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

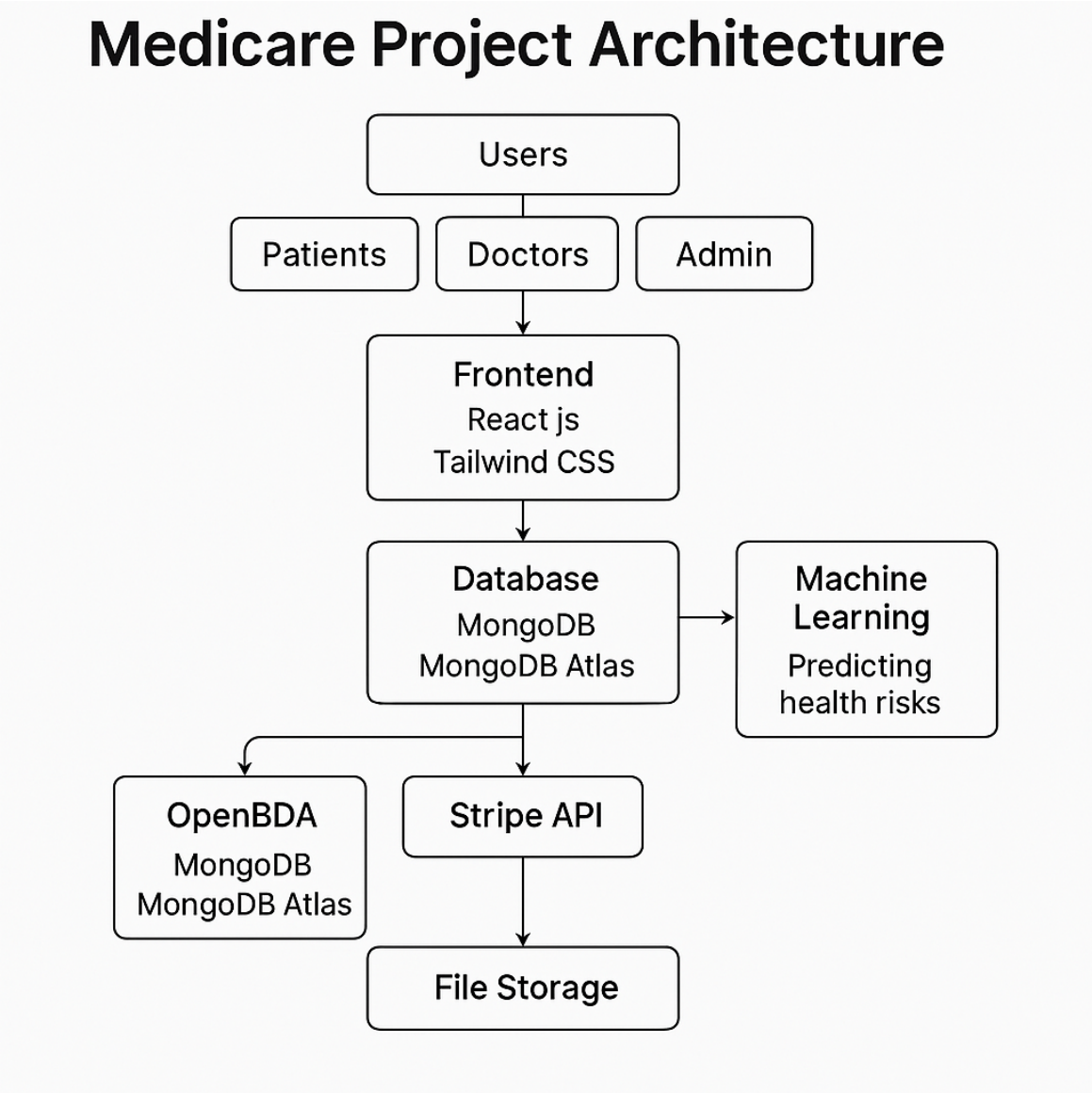
Date	31 January 3035
Team ID	SWTID1743517106
Project Name	Book a Doctor
Maximum Marks	4 Marks

Technical Architecture:

The system follows a modular architecture comprising frontend, backend, and database components.

- **Frontend:** Developed using React.js, Tailwind CSS to provide an intuitive user interface.
- **Backend:** Implemented with Node.js and Express.js to handle application logic and API endpoints.
- **Database:** Utilizes MongoDB for storing and managing healthcare data.
- **Infrastructure:** Deployed on cloud platforms such as IBM Cloud or AWS to ensure scalability and reliability.

Example: Architectural Diagram



S.No.	Component	Description	Technology
1.	User Interface	Web-based interface for users	React.js, Tailwind CSS
2.	Application Logic-1	Handles core functionalities like appointments	Node.js, Express.js
3.	Application Logic-2	Manages user authentication and authorization	Passport.js, JWT
4.	Application Logic-3	Processes billing and payment operations	Stripe API
5.	Database	Stores patient records and other data	MongoDB

6.	Cloud Database	Cloud-hosted database service	MongoDB Atlas
7.	File Storage	Stores medical reports and documents	AWS S3 or IBM Cloud Object Storage
8.	External API-1	Retrieves drug information	OpenFDA API
9.	External API-2	Verifies patient identity	Aadhar API
10.	Machine Learning Model	Predicts patient health risks	TensorFlow, Scikit-learn
11.	Infrastructure (Server / Cloud)	Hosts the application	IBM Cloud, AWS EC2

Table-1: Components & Technologies

S.No.	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source libraries and frameworks	Node.js, Express.js, React
2.	Security Implementations	Implements robust security measures including encryption and IAM controls	SHA-256, HTTPS, JWT, OAuth 2.0
3.	Scalable Architecture	Designed with microservices architecture for scalability	Docker, Kubernetes
4.	Availability	Ensures high availability through load balancing and redundancy	Nginx, AWS ELB, IBM Cloud Load Balancer
5.	Performance	Optimized for performance with caching and CDN integration	Redis, CloudFront, IBM CDN

Table-2: Application Characteristics

References:

https://c4model.com/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

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