****

**《高级编程技术》**

**课程项目实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院名称** | **：** | 数据科学与计算机学院 | | | | | |
| **课程名称** | **：** | 高级编程技术 | | | | | |
| **小组成员** | **：** | 田晓晴 16337223  顾可心 16337066  郭林杏 16337070  卢明月 16337168 | | | | | |
| **时间** | **：** | 2018 | 年 | 7 | 月 | 10 | 日 |

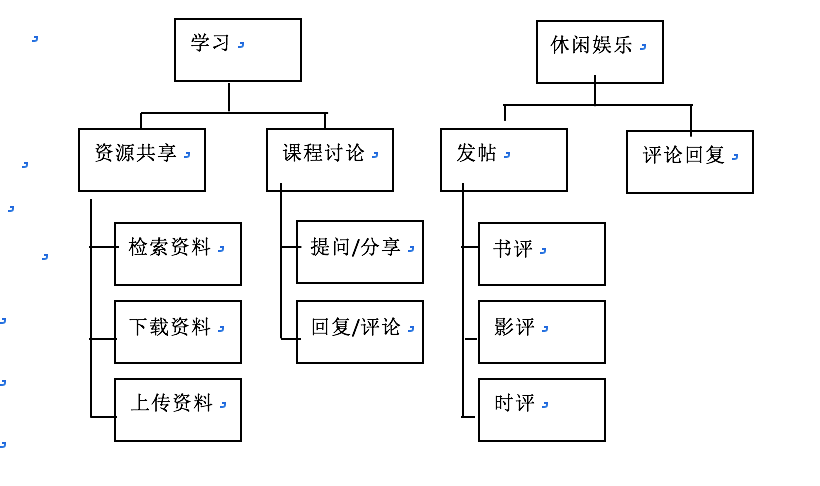
**双鸭山乐园**

**——中大学子分享交流网站**

**一、实验介绍**

1. 中大专属的资料分享与生活交流网站
2. 基于Django框架的web开发
3. 运行环境：Windows/Linux/OS X

**二、功能设计**

****

**功能描述：**

1、上传分享：用户可以上传自己手中的学习资料，如各科教材的电子版、课后 习题答案、各科历届考试试题、大神总结的学习笔记等，上传时应添加学科 标签。

2、免费下载：凡中大学生可免费下载网站上所有资料。

3、搜索功能：可按关键字搜索或按学科分类检索。

4、课程讨论：按照每学期学校开设的课程划分成不同专栏，每个专栏对应一门 课程。用户可发布求助帖或心得帖，其他同学可回复和评论。

5、休闲娱乐：开设多个专栏，如影评、书评、乐评、时评、体育、趣闻等，用户可以分享自己的看法心得，其他同学也可以留言回复讨论。

**三、主要技术**

1、通过django 搭建网站框架，创建模型，编写视图和模板

主要板块：

models: 包含要创建的各种模型（类），告诉django如何处理应用程序中存储的数据（数据库）

views:包含网页中各项功能的视图函数，供用户操作

urls: 包含要创建的网页，决定所有界面的url是什么样的

settings: 告诉django如何与系统交互以及如何管理项目

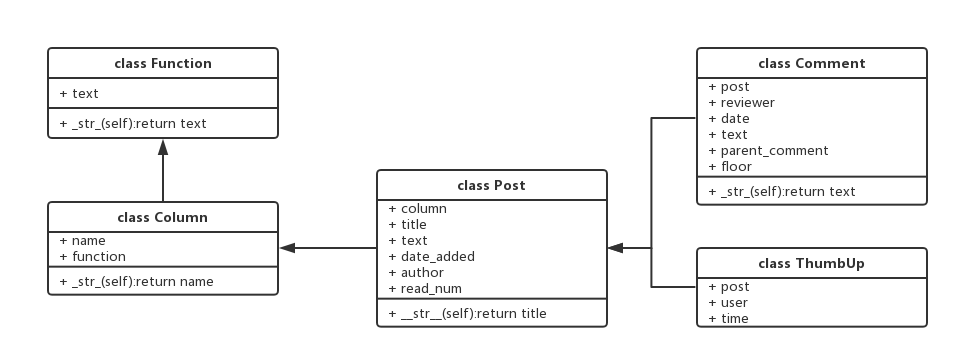
2、用户账户管理，包括注册登陆，权限管理，数据关联等

3、Bootstrap 界面优化

**四．核心代码**

**1、论坛部分（应用程序app1）**

（1）、函数模型 (app1/models.py)



（2）、视图函数（app1/views.py）：

① columns(request)*"""显示所有的栏目"""*

② column(request, column\_id):*"""显示单个栏目及其所有的条目"""*

③ new\_column(request):*"""添加新栏目""*

④ new\_post(request, column\_id):*"""在特定的栏目中建新帖"""*

⑤ edit\_post(request, post\_id):*"""编辑既有条目"""*

⑥ search\_post(request): *"""搜索帖子"""*

⑦view\_post(request, post\_id): *"""显示帖子具体内容"""*

@login\_required  
**def** view\_post(request, post\_id):  
 *"""显示帖子具体内容"""* post = Post.objects.get(id=post\_id)  
 **if not** request.COOKIES.get(**'post\_%s\_readed'** % post\_id):  
 post.read\_num += 1  
 post.save()  
 comments = post.comment\_set.order\_by(**'date'**)  
 thumbups = len(post.thumbup\_set.order\_by(**'user'**))  
 reply = 0  
 parent\_floor = 0  
 *# 未提交数据： 创建一个新表单* form = CommentForm()  
 **if** request.method == **'POST'**:  
 **if 'reply' in** request.POST:  
 *# 回复评论* reply = int(request.POST.get(**'reply'**))  
 parent\_floor = Comment.objects.get(id=reply).floor  
 **elif 'submit' in** request.POST:  
 **"""添加新评论"""** *# POST提交的数据,对数据进行处理* form = CommentForm(request.POST)  
 **if** form.is\_valid():  
 new\_comment = form.save(commit=False)  
 new\_comment.reviewer = request.user  
 new\_comment.post = post  
 new\_comment.floor = len(post.comment\_set.order\_by(**'date'**))+1  
 reply = int(request.POST.get(**'submit'**))  
 **if** reply != 0:  
 new\_comment.parent\_comment = Comment.objects.get(id=reply)  
 new\_comment.save()  
 **return** HttpResponseRedirect(reverse(**'app1:view\_post'**,

args=[post\_id]))  
 **elif 'thumbup' in** request.POST:  
 **"""点赞"""** new\_thumbup = ThumbUp()  
 new\_thumbup.post = post  
 new\_thumbup.user = request.user  
 new\_thumbup.save()  
 **return** HttpResponseRedirect(reverse(**'app1:view\_post'**,

args=[post\_id]))  
 context = {**'comments'**:comments,**'post'**: post,**'form'**: form,

**'thumbups'**:thumbups,**'reply'**: reply, **'pf'**: parent\_floor}  
 response = render(request,**'app1/view\_post.html'**,context)  
 response.set\_cookie(**'post\_%s\_readed'**% post\_id,**'true'**)  
 **return** response

（3）url链接（app1/urls.py）

urlpatterns = [  
 *# 主页* url(**r'^$'**, views.index, name=**'index'**),  
 *# 讨论区* url(**r'^discuss/$'**, views.discuss, name=**'discuss'**),  
 *# 资源库* url(**r'^resource/topics/$'**, views.resource, name=**'resource'**),  
 *# 休闲区* url(**r'^entertainment/$'**, views.entertainment, name=**'entertainment'**),  
 *# 显示所有的栏目* url(**r'^columns/$'**, views.columns, name=**'columns'**),  
 *# 特定栏目的详细页面* url(**r'^column/(?P<column\_id>\d+)/$'**, views.column, name=**'column'**),  
 *# 用于添加新主题的网页,* url(**r'^new\_column/$'**, views.new\_column, name=**'new\_column'**),  
 *# 用于添加新帖的页面* url(**r'^new\_post/(?P<column\_id>\d+)/$'**, views.new\_post, name=**'new\_post'**),  
 *# 用于编辑帖子的页面* url(**r'^edit\_post/(?P<post\_id>\d+)/$'**, views.edit\_post,name=**'edit\_post'**),  
 *# 特定文章的详细页面* url(**r'^view\_post/(?P<post\_id>\d+)/$'**, views.view\_post, name=**'view\_post'**),  
 *# 用于添加评论* url(**r'^view\_post/(?P<post\_id>\d+)/$'**, views.view\_post, name=**'new\_comment'**),  
 *#用于点赞* url(**r'^view\_post/(?P<post\_id>\d+)/$'**, views.view\_post, name=**'thumb\_up'**),  
 *# 用于回复评论* url(**r'^view\_post/(?P<post\_id>\d+)/$'**, views.view\_post, name=**'reply'**),  
 *# 搜索讨论区帖* url(**r'^search\_post/$'**, views.search\_post, name=**'search\_post'**),  
 *# 搜索休闲区帖* url(**r'^search\_post/$'**, views.search\_entertain\_post, name=**'search\_entertain\_post'**),  
]

（4）html模板(app1/templates/app1)

① base.html:父模板，导航栏

② index.html：主页

③ DiscussColumn.html,Entertainment.html：讨论区和休闲区栏目列表

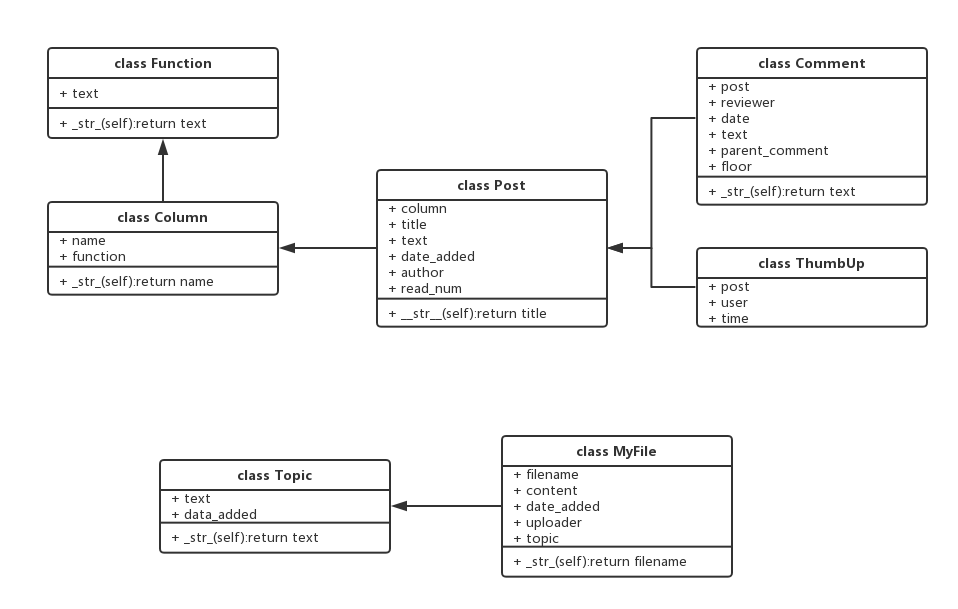
④ Column.html：特定栏目下所有帖子列表

⑤ new\_post.html,edit\_post.html：创建、编辑帖子

⑥ view\_post.html：显示帖子具体信息，包括标题、作者、创建时间、阅读量、 正文；并且可点赞文章、评论。

**2、资源部分(**app\_name = resource)

（1）函数模型（resource/models.py）



（2）视图函数（resource/views.py）

* topics(request) #显示所有主题
* topic(request, topic\_id) #显示特定主题和该主题包含的所有文件
* download(request, file\_id) #下载文件
* upload(request) #上传文件
* upload\_success(request) #上传成功界面
* search(request) #搜索文件

部分函数详细代码：

@login\_required  
**def** download(request, file\_id):  
 myfile = MyFile.objects.get(id=file\_id)  
 the\_file\_name = myfile.filename  
 file\_path = **'upload/data/'** + the\_file\_name  
  
 **def** read\_file(filename, chunk\_size=512):  
 **with** open(filename, **'rb'**) **as** f:  
 **while** True:  
 c = f.read(chunk\_size)  
 **if** c:  
 **yield** c  
 **else**:  
 **break** response = StreamingHttpResponse(read\_file(file\_path))  
 response[**'Content-Type'**] = **'application/octet-stream'** response[**'Content-Disposition'**] = **'attachment; filename=%s'** % myfile.filename  
 **return** response  
  
@login\_required  
**def** upload(request):   
 **if** request.method == **'POST'**:   
 form = MyFileForm(request.POST,request.FILES)  
 **if** form.is\_valid():  
 new\_file = form.save(commit=False)  
 new\_file.uploader = request.user  
 new\_file.save()  
 **return** HttpResponseRedirect(reverse(**'resource:upload\_success'**))  
 **else**:  
 form = MyFileForm()  
 **return** render(request, **'resource/upload.html'**, {**'form'**:form})

**def** search(request):  
 **if** request.method == **'GET'**:  
 file\_name = request.GET.get(**'search'**)  
 **try**:  
 results = MyFile.objects.filter(filename\_\_icontains=file\_name)  
 **except** MyFile.DoesNotExist:  
 results = None  
 **return** render(request, **'resource/search.html'**, {**'myfiles'**: results})  
 **else**:  
 **return** render(request, **'resource/search.html'**, {})

（3）url设置（resource/urls.py）

urlpatterns = [

path('topics', views.topics, name='topics'),

path('topics/(<topic\_id>\d+)/', views.topic, name='topic'),

path('download/(<file\_id>\d+)/', views.download, name='download'),

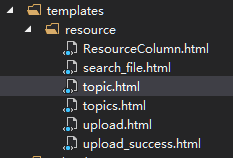
path('upload/', views.upload, name='upload'),

path('upload/success/', views.upload\_success,name='upload\_success'),

path('search\_file/', views.search\_file, name='search\_file'),

]

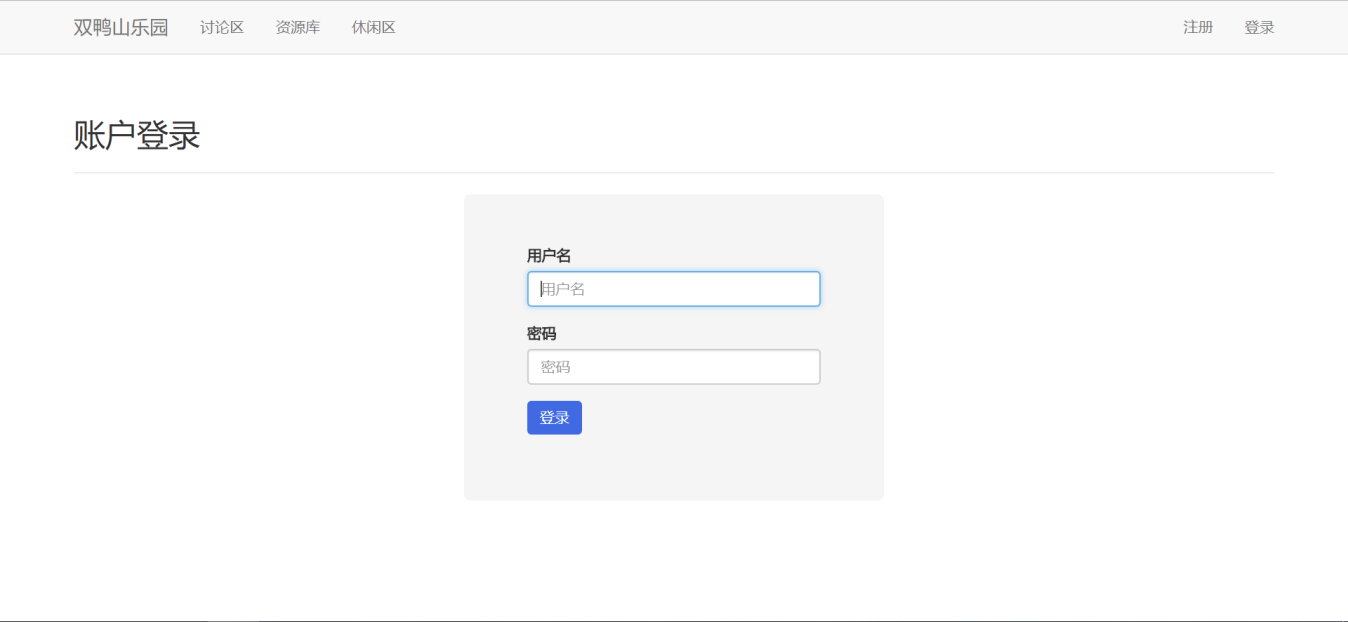
（4）html模板（resource/templates/resource）



**五、效果展示**

1. 登录网站

（1）登录页面



1. 首页

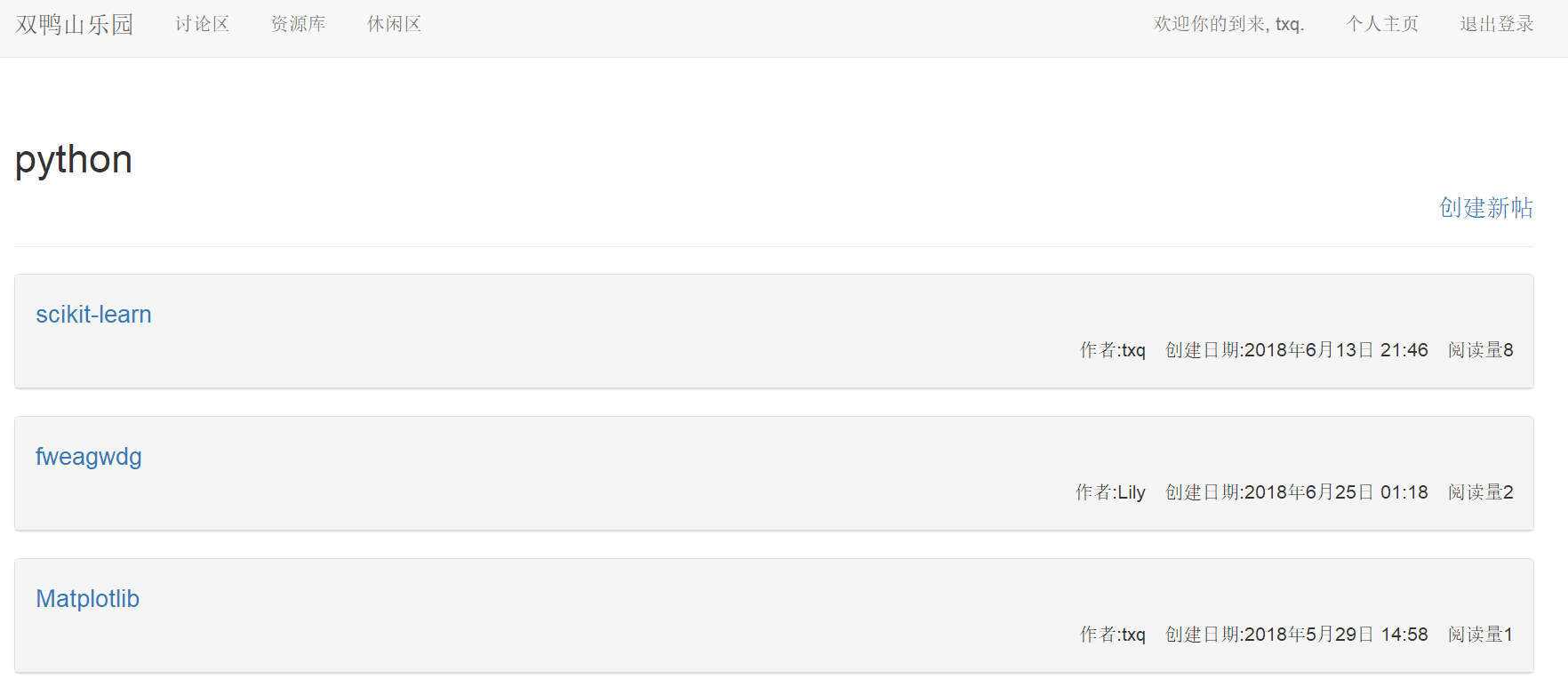


1. 讨论区

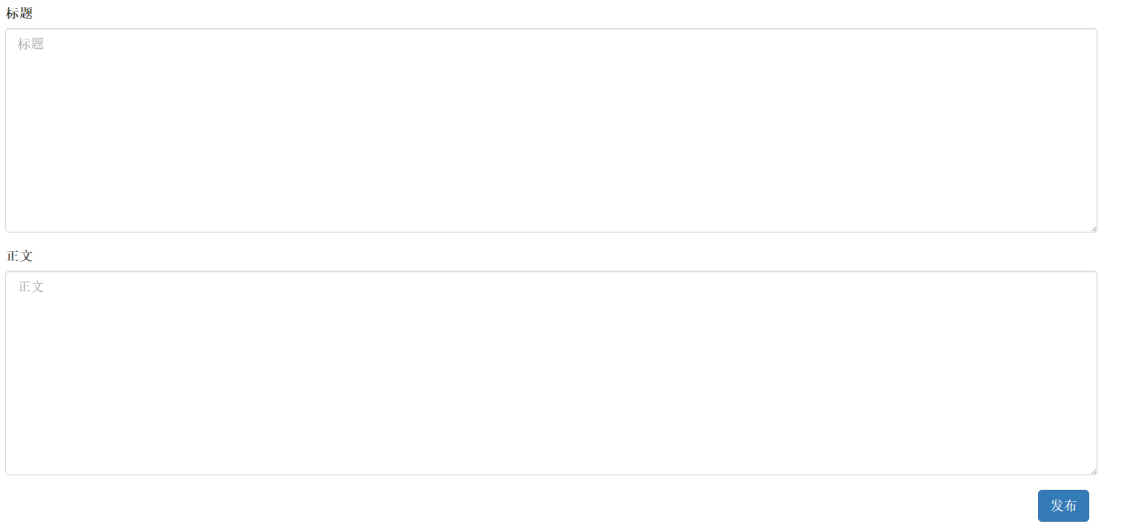
（1）课程列表



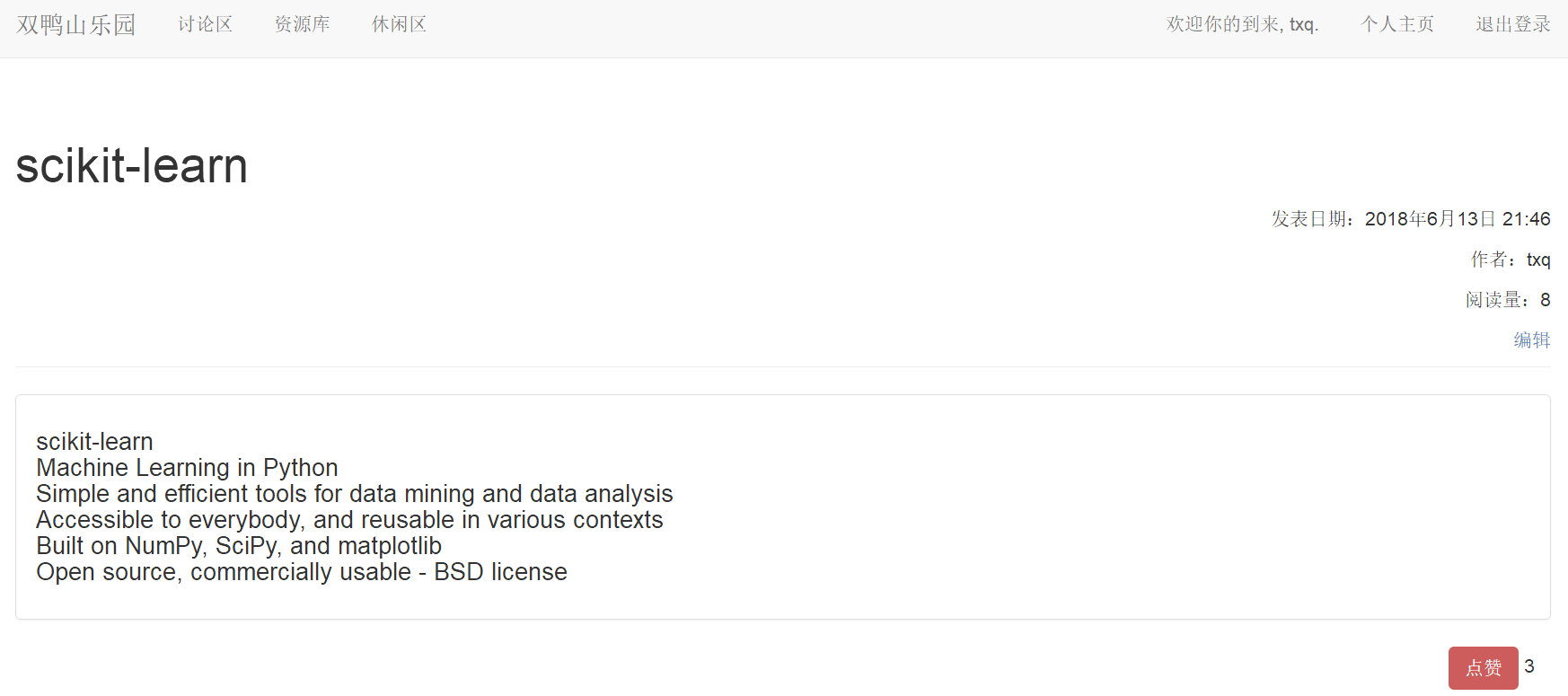
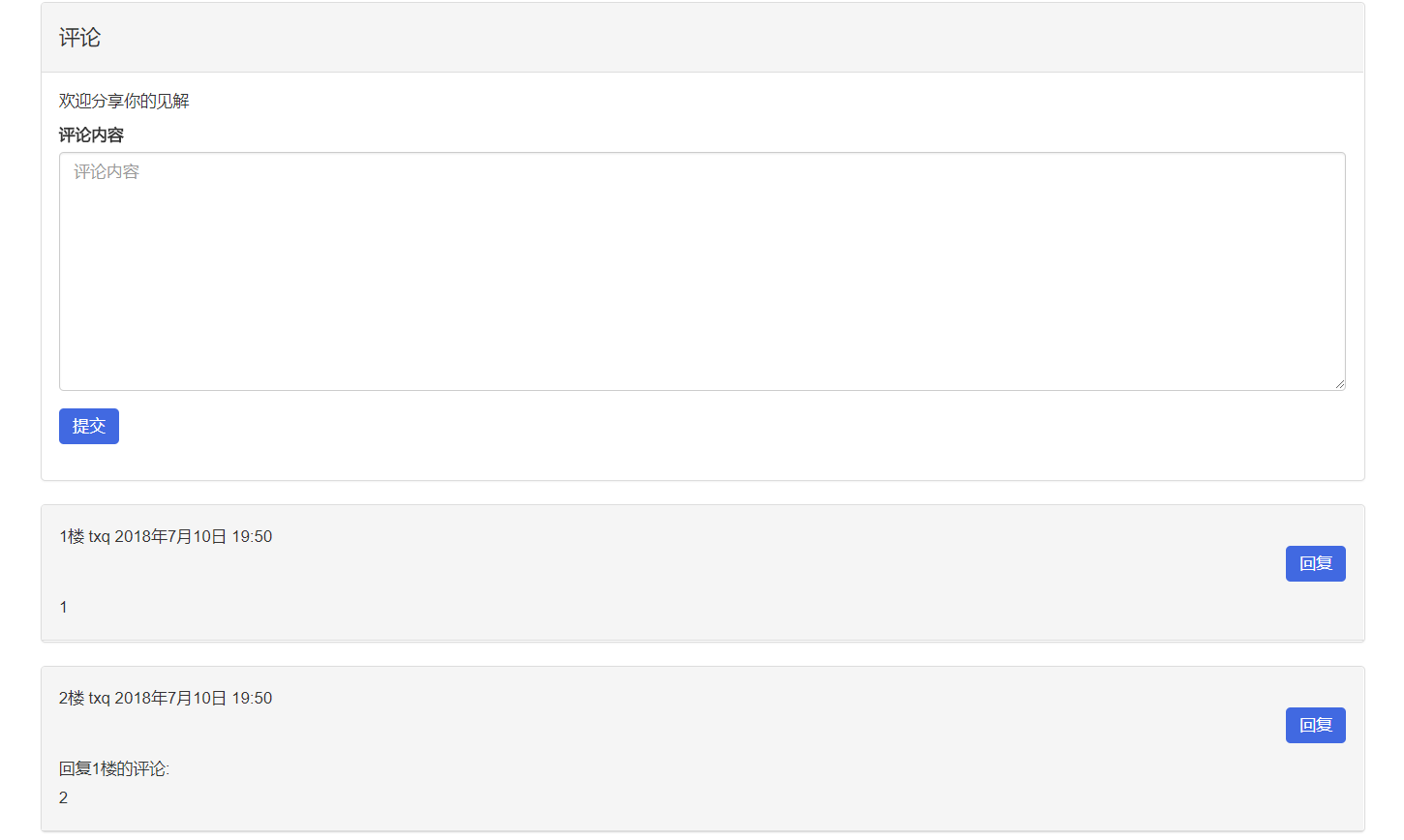
（2）某一课程下帖子列表：（例：python）



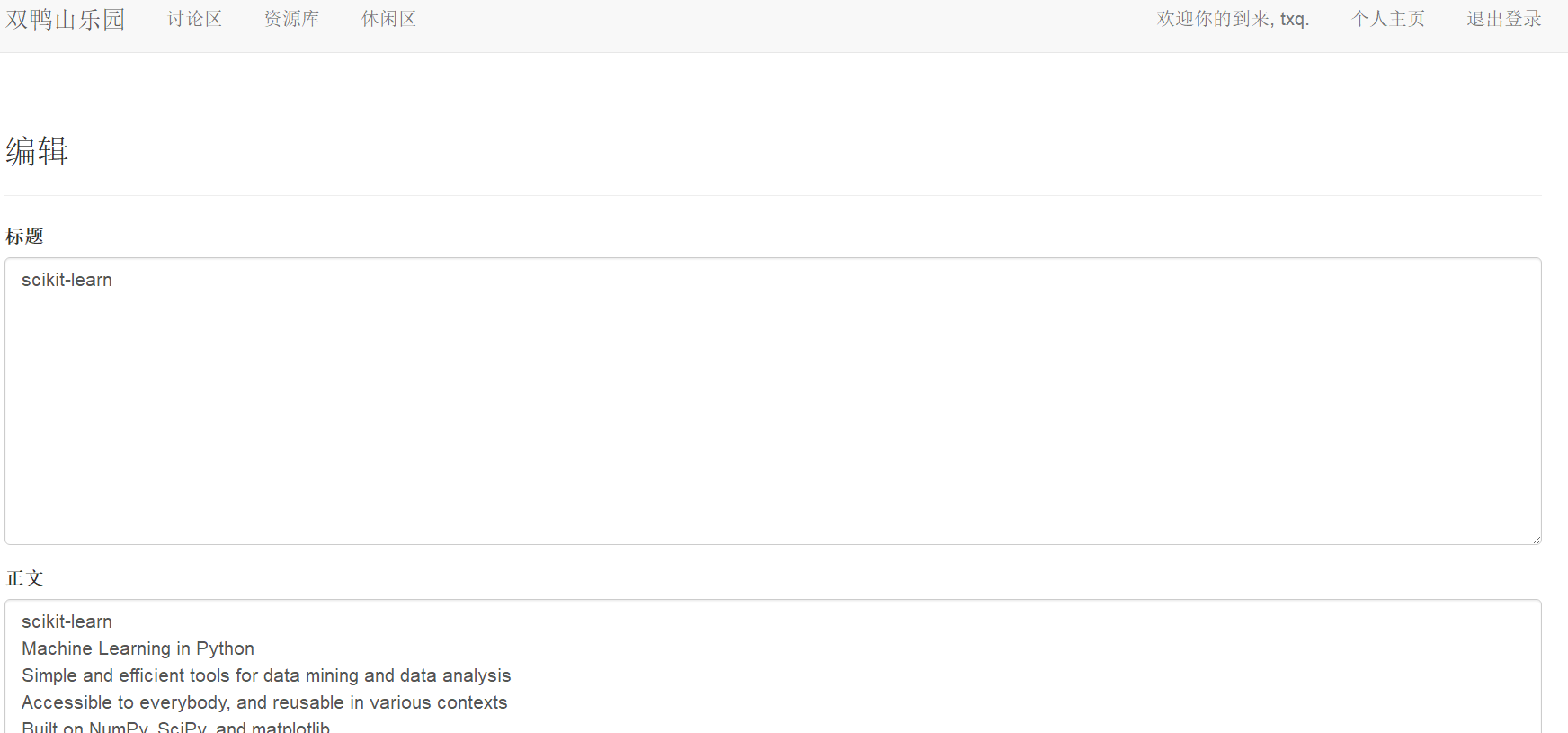
（3）创建新帖：



（4）帖子详细页面：

 （5）评论和回复

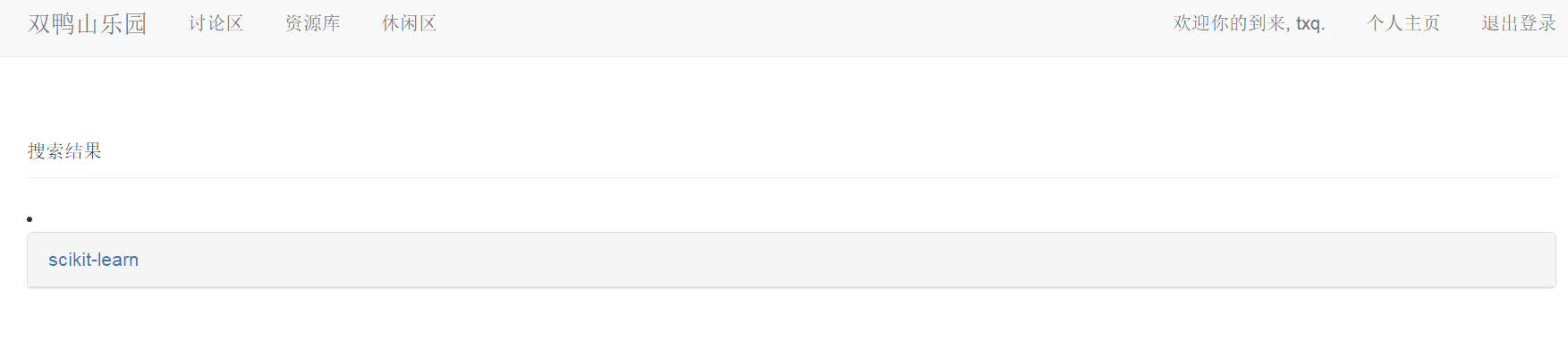
（6）编辑帖子：



（7）搜索帖子：

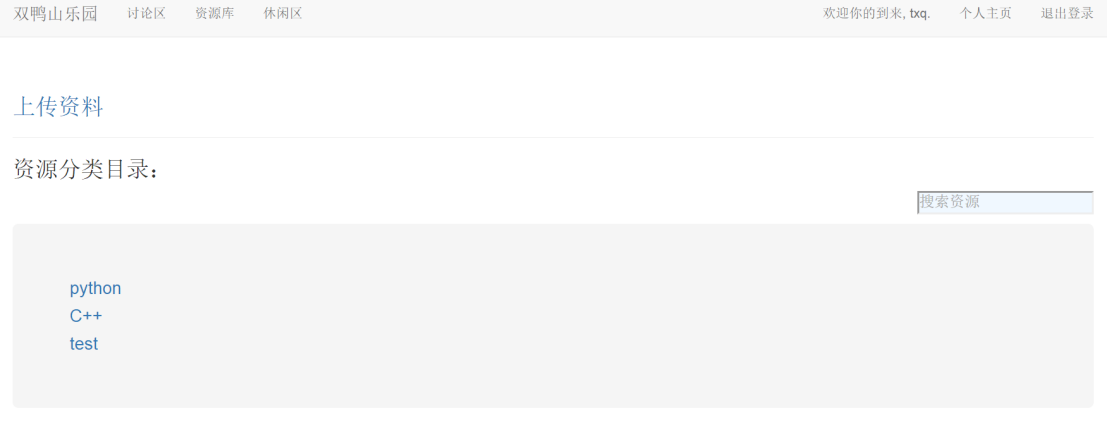


（7）搜索结果：

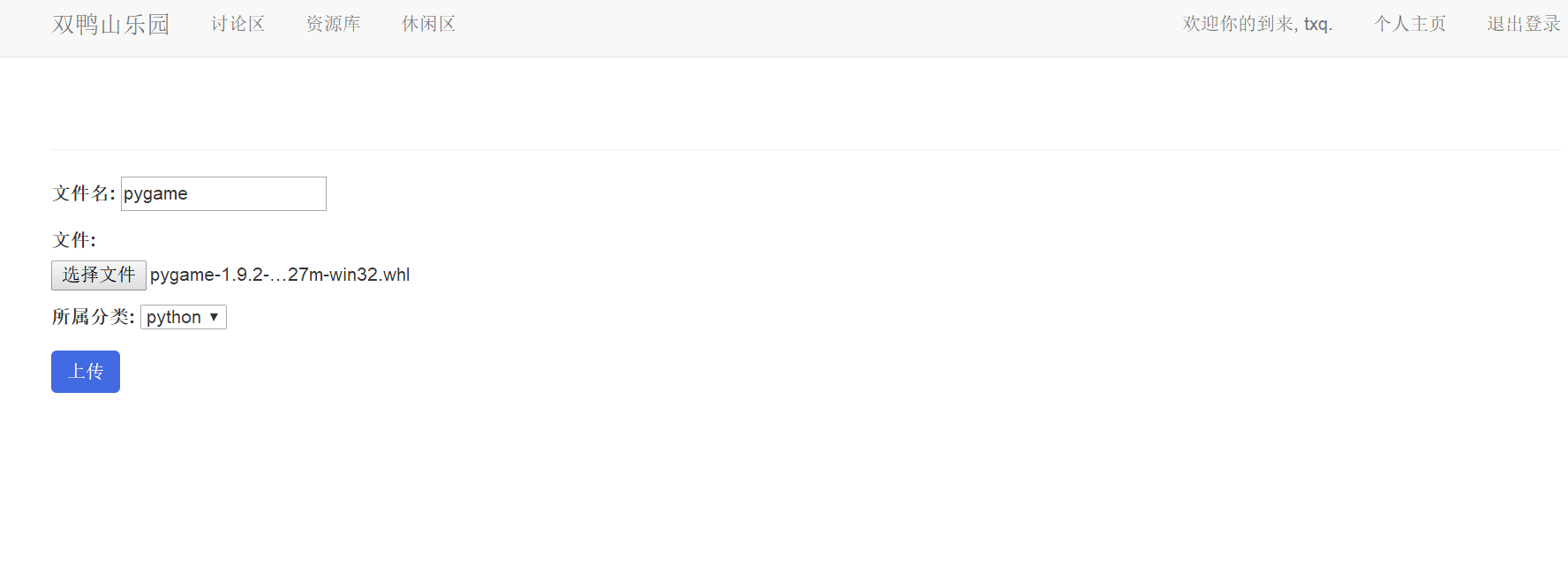


1. 资源区

（1）资源区总页面

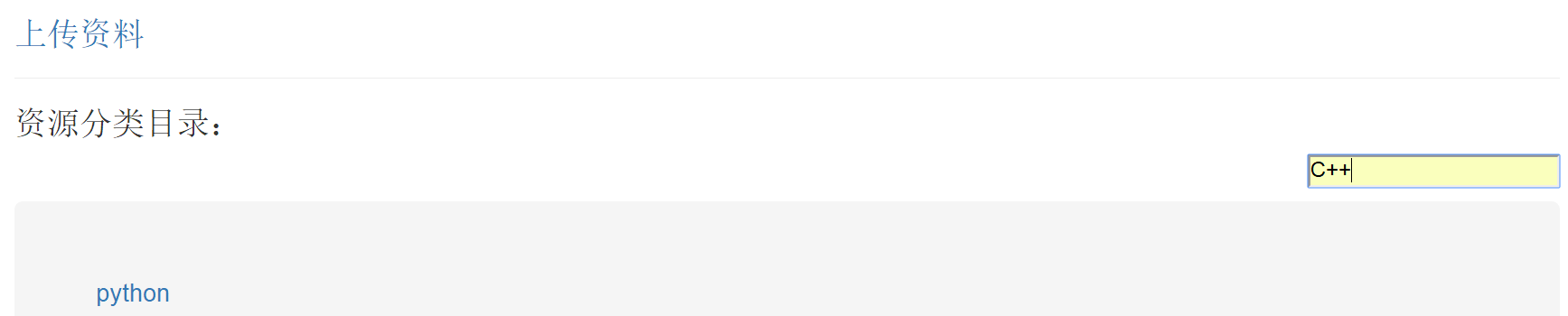


（2）上传资料：（选择上传文件名为：pygame，分类：python）

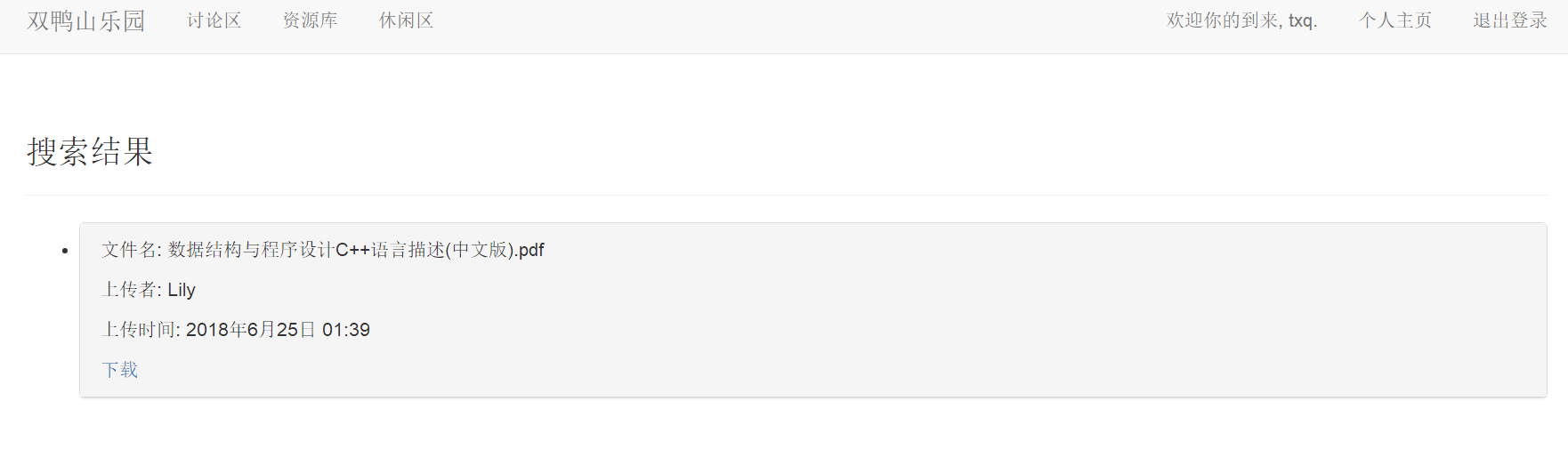


（3）某一分类资源列表：

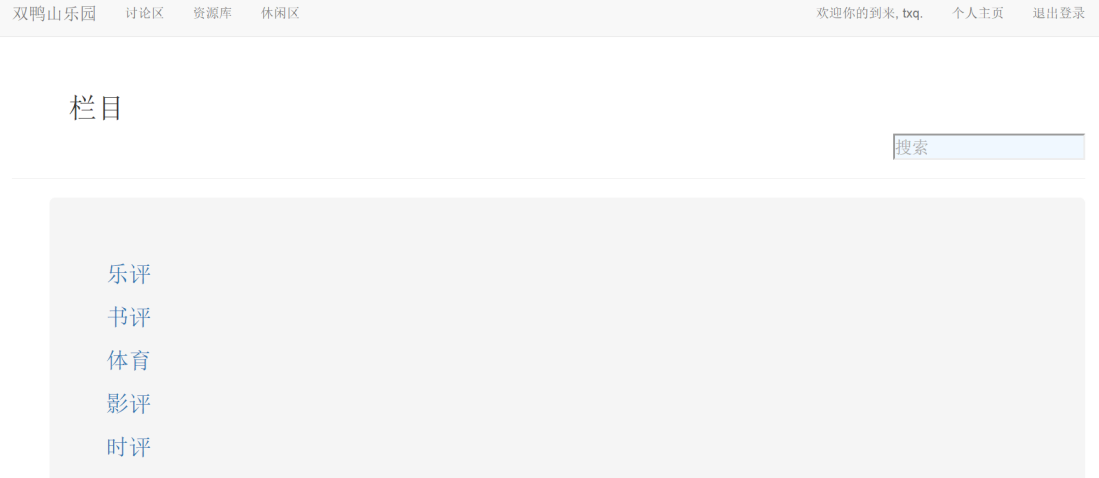


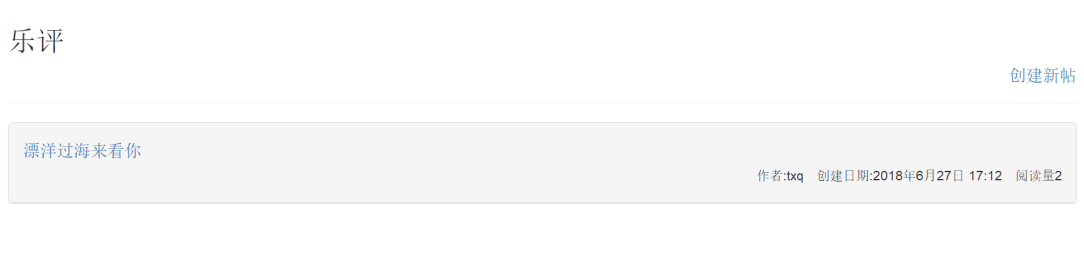
（4）搜索资源：

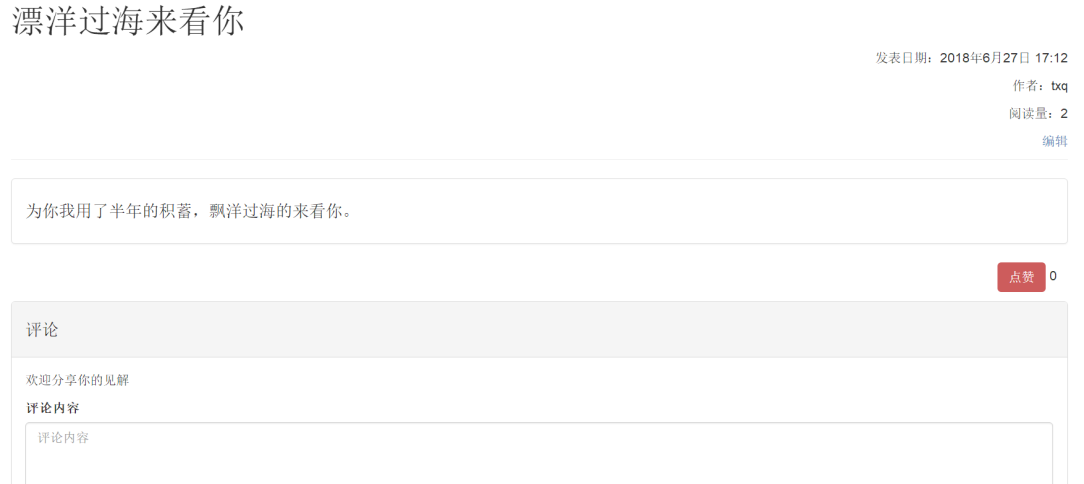
（5）搜索结果：



4、休闲区（休闲区功能与讨论区相似，部分展示如下：）

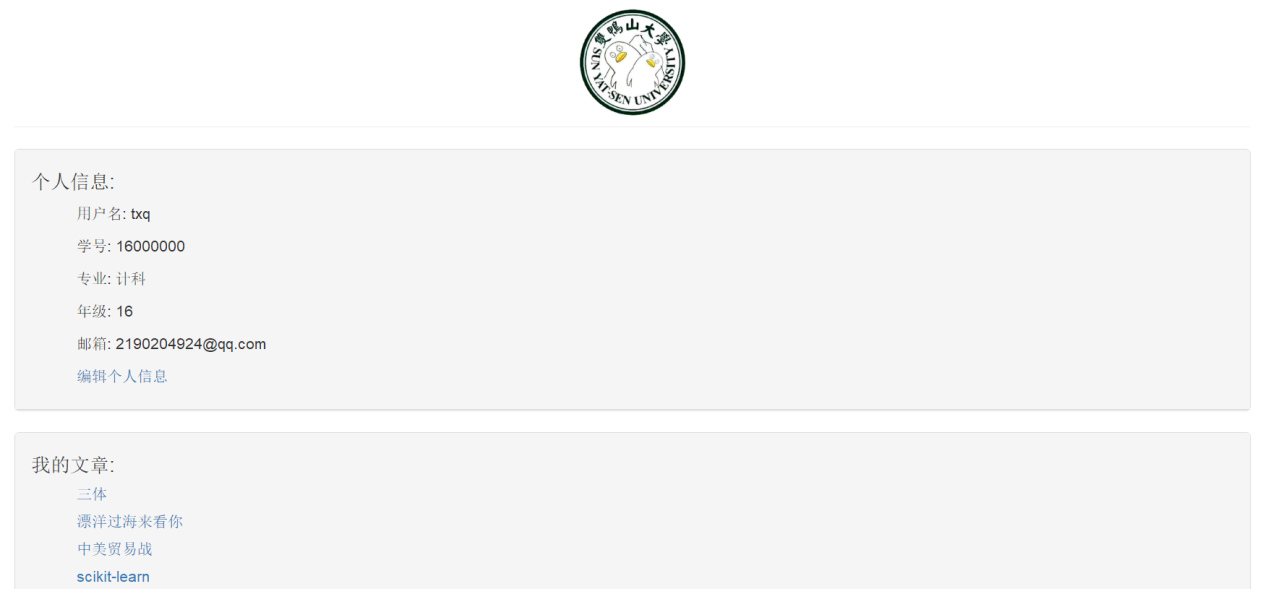






1. 个人主页

（1）主页详情



（2）编辑个人信息：

