



National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir

Submitted by:

Ahmad Fawad

Reg number:

1129

Lab no. :

11

Semester:

5th

Docker file code Backend:

FROM node:lts-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY server.js .

EXPOSE 3000

```
CMD ["node", "server.js"]
```

Dockerfile code frontend:

```
FROM nginx:trixie
```

```
COPY index.html /usr/share/nginx/html/
```

```
EXPOSE 80
```

Docker-compose.yml

```
version: "3.8"
```

```
services:
```

```
  mongodb:
```

```
    image: mongo:7-jammy
```

```
    container_name: guestbook-db
```

```
  volumes:
```

```
    - mongo_data:/data/db
```

```
  networks:
```

```
    - guestbook-net
```

```
  api:
```

```
    build: ./backend
```

container_name: guestbook-api

environment:

- MONGO_URL=mongodb://mongodb:27017/guestbook

ports:

- "3000:3000"

depends_on:

- mongodb

networks:

- guestbook-net

web:

image: nginx:alpine

container_name: guestbook-web

ports:

- "8080:80"

volumes:

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

depends_on:

- api

networks:

- guestbook-net

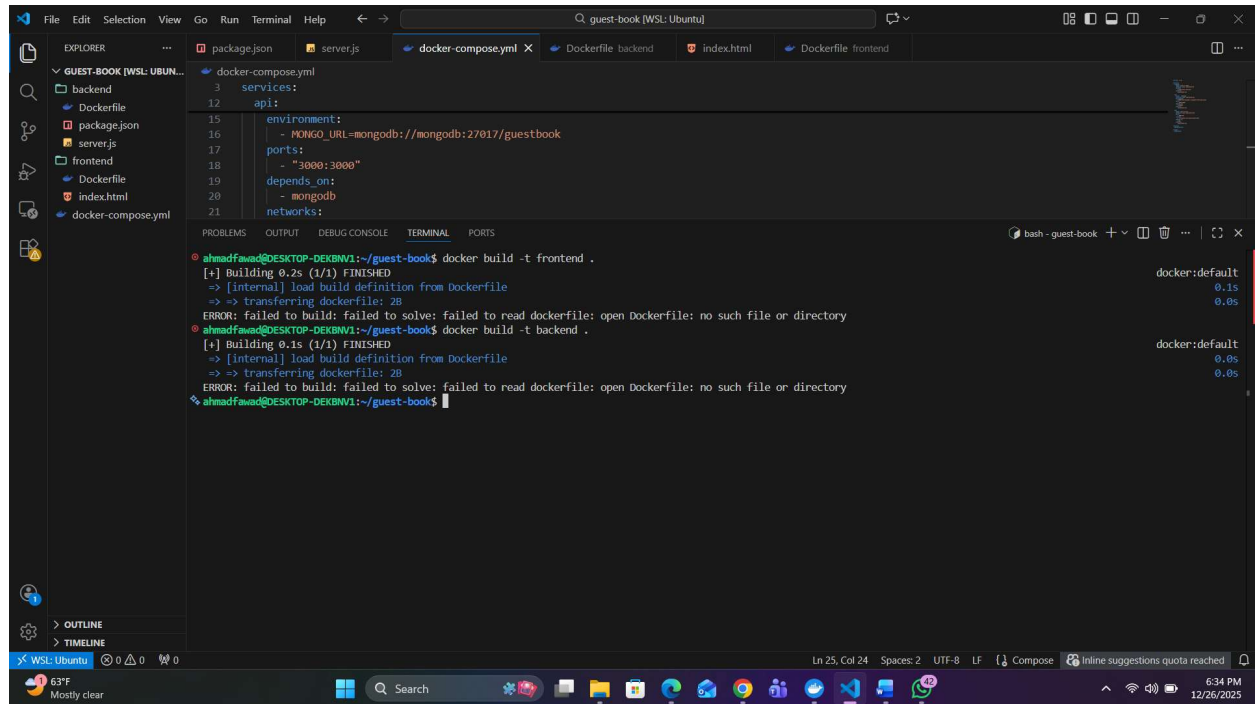
networks:

guestbook-net:

volumes:

mongo_data:

image build both front and back end:



The screenshot shows a Visual Studio Code editor window with a project named 'guest-book' (WSL: Ubuntu). The Explorer sidebar on the left shows the file structure: 'backend' (containing 'Dockerfile'), 'package.json', 'server.js', 'frontend' (containing 'Dockerfile'), 'index.html', and 'docker-compose.yml'. The main editor displays the 'docker-compose.yml' file with the following content:

```
3 services:
12   api:
15     environment:
16       - MONGO_URL=mongodb://mongodb:27017/guestbook
17     ports:
18       - "3000:3000"
19     depends_on:
20       - mongodb
21     networks:
```

The terminal at the bottom shows the execution of 'docker build' commands for the 'frontend' and 'backend' services. The output for 'frontend' is:

```
ahmadfawad@DESKTOP-DEKBNV1:~/guest-book$ docker build -t frontend .
[+] Building 0.2s (1/1) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2B
ERROR: failed to build: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
```

The output for 'backend' is:

```
ahmadfawad@DESKTOP-DEKBNV1:~/guest-book$ docker build -t backend .
[+] Building 0.1s (1/1) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2B
ERROR: failed to build: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
```

The terminal shows that the Dockerfiles for both services are not found, resulting in build errors. The status bar at the bottom indicates 'Ln 25, Col 24', 'Spaces: 2', 'UTF-8', 'LF', and 'Compose'.

Final output:

