

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	15 February 2026
Team ID	LTVIP2026TMIDS80358
Project Name	Orderonthego : your on-demand food ordering solution
Maximum Marks	4 Marks

Technical Architecture: -

The DocSpot application follows a 3-Tier Client–Server Architecture:

- Presentation Layer (Frontend): React.js web interface where users, doctors, and admins interact with the system.
- Application Layer (Backend): Node.js and Express.js handle API requests, authentication, appointment processing, and role-based access control.
- Data Layer (Database): MongoDB stores user profiles, doctor details, appointment records, and notifications.

The frontend communicates with backend REST APIs using Axios, while JWT authentication secures protected routes. File uploads such as medical documents are handled through Multer middleware.

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	Web interface for patients, doctors, and admin dashboards	React.js, HTML, CSS, Bootstrap, Material UI
2.	Application Logic-1	Authentication & Role Management	Node.js, Express.js, JWT
3.	Application Logic-2	Appointment Booking & Scheduling	Express.js REST APIs

4.	Application Logic-3	Notification & Status Management	Node.js Controllers
5.	Database	Stores users, doctors, appointments data	MongoDB, Mongoose
6.	Cloud Database	Cloud-hosted NoSQL database for storing users, doctors, appointments, and notifications with remote access and scalability	MongoDB Atlas
7.	File Storage	Medical document upload & storage	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	HTTP communication between frontend & backend	Axios
9.	External API-2	Not Applicable	—
10.	Machine Learning Model	Not used in this project	—
11.	Infrastructure (Server / Cloud)	Local development deployment	Node.js Local Server

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used to build UI & Backend APIs	React.js, Express.js, Node.js
2.	Security Implementations	Role-based authentication, encrypted passwords, protected routes	JWT, bcryptjs, Middleware
3.	Scalable Architecture	3-Tier architecture separating UI, backend logic, and database	REST Architecture, MongoDB
4.	Availability	Web application accessible anytime through browser	Node.js Server
5.	Performance	Fast API responses and asynchronous communication	Axios, Express.js