Assignment 5

1. Frontend (Node.js with Express)

Explanation:

You are required to build a frontend using Node.js and Express. This frontend should include a form (like Assignment 2) that will send data to a Flask backend.

Steps:

- Set up a basic Express server.

- Create a form with fields like ‘Name’, ‘Email’, or ‘Item’ depending on your design.

- Configure the form to send POST requests to the Flask server.

Commands:

Npm init -y

Npm install express body-parser

Example Code (server.js):

Const express = require(‘express’);

Const bodyParser = require(‘body-parser’);

Const app = express();

App.use(bodyParser.urlencoded({ extended: false }));

App.post(‘/submit’, (req, res) => {

Console.log(req.body);

Res.send(‘Form submitted’);

});

App.listen(3000, () => console.log(‘Server running on port 3000’));

1. Backend (Flask)

Explanation:

You will receive the form data from the Node.js frontend and process/store it in Flask.

Steps:

- Create a Flask route `/submit` that accepts POST requests.

- Process and validate incoming form data.

Code (app.py):

From flask import Flask, request

App = Flask(\_\_name\_\_)

@app.route(‘/submit’, methods=[‘POST’])

Def handle\_submit():

Data = request.form

Print(data)

Return “Data received”, 200

If \_\_name\_\_ == “\_\_main\_\_”:

App.run(debug=True, port=5000)

1. Folder Structure

Structure:

Project-root/

├── frontend/ (Node.js)

└── server.js

├── backend/ (Flask)

└── app.py

├── docker-compose.yaml

├── .gitignore

1. Docker Configuration

Dockerfile for Node.js (frontend/Dockerfile):

FROM node:14

WORKDIR /app

COPY . .

RUN npm install

EXPOSE 3000

CMD [“node”, “server.js”]

Dockerfile for Flask (backend/Dockerfile):

FROM python:3.8

WORKDIR /app

COPY . .

RUN pip install flask

EXPOSE 5000

CMD [“python”, “app.py”]

Docker-compose.yaml:

Version: “3.8”

Services:

Frontend:

Build: ./frontend

Ports:

* “3000:3000”

Networks:

* Sharednet

Backend:

Build: ./backend

Ports:

* “5000:5000”

Networks:

* Sharednet

Networks:

Sharednet:

5. Final Upload

- Push both images to Docker Hub using:

Docker tag <image\_id> yourdockerhubusername/image-name

Docker push yourdockerhubusername/image-name

* Push your project to GitHub:

Git init

Git add .

Git commit -m “Docker assignment submission”

Git remote add origin <https://github.com/yourusername/docker-assignment.git>

Git push -u origin main

* .gitignore file content:

Node\_modules/

.vscode/

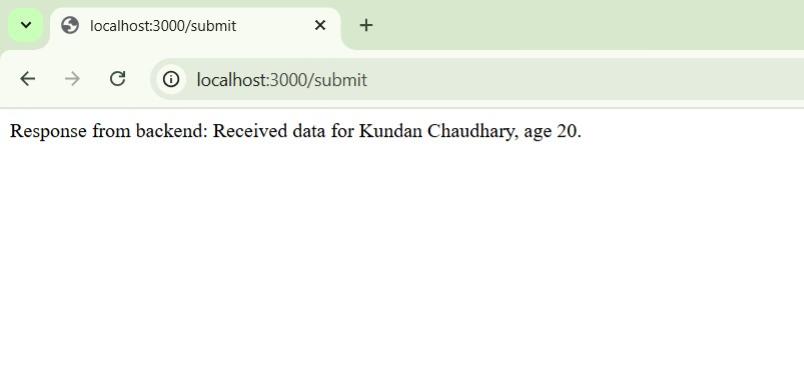
\_\_pycache\_\_/

\*.pyc

.env

Output:





Repo link: https://github.com/BabluuMemes20005/Tutude-Devops-Assignment