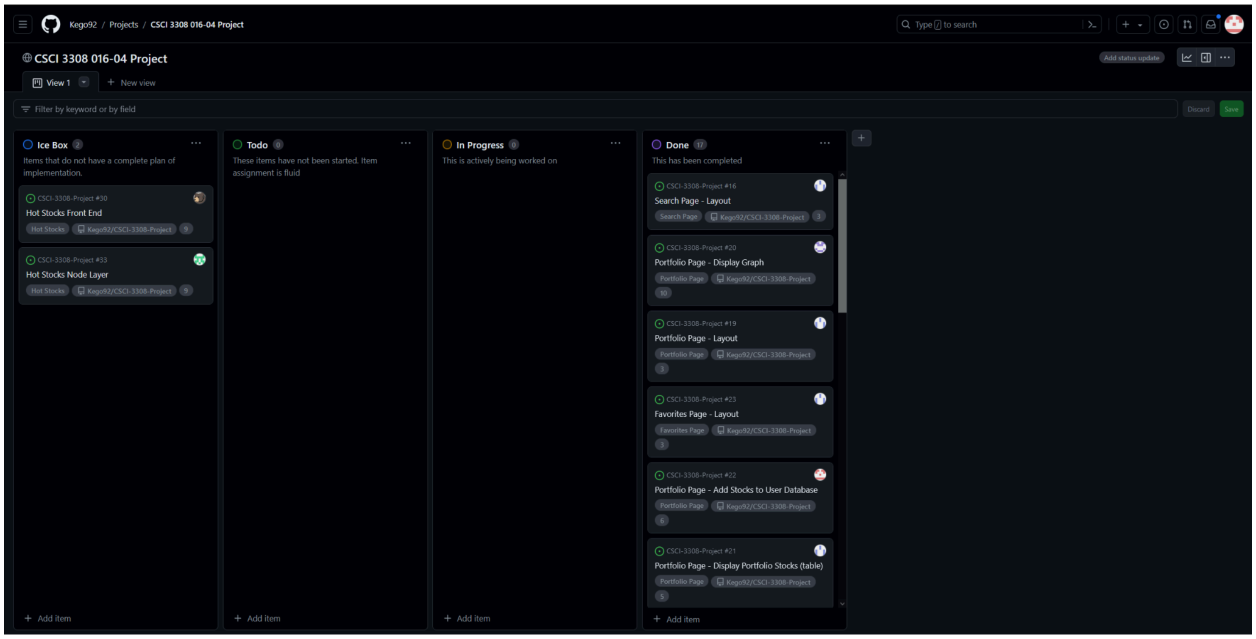
Stock N Load

Group Members:
Jack Bayer - https://github.com/Jaxful
Logan Bird - logan-bird1
Alex Chekal - alexander-chekal3
Keegan Gore - Kego92
Eli Jordan - eljo5115

Project Description: Stock N Load is the stock tracking platform for the modern day investor. There are numerous stock trading platforms out on the market currently but they all suffer from the same fatal flaw; too much. That is the goal of Stock N Load, to create a simple, easy to use platform where the end user is able to track their stocks without being distracted by pop-ups, arbitrary info, etc. Our Platform displays to the user precise info on every stock that they have favorited via interactive graphs that show the history of the stock and daily changes along with an array of tools to allow quality analysis. Further the user can see the total value of their portfolio over the past three days under the portfolio tab. This allows the user a deeper understanding of how their stocks are performing, promoting deeper analytics. To continue, for ease of use our site allows users to search for any stock they want via a simplistic search page. This is done with a search bar where the user can search the ticker symbol of their desired stock and click the green add button. If the user has sold all their shares or changed their mind on a stock they can also remove the stock from their favorites on this page with the red delete button.

Project Tracker: https://github.com/users/Kego92/projects/2


Kego92 / Projects / CSCI 3308 016-04 Project

CSCI 3308 016-04 Project

View 1 · New view

Filter by keyword or by field

Ice Box 2
Items that do not have a complete plan of implementation.
- CSCI-3308-Project #30 — Hot Stocks Front End — Hot Stocks · Kego92/CSCI-3308-Project · 9
- CSCI-3308-Project #33 — Hot Stocks Node Layer — Hot Stocks · Kego92/CSCI-3308-Project · 9

Todo 0
These items have not been started. Item assignment is fluid

In Progress 0
This is actively being worked on

Done 17
This has been completed
- CSCI-3308-Project #16 — Search Page - Layout — Search Page · Kego92/CSCI-3308-Project · 3
- CSCI-3308-Project #20 — Portfolio Page - Display Graph — Portfolio Page · Kego92/CSCI-3308-Project · 10
- CSCI-3308-Project #19 — Portfolio Page - Layout — Portfolio Page · Kego92/CSCI-3308-Project · 3
- CSCI-3308-Project #23 — Favorites Page - Layout — Favorites Page · Kego92/CSCI-3308-Project · 3
- CSCI-3308-Project #22 — Portfolio Page - Add Stocks to User Database — Portfolio Page · Kego92/CSCI-3308-Project · 6
- CSCI-3308-Project #21 — Portfolio Page - Display Portfolio Stocks (table) — Portfolio Page · Kego92/CSCI-3308-Project · 5

Video: Stock N Load Demo

VCS: https://github.com/Kego92/CSCI-3308-Project

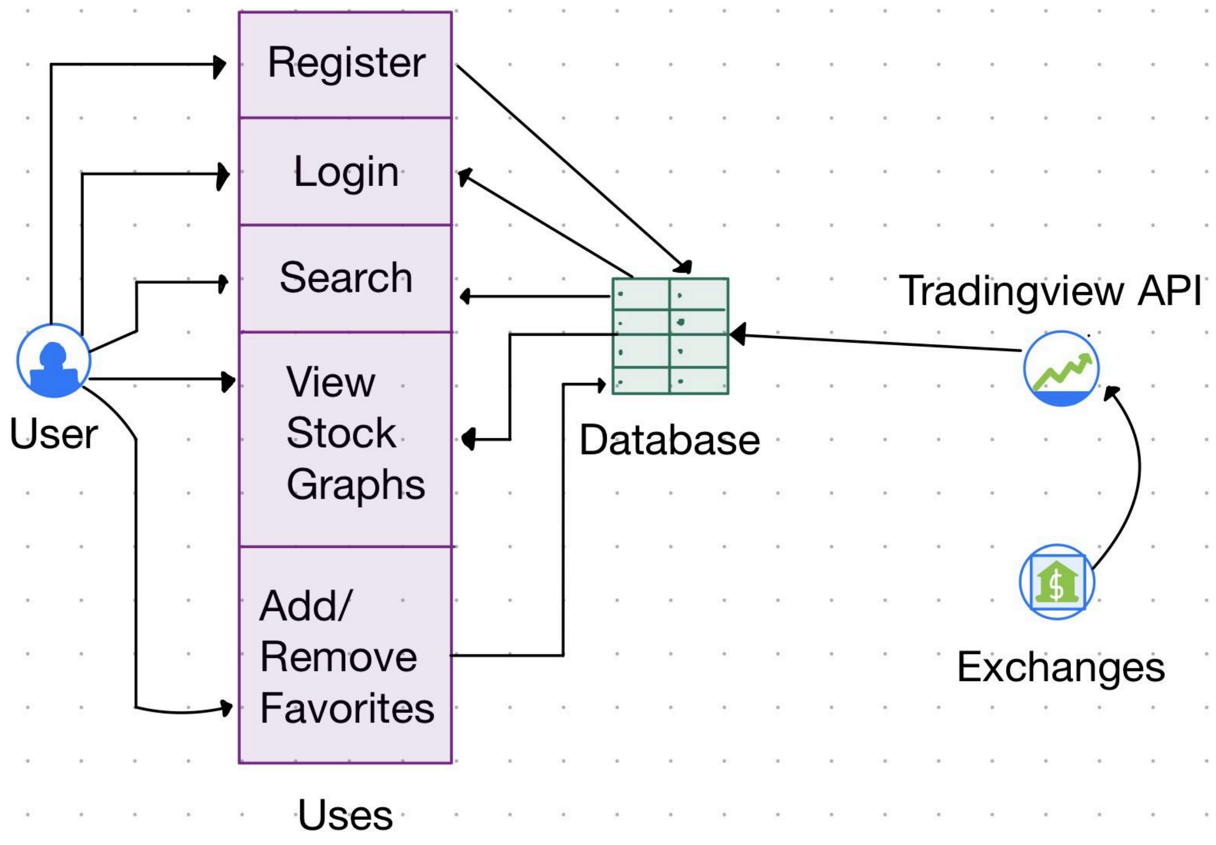
Contributions:

- **Logan:** My immediate long-term obligation for this project was the creation of the search page, which involved querying an API for a list of stocks and then rendering it as cards (the button functionality that linked it with the favorites page was later added by Alex). My second was creating the graph for the portfolio page, which used the external plugin Chart.js to compile processed API data into a visible line graph. Additionally, I also drafted the initial diagrams for the construction of our database, and made numerous alterations to it while work on the pages proper was in progress.
- **Jack:** While working on this project, I spent most of my time constructing the back-end of the application. I implemented many of the endpoints used to access different pages, the authentication middleware to ensure that only users could access the services, and session management ensuring that each session corresponds to a unique user. I also implemented the “favorites” page where the charts for each user’s favorite stocks are displayed. This includes both the

GET endpoint for the page, as well as the handlebars HTML for serving the dynamic webpage.

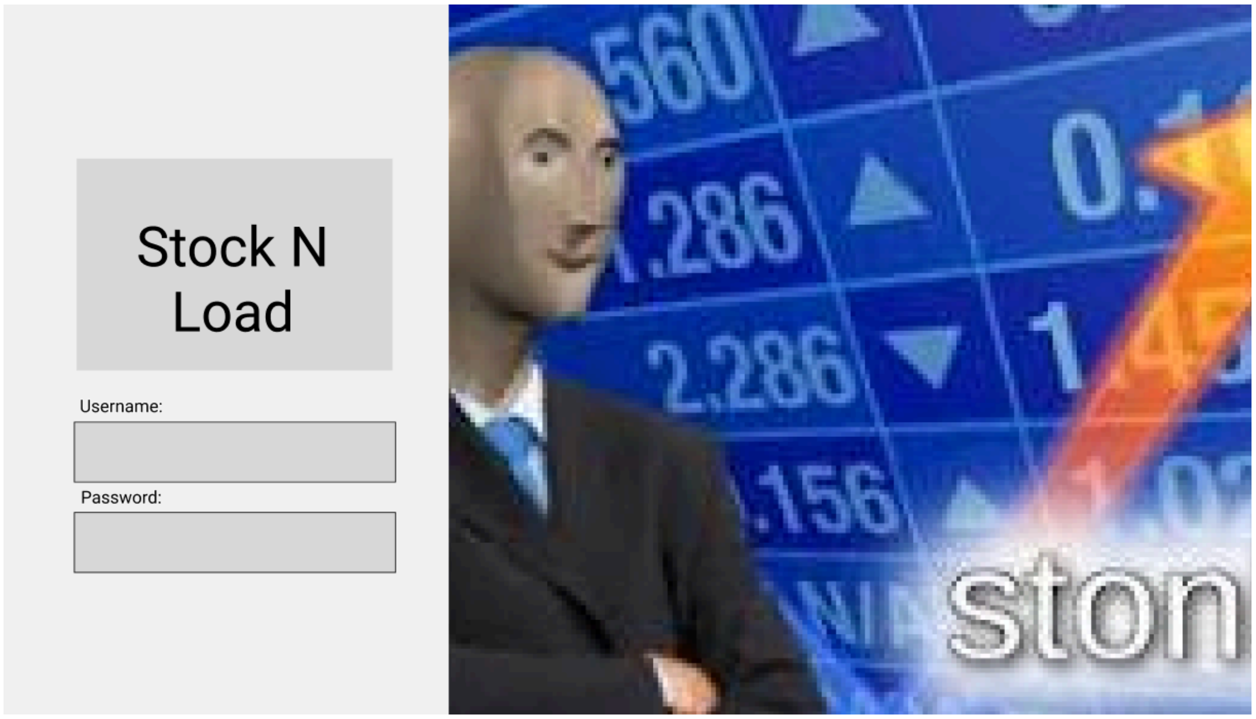
- **Alex:** During this project, I initially focused on developing the backend functionality for the registration features using JavaScript and SQL. I designed the SQL queries needed to interact with our database through the get and post routes. After, I tested using Postman requests and Docker to manage it locally. Following this, I contributed to the early development of the favorites feature on the search page. On the frontend, I developed the early user interface components, which included a button for adding and removing favorites. This was not the final version of the button, but it was primarily used for early testing. On the backend, I implemented the add and remove functionality, handling the logic necessary for updating the user's favorites in the database.
- **Keegan:** I contributed to this project by creating and managing our Github Repository by creating the original directory structure and viewing PR’s. I also helped with login and register endpoints in NodeJS to function as expected with our unit tests, which I created using Mocha/Chai. I created our original DB in PostgreSQL. I helped develop the endpoints when a user adds/deletes a stock from their favorites on the search page. I helped to plan our project, creating TODO lists and coordinating with team members to ensure even workloads and project completion.
- **Eli:** I created the handlebars templates for the front end pages. I also implemented all of the CSS for the pages to format the webpages as well as developing a color scheme consistent with the goals of the project. From the management side, I created the release reports for the GitHub repository and hosted the website with Azure.

Use Case Diagrams:

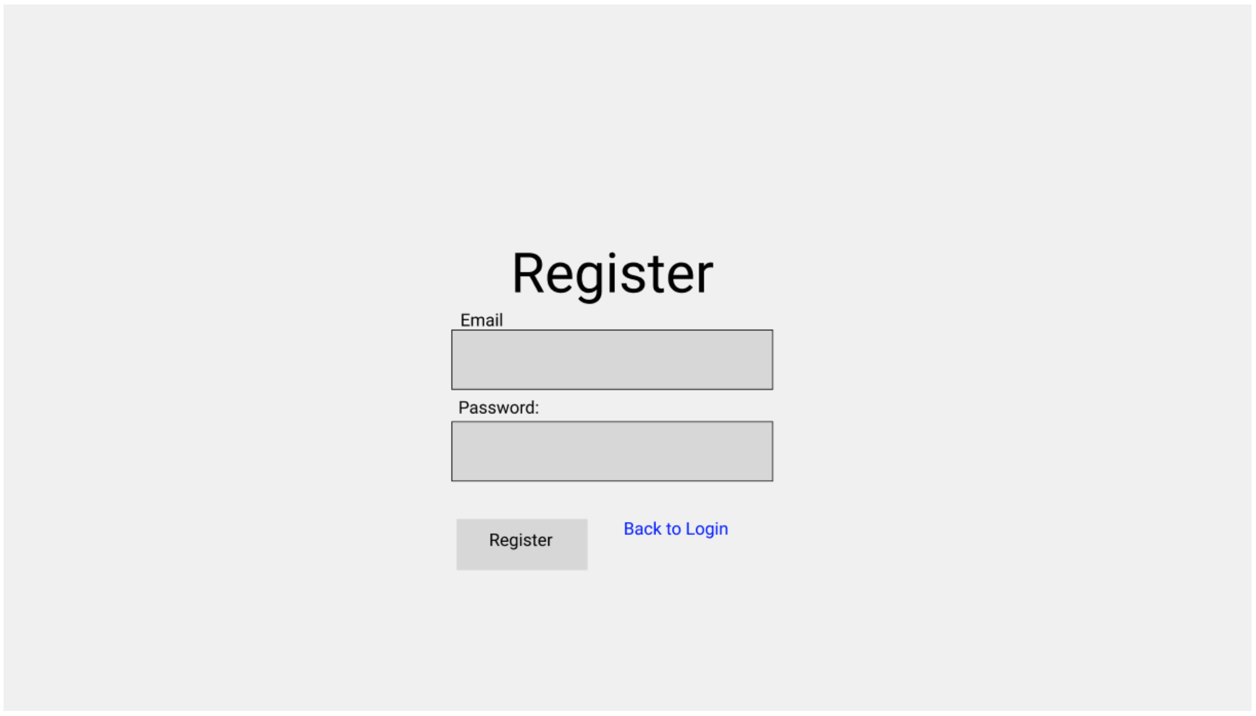


Wireframes:

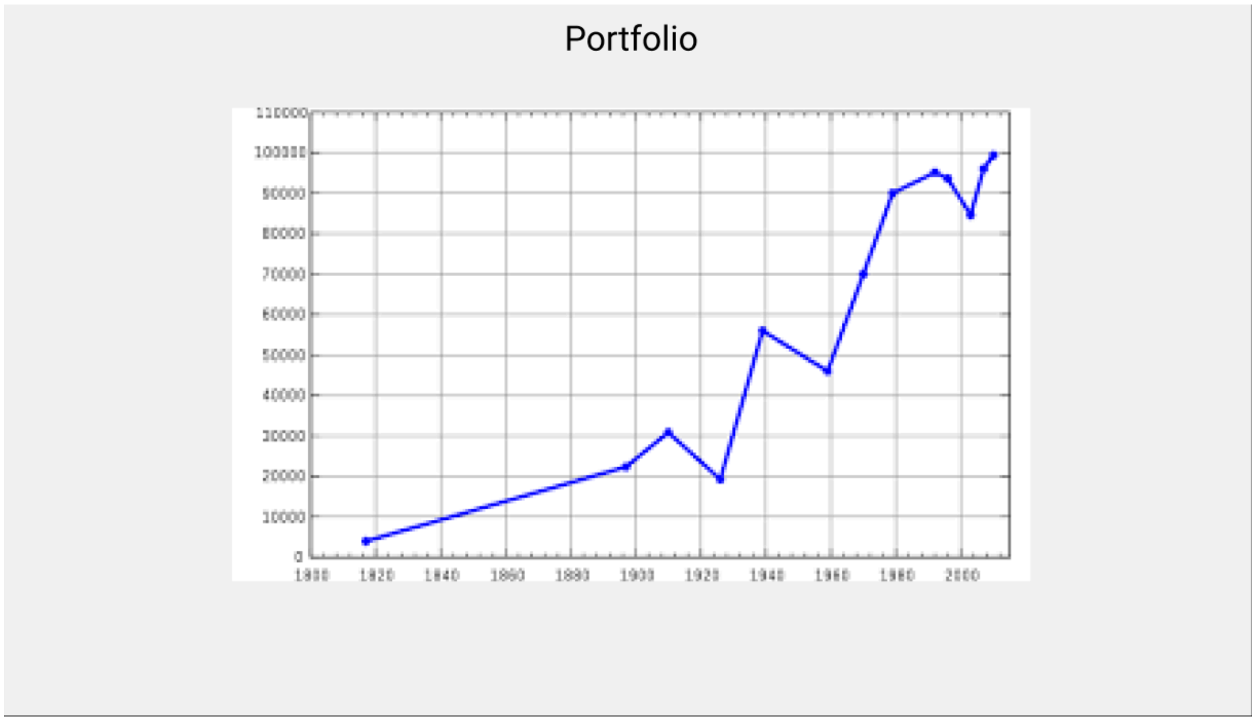
Login Page:



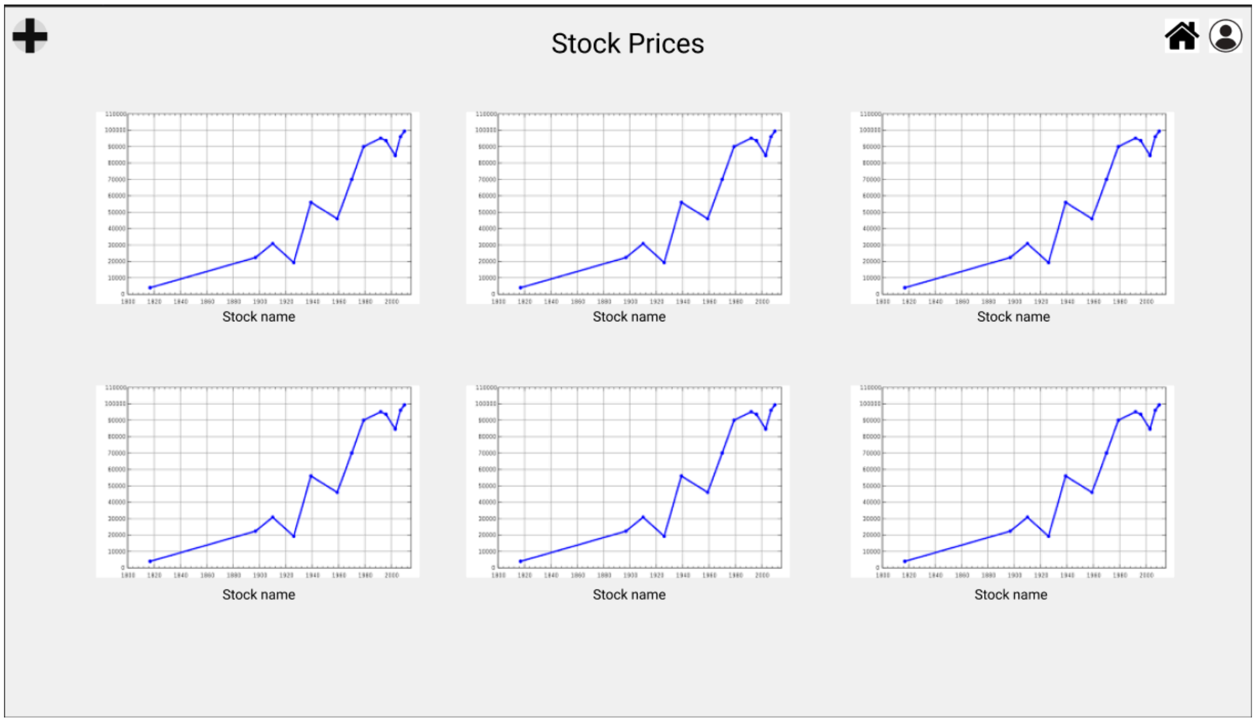
Register Page:



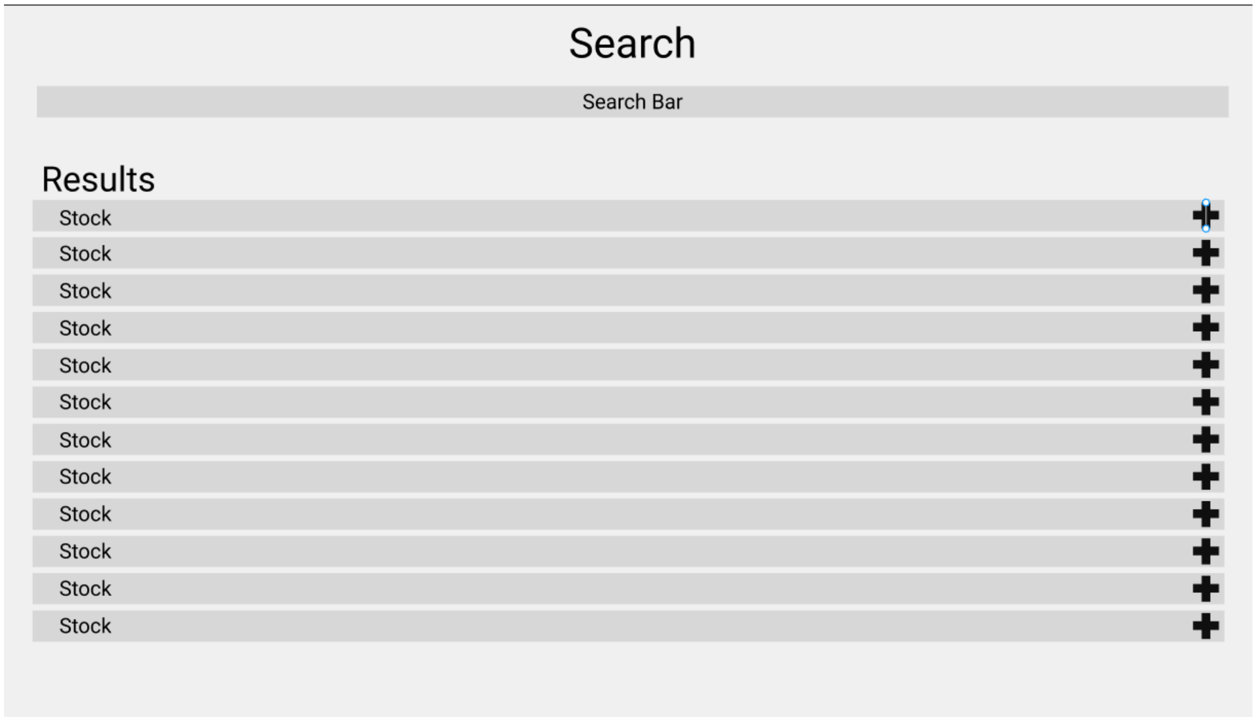
Portfolio Page:



Favorites Page:



Search Page:



Test Results

We created 3 user acceptance tests to test the functionality and flow of our application.

The first User Acceptance Test scenario that we tested was the functionality of the search bar for adding and removing stocks from a users list of favorites. Testing this feature went quite well since it is a relatively simple process to use the search bar. When searching for a ticker symbol for some security, or a company name, the user was able to find the stock from the list of different stocks that matched the entered expression.

The second User Acceptance Test scenario that we tested was the functionality of the add and remove buttons on the search page that can be used to add or remove stocks from a users list of favorites. The buttons that we added for this functionality are very intuitive and the testing went well. The user simply executed a search using the name of some company, then from the list of securities presented to them they were easily able to add the desired securities to their favorites list, and later remove them from the list using the remove button. There is some room for improvement as the users had to think before removing a stock as they didn’t immediately know where to go on the site.

The third User Acceptance Test scenario that we wanted to test was proper displaying of the portfolio page, which was meant to contain a graph that showed users their total portfolio value over time, so they could see how their investment portfolio has been performing. Given the constraint of this being a school project, we were not able to perform API calls back more than three days. We still performed testing where users were able to understand the page and it function as we expected it to, but there was confusion as to why it was only three days worth of data.

Hosting (If not out of credits): dozencrust.com:3000