

DocSpot – Doctor Appointment Booking System

Introduction

In today's fast-paced world, scheduling timely medical appointments can be a daunting task for both patients and healthcare professionals. *DocSpot* is a modern, full-stack web application designed to bridge this gap efficiently. Developed using the powerful MERN stack (MongoDB, Express.js, React.js, Node.js), the platform empowers patients to book appointments seamlessly while allowing doctors and administrators to manage schedules and maintain operational transparency.

Problem Statement & Objective

In many healthcare setups, patients still rely on traditional systems—manual registers, in-person visits, and telephonic coordination—to book appointments. This often leads to mismanagement, overlaps, and long wait times.

Objective:

To develop an intuitive, responsive, and secure appointment booking platform where:

- **Patients** can view doctor availability and book slots easily.
- **Doctors** can manage schedules, approve or cancel appointments.
- **Admins** maintain platform control with verification, access, and monitoring tools.

Project Overview

DocSpot is a comprehensive web-based solution aimed at revolutionizing how patients interact with healthcare providers. By integrating modern web technologies and following a role-based architecture, it eliminates the dependency on traditional appointment systems. The project streamlines the booking process while maintaining security, transparency, and flexibility for all user types.

Patients can explore doctor profiles, including their specialization, experience, and availability, and schedule appointments based on real-time data. Doctors, on the other hand, get access to a personalized dashboard where they can manage upcoming appointments, view patient history, and update their availability. The admin plays a central role in the system's integrity by approving or rejecting doctor applications and monitoring user activity.

This modular and scalable system is built to serve not only clinics but also larger hospitals aiming to digitize their outpatient services. DocSpot promotes digitization, reduces no-shows with notifications, and contributes to better time management for both patients and medical practitioners.

Key modules include:

- **User Role Management:** Customized experience for each user type.

- **Smart Booking System:** Prevents double-booking using real-time slot checking.
- **Notification Engine:** Ensures every action is acknowledged with alerts.
- **Admin Approval Workflows:** Safeguards system from unauthorized usage.

System Architecture

DocSpot follows a modular, service-oriented architecture, enabling high scalability, code maintainability, and ease of deployment. The system is divided into three primary components: frontend, backend, and database.

Frontend Layer:

The client-side application is built using React.js with TypeScript, providing type safety and improved developer experience. It is enhanced with Redux for centralized state management, ensuring that data flow between components is predictable and consistent. The use of Tailwind CSS enables rapid UI development with a clean, responsive design across all device sizes.

The routing logic is handled by React Router DOM, allowing nested and role-based routes to restrict or allow access depending on the user's role (Admin, Doctor, Patient). Components such as Doctor Card, Login Model, and Notification Bar are designed to be reusable and follow atomic design principles.

Backend Layer:

The backend is implemented using Node.js and Express.js, exposing a robust RESTful API that interacts with the database and handles core logic like authentication, authorization, booking, and admin controls. Each route is protected using middleware that validates JWT tokens and checks user roles to prevent unauthorized access.

Database Layer:

MongoDB, a NoSQL database, is used to store unstructured yet relationally grouped data. Collections are modeled using Mongoose schemas, ensuring data integrity and ease of querying. Indexing and relationships between users, doctors, and appointments are handled efficiently to support fast lookups and updates.

Technology Stack

DocSpot leverages a modern technology stack that balances developer productivity with application performance and scalability. Each technology was chosen to fulfill specific project requirements:

- **React.js:** Component-based UI rendering, virtual DOM for performance.
- **TypeScript:** Static typing reduces bugs and improves readability.
- **Redux Toolkit:** Streamlines state management with fewer boilerplate configurations.
- **Tailwind CSS:** Utility-first CSS framework that accelerates UI development with built-in responsiveness.

- **Node.js**: Non-blocking I/O, ideal for high-performance, real-time applications.
- **Express.js**: Simplifies REST API creation, supports middleware layering.
- **MongoDB (Mongoose)**: Flexible document-based data model; Mongoose adds structure and validation.
- **JWT**: Provides stateless, secure user authentication.
- **Postman**: Used for API testing and debugging.
- **Git & GitHub**: Version control and team collaboration.

Optional deployment platforms include **Vercel** (for the frontend) and **Render/Heroku** (for the backend).

Setup Instructions

To run DocSpot locally or on a cloud environment, follow the structured setup below:

System Prerequisites:

- **Node.js**: Version 16 or higher is recommended.
- **npm or Yarn**: For dependency management.
- **MongoDB Atlas** or local MongoDB instance.
- **Code Editor**: VS Code is recommended.
- **Browser**: Modern browsers like Chrome, Firefox, or Edge.

Step-by-Step Installation:

1. Clone the project repository from GitHub:

```
https://github.com/bMadhavvarma/Docspot.git
```

2. Navigate to the client folder and install frontend dependencies:

```
cd client
```

```
npm install
```

3. Navigate to the server folder and install backend dependencies:

```
cd ../server
```

```
npm install
```

4. Set up environment variables by creating a .env file in both client and server directories.

Frontend .env:

```
REACT_APP_API_URL=http://localhost:5000/api/v1
```

Backend .env:

PORT=5000

MONGO_URI=<Your MongoDB Connection String>

JWT_SECRET=<Your JWT Secret Key>

JWT_EXPIRES_IN=1d

5. Start the development servers:

- o Frontend: cd client && npm start
- o Backend: cd server && npm start

Visit the application on <http://localhost:3000>.

API Documentation

DocSpot's backend exposes a RESTful API architecture that adheres to modern standards of scalability, modularity, and clarity. APIs are segmented into user-specific endpoints, allowing Admins, Doctors, and Patients to access only the routes relevant to their roles. Every endpoint follows proper HTTP standards with appropriate request methods (GET, POST, PUT, DELETE).

User APIs:

- POST /signup – Registers a new patient or doctor based on the role selected.
- POST /login – Authenticates user and issues JWT token for access.
- GET / – Fetches a list of all users (admin access only).
- GET /:id – Fetches user details by unique ID.
- DELETE /:id – Deletes a user account.
- GET /verify-user/:id – Verifies and updates user status (admin only).
- POST /book-appointment – Allows a patient to book an appointment with a doctor.
- GET /user-appointments/:id – Retrieves all appointments linked to a user.
- POST /mark-all-notification-as-seen – Marks notifications as read.
- POST /delete-all-notifications – Deletes all previous notifications.
- POST /change-doctor-status – Admin-level API to approve or reject doctor requests.

Doctor APIs:

- GET / – Retrieves a list of all registered doctors.
- GET /approved-doctors – Fetches only approved and verified doctor profiles.
- POST /signup – Registers a new doctor.
- GET /:id – Retrieves specific doctor information.
- PUT /:id – Updates a doctor's profile or availability status.
- GET /appointments/:id – Gets all appointments associated with a doctor.
- GET /booked-appointments/:id – Lists confirmed appointments for a doctor.
- POST /change-appointment-status – Allows doctors to accept or cancel appointments.
- POST /check-booking-availability – Real-time check to prevent booking conflicts.

All APIs are secured using middleware authentication, and invalid or unauthorized requests return proper status codes like 401 Unauthorized or 403 Forbidden.

Authentication

Authentication in DocSpot is implemented using **JWT (JSON Web Tokens)**, ensuring that each user session is securely handled without the need for repeated database lookups. Upon successful login, the server issues a signed token containing the user's ID and role, which is then stored on the client side (usually in local storage).

Key Features:

- **Stateless Session Management:** No need to store sessions on the server.
- **Role-based Route Protection:** Admin, Doctor, and Patient roles are each restricted to specific components and pages using both backend and frontend route guards.
- **Token Expiry:** JWTs expire in 24 hours and must be refreshed via login, improving security.
- **Route Middleware:** The backend uses middleware to verify tokens and check roles before executing controller logic.
- **Local Storage Usage:** On the frontend, the token is safely stored and used in headers for API requests.

This architecture ensures security, scalability, and the ability to maintain a seamless user experience.

User Interface

DocSpot's user interface (UI) is designed to be both elegant and intuitive. Built with **Tailwind CSS**, it ensures that the application looks modern across all devices and screen sizes. The layout adapts automatically for mobile users, ensuring accessibility and usability for a wide demographic of patients and healthcare professionals.

Core Pages:

- **Login & Registration:** Clean and simple forms with field validation.
- **Doctor Directory:** Grid layout displaying doctors with images, specializations, and availability.
- **Appointment Booking:** Calendar-based slot selection, along with real-time conflict detection.
- **Admin Dashboard:** Includes user metrics, doctor approval panel, and system health insights.
- **Doctor Dashboard:** Lists upcoming appointments, patient information, and availability toggles.
- **Notification Center:** Bell icon component with dropdown notifications for all roles.

UI components follow a consistent design system with reusable elements like buttons, modals, inputs, and alerts. Animations and transition effects add subtle interactivity without overwhelming the user.

Testing

A strong emphasis was placed on testing throughout the development lifecycle to ensure functionality, security, and usability.

API Testing:

- **Tool Used:** Postman
- Every endpoint was rigorously tested with different roles and permission levels.
- Negative tests (invalid tokens, expired tokens, incorrect payloads) were performed to ensure the system handles errors gracefully.

Manual Testing:

- Conducted across Chrome, Firefox, and mobile browsers.
- All major user workflows were tested manually including:
 - Login/Signup flows
 - Booking and cancellation
 - Admin approvals
 - Role switching and route protection
- Edge cases, such as booking overlapping appointments, were tested.

Planned Automation (Future Scope):

- Jest or Mocha for unit testing backend logic.
- Cypress for E2E frontend testing.

Screenshots or Demo

The following screenshots from the DocSpot platform demonstrate its core features, responsive UI, and overall user experience. Each page has been carefully designed to cater to its respective user role, ensuring accessibility and efficiency.

Login Page:

The login page features a white header bar with the word "Login". Below it is a form with two input fields: "Email" and "Password", each with a placeholder and a clear button. A blue "Login" button is centered below the fields. To the right is a large, colorful gradient background with a circular logo containing a stylized heart or DNA helix.

Welcome to MD Clinic
New here? [Create a new account](#)

Login

Email

Password

Login

Welcome to MD Clinic

New here? Create a new account

Sign Up Page:

The sign up page has a white header bar with the text "Create an Account". Below it is a form with four input fields: "Name", "Email", "Mobile Number", and "Password", each with a placeholder and a clear button. A blue "Sign Up" button is centered below the fields. To the left is a large, colorful gradient background with a circular logo containing a stylized heart or DNA helix.

Create an Account

Name

Email

Mobile Number

>Password

Sign Up

Get Started

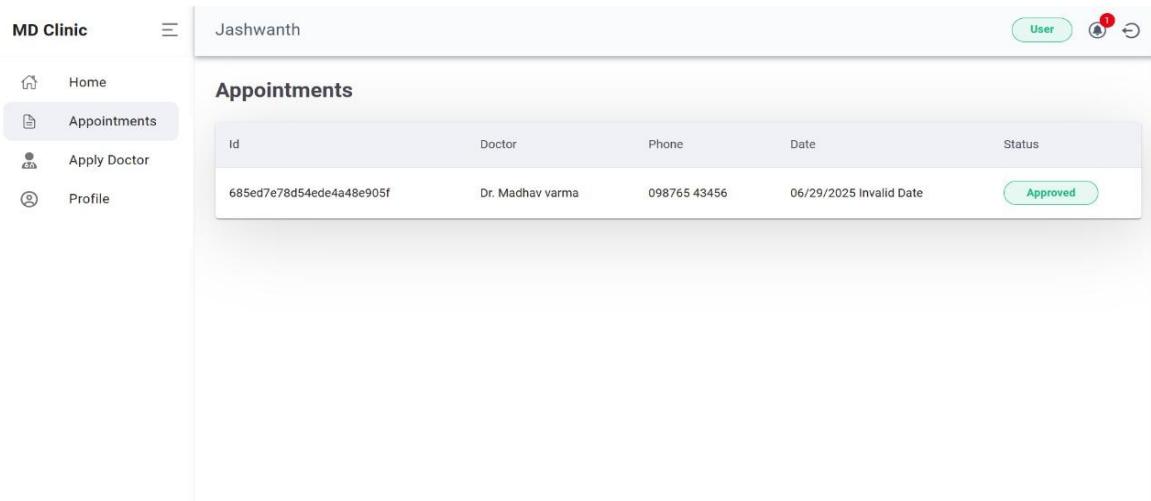
Already have an account ? [Login](#)

User dashboard:

The user dashboard shows a sidebar with navigation links: "MD Clinic", "Home", "Appointments", "Apply Doctor", and "Profile". The main area displays a list of available doctors with their contact information and appointment details.

Doctor Name	Specialty	Phone Number	Address	Fee Per Visit	Timings
Dr. Madhav varma	(Neurologist)	098765 43456	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Divya	(psychiatrist)	092432 45645	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Jeswanth	(cardiology)	092432 45699	Renigunta	499	9:00 AM to 7:00 PM
Dr. Suma	(ENT Specialist)	076857 88699			
Dr. Aarshita	(Dentist)	092432 45888			
Dr. D Jashwanth	(Surgeon)	072074 79789			

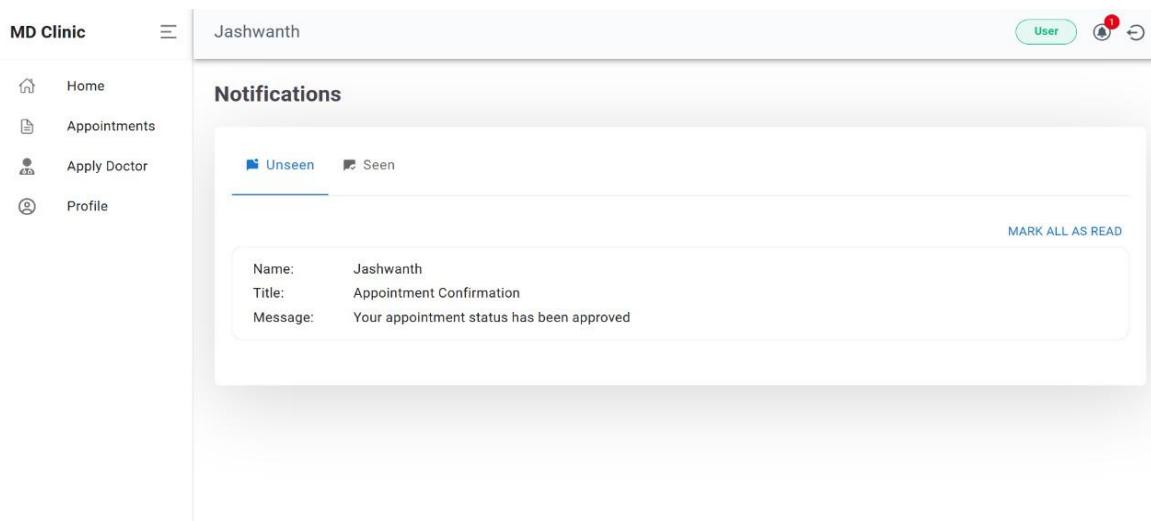
User Appointments status:



The screenshot shows the user's appointment details. The appointment is listed as approved.

ID	Doctor	Phone	Date	Status
685ed7e78d54ede4a48e905f	Dr. Madhav varma	098765 43456	06/29/2025 Invalid Date	Approved

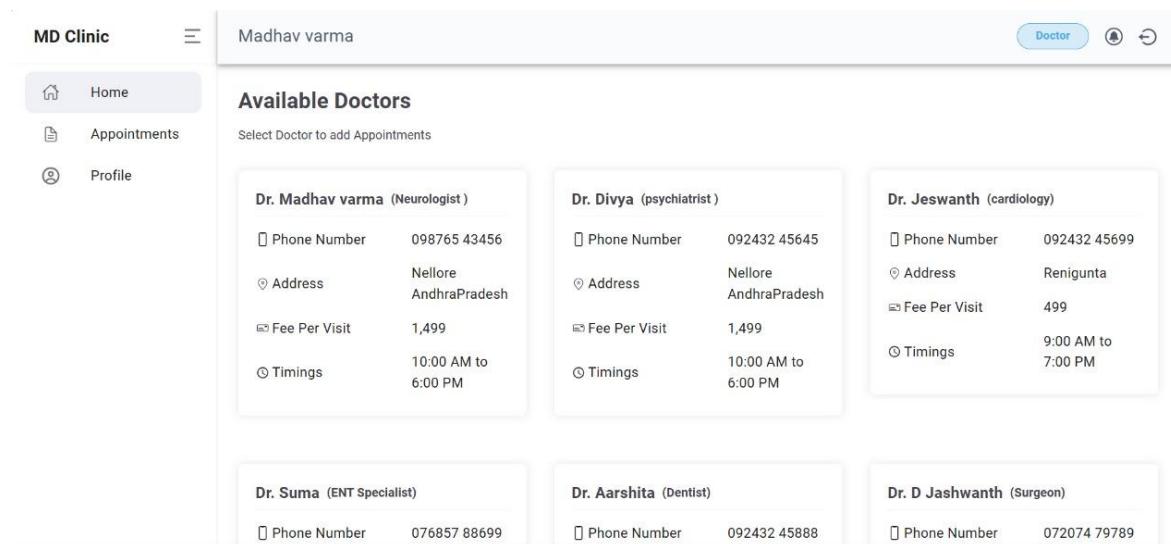
User Notifications:



The screenshot shows a notification message from the clinic. The message is an appointment confirmation stating that the user's appointment status has been approved.

Name: Jashwanth
Title: Appointment Confirmation
Message: Your appointment status has been approved

Doctor Dashboard:



The screenshot shows the doctor dashboard with a list of available doctors. Dr. Madhav varma is selected.

Doctor Name	Specialty	Phone Number	Address	Fee Per Visit	Timings
Dr. Madhav varma	(Neurologist)	098765 43456	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Divya	(psychiatrist)	092432 45645	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Jeswanth	(cardiology)	092432 45699	Renigunta	499	9:00 AM to 7:00 PM
Dr. Suma	(ENT Specialist)	076857 88699			
Dr. Aarshita	(Dentist)	092432 45888			
Dr. D Jashwanth	(Surgeon)	072074 79789			

Doctor Appointments:

The screenshot shows the 'Appointments' section for Dr. Madhav Varma. There are two entries in the table:

ID	Patient	Phone	Date	Status	Actions
685ecc155287bec926ad9975	jashwanth d	072074 79740	06/28/2025 Invalid Date	Approved	✓
685ed7e78d54ede4a48e905f	Jashwanth	072074 79729	06/29/2025 Invalid Date	Approved	✓

Doctor's Profile:

The screenshot shows the 'Profile' section for Dr. Madhav Varma. It is divided into two tabs: 'Basic Information' and 'Professional Information'.

Basic Information:

Prefix	Full Name	Mobile Number
Dr.	Madhav varma	+91 98765-43456
Website	Address	
https://testbook.com/10th-based-gover	Nellore AndhraPradesh	

Professional Information:

Specialization	Experience	Fee Per Consultation
Neurologist	4	1499
Start Time	End Time	

Admin Dashboard:

The screenshot shows the 'Available Doctors' section of the Admin dashboard. It lists several doctors with their contact information and availability.

Doctor	Specialization	Phone Number	Address	Fee Per Visit	Timings
Dr. Madhav varma	(Neurologist)	098765 43456	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Divya	(psychiatrist)	092432 45645	Nellore AndhraPradesh	1,499	10:00 AM to 6:00 PM
Dr. Jeswanth	(cardiology)	092432 45699	Renigunta	499	9:00 AM to 7:00 PM
Dr. Suma	(ENT Specialist)	076857 88699			
Dr. Aarshita	(Dentist)	092432 45888			
Dr. D Jashwanth	(Surgeon)	072074 79789			

User's Controls by Admin:

Name	Email	Date	Roles	Actions
Admin	admin@gmail.com	06/27/2025, 4:21 PM	Owner	
Madhav varma	doctor1@gmail.com	06/27/2025, 4:24 PM	Doctor	
Divya	doctor2@gmail.com	06/27/2025, 4:29 PM	Doctor	
Jeswanth	doctor3@gmail.com	06/27/2025, 4:31 PM	Doctor	
Suma	doctor4@gmail.com	06/27/2025, 4:33 PM	Doctor	
Aarshita	doctor5@gmail.com	06/27/2025, 4:36 PM	Doctor	
patient1	patient1@gmail.com	06/27/2025, 4:50 PM	User	Delete

Doctor's Controls by Admin:

Name	Specialty	Email	Phone Number	Date	Status	Actions
Dr. Madhav varma	Neurologist	doctor1@gmail.com	098765 43456	06/27/2025, 4:26 PM	Approved	Block
Dr. Divya	psychiatrist	doctor2@gmail.com	092432 45645	06/27/2025, 4:30 PM	Approved	Block
Dr. Jeswanth	cardiology	doctor3@gmail.com	092432 45699	06/27/2025, 4:33 PM	Approved	Block
Dr. Suma	ENT Specialist	doctor4@gmail.com	076857 88699	06/27/2025, 4:35 PM	Approved	Block
Dr. Aarshita	Dentist	doctor5@gmail.com	092432 45888	06/27/2025, 4:38 PM	Approved	Block
Dr. D Jashu	Surgeon	jashu123@gmail.com	072074 79749	06/27/2025, 10:28 PM	Pending	Approve
Dr. D Jashwanth	Surgeon	jashu66@gmail.com	072074 79789	06/27/2025, 11:18 PM	Approved	Block

Future Enhancements

Several enhancements have been planned to extend the functionality and robustness of the platform:

- Doctor Search & Filter:** Introduce advanced filters based on gender, rating, fee, experience.
- Upload & Share Documents:** Enable report, test result, and prescription sharing.
- Live Chat or Teleconsultation Module:** Optional video/audio calling features for remote consultation.
- WebSocket Integration:** Real-time updates for appointment confirmations and notifications.
- Graph-based Analytics Dashboard:** Integrate chart libraries for visual admin insights.
- Email & SMS Integration:** Appointment reminders and verification messages.

- **Multi-language Support:** Expand accessibility with translation features.
- **Accessibility Compliance:** Ensure the site follows WCAG guidelines for differently-abled users.

Conclusion

DocSpot is not just an appointment scheduler—it's a step toward the future of digital healthcare infrastructure. By combining the robustness of the MERN stack with a user-first design, it provides a scalable foundation for real-world healthcare applications. While some advanced features remain in the pipeline, the current platform is functional, modular, and designed with future extensibility in mind. As telemedicine and digital health services continue to grow, DocSpot is well-positioned to evolve alongside this trend.

Deployed Application Link

You can access the live application here:

<https://docspot-mauve.vercel.app/login>