Project Report: Doctor Booking Website

Team Lead : S. Vishwa
Project Contributors: Dinesh, Gowtham, Senthilkumar

Introduction :

In today's digital era, the healthcare industry is increasingly moving toward online platforms that simplify and enhance patient-doctor interactions. Our project, a doctor booking website, is designed to offer patients an easy, efficient way to schedule appointments, browse doctor profiles, and leave feedback. This report outlines the development of this user-friendly platform, which utilizes the MERN stack—MongoDB, Express, React, and Node.js. The website is structured for optimal performance, flexibility, and an intuitive user experience, all grounded in modern web development best practices.

Technologies Used :

Our team selected the *MERN stack* for its robust capabilities in full-stack development, seamless front-to-back integration, and scalability. Each component plays a crucial role in making the system efficient and maintainable:

MongoDB : NoSQL database that provides a flexible schema, ideal for managing varied doctor and patient information.
Express : Backend framework that simplifies server-side logic and API development, providing RESTful services for data interaction.
React : Frontend library known for its component-based architecture and responsive user interface, perfect for creating dynamic and interactive web pages.
Node.js : Server environment that allows JavaScript to be used for backend development, facilitating faster development with a single language across the stack.

Development Overview :

User-Focused Design :

Our doctor booking website places user experience at the forefront, with simple and accessible language, intuitive navigation, and a clean layout. Patients can search for doctors by specialty, location, and availability, while doctors have the tools they need to manage appointment schedules effectively.

Project Structure :

We meticulously organized the project files to separate concerns, ensuring that frontend and backend code is clearly divided and each component is easy to locate and update. This project structure supports agile development practices and allows for future scalability.

The project includes:

Frontend (React): Focuses on user interactions, including search functionality, booking forms, and feedback mechanisms.
Backend (Express/Node): Manages server-side logic, routes, and API calls, ensuring secure and efficient data handling.
Database (MongoDB): Stores structured information about users, doctors, appointments, and reviews, supporting flexible data queries.
Environment Configuration: Sensitive data, such as database connection strings, is stored securely using environment variables to maintain security standards.

Prerequisites :

To facilitate smooth development, we ensured that the project environment was thoroughly prepared. Essential tools include **Node.js** and **npm** for package management, **MongoDB** for the database setup, and **Postman** for API testing. Code versioning was handled with **Git** to allow team collaboration and maintain a seamless workflow.

ER Diagram and Project Flow :

A well-defined ER (Entity-Relationship) diagram was created to visualize and structure database relationships, providing clarity on how users (patients), doctors, appointments, and reviews interact. This ER diagram served as a guide for implementing efficient database schemas and RESTful API endpoints. Additionally, a clear flowchart was developed to represent the user journey from login to appointment booking and feedback submission.

Project Contributions :

Under the guidance of **Team Lead** S. Vishwa , each team member made significant contributions to the project:

Dinesh: Spearheaded the frontend development, ensuring that the interface is both responsive and accessible.

Gowtham: Focused on backend development, optimizing API endpoints and data handling efficiency.

Senthilkumar : Assisted in database schema design, ER diagram development, and database queries for enhanced performance.

Conclusion :

Our doctor booking website is an accessible, efficient platform aimed at improving patient-doctor interactions in the healthcare space. By implementing modern web technologies and focusing on user-friendly design, our team created a streamlined, scalable solution ready to serve users effectively. This project stands as a testament to our team's collaborative efforts and dedication to quality development. We are excited about the future potential of this platform and the enhancements it will continue to receive as healthcare technology evolves.

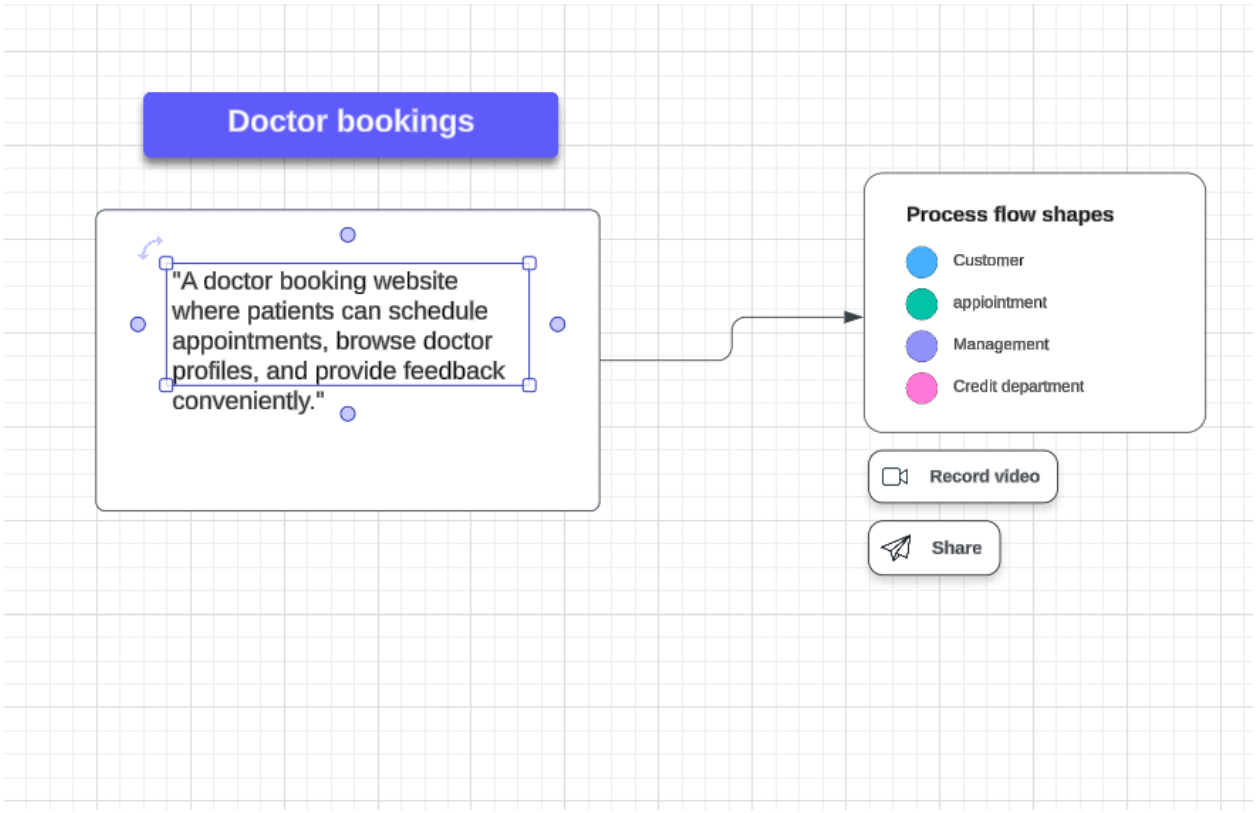
Prerequisites :

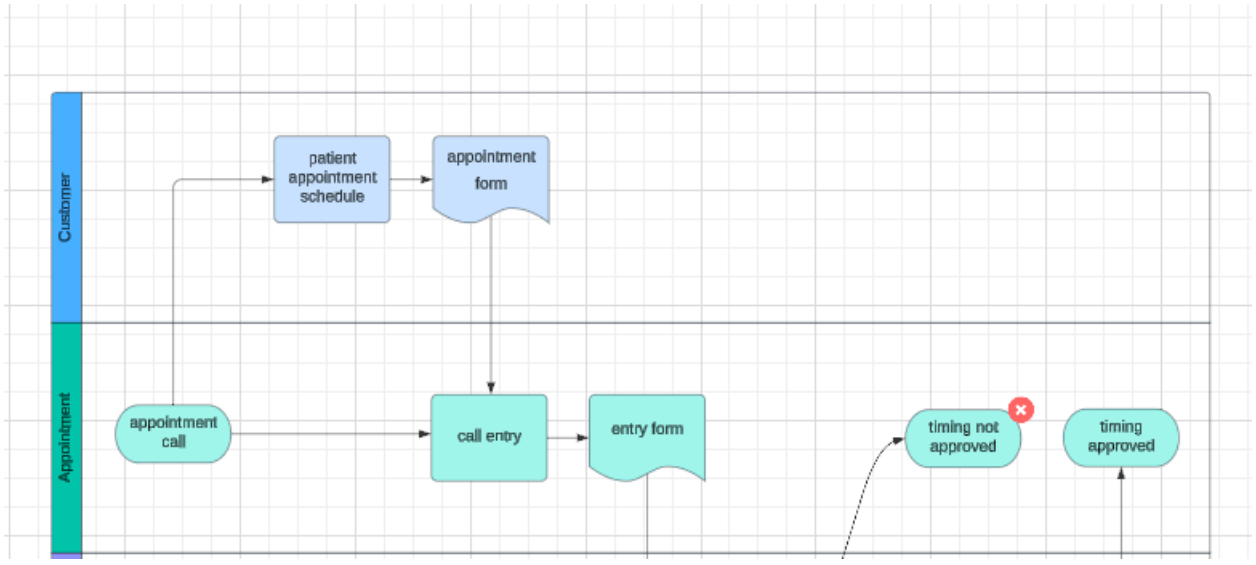
plaintext

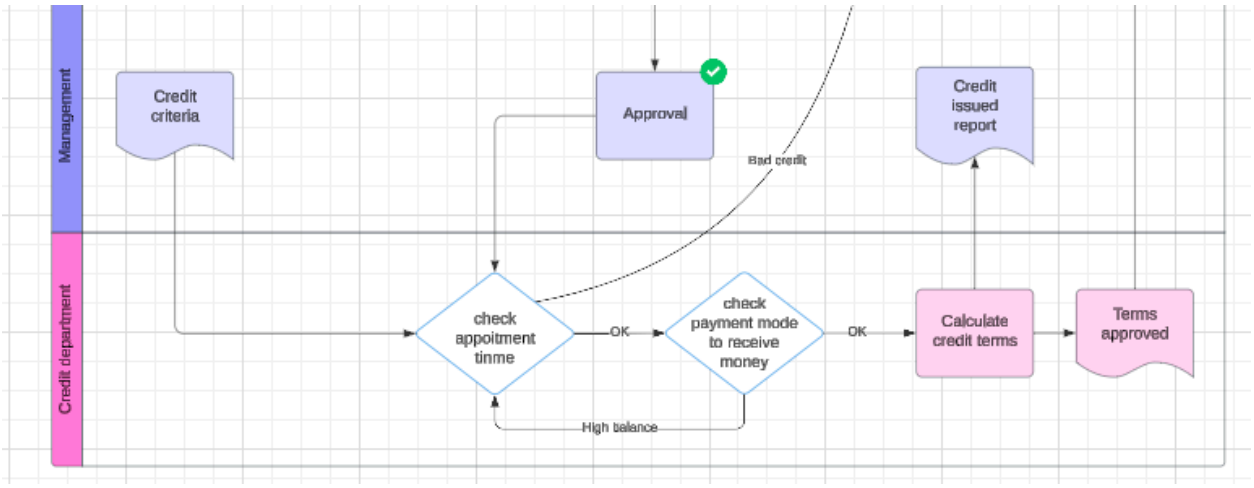
```
doctor-booking-website/
│
├── client/                    # React frontend
│   ├── public/                # Public assets
│   └── src/
│       ├── components/        # Reusable React components
│       ├── pages/             # Main pages (Home, DoctorProfile, Booking)
│       ├── services/          # API calls and client services
│       ├── App.js             # Main React App component
│       └── index.js           # Entry point for React
│
├── server/                    # Backend server
│   ├── config/                # Database and environment configuration
│   ├── controllers/           # Business logic for each route
│   ├── models/                # Mongoose schemas for Doctor, Appointment, etc.
│   ├── routes/                # Routes for User, Doctor, Appointment
│   ├── middlewares/           # Authentication, error handling
│   ├── server.js              # Main server file
│
├── .env                       # Environment variables (MongoDB URI, API keys)
├── .gitignore                 # Files and folders to ignore in Git
├── package.json               # npm scr  s and dependencies
└── README.md                  # Project documentation
```

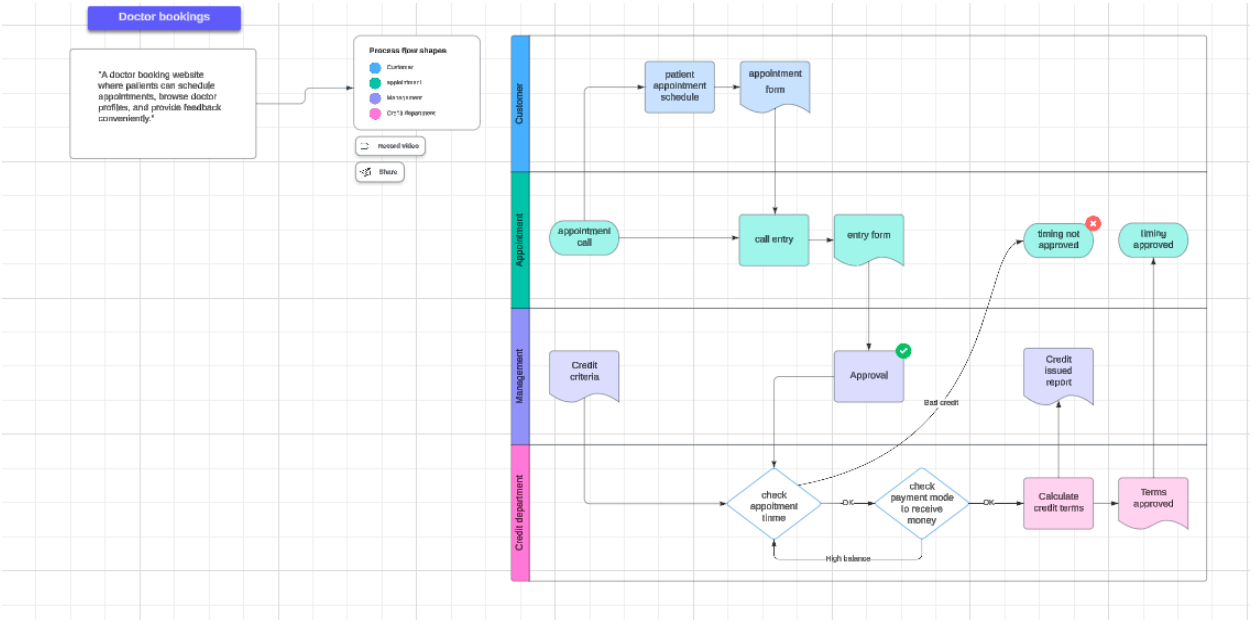
ER diagram Structural format :

https://lucid.app/lucidchart/ec050c8c-06f4-4088-9672-e246166fd808/edit?view_items=xmcVClwwzFBQ&invitationId=inv_dbb3ecd8-d37c-43bf-ab53-218e69a9a7f6


Doctor bookings
"A doctor booking website where patients can schedule appointments, browse doctor profiles, and provide feedback conveniently."
Process flow shapes
Customer
appiointment
Management
Credit department
Record video
Share


Customer
Appointment
patient appointment schedule
appointment form
appointment call
call entry
entry form
timing not approved
timing approved


Management
Credit department
Credit criteria
Approval
Credit issued report
Bad credit
check appoitment tinme
OK
check payment mode to receive money
OK
Calculate credit terms
Terms approved
High balance


Doctor bookings
"A doctor booking website where patients can schedule appointments, browse doctor profiles, and provide feedback conveniently."
Process flow shapes
Customer
Appointment
Management
Credit department
patient appointment schedule
appointment form
appointment call
call entry
entry form
timing not approved
timing approved
Credit criteria
Approval
Credit issued report
Bad credit
check appoitment tinme
OK
check payment mode to receive money
OK
Calculate credit terms
Terms approved
High balance