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

Date	31 January 3035
Team ID	LTVIP2026TMIDS81521
Project Name	SmartDoc Appointment System
Maximum Marks	4 Marks

Technical Architecture:

Architecture Type:

3-Tier Web Architecture (Presentation Layer – Application Layer – Database Layer)

Infrastructure Demarcation:

Layer	**Deployment**
Frontend (React)	**Localhost / Vercel (Cloud)**
Backend (Node + Express)	**Localhost / Render (Cloud)**
Database (MongoDB Atlas)	**Cloud Database**
Authentication	**JWT + Bcrypt**
Email Service	**Nodemailer / SMTP**

◈ Architecture Explanation

1 Presentation Layer (Frontend)

- **Built using React.js**
- **Handles UI rendering**

- **Communicates with backend via REST APIs**
- **Responsive for mobile and desktop**

2 Application Layer (Backend)

- **Built using Node.js & Express.js**
- **Handles:**
  - **Authentication**
  - **Appointment booking logic**
  - **Doctor approval logic**
  - **Admin controls**
- **Uses middleware for security**

3 Data Layer

- **MongoDB Atlas (Cloud NoSQL database)**
- **Stores:**
  - **Users**
  - **Doctors**
  - **Appointments**
  - **Admin Data**

Table – 1: Components & Technologies

S.No	Component	Description	Technology

1	User Interface	Web-based UI for patients, doctors, and admin	React.js, HTML, CSS, JavaScript
2	Application Logic – Authentication	User registration, login, JWT validation	Node.js, Express.js, JWT, Bcrypt
3	Application Logic – Appointment System	Booking, cancelling, rescheduling appointments	Node.js, Express.js
4	Application Logic – Doctor Management	Doctor approval, availability management	Node.js
5	Database	Stores user, doctor & appointment data	MongoDB (NoSQL)
6	Cloud Database	Hosted cloud database	MongoDB Atlas
7	File Storage	Stores profile images	Cloudinary / Local Storage
8	External API – Email Service	Sends appointment confirmations	Nodemailer / SMTP
9	External API – Payment Gateway (Optional)	Online payment for booking (if implemented)	Razorpay / Stripe
10	Machine Learning Model	Not Applicable (No ML used)	
11	Infrastructure (Server / Cloud)	Deployment environment	Localhost (Development), Render/Vercel (Production)

Table – 2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for frontend and backend development	React.js, Express.js, Node.js
2	Security Implementations	Password encryption, token authentication, role-based access	JWT, Bcrypt, HTTPS
3	Scalable Architecture	Modular folder structure & REST API-based architecture	MERN Stack

4	Availability	Cloud-hosted backend and database ensure 24/7 availability	MongoDB Atlas, Render
5	Performance	Fast API response, optimized database queries	Express.js, MongoDB Indexing
6	Reliability	Data validation & error handling middleware	Express Middleware
7	Compatibility	Cross-browser & mobile-friendly UI	React Responsive Design