ASSIGNMENT-3

J.BRINDHA

VV COLLEGE OF ENGINEERING

III-YEAR CSE

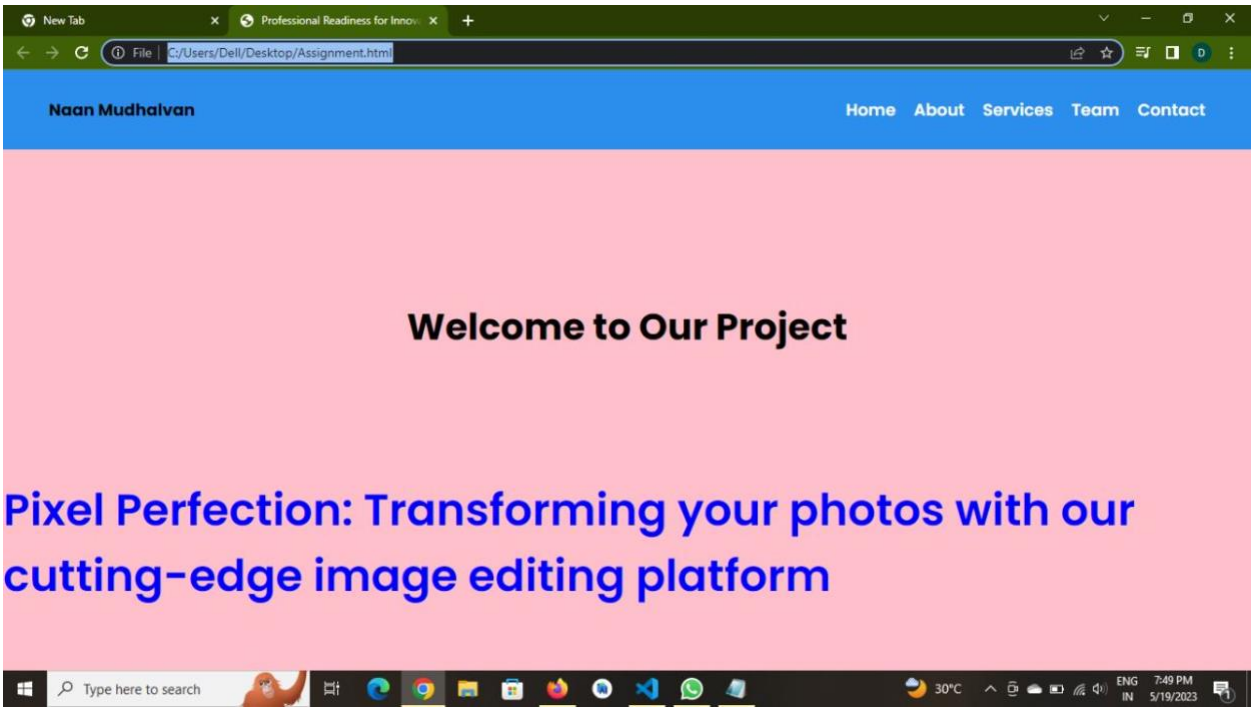
In the Assignment, you created 3 HTML files (index, profile and log-in), CSS files and stored the registration data in IBM db2 and did login validation.

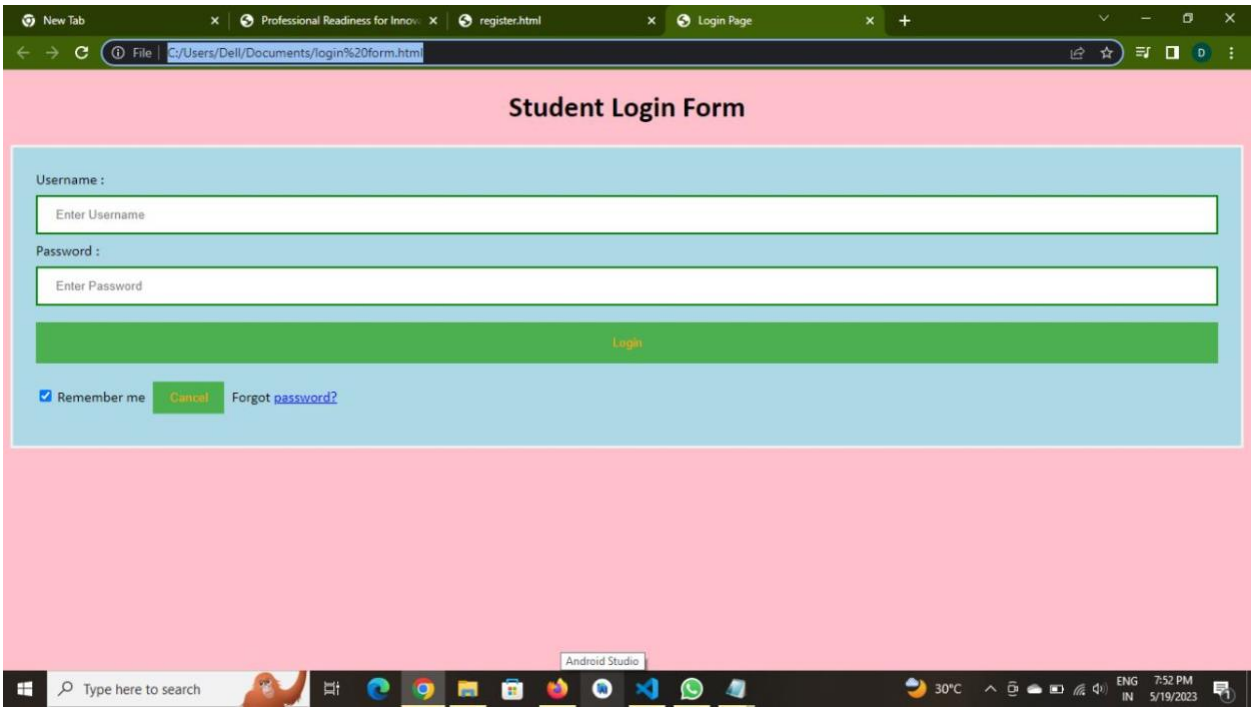
In this assignment follow these tasks

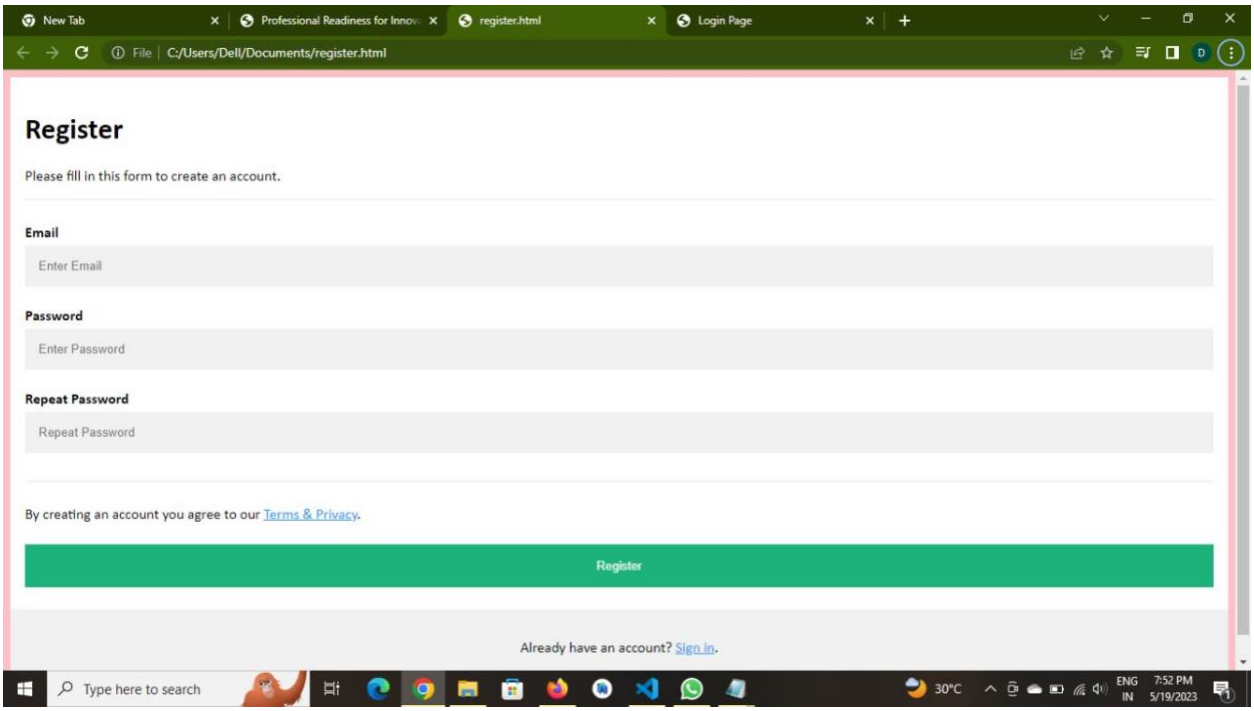
Task 1: You have to containerize the whole application by using the docker

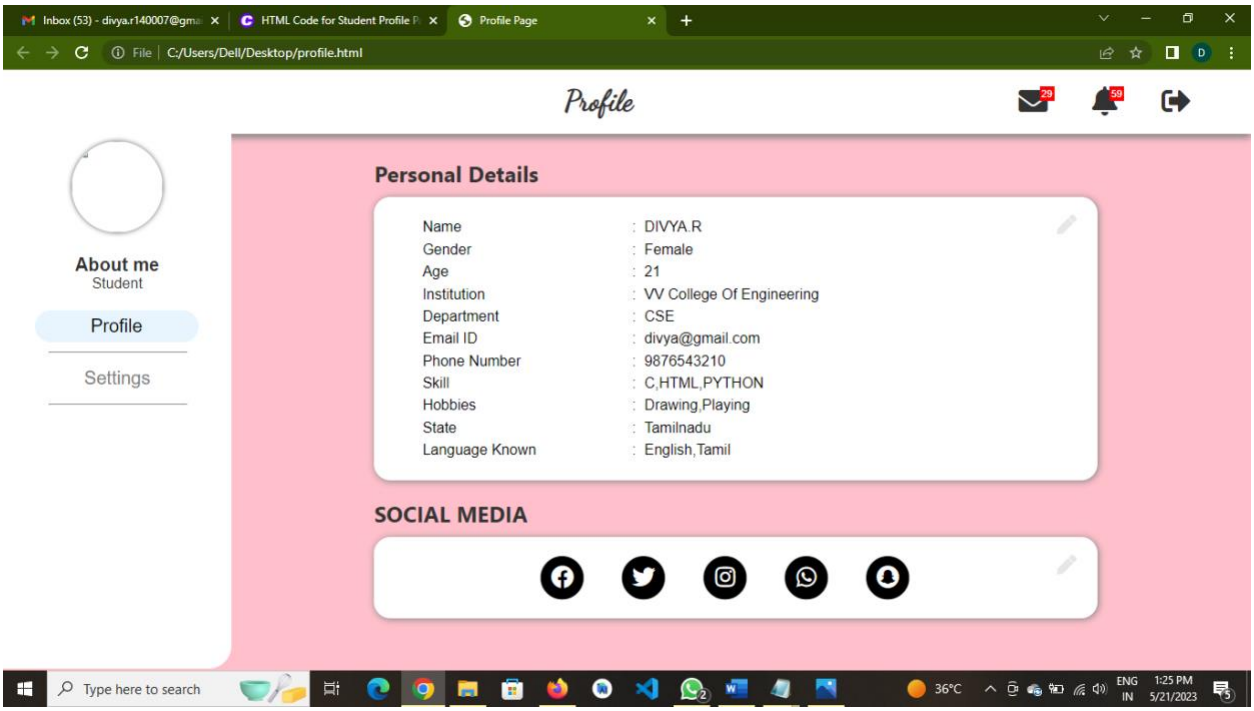
Task 2: Upload this docker image to the IBM container registry

Task 3: Then orchestrate the docker container using Kubernetes.


New Tab
Professional Readiness for Innov
File
C:/Users/Dell/Desktop/Assignment.html
Naan Mudhalvan
Home
About
Services
Team
Contact
Welcome to Our Project
Pixel Perfection: Transforming your photos with our cutting-edge image editing platform
Type here to search
30°C
ENG
IN
7:49 PM
5/19/2023


New Tab
Professional Readiness for Innov
register.html
Login Page
File
C:/Users/Dell/Documents/login%20form.html
Student Login Form
Username :
Enter Username
Password :
Enter Password
Login
Remember me
Cancel
Forgot password?
Android Studio
Type here to search
30°C
ENG
IN
7:52 PM
5/19/2023


New Tab
Professional Readiness for Innov
register.html
Login Page
File | C:/Users/Dell/Documents/register.html
Register
Please fill in this form to create an account.
Email
Enter Email
Password
Enter Password
Repeat Password
Repeat Password
By creating an account you agree to our Terms & Privacy.
Register
Already have an account? Sign in.
Type here to search
30°C
ENG
IN
7:52 PM
5/19/2023


Inbox (53) - divya.r140007@gma
HTML Code for Student Profile P
Profile Page
File | C:/Users/Dell/Desktop/profile.html
Profile
About me
Student
Profile
Settings
Personal Details
Name : DIVYA.R
Gender : Female
Age : 21
Institution : VV College Of Engineering
Department : CSE
Email ID : divya@gmail.com
Phone Number : 9876543210
Skill : C,HTML,PYTHON
Hobbies : Drawing,Playing
State : Tamilnadu
Language Known : English,Tamil
SOCIAL MEDIA
Type here to search
36°C
ENG
IN
1:25 PM
5/21/2023

Re: Welcome to the Professional | IBM | upload.html

File | C:/Users/Dell/Documents/upload.html

Click on the "Choose image" button to upload a image:

Choose File | No file chosen | Submit

Type here to search | 29°C Mostly clear | ENG IN | 8:08 PM 5/19/2023

Re: Welcome to the Professional | IBM | Build An Image Editor in HTML | Image Editor in JavaScript | Cod

File | C:/Users/Dell/Documents/image1.html

Easy Image Editor

Filters

Brightness | Saturation | Inversion | Grayscale

Brightness

100%

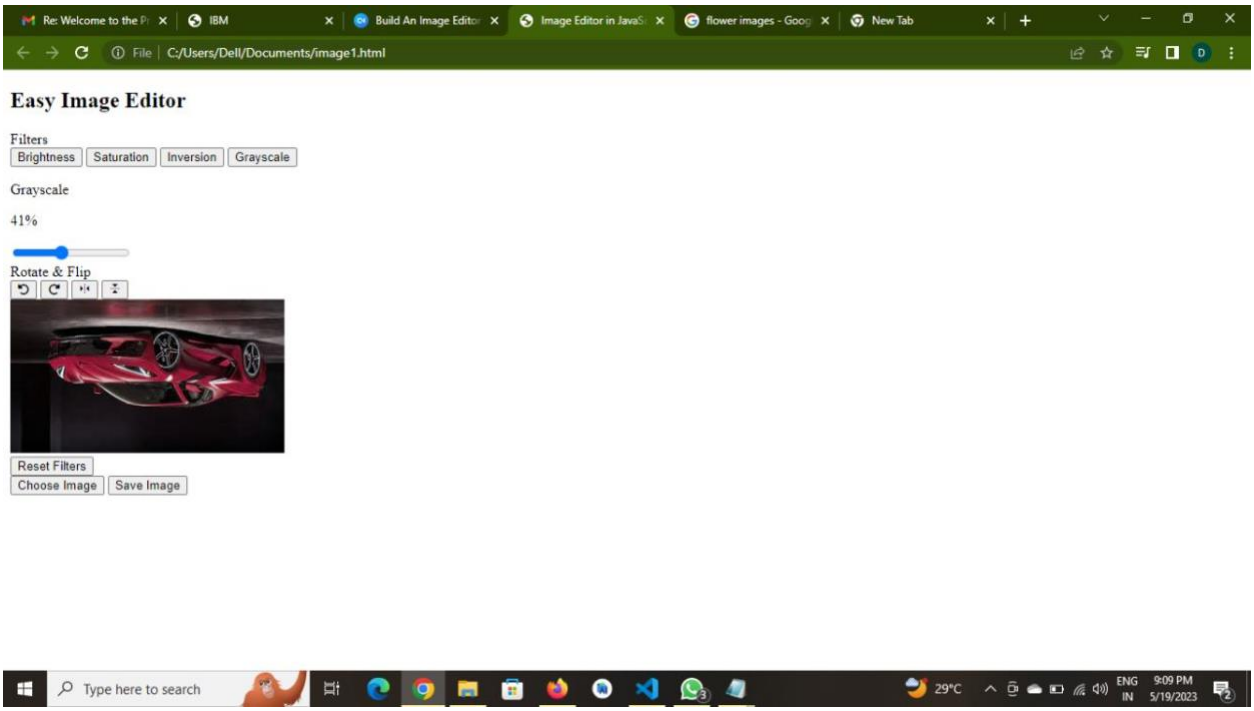
Rotate & Flip

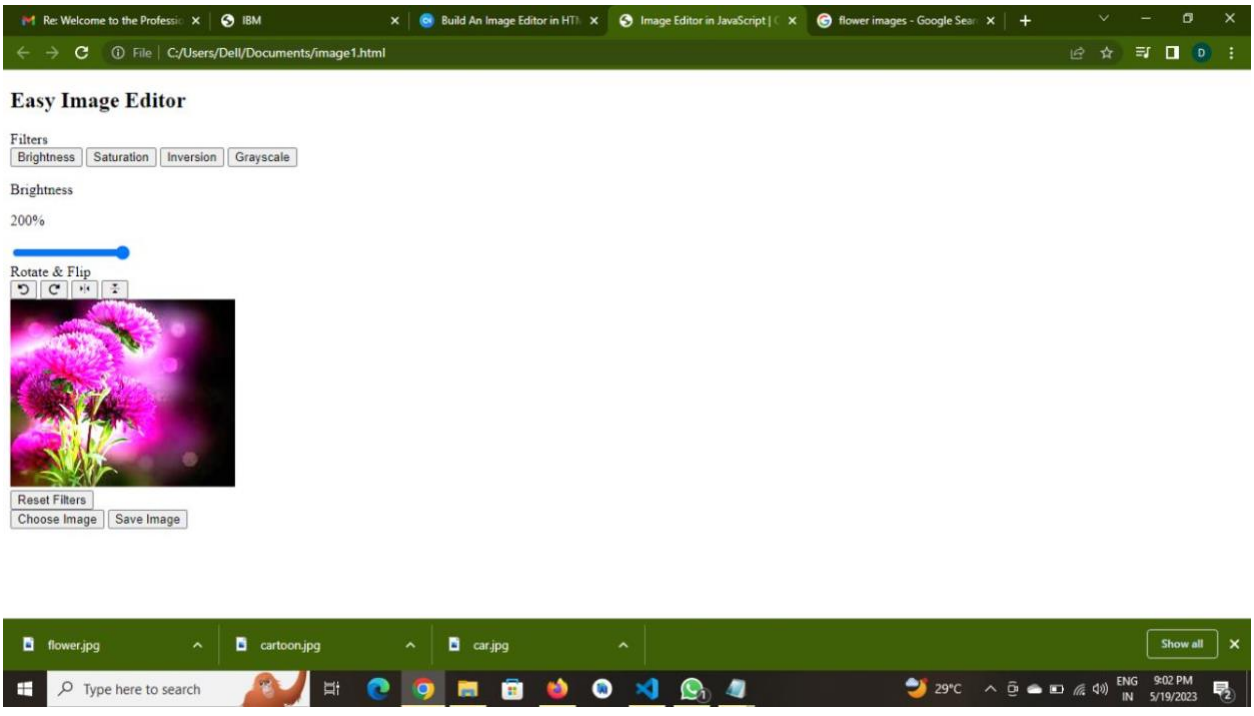
Reset Filters

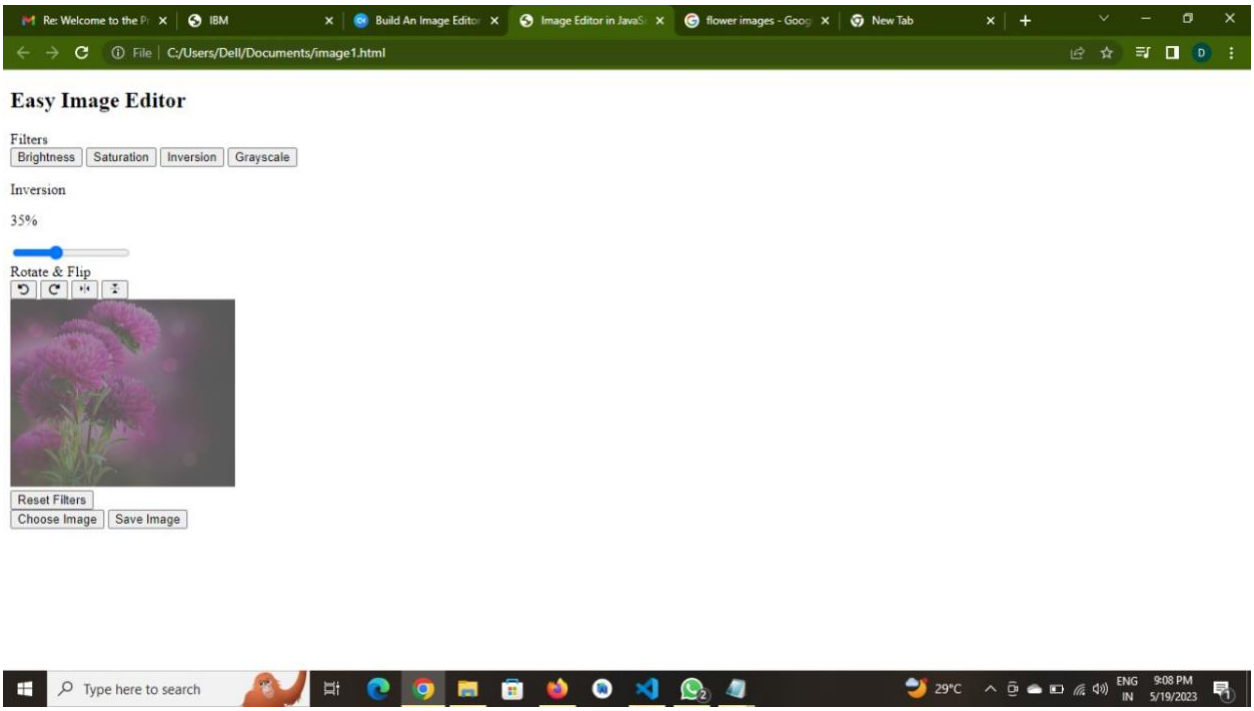
Choose Image | Save Image

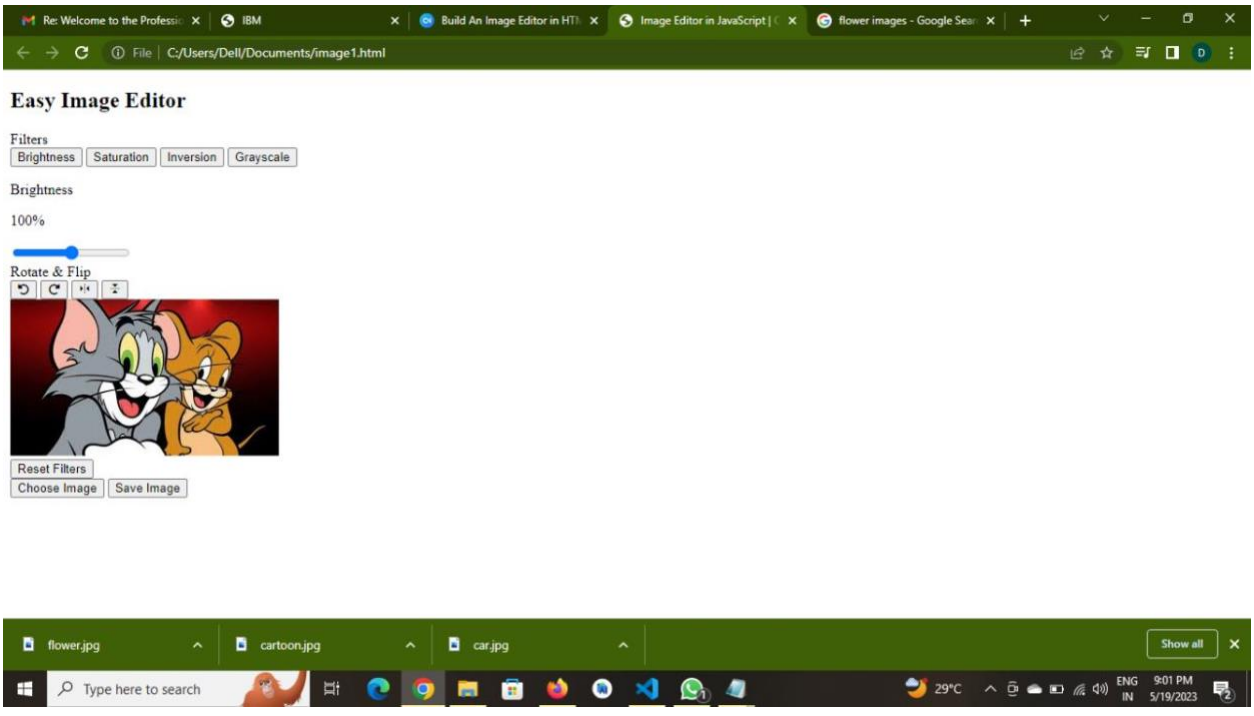
car.jpg | Show all

Type here to search | 29°C | ENG IN | 8:58 PM 5/19/2023


Re: Welcome to the Pr
IBM
Build An Image Editor
Image Editor in JavaS
flower images - Goog
New Tab
File | C:/Users/Dell/Documents/image1.html
Easy Image Editor
Filters
Brightness
Saturation
Inversion
Grayscale
Grayscale
41%
Rotate & Flip
Reset Filters
Choose Image
Save Image
Type here to search
29°C
ENG
IN
9:09 PM
5/19/2023


Re: Welcome to the Professic
IBM
Build An Image Editor in HTI
Image Editor in JavaScript | (
flower images - Google Sear
File | C:/Users/Dell/Documents/image1.html
Easy Image Editor
Filters
Brightness
Saturation
Inversion
Grayscale
Brightness
200%
Rotate & Flip
Reset Filters
Choose Image
Save Image
flower.jpg
cartoon.jpg
car.jpg
Show all
Type here to search
29°C
ENG
IN
9:02 PM
5/19/2023


Re: Welcome to the Pr
IBM
Build An Image Editor
Image Editor in JavaS
flower images - Goog
New Tab
File | C:/Users/Dell/Documents/image1.html
Easy Image Editor
Filters
Brightness
Saturation
Inversion
Grayscale
Inversion
35%
Rotate & Flip
Reset Filters
Choose Image
Save Image
Type here to search
29°C
ENG
IN
9:08 PM
5/19/2023


Re: Welcome to the Professic
IBM
Build An Image Editor in HTI
Image Editor in JavaScript |
flower images - Google Sear
File | C:/Users/Dell/Documents/image1.html
Easy Image Editor
Filters
Brightness
Saturation
Inversion
Grayscale
Brightness
100%
Rotate & Flip
Reset Filters
Choose Image
Save Image
flower.jpg
cartoon.jpg
car.jpg
Show all
Type here to search
29°C
ENG
IN
9:01 PM
5/19/2023

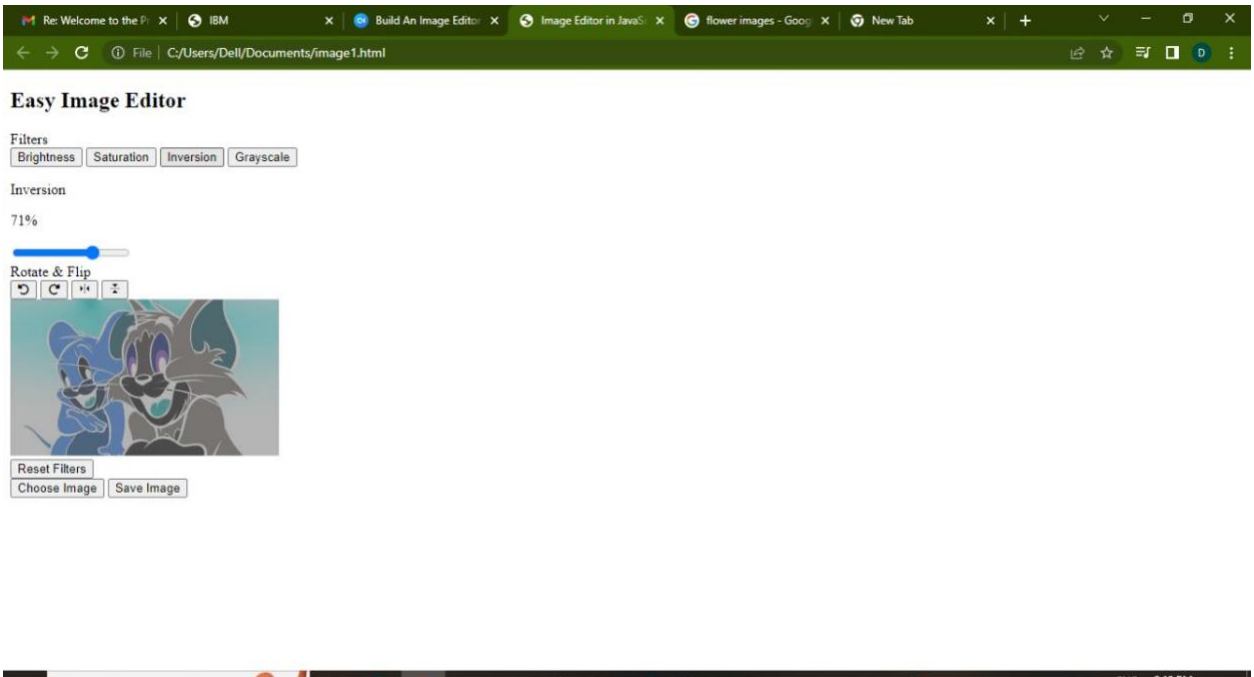


Table definition

USER1

No statistics available.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0
USERD	INTEGER	Y		0

FNC49721.USER1

Back

Export to CSV

NAME	EMAIL	PASSWORD	USERD
test1	test1@gmail.com	test1	1
test2	test2@gmail.com	test2	2

Table definition

IMAGE_URL

No statistics available.

Name	Data type	Nullable	Length	Scale
USERD	INTEGER	Y		0
IMAGE_BG	LONG VARCHAR	Y	32700	0
VEHICLE_BG	LONG VARCHAR	Y	32700	0
CARTOON_IMG	LONG VARCHAR	Y	32700	0
SKIN_BEAUTY	LONG VARCHAR	Y	32700	0
SKIN_DISEASE	LONG VARCHAR	Y	32700	0
IMG_MOTION	LONG VARCHAR	Y	32700	0
3DCARTOON_IMG	LONG VARCHAR	Y	32700	0

FNC49721.IMAGE_URL

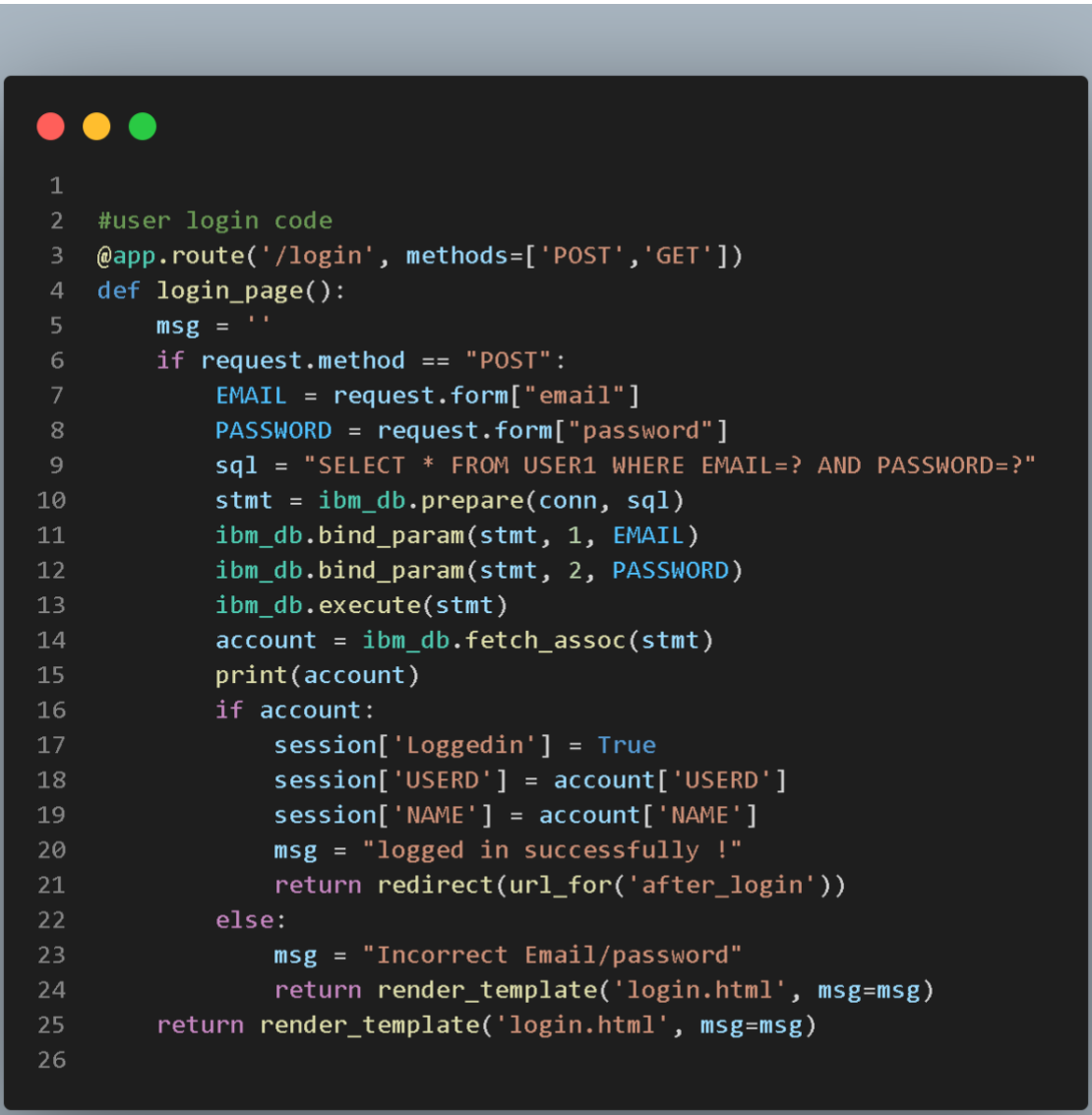
Back

Export to CSV

USERD	IMAGE_BG	VEHICLE_BG	CARTOON_IMG	SKIN_BEAUTY	SKIN_DISEASE	IMG_MOTION	3DCARTOON_IMG
1	https://ai-result-rapidapi.ailabtools.com/cutout/segmentCommodity/2023/02/28/100436-8394cf02-0827-70fb-6c42-10ea39c96cbb-1677578676.png						
1			https://ai-result-rapidapi.ailabtools.com/faceBody/generateHumanAnimeStyle/2023/03/05/084320-680f116c-eea2-f2b8-8414-a55d8bb322db-1678005800.png				
1	https://ai-result-rapidapi.ailabtools.com/cutout/segmentCommodity/2023/03/05/094439-6dc63a0d-a844-aec9-05e2-0a65ba9a07c6-1678009479.png						
1	https://ai-result-rapidapi.ailabtools.com/cutout/segmentCommodity/2023/03/05/094051-0fed12e6-d567-c314-4e2f-aa6ab31b11eb-1678009251.png						
1		https://ai-result-rapidapi.ailabtools.com/cutout/segmentVehicle/2023/03/04/171555-4bcb69e2-35dd-5608-3be4-e2004a54e98f-1677950155.png					
1				https://ai-result-rapidapi.ailabtools.com/faceBody/retouchSkin/2023/03/03/185451-			

```python
from flask import Flask, render_template, request, redirect, url_for, session


app = Flask(__name__)
app.secret_key = 'a'

@app.route("/")
def home():
    return "Hello world"

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

```python
#routing to home page
@app.route('/', methods=['POST','GET'])
def homepage():
    return render_template('index.html')

#routing to home page after login of user
@app.route('/home', methods=['POST','GET'
])def after_login():
    return render_template('home.html')
```



```python

#user login code
@app.route('/login', methods=['POST','GET'])
def login_page():
    msg = ''
    if request.method == "POST":
        EMAIL = request.form["email"]
        PASSWORD = request.form["password"]
        sql = "SELECT * FROM USER1 WHERE EMAIL=? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, EMAIL)
        ibm_db.bind_param(stmt, 2, PASSWORD)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin'] = True
            session['USERD'] = account['USERD']
            session['NAME'] = account['NAME']
            msg = "logged in successfully !"
            return redirect(url_for('after_login'))
        else:
            msg = "Incorrect Email/password"
            return render_template('login.html', msg=msg)
    return render_template('login.html', msg=msg)

```

```python
#user registartion code
@app.route('/register', methods=['POST','GET'])
def register_page():
    msg = ''
    if request.method == 'POST':
        NAME = request.form["name"]
        EMAIL = request.form["email"]
        PASSWORD = request.form["password"]
        sql = "SELECT* FROM USER1 WHERE EMAIL= ? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, EMAIL)
        ibm_db.bind_param(stmt, 2, PASSWORD)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = "Your deatils are already exists in the database Please logi
n"          return render_template('login.html')
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', EMAIL):
            msg = "Invalid Email Address!"
        else:
            sql = "SELECT count(*) FROM USER1"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            length = ibm_db.fetch_assoc(stmt)
            print(length)
            insert_sql = "INSERT INTO USER1 VALUES (?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, NAME)
            ibm_db.bind_param(prep_stmt, 2, EMAIL)
            ibm_db.bind_param(prep_stmt, 3, PASSWORD)
            ibm_db.bind_param(prep_stmt, 4, length['1']+1)
            ibm_db.execute(prep_stmt)
            msg = "Successfully registed!"
            return render_template('login.html', msg=msg)
    return render_template('register.html', msg=msg)
```

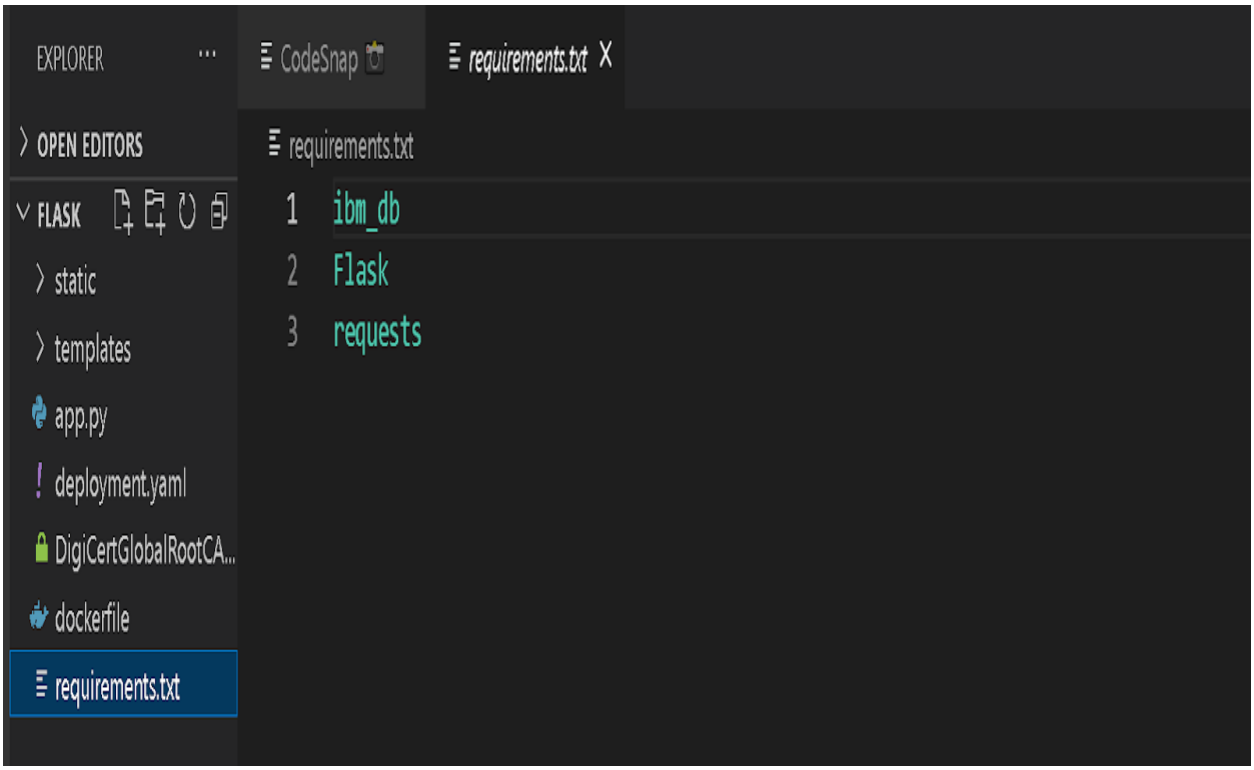
```python
# for logout the user start
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('USERD', None)
    return render_template('index.html'
)
```

```python
# this lines of code for displaying the images retrived from db2  sta
@app.route('/beauty_images')
def beauty_img_ai():
    user='login'
    # user='loggedin'
    sql = "SELECT * FROM IMAGE_URL WHERE USERD=" +str(session['USERD'
])  stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    row = []
    while True:
        data = ibm_db.fetch_assoc(stmt)
        if not data:
            break
        else:
            row.append(data)
    print('rows: ', row)
    return render_template("My_images.html", rows=row, user1=user)
```

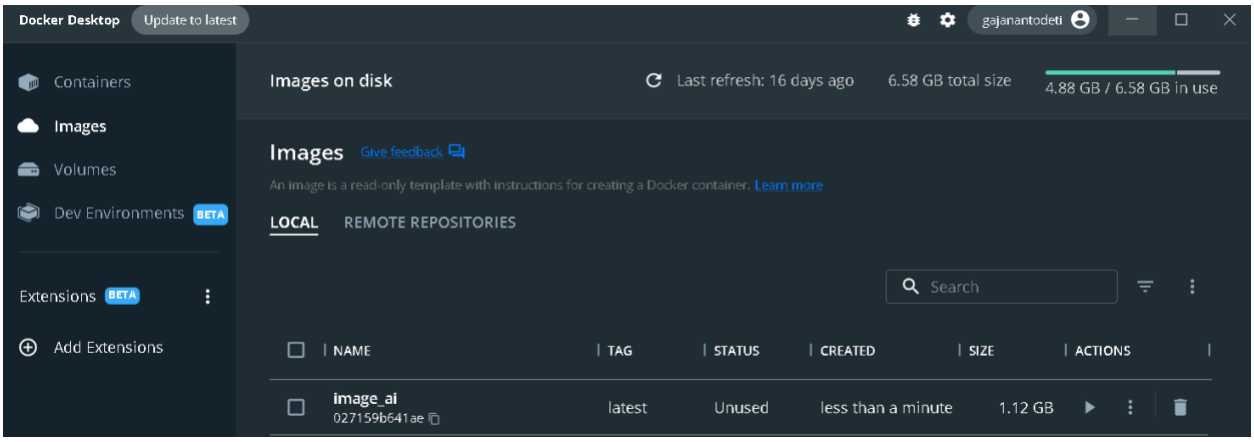
1 # main function start
2 if __name__=='__main__':
3 app.run(debug=True,host='0.0.0.0'
)

Task 1:

EXPLORER
CodeSnap
dockerfile
OPEN EDITORS
FLASK
static
templates
app.py
deployment.yaml
DigiCertGlobalRootCA...
dockerfile
requirements.txt
FROM python
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
EXPOSE 5000


EXPLORER
CodeSnap
requirements.txt
OPEN EDITORS
FLASK
static
templates
app.py
deployment.yaml
DigiCertGlobalRootCA...
dockerfile
requirements.txt
ibm_db
Flask
requests

Task 2:

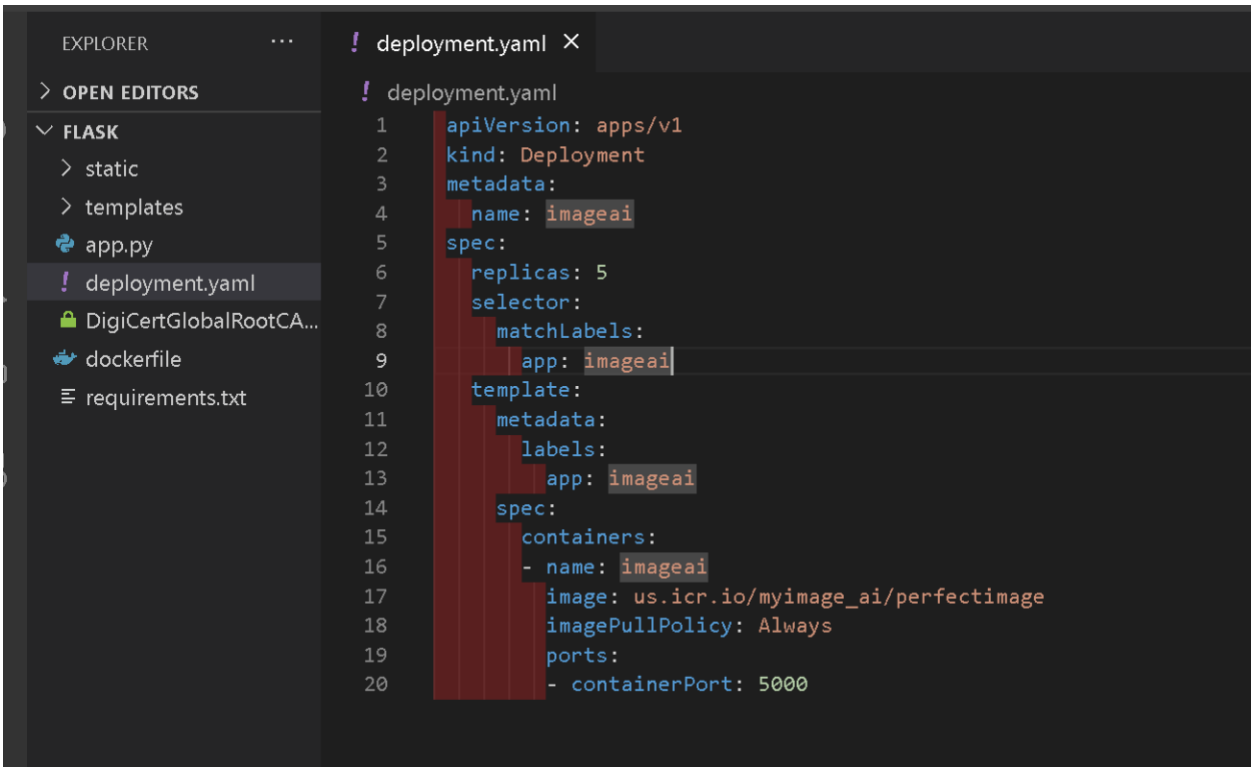


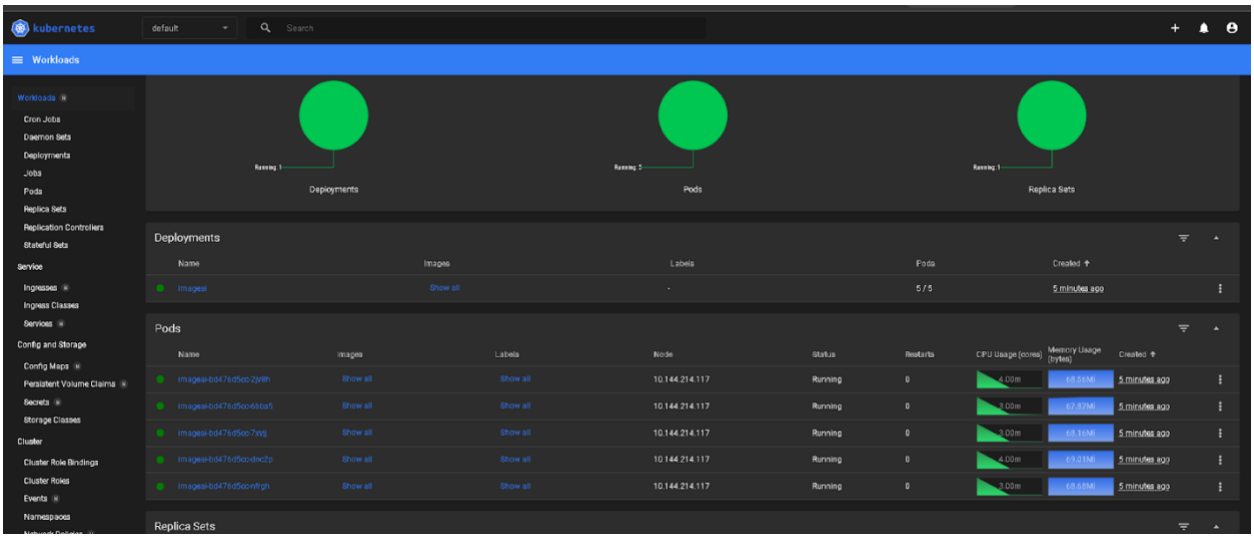

```
C:\Users\DELL> ibmcloud login
API endpoint: https://cloud.ibm.com
Region: eu-de
```

```
C:\Users\DELL>ibmcloud plugin install container-registry
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 1.0.6' found in repository 'IBM Cloud'
ac504d030134: Pushing [==============>                                    ]  160.3MB/528.8MB
6' or not? [y/N] > y
Attempting to download the binary file...
 11.91 MiB / 11.91 MiB [-------------------------------------------------------------------------------] 100.00% 5s
12484608 bytes downloaded
ac504d030134: Pushing [==============>                                    ]  158.7MB/528.8MB
OK
Plug-in 'container-registry 1.0.6' was successfully installed into C:\Users\DELL\.bluemix\plugins\container-registry. Use 'ibmcloud pl

C:\Users\DELL>ibmcloud cr region-set us-south
The region is set to 'us-south', the registry is 'us.icr.io'.

ac504d030134: Pushing [==============>                                    ]  157.5MB/528.8MB

C:\Users\DELL>ibmcloud cr namespace-add myimage_ai
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'myimage_ai' in resource group 'Default' for account gajanan todeti's Account in registry us.icr.io...

Successfully added namespace 'myimage_ai'

ac504d030134: Pushing [=============>                                     ]  147.6MB/528.8MB

ac504d030134: Pushing [--------------->                                   ]    157MB/528.8MB
Logging 'docker' in to 'us.icr.io'...
ac504d030134: Pushing [--------------->                                   ]  155.9MB/528.8MB
```

```
:\Users\DELL>docker tag image_ai us.icr.io/myimage_ai/perfectimage

:\Users\DELL>docker tag push us.icr.io/myimage_ai/perfectimage
rror response from daemon: No such image: push:latest

:\Users\DELL>docker push us.icr.io/myimage_ai/perfectimage
sing default tag: latest
he push refers to repository [us.icr.io/myimage_ai/perfectimage]
cd1e28035ea: Pushed
050e41efaf9: Pushed
4dd8426b147: Pushed
7985d454104: Pushed
041520c0f1e: Pushed
f706e6a9f98: Pushed
aa05faf31a: Pushed
504d030134: Pushing [==============>                                    ]  145.4MB/528.8MB
```

Task 3:


EXPLORER
deployment.yaml
OPEN EDITORS
FLASK
static
templates
app.py
deployment.yaml
DigiCertGlobalRootCA...
dockerfile
requirements.txt
deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: imageai
5 spec:
6 replicas: 5
7 selector:
8 matchLabels:
9 app: imageai
10 template:
11 metadata:
12 labels:
13 app: imageai
14 spec:
15 containers:
16 - name: imageai
17 image: us.icr.io/myimage_ai/perfectimage
18 imagePullPolicy: Always
19 ports:
20 - containerPort: 5000


kubernetes
default
Search
Workloads
Deployments
Pods
Replica Sets
Deployments
Pods
Replica Sets