

PROJECT REPORT

DocSpot : Seamless Appointment Booking for Health

Date : 27-06-2025

Project Name : Docspot

Team ID : LTVIP2025TMID20459

Team members:

Team Leader : Modugu Vidya

Team member : Meer Apsarunnisa

Team member : Medepalli Sruthi

Team member : Merugumala Sobha Vinay Babu

Introduction

The healthcare industry is undergoing a digital transformation, with increasing emphasis on **enhancing accessibility, convenience, and efficiency** through technology. This shift has amplified the need for robust online systems that simplify healthcare services such as finding doctors, booking appointments, and managing patient records.

The **Book a Doctor App** is designed to meet this demand. It provides a **comprehensive, user-friendly web platform** that seamlessly connects patients with healthcare providers. By streamlining the appointment booking process and improving communication among patients, doctors, and administrators, the application reduces the friction typically associated with managing medical visits.

This project is built using the **MERN stack**—**MongoDB, Express.js, React.js**, and **Node.js**—demonstrating a practical and scalable approach to full-stack web development in the healthcare domain.

Project Overview

The **Book a Doctor App** is a comprehensive web-based solution that empowers users to **search, schedule, and manage medical appointments** online with ease. The platform is structured around three core user roles, each with dedicated functionality:

- **Patients** can register, log in, browse doctors by specialization or location, book appointments, upload relevant documents (e.g., prescriptions), and track their upcoming and past consultations.
- **Doctors** can apply to join the platform, manage their profiles and availability, respond to appointment requests, and update appointment statuses and medical records.
- **Administrators** oversee the platform by verifying doctor registrations, managing users, monitoring platform activity, and resolving any disputes or issues.

Key Features:

- Real-time booking with live doctor availability
- Secure authentication using **JWT (JSON Web Tokens)**
- **Role-based access control** to ensure appropriate permissions
- Custom dashboards tailored for patients, doctors, and administrators
- Notifications for appointment confirmations and status updates
- **Responsive React.js design** that delivers a smooth experience across devices

The application's architecture follows a modular and component-based structure, aligning functionality with user roles. The frontend, built with React.js, offers a dynamic single-page experience. On the backend, Node.js and Express.js handle server-side operations, while MongoDB serves as the flexible, document-oriented database.

Purpose

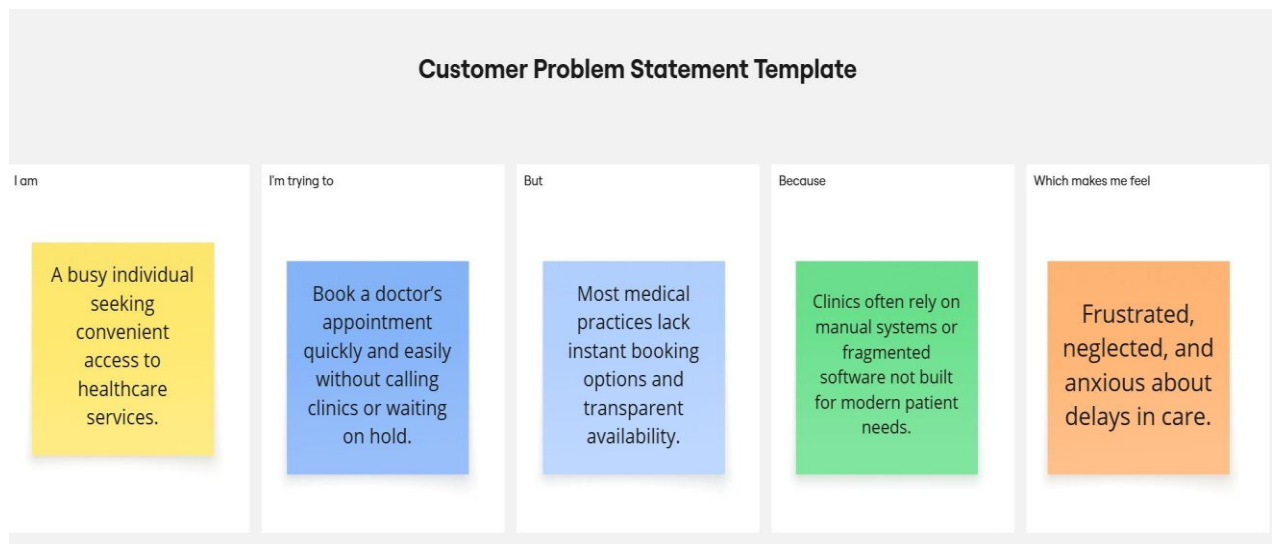
The main **purpose** of this project is to **simplify and digitalize the process of booking medical appointments**, especially in a world where timely healthcare access is critical. Traditionally, patients need to call hospitals or clinics, wait in queues, or face scheduling conflicts. This application removes those obstacles by offering an efficient and secure alternative.

Additional goals include:

- ❑ **Improving patient-doctor communication:** The app provides tools for both parties to stay informed about appointment status, updates, and follow-ups.

- ❑ **Showcasing modern web development skills:** It demonstrates how the MERN stack can be used to build a fully functional, scalable, and maintainable full-stack application.
- ❑ **Enhancing healthcare management:** By storing and managing user data, documents, and booking history, the system supports better organization for both patients and providers.
- ❑ **Creating a secure and user-friendly experience:** Security features like JWT-based login and role-based access ensure data privacy, while the responsive UI improves accessibility.

Ideation Phase



Problem Statement:

1. Patients face difficulty booking timely and convenient medical appointments.
2. Existing systems often lack real-time availability, streamlined communication, and a user-friendly experience.
3. There's a disconnect between patients and verified healthcare professionals, leading to inefficiencies and frustration.

Proposed Solution:

1. Develop a **MERN stack web application** as a centralized healthcare appointment system.
2. Provide **patients** with an intuitive interface to:
 - Register and log in securely.
 - Search for doctors based on specialty, location, and availability.

- Book, reschedule, or cancel appointments in real time.
- 3. Enable **doctors** to:
 - Manage their profiles and availability.
 - View and update their appointment calendar.
 - Communicate with patients if necessary.
- 4. Empower **administrators** to:
 - Oversee all user activity.
 - Manage platform content (specialties, locations, user verifications).
 - Ensure data integrity and system security.
- 5. Integrate **notification and reminder systems** (email/SMS) to reduce no-shows and enhance communication.
- 6. Ensure **responsive design** for accessibility across devices (desktop, tablet, mobile).
- 7. Implement **role-based authentication** and **data protection measures** to safeguard user information.

Empathy Map Canvas

This empathy map is based on the primary user: **the patient**.

Empathy Map Canvas		Designed for:	Designed by:	Date:	Version:
<p>1 WHO are we empathizing with? Who is the person we want to understand? What is the situation they are in? What is their role in the situation?</p>	<p>GOAL</p>		<p>2 What do they need to DO? What do they need to do differently? What job(s) do they want or need to get done? What decision(s) do they need to make? How will we know they were successful?</p>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>7 What do they THINK and FEEL?</p> <p>PAINS What are their fears, frustrations, and anxieties?</p> <p>GAINS What are their wants, needs, hopes and dreams?</p> </div> <div style="width: 45%;"> <p>3 What do they SEE? What do they see in the marketplace? What do they see in their immediate environment? What do they see others saying and doing? What are they watching and reading?</p> </div> </div>					
<p>6 What do they HEAR? What are they hearing others say? What are they hearing from friends? What are they hearing from colleagues? What are they hearing second-hand?</p>	<p>What other thoughts and feelings might motivate their behavior?</p>		<p>4 What do they SAY? What have we heard them say? What can we imagine them saying?</p>		
<p>5 What do they DO? What do they do today? What behavior have we observed? What can we imagine them doing?</p>					
<p>Last updated on 16 July 2017. Download a copy of this canvas at http://gamestorming.com/empathy-map/ © 2017 Dave Gray, xplane.com</p>					

Brainstorming & Idea Prioritization

Objective:

The objective of this phase was to explore, evaluate, and prioritize innovative ideas to create a seamless, user-friendly digital platform for booking healthcare appointments. Through collaborative ideation and strategic selection, the team focused on identifying features that would immediately resolve real-world scheduling issues while laying a scalable foundation for future healthcare tech enhancements.

Key Activities:

Step 1: Team Collaboration and Problem Statement Selection

- The team convened on 31 January 2025 to discuss gaps in healthcare scheduling systems.
- Through research and shared experiences, the team finalized the problem: *"Manual booking systems in healthcare cause delays, scheduling errors, and communication breakdowns for patients, doctors, and administrators."*

Step 2: Brainstorming, Idea Listing, and Grouping

- All team members contributed ideas openly without judgment.
- Suggestions were grouped into categories:

Patient-Centered Ideas:

- Register/login with role selection (patient, doctor)
- Browse/filter doctors based on specialty, time, and location
- Upload reports/prescriptions during appointment booking
- View appointment history and current status
- Receive confirmation and reminders via alerts/modals

Doctor-Centered Ideas:

- Dashboard showing upcoming appointments
- Ability to confirm/reschedule/cancel bookings
- Access to uploaded patient documents
- Add consultation notes and visit summaries

Admin-Centered Ideas:

- Vet and approve new doctor registrations
- View platform analytics and booking activity
- Manage user access and handle complaints

Stretch/Phase-2 Ideas:

- Real-time chat between patients and doctors
- Video consultation module
- Patient reviews and ratings
- AI-based slot recommendation

Step 3: Idea Prioritization

- The team used an impact–effort scale to select MVP features.
- Priority was given to high-impact, low-to-medium effort features:

Selected for MVP (Phase-1):

- Patient and doctor registration/login
- Doctor directory with filters
- Appointment booking with file uploads
- Admin approval of doctors
- Appointment status updates (Pending, Confirmed, Completed, Cancelled)
- Role-based dashboards for patient, doctor, and admin

Deferred for Future Phases:

- Real-time messaging
- Teleconsultation
- AI-enhanced scheduling
- Feedback/review system

Customer Journey Map

Purpose: To capture and visualize the complete user experience across all stages of interacting with the healthcare appointment system. This helps in identifying user needs, expectations, emotional responses, and areas for improvement.

Key Components:

1. User Actions:

- Registering and logging into the platform.
- Searching for doctors based on specialty, location, or availability.
- Booking, rescheduling, or canceling appointments.
- Receiving notifications and reminders.
- Providing feedback post-consultation.

2. User Thoughts and Emotions:

- *Before:* Anxious about delays and doctor availability.
- *During:* Relief or confidence when finding a match quickly.
- *After:* Satisfaction (or dissatisfaction) depending on ease of use and service quality.

3. Touchpoints:

- Website pages (home, search, doctor profiles, booking form).
- Email/SMS notifications.
- Chat support or FAQs.
- Appointment calendar and dashboard.

4. Pain Points and Opportunities:

- Confusion during registration → Improve onboarding UX.
- Long search times or irrelevant results → Optimize search filters.
- Missed or forgotten appointments → Add real-time reminders.
- Lack of post-visit feedback → Enable user reviews and satisfaction ratings.

Insight: Mapping this journey allows for **user-centered design decisions**, paving the way for a smoother experience, better engagement, and improved retention.

Solution Requirements

1. Functional Requirements:

These define what the system should do.

- **User Registration and Login:** Secure authentication with role-based access (patient, doctor, admin).
- **Doctor Search and Filter:** Search doctors by specialization, availability, location, and consultation mode.
- **Appointment Booking:** Real-time slot selection, with options to reschedule or cancel.
- **Doctor Dashboard:** View upcoming appointments, manage availability, edit profile.
- **Admin Panel:** Manage users, verify doctor credentials, monitor appointments, and control platform content.
- **Notification System:** Send confirmations, reminders, and updates via email/SMS.
- **Feedback Module:** Allow patients to rate and review doctors post-consultation.
- **Responsive UI:** Fully functional on desktop, tablet, and mobile devices.

2. Non-Functional Requirements:

These describe the system's operational attributes.

- **Performance:** Fast response times and minimal downtime.
- **Scalability:** Capable of handling growth in users, doctors, and appointment volume.

- **Security:** Data encryption, secure APIs, and protection against common vulnerabilities (e.g., SQL injection, XSS).
- **Usability:** Clean, intuitive interface for all user roles.
- **Maintainability:** Modular codebase for easier updates and debugging.

3. System Requirements:

These outline the technical stack and environment setup.

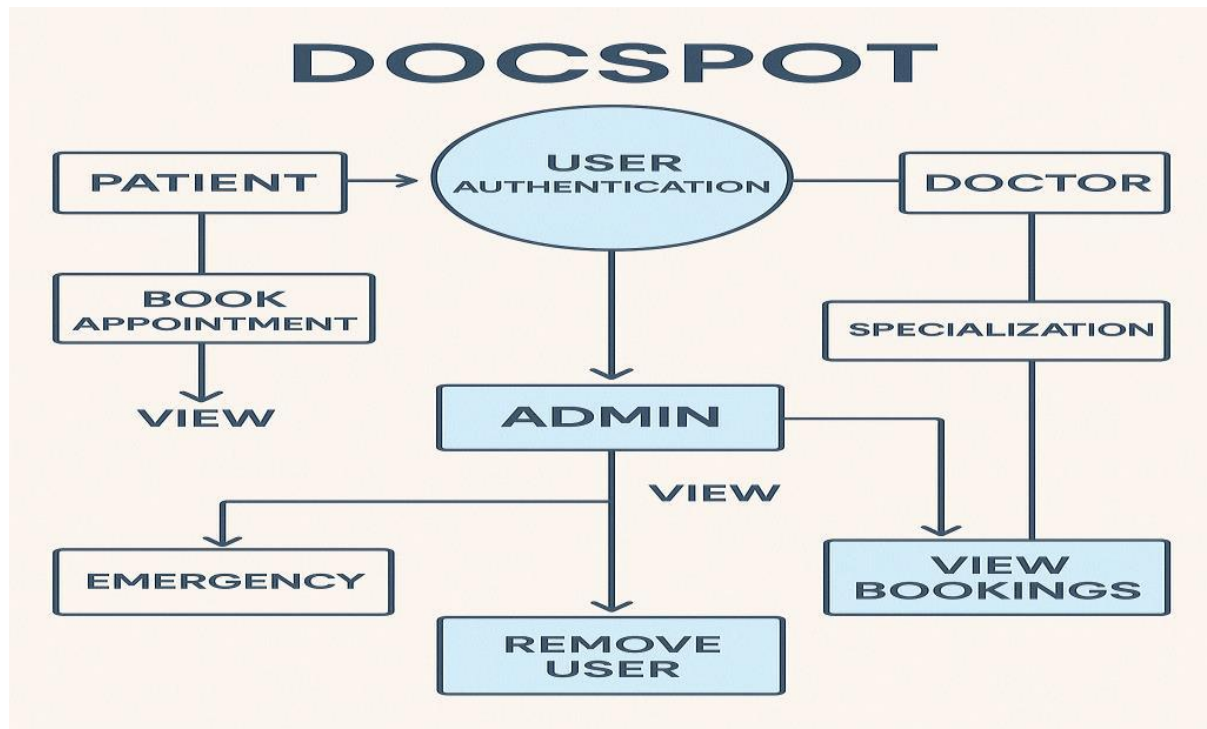
- **Frontend:** React.js with Material UI or Bootstrap.
- **Backend:** Node.js with Express.js..
- **Database:** MongoDB with Mongoose for schema management.
- **Hosting:** Cloud deployment (e.g., Render, Vercel, or AWS).
- **Authentication:** JWT-based login with token expiration.

Data Flow Diagrams:

A **Data Flow Diagram (DFD)** illustrates how data moves within the Freelance Finder platform. It captures how users (freelancers and clients) interact with the system, how information flows between different components, and where the data is stored.

As a Patient, I want to:

- Register and log in securely so I can access my appointments.
- Search for doctors based on specialty, location, and availability to find the right match for my needs.
- Book an appointment by choosing a convenient time and uploading my medical reports.
- View my booking history and track the status of upcoming appointments.
- Reschedule or cancel appointments if my plans change.
- Receive reminders for my upcoming visits via SMS or email.
- Leave a review after my consultation to share my experience.



As a Doctor, I want to:

- Register and create a professional profile with specialization and experience details.
- Set my available time slots so patients can book accordingly.
- View appointment requests and confirm or reject them.
- Access uploaded patient documents before the consultation.
- Add visit notes or digital prescriptions for future reference.
- Manage my daily schedule from a personal dashboard.

As an Admin, I want to:

- Review and approve doctor registrations to maintain platform integrity.
- Monitor bookings and user activity from an admin dashboard.
- Remove or suspend users in case of rule violations or issues.
- View platform analytics like appointment volume or doctor activity.
- Handle user disputes or flagged bookings efficiently.

Technology Stack:

Node.js, Express.js, JWT, Multer

Responsibilities:

- User authentication and authorization (JWT)
- Appointment management and scheduling validations
- Doctor profile approval and role-based routing
- Admin operations, document handling, and notifications
- REST API endpoints to serve frontend requests

Components & Technologies – DocSpot

S.No	Component	Description	Technology Used
1	User Interface	Web-based responsive UI for patients, doctors, and admins	React.js, HTML, CSS, JavaScript, Bootstrap, Tailwind CSS
2	Routing & Navigation	Handles client-side page switching and protected routes	React Router DOM, Context API
3	API Communication	Facilitates frontend-backend interaction	Axios (HTTP client)
4	Authentication System	Secure login and role management	JSON Web Tokens (JWT), bcrypt.js
5	Appointment Logic	Booking, status management, and slot scheduling	Node.js, Express.js
6	Admin Panel	Doctor approval, user management, and appointment oversight	React.js (admin views), Express.js (backend logic)

DocSpot – Application & Characteristics

S.No	Application Module	Key Characteristics
1	User Interface (UI)	Responsive, role-based design using React.js; mobile-friendly and intuitive
2	User Authentication	JWT-based login system with role verification for patients, doctors, and admins
3	Appointment Booking	Real-time slot availability, status tracking (pending/approved/completed)
4	Doctor Profile Management	Editable profiles with specialization, availability, and experience info

Problem–Solution Fit Overview

Purpose:

- Make appointment booking easy and quick.
- Help doctors avoid scheduling conflicts.
- Improve access to care using smart search.
- Reduce missed appointments with reminders.

Problem:

- Booking is complicated and slow.
- Doctors struggle with managing overlapping schedules.
- Poor reminders lead to missed appointments.

- It's hard to find available doctors fast.
- Juggling appointments across systems is a hassle.

Solution:

- Smart filters for fast and easy doctor search.
- Real-time calendar for providers.
- Automatic reminders to patients.
- Secure patient accounts with history.
- Built-in teleconsultation features.
- Admin controls for smooth platform management.

Proposed Solution – DocSpot App

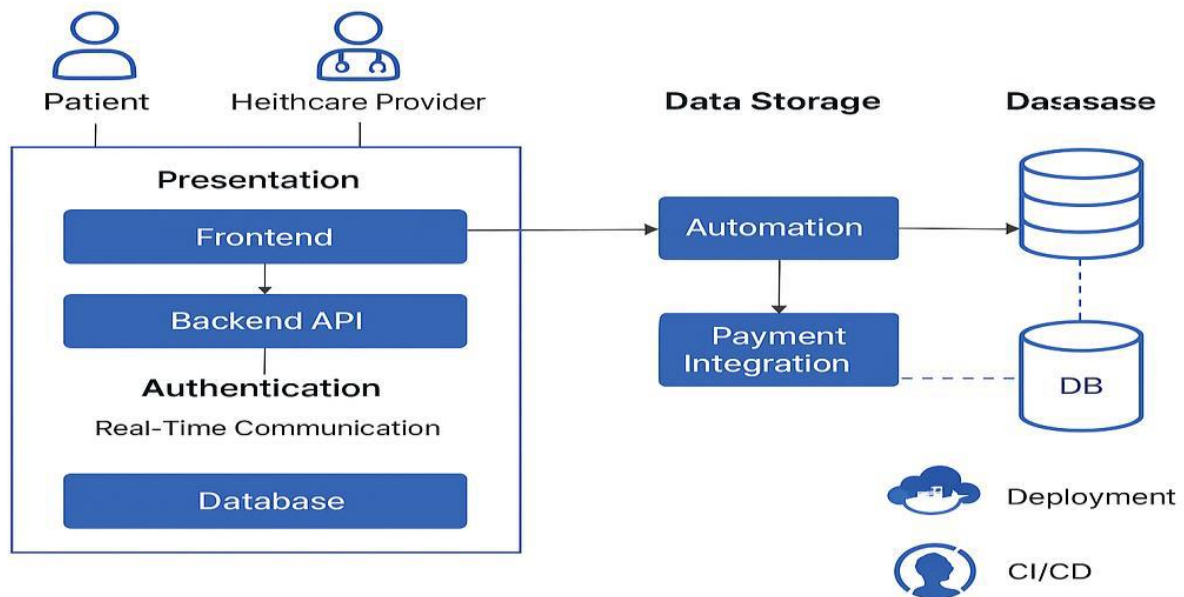
S. No.	Parameter	Description
1	Problem Statement	Patients face delays and confusion; providers struggle with scheduling and communication.
2	Idea / Solution	A full-stack MERN web app offering real-time booking, reminders, secure access, and admin tools.
3	Novel Features	Real-time availability, role-based access, secure payments, SMS/email reminders, video consultations.
4	Social Impact	Reduces no-shows and wait times, expands rural access via telehealth, improves digital communication.
5	Revenue Model	Freemium for clinics, subscriptions for extras, optional transaction fees, white-label deals with hospitals.
6	Scalability	Global-ready (multi-language/timezone), mobile-first design, and modular backend for future features and growth.

Solution Architecture

Solution architecture describes the organization and relationship of the system components. It usually follows architectural principles such as modularity, separation of concerns, and scalability.

Solution Architecture

DocSpot: Seamless Appointment Booking for Health



- ❑ **Client-Server Model:** The system uses a client-server architecture where:
 - o *Client (Frontend)*: Responsible for presenting the interface to users, capturing input, and displaying data.
 - o *Server (Backend)*: Handles business logic, data processing, authentication, and serves data via APIs.
- ❑ **Component Interaction:**
 - o The frontend sends requests to the backend APIs via HTTP calls.
 - o The backend queries the database and returns responses.
 - o The system supports asynchronous communication for notifications and real-time updates.

- ☐ **Data Storage:**

- o User profiles, doctor information, appointments, and records are stored in a database.

- o Data security measures such as encryption and access control are applied.

- ☐ **Extensibility and Maintenance:**

- o Clear modular components allow future enhancements without disrupting the entire system.

- o Use of established frameworks and standards simplifies maintenance and onboarding of new developers.

Architecture diagrams and documents complement this theoretical description to provide visual clarity.

PROJECT PLANNING & SCHEDULING:

Project Planning

Project planning involves defining how the project will be executed, monitored, and controlled to achieve its objectives within constraints like time, cost, and resources.

- ☐ **Scope Definition:** Clearly outline what will and will not be included in the project to avoid scope creep.

- ☐ **Task Breakdown:** Divide the project into smaller manageable tasks or modules (e.g., user registration, doctor management, appointment booking, notifications).

- ☐ **Resource Allocation:** Assign team members, technologies, and tools needed for each task.

- ☐ **Timeline Estimation:** Estimate time durations for each task considering dependencies and critical path.

- ☐ **Milestones & Deliverables:** Define key milestones (e.g., prototype ready, backend APIs complete, UI design finished) and what deliverables are expected.

- ☐ **Risk Management:** Identify potential risks (technical challenges, delays) and mitigation strategies.

- ☐ **Communication Plan:** Set up regular meetings, reporting formats, and stakeholder communication channels.

Effective project planning ensures timely delivery, cost control, and quality outcomes by aligning team efforts and expectations.

FUNCTIONAL AND PERFORMANCE TESTING:

Performance Testing

Performance testing for **DocSpot** focuses on verifying that the application is reliable and responsive under varying loads:

- ☐ **Objectives:**

- o Ensure multiple users can simultaneously search doctors, book, and manage appointments without lag.

- o Maintain quick response times (ideally < 2 seconds) for core functionalities like appointment booking and confirmation.

o Detect and resolve bottlenecks in backend processes (database queries, API endpoints).

□ Testing Types Applied:

o *Load Testing*: Simulate expected daily user volume to ensure stable performance.

o *Stress Testing*: Push beyond typical loads (e.g., high spikes in bookings) to test system limits and graceful degradation.

o *Scalability Testing*: Test how the system handles increasing data (doctors, appointments) to ensure consistent speed.

o *Endurance Testing*: Run the system under normal load for extended periods to check for resource leaks or slowdowns.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	User Authentication	USN-1	As a user, I can sign up and log in securely.	3	High
		USN-2	As a user, I can reset my password if I forget it.	2	Medium
Sprint-2	Doctor Discovery & Booking	USN-3	As a patient, I can search and filter doctors by location and specialty.	2	High
		USN-4	As a patient, I can book appointments based on available time slots.	3	High
Sprint-3	Profile Management	USN-5	As a doctor, I can set and update my profile and availability.	3	High
		USN-6	As a patient or doctor, I can edit and manage my profile details.	2	High
		USN-7	As a user, I can view my past and upcoming appointments.	2	Medium
Sprint-4	Notifications, Reviews & Settings	USN-8	As a user, I receive reminders and alerts for appointments via SMS or email.	2	High
		USN-9	As a user, I can leave a review for a doctor after the appointment.	2	Medium
		USN-10	As a user, I can change my password securely from my dashboard.	2	Medium

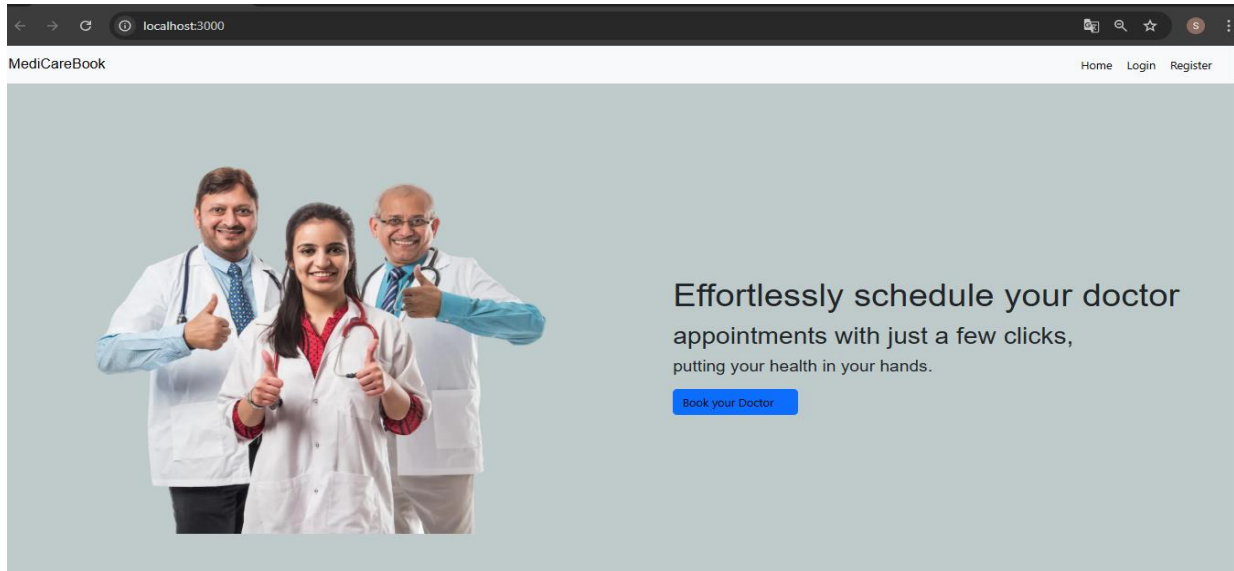
RESULTS:

Output Screenshots

Links :

<https://drive.google.com/drive/folders/1VbMUlliWSSCypsVETeppE4l7Sb0Cpuq?usp=sharing>

LANDING PAGE :



REGISTRATION:


A screenshot of the 'Sign up to your account' registration page on 'MediCareBook'. The page has a light blue header with the site name 'MediCareBook' on the left and navigation links 'Home', 'Login', and 'Register' on the right. The main content area is titled 'Sign up to your account' and contains a registration form with the following fields: 'Full name', 'Email', 'Password', and 'Phone'. Below these fields are two radio buttons for 'Admin' and 'User', with 'User' selected. A black 'Register' button is positioned below the radio buttons. A link that says 'Have an account? Login here' is located below the 'Register' button. At the bottom of the page, there is a cartoon illustration of a doctor in a white coat and stethoscope sitting at a desk, talking to a patient.

REGISTRATION SUCCESS

LOGIN PAGE :

MediCareBook

Home Login Register



Sign in to your account

Email

Password

Login

Don't have an account? Register here

REGISTER FOR APPOINTMENT

Book Appointment with
Doctor

Your Name:

Priya

Your Age:

25

Select Slot:

2025-06-29

Upload Documents:

Choose files No file chosen

Book Appointment

BOOK APPOINTMENT

Book Appointment with
Doctor

Your Name:

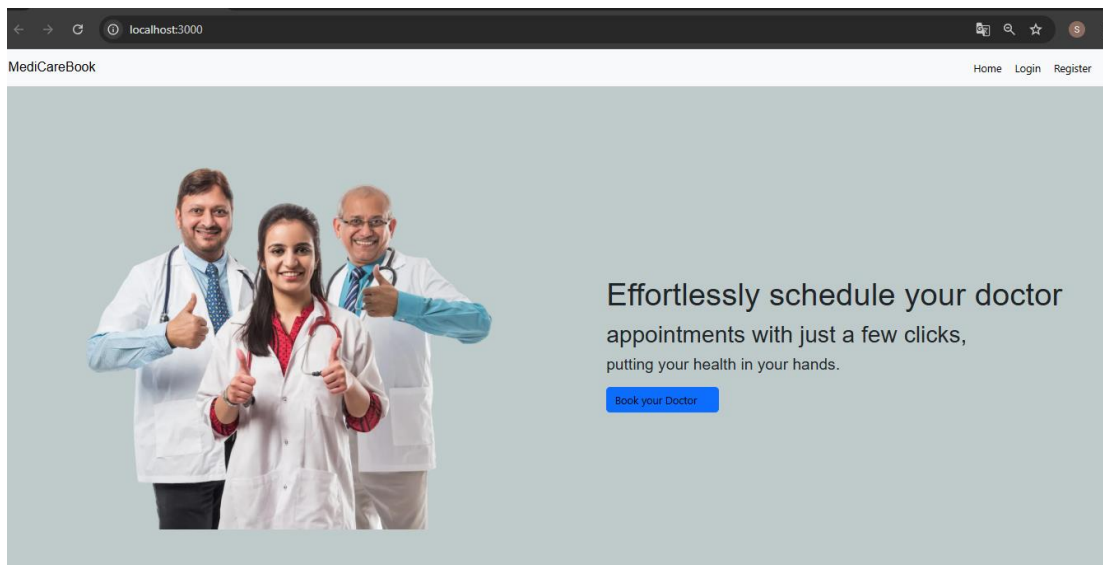
Your Age:

Select Slot:

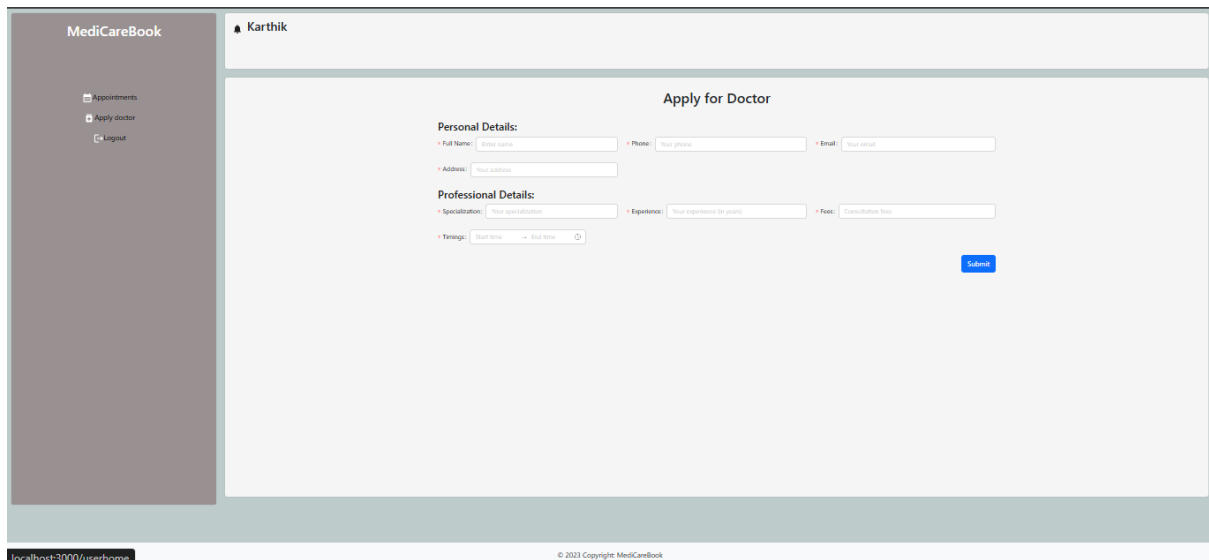
Upload Documents:
 No file chosen

☒ Appointment booked!

BACK TO DASHBOARD



APPLY NOW



ADMIN APPROVED

CONCLUSION:

The "**Book a Doctor Using MERN**" application demonstrates a full-stack implementation of a real-world healthcare booking system. Built using the **MERN stack** (MongoDB, Express.js, React.js, and Node.js), it offers a modern, scalable solution for managing doctor appointments with distinct user roles including **patients**, **doctors**, and **administrators**.

Through well-structured **APIs**, a responsive **user interface**, and robust **authentication and authorization**, the system allows users to register, book appointments, manage schedules, and administer the platform effectively. Security measures like **JWT-based authentication** and **role-based access control** ensure safe and reliable interactions.

While there are a few known limitations (e.g., time zone handling, pagination), the system provides a solid foundation that can be extended and enhanced with additional features such as notifications, analytics, and real-time chat between patients and doctors.

This project serves as both a **learning tool** and a **practical application**, showcasing how to design and develop a complete web-based system using the MERN stack.

FUTURE SCOPE:

Potential areas for future enhancement include:

- ☐ Integration of **video consultations** for telemedicine.
- ☐ Implementation of **AI-driven doctor recommendations** based on patient history.
- ☐ Advanced **analytics dashboard** for doctors and admins to track appointment trends.
- ☐ Mobile app versions for iOS and Android to widen accessibility.
- ☐ Integration with **electronic health records (EHR)** systems for automatic updates.
- ☐ Multilingual support to reach a broader user base.

APPENDIX:

- ☐ **Source Code:** <https://github.com/Vinaymerugumala/code>

□ **Project Demo Link:** <https://drive.google.com/file/d/1CZZVFcpIw96s-zSTn20ECLhh-ulC6zC/view?usp=sharing>