Custom user model

برای ساخت مدل ها ، ما تا الان به این شیوه بوده که کد های ساخت model را داخل فایلی به اسم models.py می نوشتیم

خب در قبل گفته شد که ،‌ما زمانی که یک مدل جدید تعریف می کنیم ، باید اول یک make migrations بزنیم و در قدم دوم ، migrate کنیم .

این make migrations کاری که می کنه اینکه کد ما را به یک کد میانی تبدیل می کند و ما زمانی که migrate می کنیم ، در واقع داریم این کد های میانی را در دیتا بیس می سازیم،

\*زمانی که اول کار ما هنوز make migration نزده ایم ، پس چطور می توانیم migrate کنیم ؟ پاسخ این سوال این است که خود جنگو یک app ساخته است به اسم auth که شامل مدلی به اسم User است که این یوزر در داخل خود قسمت کد های میانی راساخته است. ولی این کد های میانی ای که از قبل ساخته است و دارد شامل چه ستون هایی می باشد ؟

شامل مواردی مثل username , password , email و .. است و ما زمانی که migrate را می زنیم سریع می رود و این تغییرات را داخل دیتابیس ذخیره می کند. و در واقع یک جدولی به اسم user را می سازد که تمام اطلاعات مربوط به login را در آن جدول ذخیره می کند.

مشکل اینجاست که زمانی که ما می آيیم و از این مدل auth ، جدولی در دیتابیس می سازیم در واقع داریم دیتابیس را به auth وابسته می کنیم . و این باعث می شود اگر زمانی در قسمت لاگین خواستیم اطلاعات دیگری از کاربر نیز دریافت کنیم نیاز باشد که خود کلاس های جنگو را تغییر بدهیم و این کار بسیار سخت است و خیلی خطرناک می باشد .خب پس باید :

* اول پروژه یک کاری را باید انجام دهیم و اون این است که user model را عوض کنیم . به این معنی که قبل از هر کاری بیاییم و یک مدل جدید بسازیم مثلا به اسم custom user یا هرچیز دیگری و بگوییم که از AbstractUser ارث بری کند . با این کار تمام مزایا و خوبی های auth را داریم با این تفاوت که می توانیم قسمت های دلخواه را به آن اضافه کنیم .

ساخت پروژه جدید کتابفروشی

مراحل اولیه ساخت پروژه ، با این تفاوت که ابتدا ساخت app مربوط به account را انجاام داد ، اضافه کردن app به ستینگ فراموش نشود .

ساخت مدل Custom User :

در فایل models.py ، یک کلاس به اسم CustomUser() می سازیم .

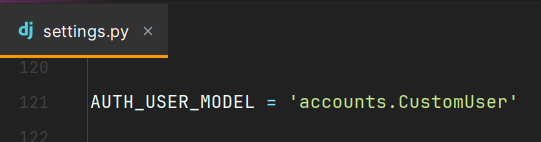
حالا می بایست ابتدا کلاس هایی که می خواهیم از آنها ارث بری کنیم را ایمپورت کنیم :‌

from django.contrib.auth.models import AbstractUser

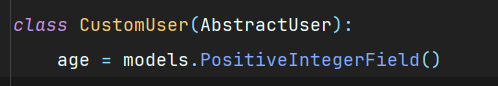
AbstractUser کلاسی است که ما از آن ارث بری می کنیم تا قسمت های ستون های آماده ی مربوط به لاگین را برای ما بسازد . البته که قسمت AbstractBaseUser هم هست که میتوان از آن ارث بری کرد. ولی تفاوتی که دارند این است که قسمت AbstractBaseUser خیلی بیسیک تر و پایه ای تر است و خب قابلیت های کمتری نسبت به AbstractUser دارد .

در مرحله ی بعدی به مشکلی برمیخوریم ، و آن این است که جنگو به صورت دیفالت برای ساخت قسمت دیتابیس برای لاگین میاید و از auth app 🡪 models.User استفاده می کند و ما باید این دیفالت را تغییر دهیم . به این شیوه که :‌

در آخر فایل setting.py یک متغیر به اسم AUTH\_USER\_MODEL می گذاریم و می گوییم که برو از app accounts و از مدل CustomUser استفاده کن.



و حالا با این کار مدل CustomUser ما تمام قسمت هایی مثل username , firstname , lastname, email , password , … را داخل خود جای می دهد و تازه اگر خواستیم قسمتی را به این موارد اضافه کنیم ، کافی است در قسمت این مدل اضافه کنیم (‌برای مثال age , nat\_idو ...)



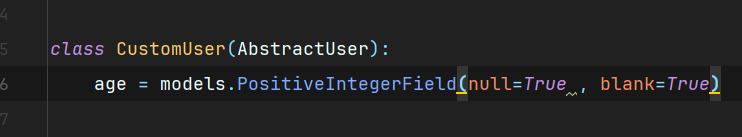
چند تغییر کوچک...

برای مشخص کردن اجباری یا اختیاری وارد کردن بعضی از فیلد ها می توانیم کار باحالی کنیم که به صورت زیر است .

ما برای هر ستونی که می سازیم ،‌می توانیم دو ورودی بهش بدیم :‌یکی ورودی null , blank : خب تفاوت این ها چیست؟( هرکدام فقط True , False )

Null مال دیتا بیس است ، یعنی اینکه وقتی خود دیتا بیس در یک فیلد کاملا خالی باشد

ولی ‌blank برای فرم ها است. و می گوید اگر کاربر این قسمت از فرم راخالی گذاشت بهش ارور نگیر. ( اگر یادت باشه ارور مربوط به فرم ها را validation می داد . پس در واقع این قسمت مربوط به validation است )



نکته مهم اینجاست ، که زمانی که ما از charField ها استفاده می کنیم ، زمانی که هیچ مقداری وارد نکنند ، یک استرینگ خالی ذخیره می شود . و این استرینگ خالی درون دیتا بیس ذخیره می شود. پس اگر null را هم True کرده باشیم ، باعث می شود که دوتا مقدار برای خالی در نظر گرفته شود .

پس خود جنگو توصیه می کند که برای استرینگ ها فقط blank را خالی بگذاریم و null را دست نزنیم.

فرم های Custom User :‌

دوجا هست که ما فرمی داریم که مربوط به user است . یکی فرم مربوط به sign up و یک هم فرمی که داخل خود پنل ادمین جنگو بود و روش که کلیک می کردیم اطلاعات یوزر ها را می توانستیم ببینیم.

حالا زمانی که ما CustomUser را ساختیم باید کاری کنیم که به این دوتا فرم هم بگوییم که بیا و از این مدل جدید استفاده کن.

خب این کار خیلی ساده است . باید داخل خود app مربوط به accounts می آییم و یک فایل جدید به اسم forms.py می سازیم .

وداخل این فایل باید دوتا چیز بسازیم به اسم های :‌

UserCreationForm 🡪 Sign up

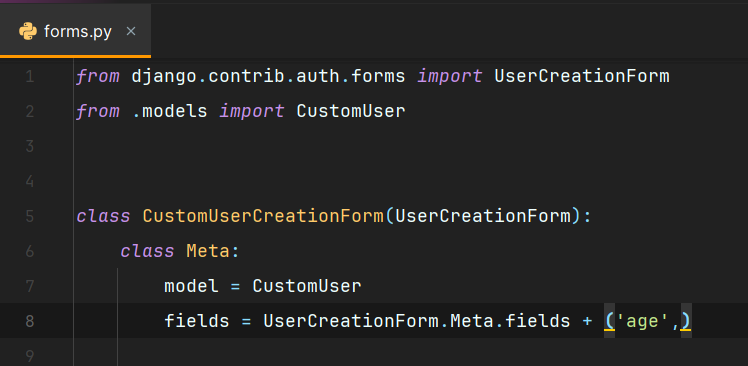
UserChangeForm 🡪 Admin

که بالایی برای sign up است و دومی برای admin .

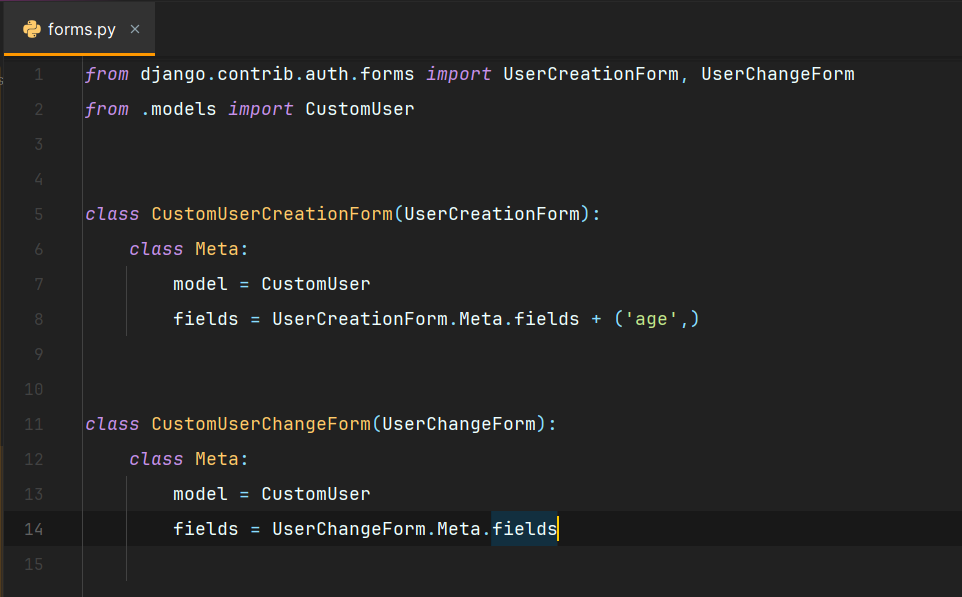
پس یک کلاس می سازیم برای CreationForm و چون قراره یکسری اطلاعات اضافه به فرم بدهیم ، از کلاس Meta استفاده می کنیم . و داخل اون میگیم که آقاااا مدلی که قراره استفاده کنی ، اون قبلیه نباشه هااا و از این مدل جدید CustomUser استفاده کن .

و بعدش چون می خواهم فیلد جدیدی به این مدل اضافه کنم ، باید اون را بهش معرفی کنم

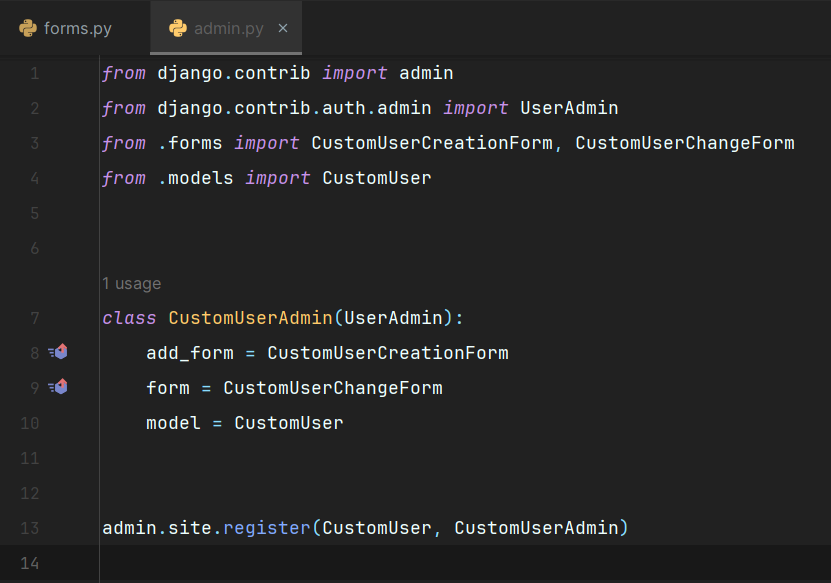
( فیلد جدید هایی که بهش معرفی می کنی باید از نوع tuple باشد )



و یک کلاس دیگه هم باید بسازم به اسم CustomUserChangeForm که باید از UserChangeForm ارث بری کند



در قدم بعدی : در داخل فایل admin.py در app اکانت ها یک کلاس جدید می سازیم که بتوانیم userAdmin اصلی خود جنگو را بتوانیم ادیت کنیم . وقست های مربوط به اضافه کردن به فرم و فرم کلی را و همچنین اسم مدل را بهش میگیم. و در نهایت هم رجیستر می کنیم :



حالا چون خودمان هنوز مدل جدید نساختیم ،‌ داخل ترمینال می نویسیم که makemigrations انجام بده برای app accounts .



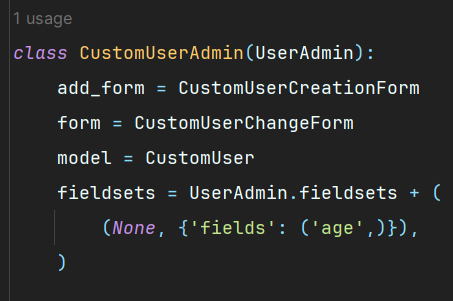
و در قدم بعدی migrate می کنیم .



و حالا میتوانیم یک سوپر یوزر بسازیم و سرور را ران کنیم :‌

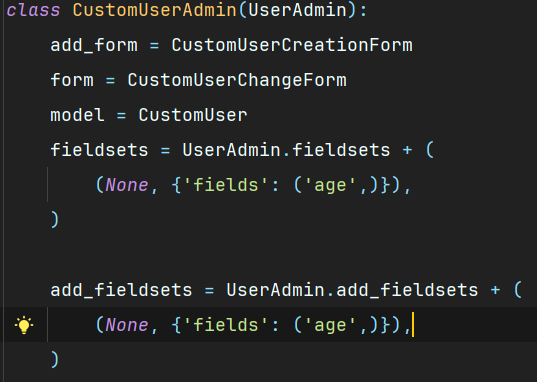
ولی وقتی وارد پنل ادمین شدیم ، هنوز قسمت age در داخل یوزر ها نبود . و برای اینکه اضافه بشود یک تغییر دیگر لازم است که دلیل کد هاشو نگفت و گفت که فقط حفظ کنید.

به این ترتیب که داخل فایل admin.py باید قسمتی تحت عنوان fieldsets اضافه کنیم.



این fieldsets تمام فیلد هایی است که داخل پنل ادمین برای ویرایش یوزر وجود دارد . پس با این کار یکی اضافه کردیم بهش .

حالا برای اینکه در صفحه ی ساین آپ نیز ‌age اضافه شود می بایست add\_fieldsets را نیز ااضافه کنیم.

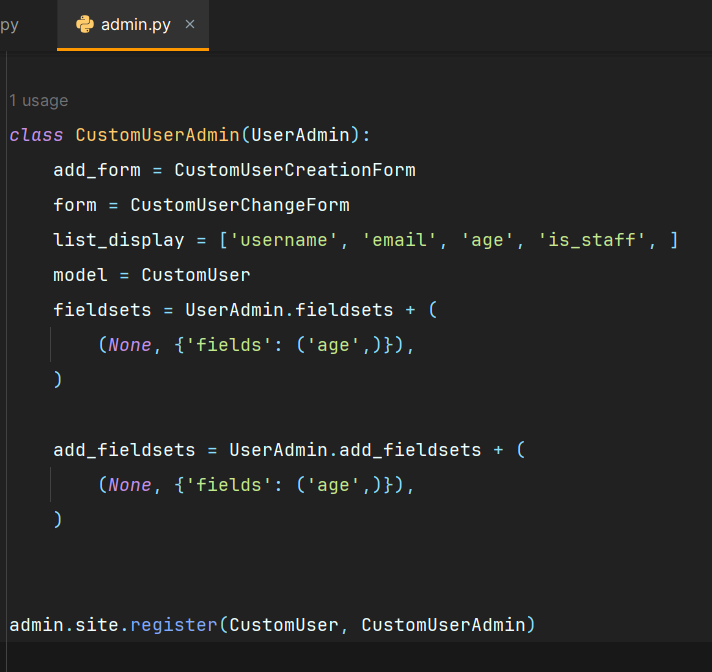


برای اینکه age در لیست مشخصات یوزر نمایش داده شود هم که قبلا گفته شده باید از قسمت

List\_display استفاده کنیم .



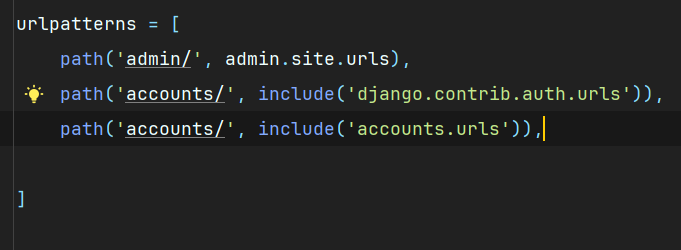
نمای کلی :‌



ساخت لاگین با مدل یوزر جدید :‌

در این قسمت قصد داریم که صفحه های لاگین و ثبت نام را با استفاده از customuser بسازیم .

در مرحله ی اول در قسمت urls مربوط به config می آییم و urls های auth را include می کنیم و همچنین برای قسمت ثبت نام نیز ، قسمت تمام urls های موجود در app accounts را include می کنیم .

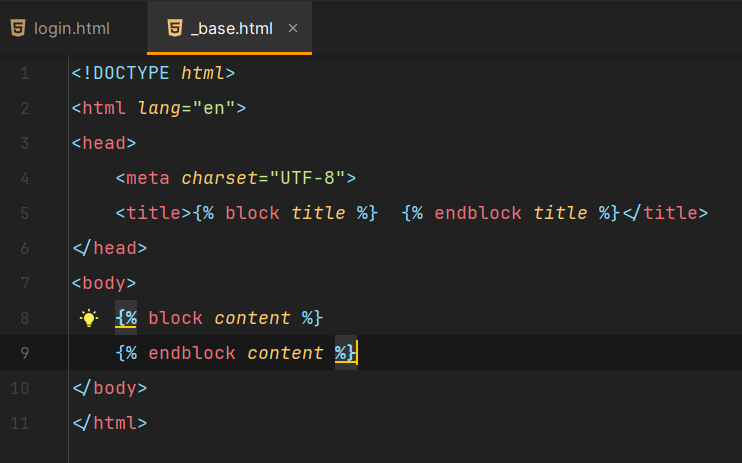


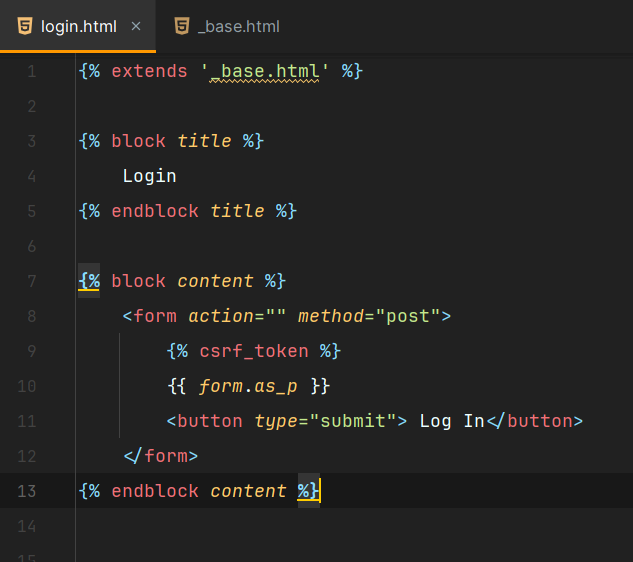
از آنجایی که ما urls را در app accounts نساخته ایم . این کار را انجام می دهیم .

و در این فایل ساخته شده قسمت ثبت نام ( sign up )‌ را مسیر دهی می کنیم. خب چرا قسمت لاگین را اینجا مسیردهی نمی کنیم ؟ چون اون را خود جنگو برای ما انجام داده است .

خب حالا باید ابتدا فایل templates مربوط به عملیات لاگین را معرفی کنیم . برای همین یک دایرکتوی به اسم templates ساخته و داخل آن یک دایرکتوری به اسم registration می سازیم . و فایل html را درون آن می سازیم . و از \_base.html ارث بری می کنیم .

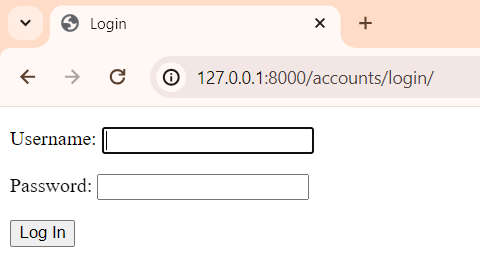
خب ما که هنوز فایل \_base.html را نساخته ایم . پس اون را هم میسازیم و فعلا یک ظاهر خیلی ساده با کد های معمولی به آن می دهیم .





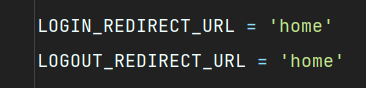
ولی اگر الان سرور را ران کنیم باعث می شود که ارور بگیرد و دلیل آن این است که اسم templates را در قسمت setting وارد نکرده ایم .





تکمیل لاگین و ساخت ثبت نام :

این مراحل معمولا در همه ی پروژه ها یکسان است به این ترتیب که اول در ستینگ دو url زیر را وارد می کنیم .



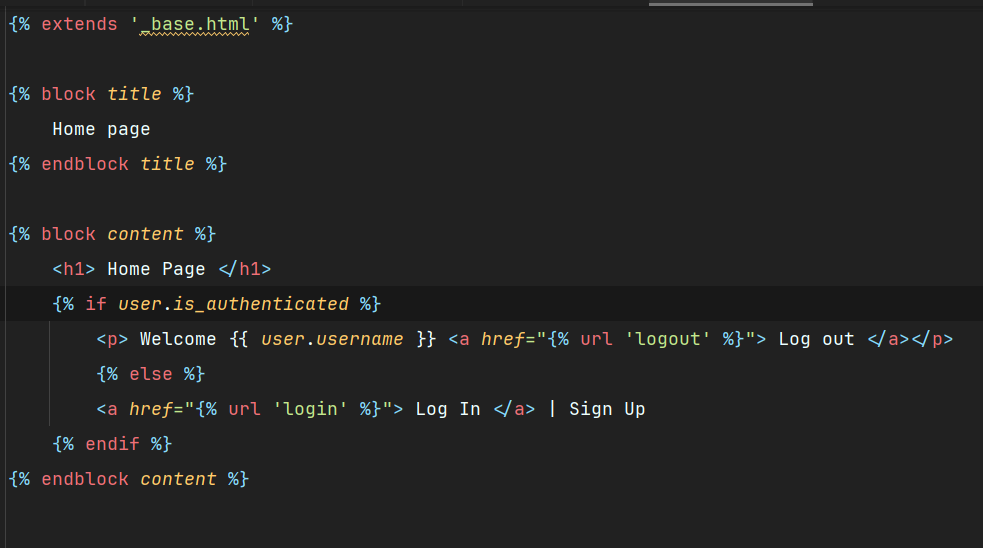
یعنی که اگر لاگین شدی یا لاگ اوت شدی برو به صفحه ی home

خب ما home را هنوز نساختیم . پس اول home را می سازیم . و چون فعلا نیاز نداریم که app جدیدی بسازیم ، می خواهیم با یک تمپلیت خیلی ساده و تک خطی بدون ساخت هیچ اپ دیگری این کار را انجام دهیم . به این شیوه که می توانیم در قسمت تعریف مسیر home که در قسمت config/urls.py است . با استفاده از وارد کردن templateview از آن استفاده کنیم.

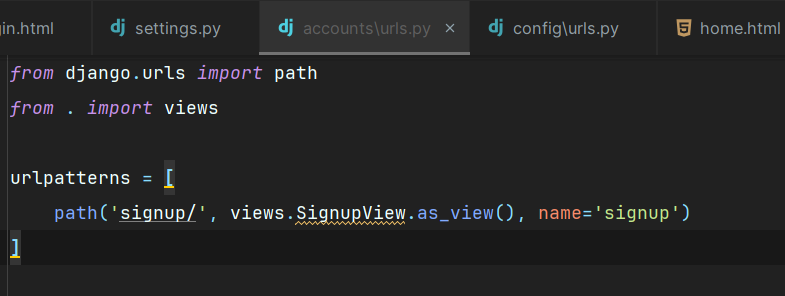




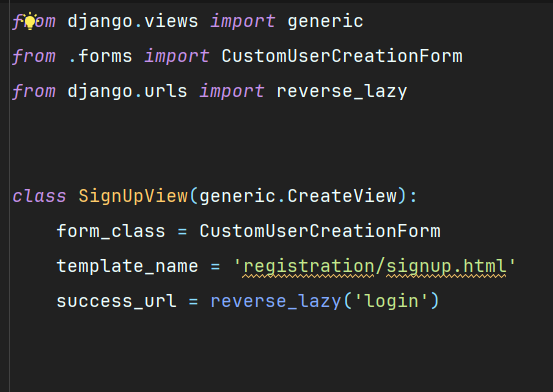
و حالا فایل تمپلیت خانه را می سازیم . ( باید در فولدر templates کلی بسازیمش )



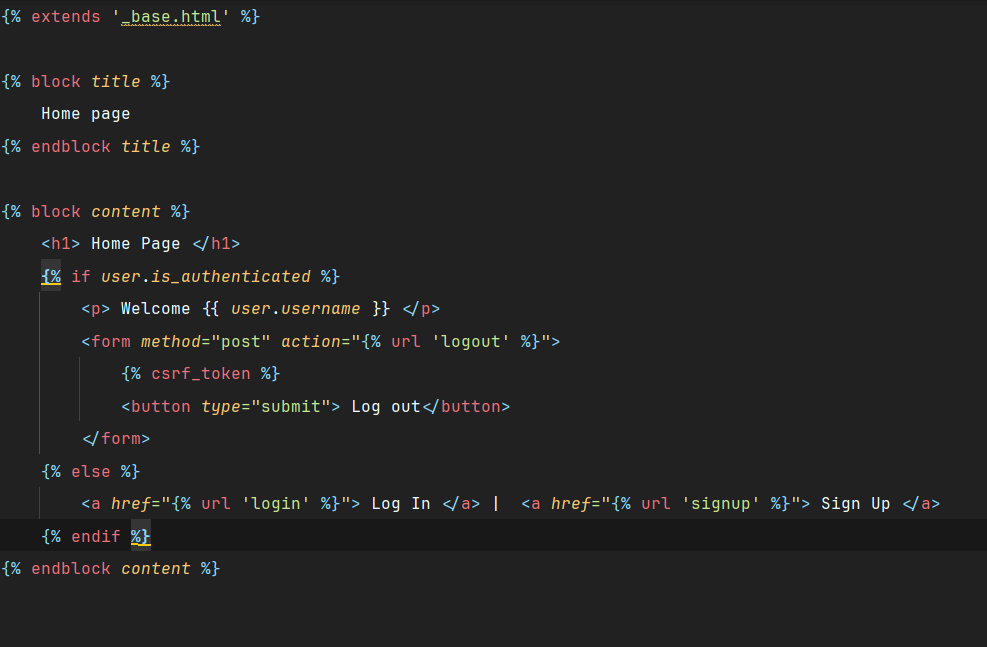
و حالا قسمت Sign Up را می سازیم تا بتوانیم لینکش را در این فایل html قرار دهیم.



خب حالا قسمت views : برای ساخت قسمتی برای sign up می بایست form\_ class را بهش بدیم که قبلا مدل را میدادیم. ولی در این روش چون customUserCreationForm را ساخته ایم، از آن باید استفاده کنیم .



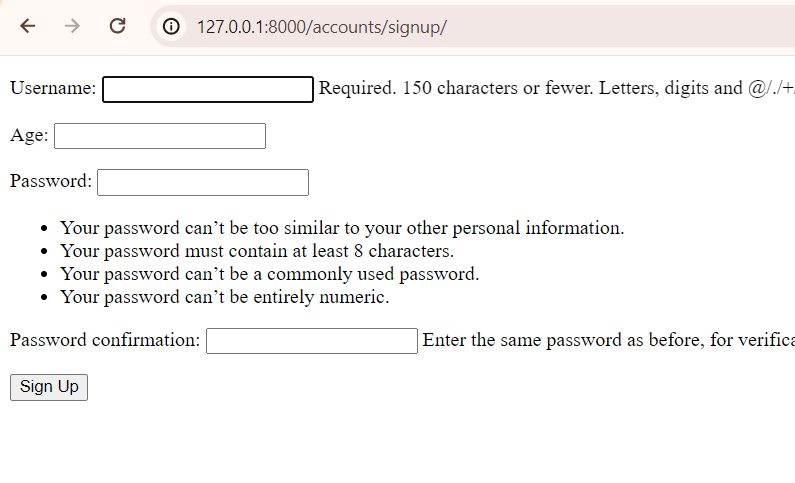
خب حالا که sign up کردن هم اسم داره ، در صفحه یhome ، لینکش را قرار می دهیم .



نکته : حواست به این باشه که قسمت لاگ اوت را تغییر دادم تا به ارور نخوره . چون در ورژن ها جدید جنگو امکان استفاده از تگ a برای لاگ اوت نیست .

حالا باید تمپلیت مربوط به sign in را درست کنیم . ( از همون مال Login استفاده می کنیم )





و می بینیم که این بار age نیز به فیلد ها اضافه شده است.

تغییر فرم های Custom User :

خب خوبی این روش این بود که می توانیم فیلد های ثبت نام را تغییر دهیم . و در این قسمت می خواهیم همین کاررا تست کنیم.

قسمتی مربوط به username را و همچنین password تنها مواردی هستند که به صورت دیفالت خود فرم ساین آپ جنگو دارد. و مابرای اینکه به آنها اضافه کنیم در فایل forms.pyبه شیوه زیر عمل می کنیم .

