(venv) PS D:\Python\Projects\django> django-admin startproject NewsPaper

(venv) PS D:\Python\Projects\django> cd NewsPaper

(venv) PS D:\Python\Projects\django\NewsPaper> python manage.py startapp news

(venv) PS D:\Python\Projects\django\NewsPaper> python manage.py startapp accounts

Migrations for 'news':

news\migrations\0001\_initial.py

- Create model Author

- Create model Category

- Create model Post

- Create model PostCategory

- Add field postCategory to post

- Create model Comment

(venv) PS D:\Python\Projects\django\NewsPaper> python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, contenttypes, news, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying auth.0010\_alter\_group\_name\_max\_length... OK

Applying auth.0011\_update\_proxy\_permissions... OK

Applying auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying news.0001\_initial... OK

Applying sessions.0001\_initial... OK

(venv) PS D:\Python\Projects\django\NewsPaper> python manage.py shell

Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from news.models import \*

>>> user1 = User.objects.create(username='Michael', first\_name = 'Tyson')

>>> Author.objects.create(authorUser=user1)

<Author: Author object (1)>

>>> user2 = User.objects.create(username='Peter', first\_name = 'Jakson')

>>> Author.objects.create(authorUser=user2)

<Author: Author object (2)>

>>> Category.objects.create(name='Politica')

<Category: Category object (1)>

>>> Category.objects.create(name='Education')

<Category: Category object (2)>

>>> Category.objects.create(name='Sport')

<Category: Category object (3)>

>>> Category.objects.create(name='Movies')

<Category: Category object (4)>

>>> Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username='Michael')), categoryType = 'NW', title = 'Title 1', text='Text 1')

<Post: Post object (1)>

>>> Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username='Michael')), categoryType = 'AR', title = 'Title 2', text='Text 2')

<Post: Post object (2)>

>>> Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username='Peter')), categoryType = 'AR', title = 'Title 3', text='Text 3')

<Post: Post object (3)>

>>> post1 = Post.objects.get(title='Title 1')

>>> post2 = Post.objects.get(title='Title 2')

>>> post3 = Post.objects.get(title='Title 3')

>>> ctg1 = Category.objects.get(name='Politica')

>>> ctg2 = Category.objects.get(name='Education')

>>> ctg3 = Category.objects.get(name='Sport')

>>> ctg4 = Category.objects.get(name='Movies')

>>> post1.postCategory.add(ctg1)

>>> post2.postCategory.add(ctg2)

>>> post3.postCategory.add(ctg3, ctg4)

>>> Comment.objects.create(commentUser=User.objects.get(username='Michael'), commentPost=Post.objects.get(title='Title 1'), text='Comment 1')

<Comment: Michael>

>>> Comment.objects.create(commentUser=User.objects.get(username='Michael'), commentPost=Post.objects.get(title='Title 2'), text='Comment 2')

<Comment: Michael>

>>> Comment.objects.create(commentUser=User.objects.get(username='Peter'), commentPost=Post.objects.get(title='Title 2'), text='Comment 3')

<Comment: Peter>

>>> Comment.objects.create(commentUser=User.objects.get(username='Peter'), commentPost=Post.objects.get(title='Title 3'), text='Comment 4')

<Comment: Peter>

>>> Post.objects.get(title='Title 1').like()

>>> Post.objects.get(title='Title 1').like()

>>> Post.objects.get(title='Title 1').like()

>>> Post.objects.get(title='Title 1').like()

>>> Post.objects.get(title='Title 1').dislike()

>>> Post.objects.get(title='Title 1').dislike()

>>> Post.objects.get(title='Title 2').like()

>>> Post.objects.get(title='Title 2').like()

>>> Post.objects.get(title='Title 2').like()

>>> Post.objects.get(title='Title 2').like()

>>> Post.objects.get(title='Title 2').like()

>>> Post.objects.get(title='Title 2').dislike()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').like()

>>> Post.objects.get(title='Title 3').dislike()

>>> Comment.objects.get(pk=1).like()

>>> Comment.objects.get(pk=1).like()

>>> Comment.objects.get(pk=1).like()

>>> Comment.objects.get(pk=1).dislike()

>>> Comment.objects.get(pk=1).dislike()

>>> Comment.objects.get(pk=2).like()

>>> Comment.objects.get(pk=2).dislike()

>>> Comment.objects.get(pk=3).like()

>>> Comment.objects.get(pk=3).like()

>>> Comment.objects.get(pk=3).like()

>>> Comment.objects.get(pk=3).like()

>>> Comment.objects.get(pk=3).like()

>>> Comment.objects.get(pk=3).dislike()

>>> Comment.objects.get(pk=3).dislike()

>>> Comment.objects.get(pk=3).dislike()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).like()

>>> Comment.objects.get(pk=4).dislike()

>>> Author.objects.get(authorUser=User.objects.get(username='Michael')).update\_rating()

>>> Author.objects.get(authorUser=User.objects.get(username='Peter')).update\_rating()

>>> Author.objects.get(authorUser=User.objects.get(username='Michael')).ratingAuthor

19

>>> Author.objects.get(authorUser=User.objects.get(username='Peter')).ratingAuthor

25

>>> bestrating = Author.objects.all().order\_by('-ratingAuthor').values('authorUser','ratingAuthor')[0]

>>> print(bestrating)

{'authorUser': 2, 'ratingAuthor': 25}

>>> a = Author.objects.order\_by('-ratingAuthor').first()

>>> print(f'Имя пользователя: {a.authorUser.username}, Рейтинг: {a.ratingAuthor}')

Имя пользователя: Peter, Рейтинг: 25

>>> bestpost = Post.objects.all().values('dateCreation','author','rating','title').order\_by('-rating')[0]

>>> print(bestpost,textpreview)

>>> print(bestpost)

{'dateCreation': datetime.datetime(2023, 9, 3, 0, 12, 3, 330473, tzinfo=datetime.timezone.utc), 'author': 2, 'rating': 6, 'title': 'Title 3', 'pk': 3}

>>> Comment.objects.filter(commentPost\_\_pk = 3).values('dateCreation','commentUser','rating','text')

<QuerySet [{'dateCreation': datetime.datetime(2023, 9, 3, 0, 23, 19, 752501, tzinfo=datetime.timezone.utc), 'commentUser': 2, 'rating': 5, 'text': 'Comment 4'}]>