Full Stack Development with MERNs

Project Documentation

**1. Introduction**

• Project Title: [**DocSpot: Seamless Appointment Booking for Health**]

• Team Members: List team members

Team ID : LTVIP2025TMID56531

Team Size : 4

Team Leader : Cheeli Rupa Pavani

Team member : Ande Lokesh

Team member : Baki Bindhu Sri

Team member : Anumakonda Swapna Naga Priya

**2. Project Overview**

• Purpose: Briefly describe the purpose and goals of the project.

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.

• Features: Highlight key features and functionalities.

**Scenario-based Case Study:**

Scenario: Booking an Appointment with a Doctor

**User Registration:** John, who needs to see a doctor for a routine check-up, visits the Book a Doctor app and signs up as a Customer. He provides his email and creates a password.

**Browsing Doctors:** Upon logging in, John is presented with a dashboard displaying a list of available doctors and healthcare providers.

He filters the list based on his preferences, such as specialty, location, or availability.

**Booking an Appointment**: John finds a suitable doctor and clicks on "Book Now." A form appears where he selects the desired appointment date and uploads any necessary documents, such as medical records or insurance information.

After submitting the form, John receives a confirmation message indicating that his appointment request has been received.

**Appointment Confirmation**: The doctor reviews John's appointment request and availability. Once confirmed, the appointment status changes to "scheduled."

John receives a notification confirming his appointment and providing details such as the date, time, and location.

**Appointment Management:** As the appointment approaches, John can view and manage his upcoming appointments in the booking history section of his dashboard.

He has the option to cancel or reschedule appointments if needed and can update the status accordingly.

**Admin Approval (Background Process):** In the background, the admin reviews new doctor registrations and approves legitimate applicants.

Approved doctors are then registered in the app and can start managing their appointments.

**Platform Governance:** The admin oversees the overall operation of the appointment booking system and ensures compliance with platform policies, terms of service, and privacy regulations.

The admin addresses any issues or disputes to maintain a smooth user experience.

**Doctor's Appointment Management:** Dr. Smith, an approved doctor on the platform, logs into his account and manages his appointments.

He views his schedule, confirms or reschedules appointments, and updates appointment statuses based on patient interactions.

**Appointment Consultation:** On the day of the appointment, John visits the doctor's office for his check-up.

Dr. Smith provides medical care and advice during the consultation, fulfilling John's healthcare needs.

**Post-Appointment Follow-up:** After the appointment, Dr. Smith updates John's medical records and may prescribe medication or recommend further treatment if necessary.

**3. Architecture**

• Frontend: Describe the frontend architecture using React.

• Backend: Outline the backend architecture using Node.js and Express.js.

• Database: Detail the database schema and interactions with MongoDB.

. Open the backend folder to install necessary tools

For backend, we use:

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

**4. Setup Instructions**

• Prerequisites: List software dependencies (e.g., Node.js, MongoDB).

• Installation: Step-by-step guide to clone, install dependencies, and set up the

environment variables.

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, and 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/>

**npm init**

* **Express.js:**

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating 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.

Set up a MongoDB database to store your application's data.

Download: <https://www.mongodb.com/try/download/community>

Installation instructions: <https://docs.mongodb.com/manual/installation/>

* **Moment.js:**

Momentjs is a JavaScript package that makes it simple to parse, validate, manipulate, and display date/time in JavaScript. Moment. js allows you to display dates in a human-readable format based on your location. Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: [https://momentjs.com/](https://momentjs.com/%20)

* **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>

* **Antd:**

Ant Design is a React. js UI library that contains easy-to-use components that are useful for building interactive user interfaces. It is very easy to use as well as integrate. It is one of the smart options to design web applications using react.

Follow the installation guide: <https://ant.design/docs/react/introduce>

* **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. 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 booking room, status of the booking, and user interfaces for the admin dashboard.

For making better UI we have also used some libraries like material UI and boostrap.

Install Dependencies:

• Navigate into the cloned repository directory:

cd book-a-doctor

• Install the required dependencies by running the following commands:

cd frontend

npm install

cd ../backend

npm install

Start the Development Server:

• To start the development server, execute the following command:

npm start

• The book a doctor app will be accessible at [http://localhost:3000](http://localhost:3000/)

You have successfully installed and set up the online complaint registration and management app on your local machine. You can now proceed with further customization, development, and testing.

**5. Folder Structure**

• Client: Describe the structure of the React frontend.

• Server: Explain the organization of the Node.js backend.

**6. Running the Application**

• Provide commands to start the frontend and backend servers locally.

o Frontend: npm start in the client directory.

o Backend: npm start in the server directory.

**7. API Documentation**

• Document all endpoints exposed by the backend.

• Include request methods, parameters, and example responses.

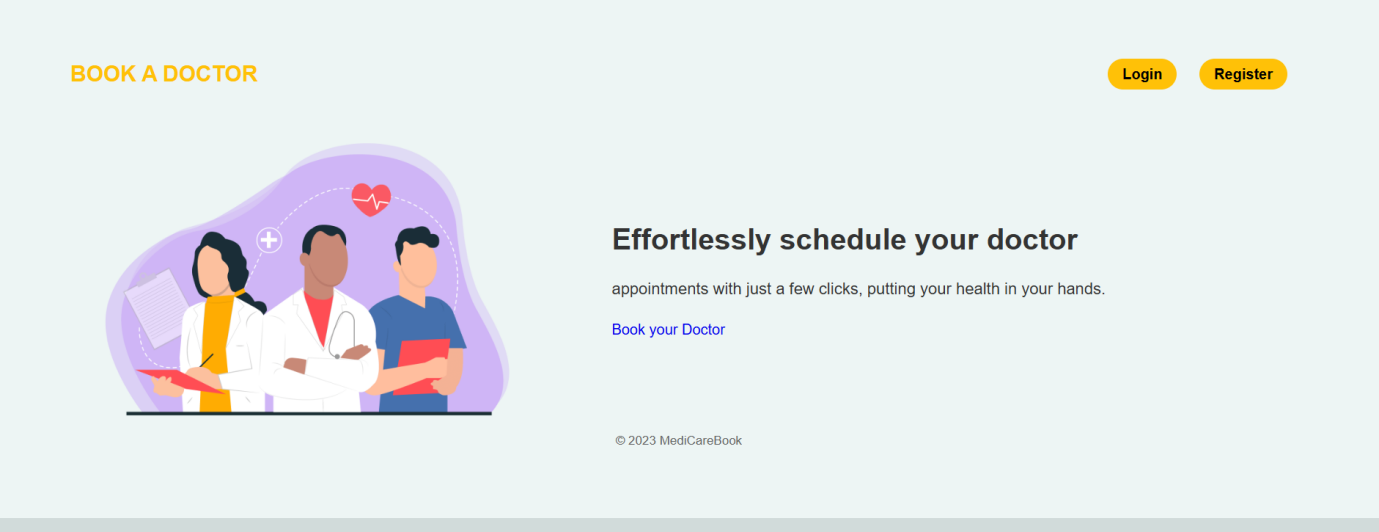
**8. Authentication**

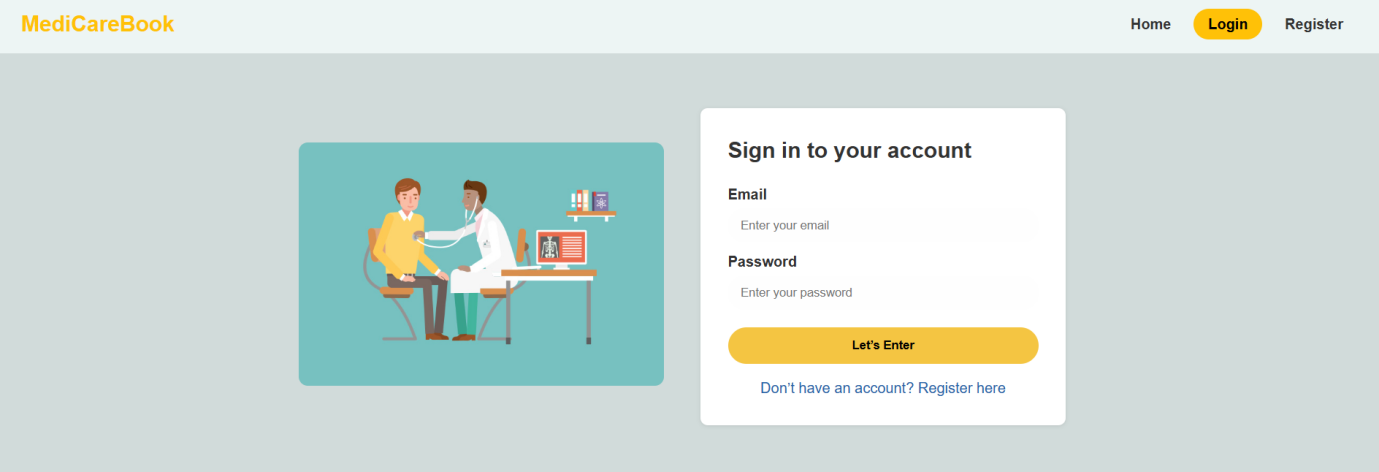
• Explain how authentication and authorization are handled in the project.

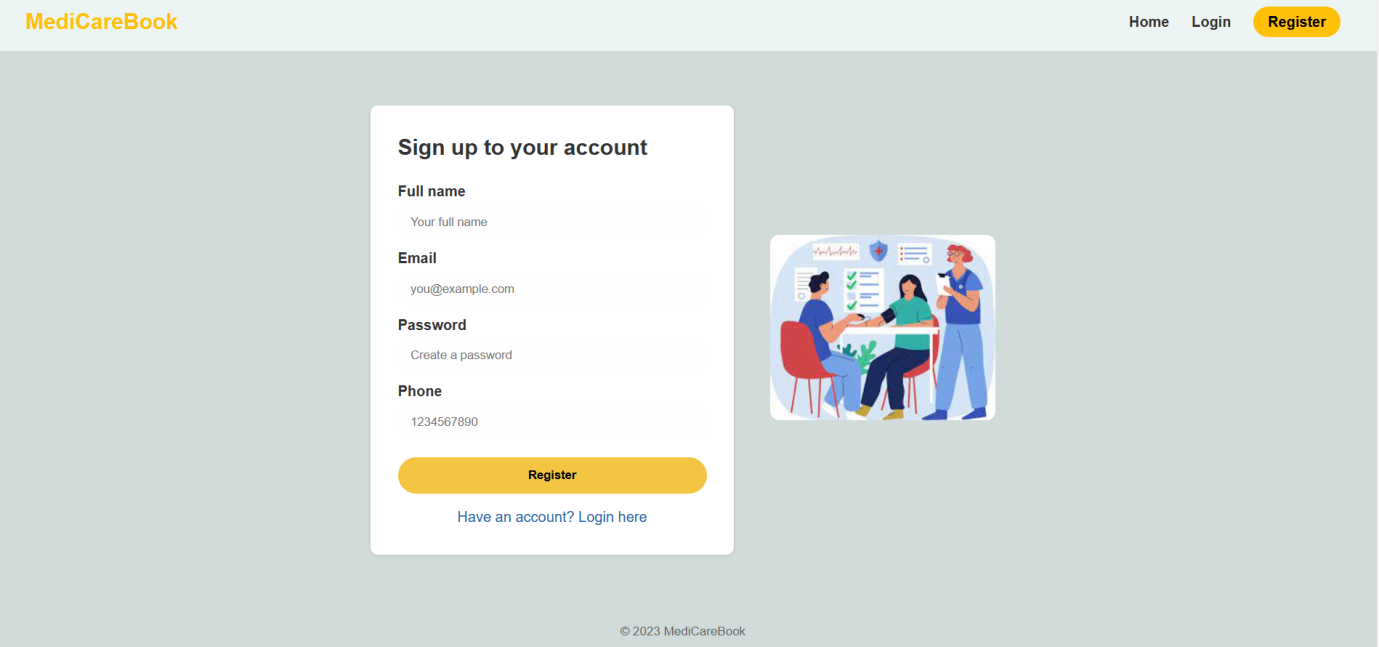
• Include details about tokens, sessions, or any other methods used.

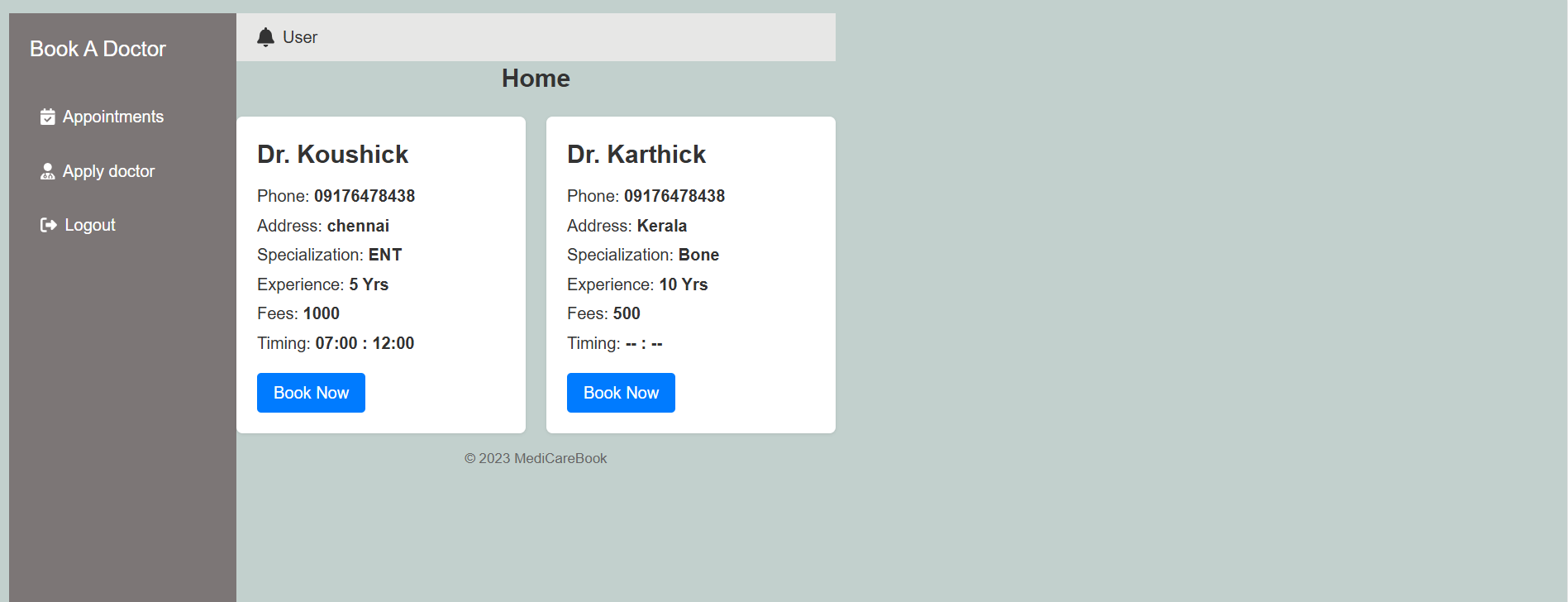
**9. User Interface**

• Provide screenshots or GIFs showcasing different UI features.



****

****

****

**10. Testing**

• Describe the testing strategy and tools used.

**Tested successfully.**

**11. Known Issues**

• Document any known bugs or issues that users or developers should be aware of.