Project Title:seamless online booking
Team Members:

Team Leader :**Veerabhadra Shanmukha Srinivasa Reddy**
Team member : **Naladhi Malleswari**
Team member : **Shaik Vasim Basha**
Team member :**Kamepalli Manoj**

Project Overview

Purpose:

Seamless booking in healthcare allows patients to easily schedule appointments online or through mobile apps, reducing wait times and administrative burdens. It improves patient experience, increases accessibility, and enables efficient resource allocation. This streamlined process also reduces no-show rates and enhances patient engagement.

Features:

- Online Booking

- Real-Time Availability

- Automated Reminders

- Easy Rescheduling and Cancellation

- Integration with EMR/EHR

ER DIAGRAM

Users:
_id: (MongoDB creates by unique default)
name
email
notification
password
isdoctor
type
phone
Doctor:
userID: (can be act as foreign key)
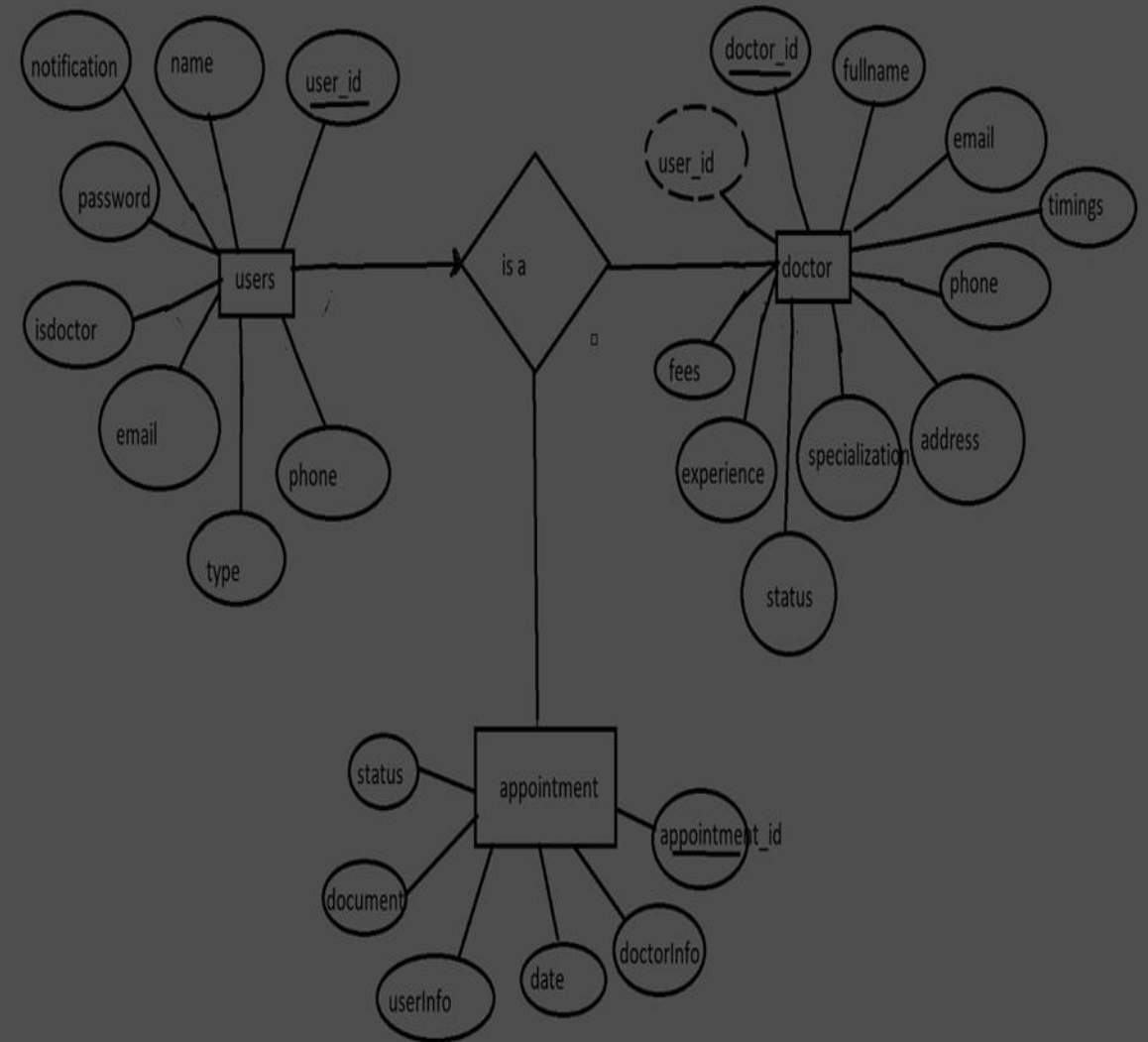_id: (MongoDB creates by unique default)
fullname
email
timings
phone
address
specialization
status
experience

PRE-REQUISITES
Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.
Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.
Download: https://nodejs.org/en/download/
Installation instructions: https://nodejs.org/en/download/package-manager/

Express.js:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating.jg robust APIs and web applications, offering features like routing, middleware support, and modular architecture.
Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.
Installation: Open your command prompt or terminal and run the following command:

npm install express

MongoDB:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.
Install React.js, a JavaScript library for building user interfaces.
Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.js. To Connect the Database with Node JS go through the below provided link:
https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

Front-end Framework: Utilize Reactjs to build the user-facing part of the application, including entering complaints, status of the complaints, and user interfaces for the admin dashboard.
For making better UI we have also used some libraries like material UI and boostrap.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process.
Platforms like GitHub or Bitbucket can host your repository.
Git: Download and installation instructions can be found at: https://git-scm.com/downloads

Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from https://code.visualstudio.com/download

To run the existing Video Conference App project downloaded from GitHub:
Follow below steps:
Clone the Repository:
Open your terminal or command prompt.
Navigate to the directory where you want to store the e-commerce app.
Execute the following command to clone the repository:
git clone: https://github.com/awdhesh-student/complaint-registery.git

Install Dependencies:

• Navigate into the cloned repository directory:
cd complaint-registery
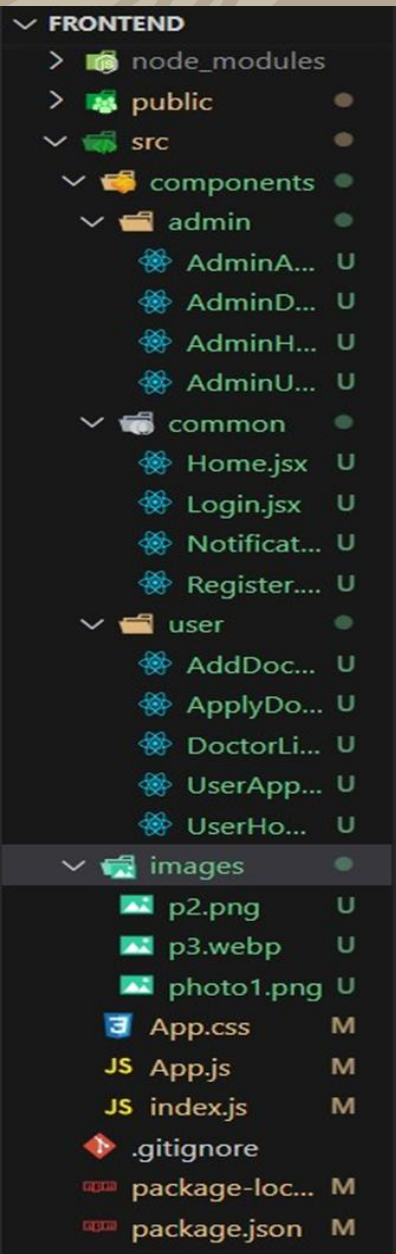• Install the required dependencies by running the following commands:
cd frontend
npm install
cd ../backend
npm install
Start the Development Server:
• The online complaint registration and management app will be accessible at http://localhost:3000

PROJECT 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

BACKEND
- config
  - connectToDB.js
- controllers
  - adminC.js
  - doctorC.js
  - userC.js
- middlewares
  - authMiddleware.js
- node_modules
- routes
  - adminRoutes.js
  - doctorRoutes.js
  - userRoutes.js
- schemas
  - appointmentMo...
  - docModel.js
  - userModel.js
- uploads
  - a77e910e017f4b...
  - af8555e0fb38fe...
- .env
- .gitignore
- index.js
- package-lock.json
- package.json

Application Flow: The project has 2 type of user – Customer and Doctor and other will be Admin which takes care to all the user. The roles and responsibilities of these two types of users can be inferred from the API endpoints defined in the code. Here is a summary:

Customer/Ordinary:

1. Create an account and log in to the system using their email and password.
2. They will be shown automatically all the doctors in their dashboard.
3. After clicking on the Book Now, a form will generate in which date of appointment and documents need to send.
4. They can sees the status of their appointment and can get a notification if the appointment is schedule or not.
5. The user can also cancel it`s booking in booking history page and can change the status of booking.

Admin:

1. Manage and monitor the overall operation of the appointment and the type of users and doctors to the application.
2. He monitors the applicant of all doctors and approve them and then doctors are registered in the app.
3. Implement and enforce platform policies, terms of service, and privacy regulations.

Doctor:

Gets the approval from the admin for his doctor account.
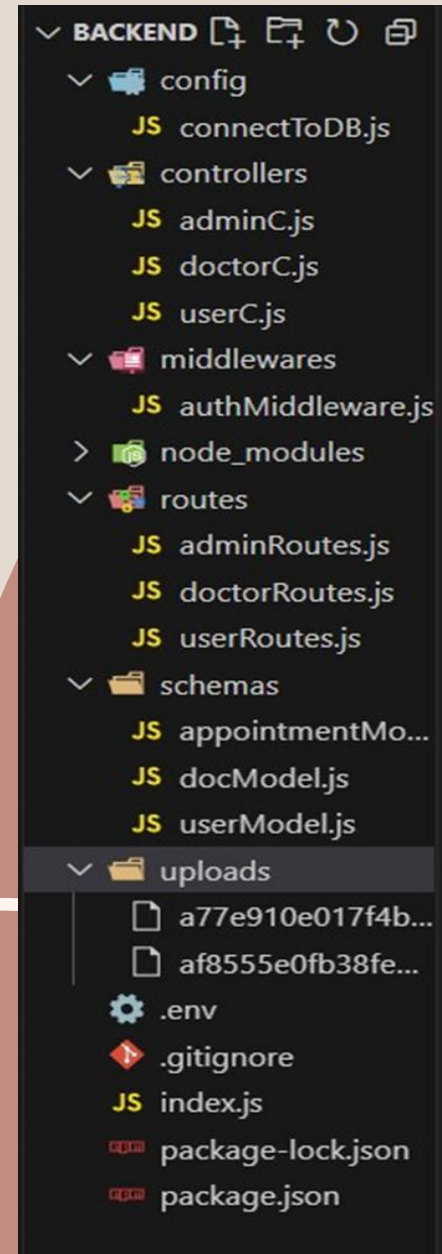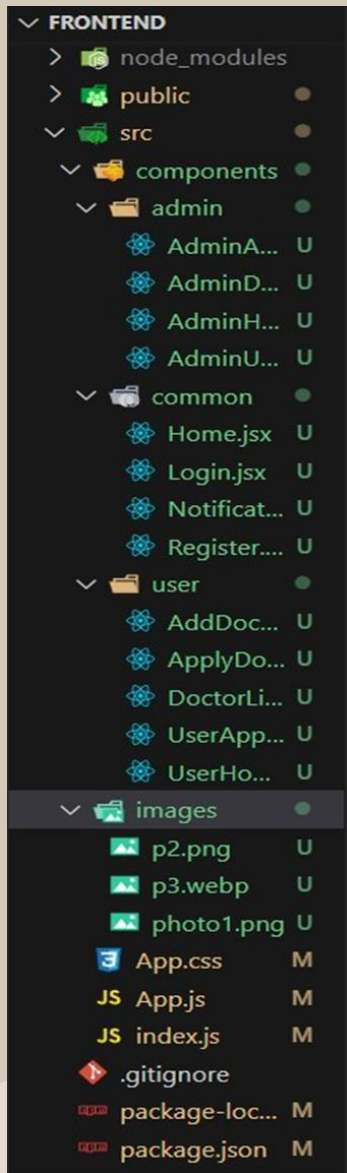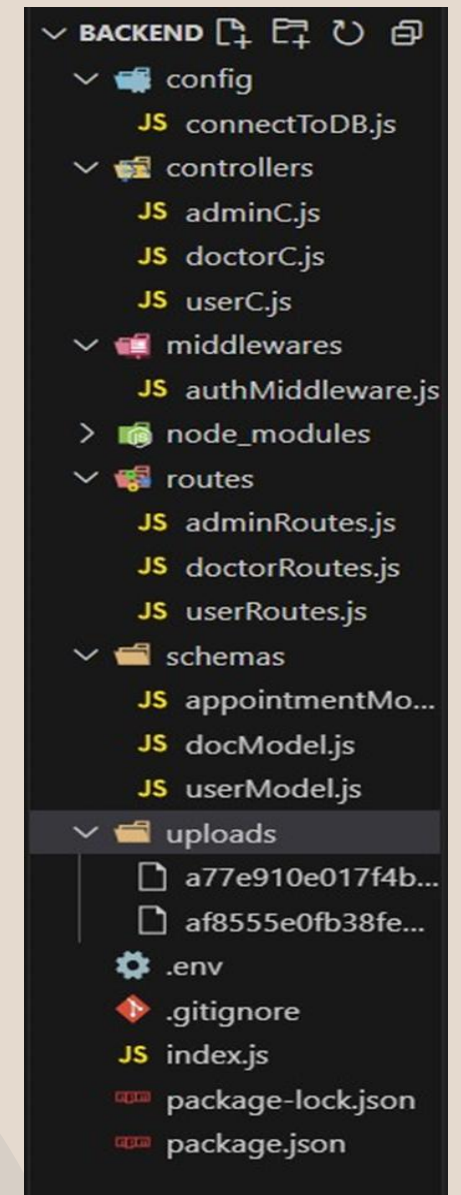
Folder setup

•Folder setup:
1.Create frontend and
2.Backend folders
Open the backend folder to install necessary tools

For backend, we use:

•cors
•bcryptjs
•express
•dotenv
•mongoose
•Multer
•Nodemon
•jsonwebtoken

Setup express server
Create index.js file in the server (backend folder).
define port number, mongodb connection string and JWT key in env file to access it.
Configure the server by adding cors, body-parser.

Add authentication: for this,
You need to make a middleware folder and in that make an authMiddleware.js file for the authentication of the projects and can use in.

Configure MongoDB

•**Configure MongoDB**
1.Import mongoose.
2.Add database connection from config.js file present in config folder
3.Create a model folder to store all the DB schemas like renter, owner and booking, properties schemas.

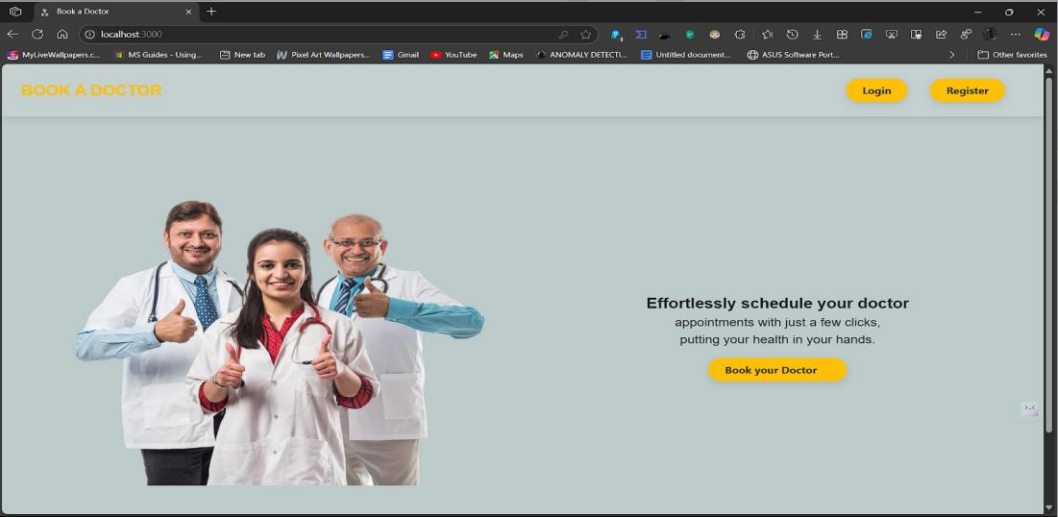## Installation of required tools
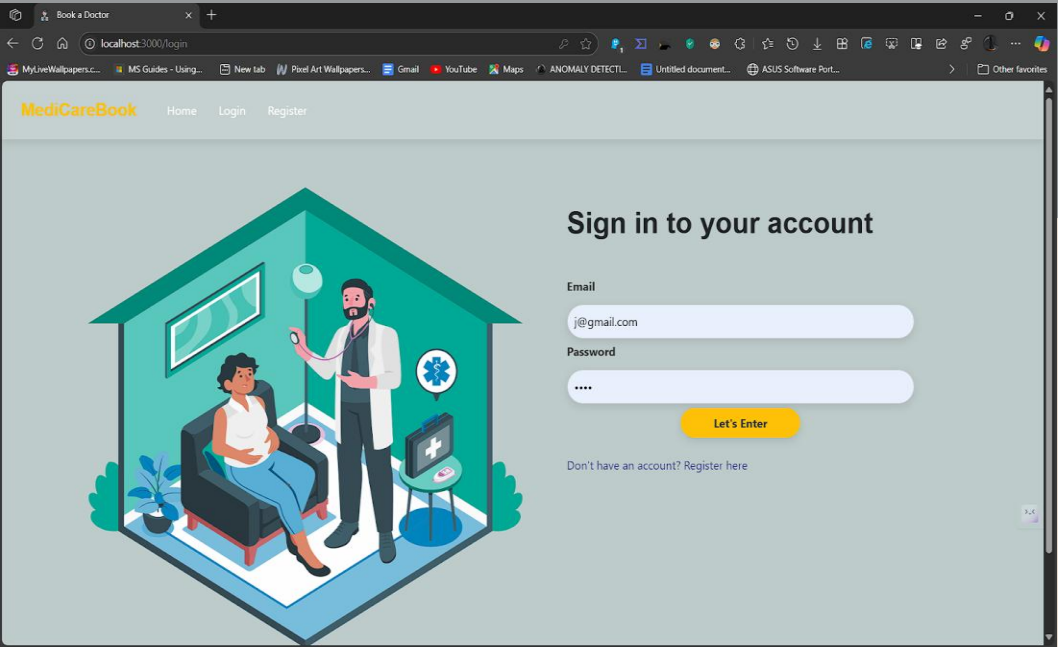
•**Installation of required tools:**
•For frontend, we use:
1.React
2.Bootstrap
3.Material UI
4.Axios
5.Antd
6.mdb-react-ui-kit
7.react-bootstrap

Landing page
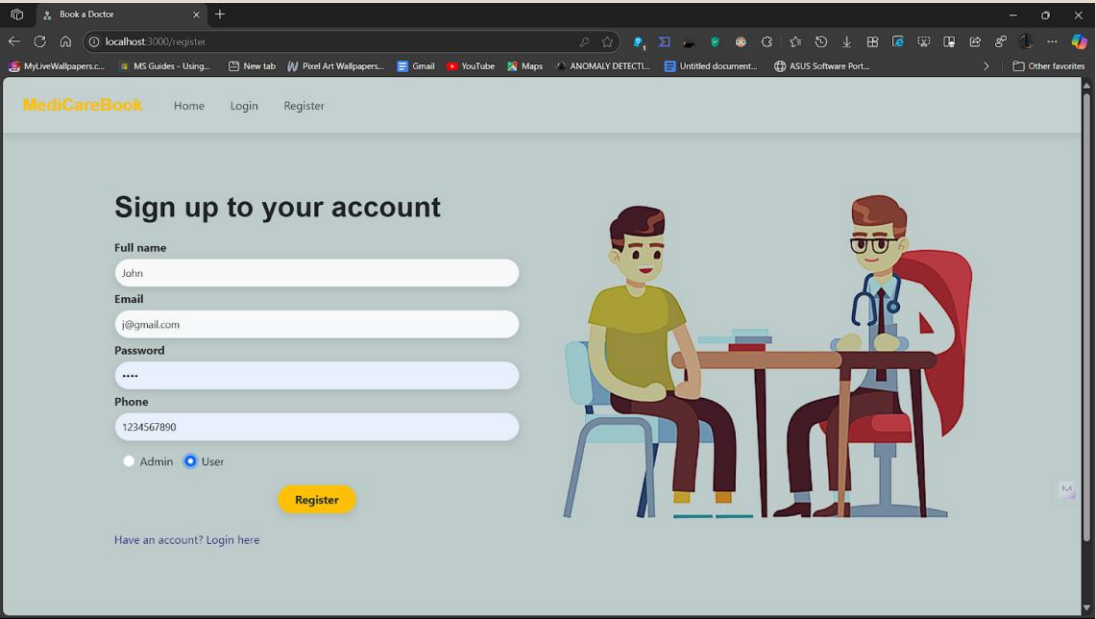

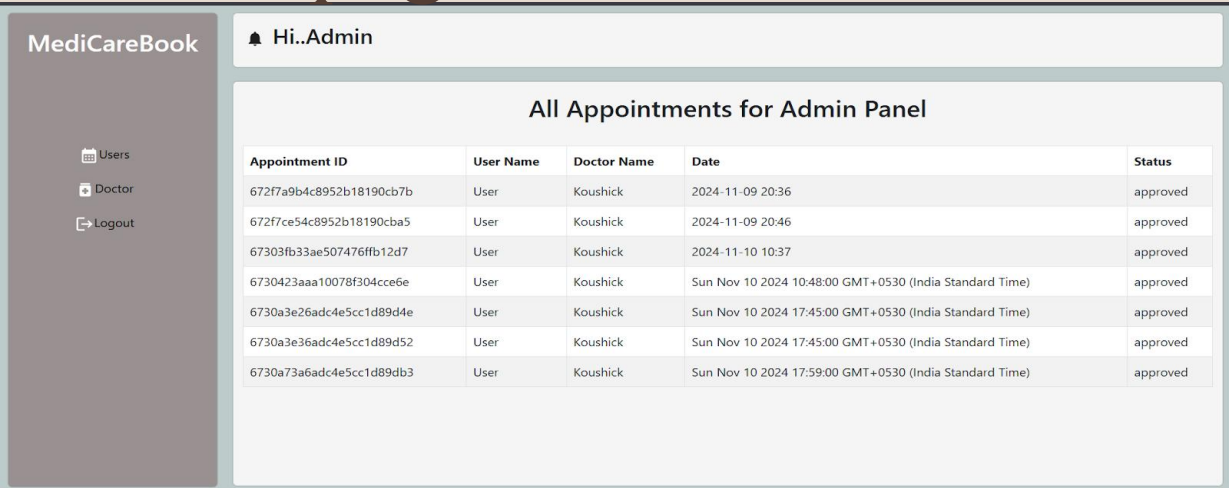# Register page

Login page

Admin page:

My History Page:

My Orders

Order ID: 6614b085c30b51d3c700f20a
Name: syed arshad
Phone: 9505221870
Date: 2024-04-09T03:05:41.563Z
Price: 400
Status: Pending
Payment Method: credit

Place Order Page:-

My History

Order ID: 6614a8ddc30b51d3c700f1b4
Name: syed arshad
Phone: 9505221870
Date: 2024-04-09T02:36:47.498Z
Price: 400
Status: Delivered
Payment Method: debit

Users Page:-

Order Details

First Name:
Enter your first name

Last Name:
Enter your last name

Phone:
Enter your phone number

Quantity:
Enter the quantity

Address:
Enter your address

Payment Method:
Cash on Delivery (COD)

Submit

localhost:3000/login

Add Product page:-

Add Product

Product Name   **Rating**   **Price**
Enter product name   Enter product rating   Enter product price

Image URL   **Category**   **Count in Stock**
Enter image URL   Select Category   Enter count in stock

Description
Enter product description

Add Product

Doctor dashboard

🔔 User

✅ Doctor Registration request sent successfully

Apply for Doctor

📅 Appointments

➕ Apply doctor

🔓 Logout

Personal Details:

* Full Name: SHIVA   * Phone: 91755584121   * Email: user@gamil.com

* Address: chennnai

Professional Details:

* Specialization: Blood   * Experience: 2   * Fees: 5001

* Timings: 06:00 → 12:00 🕐

Submit

User dashboard

🔔 User

Home

📅 Appointments

➕ Apply doctor

🔓 Logout

Dr. Koushick

Phone: **09176478438**

Address: **chennai**

Specialization: **ENT**

Experience: **5 Yrs**

Fees: **1000**

Timing: **07:00 : 12:00**

Book Now

Dr. Karthick

Phone: **09176478438**

Address: **Kerala**

Specialization: **Bone**

Experience: **10 Yrs**

Fees: **500**

Timing: **:**

Book Now

Thank You