به نام خدا

**گزارش پروژه نهایی دوره پایتون و جنگو**

پریسا اعتمادی نژاد، سپیده نیک طبع

زمستان 1399

فهرست

[شرح موجویت ها 3](#_Toc68380447)

[Apps 4](#_Toc68380448)

[profiles app 4](#_Toc68380449)

[مدل CustomUser 4](#_Toc68380450)

[مدل Relationship 5](#_Toc68380451)

[Authentication با شماره تلفن و ایمیل 5](#_Toc68380452)

[توابع view 10](#_Toc68380453)

[Autocomplete 19](#_Toc68380454)

[سیگنال 20](#_Toc68380455)

[فرم 21](#_Toc68380456)

[Context processor 22](#_Toc68380457)

[Posts app 23](#_Toc68380458)

[مدل post: 23](#_Toc68380459)

[مدل like : 23](#_Toc68380460)

[مدل comment : 24](#_Toc68380461)

[توابع view 24](#_Toc68380462)

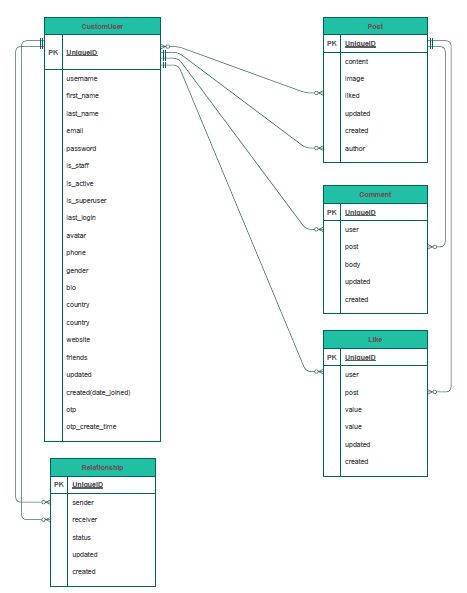
[فرم 27](#_Toc68380463)

[کتابخانه ها: 27](#_Toc68380464)

# 

# شرح موجویت ها

در این پروژه از 2 app شامل profiles و posts و 5 مدل شامل post، like، comment، CustomUser و Relationship استفاده شده است. در ادامه به شرح app ها و مدل ها میپردازیم.



شکل 1- نمودار ERD پروژه شبکه اجتماعی

# Apps

در این پروژه دو app تعریف شده است.

1. Profiles

این app برای مدیریت مشخصات کاربر و تمامی عملیات مربوط به کاربر طراحی شده است.

1. posts

این app برای مدیریت پست هایی که کاربر ارسال می کند و تمامی عملیاتی که مربوط به پست ها است طراحی شده است.

## profiles app

### مدل CustomUser

این مدل معرف یک کاربر در پروژه شبکه اجتماعی می باشد. در این مدل ما مدل پیشفرض User در جنگو را override کردیم. ویژگی های این مدل به شرح زیر است:

1. first\_name
2. last\_name
3. username
4. avatar
5. profile picture
6. phone
7. gender
8. bio
9. country
10. validators
11. website
12. email
13. slug
14. is\_active
15. is\_superuser
16. is\_staff
17. friends
18. updated
19. created
20. otp
21. otp-created-time

### مدل Relationship

این موجودیت رابطه بین دو کاربر شبکه اجتماعی را بیان می کند که ویژگی های این موجودیت به شرح زیر است:

1. Sender

کاربری که درخواست دوستی را فرستاده است

1. Receiver

کاربری که درخواست دوستی را دریافت کرده است

1. Status

وضعیت درخواست دوستی که می تواند ارسال شده یا پذیرفته شده باشد

1. Updated

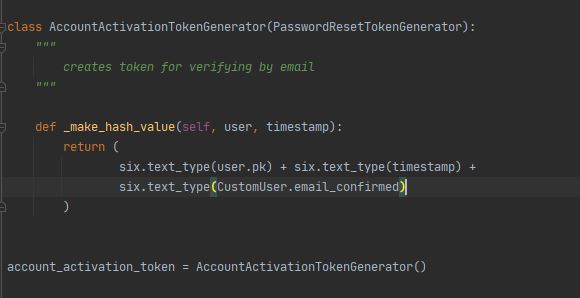
زمان بروزرسانی درخواست

1. Created

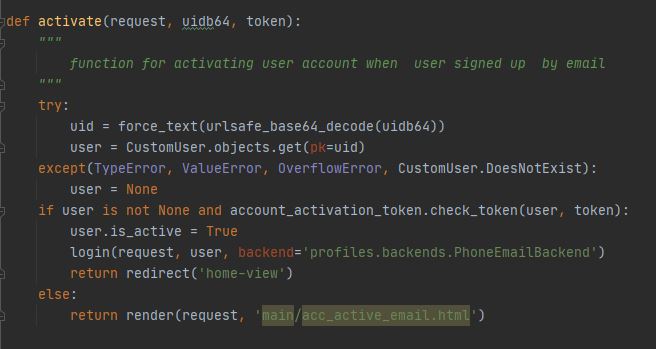
زمان ارسال درخواست

### Authentication با شماره تلفن و ایمیل

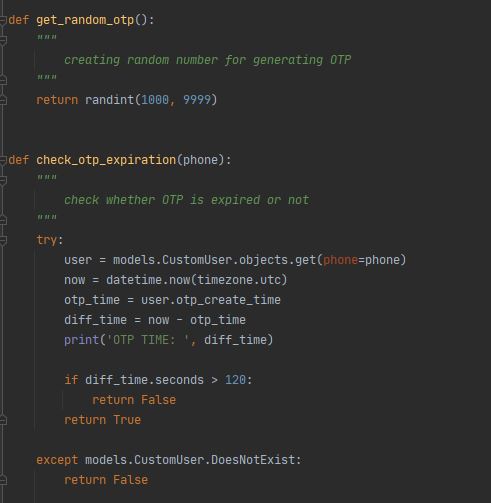
از کلاس زیر برای ایجاد یک token برای verify با ایمیل استفاده می شود.



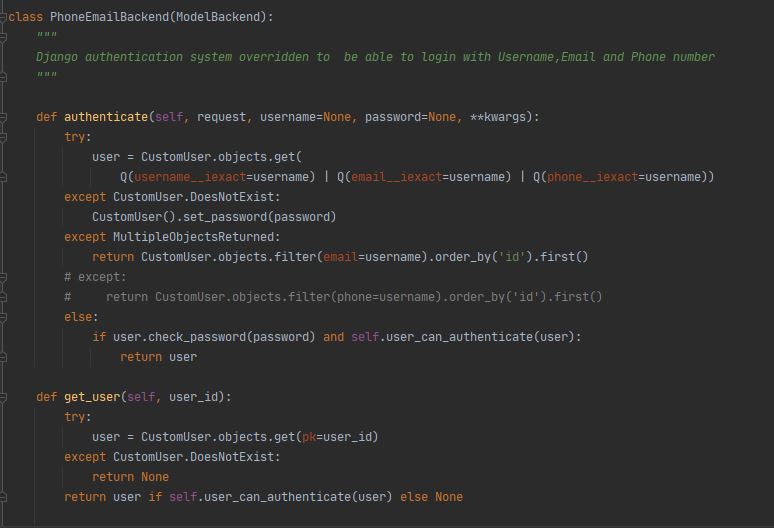
متد زیر عملیات active کردن با ایمیل را مدیریت می کند



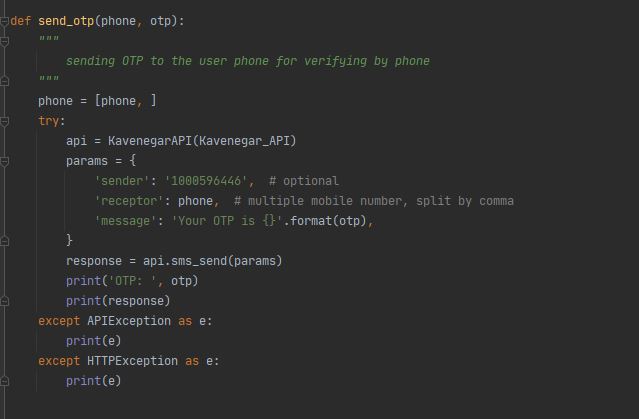
از متد زیر برای تولید رمز یک بار مصرف و بررسی تاریخ انقضا داشتن رمز یک بار مصرف با استفاده از کاوه نگار استفاده می شود.



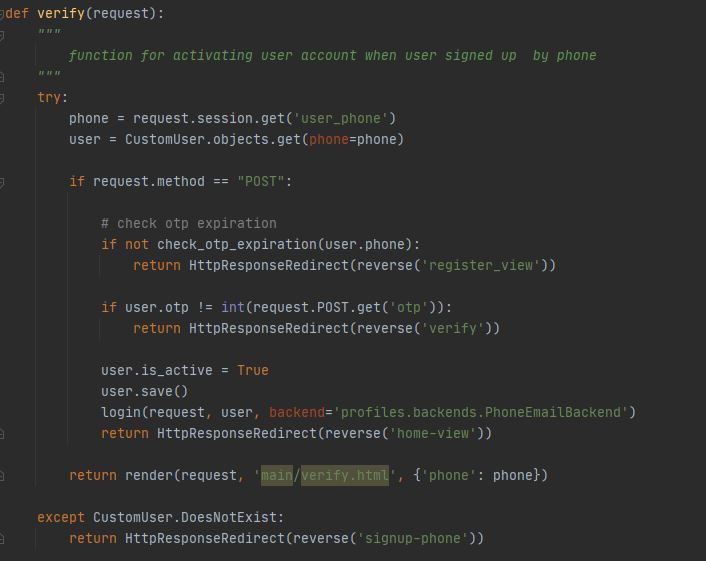
با استفاده از متد زیر می توان backend احراز هویت در جنگو را تغییر داد



از متد زیر برای ارسال رمز یک بار مصف استفاده می شود.



از متد زیر برای احراز هویت کاربر با استفاده از رمز یک بار مصرف استفاده می شود

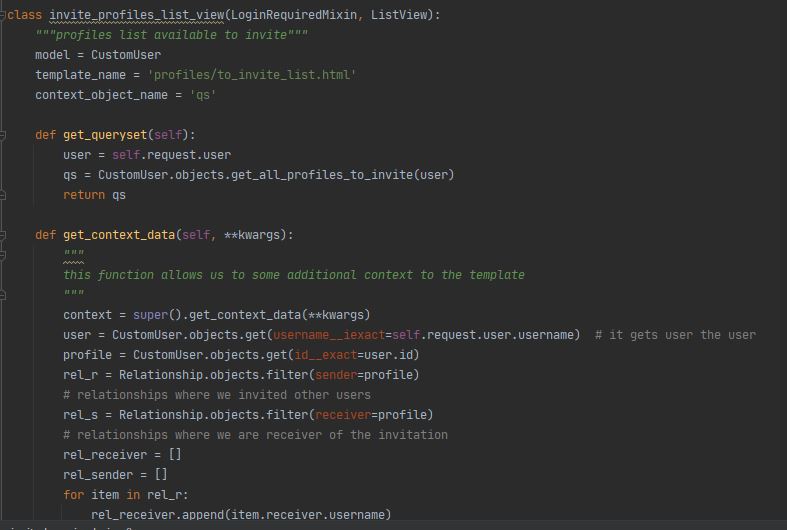


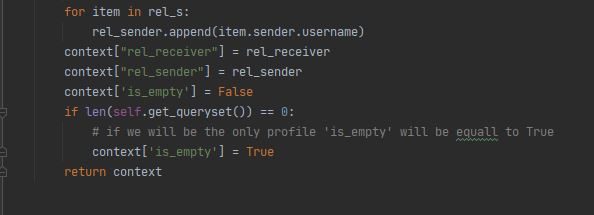
### توابع view

وقتی کاربر از سمت دوست خود یک درخواست دریافت می کند دو عملکرد می تواند انجام دهد یکی اینکه درخواست را قبول کند یا اینکه آن را رد کند که در توابع زیر این عملیات مدیریت شده است.

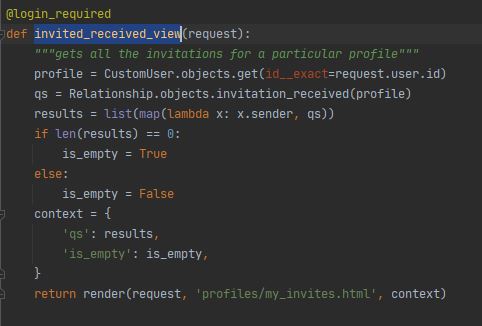


هر کاربر می تواند بعد از login لیست تمام کسانی را که امکان ارسال درخواست برای آنها وجود دارد را ببیند که این عملیات با کلاس زیر مدیریت می شود.

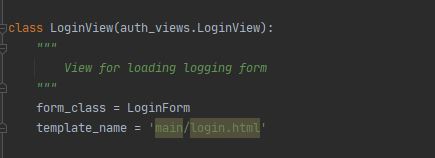




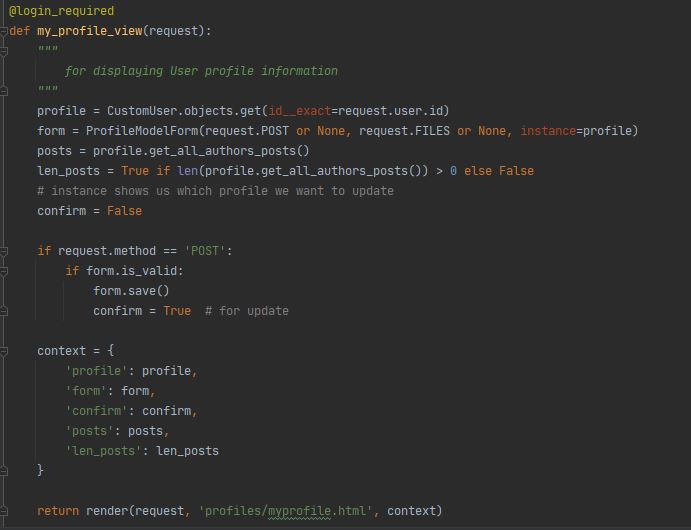
هر کاربر می تواند در پروفایل خود بعد از login لیست تمام دعوت هایی که دریافت کرده است را ببیند که این عملیات با تایع زیر مدیریت می شود.



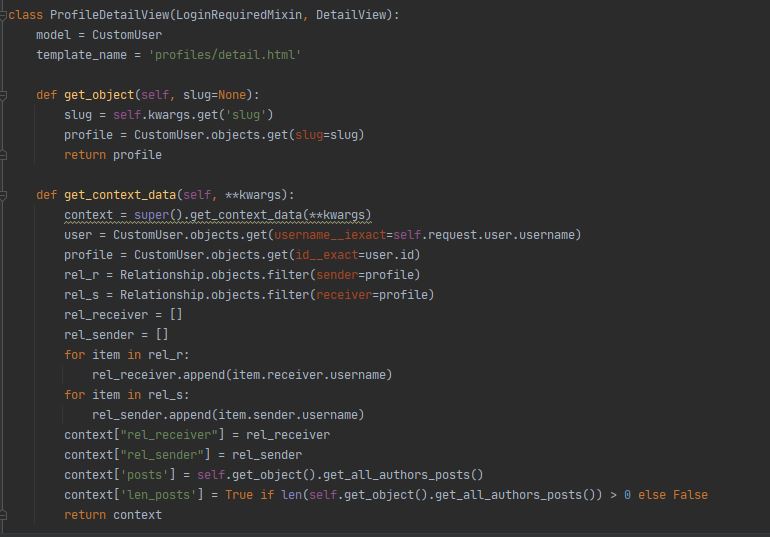
هر کاربر می تواند در پروژه login کند که این عملیات با تایع زیر مدیریت می شود



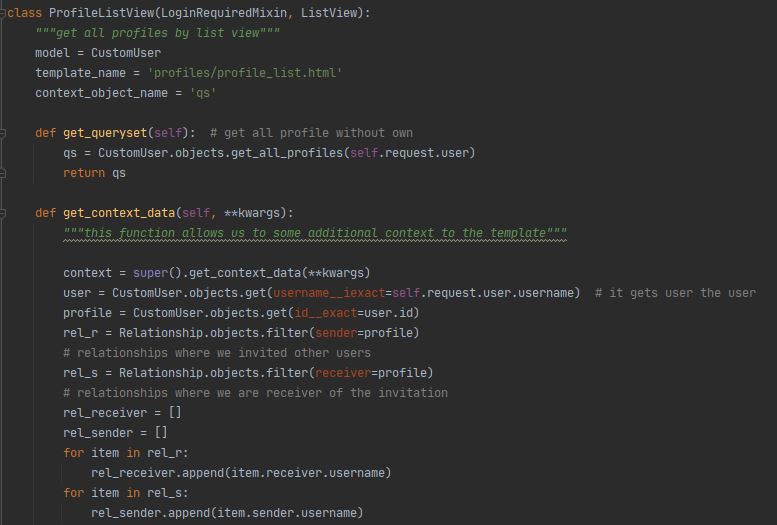
هر کاربر می تواند در پروفایل خود بعد از login اطلاعات کاربری خود را ببیند.

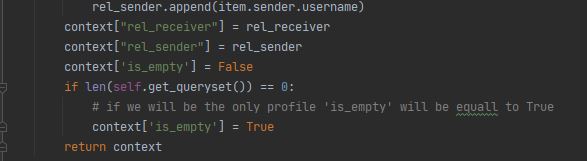


هر کاربر می تواند برای کاربر دیگر درخواست دوستی ارسال کند و یا اینکه از سمت کاربر دیگر درخواست دوستی دریافت کند که با استفاده از کلاس زیر می توان لیستی از درخواست های دریافت شده و لیستی از درخواست ارسال شده را بدست آورد و با بررسی مقادیر این لیست ها در مورد نشان دادن دکمه های مربوطه در template تصمیم گرفت.

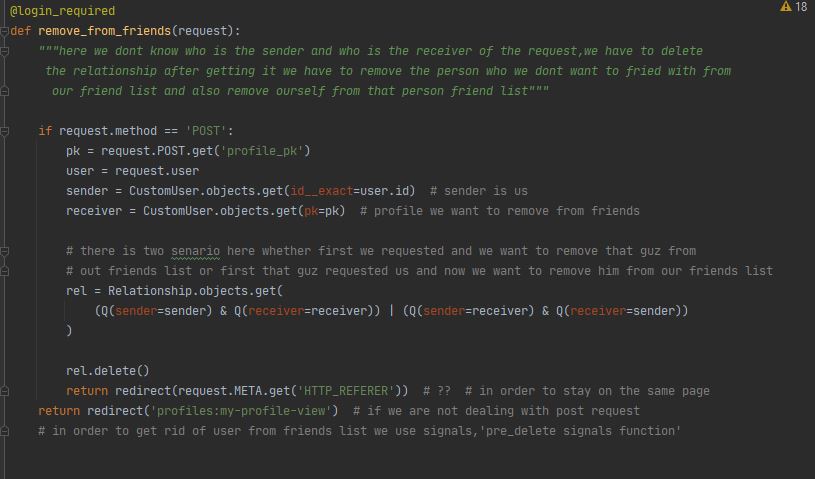


در کلاس زیر لیست همه ی کاربران با وضعیتشان نسبت به کاربر فعلی که در برنامه login است را نشان می دهد

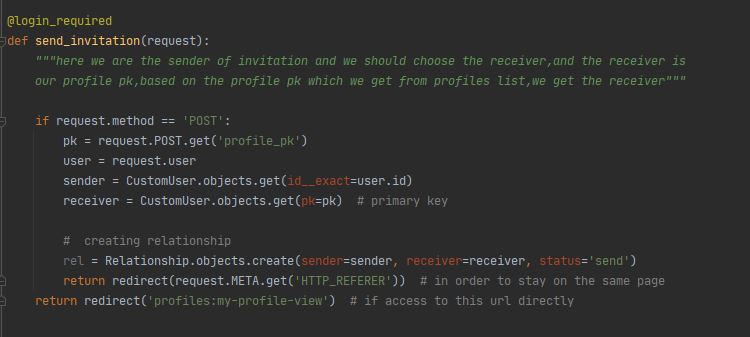




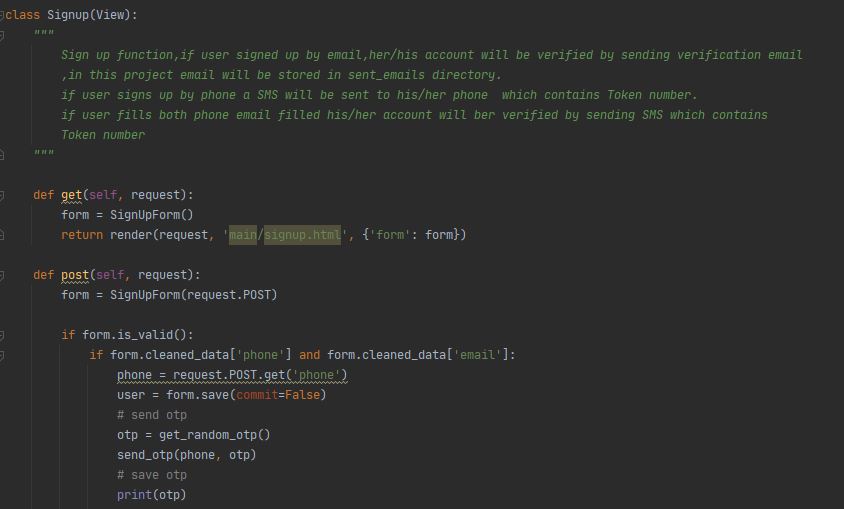
از متد زیر برای حذف یک کاربر از لیست دوستان کاربر فعلی استفاده شده است.

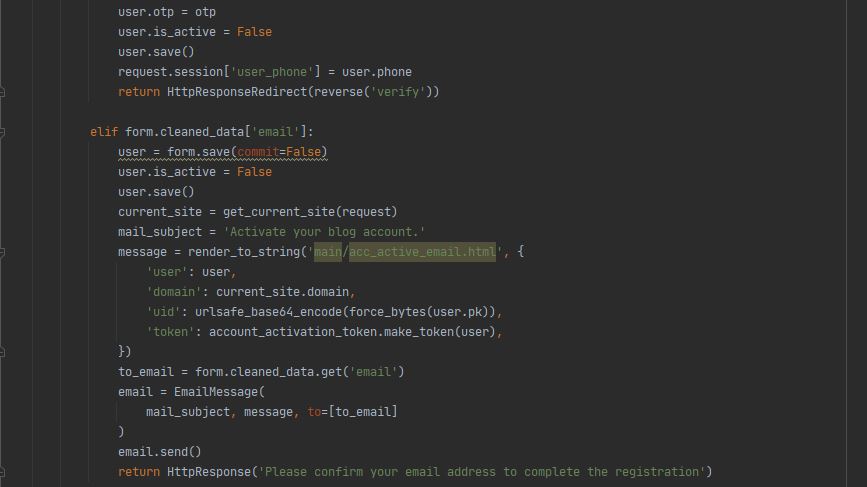


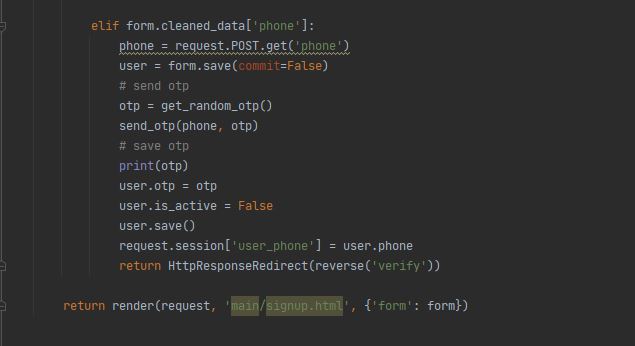
از متد زیر برای ارسال درخواست برای افراد موجود در این برنامه می توان استفاده کرد.



متد زیر signup با استفاده از email و شماره تلفن را مدیریت می کند که اگر کاربر شماره تلفن وارد کند برای او یک پیامک حاوی یک عدد 4 رقمی برای تایید آن ارسال می شود و اگر کاربر با ایمیل وارد شود در فایل send email پروژه یک لینک برای فرد ساخته می شود که با کلیک بر روی آن وارد صفحه کاربری خود می شود حال اگر هر دو مورد را وارد کند بصورت پیشفرض با شماره تلفن login می شود

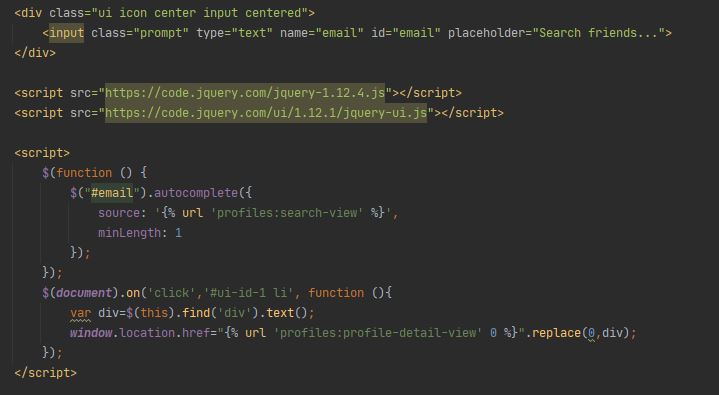




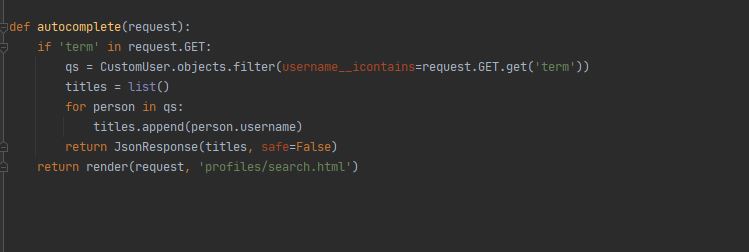


### Autocomplete

برای بخش autocomplete پروژه از تابع آن در javascript استفاده می کنیم که بصورت زیر نوشته می شود. مقدار minlength در این تابع برابر 1 در نظر گرفته می شود که این به معنای حداقل تایپ یک کاراکتر برای شروع جست و جو است.



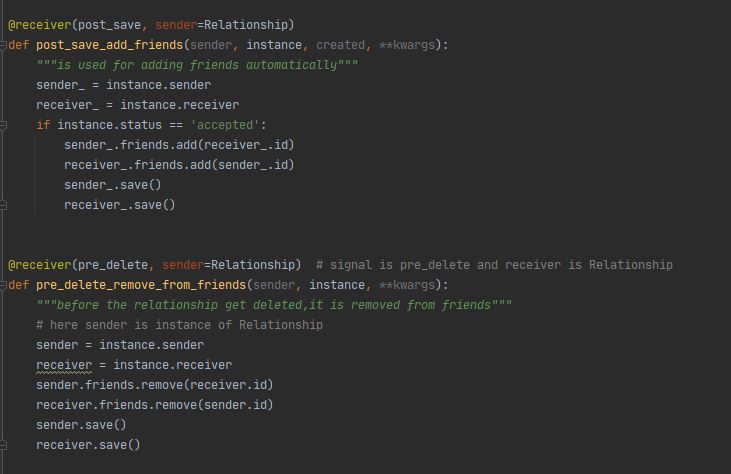
در view مربوط به profile از تابع زیر برای بخش autocomplete پروژه استفاده می شود. در این تابع به ازای تایپ هر کاراکتر در دیتابیس می گردد و نتیجه را نشان می دهد.



### سیگنال

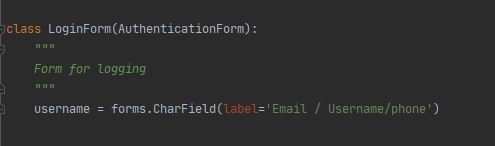
برای اینکه وقتی کاربر وضعیت یک relation را تغییر می دهد همان لحظه نیز در پروفایل کاربر مورد نظر بعنوان friend ثبت شود یا از لیست friend ها حذف شود، از سیگنال استفاده می شود که برای کار با سیگنال بایستی تایع ready در فایل apps باز نویسی شود و در فایلی بنام signals سیگنال های مربوطه نوشته شود.





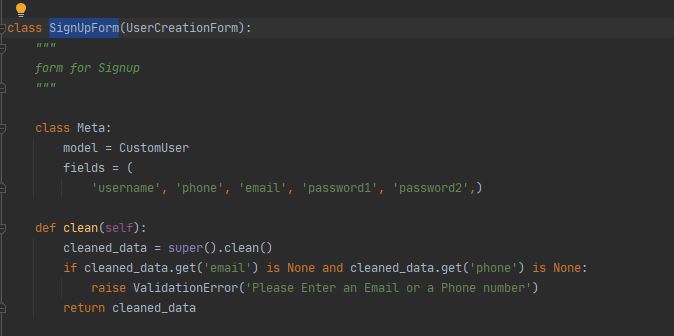
### فرم

از فرم زیر برای login کاربر استفاده شده است.



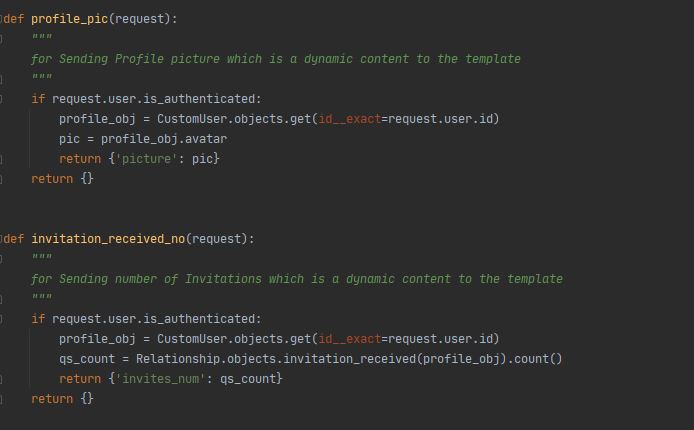
از قرم زیر برای مدیریت اطلاعات کاربر در پروفایل کاربر استفاده می شود.





### Context processor

برای ارسال مقادیر پویا به template می توان از context processor بصورت زیر استفاده کرد.



## Posts app

### مدل post:

این موجودیت مشخصات یک پست را بیان می کند. ویژگی های این موجودیت شامل موارد زیر است:

1. Content
2. Image
3. Liked
4. Updated
5. Created

### مدل like :

این موجودیت like هر پست را مشخص می کند. ویژگی های این موجودیت شامل موارد زیر است.

1. User

کاربری که لایک کرده مشخص می شود

1. Post

پستی که لایک شده مشخص می شود

1. Updated

زمان بروزرسانی لایک

1. Created

زمان انجام لایک

.

### مدل comment :

این موجودیت کامنتی که کاربر زیر هر پست درج می کند را مشخص می کند.

1. User

کاربری که کامنت را درج کرده است

1. Post

پستی که کامنت زیر آن درج شده است

1. Body

محتوایی که کامنت شده است

1. Updated

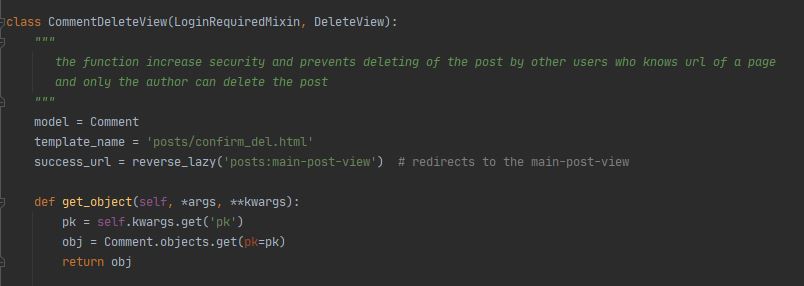
زمان بروزرسانی کامنت

1. Created

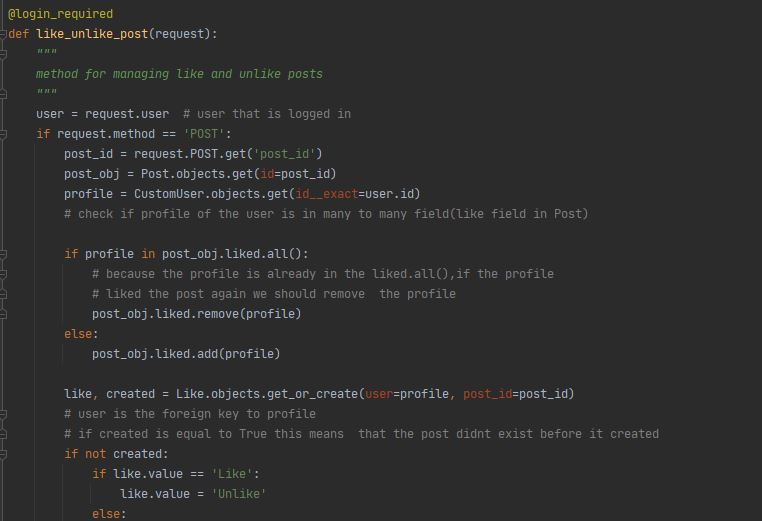
زمان درج کامنت

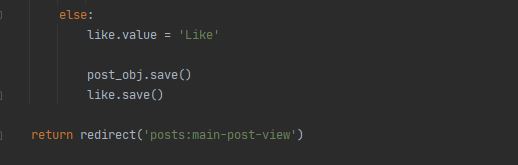
### توابع view

از کلاس زیر برای حذف کامنت یک پست استفاده می شود.

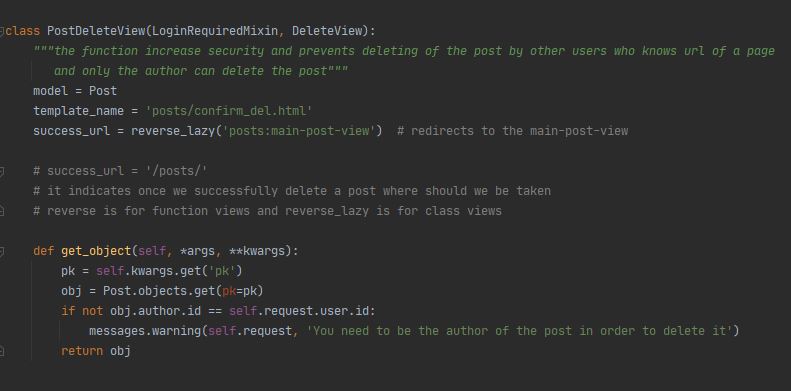


متد زیر like و unlike کردن یک پست را مدیریت می کند.

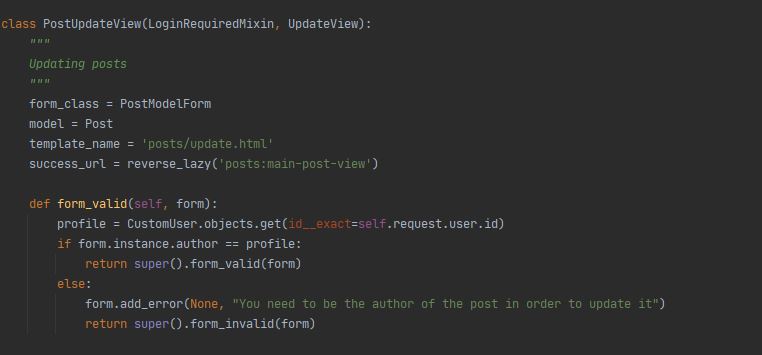




از متد زیر برای حذف یک پست تنها توسط نویسنده پست استفاده می شود.



از کلاس زیر برای بروزرسانی یک پست استفاده می شود



### فرم

از فرم زیر برای ثبت کامنت استفاده می شود.



از فرم زیر برای ثبت پست جدید استفاده می شود.



# کتابخانه ها:

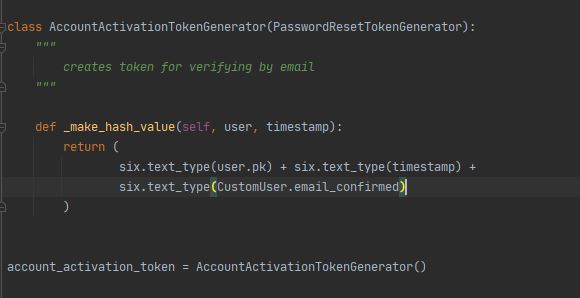
از کتابخانه های زیر در پروژه علاوه بر کتابخانه های معمول استفاده شده است.

کتابخانه pillow:

از این کتابخانه برای مدیریت عکس ها و دریافت عکس از کاربر استفاده می شود.

کتابخانه six:

از این کتابخانه برای بخش authentication در قسمت تاییدیه با ایمیل استفاده می شود.



کتابخانه kavehnegr:

از این کتابخانه برای ارسال پیامک حاوی عددی 4 رقمی برای تاییدیه ثبت نام استفاده شده است.