

**基于python的中小学生在线家庭作业系统的设计与实现**

姓 名： 王语婕、程章一宽

学 号： 1230651169、1210514022

指导教师： 刘宁

**2025年7月**

摘 要

在各学校的教学过程中，学生的作业管理是一项非常重要的事情。随着计算机多媒体技术的发展和网络的普及，“基于网络的学习模式”正悄无声息的改变着传统的教室学习模式，“基于网络的教学平台”的研究和设计也成为教育技术领域的热点课题。采用当前流行的B/S模式以及3层架构的设计思想通过Python技术来开发此系统的目的是建立一个配合网络环境的可以师生互动的电子作业提交与反馈的平台，这样可以有效地解决作业管理混乱的局面。本系统设计了3种角色：管理员，学生和教师。系统主要包括个人中心、学院管理、专业管理、班级管理、学生管理、教师管理、课程信息管理、作业布置管理、作业提交管理、作业批改管理等功能模块。

本文首先介绍了学生作业管理系统管理技术的发展背景与发展现状，然后遵循软件常规开发流程，首先针对系统选取适用的语言和开发平台，根据需求分析制定模块并设计数据库结构，再根据系统总体功能模块的设计绘制系统的功能模块图，流程图以及E-R图。然后，设计框架并根据设计的框架编写代码以实现系统的各个功能模块。最后，对初步完成的系统进行测试，主要是功能测试、单元测试和性能测试。测试结果表明，该系统能够实现所需的功能，运行状况尚可并无明显缺点。

关键词：学生作业管理系统；Django ；Mysql数据库

# 

# ABSTRACT

In the teaching process of each school, student's homework management is a very important thing. With the development of computer multimedia technology and the popularity of the network, "web-based learning mode" is quietly changing the traditional classroom learning mode, "web-based teaching platform" research and design has become a hot topic in the field of education technology. Using the current popular B/S mode and three-layer architecture design ideas through Python technology to develop this system is to establish a network environment can be interactive with teachers and students electronic homework submission and feedback platform, so that can effectively solve the chaotic situation of homework management. This system designs three kinds of roles: administrator, student and teacher. The system mainly includes individual center, college management, professional management, class management, student management, teacher management, course information management, homework assignment management, homework submission management, homework correction management and other functional modules.

This article first introduces the student job management system development background and development status quo of management technology, and then follow the routine software development process, first of all, in view of the system and the selection of suitable language development platform, according to the requirement analysis module and database structure design, and then based on the system's overall function module design rendering system function module chart, flow diagram and e-r diagram. Then, design the framework and write code according to the designed framework to achieve each functional module of the system. Finally, the preliminary completed system is tested, mainly functional test, unit test and performance test. The test results show that the system can achieve the required functions, and the running condition is fair and there is no obvious defect.

**Key words：**tudent homework management system; Django. The Mysql database

目 录

[第一章 概述 4](#_Toc1899)

[1.1引言 4](#_Toc23913)

[1.2课题研究目的 4](#_Toc2651)

[1.3开发现状分析 5](#_Toc9908)

[1.4课题开发意义 6](#_Toc6220)

[第二章 开发技术介绍 7](#_Toc7942)

[2.1 系统开发平台 7](#_Toc10759)

[2.2 平台开发相关技术 7](#_Toc22122)

[第三章 系统分析 10](#_Toc10304)

[3.1 可行性分析 10](#_Toc12340)

[3.2性能需求分析 11](#_Toc22693)

[第四章 系统设计 11](#_Toc13403)

[4.1功能结构 11](#_Toc8267)

[4.2 数据库设计 12](#_Toc9394)

[第五章 系统功能实现 19](#_Toc11909)

[5.1管理员功能模块 19](#_Toc15589)

[5.2教师功能模块 31](#_Toc31593)

[参考文献 41](#_Toc31588)

[附录 42](#_Toc675)

[一、根据第四章内容通过SQL创建的数据表示例： 42](#_Toc30426)

[二、检查代码完整性的测试性代码 44](#_Toc14811)

# 第一章 概述

## 1.1引言

信息化是当今世界经济和社会发展的大趋势，以网络和多媒体为核心的信息技术已经成为拓展人类能力的创造性工具。随着网络技术、多媒体技术在教育领域应用的不断发展，网络教育已成为当前教育发展的热点。随着计算机应用的普及，使计算机教学深入到课堂。另一方面，校园网已从办公室、教室、图书馆延伸到学生宿舍，这为网上教学提供了广阔的舞台。开发和充分利用网络教学，是当前教育发展的必然趋势。作业是教学过程中教师考核学生成绩的一种方式，新型的网络教学模式的发展渐渐地替代了传统的纸张提交作业的方式。老师布置的作业，学生先在作业本上完成，然后通过班委统一的交给老师，老师手工批改，再把作业发给学生。这种反馈过程下来时间就长了，就很难全面的考核学生这段的学习情况，不利于学生学习，也不利于教师全面的了解学生最近阶段的学习状况。如何缩短从教师布置作业到批改作业的时间差，提高批改作业的速度和质量，已经成为教师们所面临的一大问题。所以随着时代的发展，网络技术、多媒体技术、开发一个学生作业上传和教师批改作业系统是很有必要的。与时俱进，操作简单，更高效地实现学生在线上传系统的无纸化管理和实时性服务。

随着网络技术的不断发展，多媒体技术应用渐渐的出现在教育领域中，网络教育已经成为教育发展的一个热门话题。互联网应用在全球范围内日益普及，在许多的网络服务中，Web给人耳目一新的感觉。在这其中，网络平台开展的工作已经越来越流行，当今网络技术的发展，已经成为一种新的工作方式，人们可以通过网络所提供的平台解决相关事务。在众多网络开发技术中，Python是当前很热门的一种软件，因为它可以进行数据库操作及方便用户控制管理。

## 1.2课题研究目的

1. 便于教师网上办公

随着信息技术在管理中的应用越来越深入和广泛，管理信息系统的实施在技术上已逐步成熟。管理信息系统是一个不断发展的新型学科，任何一个单位要生存要发展，要高效率地把内部活动有机地组织起来，就必须建立与自身特点相适应的管理系统。根据现在学校学生作业管理的状况，设计成一个能提供对学生作业查看、作业提交以及教师布置作业、查看作业、批改作业的系统，该系统不仅仅要方便，而且具有很大的实用性和可行性，大量简化管理人员的工作量，能为学校管理提供方便。

1. 减轻教师的工作负担，保证数据的正确性

作业管理系统主要目的是解决教师批改作业和学生提交作业都得靠纸质交流，或者是用单一的上传，拷贝等方法交流不方便的问题。提供给教师一个管理和布置作业的平台，也让学生随时随地提交作业。改变了传统教育的局限，突破了时间和空间的限制，使教学更加灵活，交互性强，为师生的交流了新途径。系统具有操作简便，界面友好的特性。管理员能够对用户和权限进行方便的修改和管理

1. 学生完成作业效率

目前许多高校教师是在教室上课时布置作业，学生利用课余时间完成，教师手工批改，再把作业发给学生。这样一个流程下来，学生完成的效率不高，时间也会被耽误。本系统它大大提高了学生完成作业的效率和时间。

## 1.3开发现状分析

本系统能够满足学生完成作业学习知识的要求，同时也能方便老师及时检查学生进度以及进行新的作业布置等功能。自适应性使学生不会在学习的旅途中感觉盲目，迷失方向。大大满足了学生学习的个性化，教师工作时效性的双方面的要求。这些将会吸引大量用户对本系统的兴趣与爱好。并且此系统操作简单，在操作方面都会有说明进行提示，比如新作业的布置及其显示，完成作业的情况，学生的登录板块有专门的作业提交界面。管理权限的设置使的各自的功能实现起来更加简洁方便，不会使用户不知何处下手，只要是懂得上网和识字的用户都能够对本系统操作的游刃有余。

同时所开发的高校学生作业管理系统，它具有以下特点：

1. 实用性：方便教师布置作业、批改作业和学生随时随地提交作业。
2. 操作简便：本系统适合教师和学生使用，同时系统操作简单，而且系统不复杂繁琐。
3. 代码可读性好：文中的代码使用方便，简洁易懂。

## 1.4课题开发意义

本系统学生作业管理系统是当今技术发展的产物，通过Web的形式完成学生作业管理系统，本系统不仅方便学生提交作业，而且更为老师提供检查作业的。本系统功能简单、明了，基本符合实际开发需求，学生可以查看作业、上传作业、查看作业批改情况等。教师可以发布作业，批改作业等。由此可以得出，网络教育学习及网格各种信息交流的好与坏，严重影响着我们学习生活的各个角落，我们需要一套有效的网上学习管理系统为我们快节奏高效率的工作学习提供一种保障。同时扩大系统使用的效率和应用范围，为更多的有这种需求的教师和广大的学生提供必要和及时的服务。因此学生作业管理系统的实现开发出现了其必要性和可行性。相信随着时间的推移，学生作业管理系统会在未来的高校发展管理模式占十分重要的地位，而且还有很大的发展空间来展现时效的作用。

通过Internet来实现网上作业提交，是现代教育技术的一个具体实现，具有很重要的现实意义。网络化教育代表了教育改革的一个发展方向，已经成为现代教育的一个特征，并对教育的发展形成新的动力。可以实现学生在线提交作业的无纸张化管理，可以有效利用校园网的软硬件资源，使其发挥最大效力，更好的为学校的教学、科研、管理服务，随时随地的可以让学生联机操作提交。

# 第二章 开发技术介绍

此系统的关键技术和架构，Python技术、B/S结构、Django框架和Mysql数据库，是本系统的关键开发技术，对系统的整体、数据库、功能模块、系统页面以及系统程序等设计进行了详细的研究与规划。

## 2.1 系统开发平台

在该在线学生作业管理系统中，Eclipse能给用户提供更多的方便，其特点一是方便学习，方便快捷；二是有非常大的信息储存量，主要功能是用在对数据库中查询和编程。其功能有比较灵活的数据应用，只需利用小部分代码就能实现非常强大的功能。因此，利用Eclipse 技术进行系统代码管理是该系统数据库的首选。

## 2.2 平台开发相关技术

### 2.2.1 Python技术

Python 是一个高层次的脚本语言结合了解释性、编译性、互动性和面向对象的。Python 的设计，相比其他语言经常使用英文关键字和其他语言的一些标点符号，它具有比其他语言更有特色语法结构，具有很强的可读性。

解释型语言：类似于PHP和Perl语言，这意味着开发过程中没有了编译这个环节。

交互式语言：可以在一个 Python 提示符 >>> 后直接执行代码。

面向对象语言：Python支持面向对象的风格或代码封装在对象的编程技术。

### 2.2.2 Mysql数据库介绍

利用Mysql的数据独立性、安全性等特点，在软件项目中对数据进行操作，可以保证数据准确无误，并降低了程序员的应用开发时间。

Mysql的特点是支持多线程，能方便的对系统资源充分利用，有效提高速度，还提供多种方式途径来对数据库进行连接；Mysql的功能相对弱小、规模也小，但本系统要求不高，Mysql完全可以满足本系统使用。

利用Mysql建立系统数据库，不仅有利于数据处理业务的早期整合，还能利于发展后两种数据扩展的操作。

### 2.2.3 Mysql环境配置

本系统的数据使用的是Mysql,所以要将Mysql安装到指定目录，如果下载的是非安装的Mysql压缩包，直接解压到指定目录就可以了。然后点击C:\Program Files\Mysql\bin\winMysqladmin.exe这个文件其中C:\Program Files\Mysql是Mysql安装目录。输入winMysqladmin的初始用户、密码（注：这不是Mysql里的用户、密码）随便填不必在意，确定之后右下角任务的启动栏会出现一个红绿灯的图标，红灯亮代表服务停止，绿灯亮代表服务正常，左击这个图标->winnt->install the service 安装此服务，再左击这个图标->winnt->start the service 启动Mysql服务。

修改Mysql数据库的root密码。用cmd进入命令行模式输入如下命令:

cd C:\Program Files\Mysql\bin

Mysqladmin -u root -p password 123

回车出现Enter password: ，这是要输入原密码. 刚安装时密码为空,所以直接回车，此时Mysql 中账号 root 的密码被改为 123 安装完毕。

### 2.2.4 B/S架构

B/S结构是目前使用最多的结构模式，它可以使得系统的开发更加的简单，好操作，而且还可以对其进行维护。使用该结构时只需要在计算机中安装数据库，和一些很常用的浏览器就可以了。浏览器就会与数据库进行信息的连接，可以实现很多的功能，B/S结构是可以直接进行使用的，而且B/S结构在使用中极大的减少了工作的维护。基于B/S的软件，所有的数据库之间都是相互独立的，因此是非常安全的。因为基于B/S结构可以清楚的看到系统正在处理的业务，并且能够及时的让管理人员做出决策，这样就可以避免企业的损失。B/S结构的基本特点是集中式的管理模式，用户使用系统生成数据后，这些数据就可以存储到系统的数据库中，方便日后能够用到，这样就可以满足人们的所有的需求。



图2-1 B/S模式三层结构图

### 2.2.5 Django框架

Django是一种开源的大而且全的Web应用框架，是由python语言来编写的。他采用了MVC模式，Django最初是被开发来用于管理劳伦斯出版集团下的一些以新闻为主内容的网站。一款CMS（内容管理系统）软件。并于 2005 年 7 月在 BSD 许可证下发布。这套框架是以比利时的吉普赛爵士吉他手 Django Reinhardt 来命名的。

Django是Python语言中的一个web框架，并遵循MVC设计。Python语言中主流的web框架有Django、Tornado、Flask 等多种，Django相较与其它WEB框架，其优势为：大而全，框架本身集成了ORM、模型绑定、模板引擎、缓存、文件管理、认证权限Session等功能，是一个全能型框架，拥有自己的Admin数据管理后台，第三方工具齐全，性能折中。Django的主要目的是简便、快速的开发数据库驱动的网站。

# 第三章 系统分析

## 3.1 可行性分析

一个完整的系统，可行性分析是必须要有的，因为他关系到系统生存问题，对开发的意义进行分析，能否通过本系统来补充线下资产管理模式中的缺限，去解决其中的不足等，通过对本系统，不仅能使工作量不断地减少，还能使工作和管理的效率更加高。所以开发该系统能实现更大的意义和价值， 系统完成后，能否达到预期效果就要通过可行性分析，分析之后，决定此系统是否开发。该学生作业管理系统的开发设计中，对技术、经济、操作方面进行了可行性分析。

### 3.1.1 技术可行性

本系统开发选择Python语言，它被研究的目的就是在于能够为网页创建等可以看到的信息。随着移动互联网技术的不断发展和创新，Python俨然已成为下一代互联网的Web标准。所以后台设计选择使用Mysql,数据库主要用来的建立和维护信息。对于前台开发要求应具备功能完善、易于操作等优点，后台数据库的要求则是能够建立和维护数据信息的统一性和完整性。

### 3.1.2 操作可行性

现在随着科技的飞速发展，计算机早已经进入了人们的日常生活中，人们的工作环境也不像以前有那么多的要求，需要员工一定要到公司办公，有的工作在家也可以完成。这使得人们的工作效益有了很大的提高。操作的多样性也变高了。因此，管理的计算机化，智能化是社会发展而带来的必然趋势，各种智能的软件层出不穷，不同的软件能完成用户不同的需求，这不仅提高了工作效率还能完成一些客户特定的一些需求。本系统不仅界面简洁明了还采用可视化界面，用户只要用鼠标和键盘就可以完成对相关信息的修改，删除，添加等操作。因为这个系统的操作十分简单，方便上手，对于第一次使用系统的人，只需要很少的时间就可以上手操作。由此可见，本系统在操作上是可行的。

### 3.1.3经济可行性

基于Django框架的学生作业管理系统，该系统软件开发仅需要一台普通的计算机便可完成实现开发，其成本很低。另外，作为毕业设计作品来讲，开发成本基本上可以忽略不计，且该系统软件的投入使用，可以实现更加快速高效的资产管理，同时还能实现对人力资源和管理资源的有效节约，该学生作业管理系统在经济上完全可行。

## 3.2性能需求分析

对系统的性能，从（功能、运行、界面、安全）等方面进行，下面我们逐一进行分析；

1. 系统的功能是否完整进行分析：系统的功能，能对应设计出原始代码和算法，以表格同文字的形式进行详细介绍个人信息保证功能完整；

2. 系统的运行是否通畅进行分析：系统的每个功能都有编写数据的关系和应对的代码，通过需求分析和可行性分析进行分析和显示系统的物理数据，保证其进行通畅；

3. 系统的界面设计进行分析：对系统中的软件进行处理与分析的方式是由不同代码来进行的；从而使界面容易操作。

4. 系统的安全性进行分析：这样才可以每个角色的不同对应的信息也就不同，在登录系统务必使用自己的账号，密码登录，账号与密码错误自然就登录失败了。登录成功可以对自己的信息进行操作，不能对别人的账号的信息进行查看等操作，这样自然保证系统的安全性。

# 第四章 系统设计

## 4.1功能结构

基于python的中小学生家庭作业管理系统含有三个角色，包含：学生、教师、管理员；学生主要包含：作业布置管理、作业提交管理、作业批改管理以及个人信息管理；教师角色主要包含：课程信息管理、作业布置管理、作业提交管理、作业批改管理以及个人信息管理等； 管理员角色包含：学院管理、专业管理、班级管理、学生管理、教师管理、课程信息管理、作业布置管理、作业提交管理、作业批改管理等；为了更好的去理清本系统整体思路，对该系统以结构图的形式表达出来，设计实现该学生作业管理系统的功能结构图如下所示：



图4-1 系统总体结构图

## 4.2 数据库设计

### 4.2.1 数据库E/R图

ER图是由实体及其关系构成的图，通过E/R图可以清楚地描述系统涉及到的实体之间的相互关系。在系统中对一些主要的几个关键实体如下图：

(1) 学生管理E/R图如下所示：



图4-2学生管理E/R图

(2) 课程信息管理E/R图如下所示：

图4-3课程信息管理E/R图

(3) 教师信息E/R图如下所示：



图4-4教师信息E/R图

1. 作业布置E/R图如下所示：

 图4-5作业布置E/R图

### 4.2.2 数据库表

数据库表的设计，如下表：

表4-1：作业布置

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| kechengmingcheng | varchar | 200 | 课程名称 |  |  |
| kechengleixing | varchar | 200 | 课程类型 |  |  |
| nianji | varchar | 200 | 年级 |  |  |
| zhuanye | varchar | 200 | 专业 |  |  |
| xueqi | varchar | 200 | 学期 |  |  |
| zuoyeyaoqiu | varchar | 200 | 作业要求 |  |  |
| faburiqi | date |  | 发布日期 |  |  |
| jiaoshigonghao | varchar | 200 | 教师工号 |  |  |
| jiaoshixingming | varchar | 200 | 教师姓名 |  |  |

表4-2：专业信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| zhuanye | varchar | 200 | 专业 |  |  |

表4-3：学院表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| xueyuan | varchar | 200 | 学院 |  |  |

表4-4：学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| xuehao | varchar | 200 | 学号 |  |  |
| mima | varchar | 200 | 密码 |  |  |
| xueshengxingming | varchar | 200 | 学生姓名 |  |  |
| xingbie | varchar | 200 | 性别 |  |  |
| touxiang | varchar | 200 | 头像 |  |  |
| xueyuan | varchar | 200 | 学院 |  |  |
| zhuanye | varchar | 200 | 专业 |  |  |
| banji | varchar | 200 | 班级 |  |  |
| shouji | varchar | 200 | 手机 |  |  |
| youxiang | varchar | 200 | 邮箱 |  |  |

表4-5：课程信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| kechengmingcheng | varchar | 200 | 课程名称 |  |  |
| kechengleixing | varchar | 200 | 课程类型 |  |  |
| tupian | varchar | 200 | 图片 |  |  |
| nianji | varchar | 200 | 年级 |  |  |
| zhuanye | varchar | 200 | 专业 |  |  |
| xueqi | varchar | 200 | 学期 |  |  |
| faburiqi | date |  | 发布日期 |  |  |
| jiaoshigonghao | varchar | 200 | 教师工号 |  |  |
| jiaoshixingming | varchar | 200 | 教师姓名 |  |  |

表4-6：教师信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| jiaoshigonghao | varchar | 200 | 教师工号 |  |  |
| mima | varchar | 200 | 密码 |  |  |
| jiaoshixingming | varchar | 200 | 教师姓名 |  |  |
| xingbie | varchar | 200 | 性别 |  |  |
| zhaopian | varchar | 200 | 照片 |  |  |
| zhicheng | varchar | 200 | 职称 |  |  |
| lianxidianhua | varchar | 200 | 联系电话 |  |  |
| jiaoshiyouxiang | varchar | 200 | 教师邮箱 |  |  |

表4-7：班级信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| banji | varchar | 200 | 班级 |  |  |

表4-8：作业提交表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT\_TIMESTAMP |
| kechengmingcheng | varchar | 200 | 课程名称 |  |  |
| kechengleixing | varchar | 200 | 课程类型 |  |  |
| nianji | varchar | 200 | 年级 |  |  |
| zhuanye | varchar | 200 | 专业 |  |  |
| xueqi | varchar | 200 | 学期 |  |  |
| zuoye | varchar | 200 | 作业 |  |  |
| tijiaoriqi | date |  | 提交日期 |  |  |
| jiaoshigonghao | varchar | 200 | 教师工号 |  |  |
| jiaoshixingming | varchar | 200 | 教师姓名 |  |  |
| xuehao | varchar | 200 | 学号 |  |  |
| xueshengxingming | varchar | 200 | 学生姓名 |  |  |
| banji | varchar | 200 | 班级 |  |  |
| sfsh | varchar | 200 | 是否审核 |  | 否 |
| shhf | longtext | 4294967295 | 审核回复 |  |  |

表4-9：作业批改表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  |  |
| kechengmingcheng | varchar | 200 | 课程名称 |  |  |
| kechengleixing | varchar | 200 | 课程类型 |  |  |
| zhuanye | varchar | 200 | 专业 |  |  |
| xueqi | varchar | 200 | 学期 |  |  |
| pingfen | int |  | 评分 |  |  |
| pigaiyijian | longtext | 4294967295 | 批改意见 |  |  |
| dengjiriqi | date |  | 登记日期 |  |  |
| jiaoshigonghao | varchar | 200 | 教师工号 |  |  |
| jiaoshixingming | varchar | 200 | 教师姓名 |  |  |
| xuehao | varchar | 200 | 学号 |  |  |
| xueshengxingming | varchar | 200 | 学生姓名 |  |  |
| banji | varchar | 200 | 班级 |  |  |
| crossuserid | bigint |  | 跨表用户id |  |  |
| crossrefid | bigint |  | 跨表主键id |  |  |

# 第五章 系统功能实现

## 5.1管理员功能模块

**1）**管理员登录，管理员通过输入用户名、密码，选择角色并点击登录进行系统登录操作，如图5-1所示。



图5-1管理员登录界面图

|  |
| --- |
| 代码1 登录界面 |
|  |
| def users\_login(request):  if request.method in ["POST", "GET"]:  msg = {'code': normal\_code, "msg": mes.normal\_code}  req\_dict = request.session.get("req\_dict")  if req\_dict.get('role')!=None:  del req\_dict['role']  datas = users.getbyparams(users, users, req\_dict)  if not datas:  msg['code'] = password\_error\_code  msg['msg'] = mes.password\_error\_code  return JsonResponse(msg)  req\_dict['id'] = datas[0].get('id')  return Auth.authenticate(Auth, users, req\_dict) |
|  |

**2）**管理员登录系统后，可以对个人中心、学院管理、专业管理、班级管理、学生管理、教师管理、课程信息管理、作业布置管理、作业提交管理、作业批改管理等功能进行相应操作，如图5-2所示。



图5-2管理员功能界图面

|  |
| --- |
| 代码2 管理员界面 |
|  |
| def users\_session(request):  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code,"msg":mes.normal\_code, "data": {}}  req\_dict = {"id": request.session.get('params').get("id")}  msg['data'] = users.getbyparams(users, users, req\_dict)[0]  return JsonResponse(msg) |
|  |

**3）**学生管理，在学生管理页面可以对索引、学号、学生姓名、性别、头像、学院、专业、班级、手机、邮箱等进行详情、修改、删除或新增等操作，学生管理页面如图5-3所示。

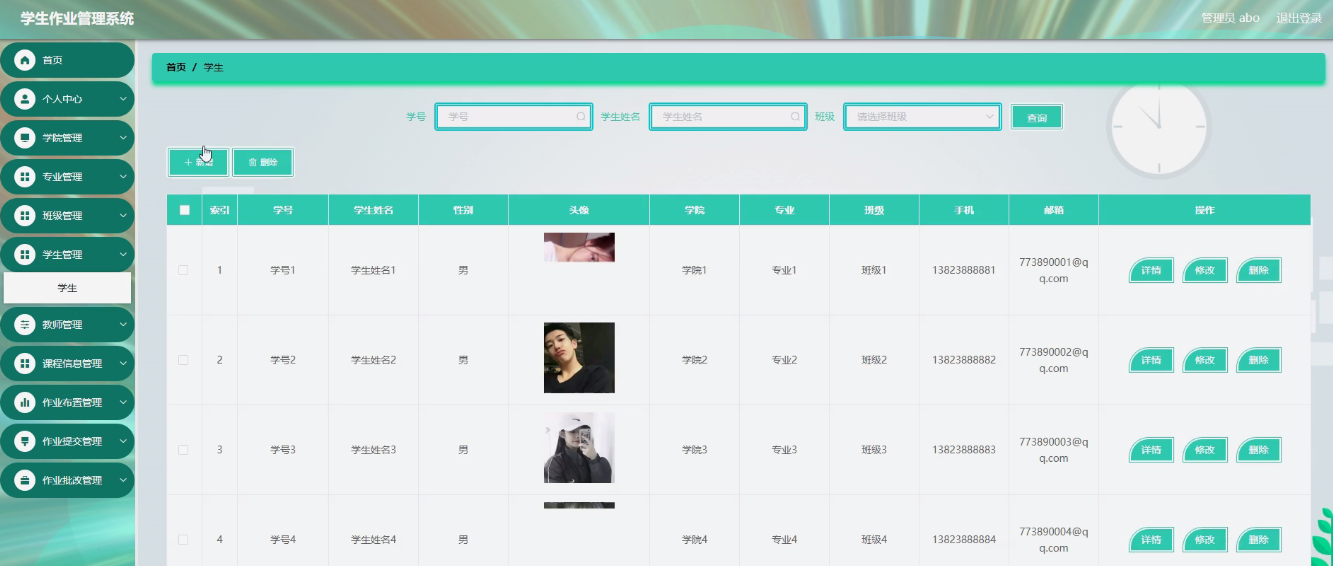


图5-3学生管理界面图

|  |
| --- |
| 代码3 学生管理界面 |
|  |
| def xuesheng\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  #获取全部列名  columns= xuesheng.getallcolumn( xuesheng, xuesheng)  #当前登录用户所在表  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=xuesheng.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  #当项目属性hasMessage为”是”，生成系统自动生成留言板的表messages，同时该表的表属性hasMessage也被设置为”是”,字段包括userid（用户id），username(用户名)，content（留言内容），reply（回复）  #接口page需要区分权限，普通用户查看自己的留言和回复记录，管理员查看所有的留言和回复记录  try:  \_\_hasMessage\_\_=xuesheng.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  # 判断当前表的表属性isAdmin,为真则是管理员表  # 当表属性isAdmin=”是”,刷出来的用户表也是管理员，即page和list可以查看所有人的考试记录(同时应用于其他表)  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  # 当前表也是有管理员权限的表  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  #非管理员权限的表,判断当前表字段名是否有userid  if tablename!="users" and 'xuesheng'[:7]!='discuss'and "userid" in xuesheng.getallcolumn(xuesheng,xuesheng):  req\_dict["userid"] = request.session.get("params").get("id")  #当列属性authTable有值(某个用户表)[该列的列名必须和该用户表的登陆字段名一致]，则对应的表有个隐藏属性authTable为”是”，那么该用户查看该表信息时，只能查看自己的  try:  \_\_authTables\_\_=xuesheng.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =xuesheng.page(xuesheng, xuesheng, req\_dict)  return JsonResponse(msg) |
|  |

**4）**教师管理，在教师管理页面可以对索引、教师工号、教师姓名、性别、照片、职称、联系电话、教师邮箱等内容进行详情、修改、删除、新增等操作，教师管理页面如图5-4所示。

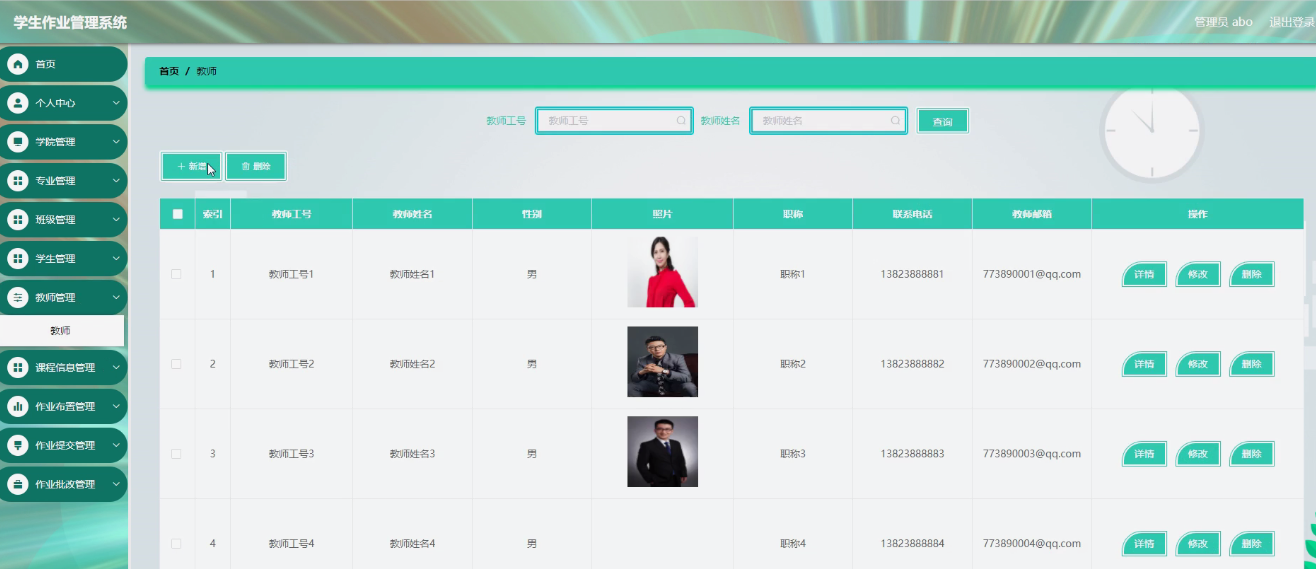


图5-4教师管理界面图

|  |
| --- |
| 代码4 教师管理界面 |
|  |
| def jiaoshi\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  columns= jiaoshi.getallcolumn( jiaoshi, jiaoshi)  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=jiaoshi.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  try:  \_\_hasMessage\_\_=jiaoshi.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  if tablename!="users" and 'jiaoshi'[:7]!='discuss'and "userid" in jiaoshi.getallcolumn(jiaoshi,jiaoshi):  req\_dict["userid"] = request.session.get("params").get("id")  try:  \_\_authTables\_\_=jiaoshi.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =jiaoshi.page(jiaoshi, jiaoshi, req\_dict)  return JsonResponse(msg) |
|  |

**5）**学院管理，在学院管理页面可以对索引、学院名称等内容进行详情、修改、删除或新增等操作，学院管理页面如图5-5所示。

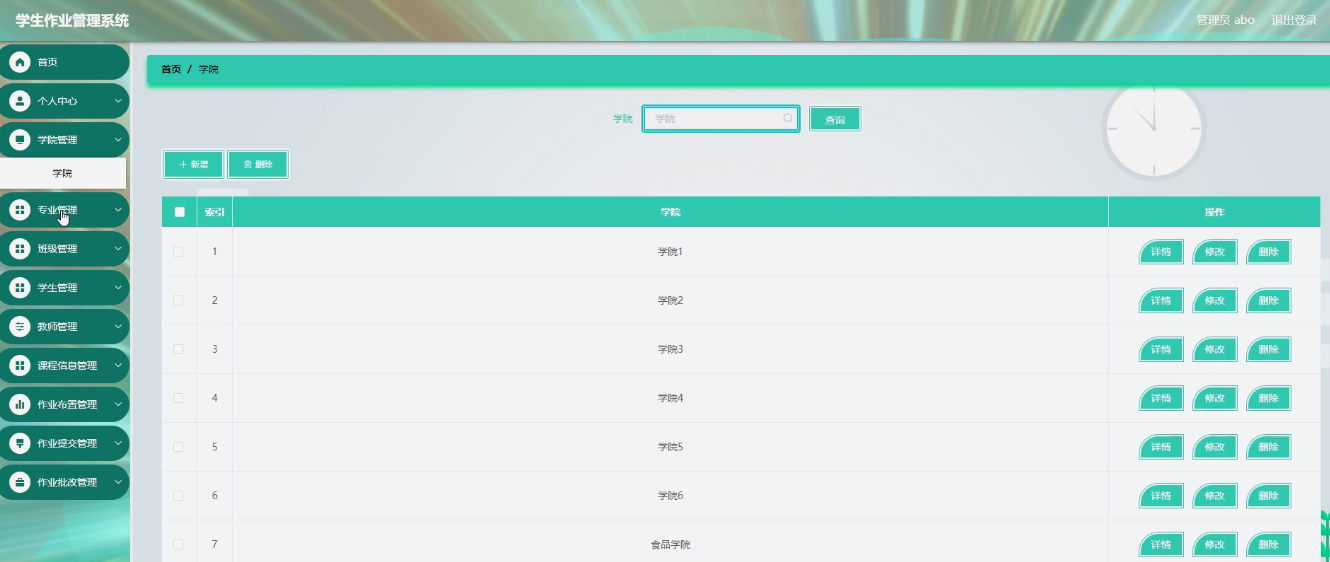


图5-5学院管理界面图

|  |
| --- |
| 代码5 学院管理界面 |
|  |
| def xueyuan\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  columns= xueyuan.getallcolumn( xueyuan, xueyuan)  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=xueyuan.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  try:  \_\_hasMessage\_\_=xueyuan.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  if tablename!="users" and 'xueyuan'[:7]!='discuss'and "userid" in xueyuan.getallcolumn(xueyuan,xueyuan):  req\_dict["userid"] = request.session.get("params").get("id")  try:  \_\_authTables\_\_=xueyuan.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =xueyuan.page(xueyuan, xueyuan, req\_dict)  return JsonResponse(msg) |
|  |

**6）**专业管理，在专业管理页面可以对索引、专业等内容进行详情、修改、删除或新增等操作，专业管理页面如图5-6所示。

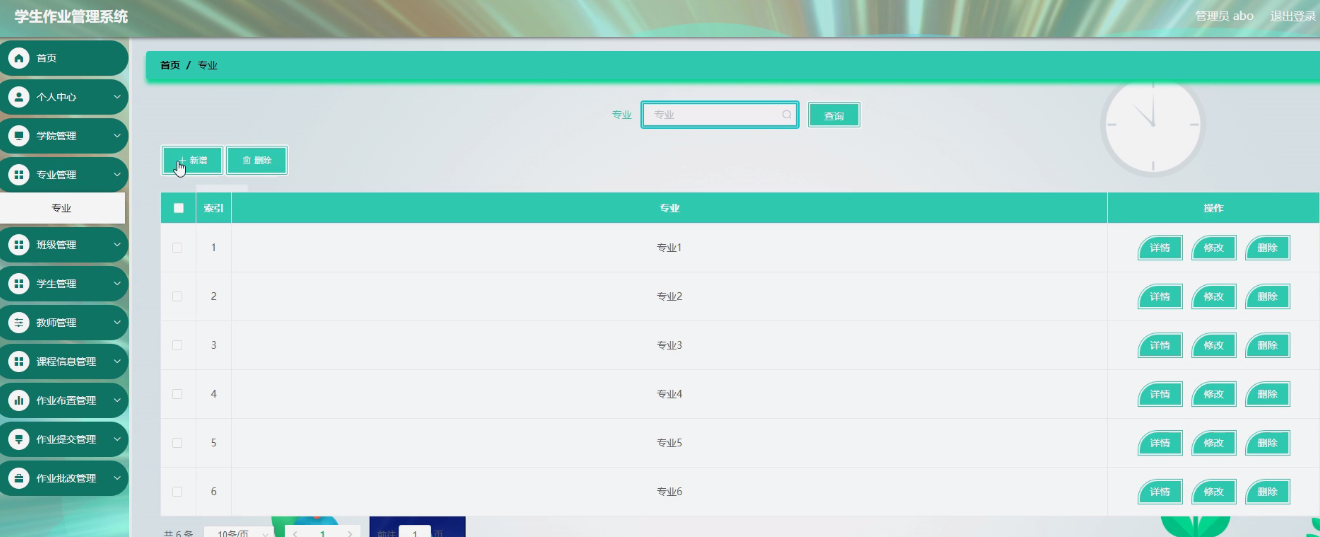


图5-6专业管理界面图

|  |
| --- |
| 代码6 专业管理界面 |
|  |
| def zhuanye\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  columns= zhuanye.getallcolumn( zhuanye, zhuanye)  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=zhuanye.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  try:  \_\_hasMessage\_\_=zhuanye.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  if tablename!="users" and 'zhuanye'[:7]!='discuss'and "userid" in zhuanye.getallcolumn(zhuanye,zhuanye):  req\_dict["userid"] = request.session.get("params").get("id")  try:  \_\_authTables\_\_=zhuanye.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =zhuanye.page(zhuanye, zhuanye, req\_dict)  return JsonResponse(msg) |
|  |

**7）**课程信息管理，在课程信息管理页面可以对索引、课程名称、课程类型、图片、年级、专业、学期、发布日期、教师工号、教师姓名等内容进行详情、修改、删除、或新增等操作，课程信息操作页面如图5-7所示。

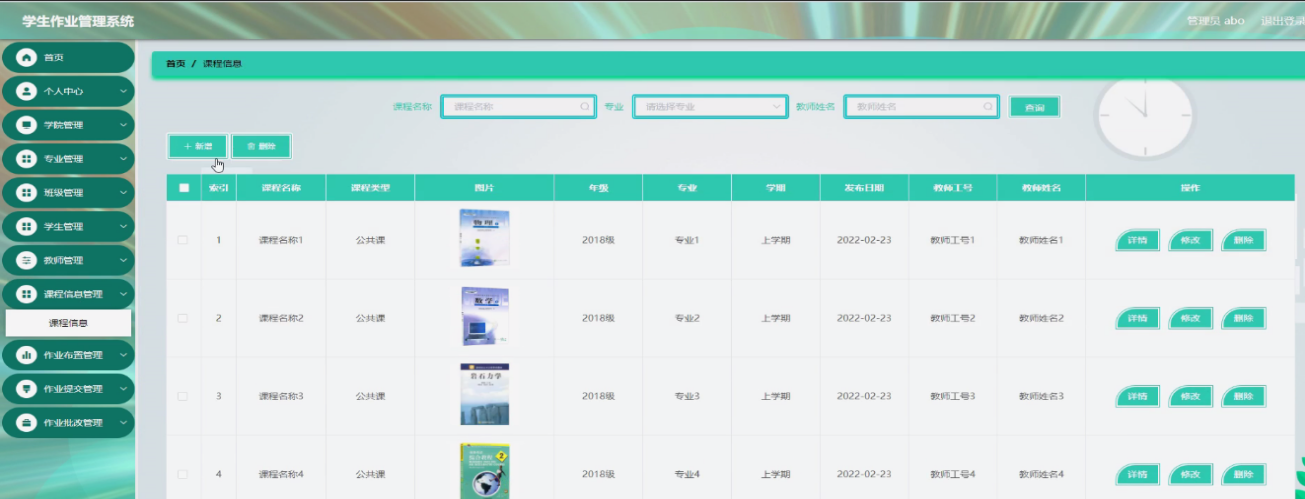


图5-7课程信息管理界面图

|  |
| --- |
| 代码7 课程信息界面 |
|  |
| def kechengxinxi\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  columns= kechengxinxi.getallcolumn( kechengxinxi, kechengxinxi)  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=kechengxinxi.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  try:  \_\_hasMessage\_\_=kechengxinxi.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  if tablename!="users" and 'kechengxinxi'[:7]!='discuss'and "userid" in kechengxinxi.getallcolumn(kechengxinxi,kechengxinxi):  req\_dict["userid"] = request.session.get("params").get("id")  try:  \_\_authTables\_\_=kechengxinxi.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =kechengxinxi.page(kechengxinxi, kechengxinxi, req\_dict)  return JsonResponse(msg) |
|  |

## 

## 5.2教师功能模块

**1）**教师登录，教师通过输入用户名、密码、选择登录角色进行登录到学生作业管理系统后台，教师登录如图5-8所示。



图5-8教师登录界面图

|  |
| --- |
| 代码8 教师登录界面 |
|  |
| def users\_login(request):  if request.method in ["POST", "GET"]:  msg = {'code': normal\_code, "msg": mes.normal\_code}  req\_dict = request.session.get("req\_dict")  if req\_dict.get('role')!=None:  del req\_dict['role']  datas = users.getbyparams(users, users, req\_dict)  if not datas:  msg['code'] = password\_error\_code  msg['msg'] = mes.password\_error\_code  return JsonResponse(msg)  req\_dict['id'] = datas[0].get('id')  return Auth.authenticate(Auth, users, req\_dict) |
|  |

**2）**教师登录到学生作业管理系统后台后，可以对个人中心、课程信息管理、作业布置管理、作业提交管理、作业批改管理等功能进行相应操作，如图5-9所示。

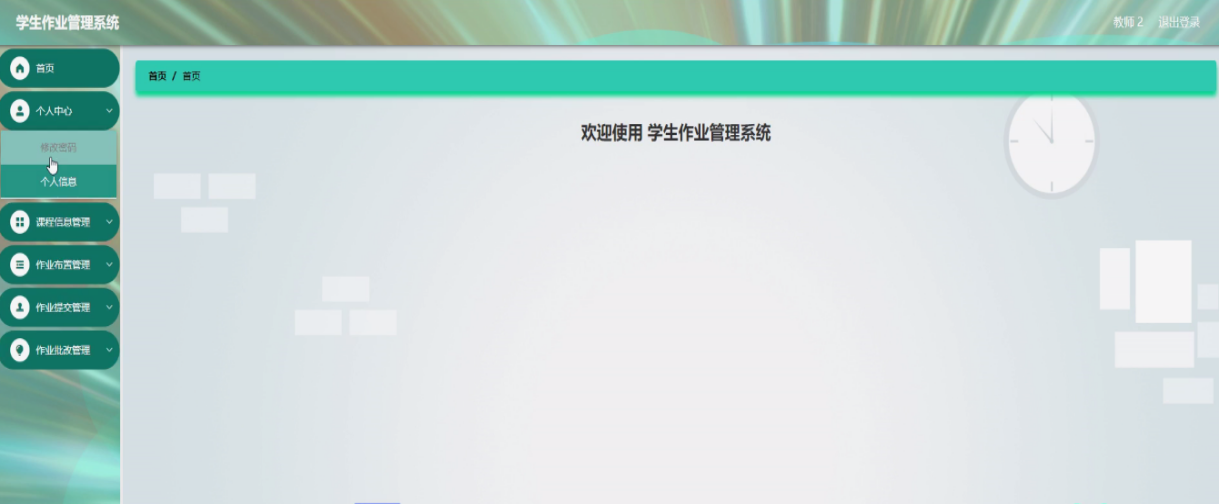


图5-9教师功能界面图

|  |
| --- |
| 代码9 教师功能界面 |
|  |
| def users\_session(request):  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code,"msg":mes.normal\_code, "data": {}}  req\_dict = {"id": request.session.get('params').get("id")}  msg['data'] = users.getbyparams(users, users, req\_dict)[0]  return JsonResponse(msg) |
|  |

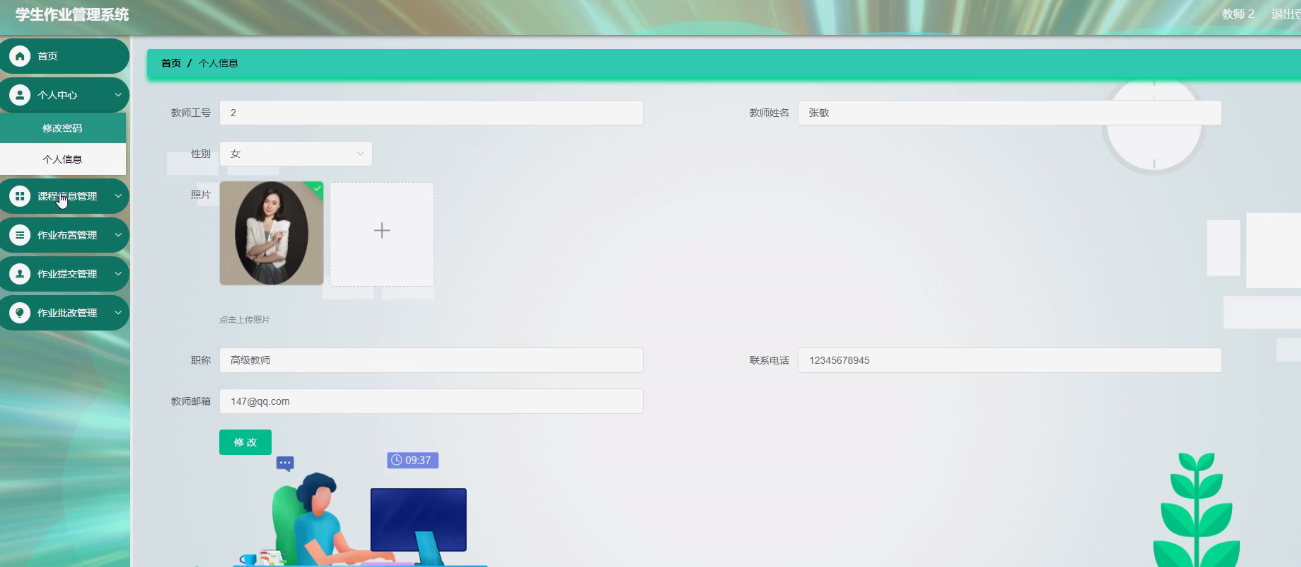
**3）**个人中心，在个人中心页面，教师通过输入教师工号、教师姓名、性别、照片、职称、联系电话、教师邮箱等内容来修改个人信息，也能修改登录密码，个人中心页面如图5-10所示。

图5-10个人中心界面图

|  |
| --- |
| 代码10 个人中心界面 |
|  |
| def users\_session(request):  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code,"msg":mes.normal\_code, "data": {}}  req\_dict = {"id": request.session.get('params').get("id")}  msg['data'] = users.getbyparams(users, users, req\_dict)[0]  return JsonResponse(msg) |
|  |

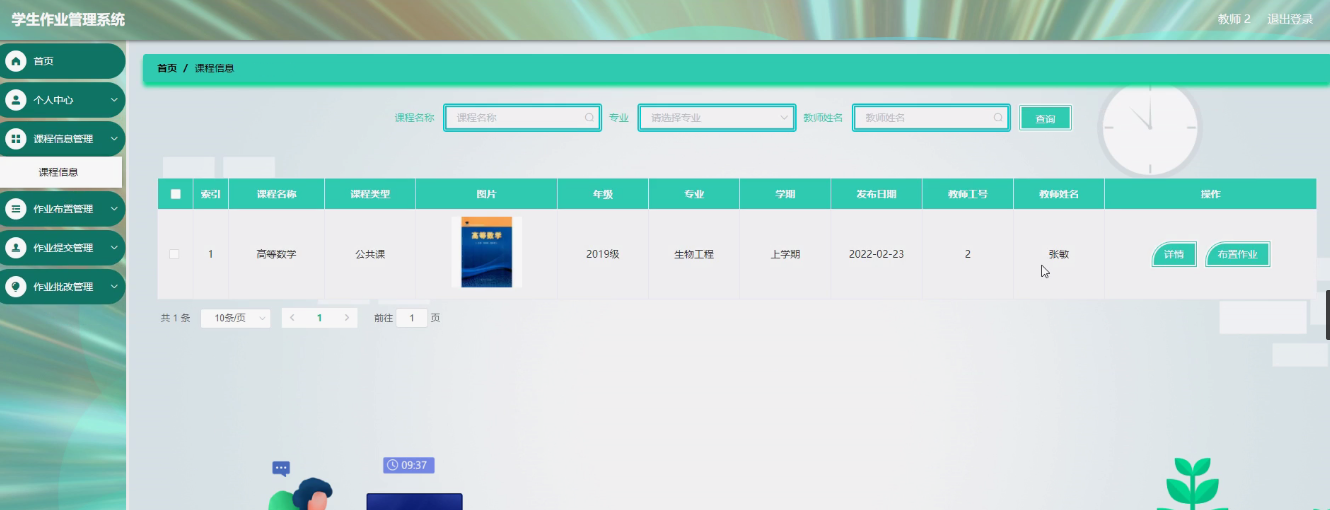
**4）**点击课程信息管理，教师可以对索引、课程名称、课程类型、图片、年级、专业、学期、发布日期、教师工号、教师姓名等内容进行详情或布置作业等操作，课程信息管理页面如图5-11所示。

图5-11课程信息管理界面图

|  |
| --- |
| 代码11 课程信息管理界面 |
|  |
| def kechengxinxi\_list(request):  '''  前台分页  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  #获取全部列名  columns= kechengxinxi.getallcolumn( kechengxinxi, kechengxinxi)  #表属性[foreEndList]前台list:和后台默认的list列表页相似,只是摆在前台,否:指没有此页,是:表示有此页(不需要登陆即可查看),前要登:表示有此页且需要登陆后才能查看  try:  \_\_foreEndList\_\_=kechengxinxi.\_\_foreEndList\_\_  except:  \_\_foreEndList\_\_=None  if \_\_foreEndList\_\_=="前要登":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  #forrEndListAuth  try:  \_\_foreEndListAuth\_\_=kechengxinxi.\_\_foreEndListAuth\_\_  except:  \_\_foreEndListAuth\_\_=None  #authSeparate  try:  \_\_authSeparate\_\_=kechengxinxi.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_foreEndListAuth\_\_ =="是" and \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict['userid']=request.session.get("params",{"id":0}).get("id")  tablename = request.session.get("tablename")  if tablename == "users" and req\_dict.get("userid") != None:#判断是否存在userid列名  del req\_dict["userid"]  else:  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  #非管理员权限的表,判断当前表字段名是否有userid  if "userid" in columns:  try:  # 本接口可以匿名访问,所以try判断是否为匿名  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  #当列属性authTable有值(某个用户表)[该列的列名必须和该用户表的登陆字段名一致]，则对应的表有个隐藏属性authTable为”是”，那么该用户查看该表信息时，只能查看自己的  try:  \_\_authTables\_\_=kechengxinxi.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={} and \_\_foreEndListAuth\_\_=="是":  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  if kechengxinxi.\_\_tablename\_\_[:7]=="discuss":  try:  del req\_dict['userid']  except:  pass  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] = kechengxinxi.page(kechengxinxi, kechengxinxi, req\_dict)  return JsonResponse(msg) |
|  |

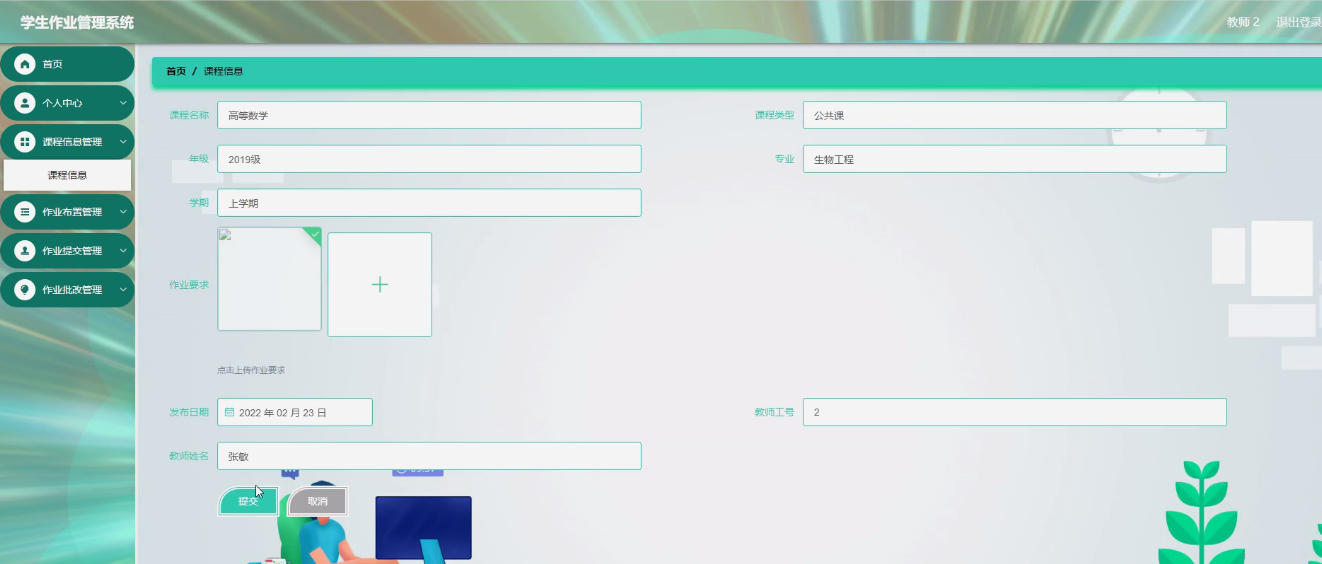
**5）**在课程信息管理页面，点击布置作业，教师通过输入课程名称、课程类型、年级、专业、学期、作业要求、发布日期、教师工号、教师姓名等内容，点击提交进行作业布置，作业布置新增如图5-12所示。

图5-12布置作业界面图

|  |
| --- |
| 代码12 布置作业界面 |
|  |
| def zuoyebuzhi\_page(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data":{"currPage":1,"totalPage":1,"total":1,"pageSize":10,"list":[]}}  req\_dict = request.session.get("req\_dict")  columns= zuoyebuzhi.getallcolumn( zuoyebuzhi, zuoyebuzhi)  tablename = request.session.get("tablename")  #authColumn=list(\_\_authTables\_\_.keys())[0]  #authTable=\_\_authTables\_\_.get(authColumn)  # if authTable==tablename:  #params = request.session.get("params")  #req\_dict[authColumn]=params.get(authColumn)  '''\_\_authSeparate\_\_此属性为真，params添加userid，后台只查询个人数据'''  try:  \_\_authSeparate\_\_=zuoyebuzhi.\_\_authSeparate\_\_  except:  \_\_authSeparate\_\_=None  if \_\_authSeparate\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users" and 'userid' in columns:  try:  req\_dict['userid']=request.session.get("params").get("id")  except:  pass  try:  \_\_hasMessage\_\_=zuoyebuzhi.\_\_hasMessage\_\_  except:  \_\_hasMessage\_\_=None  if \_\_hasMessage\_\_=="是":  tablename=request.session.get("tablename")  if tablename!="users":  req\_dict["userid"]=request.session.get("params").get("id")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  if \_\_isAdmin\_\_ == "是":  if req\_dict.get("userid"):  del req\_dict["userid"]  else:  if tablename!="users" and 'zuoyebuzhi'[:7]!='discuss'and "userid" in zuoyebuzhi.getallcolumn(zuoyebuzhi,zuoyebuzhi):  req\_dict["userid"] = request.session.get("params").get("id")  try:  \_\_authTables\_\_=zuoyebuzhi.\_\_authTables\_\_  except:  \_\_authTables\_\_=None  if \_\_authTables\_\_!=None and \_\_authTables\_\_!={}:  try:  del req\_dict['userid']  except:  pass  for authColumn,authTable in \_\_authTables\_\_.items():  if authTable==tablename:  params = request.session.get("params")  req\_dict[authColumn]=params.get(authColumn)  break  msg['data']['list'], msg['data']['currPage'], msg['data']['totalPage'], msg['data']['total'], \  msg['data']['pageSize'] =zuoyebuzhi.page(zuoyebuzhi, zuoyebuzhi, req\_dict)  return JsonResponse(msg) |
|  |

1. 点击作业提交管理，教师可以查看课程名称、课程类型、年级、专业、学期、作业、提交日期、教师工号、教师姓名、学号、学生姓名、班级、审核回复、审核状态等内容，并进行审核、批改等操作，作业提交管理页面如图5-13所示。



图5-13作业提交管理界面图

|  |
| --- |
| 代码13 作业提交界面 |
|  |
| def zuoyebuzhi\_save(request):  '''  后台新增  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data": {}}  req\_dict = request.session.get("req\_dict")  tablename=request.session.get("tablename")  \_\_isAdmin\_\_ = None  allModels = apps.get\_app\_config('main').get\_models()  for m in allModels:  if m.\_\_tablename\_\_==tablename:  try:  \_\_isAdmin\_\_ = m.\_\_isAdmin\_\_  except:  \_\_isAdmin\_\_ = None  break  columns= zuoyebuzhi.getallcolumn( zuoyebuzhi, zuoyebuzhi)  if tablename!='users' and req\_dict.get("userid")!=None and 'userid' in columns and \_\_isAdmin\_\_!='是':  params=request.session.get("params")  req\_dict['userid']=params.get('id')  error= zuoyebuzhi.createbyreq(zuoyebuzhi,zuoyebuzhi, req\_dict)  if error!=None:  msg['code'] = crud\_error\_code  msg['msg'] = error  return JsonResponse(msg) |
|  |

**7）**在作业提交管理页面，点击批改，通过输入课程名称、课程类型、专业、学期、评分、登记日期、教师工号、教师姓名、学号、学生姓名、班级批改意见等内容，点击提交进行作业批改，如图5-14所示。

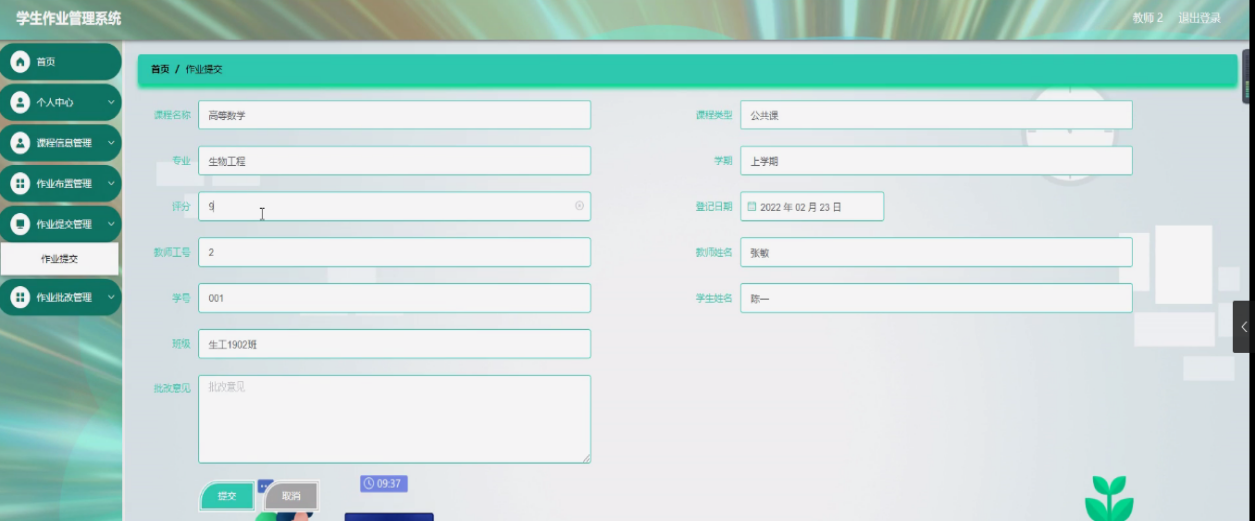


图5-14作业批改界面图

|  |
| --- |
| 代码14 作业批改界面 |
|  |
| def zuoyebuzhi\_update(request):  '''  '''  if request.method in ["POST", "GET"]:  msg = {"code": normal\_code, "msg": mes.normal\_code, "data": {}}  req\_dict = request.session.get("req\_dict")  if req\_dict.get("mima") and req\_dict.get("password"):  if "mima" not in zuoyebuzhi.getallcolumn(zuoyebuzhi,zuoyebuzhi) :  del req\_dict["mima"]  if "password" not in zuoyebuzhi.getallcolumn(zuoyebuzhi,zuoyebuzhi) :  del req\_dict["password"]  try:  del req\_dict["clicknum"]  except:  pass  error = zuoyebuzhi.updatebyparams(zuoyebuzhi, zuoyebuzhi, req\_dict)  if error!=None:  msg['code'] = crud\_error\_code  msg['msg'] = error  return JsonResponse(msg) |
|  |

实验总结

在本次实验中，我们设计并实现了一个基于Python的中小学生在线家庭作业系统。这个项目不仅帮助我们深化了对Python编程语言的理解，也让我们熟悉了Web开发框架Django以及数据库管理工具MySQL的实际应用。

在整个开发过程中，我们遇到了一些困难的部分。例如，在设计数据库模型时需要考虑不同角色（管理员、教师、学生）之间的权限管理和数据隔离问题。对此，我们通过精心设计数据库结构，并结合Django提供的认证系统，成功实现了各角色间的数据安全隔离。此外，对于复杂的业务逻辑处理，如作业的提交与批改流程，我们采取了模块化的设计思路，确保每个功能模块都能独立运行，同时也易于后续的功能扩展。

通过这次项目实践，我不仅掌握了Python Web开发的基本技能，还学会了如何将理论知识应用于实际问题的解决之中。更重要的是，这次经历让我深刻体会到团队合作的重要性。从需求分析到系统上线，每一个环节都离不开团队成员间的密切配合。未来，我希望能够继续深入学习相关技术，探索更多创新性的教育技术解决方案，为推动教育信息化贡献自己的力量。

基于当前中小学生在线家庭作业系统的设计与实现，未来可以从以下几个方向进行改进和优化：

一、增强用户体验：

·提升系统的响应速度和界面友好性，使用户操作更加流畅。

·引入个性化推荐功能，根据学生的学习情况自动推荐适合的练习题目或学习资源。

二、增加互动性和社交元素：

·在系统中加入讨论区或者问答平台，促进师生之间、学生之间的交流互动。

·实现小组作业功能，支持团队协作完成任务，并提供相应的评价体系。

三、完善数据分析与反馈机制：

·利用大数据分析技术对学生的学习行为进行深度挖掘，为教师提供更科学的教学建议。

·增强对作业完成情况的数据统计和可视化展示，帮助家长更好地了解孩子的学习进度。

参考文献

[1]张海藩 网站管理系统的分析与设计[D].厦门大学,2016

[2]丁宝康　信息管理系统[D].电子科技大学,2018

[3]孙印杰 信息管理系统的设计与实现[D].电子科技大学,2017

[4]马宁 基于Python的网站管理系统分析与设计[D].云南大学,2017.

[5]求是科技 Python信息管理系统开发实例导航[M]．人民邮店出版社．2015

[6]杨宗志 Python入门与实作[M]．科学出版社，2018

[7]曹锰　舒新峰 C#与Python程序设计[M]．西安交通大学出版社，2017

[8]虞益诚 SQL Server 2000[M]．中国铁道出版社，2019[9]董征宇 我国中小企业电子商务盈利策略探析[J].中国商贸,2018,22:94-95

[10]张爱军 电子商务技术的创新发展趋势[J].电脑知识与技术,2016,26:61-67

[11]马桂林 中小音乐企业经销商电子商务管理系统的设计[J].价值工程,2018,23:148-149

[12]蔡翔宇 电子商务与计算机网络应用[J].计算机应用研究,2017,2:366

[13]西尔伯沙茨(Silberschatz.A.) 计算机科学丛书：数据库系统概念(原书第6版)[M]. 机械工业出版社,2017,03．.

[14]萨师煊 王珊 数据库系统概论[M]．北京:高等教育出版社，2017：10-180．

[15]陈刚 Eclipse从入门到精通[M]．(第2版)．北京:清华大学出版社，2018：17-380．

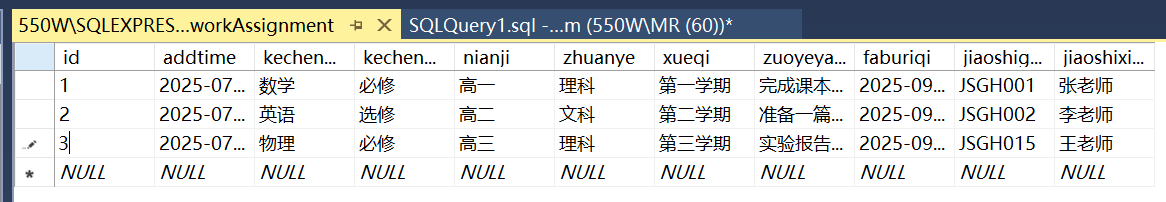
## 注：实验种提及的代码及其源码已上传至GitHub提供开源学习

地址：**https://github.com/ivy0819888/django\_web**

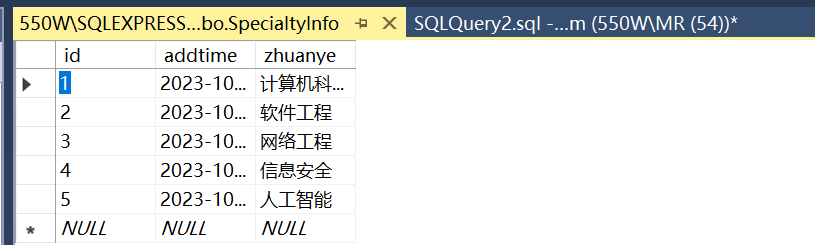
附录

一、根据第四章内容通过SQL创建的数据表示例：

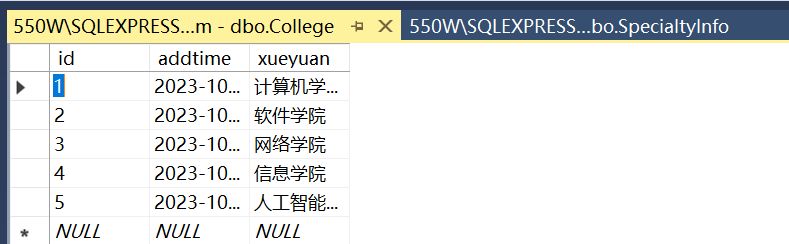
①作业布置



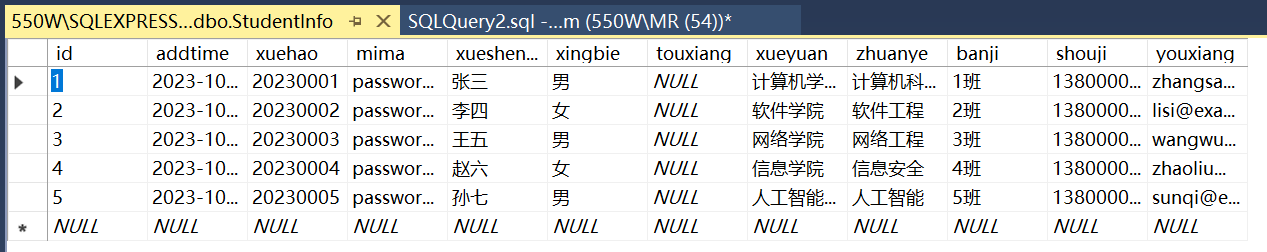
②专业信息



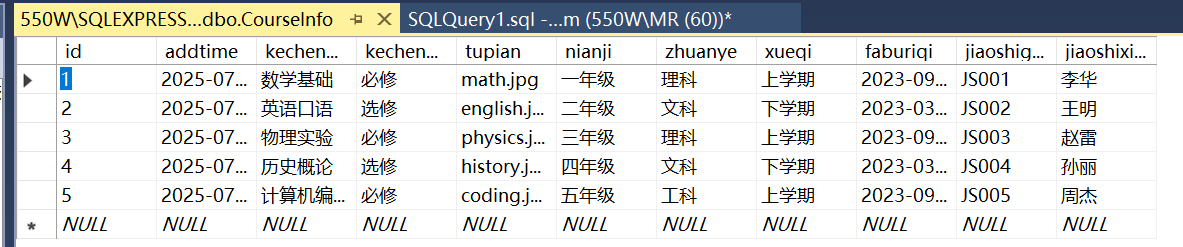
③学院信息



④学生信息表



⑤课程信息表



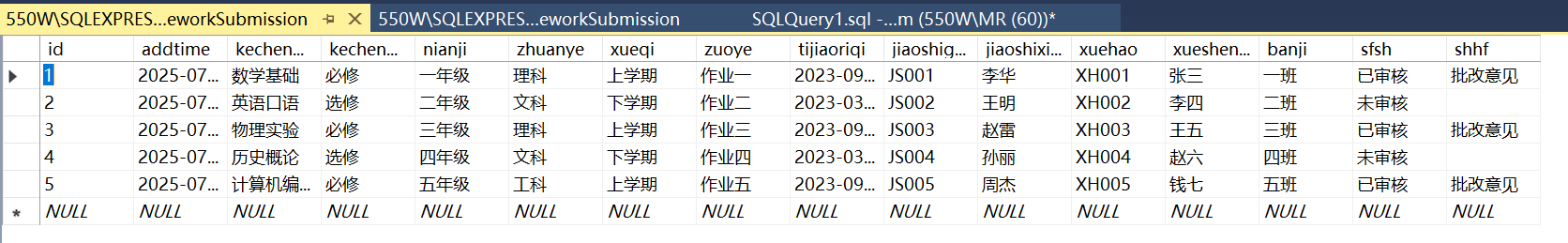
⑥教师信息表



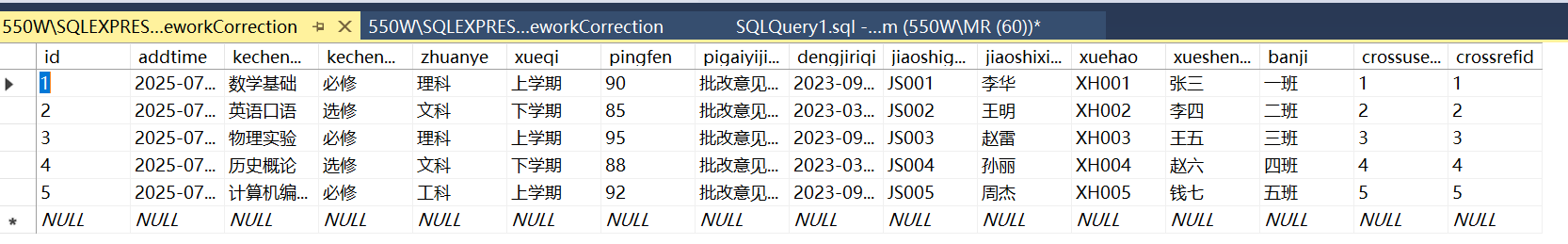
⑦班级信息表



⑧作业提交表



⑨作业批改表



1. 检查代码完整性的测试性代码

|  |
| --- |
| 附录代码 检查代码完整性 |
|  |
| import ast  import os  import re  def check\_syntax(file\_path):  """检查文件的语法是否正确"""  try:  with open(file\_path, "r", encoding='utf-8') as source:  ast.parse(source.read())  print(f"{file\_path} 语法检查通过")  return True  except SyntaxError as e:  print(f"{file\_path} 存在语法错误: {e}")  return False  def check\_style(file\_path):  """简单的代码风格检查（例如，是否有多余空格等）"""  with open(file\_path, "r", encoding='utf-8') as file:  lines = file.readlines()    style\_issues = False  for line\_number, line in enumerate(lines, start=1):  # 检查行尾空白字符  if re.search(r'\s+$', line):  print(f"{file\_path} 第{line\_number}行存在行尾空白字符")  style\_issues = True  # 可以添加更多的代码风格检查规则...    if not style\_issues:  print(f"{file\_path} 风格检查通过")  return not style\_issues  def check\_comments(file\_path):  """检查是否每一段代码都有注释"""  with open(file\_path, "r", encoding='utf-8') as file:  content = file.read()  # 简单地检查每个函数或类定义前是否有注释  comments\_missing = False  for match in re.finditer(r'(def |class )', content):  preceding\_text = content[:match.start()]  if not re.search(r'#.+\n$', preceding\_text, re.MULTILINE):  print(f"{file\_path} 在位置 {match.start()} 前缺少注释")  comments\_missing = True    if not comments\_missing:  print(f"{file\_path} 注释检查通过")  return not comments\_missing  def check\_functionality(file\_path):  """检查特定功能实现的完整性"""  # 这里需要根据实际的功能需求编写相应的检测逻辑  # 例如，检查是否所有定义的函数都被调用了，或者某些关键功能是否存在  pass  def main():  directory = '.' # 设定为你的代码目录  all\_passed = True  for root, dirs, files in os.walk(directory):  for file in files:  if file.endswith(".py"):  file\_path = os.path.join(root, file)  if not (check\_syntax(file\_path) and check\_style(file\_path) and check\_comments(file\_path)):  all\_passed = False  if all\_passed:  print("所有检查通过！")  else:  print("一些检查未通过，请查看输出进行修正。")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
|  |