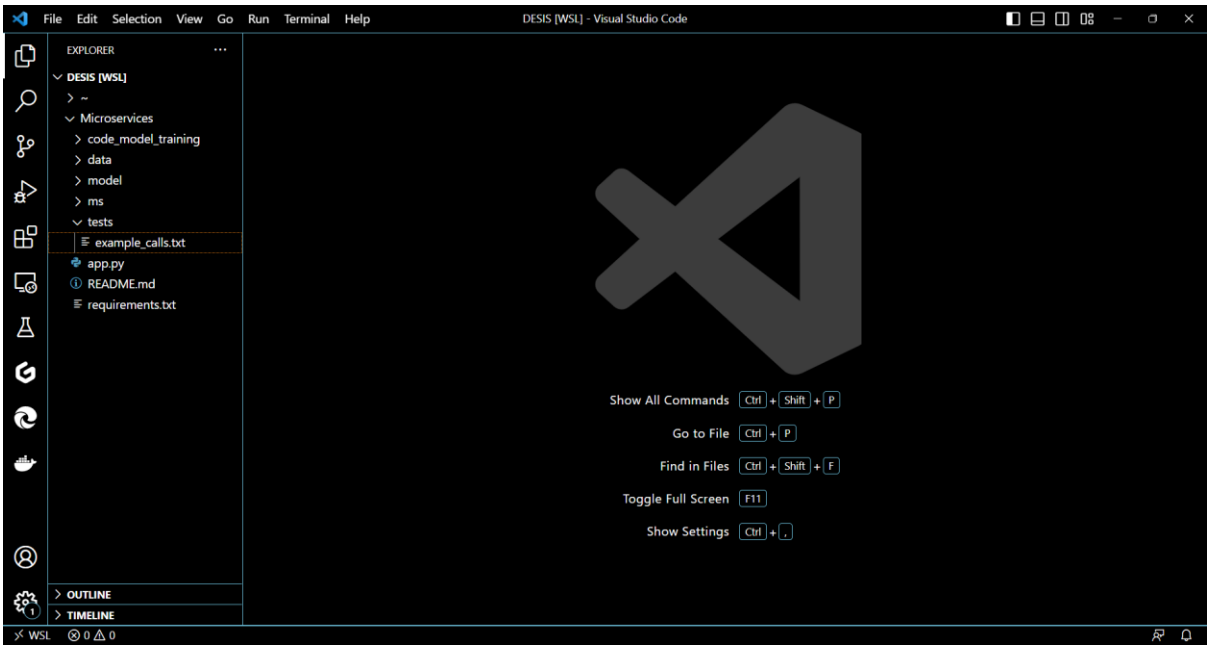
# Mayur Pawar

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

:



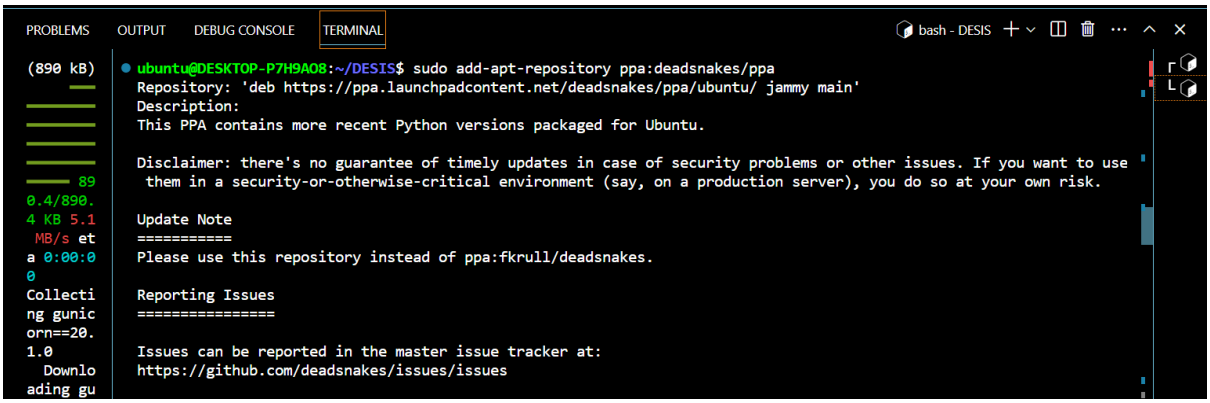

2) Set up Visual Studio code on Ubuntu VM.

:

```
ubuntu@DESKTOP-P7H9AO8:~/DESIS$ sudo apt update
[sudo] password for ubuntu:
Sorry, try again.
[sudo] password for ubuntu:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [634 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [855 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [149 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.0 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [656 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [209 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [668 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [104 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [764 kB]
```

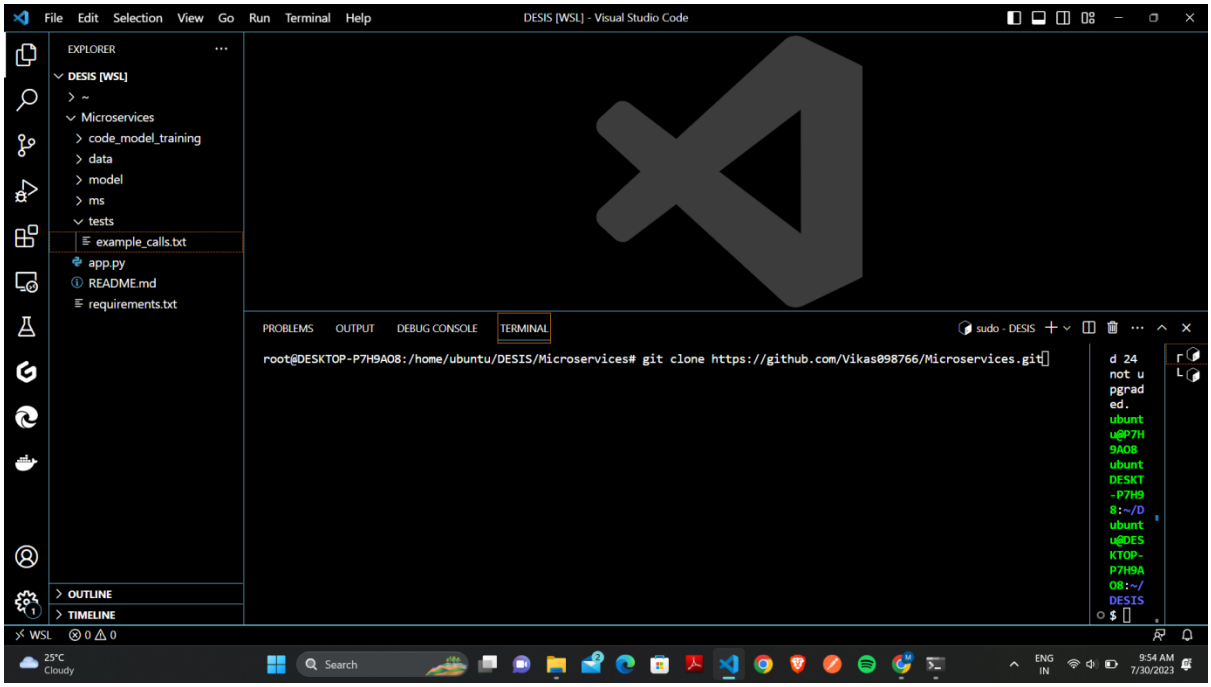
3) Set up Python.

:



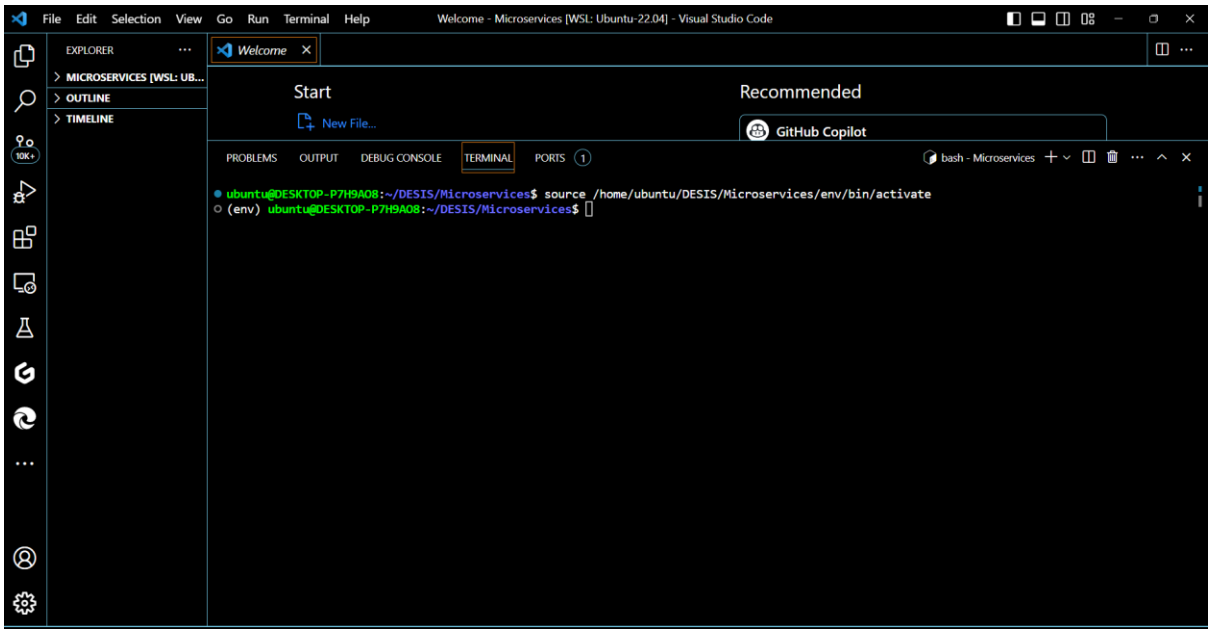

4) Clone this Github repository.

:




5) Create a Virtual Environment.

:




6) Install the dependencies from requirements.txt file.

:

```
PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu-22.04\home\ubuntu\DESIS\Microservices> pip install -r requirements.txt
DEPRECATION: Loading egg at c:\python311\lib\site-packages\vboxapi-1.0-py3.11.egg is deprecated. pip 23.3 will enforce this behaviour change. A possible rep
lacement is to use pip for package installation..
Collecting click==8.0.3 (from -r requirements.txt (line 1))
  Using cached click-8.0.3-py3-none-any.whl (97 kB)
Requirement already satisfied: cycler==0.11.0 in c:\python311\lib\site-packages (from -r requirements.txt (line 2)) (0.11.0)
Collecting Flask==2.0.2 (from -r requirements.txt (line 3))
  Using cached Flask-2.0.2-py3-none-any.whl (95 kB)
Collecting fonttools==4.28.5 (from -r requirements.txt (line 4))
  Downloading fonttools-4.28.5-py3-none-any.whl (890 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 890.4/890.4 kB 4.3 MB/s eta 0:00:00
Collecting gunicorn==20.1.0 (from -r requirements.txt (line 5))
  Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.5/79.5 kB 4.3 MB/s eta 0:00:00
Collecting itsdangerous==2.0.1 (from -r requirements.txt (line 6))
  Using cached itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting Jinja2==3.0.3 (from -r requirements.txt (line 7))
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.6/133.6 kB 7.7 MB/s eta 0:00:00
Collecting joblib==1.1.0 (from -r requirements.txt (line 8))
  Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.0/307.0 kB 6.3 MB/s eta 0:00:00
Collecting kiwisolver==1.3.2 (from -r requirements.txt (line 9))
  Downloading kiwisolver-1.3.2.tar.gz (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.6/54.6 kB 3.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting MarkupSafe==2.0.1 (from -r requirements.txt (line 10))
  Using cached MarkupSafe-2.0.1.tar.gz (18 kB)
  Preparing metadata (setup.py) ... done
Collecting matplotlib==3.5.1 (from -r requirements.txt (line 11))
  Downloading matplotlib-3.5.1.tar.gz (35.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 35.3/35.3 MB 5.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting numpy==1.22.0 (from -r requirements.txt (line 12))
  Downloading numpy-1.22.0.zip (11.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.3/11.3 MB 6.6 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
```

7) Train and save the model

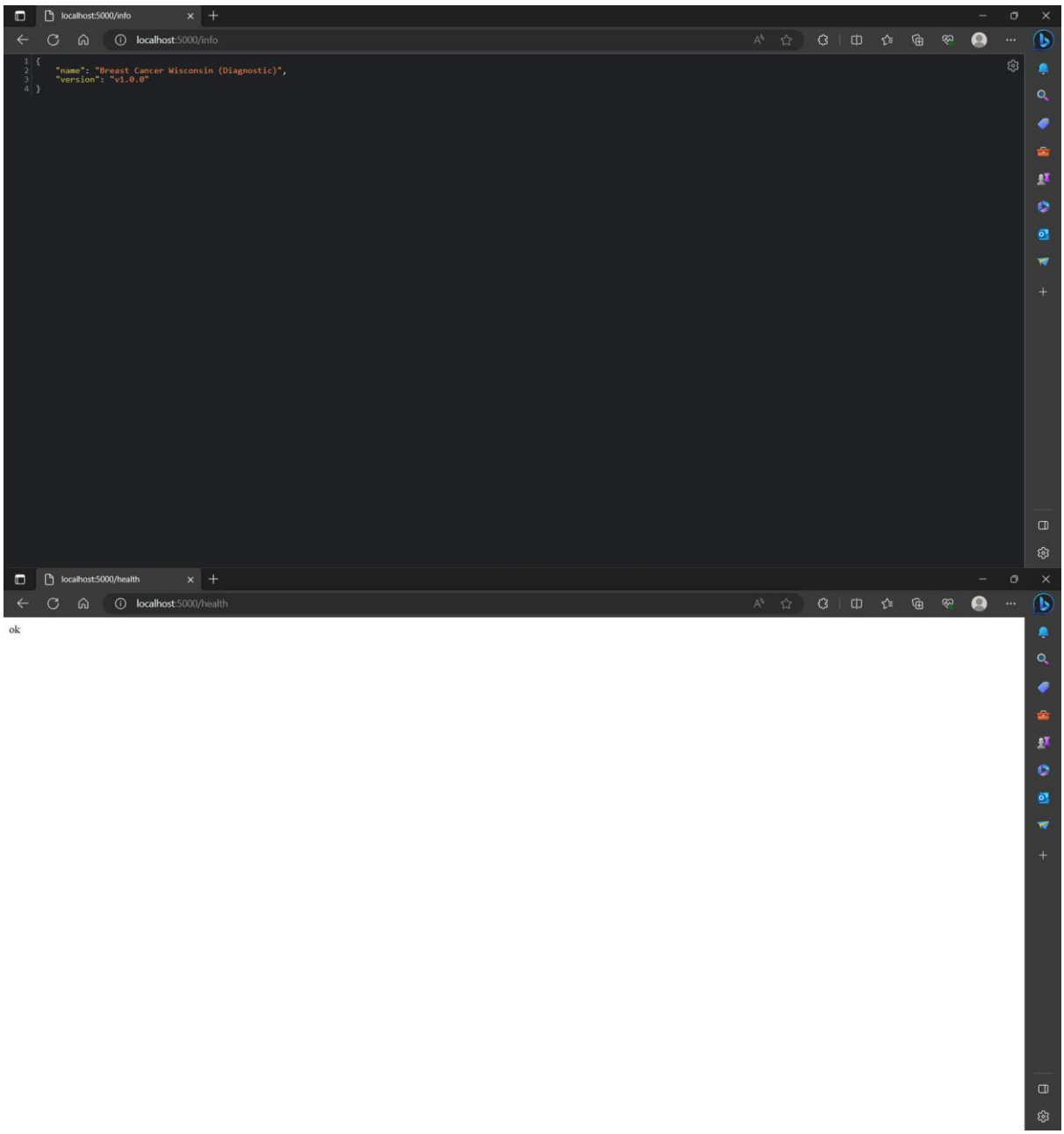
:

```
train.py U ×
code_model_training > train.py > ...
41      ('imputer', SimpleImputer
42      ('scaler', MinMaxScaler(f
43      ('model', ensemble)  # En
44  ])
45
46  # Train the model
47  pipe.fit(X_train, y_train)
48
49  # Test Accuracy
50  print("Accuracy: %s" % str(pi
51
52  # Plot confusion matrix
53  print(ConfusionMatrixDisplay.from_estimator(pipe, X_test, y_test))
54  plt.show()
55
56  # Export model
57  joblib.dump(pipe, gzip.open('model/model_binary.dat.gz', "wb"))

(method) def from_estimator(
    estimator: BaseEstimator,
    X: MatrixLike | ArrayLike,
    y: ArrayLike,
    *,
    labels: ArrayLike | None = None,
    sample_weight: ArrayLike | None = None,
    normalize: Literal['true', 'pred', 'all'] | None = None,
    display_labels: ArrayLike | None = None,
    include_values: bool = True,
    xticks_rotation: float | Literal['vertical', 'horizontal'] = "horizontal",
    values_format: str | None = None,
    cmap: Colormap | str = "viridis",
```

8) Test the Flask web application.

:




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

New Import

Overview | POST http://172.21.243.7:8C | GET http://localhost:8000/h | No Environment

DesignIS
- POST http://172.21.243.7:8000/predict
- GET http://localhost:8000/info
- GET http://localhost:8000/health
- New Collection

DesignIS / http://172.21.243.7:8000/predict

Save

POST http://172.21.243.7:5001/predict Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify


```
    "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
```


Body Cookies Headers (4) Test Results Status: 200 OK Time: 36 ms Size: 189 B Save as Example

Pretty Raw Preview Visualize JSON


```
{
  "label": "M",
  "prediction": 1,
  "status": 200
}
```


New Import

Overview | POST http://172.21.243.7:8C | GET http://localhost:8000/h | No Environment

DesignIS
- POST http://172.21.243.7:8000/predict
- GET http://localhost:8000/info
- GET http://localhost:8000/health
- New Collection

DesignIS / http://localhost:8000/info

Save

GET http://localhost:5001/info Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

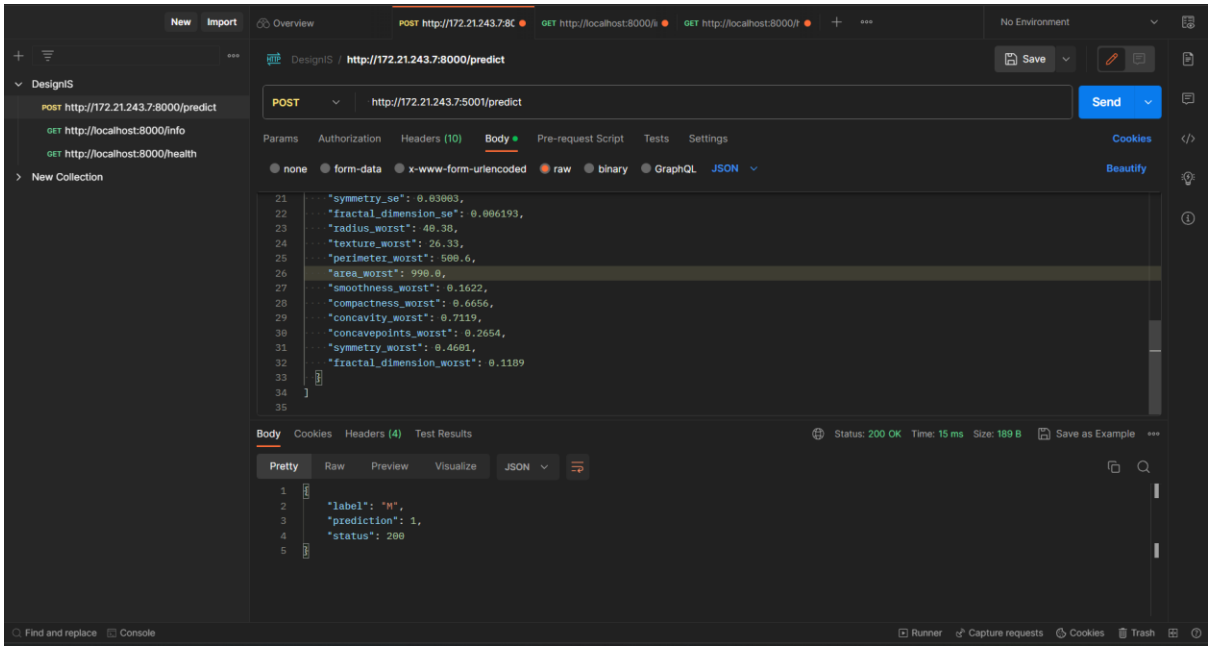
Query Params

| Key | Value | Description | Bulk Edit |
|---|---|---|---|
| Key | Value | Description | |

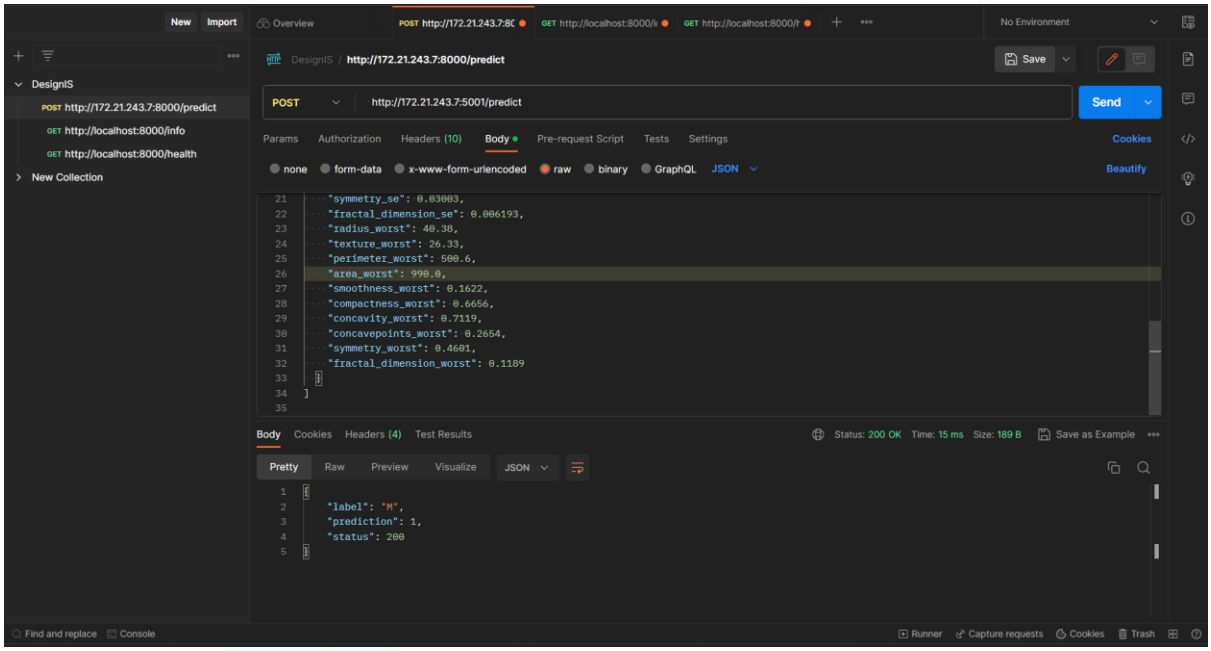
Body Cookies Headers (4) Test Results Status: 200 OK Time: 19 ms Size: 214 B Save as Example

Pretty Raw Preview Visualize JSON


```
{
  "name": "Breast Cancer Wisconsin (Diagnostic)",
  "version": "v1.0.0"
}
```

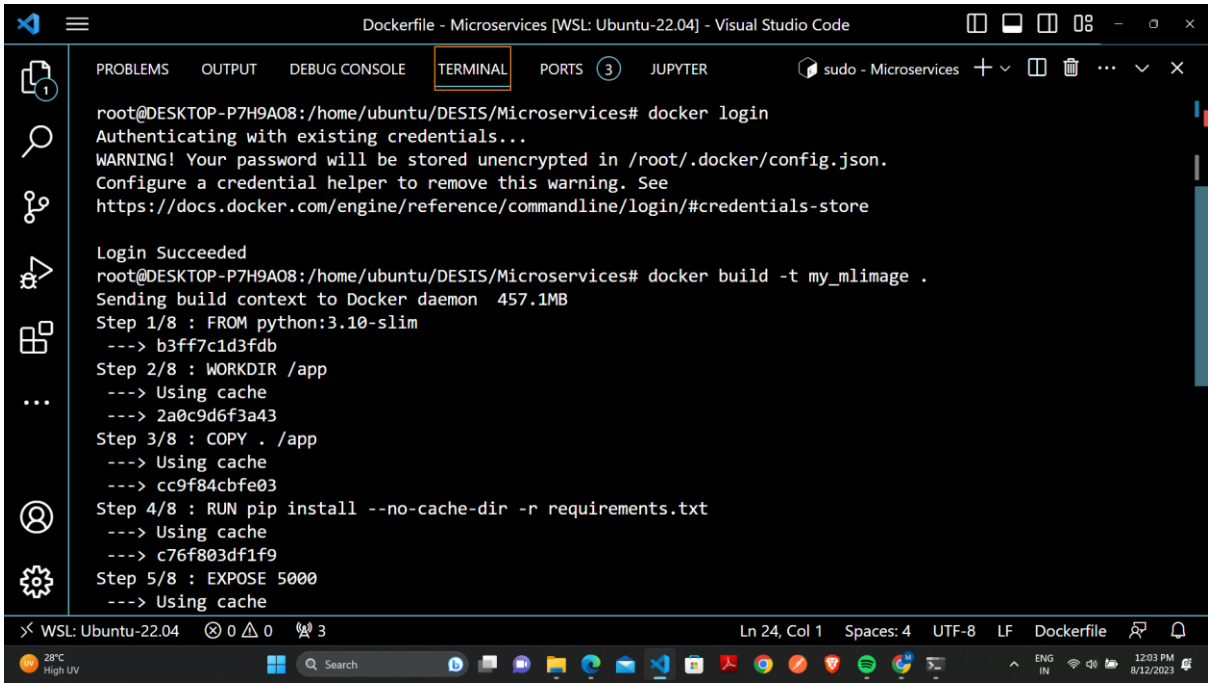



:




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

:




```
root@DESKTOP-P7H9AO8:/home/ubuntu/DESIS/Microservices# docker build -t my_mlimage .
Sending build context to Docker daemon  457.1MB
Step 1/8 : FROM python:3.10-slim
 ---> b3ff7c1d3fdb
Step 2/8 : WORKDIR /app
 ---> Using cache
 ---> 2a0c9d6f3a43
Step 3/8 : COPY . /app
 ---> Using cache
 ---> cc9f84cbfe03
Step 4/8 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> c76f803df1f9
Step 5/8 : EXPOSE 5000
 ---> Using cache
 ---> e131f3f52c57
Step 6/8 : ENV FLASK_APP=app.py
 ---> Using cache
 ---> 2526ad38f613
Step 7/8 : ENV FLASK_RUN_HOST=0.0.0.0
 ---> Using cache
 ---> fdffe2e10aab
```

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

:

```
Successfully built e21bea1b323c
Successfully tagged my_mlimage:latest
root@DESKTOP-P7H9AO8:/home/ubuntu/DESIS/Microservices# sudo lsof -i :5000
root@DESKTOP-P7H9AO8:/home/ubuntu/DESIS/Microservices# docker run -p 5001:5000 my_mlimage
 * Serving Flask app 'app.py' (lazy loading)
 * 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)
172.17.0.1 - - [12/Aug/2023 06:32:06] "GET / HTTP/1.1" 404 -
172.17.0.1 - - [12/Aug/2023 06:32:07] "GET /favicon.ico HTTP/1.1" 404 -
```