F.T.A.H.

Hayden Pour, Julian Beaulieu, Mohamed Rayyan, Padraic Reilly, Nicholas Cardinal

Artifact

[Historical daily prices and volumes of all U.S. stocks]

Data we have:

1. Trading Date
2. Opening Price
3. Daily Price (High)
4. Daily Price (Low)
5. Closing Price
6. Volume Sold

Platform:

Web Application / Desktop Application

Programming Languages:

- JavaScript
- HTML
- CSS
- Python

Feature List (Question of Interest):

Important features:

1. ***How does a stock's price change over a given period?***
2. ***How does the volume of stock sold change over a given period?***
3. ***What is the moving average for a stock over a given period?***
4. ***What is the highest/lowest closing price over a given period?***
5. ***Which stock has the largest margin over a given period?***

6. ***What days had the largest increases or decreases in price? (Useful for correlating to real world events)***
7. ***How did a specific stock's daily change compare to the market average change? (High or low performing stocks)***

Implement if we have time, because we need to use data outside of the database:

1. ***How do real-world events affect stock prices?***
2. ***What is the predicted opening and closing price of a stock?***
3. ***What stocks are the best to trade for the day?***

Sprint-2:

Action Items:

- **Client/UI**
  - **JS Promises**
  - **Create import routine for data retrieved from server**
  - **Sort retrieved labels alphabetically**
  - **Sort retrieved data by date**
  - **Display graph from label click**
  - **Display other various information on label click**
- **Server**
  - **Update csv file name to stock ticker name - stock name**
  - **Return labels in chucks to client backend**
  - **Parallel processing when importing from csv**

Tests:

- **Client/UI**
  - **JS Promises**

    Correct Output: The user's data will load completely from server before displaying information to the UI

  - **Create import routine for data retrieved from server**

    Correct Output: The website displays stock information when user clicks the stock they want to view

  - **Sort retrieved labels alphabetically**

    Correct Output: The website will display the stock list in alphabetical order
  - **Sort retrieved data by date**

    Correct Output: The data received by the server is properly sorted by date
  - **Display graph from label click**

    Correct Output: When a label is clicked, the website will display all pertinent information on the right side of the UI

- **Server**
    - **Update csv file name to stock ticker name - stock name**
        Correct Output: All files in our dataset are renamed correctly

    - **Return labels in chucks to client backend**
        Correct Output: When called, the website requests data from the server and it is returned in a format that can be understood by the website

    - **Parallel processing when importing from csv**
        Correct Output: Multiple files can be read from at any given time