



DOSSIER PROFESSIONNEL (DP)

<i>Nom de naissance</i>	▶ Aliyev
<i>Nom d'usage</i>	▶ Aliyev
<i>Prénom</i>	▶ Ogtay
<i>Adresse</i>	▶ 42 rue Saint André, 57950 Montigny-les-Metz

Titre professionnel visé

Concepteur développeur d'applications– NIVEAU 6

MODALITE D'ACCES :

- 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>

Sommaire

Exemples de pratique professionnelle

Développer une application sécurisée	p. 5
▶ Mise en place de l'environnement de développement et participation à la gestion de projet.....	p. 5
▶ Conception de l'interface utilisateur et optimisation avec réactivité.....	p. 8
▶ Expertise en Développement de Composants Métier Full-Stack avec Java Spring Boot	p. 11
Concevoir et développer une application sécurisée organisée en couches	p. 14
▶ Prototypage d'application et modélisation MCD-MLD de la base de données.....	p. 14
▶ Création d'une base de données relationnelle avec Spring Boot et Hibernate.....	p. 16
▶ Conception logicielle d'une application multiservices.....	p. 19
Préparer le déploiement d'une application sécurisée	p. 22
▶ Préparation et exécution de tests dans l'environnement IDE, avec validation via Postman.....	p. 25
▶ Préparation et documentation le déploiement d'une application.....	p. 28
▶ Participation à la mise en production dans une approche DevOps.....	p. 29
Titres, diplômes, CQP, attestations de formation (<i>facultatif</i>)	p. 32
Déclaration sur l'honneur	p. 33
Documents illustrant la pratique professionnelle (<i>facultatif</i>)	p. 34
Annexes (<i>Si le RC le prévoit</i>)	p. 35

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 Développer une application sécurisée

Exemple n°1 ► Configuration de l'Environnement de Développement et contribution à la gestion de projet

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

Dans le cadre de mes projets fil rouge « Garage Associatif », j'accorde une attention particulière à l'installation et à la configuration optimales de mon environnement de travail. Cette démarche est cruciale pour garantir une productivité maximale et des résultats de qualité. Lorsque je débute un nouveau projet, ma première étape consiste à analyser ses besoins spécifiques en termes de technologies et d'outils. Je commence par une lecture minutieuse du cahier des charges afin de bien comprendre les attentes en termes de fonctionnalités, de performances et de contraintes techniques. Cela me permet d'identifier les langages de programmation, les Framework et les bases de données les mieux adaptés au projet.

Je commence par sélectionner les bons outils en fonction des exigences du projet, en prenant en compte des éléments tels que les langages de programmations (Java, JavaScript), le Framework (React JS, Spring Boot), la base de données (MySQL), et les autres technologies requises. Une fois les outils choisis, je procède à leur installation en suivant les meilleures pratiques et les recommandations des développeurs de la communauté. Je m'assure également de configurer chaque composant de manière à optimiser les performances et à garantir la compatibilité entre eux. Pour faciliter la gestion de mes projets, j'utilise souvent des outils de gestion de versions tels que Git et Trello, qui me permettent de collaborer efficacement avec mon binôme et de suivre l'évolution du code source.

Après avoir installé et configuré mon environnement de développement, je configure Docker Desktop pour faciliter le déploiement et l'orchestration de conteneurs, assurant ainsi une mise en production plus fluide et des environnements de test cohérents.

J'ai également installé Postman pour tester les requêtes API, valider le bon fonctionnement des endpoints de l'API et automatiser les tests de régression, assurant ainsi la qualité et la fiabilité des services fournis par l'application web.

En parallèle, je configure mon environnement de développement intégré (IDE) comme IntelliJ et WebStorm en personnalisant les paramètres, les extensions pour faciliter la gestion du code, les raccourcis clavier et les plugins pour répondre aux besoins spécifiques du projet. J'ai utilisé React Snippets qui offre des fonctionnalités telles que la complétion automatique, les raccourcis de code et la navigation améliorée pour travailler plus efficacement avec React.js. Pour le développement avec Spring Boot dans IntelliJ IDEA, je suis également utilisé des plugins spécifiques pour améliorer mon productivité comme : Lombok Plugin, Database Tools and SQL

Dans le cadre de mon projet fil rouge, mon binôme et moi nous sommes réunis pour définir les outils de communication, de stockage et de partage des fichiers afin d'optimiser notre efficacité. Après avoir analysé différentes options, J'ai opté pour des environnements et des outils de partage et de stockage dans le cloud tels que MegaCloud et GitHub. J'ai également défini nos échéances pour le développement des fonctionnalités de l'application.

DOSSIER PROFESSIONNEL (DP)

Pour une gestion efficace des tâches, nous avons utilisé Trello. Chaque carte sur Trello représentait une tâche ou un objectif spécifique. Nous avons structuré notre tableau Trello en plusieurs colonnes : "À faire", "En cours", "En révision" et "Terminé".

Nous avons adopté la méthode Agile Scrum pour structurer notre travail et optimiser notre efficacité. Chaque semaine, nous organisons des sprints, durant lesquels nous définissons des objectifs clairs et atteignables. Lors de la planification de sprint, nous utilisons le backlog pour prioriser les tâches en fonction de leur importance et de leur urgence.

Chaque deux jours, nous réalisons des réunions d'avancement de type "stand-up meeting" où chacun exposait ce qu'il avait fait la veille, ce qu'il comptait faire aujourd'hui, et les obstacles rencontrés. Cela nous a permis de rester concentrés et de réagir rapidement en cas de problème.

À la fin de chaque sprint, nous tenions une réunion de rétrospective pour évaluer ce qui avait bien fonctionné et ce qui pouvait être amélioré. Nous avons utilisé ces retours pour adapter notre manière de travailler et pour continuellement améliorer notre processus

Enfin, J'ai fixé la date de notre prochaine réunion pour faire le point sur l'avancement du projet et ajuster notre plan si nécessaire. Cette approche nous permet de rester organisés, de travailler de manière collaborative et de progresser efficacement vers nos objectifs communs.

Pour le développement de notre projet, nous avons utilisé GitHub comme système de gestion de versions. Le code source, tant pour le front-end que pour le back-end, était stocké sur GitHub. Chaque modification du code était clairement documentée dans les commentaires des commits, ce qui facilitait la collaboration et la compréhension des changements effectués

2. Précisez les moyens utilisés :

Docker Desktop : Déploiement portable.

Postman : Test d'API simplifié.

IntelliJ IDEA et WebStorm : Développement avancé.

GitHub : Collaboration efficace.

Documentation

- W3school
- StackOverflow
- GitHub
- Udemy

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

4. Contexte

Nom de l'entreprise, organisme ou association ► **METZ NUMERIC SCHOOL**

Chantier, atelier, service ► Service

Période d'exercice ► Du 25/01/2024 au 31/05/2024

5. Informations complémentaires *(facultatif)*

Cliquez ici pour taper du texte.

Activité-type 1 Développer une application sécurisée

Exemple n°2 ▶ Développement Front-End et Optimisation avec Réactivité

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

J'ai commencé à la coder en utilisant HTML, CSS et JavaScript. Ce processus a nécessité une attention particulière aux détails et une compréhension profonde des besoins fonctionnels du projet. En parallèle, J'ai continué à collaborer avec mon binôme pour assurer une synchronisation efficace et une communication transparente tout au long du développement

Pour garantir une expérience utilisateur optimale sur une variété de périphériques, J'ai intégré des principes de conception responsive à mon travail. Cela impliquait l'utilisation de media queries et de grilles flexbox pour que l'interface s'adapte de manière fluide aux différentes tailles d'écrans, allant des smartphones aux ordinateurs de bureau. Chaque élément de l'interface a été minutieusement testé pour assurer sa convivialité sur tous les appareils, offrant ainsi une accessibilité maximale à tous les utilisateurs.

Dans le souci d'optimiser la structure et la sémantique du code, J'ai utilisé des balises HTML5 sémantiques telles que <header>, <nav>, <main>, <section> et <footer>. En utilisant telles sémantiques J'ai amélioré la lisibilité du code pour les développeurs, mais aussi renforce la compréhension du contenu par les navigateurs et les moteurs de recherche, favorisant ainsi un meilleur référencement et une meilleure accessibilité.

J'ai utilisé les polices "Montserrat" comme principales familles de polices dans mon design, avec une sauvegarde générique de type sans-serif au cas où ces polices spécifiques ne seraient pas disponibles sur le système de l'utilisateur.

L'organisation méthodique des règles CSS et l'utilisation de commentaires clairs ont également simplifié la maintenance à long terme du code, facilitant ainsi les futures mises à jour et les ajustements nécessaires. J'ai défini un ensemble de variables CSS personnalisées pour les couleurs et la typographie, et les déclare au niveau de la racine du document (:root), ce qui permet de les utiliser dans toute la feuille de style. L'utilisation de variables CSS comme celles-ci permet une gestion plus efficace et flexible des styles dans une feuille de style, facilitant la personnalisation et la maintenance du code.

J'ai commencé par réaliser la page principale de l'application, puis j'ai ajouté une page de connexion / inscription, une section dédiée aux voitures d'occasion, une fonctionnalité de location des box garage, et un système de réservation pour l'entretien des véhicules, en veillant à ce que l'ensemble soit responsive.

Pour la page d'inscription, j'ai inclus un processus de double authentification, comprenant la connexion, l'inscription et l'activation du compte utilisateur. De plus, j'ai ajouté une fonctionnalité de réinitialisation de mot de passe.

En utilisant la librairie react-router-dom, J'ai réussi à rendre mon application plus réactive et rapide. J'ai utilisé useState et d'autres hooks pour créer les fonctionnalités telles que la déconnexion, la modification et la création de réservations. J'ai également utilisé localStorage pour sauvegarder le token envoyé par le backend, permettant ainsi à l'utilisateur de rester connecté à son compte tant que le token est actif.

DOSSIER PROFESSIONNEL (DP)

J'ai utilisé react-calendar-datetime pour organiser les réservations et les locations, permettant ainsi aux utilisateurs de planifier facilement leurs activités. En parallèle, J'ai intégré react-hook-form pour récupérer et valider les données des formulaires.

Cela m'a permis de maintenir un code propre et organisé, en centralisant la logique de gestion des formulaires et en assurant une expérience utilisateur fluide et sans erreur lors de l'envoi des données au backend.

"Pour rendre l'interface plus interactive, j'ai utilisé les `react-modal`. Ainsi, lorsque l'utilisateur sélectionne une option dans la liste de réservations ou de locations, ou choisit son véhicule, une petite fenêtre (modal) s'ouvre pour afficher les informations concernant son choix.

Les modals sont des fenêtres superposées à l'interface principale qui permettent d'afficher des informations ou des actions sans quitter la page en cours. Ils sont souvent utilisés pour confirmer des actions, afficher des formulaires ou des informations supplémentaires de manière plus conviviale et sans redirection."

De plus, J'ai implémenté un système de réservation pour l'entretien des véhicules, en utilisant des fonctionnalités avancées telles que la déconnexion, la modification et la création de réservations. En sauvegardant le token d'authentification dans le localStorage, J'ai assuré que les utilisateurs restent connectés à leur compte tant que le token est actif, offrant ainsi une expérience utilisateur continue et sécurisée

J'ai créé une fonctionnalité de réinitialisation de mot de passe. Si l'utilisateur oublie son mot de passe, dans la rubrique 'Mot de passe oublié', il peut saisir l'adresse email qu'il a utilisée lors de l'inscription. Ensuite, il recevra un code avec lequel il pourra réinitialiser son nouveau mot de passe.

J'ai utilisé des commentaires pour rendre mon code plus lisible et compréhensible pour mon binôme travaillant sur le projet, ainsi que pour moi-même à l'avenir.

2. Précisez les moyens utilisés :

Ordinateur Portable.

Connexion Internet.

J'utilisé IDE WebStorm pour éditer et créer des pages en HTML, CSS, JS et Pour travailler avec Framework React JS

Google Fonts- Pour intégrer les polices principales dans mon projet.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

Communications régulières avec Intervenant pour information supplémentaires et correction, test de mon code.

4. Contexte

Nom de l'entreprise, organisme ou association ► **METZ NUMERIC SCHOOL**

Chantier, atelier, service ► Service

Période d'exercice ► Du 25/01/2024 au 30/05/2024

5. Informations complémentaires (*facultatif*)

Cliquez ici pour taper du texte.

Activité-type 1 Développer une application sécurisée

Exemple n°3 ▶ Expertise en Développement de Composants Métier Full-Stack avec Java Spring Boot et React.js

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

Dans le cadre de mon développement professionnel, J'ai acquis une expertise significative dans la création de composants métier pour des applications full-stack. En utilisant Java Spring Boot pour construire des Back-End robustes et sécurisés, J'ai développé une compréhension approfondie de la gestion des données et de la logique métier.

En collaborant étroitement avec des équipes Front-End, J'ai intégré des technologies telles que React.js pour créer des interfaces utilisateur dynamiques et réactives. En associant ces technologies, J'ai pu concevoir des systèmes complets offrant des fonctionnalités avancées tout en garantissant une expérience utilisateur fluide et intuitive. En tant que développeur de composants métier, J'ai travaillé sur une gamme de fonctionnalités allant de la gestion des utilisateurs et des autorisations avec JWT (JSON Web Tokens) à la mise en œuvre de workflows complexes et à la manipulation sécurisée des données sensibles.

Pour garantir une interaction fluide avec la base de données, J'ai mis en œuvre des repositories Spring Data, qui ont permis de créer des liaisons transparentes entre les objets métier et la couche de persistance. Grâce à ces repositories, les opérations CRUD (Créate, Read, Update, Delete) ont été simplifiées et standardisées, améliorant ainsi la maintenabilité du code et réduisant le risque d'erreurs.

J'ai créé des repositories pour chaque entité métier afin de faciliter la manipulation des données dans la base de données. Ces repositories sont responsables de la persistance des objets métier et permettent d'effectuer des opérations CRUD de manière efficace et sécurisée.

De plus, J'ai intégré la bibliothèque Java Mail pour gérer l'envoi d'e-mails à partir de l'application. Cela inclut la configuration des paramètres SMTP, la création et l'envoi des messages électroniques pour des fonctionnalités telles que la confirmation d'inscription, la réinitialisation de mot de passe ou d'autres notifications importantes pour les utilisateurs. L'utilisation de Java Mail assure une communication fiable et sécurisée avec les utilisateurs, renforçant ainsi l'expérience utilisateur globale de l'application. J'ai utilisé Java Mail pour configurer l'envoi de notifications par e-mail. Pour garantir une interaction fluide avec Gmail, J'ai configuré les paramètres SMTP avec le port 587, en utilisant le protocole de sécurité STARTTLS. Cela permet d'assurer une connexion sécurisée et fiable avec le serveur Gmail pour l'envoi des e-mails de notification.

En plus de cela, J'ai développé des services métier Java pour encapsuler la logique métier complexe. Ces services ont été conçus pour être modulaires et réutilisables, facilitant ainsi l'extension et la maintenance du système. J'ai veillé à ce que ces services respectent les principes de conception SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) pour garantir une architecture flexible et évolutive. Pour assurer la sécurité des données sensibles, J'ai implémenté des mécanismes de cryptage et de hachage au niveau des couches applicatives et de persistance. Cela a permis de protéger les informations confidentielles contre les accès non autorisés et les violations de données.

J'ai écrit un Dockerfile pour définir l'environnement de l'application, en spécifiant les dépendances nécessaires, les configurations et les commandes d'exécution. Ce fichier permet de construire une image Docker qui encapsule toutes les composantes nécessaires à l'application. En utilisant Docker, j'ai construit l'image de l'application et l'ai testée localement pour m'assurer qu'elle fonctionne correctement dans un environnement isolé. Cela garantit que l'application se comporte de la même manière, quel que soit l'endroit où elle est déployée.

En somme, mon expertise en développement de composants métier s'étend à la conception et à la mise en œuvre de modèles de données, de repositories, de services métier et de mécanismes de sécurité, garantissant ainsi la robustesse, la performance et la sécurité des applications full-stack que j'ai contribué à développer.

2. Précisez les moyens utilisés :

Ordinateur Portable.

Connexion Internet.

IDE -Webstorm avec framework REACT JS

IDE-INTELIJ avec framework Spring Boot

hub.docker.com-pour documentations de Docker.

Google compte – créer et chiffré un mot de passe google application pour utiliser dans mon application.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

4. Contexte

Nom de l'entreprise, organisme ou association ► **METZ NUMERIC SCHOOL**

Chantier, atelier, service ► Service

Période d'exercice ► Du 25/01/2024 au 30/05/2024

5. Informations complémentaires *(facultatif)*

Cliquez ici pour taper du texte.

Activité-type 2

Concevoir et développer une application sécurisée organisée en couches

Exemple n°1 ▶ Maquettage d'application et model MCD-MLD base des données

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

J'ai me suis concentrer sur la création de maquettes pour l'interface utilisateur. Pour ce faire, J'ai utilisé l'outil de conception Figma, qui m'a permis de créer des maquettes interactives en intégrant des éléments de conception pour les formats web et mobile. En utilisant des fonctionnalités de Figma telles que les composants et les variantes, J'ai assuré la cohérence visuelle tout en facilitant les ajustements nécessaires pour différentes tailles d'écrans.

Dans mes maquettes, J'ai également intégré des icônes pertinentes pour améliorer l'expérience utilisateur et rendre l'interface plus intuitive. J'ai choisi des icônes appropriées en utilisant des extensions intégrées dans le logiciel pour chaque fonctionnalité de l'application, en veillant à ce qu'elles soient reconnaissables et esthétiquement cohérentes avec le reste du design. J'ai choisi avec soin chaque icône, en m'assurant qu'elle respecte les directives de la politique de conception.

Après avoir travaillé sur ces maquettes initiales, j'ai ensuite commencé à créer un wireframe de base fidèle pour montrer comment l'application va réellement fonctionner. L'objectif principal était de visualiser la disposition générale des éléments de l'interface utilisateur, en mettant l'accent sur la navigation, les interactions principales et la présentation des informations. En élaborant ce wireframe, je voulais donner une idée concrète de l'apparence et du fonctionnement de l'application, en mettant en évidence les flux utilisateur principaux et les points d'interaction clés.

Après avoir travaillé sur les maquettes et les wireframes, je vais passer à la création de prototypes interactifs. L'objectif principal de cette étape est de donner vie à l'interface utilisateur et de simuler le comportement de l'application. En créant des prototypes, je pourrai mieux démontrer les flux de navigation, les interactions utilisateur et le fonctionnement général de l'application.

Les prototypes permettent aux parties prenantes, y compris les membres de l'équipe de développement et les utilisateurs finaux, de visualiser concrètement l'expérience utilisateur attendue. Cela permet également de recueillir des commentaires précieux et d'identifier les éventuels problèmes ou améliorations dès les premières étapes du processus de développement.

J'ai choisi les photos sur internet en respectant les couleurs et le design de l'interface, et J'ai effectué quelques retouches pour enlever certains arrière-plans afin qu'elles correspondent parfaitement à mon design. En sélectionnant attentivement les images en ligne, J'ai veillé à ce qu'elles s'harmonisent avec le style global de l'application, renforçant ainsi la cohérence visuelle et l'immersion de l'utilisateur dans l'expérience.

Par ailleurs, pour la modélisation de la base de données, J'ai utilisé le logiciel de modélisation JMERISE. Ce logiciel m'a permis de représenter visuellement les différentes entités, relations et attributs de la base de données, en suivant les méthodes MCD, MLD et Merise que j'avais préalablement étudiées. Grâce à JMERISE, J'ai pu créer une représentation claire et structurée de la base de données, facilitant

ainsi sa compréhension et son implémentation ultérieure dans le projet. Les fonctionnalités avancées de ce logiciel m'ont également permis d'exporter mes maquettes au format JPG et PDF, ce qui m'a donné la possibilité de les partager facilement avec les membres de l'équipe pour obtenir des retours et effectuer des corrections si nécessaire.

En combinant une approche méthodique de conception avec l'utilisation d'outils modernes et spécialisés, J'ai pu créer des maquettes d'interface utilisateur et des modèles de base de données qui serviront de fondement solide pour le développement de l'application web du garage associatif. Après avoir terminé les maquettes, J'ai organisé des sessions de test avec mon binôme afin de recueillir des retours utiles sur l'expérience utilisateur. Ces tests m'ont permis d'identifier des points à améliorer et de résoudre rapidement les problèmes de navigation ou d'ergonomie. Pendant ce temps, J'ai également continué à peaufiner la modélisation de la base de données, en optimisant les relations et la structure pour assurer des performances optimales

2. Précisez les moyens utilisés :

Utilisation du Logiciel Figma pour la création des maquettes Frontend en haute et basse fidélité.

Concernant les diagrammes de base des données J'ai utilisé application JMERISE.

Une base de données MySQL.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **METZ NUMERIC SCHOOL**

Chantier, atelier, service ▶ Service

Période d'exercice ▶ Du 25/01/2024 au 30/05/2024

5. Informations complémentaires (*facultatif*)

Activité-type 2

Concevoir et développer une application sécurisée organisée en couches

Exemple n°1 ▶ Conception d'une Base de Données Relationnelle avec Spring Boot et Hibernate

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

Dans le cadre de mon projet de développement avec Spring Boot, j'ai entrepris la création et le remplissage des tables pour la base de données. En utilisant Hibernate, j'ai défini les entités Java correspondantes à chaque table, en annotant les classes avec des directives telles que `@Entity`, `@Table` et `@Column` pour définir la structure des données.

J'ai utilisé les annotations telles que `@LocalDate` pour définir les champs de type date locale et l'annotation `@Date` pour spécifier les dates. J'ai également utilisé `@GeneratedValue` pour que l'identifiant de chaque table soit auto-incrémenté, et l'annotation `@Temporal(TemporalType.TIMESTAMP)` pour gérer les types de données temporelles.

Les entités ont été soigneusement conçues pour refléter les besoins métier de l'application. Par exemple, j'ai créé une entité `User` pour représenter les utilisateurs de l'application, avec des attributs tels que l'identifiant, le nom d'utilisateur et le mot de passe. En parallèle, j'ai utilisé des annotations telles que `@OneToMany` et `@ManyToOne` pour établir les relations entre les entités, assurant ainsi une modélisation efficace de la base de données.

En plus de cela, j'ai également mis en place des validations au niveau des entités en utilisant des annotations telles que `@NotNull`, `@Size`, et `@Email` pour garantir l'intégrité et la validité des données. L'intégration de ces validations permet de s'assurer que les données respectent les contraintes définies avant d'être persistées dans la base de données.

Une fois les entités définies, j'ai procédé au peuplement des tables avec des données fictives à l'aide de générateurs intégrés à Spring Boot. Ces générateurs ont permis de créer des enregistrements aléatoires réalistes, ce qui m'a permis de tester les fonctionnalités de l'application dans divers scénarios. Cependant, j'ai veillé à ce que les données générées respectent les contraintes de validation définies dans les entités, afin d'éviter toute incohérence ou erreur lors de l'insertion dans la base de données. Ce processus de création et de peuplement des tables a été essentiel pour le développement itératif de l'application, en me permettant de tester et d'itérer rapidement sur les fonctionnalités tout en assurant la qualité et la cohérence des données. Ce processus de développement a également renforcé ma compréhension de la manière dont les entités Java sont liées à la structure de la base de données relationnelle, tout en me permettant d'explorer les fonctionnalités avancées de Hibernate, telles que les associations et les générateurs de données.

En ce qui concerne les jointures entre les tables, Spring Boot facilite la gestion des relations grâce à des annotations telles que `@OneToMany`, `@ManyToOne`, `@OneToOne`, et `@ManyToMany`. Ces annotations permettent de définir des relations entre les entités et d'effectuer des opérations de jointure de manière transparente. Par exemple, dans une application de commerce électronique, une entité `Order` pourrait être liée à plusieurs entités `Product` via une relation `@ManyToOne`, où chaque commande contient plusieurs produits. De même, une entité `Product` pourrait être liée à plusieurs entités `Order` via une relation `@OneToMany`, représentant le fait qu'un produit peut être inclus dans plusieurs

commandes. Ces mécanismes de jointure simplifient la manipulation des données et permettent de modéliser efficacement les relations entre les différentes entités de l'application.

En conclusion, l'approche méthodique utilisée pour créer et peupler les tables a été un pilier essentiel de la phase de développement, contribuant à la robustesse et à la fiabilité de l'application. Les fonctionnalités avancées de Hibernate et les mécanismes de jointure de Spring Boot ont permis une modélisation efficace des relations entre les entités, renforçant ainsi la qualité et la cohérence des données.

2. Précisez les moyens utilisés :

Spring Boot : J'ai choisi Spring Boot comme Framework principal pour le développement de l'application, en raison de sa simplicité de configuration et de sa facilité d'intégration avec d'autres technologies.

Hibernate : J'ai utilisé Hibernate comme ORM (Object-Relationnel Mapping) pour faciliter la manipulation des données et la gestion des entités Java dans la base de données relationnelle.

Base de données relationnelle : Pour le stockage des données, j'ai opté pour une base de données relationnelle, telle que MySQL en fonction des besoins spécifiques du projet.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

Intervenant spécialise conception la base de données pour avoir des conseils sur les conception et cardinalités entre les tables.

4. Contexte

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

Chantier, atelier, service ▶ Service

Période d'exercice ▶ Du 25/01/2024 au 30/05/2024

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer une application sécurisée organisée en couches

Exemple n°3 ▶ Architecture Logicielle d'une Application Full Stack Multiservice avec React.js, Spring Boot, MySQL et Docker

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

Je vais définir l'architecture logicielle de mon application, qui est Microservice, combinant différentes technologies pour répondre à divers besoins. En premier lieu, je vais structurer mon application en suivant une architecture Microservice, où chaque service représente une fonctionnalité ou un domaine spécifique de l'application. Cela permet une modularité accrue, une évolutivité indépendante et une maintenance simplifiée.

Pour le front end, je vais utiliser React.js, un Framework JavaScript populaire pour créer des interfaces utilisateur dynamiques et réactives. Je vais concevoir des composants réutilisables et bien organisés pour garantir une expérience utilisateur cohérente et intuitive.

Concernant le back end, je vais opter pour Spring Boot, un Framework Java qui facilite le développement d'applications web robustes et extensibles. Chaque service back end sera responsable de la logique métier spécifique, exposant des API RESTful pour la communication avec le front end et les autres services. Pour assurer la communication entre le front end et le back end, je vais mettre en place des API RESTful. Le front end utilisera la bibliothèque Axios, une populaire librairie JavaScript pour effectuer des requêtes HTTP, aux endpoints correspondants du back end pour récupérer ou envoyer des données. Cette librairie offre une interface simple et flexible pour interagir avec les API, facilitant ainsi le développement de l'application.

Pour faciliter les échanges entre le frontend et le backend, j'ai opté pour l'utilisation d'API RESTful. Ces dernières permettent une communication simple et efficace en utilisant des requêtes HTTP. Ainsi, par le biais de requêtes GET, POST, PUT, et DELETE, le frontend peut récupérer, envoyer, mettre à jour, ou supprimer des données sur le backend. Cette approche garantit une séparation claire des responsabilités entre les différentes parties de l'application.

Pour garantir une mise en production fluide et une portabilité de l'application, on a choisi d'utiliser Docker pour la conteneurisation. Docker permet de packager l'application et ses dépendances dans des conteneurs légers et portables, assurant ainsi une exécution cohérente sur différents environnements. Grâce à Docker, on peut facilement déployer l'application sur n'importe quel serveur compatible sans se soucier des différences d'environnement.

En ce qui concerne la gestion des données dans le back end, Spring Boot fournira une couche d'accès aux données robuste, utilisant des technologies telles que Hibernate pour interagir avec la base de données MySQL. Les entités et les repositories Hibernate seront soigneusement conçus pour refléter la structure de la base de données et faciliter les opérations de lecture/écriture. Pour assurer la cohérence des données entre le front end et le back end, des validations appropriées seront effectuées à chaque étape du processus, depuis la saisie des données dans l'interface utilisateur jusqu'à leur persistance dans la

DOSSIER PROFESSIONNEL (DP)

base de données. Des mécanismes de gestion des erreurs seront également mis en place pour gérer les cas où des erreurs se produisent lors de la communication avec la base de données.

2. Précisez les moyens utilisés :

Un ordinateur portable.

Outils de développement : IntelliJ IDEA et WebStorm.

Langages de programmation : Java.

Langages Web : HTML5, CSS, JavaScript.

Framework et bibliothèques : Spring Boot/ React JS

Docker Desktop.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

Echanges d'informations avec les intervenants.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **METZ NUMERIC SCHOOL**

Chantier, atelier, service ▶ Service

Période d'exercice ▶ Du 25/05/2024 au 30/05/2024

5. Informations complémentaires (*facultatif*)

Activité-type 3 Préparer le déploiement d'une application sécurisée

Exemple n°1 ▶ Application des règles de sécurité et des autorisations nécessaires pour garantir l'intégrité et la confidentialité des données.

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

Dans le cadre de mon projet Java-Spring Boot, je suis en train de mettre en œuvre des mesures de sécurité essentielles pour protéger les données des utilisateurs stockées dans la base de données et sécuriser les échanges entre le front-end et le back-end. J'utilise des JWT Tokens pour l'authentification et l'autorisation, avec une durée de vie limitée, garantissant ainsi un accès sécurisé aux ressources de l'application. J'ai défini une durée de vie pour mes tokens afin de limiter les risques d'utilisation frauduleuse et de renforcer la sécurité des sessions utilisateur. Pour renforcer davantage la sécurité, j'applique des algorithmes de hachage pour crypter les mots de passe des utilisateurs, assurant ainsi leur confidentialité même en cas de violation de la base de données. Cette approche globale renforce la sécurité de l'application et assure une protection robuste des données sensibles et des transactions entre les différentes couches de l'application.

Dans le cadre de la sécurité, J'ai mis en place un service de validation lors de l'inscription des utilisateurs. Dès qu'un utilisateur s'inscrit, il reçoit un code de validation à son adresse e-mail. Cette étape supplémentaire garantit que seuls les utilisateurs authentiques peuvent accéder à l'application, renforçant ainsi la protection des données et la confidentialité des comptes. En intégrant cette fonctionnalité, je m'assure de respecter les meilleures pratiques en matière de sécurité des comptes utilisateur et de prévenir toute tentative d'accès non autorisé.

J'ai implémenté un système de gestion des accès basé sur des rôles et des permissions, permettant de contrôler précisément l'accès des utilisateurs aux fonctionnalités et ressources de l'application. Chaque utilisateur se voit attribuer un rôle spécifique, déterminant les actions qu'il peut effectuer et les données auxquelles il peut accéder, garantissant ainsi une sécurité optimale et une gestion efficace des autorisations. Pour renforcer encore davantage la sécurité des mots de passe utilisateurs, J'ai intégré un validateur de mots de passe robuste. Ce validateur s'assure que les mots de passe choisis par les utilisateurs respectent des critères de complexité stricts, tels que la longueur minimale, l'utilisation de caractères spéciaux, de chiffres et de lettres majuscules et minuscules. En imposant ces exigences, je m'assure que les mots de passe sont suffisamment robustes pour résister aux attaques par force brute et aux tentatives de devinette. Par ailleurs, pour minimiser les risques liés aux attaques de type Cross-Site Scripting (XSS), J'ai appliqué des méthodes de validation et de filtrage des données entrantes. En désactivant l'exécution de scripts potentiellement dangereux et en échappant correctement les caractères spéciaux, je protège l'application contre les tentatives d'injection de scripts malveillants, préservant ainsi la sécurité des utilisateurs.

Grâce à ces mesures de sécurité, J'ai réussi à assurer une authentification et une autorisation sécurisées des utilisateurs, protéger les mots de passe des utilisateurs même en cas de violation

de la base de données, garantir que seuls les utilisateurs vérifiés peuvent créer des comptes, gérer efficacement les permissions des utilisateurs pour contrôler l'accès aux fonctionnalités et ressources de l'application, renforcer la robustesse des mots de passe pour résister aux attaques par force brute, et protéger l'application contre les attaques de type XSS, préservant ainsi la sécurité et l'intégrité des données. En réalisant ces tâches, J'ai développé et renforcé plusieurs compétences clés, notamment la maîtrise des concepts et techniques d'authentification et d'autorisation avec JWT, l'implémentation de mesures de sécurité pour la gestion des mots de passe (hachage et validation), la mise en œuvre de mécanismes de validation des utilisateurs et de gestion des accès basés sur des rôles, la protection des applications web contre les attaques XSS et autres menaces de sécurité, et une meilleure compréhension des pratiques de sécurité des données dans le développement d'applications web.

2. Précisez les moyens utilisés :

Outils de développement : IntelliJ IDEA

Langages de programmation : Java

Framework et bibliothèques : Spring Boot

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

4. Contexte

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

Chantier, atelier, service ▶ Service

Période d'exercice ▶ Du 25/01/2024 au 30/05/2024

5. Informations complémentaires *(facultatif)*

Cliquez ici pour taper du texte.

Activité-type 3

Préparer le déploiement d'une application sécurisée

Exemple n°1 ► Préparation et Exécution de Tests dans l'Environnement IDE et Validation avec Postman

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

Dans ce dossier, je présente méticuleusement ma méthodologie pour tester chaque composant de l'application, mettant en lumière une approche itérative et axée sur la qualité. En tant que développeur, mon objectif est d'assurer le bon fonctionnement de chaque micro-service, ainsi que leur intégration harmonieuse dans l'écosystème global de l'application.

Pour exécuter une fois que j'ai écrit mes tests unitaires, d'intégration et de frontend dans mon IDE, j'utilise les fonctionnalités intégrées pour les exécuter de manière efficace et pratique. Dans IntelliJ IDEA et WebStorm, pour les tests unitaires, j'utilise JUnit et Mockito pour simuler les comportements des objets dépendants. Pour les tests unitaires, je teste mon service, mon repository et mon Controller. Je peux lancer mes tests directement depuis l'IDE en quelques clics. Je sélectionne simplement les tests individuels, les classes de tests ou même des suites de tests complètes, puis je lance l'exécution. Les résultats sont rapidement affichés, ce qui me permet d'identifier immédiatement tout échec éventuel et de prendre des mesures pour les corriger.

Pour réaliser ces tests, j'utilise une combinaison d'outils puissants intégrés à mon flux de travail de développement. Tout d'abord, j'utilise IntelliJ IDEA et WebStorm comme environnements de développement intégrés (IDE) pour écrire, déboguer et exécuter du code. Pour vérifier l'intégration fluide de mes différents composants, notamment en ce qui concerne l'authentification JWT, je crée des tests d'intégration. Ces tests peuvent impliquer des composants Spring tels que MockMvc pour simuler les requêtes HTTP et vérifier les réponses. IntelliJ IDEA facilite l'exécution des tests directement à partir de l'IDE. Je peux exécuter des tests individuels, des classes de tests ou même des suites de tests complètes en quelques clics.

En ce qui concerne les tests frontend, j'utilise la bibliothèque `@testing-library/react`, qui offre une approche axée sur l'utilisateur pour tester les composants React. Cette bibliothèque met l'accent sur le test du comportement observable des composants, similaire à la façon dont ils sont utilisés par les utilisateurs réels. Avec `@testing-library/react`, je peux écrire des tests clairs et robustes qui garantissent le bon fonctionnement de mes composants, indépendamment de leur implémentation interne. Ces tests me permettent de m'assurer que mon application réagit comme prévu aux interactions utilisateur et aux changements d'état

Pour les tests d'API, je me tourne vers Postman, un outil polyvalent qui me permet de créer, partager et exécuter des requêtes HTTP de manière simple et intuitive. Avec Postman, je peux valider le comportement des API RESTful, tester les différentes routes et méthodes, ainsi que surveiller les réponses pour détecter les. Pour chaque requête, j'utilise l'autorisation avec Bearer Token dès que j'utilise JWT (JSON Web Token). Cela me permet de simuler l'authentification des utilisateurs et de tester les API sécurisées. Pour effectuer des actions spécifiques, l'utilisateur doit être connecté et actif, ce qui me permet de vérifier que les contrôles d'accès et les autorisations sont correctement mis en œuvre.

En outre, Postman offre des fonctionnalités avancées comme l'automatisation des tests avec des scripts, ce qui me permet de valider les réponses et d'assurer que les API fonctionnent comme prévu. Je peux également organiser mes tests en collections et les partager avec mon équipe, facilitant ainsi la collaboration et la revue des tests

Ensuite, j'installe plusieurs navigateurs pour vérifier manuellement l'interface de mon application et m'assurer qu'elle fonctionne correctement sur différents navigateurs. Ceci est crucial pour garantir une expérience utilisateur cohérente et de haute qualité, quel que soit le navigateur utilisé par les utilisateurs finaux.

2. Précisez les moyens utilisés :

Un ordinateur portable.

Outils de développement : IntelliJ IDEA et WebStorm.

Docker Desktop.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

4. Contexte

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

Chantier, atelier, service ▶ Service

Période d'exercice ▶ Du 25/01/2024 au 30/05/2024

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 3

Préparer le déploiement d'une application sécurisée

Exemple n°1 ▶ **Préparation, Documentation et Contribution au Déploiement dans une Approche DevOps**

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

Dans la phase de préparation et de documentation du déploiement d'une application multiservice utilisant Spring Boot, React JS et MySQL, j'ai franchi plusieurs étapes cruciales pour assurer un déploiement fluide et efficace.

Tout d'abord j'ai créé les fichiers Docker File pour configurer l'environnement, le port, le volume, après j'ai créé les images et j'ai installé les images dans Docker Desktop. Pour un projet combinant Spring Boot (Java) et React.js (Node.js), j'ai choisi des images contenant à la fois Java et Node.js et MySQL pour la base de données. Si des dépendances supplémentaires étaient nécessaires, telles que des dépendances Java ou des modules Node.js, j'ai veillé à les installer à l'intérieur du conteneur Docker en utilisant les commandes RUN, par exemple `npm install` pour les dépendances Node.js et `mvn install` pour les dépendances Java.

Ensuite, j'ai procédé à la construction de l'application Spring Boot et de l'application React.js à l'intérieur des conteneurs Docker. Cela s'est fait en exécutant les commandes de construction nécessaires telles que `mvn package` pour Spring Boot et `npm run build` pour React.js.

Concernant l'exposition des ports, j'ai veillé à exposer le port sur lequel l'application Spring Boot écoute (généralement le port 8080) ainsi que le port sur lequel l'application React.js est servie (généralement le port 3000 en développement) en utilisant l'instruction EXPOSE dans les Dockerfiles.

J'ai utilisé des Dockerfiles pour spécifier les dépendances et les étapes de construction, puis j'ai construit ces images à l'aide de la commande `docker build`.

Ensuite, j'ai configuré un fichier Docker Compose pour définir les services, les réseaux et les ports nécessaires à l'application. J'ai référencé les images Docker créées précédemment et exposé les ports appropriés.

Une fois les images Docker prêtes, j'ai choisi un serveur distant chez Hostinger pour héberger l'application. J'ai installé Apache Maven sur ce serveur pour la gestion des dépendances Java et la construction des projets.

En transférant ensuite les images Docker sur le serveur distant à l'aide des commandes `docker save` et `docker load`, j'ai démarré les conteneurs à l'aide de Docker Compose.

Enfin, j'ai vérifié le déploiement en m'assurant que les services étaient accessibles via les ports spécifiés et qu'ils interagissaient correctement.

En documentant chaque étape de ce processus, j'ai assuré sa reproductibilité et sa facilité de maintenance à long terme.

Dans le cadre de la contribution à la mise en production dans une approche DevOps, j'ai également utilisé des GitHub Actions pour automatiser le processus de déploiement. J'ai créé un fichier de workflow nommé `build.yml` pour définir les étapes de construction et de déploiement de l'application.

Après avoir créé ce fichier, j'ai exploré la GitHub Marketplace pour trouver des actions et des configurations préconstruites qui correspondent à mes besoins spécifiques. Ces actions préconstruites m'ont permis d'accélérer le processus de mise en place de pipelines CI/CD tout en garantissant une haute qualité et une sécurité accrue.

En intégrant ces actions dans mon flux de travail GitHub, j'ai pu automatiser des tâches telles que la compilation du code, les tests automatisés, la construction d'images Docker, le déploiement sur des environnements de test et de production, ainsi que la surveillance continue de la qualité du code et des performances de l'application.

Cette approche m'a permis de bénéficier des avantages de l'intégration et du déploiement continus (CI/CD) en réduisant les délais de déploiement, en améliorant la qualité du code et en garantissant une plus grande fiabilité dans les mises en production. De plus, en documentant ces processus dans le cadre de ma démarche DevOps, j'ai assuré la reproductibilité et la facilité de maintenance à long terme de l'infrastructure et des pipelines de déploiement.

2. Précisez les moyens utilisés :

Un ordinateur portable

Docker Desktop

Serveur WEB (Hostinger)

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Pour la réalisation de projet Garage associatif, comme c'était une travaille de groupe je travaillé avec mon binôme - Pierre Louis Adam

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 25/01/2024 au 30/05/2024

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
BAC +3 Mathématique et Informatique	Ministère Education de l'Azerbaïdjan	2003-2007
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.
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.

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] *Ogtay Aliyev*,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je
suis l'auteur(e) des réalisations jointes.

Fait à *Metz* le *17/07/2024*

pour faire valoir ce que de droit.

Signature : *Aliyev Ogtay*

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)