Doctor Appointment Booking System (MERN Stack)

Final Internship Report

Submitted by:
Team Leader: J Yamini
Team Members: Bobby Mullagiri, Komati Nandini
Team ID: LTVIP2025TMID53677
Institution: [Sri Sarathi institute of Engineering & technology]
Date: June 2025

---

1. Introduction

The “Book a Doctor Appointment” web application is an advanced healthcare solution built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It enables seamless appointment booking for patients and simplifies doctor-patient communication. With a responsive frontend, secure authentication, and robust backend architecture, this platform serves patients, doctors, and administrators efficiently.

Key Technologies

- Frontend: React.js, Bootstrap, Material UI
- Backend: Node.js, Express.js, MongoDB, JWT, Bcrypt, Axios
- Scheduling: Moment.js

---

2. Features

Patient Features

- Registration and profile management
- Search doctors by location, specialty, and availability
- Book, cancel, and reschedule appointments
- Receive notifications and reminders

Doctor Features

- Profile creation and availability scheduling
- View, confirm, or cancel appointments
- Access patient history and provide post-visit summaries

Admin Features

- Manage user and doctor accounts

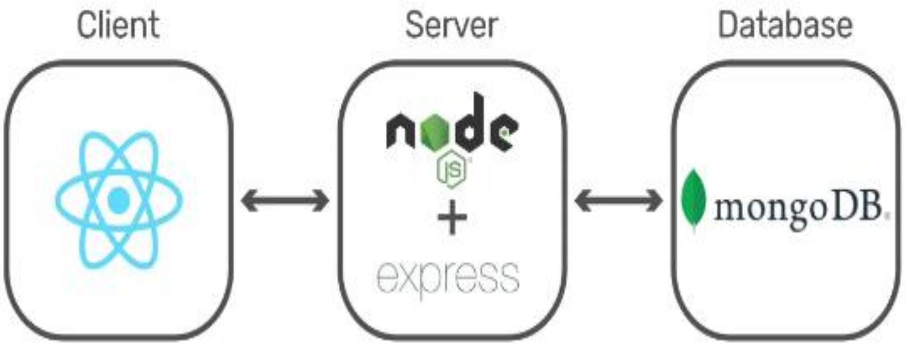
- System monitoring and compliance
- Approve doctor registrations

Common Features

- Responsive design for mobile and desktop
- Secure authentication (JWT + Bcrypt)
- Real-time updates and notifications

---

3. Technical Architecture



MERN Stack Overview

- **Frontend (React.js):** Interactive UI with React components using Material UI and Bootstrap.
- **Backend (Node.js & Express.js):** RESTful APIs, authentication, and role-based access control.
- **Database (MongoDB):** Stores users, appointments, and doctor profiles.

Deployment & Security

- Hosted on platforms like Heroku or MongoDB Atlas
- Secure user sessions using JWT
- Password hashing using Bcrypt
- Environment variables for sensitive data (.env file)

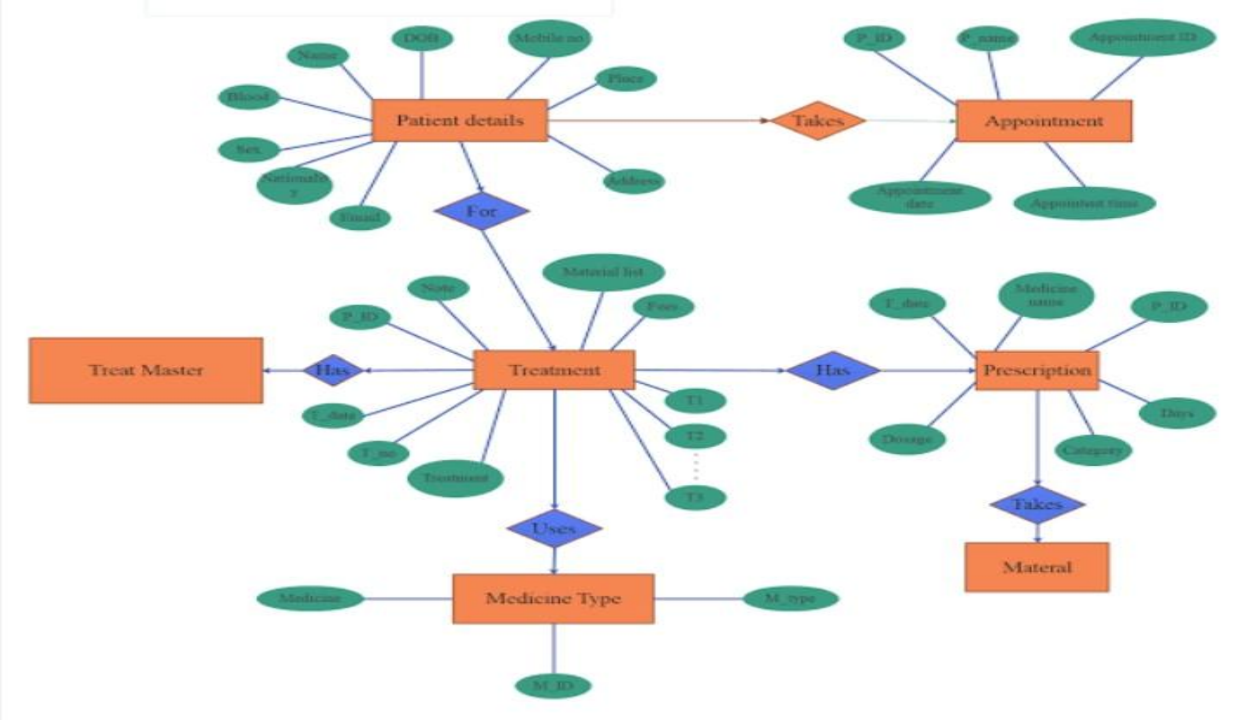
---

4. Scenario-Based Case Study

1. **User Registration:** John signs up and creates a patient profile.
2. **Doctor Browsing:** Searches for family physicians nearby.

3. **Appointment Booking:** Selects date/time and uploads records.
4. **Confirmation:** Dr. Smith approves the appointment.
5. **Reminders:** John receives SMS/email notifications.
6. **Doctor Dashboard:** Manages all scheduled appointments.
7. **Admin Review:** Verifies and approves new doctor accounts.
8. **Post-Visit:** Dr. Smith uploads follow-up notes and prescriptions.

5. Entity-Relationship Diagram (ERD)



Entities:

- Doctor (DoctorID, Name, Specialty, Availability, etc.)
- Patient (PatientID, Name, Contact, etc.)
- Appointment (AppointmentID, DoctorID, PatientID, Status, etc.)
- Admin/User (UserID, Role)

Relationships:

- One doctor to many appointments
- One patient to many appointments
- One admin managing many doctors

6. Project Setup & Installation

Prerequisites

- Node.js and npm
- MongoDB or MongoDB Atlas
- Code Editor (e.g., VS Code)
- Git for version control

Backend Setup

```
npm init -y
npm install express mongoose cors bcryptjs jsonwebtoken dotenv multer
```

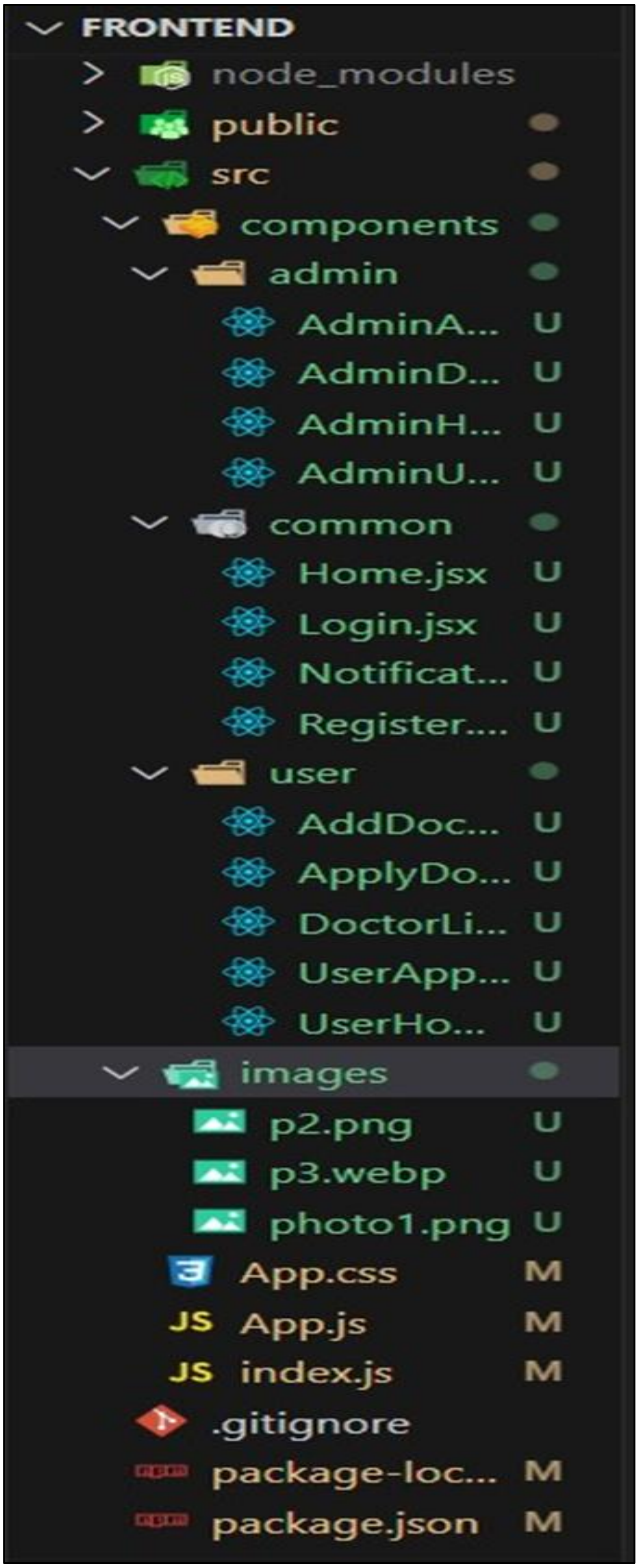
- Create `server.js`, connect MongoDB, define models and routes.

Frontend Setup

```
npx create-react-app client
cd client
npm install axios react-router-dom
```

- Develop components (Login, Register, BookAppointment, Dashboard)

7. Folder Structure

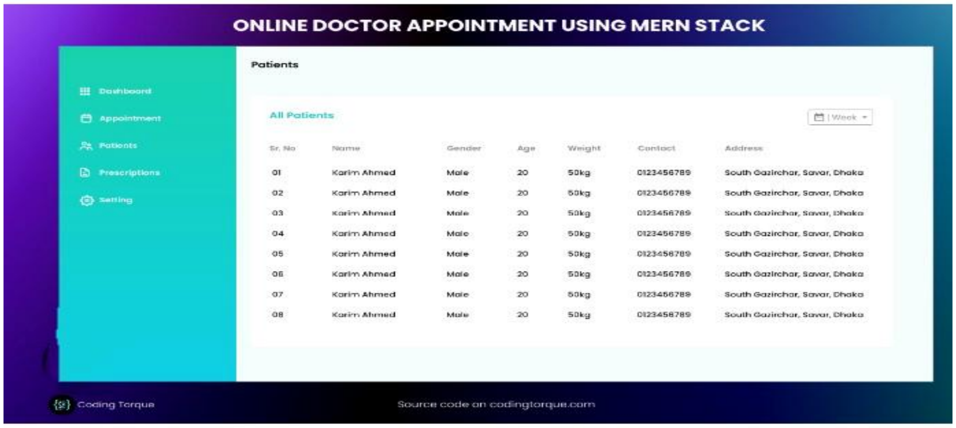


```
FRONTEND
  > node_modules
  > public                 ●
  ∨ src                    ●
    ∨ components           ●
      ∨ admin              ●
          AdminA...        U
          AdminD...        U
          AdminH...        U
          AdminU...        U
      ∨ common             ●
          Home.jsx         U
          Login.jsx        U
          Notificat...     U
          Register....     U
      ∨ user               ●
          AddDoc...        U
          ApplyDo...       U
          DoctorLi...      U
          UserApp...       U
          UserHo...        U
    ∨ images               ●
        p2.png             U
        p3.webp            U
        photo1.png         U
      App.css              M
      App.js               M
      index.js             M
    .gitignore
    package-loc...         M
    package.json           M
```

```
project-root/
├── backend/
│   ├── config/
│   ├── controllers/
│   ├── models/
│   ├── routes/
│   └── server.js
├── frontend/
│   ├── src/components/
│   ├── src/pages/
│   └── App.js
│
```

---

9. Application Flow



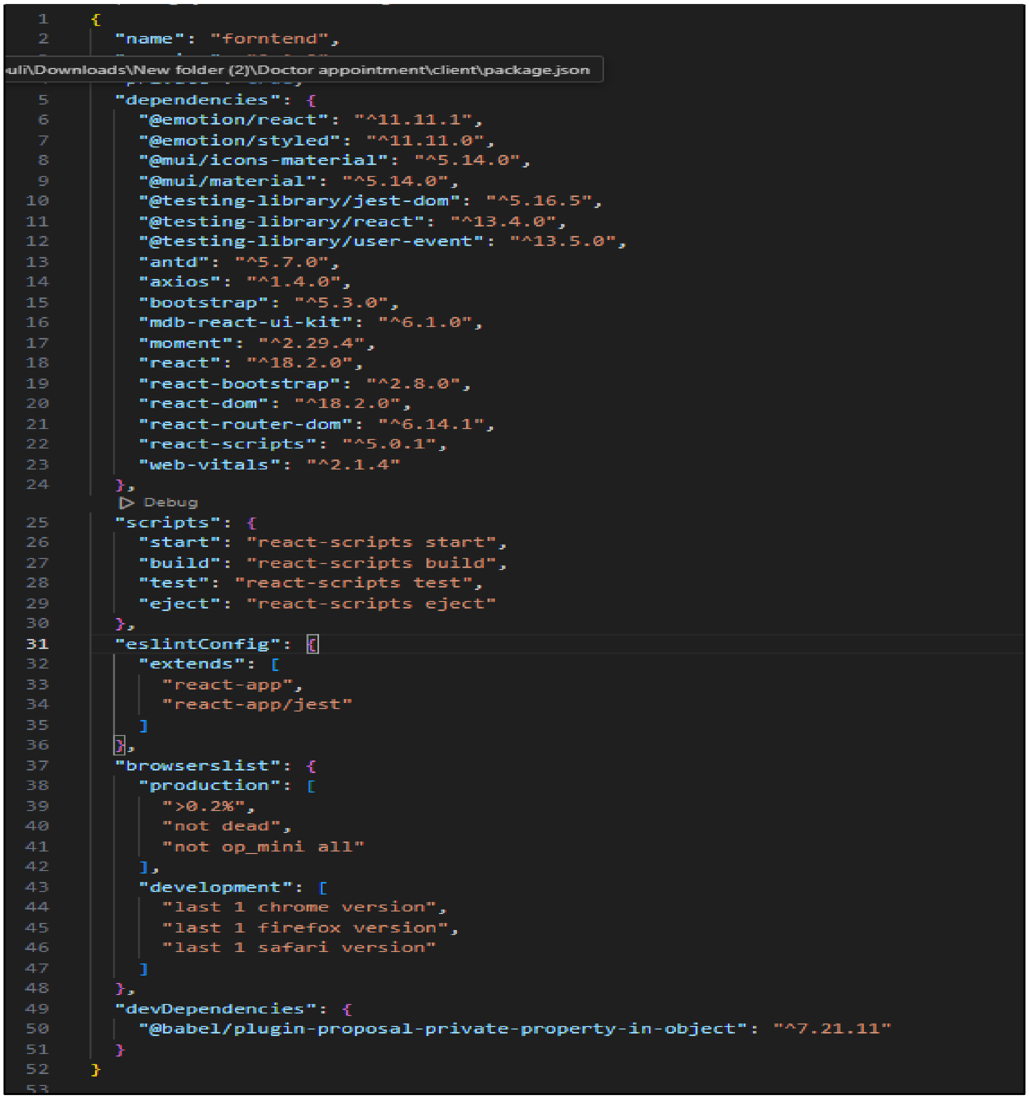
Sr. No	Name	Gender	Age	Weight	Contact	Address
01	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
02	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
03	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
04	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
05	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
06	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
07	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka
08	Karim Ahmed	Male	20	50kg	0123456789	South Gazirchar, Savar, Dhaka

Coding Torque

Source code on codingtorque.com

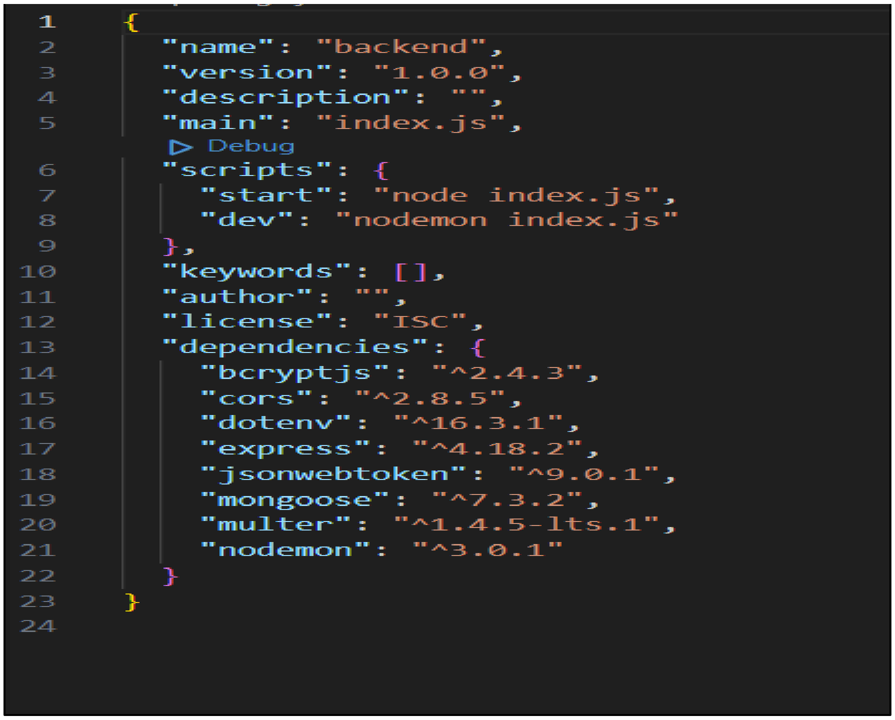
1. User registers and logs in
2. Patient browses doctors and books an appointment
3. Doctor confirms and reviews upcoming schedule
4. Admin monitors system activity and compliance
5. Post-visit summaries are recorded

6.



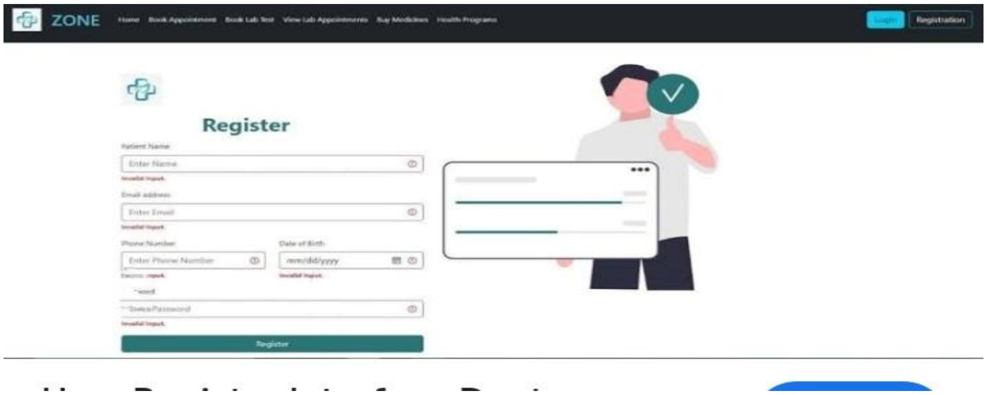
uli\Downloads\New folder (2)\Doctor appointment\client\package.json

```json
{
  "name": "forntend",
  "dependencies": {
    "@emotion/react": "^11.11.1",
    "@emotion/styled": "^11.11.0",
    "@mui/icons-material": "^5.14.0",
    "@mui/material": "^5.14.0",
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^5.7.0",
    "axios": "^1.4.0",
    "bootstrap": "^5.3.0",
    "mdb-react-ui-kit": "^6.1.0",
    "moment": "^2.29.4",
    "react": "^18.2.0",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.14.1",
    "react-scripts": "^5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "@babel/plugin-proposal-private-property-in-object": "^7.21.11"
  }
}
```



```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "dev": "nodemon index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.1",
    "mongoose": "^7.3.2",
    "multer": "^1.4.5-lts.1",
    "nodemon": "^3.0.1"
  }
}
```

Register page:



Register

Patient Name

Enter Name

Invalid input.

Email address

Enter Email

Invalid input.

Phone Number

Enter Phone Number

Invalid input.

Date of Birth

mm/dd/yyyy

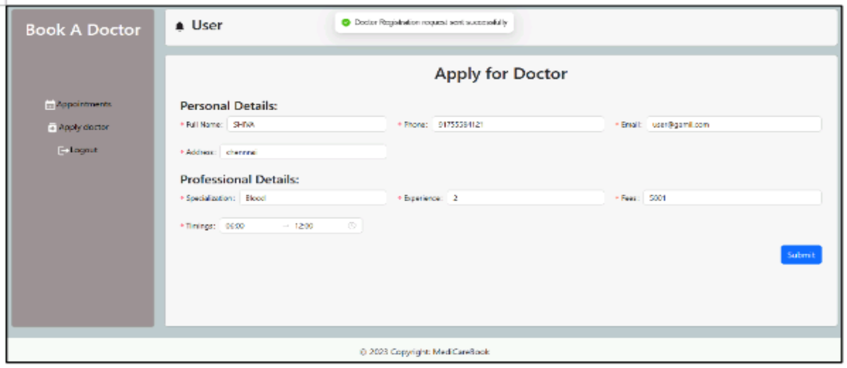
Invalid input.

Password

Enter Password

Invalid input.

Apply as Doctor:



Doctor Registration request sent successfully

Apply for Doctor

Personal Details:

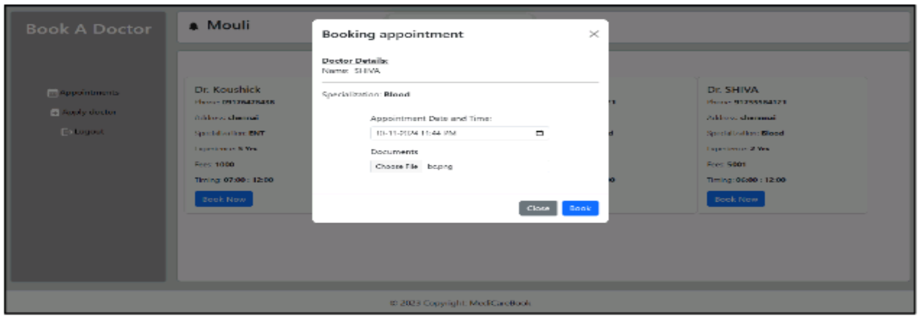
- Full Name: SHIVA
- Phone: 91755384121
- Email: user@gamil.com
- Address: chennai

Professional Details:

- Specialization: Blood
- Experience: 2
- Fees: 5001
- Timings: 06:00 → 12:00

© 2023 Copyright: MediCareBook

Book Doctor :



Dr. Koushick
Phone: [illegible]
Address: chennai
Specialization: ENT
Experience: 5 Yrs
Fees: 1000
Timing: 07:00 : 12:00

Dr. SHIVA
Phone: 91755384121
Address: chennai
Specialization: Blood
Experience: 2 Yrs
Fees: 5001
Timing: 06:00 : 12:00

Booking appointment

Doctor Details:
Name: SHIVA

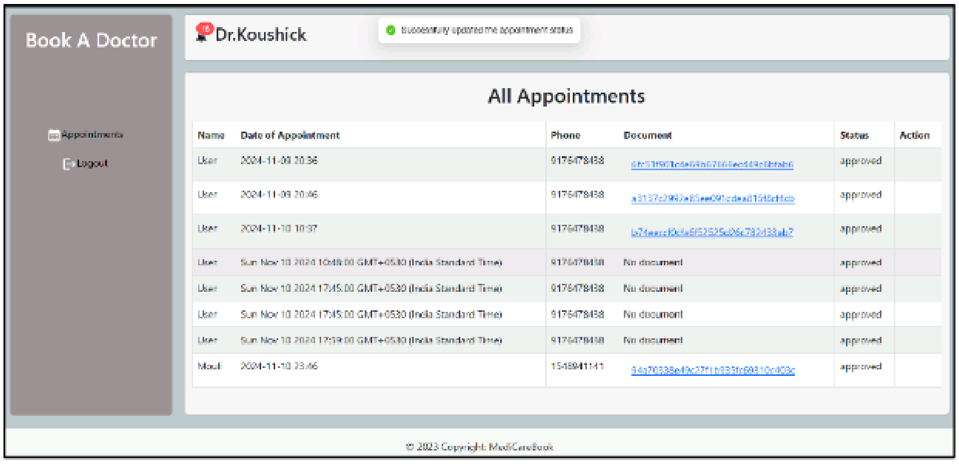
Specialization: Blood

Appointment Date and Time:
10-11-2024 11:44 PM

Documents
Choose File bc.png

© 2023 Copyright: MediCareBook

Doctor Approve User appointment :



Successfully updated the appointment status

All Appointments

Name	Date of Appointment	Phone	Document	Status	Action
User	2024-11-09 20:36	9176478438	[illegible]	approved	
User	2024-11-09 20:46	9176478438	[illegible]	approved	
User	2024-11-10 10:37	9176478438	[illegible]	approved	
User	Sun Nov 10 2024 10:48:00 GMT+0530 (India Standard Time)	9176478438	No document	approved	
User	Sun Nov 10 2024 17:45:00 GMT+0530 (India Standard Time)	9176478438	No document	approved	
User	Sun Nov 10 2024 17:45:00 GMT+0530 (India Standard Time)	9176478438	No document	approved	
User	Sun Nov 10 2024 17:59:00 GMT+0530 (India Standard Time)	9176478438	No document	approved	
Mauli	2024-11-10 23:46	1548941141	[illegible]	approved	

© 2023 Copyright: MediCareBook

All History:

---

9. Project Demonstration & Resources

- **Source Code:** GitHub Repository

---

10. Conclusion

This project demonstrates a complete, scalable, and secure medical appointment booking platform. It enables effective interactions between patients, doctors, and administrators. The MERN stack ensures performance, maintainability, and scalability, making it ideal for real-world healthcare systems.

References

1. MongoDB Atlas - https://www.mongodb.com/cloud/atlas
2. React.js - https://reactjs.org/
3. Node.js - https://nodejs.org/en/
4. Express.js - https://expressjs.com/
5. Material UI - https://mui.com/
6. Bootstrap - https://getbootstrap.com/
7. JWT Auth - https://jwt.io/
8. Bcrypt - https://www.npmjs.com/package/bcryptjs