

**生产实习**

**课程设计报告**

**课设题目：BUCTCODER 系统开发软件设计**

**方案**

**班 级： 计科2104**

**姓 名：**

**学 号：**

**校内导师： 刘勇**

**企业导师：**

2024年7月

目录

摘 要

BUCTCODER 系统开发软件设计方案的课程设计内容主要包括以下几个方

面。开发环境搭建：安装 Java、Vue、MySQL 等必要的开发环境，并配置好相

应的开发工具（如 IDE、前端构建工具等）。技术文档和讲解：提供详细的技术

文档和讲解，包括后端开发技术（Java Spring Boot、MySQL、MyBatis、SwaggerUI、

Knife4j）和前端开发技术（Vue、ElementUI、HTML/JS/CSS、Echarts）。文档

涵盖了技术的基本概念、开发流程、常用操作等内容，帮助我们快速了解和上手

开发所需的技术。基础代码框架：提供基础的代码框架，包括后端和前端的代码

结构、配置文件、常用工具类等。这些代码框架可以作为开发的起点，我们可以

在其基础上进行功能开发和优化。分组开发任务：每个开发小组由约 3 个同学组

成，通过 svn/git 管理源代码。根据课程设计要求，给出具体的系统开发任务，

要求学生在规定的时间内完成相应的功能模块或系统功能开发。系统测试与调试：

开发完成后，进行系统测试和调试，确保系统的功能完善和稳定性。需要按照测

试方案进行测试，排查和修复可能存在的问题。

通过 BUCTCODER 系统开发软件设计方案的课程设计，我们将学习到企业

级系统开发最新、广泛使用的前后端分离开发技术栈。掌握这些技术可以帮助学

生更好地适应互联网行业的需求，提高系统开发的能力和就业竞争力。同时，通

过实际的开发任务和系统测试，能够锻炼问题解决能力、团队协作能力和系统调

试能力。这些综合能力的提升将对我们今后的学习和工作产生积极的影响。

1. 项目背景及意义
   1. 项目背景

随着信息技术的快速发展和互联网的普及，企业对于高效、稳定、可扩展的

系统需求日益增长。BUCTCODER 系统开发软件设计方案的课程设计应运而生，

旨在培养学生在企业级系统开发中所需的技术和能力。

随着前后端分离开发模式的兴起，该方案特别关注前后端分离技术栈的应用

和实践。这种开发模式将前端和后端的开发过程分离，使得团队能够更加专注于

各自的领域，提高开发效率和系统的可维护性。

本项目选择使用 Java Spring Boot 作为后端开发框架，并结合 MySQL 和

MyBatis 进行数据处理和持久化。这些技术在企业级系统开发中被广泛使用，具

有良好的稳定性和可扩展性。

同时，为了实现优秀的用户界面和数据可视化效果，项目采用了 VUE 作为

前端框架，并结合 ElementUI 和 Echarts 进行页面布局、组件设计以及数据可视

化操作。这些前端开发技术具有良好的用户体验和可视化效果，能够满足用户对

于系统界面的高要求。

通过该课程设计，学生将通过实践学习和团队协作，掌握企业级系统开发的

最新技术栈。他们将学习前后端开发技术、代码管理工具和系统测试与调试等实

际应用，提升自己的问题解决能力和团队合作能力。这将为他们今后在互联网行

业的就业和职业发展奠定坚实基础。

* 1. 本课题主要研究内容

本课题的主要研究内容是 BUCTCODER 系统的开发软件设计方案。为了实

现该系统的开发，研究内容将涵盖以下几个方面：

1.系统需求分析：对 BUCTCODER 系统进行全面的需求分析，包括功能需求、

性能需求、安全需求等。通过与相关领域的专家和用户的沟通和参与，明确系统

的核心目标和使用场景，为后续的系统设计和开发提供指导。

2.系统设计：在需求分析的基础上，进行系统的整体架构设计。这包括前端

界面设计、后端业务逻辑设计、数据库设计等。通过合理的架构设计，保证系统

具有良好的可扩展性、可维护性和高性能。

3.前端开发：采用 VUE 作为前端开发框架，结合 ElementUI 进行页面布局和

组件设计。同时，使用 HTML、JS 和 CSS 等前端技术实现系统的交互效果，并

利用 Echarts 进行数据可视化展示。

4.后端开发：使用 Java Spring Boot 框架进行后端开发。借助 Spring Boot 的

快速开发特性和丰富的生态系统，实现系统的核心业务逻辑以及与数据库的数据

交互。使用 MySQL 作为数据库存储系统，并结合 MyBatis 进行数据的持久化操

作。

5.接口文档和测试：利用 SwaggerUI 和 Knife4j 等工具生成系统的 API 文档，

方便接口的管理和调试。同时，进行系统的单元测试、集成测试和性能测试，确

保系统的功能正确性和稳定性。

6.项目管理和团队合作：通过使用 git 等版本控制工具进行源代码管理，

并采用敏捷开发方法，进行项目管理和团队协作。确保项目的进度控制、任务分

配和沟通协作的有效性。

综上所述，本课题的主要研究内容将涵盖 BUCTCODER 系统的需求分析、

系统设计与开发、接口文档和测试、以及项目管理和团队合作等方面。通过对这

些内容的研究和实践，旨在培养学生在企业级系统开发中所需的技术和能力，并

提高他们的问题解决能力和团队协作能力。

1. 项目实施方法设计
   1. 技术架构

前端开发：

（1）使用 Vue.js 框架进行前端开发，利用其组件化和响应式的特性。

（2）使用 Vue Router 进行页面路由管理，创建不同的页面和组件。

（3）使用 Vue CLI 进行项目脚手架搭建和开发环境配置。

（4）使用 Axios 库进行前端与后端的数据交互，发送异步请求获取数据。

后端开发：

（1）使用 Node.js 的 Express 框架。

（2）创建后端 API 接口，用于前端与后端数据的交互和处理。

（3）通过后端接口与数据库进行数据的增删改查操作。

数据库：

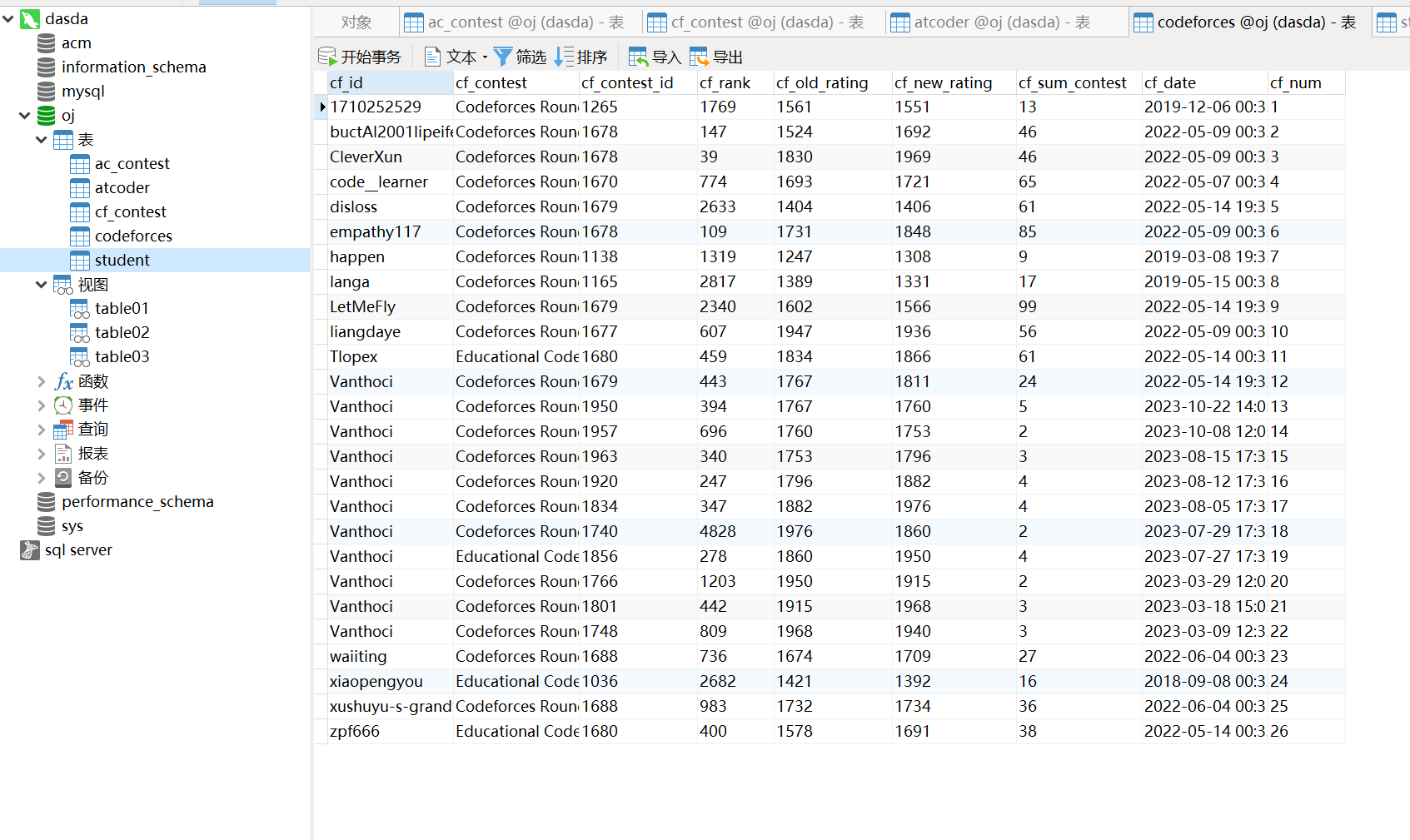
（1）根据需求选择合适的MySQL数据库。

（2）创建学生信息的数据表，包括学生基本信息和解题信息等，建立合适的

关系。

* 1. 数据库设计

数据库包含五个表和三个视图，表格有ac\_contest（ac比赛信息）、atcoder（ac学生做题数据）、cf\_contest（cf比赛信息）、codeforces（cf学生做题数据）、student（学生信息表）；视图有table01（学生做题数量表）、table02（学生新旧积分对比表）、table03（各班学生积分总和对比表）。



* 1. 系统用例图
  2. 难点功能技术分析与实现
  3. 系统架构

软件采用三层架构设计包括：数据层、业务层和应用层，如图3.1-1所示。

1）数据层

数据层存储所需的相关数据，由爬虫模块获取。包括：学生基本信息，学生AtCoder和Codeforces做题数据。学生基本信息存储用户的相关信息如学号、名字、班级名称、atoder账号、codeforces账号等信息；Atcoder数据包含id、比赛名称、日期、积分变化、最高积分、参与比赛次数，新积分、排名、表现等信息；Codeforces包含id、比赛时间、比赛排名、最新积分、参与比赛次数等信息。

2）业务层

业务层调用数据层的相关数据，Java后端系统调用爬虫所获取到的数据，为前端提供数据接口。

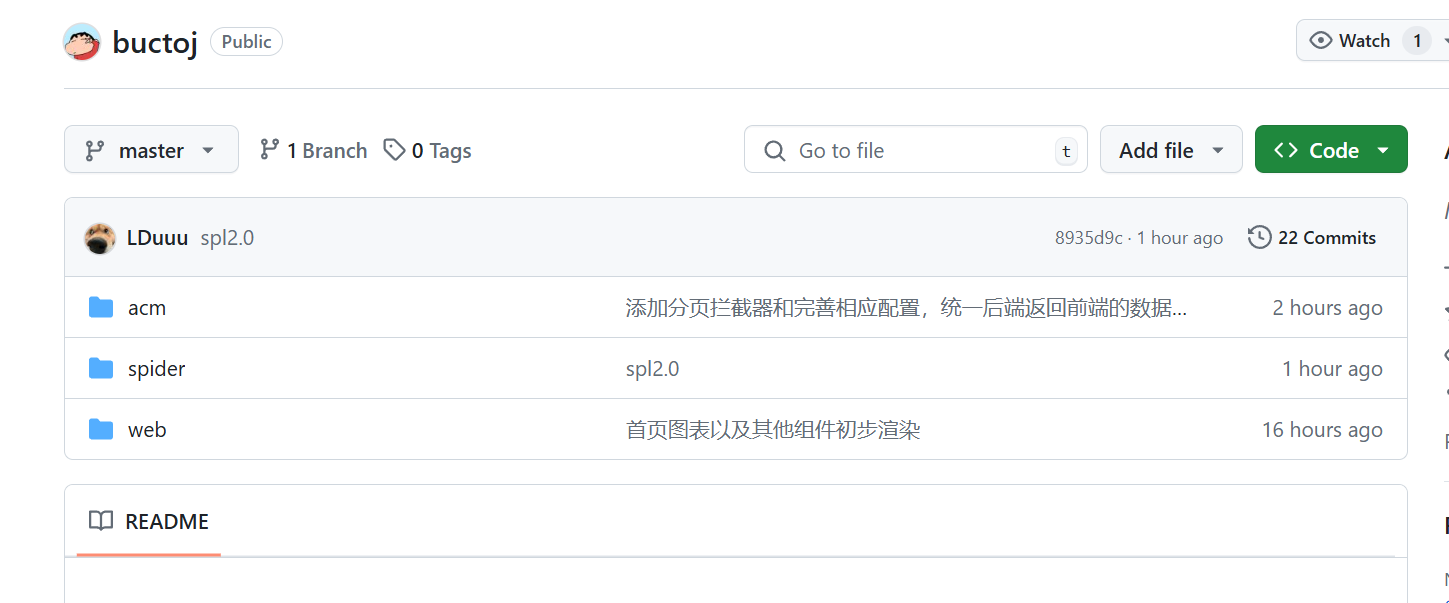
3）应用层

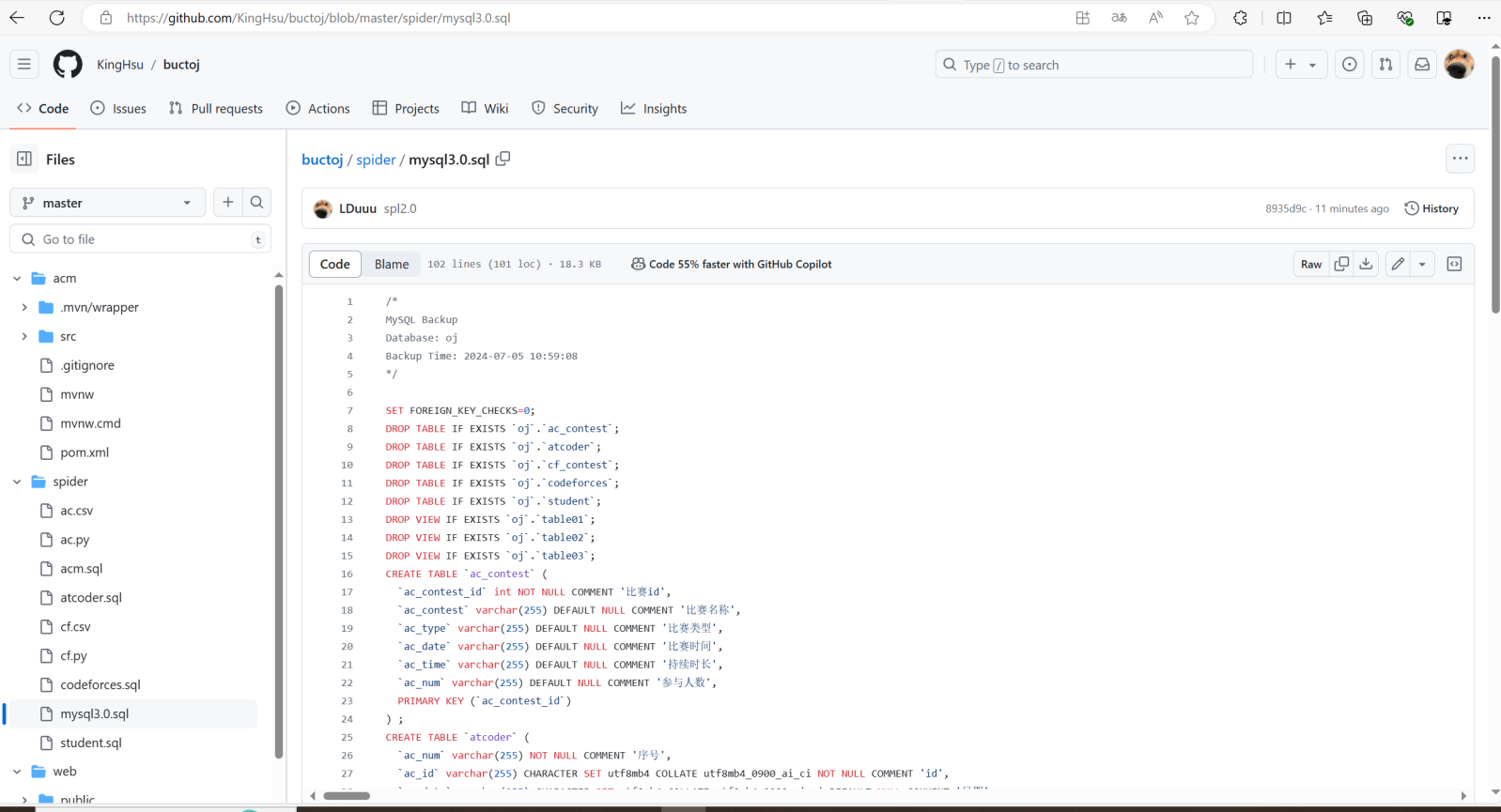
应用层与用户进行直接交互，用于展示学生刷题数据。应用层包括：以表格以及各种图表形式分析和展示学生做题数据。

1. 项目实现与测试

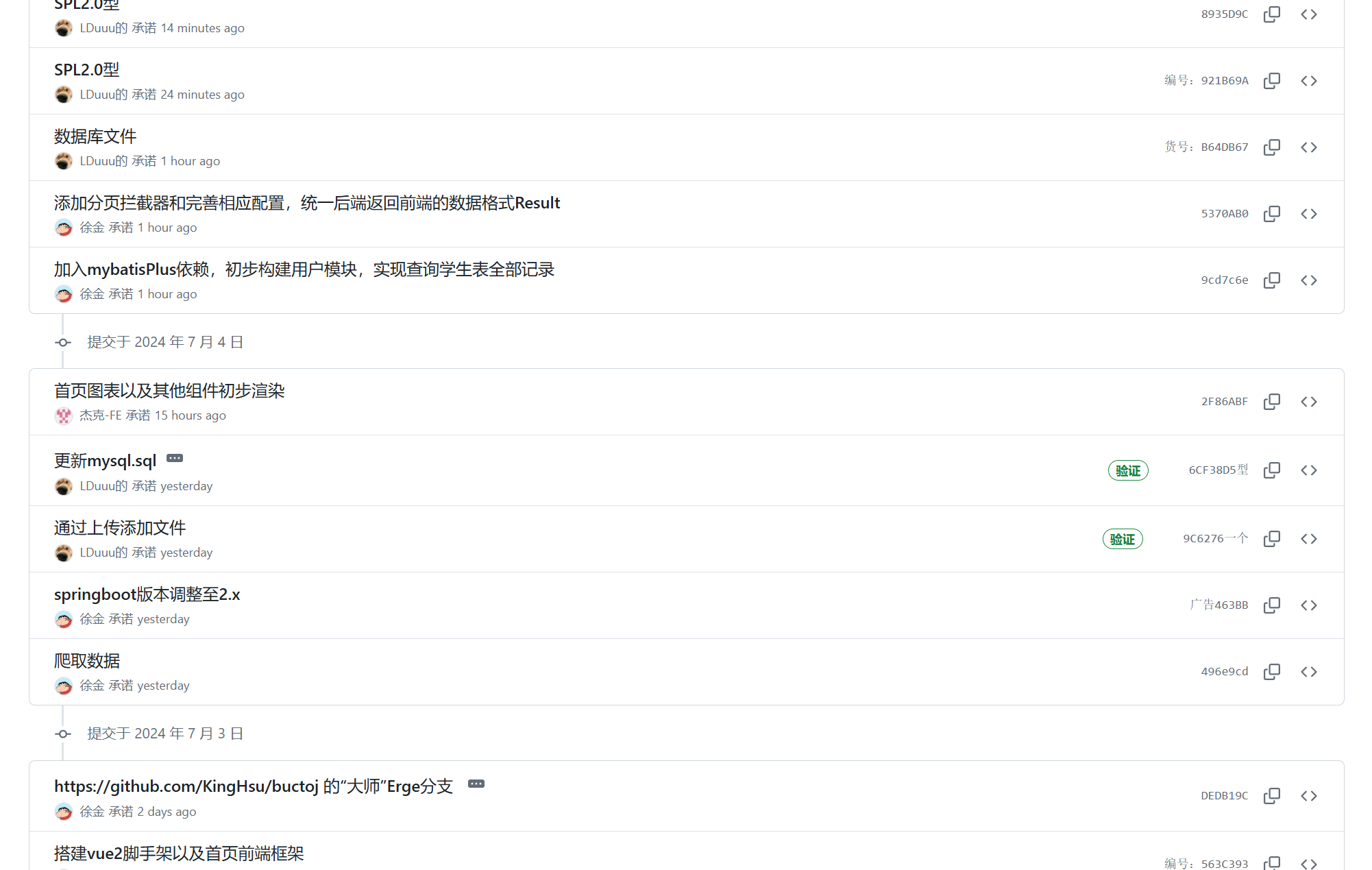
3.1 项目源代码分析

结合Git代码管理工具提供的数据和图表进行分析，在github上创建仓库[KingHsu/buctoj (github.com)](https://github.com/KingHsu/buctoj)，分三个文件夹：前端（web）、爬虫及数据库（spider）、后端（acm）提交代码。





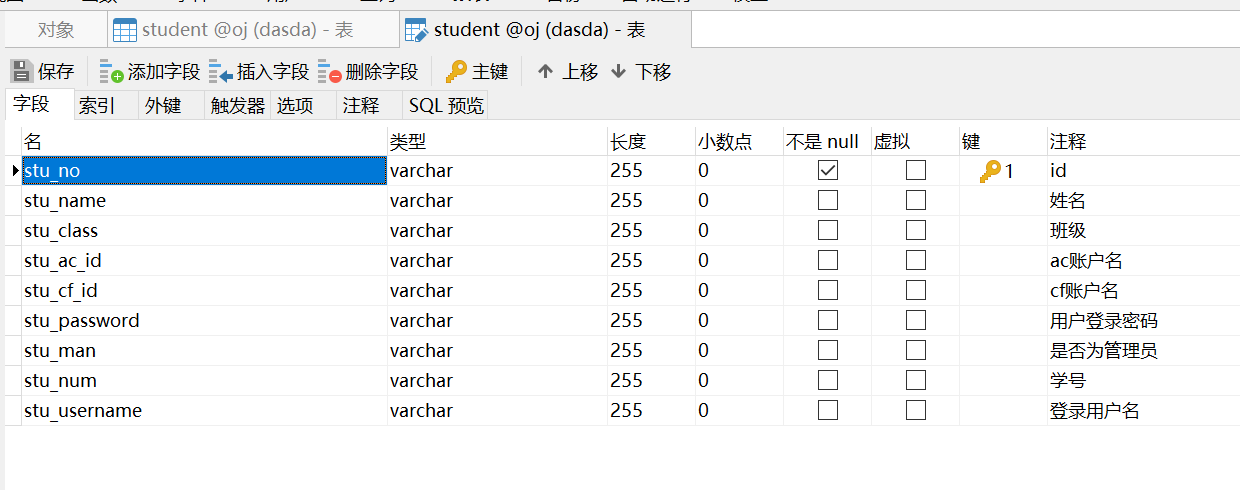
在github提交日志中可以看到，组员在不同时间和日期进行一系列的修改和删除操作。通过这些操作可以了解到团队成员在不同时间点对代码进行的不同的改动。这些改动不断完善这我们的项目，丰富项目功能。

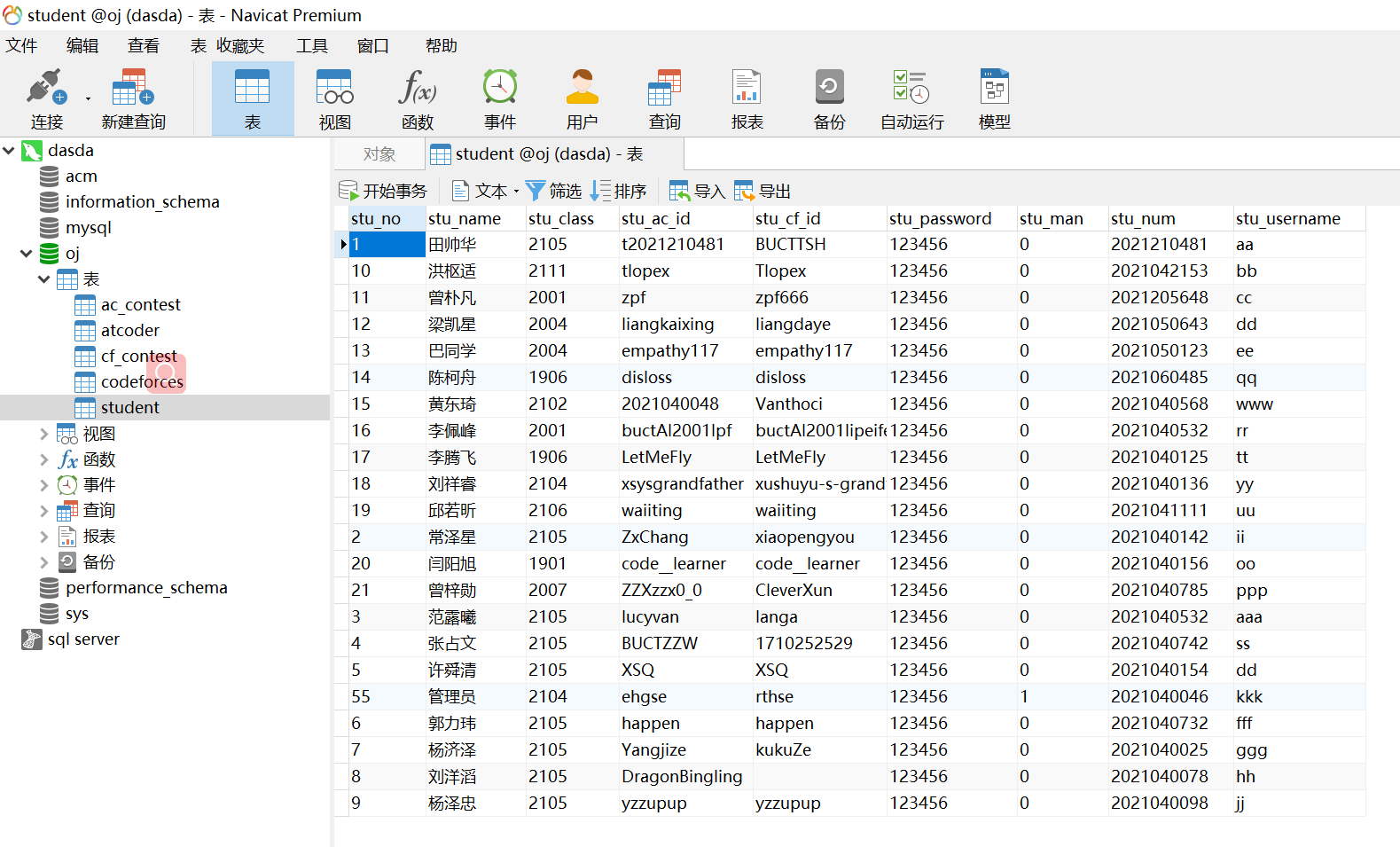


3.2 项目实现效果

3.2.1 数据库实现效果

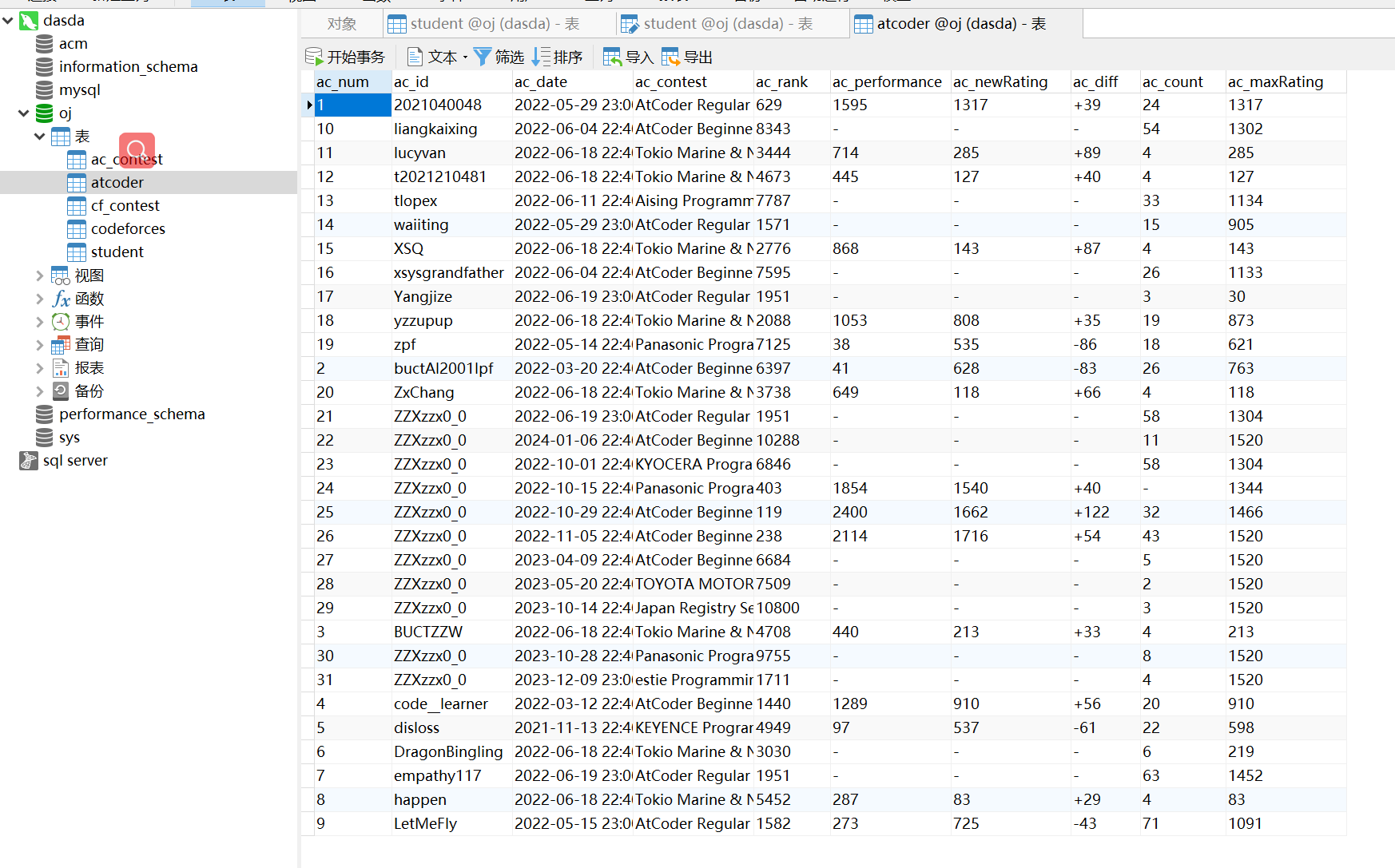
（1）student表：学生基本信息



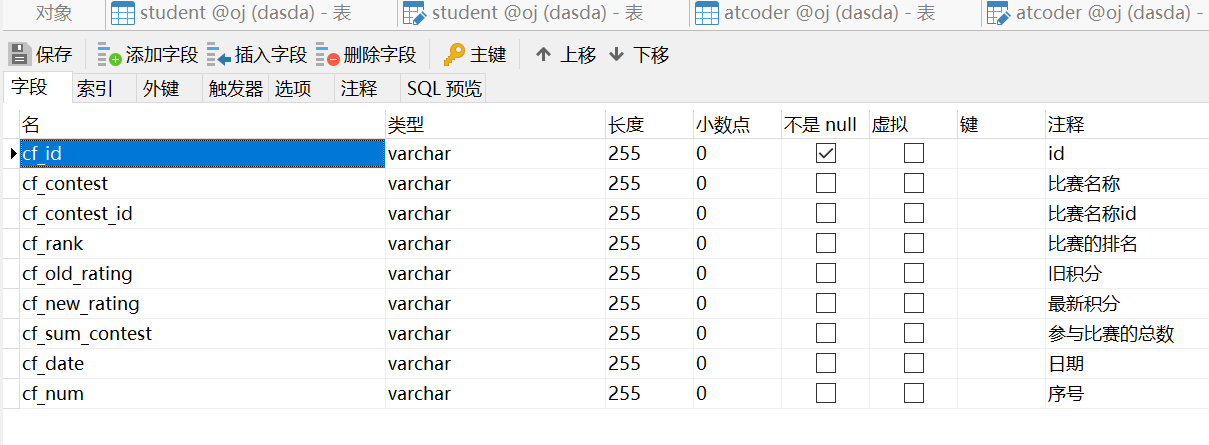


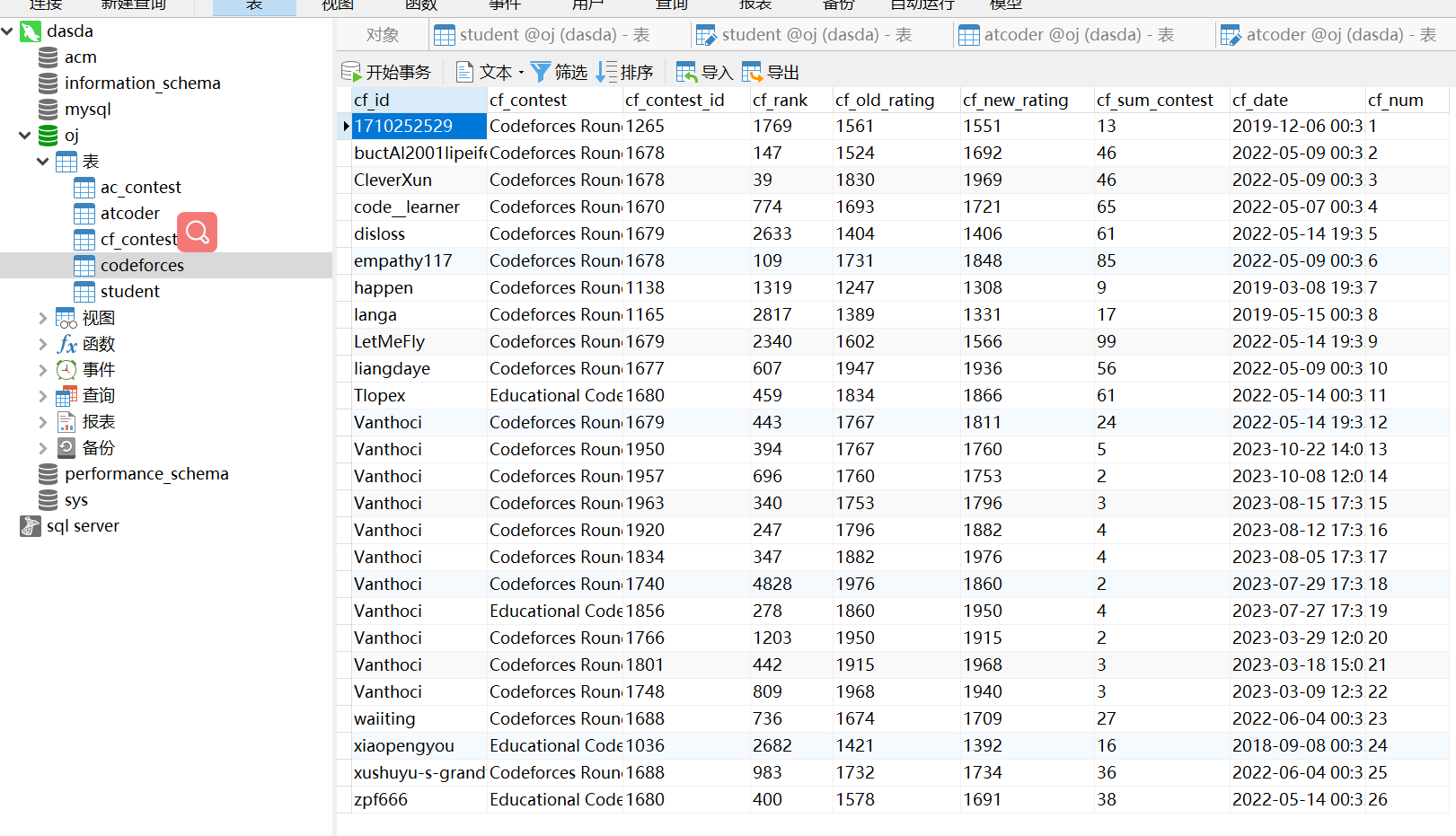
1. atcoder表（ac做题数据）





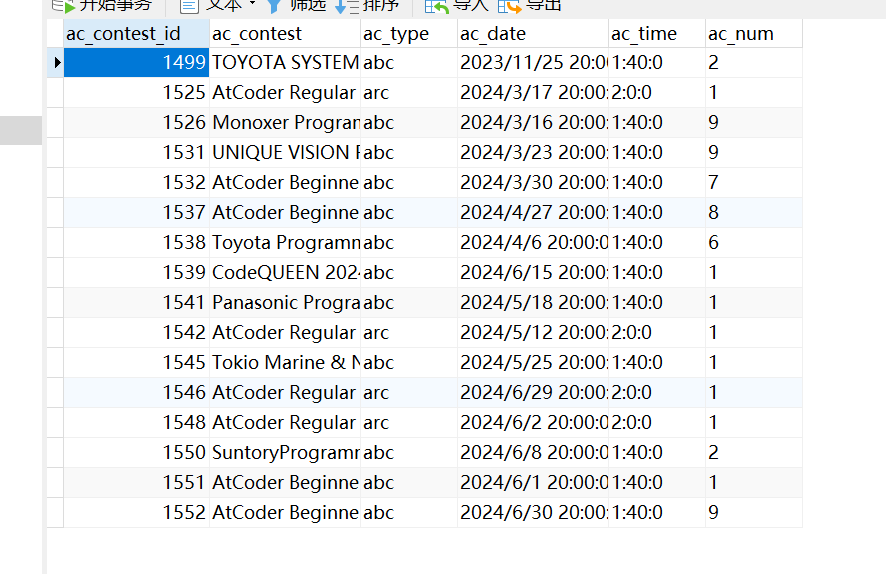
1. codeforces表：cf平台上的学生做题数据





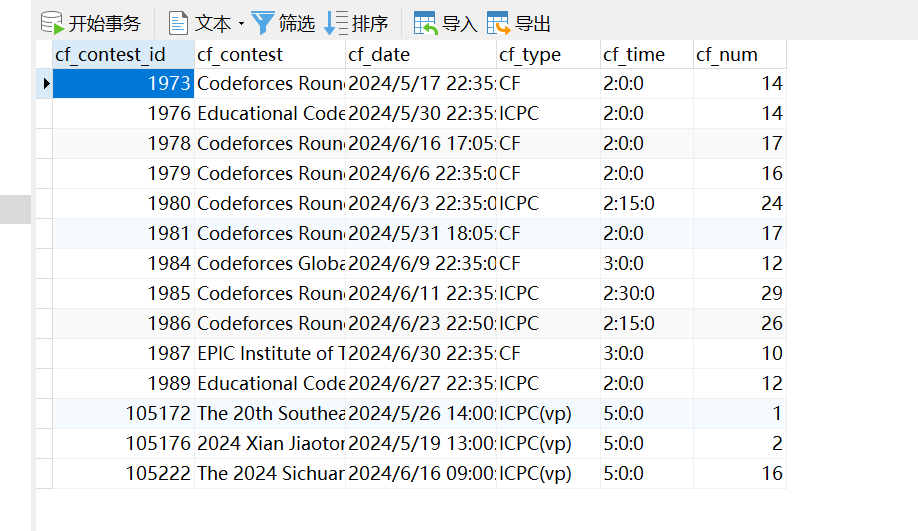
1. ac\_contest表：ac网站比赛信息



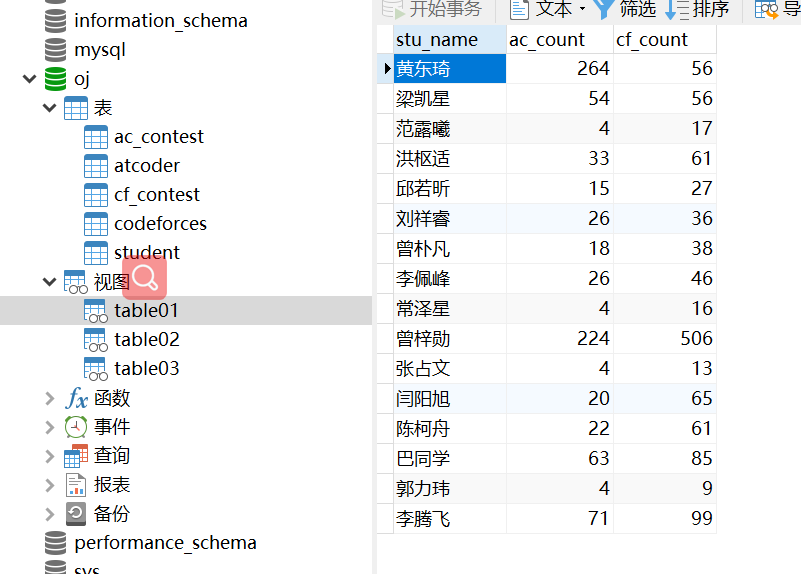


1. cf\_contest表：cf网站比赛信息

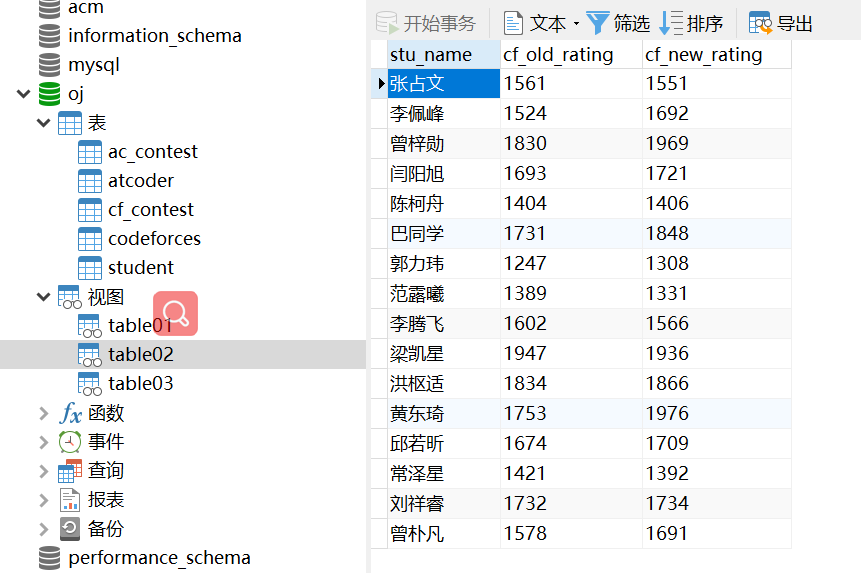




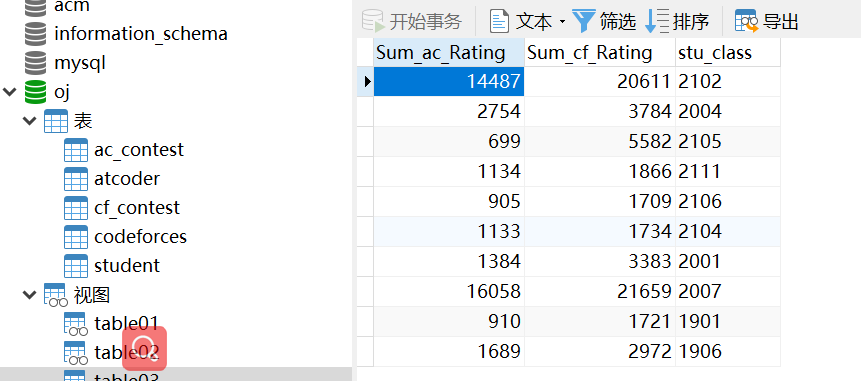
1. Table01表：学生做题数量表



1. Table02表：学生新旧积分对比表



1. Table03表：各班学生积分总和对比表



3.2.2 主页实现效果

（1）主页展示效果：分四个模块：学生做题数量占比图、已结束的比赛信息、codeforces学生积分排行榜、atcoder学生积分排行榜



3.2.3 用户列表实现效果

3.2.4 codeforce比赛列表实现效果

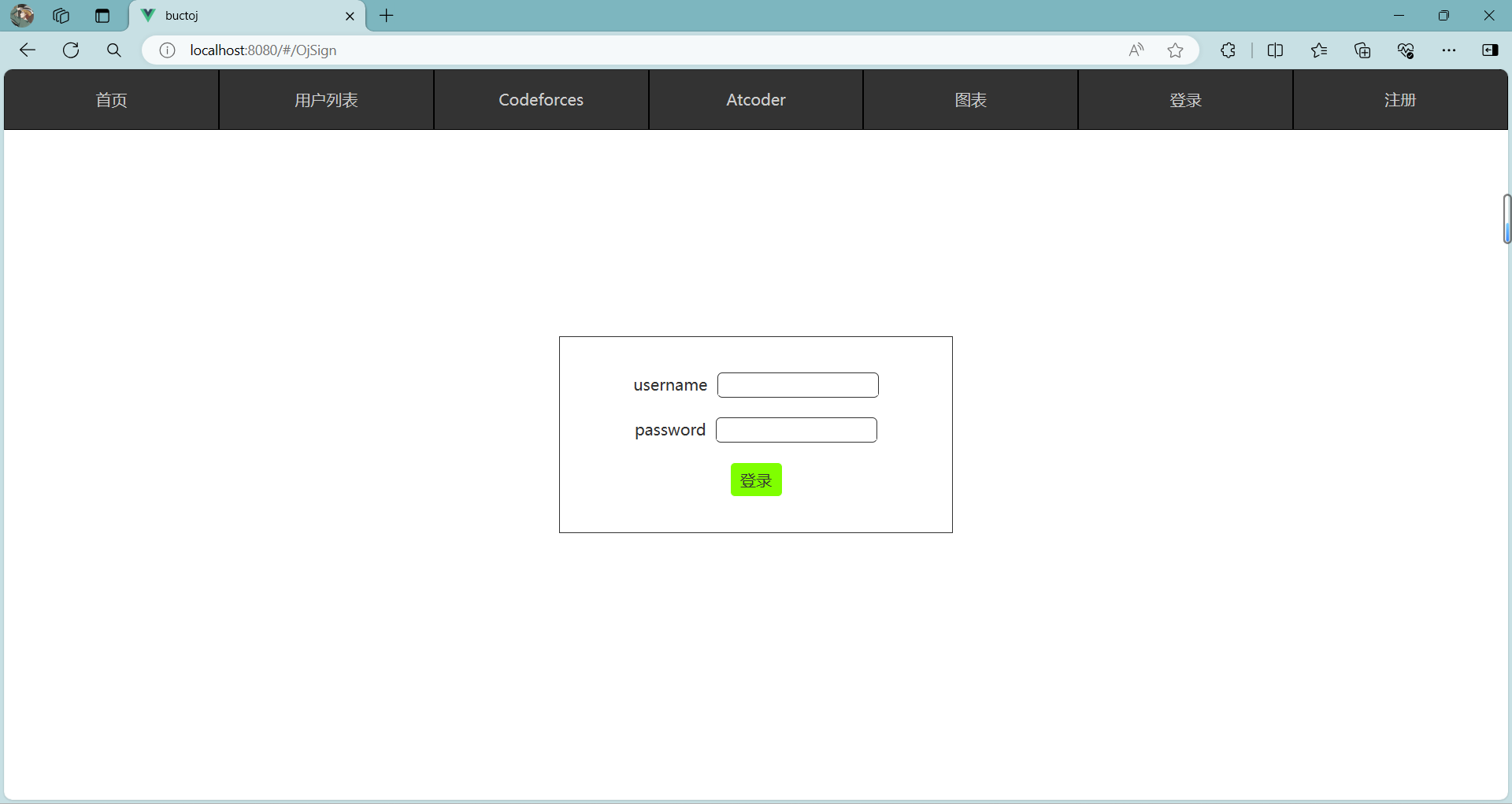
3.2.5codeforce账户排名实现效果

3.2.6atcoder比赛列表实现效果

3.2.7atcoder账户排名实现效果

3.2.8用户注册及登录页面实现效果

（1）用户登录页面：输入用户名及密码登录账户。

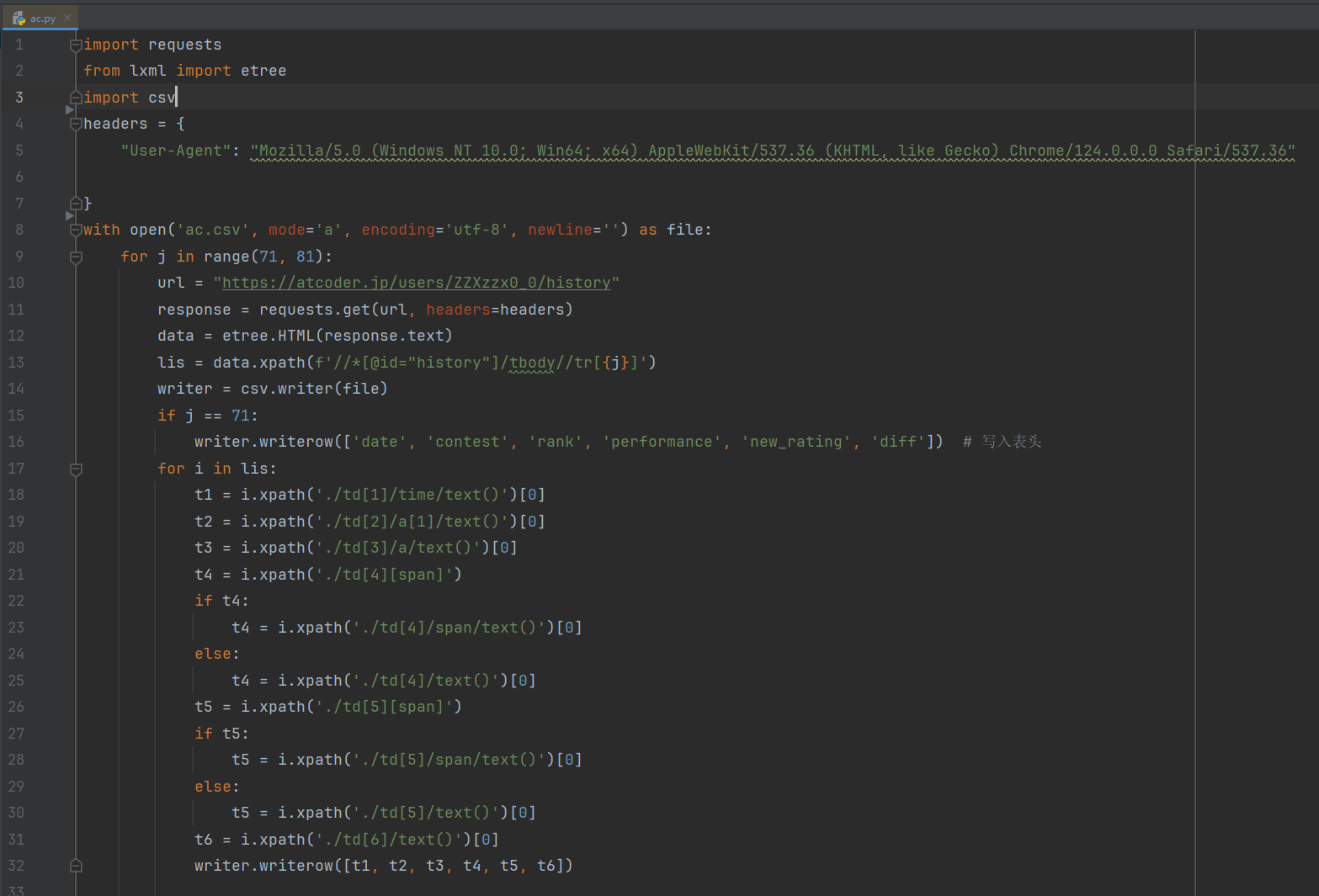


3.2.9图表实现效果

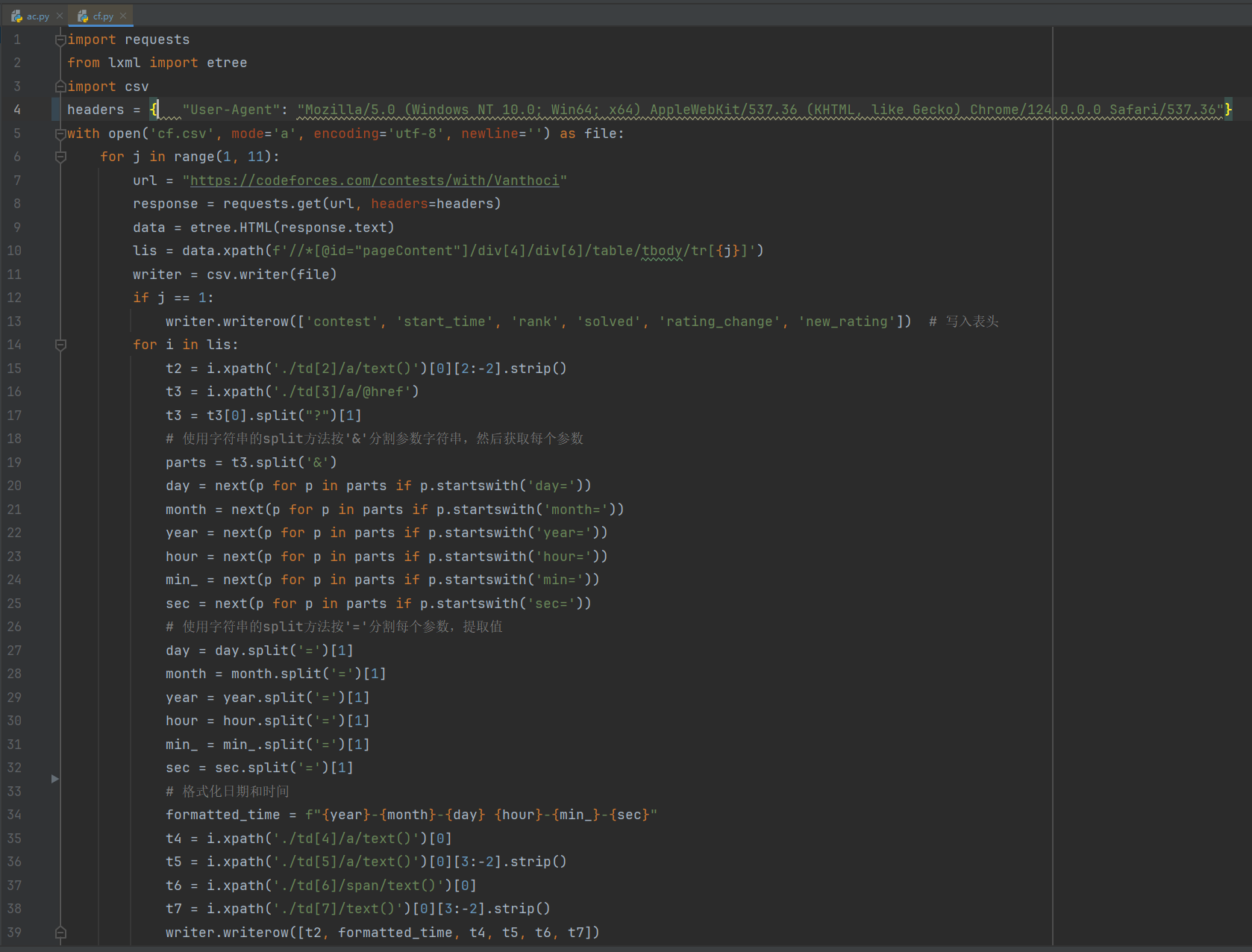
3.3 代码规范

3.3.1 爬虫代码规范

（1）编写python代码，爬取https://atcoder.jp/ 解析atcoder网页可以获得比赛数据，存入csv文件中。



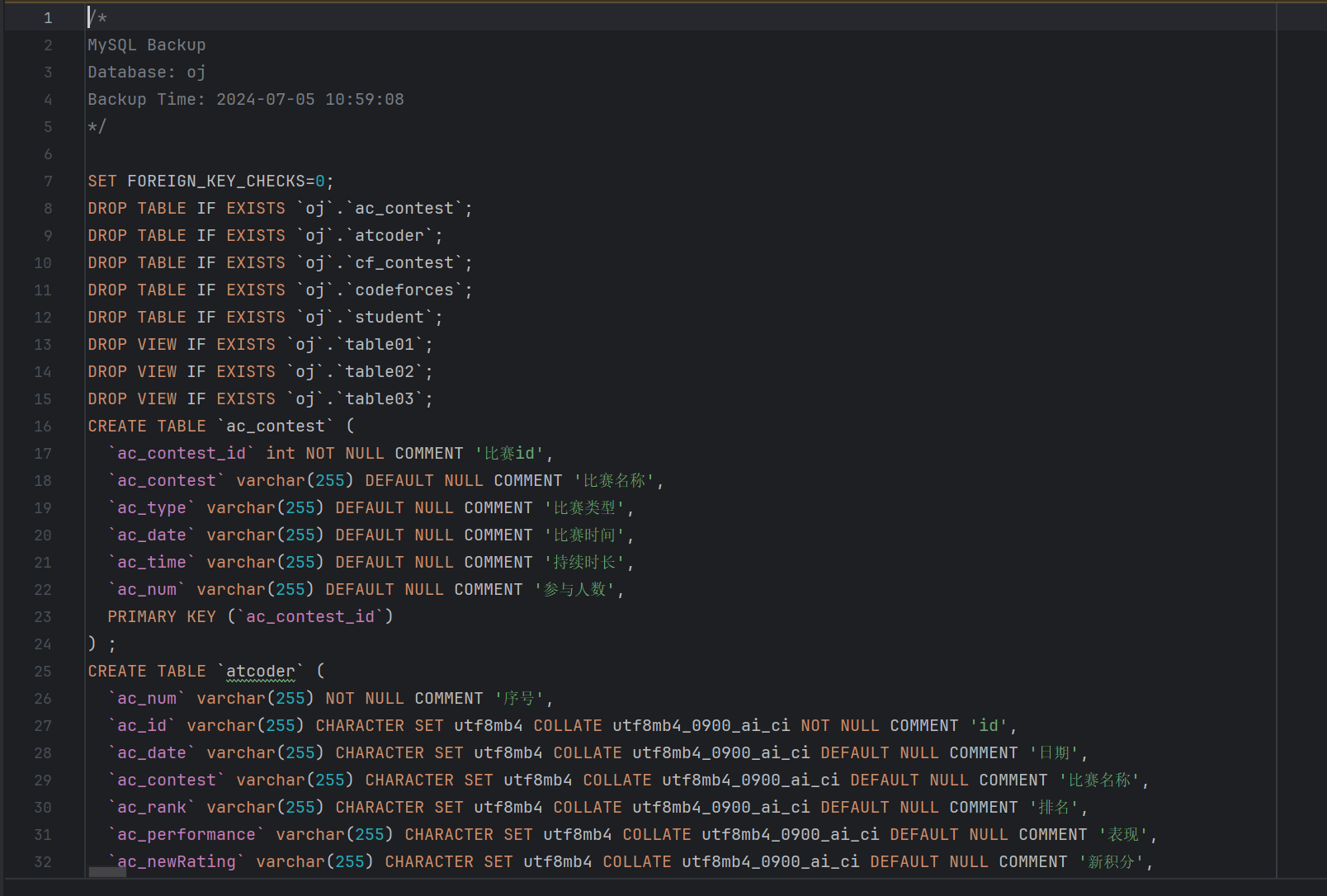
（2）编写python代码，爬取https://codeforces.com/apiHelp 解析codeforces网页通过cf API可以获得比赛数据，存入csv文件中。

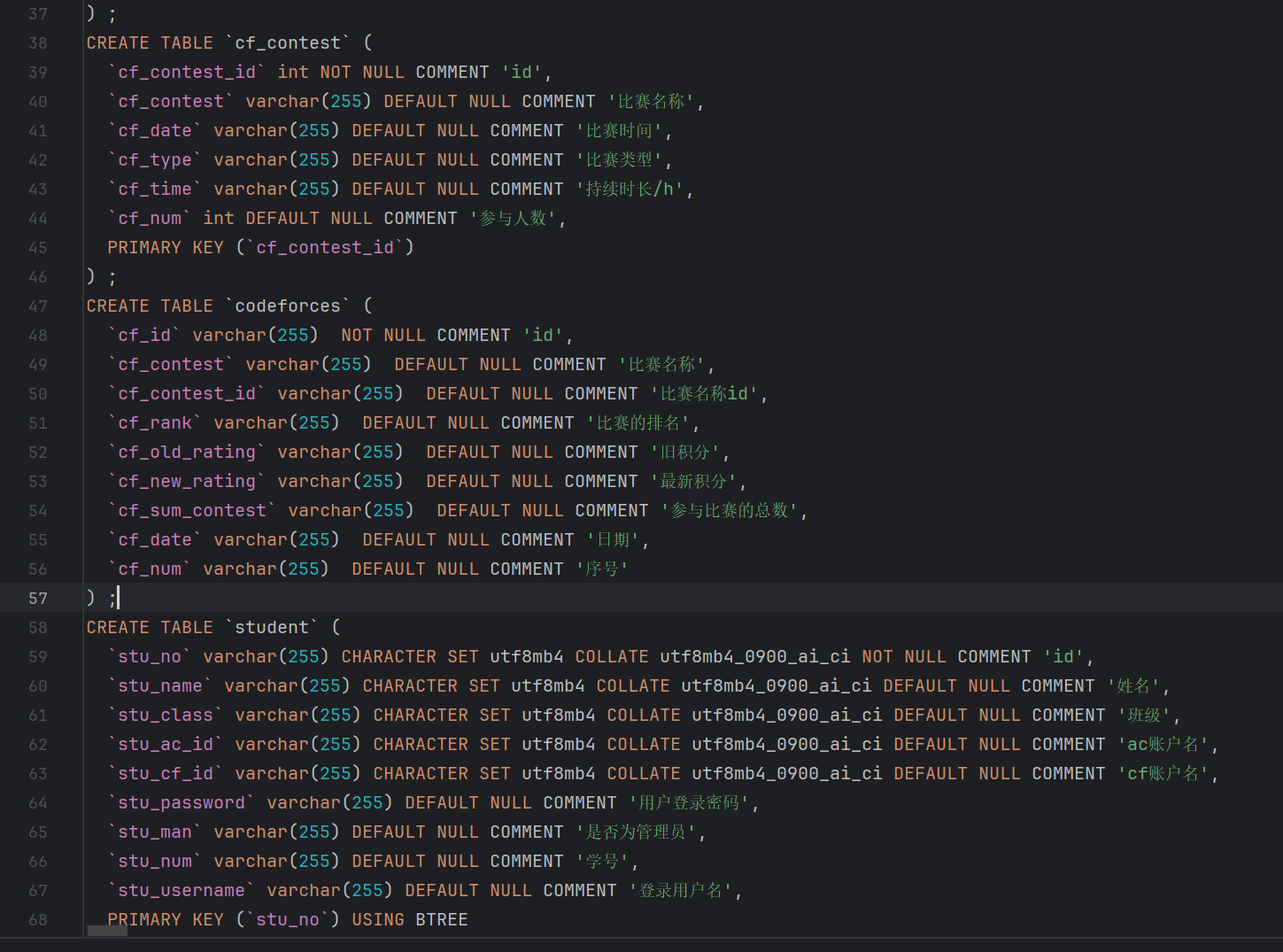


3.3.2 数据库代码规范

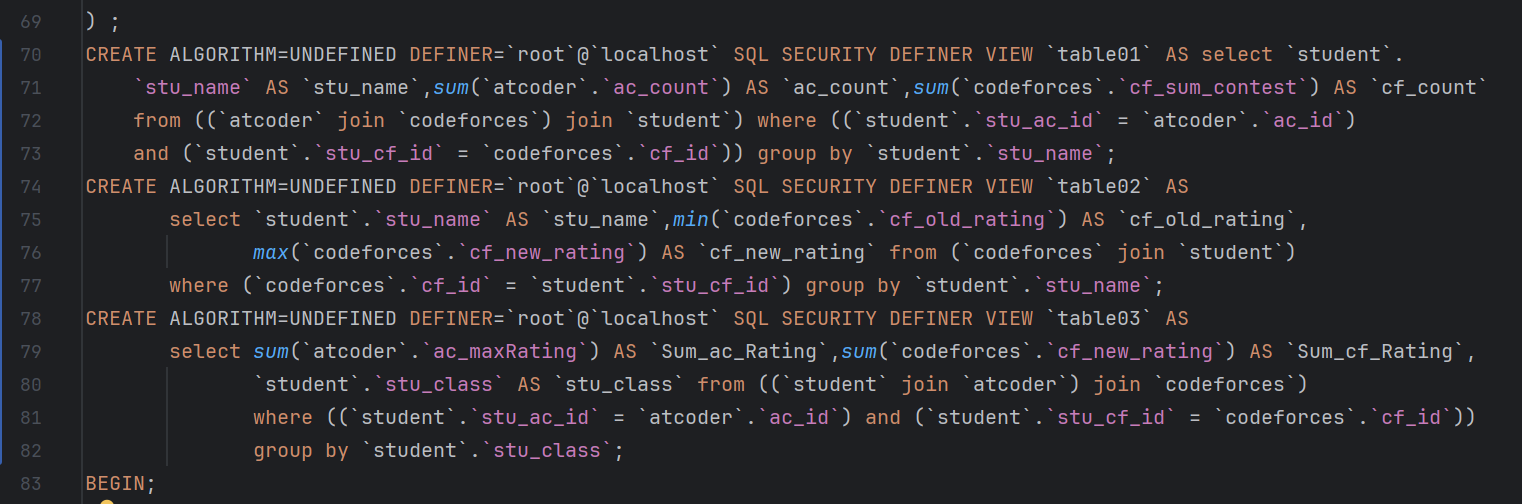
编写sql代码，建表，插入数据。

（1）创建五个表：

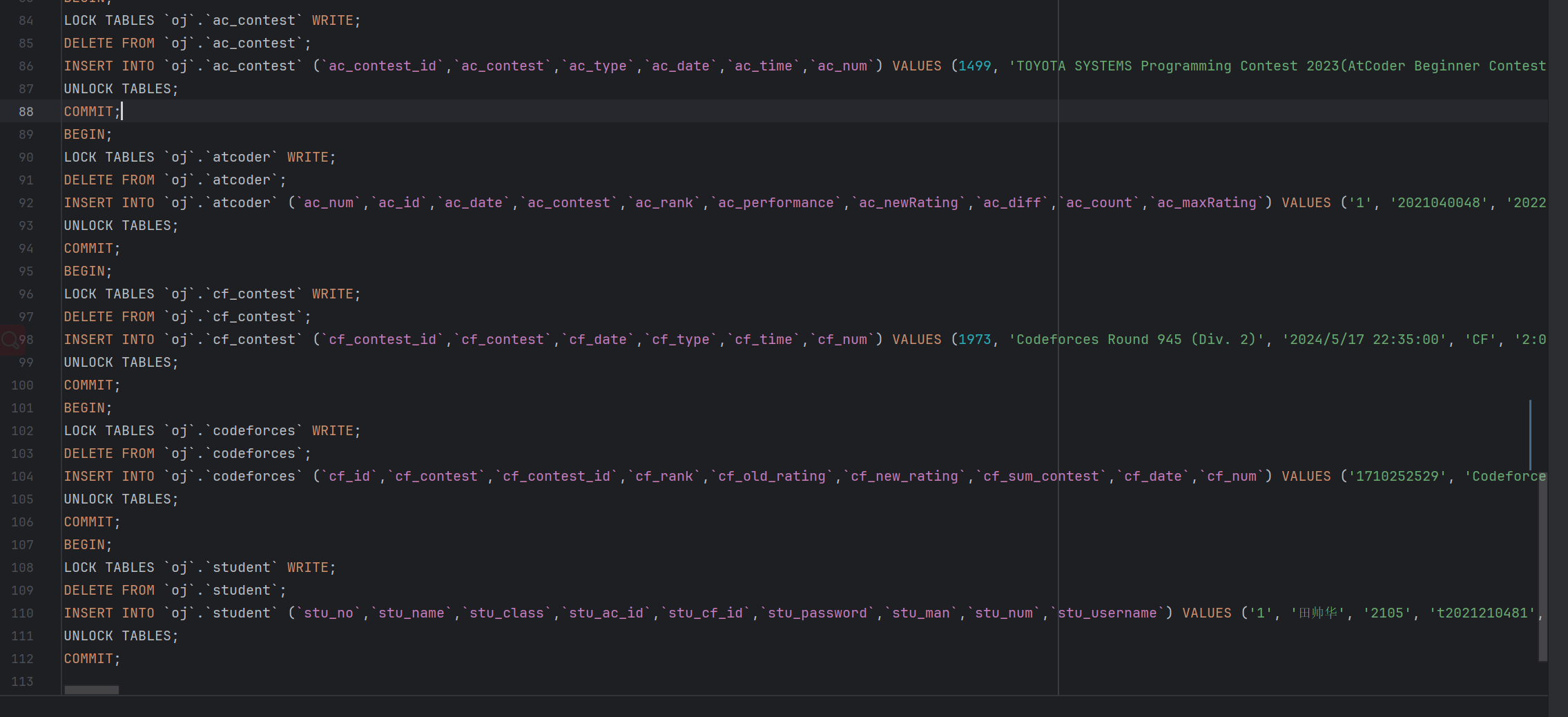




1. 创建三个视图



1. 插入数据



3.3.3 前端代码规范

(1)main.js:初始化了一个Vue.js应用程序，配置了路由、全局样式、图标字体样式，并集成了ECharts图表库和Vue路由器，最后挂载到DOM元素上。



(2)OjIndex.vue:这段代码定义了一个Vue组件，它具有一个包含顶部导航、左侧多个功能模块（提交数图表、智能训练、已结束和即将开始的比赛列表）以及右侧排名信息（Codeforces和Atcoder排名）的布局结构。



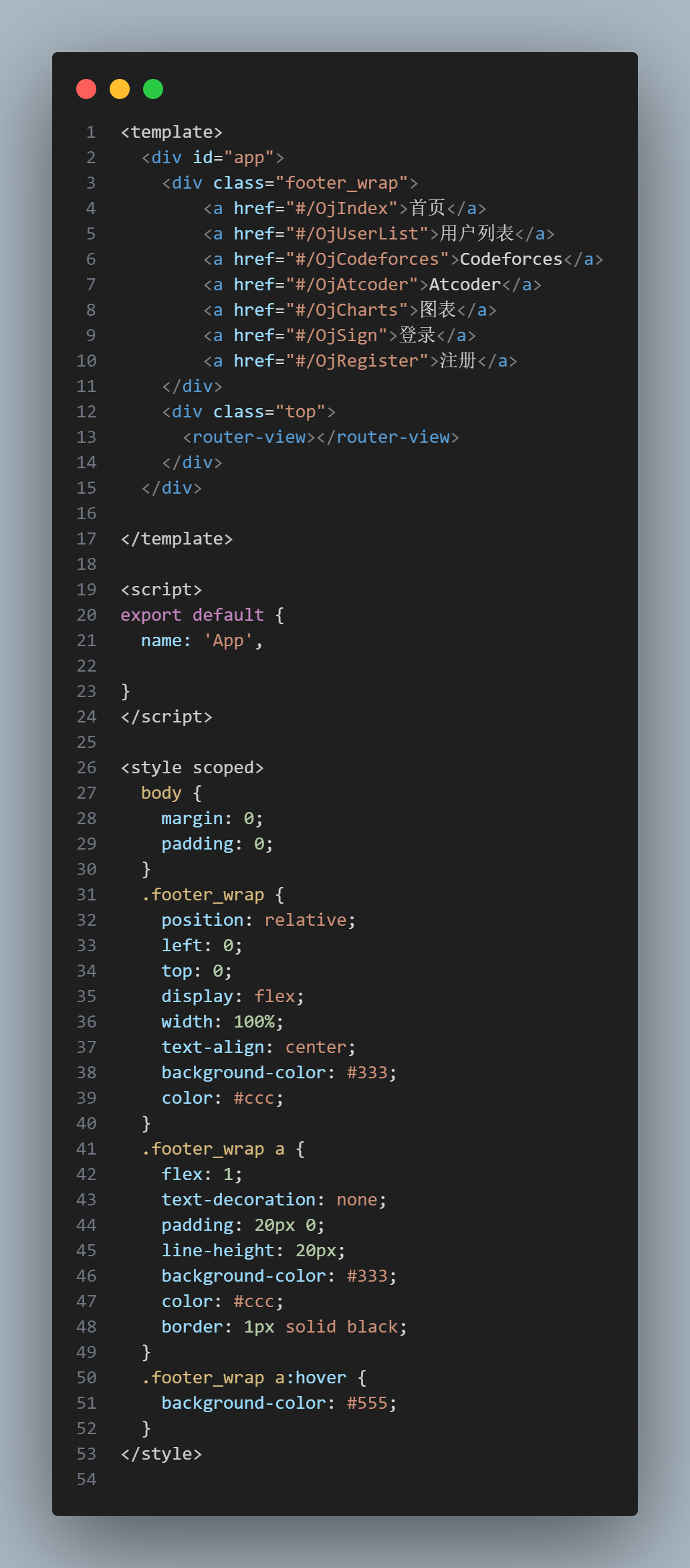
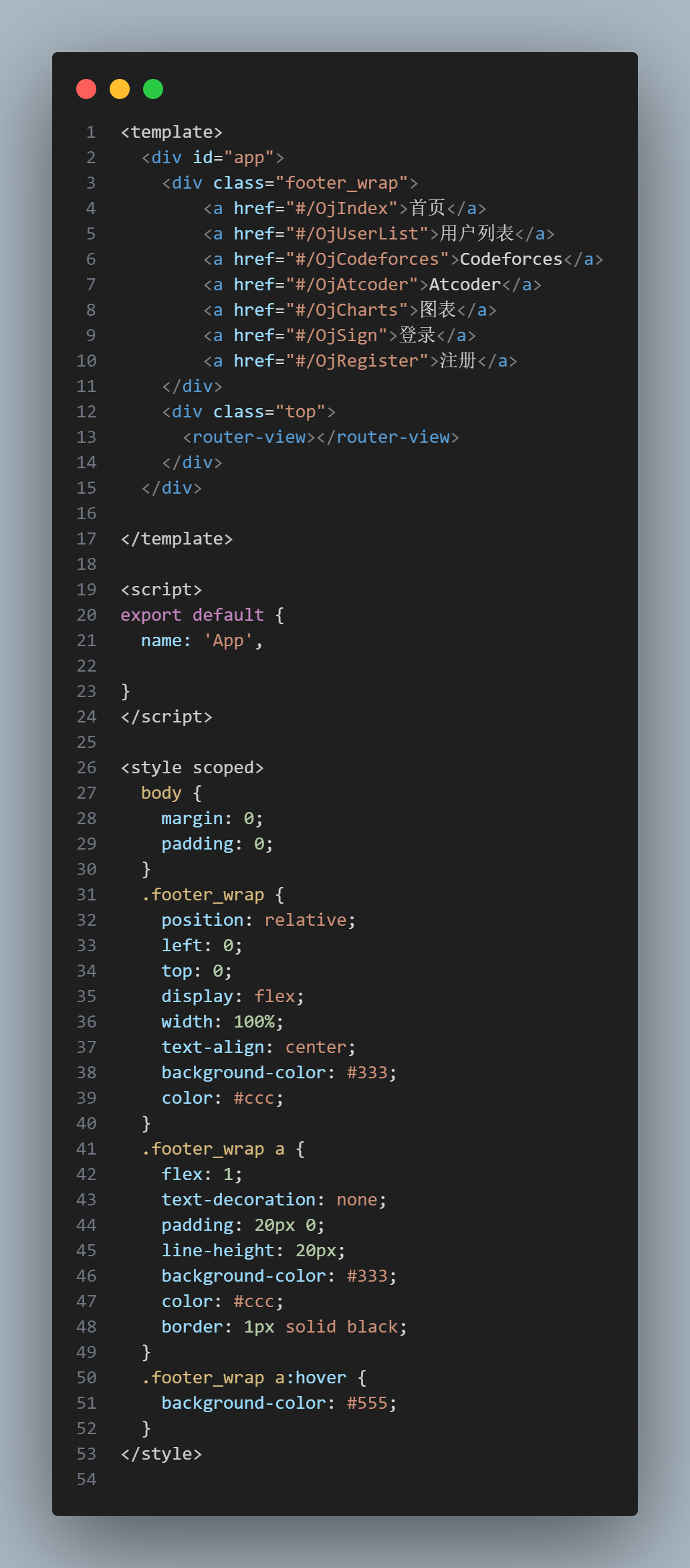
(3)OjSign.vue:这段代码创建了一个具有用户名和密码输入字段以及登录按钮的用户登录界面，包括基本的样式布局和数据绑定。



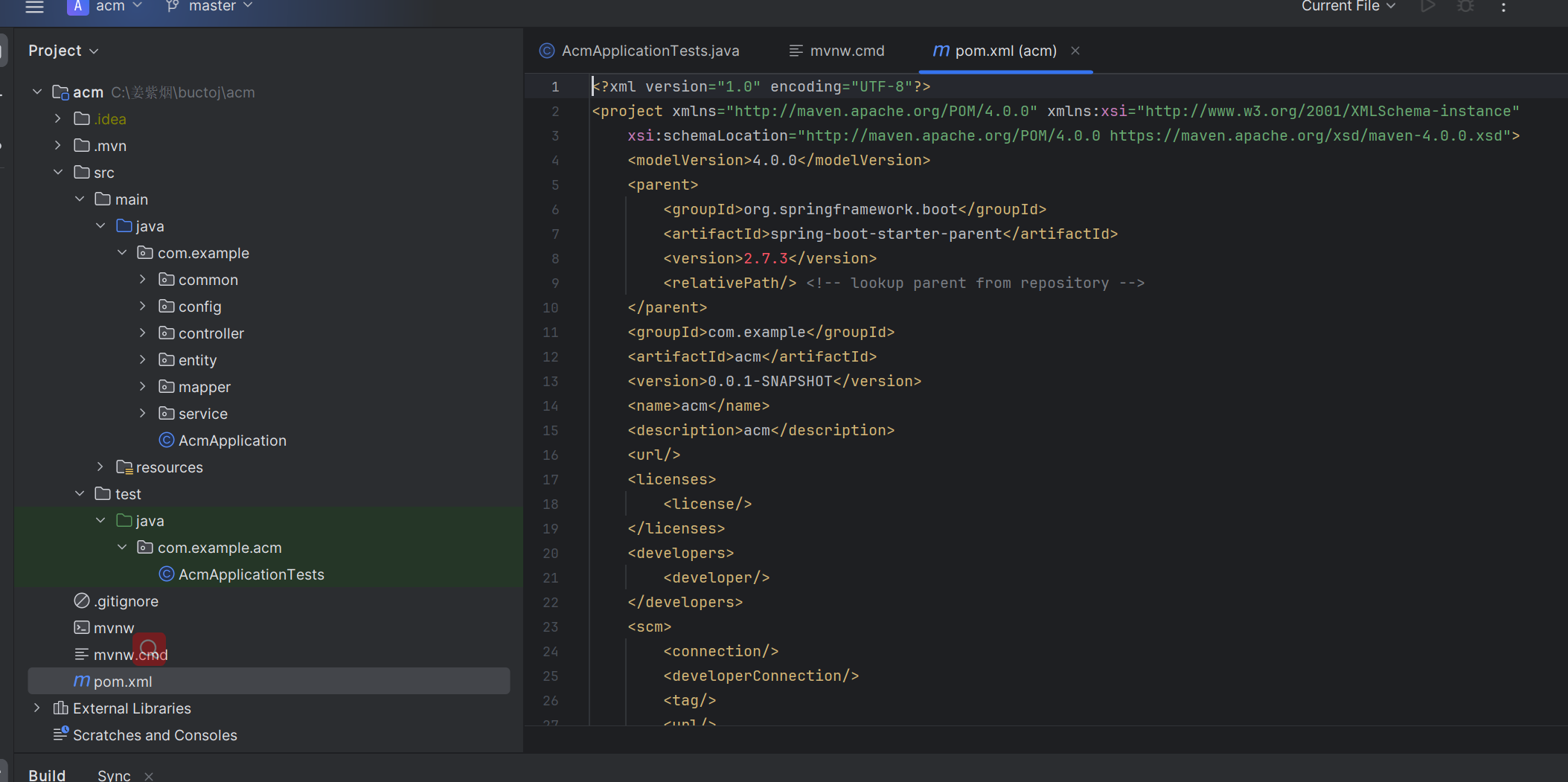
(4)BuctojChart.vue:定义了一个Vue组件，用于初始化并显示一个ECharts图表，该图表展示了近七天的提交数统计，包括赛时未提交数、补题未通过数、赛时AC数和补题AC数的数据，并在组件挂载后自动渲染。

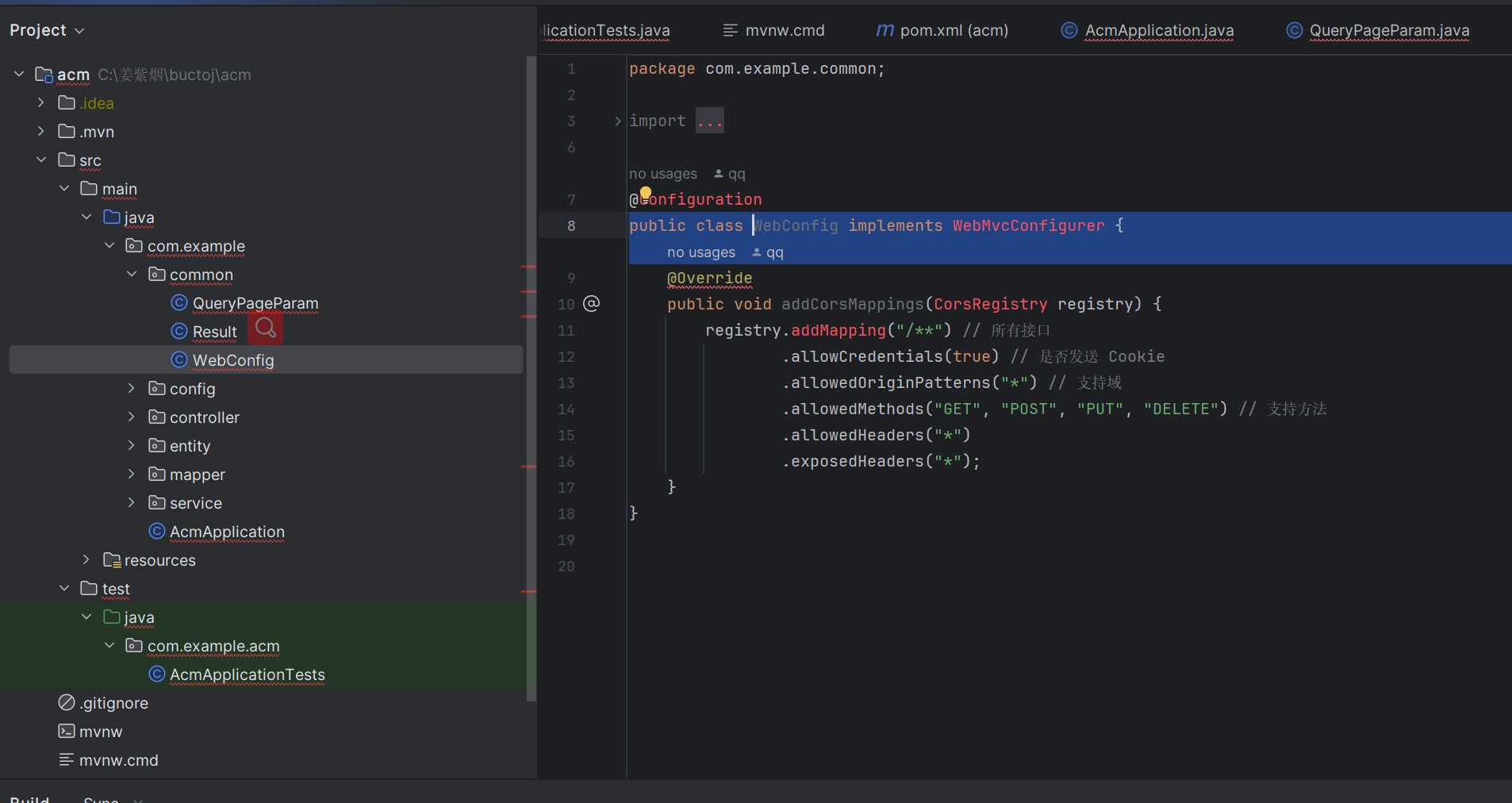


(5)App.vue:定义了一个Vue应用的底部导航栏，包含指向不同页面的链接，以及一个用于展示当前活动路由组件的`<router-view>`插槽。



3.3.4 后端代码规范





3.4 生成式AI工具辅助编程情况介绍

3.5 项目部署与测试情况

1. 结论与展望
2. 参考资源

1.《BUCTCODER 系统开发—软件设计方案》

2. vue.js 官网：https://cn.vuejs.org/ https://v3.cn.vuejs.org/

教程：https://cn.vuejs.org/v2/guide/ https://v3.cn.vuejs.org/

3. ECharts 官网：https://echarts.apache.org/zh/index.html