

## 1. Introduction and System Overview

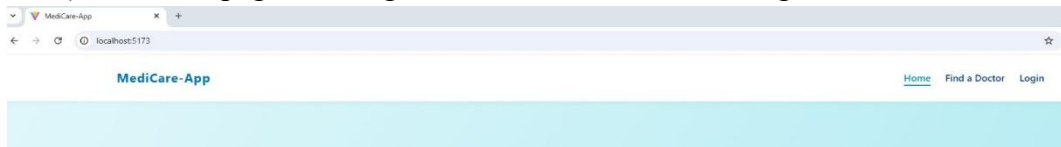
The Doctor/Dentist Appointment System, which we named MediCare App, is a web-based application designed to simplify and automate the process of booking and managing medical appointments. Traditional appointment scheduling methods often involve phone calls, long waiting times, and inefficient record handling. 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.

The "Are you a Doctor or Dentist?" section features a prominent blue button labeled "Register as a Healthcare Provider" and a link that reads "Already have an account? Login".

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

The footer contains three links: "About", "Contact", and "Terms & Privacy". Below these links is the copyright notice "© 2026 MediCare App".

- g) Inside Find a doctor tab

The "Available Doctors & Dentists" section includes a search bar with fields for "Doctor name or specialty", "All Specialties" (a dropdown menu), and "Location (optional)", followed by a "Search" button. Below the search bar, two doctor profiles are displayed. The first profile is for Charles Onye, a General Practitioner with a 5.0 star rating, featuring a "View Profile" button. The second profile is for Dr Hart Machi, also a General Practitioner with a 0.0 star rating, also featuring a "View Profile" button.

- h) New patient registration tab

The "Patient Account" registration form is titled "Sign up to book appointments". It includes input fields for "Full Name" (containing "John Doe"), "Email Address" (containing "twecharles@gmail.com"), "Password", and "Confirm Password". A large blue "Register as Patient" button is positioned below the password fields. At the bottom, there is a link that reads "Already have an account? Login".

i) Patient Login tab

## Patient Login

Sign in to book and view appointments

Email Address

twecharles@gmail.com

Password

\*\*\*\*\*

Login

Don't have an account? [Sign up](#)

Are you a Doctor? [Login here](#)

j) Patient Dashboard Home tab

MediCare-App Dashboard Find a Doctor Logout

**My Health**

- Home
- Appointments
- Book Appointment
- Find Doctors
- Medical History
- Profile
- Settings

Welcome, Andy Machil

Upcoming Appointments

0

Past Visits

0

Messages

0

k) Patient Appointment tab

**My Health**

- Home
- Appointments
- Book Appointment
- Find Doctors
- Medical History
- Profile
- Settings

### Appointments

Doctor Name	Date	Time	Status	View
Charles Onye	15/02/2026	09:00 AM	Pending	<a href="#">View</a>

l) Patient's Appointment booking tab

**My Health**

- Home
- Appointments
- Book Appointment
- Find Doctors
- Medical History
- Profile
- Settings

### Book a New Appointment

Select Doctor

Charles Onye (General Practitioner)

Date & Time

15/02/2026 09:00

Notes (Optional)

Stomach ache

Confirm Appointment

m) Patient Appointment Details tab

My Health

Home

Appointments

Book Appointment

Find Doctors

Medical History

Profile

Settings

Appointment Details

Doctor: Charles Onye

Date: 16/02/2026

Time: 12:00 PM

Status: Confirmed

Notes: prescription

Cancel Appointment

Reschedule Appointment

New Date:

dd / mm / yyyy

New Time:

-- : --

Reschedule

Medical Record

Clinical notes from this visit will appear in your Medical History.

Leave a Review

Rating

★ ★ ★ ★ ★

Comment (Optional)

Share your experience...

Submit Review

n) Patient Medical History

My Health

Home

Appointments

Book Appointment

Find Doctors

Medical History

Profile

Settings

Medical History

Diagnosis: Severe fever, acute headache and tiredness

Doctor: Charles Onye (General Practitioner)

15/02/2026

Prescription

Paracetamol - twice daily  
Ibuprofen - once a day

Treatment Plan

Check up after fortnight

o) Patient Profile showing date registered.

Profile

AM

Andy Machi

Email: twecharles@gmail.com

Member Since: 14/02/2026

p) Patient Settings showing the delete account tab.

The image shows a patient settings page. On the left is a sidebar titled "My Health" with links: Home, Appointments, Book Appointment, Find Doctors, Medical History, Profile, and Settings (highlighted in blue). The main content area is titled "Settings". It contains a "Preferences" section with "Enable Notifications" (checked) and "Dark Mode" (unchecked). Below this is a "Danger Zone" section with a warning: "Once you delete your account, there is no going back. Please be certain." and a red "Delete Account" button.

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

The image shows a "Doctor Registration" form. It starts with the heading "Doctor Registration" and a subtext "Join our network of healthcare professionals". The form has four input fields: "Full Name (Dr.)" with the value "Dr. John Doe", "Email Address" with the value "twecharles@gmail.com", "Password" with masked characters "\*\*\*\*\*", and "Confirm Password" with masked characters "\*\*\*\*\*". Below the fields is a blue "Register as Doctor" button. At the bottom, there is a link: "Already have an account? [Login](#)".

r) Medical Practitioner's login page

The image shows a "Doctor Portal" login form. It has the heading "Doctor Portal" and the subtext "Sign in to manage your appointments". The form has two input fields: "Email Address" with the value "twecharles@gmail.com" and "Password" with masked characters "\*\*\*\*\*". Below the fields is a blue "Login as Doctor" button. At the bottom, there is a link: "Don't have a doctor account? [Sign up here](#)".

s) Doctor's dashboard home tab

**Doctor Dashboard**  
Home  
Appointments  
Book Appointment  
Patients  
Profile  
Settings

Welcome, Charles Onyiah  

Total Appointments  
1

Today's Schedule  
0 Patients

Pending Requests  
1

t) Doctor's Appointment tab

**Doctor Dashboard**  
Home  
**Appointments**  
Book Appointment  
Patients  
Profile  
Settings

**Appointments**  

Patient Name	Date	Time	Status	
Andy Machi	15/02/2026	09:00 AM	Pending	View Confirm Reject

u) Doctor's Appointment details and clinical note

**Doctor Dashboard**  
Home  
**Appointments**  
Book Appointment  
Patients  
Profile  
Settings

**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

**Doctor Dashboard**  
Home  
Appointments  
**Book Appointment**  
Patients  
Profile  
Settings

**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.

**Doctor Dashboard**  
Home  
Appointments  
Book Appointment  
**Patients**  
Profile  
Settings

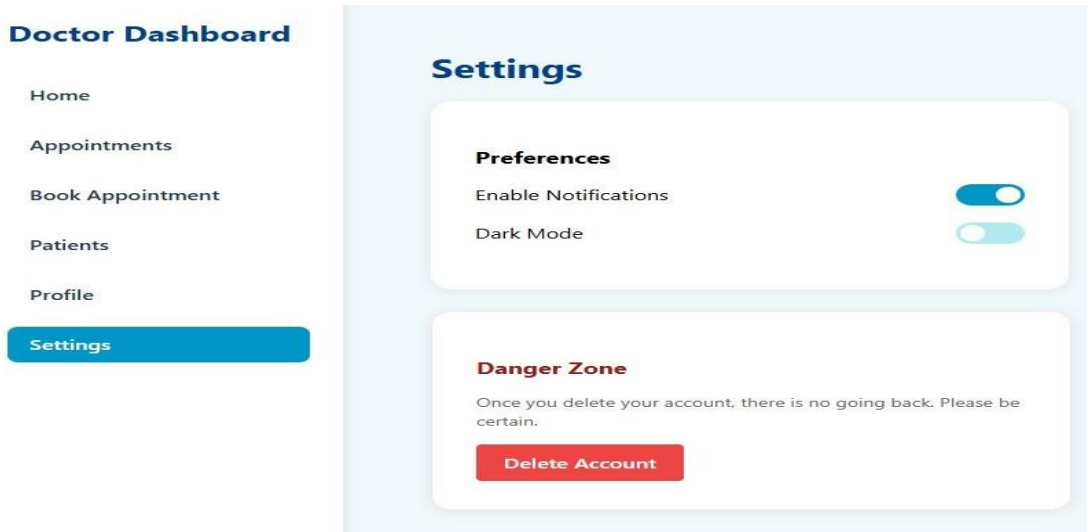
**Patients**  

Name	Condition (Latest Note)	Last Visit	Status	Action
Andy Machi	General Checkup	N/A	proposed	Book

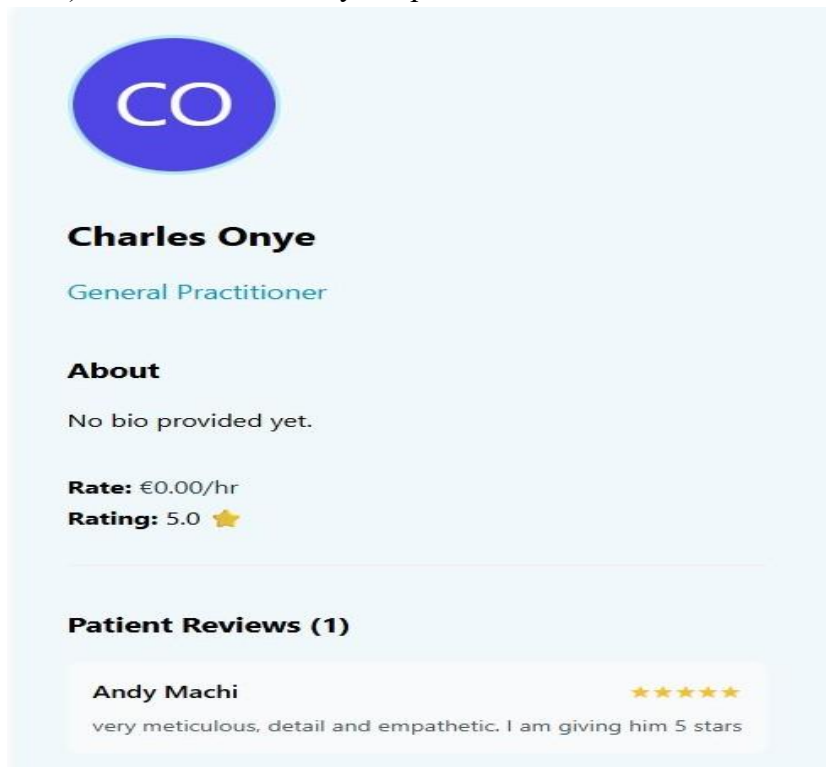
x) Doctor's profile with his pay rate and email address



y) Doctor's setting page showing the delete tab



z) Doctor's Review by the patient



### 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

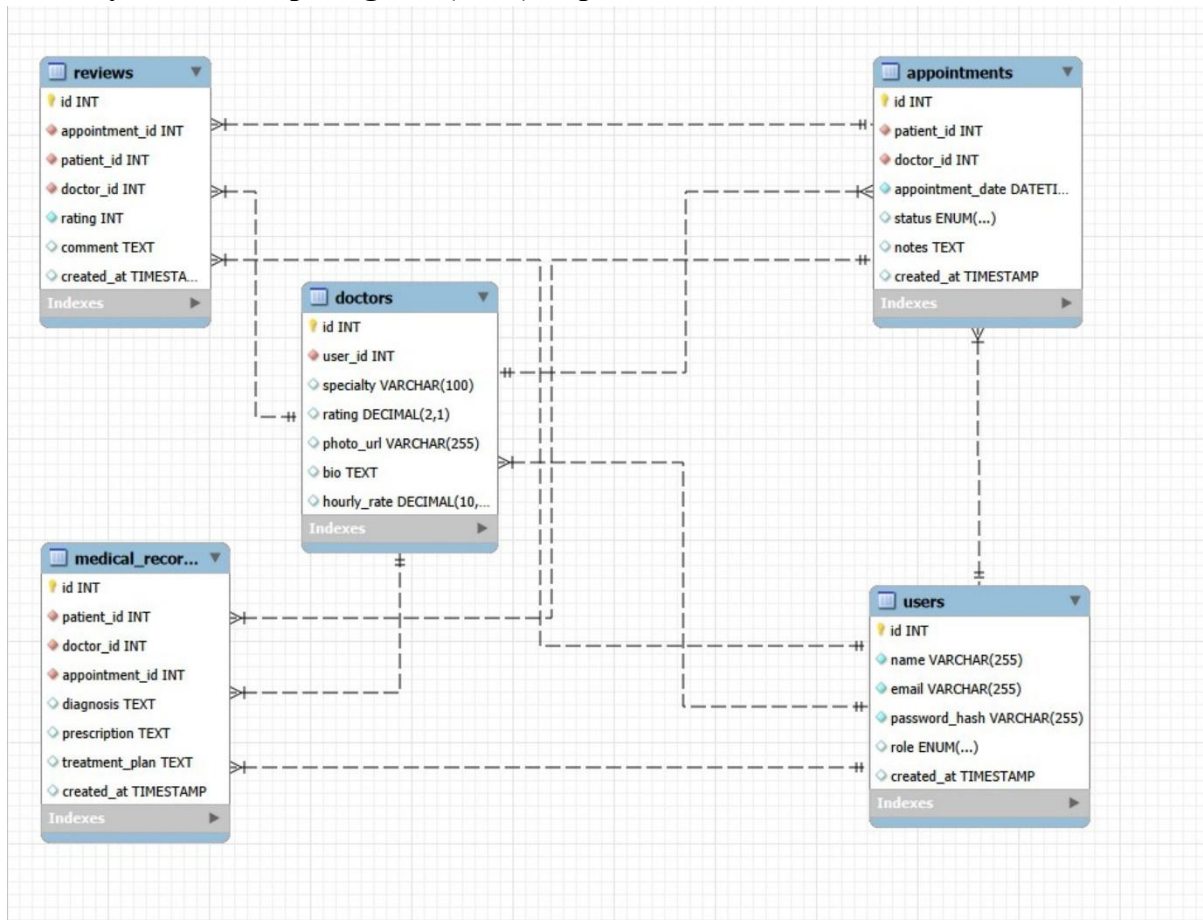
### Database Schema Highlights

- 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/oYDPC59mEA>