Task 6

Python Hello World
Use the official Python alpine image
Create a simple Python script that prints "Hello Docker!"
Make the script run when the container starts
12:09 pm


```
  GNU nano 7.2
FROM python:3.11-alpine

WORKDIR /DII

COPY index.py .

CMD ["python", "index.py"]
```

Task 7

Multi-stage Build
Create a Dockerfile that:
Uses node image to build a React application
Then copies just the built static files to an nginx image
1:32 pm


```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.
Need to get 47.1 kB of archives.
After this operation, 111 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 tree amd64 2.1.1-2ubuntu3 [47.1 kB]
Fetched 47.1 kB in 1s (47.2 kB/s)
Selecting previously unselected package tree.
(Reading database ... 211230 files and directories currently installed.)
Preparing to unpack .../tree_2.1.1-2ubuntu3_amd64.deb ...
Unpacking tree (2.1.1-2ubuntu3) ...
Setting up tree (2.1.1-2ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
reem@reem-host:~/Docker$ tree
.
├── Dockerfile
├── package.json
├── public
│   └── index.html
└── src
    ├── App.js
    └── index.js

3 directories, 5 files
reem@reem-host:~/Docker$
```

reem@reem-host: ~/Docker

```
# Stage 1: Build React app
FROM node:18 AS builder

# Set working directory
WORKDIR /app

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

# Copy app source
COPY . .

# Build the React app
RUN npm run build

# Stage 2: Serve app with nginx
FROM nginx:alpine

# Copy build output to nginx html folder
COPY --from=builder /app/build /usr/share/nginx/html

# Expose port 80
EXPOSE 80

# Start nginx
CMD ["nginx", "-g", "daemon off;"]
~
~
~
"Dockerfile" 27 lines, 466 bytes
```

localhost:8080/

localhost:8080

It works!