DEVOPS PROGRAM 2:

Develop a Multi-Stage Dockerfile for Container Orchestration.

PROJECT STRUCTURE:

```
   program2/
—Dockerfile(without any extension)
—src/|
        |index.js
—package.json
—package-lock.json
—node_modules(created during npm install)
```

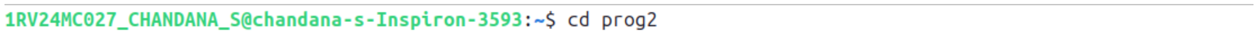
Step 1: Create Project Directory

mkdir prog2

Creates a new directory named program2

Step 2: Navigate to the Project Directory

cd prog2



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ cd prog2
```

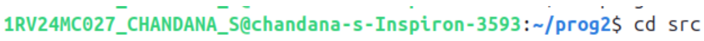
Changes the working directory to program1, where all files will be created.

Step 3: Create sub Project Directory

mkdir src

Navigate to sub directory

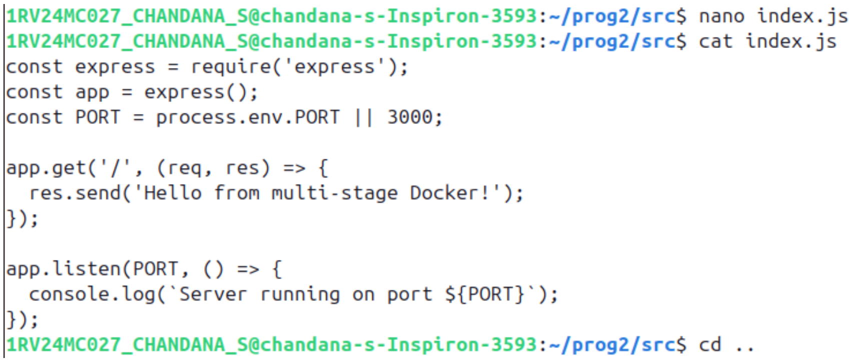
cd src



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ cd src
```

Step 4: Create and Edit index.js

nano index.js

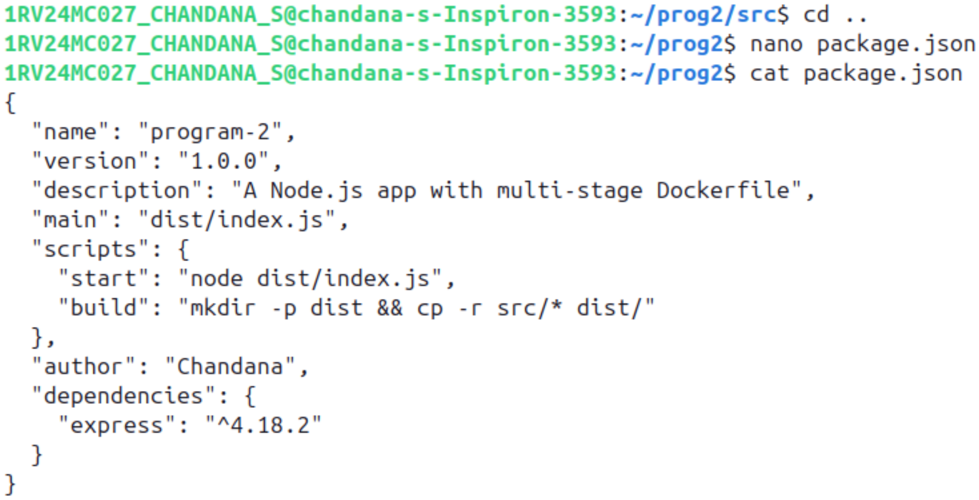


```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2/src$ nano index.js
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2/src$ cat index.js
const express = require('express');
const app = express();
const PORT = process.env.PORT || 3000;

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

app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2/src$ cd ..
```

cd ..

Step 5: Create and Edit package.json

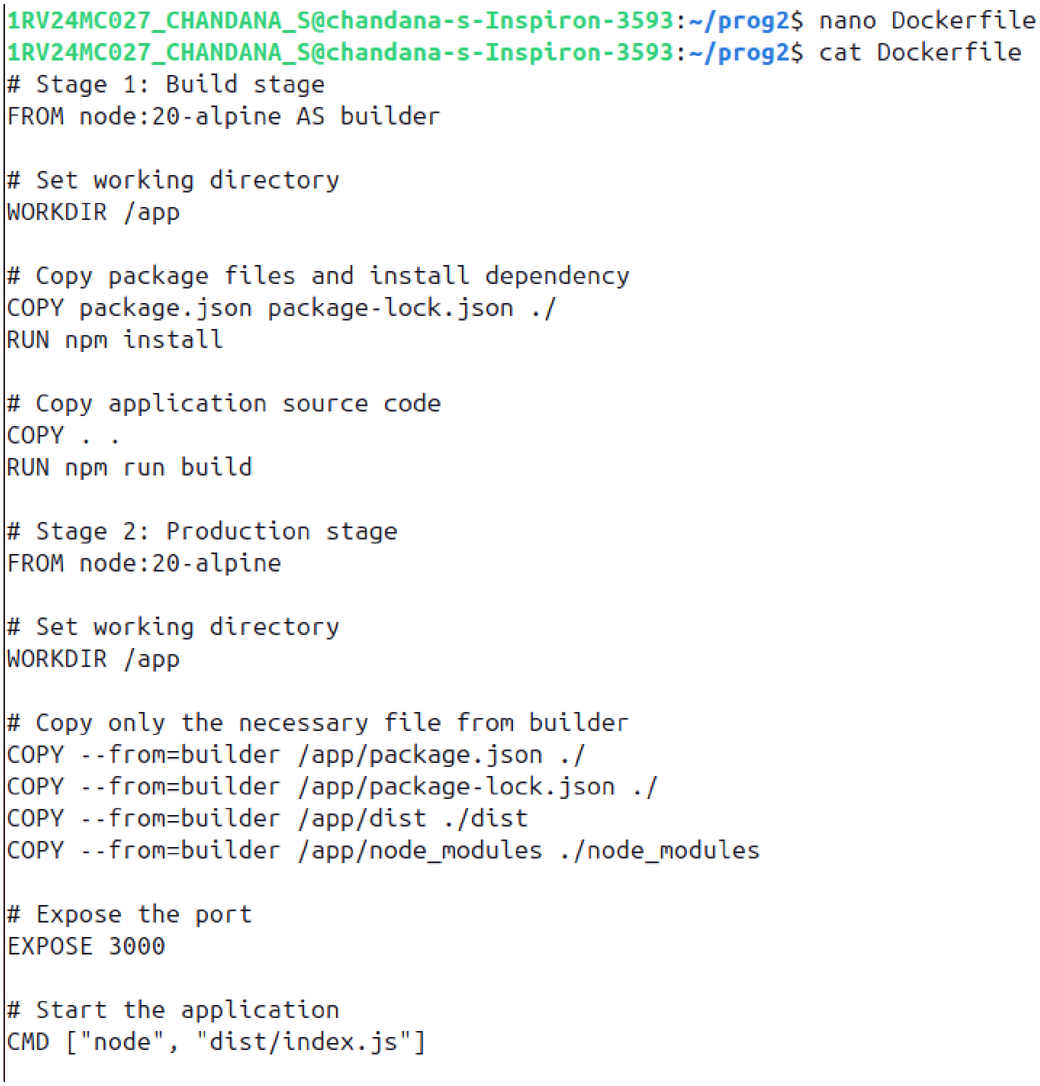
nano package .json or run npm init -y



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2/src$ cd ..
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ nano package.json
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ cat package.json
{
  "name": "program-2",
  "version": "1.0.0",
  "description": "A Node.js app with multi-stage Dockerfile",
  "main": "dist/index.js",
  "scripts": {
    "start": "node dist/index.js",
    "build": "mkdir -p dist && cp -r src/* dist/"
  },
  "author": "Chandana",
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

Step 6: Create and Edit Dockerfile

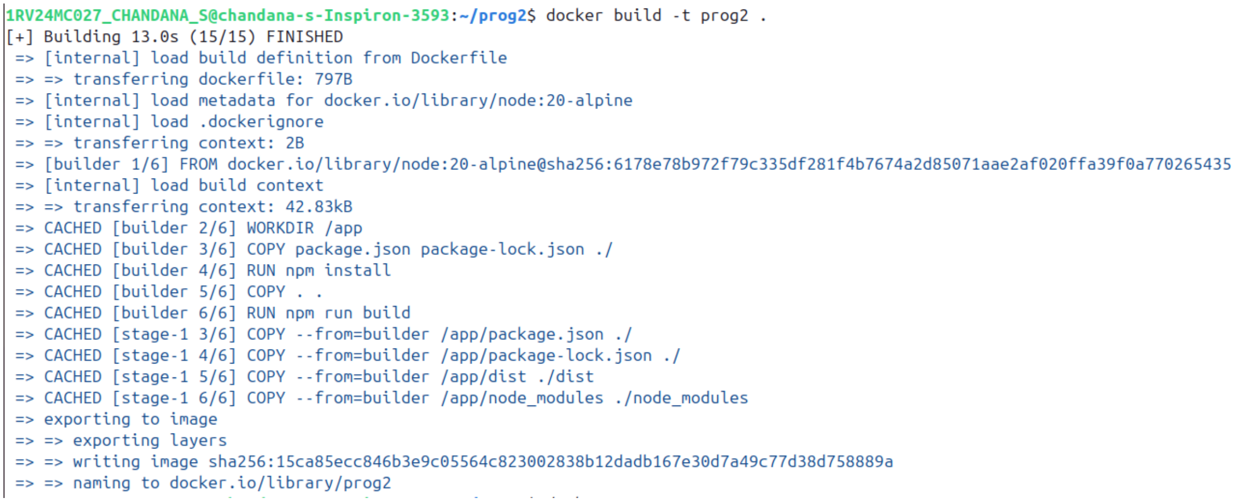
nano Dockerfile



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ nano Dockerfile
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ cat Dockerfile
# Stage 1: Build stage
FROM node:20-alpine AS builder

# Set working directory
WORKDIR /app

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

# Copy application source code
COPY . .
RUN npm run build

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

# Set working directory
WORKDIR /app

# Copy only the necessary file from builder
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 port
EXPOSE 3000

# Start the application
CMD ["node", "dist/index.js"]
```

Step 7: Build the Docker Image

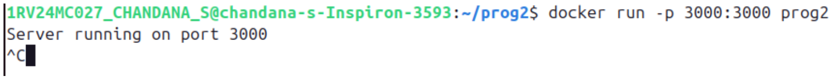
docker build -t prog2 .



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ docker build -t prog2 .
[+] Building 13.0s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 797B
 => [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: 42.83kB
 => CACHED [builder 2/6] WORKDIR /app
 => CACHED [builder 3/6] COPY package.json package-lock.json ./
 => CACHED [builder 4/6] RUN npm install
 => CACHED [builder 5/6] COPY . .
 => CACHED [builder 6/6] RUN npm run build
 => CACHED [stage-1 3/6] COPY --from=builder /app/package.json ./
 => CACHED [stage-1 4/6] COPY --from=builder /app/package-lock.json ./
 => CACHED [stage-1 5/6] COPY --from=builder /app/dist ./dist
 => CACHED [stage-1 6/6] COPY --from=builder /app/node_modules ./node_modules
 => exporting to image
 => => exporting layers
 => => writing image sha256:15ca85ecc846b3e9c05564c823002838b12dadb167e30d7a49c77d38d758889a
 => => naming to docker.io/library/prog2
```

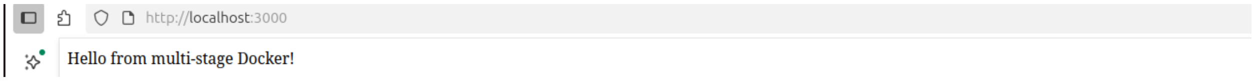
Step 8: Run the Docker Container

docker run -p 3000:3000 prog2



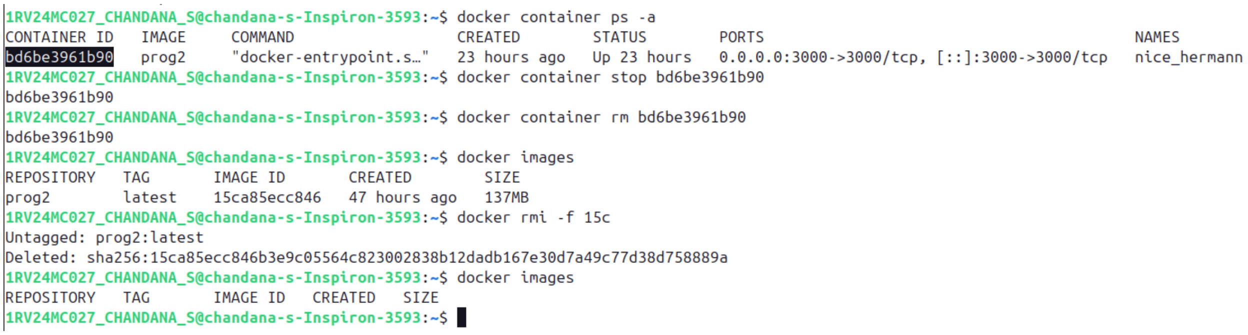
```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog2$ docker run -p 3000:3000 prog2
Server running on port 3000
^C
```

Step 9:Check the output



Hello from multi-stage Docker!

Step 10: Verify Running Container,stop the container,remove the container



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker container ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                         NAMES
bd6be3961b90   prog2     "docker-entrypoint.s…"   23 hours ago   Up 23 hours   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   nice_hermann
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker container stop bd6be3961b90
bd6be3961b90
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker container rm bd6be3961b90
bd6be3961b90
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
prog2        latest    15ca85ecc846   47 hours ago   137MB
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker rmi -f 15c
Untagged: prog2:latest
Deleted: sha256:15ca85ecc846b3e9c05564c823002838b12dadb167e30d7a49c77d38d758889a
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~$ 
```