|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | Plazzotta |
| *Prénom* |  | Aurélien |
| *Adresse* |  | 183 rue de l’Aumône 45160 OLIVET |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| Développeur web et web mobile | |
|  | |
| **Modalité d’accès :** | |
|  | |
|  | Parcours de formation |
|  | Validation des Acquis de l’Expérience (VAE) |
|  | |

**1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Présentation du dossier** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l’emploi.**  Le DP appartient au candidat. Il le conserve, l’actualise durant son parcours et le présente **obligatoirement à chaque session d’examen**.  Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.  Il est consulté par le jury au moment de la session d’examen. | |  |
|  | | | |
| **Pour prendre sa décision, le jury dispose :** | | | |
| 1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l’entretien professionnel ou de l’entretien technique ou du questionnement à partir de productions. 2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle 3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d’un parcours de formation 4. de l’entretien final (dans le cadre de la session titre).   *[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels*  *du ministère chargé de l’Emploi]* | | | |
| **Ce dossier comporte :** | | | |
| * pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ; * un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d’un titre, d’un diplôme, d’un certificat de qualification professionnelle (CQP) ou des attestations de formation ; * une déclaration sur l’honneur à compléter et à signer ; * des documents illustrant la pratique professionnelle du candidat (facultatif) * des annexes, si nécessaire. | | | |
| *Pour compléter ce dossier, le candidat dispose d’un site web en accès libre sur le site.* | | | |
|  | | [**http://travail-emploi.gouv.fr/titres-professionnels**](http://travail-emploi.gouv.fr/titres-professionnels) | |

**2**

**3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sommaire** | | | |
| **Exemples de pratique professionnelle** | | | |
| **Développer la partie dynamique des interfaces utilisateur web ou web mobile** | | **p.** | **5** |
|  |  Réaliser des interfaces utilisateur statiques web ou web mobile p. | p. | 5 |
|  |  Installer et configurer son environnement de travail en fonction du projet web ou web mobile p. | p. | 8 |
|  |  |  |  |
|  |  |  |  |
| **Développer la partie back-end d’une application web ou web mobile sécurisée** | | **p.** | **11** |
|  |  Développer des composants d’accès aux données SQL  p. | p. | 11 |
|  |  Développer des composants métier côté serveur  p. | p. | 14 |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | p. | 17 |
| **Déclaration sur l’honneur** | | p. | 18 |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | p. |  |
| **Annexes** *(Si le RC le prévoit)* | | p. |  |

**4**

**Exemples de pratique**

**professionnelle**

**5**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **1** | | | **Développer la partie dynamique des interfaces utilisateur web ou web mobile** | | | | |
| ***Exemple n°1***  | | | | ***Réaliser des interfaces utilisateur statiques web ou web mobile*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Après écoute et extraction des besoins exprimés par Carlos de Sousa, chef d’entreprise, il est convenu d’intégrer un diaporama aléatoire de photographies afin d’égayer la page d’accueil présenter un porte-folio aux visiteurs grâce à l’introduction d’un **diaporama d’images**.  Je crée un diagramme de cas d’utilisations grâce au langage de modélisation **SysML** (successeur de UML) en usant de l’outil en ligne draw.io afin d’identifier la liste des fonctionnalités à intégrer.  Je crée un diagramme de navigation pour formaliser l’interaction nouvelle, rendue possible par la mise en place du diaporama, entre la page d’accueil et la page des chantiers réalisés.  Commence alors la phase de maquettage du projet logiciel. Le but est de communiquer visuellement le positionnement et l’exécution séquentielle des actions entre les différents composants d’une application et s’assurer *in fine*, de l’adéquation de la solution proposée et des besoins du client.  Grâce à l’outil en ligne figma.com, je crée alors la première phase de maquettage, dite basse fidélité, ou zonage, chargée d’identifier les parties fixes (contenu éditorial immuable) et les parties variables (contenu généré programmatiquement, savoir dynamique).  Je produis ensuite un fil de fer, phase dite moyenne fidélité, présentant la typologie des composants (i.e. images, texte, champ de formulaire, bouton, etc.).  J’ai écrit une page web avec le langage de structuration des données **HTML** afin d’introduire le contenu éditorial associé au domaine d’activité de l’entreprise. Puis j’ai écrit une feuille de style avec le langage de présentation des données **CSS** pour assurer un squelette de modèle de page ergonomique et intuitif ainsi qu’une couche visuelle attrayante et respectueuse de la charte graphique de l’organisation.  Puis ai-je mis en place une fonctionnalité de diaporama dynamique de photographies des chantiers effectués par cette même organisation grâce au langage de programmation orienté web **JavaScript** et le cadriciel CSS Bootstrap.  Ce diaporama est respectueux des bonnes pratiques de codage en adoptant le mode strict de JavaScript, qui permet d’empêcher l’emploi des mots clefs réservés pour de futures publications du standard ECMAScript (responsable de l’évolution des spécifications techniques de JavaScript), la levée d’exceptions, permet d’identifier et capturer des erreurs jusqu’alors insoupçonnées, car autorisées silencieusement et enfin de neutraliser le comportement anciennement par défaut de mauvaises pratiques, comme les déclarations multiples d’un même nom de variables.  Accéder à la page accueil du site web démarre automatiquement le diaporama, qui inclue un temporisateur évitant ainsi de générer une pollution visuelle et offre aux visiteurs un laps de temps avant de focaliser son attention sur les illustrations.  Le diaporama présente les particularités suivantes :   * Affichage des images aléatoirement pour servir aux visiteurs un large spectre de chantiers disponibles afin de communiquer richesse et adaptabilité aux prospects. * Glisser le curseur sur une image suspend temporairement le défilé du diaporama. * Cliquer sur une photographie redirige le visiteur vers la page détaillée du chantier. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| * Editeur de texte **Helix, Visual Studio Code** * Système de contrôle de source **git** * Gestion du dépôt distant via **github.com** * Outil de containerization**Docker** * Transferts de fichiers en protocole FTP via **FileZilla** * Outil de maquettage en ligne **trello.com** * Outil de modélisation en ligne **drawio.net** * Téléphone * Outil de visio-conférence **Microsoft Teams** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| Le chef d’entreprise, Monsieur Carlos de Sousa, suggère d’implémenter un diaporama d’images pour servir de vitrine commerciale et assurer aux prospects le respect de notre engagement qualité.  Les pré-requis techniques sont libres, n’étant point limités par l’existant car le site web est nouveau. C’est pourquoi j’obtiens « carte branche » pour user d’une technologie de mon choix et propose d’user de **Bootstrap**, cadriciel CSS populaire, facile à maintenir et bien connu sur le marché des développeurs web.  Sa riche documentation, son caractère gratuit, open source, et son usage répandu facilitant sa maintenance convainc Monsieur de Sousa d’adopter Bootstrap.  De là, je travaille en autonomie pour intégrer ladite fonctionnalité puis rend compte de mes travaux via l’outil de visio-conférence **Microsoft Teams**. | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***UBC Création*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Service développement web** | | | | | |
| **Période d’exercice** | |  | **Du** | | **04/03/2024** | | **au** | **19/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| J’installe la dépendance logicielle grâce au gestionnaire de dépendances PHP via la commande suivante le sous-dossier correspondant au projet sous Windows Subsystem Linux v2 (*i.e.* WSL2) :  `composer require twbs/bootstrap`  J’utilise le paquet assetmapper pour rappatrier la bibliothèque depuis NodeJs :  `php bin/console importmap :require bootstrap`  J’écris du code CSS dans le fichier custom.scss puis recompile avec l’outil asset-map pour produire une feuille de style en css depuis le pré-processeur CSS « scss » grâce à la commande :  `php bin/console asset-map :compile` | | | | | | | | |

**7**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **1** | | | **Développer la partie dynamique des interfaces utilisateur web ou web mobile** | | | | |
| ***Exemple n°2***  | | | | ***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 :** | | | | | | | | |
| Je propose à Madame Emilie de Sousa, responsable administrative, d’implémenter en parallèle du projet de site web un environnement type « bac à sable » pour expérimenter de nouvelles fonctionnalités, évaluer leur pertinence et leur périmètre et leur intégration sans transition avec le projet principal.  Cette dernière souhaite tester la possibilité d’introduire une liste de tâches simple, accessible sur une page isolée, avec pour ultime objectif de l’introduire dans le panneau administratif pour supporter la gestion des tâches administratives et commerciales.  Etant moi-même en auto-formation sur les languages populaires orientés web JavaScript et TypeScript associés au cadriciel Angular, très utilisé en entreprises pour les métiers front-end et doté d’une littérature riche, je soumets la proposition et Madame de Sousa accepte son introduction.  Une nouvelle base de code, indépendante du site web, est alors écrite sur un serveur de données séparé, m’évitant de reconfigurer une image Docker et de déployer un conteneur.  J’installe le cadriciel Angular sur l’environnement suivant :   * VoidLinux * environnement d’exécution JavaScript orienté applications web nodeJs v18.19.1 * navigateur web Google Chrome 126 * gestionnaire de paquets JavaScript pnpm 9.6   *via* l’exécution des commandes suivantes :  `sudo xbps-install –Suy nodejs pnpm`  `pnpm i --save-dev @angular@cli@18.1.2`  Je génère le projet Angular et installe le cadriciel de tests unitaires en JavaScript **Cypress**, puis le **linteur TypeScript**, un outil de recommandation de formatage et de bonnes pratiques liées à la syntaxe.  J’installe alors les dépendances manquantes via : `pnpm install`.  Je compile le projet et sert la page d’accueil en démarrant l’éxécution du serveur web : `ng serve -o`  J’installe alors le cadriciel Bootstrap pour définir facilement des styles CSS pour mes fonctionnalités nouvelles ainsi qu’une police de caractères populaire :  `pnpm i --save-exact [bootstrap@5.3.3](mailto:bootstrap@5.3.3) @fortawesome/fontawesome-free@6.5.2`  J’importe les feuilles de style CSS puis définis un template (*i.e.*un squelette de modèle) pour le component Angular :  <main class="container" style="margin-top: 70px;">  <h1>Ponyracer</h1></main>  Puis j’intrègre mes premiers tests unitaires grâce à l’environnement de tests Karma en saississant la commande suivante dans le terminal :  `ng generate config karma`  Enfin, j’introduis des tests de bout-en-bout (*viz.* end-to-end tests) en configurant le cadriciel de tests JavaScript Cypress via l’écriture d’un fichier `cypress.config.ts`.  Pour conclure avec les bonnes pratiques, je lance le **Linter TS** avec l’option de correction pour m’assurer de produire un code conforme aux conventions d’écriture : `ng lint --fix` | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| * **Angular CLI** *(viz.* Command Line Interface) est utilisé pour installer l’environnement de développement. * L’environnement d’exécution (*i.e.* runtime) JavaScript pour applications web **NodeJs** * Gestionnaire de paquets JavaScript `pnpm` (successeur de `npm`) * Editeur Visual Studio Code * **Cypress** * **Karma** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| Je propose d’implémenter la fonctionnalité avec ma tutrice de stage Emilie de Sousa, responsable administrative et commerciale.  Je travaille alors en autonomie pour intégrer ladite fonctionnalité puis rend compte de mes travaux via l’outil de visio-conférence **Microsoft Teams**. | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***UBC Création*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Service Développement web** | | | | | |
| **Période d’exercice** | |  | **Du** | | **04/03/2024** | | **au** | **19/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| La configuration de l’environnement de développement via **Angular** a nécessité la ré-écriture des fichiers suivants :   * configuration du compilateur TypeScript tsconfig.json * configuration de construction de l’espace de travailangular.json * définition des dépendances logiciellespackage.json | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **2** | | | **Développer la partie back-end d’une application web ou web mobile sécurisée** | | | | |
| ***Exemple n°1***  | | | | ***Développer des composants d’accès aux données SQL*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre de la formation, lors de l’apprentissage de la programmation orientée object en usant du patron de conception **MVC** (*viz*. Modèle Vue Contrôleur), il est question de développer un **composant d’interface** pour « visualiser » (*i.e.* mapper) la **couche métier PHP** avec une base de données relationnelles SQL **MySQL**.  A partir d’une user story définie au préable, je commence par créer le domaine métier, soit le modèle de données de la base en créant le fichier src/Entity.Database.php, en usant le mode scrict de PHP:  <?php declare(strict\_types=1) ;)  Je définis la classe en usant de bonnes pratiques liées à PHP 8.1 comme la promotion des propriétés du constructeur pour réduire la verbosité du code.  J’instancie un objet de la classe système PDO pour insérer les identifiants (nom de l’hôte, nom de la base, nom d’utilisateur, mot de passe).  Je fais appel au patron de conception dit « **Factory** » en déclarant uniquement des classes statiques afin d’éviter l’instanciation d’une classe pour appeler ses méthodes. Factory présente 2 avantages :   * Impose un point d’entrée unique pour toute l’application : facilite la compréhension du code. * Evite les instanciations multiple : réduit l’usage mémoire et la complexité   public static function getInstance(): Database {  if (self::$instance == null) self::$instance = new Database();  return self::$instance;  }  Puis j’implémente le *CRUD* (*viz.* Create, Read, Update, Delete), soit les 4 opérations nécessaires à l’exploitation d’un composant logiciel.  J’écris la classe métier pour définir un questionnaire (*i.e.* Quizz.php) et ses getters/setters pour retourner et définir la valeur de ses propriétés, ses 4 méthodes susmentionnées ainsi que le filtrage par sous-chaîne dans le titre.  Pour l’aspect sécurité, j’ai introduit 3 mécanismes pour lutter contre 2 types de faille.   * Requête préparée dans le CRUD en PHP pour contrer l’**injection de code arbitraire SQL :**   $stmt = Database::getInstance()->getConnexion()->prepare("select \* from Quizz where title like :textSearched;");   * La méthode htlmentities() dans le formulaire d’enregistrement HTML pour convertir les caractères en représentation chaînée afin de contrer l’**injection XSS** (*i.e.* Cross Site Scripting, surface d’attaque exploitée par la confiance accordée par la base de code applicative and le navigateur web de l’utilisateur. « Never trust user input ! ». Voici un example de neutralisation des paramètres saisis par l’utilisateur et passés au routeur :   $params[$param->name] =htmlentities($\_POST[$param->name]);   * La méthode filter\_var() pour s’assurer que le format de l’email correspond bien au format attendu, comme le ferait une expression rationnelle :   if (!$email = filter\_input(INPUT\_POST,$param->name, FILTER\_VALIDATE\_EMAIL))  Concernant la gestion du contrôle de code source, j’utilise l’outil git afin de normaliser la gestion du projet, permettre les rollback, surveillancer l’introduction de régressions dans l’histoire des « *commit* » et téléverser les itérations logicielles dans des branches de développement distinctes et ainsiéviter de casser l’existant ou d’écraser les travaux des autres développeurs. Voici un échantillon de commandes couramment exmployées durant le cycle de vie applicatif :   * git add remote origin [git@github.com:<user>/<repository>.git](mailto:git@github.com:%3cuser%3e/%3crepository%3e.git): démarre le pistage d’une branche distante pour téléverser depuis le dépôt local ; * git add .: met à jour l’index dans l’espace de travail pour préparer les fichiers modifiés à la prochaine soumission dans le dépôt local ; * git commit –mS “<description>”: soumet les changements dans le dépôt et déplace le curseur logique HEAD vers ce nouveau noeud dans l’arbre des soumissions. Chiffre le commit à l’aide d’une clé signée selon le protocole **GPG**. * git push –u origin <branch>: téléverse les changements présents dans le dépôt local vers le dépôt distant après calcul d’un delta.   Le site web *github.com* est également employé pour hébergement le dépôt distant et pouvoir rappatrier l’avancée des travaux vers mon ordinateur de bureau personnel ou mon ordinateur portable de formation.  Concernant l’organisation du sous-projet, l’outil en ligne trello.com est adopté pour gérer la liste de tâches.  Enfin, des tests unitaires ont été effectués à l’aide du paquet phpunit :   * symfony composer require phpunit --dev : installe l’outil de tests unitaires * php bin/phpunit : lance une batterie de tests unitaire depuis le sous-dossier tests/units   Testons la date de naissance d’un utilisateur :  public function testBirthDate()  {  $user = new User(date: new DateTimeImmutable('1985-02-16'));  $this->assertEquals(new DateTimeImmutable('1985-02-16'), $user ->getDate());  } | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| * Emulateur de terminal : **Konsole**/**Kitty** * Couche de compatibilité native pour binaires exécutables au format ELF : Partition logique WSL 2 **Debian** * Editeur **Visual Studio Code** * VCS (*viz.* Version Control System) : **git**/ github.com * Outil de tests unitaires en PHP **phpunit** * Gestion de projet collaboratif : **trello.com** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé en binôme avec un camarade de promotion grâce à l’outil **git** et au maintien d’un dépôt distant sur github.com.  J’ai rédigé le modèle de données et les test unitaires, grâce au **MCD** (*viz.* Modèle Conceptuel de Données) et la **user story** réalisés en commun. | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***UBC Création*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Service Développement web** | | | | | |
| **Période d’exercice** | |  | **Du** | | **04/03/2024** | | **au** | **19/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| L’outil **docker** est employé pour garantir la confirmité de l’environnement de travail entre les membres de l’équipe et éviter les disparités entre versions de dépendances et /ou de systèmes d’exploitation, grâce à l’écriture du manifeste docker-compose.yaml et les fichiers DockerFile associés aux services MySQL et PHP pour l’installation des paquets nécessaires à leur bon fonctionnement au sein des conteneurs. | | | | | | | | |
| **6** |  | | |  | | | | |
| **Activité-type** | **2** | | | **Développer la partie back-end d’une application web ou web mobile sécurisée** | | | | |
| ***Exemple n°2***  | | | | ***Développer des composants métier côté serveur*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le contexte de la formation, le référentiel technique couvre le développement d’un service web.  Je commence donc par créer un projet avec **Symfony** et installe les dépendances requises en usant de l’émulateur de terminal :  symfony new ws\_places--webapp  composer require annotations twig symfony/maker-bundle orm-pack serializer-pack  Je configure l’accès à la base de données MySQL grâce au fichier .env et édite la constante DATABASE\_URL.  Je crée alors la base de donnée avec l’outil doctrine : `php bin/console doctrine :database :create`  Je crée maintenant le modèle de données composé d’une entité Place grâce au paquet *maker-bundle* via la commande : `php bin/console make :entity Place`  Je valide ensuite la modification du schéma de données grâce aux commandes `make :migration` et `doctrine :migrations :migrate`  Je vérifie désormais que ma base de données existe en me connectant au panneau d’administration PhpMyAdmin sur le port approprié.  Je crée alors un contrôleur pour communiquer avec le modèle, les services et les constructeurs de formulaire via la commande : `php bin/console make :controller Place`  J’écris désormais les méthodes adéquates dans mon contrôleur et commence par lire tous les enregistrements, les convertit en format JSON et retourne une requête HTTP Response avec le code HTTP de status 200 pour notifier le serveur web que la ressource est générée avec succès. J’ajoute aussi une méthode pour trouver un enregistrement par identifiant pour autoriser l’utilisateur à effectuer une recherche sur le lieu.  Je pense à contrôler que mes noms de routes pointent vers les URL prévues à cet effet via :  `php bin/console debug :route`  Comment m’assurer que le service web puisse transformer un objet PHP en un fichier JSON et vice versa ? Je dois sérialiser mon entité, c’est à dire « mettre en série » des données pour les inclure dans un flux, afin d’être convertible dans un format spécifique.  Je modifie le fichier de configuration du service config/services.yaml puis actualise la méthode index() pour assurer la nouvelle normalisation des données après accès à la base.  Comment permettre aux utilisateurs de déclarer aimer (i.e. « like ») un lieu ?  Je mets à jour mon modèle de données en ajoutant l’entité Person et dois maintenant ajouter une relation entre cette nouvelle entité et Place de type N-N et valide le nouveau schéma de la base de données.  Je crée son contrôleur et ajoute bien sûr ses méthodes, pour lire tous les enregistrements et effectuer une recherche par identifiant.  Je me rend sur l’URL pointant vers l’entité personne de l’API : <https://localhost:8000/api/person/1> et découvre qu’une erreur de reférence circulaire survient lors de la sérialisation de l’entité Person.  Je parcours la documentation officielle de Symfony et apprend qu’une mécanique du nom d’attribut Groups résoud ce problème. Le mécanisme attribut est très récent (branche PHP 8.X) et remplace favorablement les annotations. J’ajoute les attributs `Groups` dans chacune des 2 entités du domaine métier.  Je mets à jour la méthode de chacun de mes contrôleurs :  normalized = $normalizer->normalize($person, null, [‘groups’ => ‘person:read’]);  Dernière étape du sous-projet, mise en place de la sécurisation par JSON Web Token(JWT) via le protocole openSSL et le paquet lexik/jwt-authentication-bundle  Je génère une clé privée et une clé publique AES 256bit  Je dois dès à présent créer une entité User pour permettre aux utilisateurs de s’identifier et par la présente, sécuriser l’accès au service web. Après lecture de la section appropriée de la documentation officielle de Symfony, j’édite le fichier security.yaml (notamment les sous-sections firewalls, encoders, providers et access\_control).  J’ajoute ensuite un contrôleur pour gérer l’authentification utilisateur et les requêtes de réponse HTTP qui communiquent entre le navigateur web et le serveur web.  Enfin, des tests fonctionnels sont réalisés avec l’outil Postman qui évite d’écrire un environnement d’acception en émulant les requêtes définies par des utilisateurs finaux :   * Test de sécurisation : i.e. GET http://localhost:8000/api/place/1 * Test d’enregistrement d’un utilisateur : GET http://localhost:8000/register * Connexion et récupération du jeton d’identification :   GET http://localhost:8000/api/login\_check   * Consommation du service web en supprimant délibéremment le jeton pour contrôler l’interdiction de servir la ressource en retournant au navigateur dans l’entête de réponse un code de status HTTP 403 (*i.e.* forbidden). | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| * Interface de Programmation Applicative (*viz.* API) d’intégration de tests : **Postman** * Cadriciel PHP : **Symfony** * Outil de visualisation (*viz.* mappage) entre couche métier et base de données : **Doctrine** * Editeur de texte : **Visual Studio Code** * Navigateur web : **Mozilla Firefox** * Moteur de squelettes de modèle : **twig** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| Ce sous-projet est entièrement réalisé par moi-même, qu’il s’agisse de son implémentation ou de la recherche documentaire. | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***UBC Création*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Service Développement web** | | | | | |
| **Période d’exercice** | |  | **Du** | | **04/03/2024** | | **au** | **19/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Le site web **discord.com** est visité pour être mis en relation avec des développeurs **PHP** du monde entier afin d’obtenir des réponses aux questions que j’ai pu posées pour garantir la qualité du code et la bonne compréhension et prise en main de la plateforme de tests **Postman**. | | | | | | | | |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| **Master Management et Stratégie d’Entreprises** | **Université d’Orléans – 45100 ORLEANS La Source** | **29/06/2013** |
| **License en Transactions Immobilières** | **Université d’Orléans – 45100 ORLEANS La Source** | **31/06/2011** |
| **BTS Négociation et Relation Client** | **Lycée Saint Paul Bourdon – 45000 ORLEANS** | **02/07/2006** |
| **Bac STT Informatique de Gestion** | **Lycée Benjamin Franklin – 45000 ORLEANS** | **09/07/2003** |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |

**8**

|  |
| --- |
| **Déclaration sur l’honneur** |
|  |
|  |

*Aurélien Plazzotta*

Je soussigné(e) [prénom et nom] ,

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

*Vierzon*

02/08/2024

Fait à le

pour valoir ce que de droit.

Signature :

**9**

|  |
| --- |
| **Documents illustrant la pratique professionnelle** |
|  |
| *(facultatif)* |
| **Intitulé** |
| Cliquez ici pour taper du texte. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**10**

|  |
| --- |
| **Annexes** |
|  |
| *(Si le RC le prévoit)* |
|  |

**11**