查课选课功能实现

package cn.edu.cqu.nsers.Controller;  
  
import cn.edu.cqu.nsers.Service.\*;  
import cn.edu.cqu.nsers.pojo.Course;  
import cn.edu.cqu.nsers.pojo.Student\_Course;  
import cn.edu.cqu.nsers.pojo.Teacher;  
import cn.edu.cqu.nsers.pojo.Teacher\_Course;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.\*;  
import java.util.\*;  
  
@Controller  
@RequestMapping("/functionPage")  
public class Func\_Course {  
 @Autowired  
 public StudentService studentService;  
 @Autowired  
 public CourseService courseService;  
 @Autowired  
 public TeacherService teacherService;  
 @Autowired  
 public Teacher\_CourseService teacher\_courseService;  
 @Autowired  
 public Student\_CourseService student\_courseService;  
  
 @GetMapping("/showCoursePage")  
 public String showCoursePage() {  
 return "showCoursePage";  
 }  
  
 @PostMapping("/showCourse") //在查课页面一开始展示所有课程信息+授课教师  
 public ResponseEntity<Map<String, Object>> showCourse()  
 {  
 Map<String, Object> response = new HashMap<>();  
 List<Course> course = courseService.list();  
 Map<Integer, List<String>> CourseWithTeacher = new HashMap<>();  
 //put老师的名字进值里  
 for (int i = 0; i < course.size(); i++)  
 {  
 QueryWrapper<Teacher\_Course> teacher\_courseQueryWrapper = new QueryWrapper<>();  
 Course course1 = course.get(i);  
 int cid = course1.getCid();  
 teacher\_courseQueryWrapper.eq("cid", cid);  
 List<Teacher\_Course> teacher\_courseList = teacher\_courseService.list(teacher\_courseQueryWrapper);  
 List<String> tnamelist = new ArrayList<>();  
 for (Teacher\_Course teacher\_course : teacher\_courseList) {  
 Integer tid = teacher\_course.getTid();  
 Teacher teacher1 = teacherService.getById(tid);  
 tnamelist.add(teacher1.getTname());  
 }  
 //put课程剩余的属性进值里  
 tnamelist.add("division");  
 tnamelist.add(course1.getCname());  
 tnamelist.add(course1.getCaddress());  
 tnamelist.add(course1.getCtime());  
 CourseWithTeacher.put(i, tnamelist);  
 }  
 response.put("CourseWithTeacher", CourseWithTeacher);  
 response.put("success",true);  
 return ResponseEntity.*ok*(response);  
 }  
  
 @PostMapping("/searchCourse") //查课时根据学生搜索的课程名称返回课程信息+授课教师  
 public ResponseEntity<Map<String, Object>> getCourse(@RequestParam("cname") String cname)  
 {  
 Map<String,Object> response = new HashMap<>();  
 List<String> CourseWithTeacher = new ArrayList<>();  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname",cname);  
 Course course = courseService.getOne(courseQueryWrapper);  
 QueryWrapper<Teacher\_Course> teacher\_courseQueryWrapper = new QueryWrapper<>();  
 teacher\_courseQueryWrapper.eq("cid", course.getCid());  
 List<Teacher\_Course> teacher\_courseList = teacher\_courseService.list(teacher\_courseQueryWrapper);  
 for (int j = 0; j < teacher\_courseList.size(); j++)  
 {  
 Teacher\_Course teacher\_course = teacher\_courseList.get(j);  
 Integer tid = teacher\_course.getTid();  
 Teacher teacher = teacherService.getById(tid);  
 CourseWithTeacher.add(teacher.getTname());  
 }  
 CourseWithTeacher.add("division");  
 CourseWithTeacher.add(course.getCaddress());  
 CourseWithTeacher.add(course.getCtime());  
 response.put("CourseWithTeacher",CourseWithTeacher);  
 response.put("success",true);  
 return ResponseEntity.*ok*(response);  
 }  
  
 @GetMapping("/selectCoursePage")  
 public String showSelectCoursePage() {  
 return "selectCoursePage";  
 }  
  
 @PostMapping("/showSelectCourse1") //在选课页面一开始展示所有课，可能是已选、已选满或可选  
 public ResponseEntity<Map<String, Object>> selectCourse(@RequestParam("sid") Integer sid) {  
 Map<String, Object> response = new HashMap<>();  
 Map<Integer,List<String>>CourseStatus = new HashMap<>();  
 List<Course> course = courseService.list();  
 for(int i = 0;i < course.size(); i++)  
 {  
 Course course1 = course.get(i);  
 int cid = course1.getCid();  
 List<String>Info = new ArrayList<>();  
 Info.add(course1.getCname());  
 Info.add(course1.getCaddress());  
 Info.add(course1.getCtime());  
 Student\_Course student\_course = student\_courseService.getRelation(sid,cid);  
 if(student\_course!=null) //学生已经选了该门课  
 {  
 Info.add("已选");  
 }  
 else  
 {  
 if(course1.getCcnt()<course1.getCnumber()) //学生未选该门课，且该门课人数还没满  
 {  
 Info.add("可选");  
 }  
 else  
 {  
 Info.add("已选满"); //学生未选该门课，但该门课人数已满  
 }  
 }  
 CourseStatus.put(i,Info);  
 }  
 response.put("CourseStatus", CourseStatus);  
 response.put("success", true);  
 return ResponseEntity.*ok*(response);  
 }  
  
 @PostMapping("/showSelectCourse2") //点击要选的课程后展示哪些老师教这门课  
 public ResponseEntity<Map<String, Object>> selectTeacher(@RequestParam("sid") Integer sid,  
 @RequestParam("cname") String cname)  
 {  
 Map<String,Object> response = new HashMap<>();  
 List<String>TeacherStatus = new ArrayList<>();  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname",cname);  
 Course course = courseService.getOne(courseQueryWrapper);  
 QueryWrapper<Teacher\_Course> teacher\_courseQueryWrapper = new QueryWrapper<>();  
 int cid = course.getCid();  
 teacher\_courseQueryWrapper.eq("cid",cid);  
 List<Teacher\_Course> teacher\_courseList = teacher\_courseService.list(teacher\_courseQueryWrapper);  
 TeacherStatus.add(course.getCaddress());  
 TeacherStatus.add(course.getCtime());  
 for (int j = 0; j < teacher\_courseList.size(); j++)  
 {  
 Teacher\_Course teacher\_course = teacher\_courseList.get(j);  
 Integer tid1 = teacher\_course.getTid();  
 Teacher teacher = teacherService.getById(tid1);  
 TeacherStatus.add(teacher.getTname());  
 Student\_Course student\_course = student\_courseService.getRelation(sid,cid);  
 if(student\_course != null) //学生已经选择该老师的该门课  
 {  
 Integer tid2 = student\_course.getTid();  
 if(Objects.*equals*(tid1, tid2))  
 {  
 TeacherStatus.add("已选");  
 }  
 else  
 {  
 if(teacher\_course.getCnt()<teacher\_course.getNum())  
 {  
 TeacherStatus.add("已选，但选的不是这个老师,老师人没满");  
 }  
 else  
 {  
 TeacherStatus.add("已选，但选的不是这个老师，且该老师已被选满");  
 }  
 }  
 }  
 else  
 {  
 if(teacher\_course.getCnt()<teacher\_course.getNum())  
 {  
 TeacherStatus.add("可选");  
 }  
 else  
 {  
 TeacherStatus.add("该老师已被选满（不可选）");  
 }  
 }  
 }  
 response.put("TeacherStatus",TeacherStatus);  
 response.put("success",true);  
 return ResponseEntity.*ok*(response);  
 }  
  
 @PostMapping("/showSelectCourse3") //执行选课操作  
 public ResponseEntity<Map<String,Object>> select(@RequestParam("sid") Integer sid,  
 @RequestParam("tname") String tname,  
 @RequestParam("cname") String cname)  
 {  
 Map<String, Object> response = new HashMap<>();  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname", cname);  
 Course course = courseService.getOne(courseQueryWrapper);  
 course.setCcnt(course.getCcnt() + 1);  
 courseService.updateById(course); //更新该课程的已选人数  
 int cid = course.getCid();  
 QueryWrapper<Teacher> teacherQueryWrapper = new QueryWrapper<>();  
 teacherQueryWrapper.eq("tname", tname);  
 Teacher teacher = teacherService.getOne(teacherQueryWrapper);  
 int tid = teacher.getTid();  
 Teacher\_Course teacher\_course = teacher\_courseService.getRelation(cid, tid);  
 int total = teacher\_courseService.updateCntById(cid,tid,teacher\_course.getCnt() + 1); //更新该老师这门课的学生 +1  
 Student\_Course student\_course = new Student\_Course();  
 student\_course.setSid(sid);  
 student\_course.setCid(cid);  
 student\_course.setTid(tid);  
 student\_courseService.save(student\_course); //记录该学生选了这个老师的这门课  
 response.put("success", true);  
 response.put("message", "选课成功！");  
 return ResponseEntity.*ok*(response);  
 }  
  
 @PostMapping("/searchSelectCourse") //选课时根据学生搜索的课程名称返回课程信息+授课教师  
 public ResponseEntity<Map<String, Object>> Search(@RequestParam("sid") Integer sid,  
 @RequestParam("cname") String cname) {  
 Map<String, Object> response = new HashMap<>();  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname",cname);  
 Course course1 = courseService.getOne(courseQueryWrapper);  
 int cid = course1.getCid();  
 List<String>Info = new ArrayList<>();  
 Info.add(course1.getCaddress());  
 Info.add(course1.getCtime());  
 Student\_Course student\_course = student\_courseService.getRelation(sid,cid);  
 if(student\_course!=null) //学生已经选了该门课  
 {  
 Info.add("已选");  
 }  
 else  
 {  
 if(course1.getCcnt()<course1.getCnumber()) //学生未选该门课，且该门课人数还没满  
 {  
 Info.add("可选");  
 }  
 else  
 {  
 Info.add("已选满"); //学生未选该门课，但该门课人数已满  
 }  
 }  
 response.put("success",true);  
 response.put("Info",Info);  
 return ResponseEntity.*ok*(response);  
 }  
}



