**Requirement Analysis**

**Technology Tack**

|  |  |
| --- | --- |
| Date | 27-06-2025 |
| Team ID | LTVIP2025TMID46688 |
| Project Name | DocSpot |
| Maximum Marks | 4 Marks |

**Technology Track – DocSpot**

The **Technology Track** outlines the core technologies, tools, and frameworks used in the development of the DocSpot platform. This full-stack web application was built using the **MERN stack** (MongoDB, Express.js, React.js, and Node.js), along with supporting libraries and services to enhance performance, usability, and security.

**1. Frontend Technologies**

The frontend of DocSpot is designed to be clean, responsive, and user-friendly, providing smooth navigation and interactions for users, doctors, and administrators.

* **React.js**:  
  Used to build component-based UI with dynamic rendering and routing.
* **HTML5 & CSS3**:  
  Standard markup and styling for structuring pages and making them visually appealing.
* **Bootstrap**:  
  Ensures responsive design and consistent UI components across all device sizes.
* **React Router**:  
  Enables seamless navigation between different components and routes.
* **Axios**:  
  Handles asynchronous HTTP requests between frontend and backend.

**2. Backend Technologies**

The backend is responsible for handling logic, authentication, APIs, and communication with the database.

* **Node.js**:  
  JavaScript runtime environment for building fast and scalable backend services.
* **Express.js**:  
  Lightweight web application framework that simplifies routing, middleware handling, and API creation.
* **JWT (JSON Web Token)**:  
  Used for securing authentication and maintaining user sessions.
* **Bcrypt.js**:  
  Hashes user passwords to ensure secure storage in the database.

**3. Database**

* **MongoDB (NoSQL)**:  
  A flexible, scalable, and document-oriented database used to store user data, appointment records, doctor profiles, etc.
* **Mongoose**:  
  ODM (Object Data Modeling) library for MongoDB and Node.js, providing schema and validation support.

**4. Tools and Platforms**

* **Postman / Thunder Client**:  
  Used for testing APIs during development.
* **Visual Studio Code (VS Code)**:  
  Main IDE used for coding with extensions for React and Node.
* **Git & GitHub**:  
  Version control and collaborative development platform for maintaining project source code.
* **Deployment Platforms (Optional)**:
  + **Render / Vercel / Netlify** (if deployed): Used for deploying frontend and backend servers.

**5. Development Methodology**

* The project follows an **Agile-like approach**, allowing iterative development and testing.
* Development was divided into phases: UI Design → Authentication → Dashboards → Features → Testing.

**Outcome**

Using a modern and scalable technology stack helped in building:

* A responsive frontend
* A secure and efficient backend
* Seamless integration between all components
* A robust, maintainable, and scalable full-stack web application