

# Full Stack Development with MERN: Project Documentation

---

## 1. Introduction

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

**Team ID:** LTVIP2025TMID58589

**Team Members:**

1. Putti Hyndavi
2. Laya Pennem
3. Yelesam Sreenu
4. Shaik Abdul Hanif

## 2. Project Overview

**Purpose:**

DocSpot is a MERN-based health-tech platform that simplifies appointment booking, doctor discovery, and virtual consultation management for patients, doctors, and admins..

**Key Features:**

- Secure JWT-based login/signup for Patient, Doctor, Admin roles
- Doctor profile listing with specialization, timings, fees, and reviews
- Advanced search with filters (e.g., location, availability, ratings)
- Appointment requests, approvals, and rescheduling
- Admin dashboard for verification and platform governance
- In-app messaging/reminders for confirmed consultations

## 3.Architecture

### Frontend:

- Built using React.js
- Styled with Tailwind CSS and Bootstrap
- Utilizes Axios for API communication

### Backend:

- Node.js + Express.js based RESTful API
- JWT authentication
- Modular routing and middleware-based access control

### Database:

- MongoDB with Mongoose ODM
- Collections: Users, Doctors, Appointments, Admin Logs

## 4.Setup Instructions

### Prerequisites:

- Node.js ( $\geq 14.x$ )
- MongoDB

### Installation:

1. Clone the repo
2. Run “npm install” in both “/client” and “/server”
3. Set environment variables (.env ‘ file) for MongoDB URI and JWT secret
4. Start servers using npm Start

## 5.Folder Structure

### ○ Client (React Frontend):

- /client
  - /src
    - /components
    - /pages
    - /assets

```
    /services
App.js
○ Server (Node Backend):
  /server
  /controllers
  /models
  /routes
  server.js
```

## 6. Running the Application

### Frontend:

```
cd client
npm start
```

### Backend:

```
cd server
npm start
```

## 7. API Documentation

### Sample Endpoints:

- POST `/api/auth/register` - User registration (Patient/Doctor/Admin)
- POST `/api/auth/login` - User login
- GET `/api/doctors` - Browse all doctor profiles
- POST `/api/appointments` - Book an appointment
- PUT `/api/appointments/:id` - Reschedule or cancel

## 8. Authentication

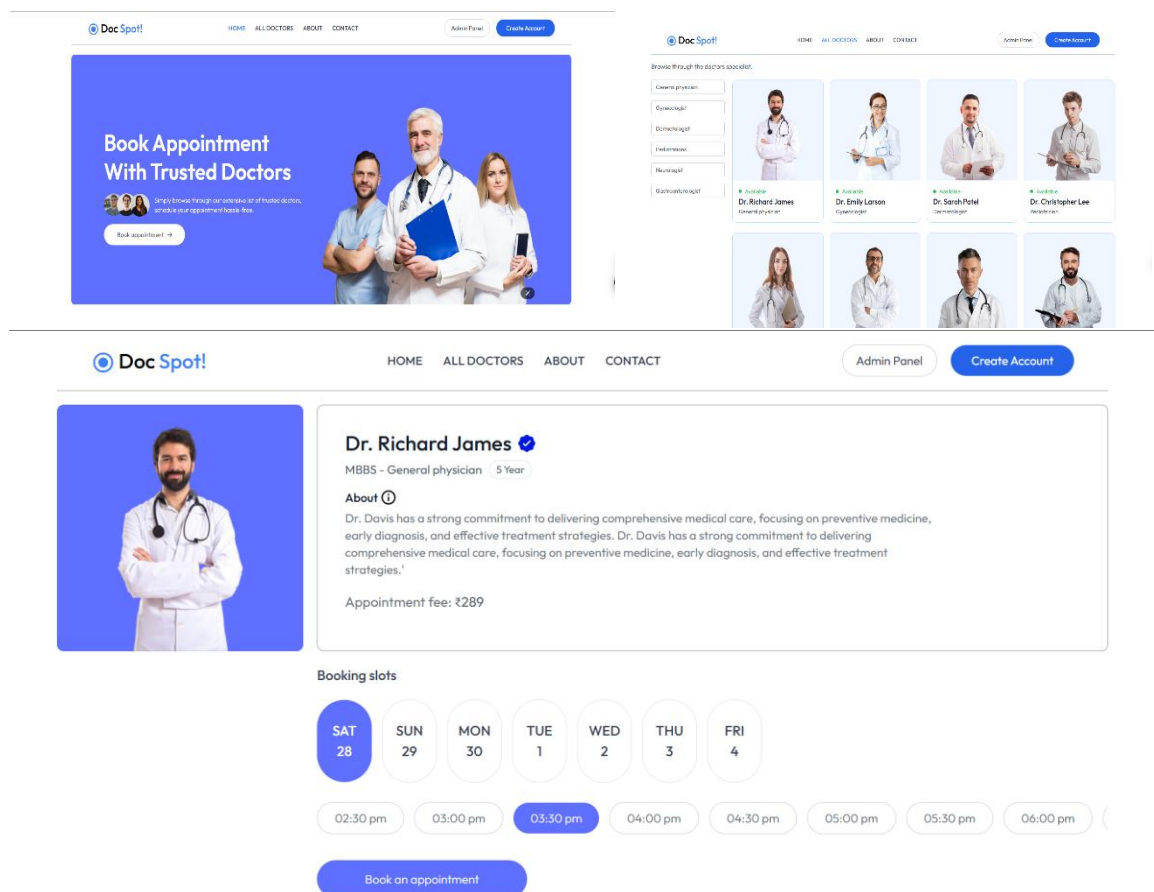
- JWT-based token system in localStorage
- Role-based access (Patient / Doctor / Admin)
- Protected routes via Express middleware

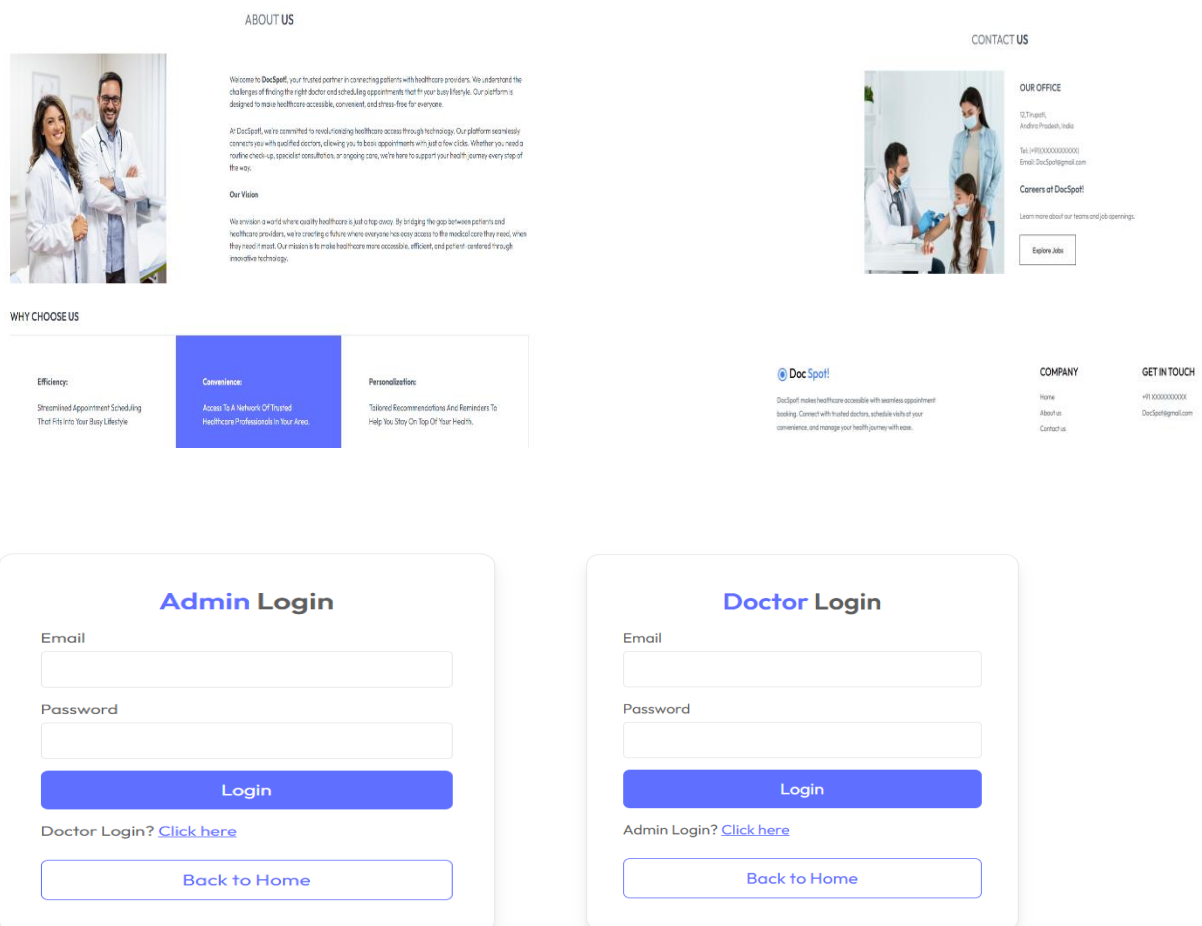
## 9.User Interface

Responsive, role-based UI with tailored dashboards for:

- Patients → Bookings & Health History
- Doctors → Appointment Management
- Admins → Verification & Analytics

## 10.Screenshots:





## 11. Testing

### Manual testing with demo credentials:

- Patient: : <https://docspot-com.vercel.app/>
- Doctor & Admin: <https://docspots-admins.vercel.app/>

## 12. Known Issues

- No integrated payment gateway
- Limited responsiveness on older mobile devices
- No real time chat yet

## **13.Future Enhancements**

- Calendar-based booking conflict resolution
- Stripe/PayPal payment integration
- Real-time chat using WebSocket or Firebase
- Convert to PWA for offline support