



Application Full Stack pour la Gestion d'un Cabinet Médical

Cahier des charges

1. Introduction

1.1 Contexte

Le cabinet médical souhaite moderniser ses processus de gestion en mettant en place une application web Full Stack. Cette application doit permettre la gestion efficace des patients, des rendez-vous, des consultations, des dossiers médicaux et du personnel médical. L'application sera développée en utilisant les technologies ReactJS, NodeJS, NextJS, et MongoDB.

1.2 Objectifs

- Fournir une interface utilisateur intuitive pour le personnel médical.
- Faciliter la gestion des rendez-vous, des consultations et des dossiers médicaux.
- Assurer une sécurité optimale des données médicales.
- Offrir des fonctionnalités de reporting et de statistiques pour le suivi des activités du cabinet.

2. Description du Projet

2.1 Fonctionnalités Principales

2.1.1 Gestion des Patients

- Création, modification et suppression de fiches patients.
- Consultation de l'historique médical des patients.
- Gestion des informations personnelles et des contacts d'urgence.

2.1.2 Gestion des Rendez-vous

- Planification et gestion des rendez-vous.
- Consultation du calendrier des rendez-vous.
- Suivi des ordres de sortie et de mission.

2.1.3 Gestion des Consultations

- Enregistrement des détails des consultations (diagnostics, prescriptions, notes).
- Accès aux antécédents médicaux lors des consultations.
- Génération de prescriptions et de certificats médicaux.

2.1.4 Gestion des Dossiers Médicaux

- Stockage et accès sécurisé aux dossiers médicaux.
- Possibilité d'ajouter des documents et des résultats d'examens.
- Recherche avancée dans les dossiers médicaux.

2.1.5 Gestion des Médecins et du Personnel

- Ajout, modification et suppression des fiches médecins et personnel.
- Gestion des horaires et des disponibilités des médecins.
- Attribution des rendez-vous aux médecins.

2.1.6 Tableau de Bord et Reporting

- Tableau de bord pour une vue d'ensemble des activités du cabinet.
- Rapports sur les consultations, les rendez-vous et les patients.
- Statistiques sur les performances et l'occupation des médecins.

2.2 Fonctionnalités Avancées

2.2.1 Authentification et Sécurité

- Authentification sécurisée des utilisateurs.
- Gestion des rôles et des permissions (administrateur, médecin, secrétaire).
- Chiffrement des données sensibles.

2.2.2 Notifications et Alertes

- Notifications par email pour les rappels de rendez-vous.
- Alertes pour les consultations à venir et les tâches importantes.

2.2.3 Intégration avec des Systèmes Externes

- Intégration avec les systèmes de paiement pour les consultations en ligne.
- Intégration avec les laboratoires pour l'importation des résultats d'examens.

3. Architecture Technique

3.1 Technologies Utilisées

- **Front-end** : ReactJS
- **Back-end** : NodeJS, ExpressJS
- **Framework** : NextJS
- **Base de Données** : MongoDB
- **Serveur** : Hébergement sur un serveur cloud (ex. AWS, Heroku)
- **Outils de Suivi et de Collaboration** : GitHub, Trello

3.2 Structure de l'Application

3.2.1 Front-end (ReactJS et NextJS)

- Utilisation de composants réutilisables pour les interfaces utilisateur.
- Mise en place de la navigation dynamique avec NextJS.

- Gestion de l'état global avec Redux ou Context API.
- Tests unitaires avec Jest et React Testing Library.

3.2.2 Back-end (NodeJS et ExpressJS)

- API RESTful pour la communication avec le front-end.
- Gestion de l'authentification avec JWT.
- Implémentation des services de notification (ex. Twilio pour les SMS).

3.2.3 Base de Données (MongoDB)

- Schémas de données pour les patients, les rendez-vous, les consultations et les utilisateurs.
- Indexation pour des performances de requête optimales.

4. Sécurité

4.1 Sécurité des Données

- Chiffrement des données sensibles (ex. mots de passe).
- Utilisation de HTTPS pour les communications sécurisées.
- Sauvegardes régulières de la base de données.

4.2 Gestion des Accès

- Contrôle des accès basé sur les rôles (RBAC).
- Journalisation des activités des utilisateurs.

5. Planning et Livrables

5.1 Phases du Projet

5.1.1 Phase de Conception (2 semaines)

- Définition des spécifications détaillées.
- Création de maquettes et prototypes.

5.1.2 Phase de Développement (8 semaines)

- Développement des fonctionnalités back-end et front-end.
- Intégration et tests unitaires.

5.1.3 Phase de Test et de Validation (2 semaines)

- Tests fonctionnels et de performance.
- Correction des bugs et ajustements.

5.1.4 Phase de Déploiement (1 semaine)

- Déploiement sur le serveur de production.
- Configuration des outils de monitoring et de sauvegarde.

5.1.5 Phase de Formation et Documentation (1 semaine)

- Création de la documentation utilisateur.
- Formation des utilisateurs clés.

5.2 Livrables

- Application web fonctionnelle et déployée.
- Documentation technique et utilisateur.
- Rapport de test et de validation.

6. Ressources

- Chef de projet
- Développeurs front-end et back-end
- Designer UX/UI
- Testeurs QA
- Spécialiste DevOps