个人总结

Chenzx

一．项目初衷与初期设想

大学生活中，同学之间的交流在学习生活中扮演者重要的角色，在交流的过程中，我们不仅能够收获从学长学姐那里了解到课程中隐藏的“坑”，同时也能收获对课程的新的认识，对于提升自己的学习有着巨大的帮助。但是目前并不存在一个能够供同学们交流学习经验的平台。基于此想法，我们小组希望设计一个课程交流网站，来方便同学们进行学业上的交流，促进同学们在学业上进步。

由于大家都没有过任何软件或者网页开发的基础，所以初期我们在阅读学长的个人总结并听取了老师在网页开发方面的建议后，很快确定了以比较容易入门的 Django 作为我们开发的框架。同时开发过程的初期以学习为主，并且将《跟老齐学 Django》这本书作为我们共同学习的内容。

通过讨论，我们确定了第一轮 sprint 和第二轮 sprint 所要完成的内容的整体框架。第一轮主要实现提问、回答、分区等基本功能；第二轮则实现用户注册，搜索，评论，按热度排序等进阶功能。

二．第一轮迭代

第一轮 sprint 中，我们将负责前端的同学与负责后端的同学进行了分工，将不同的 app/前端模板细化给每个同学。另外，第一轮 sprint 我们就明确了统一使用 pycharm 这款 IDE，减少了因为 IDE 不同而导致的额外问题，同时 pycharm 不需要额外的进行配置，所以对于之前不了解 python 的同学也非常友好。

我被分配到了后端的分区功能的设计任务。所谓万事开头难，我对于 django 的理解还不够深刻，对于分区这一功能如何实现也没有思路。幸好有学长之前做的网站源代码可以参考，通过理解学长的代码，让我逐渐对于分区功能有了一定的认识。同时，实现分区功能的过程也让我对 django 的 MTV 开发模式有了一定的理解。首先在 Model 中明确功能涉及到的属性，如分区功能涉及到的分区名、分区编号这两个属性；在 Model 与数据库建立了联系之后就可以在 templates 文件夹中单独建立一个名为 category 的文件夹，将分区相关的展示页面写到其中；最后在 views 中编写视图函数，与前端建立联系。在 django 框架下主要的开发过程大概就是这样，当然设置 url 路径等一些操作也是不能忘记的。

```python
class Category(models.Model):
    """
    分组模块
    """
    name = models.CharField(max_length=20, unique=True, verbose_name='分组名称')
    number = models.IntegerField(unique=True, verbose_name='分组序号')

    def __str__(self):
        return self.name
```

由于我们初期后端同学的进度比较快，所以后面我又被分配了搜索功能的实现这一任务。刚开始着手实现搜索功能的时候还是一头雾水，通过查阅网上的代码，我明白了搜索其实就是从数据库中进行关键词查找，并且了解到了 icontians 这个可以进行集合过滤的字段。这

样子搜索功能的后端实现其实就变得简单了很多。只需要通过 request.GET.get('key')获取前端传递过来的关键词，之后利用 icontains 这个字段进行 filter 查询，如 question_list = Question.objects.get(questionTitle__icontains=keyword)，从数据库中获取题目包含 keyword 的问题就可以实现搜索这一功能。最后将搜索到的 question_list 返回前端进行展示，搜索功能就大功告成了。另外，我还在查询语句的后面加入了 order_by()，使得搜索的结果能过够按照点赞数、发表日期等属性进行排序。

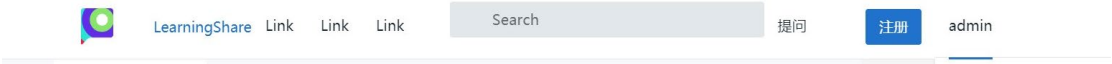
```
def search(request):

    keyword = request.GET.get('keyword')
    err_msg = ''

    #无法找到关键词
    if not keyword:
        err_msg = '请输入关键词'
        return render(request, 'question/search.html', {'err_msg': err_msg})

    #按照赞数、时间、名称进行排序
    question_list = Question.objects.filter(questionTitle__icontains=keyword).order_by('-goodNum', 'created', 'que
    #按照发布时间、问题进行排序
    answer_list = AnswerModel.objects.filter(answer_text__icontains=keyword).order_by('-pub_date', 'question')
    #按照用户名进行排序
    user_list = User.objects.filter(username__icontains=keyword).order_by('-username')

    return render(request, 'question/search.html', {'err_msg': err_msg, 'question_list': question_list,
                                                    'answer_list': answer_list, 'user_list': user_list,
                                                    'keyword': keyword})
```

三．第二轮迭代

第一轮 sprint 的过程中，我们实现了分区、回答、提问这些基本功能，同时还基本实现了搜索和注册两个原定于第二轮进行的功能。但是搜索到的内容还并不完善，比如，搜索到的用户，并没有相应的模板展示用户对应的问题和回答。所以我继续对这个功能进行了完善，增加了一个展示用户提出的问题与回答的页面。有了前面搜索的经验，这部分内容非常好做，只需要将用户对应的 id 传递到后端进行问题与回答的查找，之后将它们传递到前端即可。但是这里也遇到了一个问题，就是用户名对应的连接需要传递用户对应的 id 才行，否则是无法跳转到指定的用户页面的。

```
<div class="tl-item-title">
    <a href="../show_question/?user={{user.id}}">{{user.username}}</a>
</div>
```

由于前端同学在第一轮中主要处于学习阶段，所以前端的进展并不大。于是我和单好思同学加入到了前端页面的开发中。首先负责的任务是对整个前端页面进行优化。由于之前的前端页面都是基于《跟老齐学 django》中的内容进行编写，所以非常简单，而且还有很多地方看起来还并不协调，所以优化的任务工作量其实很大，要将几乎每个页面都进行相应的修改才行。加上之前并没有对 html、css、javascript 进行系统地学习，所以接到这个任务的时候压力还是蛮大的。因为从头编写 css 进行优化实在是太过艰难，所以我首先想到的还是套用模板在现有的页面布局基础上进行修改。正好王晨宇同学在群里发过类似地模板内容，但是并没有应用。于是我就依托于这套模板，对页面内容进行了一定的修改。



因为刚开始接触前端，加上对模板中的结构并不熟悉，我在一开始遇到了很大的困难，光是一个导航栏就弄了大概两三个小时才改好。这其中我不仅修改了原有的 bootstrap 的栅

格模式，同时由于打乱了之前的页面结构，所以很多页面中的内容跟出现了位移。需要不断调试才能使其回归一种正常的状态。

另外，因为知乎可以对搜索结果的类别进行选择，我们组也想实现之一功能。这时候需要对想要搜索的类型进行判断，比如 type 是 question 则显示 question_List 中的内容，而这个 type 的值则通过选项按钮点击之后传递给后端，后端获取并返回给前端页面，由前端页面对 type 的值进行判断。但是这个时候就出现了另一个问题，type 返回给后端的时候 keyword 和之前的搜索结果 question_list 等就会丢失。为了保存这些内容，我将这些量设置为全局变量，同时在之前的搜索语句后面加上了判断，如果搜索非空则对 question 对他们重新赋值，搜索结果为空的时候则保留之前。这样子就能够完成对搜索结果的分类显示。

```
<li role="presentation"><a href="../search/?type=question">问题搜索结果：</a></li>
<li role="presentation"><a href="../search/?type=answer">回答搜索结果：</a></li>
<li role="presentation"><a href="../search/?type=user">用户搜索结果：</a></li>
```

```
#按照赞数、时间、名称进行排序
q = Question.objects.filter(questionTitle__icontains=keyword).order_by('-goodNum', 'created', 'questionTitle')
if q != '':
    question_list = q
```

四．个人技术体会与反思

1. Git 与 Github

软件工程这门课让我了解到了 Git 与 Github，尤其是 Github 上能看到很多大佬分享的开源代码，可以作为很好的资源进行学习。

2. Django

在这门课之前我并没有任何网页开发的经验，所以我们选取的 django 开发框架对于我这种新手来说非常友好。Django 框架的开发思路简单易懂，通过学习官方文档以及参考《跟老齐学 Django》这本书中的内容，我学会了配置 url 路径，编写 view 函数，编写 model 以及 form，还有 template，理解了 django 如何创建工程，如何创建功能这些基础操作。同时在项目开展过程中我也学会了用户注册，搜索，发表文章等功能的实现。现在的我基本具备了独立使用 django 框架开发一个简单的小型网站的能力

3. Javascript\css\html

虽然我是在第二轮 sprint 才开始进行前端内容的学习，但是这部分内容的学习让我受益匪浅。首先我了解了 css，javascript 的基本语法。之后在对前端页面进行优化的过程中，通过实践来进行进一步深入学习。编写自己的 css 代码的过程以及阅读响应式页面代码让我认识到了前端开发的工作量其实很大，特别是想自己从零完全写一个比较好的前端页面更是难于登天。这里面可能更多需要的是将别人已经写过的内容拿过来加入自己的代码之中，从而避免重复性的开发。同时也要非常注意不同的样式之间的关系，是否会因为大小不合适而导致不同内容在页面中的布局发生冲突，这些都是非常值得注意的内容。总的来说，对于 js 和 css 我只能说我还只是一个新手，今后还有很多内容要进行学习。

4. Refactor

关于重构我还是由很多要反思的内容。首先，在开发过程的开始就应该为重构做好准备，变量的命名都应该秉持一个标准，不能随心所欲，这样子会对后面的重构造成巨大的困难。其次，由于是团队进行开发，在编写任何一个功能的过程中，都应该保持写注释的习惯，否则后面其他同学想要对这部分代码进行修改的过程会变得比较艰难。并且，写注释能够让自己更清楚自己写的每行代码的功能是什么，能够从一定程度上避免无用代码的出现。另外，重构的过程也是一个重新阅读自己写过的代码的过程，能够从一个新的角度审视自己写过的代码，在对之前代码进行反思的过程中也能加深项目的理解。

五．团队合作方面的心得与反思

由于是团队进行开发，所以在提交代码的时候，尚未完成的工作使用 TODO 进行标记对于团队开发是非常重要的一点，这样可以很大程度上避免因为尚未完成的功能而导致的一系列代码冲突等情况的出现。同时，标记 TODO 也可以让自己快速找到上次工作中未完成的部分，对后续的开发有着比较重要的导向作用。

在开发的过程中，同学们在修改了一部分代码之后总是不及时提交，所以当其他同学也对这部分代码做出修改之后很容易就会在 git 提交的时候产生冲突。关于这一点，我的理解是在完成了任何一部分代码，哪怕非常少的一部分，也要及时地提交到 github 上面，这样子能在很大程度上避免这种冲突。同时在每次编写代码之前都先对 github 上的内容进行一次 pull 并且通过 github 上的提交记录了解 其他同学所提交的内容。这也是一种避免冲突的不错手段。

团队开发的过程中避免不了于其他同学的沟通。但是同学之间的共同和真正公司中团队成员之间的沟通存在很大的差距。主要是沟通的及时性问题，因为同学们使用社交软件的时间经常有差别，有的同学甚至很少查看微信上的内容，这就导致了沟通上存在很大的障碍。从一定程度上说就是会对团队开发造成阻碍。我个人的观点是，为了避免这种沟通问题，我认为虽然是小组合作，但是对于小组中的每个成员，都应该有自己单独承担的内容。而不应该出现两个人三个人一起承担某项功能的情况，这样很容易就会导致有人划水而其他人承担了所有情况的现象发生。并且这种多人承担同一项内容的情况也会出现沟通上的不及时性问题。每个人有自己的明确任务的话对于促进每个人的开发动力也会有比较好的效果。

对于软件工程这门课我的一个比较重要的体会是同学们之间的相互学习。这种相互学习并不只限于同组之间的相互学习，同时包含向其他小组成员学习，还有从学长学姐的报告中吸取经验。同组之间的相互学习对于整个项目开发都比较重要。项目初期，大家相互交流自己学到的内容， 同时解决其他同学在学习中遇到的问题。开发过程中，对于遇到的困难，可以拿到每周的小组讨论上请组内的大佬帮忙解决。组内的相互学习对于构建一个良好的开发氛围也有着很大的意义。同时，当想要开发的新功能并没有思路的时候，我也会去找其他组的同学进行交流，从他们那里取经，这帮助我解决了很多开发过程中遇到的问题。另外，学长学姐的软工总结和他们的项目代码也是很好的学习对象。我们这次开发的网站一定程度上参考了前几届学长学姐的代码。同时从他们的软工总结中，也能吸取到很多开发过程中的小经验、小技巧，并且对他们曾经踩过的坑有一个了解。这都对我们的开发有着很大的帮助。所以说相互学习对于项目的开发，有着极为重要的作用。

六．学习方法总结与反思

在开发过程中难免会遇到 bug，这个时候首先应该尝试自己进行解决，很多时候网页上都会有显示相应的错误提示，这对于明确 bug 出现的原因有很重要的帮助，可以通过上网搜索相关的解决方法进行参考。当遇到不知道解决的 bug 时，最好及时找组内的大佬来帮忙，而不是拖到小组讨论的时候再解决，毕竟小组讨论不是用来帮忙 debug 的时间。从这次开发中我认识到，出现 bug 是不可避免的事情，重要的是从 bug 中能学习到很多内容。首先是在经历了很多 bug 以后，就逐渐能从报错信息中学习到错误出现的原因与位置，逐渐就学会了自己解决相应的 bug。同时 bug 的出现也在一定程度上让我认识到自己在编程时没有注意到的内容，通过修复 bug 也让我对 django 等内容有了一个新的认识，可以说是很好的从错误中学习的经历。

这次开发过程也让我明白了先思考，再动手写代码的重要性。很多时候如果着急想要写代码，往往结果会是适得其反。因为根本没有清晰的思路，所以很多时候只是像无头苍蝇一样乱撞。通过这次软工开发，我养成了先思考再动手写代码的习惯，先静下心来把要实现的

功能想清楚，再动手去实现，这种方式使得我的效率提升了很多，同时也减少了很多做无用功而浪费掉的时间。当我遇到问题的时候，我也会先想清楚问题出现的原因，抓准了问题的根源就容易找到解决的方法，而避免了在不经思考地尝试过程中浪费的时间与精力。

这次开发过程也让我认识到了严谨的重要性。因为在修改前端页面的时候，改了 index.html 和 head.html 以后对于继承这两个模板的 html 文件都会造成影响，所以后续对其他页面的修改也花费了很多精力，几乎是把全部页面都进行了一定的修改。这个过程也让我认识到了严谨的重要性，如果不够细致的话就难免会在某些页面留下不协调的地方，所以要非常仔细地进行修改。

还有这次开发让我认识到了模仿与独立编程之间的巨大区别。刚开始我们是按照书上的内容进行学习，很多地方的坑都是作者已经踩过所以我们在学习书中内容的时候就很难会遇到类似的坑。但是当自己进行开发的时候就很容易会踩坑。所以说，根据教程进行操作只是一定程度上模仿他人的经验，重要的是要形成自己对这件事情的理解，这样在后面才能为我所用，而不是离开了教程就无所作为。另外，在刚进入前端进行开发的时候，我只是了解了基本的 css 和 js 语法。很多内容都是从手里的 css 模板代码中学到的。虽然这个过程比较艰难，仅仅是一个导航栏就花了我很久的时间。但是这个过程中经历的弯路却也为铺平了后面的道路，因为我形成了自己的认识，这样后面的开发就变得相对轻松了许多。

七．结语

这一门软件工程确实让我学到了很多东西，不仅仅是 django、html、css、javascript 等技术方面的内容，还让我在开发中与同伴沟通的技巧等软实力方面有了提升，远非一学分所能代表。虽然整个过程中经历了很多困难，但是非常感谢和我老师、助教以及同组的同伴们给予我的帮助。正所谓苦尽甘来，相信这次软件工程中学到的内容会成为日后宝贵的财富。