

## ▀ ▀ ▀ Modèle de copie :

**Évaluation en cours de formation**

**Développeur Web et Web Mobile**

**Ceci est un modèle de copie. N’oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom : Zakaria**

**Nom : Chegdali**

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : ECF-Cold-Room

Lien Github du projet : https://github.com/youamzak/ecf-cold-room.git

Lien Drive du projet (si nécessaire) : ………

**Attention ! Merci de bien classer vos documents dans votre Github ou votre drive.**

URL du site (si vous avez mis votre projet en ligne) : https://ecf-cold-room.herokuapp.com/

**Description du projet**

1. Liste des compétences du référentiel qui sont couvertes par le projet

Maquetter une application

Développer une interface web utilisateur web dynamique

Réaliser une interface utilisateur avec une solution de gestion de contenu

Créer une base de données

Développer les composants d’accès aux données

Développer la partie back-end d’une application web

1. Résumé du projet en français d’une longueur d’environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

Le but du projet est de créer une application pour la société RD Temp qui est spécialisée dans la réfrigération.

Cette application aura la charge d’afficher des courbes de températures et d’hygrométrie provenant d’une chambre froide qui est installée dans une officine.

Afin d’alimenter la base de données, un technicien transmettra les données, sous un format csv, quotidiennement via l’application.

Le client (officine) pourra se connecter de façon sécurisée pour consulter les courbes et valider une journée, ce qui signifie que les données sont conformes aux attentes.

La société RD Temp, via le responsable de l’application, de son côté pourra gérer les utilisateurs (administrateur, technicien et officine) ainsi que la création d’officines et de chambres froides.

1. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

|  |  |
| --- | --- |
| **User storie** | **Description** |
| US1- Créer le compte officine | *Utilisateurs concernés : administrateur*   * L’administrateur sera un responsable de l’application chez RD Temp. Lui seul pourra créer le compte d’une officine. * Les informations nécessaires seront : le nom, l’adresse postale, la ville, le numéro de téléphone. * Un identifiant unique (ex: Pharma-62-1) et un mot de passe sécurisé résistant aux injections SQL devront être automatiquement générés, afin que l’officine puisse s’authentifier |
| US2- Gérer les techniciens | *Utilisateurs concernés: administrateur*  Les techniciens sont les employés de RD Temp qui s’occupent de la surveillance des chambres froides. Chacun d’eux devra avoir un identifiant unique et un mot de passe sécurisé. Si besoin, le technicien pourra changer son propre mot de passe ou un administrateur peut le faire à sa place. |
| US3- Se connecter | *Utilisateurs concernés: administrateur, officine, technicien* |
| US4- Gérer les chambres froides et ses sondes | *Utilisateurs concernés: administrateur*   * Une officine peut être détentrice de plusieurs chambres froides. * Dans une chambre froide, nous trouverons toujours 2 sondes : une de type “température” qui aura des données en degrés, une de type “hygrométrie” qui aura des données en pourcentage. |
| US5- Importer les données de sonde | *Utilisateurs concernés: technicien*   * Quotidiennement, un technicien chargé de la surveillance de la chambre froide transmettra, pour chaque sonde associée, le fichier de données qu’il aura récupéré. Celui-ci devra être uploadé dans un format csv (exemple en annexe). * Automatiquement, toutes les données comprises dans le fichier seront enregistrées dans la BDD. |
| US6- Afficher les courbes de température et hygrométrie | *Utilisateurs concernés: officine*   * Lorsqu’elle se connecte, une officine découvre les chambres froides qu’elles possèdent. * Lorsqu’elle en choisit une, l’officine peut alors sélectionner une date. * Sans rechargement de la page, les données de la date en question seront représentées sous la forme de 2 courbes :   La température (les données seront toujours comprises entre 0 et 10 degrés) qui récupère la sonde de type “température” associée à cette chambre froide.  L’hygrométrie (Entre 0 et 100%) qui récupère la sonde de type “hygrométrie” associée à cette chambre froide. |
| US7- Valider une journée | *Utilisateurs concernés: officine*  Après un examen des courbes du jour, l’officine peut décider de “valider” la journée. Cela signifie pour elle que les données sont conformes aux résultats qu’elle attendait. Un commentaire peut être donné lors de la validation. |

1. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Back :

* Node JS
* Express JS
* MongoDB

Front :

* React JS
* React Bootstrap
* CSS Module + Styled COmponent

Serveur :

* Mongo DB

1. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Afin de m’aider dans la mise en place des sécurités, j’ai suivi les recommandations proposées par le site [ExpressJS](https://expressjs.com/fr/advanced/best-practice-security.html).

Voici la liste des éléments mis en place côté serveur :

* Bcrypt pour le cryptage des mots de passe
* Express-joi-validation + JOI pour la validation des requêtes
* Helmet (voir documentation)
* JsonWebToken pour la gestion de l’authentification de l’utilisateur
* XSS pour la gestion des attaques XSS

1. Description d’une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

La plus grosse difficulté que j’ai rencontré en terme technique pour la construction de l’application a été la gestion des globals states.

La ressource principale que j’ai utilisé est le site officielle de Redux Toolkit (https://redux-toolkit.js.org/tutorials/overview)

1. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

*Page reference : https://redux-toolkit.js.org/usage/usage-guide*

*The Redux core library is deliberately unopinionated. It lets you decide how you want to handle everything, like store setup, what your state contains, and how you want to build your reducers.*

*This is good in some cases, because it gives you flexibility, but that flexibility isn't always needed. Sometimes we just want the simplest possible way to get started, with some good default behavior out of the box. Or, maybe you're writing a larger application and finding yourself writing some similar code, and you'd like to cut down on how much of that code you have to write by hand.*

*As described in the Quick Start page, the goal of Redux Toolkit is to help simplify common Redux use cases. It is not intended to be a complete solution for everything you might want to do with Redux, but it should make a lot of the Redux-related code you need to write a lot simpler).*

Le noyau de la librairie Redux est délibérément agnostique. Il vous laisse le choix de tout gérer, comme le réglage du store, ce que les states contiennent et comment construire vos reducers.

Dans la plupart des cas cela fonctionne, car il est flexible, mais ceci n’est pas toujours nécessaire. Parfois, on souhaite la façon la plus simple de démarrer, avec un comportement par défaut déjà prêt. Ou, peut-être vous écrivez une plus grosse application et vous remarquez que vous écrivez souvent le même code, et vous souhaiteriez réduire la quantité de code à écrire à la main.

Comme décrit dans la section Démarrage Rapide, le but de Redux Toolkit est d’aider à simplifier l’usage de Redux. Ce n’est pas une tentative de répondre à tous les problèmes que vous rencontrez avec Redux, mais ça devrait simplifier de nombreux bout de code reliés à Redux .

1. Autres ressources

Sites internet

MDN Web Docs

STUDI

Stackoverflow

ReactJS

React Bootstrap

NPM

Youtube

1. Informations complémentaires

Sur cette application j’ai rencontré une difficulté liée au temps imparti pour sa réalisation.

Effectivement, étant en emploi il m’a été très difficile d’organiser mon temps de travail.

1. Informations d’identifications + autres

Connexion à la base de données via MongoDb Compass

**mongodb+srv://mongo:n7SxLVWZVEnQmoD4@cluster0.qspei.mongodb.net/test?authSource=admin&replicaSet=atlas-a3c6p6-shard-0&readPreference=primary&appname=MongoDB%20Compass&ssl=true**

Compte pré-enregistré sur l’application

Login pour administrateur : **admin** / psw : **01234567**

Login pour technicien : **tech** / psw : **01234567**

Login pour officine : **officine1** / psw : **01234567**

Un fichier csv est disponible sur le repo dans le dossier **data\_sample**

Pour visualiser les graphes, il faut sélectionner la date du 18/10/21 (date du premier enregistrement dans le csv)