



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*

► *Detroy*

*Nom d'usage*

► *Entrez votre nom d'usage ici.*

*Prénom*

► *Alexandre*

*Adresse*

► *Quartier pré de la Roque, 83830 Figanières*

## Titre professionnel visé

*Développeur Web et Web Mobile*

### MODALITÉ D'ACCÈS :

- Parcours de formation
- Validation des Acquis de l'Expérience (VAE)



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.  
**Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen**.

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

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

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

### Ce dossier comporte :

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

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



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

 <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 Installer et configurer son environnement de travail en fonction du projet web ou web mobile. p. 5
- ▶ CP 2 Réaliser une interface statique et adaptable. p. 14
- ▶ CP 3 Développer une interface utilisateur web dynamique. p. 21
- ▶ CP 4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce. p. 33

**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. 46
- ▶ CP 6 Développer les composants d'accès aux données. p. 52
- ▶ CP 7 Développer la partie back-end d'une application web ou web mobile. p. 64
- ▶ CP 8 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce . p. 75

**Titres, diplômes, CQP, attestations de formation (facultatif)** p. 76

**Déclaration sur l'honneur** p. 77

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

**Annexes (Si le RC le prévoit)** p.



# DOSSIER PROFESSIONNEL (DP)

## EXEMPLES DE PRATIQUE PROFESSIONNELLE



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Activité-type 1

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

**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 :

#### Introduction

Dans le cadre de la formation j'ai réalisé un exercice d'une barre de recherche autocomplétante en Php et Javascript. J'ai mis en place l'environnement de travail par l'installation et la configuration de trois outils essentiels : Visual Studio Code ( VS Code ), Git et Laragon. Chacun de ces outils offrant des fonctionnalités cruciales pour le développement, la gestion et le déploiement de projets.

#### Installation et Configuration de VS Code

Visual Studio Code est un éditeur de code source puissant et extensible, développé par Microsoft. Il nous a été fortement recommandé car il est particulièrement apprécié pour l'interface utilisateur moderne et personnalisable, la vaste bibliothèque d'extensions pour améliorer les fonctionnalités de base ainsi que l'intégration native avec Git et d'autres systèmes de contrôle de version.

The screenshot shows the official Visual Studio Code download page. At the top, there's a navigation bar with links for 'Visual Studio Code', 'Docs', 'Updates', 'Blog', 'API', 'Extensions', 'FAQ', and 'Learn'. A 'Download' button is also present. Below the navigation, a message states 'Version 1.91 is now available! Read about the new features and fixes from June.' A search bar is located next to the download button. The main section is titled 'Download Visual Studio Code' with the sub-instruction 'Free and built on open source. Integrated Git, debugging and extensions.' Below this, there are three large download buttons for different operating systems: 'Windows', 'Linux', and 'Mac'. Each button has a corresponding icon (Windows logo, Tux logo, Apple logo) and a 'Download' link. Under each OS section, there are smaller download links for various architectures (x64, Arm64, etc.) and specific file types (.deb, .rpm, .tar.gz, Snap, CLI). The Mac section also includes links for Intel chip and Apple silicon.

Figure 1: Page de téléchargement de VS Code

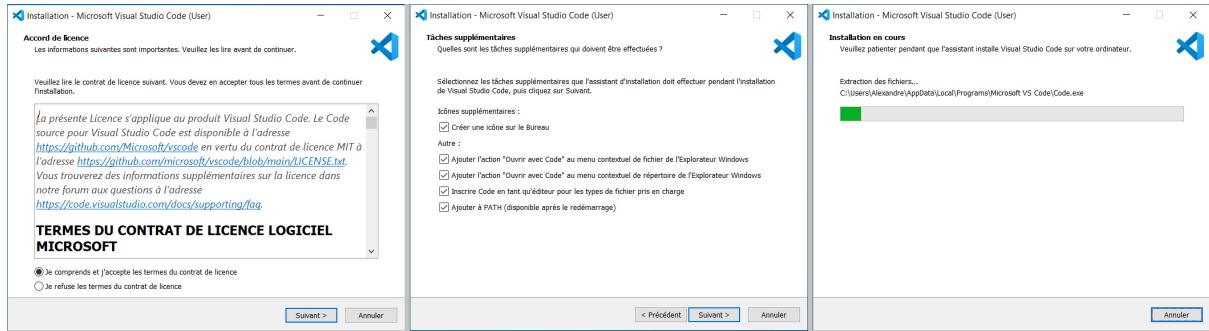
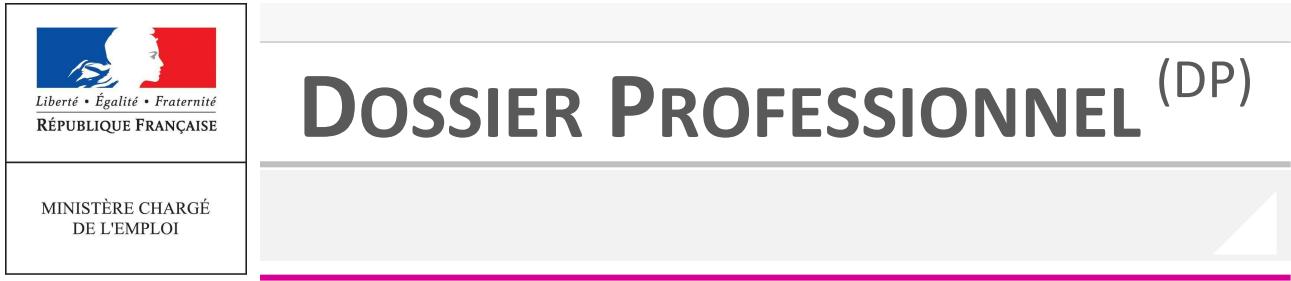


Figure 2: Différentes étapes de l'installation de VS Code

Sur le site officiel de Visual Studio Code ([code.visualstudio.com](https://code.visualstudio.com)) j'ai téléchargé la dernière version correspondant à mon système d'exploitation ( Windows ) et lancez l'installateur en suivant les instructions à l'écran.

Le choix des options d'installation telles que l'ajout de VS Code au PATH pour un accès en ligne de commande a été utilisé. Après la finalisation de l'installation, j'ai accès à VS Code sur la machines.

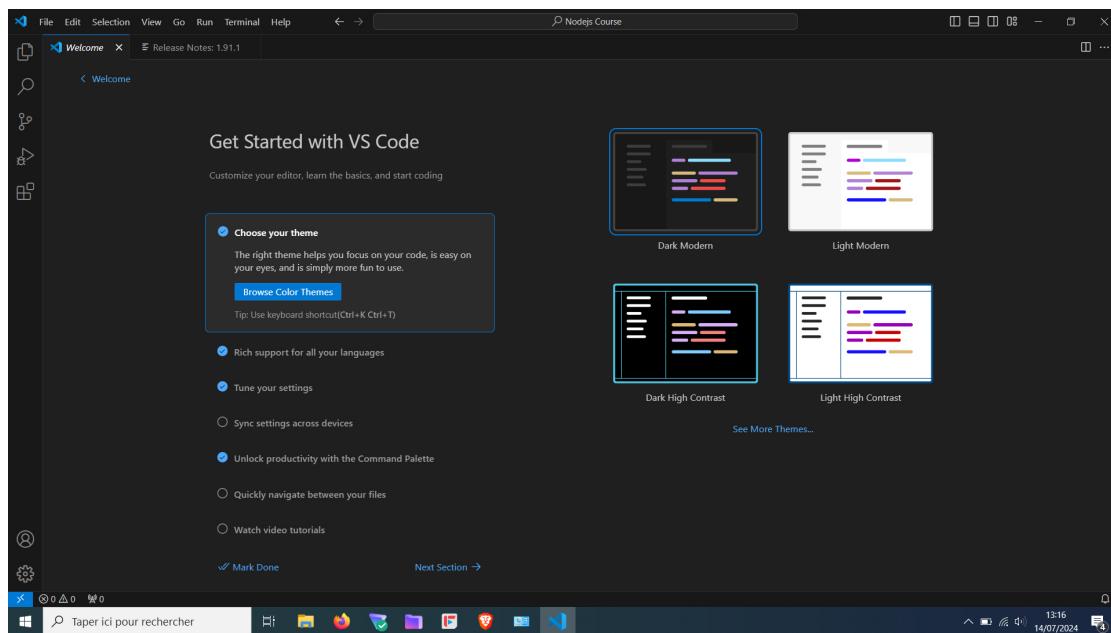
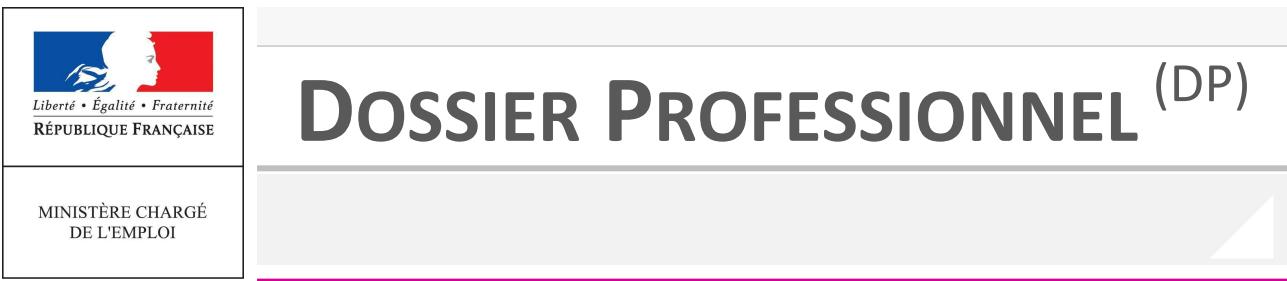


Figure 3: Page d'accueil du logiciel VS Code

Après l'installation j'ai personnalisé les paramètres selon mes préférences via Fichier > Préférences > Paramètres. L'installation d'extensions nécessaires pour le projet, comme les extensions pour Php, JavaScript, HTML, CSS, et d'aide au formatage de code comme l'extension Prettier ont été effectué.



# DOSSIER PROFESSIONNEL (DP)

## Installation et Configuration de Git

Git est un système de contrôle de version distribué, essentiel pour le suivi des modifications dans le code source tout au long du cycle de vie du développement. Ses principaux avantages incluent la gestion de version permettant de suivre l'historique des modifications, de gérer les versions et de collaborer efficacement avec d'autres développeurs tout en facilitant la gestion des branches pour le développement de fonctionnalités distinctes et leur fusion en une branche principale. J'utilise la plateforme collaborative GitHub lors du développement du projet.

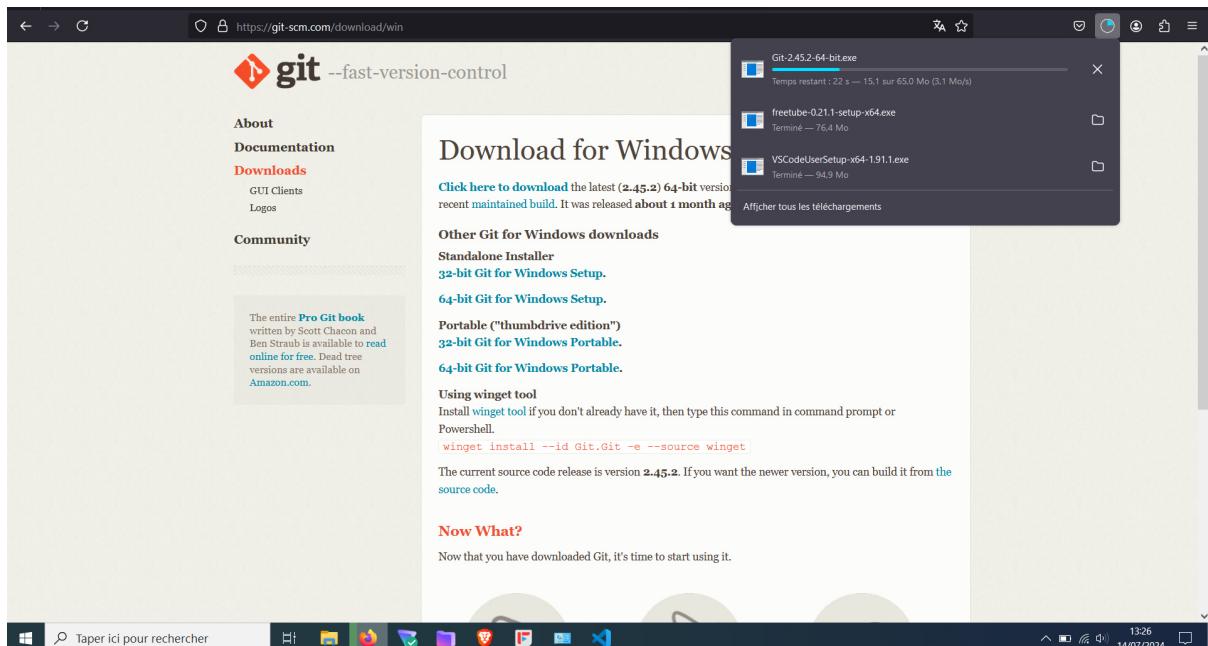


Figure 4: Téléchargement de Git sur le site officiel

Sur le site officiel de Git ([git-scm.com](https://git-scm.com)) j'ai pu télécharger la version 64-bit pour Windows, appropriée pour mon système d'exploitation puis exécuté l'installateur tout en suivant les étapes d'installation.

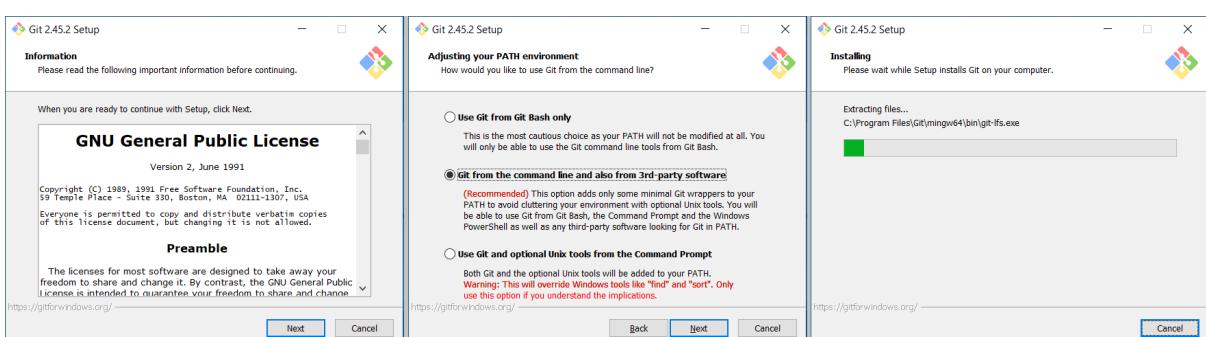


Figure 5: Différentes étapes d'installation de Git



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

Lors de l'installation, j'ai choisi de configurer les options telles que l'intégration de Git avec un éditeur de texte (comme VS Code). Après l'installation j'ai vérifiés en utilisant la ligne de commande Windows la version de Git téléchargé avec la commande git -v, celle-ci correspondant à la version 2.45.2.windows.1.

Invite de commandes

```
Microsoft Windows [version 10.0.19045.4412]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Alexandre>git -v
git version 2.45.2.windows.1

C:\Users\Alexandre>■
```

## Configuration de Git pour VS Code

J'ai ensuite configuré Git en utilisant l'interface de ligne de commande Git Bash de VS Code en rentrant les informations de configuration de l'utilisateur avec les commandes suivante :

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

Figure 6: Vérification de la version de Git

```
*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Alexandre@DESKTOP-O3R05D7.(none)')

Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (test)
● $ git config --global user.email "a.destroy@hotmail.fr"

Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (test)
○ $ git config --global user.name "Alexandre2383"■
```

Figure 7: Configuration de l'utilisateur avec Git Bash



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

Afin de tester la bonne configuration de Git sur la machine j'ai cloné mon repository Github créé dans le cadre de l'exercice Autocomplétion.

Invite de commandes

```
Microsoft Windows [version 10.0.19045.4412]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Alexandre>git -v
git version 2.45.2.windows.1

C:\Users\Alexandre>git clone https://github.com/Alexandre2383/autocompletion.git
```

```
Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (main)
$ git branch
* main
  test
●

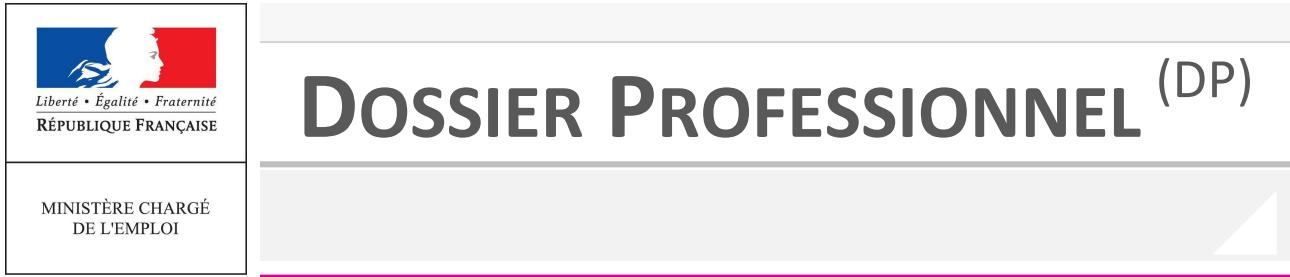
● Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (main)
$ git checkout test
Switched to branch 'test'

● Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (test)
● $ git add .

● Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (test)
● $ git commit -m "test de configuration git"
[test 3280da2] test de configuration git
  1 file changed, 1 insertion(+), 1 deletion(-)

● Alexandre@DESKTOP-O3R05D7 MINGW64 ~/autocompletion (test)
● $ git push --set-upstream origin test
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'test' on GitHub by visiting:
remote:     https://github.com/Alexandre2383/autocompletion/pull/new/test
remote:
To https://github.com/Alexandre2383/autocompletion.git
 * [new branch]      test -> test
branch 'test' set up to track 'origin/test'.
```

Figure 8: Série de commande de gestion de version



Puis j'ai effectué une création de branche appelée 'test' et je me suis connecté à celle-ci afin de réaliser un test de fonctionnement correct de version. J'ai ajouté un commentaire dans le fichier index.php et utilisé la commande git add, suivi d'un commentaire de version ( commit ) et finalement l'envoi de la nouvelle version tout en publiant la branche 'test'.

Enfin, je vérifie que l'envoi de la nouvelle branche avec sa version sur Github a été correctement effectué. Git est maintenant correctement configuré avec VS Code sur la machine.

The screenshot shows a GitHub repository interface. At the top, there is a header with a profile icon, the repository name 'autocomplete' (marked as 'Private'), and a 'Unwatch' button. Below the header, a yellow banner indicates a recent push to the 'test' branch: 'test had recent pushes 1 minute ago' with a 'Compare & pull request' button. The main area shows a list of branches ('main', '3 Branches', '0 Tags') and tags. A modal window titled 'Switch branches/tags' is open, showing a search bar and a list of branches: 'main' (selected), 'dev', and 'test'. On the right side of the interface, a list of commits from user 'Alexandre2383/dev' is displayed, all made 3 weeks ago. The commits are: 'Création de la connexion à la bdd et développement de la cl...', 'modif read.md', 'Création de la connexion à la bdd et développement de la cl...', 'Finalisation du projet autocompletion par l'ajout des logique...', 'Finalisation du projet autocompletion par l'ajout des logique...', and 'Finalisation du projet autocompletion par l'ajout des logique...'. Each commit includes a timestamp ('3 weeks ago') and a '6 Commits' link.

*Figure 9: Indication de Github d'une récente publication*

### Installation et Configuration de Laragon

Laragon est un environnement de développement léger, rapide et portable pour PHP, Node.js, Python et autres. Il est conçu pour être simple à utiliser, tout en offrant des fonctionnalités puissantes. Le serveur local est facile à configurer pour exécuter les projets localement et il intègre un gestionnaire de base de données comme MySQL/MariaDB et d'autres SGBD.

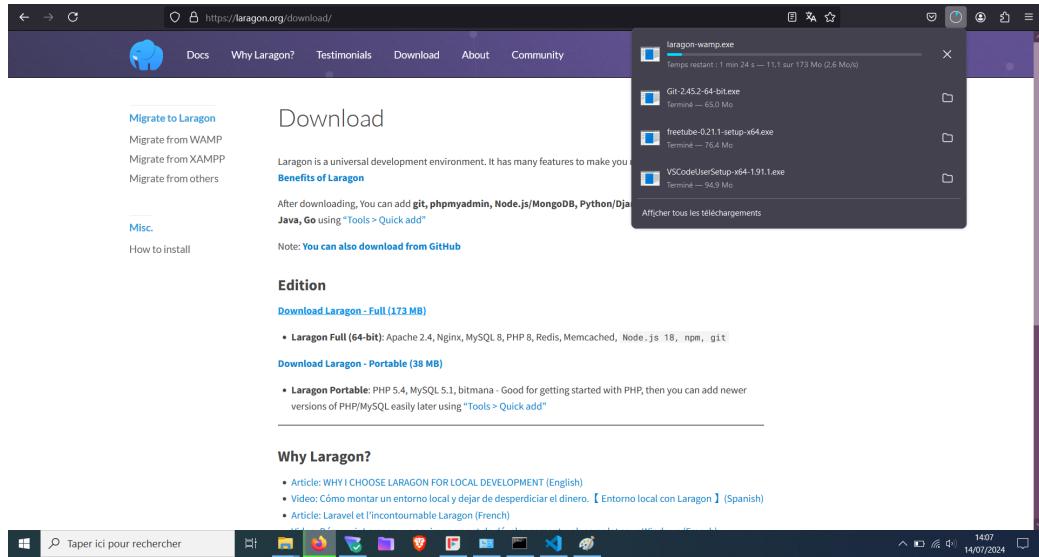
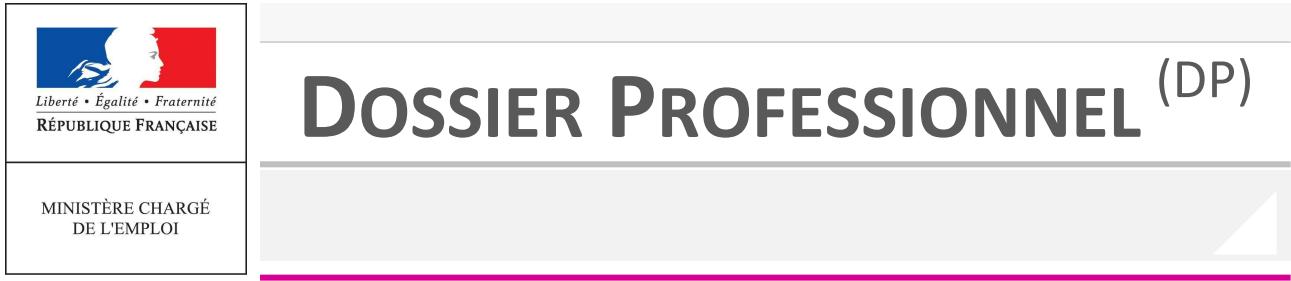


Figure 10: Téléchargement de Laragon sur le site officiel

Sur le site officiel de Laragon ([laragon.org](https://laragon.org)) j'ai téléchargé l'installateur correspondant au système d'exploitation Windows et exécutez l'installation en suivant les instructions à l'écran tout en choisissant le répertoire d'installation et les composants à installer.

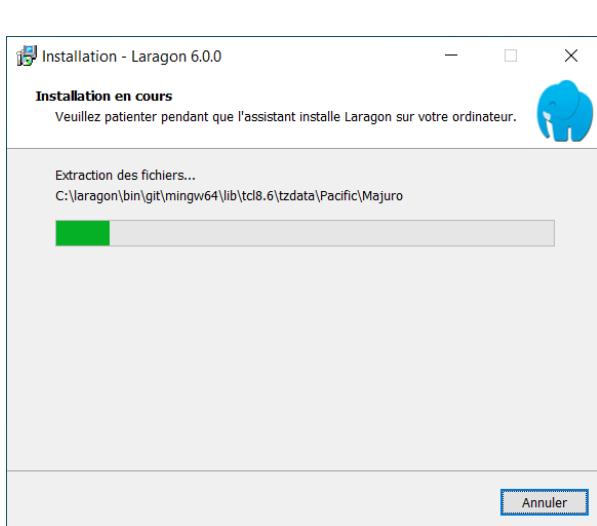


Figure 12: Installation de Laragon

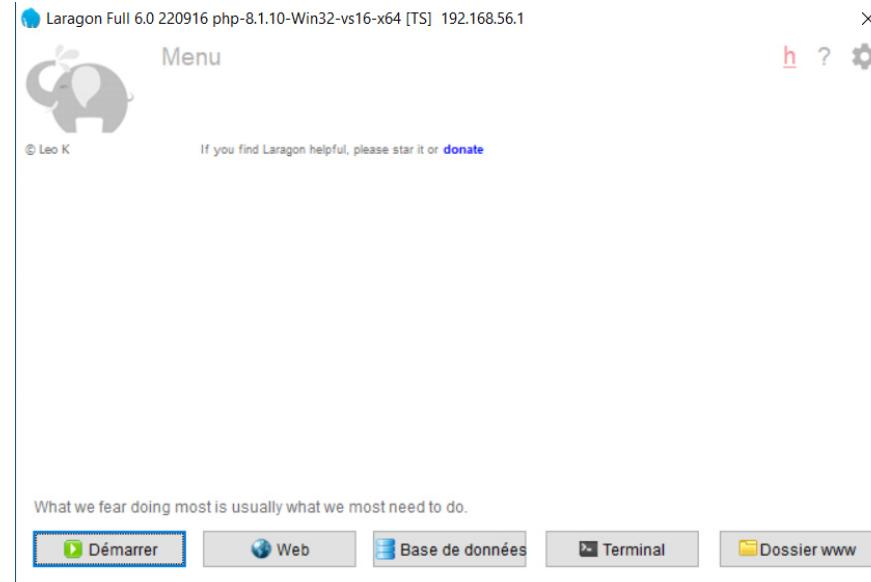


Figure 11: Panneau de gestion de Laragon



# DOSSIER PROFESSIONNEL (DP)

Après l'installation un redémarrage de la machine est nécessaire pour permettre à Laragon de s'installer correctement. Le panneau de démarrage de Laragon ouvert, celui-ci permet plusieurs configuration notamment des versions de Php, Apache ou Nodejs. Afin de tester le bon fonctionnement du serveur en local, je dépose le projet Autocomplétion dans le dossier C:\laragon\www et j'accéde à l'hôte local avec l'URL <http://localhost/>, celui-ci affichant le fichier index.php du projet.

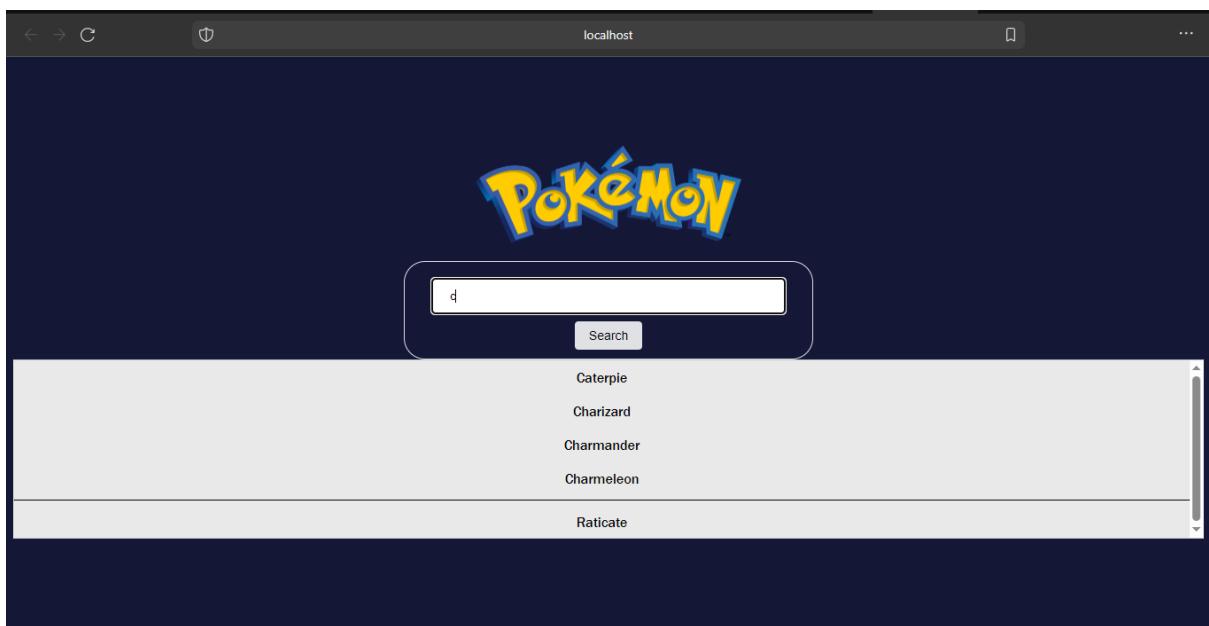


Figure 13: Index du projet autocomplétion

L'installation et la configuration de VS Code, Git et Laragon constituent une base solide pour le développement du projet. VS Code offre un environnement de codage puissant et personnalisable, Laragon fournit un serveur local complet pour tester et développer le projet, et Git assure un suivi et une gestion efficace des versions du code source.

## 2. Précisez les moyens utilisés :

Pour réalisé l'ensemble de l'installation et de la configuration de VS Code, Git et Laragon je me suis servi de leur documentation respective et d'un deuxième ordinateur pour tester efficacement le versionnage du projet.



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

## 4. Contexte

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

Chantier, atelier, service ► *Autocompletion*

Période d'exercice ► Du : *03/06/2024* au : *08/06/2024*

## 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 sécurisée.

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

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

#### Introduction

Dans le cadre de la formation, il nous a été demandé de réaliser un Portefolio Responsive avec pour objectif de souligner l'importance de l'identité numérique dans le domaine du développement web. J'ai conçu les wireframes et les user stories à partir des consignes de l'exercice.

#### User Stories

Les user stories sont des scénarios décrivant les besoins des utilisateurs. Ce sont des descriptions brèves et claires des attentes d'un utilisateur de l'application. Elles sont formulées ainsi : "En tant que [rôle], je veux [objectif] afin de [motif]." Chaque user story est autonome et détaille une fonctionnalité spécifique. Les user stories assurent que les fonctionnalités développées répondent bien aux attentes des utilisateurs et que leurs besoins sont correctement interprétés.

En tant que	Je veux	Afin de
Utilisateur	Accéder à la page d'accueil	Accéder au contenu général du site
Utilisateur	Accéder à la section Présentation	Consulter les informations général du portefolio ( qui est le créateur, quels sont ses expériences )
Utilisateur	Accéder à la section Langages & Compétences	Consulter les différents langages utilisé par le créateur du portefolio et ses compétences
Utilisateur	Accéder à la section Portefolio	Visualiser et tester les différents projets posté
Utilisateur	Accéder à la section Contact	Pouvoir rechercher le créateur et prendre contact

#### Wireframes

Les wireframes sont des maquettes en nuances de gris, annotées et épurées d'une



# DOSSIER PROFESSIONNEL (DP)

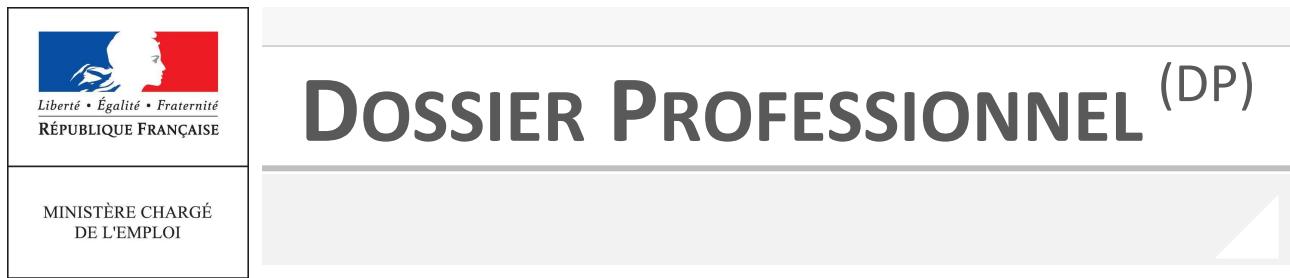
application. Elles ne contiennent aucun choix graphique et illustrent de manière schématique la structure des pages. Pour ce projet, j'ai d'abord réalisé les wireframes en mobile-first. Pour réaliser la partie maquette j'ai utilisé l'outil de conception Figma.



Figure 14: Accès page d'accueil Mobile



Figure 15: Ouverture du menu burger sur Mobile

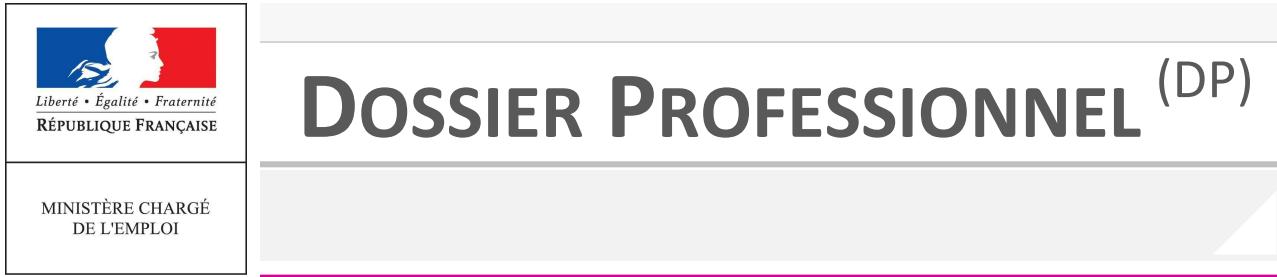


Lors de l'accès à la page d'accueil du portefolio la section 'Home' est affiché, composé d'une barre de navigation ayant un menu burger, d'un message d'accueil de la part de l'auteur et d'une image d'arrière-plan.



*Figure 17: Section Langages et compétences*

*Figure 16: Section Présentation*



La section



Figure 19: Section Contact

Présentation contient une photo de l'auteur du portefolio et d'une description. Puis la section est suivie d'un tableau des expériences récentes permettant d'appuyer la description personnel. La section Langages et Compétences est conçu par carte de compétences, cela permet de mieux détailler et d'avoir la possibilité d'afficher des images appuyant la description.

Figure 18: Section Portefolio



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

J'ai repris le même principe pour la section Portefolio, avec en plus l'ajout de bouton permettant d'accéder à la page du projet sélectionné. Enfin la section Contact est composé d'une description sur les moyens de contacter l'auteur en utilisant les liens qui s'en suit. Puis un détail des sources potentiel et l'insertion du copyright. Suivant le même principe de la conception des Wireframes au format mobile, j'ai réalisé ensuite la conception au format Desktop.

### Langages & Compétences

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.  
Sed ut persiciatis und Sed ut persiciatis. Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium ut spiciatis.

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.  
Sed ut persiciatis und Sed ut persiciatis. Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium ut spiciatis.

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.  
Sed ut persiciatis und Sed ut persiciatis. Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium ut spiciatis.

### Portefolio

**Image de projet**

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.

**Button**

**Image de projet**

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.

**Button**

**Image de projet**

Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque Sed ut persiciatis und Sed ut persiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.

**Button**

### Contact

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

liens https

liens https

liens https

Source      "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Copyright

Figure 21: Section Langages et Compétences, Portefolio et Contact



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

La réalisation de ces wireframes a été une étape cruciale dans la conception du portefolio. Elles ont permis de visualiser et de planifier l'apparence et la fonctionnalité du site avant de passer à l'étape de développement.

## 2. Précisez les moyens utilisés :

En basant sur le cours du formateur et la chaîne Youtube Alienai Cai, j'ai conçu la maquette en utilisant l'outil Figma.

## 3. Avec qui avez-vous travaillé ?

'ai travaillé seul sur ce projet.

## 4. Contexte

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

Chantier, atelier, service ► *Site Perso*

Période d'exercice ► Du : *03/06/2024* au : *04/06/2024*

## 5. Informations complémentaires (facultatif)

La maquette pour la partie mobile est accessible à partir de ce lien :

<https://www.figma.com/design/lf7rn1d7nYKK2o0orGXnfy/Portfolio-Mobile?node-id=0-1>

La maquette pour la partie Desktop est accessible à partir de ce lien :

<https://www.figma.com/design/f5Q6Lbg5wFvApmieMeCmxu/Portefolio-Desktop?node-id=0-1>



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Activité-type 1

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

*CP 3 ▶ Réaliser des interfaces utilisateurs statiques web ou web mobile.*

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

#### Introduction

Après avoir défini un dossier de conception clair et structuré, comprenant une maquette mobile et desktop ainsi que l'utilisation de Bootstrap pour la partie responsive, j'ai entrepris de réaliser les interfaces utilisateur web et mobile de mon portfolio. Mon objectif principal était de tenir compte de l'expérience utilisateur (UX) afin de garantir une navigation intuitive et agréable. Je décide d'utiliser des éléments de Bootstrap pour chaque section de mon portfolio afin d'offrir une interaction fluide, avec une attention particulière portée à l'accessibilité et à la clarté de l'information.

#### La balise de navigation et de header

La balise `<nav>` est utilisée pour créer une barre de navigation fixe en haut de la page (`fixed-top`). Elle permet aux utilisateurs de naviguer facilement entre les différentes sections du portfolio en fournissant des liens clairs et accessibles.

```
<nav class="navbar navbar-expand-lg fixed-top">
  <div class="container">
    <a id="anchor_title" class="navbar-brand" href="#home">Home</a>
    <button
      class="navbar-toggler"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <span id="nav_burger" class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item">
          <a id="anchor_title" class="nav-link" href="#about">Présentation</a>
        </li>
        <li class="nav-item">
          <a id="anchor_title" class="nav-link" href="#langages">Langages & Compétences</a>
        </li>
        <li class="nav-item">
          <a id="anchor_title" class="nav-link" href="#portfolio">Portfolio</a>
        </li>
        <li class="nav-item">
          <a id="anchor_title" class="nav-link" href="#contact">Contact</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Les Classes de la balise `<nav>`: `navbar`, `navbar-`

Figure 22: Barre de navigation



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## expand-lg et fixed-top

La classe navbar est une classe Bootstrap qui applique les styles de base d'une barre de navigation (<https://getbootstrap.com/docs/5.3/components/navbar/#how-it-works>). Navbar-expand-lg permet à la barre de navigation de se déployer sur des écrans larges, mais de se replier sur des écrans plus petits, garantissant ainsi la compatibilité mobile. L'utilisation de fixed-top permet de fixer la barre de navigation en haut de la page, assurant qu'elle reste visible lorsque l'utilisateur défile vers le bas de la page. La couleur de background sera définie par l'utilisation d'un fichier CSS (appelé 'style.css') appliquant un fond de couleur noir.

## Bouton de Toggle pour Navigation Mobile

Le bouton avec les classes navbar-toggler et navbar-toggler-icon permet de replier et de déplier la barre de navigation sur les appareils mobiles. Cela assure une meilleure expérience utilisateur sur les écrans de petite taille.

Les attributs data-bs-toggle="collapse" et data-bs-target="#navbarSupportedContent" contrôlent le comportement du bouton, permettant de masquer ou d'afficher le contenu de la navigation. En utilisant les classes et les attributs ARIA appropriés, la barre de navigation est accessible pour les utilisateurs de lecteurs d'écran.

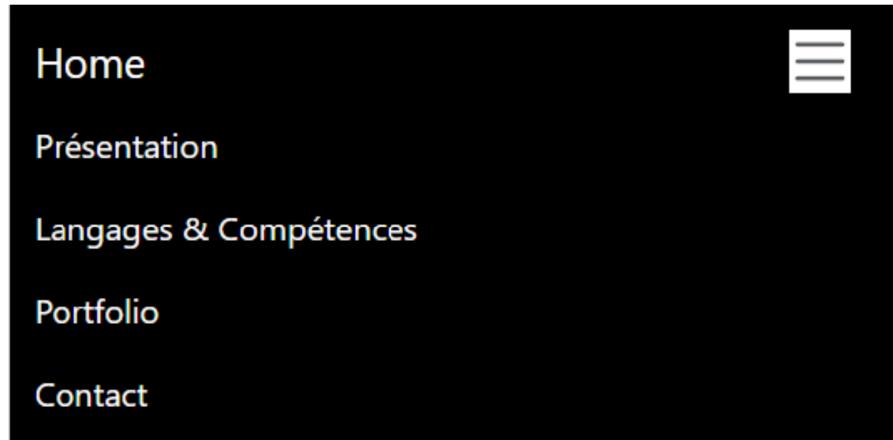
```
nav {  
    background-color: #000000;  
}
```



Figure 23: Barre de navigation pour support supérieur à 991px

## Liens

Figure 24: Menu Burger pour support inférieur ou égal à 991px





# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## de Navigation

Chaque élément de navigation est un élément de liste (<li>) à l'intérieur d'un conteneur <ul>. Les liens (<a>) sont stylisés avec la classe nav-link de Bootstrap. L'identifiant id="anchor\_title" est utilisé pour styliser uniformément les liens. Les liens dirigent l'utilisateur vers des sections spécifiques de la page (#about, #langages, #portfolio, #contact).

### Conteneur div

```
<header class="bgimage" id="home">
  <article class="container-fluid">
    <div class="row">
      <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 hero-text">
        <h2 class="hero_title">Salut, je suis Alex!</h2>
        <p class="hero_desc">Développeur Apprenant habitant en France</p>
      </div>
    </div>
  </article>
</header>
```

Figure 25: Header

Le conteneur <div class="collapse navbar-collapse" id="navbarSupportedContent"> contient les éléments de la navigation et assure qu'ils sont repliables sur les appareils mobiles.

La balise <header> est utilisée pour définir la section d'en-tête du portfolio. Cette section est essentielle pour introduire l'utilisateur à la page et fournir une première impression visuelle grâce à une image de fond, appliquer par la classe bgimage, et des informations courtes par l'utilisation d'un titre suivi d'une description courte.

Cette section d'en-tête a été conçue pour captiver les visiteurs, leur fournissant une introduction claire et accueillante au portfolio tout en s'adaptant parfaitement à différents dispositifs grâce à l'utilisation de Bootstrap.

### Identifiant home

L'attribut id="home" permet de cibler cette section avec des liens de navigation. Cela permet à l'utilisateur de revenir facilement au début de la page en cliquant sur un

```
.bgimage {
  height: 100vh;
  background: url('images/mountains-wallpaper.jpg');
  background-size: cover;
  position: relative;
}

.hero_title {
  font-size: 4.5rem;
}

.hero_desc {
  font-size: 2rem;
}

.hero-text {
  text-align: center;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  color: white;
  width: 100rem;
}
```



# DOSSIER PROFESSIONNEL (DP)

lien correspondant dans la barre de navigation.

## Conteneur article avec container-fluid

L'utilisation de `<article class="container-fluid">` garantit que le contenu de l'en-tête s'étend sur toute la largeur de l'écran, offrant une mise en page fluide qui s'adapte bien à différentes tailles d'écran.

```

<section id="about">
  <div class="container mt-4 pt-4">
    <h1 class="text-center">Présentation</h1>
    <div class="row mt-4">
      <article class="col-lg-6">
        <div class="conteneur_image_presentation">
          
        </div>
      </article>
      <article class="col-lg-6">
        <div class="card servicesText">
          <div class="card-body">
            <p class="card-text mt-3">
              Je m'appelle Alexandre, je suis passionné par le développement web et les nouvelles technologies, je suis un apprenant développeur en création de sites web et d'applications interactives. Après avoir commencé ma carrière en tant que mécanicien automobile chez Volkswagen, j'ai découvert ma véritable passion pour le développement web en 2023. Depuis, je me suis pleinement consacré à cette nouvelle voie en suivant une formation de développeur web à
            <a href="https://laplateforme.io/qui-sommes-nous/" target="_blank">La Plateforme</a>
            .
          </p>
        </div>
      </div>
    </div>
  </div>
<article id="conteneur_experience" class="row">
  <h2 class="titre_experience">Mes récentes expériences</h2>
  <table class="table table-hover">
    <thead>
      <tr>
        <th scope="col" colspan="2"></th>
        <th scope="col">Formation et Entreprise</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row">MARS 2024 --</th>
        <td>
          <p>La Plateforme - Développeur Web et Web Mobile</p>
          <p>Formation DWWM au sein de l'école La Plateforme_. Apprentissage au développement de site web et d'application : PWA, SPA et fullstack.</p>
        </td>
        <td>
          <p>La Plateforme - Développeur Logiciel</p>
          <p>Stage au Centre de Recherche en Psychologie et Neurosciences d'Aix-Marseille. Développement d'un logiciel Java d'analyse auditive embarqué (Arduino, DAO).</p>
        </td>
      </tr>
      <tr>
        <th scope="row">FEV 2024</th>
        <td>
          <p>CRPN Marseille - Développeur Logiciel</p>
          <p>Stage au Centre de Recherche en Psychologie et Neurosciences d'Aix-Marseille. Développement d'un logiciel Java d'analyse auditive embarqué (Arduino, DAO).</p>
        </td>
        <td>
          <p>NOV 2023 - FEV 2024</p>
          <td colspan="2">
            <p>La plateforme - Développeur Web</p>
            <p>Formation découverte de la programmation et du développement de site Web.</p>
          </td>
        </td>
      </tr>
      <tr>
        <th scope="row">SEPT 2018 - SEPT 2023</th>
        <td colspan="2">
          <p>Volkswagen - Mécanicien Automobile</p>
          <p>Mécanicien atelier, maintenance et diagnostique des véhicules. Formation interne à la maintenance des véhicules électriques Volkswagen.</p>
        </td>
      </tr>
    </tbody>
  </table>
</article>
</div>
</section>

```

Figure 26: Section à propos

## Disposition en Grille avec row et col- Classes



# DOSSIER PROFESSIONNEL (DP)

Le conteneur `<div class="row">` organise le contenu en ligne, tandis que `<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 hero-text">` s'assure que le texte de l'héro est centré et occupe toute la largeur disponible, quelle que soit la taille de l'écran.

## La balise section about

La balise `<section id="about">` du portfolio est dédiée à ma présentation. Elle combine des éléments visuels et textuels pour fournir une vue d'ensemble de mon parcours personnel, ma formation, et mes expériences professionnelles récentes afin d'informer efficacement les visiteurs.

Le titre principal "Présentation" est centré de manière clairement identifiable afin d'introduire la section.

## Conteneur Principal avec container

La classe container de la balise `<div class="container mt-4 pt-4">` assure que la section a des marges appropriées, et les classes `mt-4` et `pt-4` ajoutent des marges supérieures pour espacer cette section des autres éléments de la page.

## Disposition en Deux Colonnes

La première colonne `<article class="col-lg-6">` contient une photo, qui aide à personnaliser la section et à rendre la présentation plus visuelle. L'image `` est affichée avec une classe `imageAboutPage` pour la styliser correctement.

La deuxième colonne `<article class="col-lg-6">` contient une description personnelle rapide de ma situation actuelle contenu dans la balise `<div class="card servicesText">` utilisant la classe `card` pour encadrer la description, ajoutant un élément visuel propre et organisé.

```
.imageAboutPage {  
    max-width: 300px;  
    margin: auto;  
    width: 100%;  
    border-radius: 25px;  
}  
.conteneur_image_presentation {  
    width: 100%;  
    display: flex;  
}  
.titre_experience {  
    text-align: center  
}
```

## L'article des Expériences

Le sous-titre "Mes récentes expériences" contenu dans la balise `<h2 class="titre_experience">Mes récentes expériences</h2>` introduit la liste des expériences professionnelles.

L'utilisation d'un tableau `<table class="table table-hover">` avec la classe `table-hover` permet d'afficher les expériences professionnelles de manière structurée, composé d'un entête `<thead>` contenant les titres des colonnes du tableau, avec une colonne pour les dates et une pour les descriptions des formations et entreprises. Suivi du corps du tableau `<tbody>` listant les différentes expériences professionnelles avec des dates et des descriptions détaillées.



# DOSSIER PROFESSIONNEL (DP)

```

<section id="languages">
  <div class="container">
    <h1 class="text-center">Langages & Compétences</h1>
    <div class="row">
      <article class="col-lg-4 mt-4">
        <div class="card languagesText">
          <div class="card-body">
            <div class="frontend_card">
              
              
              
              
              
            </div>
            <h4 class="card-title mt-3">Front-end</h4>
            <p class="card-text mt-3">
              Compétence en HTML/CSS, JavaScript ainsi que Tailwind pour des fonctionnalités dynamiques et interactives. Apprentissage de React pour le développement d'applications web modernes et performantes, assurant une expérience utilisateur optimale.
            </p>
          </div>
        </div>
      </article>
      <article class="col-lg-4 mt-4">
        <div class="card languagesText">
          <div class="card-body">
            <div class="frontend_card">
              
              
              
            </div>
            <h4 class="card-title mt-3">Back-end</h4>
            <p class="card-text mt-3">
              Expérience PHP pour le développement de sites web dynamiques. Compétence en Node.js pour des applications serveur performantes et évolutives. Apprentissage de Python pour des scripts efficaces et le développement de solutions backend utilisant FastAPI.
            </p>
          </div>
        </div>
      </article>
      <article class="col-lg-4 mt-4">
        <div class="card languagesText">
          <div class="card-body">
            <div class="frontend_card">
              
              
              
            </div>
            <h4 class="card-title mt-3">Outils et Collaboration</h4>
            <p class="card-text mt-3">
              Utilisation de Git pour un contrôle de version efficace et des flux de travail collaboratifs, ainsi que de GitHub pour la gestion de projets et la collaboration en équipe. Compétence pour la conception UI/UX et le prototypage interactif avec Figma.
            </p>
          </div>
        </div>
      </article>
    </div>
  </div>

```

Figure 27: Section langages et compétences du portefolio



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## La balise section langages

Cette section "Langages & Compétences" est conçue pour offrir une présentation détaillée et bien organisée de mes compétences techniques, en mettant l'accent sur la clarté et l'accessibilité. Elle est organisée par thème, en reprenant l'utilisation de la classe card de Bootstrap. Cette structure permet de montrer plus clairement les différentes technologies et compétences acquises, accompagnées d'icônes visuelles.

## Disposition en Trois Colonnes

Chaque compétence est présentée dans une colonne distincte, assurant une lecture fluide et une présentation claire. Chaque balise article `<article class="col-lg-4 mt-4">` contient une carte présentant les compétences et langages utilisés défini par le titre de la carte, `<h4 class="card-title mt-3">Front-end</h4>` par exemple pour le thème "Front-end", indiquant clairement les compétences présentées. Chaque carte contient des icônes `` représentant les technologies maîtrisées.

## Intérêt des classes col et row

L'utilisation des classes row et col de Bootstrap permet de créer des mises en page réactives, bien alignées, et modulaires. Ici chaque colonne occupe 4 unités de la grille (12 unités au total), créant trois colonnes égales sur les écrans larges, permettant une disposition uniforme et esthétique du contenu. En ajoutant une nouvelle ligne avec une classe row, une nouvelle série de cartes peut être ajoutée en dessous sans perturber la disposition actuelle assurant que les colonnes enfants s'alignent horizontalement. Cela crée un conteneur avec des marges négatives sur les côtés pour compenser les marges des colonnes, alignant ainsi les colonnes avec le reste du contenu.

```
<div class="row">
    <article class="col-lg-4 mt-4">
        //...
```



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

```
<section id="portfolio">
  <div class="container mt-3">
    <h1 class="text-center">Portfolio</h1>
    <div class="row">
      <article class="col-lg-4 mt-4">
        <div class="card">
          
          <div class="card-body">
            <h4 class="card-title">Pokédex</h4>
            <p class="card-text">
              Pokédex des 151 premiers Pokémon développé avec Angular,
              utilisation de l'API
              <a href="https://pokebuildapi.fr/api/v1" target="_blank">
                Pokebuild</a>
              >
              et implémentation de la Game Boy avec l'exemple
              <a href="#">CodePen</a>. Non fini.
            </p>
            <div class="text-center">
              <a href=".//Pokedex/index.html" class="btn btn-primary">
                Voir</a>
              >
            </div>
          </div>
        </div>
      </article>
    </div>
  </div>
</section>
```

Figure 28: Section portefolio

```
<section id="contact">
  <div class="container mt-3 contactContent">
    <h1 class="text-center">Me Contacter</h1>

    <div class="row mt-4">
      <article class="col-lg-6">
        <div class="card langagesText">
          <div class="card-body">
            <p class="card-text mt-3">
              Je suis toujours à la recherche de nouvelles opportunités pour
              collaborer sur des projets stimulants et à contribuer dans le
              domaine du développement. N'hésitez pas à me contacter soit
              par les plateformes Github et Linkedin, soit par Mail pour
              discuter de potentielles collaborations.
            </p>
          </div>
        </div>
      </article>
    </div>
  </div>
</section>
```

Figure 29: Section contact



# DOSSIER PROFESSIONNEL (DP)

```
<footer class="bg-body-tertiary text-center text-md-start">
  <div class="container p-4">
    <div class="row">
      <div class="col-lg-10 col-md-12 mb-4 mb-md-0">
        <h5 class="text-uppercase">Source</h5>
        <p>
          Mes ressources dans la réalisation de ce portfolio :
          <a href="https://www.wallpaperflare.com/static/872/157/141/mountains-asus-zenfone-stock-hd-wallpaper.jpg">Image de background de la section Home</a>, <a href="https://getbootstrap.com" target="_blank">Bootstrap pour la mise en page</a> et <a href="https://icons8.com" target="_blank">Icon8 pour les logos.</a></p>
        </div>
      </div>
    </div>
    <div class="text-center p-3">
      © 2023 Copyright : Ce portefolio est libre de droit.
    </div>
  </div>
</footer>
```

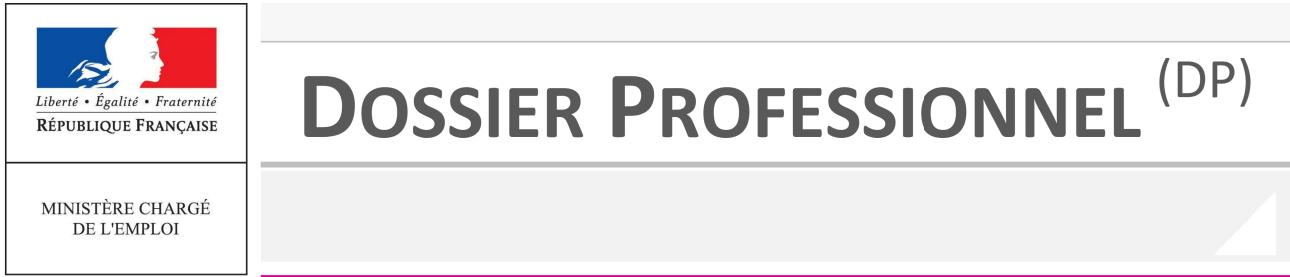
Figure 30: Section footer

## La balise section portfolio, contact et la balise footer

Les sections `<section id="portfolio">` et `<section id="contact">`, ainsi que la balise footer `<footer class="bg-body-tertiary text-center text-md-start">` du portfolio sont dédiées à la présentation des projets réalisés, à la description et aux moyens de contact, suivi des sources utilisés. Ces sections du Portfolio ne seront pas détaillées car elles reprennent les principes des classes container, col et row, ainsi que card afin de générer de la même manière que la section langages et la section about une mise en valeur des projets et de la partie contact de manière structurée et visuelle.

## Référencement

Pour rendre mon portfolio accessible au public, j'ai publié les pages web statiques de manière sécurisée sur un serveur web. J'ai utilisé Plesk pour le déploiement, assurant une configuration sécurisée et fiable.



This screenshot shows the Plesk control panel interface. On the left is a sidebar with various management options like Websites & Domains, Mail, Applications, Files, Databases, Statistics, Users, Account, WordPress, and SEO Toolkit. The main area displays the dashboard for the website `alexandre-detroy.students-laplateforme.io`. The dashboard includes a preview of the website showing a snowy mountain landscape with the text "Salut, je suis Alex! Développeur Apprendant habitant en France". It also shows disk usage (0 MB), traffic (0 MB/month), and a link to Web Statistics SSL/TLS. On the right, there are sections for Files & Databases, Dev Tools, and Security, each with several sub-options.

En parallèle, j'ai optimisé le référencement (SEO) de mon portfolio pour le rendre visible sur les moteurs de recherche en incluant des balises métas pertinentes

```

<meta name="description" content="Student Portfolio Alexandre" />
<link rel="canonical" href="https://alexandre-detroy.students-laplateforme.io/" />
<!-- META SEO OG -->
<meta property="og:image" content="./images/image-seo.jpg" />
<meta property="og:image:width" content="1336" />
<meta property="og:image:height" content="614" />
<meta property="og:image:alt" content="Aperçu du portfolio d'Alex, développeur web présentant des projets en front-end et back-end" />
<meta property="og:type" content="content" />
<meta property="og:url" content="https://alexandre-detroy.students-laplateforme.io/" />
<meta property="og:title" content="Student Portfolio" />
<meta property="og:description" content="My student Portfolio Website, create during this years of study" />

<title>Portfolio Website Alex</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" href="style.css" />

```

Figure 31: Ajout des balises meta de référencement

L'utilisation d'une description et d'un lien canonical permet aux moteurs de recherche d'utiliser la description pour afficher un résumé de la page dans les résultats de recherche et de spécifier l'adresse principale de la page, aidant les moteurs de recherche à comprendre quelle URL utiliser pour indexer le contenu. Les balises ayant og:image en propriété définissent l'image utilisée lors du partage de la page sur les réseaux sociaux, de sa taille en largeur et hauteur ainsi que de la description de celle-ci.



# DOSSIER PROFESSIONNEL (DP)

Les propriétés de type, d'url, titre et description permettent de spécifier pour les réseaux sociaux le type de contenu de la page, l'URL de la page web, le titre et une description de la page.

Après avoir mis le site en ligne, j'effectue un test d'analyse par l'intermédiaire du site PageSpeed Insights (<https://pagespeed.web.dev/>) afin de vérifier les performances, l'accessibilité, les bonnes pratiques et le SEO.

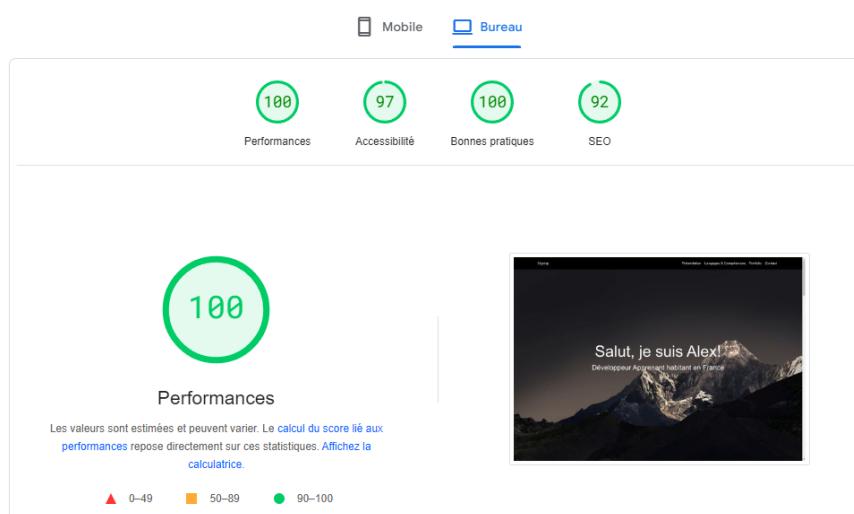


Figure 32: Résultat d'analyse du portfolio par PageSpeed Insights

## 2. Précisez les moyens utilisés :

J'ai utilisé Bootstrap pour certains composants, en me référant à la documentation ainsi que celle de MDN Web Docs. J'ai utilisé le site de partage d'icônes Icons8 (<https://icons8.com/>) ainsi que wallpaperflare (<https://www.wallpaperflare.com>) pour les images. Le portfolio est accessible à partir de l'URL <https://alexandre-detroy.students-laplateforme.io/>.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_*

Chantier, atelier, service ► *Site Perso*

Période d'exercice ► Du : *03/06/2024* au : *07/06/2024*

## 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 sécurisée.

**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 :

### Introduction

Au cours de la formation nous avons eu à réaliser une interface utilisateur composé d'une horloge, d'un minuteur, d'un chronomètre et d'une alarme. J'ai développé en parallèle la possibilité pour l'utilisateur d'obtenir la météo pour mettre d'avantage l'accent sur la création d'une interface utilisateur dynamique et interactive.



Le code HTML est structuré pour afficher quatre sections principales correspondant à chaque fonction de l'application : Horloge, Minuteur, Chronomètre et Alarme. Chaque section est initialement masquée sauf l'horloge, qui est affichée par défaut. Les utilisateurs peuvent naviguer entre ces sections via une barre de navigation. Dans cette présentation, nous nous concentrerons



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

sur la section Horloge et sa fonctionnalité de Météo. Présentation du code HTML de la section Horloge et Météo

La section HTML ci-dessous combine les fonctionnalités d'une horloge numérique et d'une prévision météorologique. Elle sépare clairement les différentes fonctionnalités et fournit une base solide pour l'interaction JavaScript.

```
<section id="clockSection" class="clock">
  <div class="card">
    <div class="container_title">
      
      <h1 class="title">Clock</h1>
    </div>
    <div><p id="clockDisplay" class="numerique_clock"></p></div>
    <div class="container_weather">
      <div class="">
        <h2>Actual Weather</h2>
        <label for="searchButton">
          <input
            type="text"
            id="locationInput"
            placeholder="Enter a city"
          />
          <button id="searchButton" class="search">Search</button>
        </label>
      </div>
      <div class="weather_info">
        <h2 id="location"></h2>
        <img
          id="weatherIcon"
          src=""
          alt="Weather Icon"
          class="weather_icon"
        />
        <div class="weather_info">
          <p id="temperature"></p>
          <p class="p_humidity">humidity<span id="humidity"></span></p>
        </div>
        <p id="description"></p>
      </div>
    </div>
  </div>
</section>
```

Figure 33: Section html horloge et météo



# DOSSIER PROFESSIONNEL (DP)

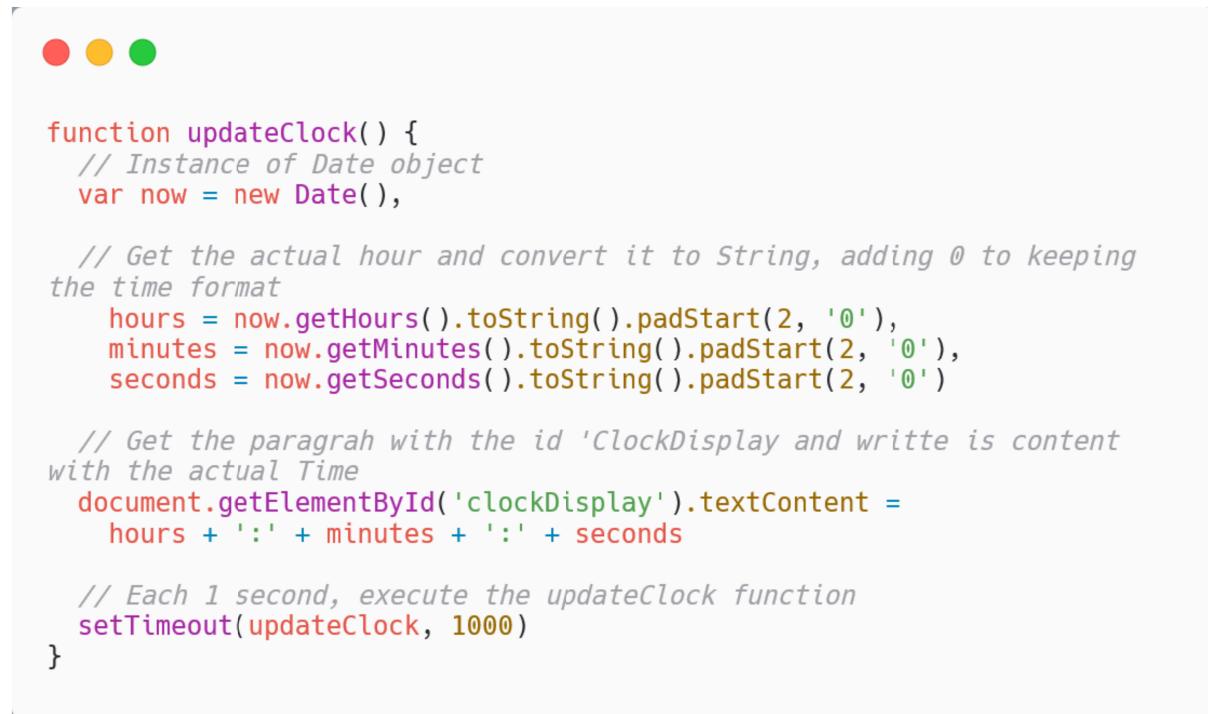
MINISTÈRE CHARGÉ  
DE L'EMPLOI

La balise <section> avec l'id="clockSection" et la class="clock" définit une section de la page dédiée à l'affichage de la partie horloge. L'utilisation de id permet une identification unique et facilite la manipulation de cette section via JavaScript.

L'horloge est contenu dans la balise <div><p id="clockDisplay" class="numerique\_clock"></p></div> et affiché par le paragraphe avec l'id="clockDisplay". L'utilisation de l'ID permet de cibler cet élément pour mettre à jour l'heure en temps réel via JavaScript.

## Mise à jour de l'affichage de l'horloge en Javascript

Afin de mettre à jour l'affichage de l'horloge numérique en temps réel j'ai développé la fonction updateClock assurant que l'horloge numérique affichée sur la page web reste synchronisée avec l'heure actuelle. Elle extrait les heures, minutes et secondes, les formate de manière appropriée, puis met à jour le contenu du paragraphe Html. Enfin, elle se réappelle toutes les secondes pour maintenir l'horloge à jour, fournissant ainsi une interface utilisateur dynamique et en temps réel.



```
function updateClock() {
    // Instance of Date object
    var now = new Date(),

        // Get the actual hour and convert it to String, adding 0 to keeping
        // the time format
        hours = now.getHours().toString().padStart(2, '0'),
        minutes = now.getMinutes().toString().padStart(2, '0'),
        seconds = now.getSeconds().toString().padStart(2, '0')

        // Get the paragraph with the id 'ClockDisplay' and write its content
        // with the actual Time
        document.getElementById('clockDisplay').textContent =
            hours + ':' + minutes + ':' + seconds

        // Each 1 second, execute the updateClock function
        setTimeout(updateClock, 1000)
}
```

Figure 34: Fonction updateClock()

L'obtention de la Date et de l'Heure actuelles par la variable var now = new Date(), créée en tant qu'instance de l'objet Date. Elle représente le moment actuel exact où la fonction est exécutée, incluant la date et l'heure. L'extraction et le formatage des Heures, Minutes et Secondes ce fait par



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

les variable hours, minutes, seconds. Par exemple la variable hours = now.getHours() récupère l'heure actuelle, puis la convertit en chaîne de caractères .toString() et ajoute un zéro devant si nécessaire pour garantir un format sur deux chiffres (ex: "08" au lieu de "8").

La mise à Jour du contenu de l'élément Html paragraphe est réalisé par l'étape document.getElementById('clockDisplay').textContent = hours + ':' + minutes + ':' + seconds, document.getElementById('clockDisplay') sélectionne l'élément HTML avec l'id clockDisplay et .textContent modifie le texte à l'intérieur de cet élément pour afficher l'heure actuelle sous la forme "HH:MM:SS".

La mise à jour automatique à chaque seconde est obtenu par la fonction setTimeout(updateClock, 1000), setTimeout étant une fonction JavaScript qui exécute une autre fonction après un certain délai, spécifié en millisecondes. La fonction updateClock est appelé de manière récursive toutes les 1000 millisecondes (1 seconde), garantissant que l'heure affichée est constamment mise à jour.

## Affichage des données météo en Javascript – première partie

Cette première partie du script weather.js configure l'interaction utilisateur pour la recherche de la météo en fonction de la ville saisie. Elle gère la récupération de la clé API OpenWeatherMap nécessaire pour accéder aux données météorologiques et s'assure que les informations sont affichées de manière dynamique sur l'interface utilisateur en mettant à jour l'affichage de la météo.



```
// Get the search button
const searchButton = document.getElementById('searchButton')

// Listen for a click event on the search button
searchButton.addEventListener('click', function () {
    // Get the value of the location input
    const city = document.getElementById('locationInput').value

    if (city) {
        // Fetch the API key
        fetchApiKey()
            // Then call the getWeather function with city and apiKey as
            parameters
            .then((apiKey) => getWeather(city, apiKey))
            .catch((error) => console.error('Error fetching the API key:', error))
    } else {
        alert('Please enter a city name.')
    }
})
```

Figure 35: Première partie du script weather.js



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI



```
// If no city is provided, fetch weather for 'Toulon' by default
fetchApiKey()
  .then((apiKey) => getWeather('Toulon', apiKey))
  .catch((error) => console.error('Error fetching the API key:', error))

// Asynchronous function for fetching the API key
async function fetchApiKey() {
  // Wait for the response from the API route
  const response = await fetch('/api-key')

  // Wait to parse the response data as JSON
  const data = await response.json()

  // Return the API key from the data
  return data.apiKey
}
```

Figure 36: Suite première partie du script weather.js

On commence par sélectionner et stocker dans la constante searchButton l'élément Html avec l'id searchButton const searchButton = document.getElementById('searchButton'). Cet élément représente le bouton <button id="searchButton" class="search">Search</button> permettant à l'utilisateur de lancer la recherche de la météo.

Un écouteur d'événement est ajouté au bouton de recherche permettant d'executer la fonction spécifié lorsque l'utilisateur clique sur le bouton searchButton.addEventListener('click', function () { }). La ligne suivante const city = document.getElementById('locationInput').value récupère la valeur entrée par l'utilisateur dans le champ de saisie avec l'id locationInput et la stocke dans la variable city.

La condition vérifie si l'utilisateur a bien entré une ville. Si city n'est pas vide, la fonction continue ; sinon, une alerte est affichée demandant à l'utilisateur de saisir un nom de ville. Dans le cas où city est défini par l'utilisateur un appel asynchrone pour récupérer la clé d'API permettant d'obtenir la météo est effectué.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI



```
fetchApiKey()  
  // Then call the getWeather function with city and apiKey as  
  // parameters  
  .then((apiKey) => getWeather(city, apiKey))  
  .catch((error) => console.error('Error fetching the API key:',  
  // error))
```

fetchApiKey() est une fonction asynchrone qui récupère la clé d'API nécessaire pour accéder aux données météorologiques, ensuite .then((apiKey) => getWeather(city, apiKey)) utilise la clé récupérée pour appeler la fonction getWeather avec la ville entrée par l'utilisateur et la clé d'API. .catch((error) => console.error('Error fetching the API key:', error)) gère les erreurs potentielles lors de la récupération de la clé API.



```
fetchApiKey()  
  .then((apiKey) => getWeather('Toulon', apiKey))  
  .catch((error) => console.error('Error fetching the API key:', error))
```

Lors du premier chargement de la page la récupération initiale de la météo est défini par défaut avec les données météorologiques de la ville de Toulon.



```
async function fetchApiKey() {  
  const response = await fetch('/api-key')  
  const data = await response.json()  
  return data.apiKey  
}
```

Enfin, on définit une fonction asynchrone nommée fetchApiKey permettant de récupérer de manière



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

asynchrone les données d'API sous fomat Json en utilisant await fetch('/api-key') pour attendre la réponse d'une requête HTTP à l'URL /api-key récupérerant la clé d'API et await response.json() qui convertit la réponse en format JSON, return data.apiKey retournant la clé d'API extraite des données JSON.

## Affichage des données météo en Javascript – deuxième partie

La deuxième partie du script est une fonction asynchrone qui récupère et affiche les informations météorologiques pour une ville donnée en utilisant l'API OpenWeatherMap. Cette fonction fournit une solution complète pour récupérer et afficher des données météorologiques dynamiques sur une interface utilisateur web, en utilisant des requêtes asynchrones et en gérant les erreurs potentielles.



```
// Asynchronous function to get weather information
async function getWeather(city, apiKey) {
    const apiUrl = 'https://api.openweathermap.org/data/2.5/weather'

    // Get elements to display weather information
    const locationElement = document.getElementById('location')
    const temperatureElement = document.getElementById('temperature')
    const weatherIconElement = document.getElementById('weatherIcon')
    const descriptionElement = document.getElementById('description')
    const humidityElement = document.getElementById('humidity')

    // Construct the URL with city and API key
    const url = `${apiUrl}?q=${city}&appid=${apiKey}&units=metric`

    try {
        // Fetch weather data from the API
        const response = await fetch(url)
        if (!response.ok) {
            throw new Error('Weather data not available')
        }
        const data = await response.json()
    }
```

Figure 37: Deuxième partie du script weather.js

La constante apiUrl définit l'URL de base pour l'API OpenWeatherMap, qui fournit les données météorologiques au format Json. Les variables suivante ( locationElement, etc.) récupèrent les références spécifiques aux éléments Html du document par leur id et servent de placeholders où les



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

informations météorologiques seront affichées.

La variable url contient l'URL construite et complète pour la requête de l'API. Elle est composée du paramètre du nom de la ville q=\${city} pour lequel les informations météorologiques sont demandées, du paramètre de la clé API appid=\${apiKey} nécessaire à l'authentification, et du paramètre units=metric spécifiant que les unités de température doivent être en Celsius.

On conditionne avec try {} un essaie de récupération des données. Si la récupération des données par la variable response est correct, utilisant une requête asynchrone fetch(url) vers le point de terminaison de l'API construit précédemment alors on convertit la réponse en format JSON response.json() pour extraire les données météorologiques.



```
// Extract weather information
const temperature = Math.round(data.main.temp)
const humidity = data.main.humidity
const weatherDescription = data.weather[0].description
const weatherId = data.weather[0].id

// Display basic weather information
locationElement.textContent = data.name
temperatureElement.textContent = `${temperature}°C`
descriptionElement.textContent = weatherDescription
humidityElement.textContent = `${humidity}%`

// Determine the weather icon to display
let weatherIcon = ''
const weatherIcons = {
  thunderstorm: './assets/icons/thunderstorm.png',
  drizzle: './assets/icons/drizzle.png',
  rain: './assets/icons/rain.png',
  snow: './assets/icons/snow.png',
  clear: './assets/icons/clear.png',
  clouds: './assets/icons/clouds.png',
  mist: './assets/icons/mist.png',
  dust: './assets/icons/dust.png',
  ash: './assets/icons/ash.png',
  squall: './assets/icons/squall.png',
  tornado: './assets/icons/tornado.png',
  unknown: './assets/icons/unknown.png',
}
```

Figure 38: Suite deuxième partie du script weather.js



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

Ensuite on extrait les informations météorologiques. La température est extraite de l'objet data.main.temp et arrondie à l'entier le plus proche avec la fonction Math.round, l'humidité est obtenue directement de data.main.humidity. La description météorologique est récupérée à partir de data.weather[0].description et l'identifiant météorologique est extrait de data.weather[0].id.

L'affichage des informations météorologiques de base sont effectués, l'affichage de la localisation ( le nom de la ville ) est assigné à l'élément locationElement et l'affichage de la température, suivie de "°C", est assignée à l'élément temperatureElement. La description météo est assignée à l'élément descriptionElement, enfin, l'humidité, suivie de "%", est assignée à l'élément humidityElement.

```
// Match weather condition to corresponding icon
if (weatherId >= 200 && weatherId <= 232) {
    weatherIcon = weatherIcons.thunderstorm
} else if (weatherId >= 300 && weatherId <= 321) {
    weatherIcon = weatherIcons.drizzle
} else if (weatherId >= 500 && weatherId <= 531) {
    weatherIcon = weatherIcons.rain
} else if (weatherId >= 600 && weatherId <= 622) {
    weatherIcon = weatherIcons.snow
} else if (weatherId === 800) {
    weatherIcon = weatherIcons.clear
} else if (weatherId >= 801 && weatherId <= 804) {
    weatherIcon = weatherIcons.clouds
} else if (weatherId >= 701 && weatherId <= 781) {
    if (
        weatherId === 701 ||
        (weatherId >= 711 && weatherId <= 721) ||
        weatherId === 741
    ) {
        weatherIcon = weatherIcons.mist
    } else if (weatherId === 731 || (weatherId >= 751 && weatherId <=
761)) {
        weatherIcon = weatherIcons.dust
    } else if (weatherId === 762) {
        weatherIcon = weatherIcons.ash
    } else if (weatherId === 771) {
        weatherIcon = weatherIcons.squall
    } else if (weatherId === 781) {
        weatherIcon = weatherIcons.tornado
    }
} else {
    weatherIcon = weatherIcons.unknown
}

// Display the weather icon
weatherIconElement.src = weatherIcon
} catch (error) {
    console.error('Error fetching weather data:', error)
}
}
```

Figure 39: Fin de la deuxième partie du script weather.js



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

La détermination de l'icône météo à afficher est défini par la variable `weatherIcon` initialisée à une chaîne vide pour le moment, et l'objet `weatherIcons` est créé, associant chaque condition météorologique à une image d'icône spécifique.

La dernière du script est l'association des conditions météorologiques avec l'icône correspondante en fonction de l'identifiant météorologique fourni par l'API. Les différentes conditions météorologiques telles que l'orage, la bruine, la pluie, la neige, le ciel clair, les nuages, et diverses conditions atmosphériques (brume, poussière, cendres, rafales, tornades) sont prises en compte. Pour chaque condition, une icône spécifique est assignée et affichée. Cela permet à l'utilisateur de visualiser facilement les conditions météorologiques actuelles grâce à une représentation iconographique appropriée.

Affichage de différent résultat en fonction de la ville choisi

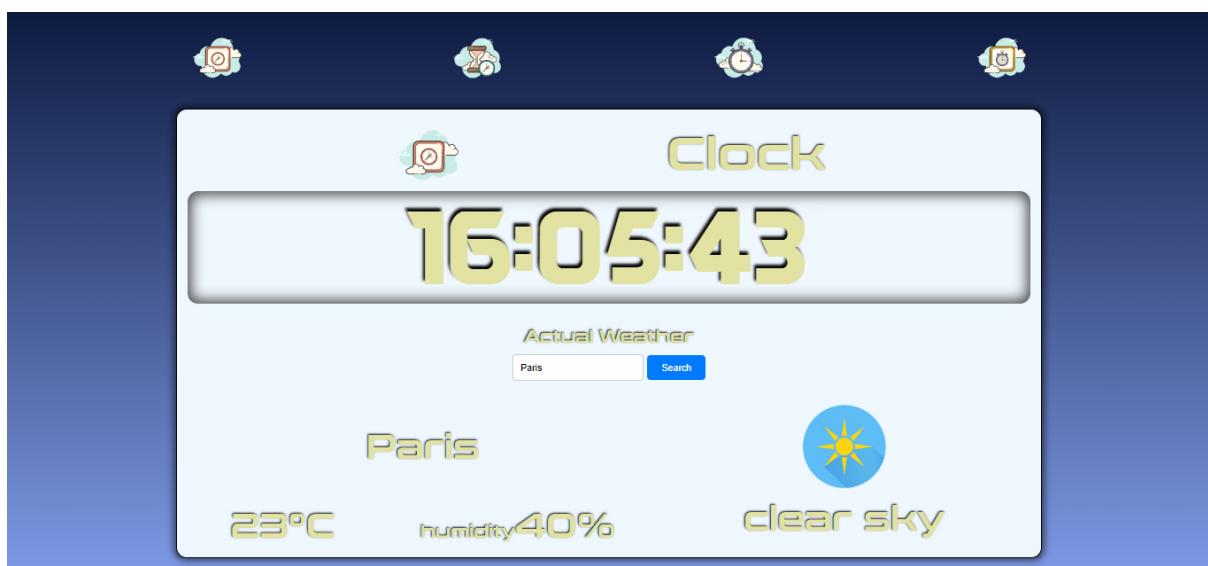


Figure 40: Météo Paris



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

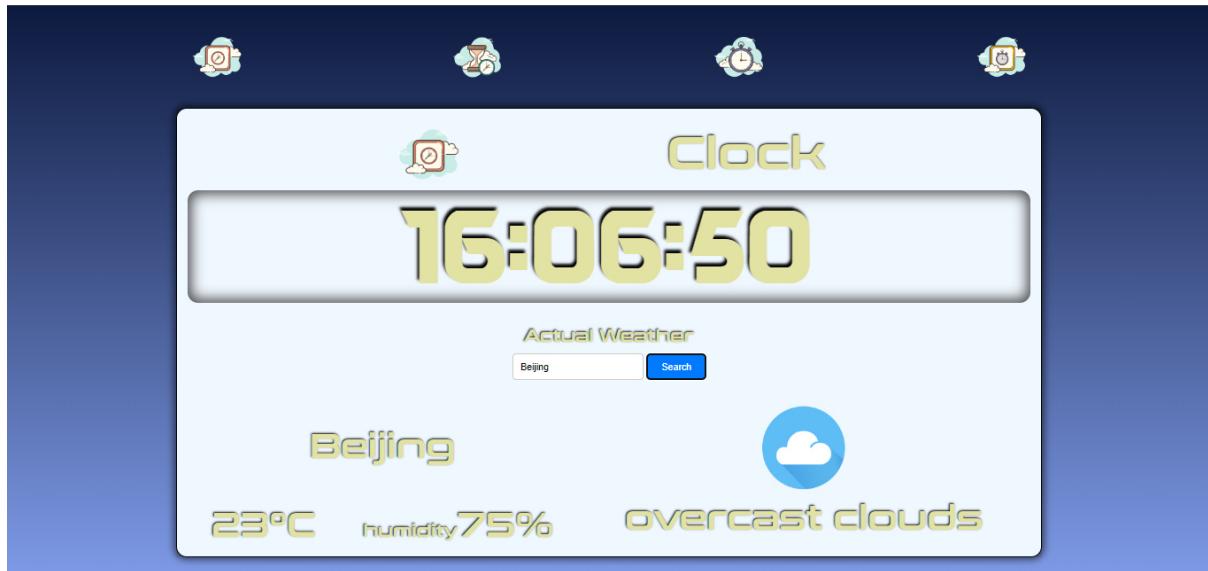


Figure 41: Météo Beijing

## 2. Précisez les moyens utilisés :

Pour réaliser ce projet j'ai utilisé la documentation MDN pour m'appuyer dans le développement, le site OpenWeather pour générer une clé d'API et leur documentation pour comprendre le format des données envoyées, Icon8 et TransparentPNG pour les icons.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *La Plateforme\_*

Chantier, atelier, service ► *Projet O'Clock*

Période d'exercice ► Du : *03/06/2024* au : *08/06/2024*

## 5. Informations complémentaires (facultatif)

L'idée d'ajouter la partie Météo à ce projet m'est venue car j'avais déjà utilisé OpenWeather dans le cadre d'un exercice effectué en Python, où j'ai créé un Bot Discord permettant l'affichage de la météo d'une ville par défaut (Toulon) ou d'une ville saisie par l'utilisateur. Ce projet peut être visualisé à cette adresse : <https://github.com/Alexandre2383/Bot-discord>



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Activité-type 2 Développer la partie back-end d'une application web ou web mobile sécurisée.

**CP 5 ▶ Mettre en place une base de données relationnelle.**

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

#### Introduction

Dans le cadre du projet de boutique en ligne, nous avons eu à réaliser un site de vente de café et de thé. Afin de pouvoir réaliser le site, nous avons eu à créer sa base de données. Nous avons reçu le brief du client fictif et nous avons utilisés la méthode Merise (méthode informatique qui facilite la création de base de données et de projets informatiques). Nous avons ainsi pu créer le Modèle Conceptuel de Données (MCD), le Modèle Logique de Données (MLD), le Modèle Physique de Données (MPD) et rédiger le dictionnaire de données.

#### MCD (Modèle Conceptuel de Données)

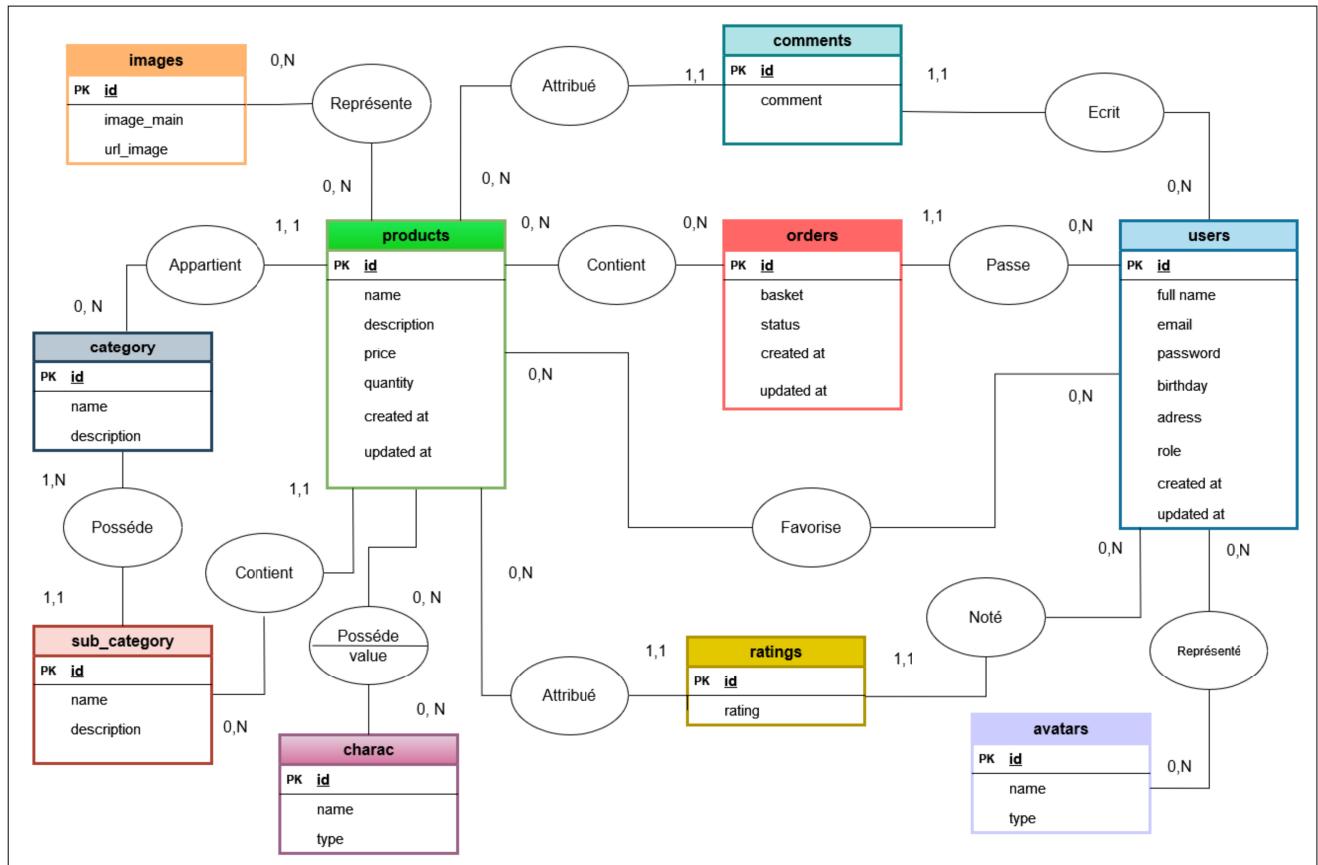
Le MCD est une modélisation des données où on y représente un système de données sous la forme d'entités et leurs relations, ainsi que leurs cardinalités. Dans le cas du site, nous avons listés 10 entités et leurs attributs, Users et Products étant les plus importantes en terme de relations.

Un utilisateur (Users) peut passer plusieurs commandes (Orders) contenant au moins un produit (Products), il peut écrire un commentaire (Comments) attribué qu'à un seul produit. Il peut donner une note (Ratings) attribué qu'à un seul produit, être représenté par une image de profil (Avatars), et aimer (attribut Favorise) un produit.

Un produit peut être représenté par une image (Images), il appartient à une catégorie (Category) qui possède une sous catégorie (Sub\_category). Il peut être contenu dans une commande (Orders). Un produit peut posséder des caractéristiques (Charac) qui ont une valeur. Enfin, il peut être noté (Ratings), commenté (Comments), ou favorisé (Favorise).



# DOSSIER PROFESSIONNEL (DP)



## MLD (Modèle Logique de Données) :

A partir des cardinalités développées dans le MCD, nous avons établis le MLD en attribuant les clés étrangères aux tables concernées et en créant 5 tables de jointures.

Par exemple, les tables Products et Orders ayant une cardinalité de 0,N dans leurs relations (Many-to-Many) une table de jointures PRODUCTSORDERS sera créée contenant l'identifiant du produit en relation avec l'identifiant de la commande. L'attribut Favorise deviendra une table de relation appelé USERS\_HAS\_PRODUCTS permettant d'ajouté ou enlevé les produits favoris sélectionner par l'utilisateur.

**AVATARS** (`id`, `url_avatar`)

**CATEGORY** (`id`, `name`, `description`)

**CHARAC** (`id`, `name`, `type`)

**COMMENTS** (`id`, `comment`, `#users_id`, `#products_id`)

**IMAGES** (`id`, `image_main`, `url_main`)

**ORDERS** (`id`, `basket`, `status`, `created_at`, `updated_at`, `#users_id`)

**PRODUCTS** (`id`, `name`, `description`, `price`, `quantity`, `created_at`, `updated_at`, `#category_id`, `#sub_category_id`)



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

**PRODUCTSCHARAC** (value, #charac\_id, #products\_id)

**PRODUCTSIMAGES** (#products\_id, #images\_id)

**PRODUCTSORDERS** (#orders\_id, #products\_id)

**RATINGS** (id, rating, #products\_id, #users\_id)

**SUB\_CATEGORY** (id, name, description, #category\_id)

**USERS** (id, full\_name, email, password, birthday, address, role, created\_at, updated\_at)

**USERSAVATARS** (#users\_id, #avatars\_id)

**USERS\_HAS\_PRODUCTS** (#users\_id, #products\_id)

## Dictionnaire de données

Voici un extrait du Dictionnaire de données, tout les champs seront conservés en langue Anglaise.

### orders

Colonne	Type			Null	Valeur par défaut	Commentaires
id ( <i>Primaire</i> )	int(11)			Non		
basket	tinyint(1)			Oui	NULL	
status	enum('en attente', 'expédié', 'livré', 'échec')			Oui	NULL	
created_at	datetime			Oui	current_timestamp()	
updated_at	datetime			Oui	current_timestamp()	
users_id	int(11)			Non		

#### Index

Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
PRIMARY	BTREE	Oui	Non	id	42	A	Non	
fk_orders_users1_idx	BTREE	Non	Non	users_id	42	A	Non	

### products

Colonne	Type		Null	Valeur par défaut	Commentaires
id ( <i>Primaire</i> )	int(10)		Non		
name	varchar(100)		Oui	NULL	
description	text		Oui	NULL	
price	float		Oui	NULL	
quantity	int(11)		Oui	NULL	
created_at	datetime		Oui	current_timestamp()	
updated_at	datetime		Oui	current_timestamp()	
category_id	int(11)		Non		
sub_category_id	int(11)		Non		

#### Index

Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
PRIMARY	BTREE	Oui	Non	id	249	A	Non	
id_product_UNIQUE	BTREE	Oui	Non	id	249	A	Non	
name_UNIQUE	BTREE	Oui	Non	name	249	A	Oui	
fk_products_category1_idx	BTREE	Non	Non	category_id	4	A	Non	
fk_products_sub_category1_idx	BTREE	Non	Non	sub_category_id	12	A	Non	

### productsimages

Colonne	Type		Null	Valeur par défaut	Commentaires
products_id	int(10)		Non		
images_id	int(11)		Non		

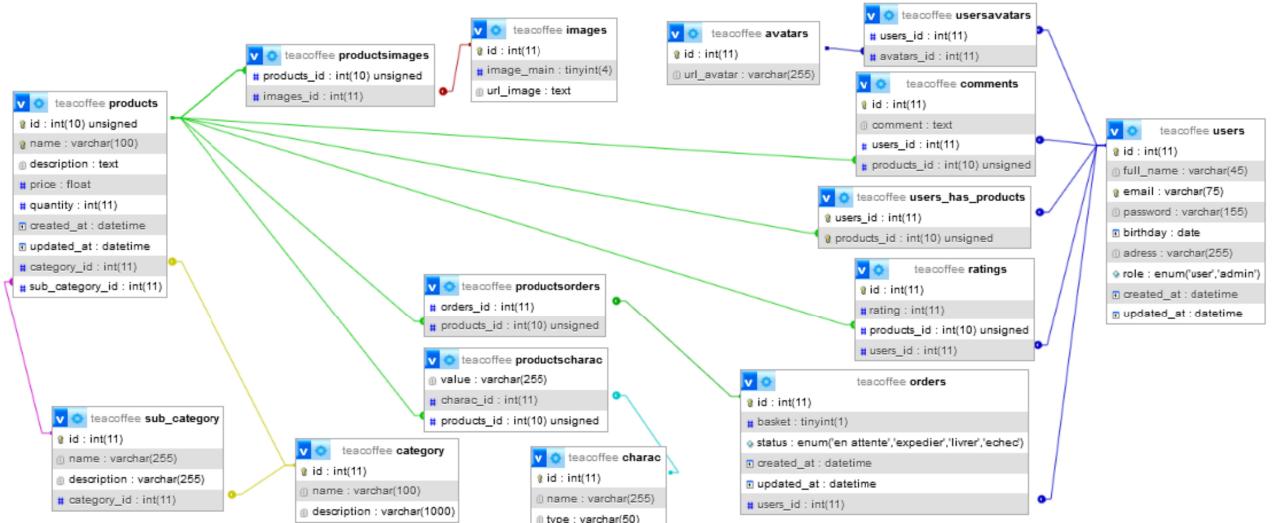
Colonne	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
fk_ProductsImages_images1_idx	BTREE	Non	Non	images_id	83	A	Non	
fk_ProductsImages_products1_idx	BTREE	Non	Non	products_id	83	A	Non	

## MPD (Modèle Physique de Données) :

Une fois les tables générées dans le SGBD MySQL, l'interface graphique PhpMyAdmin permet de visualiser le MPD en utilisant la fonctionnalité Concepteur.



# DOSSIER PROFESSIONNEL (DP)



Voici

un

```
-- Création des tables de données
CREATE TABLE `avatars` (
  `id` int NOT NULL,
  `url_avatar` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `category` (
  `id` int NOT NULL,
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL,
  `description` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `charac` (
  `id` int NOT NULL,
  `name` varchar(255) NOT NULL,
  `type` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

//...

```
-- Ajout des clés primaires et étrangères
```

```
ALTER TABLE `avatars`
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `category`
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `charac`
ADD PRIMARY KEY (`id`);
```

//...

```
-- Ajout des contraintes par table
```

```
ALTER TABLE `comments`
ADD CONSTRAINT `fk_comments_products1` FOREIGN KEY (`products_id`) REFERENCES `products` (`id`),
ADD CONSTRAINT `fk_comments_users1` FOREIGN KEY (`users_id`) REFERENCES `users` (`id`);
```

```
ALTER TABLE `orders`
ADD CONSTRAINT `fk_orders_users1` FOREIGN KEY (`users_id`) REFERENCES `users` (`id`);
```

```
ALTER TABLE `products`
ADD CONSTRAINT `fk_products_category1` FOREIGN KEY (`category_id`) REFERENCES `category` (`id`),
ADD CONSTRAINT `fk_products_sub_category1` FOREIGN KEY (`sub_category_id`) REFERENCES `sub_category` (`id`);
```



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

extrait des requêtes SQL pour la création des tables de la base de données.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## 2. Précisez les moyens utilisés :

Je me suis inspiré des cours des formateurs, de la documentation SQL ainsi que le site SQL.sh <https://sql.sh/>. J'ai utilisé draw.io pour développer le MCD, et les fonctionnalité de PhpMyAdmin une fois la base de donnée créée.

## 3. Avec qui avez-vous travaillé ?

Ce projet a été réalisé en groupe.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme*

Chantier, atelier, service ► Boutique en ligne.

Période d'exercice ► Du : *02/04/2024* au : *30/04/2024*

## 5. Informations complémentaires (facultatif)



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Activité-type 2

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

**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 :

#### Introduction

L'application Tea'Coffee suit le modèle architectural MVC (Modèle-Vue-Contrôleur) pour structurer son code de manière modulaire et efficace. Ce modèle permet de séparer la logique métier, la présentation et le contrôle des flux de données, facilitant ainsi la maintenance et l'évolution de l'application.

The screenshot shows a web browser displaying the 'boutique-js.test:8880' page for the Tea'Coffee application. The header includes the logo 'Tea'Coffee' and links for 'Accueil', 'Café', 'Thé', and 'Contact'. On the right, there are icons for a user profile, inscription, and connexion. A banner at the top left says 'Dernier produit de la boutique'. Below it, three product cards are displayed: 'Café Affogato Corsé' (12.5 €), 'Café Americano Corsé' (4.5 €), and 'Café Bélizien Corsé' (3 €). Each card has a 'Ajouter au panier' button. The background features a floral pattern.

Figure 42: Affichage des données de la table products

Le point d'entrée de l'application est le fichier index.php, qui redirige les requêtes de l'utilisateur vers les routes appropriées.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

Par exemple, la route GET '/' renvoie l'utilisateur sur la page d'accueil. Cette page utilise la méthode RenderHome du contrôleur HomeController. Cette méthode gère la réception des données et l'hydratation du template avant de renvoyer l'ensemble à la classe Render, qui se charge de la logique de rendu du contenu à l'utilisateur. La méthode de la classe Render intègre le header et le footer au template accueil avant de rendre la vue.

Rôle du fichier index.php, point d'entrée de l'application

Le fichier index.php initialise l'environnement de l'application, récupère les informations de requête, configure le routage et transmet les paramètres nécessaires à la classe de rendu. Sa fonction principale est de gérer l'initialisation de l'application et de rediriger les requêtes de l'utilisateur vers les routes appropriées. Il assure que toutes les requêtes HTTP entrantes sont correctement traitées et acheminées vers les contrôleurs et les actions correspondants.

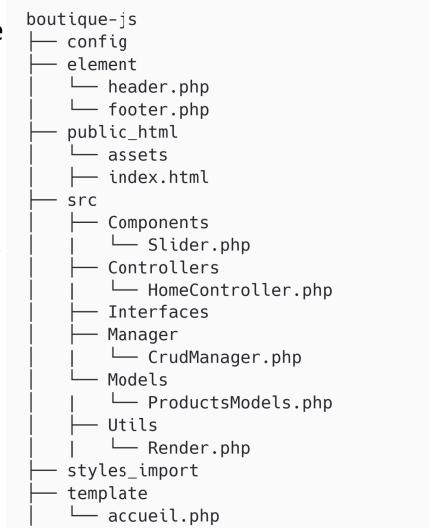


Figure 43: Structure de l'application

```

// récupération de l'autoload
require_once __DIR__ . '/../vendor/autoload.php';
// initialisation des requêtes URI et de l'hôte HTTP
$uri = $_SERVER['REQUEST_URI'];
$serverName = $_SERVER['HTTP_HOST'];
// instance des classes AltoRouter et Render
$router = new AltoRouter();
$rendering = new Render();
// Appelle de la méthode addParams afin d'ajouter la variable $uri
// et $serverName
$rendering->addParams(['uri' => $uri, 'serverName' => $serverName]);
// route de la page accueil
$router->map('GET', '/', 'HomeController#RenderHome', 'accueil');

```

Figure 44: Première partie de index.php

Gestion des requêtes utilisateur et redirection des routes

Le fichier index.php inclut l'autoloader généré par Composer, permettant le chargement automatique des classes nécessaires à l'exécution de l'application. Nous récupérons l'URI de la requête et le nom de l'hôte HTTP depuis les variables globales \$\_SERVER. Ces



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

informations sont cruciales pour déterminer quelle route doit être empruntée par l'utilisateur. Nous créons des instances de AltoRouter et de Render. AltoRouter est une bibliothèque légère pour le routage des requêtes HTTP, tandis que Render est une classe responsable de la logique de rendu du contenu. Nous appelons la méthode addParams de l'instance Render pour ajouter les variables uri et serverName aux paramètres. Cela permet de transmettre ces informations importantes pour le rendu du contenu.

## Exemple de route : La page d'accueil '/'

Le code ci-dessous montre comment notre application gère les routes, en utilisant la bibliothèque AltoRouter pour mapper les requêtes aux contrôleurs et méthodes appropriés. Cette ligne mappe une requête HTTP GET pour l'URL racine (/) à la méthode RenderHome du contrôleur HomeController. La route est nommée accueil ( \$router->map('GET', '/', 'HomeController#RenderHome', 'accueil'); ).

Nous utilisons la méthode match pour trouver la route correspondant à l'URL actuelle. Si une correspondance est trouvée, elle est stockée dans la variable \$match.



# DOSSIER PROFESSIONNEL (DP)

```
///...
// route de la page accueil
$router->map('GET', '/', 'HomeController#RenderHome', 'accueil');
$match = $router->match();
try {
    // si la route est bien enregistré avec $router->map
    if (is_array($match)) :
        $params = $match['params'];
        // assigne les valeurs du tableau si la target contient bien '#'
        if (str_contains($match['target'], '#')) :
            // $controller pour $match['target'][0], $method pour $match['target'][1]
            [$controllers, $method] = explode('#', $match['target']);
            // défni le namespace du contrôleur
            $controller = 'App\\Boutique\\Controllers\\' . $controllers;
            //condition ternaire assurant l'existence de la classe, évitant les erreurs
            $controller = class_exists($controller) ? new $controller() : false;
            // récupération des valeurs transmises par $_POST
            if (isset($_POST)) {
                foreach ($_POST as $key => $value) {
                    $match['params'][$key] = htmlspecialchars(trim($value));
                }
            }

            // ajoute le nom de domaine et l'URI dans les params à transmettre à la classe du
            controller (Pour le lien des images par exemple)
            $match['params']['render'] = $rendering;
            // condition de vérification de la présence d'une $method au $controller dans la
            target (méthode callable est non static)
            if (is_callable([$controller, $method])) :
                // utilisation de la fonction call_user_func_array afin d'instancier la classe
                chargée précédemment dans la variable $controller
                // en deuxième paramètre passage d'un tableau d'arguments récupérable dans la
                méthode déclaré $method
                echo call_user_func_array([$controller, $method], $match['params']);
            else :
                throw new MotorMvcException(ExceptionEnum::ControllerNotFound);
            endif;
        else :
            // sinon la target ne contient pas de controller
            $rendering->addParams('params', $match['params']);
            // on appelle la méthode defaultRender
            echo $rendering->defaultRender($match['target']);
        endif;
    //...
}
```

Figure 45: Deuxième partie de index.php

## Gestion de la route correspondante

Si la variable \$match est un tableau, cela signifie qu'une route correspondante a été trouvée. Lorsque la target contient un #, cela indique que la route est définie avec un contrôleur et une méthode. La target est alors divisée en \$controller et \$method. Le contrôleur est instancié dynamiquement en utilisant son nom complet, y compris le namespace App\Boutique\Controllers. Les données POST sont nettoyées à l'aide des fonctions trim et htmlspecialchars, puis ajoutées aux paramètres de la route. Ensuite, le nom de domaine et l'URI sont ajoutés aux paramètres, ce qui est particulièrement utile pour



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

des éléments tels que les liens d'images. Si la méthode est callable, elle est alors appelée avec les paramètres correspondants en utilisant call\_user\_func\_array. Dans le cas contraire, une exception est lancée (controller introuvable).

## Gestion des cibles sans contrôleur

Si la target ne contient pas de contrôleur (pas de #), les paramètres sont ajoutés à l'instance de Render, et la méthode defaultRender est appelée pour rendre le contenu.

## Rôle de la méthode RenderController, réception et traitement des données

La méthode RenderHome de la classe HomeController est responsable de traiter les données nécessaires pour afficher la page d'accueil de l'application. Elle commence par créer des instances de CrudManager pour les produits et les utilisateurs, cette classe gérant les opérations CRUD (Create, Read, Update, Delete) pour une table spécifique dans la base de données. Ici, nous créons deux instances : une pour la table products et une autre pour la table users.

```
class HomeController
{
    public function __construct()
    {
    }
    public function RenderHome(...$arguments)
    {
        //...

        // instance de CrudManager prenant en paramètre la table products et la classe du
        // modèle de données ProductsModels
        $crudManagerProduct = new CrudManager('products', ProductsModels::class);

        $crudManagerUser = new CrudManager('users', ProductsModels::class);

        // instance du composant Slider
        $horizontalSlide = new Slider();

        // condition de vérification de la connection utilisateur
        if (isset($_SESSION['isConnected'])) {
            // assignation de l'email utilisateur à $user
            $user = $crudManagerUser->getByEmail($_SESSION['email']);
            // assignation des données favoris de l'utilisateur
            $products = $crudManagerProduct->getAllProductFav($user->id);
        } else {
            // assignation des données de la table Products
            $products = $crudManagerProduct->getAllProduct();
        }
        // hydratation du slider par l'import des id favoris de l'utilisateur et des
        // données de la table Products
        $allProducts = $horizontalSlide->generateProductList($products, 'id-scroll-x-1');
    }
}
```

Figure 46: Première partie de HomeController.php



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

Ensuite, une instance du composant Slider est utilisée pour afficher les produits dans un format de diaporama horizontal sur la page d'accueil. Si l'utilisateur est connecté, identifié par `$_SESSION['isConnected']`, on récupère les informations de l'utilisateur via son email. Ensuite, on récupère ses produits favoris ; sinon, on ne récupère que les produits de la table products. Le composant Slider est alors hydraté avec les données des produits, et, uniquement si l'utilisateur est connecté, ses favoris.

## Hydratation du template avec les données, la méthode getAllProduct

Le processus d'hydratation du template avec les données consiste à peupler le composant Slider avec les informations des produits obtenues par le CrudManager. La méthode `getAllProduct`, appartenant à la classe CrudManager, récupère l'ensemble des produits depuis la base de données avec une jointure sur l'URL de l'image associée.

```
/*
 * Method getAllProduct : Renvoi l'ensemble des produits avec la jointure de l'url_image
 */
@return array
public function getAllProduct(): array
{
    $connect = $this->_dbConnect;
    // La désactivation permet de passer un booléen à la requête PDO et implémenter la
    // pagination
    $connect->setAttribute(\PDO::ATTR_EMULATE_PREPARES, false);
    // requête de la table Products
    $req = $connect->prepare(
        "SELECT p.*, i.url_image
         FROM {$this->_tableName} p
         INNER JOIN productsimages pi ON p.id = pi.products_id
         INNER JOIN images i ON pi.images_id = i.id
         WHERE p.id = pi.images_id
         LIMIT :limit OFFSET :offset",
    );
    $req->execute([':limit' => $this->limit, ':offset' => $this->offset]);
    $req->setFetchMode(\PDO::FETCH_CLASS | \PDO::FETCH_PROPS_LATE, $this->_objectClass);
    // renvoi l'ensemble des données récupérés
    return $req->fetchAll();
}
```

Figure 47: Méthode getAllProduct



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

La connexion à la base de données désactive l'émulation des requêtes préparées avec \ PDO::ATTR\_EMULATE\_PREPARES pour permettre le passage correct des paramètres de pagination. Ensuite, la requête SQL sélectionne tous les produits et fait une jointure sur les tables productsimages et images pour obtenir l'URL de l'image associée. La requête est exécutée avec les paramètres de pagination (:limit et :offset). Les résultats sont récupérés sous forme d'objets de la classe définie par \_objectClass.

## Hydratation du template avec la fonction generateProductList de la classe Slider

La fonction generateProductList prend un tableau de produits et un identifiant de composant (id), puis génère dynamiquement un slider contenant les informations de chaque produit. Le composant slider est initialisé avec une <div> englobante et une <ul> pour contenir les éléments des produits. L'identifiant id est utilisé pour cibler le slider de manière unique. Pour chaque produit dans le tableau \$products, on vérifie si le produit est marqué comme favori par l'utilisateur (user\_has\_product). Si le produit est favori, on ajoute une classe CSS inFav pour l'indiquer visuellement. Le chemin de l'image du produit est vérifié. Si le fichier existe, l'URL de l'image réelle est utilisée. Sinon, une image par défaut est utilisée. L'élément <img> est hydraté avec l'URL correcte et les attributs nécessaires. Les détails du produit, tels que le nom et le prix, sont insérés dans les éléments php appropriés. Un bouton interactif est ajouté, configuré pour déclencher une action JavaScript pour ajouter le produit au panier. La boucle se termine et les éléments de produits sont complétés. Les boutons de défilement horizontal sont ajoutés au slider pour permettre une navigation facile. Le slider complet et hydraté est retourné pour être enfin rendu sur la page.



# DOSSIER PROFESSIONNEL (DP)

```

public function generateProductList(array $products, string $id)
{
    // $slider contient l'ensemble du composant hydraté et concaténé
    $slider = "
        <div id='menu' class='relative flex mx-auto'>
            <ul id='{$id}' class='lg:ml-[3%] lg:mr-[3%] block list-none p-0
white-space nowrap overflow-x-auto max-with-full md:max-w-full lg:max-w-6xl'>;
    // boucle d'hydratation des données Products
    foreach ($products as $productItem):
        // assignation du favori en fonction des données de la table
        user_has_product
        if ($productItem->user_has_product != null) {
            $inFav = "inFav ";
        } else {
            $inFav = null;
        }

        $slider .= "<div class='product-container my-4 bg-gray-100 w-[20rem] sm:h-[20rem] lg:h-fit inline-block mx-4 rounded-2xl p-6 cursor-pointer lg:hover:-translate-y-2 transition-all relative shadow-[3px_3px_9px_0px_rgba(0,0,0,0.6)]'>
            <div id='{$productItem->id}' class='{$inFav}favorites bg-gray-200 w-10 h-10
flex items-center justify-center rounded-full cursor-pointer absolute top-4 right-4'>

        ...
            // exemple d'hydratation de l'identifiant du produit
            <div id='{$productItem->id}' class='article-image-container w-2/3 h-[220px]
sm:h-[5rem] md:h-[5rem] lg:h-[220px] overflow-hidden mx-auto aspect-w-16 aspect-h-8'>;
                $filename = __DIR__ . "/../../public_html/assets/images/{$productItem-
>url_image}";
                // vérification de l'existance du fichier images du produit
                if (file_exists($filename) == true){

                    $slider .= "<img src='http://{$_SERVER['HTTP_HOST']}/assets/images/{$productItem->url_image}' data-url='http://{$_SERVER['HTTP_HOST']}/assets/images/{$productItem->url_image}'";
                }else{
                    // sinon utilisation d'une image par défaut
                    $slider .= "<img src='http://{$_SERVER['HTTP_HOST']}/assets/images/tea-coffee.png' data-url='http://{$_SERVER['HTTP_HOST']}/assets/images/tea-coffee.png'>";
                }

                $slider .= "alt='image {$productItem->name}' id='{$productItem->id}'>
            </div>
            <div class='product text-center mt-4' data-price='{$productItem->price}'>
                data-id='{$productItem->id}' data-name='{$productItem->name}'>
                    <h3 id='{$productItem->id}' class='article-name text-lg font-extrabold
text-gray-800'>{$productItem->name}</h3>
                    <h4 class='article-price text-2xl text-gray-800 font-bold
mt-4'>{$productItem->price} € <span class='text-gray-400 ml-2 font-
medium'>{$productItem->pound}</span>
                </h4>
                    <button data-js='handlePost,click' data-route='/addtobasket' data-body-
param={product_id:'$productItem->id'}>
                $slider .= " type='button' class='add-to-cart w-full flex items-center
justify-center gap-3 mt-6 px-4 py-2.5 bg-transparent hover:bg-gray-200 text-base text-
[#333] border-2 font-semibold border-[#333] rounded-xl'>

        ...
            Ajouter au panier</button>
        </div>
    </div>";
    endforeach;
    // fin de la boucle d'hydratation et concaténation des boutons de scroll
    $slider .= '</ul>' . self::buttonScrollX($id);
    // renvoi le slider hydraté
    return $slider;
}
}

```

Figure 48: Méthode generateProductList générant le Slider



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Rôle de la classe Render, rendu du contenu à l'utilisateur

La classe Render est responsable de la logique de rendu des templates et de l'affichage final des produits à l'utilisateur. Elle capture les données, les intègre dans les templates php, et génère le contenu final qui est envoyé au navigateur. Dans la méthode RenderHome de la classe HomeController, les produits générés sont ajoutés aux paramètres de rendu via la méthode addParams. Cette méthode ajoute les produits, contenus dans \$allProducts, aux paramètres de rendu sous la clé 'product'. Ensuite, les arguments de rendu sont passés au template via la méthode render. Cette méthode prend en paramètre le nom du template (accueil) et un tableau d'arguments, fusionnant les arguments avec les paramètres internes pour générer le contenu final. La méthode RenderHome renvoie finalement le contenu généré pour être affiché dans le navigateur.

```
/*
 * La fonction render capture la sortie, fusionne les arguments avec les paramètres,
 * inclut les éléments header et footer, et renvoie le contenu final.
 *
 * @param string $template Le nom du template à afficher.
 * @param array ...$arguments Les arguments à fusionner avec les paramètres.
 * @return string|array Le contenu final du template.
 */
public function render(string $template, ...$arguments): string|array
{
    // Démarre la mise en mémoire tampon
    ob_start();

    // Ajoute par la méthode addParams() les données de seoConfig en fonction du
    // $template, sinon par les données par défaut
    $this->addParams('seoConfig', $this->seoConfig->{$template} ?? $this->seoConfig-
    >Default);

    // Fusionne les arguments avec les paramètres et les extrait dans des variables
    // utilisables dans le template
    extract(array_merge($arguments[0] ?? [], $this->params));

    // Inclusion du header en passant les données SEO
    require_once __DIR__ . '/../../element/header.php';

    // Inclusion de la barre de recherche
    require_once __DIR__ . '/../../element/search.php';

    // ENH EXCEPTION MotorMvcException
    if (!file_exists(__DIR__ . DIRECTORY_SEPARATOR . "../../template/{$template}.php"))
    {
        throw new MotorMvcException(ExceptionEnum::TemplateNotFound);
    } else {
        // Inclusion du template
        require_once __DIR__ . "/../../template/{$template}.php";
    }

    // Inclusion du footer
    require_once __DIR__ . '/../../element/footer.php';

    // Récupère le contenu mis en mémoire tampon et l'efface
    $content = ob_get_clean();

    // Retourne le contenu
    return $content;
}
```

Figure 49: Méthode Render gérant le rendu des templates avec controller



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

La méthode render de la classe Render est centrale pour le rendu du contenu. Elle commence par démarrer la mise en mémoire tampon avec ob\_start(), capturant ainsi tout le contenu généré. Les données de configuration SEO spécifiques au template sont ensuite ajoutées aux paramètres via addParams. Les arguments passés à la méthode render sont fusionnés avec les paramètres internes et extraits pour être utilisés directement dans le template. Le header, la barre de recherche et le footer sont inclus, suivis du template spécifique (accueil dans notre cas) après vérification de son existence. Le contenu généré est capturé et la mémoire tampon est effacée avec ob\_get\_clean(). Finalement, le contenu capturé est renvoyé pour être affiché au navigateur.

## Fonctionnement avec la méthode RenderHome du contrôleur HomeController

Lorsque la route '/' de la page d'accueil est accédée, celle-ci correspond à HomeController#RenderHome. Le contrôleur HomeController est instancié, puis la méthode RenderHome est appelée avec les paramètres nécessaires. La méthode RenderHome traite les données et les prépare pour le rendu, en utilisant un template. Ce système de routage permet une gestion flexible et modulaire des requêtes en associant dynamiquement les URL aux contrôleurs et méthodes appropriés, tout en garantissant la sécurité et la robustesse du traitement des données.

```
// $content contient le passage des données du tableau
// d'arguments 'render' à la méthode render, prennant en paramètre le
// template où les arguments sont passés
$content = $arguments['render']->render('accueil', $arguments);

// rend le template chargé avec les données au navigateur
return $content;
}
```

Figure 5

```
<section class="bg-acceuil">
  <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; margin-bottom: 10px;">
    <!-- bannière carousel -->
    <?= $carousel ?>
    <div class="sm:w-full lg:max-w-6xl mx-auto">
      <!-- Appelle du render Slider -->
      <h2 class="w-64 text-gray-700 dark:text-gray-300 ...">
        Dernier produit de la boutique
      </h2>
      <div class="mx-auto lg:w-fit">
        <?= $product ?>
      </div>
    //...
  </div>
</section>
```

Figure 51: Template accueil avec insertion du Slider \$product



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

La méthode RenderHome du HomeController gère la récupération et le traitement des données des produits pour afficher la page d'accueil. Elle utilise CrudManager pour interagir avec la base de données, récupérant soit tous les produits avec ceux favoris de l'utilisateur connecté soit les tous produits uniquement. Ces données sont ensuite utilisées pour hydrater le composant Slider, permettant un affichage dynamique et personnalisé des produits sur la page d'accueil. La fonction generateProductList illustre parfaitement comment les données des produits sont intégrées dans un template php dynamique. Chaque produit est traité et ses informations sont insérées dans des éléments php appropriés. Ce processus d'hydratation assure que le contenu affiché est à jour et personnalisé selon les données disponibles, telles que les favoris des utilisateurs et les images des produits.

La classe Render joue un rôle crucial dans la préparation et l'affichage des données à l'utilisateur. Elle capture les données, les intègre dans les templates PHP et génère le contenu final de la page. La méthode render assure que les templates sont correctement inclus et que les données sont disponibles dans le contexte des templates, permettant ainsi une présentation dynamique et cohérente des produits et autres informations sur la page. Du point d'entrée de l'application via index.php, qui initialise l'environnement et configure le routage, à la méthode RenderHome de HomeController, le flux de traitement des requêtes et de rendu de contenu est orchestré de manière fluide. Cette architecture modulaire et robuste garantit une gestion efficace des requêtes utilisateur, un traitement sécurisé des données, et un affichage final optimisé, répondant aux attentes des utilisateurs.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## 2. Précisez les moyens utilisés :

Pour réaliser ce projet, nous avons utilisé PHP en combinaison avec AltoRouter via Composer pour le routage des requêtes, et Tailwind CSS pour le style de l'interface utilisateur. La documentation de PHP comme ressource pour résoudre les problèmes techniques a été utilisée, ainsi que Git pour l'organisation de travail.

## 3. Avec qui avez-vous travaillé ?

Ce projet a été réalisé en groupe.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_*

Chantier, atelier, service ► *Boutique en ligne*

Période d'exercice ► Du : *02/04/2024* au : *30/04/2024*

## 5. Informations complémentaires (facultatif)



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Activité-type 2

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

**CP 7 ▶ Développer des composants métier côté serveur.**

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

#### Introduction

Dans le cadre de la réalisation de traitements côté serveur, le composant métier Register est un élément central de l'application Tea'Coffee, chargé de gérer les processus d'inscriptions et de connections des utilisateurs. Ce composant a été développé afin de valider les données d'inscription, vérifier l'unicité de l'email, et enregistrer les nouveaux utilisateurs de manière sécurisée et fourni des retours d'erreur clairs et redirige les utilisateurs après une inscription réussie. Il utilise des méthodes de hachage pour les mots de passe et vérifie les entrées utilisateur pour éviter les failles de sécurité.

The screenshot shows the Tea'Coffee website with a navigation bar at the top. The main content area displays a 'Créer un compte' (Create account) form. The form fields are as follows:

- Votre nom et prénom: Enter votre nom complet
- Votre Email: Enter votre email
- Mot de passe: Enter votre mot de pass
- Confirmation de mot de passe: Enter votre mot de pass
- Inscription: A button labeled 'Inscription' (Registration).

Below the form, there is a link: "Vous avez déjà un compte ?" (Do you already have an account?).



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

## Description du Code

Le composant RegisterController contient plusieurs méthodes, dont les principales sont Register et ConnectJS. Voici une description détaillée de la méthode Register.

```
public function Register(...$arguments)
{
    //...

    // vérification de la connection utilisateur
    if ($arguments['render']->has('isConnected') == true) {
        header('location:/');
    }
}
```

La méthode Register est une fonction public prenant un tableau d'arguments en paramètre. Premièrement on vérifie si l'utilisateur est connecté avec une condition de comparaison. Si le paramètre ' isConnected' du tableau ' render ' est défini à ' true ' alors l'utilisateur est déjà connecté, dans ce cas celui-ci est redirigé vers la page d'accueil ' location: / '. Le tableau d'arguments ' render ' fait référence à la class Render qui a été développé afin d'encapsuler principalement la logique de gestion des templates, des paramètres de rendu et des configurations SEO. Celle-ci contient l'information de l'utilisateur connecté ou non afin de gérer le rendu des templates.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

```
// instance de UsersRegistration
$modelUser = new UsersRegistration($arguments);
```

Dans le cas où la condition précédente n'est pas remplie, alors on initialise la variable \$modelUser par une instance de UsersRegistration prenant le tableau \$arguments en paramètre. UsersRegistration est une classe étendue de PasswordHashManager et implémenté avec la classe JsonSerializable. UsersRegistration gérera la logique des données de l'utilisateur et passera le mot de passe enregistré sous hachage en utilisant la librairie Sodium.

```
// condition de requête non vide
if (!empty($_POST)) {
    $modelUser->setPassword(
        $arguments['password'] ?? ''
    );
    $errors = ReflectionValidator::validate($modelUser);
```

Ensuite avant d'exécuté la logique de creation d'un utilisateur, on vérifie que celui-ci lors du remplissage du formulaire n'envoie pas de données vide, même si la vérification s'effectue aussi du coté Client avec JavaScript. Si ce n'est pas vide, dans ce cas on passe le mot de passe en paramètre de la méthode setPassword en utilisant l'instance \$modelUser pour définir un mot de passe.

Par mesure de sécurité on passe une valeur vide '' au cas où \$arguments['password'] ne serait pas défini, sinon setPassword pourrait recevoir une valeur nulle ou non définie, ce qui pourrait causer des erreurs ou des comportements inattendus.

Enfin, on passe alors \$modelUser en paramètre de la méthode statique validate de la classe ReflectionValidator pour la gestion des erreurs. Cette méthode permet de valider les données de \$modelUser.



# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

```
// condition de vérification des erreurs et de email
if (!$errors && isset($arguments['email'])) {

    $crudManager = new CrudManager(
        'users',
        UsersRegistration::class
    );
}
```

Si dans le traitement du mot de passe il n'y a pas eu d'erreur et que l'email est bien présent dans les arguments, alors on défini la variable \$crudManager prennant comme valeur l'instance de la classe CrudManager avec en paramètre l'entité users et la class UsersRegistration. La classe CrudManager gère la logique des requêtes SQL.



# DOSSIER PROFESSIONNEL (DP)

```
// si l'email de l'inscription existe dans la base de donnée
if ($crudManager->getByEmail($arguments['email']) !== false) {
    throw new ClientExceptions(
        ClientExceptionEnum::AccountIsRegistered
    );
} else {

    // enregistrement de l'utilisateur
$modelUser->setPassword(
    $modelUser->hash($arguments['password'])
);

$crudManager->create(
    $modelUser, ['full_name', 'email', 'password', 'role']
);
//...
header('location:/connexion');
}

}
//...
}
```

Enfin la dernière étape consiste à vérifier si l'email défini dans arguments n'est pas déjà existant dans la base de donnée en utilisant la méthode getEmail du CrudManager, si c'est le cas alors on renvoi une exception dans la class ClientExceptions avec AccountIsRegistered comme paramètre de la méthode ClientExceptionEnum, afin d'informer l'utilisateur qu'un compte existe déjà avec cette email.



# DOSSIER PROFESSIONNEL (DP)

[Accueil](#) [Café](#) [Thé](#) [Contact](#)



[Inscription](#) [Connexion](#)

# Ooops

## 404

Compte déjà enregistré avec ce mail.

[← Retourner à TeaCoffee Accueil](#)



Dans le cas contraire on va pouvoir créer un utilisateur dans la base de donnée. Pour commencer nous effecturons un hachage du mot de passe défini en appellant la méthode hash de classe étendu UsersRegistration. Ensuite on effectue une requête de création de l'utilisateur en appellant la méthode create du CrudManager prennant l'instance \$modelUser ainsi qu'un tableau des données à créer. Après cette requête on renvoi l'utilisateur sur la page connection avec un message de bienvenue.

[Accueil](#) [Café](#) [Thé](#) [Contact](#)



[Inscription](#) [Connexion](#)

[Categories ▾](#)

### Connection à votre compte

Email

Mot de passe

[pas encore inscrit ?](#)



Bienvenue sur TeaCoffe  
Alejandro Despacito, merci de  
votre inscription.



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

Après l'inscription, c'est la méthode ConnectJS qui gérer la logique de connection du nouvelle utilisateur. Voici une description détaillée de la méthode **ConnectJS**

```
public function ConnectJS(...$arguments)
{
    //...
    // vérification de la connection utilisateur
    if ($render->has('isConnected') == true) {
        header('location:/user');
    }
    // instance de UserRegistration
    $modelUser = new UsersConnect($arguments);
```

Comme pour la méthode Register, nous vérifion la connection et définissons une instance de UsersRegistration.

```
// vérification de la requête
if (!empty($_POST)) {
    $modelUser->setPassword($arguments['password'] ?? "");
    $errorsIntercept = ReflectionValidator::validate($modelUser);
    $arrayIntersecKeyCompare = ['password' => ''];
    $errors = array_intersect_key($errorsIntercept, $arrayIntersecKeyCompare);
```

Ensute nous effectuons toujours une vérification de la requête, dans le cas où la condition est remplie, nous utilisons \$modelUser pour définir le mot de passe. On utilise toujours pour la validation des données la méthode validate. On définit un tableau associatif \$arrayIntersecKeyCompare avec une clé password et une valeur vide. Ce tableau est utilisé pour spécifier quelles erreurs nous intéressent dans le tableau d'erreurs interceptées. Enfin, on utilise la fonction array\_intersect\_key pour filtrer les erreurs pertinentes. Array\_intersect\_key retourne un tableau contenant toutes les valeurs de \$errorsIntercept dont les clés existent également dans \$arrayIntersecKeyCompare.



# DOSSIER PROFESSIONNEL (DP)

```
if ($errors) {
    $this->responseJson(200, ['errors' => $errors]);
}

if (!$errors) {
    $crudManager = new CrudManager('users', UsersRegistration::class);
    $user = $crudManager->getByEmail($modelUser->getEmail());

    if (!$user) {
        $this->responseJson(200, ['errors' => ['email' => 'Une erreur avec votre
email viens de ce produire.']]);
    }
}
```

Nous allons ensuite effectuer une série de condition pour la validation de la connection. Premièrement nous vérifirons à l'aide de la variable \$errors précédemment définie si il y a une erreur de champs. Si c'est le cas, alors on renvoi à l'utilisateur l'erreur. Sinon comme pour Register nous effectuons une vérification de l'email renseigné avec la colone email de la table Users. Puis une vérification de la présence de l'email, si celui-ci n'est pas défini alors nous renverrons l'information 'Une erreur avec votre email viens de ce produire'.

```
$verifPassword = new PasswordHashManager();

if ($verifPassword->verify($user->getPassword(), $arguments['password'])) {
    $arguments['render']->addSession([
        'email' => $user->getEmail(),
        'isConnected' => true,
        'full_name' => $user->getFull_name(),
        'role' => $user->getRole(),
    ]);
}

//...

$this->responseJson(200, ['isConnected' => true]);
} else {
    $this->responseJson(200, ['errors' => ['email' => "Il n'y a aucune
correspondance entre votre email et votre mot de passe."]]);
}
}

} else {
    $render->addParams('formConnect',
UsersRegistrationForms::ConnectFormUsersRegistration($modelUser, $errors ?? null));
}

}
```

Une fois que toute ces conditions on été remplies, on initialise une variable \$verifPassword par une instance de PasswordHashManager. Cette instance va nous permettre d'appeler la



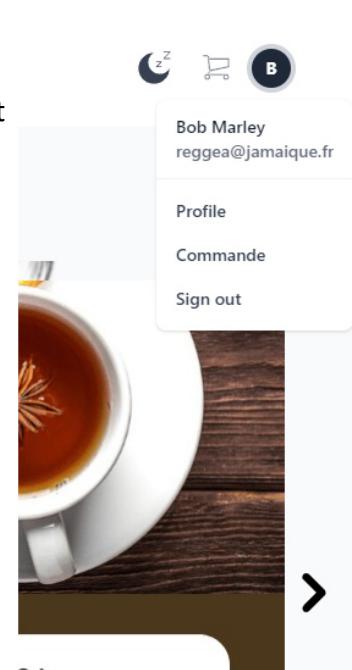
# DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ  
DE L'EMPLOI

méthode verify dans la condition suivante afin de vérifier le mot de passe entré par l'utilisateur et celui stocké en base de donnée. Si le mot de passe est correct alors on définit une Session dans les arguments de 'render'. Enfin, on renvoie une réponse comportant la valeur de ' isConnected' à 'true'.

Dans l'autre cas nous renverrons une erreur comportant 'Il n'y a aucune correspondance entre votre email et votre mot de passe' à l'utilisateur.

Le composant métier Register est conçu pour gérer efficacement et en toute sécurité l'inscription et la connexion des utilisateurs. En utilisant des pratiques de développement défensif, des validations rigoureuses et des techniques asynchrones, ce composant assure une expérience utilisateur fluide tout en maintenant une sécurité élevée.



## 2. Précisez les moyens utilisés :

Nous avons utilisé le langage PHP côté serveur lors du développement de ce composant métier. La librairie Sodium <https://doc.libsodium.org/> a été utilisé pour sécuriser les mots de passe. J'ai utilisé le site Carbon <https://carbon.now.sh/> pour la mise en page du code.



# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Ce projet a été réalisé en groupe.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *La Plateforme.*

Chantier, atelier, service ► *Boutique en ligne.*

Période d'exercice ► Du : *02/04/2024* au : *30/04/2024*

## 5. Informations complémentaires (facultatif)



# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

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

**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 :**

**2. Précisez les moyens utilisés :**

**3. Avec qui avez-vous travaillé ?**

**4. Contexte**

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

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

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

**5. Informations complémentaires (facultatif)**



# DOSSIER PROFESSIONNEL (DP)

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.



# DOSSIER PROFESSIONNEL (DP)

## Déclaration sur l'honneur

Je soussigné(e) Alexandre Detry ,  
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 Toulon le 18/07/2024

pour faire valoir ce que de droit.

Signature:



# DOSSIER PROFESSIONNEL (DP)

## Documents illustrant la pratique professionnelle

(*facultatif*)

### Intitulé

Cliquez ici pour taper du texte.



# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

*(Si le RC le prévoit)*