

پروژه پایانی جنگو:

با توجه به موضوع پایانی پروژه که پیاده‌سازی شبکه‌های اجتماعی بوده ما از دو اپ User و NewsFeed برای مدلسازی استفاده می‌کنیم که در ادامه هر کدام از آن‌ها شرح داده خواهند شد. همچنین برای استایل دادن به فرم‌های خود در جنگو از Django-crispy-forms استفاده می‌کنیم. برای دسترسی به فایل های استاتیک و همچنین استفاده از تصویر در خود مدل جنگو خطوط زیر را به settings.py اضافه می‌کنیم.

```
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
STATIC_URL = '/static/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'
```

خطوط زیر را برای redirect شدن پس از لاگین به settings.py اضافه می‌کنیم.

```
LOGIN_REDIRECT_URL = 'home'
LOGIN_URL = 'login'
```

در ادامه با توجه به نیاز به استفاده از سرویس ایمیل برای ثبت نام و همچنین فراهم کردن امکان تغییر رمز عبور با استفاده از ایمیل تنظیمات زیر را اضافه می‌کنیم که مربوط به ایمیل و رمز عبور gmail شخصی من می‌باشد.

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'fz.aligolzadehh@gmail.com'
EMAIL_HOST_PASSWORD = '2080748408ABCde'
```

در ادامه به توضیح اپ های مورد استفاده می‌پردازیم.

اپ User :

قبل از نوشتن مدل ما برای این که به طور خودکار بتوانیم از اسامی کاربران به عنوان slug استفاده کنیم ابتدا Django-autoslug را نصب می‌کنیم. همچنین برای استفاده از image به عنوان یک فیلد برای مدل User به نصب pillow نیز می‌پردازیم. ابتدا مدل پروفایل را تعریف می‌کنیم که شامل پارامترهای زیر می‌باشد: user که رابطه یک به یک با خود مدل User جنگو دارد. تصویر که برای نگه داشتن عکس پروفایل کاربران است. Slug

برای این که از Autoslug استفاده می‌کنیم. Bio که مربوط به توضیح کاربر را جمع به خودش است و میتواند خالی گذاشته شود. Friends که یک رابطه‌ی چند به چند با خود پروفایل دارد. همچنین در این کلاس توابع str و get_absolute_url را نیز مورد استفاده قرار می‌دهیم. بعد از آن مدل friends را تعریف می‌کنیم که شامل to_user نشان‌دهنده شخصی است که درخواست برای او ارسال شده‌است و from_user که نشان‌دهنده شخص ارسال‌کننده پیام می‌باشد. همچنین یک پارامتر timestamp نیز داریم که زمان ارسال درخواست را ثبت می‌کند.

در قسمت فرم‌ها از سه فرم استفاده می‌کنیم. UserRegisterForm که برای درخواست ثبت نام اولیه می‌باشد در این جا از دیفالت خود جنگو UserCreationForm استفاده می‌کنیم. UserUpdateForm که این قابلیت را به کاربر می‌دهد تا پروفایل خود را آپدیت کند یعنی ایمیل و نام کاربری و تغییرات دلخواه را اعمال کند. ProfileUpdateForm برای آپدیت تصویر و bio به کار می‌رود. در قسمت view از یک سری توابع استفاده می‌کنیم که آن‌ها را شرح خواهیم داد.

Users_list از این تابع برای نمایش کاربران استفاده می‌شود تا کاربر بتواند از بین آن‌ها request بدهد. friend_list برای نمایش تمامی دوست‌های یک کاربر به کار می‌رود. Send_friend_request برای ایجاد درخواست از کاربران مورد استفاده قرار می‌گیرد. Cancel_friend_request برای حذف request ارسالی از خود کاربر اولیه صورت می‌گیرد. Accept_friend_request برای پذیرفتن request دریافتی از کاربر دیگر مورد استفاده قرار می‌گیرد. Delete_friend_request برای حذف کردن هرگونه درخواست دریافتی استفاده می‌شود. Delete_friend برای حذف هر کدام از friend ها می‌تواند مورد استفاده قرار بگیرد. Profile_view برای نشان دادن پروفایل هر شخص استفاده می‌شود و هر فرد می‌تواند اطلاعات مربوط به پروفایل خودش را به طور کامل مشاهده کند. تابع register برای ثبت نام در سایت استفاده می‌شود. تابع edit_profile با توجه به فرم‌های ساخته‌شده قابلیت ویرایش فرم‌ها را به ما می‌دهد. تابع search_users قابلیت جستجوی بین کاربران را برای ما فراهم می‌کند.

همچنین از template های مورد نیاز برای نمایش خواسته‌های مسئله استفاده شده و html های مربوط به توابع user در قسمت templates/User قرار دارد. Layout.html به عنوان پایه برای تمامی صفحات دیگر html استفاده شده که برای استایل‌دهی به آن‌ها می‌باشد. در قسمت profile.html از {% block searchform%} برای انجام عملیات جستجو بین کاربران استفاده می‌کنیم.

اپ NewsFeed :

در این اپ ابتدا مدل **Post** را تعریف می‌کنیم. این مدل از پارامترهای اشاره‌شده در ادامه استفاده می‌کند. **Description** برای بیان توضیحات مربوط به هر پست. **Pic** برای تصویر به اشتراک گذاشته‌شده توسط کاربر که در آدرس مربوط به مدیا ذخیره می‌شود. **Date_posted** برای نمایش زمان ارسال پست مورد استفاده قرار می‌گیرد. **User_name** که رابطه یک چند با **post** به صورت کلید خارجی دارد. **Tags** برای نمایش موضوع مرتبط با پست مورد استفاده قرار می‌گیرد و با کمک آن بین پست‌ها جستجو را انجام می‌دهیم. همچنین در این قسمت نیز توابع **str** و **get_absolute_url** را تعریف می‌کنیم.

مدل بعدی **Comments** می‌باشد که شامل پارامترهای زیر می‌باشد. **Post** که به صورت کلید خارجی به پست می‌باشد چون هر کامنت متعلق به یک پست و هر پست می‌تواند شامل چندین کامنت باشد. **User_name** کلید خارجی می‌باشد که رابطه بین کاربر ارسال کننده کامنت و خود کامنت را برقرار می‌کند. **Comment** که برای خود متن آن می‌باشد و همچنین **comment_date** که مربوط به تاریخ ارسال کامنت می‌باشد.

مدل بعدی **Likes** است شامل پارامترهای **user** برای نشان دادن این که چه کسی آن را **like** کرده و **post** است که نشان می‌دهد **like** برای چه پستی صورت گرفته است.

در قسمت فرم‌ها از **NewPostForm** و **NewCommentForm** برای ایجاد پست جدید و کامنت جدید مورد استفاده قرار می‌گیرد.

در قسمت **views** از یک سری از توابع برای رسیدن به خواسته‌های مسئله بهره بردیم که در زیر شرح داده می‌شوند. **PostListView** برای نمایش پست‌ها به صورت ترتیبی می‌باشد به این صورت که پست‌های جدیدتر را زودتر نمایش می‌دهد. **UserPostListView** این تابع هم برای نمایش پست‌ها مورد استفاده قرار می‌گیرد با این تفاوت که این تابع تنها پست‌های مربوط به یک کاربر خاص را نشان می‌دهد. **Post_detail** برای نمایش جزئیات مربوط به هر پست به کار می‌رود. **Create_post** وظیفه ایجاد پست جدید را به عهده دارد. **PostUpdateView** وظیفه ایجاد تغییرات در هر پست را دارد. **Post_delete** وظیفه حذف هر پست را دارد. **Search_post** با توجه به تگ‌های هر پست بین آن‌ها جستجو می‌کند. تابع **like** وظیفه **like** برای هر پست را به عهده دارد.

Template های مربوط به این اپ نیز در قسمت **templates/NewsFeed** قرار می‌گیرد.