

Customise the Admin dashboard for the Post Model as shown below.

Make the necessary changes in the admin file.

Step 1

Create a project folder named as blog_project

Step 2

```
python -m venv venv
```

```
venv\Scripts\activate
```

```
pip install django
```

```
pip freeze > requirements.txt
```

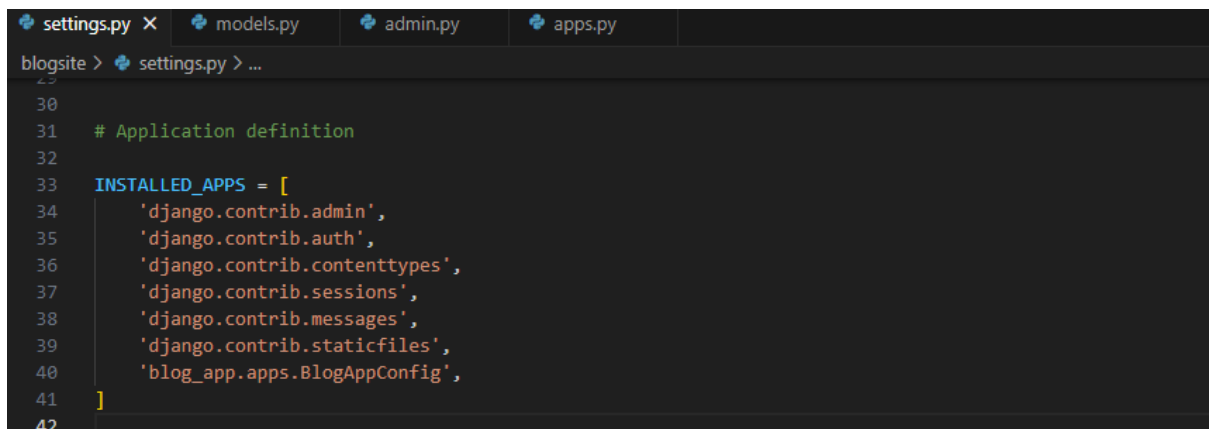
Step 3

```
django-admin startproject blogsite .
```

Step 4

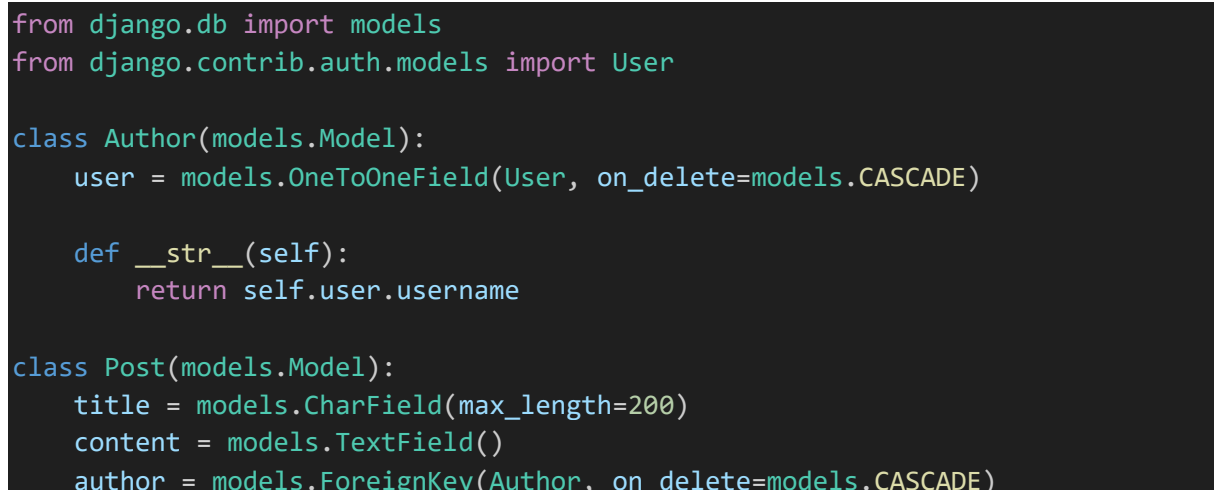
```
python manage.py startapp blog_app
```

Step 5



```
settings.py X models.py admin.py apps.py
blogsite > settings.py > ...
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'blog_app.apps.BlogAppConfig',
41 ]
42
```

Step 6



```
from django.db import models
from django.contrib.auth.models import User

class Author(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.user.username

class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
    author = models.ForeignKey(Author, on_delete=models.CASCADE)
```

```
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)

def __str__(self):
    return self.title
```

Step 7

python manage.py makemigrations

python manage.py migrate

Step 8

python manage.py createsuperuser

Step 9

```
from django.contrib import admin
from .models import Author, Post

@admin.register(Author)
class AuthorAdmin(admin.ModelAdmin):
    list_display = ('user',)

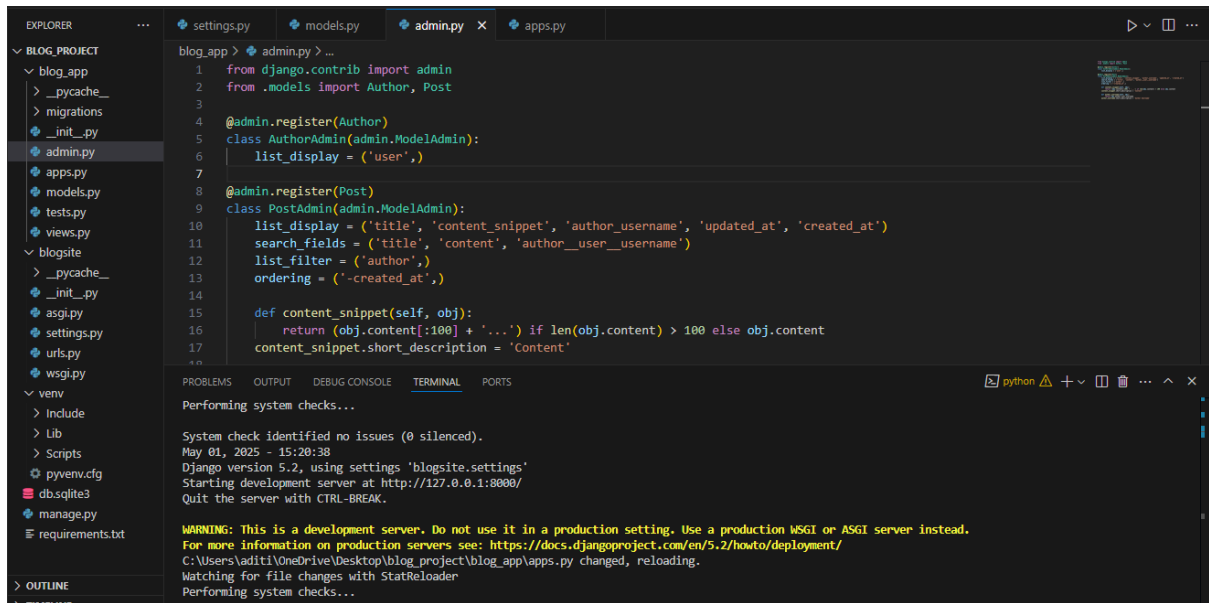
@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'content_snippet', 'author_username',
'updated_at', 'created_at')
    search_fields = ('title', 'content', 'author__user__username')
    list_filter = ('author',)
    ordering = ('-created_at',)

    def content_snippet(self, obj):
        return (obj.content[:100] + '...') if len(obj.content) > 100 else
obj.content
    content_snippet.short_description = 'Content'

    def author_username(self, obj):
        return obj.author.user.username
    author_username.short_description = 'Author Username'
```

Step 10

python manage.py runserver



The screenshot shows a code editor with the following files in the Explorer: `settings.py`, `models.py`, `admin.py`, and `apps.py`. The `admin.py` file contains the following code:

```
1 from django.contrib import admin
2 from .models import Author, Post
3
4 @admin.register(Author)
5 class AuthorAdmin(admin.ModelAdmin):
6     list_display = ('user',)
7
8 @admin.register(Post)
9 class PostAdmin(admin.ModelAdmin):
10     list_display = ('title', 'content_snippet', 'author_username', 'updated_at', 'created_at')
11     search_fields = ('title', 'content', 'author__user__username')
12     list_filter = ('author',)
13     ordering = ('-created_at',)
14
15     def content_snippet(self, obj):
16         return (obj.content[:100] + '...') if len(obj.content) > 100 else obj.content
17     content_snippet.short_description = 'Content'
```

The terminal output shows the following messages:

```
Performing system checks...
System check identified no issues (0 silenced).
May 01, 2025 - 15:20:38
Django version 5.2, using settings 'blogsite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
C:\Users\aditi\OneDrive\Desktop\blog_project\blog_app\apps.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...
```

o/p

