# Stock Market Recommender

## Background

Behavioral Finance is an exciting field and you've decided to jump in with an idea for your new start up. In an earlier experimentation you've noticed a correlation between the various social media posts on a stock symbol and that share price for that stock symbol.

You've decided to build an app that can provide a buy, hold or sell recommendation when given a stock symbol. The recommendation adjusts itself based on data.

Unfortunately, there are multiple challenges around building this app. However, as a smart CTO, you've realized that things must be done in parallel. While you look at building your backend, you've decided to start working on the frontend today.

## Technical Requirements

- Since you don't have a complete backend available to you yet, you will have to create and use your own mock data. Use the *Math.random* function to generate mock values for a given stock price and the count of social media posts that your backend service will provide, as well the recommendation for buying, holding (doing nothing) and selling the stock. Be sure to keep your code maintained in such a way that you can later replace it with a backend API. You can call these functions "stockPriceGenerator" (that takes the stock symbol and dates as parameters), "socialMediaCountGenerator" (that takes the stock symbol and social media type e.g. Twitter) and "recommendationAlgorithm" (that takes the stock prices and social media counts)
- If things go well, you expect to have a team continue building the frontend and adding more features. The code must be modern and easy to maintain.
- Your code must allow new features to be added easily. Yet, it should update seamlessly and be consistent throughout every "component" when new information is received from the "backend".
- You intend to demo the app to a lot of people who might become customers. It must be tested so that there are no surprises in demos.

## Feature Requirements

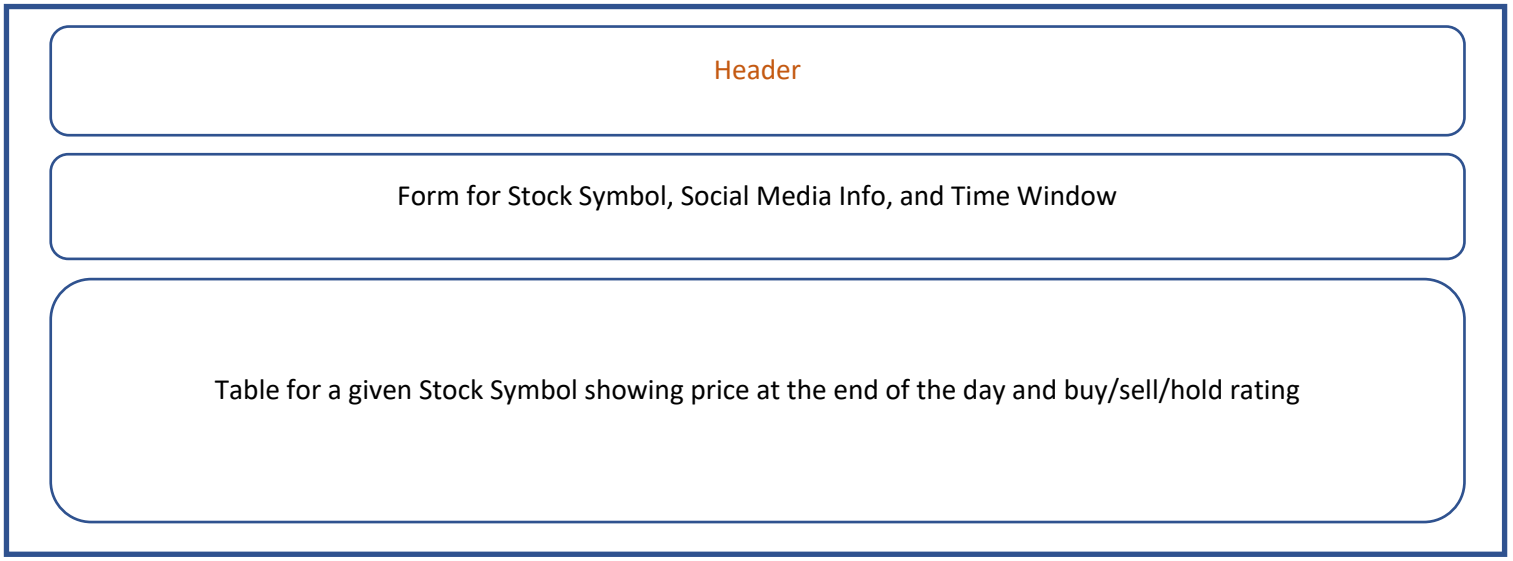
The investors are quite pessimistic of your approach and have a lot of alternative ideas that you've promised to entertain to qualify for their funds. You must design a flexible architecture for your stock prices.

- It would be simply amazing if you could swap out algorithms that recommend a buy/hold/sell rating on the fly.
- You understand that some algorithms might require more information (like constants or risk ratios) and other information in order to generate this. It would be great if you could account for this.

## Visual Requirements

The app works with a lot of data. It needs to be presented to the user in a digestible way. Even though you have access to multiple CSS libraries, you've decided to show off your skills and write out your styles directly in CSS/SASS/LESS. You only care about keeping the app's footprint to a minimum and would do anything to speed up the initial load times.

- There needs to be separate sections showing
    - user input for stock symbol, a time window that is defaulted to 10 days (including today),
    - a count of social media posts and the stock price over 10 days,
    - recommendations for when to buy, hold (do nothing), or sell in those 10 days




## Optional Challenges

Please note that the following milestones are not mandatory at all, but very cool to have in your demo.

- Some of your customers might be visually impaired. It would be nice if they're able to use the app as well.
- Assume your backend teams tell you they can provide the top 2 social media posts as well. Display them on the app at the bottom.
- Allow your users to request more than the top 2 social media posts.
- Allow your users to request top "x" number of social media post from every social media service you have integrated.
- Allow your users to add / remove any social media service from the system (recommendation and displays) on the fly.
- Build a graph (feel free to use a library to help with this part) indicating the points where your algorithm will change the buy/hold/sell recommendation.