**Clinivia Projet**

**Angular Part for Add Doctor with file**

**Étape 1 : Créer un Service Angular**

Créez un service pour gérer les opérations CRUD pour les docteurs.

**doctor.service.ts**

import { Injectable } from '@angular/core';

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Observable } from 'rxjs';

import { Doctor } from './doctor.model';

@Injectable({

providedIn: 'root'

})

export class DoctorService {

private baseUrl = ‘http://localhost:5126/api/Doctors’

constructor(private http: HttpClient) { }

getDoctors(): Observable<Doctor[]> {

return this.http.get<Doctor[]>(this.baseUrl);

}

createDoctor(doctor: Doctor, photo: File): Observable<Doctor> {

const formData = new FormData();

formData.append('username', doctor.username);

formData.append('email', doctor.email);

formData.append('specialization', doctor.specialization);

formData.append('departementId', doctor.departementId.toString());

if (photo) {

formData.append('photo', photo);

}

const headers = new HttpHeaders({

'Authorization': `Bearer ${localStorage.getItem('token')}`

});

return this.http.post<Doctor>(this.baseUrl, formData, { headers });

}

}

### Étape 2 : Créer le Modèle Doctor

#### doctor.model.ts

export class Doctor {

userId?: number;

username!: string;

email!: string;

specialization!: string;

photoPath?: string;

departementId!: number;

departementName?: string;

}

### Étape 3 : Créer le Composant pour Ajouter un Docteur

Créez un composant pour ajouter un nouveau docteur et gérer l'upload de l'image.

#### doctor-create.component.ts

import { Component, OnInit } from '@angular/core';

import { DoctorService } from './doctor.service';

import { Doctor } from './doctor.model';

import { DepartementService } from './departement.service';

import { Departement } from './departement.model';

@Component({

selector: 'app-doctor-create',

templateUrl: './doctor-create.component.html'

})

export class DoctorCreateComponent implements OnInit {

doctor: Doctor = new Doctor();

photo: File | null = null;

departements: Departement[] = [];

constructor(private doctorService: DoctorService, private departementService: DepartementService) { }

ngOnInit(): void {

this.departementService.getDepartements().subscribe(data => {

this.departements = data;

});

}

onFileChange(event: any): void {

this.photo = event.target.files[0];

}

createDoctor(): void {

this.doctorService.createDoctor(this.doctor, this.photo!).subscribe(data => {

console.log('Docteur créé avec succès', data);

});

}

}

### Étape 4 : Créer le Template HTML pour le Composant

Créez le fichier HTML pour le composant doctor-create.

#### doctor-create.component.html

<div class="container">

<form (ngSubmit)="createDoctor()">

<div class="form-group">

<label for="username">Username:</label>

<input type="text" id="username" [(ngModel)]="doctor.username" name="username" class="form-control" required>

</div>

<div class="form-group">

<label for="email">Email:</label>

<input type="email" id="email" [(ngModel)]="doctor.email" name="email" class="form-control" required>

</div>

<div class="form-group">

<label for="specialization">Specialization:</label>

<input type="text" id="specialization" [(ngModel)]="doctor.specialization" name="specialization" class="form-control">

</div>

<div class="form-group">

<label for="departement">Departement:</label>

<select id="departement" [(ngModel)]="doctor.departementId" name="departementId" class="form-control" required>

<option \*ngFor="let departement of departements" [value]="departement.departementId">

{{ departement.name }}

</option>

</select>

</div>

<div class="form-group">

<label for="photo">Photo:</label>

<input type="file" (change)="onFileChange($event)" class="form-control" name="photo" required>

</div>

<button type="submit" class="btn btn-primary">Create Doctor</button>

</form>

</div>

### Étape 5 : Créer le Service pour les Départements

Créez un service pour récupérer les départements.

#### departement.service.ts

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

import { Departement } from './departement.model';

@Injectable({

providedIn: 'root'

})

export class DepartementService {

private baseUrl = 'http://localhost:5126/api/Departements';

constructor(private http: HttpClient) { }

getDepartements(): Observable<Departement[]> {

return this.http.get<Departement[]>(this.baseUrl);

}

}

### Étape 6 : Créer le Modèle Departement

#### departement.model.ts

export class Departement {

departementId!: number;

name!: string;

}

### Étape 7 : Ajouter le Composant au Module

Assurez-vous que votre module Angular est configuré pour inclure le nouveau composant et les services.

#### app.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { AppComponent } from './app.component';

import { DoctorCreateComponent } from './doctor-create.component';

import { DoctorService } from './doctor.service';

import { DepartementService } from './departement.service';

@NgModule({

declarations: [

AppComponent,

DoctorCreateComponent

],

imports: [

BrowserModule,

FormsModule,

HttpClientModule

],

providers: [DoctorService, DepartementService],

bootstrap: [AppComponent]

})

export class AppModule { }