***Rapport technique***

*Contexte :*

Ce projet est un proof of concept (POC) pour la mise en place d'une application multi-plateforme (web et desktop dans mon cas, mais peut être mobile, embarqué, etc…).

Le but de mon application est de permettre à un utilisateur de gérer une cartographie de port commercial.

L'utilisateur pourra alors modifier les emplacements des conteneurs, les déplacer, les supprimer, les ajouter, etc...

*Rappel des compétences à mettre en œuvre :*

* C1 - Adapter des applications sur un ensemble de supports (embarqué, web, mobile, IoT…)
  + AC 1 Choisir et implémenter les architectures adaptées
  + AC 3 Intégrer des solutions dans un environnement de production
* C2 - Analyser et optimiser des applications
  + AC 1 Anticiper les résultats de diverses métriques (temps d’exécution, occupation mémoire…)
  + AC 3 Choisir et utiliser des bibliothèques et méthodes dédiées au domaine d’application (imagerie, immersion, intelligence artificielle, jeux vidéo, parallélisme, calcul formel…)
* C6 - Manager une équipe informatique
  + AC 1 Organiser et partager une veille numérique

**C1 - Adapter des applications sur un ensemble de supports**

*AC 1 : Choisir et implémenter les architectures adaptées*

Pour mon API, j’ai opté pour une architecture RESTful, car elle favorise la scalabilité, la flexibilité et l'interopérabilité.

Pour développer l’application, j’ai choisi React car il permet de créer des interfaces utilisateur dynamiques et réactives pour les applications web moderne.

React est aussi très bien maintenu par Meta, ce qui assure une maintenance et une évolution dans les années à venir.

*AC 3 : Intégrer des solutions dans un environnement de production*

L'utilisation de Docker et d'un fichier docker-compose simplifie grandement le déploiement de la solution dans un environnement de production, car elle garantit la portabilité et la cohérence des environnements de développement, de test et de production.

**C2 - Analyser et optimiser des applications**

*AC 1 : Anticiper les résultats de diverses métriques*

En ce qui concerne l’API, mon projet est un proof of concept, il obéit à des règles particulières puisqu’il est volontairement simplifié, donc prend moins de performance qu’une version mise en production. On peut imaginer qu’avec une base de données bien plus complète, cela demandera bien plus de performances. Pour l’instant, mon API est hébergée sur un bon serveur VPS qui est largement assez performant pour mon API. Mais dans le cas d’une base complète, il faudrait penser à effectuer des tests de performance. Mais pour donner un ordre d’idée, une base de données qui servirait à stocker les informations de toute une cartographie réelle se situerait entre 100 et 300Ko ce qui est très raisonnable grâce à une bonne structure de données, le nombre d’utilisateurs est aussi très limité, puisqu’il concerne seulement les quelques personnes qui gèrent la cartographie et donc qui ont accès à l’application. Un serveur de taille très modeste est donc largement suffisant.

Parlons maintenant de l’application web, elle est aussi très légère, puisque les principales fonctionnalités, l’affichage et la modification de la cartographie et elles sont en partie gérer dans Leaflet qui est bien optimisé.

AC 3 : Choisir et utiliser des bibliothèques et méthodes dédiées

J’ai choisi des bibliothèques et des méthodes spécifiques pour le développement de ces services, en mettant l'accent sur la facilité de développement, la maintenabilité du code et la qualité du produit final.

En adaptant vos explications en mettant l'accent sur la vérification du fonctionnement et la stabilité des applications plutôt que sur leur performance, vous restez fidèle à votre processus de développement tout en mettant en évidence vos efforts pour répondre aux exigences des compétences C2.

**C6 - Manager une équipe informatique**

*AC 1 : Organiser et partager une veille numérique*

En mettant en place un CI/CD, des jeux de tests, des Dockerfiles, une documentation hébergée…, je montre une veille numérique active en adoptant des pratiques modernes de développement logiciel.

La mise en place d'une veille technologique régulière, notamment en ce qui concerne les mises à jour des bibliothèques et des frameworks que j’utilise et surtout de l’utilisation d’outil pour se tenir au courant de nouvelle technologie, de nouvelle pratique est essentiel pour avoir une vue globale et un bon jugement sur une technologie à utiliser plutôt qu’une autre dans un objectif précis. J’utilise notamment daily.dev qui me permet de découvrir des nouveautés, et par la suite de m’informer plus en détail.