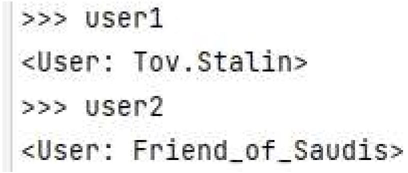
Preparatory Steps:

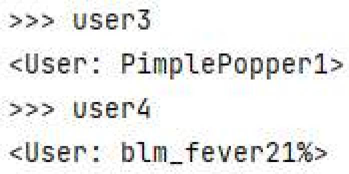
1. Start the session in shell
   - *python manage.py shell*
2. Import models
   - *from fake_news_app.models import **

Execution:

1. Create new users:
   - *user1 = User.objects.create_user(username = 'Tov.Stalin')*
   - *user2 = User.objects.create_user(username = 'Friend_of_Saudis')*
   - *user3 = User.objects.create_user(username = 'PimplePopper1')*
   - *user4 = User.objects.create_user(username = 'blm_fever21%')*

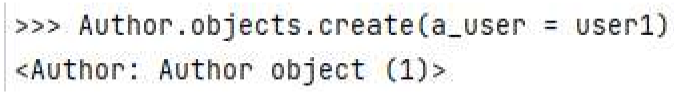


```
>>> user1
<User: Tov.Stalin>
>>> user2
<User: Friend_of_Saudis>
```



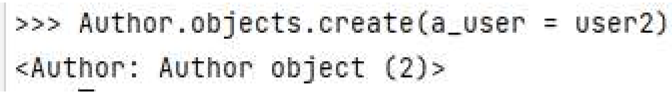
```
>>> user3
<User: PimplePopper1>
>>> user4
<User: blm_fever21%>
```

2. Add objects to Author
   - *Author.objects.create(a_user = user1)*



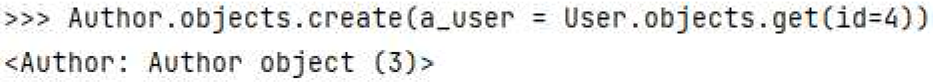
```
>>> Author.objects.create(a_user = user1)
<Author: Author object (1)>
```

   - *Author.objects.create(a_user = user2)*



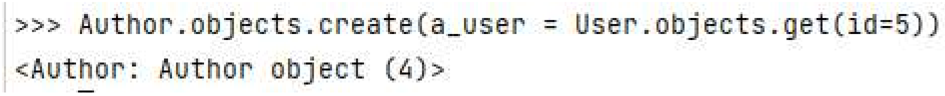
```
>>> Author.objects.create(a_user = user2)
<Author: Author object (2)>
```

   - Another code option: *Author.objects.create(a_user = User.objects.get(id=4))*



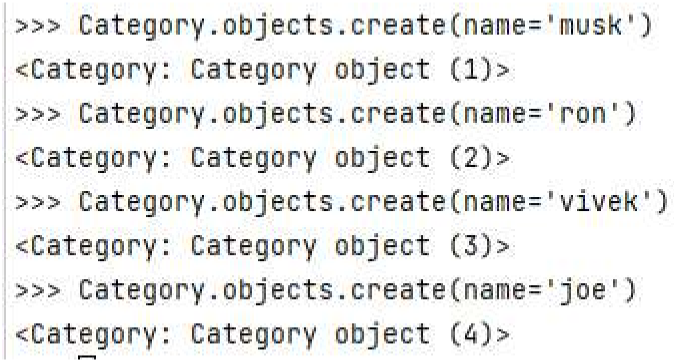
```
>>> Author.objects.create(a_user = User.objects.get(id=4))
<Author: Author object (3)>
```

   - Another code option: *Author.objects.create(a_user = User.objects.get(id=5))*



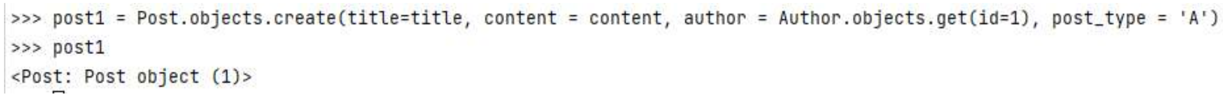
```
>>> Author.objects.create(a_user = User.objects.get(id=5))
<Author: Author object (4)>
```

3. Create 4 objects in Category
    - *Category.objects.create(name='musk')*
    - *Category.objects.create(name='ron')*
    - *Category.objects.create(name='vivek')*
    - *Category.objects.create(name='joe')*



```
>>> Category.objects.create(name='musk')
<Category: Category object (1)>
>>> Category.objects.create(name='ron')
<Category: Category object (2)>
>>> Category.objects.create(name='vivek')
<Category: Category object (3)>
>>> Category.objects.create(name='joe')
<Category: Category object (4)>
```

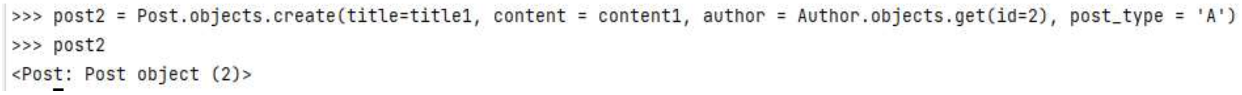
4. Add 2 articles and 1 news post:
    a. Article 1
    - *content = "Elon Musk once bought a frozen yogurt machine for his office as a reward for one of his engineers who met an ambitious deadline. He also shared his love for frozen yogurt with Vivek Ramaswamy, a presidential candidate whom he admires and supports. The two bonded over their favorite flavors and toppings during a fundraiser event."*
    - *title = "Elon's Yogurt"*
    - *post1 = Post.objects.create(title=title, content = content, author = Author.objects.get(id=1), post_type = 'A')*



```
>>> post1 = Post.objects.create(title=title, content = content, author = Author.objects.get(id=1), post_type = 'A')
>>> post1
<Post: Post object (1)>
```

b. Article 2

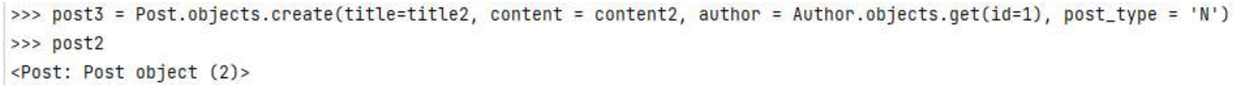
- *content1 =   context_biden= "Biden and Harris raised $97 million in the last quarter of 2023, bringing their total campaign funds to $117 million. They attributed their success to online fundraising and a loyal coalition of supporters. They also increased their email list by 15%, hoping to convert more potential donors. Trump, their likely opponent, had not disclosed his fundraising numbers, but he had raised more money than Biden in the same period of 2019."*
- *title1 = "Biden's Campaign"*
- *post2 = Post.objects.create(title=title1, content = content1, author = Author.objects.get(id=2), post_type = 'A')*



```
>>> post2 = Post.objects.create(title=title1, content = content1, author = Author.objects.get(id=2), post_type = 'A')
>>> post2
<Post: Post object (2)>
```

c. News post

- *title2 = " How to Break Up with Your Girlfriend Using Only Emojis"*
- *content2 = " The text is about the rivalry between Florida Gov. Ron DeSantis and former President Donald Trump for the 2024 Republican presidential nomination. DeSantis claimed that he could beat Biden in key states, while Trump could not. However, a recent poll showed that Trump had a huge lead over DeSantis and was seen as more electable by Republican voters. DeSantis also criticized Trump's lack of discipline and personnel and faced challenges in his own campaign. "*
- *post3 = Post.objects.create(title=title2, content = content2, author = Author.objects.get(id=1), post_type = 'N')*



```
>>> post3 = Post.objects.create(title=title2, content = content2, author = Author.objects.get(id=1), post_type = 'N')
>>> post2
<Post: Post object (2)>
```

5. Assign them categories (at least one article/news item should have no less than 2 categories).
    a. Article
        i. *post1.post_category.add(Category.objects.get(id=1))*
    b. Article
        i. *post2.post_category.add(Category.objects.get(id=4))*
    c. News Post
        i. *post3.post_category.add(Category.objects.get(id=2), Category.objects.get(id=3))*

6. Create at least 4 comments for different Post model objects (each object should have at least one comment).

    a. *post_1 = Post.objects.get(id=1)*

        - *comment_1 = Comment.objects.create(post_comment=post_1, user_comment=User.objects.get(id=2), text="This painting is a disgrace to the art world and an insult to the human eye. It looks like a toddler threw up on a canvas and called it a masterpiece.")*

    b. *post_2 = Post.objects.get(id=2)*

        - *comment_2 = Comment.objects.create(post_comment=post_2, user_comment= User.objects.get(id=4), text="This book is a pile of garbage and a waste of paper. It has the plot of a soap opera and the characters of a cardboard box.")*

    c. *post_3 = Post.objects.get(id=3)*

        - *comment_3 = Comment.objects.create(post_comment=post_3, user_comment= User.objects.get(id=5),text="Your music is a torture to the ears and a crime to the soul. It sounds like a cat being strangled by a chainsaw and a dog being run over by a train.")*
        - *comment_4 = Comment.objects.create(post_comment=post_3, user_comment= User.objects.get(id=3),text=" Your performance was the worst thing I have ever witnessed in my life. You have the charisma of a wet sock and the talent of a dead fish.")*

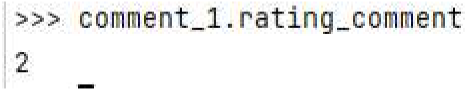
7. Using the like() and dislike() functions on articles/news and comments, adjust the ratings of these objects.

- *comment_1 = Comment.objects.get(id=1)*
- *comment_2 = Comment.objects.get(id=2)*
- *comment_3 = Comment.objects.get(id=3)*
- *comment_4 = Comment.objects.get(id=4)*

- *user_2 = User.objects.get(id=2)*
- *user_4 = User.objects.get(id=4)*
- *user_5 = User.objects.get(id=5)*
- *user_3 = User.objects.get(id=3)*

a. Add 5 likes and 3 dislikes to comment id=1:

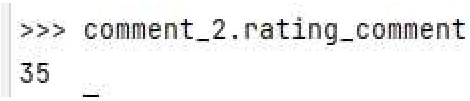
```
for i in range(5):
    comment_1.like()
    comment_1.save()

for i in range(3):
    comment_1.dislike()
    comment_1.save()
```



```
>>> comment_1.rating_comment
2
    _
```

b. Add 78 likes and 43 dislikes to comment id=2

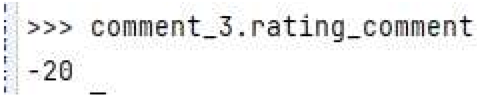
```
for i in range(78):
    comment_2.like()
    comment_2.save()

for i in range(43):
    comment_2.dislike()
    comment_2.save()
```



```
>>> comment_2.rating_comment
35  _
```

c. Add 1 like and 21 dislikes to comment id=3

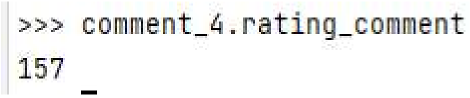
```
for i in range(21):
    comment_3.dislike()
    comment_3.save()
for i in [1]:
    comment_3.like()
    comment_3.save()
```



```
>>> comment_3.rating_comment
-20 _
```

d. Add 178 likes and 21 dislikes to comment id=4

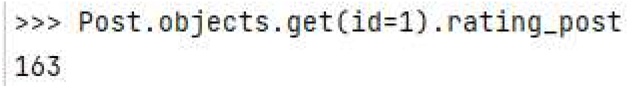
```
for i in range(178):
    comment_4.like()
    comment_4.save()
for i in range(21):
    comment_4.dislike()
    comment_4.save()
```



```
>>> comment_4.rating_comment
157 _
```

e. Add to post id=1 400 likes and 237 dislikes

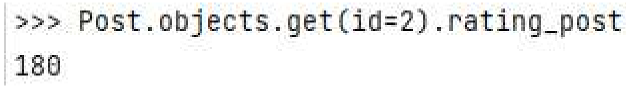
```
for i in range(400):

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

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

for i in range(237):

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

    Post.objects.get(id=1).save()
```



```
>>> Post.objects.get(id=1).rating_post
163
```

f. Add to post id=2 267 likes and 96 dislikes

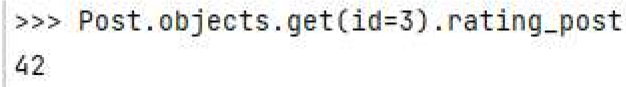
```
for i in range(276):
     Post.objects.get(id=2).like()
     Post.objects.get(id=2).save()
for i in range(96):
    Post.objects.get(id=2).dislike()
    Post.objects.get(id=2).save()
```



```
>>> Post.objects.get(id=2).rating_post
180
```

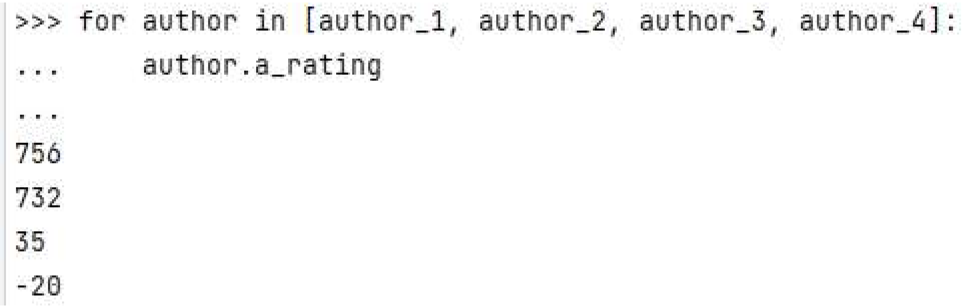
g. Add to post id_3 68 likes and 26 dislikes

```
for i in range(68):
    Post.objects.get(id=3).like()
    Post.objects.get(id=3).save()
for i in range(26):
    Post.objects.get(id=3).dislike()
    Post.objects.get(id=3).save()
```



```
>>> Post.objects.get(id=3).rating_post
42
```

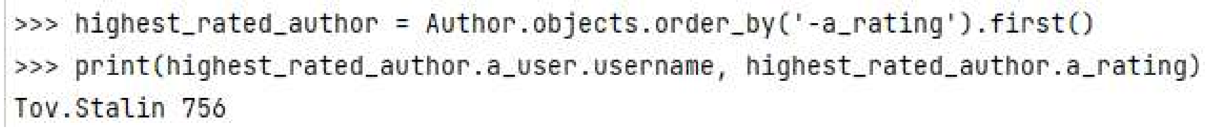
8. Update the author's ratings
   - author_1 = Author.objects.get(id=1)
   - author_1.update_rating()
   - author_2 = Author.objects.get(id=2)
   - author_2.update_rating()
   - author_3 = Author.objects.get(id=3)
   - author_3.update_rating()
   - author_4 = Author.objects.get(id=4)
   - author_4.update_rating()



```
>>> for author in [author_1, author_2, author_3, author_4]:
...     author.a_rating
...
756
732
35
-20
```

9. Display the username and rating of the best user (applying sorting and returning the fields of the first object).

   - highest_rated_author = Author.objects.order_by('-a_rating').first()
   - print(highest_rated_author.a_user.username, highest_rated_author.a_rating)



```
>>> highest_rated_author = Author.objects.order_by('-a_rating').first()
>>> print(highest_rated_author.a_user.username, highest_rated_author.a_rating)
Tov.Stalin 756
```

10. Display the date added, author’s username, rating, title, and preview of the best article, based on the likes/dislikes of this article.

- best_article = Post.objects.order_by('-rating_post').first()
- author_username = best_article.author.a_user.username
- date_added = best_article.created_at
- rating = best_article.rating_post
- title = best_article.title
- preview = best_article.preview()
- print(f"""

  Date Added: {date_added}

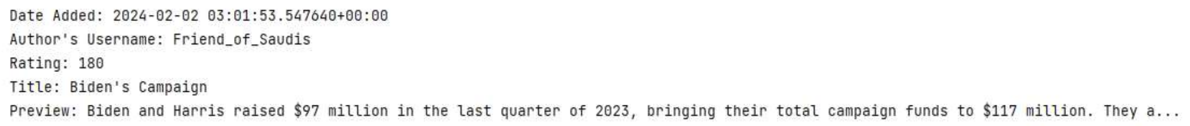
  Author's Username: {author_username}

  Rating: {rating}

  Title: {title}

  Preview: {preview}

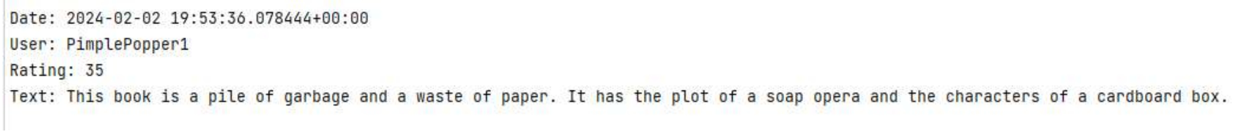
  """)



```
Date Added: 2024-02-02 03:01:53.547640+00:00
Author's Username: Friend_of_Saudis
Rating: 180
Title: Biden's Campaign
Preview: Biden and Harris raised $97 million in the last quarter of 2023, bringing their total campaign funds to $117 million. They a...
```

11. Display all comments (date, user, rating, text) for this article.

```
- comments = Comment.objects.filter(post_comment=best_article)
- for comment in comments:
          date = comment.created_at
          user = comment.user_comment.username
          rating = comment.rating_comment
          text = comment.text
          print(f"Date: {date}\nUser: {user}\nRating: {rating}\nText: {text}\n")
```



```
Date: 2024-02-02 19:53:36.078444+00:00
User: PimplePopper1
Rating: 35
Text: This book is a pile of garbage and a waste of paper. It has the plot of a soap opera and the characters of a cardboard box.
```