**Creating our fist mini project with Models**

1. **Create new directory**
2. **Create venv (virtual enviroment) in this folder and activate it**
3. **Using pip install Django**
4. **Start new Django project using django-admin startproject store**
5. **Create new application in this project from root directory (where manage.py file is) with python manage.py startapp animals**
6. **Validate everything is working by running python manage.py runserver**

**It is very important to make migrations and apply them so:**

**Python manage.py makemigrations**

**Python manage.py migrate**

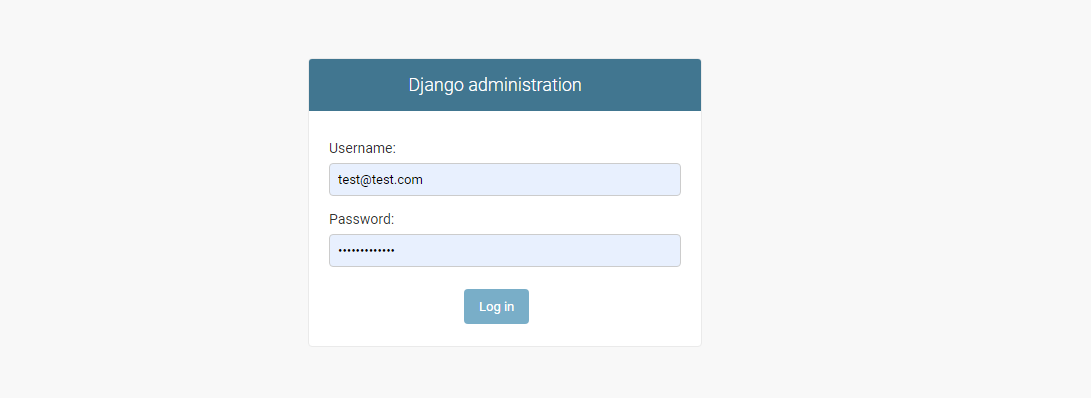
**In order to have access to our beautiful admin panel we must create our first user, but not an ordinary user – it will be super user**

**Python manage.py createsuperuser**

**The cmd will ask you 4 simple questions username, email, password and repeat password, you can leave the fisrt two blank by hitting ‘enter’ and the passwords field – just choose something easy to remember and to write, because we will use that a lot.**

**Run the server again and try to hit the following url -** <http://127.0.0.1:8000/admin/>

You will see that:

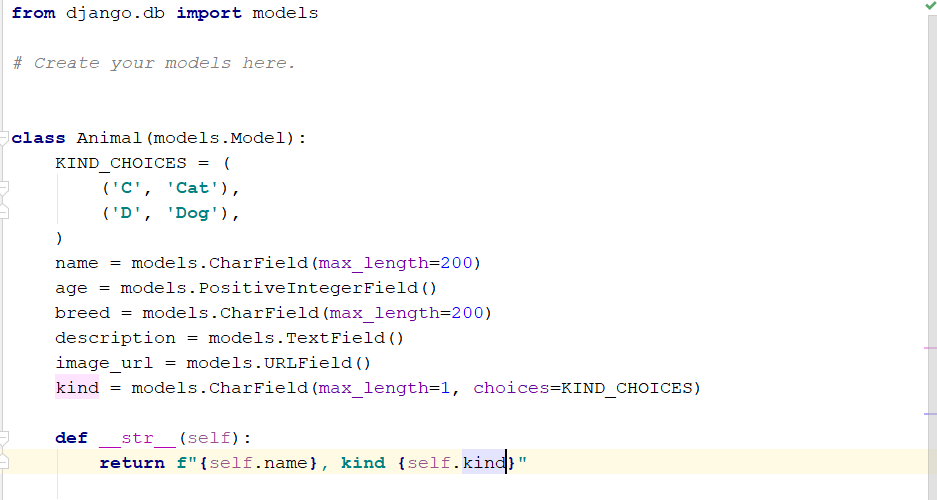
****

**NOTE!** If you haven’t run the migrate command do it now.

Try to login with your username and password. You will see only users and groups. That is normal and they are by default provided from Django after first (initial) migrate.

Let’s go and create our first model called animal.

1.Go to models.py in animals app and do the following:



It is very important to tell your Django project that animals app exists. Go to settings and in INSTALLED\_APPS add bellow ‘animals’. Then run makemigratios and migrate again.

Go to animals/admin.py and register the model

Admin.site.register(Animal), but first import it on the top.

Go to admin panel and create at least 5 animals of your choice.

After we have enough data let’s play with it with Django ORM

USING DJANGO ORM

In order to see some results let’s create a few urls and a few functions.

Configure first your main urls to work with urls of your animal app by using include function. After that create urls.py file in animals and add some urls.

1.Filter all dogs

2.Filter all cats

3.Get all animals

4.Get specific animal (by id/pk)

5.Filter by name with querystring through url

**BASIC CRUD operations**

We are going to implement a basic CRUD;

You are going to need a couple of urls here

For creating object;

For reading object/s

For updating (editing) object

For deleting objects

**HINTS:**

**1.When calling a view for animal/s make regex for one url to match exact one function and handle logic there for one or multiple animals**

**2.For creating url keep in mind we just assume that all the fields will be there**

**3. for updating -> update just the name of the animals**

**4. for deleting -> make url be like /delete/<int:animal\_id> and in function check first if we have object with such an id after that delete it or return response that this object does not exist.**

Relations

1. Implement class Owner
2. Think about the relation between two tables – animals and owners? Should be in animal or owner?
3. Implement it and migrate
4. Go to admin panel and see the differences – you have to have the opportunity to select one owner for each animal. Each animal must have only one owner

**IMPORTANT – keep in mind that this exercise is just for getting used to urls params and methods, basic iteraction with the DB -> we are not following the best practices here -> we are going to refactor this lab next time when we are going to implement templates and return them instead returning simple HttpResponses with strings or jsons**