

# 实验报告

---

## 一、网页系统功能

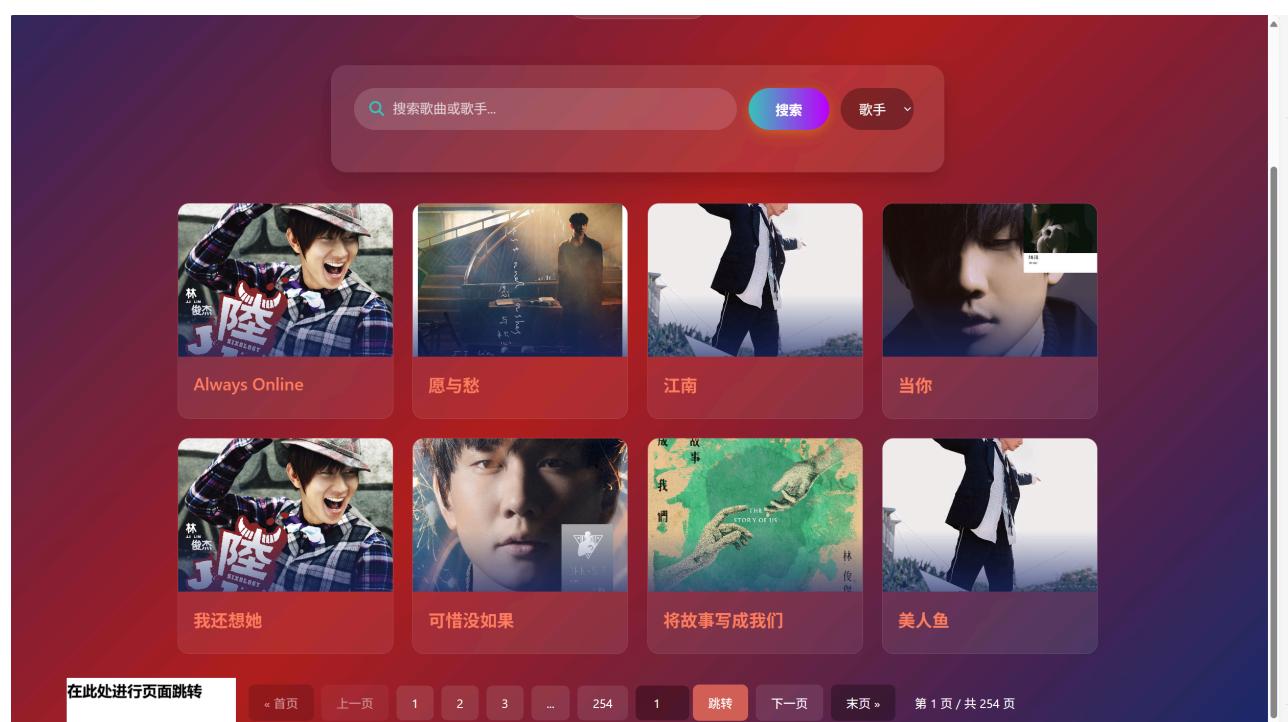
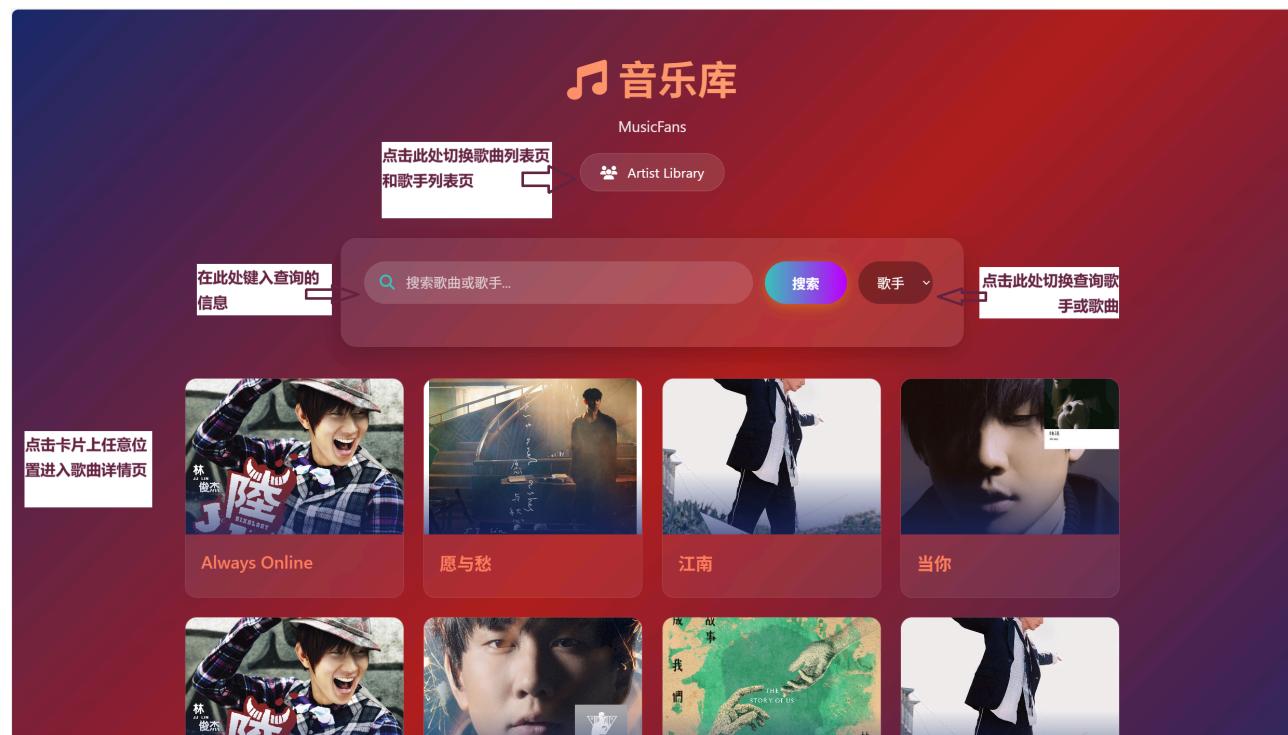
### (i) class

```
1 from django.db import models
2 from django.utils import timezone
3
4
5 class Song(models.Model):
6     num = models.IntegerField()
7     name = models.CharField()
8     artist = models.CharField()
9     lyric = models.CharField()
10    url = models.CharField()
11    pic_url = models.CharField()
12    artist_url = models.CharField()
13
14
15 class Singer(models.Model):
16     num = models.IntegerField()
17     name = models.CharField()
18     pic_url = models.CharField()
19     desc = models.CharField()
20     url = models.CharField()
21
22
23 class Comment(models.Model):
24     num = models.IntegerField()
25     context = models.CharField()
26     created_at = models.DateTimeField(default=timezone.now)
27
```

歌曲类成员变量有num(编号), name(歌曲名), artist(歌手名), lyric(歌词), url(原网站链接), pic\_url(歌曲图片url), artist\_url(歌手图片的url)。歌手类成员变量有num(编号), name(歌手名), pic\_url(图片的url), url(原网站链接), desc(歌手简介)。评论类成员变量有num(对应歌曲的编号), context(正文), created\_at(时间)。

## (ii) 页面

### 1. 主页 (歌曲列表页)



该页面为歌曲列表页。点击“Artist Library”可以切换到歌手列表页。此页面上具有搜索功能，输入非空搜索内容后点击搜索之后的画面就是搜索结果页。可以从此页面跳转到歌手列表页和歌曲详情页。分页跳转功能支持输入页数和按键增加减少两种方式。背景为45度的渐变颜色，搜索按钮颜色为初音未来色和清华紫的渐变。图标样式来自于font-awesome。

```
1 def show_mainpage(request):
2     """歌曲详情页
3
4     每页显示8首歌曲，如果用户输入的页数不是整数就跳转到第一页，如果超出范围则跳
5     转到最近的页数
6     """
7     song_list = Song.objects.all()
8     paginator = Paginator(song_list, 8)
9
10    page = request.GET.get("page")
11    try:
12        songs = paginator.page(page)
13    except PageNotAnInteger:
14        songs = paginator.page(1)
15    except EmptyPage:
16        songs = paginator.page(paginator.num_pages)
17
18    base_url = "/index/MUSIC"
19    return render(
20        request,
21        "index.html",
22        {"songs": songs, "base_url": base_url, "MEDIA_URL": settings.MEDIA_URL},
23    )
```

获得所有歌曲，并进行分页，解析index.html并传入一些参数，base\_url为跳转服务，MEDIA\_URL为本地资源的文件夹

## 2、歌曲详情页

The screenshot shows a song detail page for 'Always Online' by JJ Lin. At the top, there's a photo of JJ Lin wearing a hat and plaid shirt, with a red graphic overlay. Below the photo, the song title 'Always Online' is displayed in large letters, followed by the artist name '林俊杰' and a link to the singer's detail page. There are also links to the original website and back buttons.

**歌词**

作词 : 林怡凤  
作曲 : 林俊杰  
编曲 : 蔡政勋/陈建玮  
制作人 : 林俊杰

变色的生活 任性的挑拨  
疯狂的冒出了头  
单方的守候 试探的温柔  
还是少了点什么

遥远两端 爱挂在天空飞  
风停了也无所谓 只因为你总说  
Evervthing will be okav

**评论** 在此处提交评论，评论以倒序列表显示在下方

写下你的评论...

提交评论

2 评论时间: July 6, 2025, 5:56 p.m.

1 评论时间: July 6, 2025, 5:56 p.m.

This screenshot shows the same song detail page for 'Always Online' by JJ Lin, but with a different color theme for the header and sidebar.

**歌词**

作词 : 林怡凤  
作曲 : 林俊杰  
编曲 : 蔡政勋/陈建玮  
制作人 : 林俊杰

变色的生活 任性的挑拨  
疯狂的冒出了头  
单方的守候 试探的温柔  
还是少了点什么

遥远两端 爱挂在天空飞  
风停了也无所谓 只因为你总说  
Evervthing will be okav

**评论**

写下你的评论...

提交评论

2 评论时间: July 6, 2025, 5:56 p.m.

1 评论时间: July 6, 2025, 5:56 p.m.

此处显示总评论数 2 评论数

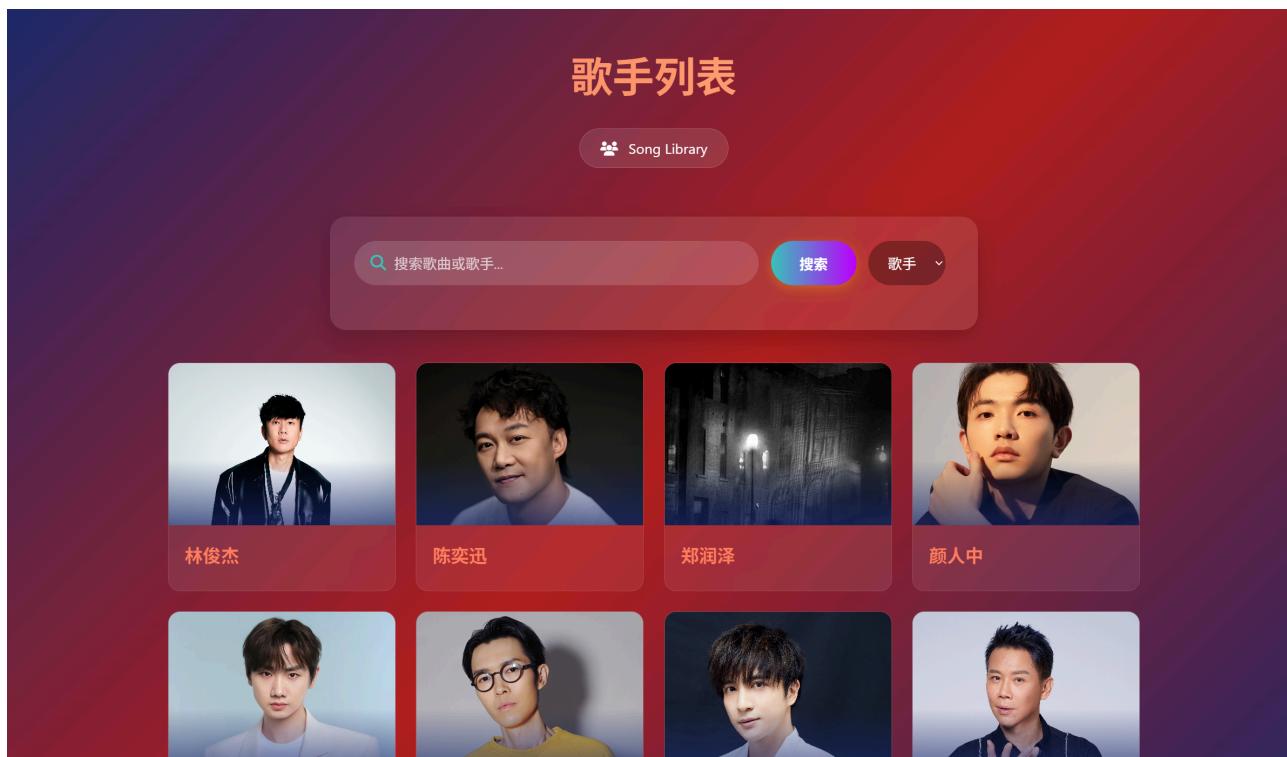
此页面点击歌手名可以跳转到歌手详情页，点击url跳转到原网站，点击Back返回歌曲列表页。评论提交内容不能为空。

```
1 def show_song(request, id):  
2     """展示歌曲信息，在数据库中滤出这首歌的所有评论数据"""  
3     song = Song.objects.get(id=id)  
4     artist = Singer.objects.get(name=song.artist)
```

```
5     artistid = artist.id
6     comments = Comment.objects.filter(num=id)
7
8     return render(
9         request,
10        "index1.html",
11        {
12            "song": song,
13            "MEDIA_URL": settings.MEDIA_URL,
14            "artistid": artistid,
15            "comments": comments[::-1],
16        },
17    )
```

获取这一首歌曲，及对应的评论，转换为倒序列表作为参数，解析index1.html

### 3、歌手列表页



页面功能和歌曲列表页一样

```
1 def show_artistlist(request):
2     """歌手列表页
```

```
3
4      每页显示8位歌手，如果用户输入的页数不是整数就跳转到第一页，如果超出范围则跳
5      转到最近的页数
6      """
7      singer_list = Singer.objects.all()
8      paginator = Paginator(singer_list, 8)
9
10     page = request.GET.get("page")
11     try:
12         singers = paginator.page(page)
13     except PageNotAnInteger:
14         singers = paginator.page(1)
15     except EmptyPage:
16         singers = paginator.page(paginator.num_pages)
17
18     base_url = "/index/MUSIC/artist"
19
20     return render(
21         request,
22         "index2.html",
23         {"singers": singers, "base_url": base_url, "MEDIA_URL": settings.MEDIA_URL},
24     )
```

## 4、歌手详情页

林俊杰

https://music.163.com/artist/desc?id=3684 点击url跳转到原歌手网页

Back to Artist Library 点击跳转到歌手列表页

简介 在此处显示简介

JJ林俊杰的创作来自最深的情感，他的声音唱出灵魂的璀璨，他把音乐和梦想当做能量，一路走到无人取代的地位，他写下华语乐坛最动人的经典乐章，撼动亚洲数十亿颗心跳。他是亚洲乐坛全能唱作天王 JJ 林俊杰。

2003年首发第一张个人创作专辑《乐行者》，取得不俗成绩；其杰出的创作才能又在之后2004年的凭借歌曲【江南】而成名，并于同年获得第15届金曲奖之「最佳演唱新人奖」。随后的【小酒窝】、【曹操】、【她说】等歌曲亦造成广大回响。2011年8月8日携手华纳，迈出世界。

2020至2021年【幸存者·如你】双维度EP，创造全新音乐视角。由JJ 林俊杰亲自主导整张专辑的企划创意与视觉，新专辑一推出便占据大中华区各大排行榜，销售量更在一个月内突破百万。

把音乐和梦想当做能量，一路走到无人取代的地位，写下华语乐

点击歌曲图片或者是歌曲名都可以跳转到歌曲详情页

歌手歌曲 详情页

Always Online 愿与愁 江南 当你

我还想她 可惜没如果 将故事写成我们 美人鱼

修炼爱情 心墙 不潮不用花钱 加油！

点击url跳转到原歌手网页，点击Back返回歌手列表页。点击歌曲封面和歌曲名可以跳转到歌曲列表页

```
1 def show_singer(request, id):
2     """展示歌手信息，在数据库中滤出歌手的歌曲"""
3     singer = Singer.objects.get(id=id)
4     song_list = Song.objects.filter(artist=singer.name)
5
6     return render(
7         request,
8         "index3.html",
9         {"singer": singer, "MEDIA_URL": settings.MEDIA_URL,
10          "song_list": song_list},
```

## 5、搜索结果页



与歌曲列表页和歌手列表页一致，仅增加了搜索结果与搜索时间。点击对应卡片可以跳转到详情页

```
1 def search(request):
2     """搜索
3
4     搜索时间为filter的查询时间
5     通过传进来的type参数，进行不同的搜索，并且解析不同的html页面
6     """
7     query = request.GET.get("q")
8     search_type = request.GET.get("type")
9     if search_type == "artist":
10         start_time = time.time()
11         singer_list = Singer.objects.filter(
12             Q(name__contains=query) | Q(desc__contains=query)
13         )
14         end_time = time.time()
15         lens = len(singer_list)
16
17         paginator = Paginator(singer_list, 8)
18
19         page = request.GET.get("page")
20         try:
21             singers = paginator.page(page)
```

```
22     except PageNotAnInteger:
23         singers = paginator.page(1)
24     except EmptyPage:
25         singers = paginator.page(paginator.num_pages)
26
27     base_url = "/index/search?type=artist&q=" + query
28
29     return render(
30         request,
31         "index4.html",
32         {
33             "singers": singers,
34             "base_url": base_url,
35             "MEDIA_URL": settings.MEDIA_URL,
36             "query": query,
37             "time": "{:.5f}".format(end_time - start_time),
38             "len": lens,
39         },
40     )
41
42     if search_type == "song":
43         start_time = time.time()
44         song_list = Song.objects.filter(
45             Q(name__contains=query)
46             | Q(artist__contains=query)
47             | Q(lyric__contains=query)
48         )
49         end_time = time.time()
50         lens = len(song_list)
51         paginator = Paginator(song_list, 8)
52
53         page = request.GET.get("page")
54         try:
55             songs = paginator.page(page)
56         except PageNotAnInteger:
57             songs = paginator.page(1)
58         except EmptyPage:
59             songs = paginator.page(paginator.num_pages)
60
61         base_url = "/index/search?type=song&q=" + query
62
63     return render(
```

```
64         request,
65         "index5.html",
66     {
67         "songs": songs,
68         "base_url": base_url,
69         "MEDIA_URL": settings.MEDIA_URL,
70         "query": query,
71         "time": "{:.5f}".format(end_time - start_time),
72         "len": lens,
73     },
74 }
```

通过搜索页面传进来的类型参数在数据库中搜索结果，搜索时间认为是filter函数的时间，保留5位小数，len为结果的数量

## 二、数据量

歌曲：2026首；歌手100位

## 三、使用的技术与算法

(1) 使用songimporter.py和singerimporter.py向数据库中导入数据

```
1 import os
2 import sys
3 import csv
4 import django
5
6
7 BASE_DIR =
8     os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
9     sys.path.append(BASE_DIR)
10    os.environ.setdefault("DJANGO_SETTINGS_MODULE",
11        "project1.settings")
12    django.setup()
13
14    from MUSIC.models import Singer
```

```
14
15 def func():
16     """将歌手信息导入数据库"""
17
18     singerfile =
19         open(r"C:\Users\14395\Desktop\git\MusicInfo\singer.csv", "r")
20
21     singreader = csv.DictReader(singerfile)
22     for row_num, row in enumerate(singreader):
23         if row_num % 2 == 0:
24             continue
25
26         singer = Singer(
27             num=row["artist_id"],
28             name=row["artist_name"],
29             pic_url=row["artist_pic_url"],
30             desc=row["artist_desc"],
31             url=row["artist_url"],
32         )
33
34
35
36     func()
37     print(f"成功导入 {singer.objects.count()} 个歌手")
38
```

```
1 import os
2 import sys
3 import csv
4 import django
5
6
7 BASE_DIR =
8     os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
9     sys.path.append(BASE_DIR)
10    os.environ.setdefault("DJANGO_SETTINGS_MODULE",
11        "project1.settings")
12    django.setup()
13
14
15 from MUSIC.models import Song
```

```

13
14
15 def func():
16     """将歌曲信息导入数据库"""
17     csvfile =
18         open(r"C:\Users\14395\Desktop\git\MusicInfo\song.csv", "r")
19         singerfile =
20             open(r"C:\Users\14395\Desktop\git\MusicInfo\singer.csv", "r")
21             reader = csv.DictReader(csvfile)
22             singreader = csv.DictReader(singerfile)
23             for row_num, row in enumerate(reader):
24                 if row_num % 2 == 0:
25                     continue
26
27                 tmp = row["artist_name"]
28                 singer_url = ""
29                 for hang in singreader:
30                     if hang["artist_name"] == tmp:
31                         singer_url = hang["artist_url"]
32                         break
33
34                 song = Song(
35                     num=row["song_id"],
36                     name=row["song_name"],
37                     artist=row["artist_name"],
38                     lyric=row["lyric"],
39                     url=row["song_url"],
40                     pic_url=row["song_pic"],
41                     artist_url=singer_url,
42                 )
43                 song.save()
44
45 func()
46 print(f"成功导入 {Song.objects.count()} 首歌曲")

```

## (2)评论采用网页重定向

```

1 def comment(request, id):
2     """评论函数"""
3     content = request.POST.get("content")

```

```
4     obj = Comment(num=id, context=content)
5     obj.full_clean() # 进行数据验证
6     obj.save()
7     return HttpResponseRedirect(f"/index/MUSIC/{id}")
8
9
10    def delcomment(request, id):
11        """删除评论"""
12        iden = request.POST.get("delete_comment")
13        comment = Comment.objects.get(id=iden)
14        comment.delete()
15        return HttpResponseRedirect(f"/index/MUSIC/{id}")
```

### (3) 利用jieba库处理歌词

```
1 def segment_text(text):
2     """由于歌曲的信息各不相同，所以前面处理歌词的函数会有遗漏。手动根据词云
3      结果添加停用词
4
5     Keyword arguments:
6         text -- 字符串
7
8     返回处理后的字符串
9
10    stopwords = [
11        "母带",
12        "h3R3",
13        "总监",
14        "录音",
15        "制作",
16        "录音师",
17        "作词",
18        "出品人",
19        "网易",
20        "法老",
21        "陶喆",
22        "Producer",
23        "编曲",
24        "SBMS",
25        "配唱",
26        "版权",
```

26 "国际",  
27 "弦乐",  
28 "贝斯",  
29 "Studio21A",  
30 "统筹",  
31 "录音室",  
32 "编写",  
33 "键盘",  
34 "NEWBAND",  
35 "公司",  
36 "音频编辑",  
37 "钢琴",  
38 "维伴",  
39 "首席",  
40 "乐队",  
41 "音响",  
42 "原唱",  
43 "编曲",  
44 "作曲",  
45 "Ltd",  
46 "打击乐",  
47 "Sound",  
48 "rapper",  
49 "Music",  
50 "混音",  
51 "牛班",  
52 "吉他",  
53 "音乐",  
54 "Studio",  
55 "Engineer",  
56 "Publishing",  
57 "何飚",  
58 "don",  
59 "PGM",  
60 "Vocal",  
61 "有限公司",  
62 "工作室",  
63 "林俊杰",  
64 "队长",  
65 "Asen",  
66 "罗言",  
67 "re",

```
68     "乐团",
69     "人声",
70     "企划",
71     "MUSIC",
72     "项目",
73     "wiz",
74     "OP",
75     "录音棚",
76     "汪苏",
77     "郎梓朔",
78     "营销",
79     "发行",
80     "Program",
81     "编辑",
82     "石行",
83     "改编",
84     "工程师",
85     "爱乐乐团",
86     "监制",
87     "Mixing",
88     "说唱",
89     "SP",
90     "Mastering",
91     "Chan",
92     "张子",
93     "陈楚生",
94     "刘卓",
95     "索尼",
96     "林梦洋",
97     "设计",
98 ]
99 words = jieba.cut(text)
100 return " ".join([word for word in words if word not in
stopwords and len(word) > 1])
```

#### (4)利用adjustText库处理散点图的名称标签，避免重合

```
1 texts = []
2 for i, artist in enumerate(artists):
3     texts.append(plt.text(x.iloc[i], y.iloc[i], artist,
4                           fontsize=6, alpha=1))
5
6
7 adjust_text(
8     texts,
9     expand_points=(2, 2),
10    expand_text=(1.2, 1.2),
11    force_text=(0.5, 0.5),
12    only_move={"points": "y", "text": "y"},
```

#### (5)其他技术：pandas, matplotlib, re, requests, BeautifulSoup等技术

## 四、实验用时与感想

(1)爬虫且完整获得数据：36h

(2)网页系统设计：36h

(3)数据分析：24h

感想：这是第一个我完成的计算机技术工程，颇有成就感。从一开始畏惧网页复杂的html源码，到熟练的找到所需内容，从一开始总是写成c++到掌握python的基础语法，第一次写html构建网站都让我收获良多。总的来说，这次实验的内容非常不错，让我掌握了许多重要的技术和技能。任务量有一点小大：(

