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'));

# 2. 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)

# 3. Folder Structure

Structure:  
 project-root/  
 ├── frontend/ (Node.js)  
       └── server.js  
 ├── backend/ (Flask)  
       └── app.py  
 ├── docker-compose.yaml  
 ├── .gitignore

# 4. 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

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