Requirement Analysis

Technology Stack (Architecture & Stack)

Date	23 June 2025
Team ID	LTVIP2025TMID48939
Project Name	Docspot: Seamless Appointment Booking For Health

Technical Architecture – DocSpot

The architecture for DocSpot is designed using the **MERN stack**, offering a scalable and modular solution for digital healthcare appointment booking. It ensures seamless interaction between patients, doctors, admins, and support roles through a secure and responsive web/mobile interface. The system emphasizes real-time bookings, role-based access, and medical data privacy.

Architecture Diagram

Frontend: React.js, Bootstrap, HTML/CSS

Backend: Node.js, Express.js, REST APIs

Database: MongoDB (NoSQL)

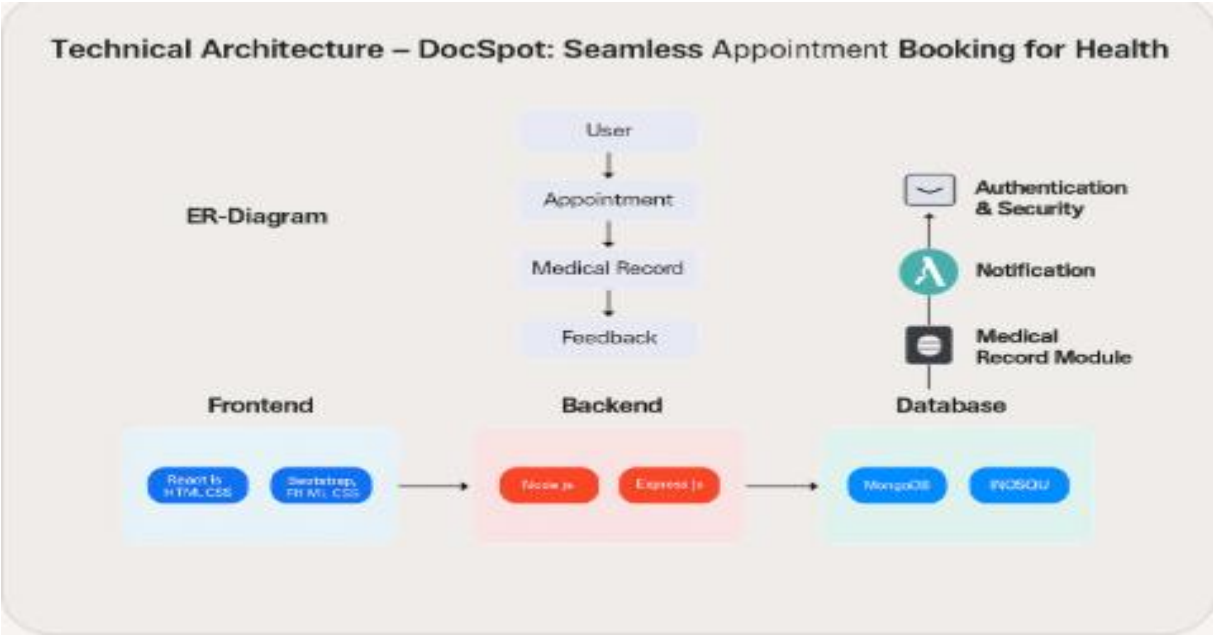

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Description: Handles booking functionality and appointment-related workflows	Technology Used: Node.js, Express.js
3.	Application Logic-2	Manages doctor schedules, availability, and appointment confirmations	Node.js, Express.js
4.	Application Logic-3	Controls doctor approvals, analytics display, and platform governance	Node.js, Express.js
5.	Database	Stores user credentials, appointment records, feedback, and health files.	MongoDB, Mongoose (for schema handling)

6.	Cloud Database	Remote database hosting with auto-scaling capabilities	MongoDB Atlas
7.	File Storage	Upload and view medical documents securely	GridFS (MongoDB), Local File System
8.	External API-1	Sends confirmation messages and reminders to users	Twilio (SMS), Nodemailer (Email)
9.	External API-2	Allows login via Gmail and Facebook credentials	Google OAuth API, Facebook Login API
10.	Machine Learning Model	Enables appointment prediction, health pattern analysis	Python, TensorFlow, Keras
11.	Infrastructure (Server / Cloud)	Hosts application code with CI/CD pipelines for automatic deployment	Vercel (Frontend), Render (Backend), GitHub Actions

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source UI and backend frameworks for faster development and community support	React.js,
2.	Security Implementations	Implements robust authentication and role-based access, data encryption, and secure API practices	JWT, bcrypt, HTTPS, OWASP guidelines, SHA-256 for secure hashing
3.	Scalable Architecture	Modular design using 3-tier architecture and service decomposition for easy scaling	MERN stack, Microservice-ready with MongoDB Atlas cloud hosting
4.	Availability	Hosted on distributed cloud platforms for high uptime and redundancy across deployments	Render, Vercel, Load Balancers, Auto-scaling groups

S.No	Characteristics	Description	Technology
5.	Performance	Optimized APIs and frontend rendering, use of caching and future CDN integration to reduce latency	Express.js (API handling), MongoDB indexing, Redis (optional caching), CDN (future scope)