Django First Project and Fist App

```
Microsoft Windows [Version 10.0.19042.1645]
(c) Microsoft Corporation. All rights reserved.

D:\Django Github\DjangoProjects\Django02 FistApp>python -m venv env
```

```
D:\Django Github\DjangoProjects\Django02 FistApp>cd env
```

```
D:\Django Github\DjangoProjects\Django02 FistApp\env>cd Scripts
```

```
D:\Django Github\DjangoProjects\Django02 FistApp\env\Scripts>activate
```

```
(env) D:\Django Github\DjangoProjects\Django02 FistApp\env\Scripts>pip
install django
Collecting django
  Using cached Django-4.0.4-py3-none-any.whl (8.0 MB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting tzdata
  Using cached tzdata-2022.1-py2.py3-none-any.whl (339 kB)
Collecting asgiref<4,>=3.4.1
  Using cached asgiref-3.5.0-py3-none-any.whl (22 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.0 django-4.0.4 sqlparse-0.4.2 tzdata-2022.1
```

```
(env) D:\Django Github\DjangoProjects\Django02 FistApp\env\Scripts>django-
admin startproject ProjectOne
```

```
(env) D:\Django Github\DjangoProjects\Django02 FistApp\env\Scripts>cd
ProjectOne
```
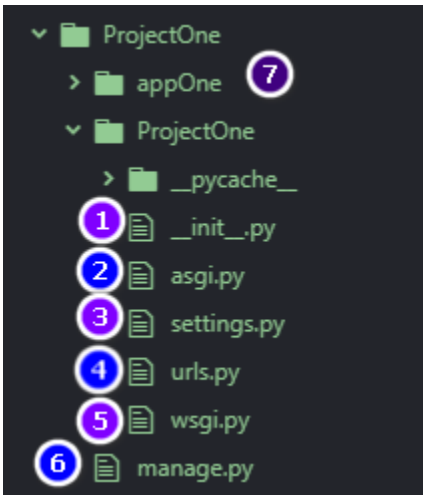
```
(env) D:\Django Github\DjangoProjects\Django02
FistApp\env\Scripts\ProjectOne>dir
 Volume in drive D is Saif
 Volume Serial Number is 5CC3-BFCE

 Directory of D:\Django Github\DjangoProjects\Django02
FistApp\env\Scripts\ProjectOne

04/24/2022  07:59 PM    <DIR>          .
04/24/2022  07:59 PM    <DIR>          ..
04/24/2022  07:59 PM               688 manage.py
04/24/2022  07:59 PM    <DIR>          ProjectOne
               1 File(s)            688 bytes
               3 Dir(s)  211,342,950,400 bytes free
```

```
(env) D:\Django Github\DjangoProjects\Django02
FistApp\env\Scripts\ProjectOne>python manage.py startapp appOne
```
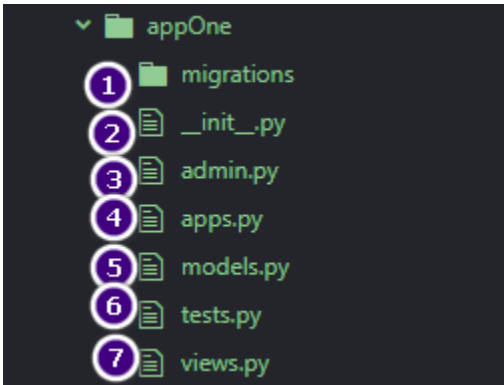
Files present in the project

__init__.py,
asgi.py,
settings.py,
urls.py,
wsgi.py
manage.py

__init__.py --> this is for python it can be treated as a package
asgi.py --> ASGI stands for Asynchronous Server Gateway Interface. It extends the capabilities of WSGI (Web Server Gateway Interface) which is a standard way of communication between the web server and the web applications in most of the python web frameworks like Django
settings.py --> this has settings for your project
urls.py --> this has the links to you project as well as function to call
wsgi.py --> this is needed if you want to perform a deployment of your project via WSGI
manage.py -->this is command line utility for executing django commands

Files present in the app

migrations folder
migrations folder is related to database where it stores models' data like adding a field, deleting a model etc.
migrations can be done using commands

__init__.py --> this is for python it can be treated as a package
admin.py   --> this helps to make the app modifiable in admin interface
app.py   --> this is a configuration file common to all django apps
models.py   --> this will store all the models in the application
tests.py   --> this has unit tests
views.py   --> this is where the application view is stored

MVT

Model View Template is a design pattern
Model provides a data access layer
It is an Interface to between models data and database data

Template is a presentation layer
It handles the User Interface

View is used to execute the business logic

---

views.py

```python
from django.shortcuts import render

# Create your views here.
from django.http import HttpResponse
import time

def d1(request):
    return HttpResponse("<h2>Welcome to Django</h2>")

def d2(request):
    t1 = time.ctime()
    return HttpResponse(t1)

def d3(request):
    msg = "Hello Django"
    return HttpResponse(msg)
```

```
urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('d1/', views.d1),
    path('d2/', views.d2),
    path('d3/', views.d3)
]
```