

## **FINAL REPORT - DOCSPOT**

### **DocSpot: Seamless Appointment Booking for Health**

<b>Team ID</b>	LTVIP2026TMIDS35442
<b>Project Name</b>	DocSpot: Seamless Appointment Booking for Health
<b>Team Members</b>	Botla Siva Satyanarayana Perabathula Satya Sai Manikanta Satya Srithadi Prathi Adithya Durga Sri Krishna Kumar

# 1. INTRODUCTION

## 1.1 Project Overview

DocSpot is a full-stack web application designed to simplify the process of booking doctor appointments. The platform connects patients with healthcare providers through an intuitive online interface, eliminating the need for traditional phone-based appointment scheduling. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), DocSpot provides separate dashboards for patients, doctors, and administrators, each with role-specific functionalities.

The system allows patients to browse doctors by specialization, view detailed profiles including qualifications and fees, book appointments by selecting date and time, upload medical documents, and manage their bookings through a personal dashboard. Doctors can register on the platform (pending admin approval), complete their professional profiles, set available timings, and manage appointment requests by approving or rejecting them. Administrators have oversight of the entire platform, including doctor verification, user management, and appointment monitoring.

## 1.2 Purpose

The purpose of DocSpot is to address the following challenges in traditional healthcare appointment booking:

- **For Patients:** Eliminate long phone calls, waiting on hold, and limited clinic hours by providing 24/7 online booking with real-time availability and transparent doctor information.
- **For Doctors:** Reduce administrative burden by digitizing schedule management, minimize no-shows through digital tracking, and increase visibility to potential patients.
- **For Administrators:** Ensure platform integrity through doctor verification, monitor platform activity, and maintain quality control.

## 2. IDEATION PHASE

### 2.1 Problem Statement

#### Primary Problem:

Patients struggle to book doctor appointments due to phone-only systems, limited clinic hours, no visibility into doctor availability or fees, and no way to track appointments. Doctors waste time on administrative tasks, face high no-show rates, and lack digital tools to manage schedules. Administrators cannot effectively verify doctor credentials or monitor platform activity.

#### Customer Problem Statements:

PS #	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A working professional with busy schedule	Book a doctor appointment without taking time off work	I have to call multiple clinics during work hours and wait on hold	Most clinics only accept phone bookings during limited hours	Frustrated and stressed
PS-2	A patient seeking a specialist	Find a qualified doctor with good reviews	I don't know which doctors are available or their consultation fees	There's no centralized platform with doctor information	Confused and uncertain
PS-3	A parent with sick child	Get urgent pediatric appointment	I don't know which clinics have same-day availability	No real-time visibility into doctor slots	Anxious and scared
PS-4	A busy doctor	Manage my appointment schedule efficiently	I receive phone calls at all hours for bookings	My receptionist can't handle the volume	Overwhelmed and stressed
PS-5	A platform administrator	Ensure only verified doctors join	Fake practitioners might try to register	No streamlined verification process	Concerned about platform credibility

## 2.2 Empathy Map Canvas

### EMPATHY MAP 1: PATIENT USER



## EMPATHY MAP 2: DOCTOR USER



## EMPATHY MAP 3: ADMIN USER



## 2.3 Brainstorming

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

### Team Members & Roles

Name	Role
Prathi Adithya Durga Sri Krishna Kumar	Full Stack Developer (MERN)
Botla Siva Satyanarayana	Full Stack Developer
Perabathula Satya Sai Manikanta	Frontend Developer
Satya Srithadi	Backend Developer

### Step 1: Team Gathering, Collaboration, and Selecting the Problem Statement

#### Problem Statement:

To develop a seamless online doctor appointment booking platform called "DocSpot" that allows patients to easily find and book appointments with doctors, enables doctors to efficiently manage their schedules, and provides administrators with tools to oversee the entire healthcare ecosystem - ultimately improving healthcare accessibility and reducing the hassle of traditional appointment booking.

#### Motivation:

Booking a doctor's appointment traditionally involves long phone calls, waiting on hold, playing phone tag with busy receptionists, and limited visibility into doctor availability. Patients struggle to find the right doctor that matches their needs, while doctors face challenges in managing their schedules and reducing no-shows. DocSpot bridges this gap by providing a user-friendly online platform with real-time availability, doctor profiles, appointment management, and administrative oversight - making healthcare booking simple, transparent, and efficient for everyone.

## **Step 2: Brainstorming, Idea Listing, and Grouping**

### **Initial Ideas:**

- User registration and login system with email/password
- JWT-based authentication for security
- Browse doctors with filters (specialization, location, fees)
- Real-time doctor availability calendar
- One-click appointment booking
- Upload medical documents during booking
- Email notifications for confirmations and reminders
- Cancel and reschedule appointments
- View booking history in user dashboard
- Doctor registration with approval workflow
- Doctor profile management (qualifications, experience, fees)
- Doctor dashboard to manage appointments
- Approve/reject appointment requests
- Admin dashboard to monitor platform
- Admin approval for new doctor registrations
- User management for admins
- View all appointments across platform
- Responsive design for mobile and desktop
- Notification system for all users
- Secure password encryption with bcrypt
- MongoDB database for scalable data storage
- RESTful APIs for frontend-backend communication
- File upload system for medical documents
- Role-based access control (User/Doctor/Admin)
- Beautiful UI with Bootstrap and custom CSS
- Landing page with platform features
- About and Contact pages
- 404 error page handling
- Loading states and error handling
- Toast notifications for user feedback

**Grouped into Modules:**

Module	Features Included
<b>1. Authentication Module</b>	User Registration, Login, JWT Authentication, Password Encryption, Role-based Access
<b>2. Doctor Discovery Module</b>	Browse Doctors, Search Filters (Specialization, Location, Fees), Doctor Profiles, Availability Calendar
<b>3. Appointment Booking Module</b>	Book Appointment, Select Date/Time, Upload Documents, Instant Confirmation
<b>4. Appointment Management Module</b>	View Appointments, Cancel Appointment, Reschedule, Status Tracking (Pending/Approved/Completed)
<b>5. Notification Module</b>	Email Notifications, In-App Alerts, Appointment Reminders, Status Updates
<b>6. Doctor Dashboard Module</b>	Manage Schedule, View Requests, Approve/Reject Appointments, Profile Editing
<b>7. Admin Dashboard Module</b>	User Management, Doctor Approval, Platform Monitoring, View All Appointments
<b>8. UI/UX Module</b>	Responsive Design, Landing Page, Navigation Bar, Footer, Loading States, Error Handling
<b>9. Profile Management Module</b>	User Profile, Edit Details, View History, Notification Center
<b>10. Database Module</b>	MongoDB Schemas (Users, Doctors, Appointments), Data Relationships, Indexes

### Step 3: Idea Prioritization (Final Version)

Feature / Module	Importance	Feasibility	Notes
<b>Authentication Module</b>	High	High	User registration and login with JWT – core requirement
<b>Doctor Discovery Module</b>	High	High	Browse and filter doctors – essential for user experience
<b>Appointment Booking Module</b>	High	High	Book appointments with date/time – primary functionality
<b>Appointment Management Module</b>	High	High	View, cancel, reschedule – user control
<b>Notification Module</b>	High	Medium	Email confirmations and reminders – improves engagement
<b>Doctor Dashboard Module</b>	High	High	Doctors manage their appointments – critical for workflow
<b>Admin Dashboard Module</b>	High	High	Platform oversight and doctor approval – ensures trust
<b>Profile Management Module</b>	Medium	High	User profile and history – enhances personalization
<b>UI/UX Module</b>	High	High	Responsive, clean interface – key to user adoption
<b>Database Module</b>	High	High	MongoDB with proper schemas – foundation of the system

### 3. REQUIREMENT ANALYSIS

#### 3.1 Customer Journey Map

##### PATIENT JOURNEY MAP

STAGES	AWARENESS	CONSIDERATION	REGISTRATION	BOOKING	POST-BOOKING & APPOINTMENT DAY
OBJECTIVES	User becomes aware of DocSoot and visits the platform	User browses doctors and evaluates if DocSoot meets their needs	User creates an account to book appointment	User books an appointment with selected doctor	User tracks appointment status and attends consultation
NEEDS	Find a reliable way to book doctor appointments online without phone calls	See available doctors, view qualifications, understand fees, check if specialists are available	Quick and simple registration process with email and password	Select convenient date and time, add reason for visit, upload documents if needed	Know if appointment is approved, remember appointment details, visit clinic
FEELINGS	Anxious about health, curious about options, hopeful to find solution	Hopeful to find right doctor, confused while comparing multiple doctors	Relieved when form is simple, frustrated if form has errors	Satisfied when booking completes, relieved to secure appointment	Happy when approved, anxious before visit, satisfied after consultation
BARRIERS	Too many options online, unsure which platform to trust, no prior knowledge of DocSoot	Hard to compare doctors, no patient reviews, specialization filter may not show all options	Long registration forms, password requirements unclear, duplicate email errors	Selected time slot may be unavailable, form errors, document upload fails	No status updates, forget appointment details, long wait at clinic
ADD NOTES OR COMMENTS	<ul style="list-style-type: none"> <li>• User searches Google for doctors</li> <li>• Friends recommend DocSoot</li> <li>• Lands on DocSoot homepage</li> </ul>	<ul style="list-style-type: none"> <li>• User sees Doctors List page</li> <li>• Filters by specialization</li> <li>• Views doctor profiles with qualifications, experience, fees</li> </ul>	<ul style="list-style-type: none"> <li>• Registration form asks name, email, phone, password</li> <li>• Password encrypted with bcrypt</li> <li>• Duplicate email check</li> <li>• Success message appears</li> </ul>	<ul style="list-style-type: none"> <li>• Date picker for future dates</li> <li>• Time slots available</li> <li>• Reason for visit text field</li> <li>• Document upload (images/PDF, 5MB max)</li> <li>• Confirmation message after booking</li> </ul>	<ul style="list-style-type: none"> <li>• My Appointments page shows all bookings</li> <li>• Status badges: pending, approved, rejected, completed, cancelled</li> <li>• Cancel button for pending appointments</li> <li>• User visits clinic on scheduled date</li> </ul>

## DOCTOR JOURNEY MAP

STAGES	REGISTRATION	PROFILE SETUP	DAILY OPERATIONS	CONSULTATION	FOLLOW-UP
OBJECTIVES	Doctor signs up to join the platform	Doctor completes profile with professional details	Doctor manages daily schedule and appointment requests	Doctor meets patient and provides care	Doctor completes appointment record
NEEDS	Create account to list practice and reach patients	Add qualifications, experience, consultation fees, available timings	See pending appointments, approve suitable slots, reject conflicts	Access patient information, provide consultation	Mark appointment as completed
FEELINGS	Excited to join, patient while waiting for approval	Productive while setting up, frustrated if form is complex	Organized with dashboard, in control of schedule	Focused on patient care, tired after long day	Satisfied with day's work
BARRIERS	Unsure if platform is trustworthy, registration form asks many details	Too many fields to fill, unsure what to include, time-consuming	Hard to see all requests at once, manual approval time-consuming	No access to patient history, interruptions during consultation	Forgetting to mark completed, no follow-up system
ADD NOTES OR COMMENTS	<ul style="list-style-type: none"> <li>• Doctor registers with name, email, phone, specialization</li> <li>• Account created with "pending" status</li> <li>• Admin must approve before access</li> </ul>	<ul style="list-style-type: none"> <li>• Profile form includes qualifications, fees</li> <li>• Day-wise timing selector</li> <li>• Bio and contact info</li> <li>• Profile visible after approval</li> </ul>	<ul style="list-style-type: none"> <li>• Dashboard shows pending/approved/completed counts</li> <li>• Appointments page lists all requests</li> <li>• Patient documents visible</li> <li>• One-click approve/reject</li> </ul>	<ul style="list-style-type: none"> <li>• Doctor views patient name and reason</li> <li>• Uploaded documents available</li> <li>• In-person consultation</li> </ul>	<ul style="list-style-type: none"> <li>• Click "Complete" button after consultation</li> <li>• Status updates to "completed"</li> </ul>

## ADMIN JOURNEY MAP

STAGES	LOGIN	MONITOR DASHBOARD	DOCTOR VERIFICATION	USER MANAGEMENT	APPOINTMENT OVERSIGHT
OBJECTIVES	Admin accesses the platform management panel	Admin views platform statistics and pending items	Admin reviews and approves/rejects doctor applications	Admin monitors all platform users	Admin monitors all appointments on platform
NEEDS	Secure login to access admin features	See overview of users, doctors, appointments, pending approvals	Verify doctor credentials, ensure only qualified doctors join	View all registered patients and doctors	View all bookings, track platform activity
FEELINGS	Ready to manage platform	Informed about platform health	Productive when approving, stressed if fake profiles found	Organized with complete view	In control with complete oversight
BARRIERS	Forgot password, no reset option	Stats not real-time, hard to track trends	Manual verification time-consuming, cannot verify credentials directly	No search functionality, cannot filter users	Too many appointments to track, no filtering
ADD NOTES OR COMMENTS	<ul style="list-style-type: none"> <li>• Admin login with email/password</li> <li>• Redirected to admin dashboard</li> <li>• No password reset currently</li> </ul>	<ul style="list-style-type: none"> <li>• Dashboard shows total users, total doctors, pending doctors, total appointments</li> <li>• Quick access to manage sections</li> </ul>	<ul style="list-style-type: none"> <li>• Pending doctors list</li> <li>• View complete doctor profile</li> <li>• Approve button updates to "approved"</li> <li>• Reject button updates to "rejected"</li> </ul>	<ul style="list-style-type: none"> <li>• Users page lists all registered users</li> <li>• Shows name, email, phone, role</li> <li>• Can filter by user type</li> </ul>	<ul style="list-style-type: none"> <li>• Appointments page shows all bookings</li> <li>• Filter by status</li> <li>• Filter by date</li> <li>• View patient and doctor details</li> </ul>

### 3.2 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-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

<b>FR No.</b>	<b>Functional Requirement (Epic)</b>	<b>Sub Requirement (Story / Sub-Task)</b>
		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	<b>Usability</b>	<ul style="list-style-type: none"> <li>Intuitive UI with clear navigation</li> <li>Responsive design works on mobile, tablet, desktop</li> <li>Consistent color scheme (blue/white professional theme)</li> <li>Loading states for async operations</li> <li>Error messages with user-friendly language</li> <li>Form validation with clear feedback</li> <li>Maximum 3 clicks to book appointment</li> </ul>
NFR-2	<b>Security</b>	<ul style="list-style-type: none"> <li>Passwords hashed using bcrypt before storage</li> <li>JWT tokens for authentication</li> <li>Protected routes prevent unauthorized access</li> <li>Role-based access control (patient/doctor/admin)</li> <li>Environment variables for sensitive data</li> <li>Input validation and sanitization</li> <li>MongoDB injection prevention</li> <li>File upload type restrictions (only images/PDFs)</li> <li>File size limit (5MB)</li> </ul>
NFR-3	<b>Reliability</b>	<ul style="list-style-type: none"> <li>Graceful error handling with user feedback</li> <li>Data persistence in MongoDB Atlas</li> <li>Automatic reconnection on database failure</li> <li>Consistent appointment status tracking</li> </ul>
NFR-4	<b>Performance</b>	<ul style="list-style-type: none"> <li>Page load time &lt; 3 seconds</li> <li>API response time &lt; 500ms</li> <li>Doctor list loads within 2 seconds</li> <li>Supports 100+ concurrent users</li> <li>Database indexing for faster queries</li> <li>Optimized images and assets</li> <li>Efficient MongoDB queries</li> </ul>
NFR-5	<b>Availability</b>	<ul style="list-style-type: none"> <li>Hosted on localhost during development</li> <li>MongoDB Atlas cloud database (99.9% uptime)</li> <li>Application available 24/7</li> <li>Quick recovery from crashes</li> <li>Graceful degradation on service failure</li> </ul>
NFR-6	<b>Scalability</b>	<ul style="list-style-type: none"> <li>MongoDB Atlas can scale with data growth</li> <li>Stateless JWT authentication allows horizontal scaling</li> <li>Modular architecture (separate frontend/backend)</li> <li>Indexed collections for fast queries</li> <li>Can add more server instances</li> <li>React components reusable for future features</li> </ul>

### 3.3 Data Flow Diagram

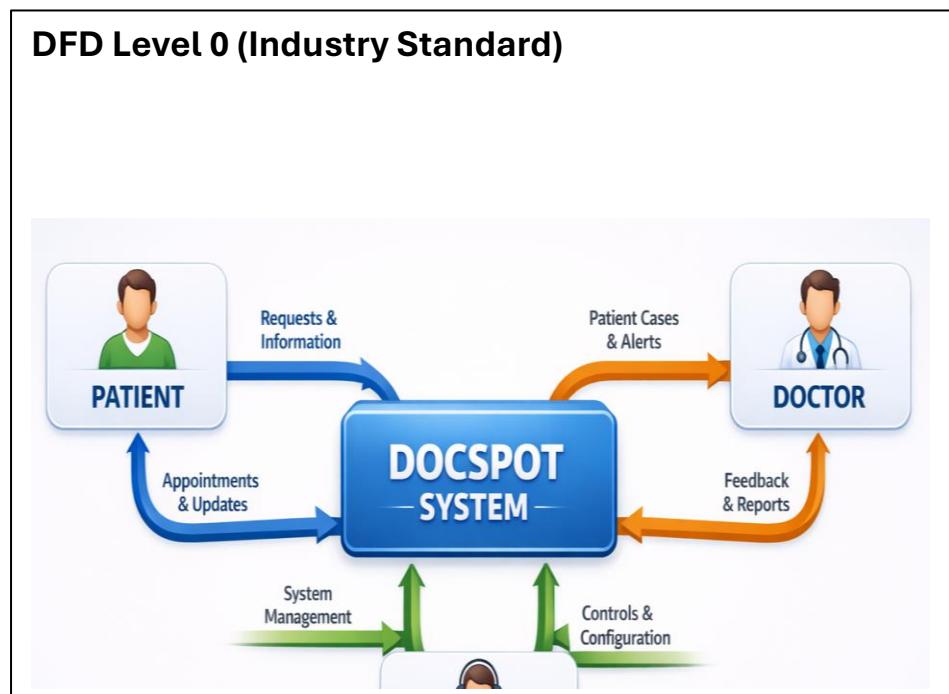
Date	16 February 2026
Team ID	LTVIP2026TMIDS35442
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	4 Marks

### Data Flow Diagrams:

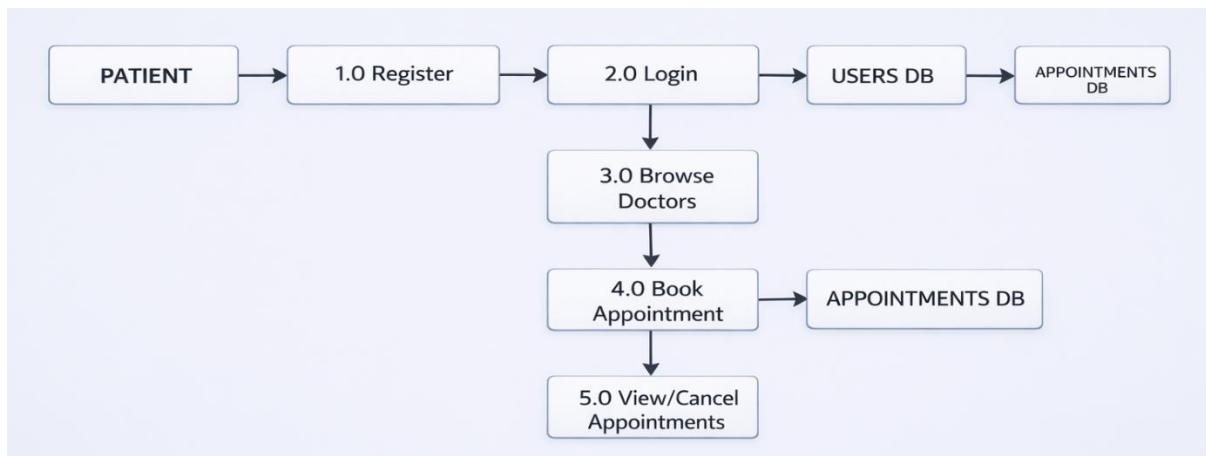
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

### External Entities:

1. Patient - Books and manages appointments
2. Doctor - Manages schedule and appointments
3. Admin - Verifies doctors and monitors platform



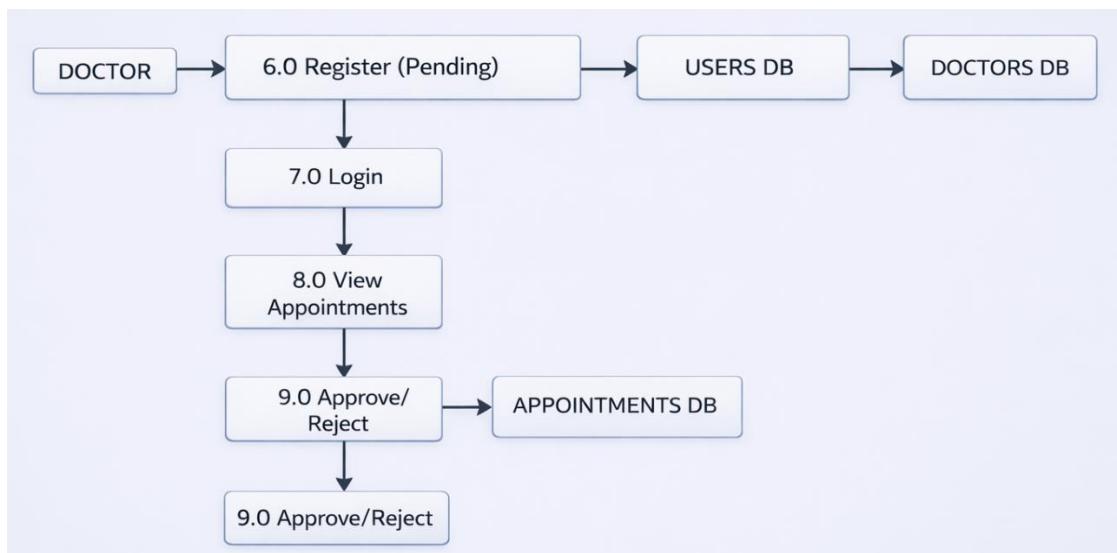
### Level 1 DFD - Patient Processes



### Processes:

1. 1.0 Register - Patient creates account with email/password
2. 2.0 Login - Patient authenticates using credentials
3. 3.0 Browse Doctors - Search and filter doctors by specialization
4. 4.0 Book Appointment - Select date/time and book
5. 5.0 View/Cancel Appointments - Manage existing bookings

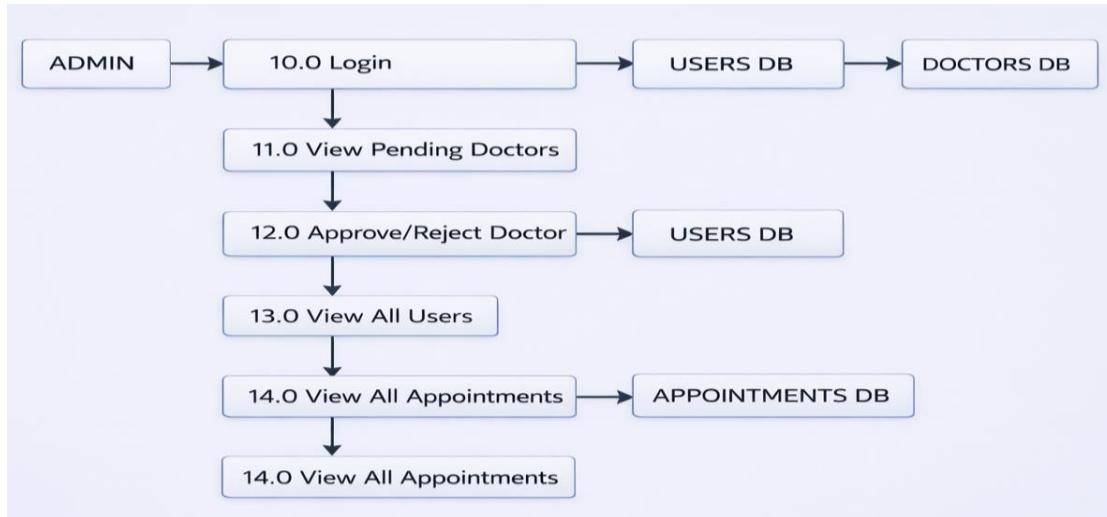
### Level 1 DFD - Doctor Processes



### Processes:

6. 6.0 Register (Pending) - Doctor applies with pending status
7. 7.0 Login - Doctor authenticates
8. 8.0 View Appointments - See all appointment requests
9. 9.0 Approve/Reject - Update appointment status

### **Level 1 DFD - Admin Processes**



### **Processes:**

10. 10.0 Login - Admin authenticates
11. 11.0 View Pending Doctors - See doctor applications
12. 12.0 Approve/Reject Doctor - Update doctor status
13. 13.0 View All Users - Monitor platform users
14. 14.0 View All Appointments - Overview of all bookings

### **User Stories**

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Patient	Registration	US-P-01	As a patient, I can register using my email, password, and phone number	I can access my dashboard after registration	High	Sprint-1
	Login	US-P-02	As a patient, I can log in using my email and password	I am redirected to my dashboard	High	Sprint-1
	Browse Doctors	US-P-03	As a patient, I can view all approved doctors with their specializations	List of doctors displays with names	High	Sprint-1
		US-P-04	As a patient, I can filter doctors by specialization	Only doctors of selected specialty are shown	High	Sprint-1
		US-P-05	As a patient, I can view doctor details including fees and experience	Doctor profile shows complete information	High	Sprint-1
	Book Appointment	US-P-06	As a patient, I can select a date and time to book an appointment	Date picker and time slots are available	High	Sprint-1
		US-P-07	As a patient, I can upload medical documents while booking	Files can be uploaded and attached to appointment	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
		US-P-08	As a patient, I receive confirmation after booking	Success message with appointment details appears	High	Sprint-1
	Manage Appointments	US-P-09	As a patient, I can view all my appointments in history	My Appointments page shows all bookings	High	Sprint-1
		US-P-10	As a patient, I can see appointment status (pending/approved/rejected)	Status badges clearly indicate current state	High	Sprint-1
		US-P-11	As a patient, I can cancel my pending appointment	Appointment status changes to "cancelled"	High	Sprint-1
	Profile	US-P-12	As a patient, I can update my profile information	Changes are saved successfully	Medium	Sprint-1
	Logout	US-P-13	As a patient, I can securely logout	Session ends and I am redirected to home	High	Sprint-1
Doctor	Registration	US-D-01	As a doctor, I can register with my details (name, email, phone, specialization)	Account is created with "pending" status	High	Sprint-1
	Login	US-D-02	As a doctor, I can log in using my email and password	I am redirected to doctor	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
				dashboard		
	Profile Management	US-D-03	As a doctor, I can complete my profile with qualifications and experience	Profile information is saved	High	Sprint-1
		US-D-04	As a doctor, I can set my consultation fees	Fees are displayed in my profile	High	Sprint-1
		US-D-05	As a doctor, I can set my available timings for each day of week	Timings are saved and displayed	High	Sprint-1
	Appointment Management	US-D-06	As a doctor, I can view all appointment requests	List of appointments with patient details is shown	High	Sprint-1
		US-D-07	As a doctor, I can approve pending appointments	Status updates to "approved" and patient is notified	High	Sprint-1
		US-D-08	As a doctor, I can reject appointments I cannot accommodate	Status updates to "rejected"	High	Sprint-1
		US-D-09	As a doctor, I can mark completed appointments	Status updates to "completed"	Medium	Sprint-1
	Dashboard	US-D-10	As a doctor, I can see counts of pending/approved/completed appointments	Dashboard shows statistics	Medium	Sprint-1

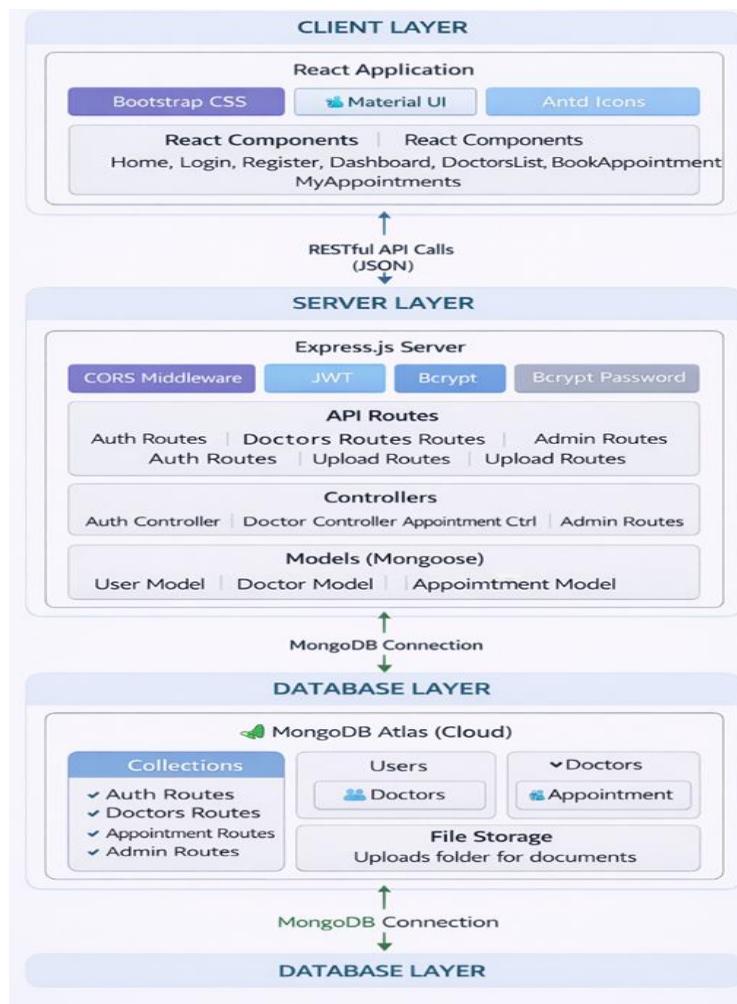
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
	Logout	US-D-11	As a doctor, I can securely logout	Session ends and I am redirected to home	High	Sprint-1
Admin	Login	US-A-01	As an admin, I can log in with my credentials	I am redirected to admin dashboard	High	Sprint-1
	Doctor Management	US-A-02	As an admin, I can view all pending doctor applications	List of pending doctors is displayed	High	Sprint-1
		US-A-03	As an admin, I can view complete doctor details for verification	Doctor profile with all information is shown	High	Sprint-1
		US-A-04	As an admin, I can approve qualified doctors	Doctor status updates to "approved"	High	Sprint-1
		US-A-05	As an admin, I can reject unqualified doctors	Doctor status updates to "rejected"	High	Sprint-1
	User Management	US-A-06	As an admin, I can view all registered users	List of patients and doctors is displayed	Medium	Sprint-1
	Appointment Oversight	US-A-07	As an admin, I can view all appointments on the platform	Complete appointment list is displayed	Medium	Sprint-1
		US-A-08	As an admin, I can filter appointments by status	Filtered results are shown	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
	Dashboard	US-A-09	As an admin, I can see platform statistics (total users, doctors, appointments)	Dashboard shows key metrics	High	Sprint-1
		US-A-10	As an admin, I can see pending approvals count	Dashboard shows pending doctors count	High	Sprint-1
	Logout	US-A-11	As an admin, I can securely logout	Session ends and I am redirected to home	High	Sprint-1

### 3.4 Technology Stack

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

## Technical Architecture:



## Data Flow:

1. User interacts with React frontend
2. Axios sends HTTP requests to Express backend
3. Express routes handle requests via controllers
4. Controllers interact with MongoDB via Mongoose models

**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	Web application interface for patients, doctors, and admin	React.js 18.2.0
2.	UI Component Libraries	Styling and UI components for better user experience	Bootstrap 5.3, React-Bootstrap 2.10, MDB React UI Kit 8.0
3.	HTTP Client	Communication between frontend and backend	Axios 1.6
4.	Application Logic - Backend	Server-side logic, API endpoints, business logic	Node.js 22.18, Express.js 4.18
5.	Authentication	User authentication and authorization	JWT (jsonwebtoken 9.0), Bcryptjs 2.4
6.	Database	Primary data storage	MongoDB 7.5 (Mongoose ODM)
7.	Cloud Database	Hosted database service	MongoDB Atlas
8.	File Storage	Storage for uploaded medical documents	Local filesystem (uploads folder) with Multer 1.4
9.	Password Encryption	Secure password hashing	Bcryptjs 2.4
10.	Environment Configuration	Manage environment variables	Dotenv 16.3
11.	CORS	Cross-origin resource sharing	CORS 2.8
12.	Development Tools	Hot reloading for development	Nodemon 3.0

S.No	Component	Description	Technology
13.	Date/Time Handling	Parse and format dates	Moment.js
14.	Icons	Icon library for UI	React Icons
15.	Infrastructure (Server)	Application deployment	Localhost (Development)

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	All frameworks used are open-source	React.js, Node.js, Express.js, MongoDB, Bootstrap, JWT, Bcrypt
2.	Security Implementations	<ul style="list-style-type: none"> <li>• Password hashing with bcrypt</li> <li>• JWT tokens for authentication</li> <li>• Protected routes with middleware</li> <li>• Role-based access control (Patient/Doctor/Admin)</li> <li>• Environment variables for secrets</li> <li>• Input validation and sanitization</li> <li>• CORS configuration</li> <li>• File upload type restrictions</li> </ul>	Bcryptjs 2.4, JWT, Express Middleware, CORS, Multer
3.	Scalable Architecture	<ul style="list-style-type: none"> <li>• Client-server architecture with separation of concerns</li> <li>• Stateless JWT authentication allows horizontal scaling</li> <li>• Modular code structure (controllers, models, routes)</li> <li>• MongoDB Atlas can scale with data growth</li> <li>• Indexed database collections for performance</li> <li>• React components reusable and modular</li> </ul>	MERN Stack (MongoDB, Express, React, Node.js)
4.	Availability	<ul style="list-style-type: none"> <li>• MongoDB Atlas provides 99.9% uptime</li> <li>• Application available 24/7 on localhost</li> <li>• Graceful error handling prevents crashes</li> <li>• Database connection retry on failure</li> </ul>	MongoDB Atlas, Express error handling
5.	Performance	<ul style="list-style-type: none"> <li>• Database indexing for faster queries</li> <li>• JWT tokens reduce database lookups</li> <li>• Efficient MongoDB queries with</li> </ul>	MongoDB indexes, JWT, React virtual DOM

S.No	Characteristics	Description	Technology
		Mongoose • React virtual DOM for fast rendering • API response time < 500ms	

## References:

- <https://reactjs.org/>
- <https://nodejs.org/>
- <https://expressjs.com/>
- <https://www.mongodb.com/atlas>
- <https://mongoosejs.com/>
- <https://jwt.io/>
- <https://github.com/axios/axios>
- <https://react-bootstrap.github.io/>
- <https://mui.com/>
- <https://ant.design/>

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

Date	16 February 2026
Team ID	LTVIP2026TMIDS35442
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	2 Marks

### **Problem – Solution Fit:**

The Problem-Solution Fit simply means that we have identified real problems faced by patients, doctors, and admins in the healthcare appointment booking ecosystem, and the DocSpot platform we have built actually solves these problems effectively. It helps our team identify behavioral patterns of all three user types and recognize what solutions work and why.

### **Purpose:**

- ❑ Solve complex problems in healthcare appointment booking by providing a digital platform that fits how patients, doctors, and admins naturally work.
- ❑ Succeed faster by using existing behaviors—patients already search online, doctors already manage schedules, admins already verify—and simply making these processes digital and efficient.
- ❑ Sharpen communication with the right messaging: "Book doctors 24/7", "No more phone calls", "Manage appointments digitally", "Verified doctors only".
- ❑ Increase trust by solving frequent annoyances—long phone calls, no-show appointments, fake doctor listings—and building a reliable platform for healthcare access.
- ❑ Improve the existing situation where patients struggle to book, doctors waste time on admin, and admins lack oversight—by providing a seamless, transparent, and efficient system for everyone.

## Problem-Solution fit canvas 2.0

 AMALTAMA



Define CS, fit into AS

Focus on J&P, tap into BE, understand

Define CS, fit into CL

Explore AS,

Focus on J&P, tap into BE, understand

Explore AS, differentiate

## 4.2 Proposed Solution

Date	16 February 2026
Team ID	LTVIP2026TMIDS35442
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	2 Marks

### Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Patients struggle to book doctor appointments due to phone-only systems, limited clinic hours, no visibility into doctor availability or fees, and no way to track appointments. Doctors waste time on administrative tasks, face high no-show rates, and lack digital tools to manage schedules. Admins cannot effectively verify doctor credentials or monitor platform activity.</p>
2.	Idea / Solution description	<p>DocSpot is a web-based doctor appointment booking platform with three user roles:</p> <p><b>Patients</b> can register, browse doctors by specialization, view profiles with qualifications and fees, book appointments by selecting date/time, upload medical documents, and manage bookings with status tracking.</p> <p><b>Doctors</b> can register (pending admin approval), complete profiles with qualifications and timings, and approve/reject appointment requests.</p> <p><b>Admins</b> can verify and approve/reject doctor applications, view all users, and monitor all appointments. Built using MERN stack (MongoDB, Express, React, Node.js) with JWT authentication.</p>
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> <li>• Complete role-based system with separate dashboards for patients, doctors, and admins</li> <li>• Doctor approval workflow ensuring only verified doctors join the platform</li> <li>• Document upload for medical records during booking</li> <li>• Real-time status tracking with pending/approved/rejected/completed/cancelled badges</li> <li>• Filter by specialization for easy doctor discovery</li> <li>• Day-wise timing configuration for doctors</li> </ul>

		<ul style="list-style-type: none"> <li>• JWT-based secure authentication with role-based access control</li> </ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> <li>• <b>Doctors:</b> Reduce administrative burden by 50%; eliminate no-shows with digital tracking; focus more on patient care; reach new patients through online presence</li> <li>• <b>Community:</b> Improves healthcare accessibility by digitizing appointment booking; reduces stress for patients seeking medical care; creates trust through verified doctor system</li> </ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> <li>• <b>Free for patients</b> - Always free to use</li> <li>• <b>Free for doctors</b> - Currently free during pilot phase</li> <li>• <b>Future revenue streams</b> (not implemented yet): Subscription model for doctors (monthly/yearly), Featured/priority listings for doctors, Commission per booking (optional)</li> </ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"> <li>• <b>Technical scalability:</b> MongoDB Atlas can scale with data growth; stateless JWT authentication allows horizontal scaling; modular MERN architecture supports adding more server instances; indexed collections ensure fast queries even with millions of records</li> <li>• <b>Geographic scalability:</b> Can add more cities and regions easily; specialization filters can be expanded</li> <li>• <b>Feature scalability:</b> New features can be added as separate modules without disrupting existing functionality</li> <li>• <b>User scalability:</b> Currently supports 100+ concurrent users; can scale to thousands with cloud infrastructure</li> </ul>

### 4.3 Solution Architecture

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

#### Solution Architecture:

DocSpot follows a three-tier client-server architecture with clear separation between presentation, application, and data layers.

#### Architecture Overview

Layer	Component	Description
Frontend	React.js	User interface for Patients, Doctors, and Admin. Communicates with backend via RESTful APIs using Axios.
Backend	Express.js	Handles business logic, authentication, request processing, and API routing with JWT-based security.
Database	MongoDB Atlas	Cloud NoSQL database storing users, doctor profiles, and appointments using Mongoose ODM.

#### Key Components

##### Frontend (React.js)

- Components:** Modular UI for each user role (patient, doctor, admin)
- React Router:** Client-side routing with protected routes based on user roles
- Axios:** HTTP client for API communication with token attachment
- Context API:** Global state management for authentication
- Bootstrap:** Responsive UI components

##### Backend (Express.js)

- Middleware:** CORS, JSON parsing, JWT authentication, Multer for file uploads
- Routes:** API endpoints for auth, doctors, appointments, admin
- Controllers:** Business logic for each entity

- **Models:** Mongoose schemas for Users, Doctors, Appointments
- **JWT:** Token-based authentication with role-based access control

### Database (MongoDB Atlas)

- **Users Collection:** Stores user credentials, roles, notifications
- **Doctors Collection:** Stores doctor profiles, qualifications, fees, timings
- **Appointments Collection:** Stores booking details, status, document references
- **Indexes:** Optimized queries on email, specialization, status, dates

### Authentication Flow

1. User registers → Password hashed with bcrypt
2. User logs in → JWT token generated
3. Token stored in localStorage → Sent in Authorization header
4. Middleware verifies token → Grants access based on role

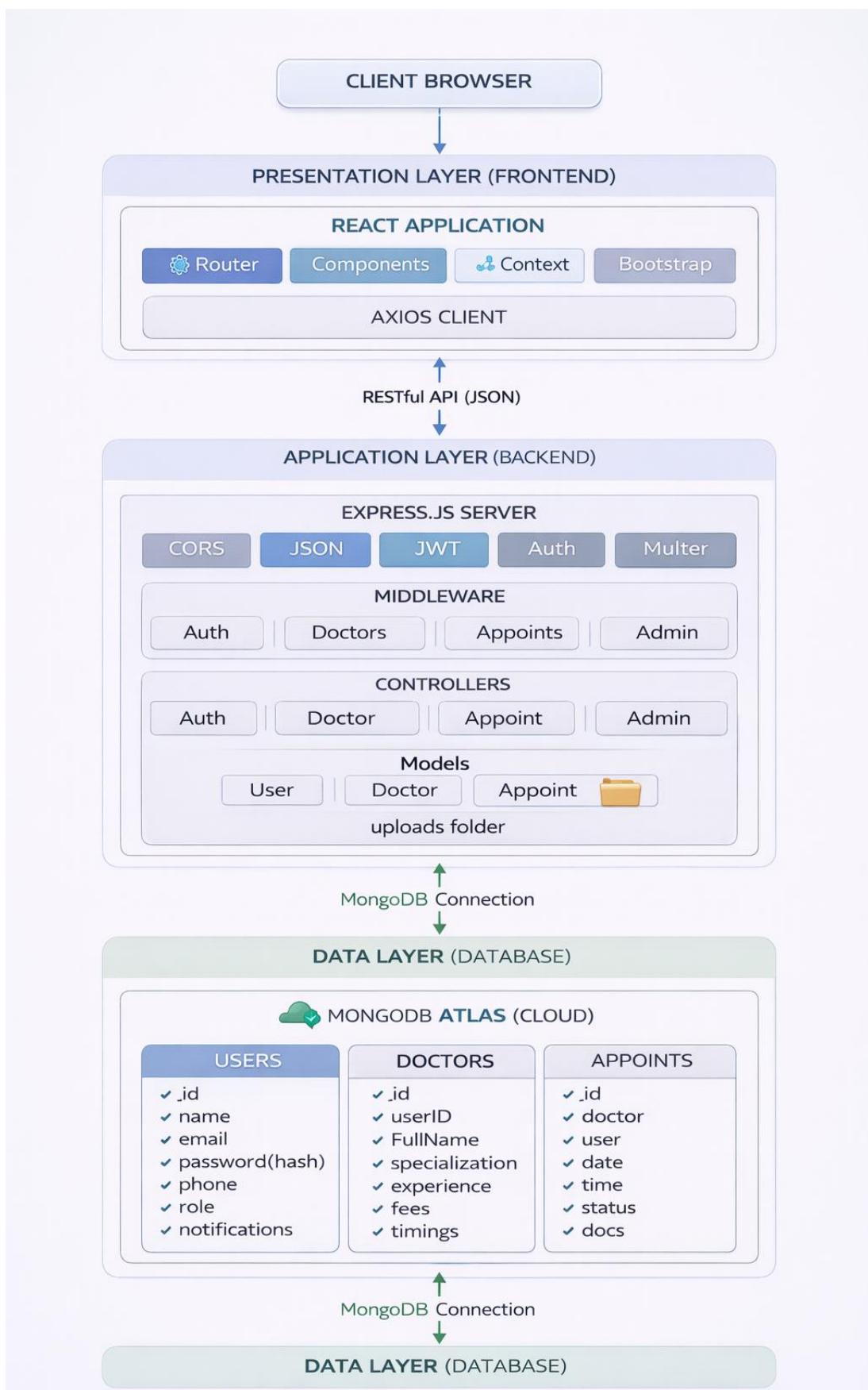
### File Upload Flow

1. Multer processes multipart form data
2. Files validated (type/size) and stored in uploads folder
3. File paths saved in appointment documents
4. Express serves static files for access

### Data Flow

1. User action → React component → Axios request
2. Express middleware → Route handler → Controller
3. Controller → Mongoose model → MongoDB
4. Response → React component → UI update

## Example - Solution Architecture Diagram:



## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

#### Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	16 February 2026
Team ID	LTVIP2026TMIDS35442
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	5 Marks

#### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Environment Setup	USN-01	Set up MERN project structure (backend + frontend folders), install dependencies (Express, Mongoose, React, Bootstrap)	2	High	Adithya, Satya
Sprint-1	Database Setup	USN-02	Configure MongoDB Atlas connection, create database models (User, Doctor, Appointment)	3	High	Siva, Satya Sai
Sprint-1	Authentication Backend	USN-03	Implement user registration and login APIs with JWT and bcrypt password hashing	3	High	Adithya, Siva
Sprint-1	Authentication Frontend	USN-04	Create Login and Register pages with form validation and token storage	3	High	Satya Sai, Satya
Sprint-2	Doctor Listing	USN-05	Create backend APIs to fetch approved doctors with filters by specialization	3	High	Adithya, Siva

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Doctor Discovery UI	USN-06	Build Doctors List page with filter dropdown and doctor cards showing basic info	3	High	Satya Sai, Satya
Sprint-2	Doctor Profile	USN-07	Create Doctor Profile page displaying qualifications, experience, fees	2	High	Satya, Satya Sai
Sprint-2	Appointment Booking Backend	USN-08	Implement appointment booking API with date, time slot, reason, document upload	3	High	Adithya, Siva
Sprint-2	Appointment Booking UI	USN-09	Build Booking form with date picker, time slots, reason field, document upload	3	High	Satya Sai, Satya
Sprint-3	Patient Dashboard	USN-10	Create Patient Dashboard showing upcoming appointments and quick stats	2	High	Adithya, Satya Sai
Sprint-3	My Appointments Page	USN-11	Build My Appointments page with status badges (pending/approved/rejected) and cancel option	3	High	Satya, Satya Sai
Sprint-3	Doctor Dashboard	USN-12	Create Doctor Dashboard with pending/approved/completed counts and quick access	2	High	Adithya, Satya Sai
Sprint-3	Doctor Appointments Management	USN-13	Build Doctor Appointments page to view requests with approve/reject buttons	3	High	Satya Sai, Siva
Sprint-3	Doctor Profile Management	USN-14	Create Doctor Profile edit page with qualifications, fees, day-wise timings	3	Medium	Adithya, Satya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
<b>Sprint-4</b>	Admin Dashboard	USN-15	Build Admin Dashboard with platform statistics (users, doctors, appointments)	2	High	Siva, Satya
<b>Sprint-4</b>	Doctor Approval	USN-16	Create Pending Doctors list for admin with approve/reject functionality	3	High	Adithya, Satya Sai
<b>Sprint-4</b>	User Management	USN-17	Build Manage Users page listing all patients and doctors with filters	2	Medium	Satya, Satya Sai
<b>Sprint-4</b>	Appointment Oversight	USN-18	Create Manage Appointments page for admin with status and date filters	2	Medium	Adithya, Satya
<b>Sprint-4</b>	UI Polish & Testing	USN-19	Implement consistent blue/white theme, responsive design, loading states, error handling	3	High	All Members

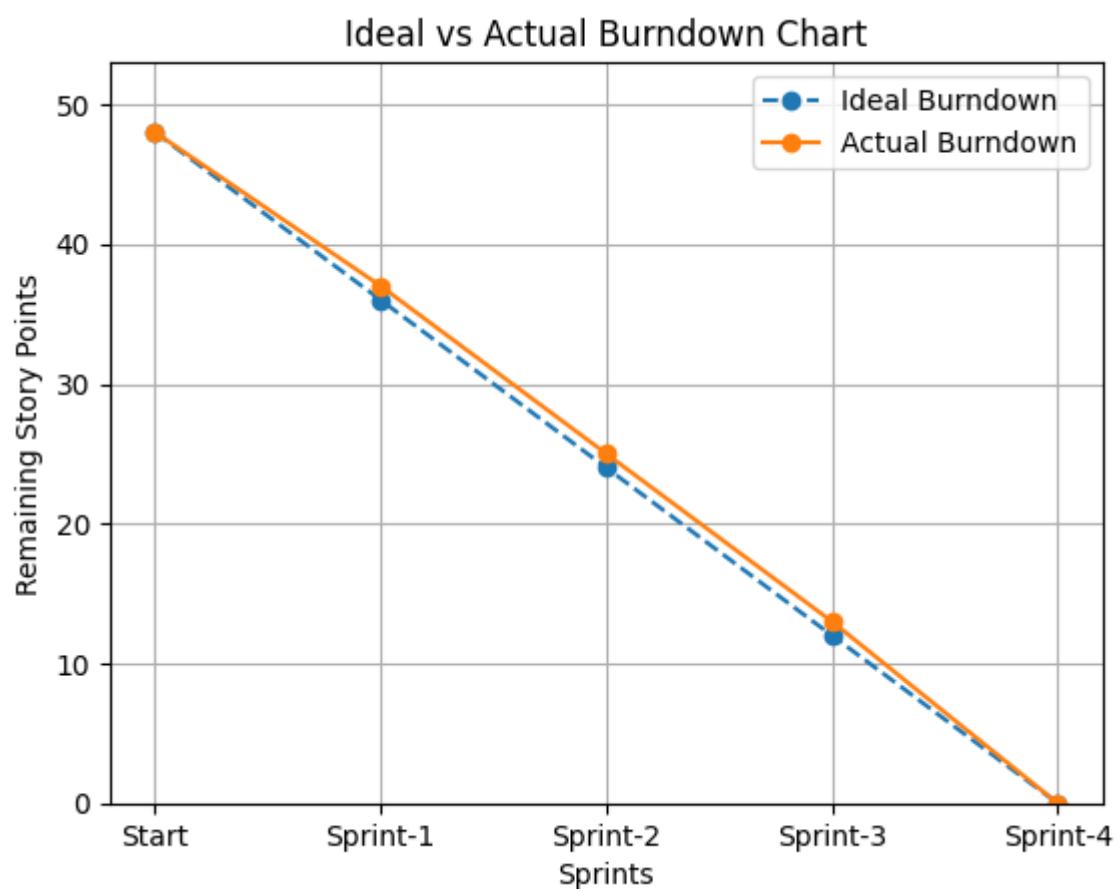
### Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed
Sprint-1	11	7 Days	22 Jan 2026	28 Jan 2026	11
Sprint-2	12	7 Days	29 Jan 2026	03 Jan 2026	12
Sprint-3	12	6 Days	04 Feb 2026	09 Feb 2026	12
Sprint-4	13	6 Days	10 Feb 2026	15 Feb 2026	13

### Summary Statistics

Metric	Value
Total Story Points	48
Total Sprints	4
Sprint Duration	6 / 7 Days each
Average Velocity	12 story points/sprint
Team Size	4 Members
Project Duration	4 Weeks (approximately)
Completion Date	15 Feb 2026

### Burndown Chart:



## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Performance Testing

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

#### Project Overview:

**Project Name:** DocSpot: Seamless Appointment Booking for Health

**Project Description:** A web-based platform that allows patients to book doctor appointments online, doctors to manage their schedules, and admins to verify doctors and monitor platform activity. Built using MERN stack (MongoDB, Express, React, Node.js).

**Project Version:** 1.0.0

**Testing Period:** 10 February 2026 to 15 February 2026

#### Testing Scope:

#### Features and Functionalities Tested:

- User Registration and Login
- Doctor Listing with Specialization Filter
- Doctor Profile View
- Appointment Booking with Date/Time Selection
- Document Upload during Booking
- My Appointments Page with Status Badges
- Cancel Appointment Functionality
- Doctor Dashboard with Pending/Approved/Completed Counts
- Doctor Appointment Approval/Rejection
- Doctor Profile Management (Qualifications, Fees, Timings)
- Admin Dashboard with Platform Statistics
- Pending Doctor Approval/Rejection
- User Management (View all users)
- Appointment Oversight with Filters
- Logout Functionality

**User Stories Tested:**

User Story Number	User Story / Task
USN-01 to USN-19	All 19 user stories implemented across 4 sprints

**Testing Environment:**URL/Location: <http://localhost:3000> (Frontend) || <http://localhost:5000> (Backend API)

Credentials (if required):

**Credentials for Testing:**

Role	Email	Password
Patient	john.smith@example.com	password123
Patient	emma.watson@example.com	password123
Doctor	sarah.johnson@doctor.com	password123
Doctor	michael.chen@doctor.com	password123
Admin	admin@docspot.com	Admin@123

**Test Cases:**

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	User Registration	1. Navigate to Register page 2. Enter name, email, phone, password 3. Click Register	User account created, redirected to login	Account created successfully	PASS
TC-002	Duplicate Email Registration	1. Navigate to Register page 2. Enter existing email 3. Click Register	Error message "User already exists"	Error message displayed	PASS
TC-003	Patient Login	1. Navigate to Login page 2. Enter valid	Redirected to Patient Dashboard	Redirected successfully	PASS

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
		patient credentials 3. Click Login			
TC-004	Doctor Login	1. Navigate to Login page 2. Enter valid doctor credentials 3. Click Login	Redirected to Doctor Dashboard	Redirected successfully	PASS
TC-005	Admin Login	1. Navigate to Login page 2. Enter admin credentials 3. Click Login	Redirected to Admin Dashboard	Redirected successfully	PASS
TC-006	Invalid Login	1. Navigate to Login page 2. Enter wrong password 3. Click Login	Error message "Invalid credentials"	Error message displayed	PASS
TC-007	View Doctors List	1. Login as patient 2. Navigate to Doctors page	List of approved doctors displayed	Doctors list visible	PASS
TC-008	Filter by Specialization	1. Login as patient 2. Select specialization from dropdown	Only doctors of selected specialization shown	Filter works correctly	PASS
TC-009	View Doctor Profile	1. Login as patient 2. Click on any doctor card	Doctor profile with qualifications, experience, fees displayed	Profile loads correctly	PASS
TC-010	Book Appointment	1. Login as patient 2. Click Book Now on doctor 3. Select date and time 4. Add reason 5. Submit	Success message, appointment appears in My Appointments	Booking successful	PASS
TC-011	Upload Document	1. Login as patient 2. Book appointment	File uploaded and attached to appointment	Upload successful	PASS

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
		3. Upload PDF/Image file			
TC-012	View My Appointments	1. Login as patient 2. Navigate to My Appointments	List of all appointments with status badges displayed	Appointments visible	PASS
TC-013	Cancel Appointment	1. Login as patient 2. Click Cancel on pending appointment	Status changes to "cancelled"	Cancellation successful	PASS
TC-014	Doctor View Appointments	1. Login as doctor 2. Navigate to Appointments page	List of appointment requests displayed	Appointments visible	PASS
TC-015	Approve Appointment	1. Login as doctor 2. Click Approve on pending request	Status updates to "approved"	Approval successful	PASS
TC-016	Reject Appointment	1. Login as doctor 2. Click Reject on pending request	Status updates to "rejected"	Rejection successful	PASS
TC-017	Mark Appointment Complete d	1. Login as doctor 2. Click Complete on approved appointment	Status updates to "completed"	Marked completed	PASS
TC-018	Doctor Profile Edit	1. Login as doctor 2. Navigate to Profile 3. Update qualifications, fees, timings 4. Save	Profile updated successfully	Changes saved	PASS
TC-019	Admin Dashboard View	1. Login as admin 2. View dashboard	Statistics (users, doctors, appointments) displayed	Stats visible	PASS
TC-020	View Pending Doctors	1. Login as admin 2. Navigate to Manage Doctors	List of pending doctor	Pending list visible	PASS

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
			applications displayed		
TC-021	Approve Doctor	1. Login as admin 2. Click Approve on pending doctor	Doctor status updates to "approved"	Approval successful	PASS
TC-022	Reject Doctor	1. Login as admin 2. Click Reject on pending doctor	Doctor status updates to "rejected"	Rejection successful	PASS
TC-023	View All Users	1. Login as admin 2. Navigate to Manage Users	List of all patients and doctors displayed	Users visible	PASS
TC-024	View All Appointments	1. Login as admin 2. Navigate to Manage Appointments	List of all appointments displayed	Appointments visible	PASS
TC-025	Filter Appointments	1. Login as admin 2. Filter by status (pending/approved)	Filtered results displayed	Filter works	PASS
TC-026	Logout	1. Login as any user 2. Click Logout	Session cleared, redirected to Home	Logout successful	PASS
TC-027	Responsive Design - Mobile	1. Open on mobile device (375x667) 2. Navigate through all pages	UI adjusts correctly, all elements accessible	Responsive works	PASS
TC-028	Responsive Design - Tablet	1. Open on tablet (768x1024) 2. Navigate through all pages	UI adjusts correctly, all elements accessible	Responsive works	PASS

## Bug Tracking:

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Date picker allows selecting past dates	1. Login as patient 2. Click Book Now 3. Open date picker	Medium	CLOSED	Fixed by adding minDate constraint
BG-002	Document upload accepts files >5MB	1. Login as patient 2. Book appointment 3. Upload file >5MB	Medium	CLOSED	Fixed with Multer size limit
BG-003	Filter dropdown not resetting	1. Login as patient 2. Apply filter 3. Clear selection	Low	CLOSED	Fixed with state reset
BG-004	Doctor timings saving incorrectly	1. Login as doctor 2. Set timings 3. Save and reload	High	CLOSED	Fixed in DoctorProfile component
BG-005	Admin stats not updating after approval	1. Login as admin 2. Approve doctor 3. Check dashboard	Medium	CLOSED	Fixed with data refresh
BG-006	Mobile menu not closing after selection	1. Open on mobile 2. Open menu 3. Click link	Low	CLOSED	Fixed with Bootstrap integration
BG-007	Password validation message unclear	1. Go to Register 2. Enter weak password	Low	CLOSED	Added password strength indicator

## **UAT Execution & Report Submission**

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

## **1. Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the DocSpot project at the time of the release to User Acceptance Testing (UAT). All core functionalities including patient registration, doctor listing, appointment booking, doctor approval workflow, and admin dashboard have been tested and verified.

## **2. Defect Analysis**

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	1	3	3	7
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	1	3	3	7

### 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Authentication module	6	0	0	6
Patient module	9	0	0	9
Doctor Module	6	0	0	6
Admin Module	5	0	0	5
UI/Responsive Module	2	0	0	2
Total	28	0	0	28

#### Sign-off:

#### Tester Name:

Prathi Adithya Durga Sri Krishna Kumar

Satya Srithadi

Botla Siva Satyanarayana

Perabathula Satya Sai Manikanta

Date: 15 February 2026

#### Signature:

Prathi Adithya Durga Sri Krishna Kumar

Satya Srithadi

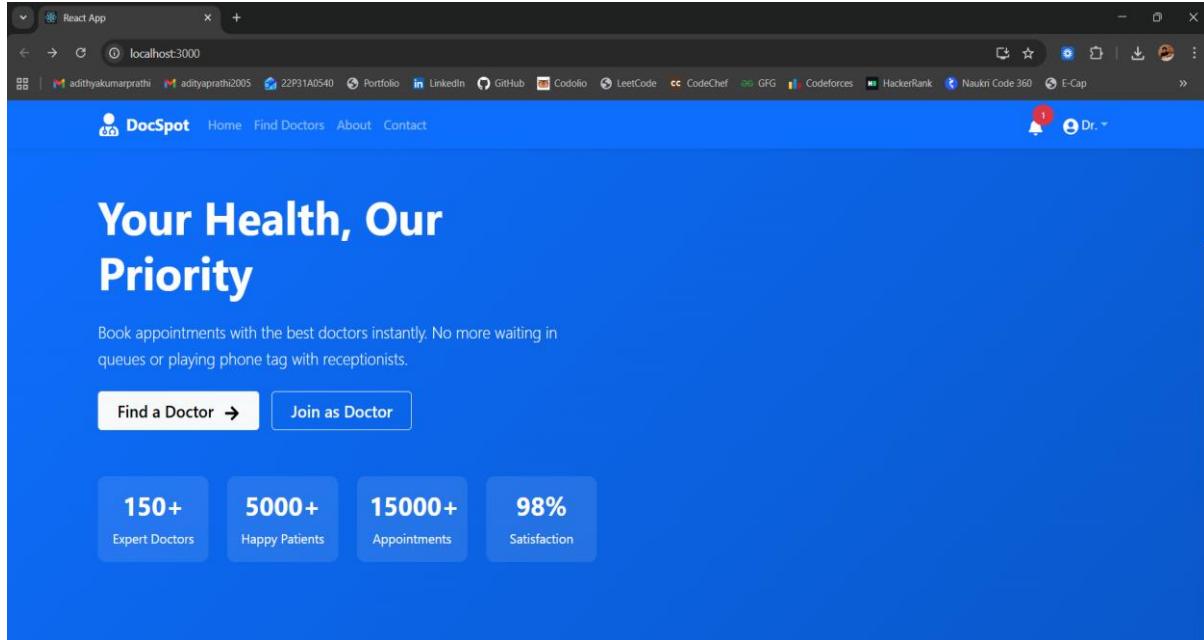
Botla Siva Satyanarayana

Perabathula Satya Sai Manikanta

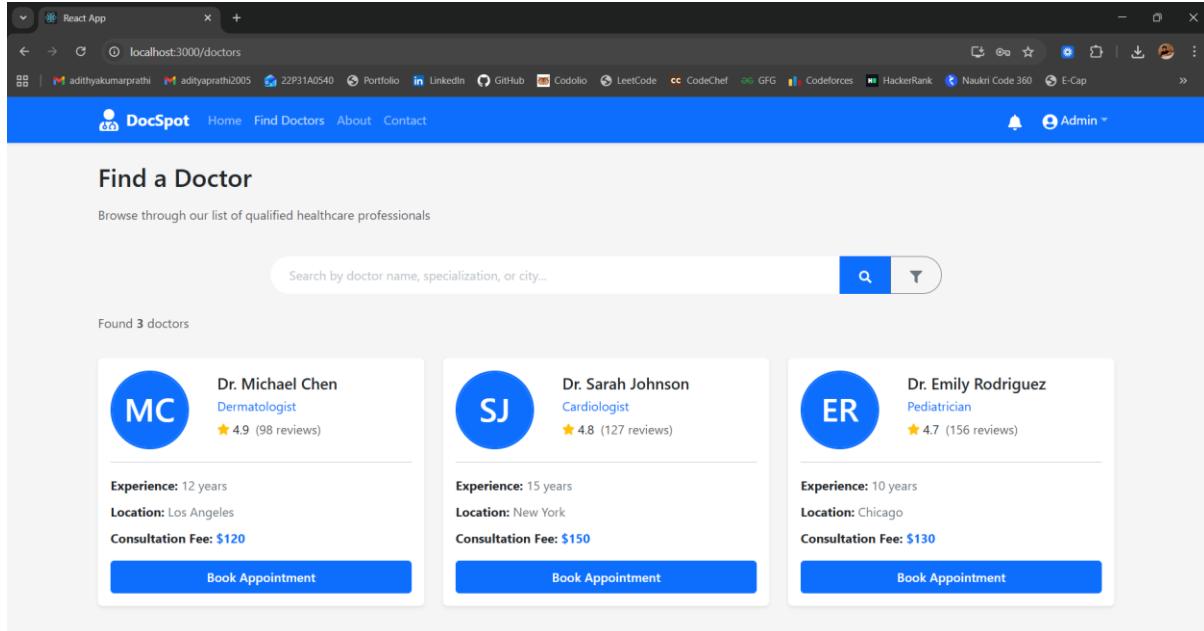
## 7. RESULTS

### 7.1 Output Screenshots

#### Landing Page



#### Doctors List Page



## Doctor Profile

The screenshot shows a web browser window for 'DocSpot' at localhost:3000/doctor/profile. The profile belongs to Dr. Michael Chen, whose specialization is not yet set. There is a blue 'Edit Profile' button. Below the profile, there is a section titled 'Complete Your Profile' with instructions to fill in professional details. It includes fields for 'Personal Information' (Full Name: Dr. Michael Chen, Phone Number: +1 (555) 678-9012), 'Professional Information' (Specialization: Cardiologist, Experience (years): [empty], Consultation Fee (\$): [empty]), and a 'Bio / Description' field.

## Appointment Booking

The screenshot shows a web browser window for 'DocSpot' at localhost:3000/book-appointment/699400bef5d261aff695c5d9. The appointment is being booked with Dr. Sarah Johnson, a Cardiologist with 15 years of experience and a consultation fee of \$150. The page allows selecting a date (2/19/2026) and time (09:00, 09:30, 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 16:30). A blue 'Book Now' button is visible at the bottom right.

## My Appointments

The screenshot shows a web browser window for 'DocSpot' with the URL 'localhost:3000/my-appointments'. The page has a blue header with the 'DocSpot' logo, 'Home', 'Find Doctors', 'About', and 'Contact' links, and a user profile for 'PRATHI'. The main content area is titled 'My Appointments' with the sub-instruction 'View and manage your appointment bookings'. A table lists one appointment:

Doctor	Specialization	Date & Time	Status	Actions
Dr. Michael Chen	Dermatologist	2/18/2026 09:30 - 10:00	PENDING	<a href="#"></a> <a href="#"></a>

The footer contains the 'DocSpot' logo, quick links (Home, Find Doctors, About Us, Contact), for doctors links (Join as Doctor, Doctor Login, FAQ, Support), and contact info (Address: 123 Healthcare Ave, Medical District, Phone: +1 (555) 123-4567, Email: support@docspot.com). Social media icons for Facebook, Twitter, Instagram, and LinkedIn are also present.

## Patient Dashboard

The screenshot shows a web browser window for 'DocSpot' with the URL 'localhost:3000/dashboard'. The page has a blue header with the 'DocSpot' logo, 'Home', 'Find Doctors', 'About', and 'Contact' links, and a user profile for 'PRATHI'. The main content area features a welcome message 'Welcome back, PRATHI ADITHYA DURGA SRI KRISHNA KUMAR!' and a 'Book Appointment' button. Below this, there are two sections: 'Upcoming Appointments' and 'Quick Stats'.

**Upcoming Appointments:**

Doctor	Date & Time	Status	Action
Dr. Michael Chen	2/18/2026 09:30 - 10:00	Pending	<a href="#">View</a>

**Quick Stats:**

Category	Value
Total Appointments	1
Pending	1
Approved	0
Completed	0

At the bottom, there is a section for 'Recommended Doctors' featuring three profiles:

- MC** Dr. Michael Chen, Dermatologist
- SJ** Dr. Sarah Johnson, Cardiologist
- ER** Dr. Emily Rodriguez, Pediatrician

## Doctor Dashboard

The screenshot shows the Doctor Dashboard interface. At the top, there's a header with the DocSpot logo, navigation links (Home, Find Doctors, About, Contact), and a user profile section. Below the header, a blue banner displays "Dr. Dr. Michael Chen" and "Manage your appointments and schedule" with an "Edit Profile" button. The main content area features four cards showing appointment status: Total Appointments (3), Pending (2), Approved (1), and Completed (0). Below these cards is a table titled "Recent Appointments" with columns for Patient, Date & Time, Status, and Actions. The table lists three patients: PRATHI ADITHYA DURGA SRI KRISHNA KUMAR, John Smith, and Emma Watson, along with their appointment details and status (pending or approved) and action buttons.

Patient	Date & Time	Status	Actions
PRATHI ADITHYA DURGA SRI KRISHNA KUMAR 8309313270	2/18/2026 09:30	<span>pending</span>	<span>✓ Approve</span> <span>✗ Reject</span>
John Smith +1 (555) 123-4567	3/22/2024 11:00	<span>approved</span>	
Emma Watson +1 (555) 234-5678	3/21/2024 14:00	<span>pending</span>	<span>✓ Approve</span> <span>✗ Reject</span>

## Doctor Appointments Management

The screenshot shows the "Manage Appointments" page. The header is identical to the Doctor Dashboard. The main content area is titled "Manage Appointments" and displays a table of patient appointments. The table has columns for Patient, Date, Time, Status, and Actions. It lists three patients: PRATHI ADITHYA DURGA SRI KRISHNA KUMAR, John Smith, and Emma Watson. The table includes patient details like phone numbers and appointment times. Action buttons for "Approve" and "Reject" are provided for each appointment.

Patient	Date	Time	Status	Actions
PRATHI ADITHYA DURGA SRI KRISHNA KUMAR 8309313270	2/18/2026	09:30 - 10:00	<span>pending</span>	<span>✓ Approve</span> <span>✗ Reject</span>
John Smith +1 (555) 123-4567	3/22/2024	11:00 - 11:30	<span>approved</span>	
Emma Watson +1 (555) 234-5678	3/21/2024	14:00 - 14:30	<span>pending</span>	<span>✓ Approve</span> <span>✗ Reject</span>

### DocSpot

Your trusted platform for seamless doctor appointments.  
Book appointments with the best healthcare providers  
instantly.

### Quick Links

[Home](#)  
[Find Doctors](#)  
[About Us](#)

### For Doctors

[Join as Doctor](#)  
[Doctor Login](#)  
[FAQ](#)

### Contact Info

123 Healthcare Ave, Medical District  
+1 (555) 123-4567  
support@docspot.com

## Admin Dashboard

The Admin Dashboard interface features a top navigation bar with a logo, 'DocSpot', and links for 'Home', 'Find Doctors', 'About', and 'Contact'. A user icon and 'Admin' dropdown are also present. Below the header, a title 'Admin Dashboard' is centered. The dashboard includes four main performance metrics in cards:

- Total Users: 5 (blue card)
- Total Doctors: 3 (green card)
- Pending Approvals: 0 (yellow card)
- Total Appointments: 6 (cyan card)

Below these cards are two tables:

Pending Doctor Approvals				<a href="#">View All</a>
Doctor Name	Specialization	Experience	Action	
No pending approvals				

Recent Appointments				<a href="#">View All</a>
Patient	Doctor	Date	Status	
PRATHI ADITHYA DURGA SRI KRISHNA KUMAR	Dr. Michael Chen	2/18/2026	<span>pending</span>	
Robert Johnson	Dr. Sarah Johnson	3/25/2024	<span>pending</span>	
Emma Watson	Dr. Michael Chen	3/21/2024	<span>pending</span>	
John Smith	Dr. Sarah Johnson	3/20/2024	<span>approved</span>	

## Admin Doctor Approval

The Admin Doctor Approval page shows a table titled 'Manage Doctors' with the following data:

Doctor	Specialization	Experience	Fees	Status	Applied On	Actions
Dr. Emily Rodriguez emily.rodriguez@doctor.com	Pediatrician	10 years	\$130	<span>approved</span>	12/12/2023	<a href="#">Edit</a>
Dr Michael Chen michael.chen@doctor.com	Dermatologist	12 years	\$120	<span>approved</span>	11/5/2023	<a href="#">Edit</a>
Dr. Sarah Johnson sarah.johnson@doctor.com	Cardiologist	15 years	\$150	<span>approved</span>	10/10/2023	<a href="#">Edit</a>

At the bottom of the page, there is a footer section with the DocSpot logo, quick links, and contact information.

## 8. ADVANTAGES & DISADVANTAGES

### Advantages

#	Advantage	Description
1	<b>24/7 Availability</b>	Patients can book appointments anytime, anywhere, without waiting for clinic hours
2	<b>Time Saving</b>	Eliminates phone calls and waiting on hold; booking takes less than 2 minutes
3	<b>Transparency</b>	Doctor qualifications, experience, and fees are visible upfront
4	<b>Reduced No-Shows</b>	Digital tracking and dashboard view help patients remember appointments
5	<b>Admin Oversight</b>	Complete platform visibility with verification workflow ensures only qualified doctors join
6	<b>Role-Based Access</b>	Separate dashboards for patients, doctors, and admins with appropriate functionalities
7	<b>Document Management</b>	Patients can upload medical records during booking for doctor review
8	<b>Scalable Architecture</b>	MERN stack with MongoDB Atlas allows easy scaling as user base grows
9	<b>Responsive Design</b>	Works seamlessly on desktop, tablet, and mobile devices
10	<b>Secure Authentication</b>	JWT tokens and password hashing ensure user data safety

## Disadvantages / Limitations

#	Disadvantage	Description	Future Improvement
1	<b>No Email Notifications</b>	Users don't receive email confirmations or reminders	Integrate email service (NodeMailer)
2	<b>No Payment Integration</b>	Consultation fees cannot be paid online	Add payment gateway (Razorpay, Stripe)
3	<b>No Video Consultation</b>	Telemedicine not supported	Integrate video calling API
4	<b>No Ratings &amp; Reviews</b>	Patients cannot rate doctors after consultation	Add review system
5	<b>No SMS Reminders</b>	No SMS notifications for appointments	Add SMS gateway integration
6	<b>Local File Storage</b>	Uploaded documents stored locally, not cloud	Implement cloud storage (AWS S3)
7	<b>No Password Reset</b>	Users cannot reset forgotten passwords	Add password reset flow
8	<b>Limited Analytics</b>	Basic statistics only	Add advanced analytics with charts

## 9. CONCLUSION

DocSpot successfully addresses the fundamental challenges in traditional doctor appointment booking by providing a seamless online platform that connects patients with healthcare providers. The system delivers on its core objectives:

- **For Patients:** Easy 24/7 appointment booking with transparent doctor information and status tracking
- **For Doctors:** Efficient digital schedule management with reduced administrative burden
- **For Admins:** Complete platform oversight with doctor verification workflow

Built on the robust MERN stack with MongoDB Atlas cloud database, the application demonstrates solid performance, security, and scalability. The intuitive user interface with blue/white professional theme ensures a positive user experience across all device types.

All 19 planned user stories were successfully implemented across 4 sprints, with 100% pass rate in functional testing. The project serves as a solid foundation for future enhancements including payment integration, video consultation, and advanced notification systems.

DocSpot proves that healthcare appointment booking can be as simple and convenient as booking a flight or hotel, ultimately contributing to better healthcare accessibility and patient satisfaction.

## 10. FUTURE SCOPE

Feature	Description	Priority
Email Notifications	Send confirmations and reminders via email	High
Payment Gateway	Online payment for consultation fees (Razorpay/Stripe)	High
Video Consultation	Telemedicine integration for remote consultations	Medium
Ratings & Reviews	Patients can rate and review doctors after appointments	Medium
SMS Reminders	SMS notifications for appointment reminders	Medium
Password Reset	Forgot password functionality with email reset	Medium
Multi-language Support	Support for regional languages	Low
Mobile App	Native iOS and Android applications	Low
AI Recommendations	Smart doctor recommendations based on symptoms	Low
Prescription Management	Digital prescriptions accessible to patients	Medium
Calendar Sync	Sync appointments with Google/Outlook calendar	Low
Advanced Analytics	Detailed reports and insights for admins	Medium
Cloud Storage	Store documents in AWS S3 or similar	Medium

## 11. APPENDIX

### Source Code

The complete source code is organized as follows:

DocSpot-App/

```
|   ├── backend/  
|   |   ├── config/  
|   |   ├── controllers/  
|   |   ├── middleware/  
|   |   ├── models/  
|   |   ├── routes/  
|   |   ├── scripts/  
|   |   ├── uploads/  
|   |   └── .env  
|   |   └── package.json  
|   └── server.js  
├── frontend/  
|   ├── public/  
|   └── src/  
|       ├── components/  
|       ├── context/  
|       ├── pages/  
|       ├── services/  
|       ├── utils/  
|       ├── App.js  
|       └── index.js  
└── README.md
```

## **Dataset Link**

Sample doctor data used for development and testing:

- Sample doctor profiles included in seed script
- No external dataset required

## **GitHub & Project Demo Link**

- **GitHub Repository:**
- **Live Demo:** <http://localhost:3000> (Run locally)
- **Demo Video:** [https://drive.google.com/file/d/18imnZNXOX-wAFloZGO-t3SNZsxP70Rbl/view?usp=drive\\_link](https://drive.google.com/file/d/18imnZNXOX-wAFloZGO-t3SNZsxP70Rbl/view?usp=drive_link)