一、爬虫及索引的建立（周小舟,516030910492）

爬虫的目标网页为 P 站（www.pixiv.net）。爬虫最主要的困难在于 P 站的防盗链。直接进入图片的发布页后，可以得到图片链接、图片访问量及收藏量、图片的标签化信息等数据，但在图片发布页获取到的图片链接并不能直接被访问。为此需先进行模拟登陆，模拟登陆时需要获得动态的“post_key”信息，每次登录时需单独获取。获取发布页的“referer”之后即可获取图片页信息，也可将图片保存到本地。由于 P 站是一个开放性的插画交流社区，门槛较低，因此需要对获取到的图片进行筛选，保留相对来说质量更由的图片。此外，还需对图片的内容加以区分，P 站上有着数量颇多的 18 禁内容（某些岛国画师的奇特口味…）需要加以去除。P 站并没有十分严格的反爬虫机制，只需每次获取图片后短暂休眠即可抑制，之前也尝试过通过更换动态 IP 来抑制反爬，不过效果并不明显于是放弃。整个过程中由于爬虫本身对网页数据进行了相当程度的处理，后续建立索引的工作也相应变得简单了。P 站上活跃着大量的某岛国的画师，相应的需要对语言进行处理。最初我们的设想是在建立索引时将必要的文字由日语翻译为中文再加以保存，但效果不佳，后续更正为将用户的输入翻译为日语。此处接入了有道的翻译 API，使用者在查询时可使用多种语言输入。

二、网站与数据库的建立(周澜轩，516030910491)

1. 数据库的建立

由于我们的搜索引擎需要通过用户搜索记录来对用户进行图片推荐，所以我们需要使用数据库来记录用户的搜索记录。数据库使用 MySQL 来建立，名称为 demo，创建一个名为 UserInfo 的 table，字段包括 UserName（用户名）、Password（密码）和 Info（搜索记录），如下图所示：

```
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| UserName | char(20) | NO   |     | NULL    |       |
| Password | char(20) | NO   |     | NULL    |       |
| Info     | longtext | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
```

用户可通过注册的方式来记录其用户名和密码，登录后数据库会自动更新用户的搜索记录，例如一个用户名为 SunKnight、密码为 123456 的用户，在搜索了 fgo（一款游戏）和 clannad（一个动画）后，其数据库内容会更新如下：

```
+-----------+----------+-----------------+
| UserName  | Password | Info            |
+-----------+----------+-----------------+
| SunKnight | 123456   |    fgo clannad  |
| MasonZLX  | 123456   |                 |
| Jerry     | 123456   |                 |
+-----------+----------+-----------------+
```

2. 语言的选择—PHP

在这次的大作业中我选择了 PHP 来进行网站的编写，而非平时上课所用发的 python 和 web 原因如下：

1、 网上和图书馆里有关 php 进行网页编程的参考教材偏多，而 python 较少，且图书馆中有关 python web 编程的数已被借完，无奈之下我只好自学 php；

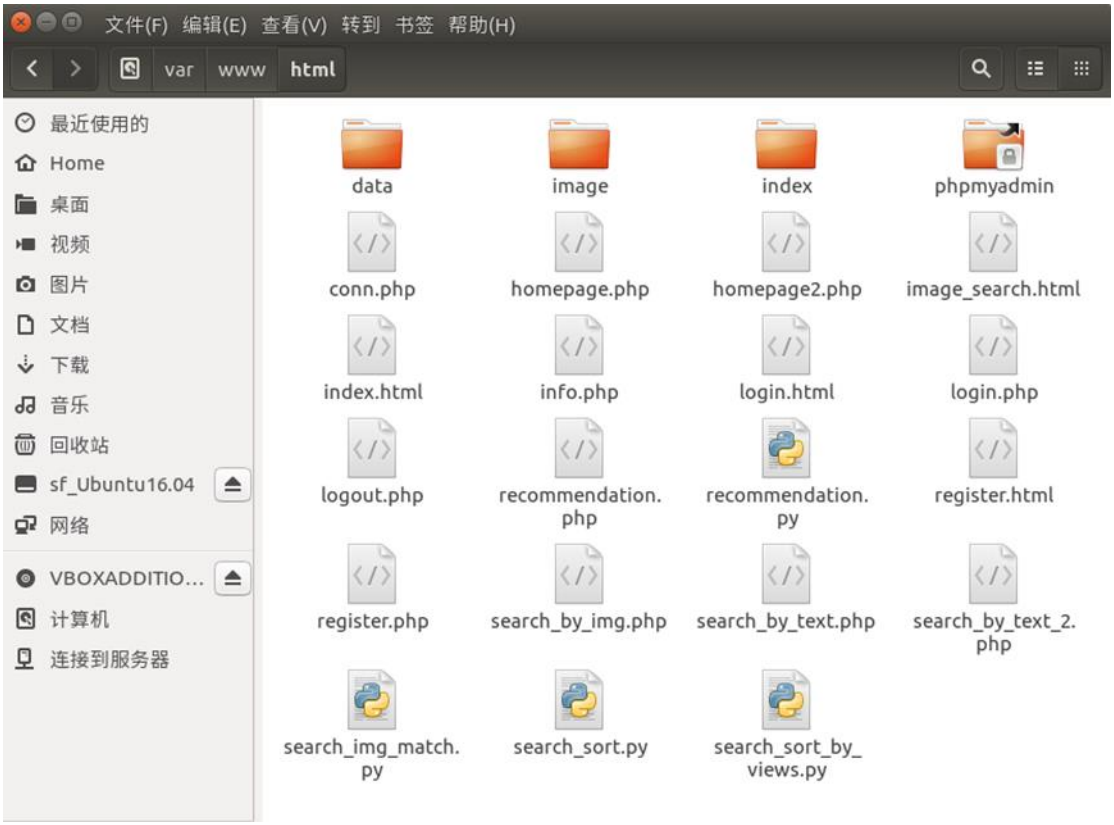
2、 php 文件可以直接编写 html，对于网页设计比较简便；

3、 对于管理 MySQL 而言，php 要比 python 简单一些；

4、 php 可以直接通过 exec 函数来给 python 文件传参，并以字符串的形式获得 python 文件的处理结果；

但实际上，在后期进行整合的时候，我发现 php 与 python 在兼容上仍存在一些问题，最突出的便是在/var/www/html 文件夹中，php 通过 exec 函数来运行 python 时无法 import lucene 和 cv2，最终我的解决方法是使用 pip 来重新安装 lucene 和 opencv。

3. 项目文件总览



其中 phpmyadmin 文件夹和 info.php 不是项目内容。

4. 文件功能详解

1、conn.php：

连接数据库的 php 文件，由于很多 php 文件都需要连接数据库，所以我将这一部分放在了 conn.php 中，由其他 php 文件来调用；

2、homepage.php、homepage2.php：

分别为“按综合排序搜索”和“按访问量搜索”的主界面，用户登录与不登录是导航栏会有区别；

3、login.html、login.php：

login.html 为登陆页面，login.php 为处理用户登录信息的 php 文件；

4、register.html、register.php：

register.html 为注册页面，register.php 为处理用户登录信息的 php 文件；

5、logout.php：

注销用户信息用的 php 文件；

6、search_by_text.php、search_sort.py：

可以根据用户输入的文本信息来得到“按综合排序搜索”结果的 Python 文件，php 文件调用 Python 文件来输出相应界面；

7、 search_by_text2.php、search_sort_by_views.py：
可以根据用户输入的文本信息来得到“按访问量排序搜索”结果的 Python 文件，php 文件调用 Python 文件来输出相应界面；

8、 image_search.html、search_by_img.php、search_img_match.py：
image_search.html 为“以图搜图”的网页，search_img_match.py 可以通过图片匹配得到相应的结果，search_by_img.php 允许用户上传本地图片，并通过调用 search_img_match.py 来输出结果；

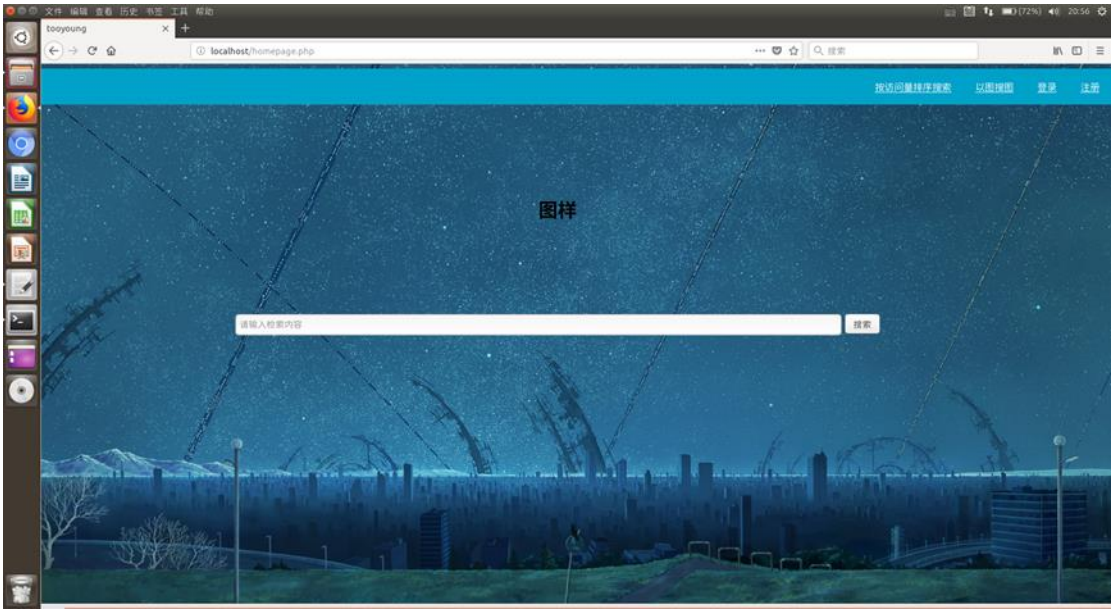
9、 recommendation.php、recommendation.py：
通过搜索记录来得到“猜你喜欢”结果的 Python 文件，php 可以得到用户的搜索记录（需要用户登录），并通过调用 Python 文件来输出结果；

10、 data 文件夹：
保存了 P 站爬取的图片；

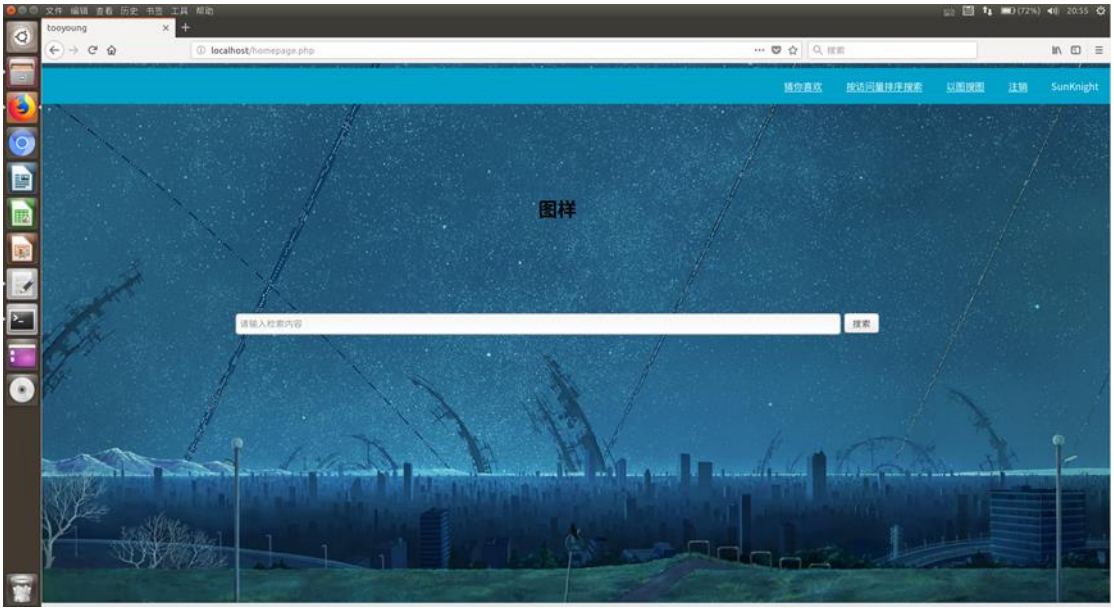
11、 image 文件夹：
用于存放网站的背景图和 logo；

12、 index 文件夹
创建索引的文件夹；

5. 登录前后网页效果预览（以“按综合排序搜索”为例）：
登录前：



登陆后：



6. 注意事项：

   测试代码时请按照“1、数据库的建立”中的内容来创建合适的数据库，并将代码放入 localhost 根目录中，通过 localhost/homepage.php 进入。

三、 搜索推荐部分（颜铭萱，516030910488）

排序算法：

在我们的搜索引擎中我们为用户提供两种不同的排序方式 :按图片浏览记录排序和综合排序。按图片浏览量排序即根据 index 文件内的图片的浏览量对与用户搜索相匹配的结果进行倒序排序, 最先返回浏览量最高的图片结果 ;综合排序即预先给所有的图片设定一个评分，根据从 P 站上爬取到的图片结果，我们利用图片的浏览记录和收藏记录为图片建立评分模型，即：

$$\text{图片评分} = 0.6 \times \text{收藏次数} + 0.4 \times \text{浏览次数}$$

这个模型侧重于图片的收藏次数，因为图片的收藏次数越多，证明该图片的质量越高、用户主观评价越高。根据评分模型返回与用户搜索相匹配的结果中评分最高的图片组。

推荐算法：

由于我们的搜索引擎可以给用户提供注册账户的功能, 所以我们进而为用户提供了推荐算法，即猜你喜欢功能。用户在登陆后，其搜索记录都会被记录在 MySQL 中，推荐算法获取到当前用户的搜索记录，提取出其中的关键词，在 index 文件中地图片标签内容对这些关键词进行匹配, 分析这些关键词的对应标签, 获得一个关键词标签集及这些标签的出现频数。提取出标签集中标签出现频数最高的 3 个标签，根据用户的收藏率（即收藏量 / 浏览量）每个标签任意获取 200 个结果中返回给用户 10 个收藏量最高的图片，共计 30 个用户可能喜欢的图片，而且这些图片也都是其他用户评价等级较高的图片（收藏率较高），以达到推荐目的。

图片匹配：

由于动漫图片、插画的特殊性，不同画师的风格可能相似、同一画师的不同作品风格相似，使得人们在看这些漫画作品时难免会出现“脸盲”，无法分清不同的人物；所以在图像匹配部分我们放弃使用 SIFT 算法以尽可能减少结果与用户预期搜索结果匹配度过低的情况发生，而选择特征向量对图片进行匹配，尽可能从整个图片的布局和颜色风格上返回给用户最

匹配的结果。