完成了model数据模型的创建后，我们就可以开始按数据库表字段创建表单，完成帖子的发布。这一过程的主要流程是：确定在settings.py中已经注册app，在views.py中完成函数编写，在forms.py中确定提交字段，将views.py中定义的函数写入urls.py路由字段中，确保能够正确调用，最后完成前端的编写，确定数据提交类型为form，使数据能够正确上传。

1. views.py视图函数的编写

依次引入下列包。

from django.shortcuts import render, redirect  
from django.http import HttpResponse  
from .forms import LostfoundPostForm  
from .models import LostfoundPost, Img

然后编写发帖函数lostfound\_create：

def lostfound\_create(request):  
 # 判断用户是否提交数据  
 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)  
 # 添加图片  
 img = Img(img\_url=request.FILES.get('img'))  
 img.save()  
 # 判断提交的数据是否满足模型的要求  
 if lostfound\_post\_form.is\_valid():  
 # 保存数据，但暂时不提交到数据库中  
 new\_lostfound = lostfound\_post\_form.save(commit=False)  
 # 指定当前用户为作者  
 new\_lostfound.author = request.user  
 # 将新文章保存到数据库中  
 new\_lostfound.save()  
 # 完成后返回到文章列表  
 return redirect("lostfound:lostfound\_list")  
 # 如果数据不合法，返回错误信息  
 else:  
 return HttpResponse("表单中有数据不合法")  
 # 如果用户请求获取数据  
 else:  
 # 创建表单类实例  
 lostfound\_post\_form = LostfoundPostForm()  
 # 赋值上下文  
 context = {'lostfound\_post\_form': lostfound\_post\_form}  
 # 返回模板  
 return render(request, 'lostfound/create.html', locals())

其中用到了两层if判断，先判断是否符合数据类型，若有错，应给提示用户提交数据有问题，之后才能够选择是否保存内容。其中，最后的 locals() 函数会以字典类型返回当前位置的全部局部变量。

但在此处，由于用户发帖是个需要事先登录的操作，所以在开头增加登录检查：

@login\_required  
def lostfound\_create(request):

同时，需要在开头引入登录判断的包：

from django.contrib.auth.decorators import login\_required

1. Forms.py表单的编写

# 引入表单类  
from django import forms  
# 引入帖子模型  
from .models import LostfoundPost

然后创建表单LostfoundPostForm

class LostfoundPostForm(forms.ModelForm):  
 class Meta:  
 # 指明数据模型来源  
 model = LostfoundPost  
 # 定义表单包含的字段  
 fields = ('title', 'type', 'location', 'address', 'objecttime', 'details')

1. Urls.py路由分发：

在一级路由的地方，我们已经编写了如下内容，确定了二级路由的导入位置。

path('lostfound/', include('lostfound.urls', namespace='lostfound')),

在二级路由处，同样要import的是path分发方式，和views函数，static确保文件上传。

from django.urls import path  
from . import views  
from django.conf import settings  
from django.conf.urls.static import static

然后在urlpattern中书写路径，最前面确定了网址访问格式，后面指定了具体调用的函数。

# 用户发帖  
path('lostfound-create/', views.lostfound\_create, name='lostfound\_create'),

1. 前端templates的编写

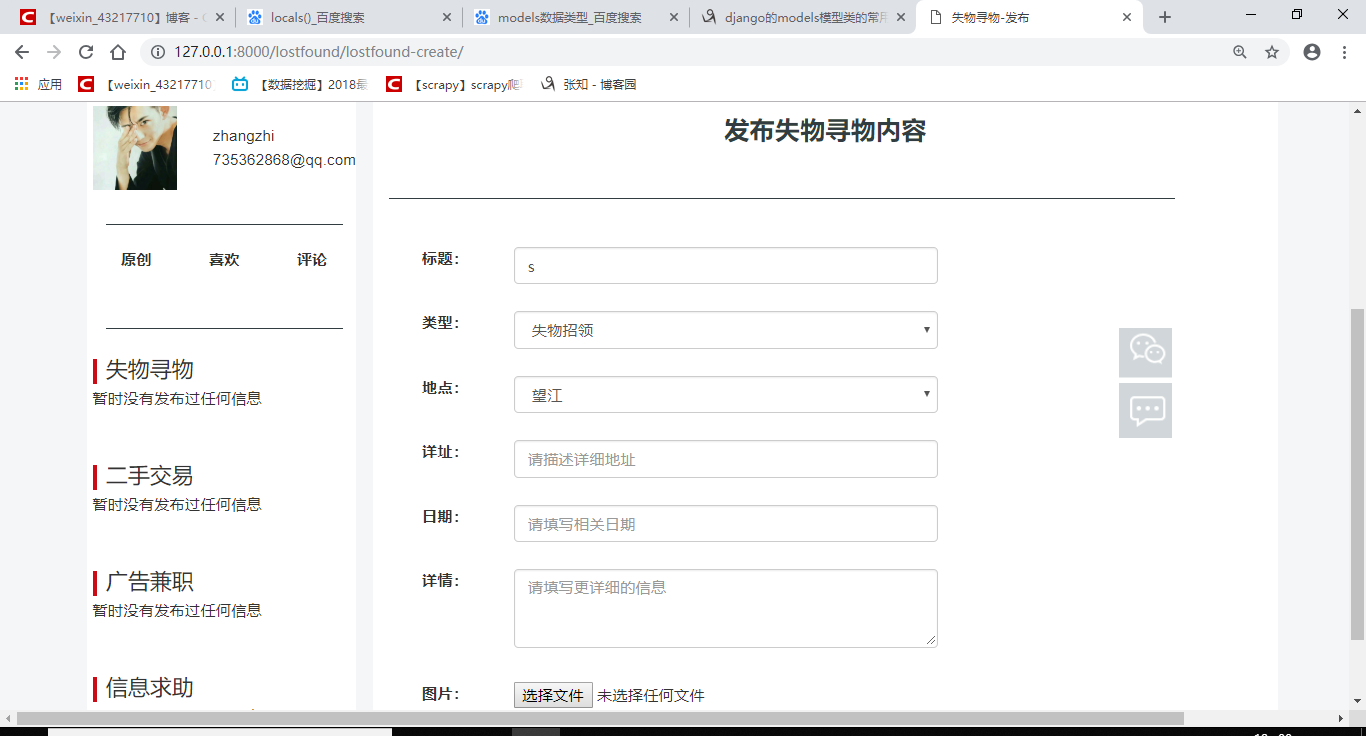
在templates/lostfound/下创建create.html，完成对basepage页面的继承，书写title。

<!-- extends表明此页面继承自 basepage.html 文件 -->  
{% extends "basepage.html" %} {% load staticfiles %}  
<!-- 写入 basepage.html 中定义的 title -->  
{% block title %} 失物寻物-发布 {% endblock title %}

页面布局使用的是bootstrap，使用form标签包裹表单提交内容。其中参数name确定了传值的内容，确保传值成功。

<div class="col-md-9" style="background:#ffffff;">  
 <div class="secondtitle">  
 <br>  
 <center>发布失物寻物内容</center>  
 </div>  
 <br>  
 <br>  
 <span style="display:block;width:90%;height:1px;background-color:#323d42;"></span>  
 <div class="col-md-8">  
 <form method="post" action="." enctype="multipart/form-data">  
 <!-- Django中需要POST数据的地方都必须有csrf\_token -->  
 {% csrf\_token %}  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-2" for="title">标题：</label>  
 <div class="col-md-10">  
 <input class="form-control" type="text" id="title" name="title" placeholder="请输入标题" required="required">  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-2 control-label" for="type">类型：</label>  
 <div class="col-md-10">  
 <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-2 control-label" for="location">地点：</label>  
 <div class="col-md-10">  
 <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-2" for="address">详址：</label>  
 <div class="col-md-10">  
 <input class="form-control" type="text" id="address" name="address" placeholder="请描述详细地址">  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-2 control-label" for="objecttime">日期：</label>  
 <div class="col-md-10">  
 <input class="form-control" type="text" id="objecttime" name="objecttime" placeholder="请填写相关日期">  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-2 control-label" for="details">详情：</label>  
 <div class="col-md-10">  
 <textarea class="form-control" type="text" id="details" name="details" rows="3" placeholder="请填写更详细的信息"></textarea>  
 </div>  
 </div>  
 <br>  
 <br>  
 <br>  
 <br>  
 <div class="form-group">  
 <label class="col-md-2 coontrol-label" for="img">图片：</label>  
 <div class="col-md-10">  
 <input type="file" id="img" name="img" placeholder="请上传图片">  
 </div>  
 </div>  
 <br>  
 <br>  
 <!-- 提交按钮 -->  
 <button type="submit" class="button">完成</button>  
 </form>  
 </div>  
 </div>

最后运行，打开浏览器，输入网址，尝试发帖。



尝试发帖，提交后，去数据库中查看记录更新：

