|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | *Lionnet* |
| *Nom d’usage* |  | *Lionnet* |
| *Prénom* |  | *Kévin* |
| *Adresse* |  | *179 rue de la république Villefranche-de-Lauragais* |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| Concepteur développeur d’applications | |
|  | |
| **Modalité d’accès :** | |
|  | |
| X | 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** | | |
| **Développer une application sécurisée** | | **p.** | **5** |
|  |  Installer et configurer son environnement de travail en fonction du projet p. | p. |  |
|  |  Développer des interfaces utilisateurs p. | p. |  |
|  |  Développer des composants métier p | p. |  |
|  |  Contribuer à la gestion d'un projet informatique Développer des composants métier………… | p. |  |
|  |  |  |  |
| **Concevoir et développer une application sécurisée organisée en couches** | | **p.** |  |
|  |  Analyser les besoins et maquetter une application p. | p. |  |
|  |  Définir l'architecture logicielle d'une application p. | p. |  |
|  |  Concevoir et mettre en place une base de donnée relationnelle p | p. |  |
|  |  Développer des composants d’accès aux données SQL et NoSQL p | p. |  |
|  |  |  |  |
| **Préparer le déploiement d’une application sécurisée** | | **p.** |  |
|  |  Préparer et exécuter les plans de tests d'une application p. | p. |  |
|  |  Préparer et documenter le déploiement d'une application p. | p. |  |
|  |  Contribuer à la mise en production dans une démarche Devops p | p. |  |
|  |  |  |  |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | **p.** |  |
| **Déclaration sur l’honneur** | | **p.** |  |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | **p.** |  |
| **Annexes** *(Si le RC le prévoit)* | | **p.** |  |

**Exemples de pratique**

**professionnelle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Développer une application sécurisée** | | | |
| ***Exemple n°1***  | | Installer et configurer son environnement de travail en fonction du projet | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Je prends ici l’exemple de mon projet d’étude **Ploybout**, jeu sur navigateur avec des personnages en 3D avec lesquels on peut défier d’autres joueurs, les améliorer et gérer le déroulement des combats asynchrones par l’intermédiaire de cartes conditionnelles s’activant automatiquement lors des duels.  Je pars du principe ici que l’on part d’une machine vierge quand je détaille plus bas les outils employés. Il s’agit ici de mettre en place l’environnement de travail qui me permet de réaliser le projet avec toutes les fonctionnalités désirées. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| En amont :  Mise en place de l’environnement de travail avec Visual Studio Code comme IDE, choisi pour sa simplicité et sa rapidité.  Installation de Node.js pour la gestion des dépendances, de Git avec GitHub pour le suivi de versions, de PostgreSQL comme système de gestion de base de données.  Côté front :  L’interface utilisateur a été développée avec React et TypeScript, initialisée via un outil de création rapide de projets.  Pour le style, j’ai utilisé un framework CSS utilitaire permettant de composer les interfaces directement avec des classes.  Selon les besoins spécifiques du projet, plusieurs outils ont été intégrés :   * Three.js : une librairie 3D adaptée à React, utilisée pour gérer des scènes interactives et des objets en trois dimensions. * React DnD : une solution de glisser-déposer, compatible avec les environnements ordinateurs et mobiles. * Axios : une bibliothèque pour la gestion des appels HTTP entre le client et le serveur. * React Helmet : un outil permettant de manipuler dynamiquement les balises de l’en-tête HTML, notamment pour l’optimisation du référencement. [Annexe n°1]   Côté back :  L’API a été développée avec Express, un framework minimaliste basé sur Node.js, permettant de structurer les routes et de gérer les requêtes efficacement.  Voici les principaux outils intégrés côté serveur :   * Validateur Express : un système de validation des données entrantes, servant à vérifier et sécuriser les entrées utilisateur avant traitement. * PostgreSQL : un système de gestion de base de données relationnelle utilisé pour stocker et organiser les données de manière fiable. * Bcrypt : utilisé pour sécuriser les mots de passe avant leur enregistrement dans la base de données. * CORS : permet d’autoriser les échanges de données entre le client (*frontend*) et le serveur (*backend*), même s’ils ne sont pas sur le même domaine. * JWT (JSON Web Token) : une solution d’authentification qui permet de générer et vérifier des jetons pour sécuriser les accès. * Seedrandom : Utilisé pour créer des résultats pseudo-aléatoires dans certaines mécaniques métier, comme les combats. * Nodemon : Un outil permettant de redémarrer automatiquement le serveur à chaque modification du code, pour un développement plus fluide. [Annexe n°2] | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (ADRAR)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Développer une application sécurisée** | | | |
| ***Exemple n°1***  | | Développer des interfaces utilisateurs | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Pour un projet personnel de jeu sur navigateur de type wargame, une attention particulière a été portée à la conception d’une interface utilisateur claire et intuitive. Il fallait permettre au joueur de naviguer facilement entre les différentes sections du jeu sans se sentir submergé.  Un autre enjeu a été de réussir à afficher un volume important d’informations sans nuire à la lisibilité. Il a fallu structurer l’affichage de manière cohérente pour que les données restent compréhensibles en un coup d’œil.  Enfin, le projet devait être entièrement *responsive*. Une réflexion a été menée pour adapter l’interface aux supports mobiles, avec des ajustements spécifiques à la disposition et à la densité des éléments. Pour guider ces choix, je me suis appuyé sur des références de jeux existants du même genre afin de m’inspirer notamment de la hiérarchisation des informations. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé Angular comme framework JavaScript afin de m’exercer et de bénéficier d’un environnement robuste et plus directif que Vue ou Svelte par exemple.  J’ai utilisé le framework CSS Tailwind afin de pouvoir itérer rapidement. J’ai prévu quelques classes réutilisables pour les boutons afin de ne pas alourdir le code HTML plus que de raison. J’ai utilisé les préfixes de *breakpoints* de Tailwind afin d’effectuer des réglages particuliers pour les écrans les plus petits. [Annexe n° 3]  Pour l’interface, je me suis modestement inspiré de jeux du genre de la firme Koei ou Paradox Interactive afin de répondre au problème spécifique de ce type de logiciel qui est la disposition des données. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet personnel* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *03/04/2025*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Développer une application sécurisée** | | | |
| ***Exemple n°1***  | | Développer des composants métier | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre du projet fil-rouge **Ploybout**, développée en *fullstack* (React / Node.js), j’ai conçu et implémenté le moteur de combat, un composant métier central du système de jeu.  Cette logique métier repose sur plusieurs règles : gestion des points de vie, initiative (vitesse), positionnement sur une grille, portée des armes, gestion de l’énergie, conditions de cartes, et effets associés (dégâts, déplacement, gain/perte de ressources, etc.). Le combat est entièrement simulé au *back-end*, de manière déterministe à partir d’une graine (*seed*), ce qui le rend reproductible et adapté à un contexte multijoueur ou compétitif.  J’ai structuré cette fonctionnalité sous forme de fonctions pures et testables, garantissant la traçabilité du combat via un combat log, qui permet à la fois d’afficher le déroulement du combat côté *front-end*, et de le rejouer ou de l’analyser. [Annexe n°4] | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| Pour réaliser ce système de combat, j’ai utilisé :   * Node.js (TypeScript) : pour le développement du moteur métier. * Librairie seedrandom : pour produire des comportements pseudo-aléatoires reproductibles. * Modélisation des entités : via les types Fighter, Card, Effect, pour encadrer les interactions. * Système de conditions / effets : permettant d’enrichir les cartes d’actions personnalisées. * Gestion des logs de combat : pour utiliser le résultat côté client. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | |  | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Développer une application sécurisée** | | | |
| ***Exemple n°1***  | | Contribuer à la gestion d'un projet informatique | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Pour le projet **Ploybout**, j’ai assuré la gestion du suivi des tâches et de l’organisation globale du projet.  J’ai défini les différentes catégories de travail, notamment les aspects visuels, la logique métier, ainsi que les normes à respecter comme le cahier des charges, les conditions générales d’utilisation et les obligations liées au RGPD.  Le suivi s’est fait en mode agile, avec des mises à jour régulières pour adapter les priorités et faire évoluer le planning. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé Notion comme outil principal de gestion de projet.  J’ai mis en place un tableau kanban structuré avec des colonnes représentant les états des tâches : « À faire », « En cours » et « Terminées », celles-ci classées par type, ce qui facilitait la visualisation des avancées et la répartition des priorités. [Annexe n°5]  Ce système m’a permis de garder une vision claire sur le planning et de piloter efficacement le déroulement du projet. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (ADRAR)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer une application sécurisée organisée en couches** | | | |
| ***Exemple n° 1***  | | Analyser les besoins et maquetter une application | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre du projet fil-rouge **Ploybout**, une application web ludique développée en *fullstack*, j’ai réalisé plusieurs livrables de conception afin d’analyser les besoins et préparer l’interface utilisateur.  J’ai d’abord identifié les fonctionnalités principales à représenter visuellement (combat, gestion de *deck* de carte, navigation entre les vues) en me basant sur des scénarios d’usage. J’ai ensuite produit un zoning, des *wireframes* puis des *mockups* pour représenter la structure de l’interface et anticiper l’expérience utilisateur.  Ces maquettes ont servi de patron de référence pour l’implémentation *front-end* en React. La représentation 3D des personnages n’ayant pas de support ergonomique dans Figma, j’ai dû faire des compromis visuels, en gardant une intention graphique cohérente tout en me concentrant sur la structure fonctionnelle. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| * Figma : pour la création des zonings, wireframes et mockups. * Inkscape : pour la création du logo. * Lospec : pour subdiviser des couleurs harmonieuses depuis la palette « APOLLO » de l’utilisateur AdamCYounis. * Game UI Database : pour m’inspirer du *layout* d’autres applications ludiques | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (ADRAR)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer une application sécurisée organisée en couches** | | | |
| ***Exemple n° 1***  | | Définir l'architecture logicielle d'une application | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Pour le projet **Ploybout**, j’ai défini une architecture logicielle organisée en couches distinctes.  La couche *back-end* est responsable de la gestion des données et de la logique métier, notamment pour gérer les combats asynchrones de manière sécurisée.  La couche *front-end*, quant à elle, se charge de l’affichage et de l’interaction utilisateur, en exploitant les données envoyées par le serveur pour représenter visuellement les combats et l’état du jeu. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai développé l’API en Node.js, où la logique métier est exécutée pour contrôler les règles du jeu et le déroulement des combats.  Pour le *front-end*, j’ai utilisé React, en structurant les composants par page et par fonctionnalité afin de faciliter la maintenance et la lisibilité du code.  Le front reçoit les données issues du serveur et les transforme en éléments visuels interactifs pour afficher les combats et l’état du jeu. Cette séparation claire entre gestion des données (côté serveur ) et présentation (côté client ) garantit la sécurité et une meilleure maintenabilité de l’application. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (ADRAR)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer une application sécurisée organisée en couches** | | | |
| ***Exemple n° 1***  | | Concevoir et mettre en place une base de donnée relationnelle | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre de la conception de mon projet personnel de type wargame sur navigateur, j’ai modélisé la base de données relationnelle à partir du modèle objet de l’application. Ce jeu comprend plusieurs entités interagissant entre elles :   * Les factions, qui possèdent des villes et des personnages. * Des personnages qui possèdent des caractéristiques. * Des villes qui possèdent des fiefs.   J’ai réalisé un diagramme de classes représentant les entités principales, leurs attributs ainsi que leurs relations (associations, compositions, agrégations). Ce travail a permis de définir une structure relationnelle cohérente, facilitant l’implémentation dans une logique métier ou un système de base de données relationnel. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé PlantUML pour formaliser ce diagramme de classes, détaillant les entités, leurs attributs et leurs relations avec les cardinalités.  Cette modélisation a servi de référence pour la conception de la base de données relationnelle et la mise en œuvre de la logique métier. [Annexe n°7] | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet personnel* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *03/04/2025*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer une application sécurisée organisée en couches** | | | |
| ***Exemple n° 1***  | | Développer des composants d’accès aux données SQL et NoSQL | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre d’un projet personnel de type *imageboard*, j’ai développé les composants d’accès aux données permettant de gérer les threads de discussion et leurs réponses.  Le but était de stocker les messages ainsi que les métadonnées des images associées, et de pouvoir y accéder ou les modifier.  Le projet devait permettre à plusieurs utilisateurs de publier et répondre à des sujets avec une certaine logique d'affichage chronologique (type *bump system*). | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé MongoDB, une base de données NoSQL orientée documents, pour stocker les messages et les réponses sous forme de collections. Pour interagir avec cette base, j’ai utilisé Mongoose, une bibliothèque Node.js qui facilite la modélisation des données et fournit une interface structurée pour effectuer des opérations CRUD. Les collections Thread et Reply ont été modélisées afin de représenter respectivement les discussions principales et les réponses associées. [Annexe n°8 ]  Les opérations CRUD sont centralisées dans un contrôleur Node.js, via des fonctions asynchrones. Ces fonctions :   * Valident les identifiants. * Traitent les fichiers image à l’aide de Sharp pour en extraire les métadonnées (largeur, hauteur) et optimiser leur format. * Appliquent une logique métier spécifique, comme la suppression automatique du thread le plus ancien lorsqu’un seuil est dépassé.   Les schémas de données intègrent également un identifiant formaté unique (pour l’UI ou les références) et une gestion des relations hiérarchiques entre les réponses. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (Odin Project)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *01/07/2023*au :*01/10/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **3** | **Préparer le déploiement d’une application sécurisée** | | | |
| ***Exemple n° 1***  | | Préparer et exécuter les plans de tests d'une application | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre d’un projet personnel de jeu de bataille navale contre l’ordinateur (**Battleship**), j’ai mis en place une série de tests unitaires dès le début du développement, en suivant une approche TDD (*Test Driven Development*).  Ces tests visaient à valider la logique de placement des navires sur la grille de jeu, en s’assurant qu’ils soient bien placés de façon cohérente et sans erreur, selon leur taille et l’espace disponible. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé le framework Jest pour rédiger et exécuter les tests unitaires.  Chaque scénario vérifiait que la fonction de placement attribuait correctement les cases aux navires, en respectant la continuité sur la grille et en tenant compte des contraintes du jeu. [Annexe n°9]  Cette méthode m’a permis de m’assurer de la fiabilité du cœur de la mécanique avant d’avancer sur les autres fonctionnalités. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Project d’étude (Odin Project)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *01/02/2023*au :*01/03/2024* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **3** | **Préparer le déploiement d’une application sécurisée** | | | |
| ***Exemple n° 1***  | | Préparer et documenter le déploiement d'une application | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre d’un projet d’étude consistant à développer une version numérique interactive d’un livre de type "Le livre dont vous êtes le héros", j’ai mis en place un système de migration avec Entity Framework Core (EF Core) afin de faciliter la gestion de l’évolution du schéma de la base de données.  Ce système permettait notamment de revenir à un état antérieur de la base de données en cas de problème survenu lors d’un déploiement, en exécutant une commande de mise à jour vers une migration précédente. Cette approche s’inscrivait dans une logique de fiabilisation du déploiement, en intégrant une stratégie de *rollback* simple et rapide à appliquer. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé l’interface en ligne de commande de .NET et les outils d’Entity Framework Core pour créer et appliquer les migrations :    dotnet ef migrations add, dotnet ef database update    Chaque modification du modèle de données était versionnée et documentée. En cas d’anomalie, un retour à une migration stable pouvait être effectué via la commande :    dotnet ef database update NomDeLaMigration    Cette méthode garantit une traçabilité des évolutions du schéma de données et permet de sécuriser le déploiement en production ou en environnement de test. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *07/04/2025*au :*25/04/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **3** | **Préparer le déploiement d’une application sécurisée** | | | |
| ***Exemple n° 1***  | | Contribuer à la mise en production dans une démarche Devops | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Dans le cadre du projet fil rouge **Ploybout**, projet *fullstack* comportant un *frontend* (React) et un *backend* (Node.js), j’ai mis en place un processus d’intégration et de déploiement continus (CI/CD) à l’aide de GitHub Actions. Ce pipeline est déclenché automatiquement à chaque *push* ou *pull request* sur la branche *main*.  J’ai configuré des *jobs* pour installer les dépendances du côté client et du côté serveur, lancer les serveurs en environnement de test, attendre qu’ils soient opérationnels, puis exécuter les tests *end-to-end* avec Cypress. En cas d’échec, les captures d’écran et vidéos sont automatiquement sauvegardées pour analyse.  Cette démarche permet de valider rapidement les modifications et d’assurer la stabilité de l’application avant toute mise en production. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| Voir l’[Annexe n°10] pour le fichier .yaml de configuration.  Outils utilisés :   * GitHub Actions : pour automatiser les étapes de CI/CD. * Cypress : pour les tests de bout en bout (end-to-end) simulant l’interaction utilisateur. * wait-on : pour synchroniser les jobs et s'assurer que les serveurs sont bien lancés. * Actions secrets : pour sécuriser les variables d’environnement sensibles (JWT, DB, etc.). * Artifact upload : pour archiver automatiquement les logs visuels en cas d’échec. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| En autonomie | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Projet d’étude (ADRAR)* | |
| Chantier, atelier, service | | | Service | | |
| Période d’exerciceDu : *16/09/2024*au :*05/06/2025* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

Cliquez ici pour taper du texte.

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.

Cliquez ici pour taper du texte.

Cliquez ici pour choisir une date

Fait à le

pour faire valoir ce que de droit.

Signature :

|  |
| --- |
| **Documents illustrant la pratique professionnelle** |
|  |
| *(facultatif)* |
| **Intitulé** |
| Annexe n°1 : Dépendances côté client |
| Annexe n°2 : Dépendances côté serveur |
| Annexe n°3 : Interface utilisateur avec Tailwind (wargame) |
| Annexe n°4 : Code du système de combat (Ploybout) |
| Annexe n°5 : Tableau kanban (Ploybout) |
| Annexe n°6 : Maquettes (Ploybout) |
| Annexe n°7 : Diagramme de classe (wargame) |
| Annexe n°8 : Utilisation d'une base de données non relationnelle (Imageboard) |
| Annexe n°9 : Tests unitaires (Battleship) |
| Annexe n°10 : Configuration CI/CD |

|  |
| --- |
| **Annexes** |
| *(Si le RC le prévoit)* Annexe n°1 : Dépendances côté client (Ploybout)Annexe n°2 : Dépendances côté serveur (Ploybout)Annexe n°3 : Interface utilisateur avec Tailwind (wargame) md :text-xl, sm :text-xs : Changent la taille de la police en fonction de la taille de l’écran.  menu-button : Classe classique CSS afin d’éviter d’alourdir le *template* Angular. Annexe n°4 : Code du système de combat (Ploybout) 14 : Importation de seedrandom (génération aléatoire contrôlée)  16 : Fonction principale  21-22 : Réception des combattants depuis le client  23-43 Initialisation des règles (30 : sécurité en cas de mauvaises données reçues)  45 : Initialisation de l’état qui va nous permettre de suivre l’avancement du combat (barres de vie, énergies accumulées, distances d’attaque, position des joueurs  55 : Début de la boucle de jeu, avec sécurité en cas de duel infini  60 - Génération de la *seed* pour gérer l’aléatoire (reproductibilité)  62-74 : Détermination du premier et tri des cartes.  78 : Cherche une carte valide  79-88 : si une carte est trouvée, appliquer ses effets  89-105 : Ajout du tour dans le répertoire (log)  106 : Si l’un des personnage a moins de vie, arrêter la boucle  111 : On ajoute un tour  114-121 : On envoie le résultat la fonction qui enverra au client le résultat du combat  Autres fonctions déterminant le déroulement des combats : Annexe n°5 : Tableau kanban (Ploybout)Annexe n°6 : Maquettes (Ploybout)Annexe n°7 : Diagramme de classe (wargame)Annexe n°8 : Utilisation d'une base de données non relationnelle (Imageboard)Annexe n°9 : Tests unitaires (Battleship)Annexe n°10 : Configuration CI/CD |