**教务管理系统说明文档**



**目录**

1. 引言……………………………………3
2. Django框架简介………………………4
3. 教务系统逻辑结构 ……………………5

四．教务系统网站详解 ……………………6

五．教务系统后台程序……………………19

六．教务系统数据库连接…………………23

七．高并发问题……………………………24

1. **引言**

本教务管理系统基于python语言，运用了由python开发的免费网站开发框架Django，可以用于快速搭建高性能，优雅的网站。Django 中提供了开发网站经常用到的模块，通过减少重复的代码，Django 使你能够专注于 web 应用上有趣的关键性的东西。

Django框架本身提供了后台程序，几行简单的代码就能让网站拥有一个强大的后台，轻松管理网站内容。拥有自带的数据库，用python的类继承，几行代码就可以拥有一个丰富，动态的数据库操作接口（API），非常适合于教务管理系统。同时程序设计师也可外接数据库，实现更有针对性的功能。

在编写语言上，Django拥有强大，易扩展的模板系统，设计简易，代码，样式分开设计，更容易管理。在阅读语言上，完全支持多语言应用，允许你定义翻译的字符，轻松翻译成不同国家的语言。

Django 中提供了开发网站经常用到的模块，常见的代码都为你写好了，通过减少重复的代码，Django 使你能够专注于 web 应用上有 趣的关键性的东西。为了达到这个目标，Django 提供了通用Web开发模式的高度抽象，提供了频繁进行的编程作业的快速解决方法，以及为“如何解决问题”提供了清晰明了的约定。

Django的理念是DRY(Don't Repeat Yourself)来鼓励快速开发。

**二．Django框架简介**

1.附件中mysystem文件夹包含了Django所有文件及代码。本教务系统基于Django 1.10.x版本。

2.主要文件介绍

1. Urls.py

网址入口，关联到对应的views.py中的一个函数（或者generic类），访问网址就对应一个函数。

1. Views.py

处理用户发出的请求，从urls.py中对应过来, 通过渲染templates中的网页可以将显示内容，比如登陆后的用户名，用户请求的数据，输出到网页。

1. Models.py

与数据库操作相关，存入或读取数据时在其中添加代码。

1. Templates文件夹

views.py 中的函数渲染templates中的Html模板，得到动态内容的网页，当然可以用缓存来提高速度。

1. Admin.py

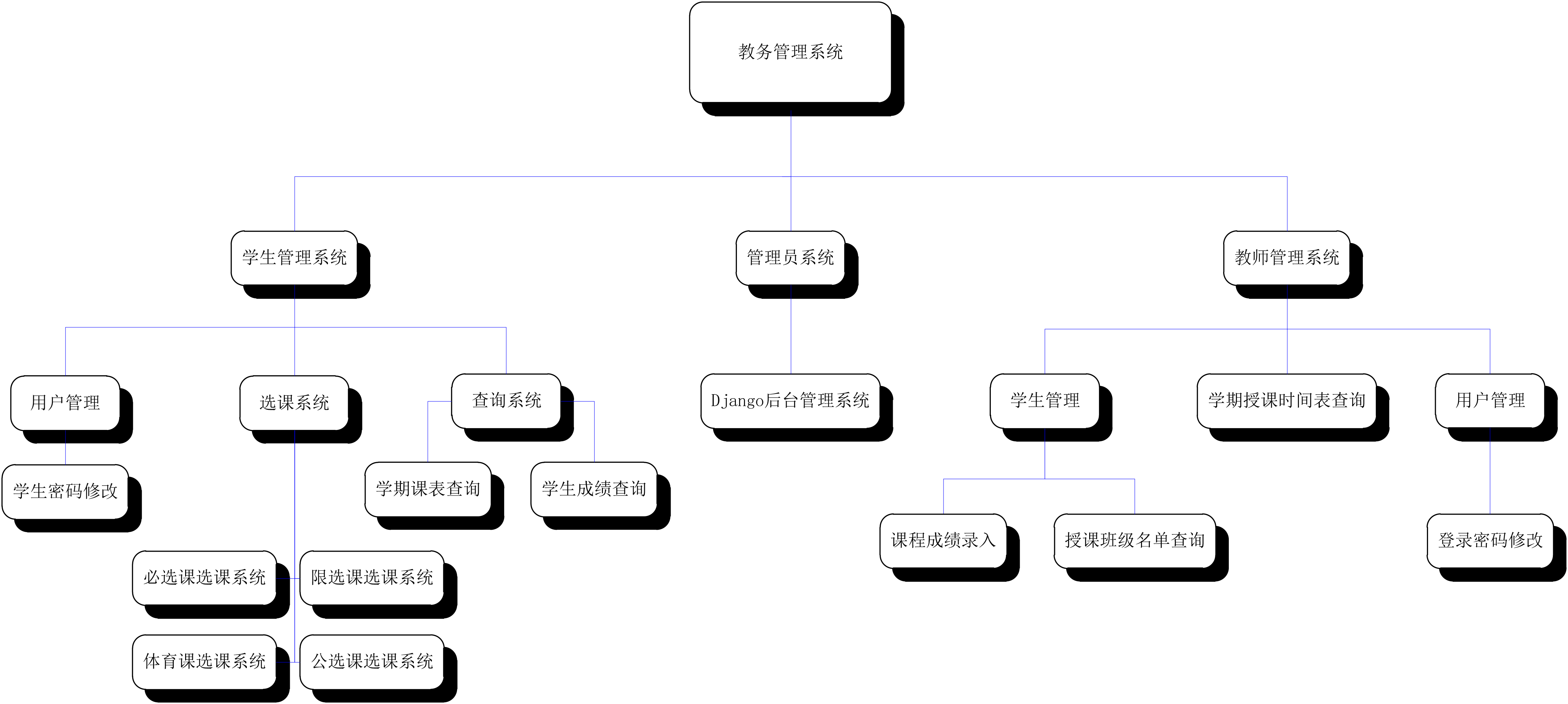
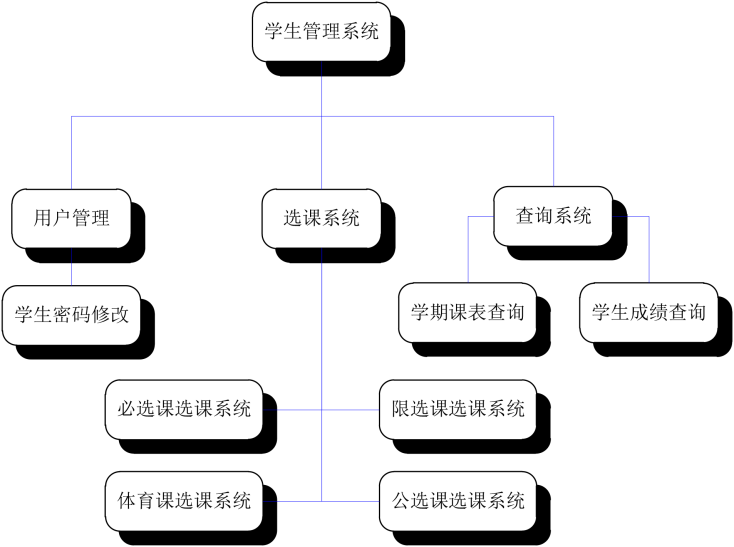
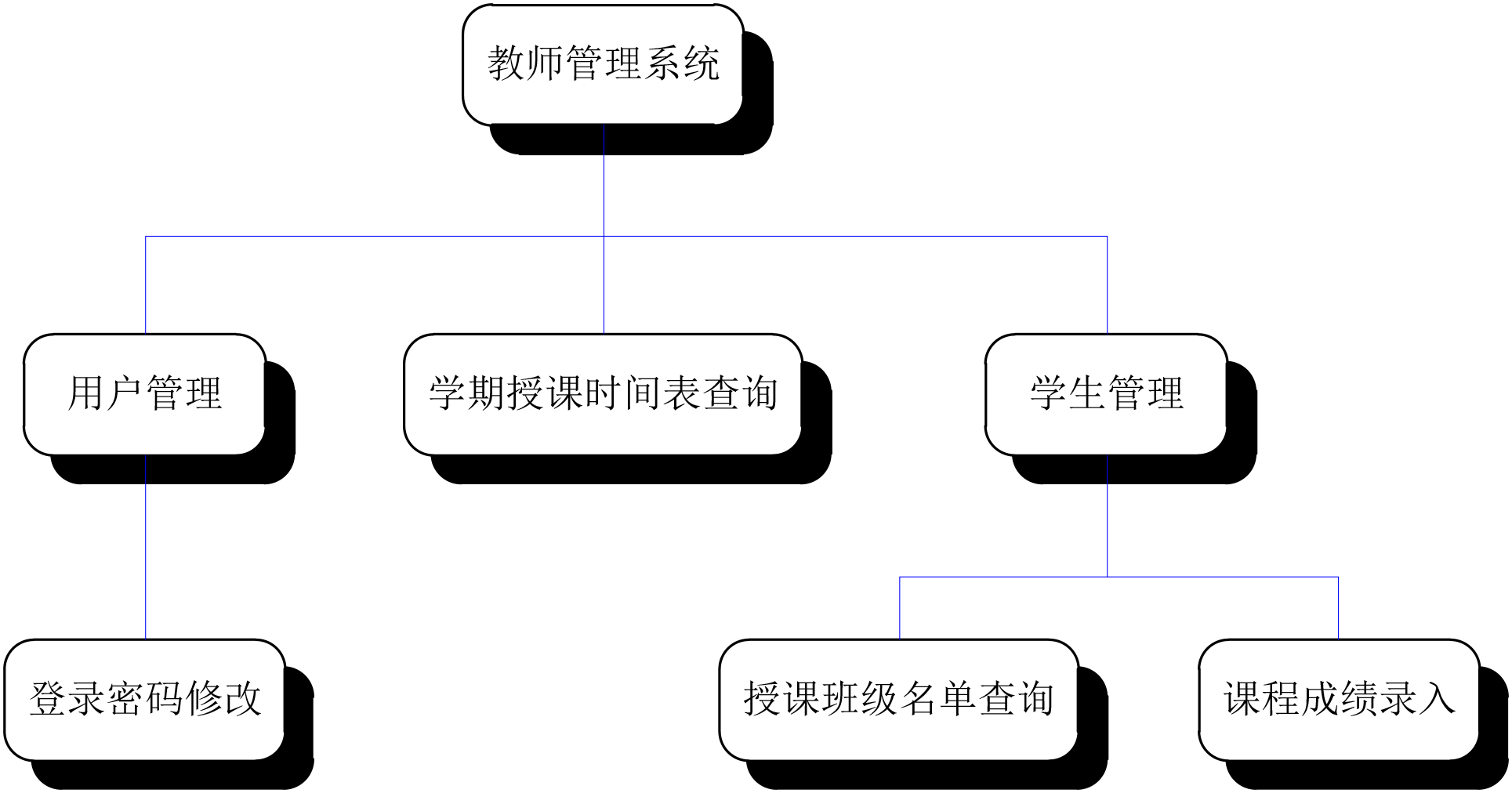
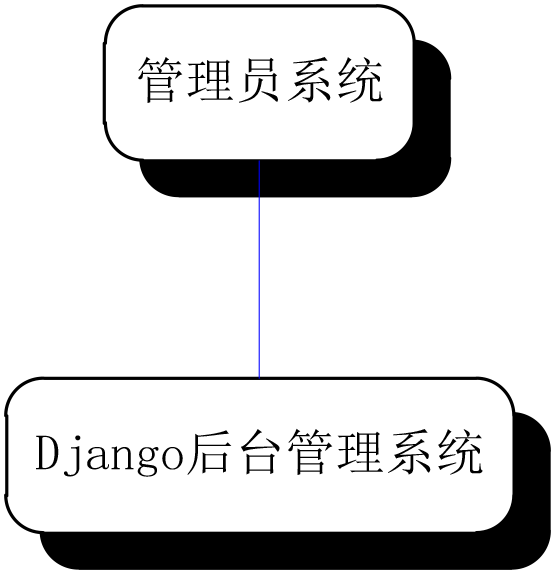
添加少量代码就可获得强大后台。

1. settings.py

Django 的设置，配置文件，比如 DEBUG 的开关，静态文件的位置等。

（7）manage.py 命令行输入启动文件。

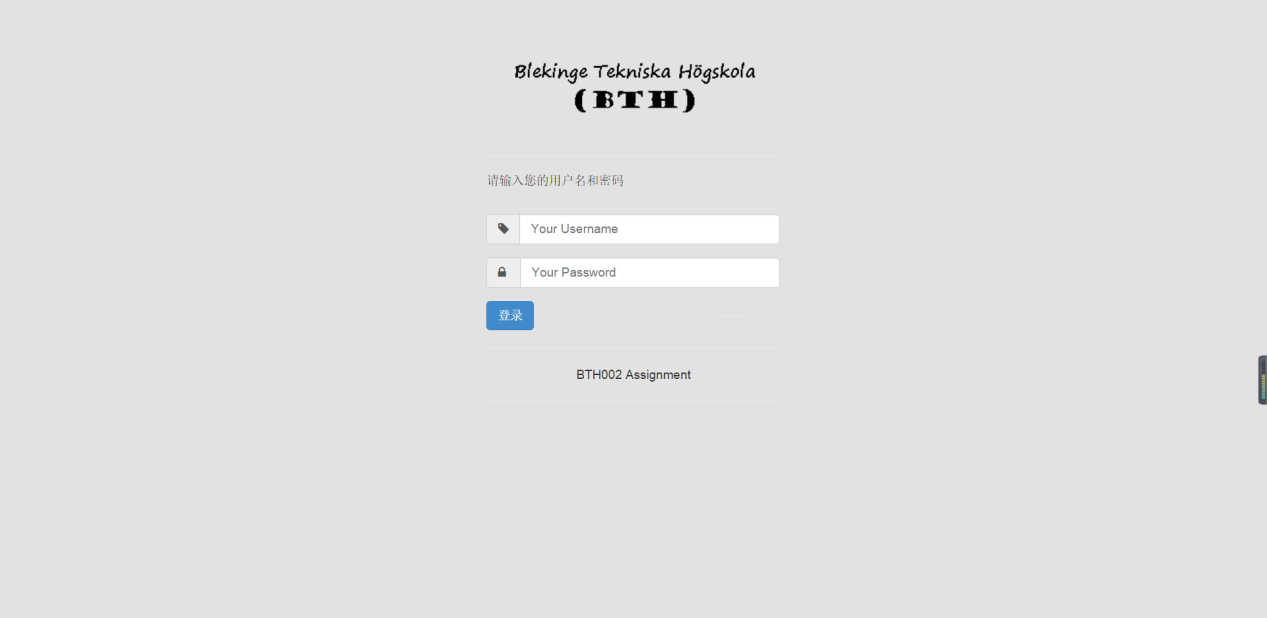
**三.教务系统逻辑结构**



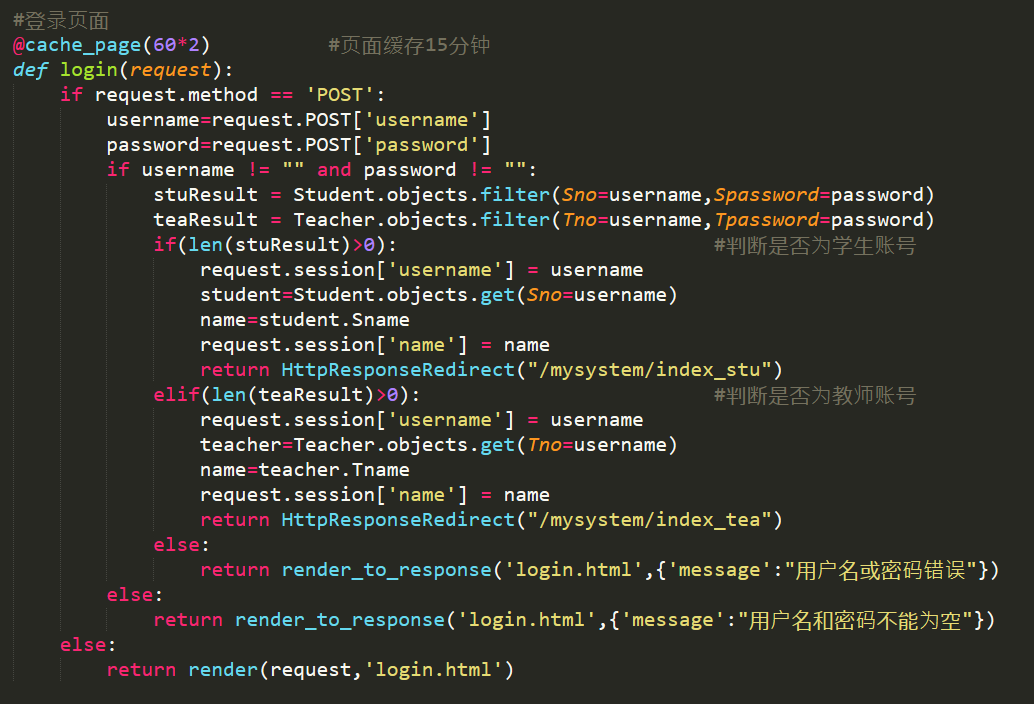
**四．教务系统网站详解**

命令行进入项目目录下后，输入python manage.py runserver。意味着启动了以本地计算机为服务器的教务管理系统网站，同一个局域网的终端可以通过访问本地服务器的IP地址来访问教务管理系统网站。(下列界面网站源码均位于edusys/templates)

1. 登陆界面(login.html)



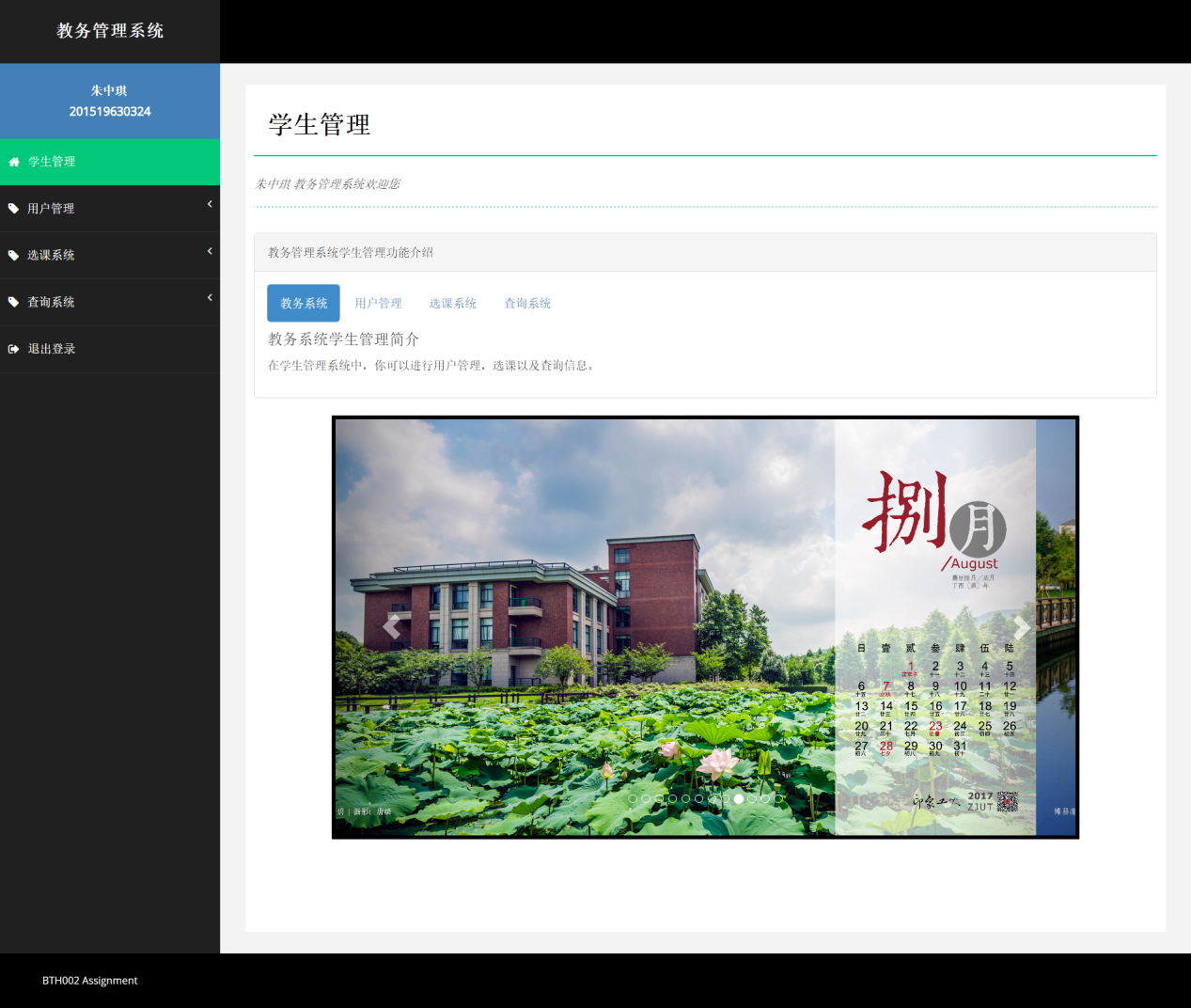
输入正确的用户名和用户名对应的密码后，点击登录，能进入下一个页面。若输入错误，页面将作出提示并返回login.html。本网站所有界面均使用了Bootstrap框架所搭建（响应式页面）。



这是验证用户名密码是否与后台数据库匹配的代码。(在views.py中)

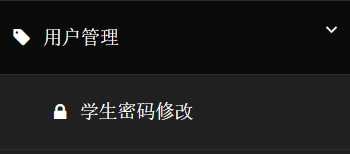
此时输入的用户名分为两种类型，一种是学生账号一种是教师账号，两种账号验证通过后将进入不同界面——学生界面、教师界面，分别为学生和教师提供不同的功能。下面将首先介绍学生界面。

1. 学生界面(index\_stu.html)

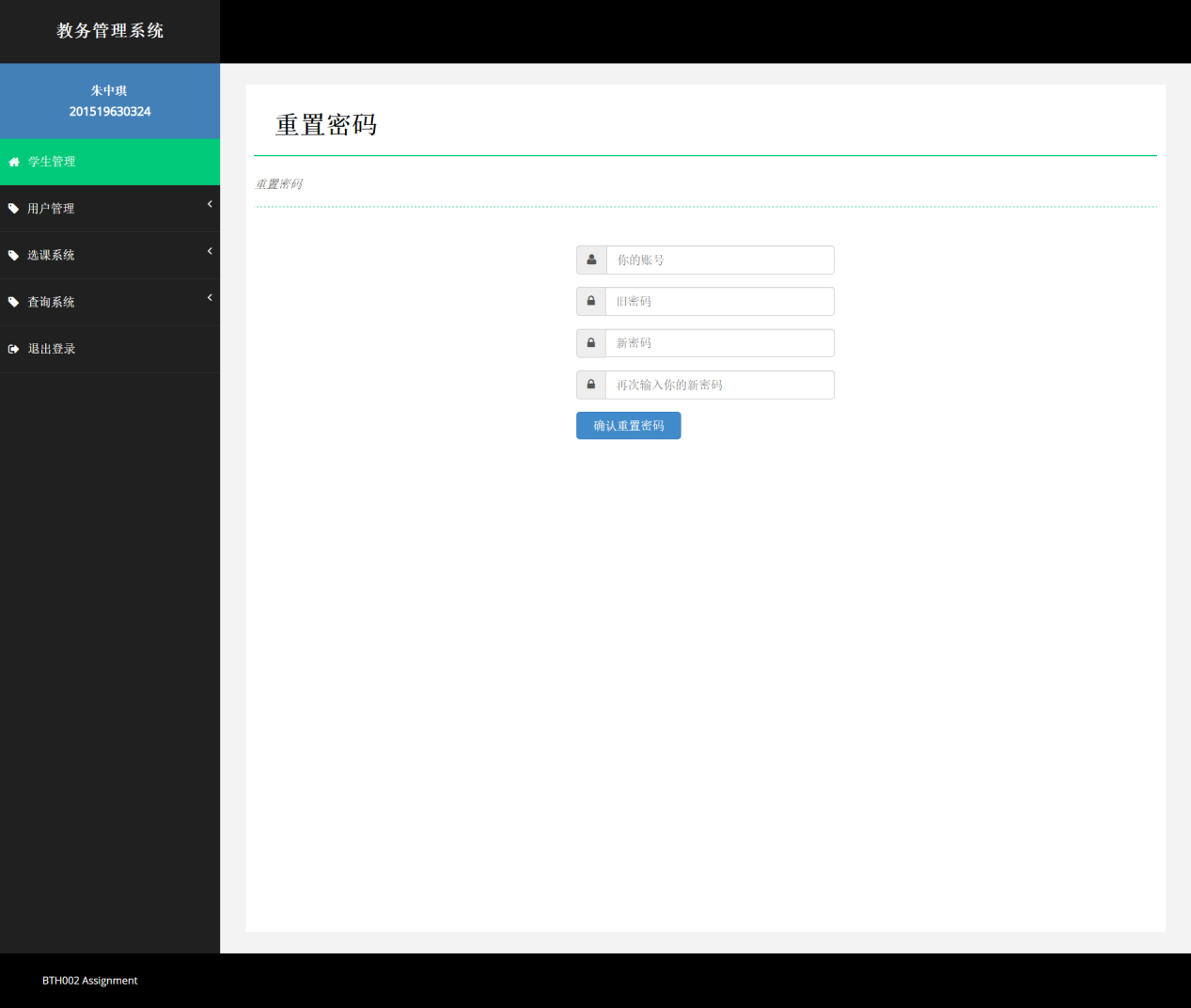


此页面的核心功能为左侧的可继承网页(base\_stu.html),右侧均为美化页面的CSS项目。

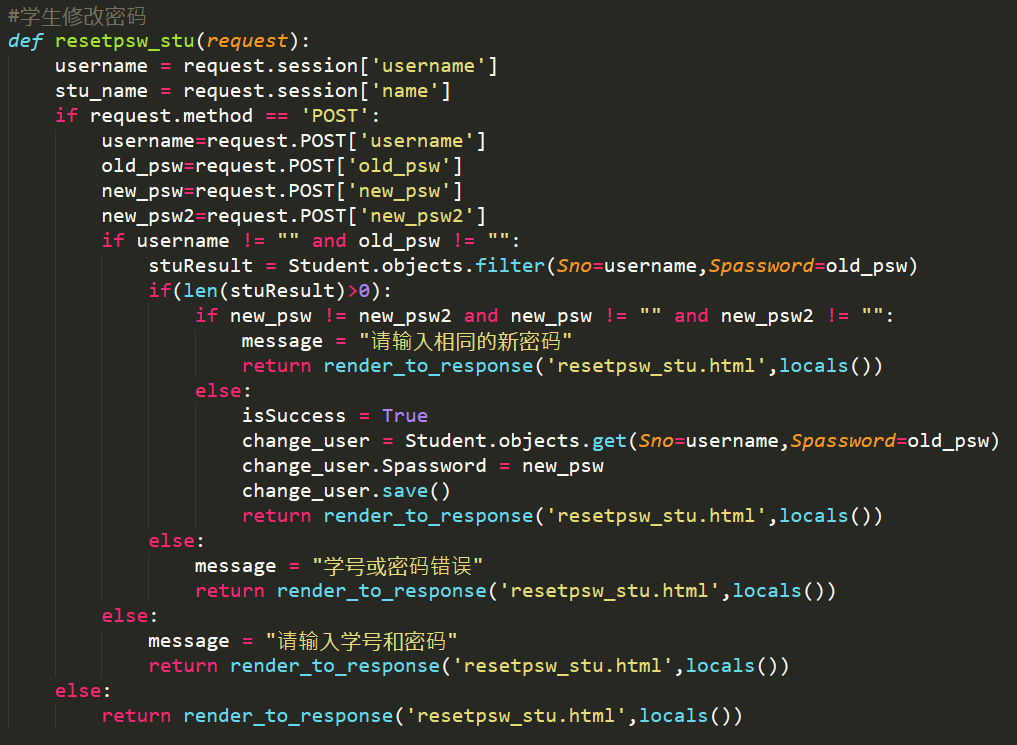
1. (1)用户管理

用户管理功能可提供学生进行密码修改（左侧即为可继承网页）

网页源码为templates中的resetpsw\_stu.html



学生输入账号及原密码还有新密码，若账号和原密码能与后台数据库相匹配，则这次密码修改将成功。



这是学生修改密码的源代码。

1. (2)选课系统

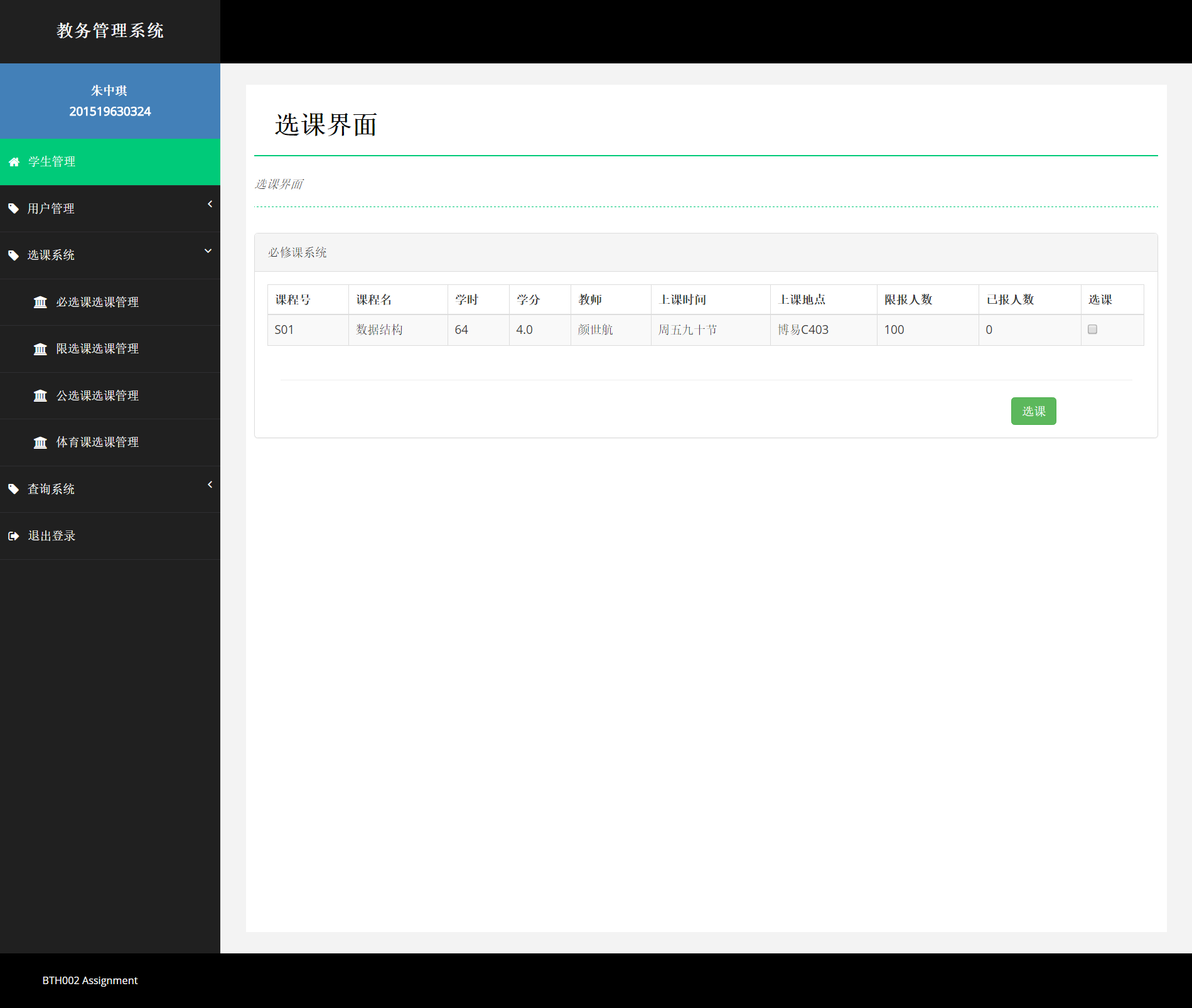
这四个界面功能相同，不同之处根本性在于课程在数据库中的属性不同。

四个界面源代码分别为：

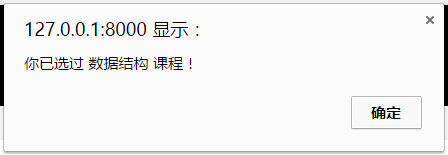
base\_courses.html diselectives\_courses.html

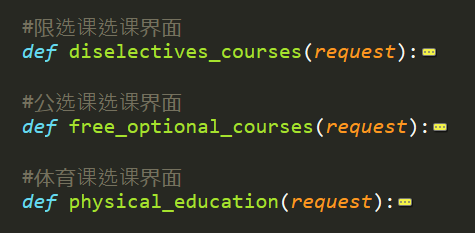
free\_optional\_courses.html

physical\_education.html



勾选课程之后，若学生所属的数据库表中没有这节课，将提示选课成功，并且将此课程加入该学生数据库表中的课程中。若学生已经选择过该课程了将弹出JS弹窗提醒学生课表中已经包含该课程了。无论最后是否选课成功都将返回选课界面。

下面为必修课选课的源代码，限选课、公选课、体育课的代码类似。



2.(3)查询系统

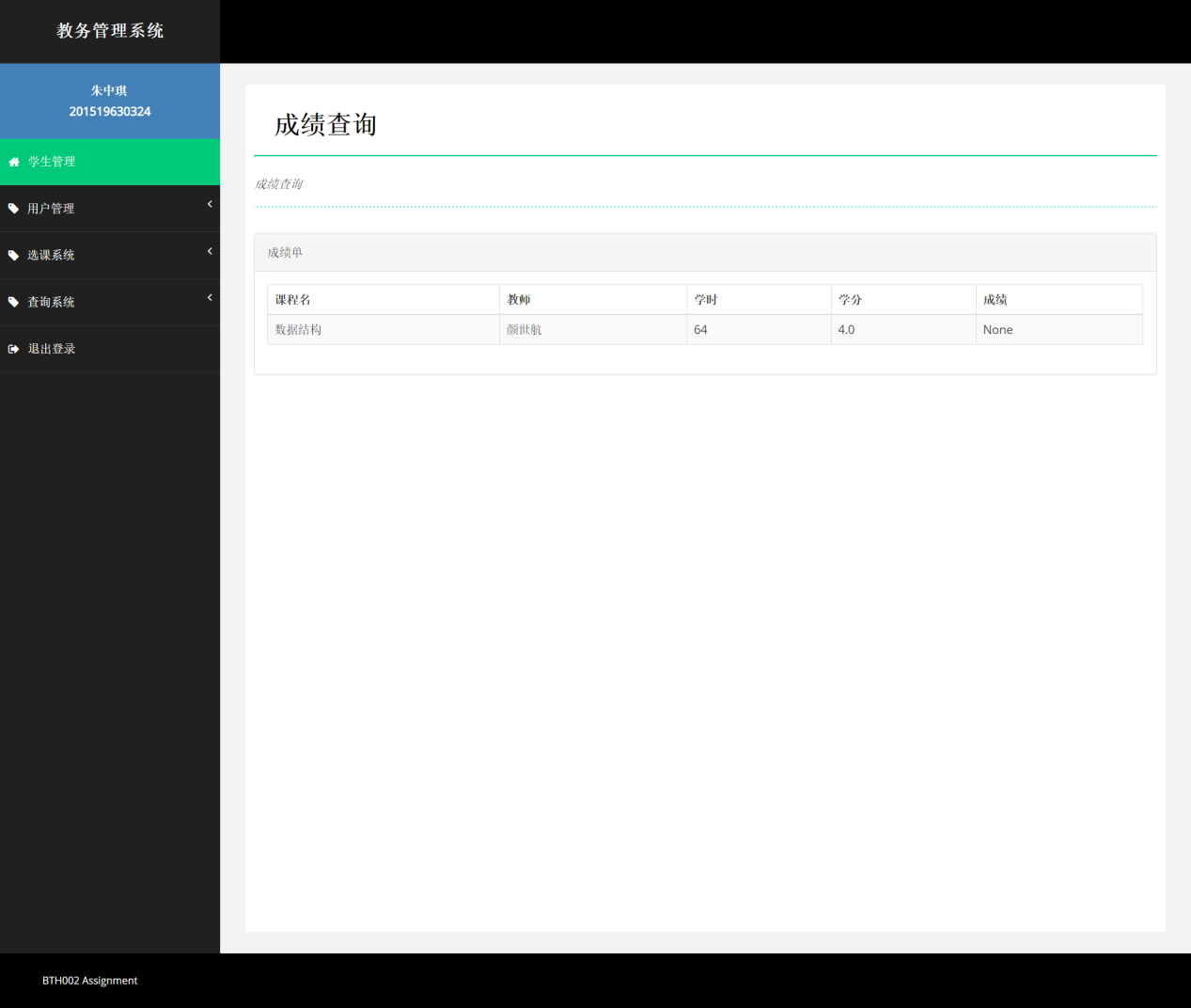
分为成绩查询和课表查询

源代码:

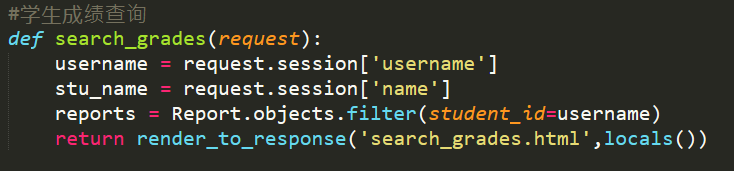
search\_grades.html

search\_courses.html

成绩查询功能：

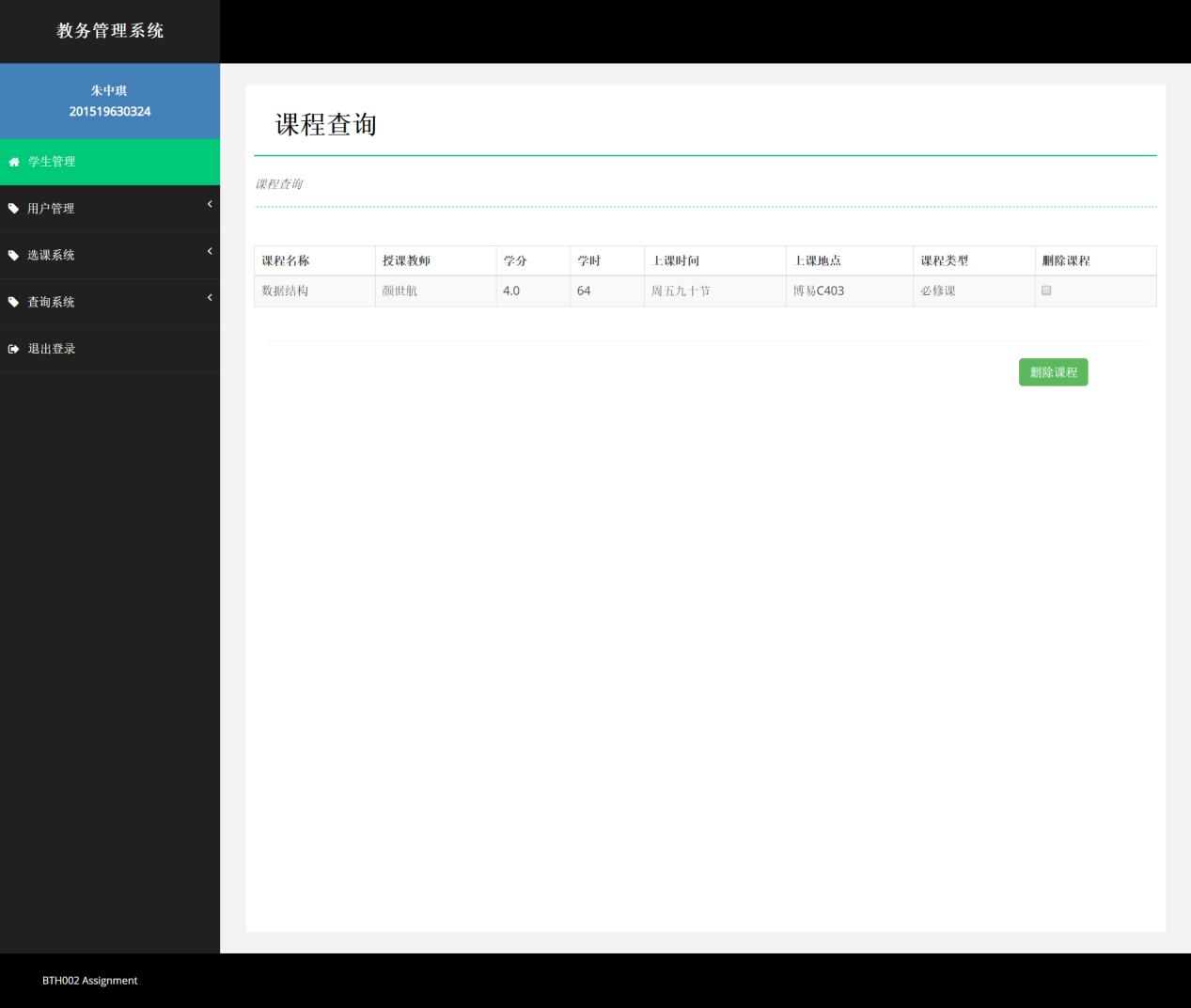


会显示学生所选课程的期末成绩，若教师此时还未输入成绩则显示None

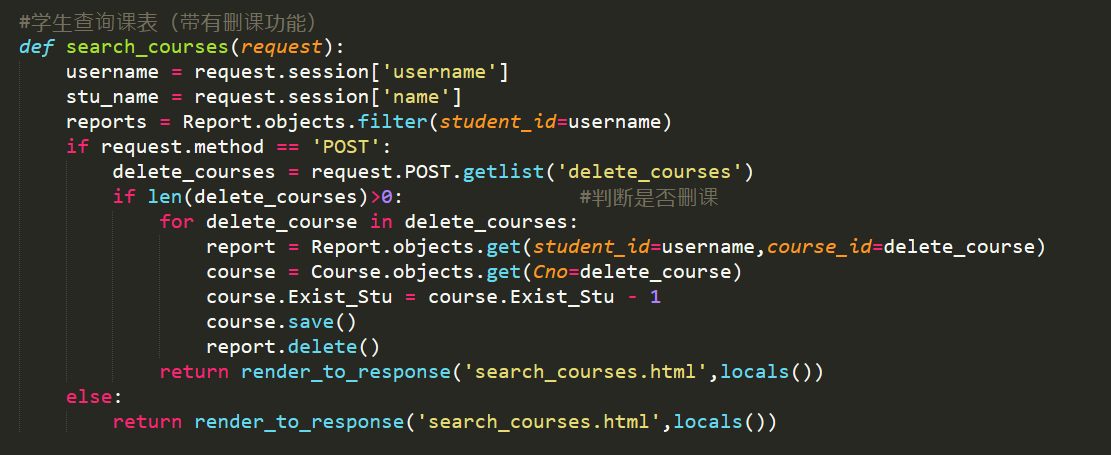


该功能源代码。

学期课表功能查询：

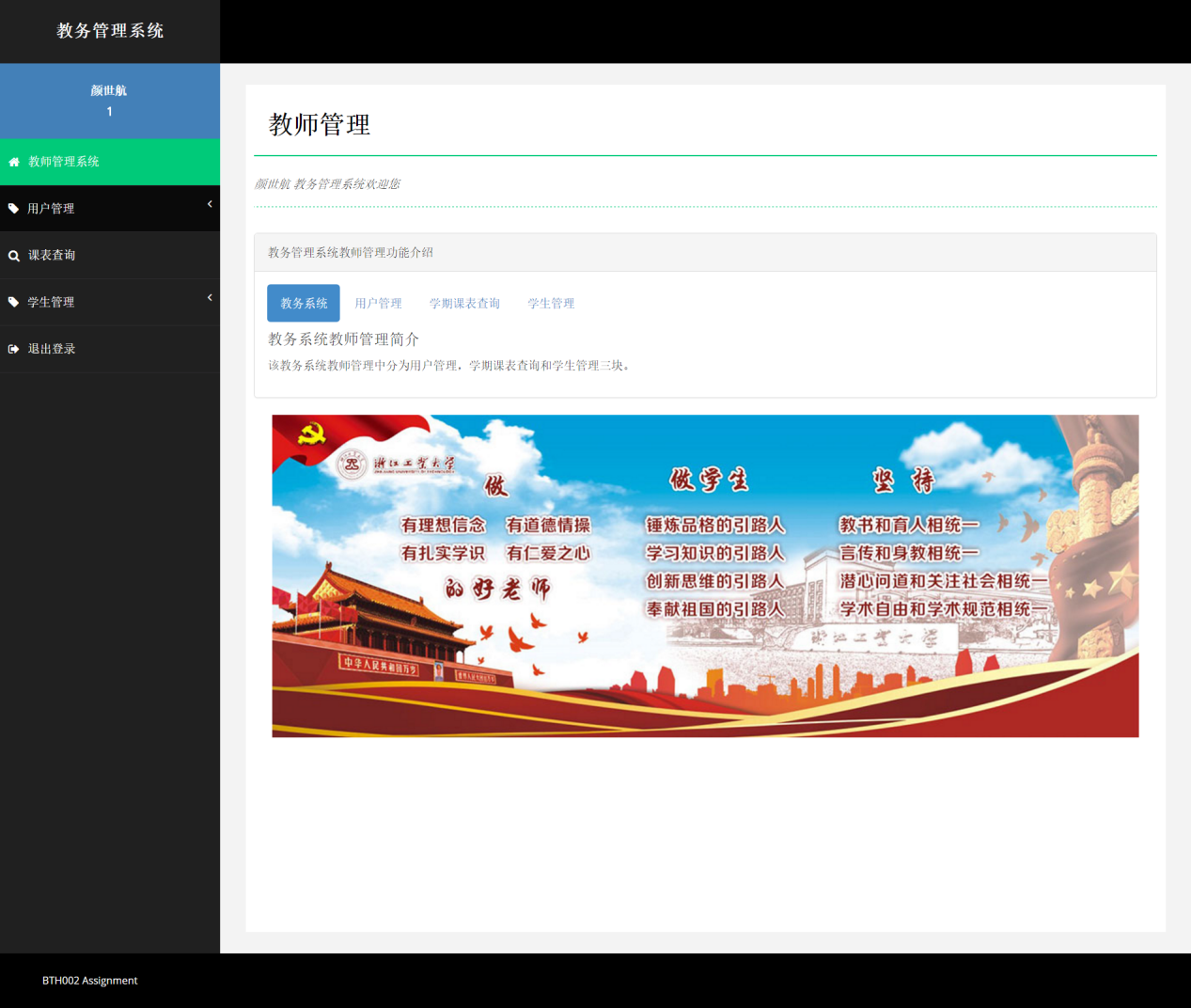


页面中会显示学生数据库表中所包含的课程，并且提供学生删除所选课程的功能。删除课程不仅是从该页面上删除，同时数据库表中该课程也将被删除。下图为查询课表源代码（带有删除功能）



1. 教师界面(index\_tea.html)

教师界面与学生界面大同小异

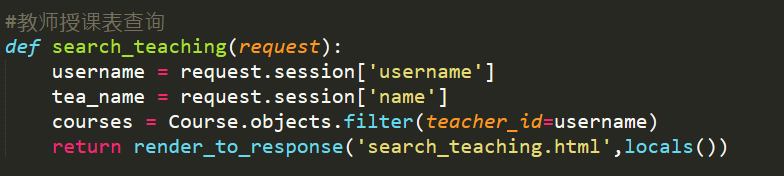


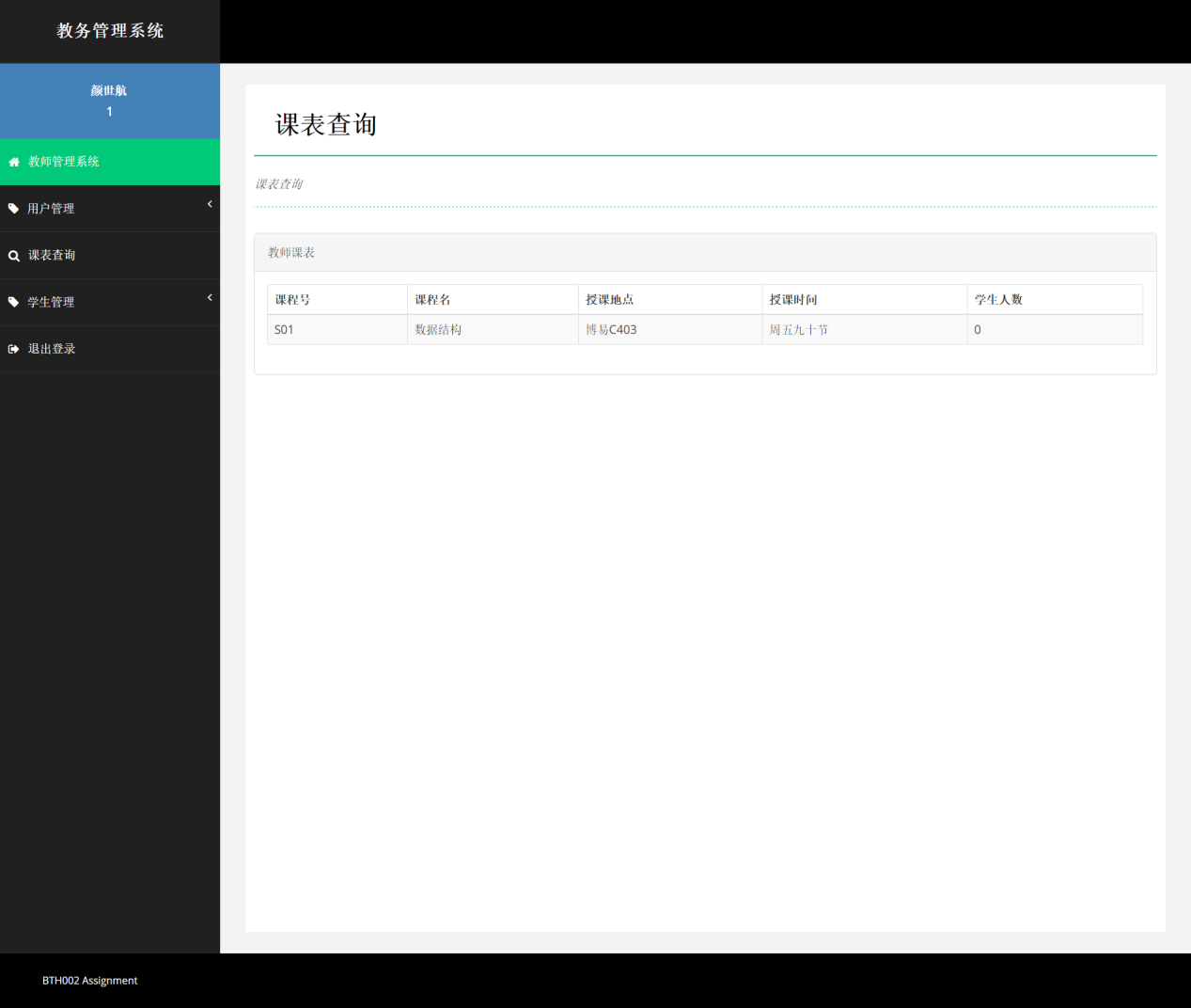
1. (1)用户管理

与学生界面相同，修改密码功能。(resetpsw\_tea.html)

1. 课表查询

教师可以通过该页面查询自己所授课程的信息。





1. (3)学生管理

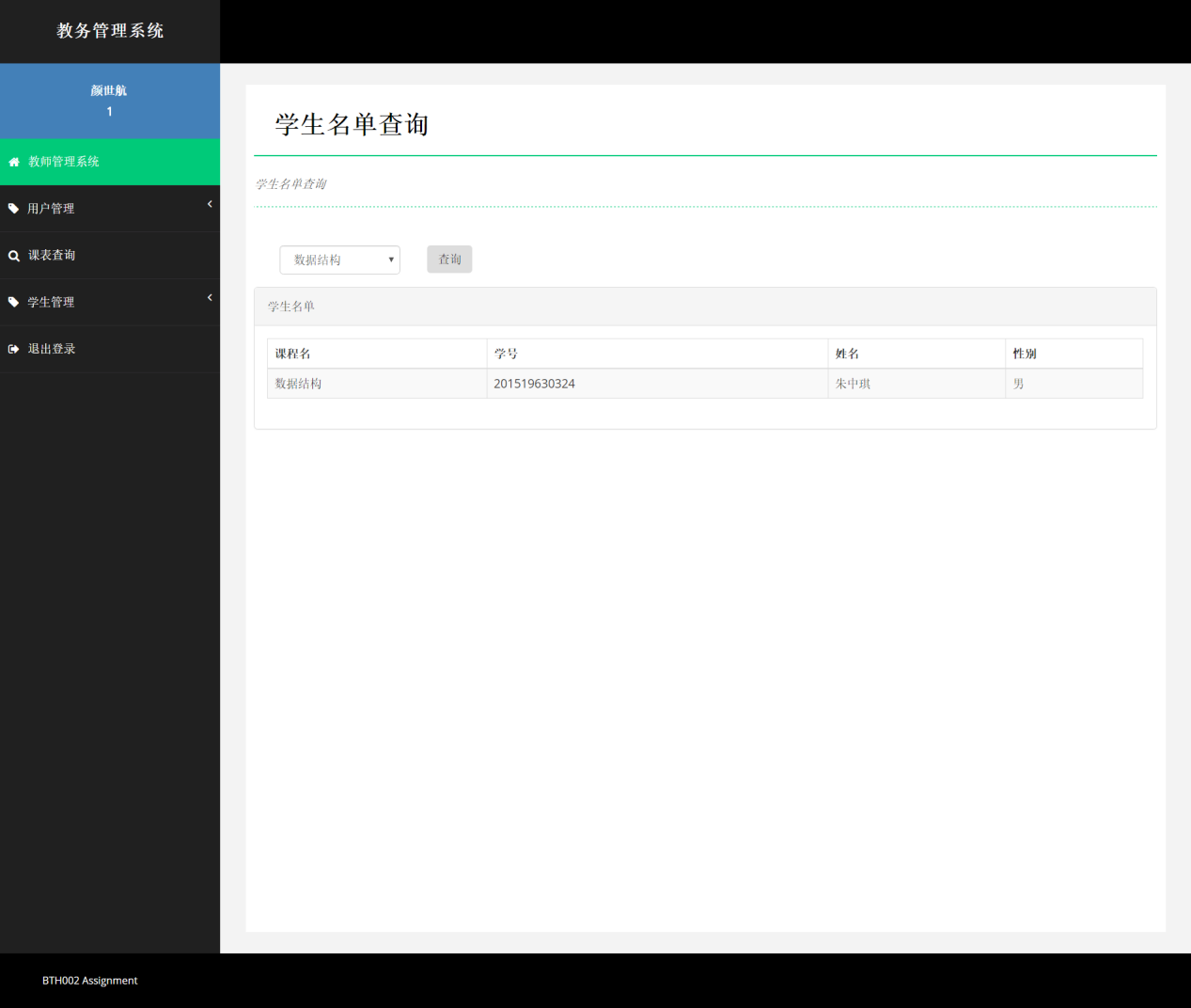
此功能下教师能对学生名单进行查询，并且录入学生的期末课程成绩。

网页源代码:

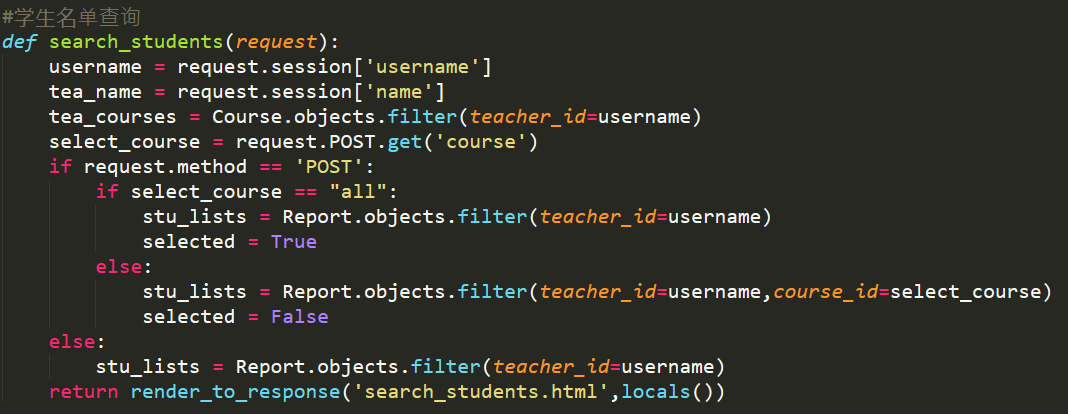
input\_grades.html

search\_students.html

学生名单查询：

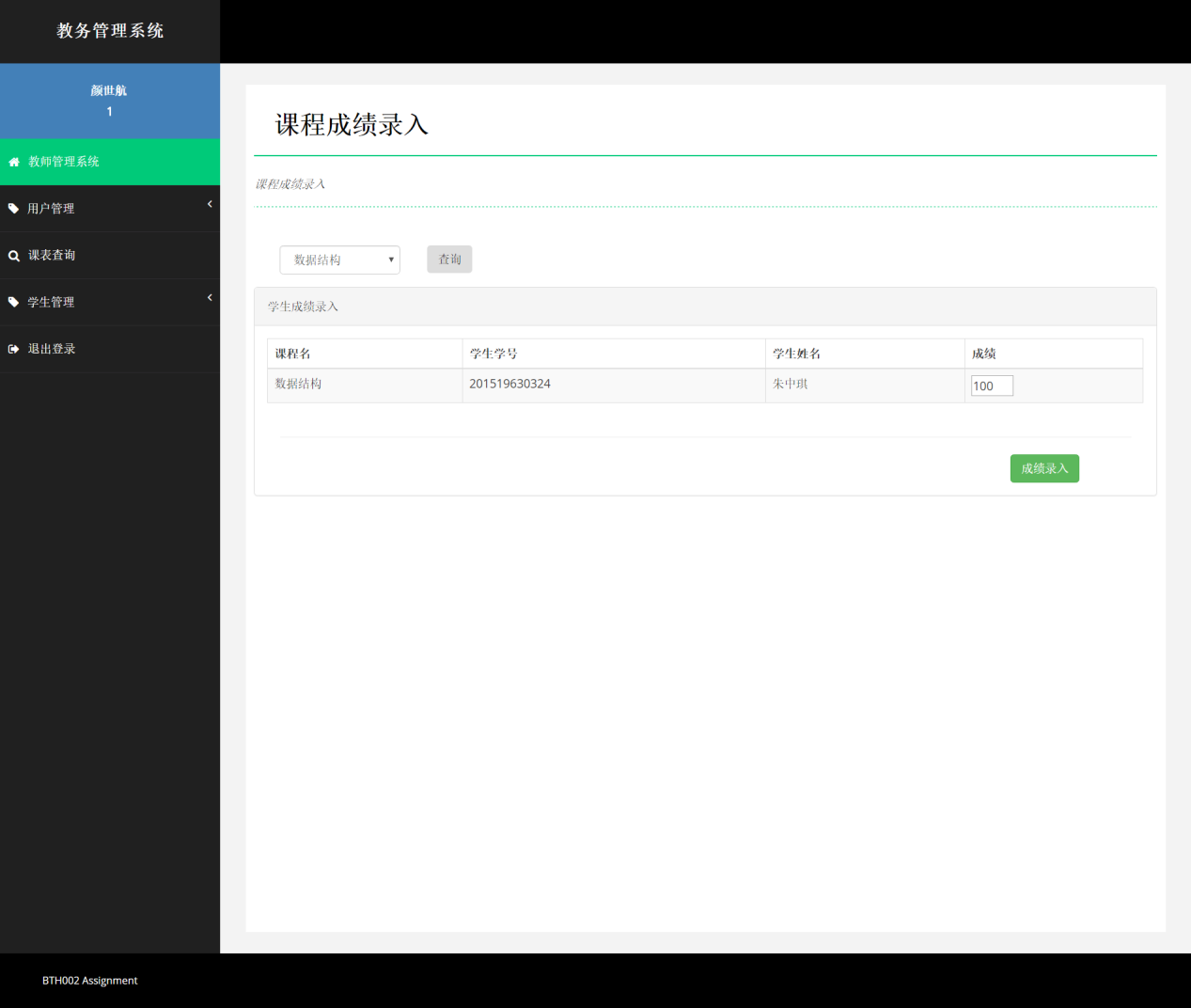


这是此功能的源代码：



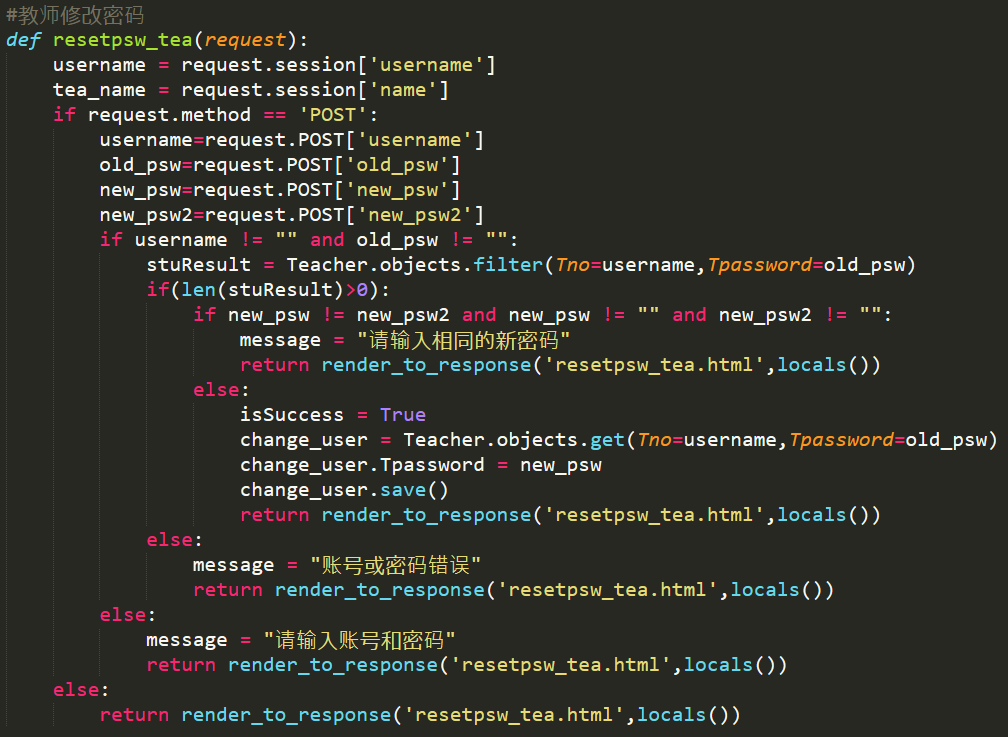
从数据库表中调出相应的数据显示在页面上。

课程成绩录入：



教师可在此页面中输入并修改学生的期末成绩。

下面是该功能的源代码



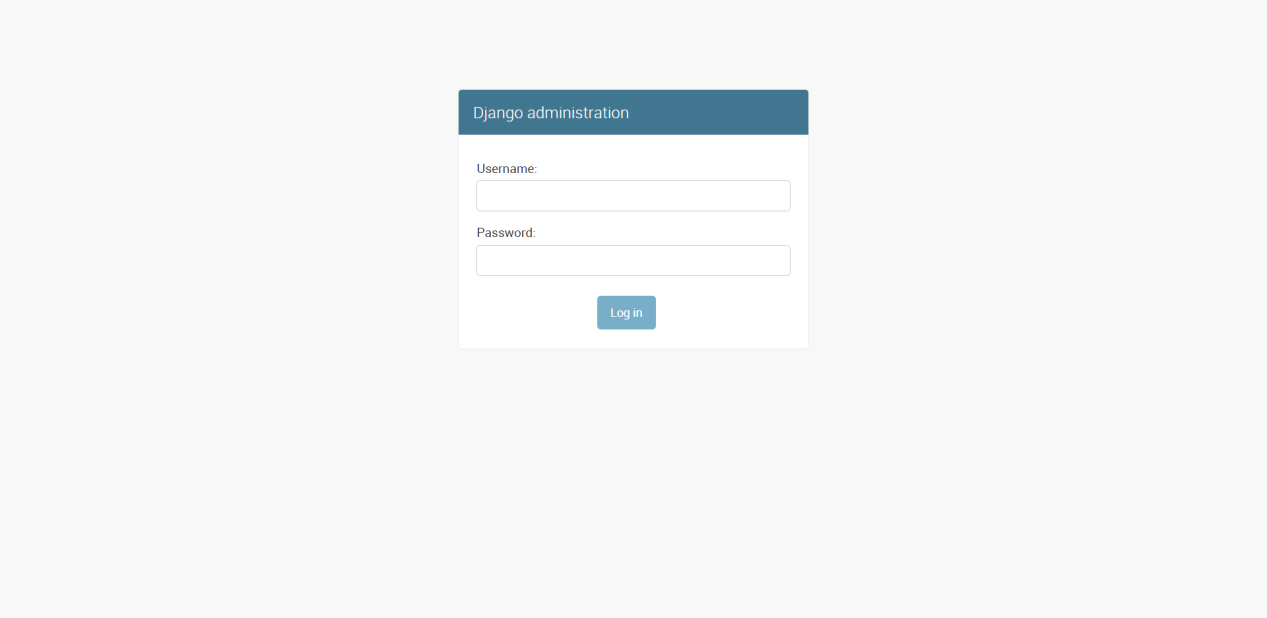
**五．教务系统后台程序**

在新建Django框架时，自带的Django后台也会同时生成，在cmd命令行中创建超级管理员账号来管理后台程序。

1. 创建超级管理员



1. 登录后台程序

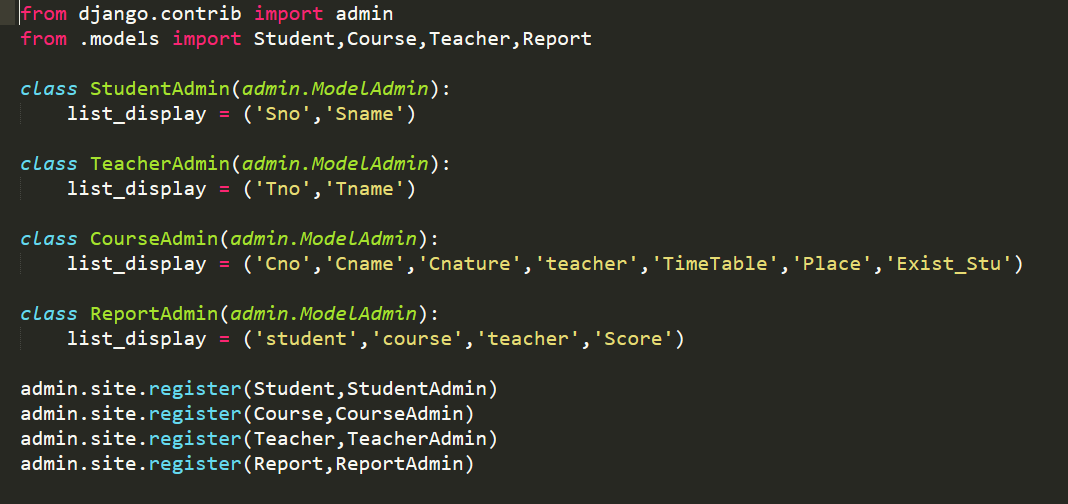


输入在cmd命令行中设置的用户名与密码，将登录到后台程序中。

1. 后台管理程序

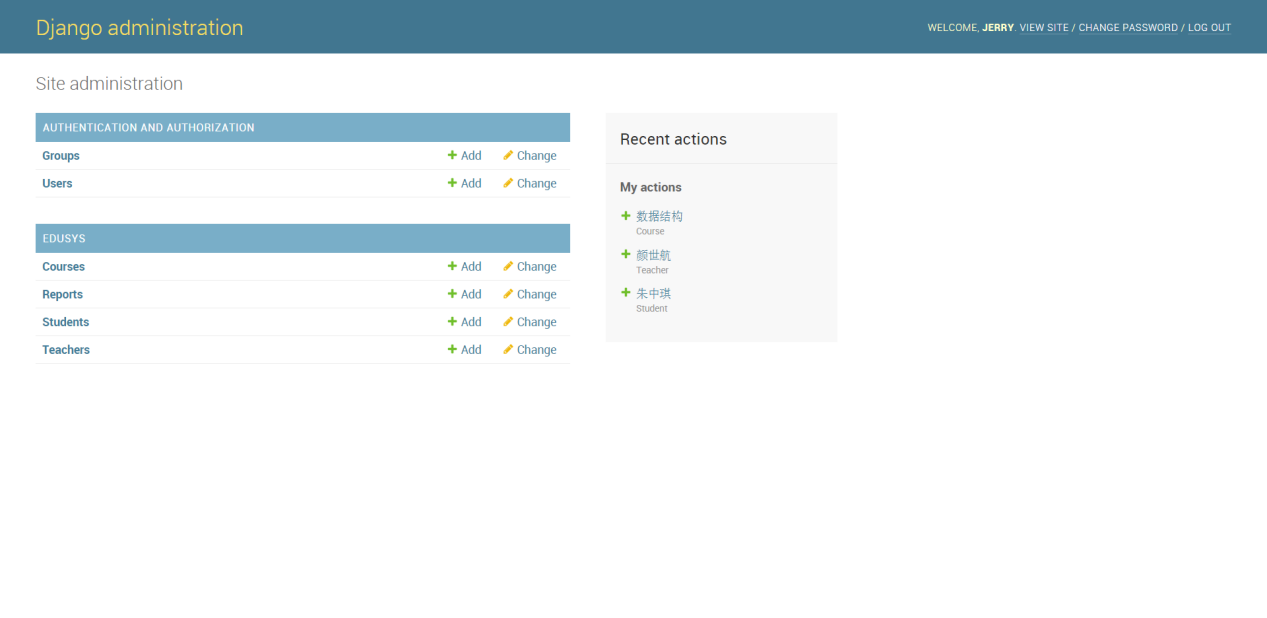
要在管理数据，首先要为后台添加数据，将后台与数据库相连。

源代码:admin.py



登录后台页面后，可见看见在admin.py中添加的四个元组：

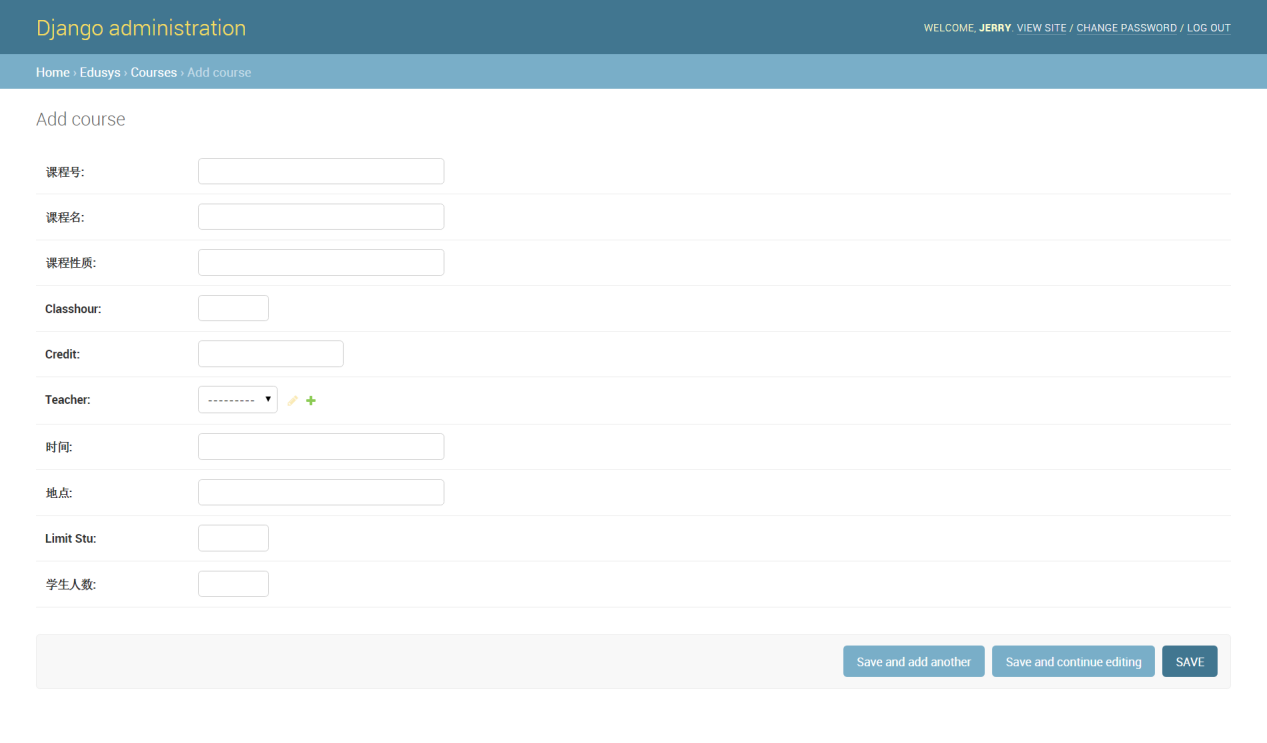
课程 成绩 学生账号 教师账号



图中Courses Reports Students Teachers相当于数据库中的表，点击Add可以进一步增加元组数。

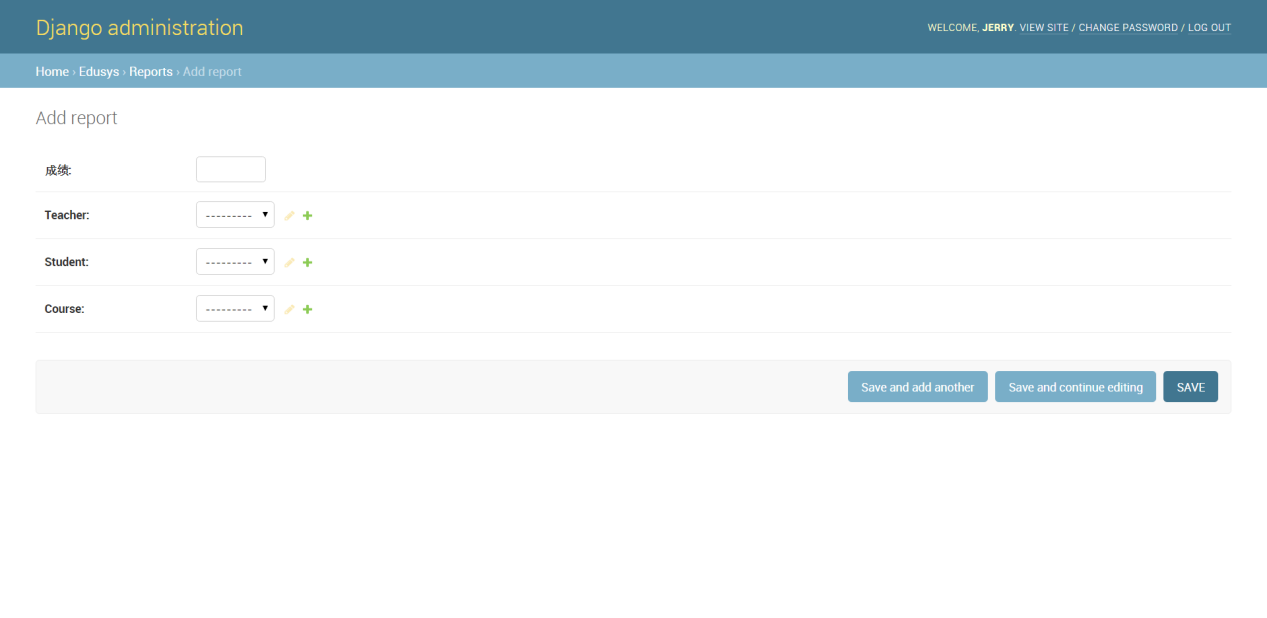
下列页面为后台为各个数据库表增加元组时赋予他们的属性。

Courses表：



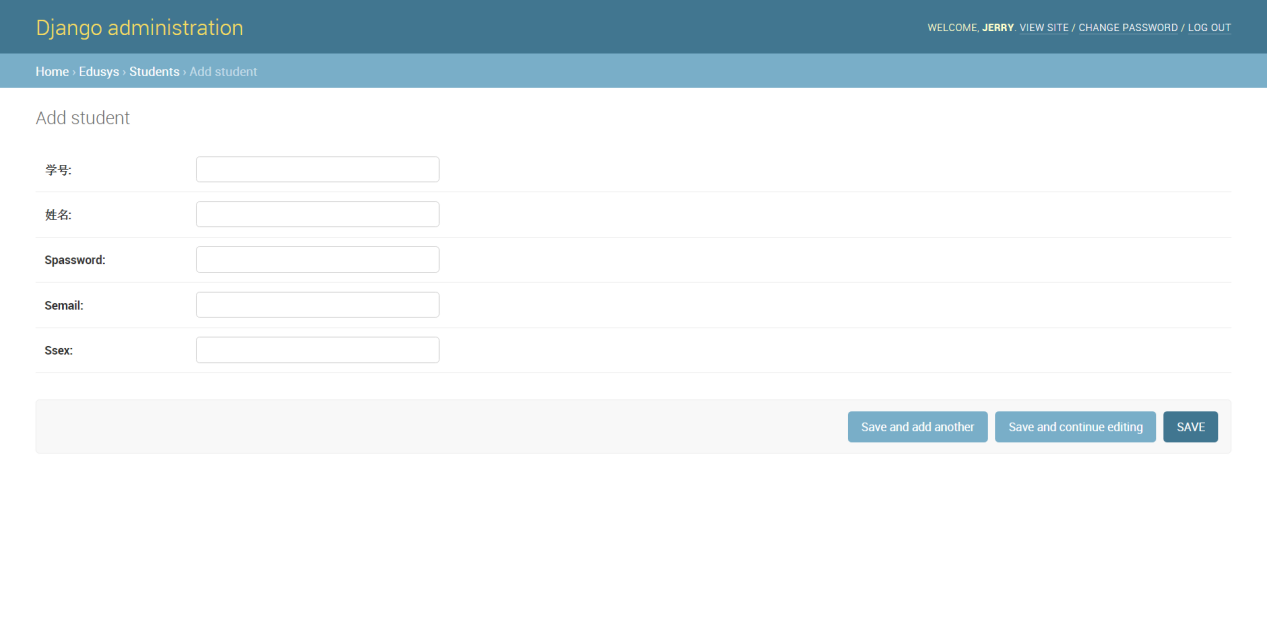
管理员通过该表在数据库中添加课程供学生选择。

Reports表:



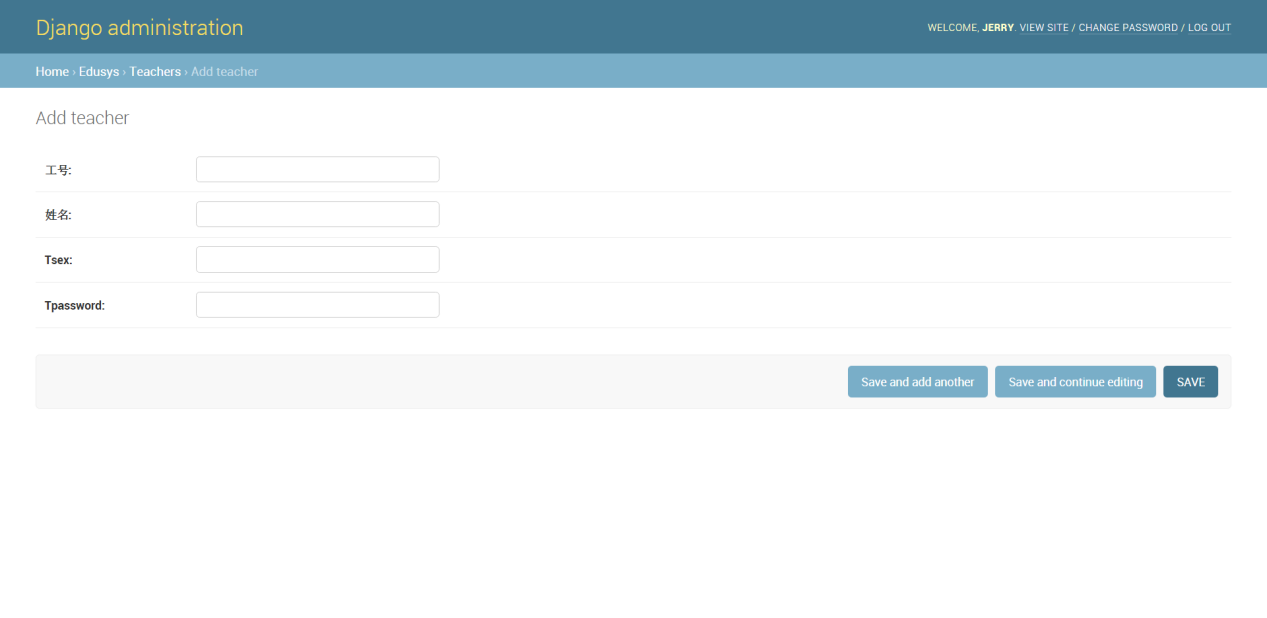
不仅教师能够修改学生成绩，管理员也可以通过该表修改所有课程学生的成绩。

Students表：



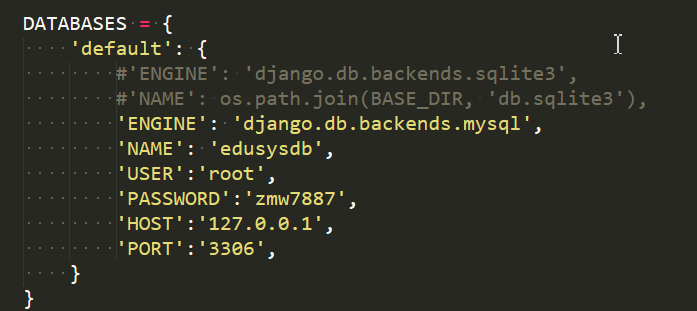
在该表中添加学生账号和对应的学生信息。

Teacher表：

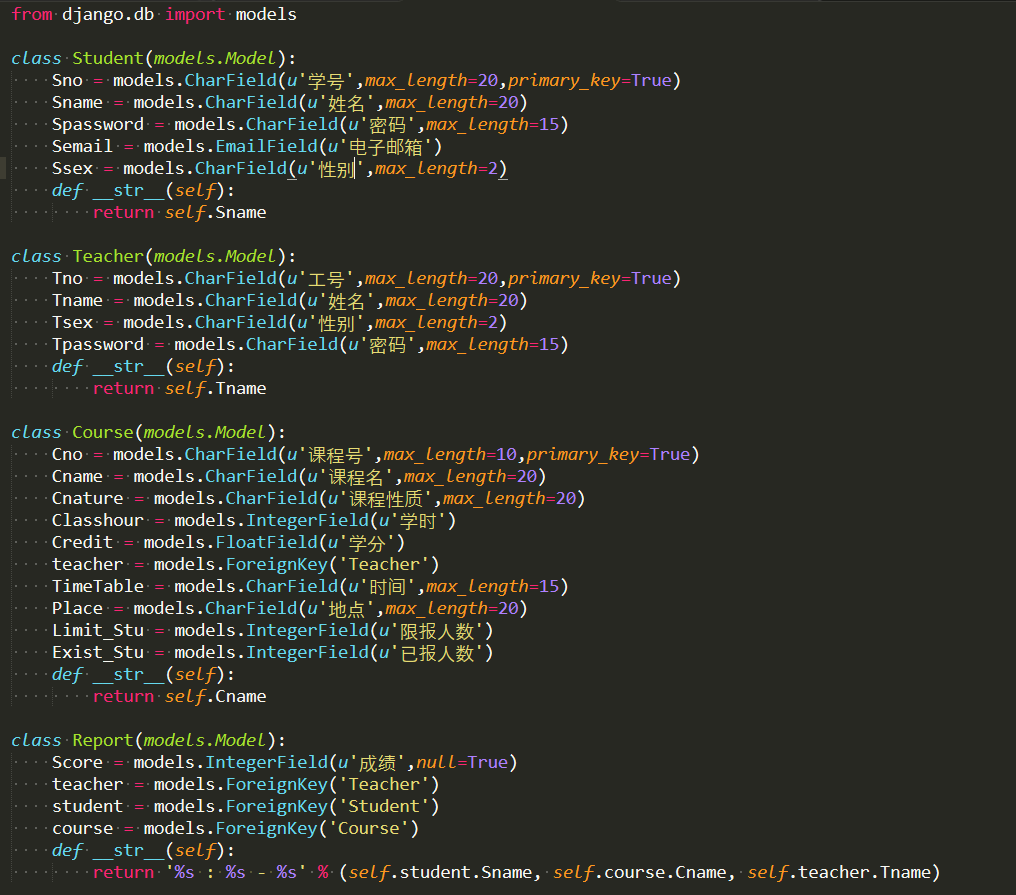


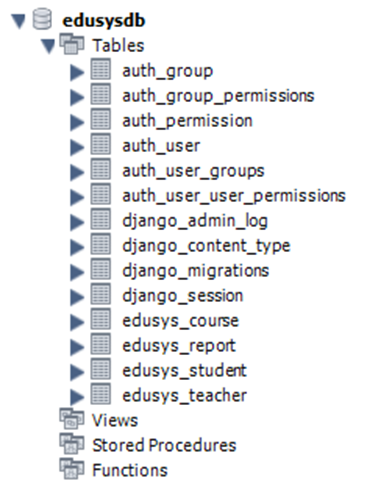
在该表中添加教师账号和对应的教师信息。

**六．教务管理系统数据库连接**



先在settings.py文件中添加要连接的数据库类型，名称，用户名及其密码，HOST和PORT。





然后models.py中创建相应各个表的对象类及相应属性，通过对象关系映射（ORM）到mysql对应数据库中。

通过在CMD命令窗口中输入 python manage.py makemigrations 和 python manage.py migrate来进行数据库同步。

**七．高并发问题**

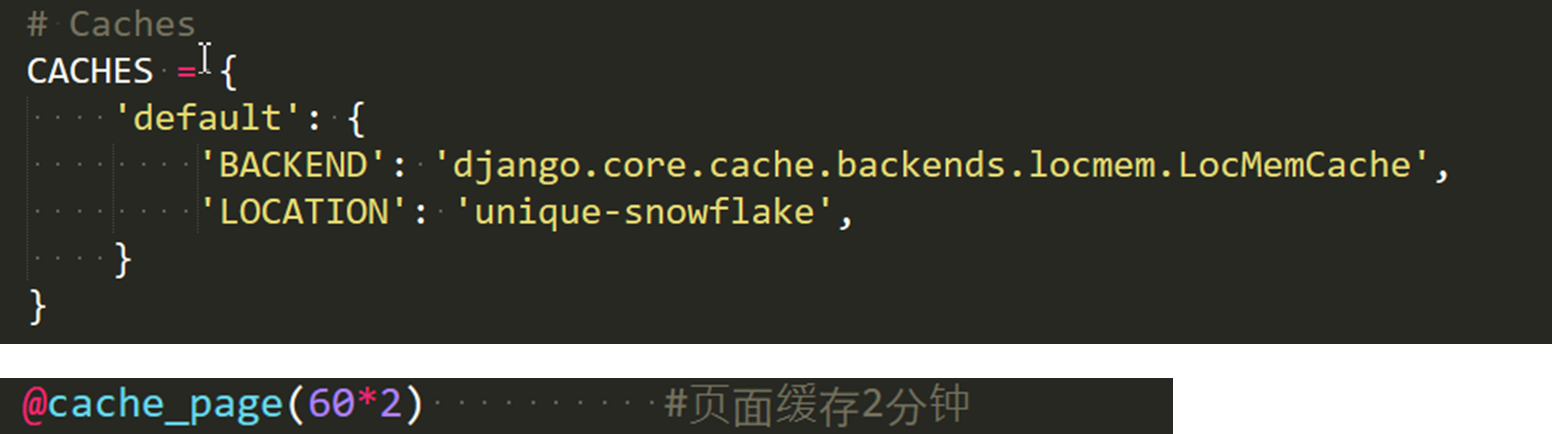
nginx + uWSGI 为Django提供高并发：

****

nginx 的并发能力超高，在纯静态的 web 服务中更是突出其优越的地方，由于其底层使用 epoll 异步IO模型进行处理，使其深受欢迎。uWSGI，是一个Web服务器，他通过uwsgi协议与应用程序进行数据交流。

因此我们把Nginx作为Uwsgi服务器的反向代理服务器，同时它也是负载均衡器。当遇到选课高并发情况时，网站可以进行负载均衡来缓解服务器压力。

缓存系统：



缓存系统工作原理：

对于给定的网址，尝试从缓存中找到网址，如果页面在缓存中，直接返回缓存的页面，如果缓存中没有，一系列操作（比如查数据库）后，保存生成的页面内容到缓存系统以供下一次使用，然后返回生成的页面内容。

我们是通过使用本地内存来缓存。

数据库读取和存储优化：

1、页面静态化，对一些不需要频繁更新的页面

2、高并发情况下，可以把多个请求的查询合并到一次进行，减少对数据库的访问次数。多次修改请求也可以合并。

3、使用索引，索引可以看作是特殊的缓存。