**A Django based personal portfolio website.**

**Technology : Django,Python,HTML,CSS**

**Working Demo:** xxxxxxx

**Github Link:** xxxxxx

As part of My Learning journey towards machine learning via Python, I made my first project in Django Framework. My personal Portfolio which displays all my work experience, my projects, education,skills, etc

**How to Start**

I will be using the following for this project:

* Django
* Python
* Pip
* Operating System-Windows 10
* Git

**Set Up Your Development Environment**

**Step:1 Install Python 3**

**Step 2: Create your project**

Whenever you are starting a new web development project, it’s a good idea to first set up your development environment. Create a new directory for your project to live in, and cd into it:

$ mkdir mk-portfolio  
$ cd mk-portfolio

**Step 3: Install virtual Environment**

Now we will use a module named virtualenv to create isolated virtual environments. But first, let’s install this module by the following command,

$ pip install virtualenv

To verify a successful installation run this

virtualenv --version

**Step 4: Create your Virtual Environment**

Once you're inside the main directory, it’s a good idea to create a [virtual environment](https://realpython.com/python-virtual-environments-a-primer/) to manage dependencies. There are many different ways to set up virtual environments, but here you’re going to use venv:

$ virtualenv myenv

**Step 5: Activate your virtual env :**

This command will create a folder venv in your working directory. Inside this directory, you’ll find several files including a copy of the Python standard library. Later, when you install new dependencies, they will also be stored in this directory. Next, you need to activate the virtual environment by running the following command.

C:\> myenv\Scripts\activate.bat

You’ll know that your virtual environment has been activated, because your console prompt in the terminal will change. It should look something like this:

(venv) $

**Step 6: Install Django on your virtual env :**

If you have already installed Django, you can skip this section and jump straight to the Setting up the project section. To Install Django on your virtual environment run the below command

pip install Django

This will install the latest version of Django in our virtual Environment. To know more about Django installation read: [How To Install Django](https://djangocentral.com/how-to-install-django/)

Note — You must install a version of Django greater than 2.0

Step 7: Create your Django app

Now run the following command in your shell to create a Django project.

django-admin startproject myproject

This will generate a project structure with several directories and python scripts.

├── myproject  
│ ├── \_\_init\_\_.py  
│ ├── settings.py  
│ ├── urls.py  
│ ├── wsgi.py  
├── manage.py

**Step 9: Create your portfolio app**

Navigate into the outer directory where manage.py script exists and run the below command.

cd mysite  
python manage.py startapp my-personal-portfolio

These will create an app named blog in our project.

├── db.sqlite3  
├── myproject  
│ ├── \_\_init\_\_.py  
│ ├── settings.py  
│ ├── urls.py  
│ ├── wsgi.py  
├── manage.py  
└── my-personal-portfolio  
 ├── \_\_init\_\_.py  
 ├── admin.py  
 ├── apps.py  
 ├── migrations  
 │ └── \_\_init\_\_.py  
 ├── models.py  
 ├── tests.py  
 └── views.py

Now we need to inform Django that a new application has been created, open your settings.py file and scroll to the installed apps section, which should have some already installed apps.

Now add the newly created app blog at the bottom and save it.

INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'my-personal-portfolio'  
]

**Step 10: Test your server**

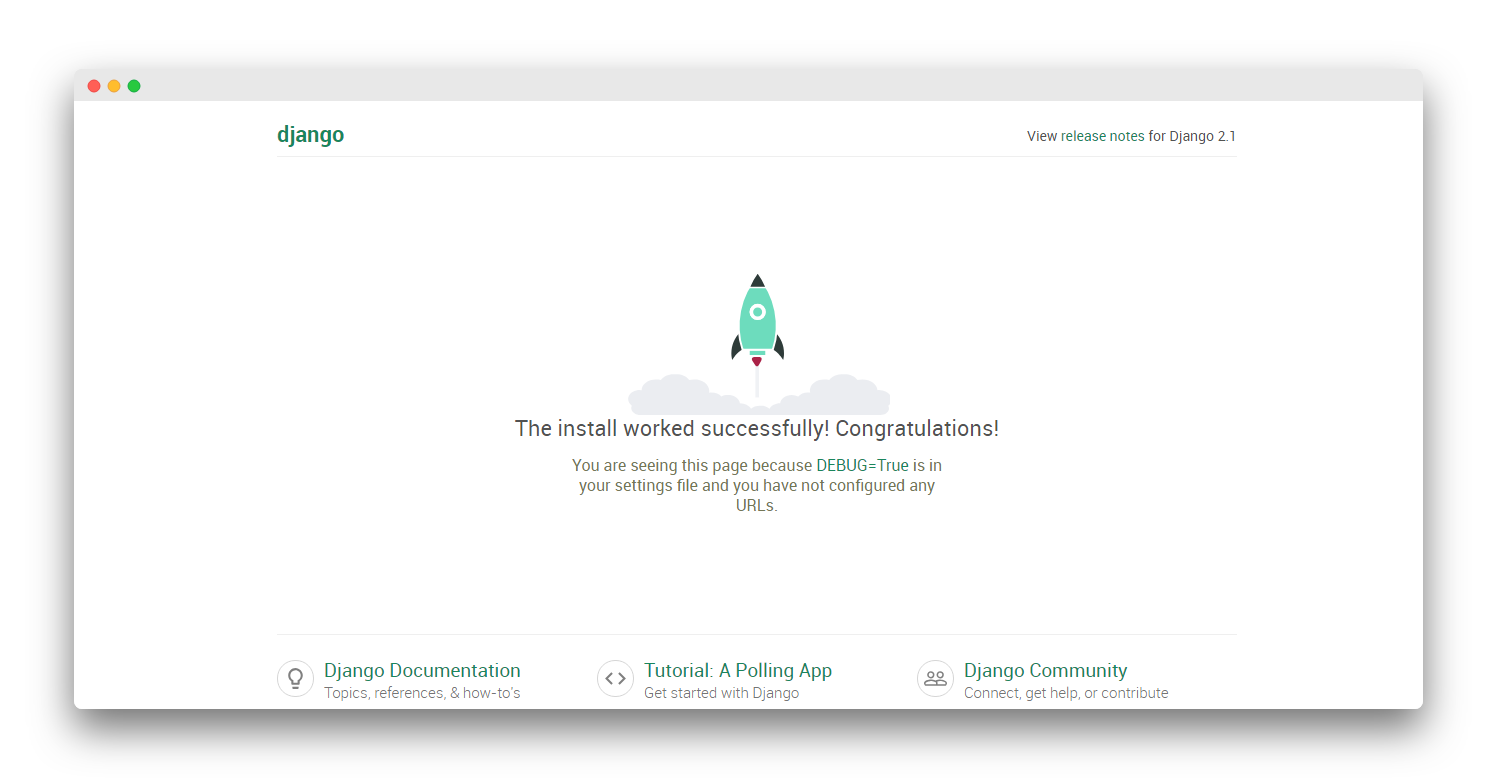
Next, make migrations.

python manage.py migrate

This will apply all the unapplied migrations on the SQLite database which comes along with the Django installation. Let’s test our configurations by running the Django’s built-in development server.

python manage.py runserver

Open your browser and go to this address http://127.0.0.1:8000/ if everything went well you should see this page.



**Step:11 Add your code**

You can visit my Github repository link and find my code.

If any queries ping me or comment down. I will be grateful to help you out.