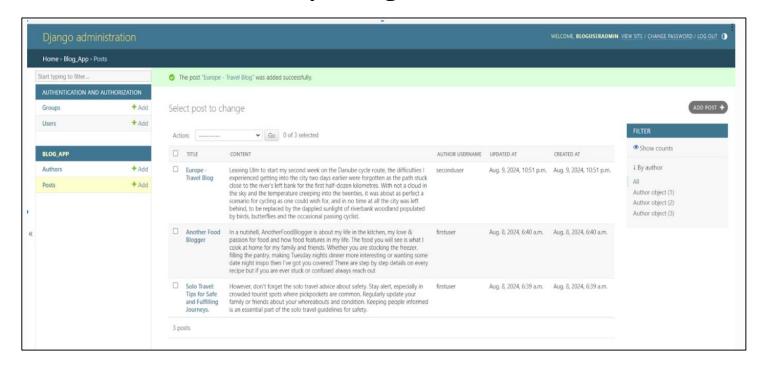
# Q. Customise the Admin dashboard for the Post Model as shown below. Make the necessary changes in the admin file.



#### **ANS:**

# models.py file:

Create a new model named "Authors" with a single field, authorName.

In the Post model, establish a relationship with the Authors model using a ForeignKey.

```
from django.db import models
from django.contrib.auth.models import User # Import the inbuilt User model

class Authors(models.Model):
    authorName=models.CharField(max_length=100)

def __str__(self):
    return self.authorName

class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
    author_username = models.ForeignKey(Authors, on_delete=models.CASCADE,null=True,blank=True)
    updated_at = models.DateTimeField(auto_now=True)
    created_at = models.DateTimeField(auto_now_add=True)
    deleted= models.BooleanField(default=False)

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

## admin.py file:

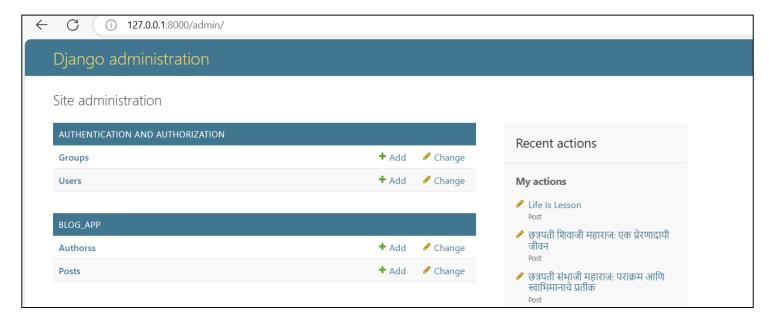
#### Then register "Authors" model in admin.py file.

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

# Register your models here.
class PostAdmin(admin.ModelAdmin):
    list_display=['title','content','author_username','updated_at','created_at','deleted']
admin.site.register(Post,PostAdmin)

class AuthorsAdmin(admin.ModelAdmin):
    list_display=['authorName']
admin.site.register(Authors,AuthorsAdmin)
```

## **OUTPUT:**



#### Next, we add AuthorName in Authors then also change in Post:

