

FRACTO

-Parth Tinna

Table of Contents

1. Introduction and Project Goals
 2. System Architecture
 3. Workflow and Design Diagrams
 4. Application Modules & API Overview
 5. Database Schema and Data Handling
 6. Testing Strategy
 7. Conclusion
-

1. Introduction and Project Goals

Problem Context

In many hospitals and clinics, patients still depend on outdated methods to book appointments, such as phone calls or in-person scheduling. These approaches are time-consuming, prone to errors, and offer little to no transparency about doctor availability. Administrators, meanwhile, must manually track appointments, often resulting in double-bookings or missed slots.

Project Goals

The Fracto system was developed with the following intentions:

- Deliver a **web-based platform** that simplifies appointment booking for patients.

- Allow **administrators to efficiently manage** doctors, patients, and schedules.
 - Guarantee **security** using JWT authentication and role-based permissions.
 - Provide **real-time updates** through SignalR to improve communication.
 - Enhance user engagement with a **doctor rating and feedback system**.
 - Support **profile image uploads** to maintain professional and user identities.
-

2. System Architecture

Chosen Technologies

- **Frontend:** Angular 16 with Angular Material for responsive, interactive UI.
- **Backend:** ASP.NET Core Web API (.NET 9) with EF Core ORM for smooth data operations.
- **Database:** SQL Server for production; SQLite during development for simplicity.
- **Security:** JWT-based authentication tokens with embedded claims.
- **File Storage:** Local file system (wwwroot/uploads).

Interaction Model

1. **Patient/Admin** accesses Angular frontend.
 2. Requests are routed to **ASP.NET Core API endpoints**.
 3. The backend uses **EF Core** to query and update the database.
 4. Uploaded files are stored and retrieved from the server directory.
-

3. Workflow and Design Diagrams

Application Workflow

- **Patient Workflow:** Register → Login → Select City → Search Doctor → Choose Slot → Book Appointment → Receive Confirmation .
- **Admin Workflow:** Login → Manage Users → Add/Edit Doctors → Approve/Reject Appointments

Design Flow

- **Frontend** handles navigation, form validation, and API requests.
 - **Backend** provides endpoints for each feature (Auth, Users, Doctors, Appointments, Ratings).
 - **Database** maintains relationships between users, doctors, and appointments.
 - **SignalR Hub** ensures real-time synchronization.
-

4. Application Modules & API Overview

Angular Modules

- **Authentication:** Components for Login and Register with Guards for route protection.
- **User Section:** DoctorSearchComponent, AppointmentBookComponent, UserAppointmentsComponent.
- **Admin Section:** Components for User Management, Doctor Management, Specialization Management, and Appointment Oversight.
- **Shared Utilities:** Navbar, Snackbar notifications, and FileUpload component.

Web API Controllers

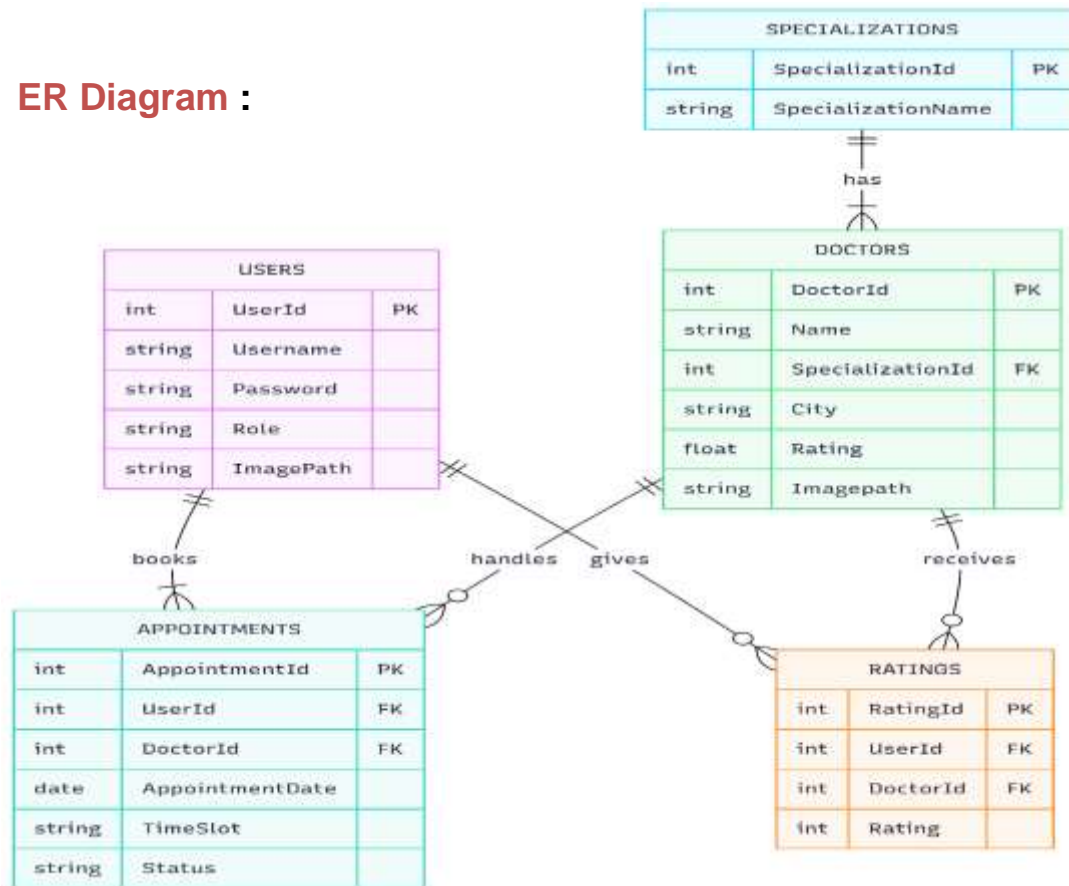
- **AuthController** → Register & Login.
- **UsersController** → Manage patients (CRUD, Admin only).
- **DoctorsController** → CRUD, search by filters, and timeslot availability.
- **AppointmentsController** → Book, reschedule, cancel, approve, reject.

5. Database Schema and Data Handling

Core Tables

- **Users:** UserId, Username, Password, Role, ProfileImagePath.
- **Doctors:** DoctorId, Name, SpecializationId, City, Rating, ProfileImagePath.
- **Specializations:** SpecializationId, Name.
- **Appointments:** AppointmentId, UserId, DoctorId, AppointmentDate, TimeSlot, Status.
- **Ratings:** RatingId, DoctorId, UserId, Rating, Comment.

ER Diagram :



Data Handling Strategies

- Referential integrity enforced with **foreign keys**.
- Indexes on **UserId**, **DoctorId**, **SpecializationId** for faster queries.
- Frontend caches data like specializations to reduce repeated API calls.

6. Testing Strategy

Backend

- **Frameworks**: xUnit with Moq.
- **Approach**: EF Core InMemory database to test controllers without affecting production.
- **Scenarios Tested**:

- User registration and login produce valid JWTs.
- Booking an appointment creates a record.
- Admin approval updates appointment status correctly.

Frontend

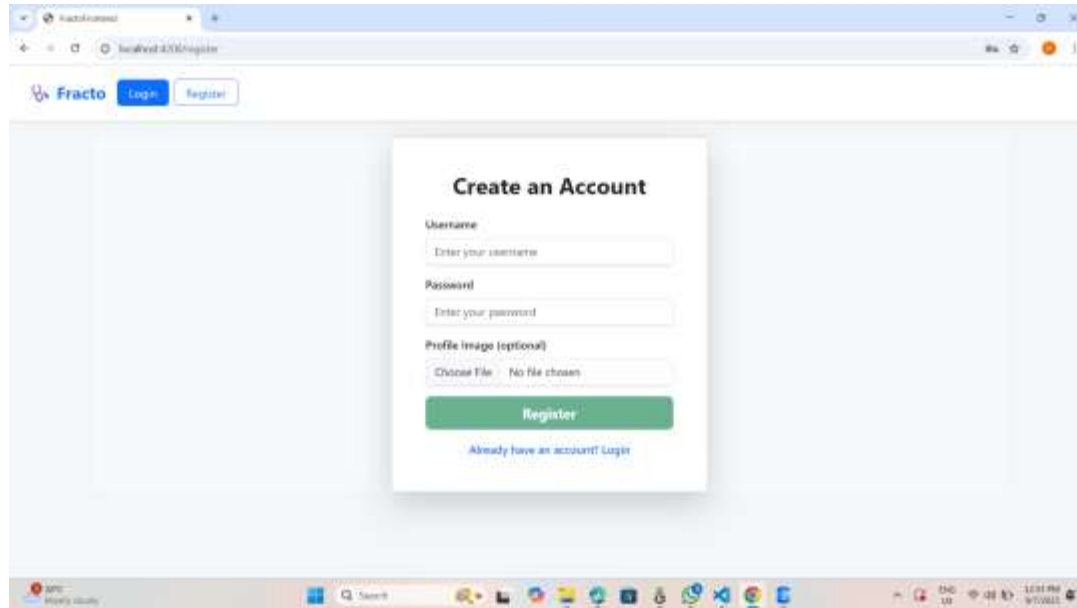
- **Tools:** Jasmine + Karma.
- **Unit Tests:**
 - Login form validates required fields.
 - Appointment booking triggers the correct API call.
 - Doctor search renders accurate data.

Integration

- **Postman** used for testing REST APIs.
- Manual linking tests with Angular frontend and ASP.NET backend.
- Future E2E scope with Cypress.

7. Implementation Screenshots

User Register:



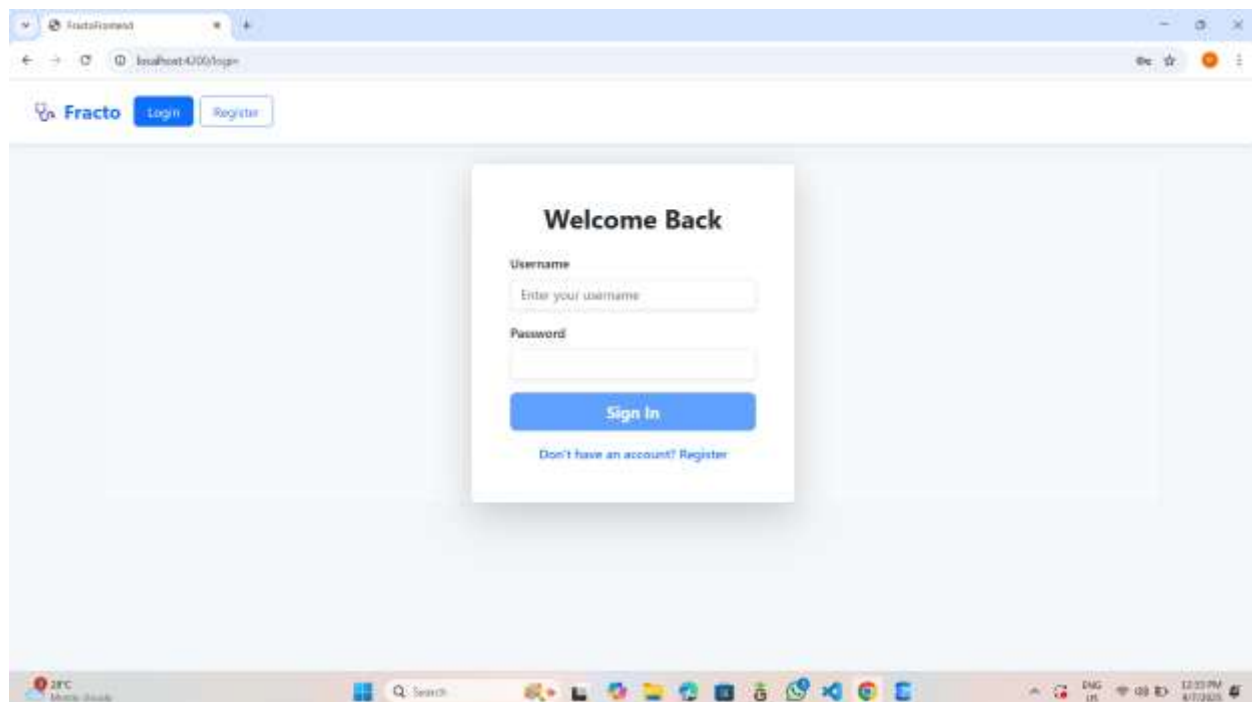
The screenshot shows a web browser window with the URL `localhost:4200/register`. The page features a "Fracto" logo and "Login" and "Register" buttons. A central modal titled "Create an Account" contains the following fields and elements:

- Username:** A text input field with the placeholder "Enter your username".
- Password:** A text input field with the placeholder "Enter your password".
- Profile image (optional):** A file selection area showing "Choose File" and "No file chosen".
- Register:** A green button to submit the registration form.
- Already have an account? Login:** A blue link to the login page.

The Windows taskbar at the bottom shows the system clock as 12:31 PM on 8/10/2021.

:

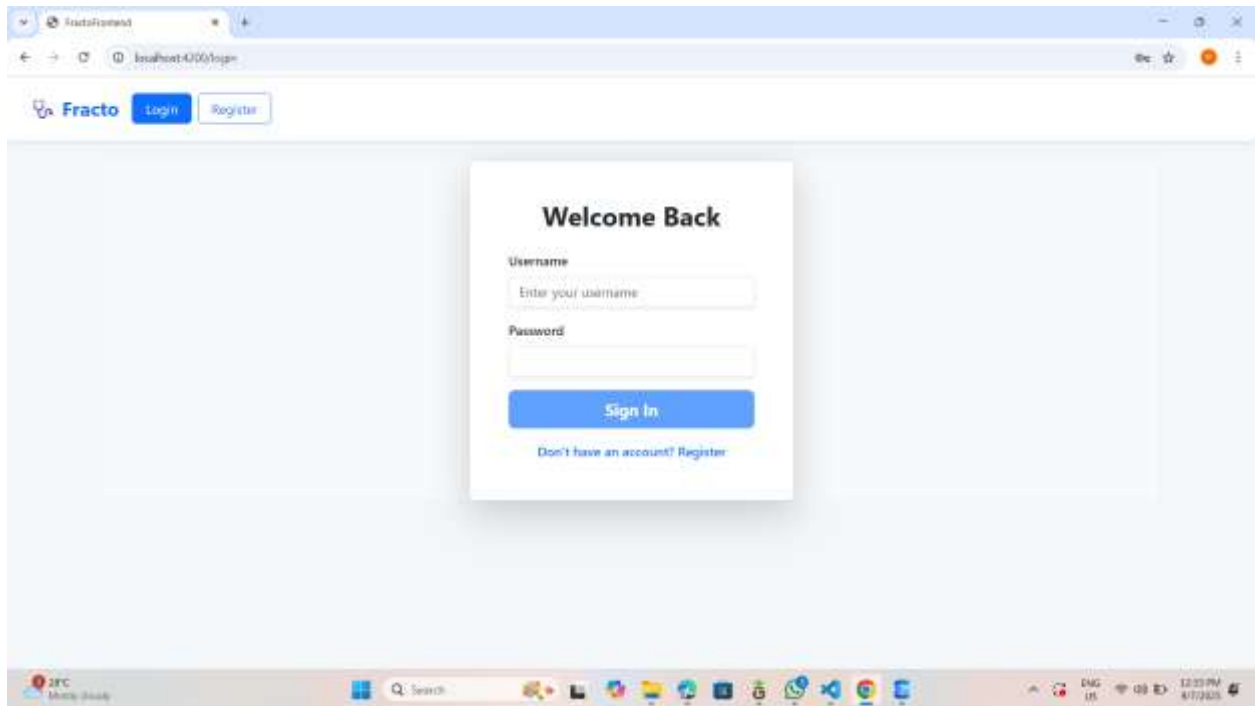
User Login:



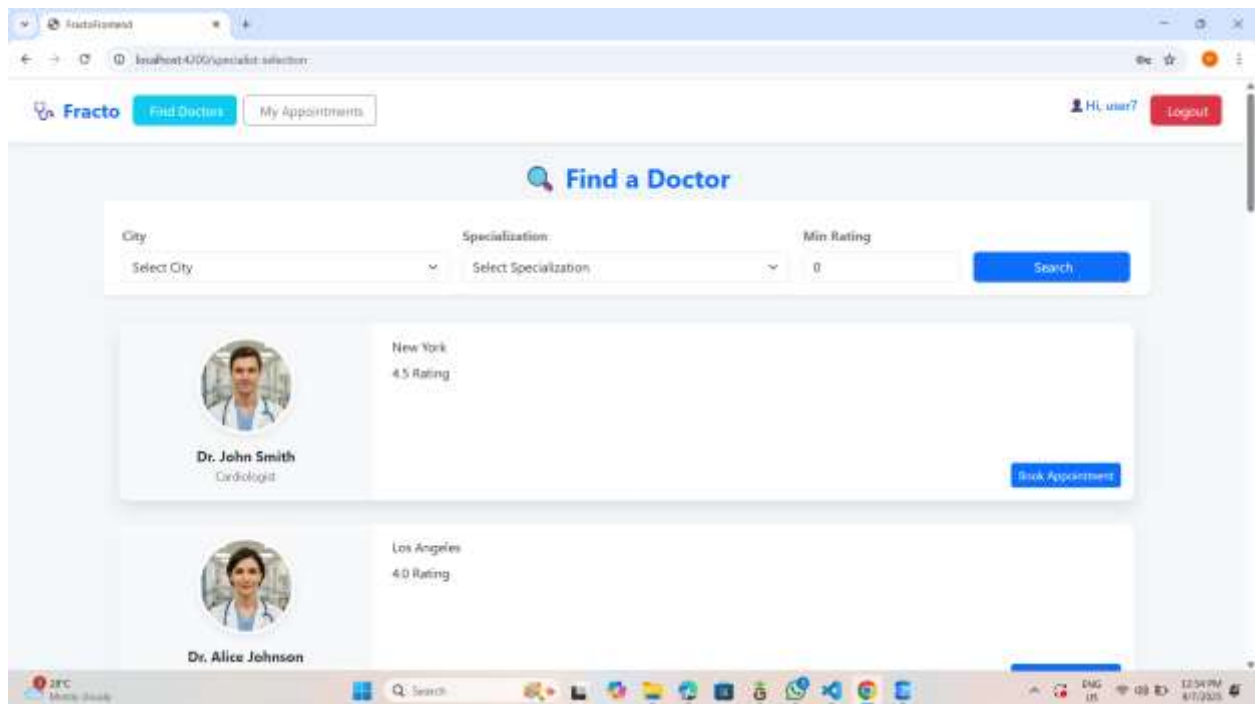
The screenshot shows a web browser window with the URL `localhost:4200/login`. The page features a "Fracto" logo and "Login" and "Register" buttons. A central modal titled "Welcome Back" contains the following fields and elements:

- Username:** A text input field with the placeholder "Enter your username".
- Password:** A text input field.
- Sign in:** A blue button to submit the login form.
- Don't have an account? Register:** A blue link to the registration page.

The Windows taskbar at the bottom shows the system clock as 12:33 PM on 8/10/2021.



Doctor's List:



Booking Appointment On Basis of Slot:

The screenshot shows a web browser window with the URL `localhost:4200/appointment-book?doctorId=1&doctorName=Dr.%20John%20Smith`. The page has a header with the Fracto logo, a 'Find Doctors' button, and a 'My Appointments' button. A user is logged in as 'Hi, user7' with a 'Logout' button. The main heading is 'Book an Appointment'. Below it is a form with the following fields:

- Doctor: Dr. John Smith
- Appointment Date: 09/17/2025
- Time Slot: 11:30 - 12:00

A blue 'Book Appointment' button is at the bottom of the form.

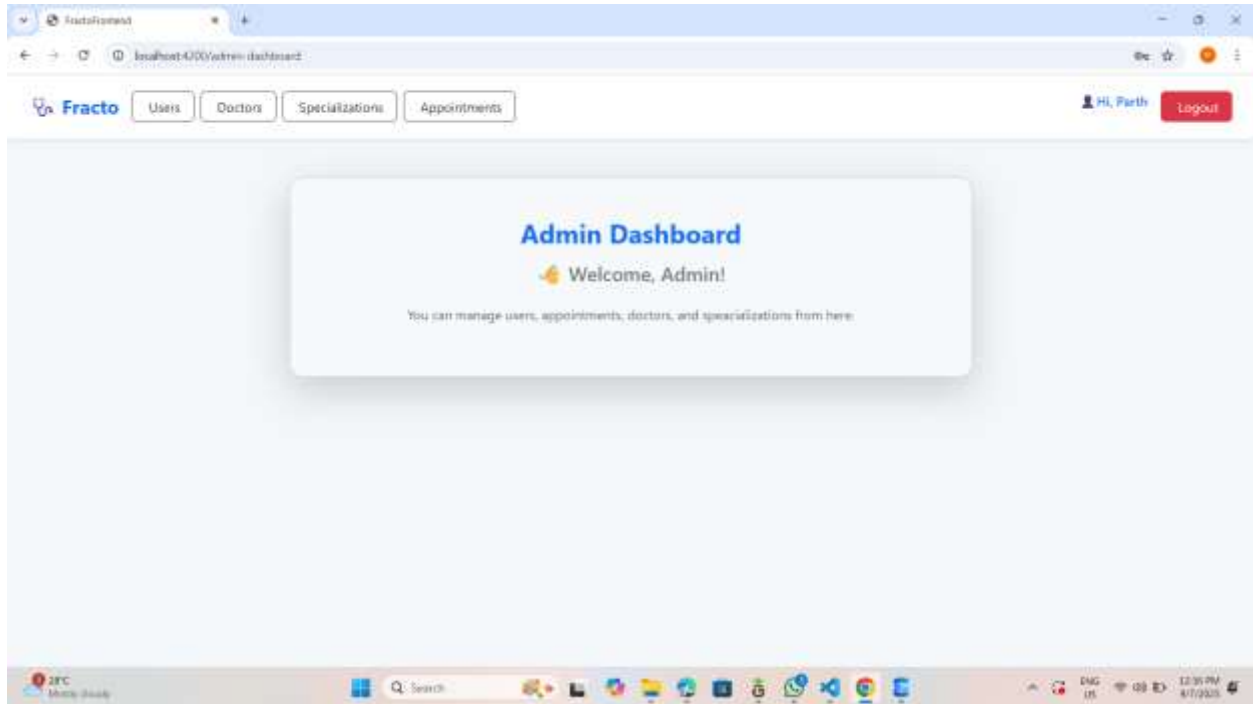
Checking Appointments:

The screenshot shows the 'My Appointments' page. It features a table with one appointment entry. The table has columns for Doctor, Date, Time Slot, Status, and Action.

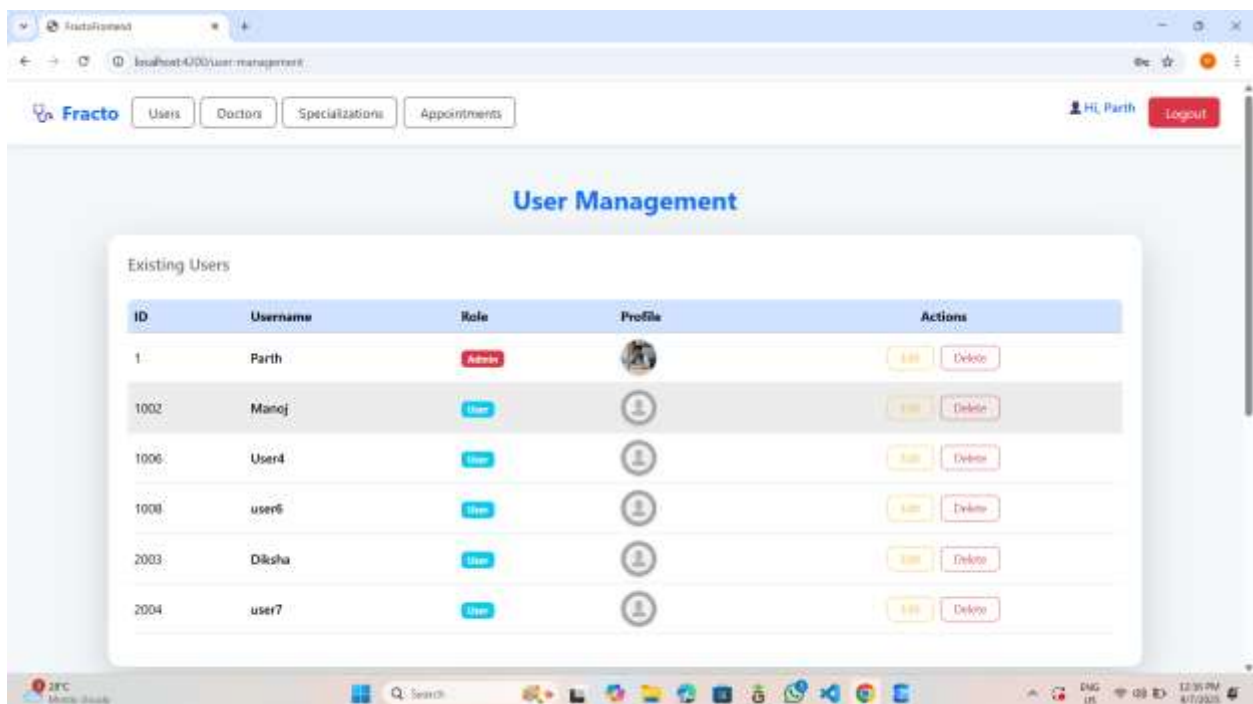
| Doctor | Date | Time Slot | Status | Action |
|----------------|------------|---------------|---------|-------------------------|
| Dr. John Smith | 2025-09-17 | 11:30 - 12:00 | Pending | <button>Cancel</button> |

Admin Side

Admins Dashboard:



Admin User Management:



Admin Doctors Management:

The screenshot shows the 'Doctor Management' page in the Fracto admin interface. The page has a navigation bar with 'Users', 'Doctors', 'Specializations', and 'Appointments' tabs. The 'Doctors' tab is active. The main content area is titled 'All Doctors' and displays a table of doctors with their details and actions.

| Name | City | Specialization | Rating | Profile | Actions |
|-------------------|--------------|----------------|--------|---------|---|
| Dr. John Smith | New York | Cardiologist | ★ 4.5 | | Edit Delete |
| Dr. Alice Johnson | Los Angeles | Dermatologist | ★ 4.0 | | Edit Delete |
| Dr. Michael Brown | Chicago | Neurologist | ★ 3.8 | | Edit Delete |
| Dr. Sarah Lee | Houston | Pediatrician | ★ 4.2 | | Edit Delete |
| Dr. David Wilson | Phoenix | Orthopedic | ★ 3.9 | | Edit Delete |
| Dr. Emily Davis | Philadelphia | Cardiologist | ★ 4.7 | | Edit Delete |

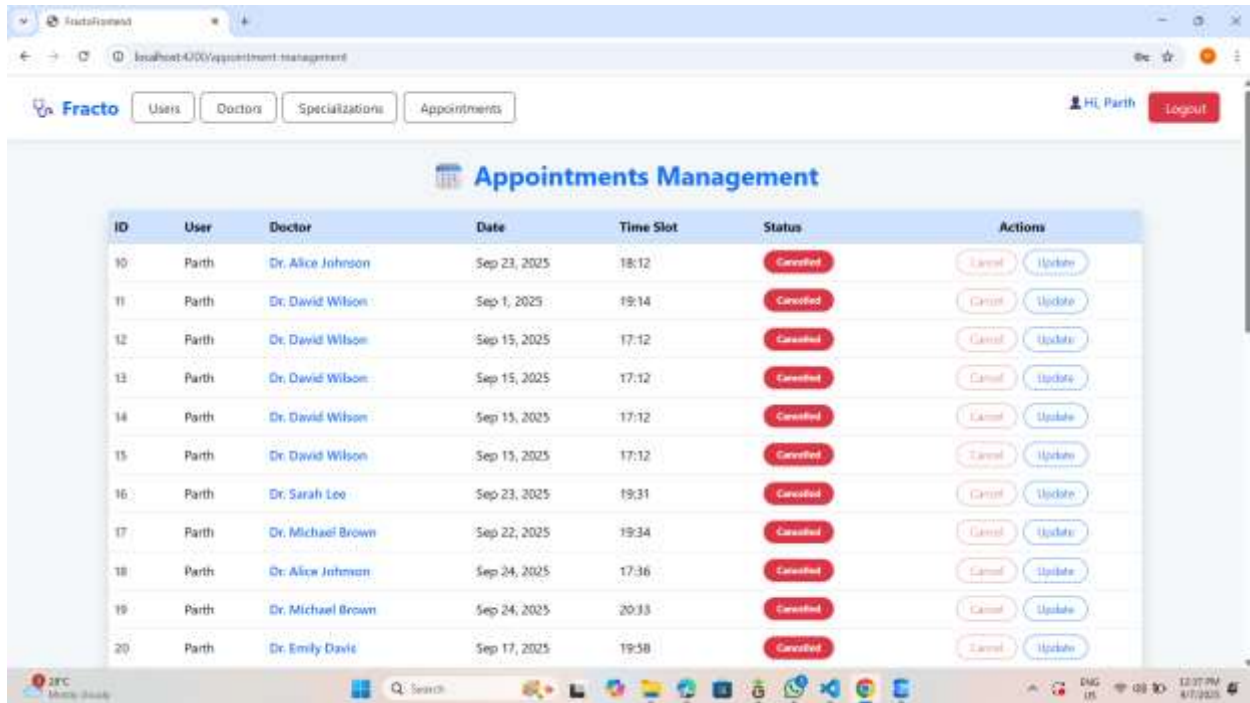
Admin Specializations Management

The screenshot shows the 'Specialization Management' page in the Fracto admin interface. The page has a navigation bar with 'Users', 'Doctors', 'Specializations', and 'Appointments' tabs. The 'Specializations' tab is active. The main content area is titled 'Specialization Management' and displays a table of specializations with their details and actions.

| # | Specialization | Actions |
|---|----------------|---|
| 1 | Cardiologist | Edit Delete |
| 2 | Dermatologist | Edit Delete |
| 3 | Neurologist | Edit Delete |
| 4 | Pediatrician | Edit Delete |
| 5 | Orthopedic | Edit Delete |

Below the table, there is a form to add a new specialization. It includes a text input field labeled 'Specialization Name' with the placeholder text 'e.g. Cardiologist, Dermatologist'. Below the input field are two buttons: 'Add Specialization' and 'Cancel'.

Admin Appointment Management:



| ID | User | Doctor | Date | Time Slot | Status | Actions |
|----|-------|-------------------|--------------|-----------|-----------|---|
| 10 | Parth | Dr. Alice Johnson | Sep 23, 2025 | 18:12 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 11 | Parth | Dr. David Wilson | Sep 1, 2025 | 19:14 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 12 | Parth | Dr. David Wilson | Sep 15, 2025 | 17:12 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 13 | Parth | Dr. David Wilson | Sep 15, 2025 | 17:12 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 14 | Parth | Dr. David Wilson | Sep 15, 2025 | 17:12 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 15 | Parth | Dr. David Wilson | Sep 15, 2025 | 17:12 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 16 | Parth | Dr. Sarah Lee | Sep 23, 2025 | 19:31 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 17 | Parth | Dr. Michael Brown | Sep 22, 2025 | 19:34 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 18 | Parth | Dr. Alice Johnson | Sep 24, 2025 | 17:36 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 19 | Parth | Dr. Michael Brown | Sep 24, 2025 | 20:33 | Cancelled | <button>Cancel</button> <button>Update</button> |
| 20 | Parth | Dr. Emily Davis | Sep 17, 2025 | 19:58 | Cancelled | <button>Cancel</button> <button>Update</button> |

8. Conclusion

Fracto has been built to streamline healthcare appointment management through a digital-first solution. It bridges the gap between patients and healthcare providers with **real-time visibility, secure access, and user-friendly interfaces**.

The project successfully delivers:

- **Ease for patients** in booking and managing appointments.
- **Efficiency for administrators** in overseeing users, doctors, and schedules.
- **Security** with robust authentication and role-based permissions.
- **Engagement** through ratings and instant notifications.