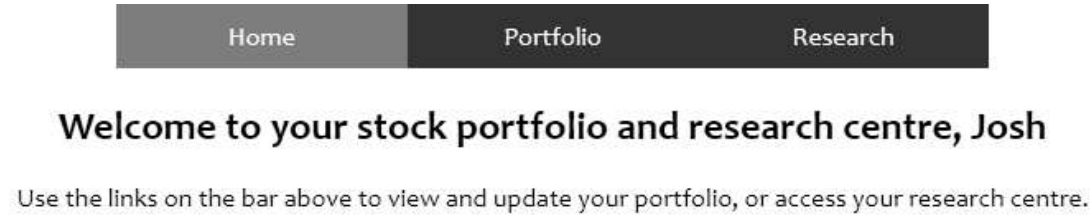
readme.md

Web Portfolio Assignment

A web container version of a simple HTTP server and stock portfolio site.

Made for 159.352 Advanced Web Development at Massey University April 2022 by Joshua Pearson.



Demo

https://jp159352.herokuapp.com/

username and password: 20019455

See run locally for information on how to launch and use the server locally.

Home page

This page contains almost nothing. I chose to have this page be built and set up with the navigation bar in order for a user to quickly access the two pages.

Click one of the two links: Portfolio or Research to get started.

Portfolio

On this page, you have a few elements including:

- A navigation bar at the top, indicating the current page we are one.
- A disclaimer link showing where we retrieve price and symbol data required by IEX.

- A portfolio containing the ticker, number, and average price of each share of a stock we own.
- A live calculation displaying the gain or loss we have made purchashing and holding the shares of each stock in the portfolio.
- Form input with autocomplete stock name, quantity, and prices with error handling on input.
- A reset button that clears the user input from the form.
- Space below the buttons will display any user relevant messages after they press update.

Research

Home | Portfolio | Research

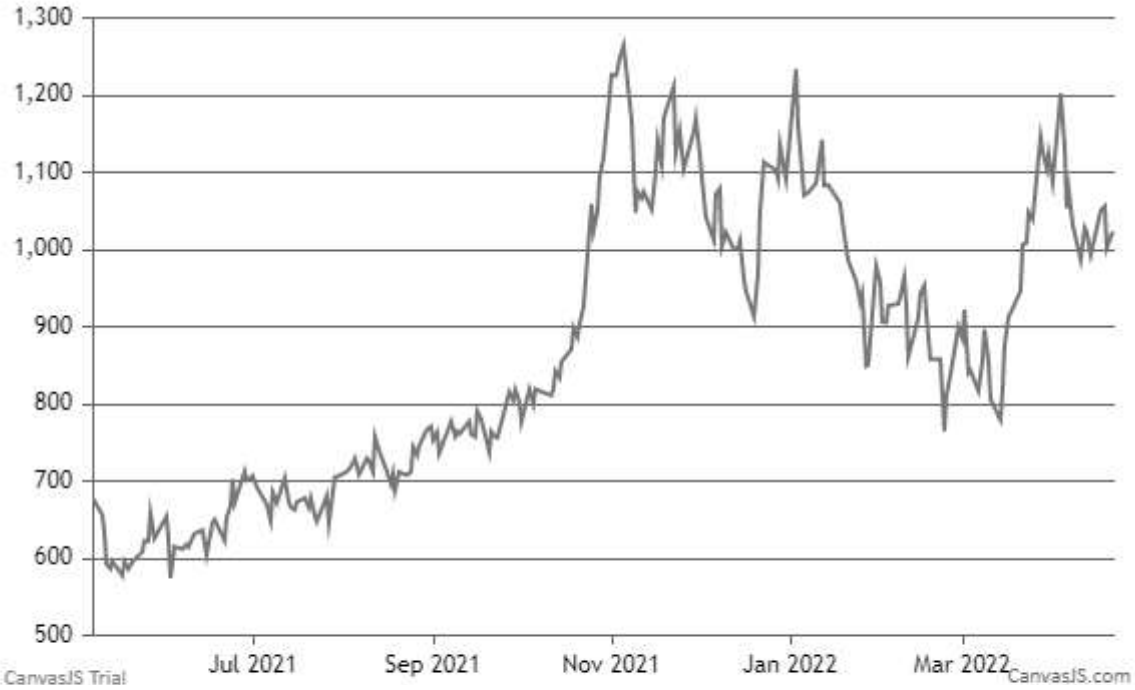
Josh's Stock Research Centre

Data provided by IEX Cloud

Stock Symbol: [ ] Research

Symbol: *TSLA*
Company Name: *Tesla Inc*
PE ratio: *120.1019*
Market Capitalization: *1,009,336,389,503*
52 week high: *1243.49*
52 week low: *546.98*

Range | 1m | 3m | 6m | YTD | 1y | All | From May 07 2021 | To Apr 22 2022



On this page, the elements are simpler before pressing research. We have:

- An input box where the user can select (with autocomplete) a stock to research.
- Once submitted if the user enters a valid stock, you will see the stocok information and a chart of the last 5 years of stock history. Note the chart defaults to only viewing 1 year data in the window for cleaner views but there are range buttons above the chart and a bar below to finetune the viewing window.

Also, note that the chart uses CanvasJS free which is techncially only valid for free for 30 days so this may stop working at any time.

Container and deployment info

This project was built almost exclusively using python 3.10. Once completed, I loaded this into a docker container and deployed it to Heroku. The dockerfile data is visible inside the repo but the relevant data is here:

```
FROM python:3.10.0a2-slim-buster
RUN pip3 install requests
COPY . /src
WORKDIR /src
CMD python server.py $PORT
```

The Dockerfile to run on Docker locally was changed to be:

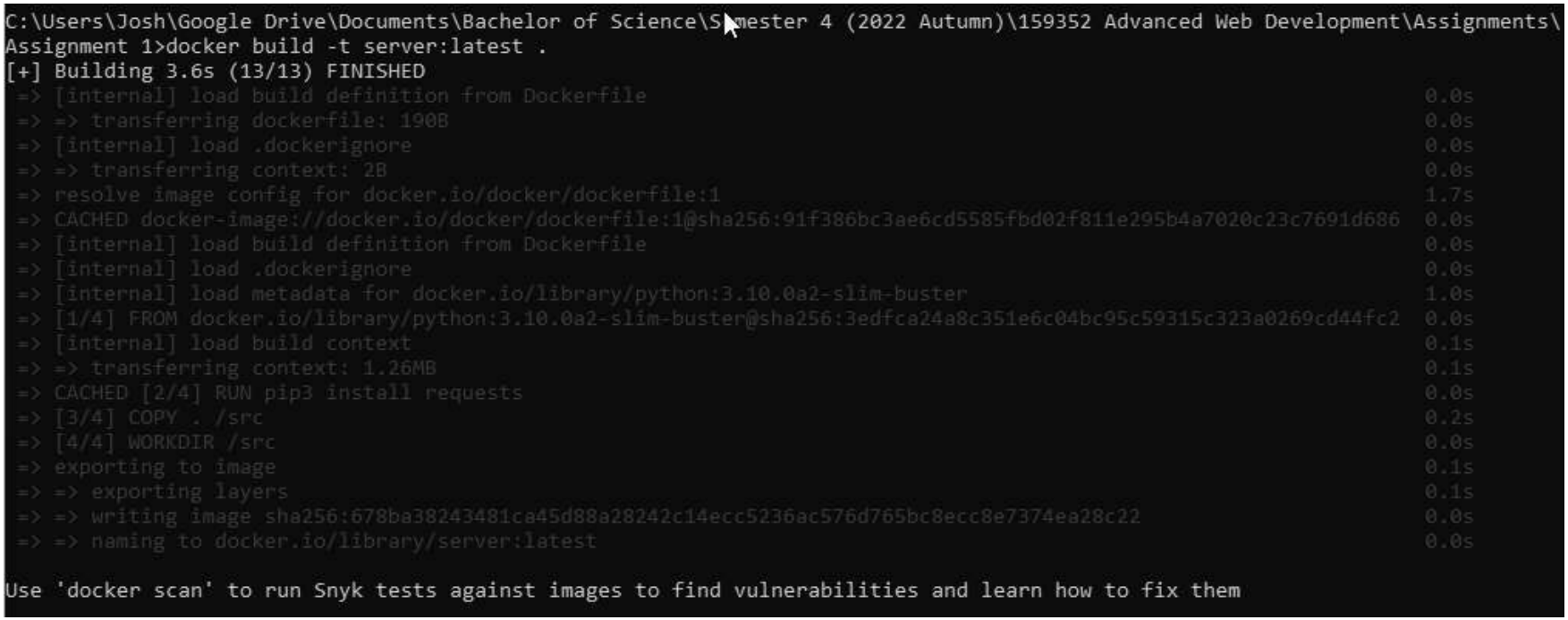
```
FROM python:3.10.0a2-slim-buster
RUN pip3 install requests
```

```
COPY . /src
WORKDIR /src
EXPOSE 8080
CMD python server.py 8080
```

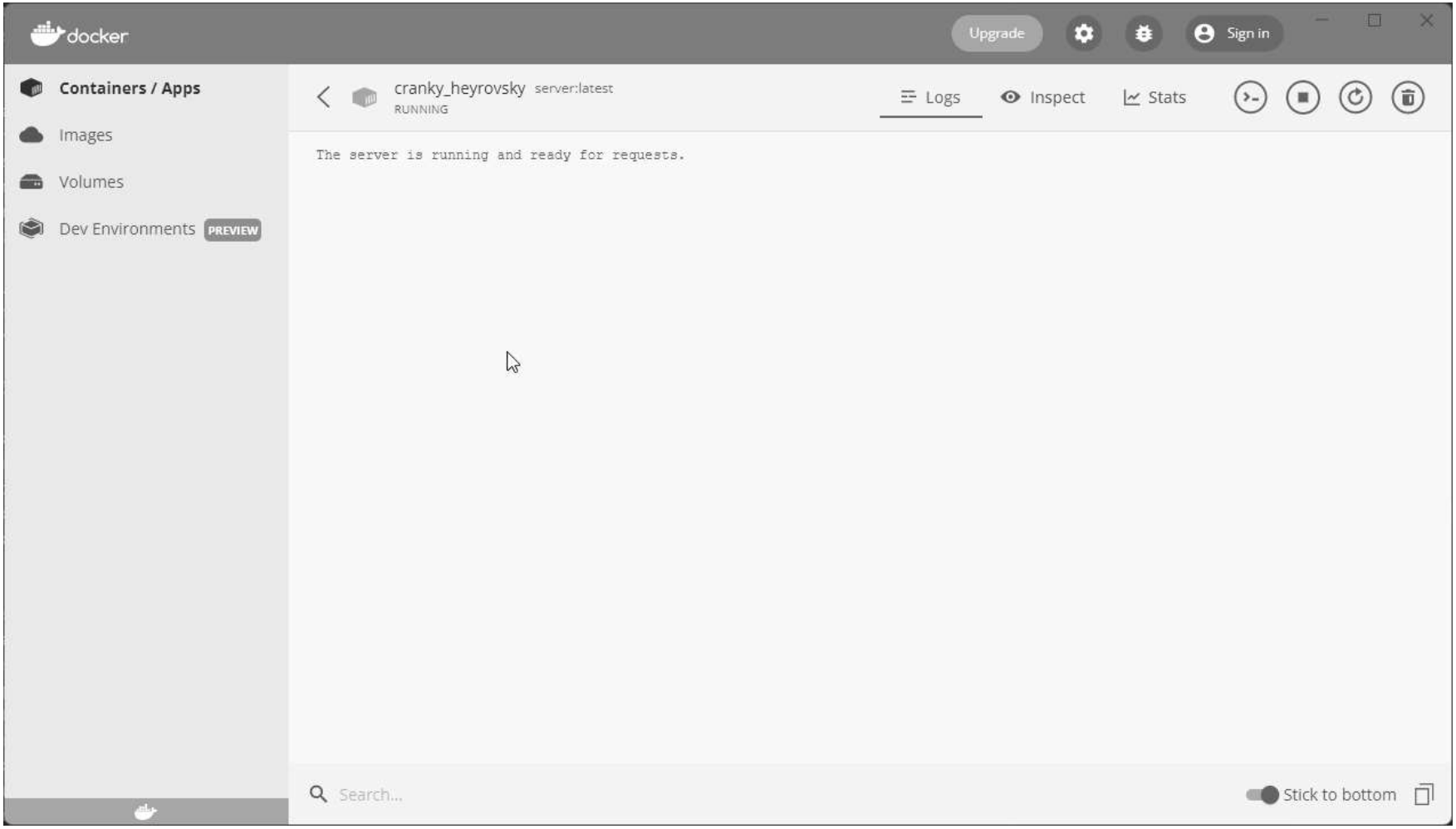
We use $PORT to allow Heroku on deployment to select the port used for hosting services.

Some images of the process are below.

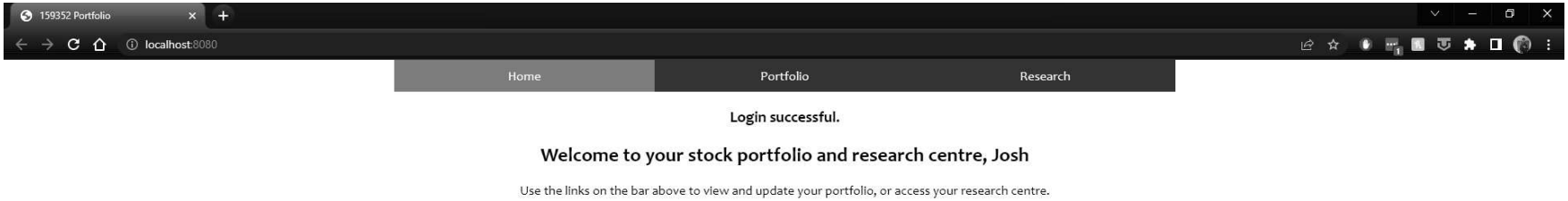
Command line:



Docker running locally:



Accessible and working completely on localhost:8080



Run Locally

- *Clone the project from Github*

```
git clone https://github.com/baconeta/159352-assignment1
```

Go to the project directory

```
cd assignment1
```

Install dependencies

```
pip3 install requests
```

Start the server (you should confirm that line 25 is commented out and line 26 is uncommented unless running in a Docker container - see below for more info)

```
python3 server.py
```

- *Run locally without cloning from git repo*

In your local environment or IDE ensure you have the requests library installed (consider using pip3 install requests)

Otherwise similarly to above using the command line:

Go to the project directory

```
cd assignment1
```

Install dependencies

```
pip3 install requests
```

Start the server. You can also skip this cmd line command and instead run it in an anaconda or venv environment inside your IDE and run the server file manually.

```
python3 server.py
```

Make sure the following code is set correctly otherwise ensure you pass the port argument into the run command. Docker and Heroku require the sys argument instead to be uncommented as it is in the repo version of server.py.

Once the server is running using either method you can access the the locally hosted site using: http://localhost:8080/

username and password: 20019455

Tech Stack

Server:

Python 3.10 using the following main modules

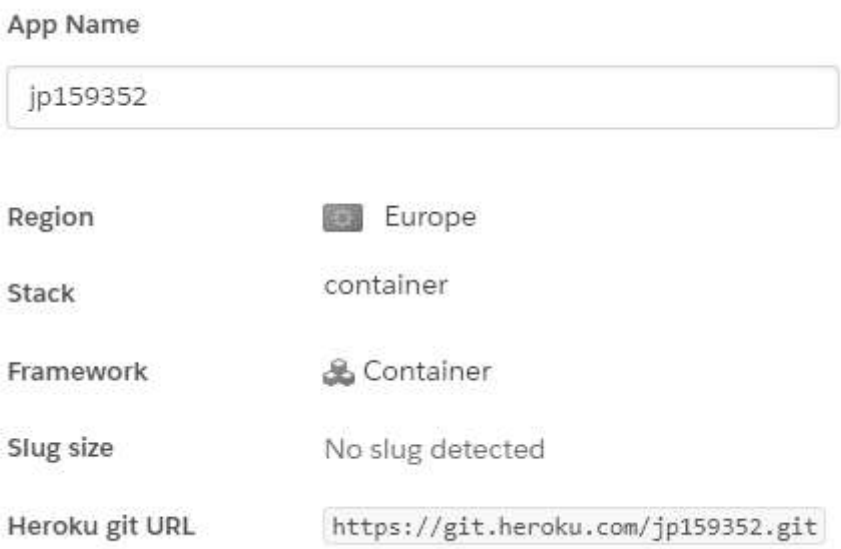
- requests
- base64 *core builtin*
- socket *core builtin*

Docker container

- Docker version 20.10.14, build a224086
- Docker Desktop 4.7.1 (77678)

Heroku

- heroku/7.60.1 win32-x64 node-v14.19.0
- config: web /bin/sh -c python\ server.py\ $PORT



Tools:

- PyCharm 2022.1 (Professional Edition)
- Anaconda3 Python Environment
- Webstorm 2022.1 (Professional Edition)
- IEX REST API (free version)
- CanvasJS (30 day free trial)

Acknowledgements and references

- CanvasJS stock chart reference
- IEX docs
- Docker documentation