Web框架编程——期中考核报告

项目完成人：许健（单独完成）

1. 工具与框架：

环境：IDEA2020,Tomcat9,Mysql8

前端：html+css+js,juquey框架

后端：java,SSM框架

二．需求分析：学生选课系统

基本说明： 要求具有学生、教师、管理员角色，并实现如下基础功能

1. 学生管理：学生的增删改查
2. 课程管理：课程的增删改查
3. 教师管理：教师的增删改查，一名教师可以教授1~n门课程
4. 选课管理：选课的增删改查，每位学生可以选择1~n门课程
5. SSM框架

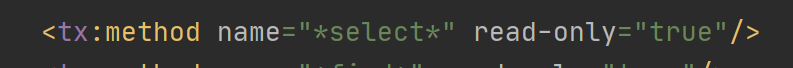
Spring是service层的框架，SpringMVC是视图层的框架,Mybatis是mapper层的框架

I)Spring特性：

1）控制反转：即不同对象之间的调用由原来的主动调用变成后来的被动创建和调用。打个比方，比如在service层的一个service类的每个方法都要调用mapper层的对象，如果不使用控制反转，每个方法要使用mapper的方法都要单独创建一个mapper对象，编码麻烦而且耦合度高，但如果使用控制反转，那么直接把要使用的mapper对象注入到这个service类中，然后每个service方法都可以直接调用注入到类中的这个mapper对象。

2）面向切面：不修改源代码的情况下，通过配置文件，动态的对一些方法进行功能的增强(调用其它方法进行增强)，增强的时机根据配置的通知类型来决定。

例如：



即名字中带有select的事务都会被增加一个只读属性

II)SpringMVC:

1. MVC模式：模型model(javabean)、视图view(jsp/img)、控制器Controller(Action/servlet)
2. SpringMvc原理：

在没有使用SpringMVC之前我们都是使用Servlet在做Web开发。但是使用Servlet开发在接收请求参数，数据共享，页面跳转等操作相对比较复杂。servlet是java进行web开发的标准，既然springMVC是对servlet的封装，那么很显然SpringMVC底层就是Servlet，SpringMVC就是对Servlet进行深层次的封装。

1. SpringMvc作用：将前台和后端代码分离

III)Mybatis:

1）支持定制化 SQL、存储过程以及高级映射。

2)避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。  
3)可以使用简单的 [XML](https://so.csdn.net/so/search?q=XML&spm=1001.2101.3001.7020) 或注解来配置和映射原生类型、接口和 Java 的 POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

三．程序结构

I)配置文件

1）



用于连接数据库

2）



用于创建数据源

配置Mybatis的核心配置文件

设置pojo与mapper文件的扫描器。

3）



用于设置service文件的扫描器

设置事务管理器

添加事务的切面

完成切面和切入点的织入

4）



设置controller文件的扫描器

设置注解驱动

设置视图解析器

5）



用于分页插页的配置

6）



添加字符编码过滤器：统一转换字符编码

注册Spring框架

注册SpringMVC框架

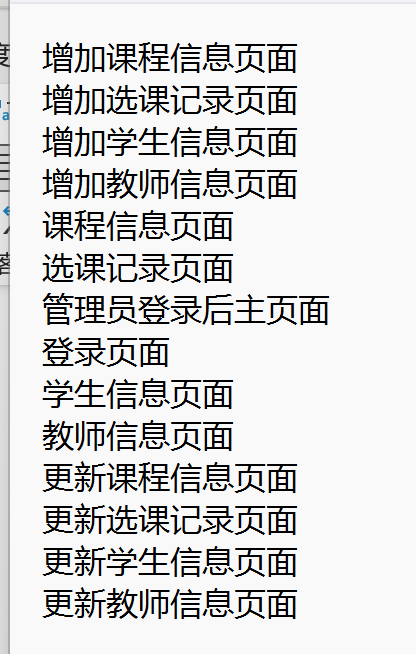
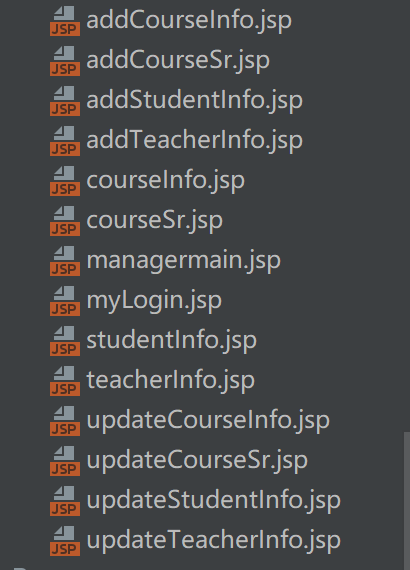
7）



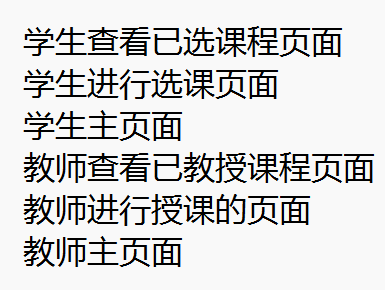
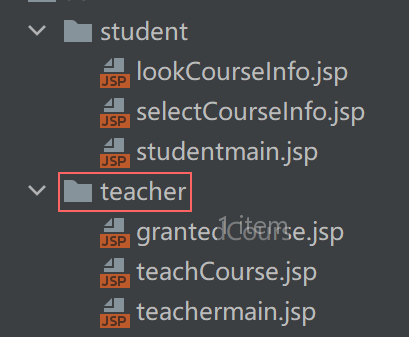
使用maven导入所使用的包

II)前端代码结构

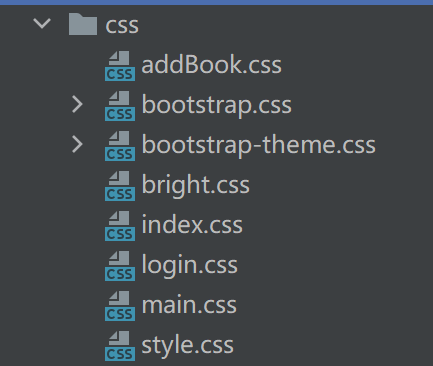
1）管理员相关页面



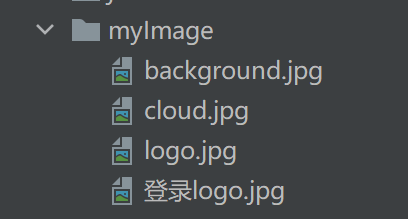
2）教师和学生相关页面



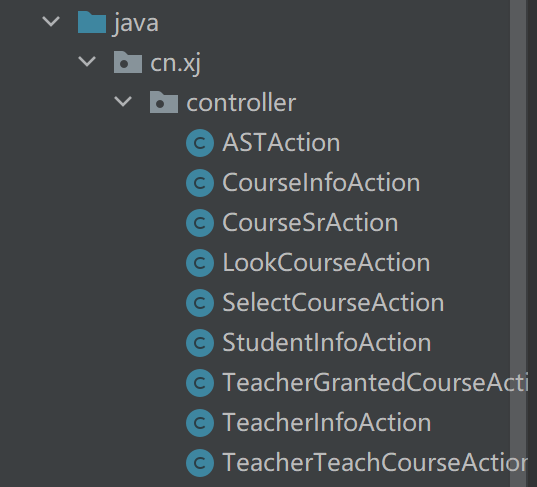
1. css相关：

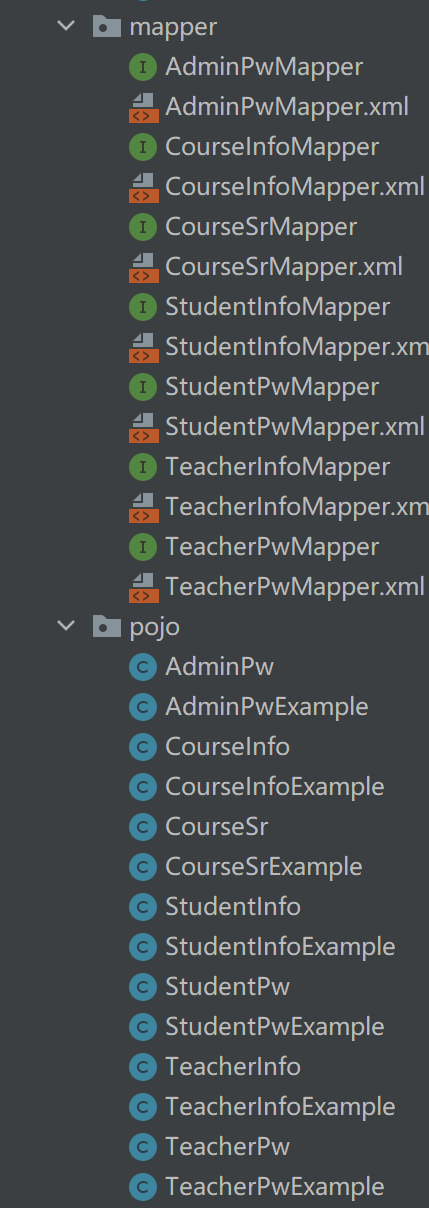


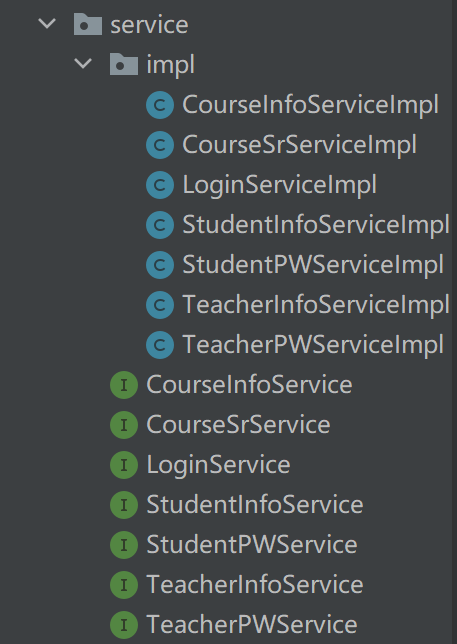
4）所使用图片资源



III)后端代码结构：



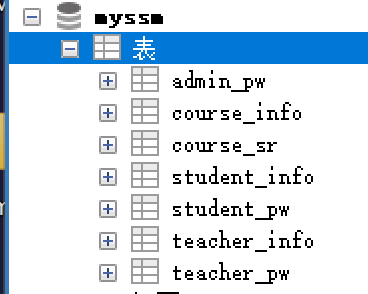




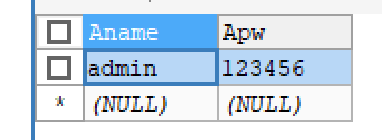
1. controller层用于和前端页面进行数据交互，并且调用service层提供的服务
2. service层为controller层提供服务，并调用mapper层提供的服务
3. mapper层用于封装数据库的操作，并为service层提供服务
4. pojo层为其他层提供实体类对象，为mapper层提供sql语言字段服务。

四．数据库表的设计：

总体：

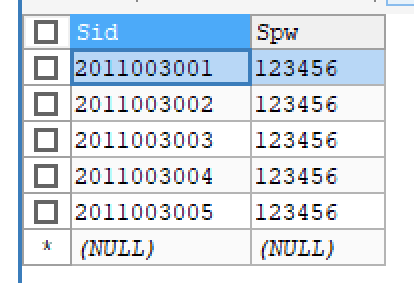


1. 



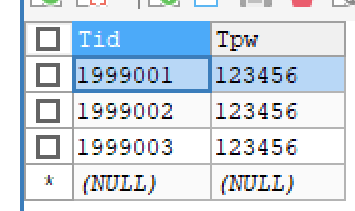
记录着管理员的账号名称和密码。

1. 



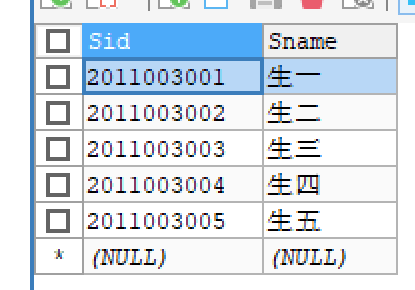
记录着学生的登录账号（学号）和密码

III）



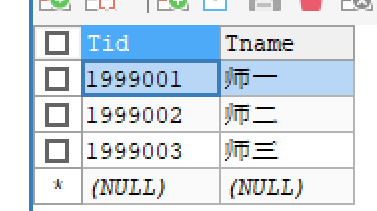
记录着教师的登录账号（工号）和密码

IV)



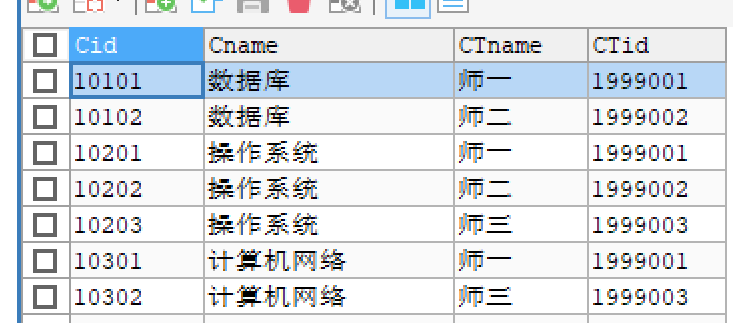
记录着学生的学号和姓名

V) 



记录着教师的工号和姓名

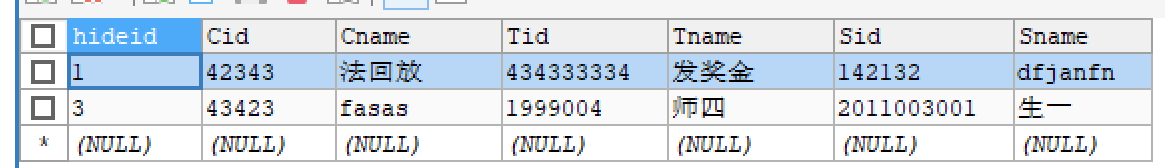
VI) 



记录着课程号，课程名，教这门课的教师姓名和教师工号，若没有教师教授这门课程，那么教师姓名和教师工号均为“0”

VII)





记录这选课记录（区别选课记录的隐藏id，被选课程的id和名，教授教师的工号和姓名，选课学生的学号和姓名）。

五．设计思路：

I）设计三中角色，管理员，老师，学生，都可以通过登录验证后再进行各自的操作。

II）管理员可以进行的行为：

1. 对学生信息的增删改查，操作基于表student\_pw和student\_info;
2. 对教师信息的增删改查，操作基于表teacher\_pw和teacher\_info;
3. 对课程信息的增删改查, 操作基于表course\_info;
4. 对选课记录的增删改查,操作基于表course\_sr
5. 教师可以进行的行为:
6. 查看自己已经教授的课程，操作基于表course\_info
7. 选择退授这门课程，操作基于表course\_info和表course\_sr
8. 选择教授某门课程，操作基于表course\_info
9. 学生可以进行的行为:

1）查看自己已经选择的课程，操作基于表course\_sr

2) 选择退选这门课程，操作基于表course\_sr

3）选择某门课程，操作基于表course\_info和表course\_sr

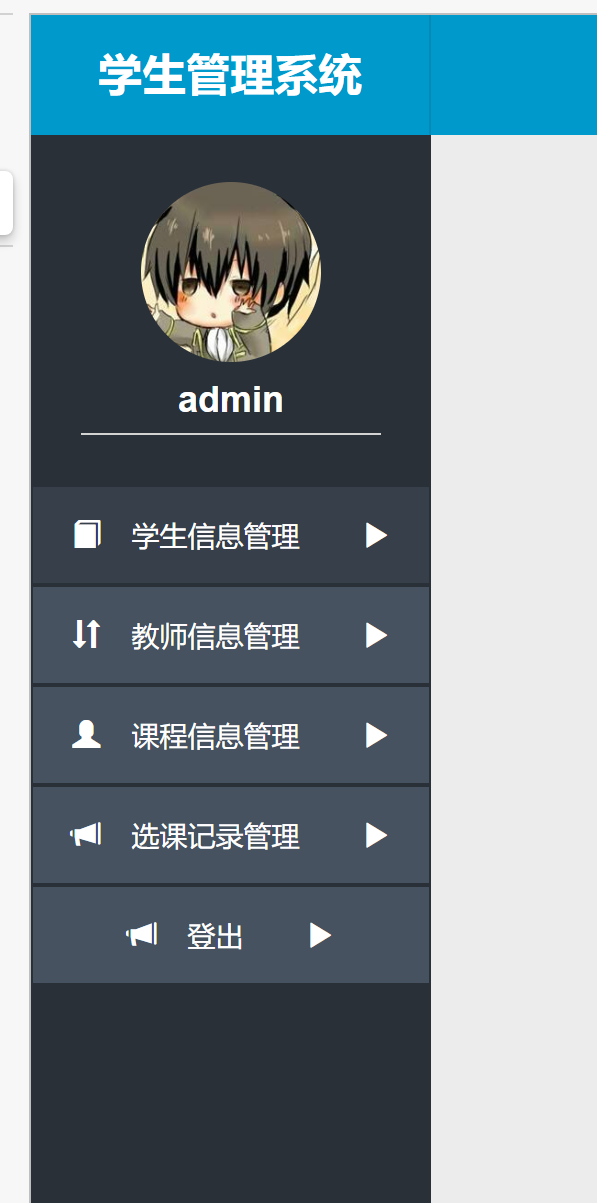
六．操作演示

I）登陆页面，admin,教师，学生的密码均是123456



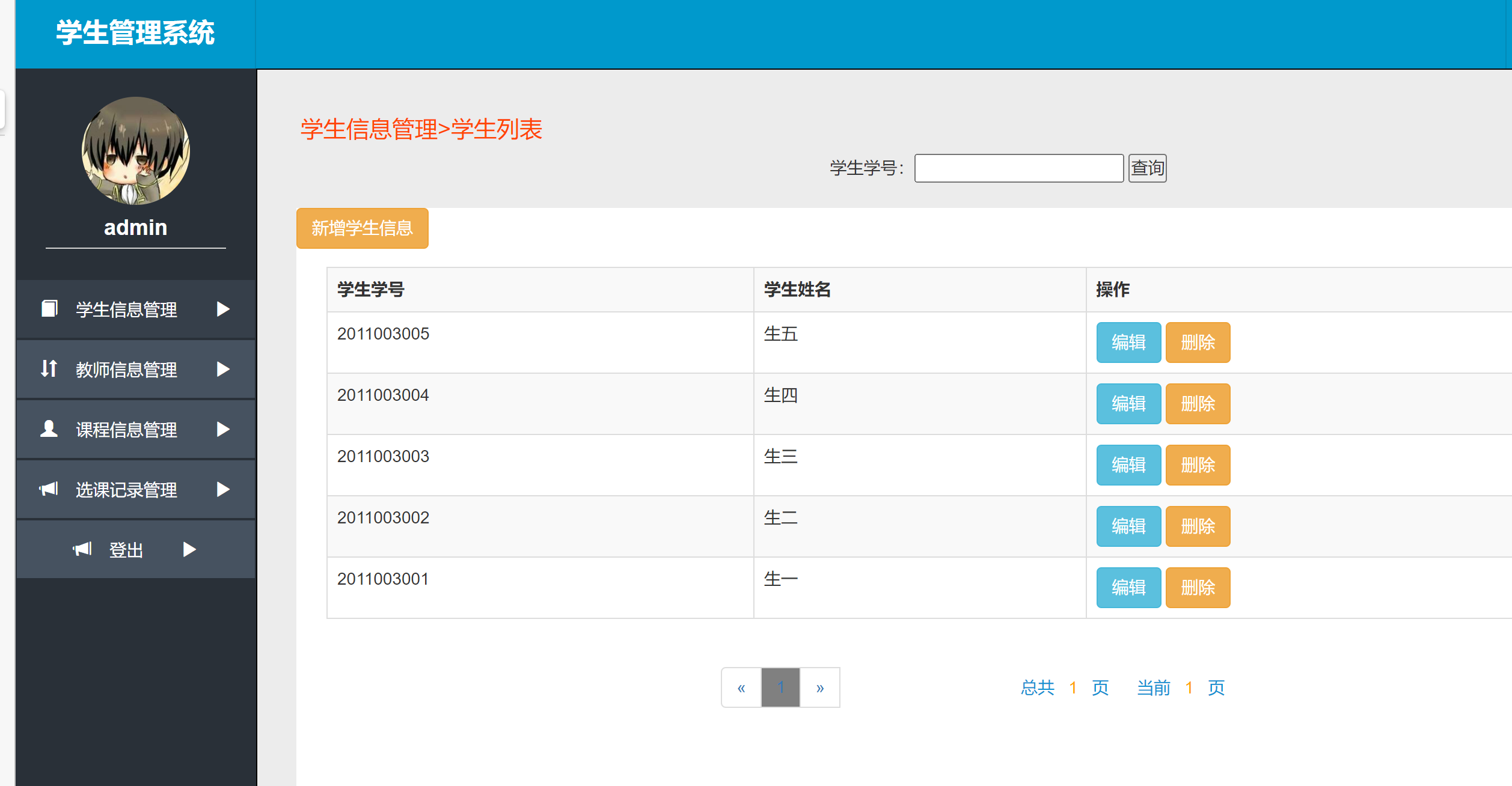
II)admin操作

1. 主页面

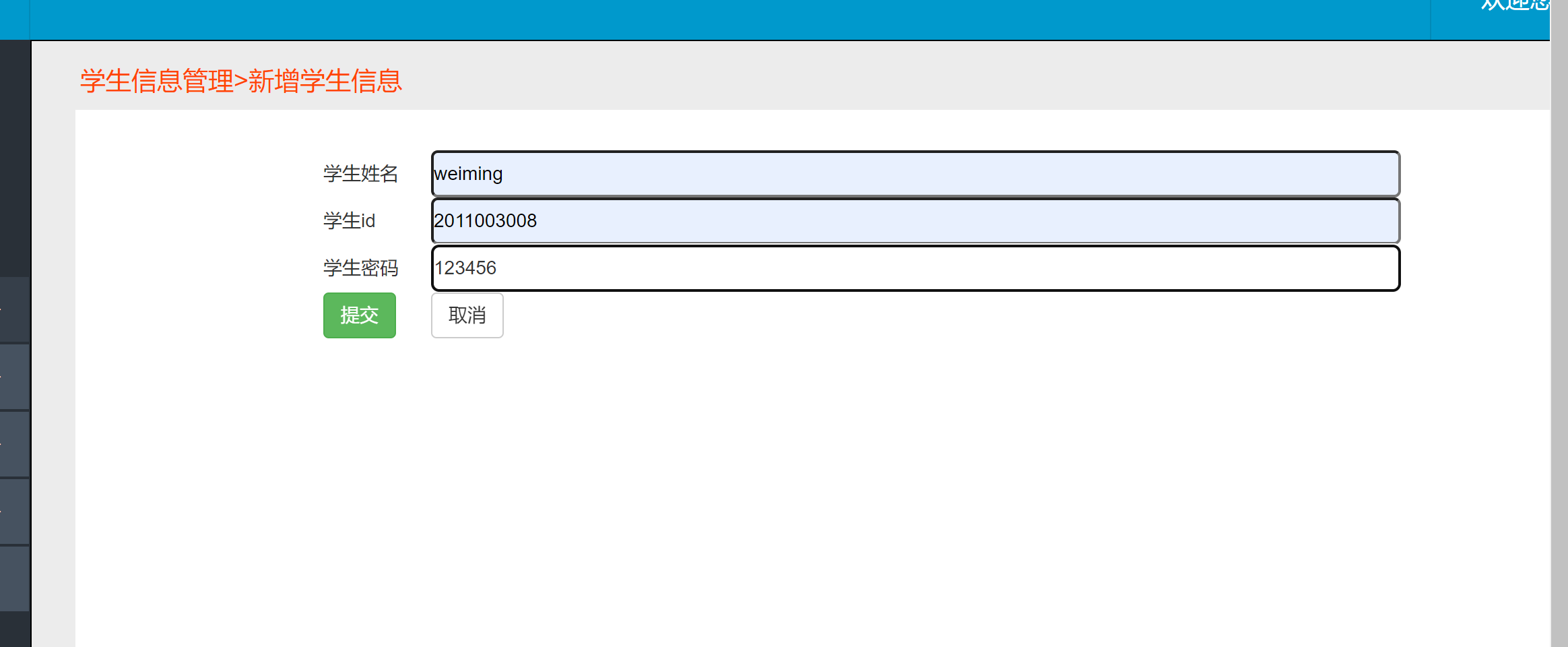


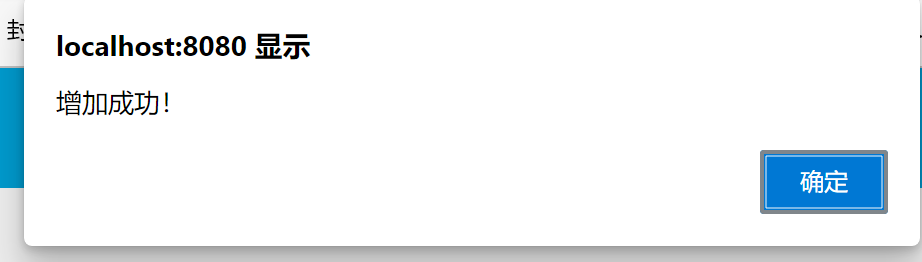
2）学生信息管理（教师信息管理，课程信息管理，选课记录管理操作相同）

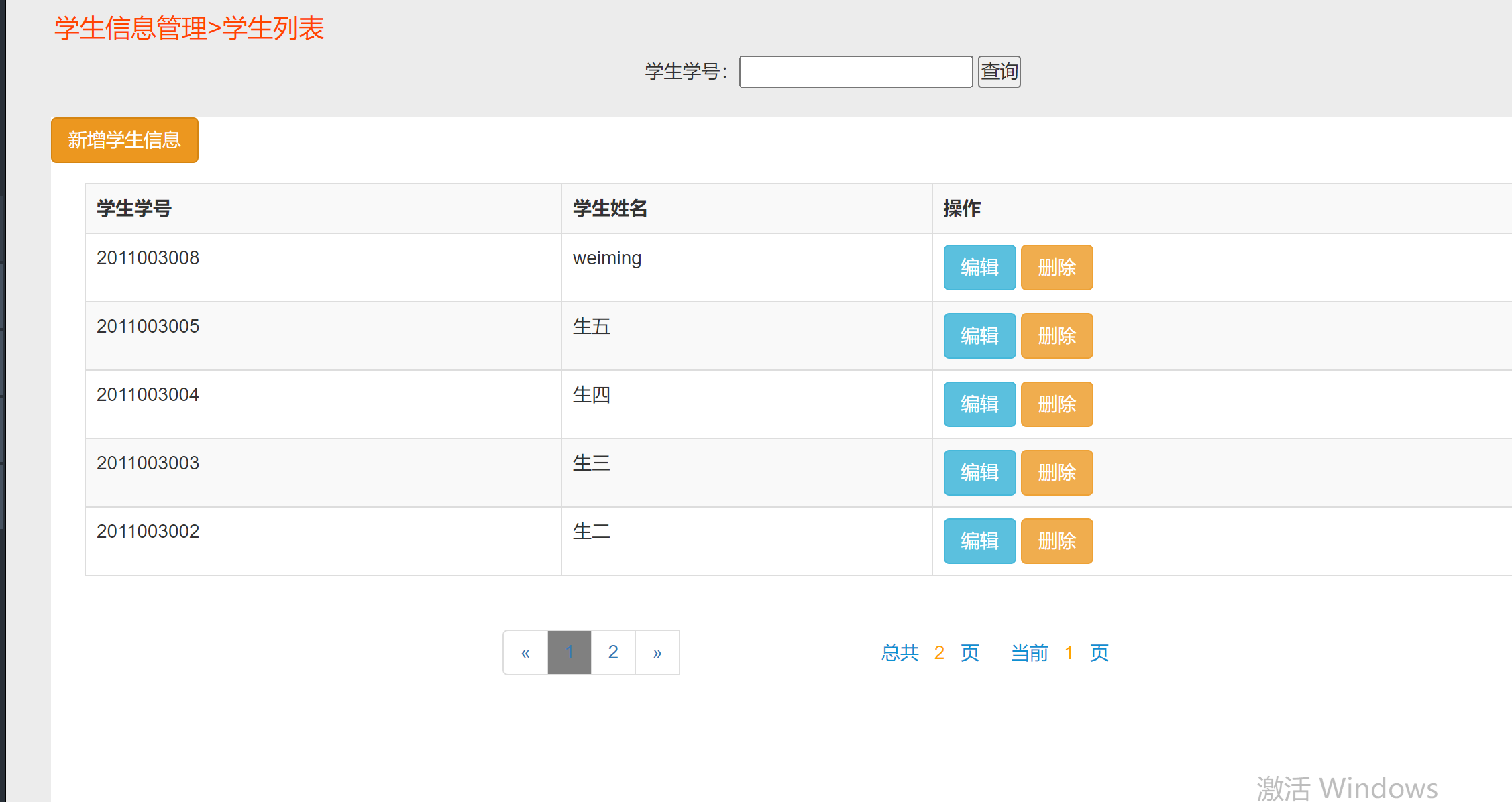
点击学生信息管理之后显示如下:



可以新增学生信息:



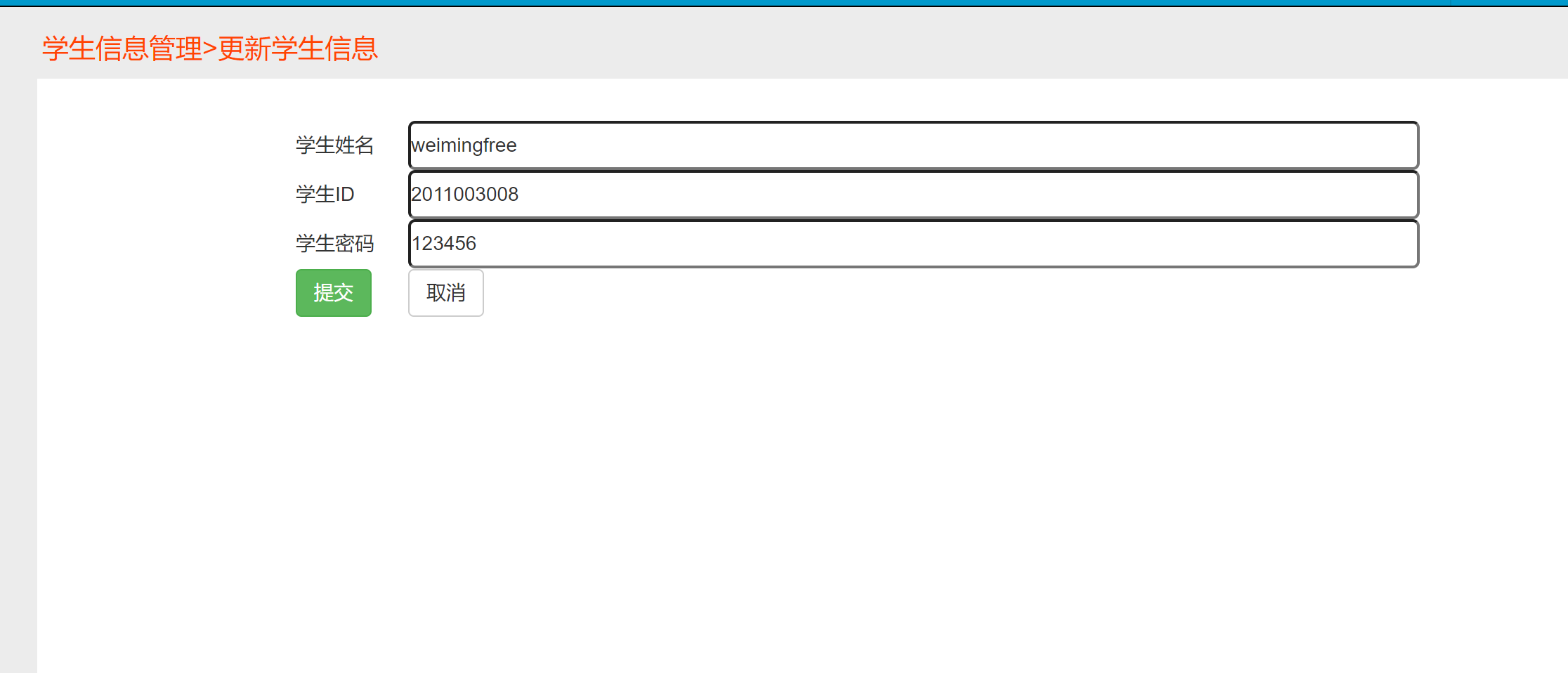


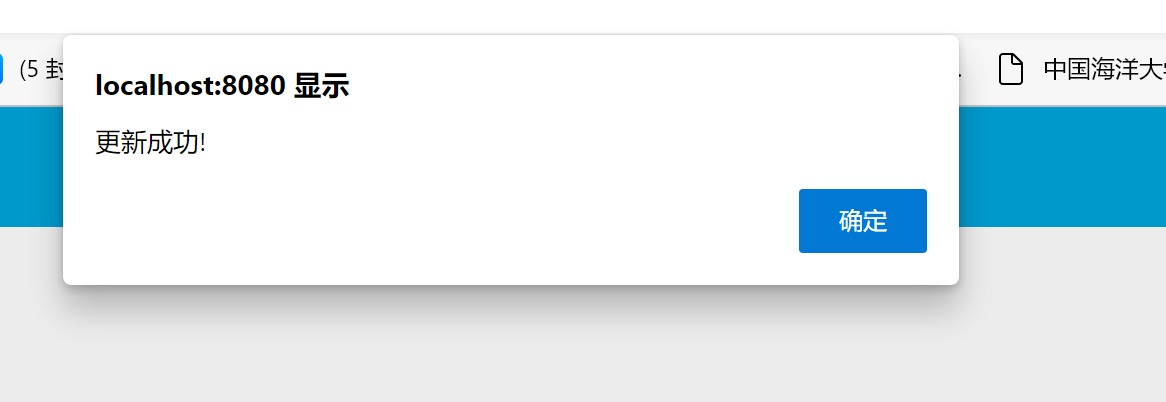


通过学号查询



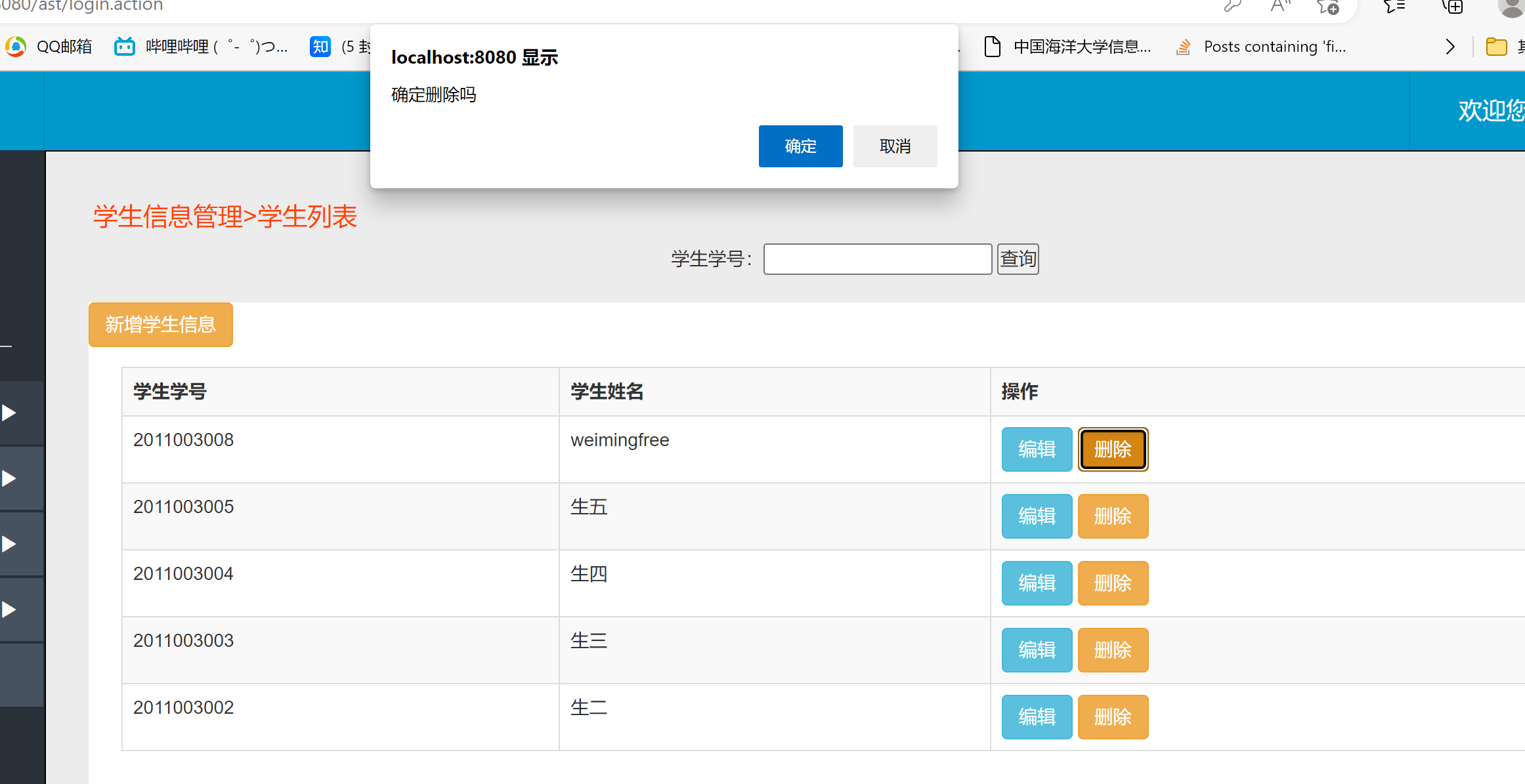
编辑此学生的信息,将此学生的姓名修改，注意：主键（这里是学生ID）是无法修改的:







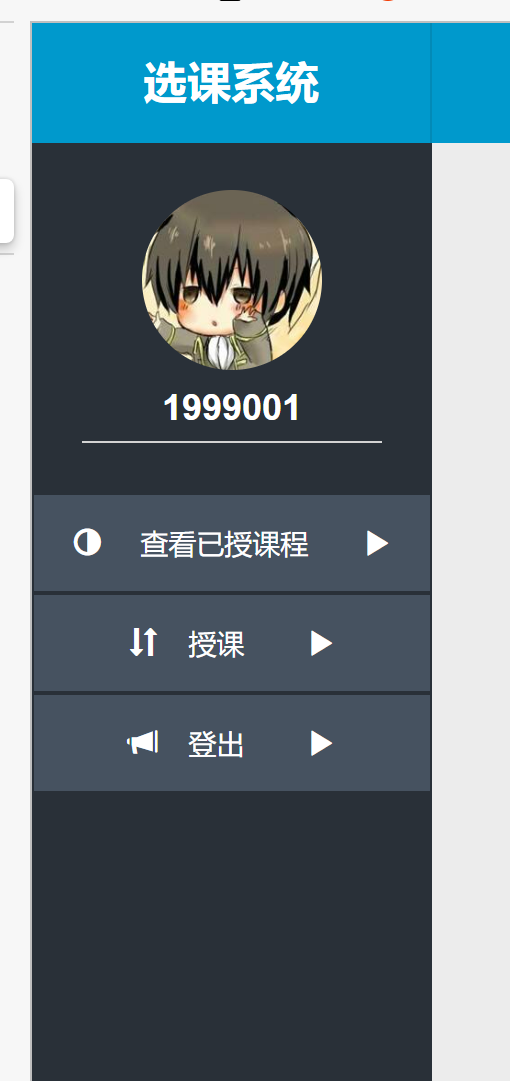
删除此学生的信息:





III）教师操作

1. 主页面:



1. 查看已授课程



这里点击退授之后学生的选课页面就不会显示这门课，同时这门课对应的选课记录也会被删除掉，但管理员的课程管理仍然会有这门课程，但是教师名和教师号都会被置0。

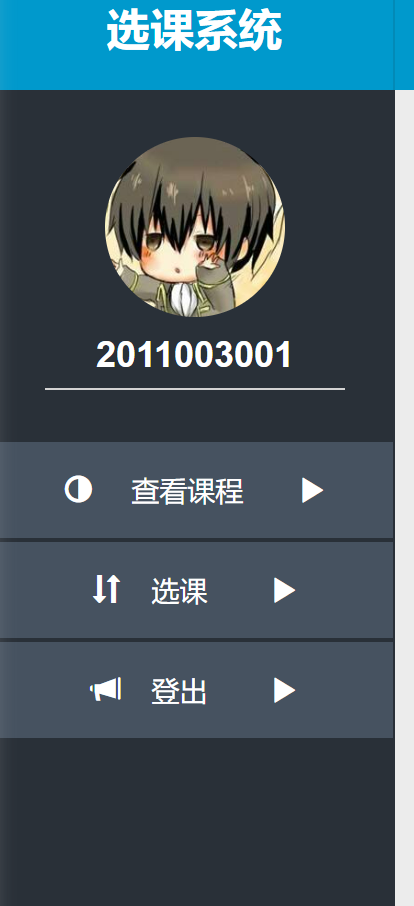
1. 授课页面:



这里只允许点击授课教师名和教师号为空的课程，否则的话会授课失败

III)学生操作:

1. 主页面:

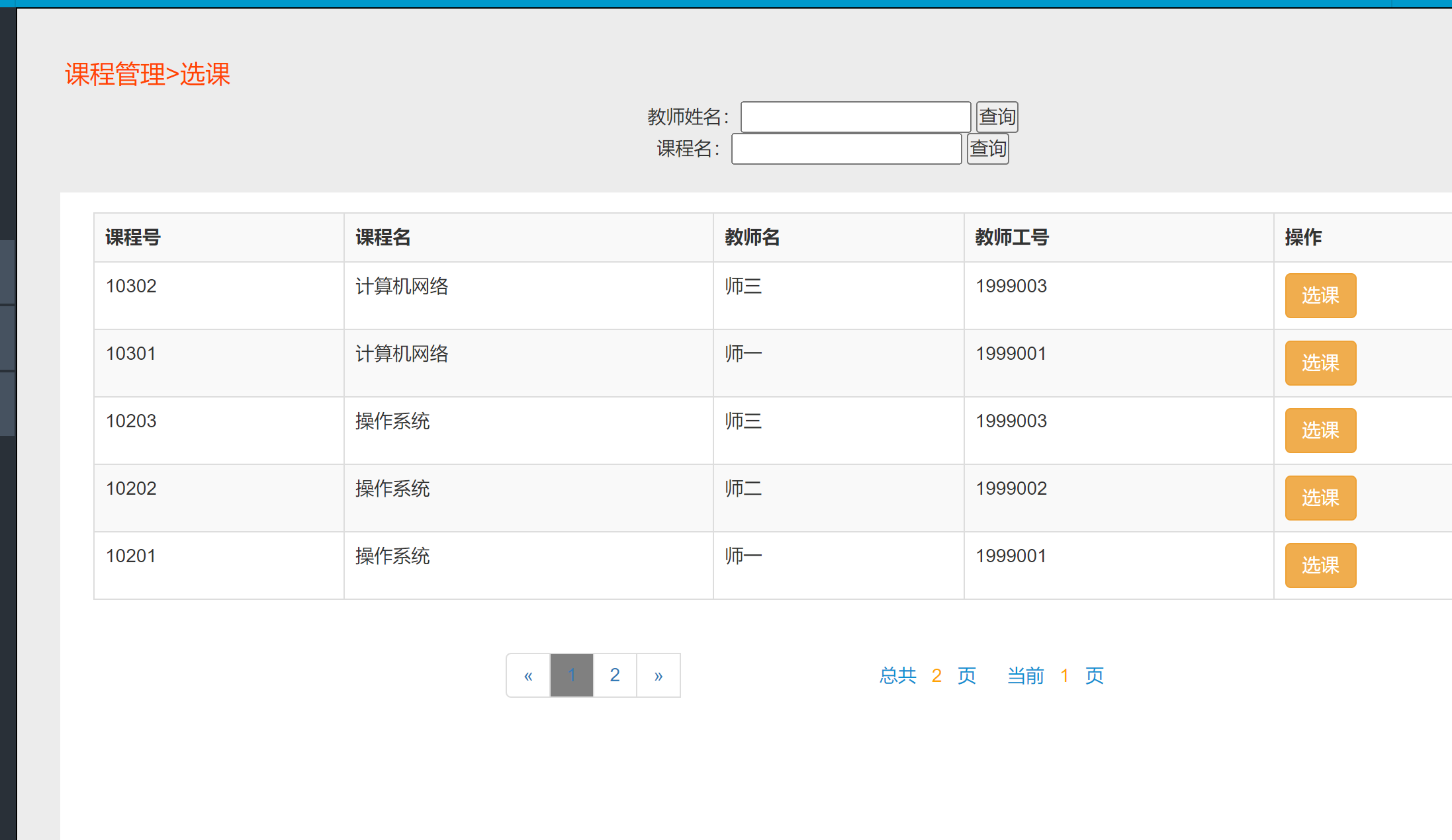


1. 查看课程:



显示此学生已选的课程，支持通过教师姓名和课程名分别查询，支持退选操作，退选会让这条选课记录被删除掉

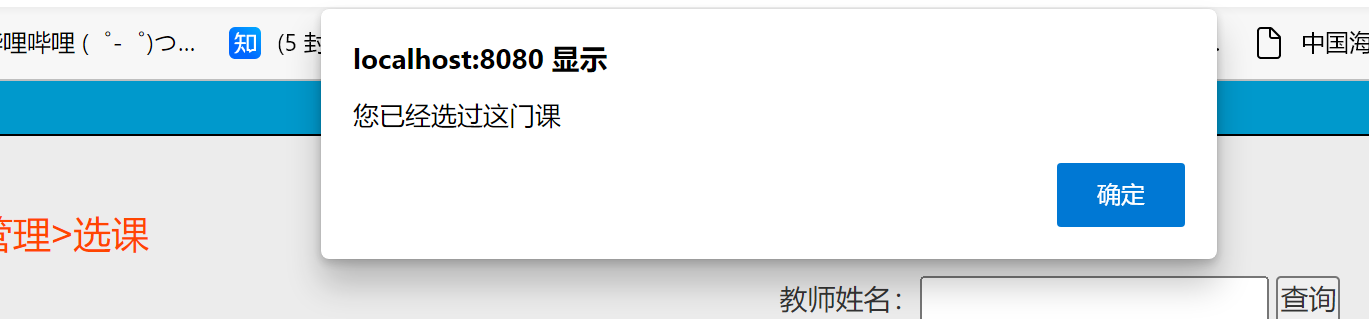
1. 选课页面:



这里可以进行选课，但是不能重复选择一门课，否则会选课失败，比如此时该学生已选择此门课：



若重复选择这门课，会进行提示:



七．典型代码展示（以学生信息为例）:

I)前端：（仅学生信息页面的js代码）

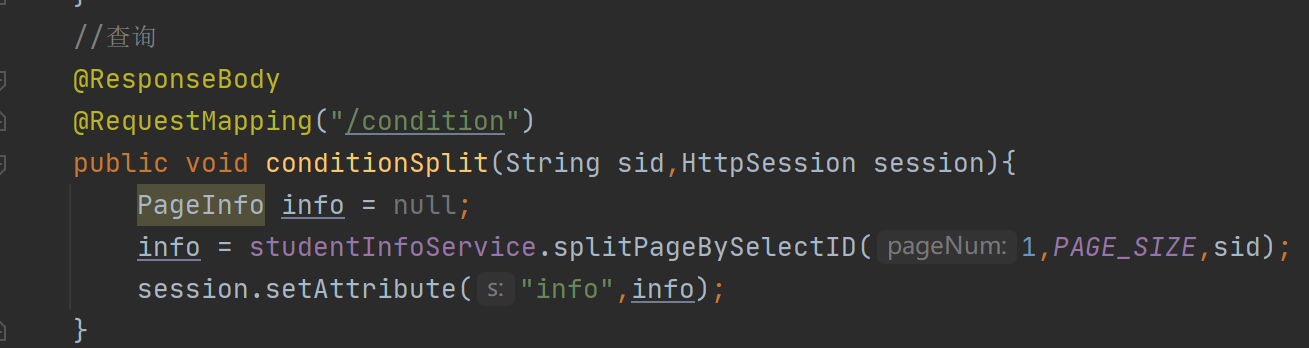




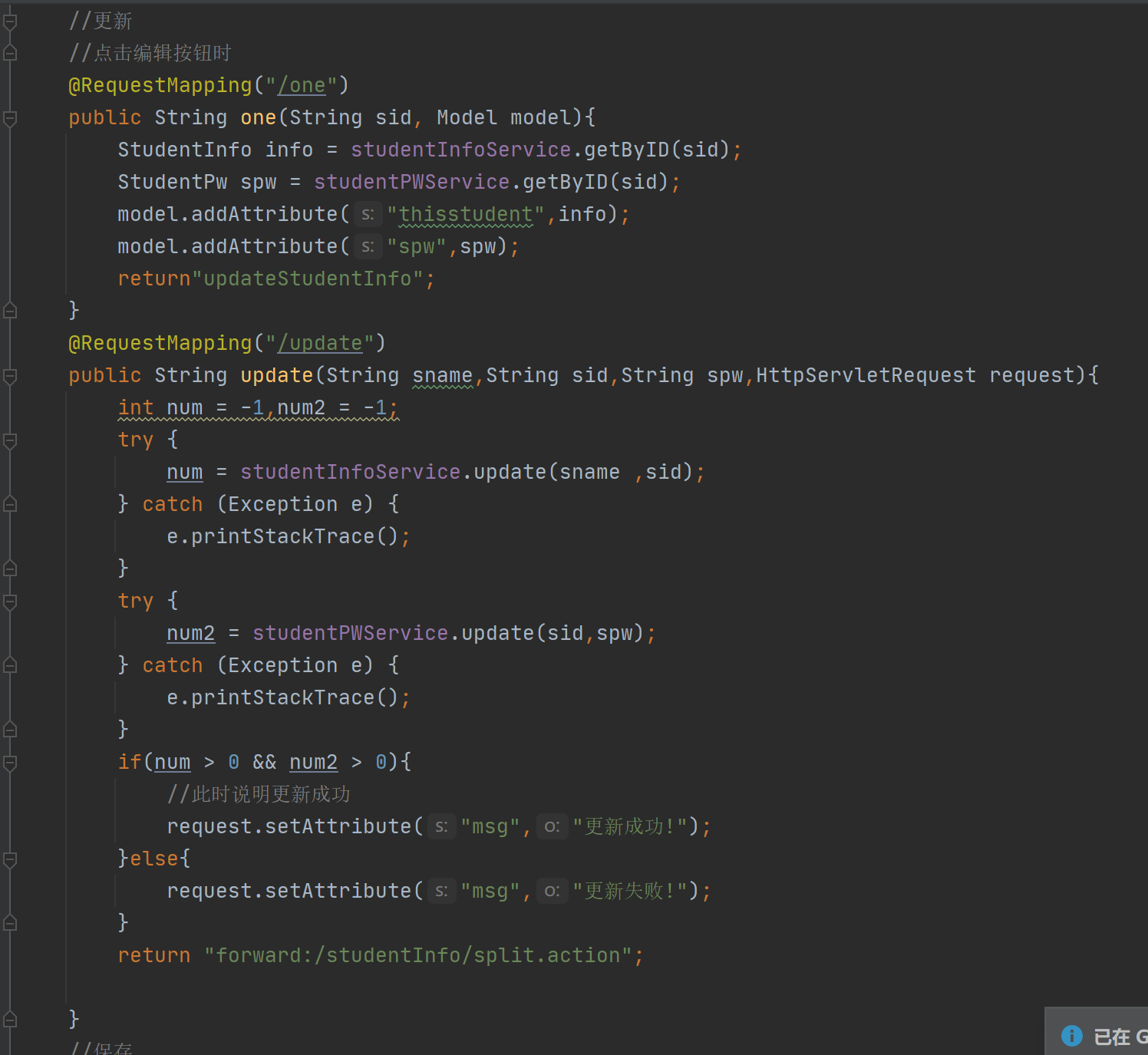


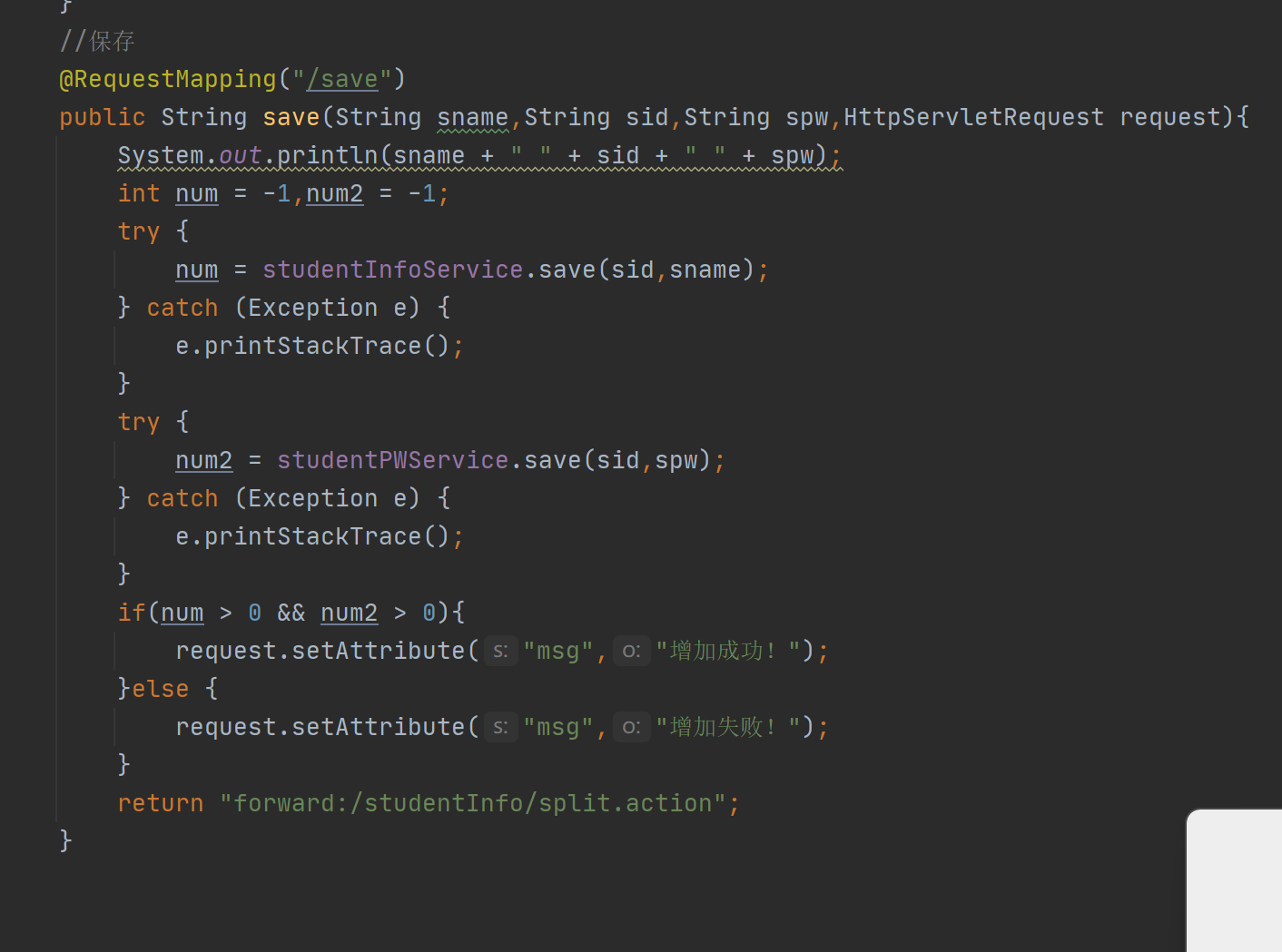
II)control层：



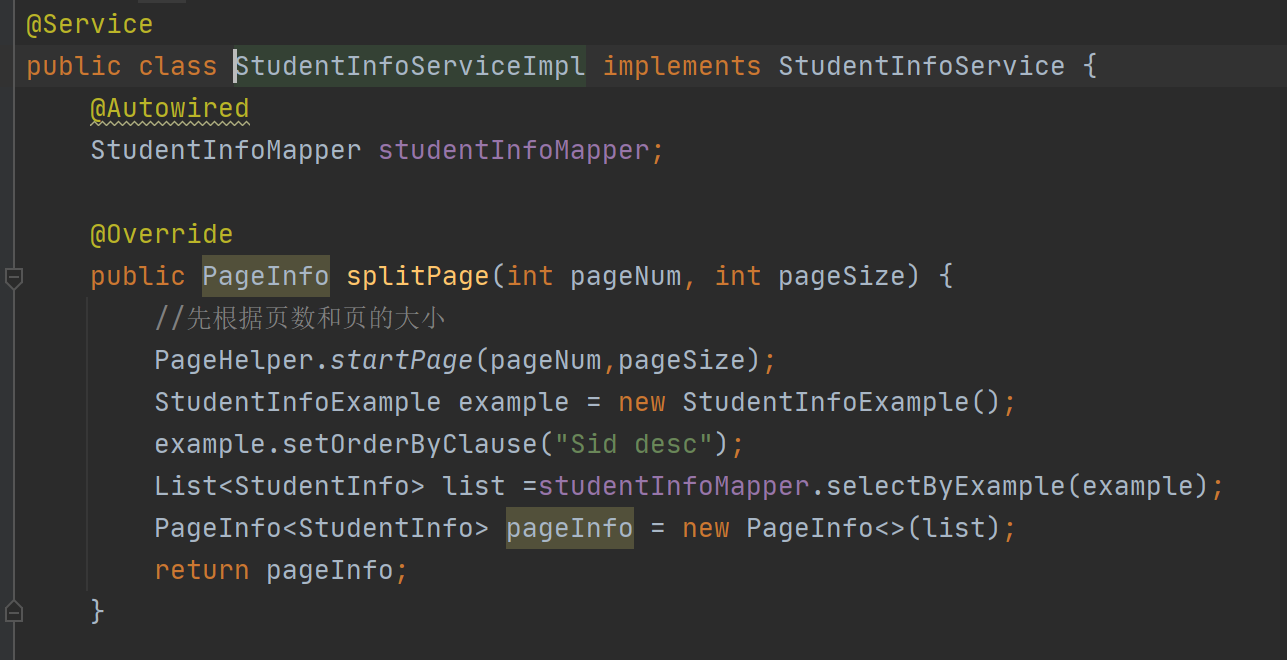


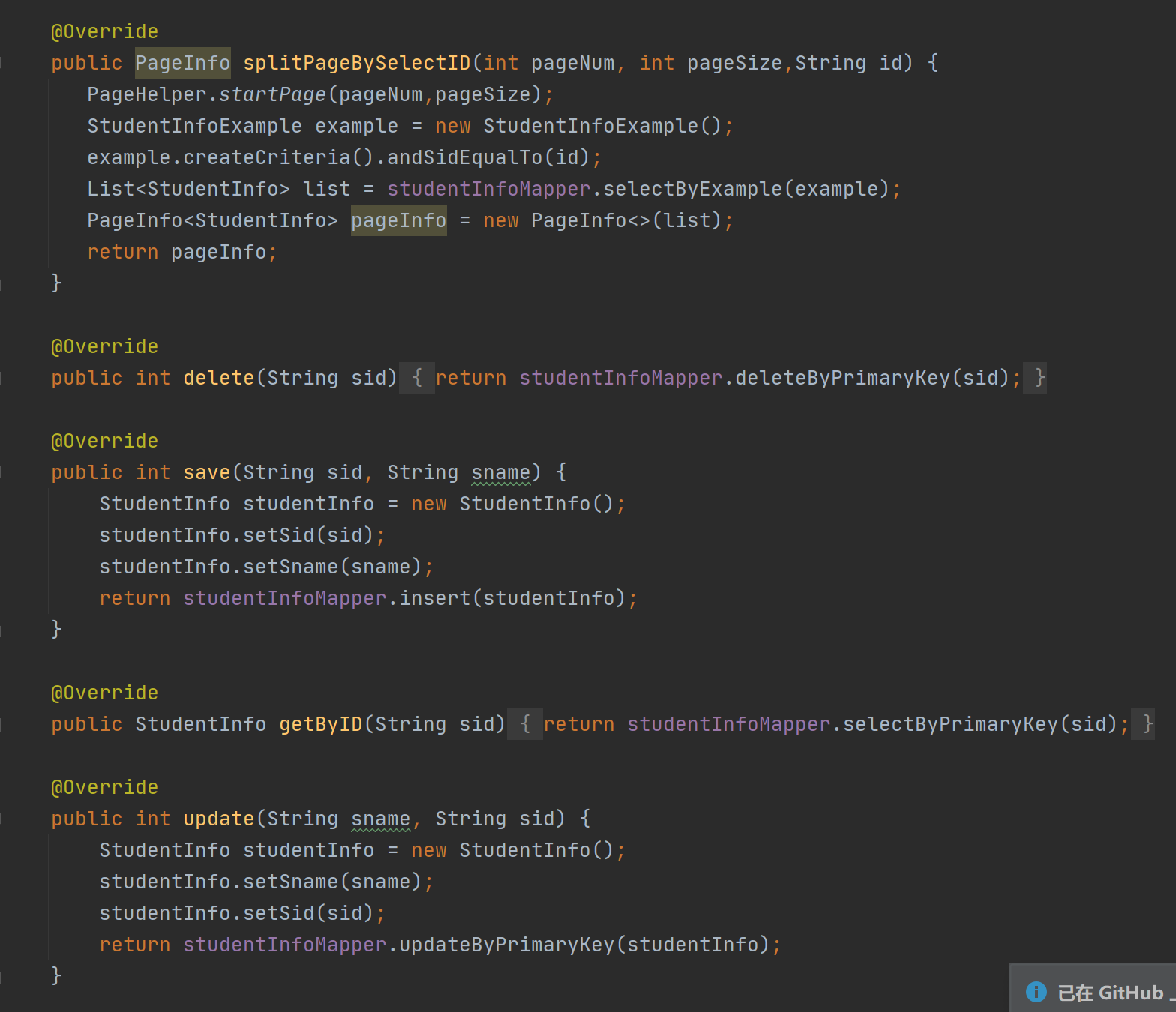






III)service层:





八．项目总结:

I)项目地址:

[weimingfree/myssm (github.com)](https://github.com/weimingfree/myssm)

II)个人进步:

在这门课之前我是java与web页面均是零基础，通过老师上课的讲解与课下b站视频的自学学到了很多东西:

1. 通过ssm框架开发简易网站的流程:数据库设计—>ssm文件的配置—>mapper,pojo层的生成-->service层，controller层的书写->与前端页面的对接;前端页面的设计->前端页面的书写->前后端的对接。
2. 熟悉了前端的html+css+js语言，以及后端的java语言
3. 熟悉了mysql，sqlyog，IDEA，tomcat工具的使用
4. 熟悉了将IDEA上的项目源码上传到github的流程

III)个人感悟:

1. 自我感觉数据库设计和前端页面设计这两个步骤比较重要,因为自己想表现出什么东西，想实现什么功能,这些功能会涉及到什么数据，什么数据和什么数据应该绑定在一个表内，前后端相互传递的参数有哪些以及它们应该怎么传递，这些在项目的开始就应该考虑清楚，否则在项目一半再进行更改就会比较麻烦。
2. 在做完这个项目之后我发觉简易网站的搭建并不困难，只是可能前端页面的书写要相对繁琐一些，如果繁琐，那么就可能出错。在我书写代码的过程中，较于后端，前端的页面跳转中更容易出错，有些时候前端测试中点击没有反应，竟是因为要跳转的页面名称忘记改变，这种错误出现了很多次，因为一个页面会有很多能跳转的地方
3. 就此项目而言，自我感觉工具的配置要比代码的书写更为困难，若是初学者就尽可能与学习视频中的工具配置一致，若对视频中的代码完全不了解就尽量集中自己的注意力避免代码错误。
4. 形成一个自己熟悉的开发模板（包括ssm配置，前端设计），在以后的开发中可以直接拿来用，节省时间。
5. 反思:
6. 此次项目因为开始数据库设计比较简单，导致实现的功能比较简单，以后的话开发之前一定要提前想好所要实现的功能从而才能设计一个比较完备的数据库
7. 项目前端不是很美观，内容比较少，自己也仅限于读懂并在原基础上稍加修改扩展的地步，若要自己独立设计出一个美观且内容丰富的页面还是有一些距离
8. 对mybatis生成的mapper层代码，主要是对数据库操作的mabtis化的sql语言不熟练，需要进一步加强，否则仅仅是selectExample(),

createCriteria（）是实现不了比较复杂的功能。