## **Technical Architecture**

## **Technical Architecture – DocSpot**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | LTVIP2025TMID55161 |
| Project Name | Docspot |
| Maximum Marks | 4 Marks |

DocSpot adopts a modular and scalable **3-tier architecture** that ensures separation of concerns, maintainability, and performance. Each layer works independently while communicating through secure and efficient APIs.

### **1. Presentation Layer (Frontend)**

* **Purpose:** Acts as the user interface for all roles—patients, doctors, and admins.
* **Technology Stack:** React.js, HTML, CSS, JavaScript, Bootstrap/Tailwind CSS
* **Responsibilities:**
  + Render responsive and dynamic UI
  + Provide role-based dashboards
  + Handle form inputs, document uploads, appointment booking
  + Communicate with backend via RESTful APIs using Axios

### **2. Business Logic Layer (Backend)**

* **Purpose:** Implements the application’s core logic and enforces workflows.
* **Technology Stack:** Node.js, Express.js, JWT, Multer
* **Responsibilities:**
  + User authentication and authorization (JWT)
  + Appointment management and scheduling validations
  + Doctor profile approval and role-based routing
  + Admin operations, document handling, and notifications
  + REST API endpoints to serve frontend requests

### **3. Data Storage Layer**

* **Purpose:** Stores and manages application data securely and efficiently.
* **Technology Stack:** MongoDB, Mongoose
* **Responsibilities:**
  + Manage collections for Users, Appointments, Documents, Notifications
  + Maintain relationships between patients, doctors, and appointments
  + Ensure data integrity, indexing for fast queries, and schema validation

### **Additional Integrations**

* **Email/SMS API (e.g., Nodemailer, Twilio):** For sending appointment confirmations and reminders.
* **Telehealth API (optional):** To support video consultations.
* **Authentication Middleware:** Secure access to protected routes.
* **File Handling:** Upload and retrieve prescriptions or medical records using Multer and MongoDB (or cloud storage in future phases).

**Components & Technologies – DocSpot**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology Used** |
| **1** | User Interface | Web-based responsive UI for patients, doctors, and admins | React.js, HTML, CSS, JavaScript, Bootstrap, Tailwind CSS |
| **2** | Routing & Navigation | Handles client-side page switching and protected routes | React Router DOM, Context API |
| **3** | API Communication | Facilitates frontend-backend interaction | Axios (HTTP client) |
| **4** | Authentication System | Secure login and role management | JSON Web Tokens (JWT), bcrypt.js |
| **5** | Appointment Logic | Booking, status management, and slot scheduling | Node.js, Express.js |
| **6** | Admin Panel | Doctor approval, user management, and appointment oversight | React.js (admin views), Express.js (backend logic) |
| **7** | Database Management | Stores users, doctors, appointments, and reviews | MongoDB, Mongoose |
| **8** | Document Uploads | Uploading and accessing prescriptions or reports | Multer (middleware), MongoDB (GridFS or binary storage) |
| **9** | Notifications System | Alerts for appointment status or changes | Email (Nodemailer), optional SMS (Twilio) |
| **10** | Deployment & Hosting | Making the app available online | Vercel (frontend), Render or Cyclic (backend) |
| **11** | Version Control & CI/CD | Source code management and deployment pipeline (optional) | Git, GitHub, GitHub Actions |

**DocSpot – Application & Characteristics**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Application Module** | **Key Characteristics** |
| **1** | User Interface (UI) | Responsive, role-based design using React.js; mobile-friendly and intuitive |
| **2** | User Authentication | JWT-based login system with role verification for patients, doctors, and admins |
| **3** | Appointment Booking | Real-time slot availability, status tracking (pending/approved/completed) |
| **4** | Doctor Profile Management | Editable profiles with specialization, availability, and experience info |
| **5** | Admin Panel | Doctor approval, user management, and platform analytics |
| **6** | Notification System | SMS/Email alerts for appointment confirmations, reminders, and status updates |
| **7** | Document Upload | Patients can upload prescriptions/reports; doctors can access pre-appointment documents |
| **8** | Teleconsultation Module | Optional video consultations via integrated APIs (future enhancement) |
| **9** | Review & Rating System | Patients can leave feedback after each appointment |
| **10** | Data Storage & Security | MongoDB with schema validation, encrypted user data, and role-based access control |