A Mern Stack Project.

Book a Doctor using mern stack.

TEAM MEMBER

VIGNESH M - 110521104053. (TL)

VAISHALI R - 110521104052.

SWETHA R - 110521104050.

TAMILARASAN P - 110521104051.

Introducing the Solution:

1 Convenient Online Platform:

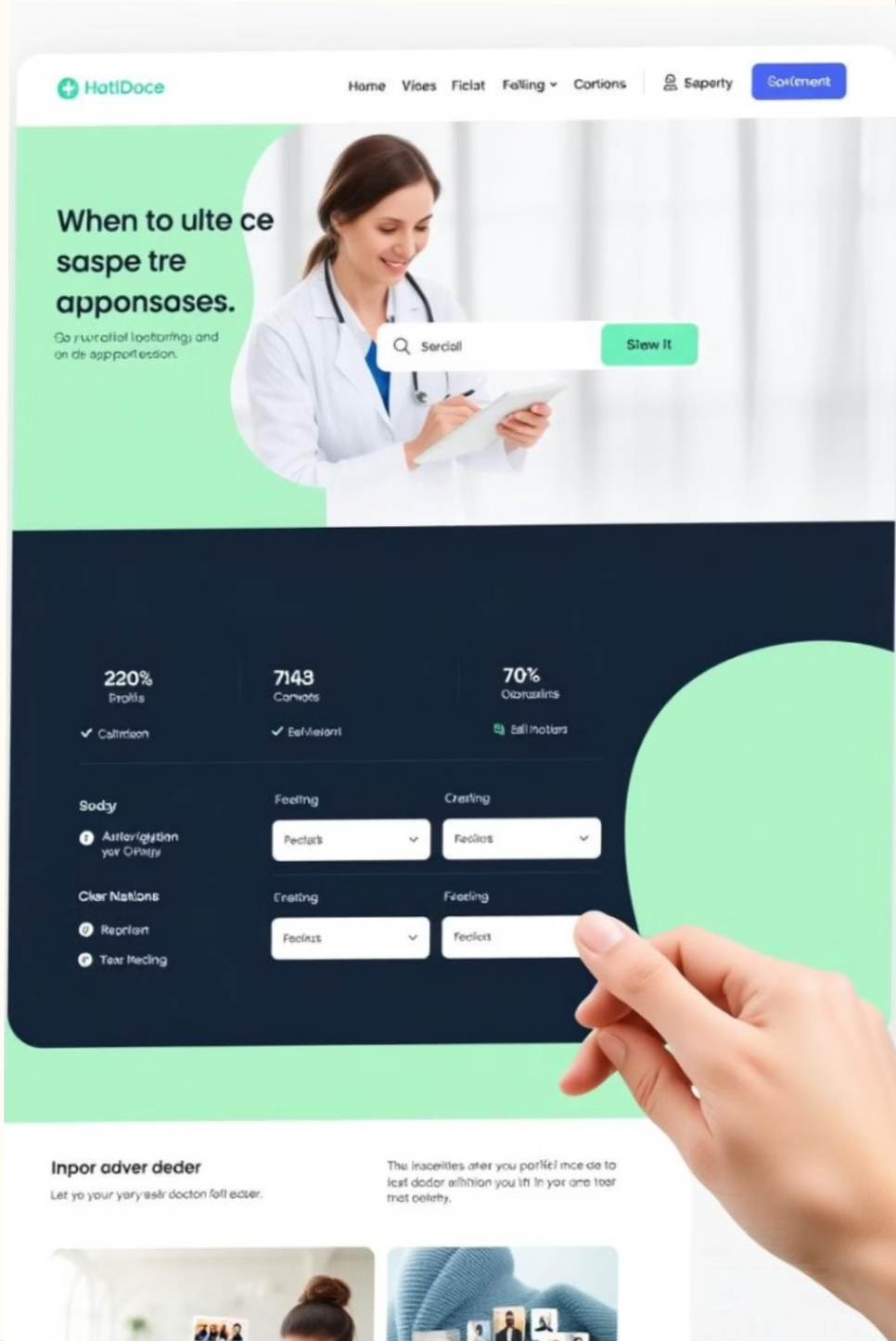
Patients can easily browse doctors, view availability, and book appointments online, from the comfort of their homes.

2 Efficient Appointment Management:

The system automates scheduling, reduces errors, and improves overall efficiency for both patients and doctors.

3 Enhanced Communication:

Real-time updates and notifications keep users informed of appointment status and any changes.



Project Overview: Book a Doctor:

Purpose:

To streamline the process of scheduling medical appointments, making it more convenient and accessible for patients.

Technology:

Utilizes the MERN stack, encompassing MongoDB for data storage, Express.js for backend logic, React for front-end development, and Node.js for server-side runtime.

Key Features:

1 **User Registration and Login:**

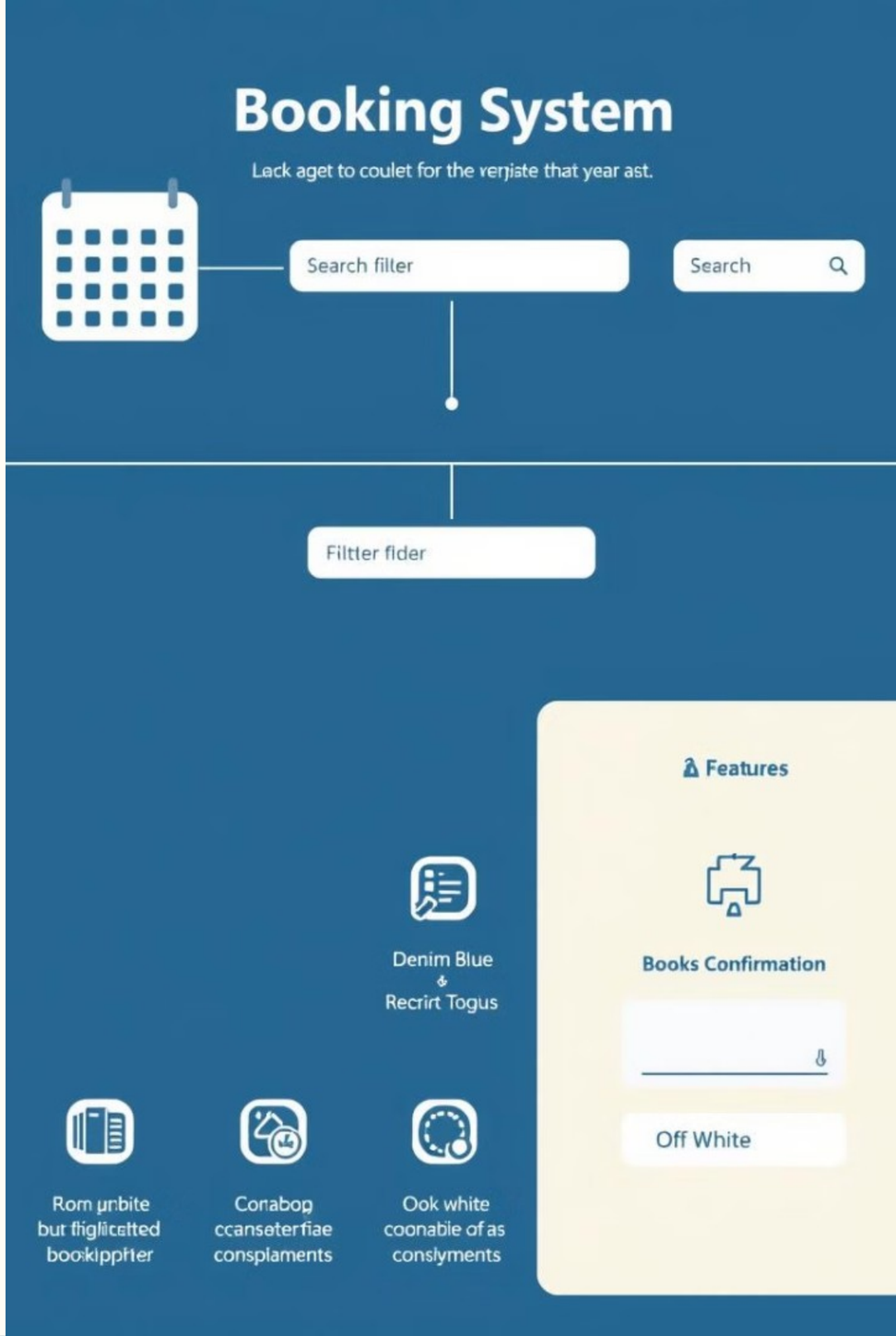
Secure account creation using email, phone number, or social media, with role-based access for users, administrators, and doctors.

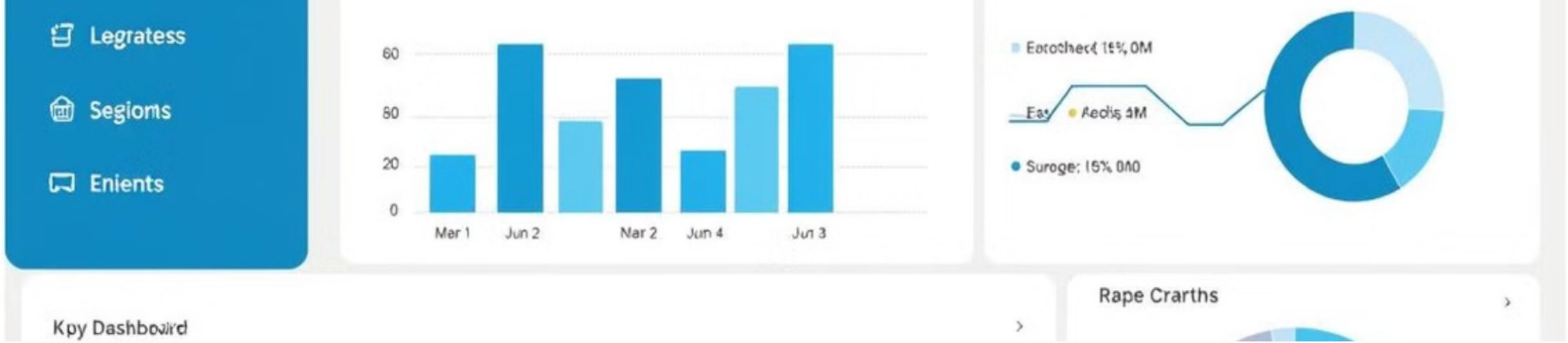
2 **Booking an Appointment:**

A simple form-based interface for filing an appointment request, with options to select appointment categories and attach supporting documents.

3 **Appointment Confirmation:**

Real-time updates on appointment status with notifications via email or SMS for status changes.





Admin Dashboard Features:

Appointment Management:

Tools for managing appointments, prioritizing cases, and assigning tasks to relevant departments.

Analytics and Reporting:

Comprehensive analytics and reporting to identify trends and optimize resolutions.

Feedback Mechanism:

Users can rate their satisfaction and provide suggestions post-resolution.

Workflow: Seamless Appointment Experience:

1

User Registration/Login

Users register by providing their details or log in with existing credentials, with role-based access ensuring users and administrators see relevant dashboards.

2

Appointment Submission

Users fill out a structured form specifying the type of appointment, description, and supporting documents, with validation and categorization for efficient routing.

3

Appointment Processing

The appointment is automatically assigned to the relevant department or team based on its category, with assigned personnel reviewing, investigating, and updating the status of the appointment.

4

Resolution and Updates

Users are notified of progress updates, including when the issue is resolved, with the appointment status changing to "Resolved" upon completion.

5

Feedback and Closure

Users can rate their satisfaction and provide feedback, which is logged for performance evaluation and system improvement.

Technical Architecture: Client-Server Model:

Frontend:

The frontend utilizes React.js for building interactive web pages and Bootstrap for creating mobile-friendly and visually appealing layouts.

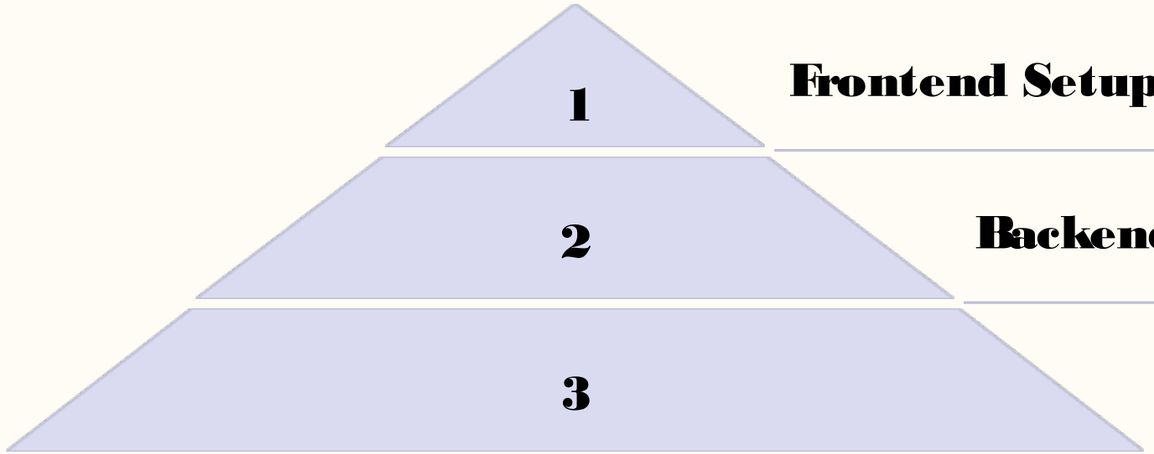
Backend:

The backend employs Node.js with Express.js to handle server-side logic and communication, using RESTful APIs to interact with the frontend.

Database:

MongoDB is used as the database to store user details, doctor information, and appointment data.

Project Implementation:



Frontend Setup (React.js)

Backend Setup (Node.js and Express.js)

Database (MongoDB)

Setting Up the Project:

1

Prerequisites

Ensure Node.js, npm, MongoDB, Git, and a text editor (like Visual Studio Code) are installed.

2

Clone the Repository

Clone the project repository from GitHub or other Git platforms using the command: git clone [repository URL]

3

Install Dependencies

Install necessary packages for both frontend and backend using npm install in respective directories.

4

Start the Server

Start the backend server using npm start, which will launch the server on a specified port (usually http://localhost:5000).

5

Run the Frontend

Run the frontend development server using npm start in the frontend directory, typically launching on http://localhost:3000.

User Interface Design

Home Page

Provides a welcoming landing page, showcasing features and encouraging users to book an appointment or register.

Registration Page

Allows new users to create accounts, providing fields for name, email, password, and other relevant information.

Login Page

Enables existing users to authenticate, requiring their email and password for secure access to the platform.

Sample Output: Book Your Doctor:

1

Doctors Listing

A user-friendly interface to browse and filter doctors based on specialization, availability, and ratings, with doctor profiles featuring photos, names, and qualifications.

2

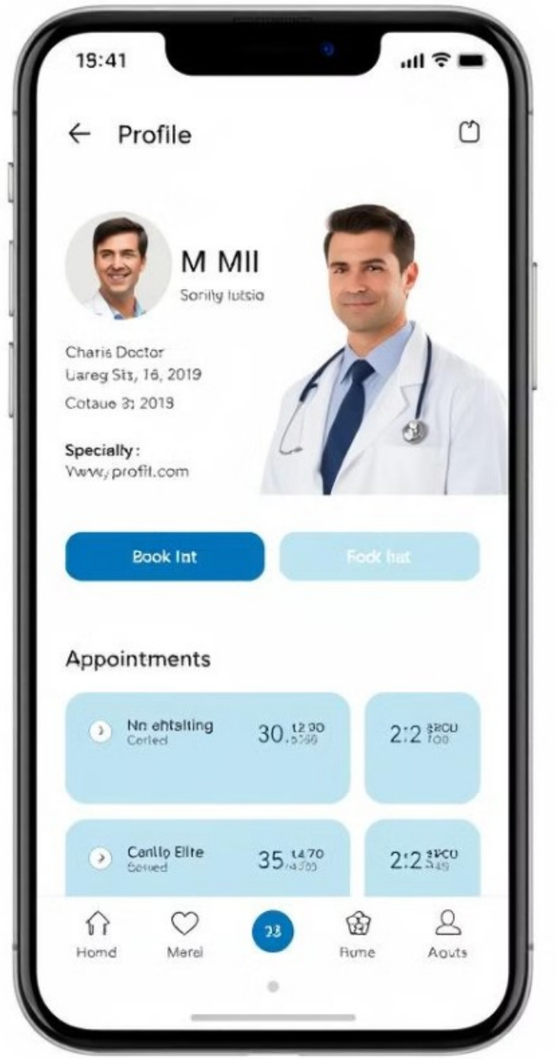
Appointment Booking

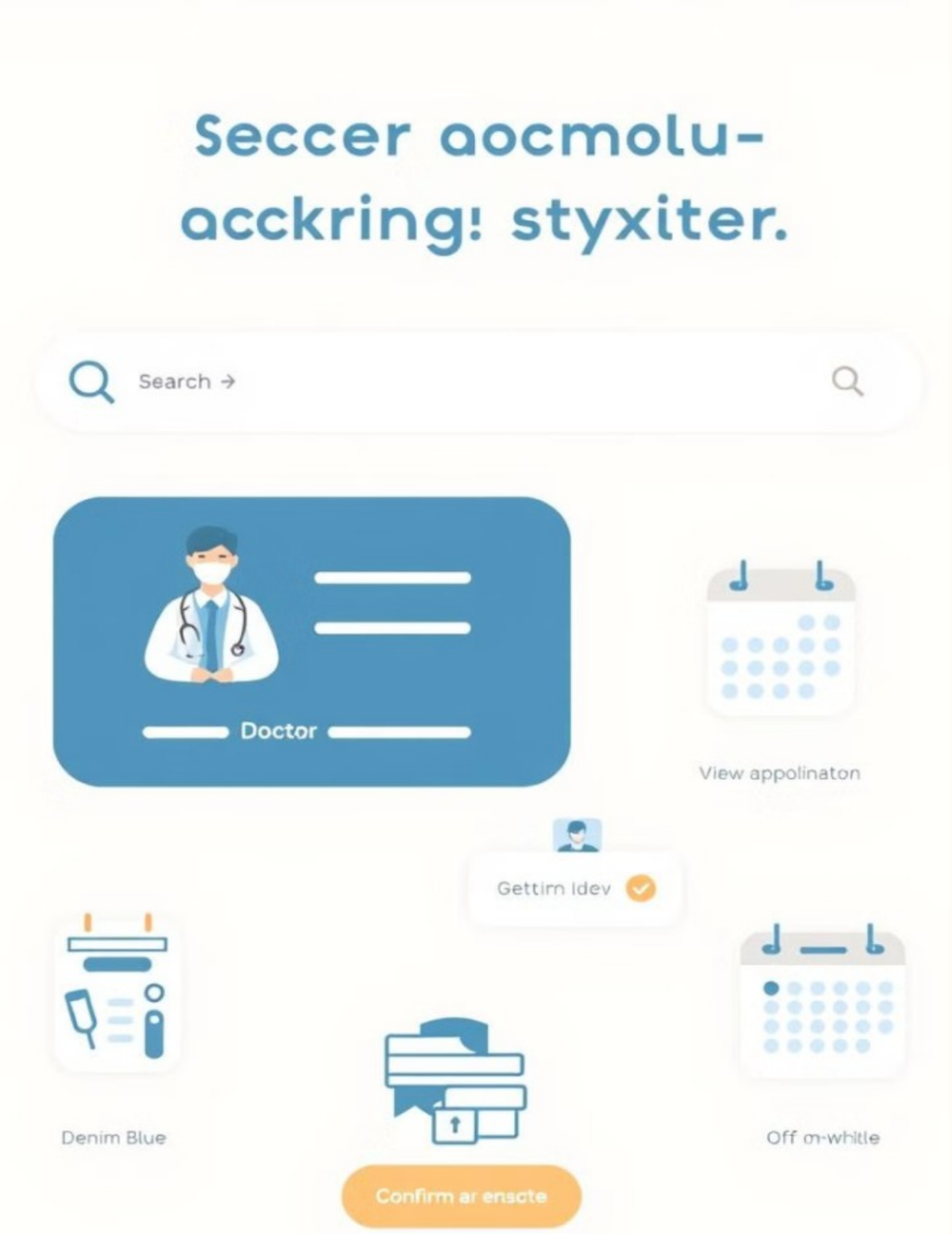
A simple process to book appointments with a selected doctor, allowing users to choose dates and available time slots.

3

Appointment Confirmation

Real-time confirmation of appointments with a detailed summary of the booked appointment, including doctor's information, date, and time.





Conclusion:

The Doctor Booking System, developed using the MERN stack, presents a robust and scalable solution for streamlining medical appointments. With its user-friendly interface, efficient workflow, and comprehensive features, this system offers a modern approach to managing healthcare appointments.

Further development can incorporate AI-driven features like appointment scheduling recommendations and automated reminders to enhance the user experience. This project demonstrates the power of the MERN stack in building innovative and effective solutions within the healthcare domain.

THANK YOU!