Node.js & MongoDB

Projet 1 : Réservation de Matériel pour un Laboratoire

L'application permet aux utilisateurs de réserver du matériel de laboratoire et aux administrateurs de gérer ces réservations. L'authentification est requise et des notifications par e-mail sont envoyées aux utilisateurs.

Objectifs

• Offrir une interface de réservation simple et efficace.
• Permettre aux administrateurs de gérer les objets et les réservations.
• Assurer une traçabilité des prêts et retours.
• Automatiser les notifications et rappels par e-mail.

Fonctionnalités

1. **Authentification & gestion des utilisateurs**
- Inscription avec email et mot de passe.
- Connexion sécurisée avec JWT.
- Mot de passe oublié (envoi d'un email de récupération).
- Rôles : Utilisateur et Admin.

2. **Gestion du matériel**
- Ajout, modification et suppression d'un objet par l'administrateur.
- Informations stockées : Nom, Description, Numéro de série, Photo, État.

3. **Réservation de matériel**
- Un utilisateur peut réserver un objet disponible.
- L'admin valide ou refuse la demande.
- Envoi d'un email de confirmation ou refus.

4. **Retour de matériel**
- L'utilisateur signale le retour de l'objet.
- L'admin valide et met à jour l'état de l'objet.
- Envoi d'un email de confirmation de retour.

5. **Notifications par e-mail**
- Confirmation de réservation, Rappel pour le retour, Mot de passe oublié.

Projet 2 : Réservation de Casiers

Cette application permet aux utilisateurs de réserver un casier en ligne, sur le même principe que la réservation de places au cinéma. Chaque casier a une durée de réservation et est libéré automatiquement une fois expiré.

Objectifs

• Permettre la réservation rapide et intuitive d'un casier.
• Automatiser l'expiration et la libération des casiers.
• Notifier l'utilisateur par e-mail à chaque étape.

Fonctionnalités

1. **Authentification & gestion des utilisateurs**
- Inscription et connexion avec JWT.
- Mot de passe oublié (e-mail de récupération).
- Rôles : Utilisateur et Admin.

2. **Gestion des casiers**
- Ajout, modification et suppression d'un casier par l'admin.
- Informations stockées : Numéro du casier, Taille, Statut, Prix.

3. **Réservation d'un casier**
- L'utilisateur sélectionne un casier disponible.
- Il choisit une durée de réservation.
- Paiement fictif via Stripe (optionnel).
- Envoi d'un e-mail de confirmation avec le numéro du casier et la durée.

4. **Expiration et libération du casier**
- L'application gère automatiquement l'expiration.
- Envoi d'un e-mail de rappel avant expiration.
- Une fois expiré, le casier est libéré et remis en stock.

5. **Notifications par e-mail**
- Confirmation de réservation, Rappel avant expiration du casier, Mot de passe oublié.

Barème d'évaluation (sur 20 points)

1. Fonctionnalités & respect du cahier des charges (8 points)

Critères	Points
Authentification (inscription, connexion, JWT, récupération de mot de passe)	2
Gestion des objets (matériel ou casiers) : CRUD + images (si matériel)	2
Réservation et gestion des statuts (disponible, réservé, etc.)	2
Notifications par e-mail (réservation confirmée, retour, expiration)	2

2. Qualité du code & architecture (5 points)

Critères	Points
Code structuré, bien organisé (MVC, routes séparées, bonne utilisation des middlewares)	2
Utilisation correcte de MongoDB (modèles bien définis, requêtes optimisées)	1
Sécurisation (hash du mot de passe, JWT bien implémenté, validation des entrées)	2

3. Frontend (4 points)

Critères	Points
UI propre et responsive (MUI ou autre)	2
Navigation fluide (React Router, gestion des états)	1
Interaction backend (requêtes bien faites, état synchronisé)	1

4. Présentation orale (3 points)

Critères	Points
Explication claire du projet (objectifs, fonctionnalités, choix techniques)	1
Démonstration fluide et pertinente (navigation, tests des fonctionnalités)	1
Capacité à répondre aux questions, maîtrise du sujet	1