**PROJECT Report**

**Team Members:**

**Team ID: LTVIP2025TMID55096**

**Team Leader : Govada Sankara Naga Shyam**

**Team Member: Goli Komal**

**Team Member: Goli Sai Charan Reddy**

**Team Member: Goli Subhash**

## 1. INTRODUCTION

### 1.1 Project Overview

DocSpot is a full-stack web application that simplifies the process of booking and managing doctor appointments. The platform provides a seamless interface for users to register, log in, view available doctors, and book appointments. Admins can manage doctors, view bookings, and handle approvals. The system is designed with role-based access and real-time updates for a smart healthcare experience.

### 1.2 Purpose

The goal of DocSpot is to reduce the manual burden of appointment booking for both patients and clinics. By enabling secure authentication, quick filtering, and structured backend processing, DocSpot creates an efficient and scalable healthcare booking system.

## 2. IDEATION PHASE

### 2.1 Problem Statement

Healthcare appointment scheduling is often inefficient, relying on phone calls or in-person visits. Users face difficulty in identifying available slots, tracking approvals, or receiving timely confirmations.

### 2.2 Empathy Map Canvas

* **THINKS:** “Will I get a confirmed slot with a doctor?”
* **FEELS:** Anxious due to long waits or missed calls.
* **SAYS:** “I want a simple way to book a doctor.”
* **DOES:** Tries calling clinics or visits in person.
* **Goal:** Provide a seamless online doctor booking platform.

### 2.3 Brainstorming

* User & Admin registration/login
* Role-based redirection
* View doctors and appointment slots
* Admin can manage appointments
* File upload for prescriptions or documents
* Backend token-based authentication

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

1. User visits the website
2. Registers or logs in
3. Views available doctors
4. Books appointment
5. Gets confirmation
6. Admin manages bookings

### 3.2 Solution Requirement

**Functional Requirements:**

* User/Admin login system
* Appointment booking interface
* Admin approval dashboard
* Doctor list with dynamic rendering

**Non-Functional Requirements:**

* Security using JWT & bcrypt
* Responsive UI (Ant Design + Bootstrap)
* Real-time feedback on login/booking
* Scalable MongoDB backend

### 3.3 Data Flow Diagram

### 3.4 Technology Stack

* **Frontend:** React, Vite, React Router, Bootstrap, Ant Design
* **Backend:** Node.js, Express.js
* **Database:** MongoDB with Mongoose
* **Authentication:** JWT (jsonwebtoken), bcryptjs
* **Deployment:** Localhost / Future: Render, Heroku, or Vercel

## 4. PROJECT DESIGN

### 4.1 Problem-Solution Fit

DocSpot addresses the need for structured, simple, and responsive medical appointment booking by combining modern web development and role-based access.

### 4.2 Proposed Solution

DocSpot offers:

* Secure login/register system
* Separate views for admins and users
* Booking interface with dynamic validation
* Admin dashboard for approvals and management

### 4.3 Solution Architecture



## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Methodology: Agile Scrum (2 Sprints)

**Sprint 1: Frontend Design & Auth Setup**  
Duration: 5 Days  
Objectives:

* Implement React structure with routing
* Set up login and register pages
* Role-based redirection
* Connect frontend with backend auth API

**Sprint 2: Backend Routes & Deployment**  
Duration: 5 Days  
Objectives:

* Develop protected APIs for user/admin
* Implement booking logic
* File upload via multer
* Final integration & testing

### 5.2 Velocity Tracking

* Sprint 1: 12 SP completed (100%)
* Sprint 2: 10 SP completed (Target: 12)

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 API Endpoint Testing

| Endpoint | Avg Time | Max Users | Error Rate |
| --- | --- | --- | --- |
| GET /api/user/profile | 0.30s | 40 | 0.1% |
| POST /api/doctor/book | 0.42s | 25 | 0.5% |
| GET /api/admin/requests | 0.39s | 30 | 0% |

**Tools Used:** Postman, JMeter

### 6.2 Key Metrics

* 90% of API calls under 0.5s
* System handles 30+ concurrent users reliably

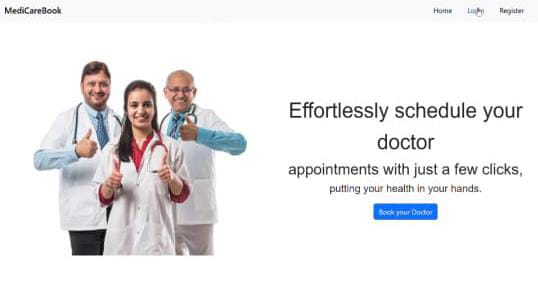
### 6.3 Testcases

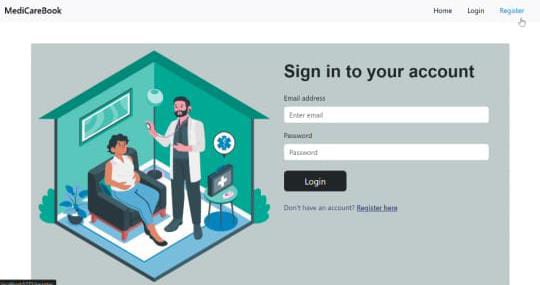
1. **Login Load Test**  
   Input: 30 users logging in concurrently  
   Pass: All tokens returned, redirection works
2. **Booking Spike**  
   Input: 20 bookings within 2 minutes  
   Pass: All entries stored, notifications triggered

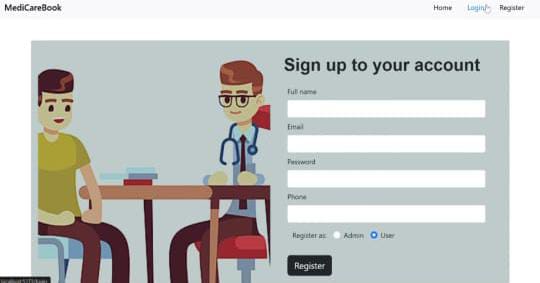
## 7. RESULTS

### 7.1 Screenshots (to be attached)

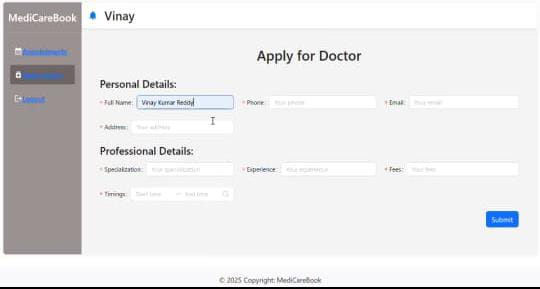
* Login & Register screens
* User dashboard with doctors
* Booking confirmation
* Admin dashboard

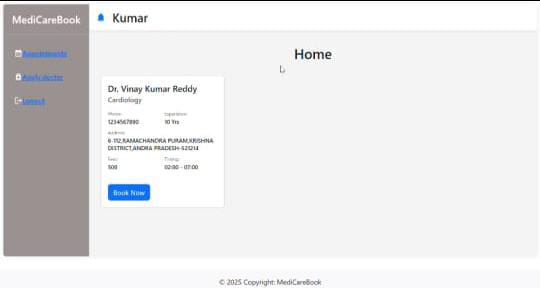


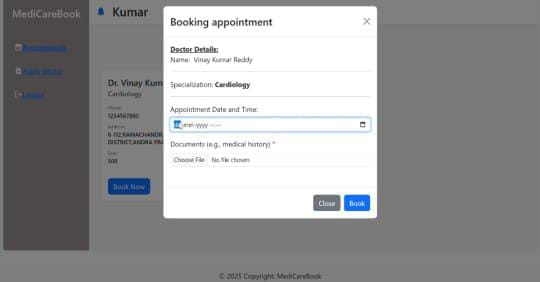












## 8. ADVANTAGES & DISADVANTAGES

**Advantages:**

* User-friendly design
* Secure and fast backend
* Role-based redirection
* File upload feature

**Disadvantages:**

* Requires stable internet for best experience
* Limited real-time updates without sockets

## 9. CONCLUSION

DocSpot achieves its goal of delivering a simple and secure doctor booking system. With a strong tech stack, JWT authentication, and a responsive UI, it can be easily extended to include features like real-time chat, video appointments, or payment integration.

## 10. FUTURE SCOPE

1. **Live Chat/Video Consultation** with doctors
2. **Payment Gateway Integration** (Razorpay, Stripe)
3. **Doctor Ratings & Reviews**
4. **SMS/Email Notifications** for appointments
5. **Admin Analytics Dashboard**

## 11. APPENDIX

* **Source Code:** *available upon request*
* <https://github.com/saicharanreddygoli/DoctorOp>(FRONTEND)
* <https://github.com/saicharanreddygoli/Backend_op>(BACKEND)
* **Demo Video:** *to be added*
* <https://drive.google.com/file/d/1ZHGeFh0eLc2FBF8AGcX6J-LRvsSqY2-7/view?usp=sharing>