Projet final

Bienvenue au projet de développement d'une application pour le cinéma NothingBetterThanAL, qui est ouvert du lundi au vendredi de 9h à 20h. Ce projet consiste à créer une API RESTful en Node.js et TypeScript comportant plusieurs fonctionnalités clés autour de la gestion d'un cinéma. Voici les détails du projet:

Gestion des Salles de Cinéma

- **Objectif** : Implémenter les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) pour les salles de cinéma. Chaque salle doit avoir un nom, une description, des images, un type, une capacité et un optionnellement un accès pour les personnes handicapées . Il est nécessaire d'avoir au moins 10 salles.
- **Contraintes** : La capacité de chaque salle varie, sans dépasser 30 places et avec un mimum de 15 places. Les salles peuvent être mises en maintenance par les administrateurs, empêchant toute séance de se tenir dans celles-ci (les utilisateurs ne devront même plus pouvoir voir les films se passant dans cette salle).
- Tout utilisateur authentifié doit être capable de regarder le planning d'une salle de cinéma sur une période qu'il choisie. il doit donc être capable par exemple de savoir tout le planning de la salle 6 sur une semaine si il le désire quelle soit dans le futur ou le passé.

Gestion des Séances

- **Objectif** : Permettre aux administrateurs l'ajout, la modification, la suppression des séances. N'importe qui pourra utiliser la consultation des séances cinématographiques. Il est important de planifier au moins un mois de séances à l'avance.
- **Contraintes** : Éviter les chevauchements de séances pour un même film dans différentes salles. Chaque séance doit durer au moins la durée du film plus 30 minutes pour les publicités et le nettoyage.
- Tout utilisateur authentifié doit être capable de regarder le planning des séances prévu dans le cinéma sur une période qu'il choisie. Il doit aussi être capable sur les séances de savoir combien de billets ont été vendu et donc combien de spectateurs ont assisté à la séance

Gestion des Films

- **Objectif** : Permettre aux administrateurs d'ajouter, modifier, supprimer. N'importe quel utilisatuer peut consulter les films disponibles dans les films dispo pour le cinéma.
- Tout utilisateur authentifié doit être capable de regarder le planning des films qui passent dans le cinéma.Exemple: il doit pouvoir voir toutes les séances de la reine des neiges sur une période choisie.

Utilisateurs de l'API

- **Rôles** : Différencier les utilisateurs en administrateurs, (BONUS) super administrateurs (pouvant gérer le planning des employés), et clients (pouvant consulter les plannings des salles, des séances et des films...).
- Un utilisateur devra pouvoir se créer un compte, s'authentifié, et se logout (supprimer tous ses tokens)
- Chaque route d'api ne doit être accessible que pour des utilisateurs authentifiés.
- Pour un administrateur il devra être possible de visualiser les utilisateurs de l'API ainsi que de voir des données précises sur leur compte et leur activité au sein du cinéma (films vues...)

(Bonus) Gestion du Planning des Employés (pour Super Administrateurs)

- **Objectif** : Gérer le planning des employés du cinéma, incluant des postes spécifiques avec la restriction d'un seul employé par poste à tout moment.
- **Poste:** confiserie, accueil, projectionniste.
- Pour que le cinéma ouvre chaque poste doit avoir un travailleur
- CRUD pour la gestion des employés.

Gestion des Billets

- **Objectif** : Implémenter un système de billetterie permettant la validation de l'accès aux séances, avec des billets qui permette d’accéder à une séance et des "Super Billets" pour accéder à 10 séances.
- Il devra être possible pour un utilisateur de savoir quels billets il a utilisé et pour accéder à quel séance.

Gestion de l’Argent

- **Objectif** : Permettre aux utilisateurs de gérer leur compte en euros, d'acheter des billets et de consulter leurs transactions et solde. Les administrateurs peuvent voir les transactions de tous les clients.
- Un utilisateur pourra donc mettre de l’argent dans son compte, en retirer, regarder un historique de ses transactions datés (achat de billet).

Statistiques et Fréquentation

- **Objectif** : Suivre quotidiennement et hebdomadairement l'affluence dans les salles et au cinéma en général, incluant un suivi en temps réel du taux de fréquentation, on doit aussi être capable de récupérer les statistiques de fréquentation sur une période donnée.

Conseils et Exigences

- mettre en production l’application et la base de donnée est obligatoire
- faire une documentation de l’utilisation de l’API (Swagger/OpenAPI recommandé) est indispensable
- Utiliser des formats de dates standardisés (ISO-8601) **conseil**
- architecturer l’API soigneusement pour simplifier son utilisation **conseil**
- Soyez logique certaines choses ne sont pas dit dans le sujet mais sont évidentes (on ne peut pas créer une séance quand le cinéma est fermé avant 9h ou après 20h. On ne peut pas vendre un billet à un utilisateur qui n’a pas d’argent...)

Bonus

- Conteneuriser l'application et la base de données(aussi bien en développement que en production),
- intégrer des tests(unitaire ou intégration ou end2end)
- un système de sauvegarde des données(backup)
- de l'observabilité(Prometheus, Grafana)
- un système de logs (log bien formatté par exemple en JSON)
- mise en place d’une intégration continue (CI) (gitlab-ci, github action, Jenkins)
- gestion des race conditions(vente de place pour une séance...).
- tout ce que je trouverais intéréssant que vous avez fait en plus (une belle architecture, une belle infrastructure pour votre application, un front ...)

Soutenance

- Préparer l'API pour une démonstration, incluant un scénario de requêtes pour illustrer chaque fonctionnalité implémentée.