Full Stack Assignment Report

Project Title: Flask + Express Full Stack Form Submission with Docker

Student Name: [Rushikesh Mulik]

# 📁 Folder Structure

flaskp/  
├── backend/  
│ ├── app.py  
│ ├── requirements.txt  
│ ├── Dockerfile  
├── frontend/  
│ ├── index.js  
│ ├── public/  
│ │ └── index.html  
│ ├── package.json  
│ ├── Dockerfile  
├── docker-compose.yaml  
├── .gitignore  
├── screenshots/

# 🌐 Frontend (Node.js + Express)

## frontend/index.js

const express = require('express');  
const path = require('path');  
const app = express();  
  
app.use(express.static(path.join(\_\_dirname, 'public')));  
app.listen(3000, () => console.log('Frontend running on http://localhost:3000'));

## frontend/public/index.html

<!DOCTYPE html>  
<html>  
<head><title>Form</title></head>  
<body>  
 <h1>Submit Form</h1>  
 <form id="userForm">  
 Name: <input type="text" name="name" required /><br/>  
 Email: <input type="email" name="email" required /><br/>  
 <button type="submit">Submit</button>  
 </form>  
 <p id="response"></p>  
 <script>  
 document.getElementById("userForm").onsubmit = async (e) => {  
 e.preventDefault();  
 const name = e.target.name.value;  
 const email = e.target.email.value;  
 const res = await fetch("http://localhost:5000/submit", {  
 method: "POST",  
 headers: { "Content-Type": "application/json" },  
 body: JSON.stringify({ name, email })  
 });  
 const data = await res.json();  
 document.getElementById("response").innerText = data.message;  
 };  
 </script>  
</body>  
</html>

## frontend/package.json

{  
 "name": "frontend",  
 "version": "1.0.0",  
 "main": "index.js",  
 "scripts": {  
 "start": "node index.js"  
 },  
 "dependencies": {  
 "express": "^4.18.2"  
 }  
}

## frontend/Dockerfile

FROM node:18  
WORKDIR /app  
COPY . .  
RUN npm install  
EXPOSE 3000  
CMD ["npm", "start"]

# 🧩 Backend (Flask)

## backend/app.py

from flask import Flask, request, jsonify  
from flask\_cors import CORS  
  
app = Flask(\_\_name\_\_)  
CORS(app)  
  
@app.route('/submit', methods=['POST'])  
def submit():  
 data = request.json  
 name = data.get('name')  
 email = data.get('email')  
 return jsonify({"message": f"Received data for {name} with email {email}"})  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(host='0.0.0.0', port=5000)

## backend/requirements.txt

flask  
flask-cors

## backend/Dockerfile

FROM python:3.10-slim  
WORKDIR /app  
COPY . .  
RUN pip install -r requirements.txt  
EXPOSE 5000  
CMD ["python", "app.py"]

# 🐳 Docker Configuration

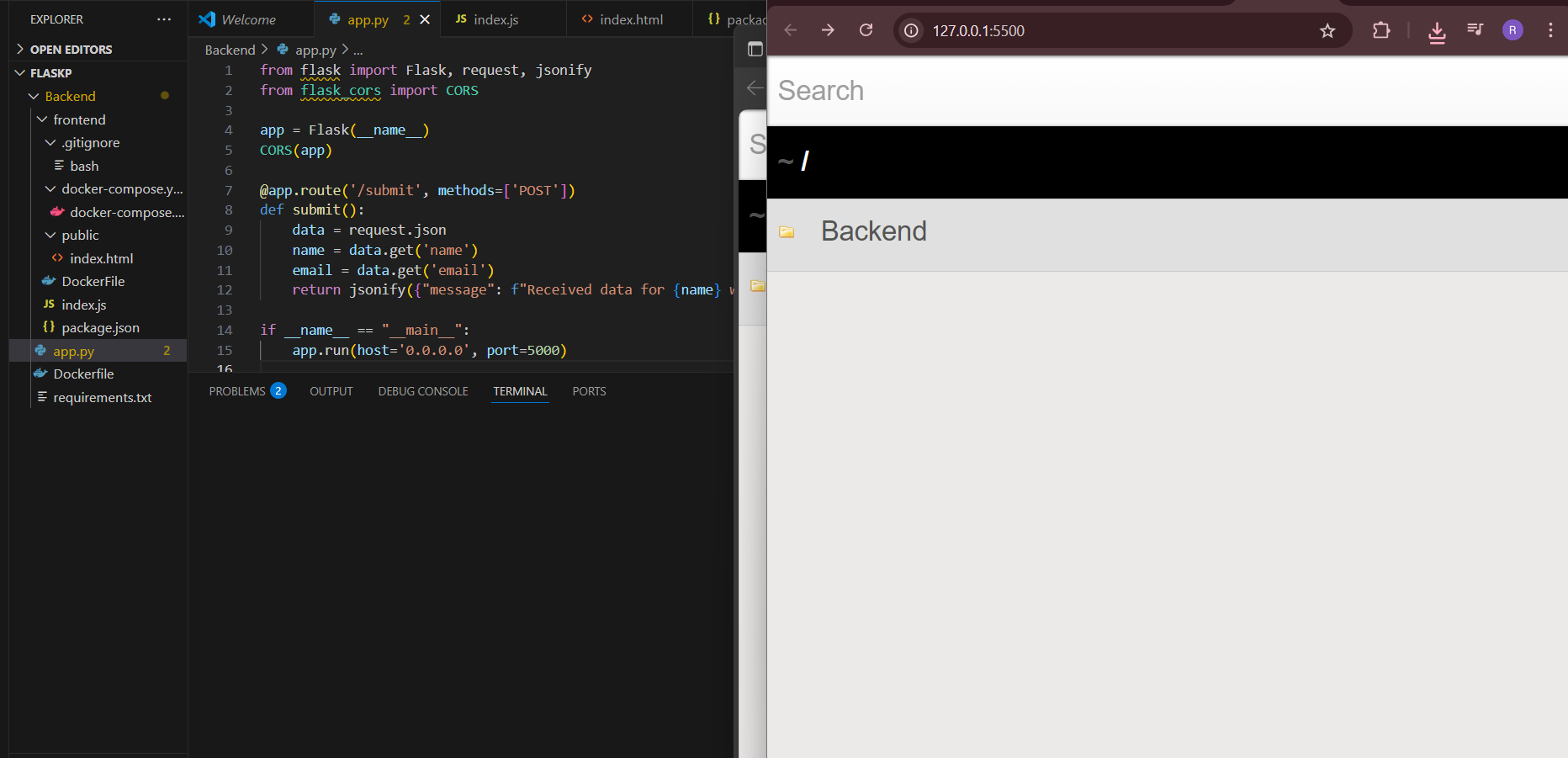
## docker-compose.yaml

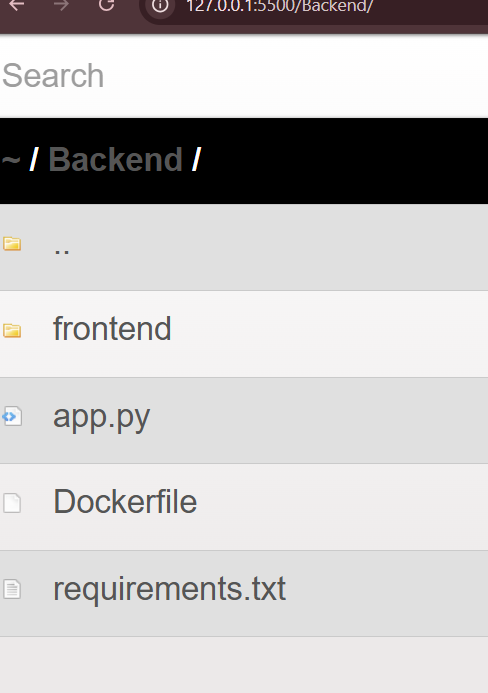
version: '3'  
services:  
 frontend:  
 build: ./frontend  
 ports:  
 - "3000:3000"  
 networks:  
 - app-network  
  
 backend:  
 build: ./backend  
 ports:  
 - "5000:5000"  
 networks:  
 - app-network  
  
networks:  
 app-network:

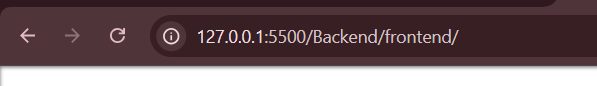
# .gitignore

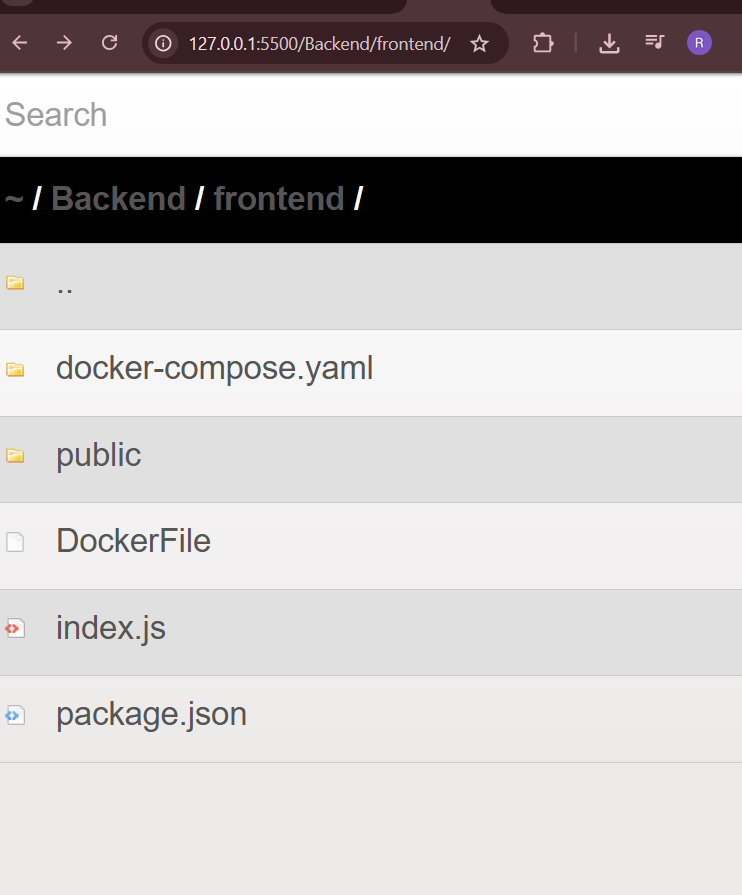
node\_modules/  
.vscode/  
\_\_pycache\_\_/  
.env  
\*.pyc

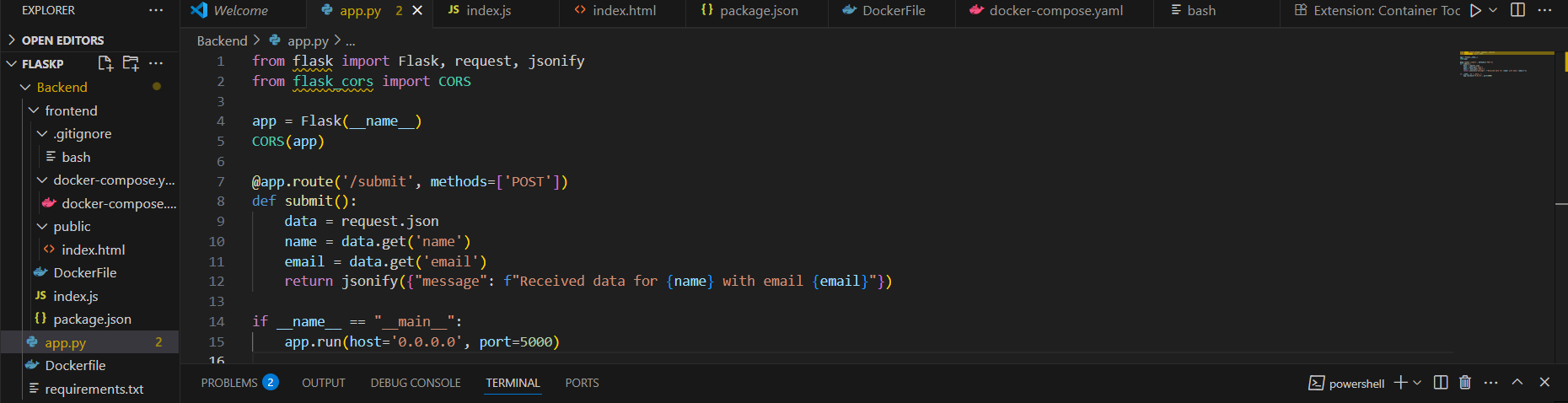
# 📸 Screenshots

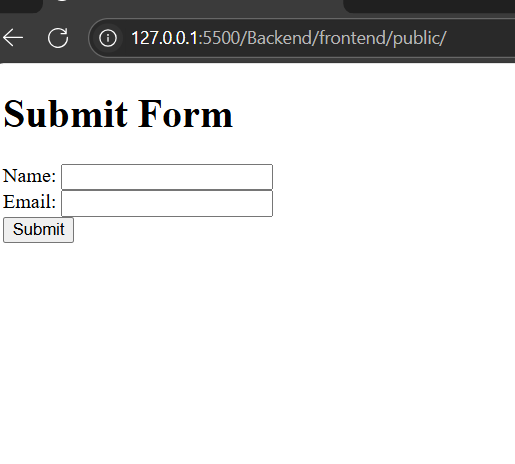










 ✅ Deployment Steps

## GitHub:

git init  
git add .  
git commit -m "Initial commit"  
git remote add origin https://github.com/Toto3107/flaskp.git  
git branch -M main  
git push -u origin main

## Docker Hub:

# Build and push frontend  
docker build -t yourdockerhubusername/frontend ./frontend  
docker push yourdockerhubusername/frontend  
  
# Build and push backend  
docker build -t yourdockerhubusername/backend ./backend  
docker push yourdockerhubusername/backend

# 📝 Conclusion

This assignment demonstrates full-stack development with Node.js, Flask, and Docker. It includes form submission through a REST API, backend data handling, frontend interaction, and containerized deployment with Docker and Docker Compose.