1. Создать двух пользователей (с помощью метода User.objects.create\_user('username')).

user1 = User.objects.create\_user(username=’user1’, password=’password111’, email=’user1@mail.com’)

user2 = User.objects.create\_user(username=’user2’, password=’password222’, email=’user2 @mail.com’)

2. Создать два объекта модели Author, связанные с пользователями.

author1 = Author.objects.create(user=user1)

author2 = Author.objects.create(user=user2)

3. Добавить 4 категории в модель Category.

category1 = Category.objects.create(name='Sports')

category2 = Category.objects.create(name='Politics')

category3 = Category.objects.create(name='Education')

category4 = Category.objects.create(name='Technology')

4. Добавить 2 статьи и 1 новость.

post1 = Post.objects.create(author=user1, post\_type='AR', title= 'Будущее спорта', text='В этой статье обсуждаются будущие тенденции в спорте и их влияние на общество.', rating=5)

post2 = Post.objects.create(author=user2, post\_type='AR', title='Достижения в IT', text='Изучение последних достижений в области информационных технологий.', rating=7)

post3 = Post.objects.create(author=user1, post\_type='NW', title='Последние события в глобальной политике', text='Эта новостная статья освещает последние события и изменения в мировом политическом ландшафте.', rating=10,)

5. Присвоить им категории (как минимум в одной статье/новости должно быть не меньше 2 категорий).

category\_sports = Category.objects.get(name='Sports')

category\_technology = Category.objects.get(name='Technology')

category\_education = Category.objects.get(name='Education')

category\_politics= Category.objects.get(name='Politics')

post1.categories.add(category\_sports)

post2.categories.add(category\_technology, category\_education)

post3.categories.add(category\_politics)

6. Создать как минимум 4 комментария к разным объектам модели Post (в каждом объекте должен быть как минимум один комментарий).

comment1 = Comment.objects.create(post=post1, user=user1, text='Отличная статья о спорте', rating=5)

comment2 = Comment.objects.create(post=post2, user=user2, text='Содержательная дискуссия о технологиях и образовании.', rating=4)

comment3 = Comment.objects.create(post=post3, user=user1, text='Интересные события в глобальной политике', rating=3)

comment4 = Comment.objects.create(post=post4, user=user2, text='Хорошая статья', rating=2)

7. Применяя функции like() и dislike() к статьям/новостям и комментариям, скорректировать рейтинги этих объектов.

post1.like()

post2.like()

post2.dislike()

post3.like()

post4.dislike()

comment1.like()

comment2.like()

comment2.dislike()

comment3.like()

comment4.dislike()

8. Обновить рейтинги пользователей.

for author in authors:

author.update\_rating()

author.save()

9. Вывести username и рейтинг лучшего пользователя (применяя сортировку и возвращая поля первого объекта).

best\_author = Author.objects.order\_by('-rating').first()

print(f"Username: {best\_author.user.username

print(f"Рейтинг: {best\_author.rating}")

10.Вывести дату добавления, username автора, рейтинг, заголовок и превью лучшей статьи, основываясь на лайках/дислайках к этой статье.

best\_post = Post.objects.order\_by('-rating').first()

print(f"Дата добавления: {best\_post.created}")

print(f"Автор: {best\_post.author.user.username}")

print(f"Рейтинг: {best\_post.rating}")

print(f"Заголовок: {best\_post.title}")

print(f"Превью: {best\_post.preview()}")

11. Вывести все комментарии (дата, пользователь, рейтинг, текст) к этой статье.

comments = Comment.objects.filter(post=best\_post)

for comment in comments:

print(f"Дата: {comment.created}")

print(f"Пользователь: {comment.user.username}")

print(f"Рейтинг: {comment.rating}")

print(f"Текст: {comment.text}")