project1. (requests2.20.0, pydantic3.0.0, flask=1.0.0)
- venv

project2
- venv

project3
- venv

SDLC (Software Development Life Cycle)

requirements gathering -> software development -> testing -> deployment -> operations/maintainance

BA - Business Analyst (MBA - investment banking)
developer
tester
devops

Development

DEV -> TEST -> UAT -> PROD

DEV (linux, windows)
python 3.10, (requests2.20.0, pydantic3.0.0, flask=1.0.0)

PROD
python 3.10

Dependency Management

- `virtualenv` is used for dependency management.
- `virtualenv` is helpful in creating isolation of environments
-

- `pip` is package manager
- `pip` gets shipped along with Python installation
- `pip` downloads packages from `pypi.org`
- `pypi.org` packages index (repository for python packages)
- `pypi.org` packages authorized by `python.org`

command

- `pip ::`
- `pip install virtualenv`
- `virtualenv venv`
- `venv venv`
- `python -m virtualenv venv`
- windows: `venv\Scripts\activate`
- Unix: `source venv/bin/activate`

`pip` commands

- `pip install <package-name>==<version>`
- `pip uninstall <package-name>`
- `pip freeze` (tells you what all packages and their versions are currently installed)

how to deactivate `virtualenv` ?

- `deactivate`

what is `requirements.txt` ?

- Its a convention to have a file called `requirements.txt` under each project
- This file tells us package dependencies required for the project.
- To install all the packages from `requirements.txt` collectively run following command

  `pip install -r requirements.txt`
- to write output of `pip freeze` into `requirements.txt` run following -
- `pip freeze > requirements.txt`

Terminology

UAT : User Acceptance Testing