目 录

[1. 课设目的 1](#_Toc17376)

[2. 开发环境 1](#_Toc21539)

[3. 课设内容 1](#_Toc499)

[4. 需求分析 2](#_Toc19561)

[5. 设计与实现过程 3](#_Toc25153)

[5.1数据库设计 3](#_Toc11838)

[5.2 系统的设计 6](#_Toc6701)

[6. 调试过程 10](#_Toc15710)

[7. 小结 19](#_Toc23230)

[8. 参考文献 20](#_Toc31215)

# 

## 1. 课设目的

《Java Web程序课程设计》是网络工程专业的一门专业必修课，是一门实践性较强的课程。旨在通过实践复习、巩固Java Web的基础知识，进一步加深对Java Web技术的理解和掌握；课程设计为学生提供了一个既动手又动脑，独立实践的机会，将课本上的理论知识和实际有机的结合起来，锻炼学生的分析解决实际问题的能力。培养学生在项目开发中团队合作精神、创新意识及能力，提高学生分析能力、创新能力和文档制作能力。

## 2. 开发环境

1.操作系统：Windows10

2.开发工具：IntelliJ IDEA Navicat Maven

3.开发语言：Java

4.框架：SpringBoot Mybatis

## 3. 课设内容

基于Java Web的教务系统

教务管理管理系统在IntelliJ IDEA中创建Web项目，然后在src目录下创建controller、dao、domain、entity、service、util、common、spring包，分别用于存放控制器、数据库、实体、工具类。同时在web目录下创建css、img包存放样式和图片。系统需要完成以下功能：

（1）管理员登录后可以管理专业、学院、学生和课程的信息，并在系统的主界面显示登录状态，点击退出，可以返回到登录界面。登录的账号密码有误显示登录失败界面。

（2）在教务系统的首页的功能列表中点击学生信息管理，可以选择添加、更改和删除学生信息。

（3）在教务系统的首页的功能列表中点击专业信息管理，可以选择添加、更改和删除专业信息。当删除专业信息时，如果该专业下有学生，则提示不能删除。

（4）在教务系统的首页的功能列表中点击课程信息管理，可以选择添加、更改和删除课程信息。

（5）在教务系统的首页的功能列表中点击学院信息管理，可以选择添加、更改和删除学院信息。当删除学院信息时，如果该学院下有专业、课程，则提示不能删除。

## 4. 需求分析

该教务管理系统的需求分析如下：

一，用户登录：系统需要提供管理员登录功能，管理员需要输入正确的账号和密码才能登录系统。如果账号或密码输入错误，系统应该给出相应的错误提示。

二，学生信息管理：

（1）添加学生信息：管理员可以添加学生的基本信息，包括学生的学号、姓名、专业、学院。

（2）修改学生信息：管理员可以修改学生的基本信息，包括学生的学号、姓名、专业、学院。

（3）删除学生信息：管理员可以删除学生的信息，删除时需要确认操作，如果该学生已经选课或有其他关联数据，删除操作应给出相应的提示。

三，课程信息管理：

（1）添加课程信息：管理员可以添加课程的基本信息，包括课程号、课程名称、所属学院。

（2）修改课程信息：管理员可以修改课程的基本信息，包括课程号、课程名称、所属学院。

（3）删除课程信息：管理员可以删除课程的信息，删除时需要确认操作，如果该课程已经有学生选课或有其他关联数据，删除操作应给出相应的提示。

四，专业信息管理：

（1）添加专业信息：管理员可以添加专业的基本信息，包括专业号、专业名称、所属学院。

（2）修改专业信息：管理员可以修改专业的基本信息，包括专业号、专业名称、所属学院。

（3）删除专业信息：管理员可以删除专业的信息，删除时需要确认操作，如果该专业下有学生或有其他关联数据，删除操作应给出相应的提示。

五，学院信息管理：

（1）添加学院信息：管理员可以添加学院的基本信息，包括学院号、学院名称。

（2）修改学院信息：管理员可以修改学院的基本信息，包括学院号、学院名称。

（3）删除学院信息：管理员可以删除学院的信息，删除时需要确认操作，如果该学院下有专业或有课程或有其他关联数据，删除操作应给出相应的提示。

以上是教务管理系统的基本功能需求。

## 5. 设计与实现过程

### 5.1数据库设计

当前系统使用Mysql数据库，采用Navicat连接管理，系统数据库名为academic\_db。数据库表结构如图5.1所示：

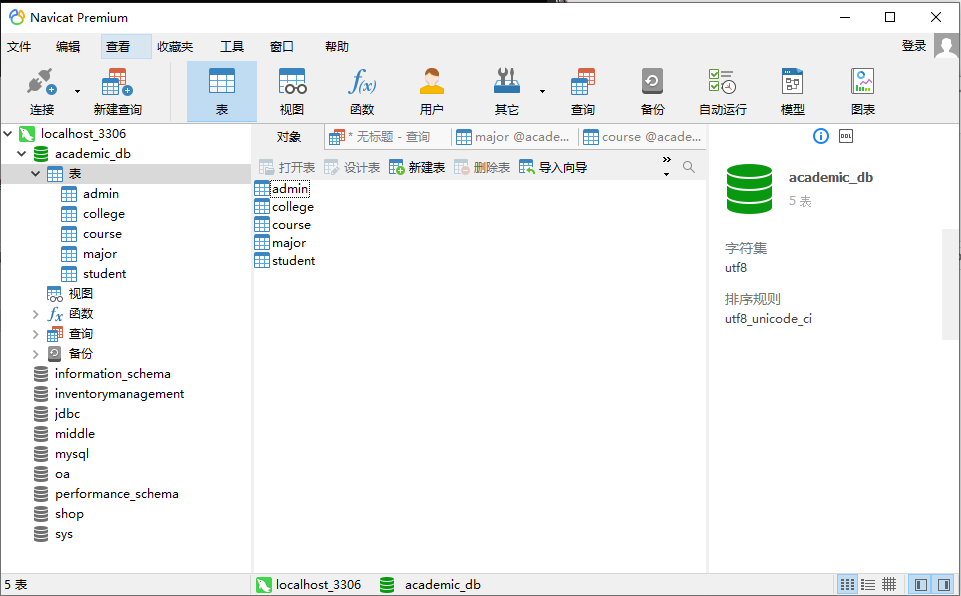


图5.1-表汇总结构

admin(用户名,密码)该表的物理结构如图5.2所示：

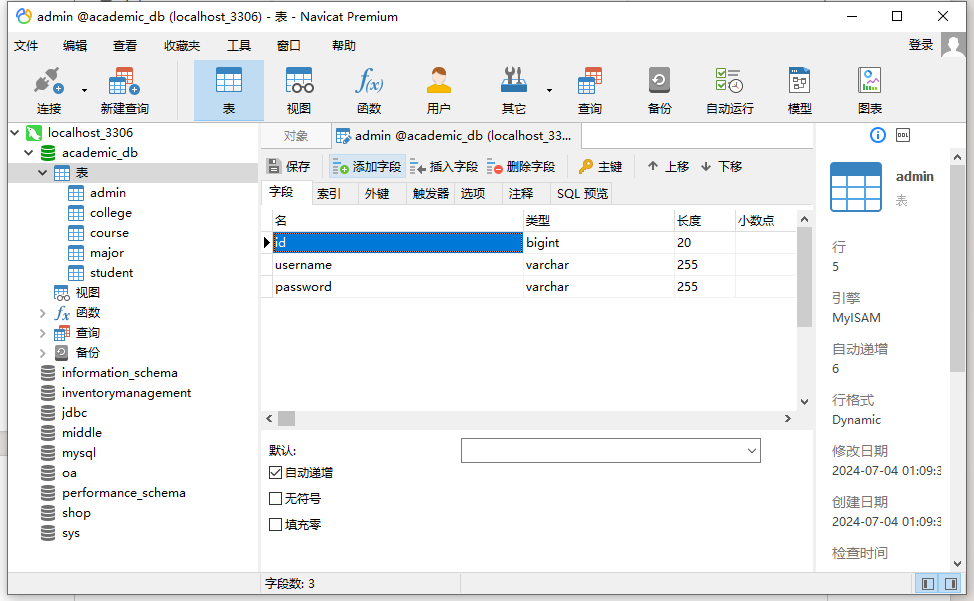


图5.2-管理员表结构

college(学院名称)该表的物理结构如图5.3所示：

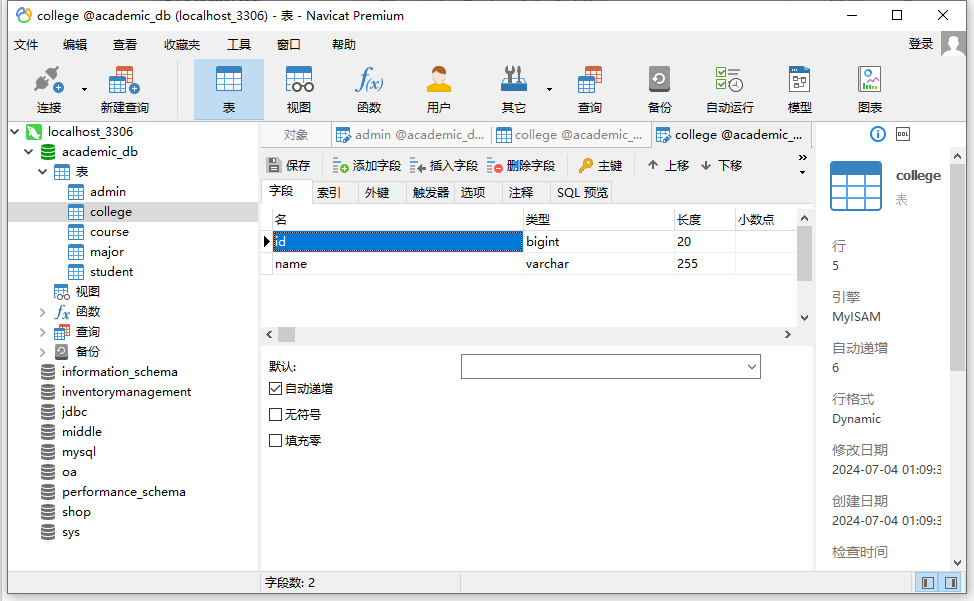


图5.3-学院表结构

course(课程名,所属学院)该表的物理结构如图5.4所示：

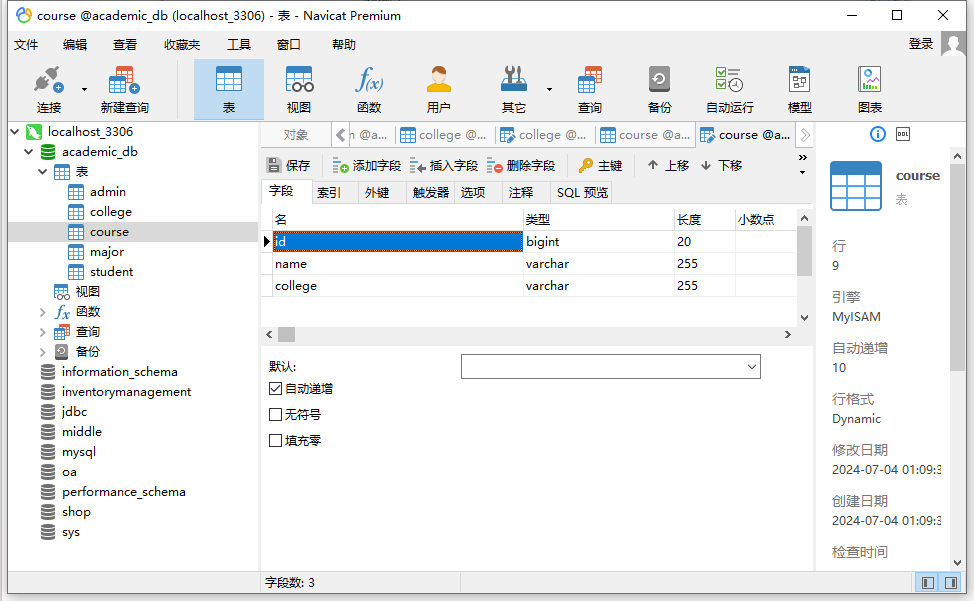


图5.4-课程表结构

major(专业名，所属学院)该表的物理结构如图5.5所示：

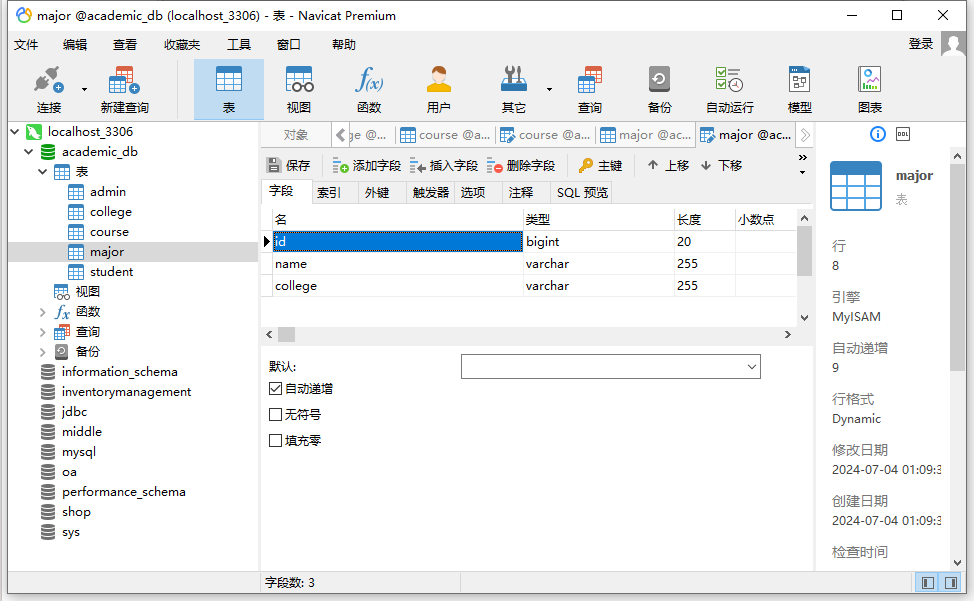


图5.5-专业表结构

student(学生名，专业，所属学院)该表的物理结构如图5.6所示：

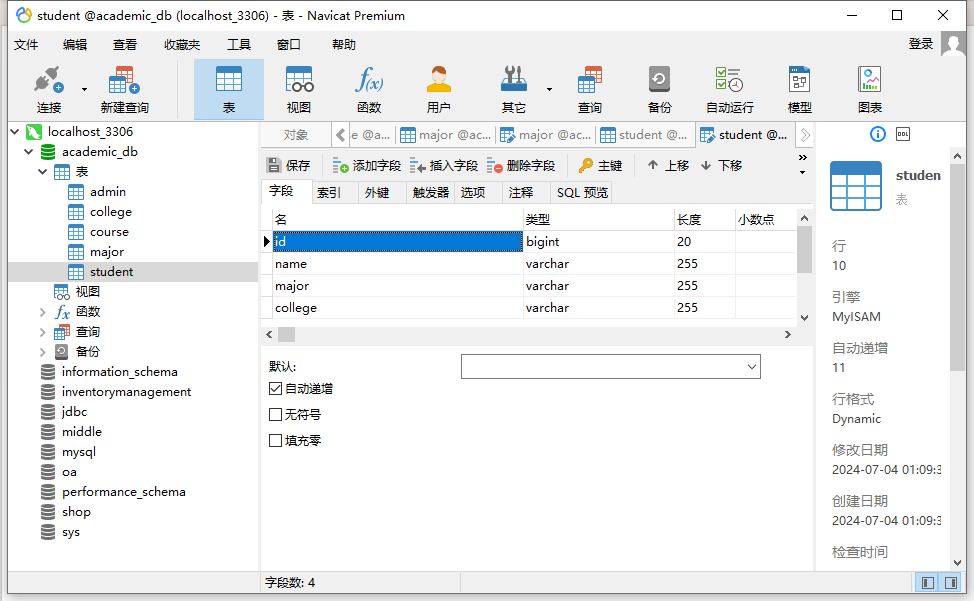


图5.6-学生表结构

### 5.2 系统的设计

前端采用纯HTML+CSS，后端则采用SpringBoot+Mybatis的设计来进行前后端分离，

主要框架包括8层分别是config层、controller层、dao层、entity层、interceptor层、service层。负责处理用户请求，控制整个请求的流程和逻辑。该层主要包括了一系列的HTTP请求处理方法（比如：处理GET、POST请求），并调用相关的service进行数据的读写操作。service层：负责实现业务逻辑，处理数据操作。相比直接在controller层进行数据操作，service层负责所有和数据操作相关的业务逻辑，包括数据库操作、事务管理等操作。在该层中还负责调用dao层进行数据的读写操作。dao层：负责进行数据的存取操作。该层一般包括了一组数据的访问接口，用于进行更新、查询等数据库操作。该层通常包含了一组实现具体业务逻辑的Java类和接口，这些类和接口一般都基于业务实体来定义，为上层业务逻辑提供抽象的领域模型。entity层：用于定义数据库数据模型，通常在MyBatis等ORM框架中使用，默认采用JavaBean的封装。

其中SpringBoot使用了 Spring Web， MyBatis，MySQL Driver， Spring Data JPA，Thymeleaf，Lombok的依赖，其作用分别是：

1. Spring Web

作用：Spring Web是Spring Framework的一部分，用于构建基于Web的应用程序。它提供了强大的MVC框架，用于处理HTTP请求和响应，支持表单提交、数据验证、视图解析等功能。

使用：在控制层（Controller）中使用Spring MVC注解（如@Controller、@RequestMapping、@GetMapping、@PostMapping等）来处理Web请求。

2. MyBatis

作用：MyBatis是一款持久层框架，用于简化对数据库的访问。它通过XML或注解的方式将SQL语句与Java对象映射起来，提供了灵活的SQL构建和映射功能。

使用：在数据访问层（DAO）中使用MyBatis的注解或XML文件编写SQL语句，进行数据的增删改查操作。

3. MySQL Driver

作用：MySQL Driver是用于连接和操作MySQL数据库的JDBC驱动。它使得Java应用程序能够通过JDBC与MySQL数据库进行通信。

使用：在Spring Boot项目中配置数据源，使用MySQL数据库进行数据持久化。

4. Spring Data JPA

作用：Spring Data JPA提供了一种简化的方式来访问数据库。它基于JPA（Java Persistence API），通过Repository接口定义数据访问层，简化了DAO层的开发。

使用：虽然项目中主要使用MyBatis，但Spring Data JPA可以作为辅助，用于更快速的开发某些数据访问功能。

5. Thymeleaf

作用：Thymeleaf是一款现代的服务器端Java模板引擎，用于生成动态HTML内容。它与Spring Boot无缝集成，能够直接在HTML页面中使用Thymeleaf标签进行数据绑定和模板渲染。

使用：在视图层（View）中使用Thymeleaf模板引擎生成动态HTML页面，展示从控制层传递过来的数据。

6. Lombok

作用：Lombok是一个Java库，通过简单的注解可以自动生成常用的Java代码（如getter、setter、构造函数、toString方法等），从而减少样板代码，提高开发效率。

使用：在实体类和其他需要生成样板代码的类中使用Lombok注解。

系统的大体思路是首先登录页面，如图5.7所示：

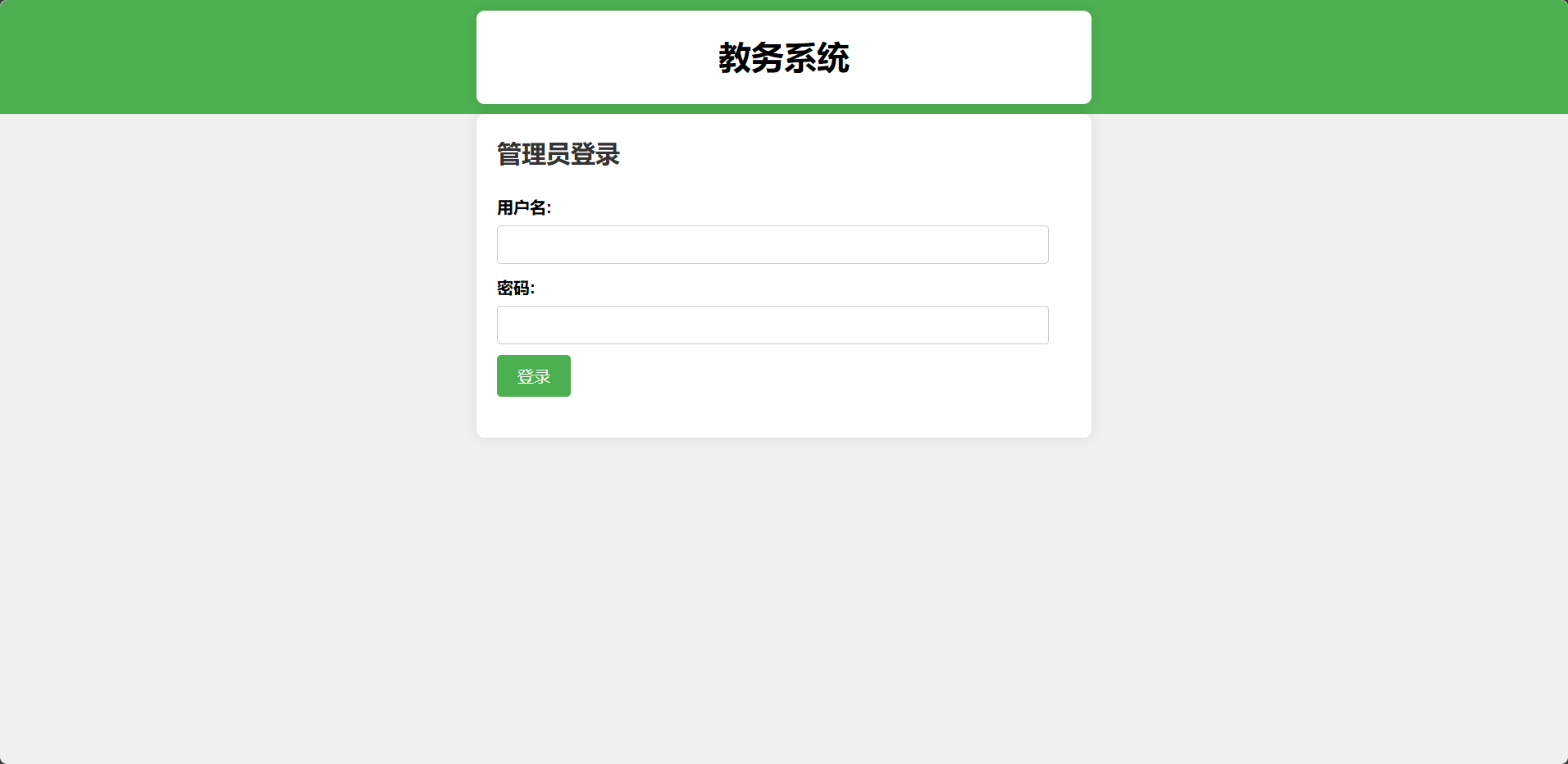


图5.7-登陆页面图

主页面，如图5.8所示：



图5.8-主页面图

列表页面，如图5.9所示：



图5.9-列表页面图

功能页面，如图5.10，图5.11所示：



图5.10-添加页面图



图5.11-编辑页面图

## 调试过程

首先访问/login实现登录时必须输出正确的用户名、密码，任何一项输入错误则将登录失败。登录界面如图6.1,图6.2所示:



图6.1-登录页面图



图6.2-登录失败图

如果输入了正确的账号密码则会跳转/home页面，此页面是主要的管理页面，提供了在线信息，各个模块的管理，登出等。如图6.3所示：



图6.3-主页面图

接下来四个管理模块分别为/student/list、/major/list、/course/list、/college/list即学生信息管理、专业信息管理、课程信息管理、学校信息管理。

首先是学生管理系统，有三个功能编辑、删除、添加。如图6.4，图6.5，图6.6所示：



图6.4-学生列表图



图6.5-编辑学生图



图6.6-添加学生图

接着是专业管理系统，有三个功能编辑、删除、添加。如图6.7，图6.8，图6.9所示：



图6.7-专业列表图



图6.8-专业编辑图



图6.9-添加专业图

然后是课程管理系统，有三个功能编辑、删除、添加。如图6.10，图6.11，图6.12所示：



图6.10-课程列表图

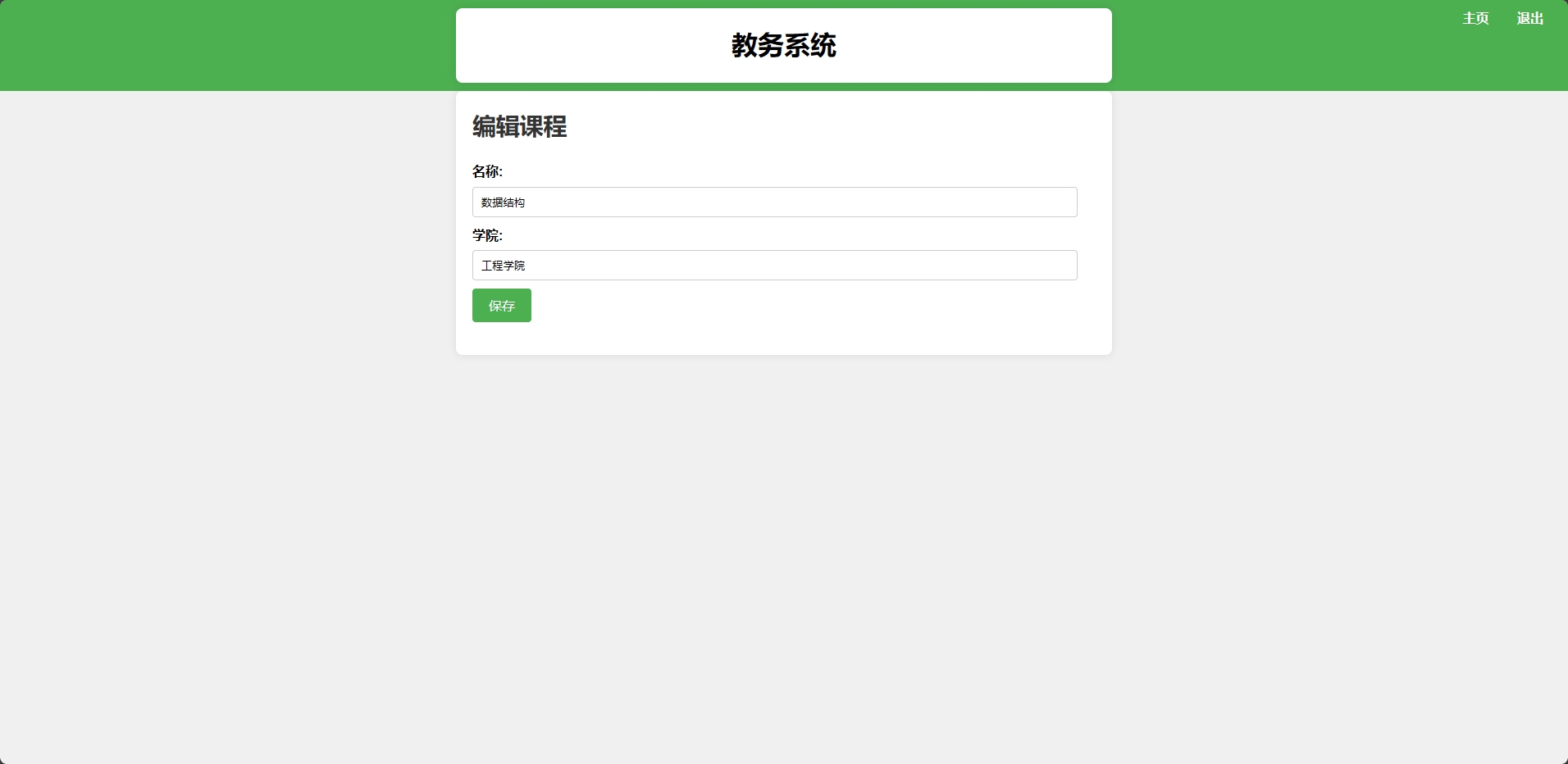


图6.11-课程编辑图

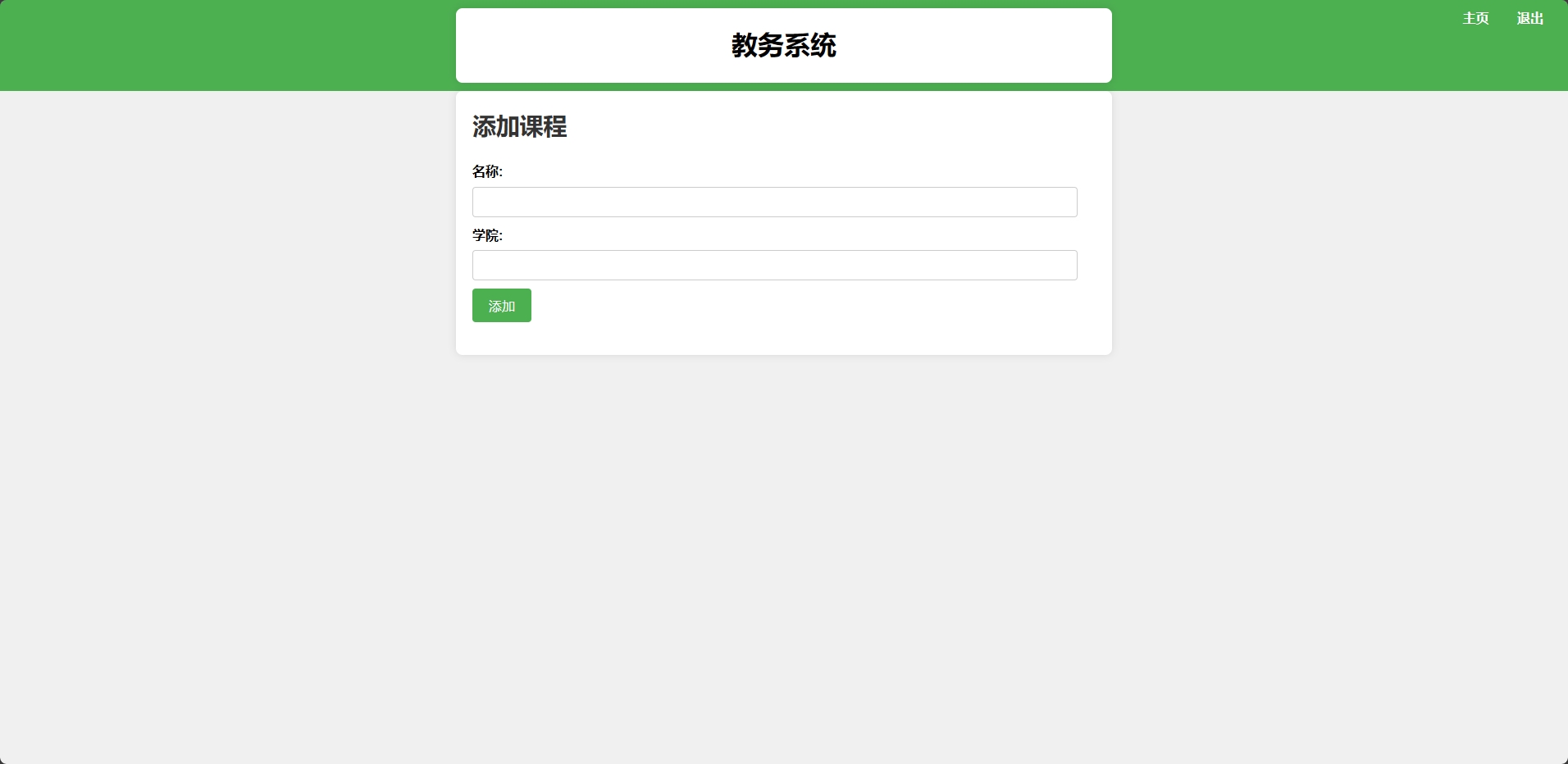


图6.12-添加课程图

最后是学院管理系统，有三个功能编辑、删除、添加。如图6.13，图6.14，图6.15所示：



图6.13-学院列表图

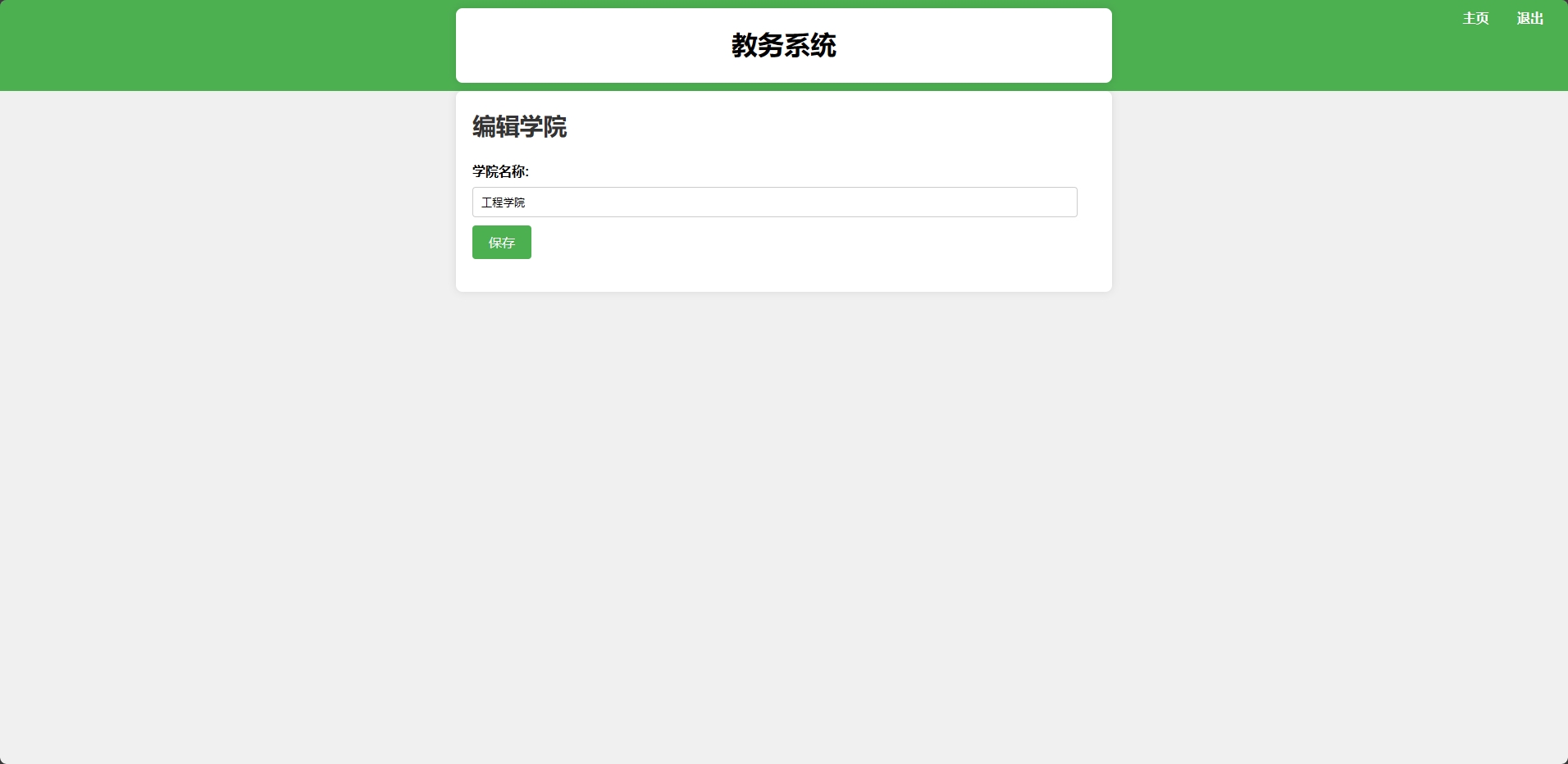


图6.14-学院编辑图

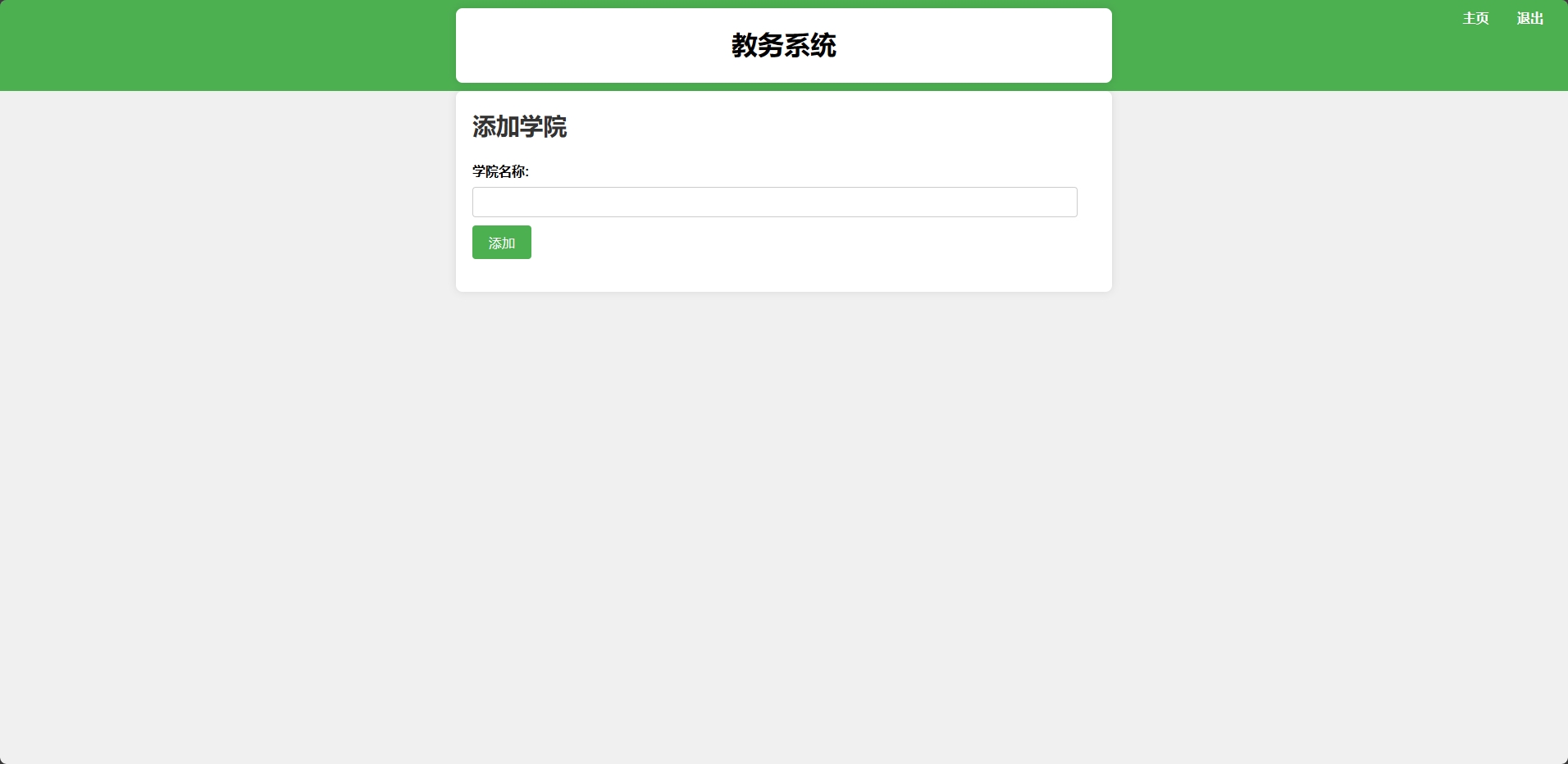


图6.15-添加学院图

接下来下面我将会进行三项安全测试

XSS跨站脚本攻击测试，我们在使用添加功能可以嵌入恶意的JS代码，来执行任意JS代码，会造成COOKIE盗取，DDOS攻击，挂黑链等等。我们插入语句后没有弹窗，防护安全如图6.16所示：



图6.16-XSS测试图

未授权访问漏洞，我们在不登陆的情况下直接访问/home，发现没有成功而是直接返回到了/login，如图6.17所示：

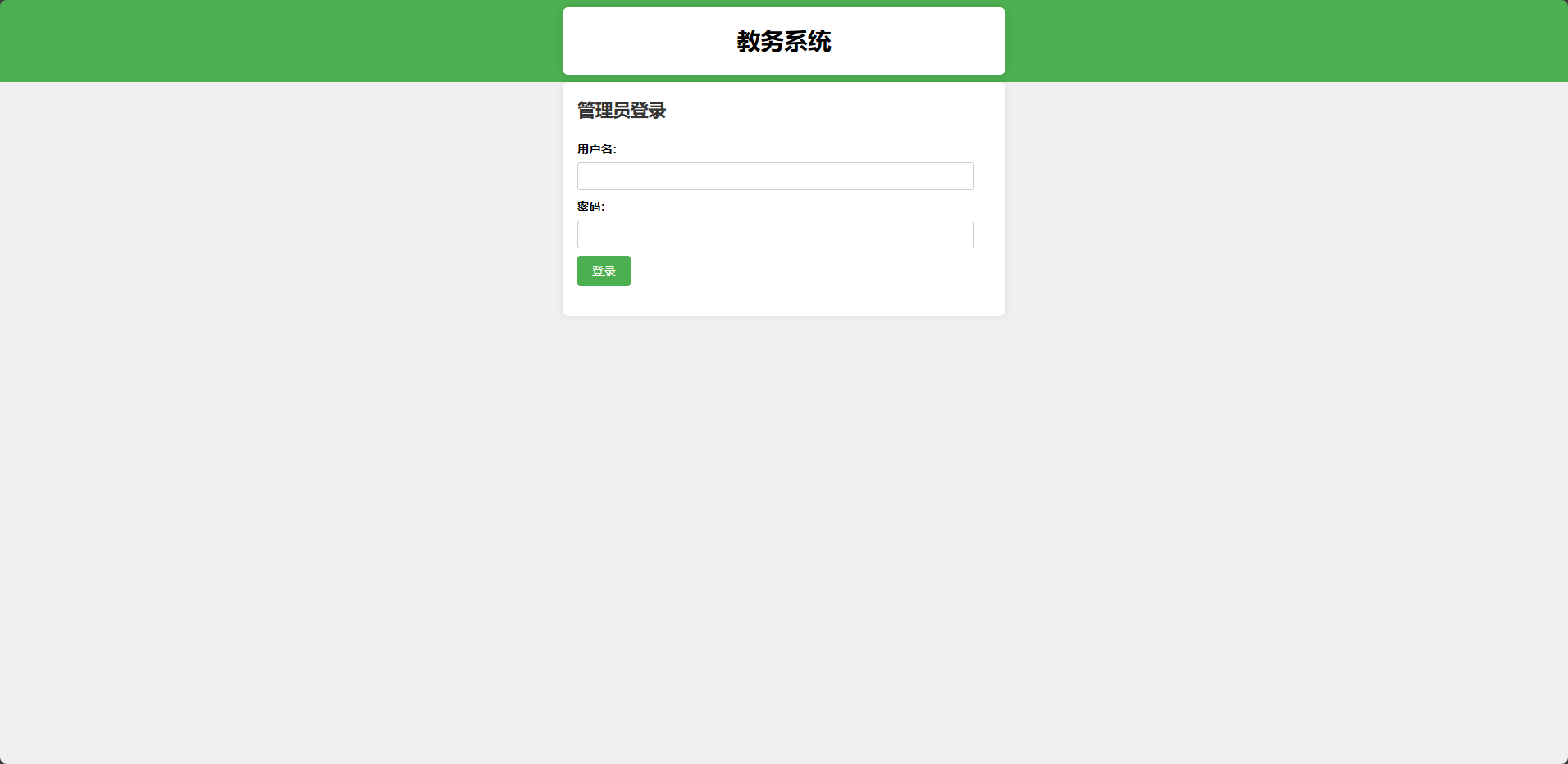


图6.17-未授权访问测试图

SQL注入，我们在登陆页面使用SQLMAP工具去进行SQL注入攻击，经过测试页面安全因为使用了预编译语句，安全性高，效率更快。如图6.18所示;

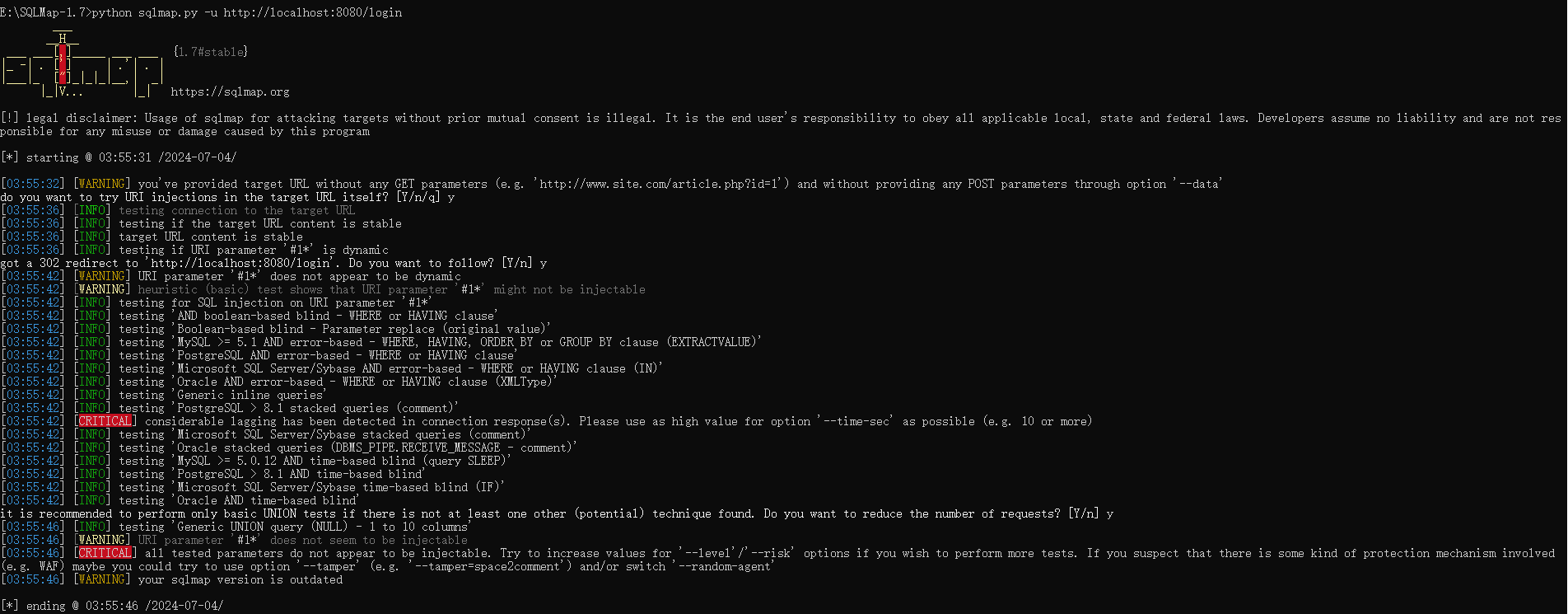


图6.18-SQL注入测试图

## 7. 小结

在这次制作基于Spring Boot的教务系统项目过程中，我经历了从项目规划、设计到实现的完整流程，收获颇丰。以下是我在这个过程中总结的一些心得体会。

#### 项目规划和需求分析，项目的开始阶段，我花了大量时间进行需求分析和项目规划。明确了教务系统的核心功能，包括管理员登录、学生信息管理、专业信息管理、课程信息管理、学院信息管理等。这一步非常重要，它决定了项目的整体架构和功能模块的划分。

#### 选择技术栈，在技术栈的选择上，我选用了Spring Boot作为后端框架。Spring Boot的自动配置和简化的开发流程使得项目的开发效率大大提高。此外，使用MyBatis作为数据持久层框架，通过简单的配置就能实现对数据库的操作。前端方面，采用了Thymeleaf作为模板引擎，它与Spring Boot无缝集成，使得前后端交互更加方便。

#### 系统设计与实现，在系统设计阶段，我遵循了分层架构的原则，分为控制层（Controller）、服务层（Service）、数据访问层（DAO）和实体层（Entity）。这样的架构使得系统的结构更加清晰，代码更加易于维护和扩展。

#### 在实际实现过程中，我遇到了许多挑战。例如，如何在删除专业信息前检查是否存在相关的学生，如何在删除学院信息前检查是否存在相关的专业或课程等。通过不断的调试和测试，我逐步解决了这些问题。特别是在处理数据完整性和业务逻辑时，Spring Boot和MyBatis提供的强大功能极大地简化了开发工作。

#### 用户体验与界面设计，在用户体验和界面设计方面，我注重简洁和易用。使用CSS对页面进行美化，使得界面更加美观和用户友好。通过添加消息显示区域，用户能够及时收到操作反馈，如删除操作的成功与否等。

#### 学习与成长在这个项目中，我不仅加深了对Spring Boot和MyBatis的理解，还学会了如何在实际项目中应用这些技术。此外，通过解决各种实际问题，我的编码能力和问题解决能力也得到了提升。

### 8. 参考文献

[1] 陈家祺. 运动图像处理在车型识别中的应用[J]. 汽车工程，1998,20(6): 343~346.

[2] 陈家祺. C程序设计教程[M]. 北京：新华出版社，1999.11.

[3] Shihang S. Fuzzy self-organizing controller and its application for dynamic processes[J]. Fuzzy Sets and Systems, 1998,26:151~164.

[4] Kaare Christian. Borland C++ Techniques & Utilitis[M]. Ziff-DavisPress，1993.