* Create a frontend using Express and [Node.js](http://node.js). Include a form similar to the one from the Flask Assignment 2.

gallery-form.html

<!DOCTYPE html>

<html>

<head>

<title>Art Gallery Submission</title>

</head>

<body>

<h1>Submit Artwork Information</h1>

<form method="POST" action="http://localhost:5000/">

<label>Artwork Title:</label><br>

<input type="text" name="art\_title"><br><br>

<label>Artist Name:</label><br>

<input type="text" name="artist\_name"><br><br>

<label>Year Created:</label><br>

<input type="text" name="year\_created"><br><br>

<label>Medium:</label><br>

<input type="text" name="medium"><br><br>

<label>Price (optional):</label><br>

<input type="text" name="price"><br><br>

<button type="submit">Submit Artwork</button>

</form>

</body>

</html>

[index.js](http://index.js)

const express = require('express');

const path = require('path');

const app = express();

const PORT = 3000;

app.get('/', (req, res) => {

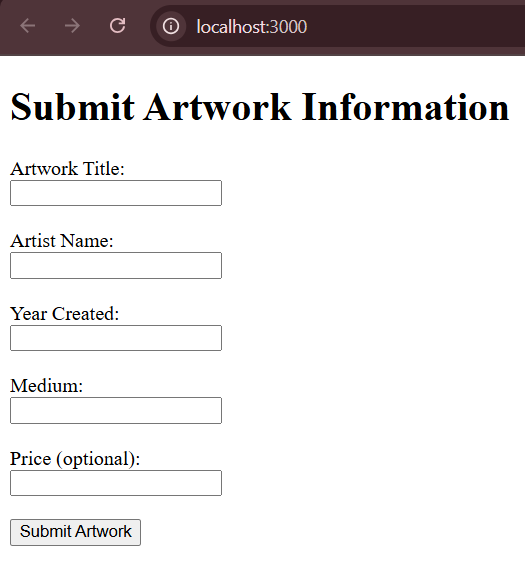
res.sendFile(path.join(\_\_dirname, 'templates', 'gallery\_form.html'));

});

app.listen(PORT, () => {

console.log(`Frontend running at http://localhost:${PORT}`);

});



* Use the Flask backend to handle the form submission and process the data.

app.py

from flask import Flask, request, redirect, url\_for, render\_template

from pymongo import MongoClient, errors

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

client = MongoClient("mongodb://localhost:27017/")

db = client["DevOps"]

collection = db["ArtGallery"]

@app.route("/", methods=["GET", "POST"])

def gallery\_form():

if request.method == "POST":

art\_title = request.form.get("art\_title")

artist\_name = request.form.get("artist\_name")

year\_created = request.form.get("year\_created")

medium = request.form.get("medium")

price = request.form.get("price")

try:

if not art\_title or not artist\_name:

raise ValueError("Artwork title and artist name are required.")

collection.insert\_one({

"art\_title": art\_title,

"artist\_name": artist\_name,

"year\_created": year\_created,

"medium": medium,

"price": price

})

return redirect(url\_for("success"))

except Exception as e:

return f"Error: {str(e)}"

return "Go to http://localhost:3000 to submit form."

@app.route("/success")

def success():

return render\_template("success.html")

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=False, host='0.0.0.0', port=5000)

[index.js](http://index.js)

const express = require('express');

const path = require('path');

const bodyParser = require('body-parser');

const axios = require('axios');

const app = express();

const PORT = 3000;

app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static('views'));

app.set('view engine', 'html');

app.engine('html', require('ejs').renderFile);

// Show form

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'views', 'gallery\_form.html'));

});

// Handle form submission → send to Flask

app.post('/submit', async (req, res) => {

try {

await axios.post('http://localhost:5000/', req.body);

res.redirect('/success');

} catch (error) {

console.error('Error sending data to backend:', error.message);

res.status(500).send('Error sending data to backend.');

}

});

// Show success

app.get('/success', (req, res) => {

res.send('<h1>Form submitted successfully!</h1>');

});

app.listen(PORT, () => {

console.log(`Frontend running at http://localhost:${PORT}`);

});

success.html

<!DOCTYPE html>

<html>

<head>

<title>Success</title>

</head>

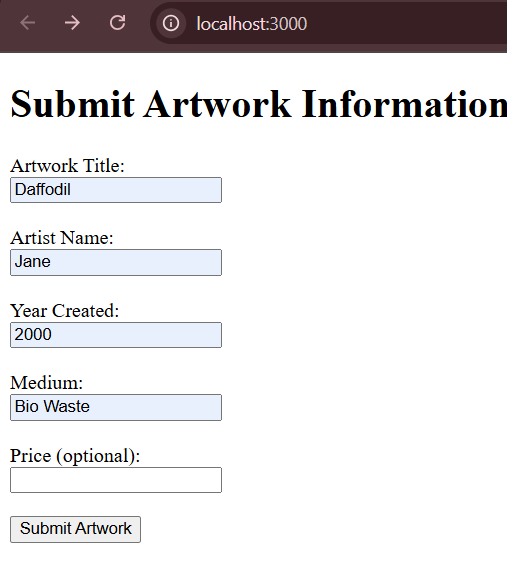
<body>

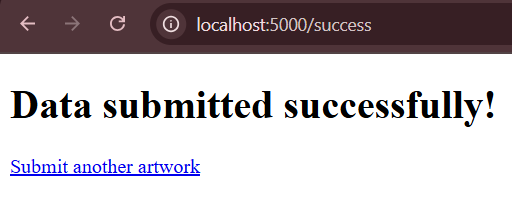
<h1>Data submitted successfully!</h1>

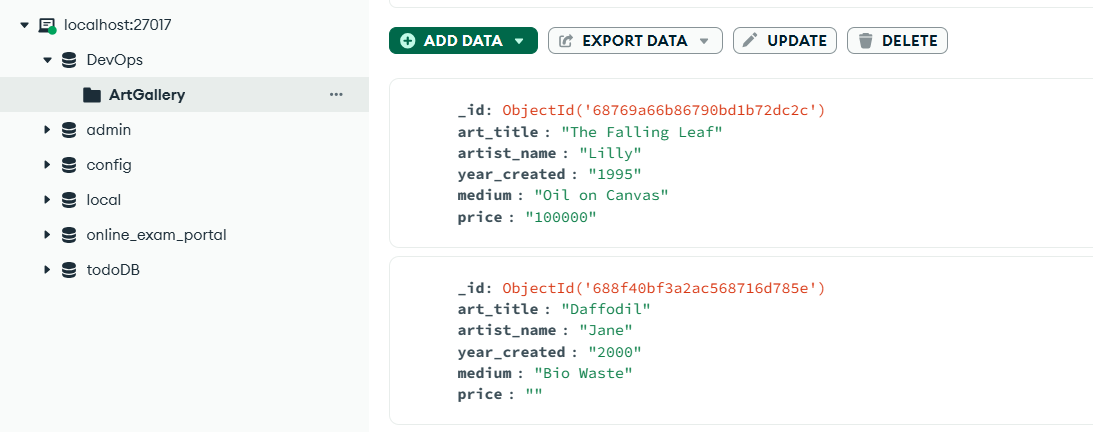
<a href="http://localhost:3000/">Submit another artwork</a>

</body>

</html>







* Organize the project with separate folders for the frontend and backend. Create a Dockerfile for both the frontend and backend. Write a .yaml file (Docker Compose) to connect both services in the same network.

version: "3.8"

docker-compose.yaml

services:

backend:

build:

context: ./backend

ports:

- "5000:5000"

networks:

- gallery-network

frontend:

build:

context: ./frontend

ports:

- "3000:3000"

depends\_on:

- backend

networks:

- gallery-network

networks:

gallery-network:

driver: bridge

Dockerfile - frontend

FROM node:18-alpine

WORKDIR /app

COPY frontend/package\*.json ./

RUN npm install

COPY frontend/ .

EXPOSE 3000

CMD ["node", "index.js"]

Dockerfile - backend

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

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

.gitgore

node\_modules/

\_\_pycache\_\_/

\*.pyc

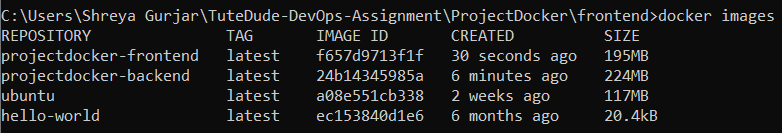
.DS\_Store

.vscode/

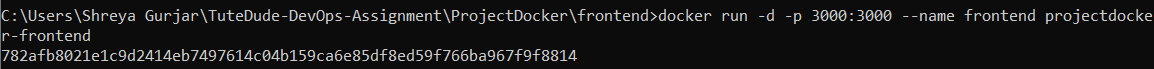
\*.log

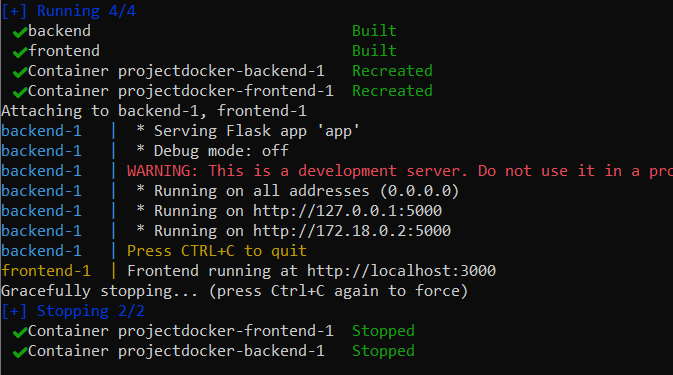
.env

\*.pid

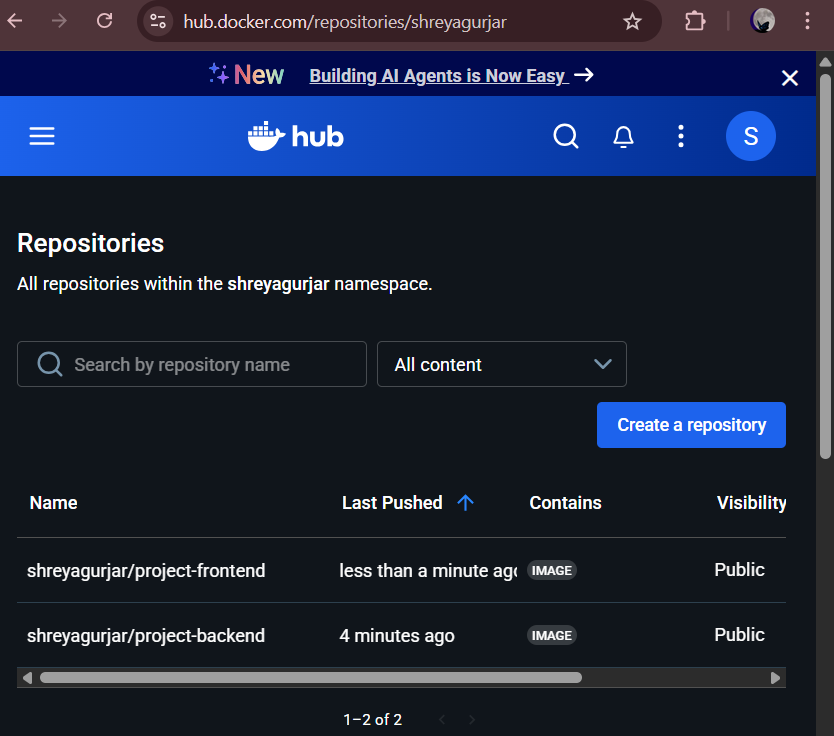
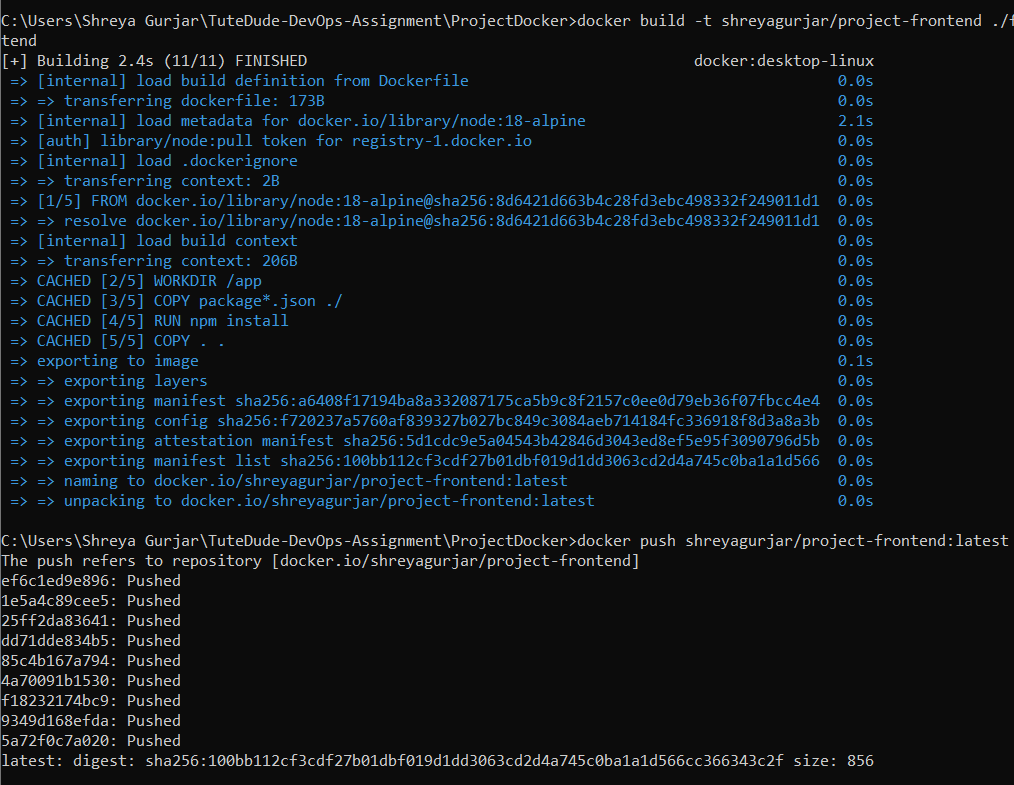
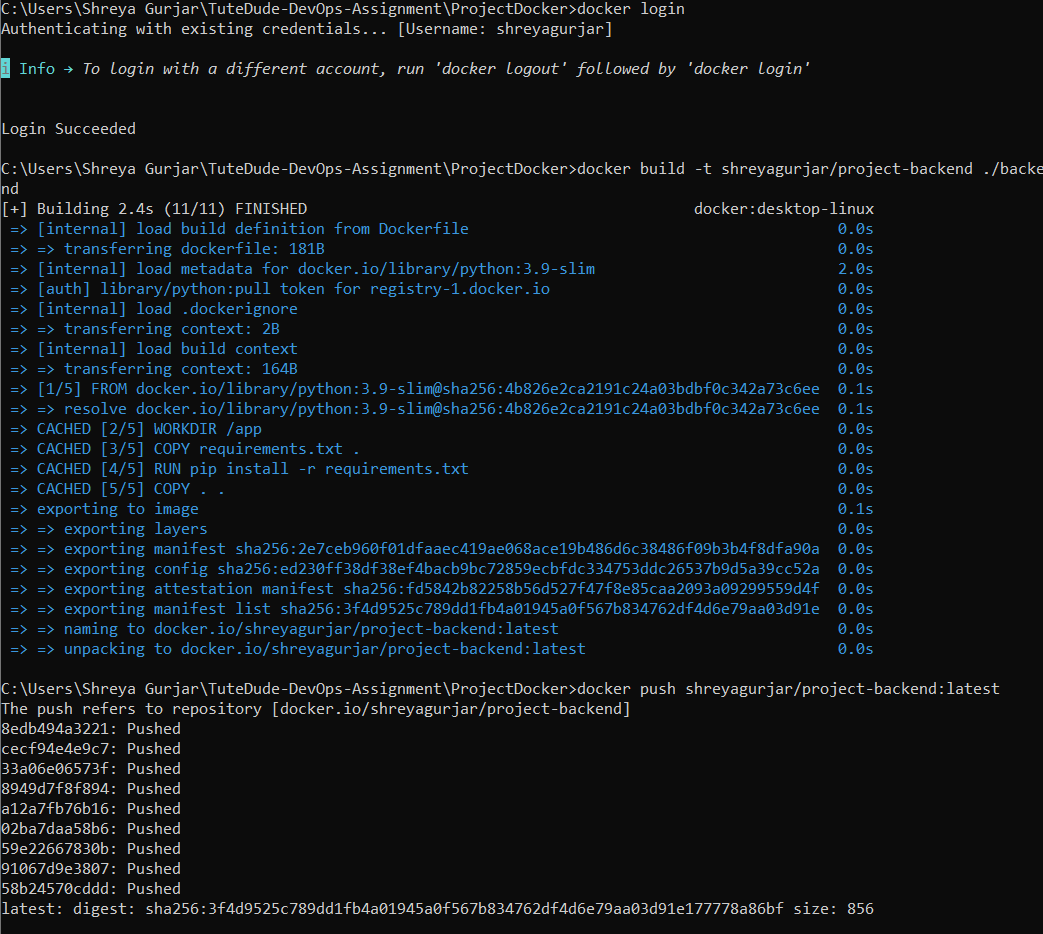








* Upload both images to docker hub



* Push your whole code to github and add the node\_modules and other non required files(.vscode) in .gitignore

