任务一实验报告

一、开发环境与技术说明

Web服务器

并未部署网站于服务器上，且数据库在个人主机上，所以源代码应该无法运行。

- 若想查看网站效果，需要在个人主机上启动前后端服务器，所以不妨联系 `20373585 夏瑞斌` 启动服务器，然后便可通过ip地址 `http://10.128.55.86:5173/` 访问（应该得在局域网内，不懂计网）
- 若想本地运行源代码，需安装下文依赖，将在本地数据库建立相应两张表，并将django连接的数据库改为本地数据库

开发语言

前端：Vue3

后端：Django

数据库

MySQL 8.0

软件依赖

后端

- python 3.10.6
- pip 22.2.1
- django 4.1.1
- djangorestframework-jwt 1.11.0

前端

- node.js v16.17.0
- npm 8.15.0
- vue 3.2.39
- vite 3.1.3
- element-plus(`npm install element-plus --save`)
- icon(`npm install @element-plus/icons-vue`)
- router(`npm install vue-router@4`)
- VueUse(`npm i @vueuse/core @vueuse/components`)
- axios(`npm install axios@0.26.1`)
- universal-cookie(`npm i universal-cookie`)
- integrations(`npm i @vueuse/integrations`)
- vuex(`npm install vuex@next --save`)、
- windicss(`npm i -D vite-plugin-windicss windicss`)

二、数据库表定义

用户填写注册信息后，将数据填入“待审批用户表”中；以管理员身份（Permission 为 0 或 1）登录系统后进行审批并赋予权限，审批通过则加入“用户信息表”中。

用户信息表（user_account）

数据项名字	数据类型	约束	备注
CodeName	char(30)	primary key	代号；相当于用户名
Password	char(20)	not null	
Permission	tinyint unsigned	not null	值越小，权限越高
Class	char(20)		职业分类
Region	char(30)		地区
Race	char(20)		种族
Mail	char(20)		邮件地址

待审批用户表

数据项名字	数据类型	约束	备注
CodeName	char(30)	primary key	代号；相当于用户名
Password	char(20)	not null	
Permission	tinyint unsigned		待管理员分配
Class	char(20)		职业分类
Region	char(30)		地区
Race	char(20)		种族
Description	text		备注

三、数据库使用方法

1.连接数据库

`/backend/__init__.py`：添加`pymysql`模块

```
import pymysql

pymysql.install_as_MySQLdb()
```

`/db_project/settings`：调整`DATABASES`字典`default`内容，连接到本地MySQL

```
from . import privacy as pv
# 本地写了个privacy.py存出相关数据库用户信息，不传到库中
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'arknights',
        'HOST': pv.db_host,
        'PORT': 3306,
        'USER': pv.db_user,
        'PASSWORD': pv.db_password[pv.db_user],
    }
}
```

具体调用（如`/backend/views.py`）：`django.db`模块的`connection`对象

```
from django.db import connection

# ...
with connection.cursor() as cursor:
    sql = 'xxx"
    try:
        cursor.execute(sql)
        one = cursor.fetchone()
        pass
    except:
        pass
return result
```

2.增

- 用户填完注册信息后，将相关信息写入“待审批用户表”
- 将“待审批用户表”中通过审核并赋予权限的信息写入“用户信息表”

```
def add_into_queue(CodeName, Class, Region, Race, Description, Password):
    sql_check_user = "select * from user_account where CodeName =
'{}'".format(CodeName)
    sql_check_application = "select * from account_approve_queue where CodeName
= '{}'".format(CodeName)
    sql_insert = "insert into account_approve_queue values('{}','{}',
{},'{}','{}','{}','{}')" \
                 .format(CodeName, Password, 2, Class, Region, Race, Description)
    try:
        with connection.cursor() as cursor:
            cursor.execute(sql_check_user)
            if cursor.fetchone() is not None:
                return fail('该用户已存在')

            cursor.execute(sql_check_application)
            if cursor.fetchone() is not None:
                return fail('该用户已在申请队列中')

            cursor.execute(sql_insert)
        return success('注册成功，请等待审核')
    except:
        return fail('注册失败QWQ')
```

3.删

- 审核不通过的注册申请信息从“待审批用户表”中

```
def reject(name):
    sql = "delete from account_approve_queue where CodeName = '{}'
".format(name)
    try:
        with connection.cursor() as cursor:
            cursor.execute(sql)
        return success('拒绝该用户的注册申请')
    except:
        return JsonResponse({'request': FAIL_DATA})
```

4.改

- 管理员审核“待审批用户表”时，可以对注册信息进行修改

```
def modify_application(CodeName, Permission, Class, Region, Race, Description):
    sql = "update account_approve_queue Set Permission = {},Class = '{}',Region
= '{}',Race = '{}',Description = " \
              "'{}' where CodeName = '{}'".format(Permission, Class, Region,
Race, Description, CodeName)
    try:
        with connection.cursor() as cursor:
            cursor.execute(sql)
        return success('申请用户信息修改成功~')
    except:
        return fail('修改失败了')
```

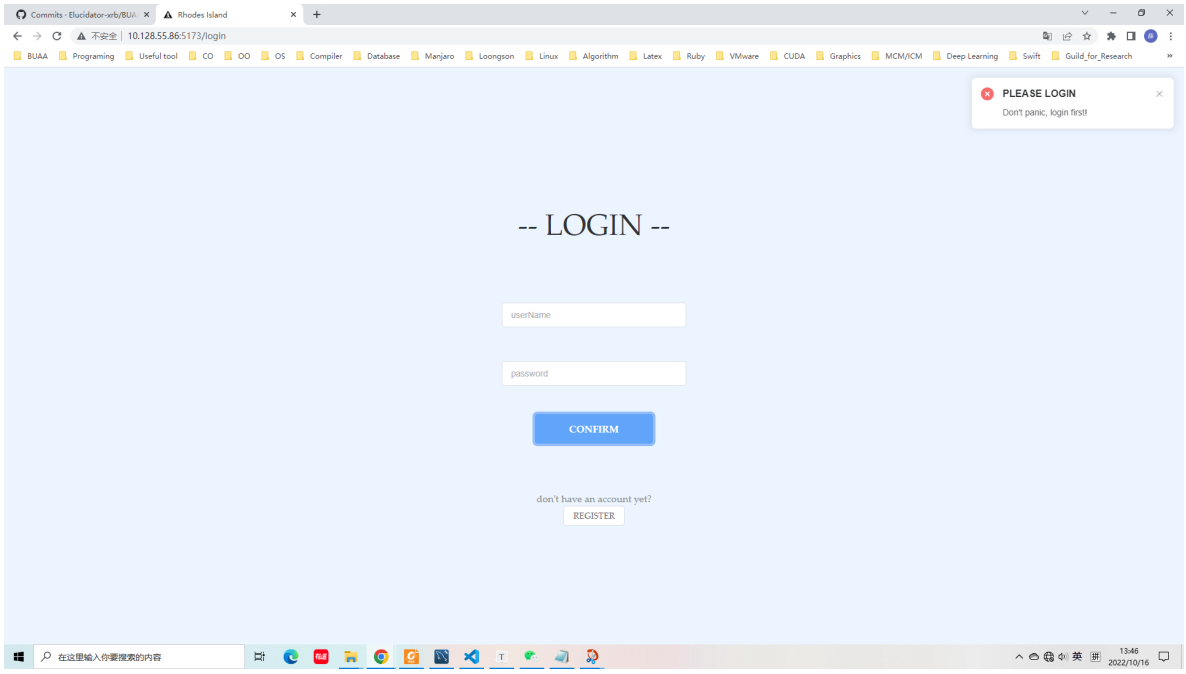
5.查

- 每次前后端的信息交互都会涉及查表，它无处不在

四、功能截图

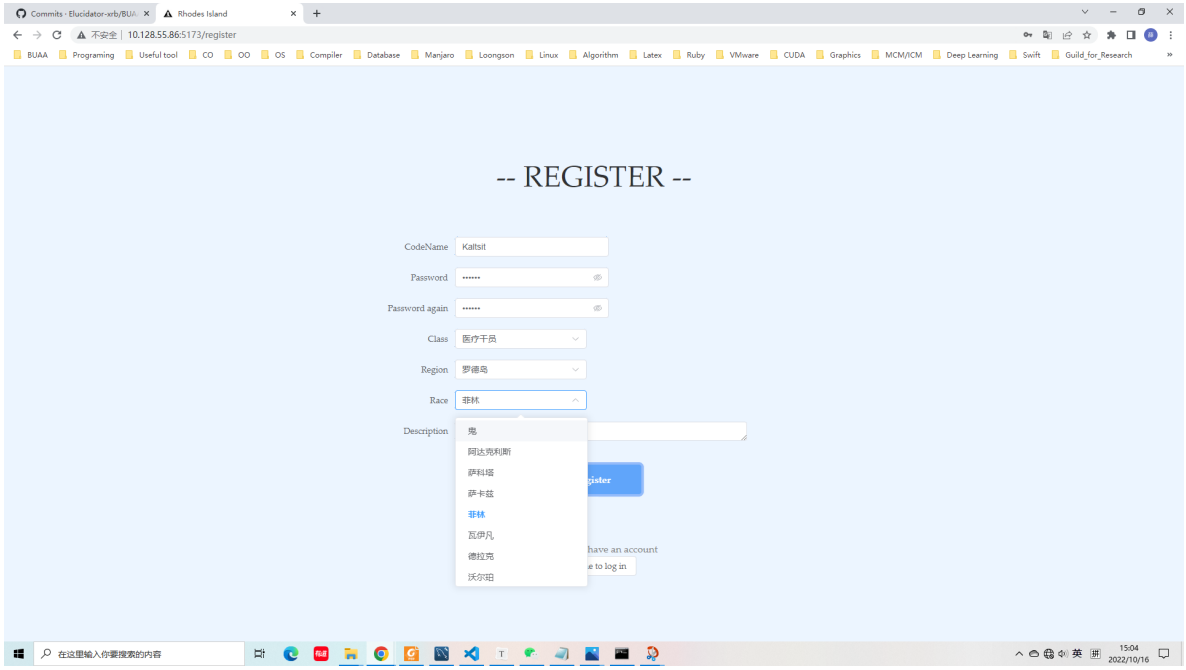
1.登录界面

打开网址后自动跳转登录界面



2.注册界面

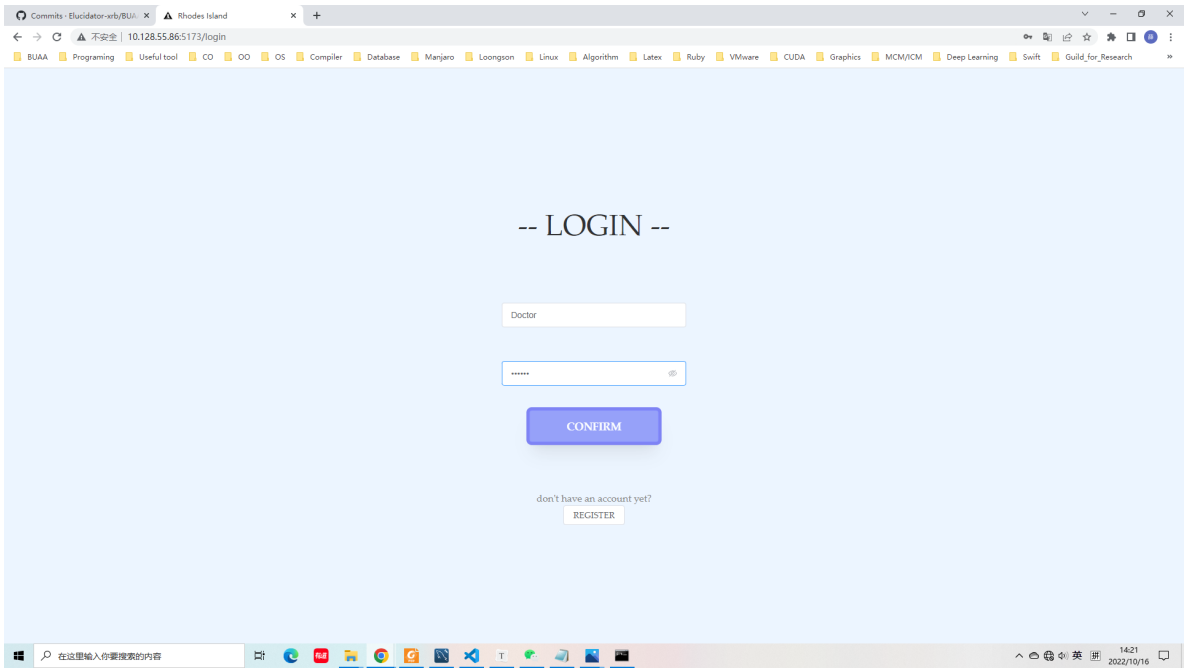
点`REGISTER`按钮跳转至注册界面，输入相关信息进行注册



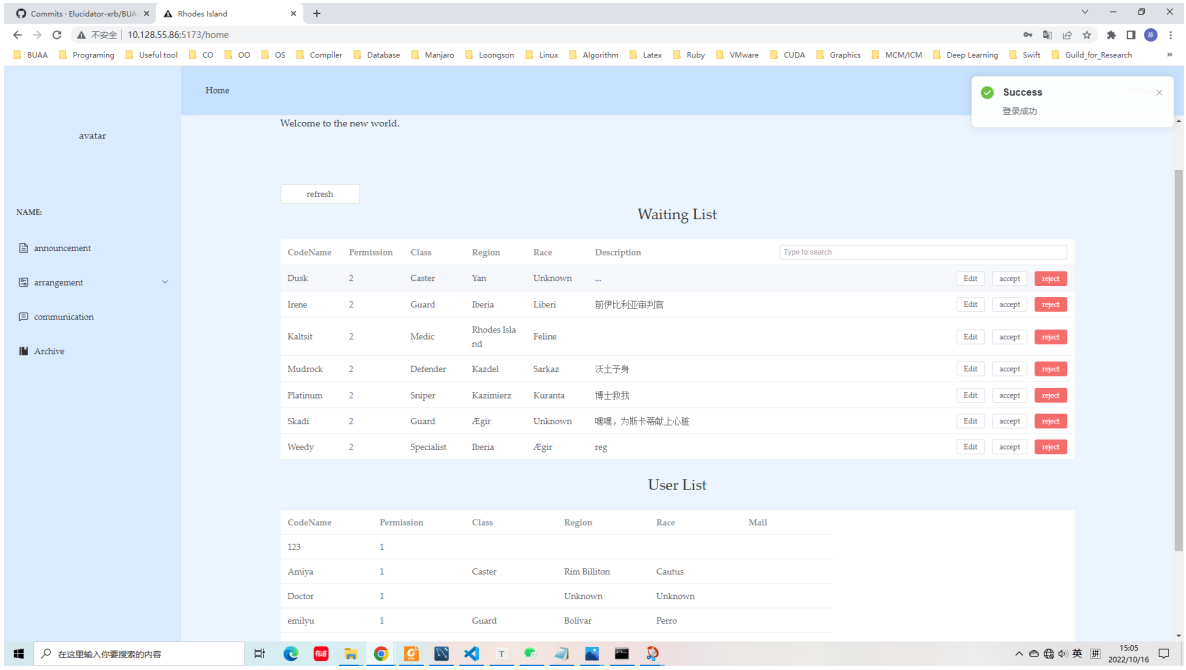
点击register按钮完成注册，若信息不符合要求，则会提示修改；若符合要求，将会将相关信息填入“待审批用户队列”，并跳转至登录界面，弹出通知提示“等待审核”

3.管理界面

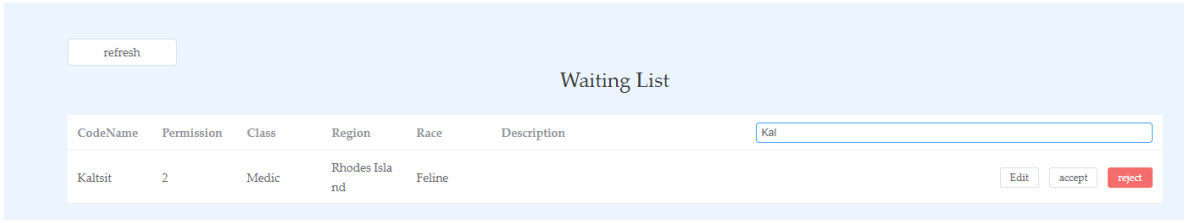
使用管理员进行登录



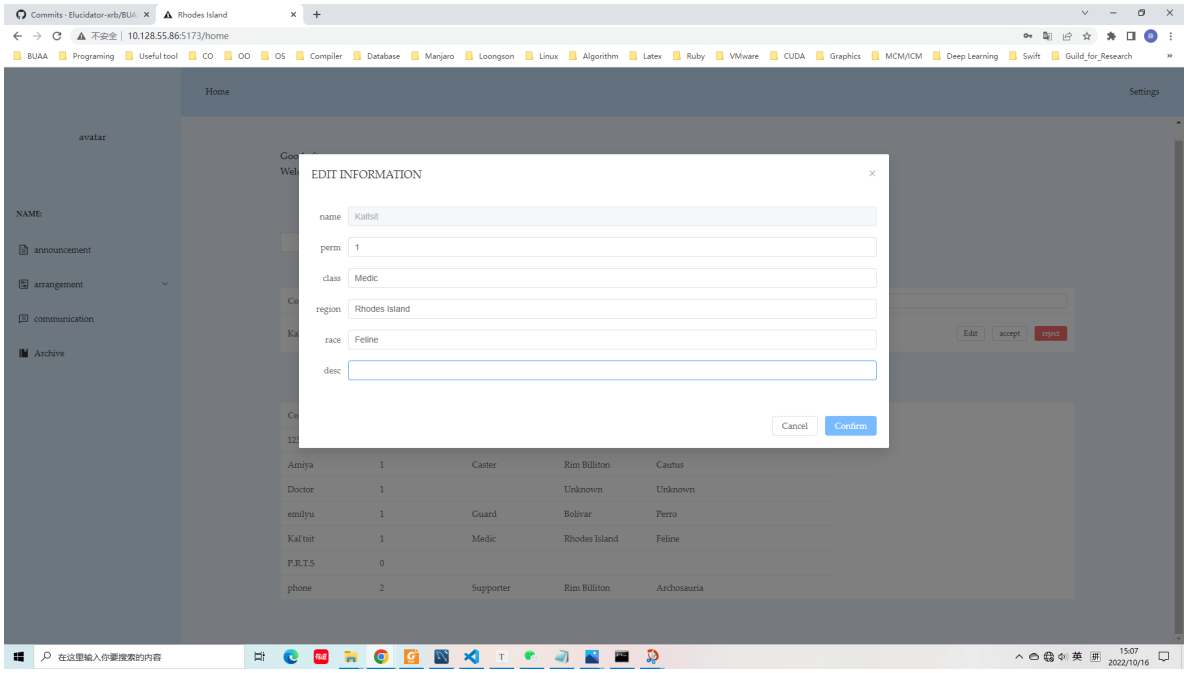
进入管理员界面主页



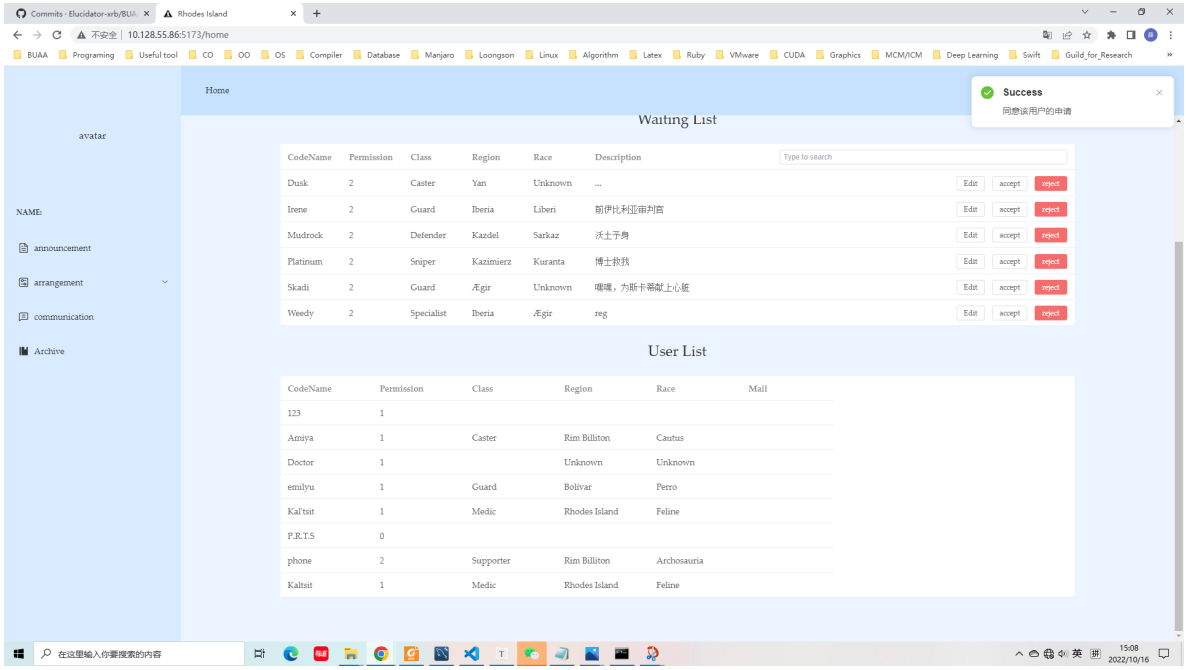
欢迎信息根据当前时间显示早/中/晚上好；可在`Waiting List`中搜索筛选显示列；点击`Edit`可对"待审批用户队列"表进行修改；`accept`注册同通过，将用户信息转填入正式的"用户信息表"中；`reject`拒绝用户注册，将信息直接从"待审批用户队列中"移除。`Waiting List`和`User List`会根据数据表的变动即时响应。



筛选显示

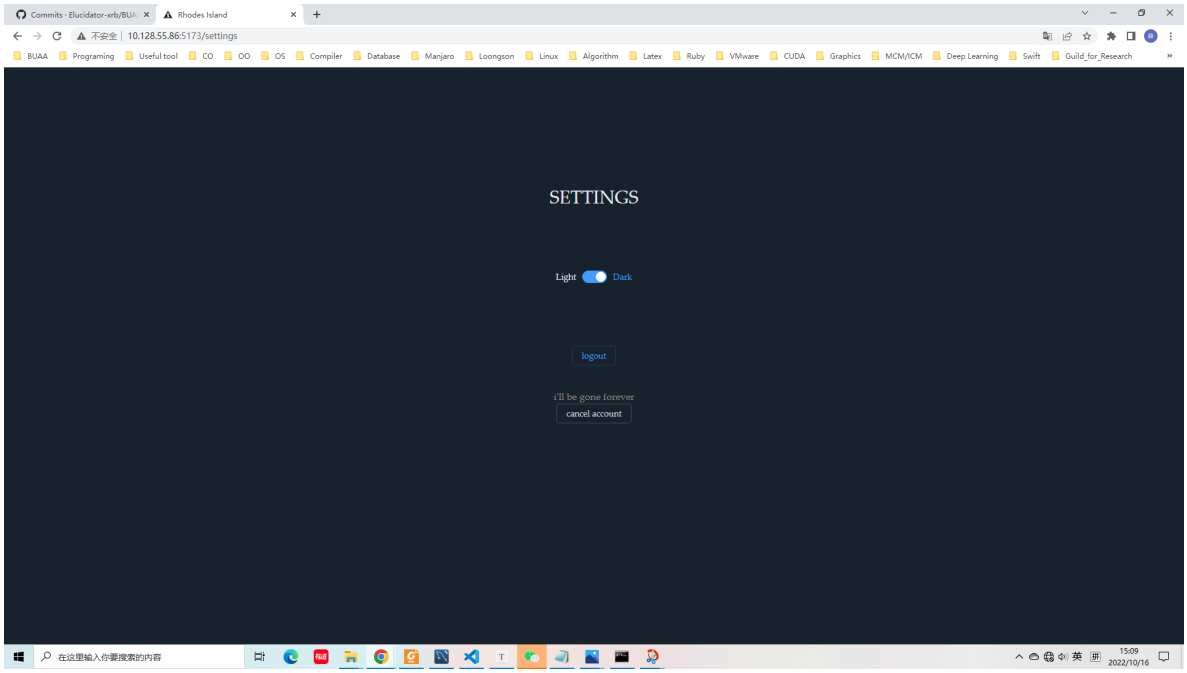


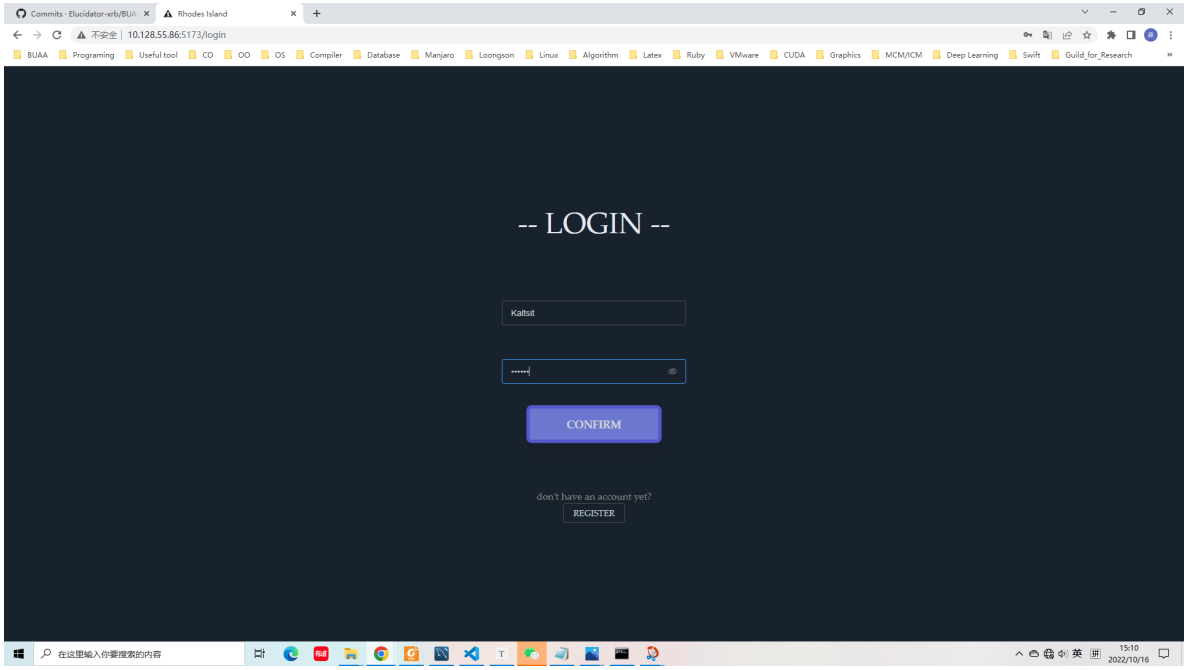
修改信息



通过审批（注意：新用户在最后一行显示）

修改界面显示色调并退出登录状态。由于新用户权限设为1，可以使用该用户重新登录并进入管理员界面




Commits · Elucidator-wrb/BUA
Rhodes Island
不安全 | 10.128.55.86:5173/login
BUAA Programing Useful tool CO OO OS Compiler Database Manjaro Loongson Linux Algorithm Latex Ruby VMware CUDA Graphics MCM/ICM Deep Learning Swift Guild_for_Research
-- LOGIN --
Kaltsit
CONFIRM
don't have an account yet?
REGISTER
在这里输入你要搜索的内容
15:10
2022/10/16