# DocSpot: Seamless Appointment Booking for Health

DocSpot is a full-stack web application designed to streamline the doctor appointment process. It supports multiple roles—Admin, Doctor, and Customer (Patient)—with real-time booking, verification, profile management, and analytics. It features a modern UI with animated dashboards and supports both online and offline workflows.

## Features

### Admin

* Approves/Rejects doctors
* Views platform analytics (users, appointments, earnings)
* Manages all users and doctors
* Access to dynamic dashboards

### Doctor

* Manages profile
* Views appointments
* Uploads documents
* Real-time appointment list
* Beautiful doctor dashboard with visualizations

### Customer (Patient)

* Registers/logins
* Books appointments
* Views appointment history
* Edits profile
* Stunning customer dashboard UI

### Smart Add-ons

* Role-based dashboards with Tailwind CSS and Framer Motion
* Profile editing (for Doctor & Customer)
* Home button (left) & profile icon (right) on all dashboards
* Authentication & Authorization via JWT
* File Upload support
* RESTful APIs with modular structure

## Tech Stack

| Area | Technology |
| --- | --- |
| Frontend | React + Vite + Tailwind CSS |
| Animations | Framer Motion |
| Icons | Lucide Icons, React Icons |
| Backend | Node.js + Express.js |
| Database | MongoDB + Mongoose |
| Auth | JWT (Role-based) |
| Image Upload | Multer (backend), stored locally |
| Deployment | Vercel (frontend), Render/Heroku (backend) |

## Installation & Running

### 1. Clone the Repo

git clone https://github.com/Manideepchopperla/DocSpot

cd docspot

### 2. Install Backend

cd server

npm install

### 3. Install Frontend

cd ../client

npm install

### 4. Environment Variables

Create .env file in /backend:

PORT=5000

MONGO\_URI=your\_mongo\_db\_uri

JWT\_SECRET=your\_jwt\_secret

### 5. Run Servers

#### Backend:

npm run dev

#### Frontend:

npm run dev

**Frontend Demo :**[**DocSpot**](https://doc-sopt.vercel.app/)

## **Frontend – React + Vite + Tailwind CSS**

### **Structure (/client)**

client/

├── public/

│ └── Doctor.png # Landing page image

├── src/

│ ├── components/ # Reusable UI components

│ │ ├── ui/ # Buttons, Cards, Tables, etc.

│ │ ├── ProfileModal.jsx # For editing user/doctor profile

│ │ └── ProfileDropdown.jsx # Profile dropdown icon (top right)

│ ├── pages/

│ │ ├── Dashboards/

│ │ │ ├── AdminDashboard.jsx

│ │ │ ├── DoctorDashboard.jsx

│ │ │ └── CustomerDashboard.jsx

│ │ ├── LoginPage.jsx

│ │ ├── RegisterPage.jsx

│ │ └── LandingPage.jsx

│ ├── App.jsx # App routes

│ └── main.jsx # Entry point

├── .env # Vite environment variables

├── tailwind.config.js # Tailwind customization

├── postcss.config.js

└── vite.config.js # Aliases, plugins, and build config

### **Key Features**

* Responsive **Landing Page** with smooth scroll and animated sections
* **Role-based dashboards** with:  
  + Home button (left)
  + Profile icon (right) with edit option (except admin)
  + Tab navigation (Overview, Analytics, etc.)
  + Animated cards, badges, tables
* Form validation on login & registration
* Typing effects, framer motion animations, icons (Lucide, React Icons)
* Modern UI with Tailwind CSS and custom themes

### **Libraries Used**

* [react](https://reactjs.org/)
* [react-router-dom](https://reactrouter.com/)
* [framer-motion](https://www.framer.com/motion/)
* [react-icons](https://react-icons.github.io/react-icons/)
* [react-scroll](https://www.npmjs.com/package/react-scroll)
* [tailwindcss](https://tailwindcss.com/)
* [lucide-react](https://lucide.dev/)

## **Backend – Express + MongoDB + JWT**

### **Structure (/backend)**

backend/

├── config/

│ └── connectToDB.js # MongoDB connection logic

├── controllers/

│ ├── adminC.js # Admin functionality (approvals, stats)

│ ├── doctorC.js # Doctor profile & appointments

│ └── userC.js # Customer booking, profile

├── routes/

│ ├── adminRoutes.js # /api/admin/\*

│ ├── doctorRoutes.js # /api/doctor/\*

│ └── userRoutes.js # /api/user/\*

├── schemas/

│ ├── appointmentModel.js # Appointment schema

│ ├── userModel.js # Users (doctors/customers/admins)

│ └── docModel.js # Doctor uploads (optional)

├── middlewares/

│ └── authMiddleware.js # JWT token verification

├── uploads/ # Uploaded documents/images

├── .env # Env vars (PORT, MONGO\_URI, JWT\_SECRET)

├── .gitignore

├── index.js # Main Express entry

└── package.json

### **API Overview**

| **Method** | **Route** | **Description** |
| --- | --- | --- |
| POST | /api/user/login | Login (role-based) |
| POST | /api/user/register | User/Doctor registration |
| GET | /api/doctor/profile | Fetch doctor profile |
| PUT | /api/doctor/profile | Update doctor profile |
| GET | /api/admin/dashboard | Admin dashboard analytics |
| PUT | /api/admin/approve/:id | Approve doctor request |
| POST | /api/user/book | Patient books appointment |
| GET | /api/user/appointments | List patient’s appointments |

### **Authentication**

* JWT-based authentication
* Token middleware (authMiddleware.js)
* Role-based access per route
* Secure file uploads via Multer (if implemented)

### **Backend Dependencies**

* [express](https://expressjs.com/)
* [mongoose](https://mongoosejs.com/)
* [jsonwebtoken](https://github.com/auth0/node-jsonwebtoken)
* [cors, dotenv, multer, bcryptjs]