

## How to do CRUD operations on tables using Admin site:

Django provides an admin site to allow CRUD (Create Read Update Delete) operations on registered app model.

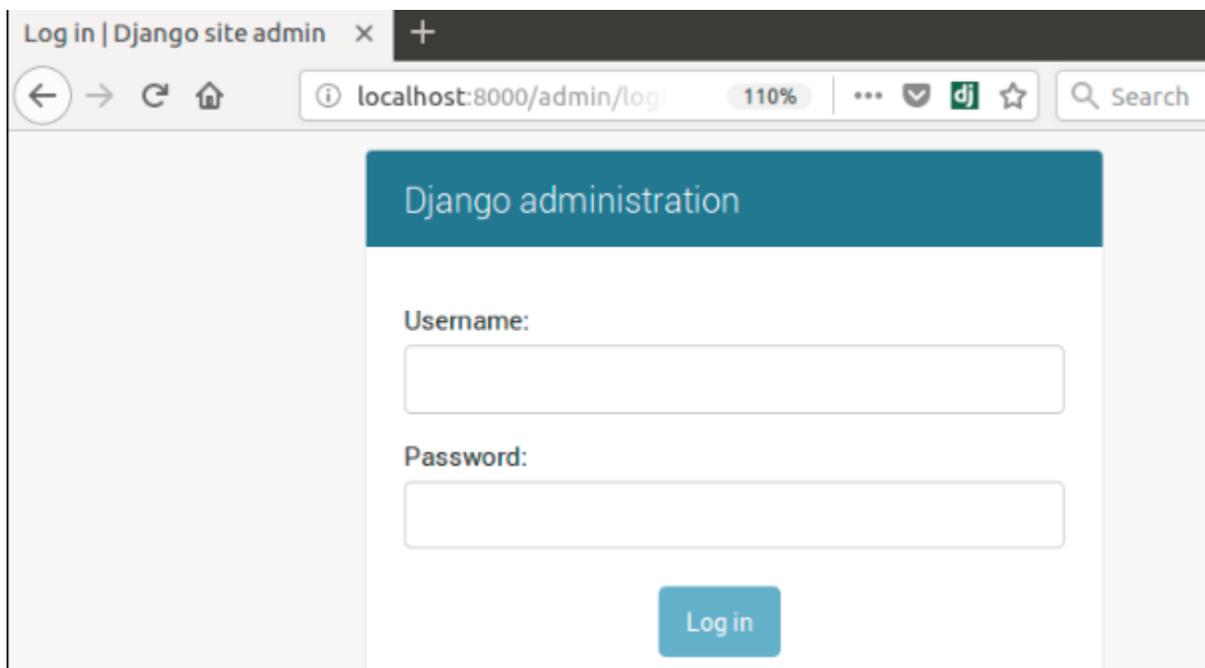
It is a built-in feature of Django that automatically generates interface for models.

We can see the url entry for admin in urls.py file, it is implicit and generated while creating a new project.

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

It can be easily accessed by after login from the admin panel, lets run the server **python manage.py runserver** and access it through the **localhost:8000/admin**.

A login form will be displayed, see the below.



The screenshot shows a web browser window with the title "Log in | Django site admin". The address bar displays "localhost:8000/admin/login". The page content includes a teal header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A teal "Log in" button is positioned below the password field.

To login, first create admin (super user) user and provide password as we did here:

```
PS D:\Django@7.30AM\FirstProject> python manage.py createsuperuser
Username (leave blank to use 'lenovo'): Virat
Email address: virat@gmail.com
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
Super user is created successfully, now login.
```

**Note:** Now storing these credentials in to one table which name as "**auth\_user**" table.

127.0.0.1:5858/admin/login/?next=/admin/

## Django administration

Username:

Password:

[Log in](#)

It shows a home page after successfully login, see below.

127.0.0.1:5858/admin/

## Django administration

WELCOME, **VIRAT**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION	
Groups	<a href="#">+ Add</a> <a href="#">Change</a>
Users	<a href="#">+ Add</a> <a href="#">Change</a>

Recent actions

---

My actions

None available

It is an admin dashboard that provides facilities like: creating groups and users. It also used to manage the models.

### Register Django Model

- By default our application models related tables are not loaded into admin site.
- If you want to load our model related tables on admin site then register our models with admin site.
- To load our models with admin site, open "admin.py" file which is available inside our application name and follow the below procedure,

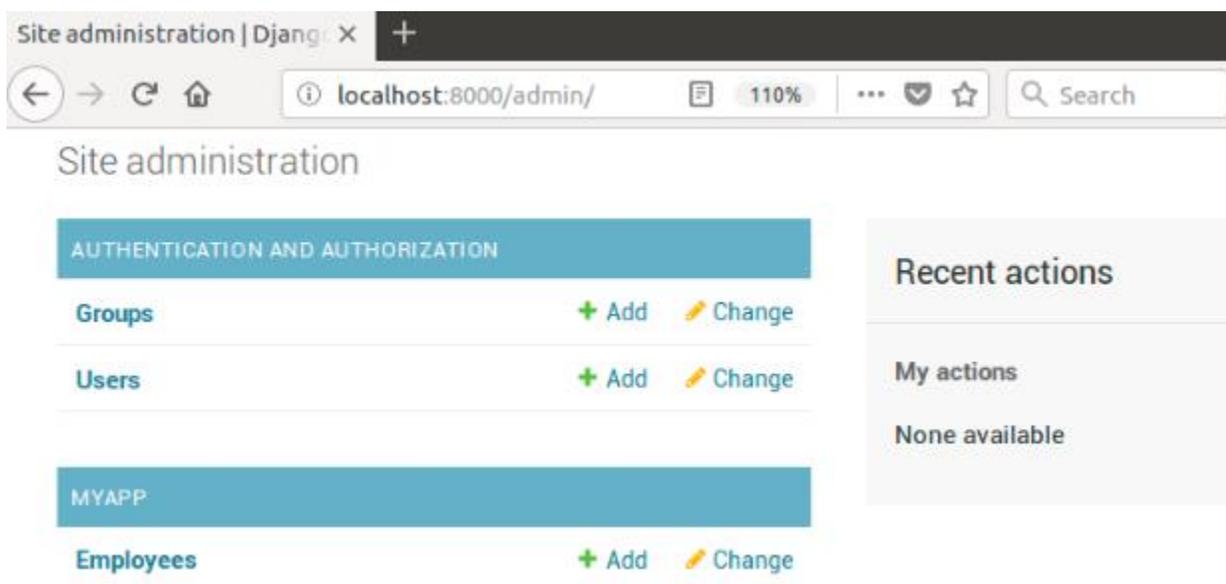
#### Procedure:

1. Import the admin module. (it is available by default)
2. Import the our required userdefined models here.
3. Create the user defined admin class by using predefined admin class like "ModelAdmin".
4. Register our model class names and user defined admin class names with admin site.

To register model in `admin.py` file. Use the `admin.site.register()` method and pass the Model name. See the example.

```
// admin.py
from django.contrib import admin
from myapp.models import Employee
admin.site.register(Employee) # Employee is registered
```

Login again and see, it has **employee** object.



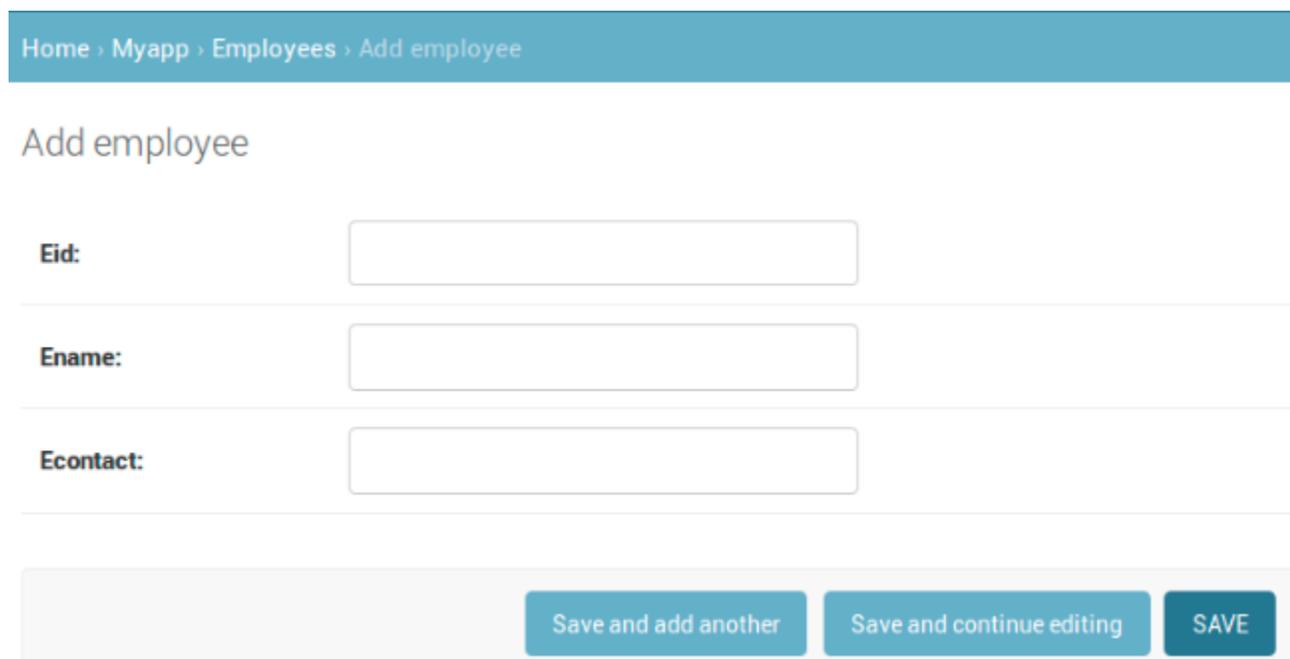
It provides auto generated interface to create new model object. Like, if i click on **add**, it renders a form with all the attributes provided in the model class.

For example, our model class contains the following code.

```
// models.py
```

```
from django.db import models
class Employee(models.Model):
    eid = models.CharField(max_length=20)
    ename = models.CharField(max_length=100)
    econtact = models.CharField(max_length=15)
    class Meta:
        db_table = "employee"
```

The auto generated form will be based on the model. We don't need to write HTML to create form. The form looks like this:



Home › Myapp › Employees › Add employee

### Add employee

**Eid:**

**Ename:**

**Econtact:**

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Lets add an employee by providing details and click on save button.

## Add employee

**Eid:**

**Ename:**

**Econtact:**

Save and add another

Save and continue editing

SAVE

After saving, record is stored into the database table, see the below MySQL table.

Using this admin dashboard, we can update and delete record also.