Django

VENV

Create an environment using the following command

```
C:\Users\KIRAN\Desktop\Django\django\python -m venv <evn_name>
```

Then if you need all the packages installed by default in this env from the main env use

```
C:\Users\KIRAN\Desktop\Django\django\python -m venv <evn_name> --system-site-packages
```

Use pip list to get the list of packages,
Use pip freeze to get the list of packages with the version name

You can copy the **pip freeze** command output and save it in a **requirements.txt** file
And can emulate the same virtual environment for future project changes or to just the files without breaking due to change in the package versions.

To emulate the new env with the same packages and versions

```
C:\Users\KIRAN\Desktop\Django\django\pip install -r requirements.txt
```

Activate the VENV

We can manually install the packages into this env but before that, we need to be in that env or the packages will be installed globally.

```
C: \Users\KIRAN\Desktop\Django\django > venv\Scripts\activate.bat
```

Once the env is activated we can install the packages now and those packages will be installed only locally (remember not globally).

You can see the installed using the **pip list** or **pip freeze.**

Deactivating this env is as simple as saying **deactivate** in the cmd

Installing and populating the folders.

First, install the Django using **pip install django** command

The following command shows the version of Django.

```
(venv) C: \Users\KIRAN\Desktop\Django\django\first_project > python - m django - -version
>>> 3.1.4 # output
```

The next step is to generate the required files

```
(venv) C: \Users\KIRAN\Desktop\Django\django > python - m django startproject
```

Run the server.

Since all the files needed to build the API are populated, we run the code off the bat
Using the following command, notice that we need to be in the <project name> folder.
That can be done by simple **cd <project name>** cmd

```
C:\Users\KIRAN\Desktop\Django\django\first_project > python manage.py runserver
```

We get this output in our cmd

The system starts a local server at the web address http://127.0.0.1:8000/

We get the same result even if we replace **127.0.0.1** with **localhost**
The launched site is just a default one that does not has anything on it but just a simple Django starter site to help newbies to learn. To stop the server after running(or to interrupt) use **Ctrl+C.**

```
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, content types, sessions.
Run 'python manage.py migrate' to apply them.
December 22, 2020 - 17:46:31
Django version 3.1.4, using settings 'first_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

In our URL file, we can see that there is an admin route by default. Let's append that to the **localhost:8000/admin.** This route takes to the admin page aka login screen.