**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 26-05-2025 |
| Team ID | LTVIP2025TMID55247 |
| Project Name | DocSpot |
| 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** |
| --- | --- | --- | --- |
|  | User Interface | Web and mobile-friendly interface for patients and providers | HTML, CSS, JavaScript / React Js etc. |
|  | Application Logic-1 | Appointment booking, calendar management, reminders | Node.js, Express.js |
|  | Application Logic-2 | |  | | --- | | Admin panel, provider management, reporting | | React js, Node js |
|  | Database | Stores user profiles, appointments, provider datas | MongoDB |

**Table-2: Application Characteristics:**

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

**References:**

**[React.js Documentation](https://react.dev/)**

**[Node js Best Practice](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)**

**[JSON Web Server Referance](https://www.npmjs.com/package/json-server)**

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