***D2.9. Итоговое задание***

Прежде чем начать выполнять задания, мы должны запустить виртуальное окружение с помощью команды:

**python -m venv venv**

***p.s*** При условии , что Django уже установлено, ***з***апускаем команду создания проекта:

**django-admin startproject prj**

Переходим в директорию проекта:

**cd prj/**

Создаем новое приложение:

**python manage.py startapp newapp**

Затем в файле models.py описываем модели представленные в задании

После делаем миграции, с помощью команд

**py manage.py makemigrations**

**py manage.py migrate**

Теперь заходим в виртуальное окружение Django Shell, с помощью команды:

**py manage.py shell**

И импортируем нашу модель, используя команду:

**from newapp.models import \***

**Теперь можем приступить к выполнению заданий.**

**Что вы должны сделать в консоли *Django*?**

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

user1=User.objects.create(username="Jane", first\_name="Snow")

user2=User.objects.create(username="Anna", first\_name="Milton")

* 1. ***Создать два объекта модели Author, связанные с пользователями.***

Author.objects.create(authorUser=user1)

Author.objects.create(authorUser=user2)

* 1. ***Добавить 4 категории в модель Category.***

Category.objects.create(name="IT")

Category.objects.create(name="Education")

Category.objects.create(name="Healthcare")

Category.objects.create(name="Travel")

* 1. ***Добавить 2 статьи и 1 новость.***

Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username="Jane")),categoryType="NW", title="sometitle", text="smthtext")

Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username="Jane")),categoryType="AR", title="sometitle2", text="smthtext2")

Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username="Jane")),categoryType="AR", title="sometitl3", text="smthtext3")

Post.objects.create(author=Author.objects.get(authorUser=User.objects.get(username="Anna")),categoryType="AR", title="sometitl4", text="smthtext4")

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

p1=Post.objects.get(pk=1)

p2=Post.objects.get(pk=2)

p3=Post.objects.get(pk=3)

p4=Post.objects.get(pk=4)

c1=Category.objects.get(name="IT")

c2=Category.objects.get(name=" Education ")

c3=Category.objects.get(name=" Healthcare ")

c4=Category.objects.get(name=" Travel ")

добавление связей

p1.postCategory.add(c1)

p2.postCategory.add(c1,c2)

p3.postCategory.add(c3)

p4.postCategory.add(c4)

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

Comment.objects.create(commentUser=User.objects.get(username="Jane"), commentPost=Post.objects.get(pk=1), text="comment text1")

Out[28]: <Comment: Jane>

Comment.objects.create(commentUser=User.objects.get(username="Jane"), commentPost=Post.objects.get(pk=2), text="comment text2")

Comment.objects.create(commentUser=User.objects.get(username="Anna"), commentPost=Post.objects.get(pk=3), text="comment2 text1")

Comment.objects.create(commentUser=User.objects.get(username="Anna"), commentPost=Post.objects.get(pk=1), text="comment2 text2")

Comment.objects.create(commentUser=User.objects.get(username="Anna"), commentPost=Post.objects.get(pk=4), text="comment text1")

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

Post.objects.get(pk=1).like()

Post.objects.get(pk=1).like()

Post.objects.get(pk=2).dislike()

Post.objects.get(pk=2).dislike()

Post.objects.get(pk=3).like()

Post.objects.get(pk=3).like()

Post.objects.get(pk=3).like()

Post.objects.get(pk=4).dislike()

Post.objects.get(pk=4).dislike()

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

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

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

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

Если хотим вывести поле рейтинга для статей и новости

Post.objects.get(pk=1).rating

Post.objects.get(pk=2).rating

Post.objects.get(pk=3).rating

Post.objects.get(pk=4).rating

Если хотим вывести поле рейтинга для комментариев

Comment.objects.get(pk=1).rating

Comment.objects.get(pk=2).rating

Comment.objects.get(pk=3).rating

Comment.objects.get(pk=4).rating

* 1. ***Обновить рейтинги пользователей.***

Author.objects.get(authorUser=User.objects.get(username="Jane")).update\_rating()

Author.objects.get(authorUser=User.objects.get(username="Anna")).update\_rating()

Если хотим вывести поле рейтинга для пользователя

a = Author.objects.get(authorUser=User.objects.get(username="Jane"))

In [55]: a.ratingAuthor

Out[55]: 9

Author.objects.get(authorUser=User.objects.get(username="Anna")).ratingAuthor

Out[56]: -5

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

best = Author.objects.all().order\_by("-ratingAuthor").values("authorUser", "ratingAuthor")[0]

print(best)

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

Рассмотрела 2 варианта:

1)post=Post.objects.get(id=1)

post.dataCreation.strftime("%Y-%m-%d %H:%M")

post.author

post.rating

post.title

post=Post.objects.get(id=2)

post.dataCreation.strftime("%Y-%m-%d %H:%M")

post.author

post.rating

post.title

2)Post.objects.all().order\_by("-rating").values("dataCreation", "author", "rating", "title")[::-1]

Вывод даты добавления стать пользователями:

post=Post.objects.get(id=1)

post.dataCreation.strftime("%Y-%m-%d %H:%M")

post=Post.objects.get(id=2)

post.dataCreation.strftime("%Y-%m-%d %H:%M")

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

Рассмотрела 2 варианта вывода:

1)Post.objects.order\_by("-rating")[0].comment\_set.all().values("dateCreation", "commentUser", "id", "rating", "text")

2)Comment.objects.all().order\_by("-rating").values("dateCreation", "commentUser\_id", "id", "rating", "text")[0].

Вывод даты добавления комментария пользователями:

comment=Comment.objects.get(id=1)

comment.dateCreation.strftime("%Y-%m-%d %H:%M")

comment=Comment.objects.get(id=2)

comment.dateCreation.strftime("%Y-%m-%d %H:%M")