from django.contrib import admin

from django.contrib import admin

from .models import Author, Book

@admin.register(Author)

class AuthorAdmin(admin.ModelAdmin):

list\_display = ('name', 'birthdate')

list\_filter = ('birthdate',)

search\_fields = ('name',)

date\_hierarchy = 'birthdate'

@admin.register(Book)

class BookAdmin(admin.ModelAdmin):

list\_display = ('title', 'author', 'published\_date', 'genre')

list\_filter = ('author', 'published\_date', 'genre')

search\_fields = ('title', 'author\_\_name')

date\_hierarchy = 'published\_date'

# Register your models here.

Let's ensure all the necessary steps and configurations are done correctly. I'll provide a complete set of files again, making sure every detail is covered:

### 1. Create Forms for Adding New Books and Authors

In the library app, create a new file named forms.py:

python

# library/forms.py

from django import forms

from .models import Book, Author

class AuthorForm(forms.ModelForm):

class Meta:

model = Author

fields = ['name', 'birthdate']

class BookForm(forms.ModelForm):

class Meta:

model = Book

fields = ['title', 'author', 'published\_date', 'genre']

### 2. Create Views to Handle Display and Form Submissions

Update library/views.py to include the new view:

python

# library/views.py

from django.shortcuts import render, redirect

from .models import Book, Author

from .forms import BookForm, AuthorForm

def library\_view(request):

if request.method == 'POST':

book\_form = BookForm(request.POST, prefix='book')

author\_form = AuthorForm(request.POST, prefix='author')

if book\_form.is\_valid():

book\_form.save()

return redirect('library\_view')

if author\_form.is\_valid():

author\_form.save()

return redirect('library\_view')

else:

book\_form = BookForm(prefix='book')

author\_form = AuthorForm(prefix='author')

books = Book.objects.all()

authors = Author.objects.all()

return render(request, 'library/library\_view.html', {

'books': books,

'authors': authors,

'book\_form': book\_form,

'author\_form': author\_form

})

### 3. Define URL for the Combined View

Add a URL pattern for the combined view. Update library/urls.py:

python

# library/urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.library\_view, name='library\_view'),

]

### 4. Create a Template for the Combined View

Create a template to render the combined view. Create library/templates/library/library\_view.html:

html

<!DOCTYPE html>

<html>

<head>

<title>Library</title>

</head>

<body>

<h1>Library</h1>

<h2>Books</h2>

<ul>

{% for book in books %}

<li>{{ book.title }} by {{ book.author.name }} ({{ book.published\_date }}) - Genre: {{ book.genre }}</li>

{% endfor %}

</ul>

<h2>Authors</h2>

<ul>

{% for author in authors %}

<li>{{ author.name }} ({{ author.birthdate }})</li>

{% endfor %}

</ul>

<h2>Register New Book</h2>

<form method="post">

{% csrf\_token %}

{{ book\_form.as\_p }}

<button type="submit">Register Book</button>

</form>

<h2>Register New Author</h2>

<form method="post">

{% csrf\_token %}

{{ author\_form.as\_p }}

<button type="submit">Register Author</button>

</form>

</body>

</html>

### 5. Update mylibrary/settings.py

Ensure INSTALLED\_APPS includes library:

python

# mylibrary/settings.py

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'library',

]

### 6. Update mylibrary/urls.py

Ensure that the main project URL configuration includes the library app URLs:

python

# mylibrary/urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('library.urls')),

]

### 7. Define Models

Ensure models are correctly defined in library/models.py:

python

# library/models.py

from django.db import models

class Author(models.Model):

name = models.CharField(max\_length=200)

birthdate = models.DateField()

def str(self):

return self.name

class Book(models.Model):

title = models.CharField(max\_length=200)

author = models.ForeignKey(Author, on\_delete=models.CASCADE)

published\_date = models.DateField()

genre = models.CharField(max\_length=100)

def str(self):

return self.title

### 8. Migrate and Create Superuser

Run the following commands to apply migrations and create a superuser:

bash

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser

### 9. Run the Server

Start the development server:

bash

python manage.py runserver

### 10. Access the Library Page

You can access the library page at:

- Library page: [http://127.0.0.1:8000/](http://127.0.0.1:8000/)

If you encounter any issues, please provide the specific error message or behavior you're experiencing so that I can help troubleshoot further.