Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	20 February 2026
Team ID	LTVIP2026TMIDS56110
Project Name	DocSpot: Seamless Appointment Booking for Health
Maximum Marks	4 Marks

Technical Architecture:

DocSpot follows a 3-Tier Architecture (Client–Server–Database Model):

1. **Presentation Layer (Frontend)**
   - Built using React.js
   - Provides UI for Patients, Doctors, and Admin
   - Handles form submissions, appointment booking interface, dashboards
2. **Application Layer (Backend / API Server)**
   - Built using Node.js & Express.js
   - Handles business logic
   - Manages authentication, appointment processing, doctor management
   - Provides REST APIs
3. **Data Layer (Database)**
   - MongoDB (NoSQL Database)
   - Stores users, doctors, appointments, authentication data

Architecture Flow

1. User interacts with React frontend

2. Frontend sends API request to Express server

3. Backend processes request & interacts with MongoDB

4. Database returns response, Backend sends response to frontend

5. UI updates dynamically

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web-based UI for patients, doctors, admin	HTML, CSS, JavaScript, React.js
2	Application Logic-1	Authentication & User Management	Node.js, Express.js
3	Application Logic-2	Appointment Booking & Scheduling Logic	Node.js, Express.js
4	Application Logic-3	Role-Based Access Control (Admin/Doctor/Patient)	JWT, Middleware
5	Database	Stores user, doctor & appointment data	MongoDB (NoSQL)
6	Cloud Database	Managed cloud database service	MongoDB Atlas
7	File Storage	Store doctor profile images	Cloud Storage / Local File System
8	External API-1	Email Notification Service	Nodemailer / SMTP Service
9	External API-2	Payment Integration (Future Scope)	Razorpay API (Optional)
10	Machine Learning Model	(Future Scope) Doctor Recommendation System	Basic Recommendation Algorithm
11	Infrastructure (Server/Cloud)	Application deployment	Local Server / Render / Vercel / AWS

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	Uses open-source MERN stack technologies	React.js, Node.js, Express.js, MongoDB

S.No	Characteristics	Description	Technology Used
2	Security Implementations	Secure authentication & data protection	JWT, bcrypt password hashing, HTTPS, CORS
3	Scalable Architecture	3-Tier architecture allows independent scaling of frontend & backend	MERN Stack, REST APIs
4	Availability	Cloud deployment ensures 24/7 availability	MongoDB Atlas, Cloud Hosting
5	Performance	Optimized API calls & database queries	Indexed MongoDB, Async/Await, Caching

Architectural Justification

The MERN stack architecture was chosen because:

- It is fully JavaScript-based (easy integration)
- Supports scalability
- Fast development & maintenance
- Suitable for real-time dynamic web applications
- Cloud deployment ready