

📄 Cahier de charge — Application *CampusLabé*

🎯 Objectif

Créer une plateforme web pour faciliter la gestion des logements étudiants à Labé, avec des rôles distincts (**étudiant**, **propriétaire**) et des fonctionnalités de réservation, communication et notification.

🏗️ Architecture technique

Composant	Technologie utilisée
Backend	Node.js + Express
Base de données	MongoDB Atlas
Authentification	Supabase Auth (JWT)
Déploiement	Render
Tests API	Postman

👤 Rôles utilisateurs

- **Étudiant** : peut consulter les chambres, réserver, envoyer des messages
- **Propriétaire** : peut ajouter/modifier/supprimer ses chambres, recevoir des réservations
- **Admin (optionnel)** : supervise les utilisateurs et les données

📦 Modèles de données (MongoDB)

🏠 chambres

```
{ "_id": ObjectId, "nom": String, "description": String, "prix": Number, "disponible": Boolean, "proprietaire_id": ObjectId }
```

📅 reservations

```
{ "_id": ObjectId, "etudiant_id": ObjectId, "chambre_id": ObjectId, "statut": String, // "en attente", "acceptée", "refusée" "date_debut": Date, "date_fin": Date }
```

👤 utilisateurs

```
{ "_id": ObjectId, "nom": String, "email": String, "role": String, // "etudiant", "proprietaire" "photo": String }
```

💬 messages

```
{ "_id": ObjectId, "expediteur_id": ObjectId, "destinataire_id": ObjectId, "contenu": String, "timestamp": Date, "lu": Boolean }
```

🔔 notifications

```
{ "_id": ObjectId, "utilisateur_id": ObjectId, "type": String, // "reservation", "message", "alerte" "contenu": String, "vue": Boolean, "timestamp": Date }
```

Routes API (backend)

Chambres

- GET /api/chambres → liste des chambres
- POST /api/chambres → ajouter une chambre
- PUT /api/chambres/:id → modifier une chambre
- DELETE /api/chambres/:id → supprimer une chambre

Réservations

- GET /api/reservations → liste des réservations
- POST /api/reservations → créer une réservation
- PUT /api/reservations/:id → modifier une réservation
- DELETE /api/reservations/:id → supprimer une réservation

Messages

- GET /api/messages → messages reçus/envoyés
- POST /api/messages → envoyer un message

Notifications

- GET /api/notifications → voir les notifications
- PUT /api/notifications/:id → marquer comme vue

Sécurité (à réintégrer plus tard)

- Middleware `auth.js` pour vérifier le token Supabase
- Middleware `checkRole.js` pour limiter l'accès selon le rôle
- Vérification que seul le propriétaire peut modifier ses chambres
- Vérification que seul l'étudiant peut réserver ou modifier ses réservations

Objectifs à court terme

- Finaliser les routes CRUD pour chambres et réservations
- Tester toutes les routes avec Postman
- Ajouter les routes pour messages et notifications
- Préparer l'intégration du frontend React