Project Report

Medi Connect – Seamless Appointment Booking for Health

---

1. INTRODUCTION

1.1 Project Overview

Project Title: Medi Connect – Seamless Appointment Booking for Health
Actual name is DocSpot - Seamless Appointment Booking for Health

Medi Connect is a full-stack web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). The system enables patients to search for doctors, check real-time availability, and book appointments online. Doctors can manage schedules and appointments, while administrators monitor and manage system operations.

The platform ensures secure authentication, role-based access control, and real-time appointment management.

Note: During the development phase, the project was initially named "DocSpot" for academic demonstration

purposes.

On 09 February 2026, we received a communication from the official DocSpot administration regarding the unintended usage of:

• The project name "DocSpot"

• Sample email domain "@docspot.com"

As the project is not affiliated with the original DocSpot organization, we immediately took corrective action to avoid any copyright or branding conflicts.

Actions Taken:

1. Removed all occurrences of the name "DocSpot" from:

    o Source code o Repository o Documentation o

    Deployment configuration

2. Replaced all email domains:

    o @docspot.com → @example.com

3. Renamed the project to:

“Medi-Connect — Seamless Appointment Booking for Healthcare”

4. Verified that no branding, trademark, or affiliation references remain in the repository or deployment.

Communication Status:

The issue was resolved amicably, and confirmation was received from the DocSpot administration acknowledging compliance.

1.2 Purpose

The purpose of this project is to:

- Simplify doctor appointment booking
- Reduce manual scheduling errors
- Avoid double booking
- Improve healthcare accessibility
- Provide secure and efficient appointment management
- Enable admin control and analytics

2. IDEATION PHASE

2.1 Problem Statement

Many hospitals and clinics still rely on manual appointment systems or phonebased booking, leading to:

- Long waiting times
- Double bookings
- Poor record management
- Lack of transparency
- Limited accessibility

Patients struggle to find doctors based on specialization and availability.

2.2 Empathy Map Canvas

Patient

Says:

- “I need a quick appointment.”

“I don’t want to wait in queues.”

•

Thinks:

- "Is this doctor available today?"
- "Will I get confirmation?"

Does:

- Searches doctors
- Books appointment
- Cancels/reschedules **Feels:**
- Frustrated with delays
- Relieved after confirmation

Doctor

Says:

- "I need organized schedule management." **Thinks:**
- "How many appointments today?"

Does:

- Confirms appointments
- Updates availability

Feels:

- Overloaded without automation
- Organized with system

2.3 Brainstorming

Ideas generated during brainstorming:

- Online doctor search by specialization
- Real-time slot availability
- Email notifications
- Admin approval workflow
- Dashboard for patients & doctors
- Appointment rescheduling & cancellation

Secure authentication using JWT

Final selected idea:
Online Doctor Appointment Management System with real-time validation and role-based dashboards.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Patient Journey

1. Visit website
2. Register/Login
3. Search doctor
4. View profile
5. Select time slot
6. Book appointment
7. Receive confirmation
8. Attend appointment **3.2 Solution Requirement**

Functional Requirements

- User Registration & Login
- Email Verification
- Doctor Registration & Admin Approval
- Doctor Search & Filter
- Appointment Booking
- Reschedule & Cancel
- Email Notifications
- Admin Dashboard
- Role-Based Access

Non-Functional Requirements

- Secure Authentication (JWT)
- Response Time < 2 seconds
- 99% availability

- Data Encryption

  Mobile Responsiveness

3.3 Data Flow Diagram Level 0 (Context Diagram)

Patient → MediConnect System → Doctor
Admin → MediConnect System

Data Stored in MongoDB Database

3.4 Technology Stack

Layer	**Technology**
Frontend	React.js
Backend	Node.js, Express.js
Database	MongoDB Atlas
Authentication	JWT
Hosting (Frontend)	Vercel
Hosting (Backend)	Render
Version Control	GitHub

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem: Manual booking causes delays and double booking.

Solution:

Online appointment system with:

- Real-time availability check
- Automatic slot validation
- Email confirmation
- Role-based dashboards

4.2 Proposed Solution

Parameter	**Description**
Problem Statement	Manual and inefficient appointment booking
Idea	Online doctor appointment booking system
Novelty	Real-time slot validation with role-based dashboards
Social Impact	Saves time, reduces hospital crowd
Business Model	Subscription-based or commission per appointment
Scalability	Can expand to multiple hospitals & telemedicine

4.3 Solution Architecture Architecture Flow:

Frontend (React - Vercel)
↓
Backend API (Node/Express - Render)
↓
MongoDB Atlas Database Features:

- REST API architecture
- JWT Authentication
- Role-Based Access Control
- Cloud Deployment

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Sprint Model Agile

Sprint	Functional Requirement (Epic)	User Story No	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	Register using email & password	3	High	Jaya Prakash
Sprint-2	User Confirmation	USN-2	Email verification link	2	High	Mounika
Sprint-3	Login	USN-3	Login using email & password	2	High	Srinivasulu
Sprint-4	Dashboard	USN-4	View patient dashboard	3	High	Jaya Prakash
Sprint-5	Profile Management	USN-5	Edit user profile	3	Medium	Mounika
Sprint-6	JWT Security	USN-6	Secure authentication	2	High	Srinivasulu

Velocity ≈ 16 Story Points per Sprint

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- API response time < 2 sec
- Page load time < 3 sec
- Double booking prevented
- Role-based route protection verified
- Security testing for SQL Injection & XSS

Load tested with concurrent users (basic simulation).

7. RESULTS

7.1 Output Screenshots

Check this in my deployed links :

- Home Page
- Login Page
- Patient Dashboard
- Doctor Dashboard
- Admin Dashboard
- Appointment Booking Page
- MongoDB Database Collections

8. ADVANTAGES & DISADVANTAGES

Advantages

- Easy booking process
- Saves time
- Prevents double booking
- Secure authentication
- Cloud hosted

Disadvantages

- Requires internet connection
- Dependent on server availability
- Basic version does not include video consultation

9. CONCLUSION

MediConnect successfully provides a scalable and secure healthcare appointment management platform. The system improves efficiency, reduces manual errors, and enhances patient experience. It demonstrates strong fullstack development skills and real-world deployment capability.

10. FUTURE SCOPE

- Online video consultation
- AI-based doctor recommendation
- Payment gateway integration
- Mobile app version
- Multi-language support
- Insurance integration

11. APPENDIX

Source Code GitHub

Repository:

https://github.com/jayaprakashroya/MediConnect-Seamless-AppointmentBooking-for-Health **Backend Deployment** https://docspot-seamless-appointment-booking-for-w39m.onrender.com **Frontend**

Deployment https://medi-connect-seamless-appointment-b.vercel.app

Project Demo Link

https://drive.google.com/file/d/1l2GIAVBAJUJ6TdQWgtPZA_cZxLefGPZ/view

Dataset

MongoDB Atlas (docspot-cluster seeded database)