

## SPRINT -4 DOCUMENTATION

**Project title:** NUTRITION ASSISTANT APPLICATION.

**Team ID:** PNT2022TMID15840

**Team members:**

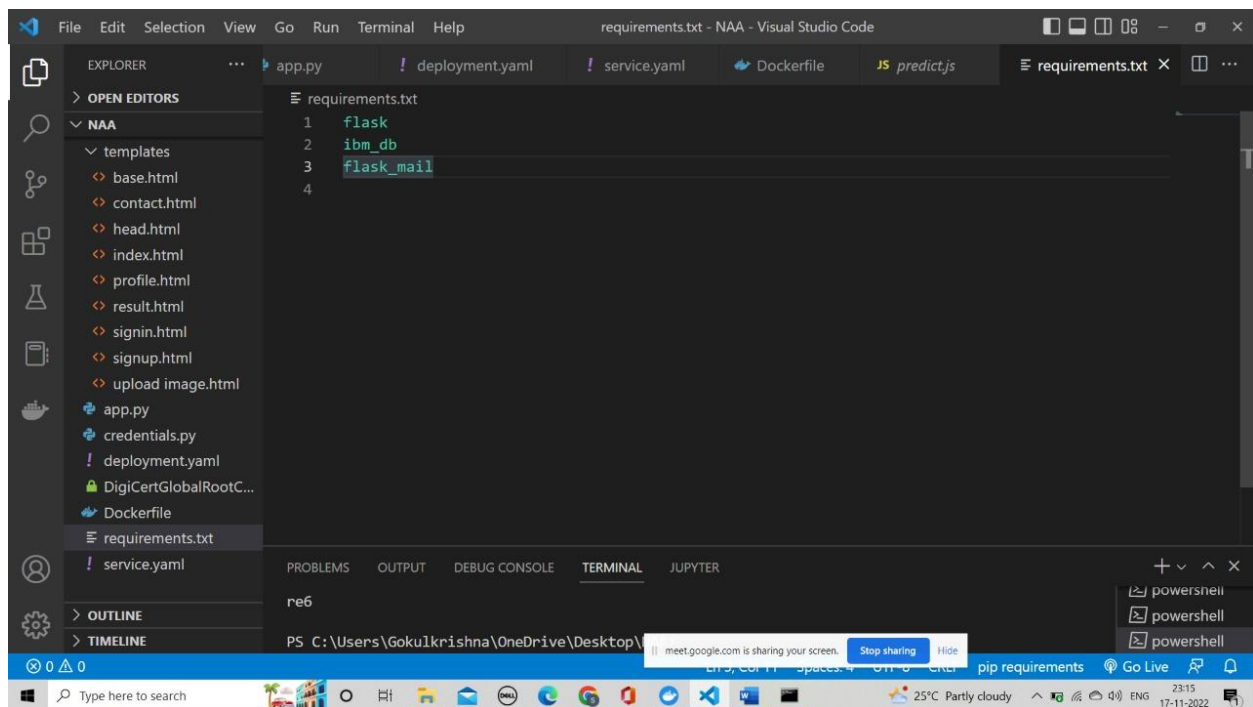
Shivaani.B

Sri Varshini.R. A

Sowjanya.G

Durga.P

Requirements.txt:

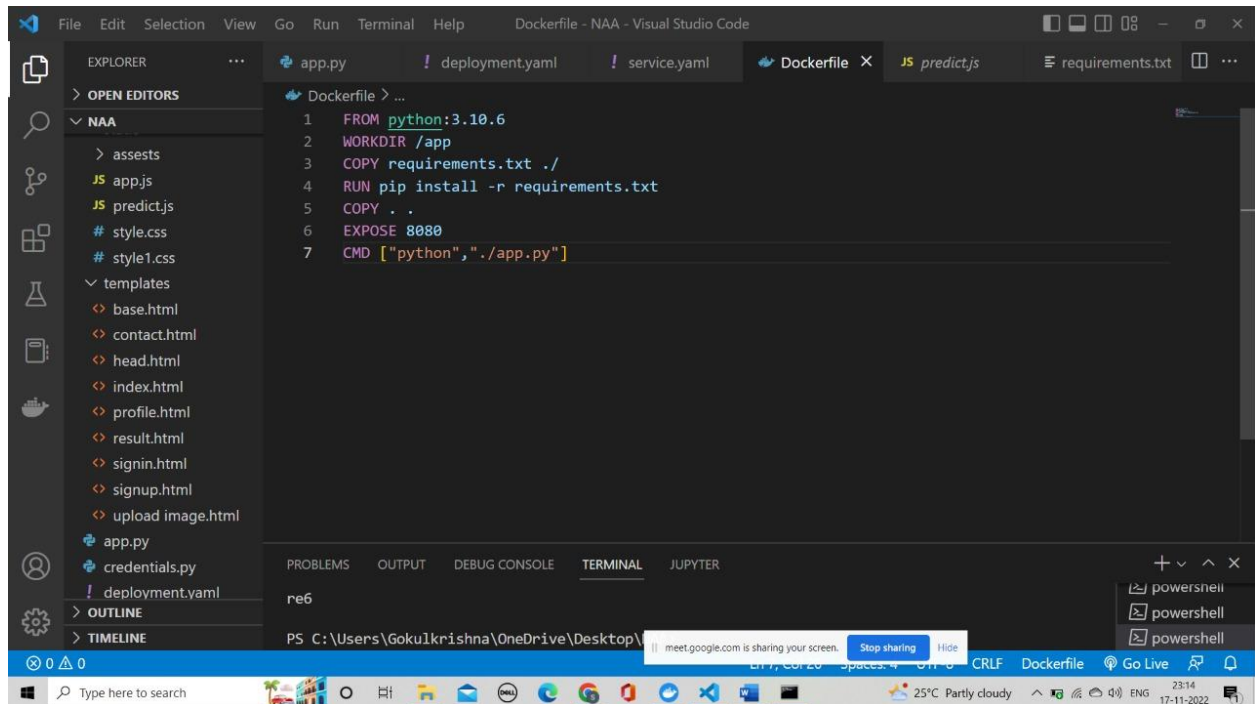


The screenshot shows the Visual Studio Code interface with the 'requirements.txt' file open in the editor. The file contains the following text:

```
requirements.txt
1 flask
2 ibm_db
3 flask_mail
4
```

The Explorer sidebar on the left shows the project structure for 'NAA', including templates, HTML files, Python files, and configuration files. The Terminal at the bottom shows the command 'pip requirements' being executed.

## Dockerfile:

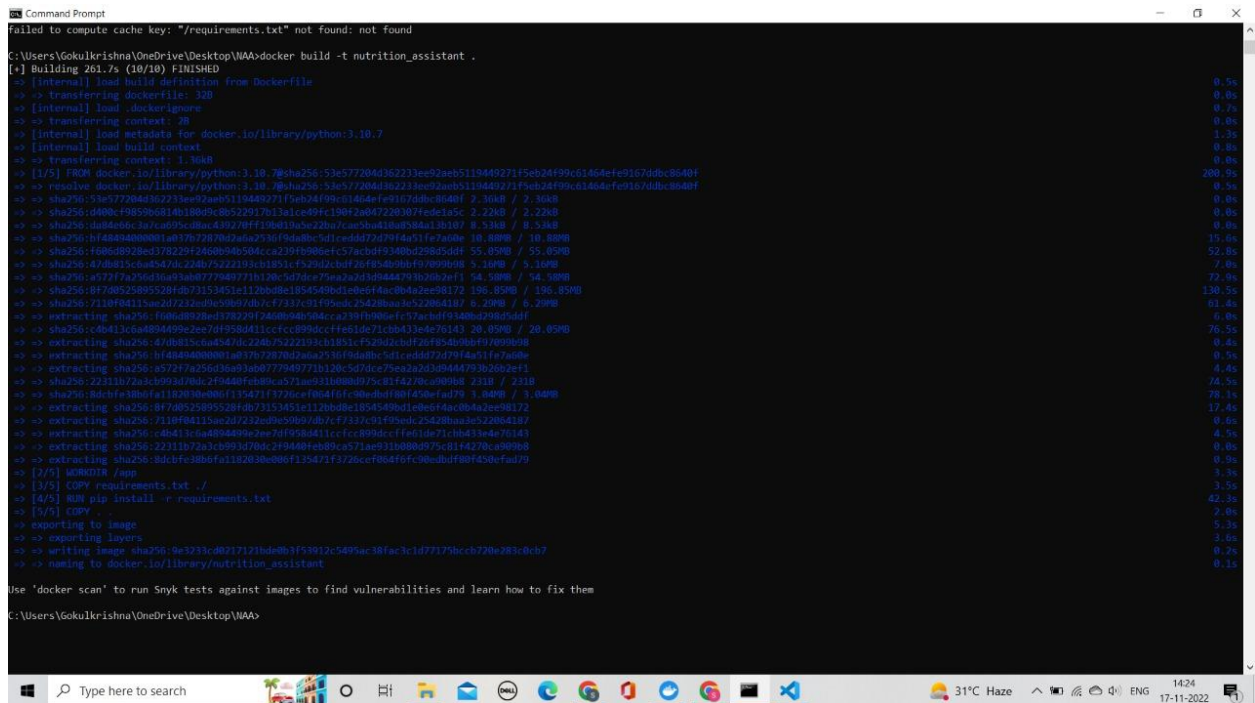


The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

```
1 FROM python:3.10.6
2 WORKDIR /app
3 COPY requirements.txt ./
4 RUN pip install -r requirements.txt
5 COPY . .
6 EXPOSE 8080
7 CMD ["python", "./app.py"]
```

The Explorer sidebar on the left shows a project structure with folders like 'NAA', 'assess', 'templates', and files like 'app.py', 'credentials.py', 'deployment.yaml', 'style.css', and 'style1.css'. The bottom status bar indicates the file is 'Dockerfile' and the workspace is 're6'.

## Building Docker Image:



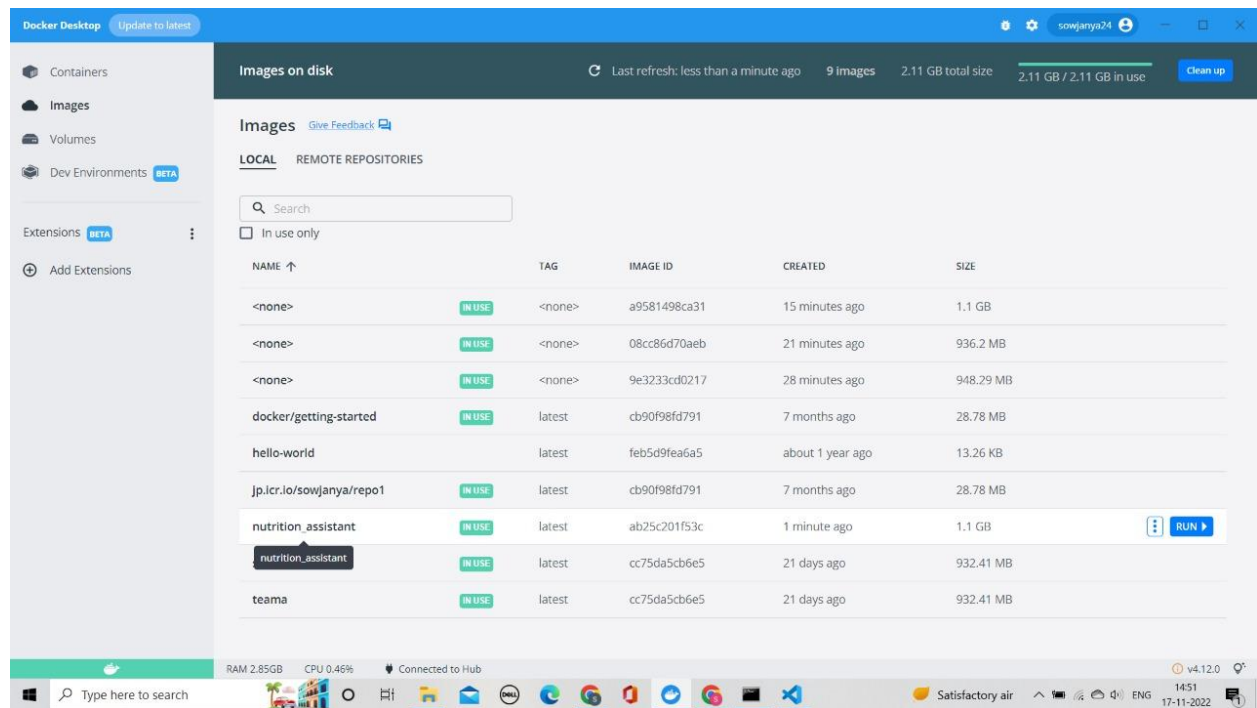
The screenshot shows a Windows Command Prompt window with the following output:

```
C:\Users\Gokulkrishna\OneDrive\Desktop\NAA>docker build -t nutrition_assistant .
[+] Building 261.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.10.6
=> [internal] load build context
=> [internal] load metadata for docker.io/library/python:3.10.6
=> [1/5] FROM docker.io/library/python:3.10.6
=> resolve docker.io/library/python:3.10.6
=> sha256:53e572204d362233ee92aeb5119449271f5eb24f90c61464ef9167ddbc8640f
=> sha256:d80bcf9859068140180d9c8522917613a1ce69c190426047220307fede15c 2.22kB / 2.22kB
=> sha256:d80bcf9859068140180d9c8522917613a1ce69c190426047220307fede15c 2.22kB / 2.22kB
=> sha256:f4a8a4a80001a017672820d9a0a2530f0a0bc5d1cedd72d79f4a51fe7a0e 10.89kB / 10.89kB
=> sha256:f4a8a4a80001a017672820d9a0a2530f0a0bc5d1cedd72d79f4a51fe7a0e 10.89kB / 10.89kB
=> sha256:f4a8a4a80001a017672820d9a0a2530f0a0bc5d1cedd72d79f4a51fe7a0e 10.89kB / 10.89kB
=> sha256:47db815c6a4547dc224b75222193cb1851cf520d2cbf26f854b0b6f97090908 5.16kB / 5.16kB
=> sha256:47db815c6a4547dc224b75222193cb1851cf520d2cbf26f854b0b6f97090908 5.16kB / 5.16kB
=> sha256:a77f7a256d3a93a0b777989771b130c5d78ce75ea2a3d3d0444793b20b2ef1 54.50kB / 54.50kB
=> sha256:0f70825095528f073153451e1120048e1854540b0e0e4ac0b4a2ee98172 196.05kB / 196.05kB
=> sha256:2231b72a3c8093d70d2f9440fcb0ca571ae9311d088d975c81f427b2a09908 231B / 231B
=> sha256:2231b72a3c8093d70d2f9440fcb0ca571ae9311d088d975c81f427b2a09908 231B / 231B
=> sha256:8dc8f3806fa1192030b00f135473720ca5f06a4f0f0e4d0f450efad79 3.04kB / 3.04kB
=> sha256:8dc8f3806fa1192030b00f135473720ca5f06a4f0f0e4d0f450efad79 3.04kB / 3.04kB
=> sha256:0f70825095528f073153451e1120048e1854540b0e0e4ac0b4a2ee98172 196.05kB / 196.05kB
=> sha256:0f70825095528f073153451e1120048e1854540b0e0e4ac0b4a2ee98172 196.05kB / 196.05kB
=> sha256:2231b72a3c8093d70d2f9440fcb0ca571ae9311d088d975c81f427b2a09908 231B / 231B
=> sha256:2231b72a3c8093d70d2f9440fcb0ca571ae9311d088d975c81f427b2a09908 231B / 231B
=> sha256:8dc8f3806fa1192030b00f135473720ca5f06a4f0f0e4d0f450efad79 3.04kB / 3.04kB
=> sha256:8dc8f3806fa1192030b00f135473720ca5f06a4f0f0e4d0f450efad79 3.04kB / 3.04kB
=> [2/5] WORKDIR /app
=> [3/5] COPY requirements.txt ./
=> [4/5] RUN pip install -r requirements.txt
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> writing image sha256:9e3233cd0217121bde0b3f53012c5495ac38fac3d477375ccc720e383c0b7
=> naming to docker.io/library/nutrition_assistant

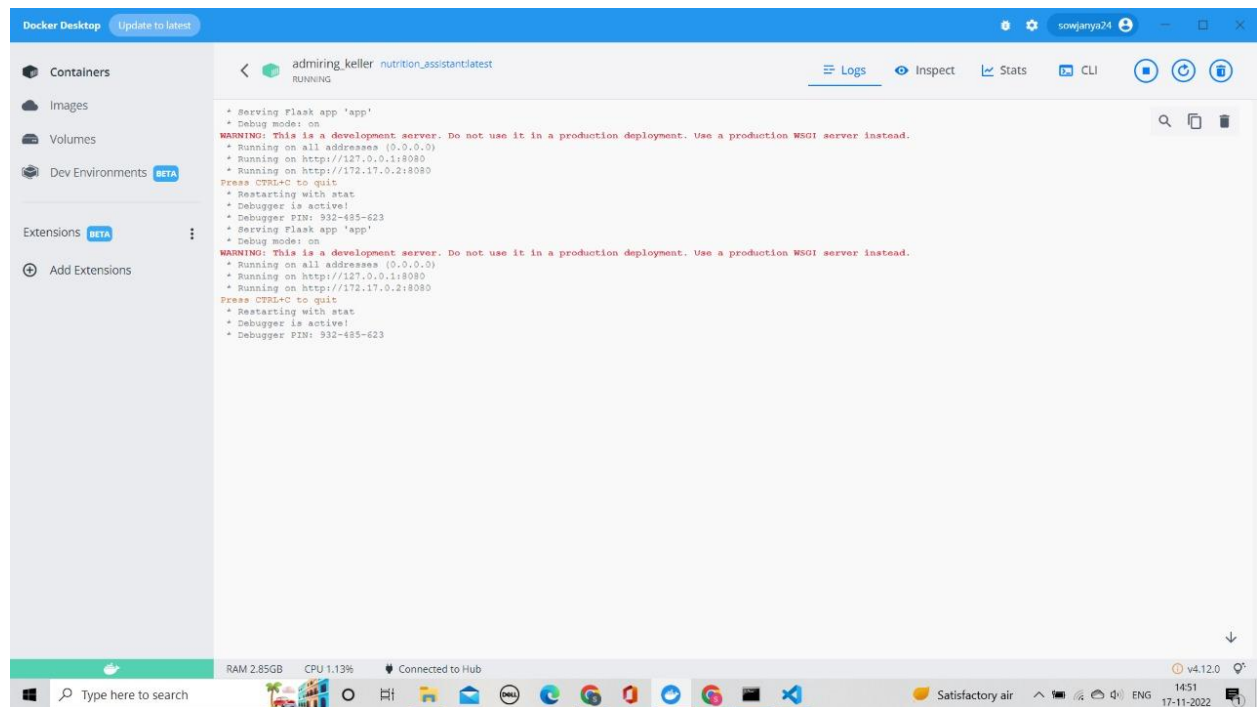
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Users\Gokulkrishna\OneDrive\Desktop\NAA>
```

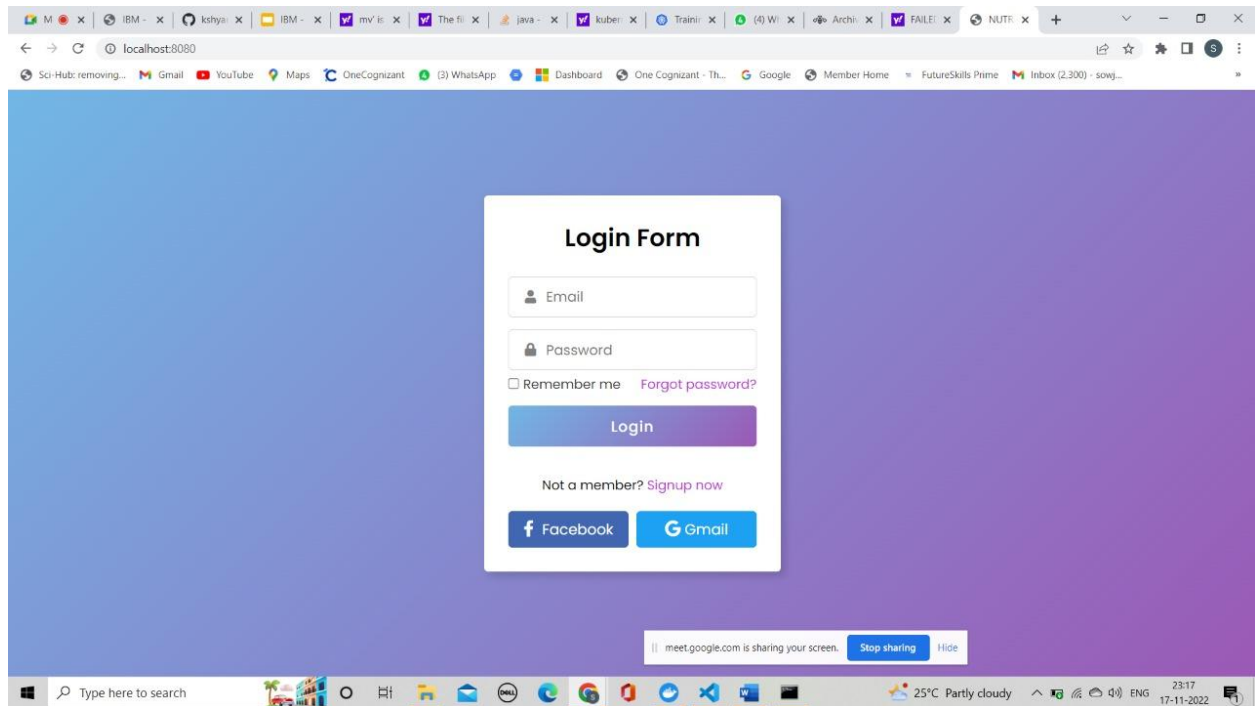
# Docker Desktop Images:



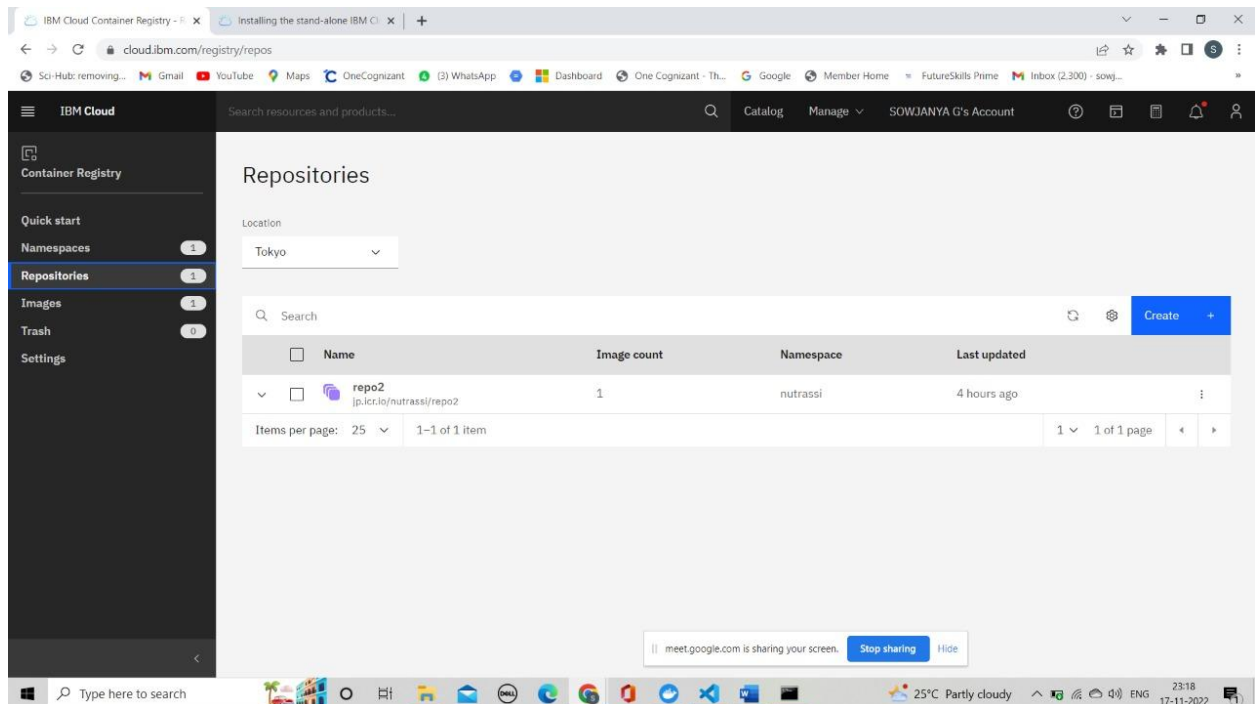
# Logs:



## Flask app deployed in Docker:



## Cloud Registry - Repository:



## Cloud Registry - Namespaces:

The screenshot shows the IBM Cloud Container Registry interface. The left sidebar contains a navigation menu with 'Namespaces' selected. The main content area is titled 'Namespaces' and shows a table with one namespace, 'nutrassi'. The table columns are Name, Resource group, Repository count, Image count, and Retention policy. The 'nutrassi' namespace is in the 'Default' resource group, has 1 repository and 1 image, and its retention policy is 'Retain all images'.

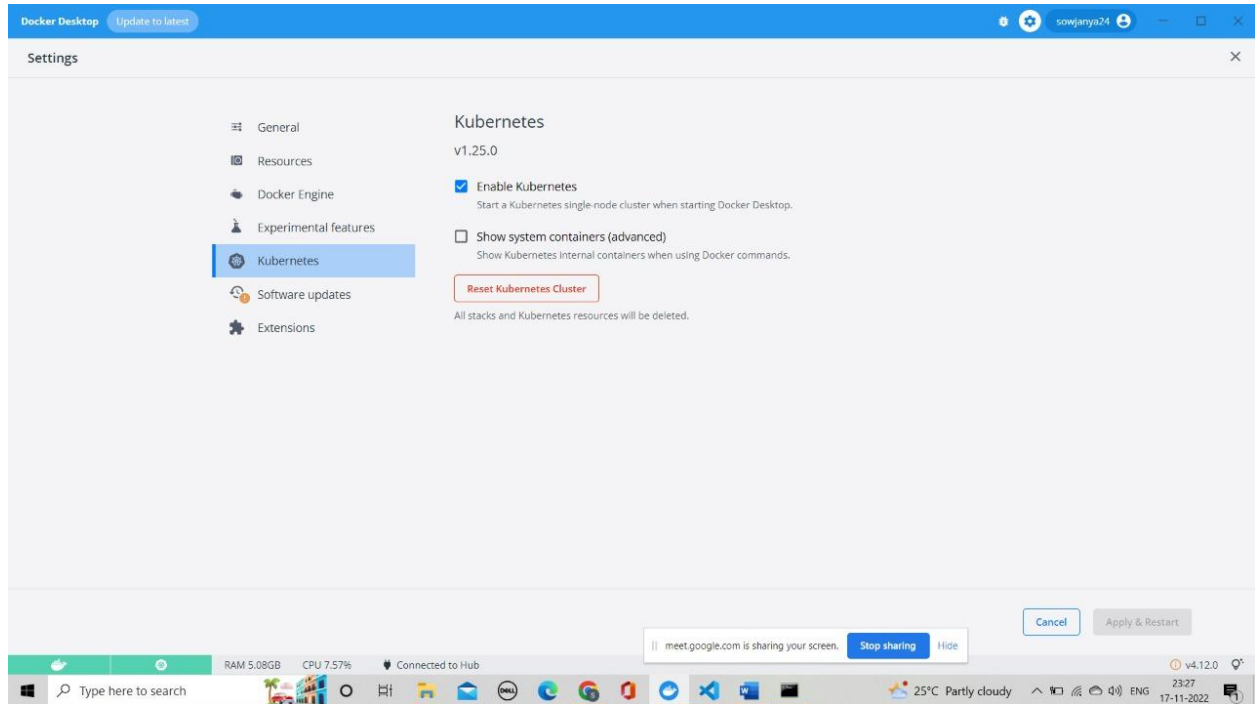
Name	Resource group	Repository count	Image count	Retention policy
nutrassi	Default	1	1	Retain all images

## Cloud Registry - Images:

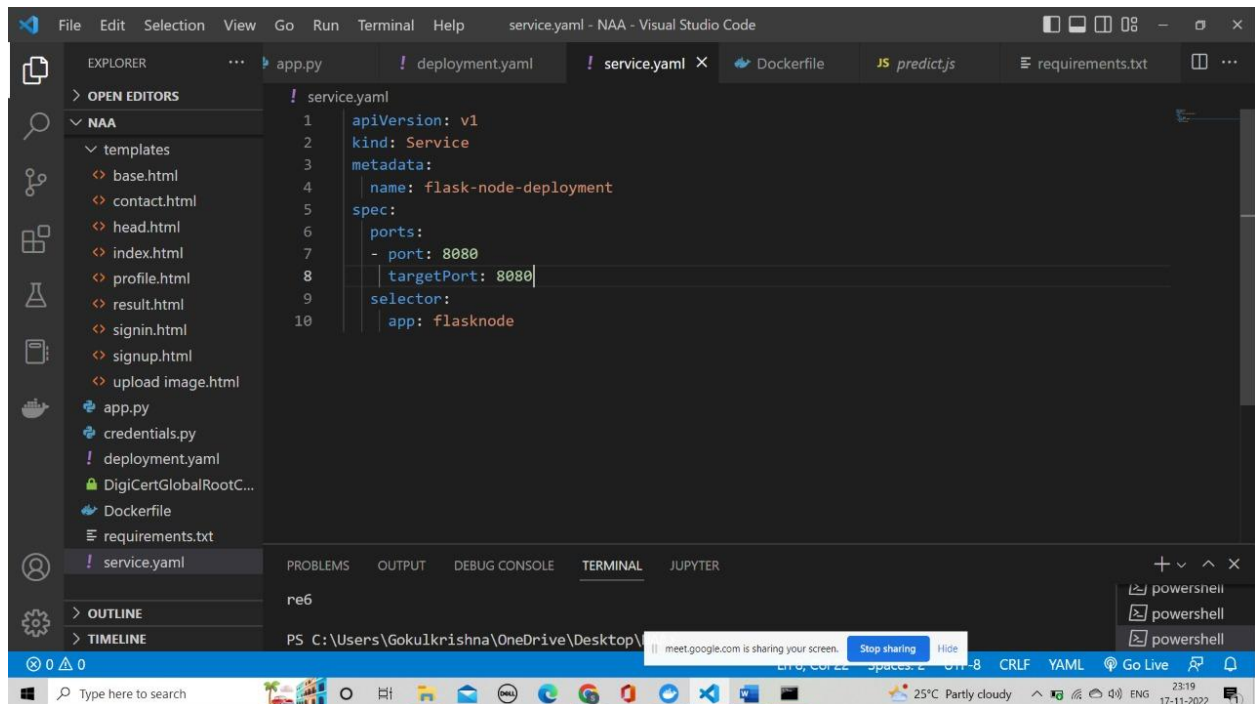
The screenshot shows the IBM Cloud Container Registry interface with the 'Images' tab selected. The main content area is titled 'Images' and shows a table with one image, 'nutrassi/repo2@sha256:9e969c549f76...'. The table columns are Repository@digest, Tags, Manifest type, Created, Size, and Security status. The image is in the 'Default' resource group, has 1 tag, and its manifest type is 'Docker'. The 'Created' column shows '4 hours ago', the 'Size' is '446 MB', and the 'Security status' is '26 issues'.

Repository@digest	Tags	Manifest type	Created	Size	Security status
nutrassi/repo2@sha256:9e969c549f76...	latest	Docker	4 hours ago	446 MB	26 issues

## Enabling Kubernetes in Docker Desktop:

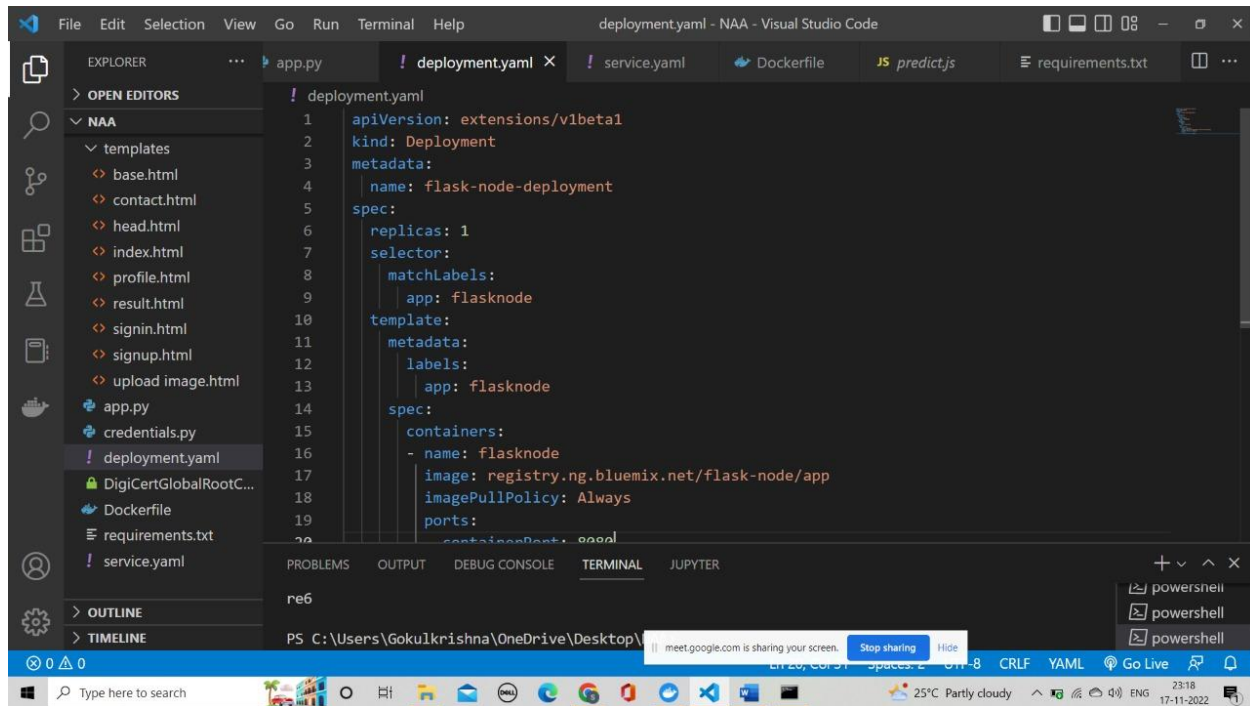


## service.yaml:





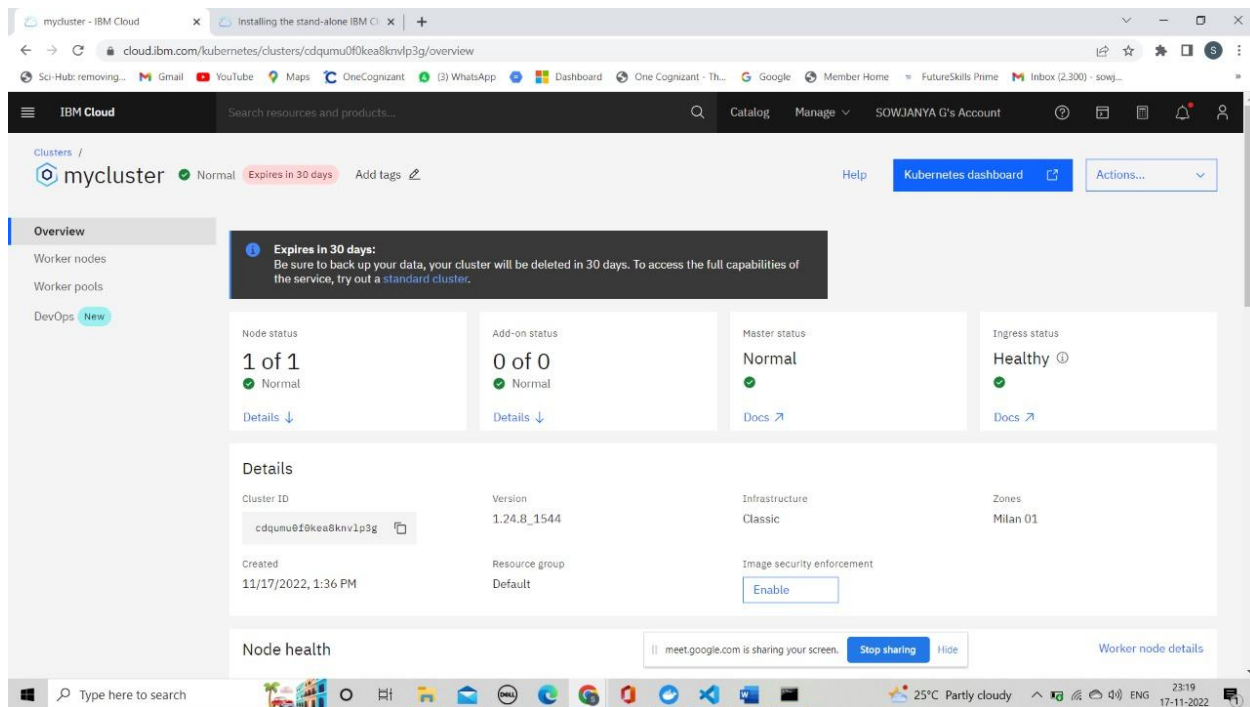
## deployment.yaml:



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'NAA' with a 'templates' folder containing several HTML files and a 'deployment.yaml' file. The code editor displays the content of 'deployment.yaml'.

```
! deployment.yaml
1  apiVersion: extensions/v1beta1
2  kind: Deployment
3  metadata:
4    name: flask-node-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: flasknode
10   template:
11     metadata:
12       labels:
13         app: flasknode
14     spec:
15       containers:
16       - name: flasknode
17         image: registry.ng.bluemix.net/flask-node/app
18         imagePullPolicy: Always
19         ports:
20         - containerPort: 5000
```

## Cluster creation:



The screenshot shows the IBM Cloud console interface for a Kubernetes cluster named 'mycluster'. The cluster is in a 'Normal' state and expires in 30 days. The overview page displays various status metrics and details.

**Overview**

- Worker nodes: 1 of 1, Normal
- Worker pools: 0 of 0, Normal
- Master status: Normal
- Ingress status: Healthy

**Details**

Property	Value
Cluster ID	cdqmu0f9kea8knvlp3g
Version	1.24.8_1544
Infrastructure	Classic
Zones	Milan 01
Created	11/17/2022, 1:36 PM
Resource group	Default
Image security enforcement	<input type="button" value="Enable"/>

**Node health**

Worker node details