

# **NAAN MUDHALVAN PROJECT REPORT**

**COLLEGE : AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING**

**PROJECT TITLE : BOOK A DOCTOR USING MERN**

**Department of Computer Science and Engineering**

## **TEAM MEMBERS:**

**MOHAMED FAIZAAN (110121104046)**

**MOHAMED ASLAM (110121104044)**

**SUKUMARAN (110121104319)**

**MUNNAF (110121104312)**

## INTRODUCTION:

Booking a doctor's appointment has never been easier. With our convenient online platform, you can quickly and effortlessly schedule your appointments from the comfort of your own home. No more waiting on hold or playing phone tag with busy receptionists. Our user-friendly interface allows you to browse through a wide range of doctors and healthcare providers, making it simple to find the perfect match for your needs.

With our advanced booking system, you can say goodbye to the hassle of traditional appointment booking. Our platform offers real-time availability, allowing you to choose from a range of open slots that fit your schedule. Whether you prefer early morning, evening, or weekend appointments, we have options to accommodate your needs.

### Pre-requisites for Developing BOOK A DOCTOR USING MERN

To develop a Doctor Booking website using **React JS**, **Node.js**, and **MongoDB**, here are the key prerequisites you need to consider:

#### 1. Node.js and npm

Node.js is required to run JavaScript code on the server side. npm (Node Package Manager) is included with Node.js, and it's used to manage project dependencies.

- **Download Node.js:**  
[Node.js download page](#)
- **Installation Instructions:**  
[Node.js installation guide for various package managers](#)

---

#### 2. MongoDB

You need MongoDB to store the data for products, users, orders, and other e-commerce information. You can install MongoDB locally or use a cloud-based MongoDB service like MongoDB Atlas.

- **Download MongoDB:**

- [Download MongoDB Community Edition](#)

- **Installation Instructions:**

- [MongoDB installation guide](#)

---

### 3. Express.js

Express.js is a web application framework for Node.js. It will handle server-side routing, middleware, and API development.

- **Install Express.js:** In your terminal, run the following command to install

- Express:

- `bash`

- Copy code

- `npm install express`

---

### 4. React.js

React.js is a powerful JavaScript library for building interactive user interfaces, and it's perfect for creating dynamic, responsive front-end applications.

- **Follow this guide to install React:**

- [React Documentation: Creating a New React App](#)

- To create a new React app, you can use **Create React App**:

- `bash`

- Copy code

- `npx create-react-app trends-ecommerce`

- `cd trends-ecommerce`

npm start

---

## 5. HTML, CSS, and JavaScript

Basic knowledge of HTML, CSS, and JavaScript is essential to structure, style, and add interactivity to the user interface of your e-commerce app.

---

## 6. Database Connectivity (MongoDB + Node.js)

To connect MongoDB with your Node.js server, you can use **Mongoose**, an Object Data Modeling (ODM) library for MongoDB and Node.js.

- **Install Mongoose:**

bash

Copy code

npm install mongoose

For more details on connecting MongoDB to Node.js, check out this guide:  
Link: [Node.js + Mongoose + MongoDB](#)

---

## 7. Front-End Framework (React)

Since your project will use React.js for the front-end, you'll be building the user-facing part of the application, including product listings, shopping carts, user profiles, and the admin dashboard using React components.

---

## 8. Version Control with Git

Use Git for version control to manage your project's codebase, track changes, and collaborate with other developers. GitHub or Bitbucket can be used to host your repository.

- **Download Git:**

[Git download page](#)

---

## 9. Development Environment (IDE/Code Editor)

Choose a code editor that you're comfortable with. Here are a few popular options:

- **Visual Studio Code:**

[Download Visual Studio Code](#)

- **Sublime Text:**

[Download Sublime Text](#)

- **WebStorm:**

[Download WebStorm](#)

---

### Set Up the Book a Doctor App

To get started with the Book a Doctor app, follow these steps:

- Clone the Repository
- Open your terminal or command prompt.
- Navigate to the directory where you want to store the Book a Doctor app.
- Clone the repository by executing the following command:
- `bash`
- Copy code
- `git clone https://github.com/your-repository-link.git`
- Alternatively, if the project files are stored in Google Drive, download them from the following link:
- Google Drive Link for Book a Doctor

---

### Install Dependencies

1. Navigate into the cloned directory:

`bash`

cd Book-a-Doctor-App-MERN

Install the required dependencies:

- bash
- Copy code
- npm install

### **Start the Development Server**

To start the development server, run the following command:

- bash
- Copy code
- npm run dev

Or you can run:

- bash
- Copy code
- npm run start

By default, the app will be accessible at: <http://localhost:3000>

You can change the port by modifying the .env file if needed.

### **Access the App**

Open your browser and navigate to <http://localhost:3000>.

You should see the homepage of the Book a Doctor app.

---

## **11. Customization & Development**

Once the Book a Doctor app is up and running, you can proceed with the following:

- **Customizing the UI:**

Modify the React components to reflect your design and theme (e.g., doctor listings, appointment booking forms, and dashboard pages).

- **Building the Backend API:**

Use Express.js and MongoDB to implement features such as:

- **Doctor Management:**

Add, update, or remove doctor profiles and expertise.

- **Appointment Scheduling:**

Handle appointment booking and management.

---

- **User Authentication:**

Enable user login and registration using JWT or other authentication methods.

- **Testing:**

Test your app locally to ensure all functionalities, including doctor browsing, appointment booking, and notifications, work smoothly.

---

## 12. Further Customizations

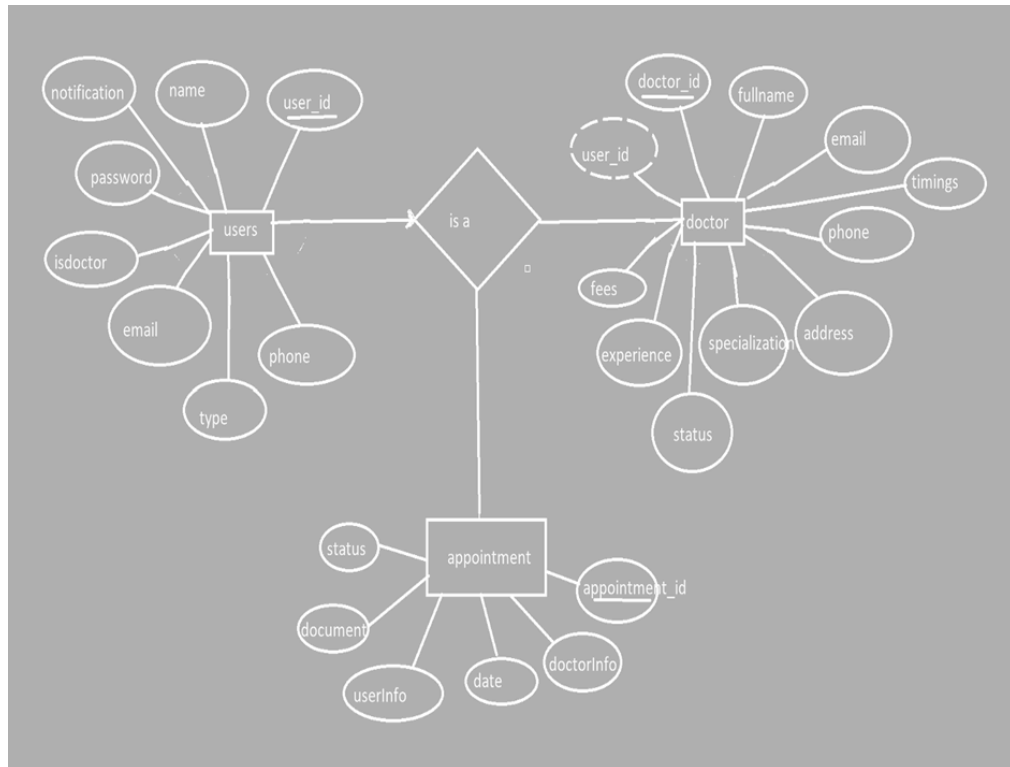
- **Customize the Front-End:**

- Adapt the UI design to match your branding and user requirements, including pages for browsing doctors, booking appointments, and viewing notifications.

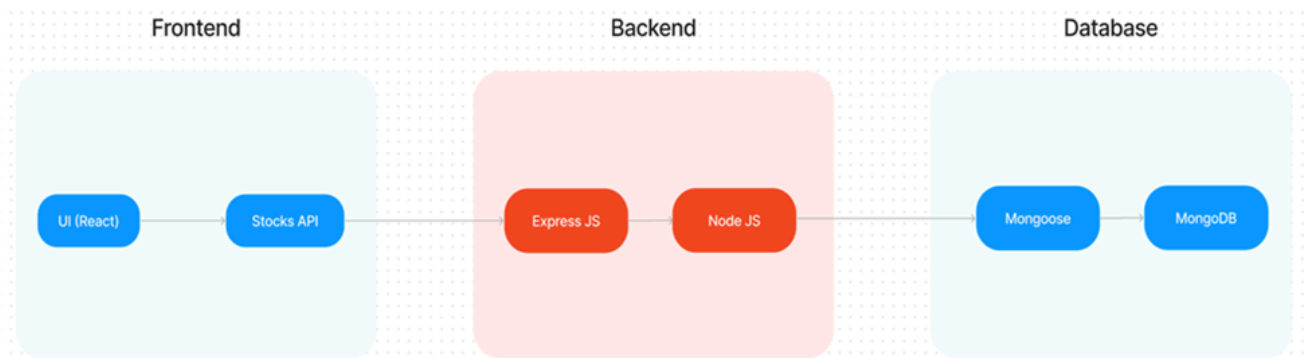
- **Additional Features:**

- **Implement User Authentication:** Secure login and registration for patients and doctors using JWT.
- **Add a Notification System:** Notify users when their appointment time is approaching or if there are changes.
- **Create an Appointments Page:** Show all booked appointments with details like date, time, and doctor's name.
- **Build an Admin Dashboard:** Allow admins to approve new doctor registrations and manage system data.

## ER-DIAGRAM



The Database section represents the database that stores collections for Users, Admin, Cart, Orders and products.





## **Book a Doctor Platform ER-Diagram Overview**

The Book a Doctor ER-Diagram (Entity-Relationship Diagram) illustrates the various entities within the platform and their relationships. It focuses on the interaction between Users (Patients), Doctors, Appointments, and Admins.

### **Here's a detailed breakdown:**

---

#### **Entities and Their Relationships**

##### **USER (Patient)**

**Description:** Represents individuals registered on the platform to book doctor appointments.

##### **Key Fields:**

User ID

Name

Email

Password

Phone Number

Address

Medical Records (optional)

##### **DOCTOR**

- Description: Represents healthcare providers available for appointments.
- Key Fields:
  - Doctor ID
  - Name
  - Specialty
  - Availability (dates and times)
  - Contact Information
  - Experience

## **APPOINTMENT**

Description: Represents the interaction between Users and Doctors for a scheduled session.

### **Key Fields:**

Appointment ID

User ID (linked to the Patient)

Doctor ID (linked to the Doctor)

Appointment Date and Time

Status (Pending, Scheduled, Completed, Cancelled)

## **ADMIN**

Description: Represents the administrative role responsible for managing platform operations.

### **Key Fields:**

- Admin ID
- Name
- Email
- Role

## **NOTIFICATIONS**

- Description: Represents system-generated alerts or reminders for users about their scheduled appointments.

### **Key Fields:**

- Notification ID
- User ID
- Appointment ID (optional, if linked to a specific appointment)
- Message Content
- Timestamp

## **Key Features of the Book a Doctor Platform**

### **Comprehensive Doctor Directory**

The platform provides a detailed list of available doctors from various specialties, including their profiles, qualifications, availability, and experience.

Users can easily browse through the directory and filter doctors based on location, specialty, or available time slots.

---

This feature helps users find the best doctor suited to their medical needs.

### **Book Appointment Button**

Each doctor's profile includes a "Book Appointment" button.

Users can click this button to start the appointment booking process quickly.

### **Appointment Details Page**

After selecting a doctor and clicking "Book Appointment", users are directed to the Appointment Details Page.

Here, users can provide the following information:

Appointment Date and Time

Medical Concerns or Symptoms

Upload Medical Records (if required)

The booking process is optimized to provide a fast and user-friendly experience, ensuring minimal hassle while reserving appointments.

### **Appointment Confirmation and Notifications**

Once an appointment is successfully booked, users receive a confirmation notification with all the details.

The Appointment Details Page displays:

Doctor's Name and Specialty

Appointment Date and Time

Uploaded Documents (if any)

Location Details (if an in-person visit is required)

Users also get reminders as the appointment date approaches.

## **User Experience for the Book a Doctor Platform**

The Book a Doctor platform is designed to simplify and enhance the healthcare appointment process by providing:

### **Comprehensive Doctor Listings**

Users can access a wide range of doctors across various specialties, ensuring that all their healthcare needs are met on a single platform.

### **Intuitive Navigation**

The user interface is clean and easy to navigate, allowing patients to search, browse, and filter doctors based on their location, specialty, or availability.

### **Seamless Appointment Booking**

The appointment booking process is fast and straightforward. Users can view doctor availability in real-time and schedule an appointment with just a few clicks.

### **Secure and Confidential Data Handling**

The platform ensures that all personal and medical data is stored securely, maintaining user privacy and trust.

### **Efficient Appointment Management**

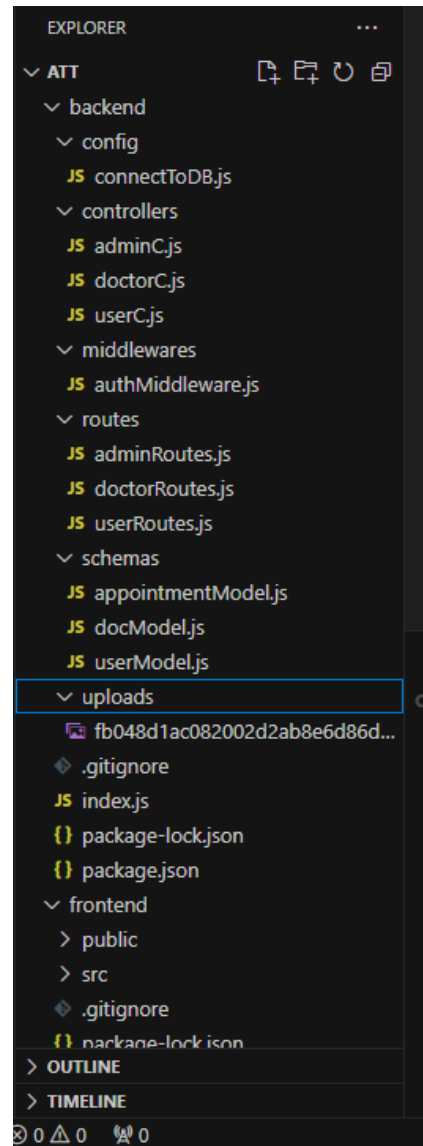
Users can manage their appointments in one place, with features to reschedule or cancel if needed.

Notifications and reminders are sent to users to ensure they don't miss their appointments.

### **Enhanced Doctor Tools**

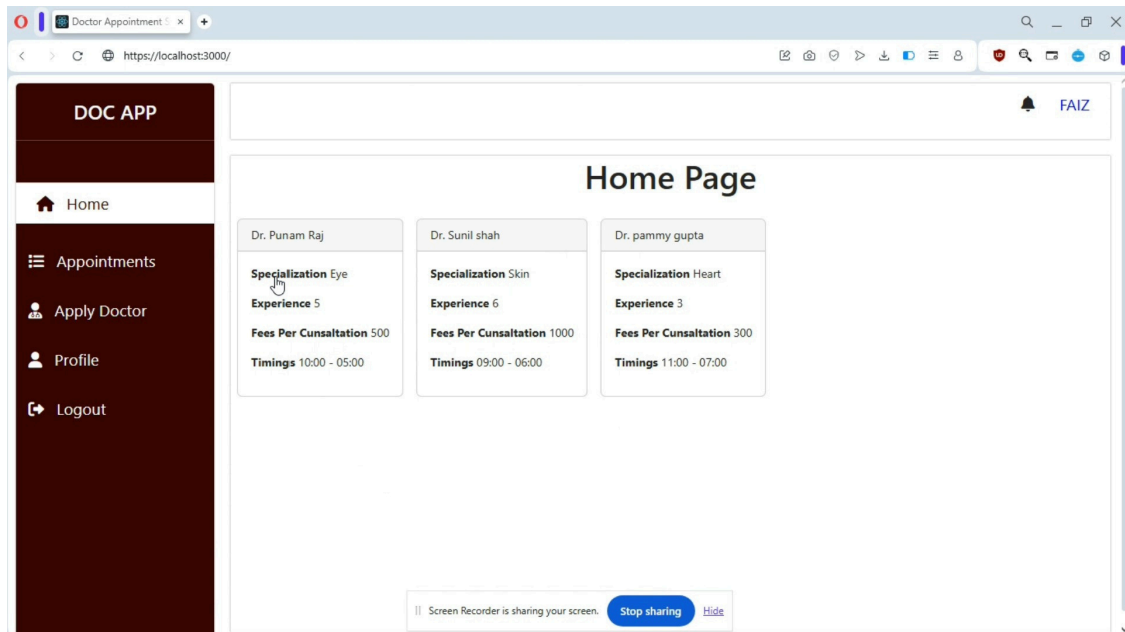
Doctors have access to tools that help them manage their schedules, approve appointments, and update patient records easily.

## PROJECT STRUCTURE:



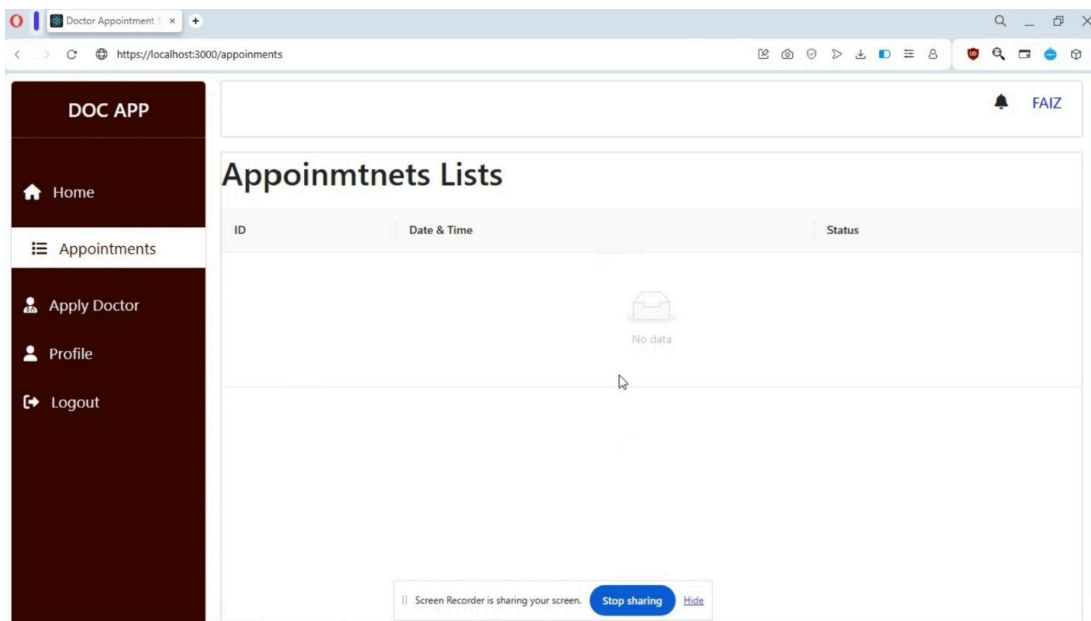
## OUTPUT:

### HOME PAGE

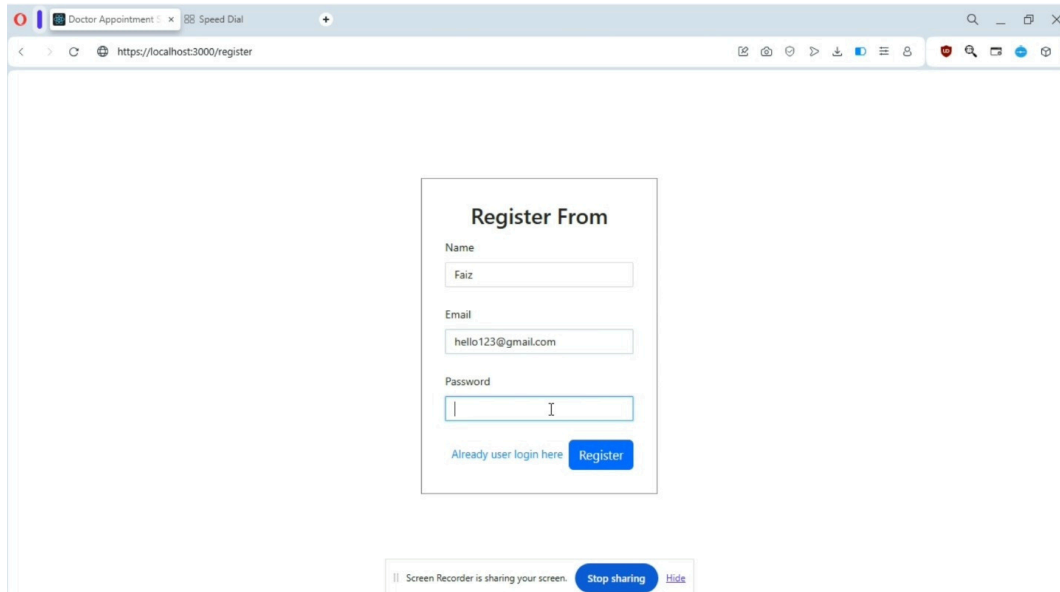


## OUTPUT:

### APPOINTMENTS PAGE



## REGISTER / LOGIN PAGE



The screenshot shows a web browser window with the title "Doctor Appointment" and a single tab. The address bar displays "https://localhost:3000/register". The main content area features a "Register From" form with the following fields: "Name" (containing "Faiz"), "Email" (containing "hello123@gmail.com"), and "Password" (empty). Below the password field is a link "Already user login here" and a blue "Register" button. At the bottom of the browser window, a notification bar states "Screen Recorder is sharing your screen." with "Stop sharing" and "Hide" buttons.

Doctor Appointment x Speed Dial

https://localhost:3000/register

**Register From**

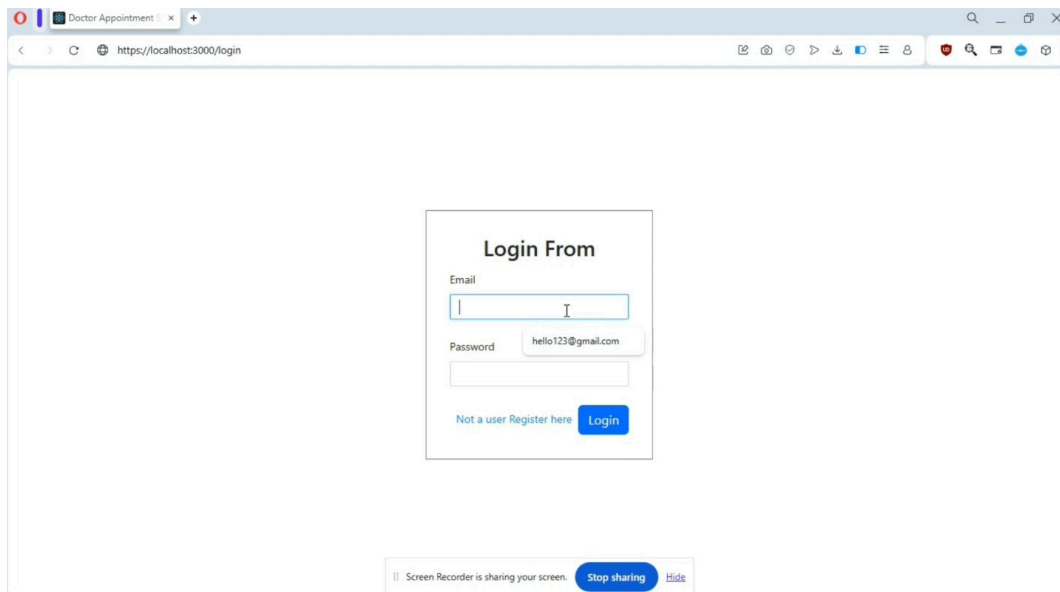
Name  
Faiz

Email  
hello123@gmail.com

Password  
| I

[Already user login here](#) [Register](#)

Screen Recorder is sharing your screen. [Stop sharing](#) [Hide](#)



The screenshot shows a web browser window with the title "Doctor Appointment" and a single tab. The address bar displays "https://localhost:3000/login". The main content area features a "Login From" form with the following fields: "Email" (empty) and "Password" (containing "hello123@gmail.com"). Below the password field is a link "Not a user Register here" and a blue "Login" button. At the bottom of the browser window, a notification bar states "Screen Recorder is sharing your screen." with "Stop sharing" and "Hide" buttons.

Doctor Appointment x

https://localhost:3000/login

**Login From**

Email  
|

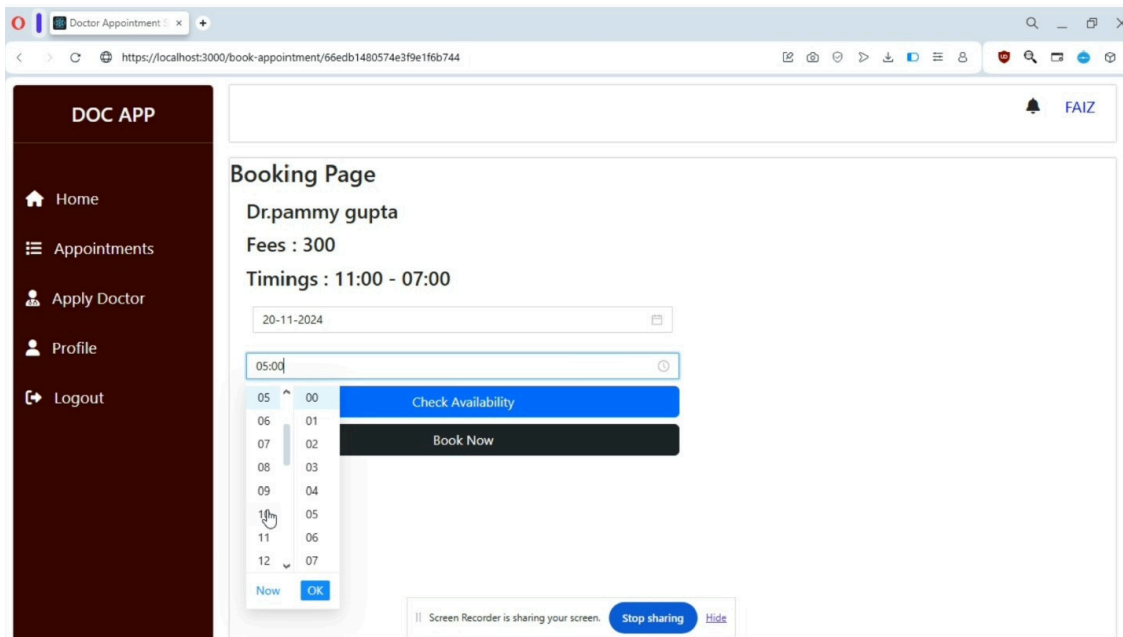
Password  
hello123@gmail.com

[Not a user Register here](#) [Login](#)

Screen Recorder is sharing your screen. [Stop sharing](#) [Hide](#)



# CALENDER BOOKING PAGE



## CONCLUSION

The Book a Doctor platform revolutionizes the healthcare appointment process by bridging the gap between patients and doctors in a seamless, efficient, and secure manner. By providing an intuitive user interface, comprehensive doctor listings, and real-time appointment scheduling, it ensures convenience for patients and streamlined management for doctors.

This platform not only simplifies the process of booking appointments but also enhances trust through secure data handling and effective communication. With its user-friendly design and robust backend, the system ensures an optimal experience for all users.

The Book a Doctor platform is a significant step forward in digital healthcare, making quality medical services more accessible and fostering a better connection between patients and healthcare providers. It's a solution designed to save time, reduce hassle, and prioritize user satisfaction, setting a new standard in healthcare technology.