package com.learnSphere.controller;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Controller;import org.springframework.web.bind.annotation.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;import com.learnSphere.services.TrainerService;@Controllerpublic class TrainerController { @Autowired TrainerService tService; @PostMapping("/addCourse") public String addCourse(@RequestParam("courseId")int courseId, @RequestParam("courseName")String courseName, @RequestParam("coursePrice")int coursePrice) { Course course=new Course(); course.setCourseId(courseId); course.setCourseName(courseName); course.setCoursePrice(coursePrice); tService.addCourse(course); return "redirect:/"; } @PostMapping("/lesson") public String lesson(@RequestParam("courseId")int courseId, @RequestParam("lessonId")int lessonId, @RequestParam("lessonName")String lessonName, @RequestParam("topics")String topics, @RequestParam("link")String link) { Course course=tService.getCourse(courseId); Lesson lesson=new Lesson(lessonId,lessonName,topics,link,course); tService.addLesson(lesson);

course.getLessons().add(lesson);

return "redirect:/";

}

}

package com.learnSphere.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.learnSphere.entity.Lesson;

public interface LessonRepository extends JpaRepository<Lesson, Integer>{

}

package com.learnSphere.entity;import jakarta.persistence.Entity;import jakarta.persistence.Id;import jakarta.persistence.ManyToOne;@Entitypublic class Lesson { @Id int lessonId; String lessonName; String topics; String link; @ManyToOne Course course; public Lesson() { super(); // TODO Auto-generated constructor stub } public Lesson(int lessonId, String lessonName, String topics, String link, Course course) { super(); this.lessonId = lessonId; this.lessonName = lessonName; this.topics = topics; this.link = link; this.course = course; } public int getLessonId() { return lessonId; } public void setLessonId(int lessonId) { this.lessonId = lessonId; } public String getLessonName() { return lessonName; } public void setLessonName(String lessonName) { this.lessonName = lessonName; } public String getTopics() { return topics; } public void setTopics(String topics) { this.topics = topics; } public String getLink() { return link; } public void setLink(String link) { this.link = link; } public Course getCourse() { return course; } public void setCourse(Course course) { this.course = course; } @Override public String toString() { return "Lesson [lessonId=" + lessonId + ", lessonName=" + lessonName + ", topics=" + topics + ", link=" + link

+ ", course=" + course + "]";

}

}

package com.learnSphere.services;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Service;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;import com.learnSphere.repository.CourseRepository;import com.learnSphere.repository.LessonRepository;@Servicepublic class TrainerServiceImplementation implements TrainerService{ @Autowired CourseRepository courseRepo; @Autowired LessonRepository lessonRepo; @Override public String addCourse(Course course) { courseRepo.save(course); return "Course added successfully!"; } @Override public String addLesson(Lesson lesson) { lessonRepo.save(lesson); return "lesson added successfully!"; } @Override public Course getCourse(int courseId) {

return courseRepo.findById(courseId).get();

}

}

KodNest to Everyone 12:33

package com.learnSphere.controller;import java.util.List;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Controller;import org.springframework.web.bind.annotation.GetMapping;import org.springframework.web.bind.annotation.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;import com.learnSphere.services.TrainerService;@Controllerpublic class TrainerController { @Autowired TrainerService tService; @PostMapping("/addCourse") public String addCourse(@RequestParam("courseId")int courseId, @RequestParam("courseName")String courseName, @RequestParam("coursePrice")int coursePrice) { Course course=new Course(); course.setCourseId(courseId); course.setCourseName(courseName); course.setCoursePrice(coursePrice); tService.addCourse(course); return "redirect:/"; } @PostMapping("/lesson") public String lesson(@RequestParam("courseId")int courseId, @RequestParam("lessonId")int lessonId, @RequestParam("lessonName")String lessonName, @RequestParam("topics")String topics, @RequestParam("link")String link) { Course course=tService.getCourse(courseId); Lesson lesson=new Lesson(lessonId,lessonName,topics,link,course); tService.addLesson(lesson); course.getLessons().add(lesson); return "redirect:/"; } @GetMapping("/showCourses") public String showCourses() { List<Course> courseList=tService.courseList();

System.out.println(courseList);

return "courses";

}

}

<!doctype html><html lang="en"> <head> <!-- Required meta tags --> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <!-- Bootstrap CSS --> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"> <title>Trainer home</title> </head> <body> <h1>Welcome trainer!</h1> <div class="d-grid gap-2"> <button class="btn btn-primary" type="button"><a href="/createCourse">Add Course</a></button> <button class="btn btn-primary" type="button"><a href="/addLesson">Add Lesson</a></button> <button class="btn btn-primary" type="button"><a href="/showCourses">View Courses</a></button> </div> <!-- Optional JavaScript; choose one of the two! --> <!-- Option 1: Bootstrap Bundle with Popper --> <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script> <!-- Option 2: Separate Popper and Bootstrap JS --> <!-- <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script> <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

KodNest to Everyone 12:33

package com.learnSphere.services;import java.util.List;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;public interface TrainerService { public String addCourse(Course course); public String addLesson(Lesson lesson);

public Course getCourse(int courseId);

public List<Course> courseList();

}

package com.learnSphere.services;import java.util.List;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Service;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;import com.learnSphere.repository.CourseRepository;import com.learnSphere.repository.LessonRepository;@Servicepublic class TrainerServiceImplementation implements TrainerService{ @Autowired CourseRepository courseRepo; @Autowired LessonRepository lessonRepo; @Override public String addCourse(Course course) { courseRepo.save(course); return "Course added successfully!"; } @Override public String addLesson(Lesson lesson) { lessonRepo.save(lesson); return "lesson added successfully!"; } @Override

public Course getCourse(int courseId) {

return courseRepo.findById(courseId).get();

}

@Override

public List<Course> courseList() {

return courseRepo.findAll();

}

}

KodNest to Everyone 13:00

<head><meta charset="UTF-8"><title>Students Details</title></head><body> <table border="1"> <tr> <th>KodId</th> <th>Name</th> <th>Branch</th> </tr> <tr> <td>th:text="${studen.kodId}"</td> <td>th:text="${studen.name}</td> <td>th:text="${studen.branch}</td>

</tr>

</table>

</body>

</html>

<table border="1"> <tr> <th>Course\_Id</th> <th>Corse\_Name</th> <th>Course\_Price</th> </tr> <tr> <td th:text="${Course.CourseId}"/> <td th:text="${Course.CourseName}"/> <td th:text="${Course.CoursePrice}"/>

</tr>

</table>

<table border="1"class="table table-success table-striped table-bordered w-50"> <thead> <tr> <th scope="col">Course Id</th> <th scope="col">Course Name</th> <th scope="col">Course Price</th> <th scope="col">Lessons</th> </tr> </thead> <tbody> <tr th:each="course:${course}"> <th th:text="${course.courseId}"></th> <td th:text="${course.coureseName}"></td> <td th:text="${course.coursePrice}"></td>

<td th:text="${course.lessons}"></td>

</tr>

</tbody>

</table>

KodNest to Everyone 13:12

<!DOCTYPE html><html xmlns:th="https://www.thymeleaf.org"><head><meta charset="ISO-8859-1"><title>View Course</title><link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"></head><body> <div> <table border="1"class="table table-success table-striped table-bordered w-50"> <thead> <tr> <th scope="col">Course Id</th> <th scope="col">Course Name</th> <th scope="col">Course Price</th> <th scope="col">Lessons</th> </tr> </thead> <tbody> <tr th:each="course:${course}"> <th th:text="${course.courseId}"></th>

<td th:text="${course.coureseName}"></td>

<td th:text="${course.coursePrice}"></td>

<td th:text="${course.lessons}"></td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

<!DOCTYPE html><html xmlns:th="https://www.thymeleaf.org"><head><meta charset="ISO-8859-1"><title>View Course</title><link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"></head><body> <div> <table border="1"class="table table-success table-striped table-bordered w-50"> <thead> <tr> <th scope="col">Course Id</th> <th scope="col">Course Name</th> <th scope="col">Course Price</th> <th scope="col">Lessons</th> </tr> </thead> <tbody> <tr th:each="course:${courseList}"> <th th:text="${course.courseId}"></th>

<td th:text="${course.coureseName}"></td>

<td th:text="${course.coursePrice}"></td>

<td th:text="${course.lessons}"></td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

package com.learnSphere.controller;import java.util.List;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Controller;import org.springframework.ui.Model;import org.springframework.web.bind.annotation.GetMapping;import org.springframework.web.bind.annotation.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Course;import com.learnSphere.entity.Lesson;import com.learnSphere.services.TrainerService;@Controllerpublic class TrainerController { @Autowired TrainerService tService; @PostMapping("/addCourse") public String addCourse(@RequestParam("courseId")int courseId, @RequestParam("courseName")String courseName, @RequestParam("coursePrice")int coursePrice) { Course course=new Course(); course.setCourseId(courseId); course.setCourseName(courseName); course.setCoursePrice(coursePrice); tService.addCourse(course); return "redirect:/"; } @PostMapping("/lesson") public String lesson(@RequestParam("courseId")int courseId, @RequestParam("lessonId")int lessonId, @RequestParam("lessonName")String lessonName, @RequestParam("topics")String topics, @RequestParam("link")String link) { Course course=tService.getCourse(courseId); Lesson lesson=new Lesson(lessonId,lessonName,topics,link,course); tService.addLesson(lesson); course.getLessons().add(lesson); return "redirect:/"; } @GetMapping("/showCourses") public String showCourses(Model model) { List<Course> courseList=tService.courseList(); model.addAttribute("courseList",courseList); // System.out.println(courseList); return "courses"; }}