# 基于Python数据可视化的毕业生就业统计管理系统

**摘 要：**我国每年都有大量高校毕业生要走向各个城市的不同岗位，虽然有海量的招聘信息供准毕业生选择，但具体到某一高校的特定专业，其毕业生就业方向、各城市人数分布往往有其自身特点。如果利用传统数据表格的形式查看历届学生就业信息耗时费力也不直观，而利用数据可视化功能如饼状、柱状图等图形来展示，一目了然。

本文研究内容旨设计一款数据可视化的应用程序用于管理、统计分析高校毕业生的就业信息供高校准毕业生及相关工作人员使用。本系统基于django框架搭建，前端可视化页面使用pyecharts制作，python则作为后端语言处理数据的交互，引入pymysql模块处理后端与数据的交互，数据库管理系统选用开源免费的mysql。系统由数据处理和可视化展示两个模块组成，数据处理包括对学生的就业信息的添加、查询、删除和修改。可视化展示模块中有条形图、饼图与中国地图，条形图用以展示就业岗位分布人数；设计了两个饼图，分别表示为就业岗位完成百分比与具体岗位类别；设计了中国地图用以展示毕业生就业地域分布。

本文研究成果可以让准毕业生迅速了解到本专业近年主要就业分布情况，帮助他们高效投替简历，选择合适的工作岗位，具有较高的实用价值。

**关 键 词：**python；可视化；pyecharts；mysql

# Graduate Employment Statistics Management System Based on Python Data Visualization

**Abstract:**Every year in my country, a large number of college graduates want to go to different positions in various cities. Although there is a huge amount of recruitment information for prospective graduates to choose from, the employment direction of the graduates and the distribution of the number of graduates in each city are often specific to a specific major of a university. Its own characteristics. It is time-consuming and labor-intensive to view the employment information of previous students in the form of traditional data tables, and it is not intuitive, while using data visualization functions such as pie and bar graphs to display it at a glance.

The research content of this paper aims to design a data visualization application program to manage and statistically analyze the employment information of college graduates for use by high-calibration graduates and related staff. This system is built based on the django framework. The front-end visualization page is made with pyecharts, and python is used as the back-end language to process data interaction. The pymysql module is introduced to handle the interaction between the back-end and data. The database management system uses open source and free mysql. The system consists of two modules: data processing and visual display. Data processing includes adding, querying, deleting and modifying students’ employment information. There are bar graphs, pie graphs, and China maps in the visual display module. The bar graphs are used to display the number of jobs distributed; two pie charts are designed to indicate the percentage of job completions and specific job categories; the map of China is designed to be used To show the geographical distribution of graduate employment.

The research results of this article can allow prospective graduates to quickly understand the major employment distribution of their majors in recent years, help them to efficiently replace their resumes and choose suitable jobs, which has high practical value.

**Key words:**python; visualization; pyecharts; mysql

# 第一章 绪论

## 1.1研究背景

高校毕业生就业困难的原因大致可以分为三点，一是高校毕业生规模较大人数过多；二是经济下行，疫情爆发，在高校毕业生增多的同时，相应的岗位却不见得有增多；三是高校毕业生对自我能力与社会现状的认知不清晰。

高校毕业生的人数逐年上涨，每年在创新高，以近两年为例，2020年高校应届毕业生874万，2021年高校应届毕业生突破九百万，预计总规模可达到909万人，相对于2020年来说同比增长30多万人，再创历史记录。受新冠疫情的影响2020届毕业生仍有部分毕业生未就业，这部分未就业的毕业生也将加入竞争中，使就业竞争更加激烈。

随着疫情的几度爆发，暂未有完结的迹象，势必也将继续影响着2021届毕业生，甚至是往后好多届的毕业生，在经济下行的情况下高校毕业的的就业环境逐渐恶化，在加上疫情的影响，就业岗位总体来说只减不增，这又是高校毕业生就业的一大痛点。

中国式教育在学生上大学以前以考试成绩为第一要务，很少学生会去思考自己喜欢什么，擅长什么，未来想从事什么职业，也少有家长会对孩子的进行引导，大部分都是以家长的心愿来定未来。上大学以后，大多数学生迷失在了青春校园轻松欢快的学习氛围中，了然忘却四年后将要面临的是一年比一年竞争激烈的就业环境。在大学中没有练就满足行业要求的就业技能，对自身能力认知的不足且缺乏引导，刚出校门时的年少轻狂、眼高手低，也照成了高校毕业生期望职位与自身不匹配的现象。

## 1.2 现状分析和研究内容

综合上方背景调查，我们可以发现高校毕业生就业难的问题是可以进行分析、改善与缓解的。第一点：高校毕业生多；第二点：社会的就业岗位少，第三点：高校毕业生自身能力不足，对自我认知不清晰，对社会的就业环境缺乏了解。第一点与第二点更多的是要政府等专业人士进行出谋划策与改变，对于第三点社会各界也作出的许多贡献与改变，比如学校针对毕业生就业问题专门开设了就业指导课程，在毕业之前各学院也有专业实习、校企合作等，派学生到企业去进行短期实习，提前了解该专业未来将要面临的工作环境与该行业所需的专业技能。而第三点正是本篇论文要着手的地方，本论文使用python设计了一个可视化的毕业生就业统计管理系统，该系统对高校各专业学生的就业情况进行分析，以便加强高校毕业生的自我认知，增强他们对社会就业环境的了解，使他们能更早的觉悟，选择一个适合自己、且方便就业的行业，同时加强自身能力以应对未来的竞争。

本论文的研究内容为设计可视化系统用以分析高校毕业生的就业趋向，毕业后各个就业方向学生的就业成功率，就业成功的毕业生中从事的各行业的毕业生人数占比，毕业后就业地域分布情况等。

一、分析高校毕业生毕业后的选择趋向，高校毕业生的毕业后选择大致可以分为如下四种：企业就业、升学、考编、从军，其中考编包含了公务员、有事业编制较为稳定的国企单位，通过分析高校毕业生们毕业后的选择可以大致了解社会环境，如果高校毕业生选择毕业后就业的人数逐年增多则说明社会就业环境较好，高校毕业生选择毕业后结业的人数逐年减少，而选择升学、考编、从军的人数逐年增多则说明社会就业环境逐渐恶化，更多的人选择了升学来提升自身的知识价值以应对恶劣的就业环境甚至于有部分人是想通过升学来暂时逃避恶劣的就业环境，在就业环境恶化的同时，大家也更趋向于找一份稳定的公务员岗位或是有事业编制单位谋生。

二、分析高校毕业生毕业后选择企业就业、升学、考编、从军这四个方向的成功率有多少，从中可以得到更多的信息，就业成功率可以看出就业市场是否激烈，考研成功率可以看出各高校的升学竞争是否激烈，考编成功率可以看出公务员、国企等事业编制的单位考试是否激烈、从军成功率可以看到应聘到军队是否激烈。

三、分析高校毕业生选择就业与考编的人数中从事各行业的毕业生人数占比，通过观察哪些行业的人数占比逐年增加便可得知哪些行业哪些岗位容易就业且前景较好，通过观察哪些行业的人数占比逐年减少便可得知哪些行业哪些岗位容易就业且前景较差，提供这些信息给高校毕业生们，供他们对未来作出更好的选择。

四、分析高校毕业生毕业后去往的城市分布情况等信息，可以知道哪个城市对高校毕业生的吸引最强，哪个城市更有就业优势。且对于不同省份的高校来说，偏向去往的城市也有所区别，可以得出哪所城市某高校的校友最多，对于同一所学校的毕业生来说校友更多的城市可以得到更多的帮助，更有优势。

## 1.3本文的组织结构

本文共分为五章，内容安排如下：

第一章：绪论。介绍了本文研究的问题、研究背景与研究现状。

第二章：关键技术与平台搭建。介绍了如何搭建本系统的运行环境。

第三章：系统设计。介绍了本系统的架构模块与使用流程。

第四章：系统实现。展示了本系统的运行界面并介绍了实现原理。

第五章：总结与展望。总结了本系统的优缺点并阐明改进方向。

# 第二章 关键技术与平台搭建

本系统使用django框架搭建，使用python3.6编写，其中主要调用了pymysql、xlrd、pyecharts等包，pymysql用于操作mysql数据库，xlrd用于导入excle表格中的数据，pyecharts用于生成可视化页面。

## 2.1 python

本设计采用python语言开发的，它是面向对象的一种解释型计算机的程序设计语言，优点在于拥有丰富和强大的库和模块，可快速开发系统，语法简单学习成本低，java等多数语言语言使用的{}来控制类，函数等其他逻辑判断，而python使用缩进来控制。

python是当下的热门语言，目前最新版本为3.9。为与其他模块结合，考虑兼容性减少因版本问题导致的bug,所以本方案使用的版本为3.6，也与接下来所使用的python第三方库有着更好的兼容性。

可以直接到python官网下载exe文件进行安装指定版本的python,若是系统空间较大也可安装Anaconda，Anaconda是一个开源的Python发行版本，包含180多个科学包及其依赖项。

python官网：<https://www.python.org/>

Anaconda官网：<https://www.anaconda.com/>

在本系统中使用的python模块及其版本

1. pymysql

python在python2中连接 MySQL 服务器使用的是mysqldb，但在python3中连接 MySQL 服务器使用的是pymysql。本系统使用的python版本是pyhton3故安装pymysql，使用命令pip install pymysql。

1. xlrd

python操作excle主要用到xlrd和xlwt这两个库，即xlrd是读excel，xlwt是写excel的库,本系统的批量录入学生信息功能是使用excle表格导入的，故需要安装xlrd模块。

目前最新的xlrd只支持.xls文件。若是使用.xlsx后缀的excle表格，pandas.read\_excel(‘xxx.xlsx’)会报错：xlrd.biffh.XLRDError: Excel xlsx file; not supported，本系统需要使用的就是1.2.0版本的xlrd，该版本可以正常导入.xlsx后缀的excle表格，安装1.2.0版本的xlrd模块命令为pip install xlrd==1.2.0。同时这里需要注意的是，待上传的表格需要放在网站的根目录下，如：C:\Users\Administrator\Desktop\VSS。

## 2.2 pyecharts库

pyecharts是echarts 与python 的对接，可以使用 pyecharts 生成独立的网页，也可以在flask，Django 等框架中集成使用，pyecharts可流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等）。

Pyecharts中包含了众多图表，例如Bar 、Bar3D、Boxplot、EffectScatter 、Funnel、Gauge、Geo、Graph、HeatMap 、Kline、Line、Line3D、Liquid、Map、Parallel、Pie、Polar、Radar、Sankey、Scatter、Scatter3D 、ThemeRiver、WordCloud等。

指定安装1.9.0版本的pyecharts：pip3 install pyecharts==1.9.0

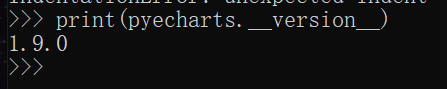


图2.1 pyecharts版本图

以bar图表为例，导包代码为：from pyecharts.charts import Bar

Pyecharts的github地址：<https://github.com/pyecharts/pyecharts>

Ptecharts的官方文档地址：<https://pyecharts.org/#/>

## 2.3 mysql数据库

MySQL是由瑞典的 MySQL AB 公司开发、发布并支持的数据库系统，它的象征符号是一只名为 Sakila 的海豚，代表着 MySQL 数据库的速度、能力、精确和优秀本质。它具有安全、跨平台、高效等优点，并且能与 PHP、Java 等主流编程语言紧密结合的。

本论文采用数据库部分是由python使用pymysql库与mysql结合，使用的mysql版本为5.7.26，如果电脑内存较小，想要快速方便一些的话可以使用phpstudy，他是mysql与多个服务的集成环境，可到官网下载安装包安装好后，使用navicate进行连接操作更方便。

## 2.4 Django

django框架是一款使用python语言开发的web网站模板，可学习他的规则，再编写部分代码，采用类似“填空”或“修改”的方法将一个网站编写成本设计所需要的模样。Django使用WSGI通信协议，MTV模型组织，能实现大部分web应用的相关功能。

本设计采用的是django2.2版本，Django2.2适配的python版本为3.5, 3.6, 3.7，而本系统采用的是python3.6可以完美结合Django2.2。

Django2.2的官方文档为：https://docs.djangoproject.com/zh-hans/2.2/

本系统使用django搭建本网站的操作如下：

（1）安装django

Windows上安装django：pip3 install Django==2.2

（2）使用django创建项目和APP

使用django创建名字为VSS的项目：django-admin startproject VSS

使用django创建名字为vsapp的APP：python manage.py startapp vsapp，

创建完APP后在settings.py里注册它，把它加到INSTALLED\_APPS里去。可以选择直接添加APP名字。

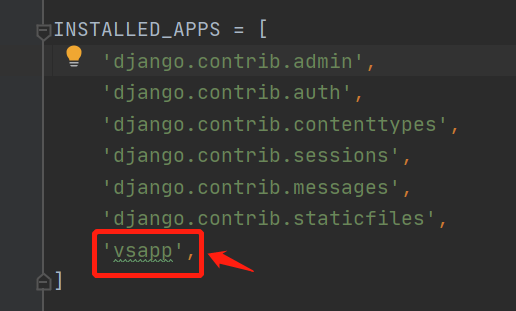


图2.2 INSTALLED\_APPS配置图

1. 数据库的配置

搭建好数据库环境，将数据库命名为students,在settings.py中进行配置，如下图所示，在DATABASES={}中需配置数据库的账号密码、主机ip、端口号以及该系统的数据库名等。

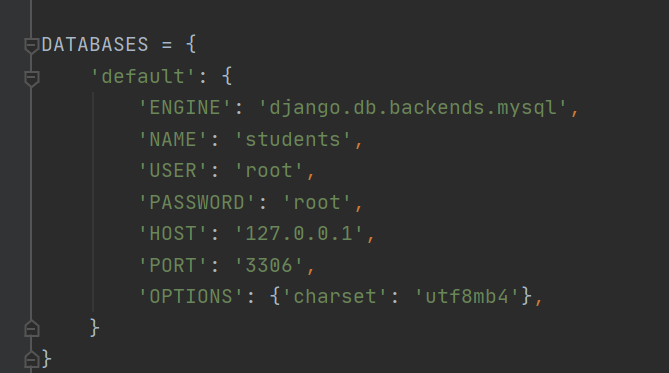


图2.3 settings.py配置图

1. 配置models.py

models.py是django与数据库之间的桥梁，model封装与定义了信息。它包含需要存储的字段和操作数据的方法，每个模型都映射了一张数据表，为Django提供了一个访问数据库的API。每个model都是继承于django.db.models.Model 的Python类，model的一个类对应数据库的一个表，一个属性对应数据表中的一个字段。

如图2.4所示为本系统中models.py的配置

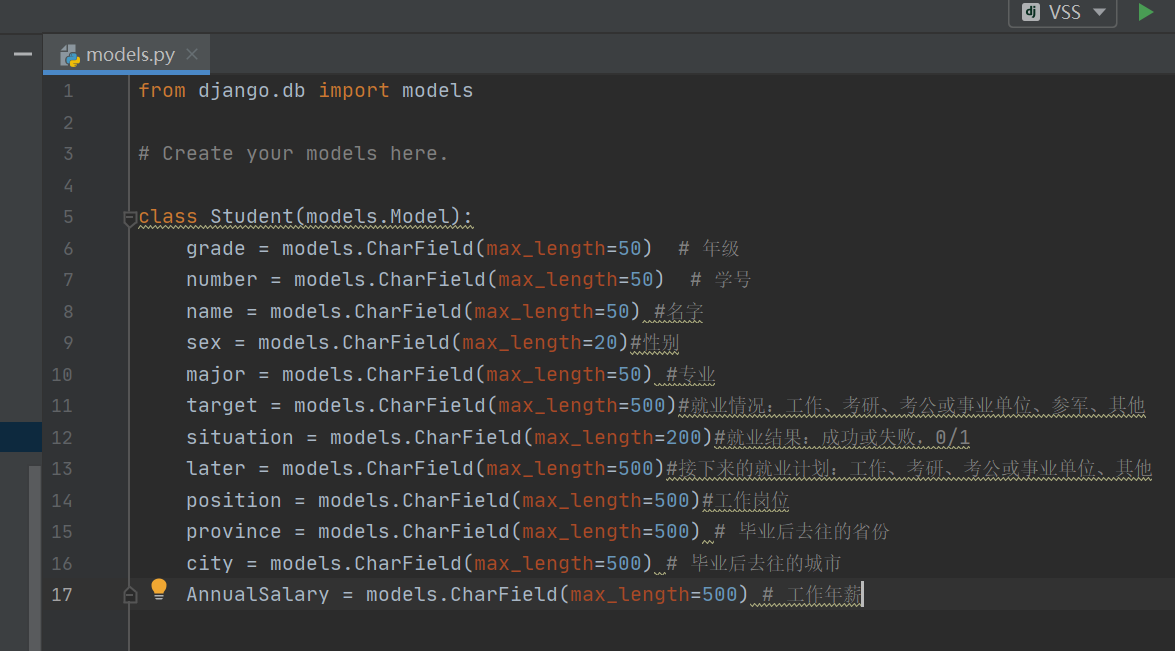


图2.4 models.py配置图

配置完models.py后需要对数据进行迁移，迁移命令如下

生成迁移文件：（根据模型类生成sql语句）python manage.py makemigrations

执行迁移：（执行sql语句生成数据表）python manage.py migrate

（5）iframe框架的配置

运行pyechart代码后会生成可视化的页面或图片，在网站上展示这些可视化页面有多种方法，本系统采用的方式是使用ifram框架。

实现过程以条形统计图为例：

调用bar( )函数，其中.render( )将生成的图形化页面指定位置在C:\Users\Administrator\Desktop\VSS\vsapp\templates\visual\bar\_is\_selected.html，然后在barVisval.html页面中使用代码<iframe frameborder="0" width="70%" height="600px" src="/bar\_is\_selected"></iframe>，即可将bar\_is\_selected.html可视化页面嵌套进barVisval.html 页面进行展示，具体流程见下方截图

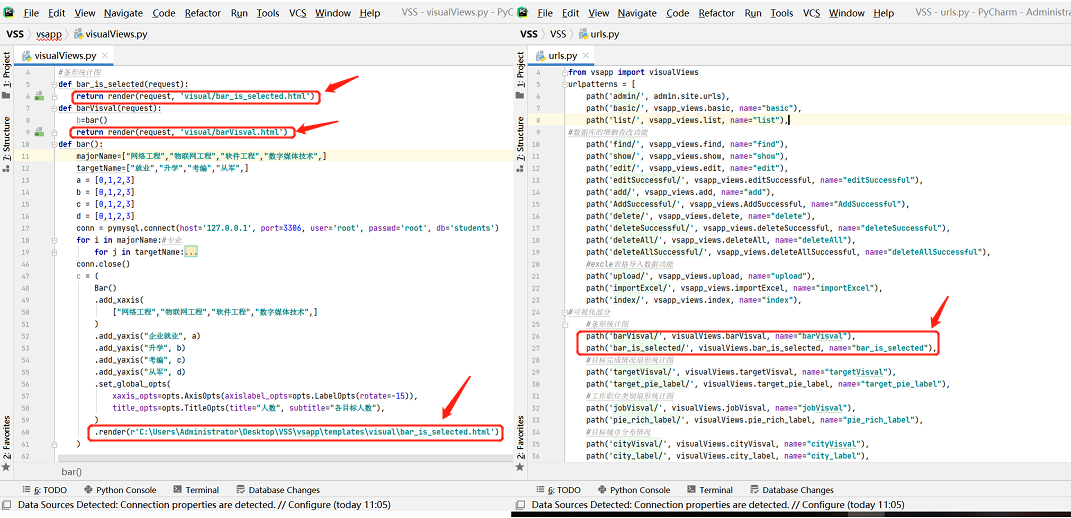


图2.5 models.py配置图

django中默认是不使用iframe标签的，所以需要在django中进行配置，如下截图所示在global\_settings.py中要修改代码为将X\_FRAME\_OPTIONS = 'DENY'X-Frame-Options修改为X\_FRAME\_OPTIONS = 'SAMEORIGIN'，只有开启X-Frame-Options，使用django框架开发的网站才能用iframe标签。

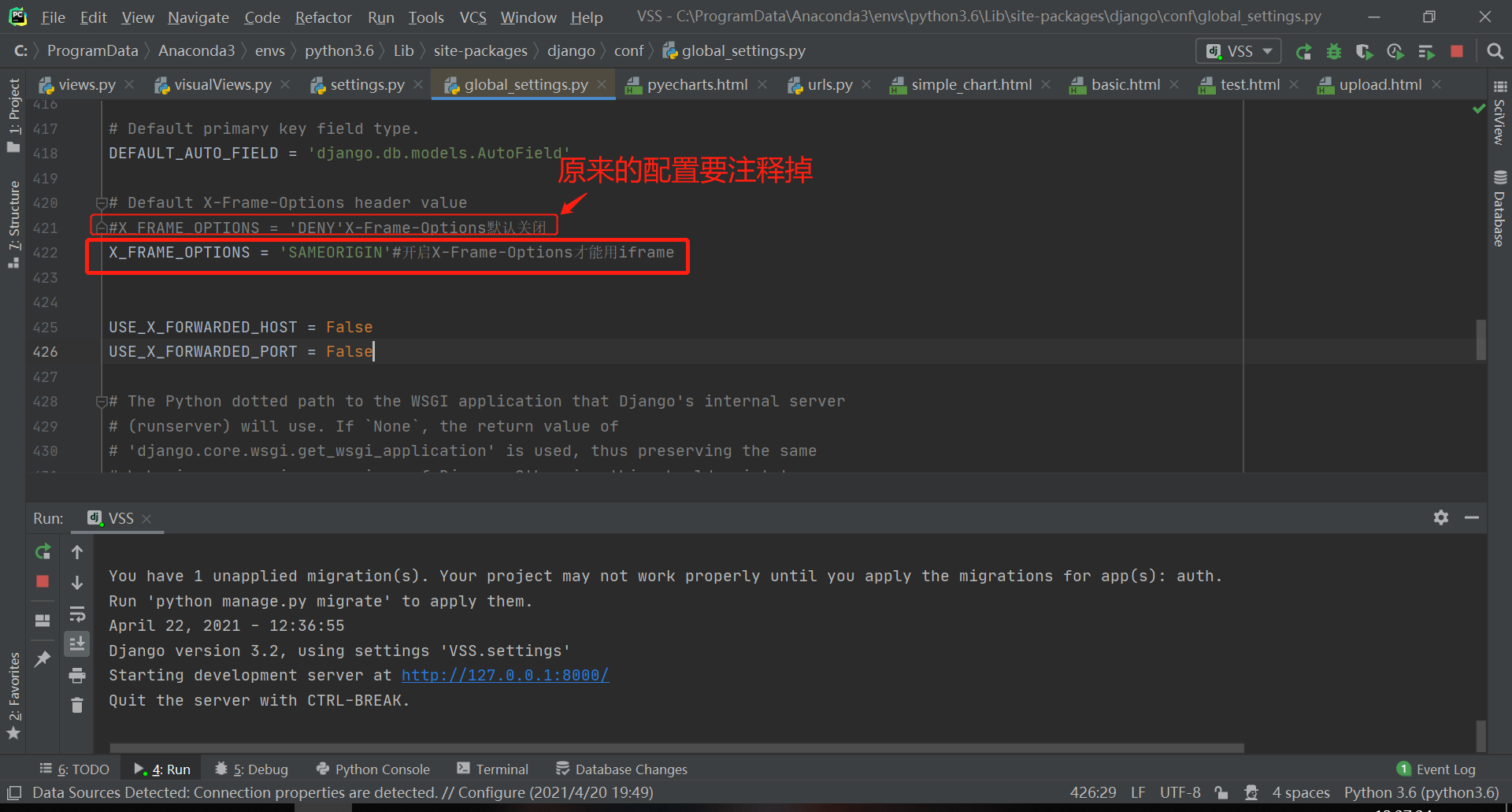


图2.6批量删除数据成功提示页面图

（6）运行Django项目，默认端口8000：python manage.py runserver ip:port

# 第三章 系统设计

## 3.1可行性分析

使用python语言开发，python不仅简单易学而且有丰富的模块和包，本系统的可视化界面就是使用python调用了pyecharts包实现的，pyecharts生成的可视化页面图形众多，能很好的起到分析数据的作用。目前需要录入的学生信息数据量不大，故选用搭建方便且开源免费的mysql数据库，有pymysql模块的存在能与python完美结合，综上分析此方案可行。

## 3.2需求分析

### 3.2.1可视化页面需求

要通过高校毕业生的就业趋向，毕业后选择各岗位方向学生的就业成功率，就业成功的毕业生中从事的各行业的毕业生人数及其占比，毕业后去往的城市分布情况等数据进行分析，需要设计四个可视化页面，这四个页面需要具有的功能如下：

1. 使用条形图为统计各专业在各就业岗位上的人数分布情况
2. 使用饼图展示毕业生在各就业岗位中成功就业的百分比
3. 使用饼图展示成功就业的毕业生从事的具体岗位有那些类别，各岗位具体人数及其占比
4. 使用中国地图展示毕业生的就业地域分布

### 3.2.2 数据库操作需求

这四个可视化页面的数据徐要从数据库中读取，这对数据库的操作需要具有如下功能：

1. 添加数据：添加单条数据、使用excle表格批量导入数据
2. 查询数据
3. 编辑数据
4. 删除数据：删除单条数据、批量删除数据

数据库E-R图

E-R图如图3.1下所示。

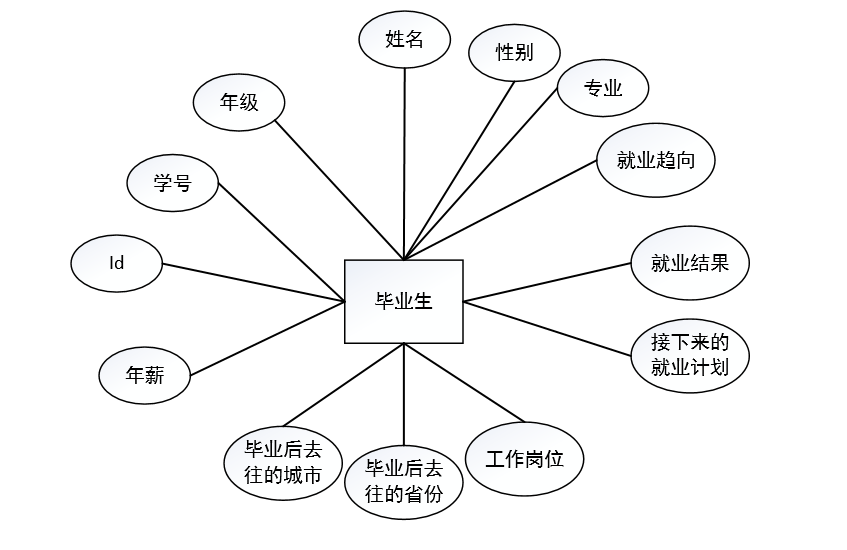


图3.1 E-R图

本项目的主要实体为毕业生，毕业生实体具有的属性有id、学号、年级、姓名、性别、专业、就业趋向、就业结果、接下来的就业计划、工作岗位、毕业后去往的省份、毕业后去往的城市、年薪。

## 3.3详细设计

### 3.3.1架构图

如下图所示为本系统的B/S架构图，B是browse的缩写指的是浏览器，S是server的缩写指的是服务器，用户在浏览器端进行操作，通过http协议提交数据到服务器，服务器接收到数据后又通过各种方式与数据库交互，之后将处理结果以html页面展示在浏览器上。

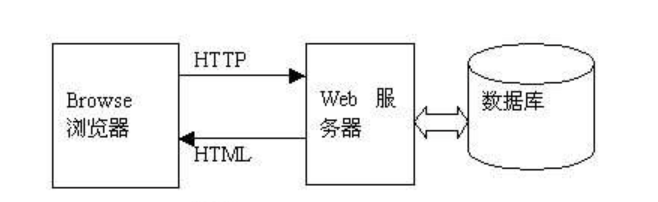


图3.2 架构图

### 3.3.2系统流程设计

系统操作流程图如下图所示，从图3.3中可以了解到访问系统，添加数据、编辑数据、查看可视化页面、删除数据等详细操作流程。

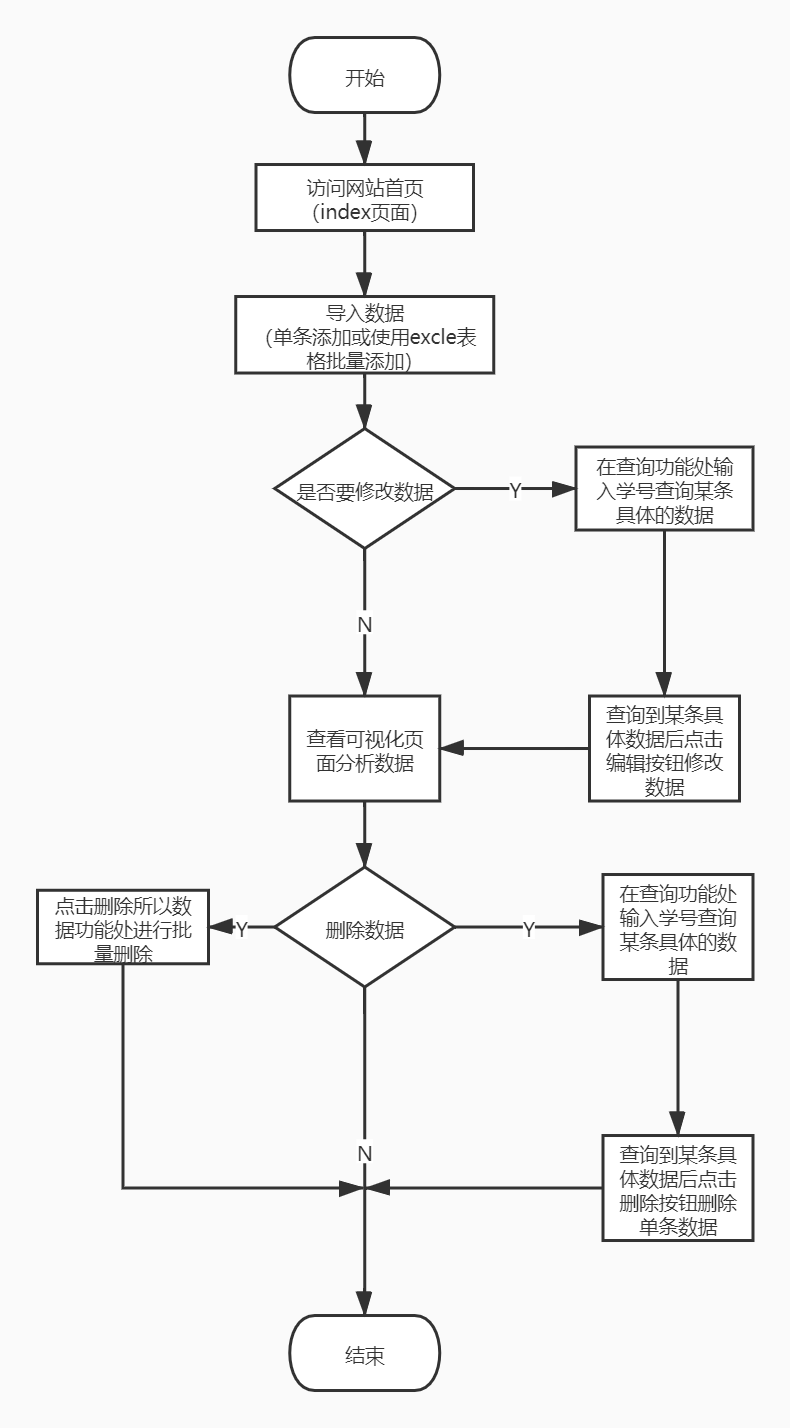


图3.3系统的流程图

### 3.3.3模块设计

系统的模块功能如图3.4所示，从功能需求出发制作前端可视化页面与数据库操作页面。

图3.4中的数据库处理各模块对应的数据库的操作页面分别为：添加数据、excle表格导入数据、毕业生就业信息列表、查询编辑或删除、删除所有数据。

“添加数据”为手动填写添加学生数据的页面，excle表格导入数据为导入excle表格批量添加数据的页面。这里也需要补充一下，手动添加单条数据、excle表格批量导入数据、编辑数据、手动删除单条数据的时候都会有个操作成功的提示页面。毕业生就业信息列表用于展示数据库中的数据，仅展示最后7条，因为展示的再多也没有意义，如果要查看具体的某条数据，一页页的翻页查看不现实。对于查询、编辑、删除单条数据的需求，可以从“查询编辑或删除”这个功能点下手，查询编辑或删除”这个功能点是根据学号查询某条学生信息的，然后可以选择性的对该数据进行修改或删除数据。“删除所有数据”为批量删除功能，这个功能点没有具体的页面，只是一个点击的功能点但是在批量删除成功后会有一个提示删除成功的页面。

图3.4中的可视化展示各模块对应的前端可视化页面分别为：首页、就业类别人数、完成情况、职位类别、城市分布。

其中首页是一个欢迎界面、这不多做解释，剩下的四个可视化页面才是本系统的重点，这四个页面中就业类别人数、完成情况、职位类别、城市分布则分别对应于高校毕业生的就业趋向，毕业后各个选择方向学生的就业成功率，就业成功的毕业生中从事的各行业的毕业生人数占比，毕业后去往的城市分布情况。

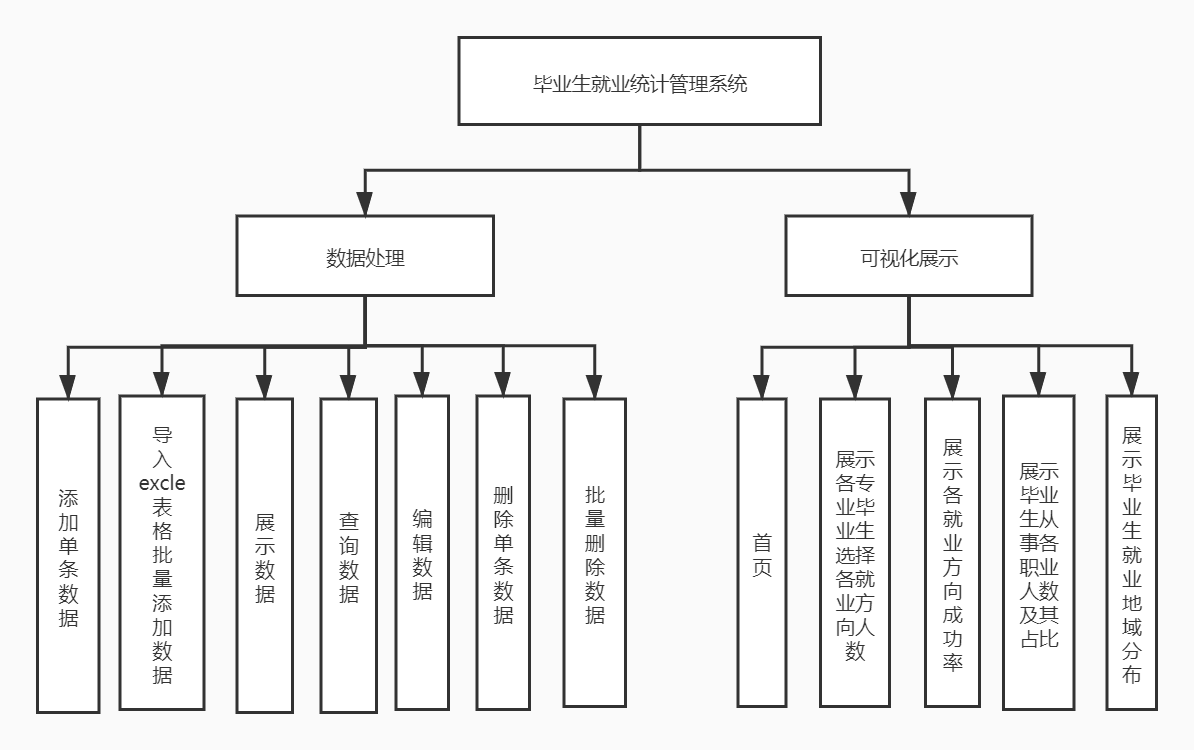


图3.4 模块功能图

|  |  |
| --- | --- |
| 模块名称 | 添加单条毕业生数据 |
| 功能描述 | 此模块实现了在数据库中的添加单条数据的操作 |
| 执行步骤 | 1、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。   1. 接收需要添加到数据库中的内容   通过request.POST.get()函数获取前端通过post方式传过来的毕业生数据。  3、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句insert into将接收到的数据插入数据库中，在执行完添加操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。  4、关闭数据库连接  使用cursor.close()函数关闭指针对象，再使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 导入excle表格批量添加毕业生数据 |
| 功能描述 | 此模块实现了通过导入excle表格对毕业生信息进行批量添加的操作 |
| 执行步骤 | 1、打开excle表格  使用request.FILES[]获取导入的excle表格文件名，使用xlrd.open\_workbook()函数打开表格，使用book.sheet\_by\_name()函数打开表格中的指定sheet页  2、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。   1. 创建for循环迭代读取excle表格中的数据   使用for r in range(1, sheet.nrows)进行迭代读取，其中sheet.nrows指表格的行数，sheet.cell(r, n).value指r行中的第n列数据,本系统中都r和n都从1开始计数用以跳过标题行和序号行。  4、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句insert into将接收到的数据插入数据库中，在执行完添加操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。重点说明一下，此功能点sql语句的执行需放在for循环中，每迭代读取一行excle表格中的数据就执行一次sql语句。  5、关闭数据库连接  使用cursor.close()函数关闭指针对象，在使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 查询毕业生数据 |
| 功能描述 | 此模块实现了对数据库中毕业生信息的查询操作 |
| 执行步骤 | 1、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。  2、接收需要添加到数据库中的内容  通过request.POST.get()函数获取前端通过post方式传过来的数据，此功能点是以学号为查询索引，所以从前端接收的数据即为学号。  3、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句select x,x from vsapp\_student where number =n在数据库中对学号为n的毕业生信息进行查询，在执行完添加操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。   1. 在前端展示从数据库中查询到的数据   使用cursor.fetchall()读取sql语句执行的结果，通过代码m\_data = cursor.fetchall()将执行结果赋值给m\_data,然后使用render()函数将m\_data数据渲染到前端进行展示。  5、关闭数据库连接  使用cursor.close()函数关闭指针对象，再使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 编辑毕业生数据 |
| 功能描述 | 此模块实现了对毕业生信息的修改操作 |
| 执行步骤 | 1. 先通过查询学号等到某条毕业生数据，然后点击编辑按钮才能进行编辑，毕业生数据只能进行单条编辑。 2. 将需要编辑的毕业生信息展示在前段   毕业生的属性较多，先将所有属性展示在前端用户只需挑选需要修改的属性进行编辑  3、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。  4、获取编辑后的毕业生数据  通过request.POST.get()函数获取前端通过post方式传过来的毕业生数据，接收以id属性为key和索引，因为id属性每个学生都有一个，这个属性不进行展示也不允许修改，这样即使用户修改的数据为学号，之后也可通过新学号查询修改后的毕业生数据。  5、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句update vsapp\_student set x,x,x where id=n更新数据库中id属性为n的毕业生数据，在执行完修改操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。  6、关闭数据库连接  使用cursor.close()函数关闭指针对象，在使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 删除单条毕业生数据 |
| 功能描述 | 此模块实现了对单条毕业生数据的删除操作 |
| 执行步骤 | 1. 先通过查询学号等到某条毕业生数据，然后点击编辑按钮才能进行单条删除。   2、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。  3、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句DELETE FROM vsapp\_student WHERE number=n删除数据库中id属性为n的毕业生数据，在执行完删除操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。  4、关闭数据库连接  使用cursor.close()函数关闭指针对象，在使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 批量删除毕业生数据 |
| 功能描述 | 此模块实现了对毕业生信息的批量删除操作 |
| 执行步骤 | 1、建立数据库连接  pymysql.connect()中填入数据库的ip、端口、账号、密码、数据库名称进行连接，conn.cursor()获取指针对象。  2、执行数据库操作  在cursor.execute()函数中传入需要被执行的sql语句，使用sql语句DELETE FROM vsapp\_student进行批量删除，在执行完批量删除操作后,需要调用一下conn.commit()方法进行提交，只有调用了conn.commit()方法数据才会真正保存在数据库中。  3、关闭数据库连接  使用cursor.close()函数关闭指针对象，在使conn.close() 关闭连接对象。 |

|  |  |
| --- | --- |
| 模块名称 | 就业岗位人数分布统计图 |
| 功能描述 | 从数据库中获取各专业在各就业岗位上的人数分布情况，数据经过处理后通过条形统计图可视化页面展示在前端 |
| 执行步骤 | 1、根据需求执行sql语句取得需要的数据  执行sql语句SELECT count(\*) FROM vsapp\_student where major=m and target=n，获取vsapp\_student表中专业为m,岗位为n的人数。  2、在.add()内填入需要展示的数据信息  .add\_xaxis()中填入条形统计图的横坐标，横坐标为专业，  .add\_yaxis()中填入条形统计图的纵坐标名称及其数值，纵坐标名称为岗位，数值就是选择该岗位为就业方向的学生人数。  3、使用.render()函数生成可视化页面到指定目录下  以下方为例，是生成条形图可视化页面到电脑的具体目录下，在python字符串中\有转义的含义，为了在windows系统当中读取文件路径可以使用\,于是在路径前面加r，即保持字符原始值。  .render(r'C:\Users\Administrator\Desktop\VSS\vsapp\templates\visual\bar\_is\_selected.html')  4、前端页面使用iframe标签，嵌套指定目录下的可视化页面进行展示 |

|  |  |
| --- | --- |
| 模块名称 | 就业岗位完成百分比 |
| 功能描述 | 从数据库中获取毕业生在各就业岗位中成功就业人数，经过计算得到成功就业人数的百分比，数据经过处理后通过饼图可视化页面展示在前端 |
| 执行步骤 | 1、根据需求执行sql语句取得需要的数据  执行sql语句SELECT count(\*) FROM vsapp\_student where target=n and situation='成功'，获取vsapp\_student表中趋向的就业岗位为m,就业成功的人数。  2、在.add()内填入需要展示的数据信息  .add\_xaxis()中填入饼图的分类及数值，可以绘制多个饼图，只需在center列表设定图例位置，radius设定饼图的内外半径大小，将多个饼图的位置摆放好。  3、使用.render()函数生成可视化页面到指定目录下  以下方为例，是生成条形图可视化页面到电脑的具体目录下，在python字符串中\有转义的含义，为了在windows系统当中读取文件路径可以使用\,于是在路径前面加r，即保持字符原始值。  .render(r'C:\Users\Administrator\Desktop\VSS\vsapp\templates\visual\target\_pie\_label.html')  4、前端页面使用iframe标签，嵌套指定目录下的可视化页面进行展示 |

|  |  |
| --- | --- |
| 模块名称 | 具体岗位类别 |
| 功能描述 | 从数据库中获取就业的毕业生从事的具体岗位有那些类别，这些具体类别的岗位分别有多少毕业生从事，的经过处理后通过饼图可视化页面展示在前端 |
| 执行步骤 | 1、根据需求执行sql语句取得需要的数据  执行sql语句SELECT count(\*) FROM vsapp\_student where position=x,获取vsapp\_student表中从事具体岗位为x的人数。  2、在.add()内填入需要展示的数据信息  .add\_xaxis()中填入饼图的分类及数值，可以绘制多个饼图，只需在center列表设定图例位置，radius设定饼图的内外半径大小，将多个饼图的位置摆放好。  3、使用.render()函数生成可视化页面到指定目录下  以下方为例，是生成条形图可视化页面到电脑的具体目录下，在python字符串中\有转义的含义，为了在windows系统当中读取文件路径可以使用\,于是在路径前面加r，即保持字符原始值。  .render(r'C:\Users\Administrator\Desktop\VSS\vsapp\templates\visual\pie\_rich\_label.html')  4、前端页面使用iframe标签，嵌套指定目录下的可视化页面进行展示 |

|  |  |
| --- | --- |
| 模块名称 | 毕业生就业地域分布 |
| 功能描述 | 从数据库中获取毕业生分布在全国各城市的人数情况，经过处理后通过中国地图可视化页面展示在前端 |
| 执行步骤 | 1、根据需求执行sql语句取得需要的数据  执行sql语句SELECT count(\*) FROM vsapp\_student where city=c,获取vsapp\_student表中就业城市为x的人数。  2、在create\_charts()配置需要展示的数据信息  通过Geo(）函数设置设置图表大小与主题风格，  在.add\_schema(）内设置地图类型、背景颜色与边框颜色，  在.add()内填入城市名称及各分布在各城市的毕业生人数。  3、使用.render()函数生成可视化页面到指定目录下  以下方为例，是生成条形图可视化页面到电脑的具体目录下，在python字符串中\有转义的含义，为了在windows系统当中读取文件路径可以使用\,于是在路径前面加r，即保持字符原始值。  .render(r'C:\Users\Administrator\Desktop\VSS\vsapp\templates\visual\city\_label.html')  4、前端页面使用iframe标签，嵌套指定目录下的可视化页面进行展示 |

### 3.3.4数据库设计

数据库表结构

在students数据库中存储毕业生信息的的核心数据表为vsapp\_student，他共有13个字段，各字段的具体信息见表3.1。

表3.1 表vsapp\_student的结构

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 字段名称 | 数据类型 | 字段含义 |
| 1 | id | int(50) | 数据编号 |
| 2 | grade | char(50) | 年级 |
| 3 | number | char(50) | 学号 |
| 4 | name | char(50) | 名字 |
| 5 | sex | char(20) | 性别 |
| 6 | major | char(50) | 专业 |
| 7 | target | char(500) | 就业趋向 |
| 8 | situation | char(200) | 就业结果 |
| 9 | later | char(500) | 接下来的就业计划 |
| 10 | position | char(500) | 工作岗位 |
| 11 | province | char(500) | 毕业后去往的省份 |
| 12 | city | char(500) | 毕业后去往的城市 |
| 13 | AnnualSalary | char(500) | 年薪 |

# 第四章 系统实现

## 4.1毕业生数据录入

### 4.1.1手动添加单条数据

手动添加单条数据可以在导航类的毕业生数据出和毕业生信息列表功能处看到，如图4.1与图4.2所示。

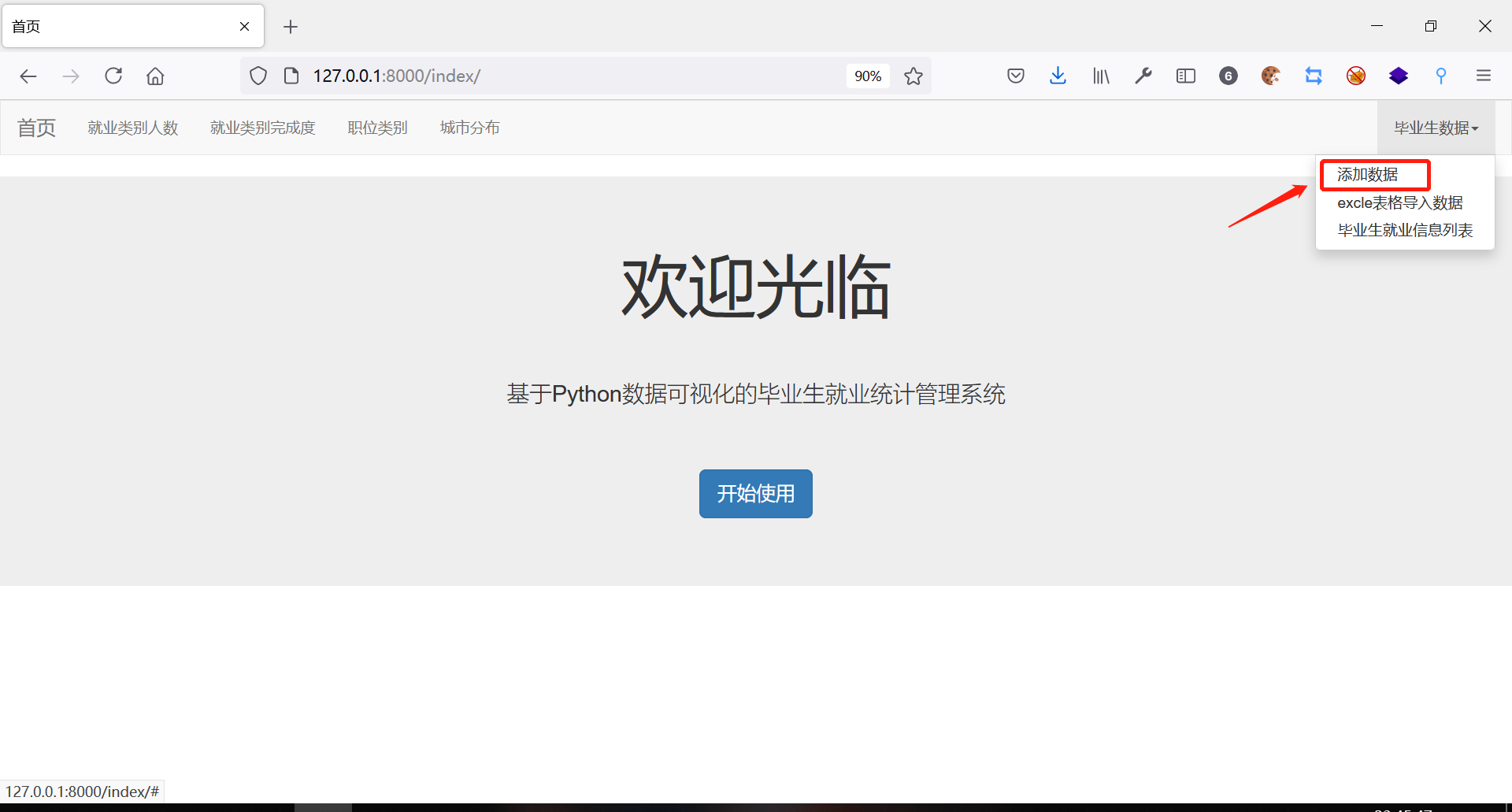


图4.1 毕业生数据处手动添加单条数据图

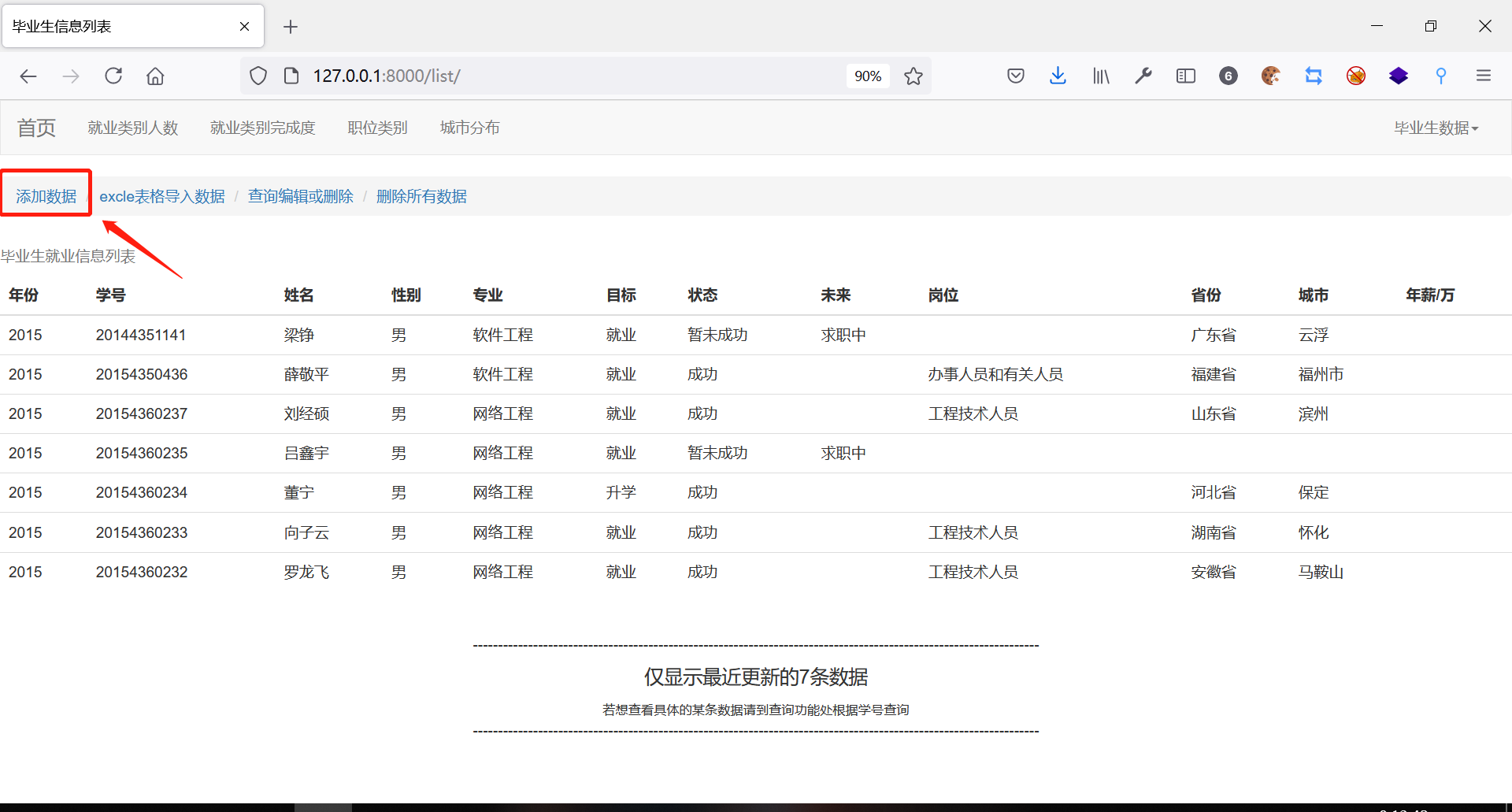


图4.2毕业生信息列表处手动添加单条数据图

手动添加单条数据的界面，如图4.3所以，可以填写毕业年份、学号、姓名、性别、专业、目标、状态、未来计划、工作岗位、目标地点、年薪，其中目标地点是由一段js代码实现的，可以将省份与城市进行二级联动。

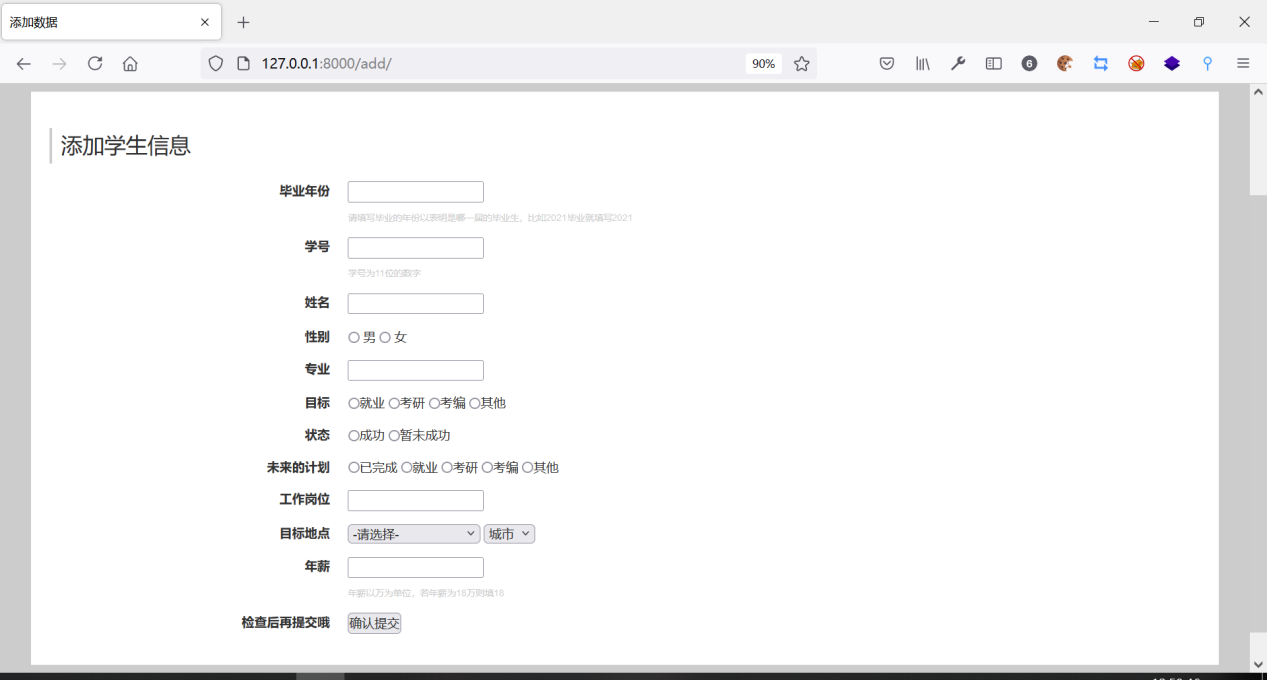


图4.3毕业生信息列表处手动添加单条数据图

手动添加单条数据的代码实现如图4.4所示，我们将add页面中post提交的数据传到后端，后端接受到数据后使用 cursor.execute()函数执行sql语句，将我们提交的数据添加到数据库中。

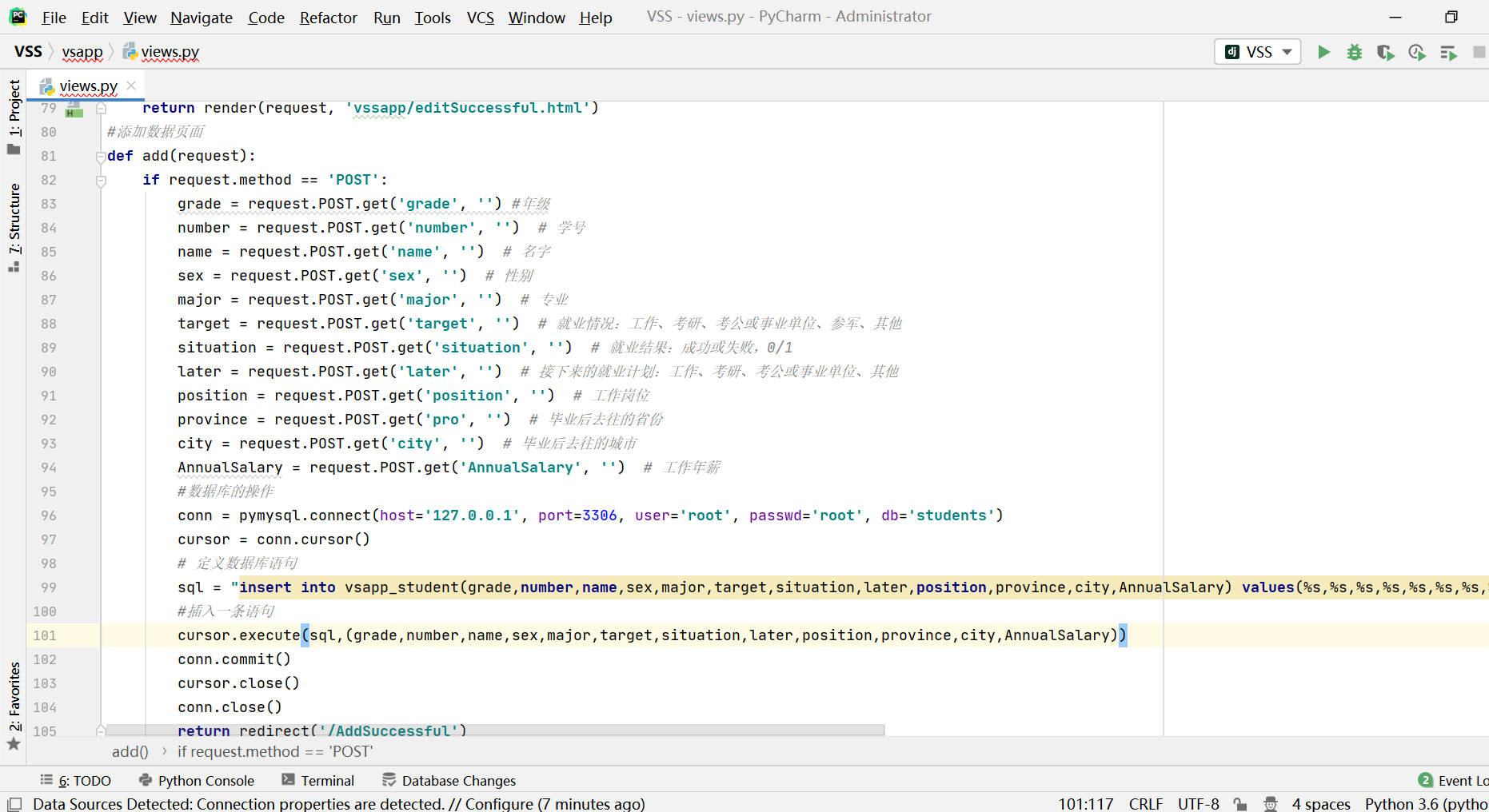


图4.4手动添加单条数据代码实现图

### 4.1.2使用excle表格批量添加数据

使用excle表格导入数据然后再批量添加的页面如图4.5、图4.6、图4.7、图4.8所示。

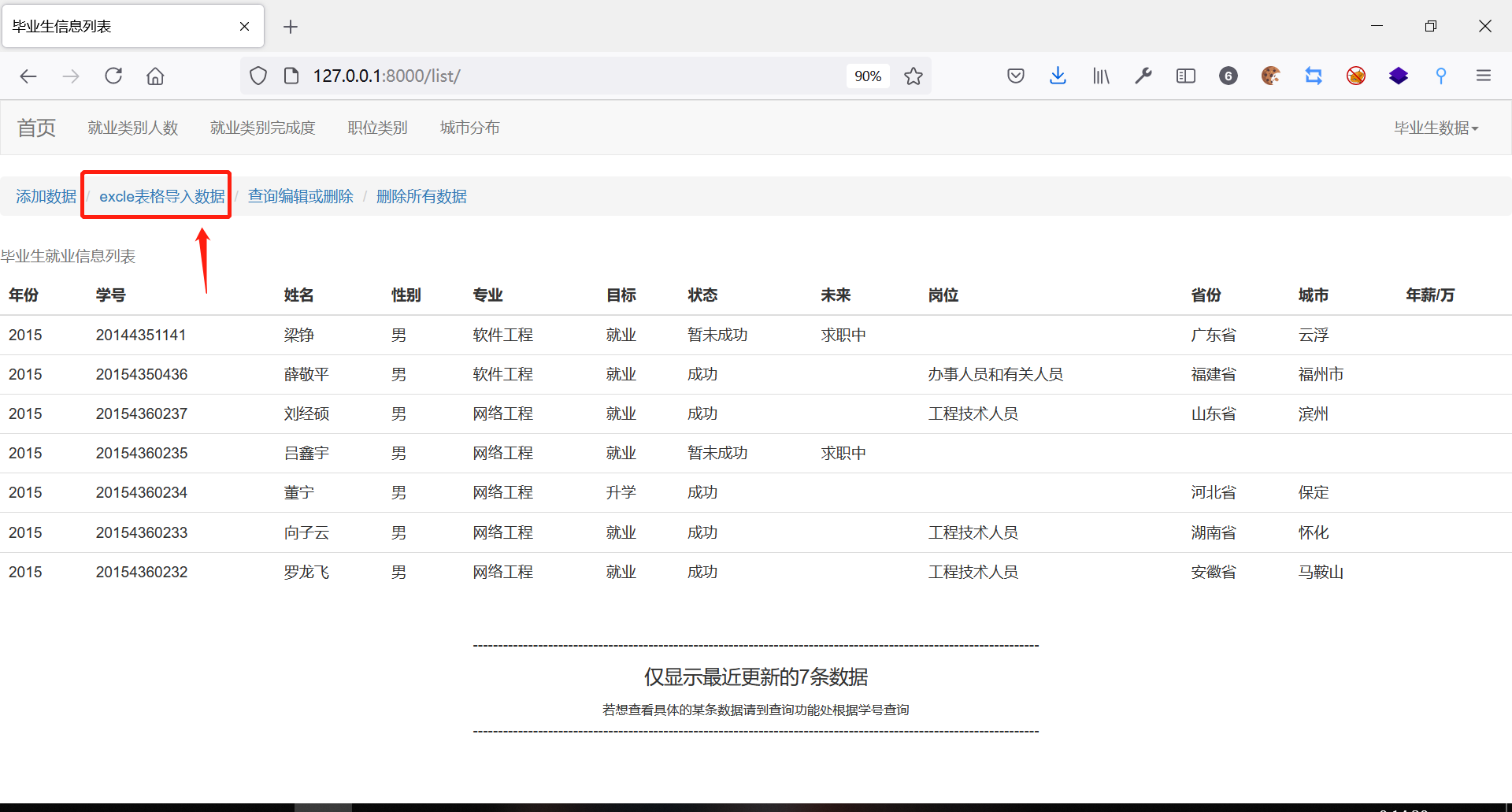


图4.5 excle表格导入数据功能点图

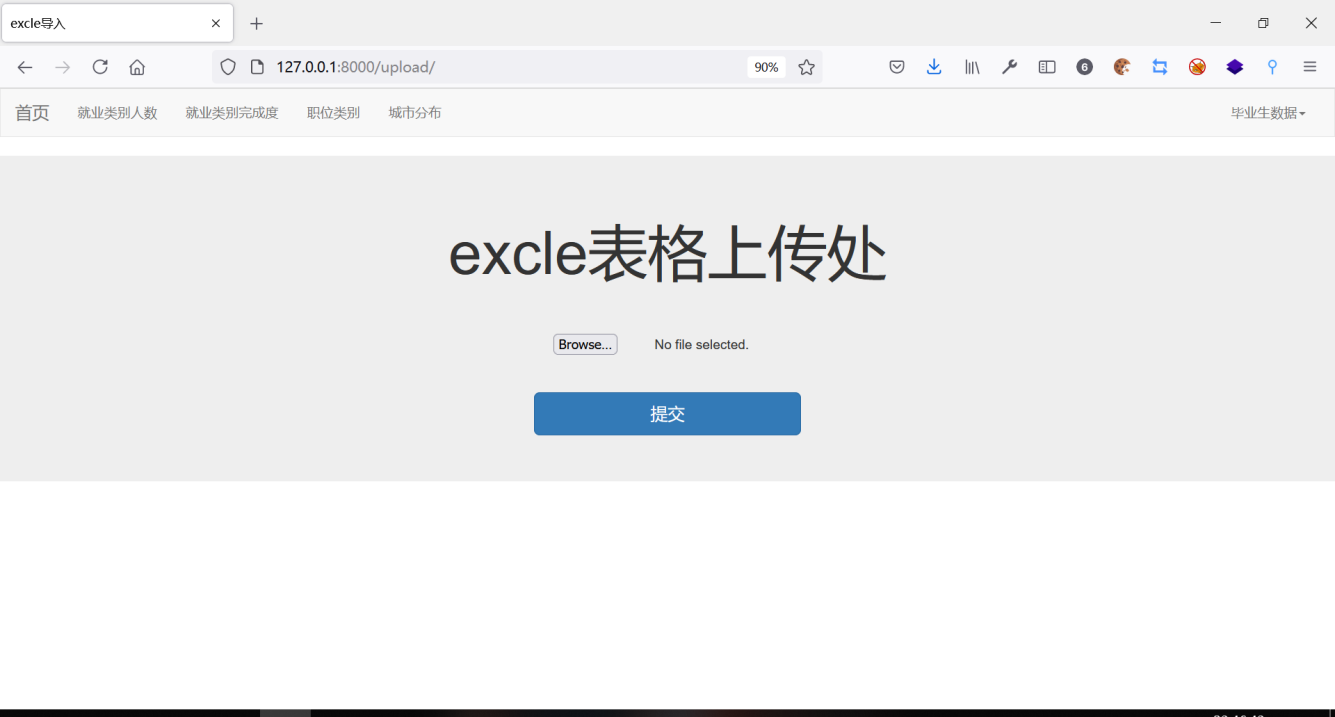


图4.6表格上传页面图

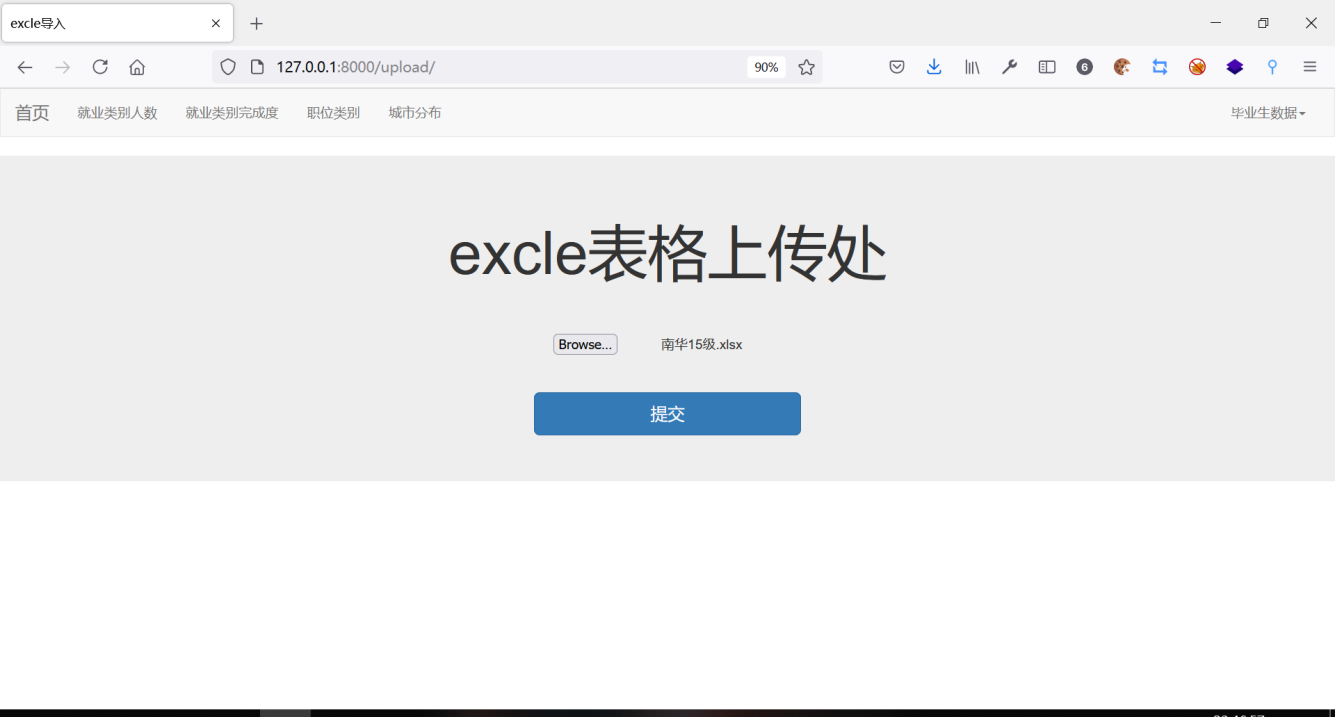


图4.7选择表格上传页面图

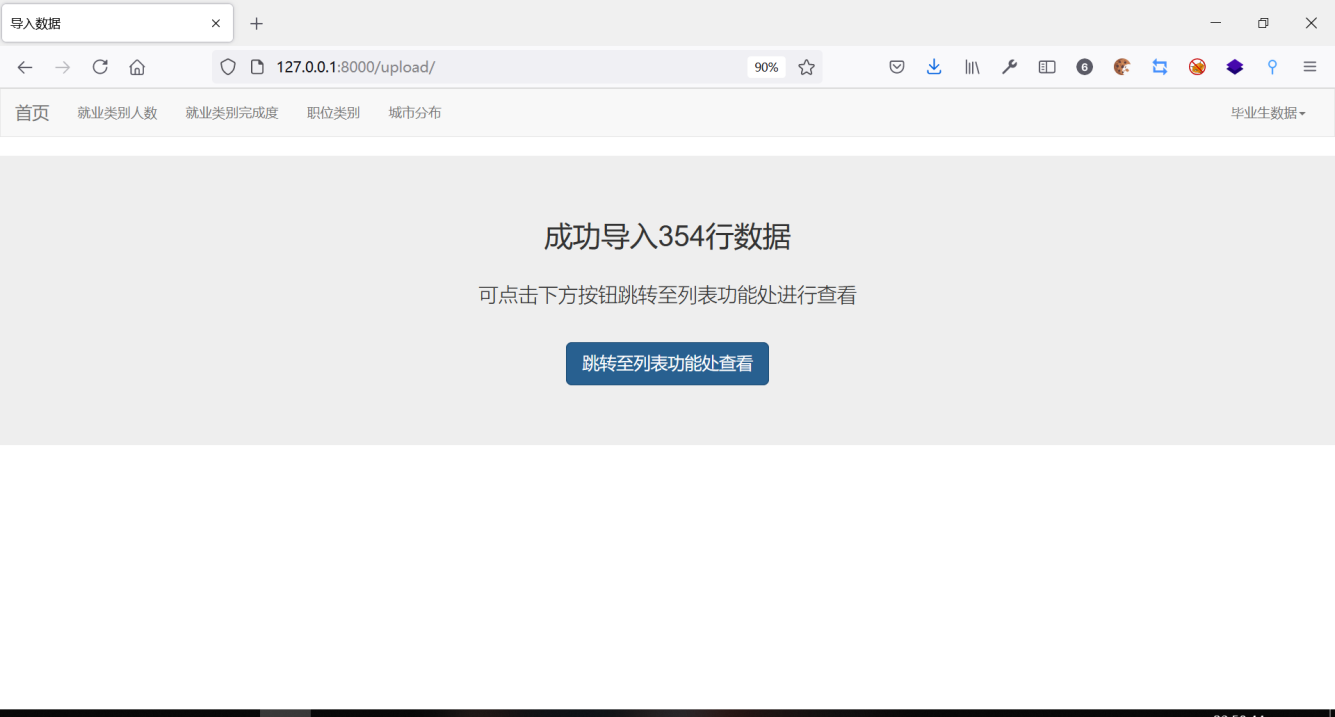


图4.8上传的表格成功导入数据图

使用excle表格批量导入数据的代码实现如图4.9所示，实现的逻辑可以总结如下：

1. 我们通过fn = request.FILES['file']获得上传的文件名，并将fn强制转换为str类型然后赋值给filename。
2. 得到文件名后通过xlrd.open\_workbook()函数打开excle文件，然后通过book.sheet\_by\_name()函数打开Sheet1表。
3. 接着通过一个for循环读出表中的每一行数据并将读出的数据写入数据库中。

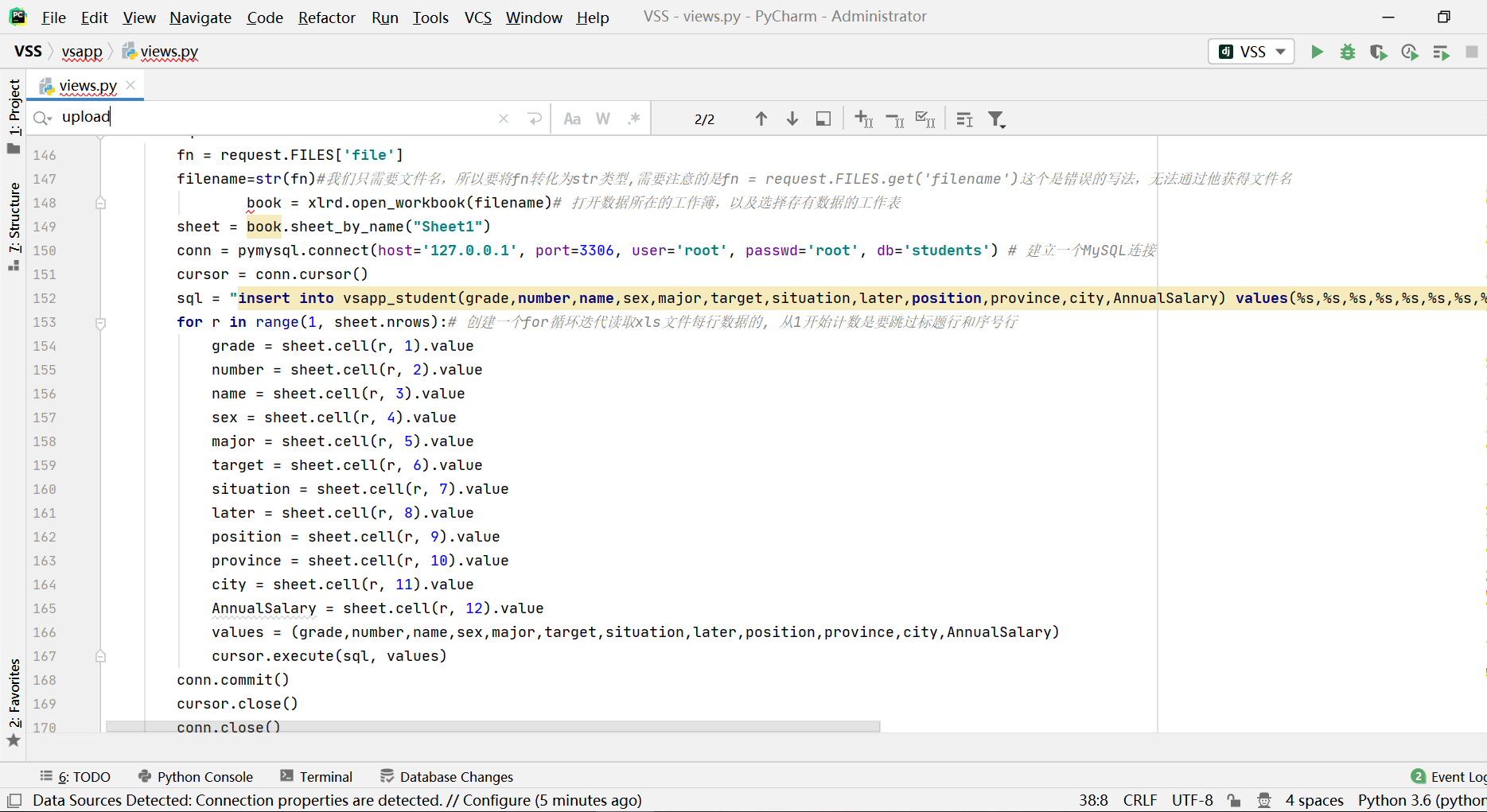


图4.9上传的表格导入数据代码实现图

## 4.2毕业生数据查询与修改

如图4.10、图4.11所示为根据学号查询的功能点

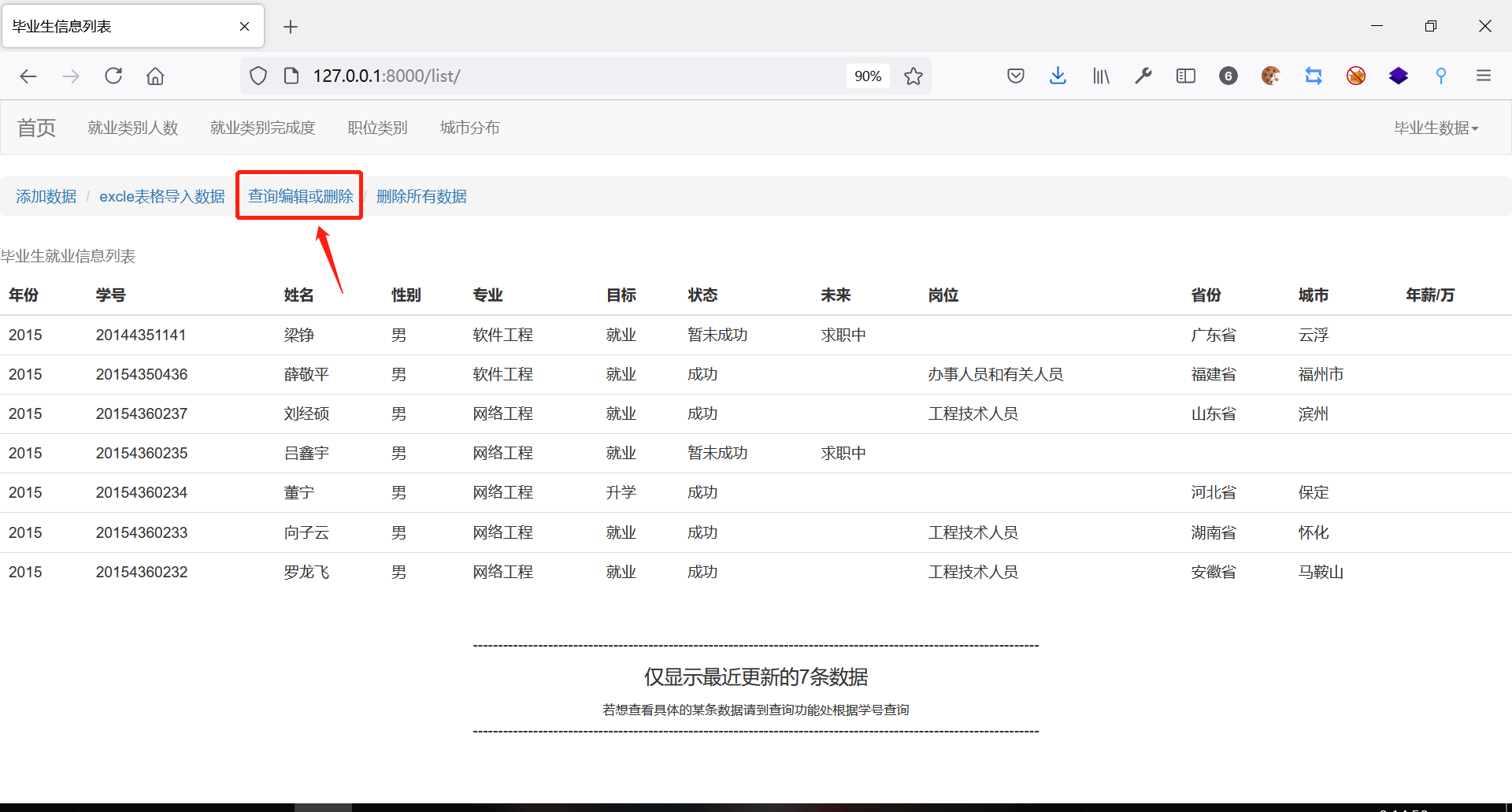


图4.10查询编辑或删除功能点图

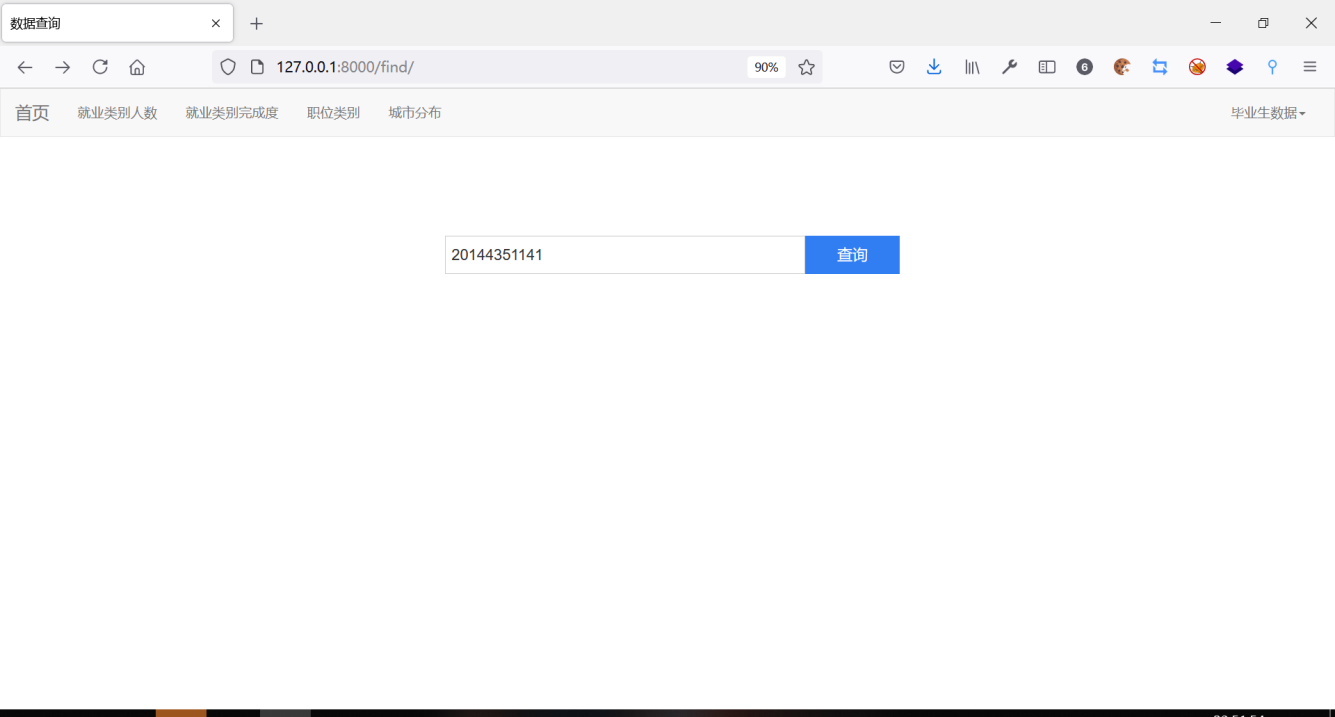


图4.11输入学号查询功能点图

查询功能的代码实现为使用cursor.execute()函数执行select id,grade,number,name,sex,major,target,situation,later,position,province,city,AnnualSalary from vsapp\_student where number = %s语句。

编辑数据功能点如下图4.12所示所示，需要输入学号查询到数据后才能对该条数据进行修改或单条删除。



图4.12查询结果图

点击编辑按钮后会逐条展示出该毕业生的信息，然后再根据需求进行修改，修改成功后我们可根据学号查询修改后的新信息如图4.13所示

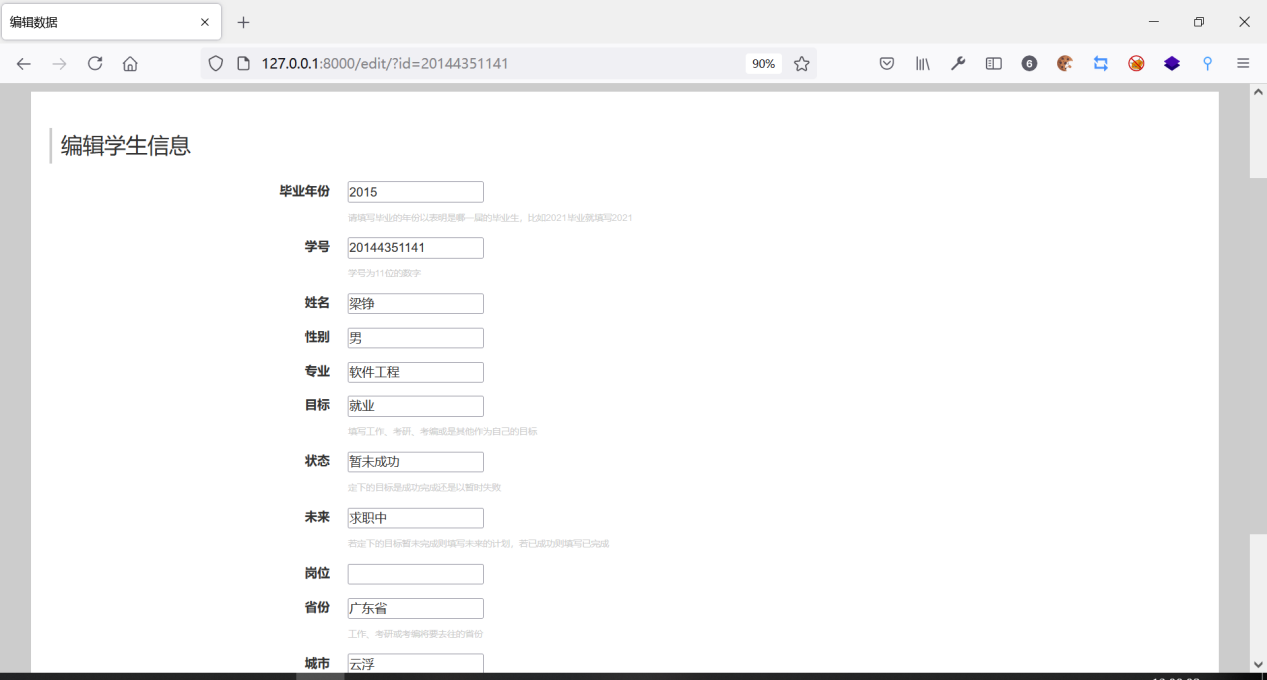


图4.13编辑页面图

编辑功能的代码实现如4.14图所示，和手动填加单条功能类似，在打开编辑页面的时候会把毕业生数据传递到前端页面展示，修改完数据后在使用post提交到后端，后端再将修改后的数据使用update语句更新数据库中该毕业生的数据信息

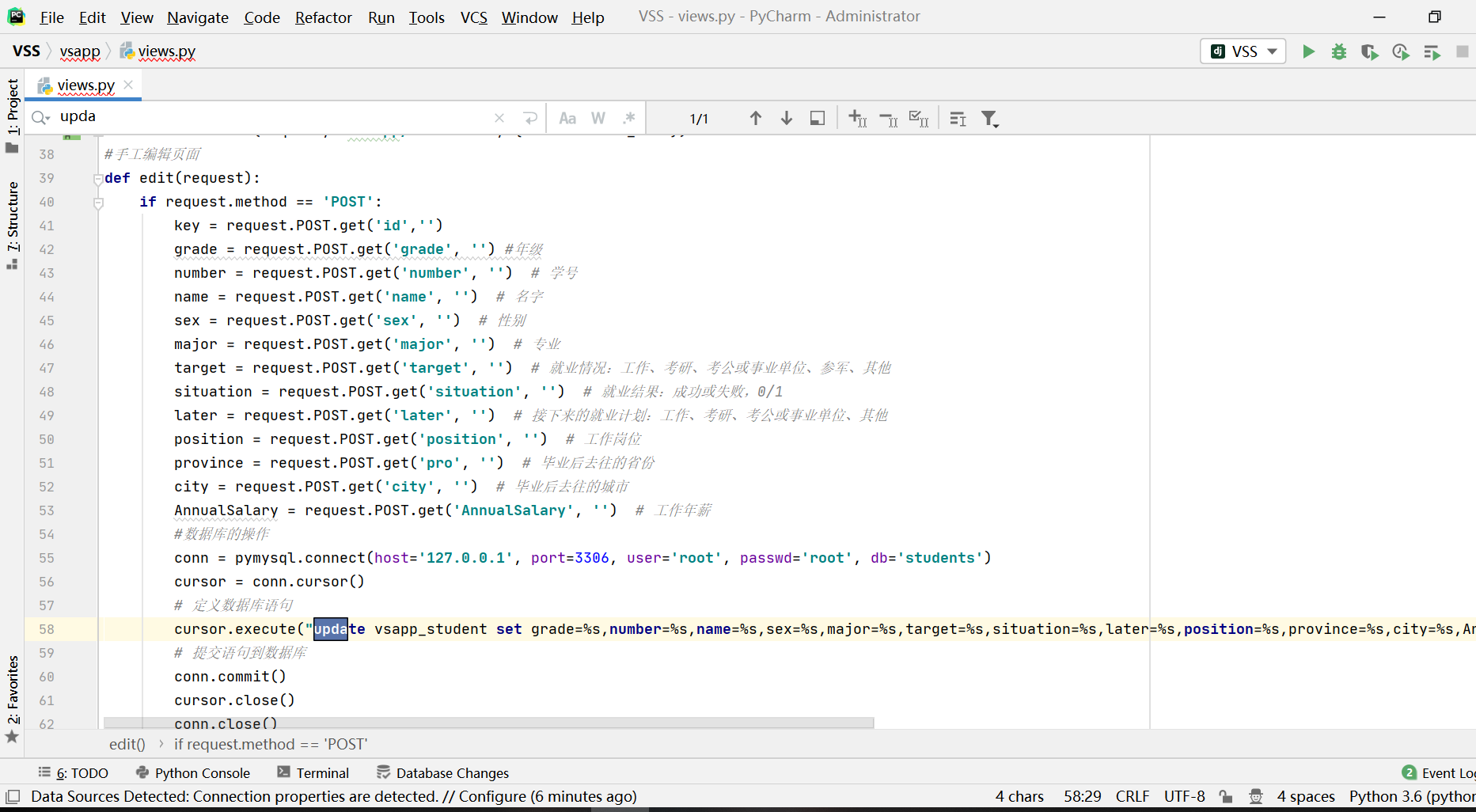


图4.14编辑数据代码实现图

编辑成功后我们点击确认修改的按钮，然后再查询界面输入学号查询就可以看到我们修改后的信息了，查询功能处虽然是以学号为索引进行查询，但在编辑信息时，我将id作为索引，这样即使修改了学号也不会扰乱我们的逻辑，若是学号被修改，我们只需输入修改后的新学号进行查询即可，如图4.15所示。



图4.15以id为索引图

## 4.3毕业生数据删除

### 4.3.1手动删除单条数据

输入学号查询到数据后才能对该条数据删除，删除单条数据功能的代码实现为使用cursor.execute()函数执行DELETE FROM vsapp\_student WHERE number=%s语句,点击删除按钮后会有删除成功的页面提示，如图4.16所示。

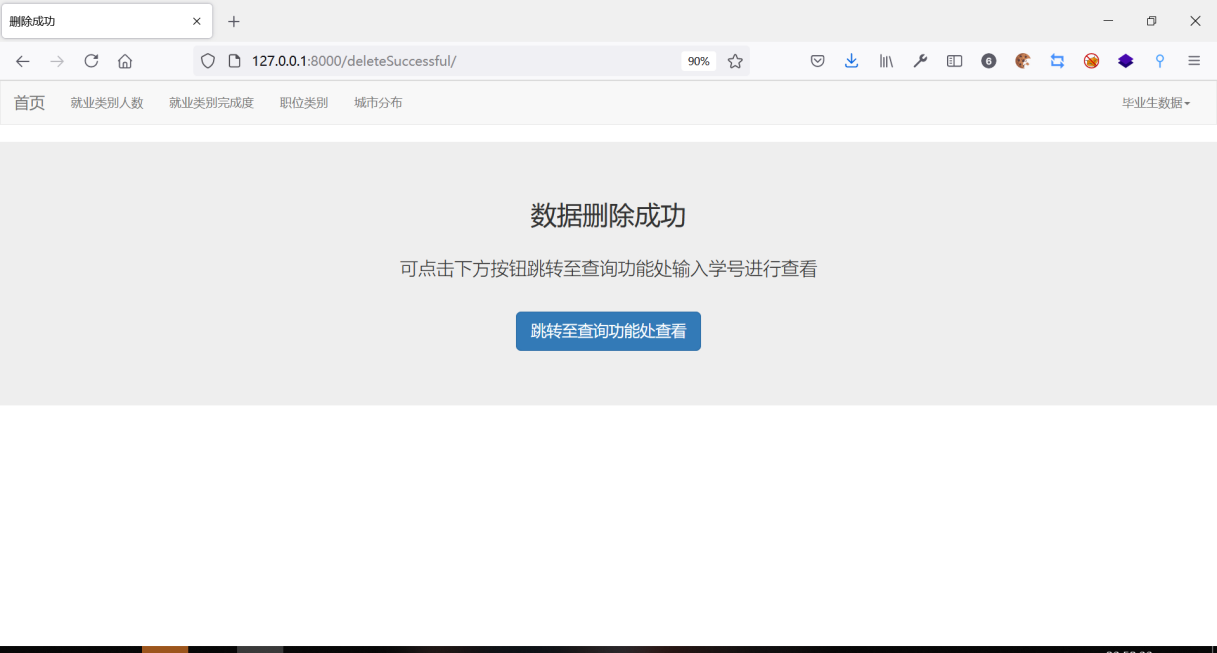


图4.16手工删除成功图

### 4.3.2批量删除数据

批量删除数据的功能点所在位置如图4.17、图4.18所示，点击批量删除数据后会有一个批量删除成功的提示界面，可以从该界面跳转至功能列表处，也可以通过导航栏点击跳转到其他功能点。

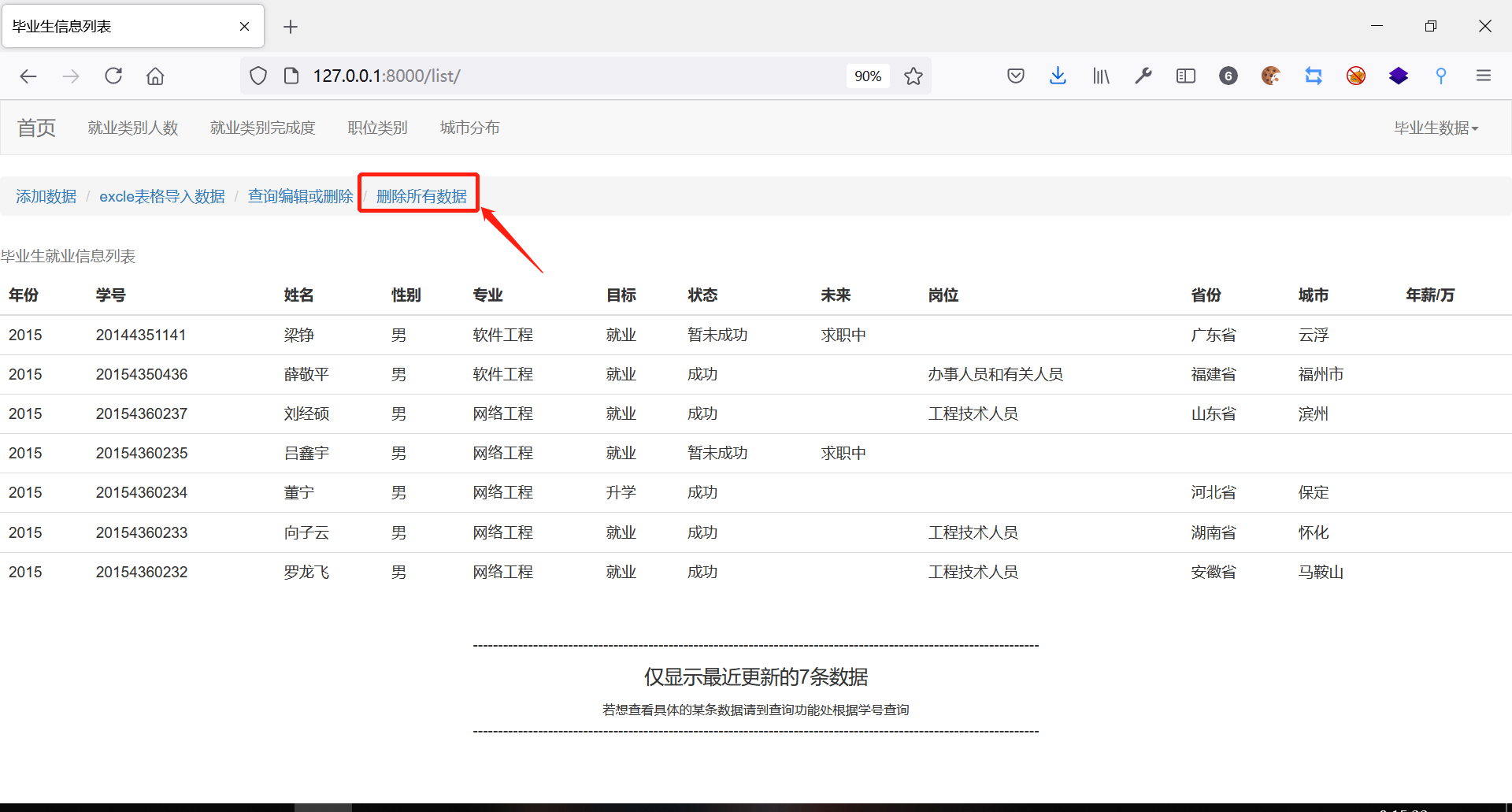


图4.17批量删除数据功能点所在位置图

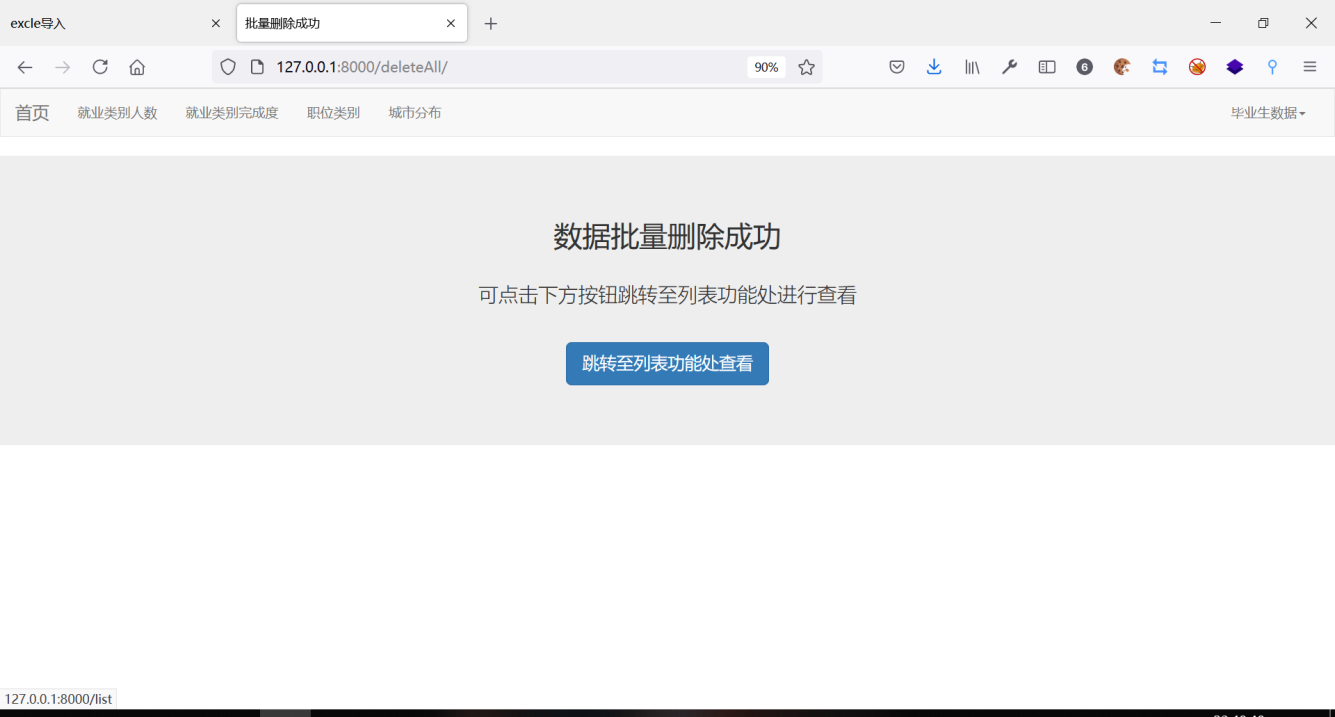


图4.18批量删除数据成功提示页面图

批量删除数据功能的代码实现为使用cursor.execute()函数执行DELETE FROM vsapp\_student语句。

## 4.4毕业生数据可视化展示

如下图4.19所示，除了首页以外，展示将数据进行可视化展示的页面共有四个，分别为：就业类别人数、完成情况、职业类别、城市分布。

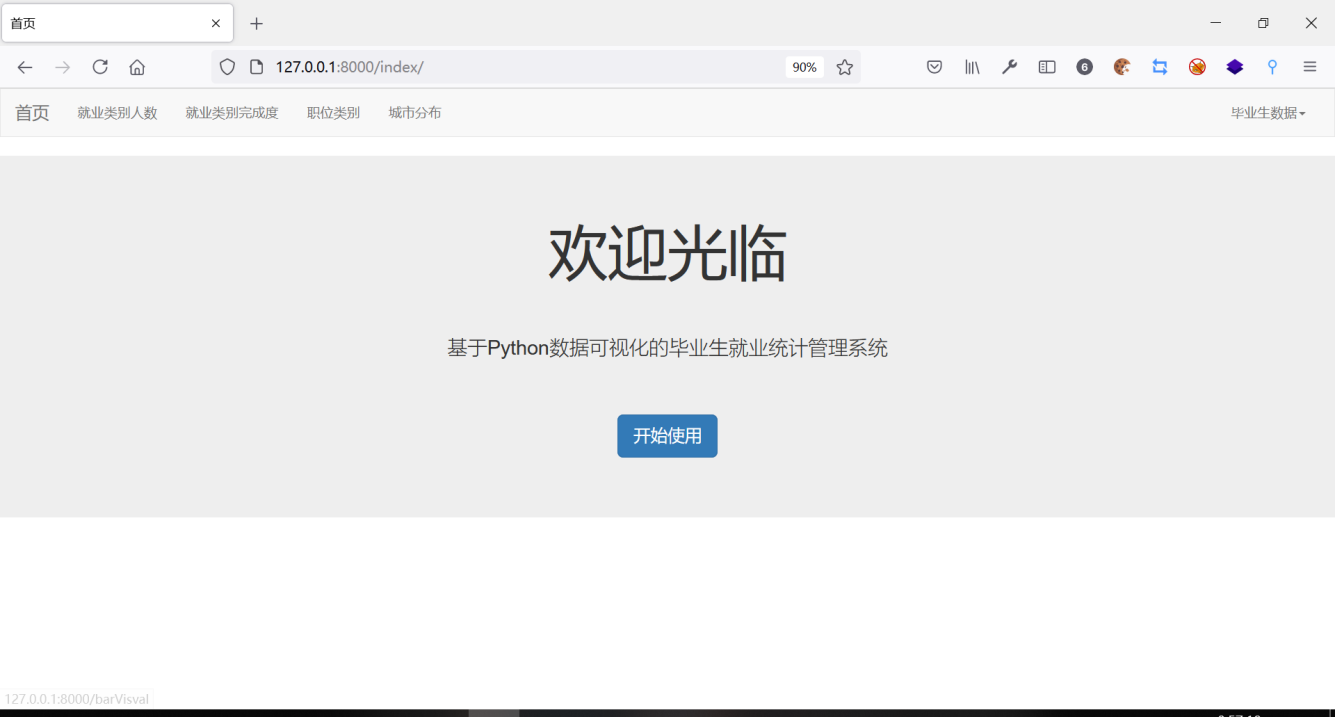


图4.19首页

### 4.4.1就业岗位分布人数统计图

图4.20为就业类别人数可视化页面，我们可以通过图4.20看出各个专业的毕业生毕业后选择的方向，本系统大致分为四个方向，分别为企业就业、升学、考编、从军。以这批数据为例，通过下方截图我们可以看出网络工程、物联网工程、软件工程、数字媒体技术的毕业生中选择企业就业的人数分别为57、54、121、42；选择升学的人数分别为1、2、35、0；选择考编的人数分别为6、10、17、5；选择从军的人数分别为1、2、1、0。

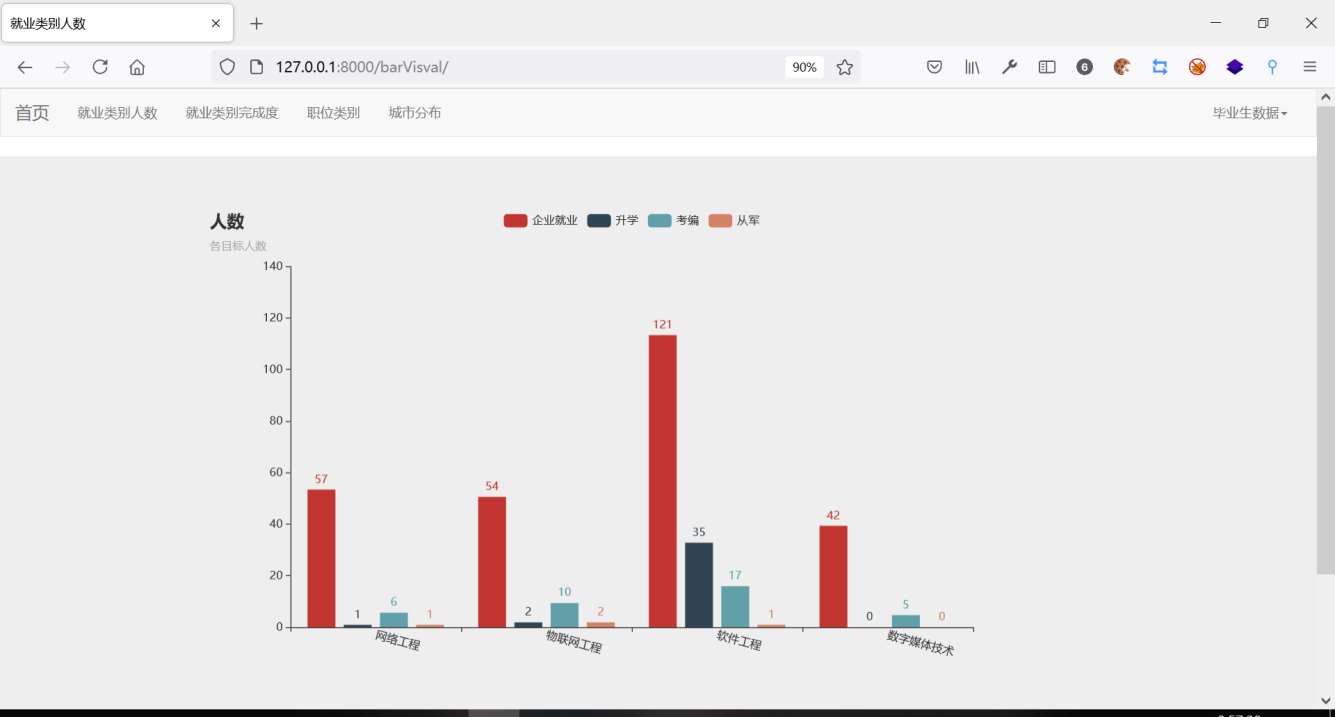


图4.20显示所有方向的就业类别人数页面图

当选择方向和专业人数较多时难以辨别数据，可以点击某个分类，使它暂时不显示，减少非目前需要的分析要素会使图表更加清晰简洁利于分析。以图4.21所示，在就业类别人数页面点击企业就业按钮，使企业就业分类暂不显示。

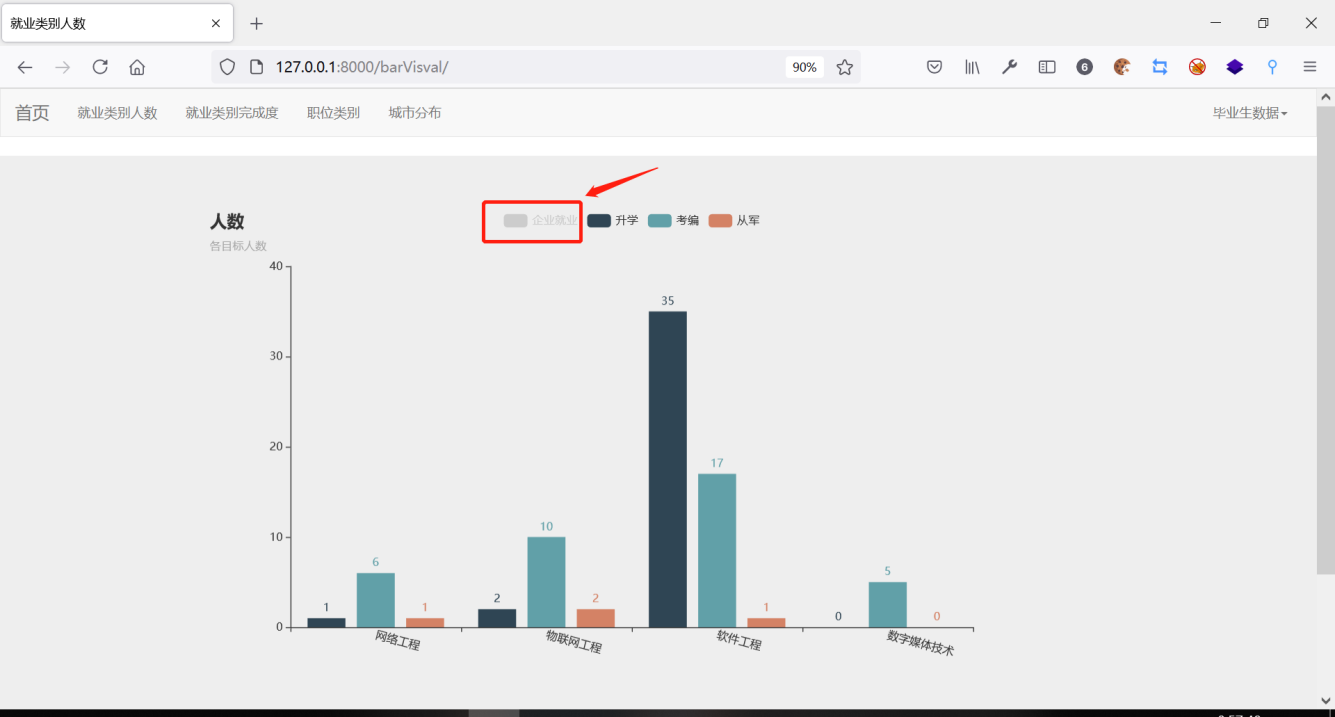


图4.21就业类别人数中暂不显示就业方向的页面图

条形统计图的代码实现如图4.22所示，以各专业名字为横坐标，以选择各就业方向的人数为纵坐标，在 .add\_xaxis( )中填入专业信息，在.add\_yaxis()中填入就业类型，在opts.TitleOpts()内填入标题等信息，最后使用.render()函数生成可视化页面到指定位置。

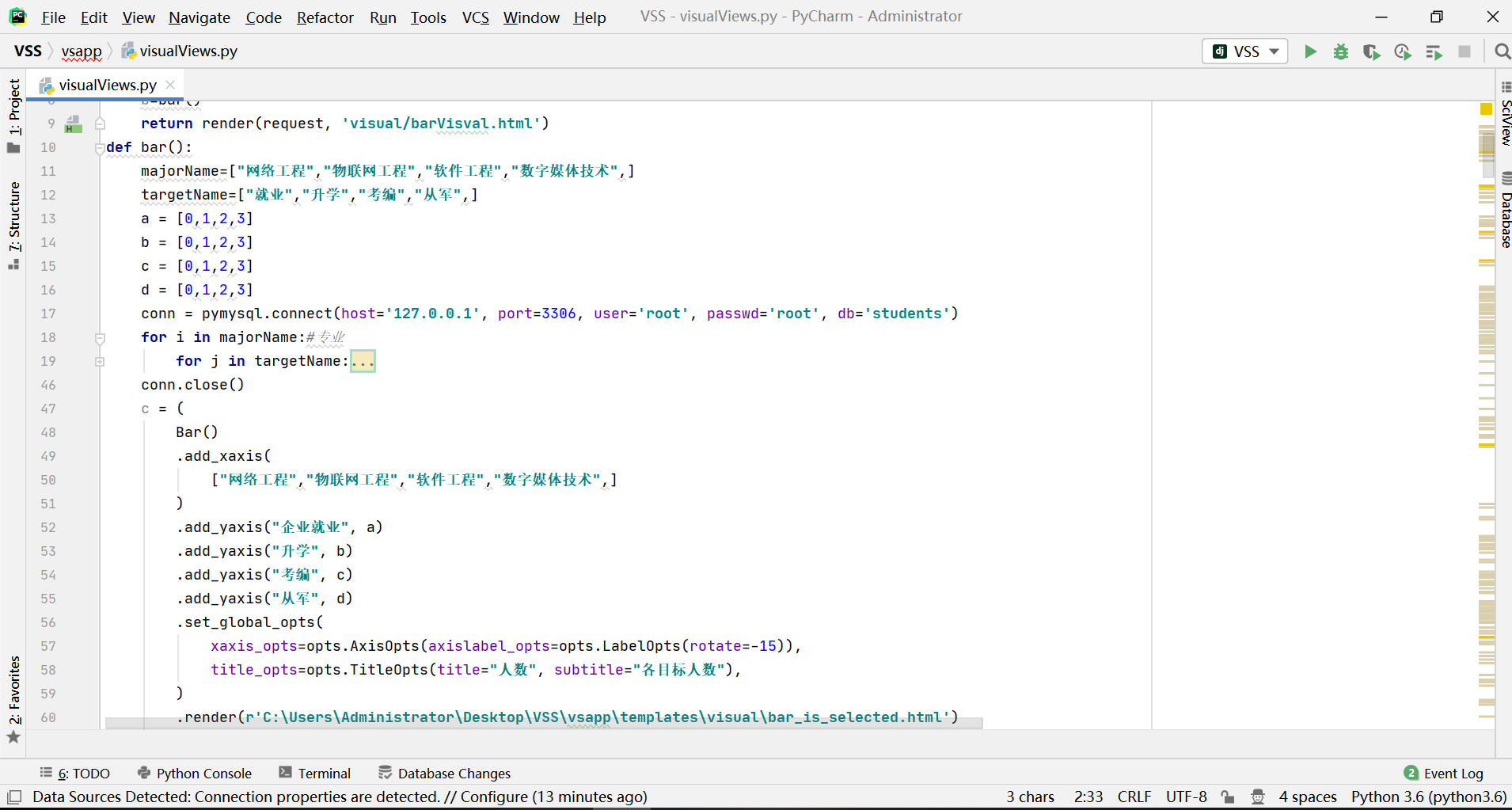


图4.22就业类别人数页面代码实现图

### 4.4.2就业岗位完成百分比

各就业方向完成情况页面如图4.23，通过图4.23我们可以看出以企业就业、升学、考编、从军为目标的成功率各为多少，以下图的这批数据为例，选择企业就业的学生中，有百分之82的学生就业成功，还有百分之18的学生暂未找到合适的工作；选择升学的学生中有百分之97的学生考研成功，还有百分之3的学生暂时未升学；选择考编，即选择考公务员或是进有编制的国企事业单位的毕业生中所有人都成功达成目标；选择从军的毕业生也都达成了目标。

企业就业成功百分比的计算方式为用的就是已签约人数除以实习人数，升学成功百分比的计算方式为用的就是考上研究生人数除以参与考研的总人数，考编成功百分比的计算方式为用的就是考上公务员与事业编制的人数除以参与考公与考编的人数之和，从军成功百分比的计算方式为用的就是当上军人的人数除以参军总人数。

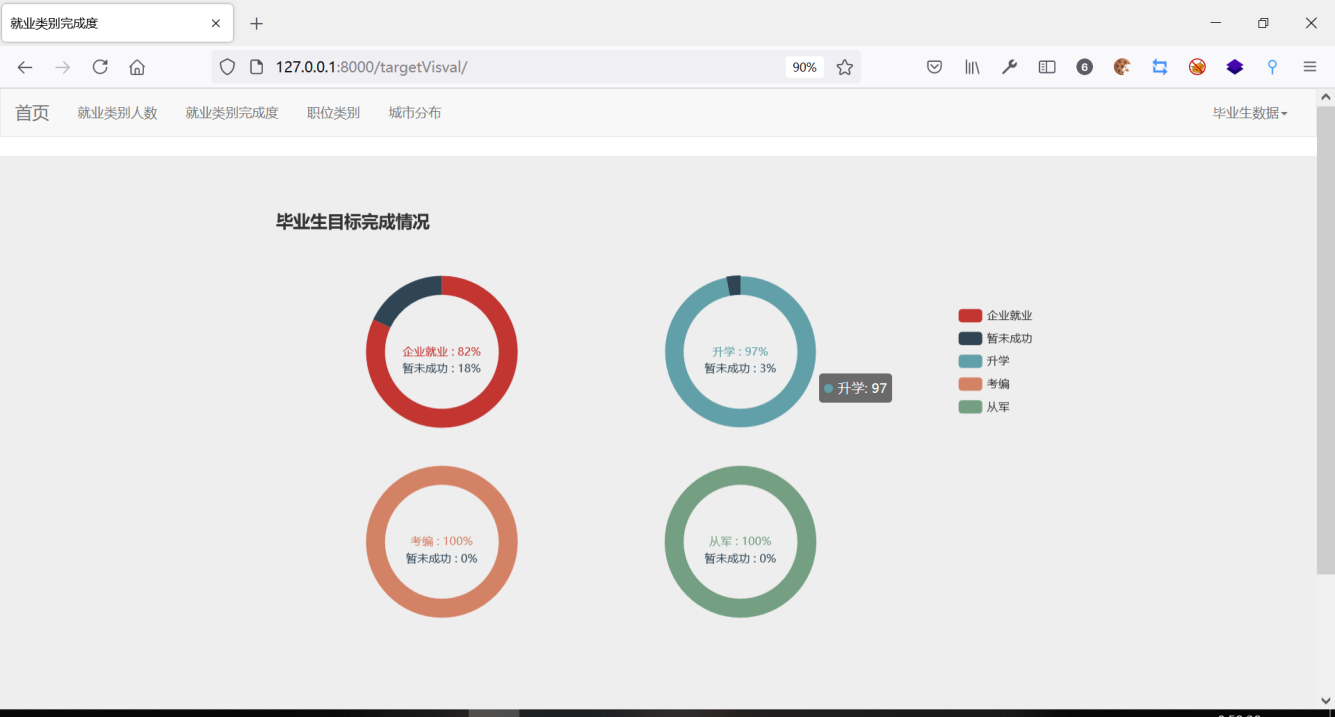


图4.23各就业类别完成度页面图

在完成情况页面点击企业就业按钮，使企业就业分类暂不显示，使可视化页面表更加清晰简洁利于我们分析，如图4.24所示。

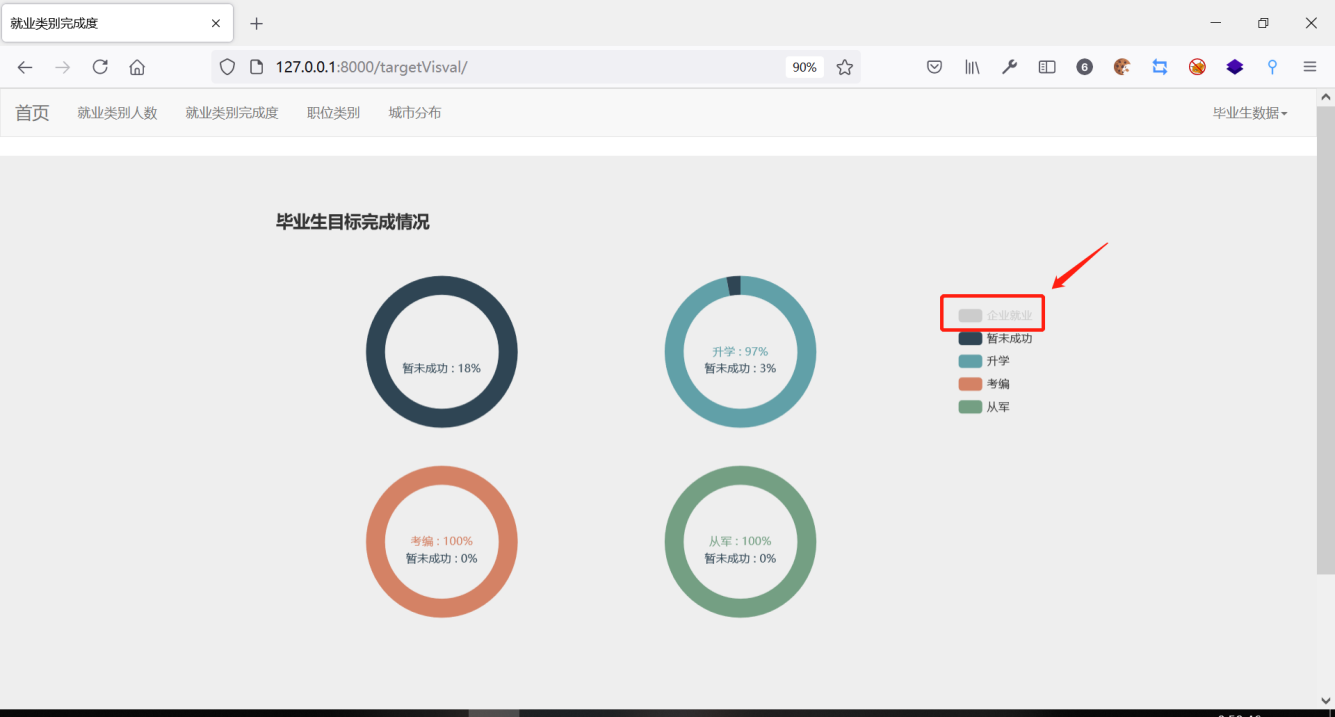


图4.24就业类别完成度暂不显示就业方向的页面图

各就业方向的完成情况的扇形图的代码实现如图4.25所示，就业、升学、考编、从军各为一个扇形图，分别展示他们的成功率占比；在.add()内填入我们想在扇形图中展示的数据等信息，使用在center列表设定图例位置，radius设定饼图的内外半径大小，数组的第一项是内半径，第二项是外半径；在opts.TitleOpts()内填入标题等信息，最后使用.render()函数生成可视化页面到指定位置。

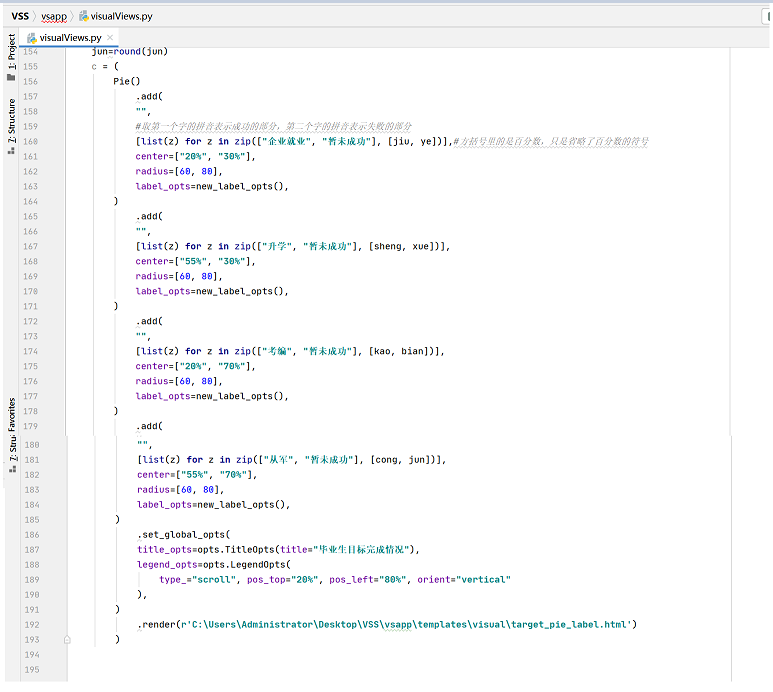


图4.25各就业方向完成情况代码实现图

### 4.4.3具体岗位类别

职位类别页面如下，有14种就业分类，分别为：办事人员和有关人员,工程技术人员,公务员,教学人员,金融业务人员,经济业务人员,军人,科学研究人员,其他人员，其他专业技术人员，商务和服务业人员,卫生专业技术人员,文学艺术工作人员,新闻出版和文化工作人员。

通过图4.26我们可以看出选择这14种就业类别的毕业生人数即其占比各为多少。

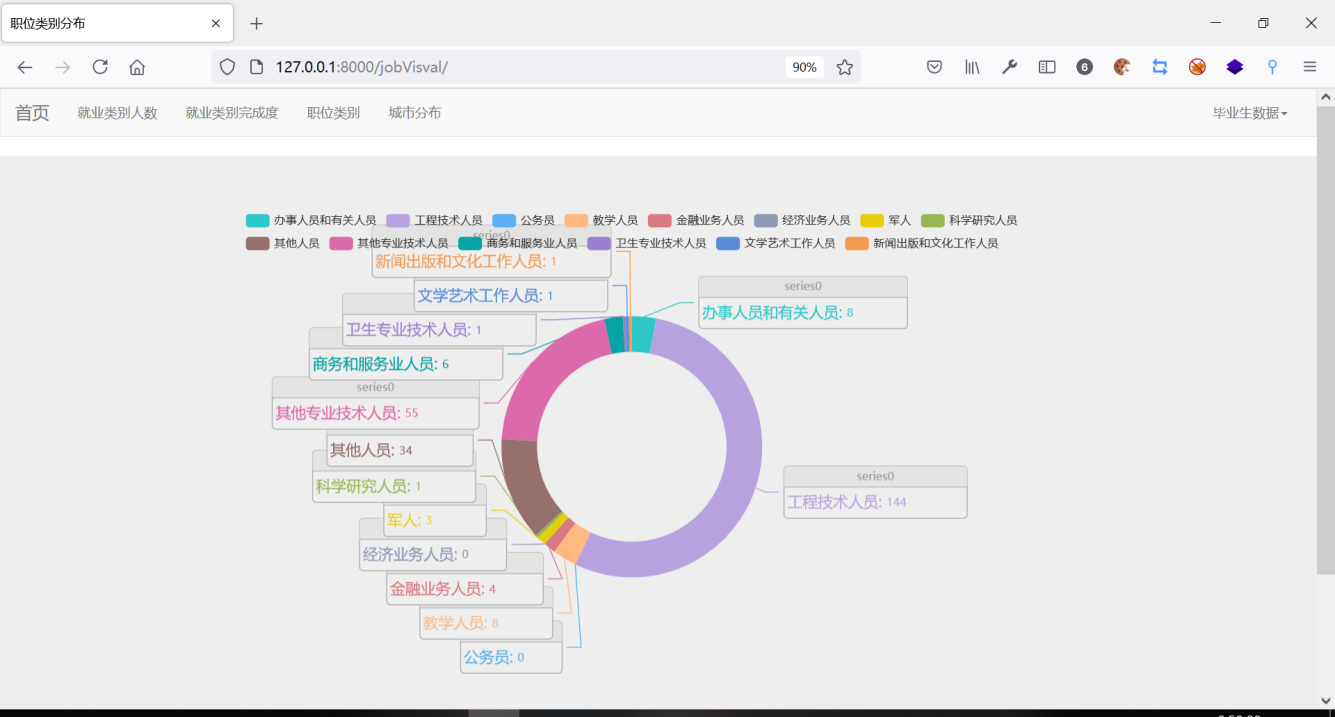


图4.26显示全部职位类别页面图

在职位类别页面点击工程技术人员、其他人员、其他专业技术人员的按钮，使这些分类暂不显示，使可视化页面表更加清晰简洁利于我们分析，如图4.27所示。

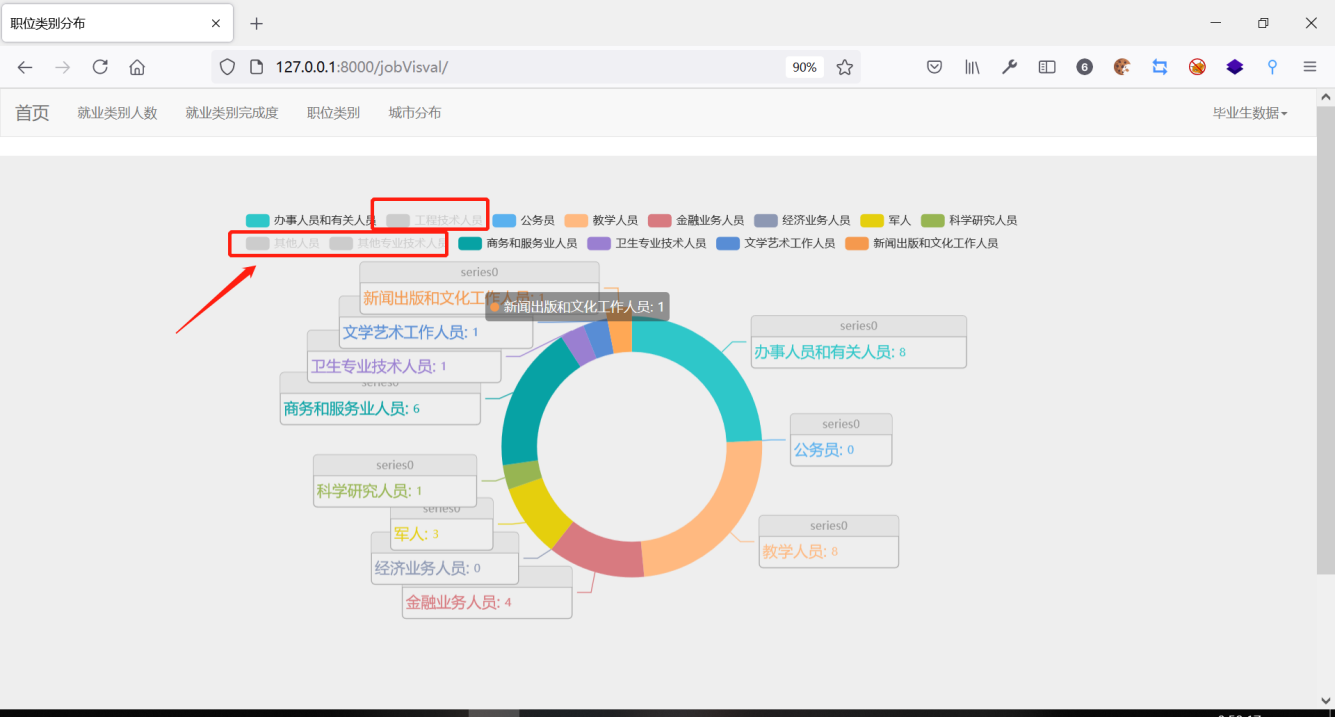


图4.27暂不显示某几类职位类别的页面图

职位类别页面的代码实现如图4.28所示，在.add()内填入我们想在扇形图中展示的数据等信息，使用在center列表设定图例位置，radius设定饼图的内外半径大小，数组的第一项是内半径，第二项是外半径；通过background\_color 设置背景颜色，border\_color设置边框颜色，border\_width设置边框宽度，border\_radius设置边框四角弧度，在opts.TitleOpts()内填入标题等信息，最后使用.render()函数生成可视化页面到指定位置。

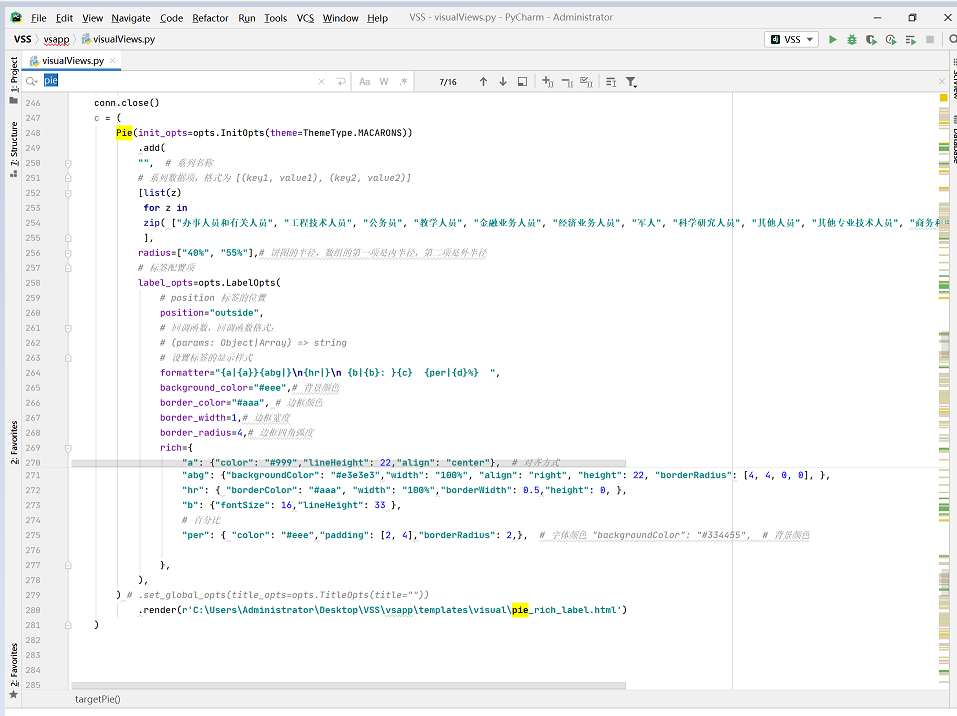


图4.28职位类别代码实现图

### 4.4.4毕业生就业地域分布

毕业生就业地域分布可视化页面共展示了190个城市，如图4.29所示。



图4.29城市分布显示全部页面图

当选择方向和专业人数较多时难以辨别数据，可以点击某个分类，使它暂时不显示。如图4.30所示，在城市分布页面的密度指数处，点击“少”按钮，使这个分类的城市暂不显示。

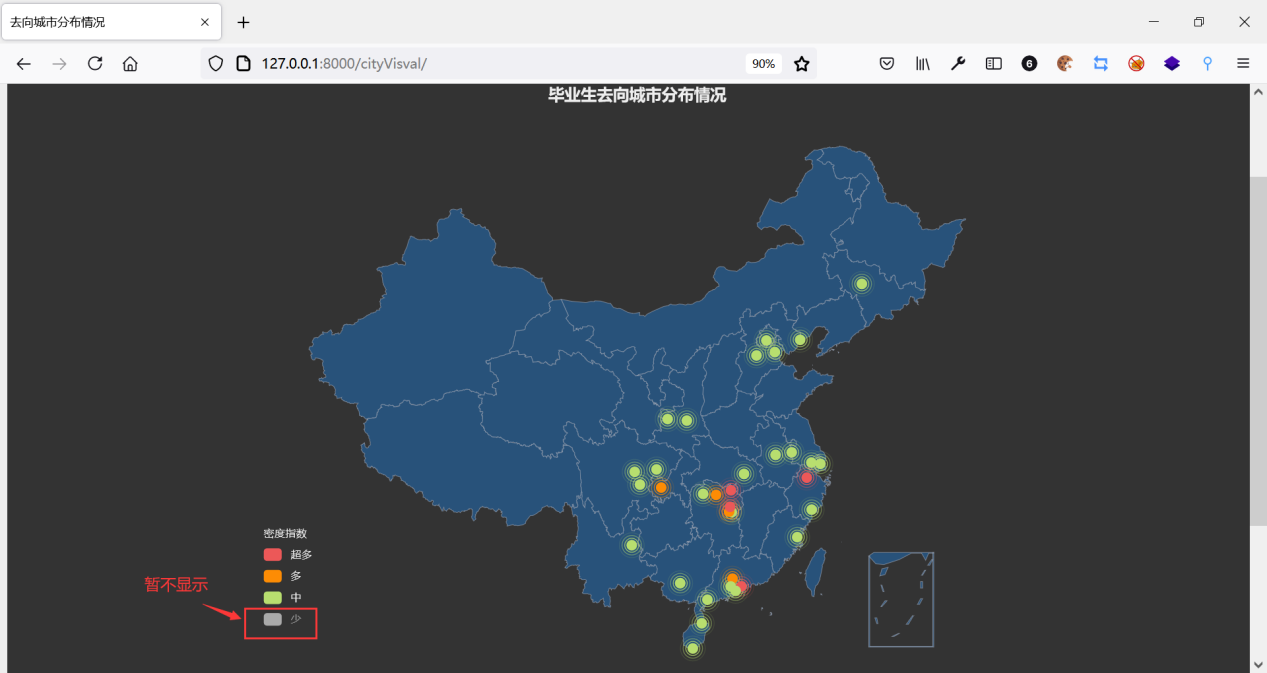


图4.30暂不显示密度指数为少的城市页面图

当鼠标指针放到某城市所在的坐标时，页面中还会显示该城市的毕业生具体人数，如图4.31所示，鼠标放在地图中长沙坐在的坐标时页面中显示出了长沙的毕业生人数为26。

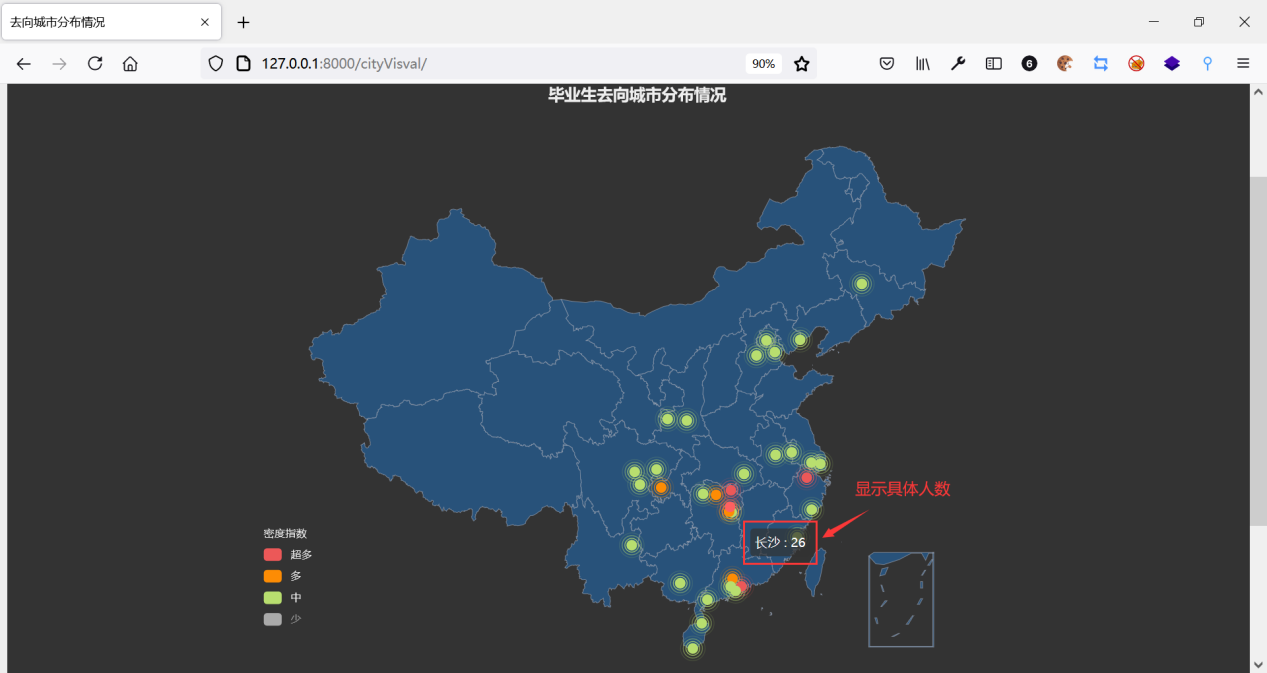


图4.31显示某城市具体人数页面图

毕业生就业地域分布图的代码实现如图4.32所示，在Geo()内定义图表大小与主题风格；在.add()内填入想在扇形图中展示的数据等信息；在opts.TitleOpts()内填入图表标题；等信息，在opts.VisualMapOpts( )内设定我们的指数分类，range\_text为分类名称，split\_number指定分几个分类，pieces则为具体定义各分类的信息；最后使用.render()函数生成可视化页面到指定位置。

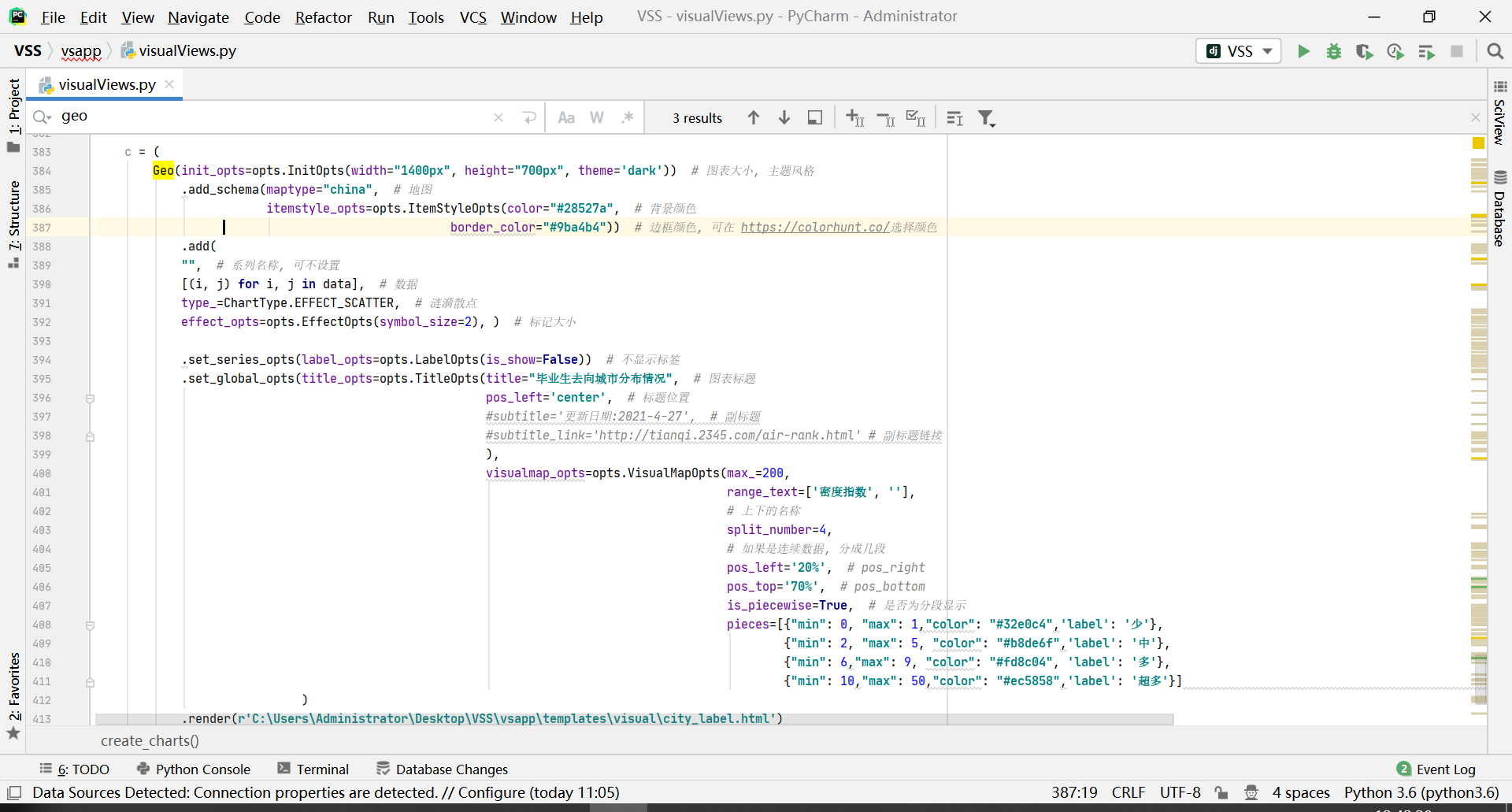


图4.32毕业生就业地域分布代码实现图

# 第五章 总结与展望

## 5.1 总结

这个系统所需用到的知识点大致可以分为四块，django框架的使用、python使用pymysql模块操作数据库、pyecharts模块的使用、前端页面的布局。系统中主要使用Django框架的基础部分：django模板、django模型、django路由、django视图；对于数据库中的多数操作则是使用了连接、执行、从执行结果中取出需要的元组等函数，其中的难点是excle表格批量导入部分需要xlrd模块与pymysql模块结合使用；pyechrats主要是版本的更新，前后版本不兼容，在网上找相关参考资料时容易因为自己使用的版本与参考的版本不同而导致报错，一开始可能难以调用，但在尝试使用几个图像模块后就能融汇贯通的编写；前端页面使用了Bootstrap与jquery，这不仅使前端看起来更美观也减少了编写前端代码的工作量。

这个系统总体来说体积较小，运行流畅，可视化界面美观，很适用于分析毕业生就业信息。

## 5.2 展望

本系统的后端使用python编写，当运行大量数据时他的反应速度较慢，本系统可以分析一个学院或是一个大学的某一届毕业生数据，但是对于数十万条数据的体量，运行本系统就会比较吃力，pyecharts的可视化界面美观，使用方便，这是有优点，但python是面向对象的一种解释型计算机的程序设计语言，并不适用于需要快速处理数据的大型系统的后端。所以对于本系统的展望则是与其他语言结合，各取所得，优势互补，改造成一个界面美观，可快速处理大量数据的系统，能帮助我们从大量的数据中更客观的分析往届毕业生的就业信息以便为高校学生提供更好的就业服务。