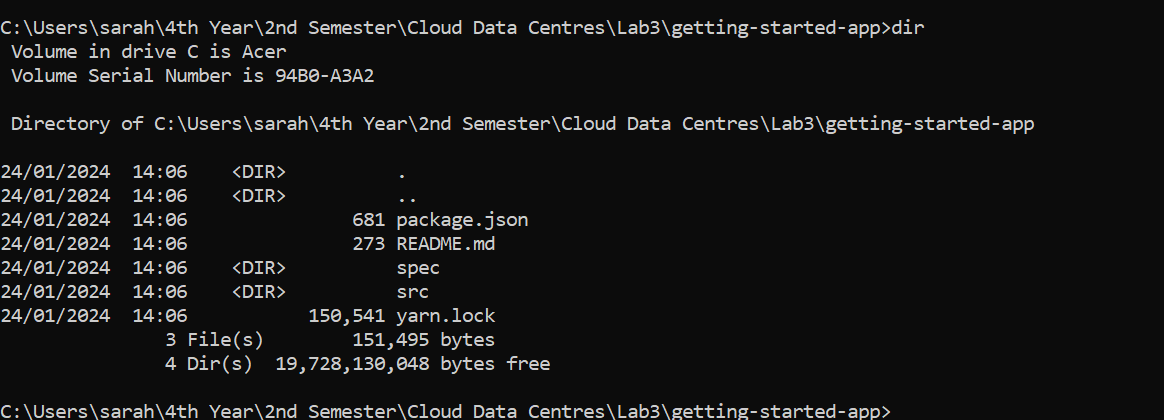
# **Cloud Data Centres: Lab 3**

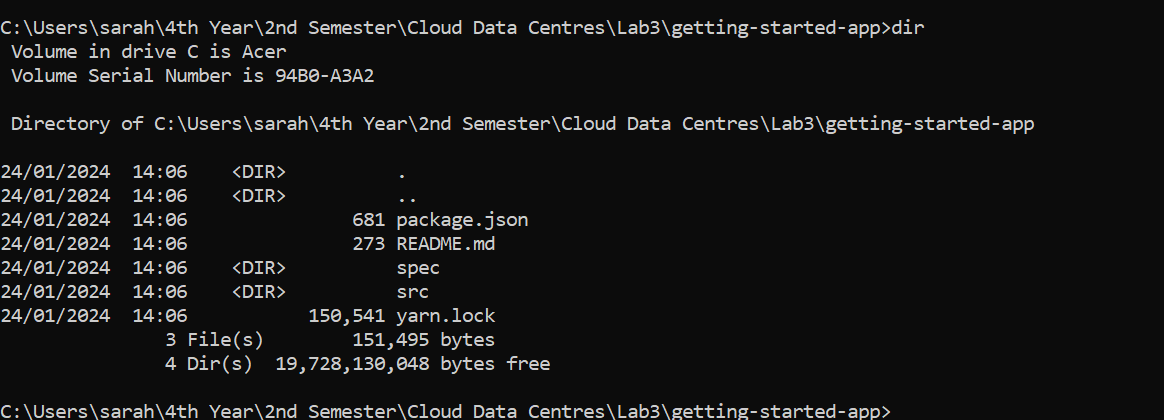
**By Sarah Martin**

**Student ID: C00257967**

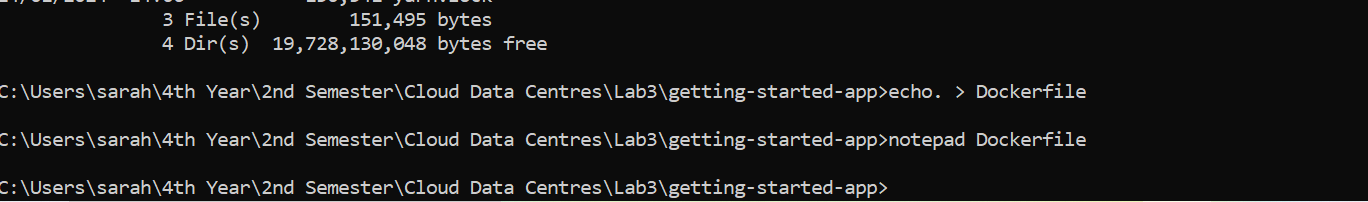
# **Part 2: Containerize An Application**

I started off by cloning a git repository from https://docs.docker.com/get-started/ into a folder called lab 3 

I then looked at the contents of the getting-started-app with this command



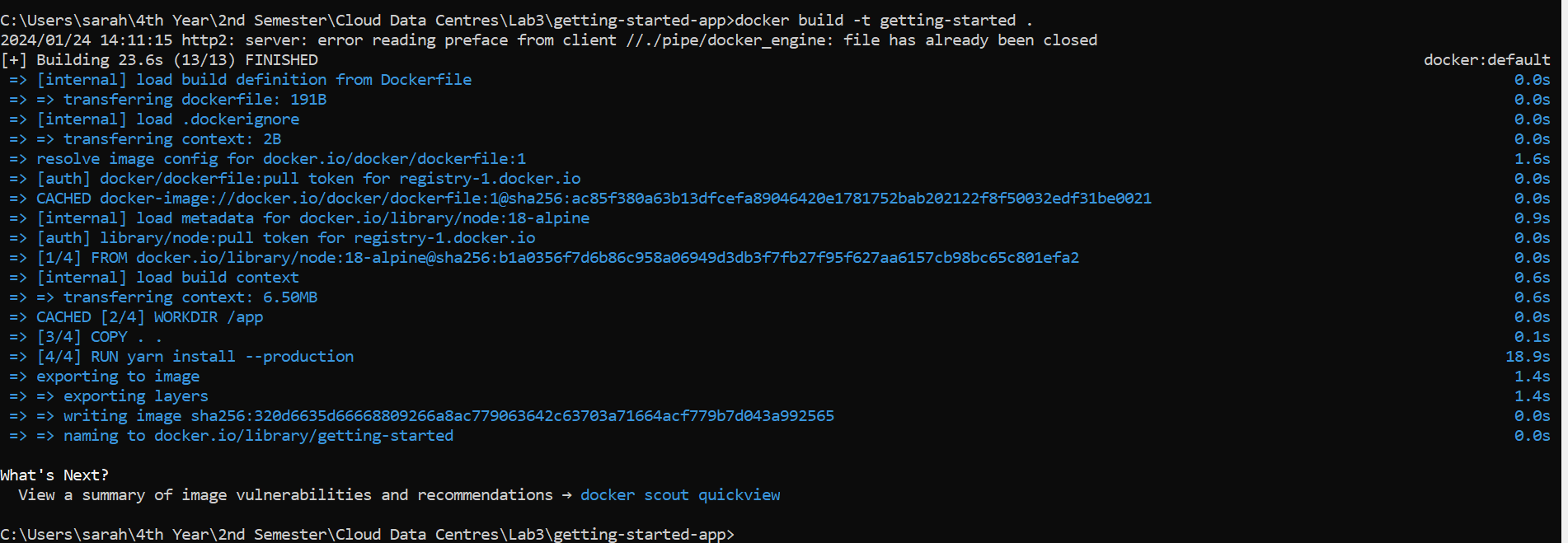
I created an empty Dockerfile and I opened it using Notepad.



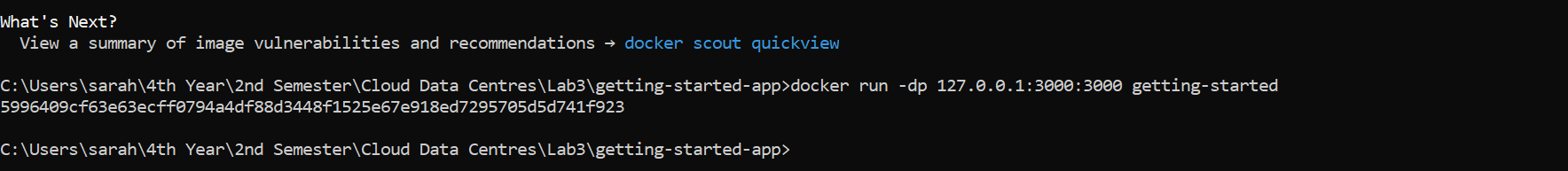
I added the following content to the Docker file.A screen shot of a computer

Description automatically generatedrfile

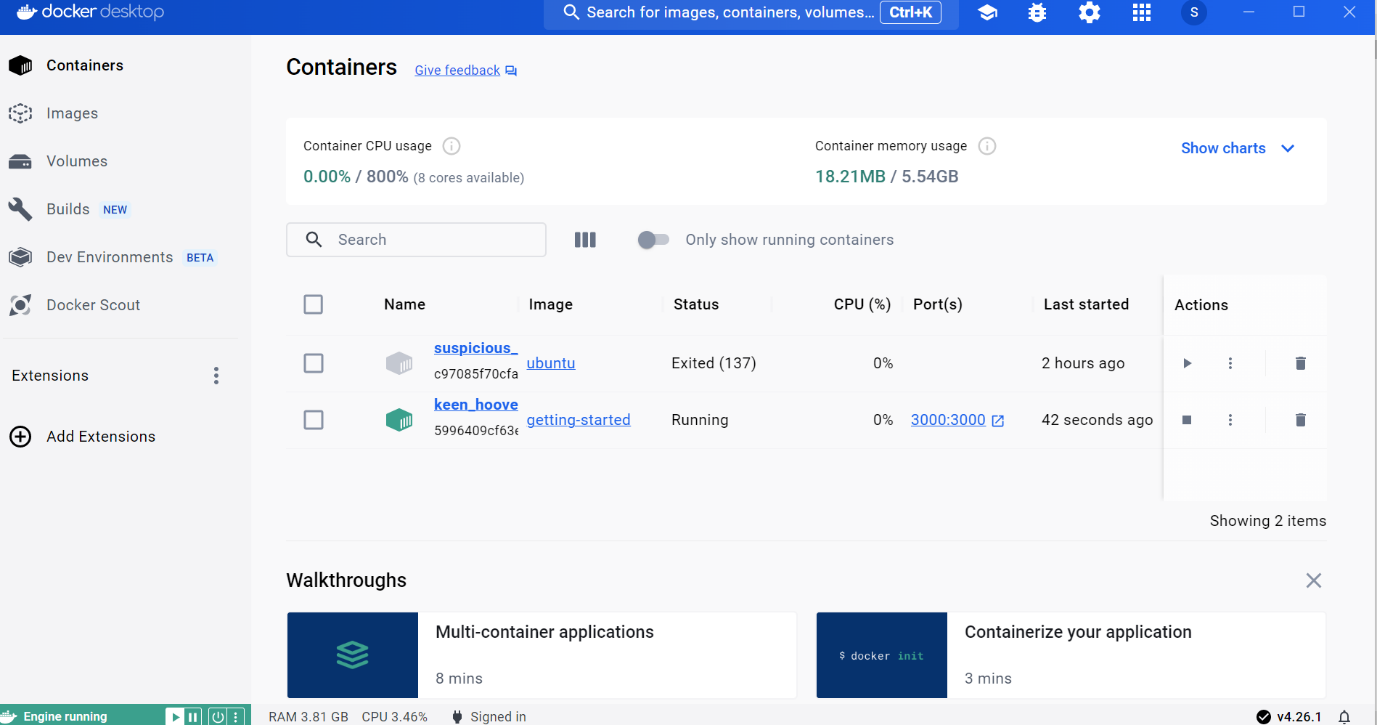
I then used this command to build the image.



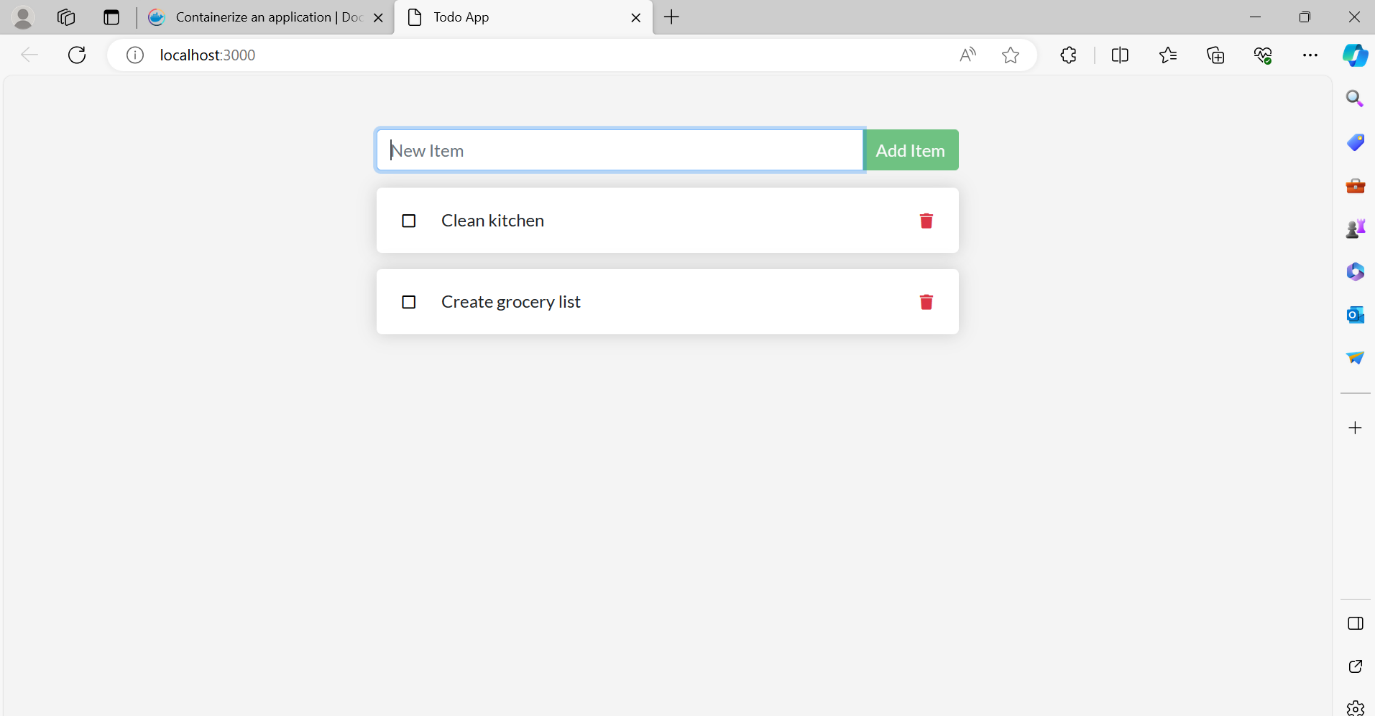
I then ran the container using this command.



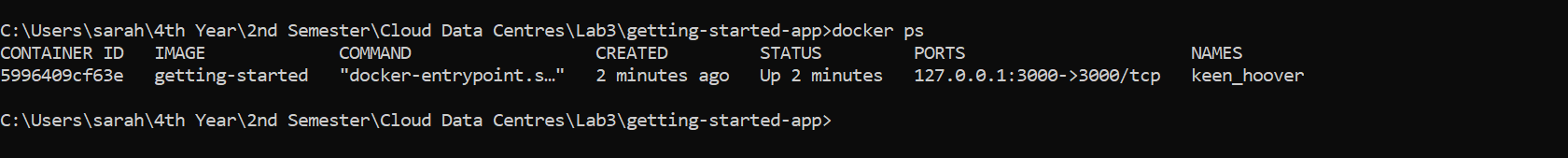
I now have this container in Docker running.



I now have this in my web browser <http://localhost:3000> and I have added some new tasks.

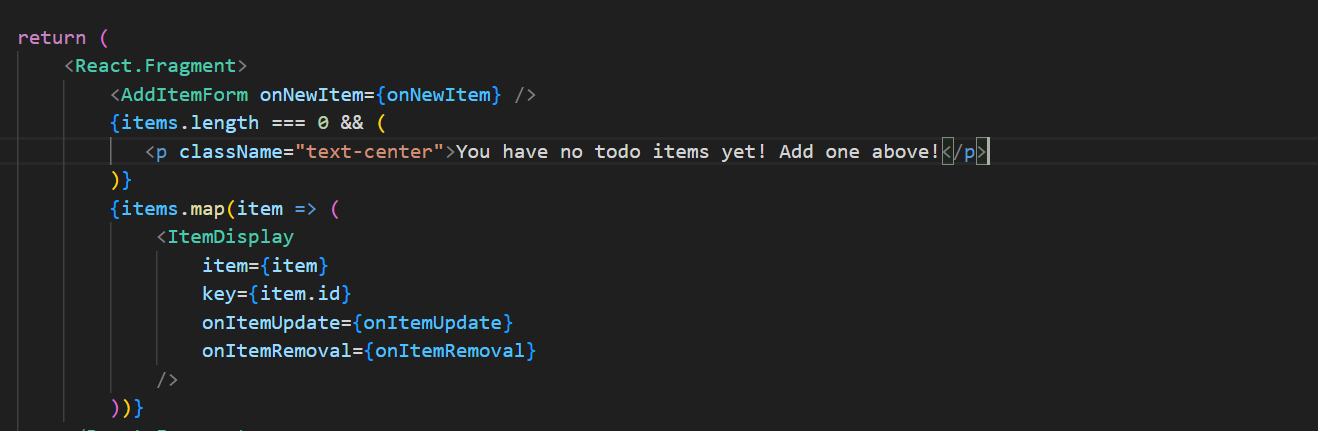


I ran the following command in the terminal to list my containers

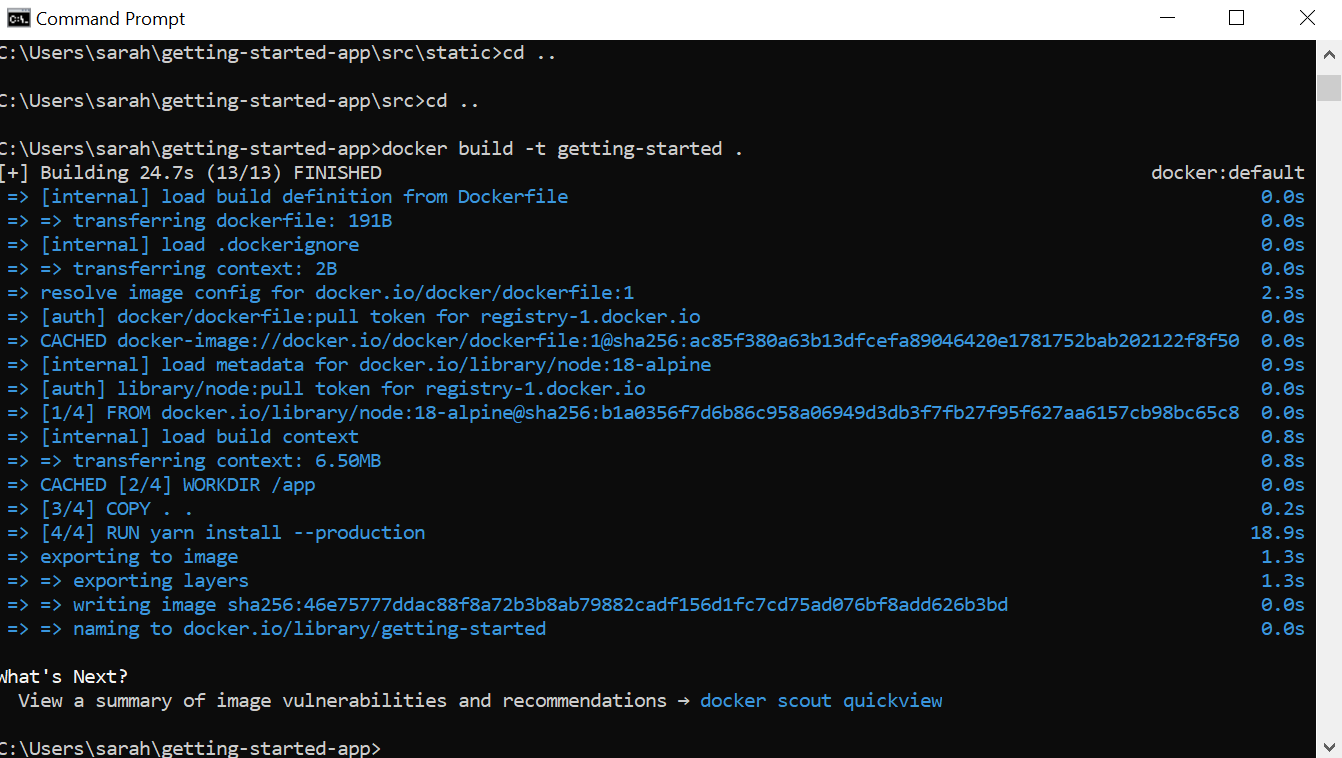


# **Part 3: Update The Application**

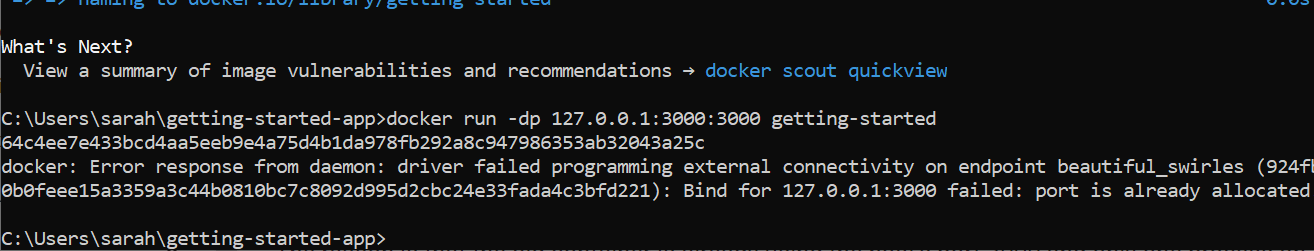
I edited the src/static/js/app.js file to update line 56 to include this.



I then built the updated version of the image using the following command

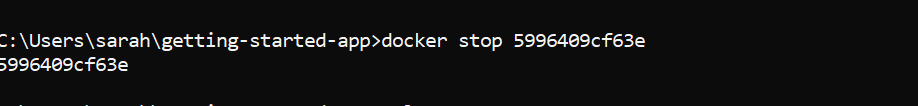


I then started a new container using the updated code and got the following error.

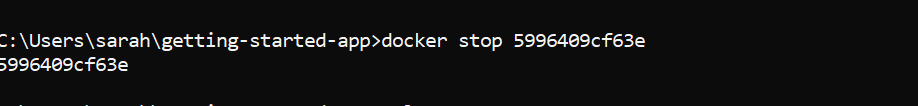


In order to fix the error I need to remove the old container.

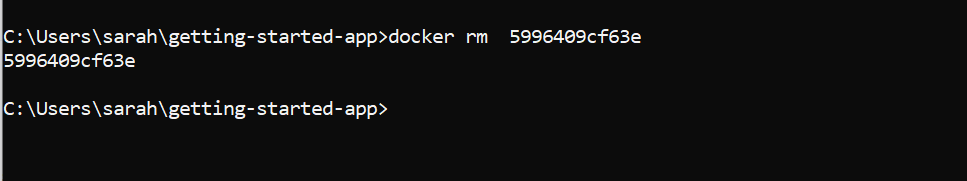
First I got the ID of the container



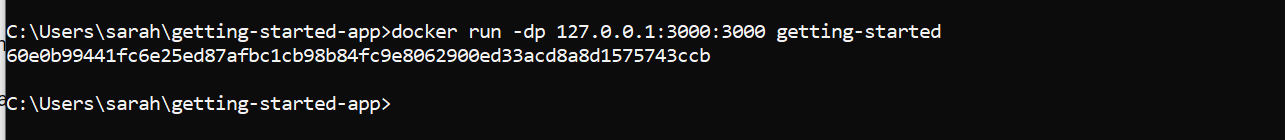
And then I stopped it from running.



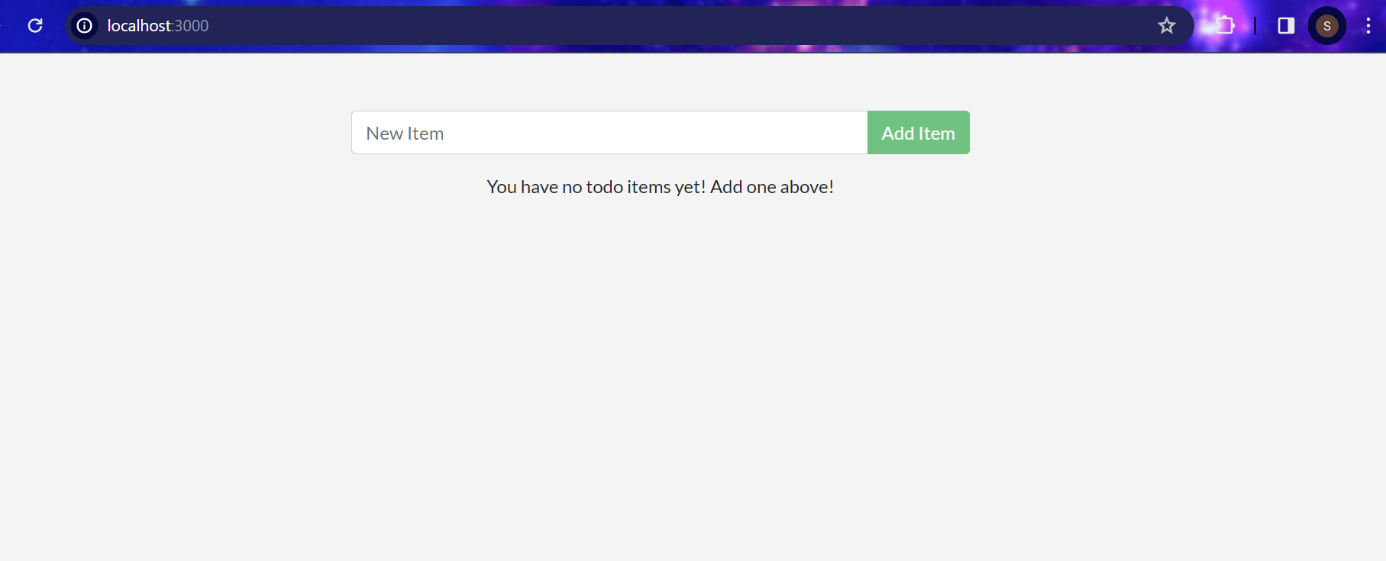
I then removed the container.



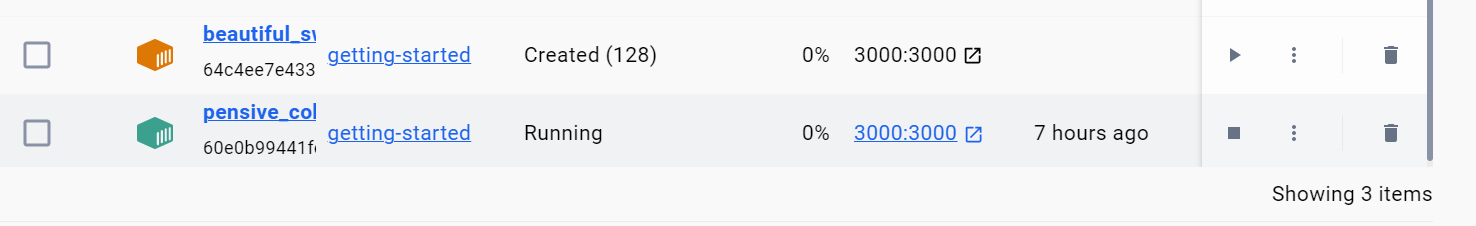
I ran this command to then start my updated app.



My web browser <http://localhost:3000> is now updated to this.

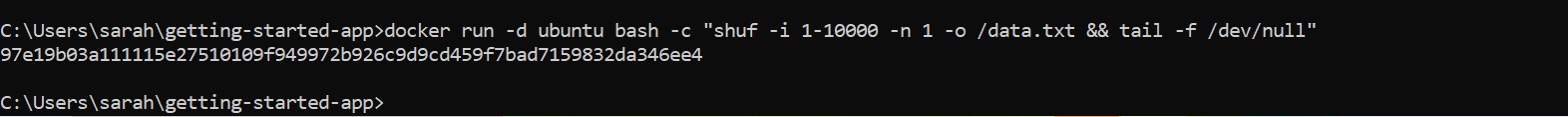


And this is my now running container.



# **Part 5: Persist the DB**

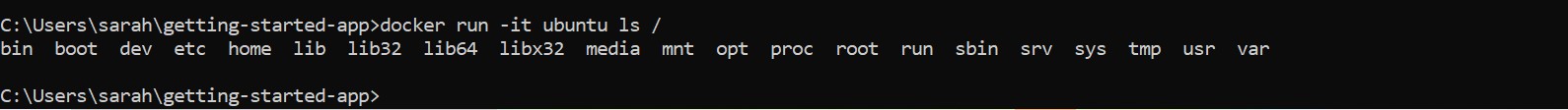
I started a ubuntu container that created a file name and a number between 1 and 1000



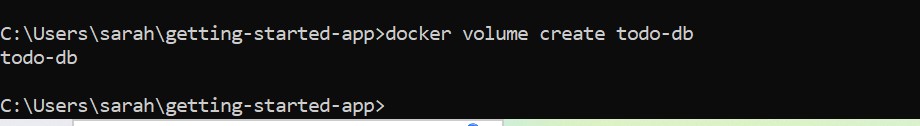
Used the docker exec command to access the container



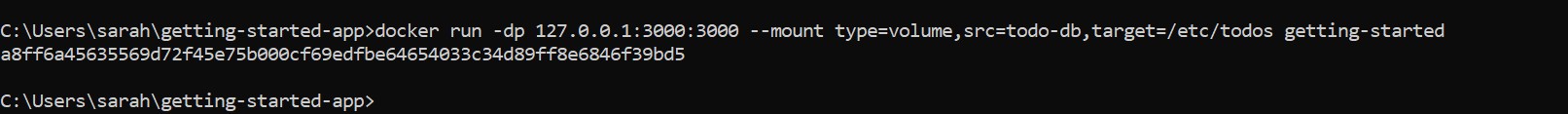
Started a new ubuntu container and I don’t have the same file. I used this command to get the following content.



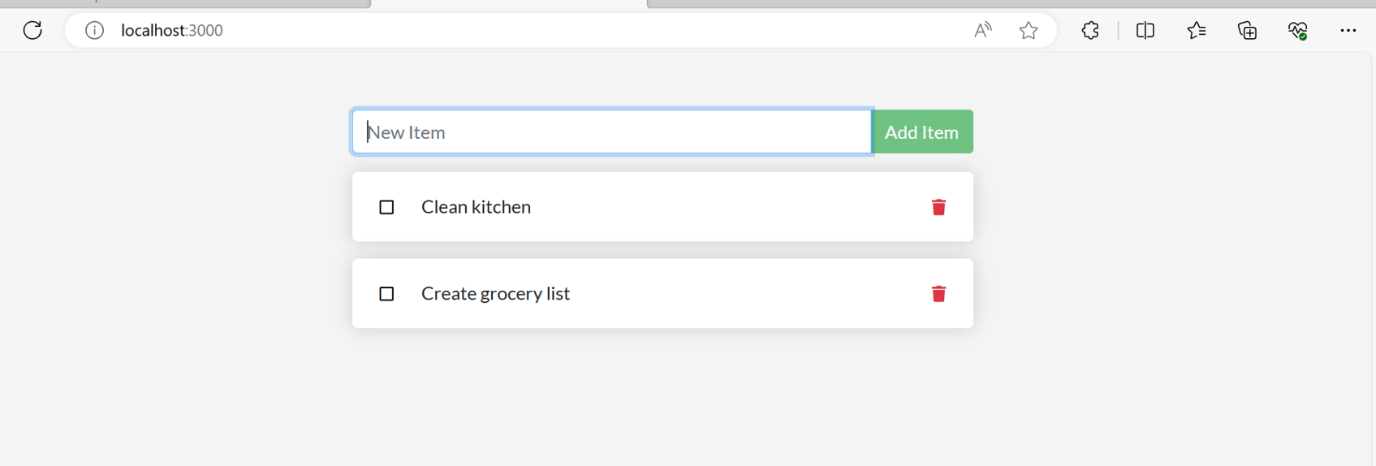
Created a volume.



Started the todo app container and specified a volume amount.



Added some items to the todo list.



I used docker ps and then removed the container for the todo app.

A black screen with white text

Description automatically generated

I inspected the docker data

A black screen with white text

Description automatically generated

# **Part 6: Use Bind Mounts**

Ran this command to start bash in an Ubuntu container with a bind amount. I then did an ls to list the containers file systems. I then changed to the src directory and created a file called “myfile.txt” here.

A screenshot of a computer

Description automatically generated

I then observed that the myfile.txt was created in the getting-started-app directory.

A screenshot of a computer

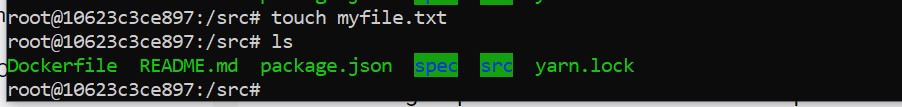
Description automatically generated

And then I proceeded to delete the myfile.txt file.

A screenshot of a computer

Description automatically generated

I then checked to confirm the file was deleted in the app directory.



I think stopped the container session with ctrl + D.

I ran this command in Powershell.

A computer screen with white text

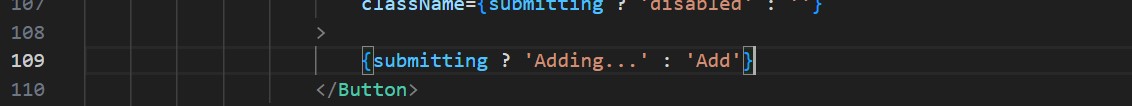
Description automatically generated

I then checked the logs.

A screen shot of a computer

Description automatically generated

I then changed the line in src/static/js/app.js on line 109. It changes the add item button to just a simple add.



The new changes due to the bind amount.

A screenshot of a computer

Description automatically generated

I added in some new items to the list by adding two new items.

A screenshot of a computer

Description automatically generated

I stopped the container.

A screen shot of a computer

Description automatically generated

And then built my new image.

A screen shot of a computer

Description automatically generated

# **Part 7: Multi Container Apps**

Created the network.

A screen shot of a computer

Description automatically generated

I started a SQL container and attached it to the network.

A computer screen with white text

Description automatically generated

I verified that my database was up and running.

A black screen with white text

Description automatically generated

I used the show databases command.

A black screen with white text

Description automatically generated

And the exited SQL.



I then started a new container using the nicolaka/netshoot image. I then used dig mysql to look up the IP address for the hostname mysql.

A screenshot of a computer

Description automatically generated

I then started the container and connected the container to the app network.

A black screen with white text

Description automatically generated

I then added a few items to the todo list.

A screenshot of a computer

Description automatically generated

I then connected to the mysql database to see the items being written to the database.

A screenshot of a computer program

Description automatically generated

# **Part 8: Use Docker Compose**

I created a file called compose.yaml inside the getting-started-app and added all of the code to it.

A screenshot of a computer

Description automatically generated

Now that I have all the code in the compose.yaml file I can now start the application.

A screen shot of a computer

Description automatically generated

I then looked at the logs to see the logs from each services interleaved into a single stream.

A screenshot of a computer

Description automatically generated

I then used this command to tear it all down. The containers will stop, and the network will be removed.

A screenshot of a computer program

Description automatically generated