



# National Textile University

## Department of Computer Science

Subject:  
Operating System

---

Submitted to:

Sir Nasir

---

Submitted by:

Shanza Batool

---

Reg. number:

23-NTU-CS-FL-1209

---

# Semester: 5th- A

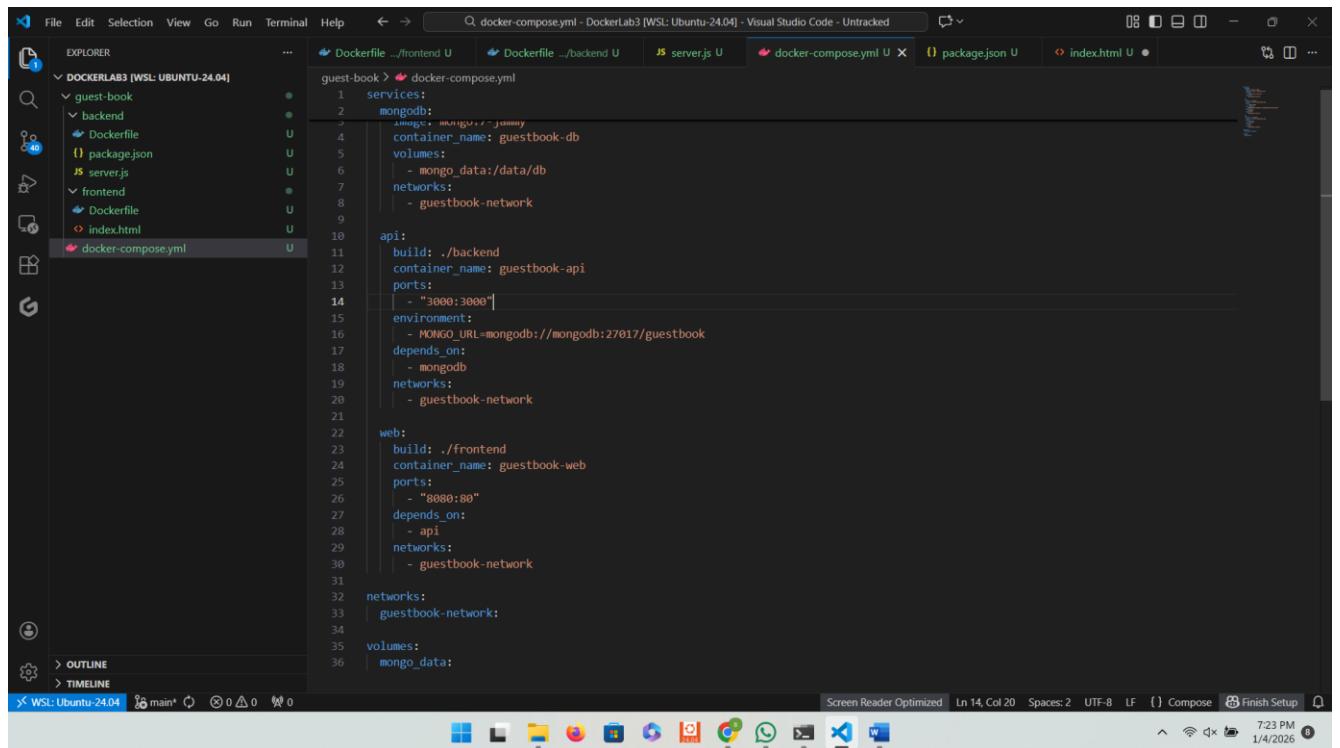
---

## Docker LAB\_3

---

**Task\_01:** Create networks to facilitate communication between frontend, backend and database.

**Code:**



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "DOCKERLAB3 [WSL: UBUNTU-24.04]". It includes a "guest-book" folder containing "backend" (with Dockerfile, package.json, server.js), "frontend" (with Dockerfile, index.html), and "docker-compose.yml".
- Code Editor:** The active tab is "docker-compose.yml". The code defines a multi-service Docker Compose configuration:

```
version: '3.8'
services:
  mongo:
    image: mongo:4.4
    container_name: guestbook-db
    volumes:
      - mongo_data:/data/db
    networks:
      - guestbook-network
  api:
    build: ./backend
    container_name: guestbook-api
    ports:
      - "3000:3000"
    environment:
      - MONGO_URL=mongodb://mongodb:27017/guestbook
    depends_on:
      - mongodb
    networks:
      - guestbook-network
  web:
    build: ./frontend
    container_name: guestbook-web
    ports:
      - "8080:80"
    depends_on:
      - api
    networks:
      - guestbook-network
networks:
  guestbook-network:
    driver: bridge
volumes:
  mongo_data:
```

- Bottom Status Bar:** Shows "WSL: Ubuntu-24.04" as the active workspace, along with other status indicators like screen reader optimization and file statistics.

## Task\_02: Running of “docker compose up –build”

```
version: '3.8'
services:
  backend:
    image: mongo:7-jammy
    container_name: guestbook-db
    volumes:
      - mongo_data:/data/db
    networks:
      - guestbook-network
  frontend:
    build: ./backend
    container_name: guestbook-apis
    ports:
      - "3000:3000"
    environment:
      - MONGO_URL=mongodb://mongodb:27017/guestbook
    depends_on:
      - mongodb
    networks:
      - guestbook-network
```

```
shanzha1209@DESKTOP-LISBTIJ:~/Labs_05/DockerLab3/guest-book$ docker compose up --build
, "environment": {"distmod": "ubuntu2204", "distrarch": "x86_64", "target_arch": "x86_64"} } }
guestbook_db | [{"t": ("date": "2026-01-04T14:34:28.282+00:00"), "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Ubuntu", "version": "22.04"} } }
guestbook_db | [{"t": ("date": "2026-01-04T14:34:28.282+00:00"), "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"bindip": "*"} } } }
guestbook_db | [{"t": ("date": "2026-01-04T14:34:28.335+00:00"), "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the wiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startupWarnings"] }
guestbook_db | [{"t": ("date": "2026-01-04T14:34:28.338+00:00"), "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": "create_cache_size=1413M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,prefix=log,statistics_log=(wait=0)),json_output=(error,message),verbose=[recovery_progress:1,checkpoint_progress:1,compact_progress:1,backup:0,checkpoint:0,compact:0,evict:0,history:store:0,recovery:0,rtsi:0,salvage:0,tiered:0,timestamp:0,transaction:0,verify:0,log:0],}" }
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:8080 /forwards/expose returned unexpected status: 500
shanzha1209@DESKTOP-LISBTIJ:~/Labs_05/DockerLab3/guest-book$
```

## Task\_03: Don't forget to create named volume with the same name you've used in dockercompose.yml

Name	Tag	Image ID	Created	Size	Actions
node-docker-app	latest	223401ef362e	14 days ago	1.58 GB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
node	lts-alpine3.23	c921b97d4b74	18 days ago	226.35 MB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
ubuntu	latest	c35e29c94501	3 months ago	119.25 MB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
mongo	7-jammy	8ddd3db4d263	16 days ago	1.14 GB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
guest-book-web	latest	c6e647bad03f	9 days ago	81.15 MB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
guest-book-api	latest	15ba84675396	9 days ago	1.61 GB	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

## **Task\_04: Screenshot of whole project file**

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

- File Explorer:** Shows the project structure under "DOCKERLAB3".
- Editor:** The "docker-compose.yml" file is open, showing the configuration for a multi-service application. It includes services for MongoDB, a guestbook API, and a frontend.
- Terminal:** The terminal shows the command "docker compose up --build" being run, followed by a list of successful pull operations for various Docker images.
- Status Bar:** Shows the current file is "index.html 4", the terminal tab is "powershell - guest-book", and the status bar indicates "Ln 9, Col 1" and "1/1/2026".

```
version: '3.8'
services:
  mongodb:
    image: mongo:7-jammy
    container_name: guestbook-db
    volumes:
      - mongo_data:/data/db
    networks:
      - guestbook-network
  api:
    build: ./backend
    container_name: guestbook-api
    ports:
      - "3000:3000"
    environment:
      - MONGO_URL=mongodb://guestbook-db:27017/guestbook

networks:
  guestbook-network:
    driver: bridge

volumes:
  mongo_data:
    driver: local
```

```
[+] Running 11/11
  ✓ mongod Pulled
  ✓ 7e49dc6c156bb Pull complete
  ✓ d7381d7512ed Pull complete
  ✓ a4c2e1a29646 Pull complete
  ✓ 487128c0e848 Pull complete
  ✓ a4867e714827 Pull complete
  ✓ d0fc8d2469ec Pull complete
  ✓ 47c219b2ddaf5 Pull complete
  ✓ 929ce23bedbe Pull complete
  ✓ 9e481d538c40 Download complete
  ✓ fe1da9a13d38 Download complete
[+] Building 12.3s (5/5) FINISHED
  => [internal] load local bake definitions
  => => reading from stdin 1.17kB
  => [api internal] load build definition from Dockerfile
  =>> => transferring dockerfile: 159B
```

**Task\_05:** Output of both image building commands e.g "docker build -t frontend."

The screenshot shows the Docker Desktop application window. The left sidebar has 'Images' selected. The main area displays a list of Docker images with columns for Name, Tag, Image ID, Created, Size, and Actions. The 'Actions' column contains icons for each image entry.

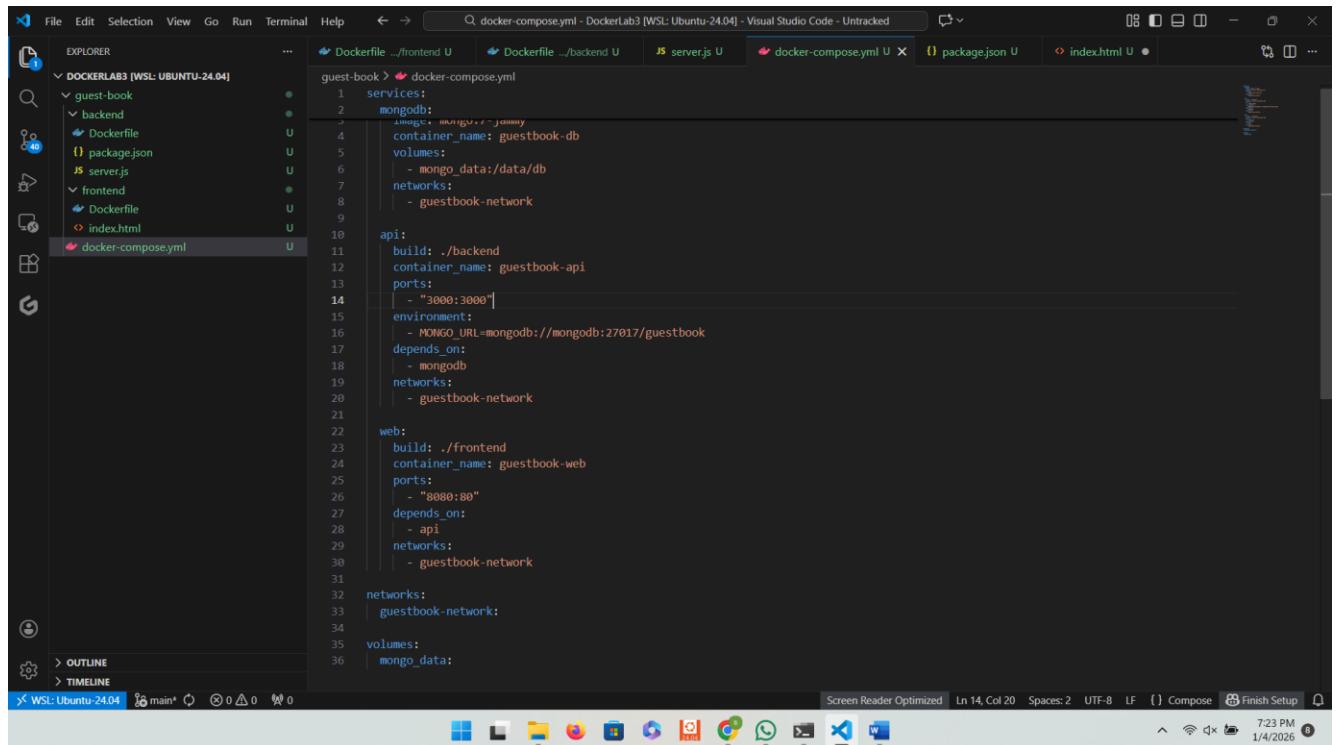
	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	node-docker-app	latest	223401ef362e	14 days ago	1.58 GB	
<input type="checkbox"/>	node	lts-alpine3.23	c921b97d4b74	18 days ago	226.35 MB	
<input type="checkbox"/>	ubuntu	latest	c35e29c94501	3 months ago	119.25 MB	
<input type="checkbox"/>	mongo	7-jammy	8ddd3db4d263	16 days ago	1.14 GB	
<input type="checkbox"/>	guest-book-web	latest	c6e647bad03f	9 days ago	81.15 MB	
<input type="checkbox"/>	guest-book-api	latest	15ba84675396	9 days ago	1.61 GB	
<input type="checkbox"/>	frontend	latest	4ccfeb5d4a62	9 days ago	81.15 MB	
<input type="checkbox"/>	backend	latest	bd222a23bdd4	9 days ago	1.61 GB	

Showing 8 items

Engine running | RAM 3.60 GB CPU 1.76% Disk: 5.70 GB used (limit 1006.85 GB) | Terminal v4.55.0 | 7:38 PM 1/4/2026

## Task\_06: Copy contents of docker-compose and both Docker files in the pdf.

Code of docker- compose file:



```
version: '3.8'
services:
  mongo:
    image: mongo:4.4
    container_name: guestbook-db
    volumes:
      - mongo_data:/data/db
    networks:
      - guestbook-network
  api:
    build: ./backend
    container_name: guestbook-api
    ports:
      - "3000:3000"
    environment:
      - MONGO_URL=mongodb://mongo:27017/guestbook
    depends_on:
      - mongo
    networks:
      - guestbook-network
  web:
    build: ./frontend
    container_name: guestbook-web
    ports:
      - "8080:80"
    depends_on:
      - api
    networks:
      - guestbook-network
volumes:
  mongo_data:
```

## Code of docker file of backend:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Shows a project structure for "DOCKERLAB3 [WSL: UBUNTU-24.04]". It includes a "guest-book" folder containing "backend" and "frontend" subfolders. Inside "backend", there are "Dockerfile", "package.json", and "server.js".
- EDITOR**: The "Dockerfile" tab is active, displaying the following Dockerfile code:

```
guest-book > backend > Dockerfile
1 # Use Node.js image
2 FROM node:18
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy package.json and install dependencies
8 COPY package.json .
9 RUN npm install
10
11 # Copy backend source code
12 COPY . .
13
14 # Expose port
15 EXPOSE 3000
16
17 # Start server
18 CMD ["node", "server.js"]
19
```

- TERMINAL**: The terminal shows the command: `shanza1209@DESKTOP-LIS8TTJ:~/Labs_05/DockerLab3/guest-book$ docker compose up --build`. Below it, a progress bar indicates "mongod" is pulling 285.1MB / 285.6MB.
- STATUS BAR**: Shows "WSL: Ubuntu-24.04" and other system information like battery level and date.

## Code of docker file of frontend:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Shows a project structure for "DOCKERLAB3 [WSL: UBUNTU-24.04]". It includes a "guest-book" folder containing "backend" and "frontend" subfolders. Inside "frontend", there are "Dockerfile", "index.html", and "docker-compose.yml".
- EDITOR**: The "Dockerfile" tab is active, displaying the following Dockerfile code:

```
guest-book > frontend > Dockerfile
1 # Use lightweight Nginx image
2 FROM nginx:alpine
3
4 # Copy Frontend HTML
5 COPY index.html /usr/share/nginx/html/index.html
6
7 # Expose port
8 EXPOSE 80
9
```

- TERMINAL**: The terminal shows the command: `shanza1209@DESKTOP-LIS8TTJ:~/Labs_05/DockerLab3/guest-book$ docker compose up --build`. The output shows the build process for multiple services, including "guest-book-api" and "guest-book-web".
- STATUS BAR**: Shows "WSL: Ubuntu-24.04" and other system information like battery level and date.