



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

- Chettouf

Nom d'usage

- Chettouf

Prénom

- Wahil

Adresse

- 102 rue douala, 83200 Toulon

Titre professionnel visé

Développeur Web Et Web Mobile

MODALITÉ D'ACCÈS :

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

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.

Ce titre est délivré par le Ministère chargé de l'emploi.

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

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

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

Pour prendre sa décision, le jury dispose :

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

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

Ce dossier comporte :

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

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

DOSSIER PROFESSIONNEL^(DP)

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

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1	p.	5
▪ Portfolio	p.	6
▪ Intitulé de l'exemple n° 2	p.	
▪ Intitulé de l'exemple n° 3	p.	
Intitulé de l'activité-type n° 2	p.	
▪ Boutique e-commerce	p.	11
▪ Blog	p.	25
▪ Intitulé de l'exemple n° 3	p.	
Intitulé de l'activité-type n° 3	p.	
▪ Intitulé de l'exemple n° 1	p.	
▪ Intitulé de l'exemple n° 2	p.	
▪ Intitulé de l'exemple n° 3	p.	
Intitulé de l'activité-type n° 4	p.	
▪ Intitulé de l'exemple n° 1	p.	
▪ Intitulé de l'exemple n° 2	p.	
▪ Intitulé de l'exemple n° 3	p.	
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	
Déclaration sur l'honneur	p.	

DOSSIER PROFESSIONNEL (DP)

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1	p.	5
▸ Portfolio	p. p.	6
▸ Intitulé de l'exemple n° 2	p. p.	
▸ Intitulé de l'exemple n° 3	p p.	
Intitulé de l'activité-type n° 2	p.	
▸ Boutique e-commerce	p. p.	11
▸ Blog	p. p.	25
▸ Intitulé de l'exemple n° 3	p p.	
Intitulé de l'activité-type n° 3	p.	
▸ Intitulé de l'exemple n° 1	p. p.	
▸ Intitulé de l'exemple n° 2	p. p.	
▸ Intitulé de l'exemple n° 3	p p.	
Intitulé de l'activité-type n° 4	p.	
Documents illustrant la pratique professionnelle (facultatif)	p.	
Annexes (Si le RC le prévoit)	p.	

DOSSIER PROFESSIONNEL^(DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

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

Exemple n°1 - Développer un portfolio

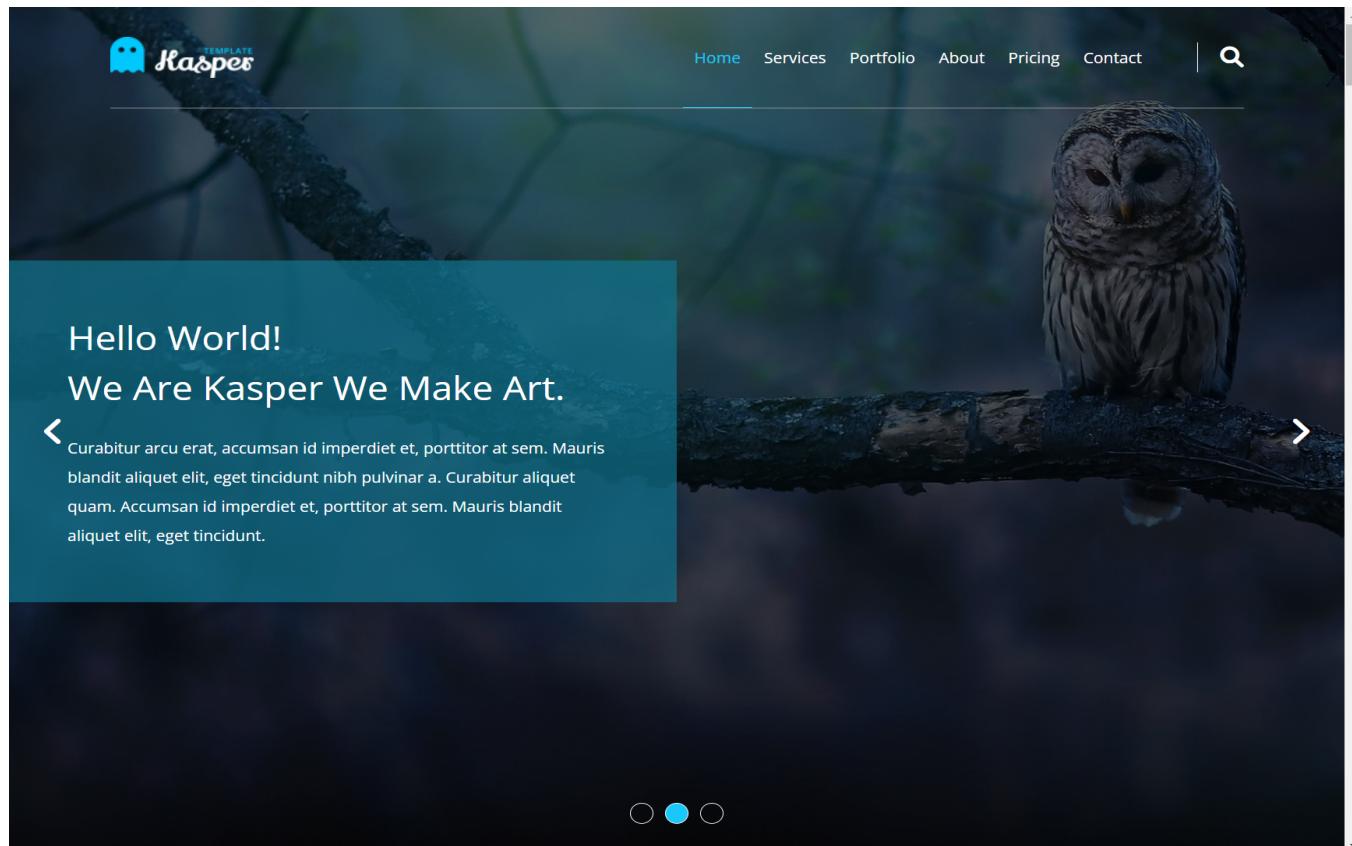
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Le but était de créer un portfolio CV pour un projet de la formation afin de regrouper les différents projets que je vais accomplir durant l'année et faire un support CV à présenter.

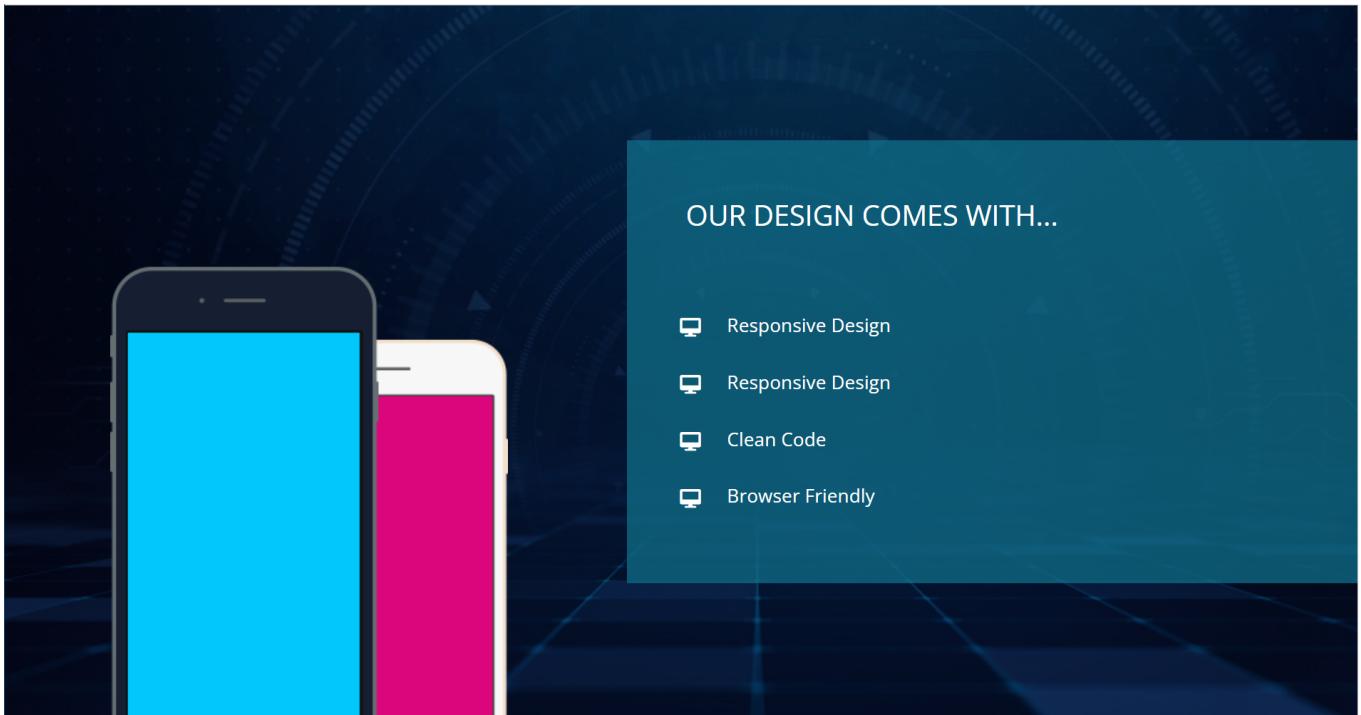
Avant d'attaquer la réalisation de maquette, j'ai listé les différents partie de site:

- Homme
- Services
- Portfolio
- Pricing
- Contact

Une fois terminé ça j'ai commencé le maquettage de mon site.



DOSSIER PROFESSIONNEL (DP)



OUR DESIGN COMES WITH...

- Responsive Design
- Responsive Design
- Clean Code
- Browser Friendly

```
<!-- ***** Main ***** -->
<main>
    <!-- ***** Landing ***** -->
    <section class="landing flex-r" id="landing">
        <div class="overlay"></div>
        <i class="fas fa-angle-left change-background fa-2x left"></i>
        <article class="art-landing flex-r">
            <div class="cadre-descript flex-c">
                <h2>Hello World!<br> We Are Kasper We Make Art.</h2>
                <p>Curabitur arcu erat, accumsan id imperdiet et, porttitor at sem. Mauris blandit aliquet elit, eget tincidunt nibh pulvinar a. Curabitur aliquet quam. Accumsan id imperdiet et, porttitor at sem. Mauris blandit aliquet elit, eget tincidunt.</p>
            </div>
        </article>
        <i class="fas fa-angle-right change-background fa-2x right"></i>
        <ul class="page flex-r">
            <li class=""></li>
            <li class="active"></li>
            <li class=""></li>
        </ul>
    </section>
```

DOSSIER PROFESSIONNEL (DP)

```
@import url('https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;0,800;1,300;1,400;1,500;1,600;1,700&display=swap');
```

```
* {  
    padding:0%;  
    margin:0;  
    box-sizing: border-box;  
    scroll-behavior: smooth;  
}  
  
/* ---- For All Title ---- */  
.main-heading {  
    width: 100%;  
    text-align: center;  
}  
  
.main-heading h2 {  
    position: relative;  
    font-weight: normal;  
    font-size: 40px;  
    text-transform: uppercase;  
    color: #000;  
    padding-bottom: 12px;  
}
```

DOSSIER PROFESSIONNEL (DP)

```
<!-- ***** Header ***** -->
<header class="flex-r">
  <div class="content-header flex-r">
    <!-- Logo -->
    <div class="cadre-logo flex-r">
      <a href="#"></a>
    </div>
    <!-- Menu -->
    <nav class="main-menu flex-r">
      <i class="humberger fas fa-bars toggle-menu"></i>
      <ul class="flex-r">
        <li class=""><a href="#">Home</a></li>
        <li class=""><a href="#services">Services</a></li>
        <li class=""><a href="#portfolio">Portfolio</a></li>
        <li class=""><a href="#about">About</a></li>
        <li class=""><a href="#pricing">Pricing</a></li>
        <li class=""><a href="#contact">Contact</a></li>
      </ul>
      <div class="cadre-search flex-r">
        <i class="fas fa-search"></i>
      </div>
    </nav>
  </div>
</header>
```

2. Précisez les moyens utilisés :

Pour ce projet, je disposais d'un poste informatique sous Windows 10.

J'ai travaillé sur Visual Studio Code et je me suis servi de mon navigateur Internet pour tester mon site en local et ajuster mon code CSS depuis l'inspecteur.

GIT/GITHUB / W3School

- Importation des polices :

<https://fonts.googleapis.com/>

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

J'ai travaillé seul sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Chantier, atelier, service ▶ Portfolio / wahil chettouf

Période d'exercice ▶ Du : 24/05/2022 au : **06/06/2022**

5. Informations complémentaires (*facultatif*)

J'ai réalisé ce projet en autodidacte, avant de commencer la formation.

github :

<https://github.com/wahil09/Portfolio-Designer-web-2>

page d'exemple :

<https://wahil09.github.io/Portfolio-Designer-web-2/>

autre projet :

<https://wahil09.github.io/Portfolio-Designer-Web/>

autre projet:

<https://wahil-chettouf.students-laplateforme.io/blog/index.php>

autre projet:

<https://wahil09.github.io/HTML-And-CSS-Template-3/>

DOSSIER PROFESSIONNEL (DP)

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

Activité-type 2

Exemple n° 1 - Ecommerce

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

Dans le cadre de ma formation de développeur web et web mobile, j'ai réalisé un projet d'Application Programming Interface (API) pour une application de commerce électronique (e-commerce).

1.1. Conceptualisation de la base de données (MCD - Modèle Conceptuel de Données) :

- J'ai débuté par la conceptualisation de la base de données en utilisant le Modèle Conceptuel de Données (MCD). J'ai défini les différentes tables nécessaires pour l'e-commerce telles que "clients", "products", "categories" et "adresses". J'ai également établi les relations entre ces tables pour représenter les liens entre les entités.

Table	Action	Lignes	Type
<input type="checkbox"/> tbl_address		6	InnoDB
<input type="checkbox"/> tbl_categories		2	InnoDB
<input type="checkbox"/> tbl_cat_prod		1	InnoDB
<input type="checkbox"/> tbl_order		5	InnoDB
<input type="checkbox"/> tbl_order_card		2	InnoDB
<input type="checkbox"/> tbl_product		6	InnoDB
<input type="checkbox"/> tbl_product_image		27	InnoDB
<input type="checkbox"/> tbl_top_category		2	InnoDB
<input type="checkbox"/> tbl_user		4	InnoDB
9 tables	Somme		55 InnoDB

DOSSIER PROFESSIONNEL (DP)

←→	cd_id	product_id	cat_id
<input type="checkbox"/>  Éditer  Copier  Supprimer	1	115	1

1.2. Transformation en MLD (Modèle Logique de Données) :

- Ensuite, j'ai transformé le MCD en Modèle Logique de Données (MLD) en précisant les types de données, les clés primaires et étrangères, ainsi que les contraintes d'intégrité pour chaque table.

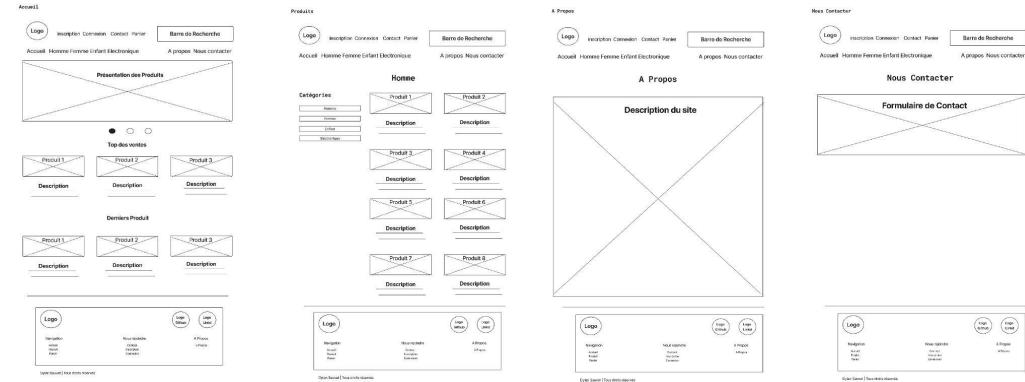
1.3. Création de la base de données :

- Une fois le MLD finalisé, j'ai créé la base de données en utilisant PHPMyAdmin.

1.4. Conception de la maquette :

- J'ai utilisé l'outil de conception graphique Figma pour créer une maquette simple de l'interface utilisateur de l'application e-commerce. Cela m'a permis de visualiser la disposition des éléments et l'expérience utilisateur.

● Simple exemple:



1.5. Gestion de projet avec Trello :

- J'ai utilisé l'outil de gestion de projet Trello pour organiser la progression de mon projet. J'ai détaillé les tâches à accomplir, établi des listes et des étapes, et suivi la progression de chaque tâche.

Example :

DOSSIER PROFESSIONNEL (DP)

Liste : Terminé

- Tâche 1 : Conceptualisation de la base de données

Description : Créer le Modèle Conceptuel de Données (MCD) pour le projet e-commerce en identifiant les entités, les attributs et les relations entre les tables.

Échéance : [Date]

- Tâche 2 : Transformation en Modèle Logique de Données (MLD)

Description : Convertir le MCD en Modèle Logique de Données (MLD) en précisant les types de données, les clés primaires et étrangères, ainsi que les contraintes d'intégrité.

Échéance : [Date]

- Tâche 3 : Création de la base de données

Description : Mettre en place la base de données en utilisant PHPMyAdmin ou tout autre outil de gestion de base de données.

Échéance : [Date]

- Tâche 4 : Conception de la maquette

Description : Utiliser Figma pour créer une maquette simple de l'interface utilisateur de l'application e-commerce.

Échéance : [Date]

Liste : En cours

- Tâche 5 : Développement du backend - Classe "Client"

Description : Créer la classe "Client" avec les propriétés et les méthodes nécessaires pour gérer les opérations liées aux clients (connexion, inscription, modification de profil, etc.).

Échéance : [Date]

- Tâche 6 : Développement du backend - Classe "Product"

Description : Créer la classe "Product" avec les méthodes pour récupérer tous les produits et récupérer un produit par son ID.

Échéance : [Date]

- Tâche 7 : Développement du backend - Classe "Category"

Description : Créer la classe "Category" avec les méthodes pour récupérer toutes les catégories et récupérer les produits d'une catégorie donnée.

Échéance : [Date]

- Tâche 8 : Développement du backend - Classe "Address"

Description : Créer la classe "Address" pour gérer les adresses des clients.

Échéance : [Date]

Liste : En attente de vérification

DOSSIER PROFESSIONNEL (DP)

- Tâche 9 : Implémenter la sécurité - Classe "Authentication"
Description : Créer la classe "Authentication" pour gérer les vérifications des entrées et sorties du site (formulaires, requêtes, etc.).
Échéance : [Date]
- Tâche 10 : Tester les requêtes Fetch API
Description : Tester les requêtes Fetch API pour les opérations CRUD avec l'API PHP.
Échéance : [Date]

Liste : À faire

- Tâche 11 : Sécurité des mots de passe
Description : Mettre en place le hachage des mots de passe pour sécuriser les données des utilisateurs.
Échéance : [Date]
- Tâche 12 : Vérification des autorisations
Description : Vérifier si l'utilisateur est connecté et s'il est un administrateur ou un simple client pour restreindre certaines fonctionnalités.
Échéance : [Date]
- Tâche 13 : Finalisation du projet
Description : Effectuer les dernières vérifications, corriger les bugs éventuels et préparer la présentation du projet.
Échéance : [Date]

1.6. Développement du backend :

- J'ai commencé par développer le backend de l'API e-commerce. J'ai créé des classes pour les entités du projet, telles que "Client", "Product", "Category" et "Address", avec les propriétés et les méthodes nécessaires pour gérer les opérations liées à chaque entité.

DOSSIER PROFESSIONNEL (DP)

```
if(!session_id()) session_start();
require_once("Bdd.php");

class Utilisateurs {
    const TB_NAME = "tbl_user";
    private $is_connected = false;
    private $user_info = [];

    public function __construct()
    {
        if(isset($_SESSION["id"])){
            $this->user_info = self::getUserById($_SESSION["id"]);

            if($this->user_info) {
                $this->is_connected = true;
            } else {
                $this->is_connected = false;
                $this->disconnect();
            }
        }
    }
}
```

- Par exemple, pour la classe "Client", j'ai créé des méthodes telles que "connect()" pour la connexion, "add_user()" pour l'inscription, etc. Ces méthodes effectuent les vérifications nécessaires sur les données entrées par les utilisateurs et effectuent les opérations CRUD (Create, Read, Update, Delete) correspondantes dans la base de données.

```
public function add_user($full_name, $email, $password, $phone, $role) {
    global $bdd;
    $sql = "INSERT INTO ". self::TB_NAME ."(full_name, email, password, phone,
    role, status) VALUES(?, ?, ?, ?, ?, false)";
    $req = $bdd->prepare($sql);
    $req->execute([$full_name, $email, $password, $phone, $role]);

    if($req->rowCount()) {
        return true;
    } else {
        return false;
    }
}
```

DOSSIER PROFESSIONNEL (DP)

```
public function connect($email, $password) {
    // Récupérer le mot de passe de l'utilisateur dans la base de données et le
    // comparer avec le mot de passe entré par l'utilisateur
    $passwordHashed = self::getPasswordByEmail($email);
    // Vérifier si le mot de passe entré par l'utilisateur correspond au mot de
    // passe dans la base de données
    if(password_verify($password, $passwordHashed)) {
        $this->is_connected = true;
        $this->user_info = self::getUserByEmail($email);
        $_SESSION["id"] = $this->user_info->id;
        return true;
    } else {
        return false;
    }
}
```

- J'ai également créé une classe "Authentication" pour gérer la sécurité du site. Cette classe vérifie tous les entrées et sorties du site en utilisant ces méthodes, comme les formulaires de connexion, d'inscription, d'édition de profil, d'ajout de produit, etc. Les données entrantes sont filtrées et validées pour éviter les failles de sécurité telles que les injections SQL.

```
class Authentication {
    // Filtrer les données entrées par l'utilisateur
    public static function process_input($inp) { ... }

    // Vérifier si le mot de passe est correct
    public static function verify_password($password, $hash) { ... }

    // Hacher le mot de passe
    public static function hash_password($password) { ... }

    // Vérifier si le mot de passe contient au moins 8 caractères, une lettre
    // majuscule, une lettre minuscule, un chiffre et un caractère spécial
    public static function is_password_strong($password) { ... }

    // Vérifier si le numéro de téléphone est valide
    public static function is_phone_valid($phone) { ... }
}
```

DOSSIER PROFESSIONNEL (DP)

```
public static function process_input($inp)
) {
    $inp = trim($inp);
    $inp = stripslashes($inp);
    $inp = htmlspecialchars($inp);
    return $inp;
}
```

```
public static function is_password_strong($password) {
    $uppercase = preg_match('@[A-Z]@', $password);
    $lowercase = preg_match('@[a-z]@', $password);
    $number = preg_match('@[0-9]@', $password);
    $special_char = preg_match('@[^w]@', $password);

    if (!$uppercase || !$lowercase || !$number || !$special_char || strlen($password) < 8) {
        return false;
    } else {
        return true;
    }
}
```

1.7. Utilisation de Fetch API :

- Pour récupérer et envoyer des données depuis et vers l'API, j'ai utilisé l'API Fetch en JavaScript.

DOSSIER PROFESSIONNEL (DP)

```
connectionForm.addEventListener('submit', (e) => {
    e.preventDefault();
    const formData = new FormData(connectionForm);

    fetch("../src/api/connexion.php", {
        method: "POST",
        body: formData,
        headers: {
            'Accept': 'application/json'
        }
    })
    .then(response => response.json())
    .then(data => {
        process_data(data)
    })
    .catch(error => console.log(error))
});
```

- Dans mon API PHP, j'ai traité les requêtes avec des conditions et des filtrages de données pour éviter que des données incorrectes soient envoyées. Par exemple, j'ai vérifié que les champs requis ne sont pas vides et que les données correspondent au type attendu.

DOSSIER PROFESSIONNEL (DP)

```
if($_SERVER["REQUEST_METHOD"] == "POST") {
    if(isset($_POST["email"]) && !empty(trim($_POST["email"]))) {
        if(filter_var(Authentication::process_input($_POST["email"]),
        FILTER_VALIDATE_EMAIL)) {
            // Vérification si l'utilisateur existe déjà
            if(Utilisateurs::is_user_exist(Authentication::process_input($_POST
            ["email"]))) {
                $email = Authentication::process_input($_POST["email"]);
            } else {
                $email_err = "Cette email n'existe pas";
            }
            $email = Authentication::process_input($_POST["email"]);
        } else {
            $email_err = "Veuillez entrer une adresse email valide";
        }
    } else {
        $email_err = "Veuillez entrer votre adresse email";
    }
}
```

- En cas de requête réussie, l'API renvoie les données sous forme de format JSON pour que JavaScript puisse les traiter et les afficher à l'utilisateur sans rafraîchissement de la page.

DOSSIER PROFESSIONNEL (DP)

```
if(empty($email_err) && empty($password_err)) {
    $user->connect($email, $password);
    if($user->isConnected()) {
        $response = array(
            "success" => "Vous êtes connecté"
        );
        echo json_encode($response);
    } else {
        $response = array(
            "error" => "email ou mot de passe incorrect"
        );
        echo json_encode($response);
    }
} else {
    // Afficher les erreurs en utilisant fetch en javascript
    $response = array(
        "email_err" => $email_err,
        "password_err" => $password_err,
    );

    echo json_encode($response);
}
```

1.8. Sécurité et protection des données :

- J'ai assuré la sécurité du site en utilisant un hachage simple pour les mots de passe (un chiffrement plus puissant sera mis en place ultérieurement). Les fichiers sensibles du projet sont protégés en utilisant le fichier .htaccess pour restreindre l'accès aux dossiers sensibles.
 - Utiliser : htmlspecialchars() / min, max, required, pattern / trim(), strlen(), is_numeric(), filter_var(), filter_var() ...

DOSSIER PROFESSIONNEL (DP)

```
// Vérifier si le mot de passe est correct
public static function verify_password($password, $hash) {
    return password_verify($password, $hash);
}

// Hacher le mot de passe
public static function hash_password($password) {
    return password_hash($password, PASSWORD_DEFAULT);
}
```

- Simple exemple pour htaccess :

```
<Files *.php>
|   allow from all
</Files>
deny from all
```

Ca pour éviter de voir ca :

	Parent Directory	-
	commandes/	2023-07-10 12:36 -
	inc/	2023-07-10 12:36 -
	panier/	2023-07-10 12:36 -
	profile/	2023-07-10 12:36 -

- J'ai utilisé des requêtes préparées (prepared statements) pour éviter les attaques par injection SQL et optimiser les performances lorsque la même requête est exécutée plusieurs fois par la même session.

DOSSIER PROFESSIONNEL (DP)

```
public static function addAddress($address_line_1, $address_line_2, $city, $state,
$zip_code, $country) {
    global $bdd, $user;
    $sql = "INSERT INTO ". self::TB_NAME . " (user_id, address_line_1,
address_line_2, city, state, zip_code, country) VALUES (?, ?, ?, ?, ?, ?, ?)";
    |
    $req = $bdd->prepare($sql);
    $req->execute([$user->getId(), $address_line_1, $address_line_2, $city, $state,
$zip_code, $country]);
    if($req->rowCount() > 0) {
        return true;
    } else {
        return false;
    }
}
```

Utilisation de Git/GitHub :

- J'ai utilisé Git/GitHub pour gérer le contrôle de version de mon projet. J'ai créé un référentiel Git pour suivre les modifications du code source tout au long du développement du projet. J'ai effectué des commits réguliers pour garder une trace des modifications apportées au code.

The screenshot shows a GitHub repository interface. At the top, there's a commit history for July 10, 2023, with three commits from the user 'wahil-chettouf':

- "resoudre le probleme de style" (resolved the style problem)
- "probleme de style.css" (problem with style.css)
- "Merge pull request #5 from wahil-chettouf/userProfile" (merged pull request #5 from wahil-chettouf/userProfile)

Below the commit history is a section titled "Your branches" which lists five branches:

Branch	Last Commit	Merge Status	Actions
userProfile	Updated 2 weeks ago by wahil-chettouf	#5 Merged	View Details
Dashboard_Product	Updated 2 weeks ago by wahil-chettouf	#4 Merged	View Details
product	Updated last month by wahil-chettouf	#3 Merged	View Details
model-de-connexion	Updated 2 months ago by wahil-chettouf	#2 Merged	View Details
wahil	Updated 2 months ago by wahil-chettouf	#1 Merged	View Details

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Langages de programmation :

- HTML, Tailwind CSS et JavaScript pour le développement du front-end de l'e-commerce.
- l'API Fetch.
- PHP pour le développement du back-end de l'API.
- Programmation Orientée Objet (OOP) .
- SQL pour les opérations sur la base de données MySQL.

Technologies :

- Git/GitHub : Système de contrôle de version utilisé pour gérer le code source du projet et faciliter la collaboration.
- Draw.io : Outil de création de diagrammes utilisé pour concevoir le Modèle Conceptuel de Données (MCD) de la base de données.

3. Avec qui avez-vous travaillé ?

J'ai d'abord travaillé sur ce projet au sein d'un groupe, puis j'ai continué à travailler individuellement à cause de difficultés personnelles.

4. Contexte

Nom de l'entreprise, organisme ou association ➤ La-plateforme

Chantier, atelier, service ➤ Projet réalisé dans le cadre de ma formation en développement web et web mobile.

Période d'exercice ➤ Du : 15/06/2023 au : 23/07/2023

5. Informations complémentaires (facultatif)

Ce projet d'e-commerce m'a permis de mettre en pratique mes compétences en développement web et web mobile, en utilisant une combinaison de langages et de technologies tels que HTML, Tailwind CSS, JavaScript, PHP et MySQL. J'ai acquis de l'expérience dans la création d'une API pour gérer les données de l'e-commerce, ainsi que dans la gestion de base de données relationnelles avec SQL. L'utilisation de Git/GitHub a facilité le suivi des modifications du code et le travail en collaboration avec d'autres développeurs. Enfin, la création du Modèle Conceptuel de Données (MCD) avec Draw.io m'a permis de planifier efficacement la structure de la base de données avant sa mise en œuvre. Ce projet a été une

DOSSIER PROFESSIONNEL ^(DP)

opportunité précieuse pour développer mes compétences professionnelles en développement web et me préparer pour une carrière dans ce domaine.

DOSSIER PROFESSIONNEL (DP)

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

Activité-type 3

Exemple n° 1 - Blog

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

Dans le cadre de ma formation de développeur web et web mobile, j'ai réalisé un projet de blog en utilisant les technologies HTML, CSS, JS, PHP, MySQL et Git/GitHub. Voici une description détaillée des tâches effectuées :

The screenshot shows a dark-themed Admin Panel interface. On the left, there's a sidebar with three menu items: "Manage Posts", "Manage Users", and "Manage Categories". At the top right, it says "Wahil Chettouf". In the center, there are two buttons: "Add Post" and "Manage Posts". Below them is a table titled "Manage Posts" with the following data:

N	title	author	action
5	cristiano-ronaldo -> le meilleur joueur du monde	admin	Edit Delete
6	Python	admin	Edit Delete
7	Counter-Strike	admin	Edit Delete
8	javascript	admin	Edit Delete

DOSSIER PROFESSIONNEL (DP)

Admin Panel

Wahil Chettouf

Manage Posts Add Post Manage Posts

Create Post

Article Name :

Categories :

Article Image : Aucun fichier choisi

Conceptualisation de la base de données (MCD - Modèle Conceptuel de Données) :

- J'ai commencé par conceptualiser la base de données du blog en utilisant le Modèle Conceptuel de Données (MCD). J'ai identifié les entités principales telles que "users" (utilisateurs), "posts" (articles), "categories" (catégories) et "comments" (commentaires), ainsi que leurs attributs et relations.

Table	Action	Lignes	Type	Interclassement
<input type="checkbox"/> categories		4	InnoDB	utf8mb4_general_ci
<input type="checkbox"/> posts		3	InnoDB	utf8mb4_general_ci
<input type="checkbox"/> users		2	InnoDB	utf8mb4_general_ci
3 tables	Somme	9 InnoDB utf8mb4_general_ci		

Création de la base de données :

- Une fois le MCD établi, j'ai créé la base de données en utilisant PHPMyAdmin. J'ai créé les tables nécessaires avec les relations définies dans le MLD.

DOSSIER PROFESSIONNEL (DP)

Conception de la maquette :

- J'ai utilisé l'outil de conception graphique Figma pour créer une maquette simple de l'interface utilisateur du blog. Cela m'a permis de visualiser la mise en page et l'expérience utilisateur.

N	Name	action
1	Bloger	Edit Delete
2	informatique	Edit Delete
3	FootBall	Edit Delete
4	Game	Edit Delete

Gestion de projet avec Trello :

- J'ai utilisé l'outil de gestion de projet Trello pour organiser la progression de mon projet.

Développement du backend :

- J'ai commencé par développer le backend du blog en utilisant PHP. J'ai créé des classes pour les entités principales du projet telles que "User", "Post", "Category" et "Comment". Chaque classe comprenait les propriétés correspondantes aux attributs de la table et les méthodes nécessaires pour gérer les opérations CRUD.
- Par exemple, pour la classe "User", j'ai créé des méthodes telles que "register()" pour l'inscription d'un nouvel utilisateur, "login()" pour la connexion, "updateProfile()" pour mettre à jour le profil de l'utilisateur, etc.

```
class ModelUsers extends Connection {
    public $tbname = "users";

    public function userConnect($email, $password) {
        $request = $this->db->prepare("SELECT * FROM ". $this->getTableName() . " WHERE
            password=:password &&email=:email");
        $request->bindValue("email", $email);
        $request->bindValue("password", $password);
        $request->execute();
        return $request->fetchObject(); // return un object s'il exist ou rien s'il
        n'est pas inscrit déjà
    }
}
```

DOSSIER PROFESSIONNEL (DP)

- Dans ces méthodes, j'ai effectué des vérifications sur les données entrées par les utilisateurs, comme la validation de l'adresse e-mail, la vérification des mots de passe, etc. J'ai ensuite utilisé des requêtes SQL pour effectuer les opérations CRUD dans la base de données en fonction des actions de l'utilisateur.

```
function setUser($username, $email, $password, $role = "user") {  
    $request = $this->db->prepare("INSERT INTO ". $this->getTableName() ."  
        (username, email, password, role) VALUES(?, ?, ?, ?)");  
    $request->bindParam(1, $username);  
    $request->bindParam(2, $email);  
    $request->bindParam(3, $password);  
    $request->bindParam(4, $role);  
    $request->execute();  
    return $request->fetch(PDO::FETCH_ASSOC);  
}
```

Sécurité et protection des données :

- Toutes les données entrantes étaient filtrées et échappées pour éviter les attaques par injection SQL. J'ai également utilisé des sessions PHP pour gérer l'authentification des utilisateurs et restreindre l'accès à certaines fonctionnalités en fonction des autorisations.
- avant chaque requête sql je confirme que les données entrer ou envoyer qui sont valide ex:
 - qui ne sont pas vides (si nécessaire).
 - le type correspondant aux données attendues.
 - Utiliser : htmlspecialchars() / isset() / empty() / min, max, required

```
<p class='post-description'><?php echo htmlspecialchars  
($post['postContent']) ?></p>  
  
if(isset($_POST["save"])) {  
    if(isset($_POST["postTitle"], $_POST["Categories"], $_POST["postContent"], $_FILES  
    ["imageToUpload"])) {  
        if(!empty($_POST["postTitle"]) && !empty($_POST["Categories"]) && !empty($_POST  
        ["postContent"]) && $_FILES["imageToUpload"]["size"] != 0) {  
            include $adminPathInclude."inc/upload_image.php";
```

Si tout les condition sont correcte :

```
$postsModel->savePost($postTitle, $postCat, $postImage, $pos  
$postCategoryId);
```

Utilisation de Git/GitHub :

DOSSIER PROFESSIONNEL (DP)

- J'ai utilisé Git/GitHub pour gérer le contrôle de version de mon projet. J'ai créé un référentiel Git pour suivre les modifications du code source tout au long du développement du projet. J'ai effectué des commits réguliers pour garder une trace des modifications apportées au code.



Commits on Jul 21, 2023

changer le type de column user->inscriptionData



wahil09 committed 20 hours ago



Commits on Jul 20, 2023

ajouter la base de données Fichier blog.sql



wahil09 committed yesterday ✓

Merge pull request #3 from wahil09/adminPanel ...



wahil09 committed yesterday

modifier le nom de la base de données wahil -> blog



wahil-chettouf committed yesterday

fin du projet



wahil-chettouf committed yesterday

DOSSIER PROFESSIONNEL (DP)

Default branch

main Updated 20 hours ago by wahil09

Default

Active branches

adminPanel Updated yesterday by wahil-chettouf

3 | 0

Stale branches

essay Updated 6 months ago by wahil09

89 | 0

2. Précisez les moyens utilisés :

- Langages de programmation : HTML, CSS, JavaScript, PHP
- Base de données : MySQL
- Outils de conception : Figma
- Outil de gestion de projet : Trello
- Outil de contrôle de version : Git/GitHub

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association ➔ La-plateforme

Chantier, atelier, service ➔ Projet réalisé dans le cadre de ma formation en développement web et web mobile.

Période d'exercice ➔ Du : xx/02/2023 au : xx/06/2023

5. Informations complémentaires (facultatif)

Ce projet a été réalisé sur une période de plusieurs semaines, en travaillant dessus de manière intermittente en raison de contraintes de temps liées à ma formation et à d'autres engagements.

DOSSIER PROFESSIONNEL^(DP)

Le lien vers le site du projet est le suivant :

<https://wahil-chettouf.students-laplateforme.io/blog/index.php>

admin email: admin@admin.com

password email : Admin@123

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

DOSSIER PROFESSIONNEL^(DP)

Déclaration sur l'honneur

Je soussigné Chettouf Wahil

,

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

Fait à Toulon

le 21/07/2023

pour faire valoir ce que de droit.



Signature :

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)