

STOCK INFORMATION WEB INTERFACE

FE 520 Final Project

Group 6: Gloria Rumao, Moksh
Ajmera, Kinjal Shukla, Ruchi
Pandey



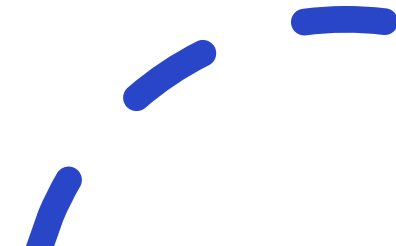
FLASK

```
5 from IPython.display import HTML
6
7 app = Flask(__name__)
8 app.secret_key = '1234'
9 @app.route('/')
10
11 def home():
12
13     return render_template('home.html')
14
15 @app.route('/data', methods = ['GET', 'POST'])
16
17 def data():
18
19     if request.method == 'POST':
20
21
22         Info = {}
23         Info['stock_symbol'] = request.form['ticker']
24         Info['start_date'] = request.form['startdate']
25         Info['end_date'] = request.form['enddate']
26
```

- Flask is a Python framework for building lightweight and dynamic web applications.



LIBRARIES USED

- Yfinance -> used to download yahoo finance data
 - json -> convert strings to Python datatypes
 - Numpy -> is a Python library used for working with arrays
 - Pandas -> library used for real world stock data analysis and manipulation in python
 - Render_template -> library in flask to create html templates
- 

Query Editor

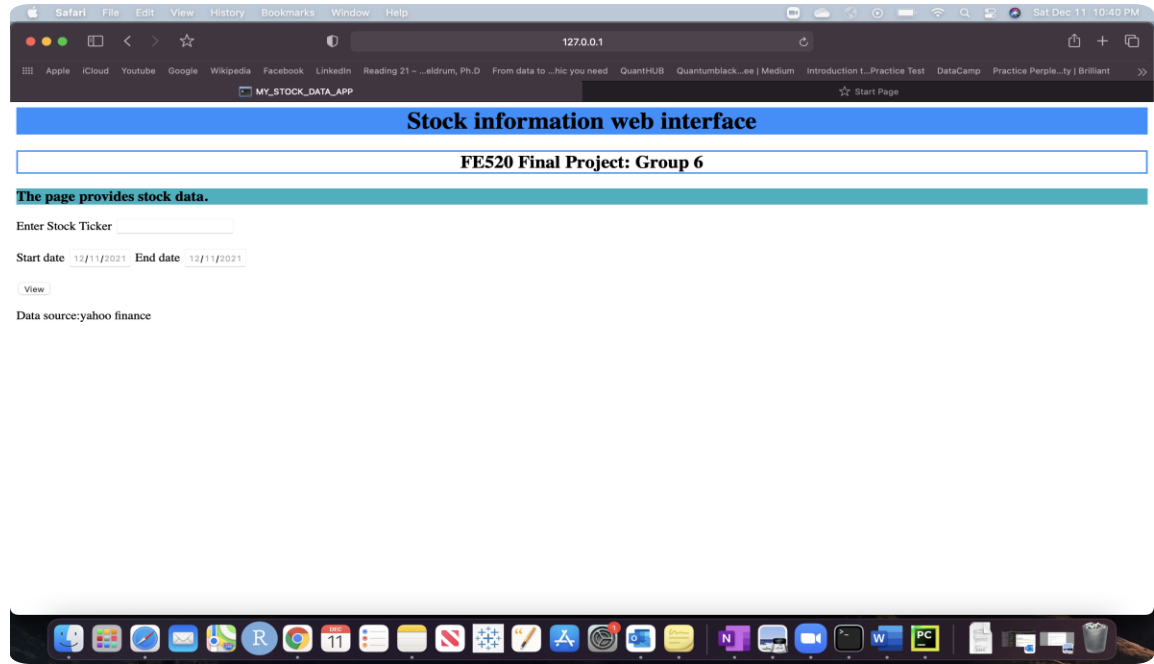
```
1 Select * from appl_records;
```

Store and

fetch data from SQL

Notifications Data Output Messages Explain Query History

	date date	open double precision	high double precision	low double precision	close double precision	adj_close double precision	volume bigint	ticker character varying (255)
1	2021-11-01	148.99000549316406	149.6999969482422	147.8000030517578	148.9600067138672	148.742919921875	74588300	AAPL
2	2021-11-02	148.99000549316406	149.6999969482422	147.8000030517578	148.9600067138672	148.742919921875	74588300	AAPL
3	2021-11-03	150.53999938964844	151.97000122070312	149.82000732421875	151.49000549316406	151.2692413330078	54511500	AAPL
4	2021-11-04	151.5800018310547	152.42999267578125	150.63999938964844	150.9600067138672	150.74000549316406	60394600	AAPL
5	2021-11-05	151.88999938964844	152.1999969482422	150.05999755859375	151.27999877929688	151.27999877929688	65414600	AAPL
6	2021-11-08	151.41000366210938	151.57000732421875	150.16000366210938	150.44000244140625	150.44000244140625	55020900	AAPL
7	2021-11-09	150.1999969482422	151.42999267578125	150.05999755859375	150.80999755859375	150.80999755859375	56787900	AAPL
8	2021-11-10	150.02000427246094	150.1300048828125	147.85000610351562	147.9199981689453	147.9199981689453	65187100	AAPL
9	2021-11-11	148.9600067138672	149.42999267578125	147.67999267578125	147.8699951171875	147.8699951171875	41000000	AAPL
10	2021-11-12	148.42999267578125	150.39999389648438	147.47999572753906	149.99000549316406	149.99000549316406	63632600	AAPL
11	2021-11-15	150.3699951171875	151.8800048828125	149.42999267578125	150	150	59222800	AAPL
12	2021-11-16	149.94000244140625	151.49000549316406	149.33999633789062	151	151	59256200	AAPL
13	2021-11-17	151	155	150.99000549316406	153.49000549316406	153.49000549316406	88807000	AAPL
14	2021-11-18	153.7100067138672	158.6699981689453	153.0500030517578	157.8699951171875	157.8699951171875	137827700	AAPL
15	2021-11-19	157.61000389648438	161.02000427246094	156.52000877929688	160.5500030517578	160.5500030517578	117305600	AAPL

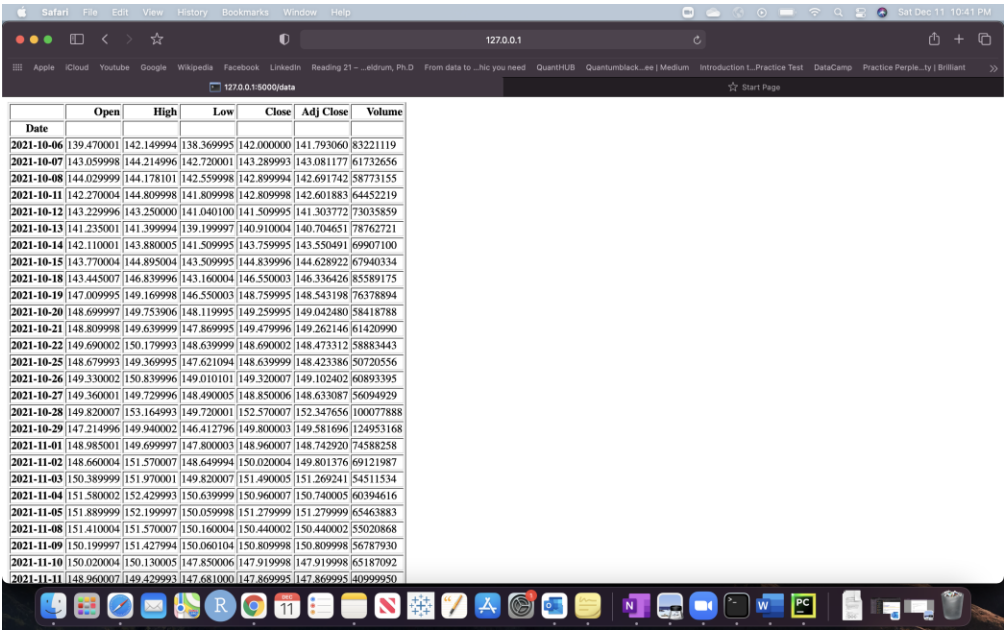


```
import psycpg2
import yfinance as yf
import numpy as np
from psycpg2.extensions import register_adapter, AsIs

psycpg2.extensions.register_adapter(np.int64, psycpg2._psycpg.AsIs)

ticker = 'AAPL'
stock_data = yf.download(ticker, start='2021-8-1', end="2021-9-1")
stock_data.index = np.datetime_as_string(stock_data.index, unit='D')
stock_data['Ticker'] = ticker
stock_data = stock_data.rename(columns={"Adj Close": "Adj_Close"})
records = stock_data.to_records(index=True)
```

WEB PAGE





Future work

- Use javascript to make the webpage more interactive
 - Add a download button and plot function and multiple ticker option
- 