暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | Django开发 | | | | | | |
| 姓 名 | 王一名 | 专业 | 人工智能 | 班级 | 091182 | 学号 | 09118219 |
| 实验时间 | 2020.8.31-2020.9.23 | | 指导教师 | 孔祥龙 | | 成绩 |  |
| **一、实验背景和目的**  在利用爬虫技术爬取了各高校的招生分数线之后，继续利用这些数据做未来分数线的预测是一种很好的想法。  与此同时，由此项目产生的分工与合作，也能够让同学们感受大厂分工的安排，学习并适应如何在组内和组外互相沟通，最终完成项目，这为将来的工作与实习打下基础。 | | | | | | | |
| **二、小组任务和个人任务**  小组任务：对接其他小组的数据和算法，通过Django框架下的ORM操作来搭建一个包含大学、专业、排名等类的数据库。  个人任务：创建登录界面、创建数据表格、合作完成colleges类的数据库存储、对接的管理，后期由于数据问题，和组员一起进行数据的清洗与修改。 | | | | | | | |
| **三、个人任务需求分析**  **编程：**  1.需要熟悉Django框架开发相关功能编写，了解在此框架下数据库的实现  2.需要创建数据表格，一一读进大学数据，主要是college类的完成  **对接：**  1.需要整合第4组和第5组（知识图谱）的反馈，及时调整数据库情况  **额外：**  1.检查之前的一分一段表和大学数据  2.根据官网正确数据修改部分大学的错误数据 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  1.INSTALLED\_APPS = [**'ormDesign'**, *#数据库搭建*]  应用的创建，后来的工作主要是在ormDesign中完成  2. **def** index(request):*#测试数据，实际使用时需要从数据库读取相关数据，以构建下拉菜单* province = {**"1"**: **"北京"**, **"2"**: **"天津"**, **"3"**:**"江苏"**}  category = {**"1"**: **"理科"**, **"2"**: **"文科"**, **"3"**:**"不分文理"**}  **return** render(request, **'index.html'**, {**'province'**: province, **'category'**:category})  响应部分的编写，便于数据库的建立  3. urlpatterns = [path(**'ormDesign/'**, include((**'ormDesign.urls'**, **'ormDesign'**), namespace=**'ormDesign'**)),]  url部分的实现：子路由的修改，实现从网页端进入查看结果  4. **class** Colleges(models.Model): *# 大学类* collegeID = models.IntegerField(primary\_key=**True**) *# 大学id（主键）* collegeName = models.TextField() *# 大学名称* provinceID = models.ForeignKey(**'Provinces'**,on\_delete=models.DO\_NOTHING,  null=**True**) *#省份id（外键->Provinces类）* project985 = models.BooleanField(default=**False**) *#是否为985大学* project211 = models.BooleanField()*#是否为211大学* top = models.BooleanField() *#是否为一流学校* **class** Meta:*#设置元信息，使用db\_table自定义表的名字* db\_table = **'Colleges'** verbose\_name = **'大学列表'**  创建数据表格  5.filename=os.getcwd() **def** getValue(x):  **if** x==1:  **return True  else**:  **return False** df=pd.read\_csv(filename+**"\extFiles\college.csv"**,encoding=**'gbk'**) **def** getCollege(request):  **for** x **in** range(0,89):  c1=models.Colleges(collegeID=df.at[x,**'collegeID'**],  collegeName=df.at[x,**'collegeName'**]，  provinceID=df.at[x,**'provinceID'**]  project985=getValue(df.at[x,**'985'**]),  project211=getValue(df.at[x,**'211'**]),  top =getValue(df.at[x,**'top'**])  c1.save()  **return** HttpResponse(**"Finished!%s"**%app\_name)  For循环逐行读取数据，存入college表格 | | | | | | | |
| **五、实验结果与分析**    最终结果如上图，整个数据库搭建完毕，最后将各个部分：college、category、major等类的代  码汇总，生成最后的大数据库。  其他组只需要从db.sqlite3文件中读取自己需要的数据，完成各项功能搭建即可。  另，如有需要json文件的组，我们组最后也和第一组一起，处理了部分有误的csv和json数据，修改完成后重新搭建了数据库，确保数据的有效性。 | | | | | | | | |
| **六、实验总结与心得体会**  经过这次大项目以后，我主要有两点体会，分别是团队协作和及时调整、及时检查。  团队协作项目，一定要注意合作。这里的合作分为组内和组外。组内大家以组长为核心，组长根据项目下该组的要求分配任务给组员；组外则主要是各组之间的互通，组长之间会有交流，每个组也都有负责对接的同学，确保其他组的需求能够及时反馈，及时修改。这里面可能会牵涉一些理解上的不同而带来的冲突（往往是组与组之间），这个时候可能应该还是从整个项目的角度出发去构想某个问题，而不是仅仅从某个组的角度，更重要的是大家之间最好也要相互理解，毕竟都是想要好好的完成这个项目，所谓对事不对人嘛。  再有就是及时检查，也是十分重要的一项工作，可以算是项目的隐式作业吧，这点在git的pull&request中也是有体现。但我觉得可能远远不够，就个人来讲，做完一个小的分工任务，需要对照要求去检查结果的合理性；另一点在这次项目体会到的就是，别人做的工作部分也需要检查，而不是默认别人做的没问题。像这次的项目，我们组在第三周的时候发现了以前许多数据的格式存在问题，甚至有的内容也有问题。这些问题如果得到更早的排查，也许也就不需要麻烦其他做算法的组修修改改他们的算法内容了。  总之，就是项目的各项安排要体现一个严谨性，严丝合缝，才能做出完美的项目。 | | | | | | | | |

2020年9月制