**Full Stack Development with MERN**

**Project Documentation – DocSpot: Seamless Appointment Booking for Health**

**Team Details**

* **Team ID:** LTVIP2025TMID54470
* **Team Size:** 4 Members

**Team Leader:**

* Idamakanti Niharika

**Team Members:**

* Devarapalli Uday Chand
* Galidevara Shanmuka
* G Harsha Vardhan

**1. Introduction**

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

**2. Project Overview**

* **Purpose:**  
  DocSpot is a digital healthcare platform designed to streamline the process of booking doctor appointments. It eliminates manual delays and allows patients to book real-time appointments from the comfort of their home.
* **Key Features:**
  + Secure multi-role registration and login
  + Gmail/Facebook OAuth integration
  + Real-time doctor availability and booking system
  + Admin panel for doctor verification
  + Medical document upload before appointments
  + Notifications for appointment confirmations and updates
  + Visit summary and follow-up support

**3. Architecture**

* **Frontend:** React.js, React Router, Bootstrap, Axios
* **Backend:** Node.js, Express.js, RESTful APIs
* **Database:** MongoDB Atlas using Mongoose ODM

**4. Setup Instructions**

**Prerequisites:**

* Node.js (v16 or later)
* MongoDB / MongoDB Atlas
* Git

**Installation Steps:**

1. Clone the repository:  
   git clone https://github.com/your-repo/docspot.git
2. Install frontend dependencies:
3. cd client
4. npm install
5. Install backend dependencies:
6. cd server
7. npm install
8. Configure .env file inside server directory:
9. MONGO\_URI=your\_mongo\_uri
10. JWT\_SECRET=your\_jwt\_secret
11. EMAIL\_USER=your\_email
12. EMAIL\_PASS=your\_email\_password

**5. Folder Structure**

**Client (React Frontend):**

client/

└── src/

├── components/

├── pages/

├── context/

└── App.js

**Server (Node.js Backend):**

server/

├── routes/

├── controllers/

├── models/

├── middleware/

└── server.js

**6. Running the Application**

**Frontend:**

cd client

npm start

**Backend:**

cd server

npm start

**7. API Documentation**

| **Method** | **Endpoint** | **Description** | **Request Body** |
| --- | --- | --- | --- |
| POST | /api/auth/register | Register new user | name, email, password |
| POST | /api/auth/login | Login user | email, password |
| GET | /api/doctors | Fetch list of doctors | Token |
| POST | /api/appointments | Book appointment | doctorId, date, userId |
| GET | /api/user/history | View past appointments | Token |

**8. Authentication**

* **JWT** for session management
* **bcrypt** for password hashing
* **Role-based access control** for Patients, Doctors, and Admins
* Protected backend routes and React route guards

**9. User Interface**

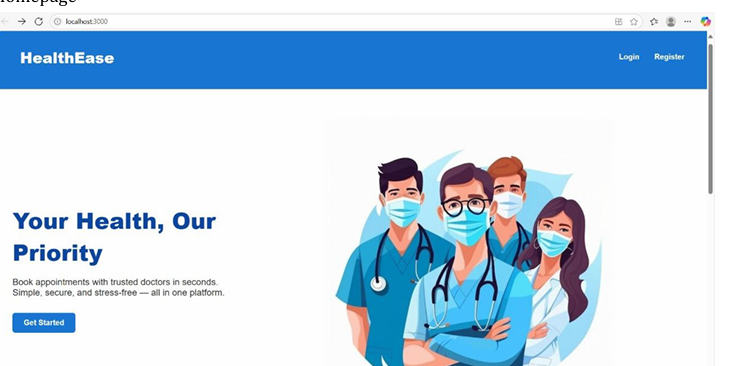
* **Tech Used:** Bootstrap 5 + Custom CSS
* **Pages:**
  + User Registration/Login
  + Role-based Dashboard
  + Doctor Search & Booking
  + Appointment History
  + Admin Panel

**10. Testing**

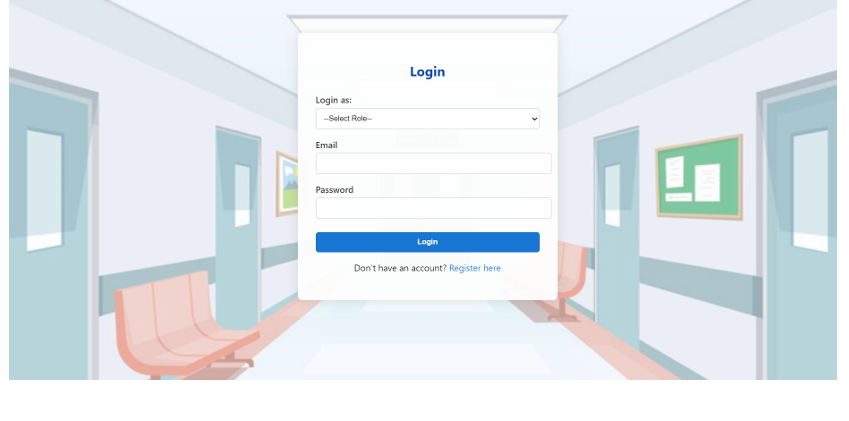
* **Tools:** Postman, Browser DevTools
* **Methods:** Manual testing for user flows, backend API testing using Postman

**11. Screenshots or Demo**

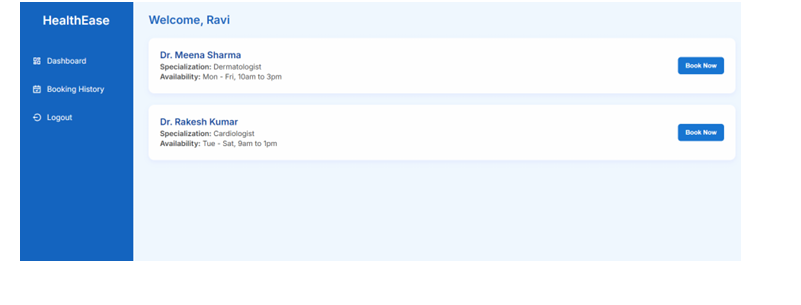
* **Home Page**



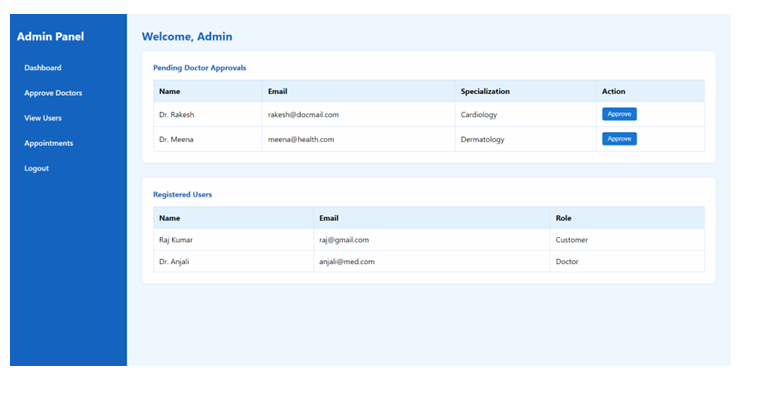
* **Registration /Login Page**



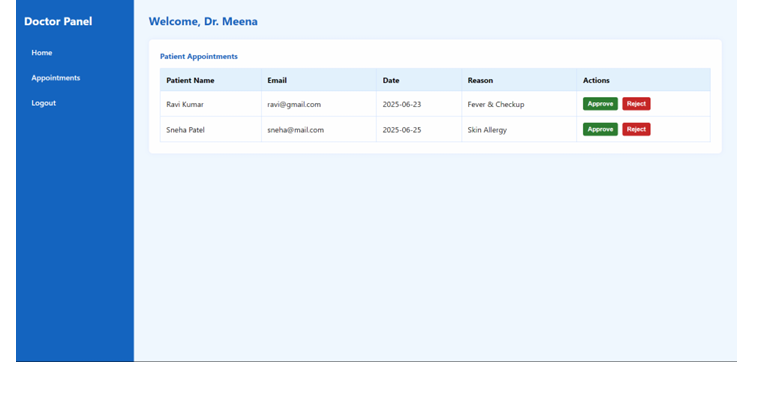
* **User Dashboard:**

****

* **Admin Doctor Approval**



* **Doctor Appointment Management Panel**

****

* **Live Demo (Optional):** https://github.com/Vishnu14git/HealthEase/blob/main/Video%20Demo/healthease.mp4?raw=true

**12. Known Issues**

* SMS OTP not yet implemented
* Chat system pending
* Limited data analytics features

**13. Future Enhancements**

* Video consultation integration (Jitsi Meet / Zoom API)
* Payment Gateway (Razorpay/Stripe)
* Mobile App version (React Native)
* AI-Based Doctor Recommendation System
* Chat system for real-time patient-doctor communication