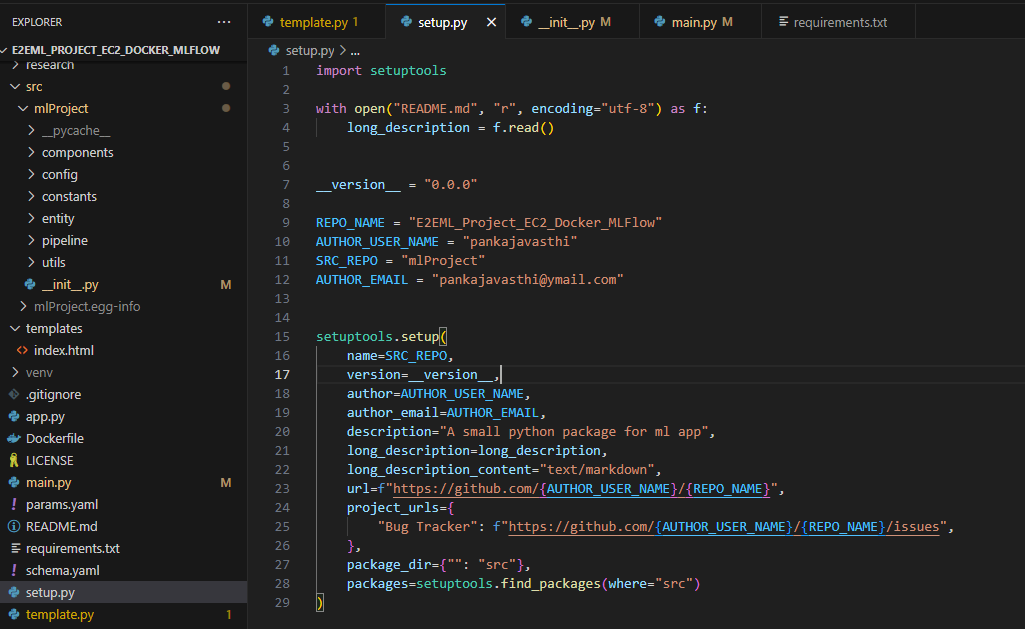
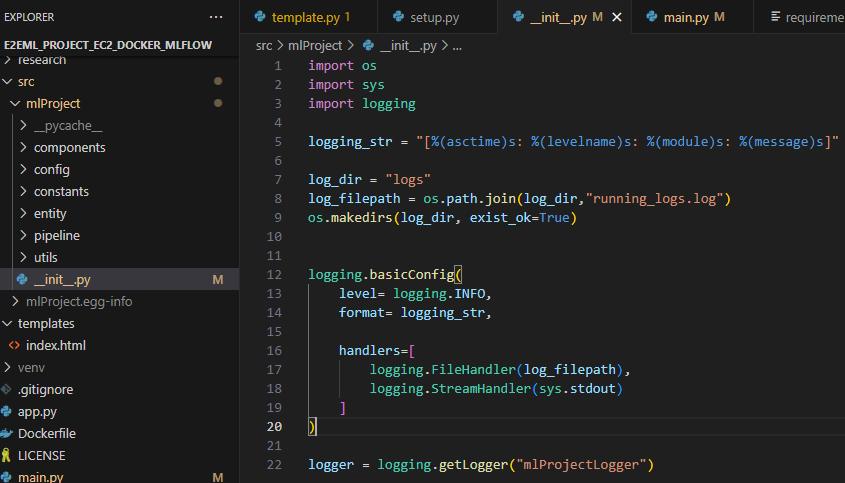
5)**Setup.py** - Before running requirement.txt, run setup.py to make all source folder as local packages using \_\_init\_\_.py constructor



6) **Logger file** – Create customer logger with below mentioned code to generate log in log folder and at terminal.



7) **Util Folder-** Util folder is create to create most common function which will be required frequently in our project. Those functions, which are required frequently in our components like reading yaml file we can create in common.py file and call it from here. Always require in Modular Coding style. Here we have functions a) Reading Yaml File b) Creating Directories like artificats, dataingestion, datavalidation, modelcreation c) save\_json, load\_json, save\_bins, load\_bins & get\_size. Kinldy refer code as we are using decorators like @ensure\_annotations and other functions like box.Exceptions and ConfigBox functionality in detail.

8) WorkFlow: Below is the process and usage of various file and folders to initiate the workflow by following modular level and object oriented programming concepts. Please see below details

1. Update config.yaml # Used to make all configuration oof our file

2. Update schema.yaml # We use Schema to define our variables used in data structure

3. Update params.yaml # All the parameter used in our project, will define here

4. Update the entity #

5. Update the configuration manager in src config

6. Update the components # Data ingestion, Validation,

7. Update the pipeline # All components will be intergrated through pipeline

8. Update the main.py # to maintain all code and run them centrally

9. Update the app.py # To create a flask app UI for our project