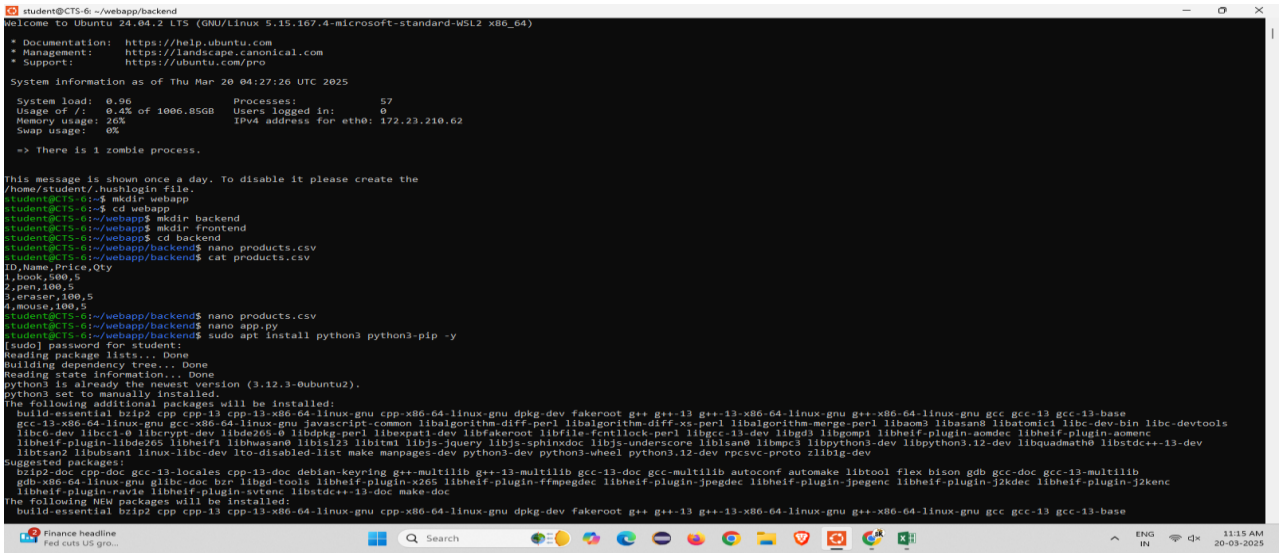
Name: B. Bharanidharan [24MCR010]

Class: I – MCA – “A”

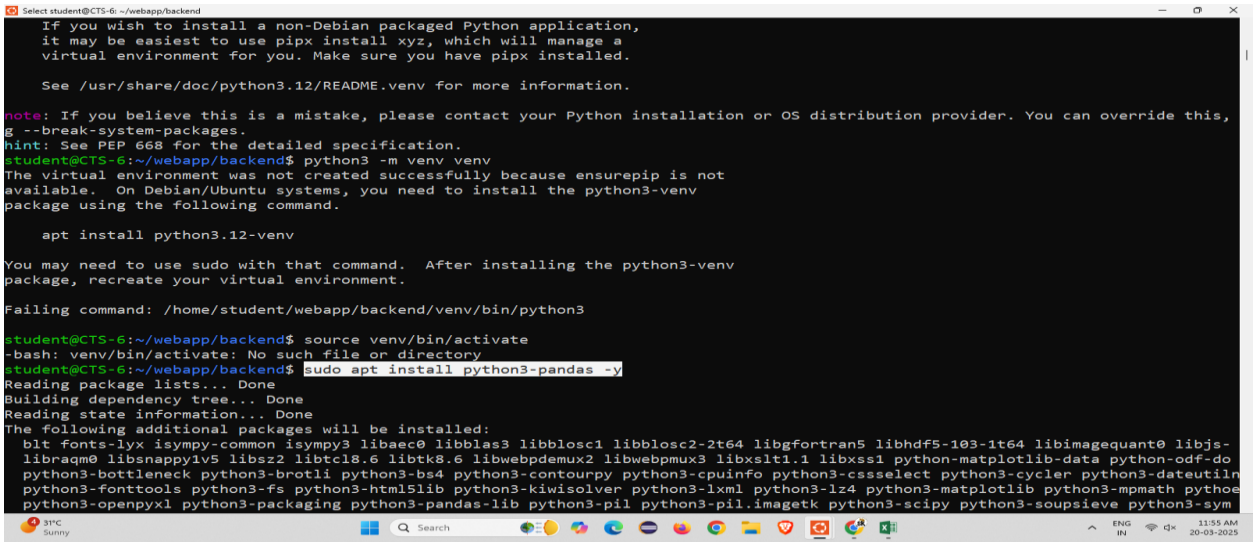
DAY 3 & 4 - DEVOPS TRAINING

Step 1:

Create the directory as webapp and make the directory frontend and backend and create a nano products.csv file and check in excel whether the format is correct

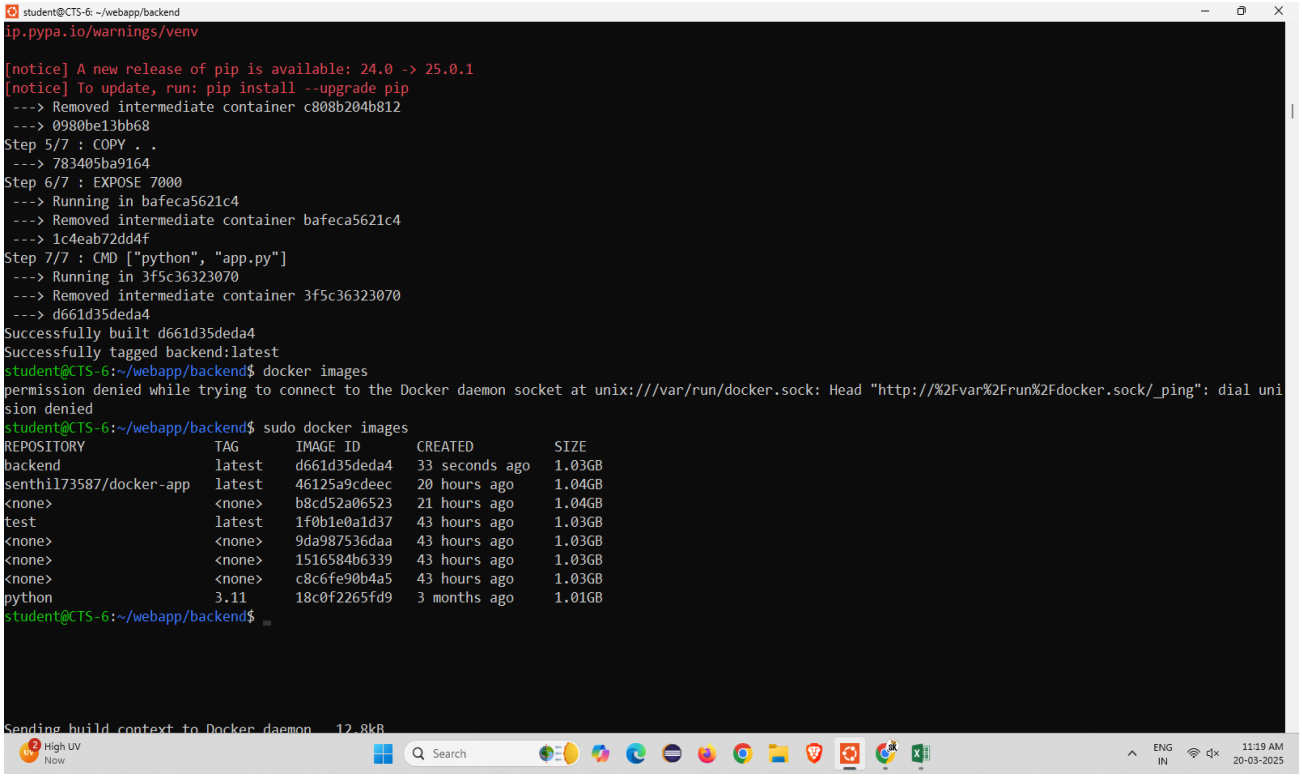


Step 2: Install the python3-pandas-y libraries into it



Step 3:

Using sudo docker images command to view list



App.py

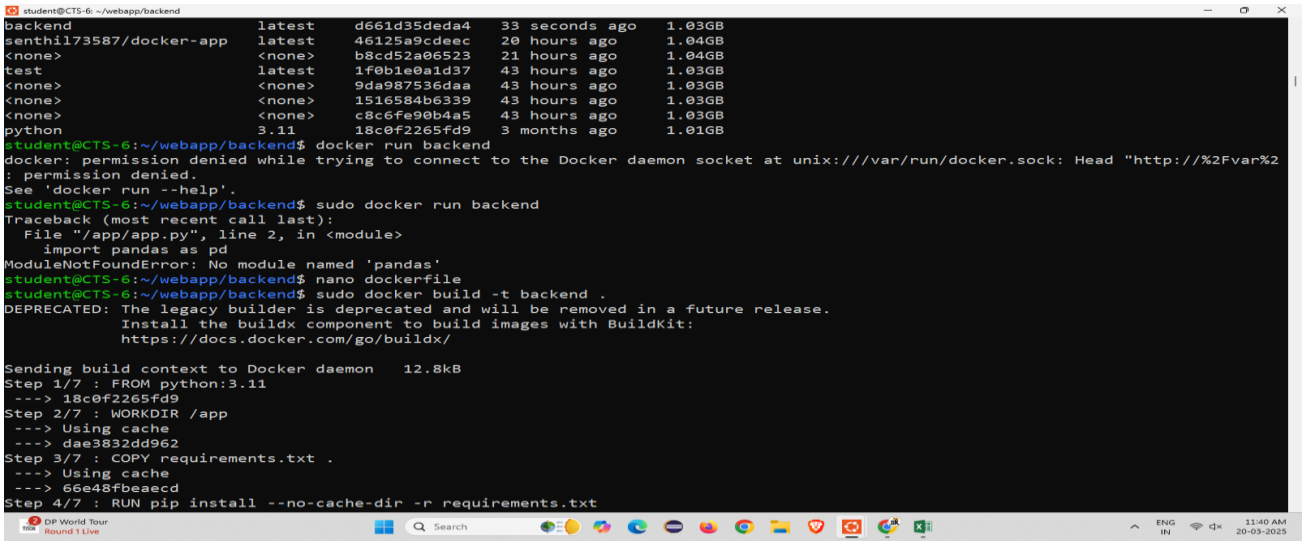
```
from flask import Flask
import pandas as pd
app = Flask(__name__)
@app.route("/products", methods=['GET']) # ✅ Fixed 'methods' issue
def read_data():
    df = pd.read_csv("products.csv")
    json_data = df.to_json()
    return json_data
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=7000)
```

Docker File

```
FROM python:3.11

WORKDIR /app

COPY requirements.txt .  # Ensure pandas is in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 7000

CMD ["python", "app.py"]
```

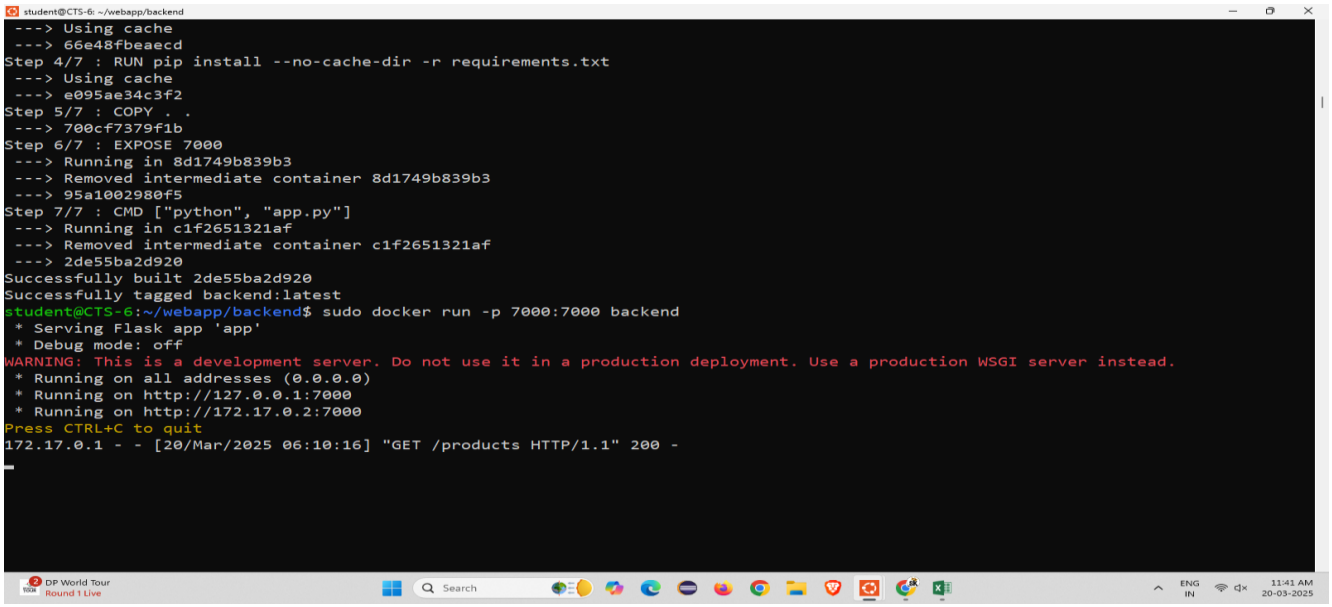
Step 4 :

nano dockerfile and nano docker built –t backend . to built the docker.



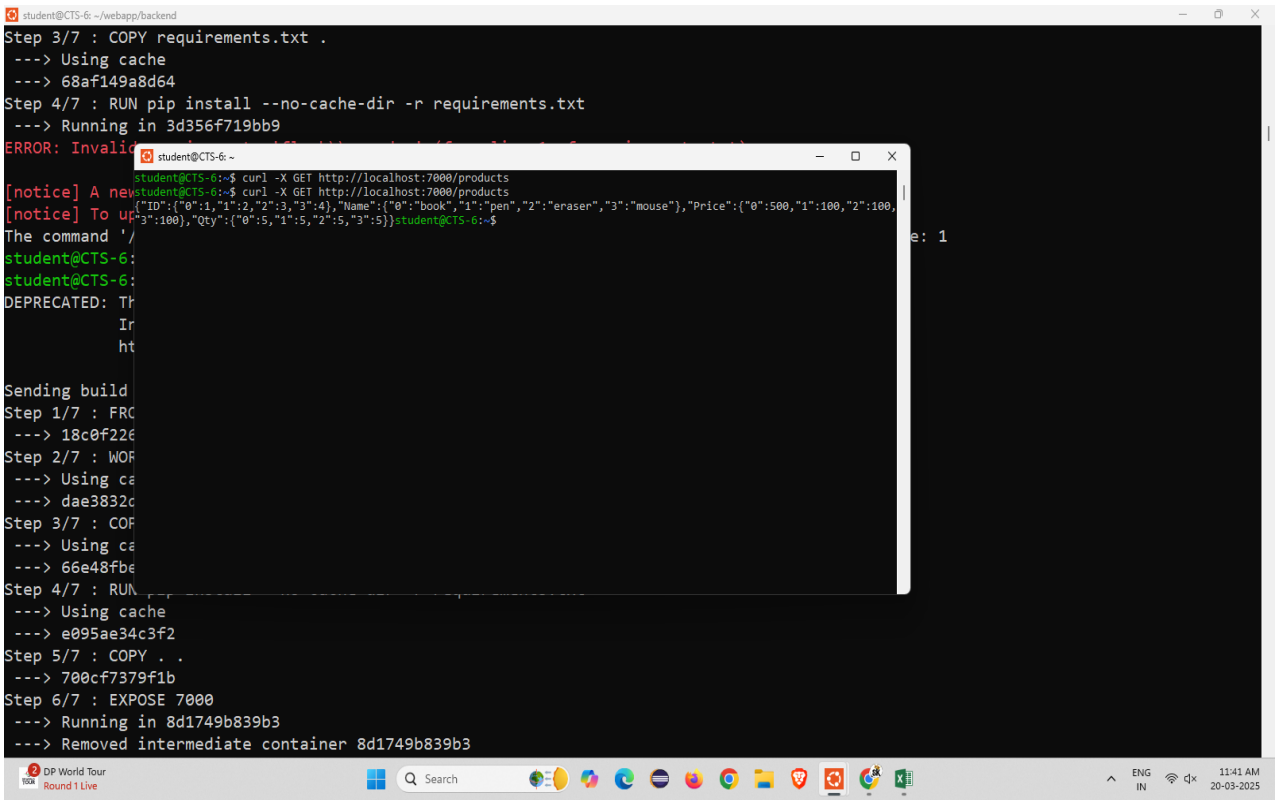
Step 5 :

using sudo docker run –p 7000:7000 backend command to run the backend

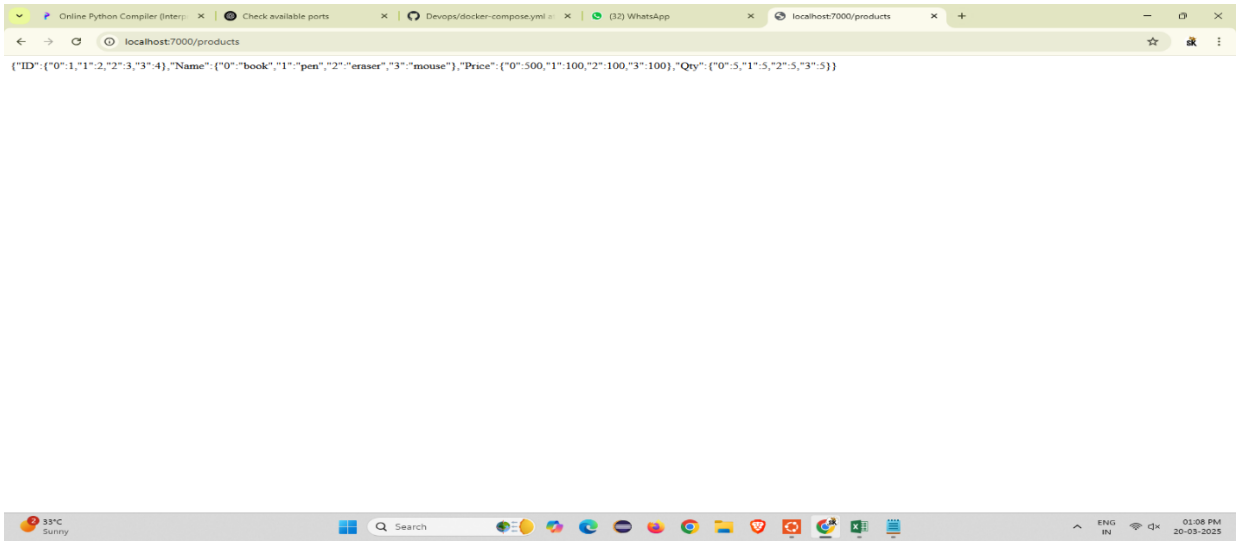


Step 6:

Open new Ubuntu and run as administration and enter the command as curl –X GET http://localhost:7000/products

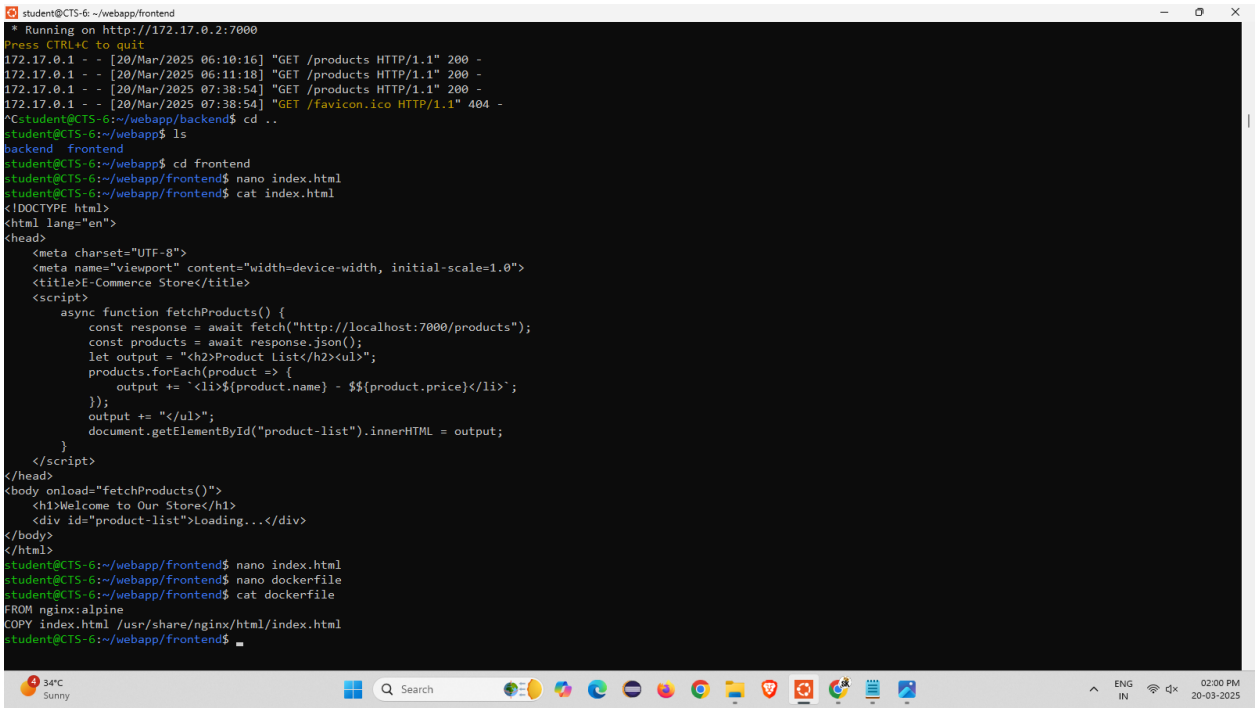


Step 7: Go to the browser and enter the url it displays the backend the backend



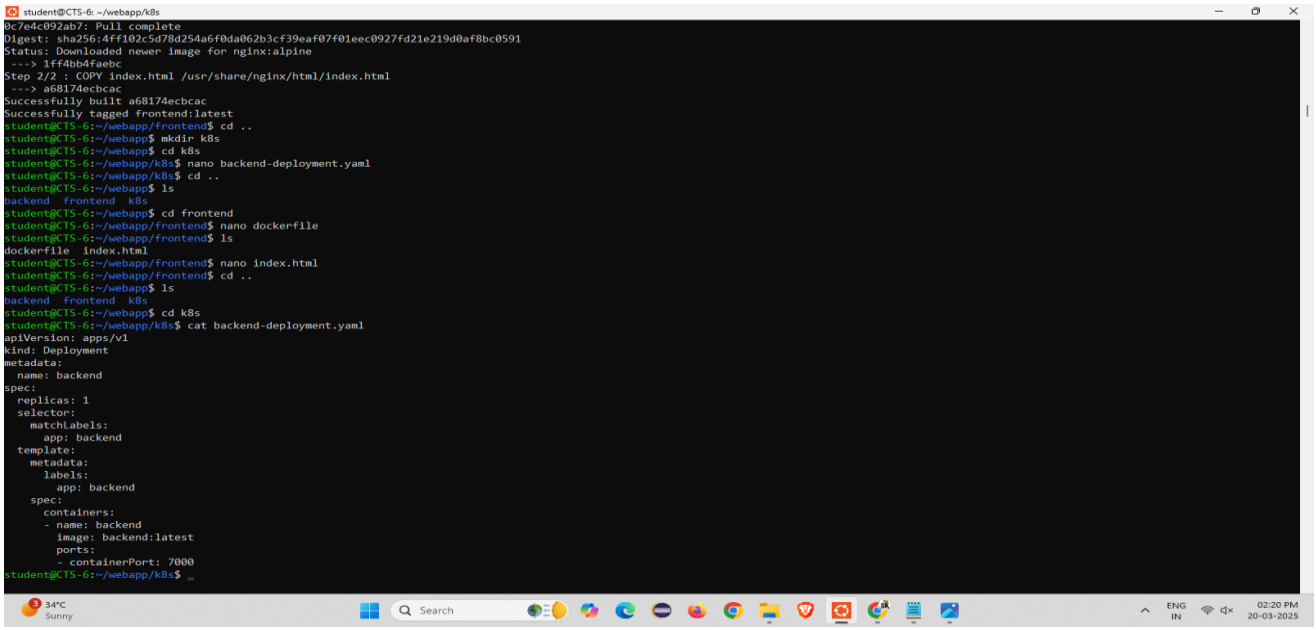
Step 8:

Change the repository to frontend and create a nano index.html file and enter the code



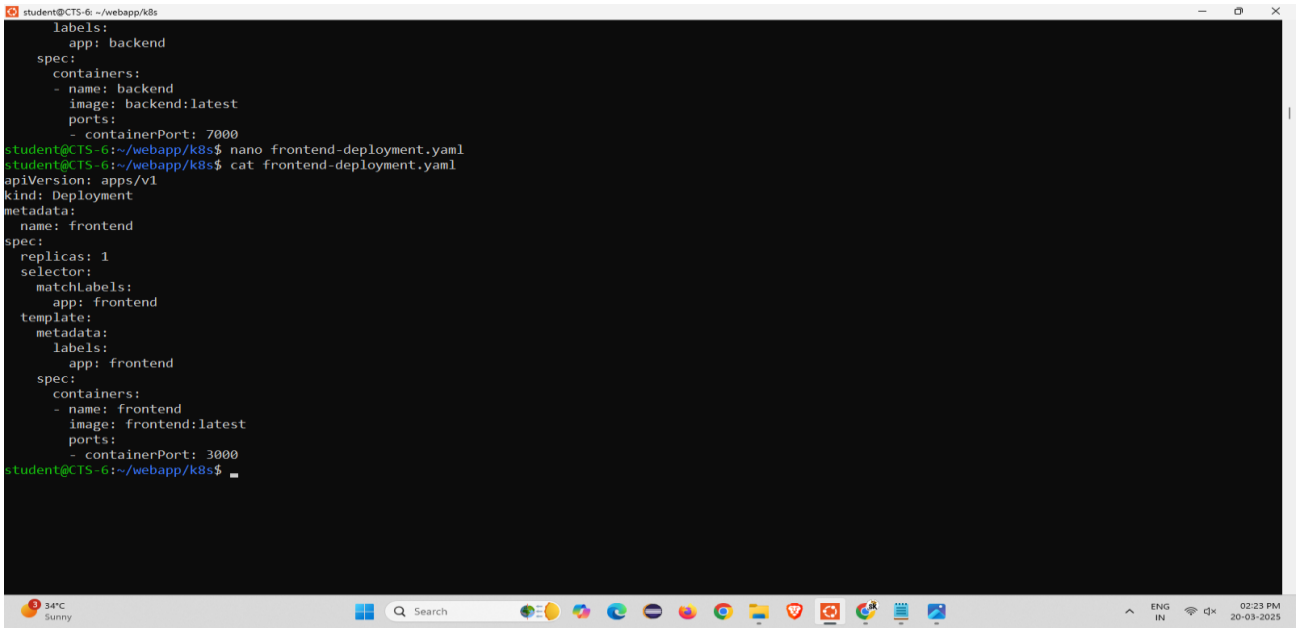
Step 9:

Change the directory and and make the directory k8s in that create a nano backend-deployment.yaml and add the code



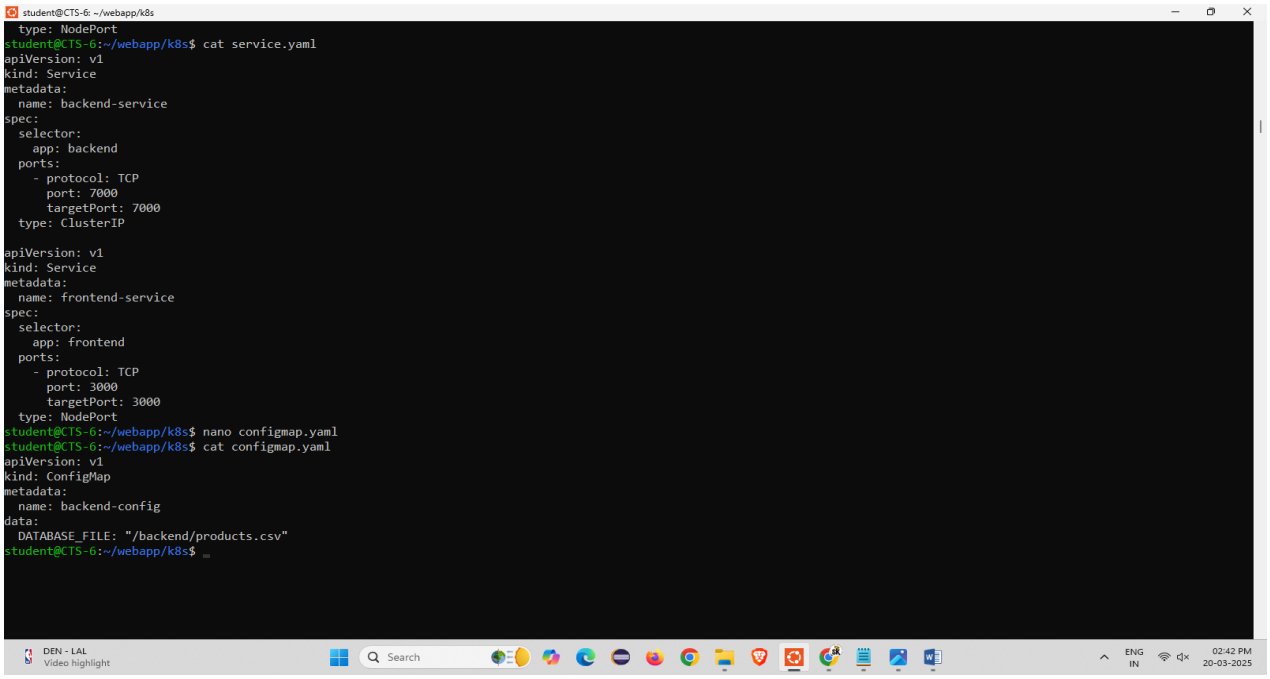
Step 10:

Create another nano frontend-deployment.yaml file and add the code



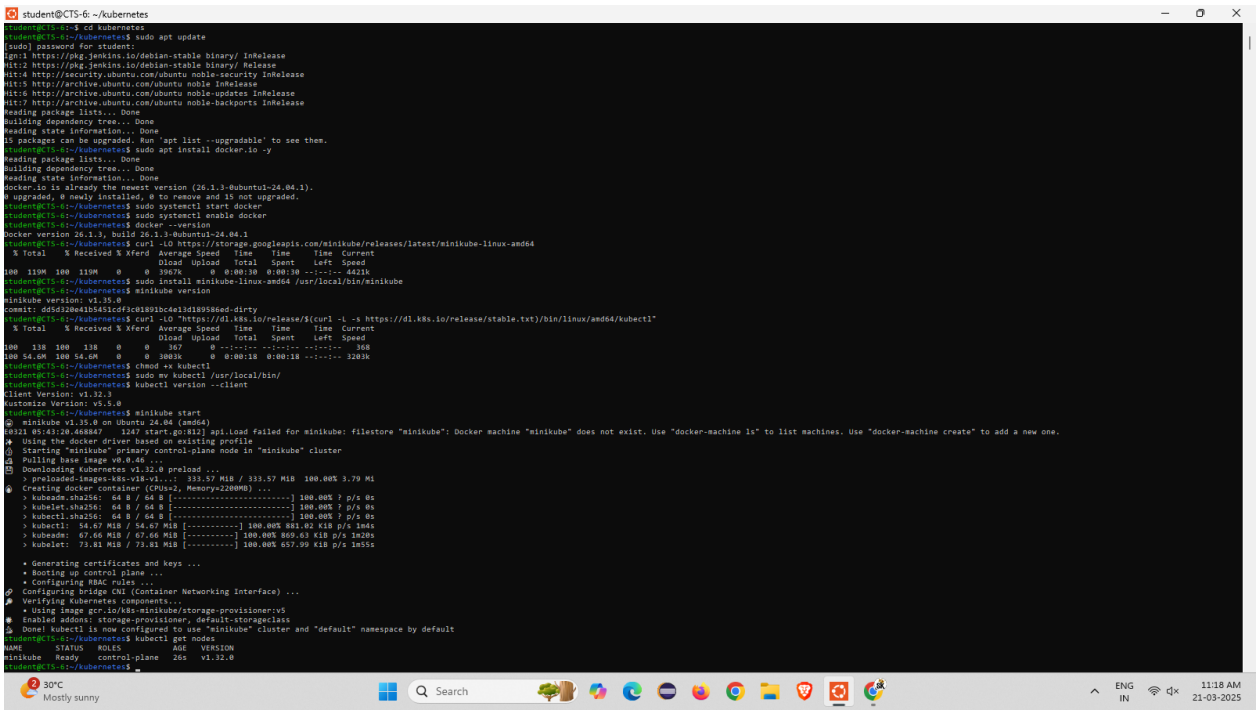
Step 11:

Create another nano service.yaml and configmap.yaml file and add the code



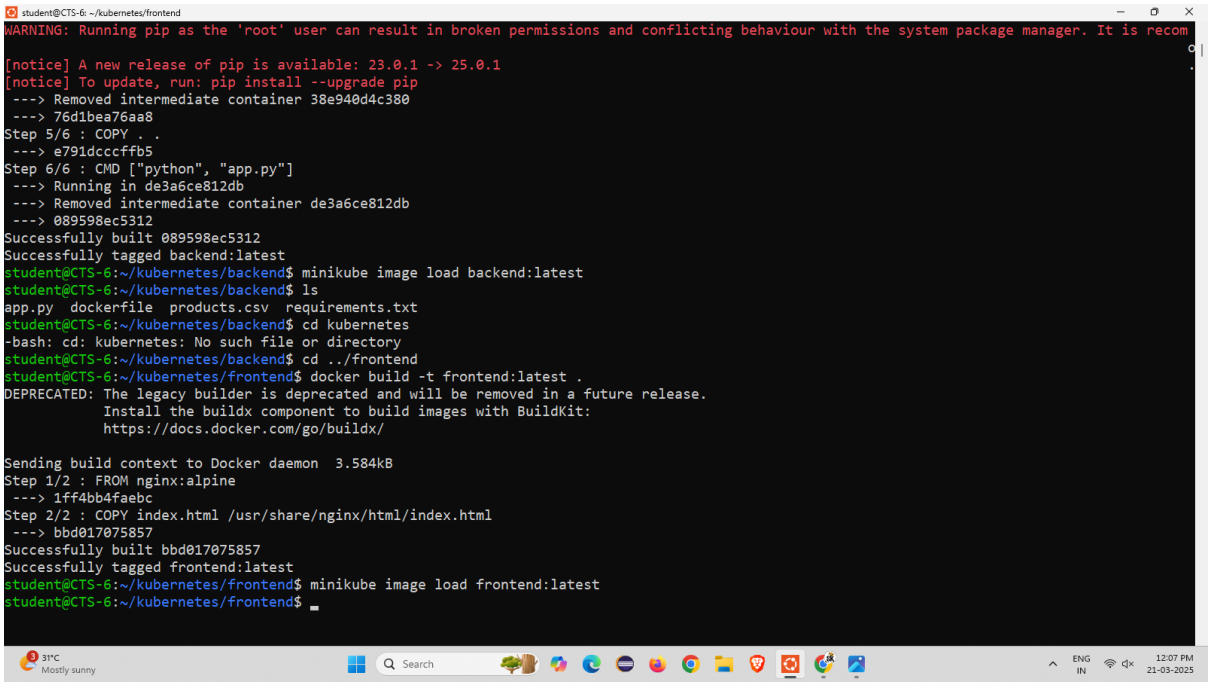
Step 12:

Clone the kubernetes github repository and run the following commands

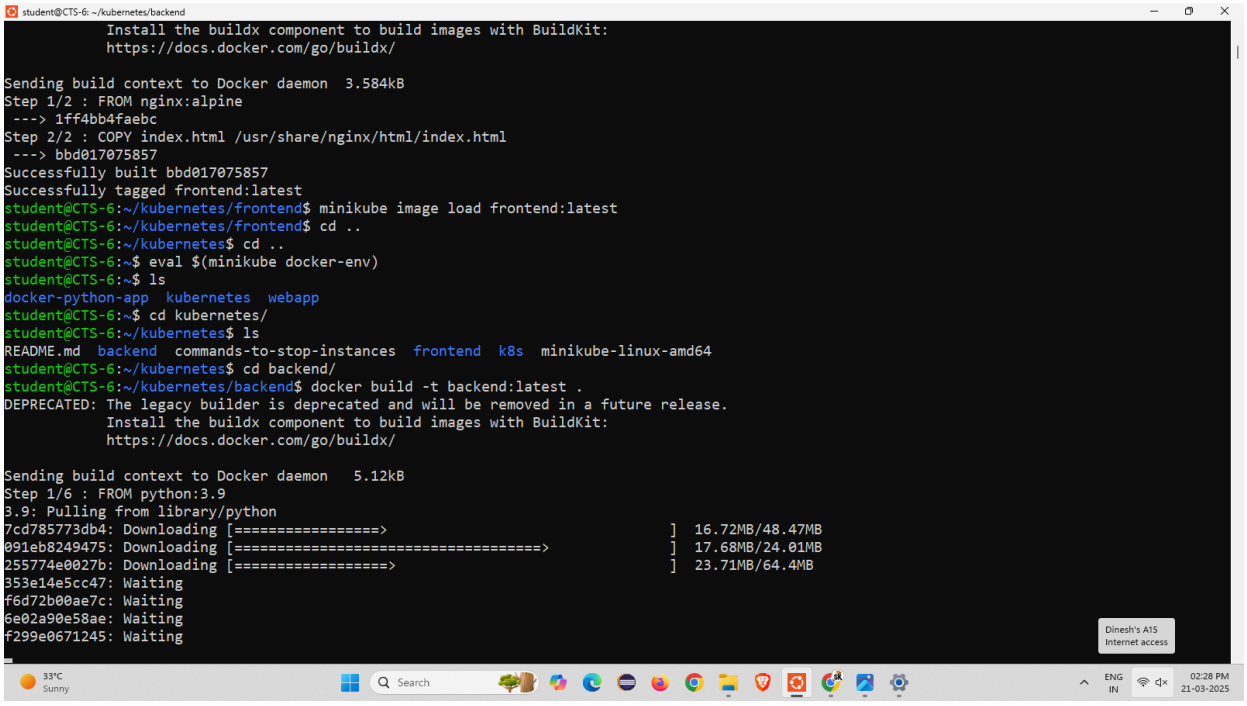


Step 13 :

Go to the kubernetes backend and frontend directory and type the command to load an image

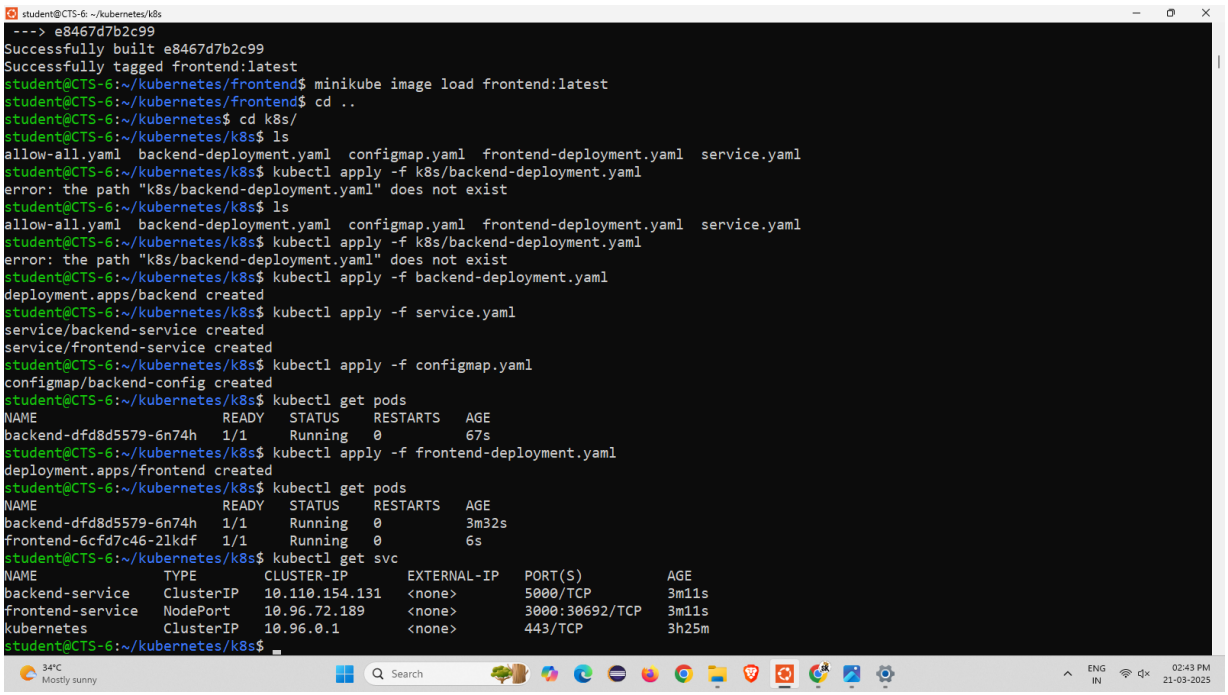


Step 14 : Come back to root directory setup minikube docker-env and build the frontend and backend



Step 15 :

Open k8s directory and list the files into it and kubectl apply –f commands and initialize get pods , get svc



Step 16 :

using minikube service frontend-service –url it will displays the https:// ip address and using curl command to run in the terminal.

```
student@CTS-6:~/kubernetes/k8s$ kubectl apply -f configmap.yaml
configmap/backend-config created
student@CTS-6:~/kubernetes/k8s$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-6n74h    1/1     Running   0          67s
student@CTS-6:~/kubernetes/k8s$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
student@CTS-6:~/kubernetes/k8s$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-6n74h    1/1     Running   0          3m32s
frontend-6cfd7c46-2lkdf    1/1     Running   0          6s
student@CTS-6:~/kubernetes/k8s$ kubectl get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
backend-service    ClusterIP   10.110.154.131   <none>        5000/TCP         3m11s
frontend-service   NodePort    10.96.72.189     <none>        3000:30692/TCP   3m11s
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP          3h25m
student@CTS-6:~/kubernetes/k8s$ minikube service frontend-service --url
http://127.0.0.1:39285
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
student@CTS-6:~/kubernetes/k8s$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{"id":2,"name":"Laptop","price":799.99},{"id":3,"name":"Headphones","price":49.99},{"id":4,"name":
"Tablet","price":199.99}]
/ #
```

Step 17:

Enter the given ip address in the browser and get the output.