**计算机科学与工程学院 实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验课程名称** | | **Python编程与数据分析project3** | | | **实验总成绩** |  |
| **专业** |  | | **班级** |  | **指导教师签字** |  |
| **学号** |  | | **姓名** |  | **实验报告批改时间** |  |
| **基于Django+爬虫的C/S架构二手房数据收集与展示** | | | | | | |
| **项目背景及思路**  1.在上实验三的实验课前我是用python Socket模块编写的C/S架构，原理十分简单。因为我project2的项目是基于pyqt5编写的，整个项目分为两部分：主界面和展示界面，其实主界面就相当于客户端，展示界面会让输入爬取的页码，内部调用爬虫函数，就相当于服务器，所以我一开始的时候只需要将这两个界面稍作修改：将原来在主界面中输入的城市名称作为客户端发送的数据发给服务器端，在服务器端接收之后进行逻辑功能（也就是爬虫）的调用，并将从postgre数据库中取到的数据存入列表再返还给客户端进行展示即可。只在原基础上加了两个函数：client()和server()。在上完实验课后，看到学长是用Django编写的项目，十分吸引我，于是就自学了Django并写了project3的第二个版本，也就是这个报告中即将展示的内容。  2.项目环境：PyCharm2020.2 + python3.7 + Django3.2；界面使用html编写；css和js使用bootstrap-3.4.1及font-awesome-4.7.0；图表展示使用echarts.js。  额外导入的包都在项目文件夹下app01/static文件夹中。  3. Django 采用了 MVT 的软件设计模式，即模型，视图和模板。M是Model，模型，是用于完成操作数据库的，进行数据处理。V是View，视图，接收请求，进行业务处理，返回应答。T是Template，模板，负责封装构造要返回的html，也就是用于展示给客户端的页面效果。路由控制器（即项目中url.py文件中的url）将请求转发给对应的视图函数，完成业务逻辑，视图函数将从model中获取的数据嵌入到template中的模板文件（html）渲染成一个页面字符串，返回给客户端的流程。  4. 在初次使用Django时，我一直在命令行中用django-admin进行项目的创建、运行与调试（因为我一直使用的是社区版的PyCharm，不能直接新建Django项目），十分不便。为了提高效率，磨刀不误砍柴工，我从网上找到了PyCharm专业版的破解方式，成功将我的PyCharm破解到了2099年12月31日。  **项目文件结构**  demo3:  ├─demo3  │ ├─\_\_init\_\_.py  │ ├─settings.py  │ ├─urls.py  │ ├─wsgi.py  │ └─asgi.py  ├─app01  │ ├─migrations  │ │ └─\_\_pycache\_\_  │ ├─static  │ │ ├─css  │ │ ├─img  │ │ ├─js  │ │ └─plugins  │ │ ├─bootstrap-3.4.1  │ │ │ ├─css  │ │ │ ├─fonts  │ │ │ └─js  │ │ └─font-awesome-4.7.0  │ │ ├─css  │ │ ├─fonts  │ │ ├─less  │ │ └─scss  │ ├─templates  │ ├─utils  │ └─\_\_pycache\_\_  └─  主要文件以及文件夹介绍：   1. \_\_init\_\_.py：inti文件，标识当前所在的项目目录是一个 Python 包。 2. settings.py：项目配置文件，用来进行app的注册以及数据库的连接。 3. urls.py：路由文件，即项目包括的网址。 4. manage.py：主文件，项目的入口，用于启动项目、创建应用和完成数据库的迁移。 5. static文件夹：是我自己创建的静态文件夹，里面是一些插件、css、js以及图片文件。 6. templates文件夹：保存自己写的html文件。 7. app01文件夹中的views.py：是视图函数文件，用来实现逻辑功能，比如数据库数据的增删改查、接受POST请求的内容等功能。 8. app01文件夹中的models.py：是模型文件，定义了所需要的表结构（如用户表、房屋信息表），通过Django的ORM来转换为SQL语句。   **运行结果展示**   1. 启动项目：     点击PyCharm控制台中生成的蓝色网址，127.0.0.1是本地环回地址表示本机，即主机号，8000是项目对应的端口号（与python Socket中的hostname和port相对应）  点击之后会自动打开浏览器并出现这个界面，1-7都可以分别打开。   1. 先进入主界面网址：<http://127.0.0.1:8000/mainscene/>   右上角有两个按钮，现在显示的分别是“登录”和“注册”，左上方是输入城市的按钮，用来获得用户的输入。下面以“长春”为例进行展示：  在搜索框中输入长春并按回车（或者点击Subbmit按钮）会自动跳转到下一个网址：    这个界面左上方有输入框用来供客户输入想要爬取的页面数，我输入6（链家网址煤业有30条二手房数据，6页就是180条）并按回车，会跳转到数据展示界面：  数据展示展示的是从数据库中读出的数据。刚才输入“6”并按下回车之后就会自动调用视图函数中的爬虫功能，爬下来的数据先直接存入数据库，再用form表单进行展示。  这里值得一提的是我自己写了一个翻页的页码条，在每一页上只显示10条数据，更加贴合网上的网站。  刚才计算出来一共有180条数据，可以看到页码条到第18页，证明正确。   1. 图表展示：   在获得爬虫数据之后点击刚才页面右上方的“展示”按钮，会跳转到展示图表的界面，这里我使用了echarts中的三种图表进行展示：直方图、折线图和饼图：      数据先设置成了这5个城市的房屋数量，从数据库中读出用ajax传入。   1. 更新数据：   点击右上方的“更新”按钮之后会跳转到更新数据的界面：  为了能够让用户有更好的体验我设置了用户可以自己输入想要更新的页数（即要新增的页数），这里我新增一页数据就输入1再按回车，等爬取完毕并存入数据库后就  原来“长春二手房数据库”中有180条数据，更新1页之后成为210条，证明更新成功。    （postgre数据库数据截图）   1. 注册和登录：   再次回到主界面中：<http://127.0.0.1:8000/mainscene/>  点击“注册”按钮会跳转到注册界面：    同样的点击“登录”按钮之后会跳转到登录界面：    按照要求输入即可完成登录和注册，并且在注册时会将用户名和密码存入数据库中的userInfo表中。    **代码展示**  1. url.py中的路由：对应刚开始运行程序时自动打开的界面上标的1-7条网址      2. views.py文件中mainscene路由对应的视图函数：    几乎所有的视图函数的逻辑都恩威这两部分：用if语句判断是“GET”请求还是“POST”请求。“GET”请求表示输入网址按下回车之后视图的动作；“POST”请求表示再页面上输入之后点击提交按钮之后视图的动作。  3.models.py中部分表：    这相当于SQL的建表，Django的ORM会自动将类转化为建表语句进行调用。  4. 页码条代码展示：  首先默认展示第一页；之后根据输入的页码数进行简单的计算展示页码：保证再页码数较多时展示当前页码的前5页和后5页（即始终保持11页页码显示在屏幕上），当是第5页时向前只展示前4页，保证不出现负数页码。  5.数据展示代码：  先将数据库中的数据作为参数传递到html界面中：    在html中接收数据，并在form中循环展示出来： | | | | | | |