天津工业大学计算机科学与技术学院

实训总结报告



　项目名称: 高数 ----- 您的高级数据信息服务平台

　合作培养企业：天津市融创软通科技股份有限公司

　专业方向：大数据方向

　项目负责人：何晶

　专业班级: 软件Z1802

姓名：李嘉威

　实训起止日期：2020.6.15-2020.7.15

**1 引言** （一级标题，小三黑体，居中）

1.1项目概述（二级标题，四号黑体，左对齐）

说明项目目的、背景、意义及参考资料（包括企业提供的素材、DEMO）等；

在当下这个信息爆炸的时代，互联网作为人们工作、生活、学习都无法脱离的平台，每天都会产生海量数据，音乐、电影、书籍、评论种种信息虽然琐碎，担当大量获取分析后将会得出许多有用的结论。

本项目采用了前后端分离的开发方式，前段采用Vue框架进行开发，后端使用SpringBoot框架开发，通过scrapy爬虫机制针对豆瓣电影、音乐、图书和当当图书、B站动漫评论进行数据收集并分析，结果保存与远程mariaDB数据库中。提供简洁明了的数据查询服务，便于查询所需条件的数据，也提供关于数据内容的分析服务，并将其以图表的方式简洁直观的展示出来，方便用户观看。用户可根据分析所获结果，快捷的寻找满意的电影、音乐、书籍等资源，此外团队远程开发利用代码托管平台gitee和代码版本控制器git。

为便于完成项目，本次实训中企业为我们提供了scrapy爬虫基础教学、SSM服务器短开发框架，LayUI前端技术以及所需要的模版、软件

1.2 需求分析

为什么做这个题目，有什么实际价值

随着社会发展，人们的生活质量越来越高，伴随着各种娱乐产业的新增，每天都有海量数据生成。单单只看其中一些数据，是毫无意义的，担当将当量数据集中金雄分析，就能从中取得许多信息，这也是大数据的意义所在。

该项目将爬取豆瓣电影、音乐、图书和当当图书、B站动漫评论的数据，然后所获得的信息进行分析，得到诸如最畅销的书籍、最受欢迎的导演等等数据，之后以表格的形式将分析直观的展示出来，以此为客户挑选电影、音乐、动漫、书籍时提供便捷。

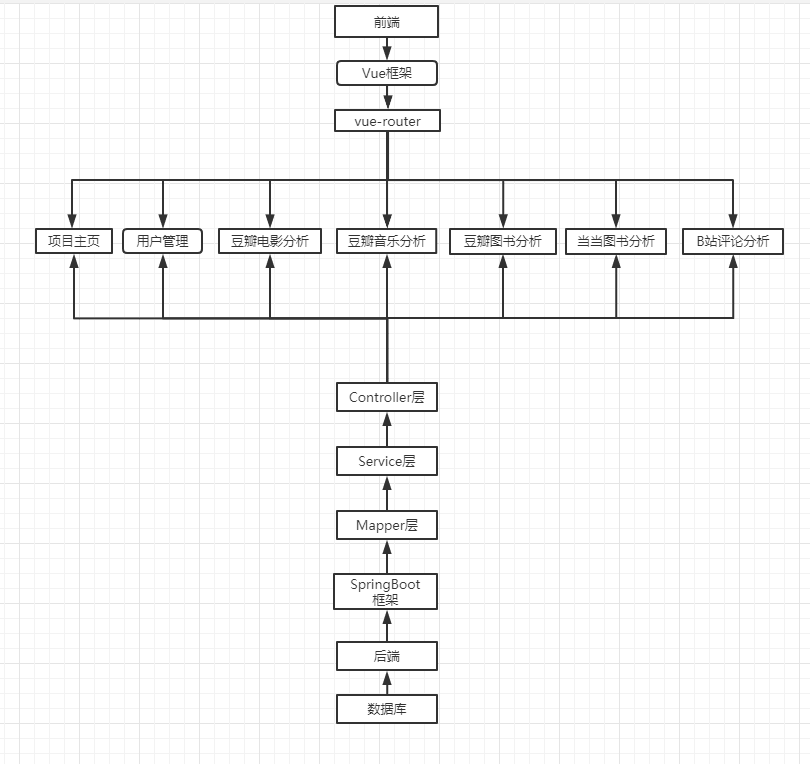
1.3 运行环境

软、硬件环境

（正文，小四号宋体，1.25倍行距）

**2 项目设计**（小三黑体，居中）（以个人主要负责内容为主）

2.1 设计思路



设计思路

系统大致可以分为三层：Vue前端、SSM后端、以及MySql数据库三层。Mysql数据库存储通过scrapy爬取到的数据，用以进行后端分析和前端展示；SSM后端主要有三层：Controller层（控制层，负责具体模块的业务流程控制，需要调用service逻辑设计层的接口来控制业务流程）、Mapper层（对数据库进行数据持久化操作，他的方法语句是直接针对数据库操作的，主要实现一些增删改查操作）、Service层（业务service层，给controller层的类提供接口进行调用。），由此向前端提供接口，完成交互；Vue前端通过接口获得相应数据，完成展示

2.2 模块功能介绍

**模块一：系统基础管理模块**

1) 平台主界面设计及开发。

2) 管理首页设计与开发，图形化当前系统情况，包括已注册用户数量、已贮存影片（书籍、评论、音乐）数量、已分析的评论数、各分类的检索前十名汇总等信息；

**模块二：当当图书模块**

1) 通过不同关键字迅速返回符合条件的数据目录

2) 将会所获得数据进行分析，结果以图表方式展出

**模块三：B站评论模块**

1) 通过不同关键字迅速返回符合条件的数据目录

2) 将会所获得数据进行分析，结果以图表方式展出

**模块四：豆瓣音乐模块**

1) 通过不同关键字迅速返回符合条件的数据目录

2) 将会所获得数据进行分析，结果以图表方式展出

**模块五：豆瓣电影模块**

1) 通过不同关键字迅速返回符合条件的数据目录

2) 将会所获得数据进行分析，结果以图表方式展出

2.3 模块结构图

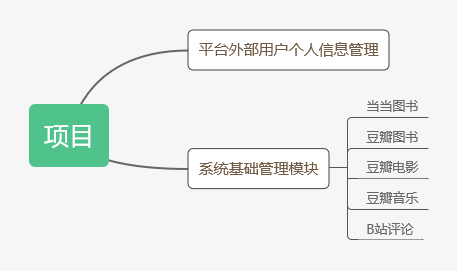
****

Figure 2模块结构图

项目大致可分为七个模块，包含两个主模块，五个小模块。主模块包括平台外部用户个人信息管理，主要负责用户注册，修改信息；系统基础管理模块负责平台页面框架的搭建，其中包含着当当图书、豆瓣图书、豆瓣电影、豆瓣音乐、B站评论五个模块的内容

2.4 程序流程图

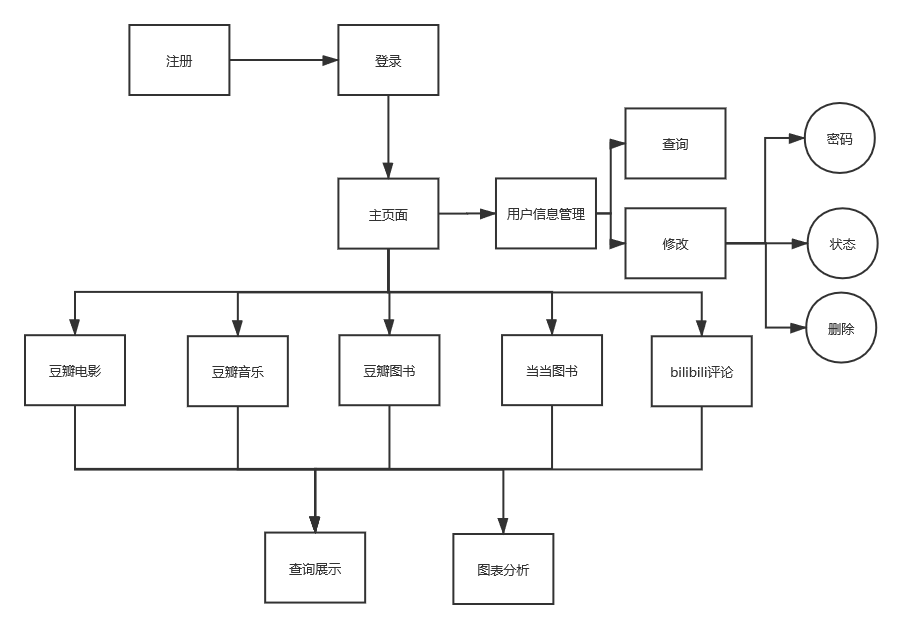


Figure 3程序流程图

2.5 功能设计分工

需详细标注每部分的内容，具体任务及任务安排等。

数据爬取：

豆瓣电影：段成源

豆瓣音乐：刘博

豆瓣图书：廖嘉伟

当当图书：薛彦涵

B站评论：李嘉威

前端框架：

主界面搭建：李嘉威

登录：薛彦涵

注册：李嘉威

用户管理：李嘉威

豆瓣电影：段成源

豆瓣音乐：刘博

豆瓣图书：廖嘉伟

当当图书：薛彦涵

后端

登录：李嘉威

注册： 刘博，薛彦涵

豆瓣电影：段成源

豆瓣音乐：刘博

豆瓣图书：廖嘉伟

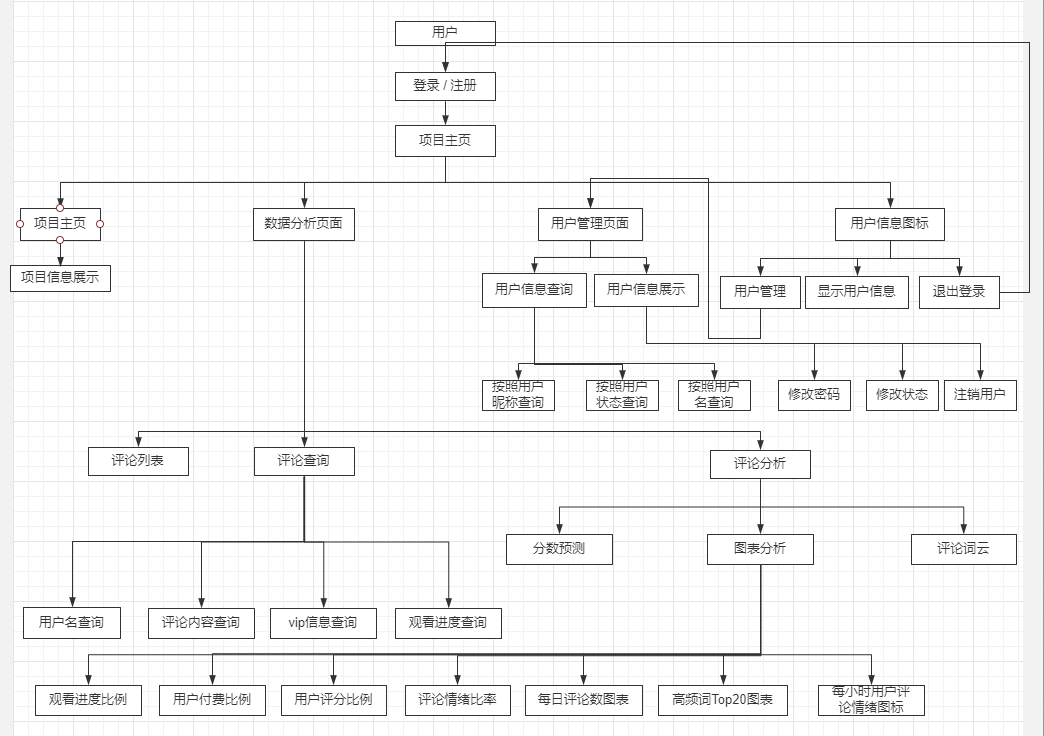
当当图书：薛彦涵

**3 详细设计**（小三黑体，居中）

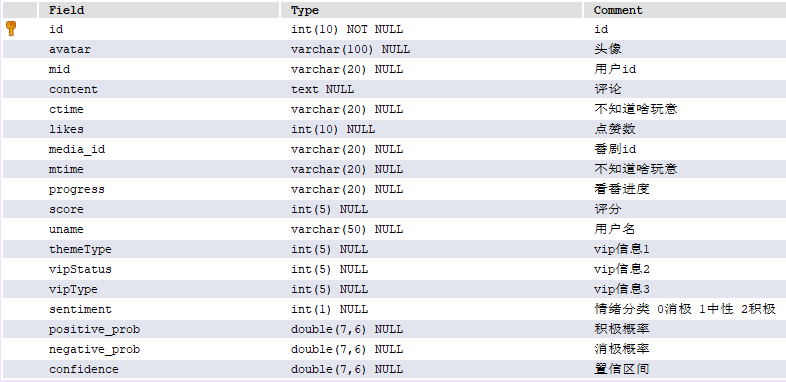
详细写明项目的设计（系统设计、数据库设计）等，并写明对所涉技术、核心功能等描述；

采用了前后端分离的开发方式，前段采用Vue框架进行开发，前端使用UI组件库为Ant-Design-Vue，前端使用Axios向后端发送请求，后端使用SpringBoot框架开发，通过scrapy爬虫机制针对豆瓣电影、音乐、图书和当当图书、B站动漫评论进行数据收集并分析，结果保存与远程mariaDB数据库中。提供简洁明了的数据查询服务，便于查询所需条件的数据，也提供关于数据内容的分析服务，并将其以图表的方式简洁直观的展示出来，方便用户观看。用户可根据分析所获结果，快捷的寻找满意的电影、音乐、书籍等资源，此外团队远程开发利用代码托管平台gitee和代码版本控制器git。

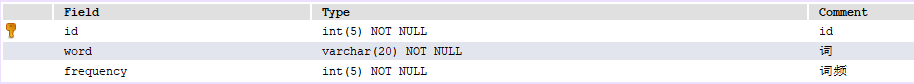
系统设计：

数据库：

1. 评论信息表



1. 词频统计表



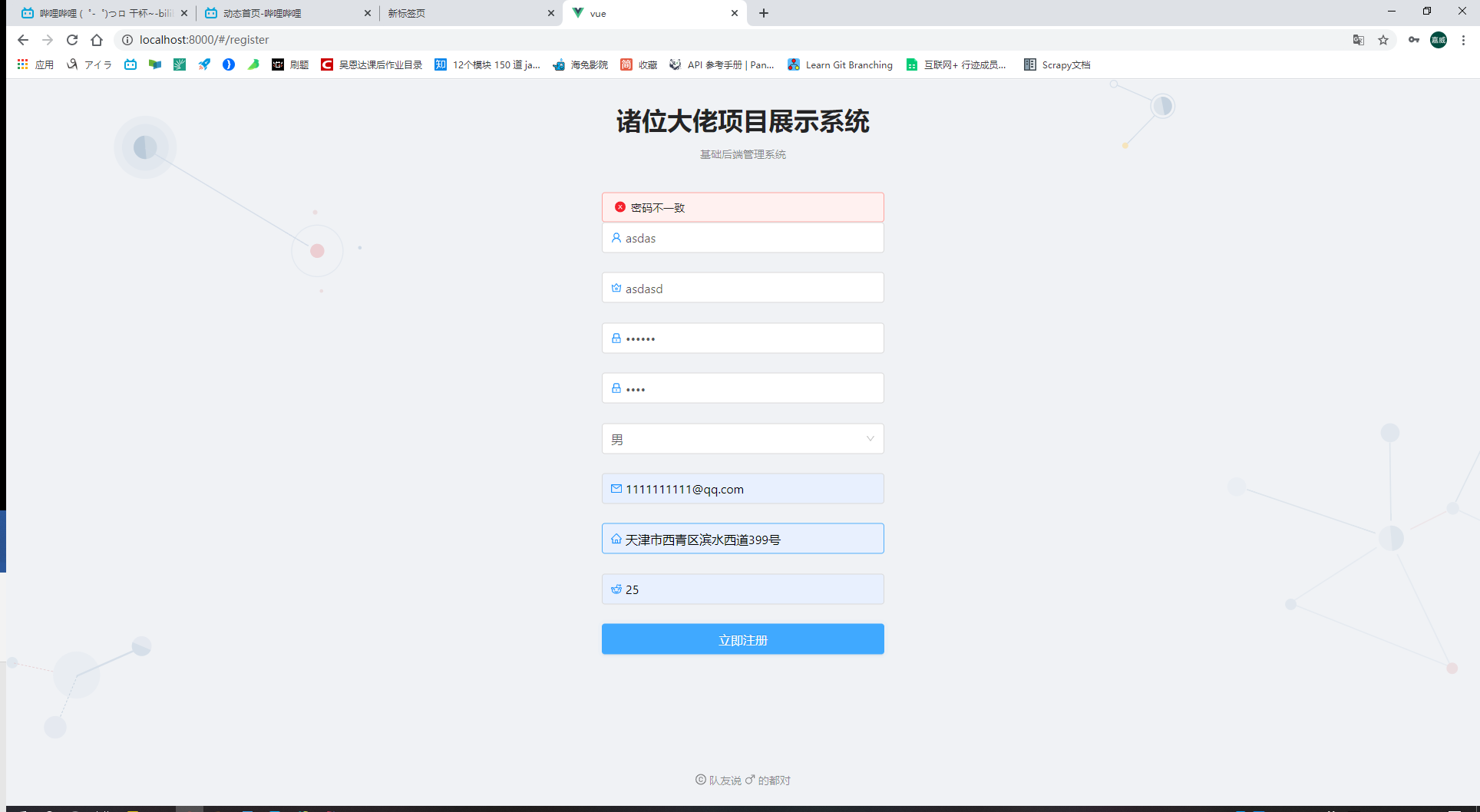
**4 实现方案**（小三黑体，居中）

详细写明项目功能实现描述、核心代码及实现结果等；

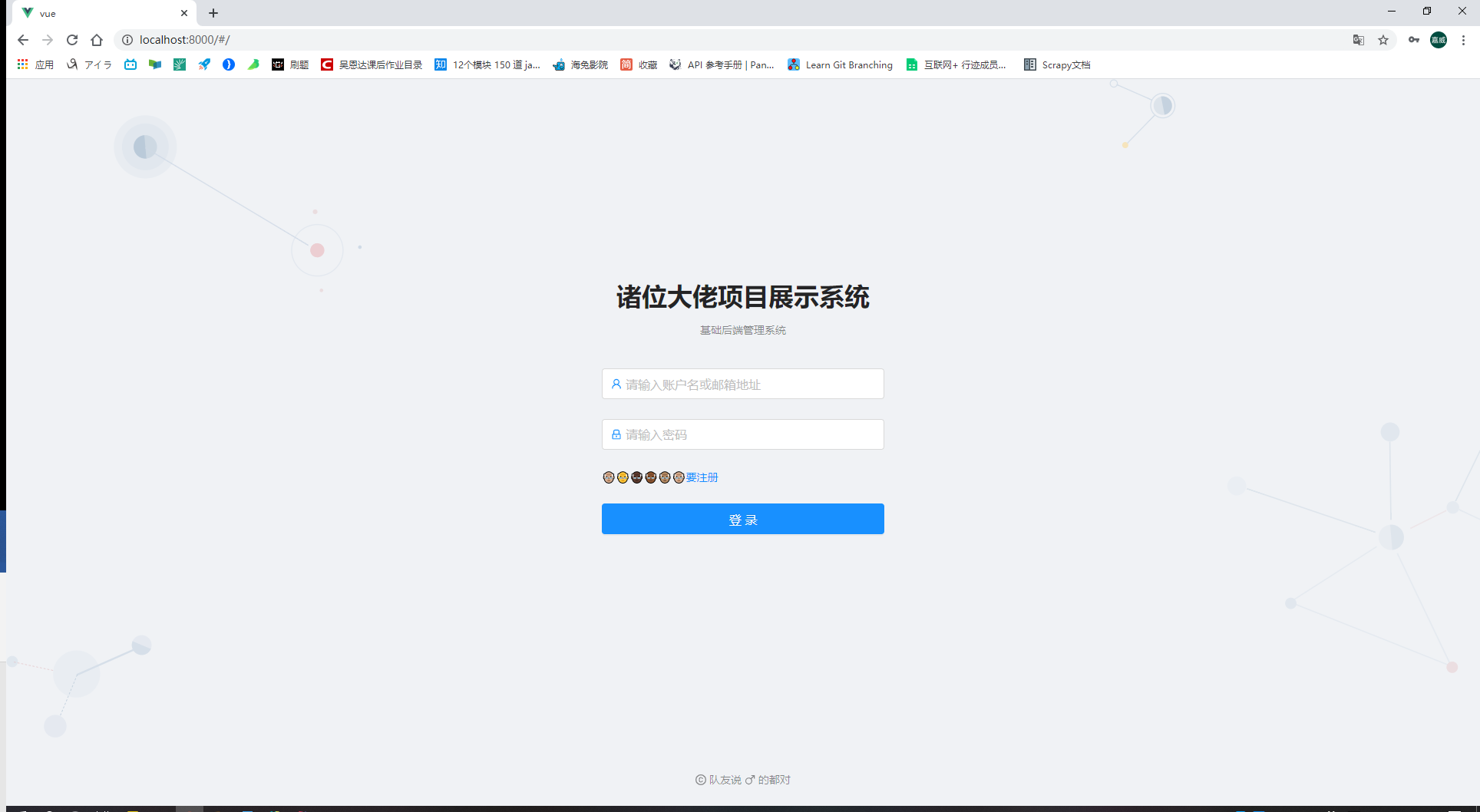
并对调试好的程序，从执行程序弹出界面开始，每一步操作截一个图，并附加说明（图片格式为JPG格式）。

1. 注册功能：

用户输入注册信息，其中包含了密码和确认密码，如果前端发现两次密码不一致，就会直接进行拦截，并且进行报错提示，如果一致，就会发送到后端进行数据库校验，如果数据库中不存在次uid，那么就成功注册，否则就不会成功注册。



注册成功之后会跳转到登陆页面等待用户登录

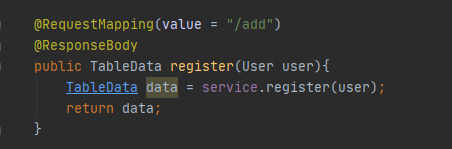


前端核心代码：

<script>  
 import axios from 'axios'  
 // import router from "../router";  
 import qs from 'qs';  
  
 export default {  
 name: 'Register',  
 data() {  
 return {  
 form: this.$form.createForm(this),  
 isLoginError: false,  
 // loginAccount: null,  
 userName:null,  
 nickName:null,  
 userPassword:null,  
 rePassword:null,  
 userGender:null,  
 userEmail:null,  
 userAddress:null,  
 userAge:null,  
 // userPassword: null,  
 }  
 },  
 methods: {  
 register: function () {  
 if (this.rePassword !== this.userPassword){  
 this.isLoginError = true  
 }  
 let data = qs.stringify({  
 uid: this.userName,  
 pwd: this.userPassword,  
 nickName: this.nickName,  
 email: this.userEmail,  
 address: this.userAddress,  
 sex: this.userGender,  
 age: this.userAge  
  
 });  
 // this.$router.push("/mainpage") 不知道哪位大神在这边加了一个跳转 BY 薛某  
 ***axios***.post('http://localhost:8080/logic/user/add', data,  
 { headers: {'Content-Type': 'application/x-www-form-urlencoded'}  
 }).then(response => {  
 data = response.data;  
 if (1000 === data.code){  
 this.$router.push("/")  
 }  
 else {  
 alert("用户已存在")  
 }  
 }).catch( error => {  
 ***console***.log(error)  
 })  
 // .then(function (dat) {  
 // if (dat.data === "0")  
 // alert("用户不存在或密码错误")  
 // else if (dat.data === "2")  
 // router.push("/mainpage")  
 // }).catch(function () {  
 // console.log("传输失败")  
 // })  
 },  
 handleSelectChange: function (value) {  
 this.userGender = value  
 }  
 }  
 }  
</script>

后端代码：

Controller



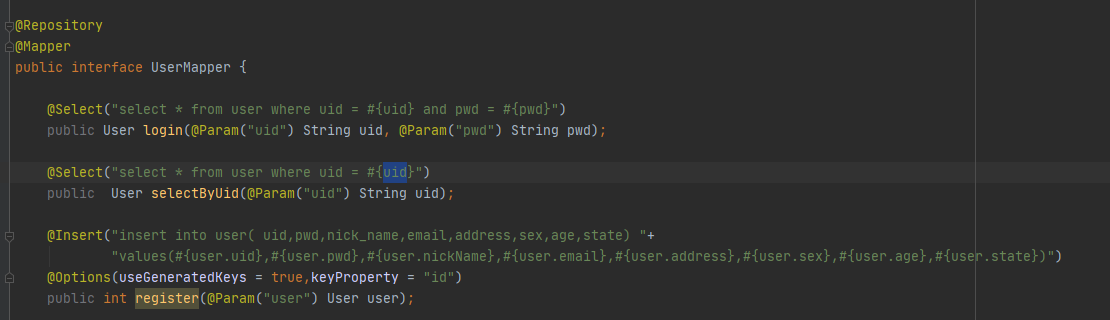
Service接口



Service实现

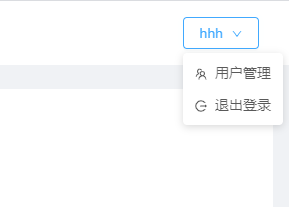


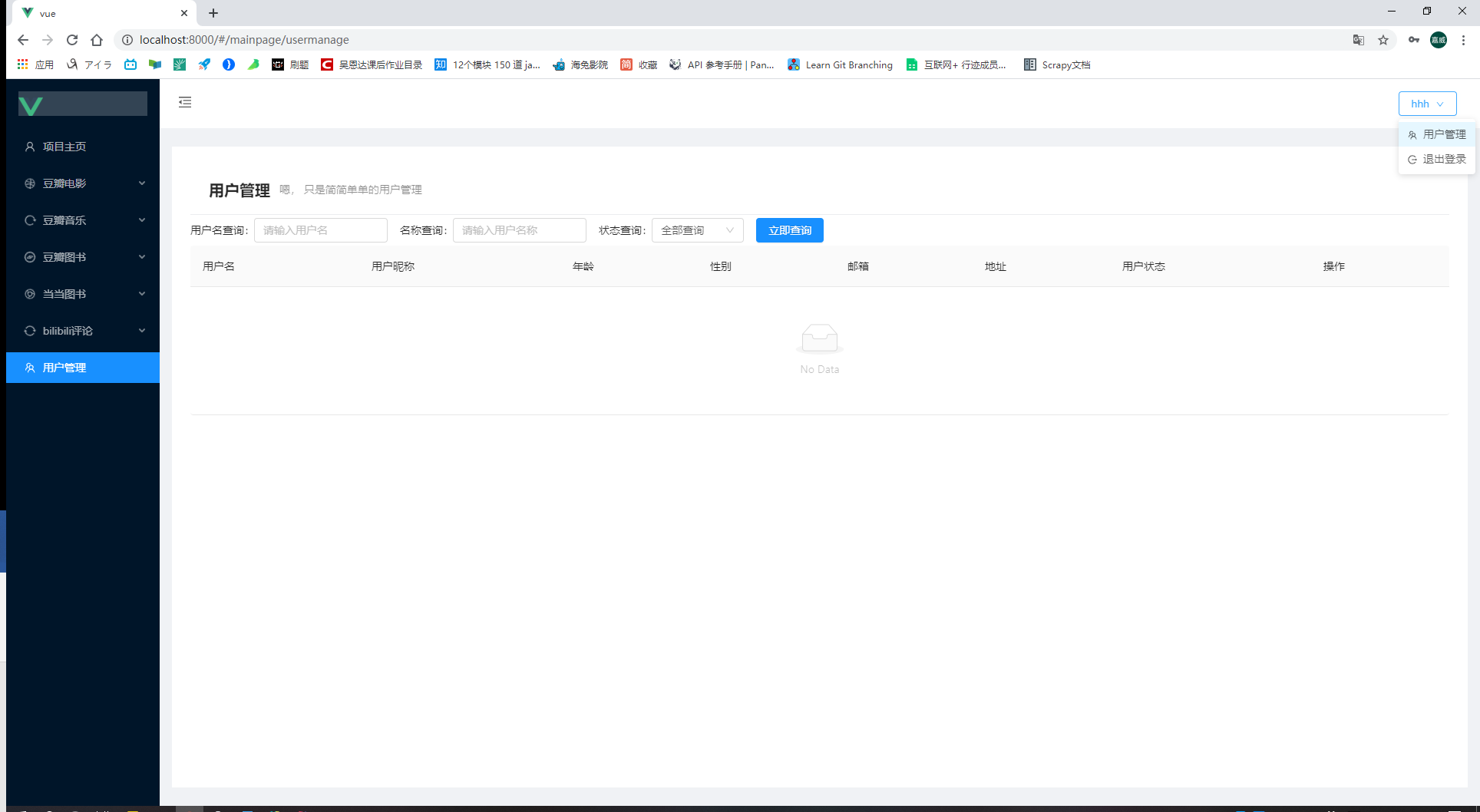
Mapper

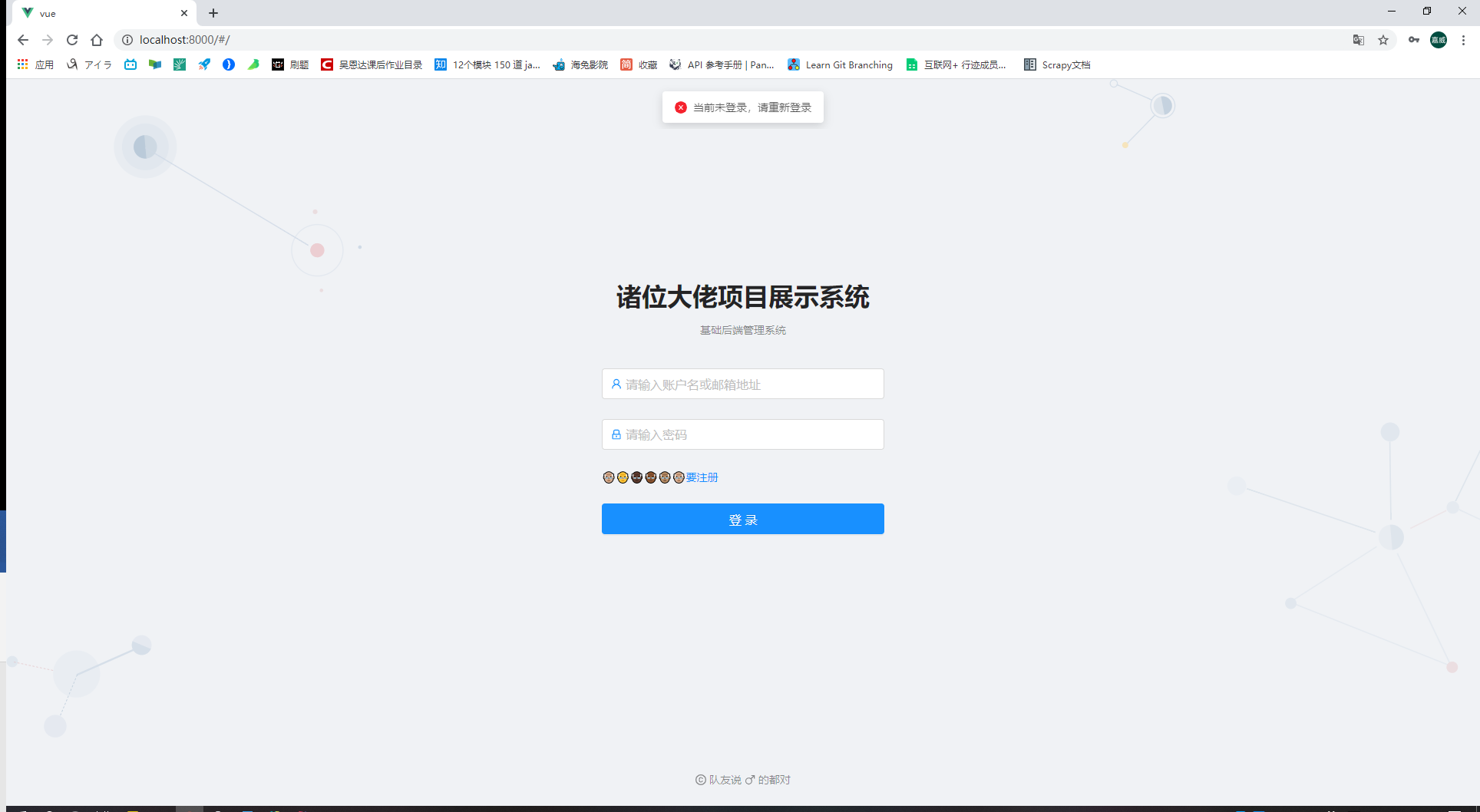


1. 用户信息

用户登陆后，会在右上角显示用户名，同时提供用户管理入口，和退出登录选项。如果企图从浏览器url通过直接输入对应页面路由进入页面，也会被强制退回到登陆界面。





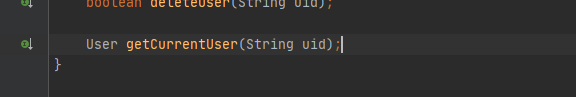


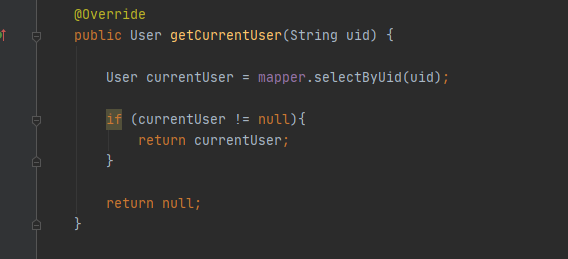
前端核心代码：

<template>  
 <a-dropdown>  
 <a-menu slot="overlay">  
 <a-menu-item key="1" v-on:click="directTo('/mainpage/usermanage')"> <a-icon type="team" />用户管理</a-menu-item>  
 <a-menu-item key="2" v-on:click="directTo('/')"> <a-icon type="logout" />退出登录</a-menu-item>  
 </a-menu>  
 <a-button style="margin-left: 8px">{{ username }}<a-icon type="down" /> </a-button>  
 </a-dropdown>  
</template>  
  
<script>  
 import ***axios*** from "axios"  
 import qs from "qs"  
 import ***router*** from "../router"  
 export default {  
 name: "UserInfo",  
 props: ['uid'],  
 data(){  
 return{  
 username: 'userName'  
 }  
 },  
 mounted() {  
 ***console***.log(this.uid)  
 ***axios***.post('http://localhost:8080/logic/user/currentUser', qs.stringify({  
 uid: this.uid  
 }), {headers:{'Content-Type':'application/x-www-form-urlencoded'}})  
 .then(response => {  
 ***console***.log(response)  
 const data = response.data;  
 if (data.code === 1000){  
 this.username = data.data.nickName  
 }else {  
 this.$message.error("当前未登录，请重新登录")  
 this.$router.push("/")  
 }  
 }).catch(error => {  
 ***console***.log(error)  
 })  
 },  
 methods:{  
 directTo(target){  
 ***router***.push(target)  
 }  
 }  
 }  
</script>

后端核心代码：

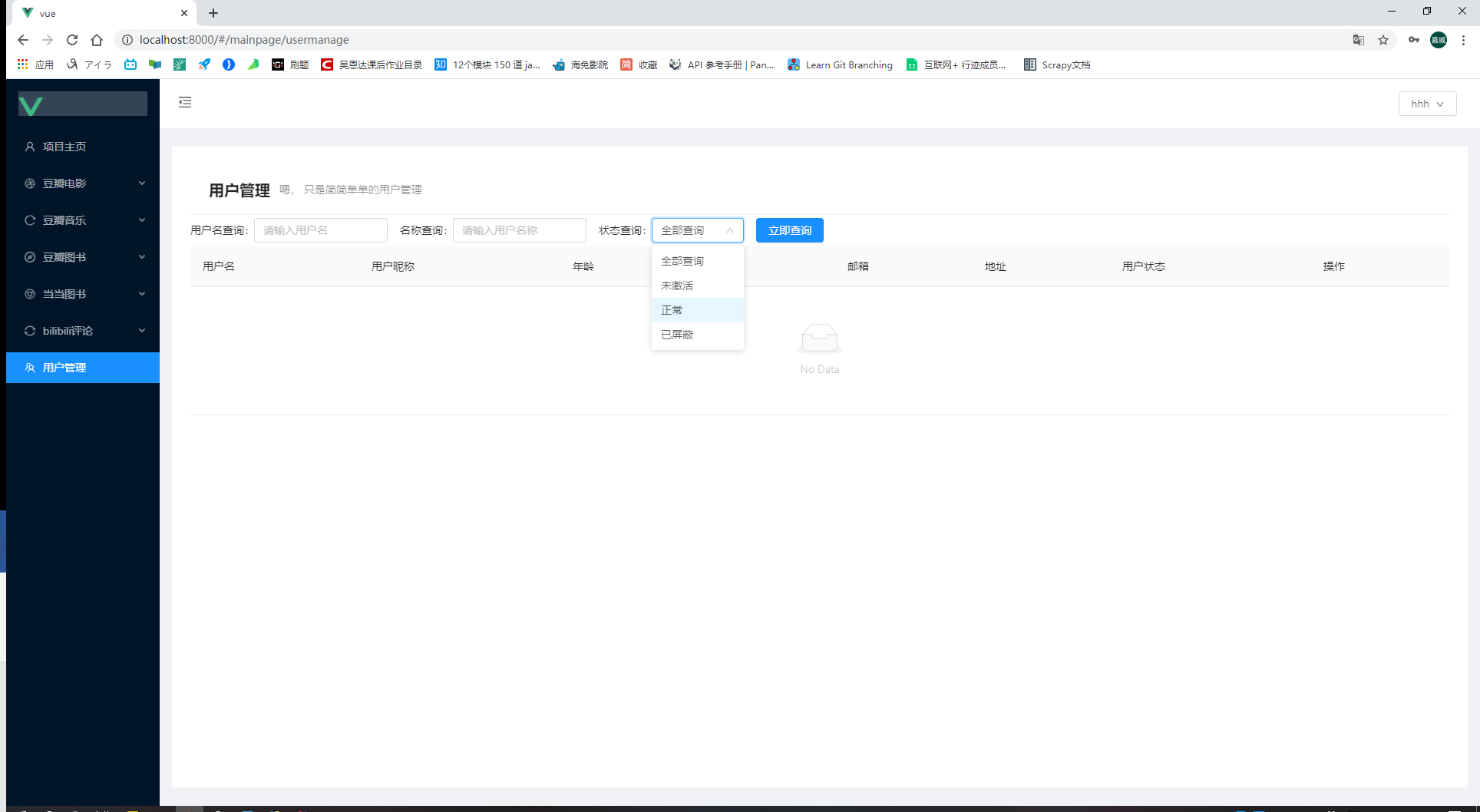


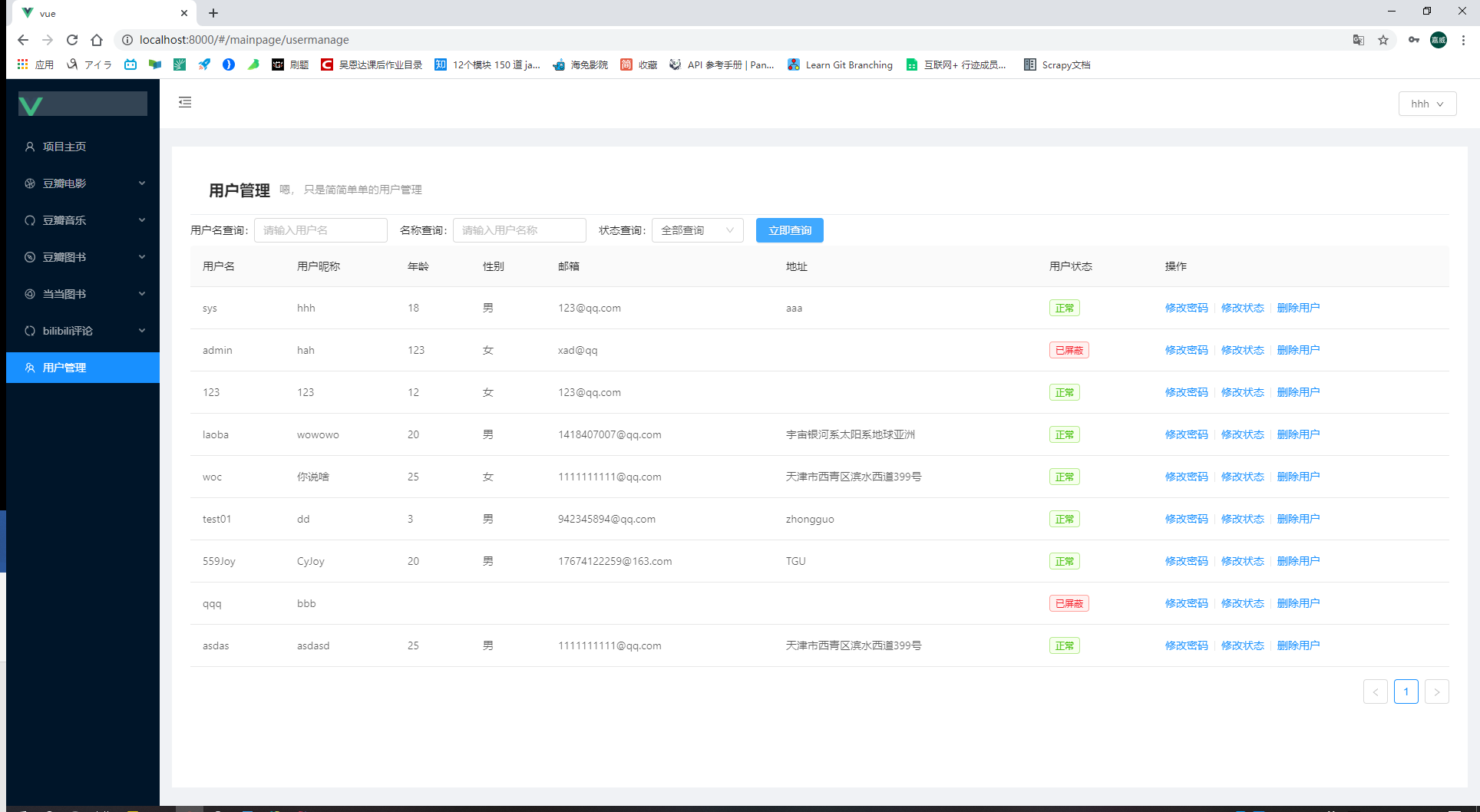


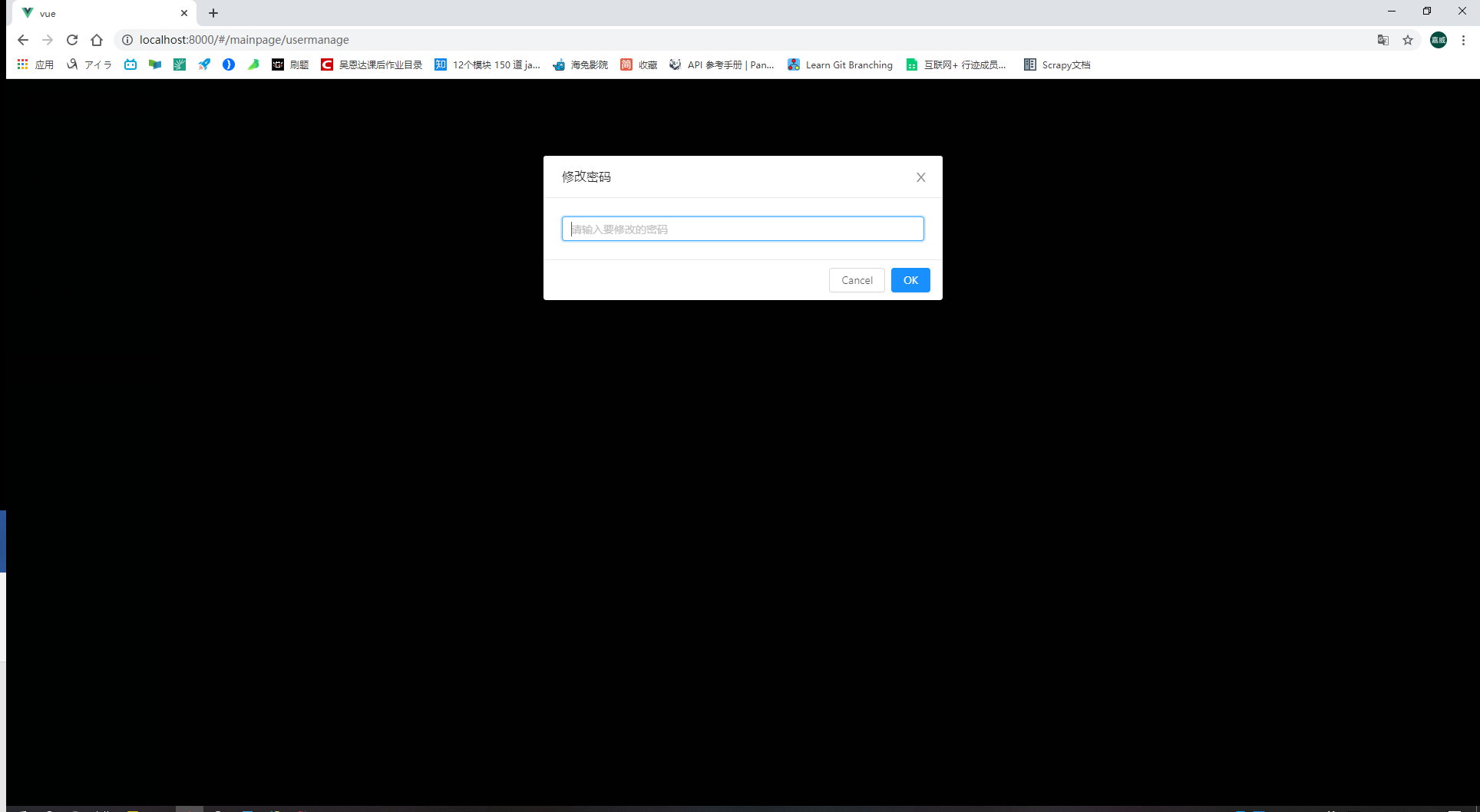


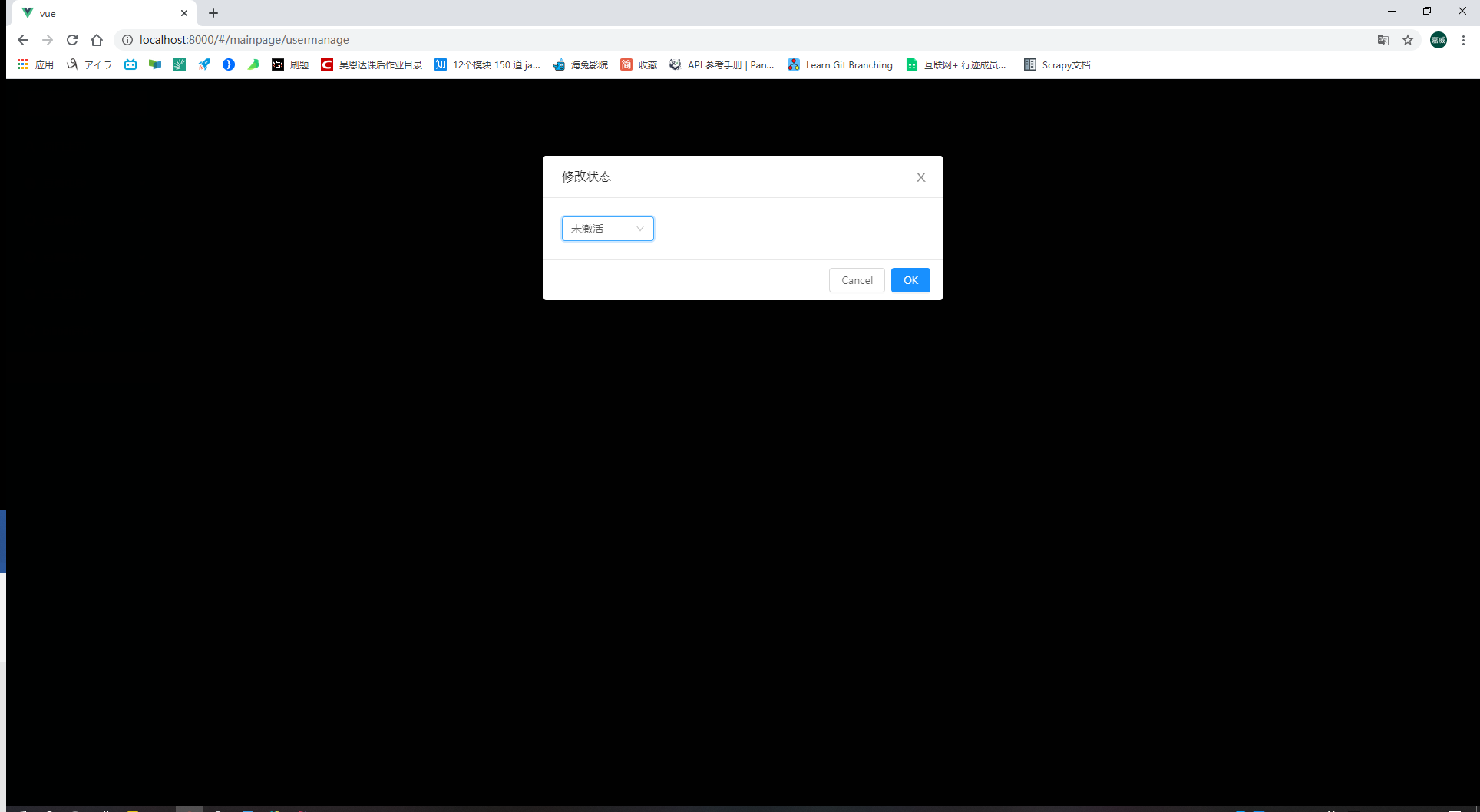
1. 用户管理

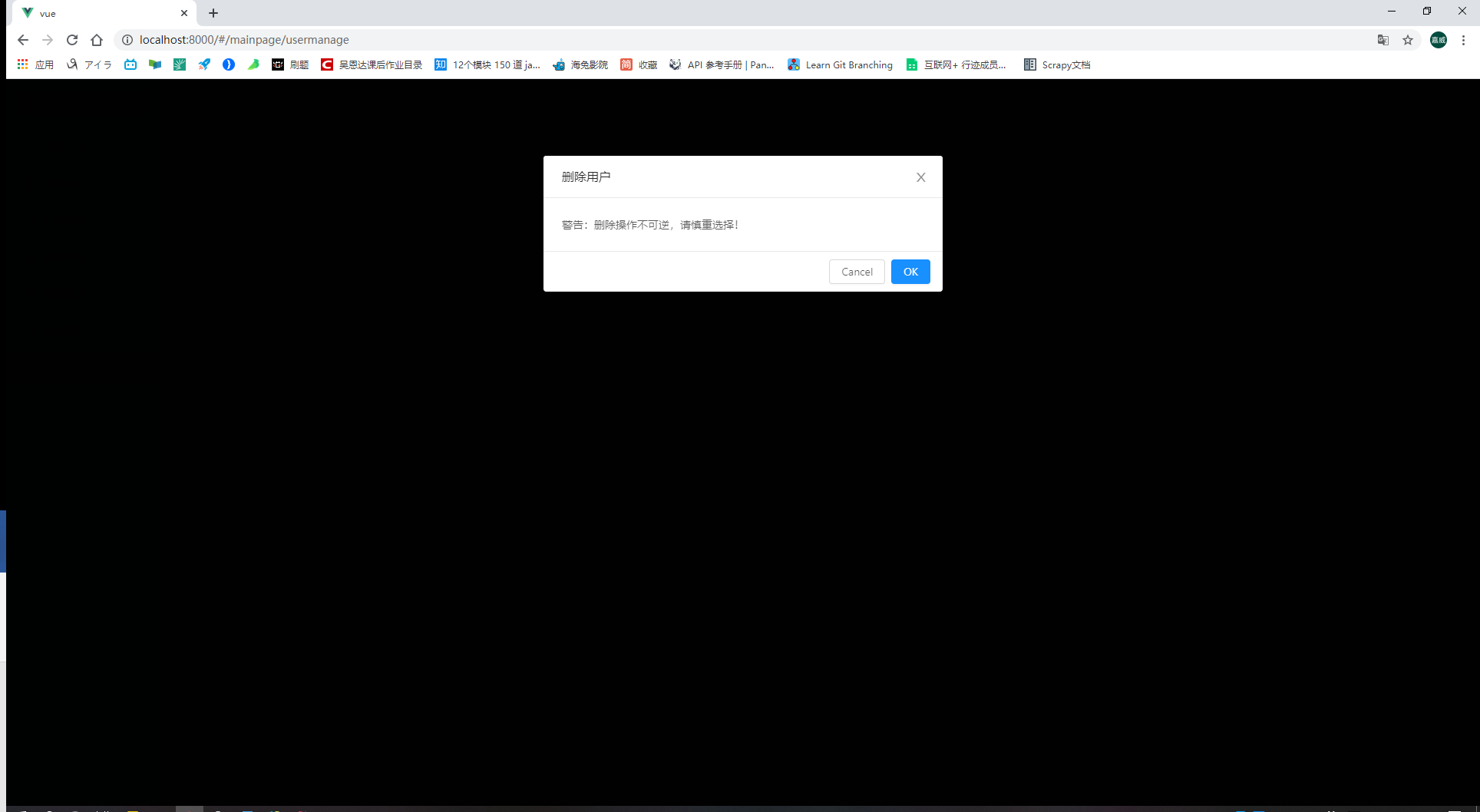
可以通过不同条件查询用户，同时对用户支持修改密码，修改状态和删除用户的操作。并且修改后会实时更近到视图层



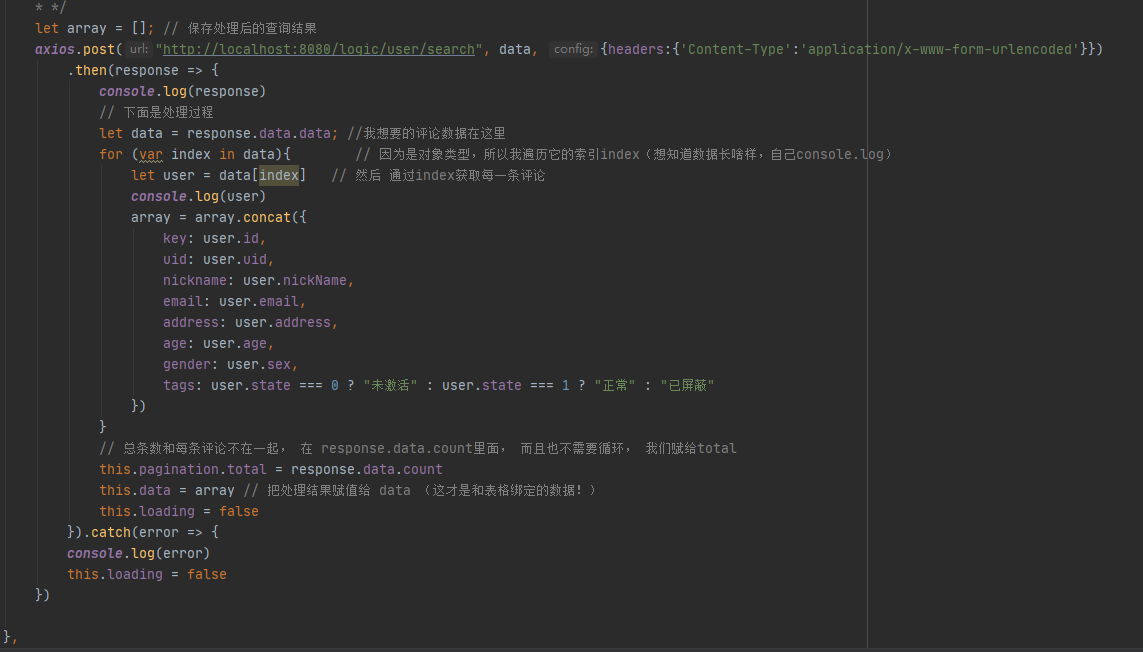




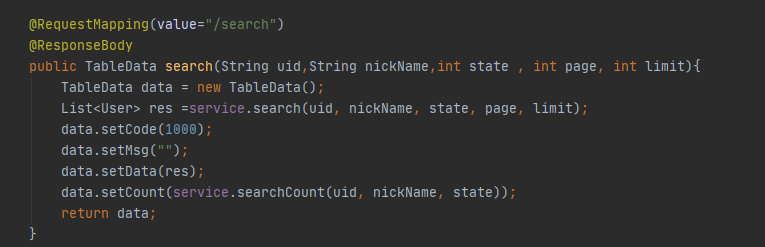


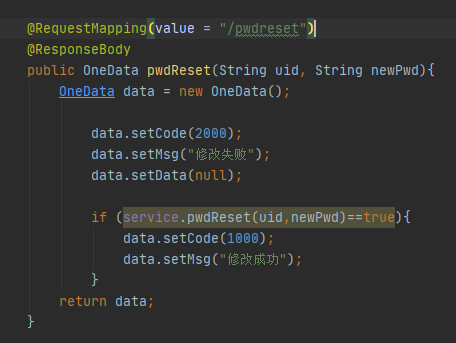


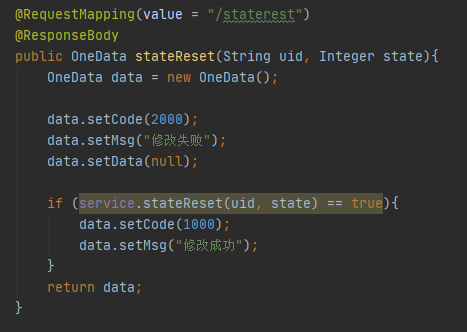
前端核心代码：

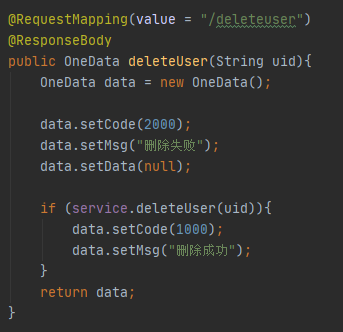


后端核心代码：







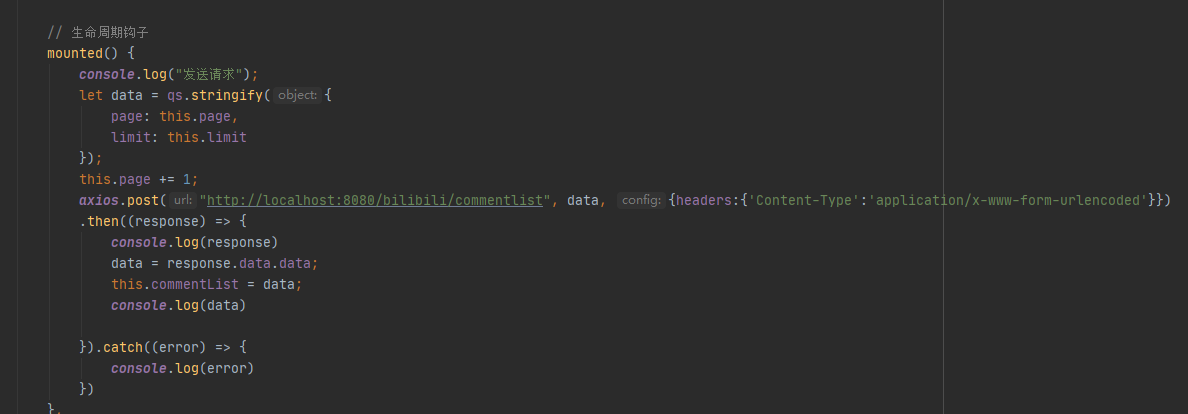


1. 评论展示

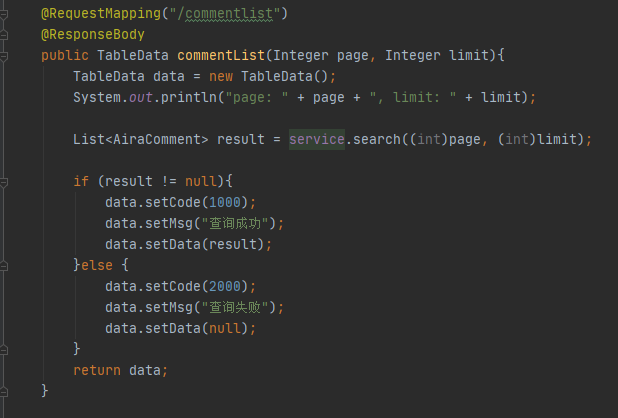
对用户评论进行列表展示，可以自己通过按钮来加载更多评论，同时支持back to the top的按钮操作

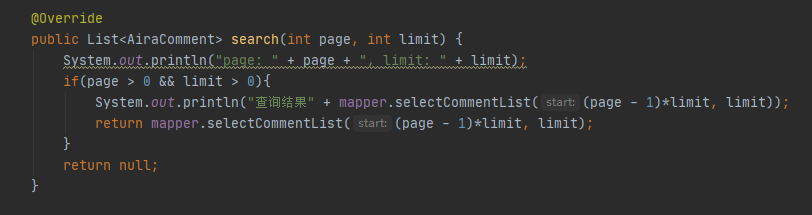


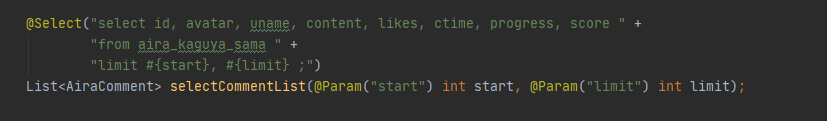


前端核心代码

后端核心代码

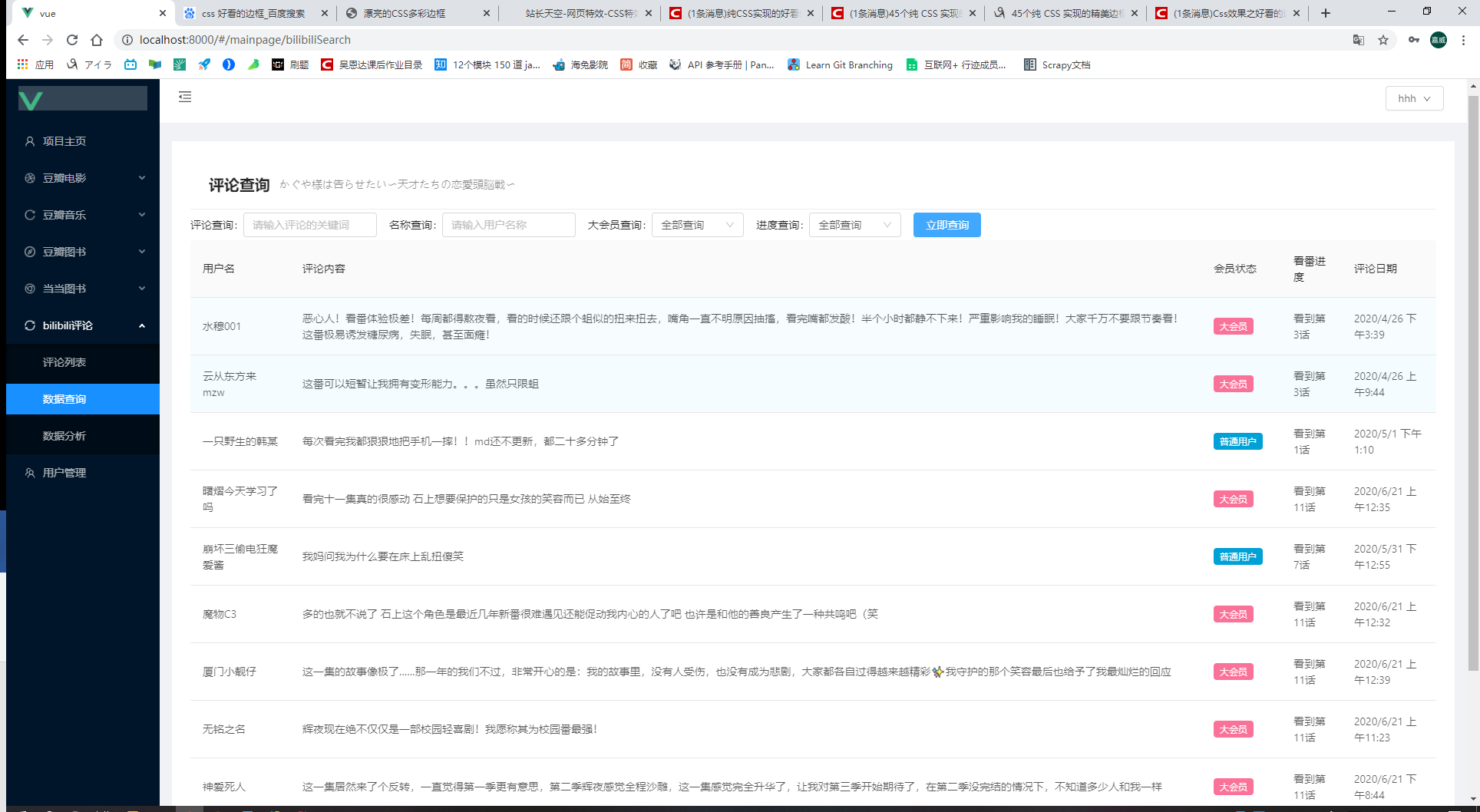




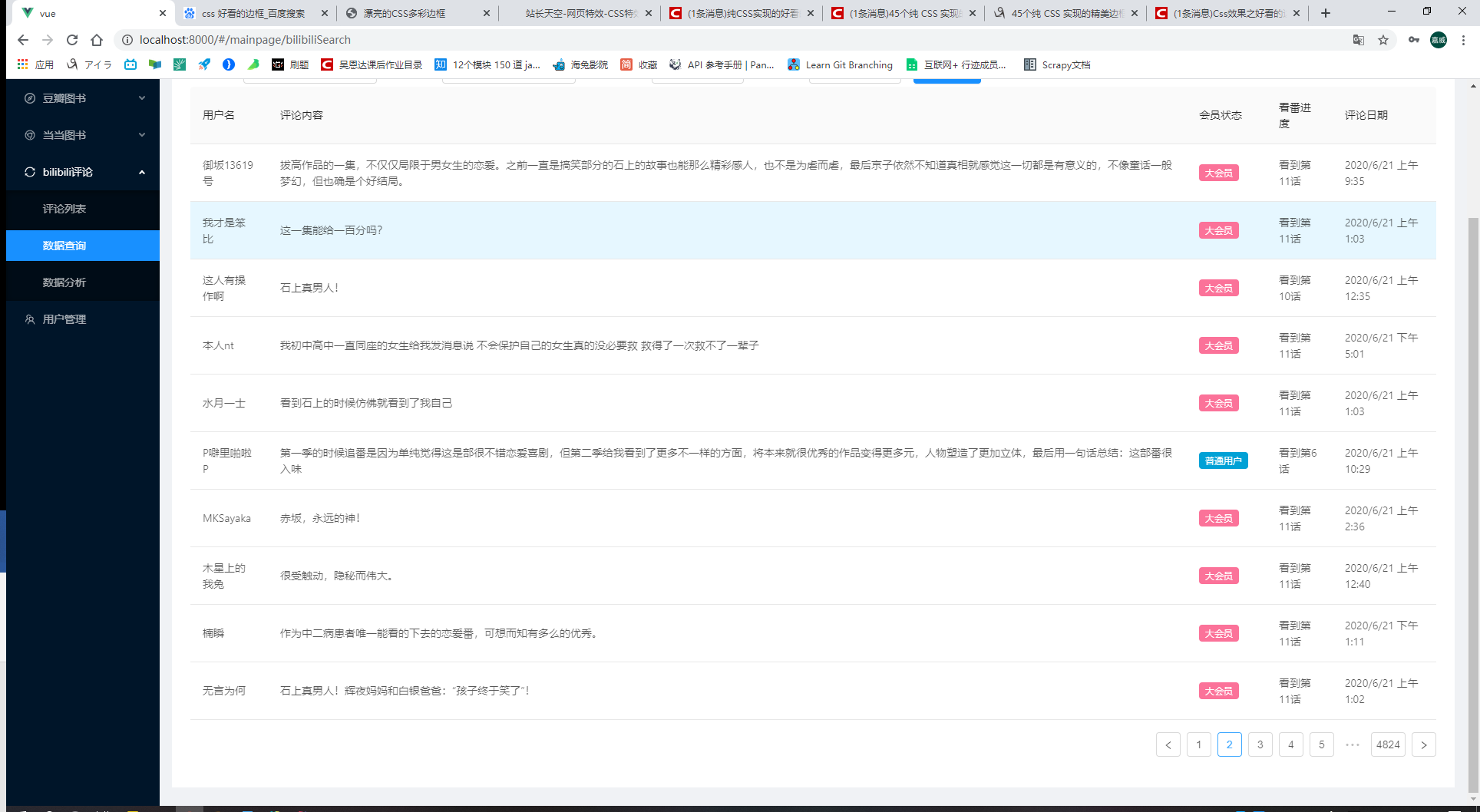


1. 数据查询

支持通过用户名，评论内容，用户是否是会员，以及观看进度进行查询



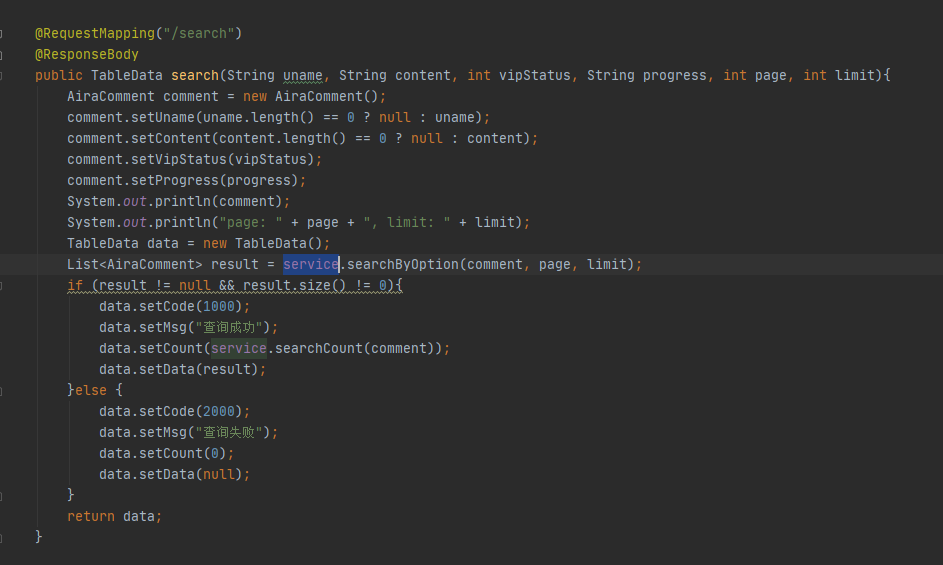
并且支持翻页

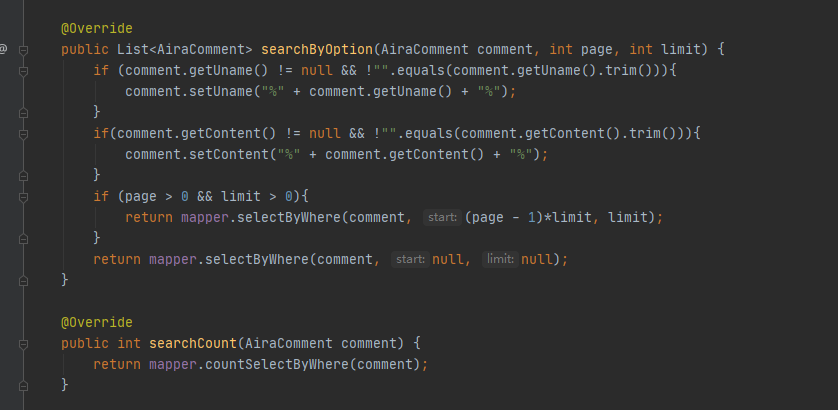


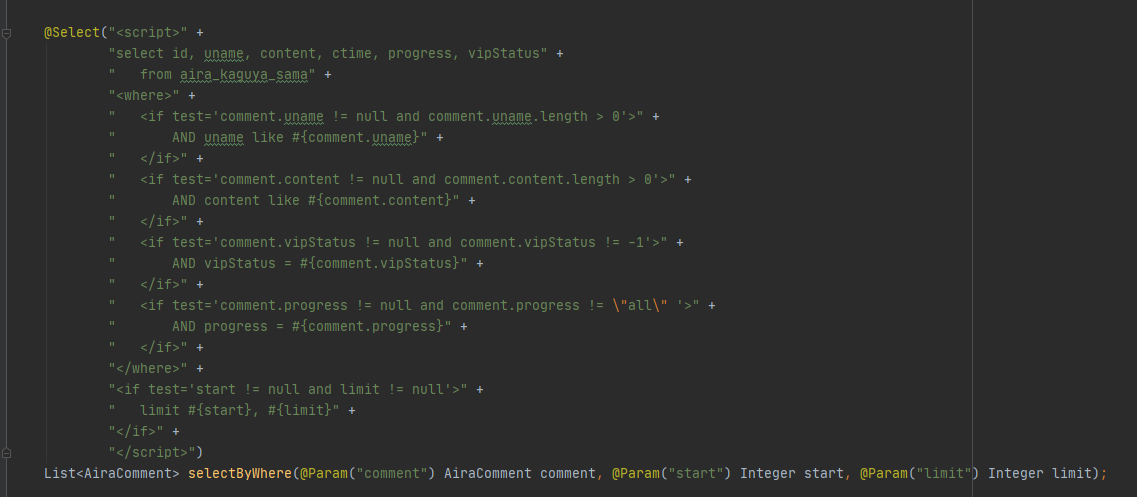
前端核心代码



后端核心代码

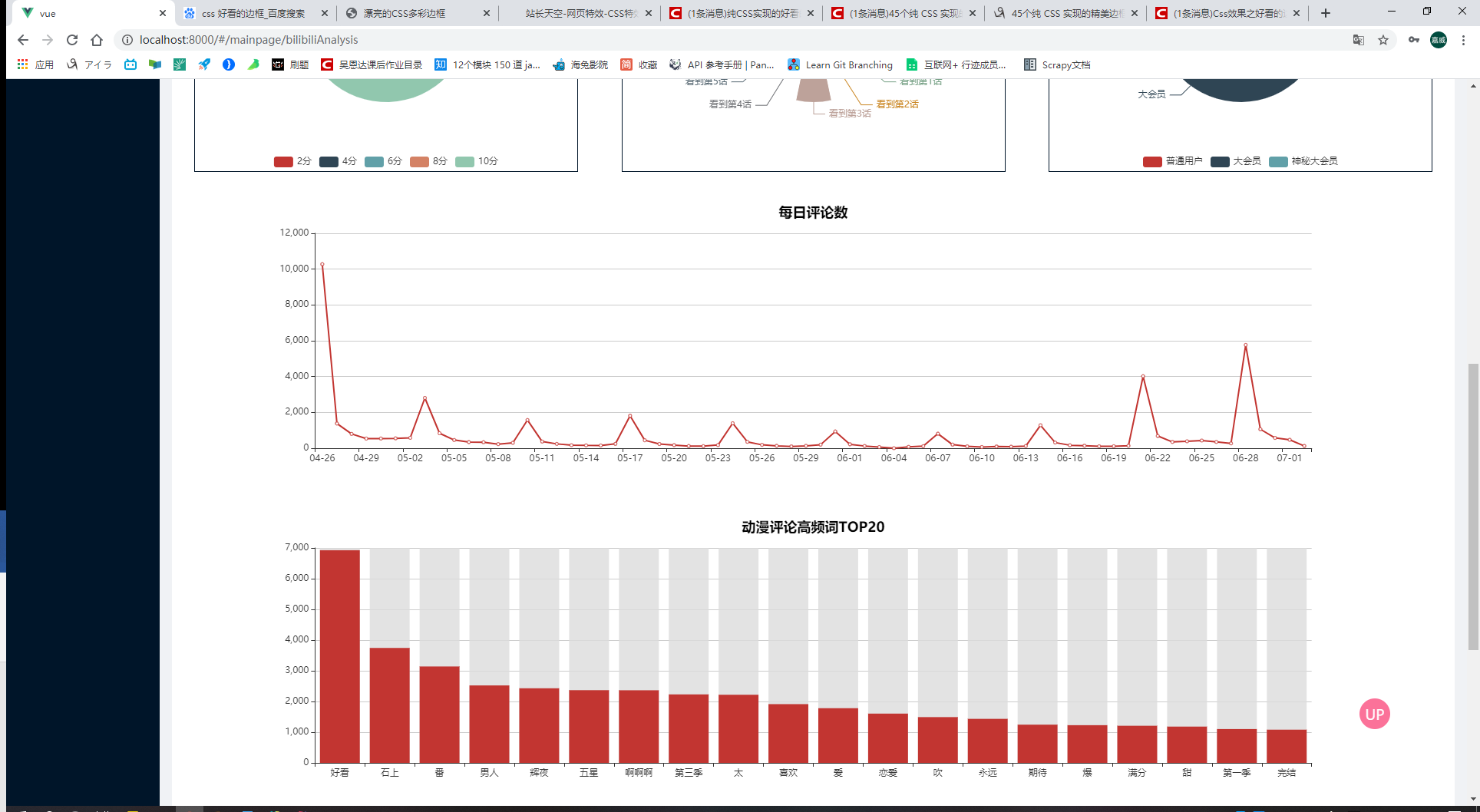






1. 数据分析







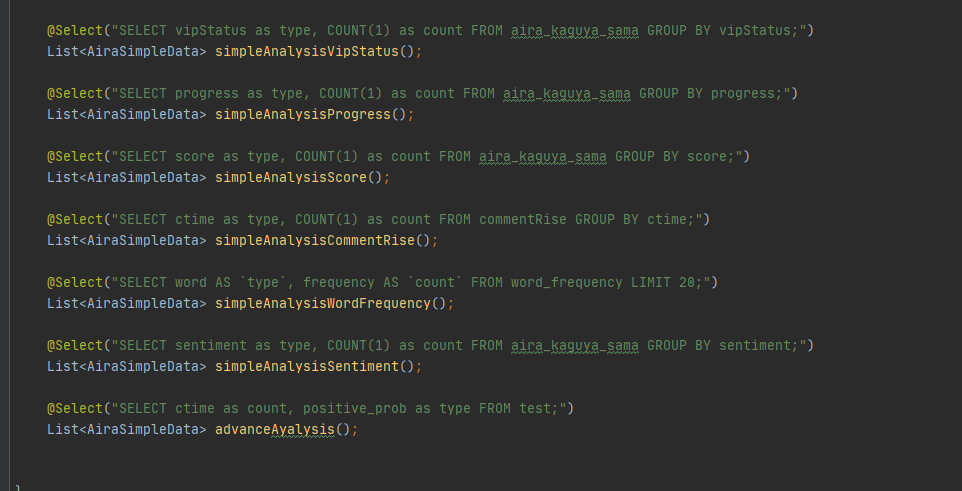
前端核心代码：

因为每个组件都独立，而且都要展示篇幅过多，此处忽略

后端核心代码：







**5 系统测试**（小三黑体，居中）

详细写明测试方法、测试描述（bug与修正方法等）、预期结果等；

查询大批量数据库缓慢，前后端引入limit 达到分页效果，预先加载首页内容，

增加新功能时，需要在首页mapping 后页mapping 和前端都进行同步修改，否则会导致数据库加载缓慢，或无法向后端发送数据，或前端无法接受数据