**DOCUMENTATION Of PROJECT**

1. **Set up Github (repo):**
   1. New environment
   2. Setup.py
   3. Requirements.txt

**1. Set up GitHub Repository**

**a. New Environment**

**Need**: To ensure that the ML project dependencies are managed separately, avoiding conflicts with other projects.

1. **Create a virtual environment**:

bash

Copy code

python -m venv env

This creates a virtual environment named env.

1. **Activate the environment**:
   * On Windows:

.\env\Scripts\activate

* + On macOS/Linux:

bash

Copy code

source env/bin/activate

**b. Setup setup.py**

**Need**: This file defines the package metadata and allows others to install it easily.

1. **Create a file named setup.py** in the project root and add the following content:

from setuptools import setup, find\_packages

setup(

name='your\_project\_name',

version='0.1.0',

packages=find\_packages(),

install\_requires=[],

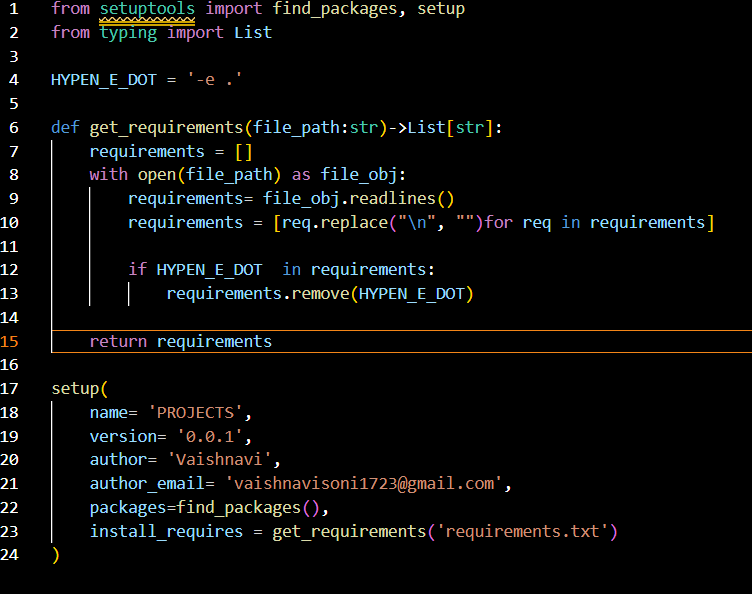
description='A brief description of your ML project',

author='Your Name',

author\_email='your\_email@example.com',

url='https://github.com/your\_username/your\_repo\_name',

)



1. **Install your package locally** (useful for development):

bash

Copy code

pip install -e .

**c. Requirements.txt**

**Need**: To list all the dependencies your project requires, ensuring reproducibility.

1. **Generate a requirements.txt file**:
   * List the dependencies manually:

text

Copy code

numpy

pandas

scikit-learn

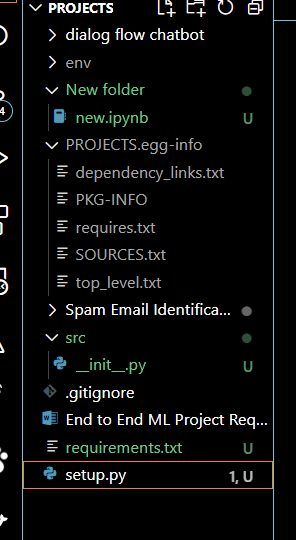
1. javascript
2. Copy code
3. - Or export them automatically:
4. ```bash
5. pip freeze > requirements.txt
6. **Install dependencies**:

bash

Copy code

pip install -r requirements.txt

1. **SRC folder and build the package**

****

**Need**: To organize the project into a well-structured package for easier collaboration and maintenance.

**Steps:**

1. **Create the folder structure**:

bash

Copy code

mkdir src

mkdir src/your\_project\_name

touch src/your\_project\_name/\_\_init\_\_.py

1. **Add your project-specific code**:
   * For example, add a file src/your\_project\_name/ml\_model.py:

python

Copy code

def train\_model():

print("Training the model...")

1. **Test the package**:
   * From the root directory, import your package:

bash

Copy code

python

>>> from src.your\_project\_name.ml\_model import train\_model

>>> train\_model()

**Final Commands**

To initialize the GitHub repository and push your project:

1. **Initialize a Git repository**:

bash

Copy code

git init

git add .

git commit -m "Initial commit"

1. **Link to GitHub**:

bash

Copy code

git remote add origin https://github.com/your\_username/your\_repo\_name.git

git branch -M main

git push -u origin main

**Summary of Files and Structure**

bash

Copy code

project-root/

│

├── env/ # Virtual environment

├── requirements.txt # Dependencies

├── setup.py # Package metadata

├── src/ # Source folder

│ └── your\_project\_name/ # Project code

│ ├── \_\_init\_\_.py # Package marker

│ └── ml\_model.py # Sample ML code

This setup ensures your ML project is well-structured, reusable, and ready for collaboration. Let me know if you'd like further clarification!