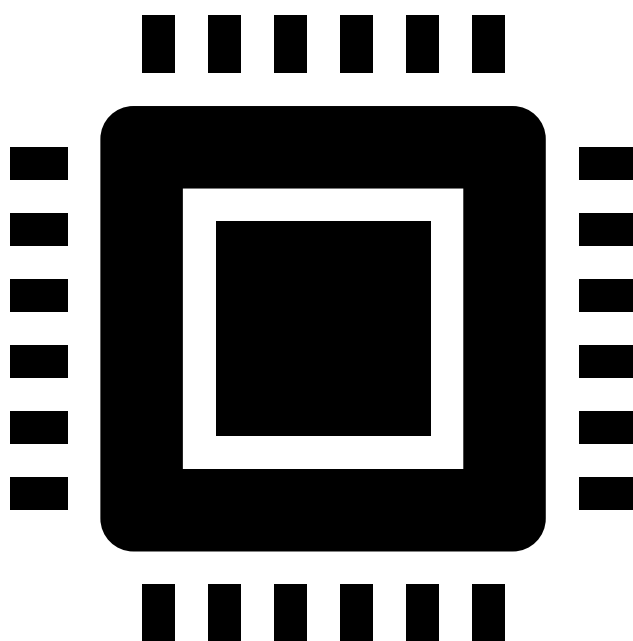


# 设计说明文档

## 《云分享》网站



## 目录

开发目的.....	3
项目环境.....	4
项目部署.....	4
使用方法.....	5
代码逻辑.....	8

# 开发目的

当下互联网已融入到每个人的生活之中，而随着互联网的发展，人们有了 使用互联网进行沟通的需求，这时就出现了各种的即时网络通信软件，而这些软件有一些弊端

1. 用户需要注册和登录
2. 用户发送的文件不可太大
3. 如果发的是视频、音频、图片 软件会放弃清晰度
4. 用户如果要使用则必须下载应用
5. 可能会泄露个人隐私
6. 因为用户容易遗忘自己存的信息，导致没有及时删除无用数据文件，服务端会存许多用户不需要的数据文件

这样就需要有一个可以解决以上问题的方案，而我就制作了一个名为"云分享"的网站，此网站拥有以下几点优点

1. 用户使用不需要注册和登录，使用户可以拥有更多时间注重内容，同时也减少了用户隐私的泄露
2. 这个解决方案为网站所以用户不需要下载任何其他应用只需一个浏览器就可以使用该应用
3. 这个网站不会对视频、音频、图片 进行任何压缩，保证了文件不会被更改
4. 此网站采用阅后即焚措施，既保护了用户的隐私，也保证了服务器不会存储无用信息文件

5. 此网站源码已经开源在 gitee 且代码清晰易读, 模块性强便于进行二次开发  
希望这个项目能切实际地使同学们可以便捷地与朋友分享自己的数据

## 项目环境

此项目使用了 **Python3.8**(计算机编程语言)和 **Django3.2**(Python Web MTV 框架)进行了网站后端的开发

而前端则使用了 **Html5**(超文本标记语言),**CSS**(层叠样式表)和 **Javascript**(解释型编程语言)以及 **Bootstrap**(前端开发框架)等技术来实现

## 项目部署

1. 解压代码文件压缩包 **webcopy.zip**
2. 在联网环境下双击 **python-3.8.5-amd64-webinstall.exe** 以安装 **Python** 依赖(电脑中已有的不用安装,但需保证 **Python** 版本为 3.8 及以上)
3. 进入 **webcopy** 文件夹, 打开命令提示符
4. 在联网环境下输入 **pip install -r requirements.txt** 安装库依赖
5. 输入 **python manage.py makemigrations** 进行迁移
6. 输入 **python manage.py migrate** 生成数据库
7. 输入 **python manage.py runserver 80** 运行网站(runserver 默认端口为 8000,这里更改为 80)

如果要备份数据库,需要将 **db.db** 和 **media/** 下的所有文件 进行备份

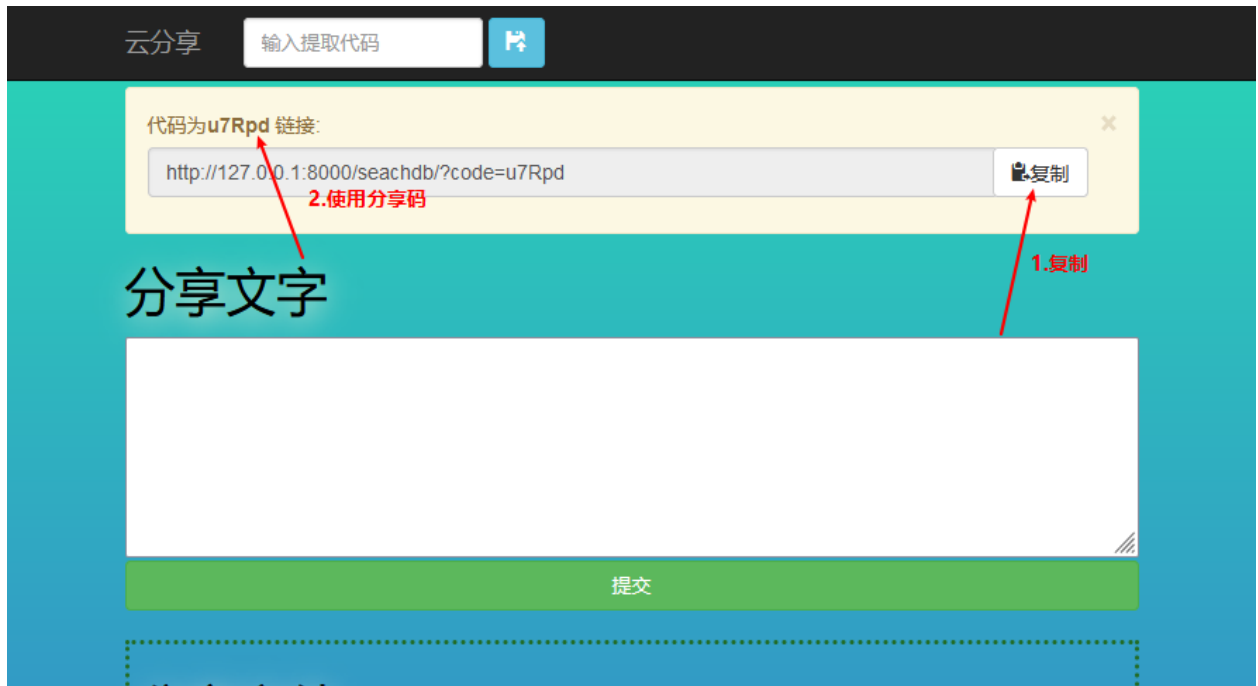
# 使用方法

**文字分享功能**可以让你将一些文字便捷地分享给别人，或在其他设备上获取内容。首先，先找到“分享文字”，然后在大输入框中输入自己想要分享的文字，写完后再点击提交按钮，信息会被提交上去



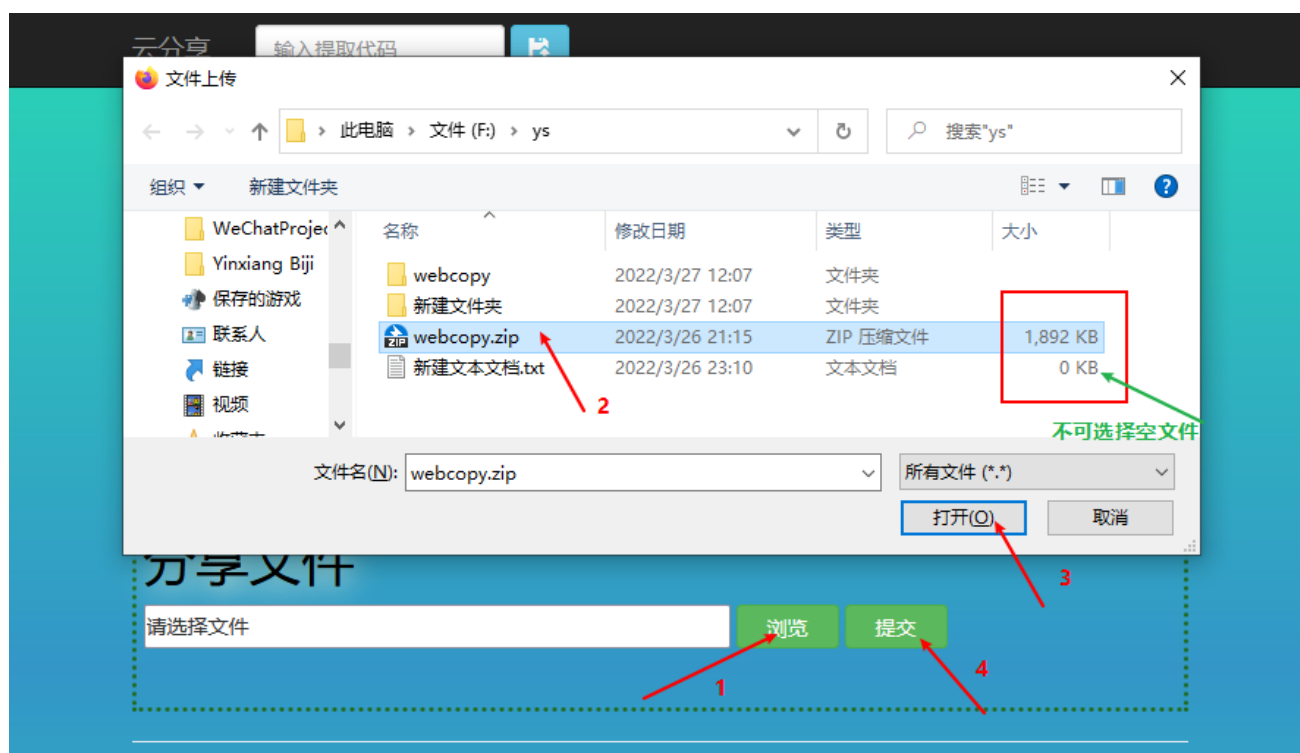
而页面会被重定向到另一个页面。这个界面顶上会有一个黄色的提示框，里面为你提供了两种分享的方式，你可以将信息对应的代码给别人，别人就可在页面顶端的提取功能，提取出你 分享的文字，你也可以点击复制按钮，将复制的链接发送给别人，别人通过访

问链接来查看你提交的内容。



**文件分享功能**可以让你将文件便捷地分享给别人，或在其他设备上获取文件。首先，先找到“分享文件”，然后点击浏览，选中一个非空的文件，点击提交按钮，文件会被提交上去，而页面会被重定向到另一个页面。这个页面顶上会有一个黄色的提示框，里面为你提供了两种分享的方式，你可以将信息对应的代码给别人，别人就可在页面顶端的提取功能，提取出你 分享的文件，你也可以点复制按钮，将复制的链接发送给别人，别人通过访问链接来查看你提交的文件。

## 《云分享》设计说明文档



如何查看其他人分享给你的文字或文件？首先，在页面的顶端会有一个小输入框，输入别人给你的分享码，点击旁边的蓝色提交按钮。



页面会被重定向到另一个页面。这个页面上会有别人上传的文字或下载按钮，取决于别人上传的是文字还是文件，如果要关闭则点击一下黑色背景即可。



## 代码逻辑

此项目的路由分为 5 个，分别负责了页面的访问，数据的查询，文字的提交，文件的提交以及文件的下载。用户访问网站后返回主页，用户提交后会生成随机字符串并将信息写入数据库，返回的主页会添加一个小提示框用来显示分享码和链接(链接都是 `/seachdb/code={分享码}` 这种形式的，对应了查询的链接)，等于用户省去了填入分享码并提交的环节。选择文件时，浏览器进行条件约束判断，确定文件不为空。用户在查看文件信息时，服务端会将生成的文件地址填入到 `a` 标签做的按钮的 `href` 里，这样用户点击后即可下载。当用户输入无效的分享码，服务器会自动返回 错误的提取码 。

下图 `urls.py` 的主要内容:



```
22 urlpatterns = [  
23     path('', indexhome),  
24     path('seachdb/', seachdb),  
25     path('uptext/', uptext),  
26     path('upfile/', fileup),  
27     path('admin/', admin.site.urls),  
28     re_path(r'media/(?P<path>.*)$', serve, {'document_root': settin  
29     re_path(r'^static/(?P<path>.*)$', serve, {'document_root': sett  
30     # path("media/<path>", down),  
31 ]
```

第 23 行用来路由到主页的代码

第 24 行用于路由到查询分享码对应的数据的函数

第 25 行用于路由到提交文本信息功能的函数

第 26 行用于路由到提交文件功能的函数

第 27 行用于路由到管理页面功能的函数

第 28 行用于路由到用户下载文件功能的函数

第 29 行用于路由到静态资源的访问功能的函数

下图 views.py 的主要内容:

```
4 > def indexhome(request): ...  
6  
7 > def seachdb(request): ...  
27  
28 > def uptext(request): ...  
38  
39 > def fileup(request): ...  
50
```

第 4 行的函数负责处理访问主页的请求

第 7 行的函数负责处理查看所提叫的内容的请求

第 28 行的函数负责处理用户提交的文字分享请求

第 39 行的函数负责处理用户提交的文件分享请求

下图 models.py 的主要内容:

```
5 class messgaelist(models.Model):
6     id=models.AutoField(primary_key=True,verbose_name='id')
7     code=models.CharField(max_length=5,verbose_name='代码',unique=True)
8     typetf=models.CharField(choices= (('t','text'),('f','file')),max_length=1,verbose_name='类型')
9     file=models.FileField(upload_to='file/'+str(uuid4())+'/',verbose_name="文件",null=True)
10    text=models.TextField(max_length=4096,verbose_name="文本",null=True, blank=True)
11    time=models.DateTimeField(auto_now_add=True,verbose_name="创建时间")
```

这个类定义了这个网站关于存取用户提交信息的数据库

第 6 行，id 为自增字段，主键

第 7 行，code 存储了随机生成的分享码，且保持唯一

第 8 行，typetf 存储了这条用户提交的数据为文字还是文件

第 9 行，file 存储了用户提交的文件对应的目录，可以为空

第 10 行，text 存储了用户提交的文本信息，可以为空

第 11 行，time 存储了用户提交信息的时间，且为自动生成