Mini Projet : ANAFLOUS Abdellatif

Spring boot+Angular+MySQL

Sujet :

Le projet c’est une application web de gestion des etudiants:

Les fonctionnalités :

* Ajouter.
* Modifier
* Supprimer
* Afficher

Backend:

La classe Sudent :

package tech.upm.studentmanagerapp.model;  
  
import javax.persistence.\*;  
import java.io.Serializable;  
@Entity  
public class Student implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*AUTO*)  
 @Column(nullable = false,updatable = false)  
 private Long id;  
 private String name;  
 private String email;  
 private String phone;  
 private String imageUrl;  
 @Column(nullable = false,updatable = false)  
 private String studentCode;  
  
 public Student() {  
 }  
 public Student(String name, String email, String phone, String imageUrl, String studentCode) {  
 this.name = name;  
 this.email = email;  
 this.phone = phone;  
 this.imageUrl = imageUrl;  
 this.studentCode = studentCode;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getPhone() {  
 return phone;  
 }  
  
 public void setPhone(String phone) {  
 this.phone = phone;  
 }  
  
 public String getImageUrl() {  
 return imageUrl;  
 }  
  
 public void setImageUrl(String imageUrl) {  
 this.imageUrl = imageUrl;  
 }  
  
 public String getStudentCode() {  
 return studentCode;  
 }  
  
 public void setStudentCode(String studentCode) {  
 this.studentCode = studentCode;  
 }  
  
 @Override  
 public String toString() {  
 return "Student{" +  
 "id=" + id +  
 ", name='" + name + '\'' +  
 ", email='" + email + '\'' +  
 ", phone='" + phone + '\'' +  
 ", imageUrl='" + imageUrl + '\'' +  
 ", studentCode='" + studentCode + '\'' +  
 '}';  
 }  
}

Interface StudentRepo :

package tech.upm.studentmanagerapp.repo;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import tech.upm.studentmanagerapp.model.Student;  
  
import java.util.Optional;  
  
public interface StudentRepo extends JpaRepository<Student,Long> {  
 void deleteStudentById(Long id);  
  
 Optional <Student> findStudentById(Long id);  
}

La classe StudentSerivice :

package tech.upm.studentmanagerapp.service;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import tech.upm.studentmanagerapp.exception.UserNotFoundException;  
import tech.upm.studentmanagerapp.model.Student;  
import tech.upm.studentmanagerapp.repo.StudentRepo;  
  
import javax.transaction.Transactional;  
import java.nio.file.attribute.UserPrincipalNotFoundException;  
import java.util.List;  
import java.util.UUID;  
  
@Service  
public class StudentService {  
 private final StudentRepo studnetRepo;  
 @Autowired  
 public StudentService(StudentRepo studnetRepo) {  
 this.studnetRepo = studnetRepo;  
 }  
  
 public Student addStudent(Student student){  
 student.setStudentCode(UUID.*randomUUID*().toString());  
 return studnetRepo.save(student);  
 }  
 public List<Student> findAllStudents(){  
 return studnetRepo.findAll();  
 }  
 public Student updateStudent(Student student){  
 return studnetRepo.save(student);  
 }  
 public Student findStudentBYiId(Long id){  
 return studnetRepo.findStudentById(id).orElseThrow(() -> new UserNotFoundException("User by id "+id+"was not found"));  
 }  
 @Transactional  
 public void deleteStudent(Long id){  
 studnetRepo.deleteStudentById(id);  
 }  
  
}

Classe Principale :

package tech.upm.studentmanagerapp;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.Bean;  
import org.springframework.web.cors.CorsConfiguration;  
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;  
import org.springframework.web.filter.CorsFilter;  
  
import java.util.Arrays;  
  
@SpringBootApplication  
public class StudentmanagerappApplication {  
  
 public static void main(String[] args) {  
  
 SpringApplication.*run*(StudentmanagerappApplication.class, args);  
 }  
  
 @Bean  
 public CorsFilter corsFilter() {  
 CorsConfiguration corsConfiguration = new CorsConfiguration();  
 corsConfiguration.setAllowCredentials(true);  
 corsConfiguration.setAllowedOrigins(Arrays.*asList*("http://localhost:4200"));  
 corsConfiguration.setAllowedHeaders(Arrays.*asList*("Origin", "Access-Control-Allow-Origin", "Content-Type",  
 "Accept", "Authorization", "Origin, Accept", "X-Requested-With",  
 "Access-Control-Request-Method", "Access-Control-Request-Headers"));  
 corsConfiguration.setExposedHeaders(Arrays.*asList*("Origin", "Content-Type", "Accept", "Authorization",  
 "Access-Control-Allow-Origin", "Access-Control-Allow-Origin", "Access-Control-Allow-Credentials"));  
 corsConfiguration.setAllowedMethods(Arrays.*asList*("GET", "POST", "PUT", "DELETE", "OPTIONS"));  
 UrlBasedCorsConfigurationSource urlBasedCorsConfigurationSource = new UrlBasedCorsConfigurationSource();  
 urlBasedCorsConfigurationSource.registerCorsConfiguration("/\*\*", corsConfiguration);  
 return new CorsFilter(urlBasedCorsConfigurationSource);  
 }  
  
  
}

La classe StudentResource :

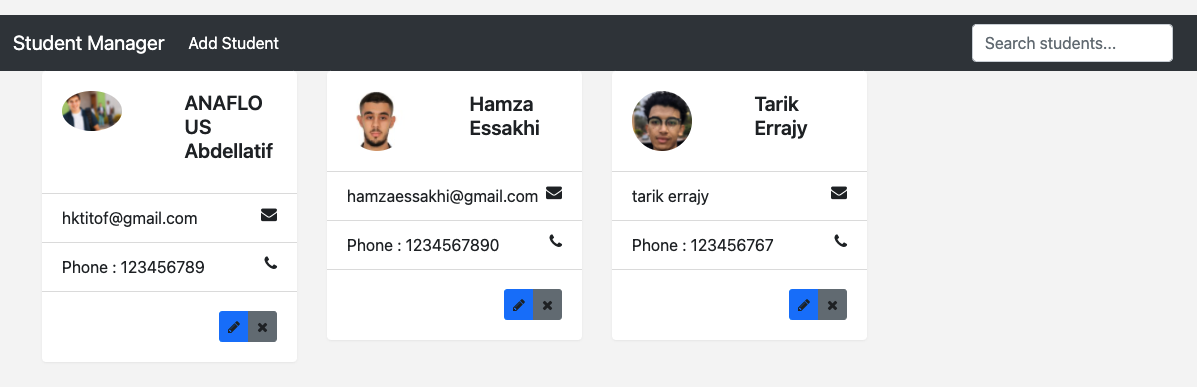
package tech.upm.studentmanagerapp;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import tech.upm.studentmanagerapp.model.Student;  
import tech.upm.studentmanagerapp.service.StudentService;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/student")  
public class StudentResource {  
 private final StudentService studentService;  
  
 public StudentResource(StudentService studentService) {  
 this.studentService = studentService;  
 }  
  
 @GetMapping("/all")  
 public ResponseEntity<List<Student>> getAllStudents(){  
 List<Student> students = studentService.findAllStudents();  
 return new ResponseEntity<>(students, HttpStatus.*OK*);  
 }  
  
 @GetMapping("/find/{id}")  
 public ResponseEntity<Student> getStudentById(@PathVariable("id") Long id){  
 Student student = studentService.findStudentBYiId(id);  
 return new ResponseEntity<>(student, HttpStatus.*OK*);  
 }  
  
 @PostMapping("/add")  
 public ResponseEntity<Student> addStudent(@RequestBody Student student){  
 Student newStudent = studentService.addStudent(student);  
 return new ResponseEntity<>(newStudent, HttpStatus.*CREATED*);  
 }  
  
 @PutMapping("/update")  
 public ResponseEntity<Student> updateStudent(@RequestBody Student student){  
 Student updateStudent = studentService.updateStudent(student);  
 return new ResponseEntity<>(updateStudent, HttpStatus.*OK*);  
 }  
  
 @DeleteMapping("/delete/{id}")  
 public ResponseEntity<?> deleteStudent(@PathVariable("id") Long id){  
 studentService.deleteStudent(id);  
 return new ResponseEntity<>(HttpStatus.*OK*);  
 }  
}

Configuration MYSQL :

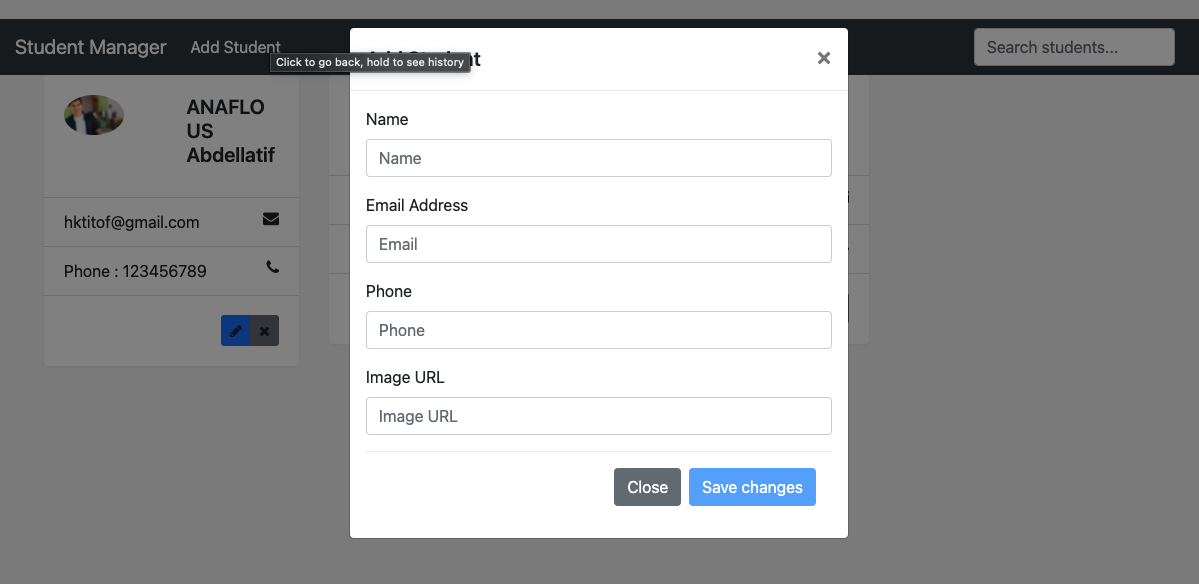
spring.datasource.url = jdbc:mysql://localhost:3306/studentmanager  
spring.datasource.username=root  
spring.datasource.password=  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

FrontEnd :

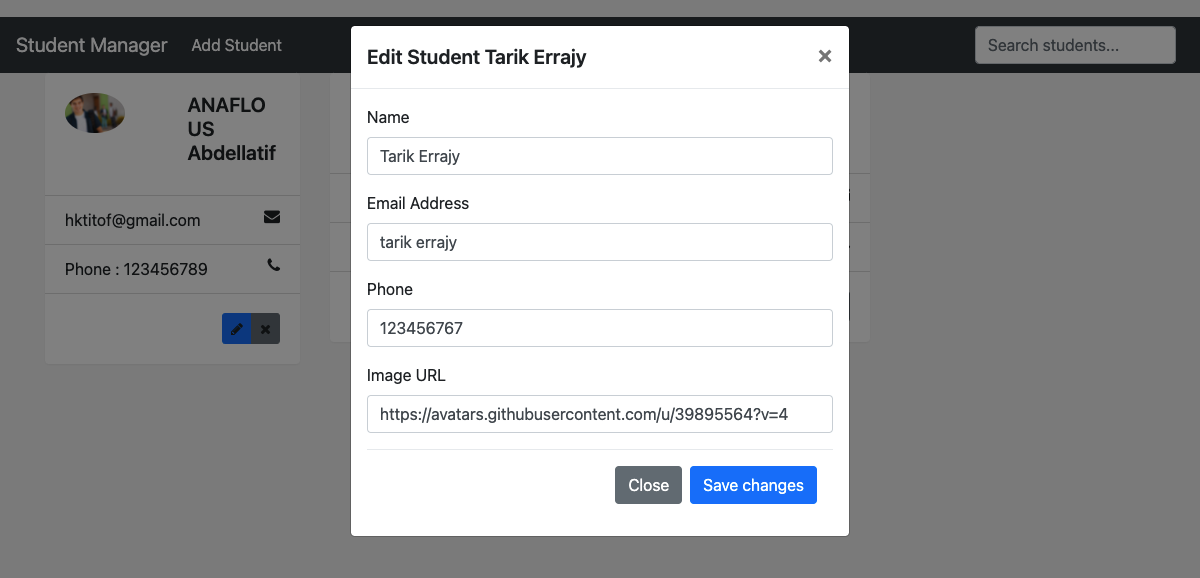
Page principale:



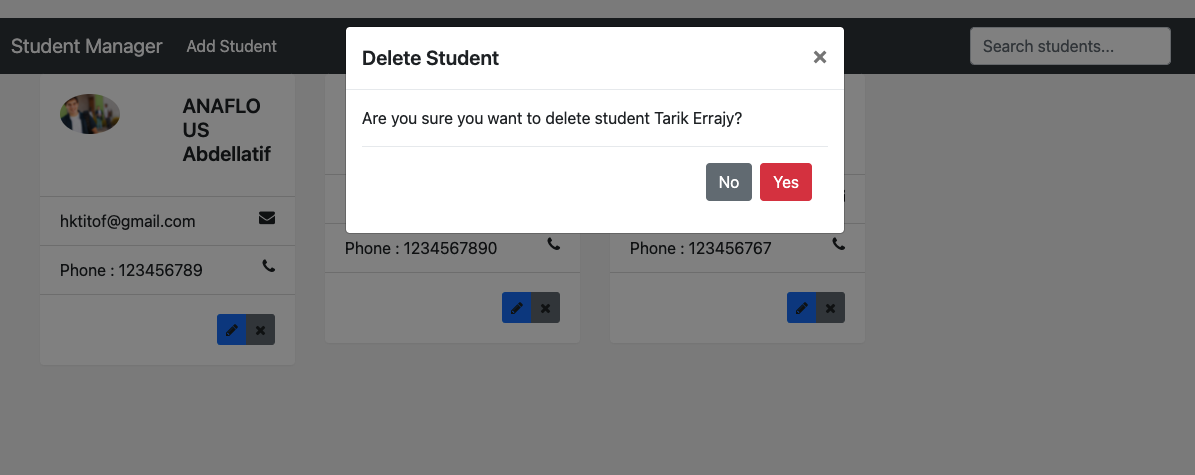
Ajouter un etudiant :



Modifier un etudiant :



Supprimer un etudiant :



Recherche un etudiant :

