



# Product Requirements & Specification Document

---

## Project Name

**MedSync - Secure Healthcare Appointment Platform**

## Description

MedSync is a secure, full-featured healthcare appointment booking and management system. It supports user authentication, role-based authorization (patients, doctors, admins), secure medical record uploads, real-time notifications, RESTful APIs, JWT authentication, and integrates with PostgreSQL and MongoDB. An admin dashboard provides analytics and user management.

---

## 1. Goals & Objectives

Goal	Description
Secure Appointment Booking	Enable patients to book, view, and manage appointments securely.
Role-Based Access	Distinct access for patients, doctors, and admins.
Medical Record Management	Secure upload and retrieval of medical files.
Real-Time Notifications	Notify users of appointment updates and reminders.
Analytics & Admin Dashboard	Provide insights and user management for admins.

---

## 2. User Roles & Permissions

Role	Permissions
Patient	Register, login, manage profile, book/view/cancel appointments, upload files
Doctor	Register, login, manage profile, view/manage appointments, view files
Admin	Full access: manage users, appointments, analytics, system settings

---

## 3. Core Features

Feature	Description
User Authentication	JWT-based registration, login, password reset
Role-Based Authorization	Access control for patients, doctors, admins
Appointment Management	Book, reschedule, cancel, and view appointments
Medical Record Uploads	Secure file upload (Multer), storage in MongoDB
Real-Time Notifications	WebSocket or push notifications for updates/reminders



Admin Dashboard	User management, analytics (appointments, user activity, system health)
RESTful API	All core operations exposed via REST endpoints

## 4. Technical Specifications

### 4.1 Architecture

- **Frontend:** Next.js, React, Tailwind CSS, TypeScript
- **Backend:** Node.js, Express, TypeScript
- **Databases:** PostgreSQL (relational data), MongoDB (file storage, logs)
- **Authentication:** JWT (access & refresh tokens)
- **File Uploads:** Multer (Express middleware)
- **Notifications:** WebSocket (Socket.io) or push notifications

### 4.2 Data Model Overview

Entity	Storage	Key Fields
User	PostgreSQL	id, name, email, password_hash, role, profile
Appointment	PostgreSQL	id, patient_id, doctor_id, datetime, status, notes
MedicalRecord	MongoDB	_id, user_id, file_url, metadata, uploaded_at
Notification	MongoDB	_id, user_id, type, message, read, created_at
Analytics	PostgreSQL	Various aggregated tables/views

### 4.3 API Endpoints (Sample)

```
POST  /api/auth/register
POST  /api/auth/login
GET   /api/appointments
POST  /api/appointments
PUT   /api/appointments/:id
DELETE /api/appointments/:id
POST  /api/records/upload
GET   /api/records/:userId
GET   /api/admin/users
GET   /api/admin/analytics
```

## 5. Security & Compliance

Area	Specification
Authentication	JWT (access/refresh), password hashing (bcrypt)
Authorization	Middleware-based role checks
File Security	Virus scan, file type/size validation, access control



Data Protection	HTTPS, encrypted storage for sensitive data
Audit Logging	All critical actions logged (MongoDB)
Compliance	GDPR-ready, HIPAA-aligned best practices

## 6. UI/UX Requirements

Area	Specification
Theme	Futuristic, clean, startup-inspired
Responsiveness	Fully responsive (mobile, tablet, desktop)
Accessibility	WCAG 2.1 AA compliance
Key Screens	Login/Register, Dashboard, Appointment Booking, Records, Admin

## 7. Analytics & Admin Dashboard

Feature	Description
User Management	List, search, edit, deactivate users
Appointment Stats	Bookings per day/week/month, cancellations, no-shows
System Health	Uptime, error rates, API usage
Export Data	CSV/Excel export for reports

## 8. Non-Functional Requirements

Area	Specification
Performance	<300ms API response for 95% of requests
Scalability	Horizontal scaling for backend and databases
Availability	99.9% uptime
Localization	Ready for i18n (English default)
Documentation	API docs (OpenAPI/Swagger), user guides

## 9. Milestones & Deliverables

Milestone	Deliverable	Timeline
MVP	Core booking, auth, file upload, dashboard	Month 1-2
Notifications	Real-time notifications	Month 3
Analytics	Admin analytics dashboard	Month 4



Final Release	Full feature set, documentation	Month 5
---------------	---------------------------------	---------

---

## 10. Acceptance Criteria

- All core features implemented and tested
- Security and compliance requirements met
- Responsive, accessible UI/UX
- Admin dashboard functional with analytics
- API documentation complete

---

**End of Document**