National Textile University

Department of Computer Science

Subject: Operating System

Submitted to: Nasir Mahmood

Submitted by: Esha Mubashir Khan

Reg number:23-NTU-CS-1151

Lab : 14

Semester:5th

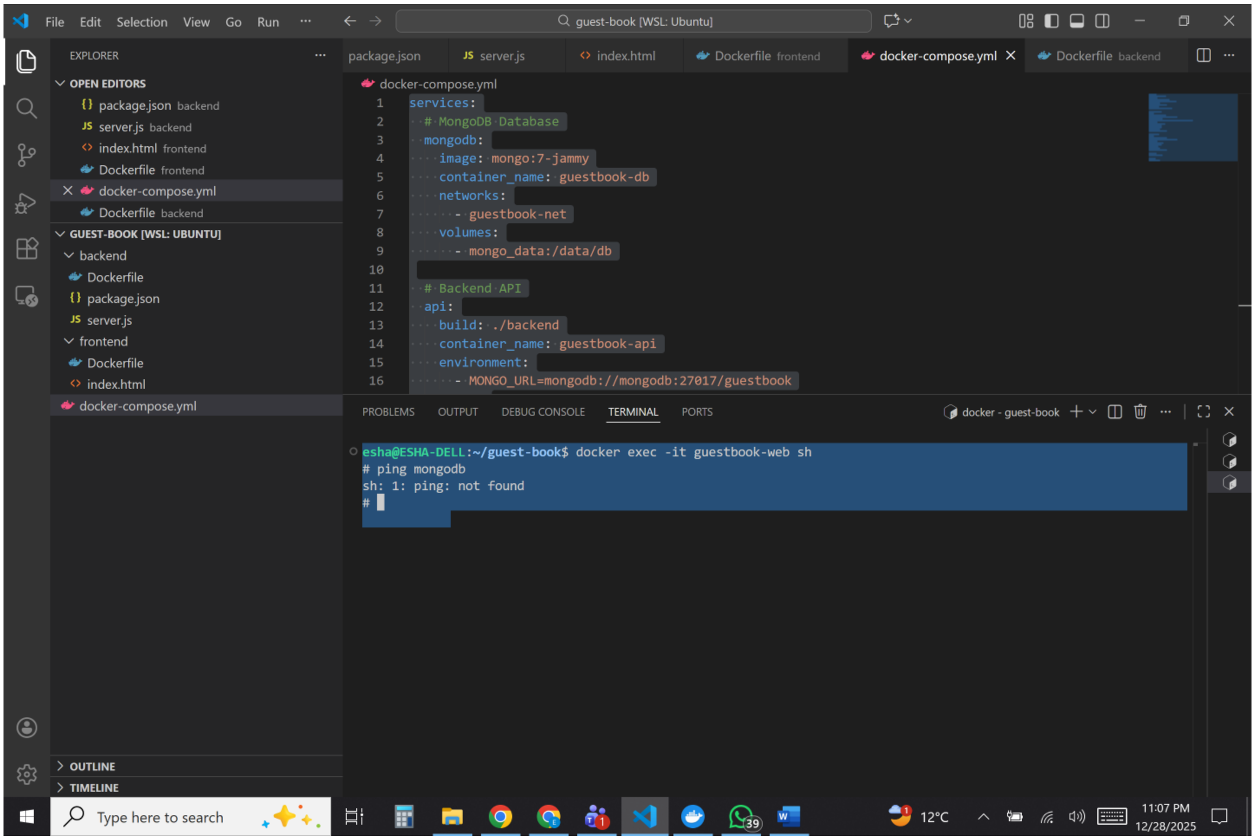
TASK 1:

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

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

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

networks:
  guestbook-net:

volumes:
  mongo_data:
```

EXPLANATION:

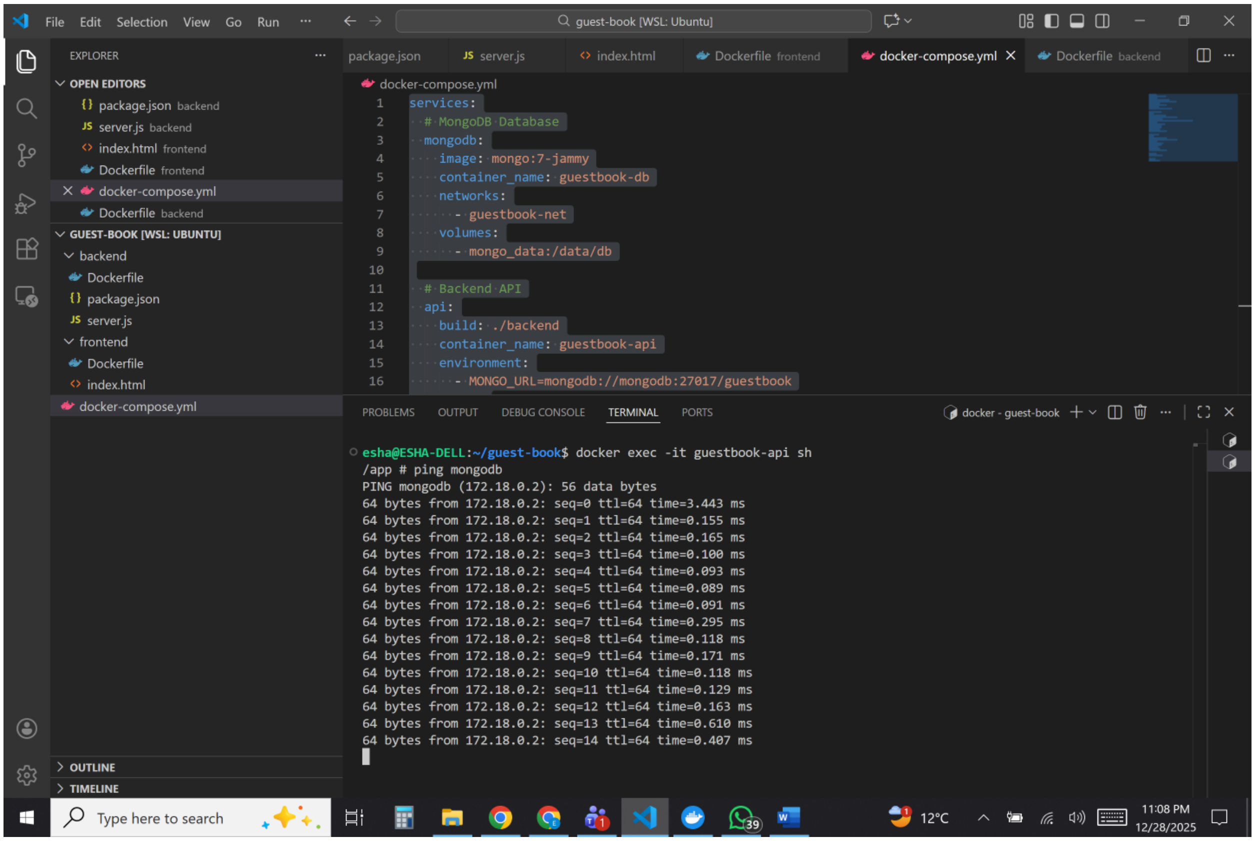
A custom Docker bridge network allows frontend, backend, and database containers to communicate securely using service names instead of IP addresses.

TASK 2:



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

  # Backend API
  api:
    build: ./backend
    container_name: guestbook-api
    environment:
      - MONGO_URL=mongodb://mongodb:27017/guestbook
```

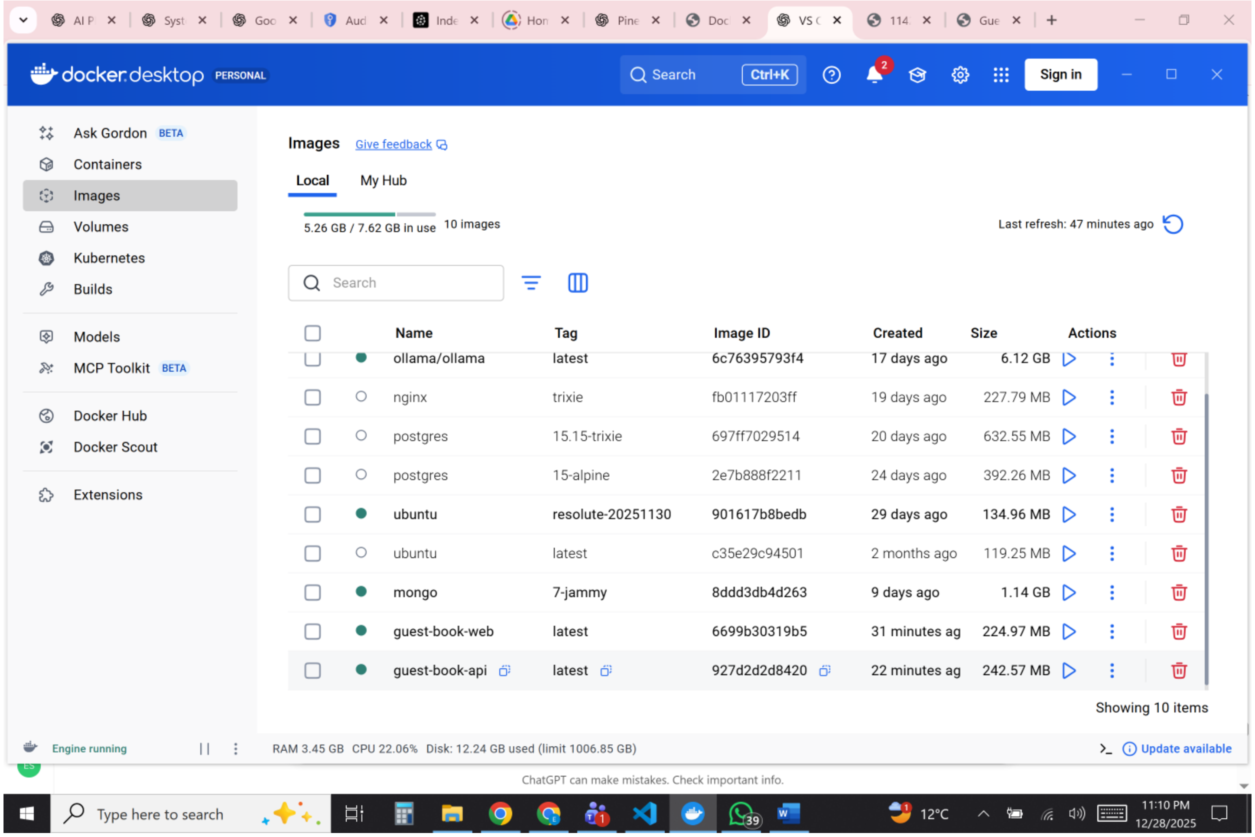
```
esha@ESHA-DELL:~/guest-book$ docker exec -it guestbook-web sh
# ping mongodb
sh: 1: ping: not found
# 
```



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

  # Backend API
  api:
    build: ./backend
    container_name: guestbook-api
    environment:
      - MONGO_URL=mongodb://mongodb:27017/guestbook
```

```
esha@ESHA-DELL:~/guest-book$ docker exec -it guestbook-api sh
/app # ping mongodb
PING mongodb (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: seq=0 ttl=64 time=3.443 ms
64 bytes from 172.18.0.2: seq=1 ttl=64 time=0.155 ms
64 bytes from 172.18.0.2: seq=2 ttl=64 time=0.165 ms
64 bytes from 172.18.0.2: seq=3 ttl=64 time=0.100 ms
64 bytes from 172.18.0.2: seq=4 ttl=64 time=0.093 ms
64 bytes from 172.18.0.2: seq=5 ttl=64 time=0.089 ms
64 bytes from 172.18.0.2: seq=6 ttl=64 time=0.091 ms
64 bytes from 172.18.0.2: seq=7 ttl=64 time=0.295 ms
64 bytes from 172.18.0.2: seq=8 ttl=64 time=0.118 ms
64 bytes from 172.18.0.2: seq=9 ttl=64 time=0.171 ms
64 bytes from 172.18.0.2: seq=10 ttl=64 time=0.118 ms
64 bytes from 172.18.0.2: seq=11 ttl=64 time=0.129 ms
64 bytes from 172.18.0.2: seq=12 ttl=64 time=0.163 ms
64 bytes from 172.18.0.2: seq=13 ttl=64 time=0.610 ms
64 bytes from 172.18.0.2: seq=14 ttl=64 time=0.407 ms
```

TASK 3:



Images

5.26 GB / 7.62 GB in use 10 images

Last refresh: 47 minutes ago

Name	Tag	Image ID	Created	Size
● ollama/ollama	latest	6c76395793f4	17 days ago	6.12 GB
○ nginx	trixie	fb01117203ff	19 days ago	227.79 MB
○ postgres	15.15-trixie	697ff7029514	20 days ago	632.55 MB
○ postgres	15-alpine	2e7b888f2211	24 days ago	392.26 MB
● ubuntu	resolute-20251130	901617b8bedb	29 days ago	134.96 MB
○ ubuntu	latest	c35e29c94501	2 months ago	119.25 MB
● mongo	7-jammy	8ddd3db4d263	9 days ago	1.14 GB
● guest-book-web	latest	6699b30319b5	31 minutes ag	224.97 MB
● guest-book-api	latest	927d2d2d8420	22 minutes ag	242.57 MB

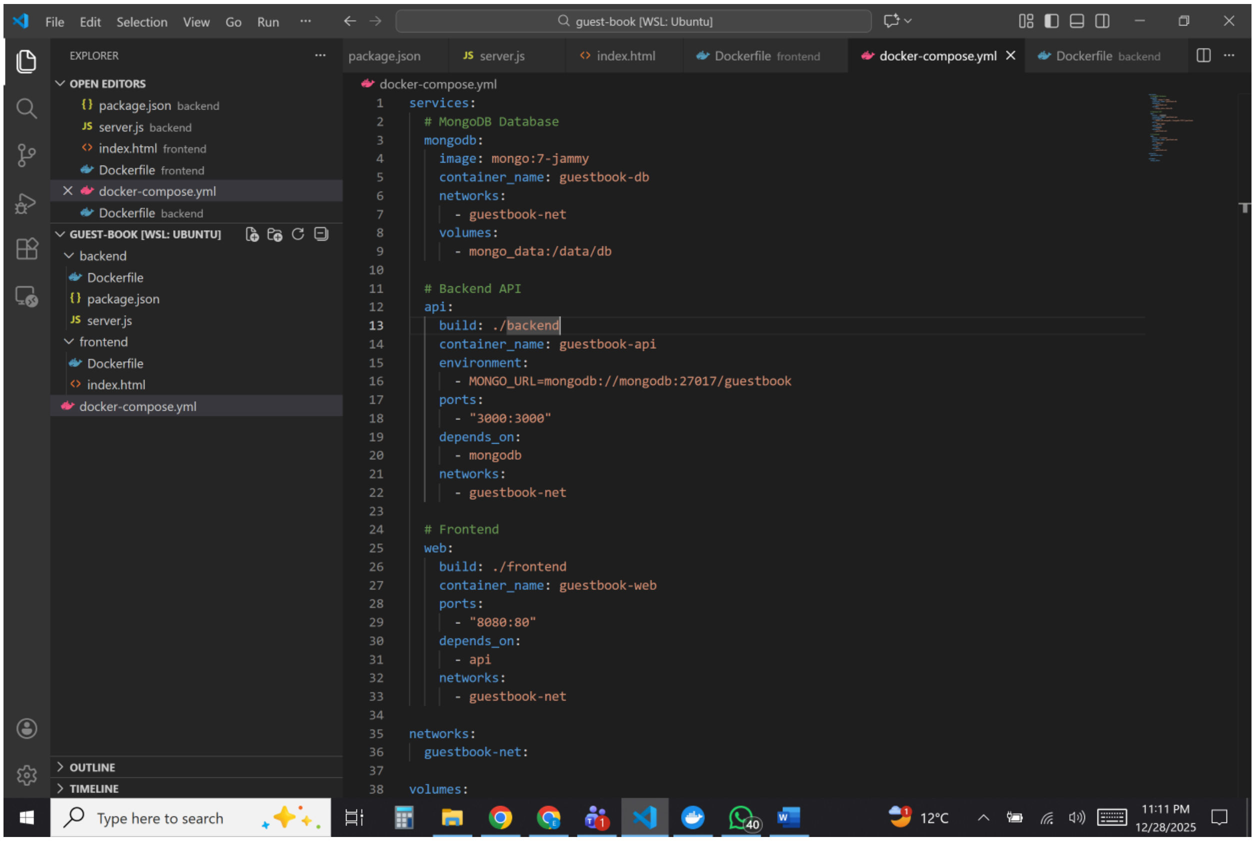
Showing 10 items

RAM 3.45 GB CPU 22.06% Disk: 12.24 GB used (limit 1006.85 GB)

ChatGPT can make mistakes. Check important info.

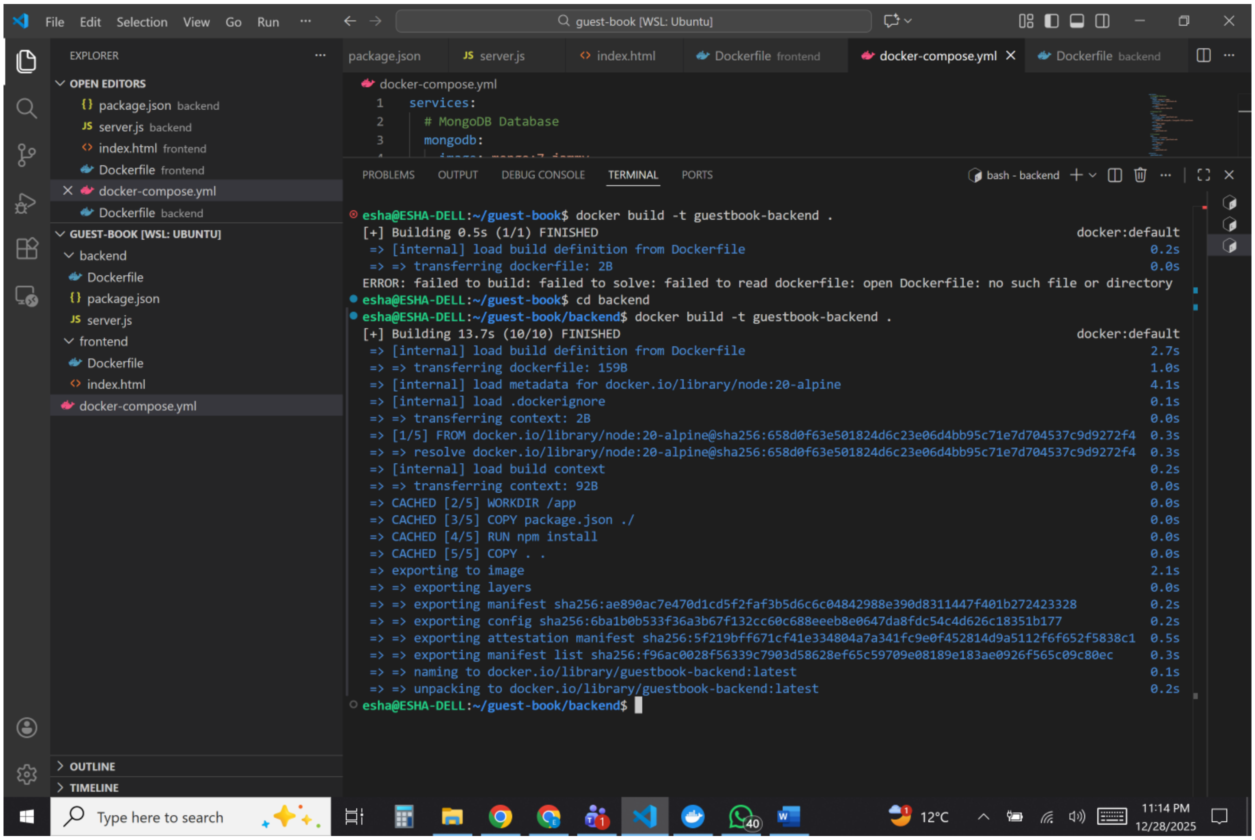
Type here to search 12°C 11:10 PM 12/28/2025

TASK 4:screenshot of whole project



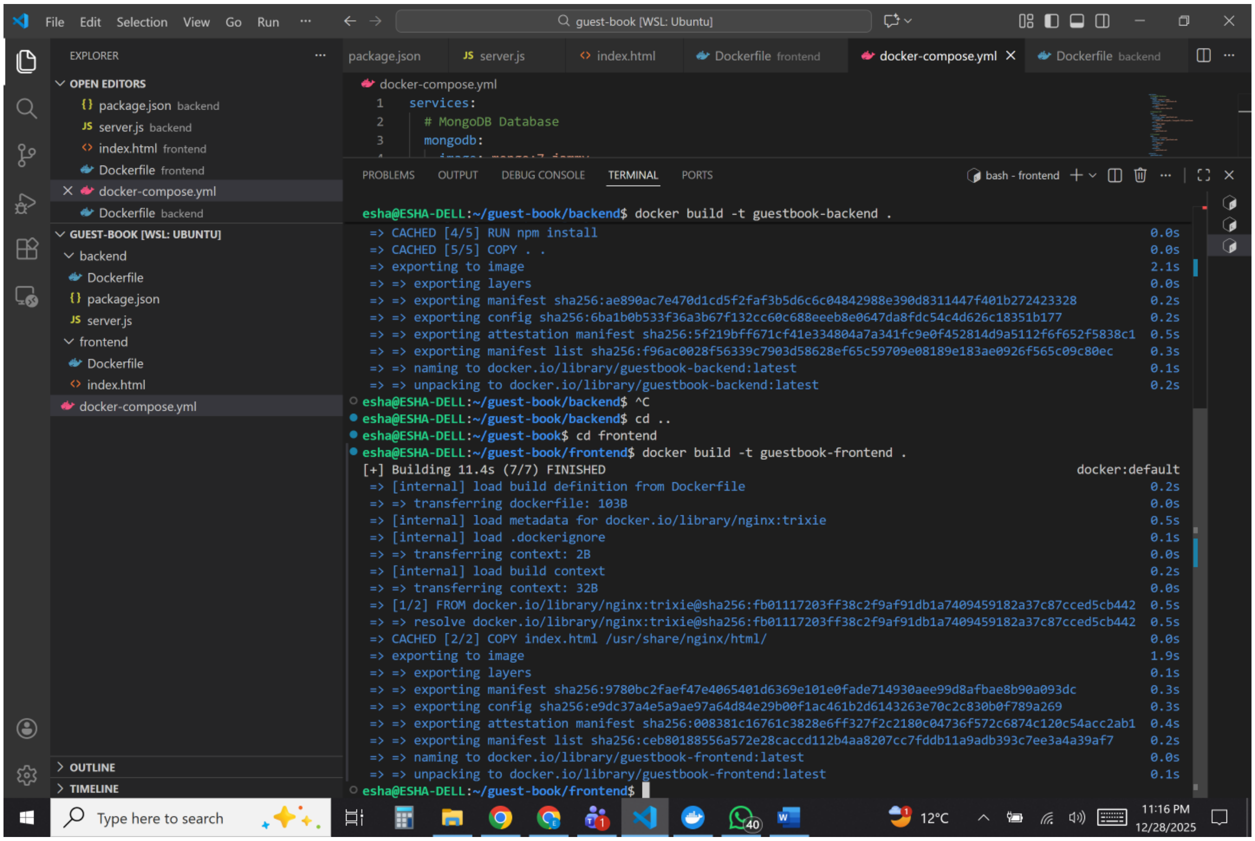
```yaml
services:
  # MongoDB Database
  mongodb:
    image: mongo:7-jammy
    container_name: guestbook-db
    networks:
      - guestbook-net
    volumes:
      - mongo_data:/data/db

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

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

networks:
  guestbook-net:

volumes:
```

TASK 5: Output of both image building commands e.g "docker build -t frontend



```
services:
  # MongoDB Database
  mongodb:
```

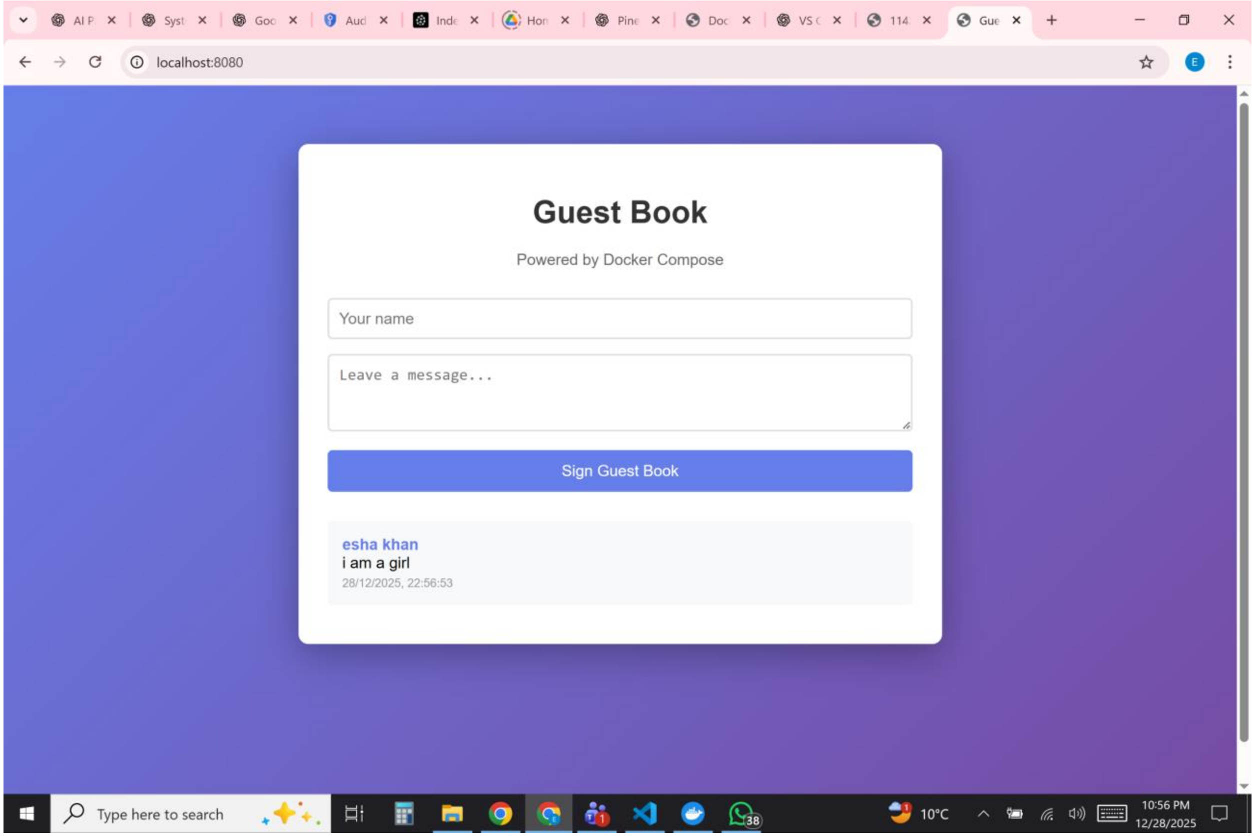
```
esha@ESHA-DELL:~/guest-book$ docker build -t guestbook-backend .
[+] Building 0.5s (1/1) FINISHED                                                       docker:default
 => [internal] load build definition from Dockerfile                                            0.2s
 => => transferring dockerfile: 2B                                                              0.0s
ERROR: failed to build: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
esha@ESHA-DELL:~/guest-book$ cd backend
esha@ESHA-DELL:~/guest-book/backend$ docker build -t guestbook-backend .
[+] Building 13.7s (10/10) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                            2.7s
 => => transferring dockerfile: 159B                                                            1.0s
 => [internal] load metadata for docker.io/library/node:20-alpine                               4.1s
 => [internal] load .dockerignore                                                               0.1s
 => => transferring context: 2B                                                                 0.0s
 => [1/5] FROM docker.io/library/node:20-alpine@sha256:658d0f63e501824d6c23e06d4bb95c71e7d704537c9d9272f4  0.3s
 => => resolve docker.io/library/node:20-alpine@sha256:658d0f63e501824d6c23e06d4bb95c71e7d704537c9d9272f4  0.3s
 => [internal] load build context                                                               0.2s
 => => transferring context: 92B                                                                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 . .                                                                       0.0s
 => exporting to image                                                                          2.1s
 => => exporting layers                                                                         0.0s
 => => exporting manifest sha256:ae890ac7e470d1cd5f2faf3b5d6c6c04842988e390d8311447f401b272423328      0.2s
 => => exporting config sha256:6ba1b0b533f36a3b67f132cc60c688eeeb8e0647da8fdc54c4d626c18351b177        0.2s
 => => exporting attestation manifest sha256:5f219bff671cf41e334804a7a341fc9e0f452814d9a5112f6f652f5838c1  0.5s
 => => exporting manifest list sha256:f96ac0028f56339c7903d58628ef65c59709e08189e183ae0926f565c09c80ec   0.3s
 => => naming to docker.io/library/guestbook-backend:latest                                     0.1s
 => => unpacking to docker.io/library/guestbook-backend:latest                                  0.2s
esha@ESHA-DELL:~/guest-book/backend$
```



```
services:
  # MongoDB Database
  mongodb:
```

```
esha@ESHA-DELL:~/guest-book/backend$ docker build -t guestbook-backend .
 => CACHED [4/5] RUN npm install                                                                0.0s
 => CACHED [5/5] COPY . .                                                                       0.0s
 => exporting to image                                                                          2.1s
 => => exporting layers                                                                         0.0s
 => => exporting manifest sha256:ae890ac7e470d1cd5f2faf3b5d6c6c04842988e390d8311447f401b272423328      0.2s
 => => exporting config sha256:6ba1b0b533f36a3b67f132cc60c688eeeb8e0647da8fdc54c4d626c18351b177        0.2s
 => => exporting attestation manifest sha256:5f219bff671cf41e334804a7a341fc9e0f452814d9a5112f6f652f5838c1  0.5s
 => => exporting manifest list sha256:f96ac0028f56339c7903d58628ef65c59709e08189e183ae0926f565c09c80ec   0.3s
 => => naming to docker.io/library/guestbook-backend:latest                                     0.1s
 => => unpacking to docker.io/library/guestbook-backend:latest                                  0.2s
esha@ESHA-DELL:~/guest-book/backend$ ^C
esha@ESHA-DELL:~/guest-book/backend$ cd ..
esha@ESHA-DELL:~/guest-book$ cd frontend
esha@ESHA-DELL:~/guest-book/frontend$ docker build -t guestbook-frontend .
[+] Building 11.4s (7/7) FINISHED                                                      docker:default
 => [internal] load build definition from Dockerfile                                            0.2s
 => => transferring dockerfile: 103B                                                            0.0s
 => [internal] load metadata for docker.io/library/nginx:trixie                                 0.5s
 => [internal] load .dockerignore                                                               0.1s
 => => transferring context: 2B                                                                 0.0s
 => [internal] load build context                                                               0.2s
 => => transferring context: 32B                                                                0.0s
 => [1/2] FROM docker.io/library/nginx:trixie@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442  0.5s
 => => resolve docker.io/library/nginx:trixie@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442  0.5s
 => CACHED [2/2] COPY index.html /usr/share/nginx/html/                                         0.0s
 => exporting to image                                                                          1.9s
 => => exporting layers                                                                         0.1s
 => => exporting manifest sha256:9780bc2faef47e4065401d6369e101e0fade714930aee99d8afbae8b90a093dc      0.3s
 => => exporting config sha256:e9dc37a4e5a9ae97a64d84e29b00f1ac461b2d6143263e70c2c830b0f789a269        0.3s
 => => exporting attestation manifest sha256:008381c16761c3828e6ff327f2c2180c04736f572c6874c120c54acc2ab1  0.4s
 => => exporting manifest list sha256:ceb80188556a572e28caccd112b4aa8207cc7fddb11a9adb393c7ee3a4a39af7   0.2s
 => => naming to docker.io/library/guestbook-frontend:latest                                    0.0s
 => => unpacking to docker.io/library/guestbook-frontend:latest                                 0.1s
esha@ESHA-DELL:~/guest-book/frontend$
```

TASK 6: Webpage at localhost:8080



Guest Book

Powered by Docker Compose

Your name

Leave a message...

Sign Guest Book

esha khan
i am a girl
28/12/2025, 22:56:53

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.