

DocSpot: Seamless Appointment Booking for Health

1. Introduction

Project Title: DocSpot: Seamless Appointment Booking for Health

Team ID : LTVIP2026TMIDS44539

Team Leader : Katta Karthik

Team member : Veerla Hanumath Kumar

Team member : Vanabathina Saritha

Team member : Kocherla Venkata Hemanth Kumar

Overview:

DocSpot is a full-stack healthcare web application that streamlines the process of booking medical appointments.

It provides a user-friendly interface for patients to book appointments online, doctors to manage their schedules,

and administrators to oversee system operations. The app includes secure role-based logins and integrated online payments.

2. Project Overview

Purpose:

To provide a centralized, convenient, and secure platform for healthcare appointment management accessible by patients, doctors, and administrators.

Key Features:

- User (Patient) registration, login, and appointment booking
- Doctor login and appointment management dashboard
- Admin panel to manage users and appointments

- Razorpay integration for online payments
- Secure authentication using JWT

3. System Architecture

Frontend (React.js):

- Built with React and React Router
- UI components from Material UI, Bootstrap, Ant Design
- Axios for HTTP requests
- Responsive design

Backend (Node.js + Express.js):

- RESTful API architecture
- Role-based access control
- Razorpay payment API integration
- JWT authentication

Database (MongoDB):

- Collections: users, doctors, appointments, admins, payments
- Mongoose for schema definitions and data modeling

4. Setup Instructions

Prerequisites:

- Node.js (v18+)
- MongoDB (Local or Atlas)
- npm, Git

Setup:

git clone https://github.com/your-username/docspot.git

Frontend:

cd client

npm install

```
PS D:\FullStack\docspot-seamless-appointment-booking-for-health\Project File> cd frontend
PS D:\FullStack\docspot-seamless-appointment-booking-for-health\Project File\frontend> npm run dev

> frontend@0.0.0 dev
> vite

VITE v5.4.19 ready in 1122 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Backend:

cd ../server

npm install

```
PS D:\FullStack\docspot-seamless-appointment-booking-for-health\Project File> cd backend
PS D:\FullStack\docspot-seamless-appointment-booking-for-health\Project File\backend> npm run dev

> backend@1.0.0 dev
> nodemon server.js

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Server started on PORT:4000
Database Connected
```

.env file:

MONGO_URI=your_mongo_uri

JWT_SECRET=your_jwt_secret

RAZORPAY_KEY=your_razorpay_key

RAZORPAY_SECRET=your_razorpay_secret

PORT=5000

5. Folder Structure

client/

└─ public/

└─ src/

│ └─ components/

│ └─ pages/

│ └─ context/

│ └─ App.js, index.js

server/

└─ controllers/

└─ routes/

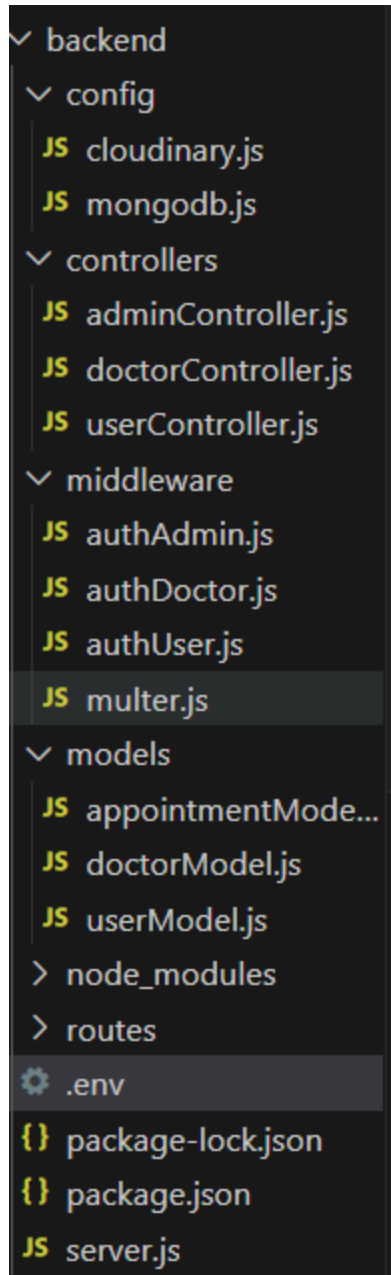
└─ models/

└─ config/

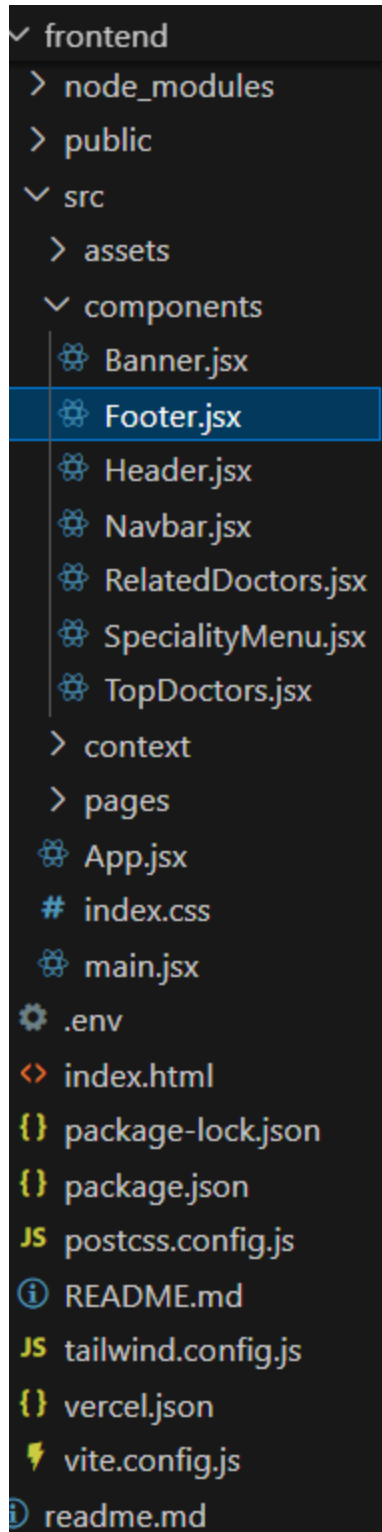
└─ middlewares/

└─ server.js

Backend Folder Structure:



FRONTEND FOLDER STRUCTURE:



6. Running the Application

Frontend:

cd client

npm start

Backend:

cd server

npm start

7. API Documentation

POST /api/users/register - Register a new user

POST /api/users/login - Login as user

POST /api/doctors/login - Doctor login

POST /api/admin/login - Admin login

POST /api/appointments/book - Book an appointment

GET /api/doctors/:id/appointments - Doctor views appointments

GET /api/admin/appointments - Admin views all appointments

POST /api/payments/order - Initiate Razorpay order

POST /api/payments/verify - Verify payment

8. Authentication & Authorization

- JWT Tokens used for secure authentication

- Protected Routes: Based on roles (User, Doctor, Admin)

- Token Flow: On login, token is generated and stored in local storage. Every protected request includes this token in the headers.

9. User Interface Screenshots

User Login:

Prescripto

HOME ALL DOCTORS ABOUT CONTACT

Create account

Create Account

Please sign up to book appointment

Full Name

Email

Password

Create account

Already have an account? [Login here](#)

User Dashboard:

Prescripto

HOME ALL DOCTORS ABOUT CONTACT

Profile icon

Gopal

CONTACT INFORMATION

Email id: shyam@gmail.com

Phone: 000000000

Address:

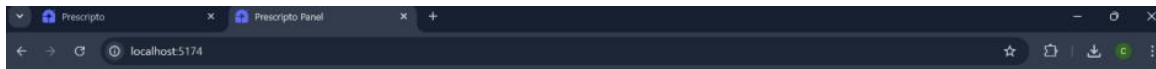
BASIC INFORMATION

Gender: Not Selected

Birthday: Not Selected

Edit

Admin Login Page:



Admin Login

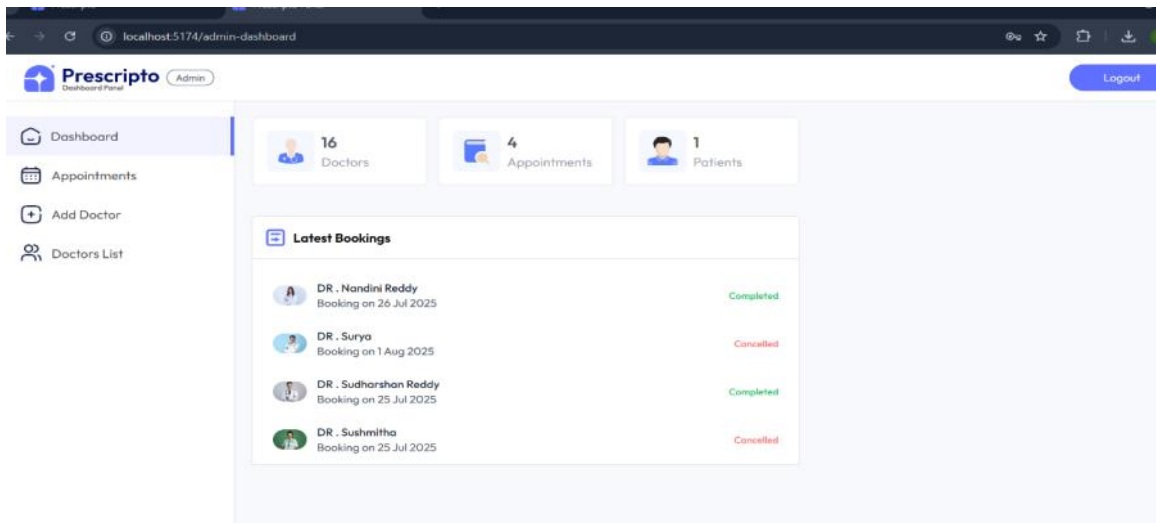
Email

Password

[Login](#)

Doctor Login? [Click here](#)

Admin Dashboard Page:



Doctor Login Page:



Doctor Login

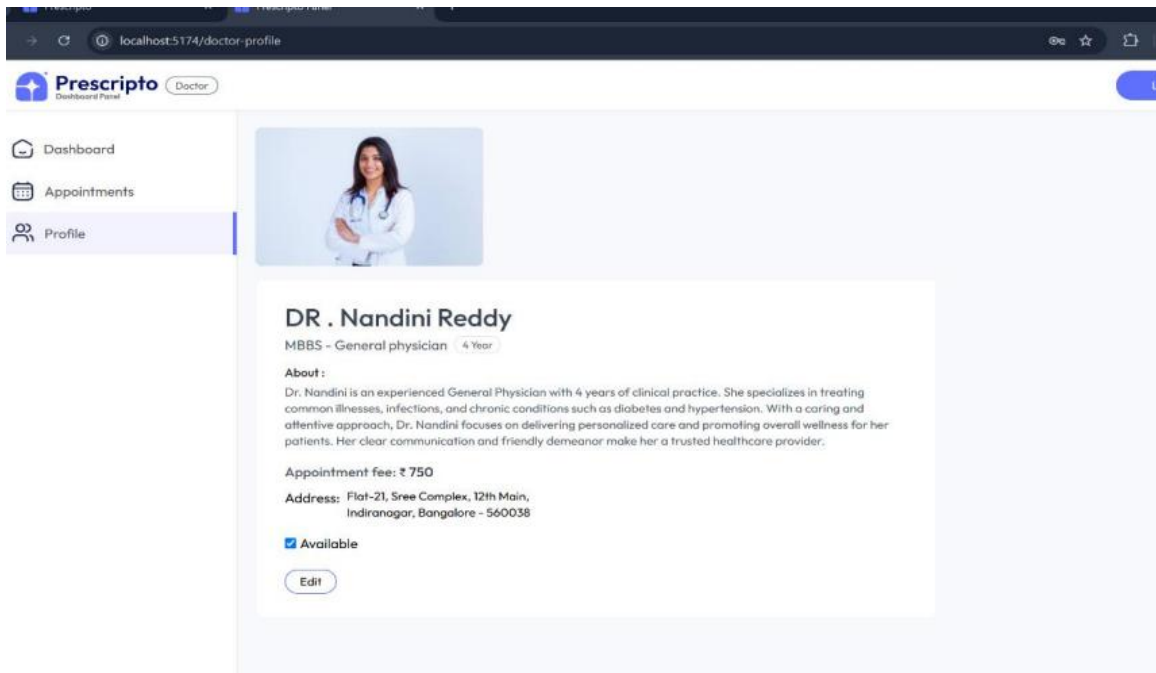
Email

Password

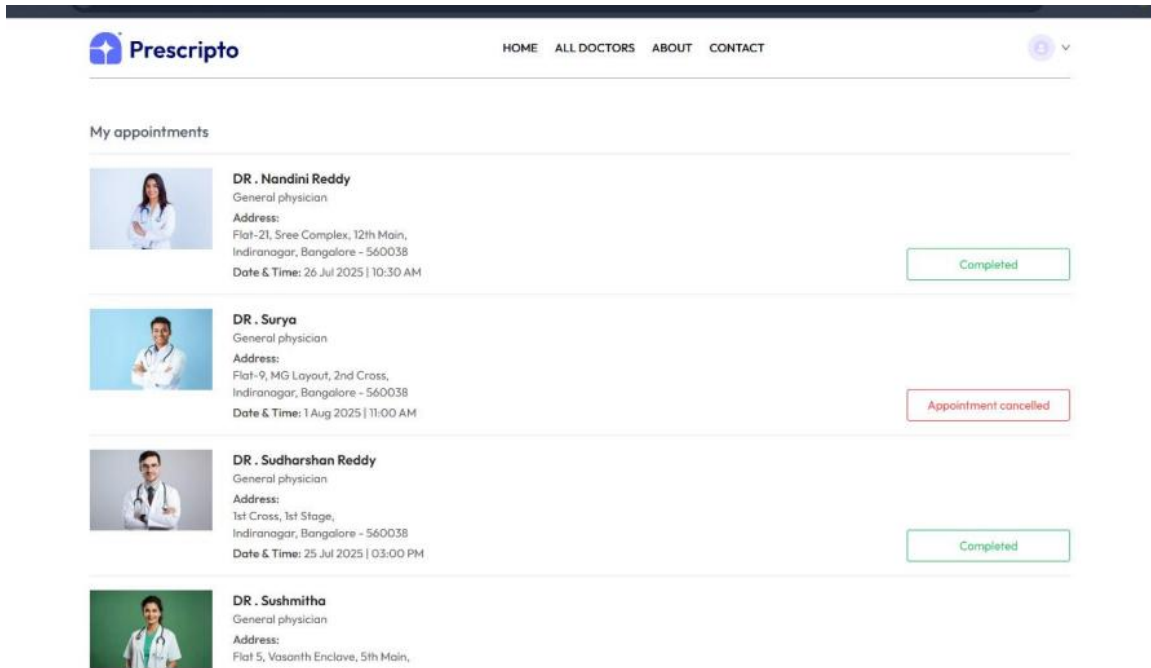
Login

Admin Login? [Click here](#)

Doctor Dashboard:



Appointment Booking:



10. Testing

Tools Used:

- Postman (API testing)
- Lighthouse (Frontend performance)
- Browser DevTools

Coverage:

- User login/logout and protected routes
- Doctor and admin access control
- Appointment conflict handling
- Payment success/failure scenarios

11. Demo or Showcase

Screenshots: As listed above

Demo Video: [Insert YouTube/Google Drive demo link]

12. Known Issues

- No pagination for appointment list
- Doctor profile editing not implemented
- No email/SMS notifications yet

13. Future Enhancements

- Implement video consultations via WebRTC or Jitsi
- Add appointment filtering (by date/speciality)
- Notifications for appointment reminders
- Email confirmations for patients
- Real-time availability for doctors

14. Conclusion

DocSpot successfully integrates frontend and backend technologies to deliver a secure and user-friendly platform for medical appointment management.

With online payments and role-based control, it lays the groundwork for scalable future enhancements in digital healthcare.