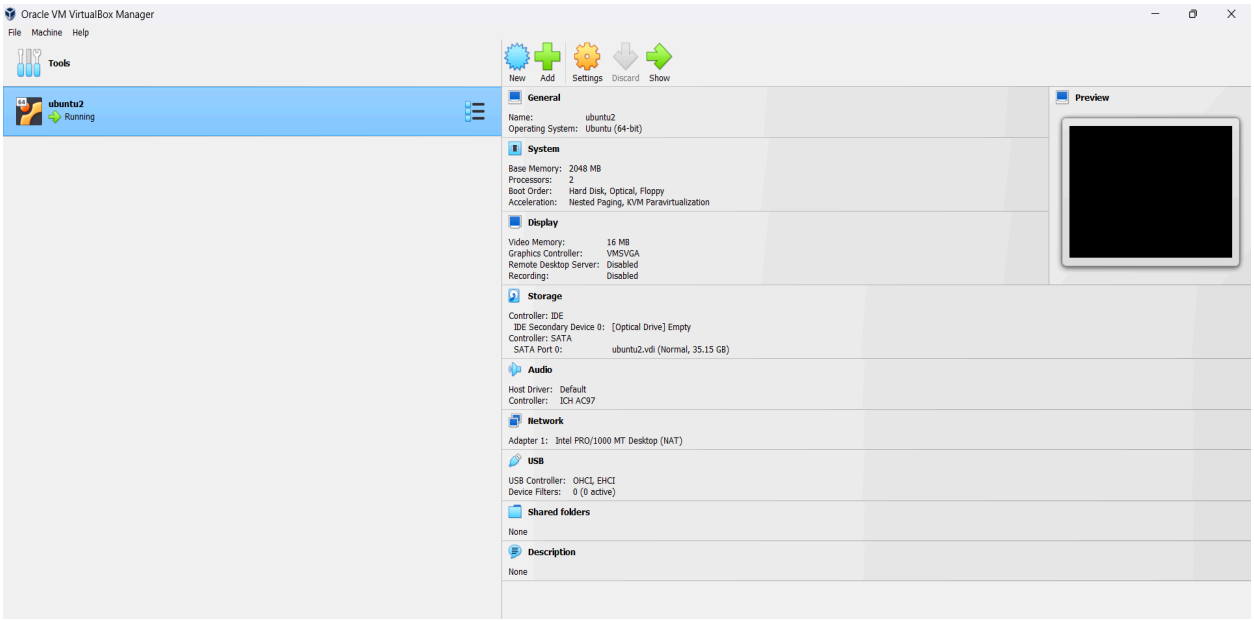
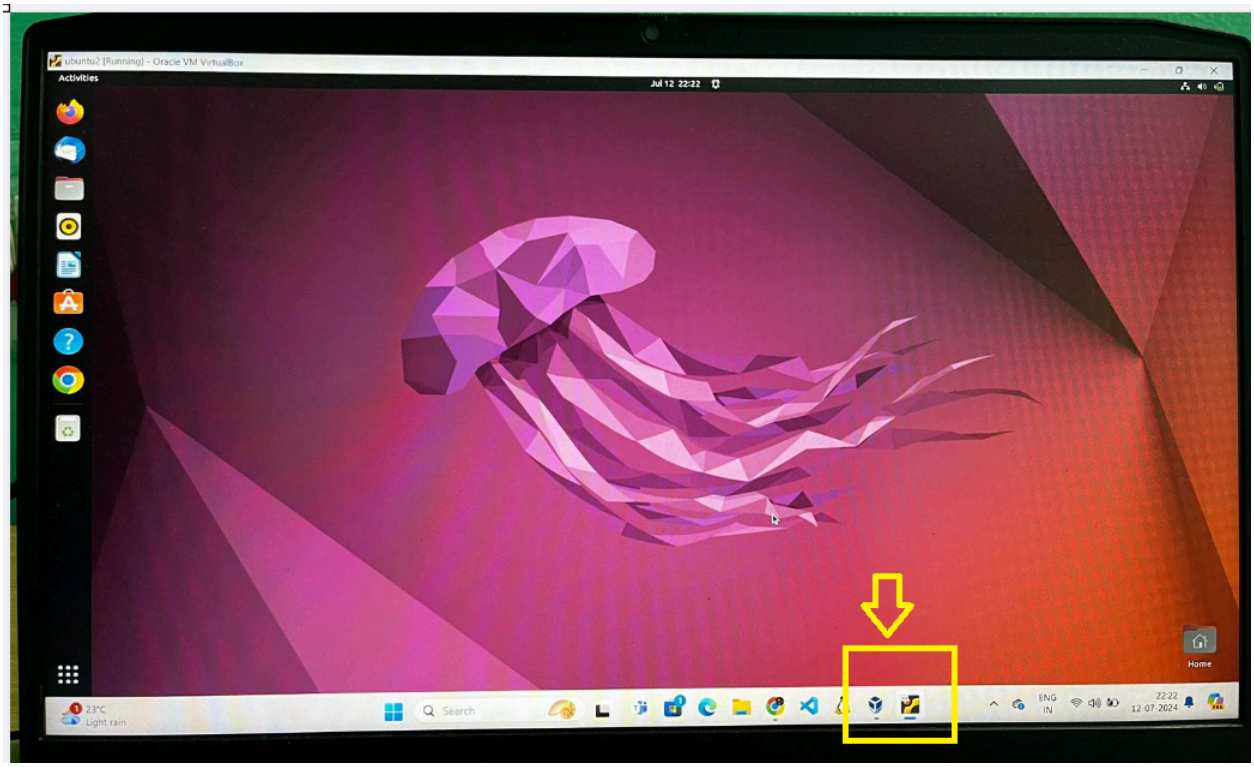
WEEK 10 - WEEK 12 ASSIGNMENT

Submitted By : Karthik S Deshpande

1. **Host a Ubuntu Virtual Machine using Oracle VM Virtual Box**

Solution : Hosted Ubuntu Virtual machine using oracle VM Virtual box

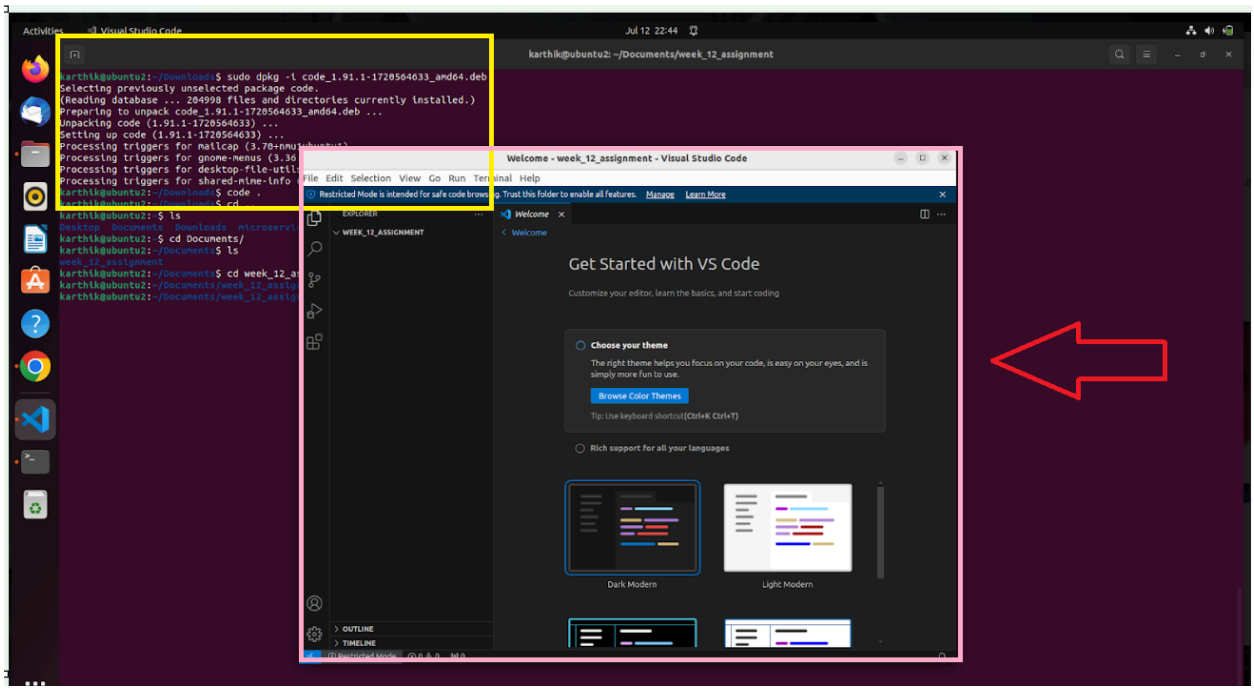




2. **Set up Visual Studio code on Ubuntu VM.**

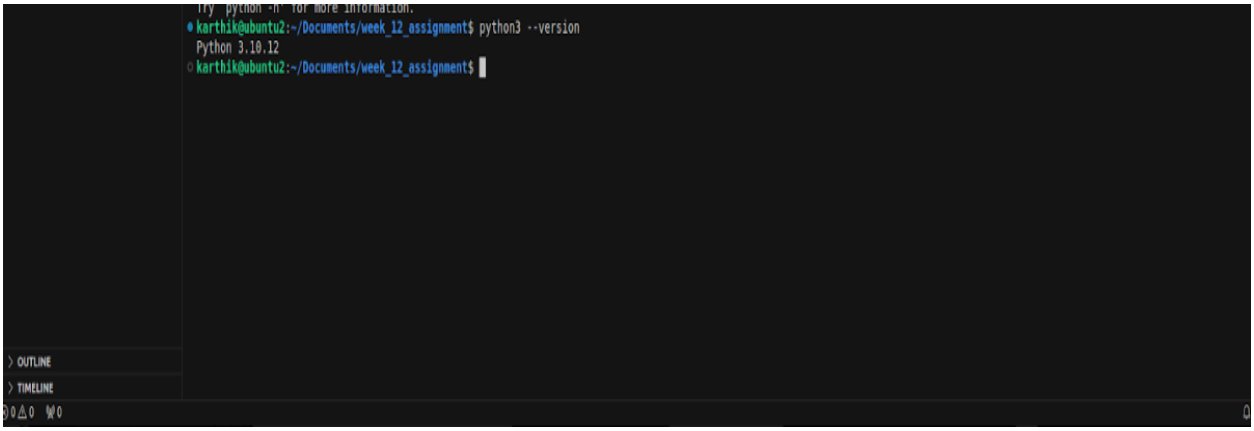
Solution : Downloaded the VScode installation file from https://code.visualstudio.com

In the terminal of Linux ran command to install VSCODE.



3. **Set up Python**

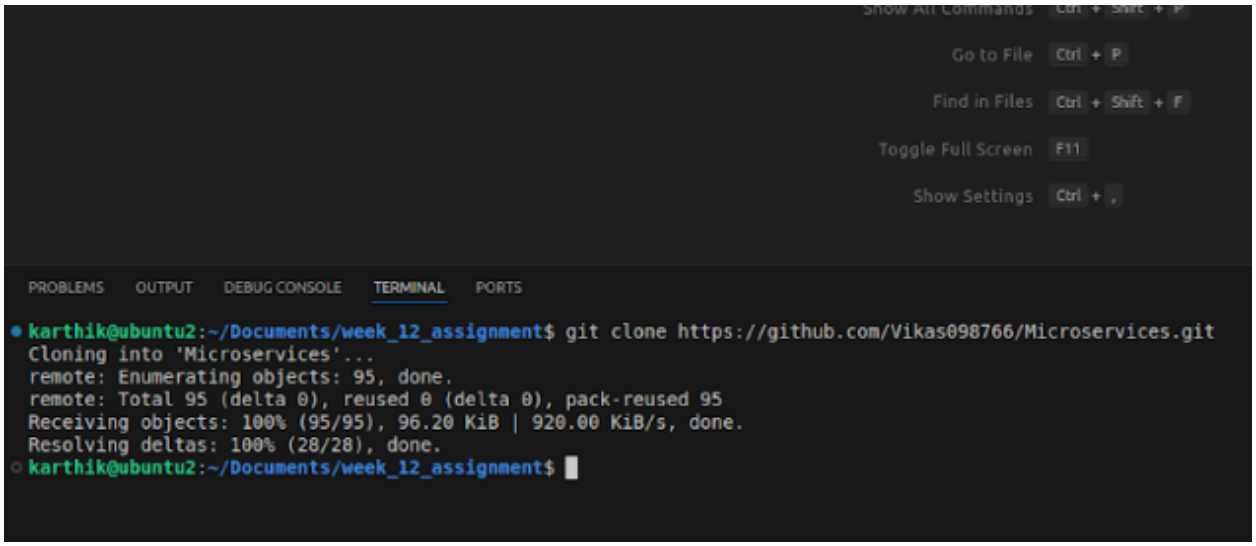
Solution : Python is set up, please find below snap with python version



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

Solution : Cloned using the command

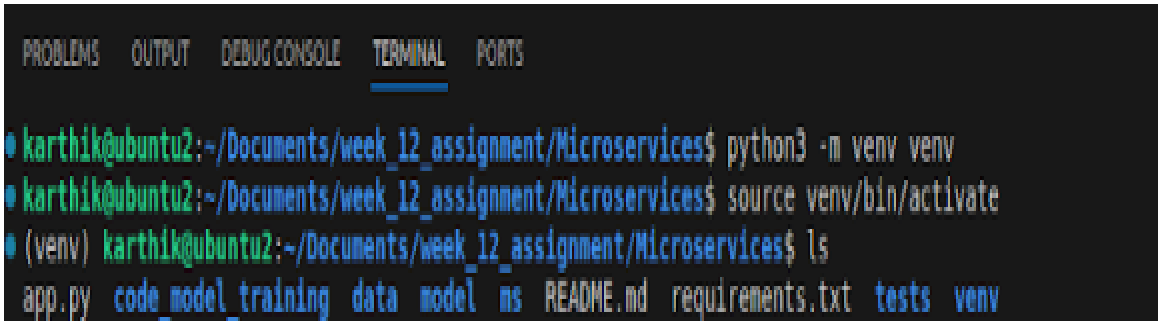
- **git clone https://github.com/Vikas098766/Microservices.git**



5. Create a Virtual Environment.

Solution : Created Virtual Environment using commands

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



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

```
(venv) karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ pip install -r requirements.txt
Collecting click==8.0.3
  Using cached click-8.0.3-py3-none-any.whl (97 kB)
Collecting cycler==0.11.0
  Using cached cycler-0.11.0-py3-none-any.whl (6.4 kB)
Collecting Flask==2.0.2
  Using cached Flask-2.0.2-py3-none-any.whl (95 kB)
Collecting fonttools==4.28.5
  Using cached fonttools-4.28.5-py3-none-any.whl (890 kB)
Collecting gunicorn==20.1.0
  Using cached gunicorn-20.1.0-py3-none-any.whl (79 kB)
Collecting itsdangerous==2.0.1
  Using cached itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting Jinja2==3.0.3
  Using cached Jinja2-3.0.3-py3-none-any.whl (133 kB)
Collecting joblib==1.1.0
  Using cached joblib-1.1.0-py2.py3-none-any.whl (306 kB)
Collecting kiwisolver==1.3.2
  Using cached kiwisolver-1.3.2-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6 MB)
Collecting MarkupSafe==2.0.1
  Using cached MarkupSafe-2.0.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (30 kB)
Collecting matplotlib==3.5.1
  Using cached matplotlib-3.5.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.9 MB)
Collecting numpy==1.22.0
  Using cached numpy-1.22.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.8 MB)
Collecting packaging==21.3
  Using cached packaging-21.3-py3-none-any.whl (40 kB)
Collecting pandas==1.3.5
  Using cached pandas-1.3.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
Collecting Pillow==9.0.0
```

7. Train and save the model.

Solution : Trained and saved the model.

- Command : python code_model_training/train.py

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                                    python3

(venv) karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ python code_model_training/train.py
Accuracy: 0.9736842105263158
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7ca36440f310>
/home/karthik/Documents/week_12_assignment/Microservices/code_model_training/train.py:54: UserWarning: Matplotlib is currently using agg, which is a non-GUI backend, so cannot sh
ow the figure.
  plt.show()
```

8. Test the Flask web application.

Solution : Tested web application by running the command.

- flask run -p 5000

```
plt.show()
(venv) karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ flask run -p 5000
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

9. **Test the application and make predictions using the example calls available in the folder /tests**

Solution : Tested the the end point **/info**

Command : curl -X GET http://localhost:5000/info

Command : curl -X GET http://localhost:5000/health

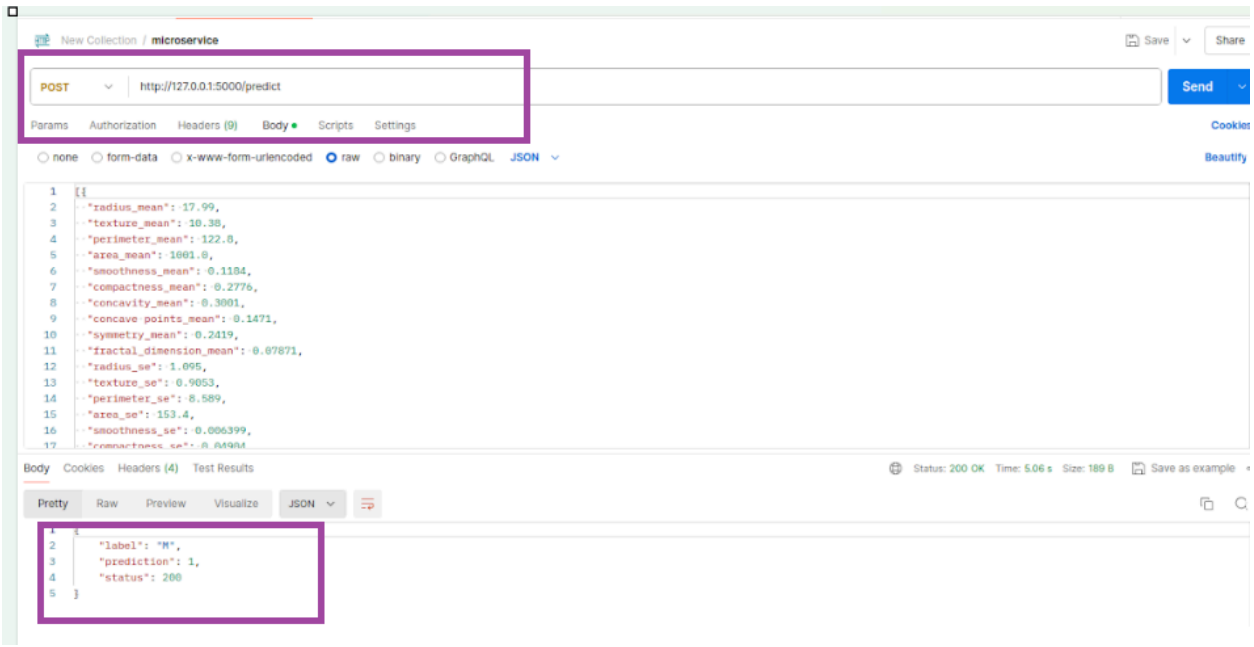
```
karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ curl -X GET http://localhost:5000/info
{"name":"Breast Cancer Wisconsin (Diagnostic)","version":"v1.0.0"}
karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ curl -X GET http://localhost:5000/health
okkarthik@ubuntu2:~/Documents/week_12_assignment/Microservices$
```

Command : 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

```
karthik@ubuntu2:~/Documents/week_12_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://0.0.0.0:5000/predict
{"label":"M","prediction":1,"status":200}
```

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

Verified using postman for more visibility



10.Create a docker image containing everything needed to run the application.

Solution : Steps to create a docker image.

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

```
karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ touch dockerfile
```

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

Open ~/Documents/week_12_assignment/Microservices **dockerfile** Save

```
# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /usr/src/app

# Copy the requirements file into the container
COPY requirements.txt ./

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application code into the container
COPY . .

# Expose the port the app runs on
EXPOSE 5000

# Define the command to run the app
CMD ["flask", "run", "--host=0.0.0.0", "--port=5000"]
```

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

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

```
karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ sudo docker build -t my-python-app .
[+] Building 246.8s (10/10) FINISHED
 => [internal] load build definition from dockerfile
 => => transferring dockerfile: 538B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3.9-slim
 => [1/5] FROM docker.io/library/python:3.9-slim@sha256:a6c12ec09f13df9d4b8b4e4d08678c1b212d89885be14b6c72b633bee2a520f4
 => => resolve docker.io/library/python:3.9-slim@sha256:a6c12ec09f13df9d4b8b4e4d08678c1b212d89885be14b6c72b633bee2a520f4
 => => sha256:a6c12ec09f13df9d4b8b4e4d08678c1b212d89885be14b6c72b633bee2a520f4 10.41kB / 10.41kB
 => => sha256:4719115deb9cc7a5479a7d3c57cfceac2be89fcaf0fed8c747e8dfb4b01a79a3 1.94kB / 1.94kB
 => => sha256:b97320a8c1caf64deeebb911ff8eb75bf12f671408a85302dd33b5ede2d1cdc1 6.90kB / 6.90kB
 => => sha256:f11c1adaa26e078479ccdd45312ea3b88476441b91be0ec898a7e07bfd05badc 29.13MB / 29.13MB
 => => sha256:c1f67e58a3d2a9d9c5f38c8c3fc629ff3bfd6e0045b935c99e9ffc4182070fa1 3.51MB / 3.51MB
 => => sha256:9370038d11852cad5a70691e76b0ddc8e669018bc770cad15c23a3def629b874 11.89MB / 11.89MB
```

4. Run the Docker Container

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

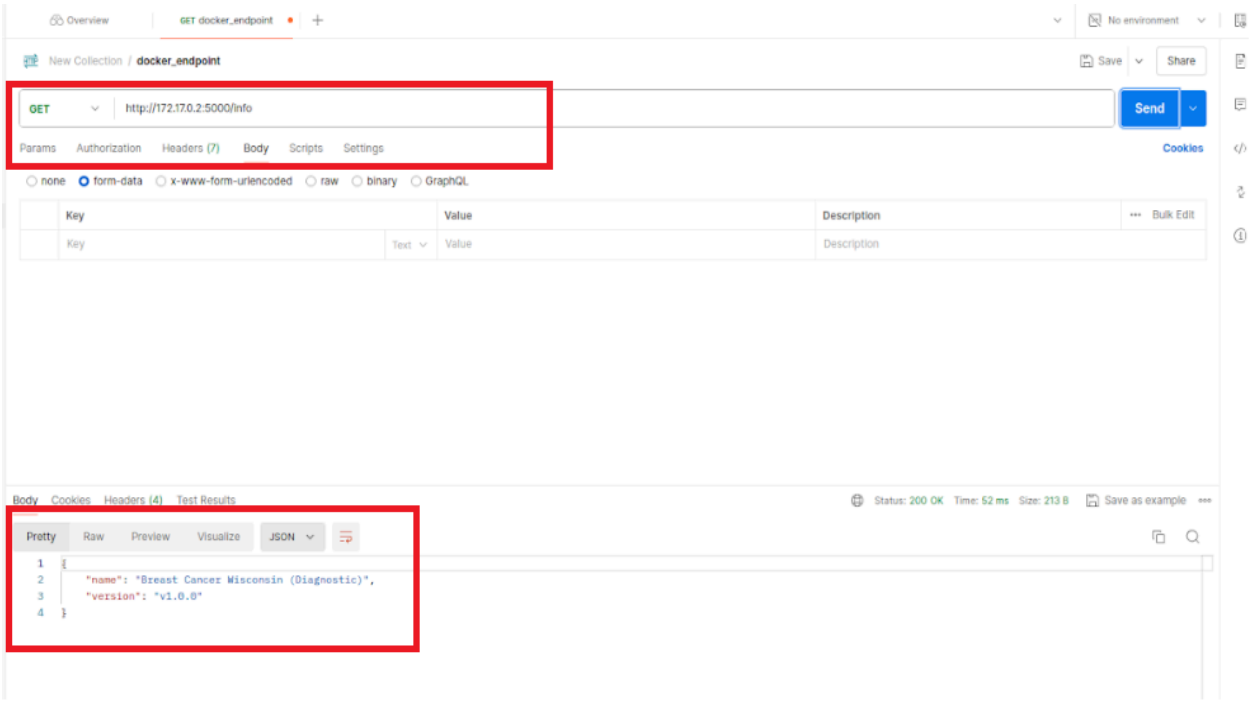
```
karthik@ubuntu2:~/Documents/week_12_assignment/Microservices$ sudo docker run -p 5000:5000 my-python-app
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```

Docker image is running successfully.

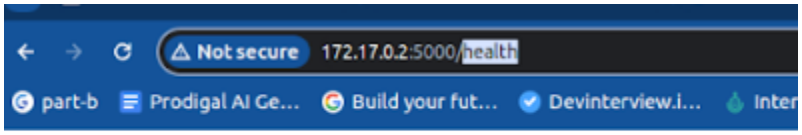
11. Run the containerized application as a prediction service and test it locally by passing some example calls and get the prediction.

Solution : To check the Docker image service locally with the help of POSTMAN end points as

- **/info**

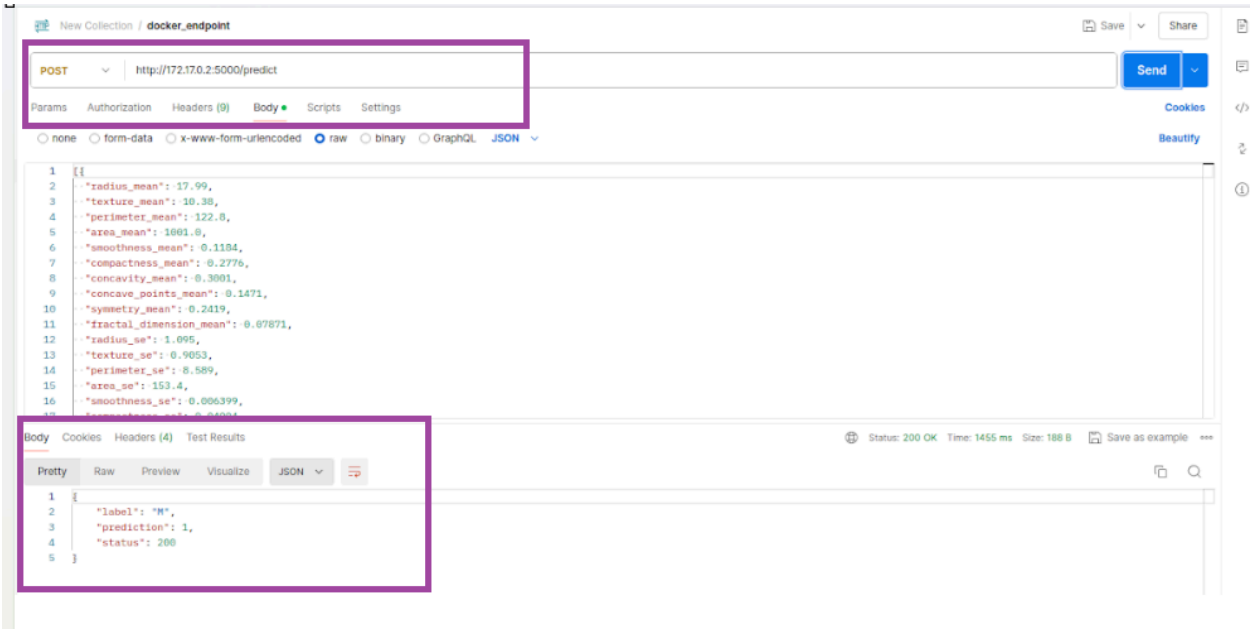


- **/health**



ok

- **/predict**



Passed parameters as

[

{

"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
}
]

********** THE END **********