接下来需要完成帖子的总视图查看与具体一条的阅读。

1.lostfound\_list帖子列表总视图

1. views.py视图函数的编写

此页面主要展示网站内所有用户发布的内容，供用户浏览与查看。内容多了就需要涉及到分页，所以我们使用django的分页插件paginator：

# 导入分页用的包  
from django.core.paginator import Paginator

编写具体的list函数如下：

def lostfound\_list(request):  
 # 取出所有网站帖子  
 lost\_list = LostfoundPost.objects.all()  
 paginator = Paginator(lost\_list, 5)  
 page = request.GET.get('page')  
 losts = paginator.get\_page(page)  
 context = {'losts': losts}  
 return render(request, 'lostfound/list.html', locals())

1. Urls.py路由的分发

在urlpattern中书写内容：

# 帖子列表  
path('lostfound-list/', views.lostfound\_list, name='lostfound\_list'),

1. 前端页面的编写

同样继承baepage页面，完成title的书写。

增加定位导航：

<section class="content-header">  
 <ol class="breadcrumb">  
 <li><a href="{% url 'index' %}">主页</a></li>  
 <li class="active">失物寻物</li>  
 </ol>  
</section>

显示内容：

<section class="content">  
<div class="row">  
<div class="col-xs-10 col-md-offset-1">  
  
 <div class="box">  
 <div class="box-body">  
 <div class="col-md-12">  
 <div class="col-md-2">  
 <div class="title"style="font-size: 30px;color: #333;font-weight:bold;height: 80px;">  
 失物招领  
 </div>  
 </div>  
 <div class="col-md-10">  
 <div class="button" type="button"><a href="{% url 'lostfound:lostfound\_create' %}">发布信息</a>  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 {% for lost in losts %}  
 <div class="list-style">  
 <div class="col-md-2">  
 <img style="margin-top:10%;width:auto;height:auto;max-width:80%;max-height:80%;" src="{% static 'images/1.jpg' %}">  
 <!-- <img style="width:auto;height:auto;max-width:50%;max-height:50%;" src="/media/{{ img.img\_url }}"/> -->  
 </div>  
 <div class="col-md-10">  
 <div class="title"><a href="{% url 'lostfound:lostfound\_detail' lost.id %}">{{ lost.title }}</a></div>  
 <div class="content">  
 信息类型：{{ lost.type }}&nbsp;&nbsp;发布人:{{ lost.author }}<br/>  
 发布日期：{{ lost.created }}&nbsp;&nbsp;校区：{{ lost.location }}  
 </div>  
 </div>  
 </div>  
 {% empty %}  
 暂无数据！  
 {% endfor %}  
 </div>  
 <div class="col-md-12" align="center">  
 <!-- 页码导航 -->  
 <div class="pagination row">  
 <div class="m-auto">  
 <span class="step-links">  
 <!-- 如果不是第一页，则显示上翻按钮 -->  
 {% if losts.has\_previous %}  
 <a href="?page=1" class="btn btn-success">  
 &laquo; 1  
 </a>  
 <span>...</span>  
 <a href="?page={{ losts.previous\_page\_number }}"  
 class="btn btn-secondary"  
 >  
 {{ losts.previous\_page\_number }}  
 </a>  
 {% endif %}  
  
 <!-- 当前页面 -->  
 <span class="current btn btn-danger btn-lg">  
 {{ losts.number }}  
 </span>  
  
 <!-- 如果不是最末页，则显示下翻按钮 -->  
 {% if losts.has\_next %}  
 <a href="?page={{ losts.next\_page\_number }}"  
 class="btn btn-secondary"  
 >  
 {{ losts.next\_page\_number }}  
 </a>  
 <span>...</span>  
 <a href="?page={{ losts.paginator.num\_pages }}"  
 class="btn btn-success"  
 >  
 {{ losts.paginator.num\_pages }} &raquo;  
 </a>  
 {% endif %}  
 </span>  
 </div>  
 </div>  
 </div>  
  
 </div>  
 </div>  
 <!-- /.box -->  
</div>  
  
<!-- /.col -->  
</div>  
<!-- /.row -->  
</section>

分页具体显示逻辑，在前端完成编写。同时增加if判断，若没有内容，则显示“暂无数据”，如果有内容，则按排版进行显示。运行项目，我们可以看到刚才发布的内容。



**2.lostfound\_detail帖子详情视图**

（1）views.py视图函数的编写

此处涉及到跨页面传值，url传过来的id值，写入为参数，保证正确的详细信息被调用。之后分别取出对应的图片和文字内容：

# 查看帖子详情  
def lostfound\_detail(request, id):  
 lostfound = LostfoundPost.objects.get(id=id)  
 img = Img.objects.get(id=id)  
 context = {'lostfound': img}  
 content\_type = {'lostfound': lostfound }  
 return render(request, 'lostfound/detail.html', locals())

1. Urls.py路由的分发

在此处增加一级访问网址，<int:id>/确保此参数能够传入下一个页面。

# 帖子详情  
path('lostfound-detail/<int:id>/', views.lostfound\_detail, name='lostfound\_detail'),

1. 前端页面的编写

使用{{ lostfound.xx }}的格式取出记录中的具体字段：

<div class="col-md-9 ">  
 <div class="col-9 alert alert-success">  
 <div class="secondtitle">{{ lostfound.title }}</div>  
 <div class="basicinfo">{{ lostfound.created }}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;{{ lostfound.type }}</div>  
 </div>  
 <div class="col-9 info">  
 <div class="col-md-9">  
 <br>  
 <div class="col-md-3"><center>发布者：</center></div>  
 <div class="col-md-9">{{ lostfound.author }}</div>  
 <br>  
 <br>  
 <span style="display:block;margin-left:20px;width:80%;height:1px;background-color:#323d42;"></span>  
 <br>  
 <div class="col-md-3"><center>类型：</center></div>  
 <div class="col-md-9">{{ lostfound.type }}</div>  
 <br>  
 <br>  
 <div class="col-md-3"><center>校区：</center></div>  
 <div class="col-md-9">{{ lostfound.location }}</div>  
 <br>  
 <br>  
 <div class="col-md-3"><center>详址：</center></div>  
 <div class="col-md-9">{{ lostfound.address }}</div>  
 <br>  
 <br>  
 <div class="col-md-3"><center>相关时间：</center></div>  
 <div class="col-md-9">{{ lostfound.objecttime }}</div>  
 <br>  
 <br>  
 <div class="col-md-3"><center>详细描述：</center></div>  
 <div class="col-md-9">{{ lostfound.details }}</div>  
 <br>  
 <br>  
 <div class="col-md-3"><center>图片：</center></div>  
 <div class="col-md-9">  
 <img style="width:auto;height:auto;max-width:50%;max-height:50%;" src="/media/{{ img.img\_url }}"/>  
 </div>  
 <br>  
 <br>  
 <br>  
 </div>  
 <div class="col-md-1">  
 <div class="box">  
 <ul>  
 <br>  
 <br>  
 <br>  
 <br>  
 <li class="weixin">  
 <div class="weixin-logo"></div>  
 </li>  
 <li class="idea"></li>  
 </ul>  
 </div>  
 </div>  
 </div>  
</div>

一些小问题

1. django2.0后，urls.py中的django.conf.urls已经被django.urls所取代。如果是路径需要在路径的后面加上一个斜杠 /

from django.urls import path  
from article import views  
  
app\_name = 'article'  
  
urlpatterns = [  
 path('article-list/', views.article\_list, name='article\_list'),  
]

1. 编写views.py文件的时候，article忘记加s，导致整个页面空白，无法显示内容

def article\_list(request):  
 articles = ArticlePost.objects.all()  
 context = {'articles': articles}  
 return render(request, 'article/list.html', context)