Dashboard.component.html

<div \*ngIf="username=='sid' && password=='sid' ">

<nav class="navbar navbar-dark bg-primary justify-content-between">

<a class="navbar-brand">UEM StudentManagement System</a>

<form class="form-inline">

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">Log Out</a>

<!-- <a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="login">login</a>

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">dashboard</a>

-->

</form>

</nav>

<form #registerForm="ngForm" (ngSubmit)="register(registerForm)">

<div class="container-fluid">

<div class=" mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Name:</label>

<input class="form-control" type="text" #name (ngModel)="name" name="name" id="name" placeholder="name">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Father Name:</label>

<input class="form-control" type="text" #fathersName (ngModel)="fathersName" name="fathersName" id="fathersName" placeholder="fathersName">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Phone Number:</label>

<input class="form-control" type="text" #phone (ngModel)="phone" name="phone" id="phone" placeholder="phone">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Email Address:</label>

<input class="form-control" type="text" #email (ngModel)="email" name="email" id="email" placeholder="email">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Home Address:</label>

<input class="form-control" type="text" #address (ngModel)="address" name="address" id="address" placeholder="address">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Percentile:</label>

<input class="form-control" type="text" #percentage (ngModel)="percentage" name="percentage" id="percentage" placeholder="percentage">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Grade:</label>

<input class="form-control" type="text" #grade (ngModel)="grade" name="grade" id="grade" placeholder="grade">

</div>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Enter Course Name:</label>

<input class="form-control" type="text" #course (ngModel)="course" name="course" id="course" placeholder="course">

</div>

<input class="btn btn-primary " type="submit" value="register">

</div>

</form>

<table class="content-table">

<thead class="content-table">

<tr class="content">

<th>Roll Number</th>

<th>Name</th>

<th>Father's Name</th>

<th>Phone Number</th>

<th>Email</th>

<th>Address</th>

<th>Percentile</th>

<th>Grade</th>

<th>Course</th>

<th>Edit</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let student of studentDetails">

<td class="color">{{student.rollNumber}}</td>

<td>{{student.name}}</td>

<td>{{student.fathersName}}</td>

<td>{{student.phone}}</td>

<td>{{student.email}}</td>

<td>{{student.address}}</td>

<td>{{student.percentage}}%</td>

<td>{{student.grade}}</td>

<td>{{student.course}}</td>

<td><button class="btn btn-primary btn btn-outline-light" type="button" data-bs-toggle="modal" data-bs-target="#exampleModal" (click) = "edit(student)">Edit</button></td>

<td><button class="btn btn-dark btn btn-outline-light" type="button" (click)="deleteStudent(student)">Delete</button></td>

</tr>

</tbody>

</table>

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Upadte Student</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<form #updateForm="ngForm" (ngSubmit)="updateStudent()">

<div class="mb-3">

<input class="form-control" type="text" disabled #rollNumber [(ngModel)]="studentToUpdate.rollNumber" name="rollNumber" id="rollNumber" placeholder="rollNumber">

</div>

<div class="mb-3">

<input class="form-control" type="text" #name [(ngModel)]="studentToUpdate.name" name="name" id="name" placeholder="name">

</div>

<div class="mb-3">

<input class="form-control" type="text" #fathersName [(ngModel)]="studentToUpdate.fathersName" name="fathersName" id="fathersName" placeholder="fathersName">

</div>

<div class="mb-3">

<input class="form-control" type="text" #phone [(ngModel)]="studentToUpdate.phone" name="phone" id="phone" placeholder="phone">

</div>

<div class="mb-3">

<input class="form-control" type="text" #email [(ngModel)]="studentToUpdate.email" name="email" id="email" placeholder="email">

</div>

<div class="mb-3">

<input class="form-control" type="text" #address [(ngModel)]="studentToUpdate.address" name="address" id="address" placeholder="address">

</div>

<div class="mb-3">

<input class="form-control" type="text" #percentage [(ngModel)]="studentToUpdate.percentage" name="percentage" id="percentage" placeholder="percentage">

</div>

<div class="mb-3">

<input class="form-control" type="text" #grade [(ngModel)]="studentToUpdate.grade" name="grade" id="grade" placeholder="grade">

</div>

<div class="mb-3">

<input class="form-control" type="text" #course [(ngModel)]="studentToUpdate.course" name="course" id="course" placeholder="course">

</div>

<input class="btn btn-primary" type="submit" value="update">

</form>

</div>

</div>

</div>

</div>

</div>

<div \*ngIf="username!='sid' && password!='sid' ">

<nav class="navbar navbar-dark bg-primary justify-content-between">

<a class="navbar-brand">UEM StudentManagement System</a>

<form class="form-inline">

<!-- <a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">Log Out</a> -->

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="login">login</a>

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">dashboard</a>

</form>

</nav>

<table class="content-table">

<thead class="content-table">

<tr class="content">

<th>Roll Number</th>

<th>Name</th>

<th>Father's Name</th>

<th>Phone Number</th>

<th>Email</th>

<th>Address</th>

<th>Percentile</th>

<th>Grade</th>

<th>Course</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let student of studentDetails">

<td class="color">{{student.rollNumber}}</td>

<td>{{student.name}}</td>

<td>{{student.fathersName}}</td>

<td>{{student.phone}}</td>

<td>{{student.email}}</td>

<td>{{student.address}}</td>

<td>{{student.percentage}}%</td>

<td>{{student.grade}}</td>

<td>{{student.course}}</td>

</tr>

</tbody>

</table>

</div>

<!-- <form #registerForm="ngForm" (ngSubmit)="register(registerForm)">

<input type="text" #name (ngModel)="name" name="name" id="name" placeholder="name">

<input type="text" #phone (ngModel)="phone" name="phone" id="phone" placeholder="phone">

<input type="text" #percentage (ngModel)="percentage" name="percentage" id="percentage" placeholder="percentage">

<input type="submit" value="register">

</form>

<table>

<thead>

<tr>

<th>Roll Number</th>

<th>name</th>

<th>phone Number</th>

<th>percentage</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let student of studentDetails">

<td>{{student.rollNumber}}</td>

<td>{{student.name}}</td>

<td>{{student.phone}}</td>

<td>{{student.percentage}}</td>

</tr>

</tbody>

</table> -->

Dashboard.component.ts

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

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

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

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

@Component({

selector: 'app-dashboard',

templateUrl: './dashboard.component.html',

styleUrls: ['./dashboard.component.css']

})

export class DashboardComponent{

title = 'StudentDashboard';

studentDetails: any;

username: any;

password: any;

studentToUpdate = {

rollNumber: "",

name: "",

fathersName: "",

phone: "",

email: "",

address: "",

percentage: "",

grade: "",

course: ""

};

constructor(private studentService: StudentService, private route: ActivatedRoute,

private router: Router) {

this.getStudentsDetails();

}

ngOnInit(){

this.username = this.route.snapshot.paramMap.get('username');

console.log(this.username);

this.password = this.route.snapshot.paramMap.get('password');

console.log(this.password);

}

register(registerForm: NgForm){

this.studentService.registerStudent(registerForm.value).subscribe(

(resp) => {

console.log(resp);

registerForm.reset();

this.getStudentsDetails();

},(err) => {

console.log(err);

}

);

}

getStudentsDetails(){

this.studentService.getStduents().subscribe(

(resp) => {

console.log(resp);

this.studentDetails = resp;

},(err) => {

console.log(err);

}

);

}

deleteStudent(student:any){

this.studentService.deleteStudent(student.rollNumber).subscribe(

(resp)=>{

console.log(resp);

this.getStudentsDetails();

},

(err)=>{

console.log(err);

}

);

}

edit(student: { rollNumber: string; name: string; fathersName: string; phone: string; email: string; address: string; percentage: string; grade: string; course: string; }){

this.studentToUpdate = student;

}

updateStudent(){

this.studentService.updateStudent(this.studentToUpdate).subscribe(

(resp) => {

console.log(resp);

},

(err)=>{

console.log(err);

}

)

}

}

Dashboard.component.css

.content-table{

border-collapse: collapse;

margin: 30px 0;

font-size: 15px;

min-width: 300px;

text-align: center;

margin-left: auto;

margin-right: auto;

margin-top: 100px;

border-radius: 8px 8px 0 0;

overflow: hidden;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);

}

.content{

background-color: #004298;

color: #ffffff;

text-align: left;

margin-left: 30px;

}

table, th, td {

border-collapse: collapse;

}

th, td {

padding: 20px;

}

td:hover{

background-color: skyblue;

}

Login.component.html

<nav class="navbar navbar-dark bg-primary justify-content-between">

<a class="navbar-brand">UEM StudentManagement System</a>

<form class="form-inline">

<!-- <a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">Log Out</a> -->

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="login">login</a>

<a class="btn btn btn-light btn-outline-info my-2 my-sm-0" href="dashboard">dashboard</a>

</form>

</nav>

<div class="hero">

<div class="form-box">

<form class="input-group">

<input class="input-field" type="text" name="username" [(ngModel)]="username" id="username" placeholder="username" >

<input class="input-field" type="password" name="password" [(ngModel)]="password" id="password" placeholder="password">

<input class="submit-btn" type="submit" value="login" (click)="loginUser()">

</form>

</div>

</div>

<router-outlet></router-outlet>

Login.component.css

\*{

margin: 0;

padding: 0;

font-family: sans-serif ;

position: center;

}

.hero{

height: 100%;

width: 100%;

background-image: linear-gradient(rgba(0,0,0,0.4),rgba(0,0,0,0.4)),url(./../banner.jpeg);

background-position: center;

background-size: cover;

position: absolute;

}

.form-box{

width: 380px;

height: 480px;

position: relative;

margin: 6% auto;

background: #fff;

padding: 30px;

overflow: hidden;

}

.input-group{

top: 180px;

position: absolute;

width: 280px;

transition: .5s;

}

.input-field{

width: 100%;

padding: 10px 0;

margin: 5px 0;

border-left: 0;

border-top: 0;

border-right: 0;

border-bottom: 1px solid #999;

outline: none;

background: transparent;

}

.submit-btn{

width: 95%;

padding: 10px 80px;

cursor: pointer;

display: block;

margin: auto;

background: linear-gradient(to right, #ff105f,#ffad06);

border: 0;

outline: none;

border-radius: 30px;

}

#login{

left: 50px;

}

Login.component.ts

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

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

@Component({

selector: 'app-login',

templateUrl: './login.component.html',

styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

username!: string;

password!: string;

constructor(private router: Router) { }

ngOnInit(): void {

}

loginUser(){

if(this.username=="sid" && this.password=="sid"){

this.router.navigate([`dashboard/${this.username}/${this.password}`]);

console.log("loged in");

}else{

console.log("not");

alert ("incorrect password");

this.router.navigate(['login']);

}

}

}

Student.service.ts

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

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

@Injectable({

providedIn: 'root'

})

export class StudentService {

constructor(private http: HttpClient) { }

API = "http://localhost:9090";

public registerStudent(studentData: any) {

return this.http.post(this.API + "/registerStudent", studentData);

}

public getStduents() {

return this.http.get(this.API + "/getStudents");

}

public deleteStudent(id: any) {

return this.http.delete(this.API + "/deleteStudent?id=" + id);

}

public updateStudent(student:any) {

return this.http.put(this.API + "/updateStudent", student);

}

}

App-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

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

import { DashboardComponent } from './dashboard/dashboard.component';

import { LoginComponent } from './login/login.component';

const routes: Routes = [

{path: '',component: DashboardComponent},

{path: 'dashboard',component: DashboardComponent},

{path: 'dashboard/:username/:password',component: DashboardComponent},

{path: 'login', component: LoginComponent}

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

spring boot:

StudentController.java

package com.example.student.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.example.student.model.Student;

import com.example.student.service.StudentService;

@RestController

@CrossOrigin(allowedHeaders = "\*", origins = "\*")

public class StudentController {

@Autowired

private StudentService studentService;

@PostMapping("/registerStudent")

public Student registerStudent(@RequestBody Student student) {

return studentService.registerStudent(student);

}

@GetMapping("/getStudents")

public List <Student> getStudent(){

return studentService.getStudents();

}

@DeleteMapping("/deleteStudent")

public void deleteStudent(@RequestParam Integer id) {

studentService.deleteStudent(id);

}

@PutMapping("/updateStudent")

public Student updateStudent(@RequestBody Student student) {

return studentService.updateStudent(student);

}

}

Student.java

package com.example.student.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import org.hibernate.annotations.GeneratorType;

@Entity

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer rollNumber;

private String name;

private String fathersName;

private String phone;

private Double percentage;

private String grade;

private String course;

private String address;

private String email;

public String getFathersName() {

return fathersName;

}

public void setFathersName(String fathersName) {

this.fathersName = fathersName;

}

public String getCourse() {

return course;

}

public void setCourse(String course) {

this.course = course;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getGrade() {

return grade;

}

public void setGrade(String grade) {

this.grade = grade;

}

public Integer getRollNumber() {

return rollNumber;

}

public void setRollNumber(Integer rollNumber) {

this.rollNumber = rollNumber;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public Double getPercentage() {

return percentage;

}

public void setPercentage(Double percentage) {

this.percentage = percentage;

}

}

studentRepository.java

package com.example.student.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import org.hibernate.annotations.GeneratorType;

@Entity

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer rollNumber;

private String name;

private String fathersName;

private String phone;

private Double percentage;

private String grade;

private String course;

private String address;

private String email;

public String getFathersName() {

return fathersName;

}

public void setFathersName(String fathersName) {

this.fathersName = fathersName;

}

public String getCourse() {

return course;

}

public void setCourse(String course) {

this.course = course;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getGrade() {

return grade;

}

public void setGrade(String grade) {

this.grade = grade;

}

public Integer getRollNumber() {

return rollNumber;

}

public void setRollNumber(Integer rollNumber) {

this.rollNumber = rollNumber;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public Double getPercentage() {

return percentage;

}

public void setPercentage(Double percentage) {

this.percentage = percentage;

}

}

StudentService.java

package com.example.student.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.example.student.model.Student;

import com.example.student.repository.StudentRepository;

@Service

public class StudentService {

@Autowired

private StudentRepository studentRepository;

public Student registerStudent(Student student) {

return studentRepository.save(student);

}

public List<Student> getStudents(){

return (List<Student>) studentRepository.findAll();

}

public void deleteStudent(Integer id) {

studentRepository.deleteById(id);

}

public Student updateStudent(Student student) {

Integer rollNumber = student.getRollNumber();

Student std = studentRepository.findById(rollNumber).get();

std.setName(student.getName());

std.setPhone(student.getPhone());

std.setPercentage(student.getPercentage());

std.setAddress(student.getAddress());

std.setCourse(student.getCourse());

std.setEmail(student.getEmail());

std.setGrade(student.getGrade());

std.setFathersName(student.getFathersName());

return studentRepository.save(std);

}

}