MockStocks

https://github.com/rt0328/TheMoneyProject

Olivia Brobin
John Cates
Rissan Karki
Jake Lowry
Grace Ohlsen
Ryan Townsend

Description

MockStocks is a website that allows users to simulate stock trading without having to invest any real money. Users can engage in this practice in a game-like setting with their friends using the groups feature, through which they can compete for the highest returns. Groups can be joined by any user with a unique code, and the creator of the group (admin user) can edit or delete it. Within each group, a leaderboard containing each user's portfolio information is displayed, as well as a custom icon and its name.

The data for real-time stock prices is pulled from the free Finnhub API. Backend functions were written to search through available stocks, recommend to beginners which stocks to buy, and add the stocks to a user's portfolio. The HTML/CSS/JS library Bootstrap was used for the front-end design. Modals, cards, carousels, and forms are used extensively to make the website approachable and easily graspable for new users. Handlebars was used for HTML templating and dynamic page rendering. PostgreSQL databases are used to store user/group information, cache stock prices to reduce API calls, and link users to groups.

Contributors

Olivia Brobin (OliviaBrobin)
Olivia handled most of the front-end design. She worked with Bootstrap to make the UI similar to that of the wireframe prototype, most particularly with the homepage and the login/registration pages.

John Cates (joca7396)
John designed the group code randomization functionality and helped with several front-end tweaks. He also created the presentation and dealt with getting everyone coordinated.

Rissan Karki (riskarki)
Rissan helped design the initial wireframe, assisted with front-end design, and dealt with various administrative tasks, such as writing reports, creating the UATs, and recording the project demo.

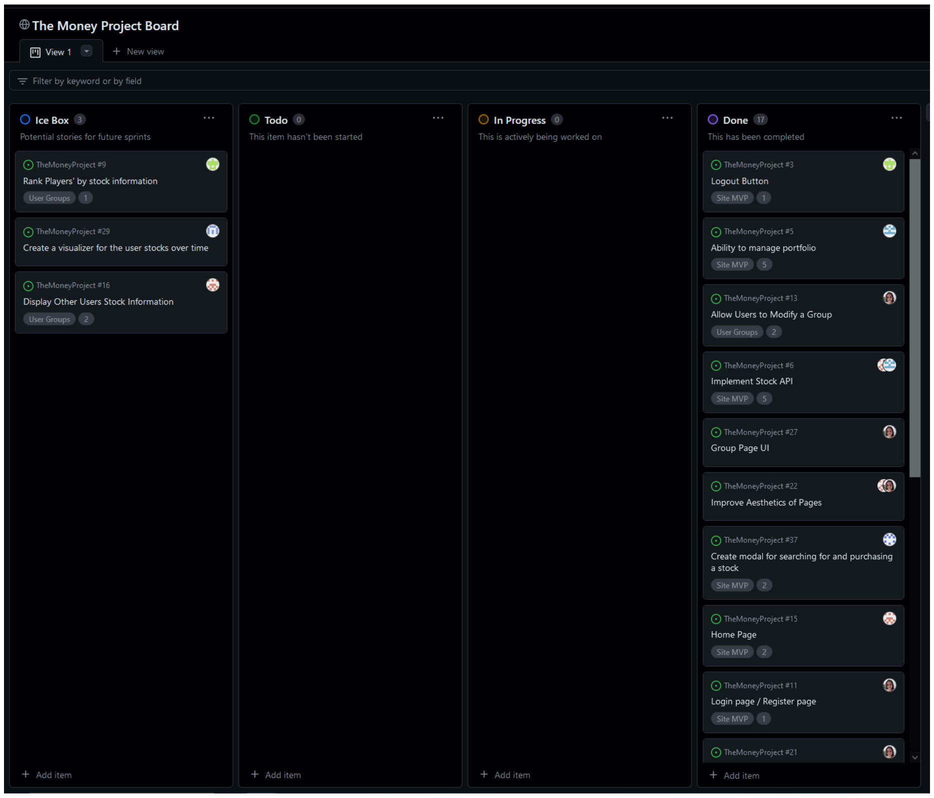
Jake Lowry (jakelowry)
Jake handled connecting the Finnhub API and wrote the backend functions that display real-time prices and allow users to search for stocks of their choosing.

Grace Ohlsen (sko-grace)
Grace designed the groups functionality (adding members, admin permissions, editing name/icon, etc.) and worked with the database structure to ensure that it worked without any bugs. She also helped in creating the initial wireframe and writing backend code.

Ryan Townsend (rt0328)
Ryan designed the SQL databases and the relationships between tables, as well as the Chai.js tests. He also dealt with the cloud computing component and assisted in several front-end tweaks.

Project Board

https://github.com/users/rt0328/projects/1/views/1



The Money Project Board

Ice Box 3
Potential stories for future sprints

- TheMoneyProject #9 — Rank Players' by stock information — User Groups 1
- TheMoneyProject #29 — Create a visualizer for the user stocks over time
- TheMoneyProject #16 — Display Other Users Stock Information — User Groups 2

Todo 0
This item hasn't been started

In Progress 0
This is actively being worked on

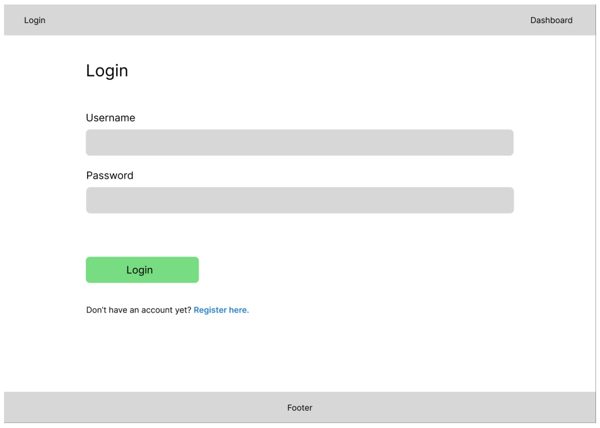
Done 17
This has been completed

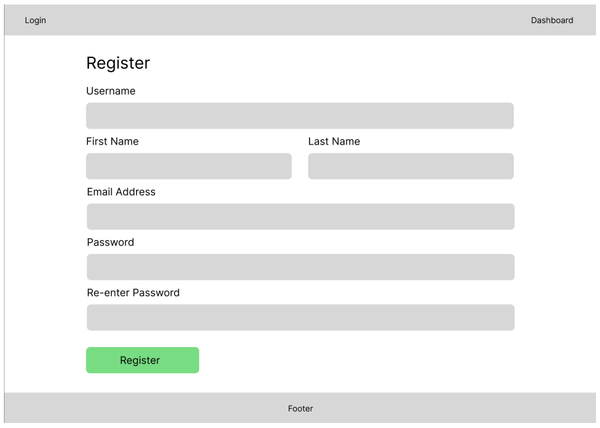
- TheMoneyProject #3 — Logout Button — Site MVP 1
- TheMoneyProject #5 — Ability to manage portfolio — Site MVP 5
- TheMoneyProject #13 — Allow Users to Modify a Group — User Groups 2
- TheMoneyProject #6 — Implement Stock API — Site MVP 5
- TheMoneyProject #27 — Group Page UI
- TheMoneyProject #22 — Improve Aesthetics of Pages
- TheMoneyProject #37 — Create modal for searching for and purchasing a stock — Site MVP 2
- TheMoneyProject #15 — Home Page — Site MVP 2
- TheMoneyProject #11 — Login page / Register page — Site MVP 1
- TheMoneyProject #21

We completed most of what we intended to get done initially, but some minor tasks remained incomplete because of a lack of time. The visualizer was to be implemented either using Google Charts or Charts.js and would pull data from an SQL table that stored users' portfolio value over the course of a game. Global rankings would use queries on the users table and sort them by balance.

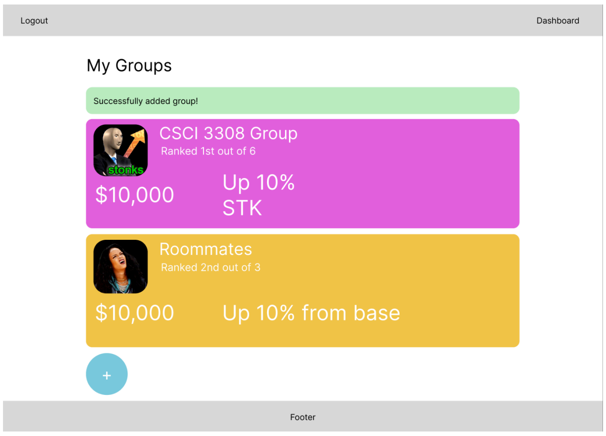
Wireframing

Login/Register

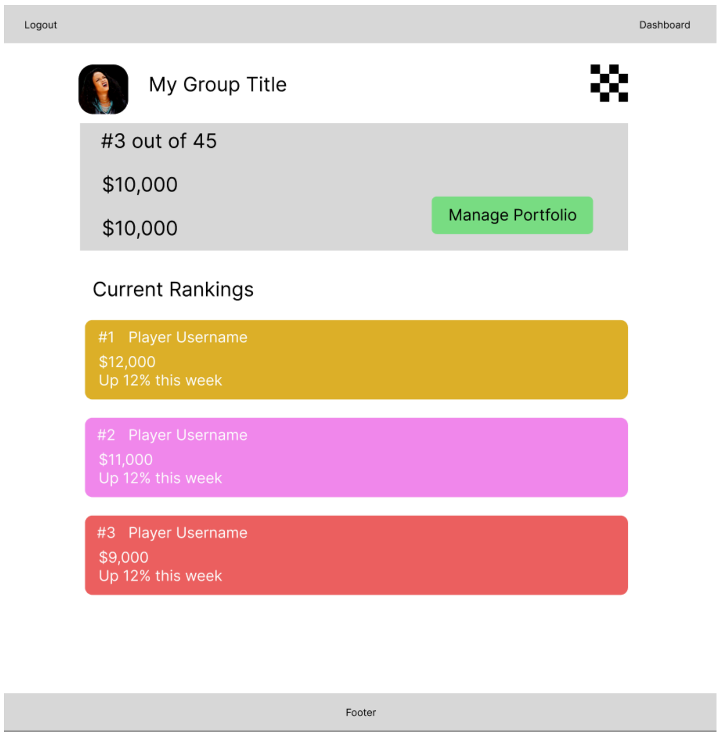




Dashboard



Group Homepage



Logout

Dashboard

My Group Title

#3 out of 45

$10,000

$10,000

Manage Portfolio

Current Rankings

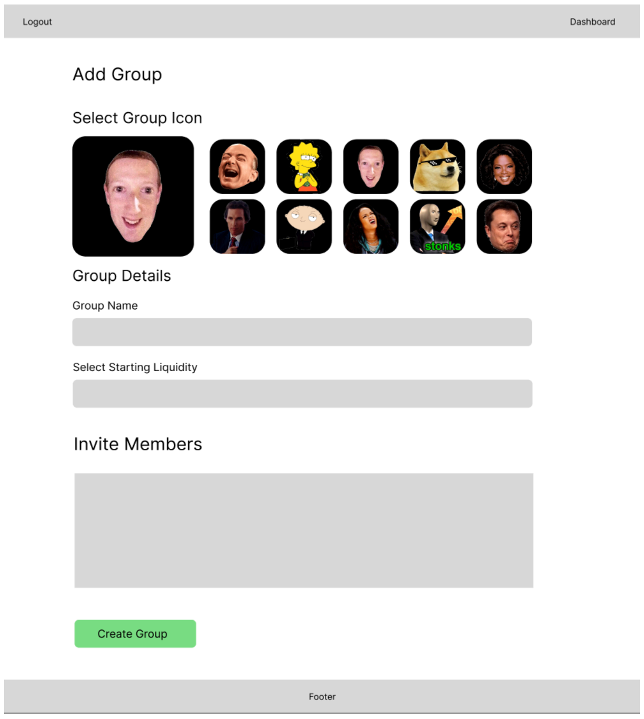
#1 Player Username
$12,000
Up 12% this week

#2 Player Username
$11,000
Up 12% this week

#3 Player Username
$9,000
Up 12% this week

Footer

Add/Edit Group



Logout

Dashboard

Add Group

Select Group Icon

Group Details

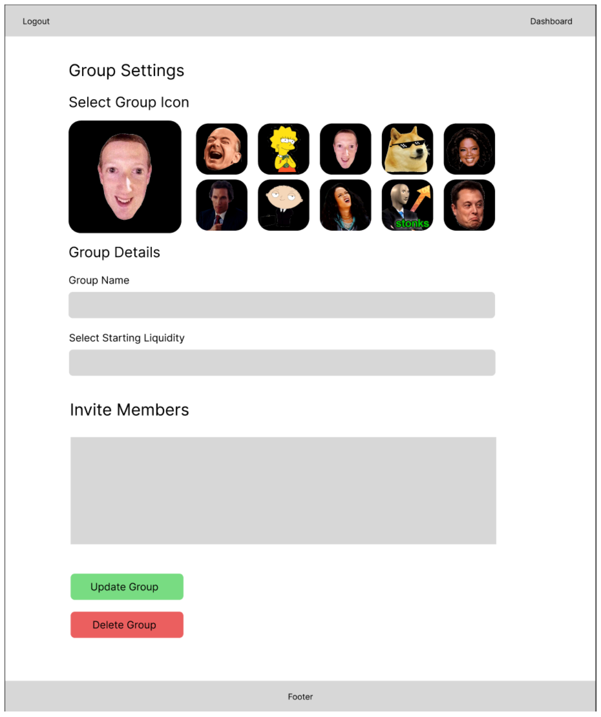
Group Name

Select Starting Liquidity

Invite Members

Create Group

Footer



Logout

Dashboard

Group Settings

Select Group Icon

Group Details

Group Name

Select Starting Liquidity

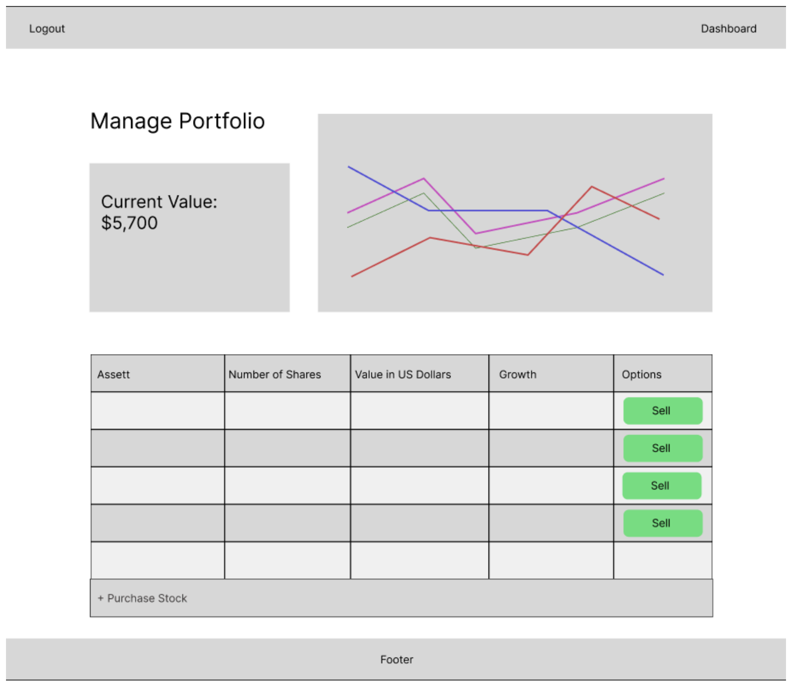
Invite Members

Update Group

Delete Group

Footer

Portfolio



Logout

Dashboard

Manage Portfolio

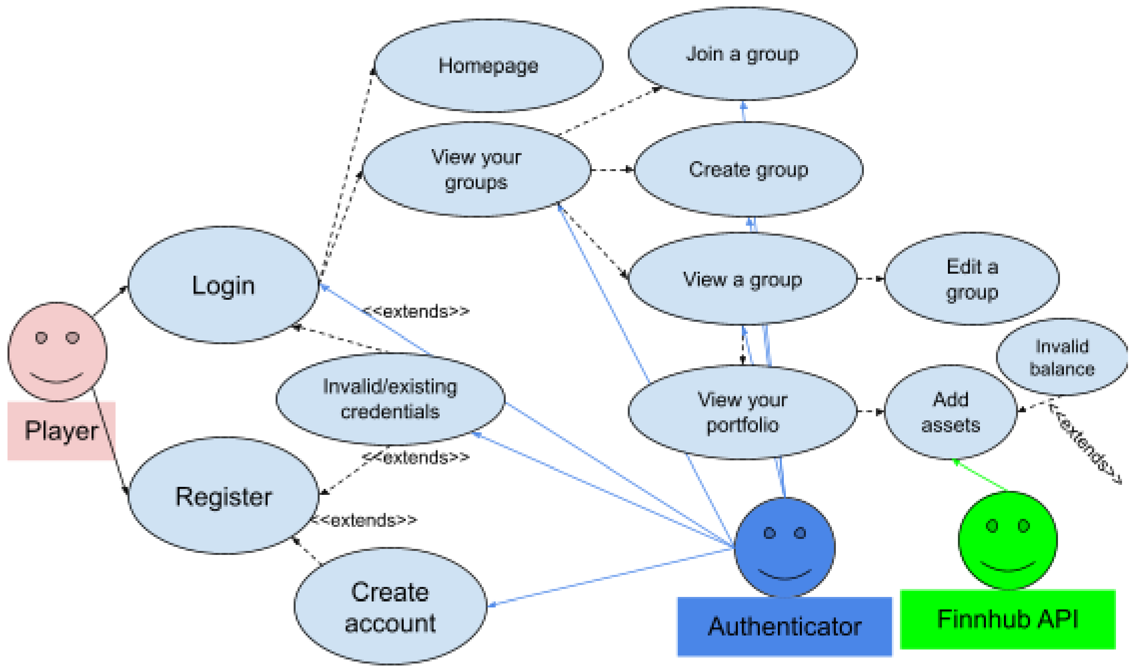
Current Value:
$5,700

Assett	Number of Shares	Value in US Dollars	Growth	Options
				Sell
				Sell
				Sell
				Sell

+ Purchase Stock

Footer

Use Case Diagram



<u>Test Results</u>

UAT 1: Testing that a user can log in with correct credentials
<u>Test case:</u> A user is asked to enter a username and password of their choice into the login form. Assuming this is their first time on the system, the system will ask them to register an account, then they should be able to log in. Testing this feature would use test data added to the user table in the users_db database. This is a localhost test.
A successful test will have users access the portfolio page.
<u>Observations:</u> When the user was greeted with the login page, they instinctively entered a username and password. The red “invalid credentials” card appeared at the top of the page, and they then navigated to the navigation bar and to the registration page, where they entered the same information. They were then redirected to the login page, where they were able to enter their information and enter the website. This feature worked exactly as intended without much deviation from what was expected; the placement of the “Register” button in the navigation bar took them a little longer to do so than if the button were, say, under the text fields on the login page.

UAT 2: Testing that a user can purchase/sell stocks
<u>Test case:</u> Once a user is logged in, they will click on the manage portfolio button, and click add a stock. For this test, they will try to buy one stock of their choice, then sell it. Testing this feature would involve adding and removing data to the stock table in the users_db database. This can either be done on a localhost or the cloud. A successful test will be done when a user can purchase a stock of their choice, view relavent data about the stock, and sell it and recieve their money back.
<u>Observations:</u> Once the user had created or joined a group, they went to their portfolio and clicked on the button to add a stock. Initially, the list of stocks was empty, but once they entered “a” into the search field, the stocks and their current prices loaded one by one into the modal. They selected the “AAPL” stock and entered in hundreds of shares, but were denied from doing so because they were attempting to purchase an amount greater than their balance. They decreased the number of shares and were able to view their updated portfolio and balance. This, too, worked as intended.

UAT 3: Testing that a user can create a group
<u>Test case:</u> For this test, we will have two predefined users set up (User A and User B). User A will create a group and invite User B to the group. User B will accept the invatation to the group. Testing this feature will involve preset data in users table, and create new data in the group table and users_to_group table. All of these tables are in the users_db database. This is a localhost test. A successful test will have both users entered into a group, where both users are able to view each other's stock purchases and growth.

Observations: User A creates a group and shares the code with User B aloud. User B enters the code and is able to view the group in their homepage. When they enter the group's page, they are able to see User A's position on the leaderboard. This worked as intended functionality-wise, but it was slightly unexpected that they read the code aloud to each other. This is a positive side effect of the feature, because the unique codes generated are simple enough that it isn't a hassle for a user to join a group.

Deployment

Because of the cost of hosting this project on a cloud platform is too high, you can access it by pulling the repository (https://github.com/rt0328/TheMoneyProject), navigating to the `ProjectSourceCode` directory, running Docker (`docker compose up -d`), and entering `localhost:3000/` in your web browser.