

National Textile University, Faisalabad

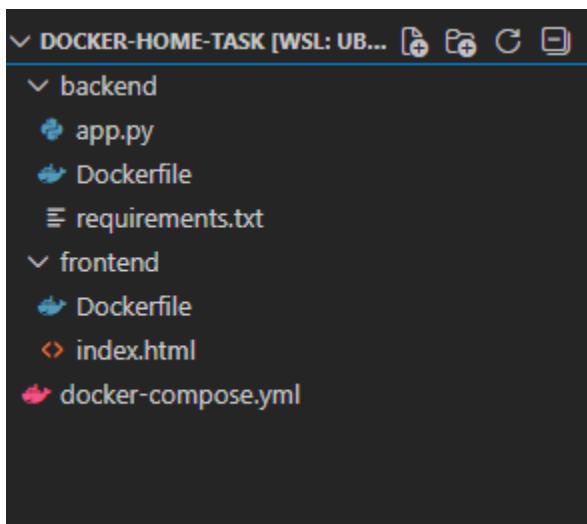


Department of Computer Science

Name	Dawood Saif
Class	SE-5 th (A)
Reg. No.	23-NTU-CS-1145
Course	Operating system
Submitted To	Sir Nasir Mehmood
Submission Date	28/12/2025
Assignment	Lab 14 (Docker Home Task)

Docker Home Task

File Structure:



Docker-Compose.yml File Code:

```
version: "3.9"
```

```
networks:
```

```
  frontend_net:
```

```
  backend_net:
```

```
volumes:
```

```
  db_data:
```

```
services:
```

```
  frontend:
```

```
    build: ./frontend
```

```
  ports:  
    - "8080:80"  
  
  depends_on:  
    - backend  
  
  networks:  
    - frontend_net
```

```
  backend:  
    build: ./backend  
  
    ports:  
      - "5000:5000"  
  
    depends_on:  
      - db  
  
    networks:  
      - frontend_net  
      - backend_net
```

```
  db:  
    image: postgres:16  
  
    environment:  
      POSTGRES_USER: demo  
      POSTGRES_PASSWORD: demo  
      POSTGRES_DB: demo_db  
  
    volumes:  
      - db_data:/var/lib/postgresql/data  
  
    networks:  
      - backend_net
```

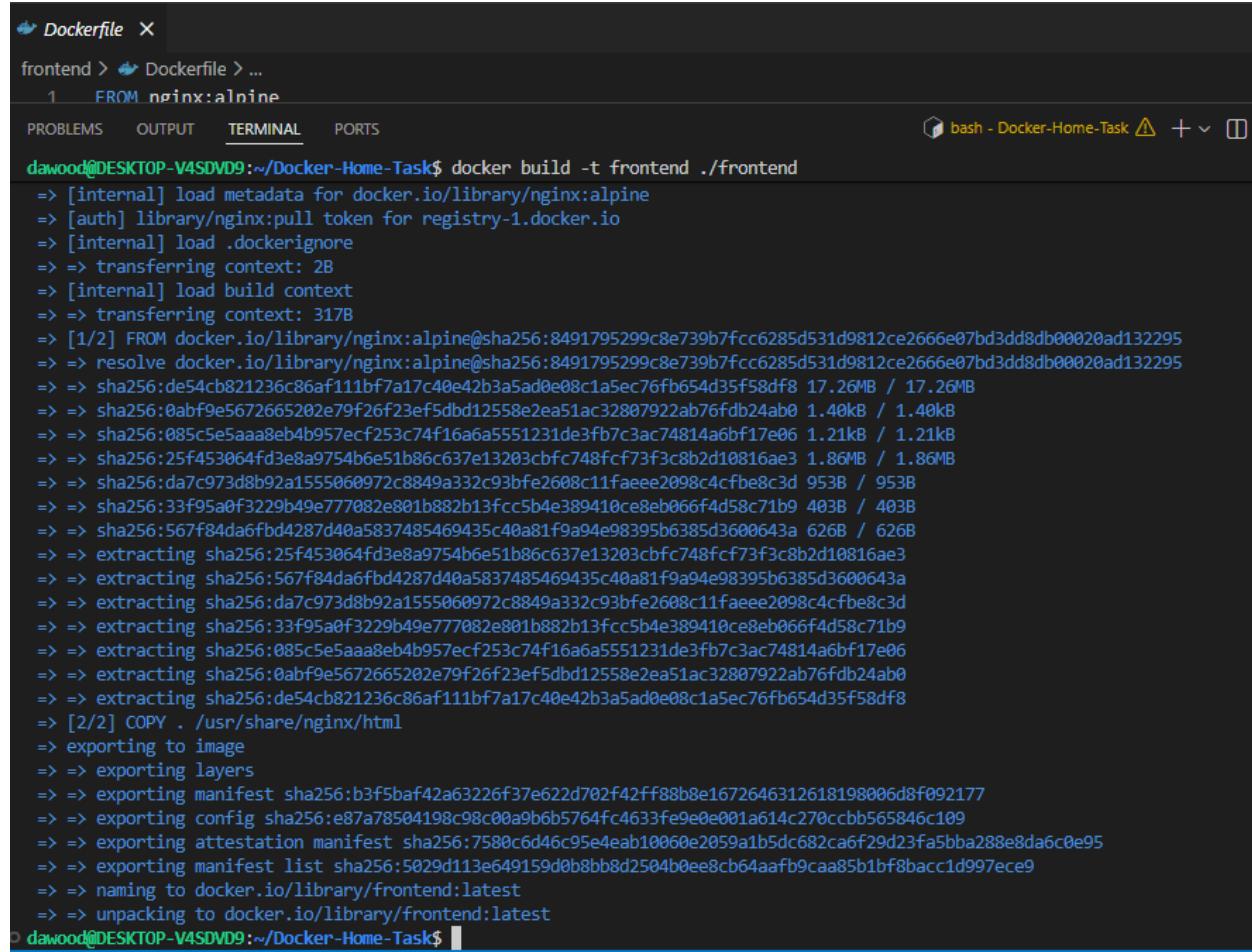
Dockerfile Frontend:

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

DockerFile Backend:

```
FROM python:3.11-slim
WORKDIR /app
COPY .
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]
```

- docker build -t frontend ./frontend :



The screenshot shows a terminal window titled "Dockerfile" with the command "docker build -t frontend ./frontend" being run. The output of the build process is displayed, showing various stages of the build, including pulling the Alpine Linux base image, extracting layers, and copying files from the host's /usr/share/nginx/html directory into the container. The terminal interface includes tabs for PROBLEMS, OUTPUT, TERMINAL, and PORTS, and a header bar with icons for bash, Dockerfile, and other tasks.

```

Dockerfile X
frontend > Dockerfile > ...
1 FROM nginx:alpine
PROBLEMS OUTPUT TERMINAL PORTS
dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ docker build -t frontend ./frontend
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 317B
=> [1/2] FROM docker.io/library/nginx:alpine@sha256:8491795299c8e739bf7fcc6285d531d9812ce2666e07bd3dd8db00020ad132295
=> => resolve docker.io/library/nginx:alpine@sha256:8491795299c8e739bf7fcc6285d531d9812ce2666e07bd3dd8db00020ad132295
=> => sha256:de54cb821236c86af111bf7a17c40e42b3a5ad0e08c1a5ec76fb654d35f58df8 17.26MB / 17.26MB
=> => sha256:0abf9e5672665202e79f26f23ef5fdb12558e2ea51ac32807922ab76fdb24ab0 1.40kB / 1.40kB
=> => sha256:085c5e5aaa8eb4b957ecf253c74f16a6a5551231de3fb7c3ac74814a6bf17e06 1.21kB / 1.21kB
=> => sha256:25f453064fd3e8a9754b6e51b86c637e13203cbfc748fcf73f3c8b2d10816ae3 1.86MB / 1.86MB
=> => sha256:da7c973d8b92a15550600972c8849a332c93bfe2608c11faeee2098c4cfbe8c3d 953B / 953B
=> => sha256:33f95a0f3229b49e777082e801b882b13fcc5b4e389410ce8eb066f4d58c71b9 403B / 403B
=> => sha256:567f84da6fdb4287d40a5837485469435c40a81f9a94e98395b6385d3600643a 626B / 626B
=> => extracting sha256:25f453064fd3e8a9754b6e51b86c637e13203cbfc748fcf73f3c8b2d10816ae3
=> => extracting sha256:567f84da6fdb4287d40a5837485469435c40a81f9a94e98395b6385d3600643a
=> => extracting sha256:da7c973d8b92a15550600972c8849a332c93bfe2608c11faeee2098c4cfbe8c3d
=> => extracting sha256:33f95a0f3229b49e777082e801b882b13fcc5b4e389410ce8eb066f4d58c71b9
=> => extracting sha256:085c5e5aaa8eb4b957ecf253c74f16a6a5551231de3fb7c3ac74814a6bf17e06
=> => extracting sha256:0abf9e5672665202e79f26f23ef5fdb12558e2ea51ac32807922ab76fdb24ab0
=> => extracting sha256:de54cb821236c86af111bf7a17c40e42b3a5ad0e08c1a5ec76fb654d35f58df8
=> [2/2] COPY . /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:b3f5baf42a63226f37e622d702f42ff88b8e1672646312618198006d8f092177
=> => exporting config sha256:e87a78504198c98c00a9b6b5764fc4633fe9e0e001a614c270ccb565846c109
=> => exporting attestation manifest sha256:7580c6d46c95e4eb10060e2059a1b5dc682ca6f29d23fa5bba288e8da6c0e95
=> => exporting manifest list sha256:5029d113e649159d0b8bb8d2504b0ee8cb64aafb9caa85b1bf8bacc1d997ece9
=> => naming to docker.io/library/frontend:latest
=> => unpacking to docker.io/library/frontend:latest
dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ 

```

- docker build -t backend ./backend :

```

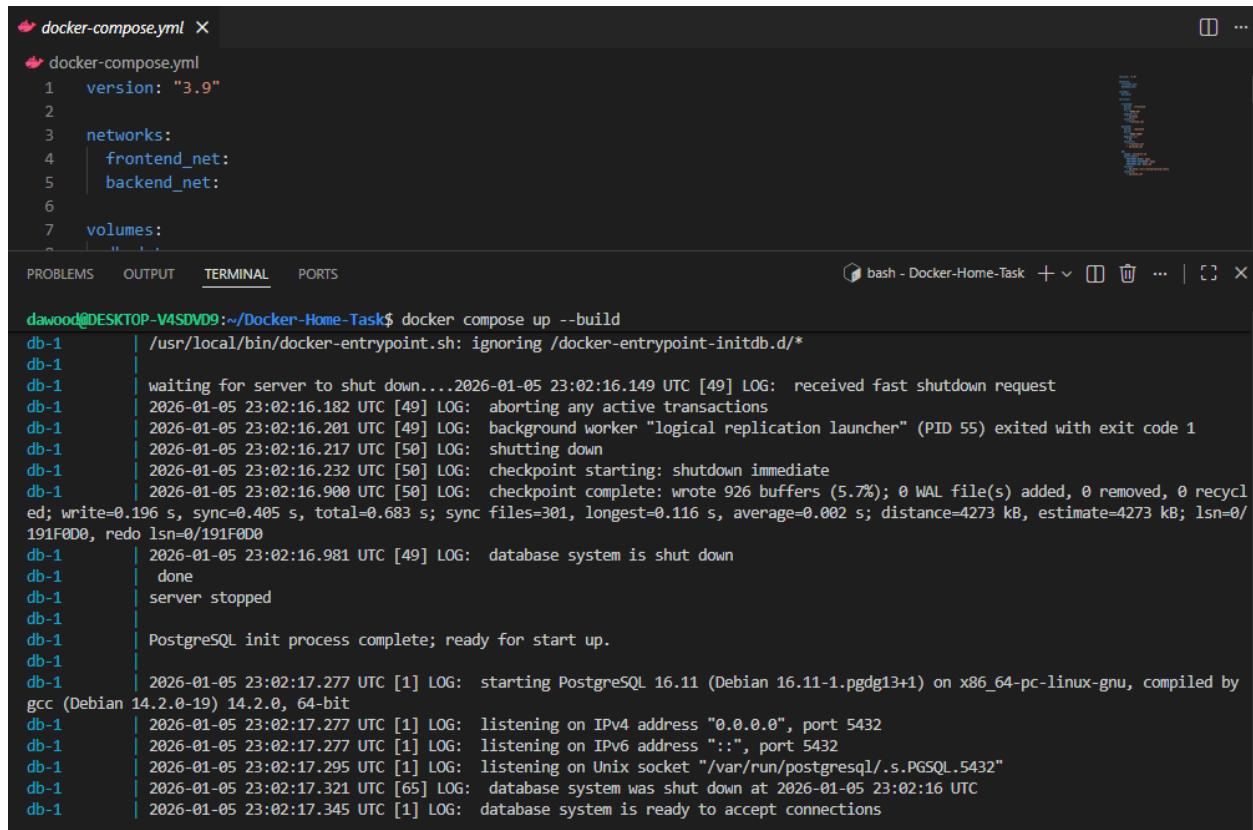
app.py 1 X
backend > app.py > ...
1   from flask import Flask
2   app = Flask(__name__)
3
4   @app.route("/")
5   def home():
6       return "Backend is running & connected to database ✓"
7

PROBLEMS 1 OUTPUT TERMINAL PORTS bash - Docker-Home-Task △ + v □

dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task/backend$ docker build -t backend ./backend
● dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ cd ..
● dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ docker build -t backend ./backend
[+] Building 23.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [internal] load metadata for docker.io/library/python:3.11-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.11-slim@sha256:1dd3dca85e22886e44fcad1bb7ccab6691dfa83db52214cf9e20696e095f3e36
=> => resolve docker.io/library/python:3.11-slim@sha256:1dd3dca85e22886e44fcad1bb7ccab6691dfa83db52214cf9e20696e095f3e36
=> [internal] load build context
=> => transferring context: 124B
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY .
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:59e9d98cce8c5ae8dcf971a3b7715308bdcc682b2d14e57c0a1938b31840307f
=> => exporting config sha256:4031bdf674e59732a81327627170166292d914e049b4f2d1335c2a8f4ccbceb0
=> => exporting attestation manifest sha256:f8dd6639d3e9919dfe0312f5bc6bea40578b5b3384e82ed7a3eb4847786778f8
=> => exporting manifest list sha256:9ee1f8db7a6cb89ad4c811f3237313ad0f0f1532de3d02a0021876aea7ef79
=> => naming to docker.io/library/backend:latest
=> => unpacking to docker.io/library/backend:latest
○ dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ 

```

- docker compose up --build;



```

dawood@DESKTOP-V4SDVD9:~/Docker-Home-Task$ docker compose up --build
db-1 | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
db-1 |
db-1 | waiting for server to shut down....2026-01-05 23:02:16.149 UTC [49] LOG:  received fast shutdown request
db-1 | 2026-01-05 23:02:16.182 UTC [49] LOG:  aborting any active transactions
db-1 | 2026-01-05 23:02:16.201 UTC [49] LOG:  background worker "logical replication launcher" (PID 55) exited with exit code 1
db-1 | 2026-01-05 23:02:16.217 UTC [50] LOG:  shutting down
db-1 | 2026-01-05 23:02:16.232 UTC [50] LOG:  checkpoint starting: shutdown immediate
db-1 | 2026-01-05 23:02:16.900 UTC [50] LOG:  checkpoint complete: wrote 926 buffers (5.7%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.196 s, sync=0.405 s, total=0.683 s; sync files=301, longest=0.116 s, average=0.002 s; distance=4273 kB, estimate=4273 kB; lsn=0/191F0D0, redo lsn=0/191F0D0
db-1 | 2026-01-05 23:02:16.981 UTC [49] LOG:  database system is shut down
db-1 | done
db-1 | server stopped
db-1 |
db-1 | PostgreSQL init process complete; ready for start up.
db-1 |
db-1 | 2026-01-05 23:02:17.277 UTC [1] LOG:  starting PostgreSQL 16.11 (Debian 16.11-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
db-1 | 2026-01-05 23:02:17.277 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db-1 | 2026-01-05 23:02:17.277 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db-1 | 2026-01-05 23:02:17.295 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.PGSQL.5432"
db-1 | 2026-01-05 23:02:17.321 UTC [65] LOG:  database system was shut down at 2026-01-05 23:02:16 UTC
db-1 | 2026-01-05 23:02:17.345 UTC [1] LOG:  database system is ready to accept connections

```

Local Host 8080 Screenshot:



Hello Dawood Frontend is Running

This page is served from Docker container.

