仲恺农业工程学院

毕 业 设 计

基于WEB的网络安全学习网站的设计

姓 名 叶凯佳

院（系） 信息科学与技术学院

专业班级 网络工程202

学 号 202010254227

指导教师 陈文文 李泽楷

职称/职务 讲师

论文答辩日期 2024年5月11日

仲恺农业工程学院教务部制

**Design of a Cybersecurity Learning Website Based on Web**

Ye Kaijia

College of Information Science and Technology  
Zhongkai University of Agriculture and Engineering  
Guangzhou,China

**Supervisor: Wenwen Chen Zekai Li**

摘 要

随着社会、经济和文化的数字化转型，人们对互联网的依赖性越来越强。金融、医疗、能源、交通等领域的数字化使得网络安全问题的解决变得尤为迫切，而在教育方面，网络安全教育直至近些年才逐渐被提起与重视，相关的从业人员与学生需要一个提供学习、实践与交流的学习社区环境，通过学习平台，人们能够获取有关网络攻防、网络协议、漏洞分析等方面的知识，能够实践使用网络攻防的工具，从而更好地理解和应对复杂的网络威胁。本平台使用时下流行的技术开发前后端，前端采用了Vue完成界面的搭建与设计，服务端采用了Java+SpringBoot框架,利用SSH协议和WebSocket通讯协议等技术实现、提供用户一个具有博客社区、网络攻防实践、密码加解密功能的网络安全学习平台。

关键词：网络安全学习平台 WebSocket Kali-Liunx Vue JavaSpringBoot

**Abstract**

With the digitization transformation of society, economy, and culture, people's dependence on the Internet is increasing. The digitization of sectors such as finance, healthcare, energy, and transportation has made solving cybersecurity issues particularly urgent. In the field of education, cybersecurity education has only recently gained attention, and professionals and students in this domain require a learning community environment that facilitates learning, practice, and communication. Through a learning platform, individuals can access knowledge about network attacks and defense, network protocols, vulnerability analysis, and more. This enables practical use of tools for network attack and defense, enhancing the understanding and response to complex cybersecurity threats. The platform is developed using popular technologies for both front-end and back-end. The front-end is built and designed using Vue, while the back-end utilizes Java and the SpringBoot framework. The platform leverages technologies such as SSH and WebSocket communication protocols to provide users with a cybersecurity learning platform featuring a blog community, network attack and defense practice, and cryptography study functionalities.

**Keywords:** Cybersecurity learning platform; WebSocke; Kali-Liunx; Vue; Java; SpringBoot;

目 录

[1 前言 1](#_Toc165113137)

[1.1 研究背景及意义 1](#_Toc165113138)

[1.2 国内外研究现状 1](#_Toc165113139)

[1.3 本文研究内容 2](#_Toc165113140)

[2 相关理论和技术介绍 2](#_Toc165113141)

[2.1 Java语言相关简介 2](#_Toc165113142)

[2.2 SpringBoot框架相关简介 3](#_Toc165113143)

[2.3 Vue.js框架相关简介 3](#_Toc165113144)

[2.4 WebSocket相关简介 4](#_Toc165113145)

[2.5 Mongo相关简介 5](#_Toc165113146)

[2.6 Security框架 5](#_Toc165113147)

[3 系统分析 6](#_Toc165113148)

[3.1 可行性分析 6](#_Toc165113149)

[3.1.1 技术可行性 6](#_Toc165113150)

[3.1.2 经济可行性 6](#_Toc165113151)

[3.1.3 市场可行性 7](#_Toc165113152)

[3.2 需求分析 7](#_Toc165113153)

[3.2.1 用户人群 7](#_Toc165113154)

[3.2.2 用户需求描述 7](#_Toc165113155)

[3.2.3 竞品分析 7](#_Toc165113156)

[3.2.4 需求概述 9](#_Toc165113157)

[3.2.5 功能分析 9](#_Toc165113158)

[3.3 系统用例图 10](#_Toc165113159)

[3.4 数据流程图 11](#_Toc165113160)

[3.5 数据字典 12](#_Toc165113161)

[3.6 系统类图 15](#_Toc165113162)

[4 系统概要设计 15](#_Toc165113163)

[4.1 系统设计原理 15](#_Toc165113164)

[4.2 系统开发环境 16](#_Toc165113165)

[4.3 系统整体架构 16](#_Toc165113166)

[4.4 系统功能结构设计 16](#_Toc165113167)

[4.5 数据库设计 17](#_Toc165113168)

[4.5.1 设计原则 17](#_Toc165113169)

[4.5.2 概念结构设计 18](#_Toc165113170)

[4.5.3 物理结构设计 20](#_Toc165113171)

[5 系统详细设计 23](#_Toc165113172)

[5.1 系统时序图 23](#_Toc165113173)

[5.2 系统流程图 28](#_Toc165113174)

[5.3 系统讲解 28](#_Toc165113175)

[5.3.1 论坛模块与导航栏 28](#_Toc165113176)

[5.3.2 密码加解密模块 31](#_Toc165113177)

[5.3.3 用户管理模块 32](#_Toc165113178)

[5.3.4 攻击机/靶机模块 32](#_Toc165113179)

[5.3.5 个人信息模块 35](#_Toc165113180)

[5.3.7 登陆注册模块 36](#_Toc165113181)

[5.3.8 资源中心模块 37](#_Toc165113182)

[6 系统测试 38](#_Toc165113183)

[6.1 测试需求 38](#_Toc165113184)

[6.2 测试目标 38](#_Toc165113185)

[6.3 测试用例 38](#_Toc165113186)

[6.4 测试总结 40](#_Toc165113187)

[7 总结与展望 41](#_Toc165113188)

[参考文献 43](#_Toc165113189)

[致谢 44](#_Toc165113190)

# 1 前言

## 1.1 研究背景及意义

在当前数字化转型的趋势下，人们对互联网的依赖加深，各行业也在加速数字化。然而，这种转型不仅带来便利，也引发了网络安全挑战。金融、医疗、能源、交通等领域的数字化转型为生产生活带来了更大的便利，但与此同时也带来了巨大的网络安全挑战。

在这个背景下，建立一个网络安全学习社区显得尤为重要。互联网的普及虽然带来了信息传递、存储和共享的便利，但也伴随着各种网络威胁，如恶意软件、网络攻击和数据泄露等。这些问题使得网络安全变得愈发复杂和紧迫，当前市场上尚未有一个成熟的、提供实践的网络安全学习平台，这为构建一个全面、系统的学习社区提供了契机。

在信息化时代，机遇与挑战同在，网络安全已经不仅仅关乎个人和企业，更关系到整个国家的安全和发展。习近平总书记多次提到网络安全与信息化是事关国家安全和国家发展、事关广大人民群众工作生活的重大战略问题。这表明网络安全已经逐渐上升到国家战略层面，对于构建网络强国的目标至关重要。

建设网络安全学习社区能够提供全面、系统的培训，帮助专业人才更好地应对复杂多变的网络安全威胁。通过学习社区，学生可以获取有关网络攻防、网络协议、漏洞分析等方面的知识，并实践使用网络安全工具。学习社区的建成将促进网络安全专业人才之间的合作与交流，为技术创新和职业发展提供更多机会。

最重要的是，网络安全学习社区的建设符合国家网络安全工作的重要内容。习近平总书记多次强调提升全民网络安全意识和技能，构建网络安全学习社区将为这一目标提供有力支持。通过这样的社区，不仅可以培养更多的网络安全专业人才，也能够普及网络安全知识，提高全民网络安全素养。

总的来说，构建网络安全学习社区不仅填补了市场空白，更对于网络安全的教育而言具有深远的意义。通过提供更全面、系统、实用的学习环境，学习社区将帮助初学者和专业者更好地应对日益复杂的网络安全挑战，为国家的网络安全工作做出积极贡献。网络安全学习社区不仅是一个技术平台，更可以是国家网络安全战略的一颗螺丝，为网络强国的建设贡献力量。

## 1.2 国内外研究现状

国内外和网络安全学习相关的社区有freebuf社区、HackMyVM社区、OWASP社区等,各个社区都有各自的聚集特点与用户对象群体。

freebuf社区更注重教学和知识的交流，拥有公开课、用户服务、行业服务、论坛等功能，其中公开课面向所有人售卖网络安全相关的知识视频，用户服务面向所有人提供一个提交漏洞与投稿的平台、而行业服务则是为政企单位等提供一个安全平台搭建的方案与服务，论坛则是给所有用户提供一个交流的平台；

HackMyVM社区更多是提供了各种各样的定制虚拟机环境文件供用户下载安装，用户在安装虚拟机环境后可以尝试攻破此环境，并且可以查看此环境被攻破的记录速度以及攻破者攻破的方法，还可以在发布环境的帖子下交流攻破虚拟机的心得，同时，HackMyVM社区还定时举办CTF竞赛供玩家交流与竞技；

OWASP社区更多是在更新一些前沿的系统或者网站漏洞，它致力于安全资源与安全资讯的分享，比较出名的分享项目有：OWASP API 安全 TOP 10 项目、OWASP机器学习安全风险 TOP 10等，它也定期举办各类网络安全峰会，通过网易云课堂分享一些免费或者付费的培训项目。

## 1.3 本文研究内容

本文研究内容是从初学者学习角度出发进行需求分析，设计一个面向网络安全学习的初学者人群的，拥有实践学习和理论学习功能结合的一个网络安全学习社区，结合本人的技术栈进行开发搭建具有实践性的学习环境，包括模拟攻击场景，论坛交流知识，在线加密解密等。

# 2 相关理论和技术介绍

## 2.1 Java语言相关简介

Java是一种跨平台、面向对象的编程语言，由Sun Microsystems（现在是Oracle Corporation）于1995年发布。它的设计目标是实现一次编写，到处运行（WORA），即编写一次代码，可以在任何支持Java的平台上运行，无需重新编译。这使得Java成为一种非常流行的语言，用于开发各种类型的应用程序，包括桌面应用程序、Web应用程序、移动应用程序和大型企业级应用程序。

它具有如下特点：

1. 跨平台性：“Write Once, Run Anywhere（一次编写，随处运行）”，Java的跨平台性是其最重要的特点之一。通过Java虚拟机（JVM），Java程序可以在不同的操作系统上运行，只需编写一次代码，就可以在Windows、Linux、macOS等系统上运行。
2. 面向对象：Java是一种纯粹的面向对象编程语言，支持类、对象、继承、多态等面向对象的特性，使得代码结构清晰、模块化程度高，易于理解和维护。
3. 高性能：Java通过即时编译技术（Just-In-Time Compilation）实现了高性能的运行效率，使得Java应用程序在执行速度上与C/C++等编程语言相媲美。
4. 安全性：Java具有严格的安全机制，包括字节码验证、安全管理器、安全沙箱等，可以有效防止恶意代码的执行，保护系统安全。

除此之外，Java本身作为一个多年的成熟的语言，其生态环境是几乎其他所有语言难以企及的，使用Java语言开发后端从很大程度上减少了本系统开发的工作量。

## 2.2 SpringBoot框架相关简介

Spring boot是Spring模块框架之一，它提供建立在流行的Java Spring框架之上的快速应用程序开发（RAD）功能，以在执行项目时提供对信息的快速访问。使用 Java 配置和运行基于 Web 的和企业开发非常容易设置 RAD[1] 。它提供了丰富的Spring模块化支持，通过简化配置和提供各种开箱即用的功能，可以帮助开发者更轻松快捷地构建出企业级应用。

它具有如下四个核心特性：

1. 自动装配：Spring Boot通过约定大于配置的原则，大大简化了应用程序的配置过程。它提供了默认的配置选项，简单的配置甚至零配置即可运行
2. 编码简单化：提供大量注解简化编码，提高效率
3. 起步依赖整合：将所需的常见依赖按组聚集在一起，形成单条依赖
4. 部署简单：Spring Boot提供了嵌入式的Web服务器（如Tomcat、Jetty等），可以直接打包成可执行的JAR文件，从而简化了应用程序的部署和启动过程。

本系统的后端架构采用了SpringBoot开发，结合Maven引入其他依赖，与多个数据库连接操作数据，实现了对数据的安全保护以及业务逻辑层面的增删改查。

SpringBoot同时还能够为一些大型项目提供非业务功能特点，包括安全检测，健康检测，无须在项目中配置Xml方式[2]。

## 2.3 Vue.js框架相关简介

Vue.js是目前国内技术界最流行的前端框架之一。它基于常见的标准 三件套：HTML、CSS 和 JavaScript 构建，并提供了一套声明式与组件化的编程模板，帮助开发者更加高效地开发用户界面。

它的主要特点包括：

1. 轻量级: Vue.js相比于其他框架要更加轻量级，体积小，学习曲线平缓，易于上手。
2. 双向数据绑定: Vue.js采用了响应式数据绑定的机制，将视图和数据双向绑定，当数据发生变化时，视图会自动更新，大大便利了开发。
3. 组件化: Vue.js将页面拆分为多个组件，每个组件具有独立的逻辑和样式，便于代码复用和维护。
4. 虚拟DOM: Vue.js通过虚拟DOM技术，将页面更新的成本降到最低，提高了页面渲染的效率。
5. 指令系统: Vue.js提供了丰富的指令系统，例如v-bind、v-if、v-for等，用于简化模板代码和实现各种交互效果。
6. 生命周期钩子: Vue.js提供了包括created、mounted等多个生命周期函数，通过这些函数，开发者可以在不同阶段添加自定义逻辑来完成页面初始化、数据更新等操作。
7. 社区支持: 作为流行的前端框架，Vue.js拥有庞大的开发者社区和丰富的第三方插件，提供了丰富的资源和支持。

本系统前端方面使用了vue2作为开发框架，将各个页面抽离成一个或者多个组件组合，构建了一个相对完善的前端界面，同时结合后端做了双端访问验证，同时保证了页面和接口访问双保障。

## 2.4 WebSocket相关简介

WebSocket是HTML5的一种通信协议，与传统的HTTP通信方式相比，它具有许多优势和特点：

1. WebSocket API通过允许浏览器与服务器保持异步套接字连接来解决直接向客户端发送数据的问题。通过保持这种连接，客户端能够立即将数据发送到服务器，无需重新建立连接。此外，只要连接保持打开状态，服务器就能够随时向客户端发送数据[3]。这种特性使得WebSocket非常适合实时通讯应用，包括本系统所需要的和liunx交互等，因为它能够实现即时的数据传输和交互。
2. WebSocket也具有较低的通信延迟和数据传输量。与每次HTTP请求都需要携带完整的HTTP头信息不同，WebSocket建立连接后只需在开始时进行一次握手，之后的数据传输只需要少量的头部信息，这大大减少了通信时的数据传输量和延迟，提升了系统的性能和效率。
3. WebSocket还具有跨域通信的能力。虽然在Web开发中跨域通信通常受到限制，但是WebSocket协议可以在不同域名之间进行通信，这为跨域应用场景提供了更多的可能性。

在本系统中，利用了WebSocket来连接前端和后端，并且后端再使用WebSocket连接到Kali Linux。这种架构使得前端通过后端和Liunx连接，使整个系统具备了实时的通信能力，可以实现用户和liunx的实时操作交互和反馈。例如，在使用IP扫描的时候，前端只需要在Kali模块输入对应命令或者按钮就可以实时扫描服务器的liunx所能扫描的ip。

## 2.5 Mongo相关简介

MongoDB属于文档型非关系数据库。非关系型数据库不使用关系型数据库管理系统（RDBMS）原则，并且不将数据存储在表中，其模式不固定且具有非常简单的数据模型。（MongoDB)在功能和操作上与关系型数据库十分相似，是功能最丰富的非关系型数据库[4]。

MongoDB有着和表相似的集合，有数据库、使用B树实现索引。MongoDB使用的是在计量级JSON上进行了扩展的BSON保存无结构数据。MongoDB有着以下优点和特性：

1. 高性能，基于文档型的数据结构，在一些数据不需要严格规定的场景下，Mongo有着更好的性能，并且它提供与关系型数据库同样强大的查询功能，并支持索引。
2. 面向集合存储，对于Web面向广大用户的项目，方便存储复杂且多变的对象类型数据。
3. 实用性高，直接存取BSON更加灵活，不受初始化格式的限制可以插入复杂的数据类型，可以无需定义。

系统为面向大众使用的产品，并且需要存储大量复杂类型的数据如对象、数组等，并且需要提供快速查询，高性能支持提高用户的使用体验。

## 2.6 Security框架

Security是依赖于强大的过滤链的一个用户安全模块。它是一个自定义比较高的安全框架。提供了安全访问控制功能，使得开发人员在开发过程中可以少编写很多不必要的重复代码，同时也使得系统更加的安全[5]。

其主要特性和功能如下：

1. 认证： Spring Security 提供了多种认证方式，包括基于表单、HTTP Basic、HTTP Digest、LDAP、OAuth、OpenID Connect 等，可以根据应用程序的需求选择合适的认证方式。
2. 授权： 支持基于角色、基于资源、基于注解等多种授权方式，可以细粒度地控制用户对资源的访问权限。
3. 攻击防护（Attack Protection）： Spring Security 提供了防止常见安全攻击的功能，如跨站请求伪造防护、跨站脚本攻击防护、SQL 注入防护等，保护应用程序免受安全漏洞的影响。
4. 会话管理： 支持会话管理功能，包括会话超时、单一会话限制、并发会话控制等，可以管理用户会话的状态和安全性。
5. 密码加密： 提供密码加密和验证功能，可以使用各种加密算法对密码进行加密存储，并在认证过程中进行密码验证。

# 3 系统分析

## 3.1 可行性分析

### 3.1.1 技术可行性

本系统前端使用时下构建页面最流行的Vue.js框架，后端则采用了Java+SpringBoot的技术栈。

Vue 结合Springboot 及 MyBatis 是当下B/S开发模式的热门技术[6]，采用时兴的前后端分离开发的模式，在提高了开发灵活性和可维护性的同时，系统的可扩展性也大大提高，前后端增删其他功能变得相对独立与不耦合，一些功能的修改也变得非常灵活。

数据库方面，大部分数据存储在MySql数据库，一些热数据存储在Redis，文档型数据则存放在MongoDB，关系型数据库和非关系型数据库结合，极大程度上保证了数据的存放于拿取的性能保证。

版本选用方面，Vue版本采用了相对稳定的Vue2，后端考虑到JDK21发布时间，采用了SpringBoot3+JDK17开发，MySql使用了稳定的8.1.0版本，Redis使用了7.2.1版本，MongoDB使用了7.0.5版本。

综上，本系统从系统稳定流畅和开发难度方面技术性分析,选用了合适的技术栈和对应的版本进行开发，保证开发进度的正常推进。

### 3.1.2 经济可行性

本系统中使用的技术栈均为开源技术,不需要再额外付费。使用的开发工具vscode和idea均是社区版本而不是付费版本，数据库方面MySql、Redis和MongoDB也是免费可以的数据库。

本系统开发过程全由本人完成，毋须借由他人之手，因此不需额外付出其他费用，从经济角度上看本系统具有极高的经济可行性。

### 3.1.3 市场可行性

我国目前还处于网络空间安全人才的培养起步阶段。2015年之前，我国网络空间安全人才培养主要依托于信息安全相关的人才培养体系[7]。而近年来，网络空间安全备受重视，不少高校陆续开设了网络空间安全专业。而比起传统的计算机专业，网络安全的学习社区的构建远远比不上类似于开发、运维等方向，因此一个良好的可供实践与交流的社区能够极大的降低初学者入门的门槛，同时对于相关专业的大学生和新转行到安全方向的新手工程师来说，在专业基础仍然不扎实的时候也可以通过本系统的平台去实践与交流学习。

也就是说，面向初学的用户，本系统能够为他们解决一定程度上的问题，并且几乎没有同类社区的竞争，因此本系统具备良好的市场前景，存在较大的市场可行性。

## 3.2 需求分析

### 3.2.1 用户人群

本系统面向所有学习网络安全方向的初学者提供了一个交流与实践的平台。

### 3.2.2 用户需求描述

目前国内外的相关方向的社区对大多数初学者的使用门槛相对较高，并且更多是理论上的传递与交流，几乎找不到实践和交流结合的学习社区，因此用户需要一个提供自带攻击机和靶机的网站，这对初学者用户来说是大大利好他们学习的，并且网站额外携带的论坛功能能够方便他们和有经验者的交流，有助于初学者进步。

### 3.2.3 竞品分析

国内外主流的网络安全相关的学习社区有如下几个。

1. 产品定位及优势对比

表1 同类产品介绍

|  |  |  |
| --- | --- | --- |
|  | 产品定位 | 特色功能 |
| FreeBuf | FreeBuf 是一个专注于网络安全领域的社区平台，致力于为网络安全爱好者提供全面的安全信息和资源。 | FreeBuf提供及时、权威的安全新闻和资讯，涵盖漏洞报告、安全事件、攻击趋势等内容，帮助用户了解安全行业的最新动态。 |
| HackMyVM | HackMyVM 是一个虚拟机平台社区，旨在为安全研究人员和渗透测试者提供实验和学习的环境。 | HackMyVM让用户能够进行真实的渗透测试和安全演练，用户可以与其他安全从业者交流经验、分享技术。 |
| OWASP | OWASP（Open Web Application Security Project）是一个开源的、非营利性的国际性组织，致力于促进Web应用程序的安全性和安全开发实践。 | 开发了各种安全工具如：OWASP ZAP、OWASP Dependency-Check等，用于辅助开发者进行安全测试和漏洞扫描。提供了丰富的安全指南和文档，包括安全开发指南、渗透测试指南等，为开发者和安全从业者提供参考和指导。 |

（2）功能对比

表2 功能对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | 提供安全资讯 | 提供安全工具 | 实战环境 |
| FreeBuf | 支持 | 支持 | 不支持 |
| HackMyVM | 不支持 | 不支持 | 支持 |
| OWASP | 支持 | 支持 | 不支持 |

（5）总结：

表3 竞品总结

|  |  |  |
| --- | --- | --- |
| 产品 | 专注点 | 内容板块 |
| FreeBuf | 专注于社区属性，提供交流平台。 | 安全新闻、漏洞分析、安全工具分享 |
| HackMyVM | 专注于实践属性，提供虚拟实验环境，为安全研究人员和渗透测试者提供实践和学习的平台。 | 漏洞靶场资源提供、CTF挑战举办、学习资源分享 |
| OWASP | 专注于百科属性，提供各种安全工具分享和安全操作指南。 | OWASP Top 10（高危漏洞排行榜）、安全工具下载、安全指南分享 |

以小见大，市面上主流的安全社区完全没有结合安全资讯共享、安全工具提供和安全实战环境的平台，特别对于入行门槛高的网络安全领域，急需要一个低入门门槛和结合实践与交流的社区来改变现状。

### 3.2.4 需求概述

首先本系统是面对广大入门的网络安全方向的初学者和专业，应该提供更高可用和低门槛学习的能力，因此比起传统的工具与社区，本系统需要部署好实践学习需要的工具与环境，本系统的目标用户为初学者入门的初学者和乐于分享技术交流的专业人员，系统应该具备管理员对网站论坛和用户群体进行管理。

用户体验上应该具备以下条件：

1. 实践环境易于入门，简便操作
2. 论坛环境交流方便，操作符合逻辑
3. 整体页面布局整齐，样式偏现代化

功能上应该满足以下条件：

1. 提供实践操作功能
2. 提供论坛交流功能
3. 提供资源分享功能
4. 提供用户管理功能（管理员）
5. 提供实践教学功能

### 3.2.5 功能分析

根据需求分析可以得出，本系统核心为交流、实践和资源分享三个方向，其中交流方向通过构建论坛实现；实践的方向则是交由时下热门的网络安全工具Kali\_Liunx和常用的Metasploitable2 靶机环境来完成，资源分享的方向则是通过搭建了MinIO服务器来实现，核心的三个方向的功能面向所有用户，而管理员对应的功能则是通过动态路由附加。

面向用户的包括以下功能：  
 （1）首页论坛模块：用户在登录鉴权后默认跳转到首页论坛模块，论坛默认展示目前存在的帖子，以及发布帖子、浏览帖子的功能，通过上方搜索框可以通过关键字或者标题搜索帖子，同时用户可以通过全局存在的导航栏跳转到其他模块。  
 （2）密码加解密模块: 用户点击导航栏中的密码加解密进入密码加解密模块，模块提供一个操作界面，用户可以通过选择加密或者解密功能，在输入对的密钥，挑选对应的算法后完成对明文的加密或密文的解密。  
 （3）Kali-Liunx模块：用户点击导航栏中的“Kali-Liunx”进入Kali-Liunx模块，在本系统里，Kali-Liunx作为攻击机安装了绝大多数的工具，并且网卡做了桥接，用户可以在这里通过操作命令完成对两个靶机的侦测与攻击。  
 （4）Msf模块：用户点击导航栏中Msf可以进入Msf模块，在本系统里，特意挑选了Metasploitable2作为 靶机环境之一，作为靶机，Metasploitable2可以用于进行渗透测试训练和演练。  
 （5）个人中心模块：用户点击右上角头像的“个人信息”进入个人中心模块，可以查看个人信息和自己发布的帖子等。

（6）信息修改模块：用户点击导航栏的“修改信息”，可以进入信息修改模块，可以修改自己的昵称、邮箱、密码、手机号码等。

管理员额外附加以下功能：  
 （1）用户管理模块：可以对所有ROLE\_USER级别的用户的所有信息进行增删改查。

（2）帖子管理模块：可以对所有帖子进行增删改查。

（3）命令管理模块：可以对所有攻击机的按钮进行增删改查。

（4）资源管理模块：可以对所有资源进行增删改查。

## 3.3 系统用例图

用例是一种描述计算机系统如何工作的分析方法,其关键特点是提供给参与者有价值的反馈[8]。以下是本系统的用例图，从用户层面将用例分成了User和Admin两个部分，User用户首先可以操作登录、论坛、个人中心、资源中心、Liunx等功能，但更多是对和自己相关的信息的操作，比如发布自己的帖子、修改自己的信息等，在Liunx模块方面可以使用攻击机和靶机，在资源中心发布和下载资源，可以而Admin用户在享有User一切权利的基础上，拥有对一切数据的控制权限，包括修改任意用户的信息、管理任意用户的帖子，管理所有资源，同时还可以修改攻击机的便捷按钮。

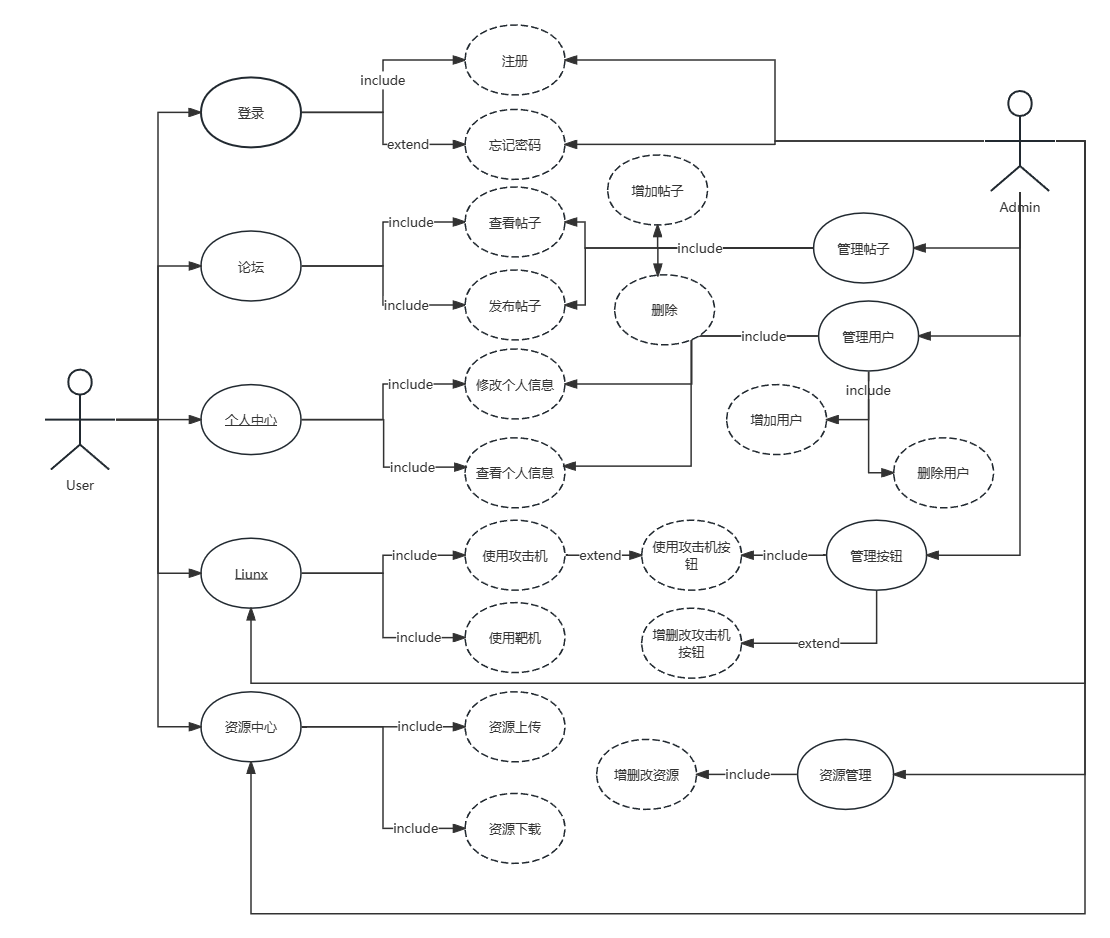


图1 系统用例图

## 3.4 数据流程图

数据流图语言是需求分析阶段最常用的一种图形语言[9]。它可以帮助人们理解系统内部数据如何流动、被处理和存储的过程。以下是本系统的顶层DFD数据流图，以用户为中心，包含本系统使用的所有数据库在内，展示了所有产生信息的活动的数据流向。

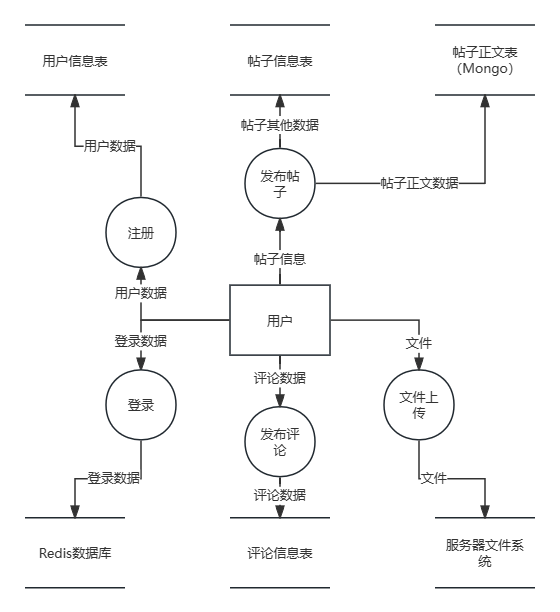


图2 顶层DFD图

## 3.5 数据字典

本系统所有数据字典如下所示：

（1）用户信息数据字典：

表4 用户信息数据字典

|  |  |
| --- | --- |
| 用户信息 | |
| 描述 | 所有用户的信息 |
| 数据结构 | 用户信息=用户ID+头像文件路径+用户名+昵称+密码+邮箱+手机号码+用户类型+用户角色 |
| 数据来源 | 任何人在本系统上进行注册 |
| 数据去处 | 用户信息表sys\_user |

（2）登录信息数据字典：

表5 登录信息数据字典

|  |  |
| --- | --- |
| 登录信息 | |
| 描述 | 所有登录系统者的信息 |
| 数据结构 | 登录系统者的信息=用户名+登录IP+凭证刷新时间+凭证过期时间 |
| 数据来源 | 用户登录 |
| 数据去处 | Redis数据库 |

（3）帖子信息数据字典：

表6 帖子信息数据字典

|  |  |
| --- | --- |
| 帖子信息 | |
| 描述 | 所有帖子的信息 |
| 数据结构 | 帖子信息=帖子ID+标题+帖子图片文件路径+帖子攥写用户ID+发帖时间 |
| 数据来源 | 用户发布帖子 |
| 数据去处 | 帖子信息表article |

（4）帖子正文信息数据字典：

表7 帖子正文数据字典

|  |  |
| --- | --- |
| 帖子正文信息 | |
| 描述 | 所有帖子正文的信息 |
| 数据结构 | 帖子正文信息=帖子正文ID+帖子ID+帖子正文 |
| 数据来源 | 用户发布帖子 |
| 数据去处 | 帖子正文信息表text(Mongo) |

（5）评论信息数据字典：

表8 评论信息数据字典

|  |  |
| --- | --- |
| 评论信息 | |
| 描述 | 所有评论信息 |
| 数据结构 | 评论信息=评论ID+用户ID+用户昵称+用户头像文件路径+被评论帖子ID+被评论评论ID+被回复用户ID+被回复用户名+评论正文+评论时间 |
| 数据来源 | 所有用户发布评论 |
| 数据去处 | 评论信息表comments |

（6）按钮信息数据字典：

表9 按钮信息数据字典

|  |  |
| --- | --- |
| 按钮信息 | |
| 描述 | 实践环境的按钮信息 |
| 数据结构 | 按钮信息=按钮ID+命令内容+创建时间+命令类型 |
| 数据来源 | 管理员创建命令按钮 |
| 数据去处 | 评论信息表comments |

（7）资源信息数据字典：

表10 资源信息数据字典

|  |  |
| --- | --- |
| 按钮信息 | |
| 描述 | 资源的详细信息 |
| 数据结构 | 资源信息=资源ID+上传id+资源详细描述+资源url+创建时间 |
| 数据来源 | 用户上传资源 |
| 数据去处 | 资源信息表resources |

## 3.6 系统类图

类图说明了软件类的规格说明和应用程序接口，与分析阶段中的概念模型不同，一个设计类图显示出了软件实体的定义而不是真实世界中的概念[10]。

本系统的包含了：

1. 评论实体类和评论传输类，评论实体类存放评论信息，评论传输类存放一篇文章的所有评论信息；
2. 用户实体类、登录实体类、用户数据类和登录传输类，其中用户实体类存放用户信息，登录实体类存放登录用户的信息，用户数据类存放用户不涉密的数据，登录传输类存放账号与密码信息；
3. 文章实体类和文章正文实体类，文章实体类存放除文章正文之外的所有文章信息，正文实体类存放了文章正文的数据以及对应的文章实体标志信息；
4. 连接信息类，存放了前端连接后端和后端连接Liunx的信息；
5. 资源信息类，存放了资源的url和创建信息。

实体类图与传输DTO类图如下：

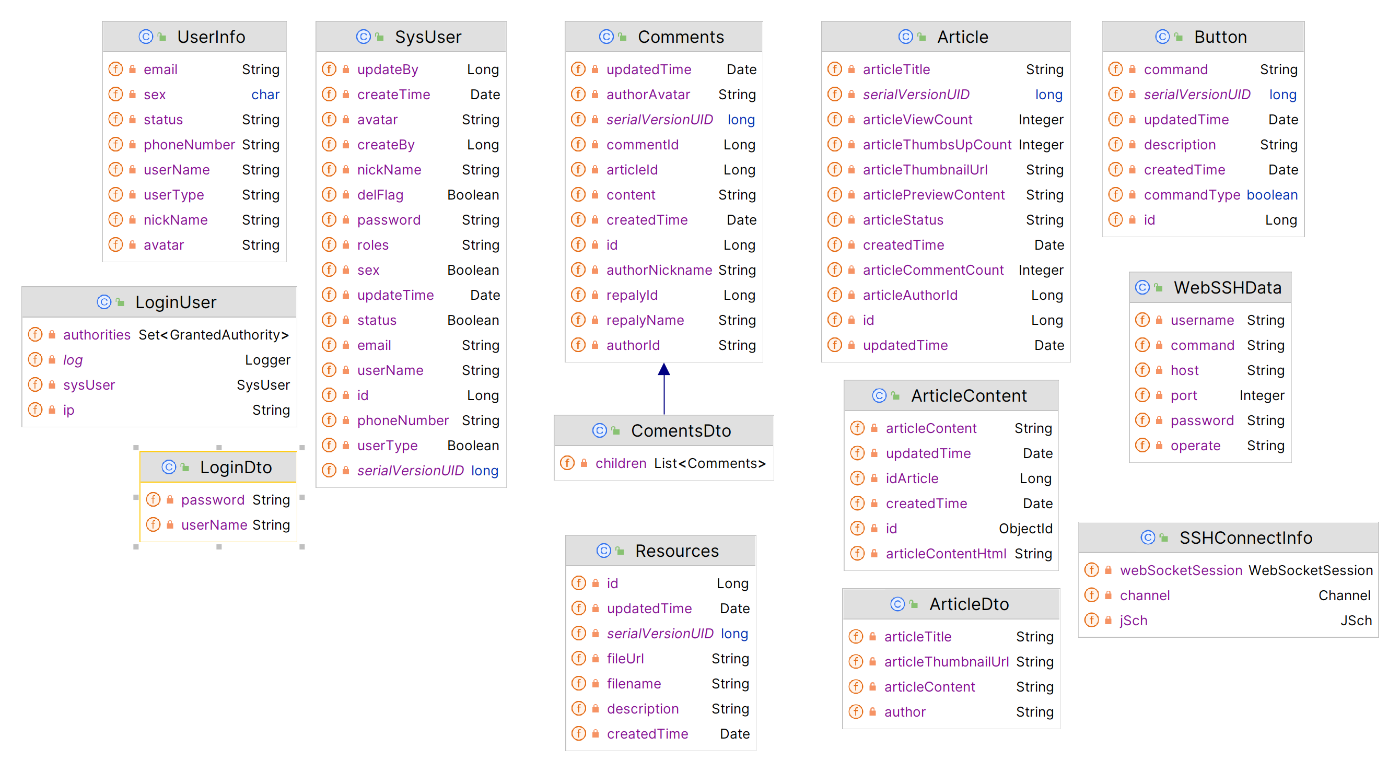


图3 系统类图

# 4 系统概要设计

## 4.1 系统设计原理

系统设计是对网站的整体规划，主要包括界面风格设计、导航设计、数据库设计等，这是网站开发的基础[11]。

明确本系统的目的是面向广大网络安全学习初学者的一个实践与交流的社区平台，从学生的角度出发，做出一个低门槛高可用的web系统。

## 4.2 系统开发环境

（1）硬件配置

处理器:AMD Ryzen 5 4500U with Radeon Graphics 2.38 GHz

机带:RAM 8.00 GB (7.40 GB 可用)

系统类型:64 位操作系统, 基于 x64 的处理器

（2）软件配置

开发IDE:Visual Studio Code、IntelliJ IDEA

数据库: IntelliJ IDEA附带的数据库工具（MySql\MongoDB\Redis）

软件工程绘图工具：ProcessOn/IDEA

## 4.3 系统整体架构

本系统可以分为三部分，分别是前端系统，后端系统，以及提供实践的虚拟机系统, 其中前端使用端口8080，后端使用端口80，虚拟机统一开放22端口作为SSH连接口。整体架构如下图所示

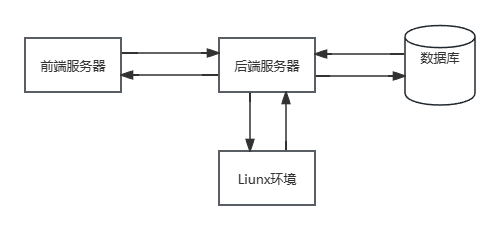


图4 系统整体架构图

## 4.4 系统功能结构设计

系统模块功能设计如图5所示：

1. 有登录注册模块，实现了用户注册功能，用户登录功能，密码重置功能；
2. 有Liunx模块，实现了攻击机功能，靶机功能；
3. 有资源中心模块，实现了资源上传功能，资源下载功能；
4. 有个人信息模块，实现了个人信息查看功能和修改功能；
5. 有论坛模块，实现了发帖和查看帖子功能，查看帖子后还可以发表评论；
6. 有用户管理模块，实现了修改用户信息和新增用户功能；
7. 有密码加解密模块，实现了密文解密和明文加密功能；

其详细功能设计图如下所示：

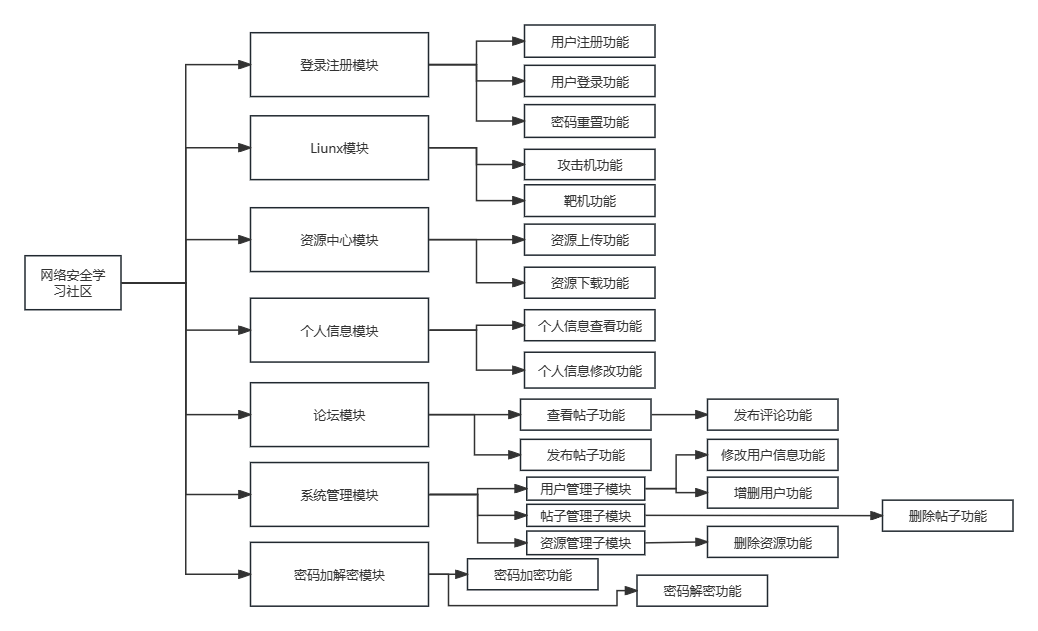


图5 功能架构

## 4.5 数据库设计

### 4.5.1 设计原则

好的数据库设计是从数据层面、安全层面以及性能层面三个层面完成。

在关系数据库中，一个数据库模式是由一组关系模式组成的，关系数据库设计的目标就是要从各种可能的关系模式组合中选取一组关系模式来构成一个数据库模式，使我们既不必存储不必要的重复信息，又可以方便地获取信息。一般而言，这样的数据库模式就是好的数据库模式[12]。

本系统的数据库设计从数据一致性、性能优化、安全性和易用性四个角度为目的设计，首先数据库命名风格高度统一，并且为了提高易用性，按照不同数据的访问热度存放在不同数据库，比如正常信息放在MySql数据库，高并发数据存放在Mongo数据库，访问频繁的小型数据存放在Redis数据库等。

而从实际开发角度来说，本系统减少了数据库的包含外键等的约束，转而在后端层面控制数据的输入，大大提高了开发效率，减少了数据库查询和写入的开销，提升了整个系统的性能，同时利用索引降低数据冗余并提高数据库的并发处理能力。

### 4.5.2 概念结构设计

本系统采用E-R图作为描述实体之间关系的数据模型，所谓 E-R 信息模型设计，就是指实体-关系模型设 计，它较为直观地反映了是实体与属性的内在联系与特点[13]。通过各种元素的组合和连接，展示了实体之间的关系，包括一对一、一对多、多对一和多对多等不同类型的关系。本系统的数据库共有6张表，分别是sys\_user（用户表）、article（文章表）、button（命令按钮表）、comments（评论表）、text（文章正文表）、resources（资源信息表）。本系统部分实体和数据库E-R图如下：

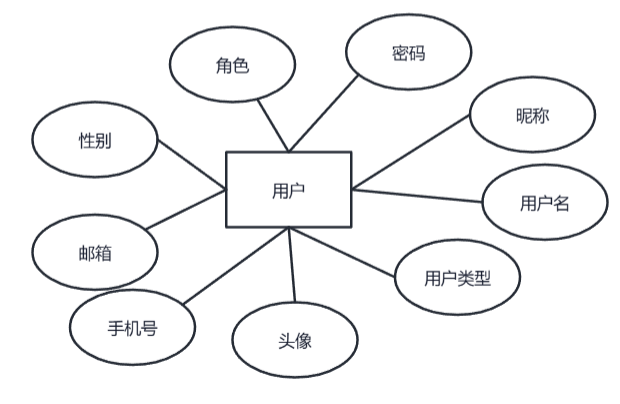


图6 用户实体

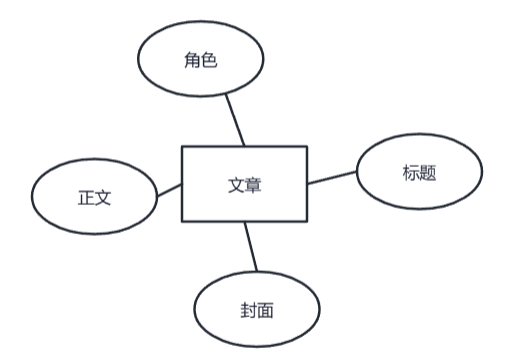


图7 文章实体

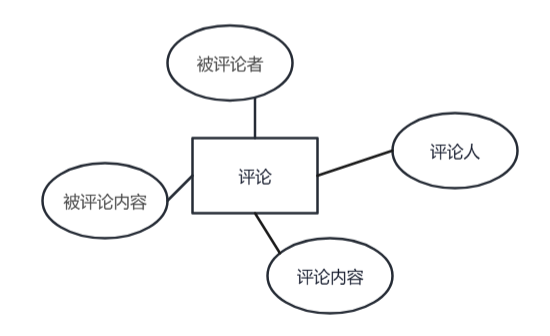


图8 评论实体

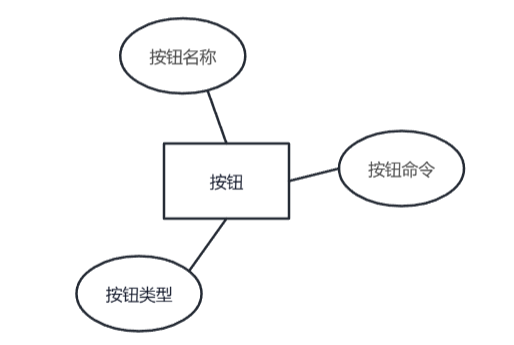


图9 按钮实体

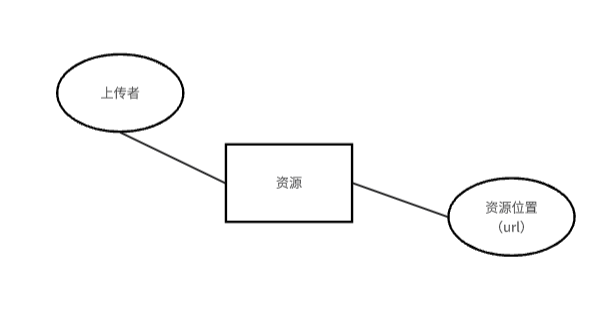


图10 资源实体

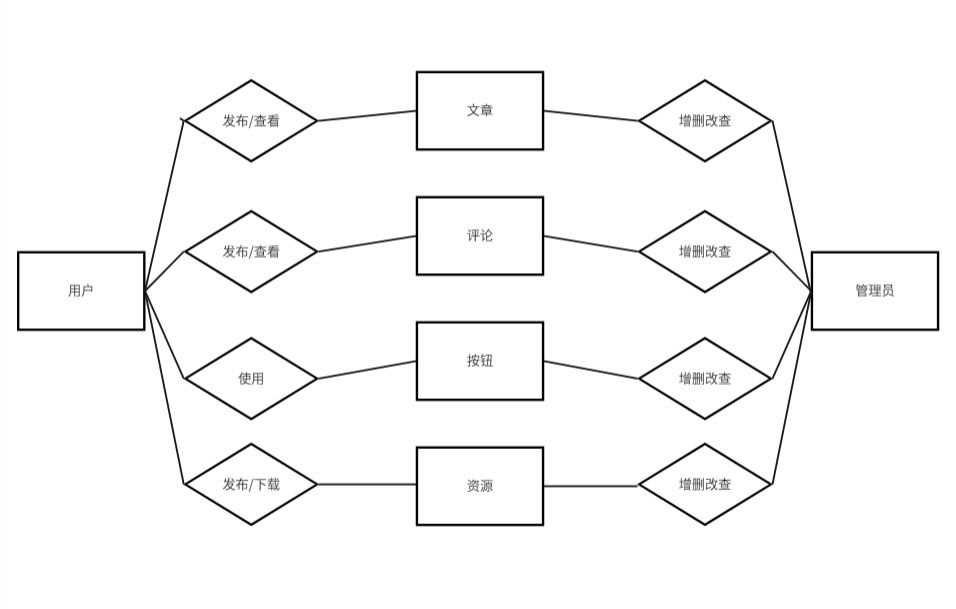


图11 系统E-R图

### 4.5.3 物理结构设计

网络安全学习社区的表和字段如下所示：

表11 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | user\_name | varchar(64) |  | 否 | 用户名 |
| 3 | nick\_name | varchar(64) |  | 否 | 昵称 |
| 4 | password | varchar(64) |  | 否 | 密码 |
| 5 | status | char |  | 是 | 账号状态 |
| 6 | email | varchar(64) |  | 是 | 邮箱 |
| 7 | phone\_number | varchar(32) |  | 是 | 手机号 |
| 8 | sex | char |  | 是 | 用户性别 |
| 9 | avatar | varchar(128) |  | 是 | 头像 |
| 10 | user\_type | char |  | 否 | 用户类型 |
| 11 | create\_by | bigint |  | 是 | 创建人的用户id |
| 12 | create\_time | datetime |  | 是 | 创建时间 |
| 13 | update\_by | bigint |  | 是 | 更新人 |
| 14 | update\_time | datetime |  | 是 | 更新时间 |
| 15 | del\_flag | int |  | 是 | 删除标志 |
| 16 | roles | varchar(255) |  | 是 | 角色 |

表12 文章表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | article\_title | varchar(128) |  | 是 | 文章标题 |
| 3 | article\_thumbnail\_url | varchar(128) |  | 是 | 文章缩略图地址 |
| 4 | article\_author\_id | bigint |  | 是 | 文章作者id |
| 5 | article\_view\_count | int |  | 是 | 浏览总数 |
| 6 | article\_preview\_content | varchar(256) |  | 是 | 预览内容 |
| 7 | article\_comment\_count | int |  | 是 | 评论总数 |
| 8 | created\_time | datetime |  | 是 | 创建时间 |
| 9 | updated\_time | datetime |  | 是 | 更新时间 |
| 10 | article\_status | char |  | 是 | 文章状态 |
| 11 | article\_thumbs\_up\_count | int |  | 是 | 点赞总数 |

表13 评论表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | content | varchar(255) |  | 是 | 评论内容 |
| 3 | author\_nickName | varchar(128) |  | 是 | 评论者昵称 |
| 4 | author\_id | varchar(128) |  | 是 | 评论者id |
| 5 | author\_avatar | varchar(128) |  | 是 | 评论者头像链接 |
| 6 | comment\_id | bigint |  | 是 | 父评论id |
| 7 | article\_id | bigint |  | 是 | 文章id |
| 8 | repaly\_id | bigint |  | 是 | 回复的评论id |
| 9 | created\_time | datetime |  | 是 | 创建时间 |
| 10 | updated\_time | datetime |  | 是 | 更新时间 |
| 11 | repaly\_name | varchar(10) |  | 是 | 回复的user |

表14 命令按钮表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | command | varchar(255) |  | 是 | 按键对应命令 |
| 3 | description | varchar(128) |  | 是 | 按键描述 |
| 4 | created\_time | datetime |  | 是 | 创建时间 |
| 5 | updated\_time | datetime |  | 是 | 更新时间 |
| 6 | command\_type | tinyint(1) |  | 是 | 命令类型 |

表15 文章正文表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | ObjectId | 是 | 否 | MongoDB 主键类型 |
| 2 | id\_article | Long |  | 是 | 整数类型 |
| 3 | article\_content | String |  | 是 | 字符串类型 |
| 4 | article\_content\_html | String |  | 是 | 字符串类型 |
| 5 | created\_time | Date |  | 是 | 日期时间类型 |

表16 资源信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | fileName | varchar(255) |  | 是 | 资源文件名 |
| 3 | author\_id | varchar(128) |  | 是 | 上传者id |
| 4 | description | varchar(255) |  | 是 | 资源描述 |
| 5 | created\_time | datetime |  | 是 | 创建时间 |
| 6 | updated\_time | datetime |  | 是 | 更新时间 |
| 7 | file\_url | varchar(255) |  | 是 | 资源文件URL |

# 5 系统详细设计

## 5.1 系统时序图

时序图表达特定场景中的对象间的通信关系,主要是显示完成某个特定方案时所涉及到的对象及方法调用的执行顺序[14]，以下介绍系统主要功能时序图。

以下是登录时序图，用户执行登录后将被获取ip，将登录IP写进token加密，并且将用户数据和token写进Redis里面，如果上述步骤执行无误将会返回统一的成功RestResult，否则返回失败的RestResult。

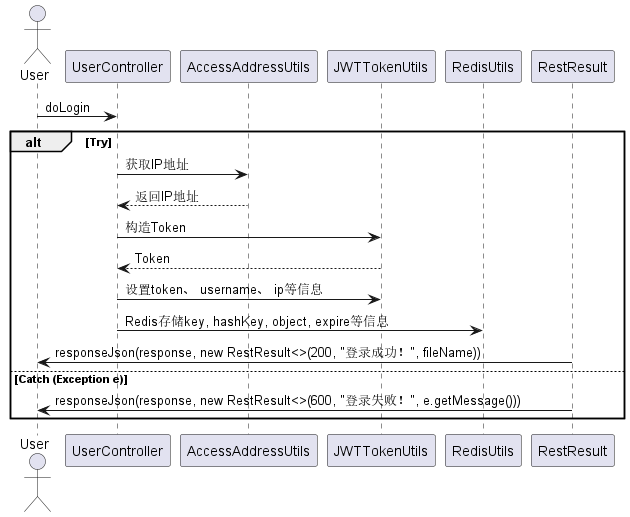


图12 系统登录时序图

以下是发帖时序图，用户发起请求之后，将直接创造一个文章信息对象和一个正文对象，通过service层将两个对象传到MySql数据库和Mongo数据库，期间全程开启事务，如果上述步骤执行无误将会create成功的RestResult并通过responeJson返回，状态码200，Message为“发帖成功”，数据为空，否则捕获异常，事务回滚数据并通过responeJson返回create的失败的RestResult，状态码600，Message为“发帖失败”，数据为异常的详细信息。

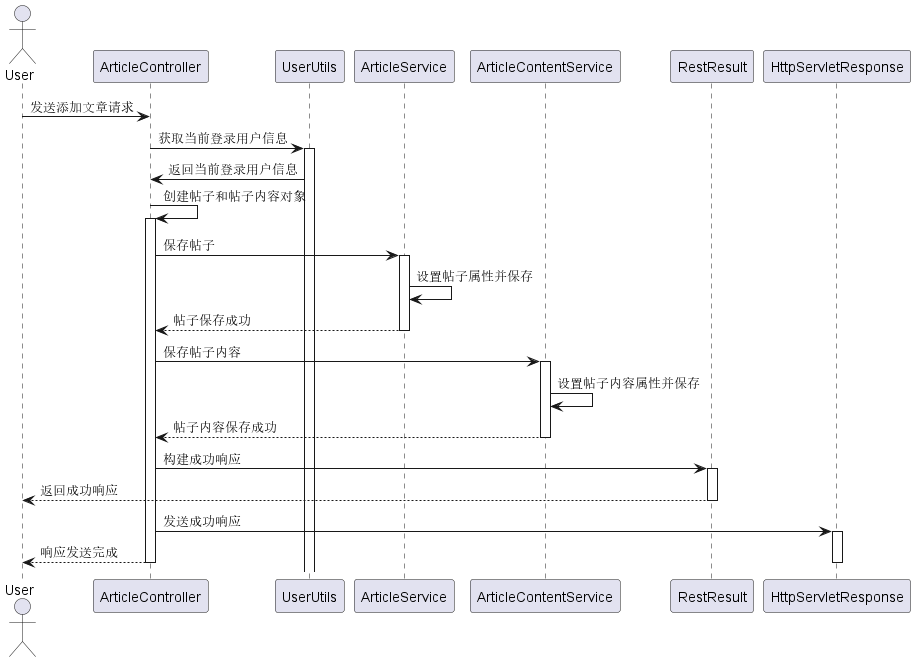


图13 发帖时序图

以下是评论时序图，用户发起请求后，将直接获取当前用户信息对应的系统信息作为作者信息填充进评论实体类，再将评论存进数据库，通过构建成功的RestResult信息返回给用户。

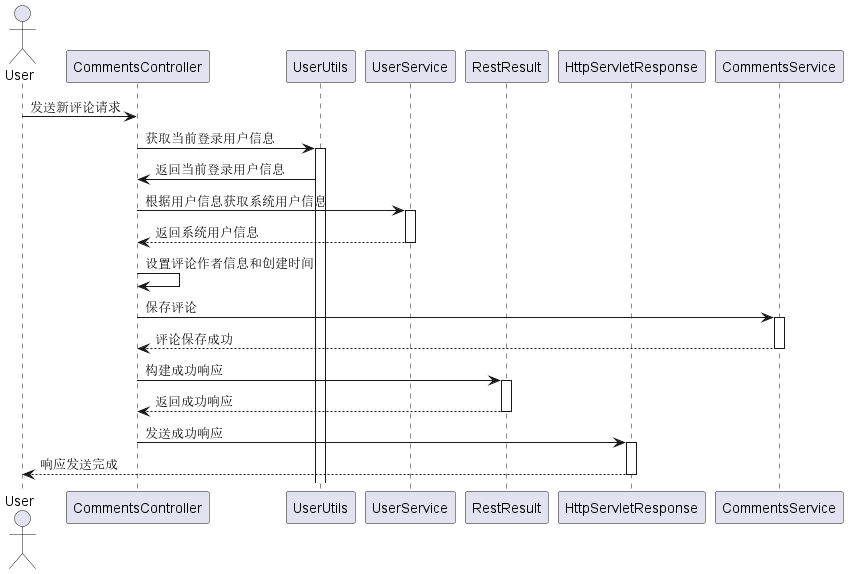


图14 评论时序图

以下是文件上传时序图，用户发起请求后，FileController将用户携带的文件的文件名取出，通过获取文件拓展名保证文件类型不被更改，再将文件上传到MinIO服务器，将返回的url返回给前端回显，状态码200，全程catch异常，如果异常出现，则携带异常信息返回，状态码600

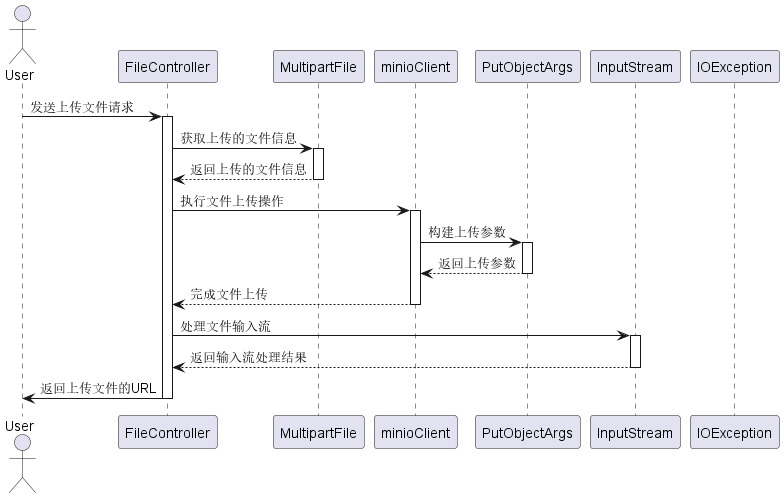


图15 文件上传时序图

以下是处理liunx命令的时序图，用户在前端通过WebSocket发送命令后，后端接受讯息后会将Json讯息转化成WebSSHData对象，通过WebSocketSessoion获取用户名，接着判断本次操作是命令操作还是连接虚拟机操作，再获取连接信息执行连接或者命令操作，最后则是关闭会话（WebSocket不通过RestResult通信）。

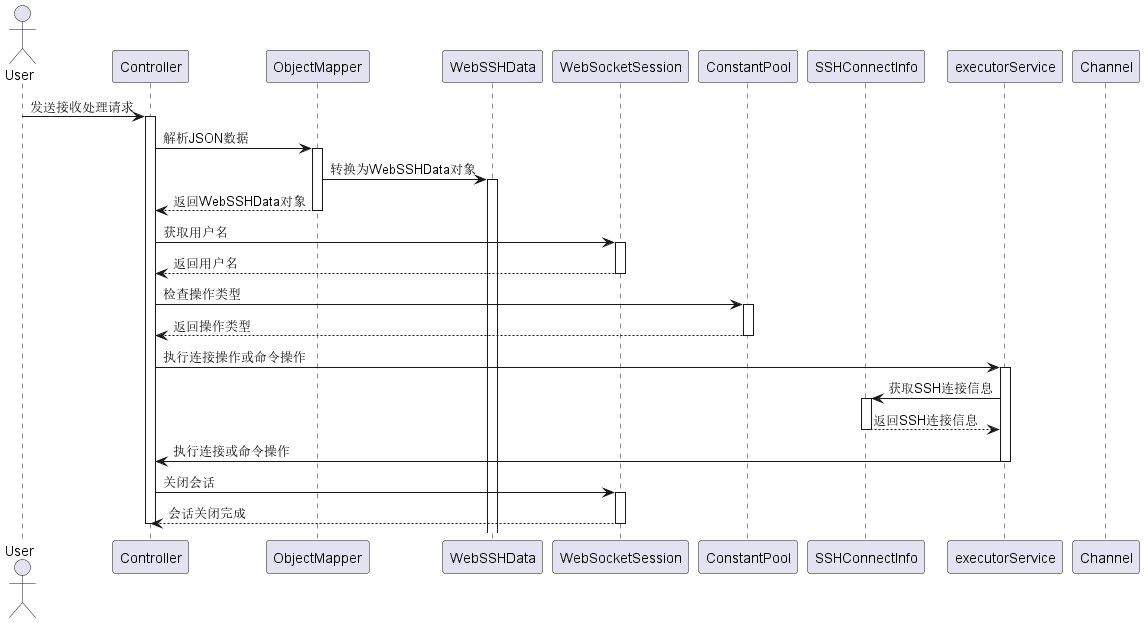


图16 liunx命令发送时序图

## 5.2 系统流程图

图16为系统流程图。用户在浏览器键入网址将直接鉴权，若登录将直接进入网址对应页面，否则一律重定向至登录页面，输入账号密码后正常登录跳转到首页，首页是本系统的论坛模块，通过论坛模块可以完成对帖子的查看与发布，点击任意帖子可以进入对应的正文页面，可以在帖子对正文以及正文的评论进行评论。

在登录成功后的任意页面，可以通过点击左侧导航栏进入不同的模块，其中进入MSF靶机模块可以对靶机进行操作，进入Kali攻击机模块可以使用攻击机进行操作，进入个人中心页面可以进行对个人信息修改以及退出登录等

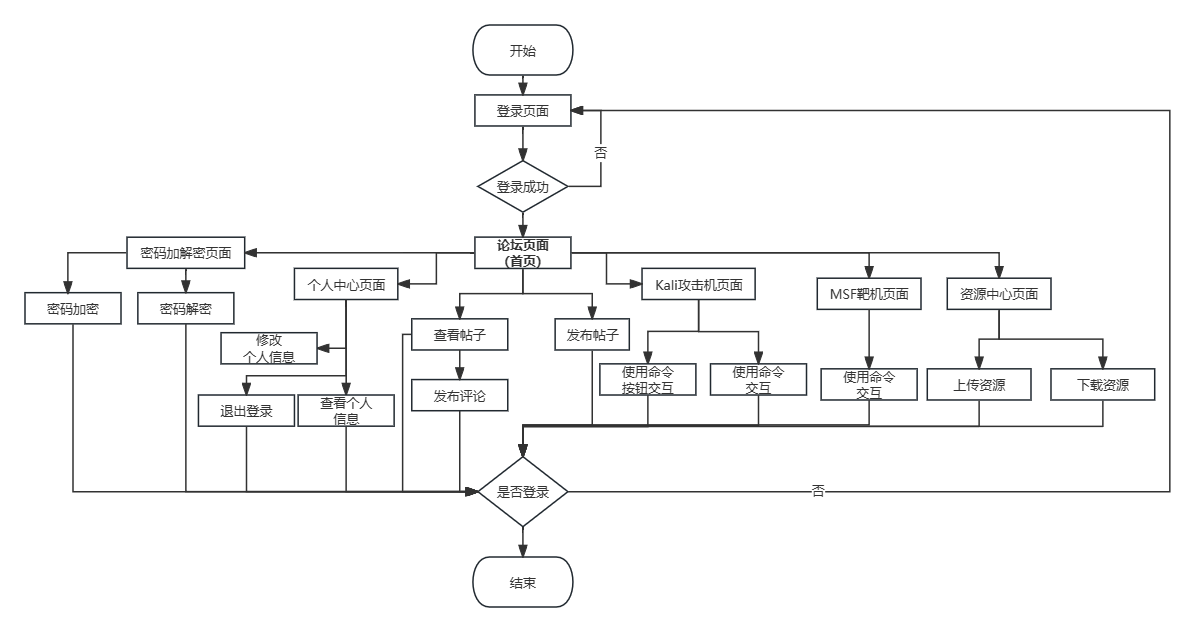


图17 系统总流程图

## 5.3 系统讲解

### 5.3.1 论坛模块

用户登录后默认跳转到论坛界面，如图19所示。论坛帖子使用了MarkDown语法展示。论坛界面可以上传帖子与浏览所有已上传帖子。点击上传帖子后，用户进入帖子正文页面，如图20所示。点击任意一个帖子，用户进入帖子预览界面，如图21 所示。用户可以在此页面发表与查看评论。论坛界面的导航栏全局显示，其中非管理员不显示且不开放用户管理等管理员功能的路由。

用户进入论坛模块的同时，前端页面会在渲染完成前向后端请求帖子数据并渲染，选中对应帖子进入对应帖子页面将向后端发送两个请求，第一个请求是向后端通过向MongoDB查询正文数据，第二个请求是向后端拿取评论数据，评论数据分为三级，文章评论、子评论以及回复评论，后端在查询评论后通过stream流完成子评论与回复评论的查询返回给前端。涉及MongoDB的帖子正文部分的增删改查使用MongoTemplate、Query条件查询器以及Criteria条件构造器完成。

点击发布帖子按钮可以进入帖子发布页面，帖子页面使用了markdown富文本编辑器来编辑帖子正文，上传封面后按下发布按钮可以发布帖子。



图18 论坛模块与导航栏



图19 帖子页面



图20 发帖预览页面

### 5.3.2 攻击机/靶机模块

攻击机/靶机模块是整个系统的核心模块，系统为用户提供攻击机和靶机，靶机页面全凭命令行交互，攻击机页面额外增加了命令按钮，通过向后端请求按钮列表生成按钮，点击按钮向后台Liunx发送命令。

前端使用Xtrem.js作为样式构建界面来交互，完善了输入，删除字符，回车发送的操作逻辑，同时，为了保持和后端的长连接通信，前端选用了WebSocket和后端进行通信，而在后端层面，为了和Kali攻击机通信，使用了JSCH框架结合WebSocket进行通信，这样就完成了前端——后端——Liunx的通信链。

由于后端使用了Security配置安全访问限制，因此，WebSocket的请求将会被Security拦截。为了解决这个问题，后端构造了WebSocket请求前置拦截器与过滤器，其中，过滤器负责将请求携带的token正常化，以便于后面的JWT过滤器正常工作获取用户信息，拦截器负责截取所有WebSocket的请求转到Handler处理器，整个请求链图如下：



图21 WebSocket请求过滤链图

由于整个系统的HTTP请求都采用了Controller-Services-Mapper的架构开发，为了统一开发风格，WebSocket请求也采用类似的Handler-Services-IO流的架构。这里的Handler处理器可以同等视作Controller，业务逻辑封装在WebSSHService里面，而对liunx的详细操作则有IO流结合Jsch框架实现。

前端发送的命令操作通过Security鉴权后将由Handler处理，，并且，由于WebSocket是长链接，可能出现应答等待等各种情况，因此设置了线程池分配线程来完成异步请求。

除了第一次链接需要初始化相关信息外，接下来所有通信与回调都通过线程异步处理与等待，并通过IO流传输数据。

前端界面如下图：

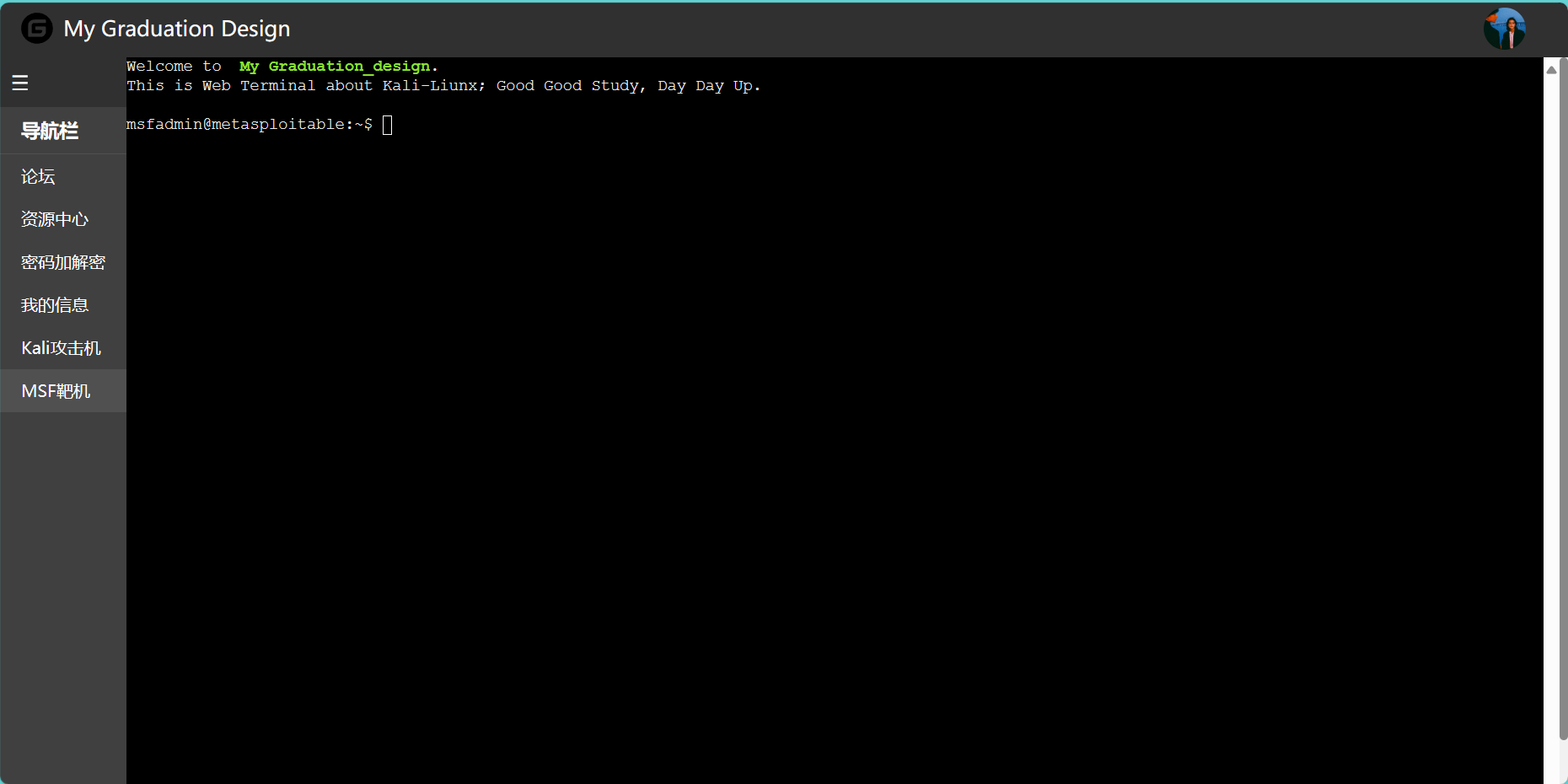


图22 靶机详情

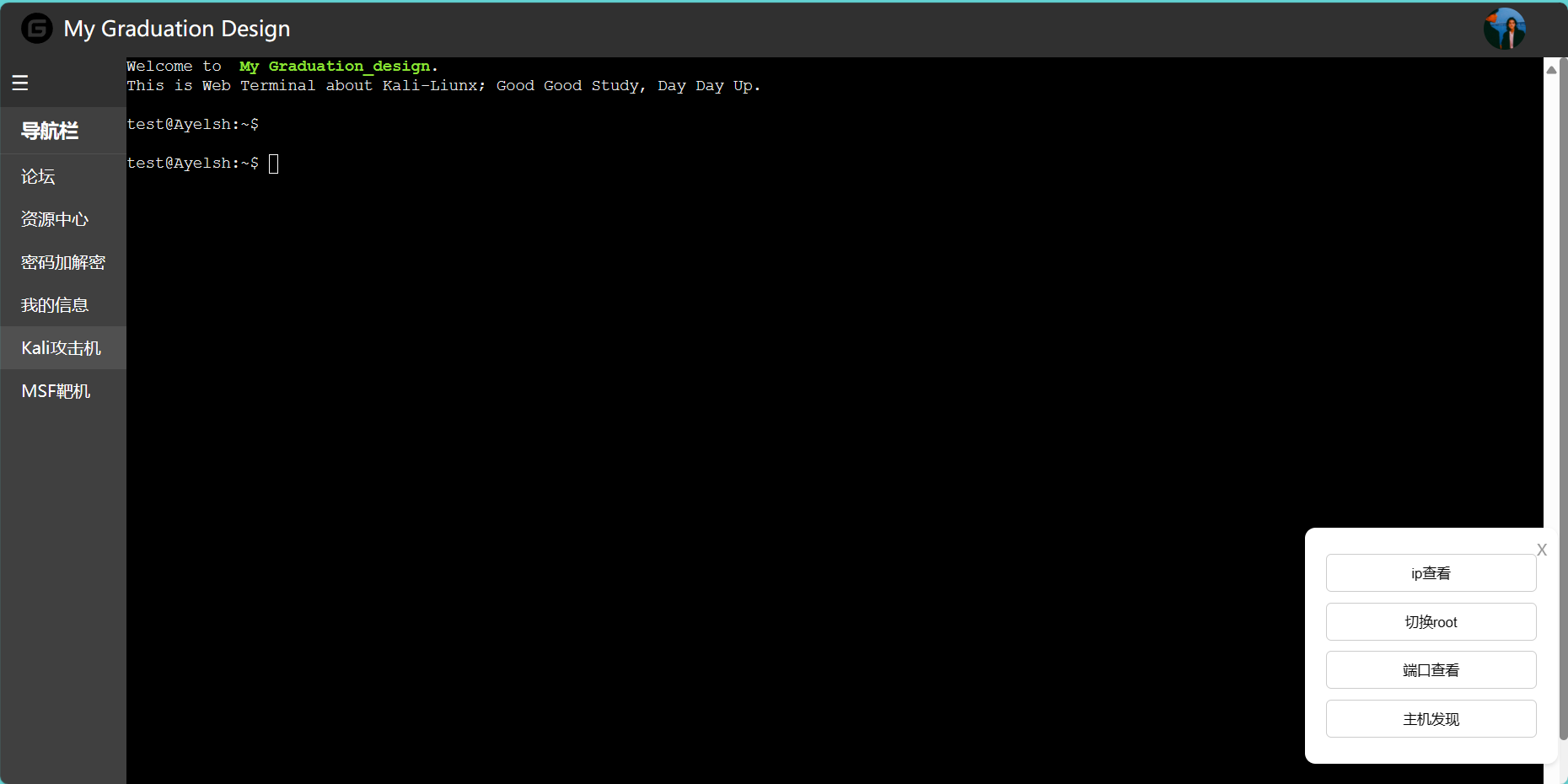


图23 攻击机详情

用户进入攻击机/靶机模块，在攻击机模块可以使用按钮去调用一些命令，分别有切换root用户、主机发现、端口查看。

以下是按钮调用图:

调用切换root:

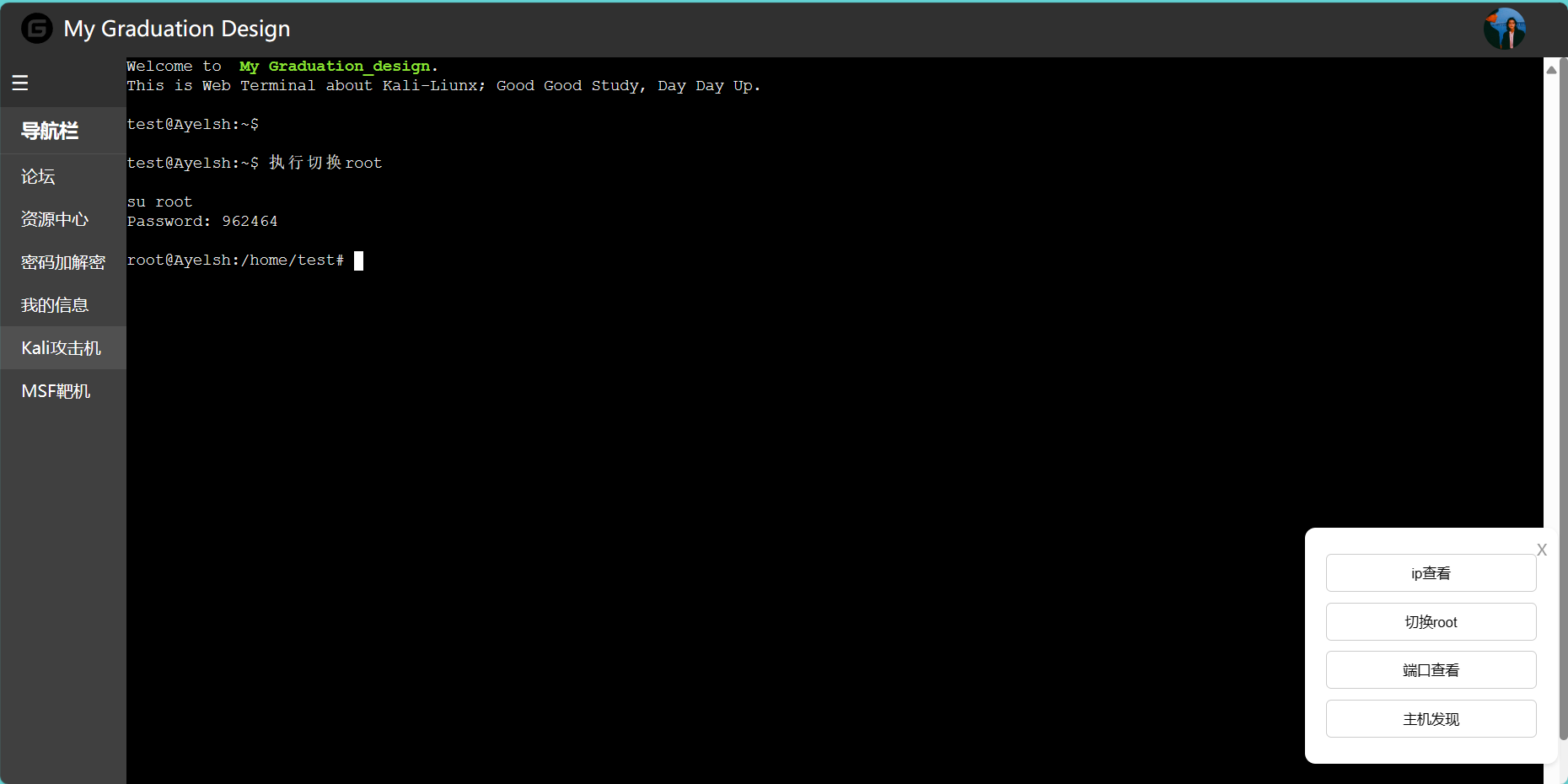


图24 调用切换root按钮

调用主机发现：

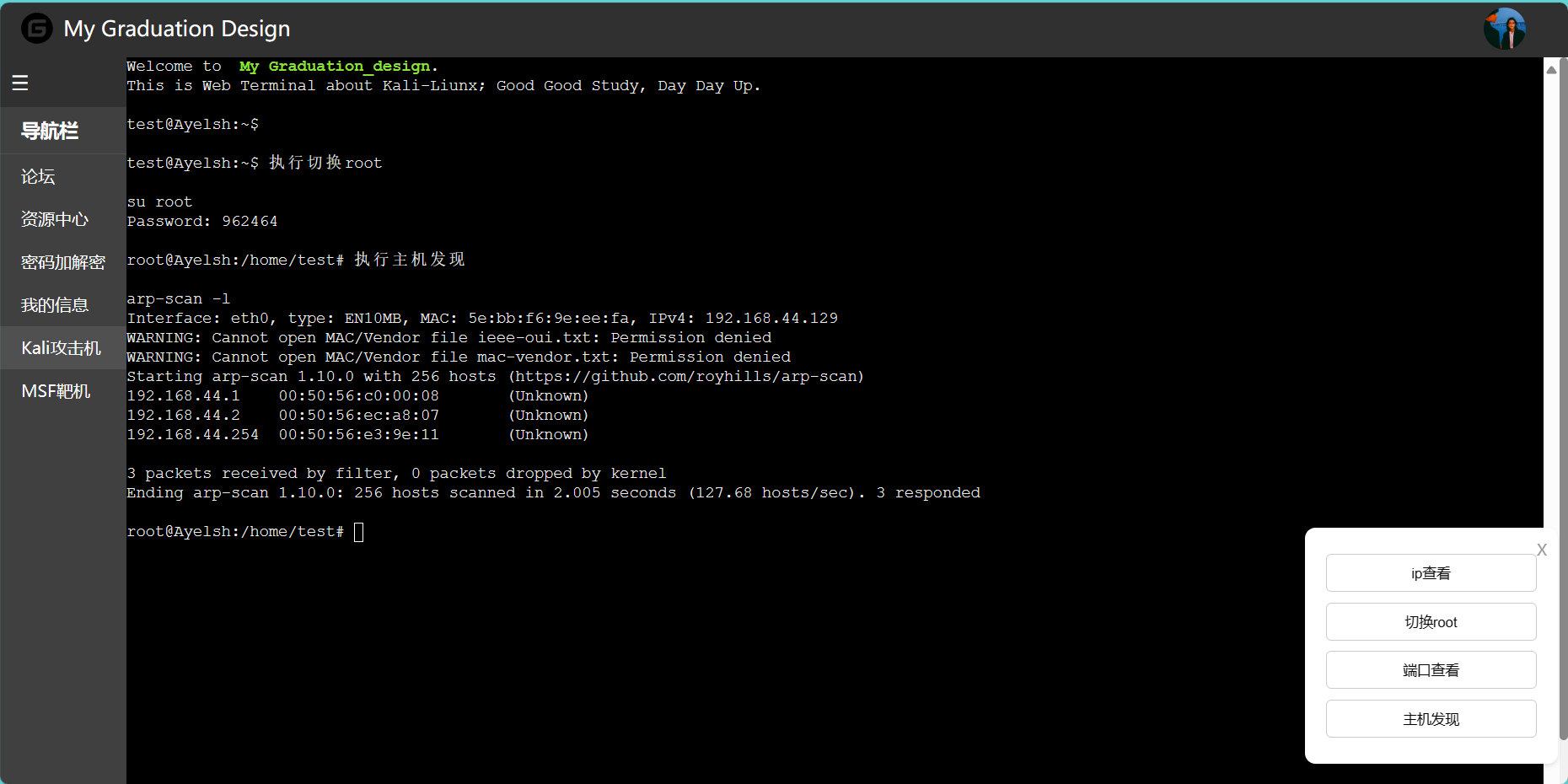


图25 调用主机发现按钮

调用端口查看的时候会提示输入IP地址：



图26 调用端口查看按钮

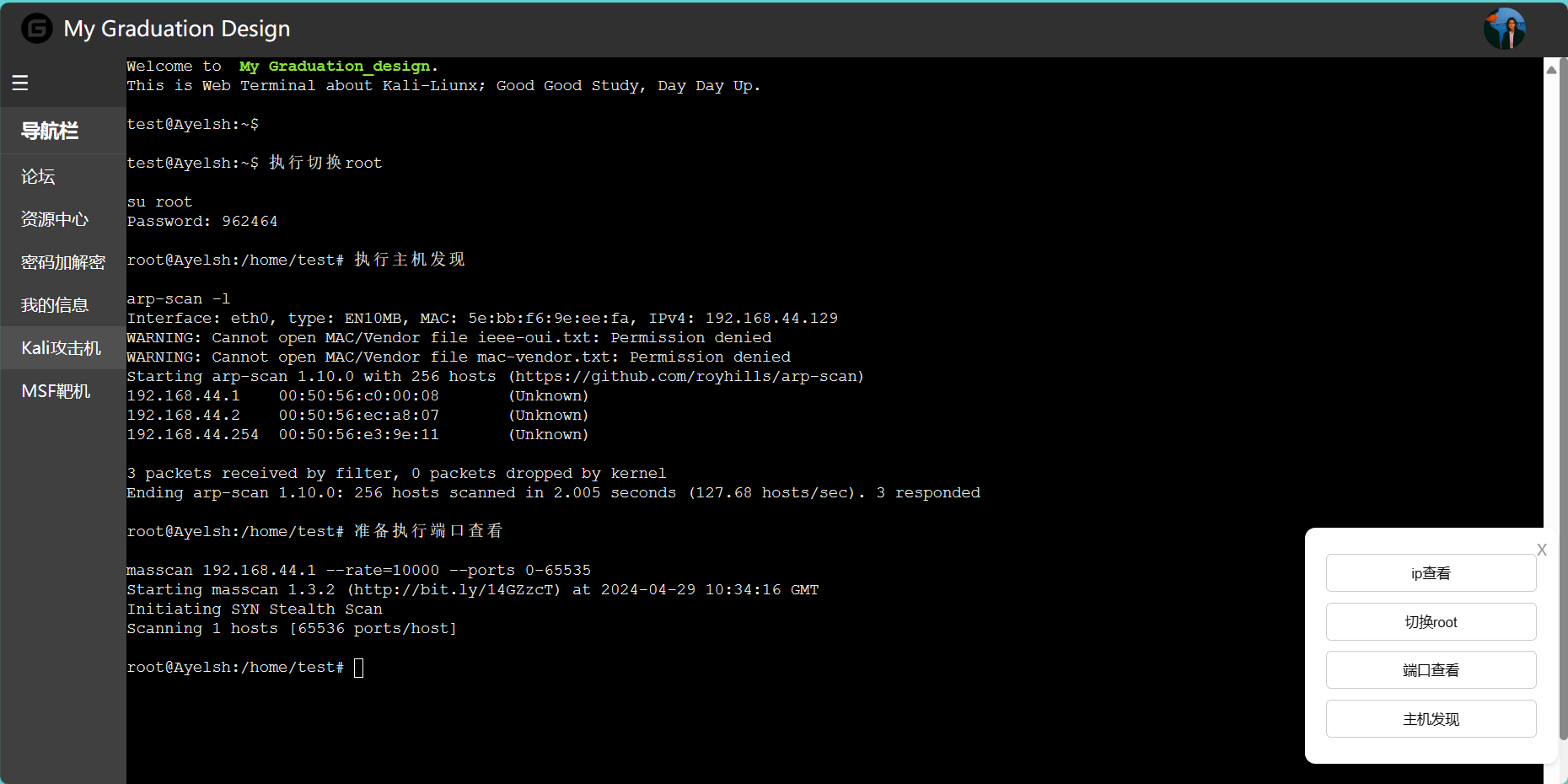


图27 端口查看按钮调用结果

### 5.3.3 资源中心模块

如图所示，所有用户可以进入资源中心模块发布资源和下载资源，所有资源文件均存储在后台搭建的Minio服务器，资源路径则以url的形式存储在后端数据库，用户通过后端拿到资源url的时候可以直接下载，用户在发布资源的时候需要详细描述资源的用处。

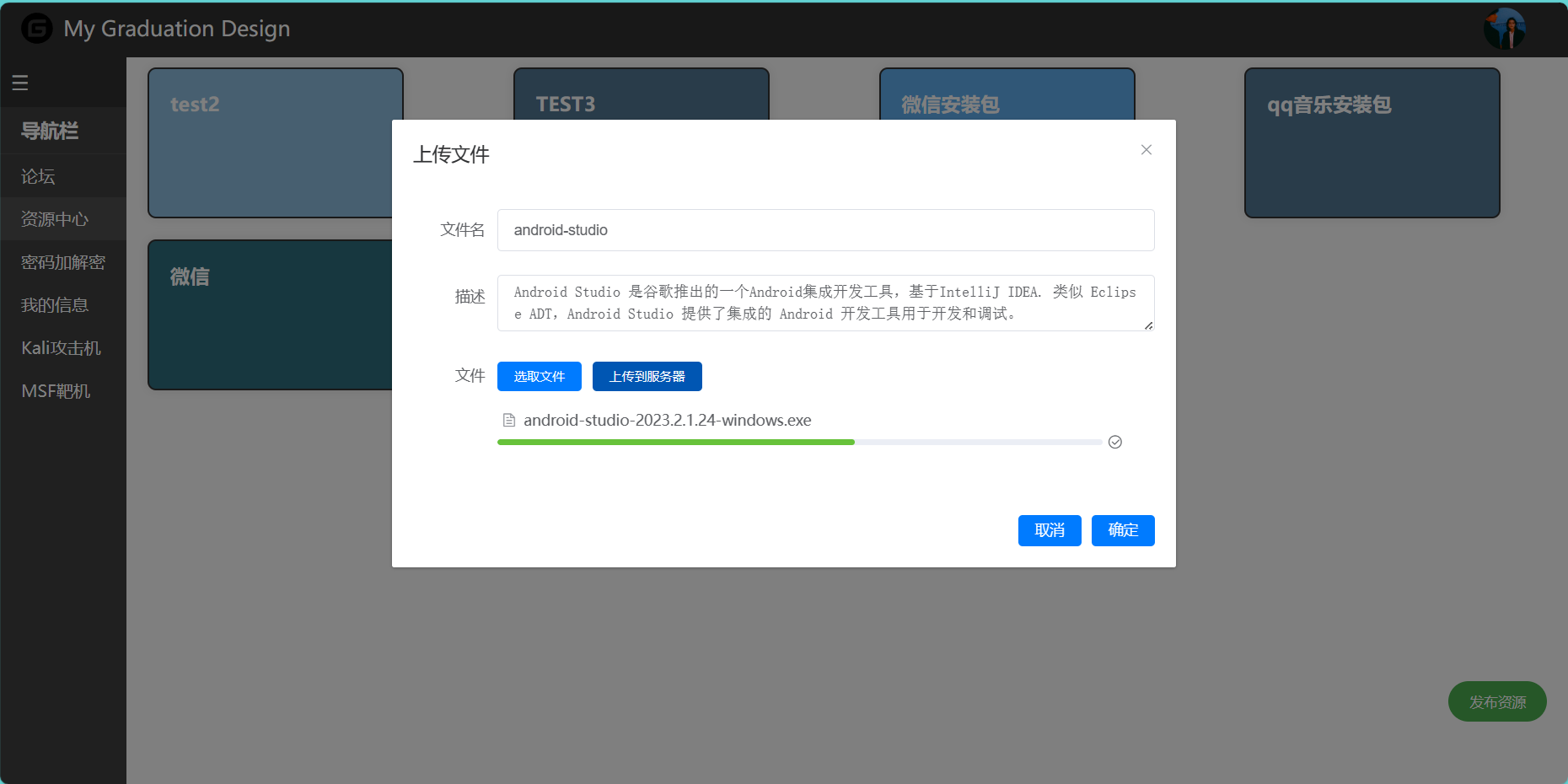


图28 资源发布页面

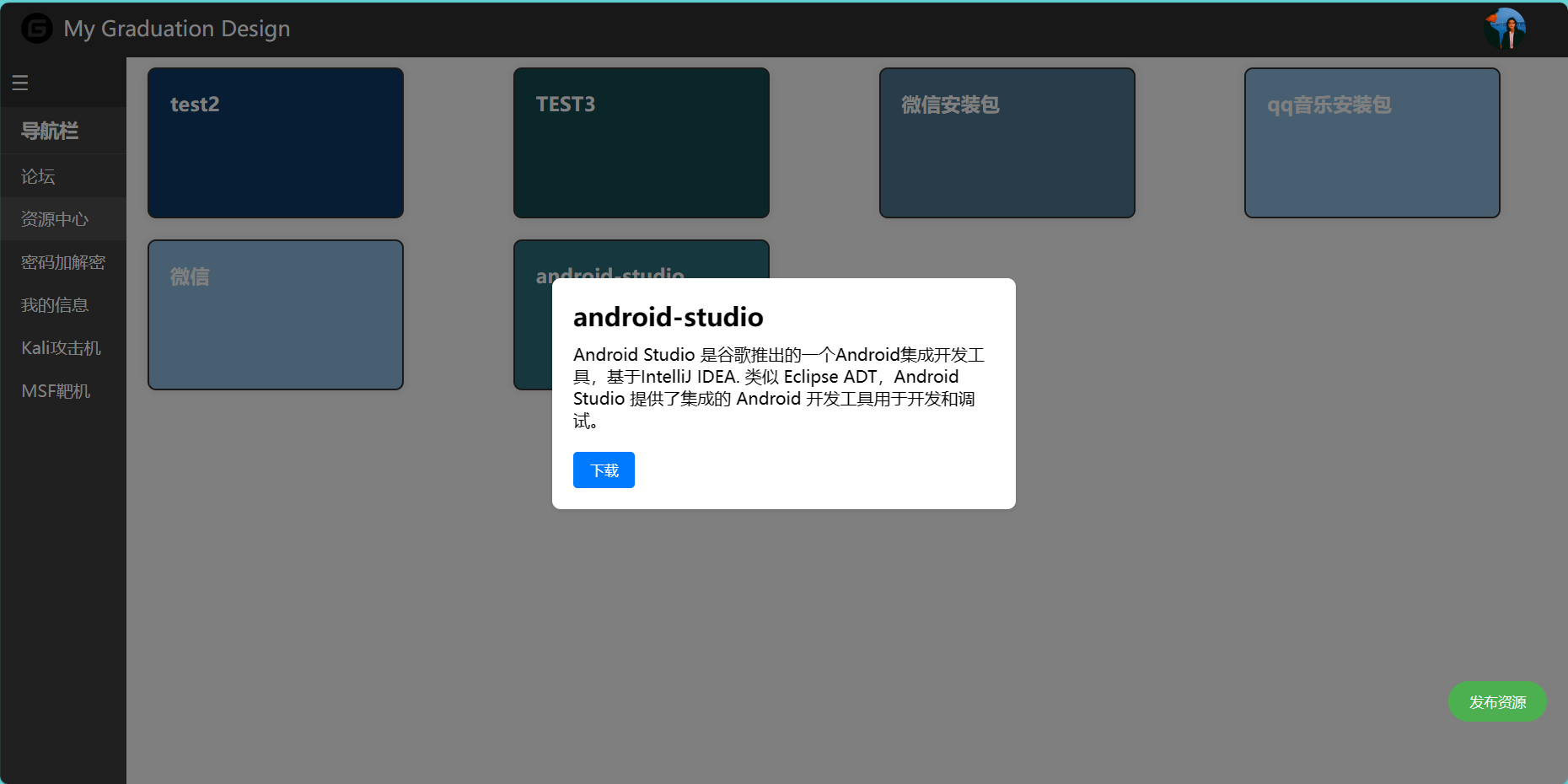
  

图29 资源下载页面

### 5.3.4 密码加解密模块

通过导航栏点击“密码加解密”，可以跳转到密码加解密界面，通过选择栏可以选择加解密以及加解密的算法。密码加解密通过前端的crypto-js库完成，界面展示如下：

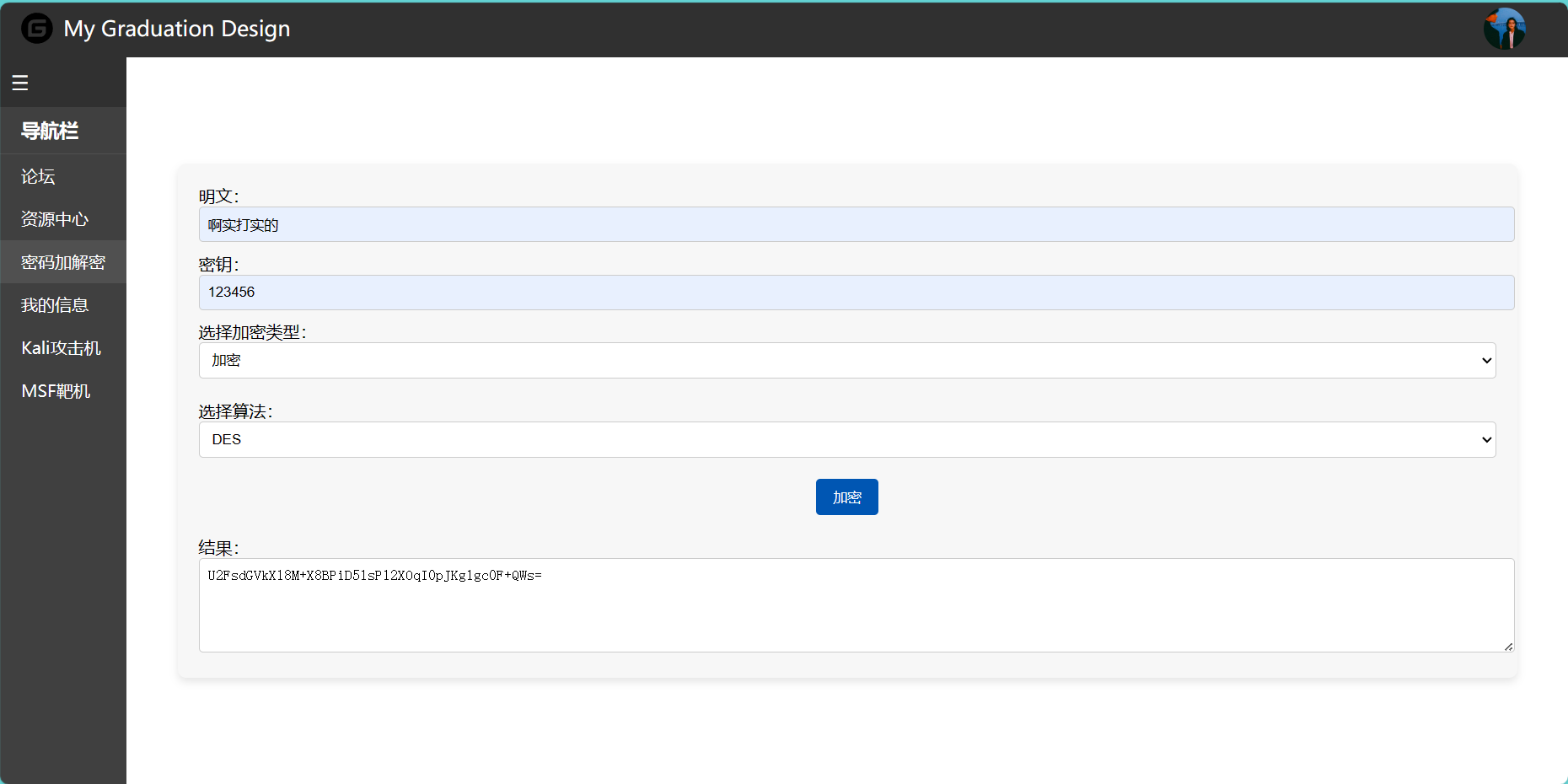


图30 密码加解密页面

### 5.3.5 个人信息模块

用户可以在导航栏“个人信息”里查看自己的个人信息，也可以在右上角“修改我的信息”选项进行修改，同时在右上角头像处还可以执行退出登录等操作。

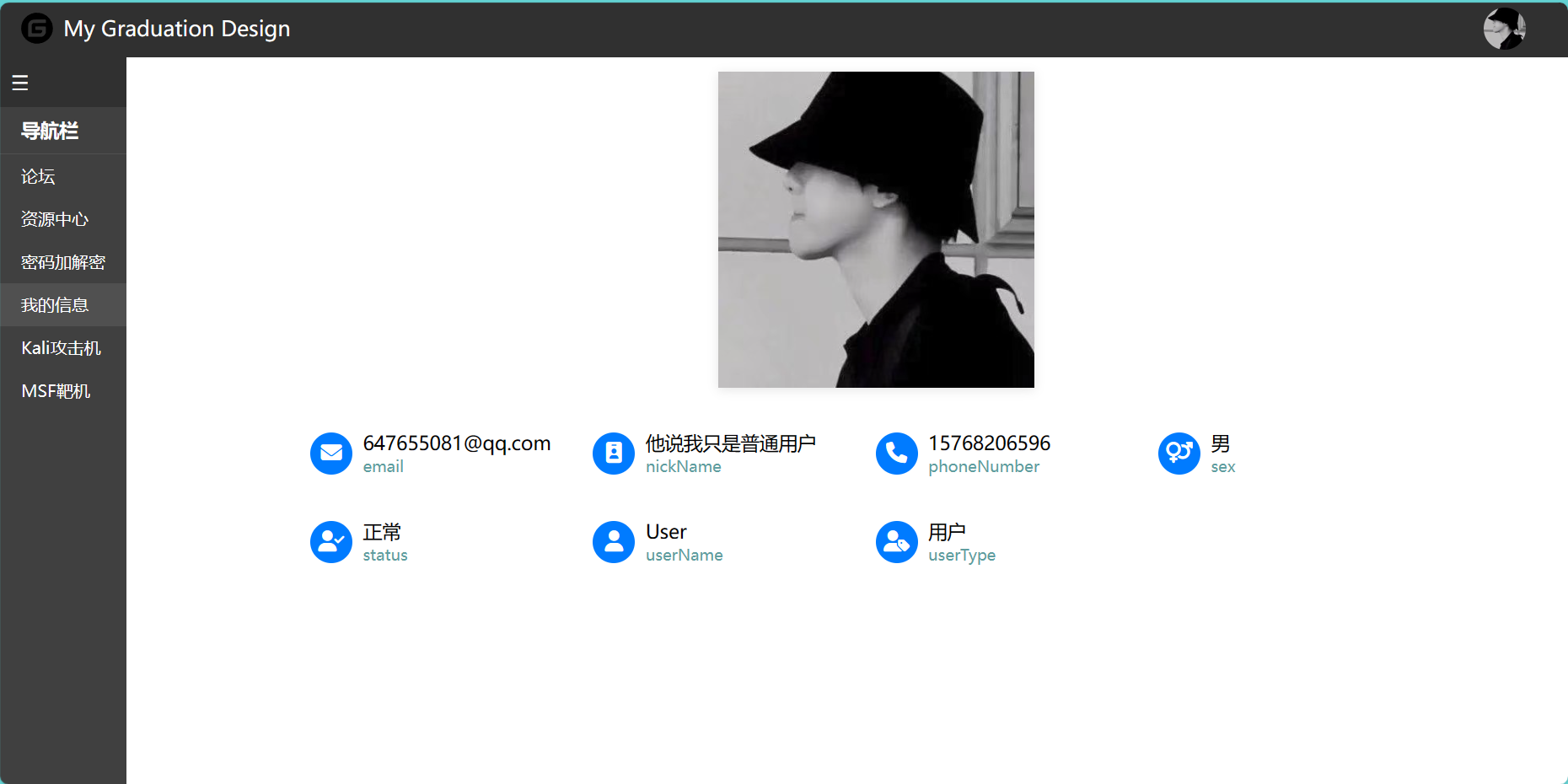


图31 查看个人信息

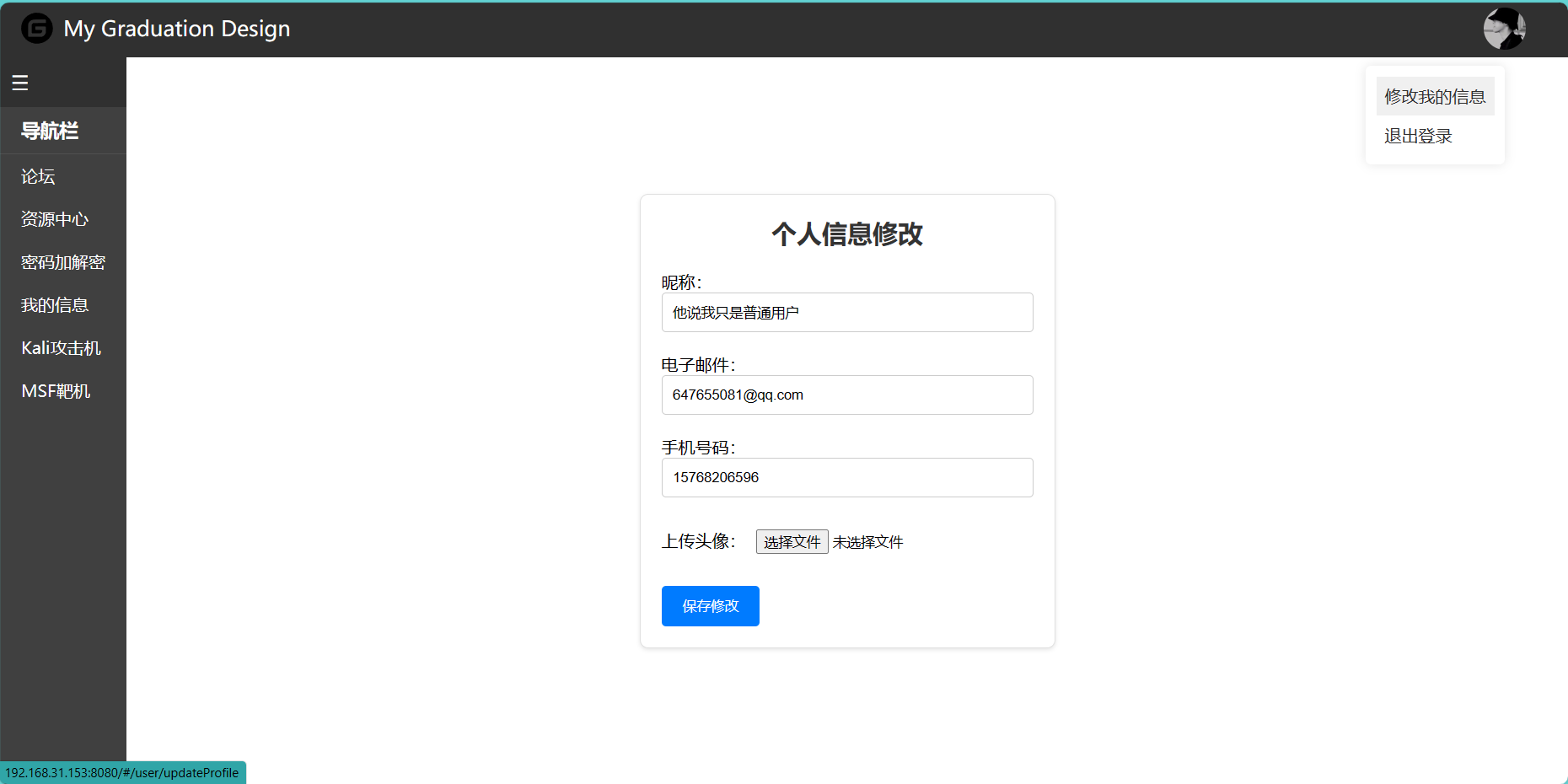


图32 修改个人信息

### 5.3.6 系统管理模块

系统管理模块是管理员才有的功能，前端在登录完成后会通过addRouter方法完成动态路由添加，并在右上角增加用户管理、帖子管理和资源管理选项，普通用户没有系统管理的三个选项也无法通过URL或者其他方法访问与操作。

通过请求后端，管理员（唯一）可以

1. 在用户管理界面对除自己外的其他用户进行信息修改、删除用户、增加用户等；
2. 在帖子管理页面删改帖子；
3. 在资源管理页面删改资源；



图33 管理员的管理选项

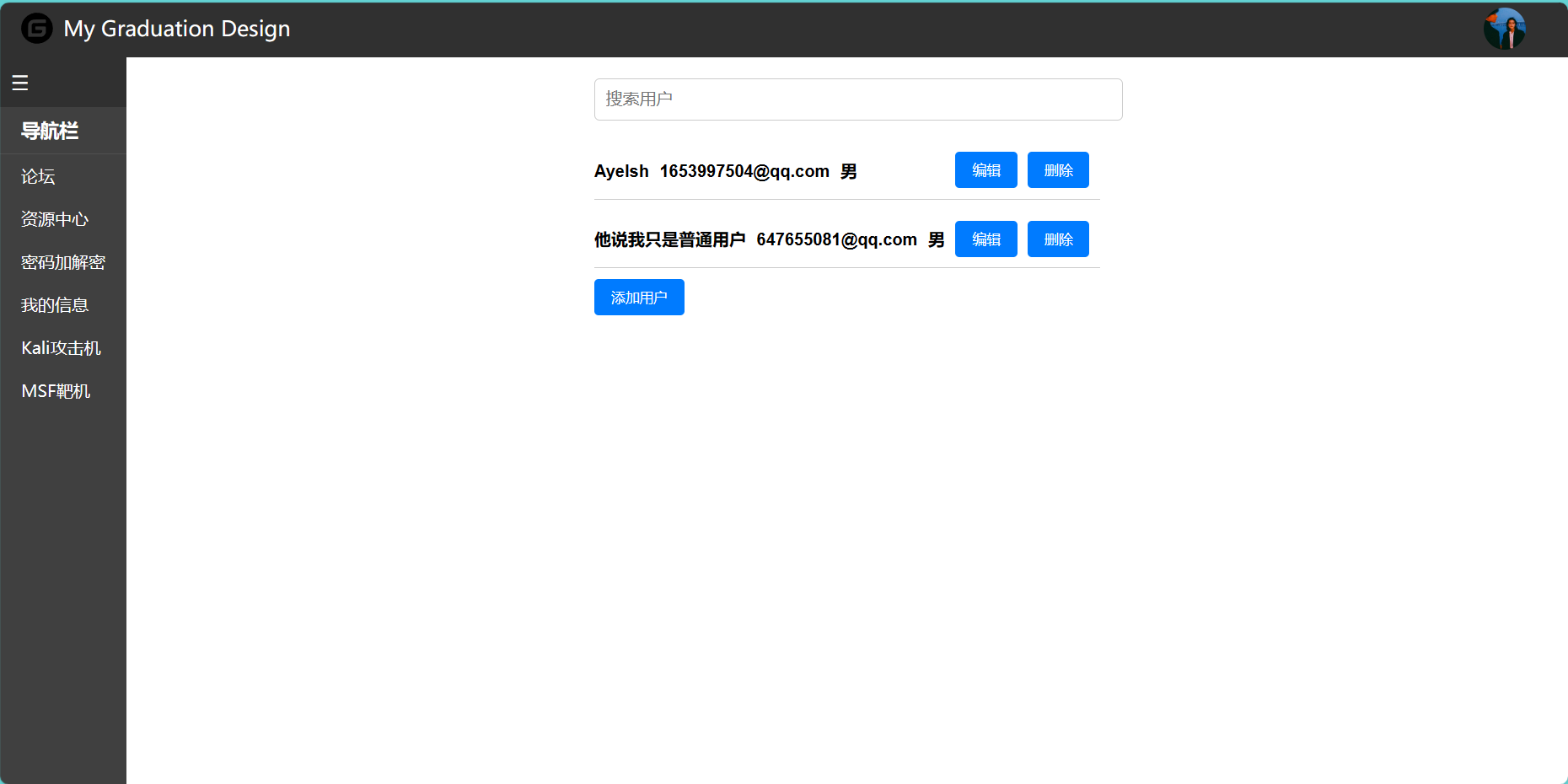


图34 用户管理界面

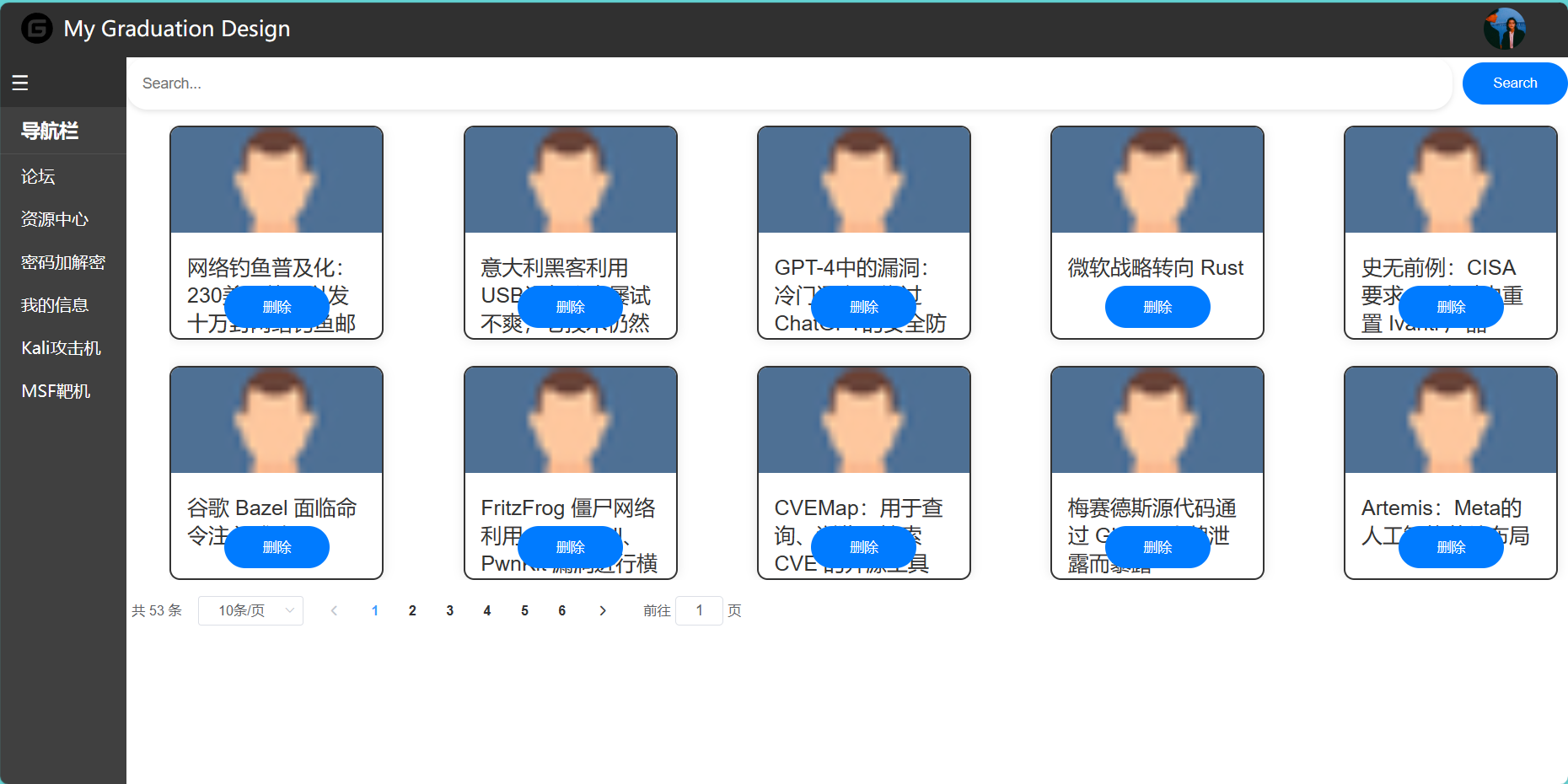


图35 帖子管理界面

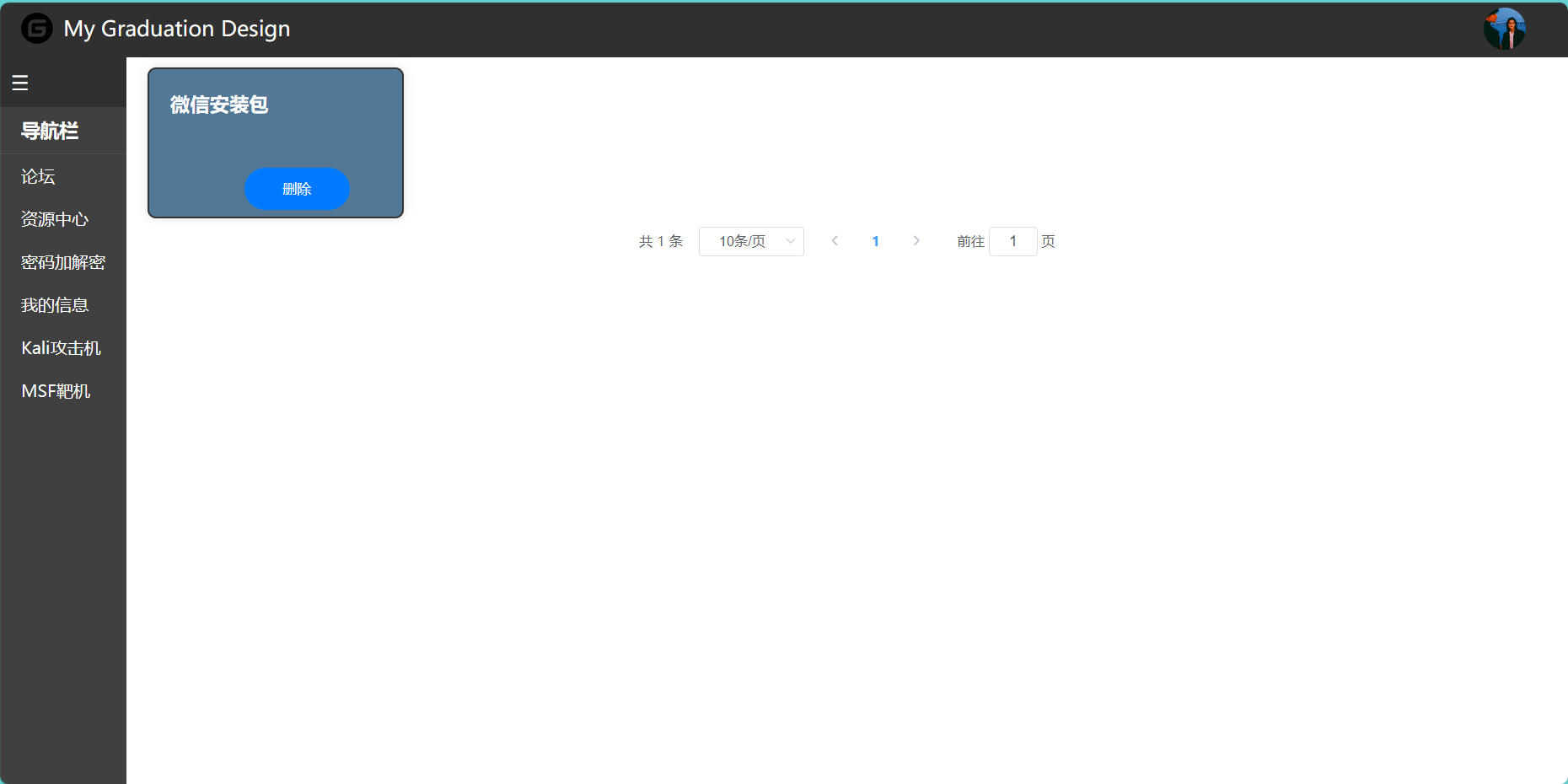


图36 资源管理界面

### 5.3.7 登陆注册模块

如图37示为用户登录界面，用户在注册之后可以登录进入系统，用户验证逻辑主要存在于后端。在后端，除了静态资源和登录/注册等接口被允许任意访问之外，其他接口将被强制要求鉴权，部分管理员操作则开启了方法级别鉴权，操作该类方法将需要一定的角色/权限。除此之外，由于前后端分离的开发模式的原因，前后端的通信凭证使用了JWT结合Redis和Security完成。每次需要鉴权的请求前端都会携带登录时候的token，后端在接受token后将各种解析后存入Security上下文，接口访问时将验证上下文存在的权限信息。

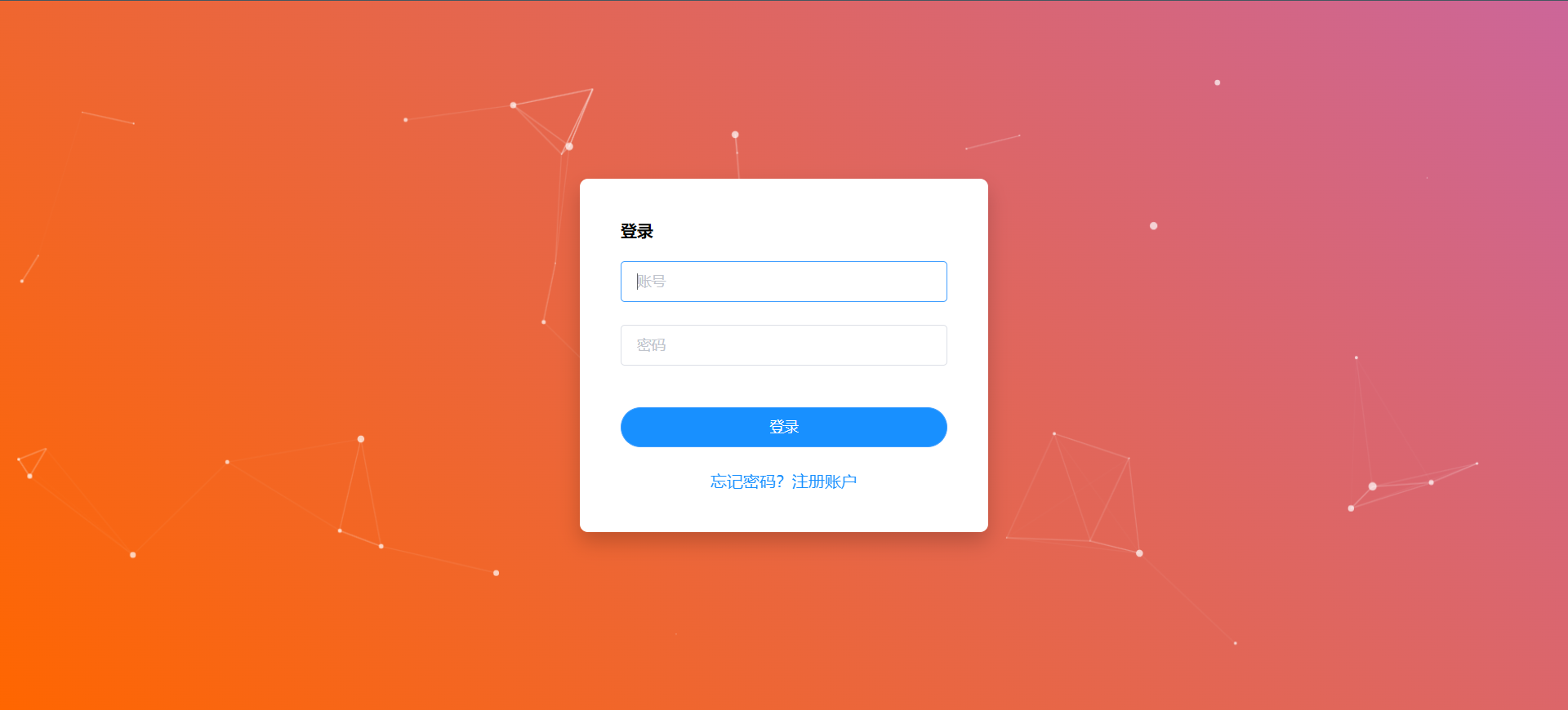


图37 登陆页面

# 6 系统测试

## 6.1 测试需求

软件测试是保证软件质量和可靠性的重要手段[15]。为了保证本系统最终上线质量，本次测试针对系统后端进行测试，通过模拟用户可能出现的各种操作编写测试用例。

## 6.2 测试目标

通过测试查找各种潜在性的bug，优化系统，解决潜在性恶性bug。

## 6.3 测试用例

### 6.3.1 登陆注册模块

本模块注册时只能使用未注册的邮箱，不然提示注册失败，登录的时候需要输入账号密码，错误的账号密码将提示登录失败，登录成功后将自动跳转进论坛界面。

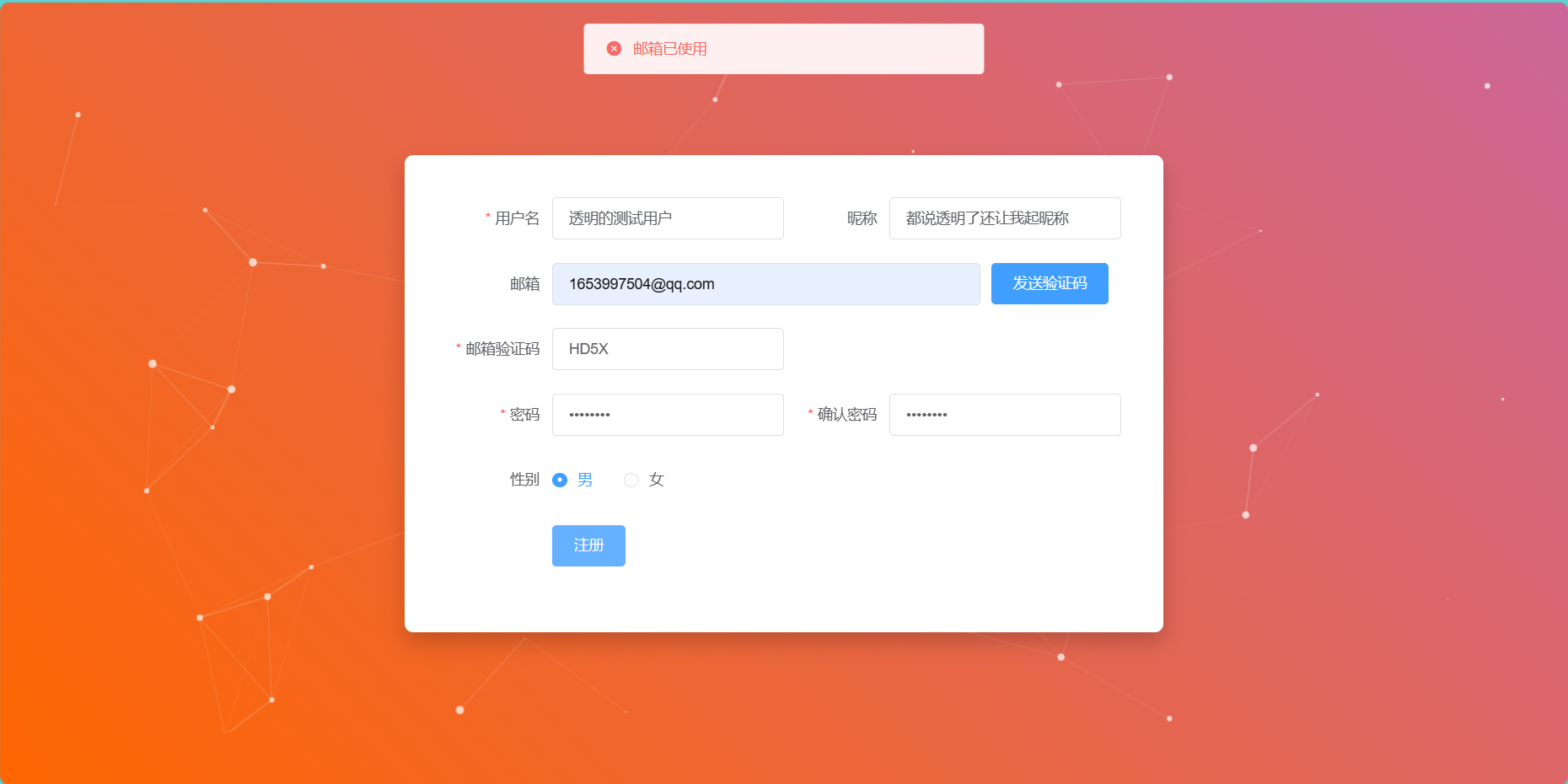


图38 邮箱注册失败

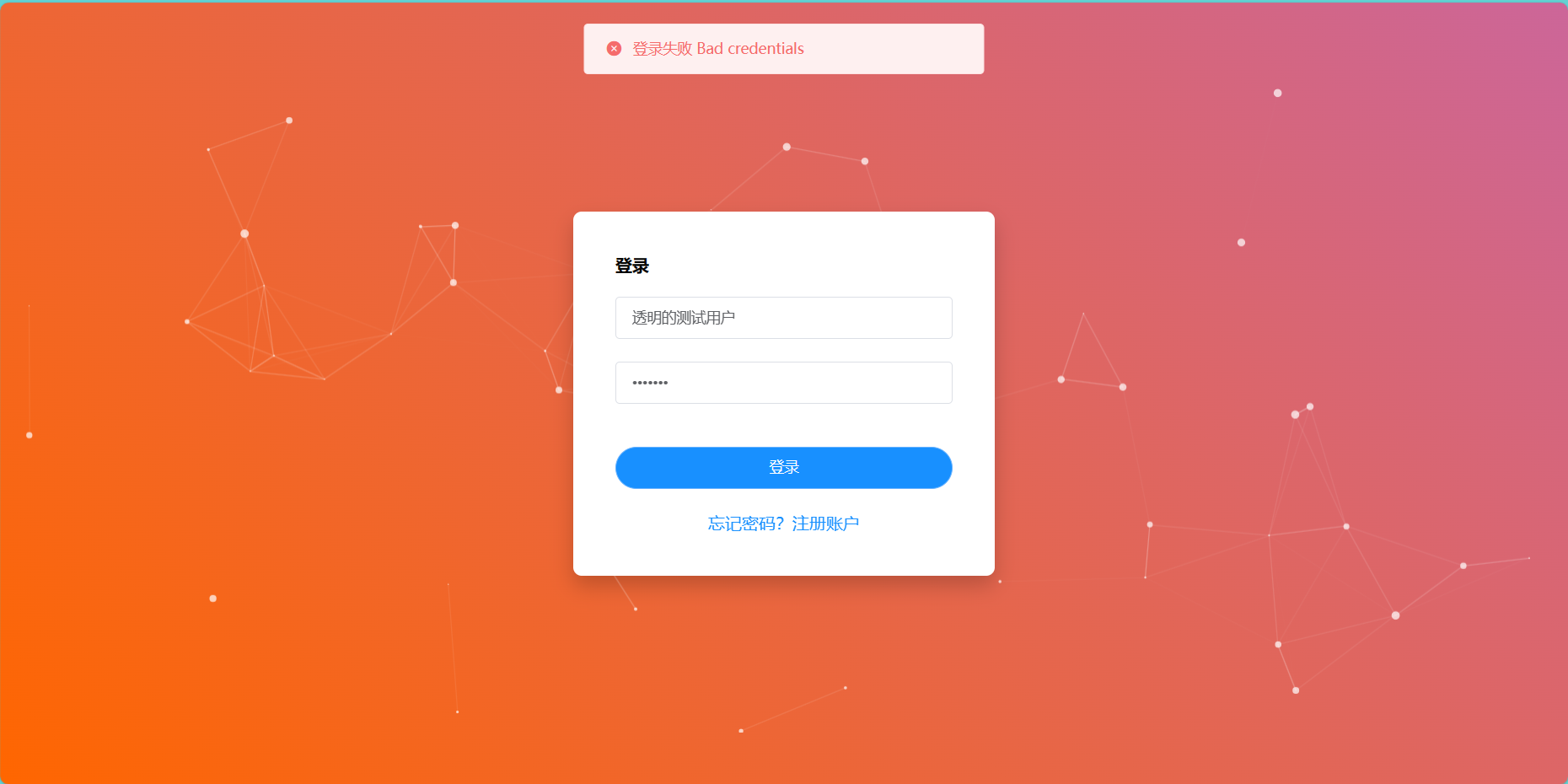


图39 邮箱注册失败

### 6.3.2 论坛模块

进入论坛后自动分页显示，点击切换分页大小，点击页数跳转到对应页面，点击帖子进入对应页面，发表子评论嵌在主评论里，点击发布按钮进入帖子编辑界面，MarkDown富文本编辑后能正确预览，以上测试均通过，符合预期。

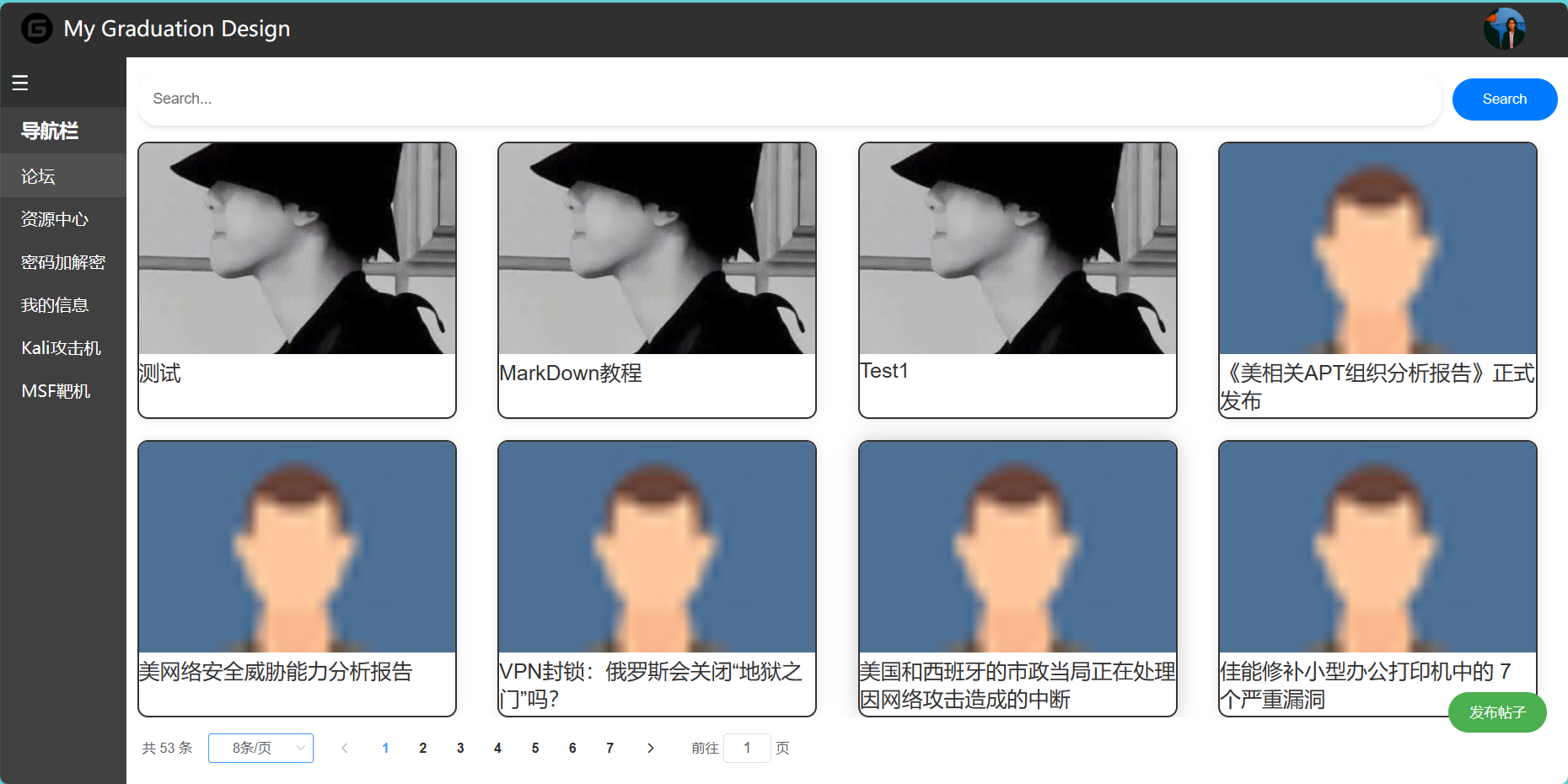


图40 论坛分页

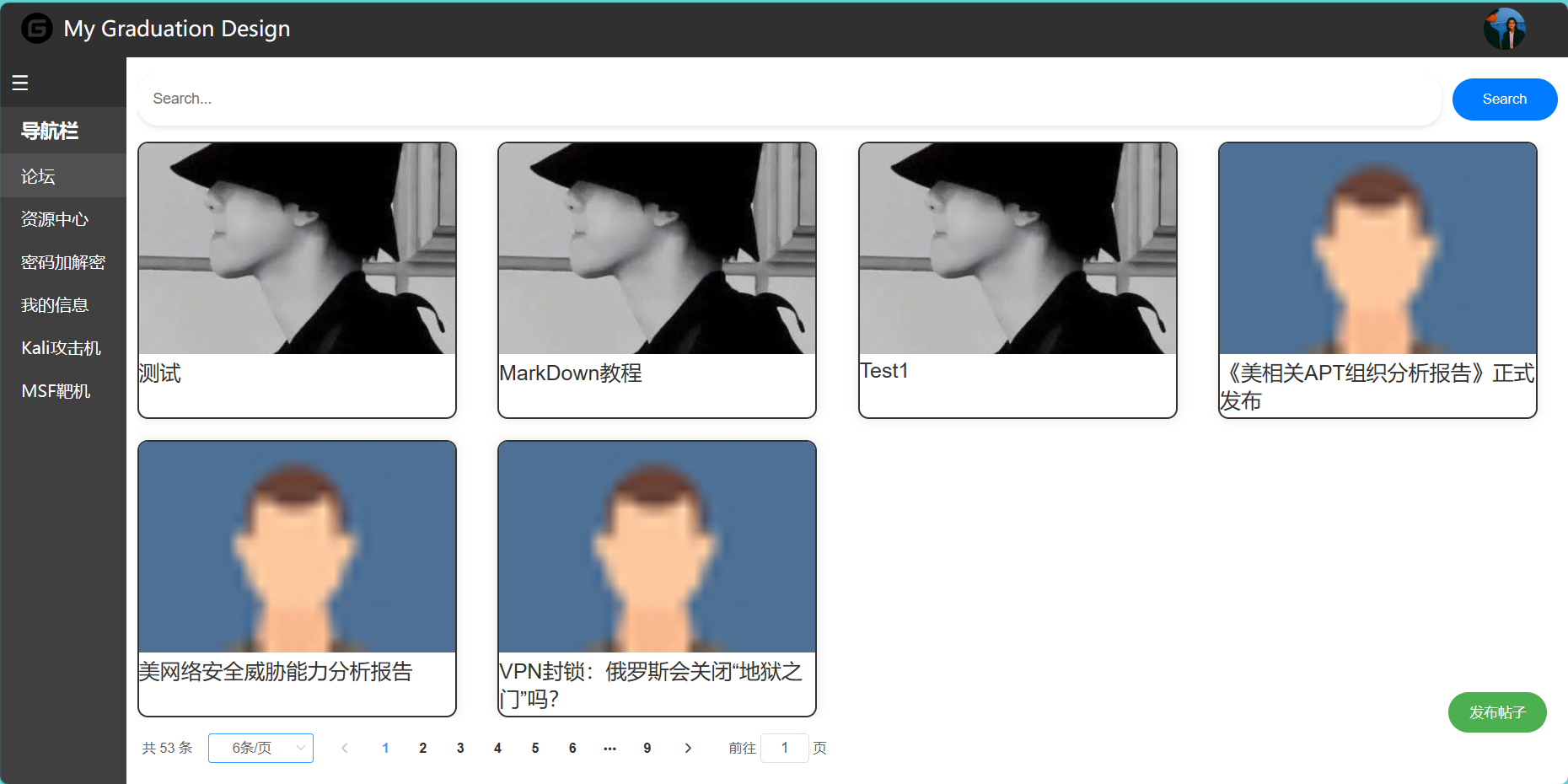


图41 调整分页与跳转页面



图42 正文页面以及评论发表功能

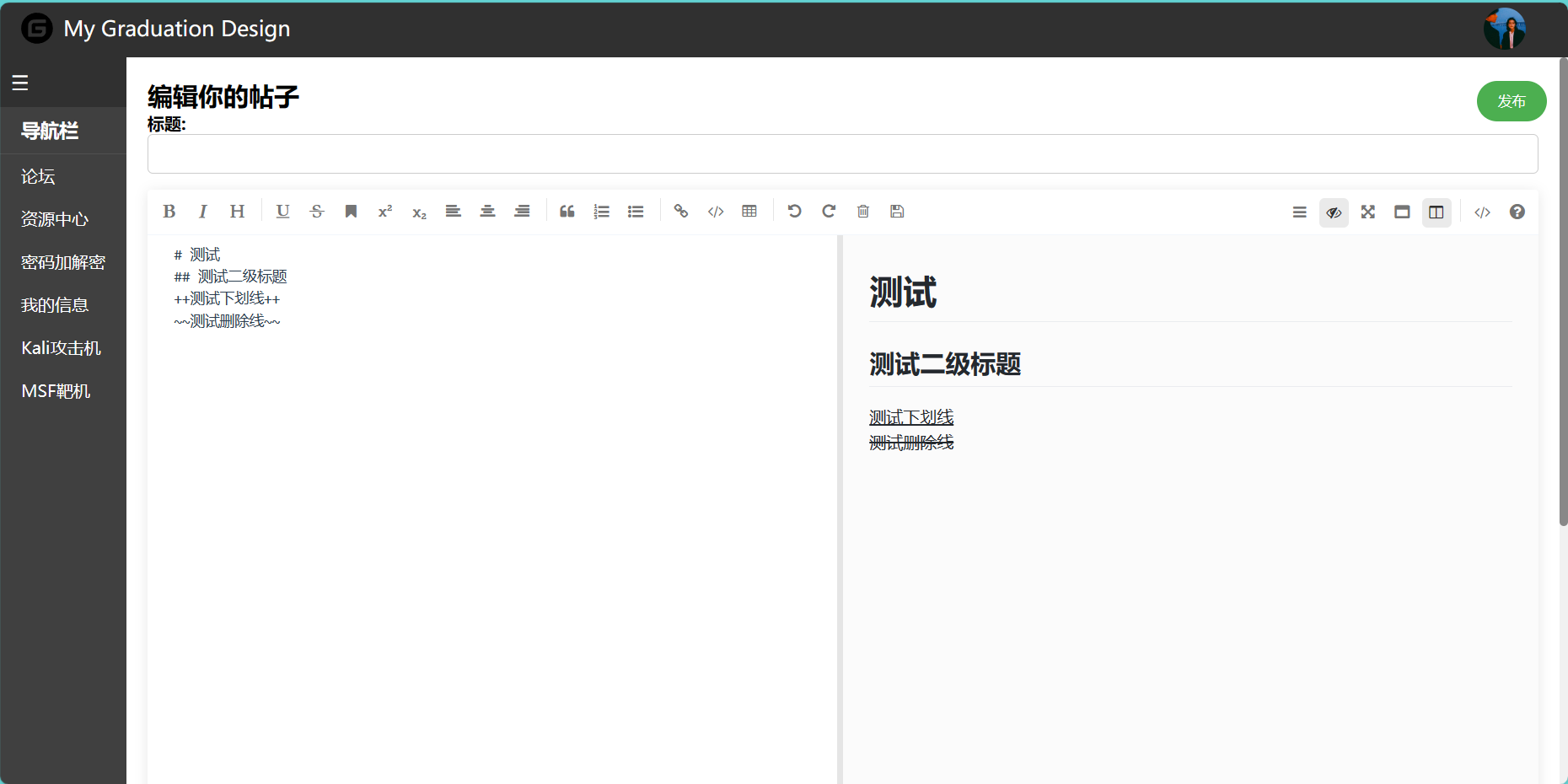


图43 富文本编辑页面

### 6.3.3 资源中心模块

进入资源中心后资源自动分页显示，点击切换分页大小，点击页数跳转到对应页面，点击下载按钮资源能正常下载，提交资源时候输入超长文本遭受限制，以上测试均通过，符合预期。

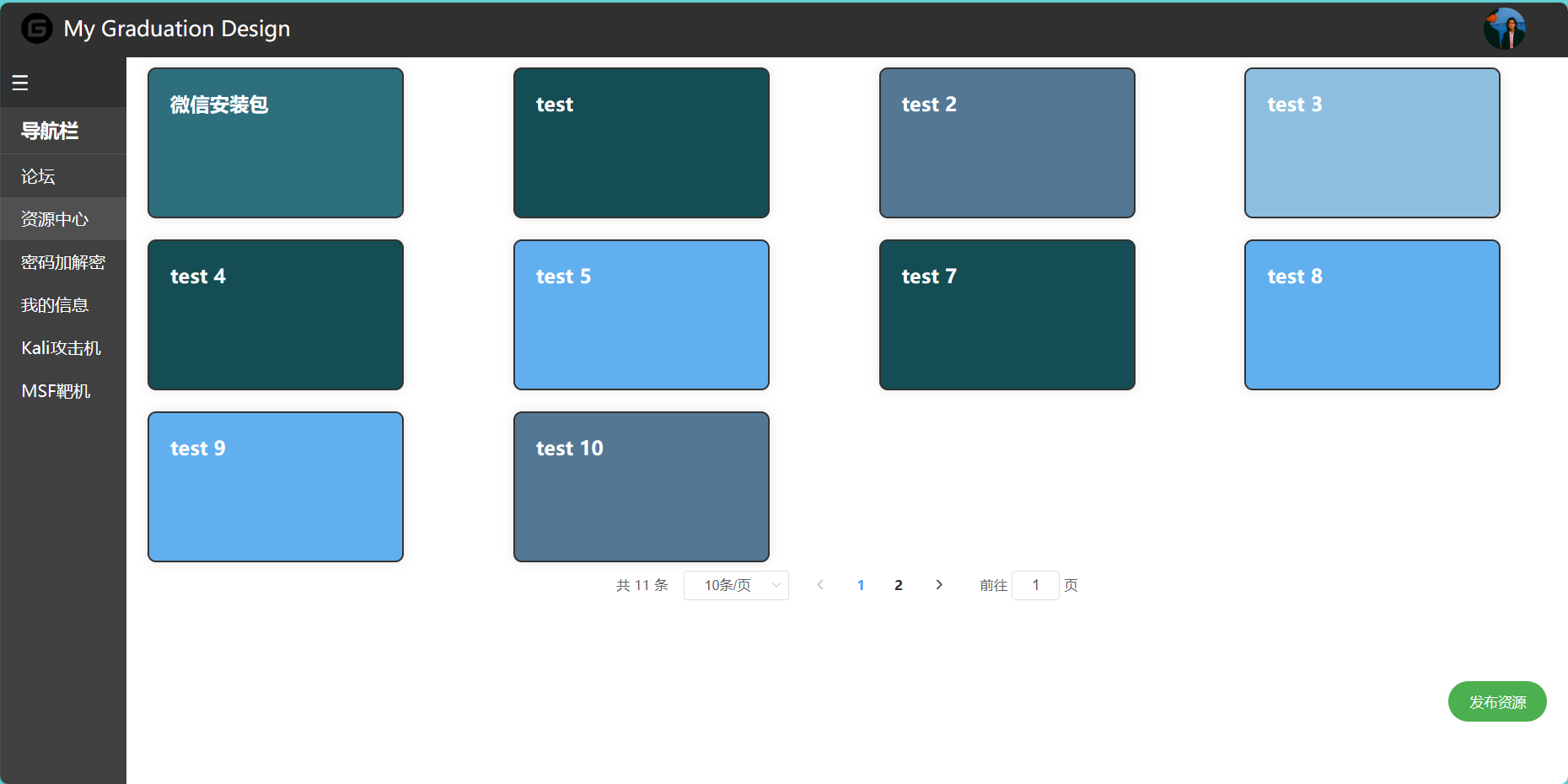


图44 资源自动分页



图45 资源中心调整分页与页面跳转

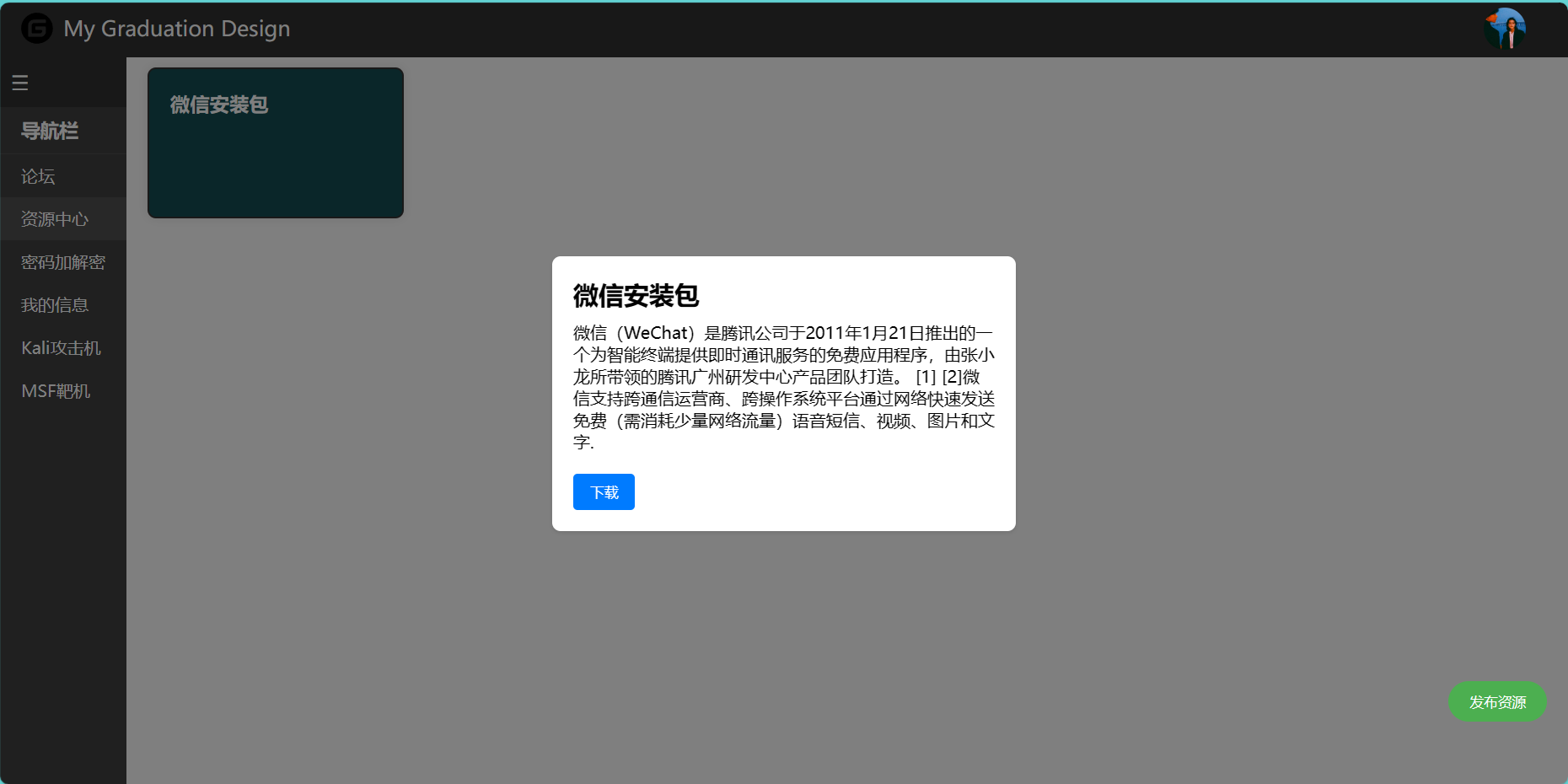


图46 资源查看与下载

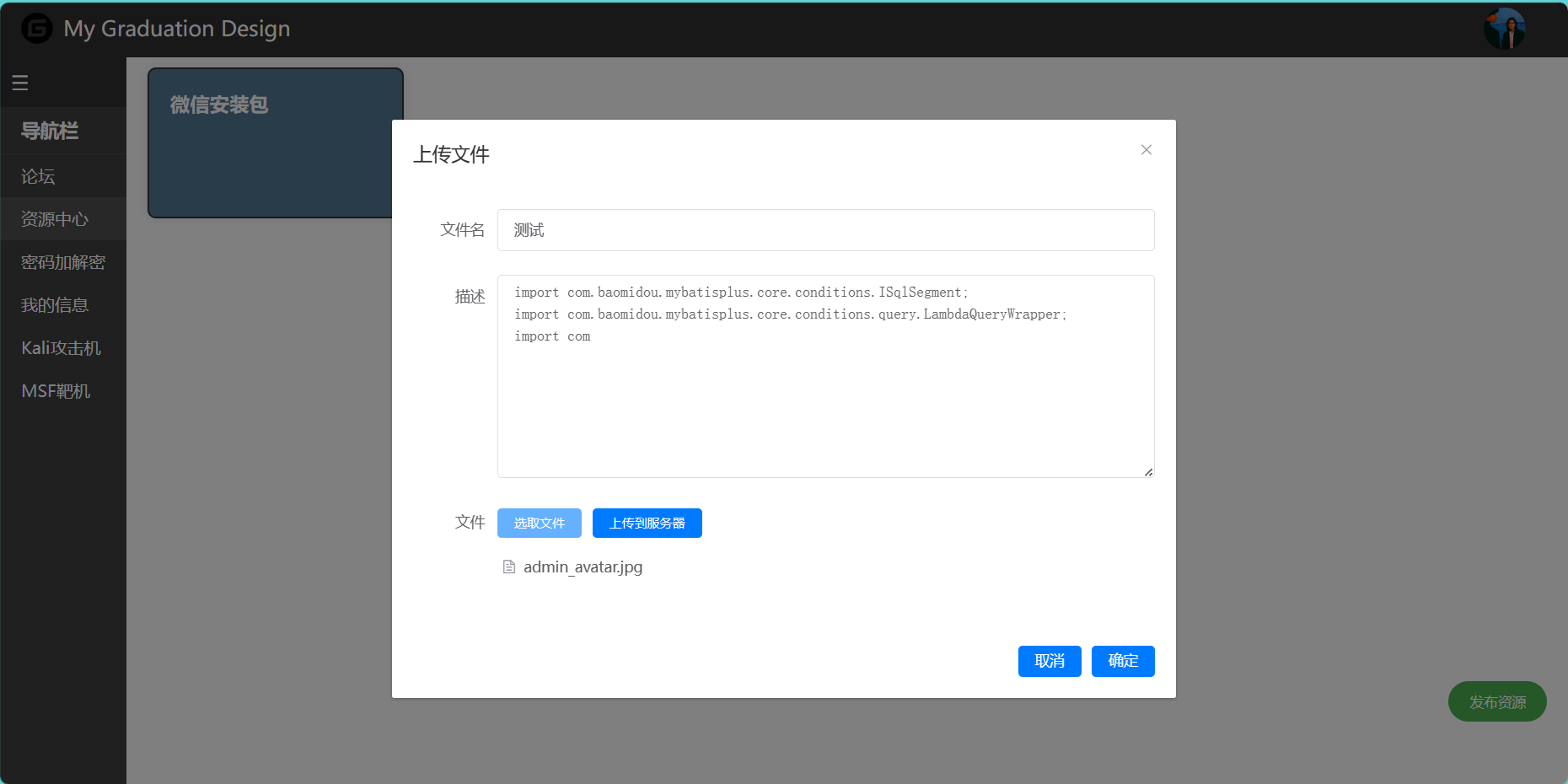


图47 资源发布

### 6.3.3 密码加解密模块

进入密码加解密模块选择加密或者解密，输入密钥，选择对应算法后加密的文本可以被同样的算法和密钥解密，以上测试均通过，符合预期。

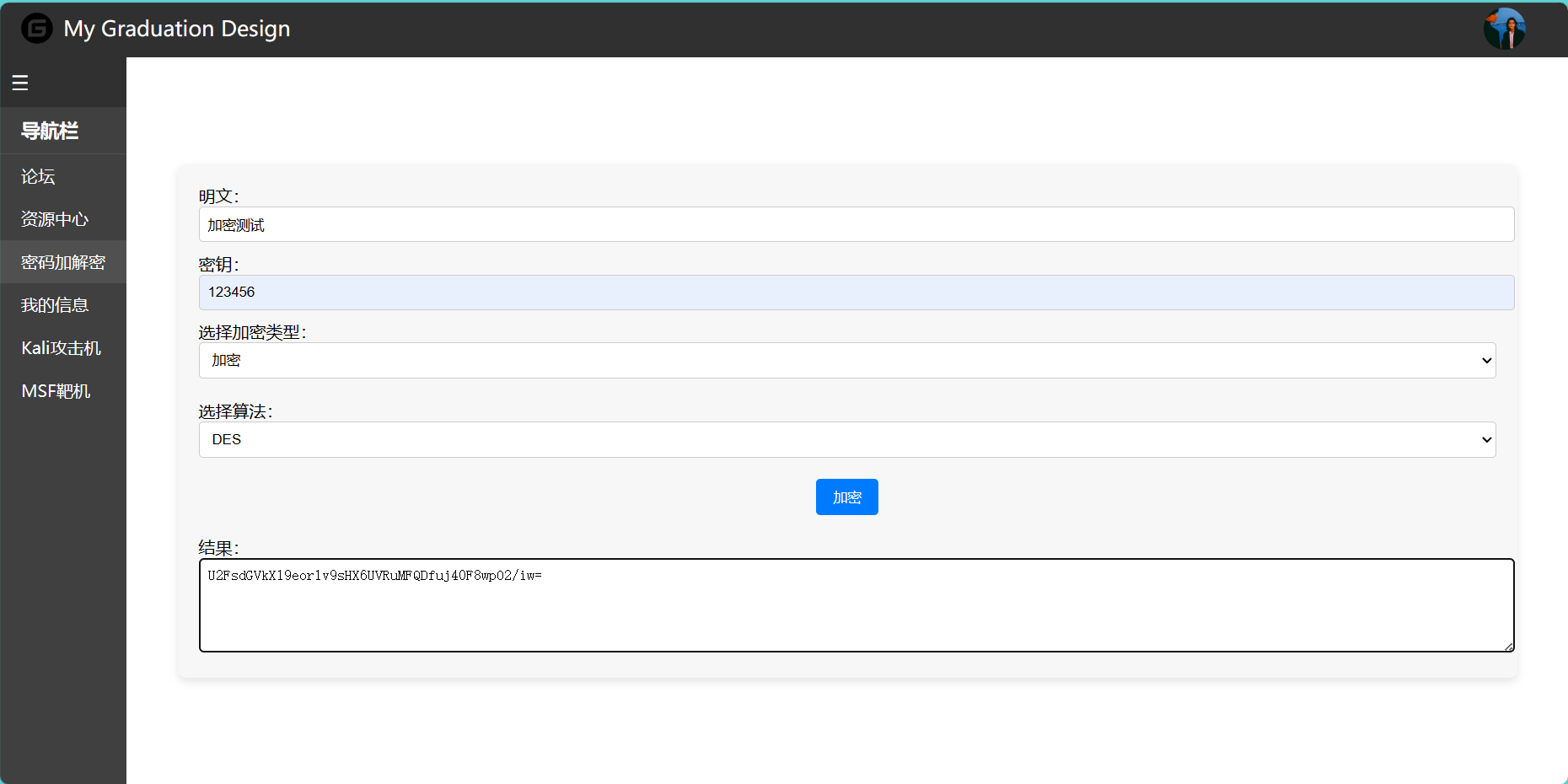


图48 明文加密

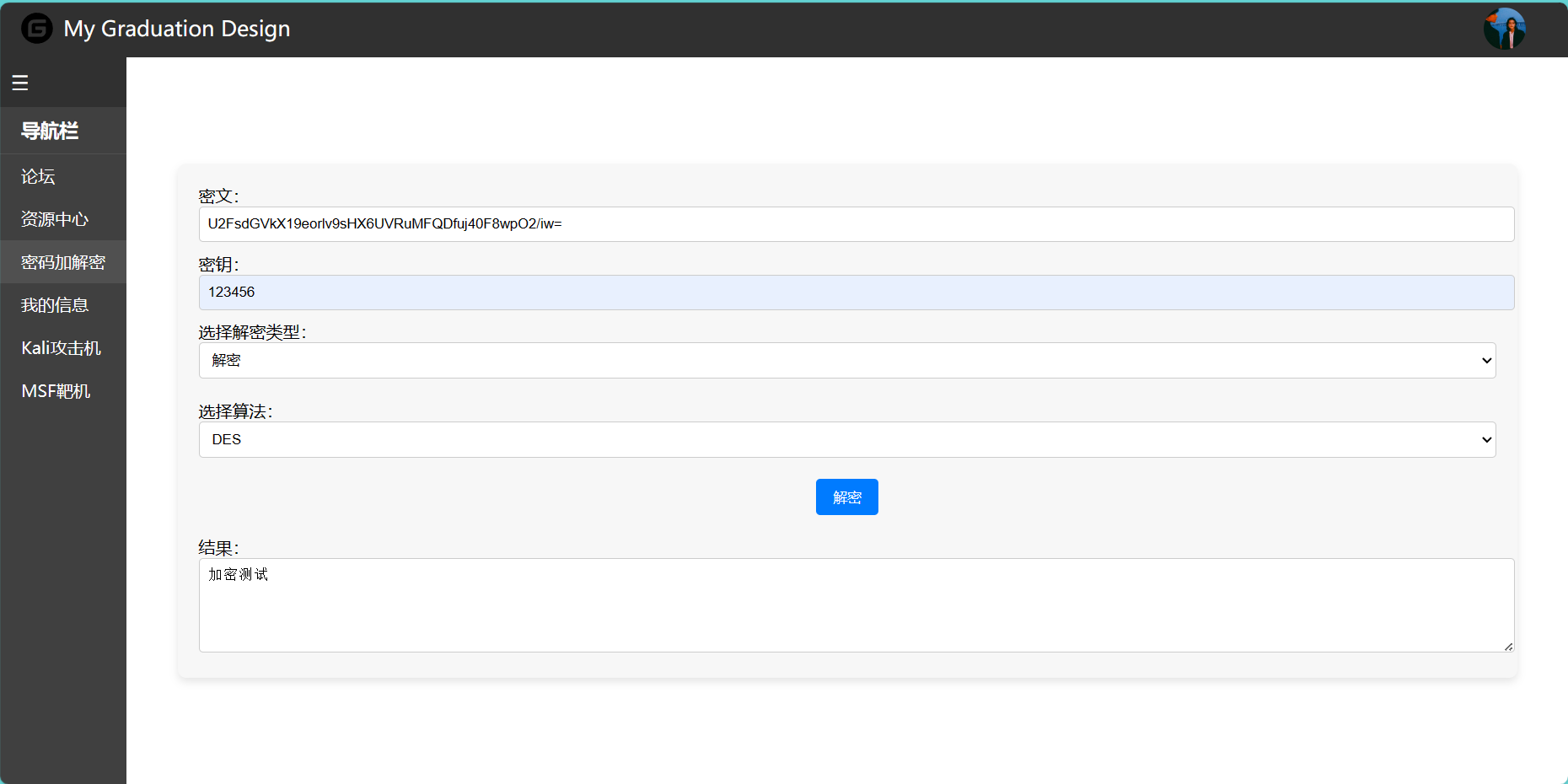


图49 密文解密

### 6.3.4 我的信息模块

在导航栏点击“我的信息”查看用户信息，点击右上角头像菜单里的“修改我的信息解密”进入修改信息界面，修改自己的个人信息与头像，修改后“我的信息”页面同步更新修改后的信息，以上测试均通过，符合预期。

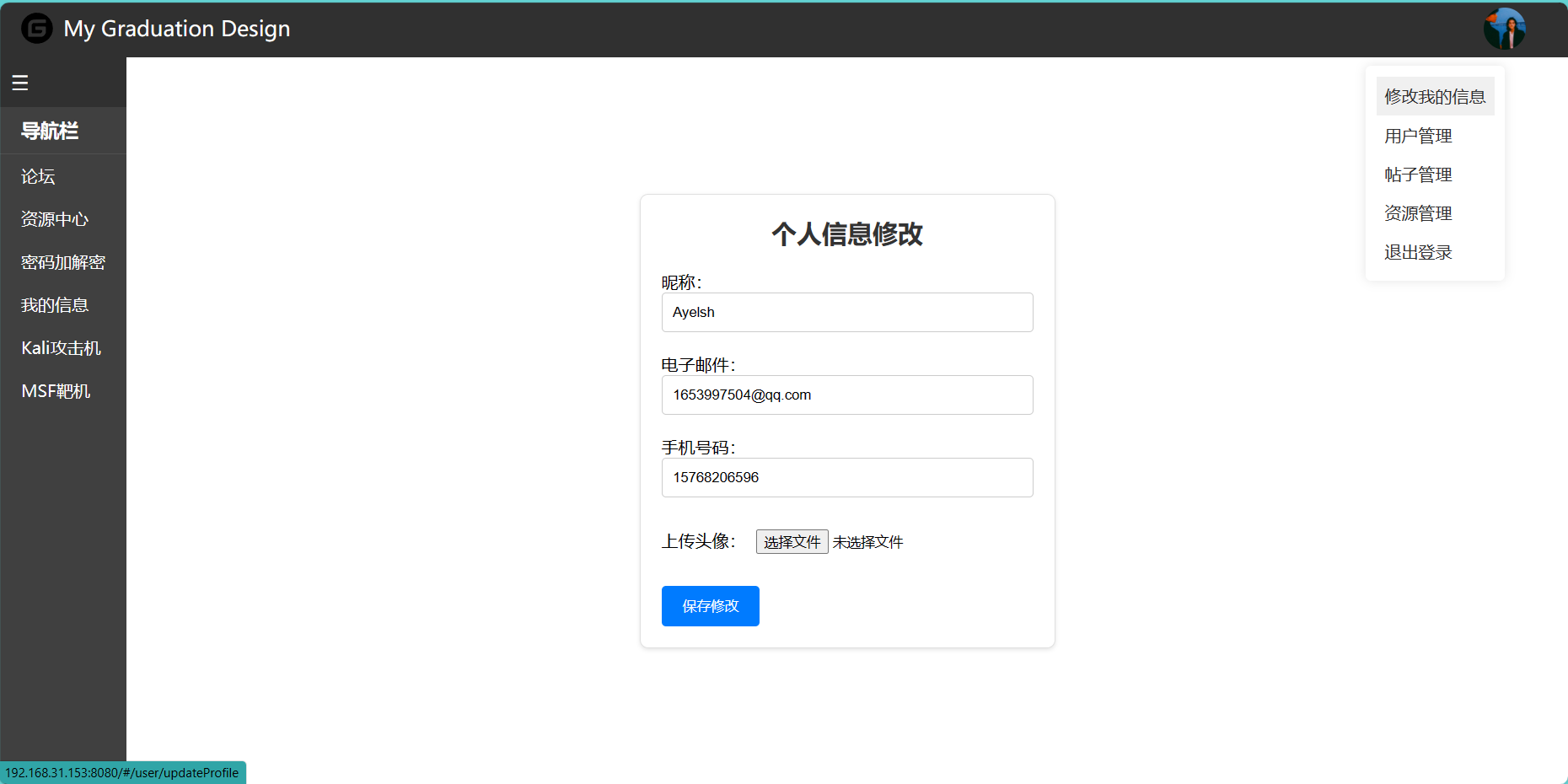


图50 修改个人信息页面

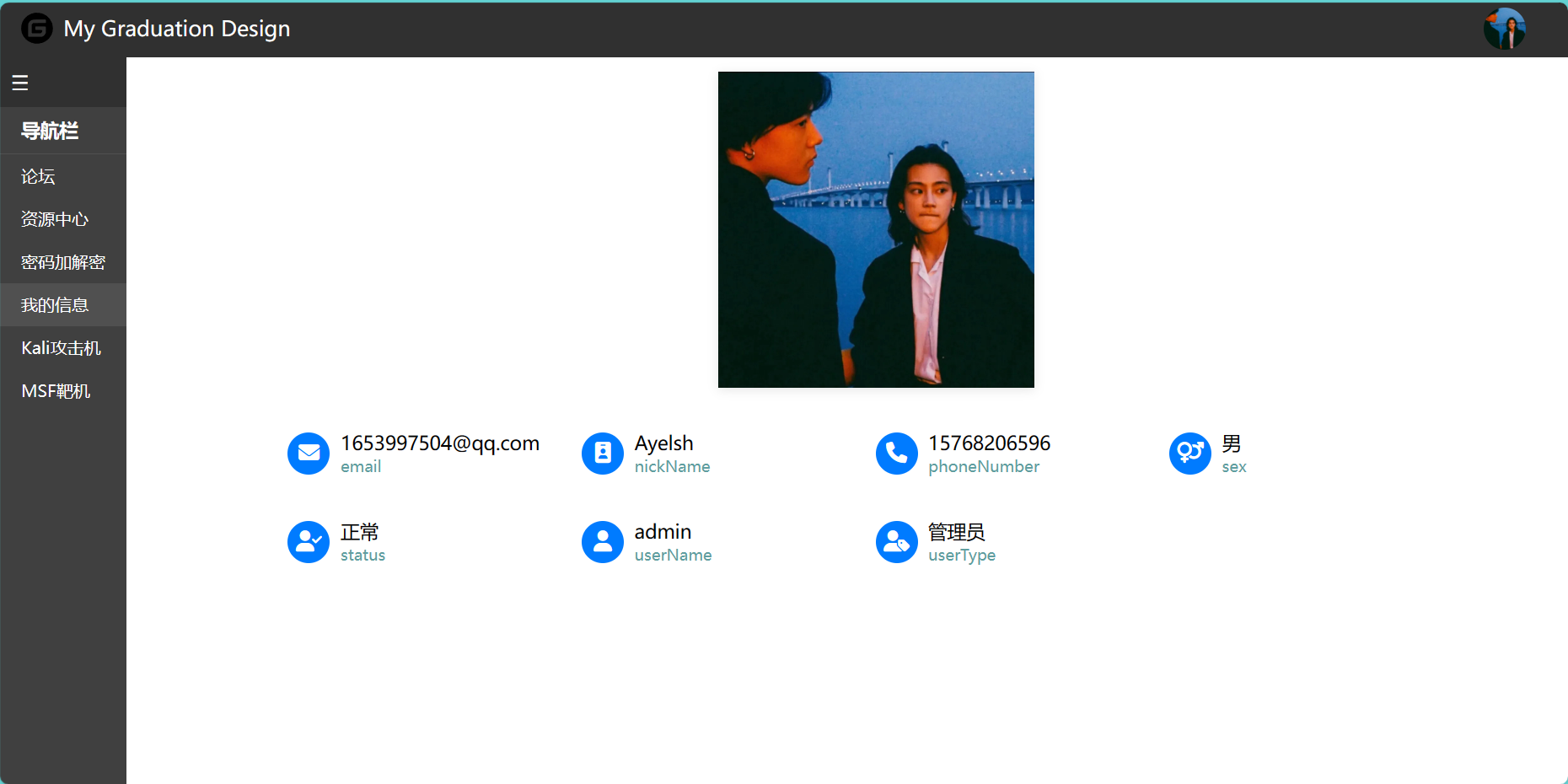


图51 修改后同步个人信息

### 6.3.6 Liunx模块

进入攻击机可以调用各种按钮与输入命令，所有命令都可以执行调用，进入靶机界面可以输入命令，回车发送命令并正常响应，以上测试均通过，符合预期。



图52 输入命令正常响应

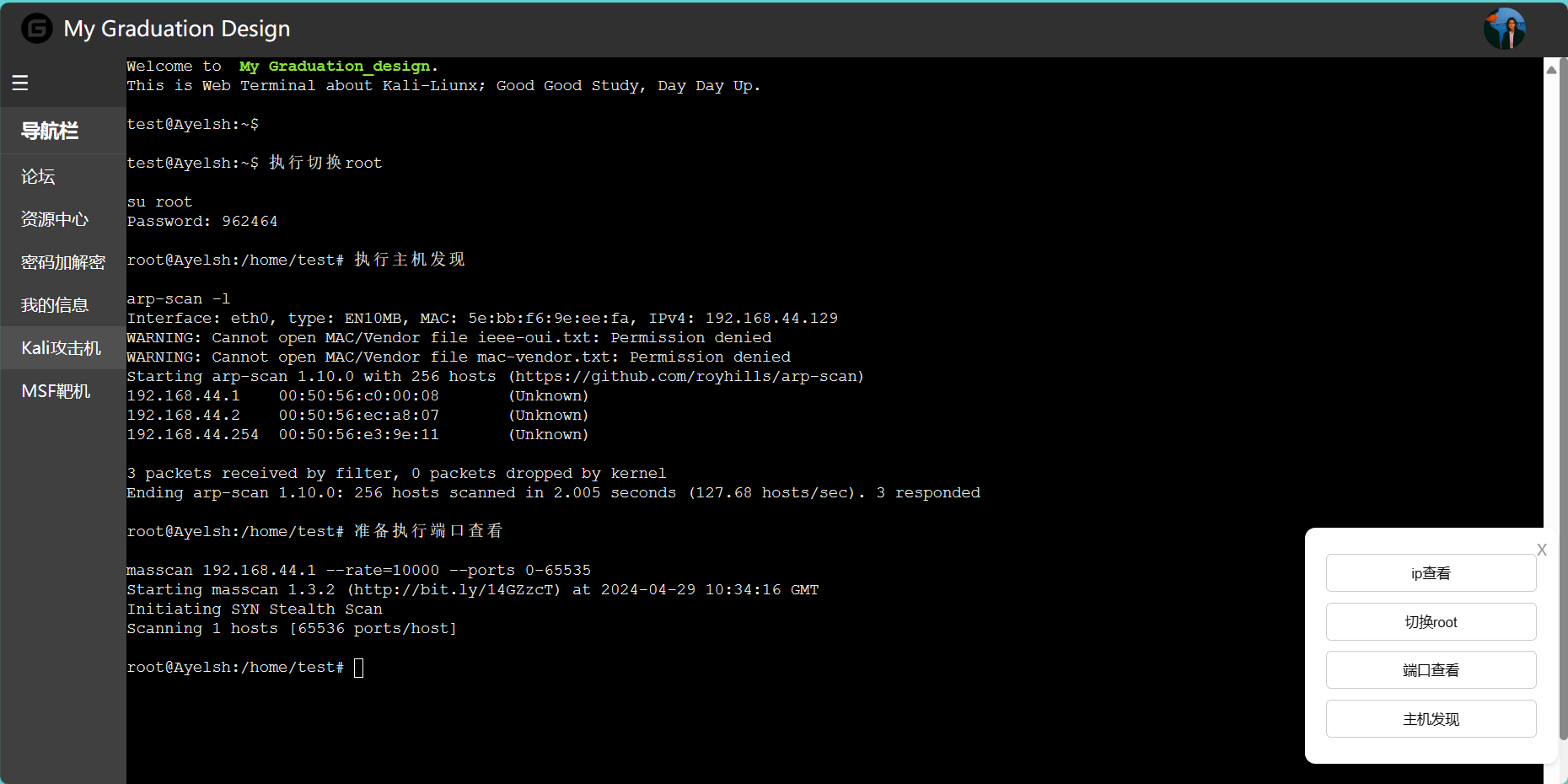


图53 调用按钮正常响应

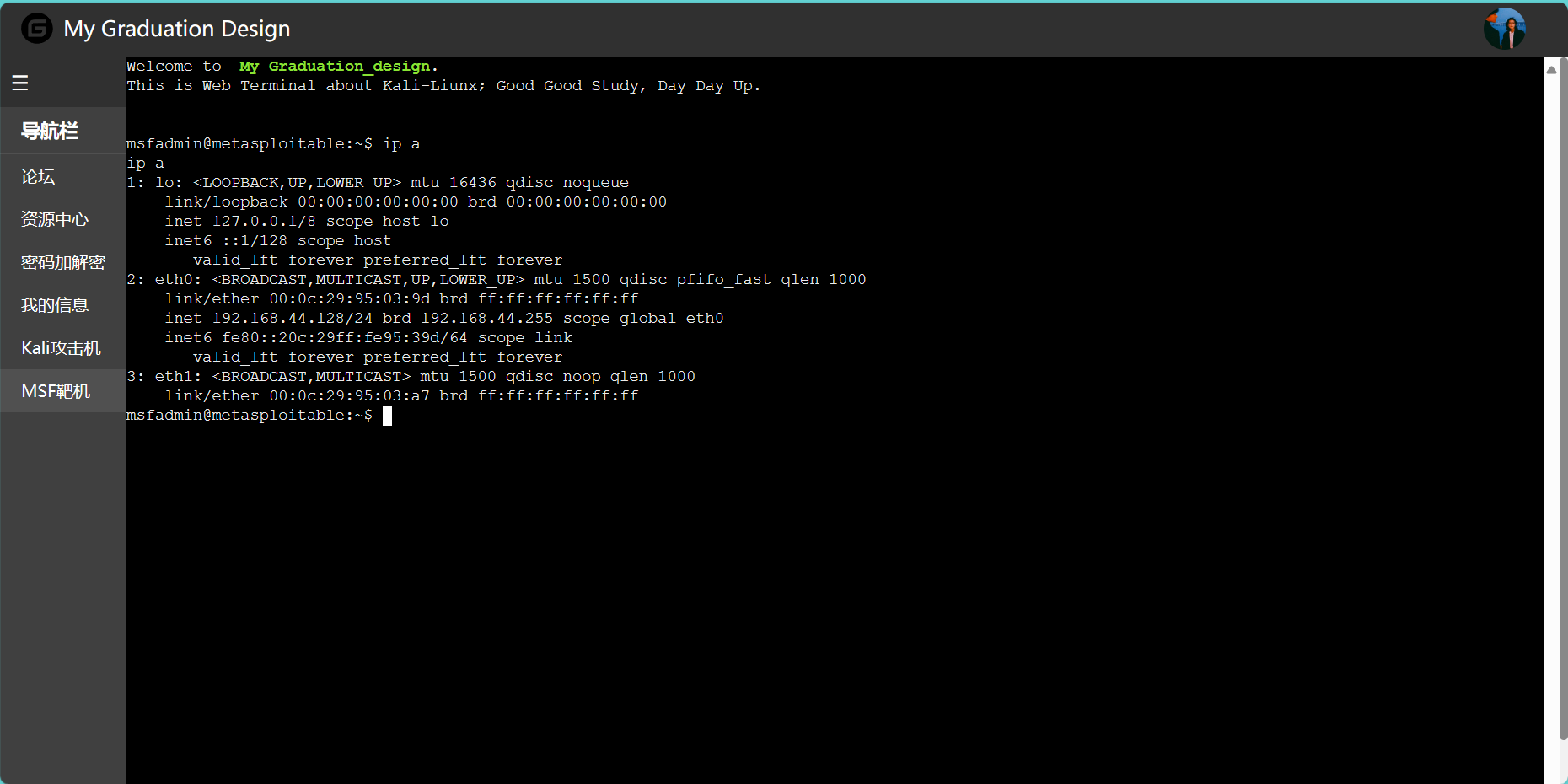


图54 靶机命令正常响应

### 6.3.7 系统管理模块

普通用户头像菜单不会有三个管理菜单，管理员用户拥有三个管理菜单，用户管理增删改用户，帖子管理删除帖子，资源管理删除资源，以上测试均通过，符合预期。

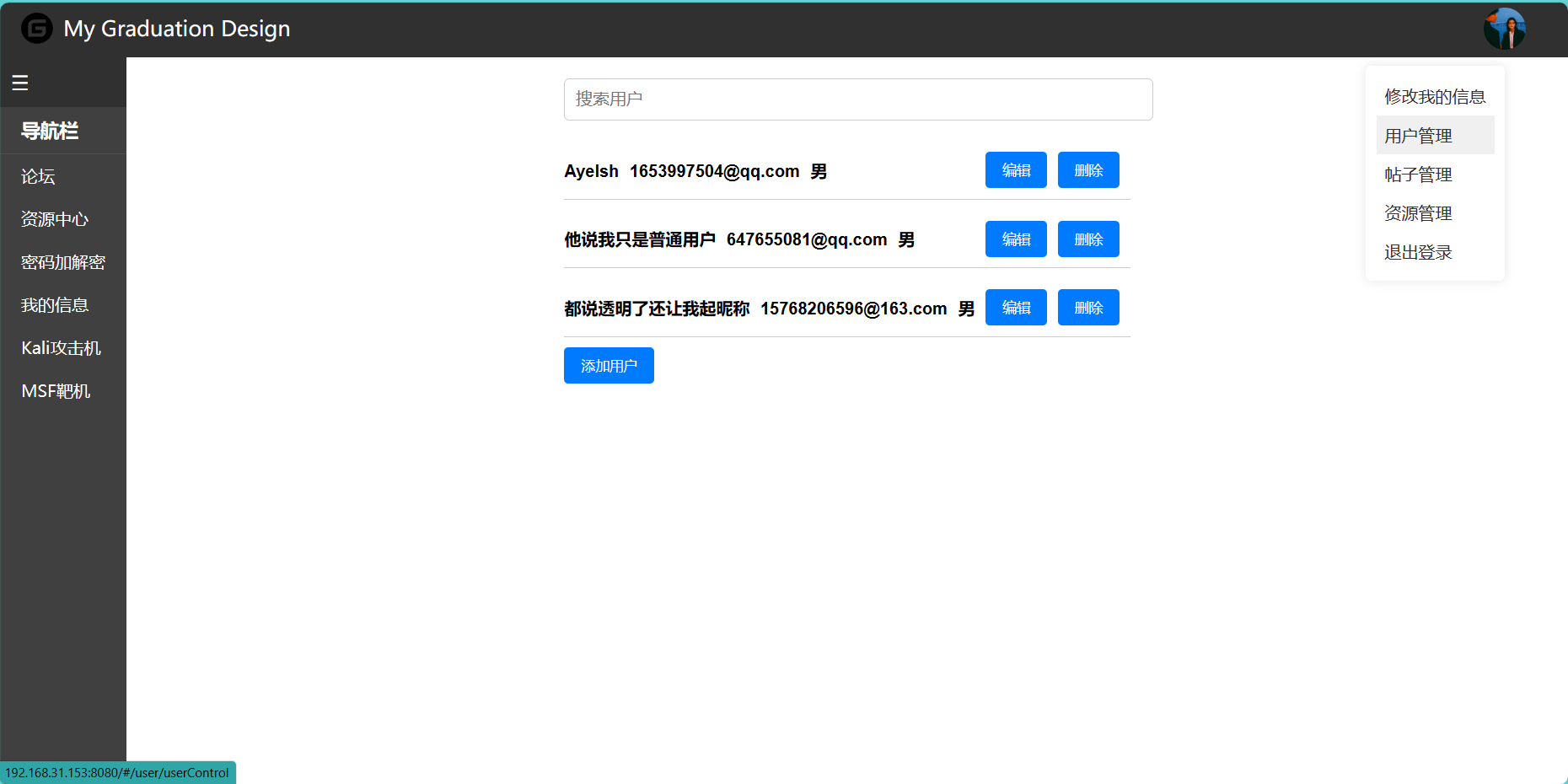


图55 三个管理菜单以及用户管理界面

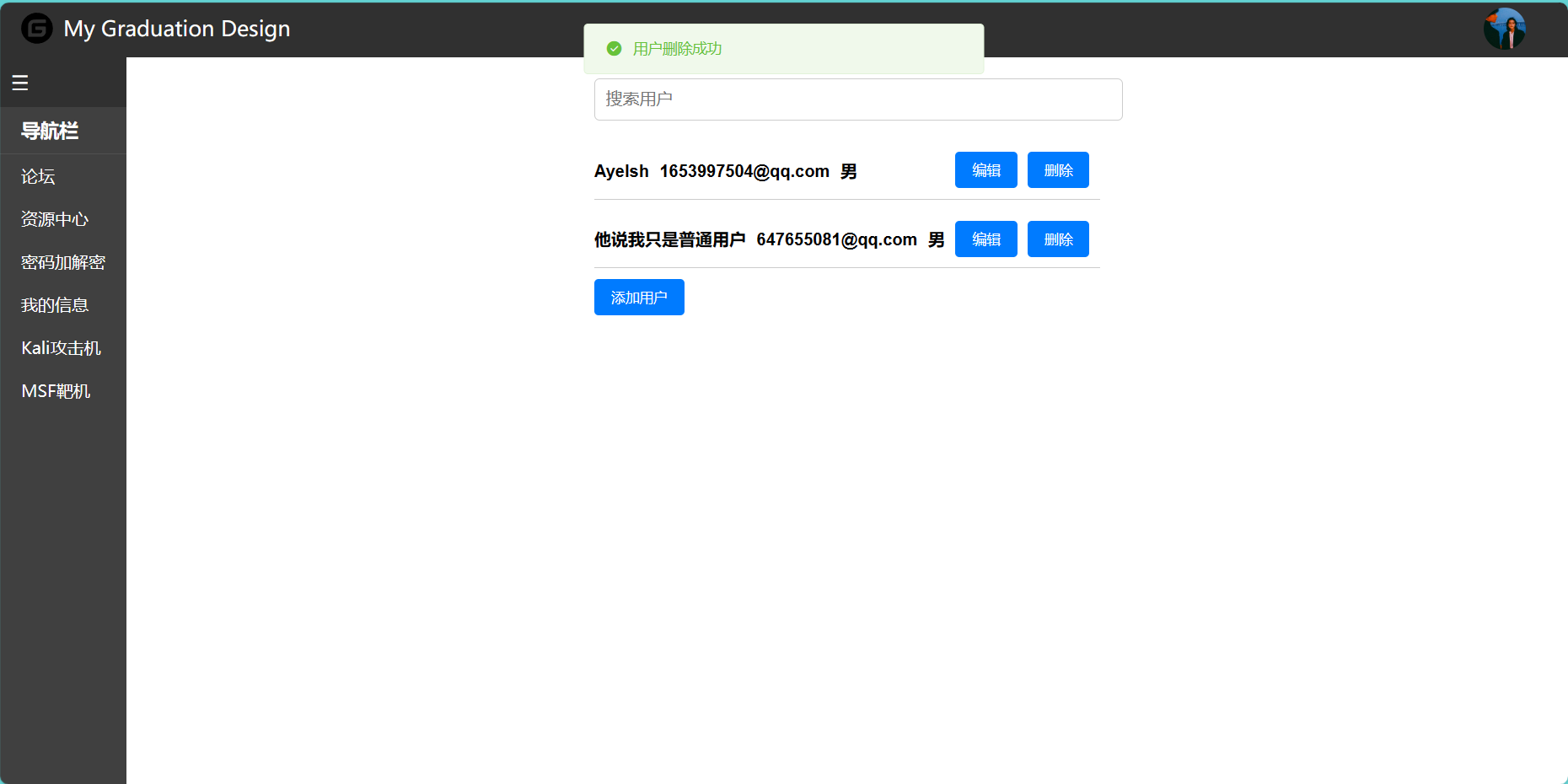


图56 成功删除用户

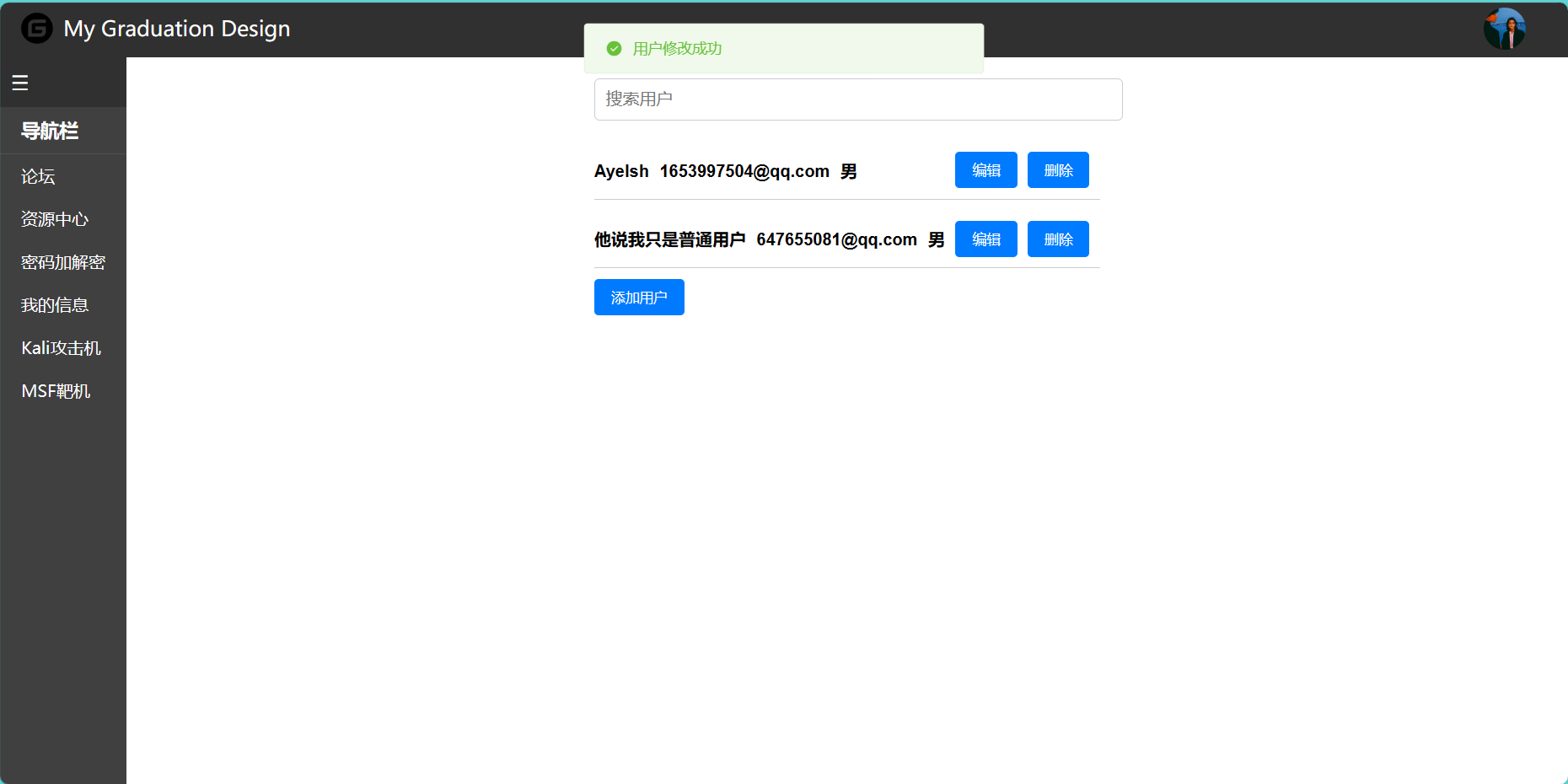


图57 成功修改用户

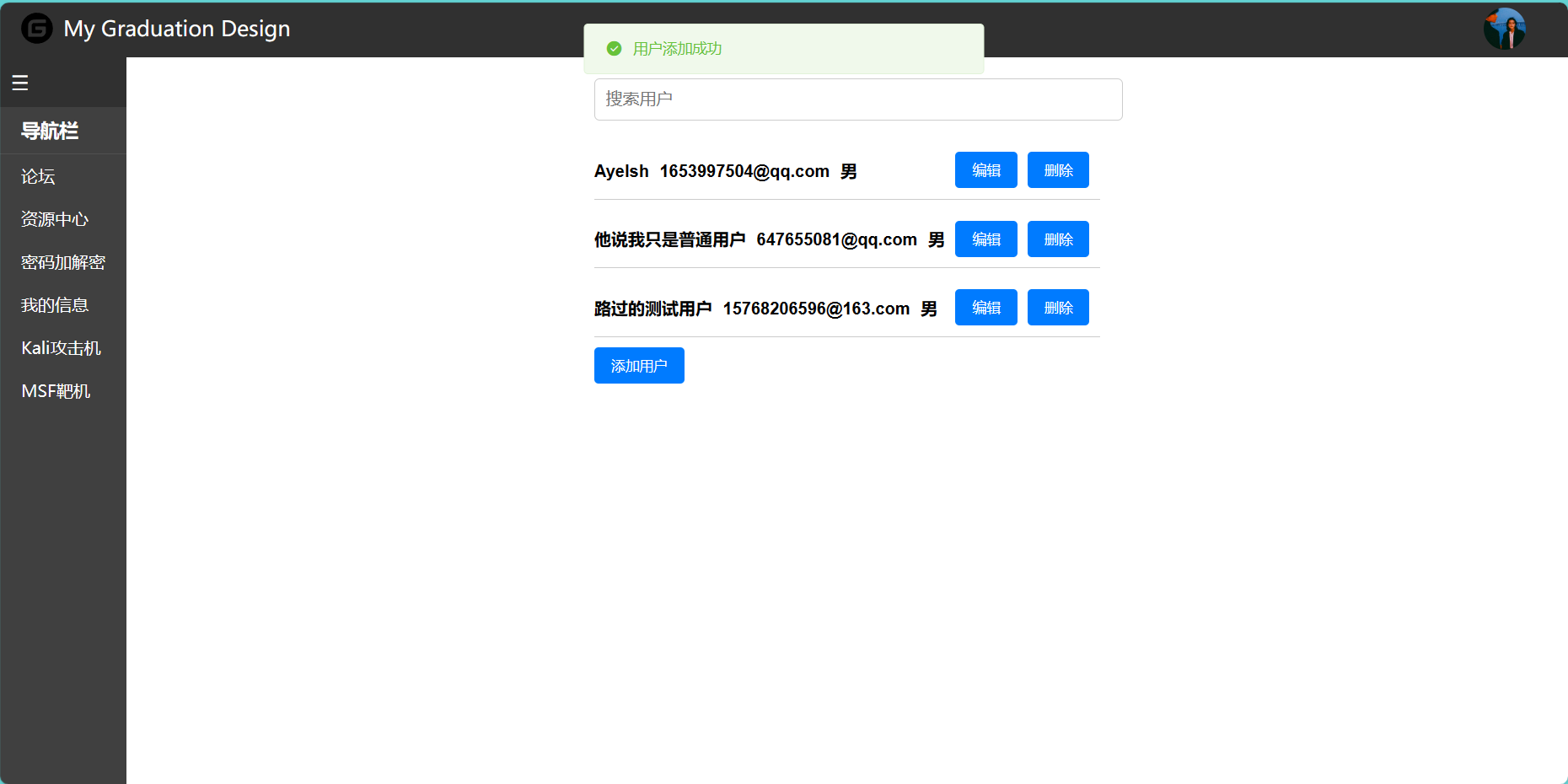


图58 成功增加用户

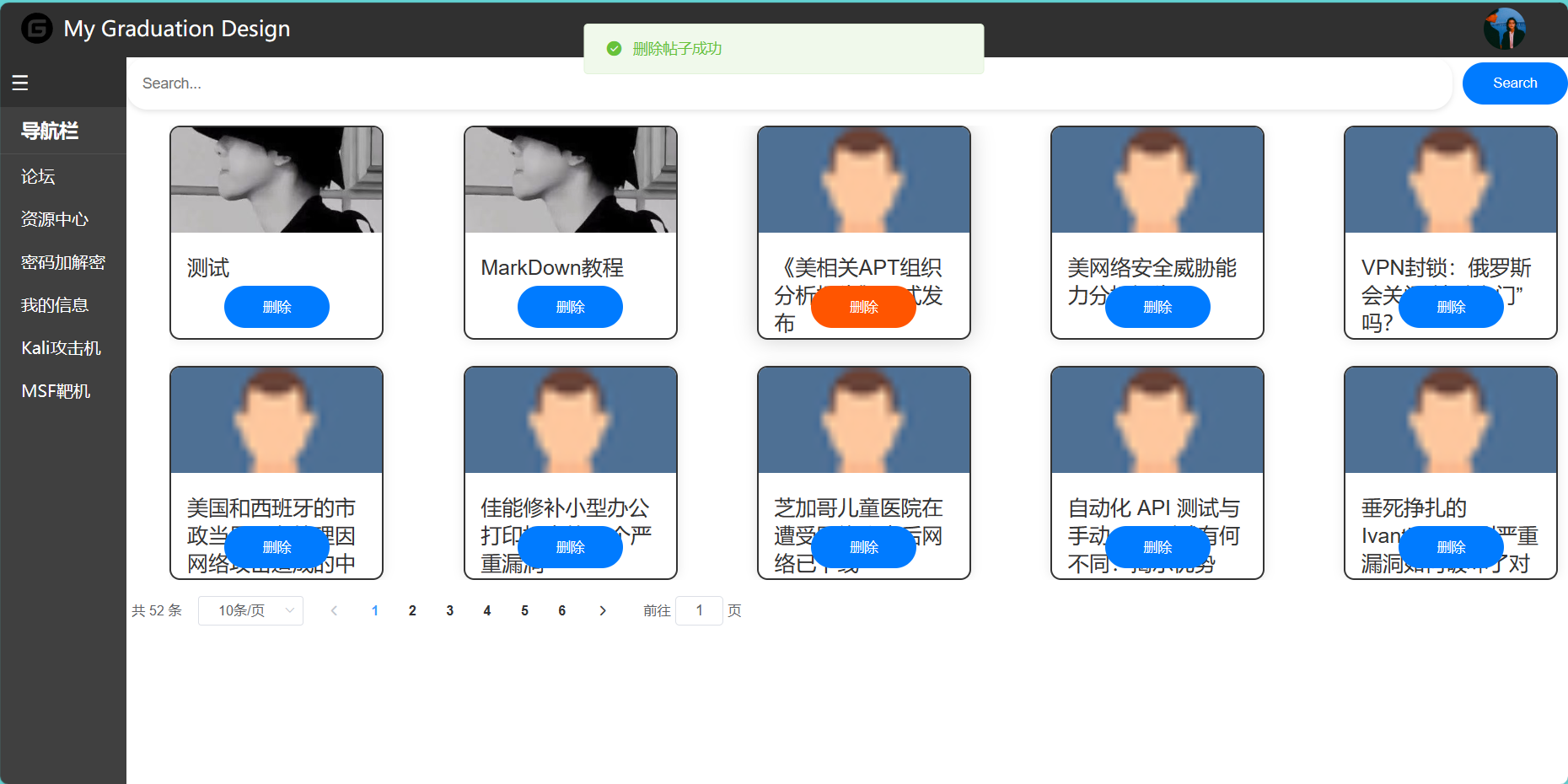


图59 成功删除帖子

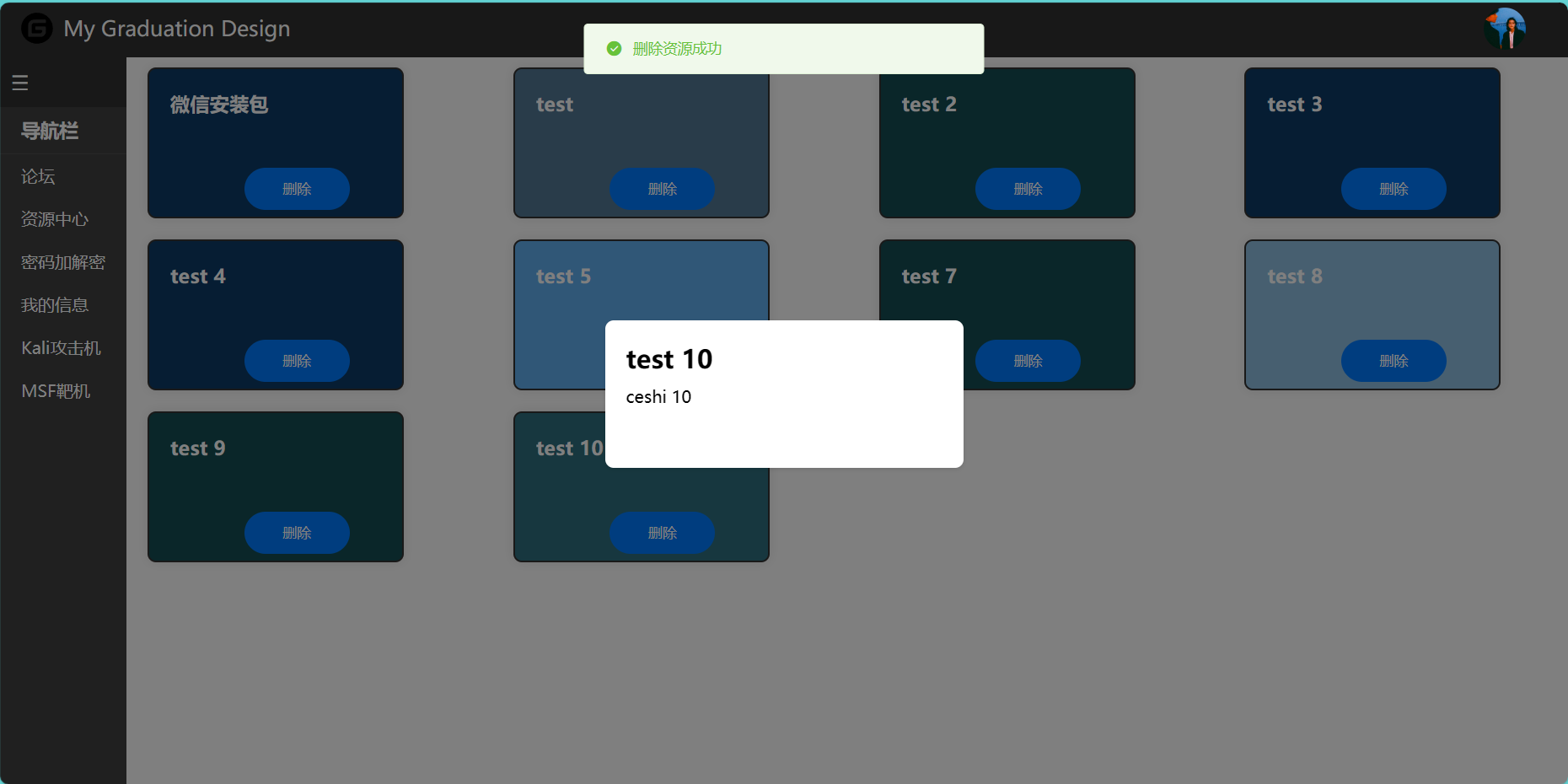


图60 成功删除资源

## 6.4 自动化测试情况

表17 用户登陆测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 |
| 1 | 登录 | 正确的账号密码 | 登录成功返回token | 登录成功返回token |
| 2 | 错误的账号或者密码 | 返回状态码600，msg显示账号或者密码错误 | 返回状态码600，msg显示账号或者密码错误 |
| 3 | 不输入账号或者密码 | 前端限制，要求输入账号或者密码 | 前端限制，要求输入账号或者密码 |

表18 用户访问受限页面测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 |
| 1 | 用户访问受限页面测试 | 有权限访问无查询条件接口 | 成功返回数据 | 成功返回数据 |
| 2 | 无权限访问任何接口 | 返回状态码403 | 返回状态码403 |
| 3 | 有权限携带正确条件访问条件查询接口 | 成功返回数据 | 成功返回数据 |
| 4 | 有权限携带错误条件访问条件接口 | 返回状态码600，msg:查询数据为空 | 返回状态码600，msg:查询数据为空 |

表19 接口压力测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 |
| 1 | 接口压力测试 | 100线程1700请求 | 通过率70.59%  有权限70.59%  无权限29.41% | 通过率70.59%  有权限70.59%  无权限29.41% |
| 2 | 300线程5100请求 | 通过率70.59%  有权限70.59%  没有权限29.41 | 程序崩溃（内存过少） |

测试报告如下图：



图61 100线程压力测试报告图

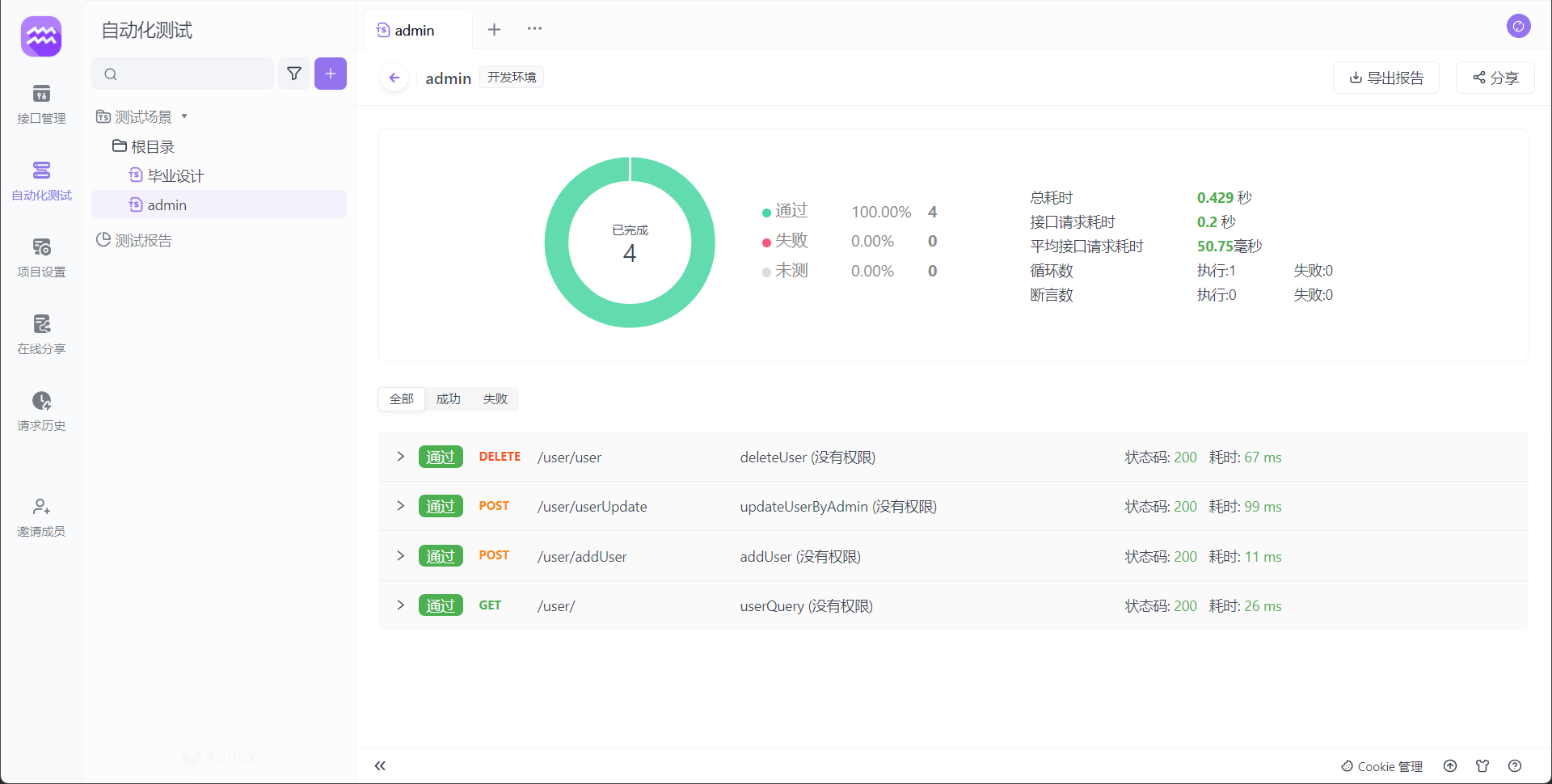


图62 admin测试报告图

## 6.5 测试总结

通过对系统部分主要功能模块设计了测试用例执行自动化测试和压力测试，未发现有明显的恶性bug，整个系统通过了相对高的并发测试，具有一定的可靠性和健壮性，整个系统利用Slf4j日志和try-catch异常捕获机制捕获错误打印。前端面向用户使用element-ui的Message弹窗显示错误/业务错误，对用户来说相对友好，对开发来说也方便简洁。

# 7 总结与展望

自从入学以来，从入门课程的《网络空间安全导论》开始，本人就萌生了以网络空间安全为职业方向的想法，但是前期的课程更多偏向于密码学方面，自学的土壤又远远贫瘠于计算机类的其他方向，导致本人的目标中道崩殂，在大二那年转向了Java开发方向的怀抱，我相信不少同好同学者如此，直到开始准备毕业设计定题前夕，我决定用我学到的知识实践解决我遇到的问题。  
 本次毕业设计，从最开始定题，到决定使用大三学习到的网络安全技术工具—KaliLiunx，再决定开发环境，到整个系统从0到1的过程，均是本人自行完成，在这次毕业设计里，我不仅仅用到了课堂上学到的网络安全相关的理论知识，也使用了自己通过实习得到的实践规范进行开发，整合了我目前该有的能力，使用了时下较新的技术，VueJS、SpringBoot、MySql数据库、JDK17、MongoDB、WebSocket通讯，本人的职业方向选择了Java开发工程师，这些技术也是一位Java开发必备的技能。

本次毕业设计，本来是准备尽善尽美，但是没有什么东西总是能尽善尽美的，目前尚有如下缺陷：

1. 系统各个地方使用的搜索功能基于MySql,在数据量小的时候仍不明显，一旦社区运营成功存活在市场，数据量将大大提高，此时搜索功能的响应速度将被拖慢，如果同时使用ElasticSearch数据库存储与搜索，依赖于其倒排索引的特性，搜索速度哪怕在数据量大的情况仍旧能快速响应。
2. 由于本地内存（8G）限制，系统后台只配备了一个靶机，完成实验尚且可行，一旦大规模部署上线运行，肯定需要多个靶机完成实践。

虽然不尽完美，但是独立完成一个系统的前后端对我本人亦是是我受益良多，小半年间断性的学习与开发，解决bug，处理业务逻辑，让我有了以下收获：

1. Vue.js从入门到熟练，从一个只会html-jquery-css前端三件套的后端学习者到Vue框架的熟练使用，是一个一个bug的解决时长和一次一次搜索错误到明了问题关键堆叠起来的，这无疑给我的职业发展道路增加了一条其他方向的可能，也钩起我对前端的兴趣。
2. 对前后端信息传递的更深入了解，当本人只学会后端的时候，和前端工程师一起开发其他项目的时候总是会因为请求的错误争论问题是出现在前端还是后端，当本人真的接触前端和后端的代码编写时，对于请求编写的规范更加深入，往往能更快定位到问题的关键。

一番言了，希望日后有时间、精力和支出能力能真正负担起这个社区的良好运行。

# 参 考 文 献

1. Mythily, M., A. Samson Arun Raj, and Iwin Thanakumar Joseph. "An analysis of the significance of spring boot in the market." 2022 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2022.
2. 熊永平.基于SpringBoot框架应用开发技术的分析与研究[J].电脑知识与技术,2019,15(36):76-77.DOI:10.14004/j.cnki.ckt.2019.4290.
3. Pimentel V, Nickerson B G. Communicating and displaying real-time data with websocket[J]. IEEE Internet Computing, 2012, 16(4): 45-53.
4. Győrödi C, Győrödi R, Pecherle G, et al. A comparative study: MongoDB vs. MySQL[C]//2015 13th international conference on engineering of modern electric systems (EMES). IEEE, 2015: 1-6.
5. 周常志,甘恒.基于SpringBoot的智慧就业服务平台的设计与实现[J].电脑知识与技术,2021,17(28):182-184+187.DOI:10.14004/j.cnki.ckt.2021.3040.
6. 陈倩怡,何军.Vue+Springboot+MyBatis技术应用解析[J].电脑编程技巧与维护,2020(01):14-15+28.DOI:10.16184/j.cnki.comprg.2020.01.005.
7. 杨良斌,周新丽,刘思涵,等.大数据背景下网络空间安全人才培养机制与模式研究[J].情报杂志,2016,35(12):81-87+80.
8. 贾晓辉,韩恺,乐嘉锦.基于UML的系统需求分析[J].计算机应用与软件,2007(08):72-74.
9. 秦晓.数据流图的形式规范[J].软件学报,1994(05):39-45.
10. 屈喜龙.UML及面向对象的分析与设计的研究[J].计算机应用研究,2005(09):74-76.
11. 谢幼如,马秀芳,余红.“网络教学设计”专题学习网站的设计与开发[J].中国电化教育,2003,(05):68-71.
12. 丁智斌,石浩磊.关系数据库设计与规范化[J].计算机与数字工程,2005,(02):114-116.
13. 杨中书,刘臣宇.基于E-R模型的关系数据库设计方法[J].价值工程,2014,33(30):242-243.DOI:10.14018/j.cnki.cn13-1085/n.2014.30.134.
14. 贾晓辉,韩恺,乐嘉锦.基于UML的系统需求分析[J].计算机应用与软件,2007(08):72-74.
15. 单锦辉,姜瑛,孙萍.软件测试研究进展[J].北京大学学报(自然科学版),2005(01):134-145.DOI:10.13209/j.0479-8023.2005.020.

致 谢

感谢所有人！