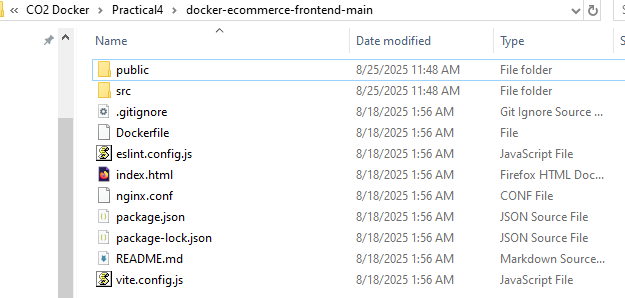
Practical 4: Build & Run Frontend App in Docker

=================================================

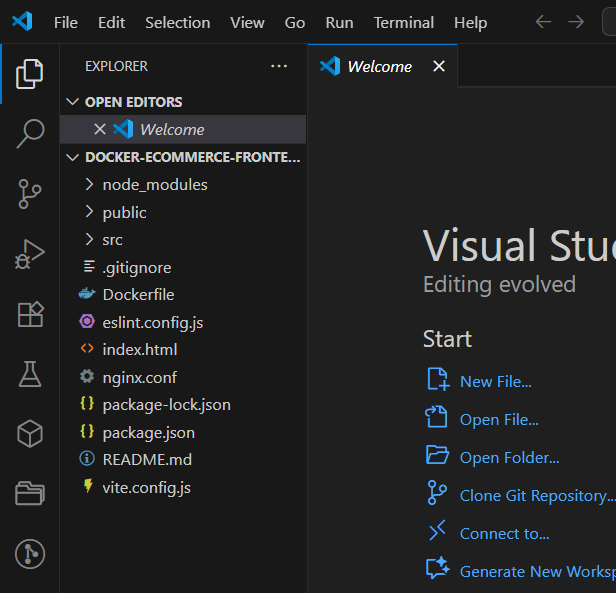
Step 1: Download the project by using the below link

<https://github.com/suneethabulla/docker-ecommerce-frontend.git>

🡪After downloading it will be in zip file extract all the files from it and then save



🡪Open the above above project from VS Code.



🡪Now click on terminal and select command prompt

A black background with white text

AI-generated content may be incorrect.

Step 2: Create the dockerfile in the project root directory(where our package.json is located)

Note: If already “dockerfile” is created ignore it and continue with the process

Dockerfile content as follows:

# ---------- Build stage ----------

FROM node:20-alpine AS builder

WORKDIR /app

COPY package\*.json ./

RUN npm ci

COPY . .

RUN npm run build

# ---------- Run stage ----------

FROM nginx:alpine

COPY nginx.conf /etc/nginx/conf.d/default.conf

COPY --from=builder /app/dist/ /usr/share/nginx/html/

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

=========================================

A screenshot of a computer program

AI-generated content may be incorrect.

🡪After creation close the ”dockerfile”

Step 3: Build and Run

🡪Now open the Docker Hub and “Docker-Desktop” and login into both.

In the vs code command prompt, execute the following commands:

$docker login docker.io //must get login success message

🡪Now to build the docker image by typing the following command

$docker build -t ecommerce-frontend .

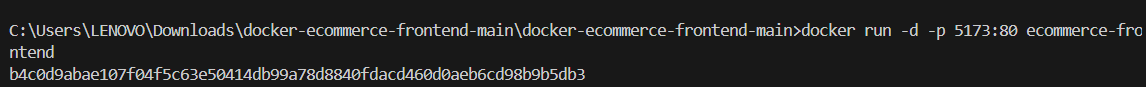
Note: at the end “.” Is compulsory

A screenshot of a computer screen

AI-generated content may be incorrect.

🡪After this, run this docker image by using the following command

$docker run -d -p 5173:80 ecommerce-frontend



It will show like this.

🡪Now open verify in the Docker-desktop, in the “image” section to see the created docker image.

A screenshot of a computer

AI-generated content may be incorrect.

🡪Now go to container section and click on the container URL / port number there.

A screenshot of a computer

AI-generated content may be incorrect.

Now you can see the output as follows:

A screenshot of a website

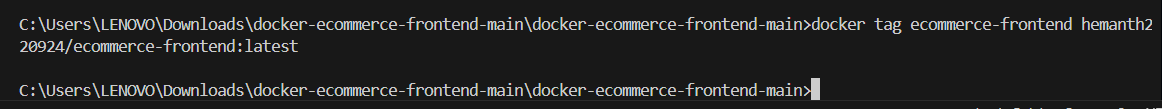
AI-generated content may be incorrect.

Step 4: 🡪Push the docker image into docker hub,

Type the following command to tag the docker image, before pushing.

$docker tag ecommerce-frontend user name/ecommerce-frontend:latest

Note: here replace it with your docker hub user name



🡪Now push it into the docker hub by the following command

$docker push xxxx/ecommerce-frontend:latest

Note : Replace xxxx with your docker hub username and run it

A screen shot of a computer program

AI-generated content may be incorrect.

It indicates that it is pushed successfully

🡪After that open Docker Hub in browser and refresh it

A screenshot of a computer

AI-generated content may be incorrect.

Here a pushed repository you can notice.

THE END …….