Contents

[https://www.krishnaik.in/playlist/PLZoTAELRMXVPS-dOaVbAux22vzqdgoGhG 1](#_Toc201402676)

[Github Setup 1](#_Toc201402677)

[Create project template in VS Code 4](#_Toc201402678)

# <https://www.krishnaik.in/playlist/PLZoTAELRMXVPS-dOaVbAux22vzqdgoGhG>

# Github Setup

1. Create Repository
2. Launch VS Code: from the required folder where project needs to be stored, Code . will launch VS Code

|  |
| --- |
| (base) C:\Users\ALL Atoz>cd E:\Amith\ML Project  (base) C:\Users\ALL Atoz> E:  (base) E:\Amith\ML Project>code .  (base) E:\Amith\ML Project> |

1. Create environment in VS Code 🡪 open terminal and use below code

|  |
| --- |
| conda create -p venv python==3.8 -y |

1. Activate environment in terminal

|  |
| --- |
| conda activate venv/ |

1. Clone repository
   1. Initialize Empty git – code “git init”
   2. Create readme file under project name

|  |
| --- |
|  |

* 1. Add readme file to git
     1. Code – “git add README.md’
  2. Commit the file to git
     1. Code – “git commit -m "first commit"
  3. If we need to check status use code – “git status”
  4. Create main branch
     1. Code – “git branch -M main”
     2. Code – “git remote add origin <https://github.com/amithku04/mlproject.git>”
  5. To check status, code- “git remote -v”
  6. Push to git hub repo
     1. Code- “git push -u origin main”

1. Create git ignore file- this will help to avoid files which are not required to commit
   1. Github site 🡪 add file
   2. Provide name as “.gitignore”
   3. Choose language as python
   4. Commit changes
2. Create Setup.py file in vs code
   1. This is responsible for converting the code to package for deployment
   2. Find package will search for “\_\_init\_\_.py” files to build package

|  |
| --- |
| from setuptools import find\_packages,setup  setup(      name='mlproject',      version='0.0.1',  author='Amith',      author\_email='amithku04@gmail.com',      packages=find\_packages(),      install\_requires=['pandas','numpy','seaborn'],  ) |

1. Create src folder and create file “\_\_init\_\_.py”
2. Create requirements.txt file
   1. Enter all the packages required for the project- this is to automate the packages to install
   2. “-e.” this code at last helps to trigger setup file to install packages
   3. Hence above code in setup needs to be dynamic

|  |
| --- |
| pandas  numpy  seaborn  -e . |

1. Updated setup file to call function and read requirement file and ignore “\n” and “-e .”
   1. This will enable automatically identify packages from requirements file

|  |
| --- |
| from setuptools import find\_packages,setup  from typing import List  HYPEN\_E\_DOT = '-e .'  def get\_requirements(file\_path:str)->list[str]:      '''      This function will return list of requirements or packages to install for project      '''      requirements = []      with open(file\_path) as file\_obj:          requirements=file\_obj.readlines()          requirements=[req.replace("\n","") for req in requirements]            if HYPEN\_E\_DOT in requirements:              requirements.remove(HYPEN\_E\_DOT)      return requirements  setup(      name='mlproject',      version='0.0.1',      author='Amith',      author\_email='amithku04@gmail.com',      packages=find\_packages(),      install\_requires=get\_requirements('requirements.txt'),  ) |

1. Run code “pip install -r requirements.txt”
   1. This help to build project and install packages
   2. “-e .” in requirements file will trigger setup.py file
2. “git add . “- to add files
3. “git status”- to review added files
4. “git commit -m "setup"”- commit to git. In this case setup indicate the description of the change
5. “git push -u origin main” – to push to main branch

# Create project template in VS Code

1. Create Components folder
   1. Create all required py files and init file
      1. Ex: Data ingestion is separate file, data transformer and model trainer as each files
2. Pipeline folder creation
   1. Training pipeline and prediction pipeline py files creation
   2. From training pipeline it should trigger components files
3. Logger, exception and utils files under src
   1. All common files to be created under src folder.

# Model Coding