



Document de présentation du projet Valentin Dubach

1. Présentation du projet choisi

Ce projet est une application de **gestion de réservations de salles et de ressources** :

- **Frontend** : React + TypeScript, avec des pages pour créer, lister et supprimer des réservations.
- **Backend** : Express + Prisma + JWT pour l'authentification.
- **Base de données** : Prisma gère les modèles User, Room, Resource, Reservation.
- **Objectif** : permettre aux utilisateurs authentifiés de réserver des salles et des ressources, avec validation des horaires et gestion des droits.

2. UML (classes + séquences)



Diagramme de classes (simplifié)

Classe	Attributs principaux	Relations
User	id, email, password	1..* Reservations
Room	id, name, capacity, location	1..* Reservations
Resource	id, type, description, available	1..* Reservations
Reservation	id, date, startTime, endTime, userId, roomId, resourceId	Associe User, Room, Resource



Diagramme de séquence (exemple : création d'une réservation)

Code

```
Utilisateur -> Frontend (ReservationFormPage): Remplit formulaire
Frontend -> Backend (POST /api/reservations): Envoie données + token
Backend -> Prisma: Vérifie User, Room, Resource
Prisma -> Backend: Réservation créée
Backend -> Frontend: Retourne JSON {reservation}
Frontend -> Utilisateur: Affiche confirmation
```

3. Maquettes UI



Pages principales

- **LoginPage** : formulaire email + mot de passe.
- **ReservationsPage** : liste des réservations avec bouton "Supprimer".
- **ReservationFormPage** : formulaire de création avec sélecteurs de salle et ressource, date et heures.
- **RoomsPage / ResourcesPage** : gestion des salles et ressources (admin).

(Les maquettes peuvent être réalisées avec Figma ou un simple wireframe : champs, boutons, listes, alertes d'erreur.)

4. Démonstration du front

- **Connexion** : l'utilisateur se connecte avec email/mot de passe → reçoit un JWT.
- **Création réservation** : formulaire validé → POST /api/reservations.
- **Liste** : GET /api/reservations → affichage des réservations de l'utilisateur.
- **Suppression** : bouton "Supprimer" → DELETE /api/reservations/:id.

5. Démonstration du back

- **Auth** : route /api/auth/login avec bcrypt + JWT.
- **Reservations** : routes GET, POST, DELETE protégées par authenticateToken.
- **Rooms/Resources** : CRUD complet (GET, POST, PUT, DELETE).
- **Validation** : middleware vérifie le token, et la logique vérifie que la réservation appartient bien à l'utilisateur.

6. Principaux prompts utilisés

- "J'ai cette erreur en voulant supprimer une réservation" → ajout de la route DELETE.
- "La page de nouvelle réservation renvoie une page blanche" → correction du formulaire et gestion des erreurs.
- "api is not a function" → refactorisation des services pour utiliser apiClient.
- "GET /api/rooms 403 Forbidden" → ajout du token dans les appels front.
- "Adapte mon ReservationFormPage.tsx" → intégration du token et validation endTime > startTime.

7. Analyse critique : apports et limites de l'IA

✓ Apports

- **Gain de temps** : corrections rapides des erreurs (404, 403, incohérences).
- **Clarté** : explications pédagogiques sur la cause des bugs.
- **Refactorisation** : harmonisation des services (roomsService, resourcesService) avec apiClient.
- **Validation** : ajout de règles métier (endTime > startTime).
- **Documentation** : génération d'un document complet (présentation, UML, démo).

Limites

- **Pas de test direct** : l'IA ne peut pas exécuter ton code ni voir ton écran.
- **Besoin de contexte** : certaines erreurs nécessitent que tu montres ton schéma Prisma ou ton code exact.

- **Pas de maquettes graphiques** : je peux décrire les UI, mais pas générer des wireframes interactifs.
- **Dépendance** : risque de sur-dépendance à l'IA pour des corrections simples au lieu d'apprendre à déboguer soi-même.

Conclusion

Ce projet illustre une **application complète de gestion de réservations** avec un front React et un back Express/Prisma. L'IA a permis de résoudre les erreurs critiques, d'améliorer la cohérence du code et de produire une documentation claire.

ANNEXES – MOCKUPS UI :

Rapport

CHECKPOINT SIX

Room reservation

Reservations >

Book a room

Meeting name (optional)

Date

Start


End

Book a room

Présentation du projet choisi.
UML (classes + séquences).
Maquettes Ui.
Démonstration du front,
Principaux prompts utili.
Analyse critique : apports et limites de de l'IA.

Nouvelle réservation

Salle

Salle de réunion A – 8 places 

Ressource (optionnel)

Aucune

Date

2024-03-01

Heure début

10:00

Heure de fin