

DocSpot – Seamless Appointment Booking for Health

Team Details

Team ID: LTVIP2025TMID30950

Team Size: 2

Team Members:

Role	Name
Team Leader	Lakshmi Sravya Savaram
Team Member	Pathela Praveen Chakravarthi

Project Summary

DocSpot is a web-based health appointment booking platform that connects patients with doctors in real time.

The system allows users to search for healthcare professionals by specialty or location, schedule appointments, and communicate securely.

It streamlines the traditional healthcare booking process into a simple and user-friendly interface, saving time and effort for both patients and doctors.

Features

- Doctor and patient registration/login
- Search doctors by specialization/location
- Appointment slot booking and calendar view
- Admin dashboard for monitoring
- Notifications and reminders
- Medical record management
- Secure login using JWT authentication

Tech Stack

Frontend: React.js, HTML, CSS, Bootstrap

Backend: Node.js, Express.js

Database: MongoDB (via Mongoose)

Authentication: JWT-based login system

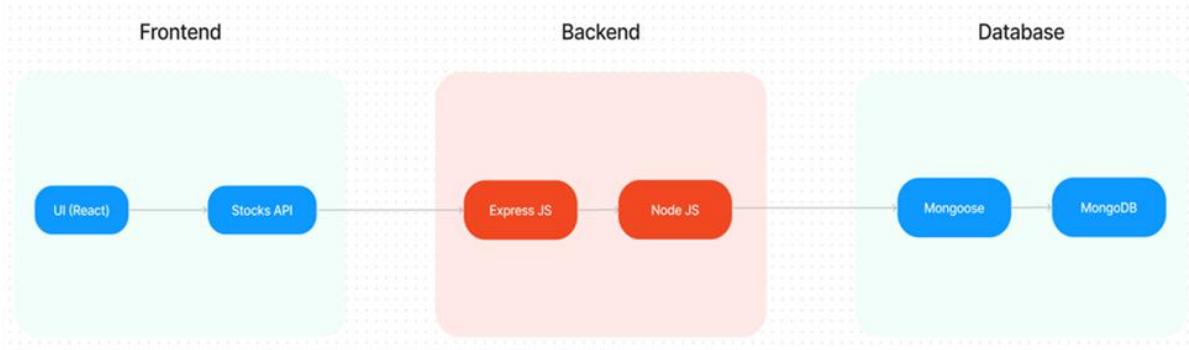
Hosting: AWS, Render, or Vercel

🔒 Authentication Flow

DocSpot uses JWT-based authentication to ensure secure access to user data.

The tokens are generated upon successful login and are used in API headers for protected routes.

TECHNICAL ARCHITECTURE



📝 Testing Strategy

Testing is performed using:

- React Testing Library for frontend components
- Postman for API testing
- Manual Testing for all user flows (booking, login, dashboard)

🚀 Future Scope

- Payment gateway integration (Stripe/Razorpay)
- Real-time chat between doctor & patient
- Email and SMS appointment reminders
- Mobile app with Flutter or React Native

- AI-driven doctor recommendation system

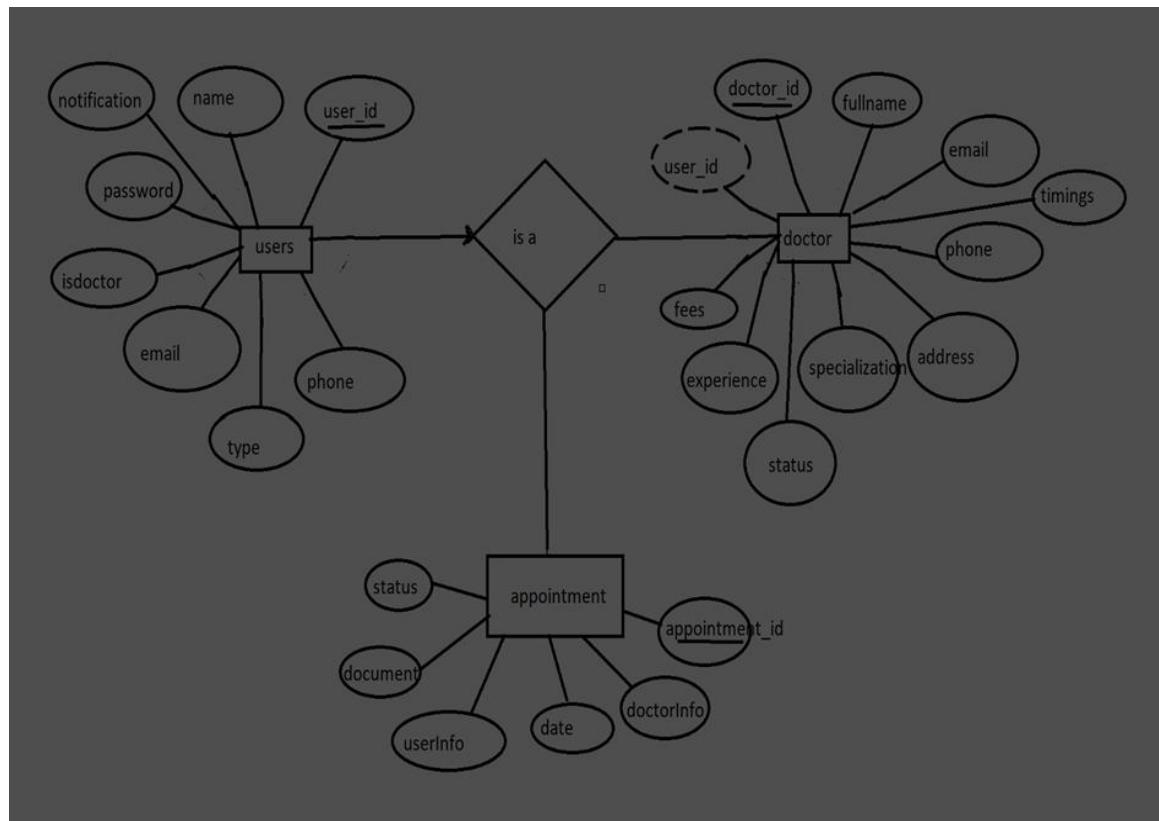
Contact

Lakshmi Sravya Savaram
 Team Leader – DocSpot Project
 [youremail@example.com]

Architecture

DocSpot follows a modern full-stack client-server architecture designed for scalability and performance.

ER DIAGRAM



Frontend (Client):

- Built using React.js with reusable components for login, dashboard, appointment view, etc.
- Axios used for HTTP communication with backend APIs.
- Routes are separated for doctors, patients, and admin access.

Backend (Server):

- RESTful APIs support operations like registration booking, - Developed using Node.js with Express framework.
and calendar management.
- MongoDB handles data persistence using Mongoose ODM.

The screenshot shows a code editor interface with the following details:

- Project Structure:** The left sidebar shows a tree view of the "DOCSPOT" project. It includes a "client" folder containing "client", "node_modules", "public", "src", ".env", ".env.development", ".env.production", ".gitignore", "package.json", "README.md", and "tsconfig.json". It also includes a "server" folder containing "node_modules", "server", "controllers", "models", "routes", "utils", ".env", ".gitignore", "app.js", "package-lock.json", "package.json", "server.js", "package-lock.json", and "package.json".
- Code Editor:** The main area displays the "server.js" file content:

```
server > server > server.js > ...
6   app.get('/', (req, res) => {
7     res.send('Server is up and running!');
8   );
9
10  app.listen(PORT, () => {
11    console.log(`Server is running on http://localhost:${PORT}`);
12  );
13
```
- Terminal:** At the bottom, the terminal window shows the output of the build process:

```
Compiled successfully
You can now view client in the browser.
Local:      http://localhost:3001
On Your Network:  http://192.168.1.12:3001
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
Files successfully emitted, waiting for typecheck results...
Issues checking in progress...
No issues found.
```
- Sidebar:** On the right, there are sections for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". A "PROBLEMS" section shows a single message: "Compiled successfully". The "OUTPUT" section shows the build logs.

Key Modules:

- User Authentication and Role-Based Access
- Doctor Profile and Slot Management
- Appointment Booking Workflow
- Admin Dashboard and Monitoring APIs

Communication Flow:

1. Patient logs in and browses available doctors.
2. API fetches filtered doctors from the database.
3. Patient books a slot and backend confirms availability.
4. Appointment is stored and updated across user dashboards.

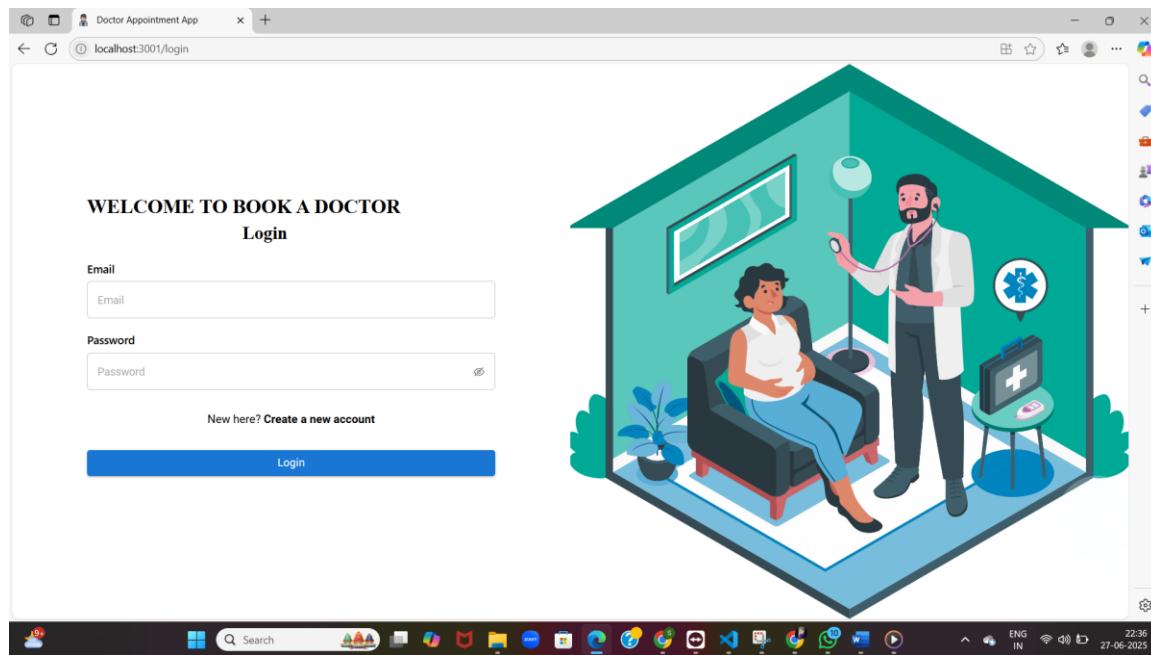
Reflections and Learning

Working on the DocSpot project has been a transformative experience in understanding real-world software development practices. From conceptualizing the initial idea to implementing key features like authentication, role-based access, and real-time booking workflows, each phase of the project contributed significantly to my growth as a full-stack developer

Key areas of learning and improvement:

- **Project Planning:** Learned to break large objectives into manageable modules with realistic timelines.
- **Frontend Mastery:** Gained hands-on experience building responsive, user-friendly interfaces using React and Material UI.
- **Backend Integration:** Deepened my knowledge of RESTful API development, database modeling, and secure authentication using Node.js and MongoDB.
- **Team Collaboration:** Improved communication and coordination while working with team members in a shared codebase.
- **Testing and Debugging:** Developed effective testing strategies using Postman and React Testing Library to improve application quality.

Login page



Learnings or Skills Gained

Summarize personal and team growth:

- Real-world development practices
- Version control (GitHub)
- Agile teamwork and task breakdown
- API integration
- UI/UX design principles

Acknowledgments

We sincerely thank our mentors, faculty, and peers who supported and guided us during the development of this project.

🏁 Conclusion

The **DocSpot** project represents a forward-thinking approach to modern healthcare management by simplifying the way patients connect with doctors. With a focus on user-friendly design, secure data handling, and real-time functionality, this platform is more than just an appointment system — it is a step toward digital transformation in healthcare.

From streamlining medical bookings to enhancing patient-doctor communication, **DocSpot** serves as a foundation for a smarter, faster, and more accessible healthcare experience. The modular and scalable architecture ensures that the system can evolve with future needs, integrating features like teleconsultation, AI-driven suggestions, and multilingual support.

This project not only addresses a critical real-world need but also showcases the power of collaboration, full-stack development, and purpose-driven innovation.

"DocSpot: Empowering Health, One Appointment at a Time."

A Strong Final Statement

End with a confident, reflective, and forward-looking message like:

"DocSpot isn't just a project—it's a commitment to digital health innovation. As healthcare continues to evolve, DocSpot has the potential to scale into a comprehensive platform that bridges gaps between patients and professionals with efficiency, empathy, and technology."