Roberto Morassi del Blanco

A0265239X

https://github.com/rmorassi/CS3219-OTOT-A1

Task A1.1

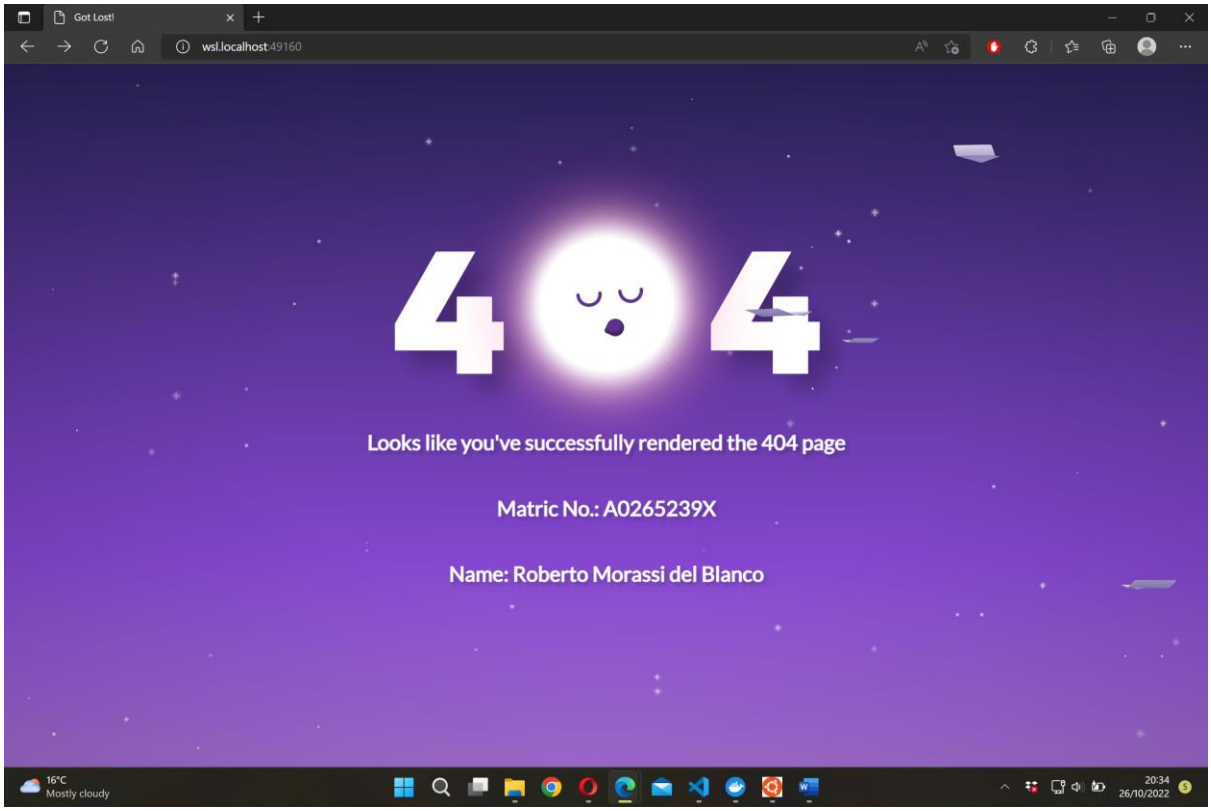
Relevant steps I followed to dockerize the app

1. Created a Dockerfile
    a. Configured the Dockerfile to use the relevant files and dependencies
    b. Configured the ports to open (8080)
    c. Configured the command that should be run on start-up
2. Created .dockerignore file and added the relevant directories and files
3. Built the docker image using “docker build” and tagged my image with “cs3219-a1.1”

Relevant steps to run the app as container

1. As I have already created the Dockerfile and .dockerignore file, these shouldn’t be created again
2. Run “docker build . -t cs3219-a1.1” in bash to build the docker image
3. Run “docker run -p 49160:8080 -d cs3219-a1.1” in bash to start the docker image in the background
4. The app will be running on port 49160

Screenshot proof of localhost showing the webpage



Task A1.2

Relevant steps I followed to dockerize the app

1. Created a Dockerfile
    a. Configured the Dockerfile to create a directory to house all the files
    b. Configured nginx by replacing the default configuration file with my own
    c. Configured the Dockerfile to use the relevant files
    d. Configured the ports to open (8080)
2. Created .dockerignore file and added the relevant directories and files
3. Built the docker image using “docker build” and tagged my image with “cs3219-a1.2”

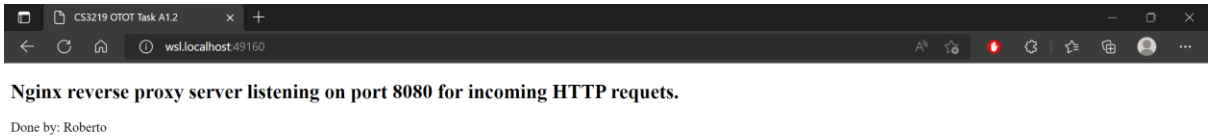
Relevant steps to run reverse proxy

1. For the proof of concept, I placed all files in a folder called “hidden” in the docker image to prove the reverse proxy
2. In the configuration of ngnix I configured the server with the open port of 8080 to point to another server on port 5000
3. I configured this secondary server on port 5000 to fetch files from the hidden folder. This returns index.html .

Relevant steps to run the app as container

1. As I have already created the Dockerfile and .dockerignore file, these shouldn't be created again
2. Run "docker build . -t cs3219-a1.2" in bash to build the docker image
3. Run "docker run -p 49160:8080 -d cs3219-a1.2" in bash to start the docker image in the background
4. The app will be running on port 49160

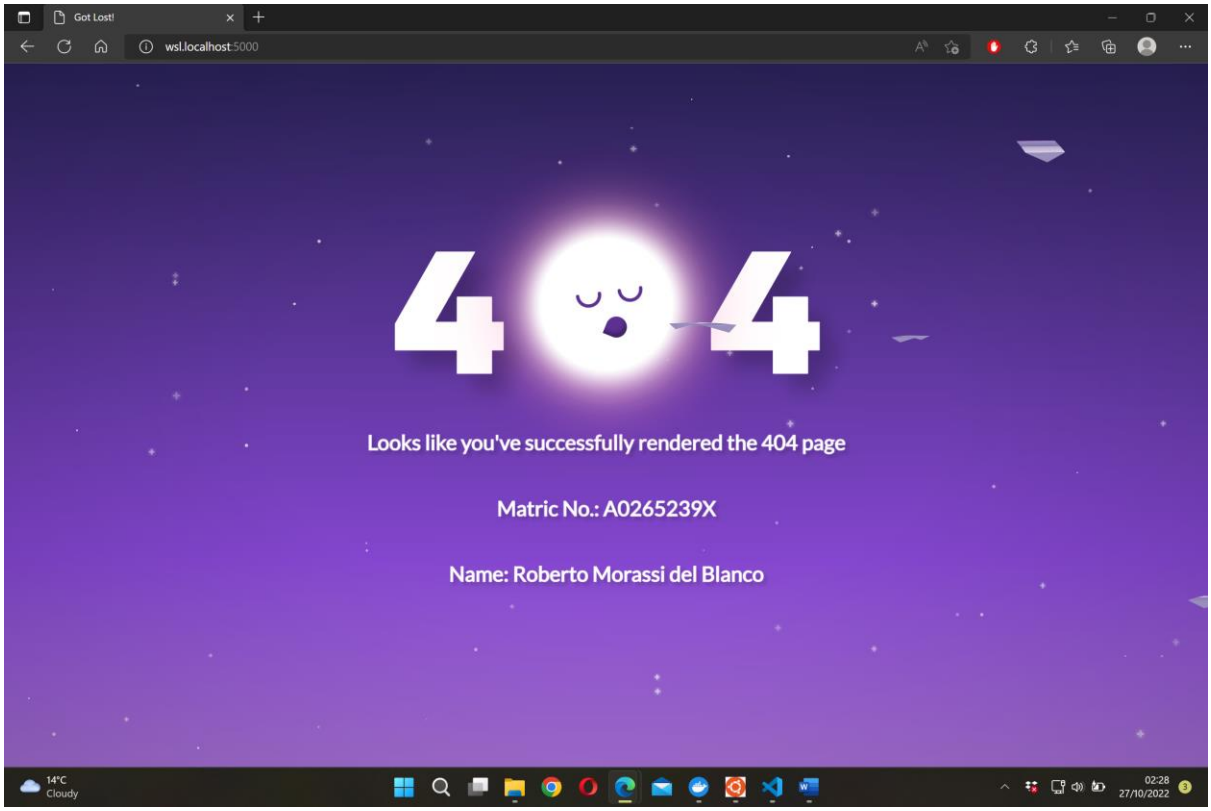
Screenshot proof of localhost showing the webpage



Task A1.3

To run the docker compose, run "docker-compose up --build" from the base directory. The nginx server with reverse proxy (into port 8080 of the app) will be running on port 5000.

Proof of completion:


Got Lost!
wsl.localhost:5000
4 4
Looks like you've successfully rendered the 404 page
Matric No.: A0265239X
Name: Roberto Morassi del Blanco
14°C
Cloudy
02:28
27/10/2022