PS C:\Users\USER\OneDrive\Masaüstü\Yazılım\Python Web Geliştirme\BlogProjesi2> python manage.py shell

7 objects imported automatically (use -v 2 for details).

Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32

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

(InteractiveConsole)

>>> from django.contrib.auth.models import User

>>> from article.models import Article

>>> User

<class 'django.contrib.auth.models.User'>

>>> Art

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Art' is not defined

>>> Article

<class 'article.models.Article'>

>>> newUser = User(username = "denemekullanici",password="123")

>>> newUser

<User: denemekullanici>

>>> newUser.save()

>>> newUser2 = User(username = "denemekullanici2")

>>> newUser2.set\_password("123")

>>> newUser2.save()

>>> newUser3 = User()

>>> newUser3.username = "denemekullanici3"

>>> newUser3.set\_password("123")

>>> newUser3.first\_name = "Mustafa"

>>> newUser3.save()

>>> article = Article(title = "Django Shell Deneme",content = "İçerik içerik",auhtor=newUser3)

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Python313\Lib\site-packages\django\db\models\base.py", line 569, in \_\_init\_\_

raise TypeError(

...<2 lines>...

)

TypeError: Article() got unexpected keyword arguments: 'auhtor'

>>> article = Article(title = "Django Shell Deneme",content = "İçerik içerik",author=newUser3)

>>> article.save()

>>> article2 = Article()

>>> article2.title = "Deneme5"

>>> article2.content = "İçerik"

>>> article2.author = "newUser3

File "<console>", line 1

article2.author = "newUser3

^

SyntaxError: unterminated string literal (detected at line 1)

>>> article2.author = "newUser3"

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Python313\Lib\site-packages\django\db\models\fields\related\_descriptors.py", line 291, in \_\_set\_\_

raise ValueError(

...<7 lines>...

)

ValueError: Cannot assign "'newUser3'": "Article.author" must be a "User" instance.

>>> article2.author = newUser3

>>> article2.save()

>>> Article.objects.create(title="deneme20",content="20",author=newUser2)

<Article: deneme20>

>>> article.title

'Django Shell Deneme'

>>> article = Article.objects.create(title="deneme20",content="20",author=newUser2)

>>> article.title

'deneme20'

>>> article.title = "Deneme 20 değişti"

>>> article.save()

>>> Article.objects.all()

<QuerySet [<Article: deneme1>, <Article: naber>, <Article: Django Shell Deneme>, <Article: Deneme5>, <Article: deneme20>, <Article: Deneme 20 değişti>]>

>>> Article.objects.get(title = "Naber2")

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Python313\Lib\site-packages\django\db\models\manager.py", line 87, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^

File "C:\Python313\Lib\site-packages\django\db\models\query.py", line 633, in get

raise self.model.DoesNotExist(

"%s matching query does not exist." % self.model.\_meta.object\_name

)

article.models.Article.DoesNotExist: Article matching query does not exist.

>>> Article.objects.get(title = "naber")

<Article: naber>

>>> article = Article.objects.get(title = "naber")

>>> article.title

'naber'

>>> article.author

<User: hasan1>

>>> article.delete()

(1, {'article.Article': 1})

>>> Article.get.all()

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: type object 'Article' has no attribute 'get'

>>> Article.objects.all()

<QuerySet [<Article: deneme1>, <Article: Django Shell Deneme>, <Article: Deneme5>, <Article: deneme20>, <Article: Deneme 20 değişti>]>

>>> article = Article.objects.get(id = 5)

>>> article.title

'Deneme5'

>>> article.delete()

(1, {'article.Article': 1})

>>> Article.objects.all()

<QuerySet [<Article: deneme1>, <Article: Django Shell Deneme>, <Article: deneme20>, <Article: Deneme 20 değişti>]>

>>> Article.objects.filter(title\_\_contains = "den")

<QuerySet [<Article: deneme1>, <Article: Django Shell Deneme>, <Article: deneme20>, <Article: Deneme 20 değişti>]>

>>>