Cahier des Charges — Projet React

Titre du projet : MedFlow — SaaS pour Cliniques & Médecins
Technologies principales : React/Next.js,

---

1. Contexte & Objectifs

- Digitaliser la gestion d'une clinique médicale.
- Permettre aux cliniques de gérer leurs patients, leurs médecins, leurs rendez-vous et leur facturation.
- Fournir un portail aux patients pour réserver, consulter et télécharger leurs documents.
- Projet destiné à l'apprentissage : les étudiants doivent pratiquer **conception logicielle**, **développement full-stack**, **sécurité**, et **déploiement**.

---

2. Périmètre Fonctionnel

2.1 Rôles Utilisateurs

- **Admin (propriétaire)** : création clinique, configuration des services, gestion du staff.
- **Médecin** : gestion agenda, dossiers médicaux, ordonnances.
- **Réceptionniste** : rendez-vous, enregistrement patients, facturation.
- **Patient (portail)** : réservation/modification, paiement, téléchargement ordonnances PDF.

2.2 Modules Principaux

1. **Authentification & RBAC** : création comptes, rôles, permissions.
2. **Gestion Patients** : CRUD complet, profil, visites, historique médical.
3. **Agenda & Rendez-vous** : prise, modification, annulation, calendrier.

4. **Consultations & Ordonnances** : diagnostic, prescription, export PDF.
5. **Facturation & Paiement** : factures, suivi paiements, intégration Stripe (mode test).
6. **Portail Patient** : espace dédié (consultation + paiement en ligne).
7. **Administration & Paramétrage** : gestion des services, tarifs, staff.

---

3. Contraintes Techniques

- **Front-end :** Next.js 14, React, Tailwind, shadcn/ui.
- **Back-end :** Next.js API Routes (ou NestJS si séparation)ou selon votre choix .
- **Base de données :** PostgreSQL ou Mysql.
- **Sécurité :** Auth.js, hashing mots de passe, validation formulaires (Zod).
- **Multi-tenant :** chaque clinique isolée par `tenantId (optional)`.
- **Déploiement :** Vercel (Front) + Railway/Render (DB + API).

---

4. Livrables Attendus

1. **Conception :**
    - Diagrammes UML (cas d'utilisation, classes, séquence).
    - Schéma de base de données (ERD).
    - Maquettes d'écrans (Figma ou autre).
2. **Développement :**
    - Code source (GitHub/GitLab).
    - Scripts de migration & seed (Prisma).
    - Documentation (README).

3. **Réalisation :**
    - Version déployée (Vercel ou autre).
    - Démonstration vidéo (2–3 min).

---

5. Planning & Méthodologie

- **Méthode agile (SCRUM)** avec 5 sprints.
- **Sprint 1 :** Auth + Onboarding + Dashboard simple.
- **Sprint 2 :** CRUD Patients + Services + Rendez-vous.
- **Sprint 3 :** Consultations + Ordonnances PDF.
- **Sprint 4 :** Facturation + Paiement + Portail Patient.
- **Sprint 5 :** Bonus (analytics, calendrier avancé, notifications Email).

---

6. Évaluation

- **Qualité conception (diagrammes, modèles, maquettes) :** 25%
- **Code & architecture :** 25%
- **Fonctionnalités livrées (MVP) :** 30%
- **UX/UI et ergonomie :** 10%
- **Documentation & démo :** 10%

👉 Les étudiants doivent couvrir **tout le cycle de vie du projet** :

- Étude et analyse des besoins.
- Conception UML & maquettes.

- Développement et intégration.
- Tests et validation.
- Déploiement et présentation finale.