Django 101: Your First Steps into Web Development

What is Django?

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

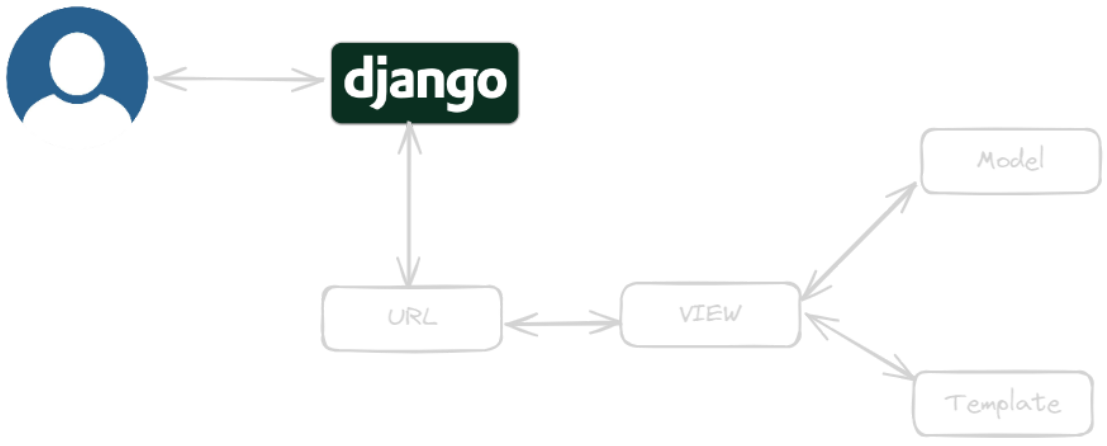
How Django Works?

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.

Although Django follows MVC pattern but maintains it?s own conventions. So, control is handled by the framework itself.

There is no separate controller and complete application is based on Model View and Template. That?s why it is called MVT application.

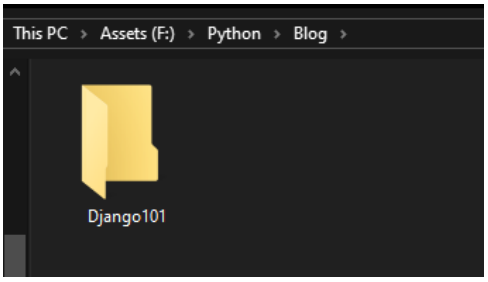


That’s how Django works !!!

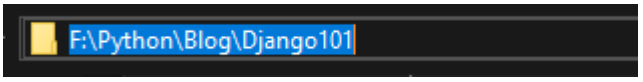
Django Installation Step by Step

It is suggested to have a dedicated virtual environment for each Django project.For the installation step first we have to create a folder in our PC/Laptop. You can give any name for your folder.

Here we are using folder name : **Django101**



Then we will go to the path of the folder:

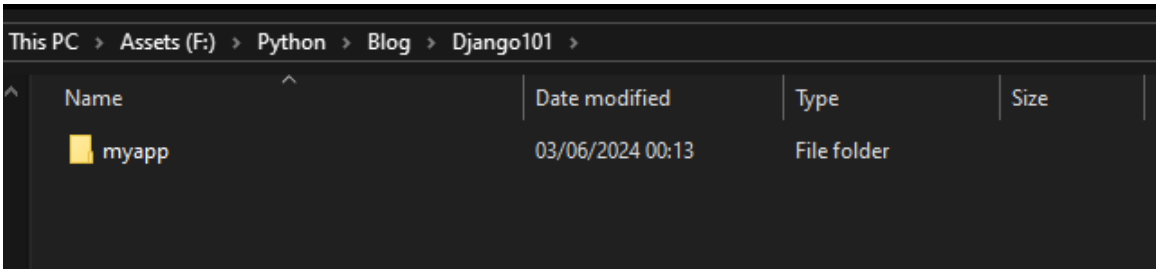


Here we will type “**cmd**” & it will open up terminal of your PC

In the terminal First we have to create a Virtual Environment.I am giving you the command step by step:

```
py -m venv myapp
```

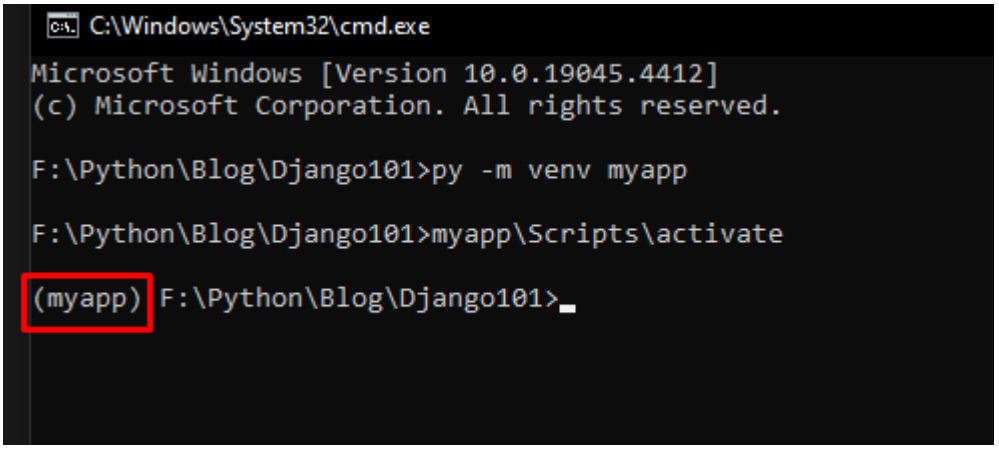
After run this command you will see a folder in your **Django101** file named **myapp**



Then you have to run a command:

```
myapp\Scripts\activate
```

After running this command you will see this name in front of your directory like this:



This means your Virtual Environment is activated now. You can start installing django now.

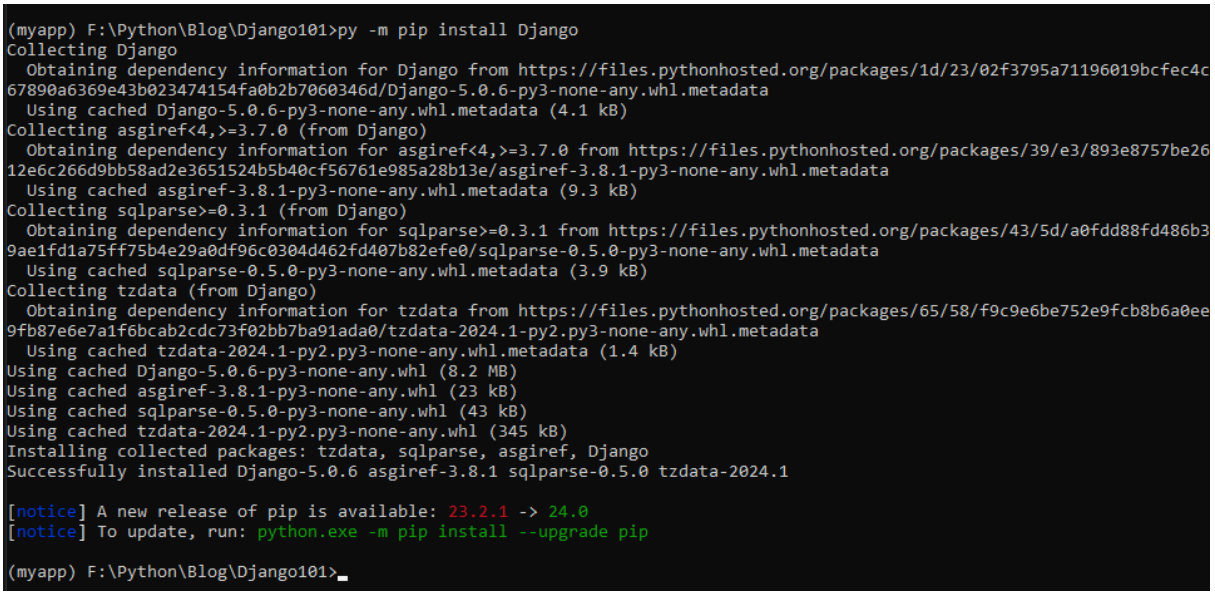
> 💡 If you are using **PyCharm** Latest version you may don't need to do this steps of creating a virtual environment. Because in pycharm it gives **Virtual Environment by default. Now** you just follow further steps.

Install Django:

Django is installed using pip, with this command:

```
py -m pip install Django
```

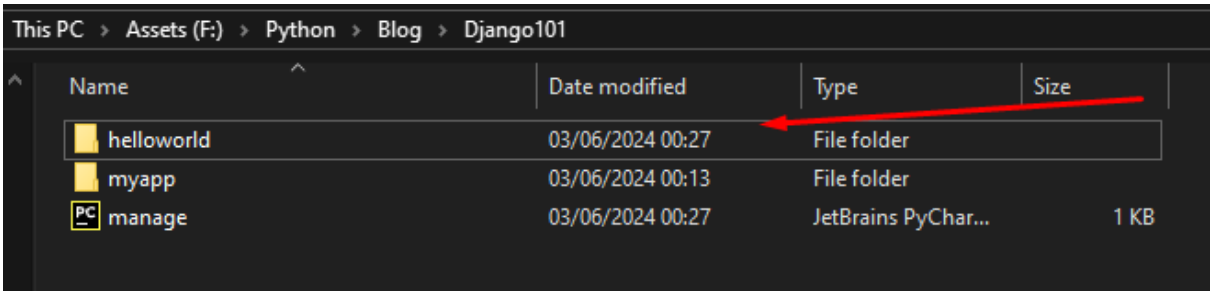
After installing it successfully then it will look something like this in your terminal:



After installing Django successfully We will **Create Project** now:

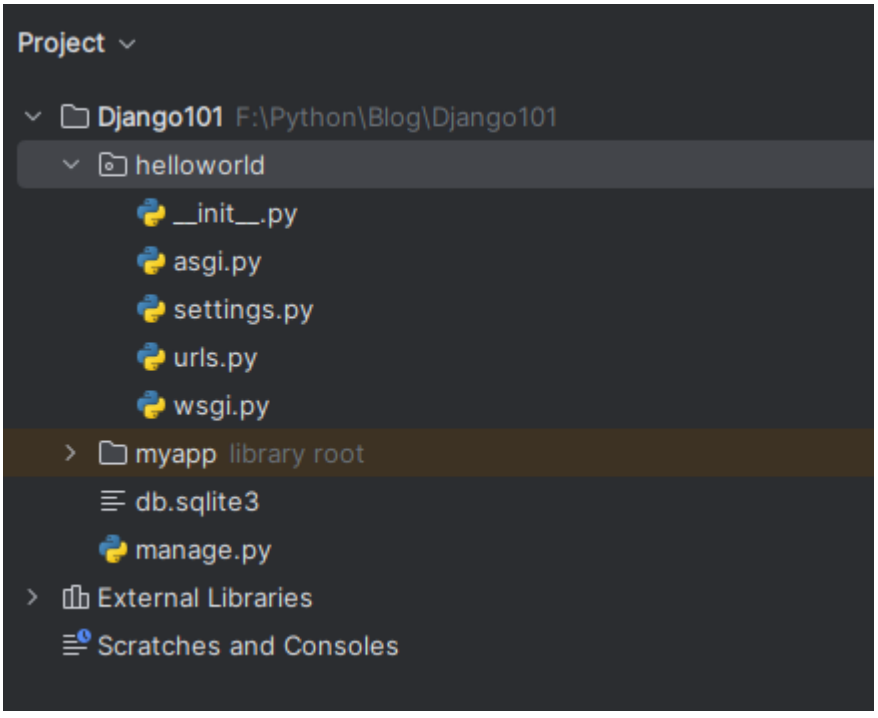
```
django-admin startproject helloworld .
```

> 💡 After run this command it will create a django project with a **manage.py** file.This is the most important file for us.Because everything we have to do here we need this file & commands.



The arrow marked file is our project which we created just now.We can open it now in VS Code or PyCharm Community Editon.

If we open the files there will be this py files:



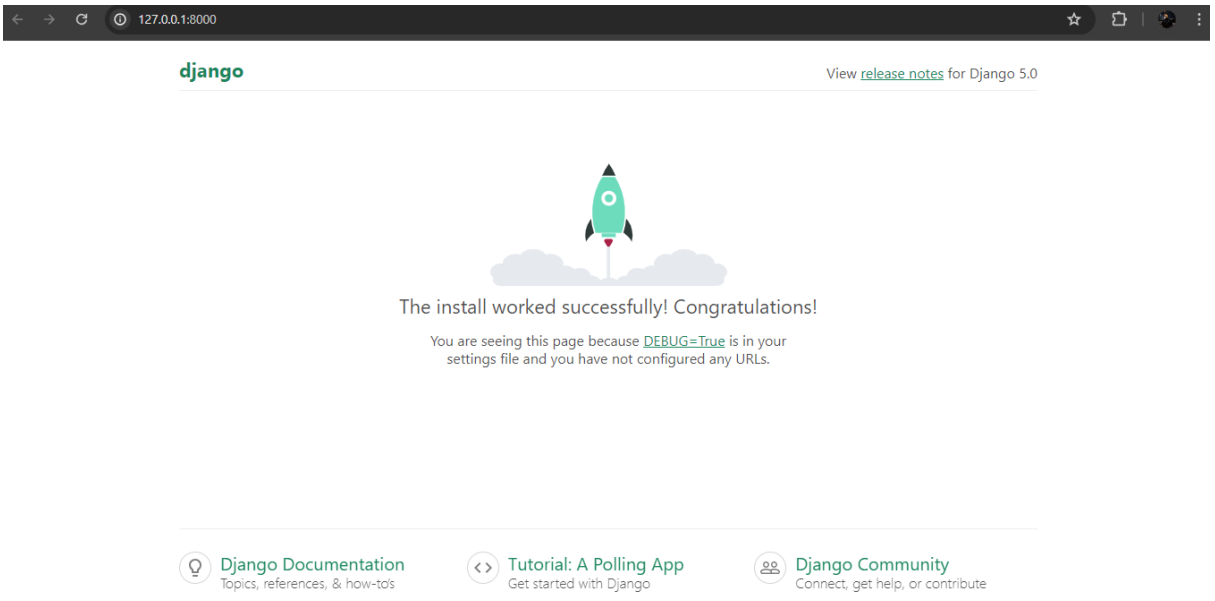
Now that you have a Django project, you can run it, and see what it looks like in a browser. To run this we have to execute a command in our terminal:

```
py manage.py runserver
```

Then it wil show this:

```
(myapp) PS F:\Python\Blog\Django101> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
June 03, 2024 - 00:34:40
Django version 5.0.6, using settings 'helloworld.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Open a new browser window and type `127.0.0.1:8000` in the address bar.Then you can see your website is running successfully:

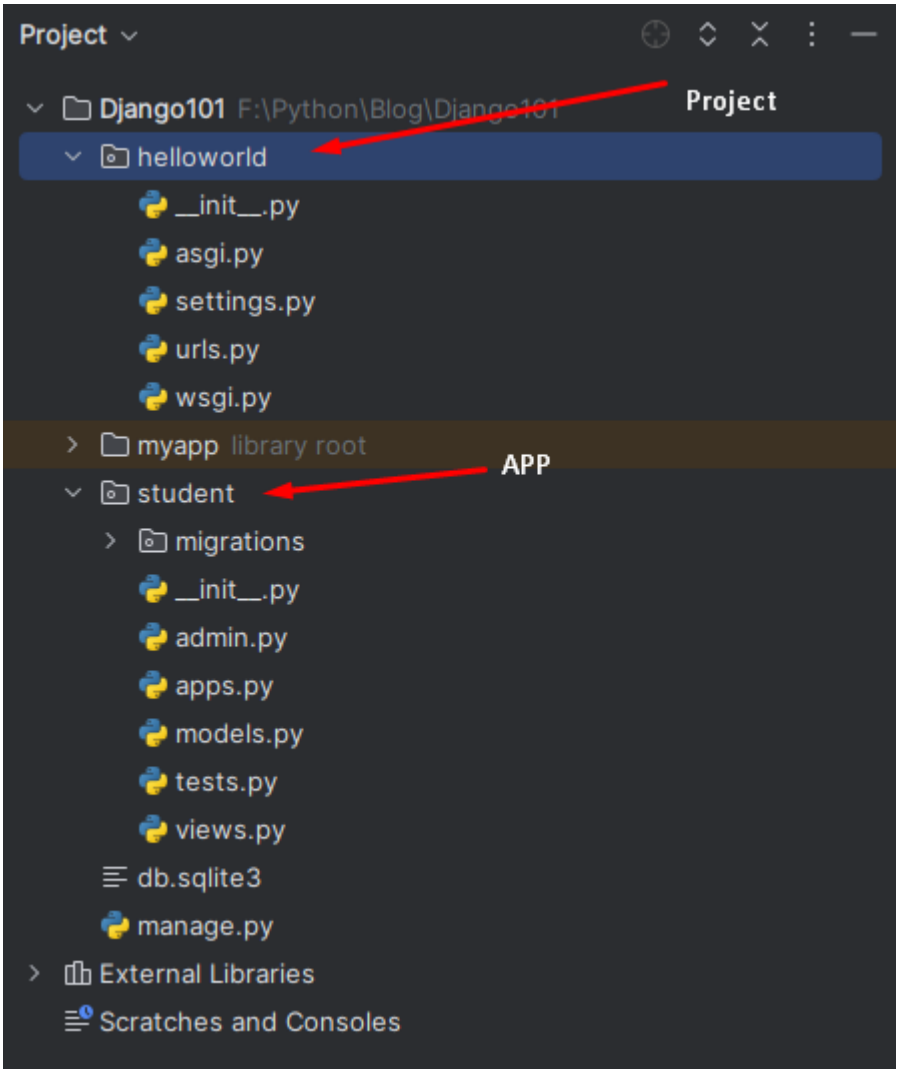


After creating project now we have to create app now:

I will name my app **student**. So we need a execute another command now for this but If the server is still running, and you are not able to write commands in terminal then, press [CTRL] [BREAK], or [CTRL] [C] to stop the server and you should be back in the virtual environment/Terminal.Then execute this command:

```
py manage.py startapp student
```

After execute this command there will be anoter folder/directory will shown up.Then the file structure will look like this:



In the app we will write our all logic & databases.We will discuss about it further.

Then we have to register this new app in our main project:

In your project directory (here is **helloworld**), Open the `settings.py` then scroll down & you will see `INSTALLED_APPS` option then there add your apps name into a quotation like this:

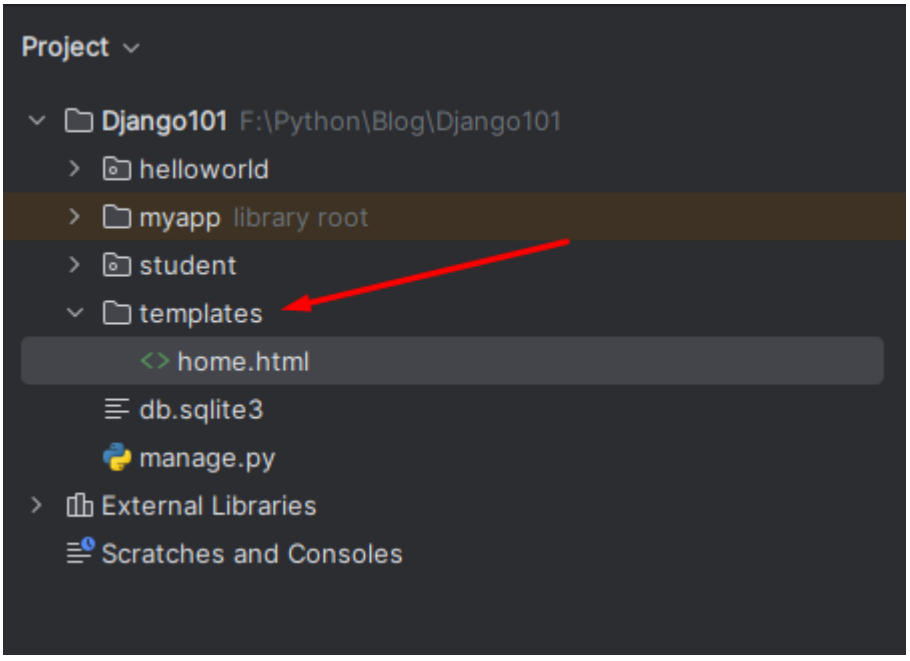
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'student'
]
```

Your app is installed on your project Now !!!!!

Now We will create template:

In the django we will see the webpage in HTML. So it should be created in a template folder so let's do that.

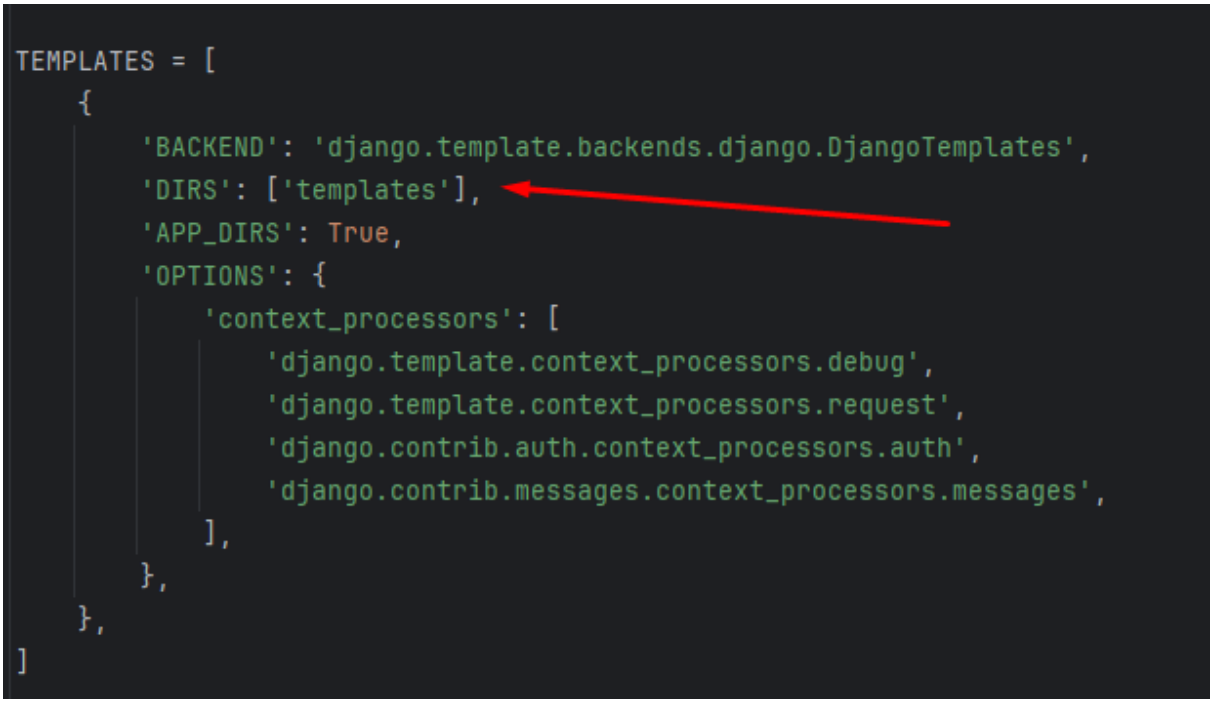
Create templates folder in your root/main directory & create a html file named "home.html"



Like this you can a make a folder & inside folder you can write any html file name you want.

Now we have to register our templates into our `settings.py` again. So you have to go your projects `settings.py` then scroll down you can see the `TEMPLATES`

option.There you have to configure your templates. Add this here `'DIRS': ['templates'],`



You are all set to go now!!!!!!!!!

Creating our first views

Go to your app file/directory then you will see a file named `views.py` .We will define functions here. This file handles all of the logics & render all templates in our project. Views are Python functions that takes http requests and returns http response, like HTML documents. It will look like this when we opens:

```
from django.shortcuts import render

# Create your views here.
```

So first we have to define our functions here & work on that .

```
from django.shortcuts import render

# Create your views here.
```

```
def home(request):
    return render(request, 'home.html')
```

This is a simple example on how to show your html in your browser. Now we can write anything in our `home.html` file we created in `templates`

```
<!--home.html-->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Home</title>
</head>
<body>
 <h1>Welcome to your first django App</h1>
</body>
</html>
```

Now we have to go our URLS to give a path that we can see this in our browser.

Creating our first URLS & Run the server:

In our project directory/file (here is helloworld), we can see a file named `urls.py` .

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


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

We have to go there & import our views & functions :

```
from django.contrib import admin
from django.urls import path
from student.views import * #we imported this.here student is

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

We imported this `from student.views import *` here from the student app we imported the views & for the views function we use * that we can get all the function in our `views.py` . You can call the function name also. Like `from student.views import home` this .But the problem is you have to add every function you write in your views .So better approach is `from student.views import *` adding this.

So after that we have to define our urls :

```
from django.contrib import admin
from django.urls import path
from student.views import *

urlpatterns = [
    path('', home, name="home"), # we defined this
    path('admin/', admin.site.urls),
]
```

here it is the first page so we make the path empty `''` & give the urls a name. You will do the same with your every URL. But remember that only the first URL will be empty (the page you want to show first) . for other urls you will adding like this :

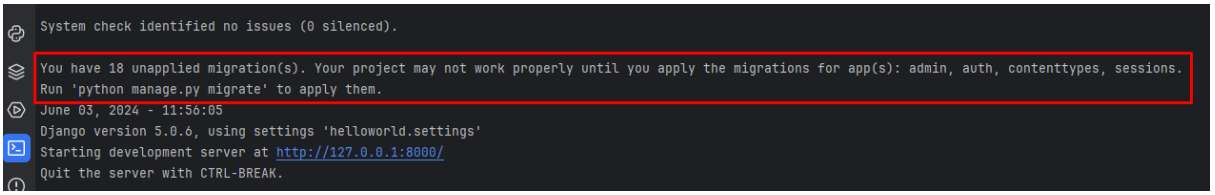
```
from django.contrib import admin
from django.urls import path
from student.views import *

urlpatterns = [
    path('', home, name="home"),
```

```
    path('about/', about, name='about'), # rest of the functi
    path('admin/', admin.site.urls),
]
```

💡 `path('about/', about, name='about'),` we don't have this function till now. I am just added this to give you an idea about it.

So here's the URL adding done.

Before run our server we have to do migration (we all talk about it later) to remove this warning in our terminal:



what we have to run it's already write it which is :

```
py manage.py migrate
```

After executing this command it will something like this in your terminal:

```
(myapp) PS F:\Python\Blog\Django101> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(myapp) PS F:\Python\Blog\Django101>
```

That means your migration is successfully completed.

Now we have to run our server from the terminal using :

```
py manage.py runserver
```

After run this command you can see there is no warning :

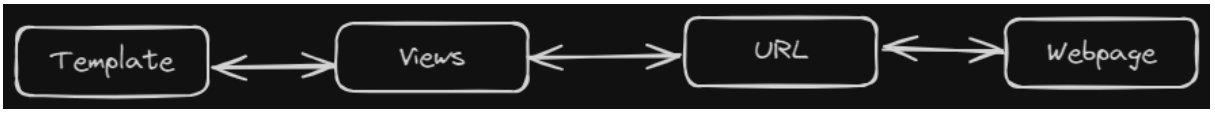
```
(myapp) PS F:\Python\Blog\Django101> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 03, 2024 - 12:04:43
Django version 5.0.6, using settings 'helloworld.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Now go to the link `http://127.0.0.1:8000/` & you will see your html :

Welcome to your first django App

Congratulations !!!!! Your first web page is ready !!!! You can customize it or you can write anything you want in your html.After writing, you just came back to your browser & refresh it you will see the changes you made in your html.

If I summarize or simplified it for you the steps are like this:



That's how it works!!!!

You can take References from here also:

Django Getting Started

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,

https://www.w3schools.com/django/django_getstarted.php

You can go through one of my Projects also for better understanding :

https://github.com/yeakiniqra/carrental

> *Written by*: **YEAKIN IQRA**