

EMAIL: dellyit001@gmail.com

Order a course at: <https://dellyit.com>

Django web development

When developing a Django website you need to be familiar with some basic python as well as graphical user interface basics in python. Python graphical user interface is actually divided into various components for instant: Django, Tkinter, kivy, Jinja2, Flask, pytorch as well as pygame. You should be first start with Tkinter after learning the basics to Python programming.

Here are some few tips I found and they really helped me become an expert in Django as well as advanced python development. They include the following;

1. Make two hours per day to learn python
2. Make sure you can run a basic python code in pycharm, visual studio, command prompt e.t.c
3. Form a group of two or three members
4. Make sure you have an aim of the project to have completed at the end of every week.
5. Post every code you code for yourself without copy pasting from someone else or the code used in tutorials to github.com or your own portfolio (after watching and practicing what you see in a tutorial please have time to sit down and try to practice what you learnt since programming is all about skills you gain at the end of every day)

History of Django

The name Django is a Romani boy's word which means "I awaken." The legendary Belgian-born jazz guitarist Django (formerly Jean Baptiste) Reinhardt's moniker, Django — because D is silent, as virtually everybody now knows — creates a strong melodic choice for any jazz enthusiast.

Introduction to Django

Start by first setting the virtual environment

```
python -m pip install --user virtualenv
```

The output is as shown:

```
Collecting virtualenv
  Downloading virtualenv-20.13.0-py2.py3-none-any.whl (6.5 MB)
    | 6.5 MB 656 kB/s
Requirement already satisfied: six<2, >=1.9.0 in c:\users\denni\appdata\roaming\python\python310\site-packages (from virtualenv) (1.16.0)
Collecting filelock<4, >=3.2
  Downloading filelock-3.4.2-py3-none-any.whl (9.9 kB)
Collecting platformdirs<3, >=2
  Downloading platformdirs-2.4.1-py3-none-any.whl (14 kB)
```

```
| 461 kB 1.1 MB/s  
Installing collected packages: platformdirs, filelock, distlib, virtualenv
```

You now need to activate the virtual environment as shown below:

```
.\env\Scripts\activate
```

When you get the following output, don't worry, I got a solution for you.

```
WARNING: The script virtualenv.exe is installed in  
'C:\Users\denni\AppData\Roaming\Python\Python310\Scripts' which is not on PATH.  
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-  
location.  
Successfully installed distlib-0.3.4 filelock-3.4.2 platformdirs-2.4.1 virtualenv-20.13.0  
WARNING: You are using pip version 21.1.2; however, version 21.3.1 is available.  
You should consider upgrading via the 'C:\Users\denni\PycharmProjects\first-  
django\venv\Scripts\python.exe -m pip install --upgrade pip' command.  
.\env\Scripts\activate : File C:\Users\denni\PycharmProjects\first-django\env\Scripts\Activate.ps1 cannot  
be loaded because running scripts is disabled on this  
system. For more information, see about_Execution_Policies at  
https://go.microsoft.com/fwlink/?LinkID=135170.  
At line:1 char:1  
+ .\env\Scripts\activate
```

Go to shell and make sure you run it as administrator, then from there you can run the following commands.

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine
```

You can now focus in the installation of all requirements which might be additionally needed in the development of Django websites

```
py -m pip install requests
```

This code installs are the required modules in the virtual environment development.

The output is as shown below:

```
Collecting requests  
Using cached requests-2.27.1-py2.py3-none-any.whl (63 kB)  
Collecting certifi<=2017.4.17  
Downloading certifi-2021.10.8-py2.py3-none-any.whl (149 kB)  
| 149 kB 386 kB/s  
Collecting charset-normalizer~2.0.0  
Downloading charset_normalizer-2.0.10-py3-none-any.whl (39 kB)  
Downloading idna-3.3-py3-none-any.whl (61 kB)  
| 61 kB 486 kB/s  
Collecting urllib3<1.27,>=1.21.1  
Using cached urllib3-1.26.8-py2.py3-none-any.whl (138 kB)  
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests  
Successfully installed certifi-2021.10.8 charset-normalizer-2.0.10 idna-3.3 requests-2.27.1 urllib3-1.26.8
```

=====CONGRATULATION, NOW YOU ARE SET FOR DJANGO INSTALLATION=====

We are now a few steps towards running our first Django website.

Now you can install Django into your system through the following method:

```
pip install django
```

Or even specify the version you need as shown below:

```
pip install django==4.0
```

You can now create your first application in Django, call the project name the name which you are best confident with

```
django-admin startproject mysite
```

(mysite is the name of my project.)

You now need to migrate to where the project is, you can do so by navigating through.

```
cd mysite
```

Run the following command:

```
python manage.py migrate
```

These SQL commands are executed in the database file by migrate. As a result, after running migrate, all of your installed apps' tables are created in your database file. You can verify this by installing SQLite browser and opening db.sqlite3. After running the migrate command, all of the tables show in the database file.

You can now run your Django project as follows:

```
python manage.py runserver
```

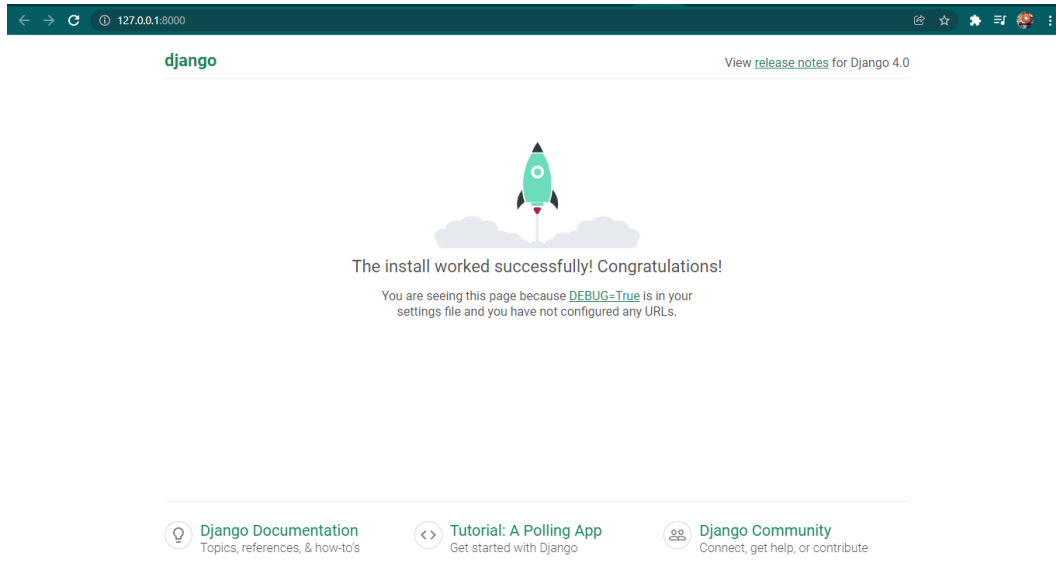
You can now create your own application as follows...

N/B: name your project according to the best name which suits your project.

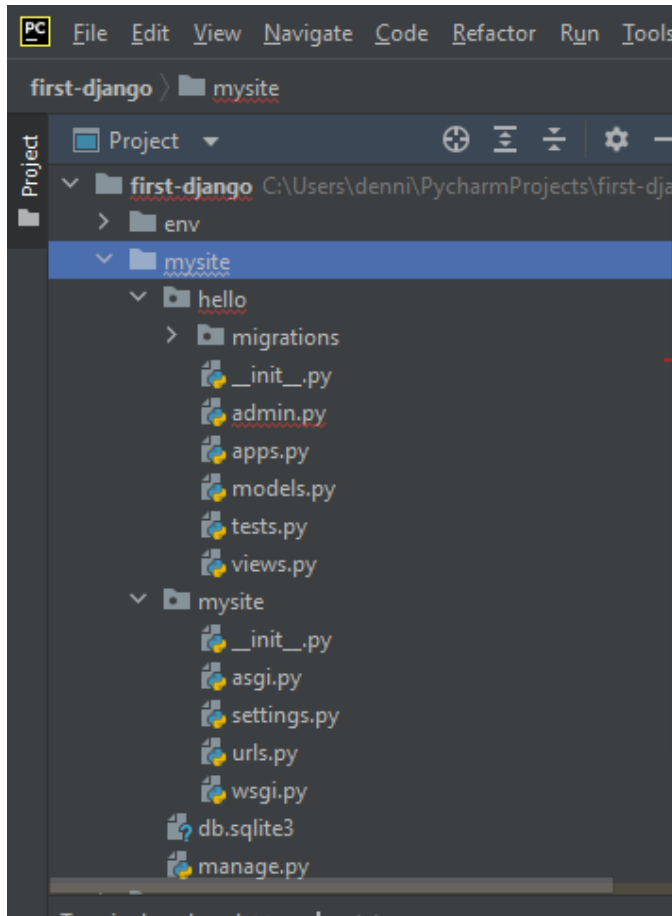
```
python manage.py startapp hello
```

(hello is my project name, you can call your project any name)

You can now run your project into a server, when you run the project into a server, you should receive such an output into a browser:



Make sure you have the following files:



These project files are:

1. The outer `_mysite/` root directory is a container for your project. The folder can be renamed.
2. `manage.py`: A command-line utility that lets you interact with this Django project in various ways. Do not touch this file.
3. The inner `_mysite/` directory is the actual Python package for your project. Its name is the Python package name you'll need to use to import anything inside it (e.g. `_mysite.urls`).
4. `_mysite/_init__.py`: An empty file that tells Python that this directory should be considered a Python package.
5. `_mysite/settings.py`: Settings/configuration for this Django project.
6. `_mysite/urls.py`: The URL declarations for this Django project; a "table of contents" of your Django-powered site.
7. `_mysite/asgi.py`: An entry-point for ASGI-compatible web servers to serve your project.
8. `_mysite/wsgi.py`: An entry-point for WSGI-compatible web servers to serve your project.

The App files

1. `admin.py`: Registering your models here for them to be manipulated on the django admin
2. `apps.py`: Contains the app configuration code
3. `models.py`: Define your models here. These are the tables used for storing data and defining their relations

4. `tests.py`: Writing tests for the app
5. `views`: Defining data to be rendered on the templates

Frequently Asked Questions:

Can I use Django to make Application?

Django is known for its one-of-a-kind app framework that can be fully managed. An app can be constructed as a completely self-contained module for any feature. This post will show you how to develop a basic app and then add functionality to it.

25 Best Django Project Ideas & Topics for Beginners

- ✚ Email Sender
- ✚ Text-to-HTML Converter
- ✚ Chat Application
- ✚ A Safe for Passwords
- ✚ Tweets Automator
- ✚ Dictionary Application
- ✚ Notes Application
- ✚ Django Blog
- ✚ Ecommerce Store
- ✚ Video Calling App
- ✚ Social Networking App
- ✚ Interactive Maps
- ✚ Django CMS
- ✚ News App
- ✚ Photo-centric App (like Pinterest)
- ✚ Login System
- ✚ To-Do App
- ✚ Weather App
- ✚ Calorie Counter
- ✚ Video Subscription App
- ✚ Online School System
- ✚ Library Management System
- ✚ Railway Enquiry System
- ✚ Quiz App
- ✚ Web Crawler

=====Have a blessed day, happy coding=====

Email me at: dennisngugi115@gmail.com or use company email: dellyit001@gmail.com

Visit our website: <https://dellyit.com>