A Django Project consists of one or more Django Applications.

First a project is created and then one more applications are created inside that project.

#1. Creating our first Django Project through Command Line

1. Create a folder any drive anywhere in the directory.

   E.g. Suppose I have created a folder **DjangoProjects** in D drive.

2. Go to command line and reach inside this folder.

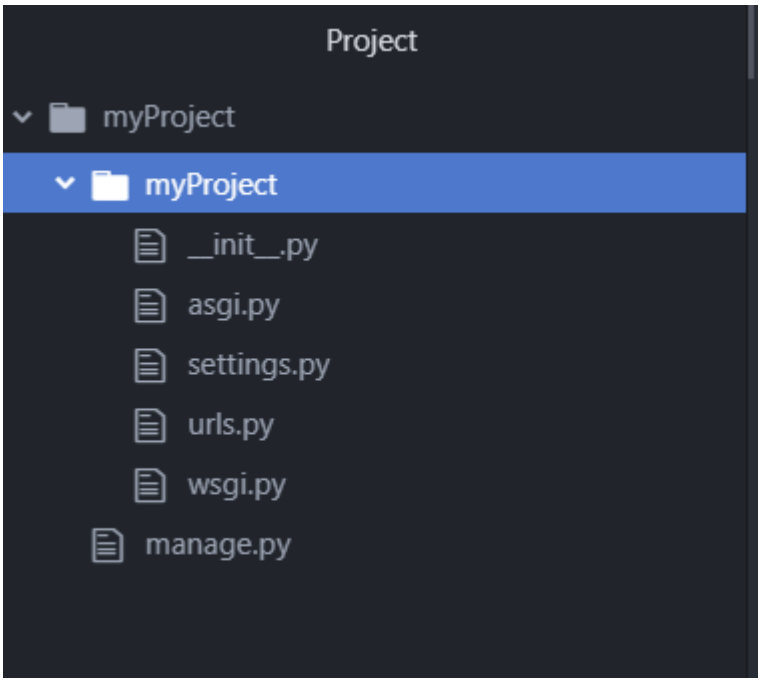
3. Use the command **django-admin startproject projectName**

   e.g. DjangoProjects> django-admin startproject myProject

4. It will create a folder having name **myProject**.

5. Open this project in any code editor like **Atom or VS Code**.

6. Look at the folder structure



a) **__init__.py**
It tells that this folder is a python package.

b) **settings.py**
All settings for the application inside this project for the middleware & also database configuration. Later on we will create applications inside this project.

c) **urls.py**
It carries the URL's pattern of our project as well as application. Mapping of views created inside application to the urls is done here.

d) **wsgi.py**
Web Server Gateway Interface. It can be used to deploy our application/project on online servers or even to the cloud.

#) **manage.py**

This is very important file as it is used to run our project and to create application inside the project.

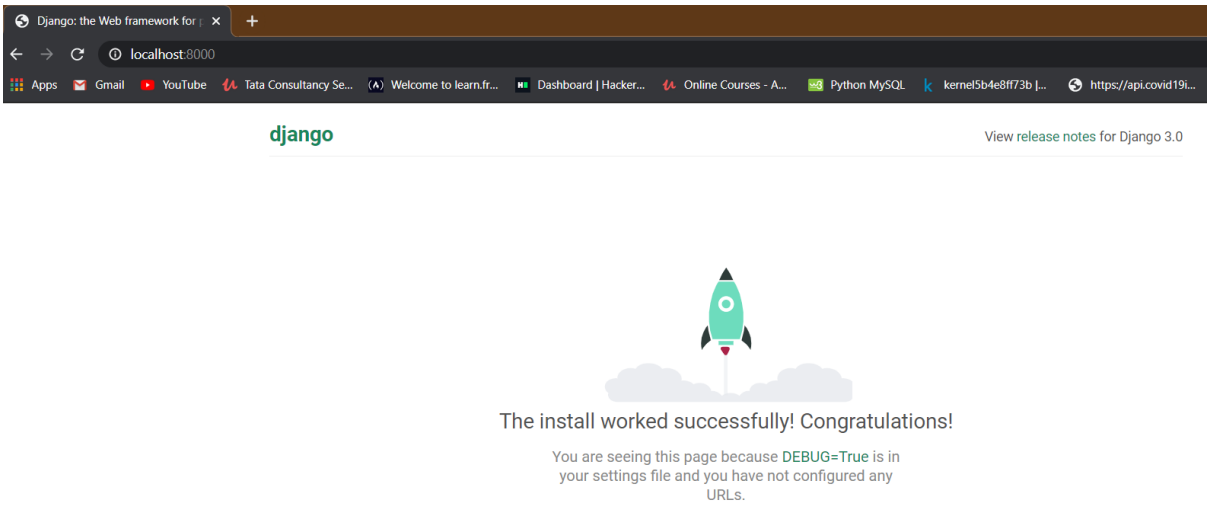
#2. Running the Project

However, we have not written any codes or configured anything, still the project is ready to be run but with default output.

1. Go inside the project i.e. **myProject**.
2. Execute the command
   myProject> python manage.py runserver

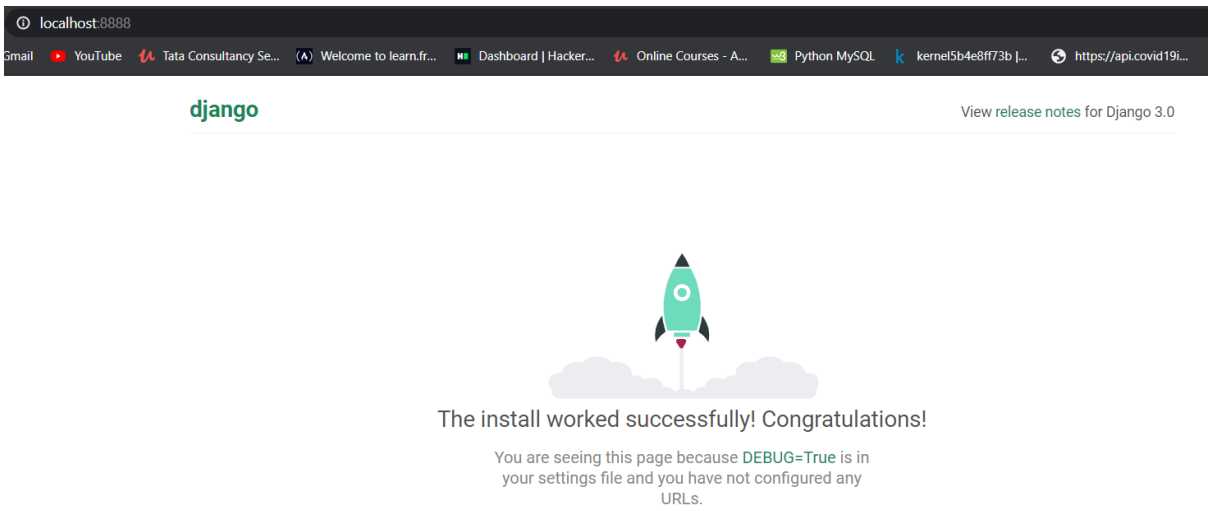
3. As python or django internally provides a default web server for us and it will launch the project on the server (**localhost:8000**).

```
D:\Projects\Udemy\Bharath Thippireddy\DjangoProjects\myProject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s)
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 24, 2020 - 22:03:03
Django version 3.0.6, using settings 'myProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



4. By default, it will be launched on **port 8000**, but it can be **changed**.

5. Changing port number
    i. Stop the server (ctrl + c)
    ii. Use the as usual command to run the project but just put the required port number at the last.
    E.g. **myProject>python manage.py runserver 8888.**



In Brief so far

Creating Project.

DjangoProjects> django-admin startproject demoProject

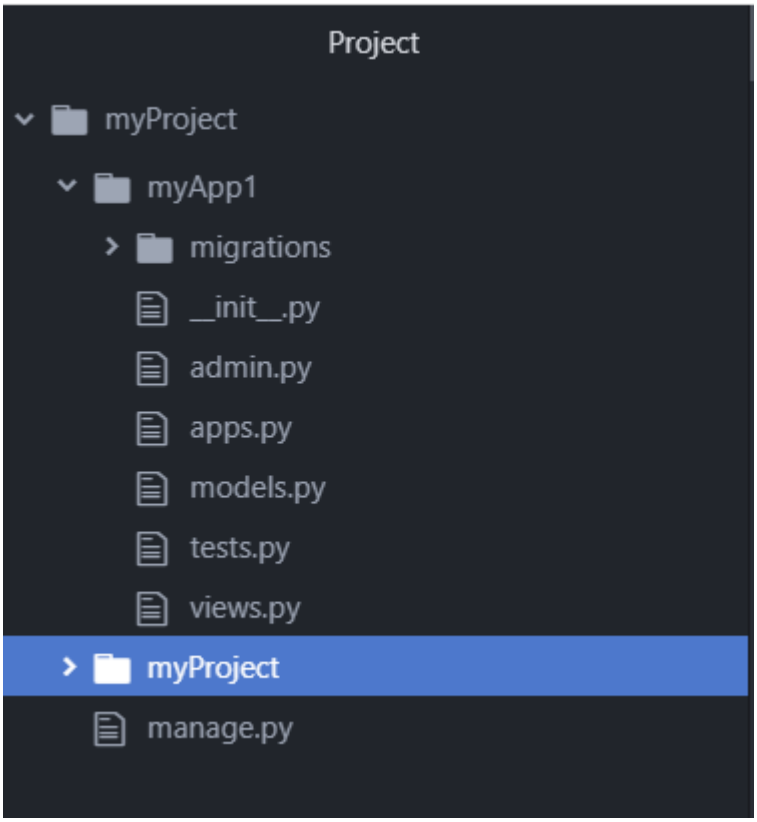
Running the Project.

1. **Get inside the project and run the command**
   demoProject>python manage.py runserver

2. **To run the project on demanded port**
   demoProject>python manage.py runserver 7777

#3. Creating application(s) inside the Project

1. Go inside the project and run the command
   myProject>**python** manage.py **startapp myApp1**
2. A folder myApp1 will be created inside the myProject
3. Look at the folder structure.



4. As we have created a new app so it needs to mentioned in the **settings**.py of the **myProject** file.

   i. **Go to settings.py**

   ii. **Go to INSTALLED_APPS**

settings.py

```
30
31  # Application definition
32
33  INSTALLED_APPS = [
34      'django.contrib.admin',
35      'django.contrib.auth',
36      'django.contrib.contenttypes',
37      'django.contrib.sessions',
38      'django.contrib.messages',
39      'django.contrib.staticfiles',
40      'myApp1'
41  ]
```

   iii. Add the name of the newly created app i.e. **myApp1** to the list at last.

   iv. We can create as many app as we need, just make sure to mention it in settings.py **Installed_Apps** list here.

#4. Creating Views inside the apps

Views mean the output or response we should get on hitting a url from the browser.

There are two ways in which we can create views.

a. Function Based Views

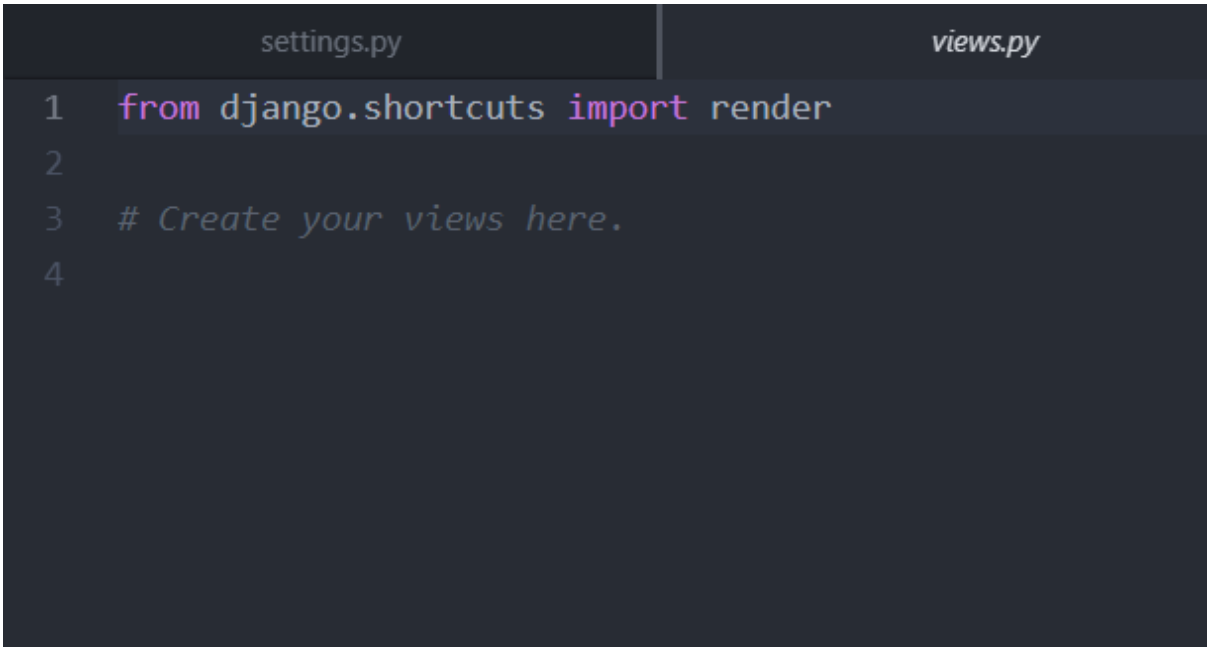
b. Class Based Views

Function Based Views

Step1 : Creating the view

A view is responsible for taking a request from the browser and sending a response back to the browser.

We have only an app i.e. myApp1, so view will be created inside that app in views.py file.



Here's the code,

settings.py views.py

```python
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def disp1(request):
    return HttpResponse("<h1>Good Morning !</h1>")

```

Step 2 : Mapping a url to this views's method(s)

We have to map this view to a unique url, so that using that url this view(method) can be invoked from the browser.

It will be done in the **urls.py** file of the myProject.

urls.py

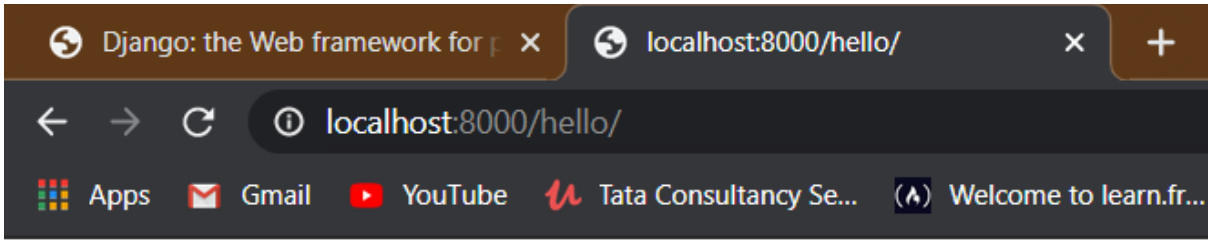
```python

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

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

```

So mapping code,

urls.py

```
1
2  from django.contrib import admin
3  from django.urls import path
4  from myApp1 import views
5
6  urlpatterns = [
7      path('admin/', admin.site.urls),
8      path('hello/',views.disp1)
9  ]
10
```

- First import that **views** in that file.
- Then add the **urlpatterns** as you can see.
- So, when a user hits the URL **localhost:8000/hello**, then due to mapping **disp1()** will be invoked and whatever it returns will be displayed on the browser.



Good Morning !