1. Creat a vertual environment in project folder.
   1. We use it using conda mehtod. Hence first open conda prompt change directory to our folder and launce vs code from there using below command.
      1. To change directory write and enter: cd /d “Directory\_path”
      2. To launce vs cdoe, further write and enter: code .
   2. In VS code launch new command prompt and create new environment by entering below code.
      1. conda create –p venv python==3.8 –y
      2. From above code the virtual environment named venv will be created with all necessary folders.
      3. To activate your project in newly created environment and come out from your base computer environment. Enter below command in terminal with our project directory.
      4. conda activate “newly\_created\_environment\_path”
      5. The above code will activate folder into venv which can be seen in new directory line.
2. Sink a repository from github
   1. Create a new project repository in github.
   2. Initialise git in your local project folder, which will create a new hiden .git folder in your project. This will letter execute all the data available in our folder to repository. To make this at terminal with our directory and environment write below code.
      1. git init
   3. You can add any file to your local git repository by simply going with below command
      1. git add “File\_name\_to\_add”
   4. In order to add this file to online git repository, you have to identify your id and repository first. To do so, write below command in terminal.
      1. git config --global user.email "you@example.com"
      2. git config --global user.name "Your Name"
      3. You can skip –global to set identity for only that folder and not for all.
   5. To check the status of what is pending with respect to adding towards git we can write below command which will show list of files pending for addition in our local git file.
      1. git status
   6. To name the branch of your local git file with online one write below code and enter.
      1. git branch -M main
   7. To commit any non added file to local git
      1. git commit –m “Write\_your\_comment”
   8. To sink in with your online project folder. If your machine is not linked with your account, you may have to login first.
      1. git remote add origin “online\_file\_path”
   9. To activate or acces already added project online, enter below command.
      1. git remote set-url origin "url\_path"
   10. To push your origin main to your online repository, enter below command. This will push your local origin to your online main w.r.t. project mentioned in above path.
       1. git push –u origin main
   11. Note that there are various other functionality that can be used with git and github. This is required to make collaboration of coders and to maintain track of project with version control.
   12. Create .gitignore file in online project file. This will ignore some common files that need not be pushed like python, and environment etc. After listing down extensions and files in this folder, git will not consider it at as pending for commit to online. We need to add file by name .gitignore in our project and list down all extensions and files that we need to ignore. There are also preworked gitignore file that we can find online.
   13. Finally all the changes made online by self or by other should be sinked with current local machine and that’s why we should call all updated files to our local machine by below command, every time when we are starting further coding.
       1. git pull
3. Creating a setup and requirement text: We need to create a setup and requirement text to pull all necessary libraries and install in our environment so that everytime a user or we run the program we can refer to all the requirements aligned. Note that thease setup will hold all the packages and modules used in python and other programs. This is necessary to build your application as a package.
   1. Create setup.py file in our project file
   2. Create requirement.txt file in our project file which our setup code will access later.
   3. Create a primery folder(src) with main file name as \_\_init\_\_.py. Now this file or any such file within the directory will be primery source of input to setup.py to install various packages. Also we can name all the modules manually or we can just add extention of requirement.txt in setup.py to pull or install all the required liberaries.
   4. To execute setup.py itself we need to write –e. in the list of packages mentioned in requirement.txt. This will execute setup.py when we run pip command in command prompt. Also note that –e should not include in our list of liberaries, hence we should make custom change in our custome function, that we will mentione in setup.py file as below.
   5. For automatic analysis of requirement.txt as a list of package required, we can write a custom function using typing module(preinstalled as python package) and execute setup file.
   6. Note that some modules are already available in python package like setuptools and typing module, we don’t need to mention them in requirement.txt file
   7. After building the setup file we need to run below code on command prompt for automatic installation of listed modules in our environment.
      1. pip install –r requirement.txt