FRACTO

Doctor Appointment System

CHILUMURI DHARANI

**Batch Name:** .Net Full Stack Angular - FY26 - C2

**Trainer:** Jyoti Patil

### **Table of Contents**

1. Problem Definition and Objectives
2. Frontend & Backend Architecture
3. Component Breakdown & API Design
4. Database Design & Storage Optimization
5. Detailed Demonstration

**1. Problem Definition and Objectives**

**1.1 Problem Statement**

Fracto is an online doctor appointment booking platform designed to simplify the process of finding doctors, booking appointments, and managing schedules.

**1.2 Project Goals and Objectives**

* Provide a user-friendly interface for patients to search doctors by city, specialization, and available time slots.
* Enable appointment booking and cancellation with real-time updates.
* Allow administrators to manage doctor profiles, appointments, and users.
* Ensure secure user authentication and role-based access control.
* Implement an optimized database for efficient querying and data management.

**2. Frontend & Backend Architecture**

### **2.1 Technology Stack**

* Frontend: Angular (TypeScript, HTML, CSS, Bootstrap)
* Backend: ASP.NET Core MVC with Web API
* Database: SQL Server using Entity Framework Core
* Authentication: JWT (JSON Web Tokens) for secure login

### **2.2 System Architecture**

**Overview:**

* Users (Patients) interact with the Angular frontend.
* Angular makes **HTTP requests to ASP.NET Core Web API** for CRUD operations.
* Web API interacts with SQL Server via **Entity Framework Core**.
* JWT handles authentication and authorization.

**System Diagram:**

**[User (Patient/Admin)]**

**|**

**v**

**[Angular Frontend]**

**|**

**v**

**[ASP.NET Core Web API (Controllers & Services)]**

**|**

**v**

**[SQL Server Database]**

**3. Component Breakdown & API Design**

### **3.1 Frontend Components**

* State Management: Angular services and RxJS observables to manage data across components.
* Routing: Angular Router for navigation between Home, Doctors List, Appointment Booking, Profile pages.
* UI Components:  
  + Header, Footer
  + Doctor Card Component
  + Appointment Form Component
  + Admin Dashboard Components

### **3.2 API Design**

**Endpoints Example:**

* POST /api/Auth/register → User registration
* POST /api/Auth/login → User login
* GET /api/Doctors → Fetch list of doctors
* POST /api/Appointments → Book an appointment
* DELETE /api/Appointments/{id} → Cancel appointment

**Authentication Mechanism:**

* Users login using **username/email and password**.
* Backend validates credentials and generates **JWT token**.
* Angular frontend stores a token in **localStorage** and sends it in Authorization header for secure API requests.

## **4. Database Design & Storage Optimization**

### **4.1 Entity-Relationship Diagram (ERD)**

**Relationships:**

* **User → Appointment**: One-to-Many
* **Doctor → Appointment**: One-to-Many

### **4.2 Optimization Techniques**

* Use **indexes** on frequently searched fields (Doctor.Specialization, Doctor.City).
* Implement **EF Core lazy loading and eager loading** for optimized queries.
* Normalize database to reduce redundancy.
* Cache frequently accessed data (doctor lists, available slots).

## **5. Detailed Demonstration**

* **Step 1: User Registration/Login**
  + Screenshots of registration form, login form, and JWT token generation.
* **Step 2: Doctor Search and Appointment Booking**
  + Search doctors by city/specialization.
  + Book or cancel appointment.
* **Step 3: Admin Dashboard**
  + Add/update doctor profiles.
  + View all appointments and users.
* **Step 4: Backend API Testing**
  + Swagger collection to show all API endpoints and responses.
* **Step 5: Database Verification**
  + Screenshots of SQL Server tables and sample queries for verification.

**Frontend code:**

https://drive.google.com/file/d/1tmFTtvp6A5MaA2wTfYKQteTvviD9zwb9/view?usp=drive\_link

**Backend code:**

<https://drive.google.com/file/d/1S8FKSLXqfUO7t3SZYuHbUx2aTdceYNkq/view?usp=drive_link>

**Database code:**

https://drive.google.com/file/d/13fMG\_6TZUnMF-3P96EcdhNYF011wgJab/view?usp=drive\_link