****

课程设计报告书

**题目：学习笔记的web程序**

**学 院 计算机科学与工程学院**

**专 业 计算机全英联合班**

**学生姓名 谭演锋**

**学生学号 202130100456**

**指导教师 张幸林**

**课程编号 045102812**

**课程学分 2**

**起始日期 2022.8.30-2022.10.18**

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

**学习笔记**

**一、选题背景**

我们学习生活中偶尔会因为纸质学习笔记的难以统一和不方便查看而困扰，而后又存在个人的学习笔记，但是没有与他人交流的空间和平台。所以，在学习Django和python的使用后我萌生了制造一个完善的学习网站的想法。我借助Django系统和sqlite3数据库做出的一个收集自己学习笔记并可以统筹管理的一个网站。在网站能够收集整理自己的学习笔记，插入课件截图等多样性的表达形式，并可以选择公开或者私有，同时，进行编辑和修改也较为方便。区别于市面上的普通的网站，提供了公开交流的可能。

**二、方案论证(设计理念)**

设计原理：借助Django系统和sqlite3数据库，通过python的编写，实现的一个收集自己学习笔记并可以统筹管理的一个网站。

要实现的功能：“学习笔记”程序主要实现用户的注册和登录功能，在网站中用文字（可以加粗、斜体高亮等）和图片均可记录和编辑所需要记录的笔记，每个笔记具有一个标题。每个用户可以根据需要是否公开自己的笔记给其他人看、修改笔记、删除笔记，查找笔记。

方案选择：Django是一种免费、开源、高级的Web框架。Django具有齐全的功能。自带大量常用工具和框架，可轻松、迅速开发出一个功能齐全的Web应用。并且其具有完善的帮助文档和良好的社区环境，十分适合初学者实现一个具有完善功能的Web应用。并且Django自带强大的数据库访问组件。可以与sqlite3和python语言完美适配，不需要使用者掌握复杂的数据库操作。总而言之，Django是使用python语言开发Web程序的不二选择。该方案兼具可实操性和完善性。

系统的安全性和数据的完整性：Django框架有自带的保护装饰器用于保护用户的数据，也有防攻击手段。数据存于数据库中，不是超级用户无法对全部数据修改和删除十分的安全。

应用的运行环境及其性能要求：python版本3.10.6，Django版本4.1.3。其他库具体版本需求见压缩包中requirement.txt文档。该程序在创建时已自带虚拟环境，只需要在pycharm中改变编译器即可实现编译，具体操作见压缩包操作指引及视频。该程序对电脑性能要求极低。

**三、过程论述**

**1.用户的注册：**

**a)用户模型：**

users.models.py 建造自定义用户模型，以便实现自定义的用户表单和登录表单，以及后续等用户的一系列升级改造。继承django的库contrib.auth

自带的models的Abstractuser类自定义users用户模型Abstractuser利用这个类带的用户名，密码等数据成员，并且只带了两个新的数据成员。

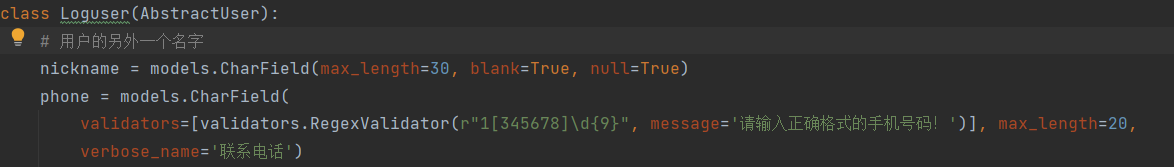


图1 用户模型

nickname：文本、最大长度30、允许空字符

phone：文本、最大长度20个字符，不允许空字符有限定的手机号码格式

**b) 用户注册表单：**

users.forms.py创建自定义用户模型对应的user表单用于注册账户，表单主要由三个空分别是username、password1、password2,phone分别对应 用户名、密码、确认密码、输入手机号码。每一个表单都有其错误信息代码，以告诉用于要按照要求注册用户。如username：



图2 用户表单

规定了每一个要求对应的错误信息，以提示用户输入正确的用户名。并且设置了表单的用户窗口。

**c) 用户注册视图函数及其模版（用于用户注册界面的显示）：**

users.views.py用于渲染登录界面，视图函数会首先验证注册的提交表单是否正确（确认提交格式）。如果注册账户符合要求那么就传输数据到登录界面进行等于，最后重定向到初始index界面。

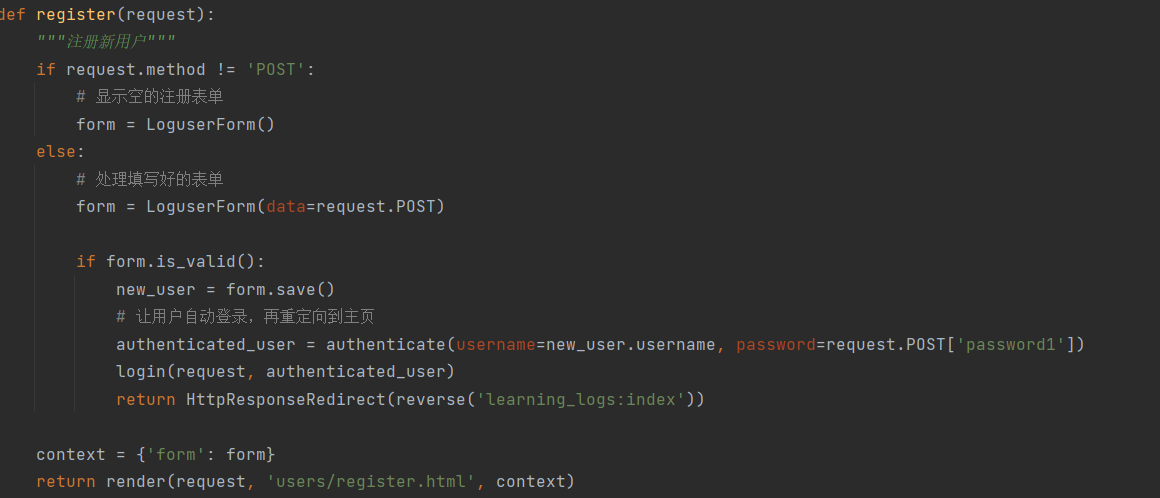


图3 注册函数代码

login是登录函数，利用用户注册数据，只要注册了就立即登录，然后return 重定向到index界面，让用户操作。最后两行表示：如果用户操作不当，数据填写错误，会把保存用户填写的表单，重新渲染注册的网页，让用户重新填写个人信息。

通过html和css制造出相应的注册网页如图：



图4 用户注册网页演示

如果用户的注册不满足格式要求，会显示报错，注册成功就会自动登录。

**2.用户的登录与登出：**

用户的登录与登出代码和思路与用户的注册大致相同，只需要调用django.contrib.auth库中的login和logout函数就能实现账号的登录与登出，并不需要额外的操作。在这里只展示用户的登录界面：

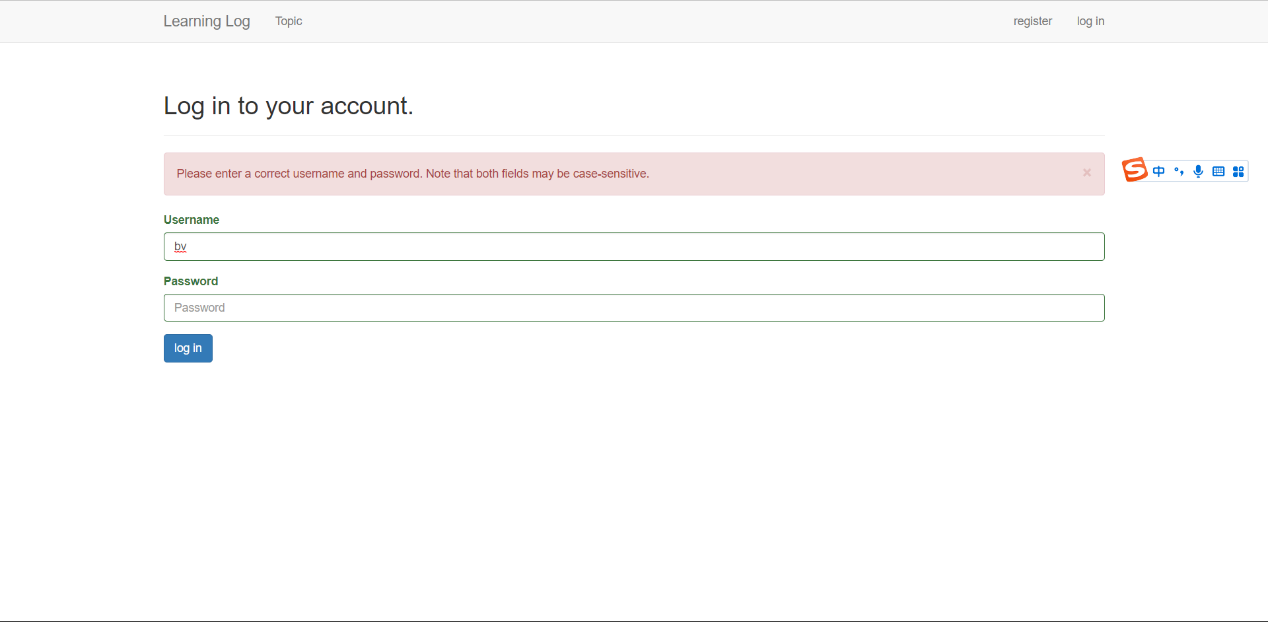


图5 用户登录网页演示

如果登录格式，或者账号密码不正确会自动报错。



成功登录后，界面的右上角会显示用户名，并且用户名右边有登出选项。



登出之后，界面的右上角会显示注册账号和登录的选项。

**3.学习笔记的Topic**

Topic是每个笔记entry（条目）的主题，用于管理不同主题的笔记项目

**a)Topic模型**

位于learning\_logs.models.py，包含topic名，topic添加的日期。topic所属的用户以及给予用户选择是否将这个topic下的笔记公开。

****

图6 Topic模型

text：标题名称限制了最大长度50

date\_added：程序自动添加的每个Topic的创建日期，用于排序

owner：创建该Topic的用户，且启用了级联删除，当用户被删除的时候，该用户名下的topic都会被取消

publicornot：一个布尔值，用于规定该Topic是公开的还是私人的

**b)Topic表单**

Learning\_logs.forms.py创建了Topic的表单TopicForm用以Topic的创建和修改操作。Form内定义了text和publicornot两个空,需要用户创建和修改表单的时候进行填写。

****

图7 Topic表单

text是Topic的文本，publicornot用于选择该标题是否公开

**c)Topic的视图函数及对应模版（用于对Topic增删改查以及显示）**

learning\_logs.views.py里定义了多个针对Topic函数用以实现topic的不同功能。

1. topic的显示：

利用python对数据库进行操作，查找出公开的Topic还有独属于用户的Topic，如果没有用户登录，则即私有数据不显示。

topics函数：

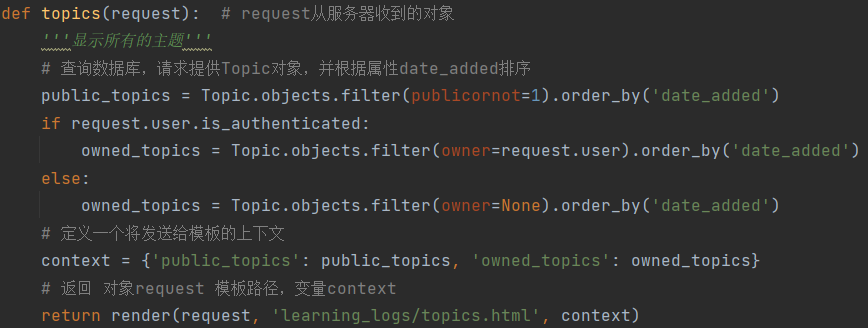


图8 Topic显示视图函数

查询数据库，首先查询出所有public的Topic数据。然后利用函数request.user.is\_authenticated判断用户是否登录。如果登录查找该用户对应的Topic数据。最后制成表格传到模版topics.html中进行渲染。

topics.html模版：

模版会区分public和own的话题进行显示。如果该Topic属于当前网页登录的用户或者可以进行删除操作，如果不属于当前用户只能进行修改操作。

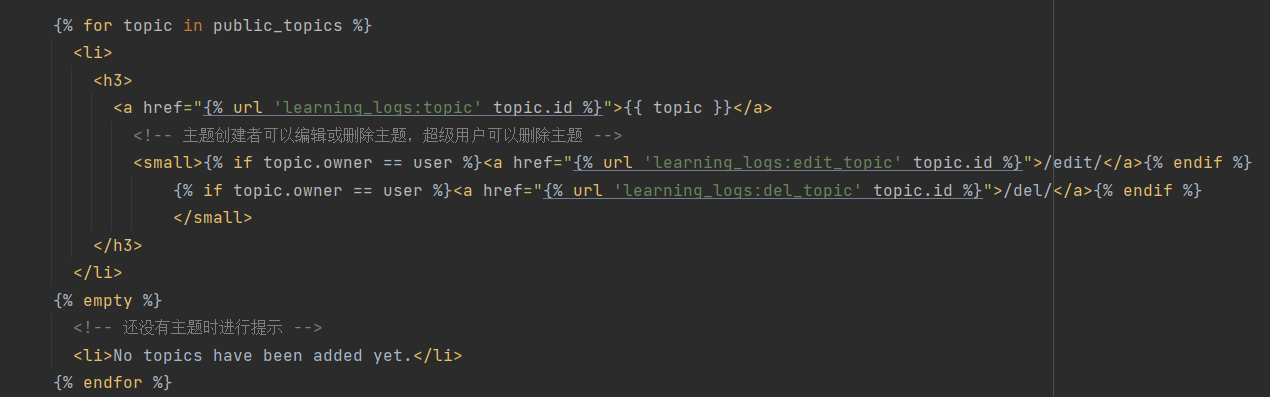


图9 Topic网页显示html模版

topics函数传入的数据会被用于topics.html显示每个topics。每个Topic都是一个连接，点击它就可以连接到对应topic名下的Entry（笔记）。并且topics.html会利用if语句判断该数据是否是的owner是否是当前网页的user，如果是就显示删除和修改的当前topic的操作选项。

如果当前没有topic，还会有文本显示提示用户要添加topic。

Public Topic：

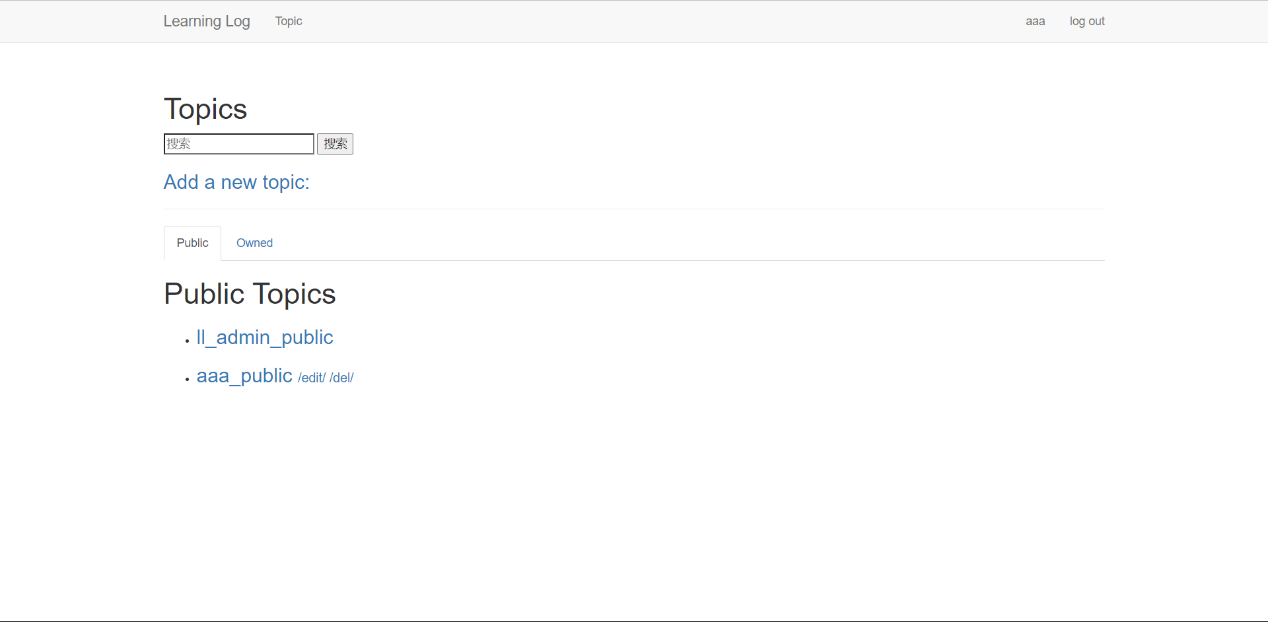


图10 Topic public网页显示

ll\_admin\_public这个topic不属于当前用户aaa的所以只能被看到，但没有修改还有删除操作的选项。aaa\_public这个topic虽然是public的但因为它属于aaa这个用户所以可以被修改和删除

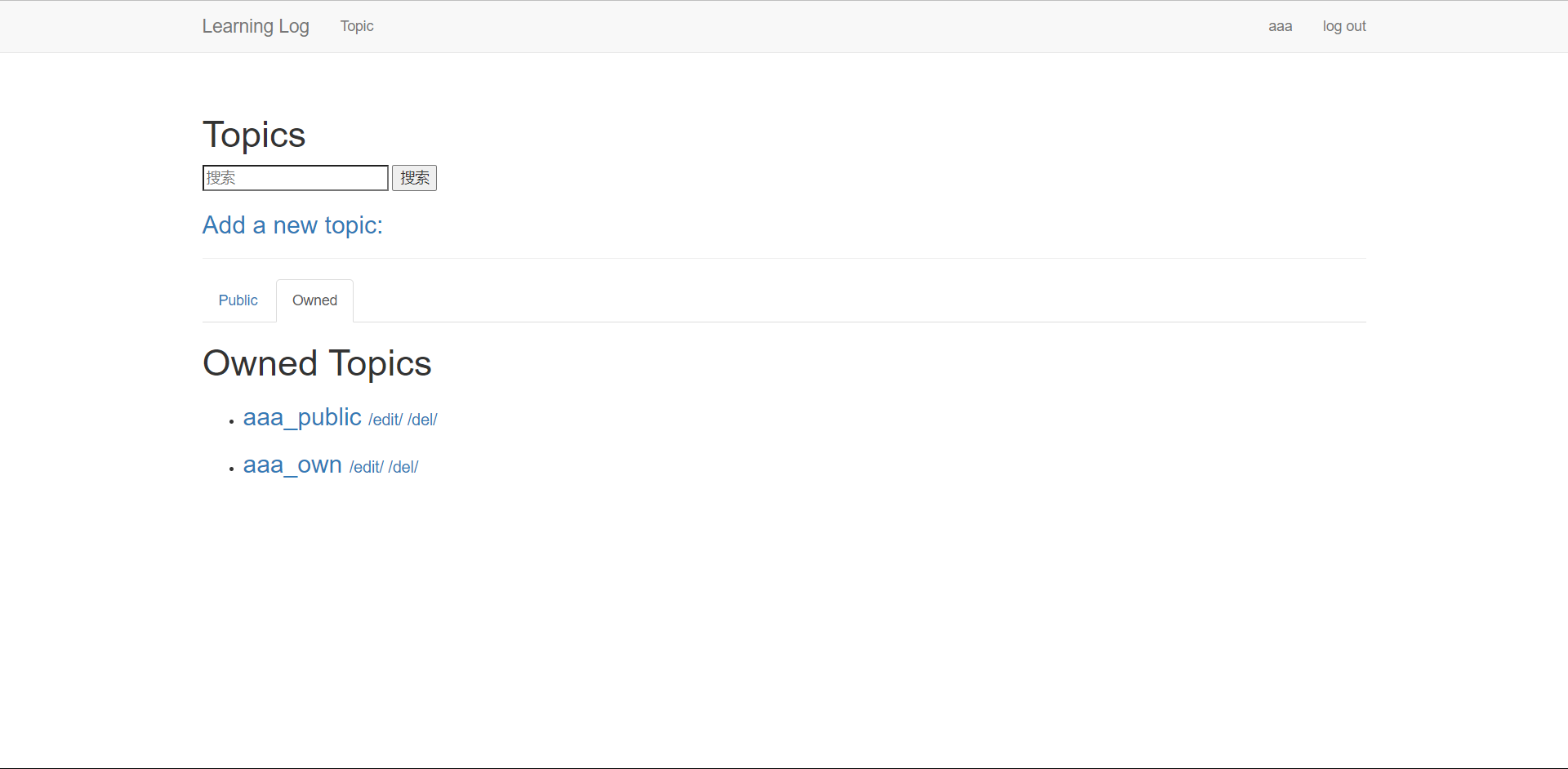


图11 Topic owned网页显示

Owned的topic因为都是属于当前用户的所以全部支持修改和操作。

（2）Topic的添加

new\_topic函数：

用于添加topic，利用TopicForm表单实现了Topic的新增操作。

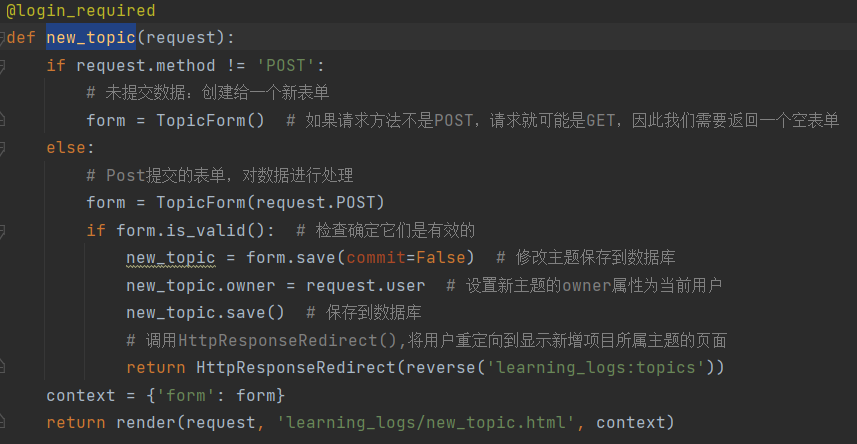


图12 新增Topic视图函数

调用了装饰器@login\_required表明当前操作需要用户登录才能进行操作不然会报错。大致思路与用户注册相同：创建空表单用于新建Topic，并且率先检查Topic的合理性，当合理的时候存到数据库，不合理要求用户重新编写。在保存前首先要绑定topic的owner为当前的user（request.user）

new\_topic.html模版：

存在两个表单选项用于添加Topic一个是文本，另外一个是选择是否公开该Topic然后进行提交

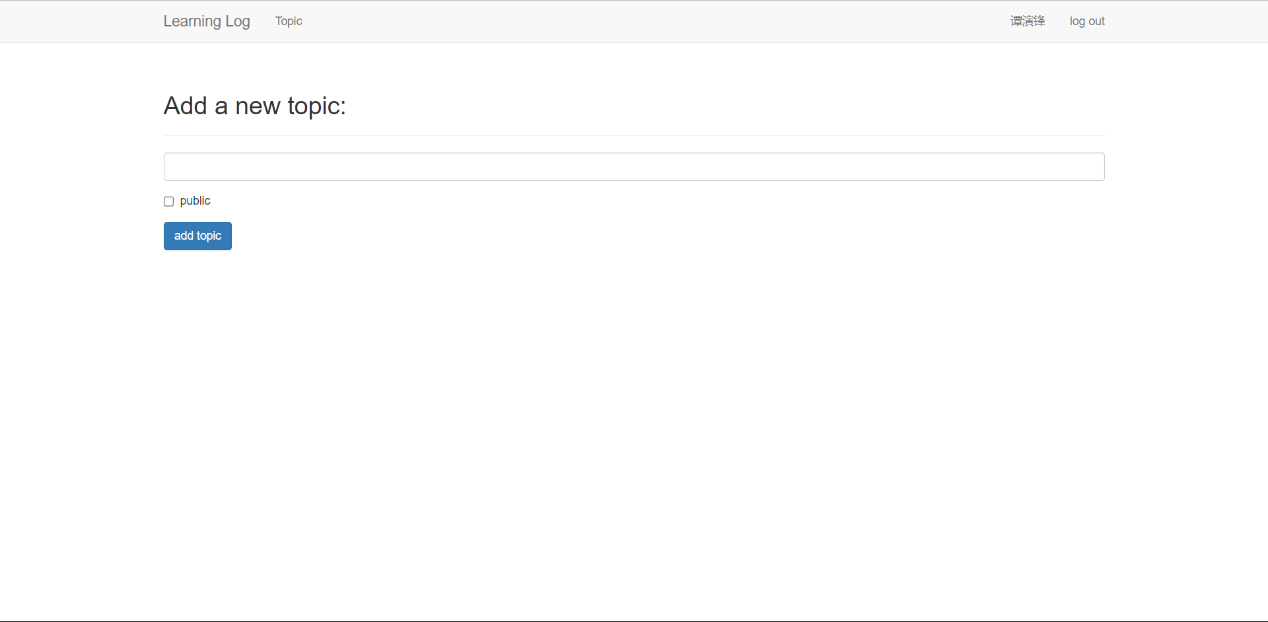


图13 新增Topic网页显示

（3）Topic的修改

Topic修改的视图函数与新增大致相同，不过你用了topic\_id提前读去了topic的内容，然后使用instance=topic 将已经存在的topic内容提前放到表单类，让用户进行修改。

****

图14 修改Topic视图函数

Topic的修改和Topic的添加大致同理，这里不作过多阐述，只是展示topic的修改界面

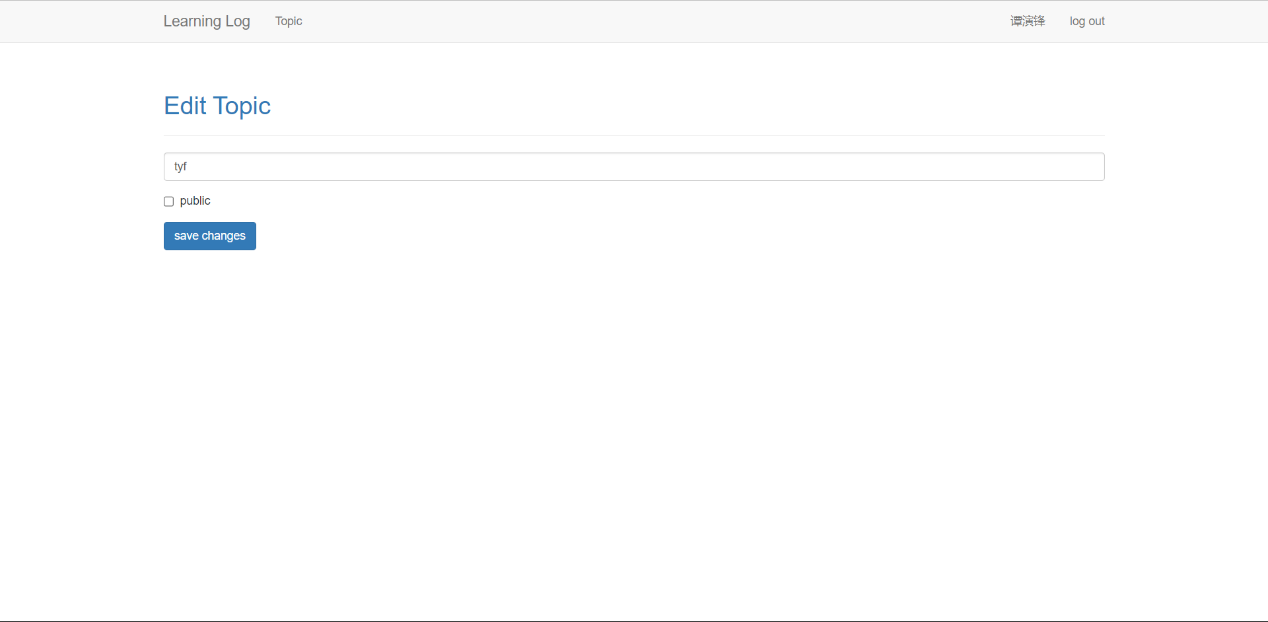


图15 修改Topic网页显示

界面功能也大致和new类似。

（4）Topic的删除

del\_topic函数：

用于删除操作。同理登录后才可进行操作，该操作会接受一个html网页发送的topic\_id参数用于对topic的删除。

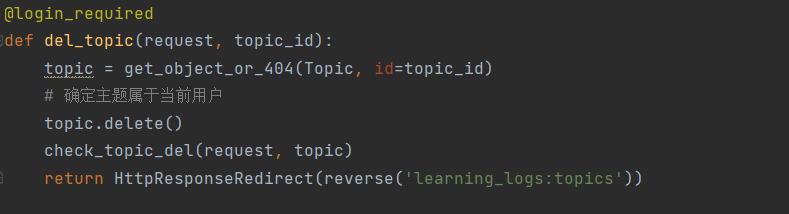


图16 删除Topic视图函数

get\_object\_or\_404(Topic,id=topic\_id)获取需要删除的topic。随后调用delete函数进行删除，并且检测该删除是否合法。最后重新回到topics.html网页。

（5）Topic的查找

该查找是关键词检索只要topic含有当前搜索的字符就会展展现出来。并且该查找还能根据标题的创建日期进行搜索。

search\_topic函数：

查找函数利用topics.html的搜索框提供的keyword对 进行检索。并且实现了多关键词检索。

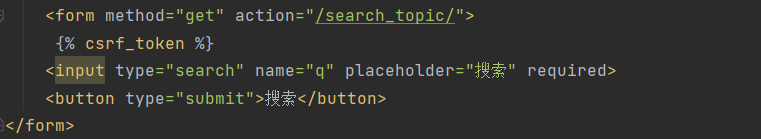


图17 搜索Topic用到的html的form语句

利用Form 在网页内构建了一个表单，使用传输方法是“get”传送表单内的数据到http://learng\_logs/serch\_topic/的网站内，input是输入框，用于搜集关键词数据，传送到视图函数search\_topic内部，帮助其筛选数据

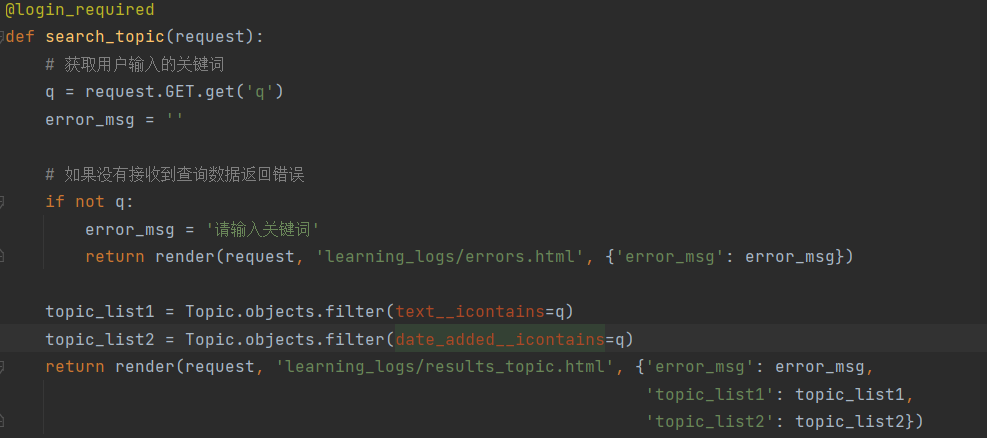


图18 搜索Topic视图函数

q是搜索框传入的keyword，如果没有keyword会有错误信息出现，如果有q则对数据进行filter操作找到目标数据，传入results\_topic.html模版中进行显示。

results\_topic.html模版：

模版构造大致与topics.html类似。



图19 搜索Topic后筛选数据的if语句

模版中的if条件语句用于判定，传入的Topic数据的拥有者是否和用户一致，或者是否公开，来进行判定是否将此Topic显示在搜索结果中。

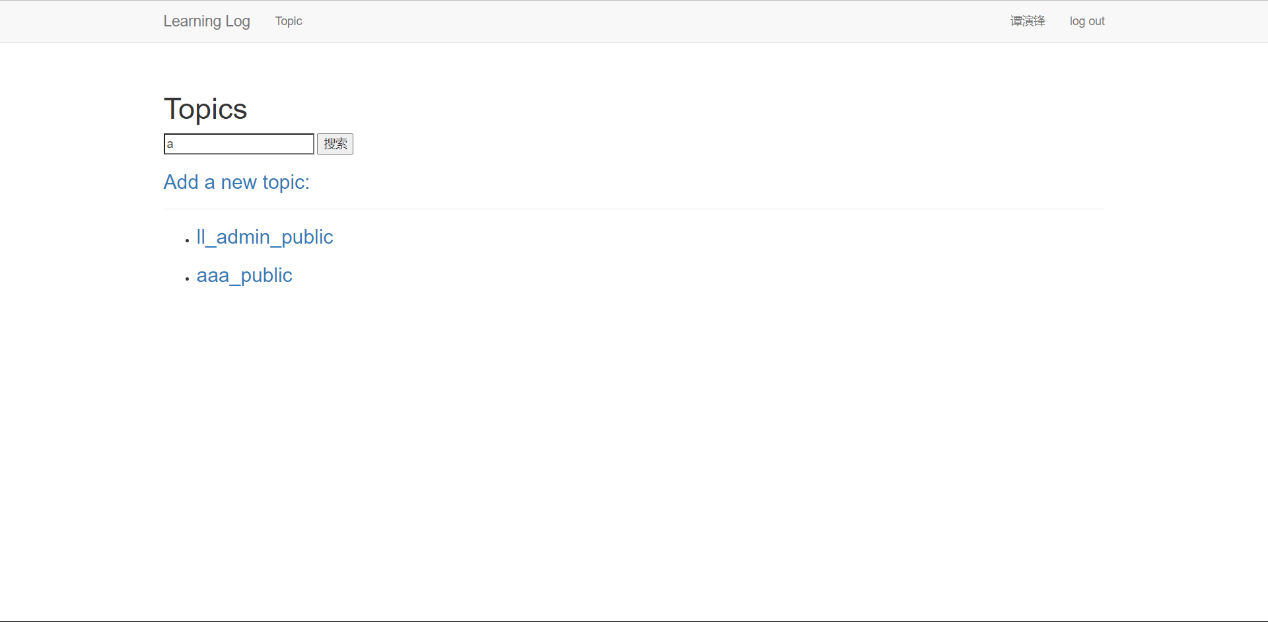


图20 利用关键词“a”搜索Topic网页演示

对a搜索就会显示出“谭演锋”这个账户可以看得到的Topic。

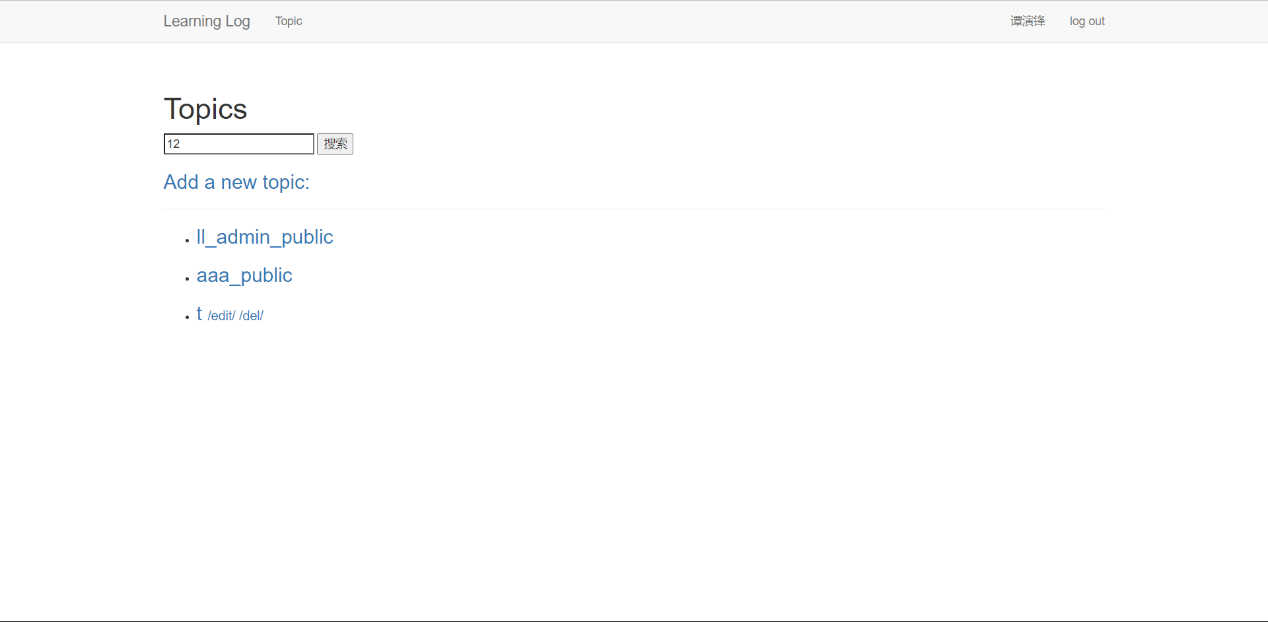


图21 利用关键词“12”搜索Topic网页演示

日期搜索12也可以看到12月份创建的Topic

**4.学习笔记的Entry**

**a)Entry模型**

* 位于learning\_logs.models.py，大致内容和Topic模型相似。但值得注意的是为了使用户的笔记能够突出重点，并且能够添加图片，超链接等功能。安装了pillow和ckeditor两个插件在learning\_log.settings.py文件中 指定富文本编辑器或其他上传文件的根目录：

MADIA\_ROOT=os.path.join(BASE\_DIR,'media')

MADIA\_URL='/media'

CKEDITOR\_UPLOAD\_PATH='upload/'

CKEDITOR\_IMAGE\_BACKEND='pillow'

在learning\_logs.models.py文件中import了RichTextUploadingField用以实现富文本编辑框。最终实现了如下效果：

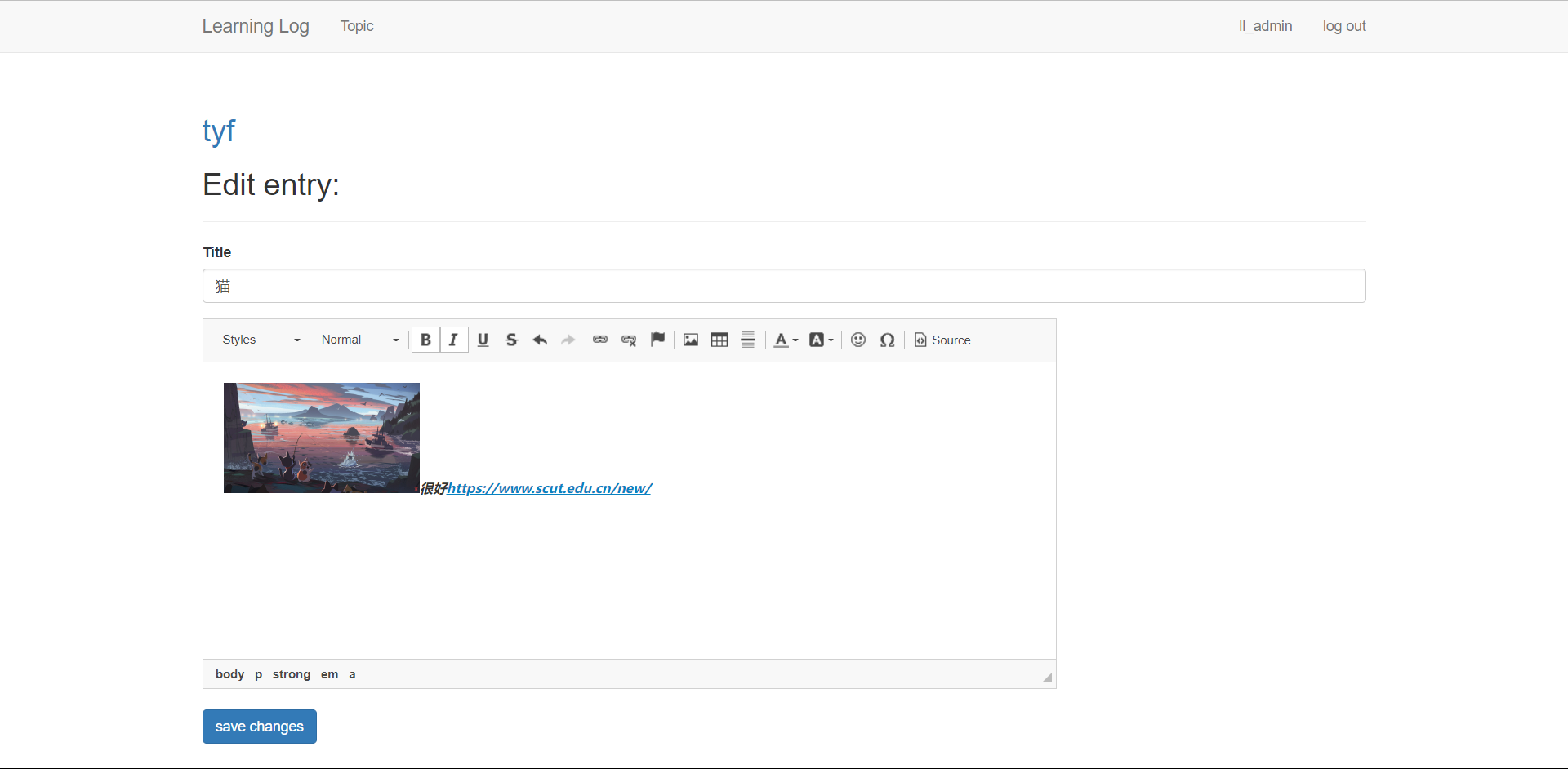


图22 Entry富文本框网页显示

用户可以在笔记文本编辑框类插入图片，超链接和斜体粗体的等文本。

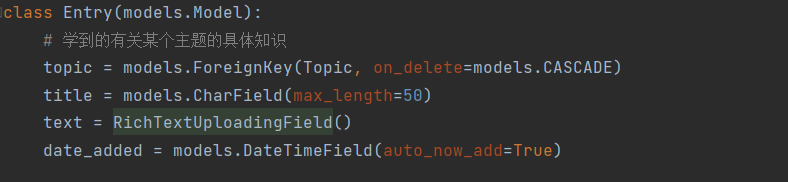


图23 Entry富文本框显示用到的模型代码

topic：使用ForeignKey将Entry和对应的Topic相连，并且使用了级联当Entry对应的Topic被消除时所有Entry也会消失。

title：是最大长度只有50的文本，是每一个Entry的标题

date\_added：程序自动添加的每个Entry的创建日期，用于排序

text：即Entry的内容。使用RichTextUploadingField创建富文本框用于显示内容。

**b)EntryForm表单**

大致与TopicForm相同，用于新增和修改Entry，定义了不同的输入框。

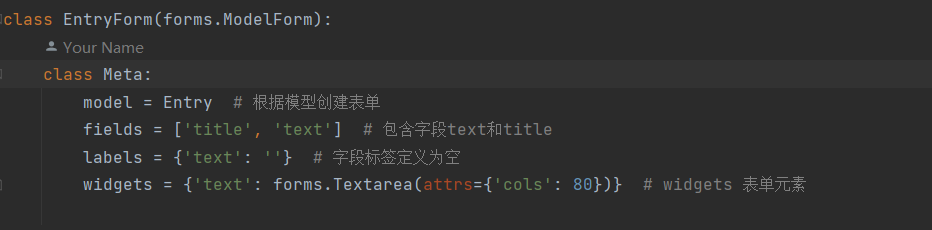


图24 EntryForm代码

widgets用于调整输入框大小使得文本输入框更大，便于对于笔记的编辑。

**c) Entry的视图函数及对应模版（用于对Topic增删改查以及显示）**

Entry的显示和增删查改操作与Topic类似，这里不作过多赘述。只讲一些区别显示出来，并显示每个Entry操作对应的网站。

1. Entry的显示：

Entry的视图函数：

因为每一个Entry对应一个Topic，所以其视图函数topic会获取Entry对应的topic\_id借此获取每一个topic的entry



图25 显示不同topic下的entry网页路径

entry的路径中命名需要用到额外的参数topic\_id，通过这个为每一个不同的topic以及其对应的entries构建显示的网站



图25 显示不同topic下的entry视图函数

Topic的内容可以工具topic\_id获取，然后根据Topic的内容，用entries = topic.entry\_set.order\_by('-date\_added')函数获取该topic对应的entires并且根据时间顺序排序，利用content字典传输到entry的模版中用以展示entry。

Entry的模版：

用于显示不同topic下的网页

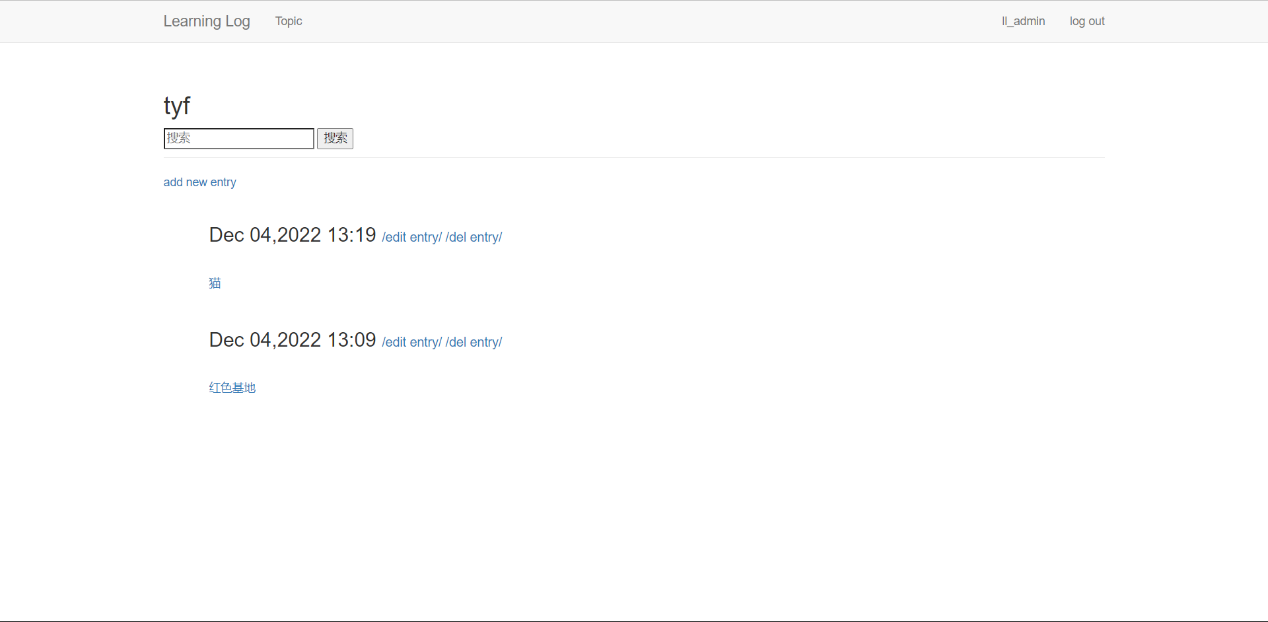


图26 不同topic下的entry显示网页

界面最上方是该Entry对应的Topic的名字，下方是搜索框可以根据日期，Title，或者Entry内容搜索相应的Entry。界面中会显示Entry对应的Title还有创建时间，可以通过点击Title查看Entry具体内容详情。

1. Entry的增加：

与Topic大致相同，只是TopicForm改为了EntryForm。通过learning\_logs.views.py 内的自定义new\_entry实现新增

Entry模版：

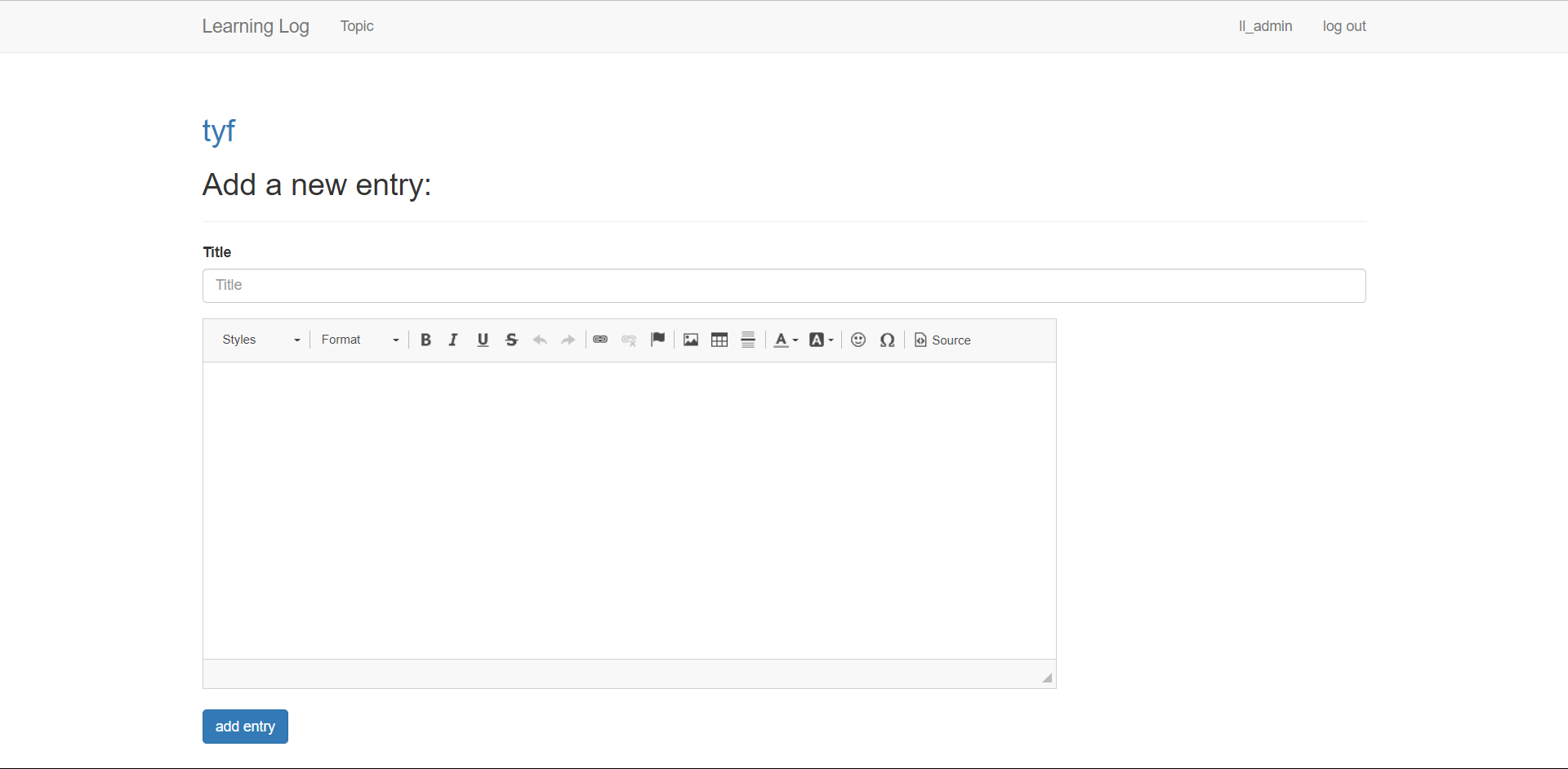


图26 添加entry网页显示

并无太多变化，值得有两个输入框一个会Title，另外一个是内容text,值得注意的是该内容text框是富文本输入框。

1. Entry的修改：

与Topic的修改大致相同，通过learning\_logs.views.py 内的自定义edit entry，进行修改只是将topic改为了entry，无太大区别。在修改网页上也是与新增Entry的网站的视图相同。

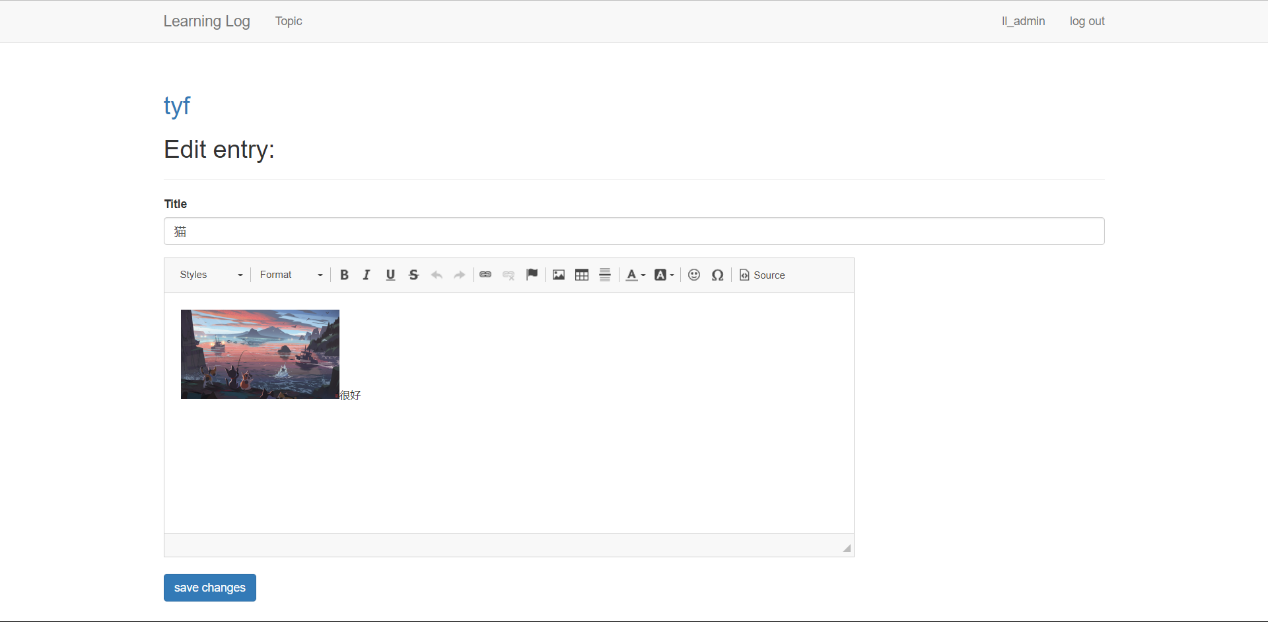


图27 修改entry网页显示

1. Entry的删除：

同理和Topic的删除大致相同，通过learning\_logs.views.py 内的自定义del\_entry函数实现了对entry的删除

1. Entry的查找：

通过learning\_logs.views.py内的自定义search\_entry函数实现了对entry的查找。与Topic查找相比，增加了一个根据Title关键词的查找。



图28 查找entry时使用的form语法

通过模版内的form进行{{topic.id}}的数据传递，实现只查找当前topic的entry内容。



图29 筛选查找到的entry时使用的if语法

在result\_entry.html模版内通过两个if语句，对于entry的内容进行筛选，如果不是当年网页内能看到的topic以及不是当前entry的topic的entry虽然依旧存在要查找的关键词但依旧无法查找。

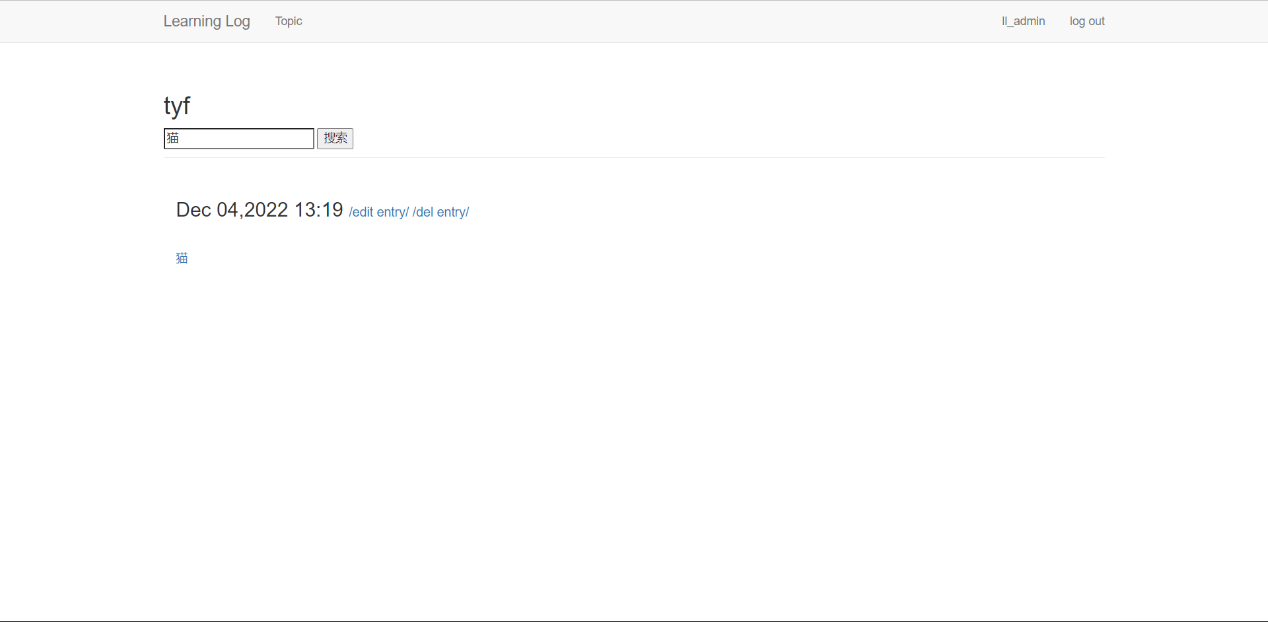


图30 查找entry的网页样式

**5.base模版以及界面美观设计**

使用base.html是其他所有网页的继承模版，利用继承大大简化了代码的工作，并且保持了网页风格的一致和美观



实现了每个网站上存在的的索引栏的设计

使用了bootstrap3和css，js等文件时间了界面上的美观设计



图31使用了bootstrap3，static文件实现美观

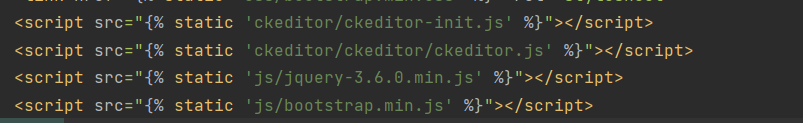


图32 Static静态文件内涵js和css的相关代码文件实现了吧界面变美观

响应式布局设计，当网页变小的时候，索引栏会自动缩小，显示一个白色点，供用户点击再显示索引栏

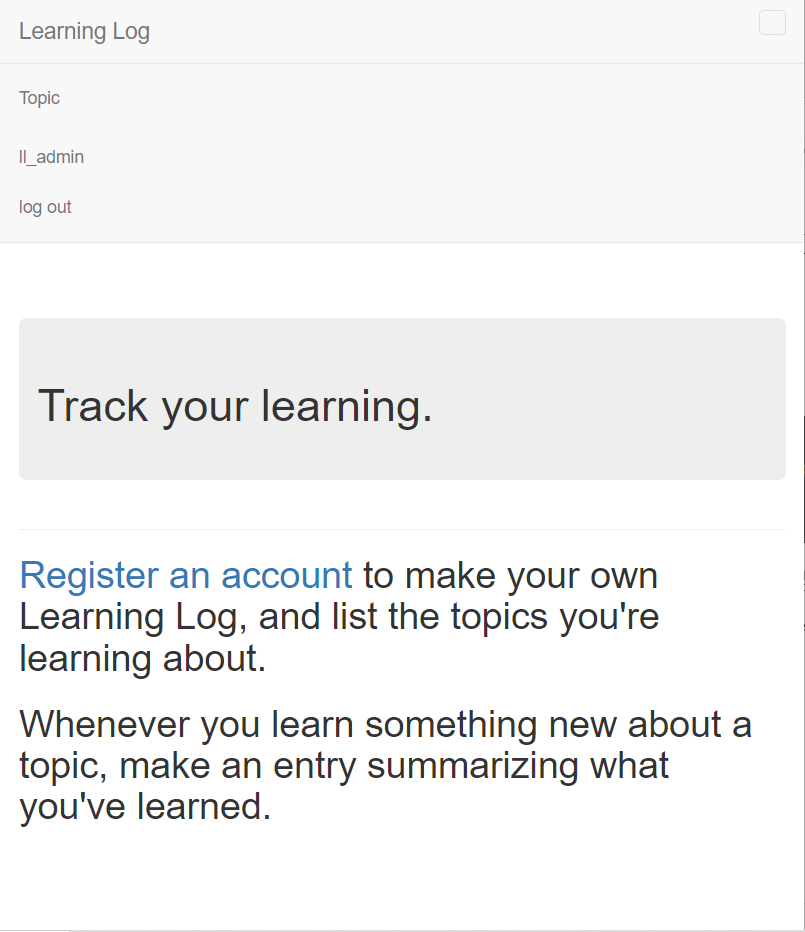
****

图32 使用响应式布局，网页缩小后的网站索引栏

**四、结果分析**

主要功能：主界面

左上角Learn\_log和Topic是导航 右上角显示当前登录的用户名，还有登出按钮。如果选择登出， 右上角会显示注册和登录按钮



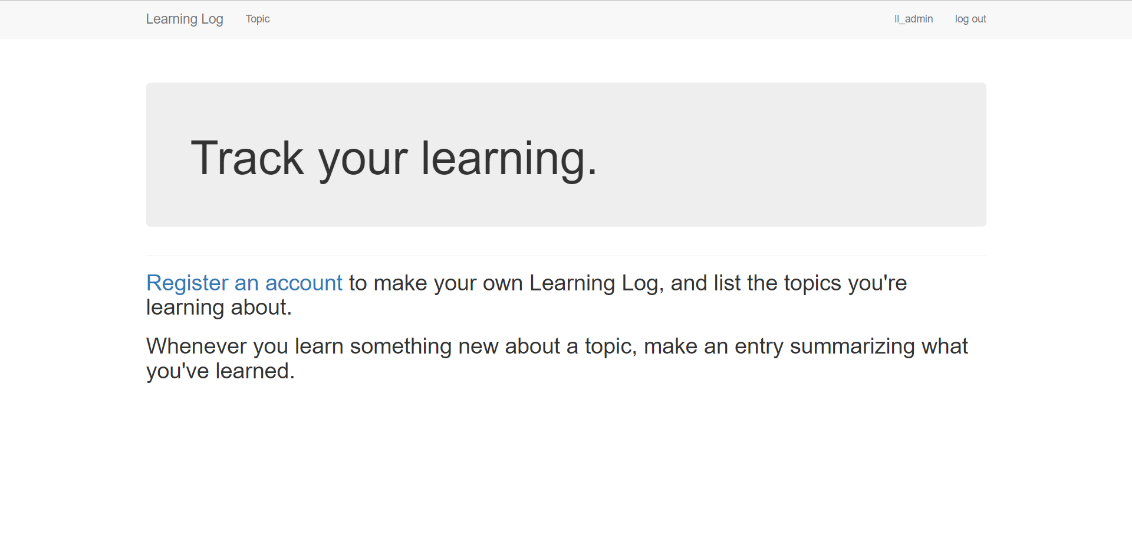


图32主界面

注册和登录界面，注册时，会要求填完所有基本资料，注册成功后会自动登录，登录时如果账号密码错误会报错。

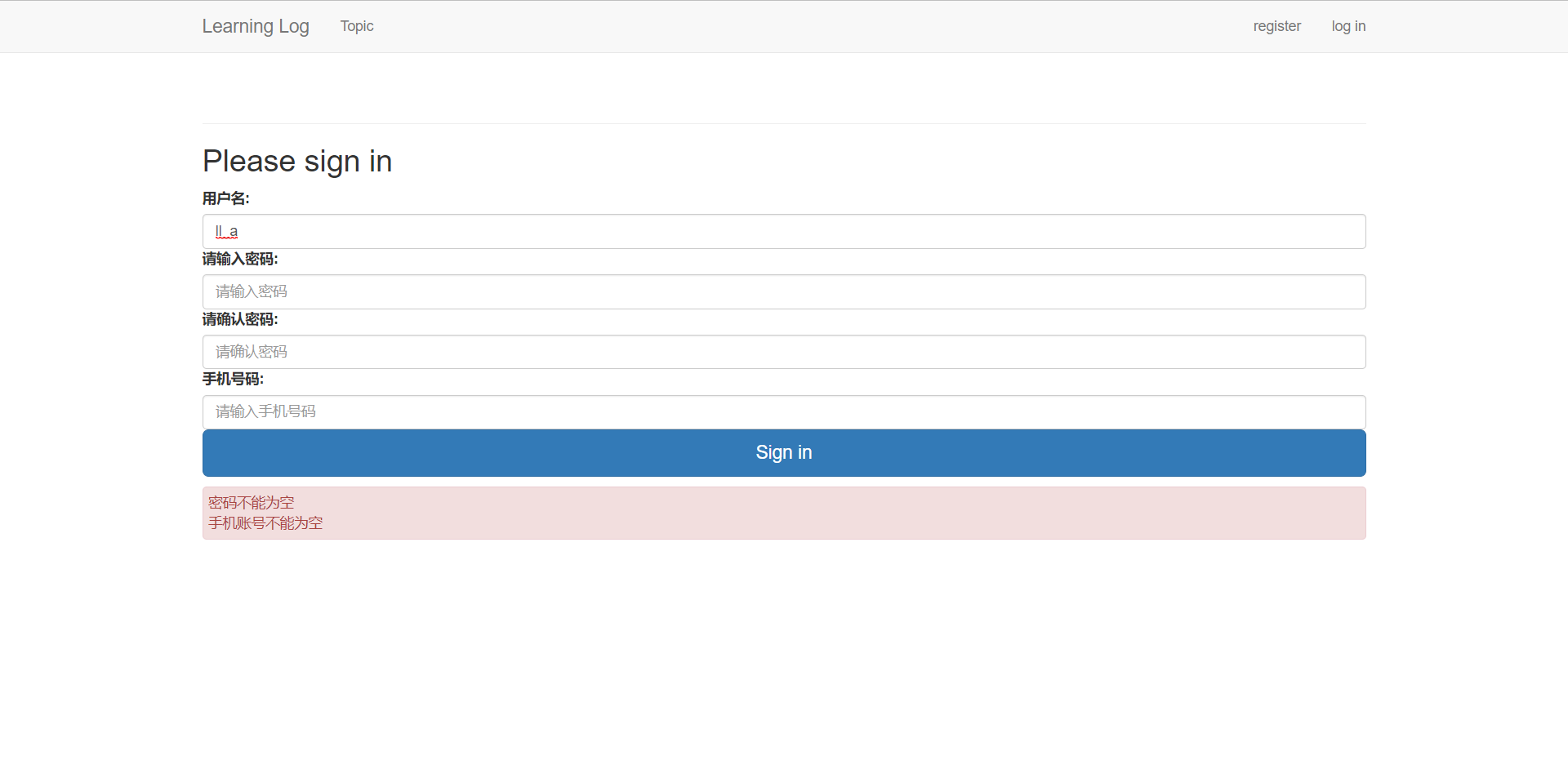


图33注册界面

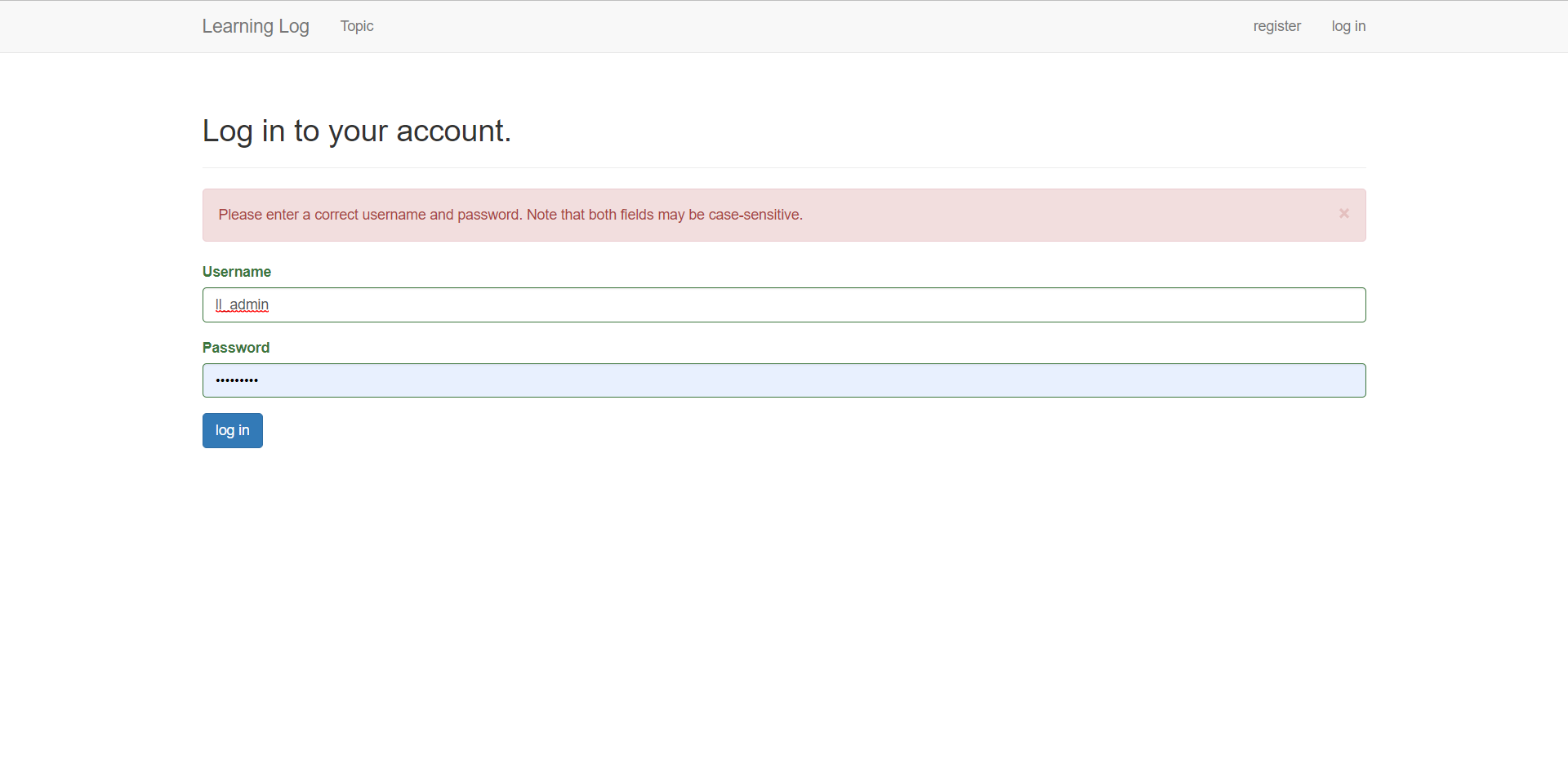


图34登录界面

点击topic，进入topic界面，topic界面会显示public（所有用户可见的笔记）或者owned（仅自己可见的笔记） topic。如果没有登录，只能看到公开的笔记。点击具体的topic进去就可以看的到topic下的笔记。Topic可以进行修改和删除，只有属于自己的topic可以进行修改和删除。还可以新增笔记，对笔记进行查找（根据日期，或者名称进行查找）。修改和新增笔记的时候，可以选择该笔记是否公开等。

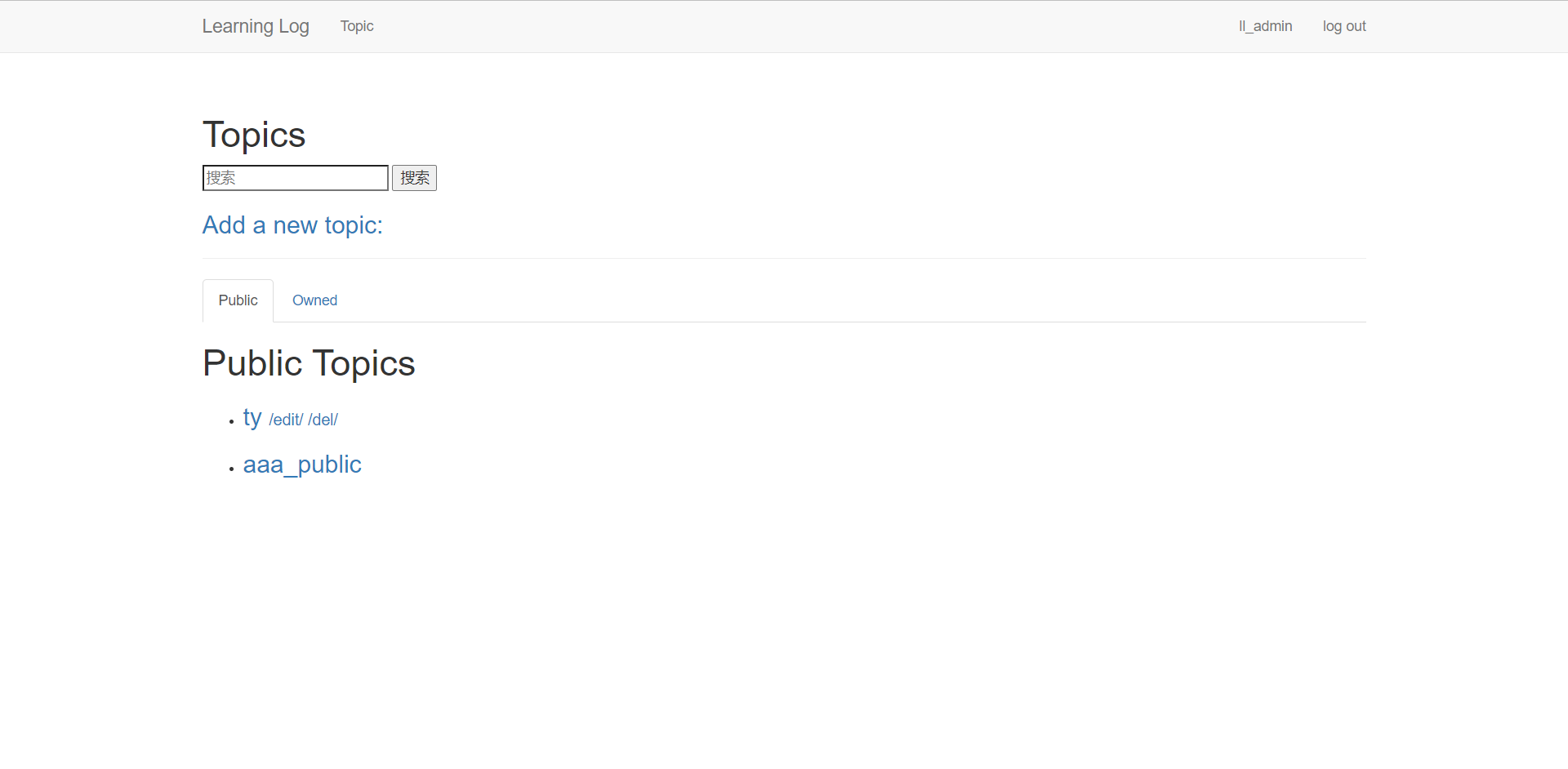
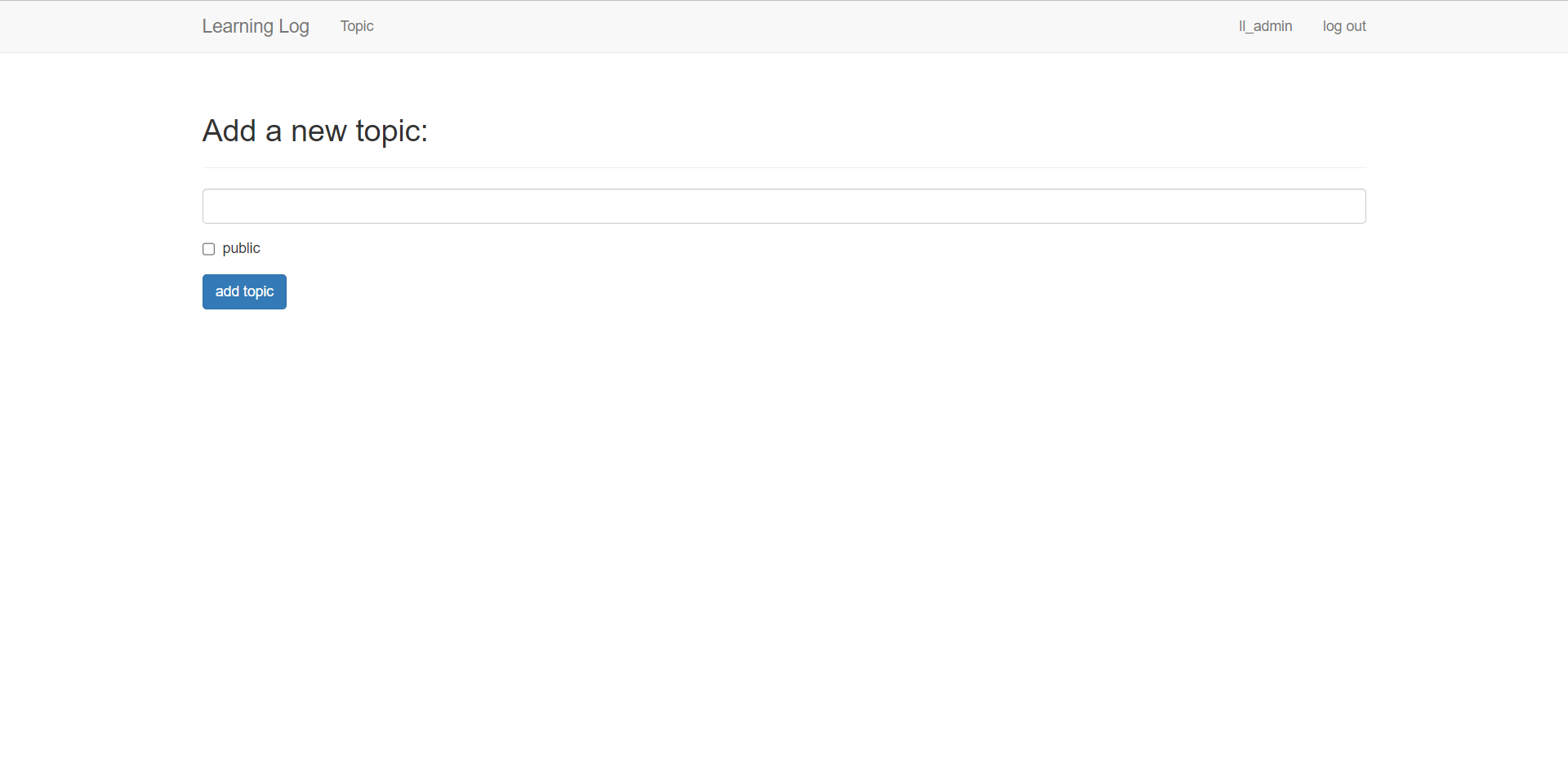


图35：topic主界面



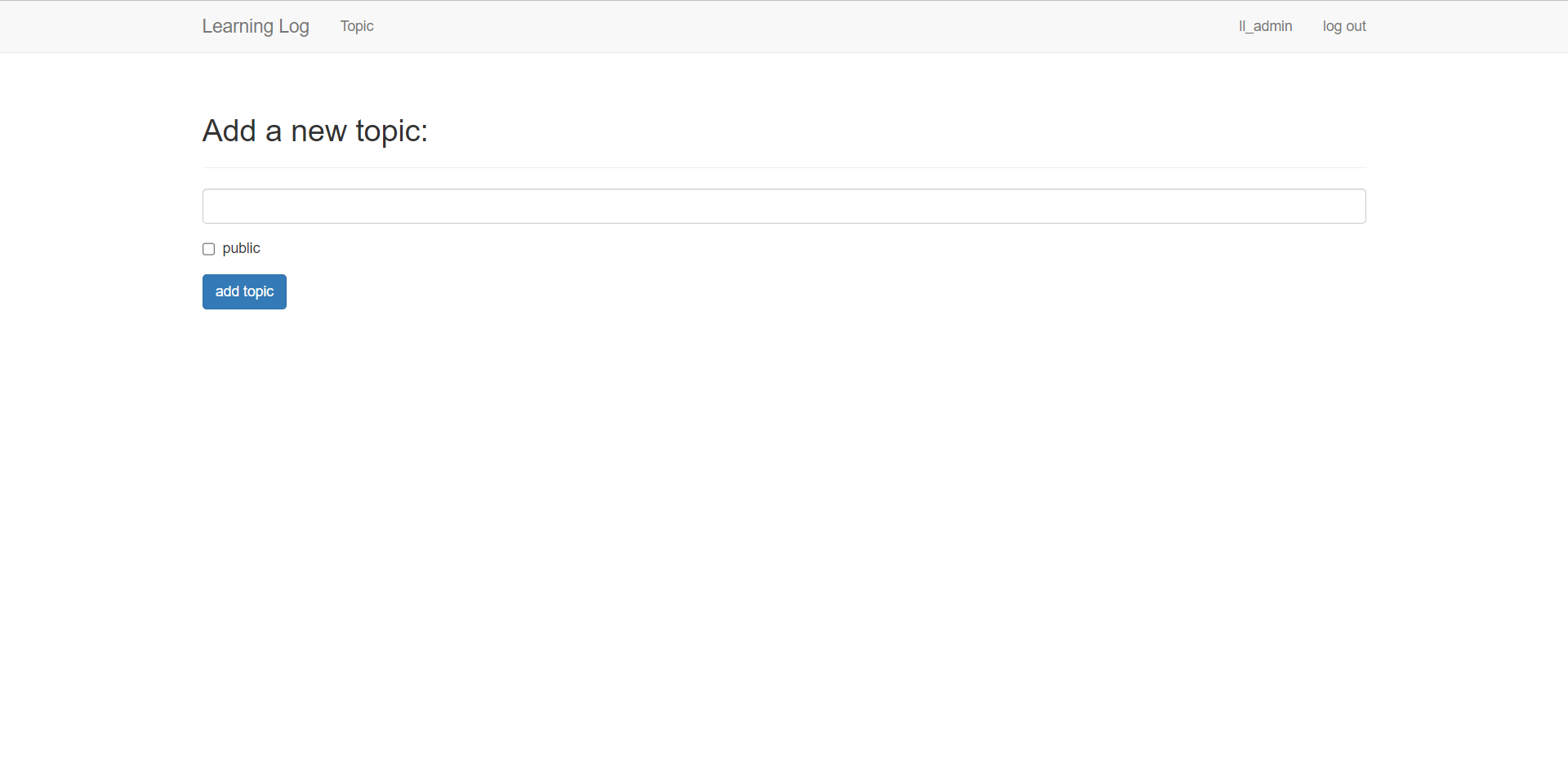


图36：topic添加界面

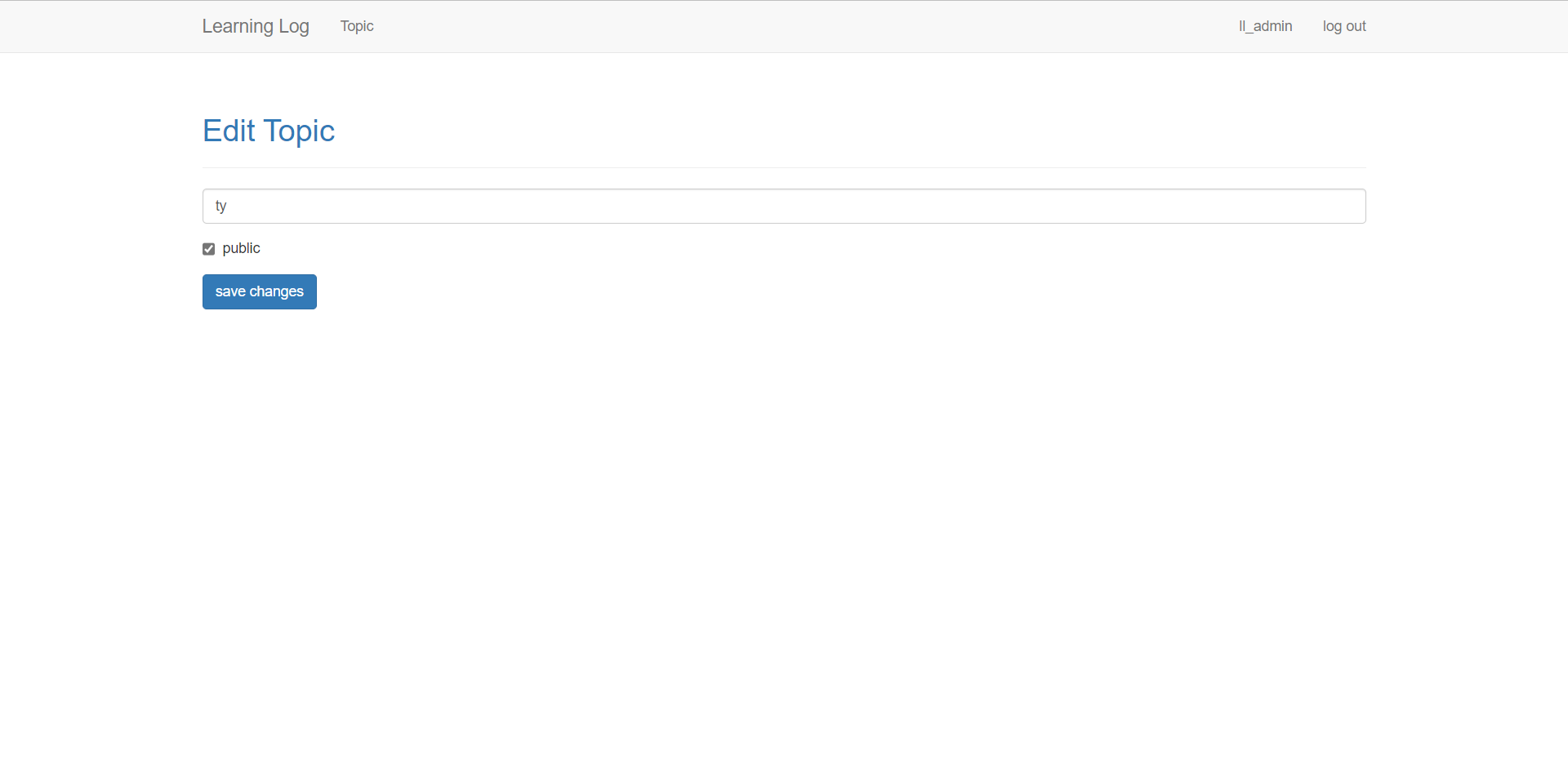


图37：topic修改界面

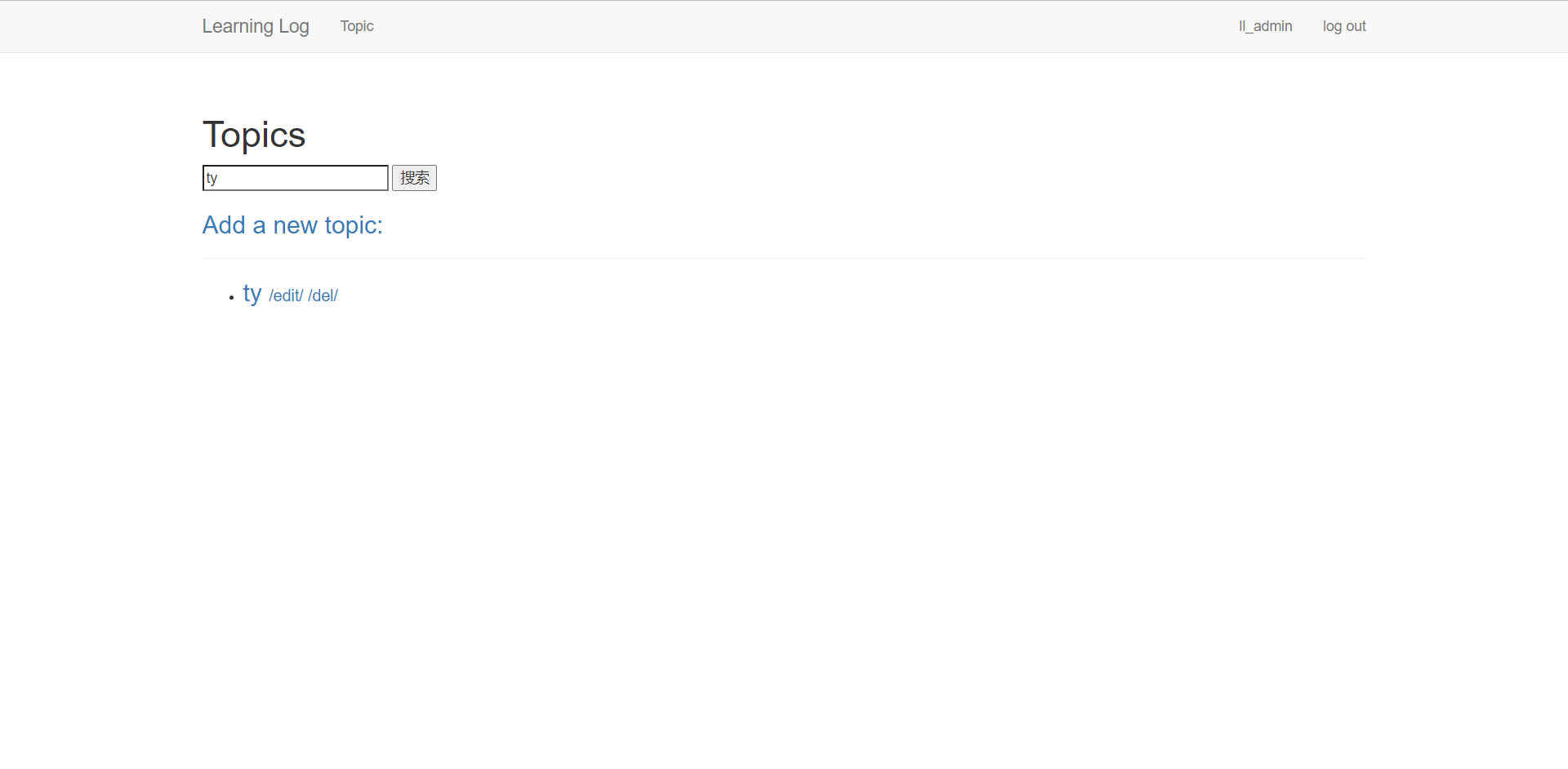


图38：topic查找界面

点击其中一个topic就可以进入entry界面了。界面显示了笔记的标题，添加这个笔记的时间，删除、修改、查找、添加等操作。Entry的内容采用的是富文本编辑框，可以对自己加粗，斜体，添加代码、图片、超链接等操作。查找时可以根据笔记的title或者他的内容进行查找。

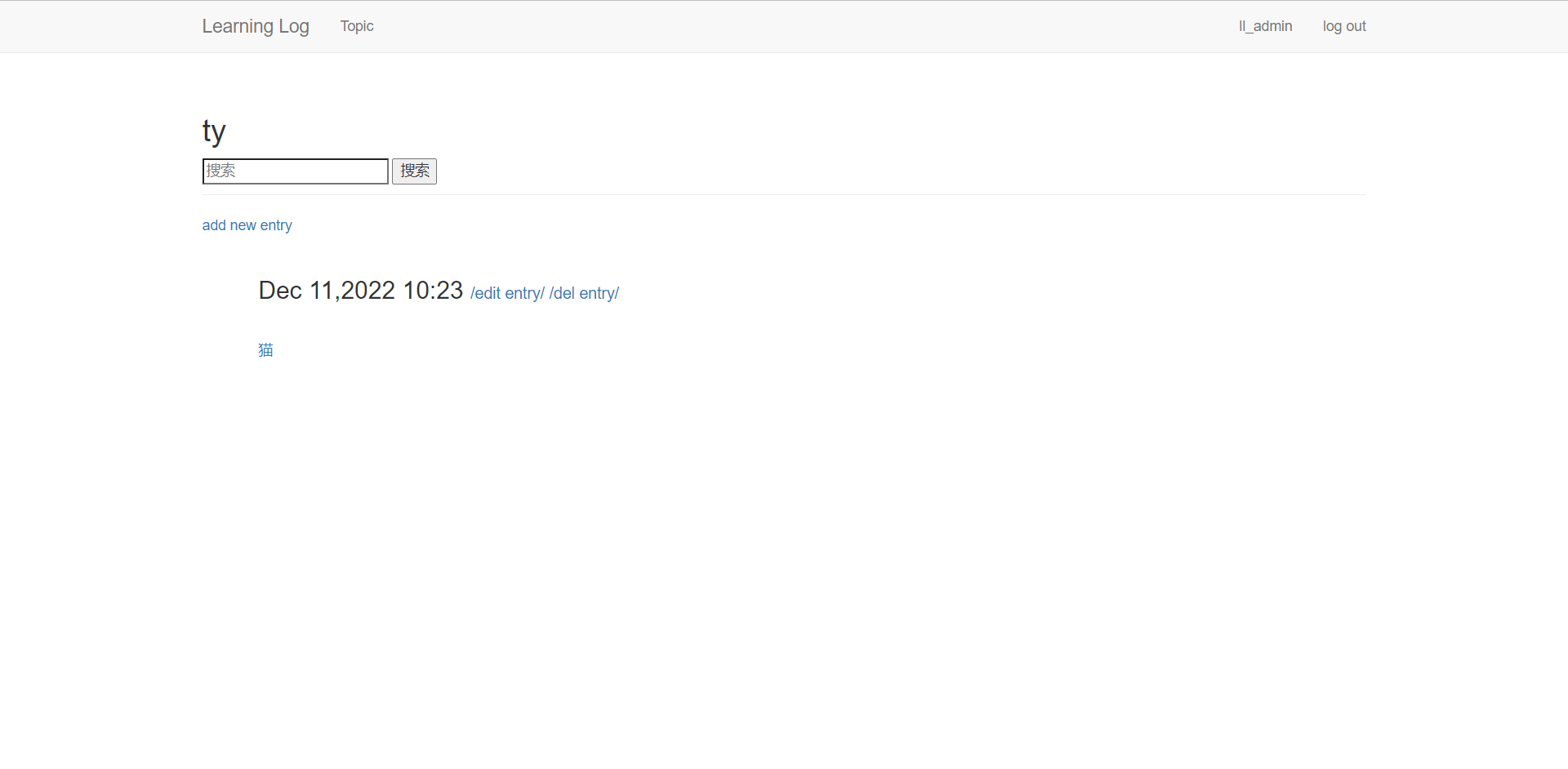


图39：entry主界面

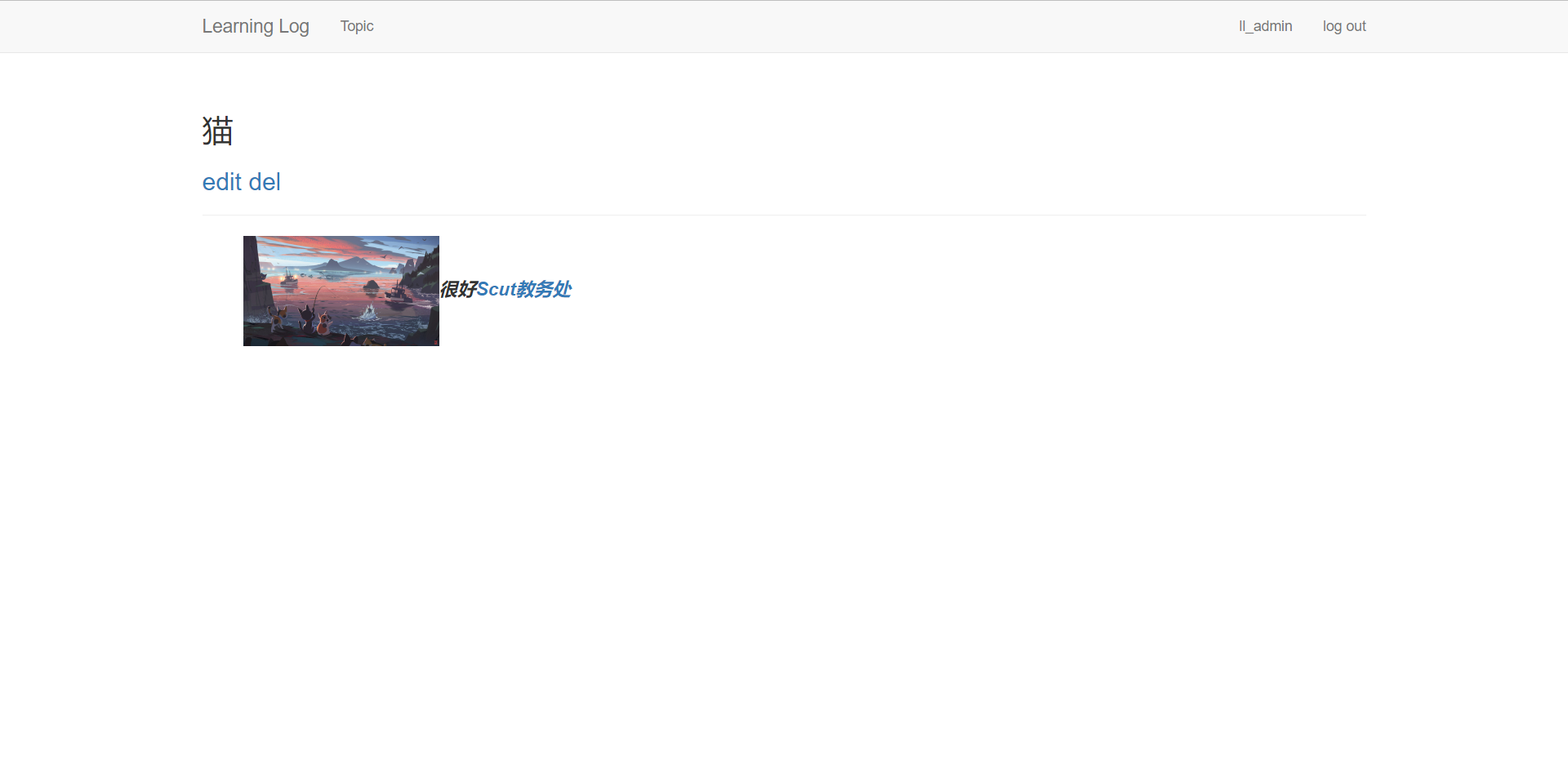


图40：entry内容界面

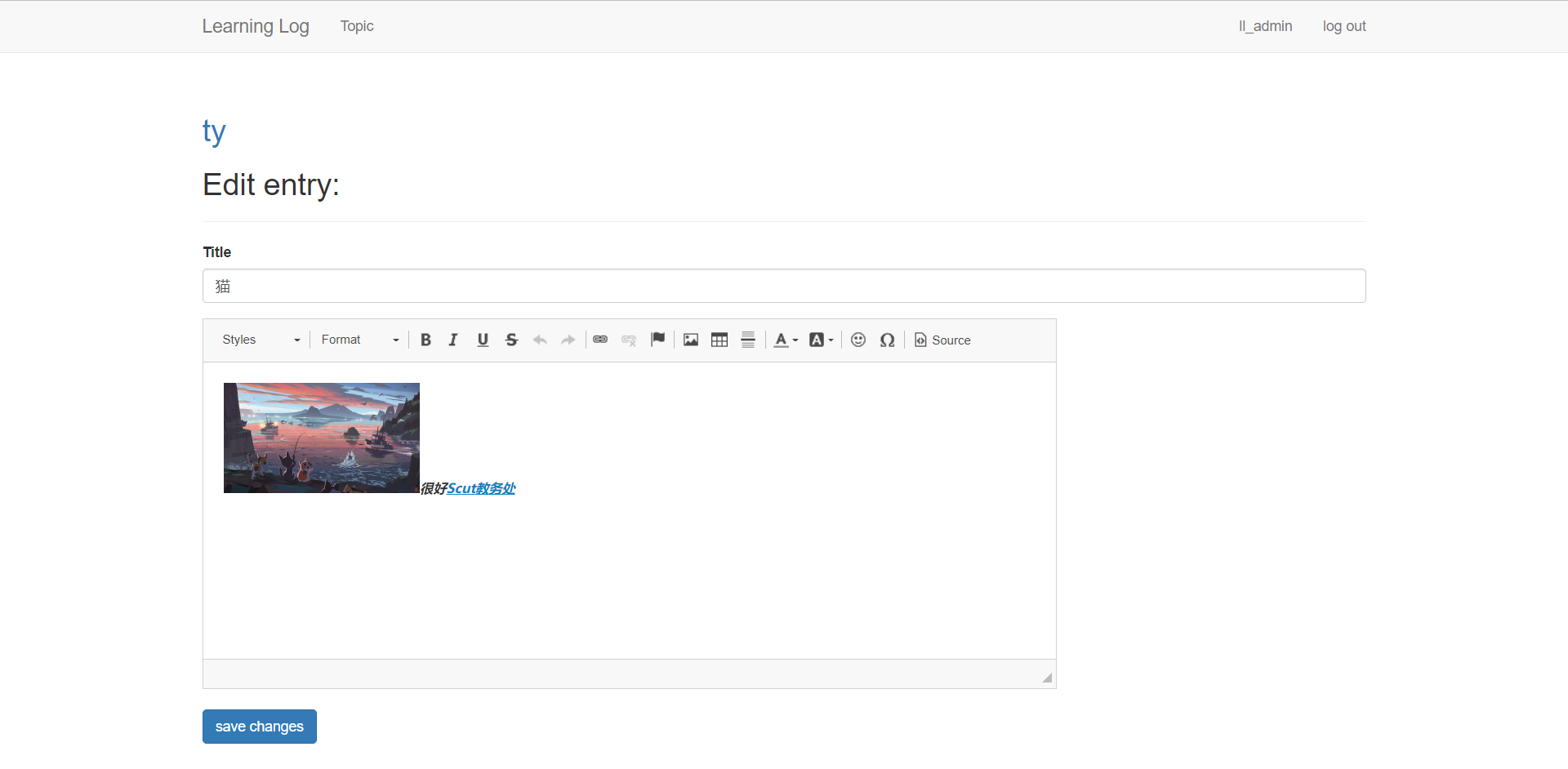


图41：entry修改界面

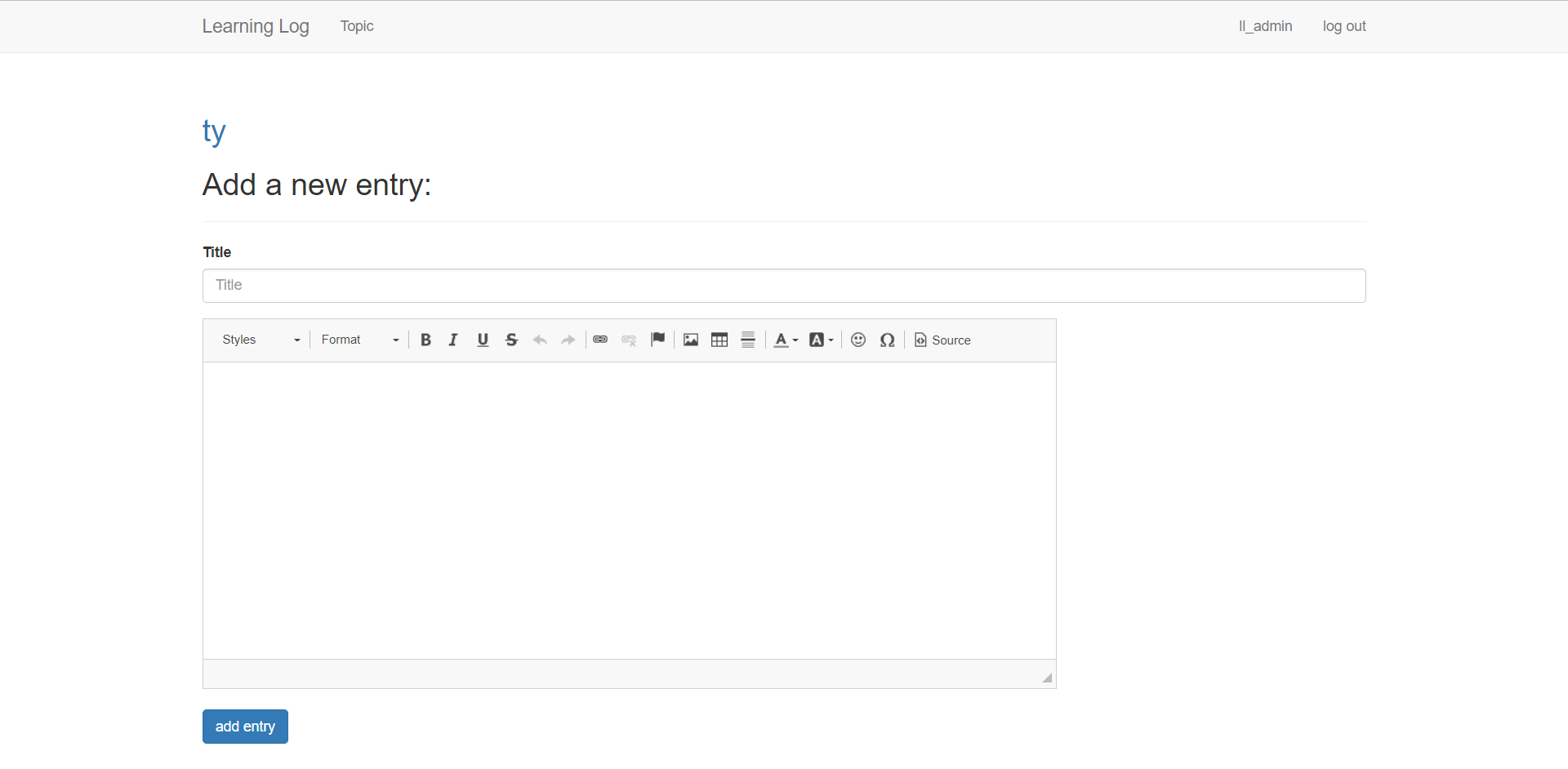


图42：entry新增界面

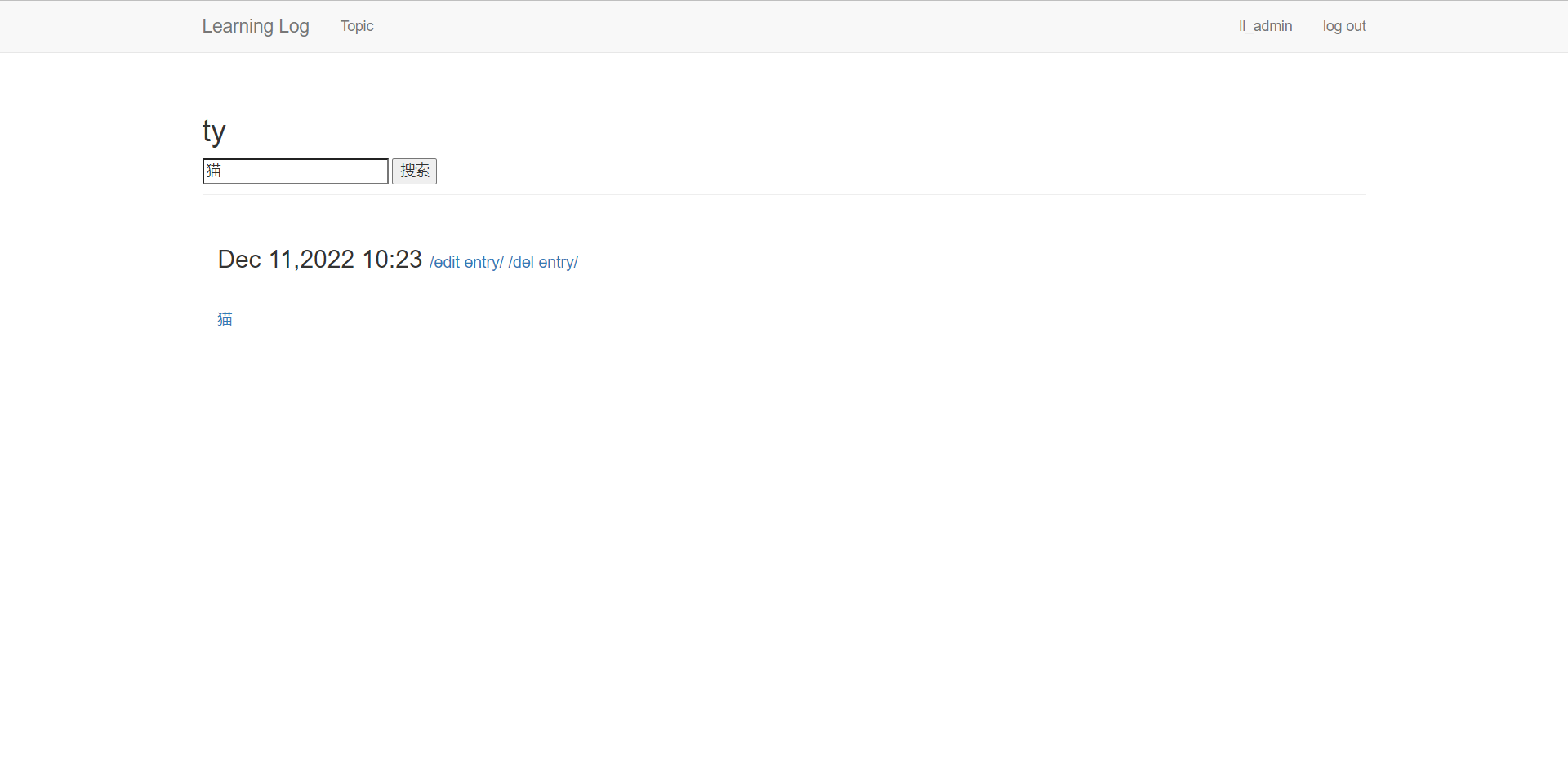


图43：entry查找界面

更多内容可以参考文件附带视频

分析：

“学习笔记”实现了一个笔记初步具备的所有需求功能，并且创造性的实现了共享笔记这一个市面少罕见的设计。我通过包括对此学习网站建立项目，创建应用程序，创建网页，管理用户，设置项目样式，建立自己的专属用户，包括注册账号，设计和使用密码，在网站中记录（用文字和图片均可）所需要记录的笔记并查看等。并且包括对所有笔记进行统筹管理，包括管理用户，编辑甚至删除笔记，查找笔记等功能。实现了一个笔记初步具备的所有需求功能 “学习笔记”这个项目实现了确定项目初期我对这个程序的所有需求。但其仍有少部分功能并不够完善（如：用户没有个人界面，个人简介等功能），并且缺乏一个开放多元的社区支持，无法与市面上的其他成熟的商品竞争。

五、课程设计总结

遇到的问题及解决问题的过程的思考：

这是我第一次接触Django，也是初次进行网页程序的系统设计。刚开始接触的时候，只能照猫画虎，跟着网上的教程去做，每一段代码的含义都不是很懂，一直都在囫囵吞枣的过程中。不过后来我通过不断的尝试着不断地修改每一段代码，查看他是网页中的变化，或查看它的报错信息。我开始逐渐明白了每一段的代码的意义以及用处，了解了“”model，view，form，url这几个Django中的主要部分的作用。一开始学习我也因为过什么都不懂，看不懂代码报错的愿意及其修复方法，但当我逐渐开始了解了每一块组件的作用，我开始逐渐学会了如何去修复bug，了解到了自己一些常犯的错误如路径输错，数据没有一一对应的传入，及其调用导致报错。Django的报错不像平时python编程一样，由编译器警告等，十分难以发现已经改正，需要的是耐心和毅力。而且报错后的提示也比之前的提示更加的晦涩难懂。一开始我对于Entry的编辑只存在于文本框之中，后来觉得只能够输入文本和我实现一个功能基本完善的“学习笔记”期望不是很相符，于是我改用了富文本的插件django-ckeditor。但一开始我并不能在网页中调出我需要的富文本输入框，无论如何我都找不出bug。后来通过阅读外国论坛我发现是版本问题，导致我重新做了一次大作业，后来发现只有管理界面能看到富文本输入框，但前端界面不行。最后是参考了其他相似项目具有富文本输入框的文件的源代码我才发现我缺少了一些css文件，最终解决了问题。根据关键词查找entry也是我的一大技术难点，因为一开始在views文件中filter出来的数据列表包含了不属于这个账户，不能被看到的entry，这显然是一个致命的bug。但我一开始无论如何都无法通过entry的显示网页传出其所属的topic.id给我的视图函数。让我感到无比的郁闷。最后是通过观察之前的代码如何传出其他数据，以及查看官方文档对于form的定义，最终找到了正确的方法。

课程设计过程的收获和体会：

通过“学习笔记”这一项目，我初步掌握了如何一个人通过上网查询有关资料，查考官方文档帮助自己成功地从无到有，从零基础开始，利用已学过的python基础知识，创建了一个功能强大，界面美观的软件。这次的大作业让我初步的了解到了网页的创建与配置的过程，初步学了Django框架的特性和使用方法。了解到了创建Web的注意事项，以及一些web程序的底层逻辑。让我更加了解了现实社会工业工程上是如何创造出一个受人喜欢的产品的。从这次实践中，我对于项目功能可能要求的代码难度有了更深度的认识，也对我们所学的知识有了更深度的成体系化的了解和掌握，应用时也当然更加的得心应手。我学会了自己去解决问题，实现猜想的实践逻辑。将多个程序文件相结合。最终形成的这么一个网站，凝聚了我很多的心血。了解到了制造出一个功能完善，界面美观的程序的不容易。希望在之后的日子里能够凭借已有的知识和能力去进行更好的程序、网页制作