

**本科实验报告**

|  |  |
| --- | --- |
| 课程名称： | B/S体系软件设计 |
| 姓 名： | 周炜 |
| 学 院： | 计算机科学与技术学院 |
| 系： | 计算机科学与技术系 |
| 专 业： | 计算机科学与技术 |
| 学 号： | 3210103790 |
| 指导教师： | 胡晓军 |

2023年 12月 12日

**浙江大学实验报告**

课程名称： B/S体系软件设计 实验类型： 课程大作业

实验项目名称： 物联网设备管理平台

学生姓名： 周炜 专业： 3210103790 学号： 3210103790

同组学生姓名： 指导老师： 胡晓军

实验地点： 实验日期： 年 月 日

目录

[1 测试介绍 3](#_Toc153746630)

[1.1 测试对象 3](#_Toc153746631)

[1.2 测试范围 3](#_Toc153746632)

[1.3 测试环境 4](#_Toc153746633)

[2 功能性测试 4](#_Toc153746634)

[2.1 登陆界面测试 4](#_Toc153746635)

[2.2 注册界面测试 7](#_Toc153746636)

[2.3 用户界面测试 13](#_Toc153746637)

[2.4 设备管理界面测试 15](#_Toc153746638)

[2.5 设备消息记录接受 24](#_Toc153746639)

[3 非功能性测试 25](#_Toc153746640)

[3.1 SQL注入攻击测试 25](#_Toc153746641)

[3.2 移动端适配测试 26](#_Toc153746642)

[3.3 docker部署测试 31](#_Toc153746643)

[3.4 服务器部署测试 32](#_Toc153746644)

# 测试介绍

## 测试对象

IoT System，全称为物联网系统，是一个基于互联网的延伸和扩展，将各种信息传感设备与网络结合形成一个庞大网络，实现人、机、物在任何时间、任何地点的互联互通的项目。物联网在工业、农业、环境、交通、物流和安保等基础设施领域得到广泛应用，推动了智能化发展，优化资源分配，提高行业效率。同时，物联网也在家居、医疗健康、教育、金融、旅游等领域改善了生活质量。

IoT System项目是为2023-2024秋冬学期《B/S体系软件设计》课程而创建的。该项目的目标是为用户提供友好的界面，适配手机移动端，并提供必要的软件项目文档，以帮助学习web应用开发技术和流程。IoT System采用了B/S结构设计，实现了数据接收、统计分析，以及用户信息和设备配置的修改。这个项目包括了web前后端、MQTT服务器以及相关文档，全部由一人独立完成。

此项目的测试对象是物联网管理系统，主要测试内容包括功能性测试和非功能性测试，以确保系统的功能正确实现。功能性测试将验证系统的各项功能是否按照预期工作，而非功能性测试则关注系统的性能、安全性和可用性等方面。

## 测试范围

测试范围包括对系统进行功能层面的黑盒测试，旨在评估系统的基本功能和性能。测试的主要模块包括登录注册、个人信息管理、设备配置、历史消息和统计信息等多个模块，每个模块下还包含多个具体功能的测试。测试将关注这些模块和功能的正确性和性能表现。

## 测试环境

操作系统 windows10 专业版

CPU 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz，23...

RAM 16GB

# 功能性测试

## 登陆界面测试

登陆界面需要验证用户账号和密码的正确性，为此我设计了以下几个测试案例，只提供一些错误性样例，正确性样例可以具体见于**用户手册**

**账号正确密码错误**

密码格式错误



邮箱格式错误



输入存在的用户账号和错误的密码并点击登录之后后，发现输出结果如下：



**账号不存在**

输入一个不存在的账号，并尝试登录，发现登录也失败了



该项测试也通过，表明登陆验证功能正常。

**空输入**

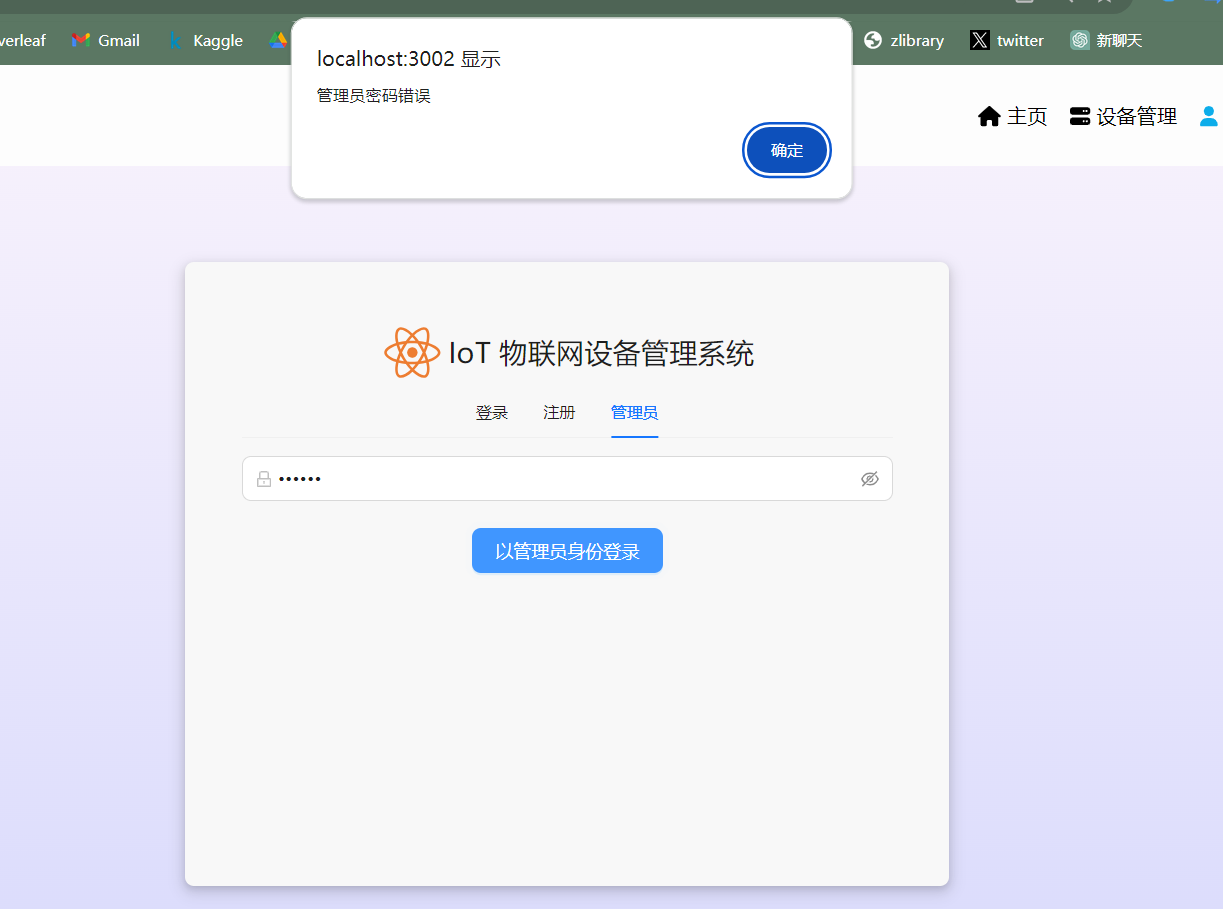
当不输入任何内容直接提交的时候，表单会检测出并提醒用户填写好表单之后再提交（此时点击登录不会有任何反应）：



只输入用户名不输入密码也一样无法登录：



Root模式下登录如果密码错误也无法登录



## 注册界面测试

注册界面要求用户填写各项已经存在的信息，并提交注册表单。其中，账号和邮箱必须是唯一的，因此我进行了以下功能性测试来验证注册界面的功能

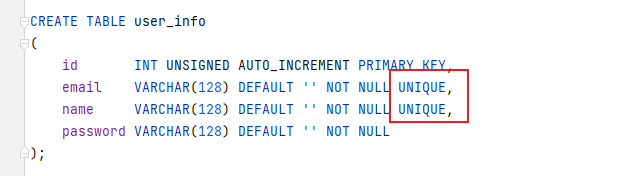
**空输入**

有空输入的表单无法提交 ，注册表单中所有项都是必填的。当有空栏的时候就会有对应的提示。

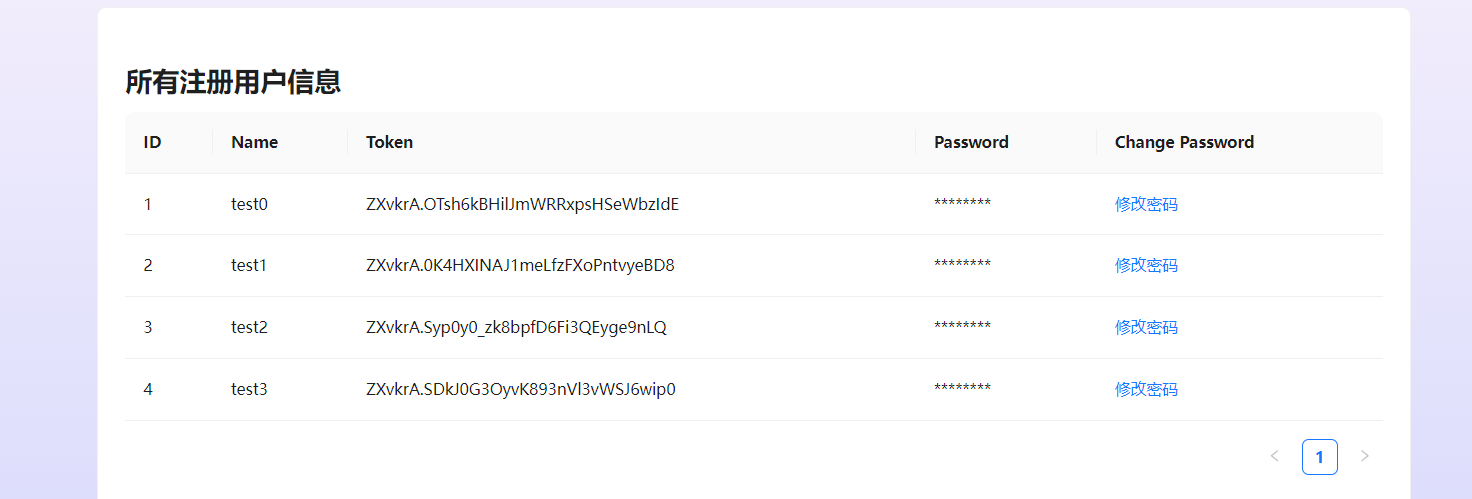


**重复的用户名或者邮箱**

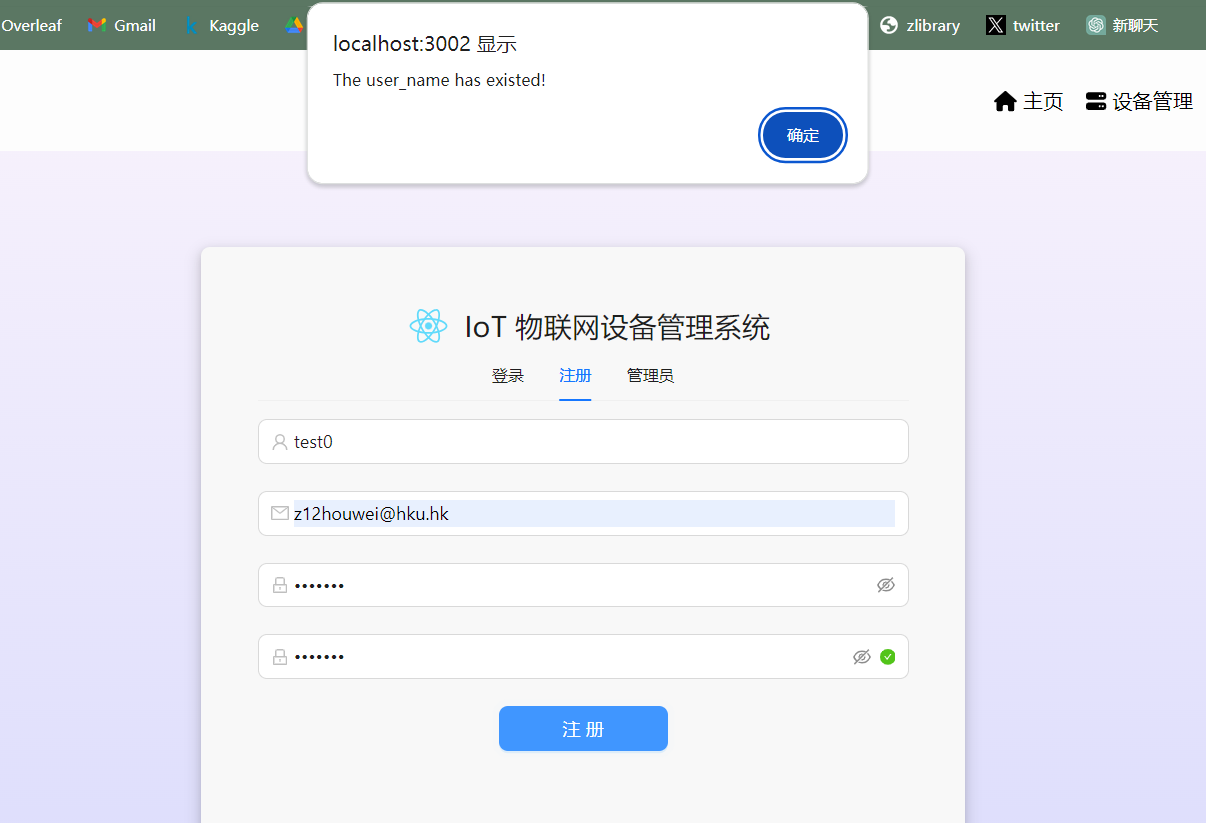
由于我的数据表中用户名和邮箱都是unique的，如果相同也会注册失败，并且会返回相应的提示



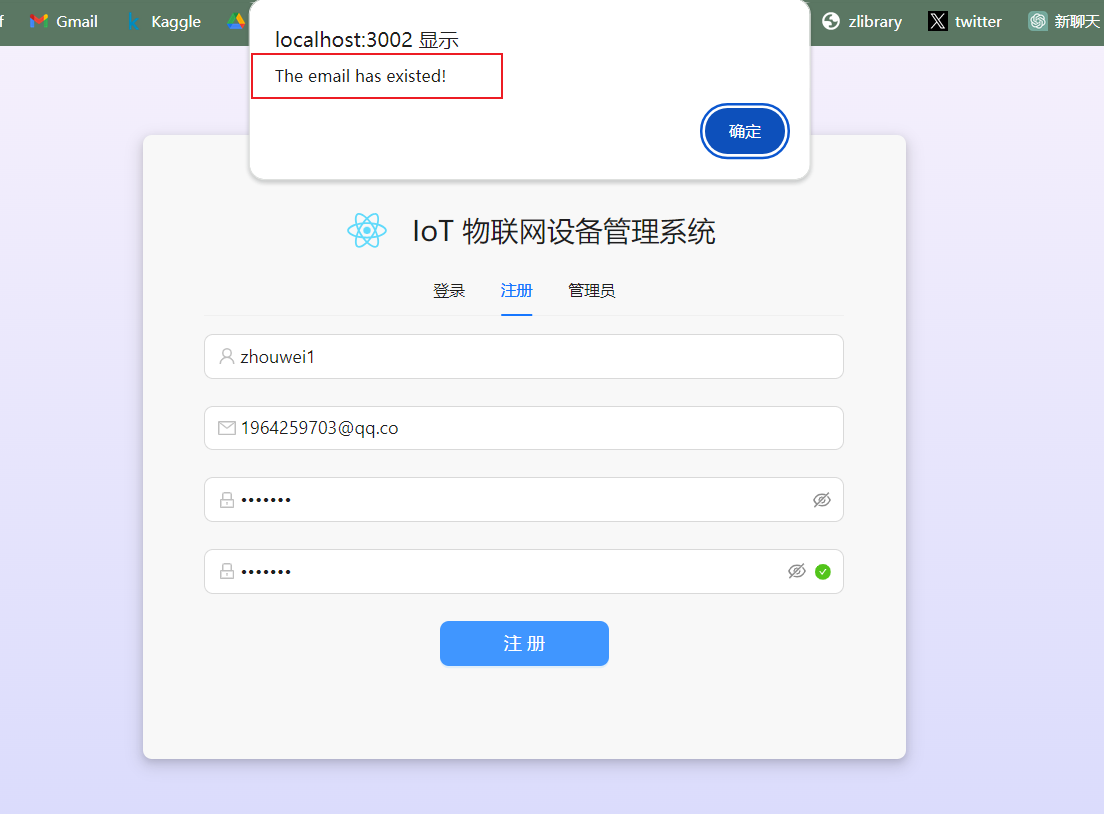
我们现在root中看一下已经存在的用户



用户名存在



邮箱存在



**前后不一致的密码**

如果输入密码和确认密码两栏中的密码不一致，表单将进行验证并显示警告信息，同时禁止提交表单：



**密码长度或者格式不符合要求**

密码长度过短

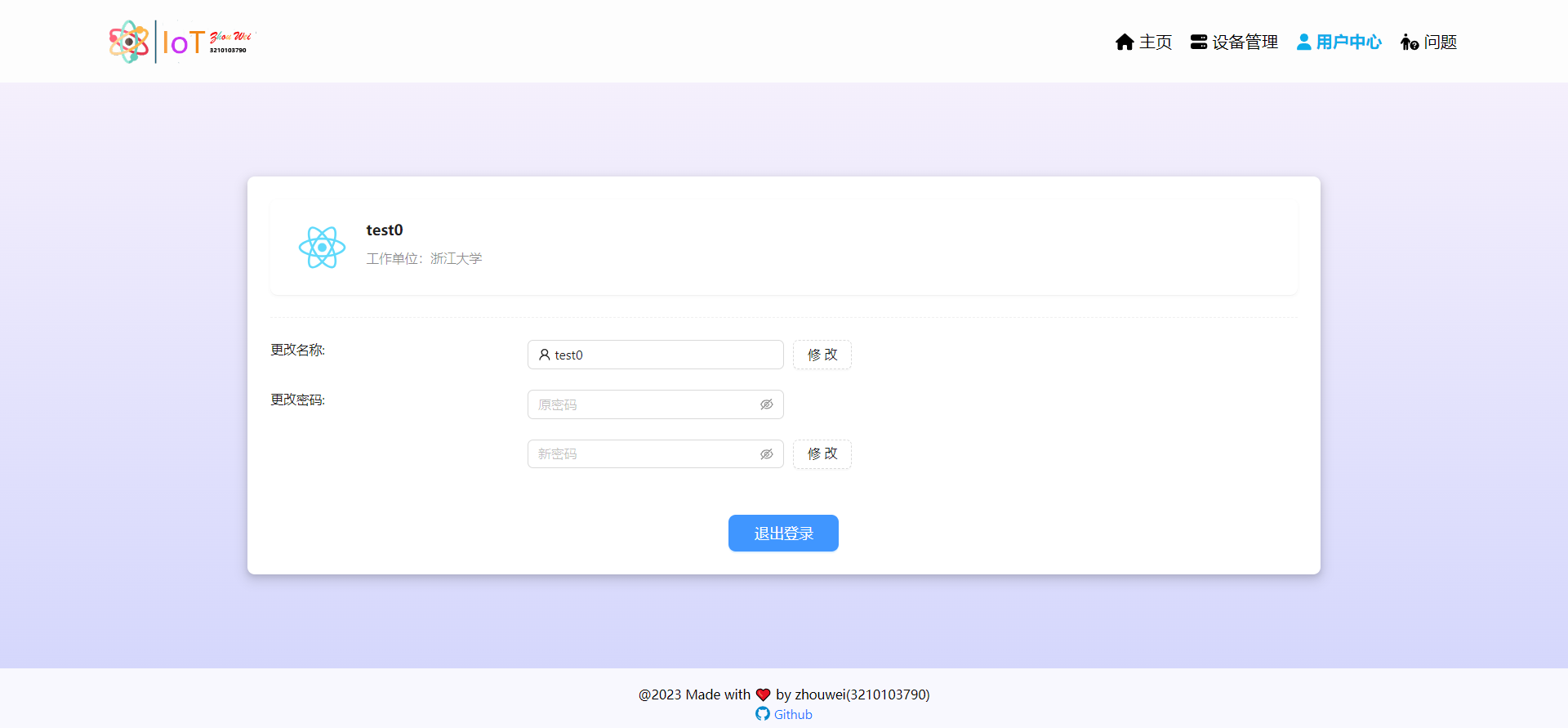


密码格式不正确

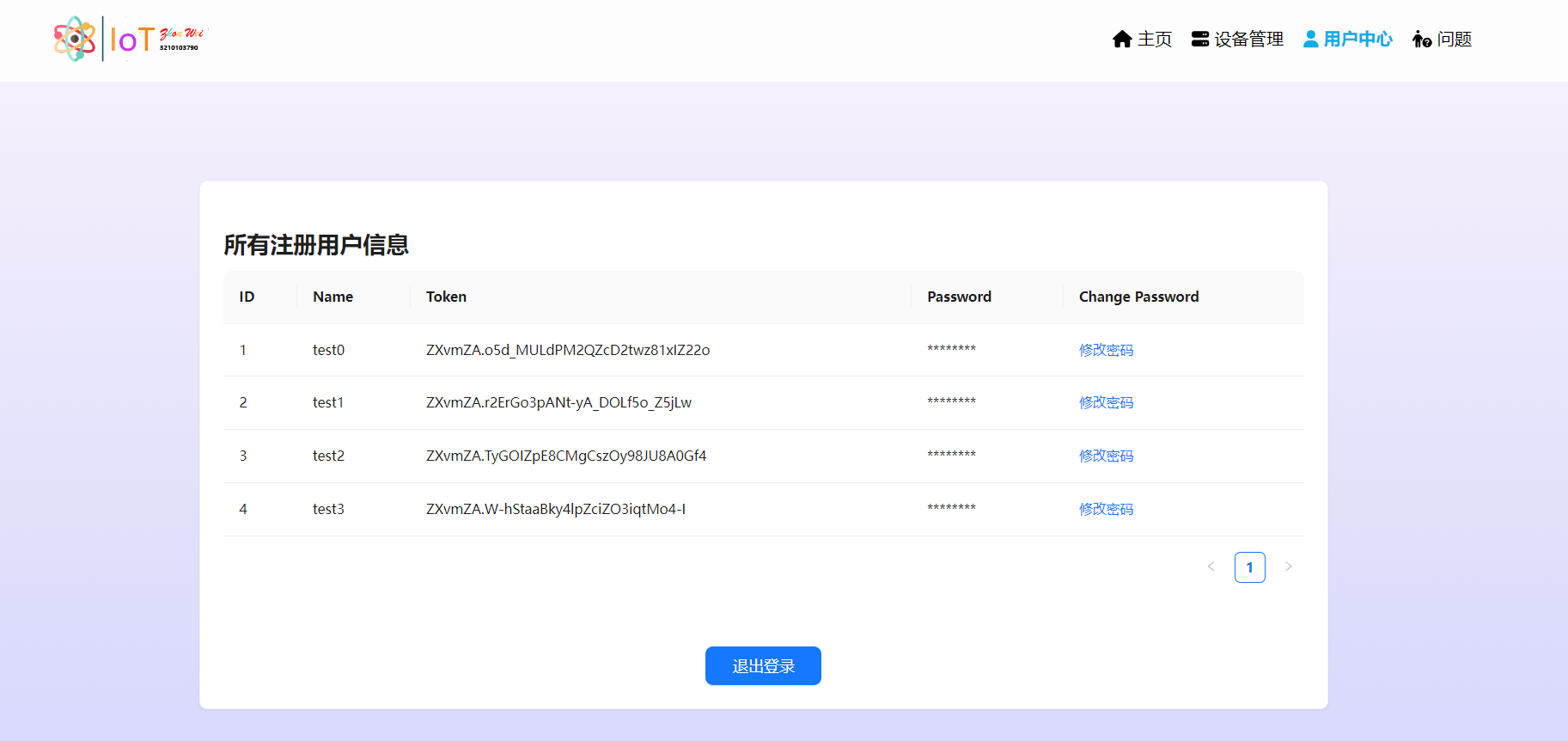


**退出登录测试**

在用户登录后，他们可以在网站的用户重谢页面看到用户登录状态栏。经过多次验证和使用多个账号进行测试后，发现可以轻松地清除浏览记录并退出网站，然后返回到登录页面。

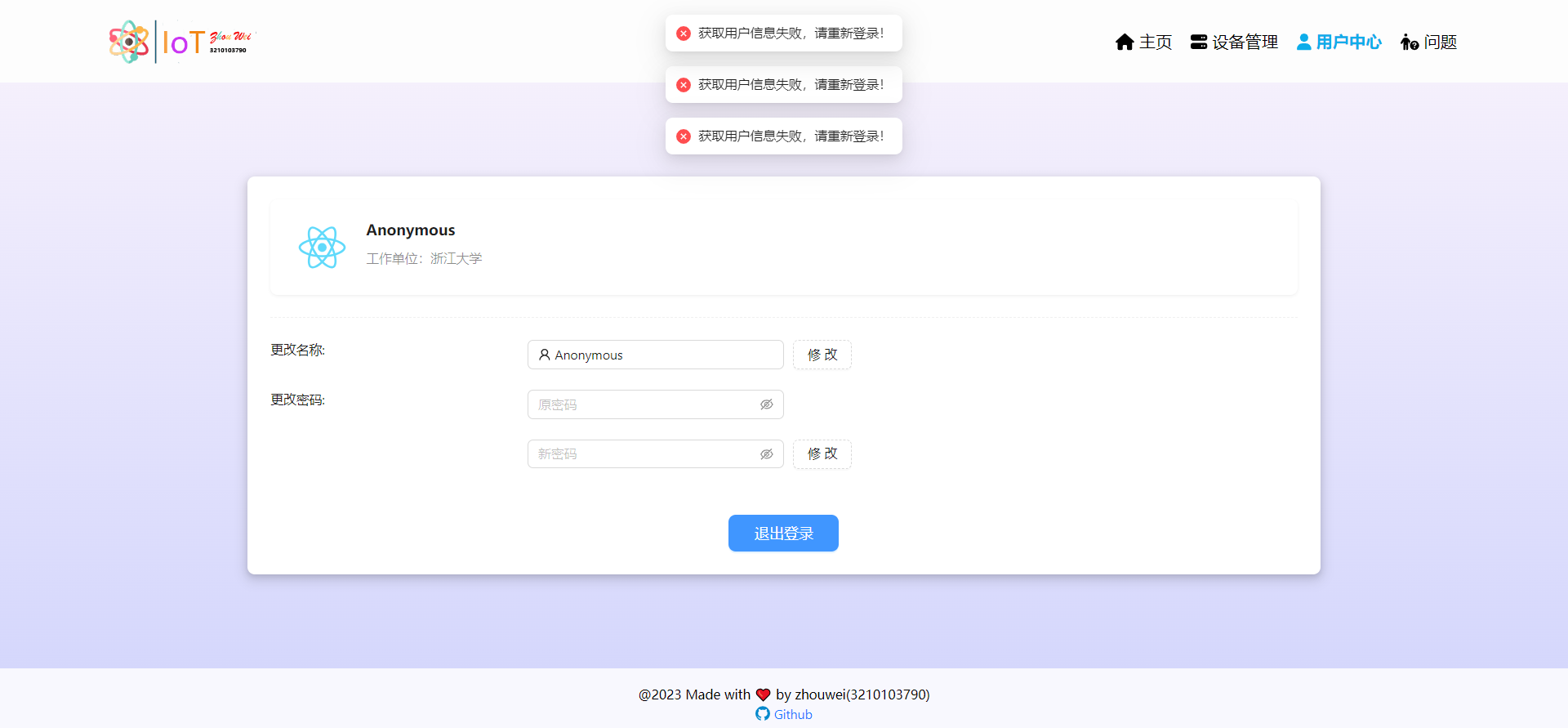


root由于不会存储token，因此刷新之后就会自动退出，回到登录界面



**Token失效自动退出验证**

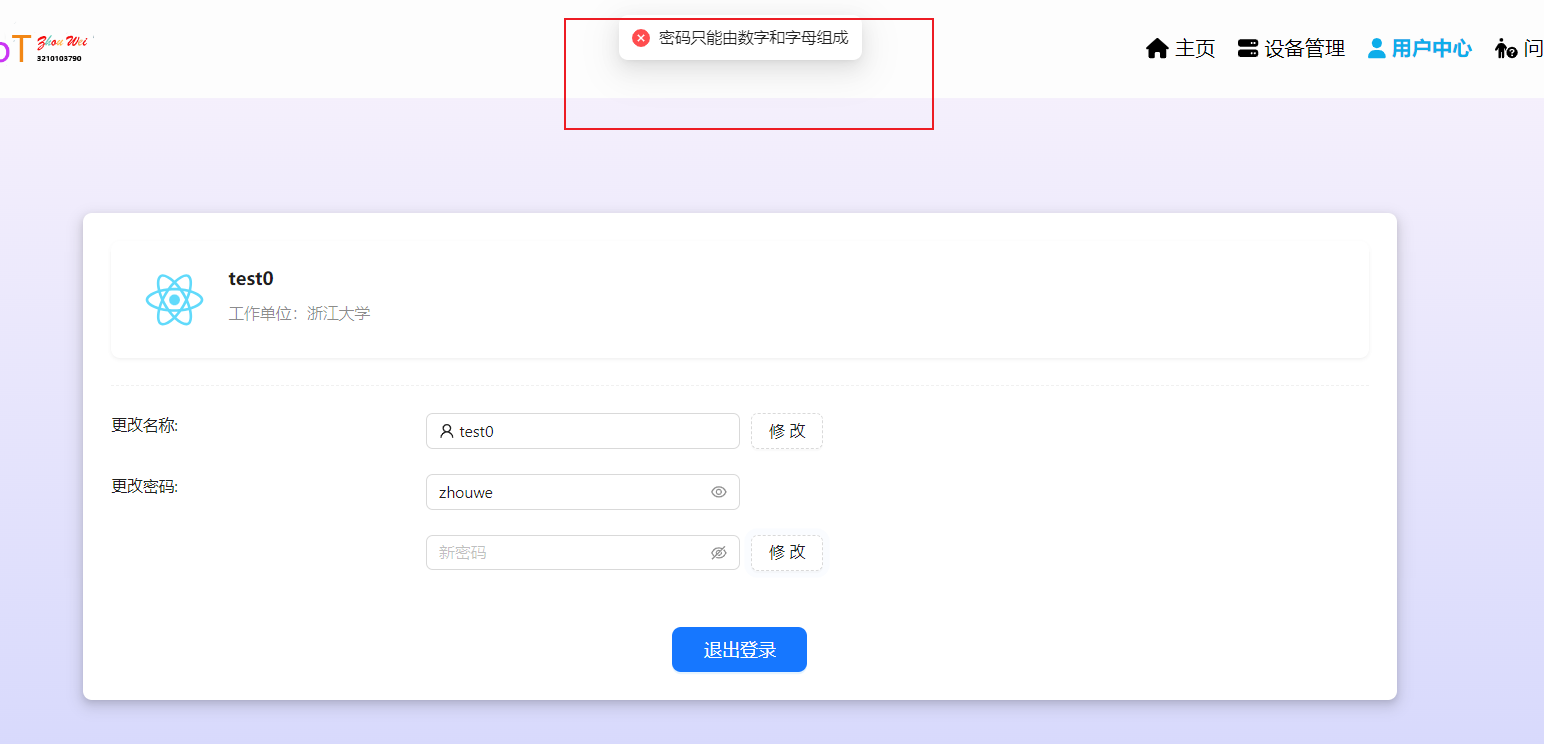
由于本网页的token存在时间为60min，超出该时间后token会自动消息，即用户的登录状态会退出，这里做了验证性实验。可以发现，超时后，用户的名称变成了默认的anonymous， 并且有连续的提示要求重新登录，并且无法拉取上一个用户的设备信息



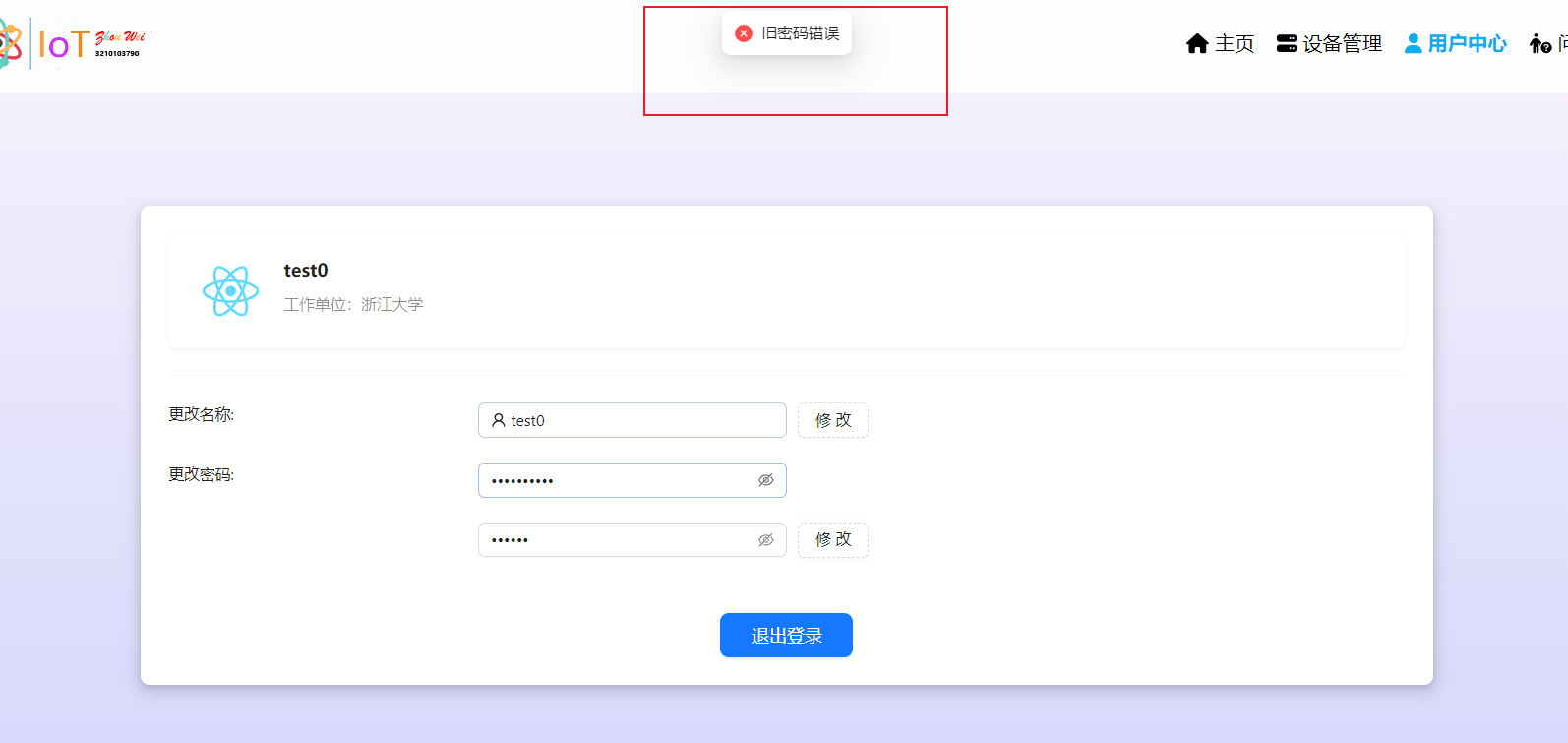
## 用户界面测试

**空输入**

在修改密码的表单中，如果存在空项目（即某些必填字段未填写），无法提交表单，并会显示相应的提示信息

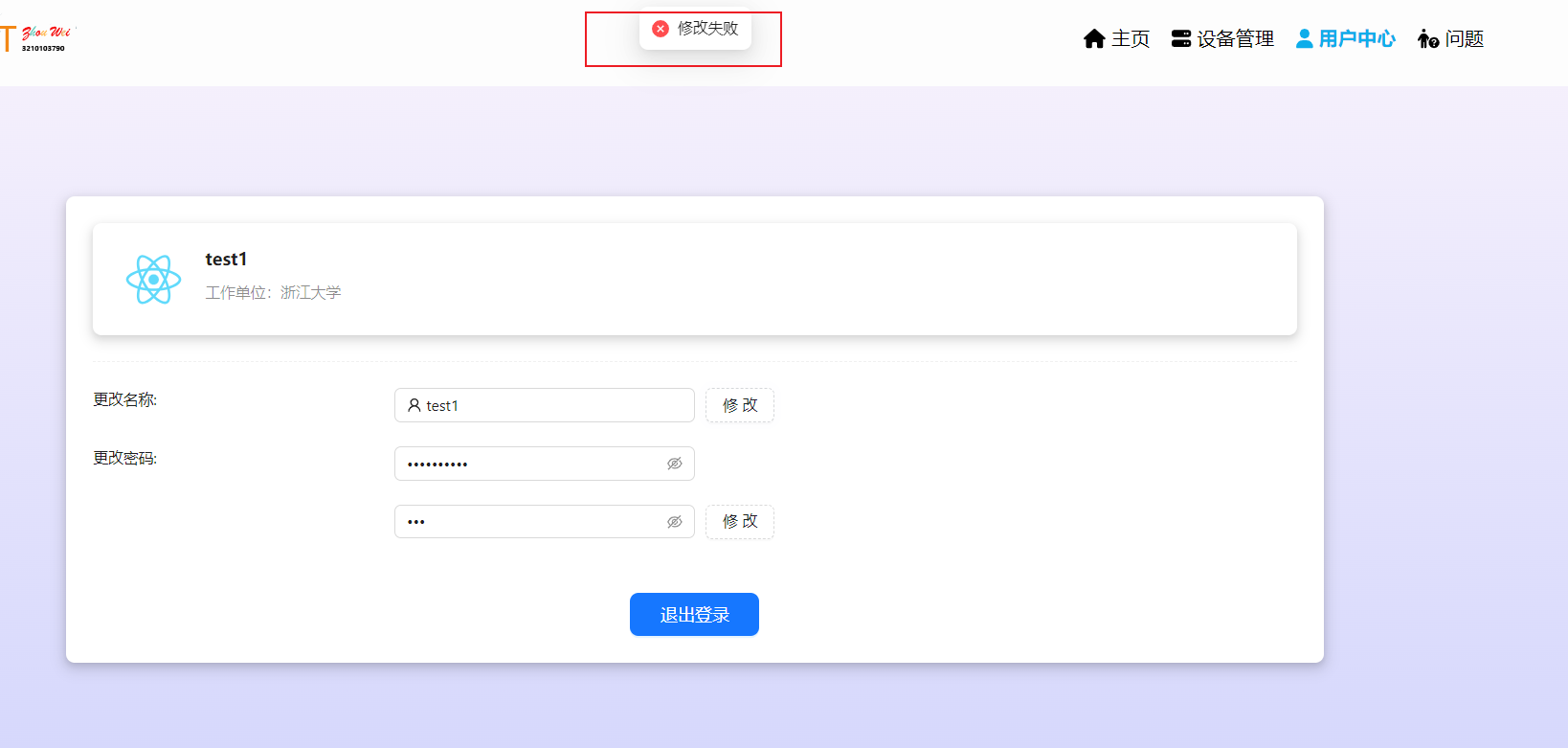


原密码错误



**修改的名称已经存在**

使用存在的用户名时网站会提示该用户名已经被占用，因此信息修改失败，并有相应的消息提示



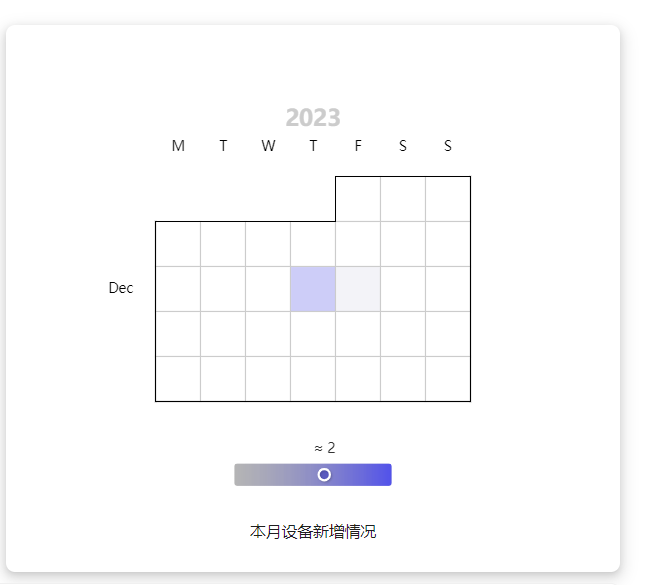
并且立刻显示回到原名称



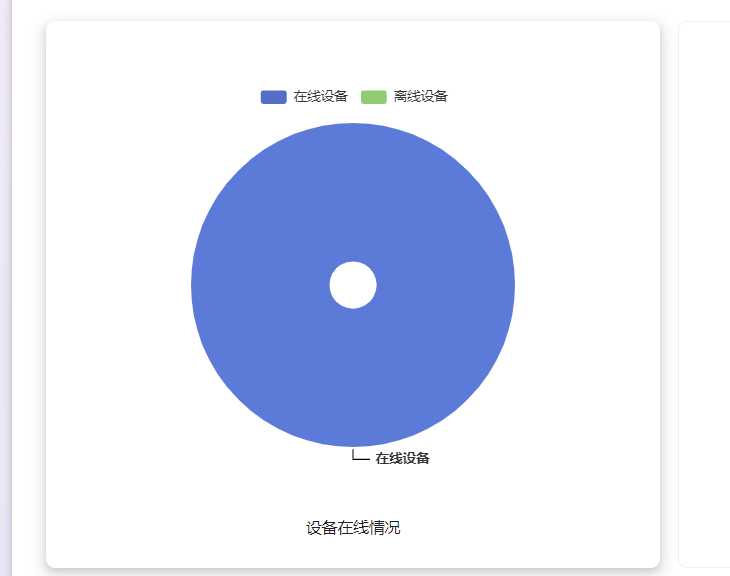
## 设备管理界面测试

**表格动画和交互正常**

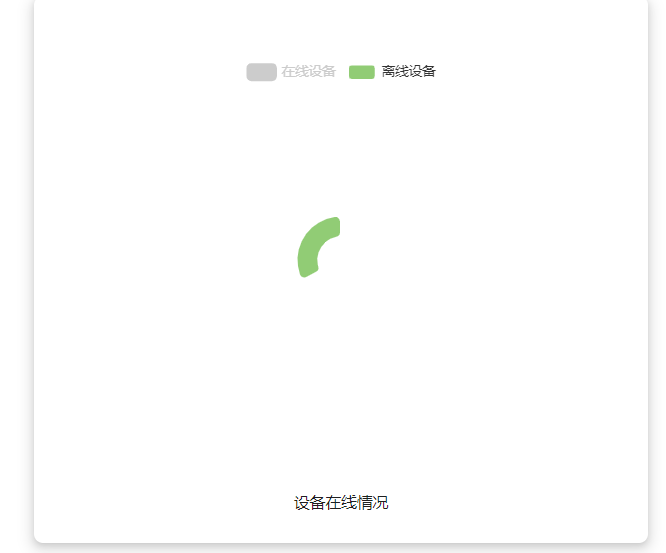
对于日历图，色带交互功能正常



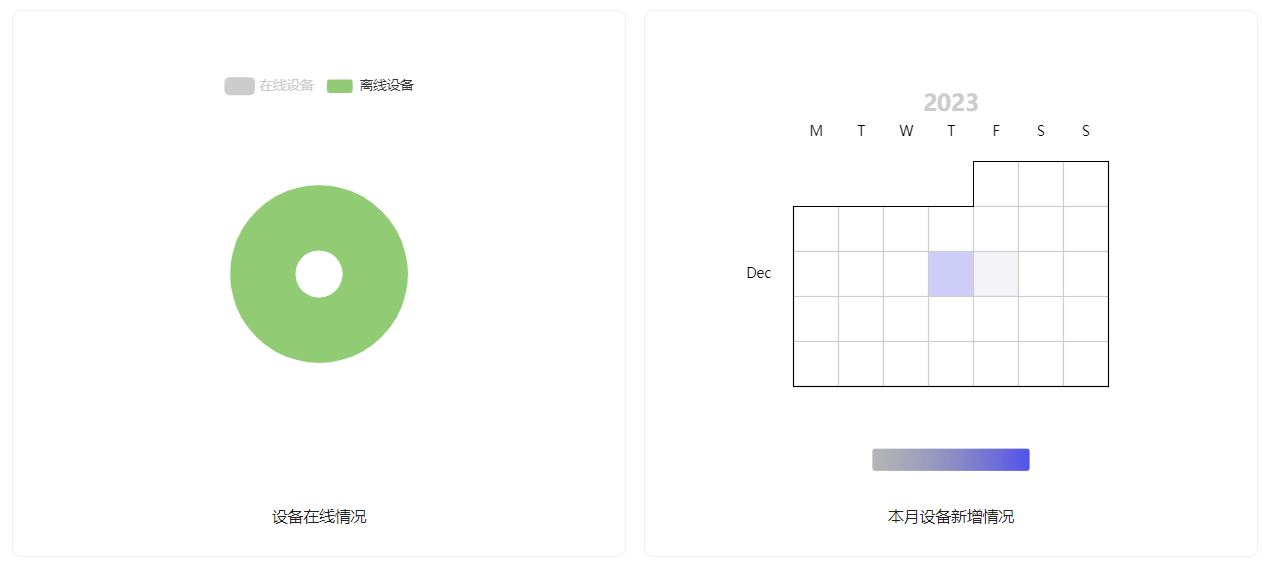
对于饼图



此时点击在线设备之后会隐藏在线设备信息



动画完成后显示最终效果



对于折线图，可以发现鼠标悬浮，可以正确展示信息



点击正常消息可以隐藏掉正常消息的信息，并且发现坐标轴的高度也发生了改变

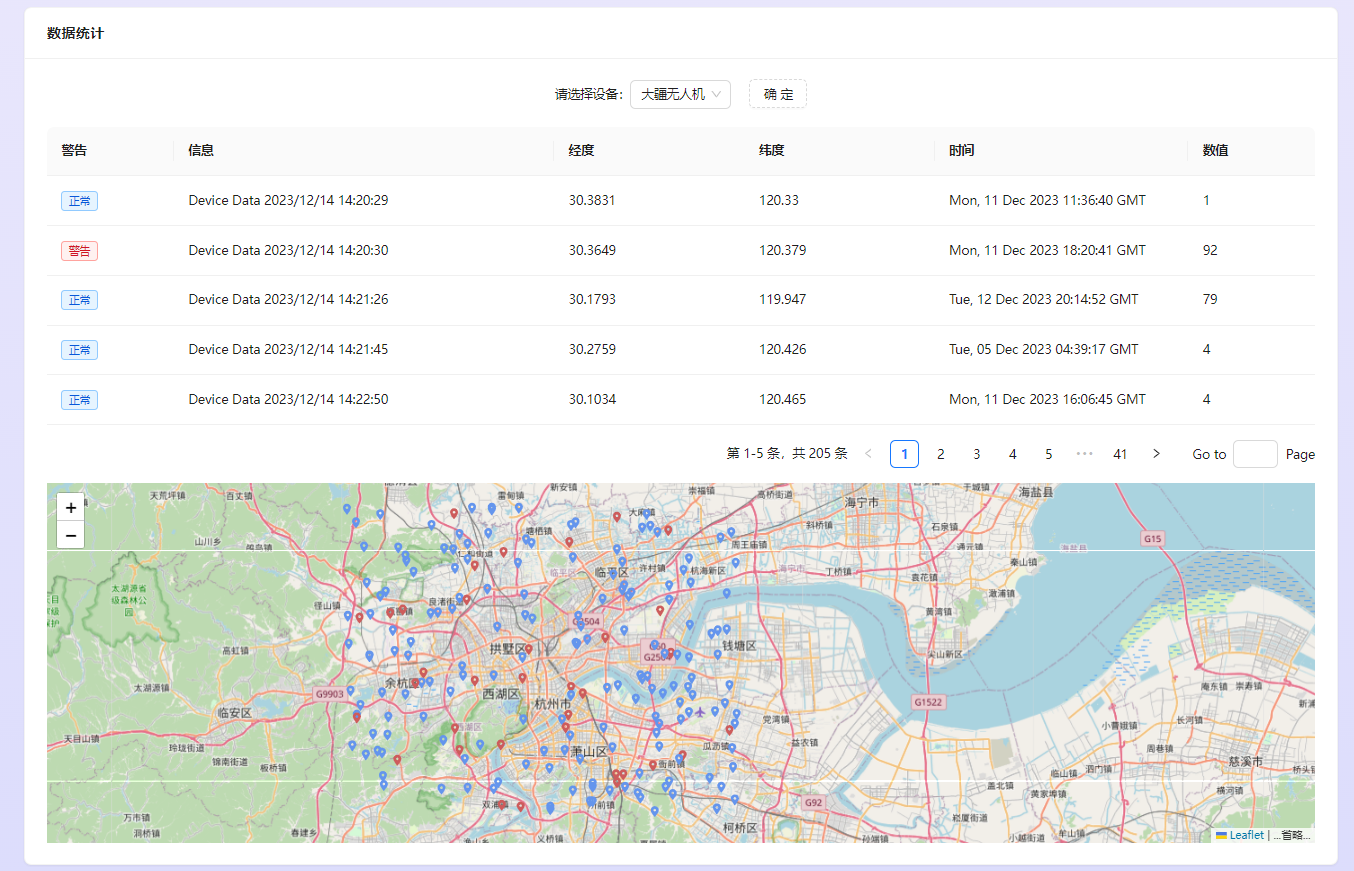


地图显示为



选择设备后数据统计的表格正常显示，并且地图上出现注记





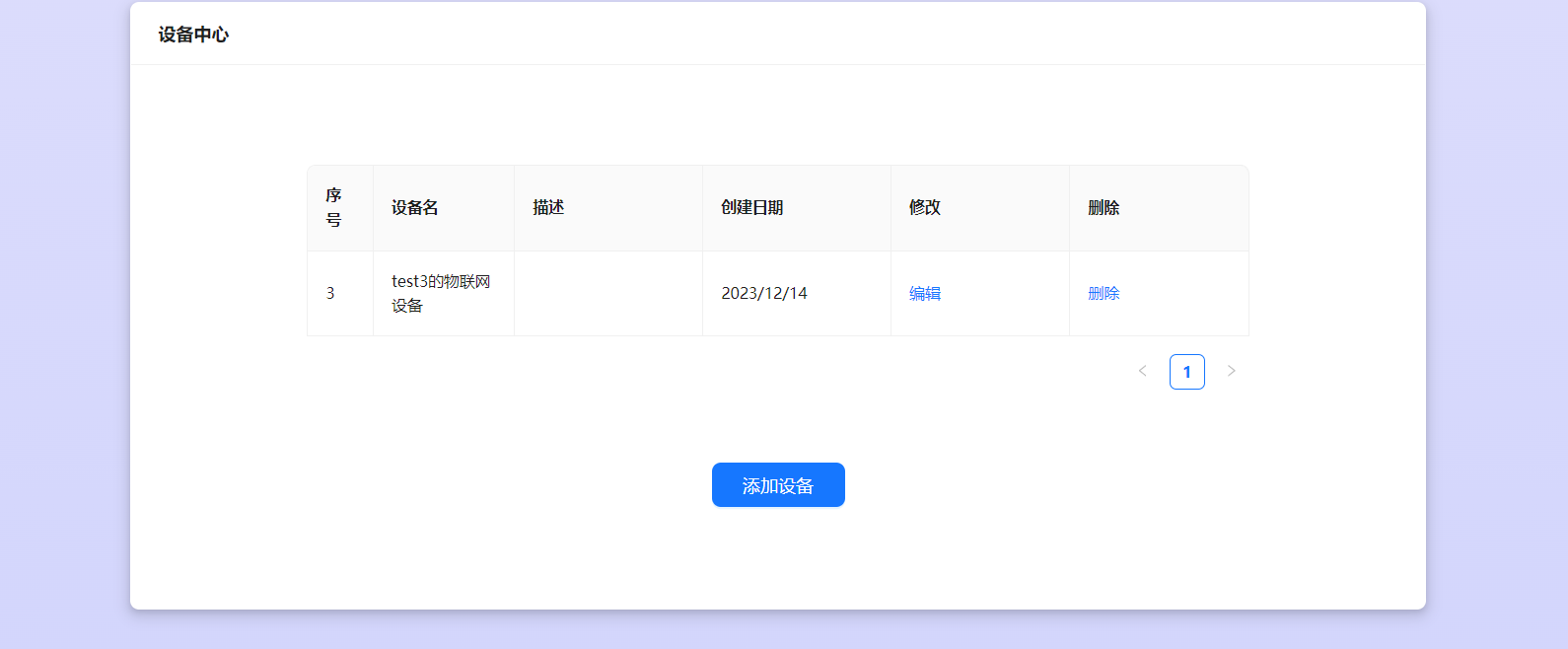
**新增设备的空提交**

与前述情况类似，当在新增物联网设备时，表单中存在任何一项为空时，系统将显示相应的提示信息，并且不允许提交表单

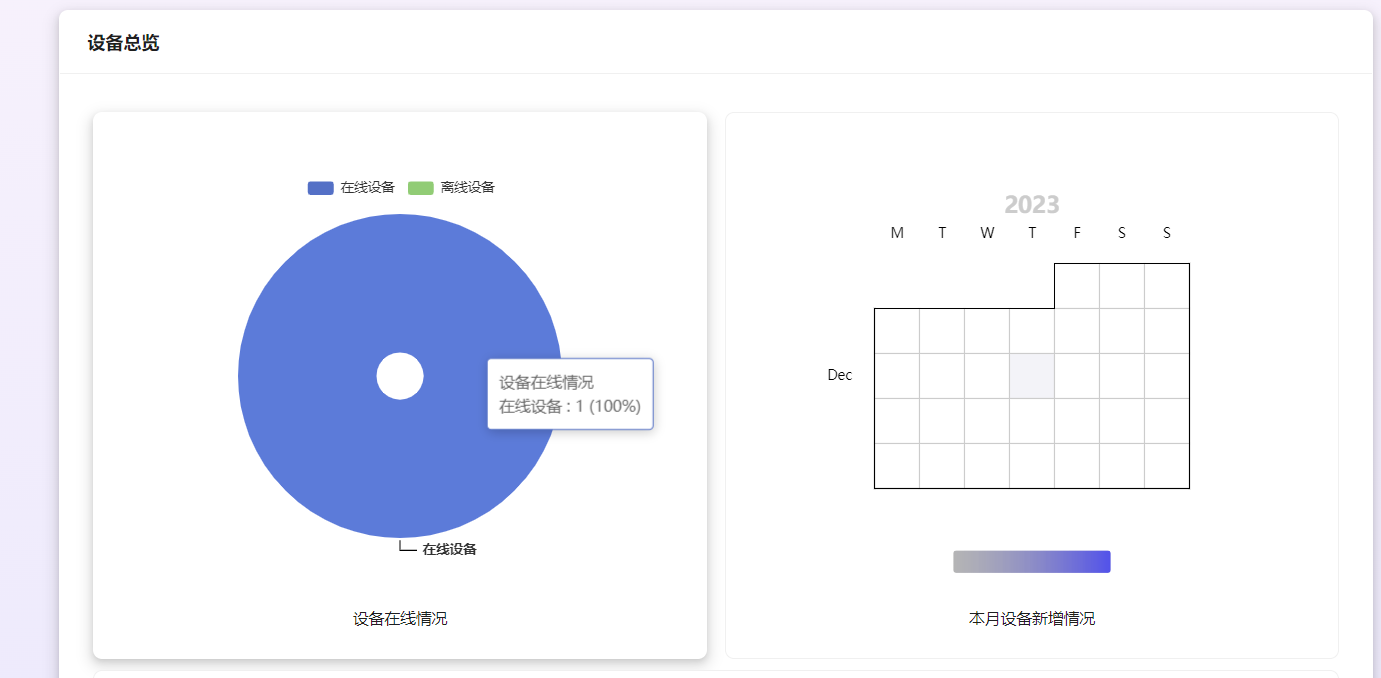


**新增设备之后的图表更新**

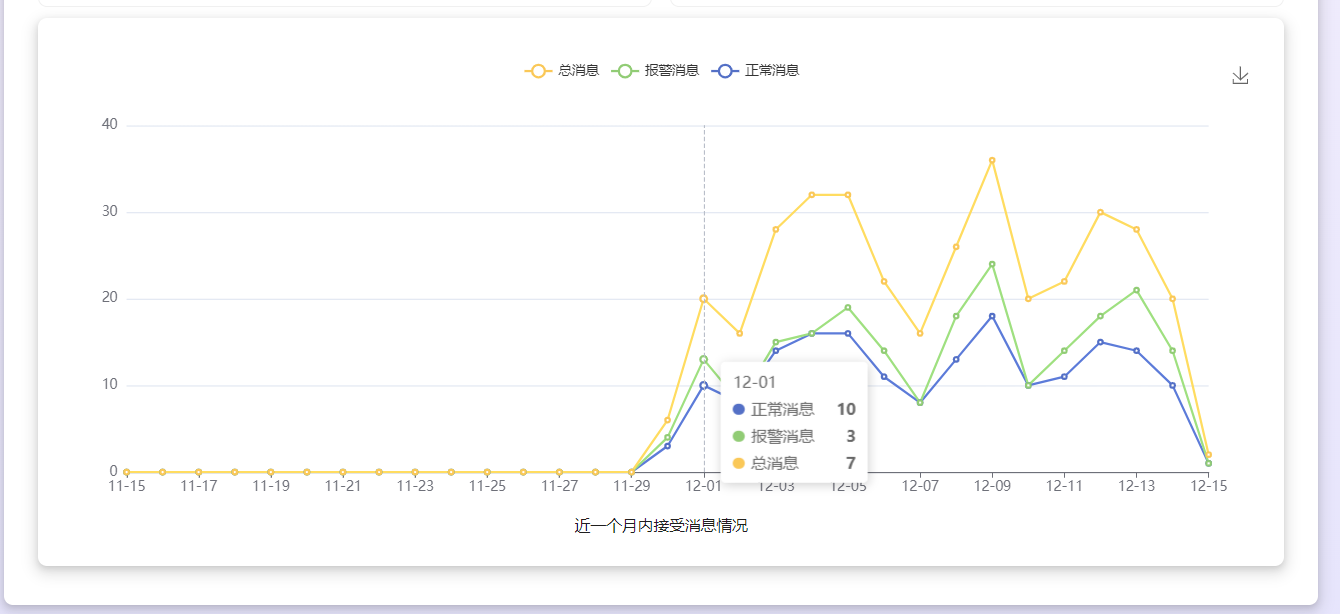
原先只有一台设备



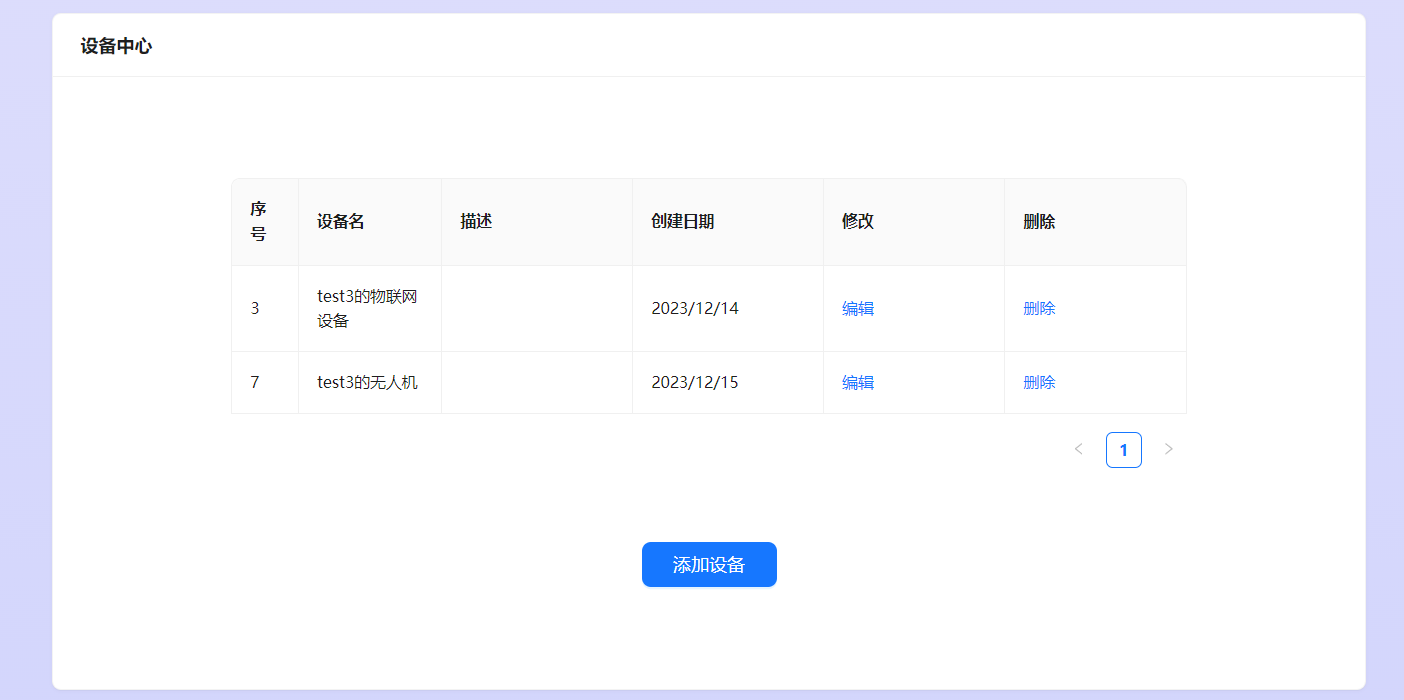
三张图表的显示情况分别如下：







此时添加一台新设备

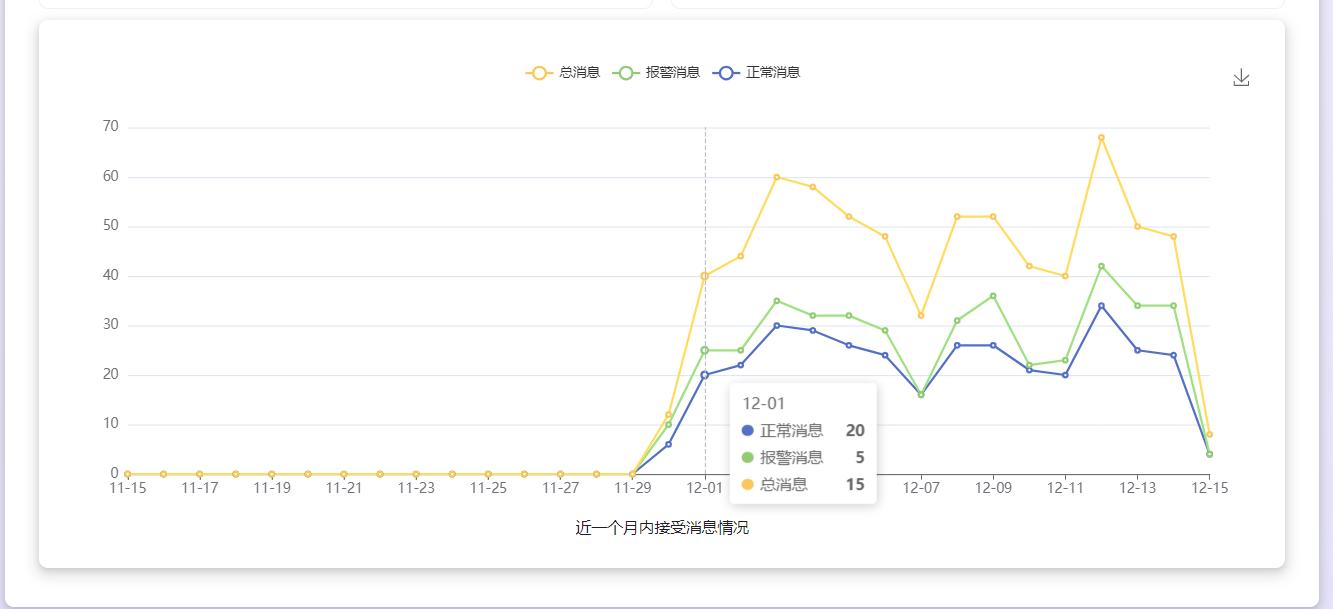


可以发现三张图表已经显示为





柱状图显示的消息数量增加了

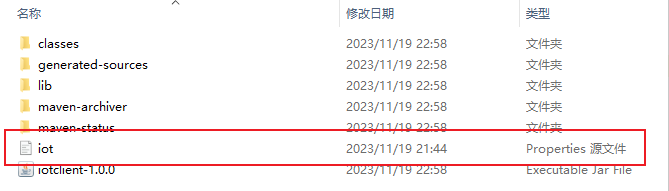


## 设备消息记录接受

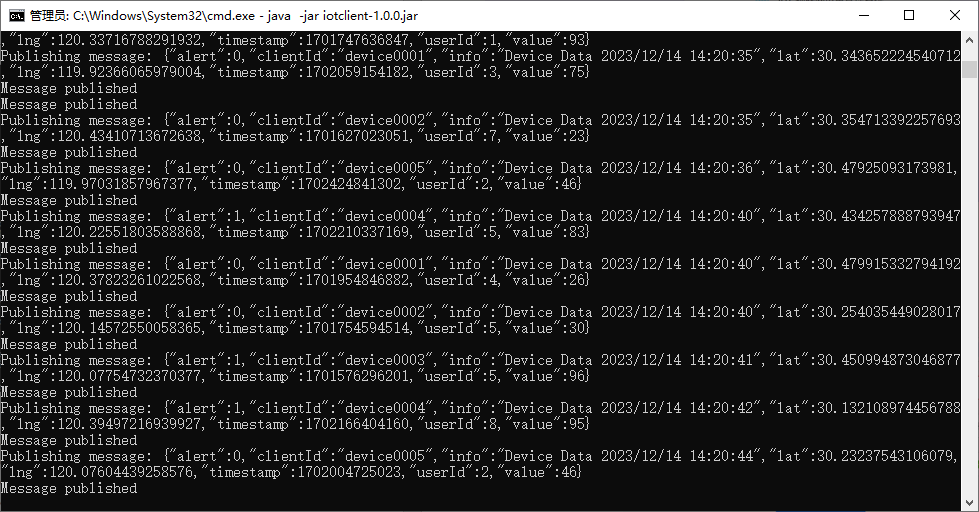
使用老师提供的java程序来进行物联网基础设备的信息发送， 可以使用cmd和下列命令完成编译：

|  |
| --- |
| # 首先要运行后端程序  mvn clean package  # 同时复制一份iot到target  cd target  java -jar iotclient-1.0.0.jar |

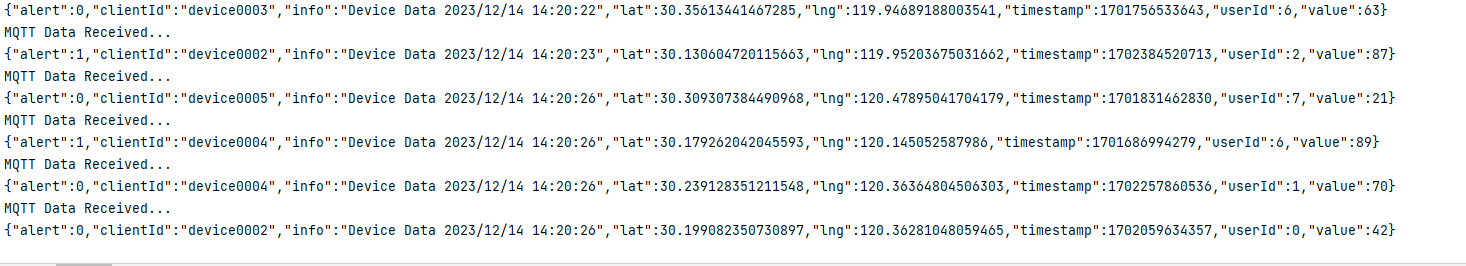
并且注意iot需要拖动一份到target



运行后，cmd的显示如下:



此时后端也相应得做出了反应



可以发现后端的接收和前端的发送能够形成很好的对应关系，说明测试无误

# 非功能性测试

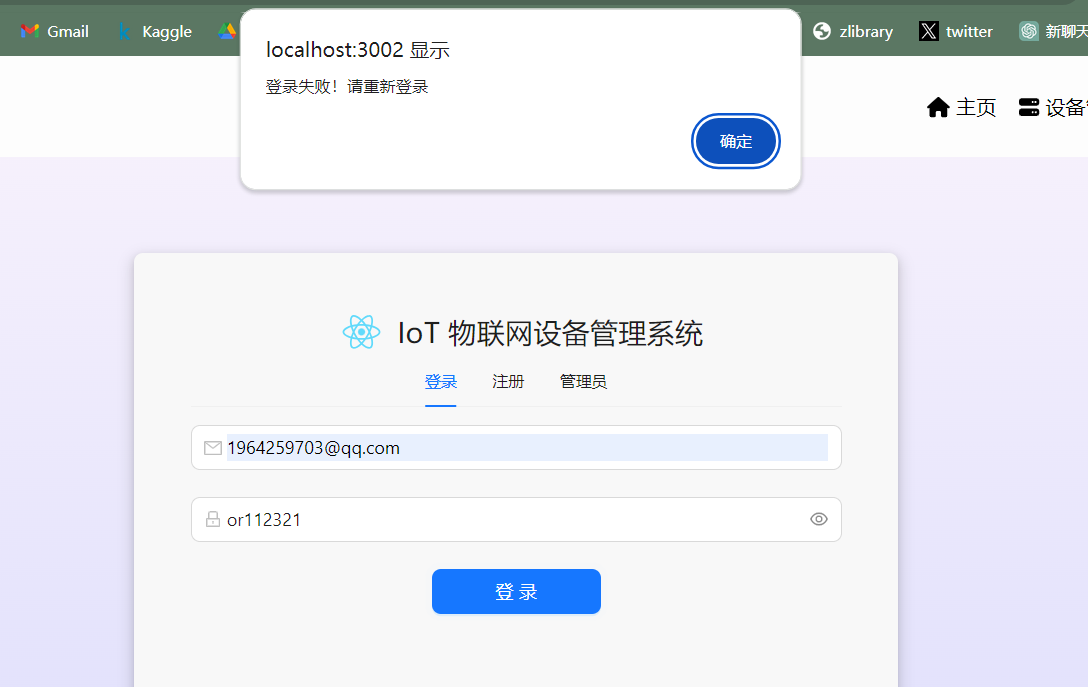
## SQL注入攻击测试

SQL注入攻击是一种常见的攻击方式，通过在表单中提交一些拼接后的非法SQL语句，攻击者试图达到网络攻击的目的。在本次测试中，以登陆界面的表单为例，我们输入了用户名后，在密码一栏输入了 **“or 1=1-**以验证系统是否容易受到SQL语句攻击。

可以发现他首先不满足密码的格式要求



修改代码，取消格式要求之后仍然失败，说明MD5这样的加密方式使得SQL注入攻击失效

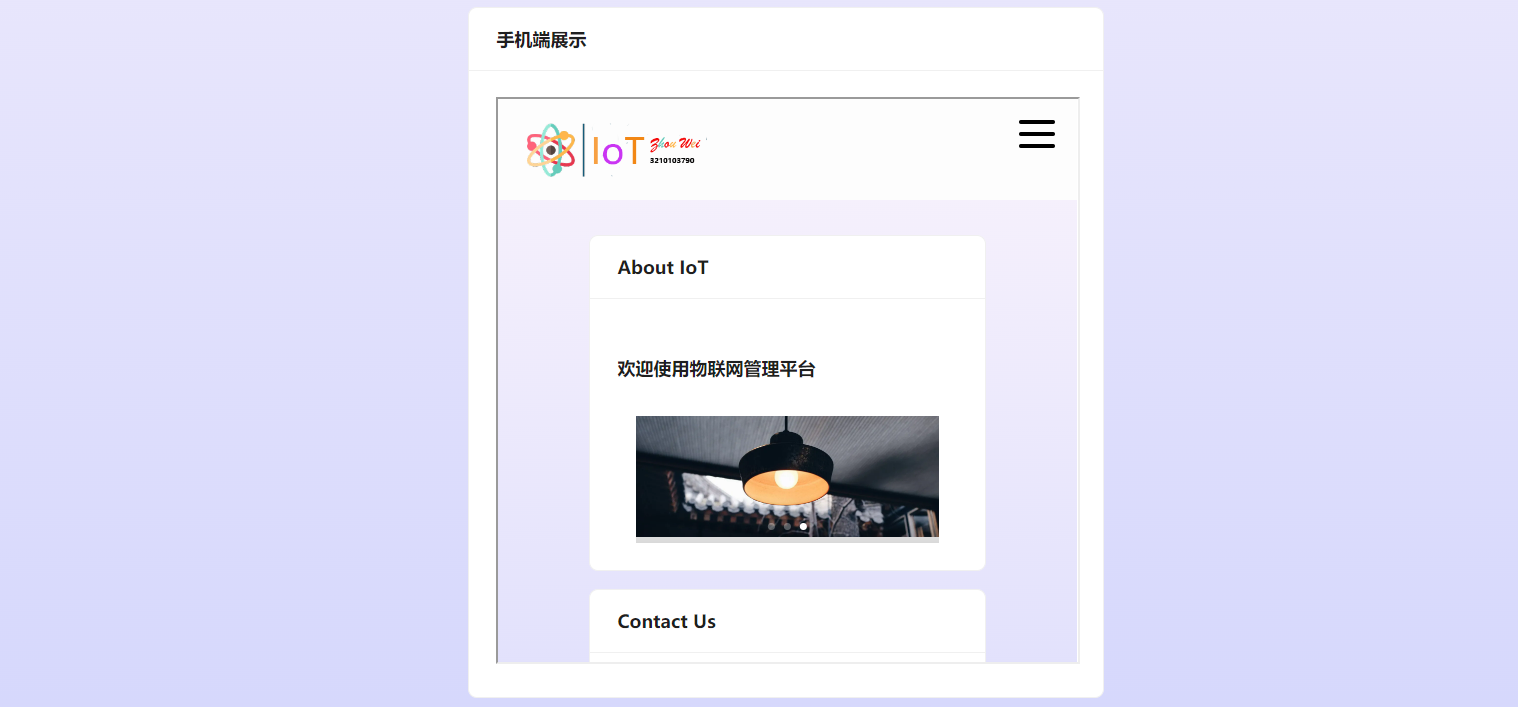


## 移动端适配测试

我在网站的QA栏中还嵌套了一个手机端页面，可以浏览手机端的显示效果



移动端的主页效果



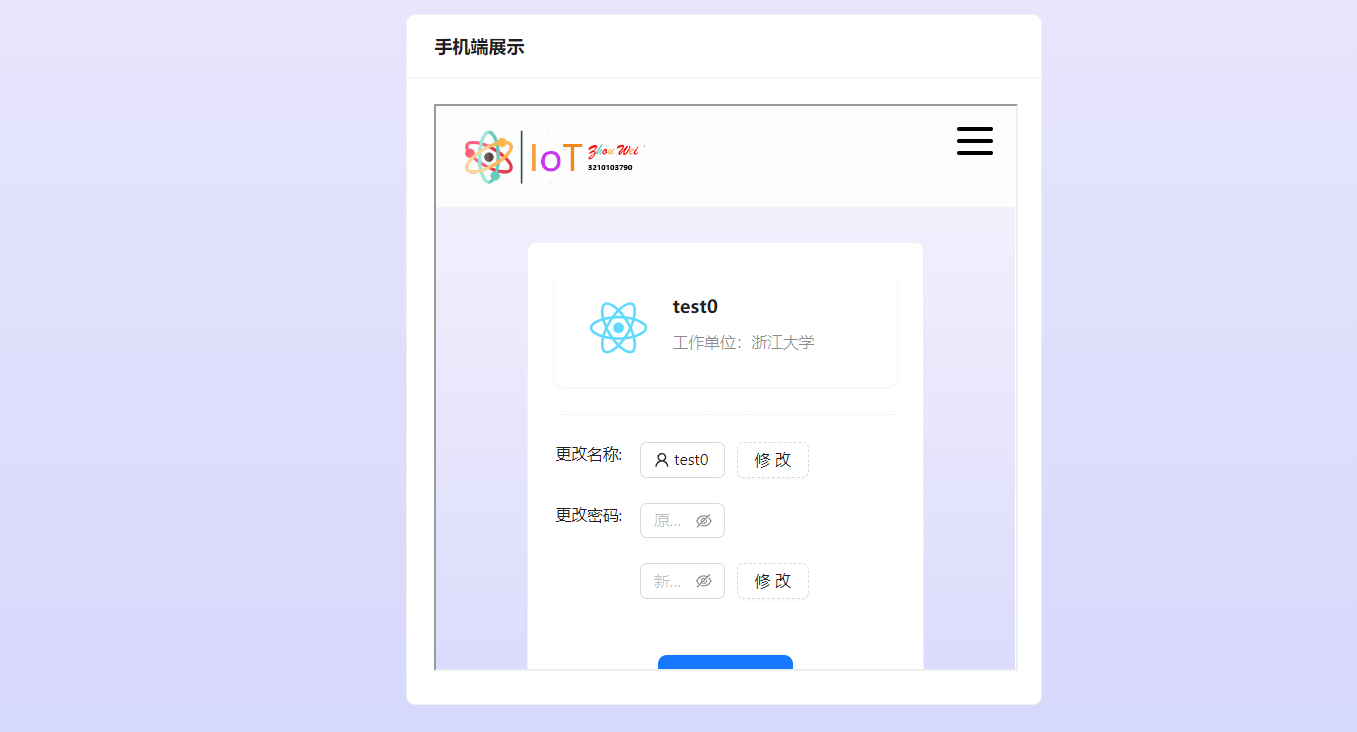
打开导航栏



登录页面



登录成功之后，用户页面显示如下



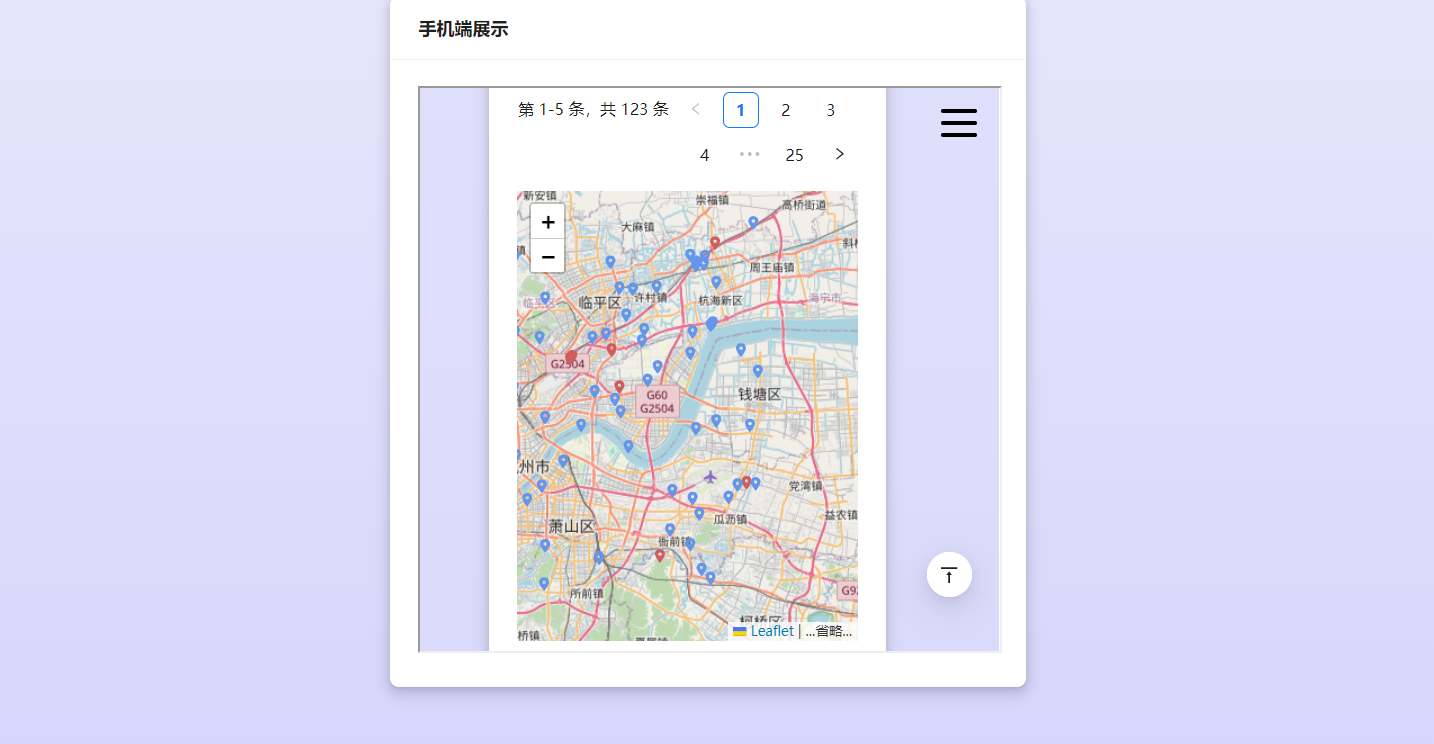
设备页面



并且较长的表格已经被自动折叠了



并且地图显示也正常



拖拽后可以显示全部内容



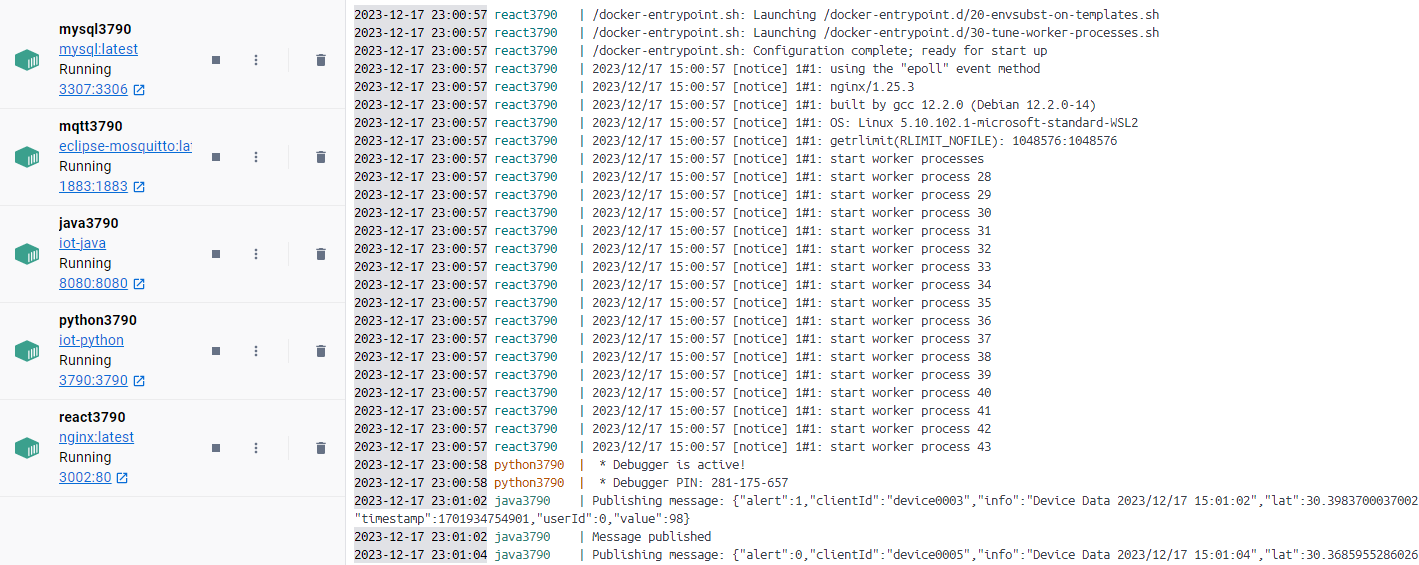
可以发现移动端显示正常，其余不再一一展示

测试正常

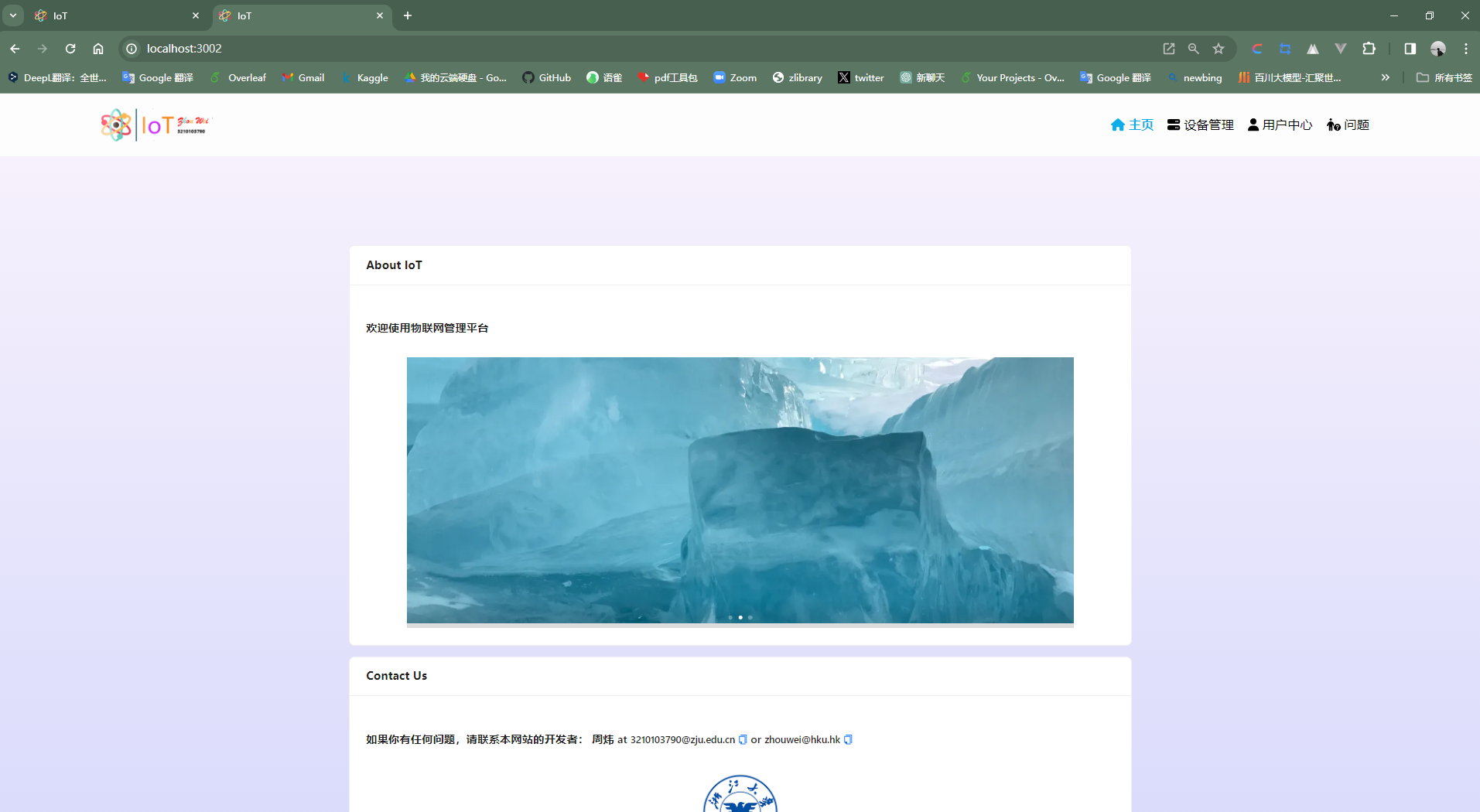
## docker部署测试

使用docker load加载本地image文件后使用docker-compose up -d命令生成容器

可以发现容器运行正常

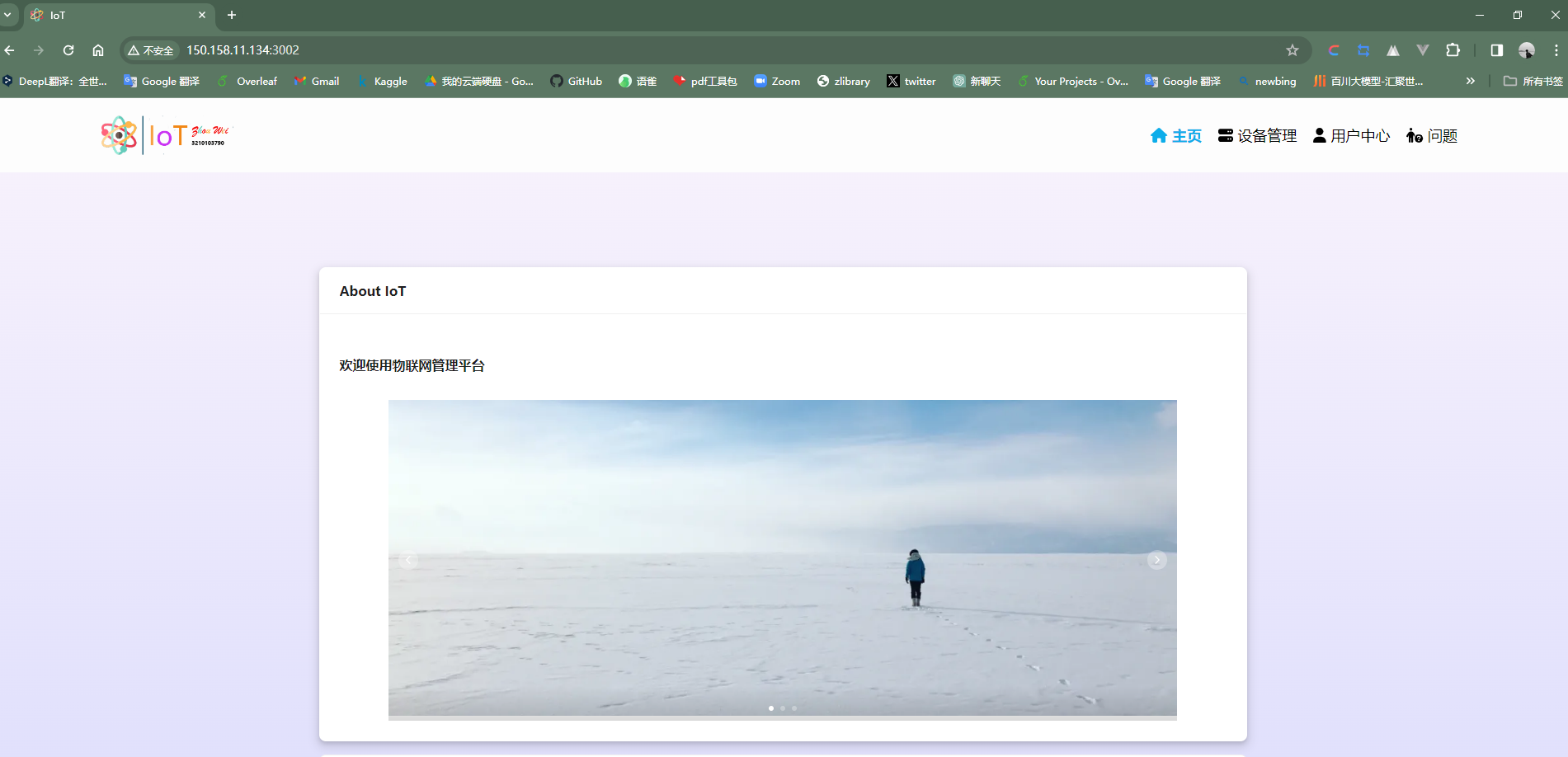


并且可以使用<http://localhost:3002/>正常访问网页

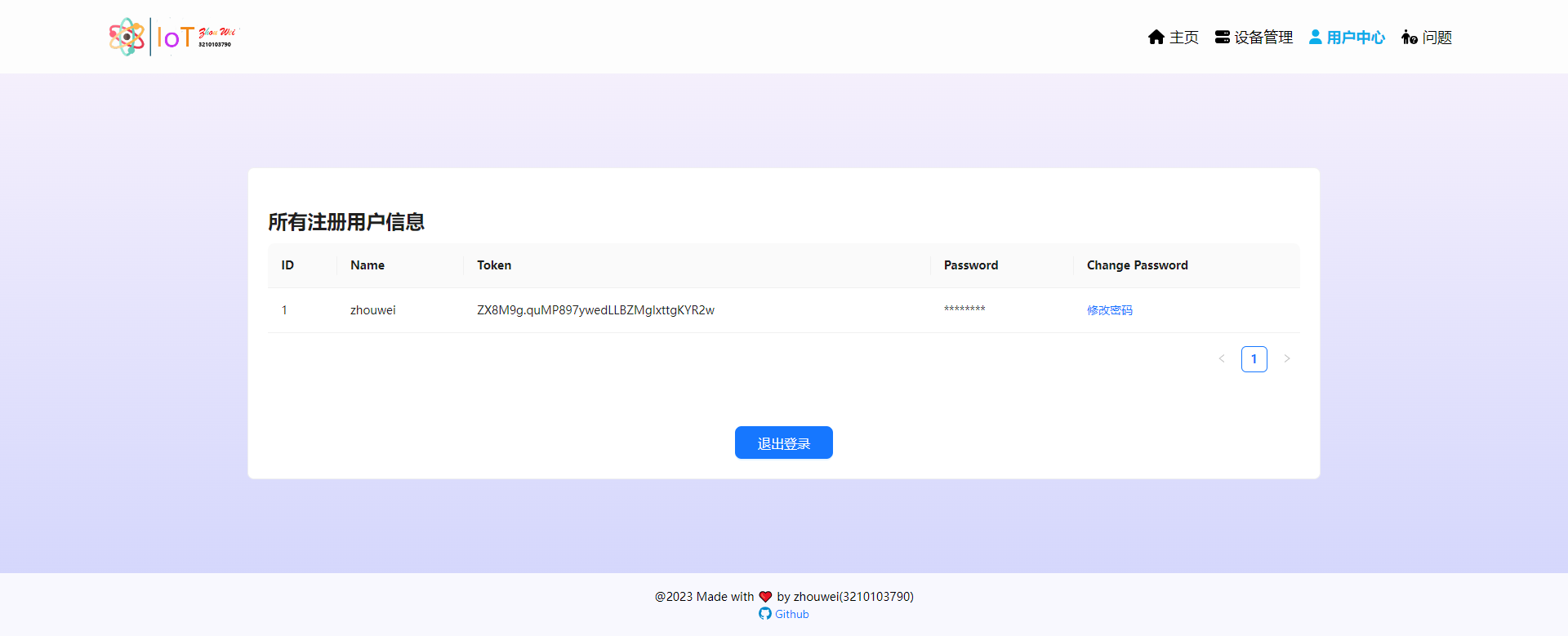


## 服务器部署测试

使用地址<http://150.158.11.134:3002/>可以正确访问网页



所有实现的功能都调用正常，不再一一赘述



并且本地和网端程序能够同时运行互不干扰

