
Mini-Projet

Gestion de Réseaux Sociaux et d'Intérêts Utilisateurs Neo4j & MongoDB

Contexte :

Dans ce projet, vous êtes chargés de développer une application backend permettant de gérer un réseau social qui intègre les interactions entre utilisateurs ainsi que la gestion des intérêts et préférences des utilisateurs. Le projet combine l'utilisation de deux bases de données NoSQL : Neo4j pour gérer les relations d'amitié entre les utilisateurs et MongoDB pour stocker des informations structurées sur les utilisateurs, tels que leurs profils et centres d'intérêt.

❖ Objectifs :

Développer des API pour gérer des profils d'utilisateurs, leurs connexions (amis), et leurs centres d'intérêt en utilisant **MongoDB** et **Neo4j**. Ce projet ne nécessite pas de développement frontal ; il se concentre uniquement sur la création des bases de données et des API.

Pas trop de détails dont l'objectif est de vous permettre de penser par vous-même et d'appliquer **votre propre logique** pour atteindre les objectifs du projet.

❖ Durée estimée : 12 heures

❖ Instructions

1. Travail en binôme :
 - Vous devez travailler en binôme. Chaque membre du groupe se concentrera sur une partie spécifique du projet : MongoDB ou Neo4j.
2. Deadline :
 - Le projet doit être remis sur GitLab avant **Mardi 10 septembre**.

3. Pas de développement frontal :

- Aucune interface utilisateur n'est requise. Concentrez-vous uniquement sur les API et les bases de données.

4. Instructions spécifiques :

- **N'hésitez pas à fournir un schéma explicatif** de votre démarche, architecture, et conception. Cela aidera à clarifier votre approche et à montrer comment vous avez structuré votre solution pour répondre .

❖ **Tâches et Répartition :**

1. Configuration de l'environnement (MongoDB, Neo4j et choix de langage pour développer les API)
2. Démarche et architecture
3. Conception de la Base de Données
4. Développement des API
 - API à créer MongoDB:
 - Ajouter un utilisateur
 - Rechercher un utilisateur par email ou centres d'intérêt
 - Mettre à jour les centres d'intérêt d'un utilisateur
 - API à créer Neo4j:
 - Ajouter une relation d'amitié
 - Rechercher les amis d'un utilisateur
 - Supprimer une relation d'amitié
5. API Requêtes :
 - Trouver les utilisateurs partageant des centres d'intérêt communs
 - Calculer le nombre d'utilisateurs par centre d'intérêt
 - Trouver les amis de N-ème (expl 3-ème niveau) niveau d'un utilisateur, ou N est variable.

Préparer un document expliquant :

1. La structure des bases de données :

- Décrire les schémas de base de données utilisés pour MongoDB et Neo4j.
- Expliquer les relations entre les différentes collections/documents (pour MongoDB) et les nœuds/relation (pour Neo4j).
- Inclure tout schéma conceptuel ou logique pertinent.

2. Les API disponibles et leur fonctionnement :

- Fournir une liste des API créées avec une brève description de chaque endpoint.
- Expliquer les méthodes HTTP utilisées (GET, POST, PUT, DELETE) et leur rôle dans chaque API.
- Décrire les formats de données attendus en entrée et les réponses en sortie (JSON, etc.).

3. Des exemples de requêtes et réponses :

- Inclure des exemples concrets de requêtes (ex. : curl) et leurs réponses correspondantes.

4. Un schéma explicatif de l'architecture et de la conception :

- Fournir un ou plusieurs schémas qui illustrent l'architecture globale du projet (ex. : flux de données, interactions entre les bases de données et les API).
- Inclure une description de la conception technique, mettant en évidence les choix architecturaux.

5. Conclusion

Résumé des acquis sur MongoDB et Neo4j :

- Résumer ce que vous avez appris en travaillant avec MongoDB et Neo4j, notamment les différences entre les deux systèmes de gestion de bases de données, et leurs cas d'utilisation respectifs.

Défis rencontrés et solutions mises en place :

- Décrire les principaux défis techniques ou conceptuels que vous avez rencontrés lors de la réalisation du projet.
- Expliquer les solutions que vous avez mises en place pour surmonter ces défis, y compris toute modification de conception ou d'implémentation

Livrables à Remettre sur GitLab

Les éléments suivants doivent être inclus dans votre dépôt GitLab (un seul par binôme) :

1. Code source :

- Tous les fichiers de code source utilisés pour implémenter les API
- Fichiers de configuration pour MongoDB et Neo4j, s'ils sont nécessaires pour exécuter les projets.

2. Fichiers de documentation :

- Un **seul** document au format PDF ou Markdown qui inclut toute la documentation mentionnée ci-dessus.
- Le fichier README du dépôt doit résumer brièvement le projet et fournir des instructions sur la façon de cloner le dépôt, installer les dépendances, et exécuter les API.

3. Exemples de requêtes :

- Un fichier texte ou script contenant des exemples de requêtes pour tester les API.

Tous les livrables doivent être soumis sur GitLab avant **Mardi 10 septembre**. Assurez-vous que votre dépôt est bien structuré et que tous les éléments requis sont clairement identifiés.

Bon travail