

## 1. Introduction and System Overview

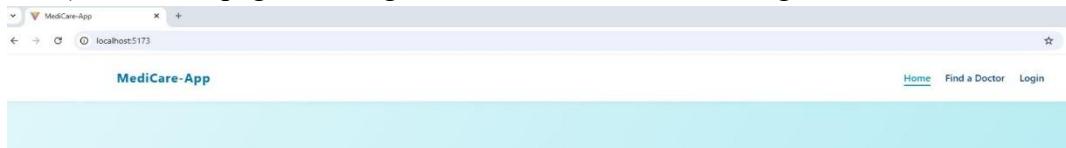
The Doctor/Dentist Appointment System, which we named MediCare App, is a web-based application designed to simplify and automate the process of booking and managing medical appointments. Traditional appointment scheduling methods often involve phone calls, long waiting times, and inefficient record handling. This system addresses these challenges by providing a digital platform where patients can register, log in, and schedule appointments with healthcare professionals in a secure and user-friendly environment.

The application enables patients to manage their appointments online, while allowing doctors and dentists to view scheduled bookings through a personalised dashboard. The system is built using modern web technologies, ensuring responsiveness, scalability, and maintainability. Emphasis is placed on usability, data security, and validation to ensure accurate and reliable handling of sensitive user information.

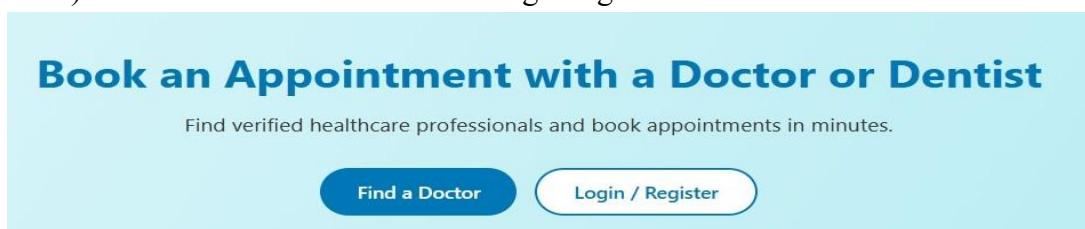
By combining a React-based front-end with a NodeJS and Express back-end and a MySQL database, the system follows a full-stack development approach and demonstrates the practical application of client-server architecture, RESTful APIs, and secure authentication mechanisms.

## 2. Screenshots of final UI

- a) The Nav page showing Home, Find a Doctor and Login Tabs



- b) The Find a Doctor and Patient login/registration tabs.



- c) Featured Doctors tab showing available medical practitioners with their review ratings.  
Also, there is a view profile tab for the doctors.



- d) How it works tab explained how the patient will book an appointment after creating an account.



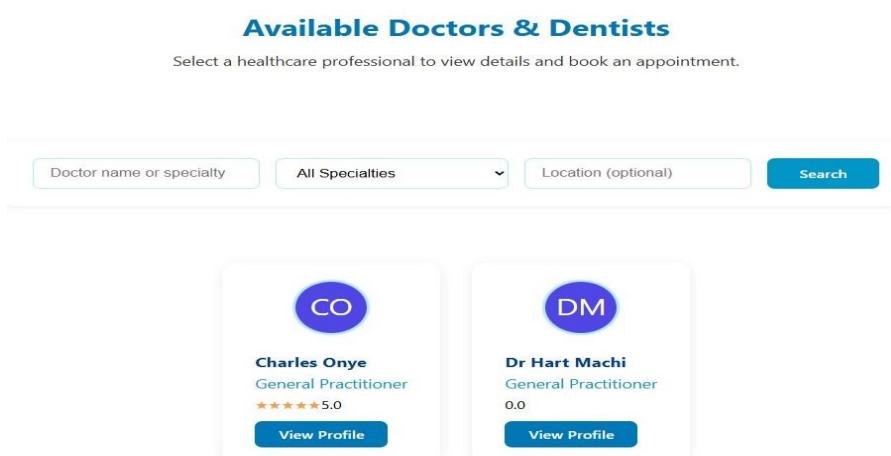
- e) Medical practitioner registration/login tab.



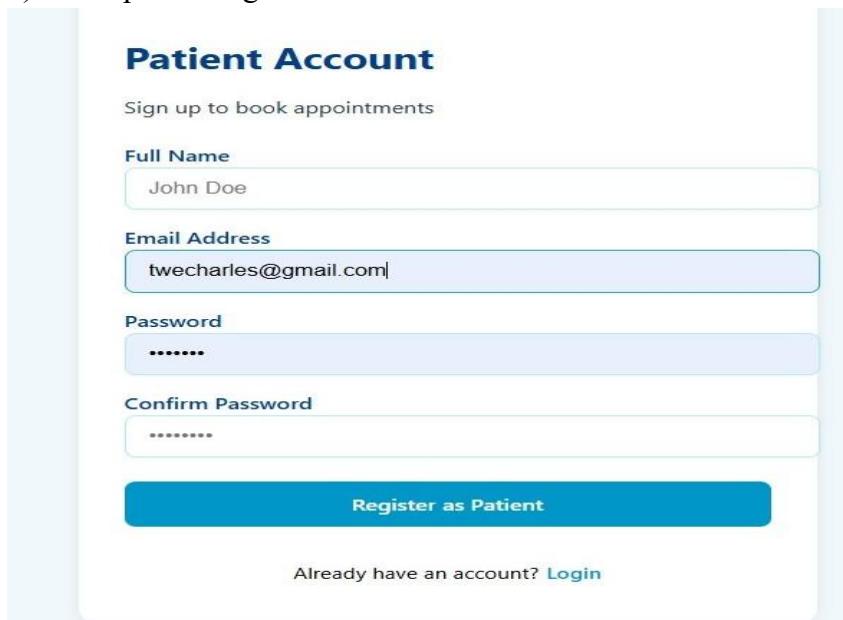
- f) Footer tab showing the about, contact, and terms & privacy



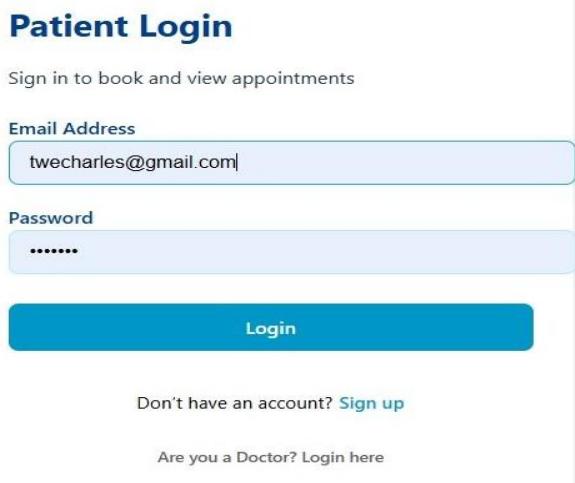
- g) Inside Find a doctor tab



- h) New patient registration tab

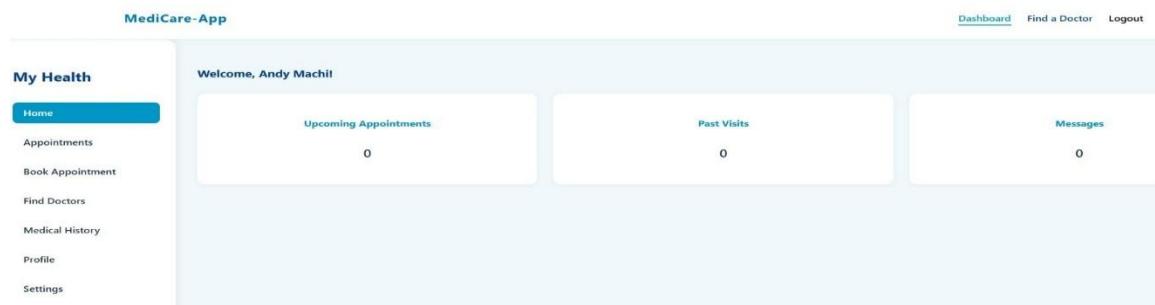


i) Patient Login tab



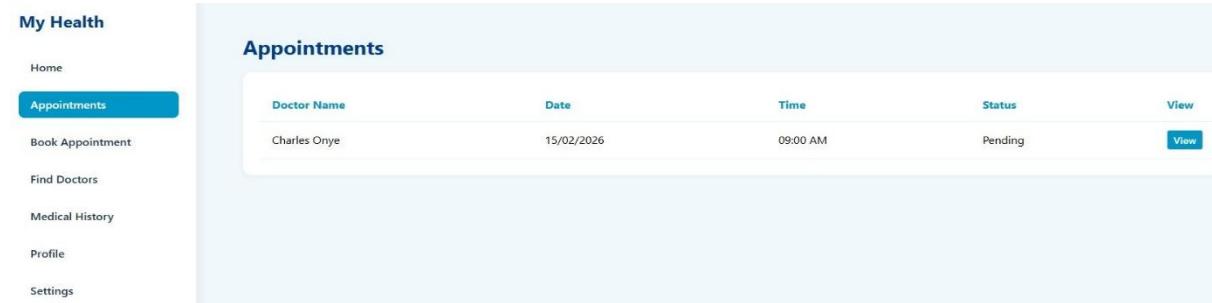
The Patient Login page features a header "Patient Login" and a sub-header "Sign in to book and view appointments". It has two input fields: "Email Address" containing "twecharles@gmail.com" and "Password" containing "\*\*\*\*\*". A blue "Login" button is centered below the inputs. Below the button, a link "Don't have an account? [Sign up](#)" is visible, along with a link "Are you a Doctor? Login here".

j) Patient Dashboard Home tab



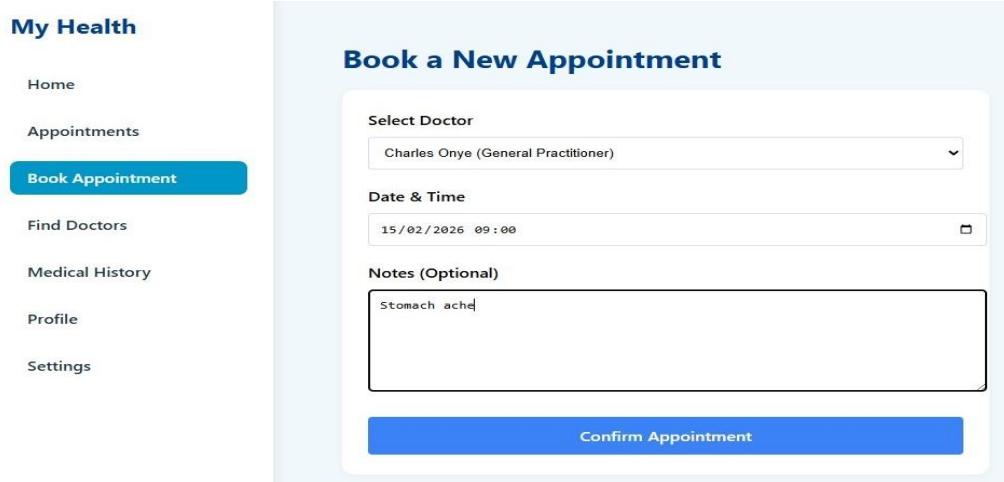
The Patient Dashboard Home page is titled "MediCare-App". It includes a top navigation bar with links "Dashboard", "Find a Doctor", and "Logout". On the left, a sidebar titled "My Health" lists "Home", "Appointments", "Book Appointment", "Find Doctors", "Medical History", "Profile", and "Settings", with "Appointments" highlighted. The main content area displays a welcome message "Welcome, Andy Machil" and three cards: "Upcoming Appointments" (0), "Past Visits" (0), and "Messages" (0).

k) Patient Appointment tab



The Patient Appointments page shows a sidebar with "My Health" and "Appointments" selected. The main content area is titled "Appointments" and lists a single entry: "Charles Onye" (Doctor Name), "15/02/2026" (Date), "09:00 AM" (Time), and "Pending" (Status). A "View" button is located to the right of the appointment details.

l) Patient's Appointment booking tab



The Book a New Appointment page has a sidebar with "My Health" and "Book Appointment" selected. The main content area is titled "Book a New Appointment". It includes fields for "Select Doctor" (set to "Charles Onye (General Practitioner)", "Date & Time" (set to "15 / 02 / 2026 09 : 00"), and "Notes (Optional)" (containing "Stomach ache"). A large blue "Confirm Appointment" button is at the bottom.

m) Patient Appointment Details tab

The screenshot shows the 'Appointment Details' section of the app. On the left, a sidebar menu includes 'Home', 'Appointments' (which is selected and highlighted in blue), 'Book Appointment', 'Find Doctors', 'Medical History', 'Profile', and 'Settings'. The main content area has a light gray background. At the top right, it says 'Appointment Details'. Below that, the appointment details are listed: Doctor: Charles Onye, Date: 16/02/2026, Time: 12:00 PM, Status: Confirmed, Notes: prescription. A red button at the bottom of this section says 'Cancel Appointment'. Below this is a 'Reschedule Appointment' section with fields for 'New Date' (dd/mm/yyyy) and 'New Time' (dropdown menu). A blue button labeled 'Reschedule' is at the bottom of this section. Underneath is a 'Medical Record' section with the text: 'Clinical notes from this visit will appear in your Medical History.' Below this is a 'Leave a Review' section. It includes a rating of 5 stars, a comment field with placeholder text 'Share your experience...', and a blue 'Submit Review' button.

n) Patient Medical History

The screenshot shows the 'Medical History' section of the app. The sidebar menu on the left is identical to the one in the previous screenshot. The main content area has a light gray background. At the top right, it says 'Medical History'. Below that, the medical history details are listed: Diagnosis: Severe fever, acute headache and tiredness, Doctor: Charles Onye (General Practitioner), Date: 15/02/2026. There are three sections: 'Prescription' (Paracetamol - twice daily, Ibuprofen - once a day), 'Treatment Plan' (Check up after fortnight), and 'Notes' (Share your experience...).

o) Patient Profile showing date registered.

The screenshot shows the 'Profile' section of the app. The sidebar menu on the left is identical to the ones in the previous screenshots. The main content area has a light gray background. At the top left, it says 'Profile'. In the center, there is a large blue circular icon containing the letters 'AM'. Below the icon, the patient's name 'Andy Machi' is displayed in bold black text. Below the name, the email address 'Email: twecharles@gmail.com' and the registration date 'Member Since: 14/02/2026' are shown.

p) Patient Settings showing the delete account tab.

The screenshot shows the 'Settings' page under 'My Health'. On the left, there's a sidebar with links: Home, Appointments, Book Appointment, Find Doctors, Medical History, Profile, and Settings (which is highlighted with a blue bar). The main area has a 'Preferences' section with 'Enable Notifications' (on) and 'Dark Mode' (on). Below it is a 'Danger Zone' section containing the warning: 'Once you delete your account, there is no going back. Please be certain.' with a red 'Delete Account' button.

q) New medical practitioner's tab for first time creating account

The screenshot shows the 'Doctor Registration' page. It starts with a welcome message: 'Join our network of healthcare professionals'. There are four input fields: 'Full Name (Dr.)' (Dr. John Doe), 'Email Address' (twecharles@gmail.com), 'Password' (.....), and 'Confirm Password' (.....). At the bottom is a large blue 'Register as Doctor' button. Below the button, a link says 'Already have an account? [Login](#)'.

r) Medical Practitioner's login page

The screenshot shows the 'Doctor Portal' login page. It features a 'Doctor Portal' header and a sub-header 'Sign in to manage your appointments'. There are two input fields: 'Email Address' (twecharles@gmail.com) and 'Password' (.....). At the bottom is a large blue 'Login as Doctor' button. Below the button, a link says 'Don't have a doctor account? [Sign up here](#)'.

s) Doctor's dashboard home tab

The screenshot shows the Doctor Dashboard with the following sections:

- Home:** Shows 1 Total Appointments.
- Today's Schedule:** Shows 0 Patients.
- Pending Requests:** Shows 1 request.

t) Doctor's Appointment tab

Patient Name	Date	Time	Status	Action
Andy Machi	15/02/2026	09:00 AM	Pending	<a href="#">View</a> <a href="#">Confirm</a> <a href="#">Reject</a>

u) Doctor's Appointment details and clinical note

**Appointment Details**

**Patient:** Andy Machi  
**Date:** 15/02/2026  
**Time:** 09:00 AM  
**Status:** Confirmed  
**Notes:** Stomach ache

✓ This appointment has been confirmed

**Medical Record**  
**Diagnosis:** Principal Diagnosis

**Prescription**  
Medications and dosage...

**Treatment Plan**  
Follow-up instructions, therapy, etc...

**Save Clinical Note**

[Back to Appointments](#)

v) Doctor's follow-up appointment booking

**Follow-up Appointment Booking**

**Select Patient:** -- Choose Patient --

**Date & Time:** dd/mm/yyyy -- : --

**Notes (Optional):**  
Add any specific instructions or reason for visit...

**Confirm Appointment**

w) Doctor's patient information.

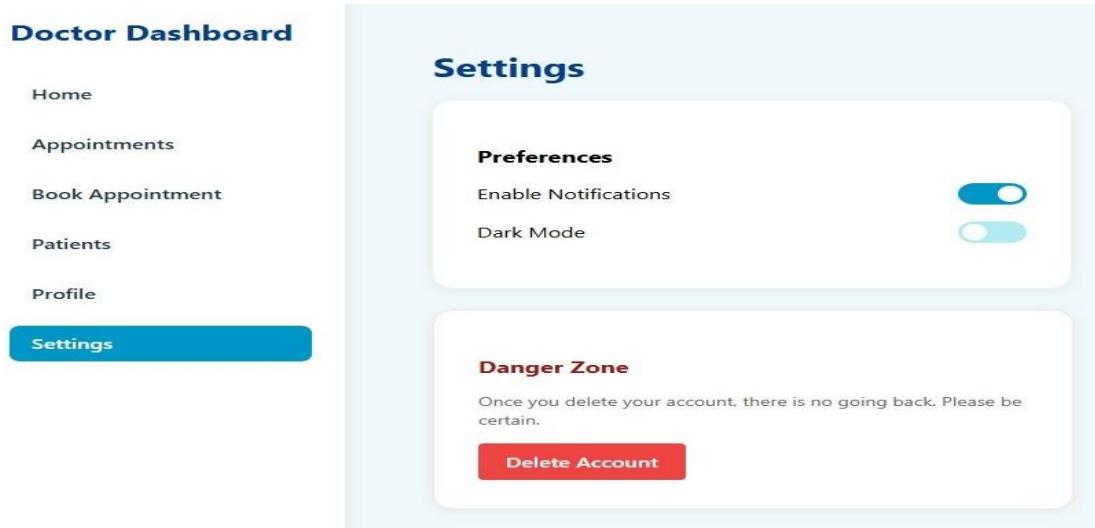
Name	Condition (Latest Note)	Last Visit	Status	Action
Andy Machi	General Checkup	N/A	proposed	<a href="#">Book</a>

x) Doctor's profile with his pay rate and email address



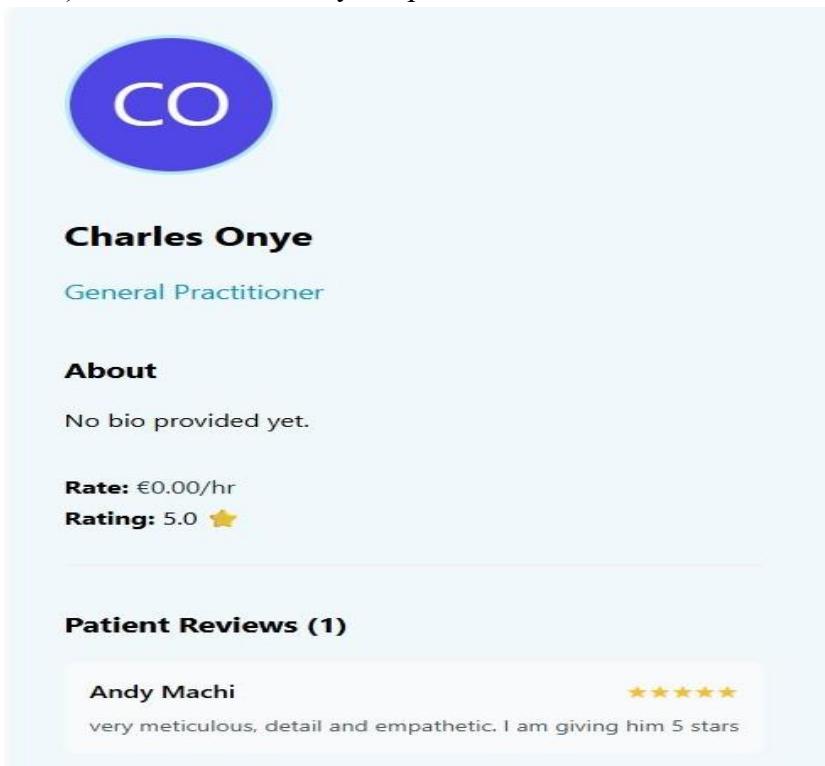
The screenshot shows the 'Profile' section of the Doctor Dashboard. On the left, there is a sidebar with links: Home, Appointments, Book Appointment, Patients, Profile (which is highlighted in blue), and Settings. The main area has a header 'Profile'. It features a circular profile picture with the letters 'CO'. Below it, the doctor's name 'Charles Onye' is displayed in bold, followed by 'General Practitioner', 'Rate: €0.00/hr', and 'Email: twecharles@yahoo.com'.

y) Doctor's setting page showing the delete tab



The screenshot shows the 'Settings' page of the Doctor Dashboard. The sidebar on the left has links: Home, Appointments, Book Appointment, Patients, Profile, and Settings (which is highlighted in blue). The main area has a header 'Settings'. It contains two sections: 'Preferences' (with 'Enable Notifications' and 'Dark Mode' toggle switches) and 'Danger Zone' (warning message: 'Once you delete your account, there is no going back. Please be certain.' with a red 'Delete Account' button).

z) Doctor's Review by the patient



The screenshot shows a patient review page for 'Charles Onye'. At the top is a circular profile picture with 'CO'. Below it, the doctor's name 'Charles Onye' is shown in bold, followed by 'General Practitioner'. A section titled 'About' contains the text 'No bio provided yet.'. Below that, 'Rate: €0.00/hr' and 'Rating: 5.0' with a yellow star icon are shown. A section titled 'Patient Reviews (1)' displays a review from 'Andy Machi' with a 5-star rating. The review text is 'very meticulous, detail and empathetic. I am giving him 5 stars'.

### **3. React front-end explanation**

#### **System Overview**

The MediCare App employs a three-tier architecture, comprising a client-side interface, a server-side application, and a relational database. Each layer is responsible for a specific function, ensuring separation of concerns and improved system reliability.

#### **Front-End (Client Side)**

The front-end of the system is developed using ReactJS, providing a dynamic and responsive user interface. The application is structured into reusable components and multiple pages (views), with client-side routing implemented to ensure smooth navigation without page reloads. The frontend is mostly controlled by the src folder.

#### **Key React pages include:**

**Home Page:** The Home page serves as the primary landing page, providing an overview of the system, including its purpose and available services. It offers navigation links to the login and registration pages.

**Login/Registration Tab:** There are two login/registration tabs, one for patients and the other for medical practitioners. The Registration tab enables new users to create an account by providing personal and login details. Input validation is applied to prevent incorrect or incomplete data entry. The login tab allows registered users to authenticate themselves using their credentials. Client-side validation is implemented to ensure required fields are completed before submission.

**Dashboard:** There are two dashboards, one for the patients and the other for the medical practitioners. After successful authentication, patients or doctors are redirected to a personalised dashboard. For the patient, the page displays relevant information, including the Home, Appointments, Book Appointment, Find Doctors, Medical History, Profile, and Settings Tabs. For the medical practitioner's tab, the page displays Home, Appointments, Book Appointments, Patients, Profile and Settings.

**How it Works tab:** Booking your healthcare appointment is simple and convenient. Just follow these easy steps.

- Search – Find a doctor or dentist by name, speciality, or location.
- Choose a Time – Select an available appointment slot that suits you.
- Book – Confirm your details and secure your appointment instantly.

The front-end uses HTML5 and CSS3 for structure and styling, ensuring accessibility and responsiveness across different devices.

#### **Frontend Documentation**

This document provides an overview of the frontend architecture, components, and data flow for the MediCare App.

## Architecture Overview

- Framework: React (Vite-based)
- Styling: Vanilla CSS
- Routing: React Router DOM (v6+)
- API Communication: Native fetch() API

## Routing Structure

The application uses a centralised routing system defined in App.jsx.

### Public Routes

- / – Landing page (Hero, Featured Doctors, How It Works)
- /login – Patient login
- /login-doctor – Doctor login
- /register – Patient registration
- /register-doctor – Doctor registration
- /doctors – Public listing of available doctors
- /doctors/:id – Detailed doctor profile

### Dashboard Routes (Protected)

- /patient-dash – Patient home overview
- /doctor-dash – Doctor home overview
- /dash/appointments – Centralised appointments view (role-dependent)
- /dash/profile – User profile management
- /dash/settings – App settings (Notifications, Dark Mode)

## Component Hierarchy & Props

Component	Description	Major Props
Navbar	Persistent top navigation	N/A
FeaturedDoctors	Grid of top doctors on homepage	Doctors (Array), onViewProfile (Fn)
DoctorCard	Reusable card for doctor summaries	Doctor (Object)
ProfileCard	Detailed user info display	Data (Object), isDoctor (Boolean)
DashboardSidebar	Dashboard navigation sidebar	basePath (String), activePage (String)
AppointmentRow	Row within the appointments table	Appointment (Object)

## State Management

### Local State (useState)

- data – Holds the API response (e.g., list of doctors)
- loading – Boolean to toggle skeleton loaders or spinners
- error – Stores API error messages for user feedback

## Global / Persistent State

The application relies on localStorage for authentication. The 'user' object stores the user's ID, name, email, role, and JWT token.

## Data Flow (Frontend Focused)

- User action (button click or page load)
- Side effect triggered via useEffect
- API call using fetch() with JWT in Authorization header
- Local state update (setLoading, setData)
- Component re-renders with live backend data

## 4. NodeJS back-end explanation

### Back-End (Server Side)

The back-end is developed using NodeJS with the Express framework. It provides RESTful API endpoints that handle communication between the front-end and the database. The server processes user requests, manages authentication, and enforces server-side validation to ensure data integrity.

#### Key back-end functionalities include:

User authentication (login and registration)

Secure handling of user sessions or tokens

Appointment management logic

Server-side input validation

API endpoints for CRUD operations

### Backend Documentation

This document details the server-side architecture, API structure, and database integration for the MediCare App.

## Technology Stack

- Runtime: Node.js
- Framework: Express.js
- Database: MySQL (MariaDB)
- Authentication: JSON Web Token (JWT)
- Drivers: mysql2/promise

## Auth Endpoints (/api/auth)

Method	Endpoint	Description
POST	/register/patient	Creates a new patient user
POST	/register/doctor	Creates a doctor user and associated profile
POST	/login	Authenticates the user and returns a JWT

### Doctor Endpoints (/api/doctors)

Method	Endpoint	Description
GET	/	List all doctors (public)
GET	/:id	Get specific doctor details
GET	/stats	(Protected) Get dashboard counts for a doctor
GET	/patients	(Protected) List patients who booked the doctor

### Appointment Endpoints (/api/appointments)

Method	Endpoint	Description
GET	/	(Protected) List all user-related appointments
POST	/	(Protected) Create a new appointment
PUT	/:id	(Protected) Update appointment status
PUT	/:id/reschedule	(Protected) Change appointment date/time

### Controller Logic

- authController.js – Handles bcrypt password hashing, JWT generation, and doctor profile creation
- doctorController.js – Manages doctor data, availability stats, and patient queries
- appointmentController.js – Booking logic, role-based filtering, and date-based queries (CURDATE())

### React – Express – MySQL Flow

React Client → Express API (JWT Auth)

→ Auth Middleware (Verify Token)

→ Controller Logic

→ MySQL Query Execution

→ JSON Response

→ React State Update & UI Re-render

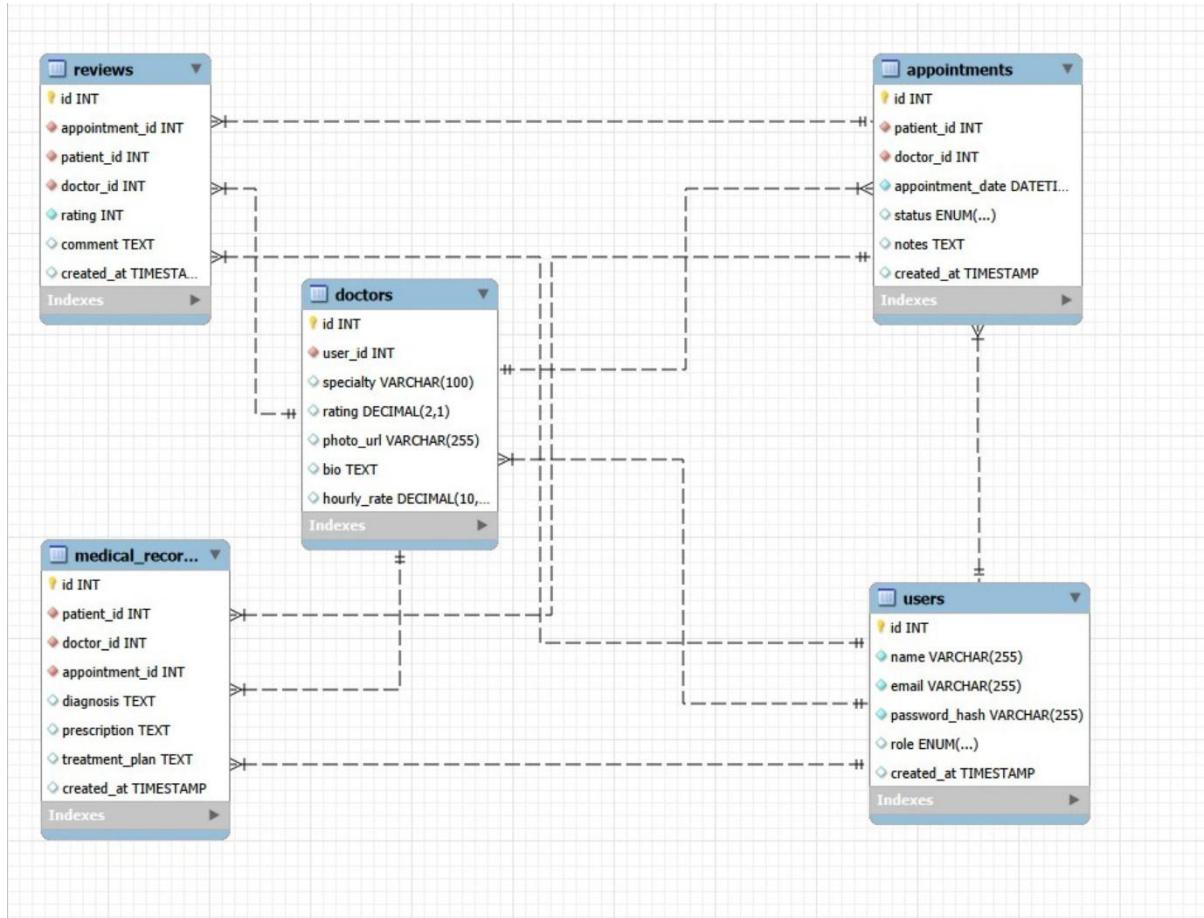
### Database Schema Highlights

- users – Central table for all accounts with role-based access
- doctors – Professional details extending user accounts
- appointments – Links patients to doctors with status tracking

### Development & Maintenance

Database initialisation is handled by config/initDb.js using schema.sql. Environment variables are managed via backend/.env for port configuration, database credentials, and JWT secrets.

## 5. Entity-Relationship Diagram (ERD) Explanation



### i. Users

This is the base table for all system users (patients, doctors, admins, etc.).

#### Primary Key (PK)

- id (INT)

#### Attributes:

- name
- email
- password\_hash
- role (ENUM – e.g. patient, doctor, admin)
- created\_at

#### Relationships

- One user can be linked to one doctor profile (via doctors.user\_id)
- One user (as a patient) can have many appointments
- One user (as a patient) can have many medical records
- One user can write many reviews

### ii. Doctors

This table stores doctor-specific details.

#### Primary Key (PK)

- id (INT)

### **Foreign Key (FK)**

- user\_id → users.id

### **Attributes:**

- speciality
- rating
- photo\_url
- bio
- hourly\_rate

### **Relationships**

- One doctor belongs to one user
- One doctor can have many appointments
- One doctor can have many medical records
- One doctor can receive many reviews

## **iii. Appointments**

This table connects patients and doctors.

### **Primary Key (PK)**

- id (INT)

### **Foreign Keys (FKs)**

- patient\_id → users.id
- doctor\_id → doctors.id

### **Attributes:**

- appointment\_date
- status (ENUM – e.g. pending, confirmed, completed, cancelled)
- notes
- created\_at

### **Relationships**

- One patient can have many appointments
- One doctor can have many appointments
- One appointment can generate one review
- One appointment can have one medical record

## **iv. Medical\_Records**

This table stores clinical information from appointments.

### **Primary Key (PK)**

- id (INT)

### **Foreign Keys (FKs)**

- patient\_id → users.id
- doctor\_id → doctors.id
- appointment\_id → appointments.id

### **Attributes:**

- diagnosis
- prescription

- treatment\_plan
- created\_at

### **Relationships**

- One patient can have many medical records
- One doctor can create many medical records
- One appointment has one medical record

### **v. Reviews**

This table stores patient feedback for doctors.

#### **Primary Key (PK)**

- id (INT)

#### **Foreign Keys (FKs)**

- appointment\_id → appointments.id
- patient\_id → users.id
- doctor\_id → doctors.id

#### **Attributes:**

- rating
- comment
- created\_at

### **Relationships**

- One patient can write many reviews
- One doctor can receive many reviews
- One appointment can have one review

### **Relationship Summary**

1. User ↔ Doctor
  - One-to-one
2. User (Patient) ↔ Appointment
  - One-to-many
3. Doctor ↔ Appointment
  - One-to-many
4. Appointment ↔ Medical Record
  - One-to-one
5. User (Patient) ↔ Medical Record
  - One-to-many
6. Doctor ↔ Medical Record
  - One-to-many
7. Appointment ↔ Review
  - One-to-one
8. Doctor ↔ Review
  - One-to-many

### **6. Video demo link is <https://youtu.be/oYDPC59rnEA>**