|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | RAHMANI |
| *Nom d’usage* |  | *Entrez votre nom d’usage ici.* |
| *Prénom* |  | Mohammad |
| *Adresse* |  | 31 Boulevard Saint Michel 72000 Le Mans |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| *Concepteur développeur d’applications* | |
|  | |
| **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 utilisateur en intégrant les recommandations de sécurité** | | **p.** | **5** |
|  |  Réalisation de l’application MDSTOCK  p. | p. | 5 |
|  |  |  |  |
| **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité** | | **p.** | **8** |
|  |  Réalisation de l’application MDSTOCK  p. | p. | 8 |
|  |  |  |  |
| **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | **p.** | **10** |
|  |  Réalisation de l’application MDSTOCK  p. | p. | 10 |
|  |  |  |  |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | **p.** | 13 |
| **Déclaration sur l’honneur** | | **p.** | 14 |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | **p.** | 15 |
| **Annexes** *(Si le RC le prévoit)* | | **p.** | 16 |

**Exemples de pratique**

**professionnelle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | | |
| ***Exemple n°1***  | | ***MDSTOCK*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| 1. Maquetter une application    * **Charte graphique :** « **coolors.co »** pour choisir les couleurs, **« fonts.google.com »** pour les polices et **« Figma »** pour la création de la charte graphique.    * **Wireframes et maquettes haute-fidélité** (sur différents formats : mobile et bureau) **:  « diagrams.net »** pour créer des wireframes et **« Figma »** pour créer des maquettes haute-fidélité.    * **Diagrammes : « diagrams.net »** pour créer des diagrammes (cas d’utilisation et activités) et «**MySQL Workbench »** pour créer le diagramme de classe. 2. Développer une interface utilisateur pour multiplateformes   J’ai utilisé **« ElectronJS »** et **« Cordova »** pour créer des applications multiplateformes. ElectronJS est un framework qui m’a permis de créer mon application pour les environnements de bureau tels que Linux et Windows. Pour concevoir la version mobile, j’ai utilisé Cordova qui créé des applications mobiles en utilisant des fichiers **HTML**, **CSS**, **JS**, etc.. Pour créer l’application mobile, j’ai dû prendre un **« build »** de la partie « front » de mon projet qui m’a fourni des fichiers pour que Cordova puisse la créer. Grâce à l’encapsulation générée par ElectronJS qui virtualise un « browser », j’ai créé mon application pour la version bureau.   1. Développer des composants d’accès aux données  * **Front-End** : Grâce à « **Axios** » qui est un HTTP Client, le Front de mon application peut communiquer avec le Back pour accéder aux données. Dans mon projet ReactJS, j’ai pu gérer les données reçues par le Back-End dans le Front-End grâce à l’aide de la bibliothèque Redux et de l’outil Redux Toolkit * **Back-End :** Grâce au framework **« ExpressJS »**, j’ai créé un serveur et **l’API**. J'ai utilisé le module **« MySQL »** pour que mon serveur puisse se connecter à la base de données.  Sur mon serveur, j’ai créé plusieurs routes pour accéder aux données sauvegardées dans la base de données. Pour réaliser le CRUD de chaque objet, j’ai utilisé les différentes méthodes HTTP citées ci-dessous :   + **POST :** Pour envoyer des données   + **GET :** Pour récupérer des données   + **PUT :** Pour modifier des données   + **DELETE :** Pour supprimer des données  1. Développer la partie Front-End d’une interface utilisateur web   La partie Front-End de mon application a été développé avec ReactJS (librairie JavaScript) que de nombreux développeurs du monde entier utilisent, en raison des fonctionnalités qu’elle offre. Avec l’aide de ReactJS, j’ai pu créer des interfaces utilisateur interactives.  J’ai utilisé du CSS pur pour styliser les composants et les pages, afin que je puisse avoir le contrôle total sur les styles des éléments dans les différents modes.  Pour une meilleure gestion des différentes parties et éléments du site, je les ai placés dans les différentes directions :   * app : Store (Redux), importation des différents **reducers** (pour gérer des states), les slices (collection de reducer et actions pour une fonctionnalité), **services** (pour envoyer des requêtes au serveur) * assets : fichiers statiques (images, icônes), CSS global * components : modulation des éléments (forms, tables, buttons, inputs) * uns : Main, User, Admin * pages : différentes pages séparées par rôles de notre projet  1. Développer la partie Back-End d’une interface utilisateur web   Le **Back-End** de mon application a été développé avec la méthode de programmation POO / OOP (Programmation Orienté Objet). J’ai aussi utilisé NodeJS (un outil qui permet d’exécuter JavaScript à l’extérieur des navigateurs) et le framework ExpressJS. Pendant ma formation développeur web et web mobile, j’avais utilisé cet outil et ses packages et j’ai donc pu les utiliser pour concevoir le Back-End du projet. Comme évoqué précédemment, j’ai réalisé le serveur et l’API avec le framework ExpressJS. J’ai aussi utilisé d’autres modules comme : bcrypt, dotenv, mysql, cors, helmet, nodemailer, sharp, jsonwebtoken, multer … | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| * UML : Figma, app.diagrams.net, mysql Workbench * Front : ReactJS, Redux (Redux toolkit), react-router-dom * Back : Nodejs (ExpressJS) * Modules : bcrypt, nodemailer, multer, sharp, dotenv * Outils : VScode, Git, GitHub, NPM | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| L’application a été réalisé en binôme avec l’action d’une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les envois d’images, la mise en production, etc. … | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *MDSTOCK* | |
| Chantier, atelier, service | | | APP | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité** | | | |
| ***Exemple n° 1***  | | ***MDSTOCK*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| 1. Concevoir une base de données   La gestion de base de données **« SGBD »** utilisée est **MySQL**. Après avoir créé les **MCD** et **MLD** avec le logiciel **« Looping »**, j’ai obtenu un script qui m’a permis de créer ma base de données. J’ai également utilisé le logiciel MySQL Workbench pour concevoir le diagramme de classe et gérer ma base de données dans un environnement graphique pendant le développement. Dans ma base de données, les tables ont des relations avec d’autres tables. Le Back-End (serveur) peut communiquer avec la base de données grâce un module **« NPM »** qu’il s’appelle **« mysql »**.   1. Mettre en place une base de données   Dans l’environnement Linux, j’ai d’abord installé MySQL à l’aide de la commande « shell » et mit son service en mode **« enable »**. En utilisant le script que le logiciel **« looping »** m’a donné et le logiciel Mysql Workbench, j’ai créé la base de données du projet.  Ensuite via la console et le compte **root**, j’ai accédé à MySQL et créé un autre compte utilisateur pour avoir uniquement l’accès à cette base de données. Ensuite j’ai donné les droits « **GRANT ALL PRIVILEGES »**, pour cet utilisateur pour qu’il puisse gérer la base de données du projet.  Comme évoqué ci-dessus, cette base de données sera mise en liaison avec mon application (Back-End) grâce au module NPM « Mysql ».  Pour la sécurité des informations de connexion à la base de données, j’ai utilisé les variables d’environnement afin que l’accès à ces informations dans les codes ne soit pas possible.  J’ai aussi créé un script qui sera indépendant pour faire des sauvegardes (backup) régulières de la base de données.  Grâce à Docker je lance la base de données dans un container avec un port différent sur mon VPS que je suis seul à connaître.   1. Développer des composants dans le langage d’une base de données   Pour mon application, j’ai créé et développé un trigger (appelé aussi déclencheur qui permet d’exécuter un ensemble d’instructions SQL juste après un évènement, ce qui permet de faciliter et d’automatiser des actions au sein d’une SGBD) et un event avec MySQL.  L’event est une fonction qui va s'exécuter à intervalle régulier au sein de la BDD. Dans mon cas, elle va vérifier si les tokens de réinitialisation ont plus de quinze minutes et si c’est le cas, elle va les supprimer.  Pour le trigger, il se déclenche avant ou après un INSERT, UPDATE, DELETE. Dans mon cas, il va se lancer après un UPDATE de mot de passe et supprimer le token de réinitialisation de mot de passe. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| * Outils : Système de gestion de base de données MySQL, MySQL Workbench, docker (sur mon VPS), Looping, pm2 * Script : NodeJS * Modules : mysql, node-cron | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| L’application a été réalisé en binôme avec l’action d’une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les envois d’images, la mise en production, etc. … | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *MDSTOCK* | |
| Chantier, atelier, service | | | APP | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **3** | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | |
| ***Exemple n° 1***  | | ***MDSTOCK*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| 1. Collaborer à la gestion d’un projet informatique et à l’organisation de l’environnement de développement   Pour réaliser ce projet, j’ai travaillé en binôme. Pour pouvoir s’organiser et communiquer efficacement, nous avons utilisés plusieurs outils :   * GitHub (stocker et partager les codes du projet) * Trello (organiser et suivre les étapes clés du projet) * Discord (communiquer avec le binôme) * Figma, app.diagrams.net, Google drive, Word, Google Docs  1. Concevoir une application   L’application a été réalisée à partir du cahier des charges et des :   * Diagrammes de classe, MCD, MLD : pour la modélisation de la base de données * Diagrammes de cas utilisation : pour connaître les rôles utilisateurs et leurs fonctionnalités. * Diagrammes activités : pour décrire chaque fonctionnalité (des diagrammes de cas utilisation) du début à la fin. * Workflow : référence à l’ensemble d’activités nécessaires pour accomplir des tâches. * Sitemap : pour connaître les routes de notre projet * Des wireframes et des maquettes haute-fidélités  1. Développer des composants métier   Pour une plus grande modularité et maintenabilité du code de chaque “entité” de notre projet à son propre routeur, contrôleur et modèle. Ces derniers sont ce que l’on peut appeler les composants métier de notre application. Lorsqu’une requête à l’API est faite, elle passe par la route de l’entité qu'elle souhaite faire appel, passe ensuite par le contrôleur qui traite les données en entrée puis, pour finir, fait appel au modèle qui est en relation avec la base de données.   1. Construire une application organisée en couches   Mon application est une application N-tiers avec une programmation orienté objet (OOP ou POO).  L’application est basée sur trois couches principales :   * Client (Front) : application multiplateforme réalisé en ReactJS * Logic (Back) : API principale en NodeJS (ExpressJS) * Data (DB) : MySQL  1. Développer une application mobile   Grâce à l’outil « Cordova », j’ai conçu mon application mobile. Pour pouvoir le faire, j’ai réalisé les étapes suivantes :   * Installer « Cordova » et « Android Studio » ; * Créé un projet avec « Cordova » ; * Pris un « build » de mon projet ReactJS ; * Copié les fichiers fournis par le build dans le dossier « www » ; * Utilisé les commandes de « Cordova » pour créer l’application mobile.  1. Préparer et exécuter les plans de tests d’une application   Pour les tests **Back-End**,j’ai utilisé plusieurs modules NPM :   * **Mocha** – **assert** : pour tester la conception de mes models en entrée/sortie fictive * **Chai** – **Chai-http** – **Supper-test** : pour tester les routes de mon API   J'ai utilisé **Postman** pour tester mes routes tout au long du développement. J’ai aussi utilisé **Swagger** pour concevoir une documentation API pour mes routes Méta.  Pour le **Front-End** j’ai utilisé l’extension **lighthouse** pour optimiser mes codes.   1. Préparer et exécuter le déploiement d’une application   L’application est déployée avec la plateforme docker sur mon VPS (Linux Ubuntu). Pour pouvoir le faire, j’ai créé 3 containers pour exécuter chaque partie de mon application :   * mysqldb : pour déployer la base de données ; * Back : pour déployer mon serveur (Back-End) ; * Front : pour déployer le Front de mon application dans un container avec le serveur web Nginx.   J’ai aussi utilisé le serveur web Nginx comme proxy pour rediriger les requêtes. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| * Outils: UML : Figma, app.diagrams.net, mysql Workbench * Front: ReactJS, Redux (Redux toolkit), react-router-dom * Back : Nodejs (ExpressJS) * Modules : mocha, chai, chai-http, Super-test * Outils : VScode, Git, GitHub, NPM, Système de gestion de base de données MySQL, MySQL Workbench, docker (sur mon VPS), Looping, pm2, Android Studio, Cordova, nginx, nushell, bash | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| L’application a été réalisé en binôme avec l’action d’une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les envois d’images, la mise en production, etc. … | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *MDSTOCK* | |
| Chantier, atelier, service | | | APP | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| **DEVELOPPEUR WEB ET WEB MOBILE** | **ARINFO** | 2022 |
| **CDA** | **ARINFO** | N/A |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

Je soussigné(e) [prénom et nom] RAHMANI Mohammad ,

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

15/10/2022

Fait à LE MANS, 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)* |