

```
1 #!/usr/bin/env python
2 """Django's command-line utility for administrative tasks
3 """
4 import os
5 import sys
6
7 def main():
8     os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
miniproject.settings')
9     try:
10         from django.core.management import
execute_from_command_line
11     except ImportError as exc:
12         raise ImportError(
13             "Couldn't import Django. Are you sure it's
installed and "
14             "available on your PYTHONPATH environment
variable? Did you "
15             "forget to activate a virtual environment?"
16         ) from exc
17     execute_from_command_line(sys.argv)
18
19
20 if __name__ == '__main__':
21     main()
22
```

File - D:\miniproject\Run Miniproject.bat

```
1 @echo off
2 python manage.py runserver
3 start localhost:8000
```

```

1 from scripts.main import ParentValues
2 import random as rand
3 import requests
4 import pickle
5
6
7 class Generator(ParentValues):
8     def generateKey(self):
9         positions = {}
10
11         if self.total_count < self.maximum_entries:
12
13             num_pos = rand.randint(0, 5)
14             if not num_pos == 0:
15                 positions[num_pos] = rand.randint(0, 9)
16
17             for i in range(1, 6):
18                 caps = False
19                 if i == num_pos:
20                     continue
21                 if rand.randint(0, 1) == 1:
22                     caps = True
23                 if caps:
24                     positions[i] = chr(rand.randint(65, 90
25 ))
26                 else:
27                     positions[i] = chr(rand.randint(97, 122
28 ))
29
30             tlink = ''
31             # print(f'{tlink} and {positions}')
32             for i in range(1, 6):
33                 tlink = tlink + str(positions[i])
34             if not self.keyPresent(tlink):
35                 fLink = self.host_name + tlink
36             else:
37                 self.generateKey()
38
39             self.total_count += 1
40             self.ageMapper[self.total_count] = fLink
41
42         else:
43
44             fLink = self.ageMapper[self.oldest]
45             # self.oldest+=1
46             if self.oldest == self.maximum_entries:

```

```

45         self.oldest = 1
46     else:
47         self.oldest += 1
48
49     return fLink
50
51     def keyPresent(self, key):
52         f = self.linkMapper.get(key)
53         if f == None:
54             return False
55         else:
56             return True
57
58     def validateURL(self, user_link):
59
60         # Part where https:// is appended to the start of
the URL string
61         if user_link[:4] == 'http':
62             if user_link[4:7] == '://':
63                 user_link = 'https://' + user_link[7:]
64
65             elif user_link[4:8] == 's://':
66                 pass
67             else:
68                 return('Invalid URL Syntax')
69
70         else:
71             user_link = 'https://' + user_link
72
73         #         try:
74         #             if user_link == 'https://':
75         #                 raise InvalidURLError
76         #         except:
77         #             print('No URL entered')
78
79         emptyLink = False
80         # print(user_link)
81         try:
82
83             if user_link == 'https://':
84                 emptyLink = True
85             try:
86                 if emptyLink:
87                     raise Exception('No URL Entered')
88             except:
89                 return('No URL Entered')

```

```

90
91
92         else:
93             validator = requests.get(user_link)
94
95             # print(f'{validator}')
96             # validator=requests.get(user_link)
97             f = True
98             for i in range(400, 452):
99                 if i == validator.status_code:
100                     return(
101                         f"Server Response Code: {
validator.status_code}\nThis link can't be parsed due to a
client error")
102                 f = False
103                 break
104             for i in range(500, 512):
105                 if i == validator.status_code:
106                     return(
107                         f"Server Response Code: {
validator.status_code}\nThis link can't be parsed due to a
server error")
108                 f = False
109                 break
110             if f:
111                 return self.addKey(user_link)
112         except requests.ConnectionError:
113             return('The site provided does not exist')
114         # return 'Lmao'
115
116     def addKey(self, user_link):
117         file = self.generateKey()
118         dat = file.split('/a?i=')
119         dat = str(dat[1])
120         self.linkMapper[dat] = user_link
121         self.writeValues()
122         print(self.linkMapper)
123         print(self.ageMapper)
124         return file
125
126
127     def writeValues(self):
128         a = {'total_count': self.total_count}
129         b = {'oldest': self.oldest}
130         with open('D:\\oldest.pickle', 'wb') as fp:
131             pickle.dump(b, fp, protocol=pickle.

```

```
131 HIGHEST_PROTOCOL)
132     with open('D:\\total.pickle', 'wb') as fp:
133         pickle.dump(a, fp, protocol=pickle.
HIGHEST_PROTOCOL)
134     with open('D:\\linkMapper.pickle', 'wb') as fp:
135         pickle.dump(self.linkMapper, fp, protocol=
pickle.HIGHEST_PROTOCOL)
136     with open('D:\\ageMapper.pickle', 'wb') as fp:
137         pickle.dump(self.ageMapper, fp, protocol=
pickle.HIGHEST_PROTOCOL)
138
```

```
1 from scripts.gen import Generator
2
3
4 g=Generator()
5 g.validateURL('google.com')
6 print(g.linkMapper)
7 print(g.ageMapper)
8
9 #NOT USED
```

```
1 import random as rand
2 import requests
3 import csv
4 import pickle
5
6
7 class ParentValues:
8     host_name = 'localhost:8000/a?i='
9     # test=10
10    try:
11        with open('D:\\linkMapper.pickle', 'rb') as fp:
12            linkMapper = pickle.load(fp)
13    except FileNotFoundError:
14        linkMapper = {}
15
16    try:
17        with open('D:\\ageMapper.pickle', 'rb') as fp:
18            ageMapper = pickle.load(fp)
19    except FileNotFoundError:
20        ageMapper = {}
21
22    try:
23        with open('D:\\total.pickle', 'rb') as fp:
24            total_count = pickle.load(fp)
25            total_count = int(total_count['total_count'])
26    except FileNotFoundError:
27        total_count = 0
28
29    try:
30        with open('D:\\oldest.pickle', 'rb') as fp:
31            oldest = pickle.load(fp)
32            oldest = int(oldest['oldest'])
33    except FileNotFoundError:
34        oldest = 1
35
36    maximum_entries = 52 * 52 * 52 * 52 * 10
37
```



```
1 import pickle
2
3 from scripts.main import ParentValues
4 from scripts.gen import Generator
5
6
7 class Retrieve(ParentValues):
8     def __init__(self, user_link):
9         self.user_link = user_link
10
11     def searchLink(self):
12         # print(ParentValues.test)
13         # with open('D:\LinkMapper.pickle', 'rb') as fp:
14         #     print(pickle.Load(fp))
15         try:
16             m=self.linkMapper[self.user_link]
17             return m
18         except Exception as efgr:
19             print(efgr)
20             exit(1)
21             # return 'bhai ye kya dale ho tum'
22
23     def lol(self):
24         return self.linkMapper
25
```

```
1 .centrewrapper{
2     text-align: center;
3 }
4
5 .ralign{
6     text-align: right;
7     font: normal 80%/normal Roboto, Helvetica, sans-serif;
8 }
9
10 .heading{
11     font: normal 500%/normal "American Purpose Casual 01",
    Helvetica, sans-serif;
12 }
13
14 input[type=text]{
15     margin-bottom: 2vw;
16     font: normal 120%/normal Calibri, Helvetica, sans-serif
17 ;
18     border: 2px solid red;
19     text-align: center;
20     border-radius: 3vw;
21     padding: 10px 10px;
22     width: 20%;
23     transition: width 0.5s ease-in-out;
24 }
25 input[type=text]:focus{
26     outline: none;
27     width: 60%;
28     background-color: ;
29 }
30
31 .short{
32     background-color: white;
33     border: 2px solid black;
34     -webkit-border-radius: 3.125em;
35     border-radius: 3.125em;
36     font: normal 150%/normal Calibri, Helvetica, sans-serif
37 ;
38     text-align: center;
39     -o-text-overflow: clip;
40     text-overflow: clip;
41     padding: 0.25em 1px;
42     width: 25%;
43     text-align: center;
```

```
44 }
45
46 .short:hover{
47     background-color: lightpink;
48     box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px
49     0 rgba(0,0,0,0.19);
50 }
51 .short:active{
52     background-color: lightcoral;
53     transform: translateY(4px);
54 }
55
```

```

1  <!--{% Load static %}-->
2  <html>
3      <head>
4          <meta name="viewport" content="initial-scale=1,
width=device-width, height=device-height viewport-fit=cover
">
5          <meta name="author" content="Deepanshu Bakhru,
Jatin Advani">
6          <meta name="application-name" content="URL
Shortener">
7          <meta name="description" content="Shortens long
URLs for easy and convenient sharing">
8          <link rel="stylesheet" type="text/css" href="../../
website/css/styling.css">
9          <title>URL Shortener</title>
10     </head>
11     <body style="background-image: url(../website/graphics/
69.png)">
12         <div class="centrewrapper">
13             <p class="ralign">
14                 Python Project by<br>
15                 Deepanshu Bakhru<br>
16                 Jatin Advani<br>
17             </p>
18             <p class="heading">
19                 URL SHORTENER
20             </p>
21             <form name="app" method="POST" action="upload">
22                 {% csrf_token %}
23                 <input type="text" name="entry" placeholder
="Enter link to shorten" value="{{input}}">
24                 <br>
25                 <button class="short" type="submit" name="
submit" >SHORTEN</button>
26                 <p name="output" >
27                     <a href="{{output}}" target="_blank"
rel="noopener noreferrer">Your link has been shortened</a>
28 <!--
                                anchor this to the whatever-->
29                 </p>
30             </form>
31         </div>
32     </body>
33 </html>

```

File - D:\miniproject\shortener\apps.py

```
1 from django.apps import AppConfig
2
3
4 class ShortenerConfig(AppConfig):
5     name = 'shortener'
6
```

File - D:\miniproject\shortener\urls.py

```
1 from django.contrib import admin
2 from django.urls import path
3 from . import views
4
5 urlpatterns = [
6     path('', views.home),
7     path('upload', views.upload),
8     path('a', views.a) #this calls views.py me def a().
   this is of the form localhost:8000/a?id=xyz
9 ]
10
```

File - D:\miniproject\shortener\admin.py

```
1 from django.contrib import admin
2
3 # Register your models here.
4
```

File - D:\miniproject\shortener\tests.py

```
1 from django.test import TestCase
2
3 # Create your tests here.
4
```



```
1 from django.http import HttpResponse
2 from django.shortcuts import render, redirect
3 from scripts.main import ParentValues
4 from scripts.gen import Generator
5 from scripts.retrieval import Retrieve
6
7 # Create your views here.
8 output = ''
9
10
11 def home(request):
12     return render(request, 'html/index.html', {'input': ''
13     , 'output': output})
14
15 def upload(request):
16     urlobject = Generator()
17     output = urlobject.validateURL(request.POST['entry'])
18     # print(output + 'lol')
19     return render(request, 'html/index.html', {'input': ''
20     , 'output': output})
21
22 def a(request):
23     part = request.GET['i'] # i is key associated to value
24     . e.g. localhost:8000/a?i=000
25     print(part)
26     # gotta append https:// to link
27     r = Retrieve(part)
28     return redirect(r.searchLink())
29     # call retrieval for 'part' variable
```

File - D:\miniproject\shortener\models.py

```
1 from django.db import models
2
3 # Create your models here.
4
```

```
1 """
2 ASGI config for miniproject project.
3
4 It exposes the ASGI callable as a module-level variable
5 named ``application``.
6
7 For more information on this file, see
8 https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/
9
10 import os
11
12 from django.core.asgi import get_asgi_application
13
14 os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
15     miniproject.settings')
16
17 application = get_asgi_application()
```

```
1 """miniproject URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more
  information please see:
4     https://docs.djangoproject.com/en/3.0/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name
   ='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(),
   name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls
   import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('
   blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18
19
20 urlpatterns = [
21     path('', include('shortener.urls')),
22     path('admin/', admin.site.urls),
23 ]
24
```

```
1 """
2 WSGI config for miniproject project.
3
4 It exposes the WSGI callable as a module-level variable
   named ``application``.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/
8 """
9
10 import os
11
12 from django.core.wsgi import get_wsgi_application
13
14 os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
   miniproject.settings')
15
16 application = get_wsgi_application()
17
```

```
1 """
2 Django settings for miniproject project.
3
4 Generated by 'django-admin startproject' using Django 3.0.4
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.0/ref/settings/
11 """
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(
16     BASE_DIR, ...)
17 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(
18     __file__)))
19
20 # Quick-start development settings - unsuitable for
21 # production
22 # See https://docs.djangoproject.com/en/3.0/howto/
23 # deployment/checklist/
24
25 # SECURITY WARNING: keep the secret key used in production
26 # secret!
27 SECRET_KEY = 'hj!wsc8d988m#)i9$6@5=v^sm*m(0$!&mw2oz=#s^tnz&
28     83a39'
29
30 # SECURITY WARNING: don't run with debug turned on in
31 # production!
32 DEBUG = True
33
34 ALLOWED_HOSTS = []
35
36 # Application definition
37
38 INSTALLED_APPS = [
39     'django.contrib.admin',
40     'django.contrib.auth',
41     'django.contrib.contenttypes',
42     'django.contrib.sessions',
43     'django.contrib.messages',
44     'django.contrib.staticfiles',
45     'shortener',
46 ]
```

```

39 ]
40
41 MIDDLEWARE = [
42     'django.middleware.security.SecurityMiddleware',
43     'django.contrib.sessions.middleware.SessionMiddleware',
44     'django.middleware.common.CommonMiddleware',
45     'django.middleware.csrf.CsrfViewMiddleware',
46     'django.contrib.auth.middleware.
AuthenticationMiddleware',
47     'django.contrib.messages.middleware.MessageMiddleware',
48     'django.middleware.clickjacking.XFrameOptionsMiddleware
',
49 ]
50
51 ROOT_URLCONF = 'miniproject.urls'
52
53 TEMPLATES = [
54     {
55         'BACKEND': 'django.template.backends.django.
DjangoTemplates',
56         'DIRS': [os.path.join(BASE_DIR, 'website')],
57         'APP_DIRS': True,
58         'OPTIONS': {
59             'context_processors': [
60                 'django.template.context_processors.debug',
61                 'django.template.context_processors.request
',
62                 'django.contrib.auth.context_processors.
auth',
63                 'django.contrib.messages.context_processors
.messages',
64             ],
65         },
66     },
67 ]
68
69 WSGI_APPLICATION = 'miniproject.wsgi.application'
70
71 # Database
72 # https://docs.djangoproject.com/en/3.0/ref/settings/#
databases
73
74 DATABASES = {
75     'default': {
76         'ENGINE': 'django.db.backends.sqlite3',
77         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),

```

```
78     }
79 }
80
81 # Password validation
82 # https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
83
84 AUTH_PASSWORD_VALIDATORS = [
85     {
86         'NAME': 'django.contrib.auth.password_validation.
UserAttributeSimilarityValidator',
87     },
88     {
89         'NAME': 'django.contrib.auth.password_validation.
MinimumLengthValidator',
90     },
91     {
92         'NAME': 'django.contrib.auth.password_validation.
CommonPasswordValidator',
93     },
94     {
95         'NAME': 'django.contrib.auth.password_validation.
NumericPasswordValidator',
96     },
97 ]
98
99 # Internationalization
100 # https://docs.djangoproject.com/en/3.0/topics/i18n/
101
102 LANGUAGE_CODE = 'en-us'
103
104 TIME_ZONE = 'UTC'
105
106 USE_I18N = True
107
108 USE_L10N = True
109
110 USE_TZ = True
111
112 # Static files (CSS, JavaScript, Images)
113 # https://docs.djangoproject.com/en/3.0/howto/static-files/
114 /
115 STATIC_URL = '/website/'
116
117 STATICFILES_DIRS = [
```



File - D:\miniproject\miniproject\settings.py

```
118     os.path.join(BASE_DIR, "website"),  
119 ]
```