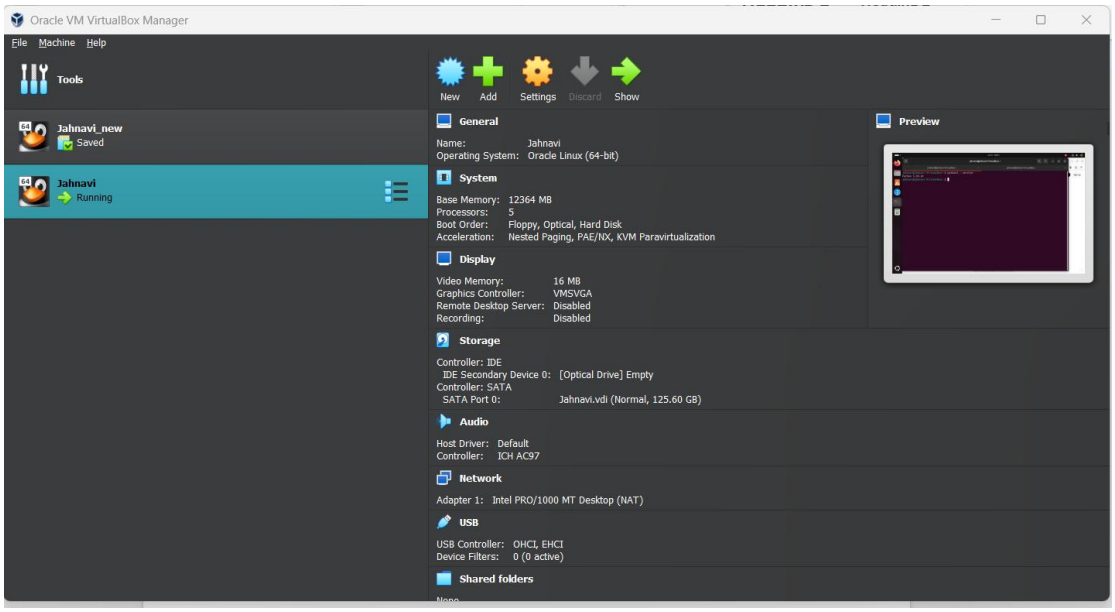
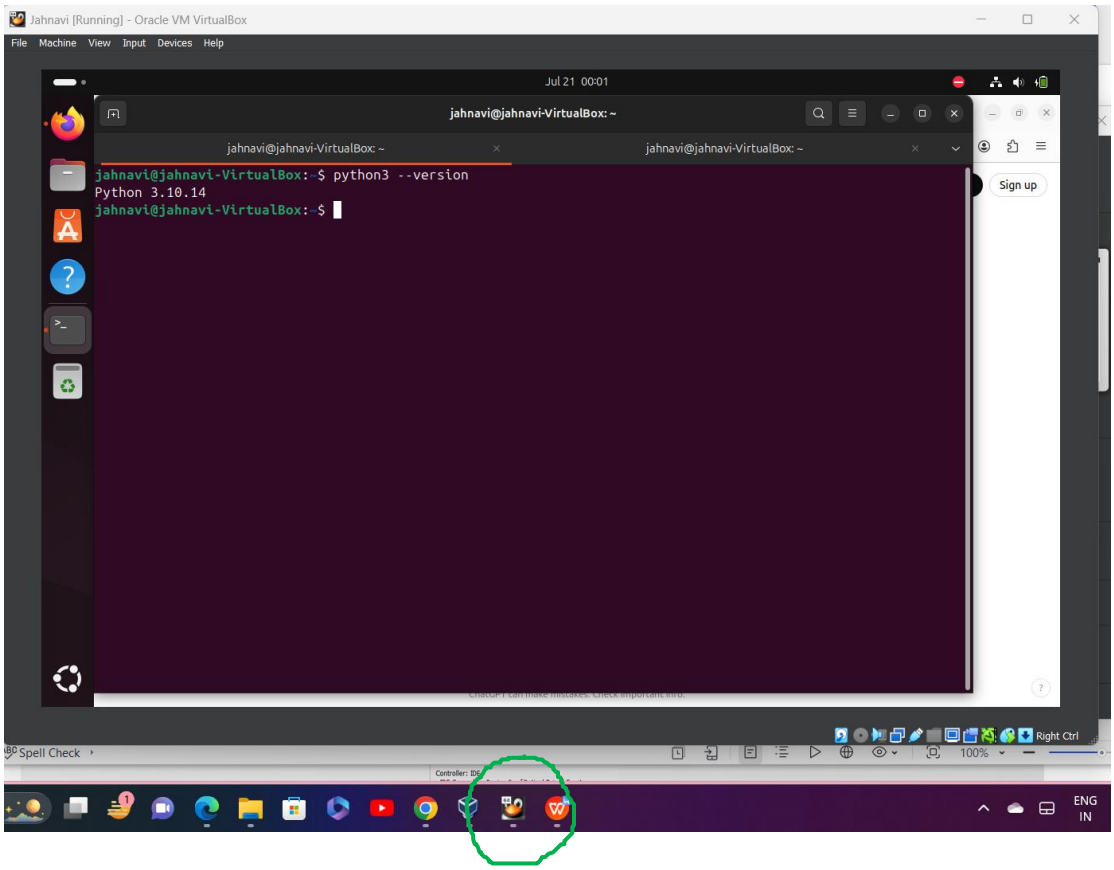
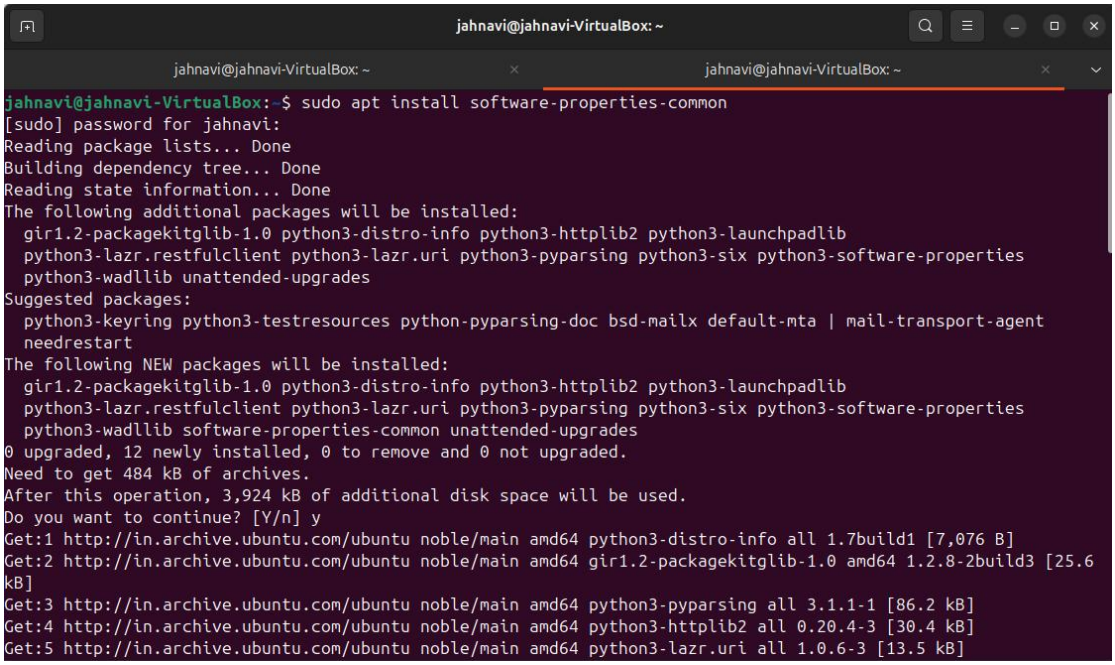
# WEEK10 -WEEK 12 ASSIGNMENT

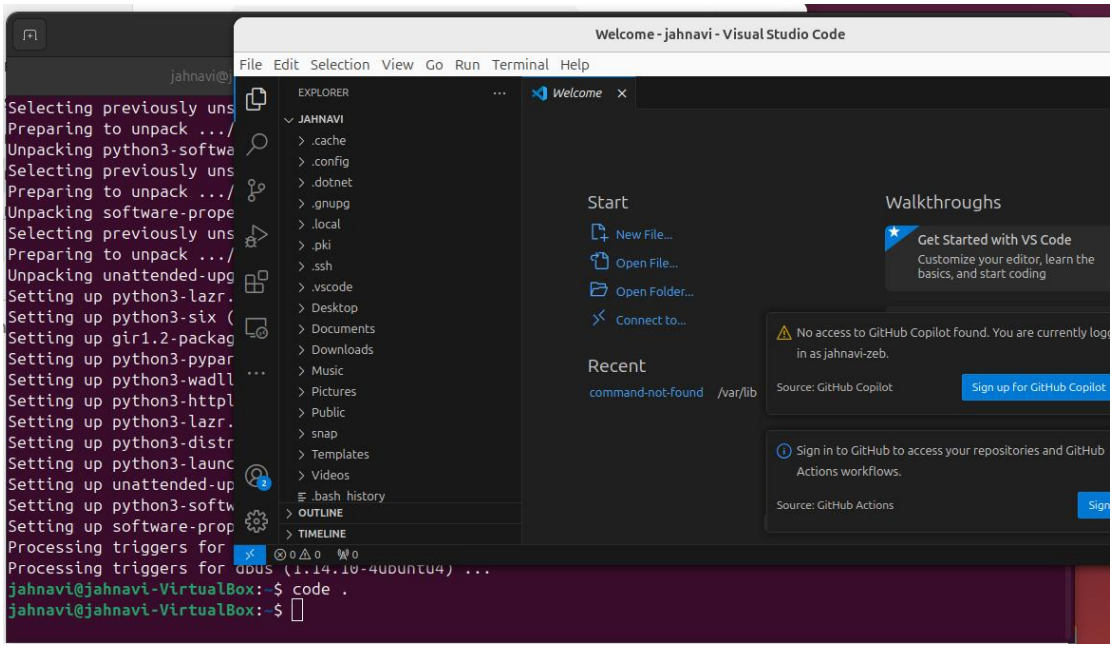
1. Host a Ubuntu Virtual Machine using Oracle VM Virtual Box
Solution : Hosted Ubuntu Virtual machine using oracle VM Virtual box



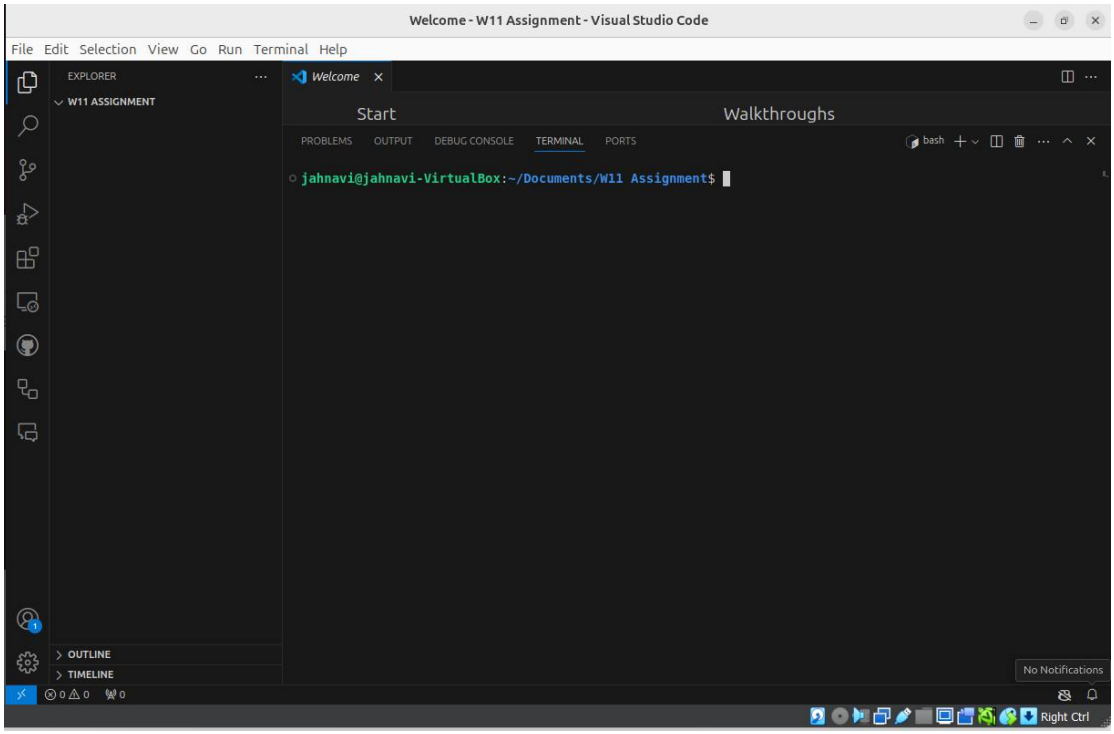






2. Set up Visual Studio code on Ubuntu VM.
Downloaded the VScode file from https://code.visualstudio.com/
Used Terminal to install it.

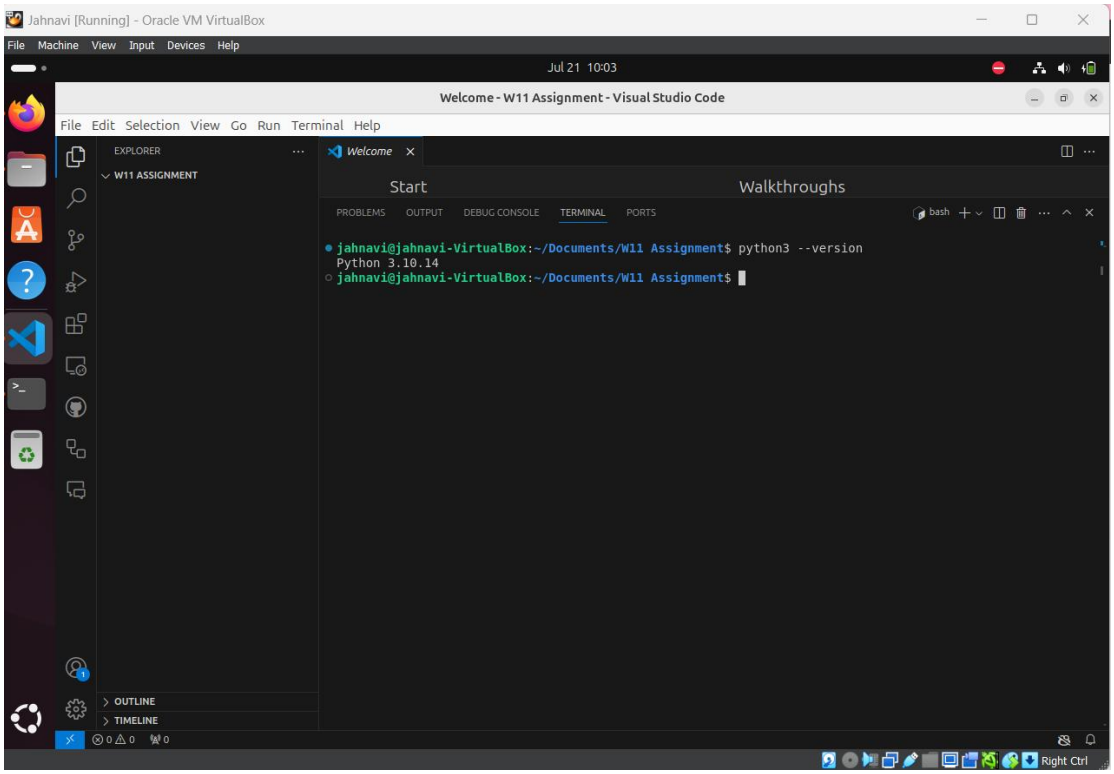


Created a document with Assignment name for this project.




3. Set up Python

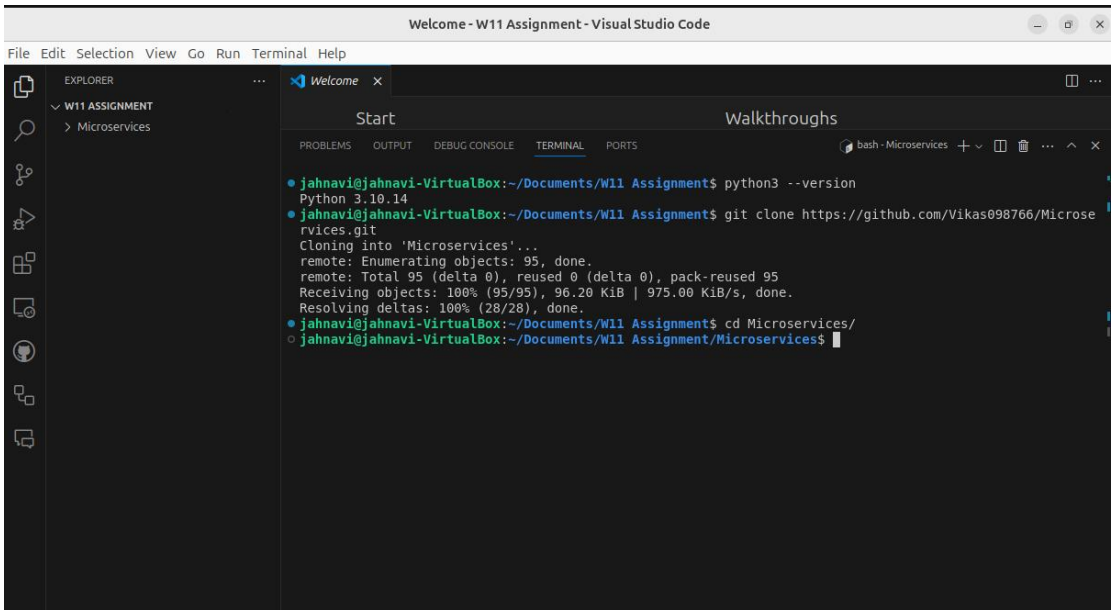
Solution : Python is set up, please find below snap with python version




4. Clone this Github repository
https://github.com/Vikas098766/Microservices.git

Solution : Cloned using the command

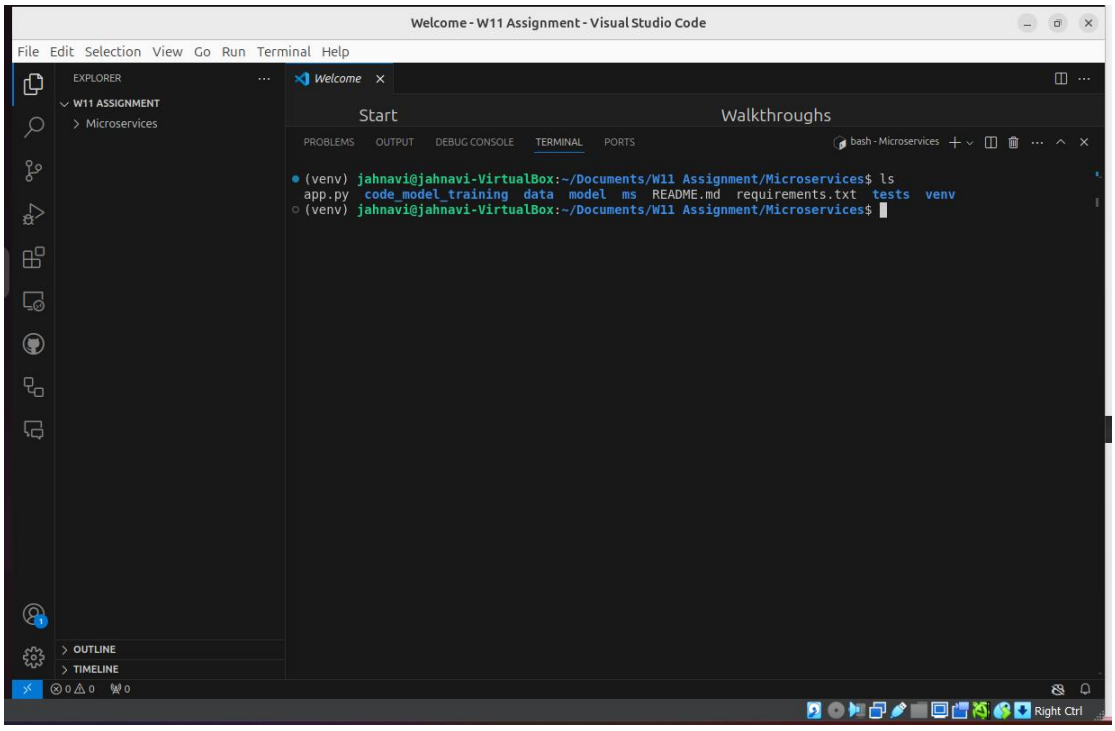
- git clone https://github.com/Vikas098766/Microservices.git




5. Create a Virtual Environment.
Solution : Created Virtual Environment using commands

- python3-m venv venv
- source venv/bin/activate

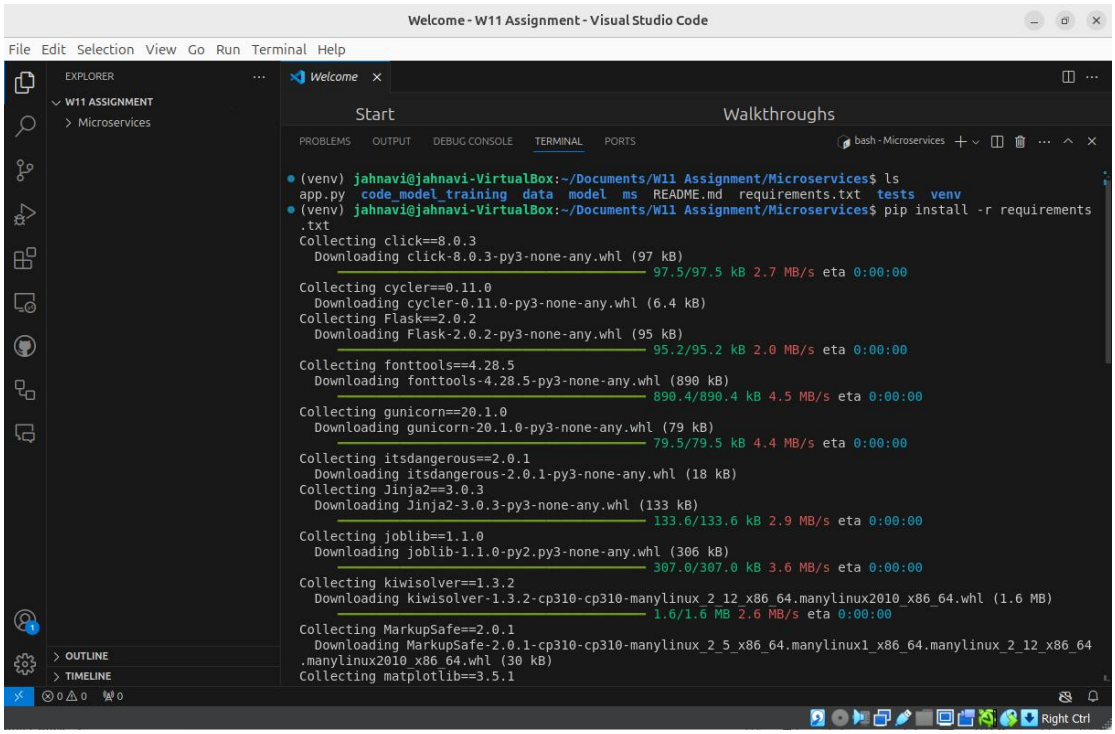
```
jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$ python3 -m venv venv
jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$ spurce venv/bin/activate
bash: spurce: command not found
jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$ source venv/bin/activate
(venv) jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$
```

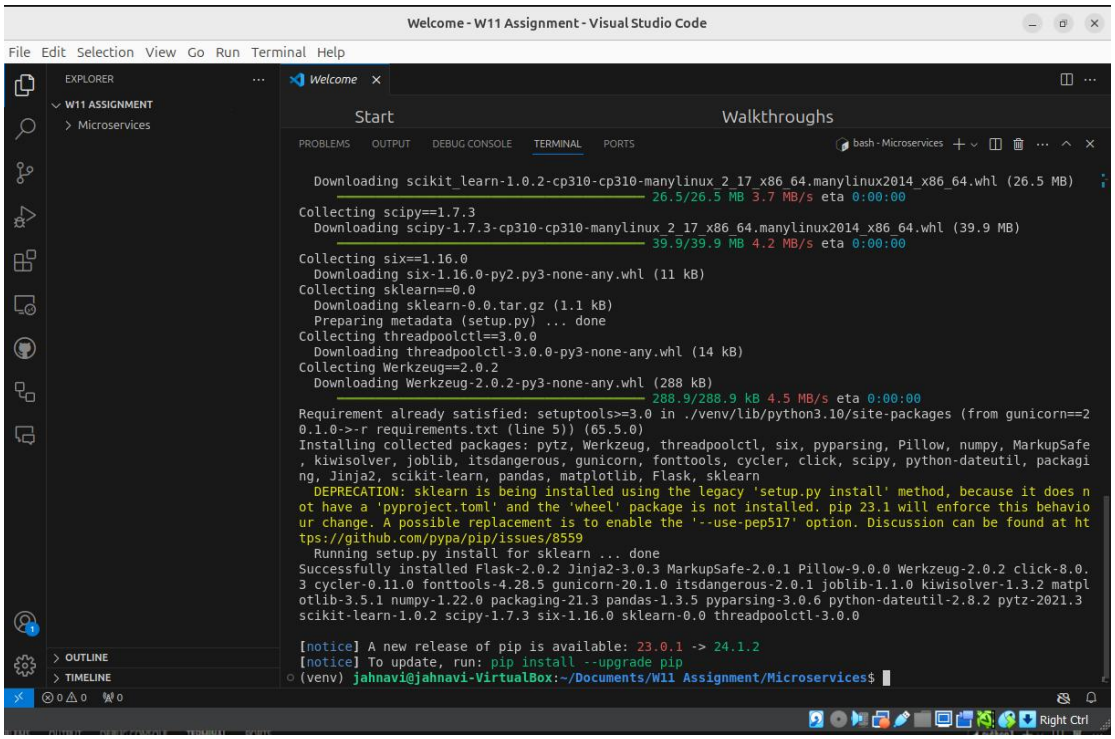



6. Install the dependencies from requirements.txt file.

Solution : Installed all dependencies present in requirements.txt file using the command

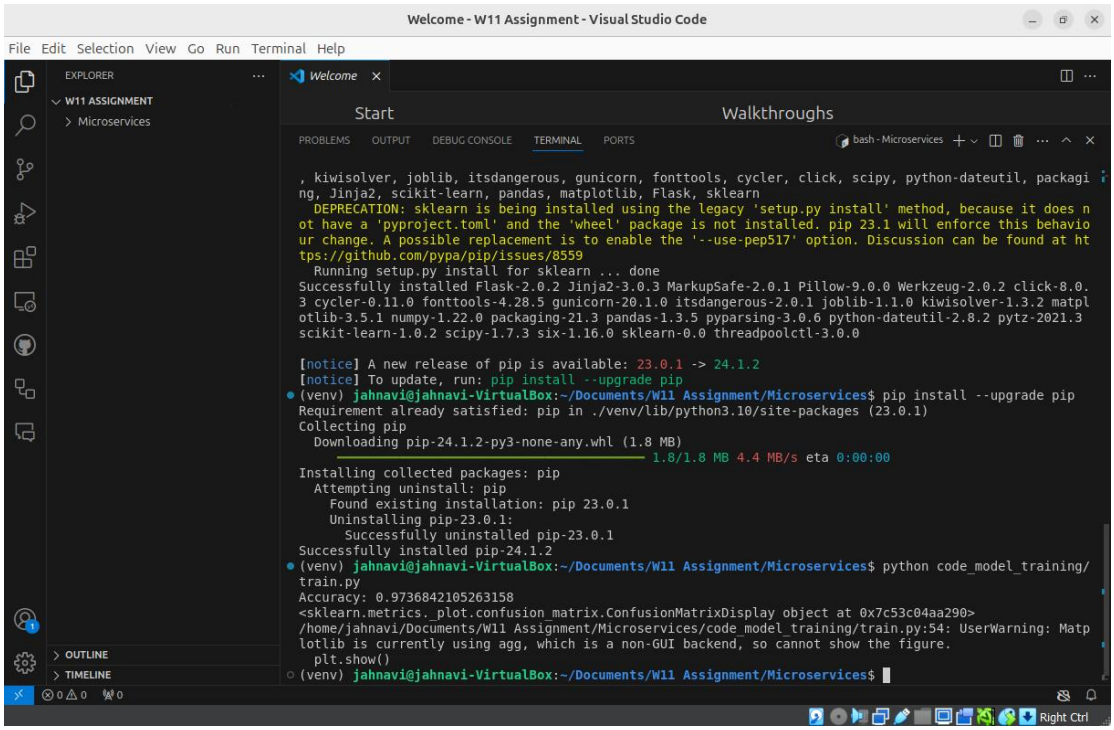
- Pip install -r requirements.txt




7. Train and save the model.

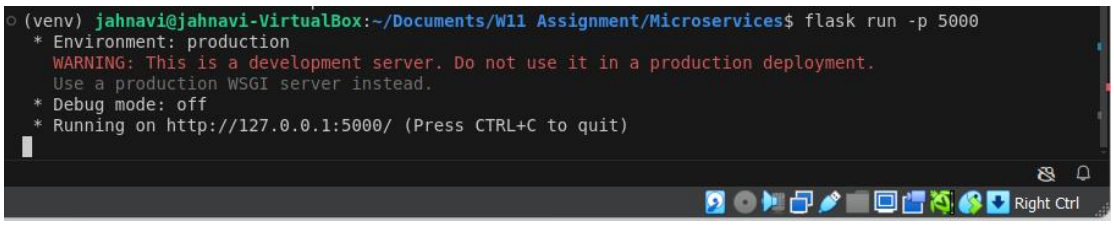
Solution : Trained and saved the model. ● Command: python code_model_training/train.py




8. Test the Flask web application.

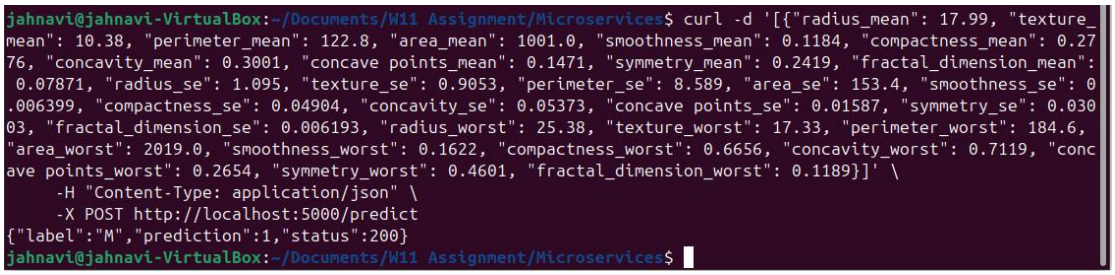
Solution : Tested web application by running the command.

- flask run -p 5000




9: Tested the the end point /info Command: curl -X GET http://localhost:5000/info Command: curl -X GET http://localhost:5000/health

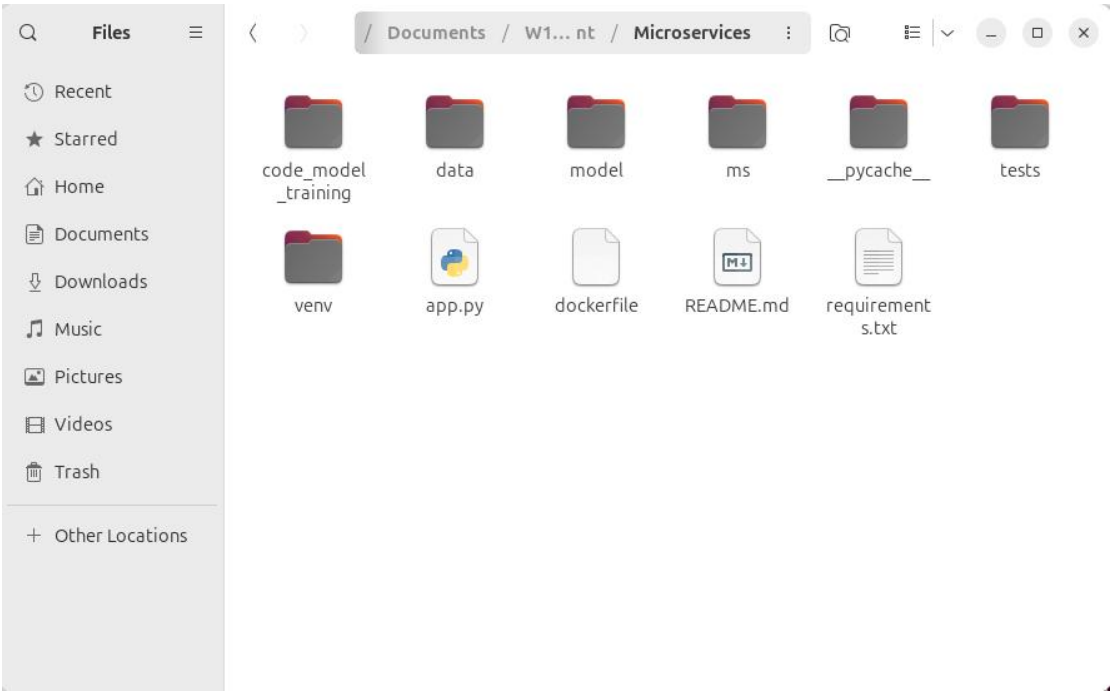
**Command USED:** curl-d '[{"radius_mean": 17.99, "texture_mean": 10.38, "perimeter_mean": 122.8, "area_mean": 1001.0, "smoothness_mean": 0.1184, "compactness_mean": 0.2776, "concavity_mean": 0.3001, "concave points_mean": 0.1471, "symmetry_mean": 0.2419, "fractal_dimension_mean": 0.07871, "radius_se": 1.095, "texture_se": 0.9053, "perimeter_se": 8.589, "area_se": 153.4, "smoothness_se": 0.006399, "compactness_se": 0.04904, "concavity_se": 0.05373, "concave points_se": 0.01587, "symmetry_se": 0.03003, "fractal_dimension_se": 0.006193, "radius_worst": 25.38, "texture_worst": 17.33, "perimeter_worst": 184.6, "area_worst": 2019.0, "smoothness_worst": 0.1622, "compactness_worst": 0.6656, "concavity_worst": 0.7119, "concave points_worst": 0.2654, "symmetry_worst": 0.4601, "fractal_dimension_worst": 0.1189}]' \-H "Content-Type: application/json" \-X POST http://0.0.0.0:5000/predict



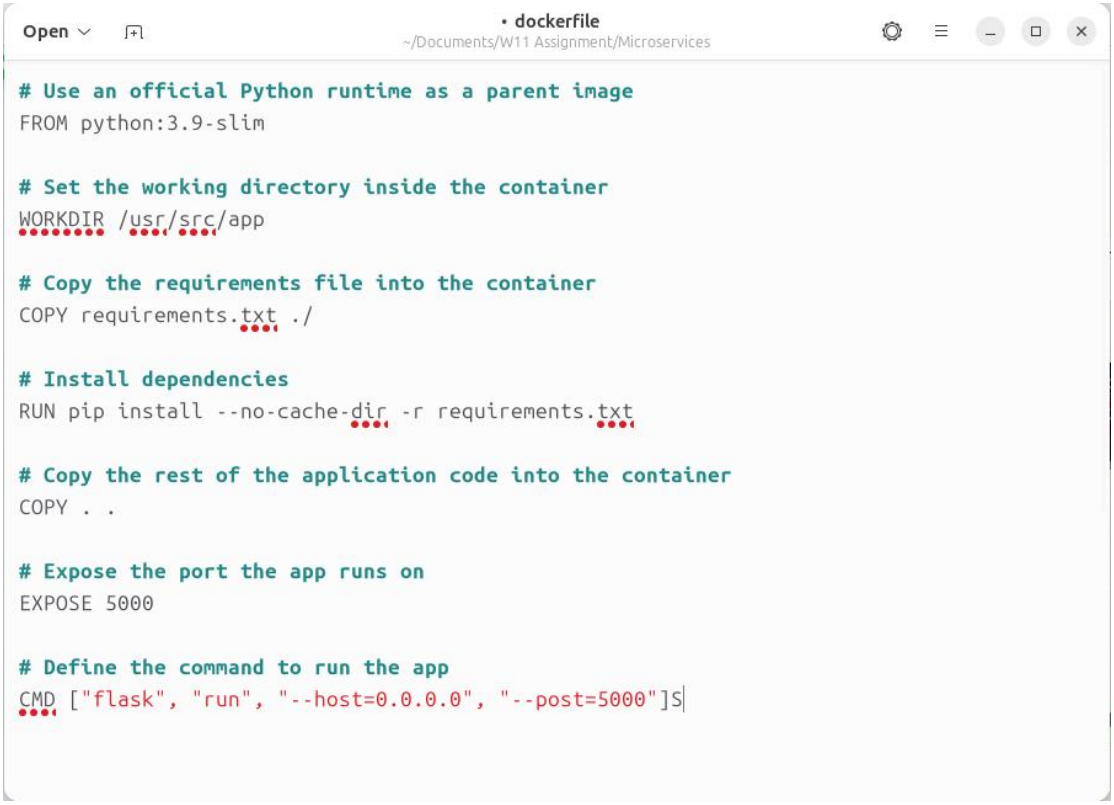

API ENDPOINT with /predict got the output as
{“label” :” M” ,” prediction” :1,” status” :200}

10. Steps to create a docker image.

1. Created the text file named dockerfile using the command as touch dockerfile

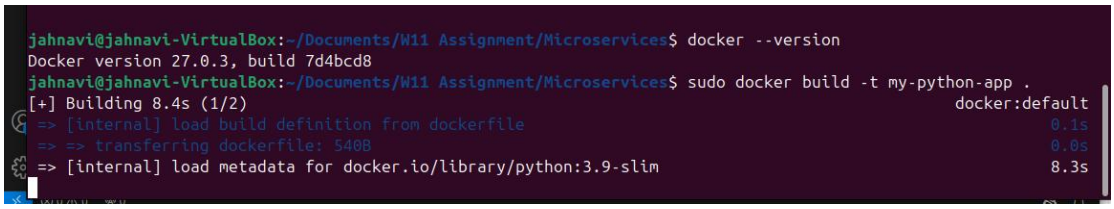


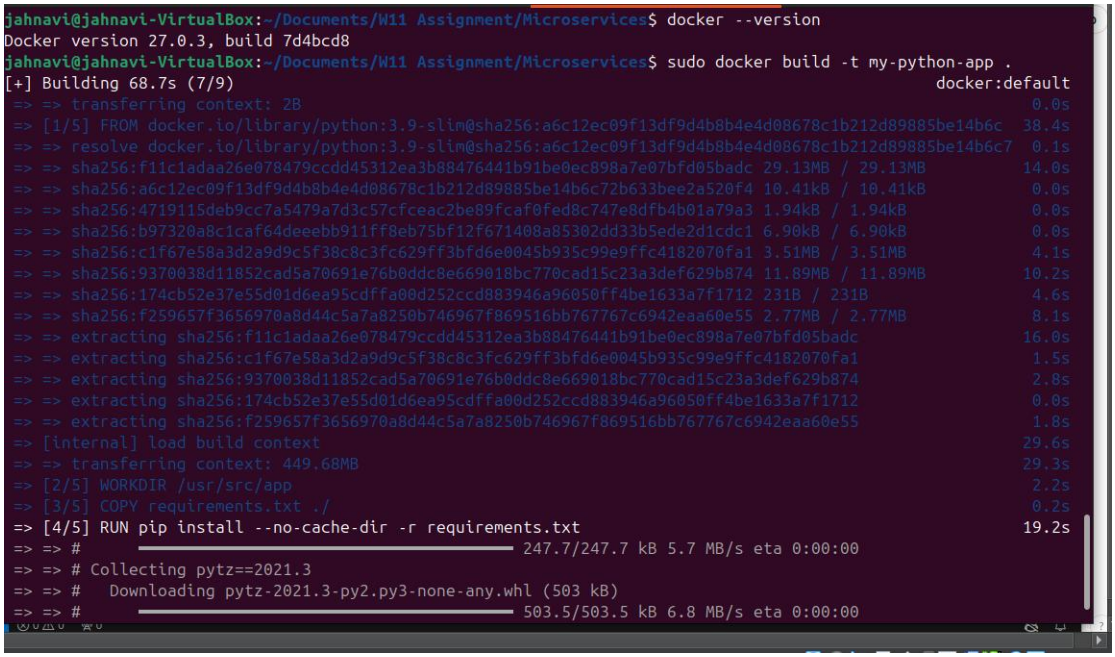

2. Within the txt file adding the following content within it.

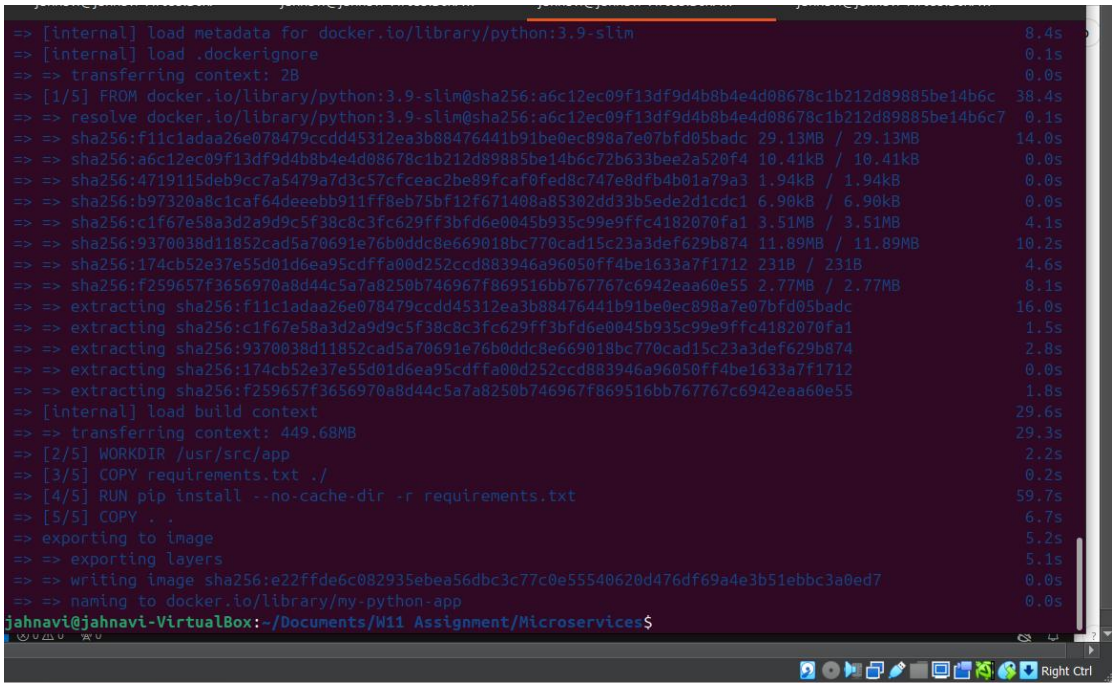



3. Build the docker image with the name as my-python-app

● Command: sudo docker build-t my-python-app .

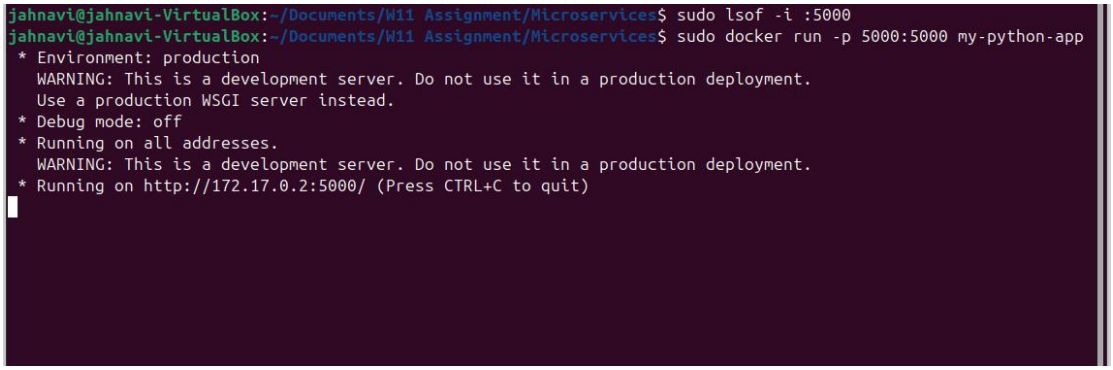


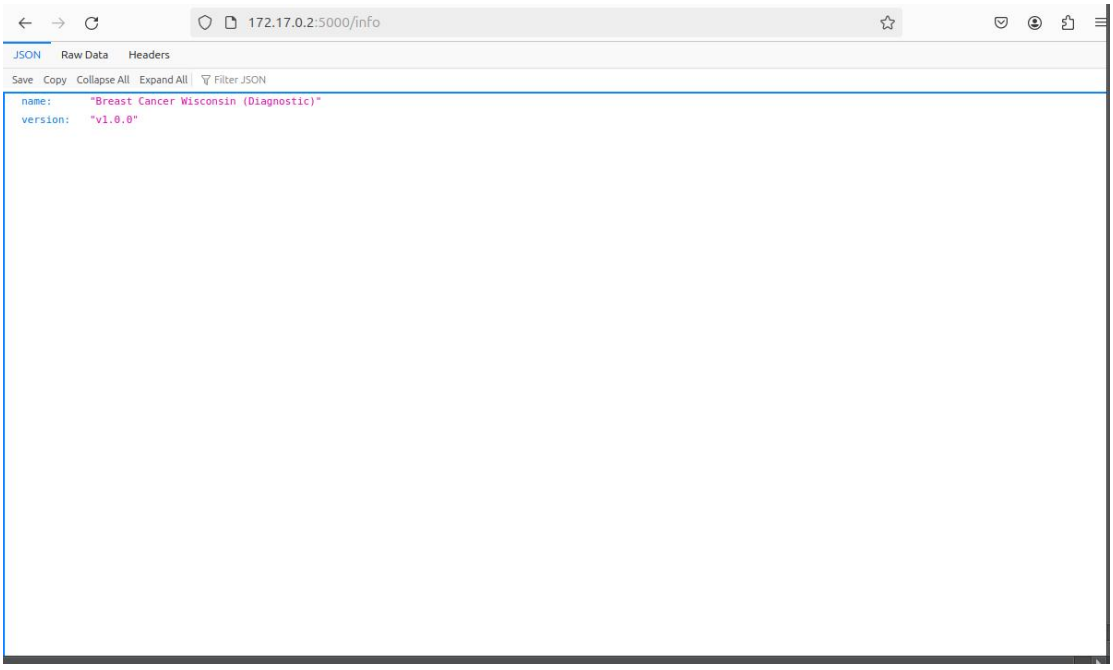
4. Run the Docker Container

- Command:sudo docker run-p 5000:5000 my-python-app

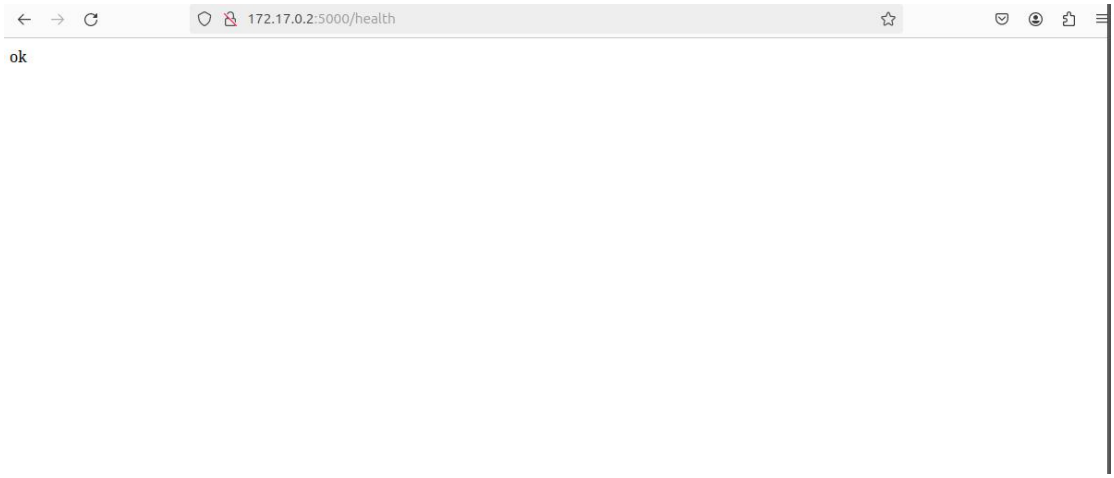



11. To check the Docker image service locally with the help of POSTMAN end points as

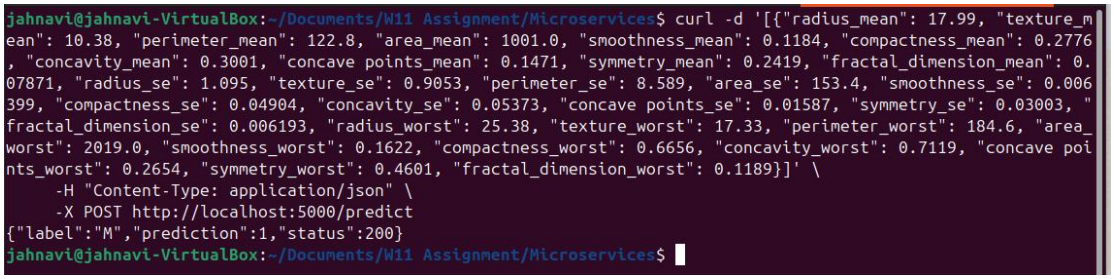
1. /info

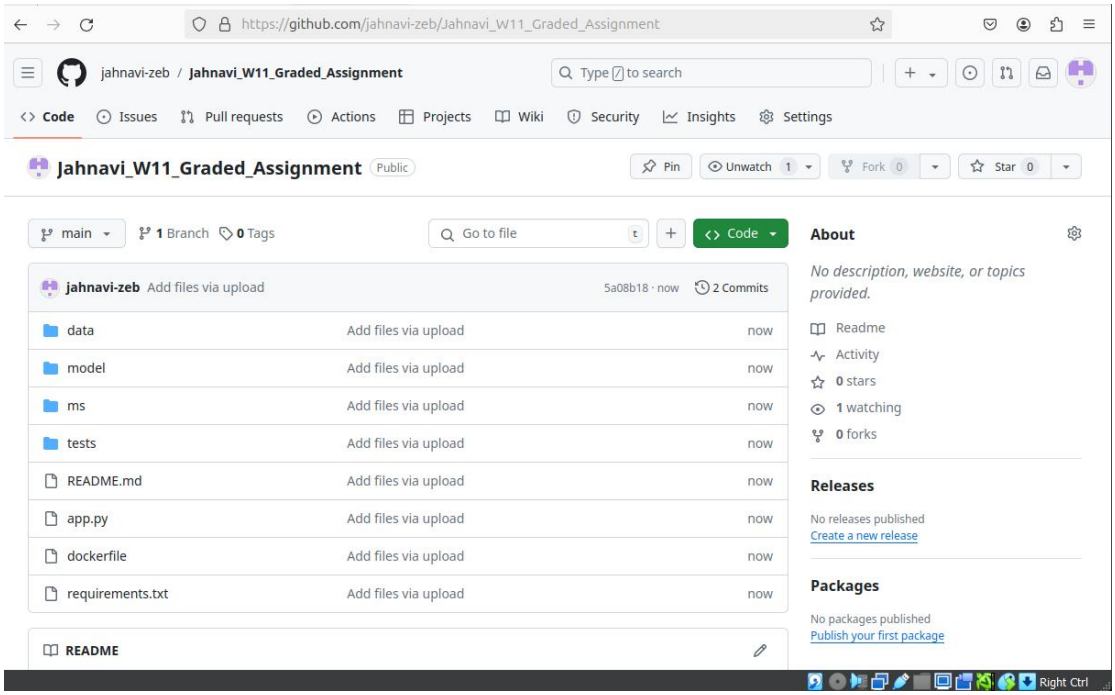



2. /health


172.17.0.2:5000/health
ok

3. /predict


jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$ curl -d '[{"radius_mean": 17.99, "texture_mean": 10.38, "perimeter_mean": 122.8, "area_mean": 1001.0, "smoothness_mean": 0.1184, "compactness_mean": 0.2776, "concavity_mean": 0.3001, "concave points_mean": 0.1471, "symmetry_mean": 0.2419, "fractal_dimension_mean": 0.07871, "radius_se": 1.095, "texture_se": 0.9053, "perimeter_se": 8.589, "area_se": 153.4, "smoothness_se": 0.006399, "compactness_se": 0.04904, "concavity_se": 0.05373, "concave points_se": 0.01587, "symmetry_se": 0.03003, "fractal_dimension_se": 0.006193, "radius_worst": 25.38, "texture_worst": 17.33, "perimeter_worst": 184.6, "area_worst": 2019.0, "smoothness_worst": 0.1622, "compactness_worst": 0.6656, "concavity_worst": 0.7119, "concave points_worst": 0.2654, "symmetry_worst": 0.4601, "fractal_dimension_worst": 0.1189}]' \
-H "Content-Type: application/json" \
-X POST http://localhost:5000/predict
{"label":"M","prediction":1,"status":200}
jahnavi@jahnavi-VirtualBox:~/Documents/W11 Assignment/Microservices$


https://github.com/jahnavi-zeb/Jahnavi_W11_Graded_Assignment
jahnavi-zeb / Jahnavi_W11_Graded_Assignment
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Jahnavi_W11_Graded_Assignment
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
jahnavi-zeb Add files via upload
5a08b18 · now
2 Commits
data Add files via upload now
model Add files via upload now
ms Add files via upload now
tests Add files via upload now
README.md Add files via upload now
app.py Add files via upload now
dockerfile Add files via upload now
requirements.txt Add files via upload now
README
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Right Ctrl