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

Date	26-05-2025
Team ID	LTVIP2025TMID58998
Project Name	DocSpot: Seamless appointment Booking for Health
Maximum Marks	4 Marks

Technical Architecture:

DocSpot is designed with a scalable 3-tier architecture consisting of:

- **Presentation Layer (Frontend):** User-friendly interface for patients and healthcare providers to book and manage appointments.
- **Business Logic Layer (Backend):** Handles appointment scheduling, notifications, user management, and telehealth integration.
- **Data Storage Layer:** Secure storage of user profiles, appointment records, and healthcare provider details.

The platform integrates with third-party APIs for notifications (SMS/email) and telehealth services to enhance usability.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web and mobile-friendly interface for patients and providers	HTML, CSS, JavaScript / React Js etc.
2.	Application Logic-1	Appointment booking, calendar management, reminders	Node.js, Express.js
3.	Application Logic-2	Admin panel, provider management, reporting	React js, Node js
4.	Database	Stores user profiles, appointments, provider datas	MongoDB

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
5.	Open-Source Frameworks	Frontend frameworks	React.js, Node.js, BootStrap, Tailwind CSS
6.	Scalable Architecture	3-tier architecture with RESTful APIs	Microservices

References:

[React.js Documentation]

[Node js Best Practice]

[JSON Web Server Referance]

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d

Technical Architecture

The Book a Doctor App is built using a modern client-server architecture. The frontend uses Bootstrap and Material UI to create a clean, responsive design that works well on all devices. Axios is used to connect the frontend with the backend through smooth API calls.The backend runs on Express.js, which handles the main server functions and processes. MongoDB is used to store important data like user details, appointments, and doctor profiles in a flexible and scalable way.For security, the app uses JWT to manage user sessions safely and bcrypt to protect passwords by encrypting them.Moment.js helps manage dates and times accurately, making appointment scheduling reliable.The admin panel lets authorized users manage doctor sign-ups, control platform settings, and ensure everything runs smoothly. Access is controlled using Role-Based Access Control (RBAC) so that users only see what they're allowed to.The app is also designed to handle growth. MongoDB scales easily to manage more data, and performance is improved through load balancing and caching, which help handle high traffic and speed up the app.

Modern client-server architecture structured in a 3-tier model:

1. Presentation Layer (Frontend):

A responsive, user-friendly web and mobile interface built with Bootstrap, Material UI, and React.js, providing seamless interactions for patients and healthcare providers.

2. Business Logic Layer (Backend):

Built using Node.js and Express.js, this layer handles:

API communication is handled efficiently using Axios, ensuring smooth data exchange

- Appointment scheduling
- Calendar management
- Notification dispatch
- Telehealth integrations
- Admin operations and role-based permissions

3. Data Storage Layer:

Employs MongoDB, a scalable NoSQL database, to store:

The DocSpot platform ensures an intuitive and responsive user experience by using:

React.js, Bootstrap, and Material UI to build clean and adaptable user interfaces that work smoothly on both web and mobile devices.

Axios, a reliable HTTP client, to manage all communication between the frontend and backend services, ensuring that data flows efficiently and consistently.

- Patient and doctor profiles
- Appointment records
- Platform configurations
- Frontend Technologies

Frontend Technologies

The DocSpot platform ensures an intuitive and responsive user experience by using:

- **React.js, Bootstrap,** and **Material UI** to build clean and adaptable user interfaces that work smoothly on both web and mobile devices.
- **Axios**, a reliable HTTP client, to manage all communication between the frontend and backend services, ensuring that data flows efficiently and consistently.-

Backend Framework

The backend of DocSpot is developed using **Express.js**, a lightweight and powerful framework built on **Node.js**. This enables:

- Fast handling of HTTP requests and API responses
- Scalable server-side logic that's easy to manage and expand as the platform grows

Database and User Authentication

For data storage and user security, DocSpot leverages:

- **MongoDB,** a flexible NoSQL database that supports large-scale data and fast performance. It securely holds user information, appointment logs, and doctor details.
- **JWT (JSON Web Tokens**) for secure, token-based user authentication. This allows users to stay logged in without needing server-side sessions.
- **Bcrypt** for password encryption, ensuring that user credentials are stored safely and are not readable even if accessed maliciously.

Admin Dashboard and Access Management

The platform includes a dedicated admin panel that allows authorized staff to:

- Approve doctor registrations
- Monitor system activity

- Update platform settings

Access to features is controlled through Role-Based Access Control (RBAC), ensuring that each type of user (patient, doctor, admin) only sees and uses what they are authorized to.

Performance and Scalability

DocSpot is designed to handle growing user demands through:

- MongoDB’s horizontal scaling to support increased data loads
- Load balancing, which distributes incoming traffic evenly across multiple servers to avoid slowdowns
- Caching mechanisms, which store frequently accessed data temporarily to speed up responses and reduce database hits

Scheduling and Time Management

Appointment scheduling and time-based features are powered by Moment.js, a library that simplifies:

- Date/time formatting
- Time zone adjustments
- Accurate scheduling for both patients and doctors