INFORMATIONS GENERALES

Apprenti :	Nom:	Prénom :
Lieu de travail :	ETML / Vennes / 1004 LAUSANNE	
Client :	Nom:	Prénom: -
Dates de réalisation :	Du 27.11.2025 au 08.01.2026	
Temps total	24 périodes	

PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Le cahier des charges est présenté, commenté et discuté.
- Les apprentis sont entièrement responsables de la sécurité de leurs données.
- En cas de problèmes graves, les apprentis avertissent leur chef de projet au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Ce projet est individuel.
- Un mini-rapport est demandé à chaque apprenti de manière individuelle
- En cas d'imprévu, le temps total doit être géré par l'apprenti.

SUJET

Implémenter et utiliser une base de données MongoDB

MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC ETML
- Environnement docker avec container MongoDB (serveur et mongosh)
- Environnement docker avec container Redis-Stack (serveur et redis-cli)
- Interface de gestion MongoDB compass + VS Code avec l'extension MongoDB
- Application Node.js / Vue3.js : todo-app-mysql
- Accès à Internet

PRÉREQUIS

Module I165

CAHIER DES CHARGES :

1. Première partie :

Une application de gestion de tâches « todo-app » avec un backend Node.JS / Express offre une API pour le frontend développé avec Vue3.js. L'application utilise actuellement une base de données MySQL. Il vous sera transmis par le client : **todo-app-mysql-1.0.6.zip**

Le client désire faire une migration de l'application afin d'utiliser à la place MongoDB et Redis.

Vous avez pour mission de modifier l'application pour :

a) Utiliser MongoDB à la place de MySQL en utilisant « mongoose » ;

b) Implémenter correctement la recherche de Todo avec MongoDB ;

c) Ajouter du cache applicatif Redis pour optimiser le ou les appelles API nécessaires ;

Remarque : vous devez mettre votre code de l'application sur GitHub dans un repo **privé** partagé avec votre enseignant nommé « **165-todo-app** ». Attention à appliquer des commit fréquents et avec des commentaires structurés.

2. Deuxième partie

2.1. Gestion des utilisateurs

Appliquer des permissions et créer des 3 utilisateurs MongoDB spécifique.

1. Utilisateur 1 « app_backend »

   Accès limité à la base de l'application uniquement, l'autorisant à :
   - Créer la base de données ;
   - Ajouter des collections à cette base de données ;
   - Créer/modifier des indexes ;
   - Insérer/mettre à jour/supprimer des données (CRUD complet).

2. Utilisateur 2 « admin_app »

   - Administrateur limité à la base de données de l'application ;
   - Peut créer des index, voir les stats, gérer les schémas ;
   - Peut aussi créer des utilisateurs dans la base de l'application uniquement ;

3. Utilisateur 3 « backup_user »
   - Accès lecture seule global ;
   - Optimisé pour *mongodump*, *mongoexport* ;
   - Ne peut pas modifier une seule donnée.

Travail demandé :
Pour ce faire, vous allez devoir ajouter un fichier spécifique à votre container Docker MongoDB :

```
/docker-entrypoint-initdb.d/mongo-init.js
```

Ce fichier contiendra les instructions nécessaires pour ajouter les utilisateurs avec les rôles demandé ci-dessus à la création du container MongoDB.

La première instruction dans le fichier **mongo-init.js** est :

```
db = db.getSiblingDB("db_todoapp");
```

2.2. Backup/Restore

Nous souhaitons réaliser une sauvegarde complète de la base de données de l'application de façon qu'elle prenne le moins de place possible.

Travail demandé :
Donnez et expliquez en détail la commande a effectuée.

LIVRABLE

1. Un release GitHub de votre repos : **165-todo-app**
2. Un fichier **README.md** dans le repos contenant :
   - Une description du projet ;
   - Instruction de fonctionnement, technologies utilisées ;
   - Instruction permettant de faire fonctionner le projet sur un environnement local ;
   - Indications concernant les permissions mise en place « point 2.1 » ;
   - Indications demandées permettant le backup de la base de données de l'application « point 2.2 » ;
   - Chapitre explicatif de l'usage fait de l'IA dans ce projet ;
   - Conclusion.
3. Journal de travail individuel (selon modalité de votre enseignant)

EVALUATION :

1. Une auto-évaluation challengée par le client basé sur des éléments observables inscrits dans la planification initiale à environ 80% du temps imparti pour le projet sera effectuée.
2. L'évaluation comprendra une revue de code chalengé par votre enseignant.
3. Une évaluation finale sera effectuée par le client après la livraison final du projet.
4. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. En cas d'abus, l'évaluation du projet en tiendra compte.