DocSpot: Doctor Appointment Booking System

Project Overview

DocSpot is a full-stack web application that allows patients to seamlessly book appointments with doctors. The application supports multiple roles-User, Doctor, and Admin-each with dedicated dashboards and functionality. It simplifies appointment booking and management with an intuitive interface and real-time scheduling.

DocSpot: Doctor Appointment Booking System

Technology Stack

Frontend:

- React.js

- Axios

- React Router

- Bootstrap / Material UI

Backend:

- Node.js

- Express.js

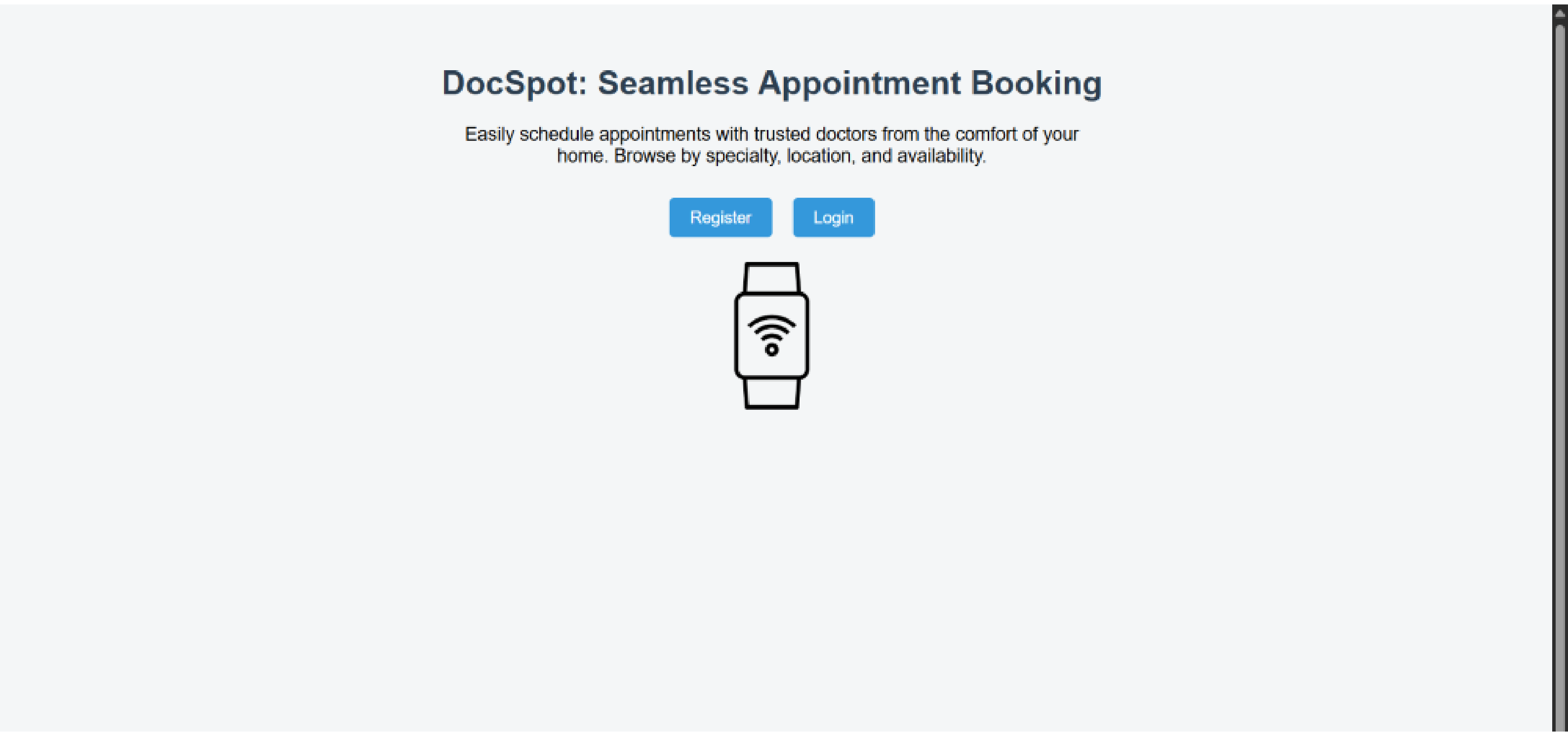
- MongoDB

- Mongoose

Other Tools:

- JWT for authentication

- bcrypt for password hashing.



DocSpot: Seamless Appointment Booking

Easily schedule appointments with trusted doctors from the comfort of your home. Browse by specialty, location, and availability.

DocSpot: Doctor Appointment Booking System

System Architecture

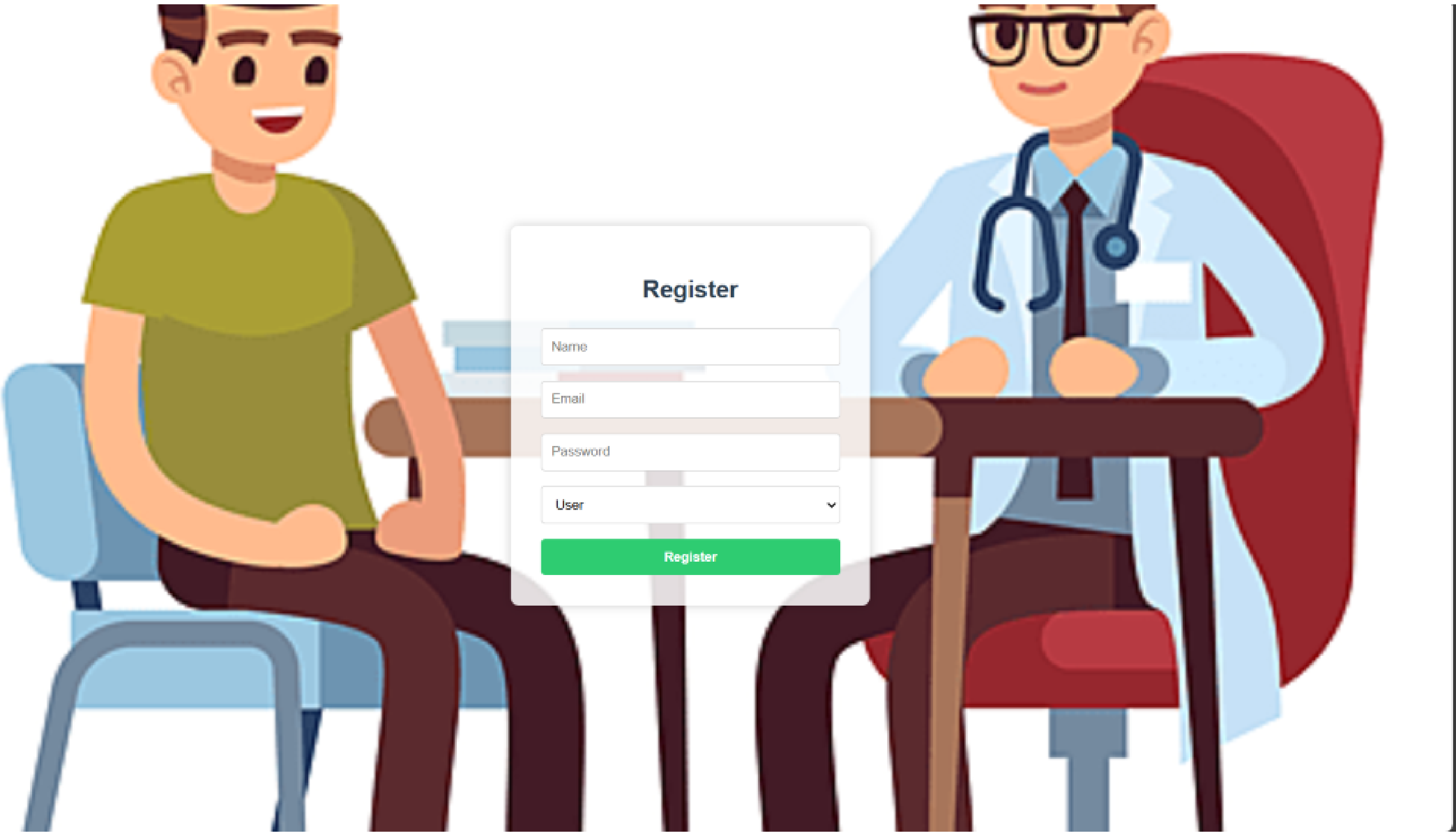
The application uses a client-server architecture:

- Frontend (React) communicates with the backend (Express.js) using REST APIs.
- MongoDB handles data persistence.
- Users authenticate using JWT tokens.
- Role-based access control ensures secure routing and functionality.

DocSpot: Doctor Appointment Booking System

Project Structure

```
doctor-appointment-app/
--- backend/
-   --- config/
-   --- controllers/
-   --- models/
-   --- routes/
-   --- server.js
--- frontend/
-   --- public/
-   --- src/
-   -   --- pages/
-   -   --- components/
-   -   --- App.js
```



DocSpot: Doctor Appointment Booking System

User Role Features

Users:

- Register/Login

- Browse doctors

- Book appointments

- View appointment history

Doctors:

- Login

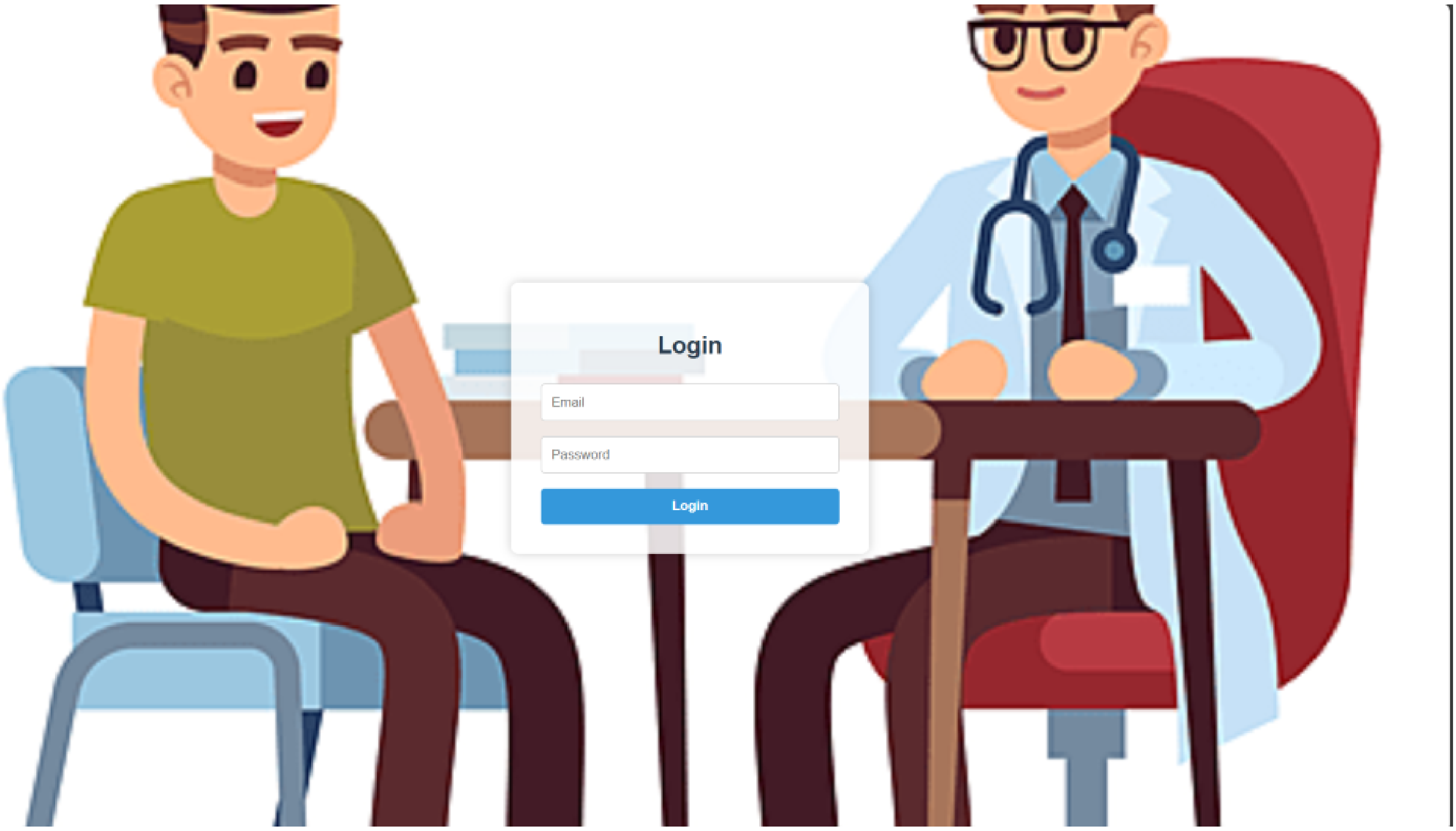
- View appointment requests

- Accept/Reject appointments

Admins:

- Approve doctors

- Manage users and appointments.



DocSpot: Doctor Appointment Booking System

Authentication

Authentication is handled using JSON Web Tokens (JWT). Passwords are encrypted using bcrypt. Token-based authentication allows users to access protected routes depending on their role.

DocSpot: Doctor Appointment Booking System

API Reference (Auth)

- POST /api/auth/register - Register a user

- POST /api/auth/login - User login and token generation

DocSpot: Doctor Appointment Booking System

API Reference (User)

- GET /api/user/doctors - Fetch available doctors

- POST /api/user/book - Book an appointment

DocSpot: Doctor Appointment Booking System

API Reference (Doctor)

- GET /api/doctor/appointments - View booked appointments

- PUT /api/doctor/update/:id - Update appointment status

DocSpot: Doctor Appointment Booking System

API Reference (Admin)

- GET /api/admin/doctors - View doctors

- PUT /api/admin/approve/:id - Approve new doctors

DocSpot: Doctor Appointment Booking System

Setup Instructions

Backend:

```
$ cd backend
$ npm install
$ npm run dev
```

Frontend:

```
$ cd frontend
$ npm install
$ npm start
```

.env:

```
PORT=5000
MONGO_URL=mongodb://127.0.0.1:27017/doctor-app
```

DocSpot: Doctor Appointment Booking System

UI Design & Navigation

The app uses Material UI and Bootstrap for styling. Pages include:

- Landing Page
- Login/Register
- Dashboards (User, Doctor, Admin)
- Appointment Forms
- Approval Panels

DocSpot: Doctor Appointment Booking System

Use Case Scenario

John logs in as a user and books an appointment with Dr. Smith. Dr. Smith approves the appointment, and John gets notified. The admin oversees this interaction and maintains platform integrity.

DocSpot: Doctor Appointment Booking System

Future Enhancements

- Email and SMS notifications
- Video consultation feature
- Payment gateway integration
- Multi-language support

DocSpot: Doctor Appointment Booking System

Author & License

Developed by Kavya M

Intern @ SmartInterns

Feel free to use and contribute to this project.