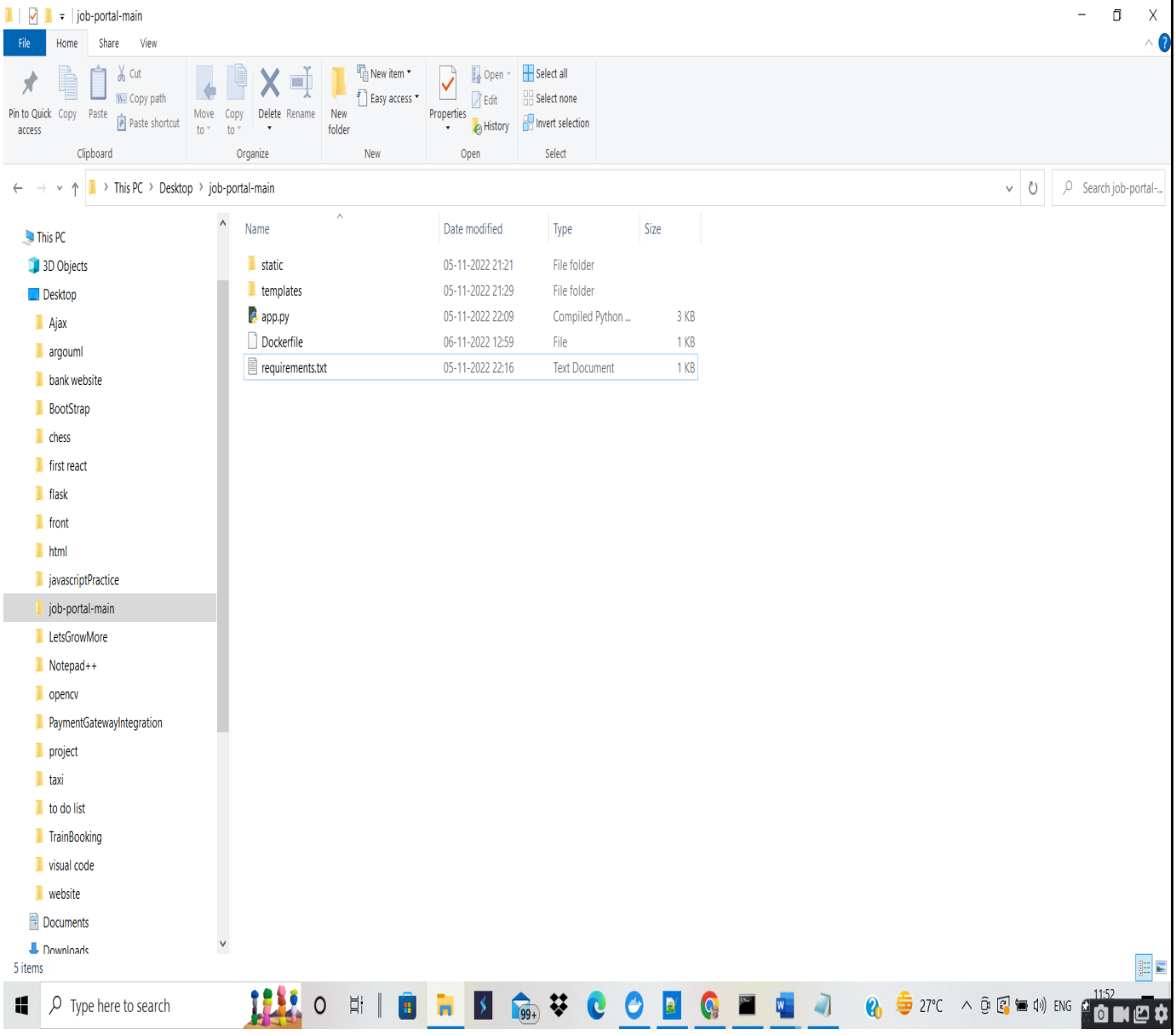
Containerize the App

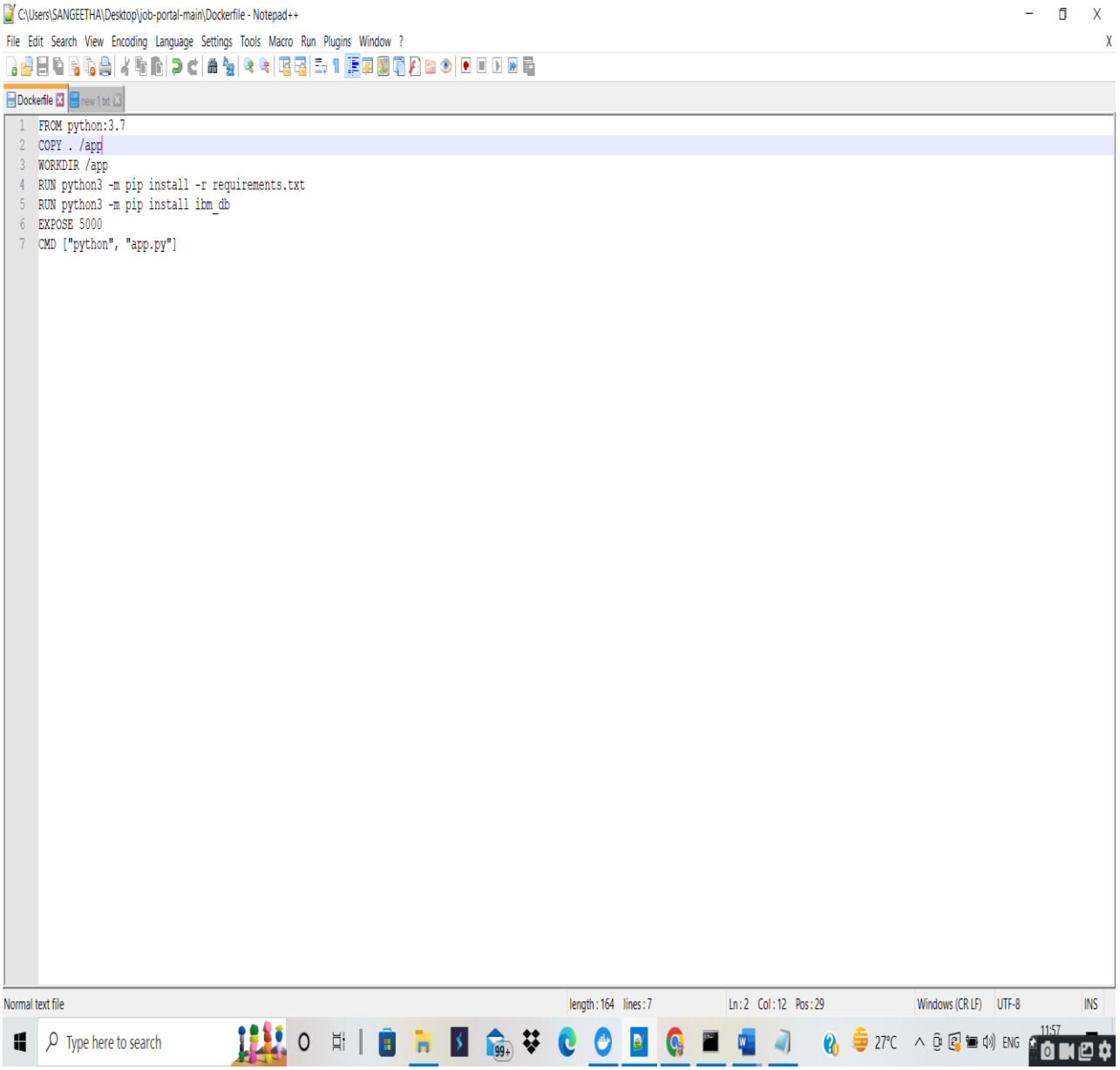
Docker Image Creation for Our Application

Date	15 November 2022
Team ID	PNT2022TMID23050
Project Name	Smart fashion recommender application

Step 1. In our project directory, we created the file named "**Dockerfile**" with no extension-A "Dockerfile" is used to indicate to Docker a base image, the Docker settings you need, and a list of commands you would like to have executed to prepare and start your new container.



Step 2. In the file, the following codes are written



Explanation and breakdown of the above Dockerfile code:

FROM python → Because this Flask application uses Python , we want an environment that supports it and already has it installed.

WORKDIR /app
ADD . /app
COPY requirements.txt/app

➔ Now it's time to add the Flask application to the image. For simplicity, copy the application under the /app directory on our Docker Image.WORKDIR is essentially a **cd** in bash, and COPY copies a certain directory to the provided directory in an image. ADD is another command that does the same thing as COPY, but it also allows you to add a repository from a URL.

RUN python3 -m pip install -r requirements.txt

→ Now that we have our repository copied to the image, we will install all of our dependencies, which is defined in the requirements.txt part of the code.

RUN python3 -m pip install ibm_db

→We used ibm_db as the database so we will install ibm_db

EXPOSE 8080

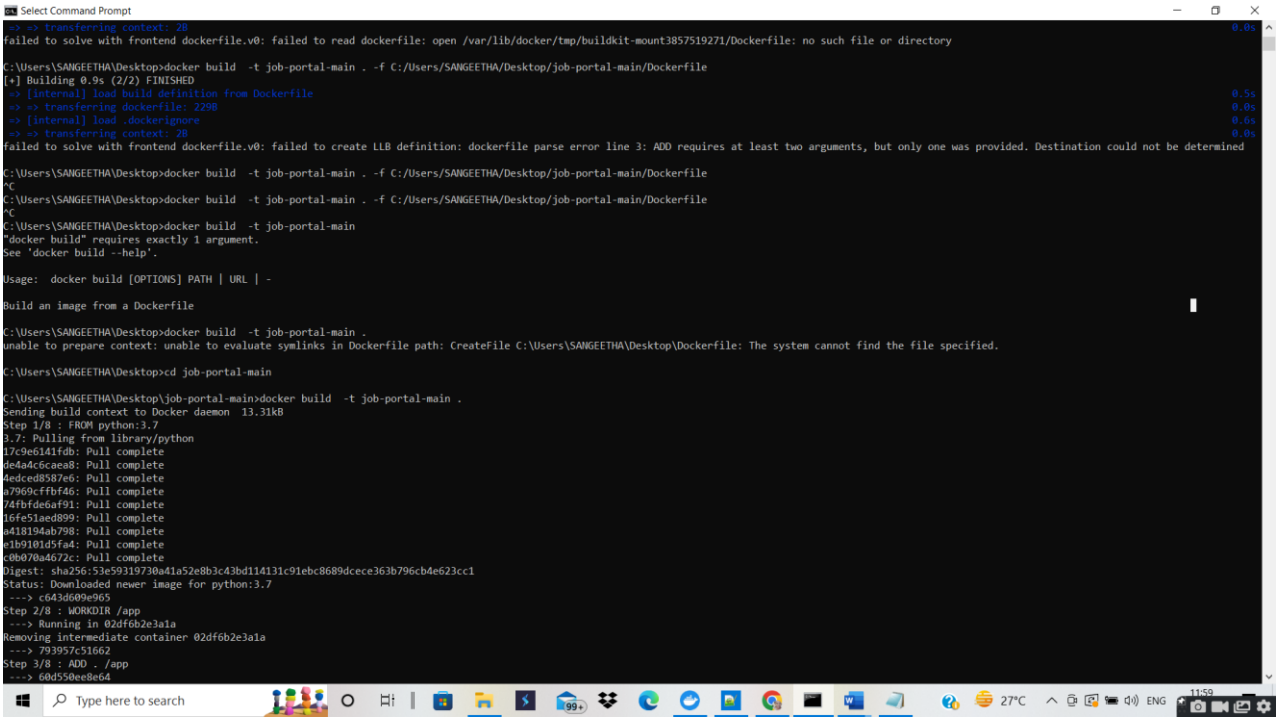
→ We want to expose the port(8080) the Flask application runs on, so we use EXPOSE.

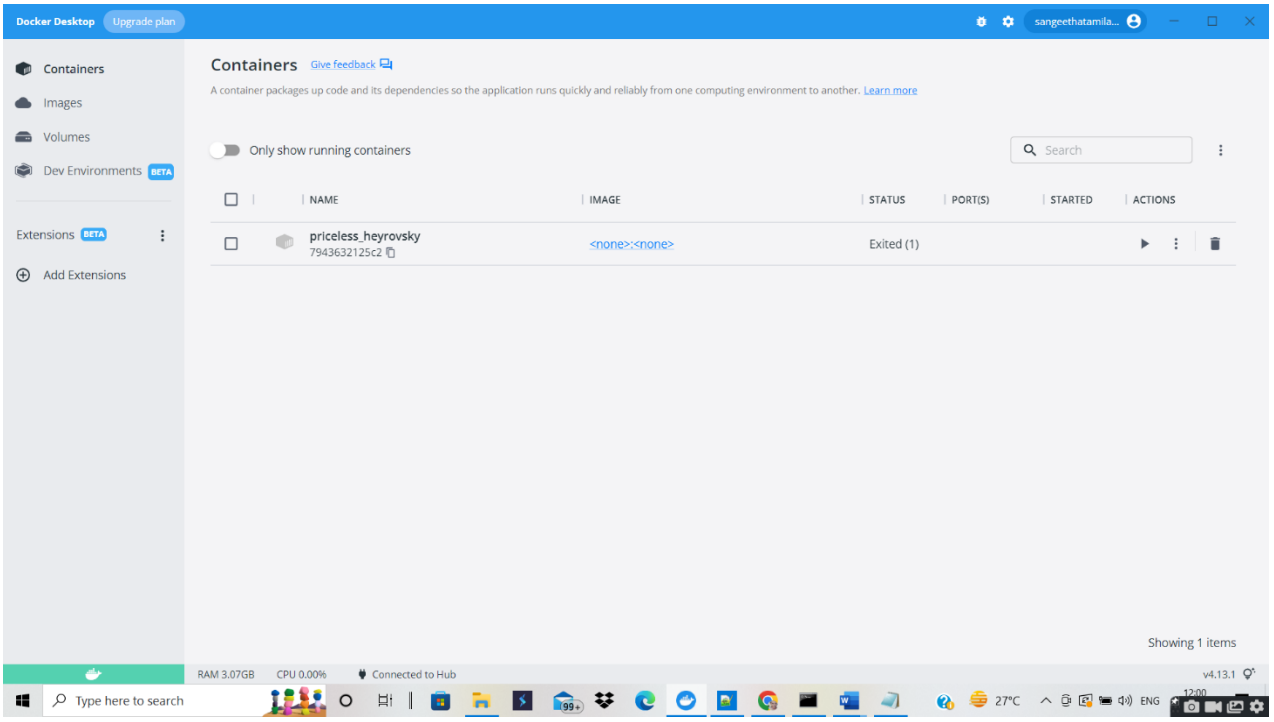
CMD [“python”, “app.py”]

→ ENTRYPOINT specifies the entrypoint of your application.

Step 3: Build an image from the Dockerfile

Open the terminal and type this command to build an image from your Dockerfile:
docker build -t <image_name>:<tag>


Select Command Prompt
=> => transferring context: 2B 0.0s
failed to solve with frontend dockerfile.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount3857519271/Dockerfile: no such file or directory
C:\Users\SANGEETHA\Desktop>docker build -t job-portal-main . -f C:/Users/SANGEETHA/Desktop/job-portal-main/Dockerfile
[+] Building 0.9s (2/2) FINISHED
=> [internal] load build definition from Dockerfile 0.5s
=> => transferring dockerfile: 229B 0.0s
=> [internal] load .dockerignore 0.6s
=> => transferring context: 2B 0.0s
failed to solve with frontend dockerfile.v0: failed to create LLB definition: dockerfile parse error line 3: ADD requires at least two arguments, but only one was provided. Destination could not be determined
C:\Users\SANGEETHA\Desktop>docker build -t job-portal-main . -f C:/Users/SANGEETHA/Desktop/job-portal-main/Dockerfile
^C
C:\Users\SANGEETHA\Desktop>docker build -t job-portal-main . -f C:/Users/SANGEETHA/Desktop/job-portal-main/Dockerfile
^C
C:\Users\SANGEETHA\Desktop>docker build -t job-portal-main
"docker build" requires exactly 1 argument.
See 'docker build --help'.
Usage: docker build [OPTIONS] PATH | URL | -
Build an image from a Dockerfile
C:\Users\SANGEETHA\Desktop>docker build -t job-portal-main .
unable to prepare context: unable to evaluate symlinks in Dockerfile path: CreateFile C:\Users\SANGEETHA\Desktop\Dockerfile: The system cannot find the file specified.
C:\Users\SANGEETHA\Desktop>cd job-portal-main
C:\Users\SANGEETHA\Desktop\job-portal-main>docker build -t job-portal-main .
Sending build context to Docker daemon 13.31kB
Step 1/8 : FROM python:3.7
3.7: Pulling from library/python
17c9e6141fdb: Pull complete
de4a4c6caea8: Pull complete
4edced8587e6: Pull complete
a7969cffbf46: Pull complete
74fbfde6af91: Pull complete
16fe51aed899: Pull complete
a418194ab798: Pull complete
e1b9101d5fa4: Pull complete
c0b070a4672c: Pull complete
Digest: sha256:53e59319730a41a52e8b3c43bd114131c91ebc8689dcece363b796cb4e623cc1
Status: Downloaded newer image for python:3.7
---> c643d609e965
Step 2/8 : WORKDIR /app
---> Running in 02df6b2e3a1a
Removing intermediate container 02df6b2e3a1a
---> 793957c51662
Step 3/8 : ADD . /app
---> 60d550ee8e64
Type here to search
27°C
ENG
11:59


Docker Desktop
Upgrade plan
sangeethatamila...
Containers
Images
Volumes
Dev Environments BETA
Extensions BETA
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
priceless_heyrovsky
7943632125c2
<none>:<none>
Exited (1)
Showing 1 items
RAM 3.07GB
CPU 0.00%
Connected to Hub
v4.13.1
Type here to search
27°C
ENG
12:00

Step 4: Run your container locally and test

After you build your image succesfully, type: docker run -d -p 8080:8080 jobport

This command will create a container that contains all the application code and dependencies from the image and runs it locally.