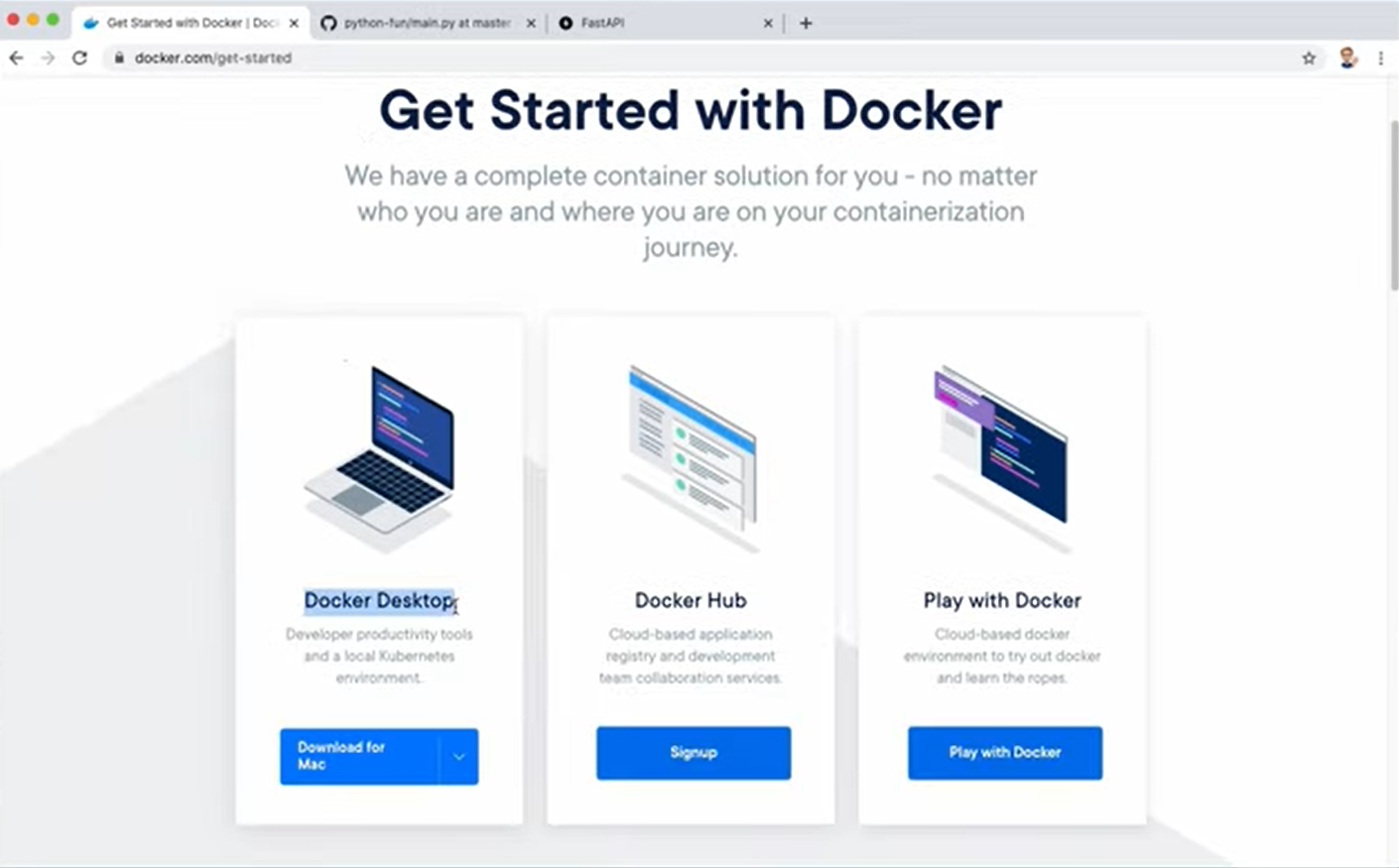
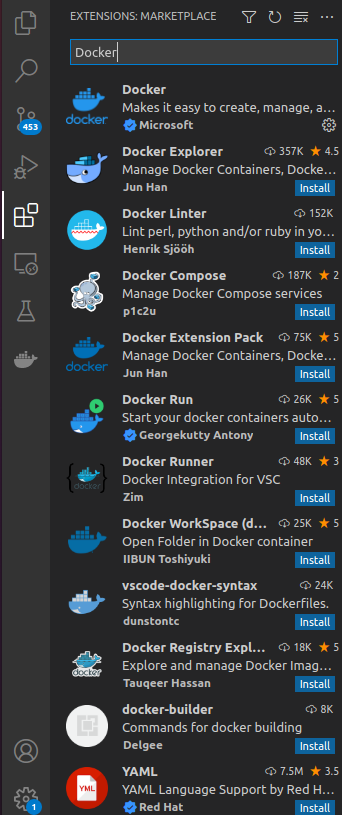
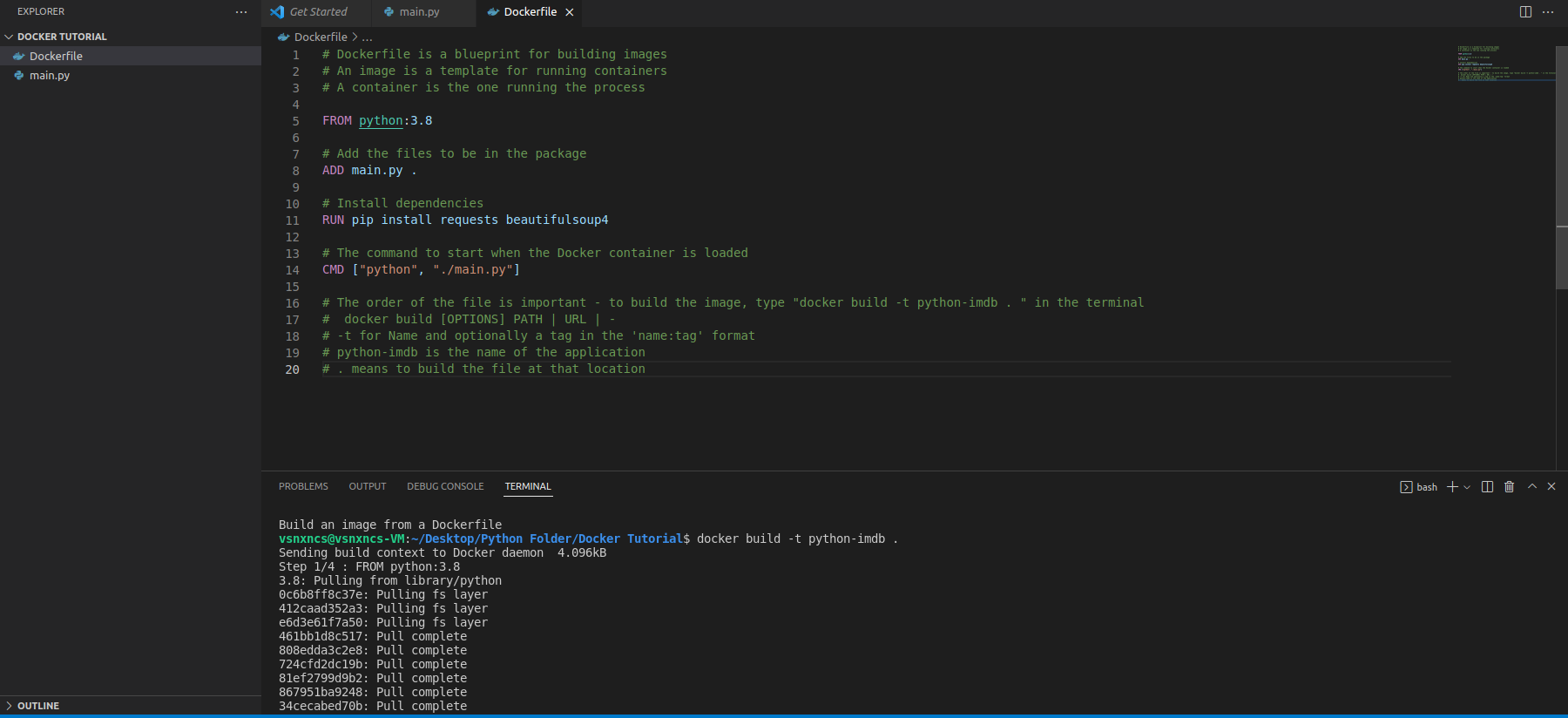
Docker Tutorial For Beginners – How to Containerize Python Applications  
  
Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Docker Desktop can be installed at docker.com/get-started

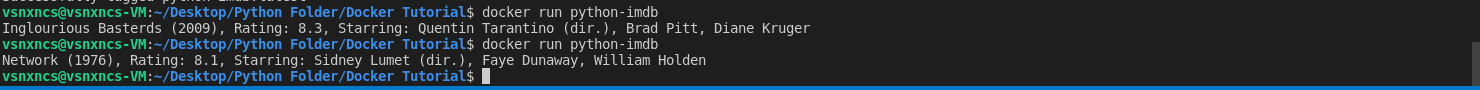
  
  
After installation, check that it has been installed by running Visual Studio and typing “docker -v”; you should see the version of Docker that was installed. Also, install the Docker extension on Visual Studio.  

Complete the Dockerfile and then run “docker build -t python-imdb .”

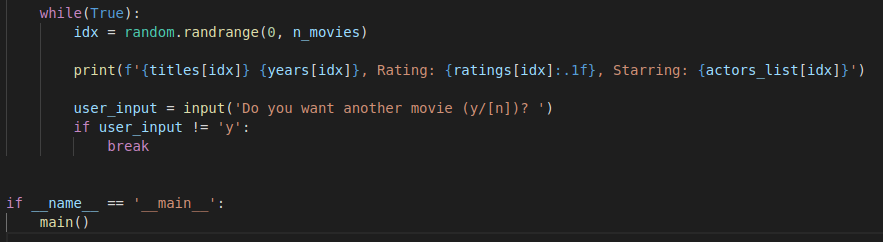


Run the containerized application by executing “docker run python-imdb” in the terminal

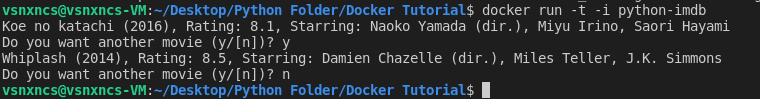


Building an application with input arguments

The code now accepts an input from the user and if the application is built and run as is, it will run into an EOF error.

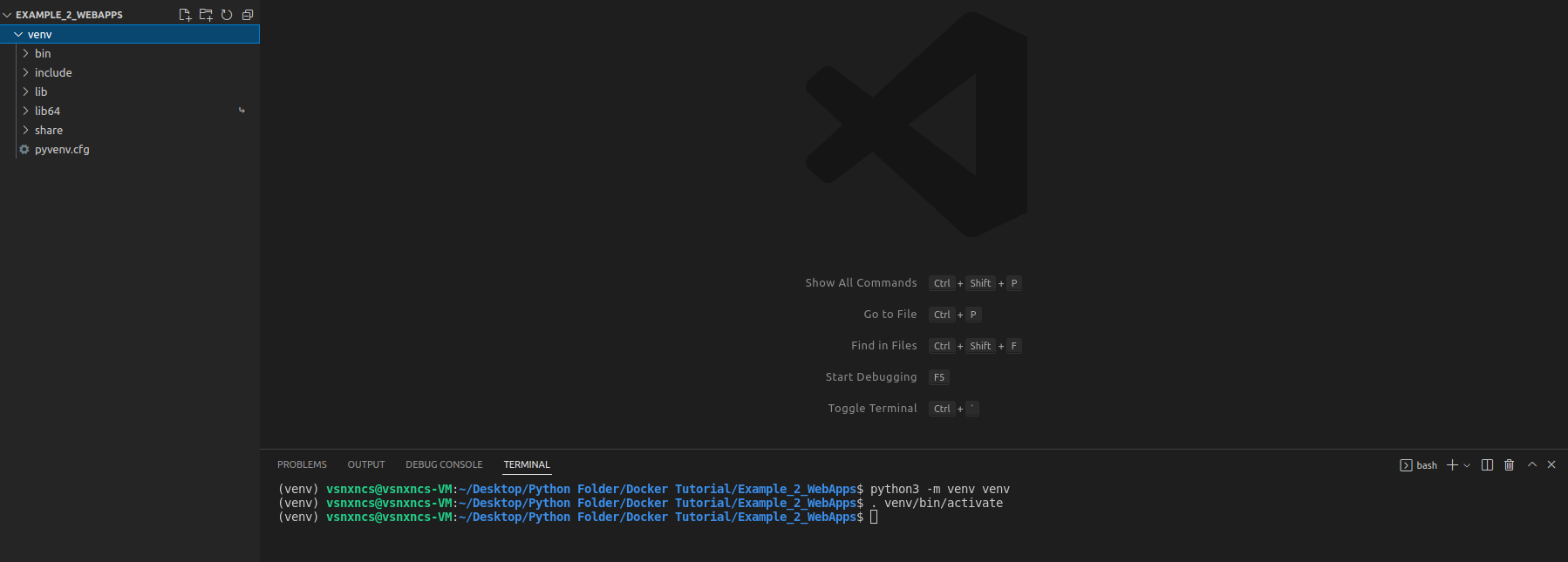


After re-building the image for the application, the command to run it in interactive mode is “docker run -t -i python-imdb”

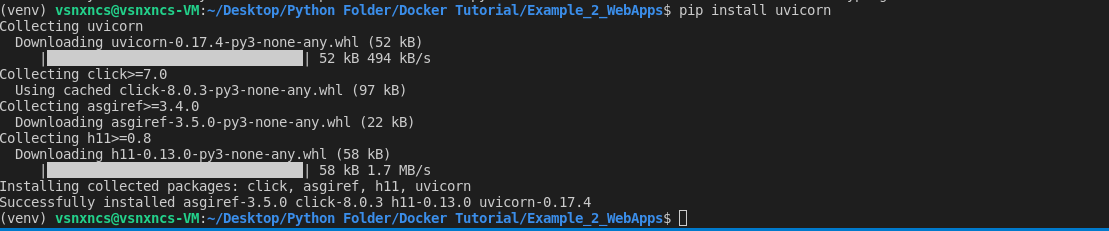


EXAMPLE 2

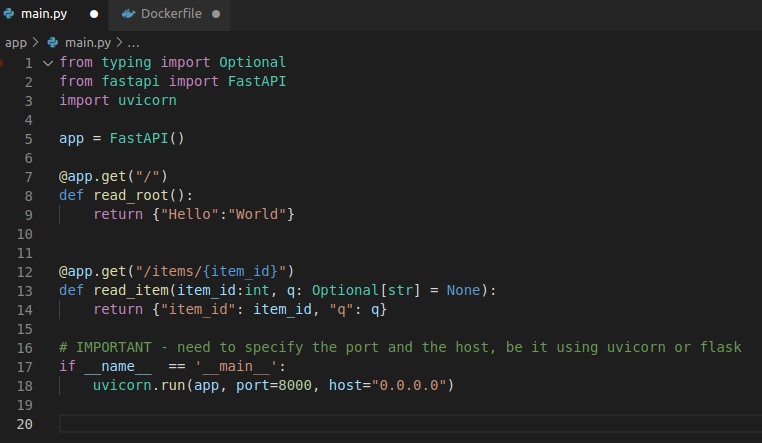
From a fresh project (empty directory), type “python3 -m venv venv” to create a venv folder. Then, activate the virtual environment by “. venv/bin/activate”



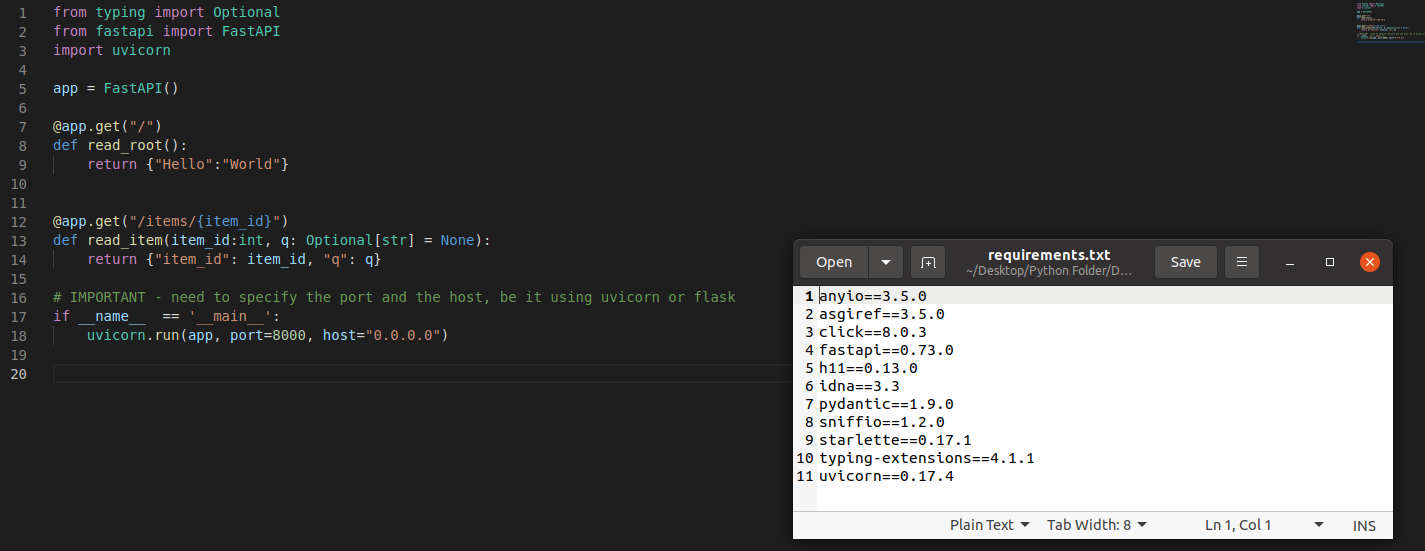
While in the virtual environment, install whatever packages your project requires



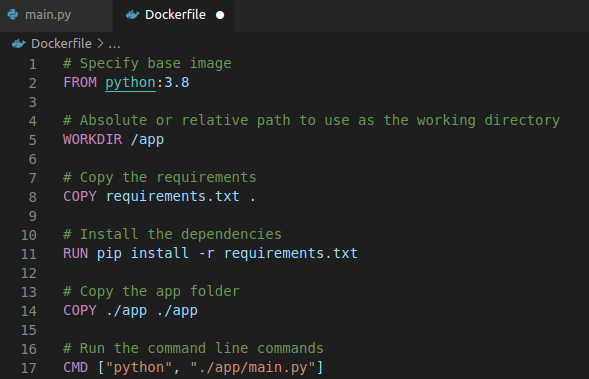
Create a new folder to run your application and write the code in the Python scripts



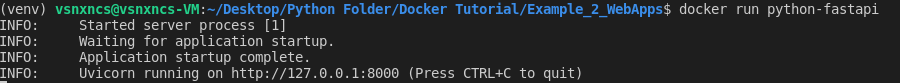
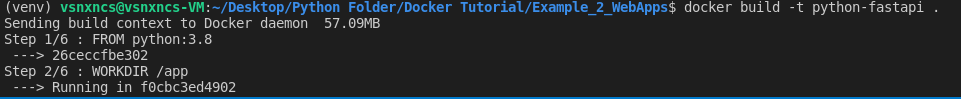
After testing that the code is able to run locally, navigate back to the root directory save the package requirements in a .txt file through the command “pip freeze > requirements.txt”



Create the Dockerfile in the root directory of the project – specify the base image, the working directory, the requirements, installing the requirements, and the command line commands to use when running the app



Build and run the Dockerized application with “docker build -t python-fastapi .” and “docker run -p 8000:8000 python-fastapi” respectively



Test that the application is working by accessing the URL and seeing the results

