

Rapport Technique sur le Développement de l'Application Web

1. Introduction

Ce projet a été réalisé dans le cadre de notre cursus scolaire, avec pour objectif de concevoir une application web visant à améliorer la gestion des ressources matérielles et des utilisateurs dans un environnement éducatif. L'application devait être intuitive, sécurisée et capable de simplifier les processus quotidiens de gestion de stock et de suivi des ressources.

2. Choix Techniques

2.1 Technologies Utilisées

- **Frontend** : Angular (Version 18.0.5)
- **Backend** : Node.js (Version 20.15.0) et Express
- **Base de données** : MongoDB Atlas, avec Mongoose pour l'ODM (Object Data Modeling)
- **Sécurité** : Utilisation de bcryptjs pour le hachage des mots de passe, helmet pour la sécurisation des en-têtes HTTP, et jsonwebtoken pour la gestion des sessions utilisateurs.
- **Journalisation** : Winston pour la capture et la gestion des logs.
- **Gestion des dépendances** : npm (Version 10.8.1)

2.2 Justification des Choix

- **Angular et Node.js** ont été imposés dans le cadre du projet scolaire, offrant une combinaison robuste pour le développement full-stack.
- **Express** offre une plateforme légère et efficace pour construire des applications web serveur, facilitant la gestion des requêtes asynchrones.
- **MongoDB Atlas** a été choisi pour sa flexibilité, sa scalabilité et son intégration facile avec Node.js, permettant une gestion efficace des données dans le cloud.

3. Architecture de l'Application

3.1 Architecture Globale

L'architecture de l'application est divisée en trois composantes principales : le frontend, le backend et la base de données. Le frontend est développé avec Angular, offrant une gestion efficace du routage, de l'authentification et des interactions utilisateur. Angular permet également de créer des applications à page unique (SPA), ce qui améliore l'expérience utilisateur.

Le backend est construit avec Node.js et Express, fournissant une plateforme puissante pour gérer les services web. Cette combinaison permet une gestion efficace des requêtes HTTP, une intégration aisée avec les bases de données, et la mise en œuvre de middleware pour la sécurité et le traitement des données.

La base de données utilise MongoDB Atlas, un service de base de données cloud qui offre une grande flexibilité et scalabilité. Grâce à Mongoose, un ODM pour MongoDB, nous avons pu structurer les données de manière efficace et sécurisée. Cette architecture assure non seulement performance et scalabilité, mais aussi une sécurité accrue grâce à des interactions bien définies entre chaque couche.

3.2 Gestion des Données

La gestion des données est cruciale dans l'architecture de l'application, garantissant l'intégrité, la sécurité et l'accessibilité des informations. Les modèles de données sont définis via Mongoose, facilitant la création de schémas de données rigoureux et leur validation côté serveur. Les principaux modèles incluent :

- **Utilisateurs** : Gère les informations d'authentification et de profil des utilisateurs, y compris les rôles pour différencier l'accès entre administrateurs et utilisateurs standards.
- **Matériel** : Stocke les détails du matériel comme le type, l'état (stocké ou utilisé), la localisation et les informations de responsabilité (utilisateur ou organisme).
- **Demandes** : Enregistre les demandes de matériel par les utilisateurs, le suivi de ces demandes et la gestion des retours.

Chaque entité est soigneusement structurée pour faciliter les requêtes et les mises à jour, tout en assurant la cohérence des données à travers l'application. Des schémas de base de données détaillés sont utilisés pour illustrer ces modèles, aidant à visualiser et comprendre les relations et les flux de données.

4. Fonctionnalités Principales

L'application a été conçue pour offrir une gestion efficace des utilisateurs, du matériel et des demandes. Chaque module est développé pour répondre aux besoins des utilisateurs et des administrateurs, assurant une interaction sécurisée et fluide à travers l'interface utilisateur.

4.1 Gestion des Utilisateurs

La gestion des utilisateurs est une composante essentielle de l'application, permettant aux administrateurs de gérer activement les comptes utilisateurs. Cette fonctionnalité inclut :

- **Création de Compte** : Les utilisateurs peuvent créer leur compte.
- **Ajout et Suppression de Comptes** : Les administrateurs peuvent ajouter ou supprimer des comptes utilisateurs, permettant un contrôle sur qui peut accéder au système.
- **Modification de Comptes** : Les utilisateurs peuvent modifier leurs informations de profil, y compris les mots de passe. Les administrateurs peuvent également modifier les comptes pour une gestion administrative efficace.

4.2 Gestion du Matériel

Le module de gestion du matériel offre des fonctionnalités robustes pour suivre et contrôler les ressources matérielles de l'organisation :

- **Ajout et Suppression de Matériel** : Les administrateurs peuvent ajouter de nouvelles ressources au système ou retirer celles qui ne sont plus en usage, maintenant ainsi un inventaire précis.
- **Consultation de Matériel** : Les utilisateurs peuvent consulter les matériels disponibles pour savoir ce qui est accessible pour l'emprunt ou l'utilisation.
- **Gestion des Attributions et des Retours** : Les utilisateurs peuvent demander l'attribution de matériel spécifique et initier des retours lorsque le matériel n'est plus nécessaire, simplifiant ainsi la circulation des ressources.

4.3 Gestion des Demandes

Cette fonctionnalité permet une administration efficace des requêtes de matériel par les utilisateurs :

- **Acceptation des Demandes d'Attribution** : Les administrateurs peuvent examiner et approuver les demandes d'attribution, ce qui met à jour automatiquement le statut et la localisation du matériel.
- **Réintégration de Matériel** : Lorsque le matériel est retourné, il est réintégré dans le pool disponible, assurant que les ressources sont efficacement recyclées et disponibles pour d'autres utilisateurs.

4.4 Interface Utilisateur et Navigation

L'application est équipée d'interfaces utilisateur distinctes adaptées aux différents types d'utilisateurs :

- **Interface Administrateur** : Offre un contrôle complet sur la gestion des utilisateurs, du matériel et des demandes, avec des outils et des rapports pour surveiller l'activité du système.
- **Interface Utilisateur** : Permet aux utilisateurs d'accéder facilement aux fonctionnalités qui leur sont pertinentes, telles que la consultation de matériel, la gestion des demandes et la mise à jour de leur profil.

5. Sécurité et Authentification

5.1 Authentification et Gestion des Sessions

- **Utilisation de JSON Web Tokens (JWT)** : Les JWT sont utilisés pour gérer les sessions des utilisateurs. Après une authentification réussie, un token signé est délivré à l'utilisateur, permettant des requêtes authentifiées à l'API sans avoir à ressaisir les informations d'identification.
- **Stockage Sécurisé des Mots de Passe** : Les mots de passe sont hashés en utilisant bcryptjs, une bibliothèque robuste qui aide à prévenir les attaques de type brute force et rainbow table. Cela garantit que les mots de passe stockés ne sont pas dans un format lisible.

5.2 Mesures de Sécurité au Niveau du Réseau

- **Configuration CORS (Cross-Origin Resource Sharing)** : Une politique CORS stricte est mise en place pour limiter les requêtes croisées uniquement aux domaines de confiance. Cela aide à protéger l'application contre les attaques telles que le Cross-Site Scripting (XSS) et le Cross-Site Request Forgery (CSRF).

- **Sécurisation des En-têtes HTTP avec Helmet** : L'utilisation de Helmet aide à protéger l'application en configurant de manière sécurisée divers en-têtes HTTP. Cela prévient les attaques comme le sniffing de type MIME et le clickjacking.

6. Conclusion

Ce projet scolaire a été une opportunité précieuse pour appliquer et approfondir nos connaissances en développement web. En construisant une application utilisant Angular pour le frontend et Node.js/Express pour le backend, intégrée à MongoDB Atlas via Mongoose, nous avons pu simuler un environnement de travail réel. Les défis techniques rencontrés nous ont permis de développer des compétences en résolution de problèmes et en prise de décision.

Nous avons mis en œuvre des fonctionnalités clés telles que la gestion des utilisateurs, du matériel et des demandes, démontrant notre capacité à créer des solutions complètes et fonctionnelles. La sécurité des données a été une priorité, avec des mesures robustes pour protéger les informations sensibles et gérer les sessions utilisateurs de manière sécurisée. Les ajustements continus et les améliorations apportées au projet ont garanti que l'application respecte les meilleures pratiques de développement et les normes de sécurité en vigueur.

En conclusion, ce projet a non seulement permis de consolider nos connaissances théoriques, mais aussi d'acquérir une expérience pratique précieuse dans le développement d'applications web. La collaboration entre les membres de l'équipe a été essentielle pour mener à bien ce projet avec succès, préparant ainsi le terrain pour nos futures carrières dans le domaine du développement logiciel.