**数据库应用系统设计**

**实验报告**

-------学生选课管理系统

计算机科学与技术学院计算机专业2020级2班

姓名：尹国泰 学号：202000120101

任课教师：彭朝晖 助教：

目 录

[一、系统开发平台 1](#_Toc209544163)

[二、数据库规划 2](#_Toc209544164)

[2.1 任务陈述 2](#_Toc209544165)

[2.2 任务目标 2](#_Toc209544166)

[三、系统定义 2](#_Toc209544167)

[3.1 系统边界 2](#_Toc209544168)

[3.2 用户视图 2](#_Toc209544169)

[四、需求分析 2](#_Toc209544170)

[4.1 用户需求说明 2](#_Toc209544171)

[4.1.1 数据需求 2](#_Toc209544172)

[4.1.2 事务需求 2](#_Toc209544173)

[4.2 系统需求说明 2](#_Toc209544174)

[五、数据库逻辑设计 2](#_Toc209544175)

[5.1 ER图 2](#_Toc209544176)

[5.2 数据字典 2](#_Toc209544177)

[5.3 关系表 2](#_Toc209544178)

[六、数据库物理设计 3](#_Toc209544179)

[6.1 索引 3](#_Toc209544180)

[6.2 视图 3](#_Toc209544181)

[6.3 安全机制 3](#_Toc209544182)

[6.4 其他 3](#_Toc209544183)

[七、应用程序设计 3](#_Toc209544184)

[7.1 功能模块 3](#_Toc209544185)

[7.2 界面设计 3](#_Toc209544186)

[7.3 事务设计 3](#_Toc209544187)

[八、测试和运行 3](#_Toc209544188)

[九、总结 4](#_Toc209544189)

[附. 参考文献 4](#_Toc209544190)

# 一、系统开发平台

1.1概述

本学生选课管理系统采用B/S架构，使用Html、Css、JavaScript进行前端界面开发，采用MySQL数据库，使用Java Servlet类和JSP进行数据库的访问。

1.2 开发工具：Eclipse IDE for Java Developers，MySQL Workbench 8.0，jdk-17.0.1

1.3 数据库：MySQL 8.0

1.4 操作系统：Windows 10

# 二、数据库规划

## 2.1 任务陈述

(1) 要求系统可以准确地记录和查询学生信息，包括学生的姓名、单位、年龄、性别以及身份证号码等。

(2) 要求系统可以准确地记录学生地每一次奖惩情况。

(3) 系统可以对学校的院系情况进行管理，包括设置学院名称、修改某学院某专业方向的名称等。

(4) 系统应该可以对基础数据进行维护。

(5)系统能够对开设的课程进行管理

(6)学生选课管理、考试（登记分数）、补考重修管理

(7)系统还应该提供强大数据统计、查询、报表生成以及打印等功能。

(8)用户权限管理

(9)异常处理

## 2.2 任务目标

本系统需要实现学生用户、教师用户、管理员用户三种用户的注册和登录，其中学生用户和教师用户需要管理员在系统录入信息后再用学号或教职工号注册，分别为每种用户实现以下各种功能：

2.2.1学生用户功能

（1）修改个人信息：可修改个人手机号、邮箱等；学号、姓名、身份证号、年龄等信息学生用户不可修改。

（2）选课：在选课阶段和退改选阶段，学生用户可按照名称、课程号等信息查询相关课程并进行选课，在选课阶段或退改选阶段结束前可以进行修改。

（3）退选课程：在退改选阶段，可进行已选课程的退选。

（4）查询考试成绩：查询各课程已公布的考试成绩，要实现成绩的打印、排序、求平均值等操作。

2.2.2教师用户功能

（1）修改个人信息：可修改个人手机号、邮箱等信息，教职工号、姓名、身份者、年龄等信息教师用户不可修改。

（2）修改授课信息：可以查询、添加、删除、修改自己所教课程的相关信息。

（3）录入成绩：考试后由教师录入成绩。

（4）查询选课同学信息：可以查询自己所教的每一门课程的学生，并且在录入成绩后可以查看每一名学生课程成绩，提供求平均值、打印等操作。

2.2.3管理员用户功能

（1）修改个人信息：可以修改所有自己的相关信息。

（2）学生管理：查询、添加、删除、修改所有学生信息。

（3）教师管理：查询、添加、删除、修改所有教师信息。

（4）课程管理：查询、添加、删除、修改所有课程信息。

（5）选课管理：查询、添加、删除、修改学生的选课情况；可以设置选课阶段或者退改选阶段

（6）院系管理：可以修改学校院系名称、添加专业、修改专业名称号码等相关信息。

# 三、系统定义

## 3.1 系统边界

系统边界描述数据库系统和企业信息系统的其他部分的接口，是信息系统内部构成元素与外部有联系实体之间的信息关系的描述与分割。它并不需要在它们之间划一条物理边界，而只需要弄清它们之间信息输入与输出的分割。

本数据库系统共包括学生信息、教师信息、课程信息、选课信息、学院信息、专业信息、授课信息及成绩信息八个模块。该信息系统的其他部分包括学生专业选择、教师配置、学生选课、学院设置、课程要求、教室分配等。

数据库系统的内部构成元素与外部其他部分之间的信息关系如下图所示：

课程信息

教师信息

学生信息

选课信息

专业信息

授课信息

成绩信息

学院信息

学生专业选择

学生选课

学院设置

课程要求

教师配置

教室分配

## 3.2 用户视图

（1）学生用户视图：

修改个人信息、选课、退选课程、查询考试成绩

（2）教师用户视图

修改个人信息、修改授课信息、录入成绩、查询选课同学信息

（3）管理员用户视图

修改个人信息、学生管理、教师管理、课程管理、选课管理、院系管理

# 四、需求分析

## 4.1 用户需求说明

### 4.1.1 数据需求

（1）学生基本信息记录：学号、姓名、性别、年龄、身份证号、手机号、邮箱、所属专业、密码

（2）教师基本信息记录：工号、姓名、性别、年龄、身份证号、手机号、邮箱、所属学院、密码

（3）管理员基本信息记录：工号、姓名、密码

（4）课程基本信息记录：课程号、课程名、学分、学时、先导课程号

（5）学院基本信息记录：学院编号、学院名称

（6）专业基本信息记录：专业编号、专业名称、所属院系

（7）选课基本信息记录：学号、课程号、成绩

（8）授课基本信息记录：课程号、授课教师工号

### 4.1.2 事务需求

1.数据录入：

（1）录入学生基本信息

（2）录入教师基本信息

（3）录入课程基本信息

（4）录入专业基本信息

（5）录入院系基本信息

（6）录入选课信息

（7）录入授课信息

（8）录入管理员信息

2.数据更新/删除：

（1）学生信息更新

（2）教师信息更新

（3）课程信息更新/删除

（4）专业信息更新/删除

（5）院系信息更新/删除

（6）选课信息更新/删除

（7）授课信息更新/删除

（8）管理员信息更新/删除

3.数据查看

学生用户视图下：

（1）学生个人信息查看

（2）可选课程信息查看

（3）个人选课信息查看

教师用户视图下：

（1）教师个人信息查看

（2）所授课程查看

（3）所授课程选课学生查看

管理员用户视图下：

（1）管理员个人信息查看

（2）所有学生信息查看

（3）所有教师信息查看

（4）所有课程信息查看

（5）所有专业信息查看

（6）所有院系信息查看

（7）所有选课信息查看

（8）所有授课信息查看

## 4.2 系统需求说明

该学生选课管理系统需要较强的数据处理功能，理论上应该能够容纳上万人的数据资料，并且在搜索方面理应具有较快的响应速度，能够处理多方面的数据请求。系统能够有效的处理各种异常，具有较好的健壮性。

**4.2.1 初始数据库大小**

（1）5个院系

（2）每个院系一到两个专业

（3）每个院系大约10名学生

（4）每个专业大约3名老师

（5）两名管理员

**4.2.2 网络和共享需求**

（1）必须能够支持至少1000名用户同时访问，需要考虑这么大数量并发访问的许可需求。

**4.2.3 性能**

高峰期：每年的选课阶段、补退选阶段

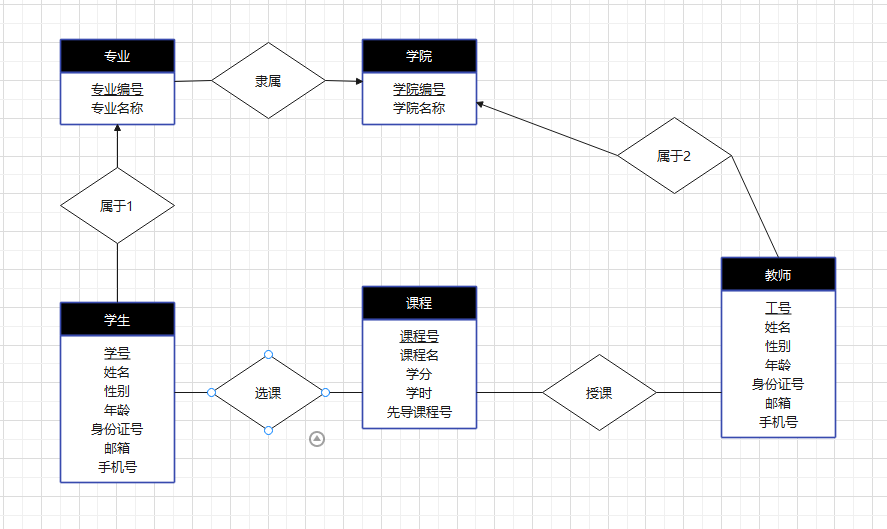
（1）单个记录查询时间少于1秒，高峰期少于5秒

（2）多个记录查询时间少于5秒，高峰期少于10秒

（3）更新/保存记录时间少于1秒，高峰期少于5秒

# 五、数据库逻辑设计

## 5.1 ER图



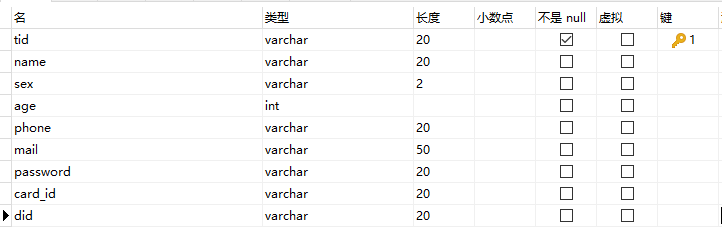
该ER图包括隶属、属于1、属于2、选课、授课五个联系。

## 5.2 数据字典

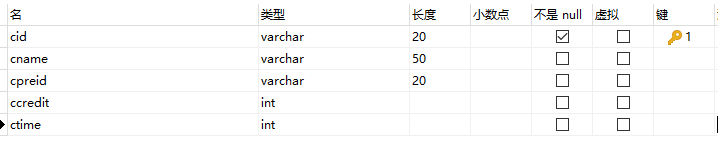
学生实体表：



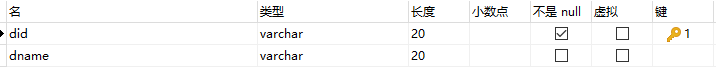
教师实体表：



课程实体表：



学院实体表：



专业实体表：



## 5.3 关系表

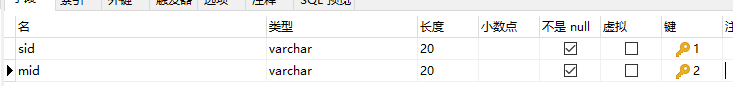
选课关系表：



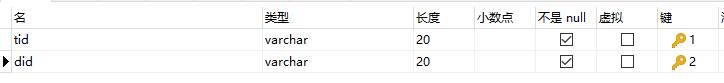
授课关系表：



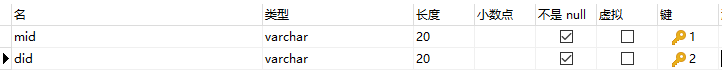
属于1：



属于2：



隶属：



# 六、数据库物理设计

## 6.1 索引

本数据库系统中为以下表中的属性建立索引：

|  |  |
| --- | --- |
| 建立索引的表 | 建立索引的属性 |
| 学生（student） | 学号（sid） |
| 教师（teacher） | 工号（tid） |
| 课程（course） | 课程号（cid） |
| 学院（department） | 学院编号（did） |
| 专业（major） | 专业编号（mid） |

## 6.2 视图

本数据库系统支持通过sql语句“create view … as …”来创建视图

## 6.3 安全机制

**6.3.1 系统安全**

1、本学生选课系统提供了充足的异常处理机制，能够处理输入类型错误、查询不规范等引起的错误

2、本系统登入需要用户名和密码来确保系统安全

**6.3.2 数据安全**

1、本系统中学生、教师、管理员用户拥有不同权限，学生、教师用户对某些数据（如课程专业信息等）不能进行更改，以此确保数据安全。

## 6.4 其他

引入受控冗余：虽存在相应关系表，但在学生实体表中仍保留所属专业这一属性，在教师实体表中仍保留所属学院这一属性，以加快检索速度、方便形成视图。

# 七、应用程序设计

## 7.1 功能模块

本学生选课系统共设置学生、教师、管理员三种用户，各种用户可以实现的功能如下：

**7.1.1学生用户功能**

（1）学生个人信息查看修改

（2）密码修改

（3）可选课程信息查看、选课，可以通过课序号、开课院系等信息搜索

（4）已选选课程信息查看、退课，可以通过课序号、开课院系等信息搜索

（5）考试成绩查看

**7.1.2教师用户功能**

（1）教师个人信息查看修改

（2）密码修改

（3）所授课程信息查看

（4）学生成绩录入，可以根据课序号来查看学生

（5）所授课程选课学生查看，可以根据课序号来查看

**7.1.1管理员用户功能**

（1）管理员个人信息查看修改

（2）所有学生信息查看修改

（3）所有教师信息查看修改

（4）所有课程信息查看修改

（5）所有专业、院系信息查看

（6）学生选课信息查看

（7）教师授课信息查看、修改

（8）设置是否处于选课阶段

以上各种信息查看可通过学号、职工号、课程号、专业、院系等信息搜索

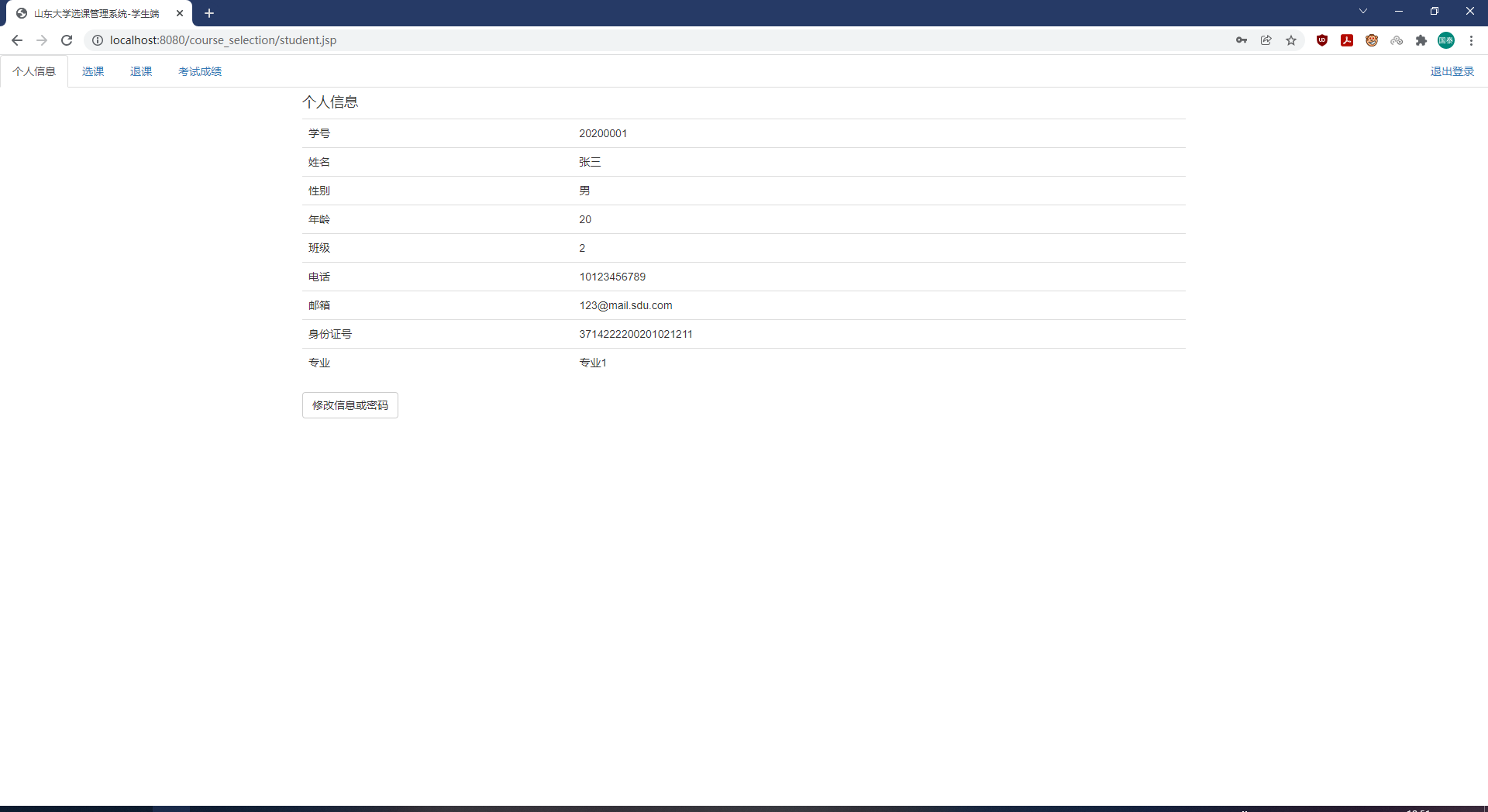
## 7.2 界面设计

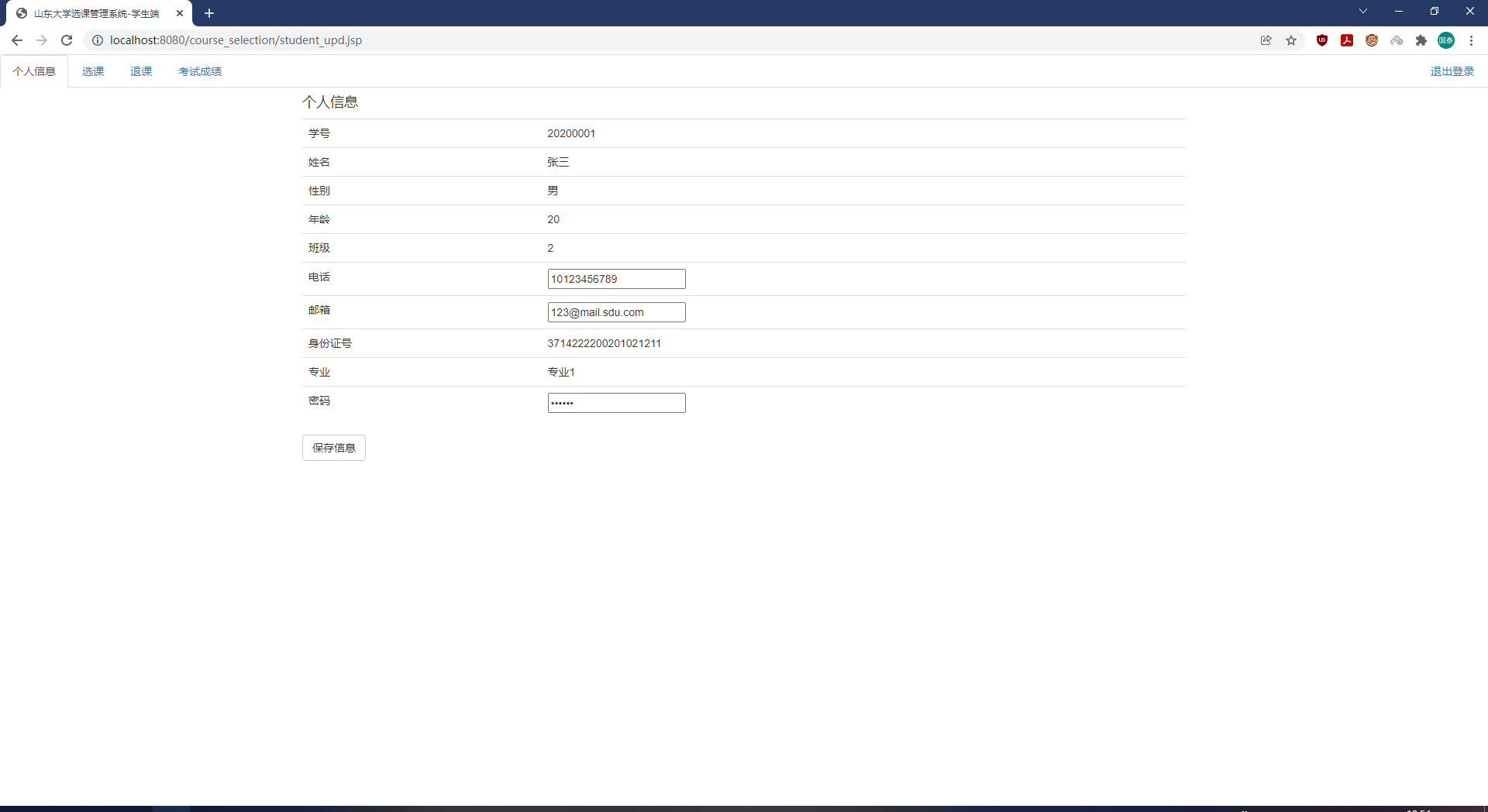
**7.2.1登录界面**

可通过左下角下拉菜单来选择用户类型

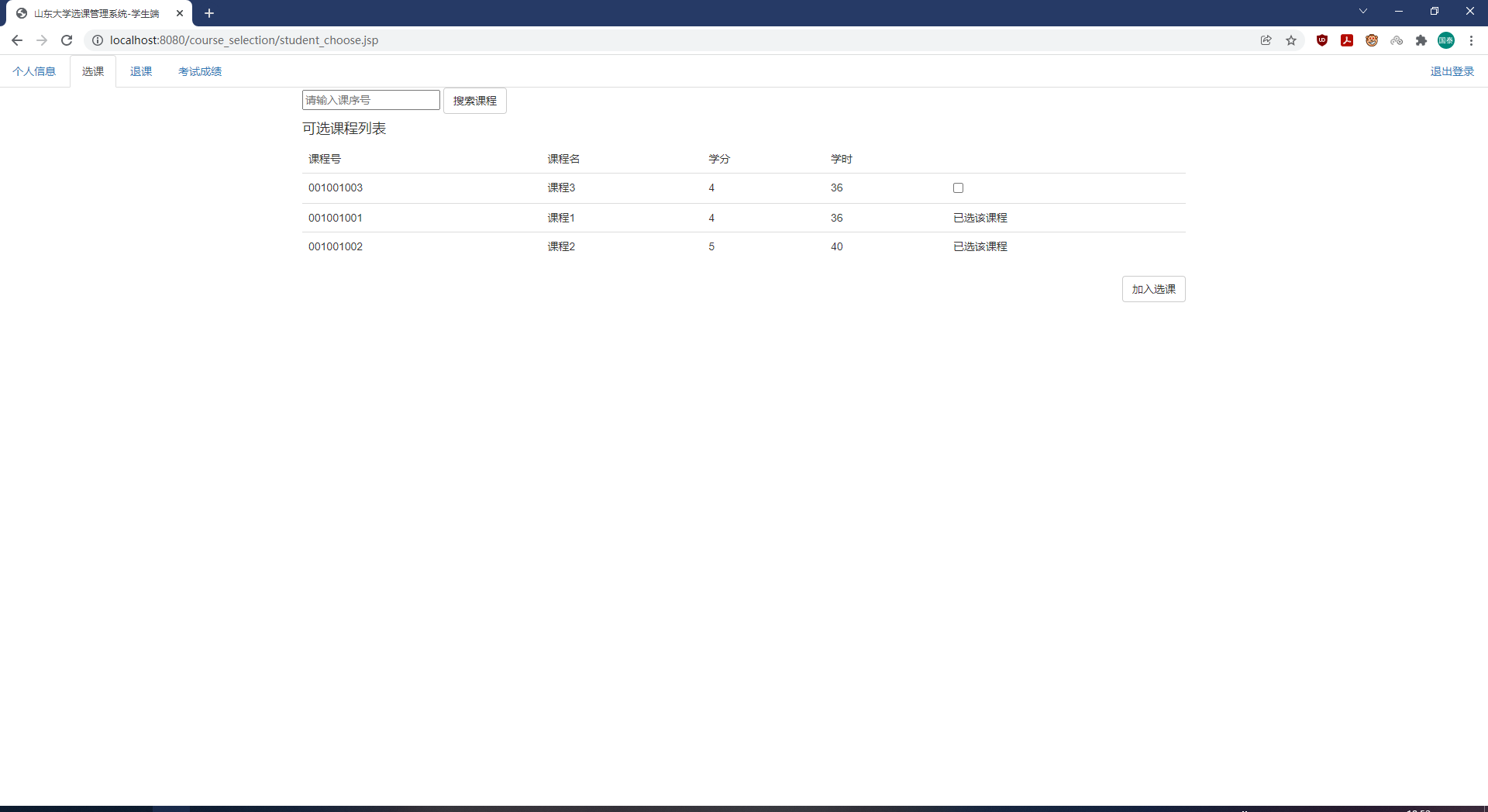


**7.2.2学生个人信息查看和修改界面**

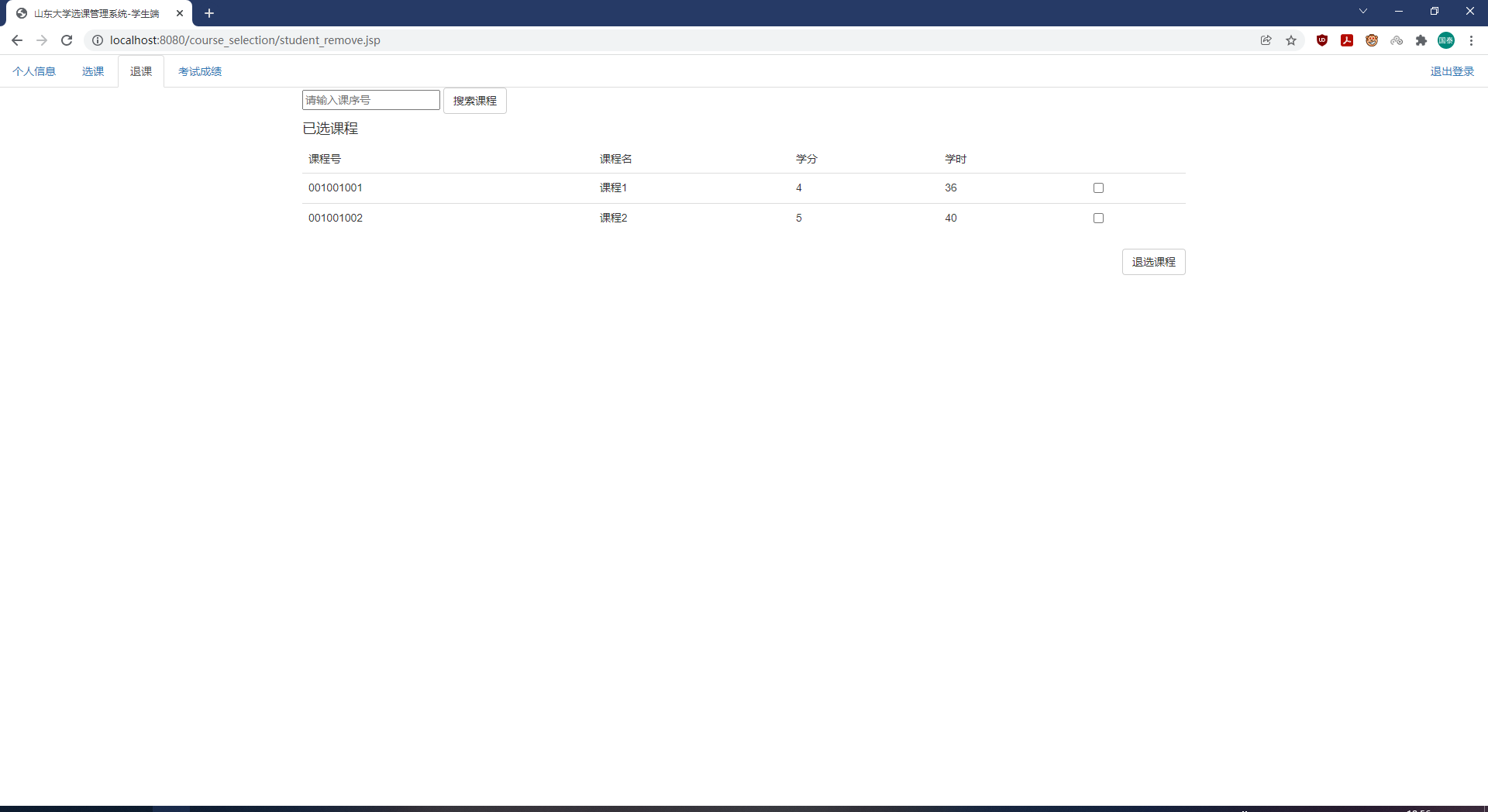




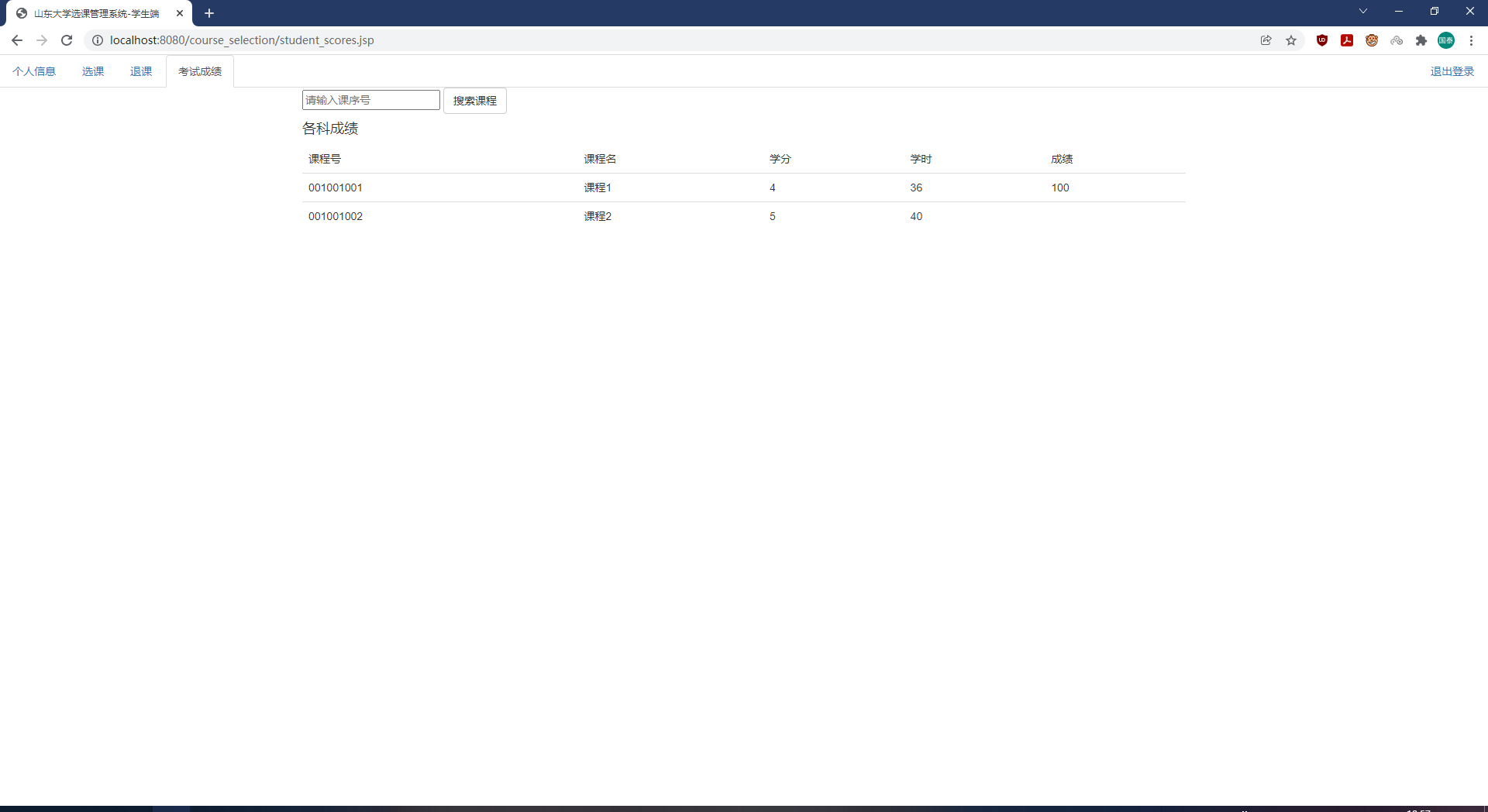
**7.2.3学生选课界面**



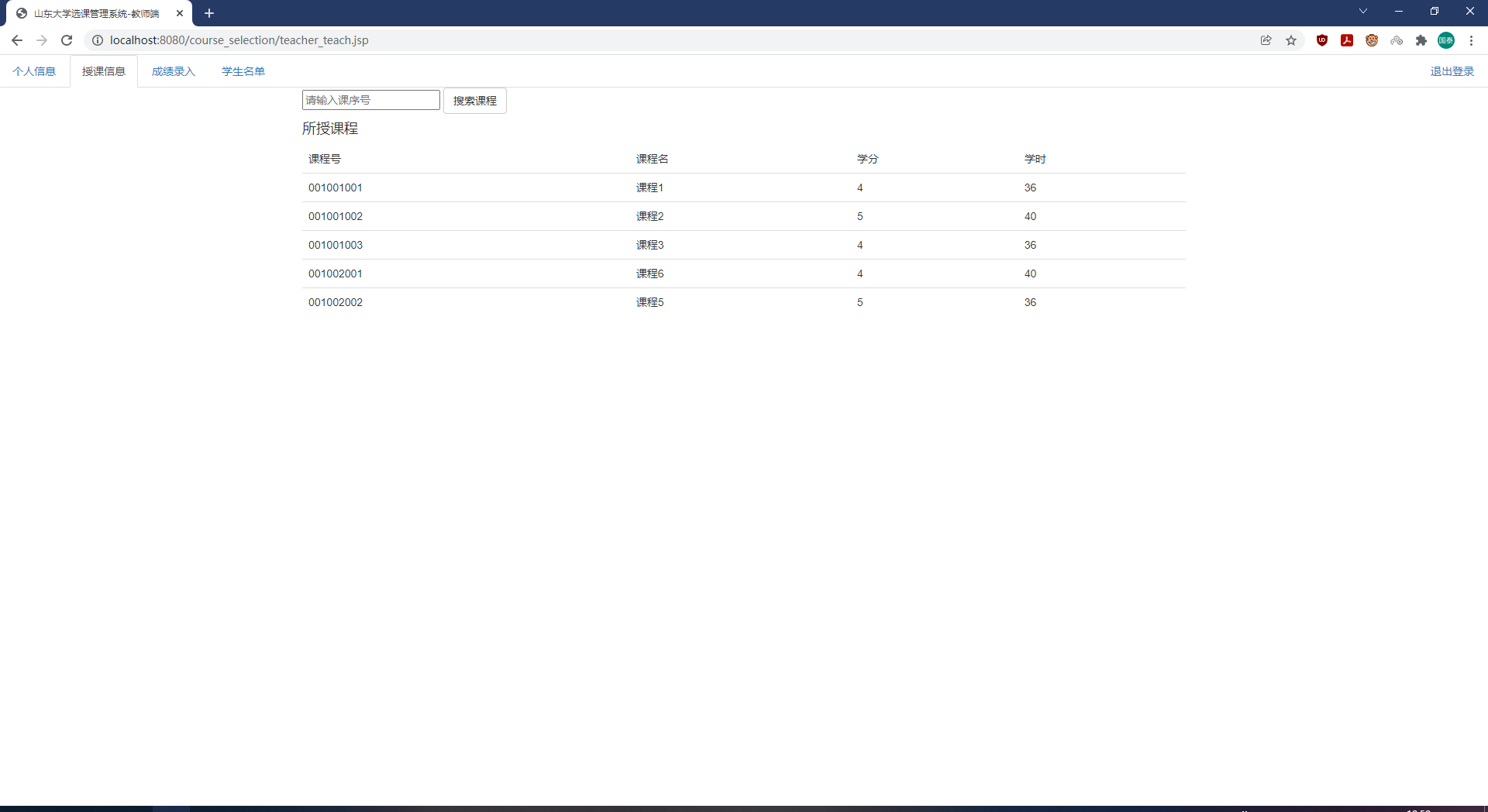
**7.2.4学生退课界面**



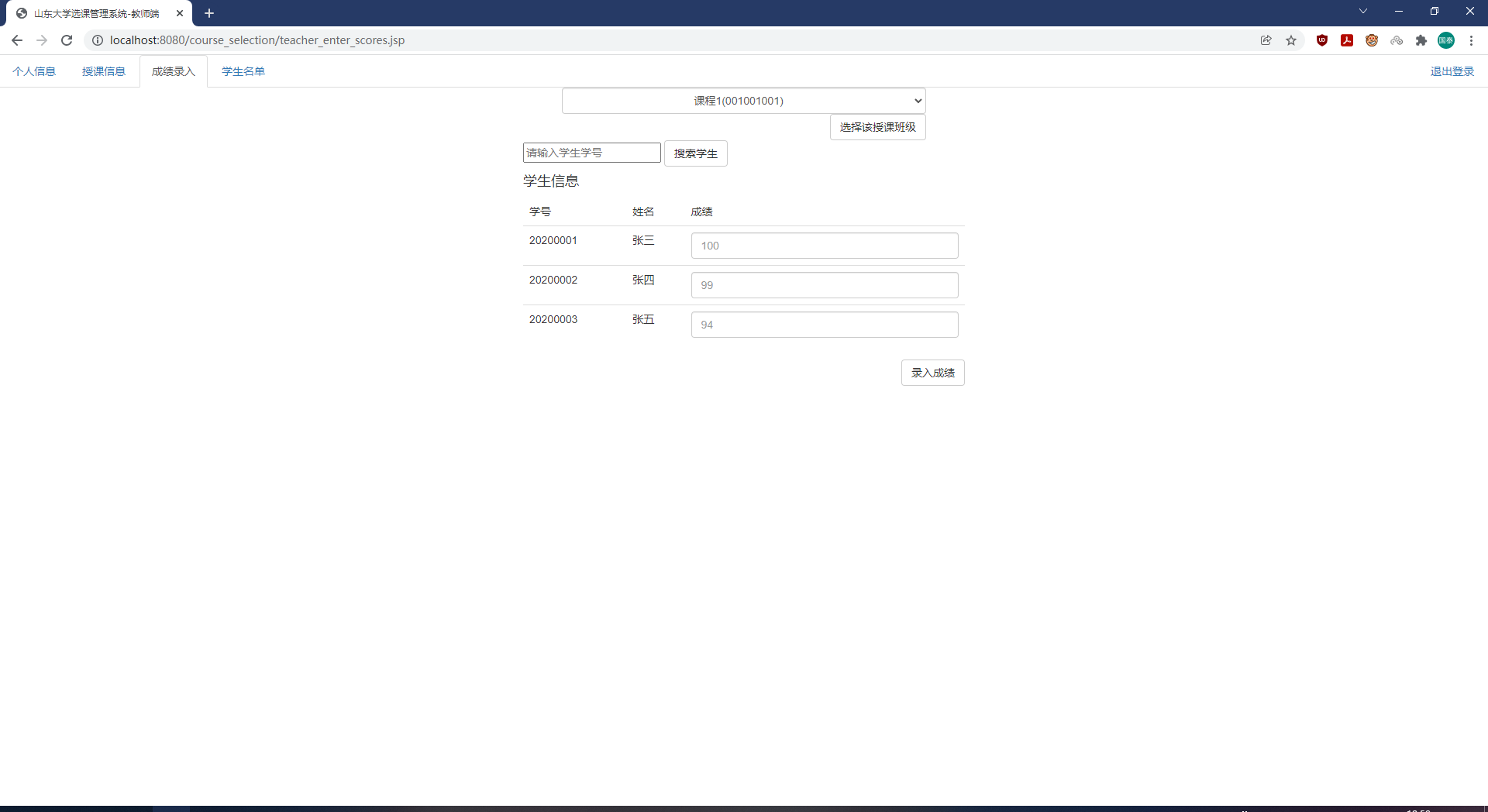
**7.2.5学生成绩查看界面**



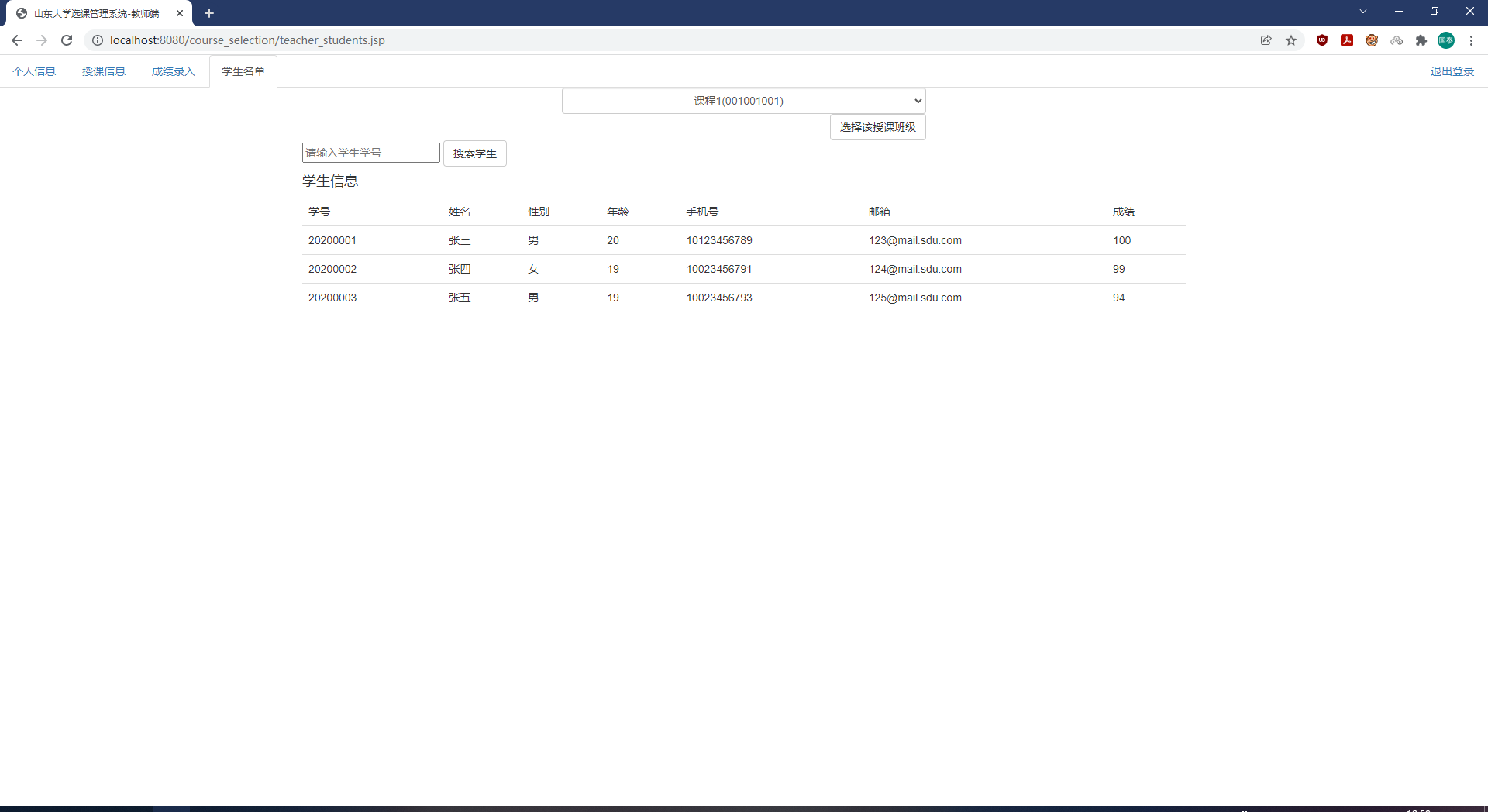
**7.2.6教师授课信息查看界面**



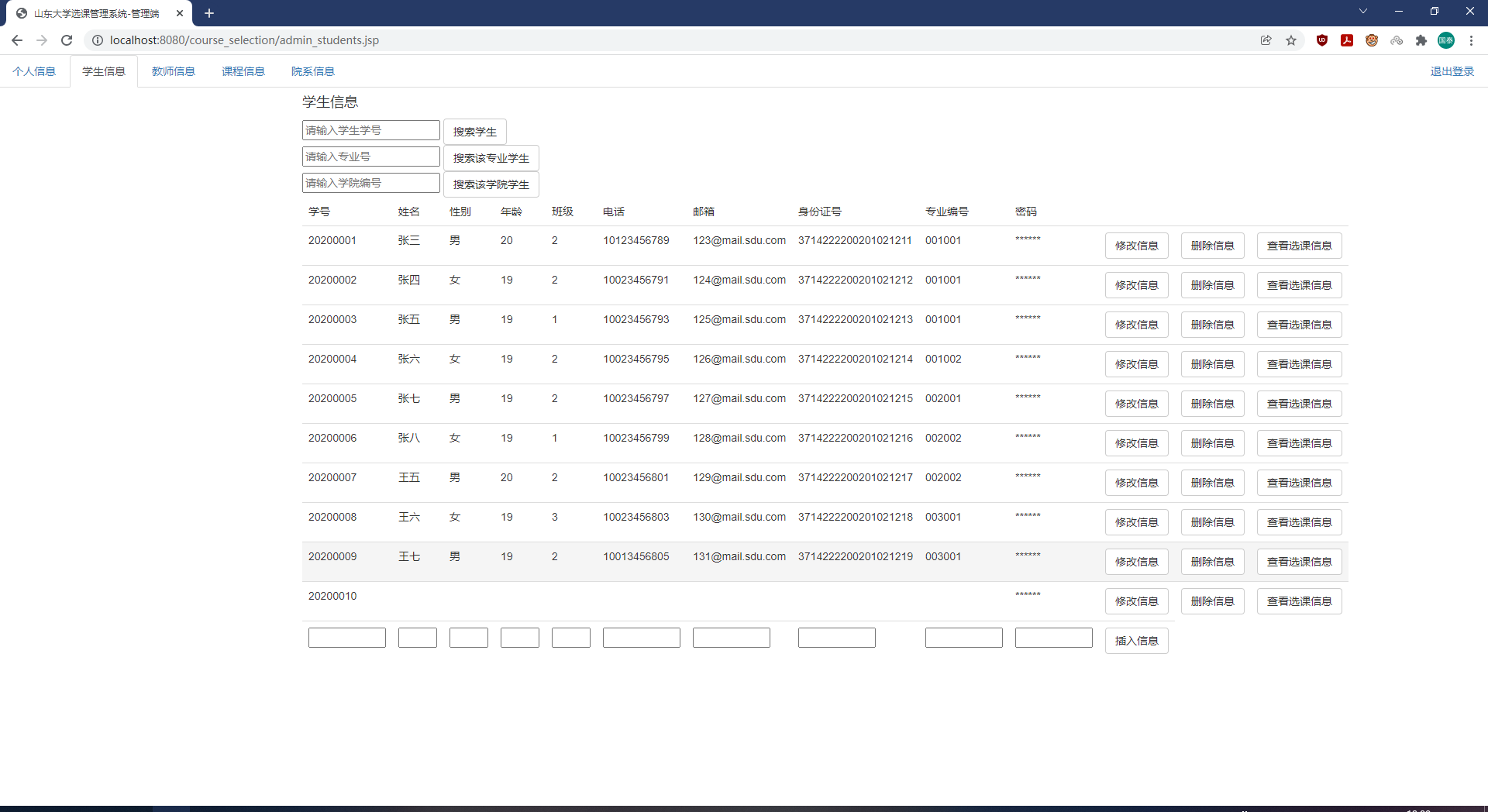
**7.2.7教师成绩录入界面**



**7.2.8教师查看学生名单界面**



**7.2.9管理员学生信息查看、插入和修改**

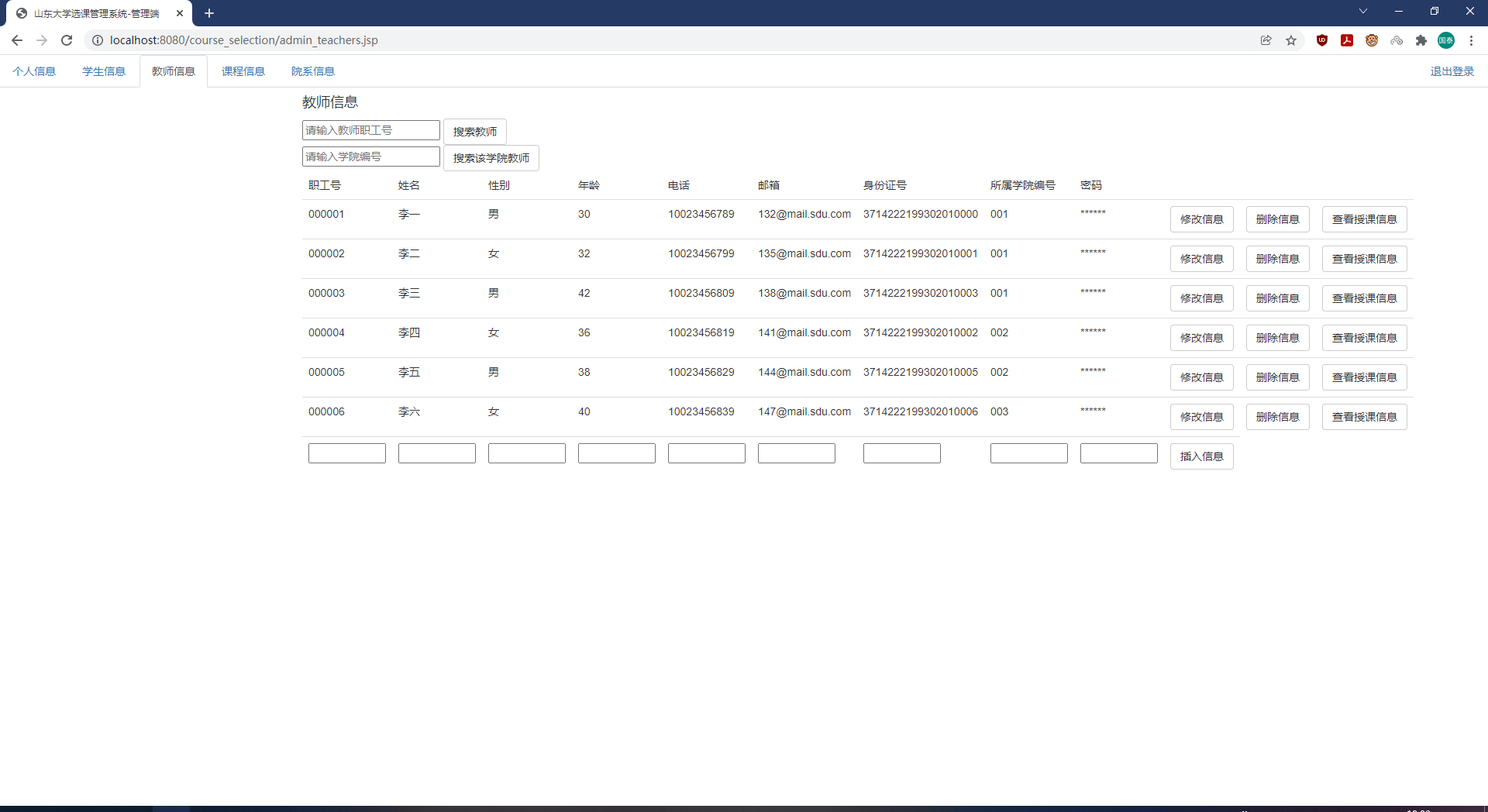




**7.2.10管理员学生选课信息查看**



**7.2.11管理员教师信息查看、插入和修改**

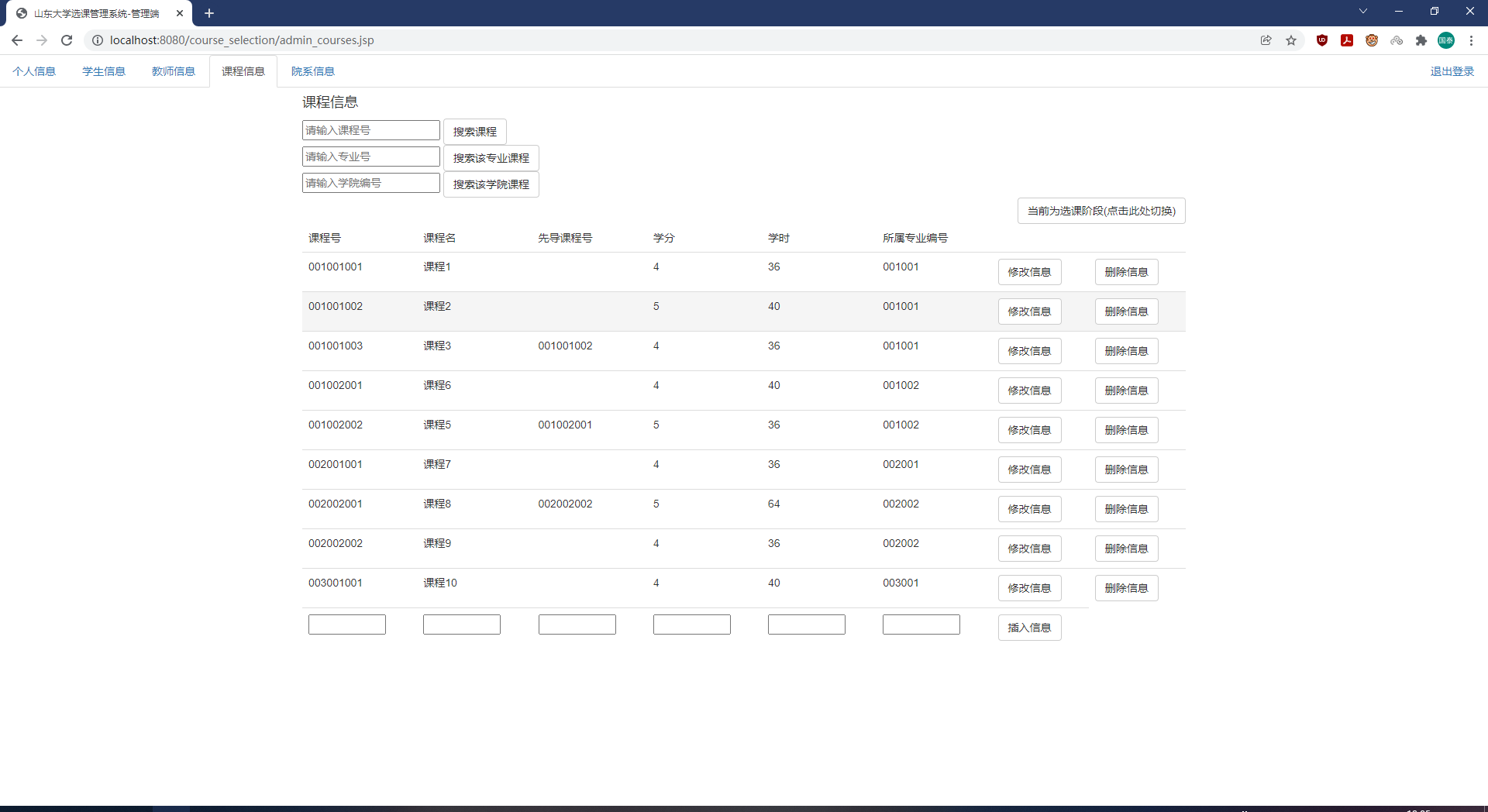


**7.2.12管理员教师选课信息查看、插入和修改**



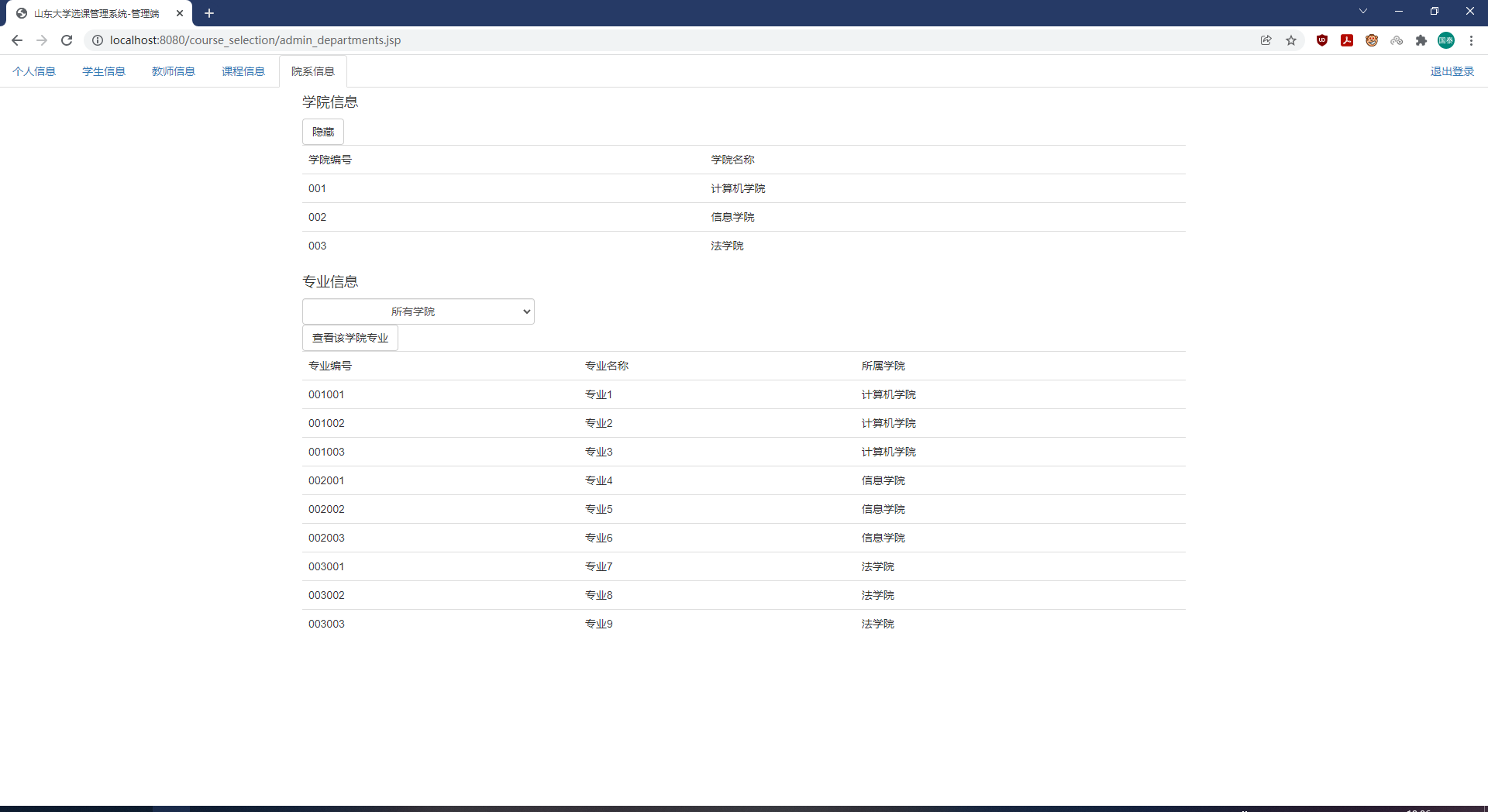
**7.2.13管理员课程信息查看、插入和修改**

可以通过右上角按钮切换选课阶段进行与否



**7.2.14管理员学院和专业信息查看界面**

其中学院信息可通过按钮显示或隐藏，专业信息可通过下拉菜单分学院查看

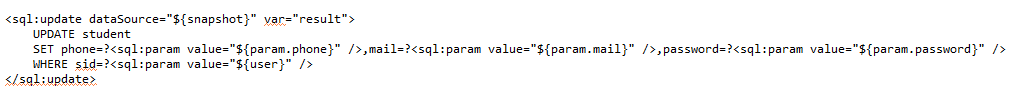


# 7.3事务设计

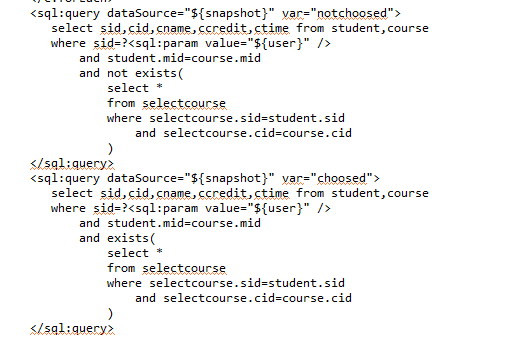
该系统中，所有关于查询的事务基本上都可以由Select语句实现，而更新的事务则用Update语句，删除的事务用Delete语句实现。

**7.3.1学生信息查看修改**

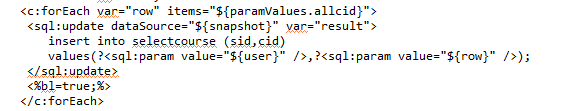


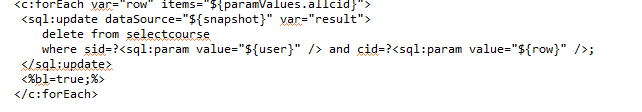


**7.3.2学生未选课程和已选课程选择**

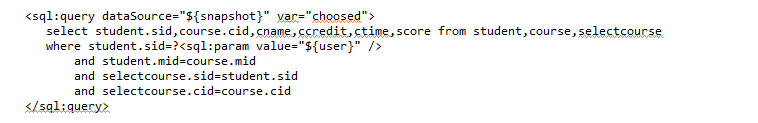


**7.3.3学生选课信息插入、删除**

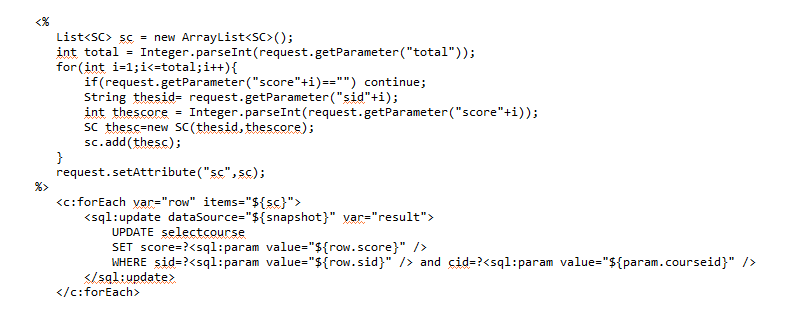




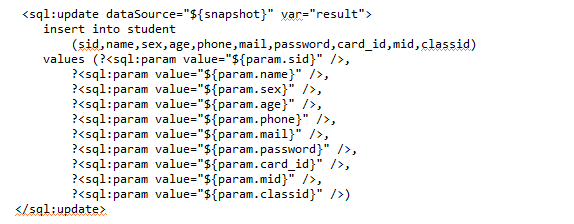
**7.3.4学生成绩查看**

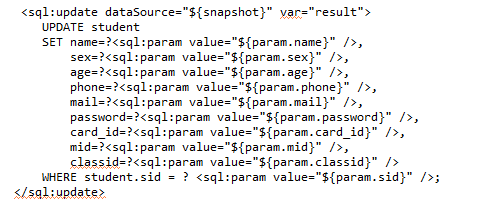


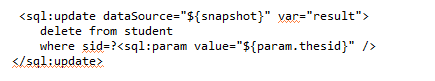
**7.3.5教师进行学生成绩录入**



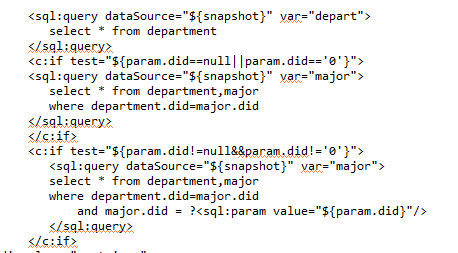
**7.3.6管理员端进行学生信息插入、修改、删除**







**7.3.7管理员端进行院系信息查看**



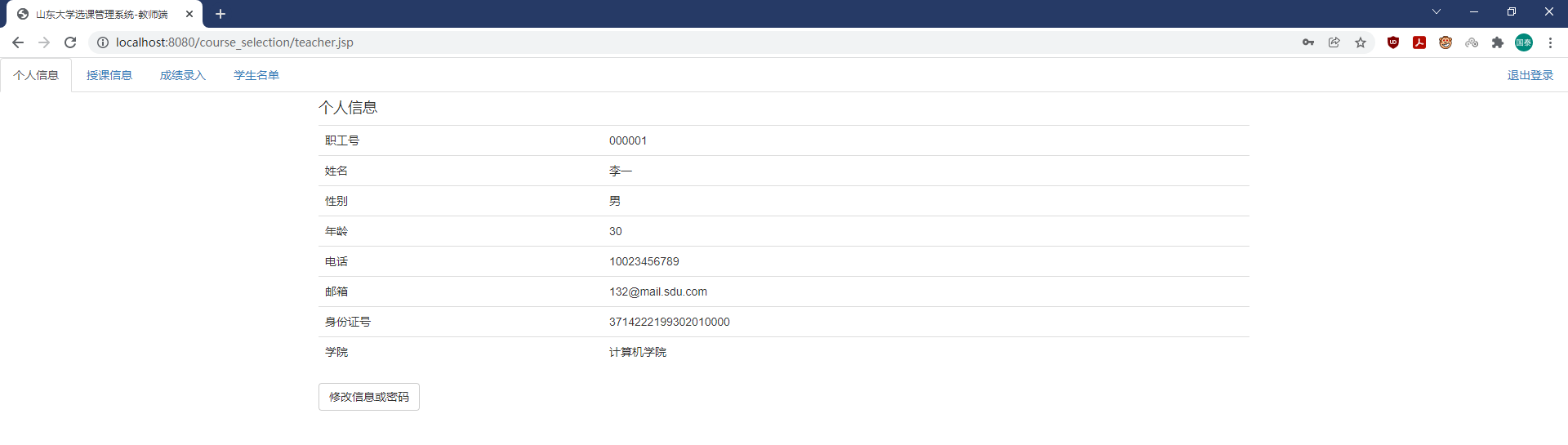
# 八、测试和运行

**8.1用户登录**

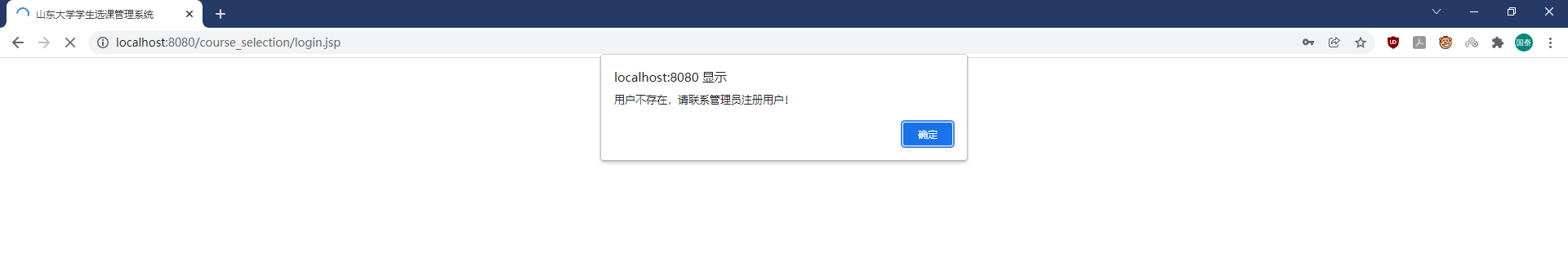
以教师为例，在下拉菜单中选择教师用户、输入职工号和密码，点击登录



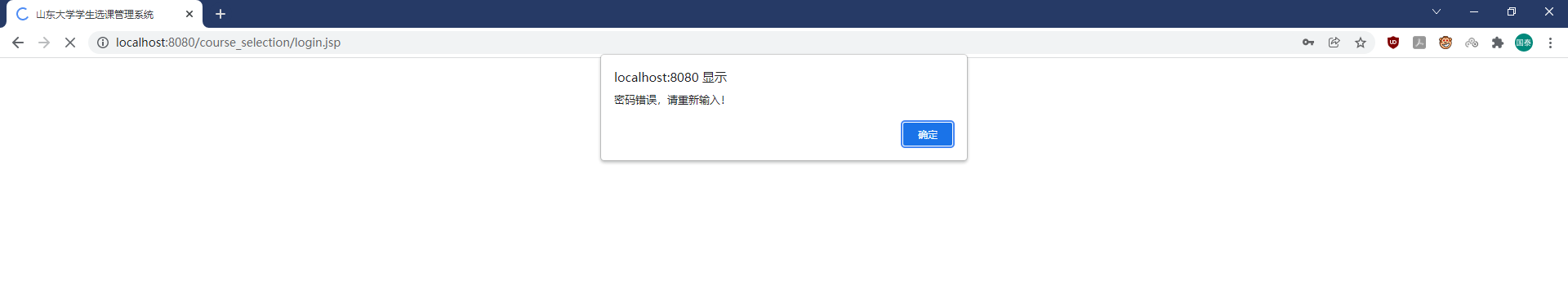
职工号和密码正确时登陆成功，进入个人信息主页面



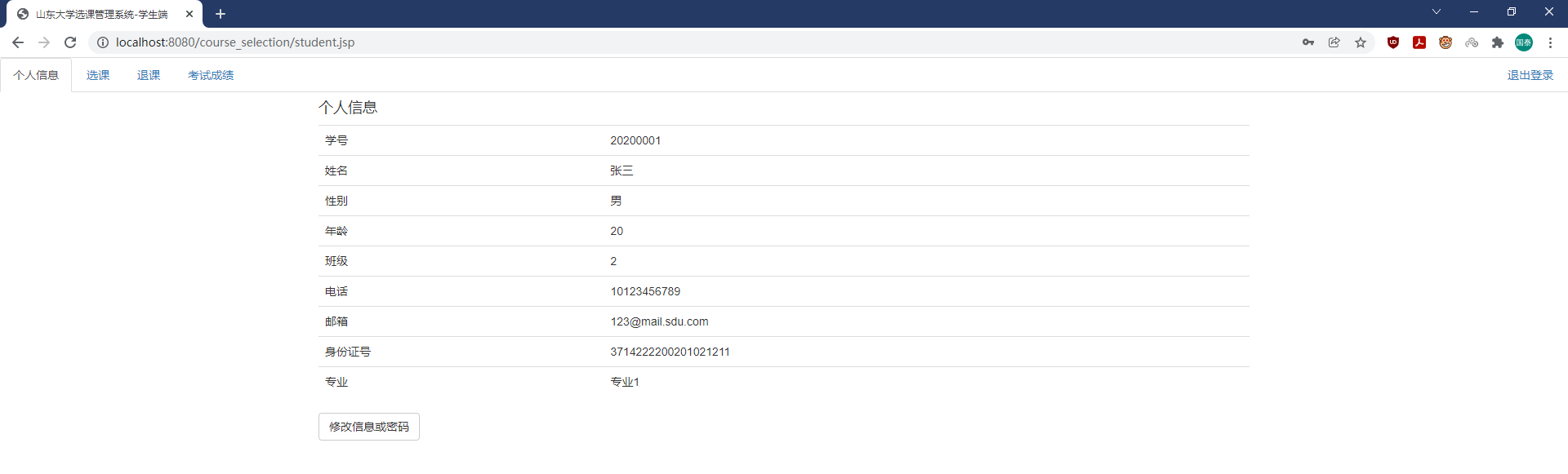
输入一个不存在的用户

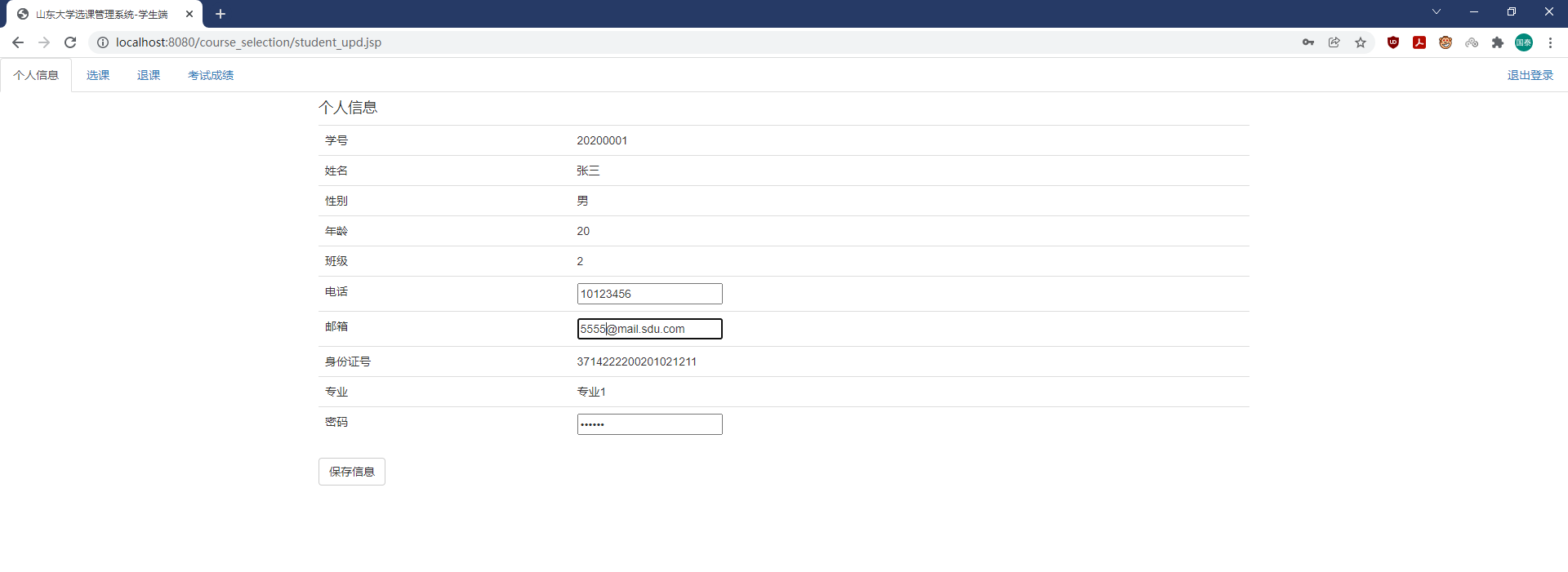


密码错误

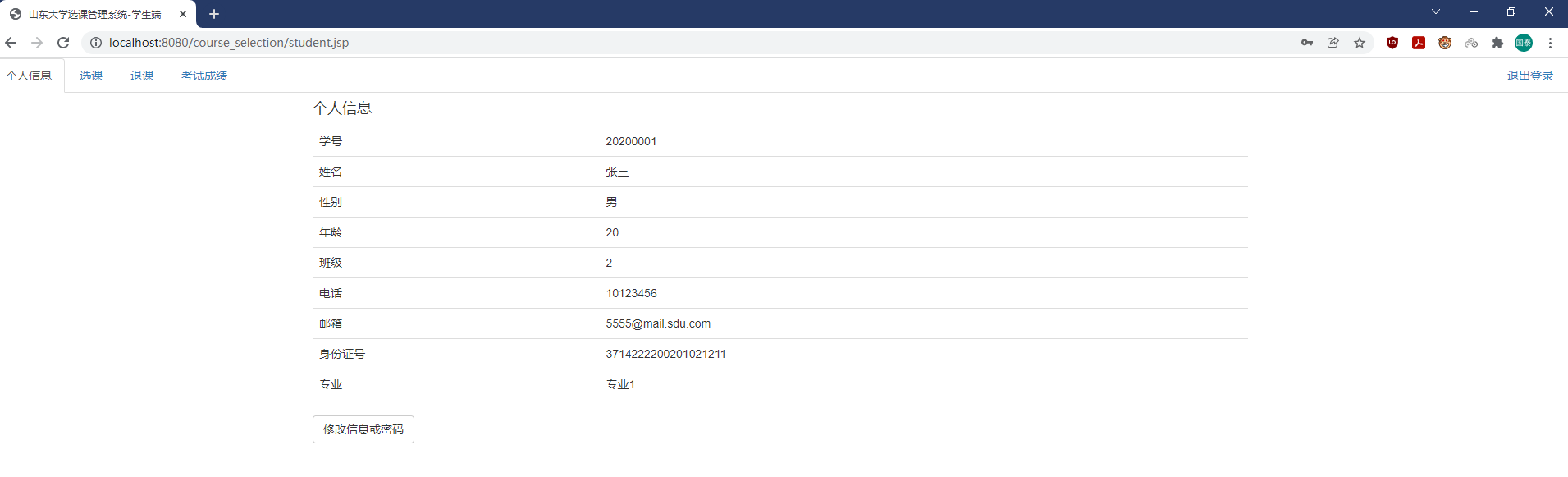


**8.2学生个人信息修改**









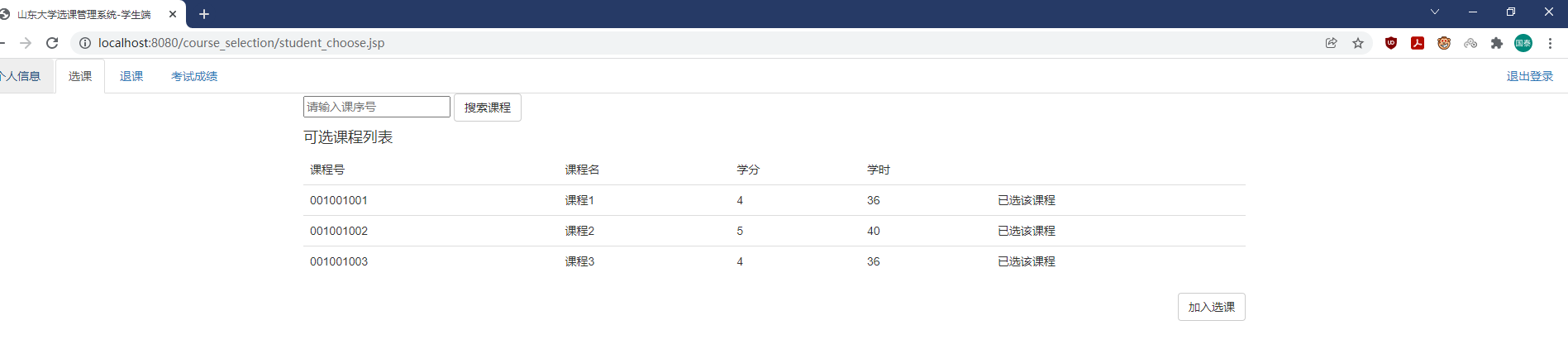
可以看到邮箱和电话修改成功

**8.3学生选课**

未选择课程可通过复选框选择后加入选课，流程如下，退课操作类似







**8.4教师端成绩录入**

通过下拉菜单选择某一授课班级，成绩输入框中显示曾经已录入的成绩，可进行修改可录入



将张三成绩改为80，张四改为87，张五保持不变，如下图录入成功

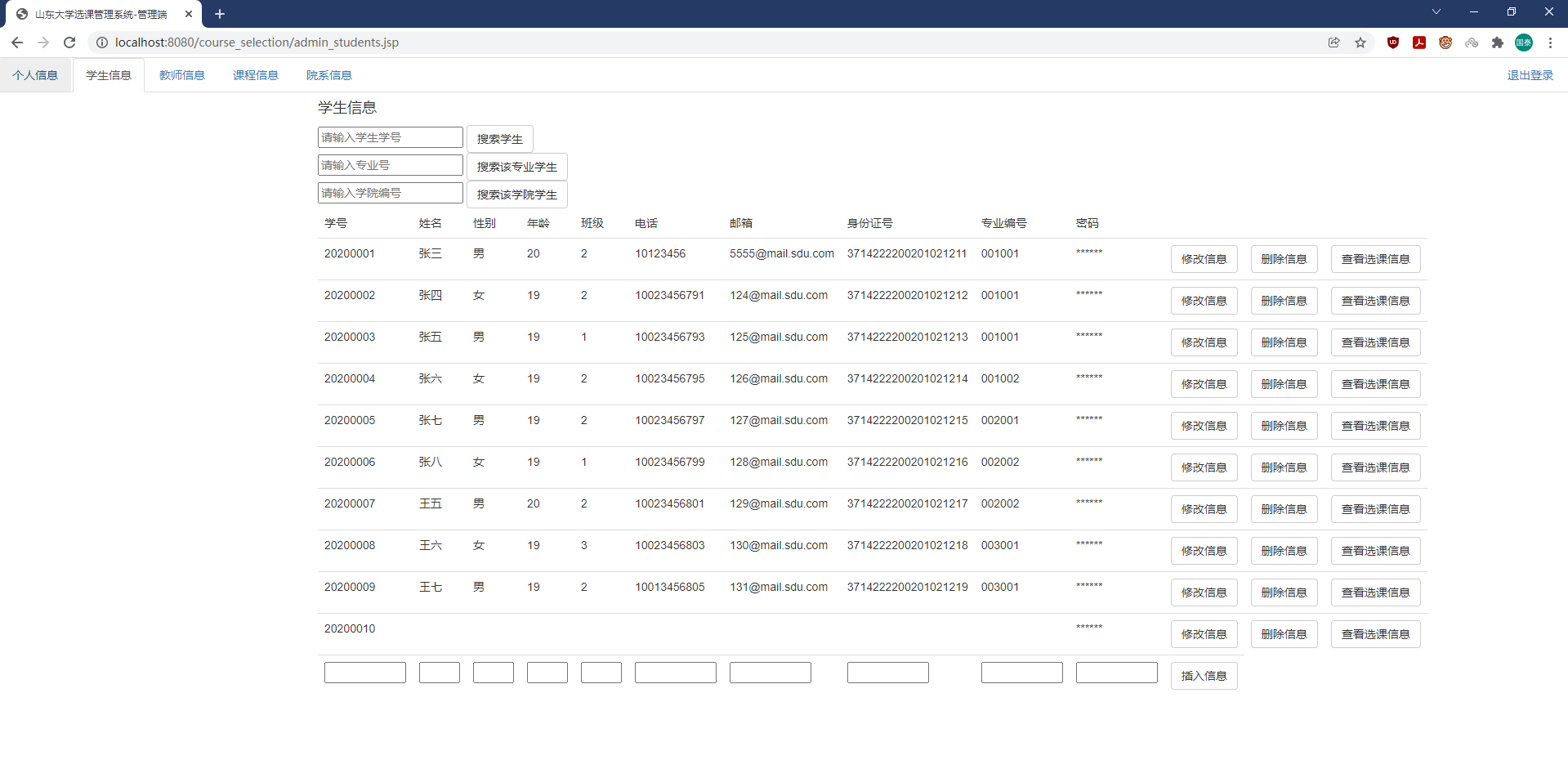


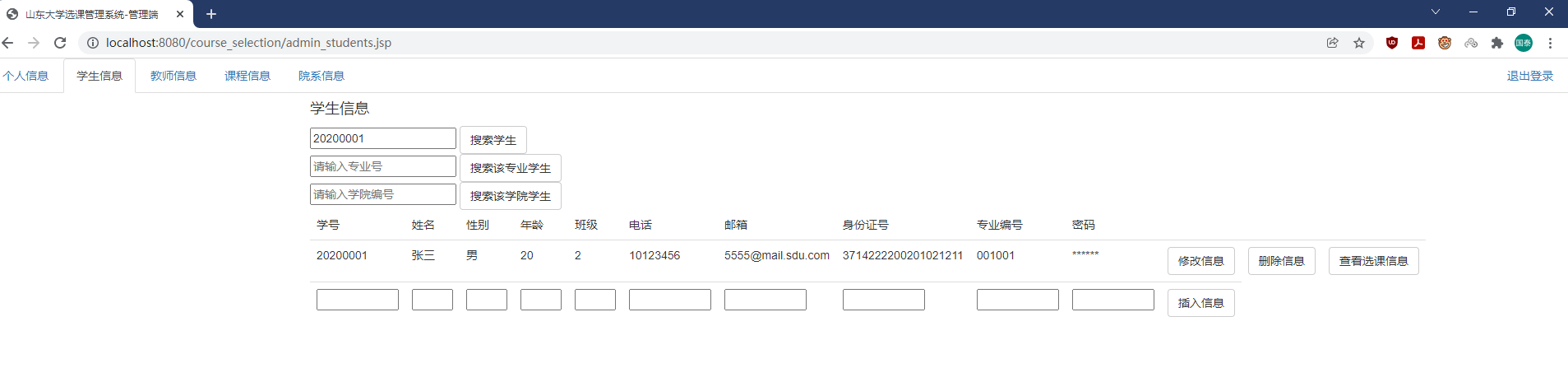


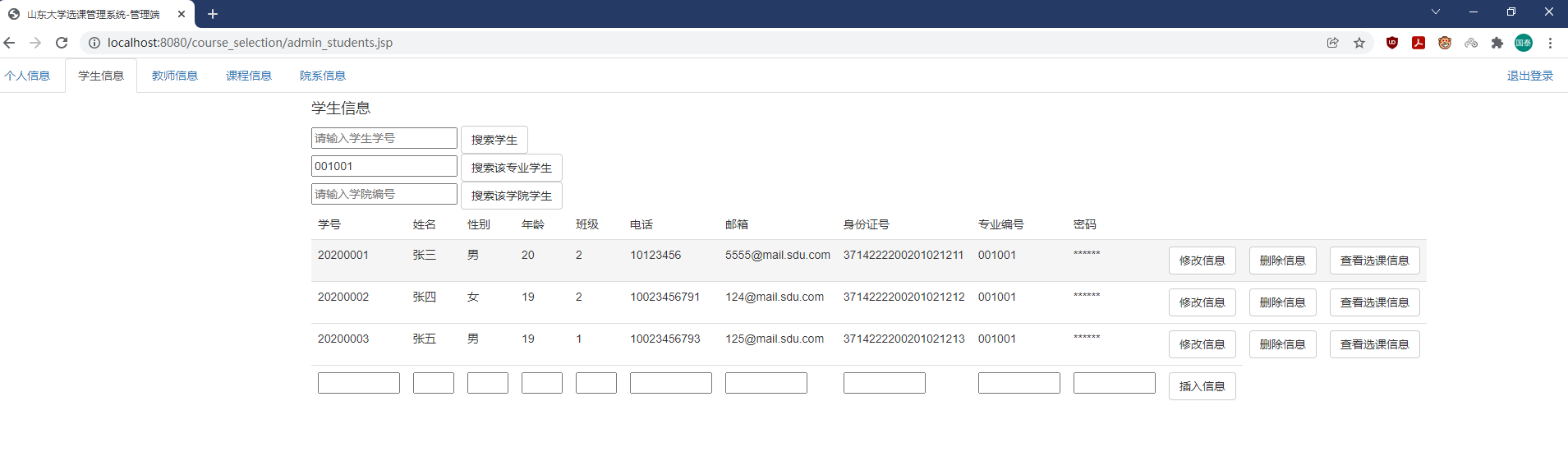


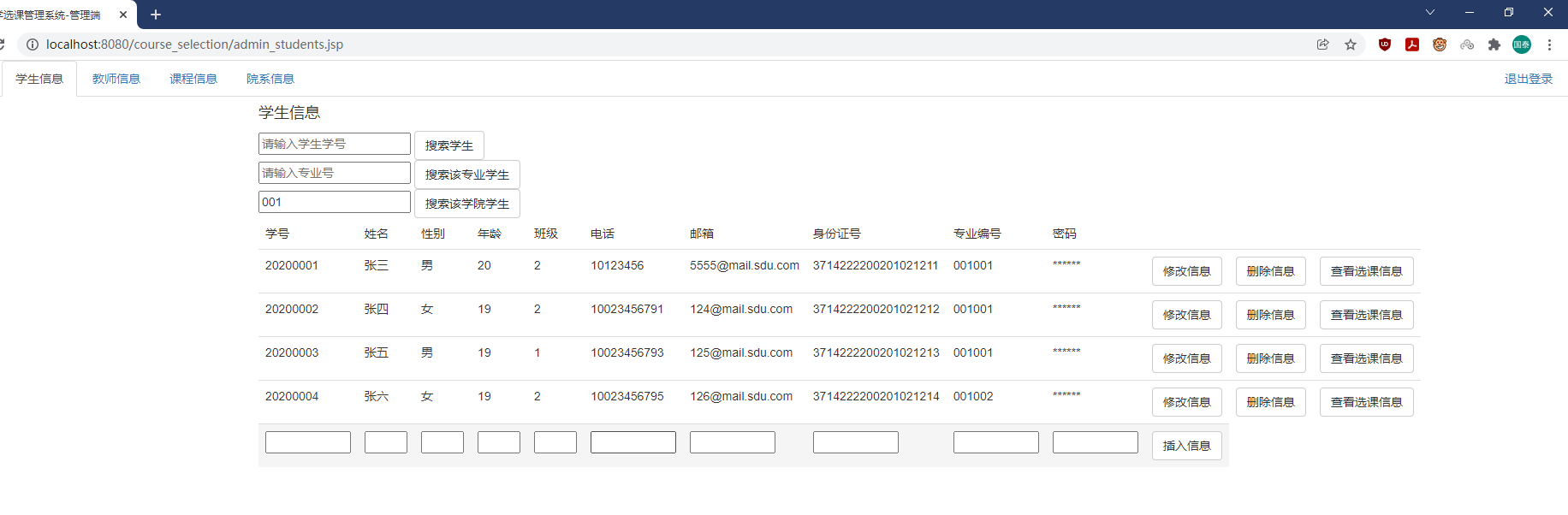
**8.5搜索功能**

本系统在多个界面实现了搜索功能，以下面管理端学生信息界面为例，分别进行了按照学号、专业、学院搜索学生的过程

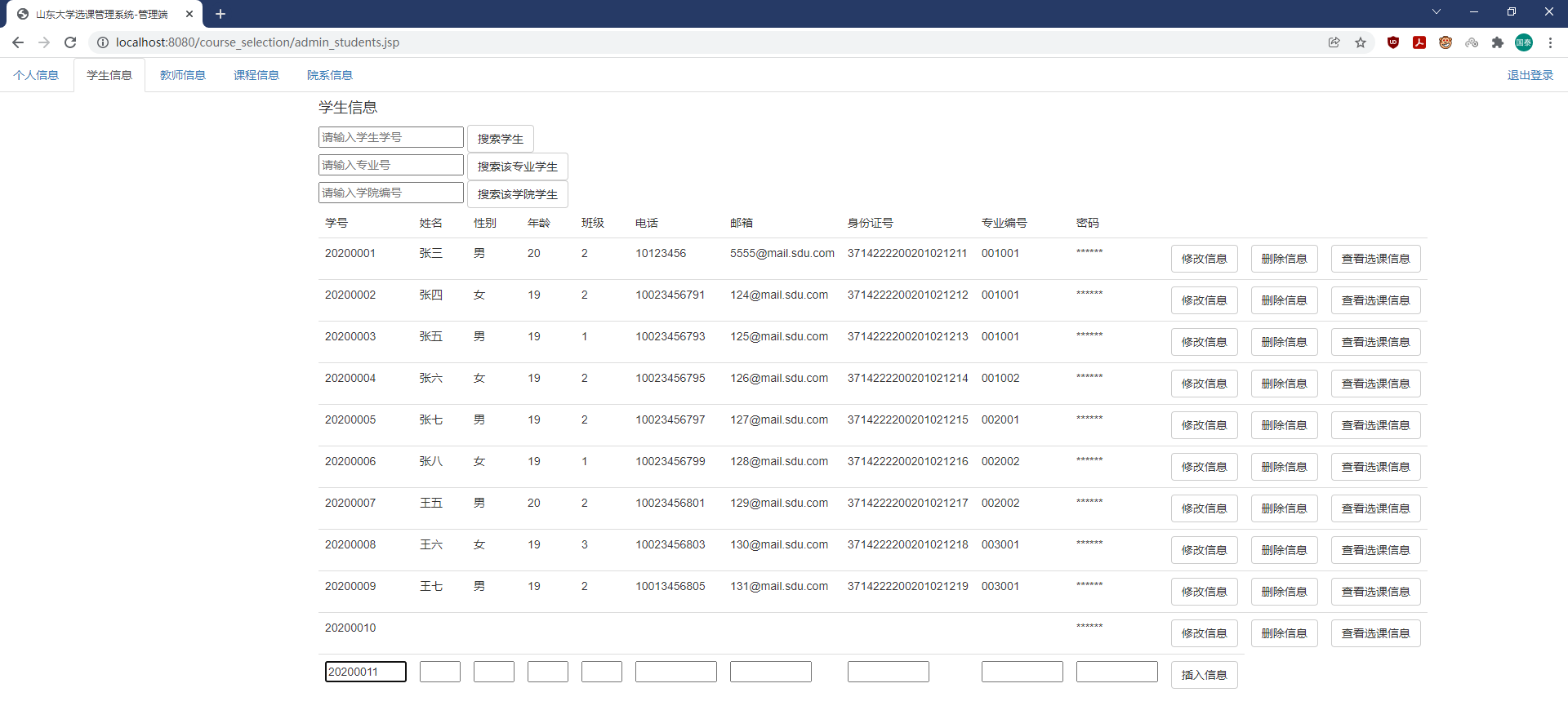




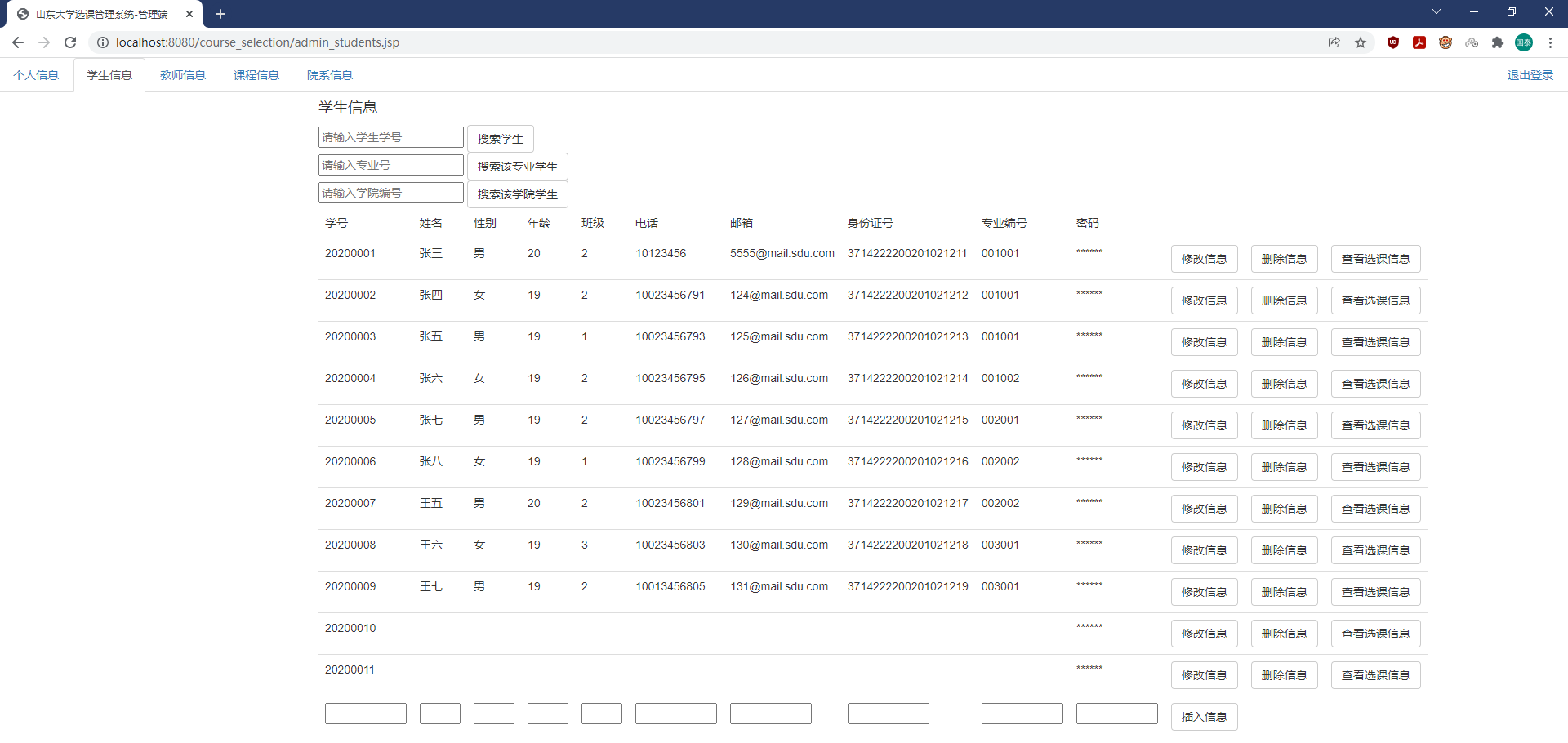




**8.6管理端学生信息修改、插入、删除**



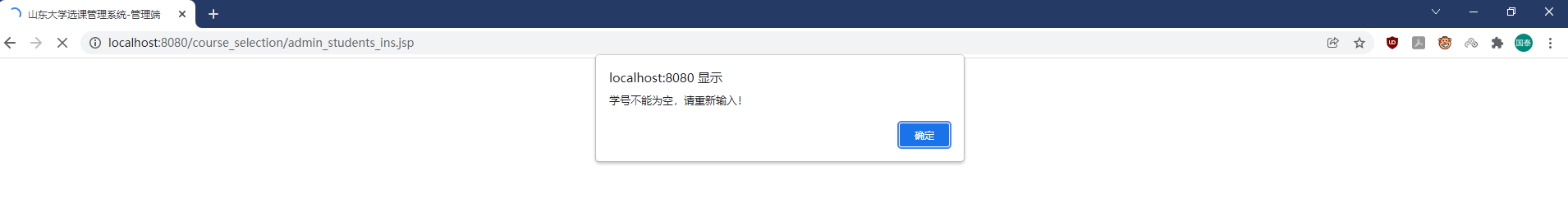
当插入一个学生，学号不能为空且唯一，其他项皆可缺省



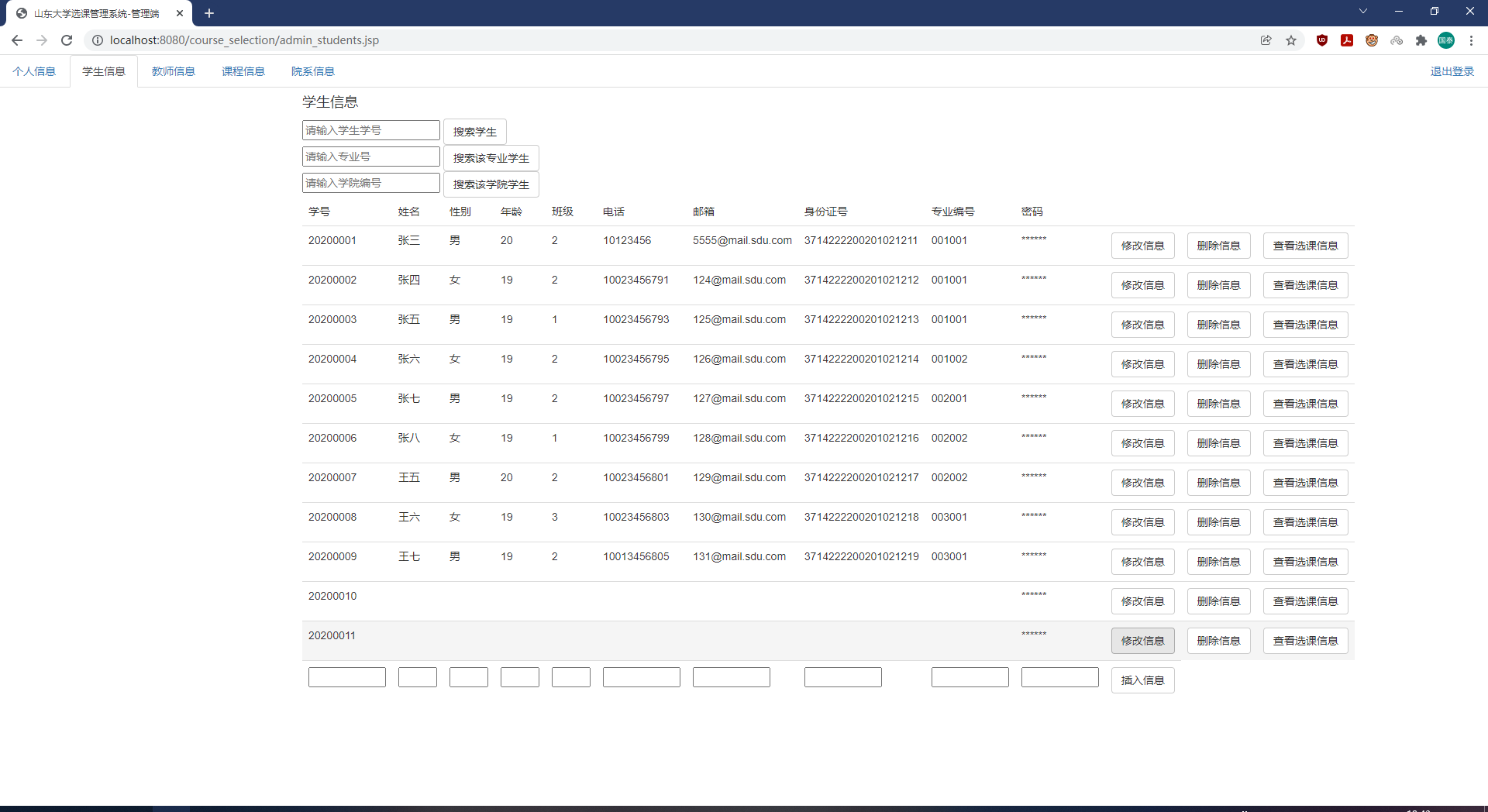
插入重复学号

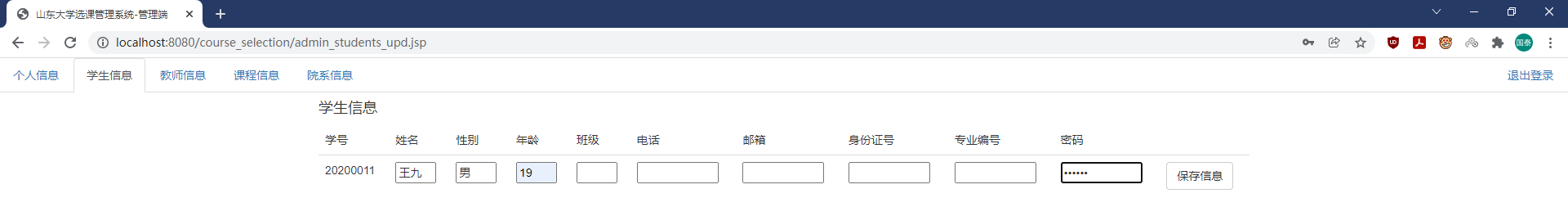


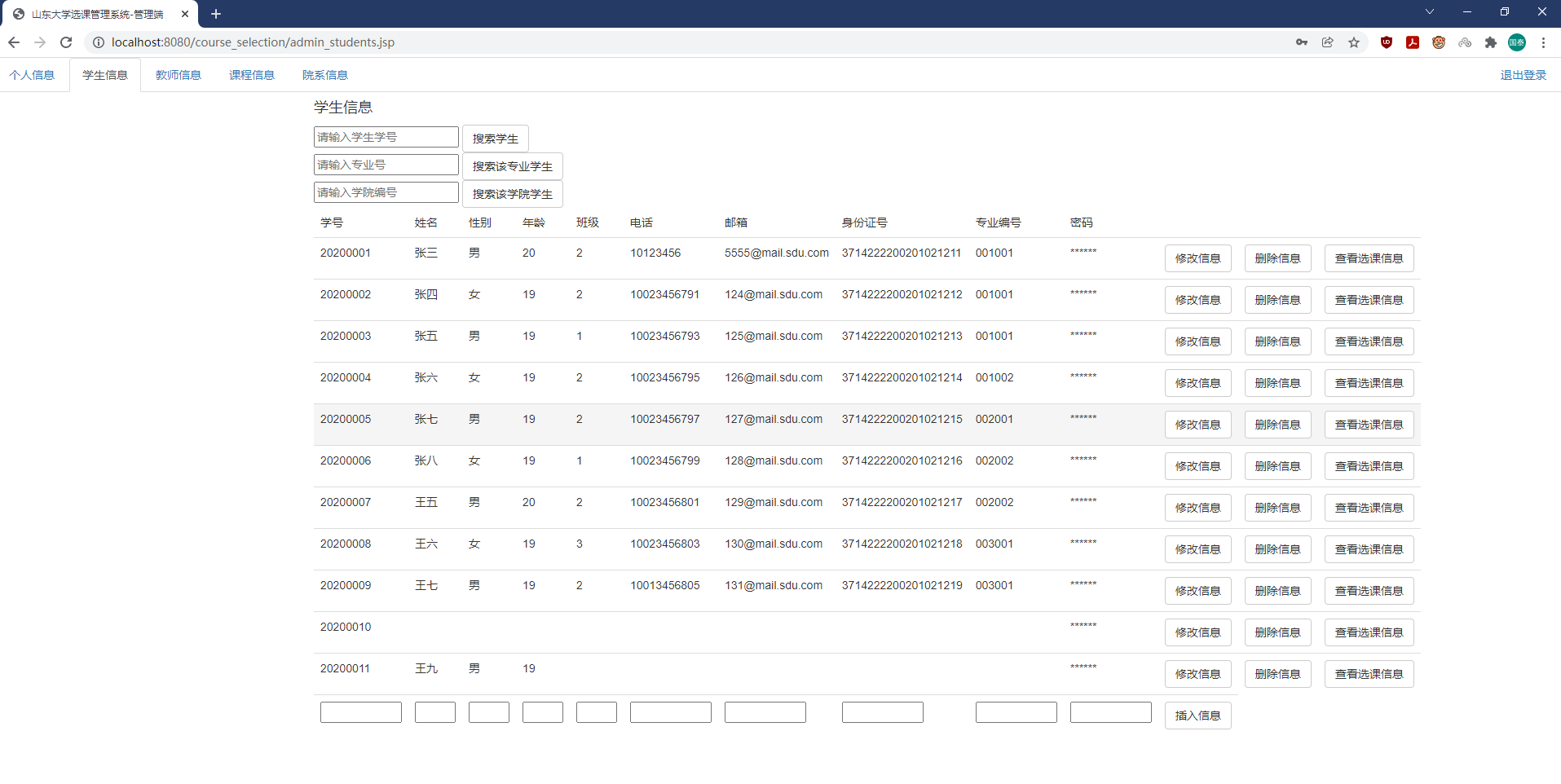
插入空学号



修改学号为20200011学生的信息

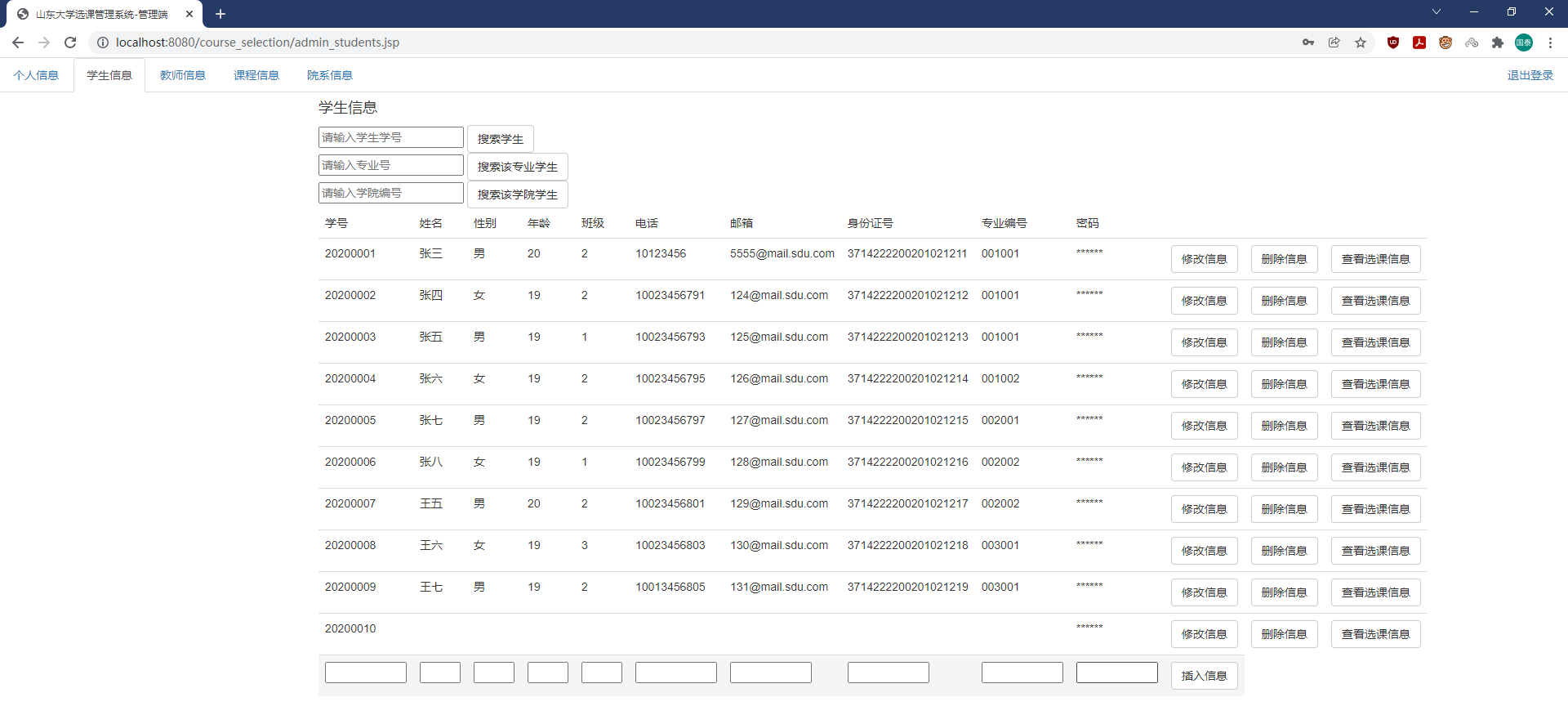






删除王九





**8.7管理端查看学生选课信息**



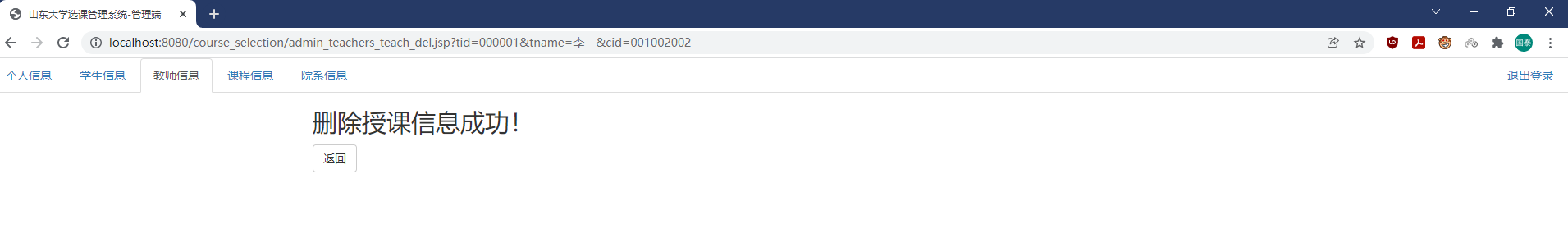


**8.8管理端进行教师授课信息的插入和删除**



删除课程5这一授课

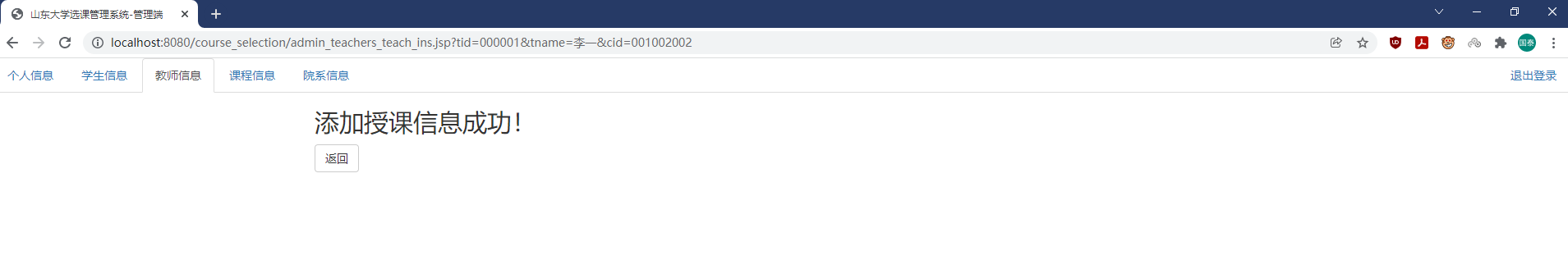






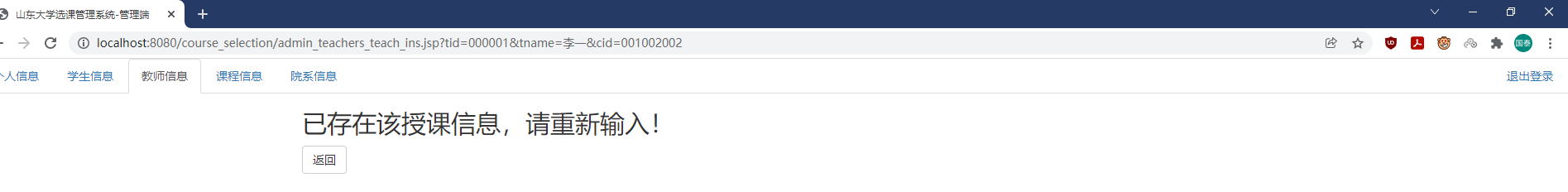
插入课程5这一授课



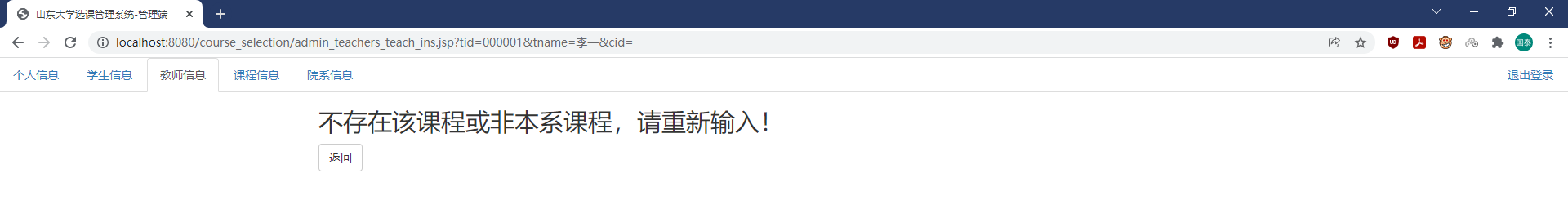




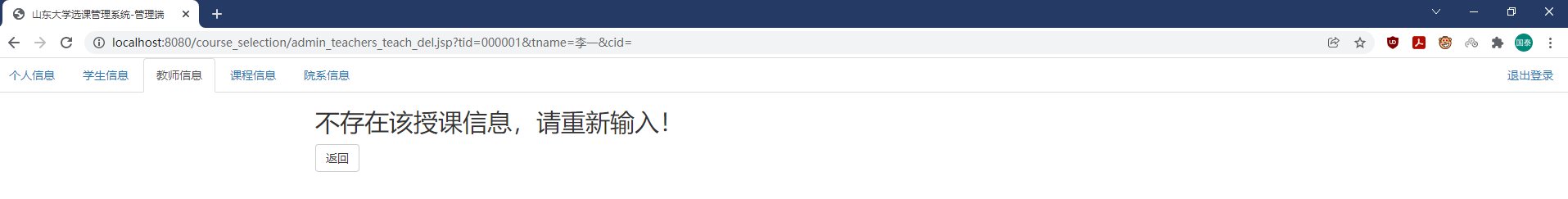
重复插入授课信息



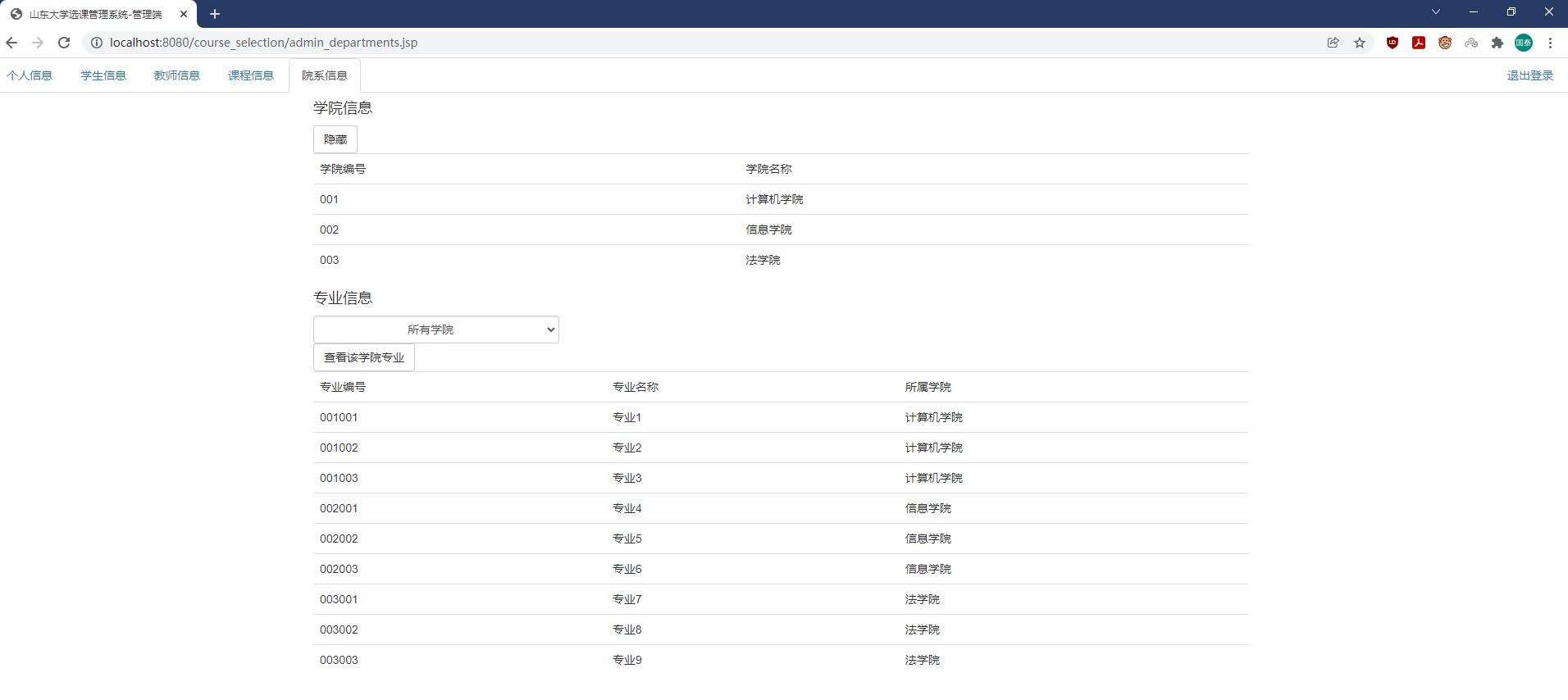
插入不存在的课程



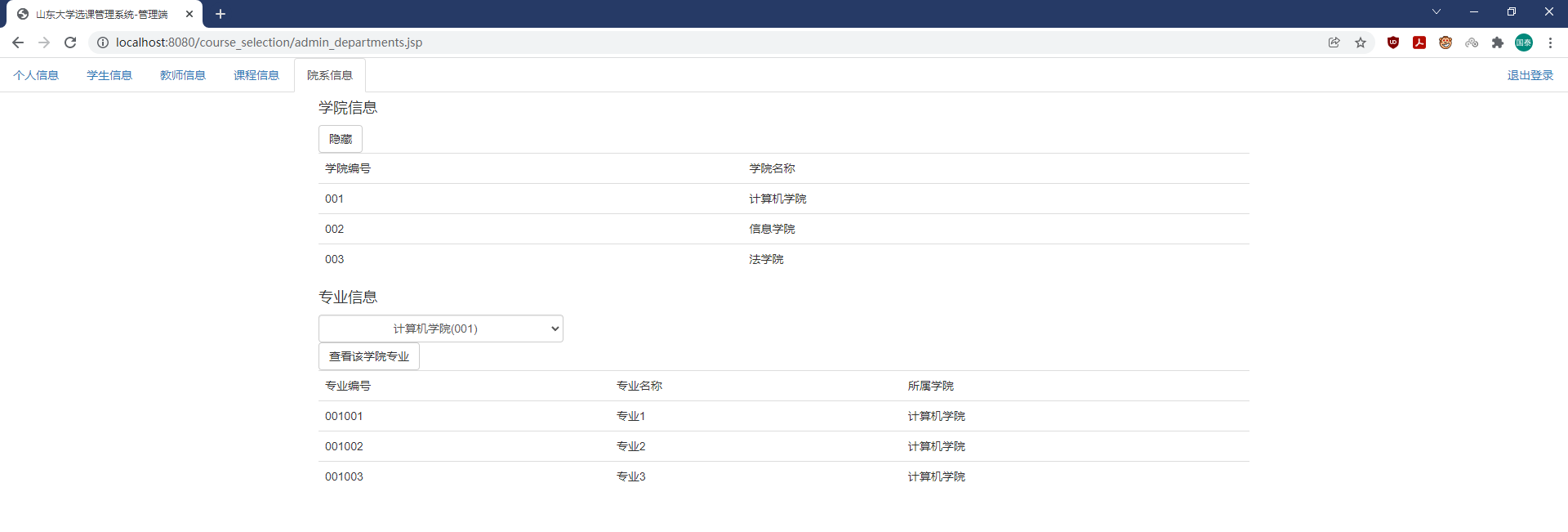
删除不存在的课程



**8.9管理端院系信息的查看**



查看计算机学院的专业



隐藏学院信息

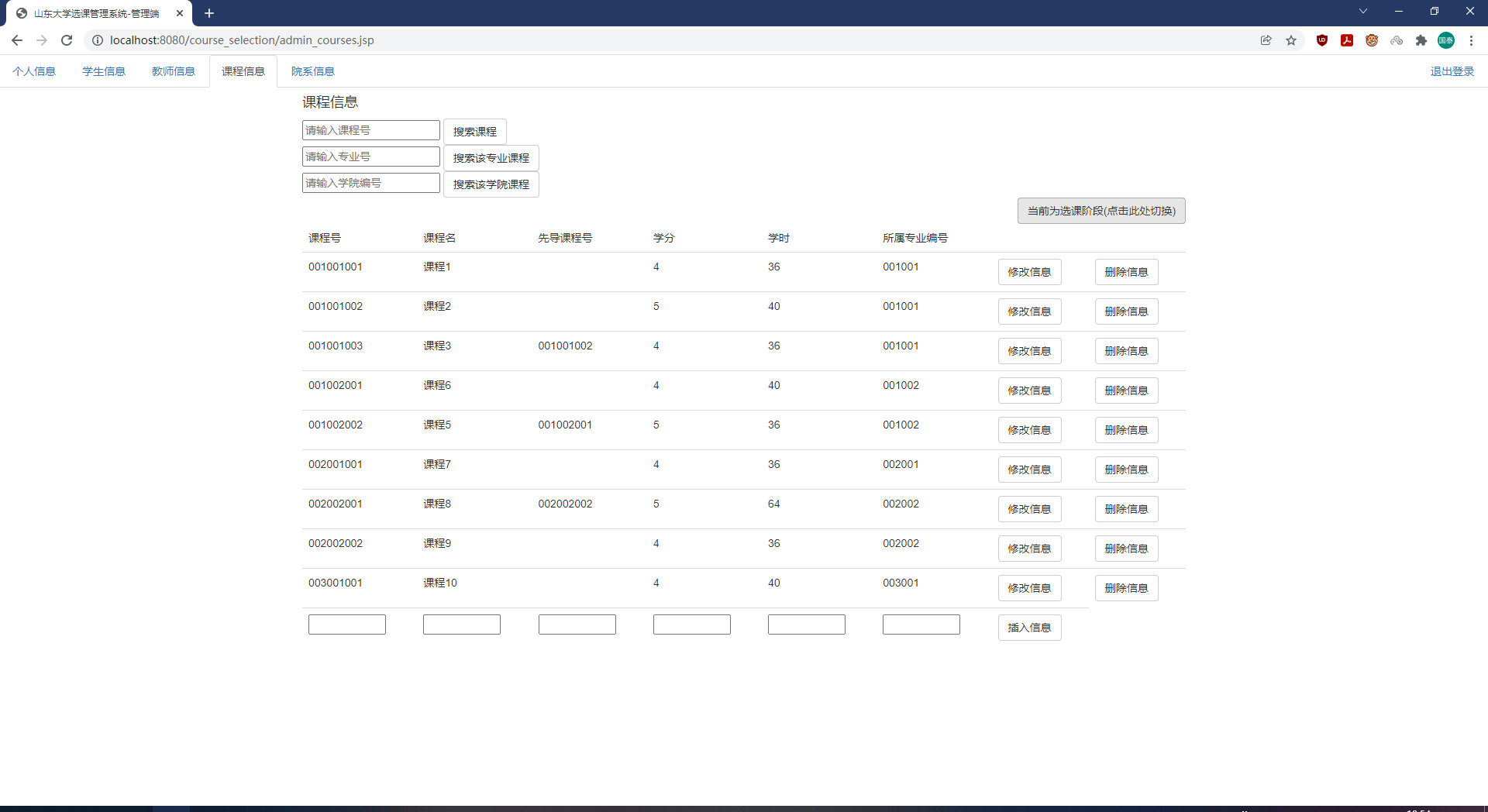


**8.10管理端选课阶段进行与否的切换**

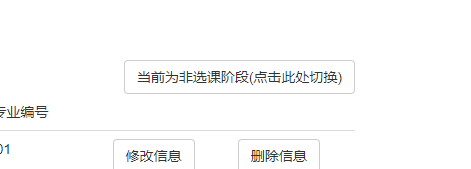
选课阶段在学生端可正常选课



切换选课阶段







在管理端按上面步骤切换为非选课阶段后再登录学生端，进入选课或退课界面显示如下



# 九、总结

**9.1系统的特色**

（1）用户权限分配合理。系统划分学生、教师、管理员三种用户，每种用户可以实现的功能各不相同，可进行的操作、查看和修改的信息有不同权限

（2）界面简洁美观。使用bootstrap3来进行UI设计，简洁美观。

（3）错误提示充分。在用户登录、学生选课、教师授课信息添加、学生教师课程的修改添加等操作中考虑到了可能出现的各种不合法操作，并给出提示信息。

（4）功能齐全。除了实现了各种信息的查看、修改、删除这些基本功能外，还支持通过学号、职工号、课程号、院系进行搜索的操作，支持通过下拉选项进行选择的操作。

（5）操作容错率较高。比如在教师端进行成绩录入时，可以一次录入多个学生的成绩，可以不必将某门课所有学生的成绩都输入后再提交，空置的输入框默认不会对成绩产生影响，没有成绩的同学和已有成绩的同学可以同时进行录入或修改。

**9.2不足和改进**

本系统中的各种搜索功能不支持模糊搜索，搜索的学号等内容必须完全相符才能匹配，可以考虑加入正则表达式来实现模糊搜索。

**9.3经验体会**

本次“学生选课管理系统”的设计让我学习了有关HTML，JSP，MYSQL等各种技术，为保证高移植性，系统采用B/S架构，依据各种需求，我从网络论坛和书籍上自学了种种与web开发相关的知识技能，可谓受益匪浅。

同时，整个系统的设计和实现是一个克服种种困难，一点一滴细化的过程。数据库课设包含了不仅仅数据库这一门课的知识，需要各种技能的掌握应用，这一段路定能为我未来各门课程打下基础。

# 附. 参考文献

菜鸟教程<https://www.runoob.com/>

JSP连接数据库<https://www.runoob.com/jsp/jsp-database-access.html>

Btootstrap3 <https://www.runoob.com/bootstrap/bootstrap-tutorial.html>

HTML <https://www.runoob.com/html/html-tutorial.html>