Fracto – Online Doctor Appointment Booking System

Project Title: Fracto

Submitted By: Guduru Durga Anvitha Lakshmi
Batch: WIPRO NGA- .Net Full Stack Angular- FY26- C2
Instructor: Jyoti Patil
Date: 8-9-2025

Table of Contents

1. Project Overview - Page 1
2. Problem Definition & Objectives - Page 1
3. System Architecture - Page 2
4. Frontend Architecture & Component Breakdown - Page 3
5. Backend Architecture & API Design

--- API Design & Endpoints -Page 4
6. Database Design (ERD)+Diagram -Page 4
7.Supporting Files -Page 5
8. Backend User Endpoints: CRUD -Page 6-11
9. User Stories -Page 11-13
10. Output Snapshots(API EndPoints+Angular) -Page 14-32
11. Conclusion -Page 32

1. Project Overview

Fracto is an online doctor appointment booking platform built using Angular (Frontend) and ASP.NET Core MVC (Backend). It allows users to search for doctors by city and specialization, view available time slots, and book or cancel appointments. Admins can manage doctors, users, and appointments.

2. Problem Definition & Objectives

Problem Statement:

Traditional appointment booking requires patients to visit hospitals or call manually, leading to inefficiency.

Objectives:

- Allow users to search for doctors by city, specialization, and rating - Enable online booking and cancellation of appointments.

-Allow users to register, login, and book/cancel appointments.
-Provide rating and filtering system for doctors.
- Provide an admin dashboard to manage doctors and appointments.
- Secure authentication using JWT tokens.

3. System Architecture

Tech Stack:
- Frontend: Angular
- Backend: ASP.NET Core MVC + Web API
- Database: SQL Server (Entity Framework Core ORM)
- Authentication: JWT Token
- API Testing: Swagger
- Image Upload: Profile images stored on server

4. Frontend Architecture & Component Breakdown Components:

- LoginComponent: login & session handling

- RegisterComponent: registration form & validation

- UserDashboardComponent: user-dashboard, home ,user-appointments, appointment-book, view, cancel appointments, profile,doctor-list.

- AdminDashboardComponent: manage users, doctor-list-admin,add-doctor,specialization-list, approve/cancel appointments.

- Services: AuthService,Doctor-service,Specialization,appointment,User service

- Auth Interceptor,Auth Guard(Session Handling)

5. Backend Architecture & API Design

- Controllers: UserController, SpecializationController , AppointmentController DoctorController, DoctorAvailabilityController, RatingController.
- Models: User, Specialization, Appointment, Doctor, Rating, AppDbContext.

API END-POINTS

Endpoint	**Method**	**Description**	**Authentication**
/User/register	POST	Register user	None

/User/login	POST	Login user & return	None
/User/refresh-token	POST	Refreshes token-user	JWT
/User/all	GET/POST/PUT/DELETE	CRUD on user	Admin
/User/upload-profileimage	POST	Image Upload	Users
/Specialization	GET/POST/DELETE	CRD on specialization	JWT+ Admin
/Doctor	GET/POST/PUT/DELETE	CRUD on doctors	JWT + Admin
/Appointment	GET/POST/PUT/DELETE	Manage appointments	JWT
/Rating	GET/POST	Rate doctor	JWT
/Booking/available-slots	GET	Slots based on doctorId	None

6. Database Design & Storage Optimization

Tables:

Users (UserId, Username, Password, Role, City, Email)
Doctors (DoctorId, Name, SpecializationId, City, Rating, ImagePath)
Specializations (SpecializationId, SpecializationName)
Appointments (AppointmentId, UserId, DoctorId, AppointmentDate, TimeSlot, Status)
Ratings (RatingId, DoctorId, UserId, Rating) Relationships:

Specializations → Doctors

Relationship: One-to-Many (1 → ∞)

- **Specializations.SpecializationId → Doctors.SpecializationId**
- Meaning:
  - Each specialization (e.g., "Cardiology", "Dermatology") can have **many doctors**.
  - A doctor **must belong to exactly one specialization**.

Doctors → Appointments

Relationship: One-to-Many (1 → ∞)

- **Doctors.DoctorId → Appointments.DoctorId**
- Meaning:
    - Each doctor can have **many appointments** booked by different users.
    - Each appointment is linked to **exactly one doctor**.

Users → Appointments

Relationship: One-to-Many (1 → ∞) •

Users.UserId → Appointments.UserId

- Meaning:
    - Each user can book **multiple appointments**.
    - Each appointment is associated with **exactly one user**.

Doctors → Ratings

Relationship: One-to-Many (1 → ∞) •

Doctors.DoctorId → Ratings.DoctorId

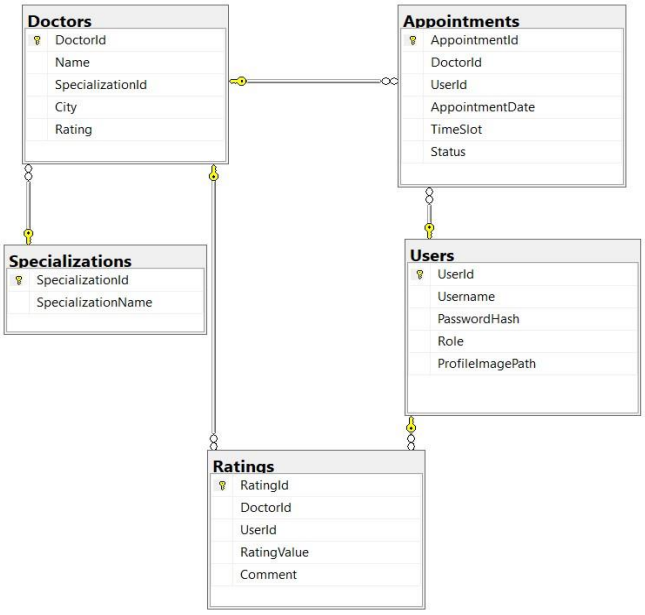
- Meaning:
    - Each doctor can have **many ratings** given by different users.

Users → Ratings

Relationship: One-to-Many (1 → ∞)

- **Users.UserId → Ratings.UserId**
- Meaning:
    - Each user can give ratings to multiple doctors (one rating per doctor ideally).
    - A rating is tied to exactly **one user** and **one doctor**.

- ## DataBase Diagram:



7.Supporting Files (if applicable needs to properly uploaded on Olympus):

- **Frontend Code: Frontend_Fracto**

https://drive.google.com/file/d/1C8YnlZZ1cpQrt1SchcJtdkajFTrr4Q2L/view?usp=drivesdk

- Frontend_Fracto.zip
- **Contains full React/Angular source code**

- **Backend Code: Backend_Fracto**

https://drive.google.com/file/d/1wuMR1kLrIaZN4KdSoq35jUjZyvstJgbD/view?usp=drivesdk

- Backend_Fracto.zip
- **Contains full ASP.NET Core Web API source code**

- **Database & Configuration Files: Database_Schema_Fracto**

https://drive.google.com/file/d/1R_n1W4ZN9OGoNqxzOhrCPGZVcKukanfQ/view?usp=drivesdk

- **Database scripts:** Database_Schema_Fracto.sql

- **Deployment : Github Fracto Project Link:**

https://github.com/anvitha31303/Fracto

- **Project needs to be pushed on GitHub and keep it as private**

8.Backend User Endpoints: CRUD Operation(Register/Login/Get/Put/Delete/AddProfilePic)

/User/register

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5189/api/User/register' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "shiv",
  "password": "shiv",
  "role": "User"
}'
```

Request URL

```
http://localhost:5189/api/User/register
```

Server response

Code	Details

200	Response body
	`{ "message": "Registered successfully" }`
	Response headers
	`content-type: application/json; charset=utf-8`
	`date: Sun,07 Sep 2025 13:26:23 GMT`
	`server: Kestrel`
	`transfer-encoding: chunked`

Responses

Code	Description
200	OK

/User/login

Curl

```
curl -X 'POST' \
  'http://localhost:5189/api/User/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "admin",
  "password": "admin123"
}'
```

Request URL

```
http://localhost:5189/api/User/login
```

Server response

Code	Details
200	Response body


```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZWlhcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L21kZW50aXR5L2NsYWltcy9uYWllaWRlbnRpZmllciI6IjEiLCJodHRwOi8vc2NoZWlhcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYWllIjoiYWRtaW4iLCJodHRwOi8vc2NoZWlhcy5taWNyb3NvZnQuY29tL3dzLzIwMDgvMDYvaWRlbnRpdHkvY2xhaW1zL3JvbGUiOiJBZGlpbiIsInVzZXJuYW1lIjoiYWRtaW4iLCJyb2xlIjoiQWRtaW4iLCJleHAiOjE3NTcyNTkzNzgsInlzcyIGIkZyYWN0byIsImFlZCIGIkZyYWN0byIsImFlZCIGIkZyYWN0b1VzZXJzIn0.5UQvH2BiSQZVe8bEZGT6BK5zoTtuXySHTNn5ECpHuy4",
  "username": "admin",
  "role": "Admin",
  "profileImage": null
}
```


Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sun,07 Sep 2025 15:21:18 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Request URL

```
http://localhost:5189/api/User/all
```

Server response

Code	Details
200	Response body


```
[
  {
    "userId": 1,
    "username": "admin",
    "role": "Admin",
    "profileImagePath": null
  },
  {
    "userId": 2,
    "username": "siva",
    "role": "User",
    "profileImagePath": "/uploads/fa378000-b2b9-4b66-818c-9d2b69ccd525_aisha.jpg"
  },
  {
    "userId": 11,
    "username": "Maanik",
    "role": "User",
    "profileImagePath": "/uploads/a2307b76-c78e-4fac-96f3-47fc217fef1c_p3.jpg"
  },
  {
    "userId": 12,
    "username": "user1",
    "role": "User",
    "profileImagePath": "/uploads/880249b7-2c7c-470d-b3cf-3cc0438277b7_sneha.jpg"
  },
  {
    "userId": 15,
```

Put /api/User/{id}

Curl

```
curl -X 'PUT' \
  'http://localhost:5189/api/User/11' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L21kZW50aXR5L2NsYWltcy9uYW11aWR1bnRpZmllciI6IjEiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8
  -H 'Content-Type: application/json' \
  -d '{
  "userId": 11,
  "username": "anvitha",
  "passwordHash": "anvitha",
  "role": "User",
  "profileImagePath": "null"
}'
```

Request URL

```
http://localhost:5189/api/User/11
```

Server response

Code	Details
200	Response body

```
{
  "message": "User updated successfully"
}
```

Download

/User/upload-profile-image

file

string($binary) Choose File sneha.jpg

Send empty value

Execute

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5189/api/User/upload-profile-image' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sneha.jpg;type=image/jpeg'
```

Request URL

```
http://localhost:5189/api/User/upload-profile-image
```

Server response

Code	Details
200	Response body

```
{
  "imagePath": "/uploads/bc647355-0365-48d4-802f-c7c3687e5e79_sneha.jpg"
}
```

DELETE/api/User/{id}

Name	Description
id * required integer($int32) (path)	16

Execute

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5189/api/User/16' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2
```

Request URL

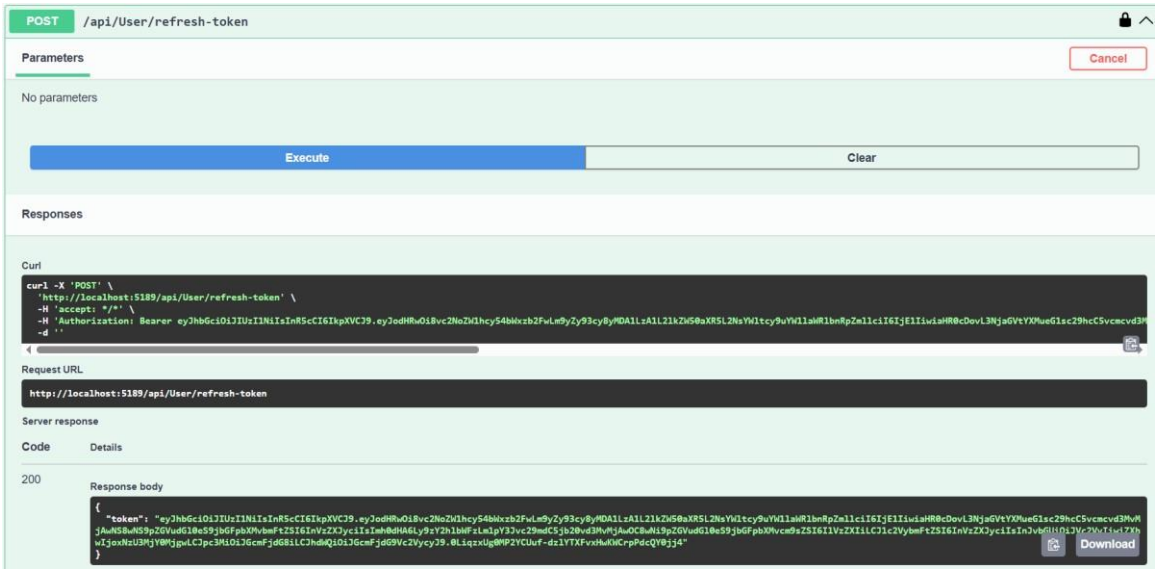
```
http://localhost:5189/api/User/16
```

Server response

Code	Details
200	Response body

```
{
  "message": "User deleted successfully"
}
```

User/refresh-token



9. Use Case Descriptions :User Stories (User Role)

1. **User Authentication (Login, Logout, Register)**

    - **Description**: A user registers with username, password, and role (default User). On login, the user gets a JWT token to access secured APIs. Logout clears the session.

    - **Actors**: User, System.

Outcome: Secure access to system features.

2. **City Selection**

    - **Description**: The user selects a city to search for doctors available in that city.

    - **Actors**: User.

    - **Outcome**: Filters doctors by location.

3. **Appointment Date Selection** ○ **Description**: The user chooses a date for booking appointments. ○ **Actors**: User.

    - **Outcome**: Narrow down doctors available on that date.

4. **Specialist Selection**

    - **Description**: User selects a specialization (Cardiology, Dermatology, etc.) from the list.

- **Actors**: User, System (loads specializations).
- **Outcome**: Filtered doctors based on specialization.

5. **View Available Doctors**
   - **Description**: System displays a list of doctors based on selected city, specialization, and date. ○ **Actors**: User, System.
   - **Outcome**: User can see available doctors.
6. **Filter by Ratings** ○ **Description**: User filters doctors by minimum rating (1–5). ○ **Actors**: User, System.
   - **Outcome**: Sorted list of quality doctors.
7. **Doctor Selection** ○ **Description**: User selects a doctor from the displayed list. ○ **Actors**: User.
   - **Outcome**: Doctor details displayed, ready for appointment booking.
8. **View Time Slots**
   - **Description**: Once a doctor is selected, available time slots for that date are shown.
   - **Actors**: User, System.
   - **Outcome**: User can choose a convenient slot.
9. **Book Appointment** ○ **Description**: User books an appointment with doctor (date + slot). ○ **Actors**: User, System, Database.
   - **Outcome**: Appointment record created in DB.
10. **Receive Confirmation**
    - **Description**: User receives confirmation (via UI or notification). ○ **Actors**: User, System.
    - **Outcome**: Booking success acknowledged.
11. **Cancel Appointment** ○ **Description**: User can cancel an existing appointment.
    - **Actors**: User, System.

- **Outcome**: Appointment marked as cancelled in DB.

12. **Rate and Review Doctor** ○ **Description**: After completed appointment, user can rate and leave review. ○ **Actors**: User, System.
    - **Outcome**: Rating updates doctor's average score.

Admin Stories (Admin Role)

1. **Admin Authentication** ○ **Description**: Admin logs in using credentials, receives JWT token. ○ **Actors**: Admin, System.
    - **Outcome**: Secure access to admin panel.
2. **CRUD on Users** ○ **Description**: Admin can add, update, delete, and view users. ○ **Actors**: Admin, System, Database.
    - **Outcome**: User management.
3. **Manage Appointments**
    - **Description**: Admin can approve, reject, or mark appointments as completed.
    - **Actors**: Admin, System. ○ **Outcome**: Appointment workflow managed.
4. **Send Confirmation** ○ **Description**: Admin confirms booking approval/rejection. ○ **Actors**: Admin, System, Notification Service (optional SignalR).
    - **Outcome**: User notified about appointment status.
5. **Cancel Appointments** ○ **Description**: Admin can cancel scheduled appointments.
    - **Actors**: Admin, System.
    - **Outcome**: Appointment removed/marked cancelled.

6. **Manage Doctors & Specializations** ○ **Description**: Admin can add/edit/delete doctors and specializations. ○ **Actors**: Admin, System, Database. ○ **Outcome**: Doctor catalog managed.

Output Snapshots:

API EndPoints & Angular User Interface Outputs **10.Backend**

Doctor Endpoints: CRUD Operations

GET/api/Doctor:

Curl

```
curl -X 'GET' \
  'http://localhost:5189/api/Doctor' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA
```

Request URL

```
http://localhost:5189/api/Doctor
```

Server response

Code	Details
200	Response body


```
[
  {
    "doctorId": 1,
    "name": "Dr. Smith",
    "city": "New York",
    "rating": 5,
    "specialization": {
      "specializationId": 1,
      "specializationName": "Cardiology"
    }
  },
  {
    "doctorId": 9,
    "name": "Dr. Yaswanth",
    "city": "Hyderabad",
    "rating": 4,
    "specialization": {
      "specializationId": 2,
      "specializationName": "Dermatology"
    }
  },
  {
    "doctorId": 13,
    "name": "Dr.Mathews",
    "city": "New York",
    "rating": 5,
    "specialization": {
```

PUT/api/Doctor/{id}

Curl

```
curl -X 'PUT' \
  'http://localhost:5189/api/Doctor/9' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bW
  -H 'Content-Type: application/json' \
  -d '{
  "doctorId": 9,
  "name": "Dr.Yaswanth",
  "city": "Hyderabad",
  "specializationId": 2,
  "rating": 4
}'
```

Request URL

```
http://localhost:5189/api/Doctor/9
```

Server response

Code	Details
200	Response body

```
{
  "doctorId": 9,
  "name": "Dr.Yaswanth",
  "city": "Hyderabad",
  "specializationId": 2,
  "rating": 4,
  "specialization": null
}
```

DELETE/api/Doctor/{id}

Name	Description
id * required integer($int32) (path)	9

Execute

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5189/api/Doctor/9' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWx
```

Request URL

```
http://localhost:5189/api/Doctor/9
```

Server response

Code	Details
200	Response body

```
{
  "message": "Doctor deleted"
}
```

POST/api/Doctor

Curl

```
curl -X 'POST' \
  'http://localhost:5189/api/Doctor' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Paul",
  "city": "Hyderabad",
  "specializationId": 1,
  "rating": 2
}'
```

Request URL

```
http://localhost:5189/api/Doctor
```

Server response

Code	Details
200	Response body

```
{
  "doctorId": 24,
  "name": "Paul",
  "city": "Hyderabad",
  "specializationId": 1,
  "rating": 2,
  "specialization": null
}
```

Backend Appointment Endpoints: CRUD Operations
POST/Appointment/book

Request URL

```
http://localhost:5189/api/Appointment/book
```

Server response

Code	Details
200	Response body


```
{
  "message": "Appointment booked successfully",
  "appointment": {
    "appointmentId": 28,
    "doctorId": 1,
    "doctor": {
      "doctorId": 1,
      "name": "Dr. Smith",
      "city": "New York",
      "specializationId": 1,
      "rating": 5,
      "specialization": null
    },
    "userId": 15,
    "user": null,
    "appointmentDate": "2025-09-07T15:44:36.24Z",
    "timeSlot": "10-11",
    "status": "Pending"
  }
}
```

Get/api/Appointment/my

```
curl -X 'GET' \
  'http://localhost:5189/api/Appointment/my' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYW1tcy9uYW1laWR1bnRpZmllciI6IjE1IiwiaHR0cDovL3Nj
```

Request URL

```
http://localhost:5189/api/Appointment/my
```

Server response

Code	Details
200	Response body


```
[
  {
    "appointmentId": 25,
    "doctorId": 13,
    "doctor": {
      "doctorId": 13,
      "name": "Dr.Mathews",
      "city": "New York",
      "specializtionId": 14,
      "rating": 5,
      "specialization": {
        "specializationId": 14,
        "specializationName": "Surgeon",
        "doctors": [
          null
        ]
      }
    },
    "userId": 15,
    "user": null,
    "appointmentDate": "2025-09-13T00:00:00",
    "timeSlot": "14:00-14:30",
    "status": "Pending"
  },
  {
    "appointmentId": 26,
    "doctorId": 15,
```


Response headers

DELETE/api/Appointment/{id}

id * required

integer($int32)

(path)

25

Execute

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5189/api/Appointment/25' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1
```

Request URL

```
http://localhost:5189/api/Appointment/25
```

Server response

Code	Details
200	Response body

```
{
  "message": "Appointment cancelled"
}
```

Backend Rating Endpoints: POST Reviews Only after appointment—

POST/api/Rating

Curl

```
curl -X 'POST' \
  'http://localhost:5189/api/Rating' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1
  -H 'Content-Type: application/json' \
  -d '{
  "doctorId": 1,
  "userId": 15,
  "ratingValue": 5,
  "comment": "excellent work!!"
}'
```

Request URL

```
http://localhost:5189/api/Rating
```

Server response

Code	Details
200	Response body

```
{
  "message": "Rating added successfully",
  "newAverage": 5
}
```

GET/api/Rating/doctor/{doctorId}

Curl

```
curl -X 'GET' \
  'http://localhost:5189/api/Rating/doctor/1' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2
```

Request URL

```
http://localhost:5189/api/Rating/doctor/1
```

Server response

Code	Details
200	Response body


```
{
  "averageRating": 5,
  "ratings": [
    {
      "ratingId": 1,
      "ratingValue": 5,
      "comment": "Great doctor, very helpful!",
      "userName": "siva"
    },
    {
      "ratingId": 2,
      "ratingValue": 5,
      "comment": "very Useful!",
      "userName": "siva"
    },
    {
      "ratingId": 3,
      "ratingValue": 5,
      "comment": "excellent work!!",
      "userName": "userrr"
    }
  ]
}
```


Backend Booking Endpoint:Available Slots

GET/api/Booking/doctor/{doctorId}/available-slots

Curl

```
curl -X 'GET' \
  'http://localhost:5189/api/Booking/doctor/1/available-slots' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L21kZW50aXR5L2NsYW1tcy9uYW11aW
```

Request URL

```
http://localhost:5189/api/Booking/doctor/1/available-slots
```

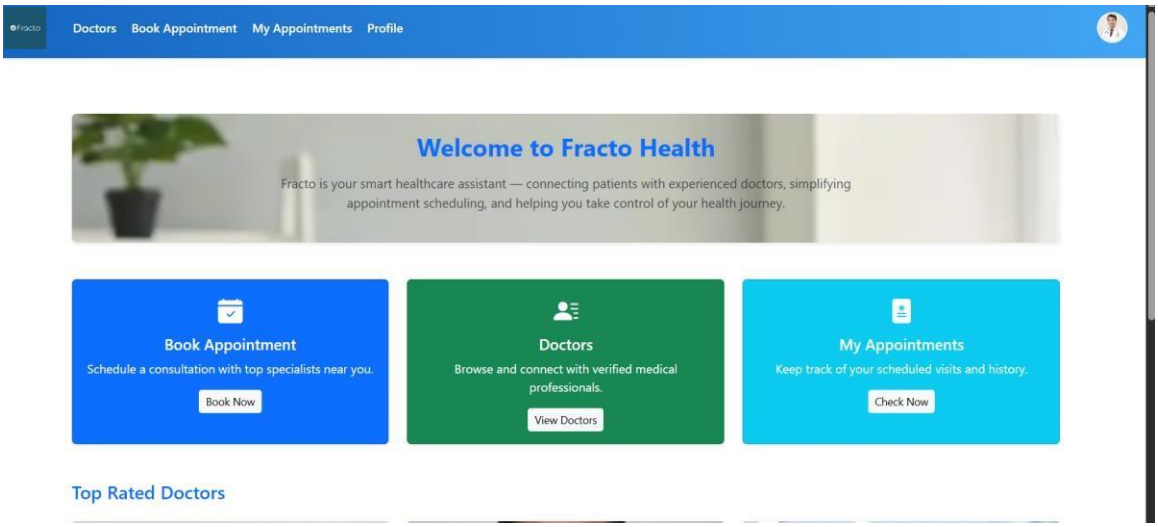
Server response

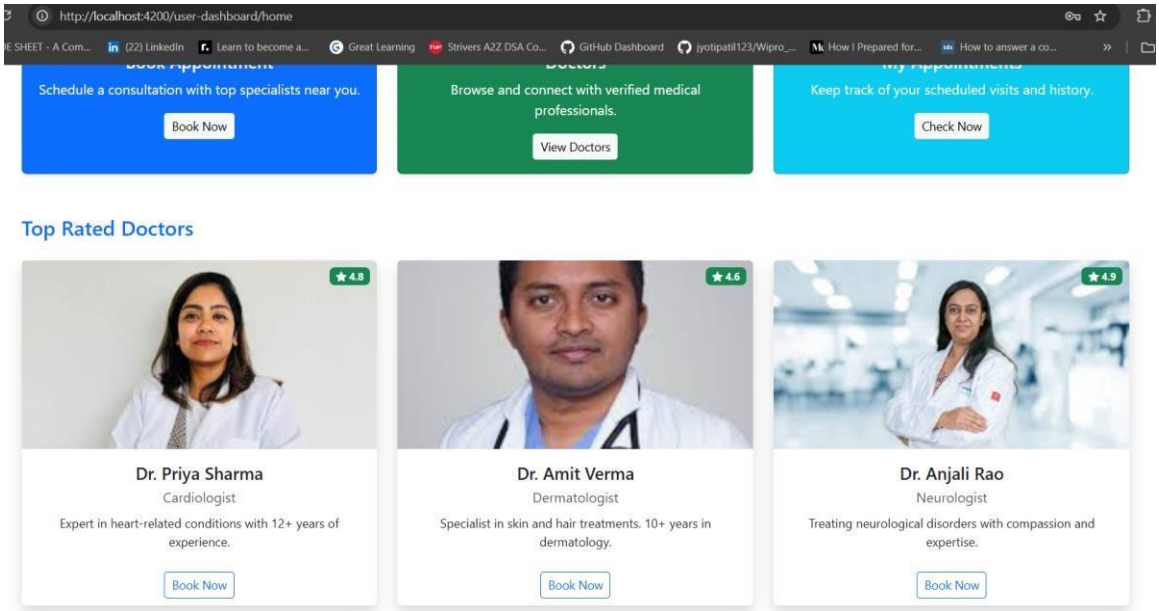
Code	Details
200	Response body

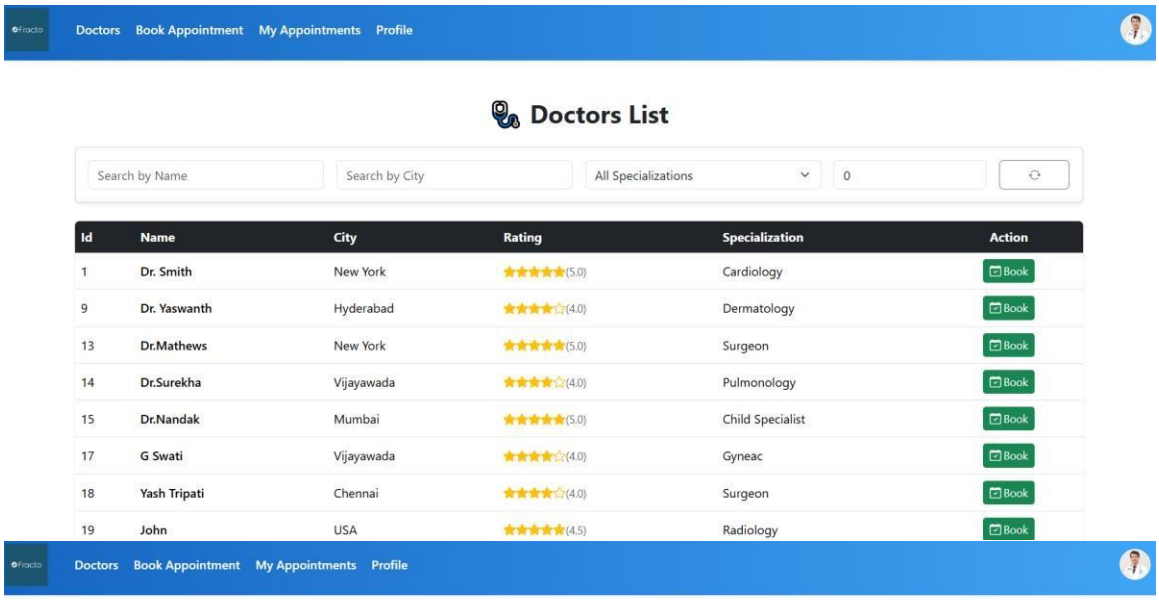
```
[
  "09:00-09:30",
  "09:30-10:00",
  "10:00-10:30",
  "10:30-11:00",
  "11:00-11:30",
  "11:30-12:00",
  "12:00-12:30",
  "12:30-13:00",
  "13:00-13:30",
  "13:30-14:00",
  "14:00-14:30",
  "14:30-15:00",
  "15:00-15:30",
  "15:30-16:00",
  "16:00-16:30",
  "16:30-17:00"
]
```

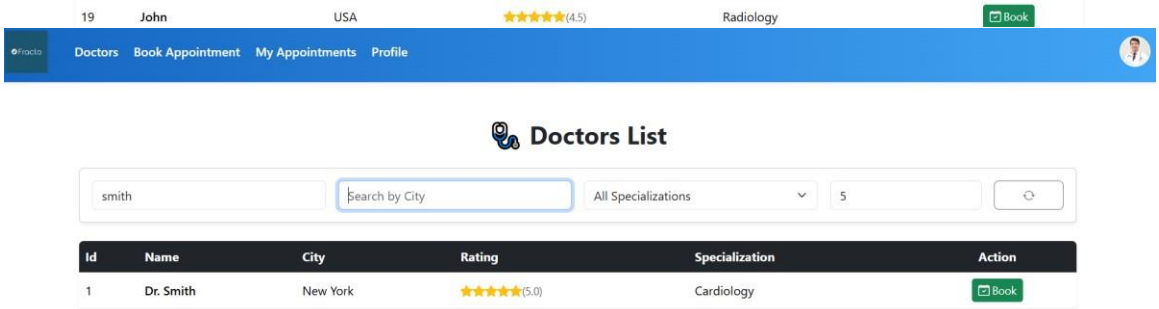
10. Output Snapshots Of Angular(Admin+User UI):

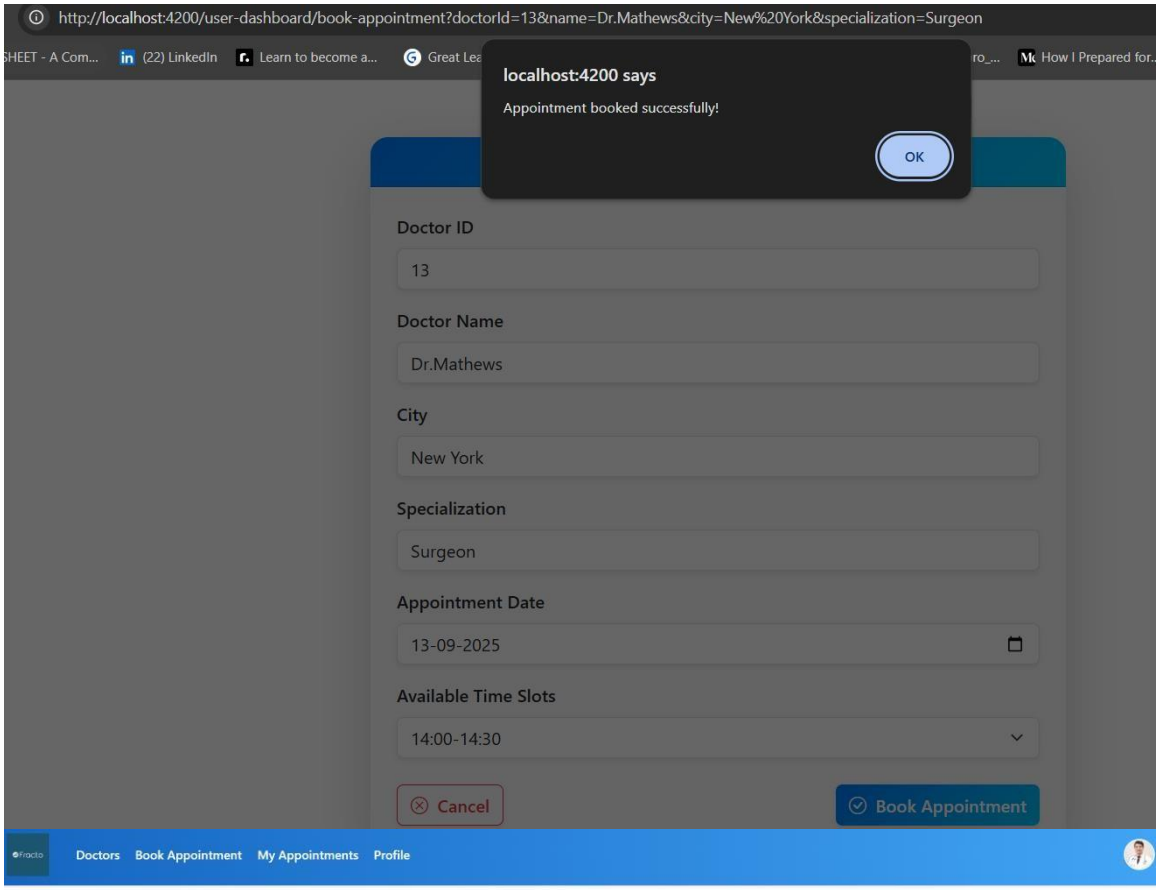
USER :

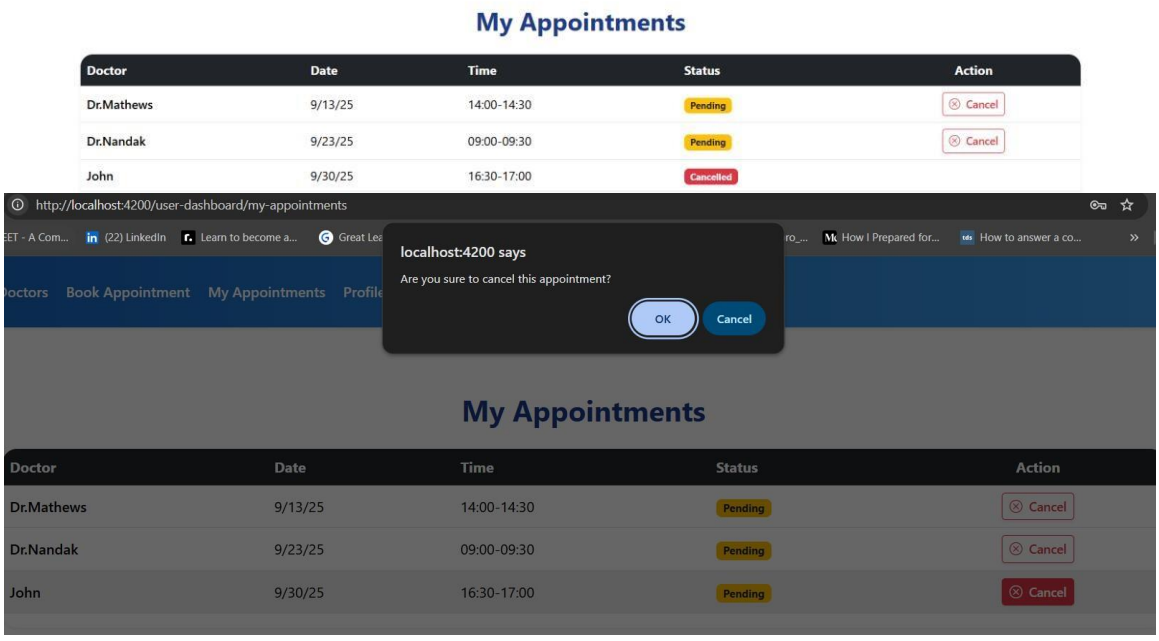


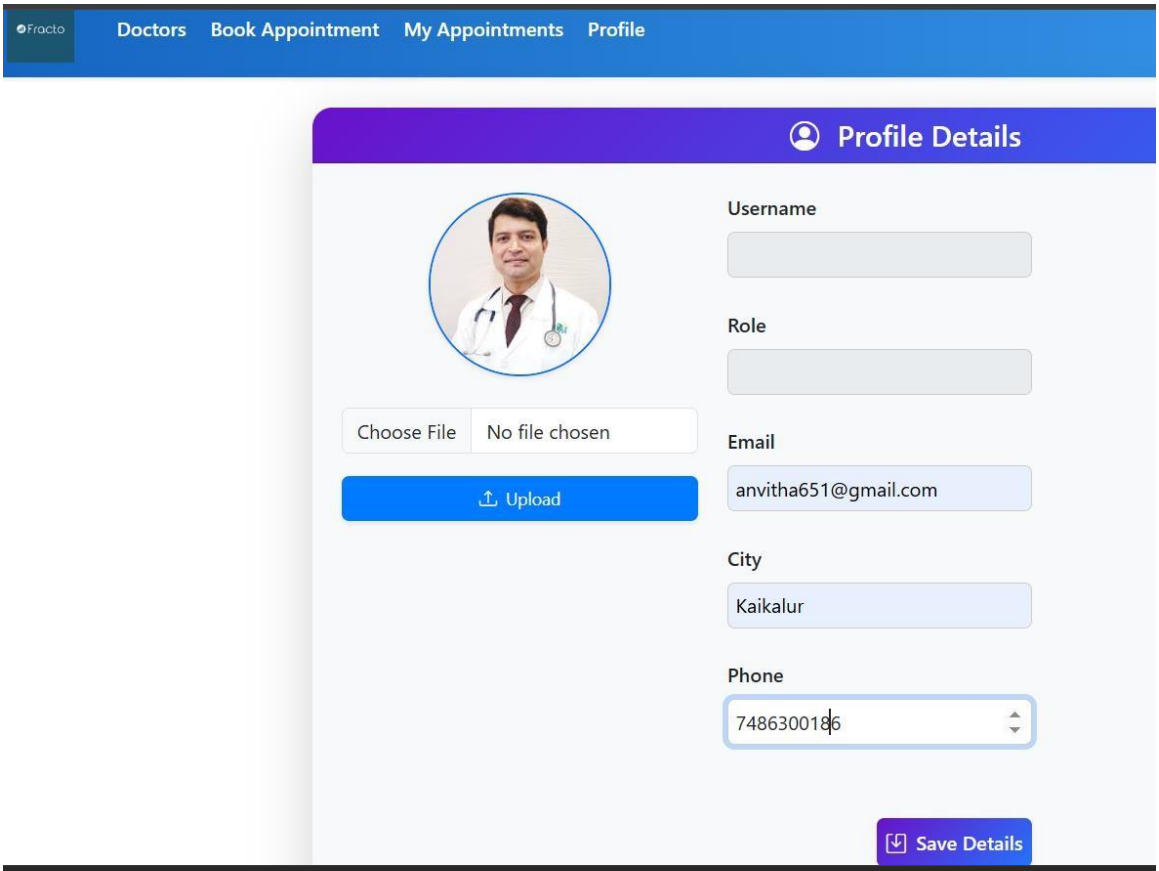

http://localhost:4200/user-dashboard/home
Schedule a consultation with top specialists near you.
Book Now
Browse and connect with verified medical professionals.
View Doctors
Keep track of your scheduled visits and history.
Check Now
Top Rated Doctors
4.8
Dr. Priya Sharma
Cardiologist
Expert in heart-related conditions with 12+ years of experience.
Book Now
4.6
Dr. Amit Verma
Dermatologist
Specialist in skin and hair treatments. 10+ years in dermatology.
Book Now
4.9
Dr. Anjali Rao
Neurologist
Treating neurological disorders with compassion and expertise.
Book Now

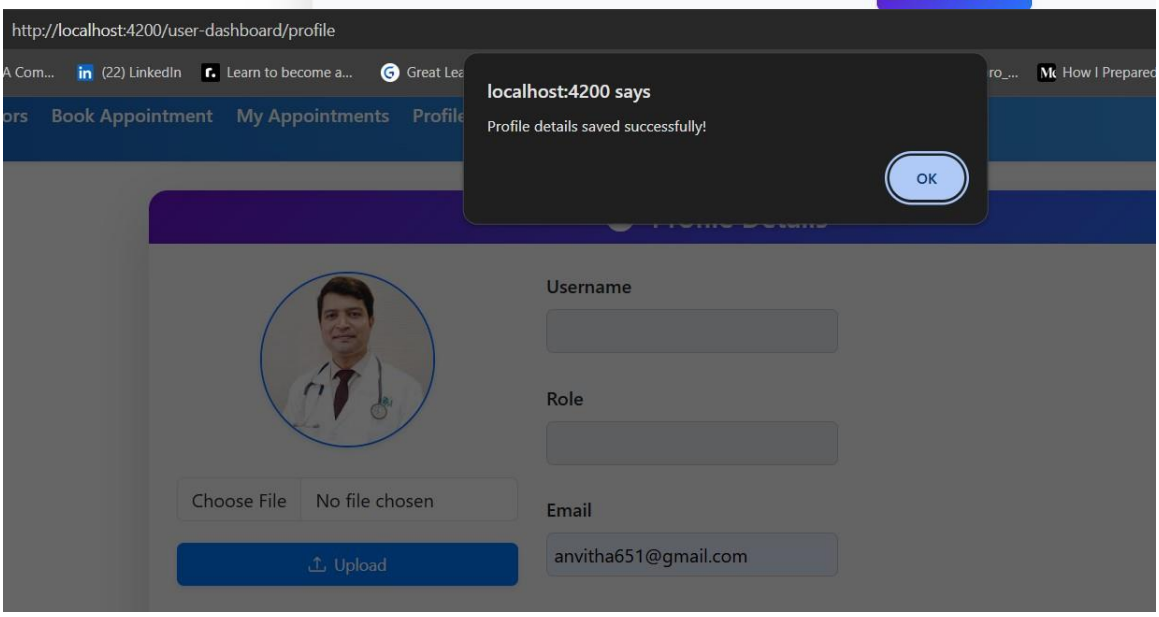

Doctors
Book Appointment
My Appointments
Profile
Doctors List
Search by Name
Search by City
All Specializations
0
Id Name City Rating Specialization Action
1 Dr. Smith New York (5.0) Cardiology Book
9 Dr. Yaswanth Hyderabad (4.0) Dermatology Book
13 Dr.Mathews New York (5.0) Surgeon Book
14 Dr.Surekha Vijayawada (4.0) Pulmonology Book
15 Dr.Nandak Mumbai (5.0) Child Specialist Book
17 G Swati Vijayawada (4.0) Gyneac Book
18 Yash Tripati Chennai (4.0) Surgeon Book
19 John USA (4.5) Radiology Book


Doctors
Book Appointment
My Appointments
Profile
Doctors List
smith
Search by City
All Specializations
5
Id Name City Rating Specialization Action
1 Dr. Smith New York (5.0) Cardiology Book


http://localhost:4200/user-dashboard/book-appointment?doctorId=13&name=Dr.Mathews&city=New%20York&specialization=Surgeon
localhost:4200 says
Appointment booked successfully!
OK
Doctor ID
13
Doctor Name
Dr.Mathews
City
New York
Specialization
Surgeon
Appointment Date
13-09-2025
Available Time Slots
14:00-14:30
Cancel
Book Appointment
Doctors
Book Appointment
My Appointments
Profile


My Appointments
Doctor
Date
Time
Status
Action
Dr.Mathews
9/13/25
14:00-14:30
Pending
Cancel
Dr.Nandak
9/23/25
09:00-09:30
Pending
Cancel
John
9/30/25
16:30-17:00
Cancelled
http://localhost:4200/user-dashboard/my-appointments
localhost:4200 says
Are you sure to cancel this appointment?
OK
Cancel
My Appointments
Doctor
Date
Time
Status
Action
Dr.Mathews
9/13/25
14:00-14:30
Pending
Cancel
Dr.Nandak
9/23/25
09:00-09:30
Pending
Cancel
John
9/30/25
16:30-17:00
Pending
Cancel


Fracto
Doctors
Book Appointment
My Appointments
Profile
Profile Details
Choose File
No file chosen
Upload
Username
Role
Email
anvitha651@gmail.com
City
Kaikalur
Phone
7486300186
Save Details


http://localhost:4200/user-dashboard/profile
localhost:4200 says
Profile details saved successfully!
OK
Book Appointment
My Appointments
Choose File
No file chosen
Upload
Username
Role
Email
anvitha651@gmail.com

ADMIN UI:

Fracto | Users | Doctors | Add Doctor | Specializations | Appointments | Welcome, Admin | Logout

Manage Users

Add New User

ID	Username	Role	Profile	Actions
1	admin	Admin		Edit Delete
2	siva	User		Edit Delete
5	Anvitha guduru	User		Edit Delete
11	Maanik	User		Edit Delete
12	user1	User		Edit Delete

Add User

Username

newUser

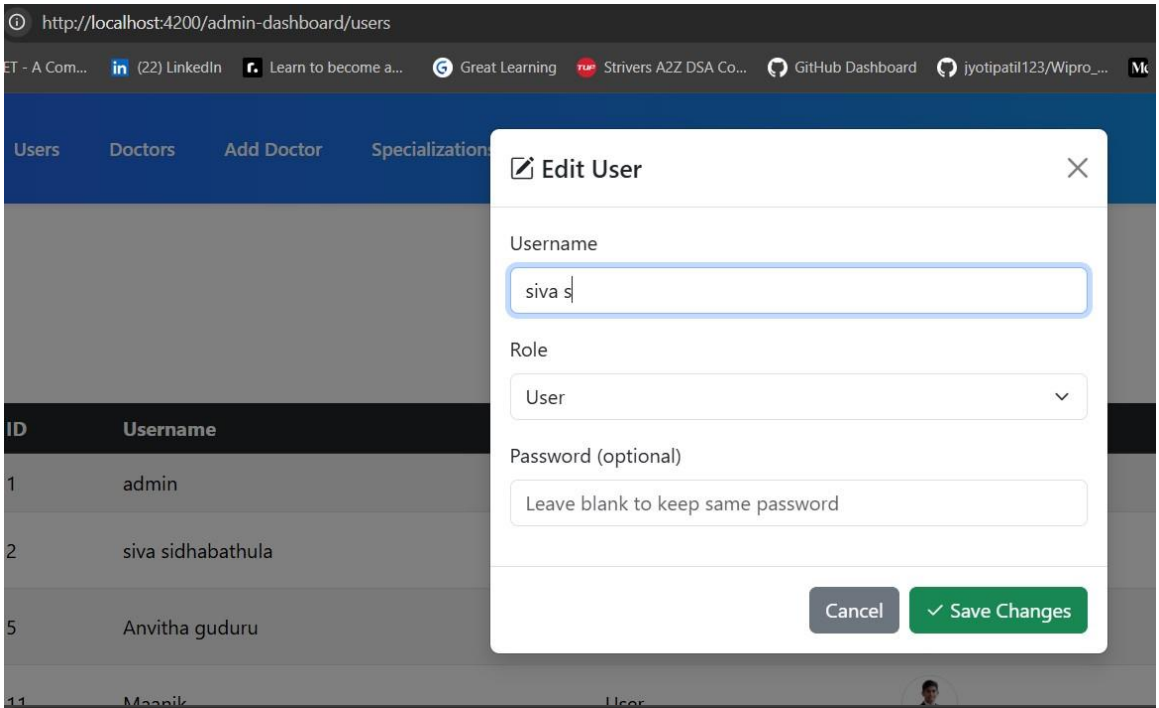
Password

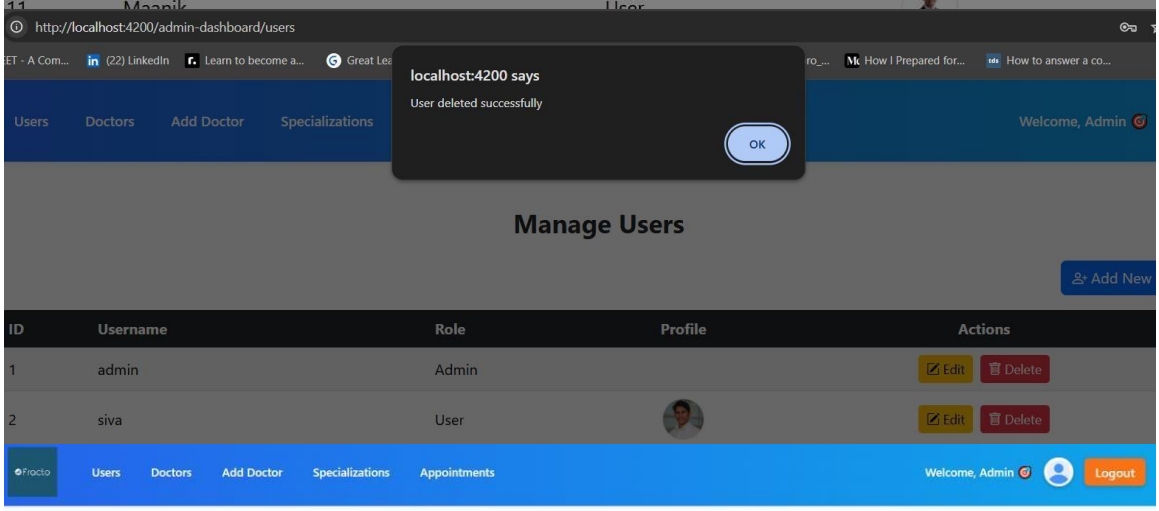
•••••••

Role

User

Cancel | Add User


http://localhost:4200/admin-dashboard/users
Users
Doctors
Add Doctor
Specializations
Edit User
Username
siva s
Role
User
Password (optional)
Leave blank to keep same password
Cancel
Save Changes
ID
Username
1
admin
2
siva sidhabathula
5
Anvitha guduru


http://localhost:4200/admin-dashboard/users
localhost:4200 says
User deleted successfully
OK
Users
Doctors
Add Doctor
Specializations
Welcome, Admin
Manage Users
Add New
ID
Username
Role
Profile
Actions
1
admin
Admin
Edit
Delete
2
siva
User
Edit
Delete
Users
Doctors
Add Doctor
Specializations
Appointments
Welcome, Admin
Logout


Doctors List
Search by Name
Search by City
All Specializations
0
Id
Name
City
Rating
Specialization
Actions
1
Dr. Smith
New York
(5.0)
Cardiology
9
Dr. Yaswanth
Hyderabad
(4.0)
Dermatology
13
Dr.Mathews
New York
(5.0)
Surgeon
14
Dr.Surekha
Vijayawada
(4.0)
Pulmonology
15
Dr.Nandak
Mumbai
(5.0)
Child Specialist
17
G Swati
Vijayawada
(4.0)
Gyneac

Doctors List

ya | Search by City | All Specializations | 4

Id	Name	City	Rating	Specialization	Actions
9	Dr. Yaswanth	Hyderabad	★★★★☆(4.0)	Dermatology	
18	Yash Tripati	Chennai	★★★★☆(4.0)	Surgeon	

http://localhost:4200/admin-dashboard/add-doctor

localhost:4200 says

✅ Doctor added successfully!

OK

Full Name

Yash Tripati

City

Chennai

Rating

4

Rating should be between 0.0 and 5.0

Specialization

Surgeon

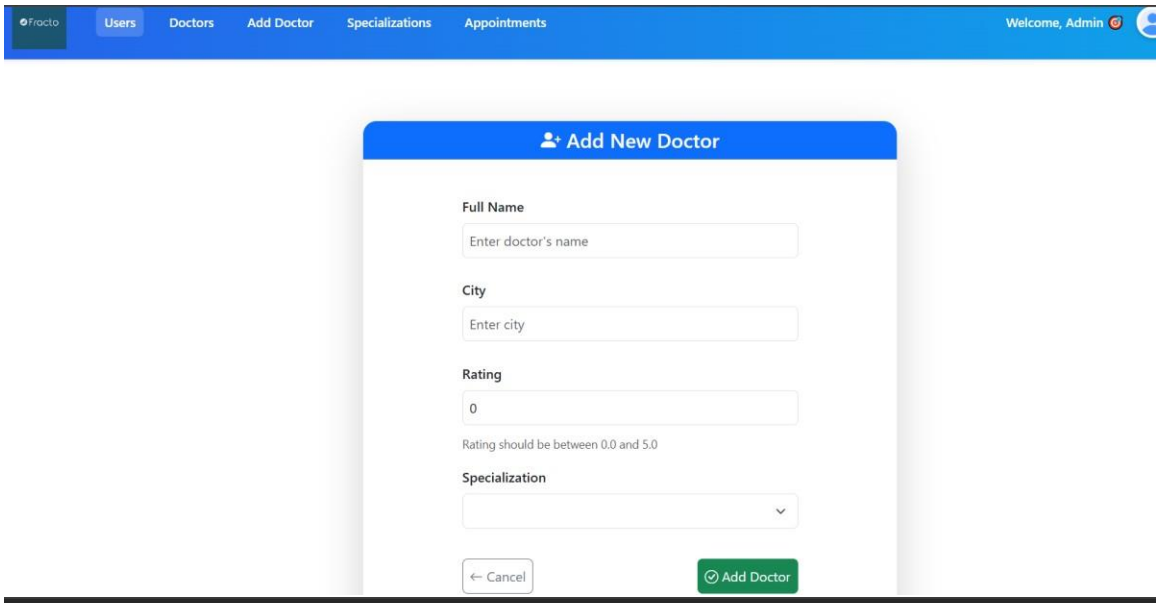
← Cancel

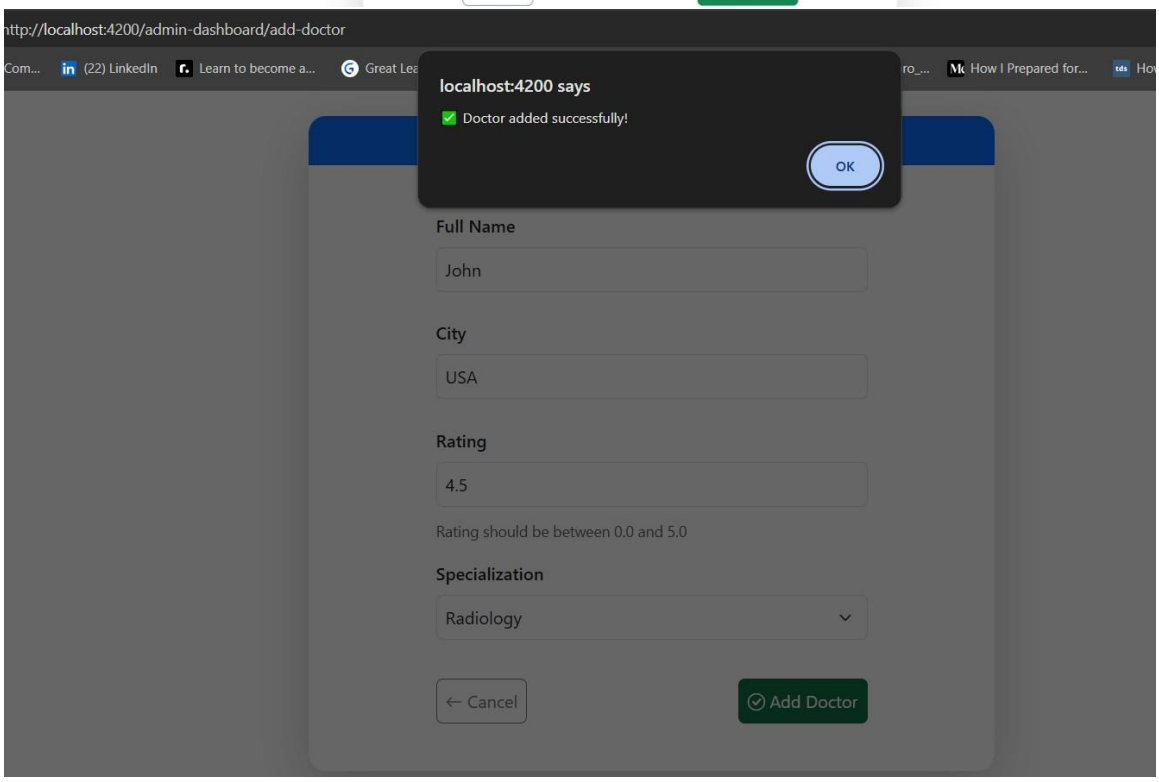
Add Doctor

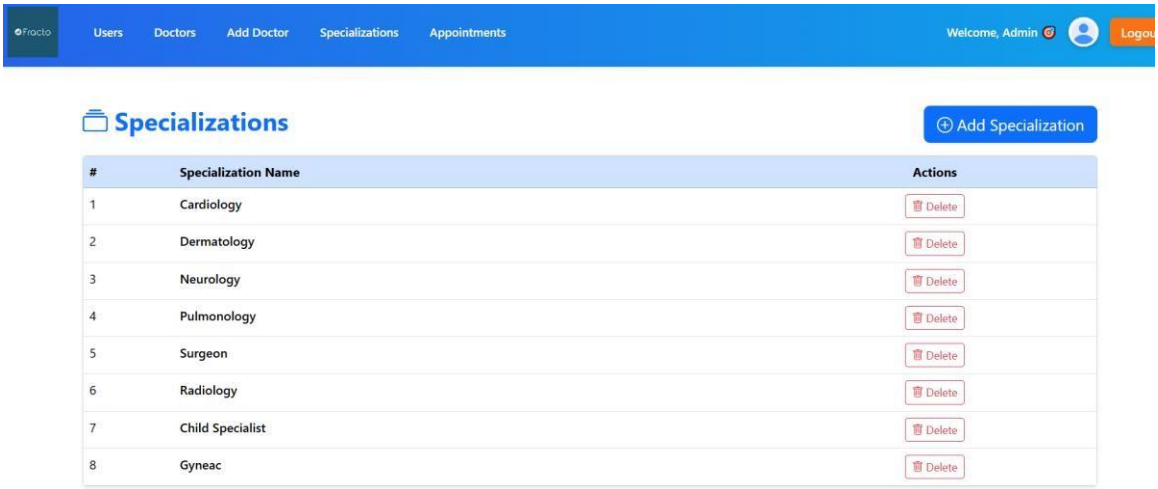
Doctors List

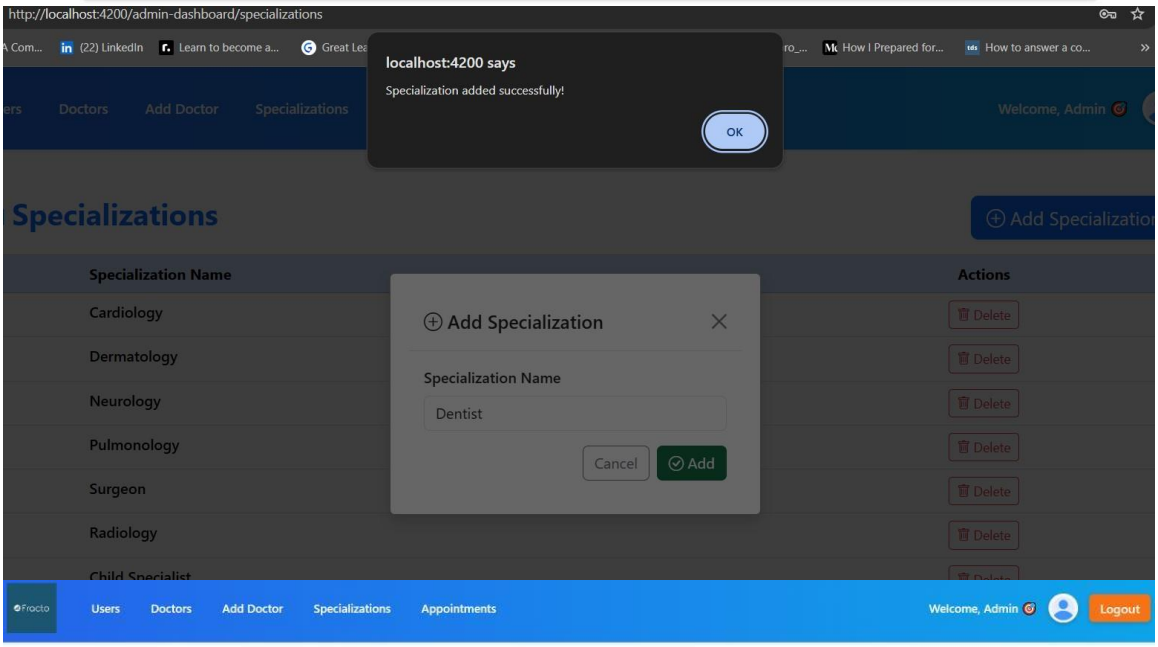
Search by Name | new | All Specializations | 0

Id	Name	City	Rating	Specialization	Actions
1	Dr. Smith	New York	★★★★★(5.0)	Cardiology	
13	Dr.Mathews	New York	★★★★★(5.0)	Surgeon	


Fracto
Users
Doctors
Add Doctor
Specializations
Appointments
Welcome, Admin
Add New Doctor
Full Name
Enter doctor's name
City
Enter city
Rating
0
Rating should be between 0.0 and 5.0
Specialization
← Cancel
Add Doctor


http://localhost:4200/admin-dashboard/add-doctor
localhost:4200 says
Doctor added successfully!
OK
Full Name
John
City
USA
Rating
4.5
Rating should be between 0.0 and 5.0
Specialization
Radiology
← Cancel
Add Doctor





11. Conclusion

The Fracto project successfully implements an online doctor appointment booking system with secure authentication, appointment management, and a user-friendly interface. Future enhancements could include payment integration, real-time notifications via SignalR, and mobile app support.