**Running python application using Docker command:**

Vi dockerfile

#choosing the base image:

FROM python:3.8-alpine

#choosing working directory for the application:

WORKDIR /test

#copying the requirements.txt file to app directory and installing packages:

#COPY requirements.txt .

#RUN pip install --no-cache-dir -r requirements.txt→ no cache dir instructs pip not to use cached copies of packages

#copying the rest of application code to the working directory:

COPY . .

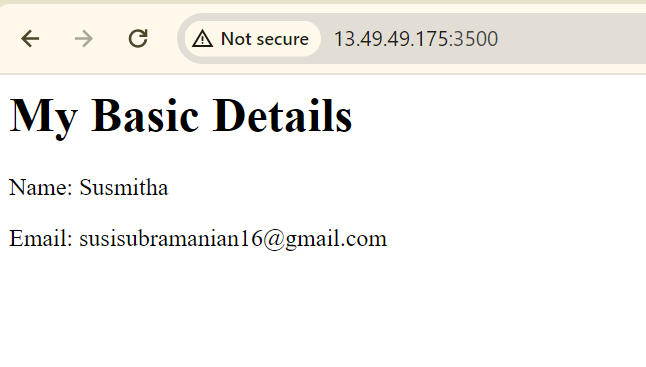
RUN pip3 install -r requirements.txt

#exposing the application:

#EXPOSE 5000

#Executing the application after creating image:

CMD ["python3", "app.py"]





**Running nodejs application using docker file:**

Vi dockerfile

#choosing the base image:

FROM node:16-alpine

#choosing working directory for the application:

WORKDIR /test

#copying the package.json file to app directory and installing packages:

COPY package.json .

RUN npm install

RUN npm install react-scripts —>React Scripts is a set of tools and configurations included with Create React App (CRA), a popular tool for rapidly setting up a new single-page React application.

#copying the rest of application code to the working directory:

COPY . .

#building the application:

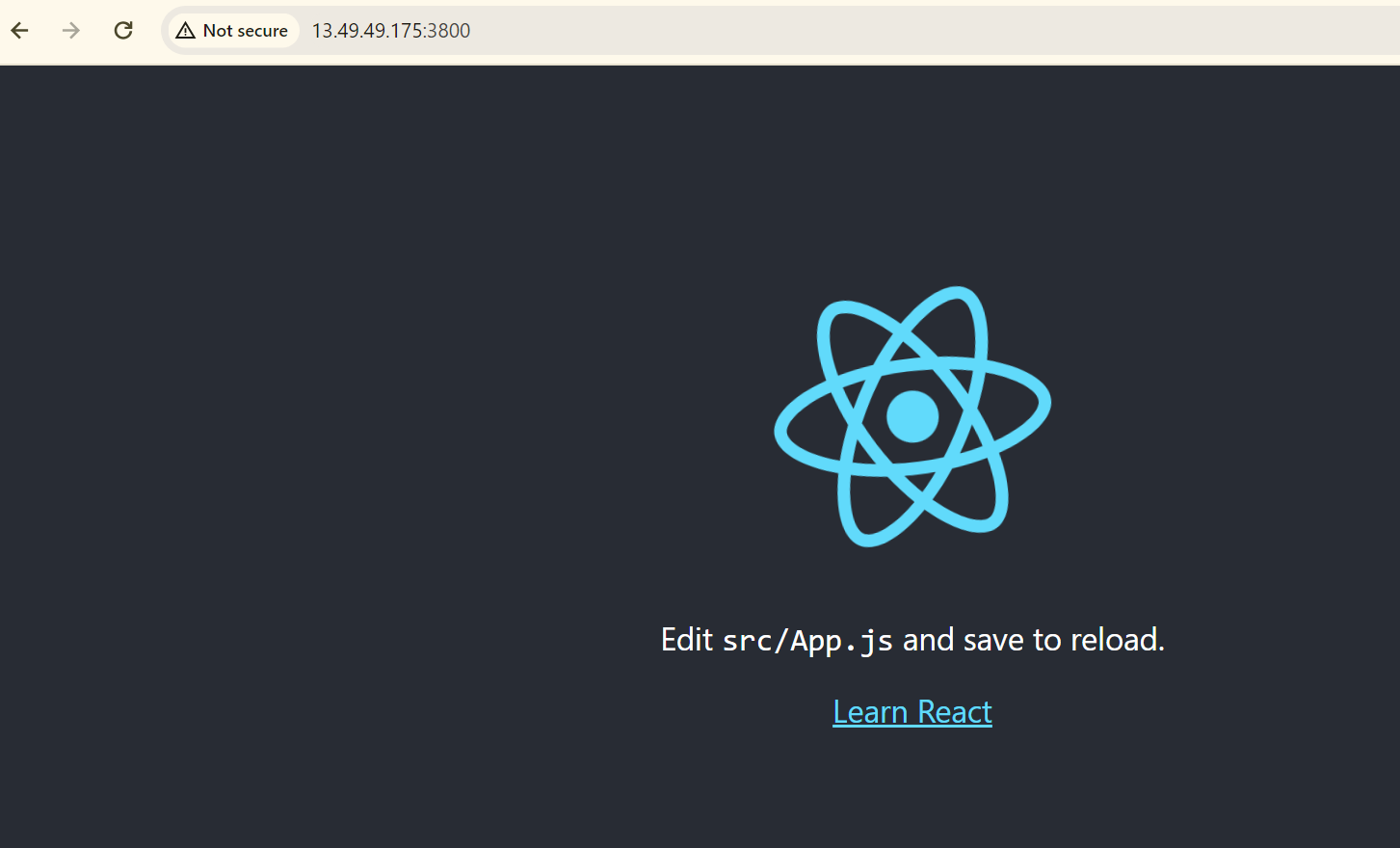
RUN npm run build →npm locates the build script within your project's package.json file and executes the commands listed within that script

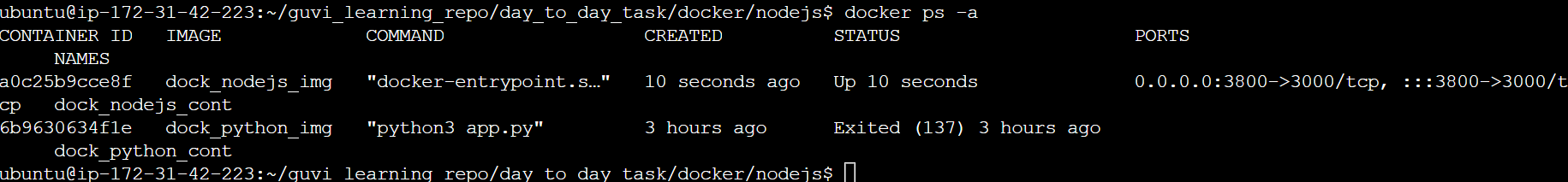
#exposing the application:

EXPOSE 3000

#Executing the application after creating image:

CMD ["npm", "start"]





**Running Java application using Docker:**

Vi dockerfile

# selecting java-17 as the base image:

FROM openjdk:17-slim

# Setting the working directory

WORKDIR /app

# Copy the JAR file into the container

COPY target/spring-boot-docker.jar .

# Expose the application to visible on the browser:

EXPOSE 8080

# Command to run the application

CMD ["java", "-jar", "spring-boot-docker.jar"]

