Program 1:
Develop a mu;ti-stage Dockerfile for container orchestration
31/10/2025

Project Directory:

|-build
|-dist
|-Dockerfile
|-package.json
|-package-lock.json
|-src
  |--index.js
|-node_modules

Program code and steps for execution:

Step 1: Open the mylab directory and create the prog2 directory navigate into that then do the initialize a new Node.js project also do the install of express

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab$ mkdir prog2
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab$ cd prog2/
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$
```

Step 2: After the installation write the Dockerfile

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ sudo nano Dockerfile
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ cat Dockerfile
### Start the multi-stage Dockerfile: build and production stages

# Stage 1: Build stage
FROM node:20-alpine AS builder

# Set working directory
WORKDIR /app

# Copy package files and install dependencies
COPY package.json package-lock.json ./
RUN npm install

# Copy application source code
COPY . .

# Build the application
RUN npm run build

# Stage 2: Production stage
FROM node:20-alpine

# Set working directory
WORKDIR /app

# Copy only necessary files from build stage
COPY --from=builder /app/package.json ./
COPY --from=builder /app/package-lock.json ./
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/node_modules ./node_modules

# Expose the application port
EXPOSE 3000

# Start the application
CMD ["node", "dist/index.js"]
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$
```

Step 3: After the initialization and installation the package.json file will be automatically added so make some changes to the package.json in order to look like this

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ npm init -y
Wrote to /home/chinmayi-m-h/mylab/prog2/package.json:

{
  "name": "prog2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}



(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ npm install expr

added 68 packages, and audited 69 packages in 2s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$
```

Step 4: Create another folder inside the prog2 as src where the index.js file will be create
Step 5: After creating the index.js add the below code inside the index.js

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ mkdir src
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ cd src/
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2/src$ sudo nano index.js
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2/src$ cat index.js
const express=require('express');
const app=express();
const port=3000;

app.get("/",(req,res)=>{
        res.send("Hello from multi-stage docker!!!");
});

app.listen(port,()=>{
        console.log(`Server running on http://localhost:${port}`);
});
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2/src$
```

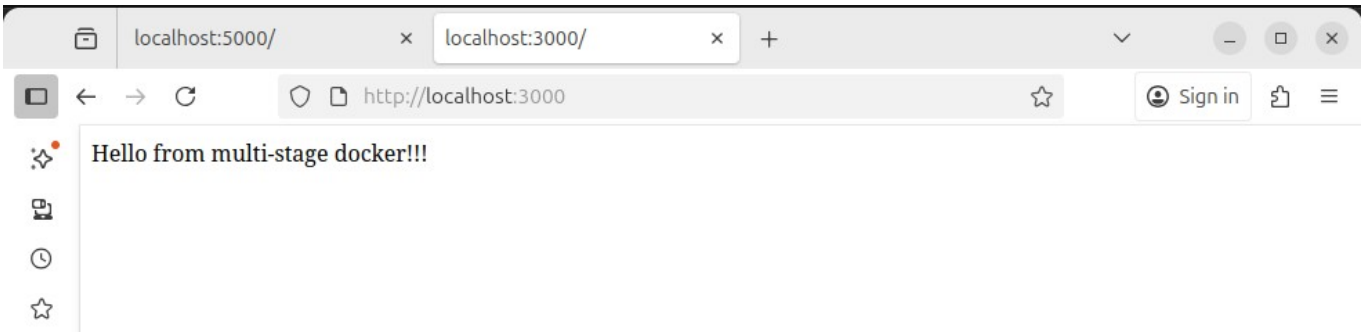
Step 6: Come out of the src folder
Step 7: Now build the docker for prog2

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2/src$ cd ..
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker build -t prog2 .
[+] Building 6.7s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 792B
 => [internal] load metadata for docker.io/library/node:20-alpine
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa39f0a770265435
 => [internal] load build context
 => => transferring context: 2.70MB
 => CACHED [builder 2/6] WORKDIR /app
 => [builder 3/6] COPY package.json package-lock.json ./
 => [builder 4/6] RUN npm install
 => [builder 5/6] COPY . .
 => [builder 6/6] RUN npm run build
 => [stage-1 3/6] COPY --from=builder /app/package.json ./
 => [stage-1 4/6] COPY --from=builder /app/package-lock.json ./
 => [stage-1 5/6] COPY --from=builder /app/dist ./dist
 => [stage-1 6/6] COPY --from=builder /app/node_modules ./node_modules
 => exporting to image
 => => exporting layers
 => => writing image sha256:fe9b99b6dc6a3b1742e144368866c54e25d908551d246331e808392742259b80
 => => naming to docker.io/library/prog2
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$
```

Step 8: Now run the docker of prog2 in interactive mode

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker run -it -p 3000:3000 prog2
Server running on http://localhost:3000
```

Step 9: Now in the browser search for http://localhost:3000



Step 10: Now stop and remove the container

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                       NAMES
8ac208075759   prog2     "docker-entrypoint.s…"   28 minutes ago   Up 28 minutes   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   vigilant_goldberg
```

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker stop vigilant_goldberg
vigilant_goldberg
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker rm vigilant_goldberg
vigilant_goldberg
```

Step 11: Remove the created image

```
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
prog2        latest    fe9b99b6dc6a   30 minutes ago   137MB
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker rmi prog2
Untagged: prog2:latest
Deleted: sha256:fe9b99b6dc6a3b1742e144368866c54e25d908551d246331e808392742259b80
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
(base) 1RV24MC030_chinmayi_m_h@chinmayi-m-h-HP-Pavilion-Laptop-15-eg3xxx:~/mylab/prog2$
```