The Stock Monitor

Kevin White - Kevin2621

Aidan O'Connell - aidanoc845

Paul Drews - MyApaulogies

Shreyas Kulkarni - ShreyasKulkarni123

Andrew Lee - anle7157

Project Description:

The Stock Monitor is a website that helps investors find information about companies — such as company description, stock price over time, address, etc. — using the company's ticker/symbol.

The home page displays a rotating carousel that will suggest stocks for the user to explore. The user may click on any stock card buttons to learn more specific information about that company. If the user decides to, they may also add it to their profile's watchlist stocks list.

The news page displays a long grid of news articles that present the latest news on companies and their stocks.

Our mission is to make it as convenient as possible to learn more about companies and make well-informed investing decisions.
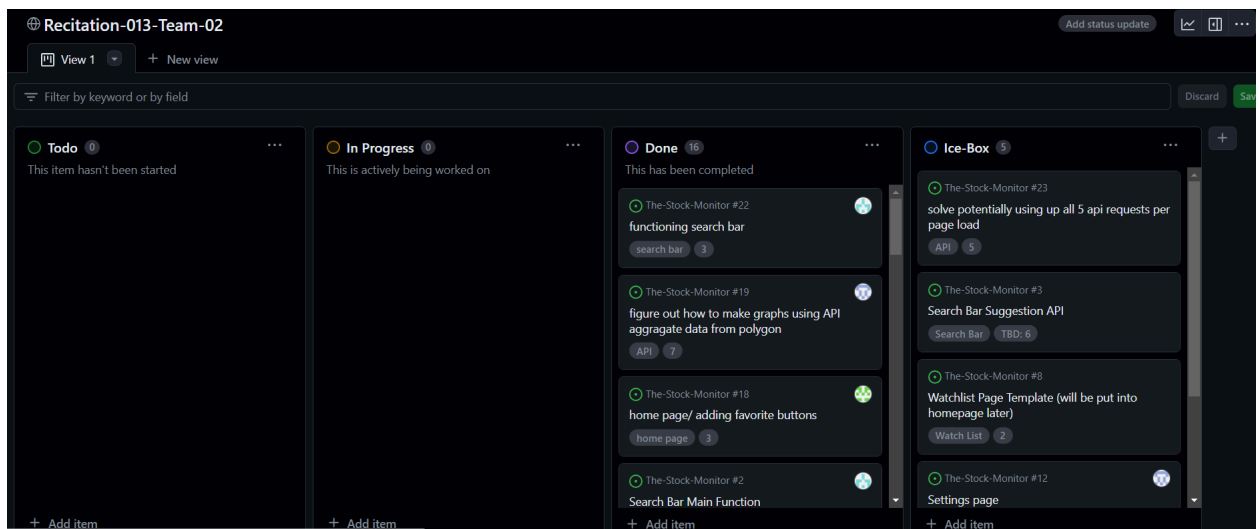
VCS:

https://github.com/ShreyasKulkarni123/The-Stock-Monitor

- ○ Source Code
- ○ Test Cases
- ○ Video demo
- ○ README.md in GitHub
- ○ Project documentation
- ○ Project Board

GitHub Project Board:

https://github.com/users/ShreyasKulkarni123/projects/1/views/1

Video:

5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.
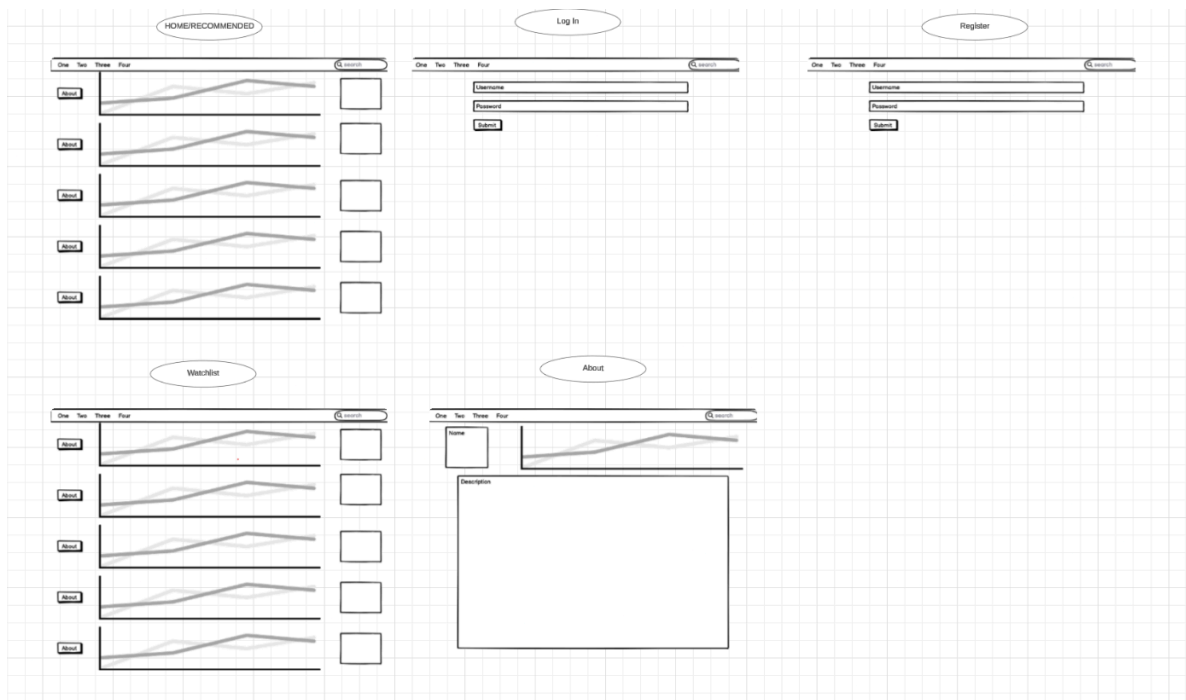
- https://drive.google.com/file/d/1jd1mV33r2_S4mX0jhCFFrJZPp9AVLzqB/view?usp=drive_link

Contributions:

- Kevin
    - Populated all the data using the API
    - Figured out the API and how to implement it into the app
    - Prettied up the home, about, and news page
    - Got all pages integrated with the API Data
    - Created the login, register, and logout partials
- Aidan
    - Created wireframe
    - Created initial nav bar partial
    - Created favorite and unfavorite buttons on featured stocks, watchlist, and about page
    - Created post and get requests to add stocks to the watchlist table and redirect to the desired page
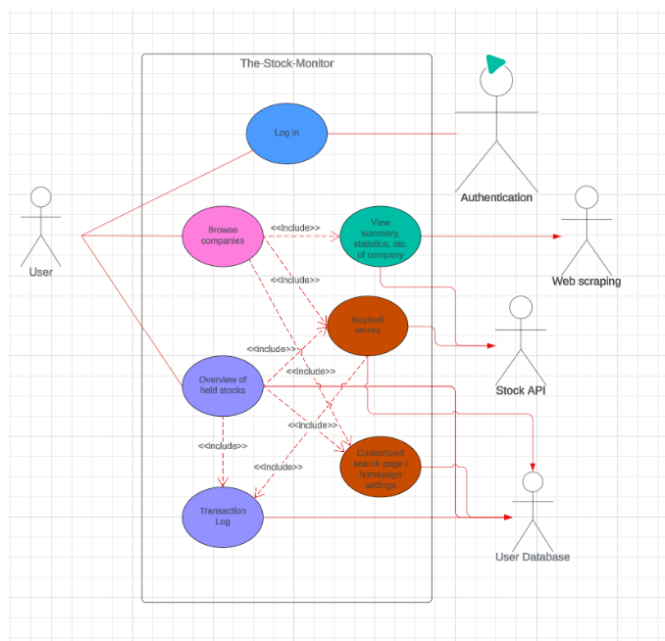- Paul

- ○ General maintenance of server-side code — fixed dependencies, Docker issues, templating issues, and database issues
- ○ Designed database, wrote and debugged SQL scripts
- ○ Made endpoint for passing data the client side without leaking the polygon API key
- ○ Made graphs
- Shreyas
  - ○ Worked a lot on the html for handlebars, worked on the carousel for the home page (Was originally used for nav bar, but used for home page instead because of api call limit). Created the about page and all of the stock info that went into that page. Also did footer/header partials in handlebars. Created a couple of basic endpoints.
  - ○ Mainly focused on front end
  - ○ Also made the use case diagram
- Andrew
  - ○ Helped organize the index.js file, structuring the project to enable modularly adding functions into the project. This helped reduce git merge conflicts.
  - ○ Created first draft of news page

Wireframes:

- The graph images were used to represent stock information not necessarily implementing a graph.
- The watchlist location was moved onto the homepage for simplicity and user-friendliness.

Use Case Diagram:

Test Plan Observations and Results:

- In Lab 11, you created a Test Plan. You need to include the test results and observations in the project report. Refer to **this** for more information
- I tried to make the unit tests work, but I kept getting errors through the node_modules folder, which makes no sense to me.
- The specific error is "TypeError: app.address is not a function" and then references a folder in node_modules.
- This is happening to all of the unit tests that I have created, no matter what test it is, including the ones given to us.
- The thing is that the unit tests were working for us at the start of the project. I do not know why they have stopped working now.
- In the end, these unit tests had no effect on development whatsoever. We as a group just kept going, and if there was an error, we just fixed it right away. All in all, it did not affect anything at all in the long run.

Deployment:

- https://the-stock-monitor.onrender.com/home (**USE FIREFOX!!!!**) Other browsers have trouble displaying information fast and have extremely long load times.