Maïmouna Sarah Diakité  
Gnanda Paule Axelle Kouamé  
Kponaho Anne-Laure Magnané  
Techniques de l’informatique  
groupe 4318

Projet 3: Explorer une ou des technologies en lien avec le Cloud

Travail présenté à  
M. Nicolas Payre  
Département des Techniques de l’informatique   
Pour le cours  
*Exploration de nouvelles technologies*

Cégep de Sherbrooke  
15 décembre 2023

Table des matières

[Introduction 3](#_Toc152795252)

[Description de la technologie choisie 3](#_Toc152795253)

[Raisons du choix de la technologie 3](#_Toc152795254)

[Node.js : 3](#_Toc152795255)

[MongoDB : 3](#_Toc152795256)

[Résultats de l'exploration technique 4](#_Toc152795257)

[Interface Utilisateur (HTML/CSS/JS) 4](#_Toc152795258)

[Backend (Node.js/Express.js) 4](#_Toc152795259)

[Base de données (MongoDB/Mongoose) 5](#_Toc152795260)

[Exemples concrets d'application 5](#_Toc152795261)

[Recommandations pour son utilisation éventuelle dans des projets 6](#_Toc152795262)

[Extensibilité du Code: 6](#_Toc152795263)

[Sécurité des Données: 6](#_Toc152795264)

[Gestion des Utilisateurs: 6](#_Toc152795265)

[Internationalisation (I18n) et Localisation (L10n): 6](#_Toc152795266)

[Optimisation des Performances: 6](#_Toc152795267)

[Tests et Assurance Qualité: 7](#_Toc152795268)

[Documentation: 7](#_Toc152795269)

[Analyse des Retours Utilisateurs: 7](#_Toc152795270)

[Sauvegarde et Récupération des Données: 7](#_Toc152795271)

[Conformité Légale et Réglementaire: 7](#_Toc152795272)

[Conclusion 8](#_Toc152795273)

# Introduction

L'objectif de ce projet était d'explorer les technologies en lien avec le Cloud en développant une application de gestion de notes. Nous souhaitions créer une application complète, englobant une interface utilisateur, une base de données pour stocker les données des notes, et un serveur API pour gérer les opérations CRUD (Create, Read, Update, Delete) sur ces notes. Notre choix s'est orienté vers un stack technologique basée sur Node.js et MongoDB.

# Description de la technologie choisie

Le stack technologique comprend Node.js pour l'environnement d'exécution côté serveur, Express.js pour la gestion des routes, MongoDB comme base de données NoSQL, et Mongoose comme ODM (Object Data Modeling). Bien que la technologie Cloud ne soit pas explicitement mentionnée dans le stack, les choix effectués ouvrent la voie à une éventuelle intégration avec des services Cloud, notamment MongoDB Atlas, une solution de base de données MongoDB hébergée dans le Cloud.

# Raisons du choix de la technologie

## Node.js :

* **Langage unifié :** L'utilisation de JavaScript côté serveur et côté client garantit une cohérence linguistique tout au long de l'application, simplifiant le développement et la maintenance du code.
* **Évolutivité :** La gestion efficace des connexions simultanées grâce au modèle asynchrone de Node.js le rend idéal pour les applications nécessitant une évolutivité rapide, assurant des performances optimales même dans des conditions de charge élevée.
* **Vaste écosystème :** La richesse de l'écosystème de bibliothèques (npm) de Node.js facilite l'intégration de modules tiers, accélérant ainsi le développement et permettant l'adoption rapide de nouvelles fonctionnalités.

## MongoDB :

* **Flexibilité de schéma** : La capacité de MongoDB à permettre l'évolution de la structure des données sans modifications majeures du schéma offre une grande flexibilité, particulièrement avantageuse pour les applications destinées à évoluer avec le temps.
* **Performances** : Les performances élevées de MongoDB dans la recherche de données, grâce à des index efficaces, contribuent à assurer une expérience utilisateur réactive et fluide.
* **Évolutivité horizontale** : La possibilité de mettre à l'échelle horizontalement MongoDB en répartissant les données sur plusieurs serveurs répond efficacement à la croissance des volumes de données, garantissant ainsi une gestion efficace des ressources.

En combinant Node.js et MongoDB, notre choix repose sur la recherche d'un stack technologique offrant une cohérence linguistique, une adaptabilité aux évolutions de schéma, des performances élevées, une évolutivité efficace, et une préparation pour une éventuelle migration vers des environnements Cloud, ce qui, ensemble, forme une base solide pour le développement d'applications modernes et évolutives.

# Résultats de l'exploration technique

## Interface Utilisateur (HTML/CSS/JS)

Une interface utilisateur a été développée de manière à permettre l'ajout et l'affichage de notes de manière conviviale.

Une image contenant texte, capture d’écran, plein air

Description générée automatiquement

## Backend (Node.js/Express.js)

Un serveur Express a été mis en place, configuré pour gérer les routes API et servir des fichiers statiques, constituant le lien essentiel entre l'interface utilisateur et la base de données.

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

## Base de données (MongoDB/Mongoose)

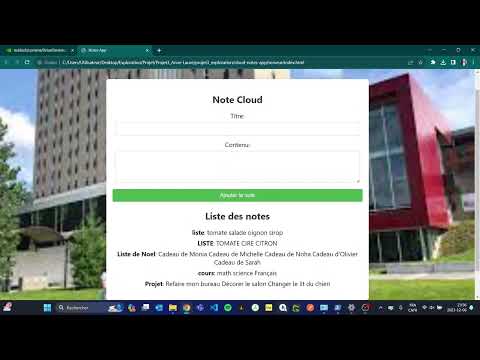
Un schéma Mongoose a été défini pour modéliser les notes, facilitant ainsi les opérations CRUD.

Une image contenant capture d’écran, texte, logiciel

Description générée automatiquement

# Exemples concrets d'application

Bien que l'application développée dans le cadre de ce projet ne tire pas immédiatement parti de services Cloud, elle offre une architecture flexible qui permet une intégration aisée avec des services Cloud pour des besoins tels que la gestion de la base de données dans le Cloud, le déploiement, et la mise à l'échelle dynamique.

[](https://www.youtube.com/embed/cAq5cgYhxAk?feature=oembed)

# Recommandations pour son utilisation éventuelle dans des projets

## Extensibilité du Code:

* **Modularité:** S’assurer que le code est structuré de manière modulaire, ce qui facilitera l'ajout de nouvelles fonctionnalités ou la modification de fonctionnalités existantes.
* **Utilisation de Design Patterns:** Adopter des design patterns appropriés pour garantir la cohérence du code et faciliter l'ajout de fonctionnalités sans introduire de complexités inutiles.

## Sécurité des Données:

* **Cryptage des Données Sensibles:** Si l'application stocke des informations sensibles, s’assurer de mettre en œuvre des mécanismes de cryptage pour garantir la confidentialité des données.
* **Gestion des Sessions:** Si l'authentification est mise en œuvre, s’assurer de gérer les sessions de manière sécurisée, en utilisant, par exemple, des tokens JWT (JSON Web Tokens) avec une expiration appropriée.

## Gestion des Utilisateurs:

* **Fonctionnalités d'Utilisateur:** Envisager d'ajouter des fonctionnalités pour la gestion des utilisateurs, telles que la création de comptes, la récupération de mot de passe, et la personnalisation des profils.
* **Gestion des Autorisations:** Si l'application évolue vers une utilisation multi-utilisateurs, mettre en place un système de gestion des autorisations pour contrôler l'accès aux ressources.

## Internationalisation (I18n) et Localisation (L10n):

* **Préparation pour l'I18n/L10n:** Si l'application est destinée à être utilisée dans des régions linguistiquement diverses, préparer le code pour l'internationalisation (I18n) et la localisation (L10n) afin de rendre l'expérience utilisateur plus accessible.

## Optimisation des Performances:

* **Caching**: Intégrer des mécanismes de mise en cache pour réduire la charge sur le serveur et améliorer les temps de réponse, notamment pour des données fréquemment consultées.
* **Compression des Ressources:** Utiliser la compression des ressources (CSS, JavaScript) pour réduire les temps de chargement de la page.

## Tests et Assurance Qualité:

* **Tests Automatisés:** Continuer à développer des tests automatisés pour garantir la stabilité de l'application à mesure que de nouvelles fonctionnalités sont ajoutées.
* **Tests de Sécurité:** Intégrer des tests de sécurité réguliers pour identifier et corriger les éventuelles vulnérabilités.

## Documentation:

* **Documentation Utilisateur:** Rédiger une documentation utilisateur complète pour guider les utilisateurs finaux dans l'utilisation de l'application.
* **Documentation Technique:** Continuer à maintenir une documentation technique à jour pour faciliter la compréhension du code par les développeurs futurs.

## Analyse des Retours Utilisateurs:

* **Collecte des Retours:** Mettre en place des mécanismes pour collecter les retours utilisateurs, que ce soit par le biais de fonctionnalités intégrées dans l'application ou par le suivi des commentaires dans les app stores.
* **Itérations Basées sur les Retours:** Utiliser les retours des utilisateurs pour planifier des itérations futures, en mettant l'accent sur l'amélioration des fonctionnalités existantes ou l'ajout de nouvelles fonctionnalités demandées.

## Sauvegarde et Récupération des Données:

* **Politique de Sauvegarde:** Établir une politique de sauvegarde régulière pour garantir la sécurité des données en cas de défaillance du système.
* **Mécanismes de Récupération:** Planifier des mécanismes de récupération des données en cas de perte accidentelle, en particulier si l'application gère des données critiques.

## Conformité Légale et Réglementaire:

* **Audit Légal:** S’assurer que l'application respecte les normes légales et réglementaires applicables à votre domaine, en particulier en ce qui concerne la confidentialité des données et la sécurité.
* **Mises à Jour de Conformité:** Se tenir informé des évolutions légales et s’assurer que l'application est régulièrement mise à jour pour rester conforme.

En suivant ces recommandations, on peut garantir que l'application de gestion de notes est prête à évoluer et à répondre aux besoins changeants des utilisateurs et des exigences technologiques. Cela contribuera également à assurer la robustesse, la sécurité et la satisfaction des utilisateurs dans des projets futurs.

# Conclusion

En résumé, ce projet a été une opportunité fructueuse pour explorer les technologies émergentes, tester des concepts pratiques, et tirer des enseignements pour notre parcours de développement. En continuant à itérer et à améliorer, cette application de gestion de notes pourrait non seulement répondre aux besoins actuels, mais également s'adapter avec agilité aux défis à venir dans le domaine du développement logiciel.