Отсюда и ло следующей полосы текст в Django shell можно вставить целиком

--------------------------------------------------------------------------------

from django.contrib.auth.models import User

from main.models import \*

import datetime

user1 = User.objects.create\_user('gyron')

user2 = User.objects.create\_user('user2')

author1 = Author(rating=0, user=user1)

author1.save()

author2 = Author(rating=0, user=user2)

author2.save()

cat1 = Category(cat\_name='Новости')

cat1.save()

cat2 = Category(cat\_name='Спорт')

cat2.save()

cat3 = Category(cat\_name='Наука')

cat3.save()

cat4 = Category(cat\_name='Актуальное')

cat4.save()

post1 = Post(author=author1,

type=PostType.POST,

topic='Заголовок статьи номер1',

text='''

Методы like() и dislike() в моделях Comment и Post, которые увеличивают/уменьшают рейтинг на единицу.

Метод preview() модели Post, который возвращает начало статьи (предварительный просмотр) длиной 124 символа и добавляет многоточие в конце.

Метод update\_rating() модели Author, который обновляет рейтинг текущего автора (метод принимает в качестве аргумента только self).

Он состоит из следующего:

суммарный рейтинг каждой статьи автора умножается на 3;

суммарный рейтинг всех комментариев автора;

суммарный рейтинг всех комментариев к статьям автора.

В качестве результата задания подготовьте файл, в котором напишете список всех команд, запускаемых в Django shell.

''',

rating=0,

date=datetime.datetime.now().isoformat())

post1.save()

post2 = Post(author=author2,

type=PostType.POST,

topic='Заголовок статьи номер2',

text='''

Создать двух пользователей (с помощью метода User.objects.create\_user('username')). Создать два объекта модели Author, связанные с пользователями. Добавить 4 категории в модель Category. Добавить 2 статьи и 1 новость. Присвоить им категории (как минимум в одной статье/новости должно быть не меньше 2 категорий). Создать как минимум 4 комментария к разным объектам модели Post (в каждом объекте должен быть как минимум один комментарий). Применяя функции like() и dislike() к статьям/новостям и комментариям, скорректировать рейтинги этих объектов. Обновить рейтинги пользователей. Вывести username и рейтинг лучшего пользователя (применяя сортировку и возвращая поля первого объекта). Вывести дату добавления, username автора, рейтинг, заголовок и превью лучшей статьи, основываясь на лайках/дислайках к этой статье. Вывести все комментарии (дата, пользователь, рейтинг, текст) к этой статье.''',

rating=0,

date=datetime.datetime.now().isoformat())

post2.save()

new1 = Post(author=author2,

type=PostType.NEW,

topic='заголовок новости',

text='''

В 2023-м трудно было ждать чего-то потрясающего и инновационного от новых машин [«Ф-1»](https://www.sports.ru/f1-championship/): все-таки потолок бюджетов, не самый свободный регламент и ограничения на тесты в аэротрубах теперь не позволят рисковать с новыми концепциями и идеями каждый год. Это в 2022-м мы радовались каждому болиду – практически все отличались друг от друга не только ливреями, никто не знал точное правильное направление и рисковал в надежде угадать.

В 2023-м же после доминирования [«Ред Булл»](https://www.sports.ru/red-bull/) и стартовой скорости [«Феррари»](https://www.sports.ru/ferrari/) не сложно было предсказать, что команды прежде всего займутся адаптацией идей лидеров, а собственные надстройки пойдут уже на тестах или первой волне обновлений к пятому-шестому Гран-при. Оставался только один вопрос: куда же все-таки пойдут лидеры?

На презентации SF-23 президент всей «Феррари» Бенедетто Винья заявил: «Новая «Феррари» будет машиной с беспрецедентной скоростью» – и после такого анонса невольно всмотришься в новинку изо всех сил.''',

rating=0,

date=datetime.datetime.now().isoformat())

new1.save()

postCategory1 = PostCategory(post=post1, category=cat4)

postCategory1.save()

postCategory2 = PostCategory(post=post2, category=cat4)

postCategory2.save()

postCategory3 = PostCategory(post=post2, category=cat2)

postCategory3.save()

postCategory4 = PostCategory(post=new1, category=cat1)

postCategory4.save()

postCategory5 = PostCategory(post=new1, category=cat3)

postCategory5.save()

comment1 = Comment(post=post1, user=user1, text='comment#1', date=datetime.datetime.now().isoformat(), rating=0)

comment1.save()

comment2 = Comment(post=post2, user=user1, text='comment#2', date=datetime.datetime.now().isoformat(), rating=0)

comment2.save()

comment3 = Comment(post=new1, user=user2, text='comment#3', date=datetime.datetime.now().isoformat(), rating=0)

comment3.save()

comment4 = Comment(post=new1, user=user2, text='comment#4', date=datetime.datetime.now().isoformat(), rating=0)

comment4.save()

comment1.like()

comment1.like()

comment1.like()

comment1.like()

comment2.like()

comment2.like()

comment2.like()

comment3.like()

comment3.like()

comment4.like()

post1.like()

post1.like()

post1.like()

post1.dislike()

post2.like()

post2.like()

post2.dislike()

new1.like()

new1.dislike()

дальше вставляем в Django shell по абзацу. После последней команды дополнительно нажать enter.

bestauthor = Author.objects.order\_by('-rating')[0]

(bestauthor.user.username, bestauthor.rating)

bestpost = Post.objects.order\_by('-rating')[0]

(f'{bestpost.date.day}.{bestpost.date.month}.{bestpost.date.year}', bestpost.author.user.username, bestpost.rating, bestpost.topic, bestpost.preview())

for comment in bestpost.comment\_set.all():

print(f'{comment.date.day}.{comment.date.month}.{comment.date.year}, {comment.user.username}, {comment.rating}, {comment.text}')