Installing Python packages with virtualenv

- So far we used normal python software environment for developing multiple django projects. So for all projects we can use same modules once installing in main python software(c-drive).
- But some times we required separate dependencies for every project. so that we will face some problems due to modules and versions.
- Typically, these situations only arise once you’re maintaining several Django projects.
- When they do, the best solution is to use virtualenv.
- This tool allows you to maintain multiple isolated Python environments, each with its own copy of the libraries and package namespace.

Creating virtual environment

- Install the Python software in our machine.
- Create a Virtual environment for our python software.
- Virtual environment is a tool, which is used to keep the dependences required by a different projects in different placess by creating virtual python environment for that.
- We can create the virtual environment for python by using "virtualenv" tool.
- "virtualenv" tool creates a folder which contains all the necessary executables to use the packages that a python package whould need.
- We can install the 'virtualenv' tool by using the pip of python software.
- pip is a tool, which will be installed automatically along with the python software.
- By using PIP command we can install the third party modules which are available in web locations like virtualenv
- PIP command communicating with web locations, searching for required module, finding required module , downloading that module and installing also.
- After installing third party modules by using PIP command, those modules available in Python software locations.
- Later we can use these modules where ever we want.

Syntax: cmd> pip install required_module_name

for example : cmd> pip install virtualenv

Create Python Virtual folder

- After installing virtualenv tool then by using this tool we can create required python virtual folder.
- create one folder with the name VirtualEnv_Projects in d-drive .
- goto d-drive VirtualEnv_Projects folder location
- D:\VirtualEnv_Projects> **C:\Users\Thinkpad\AppData\Local\Programs\Python\Python37-32\Scripts\vertualenv myvenv** ----->> click Enter
- Now vertualenv tool creates a folder like myvenv , which is acting like python virtual software folder.
- After creating this myvenv folder then we need to activate this python virtual folder
- start the vertual environment by using the activate command,
- D:\VirtualEnv_Projects>myvenv\Scripts\activate -->enter
- Then it creates virtual env like this , **(myvenv) D:\VirtualEnv_Projects>**
- Install the django software with in the virtual environment.
  (myvenv) D:\VirtualEnv_Projects> pip install django==2.0.5

Create the django projects in this vertual env locations

- Activate or Connect to the virtualenv like
  D:\VirtualEnv_Projects>myvenv\Scripts\activate -----> Click Enter

Project Creation

(myvenv)D:\VirtualEnv_Projects> django-admin startproject mysite

When we run the above command, mysite django project will be created with in the

D:\VirtualEnv_Projects> mysite folder.

Change the directory to manage.py location

```
(myvenv)D:\VirtualEnv_Projects>cd mysite
```

Create application name

```
(myvenv)D:\VirtualEnv_Projects\mysite> python manage.py startapp myapp
```

- Note: After developing the any required django project then we need to load all required modules related to this project in a file like this,
- cmd > pip freeze > requirements.txt
- When you are using a virtualenv, you can specify a requirements.txt file to install all the dependencies which are used for the current project only.
- After loading all dependencies which are used for the current project into a file then we need to install this file in a client location when ever we want.
- For this we need to use command like bellow,

  cmd > pip install -r requirements.txt
- If you want to see all dependencies which are used for the current project in a command line then we can use command like bellow,

  cmd> pip list