



NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Eman Faisal

23-NTU-CS-1149

SECTION SE: 5th(A)

Lab 3-docker

SUBMITTED TO:

Sir Nasir Mehmood

SUBMISSION DATE:

12-28-2025

1. Screenshot of Whole project structure:

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

- EXPLORER**: Shows the project structure under "Docker_3 (WSL: UBUNTU-24.04)". It includes a "backend" folder containing a Dockerfile, package.json, and server.js; and a "frontend" folder containing a Dockerfile, index.html, and docker-compose.yml.
- CODE EDITOR**: The "docker-compose.yml" file is open, displaying its contents:

```
version: '3.8'
services:
  backend:
    image: mongo:7-jammy
    container_name: guestbook-db
    networks:
      - center_network
    volumes:
      - mongo_data:/data/db
    environment:
      MONGO_URL: mongodb://guestbook-db:27017
  api:
    build: ./backend
    image: <image_name>
    container_name: guestbook-api
    environment:
      MONGO_URL: mongodb://guestbook-db:27017
```

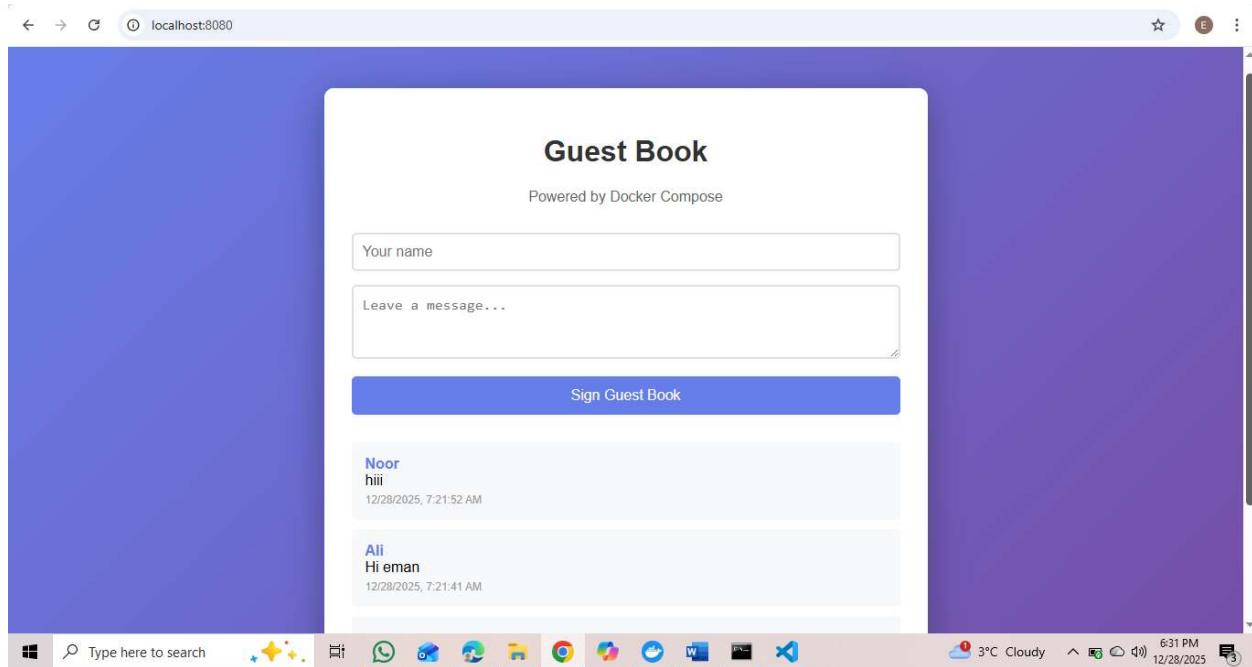
- TERMINAL**: Shows the command "root@DESKTOP-GFUS3VG:~/emouser/docker_3/frontend# docker pull mongo:7-jammy" and its output, which lists several mongo:7-jammy images from the docker.io library.
- STATUS BAR**: Shows "Ln 37, Col 16 (695 selected) Spaces:2 UTF-8 LF Compose".

2. Output of both image building commands e.g "docker build -t frontend ."

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

- EXPLORER**: Shows the project structure under "Docker_3 (WSL: UBUNTU-24.04)". It includes a "backend" folder containing a Dockerfile, package.json, and server.js; and a "frontend" folder containing a Dockerfile, index.html, and docker-compose.yml.
- CODE EDITOR**: The "server.js" file is open, showing code for a Node.js application using Express and Mongoose.
- TERMINAL**: Shows the command "root@DESKTOP-GFUS3VG:~/emouser/docker_3/frontend# docker build -t frontend ." and its output, which details the build process step-by-step, including transferring context, loading build context, resolving dependencies, exporting layers, and creating the manifest.
- STATUS BAR**: Shows "Ln 9, Col 6 Spaces:4 UTF-8 LF JavaScript".

3. Webpage at localhost:8080:



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

Docker compose file:

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

  # Backend API
  api:
    build: ./backend
    # image: <image_name>
    container_name: guestbook-api
    environment:
      - MONGO_URL=mongodb://mongodb:27017
    ports:
```

```
- "3000:3000"
depends_on:
- mongodb
networks:
- guestbook-network
# Frontend
web:
  build: ./frontend
  # image: <image_name>
  container_name: guestbook-web
  ports:
  - "8080:80"
  depends_on:
  - api
  networks:
  - guestbook-network
networks:
  guestbook-network:
volumes:
  mongo_data:
```

Docker file of backend:

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

Docker file of frontend:

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