**目录**

[一、需求分析 1](#_Toc8042)

[1. 项目概述 1](#_Toc7422)

[2. 目标用户 1](#_Toc28604)

[3. 系统功能要求 1](#_Toc6598)

[二、系统分析与设计 4](#_Toc26316)

[1.系统设计(遇到的困难) 4](#_Toc18554)

[2.数据库的设计 11](#_Toc2716)

[三、系统实现 12](#_Toc2660)

[1、 web界面显示 13](#_Toc1394)

[2、 App界面显示 16](#_Toc32329)

[四、课设总结 18](#_Toc2606)

[1、 心得、收获、体会 18](#_Toc9717)

[2、 对课程的建议 20](#_Toc27694)

# 一、需求分析

## 1. 项目概述

本系统主要实现一个包含APP端和Web端的数据采集和管理平台。其核心功能包括通过APP进行数据采集、在Web端进行数据展示和管理、生成Excel表格等。

## 2. 目标用户

* **APP端用户**：数据采集人员，主要负责在现场采集数据，并上传至系统。
* **Web端用户**：管理员或其他数据审核人员，主要负责查看和管理采集的表格数据。

## **3. 系统功能要求**

### 3.1 APP端功能要求

**登录功能**  
用户使用账号和密码登录，验证身份后才能进行数据采集。

**采集目录**  
用户登录后可以查看可用的采集表格。每个表格均有唯一标识，点击某个表格进入具体的采集页面。

**数据采集**  
用户在采集页面输入黄色单元格对应的数据，蓝色单元格的数据由系统根据公式自动计算。用户不需要手动录入蓝色单元格数据。

**自动采集数据**  
系统自动采集采集时间和地点（经纬度），这些信息无需用户输入，直接记录到数据库。

**提交数据**  
完成数据输入后，用户可以提交数据，系统会将数据存入数据库中，包括时间、地点、采集人员ID等字段。

### **3.2 Web端功能要求**

**登录功能**  
Web端用户（管理员）需登录后才能进入系统查看和管理数据。

**采集表格目录**  
登录后，管理员能够看到所有已经上传的数据采集表格列表。管理员可以通过点击进入具体表格查看数据

**表格展示与修改**  
具体表格中会展示所有采集的数据，管理员可以修改黄色单元格的数据（前端提供输入框，数据更新后自动保存至数据库）。

**数据导出功能**  
Web端支持将采集的数据以Excel表格形式导出，导出的表格格式与初始模板保持一致。

**数据审阅与管理**  
Web端需要提供审核、修改功能，以确保采集的数据的准确性。管理员可以对数据进行必要的修改。

### **3.3 数据库设计要求**

**表格命名**  
每张表格的数据库表名应与表格名称中的编号部分一致。例如，表格“1-记录表SJ01 路基清理与挖除记录表-OK”对应数据库表名为“SJ01”。

**字段命名规则**  
每个字段的名字由其中文名的首字母组成。例如，“分计留筛土质量(g)”的字段名应为“fjlstzl”。

**时间与地点字段**  
每张表格中都必须包含字段：

**time**：采集时间（记录数据采集时间戳）。

**position**：采集地点（经纬度，保存为字符串类型，保存地点信息）。

**userid**：采集人员编号（标识采集数据的用户ID）。

**字段数据类型与长度**  
所有字段的数据类型均为字符串类型，长度暂定为64个字符。

### **3.4 系统整合要求**

**APP端与Web端的数据同步**  
系统需要保证APP端采集的数据能够及时、准确地同步到Web端，管理员可以在Web端进行查看和管理。

**Excel表格导出与生成**  
在Web端和APP端均需要提供Excel表格的导出和生成功能，确保导出的Excel文件格式与用户提供的模板一致。

**自动计算功能**  
系统需要在APP端自动根据给定公式计算蓝色单元格的数据。这个功能需要在APP端的前端逻辑中实现，并通过接口与后端交互，将计算结果保存到数据库。

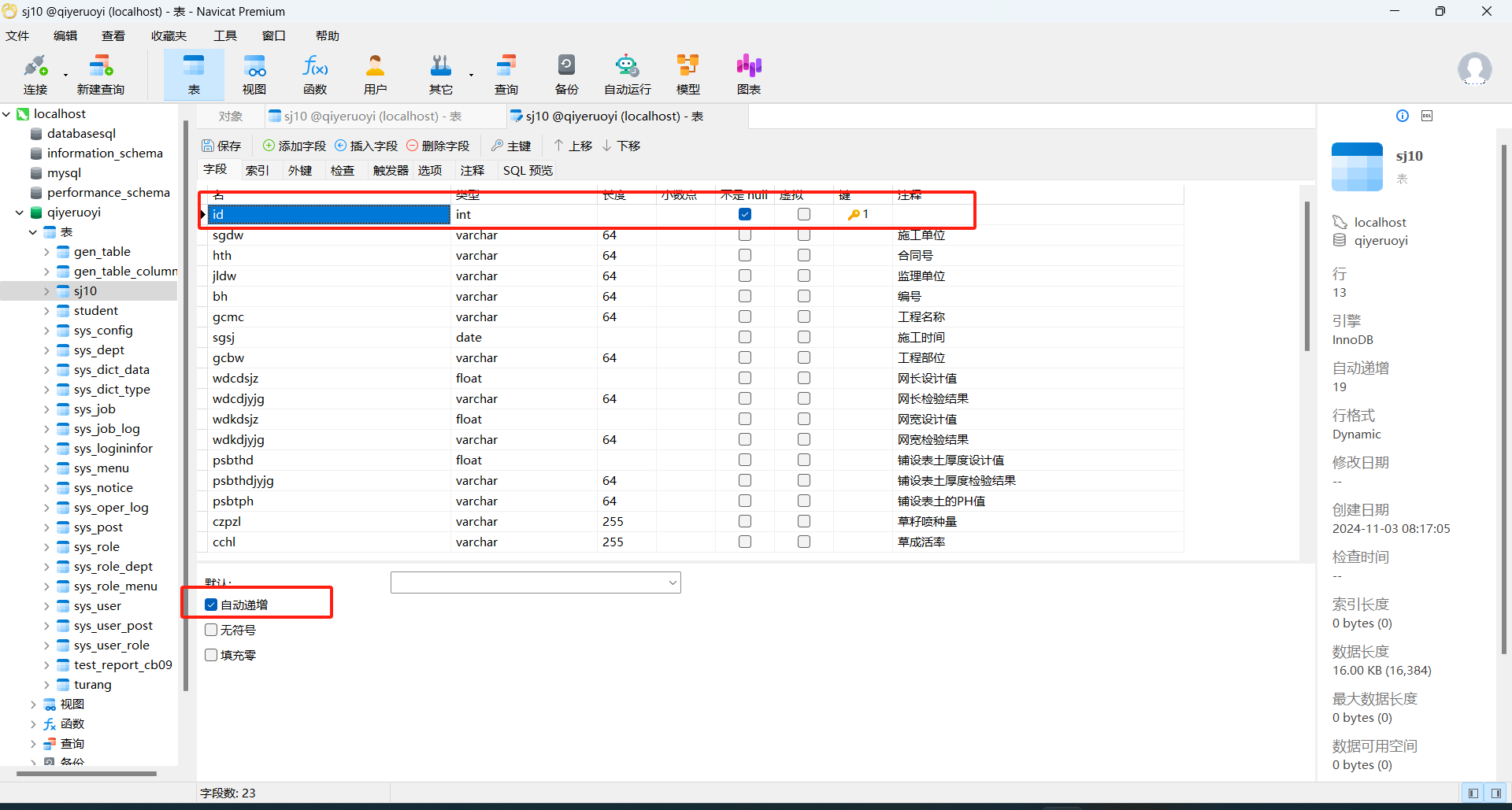
**权限管理**  
系统需要有合理的权限管理机制，确保不同角色的用户访问和操作权限不同。

本需求分析文档针对APP端和Web端的功能进行了详细的说明，涵盖了数据采集、显示、修改、导出等多项核心功能，确保系统能够高效、准确地完成数据管理和展示任务。

# 二、系统分析与设计

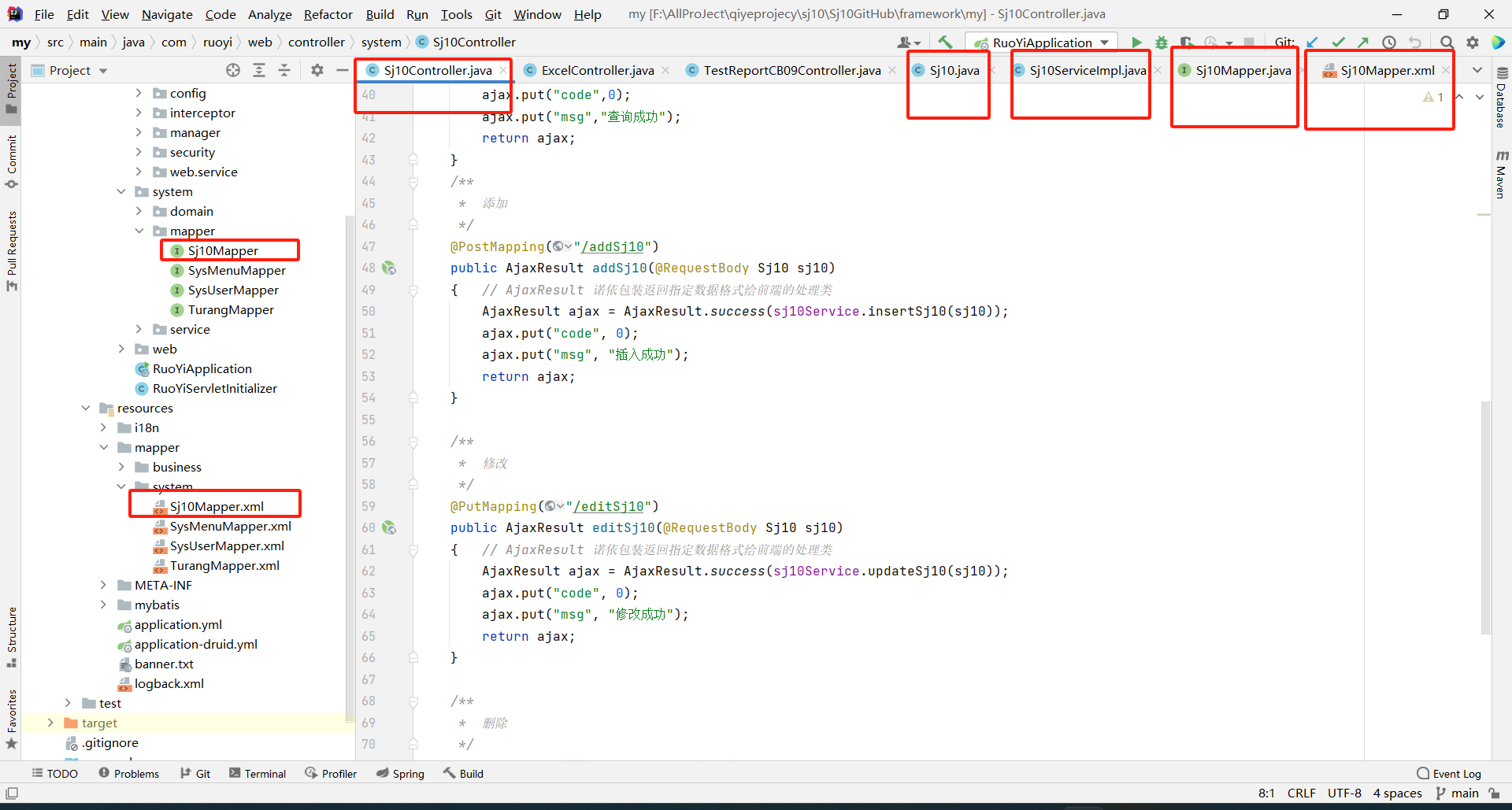
## 1.系统设计(遇到的困难)

**1、首先遇到的问题是不了解字段自动递增的含义。**



我之前没有勾选这个选项，导致我在web端上显示数据失败，无法查询到数据库中的数据。之后我查询了相关信息了解到，可能是由于没有设置自动递增，导致在添加数据的时候，因为这个id仅是用来对于数据的检索，而不是显示，因此添加的时候其是默认值0，而他又是主键，相同的值会引起冲突从而导致发生添加失败。

1. **其次是对于项目文件的结构认识不清晰，导致对于按照老师给的模板知己弄的时候会忘记添加相应的文件。**



这几个文件，我之前写的时候有点迷茫，之后去查了相关信息。

知道了这几个文件的含义。

**Controller**：处理 HTTP 请求，调用服务层方法，返回响应结果。

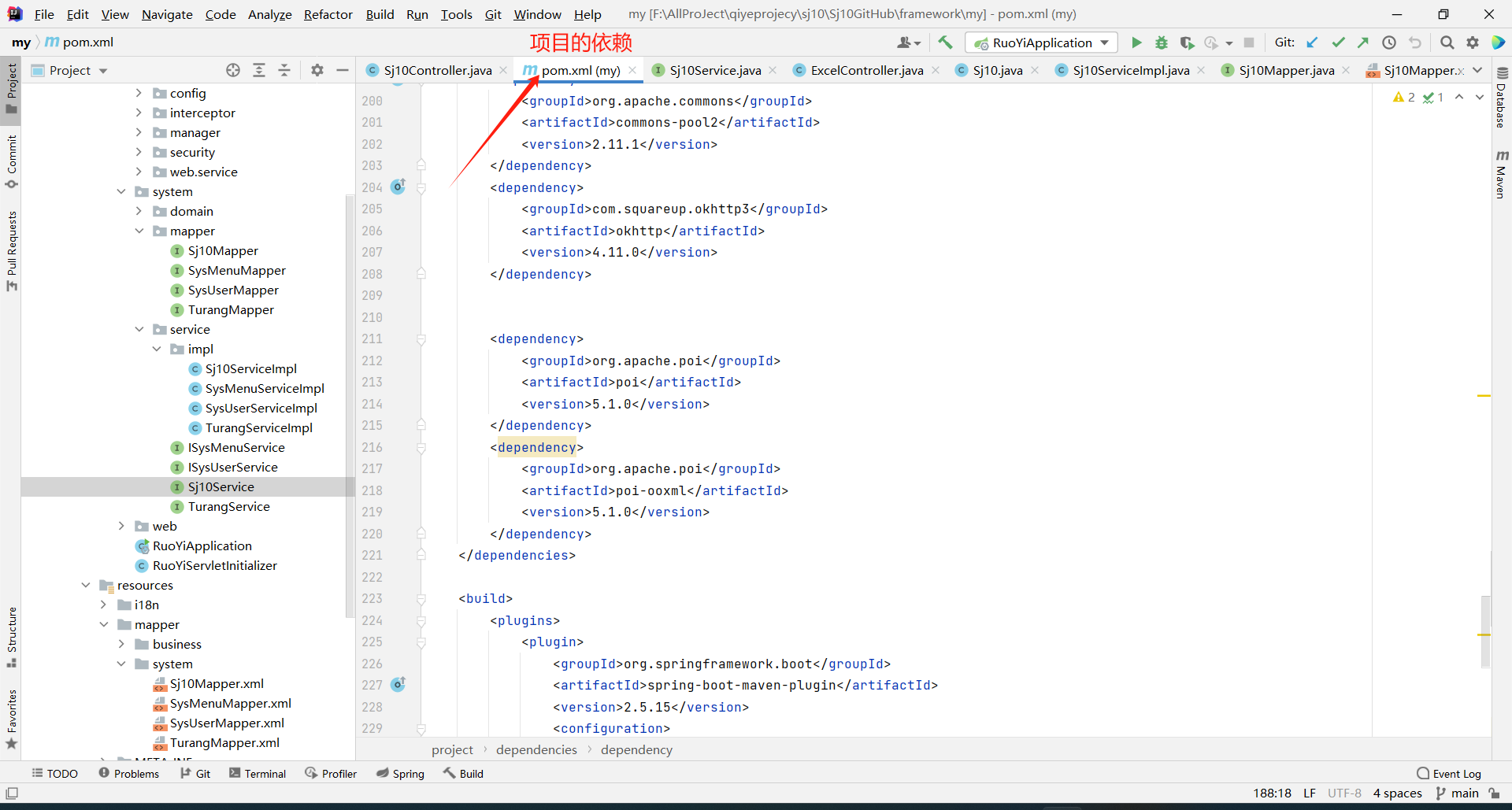
**实体类（Entity）**：表示数据库表结构，用于数据传输。

Service 实现类（**ServiceImpl**）：处理业务逻辑，调用 Mapper 层方法访问数据库。

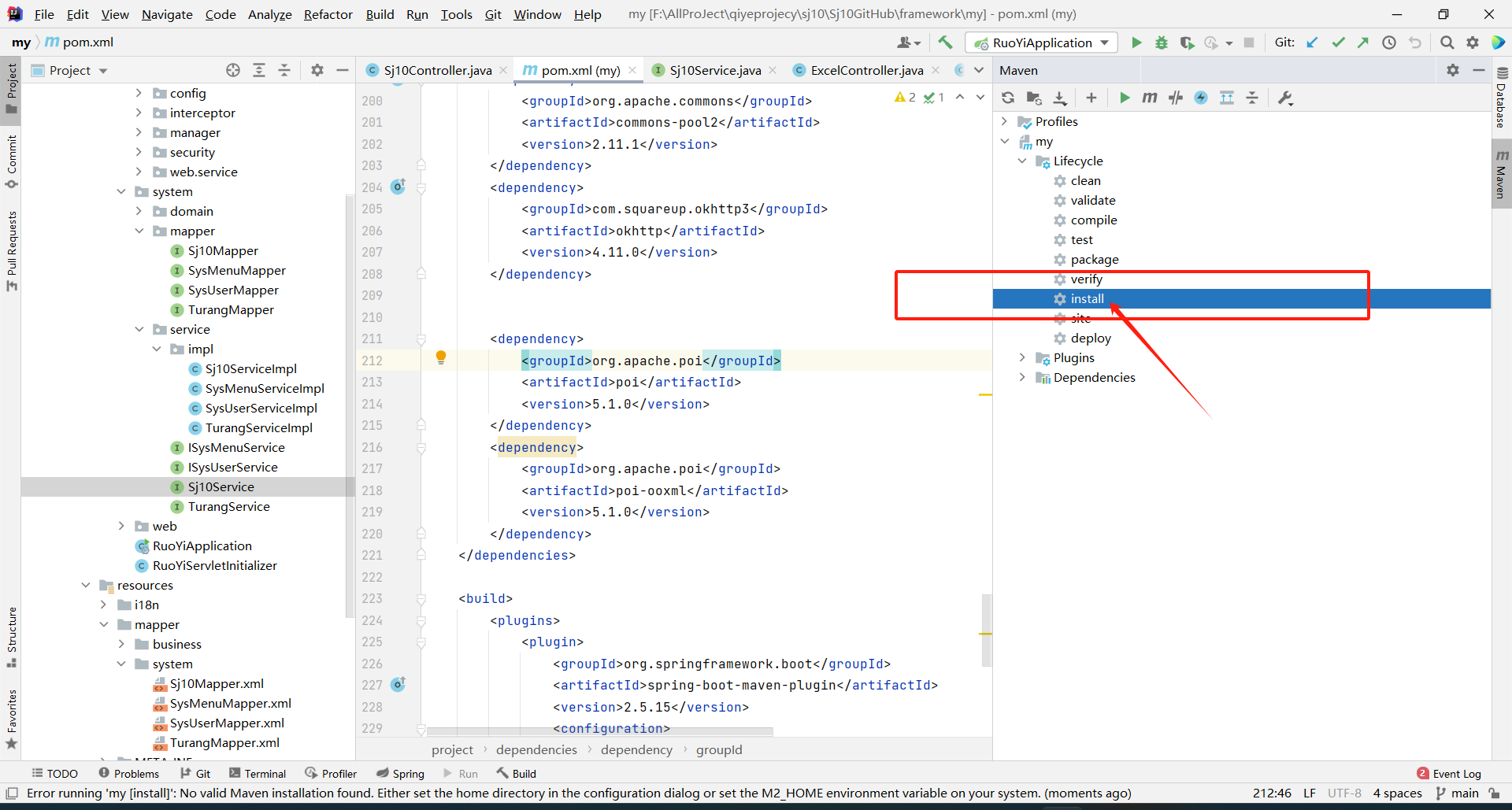
**Mapper** 文件：定义 SQL 语句和结果映射，用于与数据库交互。

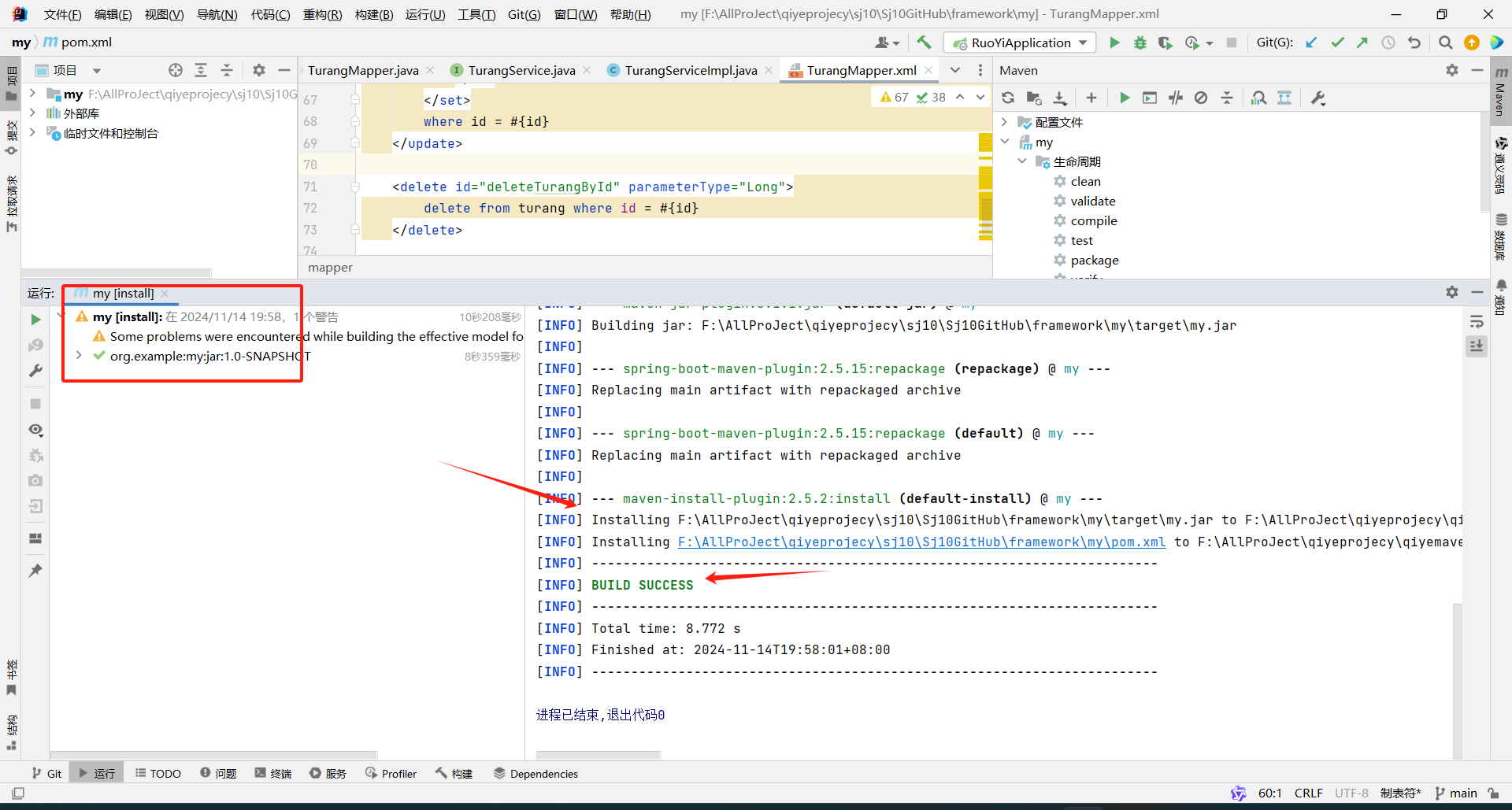
**Service**层：处理应用程序的业务逻辑。业务逻辑是指应用特有的规则、计算和流程，这些逻辑通常与具体的应用场景相关

1. **对于pom不了解，之前项目启动总是无法启动，发现是相关依赖没有安装**



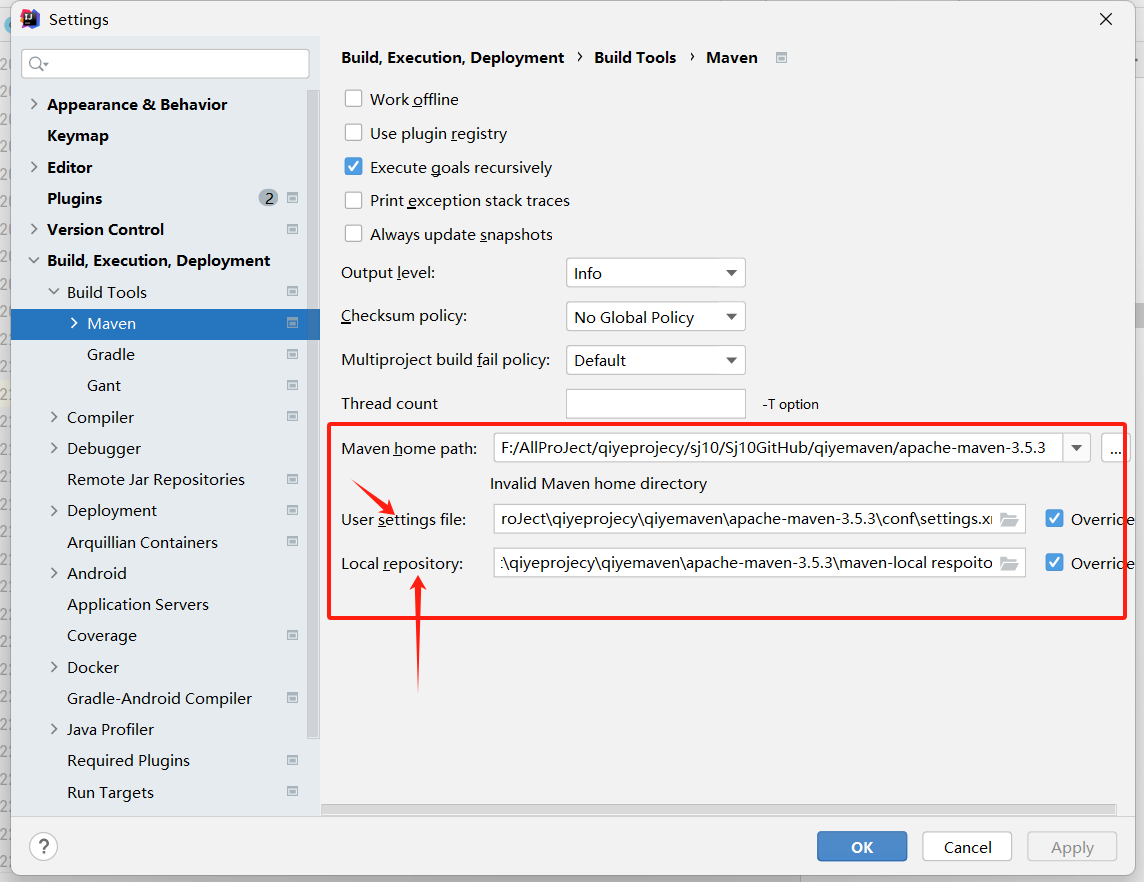
解决办法是通过点击IDEA自带的pom插件，点击项目的Maven之后在项目文件下，点击install，install的主要任务是将构建好的项目包（如 JAR、WAR 文件）安装到本地 Maven 仓库中。如下如所示。

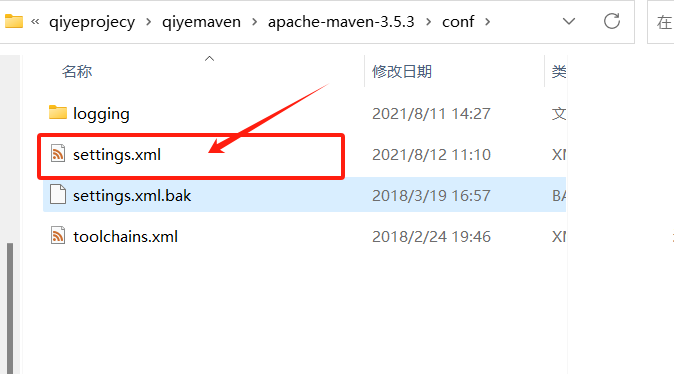


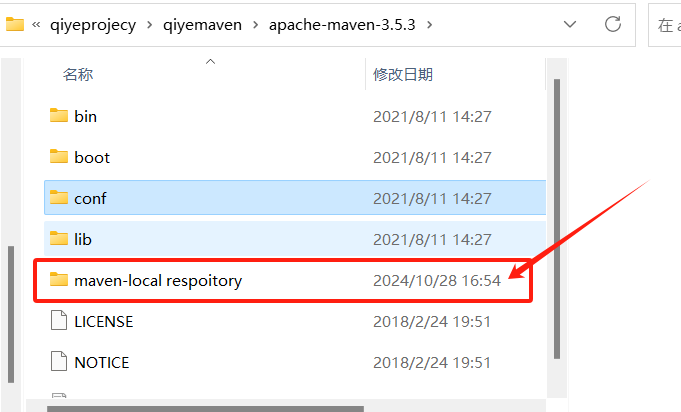


1. **Maven的配置问题**

之前Maven导入错误，使得项目无法启动。原因我发现是我的Maven的设置文件和本地仓库的路径没有选对。之后修正如下图。

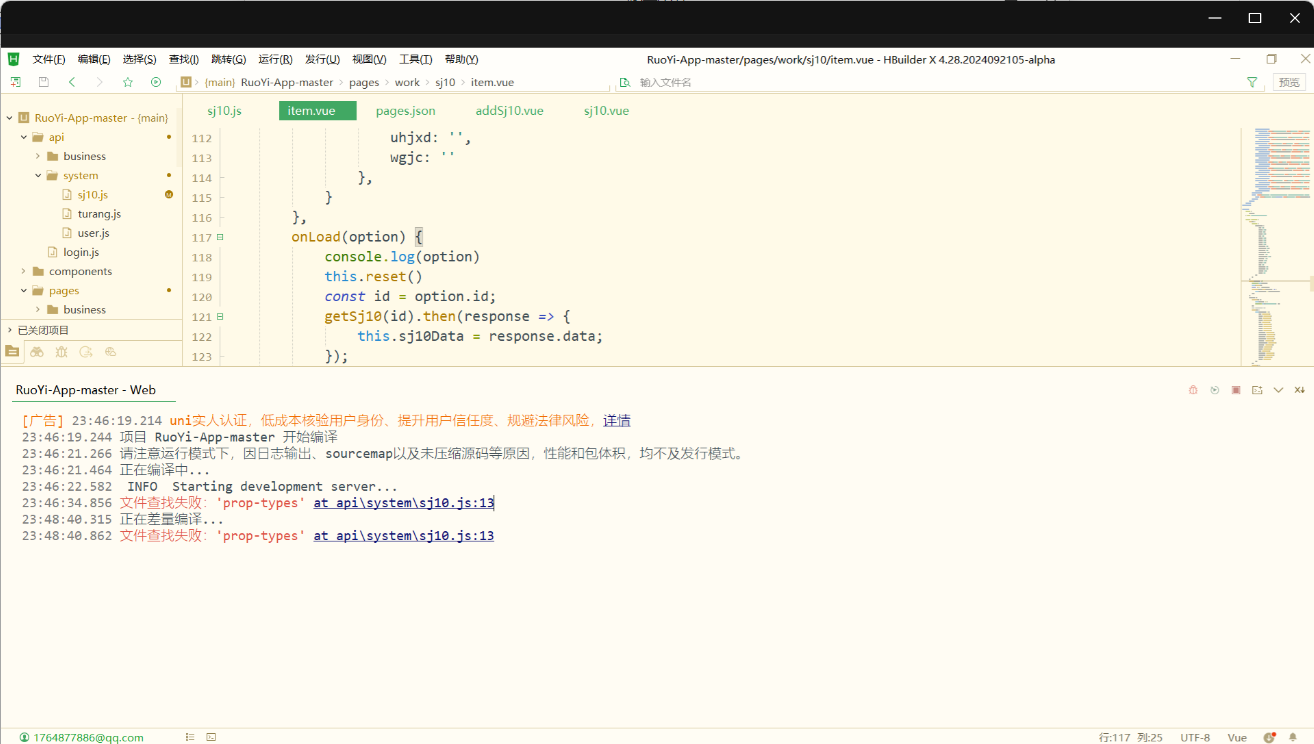




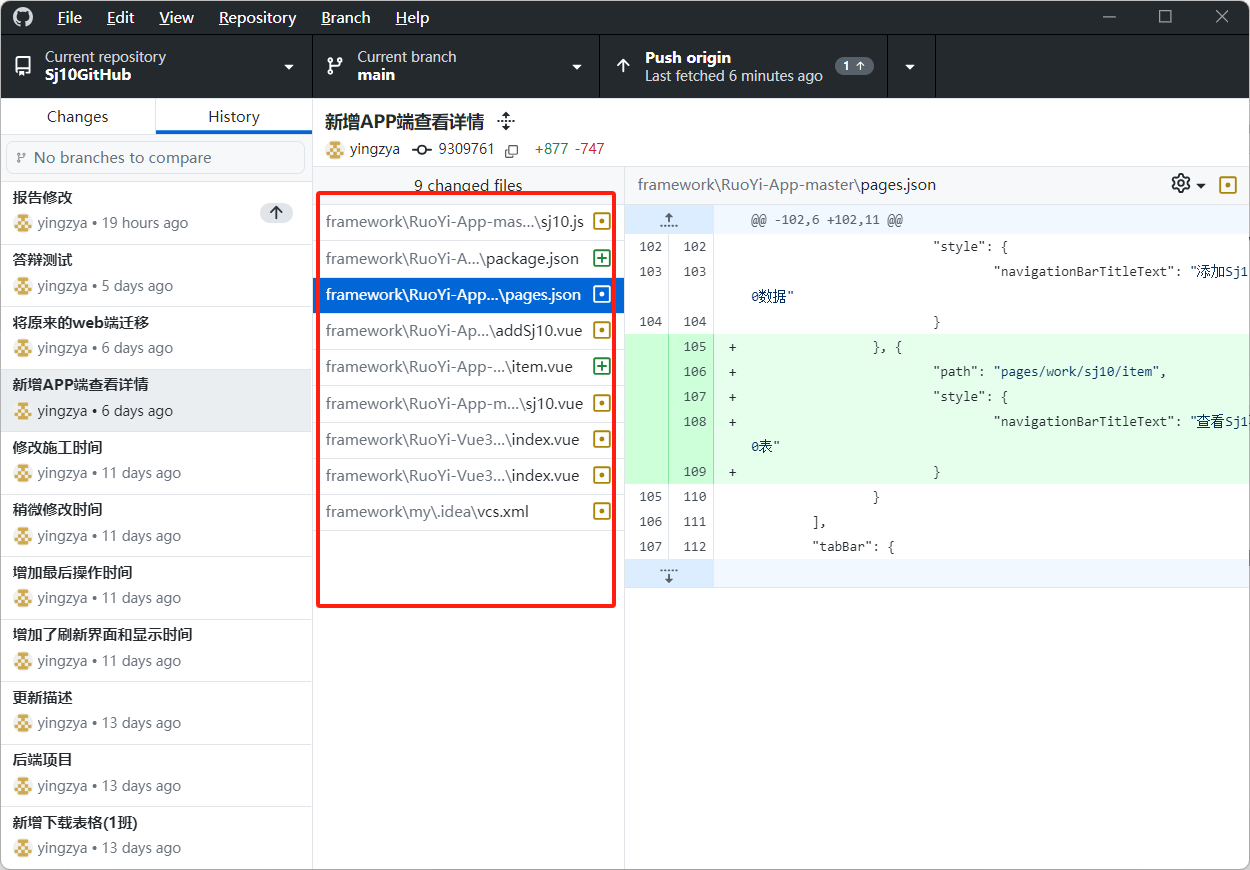


1. **App端的添加详情的增加**

这里由于我并不熟悉前端知识，是请教的同学，我按他的方法操作之后，出现了以下的情况，首先是文件查找失败。之后我通过百度发现需要安装某个包，具体的忘记了，当时忘记截图了。在安装好之后就行了。



**图2-1文件查找失败**



**图2-2修改的文件**

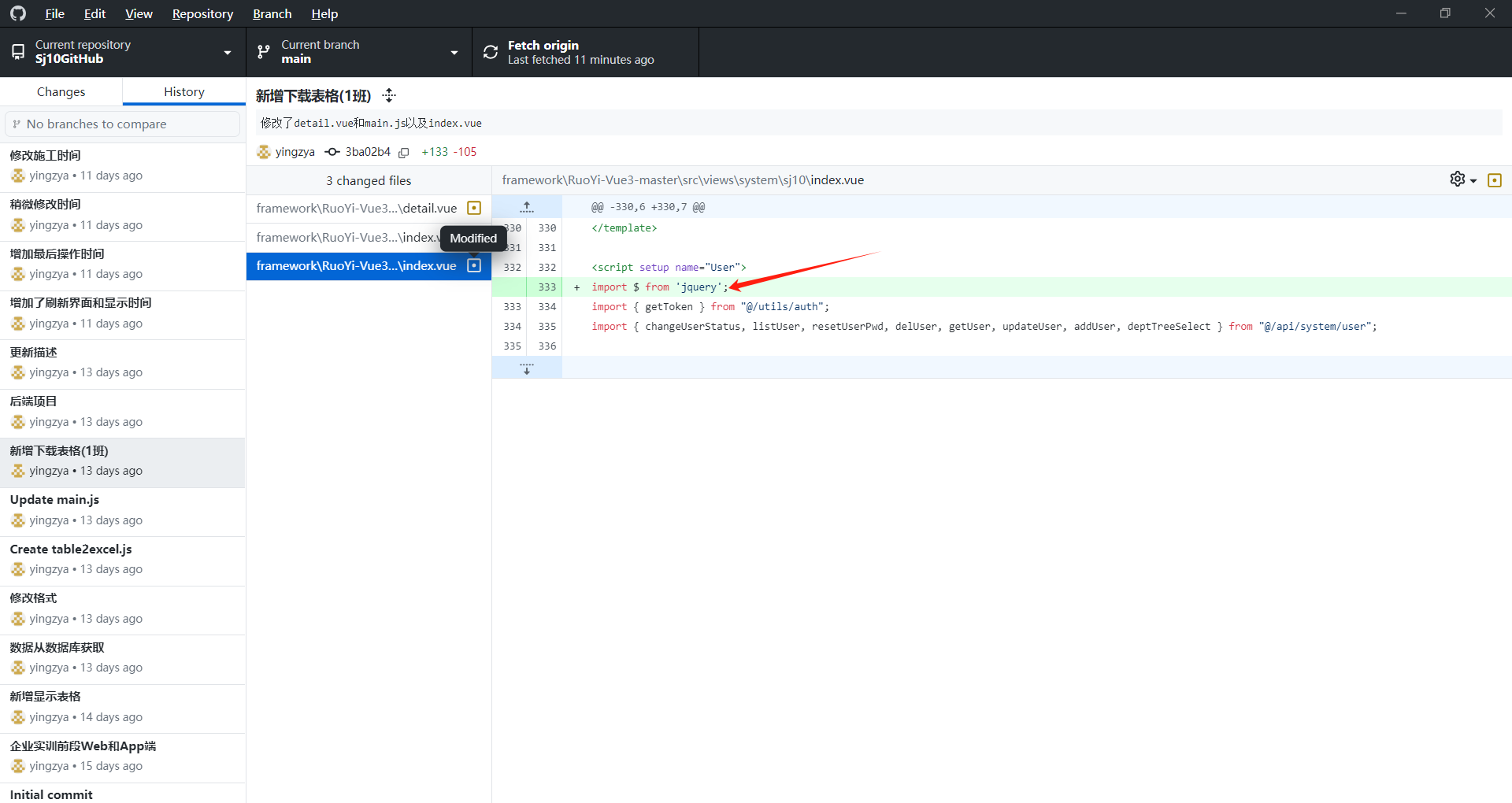
由于修改的文件较多，这里并不一一截图，只显示修改的文件了，具体可以通过我的代码仓库访问。仓库地址：<https://github.com/yingzya/Sj10GitHub.git>

1. **表格Sj10的格式代码以及生成excel文件**

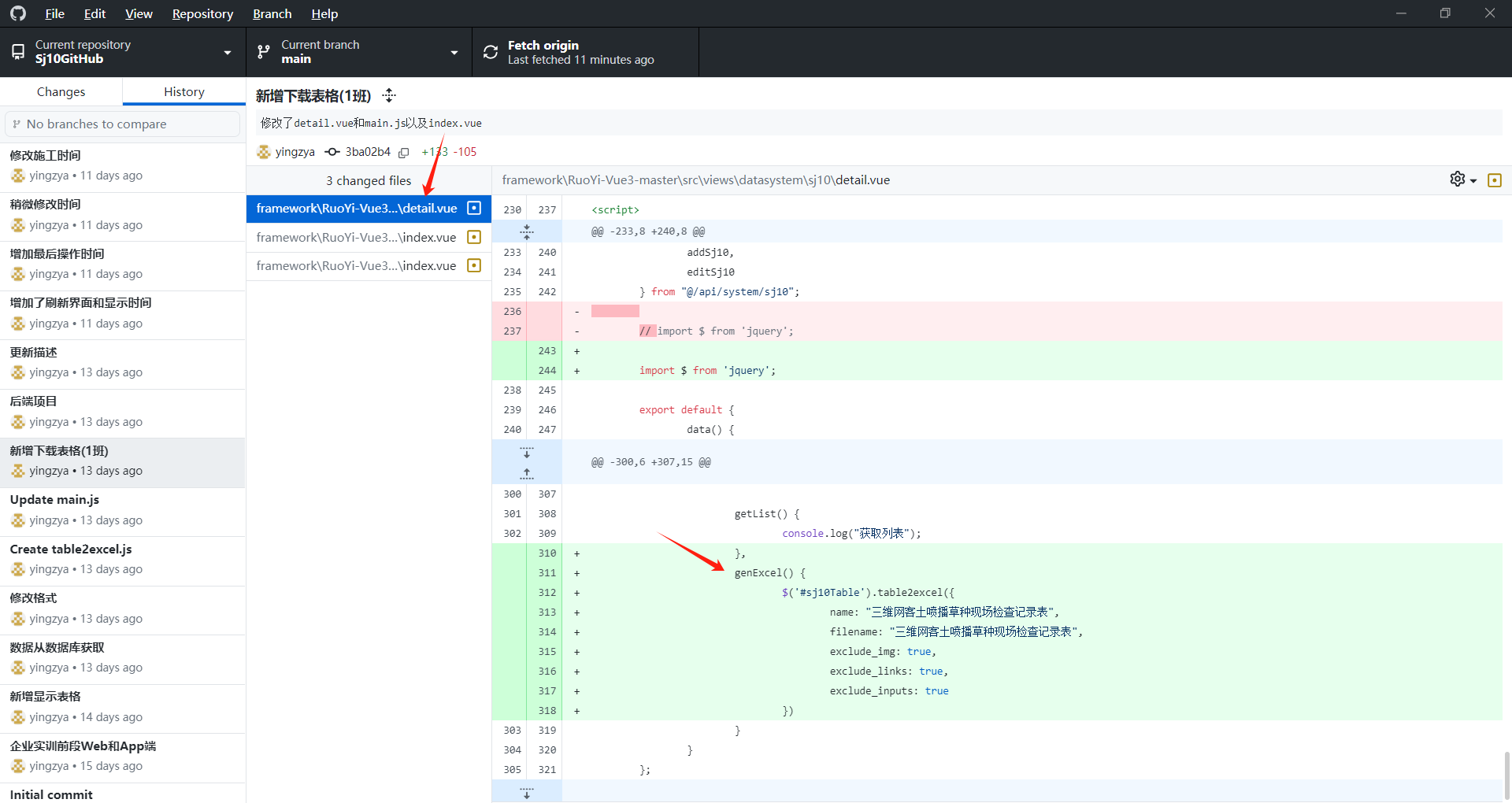
先说生成excel，由于Sj10的单元格较多，以及每行的单元格合并的数量并不一样，我之前用老师教的一个一个单元格去拼表的方式，在弄了20多个之后放弃，感觉这样太慢了。同时有时候修改还经常报错。因此我换成了1班老师的方法。解决的方法如下图2-3所示。



**图2-3生成excel文件**

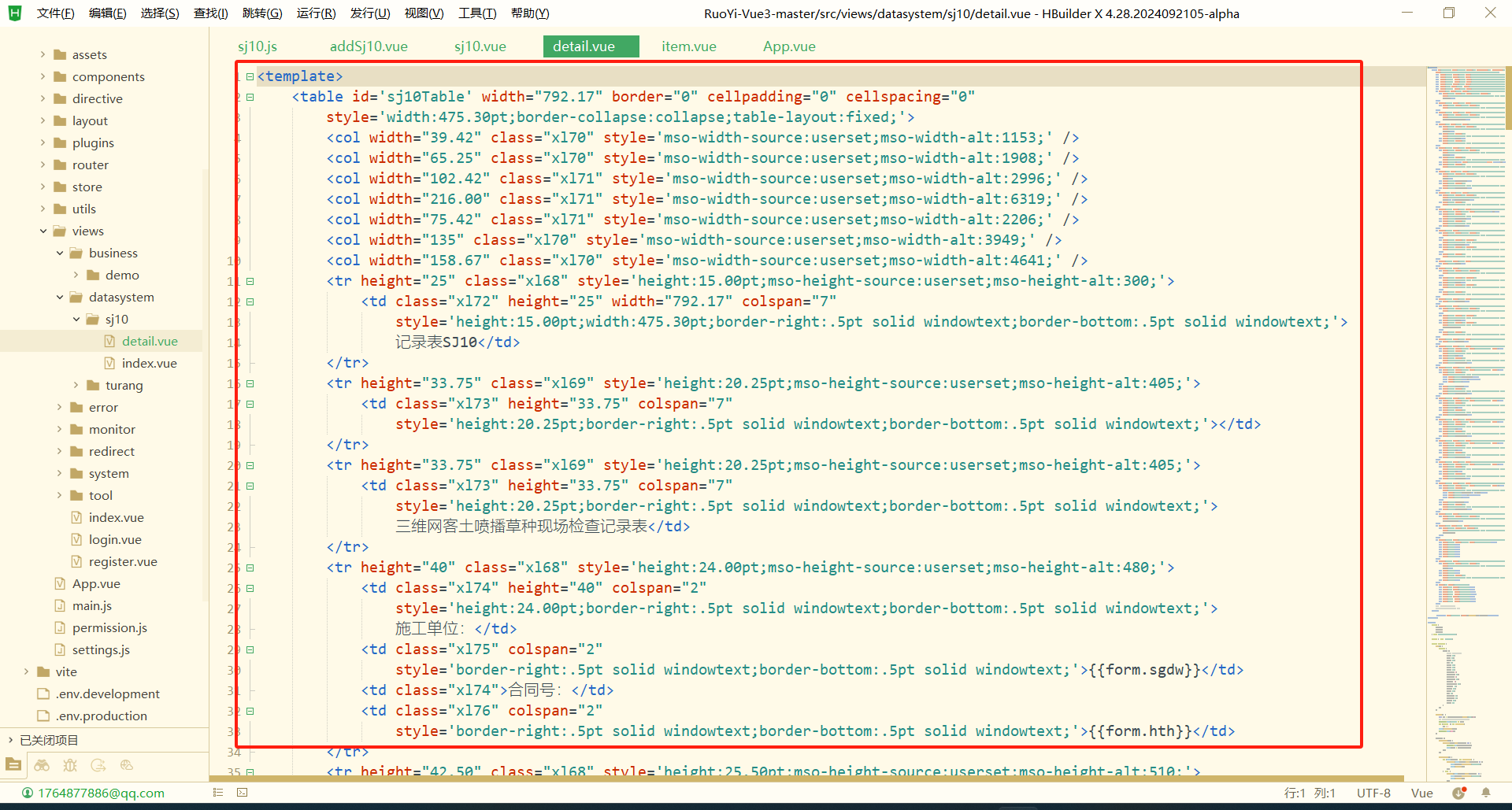
这里变动的点是sj10的index.vue文件中

**图2-4导入依赖文件**



**图2-5增加显示表格的文件以及相关触发事件的函数**

第二个困难的地方是在于表格的样式，这里解决办法是有比较取巧的。可以先打开Excel表格，另存为html，网页打开，然后右键框架源码，复制table那段，贴到vue里，就不用自己写代码搞表格了。



**图2-6表格excel的样式**

## **2.数据库的设计**

表2-1Sj10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 数据类型 | 长度 | 属性 |
| 1 | id | 主键 | int | / | PK |
| 2 | sgdw | 施工单位 | varchar | 64 | 非空 |
| 3 | hth | 合同号 | varchar | 64 | 非空 |
| 4 | jldw | 监理单位 | varchar | 64 | 非空 |
| 5 | bh | 编号 | varchar | 64 | 非空 |
| 6 | gcmc | 工程名称 | varchar | 64 | 非空 |
| 7 | sgsj | 施工时间 | date | / | 非空 |
| 8 | gcbw | 工程部位 | varchar | 64 | 非空 |
| 9 | wdcdsjz | 网长设计值 | float | 32 | 非空 |
| 10 | wdcdjyjg | 网长检验结果 | varchar | 64 | 非空 |
| 11 | wdkdsjz | 网宽设计值 | float | 32 | 非空 |
| 12 | wdkdjyjg | 网宽检验结果 | varchar | 64 | 非空 |
| 13 | psbthd | 铺设表土厚度设计值 | float | 32 | 非空 |
| 14 | psbthdjyjg | 铺设表土厚度检验结果 | varchar | 64 | 非空 |
| 15 | psbtph | 铺设表土的PH值 | varchar | 64 | 非空 |
| 16 | czpzl | 草籽喷种量 | varchar | 64 | 非空 |
| 17 | cchl | 草成活率 | varchar | 64 | 非空 |
| 18 | cg | 草高 | varchar | 64 | 非空 |
| 19 | whxtjcd | 网横向搭接长度 | varchar | 64 | 非空 |
| 20 | wzxdjcd | 网纵搭接长度 | varchar | 64 | 非空 |
| 21 | uhjxd | U型钉或J型钉 | varchar | 64 | 非空 |
| 22 | wgjc | 外观检查 | varchar | 64 | 非空 |

# 三、系统实现

## web界面显示



**图3-1登陆界面**



**图3-2 系统首页**



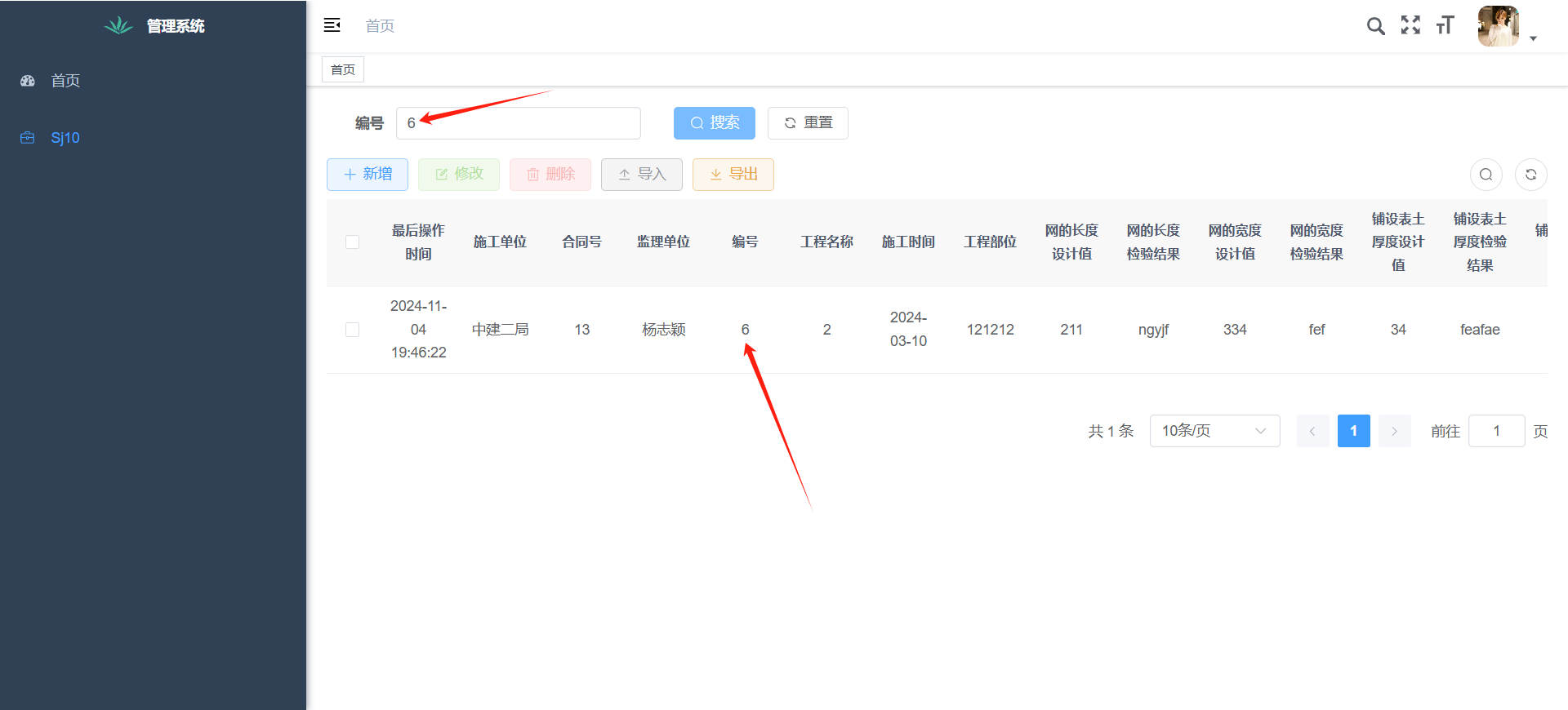
**图3-3 Sj10表格界面**



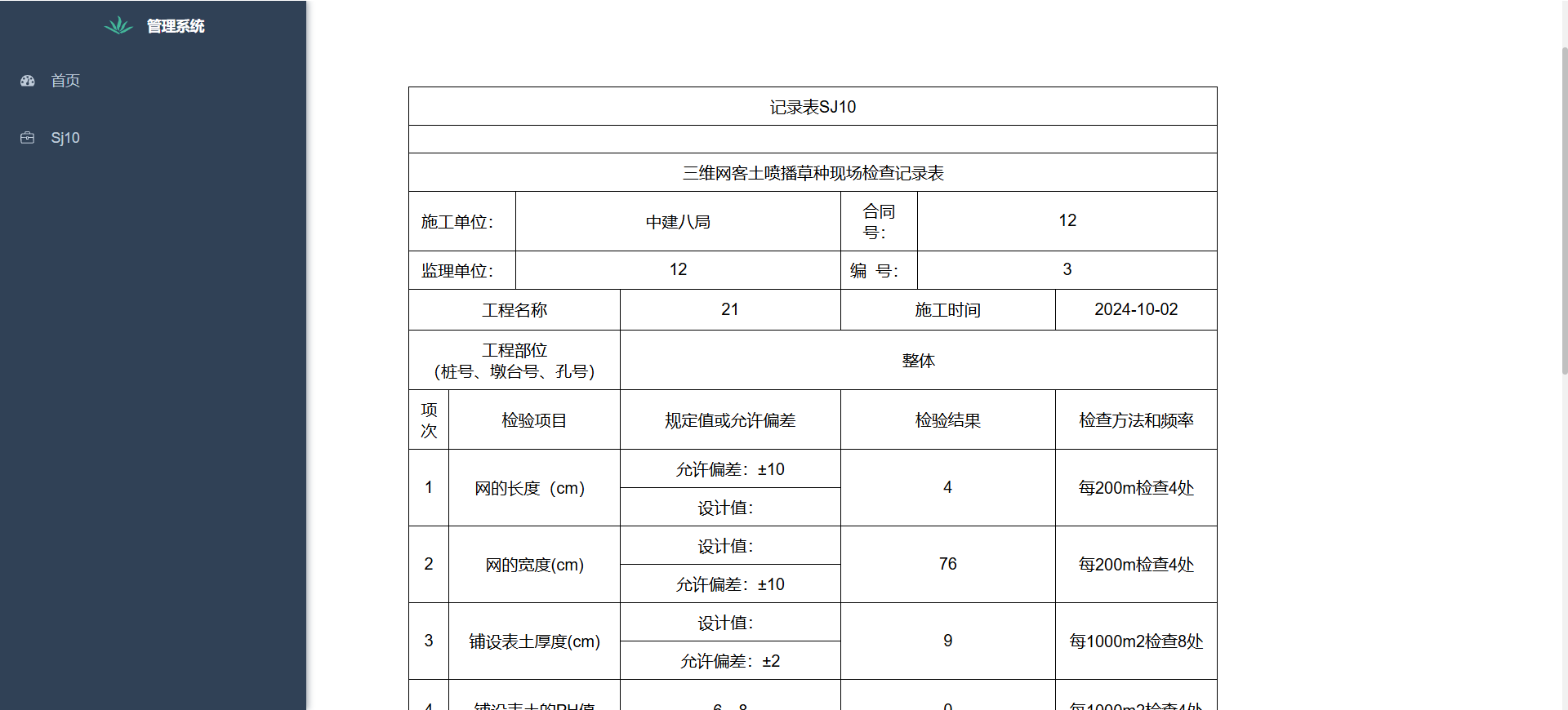
**图3-4 添加界面**



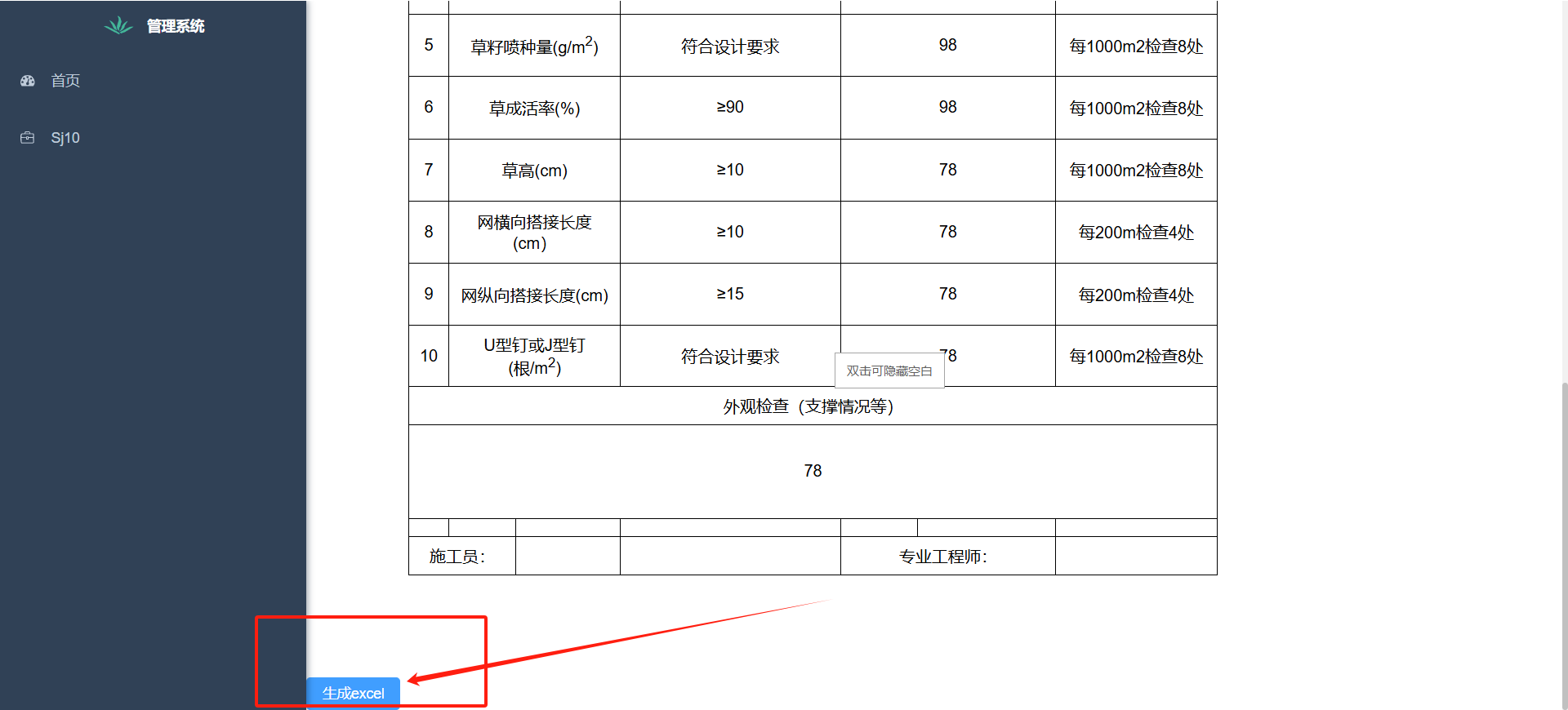
**图3-5 修改界面**



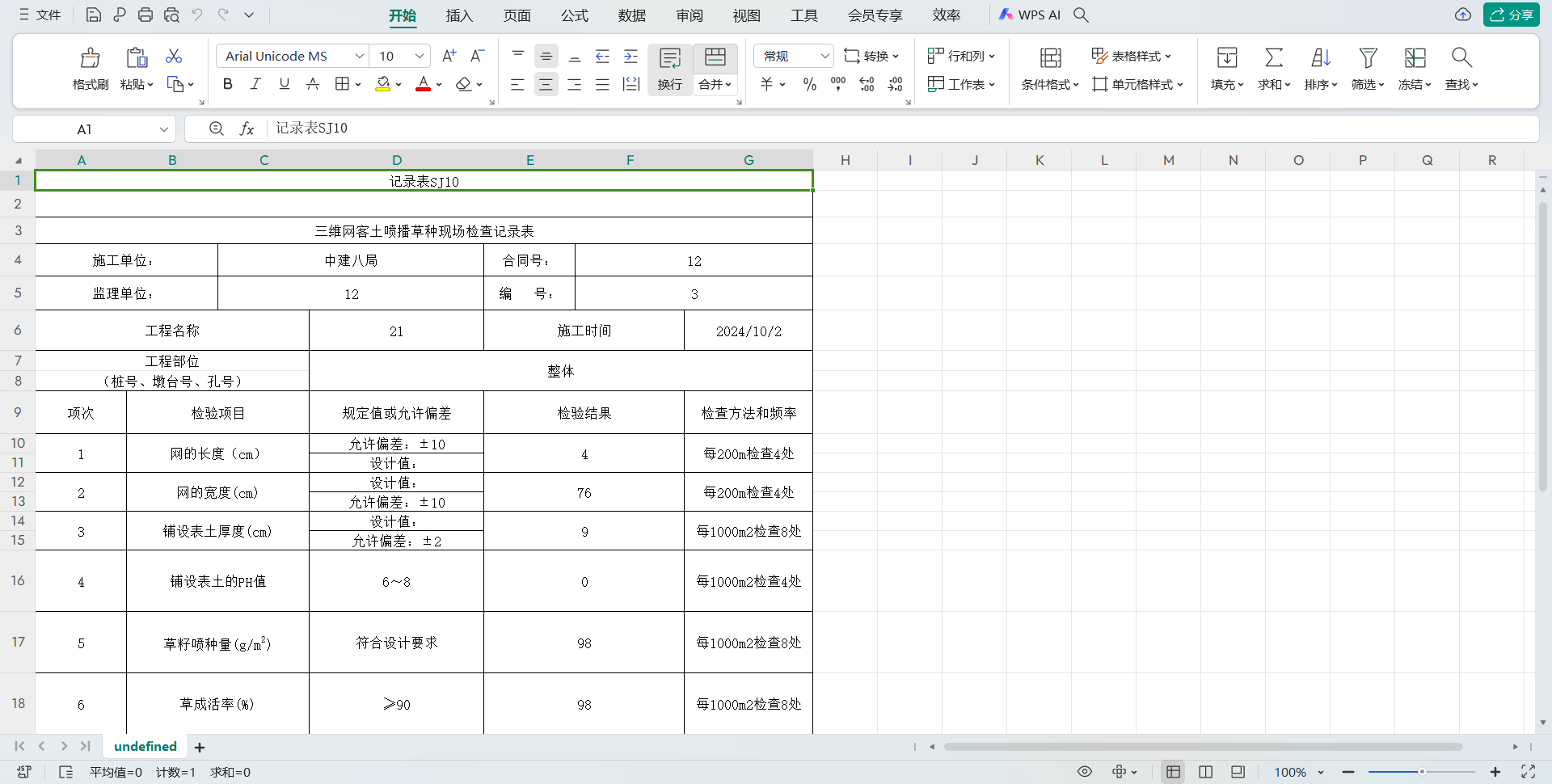
**图3-6 查询界面**



**图3-7 显示表格界面(由于表长，只显示上半部分)**

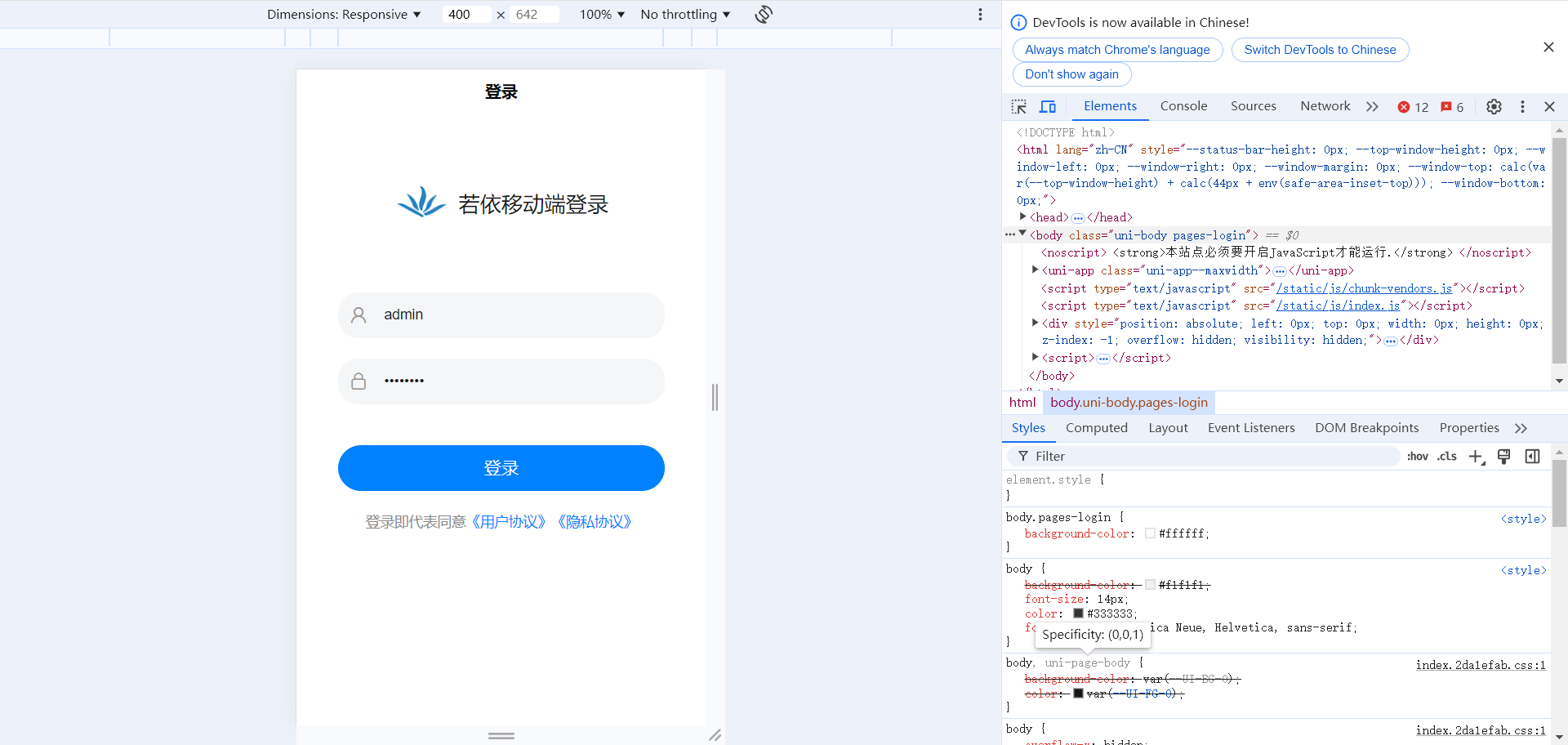


**图3-8 下载表格界面**

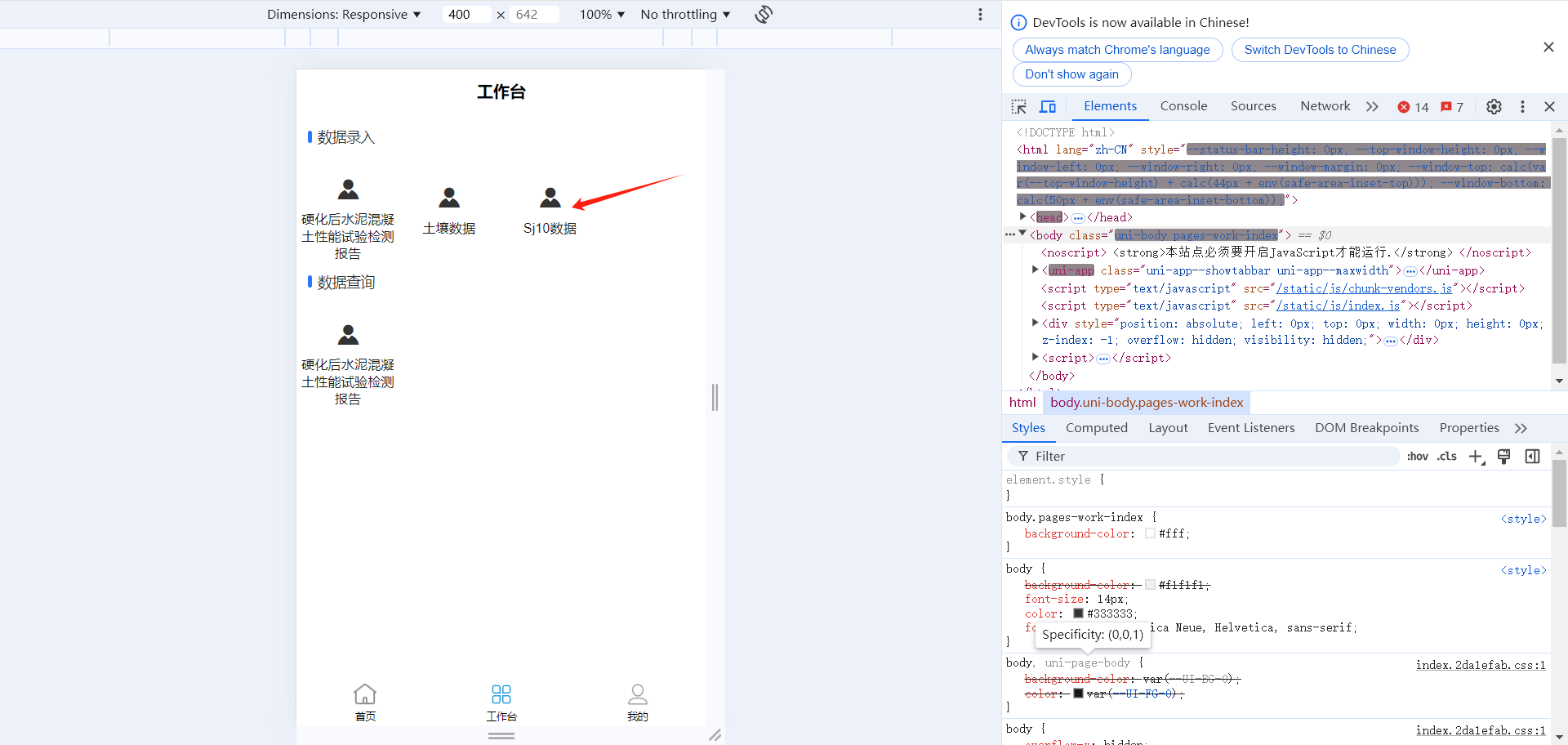


**图3-9 下载的表格**

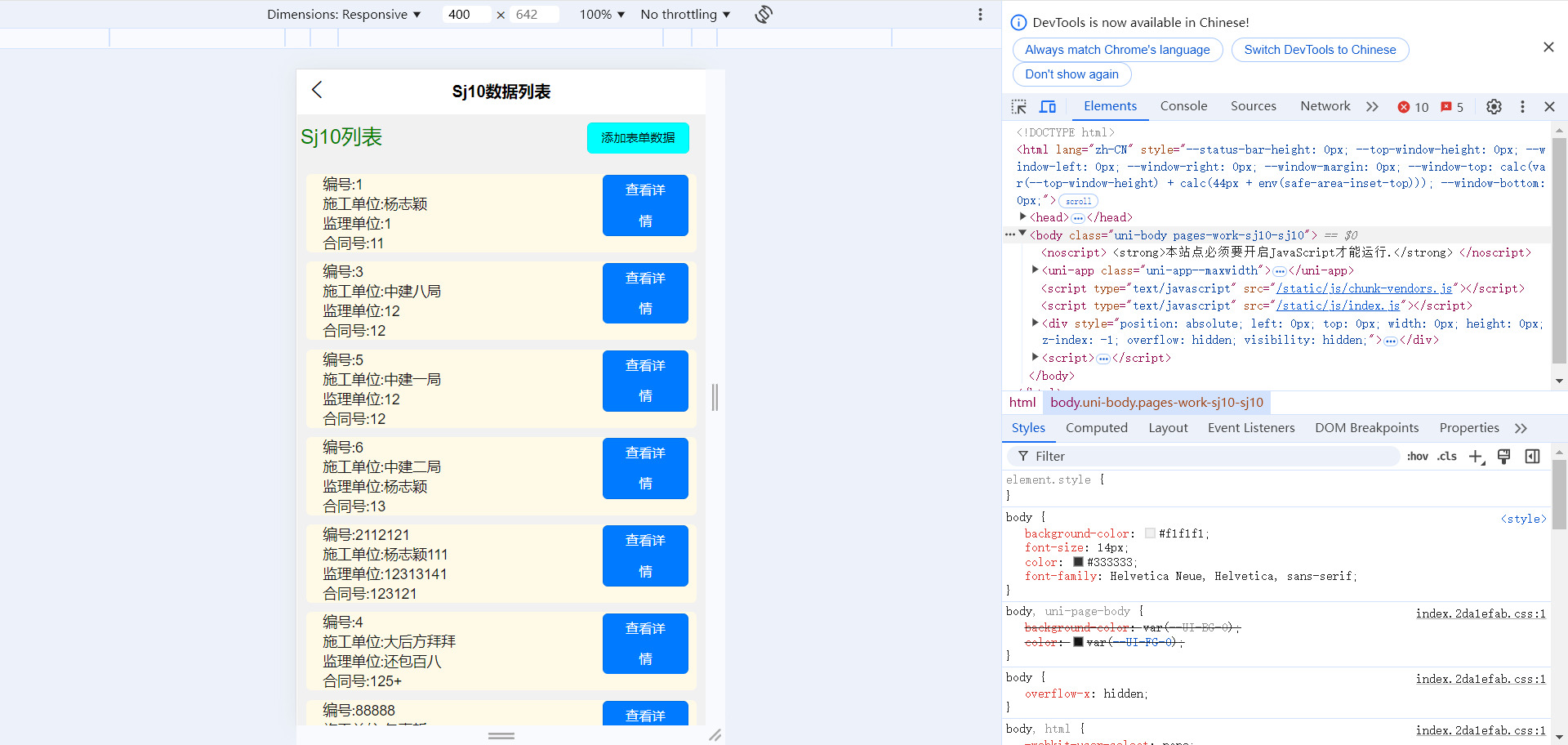
## App界面显示



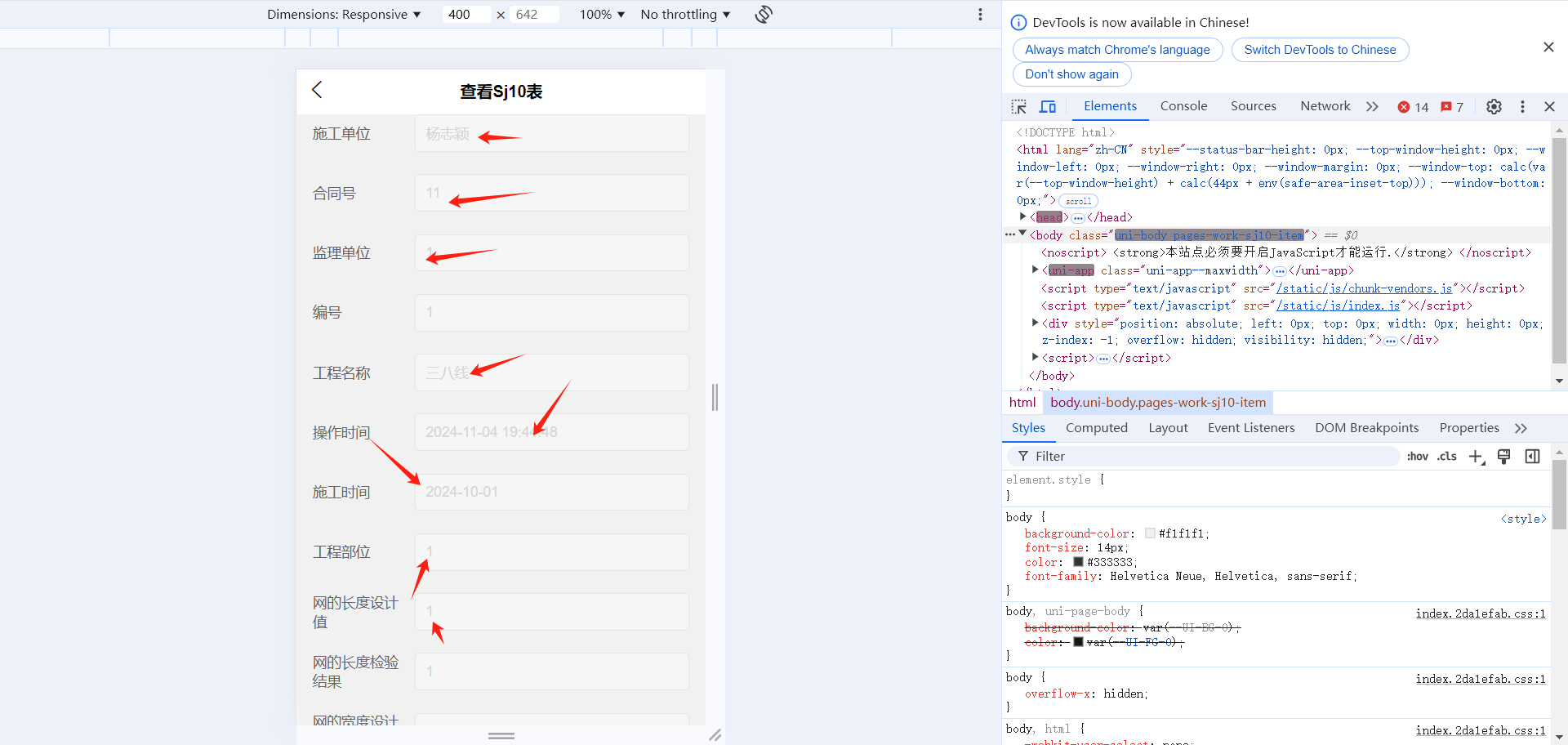
**图3-10 登陆界面**



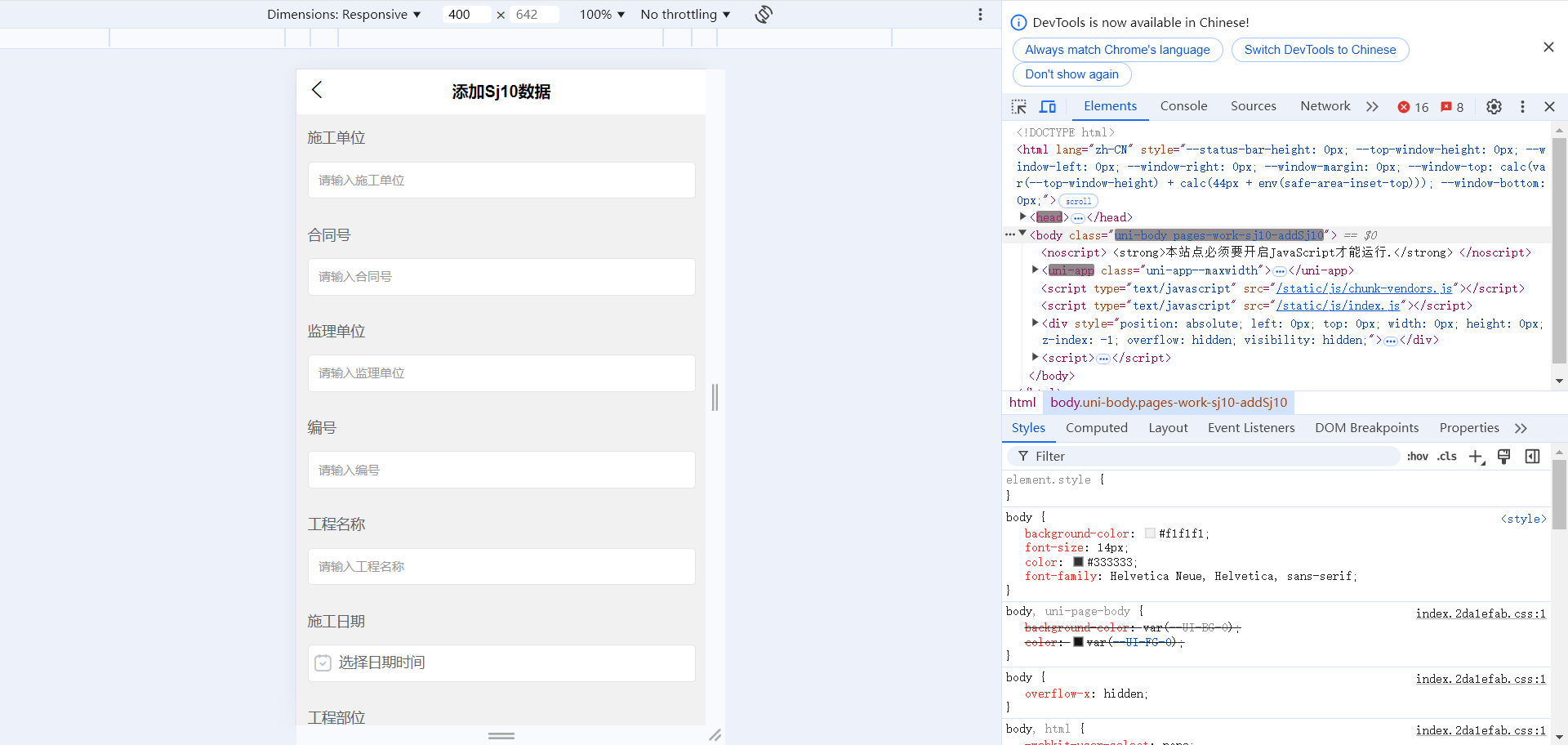
**图3-11 工作台界面**



**图3-12 数据列表**



**图3-13 查看数据**



**图3-14 添加数据**

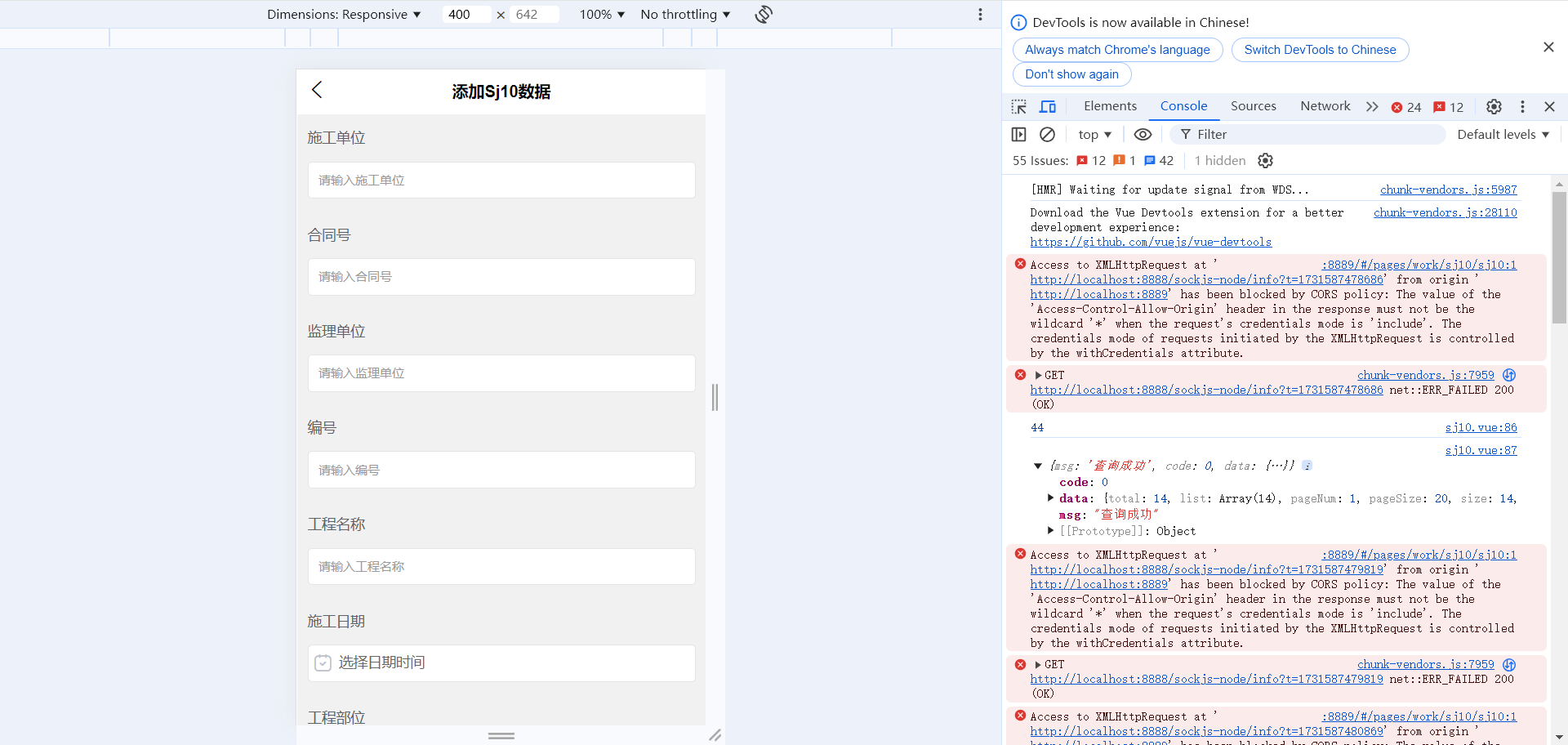
# 四、课设总结

## 心得、收获、体会

在这次实训中，我完整的体会到了前后端的开发是什么样的。在这个过程中我加深了对于IDEA的使用包括数据库，Maven工具的使用，以及怎么启动项目、创建项目。这里我比较大的收获是学会使用Maven和IDEA自带的数据库工具，同时知道了怎么连接数据库(包括怎么配置用户名、设置密码)、复习了Java以及数据库的相关语法，对于Java中的接口以及注解有了更加深刻的认识，知道了Redis是做什么的(用来存储账号密码进行校验)

对于前端和App来说，了解了Vue、js文件，知道了路由是什么，了解了相关的知识。比如在在 xxx.vue中 中，script 通常指的是 Vue 组件中的 <script> 标签，用于定义组件的逻辑部分。Vue 组件是由模板（template）、样式（style）和脚本（script）三个部分组成的。每个部分都有其特定的职责和用途。

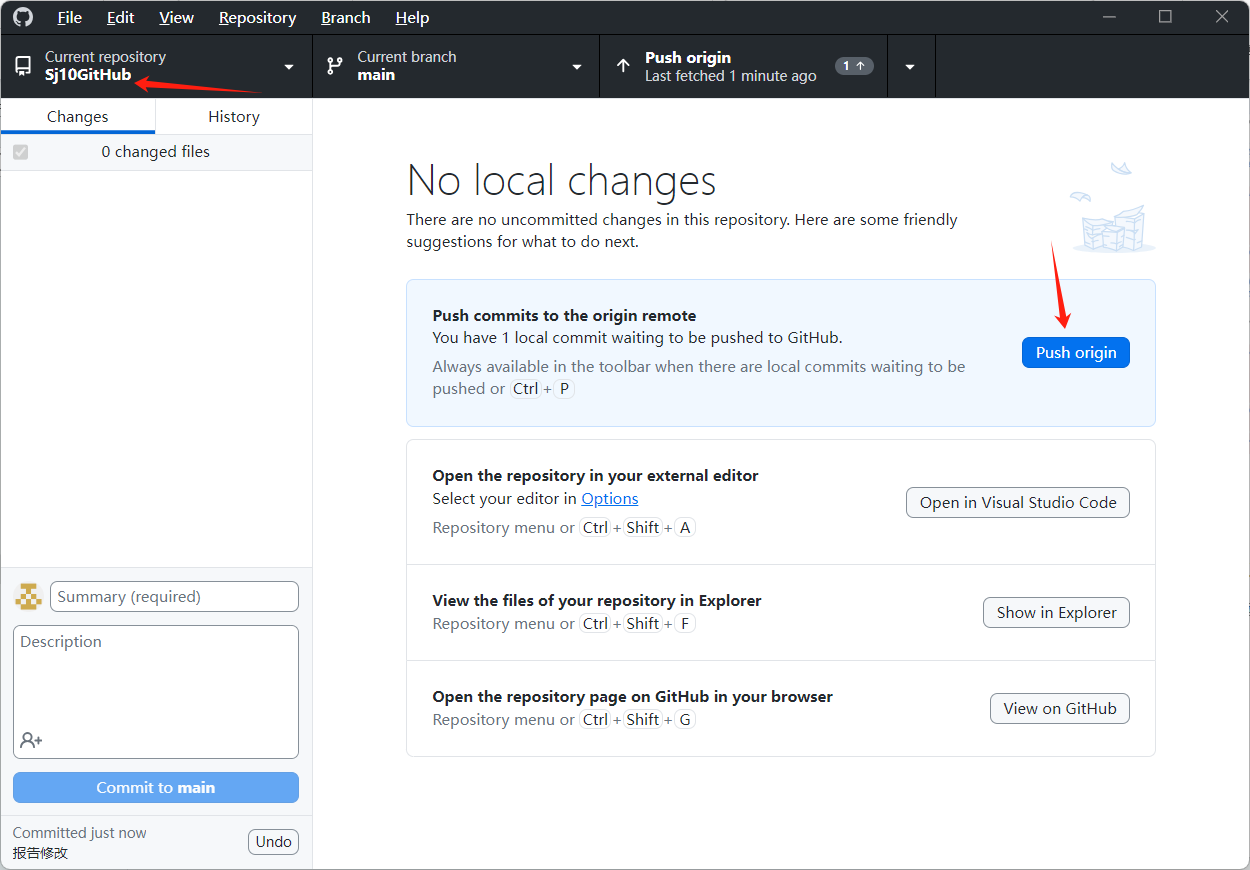
在查看错误调试方面，学会了使用浏览器的开发者模式来查看错误信息。



**图4-1 开发者模式**

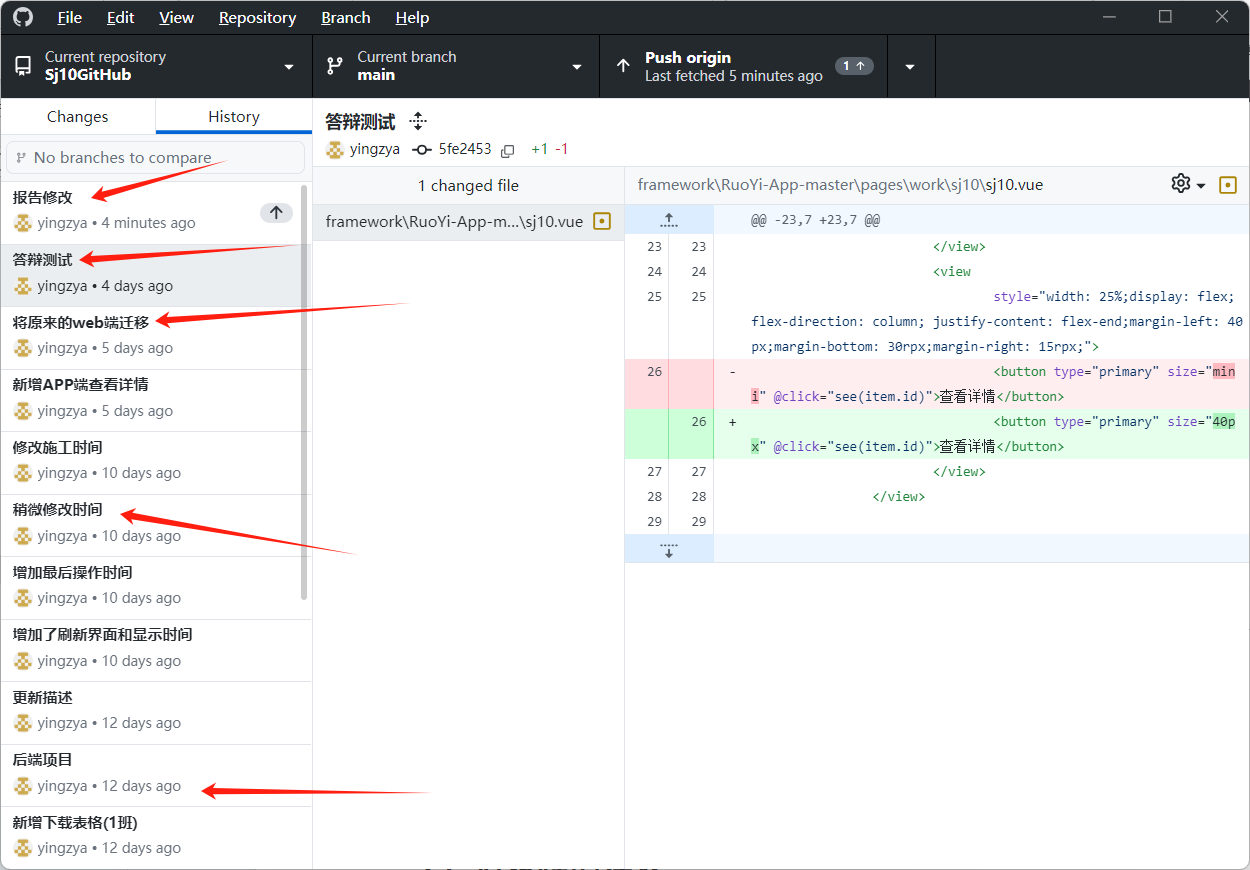
比如在该图中，我们可以看到信息查询的相关信息，以及出现的错误情况。

最后，我自己认为我学会的最大收获是会使用GitHub进行代码托管，以及在GitHub上学习优秀的开源项目。因为我在修改代码的时候有时候改完出现报错，但是忘记了之前怎么写的，于是我就去学了怎么托管代码。这样即使我修改了出现报错，还可以在GitHub上将原来的代码下载下来。这里之前我使用提交代码的方式是Git命令行的方式，之后我下载了GitHub的客户端，这极大的方便了我的代码托管以及从项目上拉取更新代码的操作。如下图

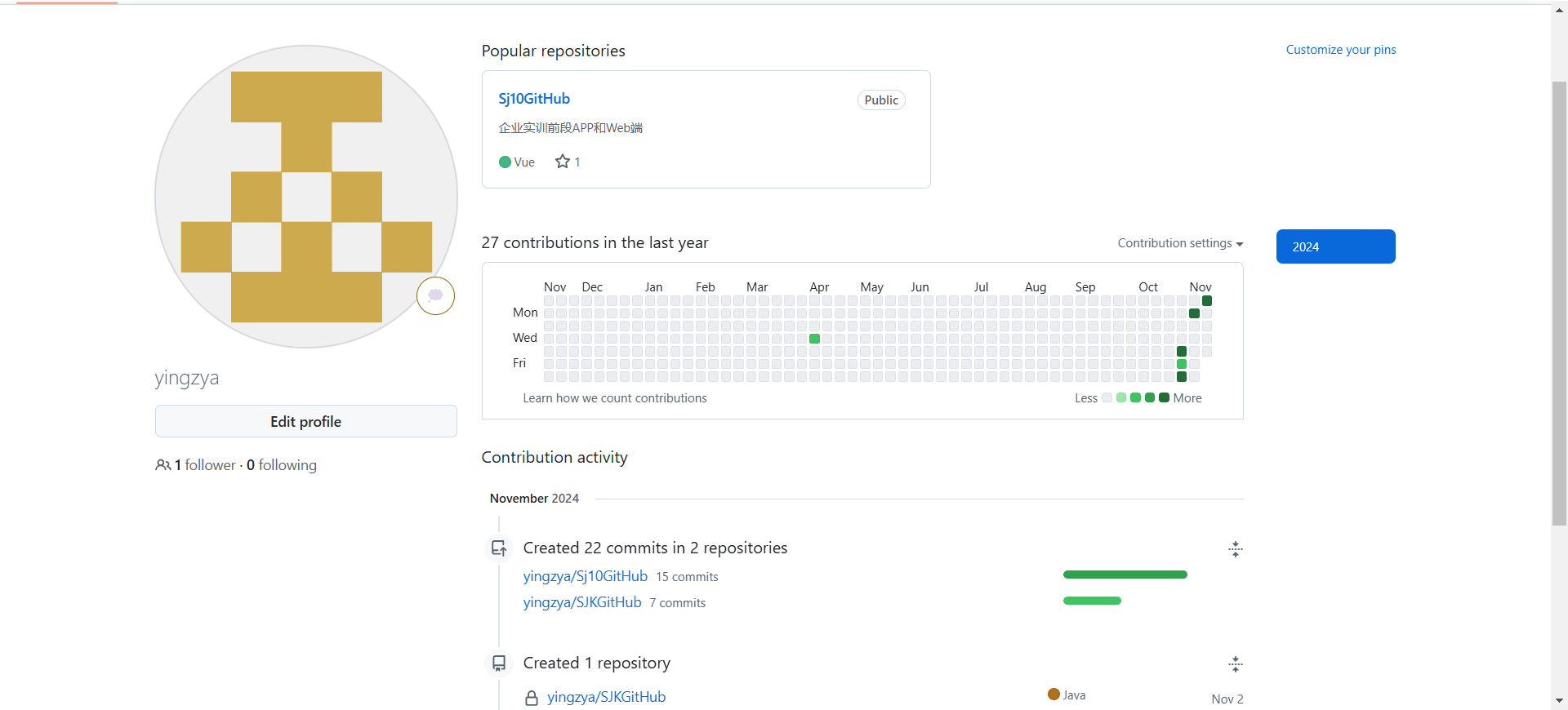


**图4-2 GitHub客户端**

比如我刚对代码进行了修改，只需在浏览器界面登陆上GitHub在客户端界面点击Push Origin即可。左上角是当前操作的远程仓库然后点击History可以查看代码的修改历史，如下图所示。



**图4-2 GitHub客户端代码修改记录**



**图4-3 我的GitHub资料**

## **对课程的建议**

我的建议是希望能在开学的前八周进行，并在寒暑假期间发布相关的前端知识学习任务以及配置相关的环境**(**IDEA、Navicat、Redis、MySQL、Vue脚手架等的安装，因为我们实训的时候安装这些就耗费了很多时间(大多同学不知道怎么配置)，而这些配置在实训时间去弄太浪费时间了，这些配置应该并不是我们学习开发的重点**)**，比如Vue、CSS、JavaScript等。提前学习知识，大体知道是啥意思，我感觉对于我们来说很重要。在实训的时候，我以及身边的朋友对于前端是第一次接触，老师写的代码我们是真看不懂，也不知道怎么改，就只能求助ai，经常因为一个问题而耽误很久。

前八周是因为刚开学课程轻松，大家都没什么事情，这个学期安排在9、10周，大多同学都有段考、考试以及课程设计、导师制课程等相关任务，导致时间非常紧张。

然后就是希望可以要求同学们对于项目代码的管理能够通过远程代码托管，我在实训的时候发现很多同学就因为改错代码不会改回去浪费很多时间，但是如果提前在托管平台(GitHub、GitCode、Gitee)有代码备份则不用担心这些问题。