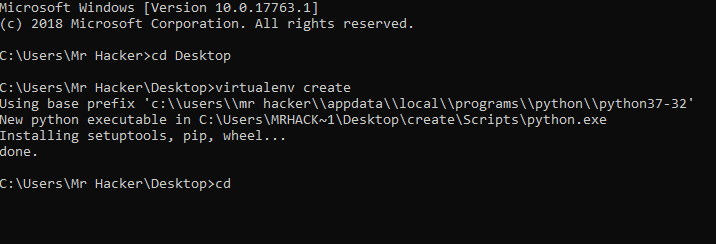
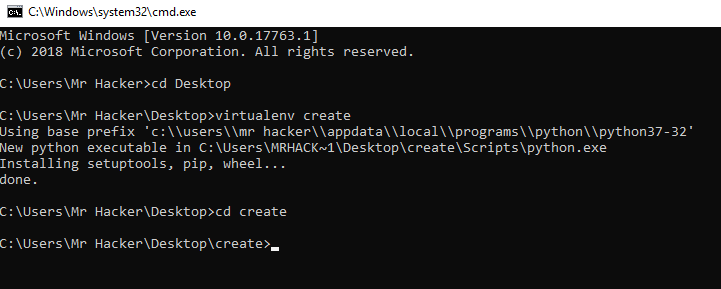
**virtualenv create (folder name can be anything)**

**why we use virtualenv????**

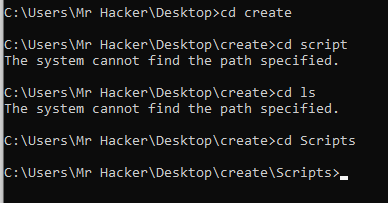
Virtual environment will help us to run the same code if the version is changed or upgraded, generally one version code faces some difficulties in another version. Virtual environment helps us to remove the version dependency. So whenever we will create a project we should use virtual env to remove version dependencies.

****

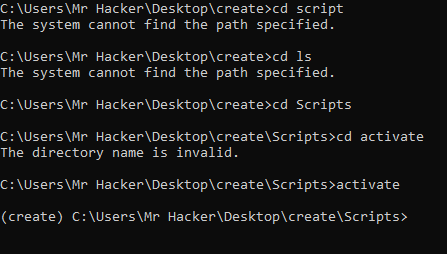
**Cd create**

****

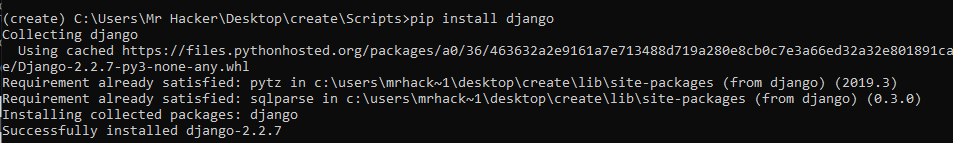
**Cd Script**

****

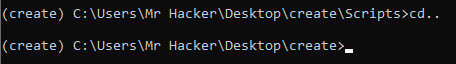
**activate**

****

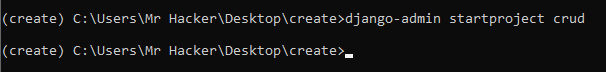
**pip install django**



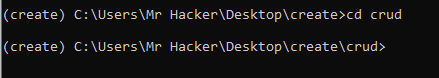
**cd..**



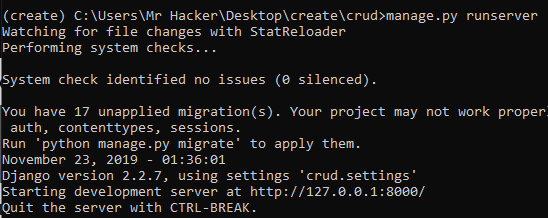
**django-admin startproject crud**



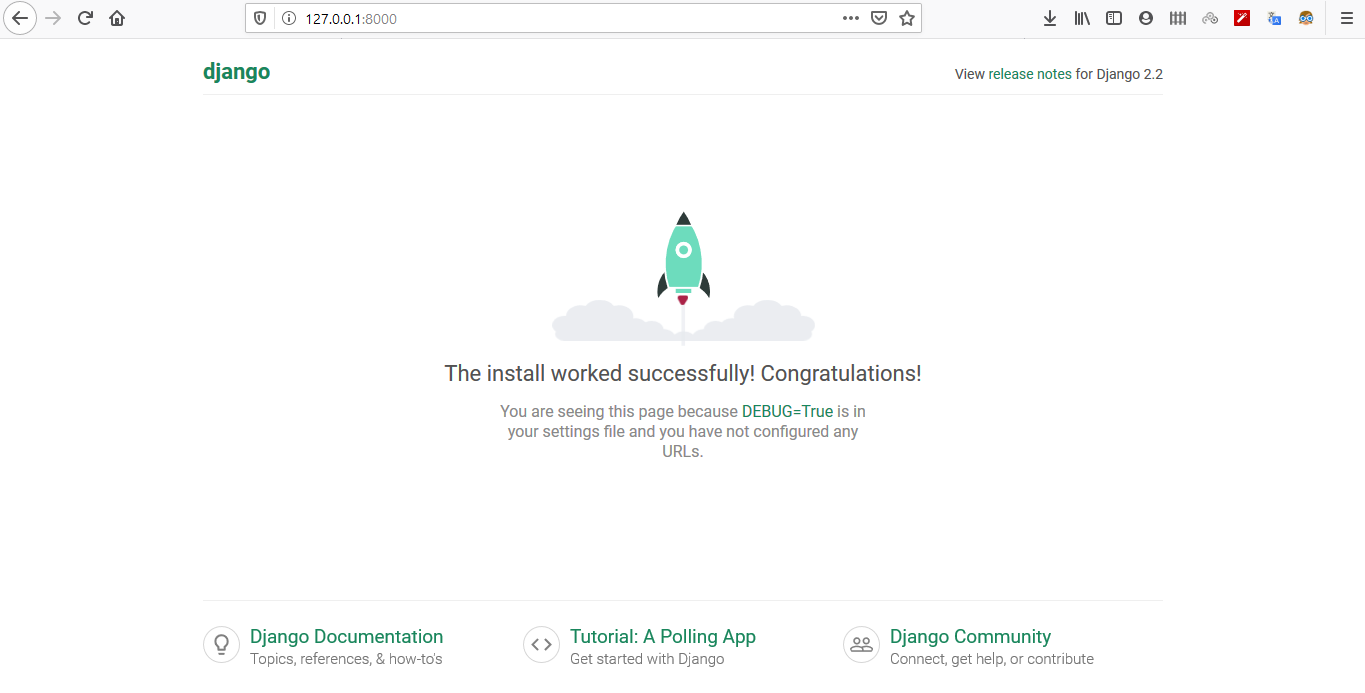
**cd crud**



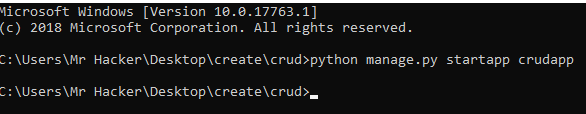
**manage.py runserver**

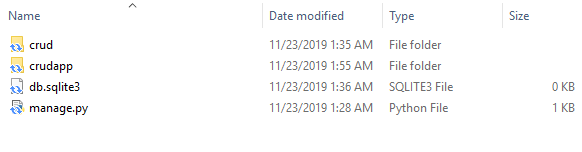


[**http://127.0.0.1:8000/**](http://127.0.0.1:8000/)



**python manage.py startapp crudapp**



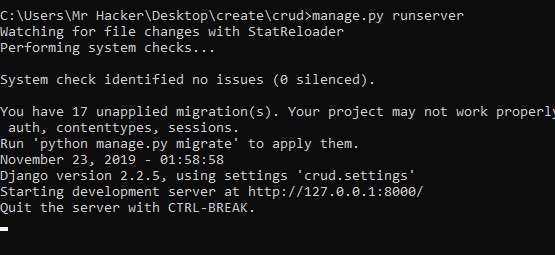


**\*\*\*\*\*\*output\*\*\*\*\***

**Now go to settings.py file and add created app name in the INSTALLED\_APPS list (Line 33)**



**manage.py runserver**



**Now go to views.py file to create a view. And then add**

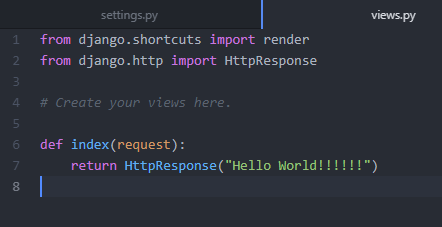
**from django.shortcuts import render**

**from django.http import HttpResponse**

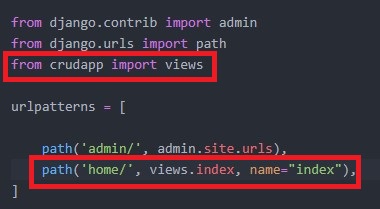
**# Create your views here.**

**def index(request):**

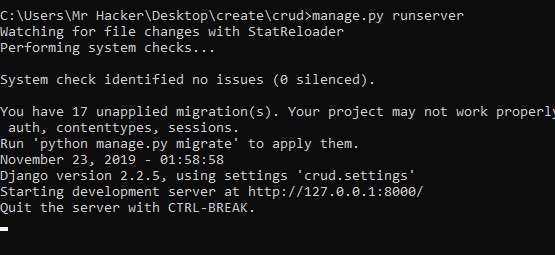
**return HttpResponse("Hello World!!!!!!")**



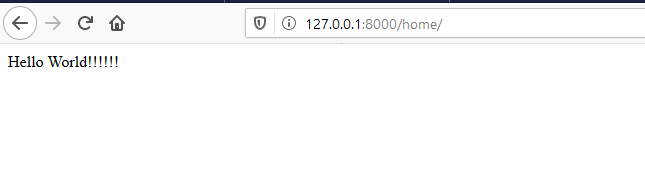
**We just create a view in views.py and now go to urls.py and set thr URL to see the views in running server. Also need to import views from existing app**

****

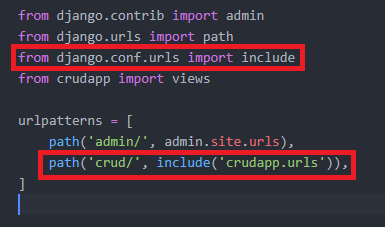
**manage.py runserver**



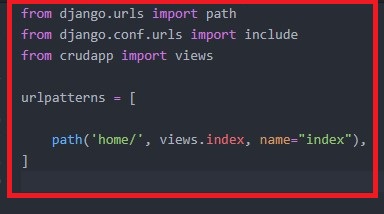
**View is working now**



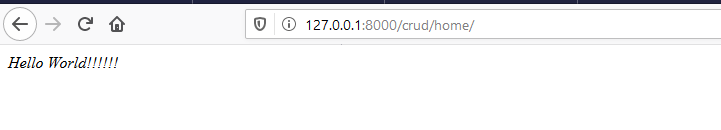
**Now we are going to map all URL for our project. We will create a new URL file inside the app folder and then add the file location to urls.py file which inside the project folder.**

****

**New urls.py file inside the app folder**

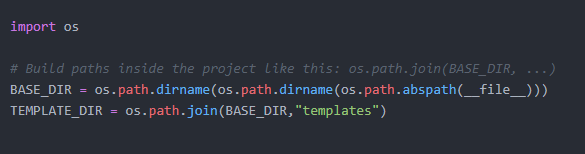
****

**And the view**

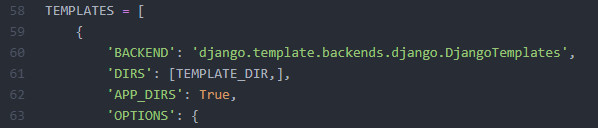


**Create a new variable for template folder inside the settings.py file.**

**TEMPLATE\_DIR = os.path.join(BASE\_DIR,"templates")**



**And then add this variable inside below of the settings.py file**

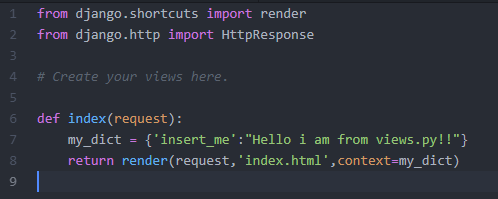


**Now in the views.py file create a view and return render the view to the index.html file**

**def index(request):**

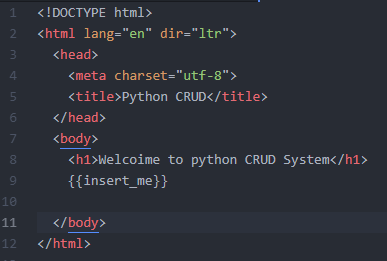
**my\_dict = {'insert\_me':"Hello i am from views.py!!"}**

**return render(request,'index.html',context=my\_dict)**



**Now create a index.html file inside the template folder. And then call the variable inside the html file.**

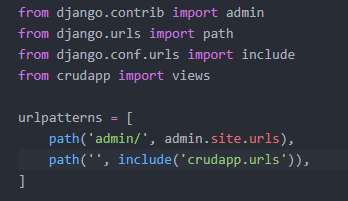
**{{insert\_me}}**

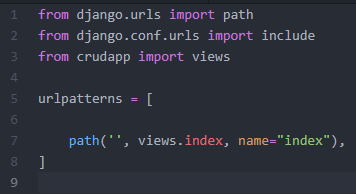


**Now check the website. Boom!!!**

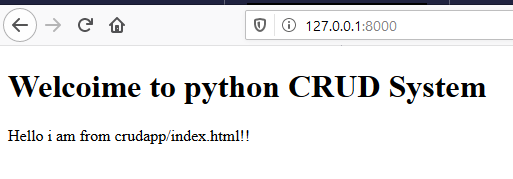


**Go to project folder and then go to urls.py >> remove path first parameter or value then go to app folder and remove as same.**

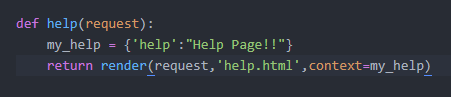




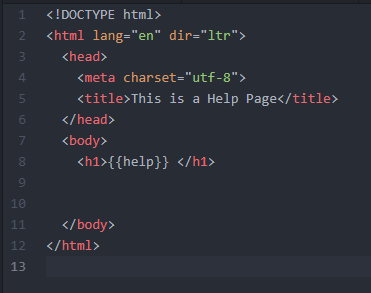
**Now hit to the IP**



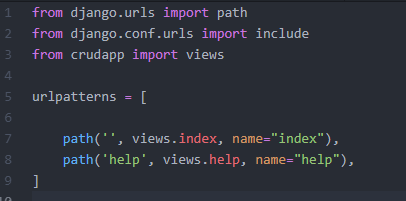
**Now go again to views.py file and create a new view help**



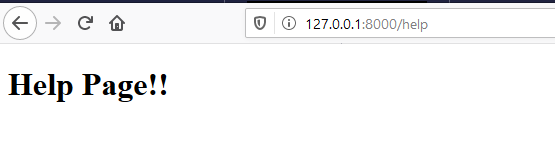
**Create a help.html file inside the template folder.**



**Now go to the app folder and change the URL mapping into like this**



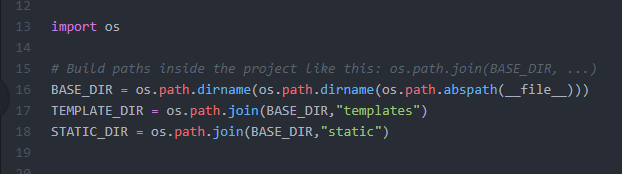
**Boom hit the ip/help**



# Static

Create a New folder (static) inside the project folder and create a (images) folder inside the static folder and then go to settings file to add the line below the base directory.

STATIC\_DIR = os.path.join(BASE\_DIR,"static")

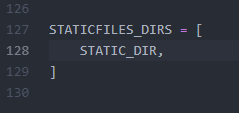


Now add a variable STATICFILES\_DIRS

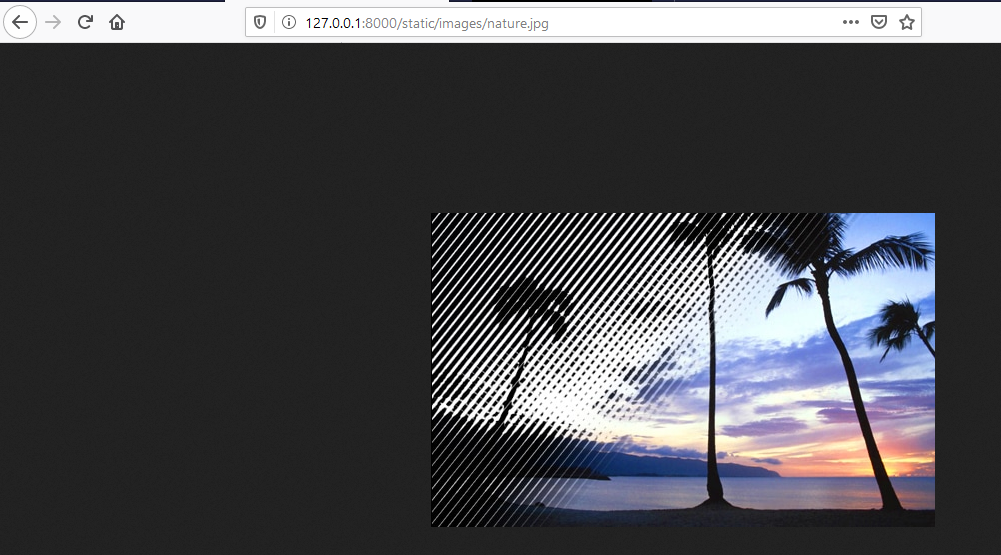
STATICFILES\_DIRS = [

STATIC\_DIR,

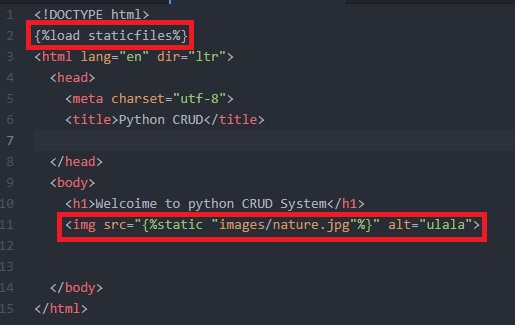
]



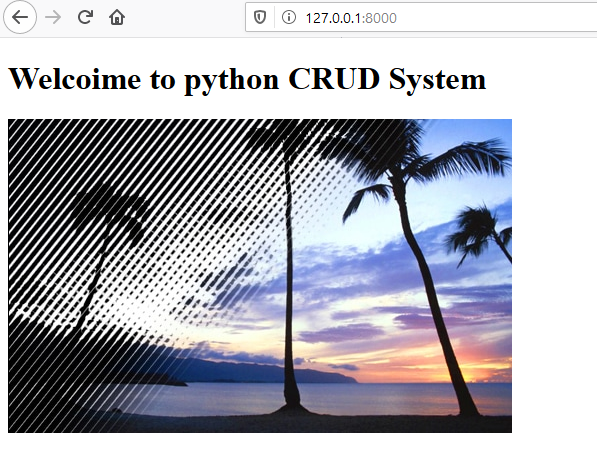
Now save a image inside the images folder and run the server and visit the URL



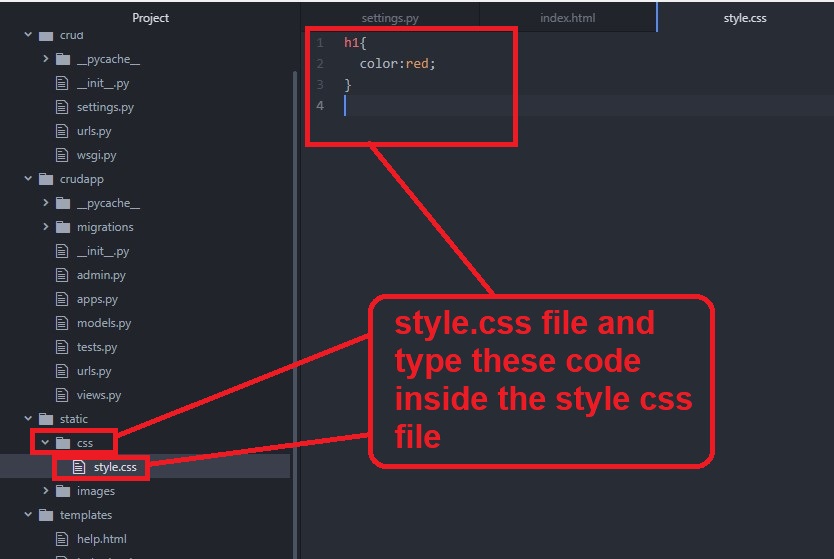
Now go to the index.html file and link the image to the index page.



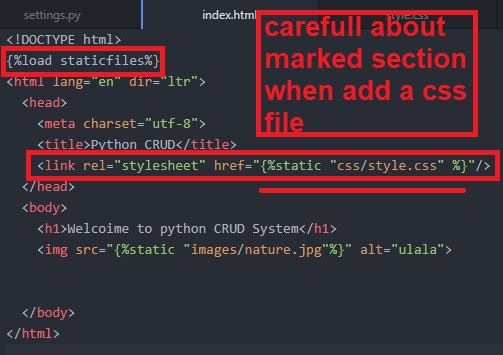
Output of the interface



# How to add CSS in django



And then add the CSS file inside the index.html file.



CSS output

