

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

## **PROJECT REPORT**

### **1. INTRODUCTION**

#### **1.1 Project Overview**

DocSpot is a full-stack web-based doctor appointment booking system developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js).

The application provides a centralized platform for patients to easily book medical appointments online. It enables doctors to manage their schedules, confirm bookings, and update appointment statuses efficiently. The admin module ensures proper governance by approving doctor registrations and monitoring system activities.

The system supports secure authentication, real-time availability tracking, and role-based access control.

#### **1.2 Purpose**

The purpose of the DocSpot project is to modernize the traditional doctor appointment system through digital transformation. It aims to eliminate long waiting times by enabling patients to book appointments online from anywhere. The system provides real-time doctor availability, allowing users to select convenient time slots instantly. It facilitates efficient online appointment management for patients, doctors, and administrators. The platform ensures secure access with role-based authentication and centralized monitoring.

### **2. IDEATION PHASE**

#### **2.1 Problem Statement**

Traditional doctor appointment booking systems are mostly manual, requiring phone calls or in-person visits. These methods are time-consuming and often lead to long waiting periods for patients. Manual record management can result in scheduling conflicts and human errors. Patients may struggle to find available doctors or suitable time slots quickly. Doctors also face difficulties in managing appointments efficiently without a proper system.

#### **2.2 Empathy Map Canvas**

**Who?** Patients, Doctors, Admin

**Think & Feel:** Patients feel frustrated with waiting times. Doctors feel overloaded managing manual appointments.

**Hear:** "Please wait on hold." "Doctor is unavailable."

**See:** Crowded hospitals Manual booking registers

**Pain:** Time-consuming booking , Miscommunication , Appointment conflicts

**Gain:** Quick booking , Real-time updates , Better appointment tracking

## 2.3 Brainstorming

- Online doctor listing
- Real-time appointment slots
- Role-based authentication (Admin, Doctor, User)
- File upload for medical records
- Notification system
- Dashboard for each role
- Secure login using JWT

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

| Stage         | Action                   | Feeling   | Opportunity              |
|---------------|--------------------------|-----------|--------------------------|
| Awareness     | User searches for doctor | Worried   | Provide online platform  |
| Consideration | Views doctor list        | Confused  | Filter by specialization |
| Decision      | Books appointment        | Relieved  | Instant confirmation     |
| Action        | Visits doctor            | Confident | Medical consultation     |

### 3.2 Solution Requirements

#### Functional Requirements:

- User Registration & Login
- Doctor Registration & Admin Approval
- Appointment Booking
- Appointment Cancellation
- Appointment Status Tracking
- Notification System
- Admin Dashboard

#### Non-Functional Requirements:

- Secure authentication (JWT)
- Scalable database (MongoDB)
- Fast API response
- User-friendly UI

### **3.3 Data Flow Diagram**

#### **Level 1 DFD:**

User → React Frontend → Express Server → MongoDB → Response → UI

#### **Level 2 DFD:**

User Input → Validation → API Call → Controller → Database → Status Update → Notification  
→ UI Display

### **3.4 Technology Stack**

**Frontend :** React.js , Bootstrap , Ant Design , Axios

**Backend :** Node.js , Express.js , JWT Authentication , Middleware

**Database :** MongoDB , Mongoose

**Tools :** Git & GitHub , Postman , VS Code

## **4. PROJECT DESIGN**

### **4.1 Problem Solution Fit**

Manual booking systems lack efficiency and transparency.

Doc Spot provides:

- Real-time scheduling
- Centralized management
- Role-based dashboards
- Automated appointment flow

### **4.2 Proposed Solution**

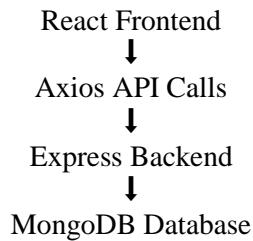
A MERN-based web application where:

- Users register and book appointments
- Doctors manage appointment requests
- Admin approves doctors and oversees platform

The system ensures secure authentication, real-time data exchange, and scalable architecture.

## 4.3 Solution Architecture

Client–Server Architecture:



## 5. PROJECT PLANNING & SCHEDULING

| Task                 | Timeline | Responsibility |
|----------------------|----------|----------------|
| Requirement Analysis | Day 1    | All members    |
| Database Design      | Day 2    | Member 1&2     |
| Backend Development  | Day 3    | Member 3&4     |
| Frontend Development | Day 4    | Member 4&5     |
| Integration          | Day 5    | All Team       |
| Testing & Deployment | Day 6    | All Team       |

## 6. FUNCTIONAL & PERFORMANCE TESTING

### 6.1 Functional Testing

- User Registration Test
- Login Authentication Test
- Doctor Approval Test
- Appointment Booking Test
- Status Update Test
- Cancellation Test

All APIs were tested using Postman.

## 6.2 Performance Testing

- Fast API response (< 2 seconds)
- Secure password hashing using bcrypt
- Token-based authentication using JWT
- Efficient CRUD operations

# 7. RESULTS

## 7.1 Output Screenshots

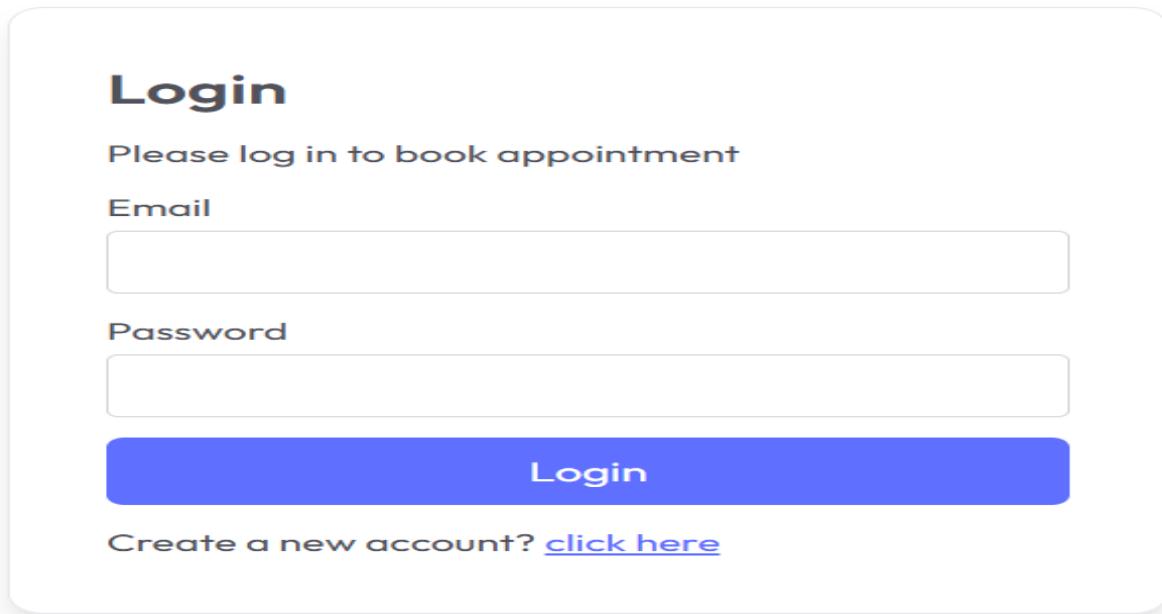
- Home Page



- User create Account page

A screenshot of the 'Create Account' page. The title 'Create Account' is at the top, followed by the instruction 'Please sign up to book appointment'. There are three input fields: 'Full Name' (with a placeholder 'John Doe'), 'Email' (with a placeholder 'john.doe@example.com'), and 'Password' (with a placeholder 'password123'). Below these is a large blue 'Create Account' button. At the bottom, there is a link 'Already have an account? [Login here](#)'.

- user login page



The image shows a user login form with a light gray background and rounded corners. At the top center is the word "Login" in a bold, dark blue font. Below it is a sub-instruction "Please log in to book appointment" in a smaller, dark blue font. There are two input fields: one for "Email" and one for "Password", both represented by white rectangular boxes with thin black borders. Below these fields is a large, solid blue rectangular button with the word "Login" in white. At the bottom left of the form is a link "Create a new account? [click here](#)" in a dark blue font.

**Login**

Please log in to book appointment

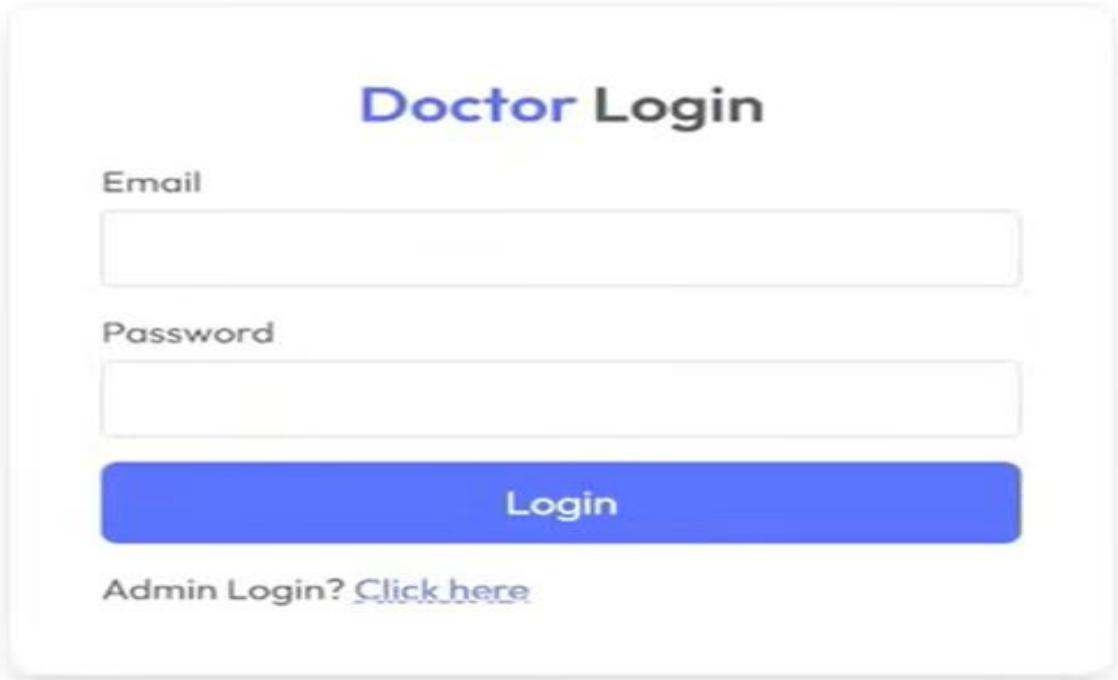
Email

Password

Login

Create a new account? [click here](#)

- Doctor login page



The image shows a doctor login form with a light gray background and rounded corners. At the top center is the text "Doctor Login" in a bold, dark blue font. Below it are two input fields: one for "Email" and one for "Password", both shown as white rectangular boxes with thin black borders. Below these fields is a large, solid blue rectangular button with the word "Login" in white. At the bottom left of the form is a link "Admin Login? [Click here](#)" in a dark blue font.

**Doctor Login**

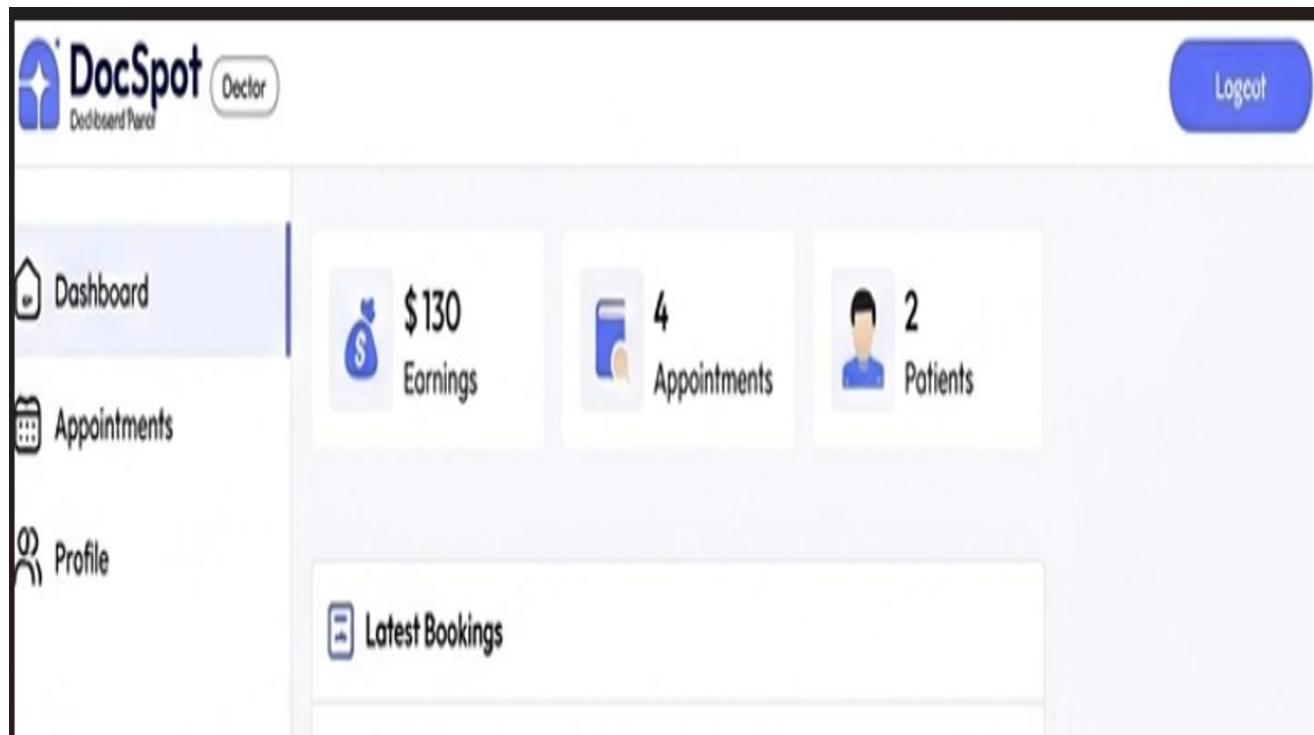
Email

Password

Login

Admin Login? [Click here](#)

- Doctor Dashboard page



- Admin login page

The screenshot shows the Admin Login page. It features a title "Admin Login" at the top center. Below it are two input fields: "Email" and "Password", each with a corresponding label and an empty input box. At the bottom is a large blue "Login" button with a white hand cursor icon. Below the button is the text "Doctor Login? [Click here](#)".

- Admin Dashboard page

The screenshot shows the DocSpot Admin Dashboard. On the left, there's a sidebar with icons for Dashboard, Appointments (selected), Add Doctor, and Doctors List. The main area is titled 'Add Doctor' and contains fields for Name, Speciality, Doctor Email, Degree, Set Password, Address, Experience, and About Doctor.

| Field        | Description       |
|--------------|-------------------|
| Your name    | Name              |
| Speciality   | General physician |
| Doctor Email | Email             |
| Degree       | Degree            |
| Set Password | Password          |
| Address      | Address 1         |
| Experience   | 1 Year            |
| About Doctor | (empty)           |

## 8. ADVANTAGES & DISADVANTAGES

### Advantages:

- Real-time booking
- Role-based access
- Secure authentication
- Easy to use
- Scalable architecture

### Disadvantages:

- Requires internet connection
- Not integrated with hospital EMR systems
- Basic notification system

## **9. CONCLUSION**

Doc Spot successfully demonstrates the implementation of a full-stack MERN application for healthcare appointment booking. It simplifies the traditional appointment system by providing real-time availability, secure authentication, and centralized management.

## **10. FUTURE SCOPE**

- Online payment integration
- Video consultation feature
- SMS/Email notifications
- Mobile application version

## **11. APPENDIX**

### **GitHub & Project Demo Link:**

GitHub Repository:

<https://github.com/YVDSAI/Doc-Spot-Seamless-Appointment-Booking-for-Health>

Project video Link:

<https://drive.google.com/file/d/1-B803CypD1f5hnSFNYKbOrXEey-Tkhw/view?usp=drivesdk>