National Textile University

Department of Computer Science

Subject:

Operating Systems

Submitted To:

Sir Nasir Mehmood

Submitted By:

Alishba Riasat

Registration No:

23-NTU-CS-1135

Lab No:

Docker Hometasks

Semester:

5th

Complete the docker compose file with the following requirements:

1. Create networks to facilitate communication between frontend, backend and database.
2. The requirement is that frontend must not be able to call the database container.
3. Don't forget to create named volume with the same name you've used in docker-compose.yml

1. Screenshot of the Whole project structure:
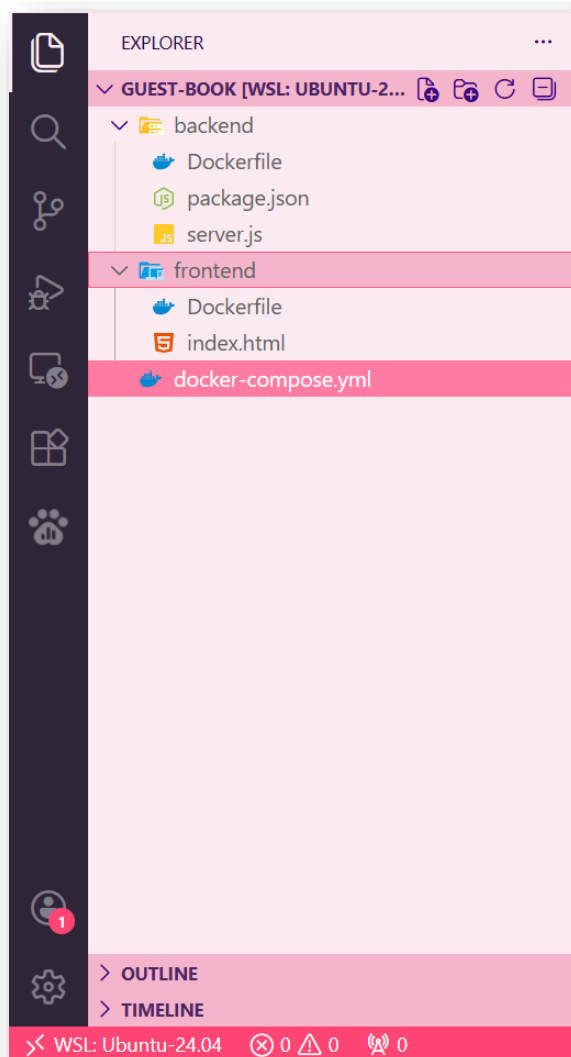
2. Output of both image building commands e.g., "docker build -t frontend."

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                              bash - guest-book  + ∨  ⬚  🗑  ⋯  ⬚ ⌄ ✕

● alishba@DESKTOP-S3K0VA5:~/guest-book$ docker build -t guestbook-backend ./backend
  [+] Building 1.3s (10/10) FINISHED                                                    docker:default
   => [internal] load build definition from Dockerfile                                            0.0s
   => => transferring dockerfile: 162B                                                            0.0s
   => [internal] load metadata for docker.io/library/node:lts-alpine3.23                          0.5s
   => [internal] load .dockerignore                                                               0.0s
   => => transferring context: 2B                                                                 0.0s
   => [1/5] FROM docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf693865e73b81005591896  0.1s
   => => resolve docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf693865e73b81005591896  0.1s
   => [internal] load build context                                                               0.0s
   => => transferring context: 63B                                                                0.0s
   => CACHED [2/5] WORKDIR /app                                                                    0.0s
   => CACHED [3/5] COPY package*.json ./                                                           0.0s
   => CACHED [4/5] RUN npm install                                                                 0.0s
   => CACHED [5/5] COPY server.js .                                                                0.0s
   => exporting to image                                                                           0.3s
   => => exporting layers                                                                          0.0s
   => => exporting manifest sha256:9683b07bf9a4ca2c2520f0921593fa0ed2356892cbf5590fe72858d5bfe44fda  0.0s
   => => exporting config sha256:a643d577d2c006d369e2405f6a1ddde29ccddfb4c64094ae6e6a22fea05849e2  0.0s
   => => exporting attestation manifest sha256:f0c45e3a1ef75a79a945744c7e4f16b112325b4484da153f50c94ee3053a9445  0.1s
   => => exporting manifest list sha256:2d1af10e52dd932c2a21899ce8c9a25aa80b48455905782119a7010f92ee85e2  0.0s
   => => naming to docker.io/library/guestbook-backend:latest                                     0.0s
   => => unpacking to docker.io/library/guestbook-backend:latest                                  0.0s
● alishba@DESKTOP-S3K0VA5:~/guest-book$ docker build -t guestbook-frontend ./frontend
  [+] Building 0.6s (7/7) FINISHED                                                     docker:default
   => [internal] load build definition from Dockerfile                                            0.0s
   => => transferring dockerfile: 103B                                                            0.0s
   => [internal] load metadata for docker.io/library/nginx:trixie                                 0.1s
   => [internal] load .dockerignore                                                               0.0s
```

```
   => => transferring context: 2B                                                                 0.0s
   => [internal] load build context                                                               0.0s
   => => transferring context: 32B                                                                0.0s
   => [1/2] FROM docker.io/library/nginx:trixie@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d  0.1s
   => => resolve docker.io/library/nginx:trixie@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d  0.0s
   => CACHED [2/2] COPY index.html /usr/share/nginx/html/                                          0.0s
   => exporting to image                                                                           0.2s
   => => exporting layers                                                                          0.0s
   => => exporting manifest sha256:a67cdc5ea31379211699bab978dba4c8744112091e2bc919323aaeb62641783d  0.0s
   => => exporting config sha256:8b89f4100eb00083099c921978c696c1a99b39d4f7789e815854446991879820  0.0s
   => => exporting attestation manifest sha256:a09aeaf49d4e5b43c92f5e76959a15fe702b71eee16ee4f295c725709481972a  0.1s
   => => exporting manifest list sha256:0b3030d24fac7c9c1bc934dee7f604455e28f3e945f1aa0bf7b787170b87dccc  0.0s
   => => naming to docker.io/library/guestbook-frontend:latest                                    0.0s
   => => unpacking to docker.io/library/guestbook-frontend:latest                                 0.0s
○ alishba@DESKTOP-S3K0VA5:~/guest-book$ ▮
```
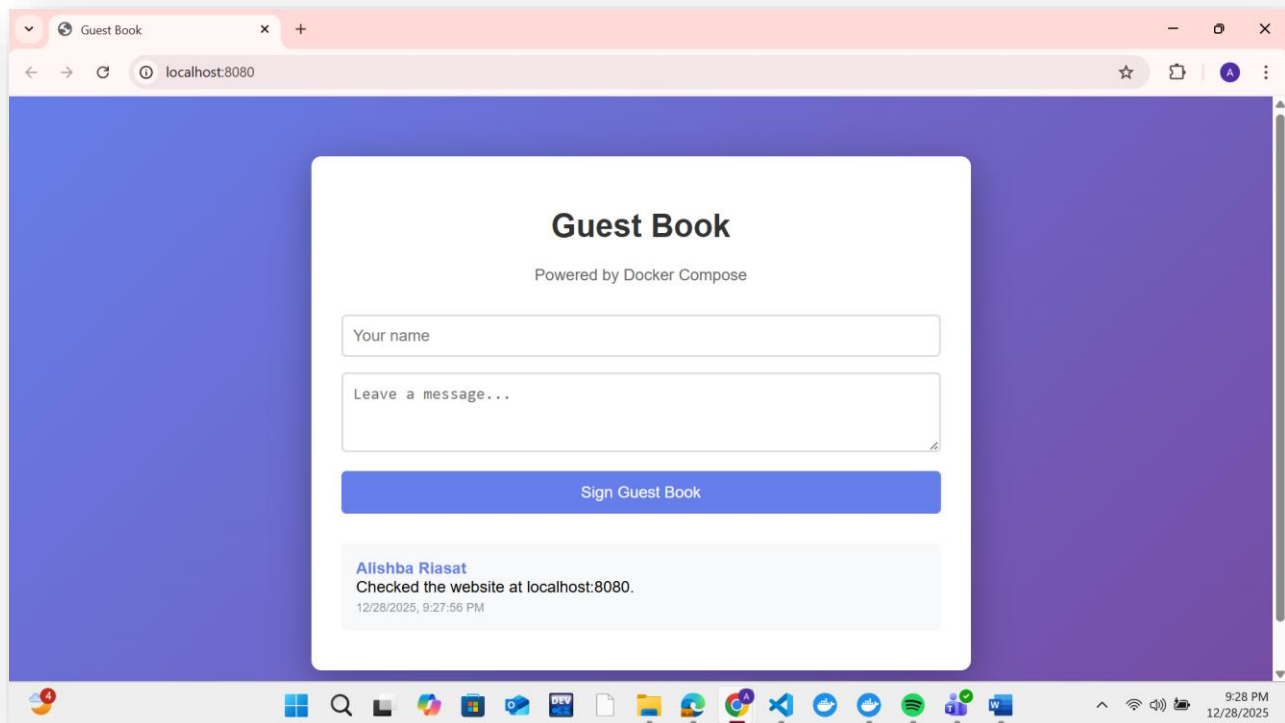
Docker containers running:

3. **Webpage at localhost:8080**

Frontend cannot access database:

```
alishba@DESKTOP-S3K0VA5:~/guest-book$ docker compose exec web ping mongodb
WARN[0000] /home/alishba/guest-book/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please
 remove it to avoid potential confusion
OCI runtime exec failed: exec failed: unable to start container process: exec: "ping": executable file not found in $PA
TH
```

Backend can access database successfully:

```
alishba@DESKTOP-S3K0VA5:~/guest-book$ docker compose exec api ping mongodb
WARN[0000] /home/alishba/guest-book/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pleas
 remove it to avoid potential confusion
PING mongodb (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: seq=0 ttl=64 time=1.634 ms
64 bytes from 172.18.0.2: seq=1 ttl=64 time=0.072 ms
64 bytes from 172.18.0.2: seq=2 ttl=64 time=0.072 ms
64 bytes from 172.18.0.2: seq=3 ttl=64 time=0.155 ms
64 bytes from 172.18.0.2: seq=4 ttl=64 time=0.071 ms
64 bytes from 172.18.0.2: seq=5 ttl=64 time=0.079 ms
64 bytes from 172.18.0.2: seq=6 ttl=64 time=0.088 ms
64 bytes from 172.18.0.2: seq=7 ttl=64 time=0.067 ms
64 bytes from 172.18.0.2: seq=8 ttl=64 time=0.121 ms
64 bytes from 172.18.0.2: seq=9 ttl=64 time=0.115 ms
64 bytes from 172.18.0.2: seq=10 ttl=64 time=0.181 ms
64 bytes from 172.18.0.2: seq=11 ttl=64 time=0.245 ms
64 bytes from 172.18.0.2: seq=12 ttl=64 time=0.094 ms
64 bytes from 172.18.0.2: seq=13 ttl=64 time=0.100 ms
64 bytes from 172.18.0.2: seq=14 ttl=64 time=0.250 ms
64 bytes from 172.18.0.2: seq=15 ttl=64 time=0.071 ms
64 bytes from 172.18.0.2: seq=16 ttl=64 time=0.089 ms
64 bytes from 172.18.0.2: seq=17 ttl=64 time=0.093 ms
64 bytes from 172.18.0.2: seq=18 ttl=64 time=0.188 ms
64 bytes from 172.18.0.2: seq=19 ttl=64 time=0.069 ms
64 bytes from 172.18.0.2: seq=20 ttl=64 time=0.102 ms
64 bytes from 172.18.0.2: seq=21 ttl=64 time=0.084 ms
```

Backend API Response:

```
^C
--- mongodb ping statistics ---
56 packets transmitted, 56 packets received, 0% packet loss
round-trip min/avg/max = 0.061/0.138/1.634 ms
alishba@DESKTOP-S3K0VA5:~/guest-book$ curl http://localhost:3000/api/messages
[{"_id":"69515a8cac0a4989b74ba013","name":"Alishba Riasat","message":"Checked the website at localhost:8080.","timestam
```

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

<u>docker-compose.yml:</u>

```yaml
version: "3.9"

services:
  # MongoDB Database
  mongodb:
    image: mongo:7-jammy
    container_name: guestbook-db
    networks:
      - backend-db-net
    volumes:
      - mongo_data:/data/db

  # Backend API
  api:
    # build: ./backend
    image: guestbook-backend
    container_name: guestbook-api
    environment:
      - MONGO_URL=mongodb://mongodb:27017
    ports:
      - "3000:3000"
    depends_on:
      - mongodb
    networks:
      - frontend-backend-net
      - backend-db-net

  # Frontend
  web:
    # build: ./frontend
    image: guestbook-frontend
    container_name: guestbook-web
    ports:
      - "8080:80"
    depends_on:
      - api
    networks:
      - frontend-backend-net

# Networks
networks:
```

```yaml
  frontend-backend-net:
  backend-db-net:

# Volumes
volumes:
  mongo_data:
```

Dockerfile of frontend:

```dockerfile
FROM nginx:trixie
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

Dockerfile of backend:

```dockerfile
FROM node:lts-alpine3.23
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY server.js .
EXPOSE 3000
CMD ["npm", "start"]
```