**Python version management:**

**Why Use Pyenv? (pyenv-win)**

**Isolation:** Pyenv helps create isolated Python environments for different projects, preventing conflicts between packages and dependencies.

**Version Control:** Easily switch between Python versions, ensuring compatibility with specific project requirements.

**Key Features:**

Version installation and removal.

Virtual environment integration (e.g., with virtualenv and venv).

Local and global Python version selection.

Plugin support for extended functionality.

**Basic idea of the structure :**

1)- you have to install pyenv as per the instructions .

2)- install specific python version using pyenv.

3)- Install virtualenv

4) - use virtualenv to create the virtual environment .

virtualenv venv --python=C:\Users\AkshayRajesh\.pyenv\pyenv-win\versions\3.9.7\python3.9.exe

5)- Activate the virtual environment

.\venv\Scripts\Activate

6) install the packages directly if you are starting the project .

7)- if you already have a requirments.txt file , install using that .

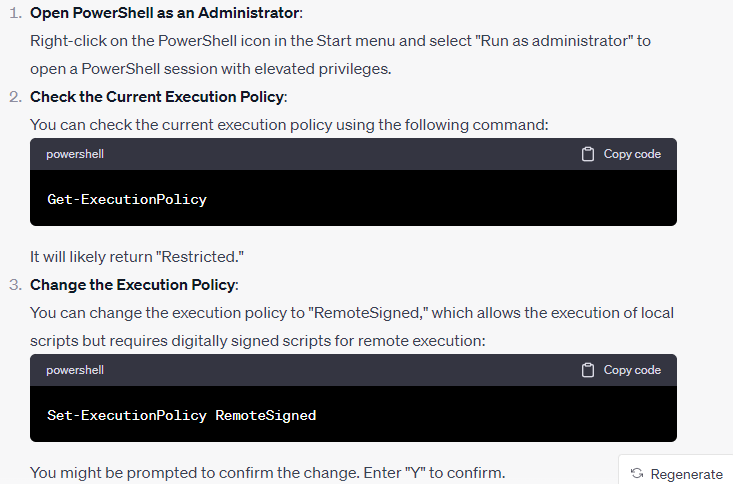
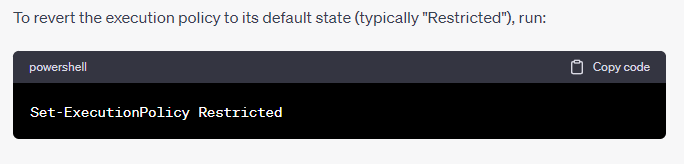
pip install -r requirements.txt

8) -Write the requiremnets.txt file , pip install -r requirements.txt

9) deactivate the evironment .

**Detailed steps:**

**How to install and use pyenv package :**

* Open powershell as admin -- **Invoke-WebRequest -UseBasicParsing -Uri "https://raw.githubusercontent.com/pyenv-win/pyenv-win/master/pyenv-win/install-pyenv-win.ps1" -OutFile "./install-pyenv-win.ps1"; &"./install-pyenv-win.ps1“**
* If you get an execution error do the below :
  + 
  + Then run the pyenv installation again -- **Invoke-WebRequest -UseBasicParsing -Uri "https://raw.githubusercontent.com/pyenv-win/pyenv-win/master/pyenv-win/install-pyenv-win.ps1" -OutFile "./install-pyenv-win.ps1"; &"./install-pyenv-win.ps1“**
  + Do remember to change back the execution policy (admin powershell)– 
  + Add it to the environment using -run the below using powershell-**[System.Environment]::SetEnvironmentVariable('PYENV',$env:USERPROFILE + "\.pyenv", [System.EnvironmentVariableTarget]::User)**

**[System.Environment]::SetEnvironmentVariable('PATH', $env:USERPROFILE + "\.pyenv\pyenv-win\bin;" + $env:PATH, [System.EnvironmentVariableTarget]::User)**

* + To check if pyenv is installed , open terminal and type this command it should show the pyenv version -- pyenv --version
  + To check for available python version to install use -- **pyenv install –list**
  + **pyenv install 3.9.7** – will install python 3.9.7
  + **pyenv global 3.9.7** – will set this as global version .

If you want a kernel in visual studio code for this version use the package ipykernel to create one .

* **pip install ipykernel** , if you want to create kernel for the python version you installed .
* **python -m ipykernel install --user --name Python\_3.9.7** – use this command to install the kernel.

**Environment management :**

**Virtual environment :**

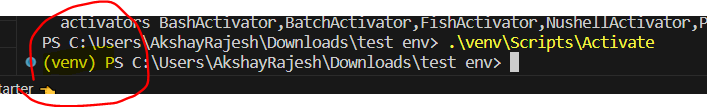
Virtualenv – is the package which allows to create virtual environments .

* to install the package -- **pip install virtualenv**
* to create a virtual env called **venv** in your workspace -- **virtualenv venv**
* **When you need to create a virtual environment with a specific python installation—**

**virtualenv venv --python=C:\Users\AkshayRajesh\.pyenv\pyenv-win\versions\3.9.7\python3.9.exe**

* to active the environment -- **.\venv\Scripts\Activate** – to activate the virtual environment.
* To deactivate the environment -- **deactivate**

The terminal will show our virtual environment name like below if its correctly activated:



**Package management :**

* **pip freeze > requirements.txt** – bundles all packages you have installed in the environment
* when you want only the used packages , we can use -- **pipreqs –force**
* This will install all the packges in your environment -- **pip install -r requirements.txt**

**Poetry :**

**pip install poetry**

Create the toml file – **poetry init**

Poetry cannot do python version management so if you want a specific python version you have to tell poetry to create your virtual environment using that python like this :  
  
**poetry env use C:\Users\AkshayRajesh\.pyenv\pyenv-win\versions\3.10.4\python3.10.exe**

**poetry install** – this will install all dependencies in the toml file

usually visual studio will show the virtual env created as the interpreter , if not you can do the below :

Once the virtual environment is created it will install all the dependencies mentioned in the toml file , now we have to find the path for the virtual environment to use it as our interpreter , copy the virtual env path and use it for your interpreter.

