التعامل مع Django Admin

إنشاء حساب Admin

يوفر Django واجهة خاصة ب Admin يساعد في إنشاء وتعديل البيانات الخاصة ب Model عن طريق إنشاء Superuser

قبل البدء بإنشاء Superuser، قم بالدخول على ملف settings.py لاحظ أن أحد التطبيقات لدينا هو تطبيق

```
INSTALLED_APPS = []
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
```

وأيضا عند فتح ملف urls.py نلاحظ وجود

```
from django.contrib import admin
from django.urls import path

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

نقوم بفتح ملف admin.py ونقوم بكتابة الأوامر التالية:

```
from django.contrib import admin
from .models import Movies_Info, Publisher, Contributor, Movie
Contributor, Review

admin.site.register(Movies_Info)
admin.site.register(Publisher)
admin.site.register(Contributor)
admin.site.register(MovieContributor)
admin.site.register(Review)
```

في الأسطر السابقة قمنا بالتالي:

- عبل import ل Movies_Info, Publisher, Contributor, MovieContributor, Review.
 - استخدام register method وهي تساعد بجعل models متاحة لتطبيق

نقوم بتنفيذ الأوامر:

```
python manage.py runserver

python manage.py createsuperuser
```

ثم نقوم بإدخال username و email و password الذي نريد، ثم كتابة الأمر.

python manage.py runserver

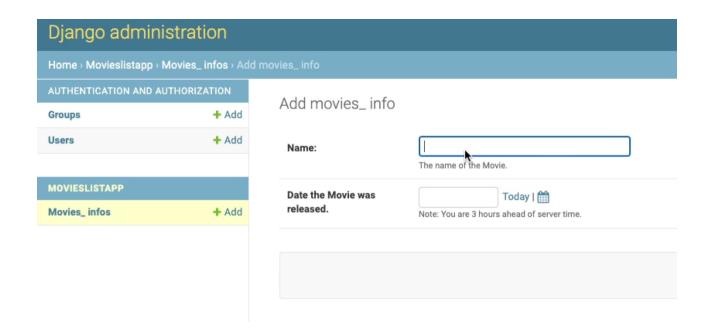
نقوم بالدخول على الرابط التالي (http://127.0.0.1:8000/admin) الآن بعد الدخول على صفحة admin سوف نلاحظ ظهور جميع Models التي أضفنا ويمكن إضافة أو تعديل البيانات.

Django administration

Site administration



• مثال: إضافة بيانات على Movies_Info.



- عند دخولنا لأحد Models على سبيل المثال (Publishers) يمكن أن نلاحظ وجود Validation بشكل تلقائي وهي تشير إلى Help text الذي قمنا بإضافته في field بداخل models.py.
 - عندما نحاول إدخال أكثر من ٥٠ حرف في الاسم لن يسمح لنا بذلك.
 - عند كتابة معلومات بشكل خاطئ مثلا: إدخال نص عادي في اسم الموقع أو البريد الإلكتروني سوف يظهر لنا Error عند كتابة معلومات بشكل خاطئ مثلا: إدخال نص عادي في اسم الموقع أو البريد الإلكتروني سوف يظهر لنا Message

Django administration						
Home > Movieslistapp > Publishers > Add publisher						
AUTHENTICATION AND AUTHORIZATION		Add publisher				
Groups	+ Add	Add publisher				
Users	+ Add	Name:				
			The name of the Publisher.			
MOVIESLISTAPP		Website:				
Contributors	+ Add		The Publisher's website.			
Movie contributors	+ Add	Email:				
Movies_infos	+ Add		The Publisher's email address.			
Publishers	+ Add					
Reviews	+ Add					

```
settings.py ×  views.py ×  Movies_list.html ×  admin.py ×  models.py ×

from django.db import models

from django.contrib import auth

lass Publisher(models_Model):
    name = models_CharField(max_length=50, help_text="The name of the Publisher.")
    website = models_URLField(help_text="The Publisher's website.")
    email = models_EmailField(help_text="The Publisher's email address.")
```

عملیات CRUD باستخدام عملیات

أولا: عملية الإضافة Create

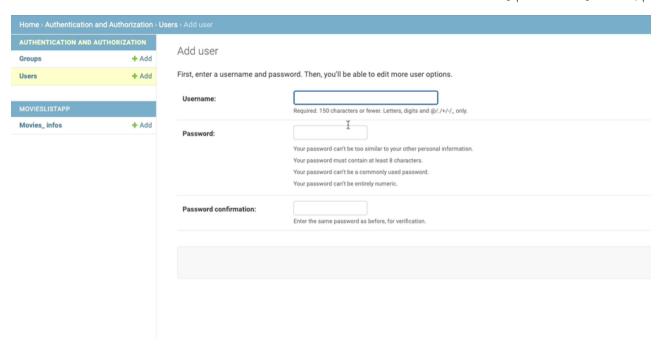
يمكن إضافة مستخدمين وإعطائهم الصلاحيات المناسبة لهم عن طريق الضغط على Add كما في الشكل التالي.

Django administration

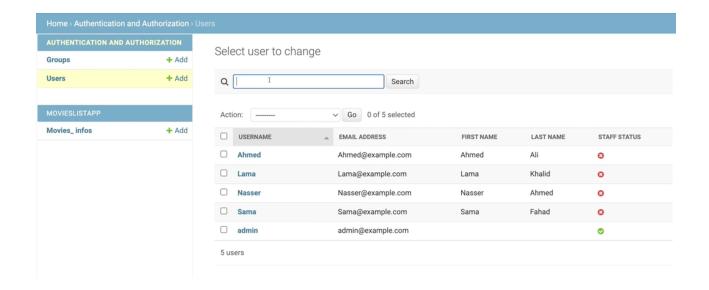
Site administration



ثم إدخال معلومات المستخدم وحفظها



ثانيا: عملية الاسترجاع Read في الأعلى. يمكن استرجاع البيانات عن طريق البحث Search Bar في الأعلى.



ويمكن أيضا عن طريق FILTER panel الموجود على يمين الصفحة من خلاله نستطيع عمل FILTER أو تصفية للمستخدمين بحسب الصلاحيات التي لديهم كالتالئ:

- المستخدمين (staff status) هم من يستطيع عمل Login لصفحة Admin و عند اختيار ves نلاحظ ظهور Admin User.
 - المستخدمين (superuser status) هم من لديهم كامل الصلاحيات أو Permissions وعند اختيار yes نلاحظ ظهور Admin User لأنه المستخدم الوحيد الذي لديه كامل الصلاحيات.
 - المستخدمين النشطين (active) وعند اختيار yes سوف نلاحظ ظهور كل المستخدمين. نستطيع تغيير حالة المستخدم إلى غير نشط عندما نريد حذف هذا المستخدم.

By staff status All Yes No By superuser status All Yes No By active All Yes No

ثالثا: عملية التحديث Update

عندما نريد التحديث على أحد بيانات المستخدمين نقوم بالدخول على المستخدم الذي نريد ونعدل البيانات ثم نضغط Soft deleting يمكن أيضا تحديث البيانات عن طريق مايسمى ب soft deleting وهذه العملية تفيد لعمل deactivate للمستخدم أو جعله غير نشط من دون حذف بيانات المستخدم يمكن تنفيذ ذلك عن طريق الدخول على المستخدم ثم حذف علامة ✓ من عند كلمة active و الضغط على save. و يمكن أيضا تعديل staff status أو يمكن أيضا تعديل staff status أو يمكن أيضا تعديل على staff status و يمكن أيضا تعديل على المستخدم ثم حذف على عن عديد كلمة السابقة.

رابعا: عملية الحذف Delete

عندما يطلب أي مستخدم حذف الحساب الخاص فيه، يقوم Admin بالدخول على صفحة User ثم يحدد User ومن عند Action ومن عند Action في الأعلى يختار Delete selected users ثم يضغط Go.

Select user to change

Q Search Action: 1 of 5 selected Go Delete selected users USERNAME LAST NAME STAFF STATUS **EMAIL ADDRESS** FIRST NAME Ahmed@gmail.com **Ahmed** Ahmed Lama Lama@example.com Khalid Lama Nasser Nasser@example.com Nasser Ahmed Sama Sama@example.com Fahad Sama admin admin@example.com 0 5 users

سوف تظهر صفحة تطلب منك التأكيد على حذف User، اضغط ves i'm sure، سوف تلاحظ أنه تم حذف User.

إضافة وحذف Users و Groups عن طريق Django

في Django يوجد تطبيق Authentication يشمل المستخدمين(Users Model) و المجموعات (Groups) (Model) و المجموعات (Permissions) والصلاحيات الخاصة فيهم (Permissions).

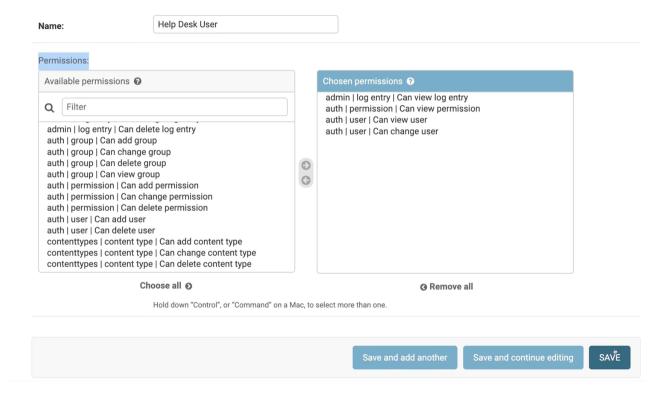
يمكن أن يكون User تابع لأكثر من Group وكل Group لديه صلاحيات خاصة فيه.

عندما نبني موقع يحتوي عدد كبير من المستخدمين تكون إدارة المستخدمين عملية صعبة على Admin لذلك يحتاج Admin تقسيم المهام على مستخدمين معينين أو مجموعات معينة بحيث يكون لديها بعض الصلاحيات لإدارة بعض المهام.

يمكن إنشاء مجموعة عن طريق التالي:

- ندخل صفحة Admin ومن عند Authentication and Authorization نضغط Groups نضغط add
 - نعطي اسم للمجموعة (مثال: Help Desk User).
 - نضيف الصلاحيات المناسبة للمجموعة عن طريق السهم بحيث يمكن نقل permissions من جهة available chosen permissions للجهة الثانية
 - ثم نضغط Save.

Add group



يمكن إضافة مستخدمين لمجموعة معينة عن طريق التالى:

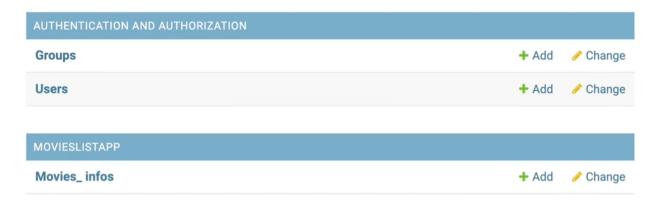
- نقوم بالدخول على Users ثم نختار أحد المستخدمين.
- نختار Staff Status حتى نسمح للمستخدم بعمل Login على صفحة .
 - من عند Groups نختار اسم
 - نضغط save.

حتى نتأكد من الصلاحيات التي أعطيناها للمستخدم، نحاول الدخول على صفحة Admin من حساب المستخدم.

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration



صفحة Admin

Django administration			WELCOME, AHMED.
Site administration			
AUTHENTICATION AND AUTHORIZATION Users	∂ Change	Recent actions	
•		My actions None available	

صفحة المستخدم الذي أعطيناه بعض الصلاحيات

نلاحظ أن Interface الخاص بالمستخدم مختلفة قليلا عن Interface على سبيل المثال:

- في Interface الخاص بالمستخدم لايوجد Groups لأننا لم نقم بإعطاء المستخدم أي صلاحيات خاصة ب Groups.
 - إذا دخلنا على أي User نلاحظ أنه لا يستطيع حذف أو إضافة User لأننا لم نعطي صلاحيات للحذف أو الإضافة فقط أعطيناه صلاحيات للتغيير أو عرض User.

تخصيص واجهات Admin

تخصيص واجهات Admin باستخدام اist_display

حتى نستطيع عمل تخصيص (customization) لموقع Admin بحيث نستطيع عرض معلومات Publisher كجدول يحتوي معلومات admin.py ثم تعديل يحتوي معلوماته يمكن تنفيذ ذلك عن طريق admin.py ثم تعديل التالى:

```
class PublisherAdmin(admin.ModelAdmin):
    list_display = ('name', 'website', 'email')
admin.site.register(Publisher, PublisherAdmin)
```

- كتابة class PublisherAdmin
- كتابة list_display مع إضافة fields التي نريد ظهور ها في الجدول.
 - تعدیل register بإضافة PublisherAdmin
 - قم بالدخول على الصفحة سوف تلاحظ أن الواجهة تغيرت كالتالى:

Select publisher to	ADD PUBLISHER +	
Action:	Go 0 of 2 selected	
NAME	WEBSITE	EMAIL
☐ Publisher_B	https://www.publisherb.com	Publisher_B@example.com
☐ Publisher_A	https://www.Publisher_A.com	Publisher_A2@gmail.com
2 publishers		

تخصيص واجهات Admin باستخدام

إحدى الطرق أيضا لعمل تخصيص على ModelAdmin عن طريق list_filter وهي:

- تسهل علينا الوصول للمعلومات عندما يكون لدينا معلومات كثيرة.
- تشبه filter الظاهر في الجانب (by staff status, by superuser status, and active

بداخل admin.py نقوم بكتابة التالى:

```
class MoviesAdmin(admin.ModelAdmin):
    list_filter = ('publisher',)
```

admin.site.register(Movies_Info, MoviesAdmin)

نقوم بتشغيل الموقع و نلاحظ أن الواجهة تغيرت وتم إضافة filter بناء على publisher كمافي الشكل التالي:

	Q ± &
	WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Select movies_ info to change	ADD MOVIES_INFO +
Action: Go 0 of 1 selected	FILTER
☐ MOVIES_INFO	By publisher
☐ Movie1	All Publisher_A
1 movies_ info	Publisher_B

الإسم الظاهر لنا في Filter هو اسم publisher وهو يمثل ماكتبناه عن طريق ميثود () _str_ الموجودة في Publisher الإسم الظاهر لنا في model