**Lab :- 9**

Navbar :-

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

@Component({

  selector: 'app-top-nav',

  templateUrl: './top-nav.component.html',

  styleUrls: ['./top-nav.component.css']

})

export class TopNavComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { TopNavComponent } from './top-nav.component';

describe('TopNavComponent', () => {

  let component: TopNavComponent;

  let fixture: ComponentFixture<TopNavComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ TopNavComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(TopNavComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});



View Student :-

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

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

import { MatSnackBar } from '@angular/material/snack-bar';

import { ActivatedRoute, Router } from '@angular/router';

import { Gender } from 'src/app/models/ui-models/gender.model';

import { Student } from 'src/app/models/ui-models/student.model';

import { GenderService } from 'src/app/services/gender.service';

import { StudentService } from '../student.service';

@Component({

  selector: 'app-view-student',

  templateUrl: './view-student.component.html',

  styleUrls: ['./view-student.component.css']

})

export class ViewStudentComponent implements OnInit {

  studentId: string | null | undefined;

  student: Student = {

    id: '',

    firstName: '',

    lastName: '',

    dateOfBirth: '',

    email: '',

    mobile: 0,

    genderId: '',

    profileImageUrl: '',

    gender: {

      id: '',

      description: ''

    },

    address: {

      id: '',

      physicalAddress: '',

      postalAddress: ''

    }

  };

  isNewStudent = false;

  header = '';

  displayProfileImageUrl = '';

  genderList: Gender[] = [];

  @ViewChild('studentDetailsForm') studentDetailsForm?: NgForm;

  constructor(private readonly *studentService*: StudentService,

    private readonly *route*: ActivatedRoute,

    private readonly *genderService*: GenderService,

    private *snackbar*: MatSnackBar,

    private *router*: Router) { }

  ngOnInit(): void {

    this.route.paramMap.subscribe(

      (*params*) => {

        this.studentId = params.get('id');

        if (this.studentId) {

          if (this.studentId.toLowerCase() === 'Add'.toLowerCase()) {

            // -> new Student Functionality

            this.isNewStudent = true;

            this.header = 'Add New Student';

            this.setImage();

          } else {

            // -> Existing Student Functionality

            this.isNewStudent = false;

            this.header = 'Edit Student';

            this.studentService.getStudent(this.studentId)

              .subscribe(

                (*successResponse*) => {

                  this.student = successResponse;

                  this.setImage();

                },

                (*errorResponse*) => {

                  this.setImage();

                }

              );

          }

          this.genderService.getGenderList()

            .subscribe(

              (*successResponse*) => {

                this.genderList = successResponse;

              }

            );

        }

      }

    );

  }

  onUpdate(): void {

    if (this.studentDetailsForm?.form.valid) {

      this.studentService.updateStudent(this.student.id, this.student)

        .subscribe(

          (*successResponse*) => {

            // Show a notification

            this.snackbar.open('Student updated successfully', undefined, {

              duration: 2000

            });

          },

          (*errorResponse*) => {

            // Log it

            console.log(errorResponse);

          }

        );

    }

  }

  onDelete(): void {

    this.studentService.deleteStudent(this.student.id)

      .subscribe(

        (*successResponse*) => {

          this.snackbar.open('Student deleted successfully', undefined, {

            duration: 2000

          });

          setTimeout(() => {

            this.router.navigateByUrl('students');

          }, 2000);

        },

        (*errorResponse*) => {

          // Log

        }

      );

  }

  onAdd(): void {

    if (this.studentDetailsForm?.form.valid) {

      // Submit form date to api

      this.studentService.addStudent(this.student)

        .subscribe(

          (*successResponse*) => {

            this.snackbar.open('Student added successfully', undefined, {

              duration: 2000

            });

            setTimeout(() => {

              this.router.navigateByUrl(`students/${successResponse.id}`);

            }, 2000);

          },

          (*errorResponse*) => {

            // Log

            console.log(errorResponse);

          }

        );

    }

  }

  uploadImage(*event*: any): void {

    if (this.studentId) {

      const file: File = event.target.files[0];

      this.studentService.uploadImage(this.student.id, file)

        .subscribe(

          (*successResponse*) => {

            this.student.profileImageUrl = successResponse;

            this.setImage();

            // Show a notification

            this.snackbar.open('Profile Image Updated', undefined, {

              duration: 2000

            });

          },

          (*errorResponse*) => {

          }

        );

    }

  }

  private setImage(): void {

    if (this.student.profileImageUrl) {

      this.displayProfileImageUrl = this.studentService.getImagePath(this.student.profileImageUrl);

    } else {

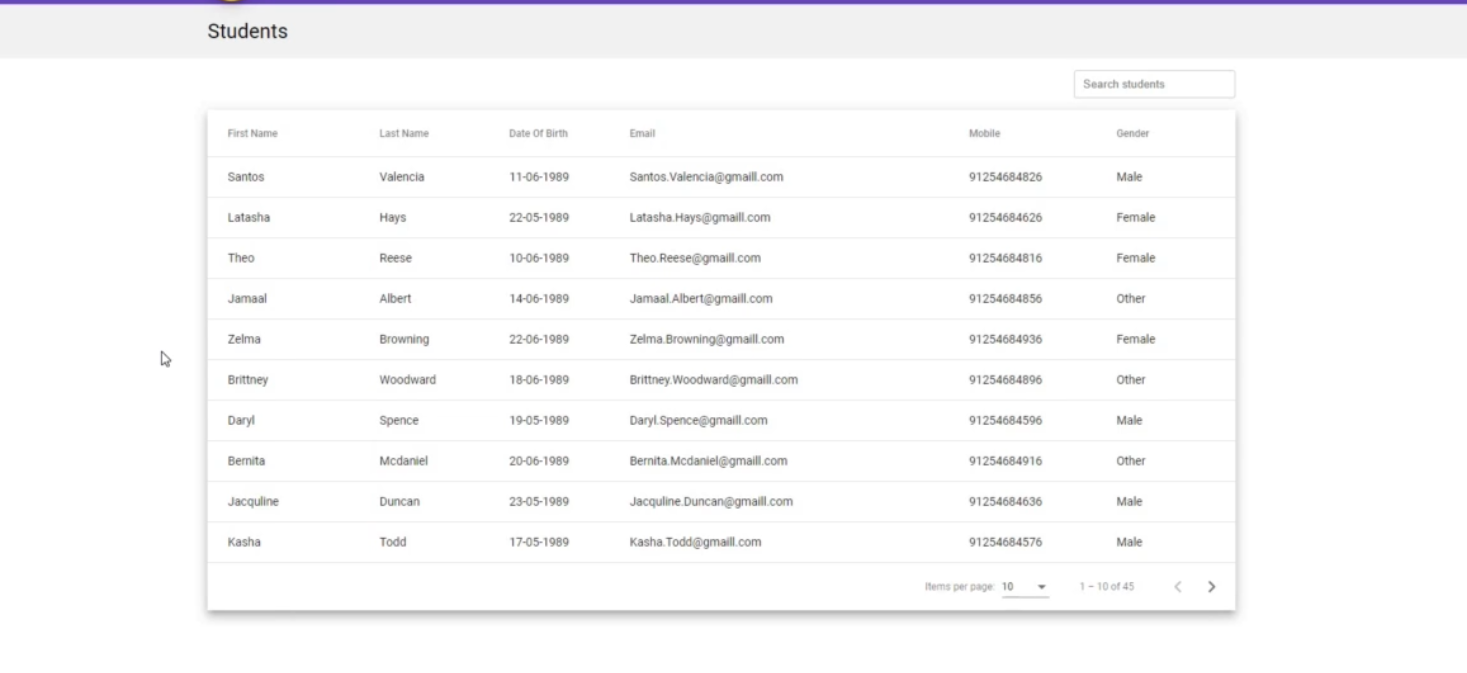
      // Display a default

      this.displayProfileImageUrl = '/assets/user.png';

    }

  }

}



Injected Student Services :-

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

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

import { Observable } from 'rxjs';

import { AddStudentRequest } from '../models/api-models/add-student-request.model';

import { Student } from '../models/api-models/student.model';

import { UpdateStudentRequest } from '../models/api-models/update-student-request.model';

@Injectable({

  providedIn: 'root'

})

export class StudentService {

  private baseApiUrl = 'https://localhost:44389';

  constructor(private *httpClient*: HttpClient) { }

  getStudents(): Observable<Student[]> {

    return this.httpClient.get<Student[]>(this.baseApiUrl + '/students');

  }

  getStudent(*studentId*: string): Observable<Student> {

    return this.httpClient.get<Student>(this.baseApiUrl + '/students/' + studentId)

  }

  updateStudent(*studentId*: string, *studentRequest*: Student): Observable<Student> {

    const updateStudentRequest: UpdateStudentRequest = {

      firstName: studentRequest.firstName,

      lastName: studentRequest.lastName,

      dateOfBirth: studentRequest.dateOfBirth,

      email: studentRequest.email,

      mobile: studentRequest.mobile,

      genderId: studentRequest.genderId,

      physicalAddress: studentRequest.address.physicalAddress,

      postalAddress: studentRequest.address.postalAddress

    }

    return this.httpClient.put<Student>(this.baseApiUrl + '/students/' + studentId, updateStudentRequest);

  }

  deleteStudent(*studentId*: string): Observable<Student> {

    return this.httpClient.delete<Student>(this.baseApiUrl + '/students/' + studentId);

  }

  addStudent(*studentRequest*: Student): Observable<Student> {

    const addStudentRequest: AddStudentRequest = {

      firstName: studentRequest.firstName,

      lastName: studentRequest.lastName,

      dateOfBirth: studentRequest.dateOfBirth,

      email: studentRequest.email,

      mobile: studentRequest.mobile,

      genderId: studentRequest.genderId,

      physicalAddress: studentRequest.address.physicalAddress,

      postalAddress: studentRequest.address.postalAddress

    };

    return this.httpClient.post<Student>(this.baseApiUrl + '/students/add', addStudentRequest);

  }

  uploadImage(*studentId*: string, *file*: File): Observable<any> {

    const formData = new FormData();

    formData.append("profileImage", file);

    return this.httpClient.post(this.baseApiUrl + '/students/' + studentId + '/upload-image',

      formData, {

      responseType: 'text'

    }

    );

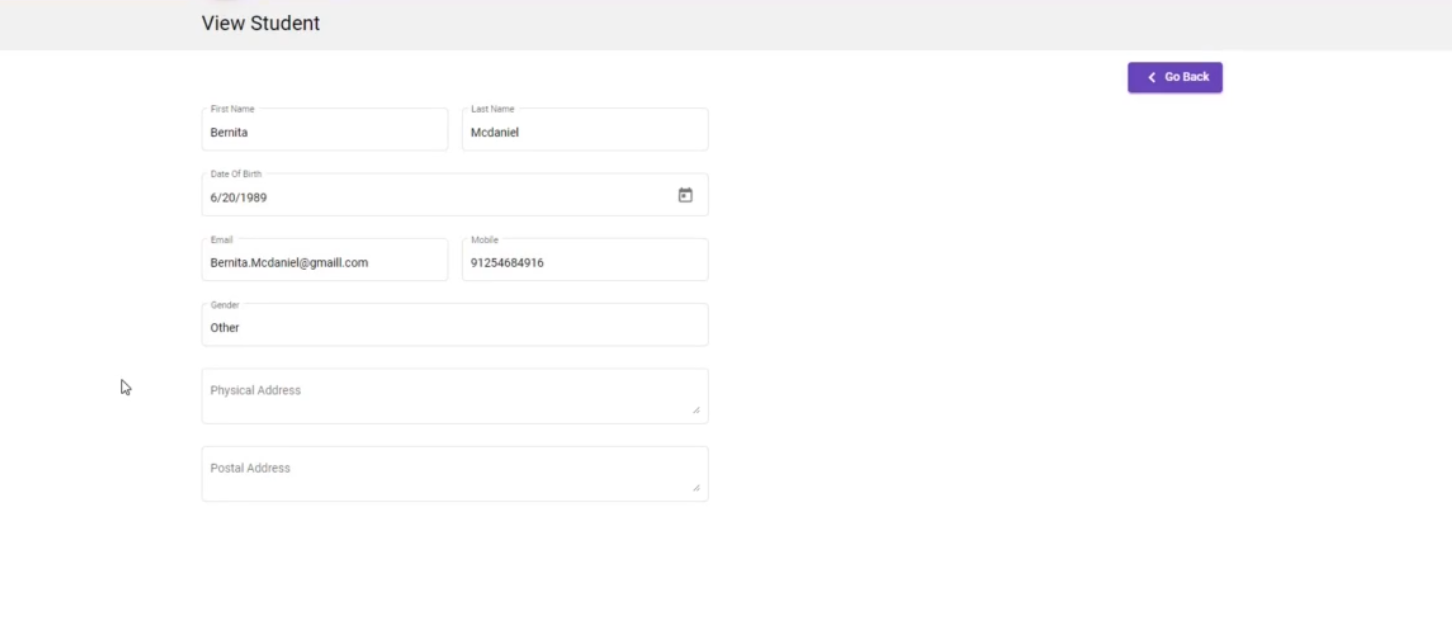
  }

  getImagePath(*relativePath*: string) {

    return `${this.baseApiUrl}/${relativePath}`;

  }

}



Update student via API

export interface UpdateStudentRequest {

    firstName: string,

    lastName: string,

    dateOfBirth: string,

    email: string,

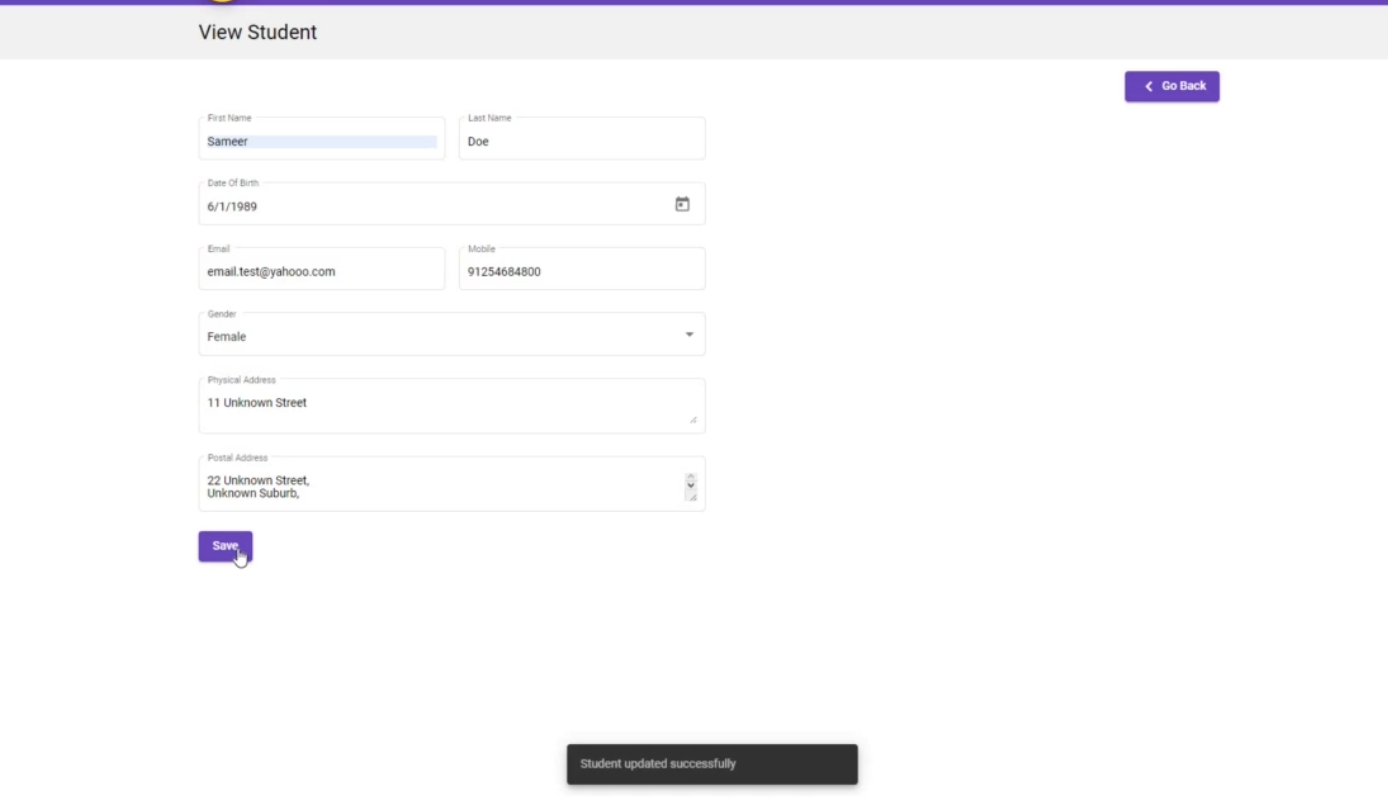
    mobile: number,

    genderId: string,

    physicalAddress: string,

    postalAddress: string

  }



Delete Student :-

