



山东科技大学

SHANDONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Python 程序设计大作业

姓名	万思昊	学号	202111070519
班级	软件工程 2021-1	项目名称	智领未来—— 大学生发展的数据分析及 可视化平台
成绩		指导教师	陈卓艳

计算机科学与工程学院

2023 年 6 月 29 日

目录

Python 程序设计大作业	1
1 概述.....	4
1.1 问题描述.....	4
1.2 设计背景.....	4
1.3 功能说明.....	5
1.4 附加功能.....	5
1.5 声明.....	5
2 作品介绍.....	6
2.1 作品简介.....	6
2.2 模块设计.....	6
2.3 使用效果.....	7
2.3.1 首页.....	7
2.3.2 校园竞赛数据系统.....	7
2.3.3 行业发展数据系统.....	11
2.3.4 城市概况数据系统.....	11
2.4 作品功能.....	12
2.4.1 校园竞赛数据系统功能.....	12
2.4.2 行业发展数据系统功能.....	12
2.4.3 城市概况数据系统功能.....	13
2.4.4 创新点.....	13
3 设计思路.....	14
3.1 设计来源.....	14
3.1.1 需求来源.....	14
3.1.2 数据来源.....	14
3.2 设计步骤.....	14
3.3 设计思想.....	15
4 技术路线.....	17
4.1 前端技术.....	17
4.1.1 前端框架.....	17

4.1.2 可视化组件.....	18
4.1.3 动态背景.....	18
4.2 后端技术.....	18
4.2.1 后端框架.....	18
4.2.2 数据库及其管理.....	19
4.3 关键技术.....	19
4.3.1 数据执行缓冲.....	19
4.3.3 构架查询工具.....	20
5 测试报告.....	21
6 安装及使用.....	22
7 编码实现.....	24
7.1 可视化颜色部分设计.....	24
7.2 嵌套查询算法.....	24
7.3 爬虫技术.....	25
7.4 路由技术.....	26
7.5 数据导入弹窗方法.....	27
7.6 其它.....	28
8 遇到的主要问题及解决方法.....	29
8.1 搜索结果有用数据的提取.....	29
8.2 Canvas 插件作为动态背景.....	29
9 总结与展望.....	30

1 概述

1.1 问题描述

在大学阶段科研竞赛无疑是提高能力和扩展技能的一种很好的方式，绝大多数大学新生保留了高中的学习习惯和思维方式，理论应用实践能力差，面对科研竞赛是很迷茫的。会产生很多疑问以至于不敢迈出走向科研创新的第一步，以往学生的历史竞赛数据就可以起到引导作用，但校园科研竞赛数据零散分散，整理起来难度大，汇总起来不直观，深度挖掘数据本身的价值，会对学校的进一步发展起到更大的促进作用。

大学接近尾声，进入社会的第一个考验就是行业和城市的选择，面对众多行业和向往又陌生的城市又会产生新的疑问，面对众多想了解的行业和城市信息却无从下手。

因此，对于**科创竞赛、行业发展、城市概况**的数据系统的分析与管理对大学生的发展具有重要意义，一个面向大学生和高校的助力大学生发展的数据可视化平台就此而生。

1.2 设计背景

对于**竞赛数据**而言，数据量大且零散，数据呈现不成体系不够直观，导致科创竞赛数据丧失了原有的价值。并且现在很多高校对科研竞赛只重视前期的竞赛的推广和参与，对后期竞赛结果数据不进行深入挖掘，会产生如下问题：

对于**学校和指导老师**而言，不了解学生的知识技能水平以及对于本校学生而言的比赛难度，盲目的推广开展竞赛，投入多回报少，事倍功半。

对于学生而言，大学更多的是自主学习，如果没有客观数据做以支撑，很难形成横向和纵向对比，有可能错过参加科创竞赛的最好时间，从而降低学习的主动性。

对于**行业发展状况数据**，很多学生可能存在刻板印象，对于当前的发展前景比较好的行业把握不准，现在行业发展快，技术迭代迅速，一个发展比较好的产业可能几年时间以后就走向没落，对行业选择和职业规划造成影响。

对于**热点城市概况数据**，一个城市的指标繁多复杂，每个城市的地理位置、人文自然环境、经济发展千差万别，没有直观的整体对比，孰优孰劣难以区分。

1.3 功能说明

智领未来数据可视化系统从多角度对数据进行可视化展示。

- ◆ 对历史竞赛情况的总结与分析，对未来学生竞赛方向做出引导，可以生成学校和各学院的竞赛情况总览表。相关数据通过**爬虫办法**获取。
- ◆ 对目前行业发展总体状况，各行业获利情况等做出**可视化展示**。
- ◆ 各大城市基础设施建设、医疗卫生、娱乐、教育等方面进行**可视化展示**。

（上述的可视化展示使用 **Apache 的 Echarts 实例**）

1.4 附加功能

- ◆ 平台支持**多主题切换**能够满足不同用户的喜好。
- ◆ 使用 **Flask 框架**集成项目，打造高可用的 **Web 应用程序**。
- ◆ 基于相应组件实现网页见的**路由功能**。
- ◆ 使用 Navicat Premium 15 连接并管理 MySQL。实现**数据库管理和使用**。

1.5 声明

竞赛数据集来源为山东科技大学和其计算机学院 2021 年留存的竞赛数据，该数据集保存在 GitHub 上。其它静态数据均从相关网页获取得到。

SQL 名称为 computer college.sql.项目将爬取该 SQL 文件。

这个 SQL 文件是校院科联通过 Excel 的方式在 Navicat 上合成的。

GitHub 地址: <https://github.com/WorkflowWan/CourseDesignForPython>,电子版版本待一并更新后上传。

2 作品介绍

2.1 作品简介

智领未来数据可视化平台是基于 Flask、MySQL、Echarts 的数据可视化平台。平台针对校园科创数据、行业发展数据、热点城市概况数据的可视化设计，以达到引导学生科技创新，推动学生发展的作用。平台多主题、分模块、深层次对数据进行深度挖掘和多形式的可视化呈现，利用 Navicat 对经过算法预处理后的数据进行可视化的精细管理。利用 Canvas、jQuery 等实现动静结合的页面主题供用户选择，在保证作品的使用价值的同时，极大提高了作品设计的舒适感、艺术感、科技感。



图 1 校园竞赛数据使用效果图

2.2 模块设计

1. 大学生竞赛数据可视化系统
 - a) 学院竞赛数据可视化
 - b) 学校竞赛数据可视化
2. 热点城市概况可视化系统
3. 行业发展概况可视化系统

2.3 使用效果

2.3.1 首页



图 2 首页使用效果图（界面一）

2.3.2 校园竞赛数据系统



图 3 竞赛界面：计算机学院科创一览描述图（界面二）

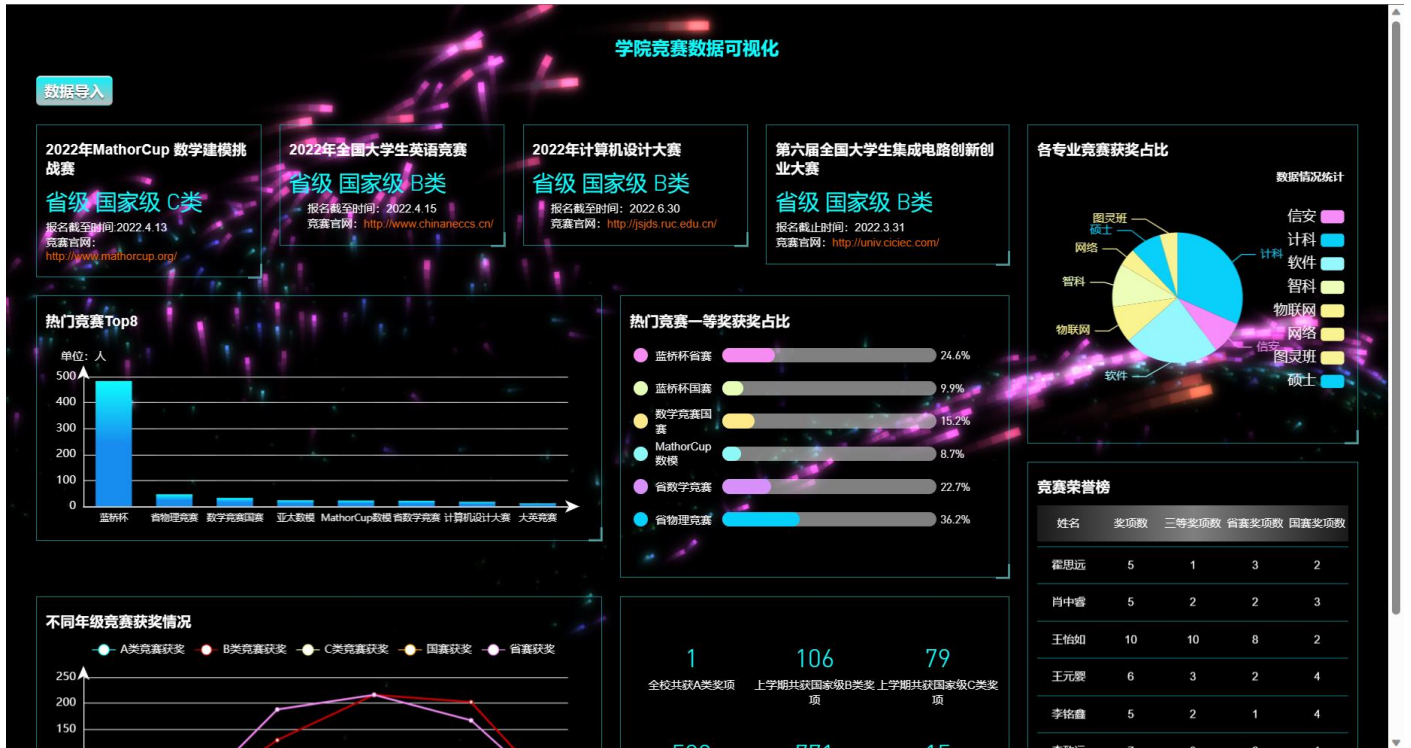


图4 校园竞赛数据系统使用效果图（界面三）

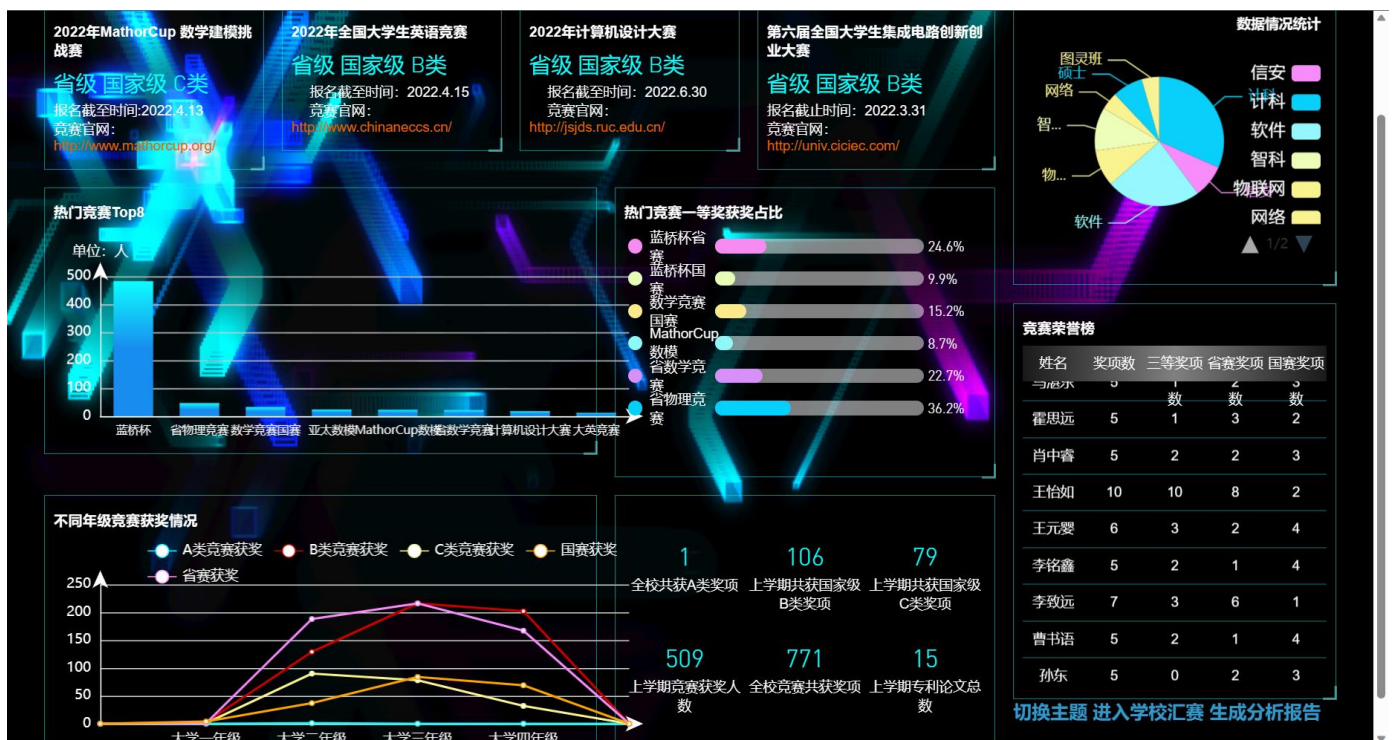


图5 校园竞赛数据系统使用效果图（界面四）

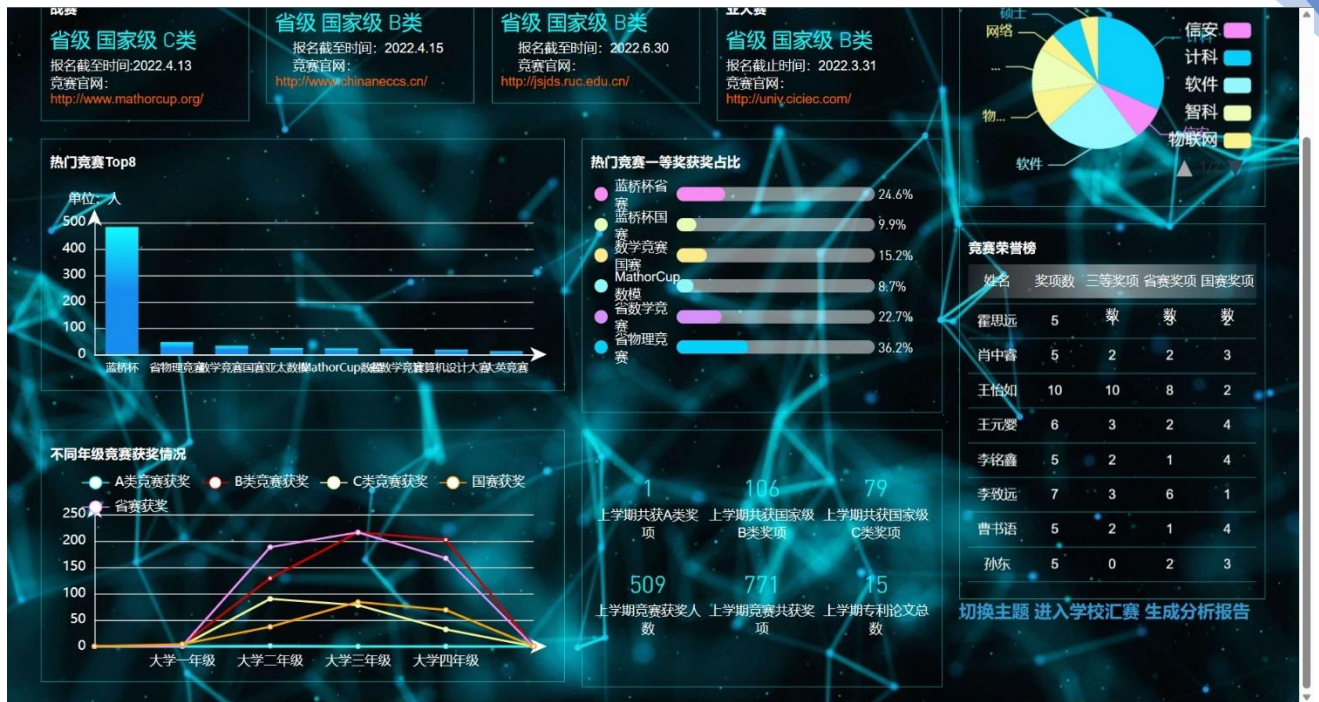


图 6 校园竞赛数据系统使用效果图（界面五）



图 7 校园竞赛数据系统使用效果图（界面六）

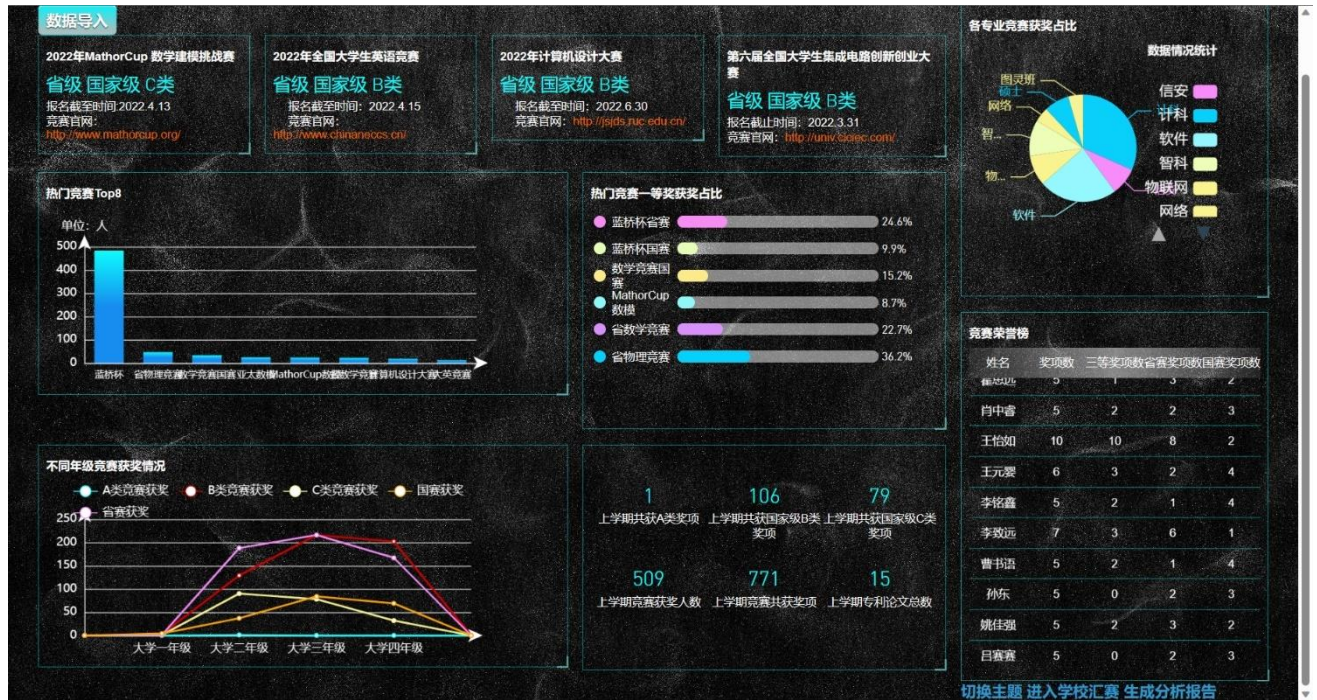


图 8 校园竞赛数据系统使用效果图（界面七）

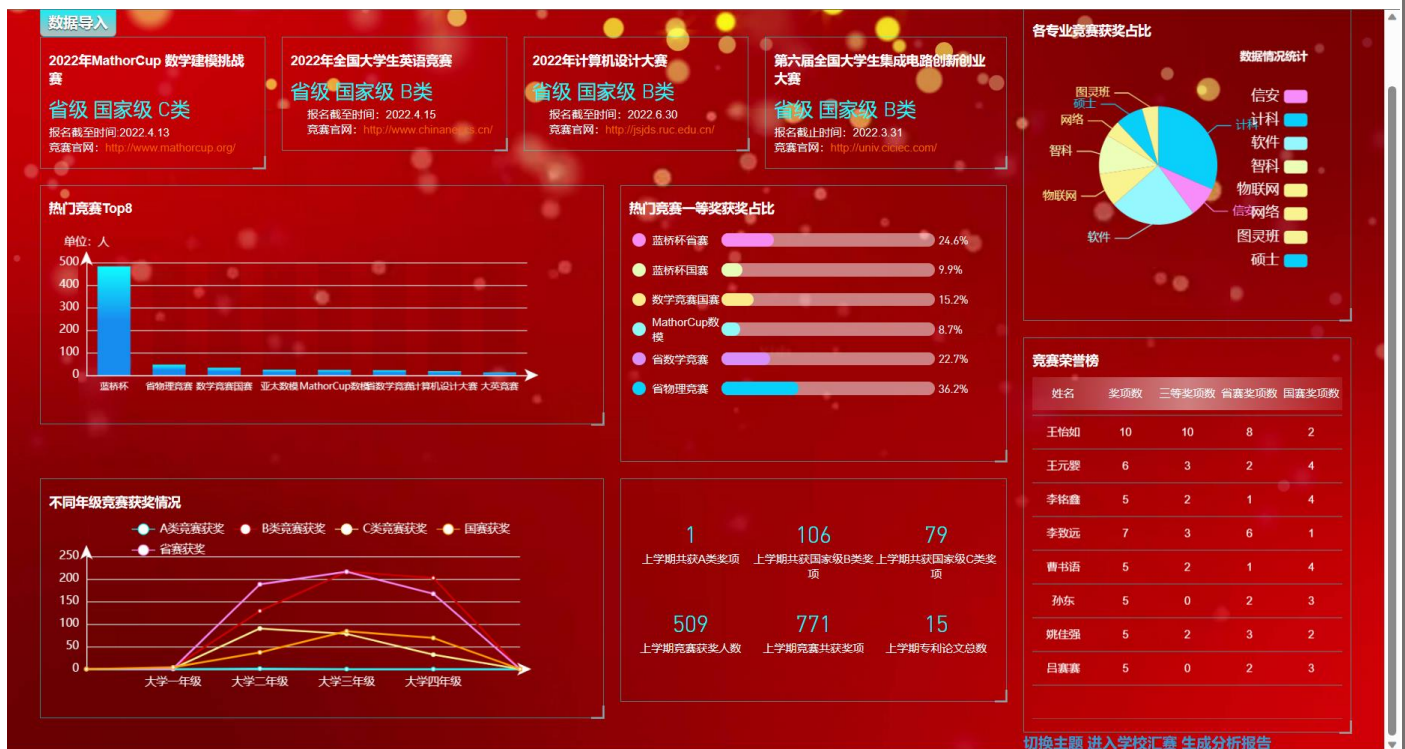


图 9 校园竞赛数据系统使用效果图（界面八）

2.3.3 行业发展数据系统



图 10 行业发展数据系统使用效果图 (界面九)

2.3.4 城市概况数据系统



图 11 热点城市概况数据系统使用效果图 (界面十)

2.4 作品功能

2.4.1 校园竞赛数据系统功能

1. 实现对近期即将举办比赛的名称、报名时间、官网网址连接的展示；
2. 对不同专业、不同学院、不同校区的获奖占比通过饼状图的形式展示出来
3. 展示学生获奖最多的竞赛前八名，以及参与度较高的竞赛一等奖获奖比例，体现出同学们参加较为积极的竞赛以及其获得优异名次的概率；
4. 将不同年级学生获奖数量通过折线图的形式展示，可以对获奖的最佳阶段有宏观的把握
5. 对竞赛优秀学生和竞赛先进学院通过滑动窗口的形式展示。
6. 右下角可以实现学院汇赛和学校汇赛的转换和主题的切换，同时可以生成数据分析报告，实现对于学院或者学校竞赛数据一览；左上角可以实现数据的导入。



图 12 竞赛界面：校园科创一览描述图（界面十一）

2.4.2 行业发展数据系统功能

1. 通过图文列表实现了全国工业发展基本信息、工业分大类行业增长值增长速度排名、工业分大类行业获利占比、各大城市各行业生产总值、热门行业近年发展情况、行业总收入排行、主要行业在或内生产总值的占比、全国热门行业企业数量动静结合的数据展示。
2. 支持对热门行业近年来的发展数据可以的数据视图和下载。

2.4.3 城市概况数据系统功能

1. 将热点城市的医疗机构数量、热点城市文化娱乐团体和公共文化城市数量、热点城市的学校数量通过柱状图的形式展示出来，可以直观的将热门城市若干指标进行对比；
2. 将热点城市人均薪资水平和可支配薪资水平通过折线图的形式表现出来，同时可以和全国人均薪资水平和人均可支配水平进行对比，体现出热门城市的生活经济水平和压力大小；
3. 通过饼状图的形式获取各大城市现有确诊人数、新增确诊人数、新冠治愈率；
4. 将各大热门城市交通拥塞水平通过螺旋柱状图的形式展示出来；并在地图上获取城市坐标展示各大城市商品住宅房单位平方米的房价。

2.4.4 创新点

- ◆ 打破了竞赛数据零散的状况，同时也突破了原来单一的竞赛数量统计、年级统计、类别统计的形式，有针对热门竞赛情况的统计、同时展示了优秀竞赛标兵，先进竞赛学院，近期举办比赛，获奖随年级时间的变化。
- ◆ 结合了行业发展大数据统计分析，城市概况大数据分析统计，对各行各业的发展近况和城市的基础设施建设和教育医疗卫生水平做了多角度的展示，对进入学校的大学生和即将离开学校的毕业生都具有推动发展指引道路的作用。

3 设计思路

3.1 设计来源

3.1.1 需求来源

从未接触过科研竞赛的**大学新生**需要以往竞赛数据作为引导，有计划有方向的提高自己。学校需要对竞赛数据系统科技的管理和分析，以便于后期学校科技竞赛相关工作的开展。

对于即将**步入社会的毕业生**而言，最新的行业发展和城市概况数据可以为今后的职业和生活的选择提供参考。

3.1.2 数据来源

在竞赛方面的数据由各学院和学校科联部门 2021 年统计的最新数据作为数据支持；行业发展情况和城市基本信息概况由国家统计局统计发布在国家数据官网数据作为数据支持。

作品中所使用的地图均是符合国家要求的中国版图。

链接：<https://data.stats.gov.cn/easyquery.htm?cn=A01>

3.2 设计步骤

1. 从需求出发考虑需要的数据，并收集相关数据。
2. 设计数据处理和呈现的方式，将处理后数据，通过 **echarts** 饼状图、柱状图、折线图，动态滚动列表、**map** 等形式展示出来。
3. 构思可视化系统的模块划分，并搭建其逻辑框架。
4. 进行初步的页面设计，划分不同的功能区域。
5. 数据的导入处理，导入数据按照事先设计好的形式和区域完成页面渲染。



图 13 总体设计思路流程图（S 轴分析图）

3.3 设计思想

智领未来是一个助力大学生发展的数据可视化平台。

● 第一模块竞赛数据系统

- 通过竞赛数据的展示，学生可以直观的了解一年中获奖最多、竞赛举办最多的时间，让学生更清楚学校里的竞赛氛围，形成横向比较；也可以了解到在大学四年中参加竞赛的最好时期和获奖概率最大的时间，形成纵向比较；
- 对于热门竞赛、正在举办的竞赛信息、和获奖等级占比的展示可以为同学竞赛的选择路线和报名提供便利，也为学校竞赛工作的开展提供便利。
- 对于页面背景的选择，开放多种主题可供选择，有静态动态、科技简约等风格，为用户提供更多选择的余地，统计报表的生成，可以使用户从总地上掌握学校学院的获奖情况，便捷直观。

- 第二模块行业发展可视化

- 通过分大类工业生产总产值、增长速率、获利占比为投身工业制造类的大学生提供参考，热门行业今年发展情况，商业大类的获利排名，主要行业生产总值占比、建厂数，为大学生的就业供参考，将各大城市的行业生产总值通过 map 的形式动态展示，为城市概况数据模块提供辅助作用。

- 第三模块城市发展模块

- 该模块选择了几个大中城市：北京、上海、广州、深圳、杭州、青岛、天津、深圳。利用柱状图的形式展示各个城市的医疗机构数量、公共文化场所数量、学校数量，交通拥塞指数，通过这些指标一定程度反应了城市的医疗卫生水平，娱乐文化水平、教育水平、交通拥塞指数，这些指标与生活息息相关；
- 将城镇人均年收入和人均可支配收入同通过一张折线图的形式展示出来，可以反映出城市的人均生活水平和生存压力以及城市基础设施水平；
- 通过饼状图对各城市现有确诊、新增确诊、新型冠状病毒患者治愈率来反映城市的疫情当前形势和防控水平；
- 通过 map 展示各大热点城市和省会城市的房价为同学们的安家立业提供参考依据。

4 技术路线



图 14 技术路线树状图

4.1 前端技术

4.1.1 前端框架

- ◆ HTML（HyperText Markup Language）超文本标记语言。
- ◆ CSS（Cascading Style Sheets）是级联样式表。
- ◆ JavaScript 用于前端页面的 DOM 处理,本项目采用快速、简洁的 JavaScript 框架 jQuery。

4.1.2 可视化组件

ECharts 是一款基于 JavaScript 的数据可视化图表库，提供直观，生动，可交互，可个性化定制的数据可视化图表。还运用了 FontScroll 文字无限循环滚动插件、NumScroll 数字滚动特效插、Wordry 文字反转插件。

4.1.3 动态背景

CanvasAPI（画布）是在 HTML5 中新增的标签用于在网页实时生成图像，并且可以操作图像内容，基本上它是一个可以用 JavaScript 操作的位图（bitmap），此外还运用了 Particles 粒子动画插件进行界面的丰富。

4.2 后端技术

4.2.1 后端框架

平台采用 FlaskWeb 微框架，Flask 是一个使用 Python 编写的轻量级 Web 应用框架。其 WSGI 工具箱采用 Werkzeug，模板引擎则使用 Jinja2。Flask 的基本模式为在程序里将一个视图函数分配给一个 URL，每当用户访问这个 URL 时，系统就会执行给该 URL 分配好的视图函数，获取函数的返回值并将其显示到浏览器上，其工作过程见图。

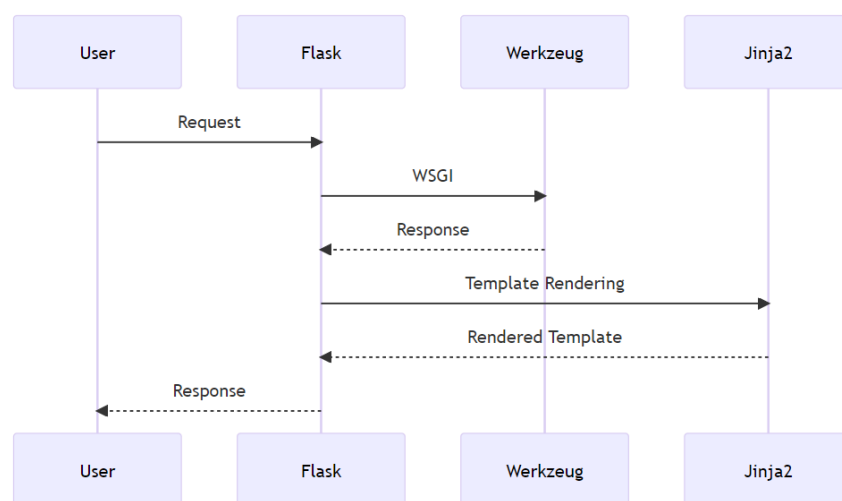


图 15 Flask 工作过程

4.2.2 数据库及其管理

平台使用的数据库是关系型数据库管理系统 MySQL，将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。其使用的 SQL 语言是用于访问数据库的最常用标准化语言。具有体积小、速度快、总体拥有成本低，等优势。可多重连接的数据库管理工具 Navicat Premium。提高了数据库的可操作性。

4.3 关键技术

4.3.1 数据执行缓冲

数据库中存放原始数据，当系统启动以后，数据库建立与 flask 的连接，执行程序部分开始并行执行，从数据库中取出数据放入数据缓存池中，进行页面渲染时，前端的多个模板通过 jinja2 的模板驱动从数据缓冲池中取出需要的数据。

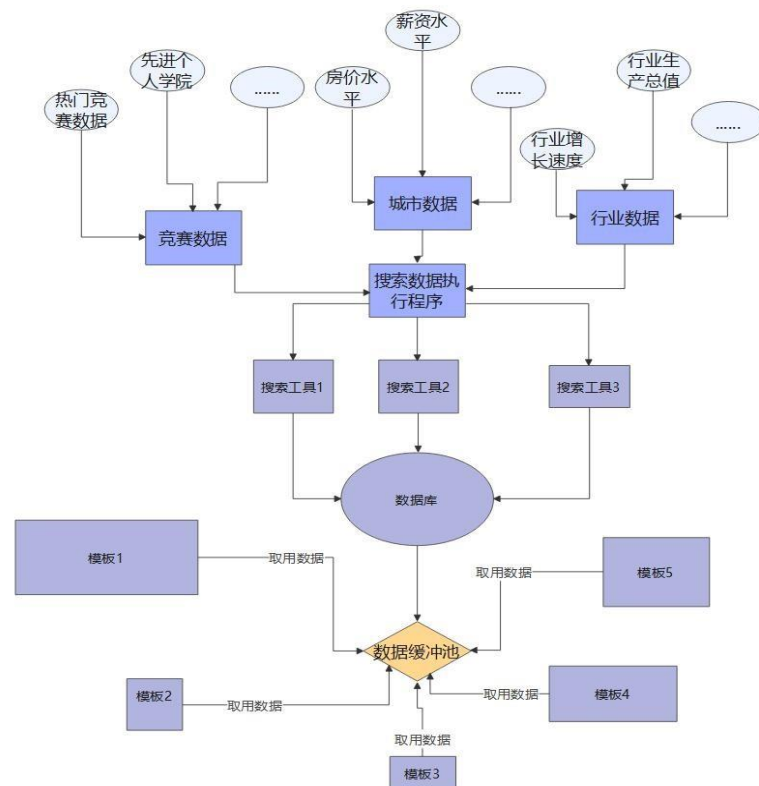
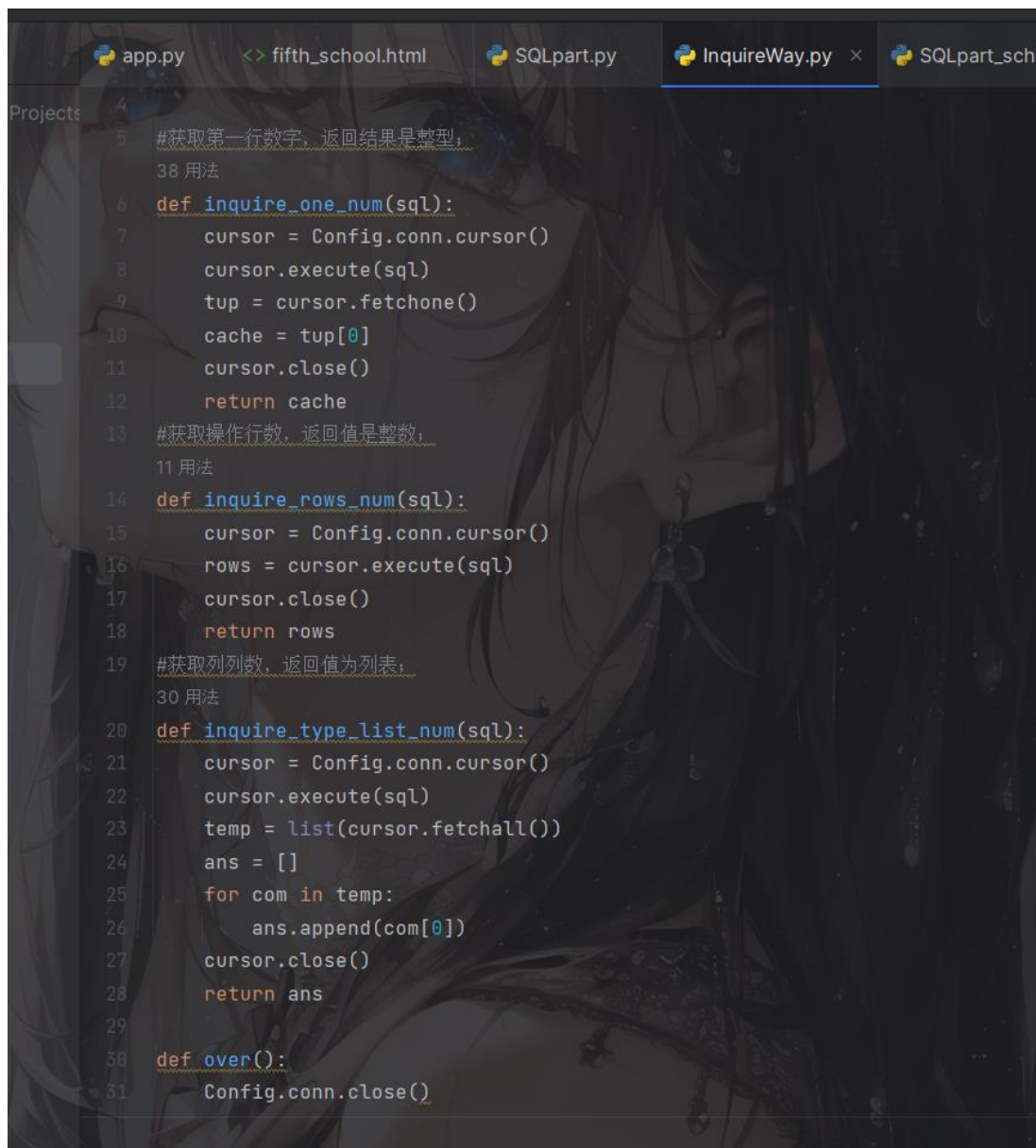


图 16 数据缓冲逻辑架构图

4.3.3 构架查询工具

对于数据搜索结果在 Navicat 中会以一个新的表的形式存在，在 pymysql 中呈现出来的。结果是以元组的形式出现的，为了从搜索结果中获得目标数据类型，制作了三个搜索工具，数字获取、列表获取、行数获取。



```
4
5 #获取第一行数字，返回结果是整型；
38 用法
6 def inquire_one_num(sql):
7     cursor = Config.conn.cursor()
8     cursor.execute(sql)
9     tup = cursor.fetchone()
10    cache = tup[0]
11    cursor.close()
12    return cache
13 #获取操作行数，返回值是整数；
11 用法
14 def inquire_rows_num(sql):
15     cursor = Config.conn.cursor()
16     rows = cursor.execute(sql)
17     cursor.close()
18     return rows
19 #获取列数，返回值为列表；
30 用法
20 def inquire_type_list_num(sql):
21     cursor = Config.conn.cursor()
22     cursor.execute(sql)
23     temp = list(cursor.fetchall())
24     ans = []
25     for com in temp:
26         ans.append(com[0])
27     cursor.close()
28     return ans
29
30 def over():
31     Config.conn.close()
```

图 17 搜索工具的实现

5 测试报告

使用开源测试工具 **ApacheJMeter** 在秒钟对平台模拟访问两千次(执行 **Get** 命令)，平台异常率为零，性能稳定，下面是平台测试的汇总报告和聚合报告。

Label	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec	平均字节数
HTTP请求	4000	5	0	42	7.45	0.00%	52.0/sec	79.38	5.94	1563.0
总体	4000	5	0	42	7.45	0.00%	52.0/sec	79.38	5.94	1563.0

图 18 测试汇总报告

Label	# 样本	平均值	中位数	90% 百分位	95% 百分位	99% 百分位	最小值	最大值	异常 %	吞吐量	接收 KB/sec	发送 KB/sec
HTTP请求	2000	2	2	5	8	15	0	21	0.00%	996.0/sec	1520.29	113.80
总体	2000	2	2	5	8	15	0	21	0.00%	996.0/sec	1520.29	113.80

图 19 测试聚合报告

6 安装及使用

本项目无需安装，通过浏览器即可打开使用浏览器版本要求：

- IE8/9/10/11
- Chrome
- Firefox
- Safari

生成部署依赖包的信息：

```
pip install -r requirements.txt
```

得到如下信息：

```
click==8.0.4
colorama==0.4.4
cycller==0.11.0
et-xmlfile==1.1.0
Flask==2.0.3
fonttools==4.32.0
greenlet==1.1.2
itsdangerous==2.1.1
Jinja2==3.0.3
kiwisolver==1.4.2
MarkupSafe==2.1.0
matplotlib==3.5.1
mysql-connector-python==8.0.28
numpy==1.22.3
openpyxl==3.0.9
packaging==21.3
Pillow==9.1.0
PyMySQL==1.0.2
pyparsing==3.0.8
```

```
python-dateutil==2.8.2
```

```
six==1.16.0
```

```
SQLAlchemy==1.4.32
```

```
Werkzeug==2.0.3
```

为启动项目，请执行：

```
python -m flask run --host=0.0.0.0
```

7 编码实现

7.1 可视化颜色部分设计

在总体的颜色设计上采用了蓝色为主色调，图表中的颜色尽可能的避开背景颜色，展示。效果明显，并且相邻颜色差异大，避免歧义。

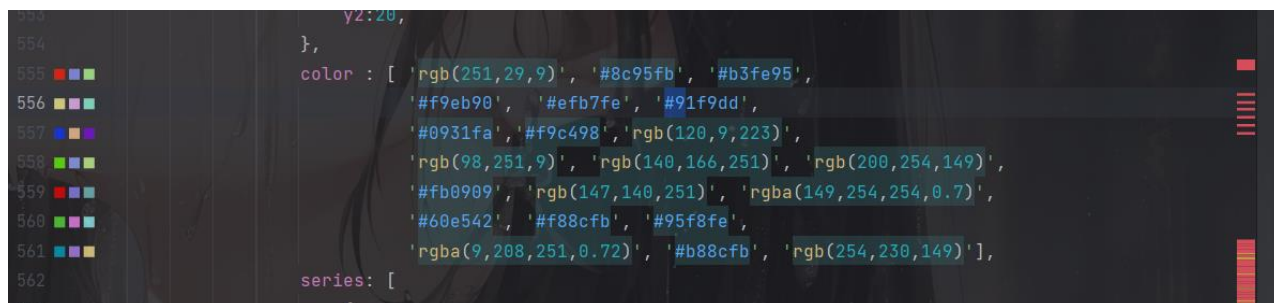


图 20 部分颜色设计

7.2 嵌套查询算法

平台在对单个表做数据搜索的时候需要对多个关键词进行嵌套式的搜索统计，在实现对 SQL 语句搜索结果生成的新表进行搜索，实现了对复杂数据的搜索需求。

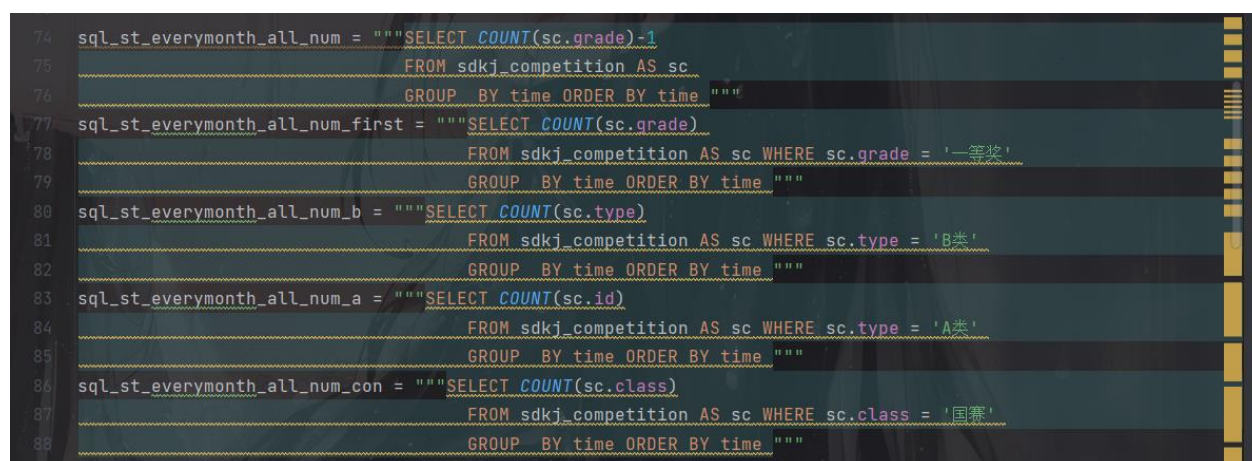
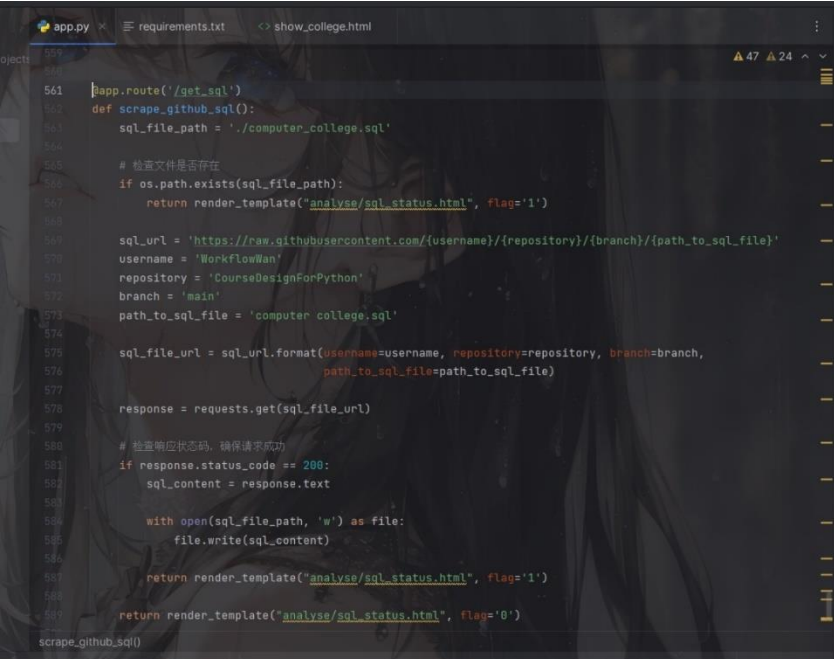


图 21 嵌套查询部分代码

7.3 爬虫技术



```
561 @app.route('/get_sql')
562 def scrape_github_sql():
563     sql_file_path = './computer_college.sql'
564
565     # 检查文件是否存在
566     if os.path.exists(sql_file_path):
567         return render_template("analyse/sql_status.html", flag='1')
568
569     sql_url = 'https://raw.githubusercontent.com/{username}/{repository}/{branch}/{path_to_sql_file}'
570     username = 'WorkflowWan'
571     repository = 'CourseDesignForPython'
572     branch = 'main'
573     path_to_sql_file = 'computer college.sql'
574
575     sql_file_url = sql_url.format(username=username, repository=repository, branch=branch,
576                                  path_to_sql_file=path_to_sql_file)
577
578     response = requests.get(sql_file_url)
579
580     # 检查响应状态码，确保请求成功
581     if response.status_code == 200:
582         sql_content = response.text
583
584         with open(sql_file_path, 'w') as file:
585             file.write(sql_content)
586
587         return render_template("analyse/sql_status.html", flag='1')
588     return render_template("analyse/sql_status.html", flag='0')
589
590 scrape_github_sql()
```

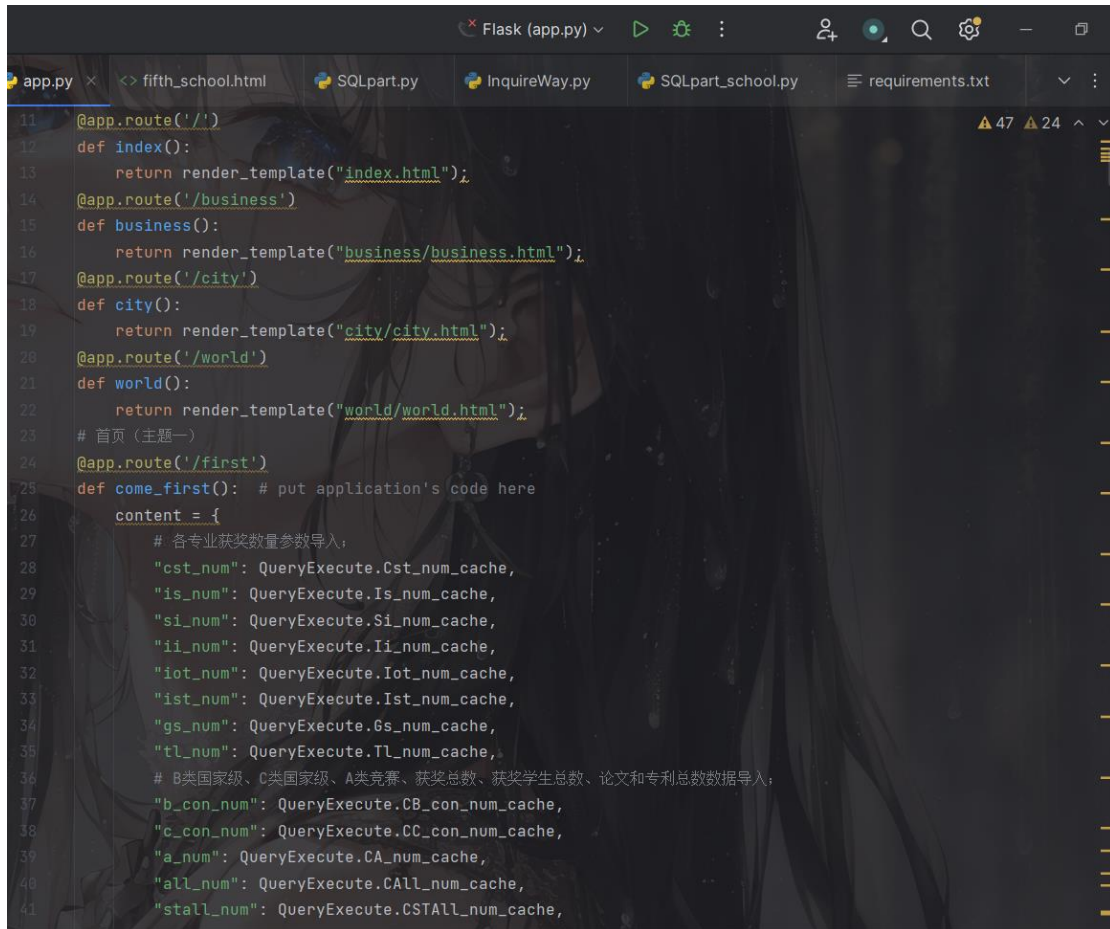
图 22 Requests 相关代码

该代码段是一个基于 Flask 框架的路由函数，用于从 GitHub 上获取 SQL 文件并保存到本地。它包含了一个路由装饰器 `@app.route('/get_sql')`，表示这是一个处理 `/get_sql` 路径的请求的函数。

该段代码执行以下操作：

1. 定义了一个变量 `sql_file_path`，用于指定保存 SQL 文件的路径。
2. 检查 SQL 文件是否已存在于指定路径中。如果存在，则返回一个渲染模板 `analyse/sql_status.html`，并传递参数 `flag` 为 1，表示文件已存在。
3. 如果 SQL 文件不存在，则根据提供的 GitHub 用户名、仓库名、分支名和 SQL 文件路径构建 SQL 文件的 URL。
4. 发起 GET 请求，获取 SQL 文件的内容。
5. 检查响应的状态码，如果为 200 表示请求成功。
6. 将获取到的 SQL 文件内容写入到本地的 SQL 文件中。
7. 返回一个渲染模板 `analyse/sql_status.html`，并传递参数 `flag` 为 1，表示 SQL 文件获取成功。
8. 如果响应状态码不为 200，返回一个渲染模板 `analyse/sql_status.html`，并传递参数 `flag` 为 0，表示 SQL 文件获取失败。

7.4 路由技术



```

11 @app.route('/')
12 def index():
13     return render_template("index.html");
14 @app.route('/business')
15 def business():
16     return render_template("business/business.html");
17 @app.route('/city')
18 def city():
19     return render_template("city/city.html");
20 @app.route('/world')
21 def world():
22     return render_template("world/world.html");
23 # 首页 (主题一)
24 @app.route('/first')
25 def come_first(): # put application's code here
26     content = {
27         # 各专业获奖数量参数导入:
28         "cst_num": QueryExecute.Cst_num_cache,
29         "is_num": QueryExecute.Is_num_cache,
30         "si_num": QueryExecute.Si_num_cache,
31         "ii_num": QueryExecute.Ii_num_cache,
32         "iot_num": QueryExecute.Iot_num_cache,
33         "ist_num": QueryExecute.Ist_num_cache,
34         "gs_num": QueryExecute.Gs_num_cache,
35         "tl_num": QueryExecute.Tl_num_cache,
36         # B类国家级、C类国家级、A类竞赛、获奖总数、获奖学生总数、论文和专利总数数据导入:
37         "b_con_num": QueryExecute.CB_con_num_cache,
38         "c_con_num": QueryExecute.CC_con_num_cache,
39         "a_num": QueryExecute.CA_num_cache,
40         "all_num": QueryExecute.CAll_num_cache,
41         "stall_num": QueryExecute.CSTALL_num_cache,

```

图 23 基于 Flask 框架的路由（含装饰器与 URL 链接）

该代码段定义了多个路由函数用于处理不同路径的请求。每个路由函数都使用@app.route()装饰器来指定对应的路径。

1. index()函数处理根路径'/'的请求，返回渲染模板 index.html。
2. business() 函数 处 理 '/business' 路 径 的 请 求 ， 返 回 渲 染 模 板 business/business.html。
3. city()函数处理'/city'路径的请求，返回渲染模板 city/city.html。
4. world()函数处理'/world'路径的请求，返回渲染模板 world/world.html。
5. come_first()函数处理'/first'路径的请求，返回渲染模板 college/first.html，并传递了一系列参数给模板，这些参数包含了各种数据用于页面展示。

7.5 数据导入弹窗方法

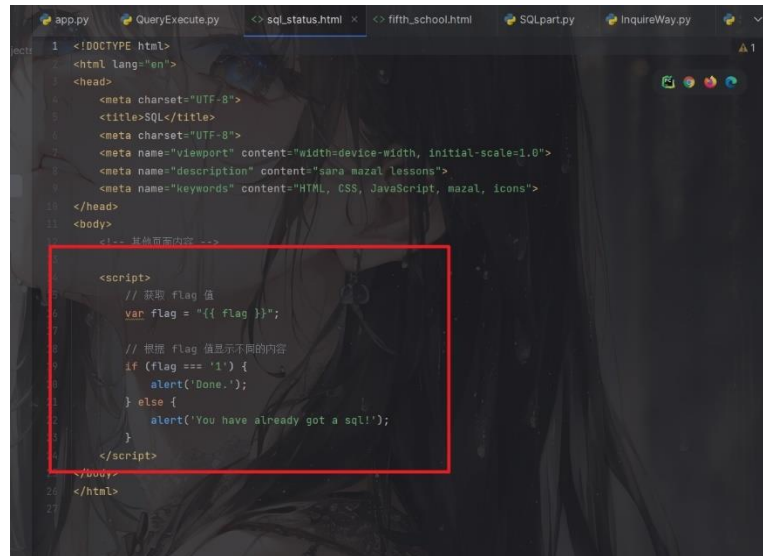


图 24 跳转 SQL 弹窗页面

该代码段使用了一个弹窗函数用于触发弹窗。

1. 爬取成功弹出“Done.”
2. 否则弹出“You have already got a sql!”

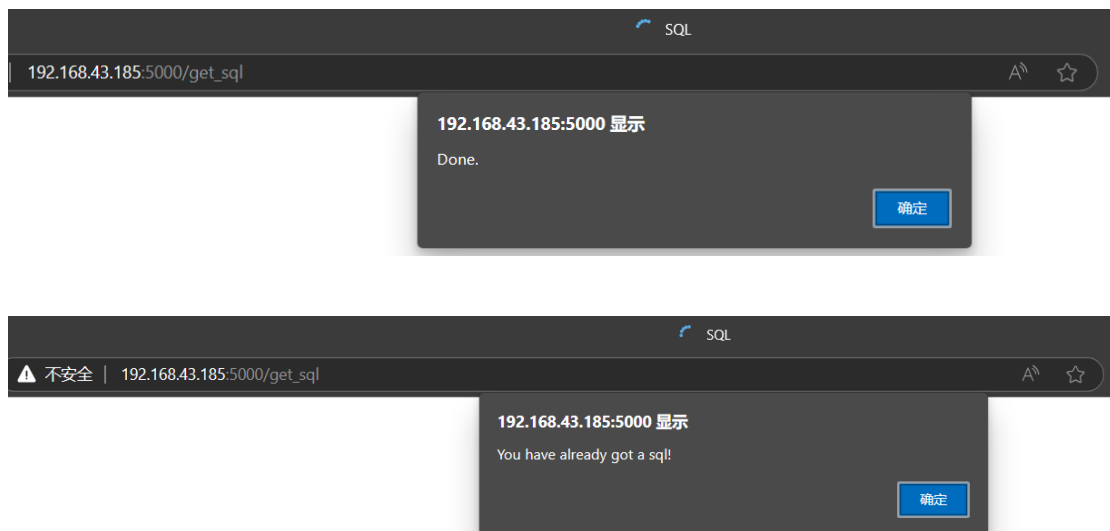
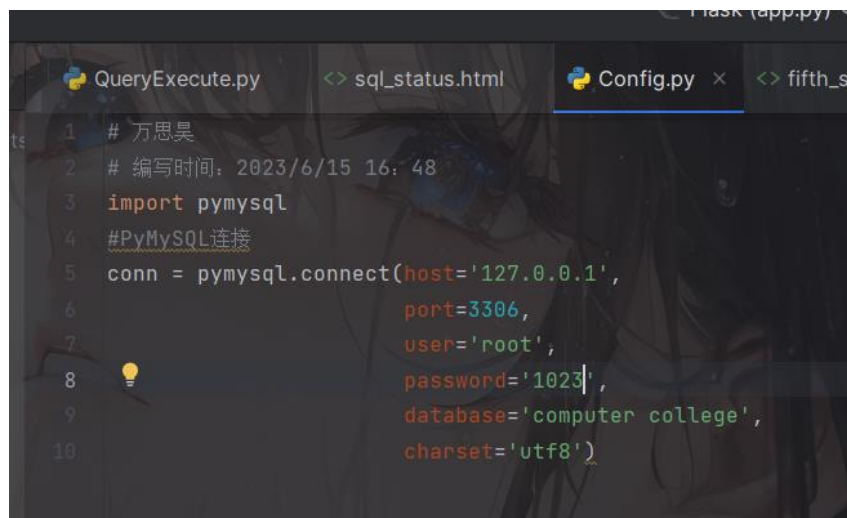


图 25 具体页面

7.6 其它



```
1 # 万思昊
2 # 编写时间: 2023/6/15 16: 48
3 import pymysql
4 #PyMySQL连接
5 conn = pymysql.connect(host='127.0.0.1',
6                         port=3306,
7                         user='root',
8                         password='1023',
9                         database='computer college',
10                        charset='utf8')
```

图 26 Config.py

调整 MySQL 密码和其它的相关内容。密码是我生日。

这些内容在 Config.py 内。

前端代码细节不表（也不是本课程重点），请阅读项目文件相关代码。

8 遇到的主要问题及解决方法

8.1 搜索结果有用数据的提取

由于 pymysql 执行 SQL 语句的搜索结果是将搜索的列表以元组的形式展示出来的，需要通过搜索工具将 SQL 语句的执行结果转化成我们所需要的数量和类型，首先对所需要的数据类型分类，针对所需要的几个类型构建 SQL 语句执行工具，将执行搜索出来的数据放在数据缓冲池中，供给前端模板使用。

8.2 Canvas 插件作为动态背景

将 Canvas 动态效果作为背景，如果不做任何处理可能会出现 HTML 部分被覆盖、HTML 不按照原来的位置展示等问题，这个时候我做一个透明的背景，将原来的 HTML 部分放在这个透明的背景上，然后将这块背景与 Canvas 结合，这个时候不做处理，可能会出现 Canvas 和 HTML 分离的情况，这个时候需要调整 canvas 的位置，将其 z 左标调整比 HTML 的透明背景板的 z 坐标小即可（以上所描述的这些问题我在实际操作中都有遇到）。

9 总结与展望

通过对本次智领未来平台的开发，我更加充分的理解了课本上的知识，能够将其加以扩展，从而应用于时间当中，达到了数据展示完整丰富，功能完备，安全可靠，具有设计价值和实际意义。

我自身是具有一定 Python 基础的，但是大二下学期我发现我很多基本知识都忘了（例如：python 的标准数据类型 API）。还好趁着程序设计课和对应的习题，我及时温习了基础知识。

我也很珍惜这一次大作业的机会，是除去考试周备考工作外的一次极好的练手机会。之前对嵌入式和人工智能比较熟练操作，但是对基于 Python 的 Web 开发还是了解太少。这次也算恶补了功课，为我暑假和大三的工程奠定基础。

我通过查阅资料的实际操作基本掌握了前端 HTML+CSS+JavaScript（jQuery）的框架组合，flask 和 MySQL 的连接和使用以及数据的调用和处理，这些内容对我开发知识有了加深和延拓，是一次宝贵的经验。

此次大作业构建的内容略大，牵扯了很多的知识点，必然会出现代码优化程度不足以及运行逻辑的冗余。此次大作业先做到这个阶段（因为大作业撰写中间还有一个考试周积压的我喘不过来）。等到 7 月第二轮考试周结束以后，可以在该项目的代码基础上继续优化，也可基于此延伸 JavaWeb 相关学习。

纸上得来终觉浅，绝知此事要躬行。理论是一成不变的，但是只有当它与实践相结合的时候，才能显示它真正的意义。