Program 2:

Develop a multi-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

Step 2: After the installation write the Dockerfile

```
FROM node:20-alpine AS builder

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .
RUN npm run build

FROM node:20-alpine

WORKDIR /app

COPY --from=builder /app/package*.json ./
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/node_modules ./node_modules

EXPOSE 5000

CMD ["node", "dist/index.js"]
```

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

```
{
  "name": "program2",
  "version": "1.0.0",
  "main": "dist/index.js",
  "scripts": {
    "start": "node dist/index.js",
    "build": "mkdir -p dist && cp -r src/* dist/"
  },
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

Step 4: Create another folder inside the prog2 as src where the index.js file will be create

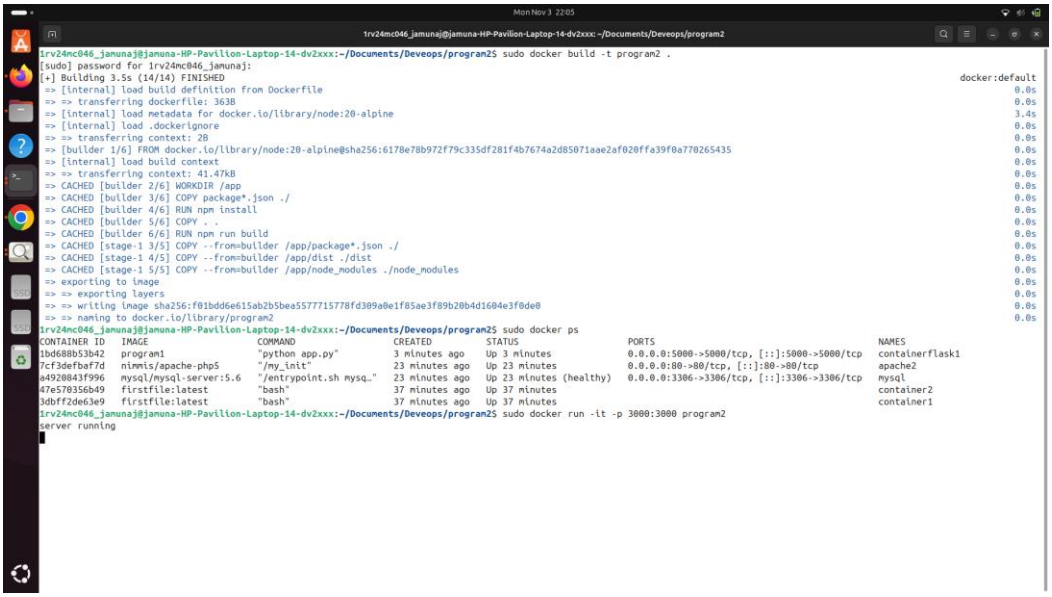
Step 5: After creating the index.js add the below code inside the index.js

```
const express = require('express');
const app=express();
const PORT=5000;

app.get('/',(req,res)=>{
        res.send('hello ');
});

app.listen(PORT,()=>{
        console.log('server running');
});
```

Step 6: Come out of the src folder

Step 7: Now build the docker for prog2

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



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