laquelle j'ai réalisé une maquette qui a pour nom :

**ONEMILLIONLINES\_**

Afin de réaliser une partie dynamique du site, dans le menu "WHO WE ARE", j'ai réalisé une partie section : `<section class="TEAM">` (voir ANNEXE 2)

La classe "team" est utilisée pour cibler cette section avec des styles CSS et le JAVASCRIPT

Afin de pouvoir créer différents profils :

```
<section class="team ">
  <h3 class="center pt-5">OUR TEAM</h3>

  <hr class="hr">

  <div class="container transition text-center">
    <div class="row">
      <div class="col-lg-3 col-md-6 mx-auto profil" id="profil1">
        <div class="infos blanc white">
          <h4>Joanne Livy</h4>
          <p>CEO</p>
        </div>
      </div>

      <div class="col-lg-3 col-md-6 mx-auto profil" id="profil2">
        <div class="infos blanc white">
          <h4>Gaston Parsley</h4>
          <p>CPO</p>
        </div>
      </div>
    </div>
  </div>
```

```

<div class="col-lg-3 col-md-6 mx-auto profil" id="profil3">
  <div class="infos blanc white">
    <h4>Manuel Huntington</h4>
    <p>Communication</p>
  </div>
</div>

<div class="col-lg-3 mx-auto profil" id="profil4">
  <div class="infos blanc white">
    <h4>Tracy Bellamie</h4>
    <p>Educator</p>
  </div>
</div>
</div>

```

Les **class de bootstrap** “**col-lg-3** et **col-md-6**” définissent la largeur de la colonne en fonction de la taille de l'écran (large ou moyen).

**col-lg-3** => affichera les 4 photos

**col-md-6** => affichera 2 photos

La **class** “**mx-auto**” centre horizontalement la colonne.

À l'intérieur, il y a une div avec les class : **infos, blanc et white** qui contient le nom et le titre du membre de l'équipe.

L'identifiant **profil1**, les class : **infos, blanc et white** sont utilisés pour être cibler avec du CSS et du JavaScript,

c'est donc avec ces différents éléments que j'ai dynamisé cette **section “team”**

Les trois autres blocs de code suivants (**profil2, profil3 et profil4**) ont une structure similaire et représentent les autres membres de l'équipe.

## Le CSS de la section 'team' :

```
.team {  
  background-color: rgba(128, 128, 128, 0.071);  
  text-align: center;  
  margin-top: 20px;  
}  
  
h4 {  
  font-family: "Roboto Slab", serif;  
  color: var(--rouge);  
}  
  
.infos {  
  padding: 0;  
  margin: 0;  
  height: 400px;  
  display: flex;  
  justify-content: center;  
  flex-direction: column;  
}
```

```
.blanc {  
  background-color: var(--blanc);  
}  
  
.white {  
  color: var(--blanc);  
}  
  
img {  
  max-width: 100%;  
}
```

## Le JavaScript pour dynamiser de la section :

```
// Page contact

let infos = document.querySelectorAll('.infos');
console.log(infos);

infos.forEach(profil => {

  // au survole de la souris sur profil
  profil.addEventListener('mouseover', () => {

    profil.classList.remove('blanc');
    profil.classList.add('white');

  })

  // quand la souris quitte profil
  profil.addEventListener('mouseout', () => {

    profil.classList.add('blanc');
    profil.classList.remove('white');

  })

});
```

En résumé, ce code JavaScript sélectionne tous les éléments HTML avec la **class infos**, puis ajoute des écouteurs d'événements **mouseover** et **mouseout** à chacun d'eux.

Lorsque la souris **survole** un élément, son style est modifié en **supprimant** la **class blanc** et en **ajoutant** la **class white**. Lorsque la souris **quitte** l'élément, son style **est restauré** en **ajoutant** la **class blanc** et en **supprimant** la **class white**.

# DOSSIER PROFESSIONNEL (DP)

1. La première ligne **let infos = document.querySelectorAll('.infos');**  
Sélectionne tous les éléments HTML avec la class **"infos"** et les stocke dans une variable appelée : **infos => let info ;**  
La méthode **querySelectorAll** est utilisée pour sélectionner plusieurs éléments correspondant au sélecteur CSS **.infos**
2. La **console.log(infos)** est utilisée à des fins de débogage. Il affiche la variable **infos** sur la console du navigateur qui affiche les éléments sélectionnés.
3. La boucle **forEach** est utilisée pour parcourir chaque élément du tableau **infos** et chaque élément est stocké dans une variable appelée **profil**.  
Le code entre les accolades { } sera exécuté.
4. L'écouteur d'événement **addEventListener** ici appelé **mouseover** est ajouté à chaque élément **profil**  
Lorsque la souris **survol l'élément**, la fonction de rappel est exécutée.
5. Elle **supprime** la class **blanc** et ajoute la class **white** à l'élément **profil**.

```
profil.classList.remove('blanc');  
profil.classList.add('white');
```

Cela permet de **changer le style de l'élément** lorsque la souris **le survole**.

6. L'écouteur d'événement **addEventListener** ici appelé **mouseout** est ajouté à chaque élément **profil**.  
Lorsque la souris **quitte l'élément**, la fonction de rappel est exécutée. Elle **ajoute** la class **blanc** et **supprime** la class **white** de l'élément **profil**.

```
profil.classList.add('blanc');  
profil.classList.remove('white');
```

Cela permet de **restaurer le style initial de l'élément** lorsque la souris **le quitte**.

## 2. Précisez les moyens utilisés :

Visual Studio code, HTML5 , CSS 3, W3C Validator, Bootstrap 5, W3SCHOOL, MDN

### 3. Avec qui avez-vous travaillé ?

J'ai travaillé seule

### 4. Contexte

Nom de l'entreprise, organisme ou association

COLOMBBUS

Chantier, atelier, service



Cliquez ici pour taper du texte.

Période d'exercice



Du :

au :

### 5. Informations complémentaires (facultatif)



## Activité- type

2

Développer la partie back-end d'une application web ou web mobile sécurisée

**Exemple n° 1** ► Mettre en place une base de données relationnelle

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

Lors de notre formation nous avons créé un site web appelé "Cinéma". (**Annexe 3**)

Pour ce projet, nous avons utilisé le langage de programmation **PHP** et le langage de requête **SQL**

Ce projet pratique nous a permis d'apprendre les bases du PHP et des requêtes SQL notamment la connexion à une base de données et les différentes syntaxes pour interagir avec celle-ci.

```
<!-- Fichier qui contient les fonctions php à utiliser dans notre site -->
<?php

session_start();

define("RACINE_SITE", "/cinema4_php/");
```

Ce code contient plusieurs éléments importants pour une application web en PHP tels que la gestion des sessions, la définition de constantes pour les chemins d'accès, et une fonction de débogage pour faciliter le développement. :

- ◆ La balise **<?php** ouvre un bloc de code PHP à exécuter par le serveur.
- ◆ **session\_start();** démarre une nouvelle session, permettant de stocker des données côté serveur.
- ◆ **define("RACINE\_SITE", "/cinema4\_php/");** définit une constante **RACINE\_SITE** contenant le chemin racine du site web "cinema4\_php".  
Cette constante sera utilisée pour construire des liens et chemins d'accès aux fichiers.

```
function debug($var)
{
    echo '<pre class="border border-dark bg-light text-primary w-50 p-3">';
    var_dump($var);
    echo '</pre>';
}
```

Sur notre site web, nous avons plusieurs tableaux de données importants comme ceux pour la connexion et l'authentification des utilisateurs. (via des formulaires)

Afin de faciliter le débogage et l'affichage formaté de ces données durant l'avancement de notre code, nous avons créé une fonction appelée **debug(\$var)** => Cette fonction évite la redondance de code et rend le débogage plus pratique et efficace lors du développement de notre site.

Cette fonction enveloppe une fonction **var\_dump(\$var)** dans une balise HTML **<pre>**, ce qui permet d'afficher le contenu de la variable **\$var** de manière lisible, y compris son type et sa valeur avec une mise en forme respectant la structure des données.

Ainsi, au lieu de répéter **echo '<pre>'; var\_dump(\$var); echo '</pre>';** à plusieurs endroits de notre code, nous appelons simplement la fonction **debug(\$var)**.

```
// On déclare des constantes d'environnement qui vont contenir les informations à la connexion à la BDD

// Constante du serveur => localhost
define("DBHOST", "localhost");

// Constante de l'utilisateur de la BDD du serveur en local => root
define("DBUSER", "root");

// Constante pour le mot de passe de serveur en local => pas de mot de passe
define("DBPASS", "");

// Constante pour le nom de la BDD
define("DBNAME", "cinema");
```

**DBHOST:** Cette constante est définie sur la valeur "localhost", qui est le nom d'hôte du serveur de base de données. Dans ce cas, le serveur de base de données s'exécute sur la même machine que le script PHP, c'est pourquoi nous utilisons "localhost".

1. **DBUSER:** Cette constante est définie sur la valeur "root", qui est le nom d'utilisateur utilisé pour se connecter au serveur de base de données. Ce nom d'utilisateur dispose des privilèges pour accéder et modifier la base de données.
2. **DBPASS:** Cette constante est définie sur une chaîne vide "", ce qui signifie qu'aucun mot de passe n'est requis pour se connecter au serveur de base de données. Ceci n'est pas recommandé pour les environnements de production, mais il est courant de ne pas avoir de mot de passe dans un environnement de développement local.

3. DBNAME: Cette constante est définie sur la valeur "cinema", qui est le nom de la base de données à laquelle le script PHP accédera.

En définissant ces constantes, le script PHP peut facilement se connecter à la base de données MySQL à l'aide des fonctions `mysqli` ou `PDO` sans avoir à coder en dur les informations d'identification de la base de données directement dans le script. Cela facilite la gestion et la mise à jour des informations d'identification de la base de données à l'avenir.

Dans l'ensemble, ce code est un moyen simple et efficace de définir des constantes pour une connexion à une base de données MySQL en PHP.

Afin de stocker et gérer les données du site, nous avons mis en place une base de données avec l'outil phpMyAdmin.

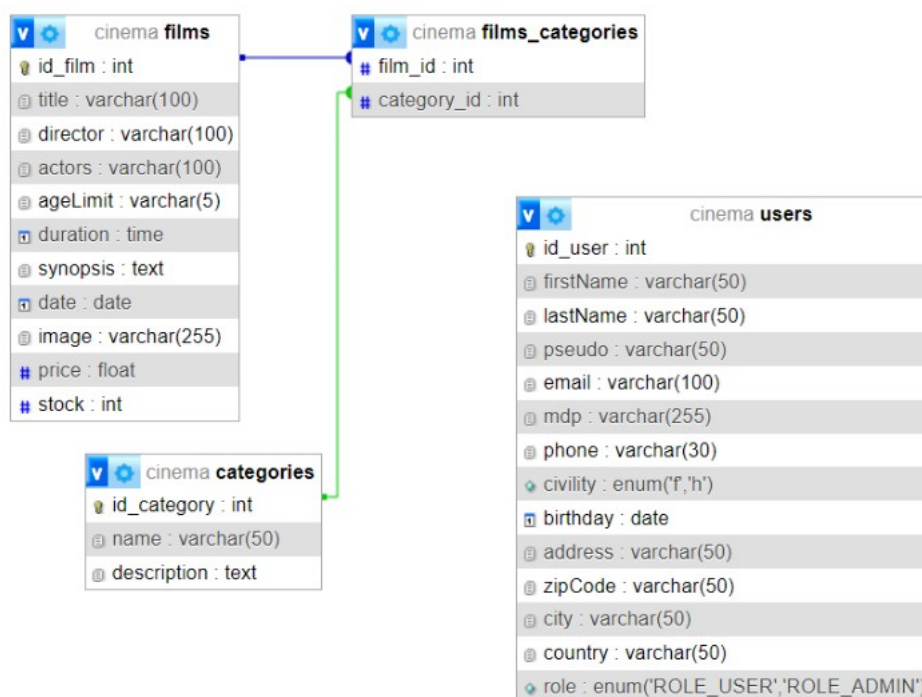
Les différentes tables nécessaires : "films", "categories", "users" ont été créées grâce à des requêtes SQL :

```
CREATE TABLE IF NOT EXISTS  
INT  
NOT NULL  
PRIMARY KEY  
VARCHAR  
ENUM
```

```
$sql = "CREATE TABLE IF NOT EXISTS users (  
    id_user INT PRIMARY KEY AUTO_INCREMENT,  
    firstName VARCHAR(50) NOT NULL,  
    lastName VARCHAR(50) NOT NULL,  
    pseudo VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    mdp VARCHAR(255) NOT NULL,  
    phone VARCHAR(30) NOT NULL,  
    civility ENUM('f', 'h') NOT NULL,  
    birthday DATE NOT NULL,  
    address VARCHAR(50) NOT NULL,  
    zipCode VARCHAR(50) NOT NULL,  
    city VARCHAR(50) NOT NULL,  
    country VARCHAR(50) NOT NULL,  
    role ENUM('ROLE_USER', 'ROLE_ADMIN') DEFAULT 'ROLE_USER'  
);
```

Ce projet pratique nous a permis d'apprendre les bases du PHP, notamment la connexion à une base de données et les différentes syntaxes pour interagir avec celle-ci.

En avançant sur ce projet de création du site web "**Cinéma**", nous avons pu au fur et à mesure mettre en application nos connaissances en back-end, en travaillant avec PHP et en interagissant avec une base de données relationnelle.



La table "**films**" contient des informations sur les différents films, tels que leur titre, leur réalisateur, leur synopsis, leur date de sortie, etc.

La table "**categories**" contient des informations sur les catégories de films, telles que l'action, la comédie, le drame, etc.

La table "**users**" contient des informations sur les utilisateurs, telles que leur nom, leur adresse e-mail, leur mot de passe, etc.

Il y a une relation de **Many-to-Many** entre les tables "**films**" et "**categories**".

# DOSSIER PROFESSIONNEL (DP)

Cela signifie qu'un film **peut appartenir** à plusieurs catégories et qu'une catégorie **peut contenir** plusieurs films. Pour gérer cette relation, il y a une table intermédiaire "films\_categories" qui contient les associations entre les films et les catégories.

## 2. Précisez les moyens utilisés :

Visual Studio Code, HTML 5 , CSS 3, Bootstrap, PhpmyAdmin, MySQL, APACHE

## 3. Avec qui avez-vous travaillé ?

Avec le formateur et les apprenants

## 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 : *Cli* au : *Cli*

## 5. Informations complémentaires (facultatif)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 2 ► Développer des composants d'accès aux données SQL et NoSQL

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

Avant de débiter dans le langage de programmation PHP, nous avons d'abord appris à créer et manipuler des bases de données à l'aide de requêtes SQL via l'interface phpMyAdmin. Cette étape nous a permis de comprendre le fonctionnement de la gestion des données et de maîtriser les requêtes les plus souvent utilisées :

**CREATE TABLE** => permet de créer de nouvelles tables

**SELECT** => permet de sélectionner et récupérer des données d'une ou plusieurs tables.  
**WHERE**,

**INSERT INTO** => permet d'insérer de nouvelles données dans une table. Elle est utilisée pour ajouter de nouvelles lignes.

**UPDATE** => permet de modifier les valeurs existantes d'une ou plusieurs colonnes dans une table. Elle est généralement utilisée avec **SET** pour définir les nouvelles valeurs et **WHERE** pour spécifier les lignes à mettre à jour.

**DELETE** => permet de supprimer des lignes d'une table. Elle est souvent utilisée avec la clause **WHERE** pour spécifier les lignes à supprimer.

**CREATE TABLE** pour créer de nouvelles tables,  
**ORDER BY**,  
**GROUP BY**, etc. pour filtrer, trier et agréger les résultats.  
**ALTER TABLE** pour modifier la structure d'une table existante,

#### Exemple d'exercices :

```
/*Créer les tables :  
- stagiaires  
- bonbons*/
```

```
CREATE TABLE STAGIAIRES (*  
id_stagiaire INT(11) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,  
prenom VARCHAR(50) NOT NULL,  
yeux VARCHAR(10) NOT NULL,  
genre VARCHAR(10) NOT NULL,  
PRIMARY KEY (id_stagiaire)  
);  
  
INSERT INTO STAGIAIRES (prenom, yeux, genre)  
VALUES ('Assia', 'maron', 'f'),  
('César', 'maron', 'm'),  
('Issa', 'maron', 'm'),  
('Julius', 'maron', 'm'),  
('Mehdi', 'maron', 'm'),  
('Fernando', 'maron', 'm'),  
('Brice', 'maron', 'm'),  
('Roberto', 'maron', 'm'),  
('Sophie', 'maron', 'f'),  
('Gwladys', 'maron', 'f'),  
('Elisha', 'maron', 'f'),  
('Malla', 'maron', 'f');
```

```
CREATE TABLE BONBONS (*  
id_bonbon INT(11) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,  
nom VARCHAR(50) NOT NULL,  
saveur VARCHAR(50) NOT NULL,  
PRIMARY KEY (id_bonbon)  
);  
  
INSERT INTO BONBONS (nom, saveur)  
VALUES ('dragibus', 'cola'),  
('tagada', 'fraise'),  
('bams', 'banane'),  
('rotella', 'reglisse'),  
('floppys', 'sucre'),  
('rainbollows', 'marshmallow'),  
('bighari', 'cerise'),  
('oursor', 'fruits'),  
('croco', 'fruits'),  
('oeufoplat', 'guimauve'),  
('flanbotti', 'caramel');
```

# DOSSIER PROFESSIONNEL (DP)

```
/*Changer le prénom du stagiaire qui a l'id 100 de Gemeeeda à Gemeda */  
UPDATE stagiaires  
SET prenom = 'Gemeda'  
WHERE id_stagiaire = 100  
ou WHERE stagiaires.id_stagiaire = 100
```

Ces différentes requêtes SQL m'a appris à interagir efficacement avec les bases de données lors de nos différents projets web en PHP.

## 2. Précisez les moyens utilisés :

Visual Studio Code, MongoDB, Mongosh,

## 3. Avec qui avez-vous travaillé ?

#### 4. Contexte

Nom de l'entreprise, organisme ou association **Colombbus**

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

Période d'exercice Du : **Cli** au : **Cli**

#### 5. Informations complémentaires (facultatif)

## Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Attestation	Declick Pro Colombbus	
Attestation PIX		
Attestation	Konexio	2022
Titre	Area Formation Créteil	2015



---

# DOSSIER PROFESSIONNEL (DP)

---

---

## Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Gwladys JACOBIN..... ,  
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 Paris..... le .....  
pour faire valoir ce que de droit.

Signature :

# DOSSIER PROFESSIONNEL (DP)

## Documents illustrant la pratique professionnelle

*(facultatif)*

### Intitulé

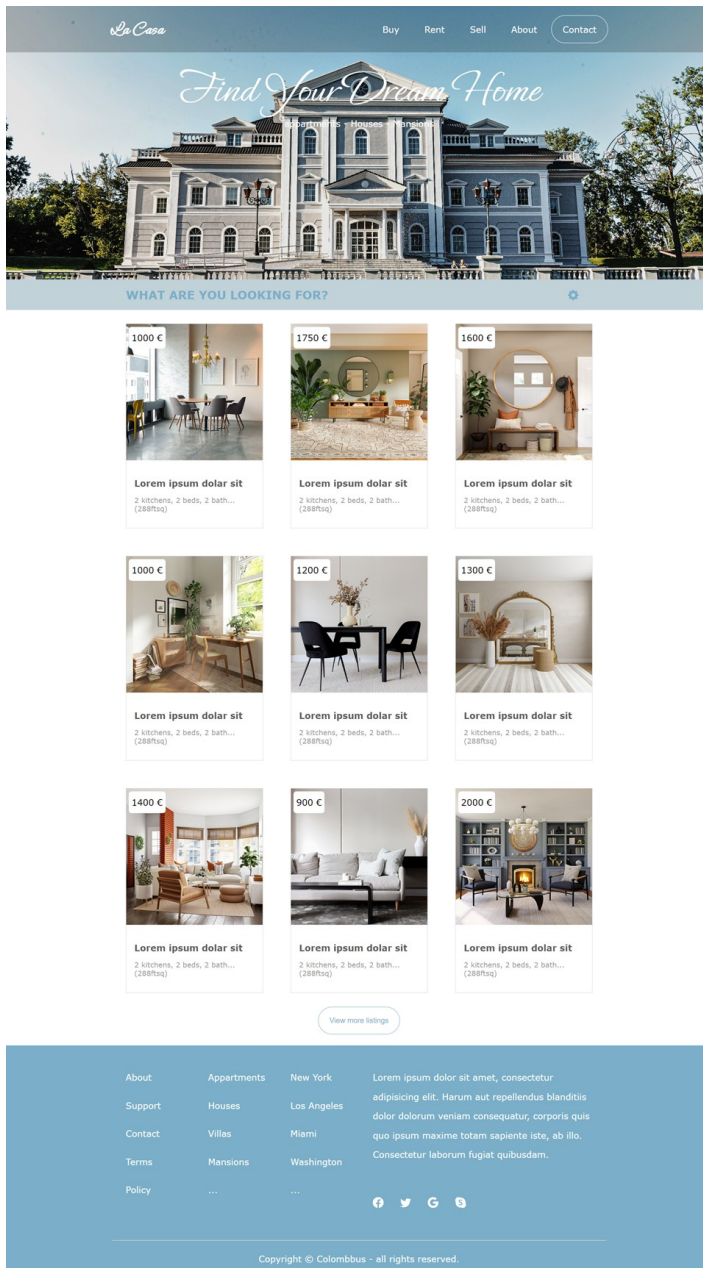
Cliquez ici pour taper du texte.

# ANNEXES

(Si le RC le prévoit)

## Annexe 1

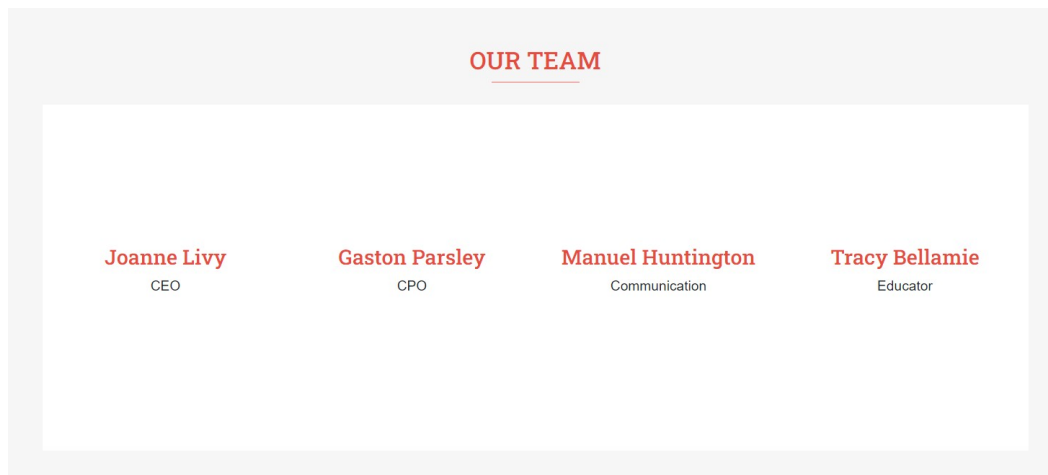
Maquette “La Casa” format normal et format téléphone



## Annexe 2

# DOSSIER PROFESSIONNEL (DP)

**Affichage de la section 'team' sans le survol de la souris** : le background est blanc, les titres sont rouges et les paragraphes sont noirs



**Affichage de la section 'team' avec le survol de la souris** : le background est blanc est devenue une image en background, les titres sont rouges et mais les paragraphes sont blancs



## Annexe 3

## Site "Cinéma"

Pages films et Dashboard coté Admin (profil) lorsque l'on se connecte avec notre identifiants, notre adresse mail et de notre mot de passe

