

رادمان محمدی

**Radman Mohammadi**

پارت سوم پروژه فاینال جنگو:

**Part3**

استاد:

عرفان اسماعیلی شیرازی

# User مدل

۱- کل کاربران سایت:

```
In [3]: User.objects.all()
Out[3]: <QuerySet [<User: amin>, <User: zahra>, <User: admin>, <User: karim>, <User: ahmadreza>, <User: sara>]>
```

۲- کل ابرکاربران (superuser\_is) سایت:

```
>>>
>>> superusers = User.objects.filter(is_superuser=True)
>>> superusers
<QuerySet [<User: admin>]>
>>> _
```

۳- کل کاربران فعال active\_is سایت

```
>>>
>>> actives = User.objects.filter(is_active=True)
>>> actives
<QuerySet [<User: admin>, <User: ali>, <User: ahmad>, <User: zahra>, <User: mohammadhossein>, <User: mahmoud>, <User: maryam>]>
>>>
```

۴- کل کاربران سایت به جز کاربرانی که ابرکاربر هستند:

```
>>>
>>> User.objects.differ(is_superuser=True)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: 'UserManager' object has no attribute 'differ'
>>> User.objects.exclude(is_superuser=True)
<QuerySet [<User: ali>, <User: ahmad>, <User: zahra>, <User: mohammadhossein>, <User: mahmoud>, <User: maryam>]>
>>> _
```

۵- کاربری که نام کاربری آن دقیقاً admin باشد.

```
>>>
>>> User.objects.filter(username="admin")
<QuerySet [ <User: admin>]>
>>> _
```

۶- چک کنید کاربری با نام کاربری reza آیا وجود دارد در دیتابیس یا خیر. (پاسخ باید True یا False باشد)

```
>>>
>>> User.objects.filter(username="reza").exists()
False
>>> _
```

۷- کاربرانی که که نام کوچک آنها با نام خانوادگیشان دقیقاً یکسان باشد.

```
>>>
>>> from django.db.models import F
>>>
>>>
>>>
>>> User.objects.filter(first_name=F("last_name"))
<QuerySet [ <User: admin>, <User: ali>]>
>>> _
```

۸- کاربرانی که ابرکاربر (is\_superuser) و کارمند (is\_staff) باشند.

```
>>>
>>> from django.db.models import Q
>>>
>>>
>>> User.objects.filter(Q(is_superuser=True) & Q(is_staff=True))
<QuerySet [ <User: admin>]>
>>> _
```

۹- کاربرانی که ابرکاربر (is\_superuser) یا کارمند (is\_staff) باشند.

```
>>>
>>> from django.db.models import Q
>>>
>>> User.objects.filter(Q(is_superuser=True)|Q(is_staff=True))
<QuerySet [ <User: admin> ]>
>>> _
```

۱۰- کاربرانی که حرف a در نام کوچک آنها وجود داشته باشد.

```
>>>
>>> User.objects.filter(last_name__contains="a")
<QuerySet [ <User: ahmad>, <User: zahra> ]>
>>>
>>>
```

۱۱- کاربرانی که کلمه test در نام کاربری آنها وجود داشته باشد

```
>>>
>>> User.objects.filter(username__contains="test")
<QuerySet [ <User: davidtest> ]>
>>>
>>>
```

۱۲- کاربرانی که نام کاربری آنها با کلمه @edu تمام شود. مثال, edu@Mohammad, edu@Reza, edu@E

```
>>>
>>> User.objects.filter(username__endswith="@edu")
<QuerySet [ <User: mohammadhossein@edu>, <User: mahmoud@edu>, <User: maryam@edu> ]>
>>>
```

۱۳- تعداد کل کاربران سایت:

```
>>>
>>> User.objects.count()
8
>>> _
```

۱۴- تعداد کل ابرکاربران superuser\_is سایت:

```
>>>
>>> User.objects.filter(is_superuser=True).count()
1
>>>
```

۱۵- اولین کاربر سایت:

```
>>> User.objects.first()
<User: admin>
>>>
```

۱۶- آخرین کاربر سایت:

```
>>>
>>> User.objects.last()
<User: davidtest>
>>> _
```

۱۷- کل کاربران سایت ( مرتب سازی شده بر مبنای نام خانوادگی بر اساس حروف الفبا ) :

```
>>>
>>>
>>> User.objects.order_by("last_name")
<QuerySet [<User: admin>, <User: ali>, <User: mahmoud@edu>, <User: maryam@edu>, <User: mohammadhossein@edu>, <User: zahra>, <User: ahmad>, <User: davidtest>]>
>>> _
```

## مدل Profile

۱- کل پروفایل‌های سایت:

```
>>>
>>> Profile.objects.all()
<QuerySet [<Profile: ali>, <Profile: ahmad>, <Profile: zahra>, <Profile: mohammadhossein@edu>, <Profile: mahmoud@edu>, <Profile: maryam@edu>, <Profile: davidtest>]>
>>> _
```

۲- کل پروفایل‌های سایت ( مرتب سازی شده بر اساس تاریخ عضویت به صورت نزولی):

```
>>>
>>> Profile.objects.order_by("membership_at")
<QuerySet [<Profile: ali>, <Profile: ahmad>, <Profile: zahra>, <Profile: mohammadhossein@edu>, <Profile: mahmoud@edu>, <Profile: maryam@edu>, <Profile: davidtest>]>
>>> _
```

۳- تعداد کل پروفایل‌هایی که نویسنده هستند.

```
>>>
>>> Profile.objects.filter(is_author=True).count()
4
>>> _
```

۴- پروفایل‌هایی که کلمه @edu در انتهای نام کاربری آنها وجود داشته باشد.

```
>>>
>>> User.objects.filter(username__contains="@edu")
<QuerySet [<User: mohammadhossein@edu>, <User: mahmoud@edu>, <User: maryam@edu>]>
>>>
>>>
```

۵- پروفایل‌هایی که ابرکاربر باشند.

۶- پروفایلهایی که جنسیت آنها آقا باشد.

```
>>>
>>> Profile.objects.filter(gender="m")
<QuerySet [<Profile: ali>, <Profile: ahmad>, <Profile: mohammadhossein@edu>, <Profile: mahmoud@edu>, <Profile: davidtest>]>
>>>
```

۷- کاربرانی که سن آنها بالای ۱۸ باشد. ( سن ۱۸ نیز قابل قبول باشد)

```
>>>
>>> Profile.objects.filter(age__gte=18)
<QuerySet [<Profile: ali>, <Profile: ahmad>, <Profile: zahra>, <Profile: mohammadhossein@edu>, <Profile: mahmoud@edu>, <Profile: maryam@edu>, <Profile: davidtest>]>
>>> _
```

۸- کاربرانی که سن آنها بین ۱۰ تا ۱۸ باشد. ( سن ۱۰ و سن ۱۸ قابل قبول نمی باشد)

```
>>>
>>> Profile.objects.filter(age__gt=10, age__lt=18)
<QuerySet [<Profile: ahmad>]>
>>> _
```

۹- کاربرانی که سن آنها بین ۲۰ تا ۳۰ باشد. ( سن ۲۰ و سن ۳۰ نیز قابل قبول باشد)

```
>>>
>>> Profile.objects.filter(age__gte=20, age__lte=30)
<QuerySet [<Profile: ali>, <Profile: zahra>]>
>>>
```

۱۰- کاربرانی که سن آنها کمتر از ۵۰ باشد. ( سن ۵۰ قابل قبول نمی باشد)

```
>>>
>>> Profile.objects.filter(age__lt=50)
<QuerySet [<Profile: ali>, <Profile: ahmad>, <Profile: zahra>, <Profile: mohammadhossein@edu>, <Profile: mahmoud@edu>, <Profile: maryam@edu>, <Profile: davidtest>]>
>>> _
```

۱۱- مجموعه دیکشنریها از کل پروفایلها که هر پروفایل شامل ویژگی های کد ملی و جنسیت باشد.

```
>>>
>>> Profile.objects.values("nc", "gender")
<QuerySet [{ 'nc': '9436475834', 'gender': 'm'}, { 'nc': '3452647589', 'gender': 'm'}, { 'nc': '2435762345', 'gender': 'f'}, { 'nc': '6251435496', 'gender': 'm'}, { 'nc': '4523956172', 'gender': 'm'}, { 'nc': '4736102845', 'gender': 'f'}, { 'nc': '3425672489', 'gender': 'm'}]>
>>>
```

۱۲- مجموعه تاپلها از کل پروفایلها که هر پروفایل شامل ویژگیهای کدملی و جنسیت باشد.

```
>>>
>>> Profile.objects.values_list("nc", "gender")
<QuerySet [( '9436475834', 'm'), ( '3452647589', 'm'), ( '2435762345', 'f'), ( '6251435496', 'm'), ( '4523956172', 'm'), ( '4736102845', 'f'), ( '3425672489', 'm')]>
>>> _
```

۱۳- یک لیست از شماره موبایل تمامی پروفایلها (نیاز به سایر ویژگی های پروفایل ها نداریم)

```
>>>
>>> Profile.objects.values('phone_number')
<QuerySet [{ 'phone_number': '09156432547'}, { 'phone_number': '09156432564'}, { 'phone_number': '09154323456'}, { 'phone_number': '09156342745'}, { 'phone_number': '09158234912'}, { 'phone_number': '09153242933'}, { 'phone_number': '09152348945'}]>
>>>
```

۱۴- تعداد کل پستهای پروفایل با نام کاربری admin (فرض براینکه این کاربر نویسنده باشد)



## مدل Post

۱- کل پستهای سایت:

```
>>>
>>> Post.objects.all()
<QuerySet [<Post: Omicron>, <Post: Corona>, <Post: Crypto Currency>, <Post: Asus Micro Computers>,
<Post: Electric Trucks>, <Post: Programming>]>
>>>
```

۲- کل پستهای سایت ( مرتب سازی شده بر اساس جدیدترین):

```
>>>
>>> Post.objects.order_by("-created_at")
<QuerySet [<Post: Programming>, <Post: Electric Trucks>, <Post: Asus Micro Computers>, <Post: Crypto Currency>, <Post: Corona>, <Post: Omicron>]>
>>>
```

۳- کل پستهایی که وضعیت آنها در حال انتشار است. ( مرتب سازی شده بر اساس جدیدترین):

```
>>>
>>> Post.objects.filter(status=2).order_by("-created_at")
<QuerySet [<Post: Programming>, <Post: Asus Micro Computers>, <Post: Omicron>]>
>>> _
```

۴- کل پست هایی که وضعیت آنها در حال انتشار و دسته بندی آنها علمی باشد ( مرتب سازی شده بر اساس جدیدترین):

```
>>>
>>> Post.objects.filter(Q(status=2) & Q(category=1)).order_by("-created_at")
<QuerySet [<Post: Asus Micro Computers>]>
>>> _
```

۵- کل پست هایی که دسته بندی آنها علمی یا آموزشی باشد:

```
>>> Post.objects.filter(Q(category=2) | Q(category=1))
<QuerySet [ <Post: Crypto Currency>, <Post: Asus Micro Computers>, <Post: Electric Trucks>, <Post: Programming>]>
>>>
```

۶- کل پست هایی که دسته بندی آنها علمی یا آموزشی باشد و وضعیت آنها در حال انتشار باشد:

```
>>>
>>> Post.objects.filter(Q(category=2) | Q(category=1) & Q(status=2))
<QuerySet [ <Post: Asus Micro Computers>, <Post: Programming>]>
>>> _
```

۷- کل پست هایی که کلمه ی important در توضیحات کامل آن وجود داشته باشد:

```
>>>
>>> Post.objects.filter(long_description__contains="important")
<QuerySet [ <Post: Crypto Currency>, <Post: Electric Trucks>]>
>>>
```

۸- کل پست هایی که شماره موبایل نویسنده آن ۰۹۱۹۰۰۰۱۱۱۱ باشد:

```
In [62]: profile = Profile.objects.get(phone_number='09190001111')
In [63]: profile.post.all()
Out[63]: <QuerySet [ <Post: Electric Trucks>]>
In [64]: _
```

۹- کل پست هایی که نام کاربری نویسنده آن ali باشد:

```
In [70]: profile = Profile.objects.get(first_name="ali")
In [71]: profile.post.all()
Out[71]: <QuerySet [ <Post: Omicron>]>
In [72]: _
```

۱۰- کل پستهایی که توسط ابرکاربران گذاشته شده باشد:

۱۱- کل پستها به جز پستهایی که خبری باشد:

```
In [15]: Post.objects.exclude(category="0")
Out[15]: <QuerySet [<Post: Crypto Currency>, <Post: Asus Micro Computers>, <Post: Electric Trucks>, <Post: Programming>]>
In [16]: _
```

۱۲- یک لیست از عنوان تمامی پستها ( نیاز به سایر ویژگی پست ها نمی باشد):

```
In [30]: Post.objects.values('title')
Out[30]: <QuerySet [{ 'title': 'Omicron'}, { 'title': 'Corona'}, { 'title': 'Crypto Currency'}, { 'title': 'Asus Micro Computers'}, { 'title': 'Electric Trucks'}, { 'title': 'Programming'}]>
In [31]:
```

۱۳- مجموعه دیکشنریها از فیلدهای id و title پستها ( نیاز به سایر ویژگی پست ها نمی باشد):

```
In [51]: dict_result = {Post.objects.values('id', 'title')}
In [52]: dict_result
Out[52]: {<QuerySet [{ 'id': 1, 'title': 'Omicron'}, { 'id': 2, 'title': 'Corona'}, { 'id': 3, 'title': 'Crypto Currency'}, { 'id': 4, 'title': 'Asus Micro Computers'}, { 'id': 5, 'title': 'Electric Trucks'}, { 'id': 6, 'title': 'Programming'}]>}
```

۱۴- کل کامنتهای پست با آیدی ۵:

```
In [77]: post = Post.objects.get(id=5)
In [78]: post.comment.all()
Out[78]: <QuerySet [<Comment: oh ithey are important cars. (-) Electric Trucks>, <Comment: it is amazing. (-) Electric Trucks>]>
In [79]:
```

۱۵- تعداد کامنت های پست با عنوان hello world :

## مدل Comment

۱- کل کامنتهای سایت:

```
In [4]: from blog.models import Comment
In [5]: Comment.objects.all()
Out[5]: <QuerySet [(<Comment: I brought a lot of information by this subject. (-) Transportaion>, <Comment: The content was excellent. (-) Drug>, <Comment: It was bad and I did not like it (-) Drug>, <Comment: Better content should be posted. (-) Range Rover Company>, <Comment: It was great (-) Range Rover Company>, <Comment: It's so scary. (-) Covid>, <Comment: The corona will probably be remove d soon (-) Covid>, <Comment: Range Rover is the best car manufacturer in the world (-) Range Rover Company>, <Comment: Transportation is very practical (-) Transportaion>]>
```

۲- کل کامنتهای پستی که id آن ۲ است. ( مرتب سازی شده بر اساس جدیدترین):

```
In [21]: post = Post.objects.get(id=2)
In [22]: post.comment.order_by("-created_at")
Out[22]: <QuerySet [(<Comment: i thought the the covid has left from the Earth about 2 mounth (-) orona>)]>
```

۳- کل کامنتهای پروفایل با نام کاربری Erfan :

```
In [34]: profile = Profile.objects.get(first_name="erfan")
In [35]: profile.comment.all()
Out[35]: <QuerySet [(<Comment: it is amazing. (-) Electric Trucks>)]>
In [36]:
```

۴- تعداد کل کامنتهایی که نویسنده پست آنها ابرکاربر باشد.

۵- کامنتهایی که وضعیت آنها مورد قبول باشد:

```
In [65]:
In [65]: Comment.objects.filter(status="a")
Out[65]: <QuerySet [(<Comment: oh that is so bad and exhausting. (-) Omicron>, <Comment: I think programming is always practical in industries nowadays. (-) Programming>, <Comment: i thought the the covid has left from the Earth about 2 mounth (-) Corona>)]>
In [66]:
```

۶- تعداد کامنت هایی که وضعیت آنها عدم تایید باشد.

```
In [66]: Comment.objects.filter(status="r").count()
Out[66]: 2
```