



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

- Bare

Nom d'usage

-

Prénom

- Esteban

Adresse

- 788 Avenue de la libération

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 :

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

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 Installer et configurer son environnement de travail en fonction du projet web ou web mobile p. 5
- CP 2 Maquetter des interfaces utilisateur web ou web mobile. p. 6
- CP 3 Réaliser des interfaces utilisateur statiques web ou web mobile p. 15
- CP 4 Développer la partie dynamique des interfaces utilisateur web ou web mobile p. 19

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. 24

- CP 5 Mettre en place une base de données relationnelle p. 24
- CP 6 Développer des composants d'accès aux données SQL et NoSQL p. 28
- CP 7 Développer des composants métier côté serveur p. 32
- CP 8 Documenter le déploiement d'une application dynamique web ou web mobile p. 36

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



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

CP 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 ma formation DWWM, j'ai dû configurer un environnement de travail pour apprendre HTML, CSS et PHP. Pour cela, j'avais besoin d'un éditeur de code et d'un programme pour lancer un serveur local. Voici les étapes que j'ai suivies :

- Choix des Outils : J'ai opté pour Visual Studio Code comme éditeur de code principal en raison de sa légèreté, de sa flexibilité et de son vaste écosystème d'extensions pour le développement web. Pour le serveur local, j'ai choisi Laragon en raison de sa facilité d'installation et de configuration pour le développement PHP.
- Installation de Visual Studio Code : J'ai téléchargé et installé Visual Studio Code à partir du site officiel. Ensuite, j'ai configuré les paramètres de base tels que les thèmes, les raccourcis clavier et les extensions nécessaires pour le développement front-end (HTML, CSS) et back-end (PHP).
- Configuration de Laragon : J'ai également téléchargé et installé Laragon, en veillant à sélectionner les composants nécessaires comme PHP, Apache et MySQL. J'ai configuré Laragon pour qu'il soit prêt à exécuter des applications PHP localement, ajustant les paramètres du serveur et créant des bases de données de test au besoin.

2. Précisez les moyens utilisés :

J'ai utilisé un navigateur web pour télécharger les installateurs de Visual Studio Code et de Laragon, puis je les ai installés et configurés selon mes besoins.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

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

Chantier, atelier, service ▶ *Exercice de formation*

Période d'exercice ▶ Du : *06/11/2023* au : *31/07/2024*

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.

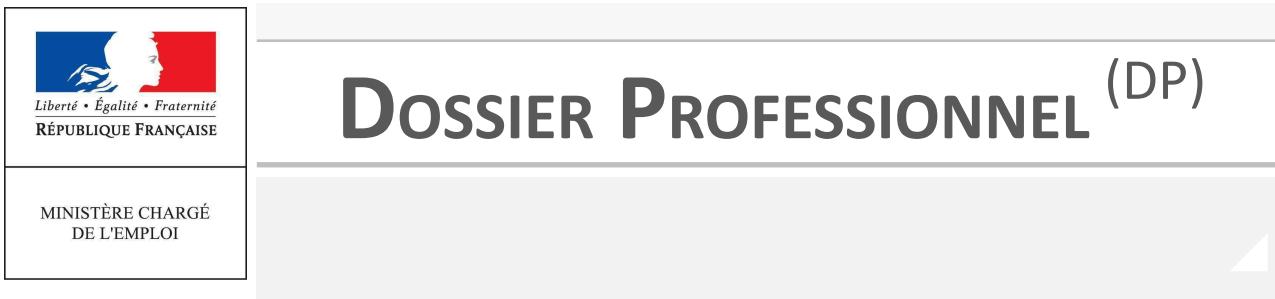
CP 2 ▶ *Maquetter des interfaces utilisateur web ou web mobile.*

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

Dans le cadre d'un projet de groupe, nous avons réalisé la maquette d'une boutique en ligne appelée TeaCoffee. Cette maquette a été créée pour optimiser notre travail et accélérer l'avancement du projet. En disposant d'une vue d'ensemble de la structure et du nombre de pages (templates), nous pouvons nous concentrer sur chaque page spécifique et travailler dessus plus facilement en nous référant à la maquette.

1. Maquette

La maquette a été développée en utilisant l'outil Figma, ce qui nous a permis de visualiser l'apparence finale de la boutique en ligne TeaCoffee. Nous avons choisi des couleurs, des polices, et des images en accord avec l'identité visuelle que nous souhaitions donner à la boutique. La maquette inclut les pages principales telles que



DOSSIER PROFESSIONNEL (DP)

la page d'accueil, la page produit, et la page de panier, chacune avec une mise en page cohérente et esthétique.

Page d'accueil :

The screenshot shows the homepage of the Tea'Coffee website. At the top, there's a navigation bar with links for Accueil, Café, Thé, and Contact. The main header features the Tea'Coffee logo and a search bar. A large banner in the center promotes their selection of coffee and tea. Below the banner, there are three sections: "Dernier produit de la boutique" (showing five Espresso Elegante items), "Meilleur vente Thé" (showing five Thé noir and Thé vert items), and "Meilleur vente Café" (showing five Espresso Elegante items). Each product item includes a small image, the name, price (€10 or €15), and a "Ajouter au panier" button. At the bottom, there are social media icons for Instagram, Pinterest, YouTube, and Email, along with a copyright notice: "© 2023 All rights reserved".

Page produit:

Pages détails du produit

Acceuil Caffé Thé Contact

TeaCoffee
Drinking tea
and coffee

connexion/incription

Q

Espresso Elegante | Coffee

€12 €16 Taxe inclu

★ ★ ★ ★ ★

A propos du café

Une tasse de café est une boisson essentielle en raison de son attrait intemporel.

Facile à préparer. Il peut être brassé selon diverses méthodes, des machines goutte à goutte aux verseurs manuels.

Disponible en différentes tailles, d'un espresso standard à un grand Americano, répondant à différentes préférences.

Vous pouvez personnaliser votre café en ajoutant de la crème, du sucre ou arômes selon vos préférences gustatives.

Commentaires (10)

Évaluation	Pourcentage
5.0 ★	66%
4.0 ★	33%
3.0 ★	16%
2.0 ★	8%
1.0 ★	6%

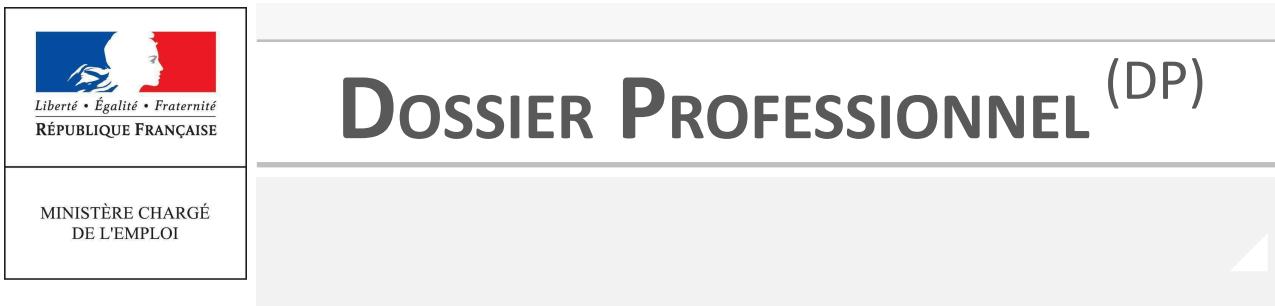
John Doe
★★★★★ 2 mins

Le service était incroyable. Je n'ai jamais eu à attendre aussi longtemps pour mon nourriture. Le personnel était sympathique et attentionné et la livraison était incroyablement rapide.

Read all reviews

cgu CGV
All rights reserved



MINISTÈRE CHARGÉ
DE L'EMPLOI

Page panier:

Panier d'achats

Accueil Caffé Thé Contact connexion/inscription

Votre Panier

Produit	Frais de port	Quantité	Prix Total
Fake Coffe #1 (\${category_name}) €10.00	€3.00	- 1 +	€13.00
Fake Coffe #2 (\${category_name}) €15.00	€3.00	- 1 +	€18.00
Fake Coffe #3 (\${category_name}) €15.00	€3.00	- 1 +	€18.00

Sous Total €40.00
Frais de port €9.00
Total €49.00

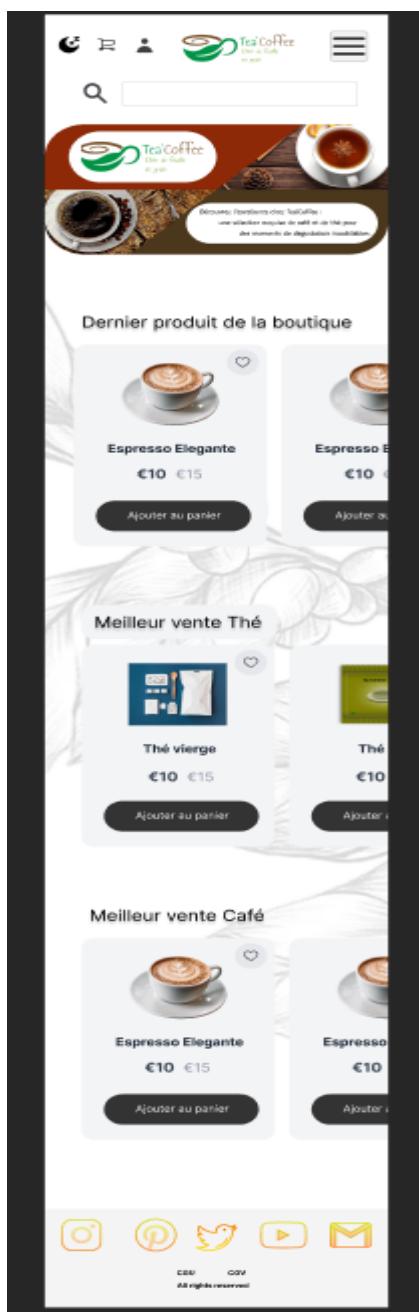
Paiement >

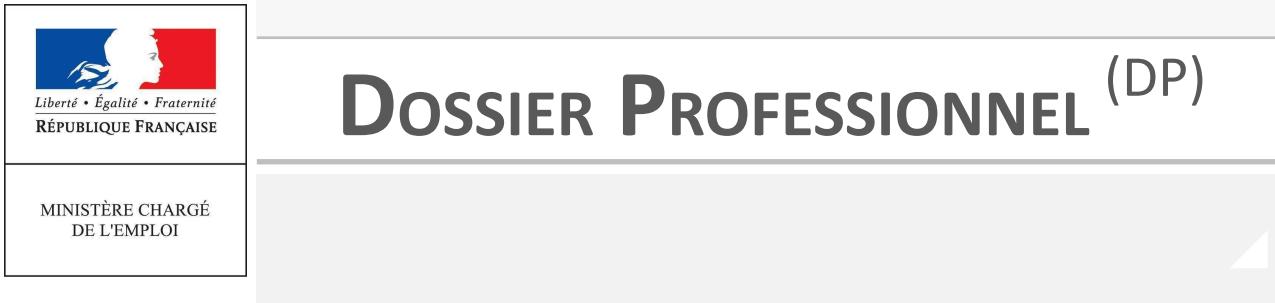
All rights reserved

This screenshot shows a shopping cart page for a coffee shop named "TeaCoffee". The page features a decorative background illustration of coffee beans and leaves. The shopping cart table lists three items: "Fake Coffe #1" at €10.00, "Fake Coffe #2" at €15.00, and "Fake Coffe #3" at €15.00. Each item includes a small image of a coffee cup and beans. The total cost is €49.00, including a delivery fee of €9.00. At the bottom, there are social media sharing icons for Instagram, Pinterest, Twitter, YouTube, and Email, along with a "Paiement" button.

Autre exemple:

Aussi la page d'accueil en responsive

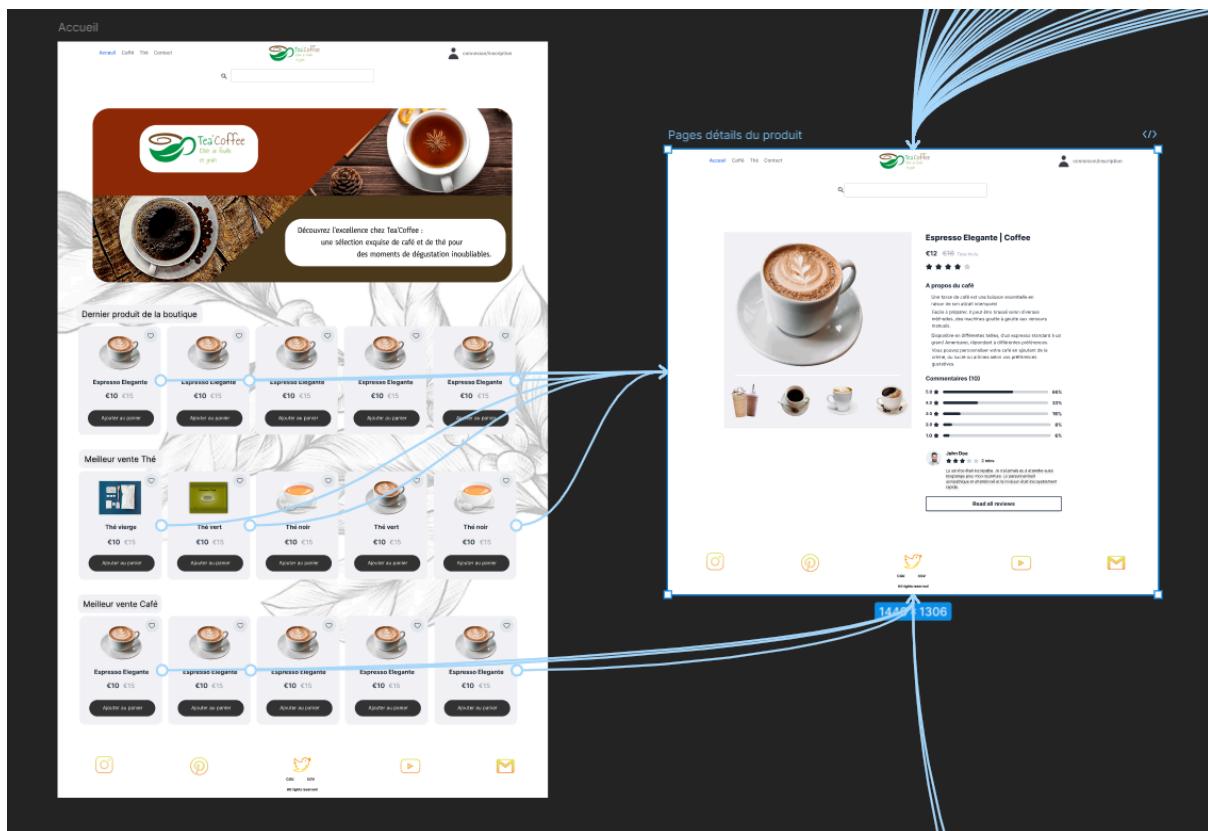




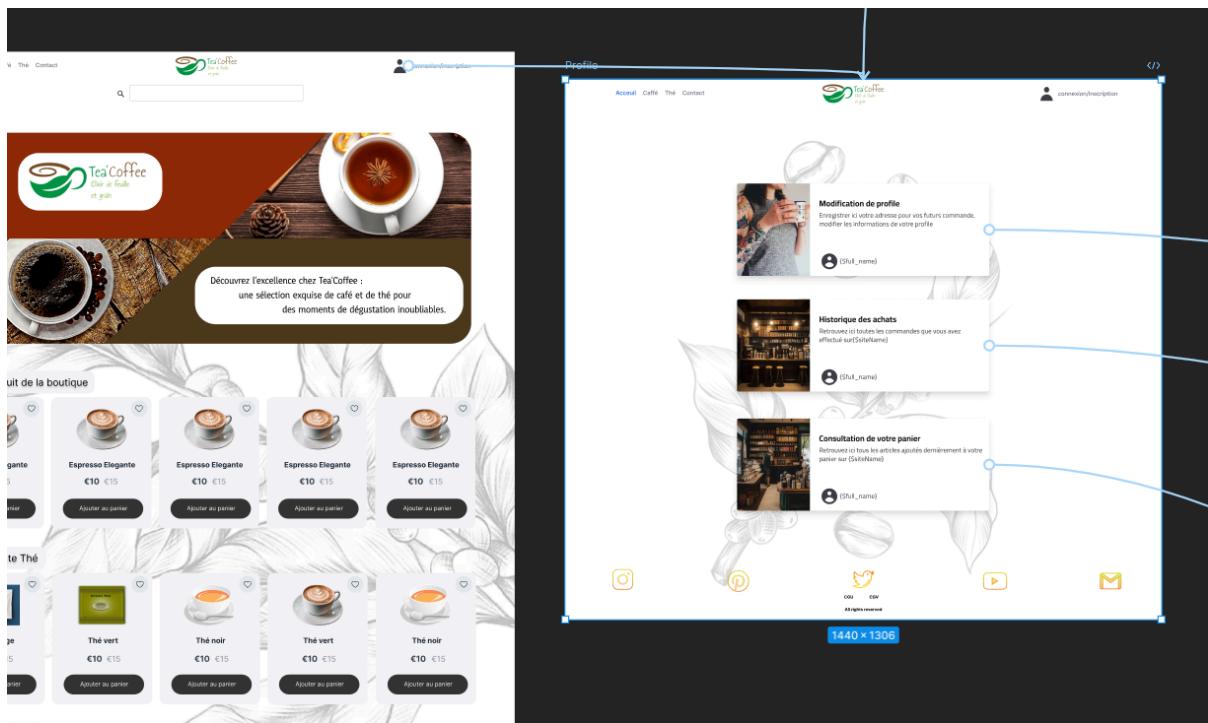
2. Wireframes

Nous avons utilisé les wireframes dans Figma pour détailler la structure et l'agencement de chaque page. Cela nous a permis de visualiser clairement la disposition des menus, boutons, images, et textes, et de simuler les interactions et les parcours utilisateur. Ces étapes nous ont aidés à affiner l'expérience utilisateur avant le développement final.

Voici des exemples de wireframes de la page d'accueil :



Ici, on peut voir que lorsque l'utilisateur clique sur un produit, cela le renvoie à la page du produit correspondante.

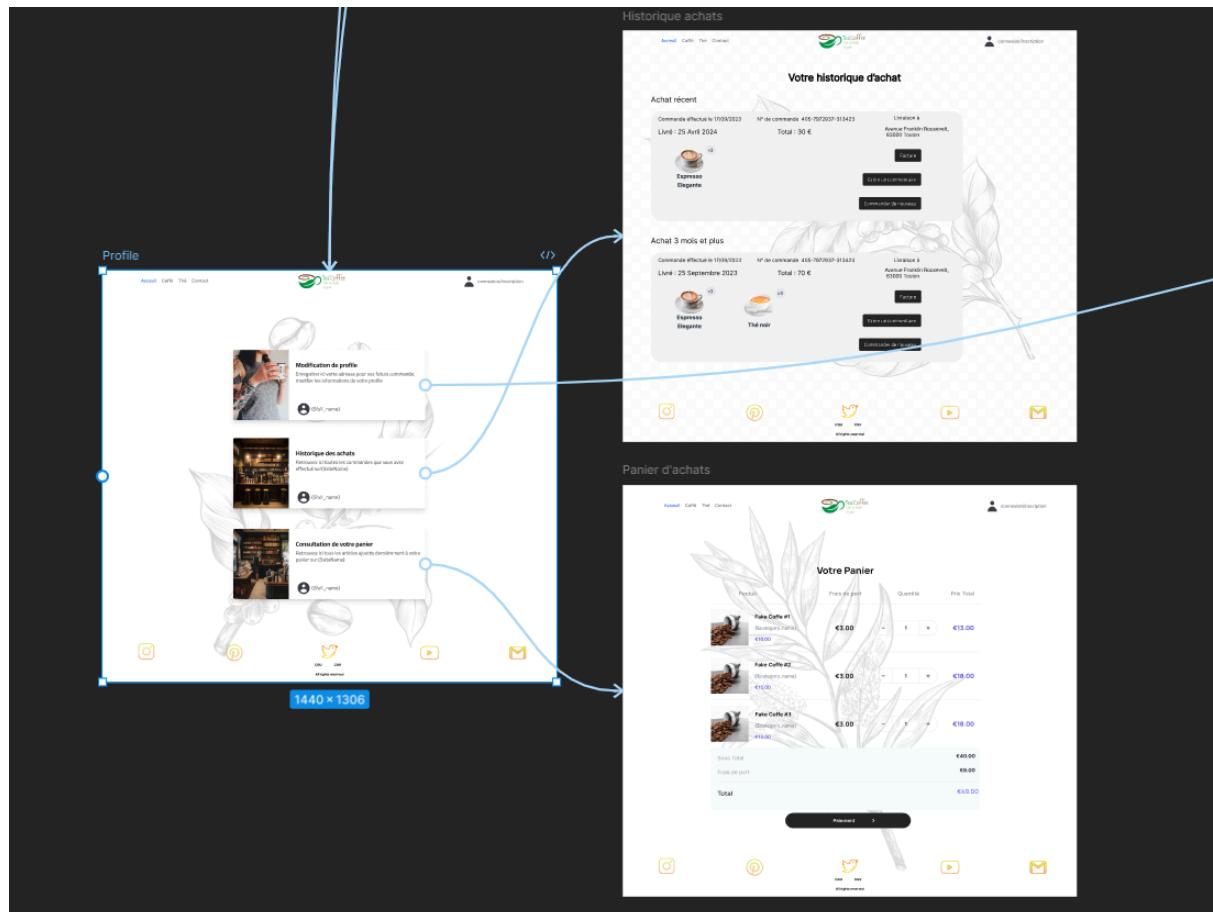


Dans cet exemple, on peut voir qu'en cliquant sur l'icône utilisateur, cela renvoie vers la page de profil.

Autre exemple:



DOSSIER PROFESSIONNEL (DP)



Ici, nous avons un bon exemple de wireframes où la page de profil permet de cliquer sur 'Panier' pour voir le panier, ou sur 'Historique d'achat' pour voir tous les achats de l'utilisateur.

2. Précisez les moyens utilisés :

Pour cette maquette on a utilisé Figma avec la fonctionnalité prototype pour le Wireframes

3. Avec qui avez-vous travaillé ?

Projet réalisé en groupe avec de compagnon de ma formation DWWM

4. Contexte

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

Chantier, atelier, service ➤ *Exercice de formation*

Période d'exercice ➤ Du : *06/11/2023* au : *31/07/2024*

5. Informations complémentaires (facultatif)



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

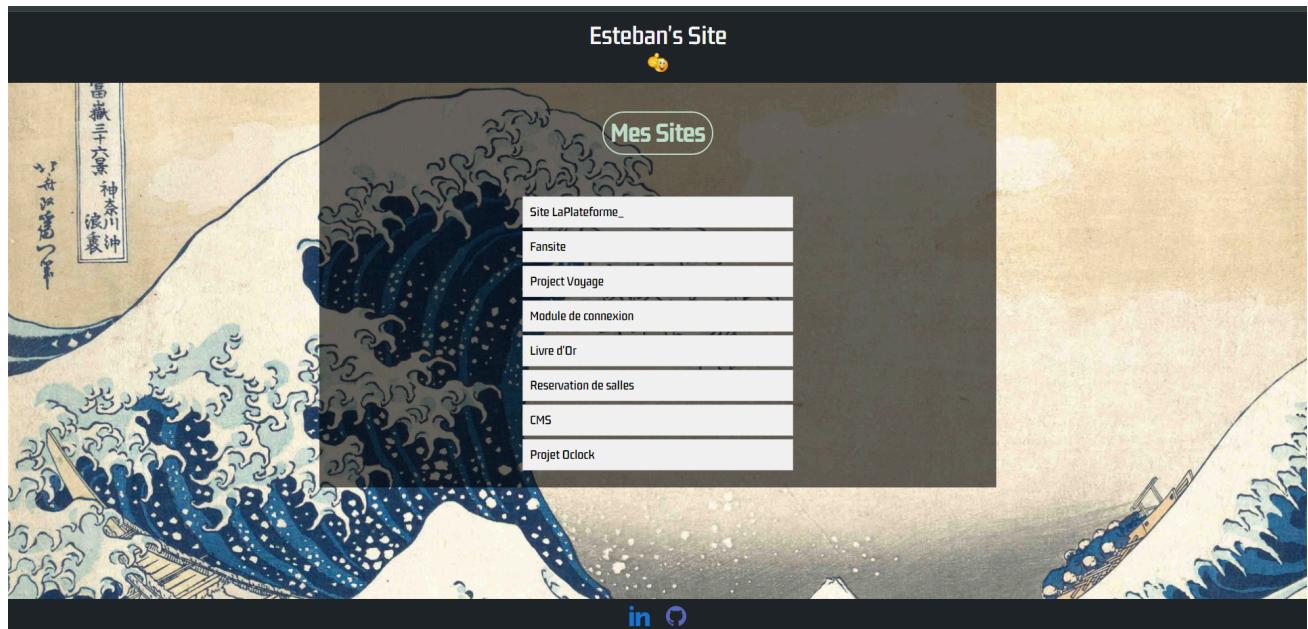
Activité-type 1

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

CP 3 - Réaliser des interfaces utilisateur statiques web ou web mobile

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

Pendant ma formation DWWM, il m'a été demandé de créer un portfolio regroupant les réalisations de mes différents projets. Ce portfolio a été conçu en utilisant HTML, CSS et un peu de JavaScript. L'interface est responsive grâce à l'utilisation de media queries dans le CSS, et le JavaScript a permis d'ajouter de l'interactivité sous forme d'accordéons avec des liens vers mes projets.



Html avec le script JS:

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Esteban's Site</title>
    <link rel="stylesheet" href="Esteban's-site.css">
    <link href="https://fonts.googleapis.com/css2?family=Kdam+Thmor+Pro&family=Montserrat:ital,wght@1,500&display=swap" rel="stylesheet">
</head>

<body>
    <header>
        <div class="flexH">
            <p class="phd">Esteban's Site</p>
            
        </div>
    </header>

    <main>
        <div class="flexh1">
            <h1>Mes Sites</h1>
        </div>
        <div class="flex">
            <div class="accordion">
                <div class="section">
                    <div class="section-header">Site LaPlateforme_</div>
                    <div class="section-content"><Site a mon image de la LaPlateforme_,<a class="links" href=".//Site-La-Plateforme/index.html">Lien</a></div>
                </div>
                <div class="section">...</div>
                <div class="section">...</div>
            </div>
        </div>
    </main>

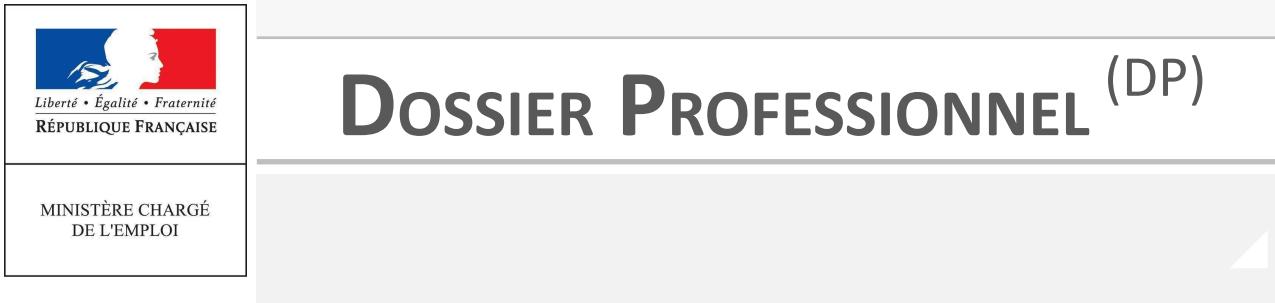
```

```
<footer>
    <a href="https://www.linkedin.com/in/esteban-bare-337927284/" target="_blank"></a>
    <a href="https://github.com/Esteban-Bare" target="_blank"></a>
</footer>

<script>
    const accordion = document.querySelector('.accordion');
    const sections = accordion.querySelectorAll('.section');

    sections.forEach(function (section) {
        var header = section.querySelector('.section-header');

        header.addEventListener('click', function () {
            sections.forEach(function (s) {
                if (s !== section) {
                    s.classList.remove('active');
                }
            });
            section.classList.toggle('active');
        });
    });
</script>
</body>
</html>
```



MINISTÈRE CHARGÉ
DE L'EMPLOI

Code CSS:

```
body {
    font-family: 'Kdam Thmor Pro', 'Montserrat', sans-serif;
    background-color: #212529;
    margin: 0;
    display: flex;
    flex-direction: column;
    min-height: 100vh;
    background-image: url("ola.jpg");
    background-size: cover;
    background-repeat: no-repeat;
}

h1 {
    text-align: center;
    border: 2px solid #c0ddca;
    border-radius: 50px;
    padding: 5px 10px;
    color: #c0ddca;
}

.flexh1, .flex {
    display: flex;
    justify-content: center;
    width: 50vw;
    margin: 0 auto;
    background-color: rgba(0, 0, 0, 0.6);
    padding: 20px;
}

.flex {
    flex: 1;
    align-items: stretch;
    padding: 20px;
}

.accordion {
    display: flex;
    flex-direction: column;
    width: 100%;
    max-width: 400px;
}

.section {
    border: 1px solid #ccc;
    margin-bottom: 5px;
}

.footer img {
    width: 30px;
    height: auto;
    margin: 0 10px;
}

@media (max-width: 768px) {
    .flexh1, .flex {
        width: 80vw;
    }
}

.section-header {
    padding: 10px;
    background-color: #f1f1f1;
    cursor: pointer;
}

.section-content {
    display: none;
    padding: 10px;
    background-color: #f1f1f1;
}

.section-header:hover, .section.active .section-header {
    background-color: #ddd;
}

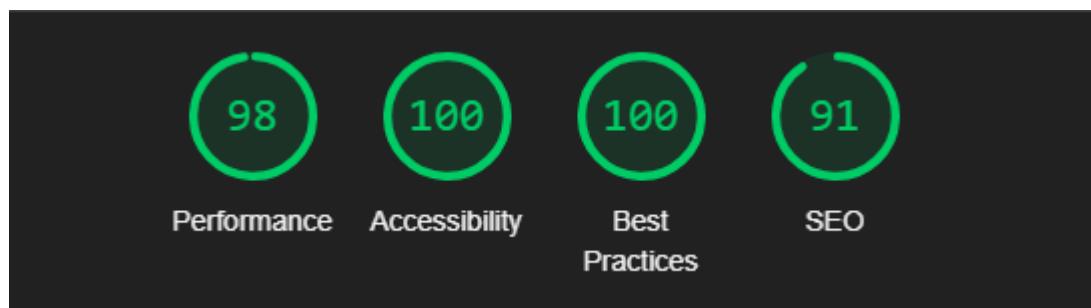
.section.active .section-content {
    display: block;
}

header, footer {
    background-color: #212529;
    color: #fff;
    padding: 10px 0;
    text-align: center;
}

.phd {
    font-size: xx-large;
    color: whitesmoke;
    margin: 0;
}

.logo {
    width: 3vw;
    max-width: 30px;
}

footer {
    background-color: #212529;
    color: white;
    text-align: center;
    padding: 10px 0;
    width: 100%;
    margin-top: auto;
    position: fixed;
    bottom: 0;
    left: 0;
    right: 0;
}
```



Pour évaluer l'accessibilité et le référencement de mon site, j'utilise Lighthouse de Google dans DevTools.

2. Précisez les moyens utilisés :

Pour ce projet, j'utilise Visual Studio Code pour coder et j'utilise les outils de DevTools pour modifier mon CSS et visualiser le rendu en format mobile. J'utilise également Lighthouse pour évaluer l'accessibilité et le référencement de mon site.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

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

Chantier, atelier, service ➤ *Exercice de formation*

Période d'exercice ➤ Du : *06/11/2023* au : *31/07/2024*

5. Informations complémentaires (facultatif)



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

CP 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 :

Pour mon projet CineTech, j'ai intégré l'API TMDB pour récupérer dynamiquement les films et séries les plus populaires ainsi que d'autres informations pertinentes comme les détails des films et les affiches actuelles. L'utilisation de cette API m'a également permis de créer une pagination, facilitant la navigation à travers les classements en profondeur.

Voici le site:

Films Home Register Login

Popular Now playing Upcoming Top rated

Popular

Titre: Inside Out 2

Titre: Despicable Me 4

Titre: The Garfield Movie

Titre: Furiosa: A Mad Max Saga

Et voici ce dont j'ai besoin pour récupérer les films :

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href=".//assets/css/movies.css">
    <link rel="stylesheet" href=".//assets/css/header.css">
</head>

<body>
    <header> ...
    </header>
    <div class="buttons">
        <button id="popular">Popular</button>
        <button id="now_playing">Now playing</button>
        <button id="upcoming">Upcoming</button>
        <button id="top_rated">Top rated</button>
    </div>
    <h1 id="contentTitle" class="h1Title"></h1>
    <div id="content" class="movies"></div>
    <div class="buttonsPage"> ...
    </div>
    <script src=".//assets/js/movies.js"></script>
    <script src=".//assets/js/1.js"></script>
    <script src=".//assets/js/2.js"></script>
    <script src="https://kit.fontawesome.com/d45865f90f.js" crossorigin="anonymous"></script>
</body>

</html>

```

L'élément clé est le script **movies.js** qui utilise l'API TMDB pour récupérer les films, ainsi que la **<div id="content"></div>** qui est initialement vide.

```
const infoDiv = document.getElementById('content');
```

Dans le script, je sélectionne la div **content** que nous avons vue précédemment.

```

function popular() {
    activePopular = true;
    activeNow_playing = false;
    activeUpcoming = false;
    activeTop_rated = false;
    infoDiv.innerHTML = "";
    contentTitle.innerHTML = "";
    buttons.forEach(button => {
        button.style.border = "none";
    });
    popularButton.style.border = "2px red solid";
    contentTitle.innerHTML = "Popular";
    fetch('https://api.themoviedb.org/3/movie/popular?language=en-US&page=' + page, options)
        .then(response => response.json())
        .then(response => {
            console.log(response);
            response.results.forEach(element => {
                const info = document.createElement('div');
                info.classList.add('movie-info');
                info.innerHTML = `
                    
                    <p>Titre: ${element.title}</p>
                `;
                infoDiv.appendChild(info);
            });
        })
        .catch(err => console.error(err));
}

```



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

Ma fonction **popular()** se charge de faire une requête à l'API TMDB en utilisant la fonction **fetch()** de JavaScript. Lorsque la réponse est reçue, elle crée des divs avec les informations du film, telles que l'image et le titre.

```
document.addEventListener('DOMContentLoaded', popular);
```

Dans ce morceau de code, on peut voir que lorsque la page est complètement chargée, j'appelle la fonction **popular()** pour obtenir les films les plus populaires de la semaine. Cela rend mon site dynamique et permet d'afficher du contenu mis à jour pratiquement tous les jours grâce à l'utilisation de l'API et de JavaScript.

La Pagination :

```
<div class="buttonsPage">
    <button id="minus">Moins</button>
    <p id="actualPage"></p>
    <button id="plus">Plus</button>
</div>
```

Voici le bouton qui permet à l'utilisateur de changer de page.

```

const plus = document.getElementById('plus');
const minus = document.getElementById('minus');
const actualPage = document.getElementById('actualPage');
actualPage.innerHTML = `Page: ${page}`;
plus.addEventListener('click', function () {
    ++page;
    if (activeNow_playing) {
        now_playing();
    } else if (activePopular) {
        popular();
    } else if (activeUpcoming) {
        upcoming();
    } else if (activeTop_rated) {
        top_rated();
    }
    actualPage.innerHTML = `Page: ${page}`;
});

minus.addEventListener('click', function () {
    if (page > 1) {
        --page;
        if (activeNow_playing) {
            now_playing();
        } else if (activePopular) {
            popular();
        } else if (activeUpcoming) {
            upcoming();
        } else if (activeTop_rated) {
            top_rated();
        }
        actualPage.innerHTML = `Page: ${page}`;
    } else {
        console.log('cannot go further');
    }
});

```

```
let page = 1;
```

On peut voir ici toute la logique de la pagination, car l'API TMDB permet de faire des requêtes sur différentes pages.

```
fetch('https://api.themoviedb.org/3/movie/popular?language=en-US&page=' + page, options)
```

Grâce à cela, lorsque l'utilisateur clique sur "+" ou "-", le site recharge la page de la catégorie que l'utilisateur souhaite voir. Par exemple, si l'utilisateur se trouve sur la page de base, dans la catégorie "populaire", il verra les premiers 20 films. En cliquant sur le bouton "+", le JavaScript détecte ce clic, vérifie la catégorie, et appelle la fonction **popular()**, comme expliqué précédemment. Cette fois, la variable page sera incrémentée à 2, ce qui déclenche une requête pour récupérer les 20 films de la page 2.



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

J'ai utilisé Visual Studio Code et les outils de développement du navigateur, ainsi que l'API TMDB pour les requêtes. Les technologies employées sont JavaScript, HTML, et CSS.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ➔ *Ecole La Plateforme*

Chantier, atelier, service ➔ *Exercice de formation*

Période d'exercice ➔ Du : *06/11/2023* au : *31/07/2024*

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.

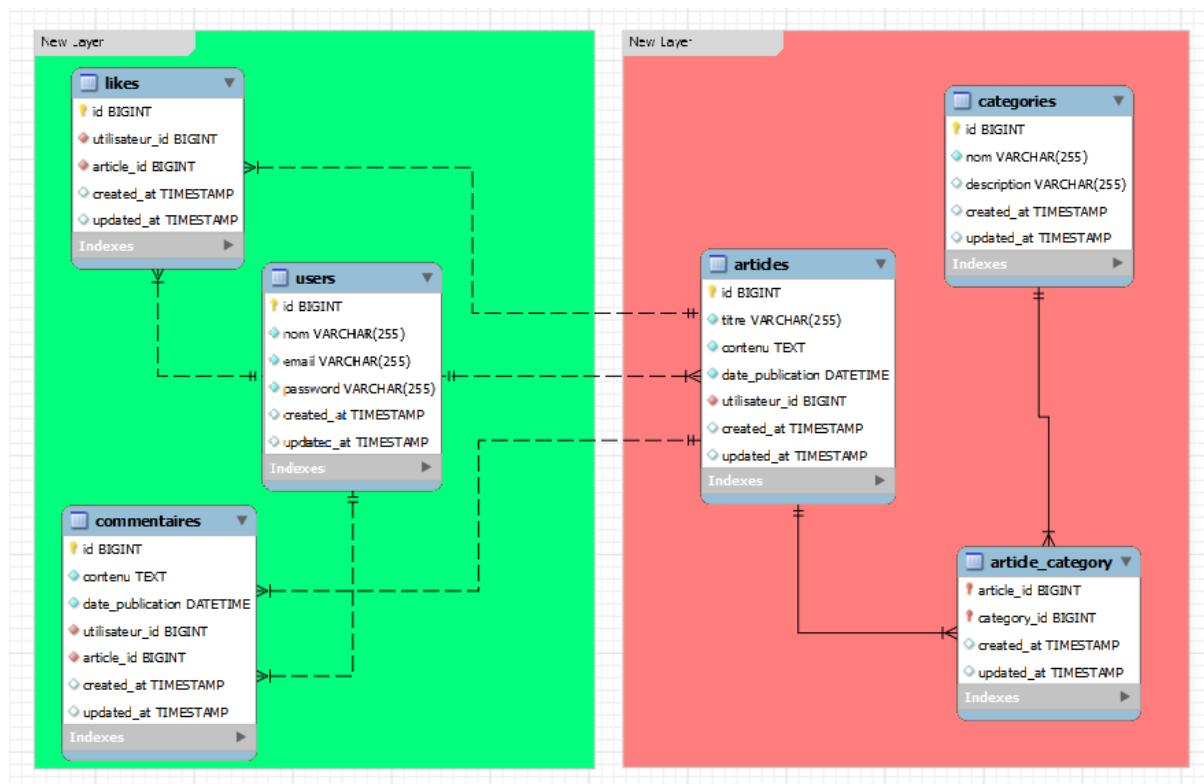
CP 5 - Mettre en place une base de données relationnelle

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

Pour mon projet personnel de DWWM, j'ai créé un site d'articles. Pour cela, j'ai eu besoin d'une base de données avec les bonnes relations, créée de manière adéquate en utilisant la méthode Merise. J'ai utilisé MySQL Workbench pour créer mon MCD (Modèle Conceptuel de Données).

Le MCD:

Pour ce MCD, j'ai utilisé MySQL Workbench, qui m'a permis de structurer mes tables et d'établir les bonnes relations.



Dans ce MCD, on peut voir deux grandes tables principales : **users** et **articles** avec leurs attributs. Il y a plusieurs relations, par exemple :

- Un utilisateur peut avoir plusieurs articles (relation OneToMany).
- Un utilisateur peut liker plusieurs articles.
- Un article peut avoir plusieurs commentaires.



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

J'ai également créé une table de pivot pour gérer une relation ManyToMany, car un article peut avoir plusieurs catégories et une catégorie peut avoir plusieurs articles.

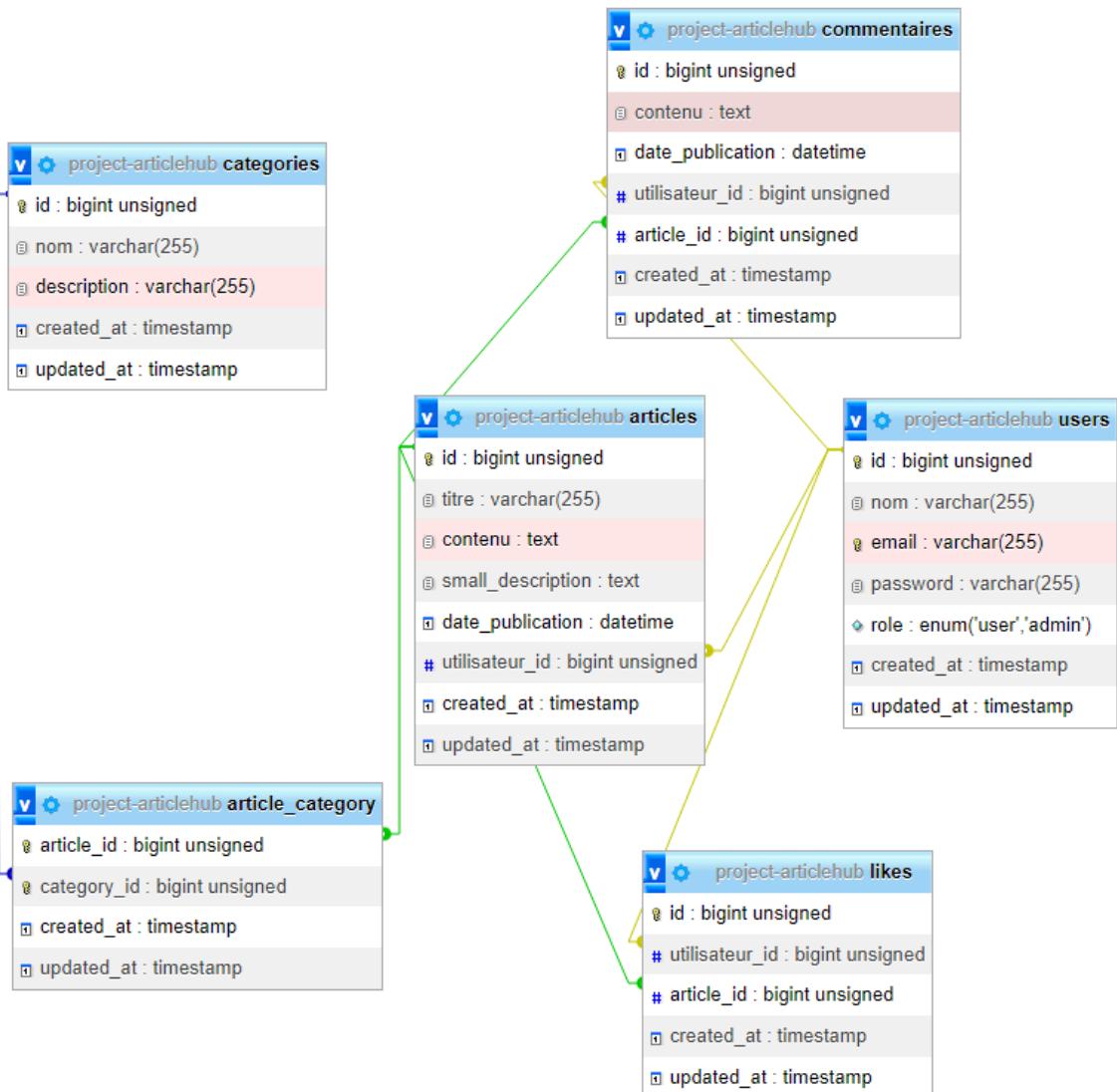
Voici les requête SQL pour créer la base de données:

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
CREATE TABLE `articles` (
  `id` bigint UNSIGNED NOT NULL,
  `titre` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `contenu` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `small_description` text COLLATE utf8mb4_unicode_ci,
  `date_publication` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `utilisateur_id` bigint UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
CREATE TABLE `article_category` (
  `article_id` bigint UNSIGNED NOT NULL,
  `category_id` bigint UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
CREATE TABLE `categories` (
  `id` bigint UNSIGNED NOT NULL,
  `nom` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `description` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
CREATE TABLE `commentaires` (
  `id` bigint UNSIGNED NOT NULL,
  `contenu` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `date_publication` datetime NOT NULL,
  `utilisateur_id` bigint UNSIGNED NOT NULL,
  `article_id` bigint UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
ALTER TABLE `migrations`
  MODIFY `id` int UNSIGNED NOT NULL AUTO_INCREMENT;
ALTER TABLE `users`
  MODIFY `id` bigint UNSIGNED NOT NULL AUTO_INCREMENT;
ALTER TABLE `articles`
  ADD CONSTRAINT `articles_utilisateur_id_foreign` FOREIGN KEY (`utilisateur_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;
ALTER TABLE `article_category`
  ADD CONSTRAINT `article_category_article_id_foreign` FOREIGN KEY (`article_id`) REFERENCES `articles` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `article_category_category_id_foreign` FOREIGN KEY (`category_id`) REFERENCES `categories` (`id`) ON DELETE CASCADE;
ALTER TABLE `commentaires`
  ADD CONSTRAINT `commentaires_article_id_foreign` FOREIGN KEY (`article_id`) REFERENCES `articles` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `commentaires_utilisateur_id_foreign` FOREIGN KEY (`utilisateur_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;
ALTER TABLE `likes`
  ADD CONSTRAINT `likes_article_id_foreign` FOREIGN KEY (`article_id`) REFERENCES `articles` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `likes_utilisateur_id_foreign` FOREIGN KEY (`utilisateur_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;
COMMIT;
CREATE TABLE `users` (
  `id` bigint UNSIGNED NOT NULL,
  `nom` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `role` enum('user','admin') COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'user',
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
ALTER TABLE `articles`
  ADD PRIMARY KEY (`id`),
  ADD KEY `articles_utilisateur_id_foreign` (`utilisateur_id`);
ALTER TABLE `article_category`
  ADD PRIMARY KEY (`article_id`, `category_id`),
  ADD KEY `article_category_category_id_foreign` (`category_id`);
ALTER TABLE `categories`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `commentaires`
  ADD PRIMARY KEY (`id`),
  ADD KEY `commentaires_utilisateur_id_foreign` (`utilisateur_id`),
  ADD KEY `commentaires_article_id_foreign` (`article_id`);
ALTER TABLE `likes`
  ADD PRIMARY KEY (`id`),
  ADD KEY `likes_utilisateur_id_foreign` (`utilisateur_id`),
  ADD KEY `likes_article_id_foreign` (`article_id`);
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `users_email_unique` (`email`);
ALTER TABLE `articles`
  MODIFY `id` bigint UNSIGNED NOT NULL AUTO_INCREMENT;
ALTER TABLE `categories`
  MODIFY `id` bigint UNSIGNED NOT NULL AUTO_INCREMENT;
ALTER TABLE `commentaires`
  MODIFY `id` bigint UNSIGNED NOT NULL AUTO_INCREMENT;
ALTER TABLE `likes`
  MODIFY `id` bigint UNSIGNED NOT NULL AUTO_INCREMENT;

```

On peut observer ici les clés étrangères ainsi que la clé auto-incrémentée.



Et voici le résultat de la base de données insérée dans phpMyAdmin.

2. Précisez les moyens utilisés :

J'ai utilisé MySQL Workbench pour le MCD, Visual Studio Code pour consulter les commandes SQL, et phpMyAdmin pour visualiser la base de données.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Ecole La Plateforme



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Chantier, atelier, service ▶

Exercice de formation

Période d'exercice

► Du : 06/11/2023 au : 31/07/2024

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 des composants d'accès aux données SQL et NoSQL

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

Pendant un exercice que j'ai dû réaliser durant ma formation DWWM, j'ai eu besoin d'utiliser une connexion à la base de données que j'ai mise en œuvre avec la POO intégrée dans PHP. Grâce à cela, j'ai pu effectuer des requêtes SQL.

1. Connexion à la base de données:

```
<?php
$host = "localhost";
$db = "draft-shop";
$user = "root";
$password = "";
$dsn = "mysql:host=$host;dbname=$db;charset=utf8mb4";

try {
    $pdo = new PDO($dsn, $user, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connexion à la base de données établie.";
} catch (PDOException $e) {
    echo "Erreur de connexion : " . $e->getMessage();
}
```

Dans ce morceau de code, on voit comment je me connecte à la base de données MySQL avec PDO et les paramètres associés.

```
class Product
{
    49 references
    private $pdo;
    44 references
    private int $id;
    45 references
    private int $category_id;
    54 references
    private string $name;
    77 references
    private $photo;
    48 references
    private int $price;

    13 references | 2 overrides
    public function __construct(PDO $pdo, $id = 0, $category_id = 0, $name = '', $photo = [], $price = 0)
    {
        $this->pdo = $pdo;
        $this->id = $id;
        $this->category_id = $category_id;
        $this->name = $name;
        $this->photo = $photo;
        $this->price = $price;
    }
}
```



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

```
public function getOneById($id)
{
    $sql = "SELECT * FROM product WHERE id = :id";
    $statement = $this->pdo->prepare($sql);
    $statement->bindParam(':id', $id, PDO::PARAM_INT);
    $statement->execute();
    $product = $statement->fetch(PDO::FETCH_ASSOC);
    $this->id = $product['id'];
    $this->category_id = $product['id_category'];
    $this->name = $product['name'];
    $this->photo = $product['photos'];
    $this->price = $product['price'];
    echo $this->getCategory_id();
    echo $this->getName();
    echo $this->getPhoto();
    echo $this->getPrice();
}
```

Ici, on peut voir deux choses : premièrement, la classe modèle 'Product' avec ses attributs qui récupèrent les valeurs du constructeur PDO pour effectuer la requête. Deuxièmement, on voit la fonction 'getOneById(\$id)' qui prend en paramètre l'identifiant du produit et effectue une requête dans la base de données en fonction de cet identifiant. On peut observer que j'utilise 'bindParam' pour ajouter de la sécurité et prévenir les injections SQL. De plus, j'utilise les getters qui me permettent de retourner des informations sur le produit depuis la base de données.

```

class Category
{
    18 references
    private $pdo;
    19 references
    private int $id;
    10 references
    private string $name;
    10 references
    private string $description;

    1 reference | 0 overrides
    public function __construct(PDO $pdo, $id = 0, $name = '', $description = '')
    {
        $this->pdo = $pdo;
        $this->id = $id;
        $this->name = $name;
        $this->description = $description;
    }
    1 reference | 0 overrides
    public function getProducts()
    {
        $pdo = $this->pdo;
        $sql = "SELECT name FROM product WHERE id_category = :id";
        $statement = $pdo->prepare($sql);
        $statement->bindParam(':id', $this->id, PDO::PARAM_INT);
        $statement->execute();
        $resultat = $statement->fetchAll(PDO::FETCH_ASSOC);
        $return = '';
        echo 'Products :<br>';
        foreach ($resultat as $result) {
            foreach ($result as $result2) {
                $return .= $result2 . '<br>';
            }
        }
        return $return;
    }
}

```

De même pour le modèle 'Category', prenez en compte ma connexion PDO. Ensuite, j'ai la possibilité d'utiliser la méthode 'getProducts()' de la classe 'Category', qui me permet d'obtenir tous les produits associés à une catégorie dans la base de données.

2. Précisez les moyens utilisés :

J'ai utilisé Visual Studio Code et phpMyAdmin pour voir la BDD

3. Avec qui avez-vous travaillé ?



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

J'ai travaillé seul pour ce projet.

4. Contexte

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

Chantier, atelier, service ➤ *Exercice de formation*

Période d'exercice ➤ Du : *06/11/2023* au : *31/07/2024*

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 7 ► Développer des composants métier côté serveur

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

Durant mon stage en entreprise, j'ai eu besoin de créer un back-end pour gérer l'authentification et l'inscription des utilisateurs afin de permettre l'utilisation des fonctionnalités dans l'application. Pour cela, j'ai souhaité pratiquer le développement de serveur back-end en JavaScript, donc j'ai utilisé Express pour le routage et pour lancer le serveur. Pour la vérification des requêtes, j'ai utilisé un token JWT (JSON Web Token) pour authentifier les utilisateurs connectés.

Serveur express:

```
import express from 'express';
import http from 'http';
import cors from 'cors';
import cookieParser from 'cookie-parser';
import authRoutes from './routes/authRoutes';
import sequelize from './db/databaseQuery'

const app = express();
const server = http.createServer(app);

app.use(cors({
    origin: 'http://localhost:4200',
    methods: ['GET', 'POST'],
    allowedHeaders: ['Content-Type'],
    credentials: true
}));
app.use(cookieParser());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

// Routes
app.use(authRoutes);

// Start server
const PORT = process.env.PORT || 3000;
server.listen(PORT, () => {
    console.log(`Server running at http://localhost:${PORT}`);
});
```



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

Ici, on peut voir que la variable `app` instance Express, ce qui me permettra, avec `server`, de créer un serveur HTTP pour lancer le serveur plus tard. Je configure diverses utilités, par exemple, CORS pour ajouter de la sécurité en tant qu'API REST, étant donné que j'accepte uniquement les requêtes provenant de mon front-end sur '<http://localhost:4200>'. On voit également que j'utilise `authRoutes`, qui est le contrôleur de mes routes que je vais montrer plus tard. À noter que j'utilise TypeScript, que je compile avant de lancer le serveur avec Node.js.

```
const router = express.Router();
const secret = 'secret-for-jwt';

router.post('/login', async (req: Request, res: Response) => {

    console.log(req.body)

    const { email, password } = req.body;

    try {
        const user = await Users.findOne({ where: { email } });

        console.log(user)

        if (!user) {
            return res.status(404).json({ msg: 'User not found' });
        }

        if (password !== user.dataValues.password) {
            return res.status(401).json({ msg: 'Invalid password' });
        }

        const payload = { name: user.dataValues.name, role: user.dataValues.role };
        const newToken = jwt.sign(payload, secret);

        res.cookie('token', newToken, { httpOnly: true });
        return res.json({ msg: 'Login successful' , name: user.dataValues.name});
    } catch (error) {
        console.error('Error during login:', error);
        // const body : ApiBody = {
        //     msg : 'Server error',
        //     data : null
        // }
        // res.status(500).json(body);
    }
});
```

Ici, on peut voir que j'utilise le routeur d'Express pour créer ma route `/login`. Sur cette route, je vérifie les données envoyées pour la connexion en utilisant l'ORM Sequelize. Je vérifie si l'utilisateur existe ; si c'est le cas, je réponds à la requête avec un cookie contenant un JWT. Le payload de ce JWT contient le nom de l'utilisateur et son rôle dans la base de données, ainsi qu'un message.

```
function checkJwt(req: Request, res: Response, next: Function) {
  const token = req.cookies.token;

  jwt.verify(token, secret, (err: any, decodedToken: any) => {
    if (err) {
      return res.status(401).json('Unauthorized, wrong token');
    }
    switch (decodedToken.role) {
      case 'admin':
        next();
        break;
      default:
        res.status(401).json({ msg: `not authorized user` });
        break;
    }
  });
}
```

Cette fonction permet de vérifier le token dans la requête à mon API REST en utilisant `jwt.verify` du package `jsonwebtoken`. `'jwt.verify'` va d'abord vérifier s'il y a un token, puis comparer le secret du token avec le mien, et enfin vérifier le payload dans le token décodé pour voir le rôle de celui-ci.

```
router.get('/users', checkJwt, async (req: Request, res: Response) => {
  try {
    const users = await Users.findAll();
    res.json(users);
  } catch (error) {
    console.error('not found', error);
    res.status(500).json({ error: 'not found' });
  }
});
```

Voici un exemple où, en utilisant une route en tant qu'API REST, le serveur vérifie le token de la personne qui fait la requête avant d'envoyer tous les utilisateurs présents dans la base de données.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai utilisé : Express et sa documentation, jsonwebtoken et sa documentation, ainsi que l'ORM Sequelize pour les requêtes avec la base de données SQL.



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *Pendant mon stage en entreprise.*

Chantier, atelier, service ➤ *Projet de stage.*

Période d'exercice ➤ Du : 17/06/2024 au : 09/08/2024

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 8 Documenter le déploiement d'une application dynamique web ou web mobile

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

Pour cet exemple de déploiement, j'utilise mon projet personnel de DWWM créé en Laravel. Les pré-requis nécessaires sont : PHP 8.2 ou supérieur , Composer, MySQL et Git (pour cloner le dépôt si possible)

1. Cloner le Répertoire GitHub

Vu que j'utilise GitHub pour sauvegarder mon travail et que mon répertoire est public, on peut cloner mon dépôt dans le terminal de Visual Studio Code :

“git clone <https://github.com/Esteban-Bare/Project-ArticleHub>”

Ensuite, accédez au répertoire cloné :

“cd votre-projet”

2. Installer les Dépendances

Il faut installer toutes les dépendances avant de configurer l'application :

“composer install”

3. Configurer l'Environnement

Copiez le fichier **.env.example** en **.env** puis mettez à jour la configuration de votre base de données dans le fichier **.env**.

4. Générer la Clé d'Application

Pour utiliser l'application Laravel, il faut une clé unique :

“php artisan key:generate”

5. Créer la Base de Données

Après avoir configuré la base de données, utilisez les migrations de mon projet pour créer les tables et les relations avec la commande :

“php artisan migrate”



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

6. Données Factices

Pour pouvoir utiliser le site en totalité, après avoir créé les tables, vous pouvez exécuter les seeders que j'ai créés dans mon projet pour avoir des données factices et accéder à toutes les fonctionnalités. Voici la commande :

"php artisan db:seed"

7. Lancer le Serveur

Enfin, lancez le serveur Laravel avec la commande :

"php artisan serve"

Cela vous renverra un message avec l'URL du site, généralement **http://127.0.0.1:8000**.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai utilisé : Laravel avec sa documentation , Composer pour la gestion des dépendances PHP et MySQL.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet.

4. Contexte

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

Chantier, atelier, service ▶ *Projet personnel DWWM*

Période d'exercice ➤ Du : 06/11/2023 au : 31/07/2024

5. Informations complémentaires (facultatif)



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

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.

Déclaration sur l'honneur

Je soussigné(e) Esteban Bare

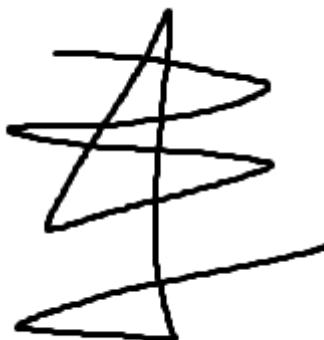
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à La Valette Du Var

le 17/07/2024

pour faire valoir ce que de droit.

Signature :





MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(*facultatif*)

Intitulé

Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)