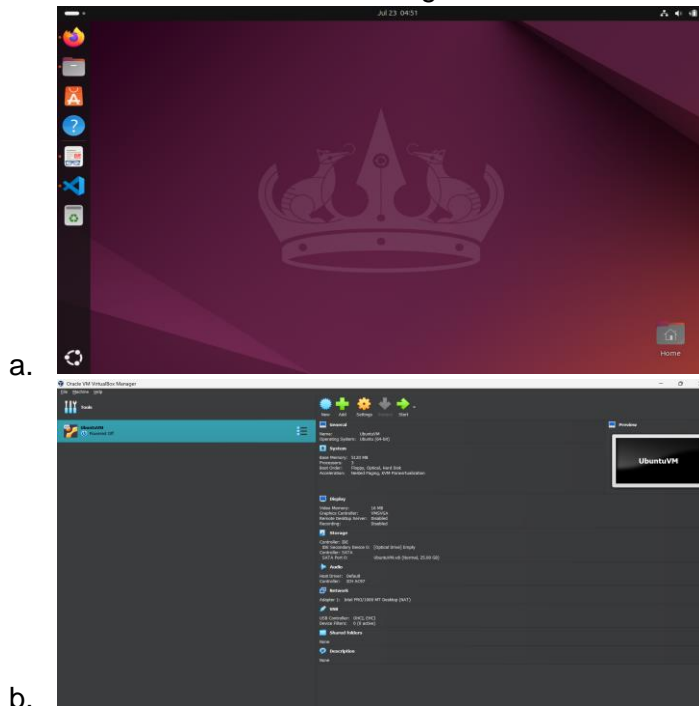
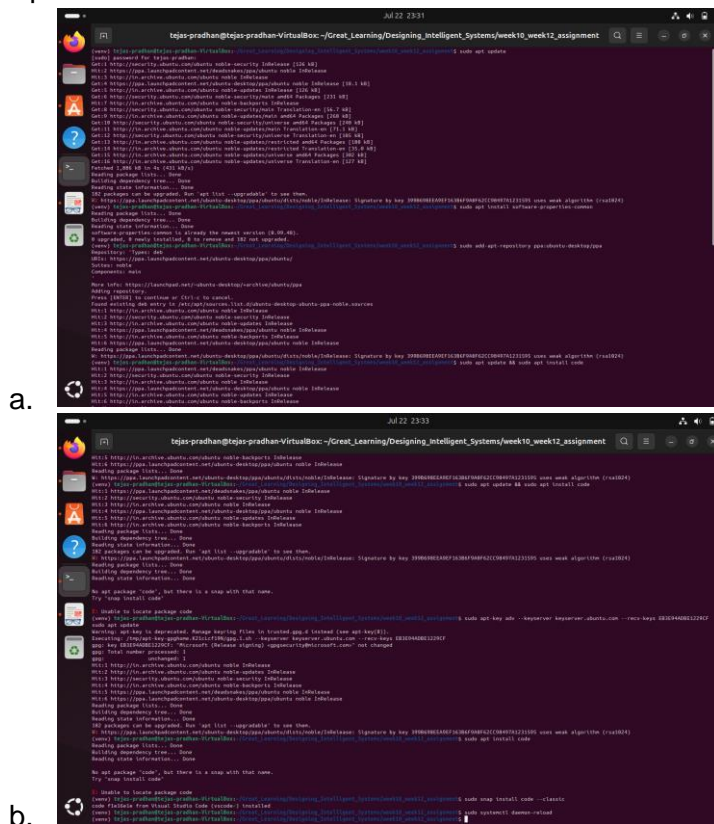
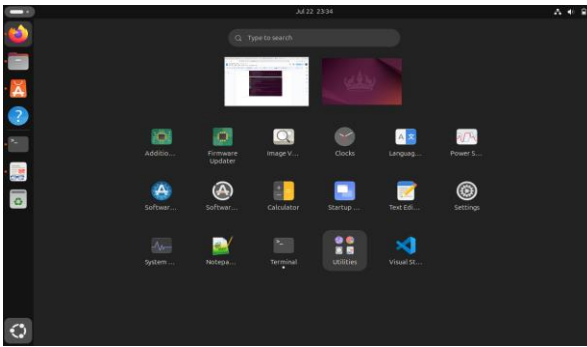


## 1. Host a Ubuntu Virtual Machine using Oracle VM Virtual Box.



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





### 3. Set up Python.

[illegible]

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

```
(venv) tejaprahandtejas-pradhan-VirtualBox:~/Great_Learning/Designing_Intelligent_Systems/week10_week11_assignments$ git clone https://github.com/Vikas090766/Microservices.git
Cloning into 'Microservices'...
remote: Enumerating objects: 95, done.
remote: Total 95 (delta 0), reused 0 (delta 0), pack-reused 95
Receiving objects: 100% (95/95), 96.20 KiB | 1.41 MiB/s, done.
Resolving deltas: 100% (28/28), done.
(venv) tejaprahandtejas-pradhan-VirtualBox:~/Great_Learning/Designing_Intelligent_Systems/week10_week11_assignments$
```

## 5. Create a Virtual Environment.

```
tejas-pradhan@tejas-pradhan-VirtualBox: /Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ python3 -m venv venv
tejas-pradhan@tejas-pradhan-VirtualBox: /Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ ls
app.py  code_model_training_data_model.py  README.md  requirements.txt  tests  venv
tejas-pradhan@tejas-pradhan-VirtualBox: /Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ source venv/bin/activate
(venv) tejas-pradhan@tejas-pradhan-VirtualBox: /Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$
```

6. Install the dependencies from requirements.txt file.

[illegible][illegible]

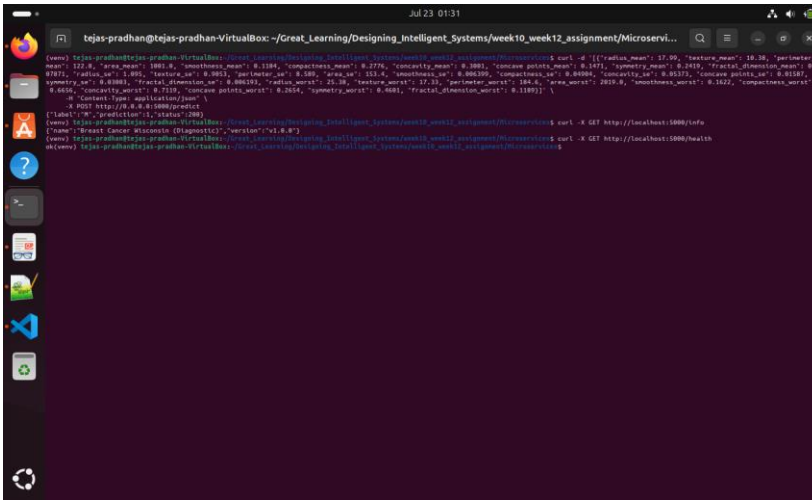


c.

```
(venv) tejas-pradhan@tejas-pradhan-VirtualBox: ~/Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ python3.10 app.py
* Serving Flask app 'ms'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.15:5000
Press CTRL+C to quit
10.0.2.15 - - [23/Jul/2024 01:23:03] "GET /health HTTP/1.1" 200 -
```

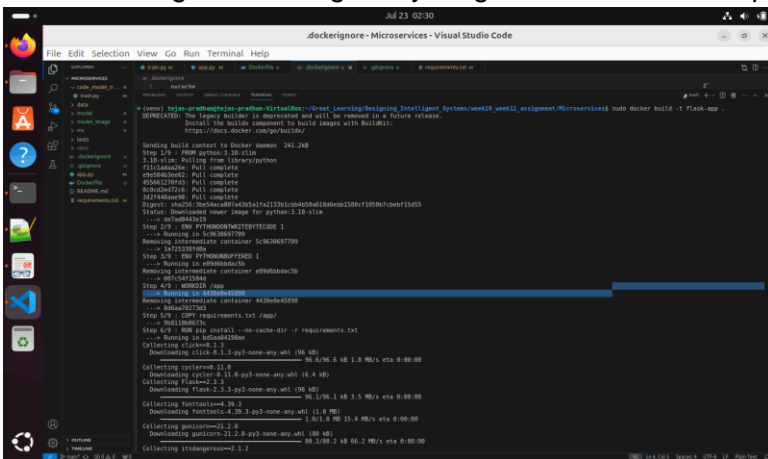
9. Test the application and make predictions using the example calls available in the folder /tests.

a.

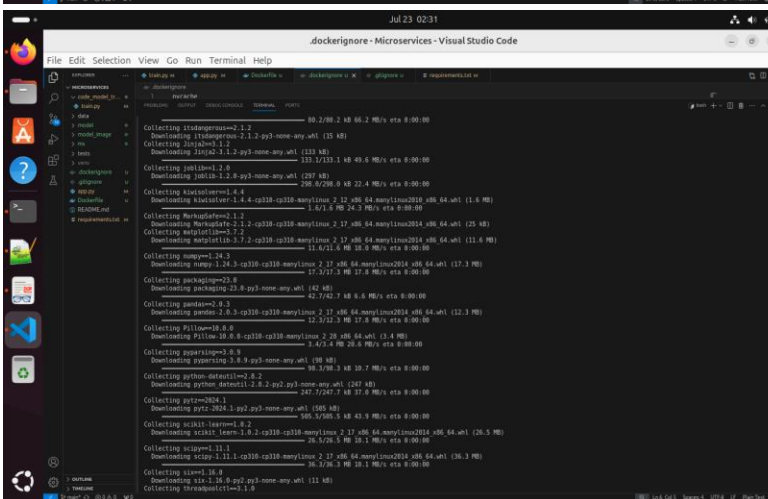


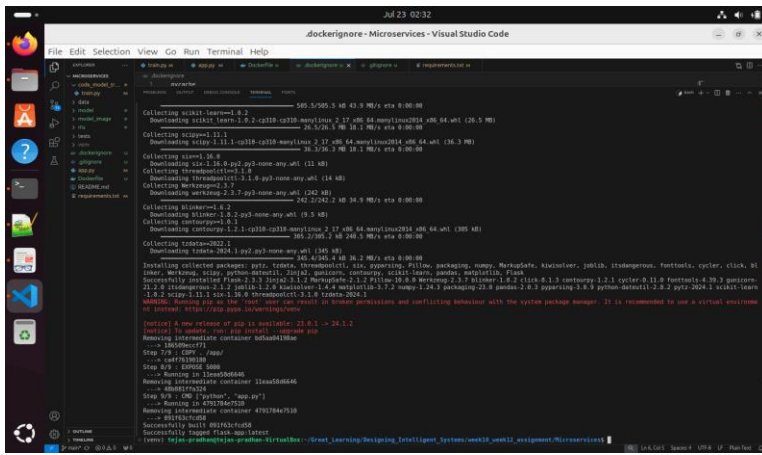
10. Create a docker image containing everything needed to run the application.

a.



b.





c.

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

```

tejas-pradhan@tejas-pradhan-VirtualBox: ~/Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ docker run -p 5000:5000 flask-app
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://172.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [22/Jul/2024 21:19:47] "GET /health HTTP/1.1" 200 -
172.17.0.1 - - [22/Jul/2024 21:19:48] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [22/Jul/2024 21:20:29] "GET /info HTTP/1.1" 200 -

```

a.

```

tejas-pradhan@tejas-pradhan-VirtualBox: ~/Great_Learning/Designing_Intelligent_Systems/week10_week12_assignment/Microservices$ curl -X POST http://172.17.0.3:5000/predict \
-H "Content-Type: application/json" \
-d '{
  "id": 842302,
  "diagnosis": "M",
  "radius_mean": 17.99,
  "texture_mean": 10.38,
  "perimeter_mean": 122.8,
  "area_mean": 1001,
  "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,
  "fractal dimension worst": 0.173,
  "label": "M", "prediction": 1, "status": 200
}'

```

b.