<!DOCTYPE html><html> <head> <meta charset="ISO-8859-1"> <title>Learn lessons</title> </head> <body> <h1>Welcome to lessons</h1> <form action="/goToLesson" method="post"> <label>Enter lesson ID: </label> <br> <input type="text" name="lessonId">

<input type="submit" value="Start Lesson">

</form>

</body>

</html>

package com.learnSphere.controller;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Controller;import org.springframework.ui.Model;import org.springframework.web.bind.annotation.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Lesson;import com.learnSphere.services.StudentServices;@Controllerpublic class StudentController { @Autowired StudentServices service; @PostMapping("/goToLesson") public String myLesson(@RequestParam("lessonId")int lessonId,

Model model) {

Lesson lesson = service.getLesson(lessonId);

model.addAttribute("lesson",lesson);

System.out.println(lesson);

return "myLesson";

}

}

package com.learnSphere.services;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Service;import com.learnSphere.entity.Lesson;import com.learnSphere.repository.LessonRepository;@Servicepublic class StudentServicesImplementation implements StudentServices{

@Autowired

LessonRepository lessonRepo;

@Override

public Lesson getLesson(int lessonId) {

return lessonRepo.findById(lessonId).get();

}

}

<!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 class="table table-success table-striped table-bordered w-50"> <thead> <tr> <th scope="col">Lesson Id</th> <th scope="col">Lesson Name</th> <th scope="col">Lesson Topics</th> <th scope="col">Lessons link</th> <th scope="col">Course Id</th> </tr> </thead> <tbody> <tr > <td th:text="${lesson.lessonId}"></td> <td th:text="${lesson.lessonName}"></td>

<td th:text="${lesson.topics}"></td>

<td th:text="${lesson.link}"></td>

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

</tr>

</tbody>

</table>

</div>

</body>

</html>

package com.learnSphere.services;import com.learnSphere.entity.Lesson;public interface StudentServices { Lesson getLesson(int lessonId);}

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.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Comments;import com.learnSphere.entity.Users;import com.learnSphere.services.CommentService;import com.learnSphere.services.UserService;@Controllerpublic class ServiceController { @Autowired UserService uService; @Autowired CommentService cService; @PostMapping("/addUser") public String addUser(@RequestParam("name")String name, @RequestParam("email")String email, @RequestParam("password")String password, @RequestParam("role")String role) { boolean emailExists=uService.checkEmail(email); if(emailExists==false) { Users user = new Users(); user.setName(name); user.setEmail(email); user.setPassword(password); user.setRole(role); uService.addUser(user); System.out.println("user registered successfully!"); return "redirect:/login"; } else { System.out.println("user already exists!"); return "redirect:/register"; } } @PostMapping("/validate") public String validate(@RequestParam("email")String email, @RequestParam("password")String password) { if(uService.checkEmail(email)) { boolean val=uService.validate(email, password); //if user is valid if(val==true) { if(uService.getUserRole(email).equals("trainer")) { return "trainerHome"; } else { return "studentHome"; } } else { System.out.println("incorrect credentials, try again!"); return "login"; } } else { return "login"; }} @PostMapping("/addComment") public String comments(@RequestParam("comment")String comment ,Model model) { Comments c=new Comments(); c.setComment(comment); cService.addComment(c); List<Comments> commentsList=cService.commentList(); model.addAttribute("comments",commentsList);

return "redirect:/";

}

}

package com.learnSphere.services;import java.util.List;import org.springframework.beans.factory.annotation.Autowired;import org.springframework.stereotype.Service;import com.learnSphere.entity.Comments;import com.learnSphere.repository.CommentRepository;@Servicepublic class CommentSeviceImplementation implements CommentService { @Autowired CommentRepository commentRepo; @Override public List<Comments> commentList() { return commentRepo.findAll(); } @Override public String addComment(Comments comment) {

commentRepo.save(comment);

return "comment added";

}

}

package com.learnSphere.repository;

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

import com.learnSphere.entity.Comments;

public interface CommentRepository extends JpaRepository<Comments,Integer>{

}

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.PostMapping;import org.springframework.web.bind.annotation.RequestParam;import com.learnSphere.entity.Comments;import com.learnSphere.entity.Users;import com.learnSphere.services.CommentService;import com.learnSphere.services.UserService;@Controllerpublic class ServiceController { @Autowired UserService uService; @Autowired CommentService cService; @PostMapping("/addUser") public String addUser(@RequestParam("name")String name, @RequestParam("email")String email, @RequestParam("password")String password, @RequestParam("role")String role) { boolean emailExists=uService.checkEmail(email); if(emailExists==false) { Users user = new Users(); user.setName(name); user.setEmail(email); user.setPassword(password); user.setRole(role); uService.addUser(user); System.out.println("user registered successfully!"); return "redirect:/login"; } else { System.out.println("user already exists!"); return "redirect:/register"; } } @PostMapping("/validate") public String validate(@RequestParam("email")String email, @RequestParam("password")String password) { if(uService.checkEmail(email)) { boolean val=uService.validate(email, password); //if user is valid if(val==true) { if(uService.getUserRole(email).equals("trainer")) { return "trainerHome"; } else { return "studentHome"; } } else { System.out.println("incorrect credentials, try again!"); return "login"; } } else { return "login"; }} @PostMapping("/addComment") public String comments(@RequestParam("comment")String comment ,Model model) { Comments c=new Comments(); c.setComment(comment); cService.addComment(c); List<Comments> commentsList=cService.commentList(); model.addAttribute("comments",commentsList);

return "redirect:/";

}

}

package com.learnSphere.services;import java.util.List;import com.learnSphere.entity.Comments;public interface CommentService { List<Comments> commentList(); String addComment(Comments comment);}