

APP

gamenews

asgi.py

```
1  """
2  ASGI config for gamenews project.
3  """
4  import os
5  from django.core.asgi import get_asgi_application
6  os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'gamenews.settings')
7  application = get_asgi_application()
8  """
```

Settings:
🌙 ☒ ☀️

Results:
All clear, no errors found

Settings.py

Some lines of standard Django are too long and impossible to shorten. Attempts to reformat line 24 created different errors

```
1  """
2  Settings for gamenews project.
3  """
4  from pathlib import Path
5  import os
6  import sys
7  import dj_database_url
8  if os.path.isfile('env.py'):
9      import env
10
11  # Build paths inside the project like this: BASE_DIR / 'subdir'.
12  BASE_DIR = Path(__file__).resolve().parent.parent
13  TEMPLATES_DIR = os.path.join(BASE_DIR, 'templates')
14
15  # Secret Key
16  SECRET_KEY = os.environ.get('SECRET_KEY')
17
18  # Debug
```

Settings:
🌙 ☒ ☀️

Results:
24: E501 line too long (102 > 79 characters)
98: E501 line too long (91 > 79 characters)
101: E501 line too long (81 > 79 characters)
104: E501 line too long (82 > 79 characters)
107: E501 line too long (83 > 79 characters)

Urls.py

<pre>1 """ 2 url configuration for gamenews project 3 """ 4 from django.contrib import admin 5 from django.urls import path, include 6 7 urlpatterns = [8 path("report/", include("report.urls"), name="report-urls"), 9 path("accounts/", include("allauth.urls")), 10 path('admin/', admin.site.urls), 11 path("", include("news.urls"), name="news-urls"), 12 path('summernote/', include('django_summernote.urls')), 13] 14 </pre>	<p>Settings:</p> <p>🌙 <input checked="" type="checkbox"/> ☀️</p> <hr/> <p>Results:</p> <p>All clear, no errors found</p>
---	--

Wsgi.py

<pre>1 """ 2 WSGI configuration for gamenews project. 3 """ 4 5 import os 6 7 from django.core.wsgi import get_wsgi_application 8 9 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'gamenews.settings') 10 11 application = get_wsgi_application() 12 </pre>	<p>Settings:</p> <p>🌙 <input checked="" type="checkbox"/> ☀️</p> <hr/> <p>Results:</p> <p>All clear, no errors found</p>
--	--




App

News

Admin.py

```
1 """admin for the news app"""
2 from django.contrib import admin
3 from django_summernote.admin import SummernoteModelAdmin
4 from .models import Thread, Response
5
6
7 @admin.register(Thread)
8 class ThreadAdmin(SummernoteModelAdmin):
9     list_display = ('title', 'slug', 'posted_on')
10    search_fields = ('title', 'posted_on')
11    list_filter = ('posted_on',)
12    prepopulated_fields = {'slug': ('title',)}
13    summernote_fields = ('body',)
14
15
16 @admin.register(Response)
17 class ResponseAdmin(SummernoteModelAdmin):
18     list_display = ('thread', 'poster', 'posted_on')
19     search_fields = ('thread', 'poster', 'posted_on')
20     list_filter = ('posted_on', 'poster',)
21     summernote_fields = ('content',)
22
```

Settings:

Results:

All clear, no errors found

Apps.py

```
1 """apps for the news app"""
2 from django.apps import AppConfig
3
4
5 class NewsConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'news'
8
```

Settings:

Results:

All clear, no errors found

Forms.py

```
1 """forms for the news app"""
2 from django import forms
3 from .models import Response, Thread
4
5
6 # form to respond to posts
7 class ResponseForm(forms.ModelForm):
8     class Meta:
9         model = Response
10        fields = ('content',)
11
```

Settings:

Results:




All clear, no errors found

Models.py

```
8 Stores a thread post related to the User model
9 """
10 title = models.CharField(max_length=150, unique=True, blank=False)
11 slug = models.SlugField(max_length=200, unique=True)
12 poster = models.ForeignKey(User, on_delete=models.RESTRICT,
13                             related_name="news_posts")
14 body = models.TextField(blank=False)
15 posted_on = models.DateTimeField(auto_now_add=True)
16
17 class Meta:
18     ordering = ["-posted_on"]
19
20 """returns a string representation of thread"""
21 def __str__(self):
22     return f"{self.title} - {self.poster}"
23
24
25 class Response(models.Model):
26     """

```

Settings:

Results:

All clear, no errors found

Urls.py

```
1 """
2 url patterns for news app
3 """
4 from django.urls import path
5 from . import views
6
7
8 urlpatterns = [
9     path('', views.Threadlist.as_view(), name='threadlist'),
10    path('<slug:slug>/', views.thread_detail, name='thread_detail'),
11    path('welcome-to-the-site/', views.thread_detail, name='about_us'),
12    path('<slug:slug>/edit_response/<int:response_id>',
13         views.response_edit, name='response_edit'),
14    path('<slug:slug>/delete_response/<int:response_id>',
15         views.response_delete, name='response_delete'),
16 ]
17
```

Settings:
🌙 ☒

Results:
All clear, no errors found

Views.py

```
89         response.thread = thread
90         response.save()
91         messages.add_message(request, messages.SUCCESS,
92                             'Response updated!')
93     else:
94         messages.add_message(request, messages.ERROR,
95                             'Error updating response!')
96
97     return HttpResponseRedirect(reverse('thread_detail', args=[slug]))
98
99
100 def response_delete(request, slug, response_id):
101     """
102     view to delete responses
103
104     **Context**
105
106     ``thread``
107     | An instance of :model:`news.Thread`
108     ``response``
109     | a response related to the thread
110     """
111     queryset = Response.objects.filter(visible=True)
112     thread = get_object_or_404(Thread, slug=slug)
113     response = get_object_or_404(queryset, pk=response_id)
```

Settings:
🌙 ☒

Results:
All clear, no errors found




App

Report

Admin.py

```
1 """admin for the report app"""
2 from django.contrib import admin
3 from django_summernote.admin import SummernoteModelAdmin
4 from .models import Report, ReportUser
5
6
7 @admin.register(Report)
8 class ReportAdmin(SummernoteModelAdmin):
9
10     summernote_fields = ('content',)
11
12
13 @admin.register(ReportUser)
14 class ReportUserAdmin(SummernoteModelAdmin):
15
16     list_display = ('reason_for_report', 'examined',)
17
```

Settings:

Results:

All clear, no errors found

Apps.py

```
1 """apps for the report app"""
2 from django.apps import AppConfig
3
4
5 class ReportConfig(AppConfig):
6     default_auto_field = 'django.db.models.BigAutoField'
7     name = 'report'
8
```

Settings:




Results:

All clear, no errors found

Forms.py

```
1 """forms for the report app"""
2 from django import forms
3 from .models import ReportUser
4
5
6 # form to report users
7 class ReportUserForm(forms.ModelForm):
8     class Meta:
9         model = ReportUser
10         fields = ('thread_url', 'user_being_reported',
11                 'reason_for_report', 'reporters_email',)
12
```

Settings:




Results:

All clear, no errors found

Models.py

```
1 """models for report app"""
2 from django.db import models
3
4
5 class Report(models.Model):
6     """
7     Stores the guidelines on reporting users
8     """
9     title = models.CharField(max_length=250)
10     updated_on = models.DateTimeField(auto_now=True)
11     content = models.TextField(blank=False)
12
13     def __str__(self):
14         return self.title
15
16
17 class ReportUser(models.Model):
18     """
19     Stores a report post on a specific user
20     """
21     thread_url = models.URLField(blank=False)
```

Settings:

Results:

All clear, no errors found

Urls.py

```
1 """urls for report app"""
2 from django.urls import path
3 from . import views
4
5
6 # url patterns for report app
7 urlpatterns = [
8     path('', views.report_guide, name='report'),
9 ]
10
```

Settings:

 ☒ 



Results:

All clear, no errors found

Views.py

```
9 # create your views here.
10 @login_required()
11 def report_guide(request):
12     """
13     Display the report page.
14
15     **Context**
16
17     ``report``
18         An instance of :model:`report.Report`
19     ``report_user``
20         The form to post reports
21     **Template:**
22
23     :template:`news/report.html`
24     """
25     report = Report.objects.all().order_by('-updated_on').first()
26     report_user = ReportUserForm()
```

Settings:

 ☒ 

Results:

All clear, no errors found