Creating a table in a database using Django. (Windows users)

Below are the Steps to follow when creating a table in a Django web application.

Requirements;

1. VS code
2. Python 3 and above
3. Bootstrap files(JS and CSS)

Step 1: Create the directory which will be storing your project files and web app files.

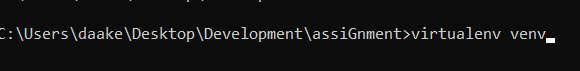
Step 2: Open the directory in CMD and install the virtual environment in which Django will be running.

i.e



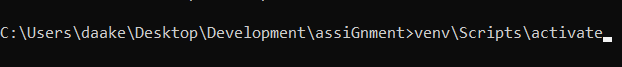
Step 3: Name your virtual environment.

i.e



Step 4: Activate the virtual environment.

i.e



Step 5: Install Django

i.e



Step 6: Start your project

i.e



Step 7: move into the project “tables” created

i.e



Step 8: Migrate Django databases

i.e



Step 9: Create a superuser which will be the admin user that will be used to edit the tables created.

i.e



Step 10: Create the app

i.e



Step 11: run the server

i.e

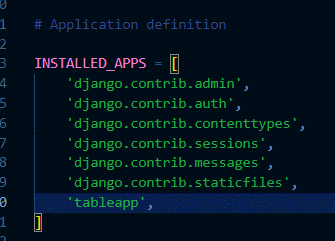


The next steps will be executed in your visual studio code Application.

Step 12: Open the directory in your Vscode application

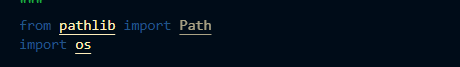
Step 13: In your settingings.py file within your project duplicate directory, add the app name to the INSTALLED\_APPS list

i.e



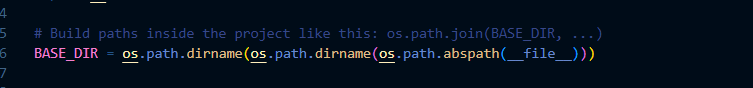
Step 14: import “os” in the settings file

i.e



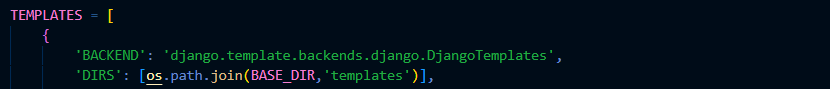
Step 15: Edit the variable BASE\_DIR

i.e



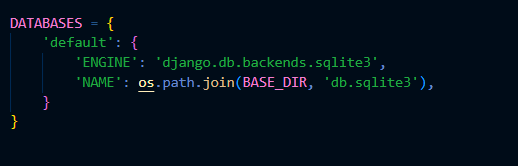
Step 16: In the TEMPLATES list edit the dict DIRS embedding a list to handle template requests.

i.e



Step 17: in DATABASE dict and NAME

i.e



Step 18: Below the variable STATIC\_URL

Create variables STATIC\_ROOT and STATICFILES\_DIR to handle Static file requests.

i.e



Step 19: Create a directory in your app directory called static which will store the CSS, JS, and image files.

Step 20: Create a directory in your original project file and in it create another directory that will store all the html files.

Step 21: in the url.py file within the duplicate project file, import include from django.urls

i.e



Step 22: Add to the urlpartterns list add a path directing url requests except ‘admin/’ to another url.py file you will create in your app directory.

i.e



Step 23: Create a url.py file in the app directory.

Step 24: Import path from django.url

i.e



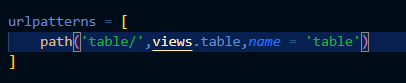
Step 25: Import views from your app

i.e



Step 26: Create a list urlpatterns and add a path that will handle your table url request

i.e



Step 27: Create a file named models.py in your app directory.

Step 28: In the models.py file import models from django.db

i.e



Step 29: Creating the tables with fields

Information will be added to the tables the admin side.

Classes are used and the classes are the table names.

The fields are created by adding Features to the Class.

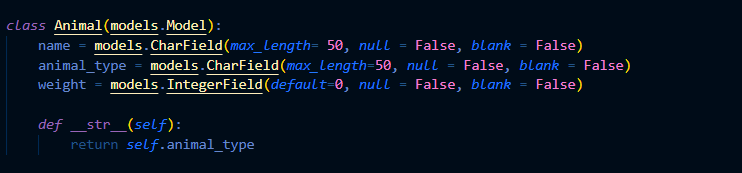
CharField is used for strings.

IntegerField is used for digits.

If you want the fields to be compulsory you add “null = False” and “blank = False” to the parenthesis.

In this case, table named Animal is created.

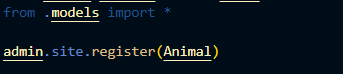
i.e



Step 30: In the app directory, open the admin.py file and import all the Classes from the models.py file.

Add the Class name.

i.e



Step 31: Back to the CMD, make migrations to add the new changes in the model then migrate.

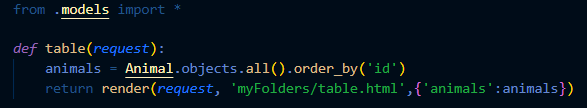




Step 32: Back to the VS code Application, in the views.py file import all the Classes From the models .py file.

Define a function that will handle the url requests

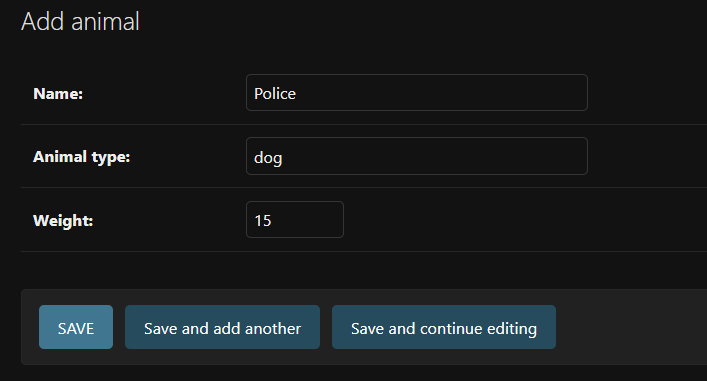
i.e



Step 33: Run the server on the CMD then log in to the Django admin page “127.0.0.1:8000/admin” using the details from the superuser.

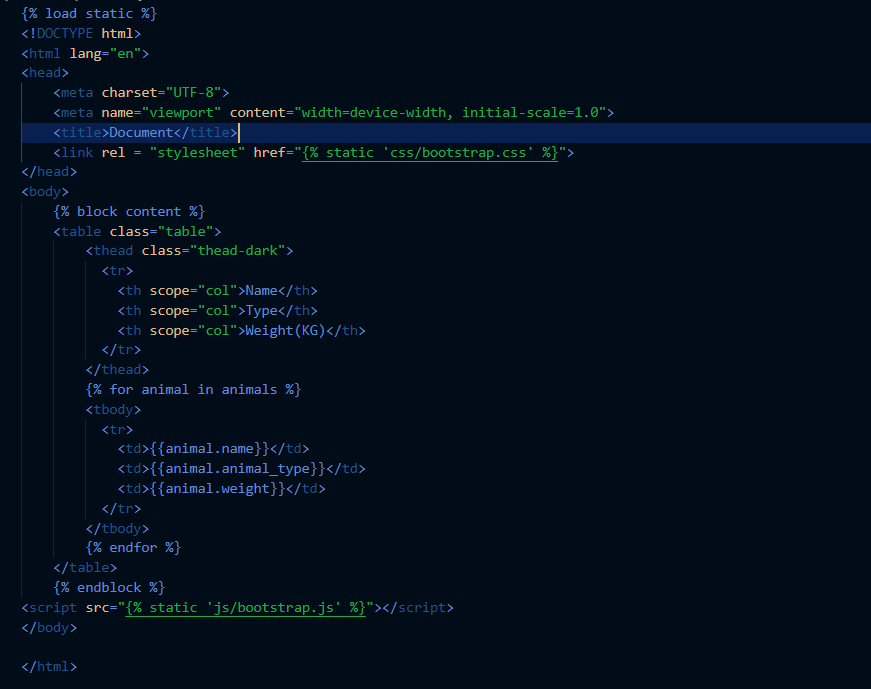
Add data to the fields in the table ”Animal”

i.e

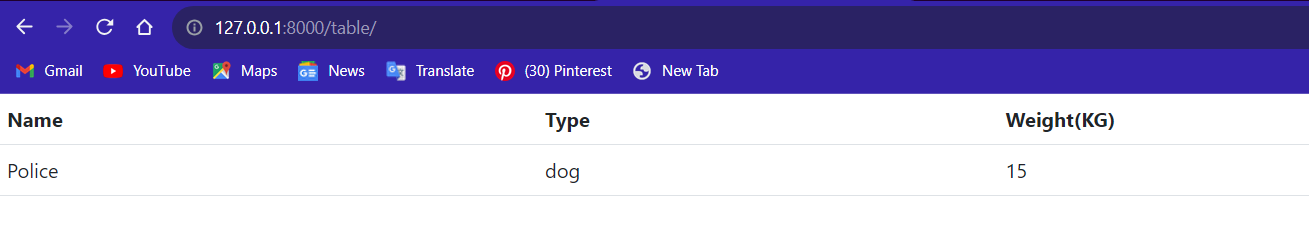


Step 34: In the table.html file, using the table tags create a table with the table heading and the table data sourcing using Django templating language to fill the data automatically.

i.e



Finally, your table should be reflecting the data entered from the admin side and should look roughly like this,



The rest is just styling.