

# **Full Stack Development with MERN Report**

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

### **1. Introduction**

**Title:** Docspot: Seamless Appointment Booking for Health

**Team members:** B. Lohith Pavan Krishna – Full Stack Developer – Frontend & Backend Development including database schema and connections.

### **2. Project Overview**

Purpose:

DocSpot is a web-based healthcare appointment booking system that allows patients to easily schedule appointments with doctors online. The main goal of the system is to replace traditional manual booking methods, reduce waiting time at hospitals or clinics, and make healthcare services more accessible and convenient for users.

Features:

- **User Registration and Login:**  
Allows users to create accounts and securely log in to access the system.
- **Doctor Listing and Specialization Filtering:**  
Displays available doctors and enables users to filter them based on specialization for easier selection.
- **Appointment Booking System:**  
Enables patients to book appointments with doctors quickly through an online interface.
- **Secure Authentication:**  
Uses secure login mechanisms (such as JWT and password encryption) to protect user data.
- **Responsive User Interface:**  
Provides a clean and mobile-friendly design for smooth usage across different devices.
- **Admin/Doctor Management:**  
Allows administrators to manage doctors, users, and appointments efficiently (if implemented).

### **3. Architecture**

#### **Frontend:**

The frontend of DocSpot is developed using React.js, which follows a component-based architecture. This approach allows the application to be divided into reusable UI components, making the code more organized and maintainable.

- Axios is used to communicate with the backend APIs.
- React Router handles navigation between different pages such as login, registration, and appointment booking.
- React Hooks (like useState and useEffect) are used for managing application state and handling lifecycle events.

#### **Backend:**

The backend is built using Node.js and Express.js, which together create a powerful server-side environment.

- The application follows a RESTful API structure, where different routes handle users, doctors, and appointments.
- Middleware is used for authentication and protecting private routes.
- Environment-based configuration (using .env files) ensures secure handling of sensitive information like database credentials and JWT secrets.

#### **Database:**

The project uses MongoDB as a NoSQL database to store application data.

- Mongoose ODM (Object Data Modeling) is used to define schemas and interact with the database efficiently.
- The main collections include:
  - Users – Stores patient and admin information.
  - Doctors – Stores doctor profiles and specialization details.
  - Appointments – Stores booking details and appointment status.

## 4. Setup Instructions

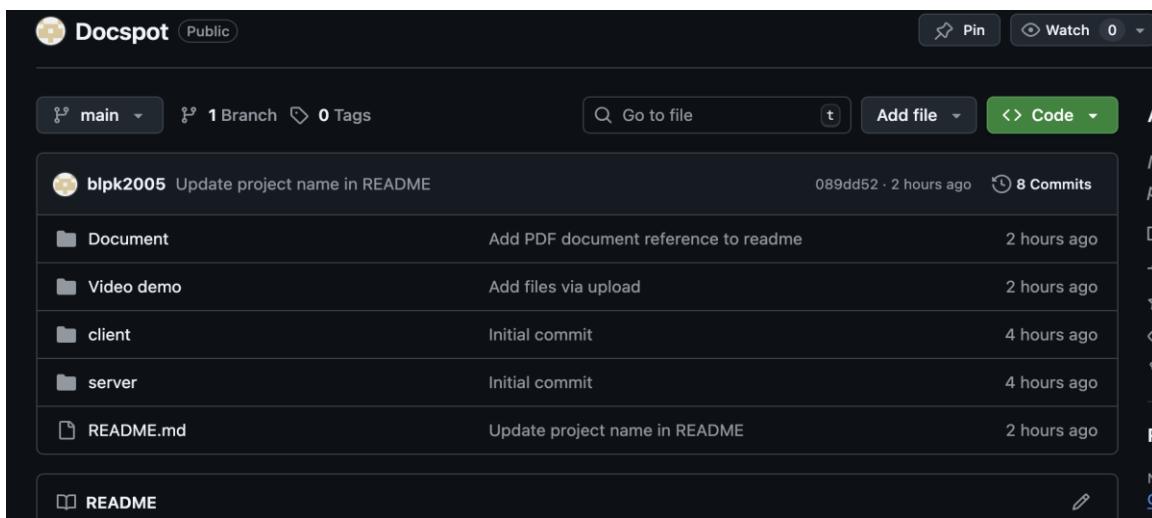
### Prerequisites:

Before running the DocSpot application, ensure the following software is installed on your system:

- **Node.js (v16 or above recommended):**  
Required to run both the frontend (React) and backend (Node.js/Express) servers.
- **MongoDB (Local Installation or MongoDB Atlas):**  
Used as the database to store user, doctor, and appointment data. You can either install MongoDB locally or use a cloud database service like MongoDB Atlas.
- **Git:**  
Required to clone the project repository from GitHub and manage version control.

## 5. Folder Structure

### Root Directory:

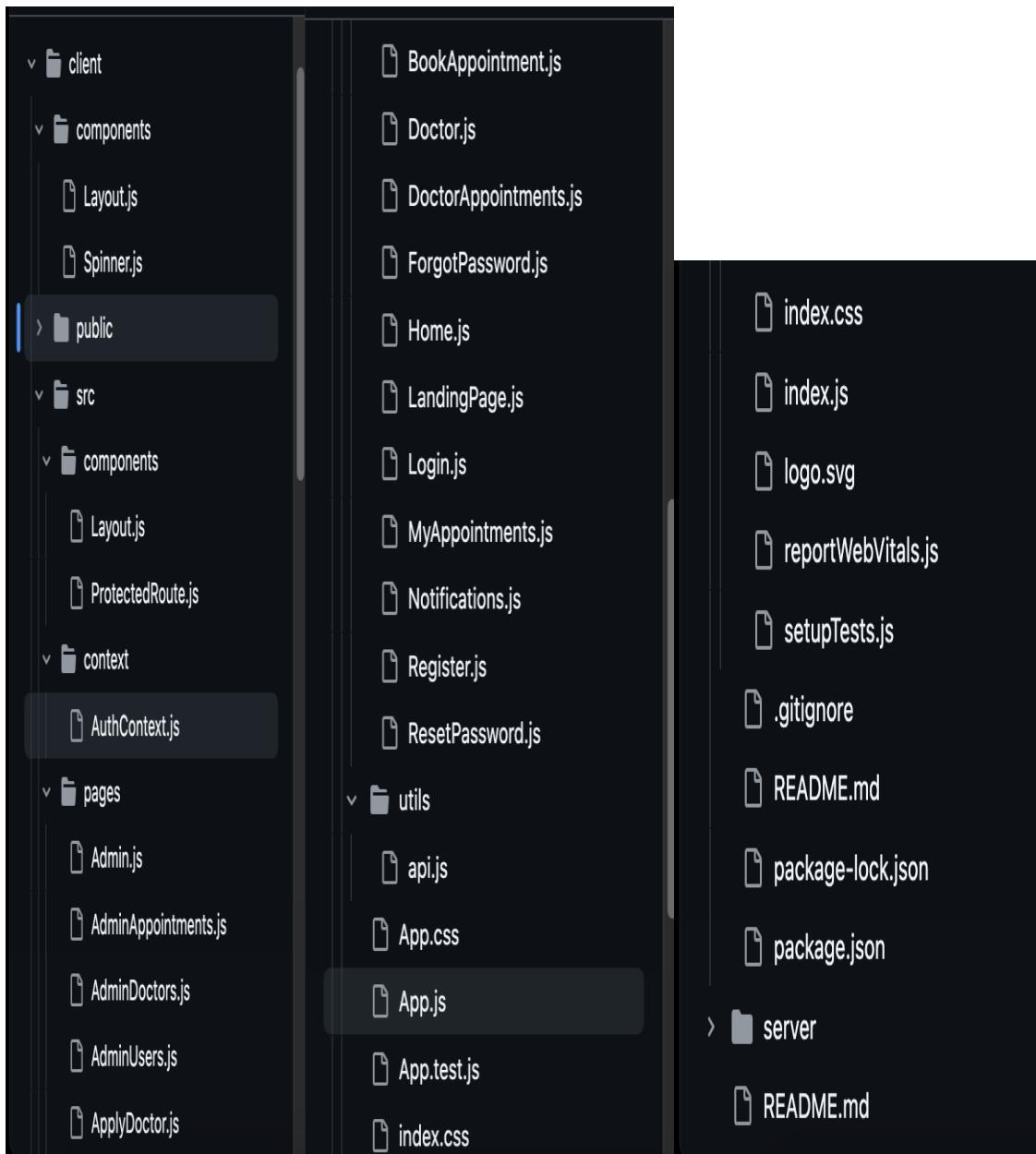


The screenshot shows a GitHub repository named "Docspot". The repository is public and has one branch ("main") and no tags. The commit history is as follows:

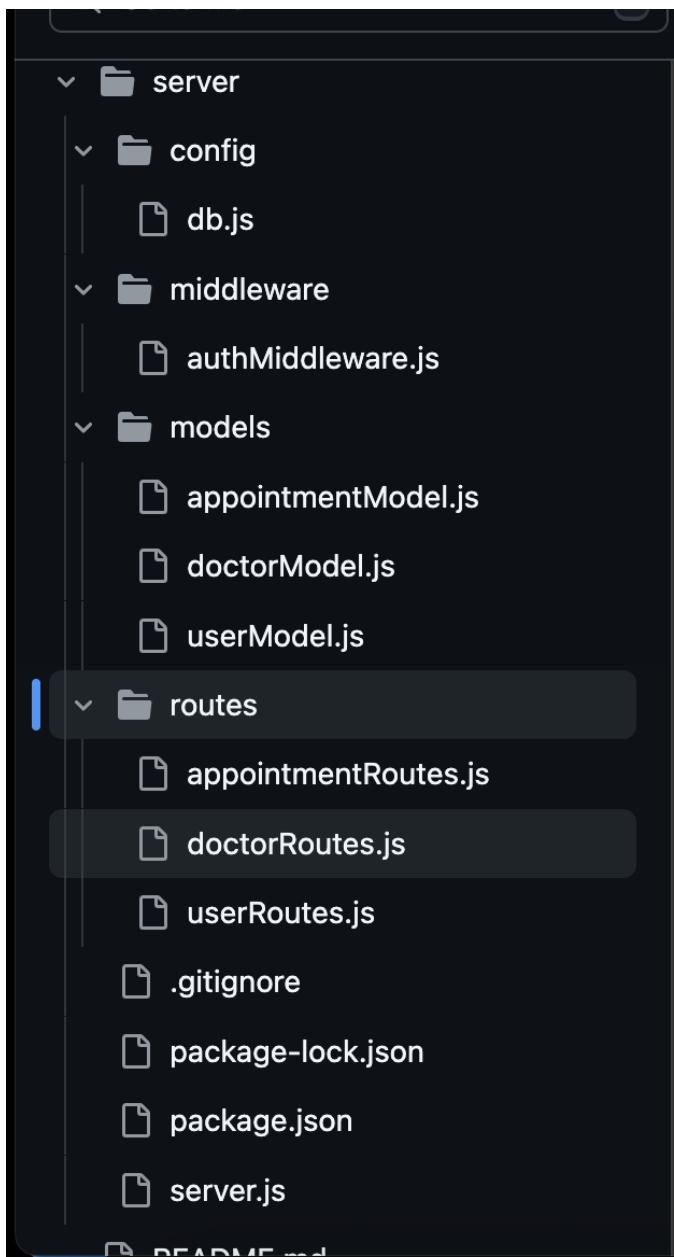
Commit	Message	Date	Commits
blpk2005	Update project name in README	089dd52 · 2 hours ago	8 Commits
Document	Add PDF document reference to readme	2 hours ago	
Video demo	Add files via upload	2 hours ago	
client	Initial commit	4 hours ago	
server	Initial commit	4 hours ago	
README.md	Update project name in README	2 hours ago	

At the bottom of the repository page, there is a "README" file listed.

## Client or Frontend:



## Server or Backend:



## Architecture Explanation

Client Side:

- Handles UI rendering
- Protected routes using ProtectedRoute.js
- Global authentication state using AuthContext.js
- API communication handled via utils/api.js

Server Side:

- db.js → MongoDB connection
- authMiddleware.js → JWT verification
- models/ → Mongoose schemas
- routes/ → API endpoint definitions
- server.js → Express app entry point

## 6. Running the Application

Start Backend:

```
cd server  
npm start
```

Server runs on: <http://localhost:5000>

Start Frontend:

```
cd client  
npm start
```

Frontend runs on: <http://localhost:3000>

## 7. API Documentation

Authentication

Method	Endpoint	Description
POST	/api/auth/register	Register new user
POST	/api/auth/login	Login user

Doctors

Method	Endpoint	Description
GET	/api/doctors	Get all doctors
GET	/api/doctors/:id	Get doctor by ID

## Appointments

Method	Endpoint	Description
POST	/api/appointments	Book appointment
GET	/api/appointments	Get user appointments

## 8. Authentication

- JSON Web Tokens (JWT) used
- Token stored in localStorage
- Middleware verifies token for protected routes
- Passwords hashed using bcrypt

## 9. User Interface

- Clean and responsive design
- Doctor cards with details
- Booking confirmation system
- Navigation bar for easy routing

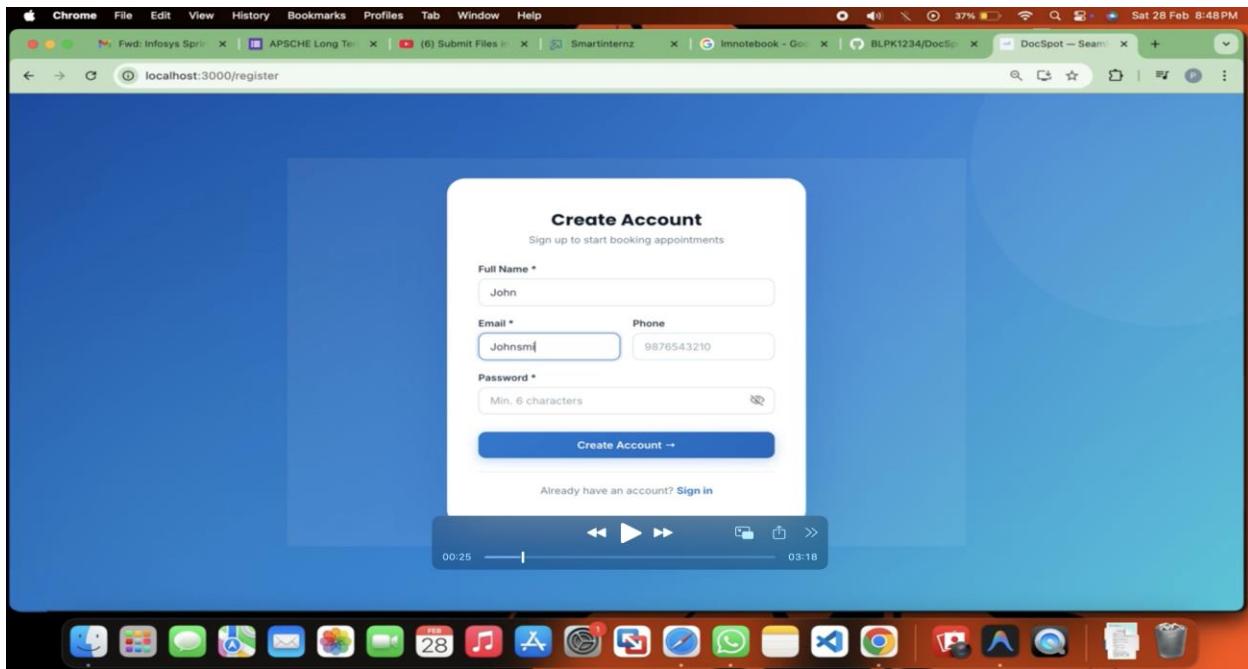
## 10. Testing

- Manual testing performed
- API tested using Postman
- Form validations implemented on frontend
- Error handling implemented in backend

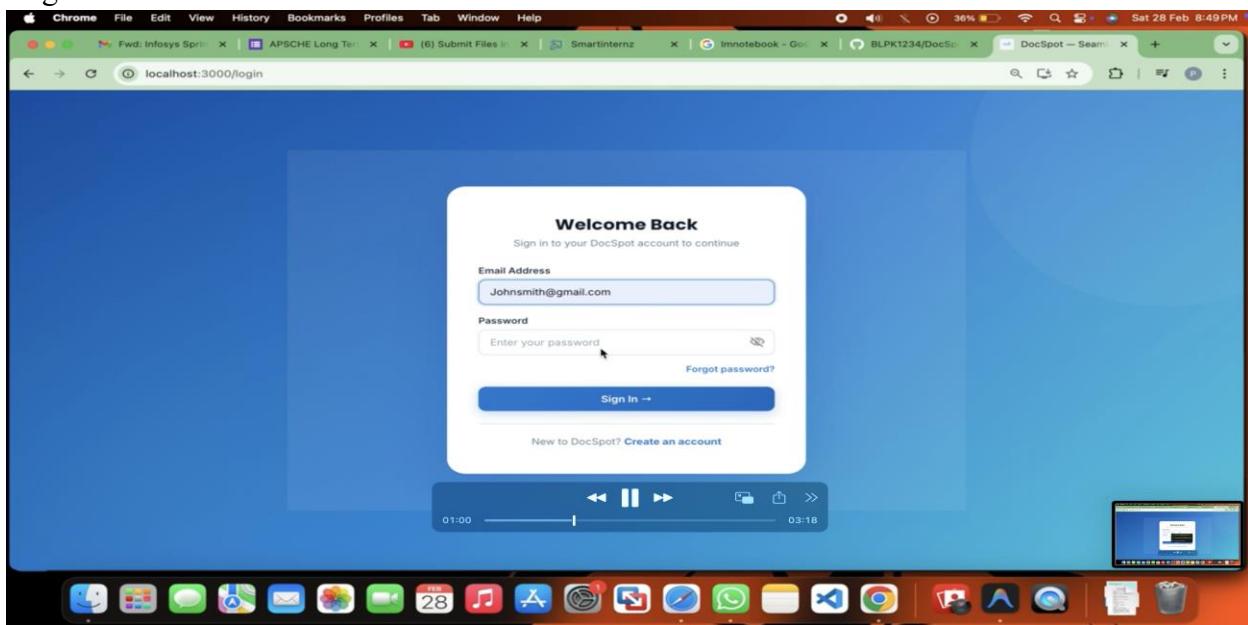
## 11. Screenshots or Demo

GitHub Repository: <https://github.com/blpk2005/Docspot>

## Landing Page:



## Login:



## User Dashboard:

The screenshot shows a web browser window for 'DocSpot - Se' on Saturday, February 28, at 8:49 PM. The URL is 'localhost:3000/dashboard'. The dashboard features a main header with tabs: 'Dashboard' (selected), 'Book Appointment', 'My Appointments', 'Apply as Doctor', and 'Notifications'. A user profile 'John PATIENT' is visible. Below the header is a large blue banner with the text 'Find Your Perfect Doctor' and a subtext: 'Browse from our network of 1+ verified doctors. Book an appointment instantly and get the care you deserve.' Two buttons are present: 'Book Appointment' and 'My Appointments'. Below the banner are four stats boxes: '1 Verified Doctors', '1 Specialties Available', '24/7 Online Booking', and '4.9★ Average Rating'. A section titled 'Available Doctors' follows, featuring a search bar and a list of doctors. One doctor listed is 'Dr. Krishna' (Cardiologist) with 5 years experience. The interface includes a navigation bar with various icons at the bottom.

## Patient or user booking a appointment:

The screenshot shows a web browser window for 'DocSpot - Se' on Saturday, February 28, at 8:49 PM. The URL is 'localhost:3000/book-appointment'. The dashboard header is visible at the top. The main content is the 'Book an Appointment' page. A green success message box says 'Appointment booked successfully! You'll be notified once confirmed.' Below it, a 'Select Doctor' dropdown is set to 'Dr. Krishna – Cardiologist (₹500)'. An 'Appointment Date' input field shows '06/03/2026'. A 'Upload Document (optional)' section has a placeholder 'Click to upload medical records, insurance, etc.' and a 'Confirm Appointment' button. To the right, a 'Doctor Details' sidebar shows 'Dr. Krishna' (Cardiologist) with 5 years of experience, located in Guntur, with phone number 911078764 and consultation fee ₹500. It also lists 'Timings: 9:00AM to 12:00pm'. At the bottom, a light blue box displays 'Your appointment date: Friday, 6 March 2026'. The interface includes a navigation bar with various icons at the bottom.

## Doctor dashboard:

The screenshot shows a Chrome browser window with the URL `localhost:3000/dashboard`. The page title is "Find Your Perfect Doctor". A top navigation bar includes "Dashboard", "Appointments", and "Notifications" (with 1 notification). A user profile for "Krishna" (y22cd009@rvrjc.ac.in, DOCTOR) is shown, along with a "Logout" button. Below the profile, there are four stats: 1 Verified Doctors, 1 Specialties Available, 24/7 Online Booking, and a 4.9★ Average Rating. A search bar and a "All Specialties" dropdown are present. A video player interface with a play button and a progress bar from 02:18 to 03:18 is overlaid on the dashboard. The Mac OS Dock at the bottom contains various application icons.

## Admin Dashboard:

The screenshot shows a Chrome browser window with the URL `localhost:3000/dashboard`. The page title is "Find Your Perfect Doctor". A top navigation bar includes "Dashboard", "Doctors", "Users", "Appointments", and "Notifications" (with 1 notification). A user profile for "Admin" (ADMIN) is shown, along with a "Logout" button. Below the profile, there are four stats: 1 Verified Doctors, 1 Specialties Available, 24/7 Online Booking, and a 4.9★ Average Rating. A search bar and a "All Specialties" dropdown are present. A video player interface with a play button and a progress bar from 02:18 to 03:18 is overlaid on the dashboard. The Mac OS Dock at the bottom contains various application icons.

## **12. Known Issues**

- No email notification system
- Limited admin dashboard features
- Appointment time slot validation can be improved