



JEPPIAAR ENGINEERING COLLEGE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BOOK A DOCTOR USING MERN



TEAM LEADER: RAJ KUMAR N (310821104076)

TEAM MEMBER 1: RAJARAJESHWARAN N (310821104075)

TEAM MEMBER 2: VIJAYA KUMAR S (310821104110)

TEAM MEMBER 3: VISHWANATH R R (310821104116)

SUPERVISOR NAME: MRS.D.JEEVITHA .M.E

ASSISTANT PROFESSOR /CSE



BOOK A DOCTOR

USING MERN STACK



Table of Contents

- 1 Introduction
- 2 Project Overview
- 3 Project Architecture
- 4 Setup Instructions
- 5 Folder Structure
- 6 Running the Application
- 7 API Documentation
- 8 Authentication
- 9 User Interface
- 10 Testing
- 11 Screenshots or Demo
- 12 Known Issues
- 13 Future Enhancements
- 14 Conclusion
- 15 Gratitude

Introduction



Overview : The "Book a Doctor" project is a web-based application designed to simplify appointment booking and management for patients, doctors, and administrators.

Objective : Provide a seamless platform for healthcare management, enhancing accessibility and efficiency for all users.

Scope : The system includes role-based functionalities for users, doctors, and admins, offering a complete healthcare solution .

Our team



Raj Kumar N

(Backend &
Testing)



Rajarajeshwaran N

(UI & UX
Design)



Vijaya Kumar S

(Database &
Routing)



Vishwanath R R

(Full Stack
Developer)




Project Overview

Purpose:

- Provide a seamless platform for booking and managing doctor appointments.
- Enable efficient interaction between users, doctors, and admin.

Features:

- Role-based login (Admin, Doctor, User).
- Appointment booking and approval process.
- Notifications for users, doctors, and admin.



MongoDB : Stores data in JSON-like documents ,
Used for managing admin, doctor, and user data
efficiently .

Express.js : Backend framework for handling API
routes and server-side logic.

React.js : Frontend library used for creating dynamic
and responsive user interfaces .

Node.js : Runtime environment for executing
JavaScript on the server .

Git and Github : Version control system and developer
platform to store, manage, and share their code .

**Technologies
used** 



Project Architecture

Frontend : Built using React.js for Single page application and interactive UI .

Backend : Express.js and Node.js manage API requests , Authorization , Authentication with other business logic .

Database : MongoDB stores user, doctor, and admin data in JSON format and we can access , modify and delete the data .

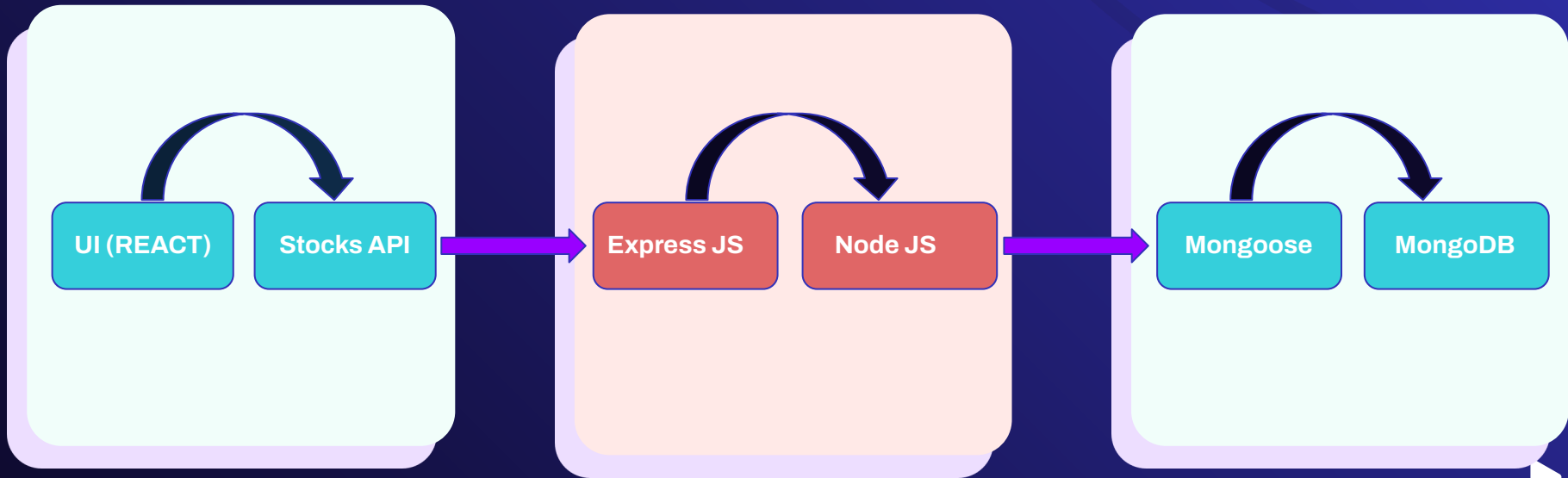
Client-Server Interaction : RESTful APIs connect the frontend and backend .

Architecture

Frontend

Backend

Database





Prerequisites:

- Node.js (v14 or later).
- MongoDB (e.g., MongoDB Compass).
- Code editor (e.g., VS Code).

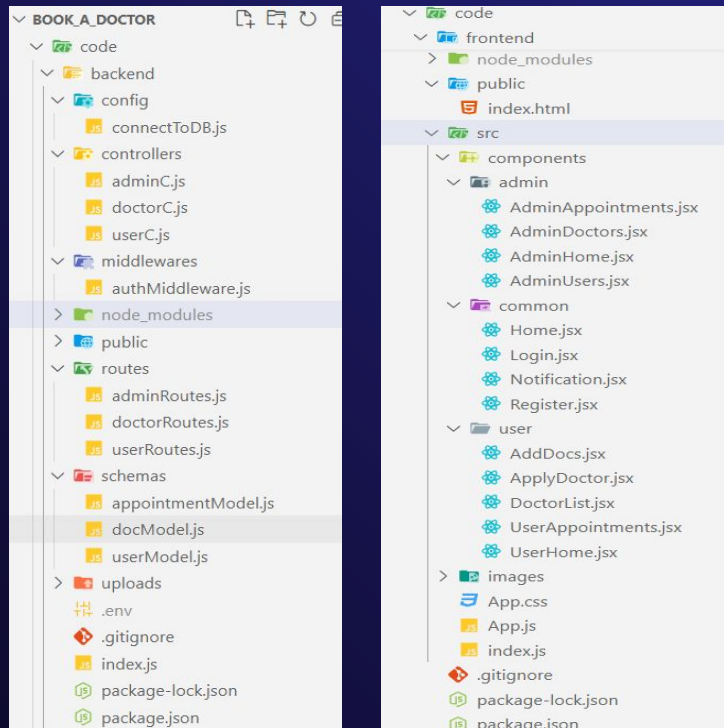
Installation:

- Clone the repository from GitHub.
- Run npm install in both the client and server directories.
- Set up environment variables for MongoDB connection and JWT secrets.

Setup Instructions



• Folder Structure



Client:

- Components, Pages, and Assets folders for React frontend.
- Organized by feature (e.g., LoginPage, Admin Dashboard).

Server:

- **Routes:** Defines API endpoints for users, doctors, and admin.
- **Models:** MongoDB schemas for user, doctor, and appointment data.
- **Controllers:** Handles business logic for each endpoint.

Frontend:

- Navigate to the client directory.
- **Run:** npm start to start the React development server.

>> npm start

Backend:

- Navigate to the server directory.
- **Run:** npm start to start the Node.js backend server.

Running The Application

API Documentation

Endpoints:

- /api/users: User registration, login, and profile management.
- /api/doctors: Doctor registration, login, and appointment handling.
- /api/admin: Manage user and doctor data, approve/reject doctor applications.

Details:

Include request methods (GET, POST, etc.), required parameters, and sample JSON responses



Implementation:

- JWT tokens for secure authentication and role-based access.
- Middleware to validate tokens and ensure access rights.

Process:

- Tokens issued upon successful login and stored in local storage.
- Admin, Doctor, and User roles verified via middleware.

Authentication



User Interface

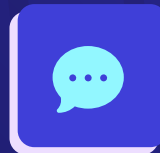
- **Landing Page:** Intuitive design with navigation to login/register.
- **Login/Registration:** Role-based forms with input validation.
- **User Dashboard:** Book appointments, view history, and apply as a doctor.
- **Doctor Dashboard:** View patient appointments and approve statuses.
- **Admin Dashboard:** Manage users, doctors, and application requests.
- **Responsive Design:** Optimized for desktops, tablets, and mobiles.
- **Consistent Design:** Uniform color scheme and user-friendly interface.
- **Notifications:** Accessible for all roles to stay updated.



Doctor Features

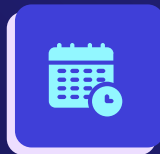
Login

Doctors Login with secure credentials.



Notifications

Receive updates on new appointments or approvals.



View Appointments

Access a list of patient appointments.
Update status after each appointment.

Admin Features

Login

Secure access to the admin panel.



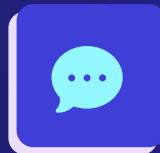
Dashboard

Dashboard to view statistics and data.



Notifications

Receive updates on pending doctor requests.



Manage Users and Doctors

View lists of registered users and doctors.
Accept or reject doctor applications.



Notifications



Users

Updates on
appointment status.



Doctors

Alerts on new
appointments or
approvals.



Admins

Notifications for
pending actions like
doctor applications.

Landing Page

A good doctor treats the disease

Book your appointments with just a few clicks.

[Book your Doctor](#)



Register Page

Sign up to your account

Full name

Email

Password

Phone

☐ Admin
 ☒ User

Register

Have an account? [Login here](#)

Login Page

Sign in to your account

Email

Password

Login

Don't have an account? [Register here](#)

All Appointments for Admin Panel

Appointment ID	User Name	Doctor Name	Date	Status
67385c262762c39ce54f7a89	User	Doctor	2024-11-18 18:00	approved
67389c652762c39ce54f7b07	Raj	Al	2024-11-17 17:51	approved
67389c982762c39ce54f7b10	Ram	Doctor	2024-11-18 17:00	approved
6738a4b02762c39ce54f7b46	Raj	Doctor	2024-11-19 10:30	approved
6738b2492762c39ce54f7bf2	Raj	Einstein	2024-11-19 17:30	pending
6738b4d42762c39ce54f7c39	Raj	Test Doc	2024-11-20 20:35	approved

Admin Dashboard

All Users

Name	Email	Phone	isAdmin	isDoctor
Admin	admin@gmail.com	4598763210	admin	No
Doctor	doctor@gmail.com	7894561230	user	Yes
User	user@gmail.com	1230456987	user	No
Ram	ram@gmail.com	7894563210	user	No
Raj	raj@gmail.com	1236547890	user	No
Albert	albert@gmail.com	1478523690	user	Yes
User2	user2@gmail.com	7896541230	user	Yes
Test	test@gmail.com	479651302	user	Yes

Users List

All Doctors

Key	Name	Email	Phone	Action
67385b5e2762c39ce54f7a1e	Doctor	doctor@gmail.com	7894561230	<div>Reject</div>
67389bcd2762c39ce54f7adb	Albert	albert@gmail.com	7894563210	<div>Reject</div>
6738b1492762c39ce54f7bc8	Einstein	einstein@gmail.com	4785693210	<div>Reject</div>
6738b4422762c39ce54f7c1b	Test Doc	test@gmail.com	4789651230	<div>Reject</div>





Doctors List

- 🏠 HomePage
- 📅 Appointments
- 👨⚕️ Apply doctor
- 🚪 Logout

Home

**Dr. Doctor**Phone: **7894561230**Address: **Chennai**Specialization: **Neurologist**Experience: **2 Yrs**Fees: **2000**[Book Now](#)**Dr. Albert**Phone: **7894563210**Address: **Bangalore**Specialization: **Orthology**Experience: **5 Yrs**Fees: **1000**[Book Now](#)**Dr. Einstein**Phone: **4785693210**Address: **Hyderabad**Specialization: **Cardiology**Experience: **4 Yrs**Fees: **1000**[Book Now](#)**Dr. Test Doc**Phone: **4789651230**Address: **Chennai**Specialization: **Dentist**

User Dashboard

-  HomePage
-  Appointments
-  Apply doctor
-  Logout

Notification

UnRead [Read](#)

Delete all read

- your appointment get approved
- your appointment get approved
- your appointment get approved

User
Notifications

Apply for Doctor


Personal Details:

* Full Name: * Phone: * Email:

* Address: * Doctor image: No file chosen

Professional Details:

* Specialization: * Experience: * Fees:

* Timings: → 

Submit

Doctor Application

Home

Booking appointment

Doctor Details:

Name: Albert

Specialization: **Orthology**

Appointment Date and Time:

mm/dd/yyyy --:-- --



Documents

Choose File

No file chosen

Close

Book

Dr. Do

Phone: 7

Address:

Specializ

Experience: **2 Yrs**Fees: **2000**





Book Now

Experience: **5 Yrs**Fees: **1000**

Book Now

Appointment




-  [HomePage](#)
-  [Appointments](#)
-  [Apply doctor](#)
-  [Logout](#)

All Appointments

Doctor Name	Date of Appointment	Status
Albert	2024-11-17 17:51	approved
Doctor	2024-11-19 10:30	approved
Einstein	2024-11-19 17:30	pending
Test Doc	2024-11-20 20:35	approved

User
Appointments

-  [HomePage](#)
-  [Appointments](#)
-  [Logout](#)

Doctor Details



Name: **Albert**

Phone : **7894563210**

Email : **albert@gmail.com**

Specialization : **Orthology**

Fees : **1000**

Address : **Bangalore**

Doctor Dashboard

Notification

UnRead

[Read](#)

Delete all read

Your Doctor account has approved

New Appointment request from Raj

Doctor
Notifications

✔ All the appointments are listed below.

✔ All the appointments are listed below.

✔ All the appointments are listed below.

Appointments

Name	Date of Appointment	Phone	Document	Status	Action
Raj	2024-11-17 17:51	1236547890	2d76eb16698721ec3db3a2733dd3c42b	approved	

Appointments made
to the doctor



Strategy:

- Unit testing for backend APIs, and user interaction.
- Integration testing to test the overall functionalities of the application.
- Manual testing for UI responsiveness and workflows.

Tools:

- Postman for API testing, browser tools for frontend testing. Github error log to check for errors.

Testing



Known Issues

- Real-time notification delay under high load.
- Limited error handling for invalid user inputs.
- Mobile Responsiveness Issues
- No Email/SMS Notifications
- No Payment Integration
- No Appointment Rescheduling/Cancellation



- Automated Doctor Approval
- Real-Time Availability Updates
- Appointment Management
- Advanced Search Filters
- Enhanced Admin Dashboard
- Mobile Optimization
- Email/SMS Notifications
- Payment Integration
- Real-time chat system

Future Enhancements



Conclusion

Summary:

- Successfully developed a full-stack web application for healthcare management.
- Delivered role-based functionalities for users, doctors, and admins.

Future Enhancements:

- Integrate video consultation features.
- Implement AI for doctor recommendations.
- Add payment gateway for online consultation fees.

+

▶

Thank You!

•

•

