# Project Design Phase

# Solution Architecture

| | |
|---|---|
| Date | 15 JUNE 2025 |
| Team ID | LTVIP2025TMID46139 |
| Project Name | DOC SPOT |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

The solution architecture of DocSpot is a systematic and structured framework that connects a real-world healthcare challenge—delays and inefficiencies in doctor appointments—to a reliable, scalable, and user-friendly technical solution. It is not just about coding the application, but about strategically aligning technology to solve business and social problems effectively.

This architecture is composed of several interconnected layers and sub-processes, each with a specific role in ensuring the system's functionality, security, and scalability.

**Goals of the Solution Architecture in DocSpot:**

- To provide an accessible and real-time platform for patients to book doctor appointments.
- To enable doctors to manage their schedules and view bookings.
- To allow admins to control and monitor doctor approvals and system usage.
- To ensure secure, role-based access and a smooth user experience.
- To develop a scalable system that can expand with new features like video consultations or prescriptions in the future.

**Core Components of the Architecture**:

1. Frontend Layer (User Interface)

Technology: React.js with Material UI / Bootstrap

Purpose: Acts as the interaction point for users. It dynamically renders content for three different user roles:

Patients: Register, login, search doctors, book appointments.

Doctors: Manage appointments, set availability.

Admins: View user records, approve/reject doctor accounts.

2. Backend Layer (Application Logic)

Technology: Node.js with Express.js

Purpose: Acts as the central logic unit of the system:

Processes requests from the frontend

Implements business rules (appointment validation, user roles, admin actions)

Secures routes using authentication and middleware

Sends and receives data from the database.

3. Database Layer

Technology: MongoDB with Mongoose

Purpose: Stores and manages persistent data:

User profiles (patients, doctors, admins)

Appointment schedules

Doctor availability and approval status

All data interactions are schema-validated and secured

4. Authentication & Authorization

Technology: JWT (JSON Web Token)

Purpose:

Authenticate users during login

Assign access based on user role

Protect sensitive API routes from unauthorized acces

5. Admin Control Panel

Purpose:

Oversee the system

Approve or reject doctor registrations

Monitor appointments and users

Ensure security and trustworthiness of the platform

6. Hosting & Deployment

Frontend: Vercel or Netlify

Backend: Render or Railway

Database: MongoDB Atlas

Purpose: To ensure availability, scalability, and performance of the system on the internet.

**Example - Solution Architecture Diagram:**

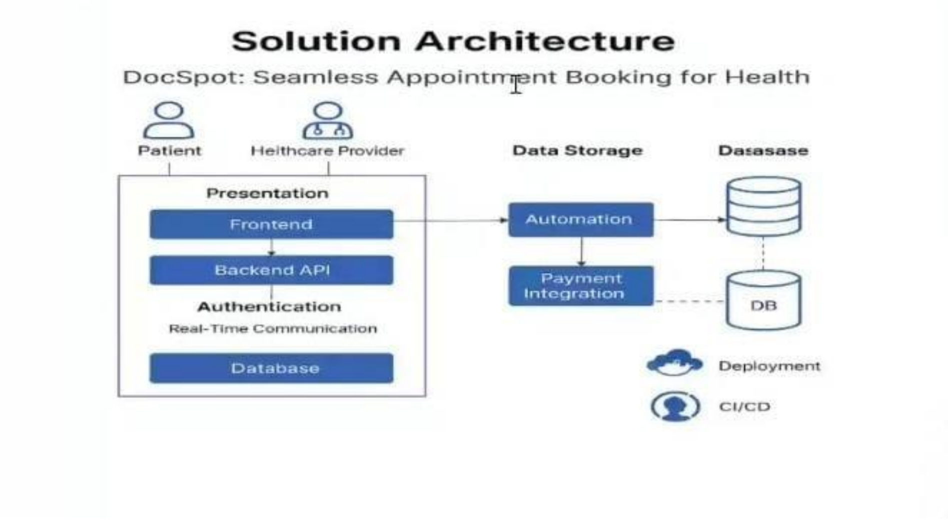

*Figure 1: Architecture and data flow of the seamless appointment booking*