



**National Textile University**

**Department of Computer Science**

**Subject**

**Operating system**

**Submitted to:**

Dr. Nasir Mehmood

**Submitted by:**

Abdul Rehman

**Registration Number**

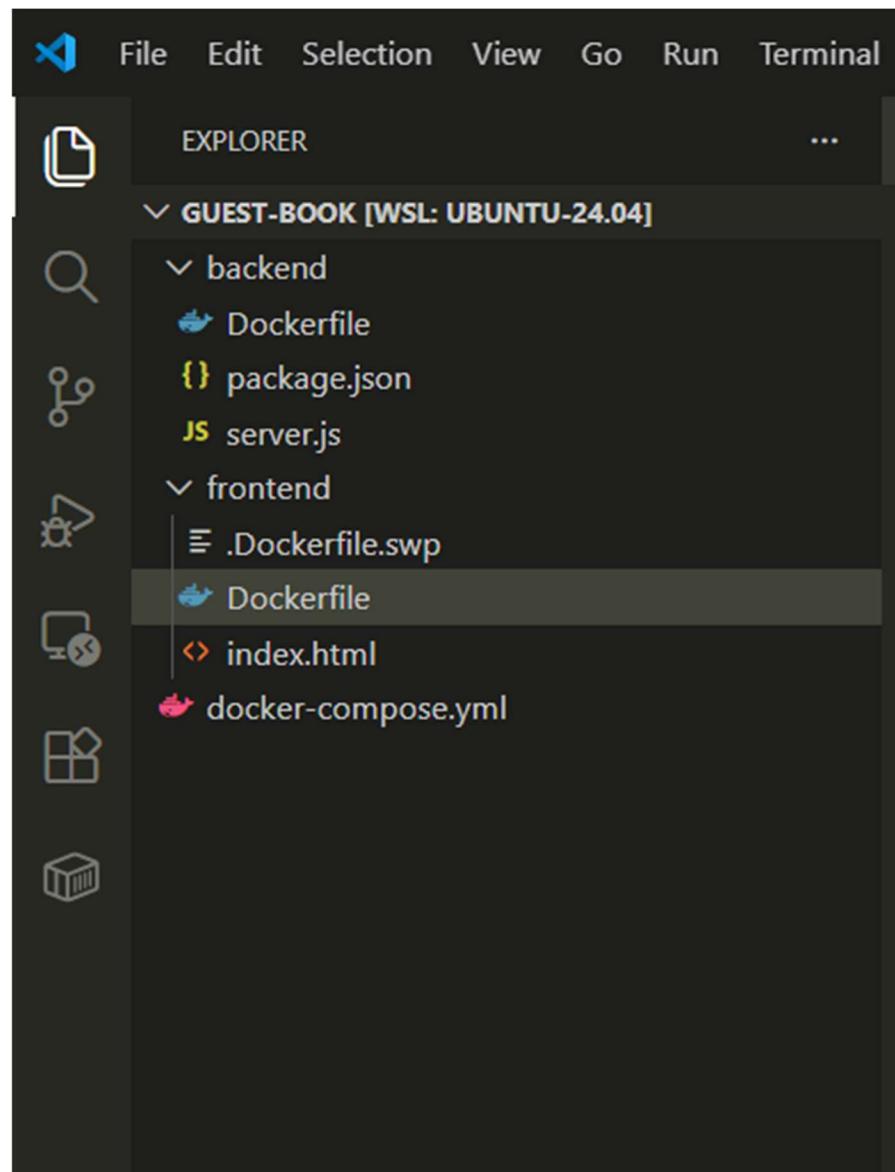
23-NTU-CS-1122

**Docker Home Task**

**Semester**

5th

**Whole project structure:**



Backend image creation:

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a project structure for "GUEST-BOOK [WSL: UBUNTU-24.04]". It includes a "backend" folder containing a "Dockerfile", "package.json", and "server.js"; and a "frontend" folder containing ".Dockerfile.swp", "Dockerfile", "index.html", and "docker-compose.yml".
- Code Editor:** Displays the contents of the "Dockerfile frontend" file. The code is as follows:

```
frontend > Dockerfile > ...
1 # Use a lightweight Nginx image
2 FROM nginx:alpine
3
4 # Copy all frontend files (like index.html) to Nginx web folder
5 COPY . /usr/share/nginx/html
6
7 # Expose port 80
8 EXPOSE 80
9
```

- Terminal View:** Shows the command "abdu@DESKTOP-NGRB9UV:~/Operating System/docker/lab3/guest-book\$ docker build -t guestbook-frontend ./frontend" being run. The terminal output shows the build process for both the frontend and backend containers.
- Bottom Status Bar:** Shows the WSL Ubuntu-24.04 environment, the search bar, and various system icons.

## Frontend image creation:

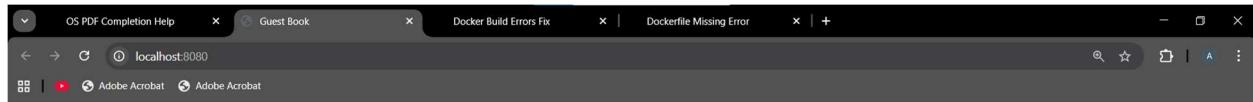
The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a project structure for "GUEST-BOOK [WSL: UBUNTU-24.04]". It includes a "backend" folder containing a "Dockerfile", "package.json", and "server.js"; and a "frontend" folder containing ".Dockerfile.swp", "Dockerfile", "index.html", and "docker-compose.yml".
- Code Editor:** Displays the contents of the "Dockerfile backend" file. The code is as follows:

```
frontend > Dockerfile > ...
1 # Use a lightweight Nginx image
2 FROM nginx:alpine
3
4 # Copy all frontend files (like index.html) to Nginx web folder
5 COPY . /usr/share/nginx/html
6
7 # Expose port 80
8 EXPOSE 80
9
```

- Terminal View:** Shows the command "abdu@DESKTOP-NGRB9UV:~/Operating System/docker/lab3/guest-book\$ docker build -t guestbook-backend ./backend" being run. The terminal output shows the build process for both the frontend and backend containers.
- Bottom Status Bar:** Shows the WSL Ubuntu-24.04 environment, the search bar, and various system icons.

## Webpage ScreenShot:



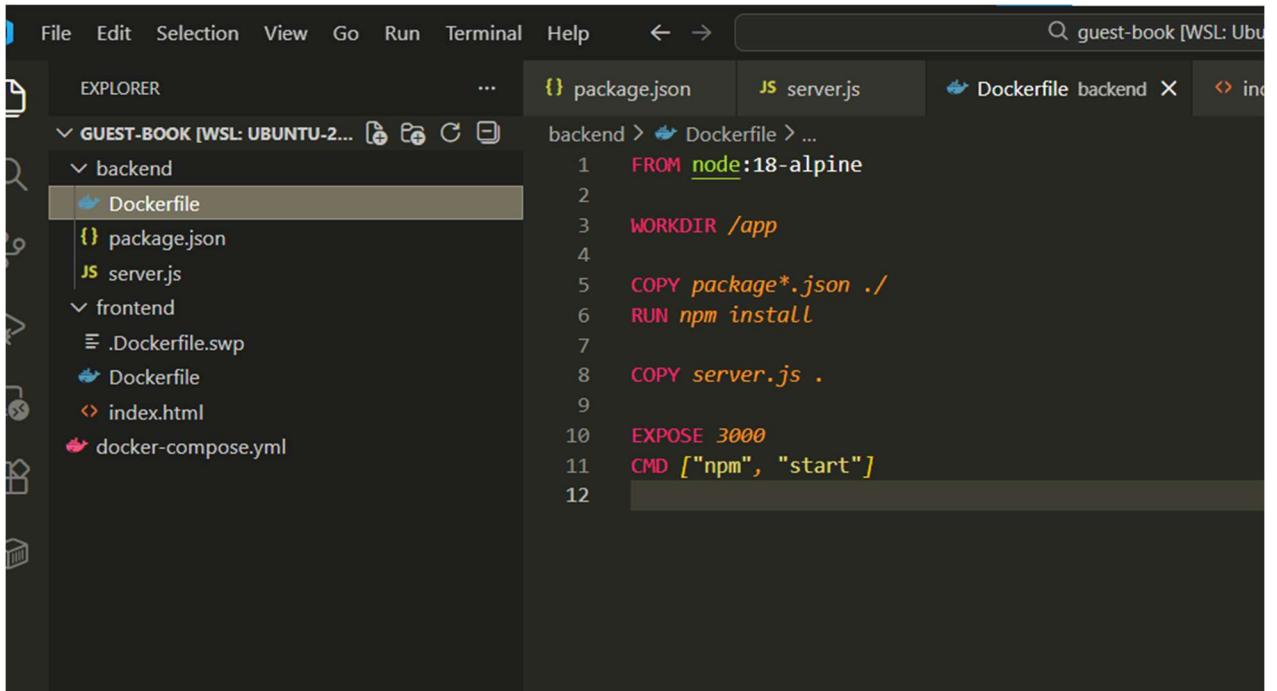
# Guest Book



docker-compose.yml File Code:

```
1  version: "3.8"
2
3  services:
4    mongodb:
5      image: mongo:7
6      container_name: guestbook-db
7      volumes:
8        - mongo_data:/data/db
9      networks:
10        - guestbook-net
11
12  api:
13    build: ./backend
14    container_name: guestbook-api
15    environment:
16      - MONGO_URL=mongodb://mongodb:27017/guestbook
17    ports:
18      - "3000:3000"
19    depends_on:
20      - mongodb
21    networks:
22      - guestbook-net
23
24  web:
25    build: ./frontend
26    container_name: guestbook-web
27    ports:
28      - "8080:80"
29    depends_on:
30      - api
31    networks:
32      - guestbook-net
33
34  networks:
35    guestbook-net:
36
37  volumes:
38    mongo_data:
39
```

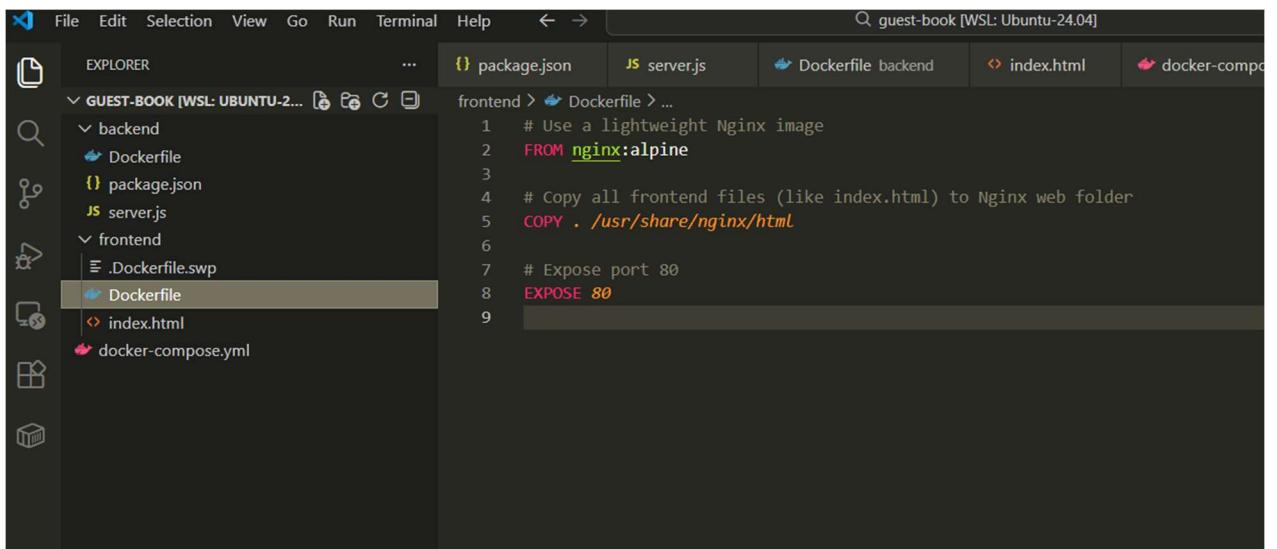
## Backend/Dockerfile:



The screenshot shows the Visual Studio Code interface with the title bar "guest-book [WSL: Ubuntu]". The Explorer sidebar on the left shows a project structure for "GUEST-BOOK [WSL: UBUNTU-2...]" with folders "backend" and "frontend", and files "Dockerfile", "package.json", "server.js", ".Dockerfile.swp", "index.html", and "docker-compose.yml". The main editor area displays the Dockerfile for the backend:

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json .
RUN npm install
COPY server.js .
EXPOSE 3000
CMD ["npm", "start"]
```

## Frontend/Dockerfile:



The screenshot shows the Visual Studio Code interface with the title bar "guest-book [WSL: Ubuntu-24.04]". The Explorer sidebar on the left shows a project structure for "GUEST-BOOK [WSL: UBUNTU-2...]" with folders "backend" and "frontend", and files "Dockerfile", "package.json", "server.js", ".Dockerfile.swp", "index.html", and "docker-compose.yml". The main editor area displays the Dockerfile for the frontend:

```
# Use a lightweight Nginx image
FROM nginx:alpine
# Copy all frontend files (like index.html) to Nginx web folder
COPY . /usr/share/nginx/html
# Expose port 80
EXPOSE 80
```