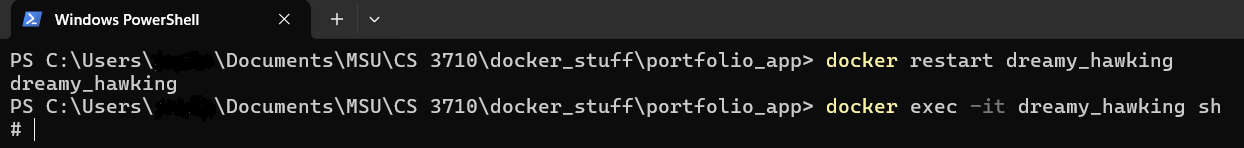
**Installing and Configuring Ruby on Rails**

For CPE3710 Project

Before beginning this process, ensure that you have completed setting up Docker. Once you have your container built from the image supplied in the class materials, you will need to do a few things to prepare that environment for the rest of the project. First, you’ll need to install Rails and all necessary gems (basically Ruby packages or libraries). Next and last, you’ll run Rails to make sure that your container is hosting successfully. Don’t worry; it’s just a test with default values and data that you don’t need to worry about.

**Installing Rails**

Open the command prompt of your choice and cd into the directory containing your locally mapped directory that you created when you set up Docker. While in this active directory, start your Docker container with the following command: # docker restart [container\_name] . Next, you’ll need to get into interactive mode, which allows you to execute command line commands from within the mapped directory. To do this, use the following command: # docker exec -it [container\_name] sh . It should look something like this:

****

Now that you’re in the right place, you’re going to build the portfolio\_app folder. (I know the image above shows that as our directory, but that’s only because I’m writing this after the fact, and I don’t want to delete anything and have to restart. For this reason, also, some of the images later on will not show results of the commands, only the commands themselves.) The first command you’ll need is: # rails new portfolio\_app --skip -bundle .

****

This creates a new rails project called portfolio\_app, and you’re skipping installing some of the bundled packages. You’ll go back and install them later. This is because you want to be sure it’s all installed inside the portfolio\_app directory. In fact, from now on, you will only execute commands from within portfolio\_app. That brings us to our next step, which is the bundle install. First, cd into portfolio\_app as follows: # cd portfolio\_app . Then, initiate the bundle install with the relatively straightforward command: # bundle install .

****

That’s it! Now you have your Rails project up and running and are ready to test. When you set up Docker, you chose a port to forward so that you could view the web app you’re building without having to host it anywhere. The default port for this given in class is 3000, and that is what I’ll be referencing, but if you used a different port, you’ll need to remember and use that, instead. To start your locally hosted server, enter the following command: # rails server -b 0.0.0.0 . It should think for a moment and then look like this:

****

To see if it worked, use your web browser to access the url <http://localhost:3000/>. It should look like this:

****

If you see this, you have succeeded! Yay! This is a pretty straightforward setup, so if you encounter any issues, the two most likely problems are command mistakes and incorrect Docker setup. Make sure you follow the commands in order and exactly as given and that you are always inside of the portfolio\_app directory, once it exists. Docker setup is tricky, so you may need to restart and refer back to the Docker document for additional help.