🗎 Solution Architecture Document

🖼 Date: 27 June 2025
🆔 Team ID: LTVIP2025TMID57021
⚒ Project Name: DocSpot – Seamless Appointment Booking for Health
👥 Team Size: 4
👤 Team Leader: Gumpena Hima Vardhini
👥 Team Members: Gogireddy Gowthami, Chitteti Josh Kumar, Peddi Ananth Krishna

---

💡 Solution Architecture

The solution architecture of **DocSpot** bridges healthcare service gaps with scalable web technology. It defines how our system components interact to provide patients, doctors, and administrators with a seamless and secure experience.

🎯 Architecture Goals

- Deliver a **scalable and secure** online booking platform.

- Map out communication between **frontend, backend, and database** layers.

- Ensure **role-based dashboards** for patients, doctors, and admins.

- Enable **JWT authentication** and secure authorization.

- Provide a **modular framework** for future upgrades (e.g., telemedicine, pharmacy integration).

🗎 Technology Stack

- **Frontend**: React.js + Tailwind CSS (responsive UI)

- **Backend**: Node.js + Express.js (API & business logic)

- **Database**: MongoDB Atlas (cloud-hosted)

- **Authentication**: JWT tokens (role-based)

- **Deployment**: Netlify (frontend), Railway / Render (backend)

🏰 System Architecture Overview

DocSpot uses a **modular client-server architecture**:

- Users interact with the **React frontend**, which communicates via RESTful APIs.

- The **Express.js backend** processes data and handles logic.

- **MongoDB** securely stores user details, appointment records, and doctor profiles.

- **JWT** ensures secure, token-based access control.

- An **Email Notification Service** triggers confirmations and reminders.

🚀 Development Phases

1. **User Authentication Module** (JWT-based login/register)
2. **Role-Based Dashboards** for Patients, Doctors, and Admins
3. **Doctor Slot Management** for real-time availability
4. **Appointment Booking & Management**
5. **Admin Analytics & Monitoring Panel**
6. **Email Notification System**

📊 Architecture Diagram