



# 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:

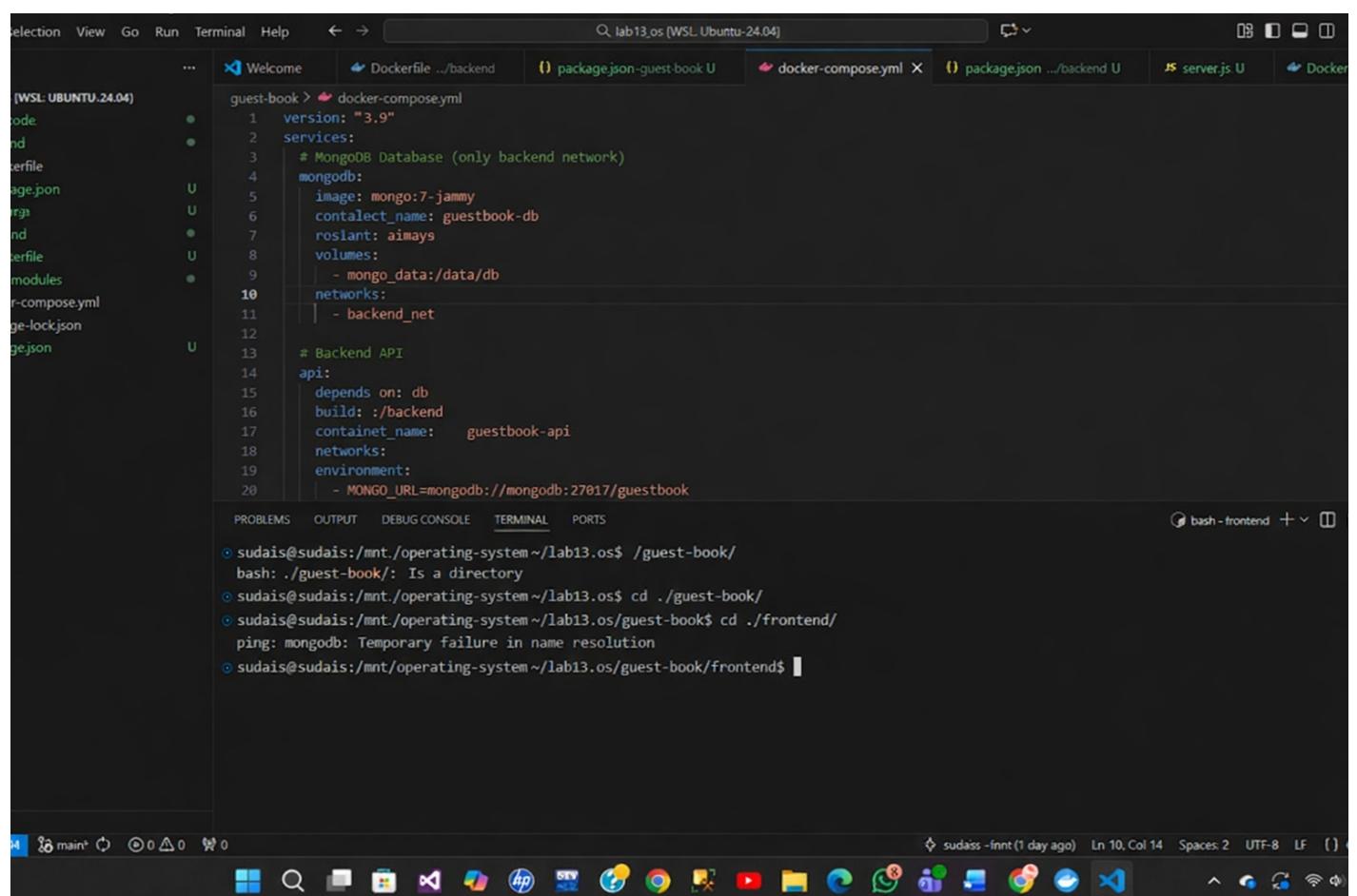
- frontend\_net ( only frontend coonection) networks:

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.**



The screenshot shows a terminal window with several tabs open. The tabs include 'Welcome', 'Dockerfile ..backend', 'package.json-guest-book U', 'docker-compose.yml X', 'package.json .../backend U', 'server.js U', and 'Docker'. The 'docker-compose.yml' tab is currently active, displaying the following YAML configuration:

```
version: "3.9"
services:
  # MongoDB Database (only backend network)
  mongodb:
    image: mongo:7-jammy
    container_name: guestbook-db
    roslnat: aimays
    volumes:
      - mongo_data:/data/db
    networks:
      - backend_net
  # Backend API
  api:
    depends_on: db
    build: ./backend
    container_name: guestbook-api
    networks:
      environment:
        - MONGO_URL=mongodb://mongodb:27017/guestbook
```

The terminal below shows a sequence of commands and their outputs:

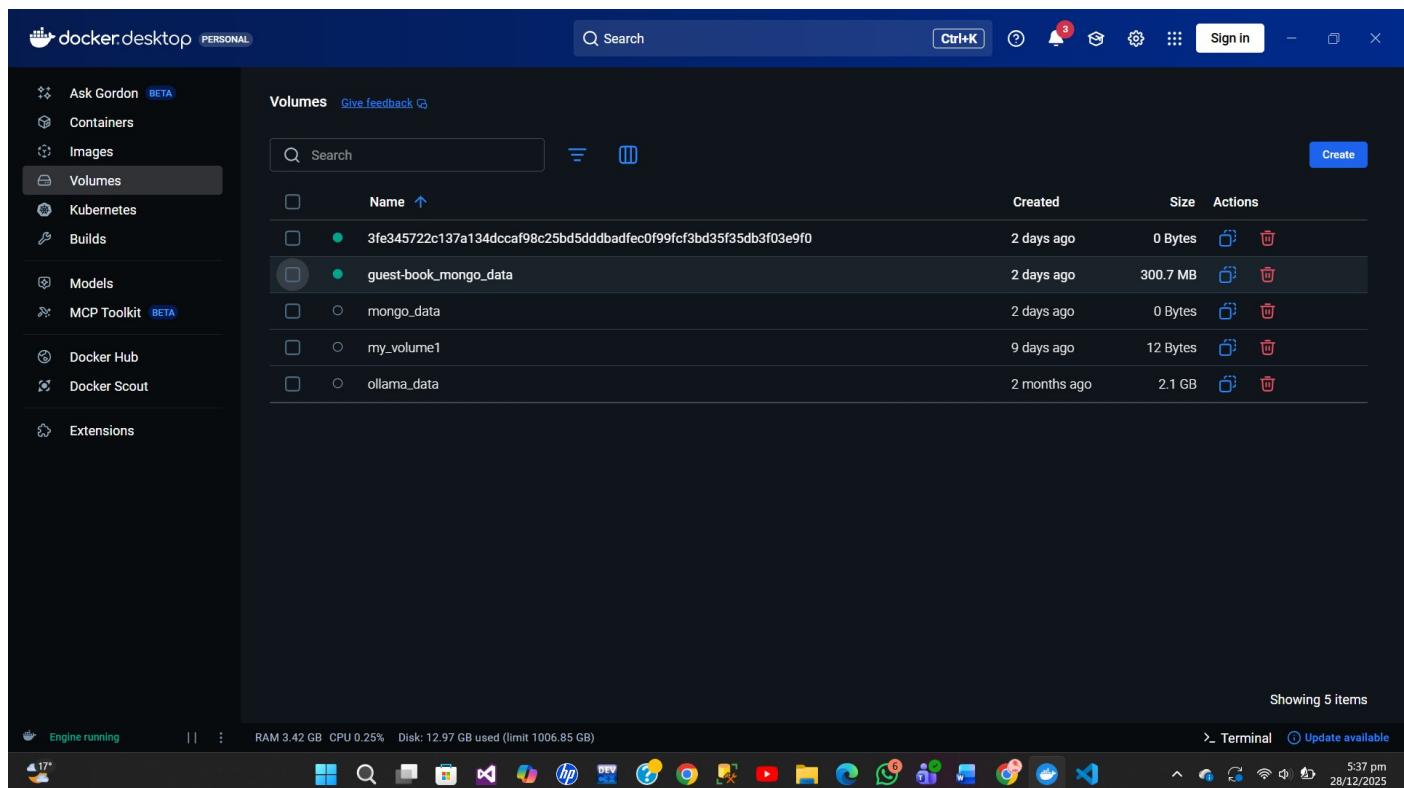
```
sudais@sudais:/mnt/operating-system~/lab13.os$ ./guest-book/
bash: ./guest-book/: Is a directory
sudais@sudais:/mnt/operating-system~/lab13.os$ cd ./guest-book/
sudais@sudais:/mnt/operating-system~/lab13.os/guest-book$ cd ./frontend/
ping: mongodb: Temporary failure in name resolution
sudais@sudais:/mnt/operating-system~/lab13.os/guest-book/frontend$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

## **Remarks :**

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

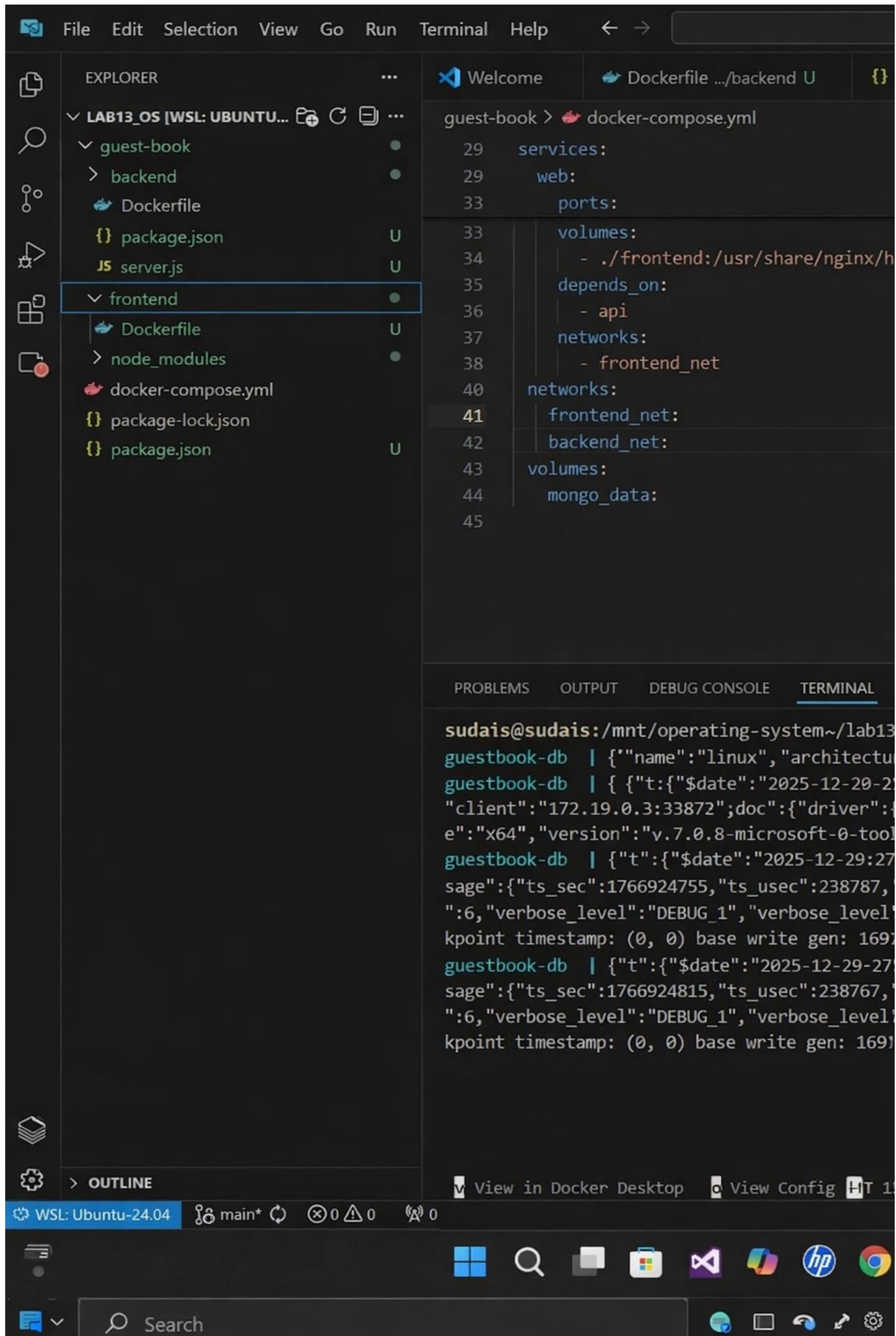
## **Task 3:**

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



## **Task 4:**

**Screenshot of Whole project structure.**



## Task 5:

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

```
selection View Go Run Terminal Help ← → Q lab13.os [WSL:Ubuntu-24.04] ↻ tash - forsend + ×

... PROLEOIS OUTRVI DESUS CONGULE TERMINAL PORTS
UURANTU...
nd
alters
open U
bclays_ U
on U



- sudais@sudais:/mnt/operating-system ~/lab13.os$ frontend
- sudais@sudais:/mnt/operating-system ~/lab13.os$ ernendy_ Eannedy/
- sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book$ frontend
- sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book$ cd . /
- sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book$ frontend/
- sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book$ cd .. ./hend
- sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book/backend$ -frntend .



```
[1] [internal] load build definition from Dockerfile
[1] [internal] load build definition from lab13.os/alpine3.23
-1 => unkerfing( COMPOSE bis-dfl Dokents)
-1 => puwl librarynode:its-alpine.io/library/node:lts-alpine3.23
-1 => pull library/nage:its-alpine
-1 => tensading shatto-tthd-sontas:tcon-1. 7e
-1 => AaSDEN [D-57] MMBERS.icon
-1 => GEPY [NNO] Catk Nocnname:576
-1 => BGNED [2-52] KME kng kapeine:77
-1 => Socker :10 no/library/node:lts-apr0fowx7edb46ecadff74b6x01327osfe6508669746635f698afe99f69558b
-1 => Wnther [1[n]l COMFET-assem
-1 => exporting laymnn: a-948

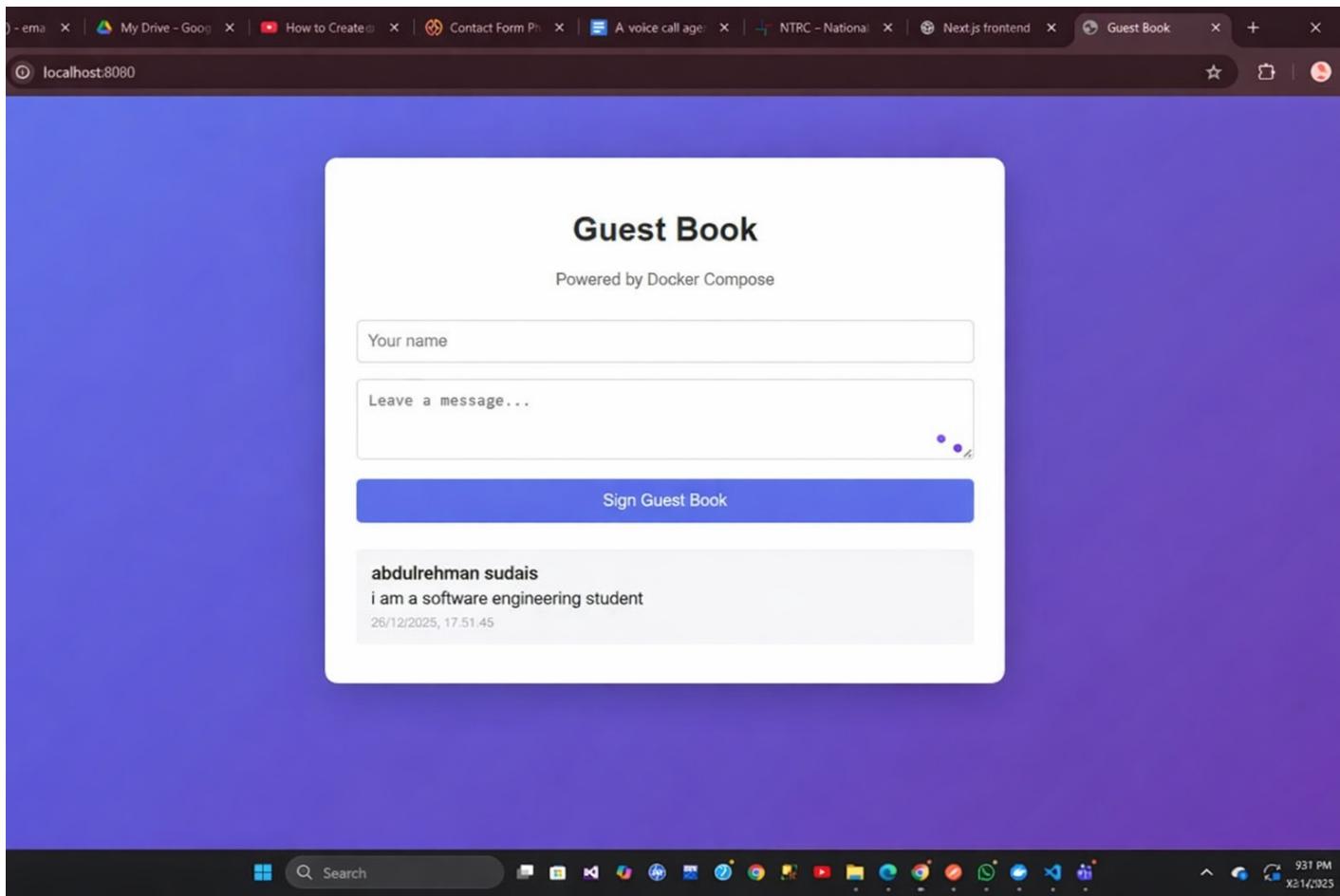
○ sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book/docker build -t frontend .
[1] [internal] load build definition from Dockerfile
-1 => minternal load detectives has docker.io/library/node:is:2.23
-1 => pesermal DETERMANT has docker.io/library/node:lts-alpine3.23
-1 => puwl librarynode:its-alpine
-1 => pull library/node:its-alpine-syv 7e*
-1 => tensading fx-BSSS: 17.06wa
-1 => SWAPEO [3-22] MMBERS.7x:ee
-1 => SCGSEE [0c8] CAR2/1library/dfmshes51-c292839508:39751bee272408637389054af3b90ft46258af698f69558a
-1 => exporting lagins: vixe-a662056234736086982652882a97c65478986588
-1 => exporting balmeilies 168*
-1 => exporting rewets 48-985

○ sudais@sudais:/mnt/operating-system ~/lab13.os/guest-book/backend$ docker::default .
[1] [internal] load build definition from Dockerfile
-1 => mssing lc.defensipsie-ducter:io/library/mange:0.9.jemmi.23
-1 => mornaing sorlibrary/mange:7.0:jemmy@stad5.cedf4c86598ba
-1 => exporting tha2SB-scorrist -0ds
-1 => exporting sha25: 17.08ma
-1 => exporting docto: 05.89
-1 => exporting sha28.slode: 245 72M0
-1 => gpe@9tne0-00-00-00-00-00: Rocker.io/library/mange:7.0-5419966f409355354198168af98f699558a
-1 => exporting rocker:1.67.00g2.7.0.jemmy@hdd5ikae65ef5f863986f866389650269995281d.3.0t
-1 => exporting kadee: 05.se
```


```

## 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:2701
```

```
  7/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.
```

