Project Design Phase-II

Technology Stack for "Book A Doc"

Field	Details
Date	10 April 2025
Team ID	SWTID1743701170
Project Name	Book A Doc

Technical Architecture Overview

Aspect	Details
Purpose	To define the technical architecture and technology stack for "Book A Doc," ensuring scalability, security, and usability for appointment booking.
Scope	Includes frontend, backend, database, payment integration, and deployment infrastructure, with a focus on MERN stack and cloud services.

Architectural Diagram

Section	Description
Diagram	*[To be inserted: A 3-tier architectural diagram showing Presentation Layer (React), Application Layer (Node.js/Express), and Data Layer (MongoDB/AWS S3). Include data flows for booking and payment via Stripe.]*
Figure Caption	Figure 1: Technical Architecture Diagram for Book A Doc

Section	Description
Reference	Adapted from standard MERN architecture principles.

***Note*: Since I cannot generate images, create a diagram using a tool like Draw.io, Lucidchart, or PowerPoint. Include:**

- **Presentation Layer: React UI for Patients, Doctors, Admins.**
- **Application Layer: Node.js/Express with APIs.**
- **Data Layer: MongoDB for data, AWS S3 for files, Stripe for payments.**
- **Arrows showing flows: User → React → Express → MongoDB → Stripe.**
- **Save as Tech_Stack_Diagram.png and insert below this table.**

Table-1: Components & Technologies

Component	Technology	Purpose
Frontend	React v18	Responsive UI for dashboards and booking forms.
Backend	Node.js v18, Express.js	Handle API requests, authentication, and business logic.
Database	MongoDB v6	Store user data, appointments, earnings, and profiles.
Payment Gateway	Stripe API	Secure online payment processing for appointments.
Deployment	AWS EC2, Docker	Host application with containerized deployment.
File Storage	AWS S3	Store profile images or documents.
Authentication	JSON Web Tokens (JWT)	Secure user authentication and role-based access control.

Component	Technology	Purpose
Testing	Jest, Postman	Unit and API testing for reliability.

Table-2: Application Characteristics

Characteristic	Description
Scalability	Supports up to 10,000 users with AWS auto-scaling and MongoDB sharding.
Performance	API response time < 2s for 100 concurrent users; dashboard load time < 3s.
Security	HTTPS enforced, AES-256 encryption for JWT, role-based access (Patient, Doctor, Admin).
Usability	Responsive UI across devices (desktop, tablet, mobile) with intuitive navigation.
Reliability	99.9% uptime with AWS failover and MongoDB replication.
Maintainability	Modular MERN codebase with clear documentation and Git version control.

References

Source	Purpose
https://www.mongodb.com/mern-stack	MERN stack overview
https://stripe.com/docs/api	Payment integration guide
https://aws.amazon.com/	Cloud deployment reference