Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	16 February 2026
Team ID	LTVIP2026TMIDS35442
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	1.1 Registration through email and password 1.2 Password encryption using bcrypt 1.3 Phone number validation 1.4 Duplicate email check
FR-2	User Login	2.1 Login with email and password 2.2 JWT token generation 2.3 Token storage in localStorage 2.4 Session management
FR-3	Role-Based Access Control	3.1 Patient role with limited access 3.2 Doctor role with appointment management 3.3 Admin role with platform oversight 3.4 Protected routes based on role
FR-4	Doctor Discovery	4.1 List all approved doctors 4.2 Filter by specialization 4.3 View doctor profiles (qualifications, experience, fees) 4.4 Doctor cards with basic info
FR-5	Appointment Booking	5.1 Select appointment date 5.2 Select time slot 5.3 Upload medical documents 5.4 Add reason for visit 5.5 Submit booking request 5.6 Display confirmation message
FR-6	Appointment Management (Patient)	6.1 View all appointments in history 6.2 See appointment status (pending/approved/rejected/completed/cancelled) 6.3 Cancel pending appointments 6.4 View uploaded documents

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-7	Appointment Management (Doctor)	7.1 View all appointment requests 7.2 Approve pending appointments 7.3 Reject appointments with reason 7.4 Mark appointments as completed
FR-8	Doctor Profile Management	8.1 Add/update qualifications 8.2 Set consultation fees 8.3 Configure available timings (day-wise) 8.4 Add bio/description 8.5 Update contact information
FR-9	Admin Doctor Approval	9.1 View pending doctor applications 9.2 Review doctor details 9.3 Approve qualified doctors 9.4 Reject unqualified doctors
FR-10	Admin User Management	10.1 View all registered users 10.2 Filter users by type (patient/doctor) 10.3 View user details
FR-11	Admin Appointment Oversight	11.1 View all appointments 11.2 Filter by status 11.3 Filter by date
FR-12	Dashboard	12.1 Patient dashboard with upcoming appointments 12.2 Doctor dashboard with stats (pending/approved/completed) 12.3 Admin dashboard with platform metrics
FR-13	File Upload	13.1 Upload medical documents (images/PDF) 13.2 Multiple file upload support 13.3 File size limit (5MB) 13.4 View uploaded files in appointment details
FR-14	Logout	14.1 Secure logout clearing session 14.2 Redirect to home page

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Intuitive UI with clear navigation • Responsive design works on mobile, tablet, desktop • Consistent color scheme (blue/white professional theme) • Loading states for async operations • Error messages with user-friendly language • Form validation with clear feedback • Maximum 3 clicks to book appointment
NFR-2	**Security**	• Passwords hashed using bcrypt before storage • JWT tokens for authentication • Protected routes prevent unauthorized access • Role-based access control (patient/doctor/admin) • Environment variables for sensitive data • Input validation and sanitization • MongoDB injection prevention • File upload type restrictions (only images/PDFs) • File size limit (5MB)
NFR-3	**Reliability**	• Graceful error handling with user feedback • Data persistence in MongoDB Atlas • Automatic reconnection on database failure • Consistent appointment status tracking
NFR-4	**Performance**	• Page load time < 3 seconds • API response time < 500ms • Doctor list loads within 2 seconds • Supports 100+ concurrent users • Database indexing for faster queries • Optimized images and assets • Efficient MongoDB queries
NFR-5	**Availability**	• Hosted on localhost during development • MongoDB Atlas cloud database (99.9% uptime) • Application available 24/7 • Quick recovery from crashes • Graceful degradation on service failure
NFR-6	**Scalability**	• MongoDB Atlas can scale with data growth • Stateless JWT authentication allows horizontal scaling • Modular architecture (separate frontend/backend) • Indexed collections for fast queries • Can add more server instances • React components reusable for future features