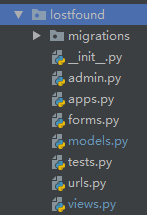
后台准备好以后，准备进入正式的开发。

我负责的模块是网站主要功能，即通过线上发帖实现平台信息互通。首先我们建立一个app，命名为lostfound，用于处理失物招领与寻物启事的帖子内容。

>python manage.py startapp lostfound，创建完成后，文件夹目录如图所示。其中migrations文件夹记录了数据库同步日志；手动创建forms.py与urls.py文件，分别用于处理表单提交内容与二级路由分发；models.py和views.py则分别用于处理数据库表结构和视图函数。



编写models.py文件：

（1）先引入需要用到的包

from django.db import models  
# timezone 用于处理时间相关事务  
from django.utils import timezone  
# 引入用户的表  
from users.models import UserProfile

1. 编写文字提交内容

class LostfoundPost(models.Model):  
 # 事先定义下拉框中的内容  
 失物招领 = '失物招领'  
 寻物启事 = '寻物启事'  
 TYPE\_CHOICE\_A = [  
 (失物招领, '失物招领'),  
 (寻物启事, '寻物启事'),  
 ]  
  
 江安 = '江安'  
 望江 = '望江'  
 华西 = '华西'  
 TYPE\_CHOICE\_B = [  
 (望江, '望江'),  
 (江安, '江安'),  
 (华西, '华西'),  
 ]  
 # 帖子创建时间。参数 default=timezone.now 指定其在创建数据时将默认写入当前的时间  
 created = models.DateTimeField(default=timezone.now)  
 # 帖子更新时间。参数 auto\_now=True 指定每次数据更新时自动写入当前时间  
 updated = models.DateTimeField(auto\_now=True)  
 # 帖子作者，自动读取作者  
 author = models.ForeignKey(UserProfile, on\_delete=models.CASCADE)  
 # 帖子标题，比较短，显示在首页  
 title = models.CharField(max\_length=100)  
 # 帖子类型  
 type = models.CharField(max\_length=10, default=0, choices=TYPE\_CHOICE\_A)  
 # 校区，可选  
 location = models.CharField(max\_length=100, default=0, choices=TYPE\_CHOICE\_B)  
 # 详细地址  
 address = models.CharField(max\_length=500, blank=True)  
 # 物品拾到/丢失时间  
 objecttime = models.CharField(max\_length=500, blank=True)  
 # 详细描述，详细页面才显示，自己添加  
 details = models.TextField(blank=True)  
  
 # 内部类 class Meta 用于给 model 定义元数据  
 class Meta:  
 # ordering 指定模型返回的数据的排列顺序  
 # '-created' 表明数据应该以倒序排列，即最新的数据永远显示在上部  
 ordering = ('-created',)  
  
 # 函数 \_\_str\_\_ 定义当调用对象的 str() 方法时的返回值内容  
 def \_\_str\_\_(self):  
 # return self.title 将帖子标题返回  
 return self.title

1. 增加上传图片功能

首先打开setting.py，设置文件上传根目录，设置完成后，文件将被上传在二级目录的media文件夹中。

# 设置上传文件的路径  
MEDIA\_URL = '/media/'  
MEDIA\_ROOT = os.path.join(BASE\_DIR,'media')

接下来修改app中的urls.py，在url末尾，添加内容：

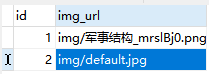
+static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

最后回到models.py，编写文件上传的数据库表：（upload\_to定义了文件保存的具体路径，此处为media/img/下；default定义了此处默认保存的图片，为img/default.jpg）

# 上传图片  
class Img(models.Model):  
 img\_url = models.ImageField(upload\_to='img', default='img/default.jpg')

全部完成后，进行数据库的同步和迁移：





Models.py文件的编写就到这里，接下来将开始依次实现增删查改。

但是仍然需要提出的是，在实现不同类别的帖子时，可能会涉及到不同的数据类型，https://www.cnblogs.com/Lin-Yi/p/7515584.html这篇文章总结了django的model中常用的数据类型，可在编写时进行参考。下面总结一下我在过程中用到的一些数据类型极其参数。

DateTimeField，时间数据类型，精确到日期时间。默认参数default=timezone.now 指定其在创建数据时将默认写入当前的时间。

created = models.DateTimeField(default=timezone.now)

参数 auto\_now=True 指定每次数据更新时自动写入当前时间。

updated = models.DateTimeField(auto\_now=True)

数据类型外键，on\_delete确定了删除方式。

author = models.ForeignKey(UserProfile, on\_delete=models.CASCADE)

短文本，max\_length确定了最长字符长度；default为默认值，choices为可选。同时要定义选项内容：  
type = models.CharField(max\_length=10, default=0, choices=TYPE\_CHOICE\_A)

失物招领 = '失物招领'  
寻物启事 = '寻物启事'  
TYPE\_CHOICE\_A = [  
 (失物招领, '失物招领'),  
 (寻物启事, '寻物启事'),  
]

长文本，blank=True为空值说明。  
details = models.TextField(blank=True)