|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | HANOUN |
| *Nom d’usage* |  |  |
| *Prénom* |  | Adil |
| *Adresse* |  | 94 rue des patineurs 59280 Armentières |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| **Cliquez ici pour entrer l’intitulé du titre professionnel visé.** | |
|  | |
| **Modalité d’accès :** | |
|  | |
|  | Parcours de formation |
|  | Validation des Acquis de l’Expérience (VAE) |
|  | |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Sommaire** | | | |
| **Exemples de pratique professionnelle** | | | |
| **Concevoir et développer des composants d’interface utilisateurs en intégrant les recommandations de sécurité** | | **p.** | **6** |
|  |  Maquetter une application p. | p. | 6 |
|  |  Développer une interface utilisateur p. | p. | 8 |
|  |  Développer des composants d’accès aux donnés p | p. | 11 |
|  |  Développer la partie frontend d’une interface utilisateur p | p. | 13 |
|  |  Développer des composants d’accès aux donnés p | p. | 15 |
|  |  |  |  |
| **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité** | | **p.** | **17** |
|  |  Concevoir une base de données p. | p. | 17 |
|  |  Mettre en place une base de données   p. | p. | 19 |
|  |  Développer des composants dans le langage d’une base de données p | p. | 22 |
|  |  |  |  |
| **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | **p.** | **25** |
|  |  Collaborer à la gestion d’un projet informatique et à l’organisation de l’environnement de développement   p. | p. | 25 |
|  |  Concevoir une application   p. | p. | 27 |
|  |  Développer des composants métier   p | p. | 30 |
|  |  Construire une application organisée en couches p | p. | 32 |
|  |  Développer une application mobile   p | p. | 35 |
|  |  Préparer et exécuter les plans de test d’une application   p | p. | 37 |
|  |  Préparer et exécuter le déploiement d’une application  p | p. | 40 |
|  |  |  |  |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | p. | 42 |
| **Déclaration sur l’honneur** | | p. | 43 |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | p. | 44 |
| **Annexes** *(Si le RC le prévoit)* | | p. | 45 |

**Exemples de pratique**

**professionnelle**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | | | | |
| **Activité-type** | **1** | | | **​​Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité​** | | | | |
| ***Exemple n°1***  | | | | ***​​***Maquetter une application***​*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre de ma formation de Concepteur Développeur d'Application, nous avons reçu un TP à réaliser en groupe. Ce TP regroupait une bonne partie des compétences vues en formation. Nous devions développer une application de pointage d’employés comprenant une partie mobile pour les salariés et une partie web pour les administrateurs.  Nous avons conçu les wireframes de l'application, représentant les grandes lignes et la structure des interfaces utilisateur pour les deux parties :   * Partie mobile (salariés) : Page de login et page d'accueil de pointage. * Partie web (administrateurs) : Page de connexion, tableau de bord et page d'ajout de salarié.   Suite à la validation des wireframes, j'ai utilisé les mêmes outils de design pour créer les maquettes détaillées de l'application, incluant les aspects graphiques et visuels pour les deux parties .    Ce TP m'a permis de mettre en pratique plusieurs compétences essentielles, telles que l'analyse des besoins, la création de wireframes, la conception UX/UI, la collaboration en équipe. Le travail en groupe, sous la supervision de notre formateur, m'a offert un cadre structuré et des retours constructifs, favorisant ainsi mon apprentissage et ma progression en tant que concepteur développeur d'application. Voir annexe A B et C | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Utilisation de Figma pour la réalisation de la maquette et wireframe.  Collaboration sur Teams.  Teams est un logiciel permettant de communiquer avec une ou plusieurs personnes. Dessus nous pouvons faire des partages d’écran et nous envoyer le travail effectué. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé avec un collègue de ma formation concepteur développeur d’application | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **M2i formation** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe** | | | | | |
| **Période d’exercice** | |  | **Du** | | **24/08/2023** | | **au** | **26/08/2023** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Pour visualiser l’intégralité de la maquette : <https://www.figma.com/design/IN4PxbErzpPe2h06Wm78Fo/TP_Final?node-id=0-1&t=vOFhwOxujGN2DLC4-0> | | | | | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **1** | | | **​​Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité​** | | | | |
| ***Exemple n°1***  | | | | ***​​***Développer une interface utilisateur de type desktop  ***​*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| j'ai réalisé un exercice visant à développer une interface de type desktop pour la gestion des employés (CRUD) et des départements. Voici comment je l'ai abordé :  Pour ce projet, j'ai utilisé la technologie Java Swing, qui fait partie de Java SE. J'ai conçu une application suivant l'architecture MVC (Modèle-Vue-Contrôleur) et intégrant une couche DAO (Data Access Object). La connexion à la base de données a été assurée via JDBC.    j'ai configuré une JFrame, qui est une fenêtre principale de l'application Swing, pour afficher l'interface.  Ensuite, j'ai créé un conteneur principal à l'aide de JPanel, qui contient lui-même deux conteneurs : un pour les employés et un autre pour les départements.  Le JButton est utilisé pour exécuter des actions spécifiques comme ajouter, éditer, et supprimer des employés.  *Extrait de code de la class EmployeeView*  Le JLabel est utilisé pour afficher des textes statiques dans l'interface  Le JTextField permet à l'utilisateur de saisir des informations.  *Extrait de code de la classe AddEmployee*  J'ai utilisé différents gestionnaires de disposition (layout managers) pour organiser les composants Swing de manière structurée et esthétique.  Le BorderLayout divise le conteneur en cinq régions : nord (NORTH), sud (SOUTH), est (EAST), ouest (WEST) et centre (CENTER). Chaque région peut contenir un seul composant.  *Extrait de code de la classe EmployeeView*  Ce projet m'a permis de renforcer mes compétences en développement d'interfaces desktop, en particulier avec Java Swing, et d'approfondir ma compréhension de l'architecture MVC et des interactions avec une base de données via JDBC. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **IntelliJ IDEA**  **Java swing**  **Jdbc, MySQL** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J'ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **M2i formation** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réaliser en classe** | | | | | |
| **Période d’exercice** | |  | **Du** | | **08/04/2024** | | **au** | **12/04/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **1** | | | **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°3***  | | | | ***​​****Développer des composants d’accès aux données****​*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre du développement de mon projet de blog sur les animaux utilisant React pour le frontend et Spring Boot pour le backend, j'ai été chargé de développer les composants d'accès aux données. Voici les tâches effectuées :  J'ai analysé les besoins en données pour identifier les différentes informations nécessaires pour le blog, incluant les articles, les commentaires, les utilisateurs, et les catégories d'animaux. J'ai conçu le modèle de données en définissant les entités, les relations et les contraintes nécessaires pour stocker ces informations. J'ai implémenté les composants d'accès aux données en utilisant Spring Boot et le framework JPA (Java Persistence API) avec Hibernate pour l'ORM.  ORM signifie "Mappage Objet-Relationnel". C'est une technique qui permet aux développeurs d'interagir avec les bases de données relationnelles en utilisant un modèle de programmation orienté objet. Autrement dit on peut interagir avec les tables de bases de données relationnelles, ce qui permet aux développeurs de voir la base de données en utilisant des concepts orientés objet, plutôt que d'écrire des instructions SQL brutes.  Pour ce faire, j'ai créé les classes d'entités correspondant aux tables de la base de données, comme **Article**, **Comment**, **User**, et **Category**. Ensuite, j'ai implémenté les interfaces de repository en étendant **JpaRepository** pour encapsuler les opérations CRUD (Create, Read, Update, Delete). J'ai configuré les connexions à la base de données MySQL en utilisant les propriétés de Spring Boot pour sécuriser les paramètres de connexion. (Voir Annexe D)  J'ai développé des contrôleurs REST pour exposer les endpoints nécessaires aux opérations sur les articles, les commentaires, les utilisateurs et les catégories. Pour cela, j'ai utilisé Spring MVC avec des annotations telles que **@RestController**, **@GetMapping**, **@PostMapping**, etc. J'ai intégré le frontend React avec les API REST développées en Spring Boot en utilisant Axios pour effectuer des requêtes HTTP et gérer les réponses. (Voir Annexe E) | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Environnement de développement : IntelliJ IDEA pour le backend (Spring Boot) et Visual Studio  Base de données : MySQL.  Frameworks et bibliothèques : Spring Boot, JPA (Hibernate) | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J'ai réalisé ce projet seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **M2i formation** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **En classe et chez moi** | | | | | |
| **Période d’exercice** | |  | **Du** | | **19/02/2024** | | **au** | **26/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : https://github.com/Adil-ha/dossier\_pro/tree/main/blog | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **1** | | | **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°4***  | | | | *Développer la partie front-end d’une interface utilisateur web* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| J'ai été chargé de développer la partie front-end de l'interface utilisateur web pour le blog. Voici les principales tâches que j'ai accomplies :  Création du projet React avec Vite : J'ai initié mon projet React en utilisant Vite, un outil de développement rapide pour les applications web modernes. Vite permet une configuration minimaliste et des temps de compilation rapides, ce qui a accéléré le processus de développement.    Gestion des dépendances avec npm: Aprés avoir créé le projet, j’ai utilisé npm pour gérer les dépendances du projet. J’ai installé les bibliothèques et les outils nécessaires tels que Bootstrap, Redux, Axios, React Router Dom en utilisant des commandes npm telles que ‘npm install @reduxjs/toolkit’    Utilisation de Bootstrap : J'ai intégré Bootstrap dans mon projet pour faciliter le développement de l'interface utilisateur en utilisant des composants préconçus et des styles prêts à l'emploi. Cela m'a permis de gagner du temps dans la conception et le stylisme des différentes pages de l'application.    Utilisation de Redux : J'ai choisi Redux comme gestionnaire d'état pour mon application React afin de centraliser et de gérer efficacement l'état global de l'application. Cela a été particulièrement utile pour gérer les données telles que les articles du blog, les commentaires, les utilisateurs connectés, etc. (Annexe F)  Utilisation d'Axios : J'ai utilisé Axios comme client HTTP pour effectuer des requêtes REST vers le backend afin de récupérer et d'envoyer des données. Axios offre une syntaxe simple et claire pour effectuer des requêtes HTTP, ce qui a simplifié la communication avec le backend. (Annexe G)    Utilisation de React Router Dom : J'ai utilisé React Router Dom pour gérer la navigation dans mon application React. Cela m'a permis de définir des routes pour chaque page de l'application et de naviguer entre elles de manière fluide en fonction des actions de l'utilisateur. (Voir annexe H)    Décomposition de l'application en différentes couches : J'ai organisé mon application React en utilisant une architecture basée sur les composants, les pages, les services et les fonctionnalités. Cette approche m'a permis de maintenir un code bien structuré, modulaire et facilement extensible.    Utilisation de slices Redux pour la gestion des états : J'ai découpé mon état global Redux en "slices" pour gérer différentes parties de l'application de manière indépendante. Chaque slice Redux est responsable d'une fonctionnalité spécifique de l'application, ce qui facilite la gestion de l'état et la mise à jour des données. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Framework Front-end : React avec Vite.  Gestionnaire de package : npm (Node Package Manager).  Bibliothèques de styles : Bootstrap.  Gestionnaire d'état : Redux.  Client HTTP : Axios.  Gestionnaire de routage : React Router Dom. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seule | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **M2i Formation** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **En classe et chez moi** | | | | | |
| **Période d’exercice** | |  | **Du** | | **23/10/2023** | | **au** | **26/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : https://github.com/Adil-ha/dossier\_pro/tree/main/blog | | | | | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **1** | | | **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°5***  | | | | *Développer la partie back-end d’une interface utilisateur web* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour le développement de la partie back-end de l'interface utilisateur web pour le blog, j'ai réalisé plusieurs tâches essentielles :  Création du projet : J'ai initié le projet en utilisant IntelliJ IDEA, un environnement de développement intégré (IDE) robuste et performant. J'ai configuré le projet en utilisant Maven comme gestionnaire de dépendances, ce qui m'a permis de gérer facilement les bibliothèques  Développement des entités et des repositorie**s** : J'ai créé les classes d'entités correspondant aux tables de la base de données, telles que **Article**, **Comment**, **User**, et **Category**. J'ai utilisé les annotations JPA pour définir les relations et les contraintes. Ensuite, j'ai développé les interfaces de repository en étendant **JpaRepository** pour faciliter les opérations CRUD sur ces entités.    Implémentation de la sécurité : J'ai configuré Spring Security pour protéger les endpoints de l'application. J'ai mis en place l'authentification et l'autorisation en utilisant JWT pour sécuriser les communications entre le front-end et le back-end. Cela comprenait la création de filtres de sécurité, de services d'authentification et de contrôleurs pour la gestion des utilisateurs et des sessions.    Développement des contrôleurs REST : J'ai créé des contrôleurs REST pour exposer les endpoints nécessaires à la gestion des articles, des commentaires, des utilisateurs et des catégories. J'ai utilisé des annotations Spring telles que **@RestController**, **@GetMapping**, **@PostMapping**, etc., pour définir les routes et les actions associées. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Environnement de développement : IntelliJ IDEA.  Gestionnaire de dépendances : Maven.  Frameworks et bibliothèques :   * Spring Boot (Web, Data JPA, Security) * JSON Web Tokens (JWT) * MySQL Connector-J * Lombok   Base de données : MySQL. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **M2i formation** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | ​​Travaux pratique réalisé en classe ​et chez moi | | | | | |
| **Période d’exercice** | |  | **Du** | | **11/12/2023** | | **au** | **19/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : https://github.com/Adil-ha/dossier\_pro/tree/main/blog | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **2** | | | **Concevoir et développer la persistance des données**  **en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°1***  | | | | *Concevoir une base de données* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour concevoir la base de données du blog, j'ai suivi une méthodologie rigoureuse en utilisant l'approche Merise, un modèle de conception de bases de données bien structuré. Voici les étapes que j'ai réalisées :  Analyse des besoins : J'ai commencé par une analyse détaillée des besoins fonctionnels et non fonctionnels du blog. Cela comprenait la définition des informations à gérer (articles, commentaires, utilisateurs, catégories, JWT) et des interactions entre ces informations.  Conception du Modèle Conceptuel de Données (MCD) : En utilisant l'outil Looping, j'ai créé le MCD pour représenter les entités et leurs relations de manière abstraite. Le MCD se concentre sur les concepts et les relations sans se préoccuper des détails techniques.  Entités et relations :  Article : Comment : User : Category :  JWT :  Relations : Un Article peut appartenir à plusieurCategory  Un Article peut avoir plusieurs Comment  Un Comment est écrit par un User  Un Comment est associé à un Article  Un User peut avoir plusieurs JWT  Conception du Modèle Logique de Données (MLD) : Après validation du MCD, j'ai transformé ce modèle en MLD. Le MLD détaille les structures de la base de données en termes de tables, de colonnes, de types de données, et de clés primaires et étrangères. (voir en annexe I J) | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Méthodologie de conception : Merise. Outil de modélisation : Looping (pour créer les diagrammes MCD et MLD). Environnement de développement de base de données : MySQL Workbench. Base de données : MySQL. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **J’ai travaillé seul** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe et chez moi** | | | | | |
| **Période d’exercice** | |  | **Du** | | **13/11/2023** | | **au** | **26/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **2** | | | **Concevoir et développer la persistance des données**  **en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°2***  | | | | *Mettre en place une base de données* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour mettre en place la base de données du blog, j'ai suivi une approche structurée en utilisant SQL pour créer les différentes tables, définir leurs relations, et imposer des contraintes afin de garantir l'intégrité et la cohérence des données. Voici les étapes que j'ai réalisées :  Création de la base de données : J'ai commencé par créer la base de données **BLOG** si elle n'existait pas déjà en utilisant la commande **CREATE DATABASE IF NOT EXISTS BLOG;**.  Utilisation de la base de données : J'ai ensuite sélectionné cette base de données pour y créer les tables en utilisant **USE BLOG;**.    Définition des tables et des colonnes : J'ai créé les tables **T\_User**, **T\_Article**, **T\_Category**, **T\_Comment**, **T\_Jwt** et **T\_Association\_Category\_Article** avec les colonnes et types de données appropriés. Chaque table a été définie avec ses clés primaires et étrangères pour assurer l'intégrité référentielle entre les tables.  Clés primaires : Chaque table a une clé primaire définie pour identifier de manière unique chaque enregistrement. Par exemple, **U\_id** est la clé primaire de la table **T\_User**, et **A\_id** est la clé primaire de la table **T\_Article**.  Clés étrangères : Les clés étrangères sont utilisées pour établir des relations entre les tables. Par exemple, la colonne **U\_id** dans la table **T\_Article** est une clé étrangère qui référence la colonne **U\_id** de la table **T\_User**. Cela garantit que chaque article est associé à un utilisateur existant.  Contraintes de non-nullité : Plusieurs colonnes sont définies comme **NOT NULL** pour s'assurer que des valeurs nulles ne peuvent pas être insérées dans ces colonnes. Par exemple, **U\_name**, **U\_email**, **U\_password**, et **U\_role** dans la table **T\_User** sont définis comme **NOT NULL**.  Types de données appropriés : J'ai utilisé des types de données appropriés pour chaque colonne pour garantir la validité des données. Par exemple, **A\_created\_at** est de type **DATE**, et **U\_email** est de type **VARCHAR**.  Définition des relations et des contraintes : J'ai configuré les clés étrangères pour assurer l'intégrité référentielle entre les tables. Par exemple, la colonne **category\_id** dans la table **T\_Article** est une clé étrangère faisant référence à la table **T\_Category**.  Ajout de données de test : J'ai inséré des données de test dans les tables **T\_User**, **T\_Category**, **T\_Article** et **T\_Comment** pour valider la structure de la base de données. Cela incluait des utilisateurs avec des rôles différents, des catégories liées aux animaux, des articles sur les animaux et des commentaires sur ces articles. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Environnement de développement de base de données : MySQL Workbench.  Base de données : MySQL. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2I Formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | ​​Travaux pratique réalisé en classe ​ | | | | | |
| **Période d’exercice** | |  | **Du** | | **20/11/2023** | | **au** | **26/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog>  Pour les bonnes pratiques de nommage je me suis appuyé sur sql.sh | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **2** | | | **Concevoir et développer la persistance des données**  **en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°8***  | | | | *Développer des composants dans le langage d’une base de données* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour le développement des composants dans le langage d'une base de données, j'ai utilisé Spring Data JPA pour simplifier les interactions avec la base de données. L'utilisation de cet ORM (Object-Relational Mapping) permet de mapper les entités Java aux tables de la base de données et de gérer automatiquement les opérations CRUD (Create, Read, Update, Delete).  Dans le cadre de ce projet de blog, j'ai effectué les tâches suivantes :  Configuration de l'application Spring Boot : J'ai configuré les paramètres de connexion à la base de données dans le fichier application.propertiesafin de définir l'URL de la base de données, les identifiants de connexion, et d'autres paramètres spécifiques à JPA.     * spring.datasource.url : URL de connexion à la base de données MySQL. Le paramètre createDatabaseIfNotExist=true permet de créer la base de données si elle n'existe pas.      * spring.datasource.username et spring.datasource.password : Identifiants de connexion à la base de données. * server.port : Port sur lequel l'application Spring Boot sera exécutée. * spring.jpa.hibernate.ddl-auto : Définit la stratégie de génération du schéma de la base de données. update permet de mettre à jour le schéma de la base de données à chaque redémarrage de l'application. * spring.application.name : Nom de l'application.   Création des entités JPA : J'ai défini des entités Java représentant les tables de la base de données, en utilisant des annotations JPA pour le mappage. Par exemple, l'entité Comment représente la table T\_Comments.  Le mappage des entités avec Spring Data JPA permet de lier les classes Java aux tables de la base de données.     * @Entity : Indique que cette classe est une entité JPA. * @Table(name = "T\_Comments") : Spécifie le nom de la table correspondante dans la base de données. * @Id : Indique le champ id comme clé primaire. * @GeneratedValue(strategy = GenerationType.IDENTITY) : Spécifie que la clé primaire est générée automatiquement par la base de données. * @JoinColumn(name = "Co\_id") : Spécifie le nom de la colonne dans la table de base de données qui est liée à ce champ.   L'annotation @ManyToOne est utilisée pour définir une relation de type plusieurs-à-un entre deux entités. Dans cet exemple, un commentaire est associé à un utilisateur et à un article, ce qui signifie que plusieurs commentaires peuvent être liés à un seul utilisateur et un seul article. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Environnement de développement : IntelliJ IDEA pour le backend (Spring Boot).  Frameworks : Spring Boot, Spring Data JPA.  Fichier de configuration : application.properties. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe** | | | | | |
| **Période d’exercice** | |  | **Du** | | **27/11/2023** | | **au** | **01/12/2023** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog> | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°9***  | | | | *Collaborer à la gestion d’un projet informatique et à l’organisation de l’environnement de développement* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre de ma formation, j'ai participé à un projet de groupe pour développer une application de pointage employé . Nous avons utilisé la méthodologie Scrum pour organiser notre travail en sprints et effectuer des daily stand-ups. Les tâches et opérations spécifiques que j'ai effectuées incluent :   * **Planification et gestion de projet :** Participation aux réunions de sprint planning et aux daily stand-ups pour discuter des progrès et des obstacles. Utilisation de Trello pour suivre les tâches et organiser le travail en fonction des priorités et des échéances. * **Développement de fonctionnalités :** J'ai travaillé sur la branche authentification et gestion des employés, en développant les fonctionnalités liées à l'inscription, la connexion, la gestion des rôles, et la gestion des employés (ajout, modification, suppression). * **Intégration du code :** Utilisation de Git pour le contrôle de version, avec une branche dev pour l'intégration continue et des branches de fonctionnalités pour chaque tâche spécifique. Une fois les fonctionnalités développées et testées, elles étaient fusionnées dans la branche dev. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| * **Méthodologie de gestion de projet :** Scrum, avec des sprints de 2 semaines. * **Outils de versionnement :** Git pour le contrôle de version, avec une branche dev pour l'intégration continue et des branches de fonctionnalités pour chaque tâche spécifique. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J'ai travaillé en collaboration avec les autres membres de mon groupe de formation. Chaque membre de l'équipe avaient des responsabilités spécifiques :   * **Moi :** Responsable de l'authentification et de la gestion des employés. * **Collègue  :** Responsable du développement des fonctionnalités de pointage. | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i Formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | ​​Travaux pratique réalisé en classe ​ | | | | | |
| **Période d’exercice** | |  | **Du** | | Cliquez ici | | **au** | Cliquez ici |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°10***  | | | | *Concevoir une application* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour la conception de mon blog, j'ai utilisé les diagrammes UML (Unified Modeling Language) pour modéliser les différentes parties de l'application. Les diagrammes UML sont essentiels pour planifier, visualiser et documenter les aspects structurels et comportementaux d'une application.    Dans le cadre de la conception du blog, j'ai utilisé plusieurs types de diagrammes UML pour représenter les différents aspects de l'application. Voici les étapes principales de cette tâche :  **Diagramme de cas d’utilisation :** J'ai créé un diagramme de cas d'utilisation pour représenter les interactions entre les acteurs (utilisateurs) et le système. Dans ce cas, j'ai identifié deux acteurs principaux : l'utilisateur standard et l'administrateur. L'utilisateur peut visiter le blog et commenter un article après s'être authentifié. L'administrateur a les mêmes pouvoirs que l'utilisateur standard, mais il doit également s'authentifier pour gérer les articles et les utilisateurs. (Voir en annexe k).  La flèche de d’héritage indique que l'utilisateur standard et l'administrateur ont des fonctionnalités communes telles que visiter le blog et commenter un article, mais l'administrateur peut également avoir des fonctionnalités supplémentaires comme gérer les articles et les utilisateurs.  La flèche d’inclusion indique ici que l'authentification est une étape nécessaire pour commenter un article. Ainsi, le cas d'utilisation "Commenter un article" inclut le cas d'utilisation "S'authentifier". Cela signifie que pour commenter un article, l'utilisateur doit d'abord s'authentifier.  La flèche d’extension indique que l'administrateur peut avoir la possibilité d'étendre ses fonctionnalités pour inclure la gestion des utilisateurs et des articles. Ces fonctionnalités sont des extensions optionnelles au cas d'utilisation de base de l'administrateur.  **Diagramme de classes :** Ce diagramme m'a permis de modéliser la structure statique du système, en définissant les classes, leurs attributs, méthodes et relations. Les principales classes incluent User, Article, Comment, Category, et Jwt. (Voir en annexe L)  Dans un diagramme de classe, la relation de composition et la relation d'agrégation sont des types de relations qui décrivent la façon dont les objets sont liés les uns aux autres dans le système. Voici comment elles s'appliquent dans le contexte des entités "Comment" et "Article" ainsi que "Article" et "Category" dans notre exemple :  Relation de Composition entre Comment et Article :   * La relation de composition est une relation forte où l'objet enfant ne peut exister indépendamment de l'objet parent. Cela signifie que si l'objet parent est détruit, tous les objets enfants associés seront également détruits. * Dans notre exemple, la relation de composition entre "Comment" et "Article" indique que chaque commentaire est spécifiquement lié à un article particulier. Si un article est supprimé, tous les commentaires associés à cet article seront également supprimés car ils n'ont pas de sens sans l'article auquel ils sont attachés.   Relation d'Agrégation entre Article et Category :   * La relation d'agrégation est une relation plus faible où l'objet enfant peut exister indépendamment de l'objet parent. Cela signifie que l'objet enfant peut être associé à plusieurs objets parents différents sans qu'il y ait de dépendance stricte. * Dans notre exemple, la relation d'agrégation entre "Article" et "Category" indique que chaque article peut être associé à une catégorie spécifique, mais cette catégorie peut également être partagée par d'autres articles. Par exemple, plusieurs articles peuvent appartenir à la même catégorie "Chats". La catégorie existe indépendamment des articles qui y sont associés.     Diagramme de séquence **:** J'ai élaboré un diagramme de séquence pour représenter le processus d'inscription et de connexion d'un utilisateur. Ce diagramme met en évidence les interactions entre l'utilisateur, le système et la base de données lors de ces opérations. (Voir en annexe M) | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| StarUML pour la création de diagrammes UML. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J'ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe et chez moi** | | | | | |
| **Période d’exercice** | |  | **Du** | | **06/11/2023** | | **au** | **10/11/2023** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°11***  | | | | *Développer des composants métier* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre du développement de mon blog sur les animaux, j'ai mis en place plusieurs composants métiers pour gérer les utilisateurs, les articles, les catégories et les commentaires.  A partir de mon diagramme de classe j’ai implémenté mes entités    Pour rappel les composants métier représentent les fonctionnalités et processus essentiels au domaine de l’application et traitent les règles et les opérations métiers. La couches services joue un rôle central dans la gestion de la logique métier  Voici un exemple de composant métier pour la gestion des commentaires, détaillant les tâches CRUD (Créer, Lire, Mettre à jour, Supprimer) et les conditions d'utilisation : la class CommentService (voir en annexe N) | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Spring boot  IntelliJ IDEA | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i Formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe et chez moi** | | | | | |
| **Période d’exercice** | |  | **Du** | | **22/01/2024** | | **au** | **26/07/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog> | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°12***  | | | | *Construire une application organisée en couches* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour construire l'application de blog sur les animaux, j'ai adopté une architecture en couches, appelée également architecture en n-tiers. Cette architecture permet de séparer les différentes responsabilités au sein de l'application, rendant le code plus modulaire, maintenable et testable.  *source :* [*https://enset-media.ac.ma/*](https://enset-media.ac.ma/)   * Couche Entité (Entity) :   Représenter les objets métiers du domaine, comme User, Article, Comment, et Category.  Les entités sont mappées aux tables de la base de données via des annotations JPA (Java Persistence API).     * Couche Repository :   Fournir des méthodes pour effectuer des opérations CRUD (Créer, Lire, Mettre à jour,Supprimer) sur les entités.  Les repositories étendent JpaRepository de Spring Data JPA, facilitant l'accès et la gestion des données.     * Couche Service :   Contenir la logique métier et orchestrer les interactions entre les répertoires et les autres services.  Les services utilisent l'injection de dépendances pour accéder aux répertoires et autres services.  Exemple : CommentService qui gère les opérations métier liées aux commentaires.     * Couche Contrôleur (Controller) :   Gérer les requêtes HTTP entrantes, appeler les services appropriés, et retourner les réponses adéquates.  Les contrôleurs exposent des API RESTful et utilisent les services pour traiter les requêtes.  Exemple : CommentController pour les opérations sur les commentaires.     * Couche DTO (Data Transfer Object) :   Transférer les données entre les couches tout en encapsulant et simplifiant les données des entités.  Les DTOs sont utilisés pour éviter l'exposition directe des entités aux couches externes.  Exemple : CommentDTO qui contient les données d'un commentaire pour les échanges entre les couches.     * Couche Mapper :   Convertir les entités en DTOs et vice versa.  Les mappers facilitent la transformation des données entre les entités et les DTOs.  pour mapper Comment à CommentDTO.   * Gestion des Exceptions :   Gérer les erreurs et exceptions dans l'application de manière centralisée.  Les exceptions personnalisées permettent de gérer les erreurs spécifiques au domaine.  Exemple : InvalidEmailException pour gérer les erreurs d'email invalide.     * Sécurité :   Gérer l'authentification et l'autorisation des utilisateurs.  Utilisation de Spring Security et JWT pour sécuriser les endpoints et gérer les sessions.  Exemple : Configuration Spring Security pour restreindre l'accès aux API. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Spring boot  IntelliJ IDEA | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **Travaux pratique réalisé en classe** | | | | | |
| **Période d’exercice** | |  | **Du** | | **05/02/2024** | | **au** | **09/02/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog> | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°13***  | | | | *Développer une application mobile* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  | | | | | | | | |
| **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)* | | | | | | | | |
| Cliquez ici pour taper du texte. | | | | | | | | |
|  |  | | |  | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°14***  | | | | *Préparer et exécuter les plans de tests d’une application* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Pour les tests de l'application du blog, j'ai réalisé des tests unitaires afin de vérifier le bon fonctionnement des différentes fonctionnalités. Voici un aperçu des tâches effectuées :   * J'ai écrit des tests unitaires pour la classe UserService, qui est responsable de la gestion des utilisateurs dans l'application.     Exemple avec un email invalide    Ce test unitaire vise à vérifier le comportement de la méthode register de la classe UserService lorsque l'email fourni est invalide. Voici une explication détaillée :   * Tout d'abord, je définis un nouvel utilisateur avec une adresse e-mail invalide en utilisant la méthode setEmail. * Ensuite, j'utilise la méthode assertThrows pour m'assurer qu'une exception de type InvalidEmailException est levée lorsque la méthode register est appelée avec cet utilisateur invalide. * Dans le bloc lambda passé à assertThrows, j'appelle la méthode register de userService avec l'utilisateur invalide. * Ensuite, je vérifie que l'exception levée contient le message d'erreur attendu à l'aide de la méthode assertEquals. * Enfin, j'utilise verify avec never() pour m'assurer que la méthode save de userRepository n'est jamais appelée, car l'utilisateur invalide ne doit pas être enregistré dans la base de données.   Chaque test unitaire vise à tester un aspect spécifique du comportement attendu de UserService, tel que l'inscription d'un nouvel utilisateur, la recherche d'un utilisateur par email, etc.    Les conditions dans lesquelles les tests ont été réalisés incluent l'utilisation de l'extension Mockito pour simuler les dépendances et de la bibliothèque JUnit Jupiter pour l'écriture des tests.    Dans ces tests, j'ai également utilisé des mocks pour simuler le comportement des dépendances telles que le UserRepository et le BCryptPasswordEncoder. Les mocks sont utilisés pour isoler le code testé des dépendances externes et garantir que les tests se concentrent uniquement sur le comportement de la classe testée. | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Pour les tests unitaires, j'ai utilisé les outils suivants :   * JUnit Jupiter : Framework de test unitaire pour la plateforme Java. * Mockito : Bibliothèque de test pour créer des mocks et simuler le comportement des dépendances. * ObjectMapper : Pour la sérialisation et la désérialisation des objets lors des tests impliquant des requêtes et des réponses JSON. * SpringExtension : Pour faciliter l'intégration des tests avec le framework Spring. * AutoConfigureMockMvc : Pour configurer automatiquement le MockMvc dans les tests de contrôleur. | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | M2i formation | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | ​​Travaux pratique réalisé en classe ​et chez moi | | | | | |
| **Période d’exercice** | |  | **Du** | | **11/03/2024** | | **au** | **15/03/2024** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog> | | | | | | | | |
|  |  | | |  | | | | |
| **Activité-type** | **3** | | | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°15***  | | | | *Préparer et exécuter le déploiement d’une application* | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| Dans le cadre du déploiement de l'application de blog, j'ai utilisé Docker pour containeriser les parties front-end et back-end de l'application, ainsi que pour déployer une base de données MySQL. Voici un aperçu des opérations effectuées :   * J'ai créé un Dockerfile pour le back-end et un autre pour le front-end. Ces Dockerfiles décrivent les étapes nécessaires pour construire des images Docker pour chaque partie de l'application.        * Pour le back-end, j'ai utilisé une image Maven pour construire le projet Java. Le Dockerfile définit les étapes pour télécharger les dépendances Maven, copier les fichiers de l'application, compiler le projet et exposer le port nécessaire pour l'exécution de l'application. (Voir DockerFile pour le back en Annexe O) * Pour le front-end, j'ai utilisé une image Node.js pour construire l'application Vue.js. Le Dockerfile installe les dépendances Node.js, copie les fichiers de l'application, construit l'application et utilise une image Nginx pour servir les fichiers statiques.      * J'ai utilisé Docker Compose pour orchestrer le déploiement de l'ensemble de l'application. Le fichier docker-compose.yml décrit les services nécessaires à l'application : MySQL pour la base de données, phpMyAdmin pour l'administration de la base de données, le back-end et le front-end. ( Voir le fichier DockerCompose en annexe P)   + Enfin, j'ai utilisé Docker Compose pour lancer tous les services avec une seule commande, ce qui simplifie le processus de déploiement et de gestion de l'application. (docker-compose up –d) | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| Docker | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| J’ai travaillé seul | | | | | | | | |
|  | | | | | | | | |
| **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)* | | | | | | | | |
| Github : <https://github.com/Adil-ha/dossier_pro/tree/main/blog> | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| **BAC** | **Lycée Paul Hazard** | **2010** |
| **BTS** | **Lycée Paul Hazard** | **2013** |
| **Titre Professionnelle de maintenance** | **Afpa Lomme** | **2017** |
| **Titre Professionnelle de DWWM** | **Afpa Roubaix** | **2022** |
| 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. |

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

*HANOUN Adil*

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.

*Villeneuve d’ascq*

25/07/2024

Fait à le

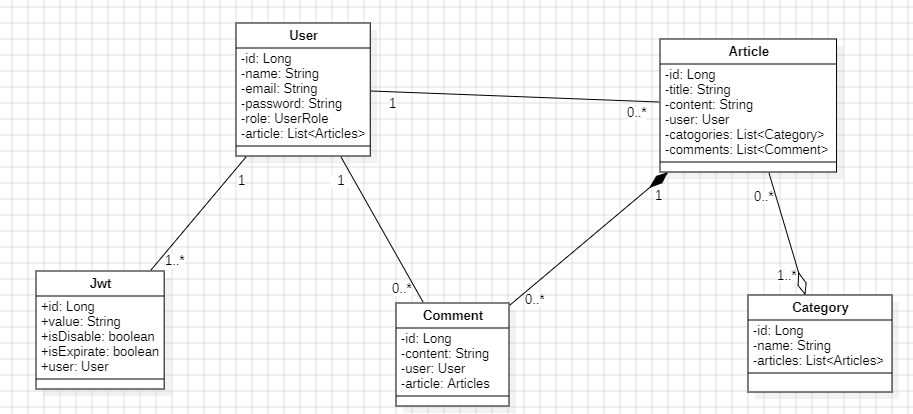
pour faire valoir ce que de droit.

Signature :

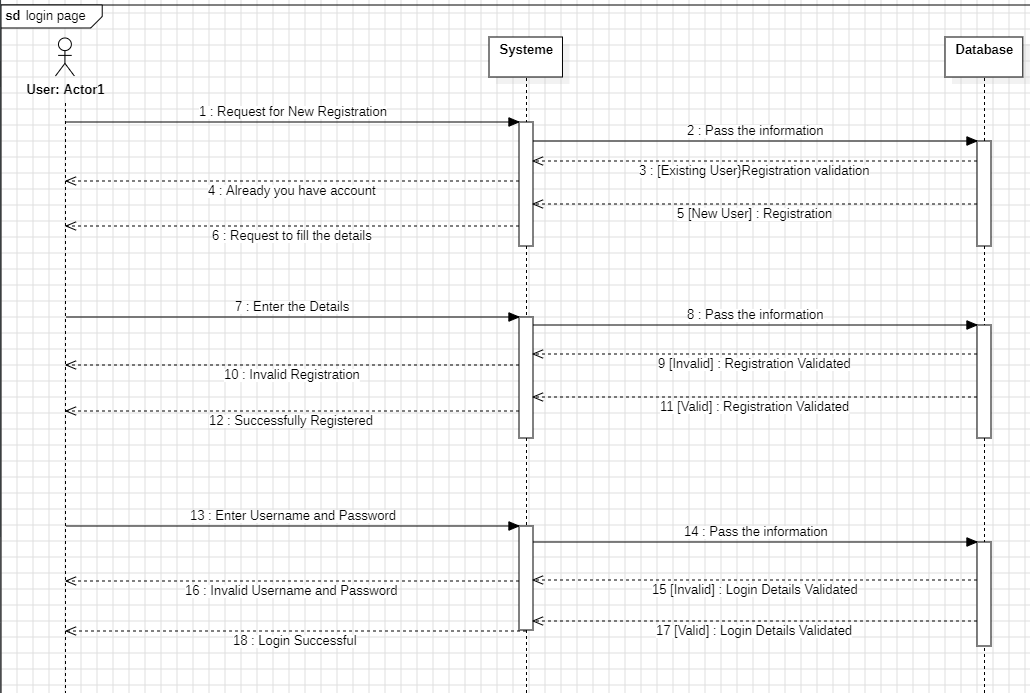
|  |
| --- |
| **Documents illustrant la pratique professionnelle** |
|  |
| *(facultatif)* |
| **Intitulé** |
| Cliquez ici pour taper du texte. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **Annexes** |
|  |
| *(Si le RC le prévoit)* |
| Annexe A      Annexe B    Annexe C    Annexe D    Annexe E    Annexe F    Annexe G    Annexe H    Annexe I    Annexe J    Annexe K |

Annexe L



Annexe M



Annexe N



Annexe O



Annexe P

