Product Requirements Document (PRD)

# 1. Application Overview

L'application ISJ (Institut Saint Joseph) est une plateforme web de gestion scolaire destinée aux établissements secondaires en RDC. Elle permet à l'administration de gérer efficacement les enseignants, élèves, parents, finances, bulletins, emploi du temps, communication, documents et paramètres de l'école. L'application est développée avec Next.js pour le frontend, Express.js pour le backend, MongoDB (Atlas) pour la base de données, Prisma comme ORM, et suit le modèle Waterfall (en cascade).

# 2. List of Pages

- /login  
- /admin/dashboard  
- /admin/users  
- /admin/teachers  
- /admin/students  
- /admin/classes  
- /admin/notes  
- /admin/finance  
- /admin/planning  
- /admin/communication  
- /admin/documents  
- /admin/settings

# 3. Functionality per Page

/admin/dashboard:  
- Statistiques dynamiques : nombre d’élèves, enseignants, classes, taux de présence, paiements, performances.  
  
/admin/users:  
- Gestion des comptes : ajouter, modifier, supprimer des utilisateurs selon leur rôle (admin, enseignant, élève, parent).  
  
/admin/teachers:  
- CRUD enseignants, assignation à des classes/matières, suivi de présence, performance.  
  
/admin/students:  
- Inscription, dossiers, répartition, transfert, abandon.  
  
/admin/classes:  
- Définition classes/sections/options, attribution matières.  
  
/admin/notes:  
- Périodes, saisie/validation de notes, génération bulletins, classement.  
  
/admin/finance:  
- Gestion frais scolaires, paiements, reçus, relances, export comptable.  
  
/admin/planning:  
- Création d’emplois du temps, gestion de salles, notifications.  
  
/admin/communication:  
- Messages internes, notifications (retards, absences, bulletins, paiements).  
  
/admin/documents:  
- Archivage, téléversement de fichiers administratifs.  
  
/admin/settings:  
- Infos école, année scolaire, coefficients, discipline.

# 4. Database Requirements

Collections principales :  
- users (role, status, email, password, etc.)  
- students (userId, classeId, parentId, dossier...)  
- teachers (userId, matières, classes...)  
- classes (nom, niveau, section, options)  
- subjects (nom, niveau, coefficient)  
- notes (studentId, subjectId, trimestre, note)  
- payments (élève, montant, date, tranche)  
- timetable (classe, jour, heure, enseignant, matière)  
- messages, notifications, documents

# 5. Design (UI/UX)

- Design moderne et responsive  
- Mode sombre avec contraste corrigé  
- Palette sombre : gris foncé, bleu nuit, vert clair, orange doux  
- Utilisation d’icônes claires et textes lisibles  
- Navigation simple via une sidebar fixe (AdminLayout)

# 6. Sequence of Development

Suivi du modèle Waterfall, chaque étape sera finalisée avant la suivante :  
1. Authentification & rôles (fait)  
2. Dashboard admin (en cours)  
3. Gestion des enseignants  
4. Gestion des élèves  
5. Gestion académique (classes/matières)  
6. Gestion notes & bulletins  
7. Gestion financière  
8. Planning & emploi du temps  
9. Communication interne  
10. Gestion documentaire  
11. Paramètres de l’école

# 7. UI Priorities

- Dashboard clair avec indicateurs visuels (cards, graphiques)  
- Formulaires intuitifs pour ajout/édition de données  
- Tableaux filtrables pour utilisateurs, paiements, notes  
- Notifications et toasts en cas d’actions