# Creating Docker Images for LMS Application.

## Step 1: Configuring Postgres Database

Created a Postgres container directly without specifying the password so the container was unable to start so we can check / trouble shoot the container using the command :

A screenshot of a computer screen

Description automatically generated

The actual command for setting up postgres database container is:

Docker container run -dt –name **“container\_name”** -e POSTGRES\_PASSWORD=**”PASSWORD”** postgres

**A computer screen with white text and numbers

Description automatically generated**

## Step 2: Building Docker Image for LMS BACKEND:

To build the we need to install node js

Command to create a node container is :  
docker container run -dt - - name n1 node:16

A screenshot of a computer screen

Description automatically generated

Once the node js and npm is installed then need to connect the backend “api” to database by providing credentials of database into .env file.

A screenshot of a computer

Description automatically generated

Now we can create the docker file and write the commands to create a separate directry

1. mkdir to create a directry
2. wkdir to make it as present work directry
3. need to copy all the files from backend directry such as file present in api into the backend directry in the container
4. npm install command is used to install all the node modules which are present package.json file
5. npm prisma db push : is used to insert the tables data into the database
6. npm run build used to build the “**Build”** folder in backend directry
7. we can expose port 8080 or any port so that the application can be started in that port command for exposing the port is **Expose “port\_number”**
8. command for building the file is build/index.js to write the same command in Docker file 🡪 **CMD [“node”,”build/index.js”]**

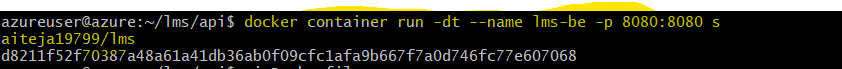
so the final Docker file is below:  
A screen shot of a computer

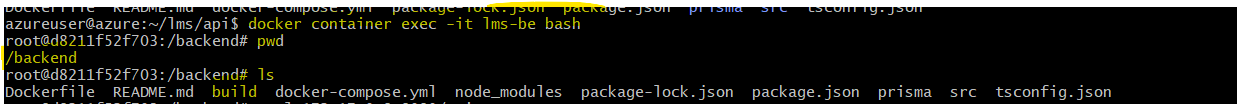
Description automatically generated

building the image from the docker file

A screen shot of a computer

Description automatically generated

Creating a container from the image:  




A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

So by this the container is created using image and tested that the backend is building perfectly.

## Step 3: Building Docker Image for LMS FRONTEND:

for the frontend the files will be available in webapp folder

First need to connect the front end to back end by giving details of backend into the front end env file

In the webapp folder need to create a docker file

Here we need to give instructions to create folder front end and make it as work directry and then install the node module 🡪 then build so that **dist** folder will be created.

Need to transfer the files present in dist folder into nginx folder /usr/share/nginx/html

So that the application would be running on port 80.

Now we can create a container which runs on port 80 then the front end application can be accessed by the user.

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A computer on a table

Description automatically generated

So we have created images for the frontend end and backend so we can use these images to build the containers and build the application within a minute.

A screenshot of a computer

Description automatically generated

The images were pushed into docker hub so now we can run the application in any machine by creating the containers using the images in docker hub.