# Dockerized Flask Backend & Express Frontend Application

# 1. Intoduction

This assignment demonstrates containerization of a two-tier application using Docker and Docker Compose. The project consists of:  
- Frontend: Node.js with Express server serving a form and proxying API requests.  
- Backend: Flask API handling form submissions.  
Both services run in isolated containers, communicate over a Docker network, and are orchestrated using Docker Compose.

# 2. Objectives

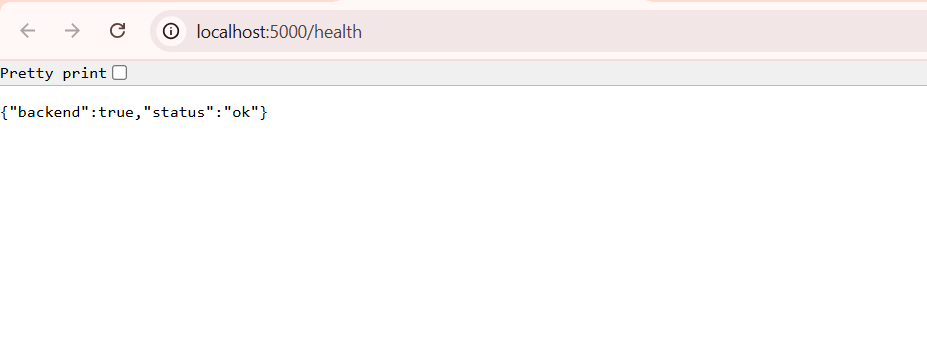
- Develop a Flask backend to handle API requests.  
- Create a Node.js Express frontend to serve a static form and proxy backend requests.  
- Containerize both applications using Docker.  
- Use Docker Compose to orchestrate both containers in a single network.  
- Push built images to Docker Hub.  
- Host the complete project on GitHub with `.gitignore` for unnecessary files.

# 3. Folder Structure

Docker-Assignment/  
│  
├── backend/  
│ ├── app.py  
│ ├── requirements.txt  
│ ├── Dockerfile  
│  
├── frontend/  
│ ├── server.js  
│ ├── public/  
│ │ └── index.html  
│ ├── package.json  
│ ├── Dockerfile  
│  
└── docker-compose.yml

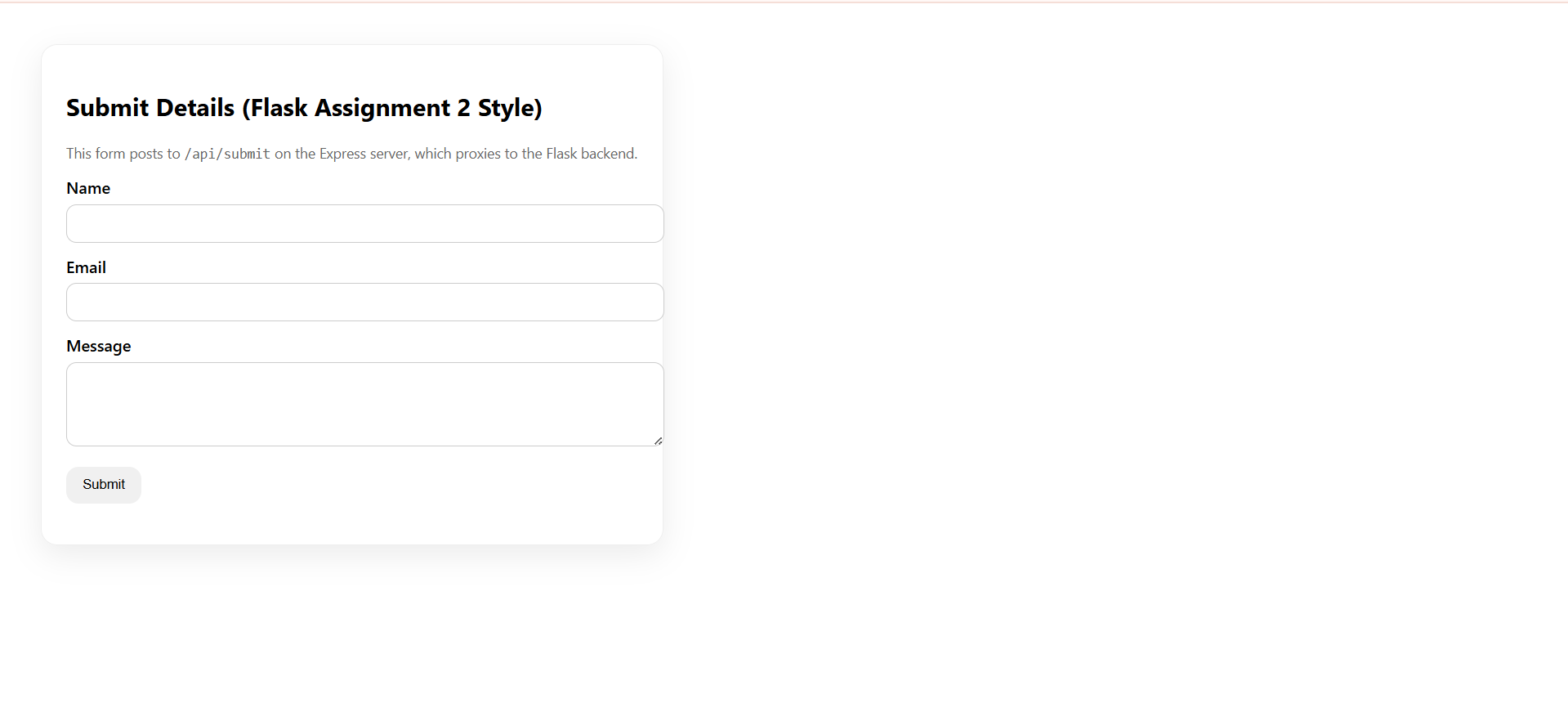
# 5. Backend Implementation (Flask)

Language: Python  
Framework: Flask  
Key Routes:  
- /health — Health check endpoint  
- /api/submit — Accepts form data in JSON and returns a success response  
Technologies: Flask, Flask-CORS



# 6. Frontend Implementation (Express)

Language: JavaScript (Node.js)  
Framework: Express  
Key Features:  
- Serves static HTML form  
- Proxies /api/submit requests to Flask backend  
Technologies: Express, Axios, Morgan

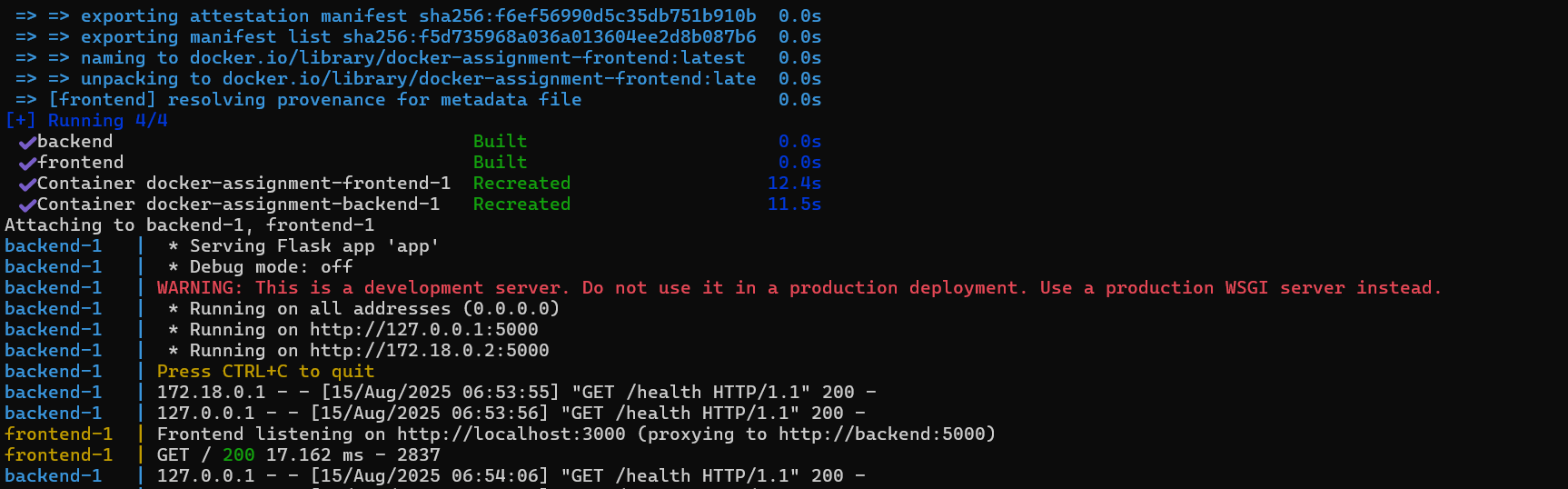


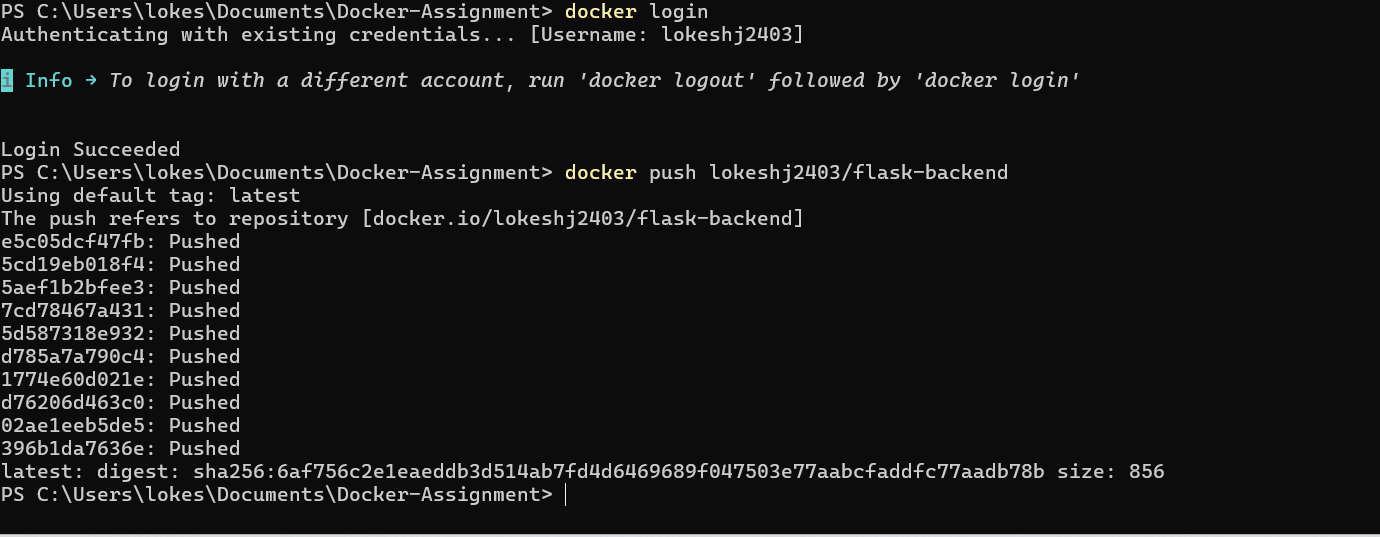
# 7. Docker Compose Configuration

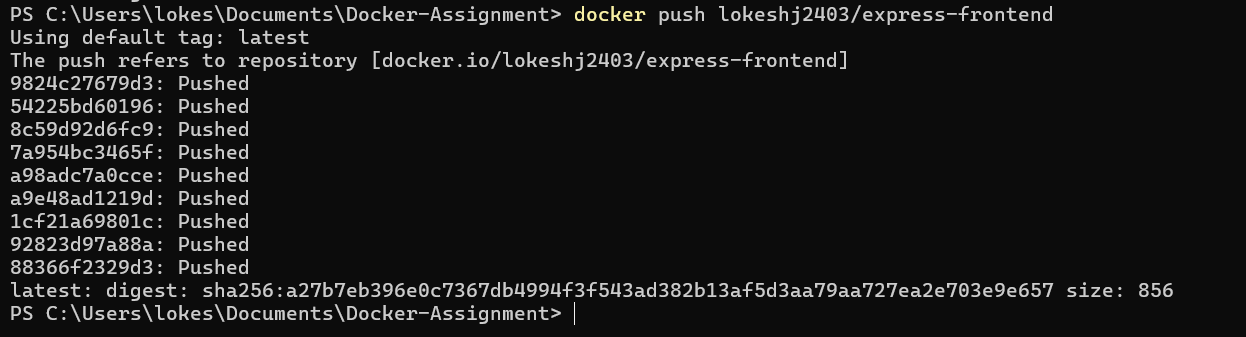


# 8. Steps to Build and Run

Build Images Locally:  
docker build -t lokeshj2403/flask-backend ./backend  
docker build -t lokeshj2403/express-frontend ./frontend  
  
Run with Docker Compose:  
docker compose up --build  
  
Access Application:  
- Frontend: http://localhost:3000  
- Backend Health: <http://localhost:5000/health>







# 9. Docker Hub & GitHub Links

- Docker Hub Backend: <https://hub.docker.com/repository/docker/lokeshj2403/flask-backend/general>

- Docker Hub Frontend: <https://hub.docker.com/repository/docker/lokeshj2403/express-frontend/general>

- GitHub Repository: https://github.com/lokeshj2403/DevOps/tree/main/Docker-Assignment

# 10. Conclusion

This assignment successfully demonstrates containerization of a multi-service application using Docker and Docker Compose. It shows how independent services can communicate over an isolated network, ensuring environment consistency and easy deployment.