

Doctor Appointment System

Complete Project Requirements Document

Full Stack Web Application

Generated: January 26, 2026

Table of Contents

1. Project Overview
2. User Roles
3. Core Features
4. Technical Stack
5. Database Schema
6. API Endpoints
7. UI/UX Requirements
 - 7.1 Navbar Specifications
 - 7.2 Home Page Components
 - 7.3 Find Doctors Page
 - 7.4 Doctor Profile Page
 - 7.5 Appointment Booking
8. Key Pages Structure
9. Additional Features
10. Development Phases

1. Project Overview

The Doctor Appointment System is a comprehensive full-stack web application designed to streamline the healthcare appointment process. This platform connects patients with qualified doctors, enabling seamless appointment booking, management, and consultation services. The system supports three distinct user roles: Patients, Doctors, and Administrators, each with specific functionalities tailored to their needs.

Key Objectives:

- Simplify the appointment booking process for patients
- Provide doctors with efficient schedule management tools
- Enable administrators to oversee and manage the entire system
- Ensure data security and privacy compliance
- Create a user-friendly and responsive interface
- Support multiple specialties and locations

2. User Roles

2.1 Patient

Primary users who seek medical consultation and book appointments with doctors.

2.2 Doctor

Medical professionals who provide consultation services and manage their appointments.

2.3 Administrator

System administrators who oversee all operations, user management, and system configurations.

3. Core Features

3.1 Authentication & Authorization

- User registration for Patients and Doctors
- Secure login/logout with JWT token authentication
- Password reset and recovery functionality
- Role-based access control (RBAC)
- Profile management and updates
- Email verification for new accounts

3.2 Patient Features

- Advanced doctor search with multiple filters (specialty, location, availability)
- View detailed doctor profiles including qualifications and experience
- Book appointments with time slot selection
- View upcoming and past appointment history
- Cancel appointments with proper notification
- Optional payment integration for consultation fees
- Download prescriptions after consultation
- Rate and review doctors based on experience
- Receive appointment reminders via email

3.3 Doctor Features

- Complete professional profile setup
- Set weekly availability schedule with time slots
- View and manage appointment requests
- Approve or reject appointment bookings
- Daily appointment calendar view
- Access patient medical history
- Upload and manage prescriptions
- View earnings dashboard and statistics
- Update consultation fees and services

3.4 Administrator Features

- Review and approve/reject doctor registrations

- Comprehensive user management system
- System-wide appointment overview and monitoring
- Analytics dashboard with key metrics
- Manage medical specialties and categories
- System configuration and settings
- Generate reports and statistics
- Handle user complaints and support tickets

4. Technical Stack

4.1 Frontend Technologies

Technology	Purpose
React 18+ with Vite	Core framework and build tool
React Router v6	Client-side routing
Redux Toolkit / Zustand	State management
Tailwind CSS + shadcn/ui	Styling and UI components
React Hook Form	Form handling and validation
Zod	Schema validation
Axios	HTTP client for API calls
date-fns / day.js	Date manipulation
Lucide React	Icon library

4.2 Backend Technologies

Technology	Purpose
Node.js + Express.js	Server framework
MongoDB + Mongoose	Database and ODM
JWT + bcrypt	Authentication and encryption
Multer / Cloudinary	File upload handling
Nodemailer	Email notifications
express-validator	Input validation
Swagger (Optional)	API documentation

5. Database Schema (MongoDB)

5.1 Users Collection

- _id: ObjectId (Auto-generated)
- name: String (Required)
- email: String (Unique, Required)
- password: String (Hashed, Required)
- role: Enum ['patient', 'doctor', 'admin']
- phone: String
- avatar: String (URL)
- createdAt: Date
- updatedAt: Date

5.2 Doctors Collection

- _id: ObjectId
- userId: ObjectId (Reference to Users)
- specialization: String
- qualifications: Array of Strings
- experience: Number (Years)
- consultationFee: Number
- address: String
- city: String
- availability: Array of Objects (day, slots)
- isApproved: Boolean
- rating: Number (0-5)
- reviewCount: Number

5.3 Appointments Collection

- _id: ObjectId
- patientId: ObjectId (Reference to Users)
- doctorId: ObjectId (Reference to Doctors)
- date: Date
- timeSlot: String
- status: Enum ['pending', 'approved', 'completed', 'cancelled']
- symptoms: String (Text area)

- prescription: String (URL to uploaded file)
- paymentStatus: Enum ['pending', 'paid']
- createdAt: Date

5.4 Reviews Collection

- _id: ObjectId
- appointmentId: ObjectId (Reference)
- patientId: ObjectId (Reference to Users)
- doctorId: ObjectId (Reference to Doctors)
- rating: Number (1-5)
- comment: String
- createdAt: Date

6. API Endpoints Structure

6.1 Authentication Routes (/api/auth)

- POST /register - User registration
- POST /login - User login
- POST /logout - User logout
- POST /forgot-password - Request password reset
- POST /reset-password - Reset password with token

6.2 User Routes (/api/users)

- GET /profile - Get user profile
- PUT /profile - Update user profile
- PUT /change-password - Change password
- GET /dashboard - Get dashboard data

6.3 Doctor Routes (/api/doctors)

- GET / - Get all doctors (with filters)
- GET /:id - Get doctor by ID
- POST /profile - Create doctor profile
- PUT /profile - Update doctor profile
- PUT /availability - Update availability schedule
- GET /appointments - Get doctor's appointments
- GET /earnings - Get earnings statistics

6.4 Appointment Routes (/api/appointments)

- POST /book - Book new appointment
- GET /patient - Get patient's appointments
- GET /doctor - Get doctor's appointments
- PUT /:id/status - Update appointment status
- DELETE /:id - Cancel appointment
- POST /:id/prescription - Upload prescription

6.5 Admin Routes (/api/admin)

- GET /doctors/pending - Get pending doctor approvals

- PUT /doctors/:id/approve - Approve doctor
- PUT /doctors/:id/reject - Reject doctor
- GET /users - Get all users
- DELETE /users/:id - Delete user
- GET /analytics - Get system analytics

7. UI/UX Requirements

7.1 Navbar Specifications

Before Login:

- Logo (left side, clickable to home)
- Menu: Home, Find Doctors, About Us, Contact, How It Works
- Right side: Login Button, Sign Up Button (highlighted)

After Login (Patient):

- Logo, Home, Find Doctors, My Appointments
- Profile Dropdown: Dashboard, My Profile, Appointment History, Settings, Logout

After Login (Doctor):

- Logo, Home
- Dashboard Menu: My Appointments, Today's Schedule, My Patients, Availability, Earnings
- Profile Dropdown: My Profile, Settings, Logout

After Login (Admin):

- Logo, Dashboard, Doctors, Patients, Appointments, Analytics
- Profile Dropdown with admin settings

7.2 Home Page Components

Section 1: Hero Section

- Headline: 'Find & Book Appointments with Best Doctors'
- Sub-headline: 'Get expert medical care at your convenience'
- Search Bar: Search by doctor name, specialty, or location
- Quick Stats: 500+ Doctors, 10,000+ Patients, 50+ Specialties, 24/7 Support
- Hero Image/Illustration
- Primary CTA: 'Find a Doctor' button
- Secondary CTA: 'Are you a Doctor? Join Us'

Section 2: How It Works

- Step 1: Find a Doctor - Search by specialty, location or name
- Step 2: Book Appointment - Choose preferred time slot
- Step 3: Get Consultation - Visit and get expert medical care

Section 3: Popular Specialties

Grid layout with specialty cards including:

- Cardiologist, Dermatologist, Pediatrician, Orthopedic
- Gynecologist, Dentist, ENT Specialist, General Physician
- Psychiatrist, Neurologist, Ophthalmologist, Urologist

Each card shows: Icon, Name, Number of available doctors

Section 4: Featured/Top Doctors

- Doctor photo and name
- Specialty and qualifications
- Experience (years)
- Rating (out of 5 stars)
- Consultation fee
- Location/City
- 'Book Appointment' and 'View Profile' buttons

Section 5: Why Choose Us

- Verified Doctors - All licensed professionals
- Easy Booking - Quick and hassle-free process
- Secure & Private - Data safety guaranteed
- 24/7 Support - Round the clock assistance

- Best Prices - Transparent, no hidden charges
- Multiple Specialties - 50+ medical specialties

Section 6: Patient Testimonials

Slider with 3 testimonial cards showing:

- Patient photo and name
- Rating (5 stars)
- Review text
- Date of review

Section 7: Statistics Counter

- 10,000+ Happy Patients
- 500+ Expert Doctors
- 50+ Specialties
- 95% Satisfaction Rate
- (Animated count-up on scroll)

Section 8: Call To Action

- For Patients: 'Ready to Get Started?' - Button: 'Find a Doctor Now'
- For Doctors: 'Are You a Doctor?' - Button: 'Register as Doctor'

Section 9: Footer

Five-column layout:

- Column 1 - About: Logo, description, social media links
- Column 2 - Quick Links: About, Find Doctors, Specialties, Blog, FAQs
- Column 3 - For Doctors: Register, Login, Dashboard, Terms
- Column 4 - Support: Contact, Help Center, Privacy, Terms, Refund Policy
- Column 5 - Contact: Address, Email, Phone, Hours (24/7)

Bottom bar: Copyright notice and payment method icons

7.3 Find Doctors Page

Filter Sidebar (Left Side):

- Search box for doctor name
- Specialty filter (checkbox list)
- Location/City (dropdown)
- Gender: Male/Female/Any
- Availability: Today/Tomorrow/This Week
- Consultation fee range slider (■0 - ■5000)
- Experience: 0-5, 5-10, 10-15, 15+ years
- Rating filter: 4+ stars, 3+ stars
- Sort by: Relevance, Rating, Experience, Fee
- Reset Filters button

Doctor Listing (Right Side):

- Results count display
- Doctor cards in list view showing:
 - - Photo with verified badge
 - - Name, specialty, qualifications
 - - Experience and rating
 - - Location and consultation fee
 - - Available slots indicator
 - - 'View Profile' and 'Book Now' buttons
- Pagination at bottom

7.4 Doctor Profile Page

Doctor Info Card:

- Large profile photo
- Name with verified badge
- Specialty and qualifications
- Years of experience
- Languages spoken
- Rating with review count
- Consultation fee
- Sticky 'Book Appointment' button

Tabbed Content:

- Tab 1 - About: Bio, education, specializations, services, awards
- Tab 2 - Availability: Weekly schedule calendar, available time slots
- Tab 3 - Location: Clinic address, map, contact number
- Tab 4 - Reviews: Rating breakdown, patient reviews with pagination
- Tab 5 - Experience: Work timeline, current and previous positions

7.5 Appointment Booking Flow

Step 1: Select Date

- Calendar view with available dates highlighted and unavailable dates disabled

Step 2: Select Time Slot

- Morning slots (9 AM - 12 PM)
- Afternoon slots (12 PM - 5 PM)
- Evening slots (5 PM - 9 PM)
- Booked slots shown as disabled

Step 3: Patient Details

- Patient name
- Age and gender
- Phone number
- Problem/symptoms (textarea)
- Booking for someone else? (checkbox)

Step 4: Confirm & Pay

- Appointment summary with all details
- Fee breakdown (consultation + platform fee)
- Payment method selection:
 - - bKash
 - - Nagad
 - - Credit/Debit Card
 - - Pay at Clinic
- 'Confirm Booking' button

Success Screen:

- Success icon/animation
- Confirmation message
- Complete appointment details
- Unique appointment ID
- Download/Print receipt option
- 'Go to Dashboard' button

8. Key Pages Structure

8.1 Public Pages (No Login Required)

- Home/Landing Page - Main entry point
- Doctor Listing Page - Browse all doctors
- Doctor Detail Page - Individual doctor profile
- Login Page - User authentication
- Register Page - New user signup
- About Us Page - Company information
- Contact Page - Contact form and info
- How It Works Page - Platform guide
- Specialties Page - All medical specialties

8.2 Patient Dashboard Pages

- Dashboard Overview - Quick stats and upcoming appointments
- My Appointments - All appointments with status
- Book Appointment - Search and book new appointment
- Appointment History - Past appointments and prescriptions
- My Profile - Personal information management
- Settings - Account and notification preferences
- Reviews - Reviews given to doctors

8.3 Doctor Dashboard Pages

- Dashboard Overview - Today's schedule and stats
- Appointment Requests - Pending approval list
- My Appointments - All appointments calendar
- Today's Schedule - Current day appointments
- My Patients - Patient history and records
- Availability Management - Set weekly schedule
- Earnings Dashboard - Financial statistics
- My Profile - Professional profile management
- Reviews - Patient feedback received
- Settings - Account preferences

8.4 Admin Dashboard Pages

- Dashboard Overview - System analytics
- Doctor Approvals - Pending doctor registrations
- Doctor Management - All doctors list
- Patient Management - All patients list
- Appointment Overview - System-wide appointments
- Analytics & Reports - Detailed statistics
- Specialty Management - Add/edit specialties
- User Management - All users administration
- System Settings - Platform configuration

9. Additional Features (Optional Enhancements)

- Real-time Notifications - Socket.io integration for instant updates
- Chat System - Direct messaging between doctor and patient
- Video Consultation - Integration with video call platforms
- Payment Gateway - Stripe/Razorpay/SSLCommerz integration
- Email Notifications - Appointment confirmations and reminders
- SMS Notifications - Text message alerts
- Multi-language Support - Bengali and English
- Dark Mode - Alternative color scheme
- Export Reports - PDF generation for appointments and prescriptions
- Calendar Integration - Google Calendar sync
- Advanced Search - AI-powered doctor recommendations
- Mobile App - React Native version
- Telemedicine - Remote consultation features
- Health Records - Digital health record storage
- Insurance Integration - Insurance verification system
- Referral System - Doctor-to-doctor referrals

10. Development Phases

Phase 1: Project Setup & Authentication (Week 1-2)

- Initialize React and Node.js projects
- Set up MongoDB database
- Implement user authentication (JWT)
- Create login and registration pages
- Set up routing and navigation
- Implement password reset functionality

Phase 2: Doctor Profile & Listing (Week 3-4)

- Create doctor profile schema
- Build doctor registration form
- Implement doctor listing page
- Add search and filter functionality
- Create doctor detail page
- Implement rating system

Phase 3: Appointment Booking System (Week 5-6)

- Design appointment schema
- Create availability management
- Build booking interface with calendar
- Implement time slot selection
- Add booking confirmation
- Set up email notifications

Phase 4: Doctor Dashboard (Week 7-8)

- Create doctor dashboard layout
- Implement appointment management
- Build schedule management interface
- Add patient history view
- Create prescription upload feature
- Implement earnings dashboard

Phase 5: Admin Panel (Week 9-10)

- Design admin dashboard
- Implement doctor approval system
- Create user management interface
- Build analytics dashboard
- Add system configuration
- Implement reporting features

Phase 6: Reviews & Additional Features (Week 11-12)

- Implement review and rating system
- Add payment integration (optional)
- Implement notifications
- Add advanced search features
- Create mobile responsive design
- Optimize performance

Phase 7: Testing & Deployment (Week 13-14)

- Comprehensive testing (unit, integration)
- Bug fixing and optimization
- Security audit
- Performance optimization
- Deploy to production server
- Set up CI/CD pipeline
- Documentation and user guides

11. Best Practices & Recommendations

11.1 Security Best Practices

- Use HTTPS for all communications
- Implement rate limiting on API endpoints
- Sanitize all user inputs to prevent SQL injection
- Use bcrypt for password hashing with appropriate salt rounds
- Implement CORS properly
- Store JWT tokens securely (HttpOnly cookies)
- Validate file uploads (type, size, content)
- Implement CSRF protection
- Regular security audits
- Keep dependencies updated

11.2 Performance Optimization

- Implement lazy loading for routes and components
- Use React.memo for expensive components
- Optimize images with proper formats and sizes
- Implement pagination for large data sets
- Use indexing in MongoDB for frequently queried fields
- Cache frequently accessed data
- Implement CDN for static assets
- Minimize bundle size with code splitting
- Use debouncing for search inputs
- Optimize API responses with field selection

11.3 Code Quality

- Follow consistent naming conventions
- Write clean, self-documenting code
- Implement proper error handling
- Use ESLint and Prettier for code formatting
- Write unit tests for critical functions
- Document API endpoints properly
- Use environment variables for configuration
- Implement proper logging

- Follow Git best practices (meaningful commits)
- Code reviews before merging

11.4 User Experience

- Provide clear loading indicators
- Show meaningful error messages
- Implement form validation with helpful feedback
- Make the interface responsive (mobile-first)
- Use consistent UI components
- Provide keyboard navigation support
- Implement proper accessibility (ARIA labels)
- Add confirmation dialogs for destructive actions
- Show success messages after actions
- Keep the interface intuitive and simple

12. Conclusion

This Doctor Appointment System is a comprehensive full-stack project that demonstrates modern web development practices and technologies. The system addresses real-world healthcare challenges by providing an efficient platform for patients to connect with doctors. The project covers all essential aspects of full-stack development including:

- Frontend development with React and modern UI libraries
- Backend API development with Node.js and Express
- Database design and management with MongoDB
- Authentication and authorization implementation
- File upload and management
- Payment integration (optional)
- Real-time features (optional)
- Responsive design and user experience
- Security best practices
- Deployment and DevOps

This project is ideal for portfolio building and demonstrates proficiency in modern full-stack development. It can be extended with additional features based on requirements and learning goals. Remember to start with core features and gradually add complexity. Focus on making each feature work properly before moving to the next phase. Good luck with your development journey!

--- End of Document ---