Ques 3: **Docker Registry, DockerHub, Create a Multi-Stage Build**

1. Docker Registry

🔷 What is a Docker Registry?

A **Docker Registry** is a storage and distribution system for Docker images. It allows you to **push**, **pull**, and **manage** images.

There are two types:

- **Public Registry**: Like DockerHub
- **Private Registry**: Self-hosted or cloud-based (e.g., AWS ECR, Azure ACR, Google GCR)

2. DockerHub

What is DockerHub?

DockerHub is the **default public registry** for Docker images.

It lets users:

- Search and download pre-built images
- Host their own public or private images
- Version control with tags
- Collaborate on teams and projects

Common DockerHub Commands:

Command	Description
`docker login`	Authenticate with DockerHub
`docker pull nginx`	Download image from DockerHub
`docker tag local-image username/image-name:tag`	Tag image for DockerHub
`docker push username/image-name:tag`	Push image to DockerHub
`docker search image-name`	Search DockerHub images

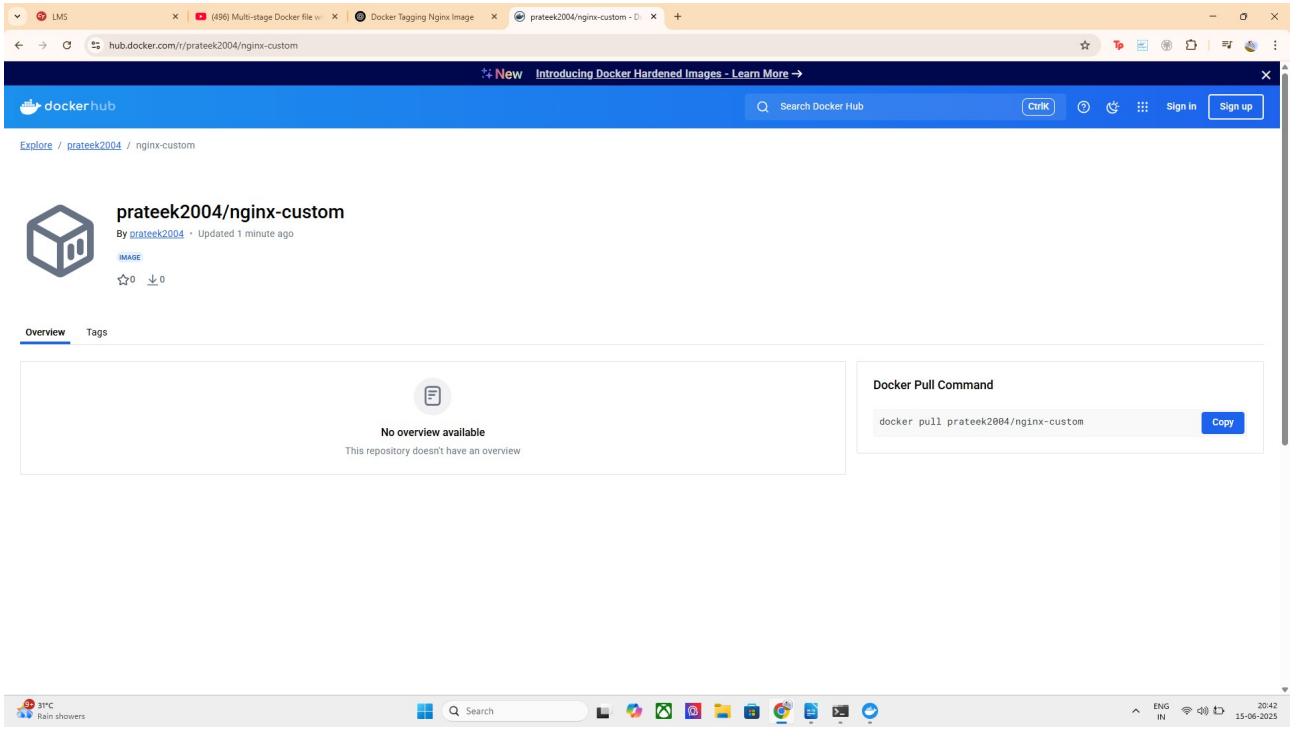
```
C:\Users\prate>docker login
Authenticating with existing credentials... [Username: prateek2004]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'


Login Succeeded

C:\Users\prate>
```

```
C:\Users\prate>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
dad67da3f26b: Already exists
3b00567da964: Pull complete
56b81cfa547d: Pull complete
1bc5dc8b475d: Pull complete
979e6233a40a: Pull complete
d2a7ba8dbfee: Pull complete
32e44235e1d5: Pull complete
Digest: sha256:6784fb0834aa7dbbe12e3d7471e69c290df3e6ba810dc38b34ae33d3c1c05f7d
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
C:\Users\prate>docker tag nginx prateek2004/nginx-custom:latest

C:\Users\prate>docker push yourusername/nginx-custom:latest
The push refers to repository [docker.io/yourusername/nginx-custom]
An image does not exist locally with the tag: yourusername/nginx-custom

C:\Users\prate>docker push prateek2004/nginx-custom:latest
The push refers to repository [docker.io/prateek2004/nginx-custom]
7e893c1b6ee8: Mounted from library/nginx
463308bed0c9: Mounted from library/nginx
4197a611afec: Mounted from library/nginx
3e96162769d5: Mounted from library/nginx
892e805f6f4f: Mounted from library/nginx
626ab8a5d57b: Mounted from library/nginx
7fb72a7d1a8e: Mounted from library/nginx
```

You can use to check :

https://hub.docker.com/r/prateek2004/nginx-custom



3. Create a Multi-Stage Build

◈ What is a Multi-Stage Build?

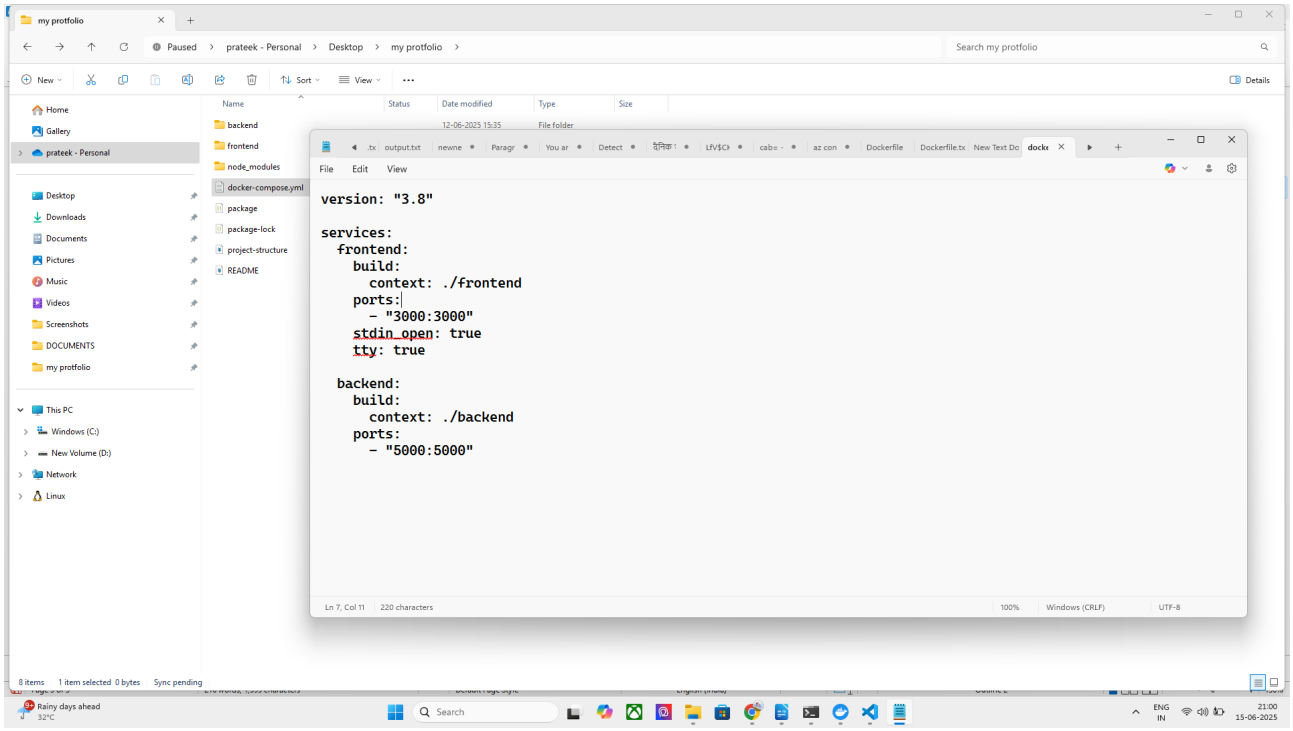
Multi-stage builds help:

- Optimize final image size
- Separate **build dependencies** from **runtime**
- Improve security and performance

Lets create 1st without multi bulid to check the efficiency :

1. Dockerfile for Frontend (No Multi-Stage)

```
# React dev environment (not optimized)
FROM node:18-alpine

WORKDIR /app
COPY . .
RUN npm install
CMD ["npm", "start"]
EXPOSE 3000
```



IN TTHIS APPROACH IT HAS MADE THE FILE OF (551 Mb) for the frontend only

```
Run 'docker image COMMAND --help' for more information on a command.
PS D:\OneDrive\Desktop\my protfolio> docker image ls
REPOSITORY                    TAG       IMAGE ID       CREATED          SIZE
myprotfolio-frontend          latest    da30a22769d8   3 minutes ago    551MB
my-python-app                 latest    44c082b8ea65   2 hours ago      127MB
ubuntu                        latest    bf16bdcff9c9   2 weeks ago      78.1MB
prateek2004/nginx-custom      latest    1e5f3c5b981a   2 months ago     192MB
nginx                         latest    1e5f3c5b981a   2 months ago     192MB
nginx                         alpine    6769dc3a703c   2 months ago     48.2MB
PS D:\OneDrive\Desktop\my protfolio>
```

LETS GO TO THE MULTI STAGE BUILT FOR THE SAME

NOW THE DOCKER FILE WILL BE

FOR FRONTEND IN 2 STAGES

```
# Stage 1: Build React app
FROM node:18-alpine AS builder
WORKDIR /app
COPY . .
RUN npm install
RUN npm run build

# Stage 2: Serve with Nginx
FROM nginx:alpine
COPY --from=builder /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

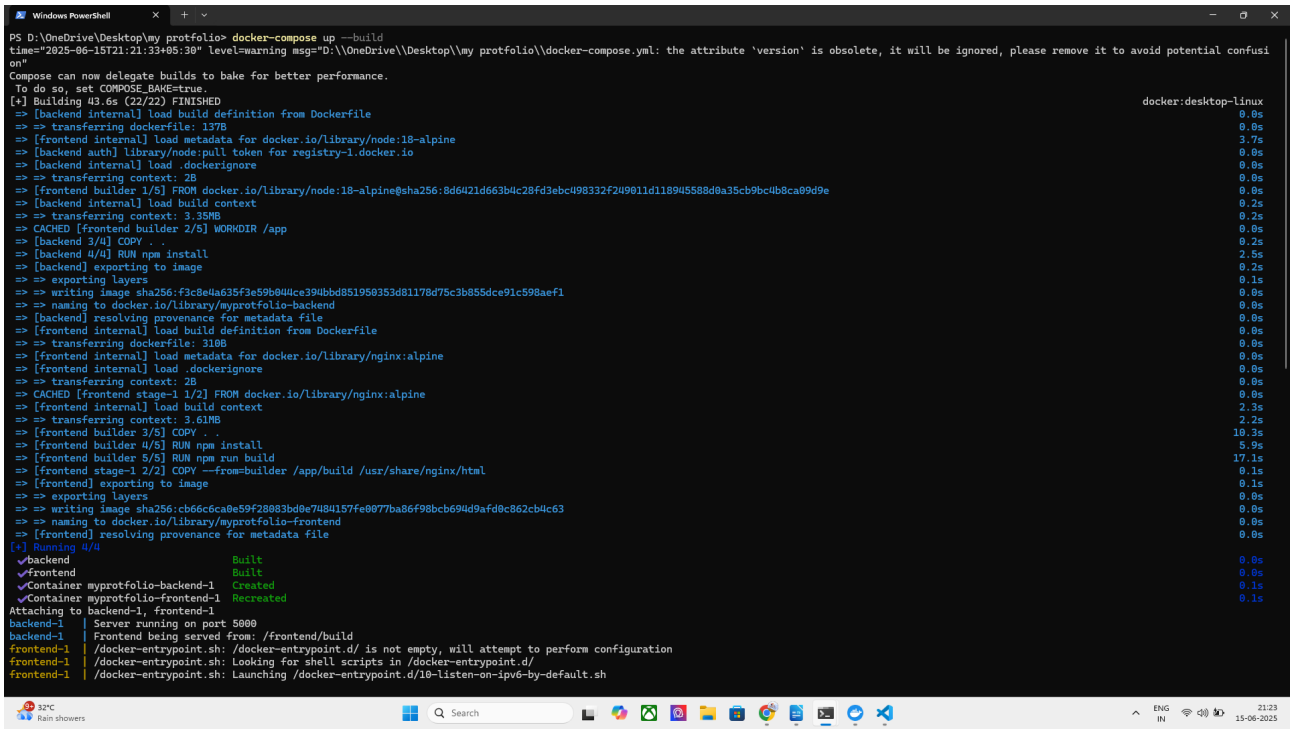
AND DOCKER-COMPOSE.YML FILE WILL BE

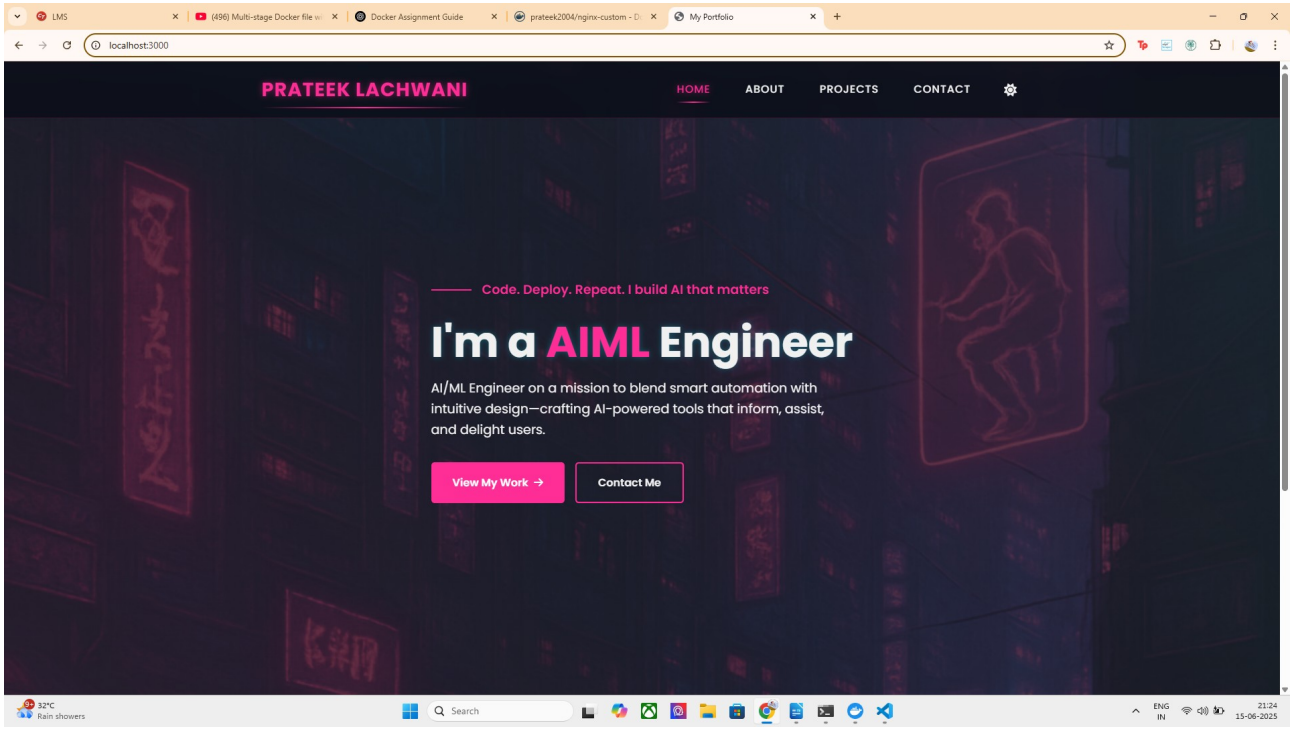
```
version: "3.8"

services:
  frontend:
    build:
      context: ./frontend
      dockerfile: Dockerfile
    ports:
      - "3000:80"
    depends_on:
      - backend

  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile
    ports:
- "5000:5000"
```

FOR BUILDING AND RUNNING THE CONTAINER

docker-compose up --build

NOW THE STEPS HAVE BEEN INCREASED TO 20





NOW THE NEW IMAGE IS OPTIMIZED WHERE FRONTEND ONLY MAKES UP (59.4 MB) THATS WHY MULTI STAAGE BUILT ARE PREFFERED