



**National Textile University**  
**Department of Computer Science Subject:**

Operating System

---

Submitted by:  
**ABDUL REHMAN SUDAIS**

---

Reg number:  
**23-NTU-CS-1123**

---

Lab: no  
**DOCKER HOME TASK**

---

Semester:  
**5<sup>th</sup>**

**Home task**

## Task 1:

### Create networks to facilitate communication between frontend, backend and database.

#### Code :

version: "3.9" services:

# MongoDB Database (only backend network)

mongodb:

image: mongo:7-jammy

container\_name: guestbook-db

restart: always volumes:

- mongo\_data:/data/db networks:

- backend\_net ( connected to backend )

# Backend API

api:

build: ./backend

image: backend ( my image of backend)

container\_name: guestbook-api environment:

- MONGO\_URL=mongodb://mongodb:27017/guestbook

ports:

- "3000:3000" depends\_on: - mongodb networks:

- frontend\_net

- backend\_net ( connected to both frontend and database)

# Frontend

web: image:

nginx:alpine ( my

image of

frontend)

container\_name:

guestbook-web

restart: always

ports:

- "8080:80"( port on which html frontend will display)

volumes:

- ./frontend:/usr/share/nginx/html:ro depends\_on:

- api networks:

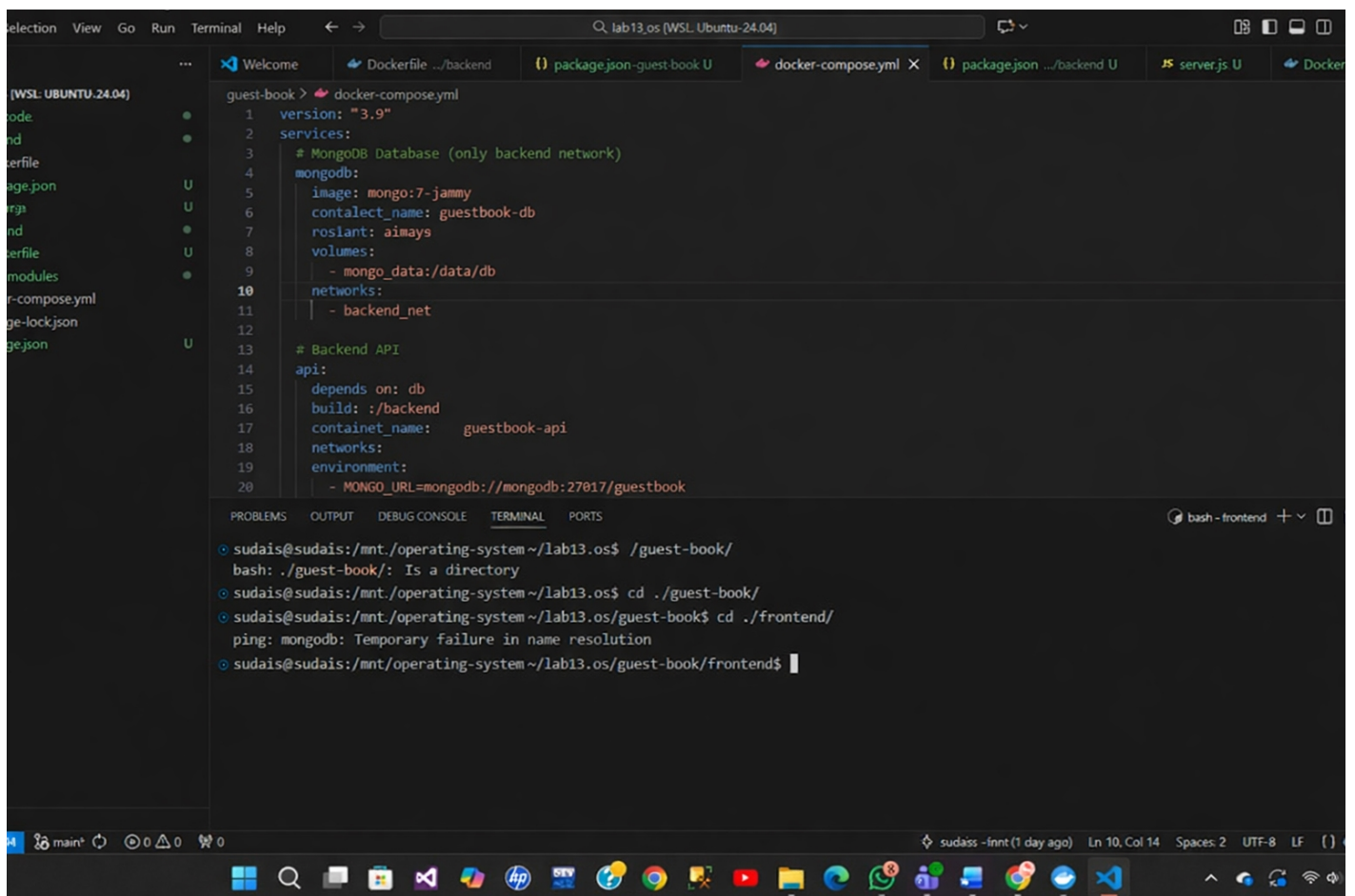
- frontend\_net ( only frontend coonection) networks:

frontend\_net: backend\_net: volumes:

mongo\_data: ( volume in docker )

## Task 2:

**The requirement is that frontend must not be able to call the database container.**



```
selection View Go Run Terminal Help
lab13_os (WSL Ubuntu-24.04)

Welcome Dockerfile ../backend package.json-guest-book U docker-compose.yml X package.json ../backend U server.js U Docker

[WSL: UBUNTU.24.04]
code
nd
erfile
age.pon
rja
nd
erfile
modules
r-compose.yml
ge-lockjson
gejson

guest-book > docker-compose.yml
1 version: "3.9"
2 services:
3   # MongoDB Database (only backend network)
4   mongodb:
5     image: mongo:7-jammy
6     container_name: guestbook-db
7     restart: always
8     volumes:
9       - mongo_data:/data/db
10    networks:
11      - backend_net
12
13   # Backend API
14   api:
15     depends_on: db
16     build: ../backend
17     container_name: guestbook-api
18     networks:
19     environment:
20       - MONGO_URL=mongodb://mongodb:27017/guestbook

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - frontend + v

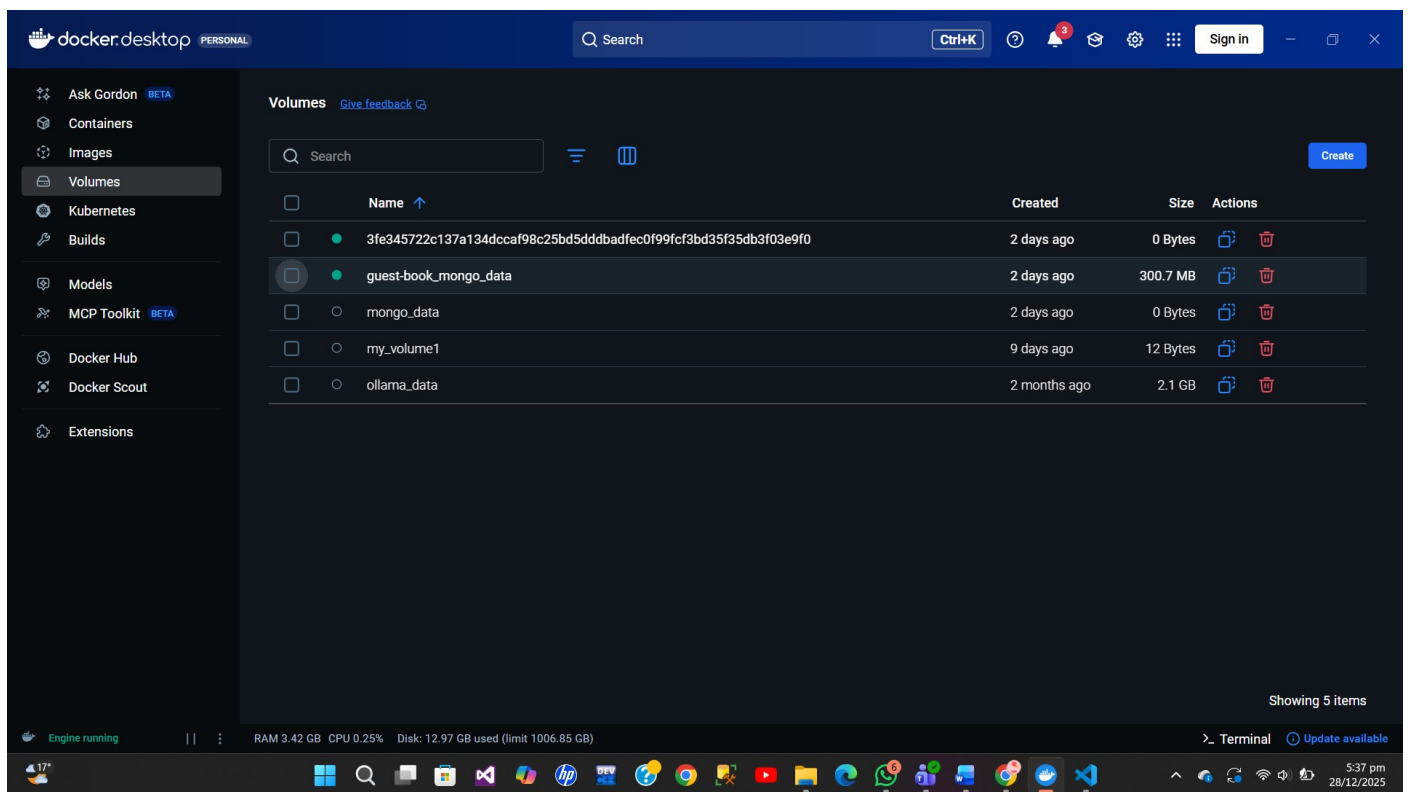
suda@suda:/mnt/operating-system~/lab13.os$ /guest-book/
bash: ./guest-book/: Is a directory
suda@suda:/mnt/operating-system~/lab13.os$ cd ../guest-book/
suda@suda:/mnt/operating-system~/lab13.os/guest-book$ cd ../frontend/
ping: mongodb: Temporary failure in name resolution
suda@suda:/mnt/operating-system~/lab13.os/guest-book/frontend$
```

## Remarks :

As we earlier explain in code that we connect frontend just with backend not with database mongo so it will not ping database .

## Task 3:

**Don't forget to create named volume with the same name you've used in dockercompose.yml**



## Task 4:

**Screenshot of Whole project structure.**

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

LAB13\_OS [WSL: UBUNTU...]

guest-book

backend

Dockerfile

package.json

server.js

frontend

Dockerfile

node\_modules

docker-compose.yml

package-lock.json

package.json

Welcome

Dockerfile .../backend U

guest-book > docker-compose.yml

29 services:

29 web:

33 ports:

33 volumes:

34 - ./frontend:/usr/share/nginx/h

35 depends\_on:

36 - api

37 networks:

38 - frontend\_net

40 networks:

41 frontend\_net:

42 backend\_net:

43 volumes:

44 mongo\_data:

45

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

sudais@sudais:/mnt/operating-system~/lab13

guestbook-db | {"name":"linux","architectu

guestbook-db | { "t":{"\$date":"2025-12-20-2

"client":"172.19.0.3:33872";doc":{"driver":{

e":"x64","version":"v.7.0.8-microsoft-0-too

guestbook-db | {"t":{"\$date":"2025-12-29:27

sage":{"ts\_sec":1766924755,"ts\_usec":238787,

":6,"verbose\_level":"DEBUG\_1","verbose\_level

kpoint timestamp: (0, 0) base write gen: 1697

guestbook-db | {"t":{"\$date":"2025-12-29-27

sage":{"ts\_sec":1766924815,"ts\_usec":238767,

":6,"verbose\_level":"DEBUG\_1","verbose\_level

kpoint timestamp: (0, 0) base write gen: 1697

WSL: Ubuntu-24.04

main\*

0

0

0

View in Docker Desktop

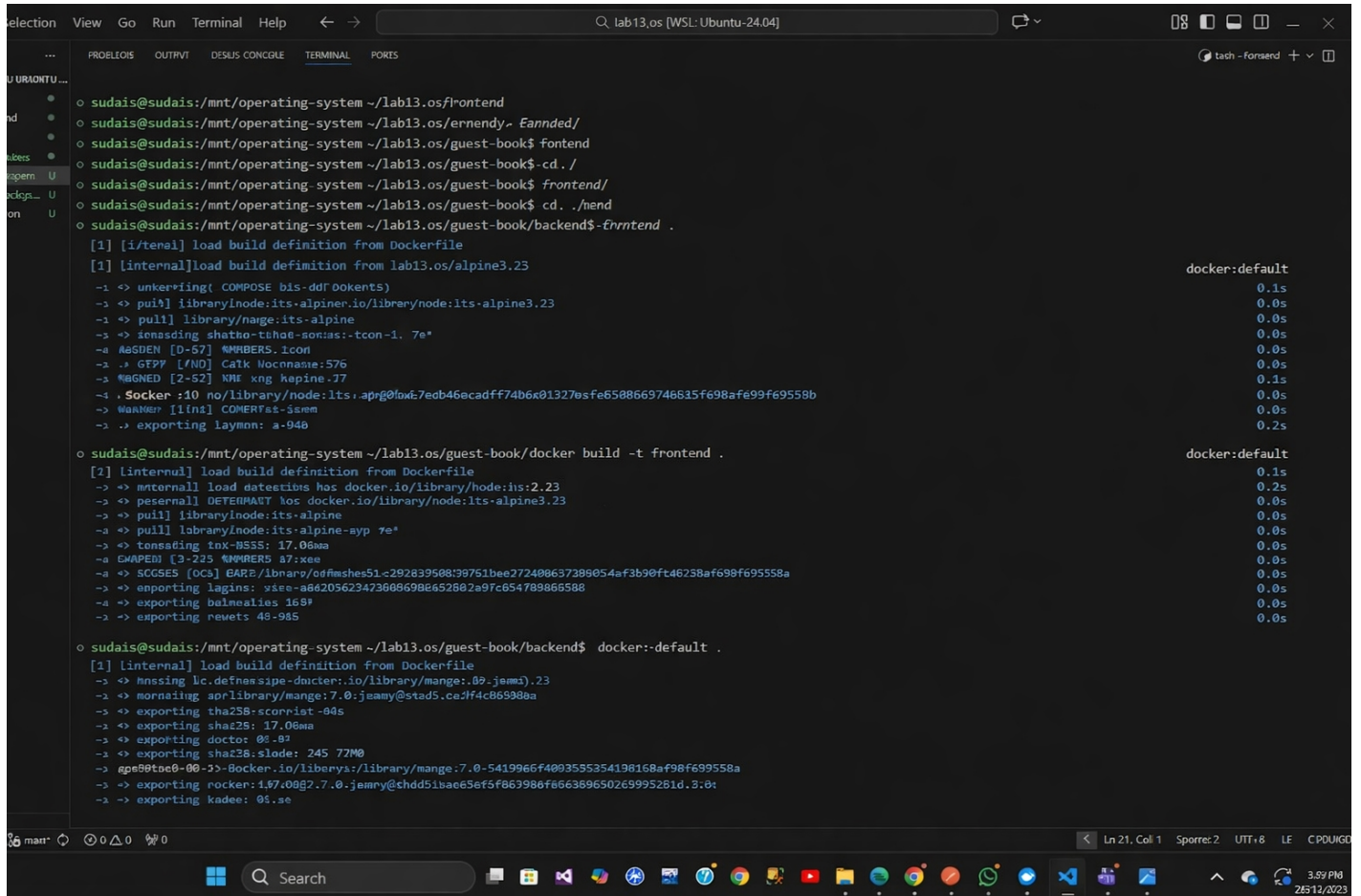
View Config

HT 1

Search

Search

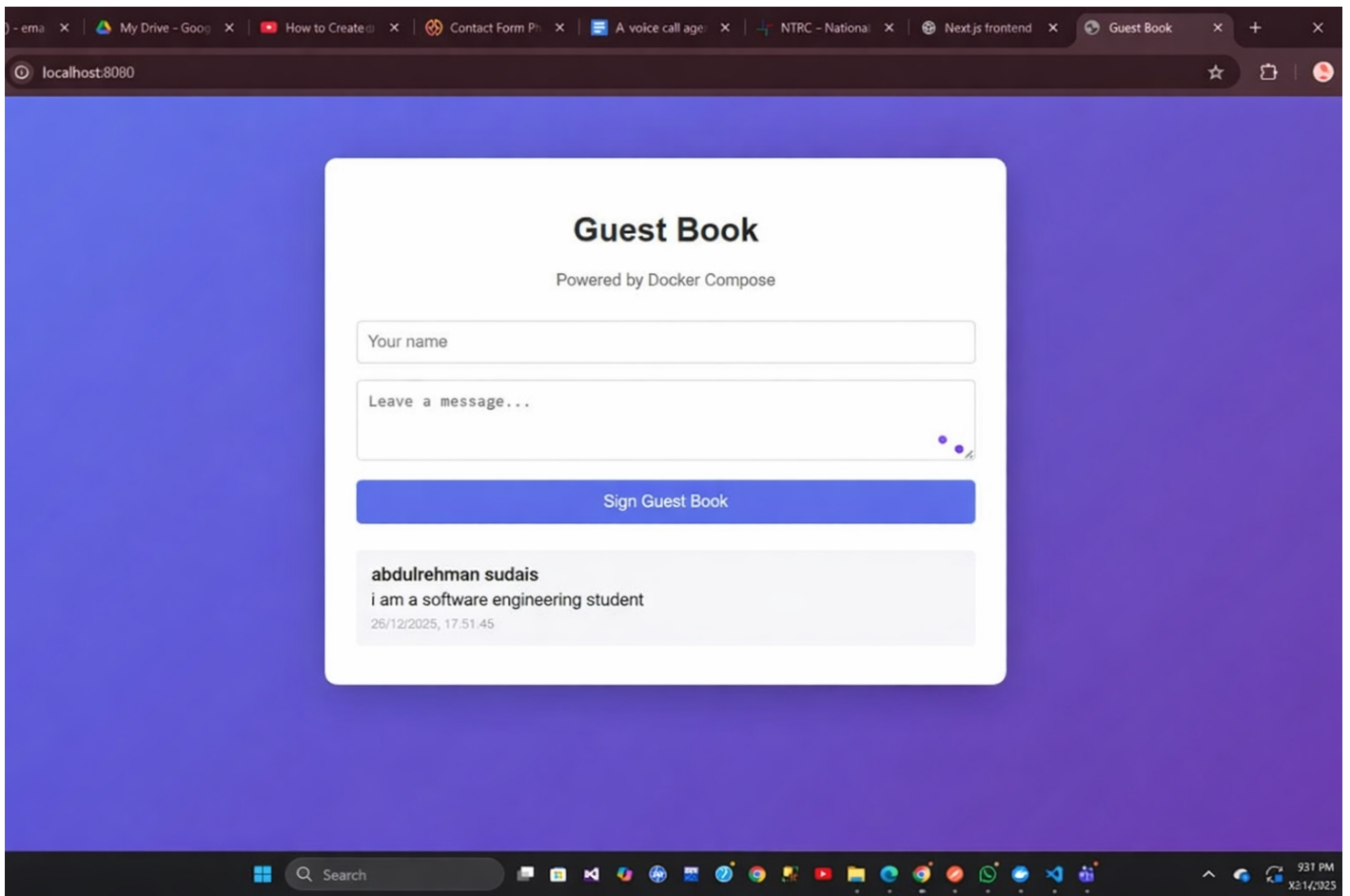
## Output of both image building commands e.g "docker build -t frontend ."



## Task 6 :

**Webpage at localhost:8080**





## Task 7 :

**Copy contents of docker-compose and both Dockerfiles in the pdf.**

**Code of docker- compose file :**

version: "3.9" services:

# MongoDB Database (only backend network) mongodb:

image: mongo:7-jammy

container\_name: guestbook-db

restart: always volumes:

- mongo\_data:/data/db networks: -

backend\_net

# Backend API

api:

build: ./backend image: backend

container\_name: guestbook-api

environment:

- MONGO\_URL=mongodb://mongodb:27017/guestbook ports:
- "3000:3000" depends\_on: - mongodb
- networks: - frontend\_net
- backend\_net

# Frontend web:

image: nginx:alpine

container\_name: guestbook-web

restart: always ports: - "8080:80"

volumes:

- ./frontend:/usr/share/nginx/html:ro

depends\_on:

- api networks: - frontend\_net

networks: frontend\_net: backend\_net:

volumes: mongo\_data:

### Code of dockerfiles :

FROM node:lts-alpine3.23

WORKDIR /app

COPY package\*.json ./

RUN npm install COPY

server.js .

EXPOSE 3000

CMD ["npm", "start"]

### To run the file :

**docker compose up --build** use this command in terminal in guestbook.



