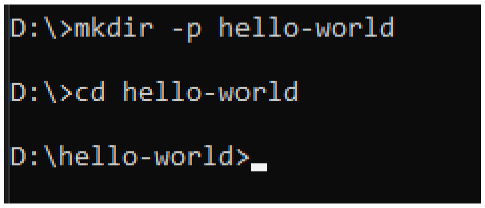
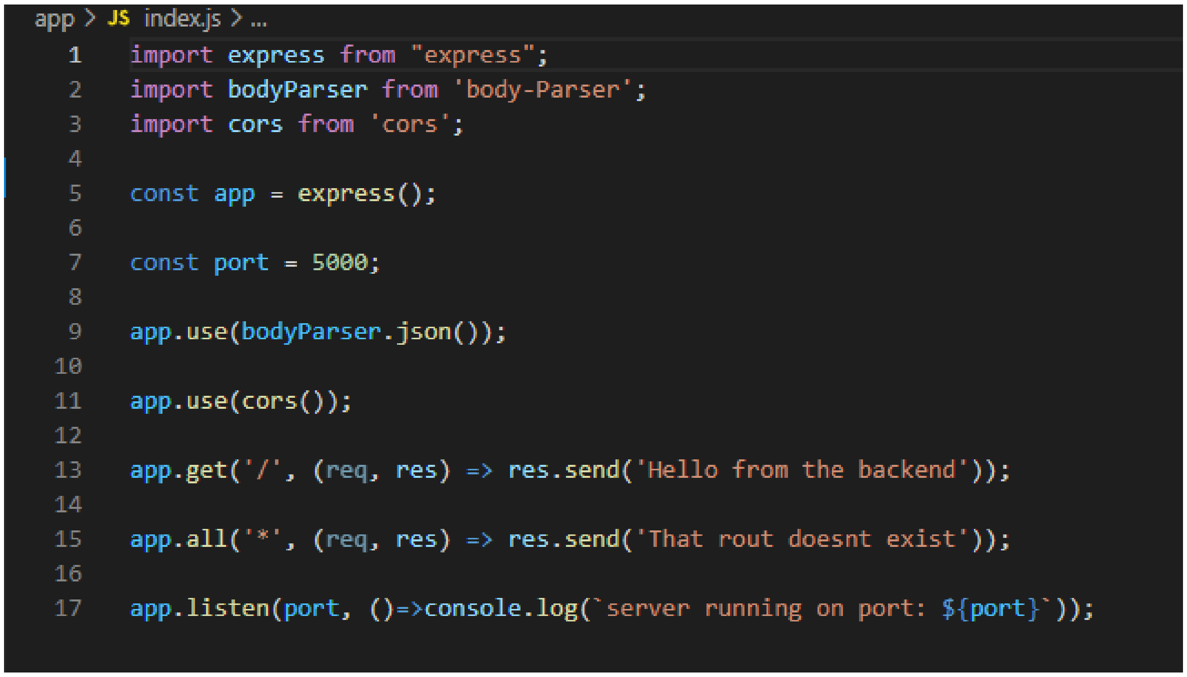
Docker Homework

1. Create folder for the Docker file



```
D:\>mkdir -p hello-world

D:\>cd hello-world

D:\hello-world>
```

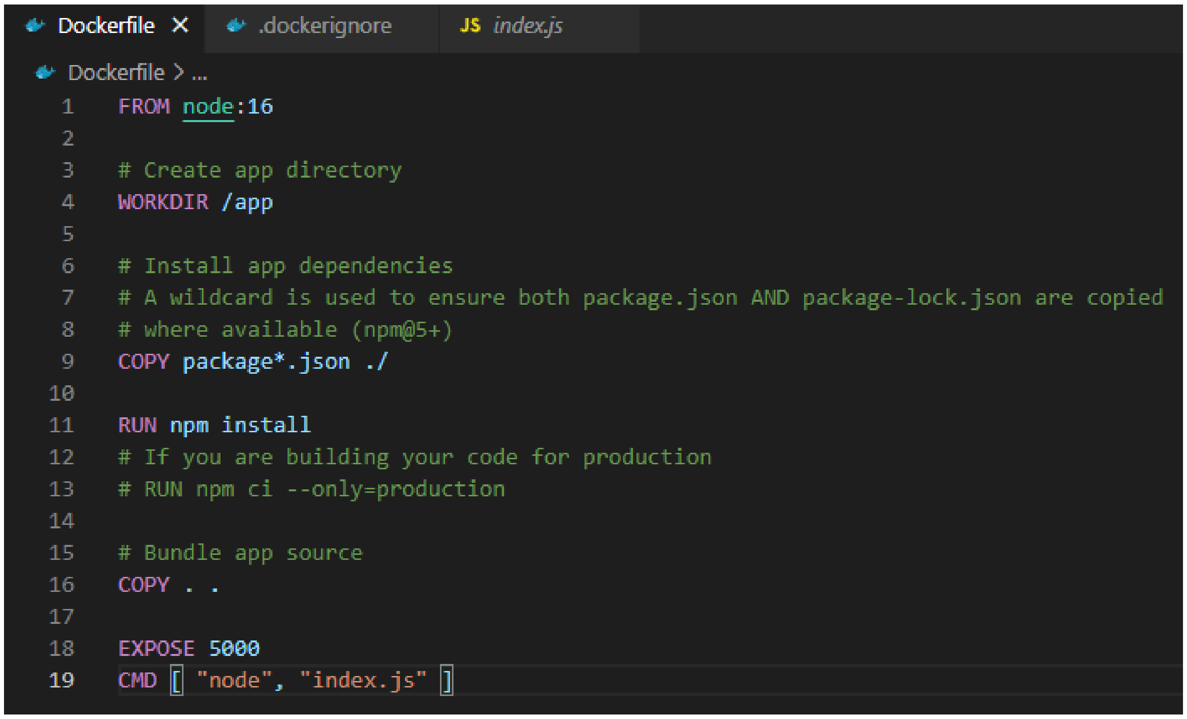
2. Create an app with “node js”



app > JS index.js > ...

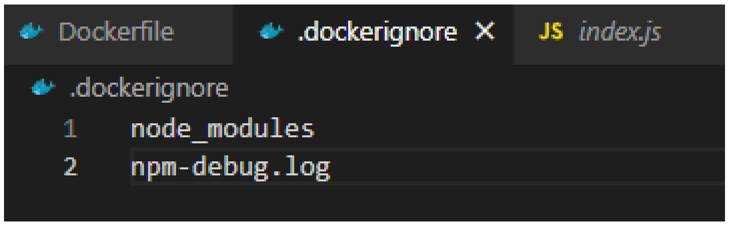
```js
import express from "express";
import bodyParser from 'body-Parser';
import cors from 'cors';

const app = express();

const port = 5000;

app.use(bodyParser.json());

app.use(cors());

app.get('/', (req, res) => res.send('Hello from the backend'));

app.all('*', (req, res) => res.send('That rout doesnt exist'));

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

3. Create a Dockerfile (for node js 16)



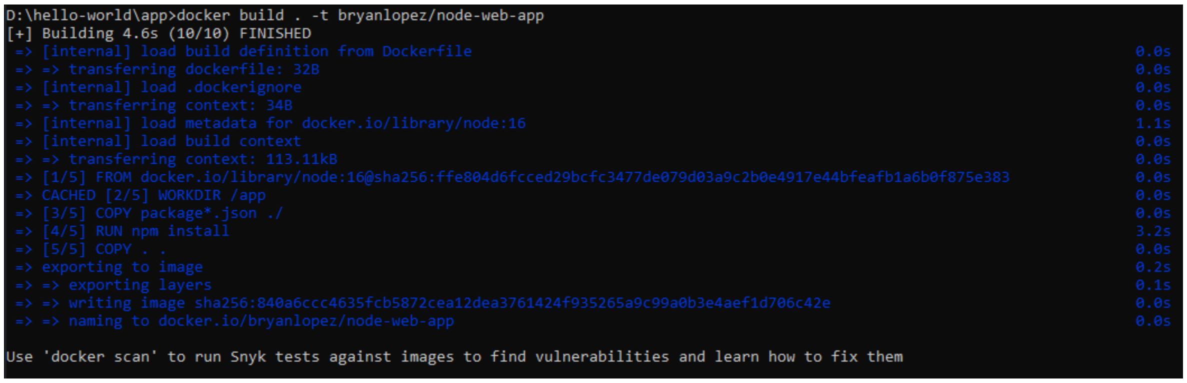
```
FROM node:16

# Create app directory
WORKDIR /app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 5000
CMD [ "node", "index.js" ]
```

4. Create a Dockerignore file in the same directory of the dockerfile



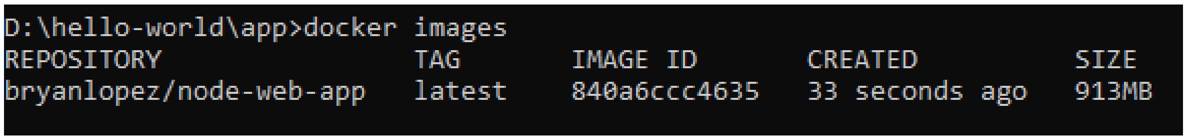
```
node_modules
npm-debug.log
```

5. Building our image



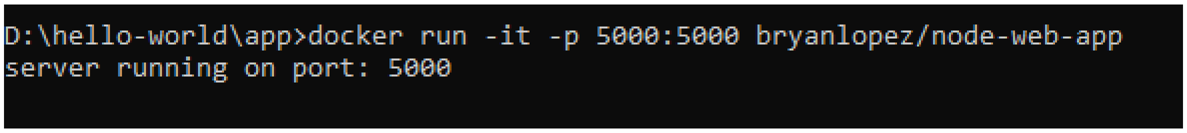
```
D:\hello-world\app>docker build . -t bryanlopez/node-web-app
[+] Building 4.6s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 32B                                                                            0.0s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 34B                                                                               0.0s
 => [internal] load metadata for docker.io/library/node:16                                                     1.1s
 => [internal] load build context                                                                              0.0s
 => => transferring context: 113.11kB                                                                          0.0s
 => [1/5] FROM docker.io/library/node:16@sha256:ffe804d6fcced29bcfc3477de079d03a9c2b0e4917e44bfeafb1a6b0f875e383  0.0s
 => CACHED [2/5] WORKDIR /app                                                                                  0.0s
 => [3/5] COPY package*.json ./                                                                                0.0s
 => [4/5] RUN npm install                                                                                      3.2s
 => [5/5] COPY . .                                                                                             0.0s
 => exporting to image                                                                                         0.2s
 => => exporting layers                                                                                        0.1s
 => => writing image sha256:840a6ccc4635fcb5872cea12dea3761424f935265a9c99a0b3e4aef1d706c42e                   0.0s
 => => naming to docker.io/bryanlopez/node-web-app                                                             0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

6. Now let¿s check the image:



```
D:\hello-world\app>docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
bryanlopez/node-web-app   latest    840a6ccc4635   33 seconds ago   913MB
```

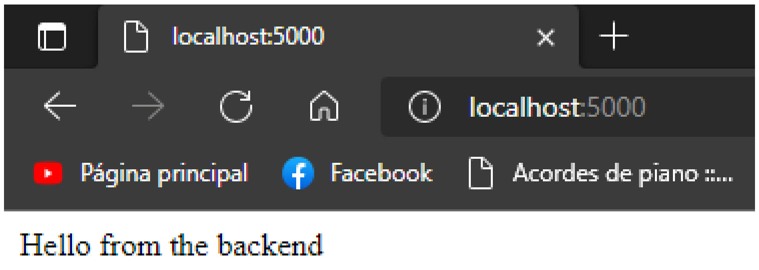
7. And finally run the image:



```
D:\hello-world\app>docker run -it -p 5000:5000 bryanlopez/node-web-app
server running on port: 5000
```

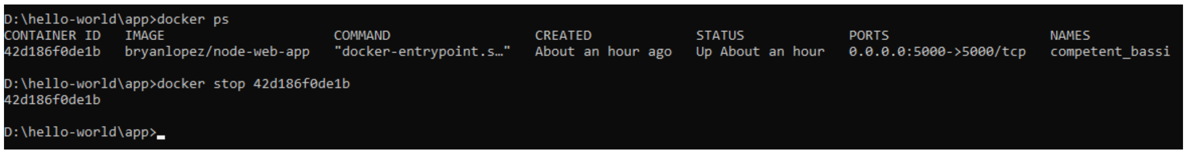
8.

9. As we can see, we successfully recievied a response



Hello from the backend

10. And our server is running



```
D:\hello-world\app>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED            STATUS             PORTS                    NAMES
42d186f0de1b   bryanlopez/node-web-app   "docker-entrypoint.s…"   About an hour ago   Up About an hour   0.0.0.0:5000->5000/tcp   competent_bassi

D:\hello-world\app>docker stop 42d186f0de1b
42d186f0de1b

D:\hello-world\app>
```

11.
12. Then stop the container