Develop a Multi-Stage Dockerfile for Container Orchestration

Step 1 - Create the following directory structure

```
1rv24mc039_hardikmehta@hardik-Aspire-E5-573:~/Devops/pgrm2$ ls
build  dist  Dockerfile  node_modules  package.json  package-lock.json  src
```

Step 2 - cd into the folder and run the command in the terminal

$npm init -y

$ npm install express

Step 3 - create the index.js file

```
const exp=require("express");
const app=exp();
const port=4000;

app.listen(port,()=>{
   console.log("server is running on port: ",port);
});

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

Step 4 - update the package.json file in the script

```
"scripts": {
   "start":"node index.js",
    "build":"mkdir -p dist && cp -r src/* dist/"
 },
```

Step 5 - Write the Dockerfile

```
1rv24mc039_hardikmehta@hardik-Aspire-E5-573:~/Devops/pgrm2$ cat Dockerfile
#program 2
#stage 1
#1 creating the node image
FROM node:20-alpine AS builder
#2 creating the working directory
WORKDIR /app
#3 copy the package file and run the command
COPY package*json ./
RUN npm install
#copy the other files
COPY . .
#run the command
RUN npm run build

#stage2
FROM node:20-alpine

WORKDIR /app
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 4000

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

Step 6 - Build the image by running

pgrm2$ docker build -t program2 .

Step 7 - Create and run a new container of the image by running

pgrm2$ docker run -d -p 4000:4000 —name program2-container program1

Step 8 - Verify output

Open the browser

Search :- http://localhost:3000

See the output as :

multi-stage docker file