接下来完成帖子的修改和删除部分。

修改和删除的部分，将写在个人中心的版块中，以实现个人对自己所发内容的操作。

**1.修改我的帖子**

1. views.py视图函数的编写

首先在个人中心>我的帖子页面添加修改入口



在html文件里添加此句：

<th><a href="{% url "users:update\_lostfound" lostfound.id %}">修改</th></a>

在app：users中修改views.py文件：

def lostfound\_update(request, id):  
 # 获取需要修改的具体帖子对象  
 lostfound = LostfoundPost.objects.get(id=id)  
 # 判断用户是否为 POST 提交表单数据  
 if request.method == "POST":  
 # 将提交的数据赋值到表单实例中  
 lostfound\_post\_form = LostfoundPostForm(data=request.POST)  
 # 判断提交的数据是否满足模型的要求  
 if lostfound\_post\_form.is\_valid():  
 # 保存新写入的 title、details 数据并保存  
 lostfound.title = request.POST['title']  
 lostfound.type = request.POST['type']  
 lostfound.location = request.POST['location']  
 lostfound.address = request.POST['address']  
 lostfound.objecttime = request.POST['objecttime']  
 lostfound.details = request.POST['details']  
 lostfound.save()  
 return redirect("users:mypost\_lostfound")  
 # 如果数据不合法，返回错误信息  
 else:  
 return HttpResponse("表单内容有误，请重新填写。")  
 # 如果用户 GET 请求获取数据  
 else:  
 # 创建表单类实例  
 lostfound\_post\_form = LostfoundPostForm()  
 # 赋值上下文，将 topic 帖子对象也传递进去，以便提取旧的内容  
 context = {'lostfound': lostfound, 'lostfound\_post\_form': lostfound\_post\_form}  
 # 将响应返回到模板中  
 return render(request, 'user/update\_lostfound.html', context)

此处同样需要通过跨页面传值获得id，通过id找出帖子后，需要先调取数据库中该条记录的旧有内容，显示在页面上；然后再保存更新内容。

1. Urls.py路由的分发

为了完成跨页面id传值，同样增加<int:id>/

path("update\_lostfound/<int:id>/", views.lostfound\_update, name='update\_lostfound'),

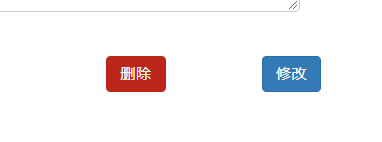
1. 前端页面的编写

大致与发布页面相同，同样需要标明name以保证页面正确传值。同样使用POST方法提交表单，使用{{ lostfound.xx }}调取和显示旧有的内容。

<form method="post" action=".">  
 {% csrf\_token %}  
 <div class=" form-group">  
 <label class="col-md-1" for="title">标题：</label>  
 <div class="col-md-6">  
 <input type="text" class="form-control" id="title" name="title" value="{{ lostfound.title }}">  
 </div>  
 </div>  
 <br>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-1 control-label" for="type">类型：</label>  
 <div class="col-md-6">  
 <select class="form-control" id="type" name="type">  
 <option value="失物招领">失物招领</option>  
 <option value="寻物启事">寻物启事</option>  
 </select>  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-1 control-label" for="location">地点：</label>  
 <div class="col-md-6">  
 <select class="form-control" id="location" name="location">  
 <option value="望江">望江</option>  
 <option value="江安">江安</option>  
 <option value="华西">华西</option>  
 </select>  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-1" for="address">详址：</label>  
 <div class="col-md-6">  
 <input class="form-control" type="text" id="address" name="address" value="{{ lostfound.address }}">  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-1 control-label" for="objecttime">日期：</label>  
 <div class="col-md-6">  
 <input class="form-control" type="text" id="objecttime" name="objecttime" value="{{ lostfound.objecttime }}"></div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-1" for="details">详情：</label>  
 <div class="col-md-6">  
 <textarea type="text" class="form-control" id="details" name="details" rows="3">{{ lostfound.details }}</textarea>  
 </div>  
 </div>  
 <br>  
 <br>  
 <!-- 提交按钮 -->  
 <button type="submit" class="btn btn-primary">修改</button>  
</form>

**2.删除我的帖子**

首先添加删除入口，在修改界面末尾增加删除按钮：



在修改的html页面增加删除按钮，鼠标点击事件绑定为comfirm\_delete()

<a href="#" onclick="confirm\_delete()"><div class="btn-delete">删除</div></a>

1. views.py视图函数的编写

在此同样需要获取id的值，以在数据库中找到指定的内容。

# 删除帖子  
def lostfound\_delete(request, id):  
 # 根据 id 获取需要删除的帖子  
 lostfound = LostfoundPost.objects.get(id=id)  
 # 调用.delete()方法删除帖子  
 lostfound.delete()  
 # 完成删除后返回帖子列表  
 return redirect("users:mypost\_lostfound")

1. 前端html调用layer弹窗

增加弹窗，可以解决用户不小心点到按钮结果删除的情况，确认后再进行删除，确保用户不是误删的。

弹窗是很常用的功能，但是想写出一个美观好用的弹窗却不容易。幸运的是我们不需要重复造轮子，早就有革命先驱做好相关的功能了，这里我们选择使用Layer弹窗组件。

layer是一款备受青睐的web弹层组件，具备全方位的解决方案。首先到官网下载Layer插件：<https://layer.layui.com/>

解压后将里面的\*\*layer文件夹（含有layer.js的）\*\*直接复制到项目的static文件夹下。

为了未来在所有页面都能使用Layer弹窗功能，在用户中心的模板中usercenter-base.html中通过标签引入：

<script src="{% static 'layer/layer.js' %}"></script>

在末尾添加调用内容：

<script>  
 // 删除文章的函数  
 function confirm\_delete() {  
 // 调用layer弹窗组件  
 layer.open({  
 // 弹窗标题  
 title: "确认删除",  
 // 正文  
 content: "确认删除这篇文章吗？",  
 // 点击确定按钮后调用的回调函数  
 yes: function(index, layero) {  
 // 指定应当前往的 url  
 location.href='{% url "users:lostfound\_delete" lostfound.id %}'  
 },  
 })  
 }  
</script>

location.href定义了点击确定键后应该前往的地址，即删除文章的url。

点击后，可以发现弹窗出现，点击确定后，方可完成删除操作。

