Rapport : Application web avec la stack (MongoDB, Express, Angular, NodeJS)

Nom :
Prenom :

I. Introduction :

1.1 Présentation du Projet :

Ce projet implique le développement d'une application web utilisant Angular pour le frontend, Node.js pour le backend et MongoDB Atlas comme base de données. L'application est destinée au personnel universitaire et vise à faciliter la gestion des matériaux de l'université. Elle permet à tout membre du personnel de demander du matériel via l'interface du projet, et l'administrateur peut ensuite accepter ou refuser cette demande. Après l'acquisition, le membre du personnel peut retourner l'objet demandé après utilisation.

1.2 Objectifs :

- Développer une interface utilisateur conviviale en utilisant Angular.
- Implémenter un backend sécurisé en utilisant Node.js.
- Intégrer MongoDB Atlas pour la gestion des bases de données.
- Assurer un contrôle d'accès basé sur les rôles pour les utilisateurs et les administrateurs.

- Fournir des mises à jour en temps réel et des notifications pour les changements de statut des objets.

II. Installation :

1. Prérequis
   - Node.js et npm (Node Package Manager)
   - MongoDB (la base de données)
   - Angular CLI
   - Compte MongoDB Atlas

2. Installation des Prérequis
   - Téléchargez et installez Node.js depuis nodejs.org.
   - Téléchargez et installez MongoDB depuis mongodb.com.
   - Installez Angular CLI globalement en utilisant npm :
     - `npm install -g @angular/cli`

3. Installation des modules

   Partie backend :

   1. Ouvrez une fenêtre terminal.
   2. *cd backend*
   3. *npm install*

   Partie frontend :

   1. Ouvrez une fenêtre terminal.
   2. *cd backend*
   3. *npm install*

4. Exécution du Projet

```
cd backend
npm run dev

cd frontend
ng serve.

mongosh "mongodb+srv://app.g22pwha.mongodb.net/" --apiVersion 1 --username
jd225025 --password 5ZIcHg1FwrMbkNNR
```

III. Conception :

L'application est basée sur une architecture à trois couches :

- Frontend : Angular, Html et CSS

  Le frontend est responsable de l'interface utilisateur et de l'interaction avec l'utilisateur. Il envoie des requêtes HTTP au backend pour récupérer ou envoyer des données et affiche les résultats de manière dynamique.

- Backend : NodeJs, Express

  Le backend gère la logique de l'application, traite les requêtes HTTP provenant du frontend, interagit avec la base de données et retourne les réponses appropriées au frontend.

- Base de Données : MongoDB et mongoose

  La base de données stocke toutes les données nécessaires, telles que les informations sur les utilisateurs, les objets matériels et les demandes. MongoDB Atlas, une base de données NoSQL, est utilisée pour sa flexibilité et sa scalabilité.

IV. Base de donnees et outils utilisees :

→ Package Mongoose :

- La base de données stocke toutes les données nécessaires, telles que les informations sur les utilisateurs, les objets matériels et les demandes. MongoDB Atlas, une base de données NoSQL, est utilisée pour sa flexibilité et sa scalabilité.
- Mongoose agit comme une couche d'abstraction sur MongoDB, facilitant la définition des schémas de données, des validations et des requêtes.
- Mongoose permet de définir des schémas de données pour chaque collection MongoDB, assurant la cohérence des données et simplifiant le développement.
- Mongoose offre des fonctionnalités intégrées de validation pour garantir l'intégrité des données avant leur enregistrement.
- Support de middleware pour gérer le cycle de vie des données (avant ou après les opérations de sauvegarde, mise à jour, suppression).
- En utilisant Mongoose, le développement d'applications basées sur MongoDB est simplifié tout en exploitant la flexibilité de MongoDB.

→ Flux de Données :

1. Demande de Matériel :
    - Un utilisateur fait une demande de matériel via l'interface utilisateur.
    - Une requête POST est envoyée au backend avec les détails de la demande.
    - Le backend met à jour la base de données pour marquer l'objet comme "demandé" et associe la demande à l'utilisateur.
2. Acceptation/Rejet par l'Administrateur :
    - L'administrateur voit toutes les demandes en attente sur son tableau de bord.
    - L'administrateur peut accepter ou rejeter les demandes.
    - Une requête PUT est envoyée au backend pour mettre à jour le statut de la demande dans la base de données.
3. Retour de Matériel :
    - Après utilisation, l'utilisateur peut retourner le matériel via l'interface utilisateur.
    - Une requête PUT est envoyée au backend pour marquer l'objet comme "disponible" à nouveau dans la base de données.
4. **C**réation d'Utilisateurs :
    - Lorsqu'un utilisateur s'inscrit via l'interface, ses informations sont envoyées au backend.
    - Le backend utilise Mongoose pour créer un nouvel utilisateur dans la collection des utilisateurs de MongoDB Atlas.
5. Page de Connexion et Authentification :
    - Sur la page de connexion, les informations d'identification de l'utilisateur sont vérifiées par le backend.
    - Le backend utilise bcrypt pour le hachage sécurisé des mots de passe avant de les comparer avec ceux stockés en base de données.
6. *Utilisation de JWT (JSON Web Tokens) :*
    - Après la connexion réussie, le backend génère un JWT qui est renvoyé au frontend.
    - Ce JWT est inclus dans les en-têtes des requêtes ultérieures du frontend au backend pour vérifier l'authenticité de l'utilisateur et contrôler l'accès aux ressources sécurisées.

V. Conclusion

Ce rapport détaille le développement et l'implémentation d'une application web utilisant la stack MongoDB, Express, Angular et Node.js. L'utilisation de ces technologies offre une gestion efficace des matériaux universitaires tout en garantissant sécurité et flexibilité. Pour toute question ou démonstration.