# 课程报告

施正祎 37220232203805

学习内容包括：

网页相关：Flask，HTML，CSS，JS

爬虫相关：Request，bs4，os，re

AI调试和调用：coze自组建电影相关的bot，coze官网的API文档

|  |
| --- |
| **README****作业文件结构如下：****作业movie\_list v1.0**  * movie\_app.py * .vscode/   + settings.json * scrapy/   + douban\_images/     - ...   + douban\_txts/     - ...   + 获取豆瓣Top250.py * templates/   + index.html   注：请不要对上述文件结构以及文件命名做任何修改，可能导致程序无法正常运行  两处python文件功能相互独立： 网页相关功能——./movie\_app.py 爬虫——./scrapy/获取豆瓣Top250.py **网页功能**  1. 前后切换电影 2. 标记为看过和取消标记 3. 独立聊天窗口，与ai交互  **目前存在的一些待解决的问题**  1. 标记没能实现永久存储，仅仅借用网页缓存达到展示效果 2. 网页页面不够美观，有待继续优化 3. 信息存取模式没有完成一体化，所有数据都借助本地存储后展示在前端 4. coze提供api访问方式存在次数限制、访问速度慢等问题 5. 目前的ai模型没有很好的符合要求，往往存在幻觉答非所问 6. 网页结构单一，仅一个界面 7. html文件内容没有对css和js做拆分，比较冗长  **希望加入的功能**  1. 影片影评的搜索功能、评价打分以及评论功能 2. 优美的UI界面设计，合适网页背景以及合理的模块划分 3. 网页功能多样化后，还需要精心设计的网页结构  **一些感想** 第一次尝试写一个具有图形界面可以和用户简单交互的程序，虽然最后看来效果并不是很好，但是在过程中我收获了很多。 很多似乎简单的问题，在代码实现中并不轻松，例如：   1. 为了解决“看过”标记的问题，需要学习js编写有关按钮反应的功能函数，为了实现标记的永久存储还有很多学问。 2. 为了完成一个网页中自由展开回收的独立聊天窗口，需要css对网页样式进行调整 3. 为了达到独立聊天窗口的实际效果，解决网页用户的文字输入到AI模型回答的网页展示是一个比原本想象中更为复杂的事情，思路为： 网页接受用户消息——发送POST请求至后端——由py请求并获取结果——借助路由返回前端——网页展示输出 4. 关于1中的永久存储思路上应该与3相似，但是仅仅为了实现展示效果，我仅用网页缓存实现 5. Coze的API文档写的非常不完善，仅仅写了shell中的curl指令方法，在python中的请求方法需要靠以往经验和文档介绍编写，相对麻烦   即便是这样一个简单的项目，似乎一个人完成也是相对吃力的，大概这就是项目团队的必要性吧 感觉这个小学期的学习过程真的非常奇妙，对于一门全新的python语言，短短四周中我们从一无所知，到上机写算法题，再到完成简单的项目，成就感也是不言而喻的。 |