TradeHub Investment Simulator
Project Report

COMP3900 - Computer Science Project

Group:

H11A-UNDEFINED

Team Members:

Domenic Prestia - z5061488
Jet Young Lim - z5195170
Jiaying Yi - z5135198
You Shi Zhu - z5060508
Yukun Shi - z5141536

Submission Date:

16/11/2020

Table of Contents

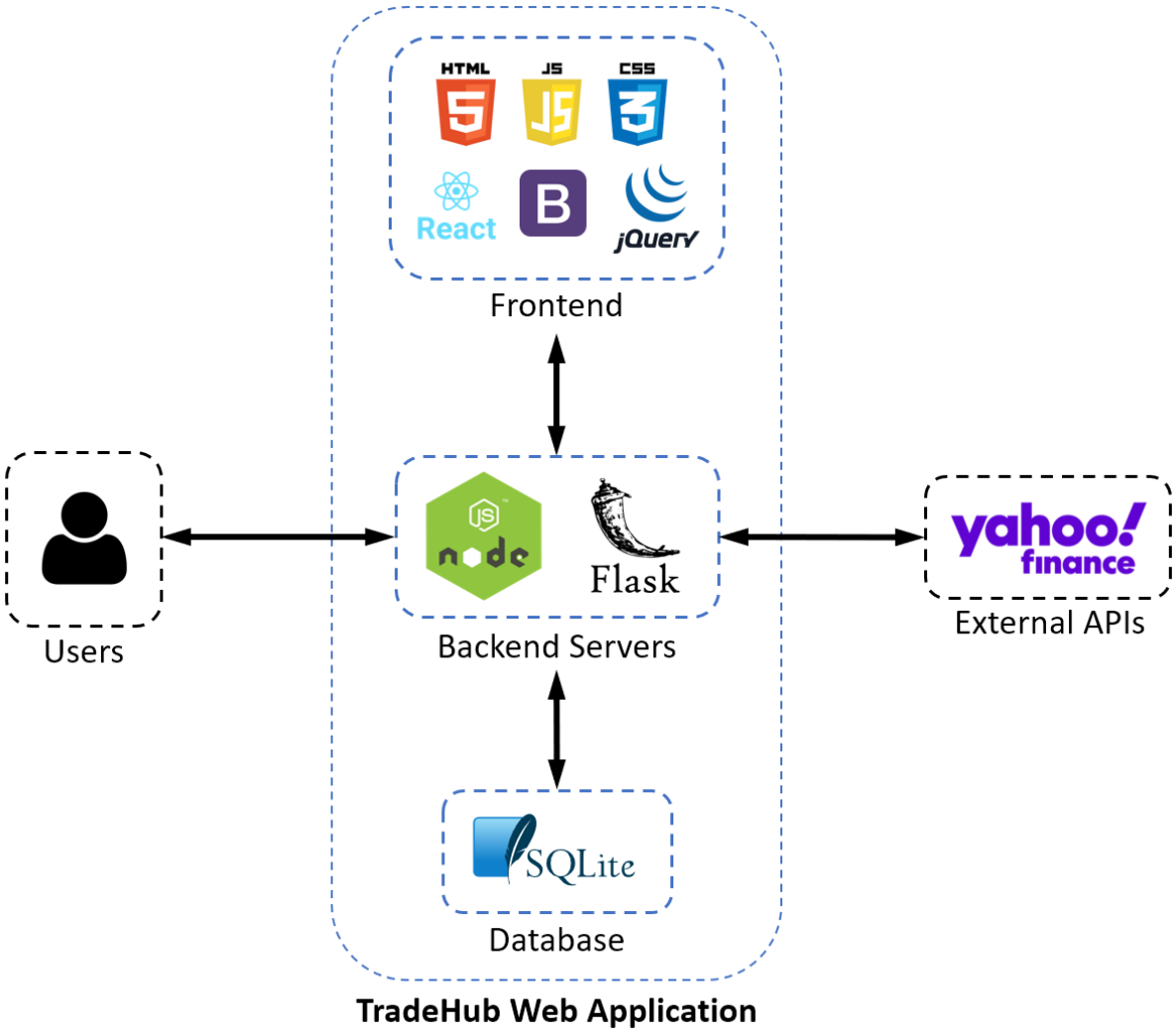
1. Introduction

Investment simulators are useful tools to traders as it allows them to practice their skills and apply knowledge in a risk-free, zero cost environment.

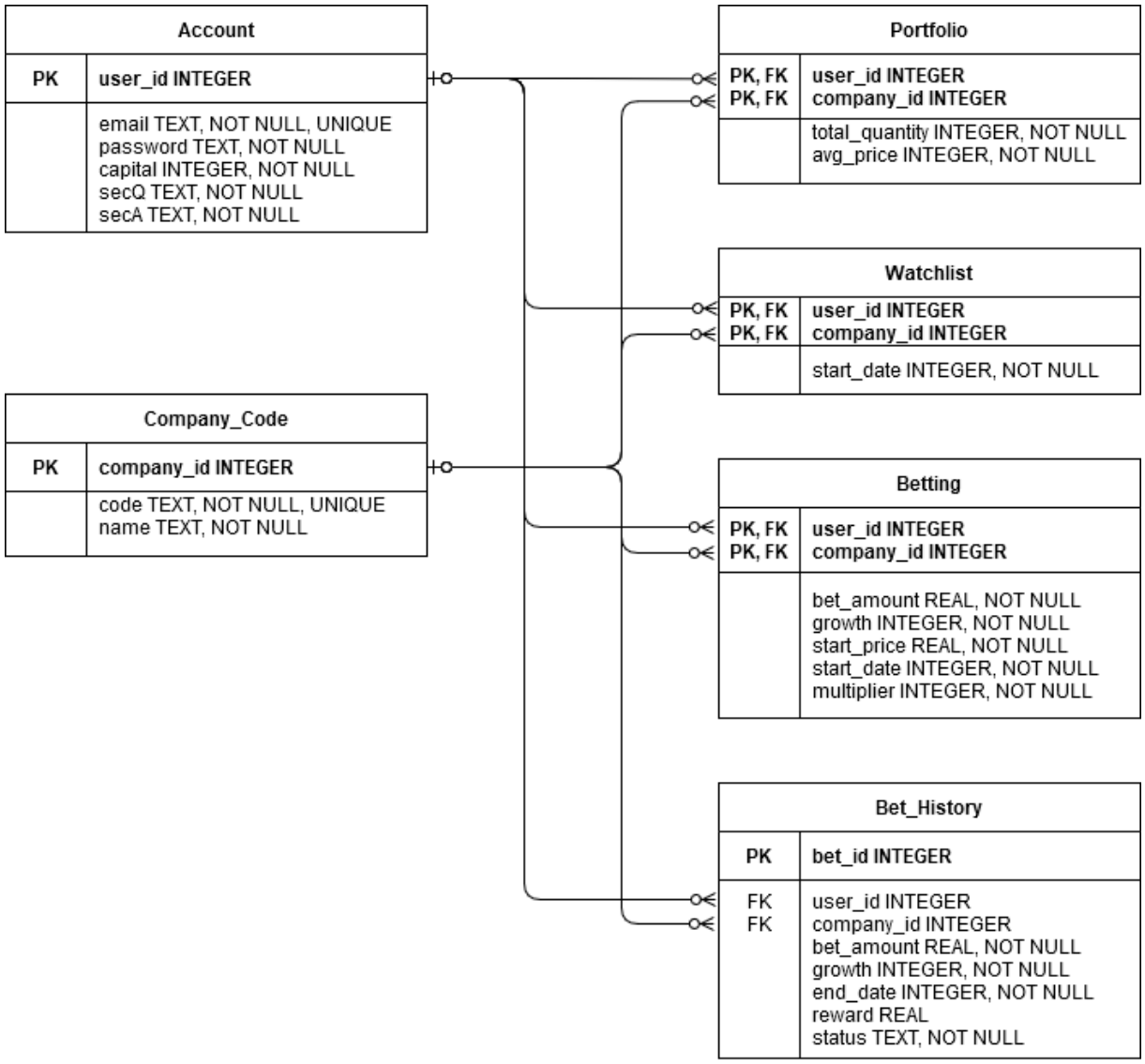
We have created an investment simulator that provides core functionality requested in the project objectives as laid out in the proposal. It also provides additional novelty features not present in current investment simulators.

In this report we describe the system architecture and third party software used, detail the functionalities that were implemented, discuss implementation challenges and include a detailed manual for using our system.

2. Overview



Users interact with the web application via HTTP requests to the NodeJS server environment, which serves the relevant HTML, CSS, JS, etc. resources back to the user's browser to render. There is no distinction in access points between unregistered and registered users. NodeJS allows users to interact with our web application and access TradeHub's functionalities. SQLite is a lightweight, fast, self-contained public domain relational database engine that allows storing user-specific information such as their credentials and portfolio. The relational schema is shown below:



Unregistered users only have access to a subset of the full functionality offered to registered users. POST requests to the backend servers are checked to see if they contain a user cookie that indicates the user is logged in. Registered users will have a valid cookie which allows them access to the full functionality of the web application, such as trading stocks and adding them to their watchlist.

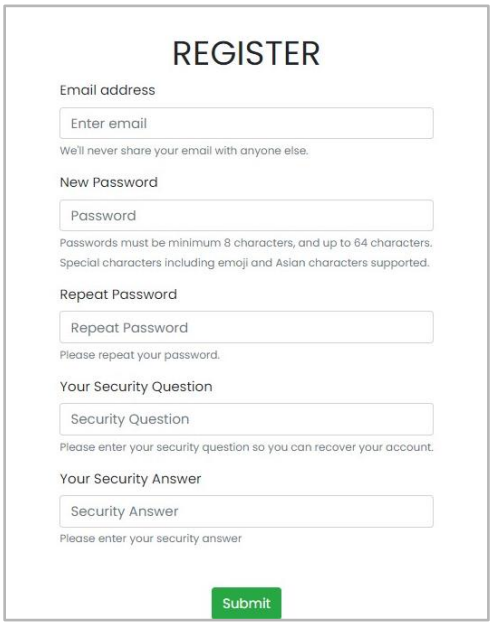
React is a JavaScript library that uses HTML, JavaScript, CSS and jQuery to fetch and render the user-interfaces/components. React has routers that handle different responses depending on the route. The routers load the correct components that are then served back to the user via NodeJS. React simplifies the process of creating interactive UI elements, allowing users to interact with the functionalities of TradeHub in a clean, easy to navigate interface.

For HTTP requests that require backend involvement, NodeJS proxies requests to Flask. Flask is a lightweight web microframework written in Python, allowing it to interact smoothly as part of a Python-based back-end. The Python backend processes data, performs API calls to yfinance and manages the database, and returns results through Flask back to NodeJS to be placed into React components and served to users. Thus, the backend performs all of the necessary functions that support the functionalities provided by the website such as registration, trading, and watchlists.

3. Software Functionalities

3.1. Account Creation

New users are able to access the register page by clicking the signup button at the navigation bar. Creating an account is not in the core project objectives, but it is needed to track user trades and other information. New users are able to register by entering email, password and security question and answer. Error messages will display in following scenarios:



- This email is already registered.
- The format of Email is invalid.
- The password doesn't follow the NIST password guidelines as detailed
- The password and its repeat are not identical.
- Any of the input fields is void. In order to prevent web attacks including malicious escape codes and SQL injection, all of input will be sanitized.

The password, repeated password field and security answer will be hashed before storing in the database. By clicking the Submit button, the backend of the website will check all the input data. If all the inputs are valid, the user will be redirected to the account page. If not, the proper error message(s) will be displayed.

3.2. Logging in/out

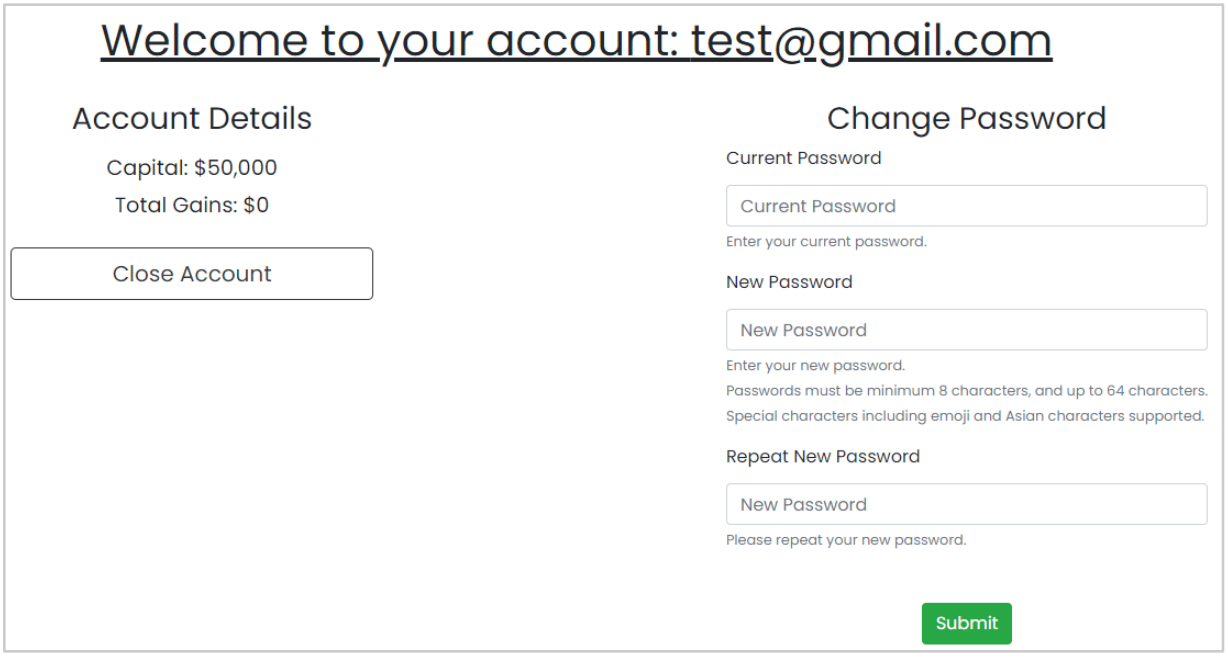
Users who create an account can login/logout of their account. Users are automatically logged into their newly created accounts when they register, otherwise there is an easily accessible login button on the navigation bar.

Although logging into an account is not strictly mentioned in the project objectives, in order to be able to track a user's trades, users must have an account and be logged in before they can access features such as trading.

To prevent SQL injection, the email input field is sanitised when passed to the backend. To slow down brute force attempts at obtaining credentials, the login page does not indicate whether unsuccessful attempts were due to a problem with either the email or the password, and after 4 unsuccessful attempts in a short amount of time, the page automatically locks for 10 seconds.

3.3. Account Details

Users are able to see details specific to them and access account related functionalities on their 'Account' page. This meets the project objective of having an accounts details page.



Users can see their account details, which include how much capital they have (money to spend on buying stocks) and their total gains. Total gains tells a user how much total money they've earnt since they've started trading. Total gains is equal to a user's current total assets value (how much all stocks they own are worth) plus their current capital, minus their starting capital.

This page also contains the change password functionality which is explained in its own section. This page contains a link to the close account functionality which is also explained in its own section.

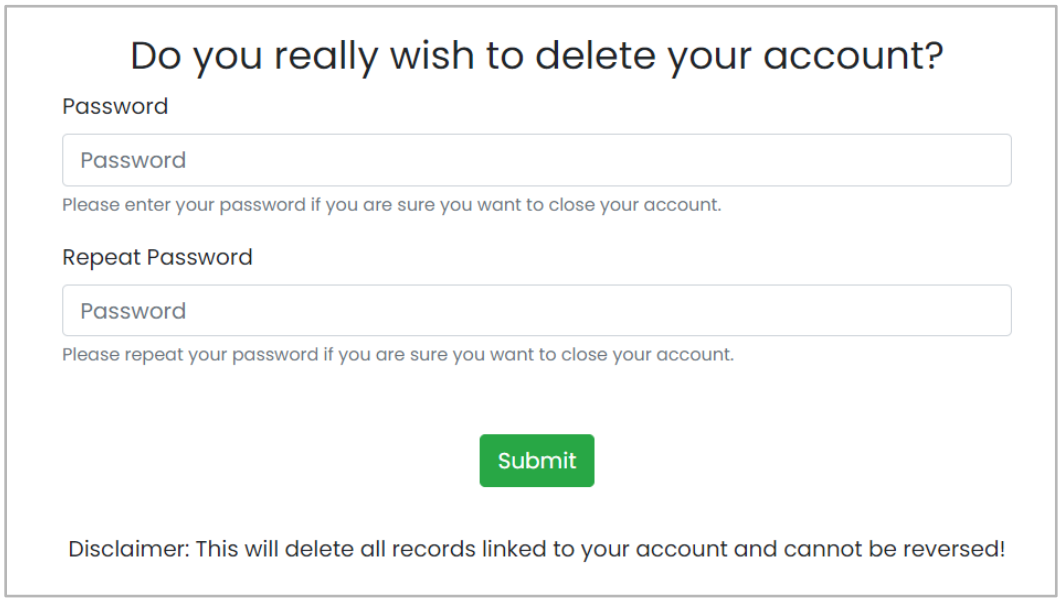
3.4. Password Reset

Users can change their password if their password becomes compromised, or if they have forgotten their password. Although password resets are not strictly mentioned in the project objectives, this functionality was considered to be a necessary part of a web application that allows creation of accounts, and added to the functionalities of the web site.

Logged in users can easily change their password on their account page by entering their current and the new password. If a user has forgotten their password and wants to recover their account, the Login page has a button that navigates to a Forgotten Password page. By entering their email, the answer to their security question, and their new password, users can recover their account and set a new password. If unsuccessful, the respective error messages will be printed on the page, e.g. Incorrect answer to the security question, invalid password, etc.

3.5. Delete Account

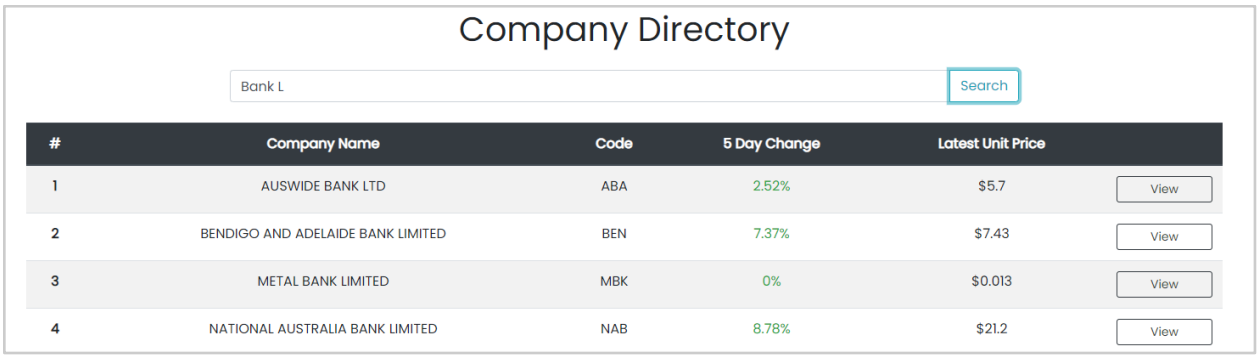
Registered users are able to delete the personal account by clicking the close account button in the account page. Users will need to input the password twices in order to proceed to the close account. If the users enter the wrong password for the account or unmatch between password and repeated password, the error message will pop up in the model. After the user inputs the correct password for this account, the cookie will be removed and all the records in the database which are associated with this account will be deleted. Users will be rendered to the main page after five seconds. Although delete accounts are not strictly mentioned in the project objectives,this functionality was considered to be a necessary part of user experience.



3.6. Stock Searching

Unregistered or registered users can search for stocks they are interested in on the ‘Market’ page. This meets the project objective of users being able to search for stocks using a stock code.

Information displayed is the company name, the stock code, the performance over the last 5 days and the latest stock price. Initially displayed are the first 10 stocks, sorted alphabetically by stock code. Users can search by stock code or company name.



The search will return the first 10 stocks that substring match the search query, sorted alphabetically. Users are able to click a link to go the details page of a specific stock. If there are no search results, the users are notified.

3.7. Stock Details

Users are able to see the details of a specific stock in the stock details page. By fetching the API, the user stock details page displays stock name, market capacity, current price, daily volume, number of shares and annual yield. Meanwhile, a table displays closing price, highest price, lowest price, daily change and volume for the latest five trading days. This meets the project objectives of having a page that lists specific stocks statistics.

Users are able to see graphs display the historical prices on each stock details page, so that they can have a visual reference of stock performance. This meets the project objectives of users being able to view historical data, such as the latest stock closing price, in a graph. The graphs are candlestick charts and set at time versus price (in AUD). Only the latest trading day’s information would be displayed on the stock details page.

By placing the mouse on each bar, a pop up dynamic window would display the open, high, low and closing price in that specific minute. If it’s trading time, the chart would update each minute's information automatically. Users are also able to zoom in to get a more detailed view by selecting a period of time. In the case of zoom in, users are able to drag the chart horizontally or reset to the overview.

Users are able to view a graph showing the historical daily closing unit price for any stock on their watchlist, where historical data are available from a week before the day when the stock was added to the watchlist to now. The functionalities of the graphs in watchlist are the same as the graphs in the stock details page.

3.8. Stock Trading

Users can ‘buy’ and ‘sell’ stocks, allowing them to ‘simul-own’ stocks as set in the project objectives. Users can trade stocks on the details page of any stock they search, or other means provided by certain pages such as the Investment Advisor page. While any trading on the TradeHub platform is only a simulation of trading on the real market, the stock price and other details reflect the real condition of the Australian Stock Exchange. Thus, holding stocks on the TradeHub platform simulates holding stocks on the real ASX.

3.9. User Portfolio

The portfolio page keeps track of the current stocks that are being traded by the user. This meets the project objective of having a page that aggregates statistics of each owned stock, displays the profit/loss of each individual stock and displays the total profit/loss for all stocks. Verified and users who have logged in will be able to have full access to the portfolio page features. The page will display the basic stock information along with investment information such as the amount own, total value, the total amount paid and the profit loss for each of stocks that the user invested in.

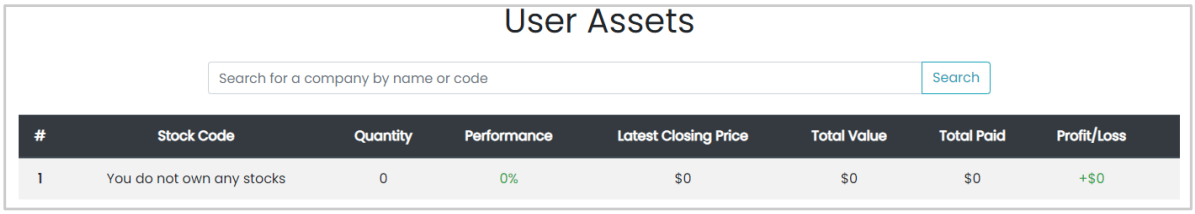
#	Stock Code	Quantity	Performance	Latest Closing Price	Total Value	Total Paid	Profit/Loss	
1	CBA	5000	4.03%	$73.14	$365700	$365700	+$0	View

The system automatically adds a new stock into the page when a buy order is executed successfully and if the user decides to sell all of a particular stock it will automatically remove it from the list. If a user decides to purchase or sell a specified amount of stocks while having the stock added to the page already, the system will update the existing investment information instead.

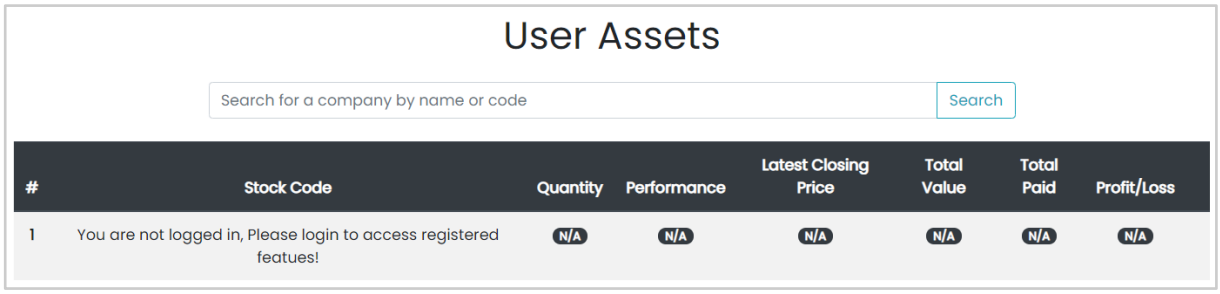
Additionally, the system keeps track of the overall total profit/loss, total amount paid and total revenue of all the stocks that the user currently owns.

Optionally, there is also a search feature in the stock page to find a specific stock that the user currently owns. The search function will lookup any stock whose name and code matches the search criteria. Unlike other search functions in different pages, there is no limit as to how many stocks can be returned from the search in the user portfolio page.

If a user does not have any existing investment to keep track of, the system will display a message on the page telling the user that they do not have any investments.



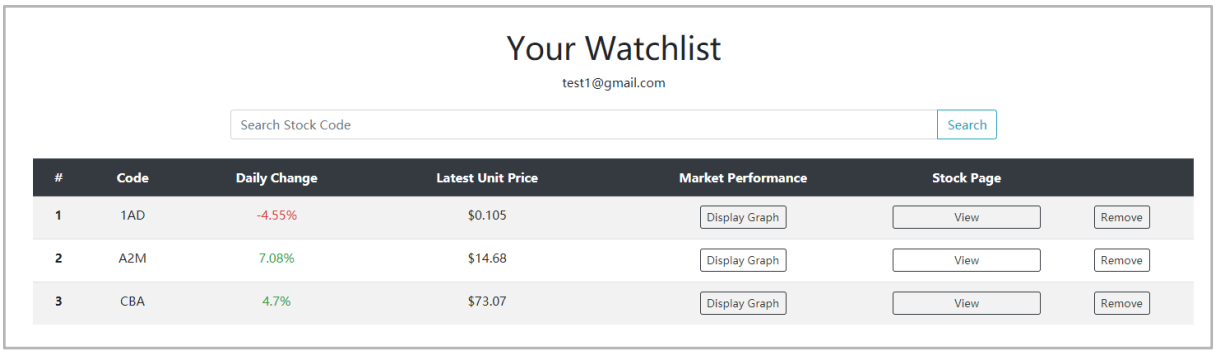
If an unregistered user or users who are not logged in tries to access the page, it will display no information and show a message to tell the user to register and login.



3.10. Watchlist

Registered users are able to access their personal watchlist through the navigation bar. The front-end of the website will check if the user has logged in, the reminder will be displayed if there is no cookie on the web browser which indicates users without login in. The basic information of stock performance including company code, daily percentage of change which is calculated from comparing the latest unit price of stock to the previous day's unit price is displayed on the page. Users are able to search stocks by company code. Any stock code matched partially by the user's input will be displayed.

Users are able to see the stock performance graph depicted from historical daily performance. Also, users are able to delete the stock by clicking the remove button. Users are able to access the stock detail page. Registered users are able to add or remove stock from the watchlist though the stock detail page .This meets the project objective of having a user watchlist.



3.11. Investment Advisor

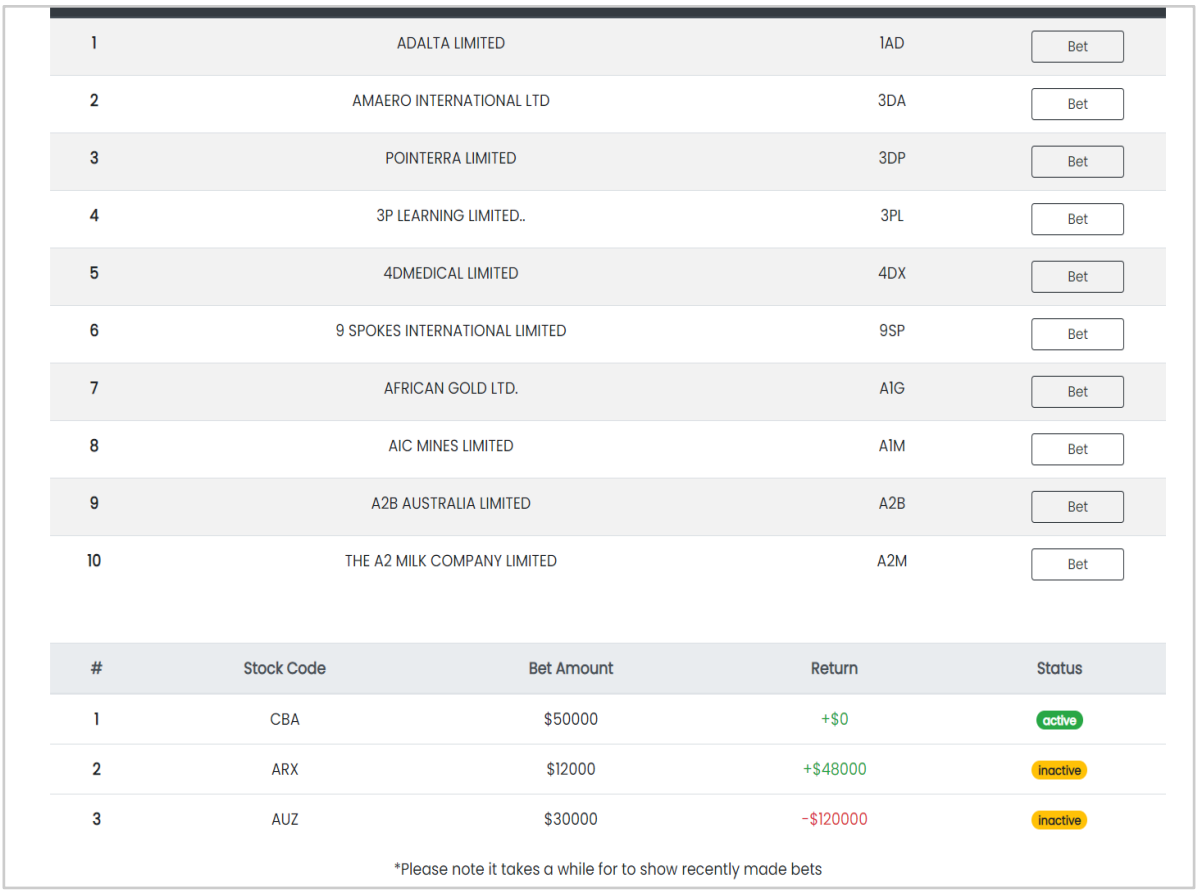
Research into investment strategies has shown that a randomised strategy typically outperforms strategies based on stock performance/historicals in the long-term, and is also less volatile. Famous examples of this are when different animals such as cats and monkeys are used to randomly choose stocks on a board, facing stock trading experts. The randomly chosen stocks outperformed both the market index and those chosen by stock trading experts. To reflect this, TradeHub has an Advisor page that suggests stocks to the user to trade. This is a novel feature that has not been implemented in any stock market trading simulators, possibly due to how the concept seems unconventional on face value.

3.12. Betting System

Users who are logged in and verified are able to access the betting feature in the game page. The betting page initially displays a list of top 10 stocks similarly to the market page. At the bottom of the bet options, the page also displays the users' bet history showing the currently active and inactive bets that the user made. Though the database will only keep track and maintain up to 3 different bets whether it be an activate bet or inactivate bet for each users' bet history. Note that the active bet will always be prioritised and placed above the inactive bets in the bet history table. This meets the additional novelty function of having a betting system that can reward the user.

The user is able to search for a specific stock to place a bet on through the search bar provided in the betting page. The search function goes through a list of filtered and valid stocks and returns the top 10 stocks with the stock name and stock code associated with the provided search word.

The betting game starts by getting the user to select a stock and place a bet on it with a choice to determine whether the selected stock's price will increase or decrease in a week. Once the user places the bet, the betting information gets processed and stored in the database. After a week, the system automatically updates the database and determines whether the user has won or lost the bet. It also automatically calculates the gain or loss for the user's return in the bet. The system will then take the return and update the user's capital with it, the user's capital might increase or decrease depending on whether the return is a gain or loss. Note that the user can make or have at most 1 active bet.



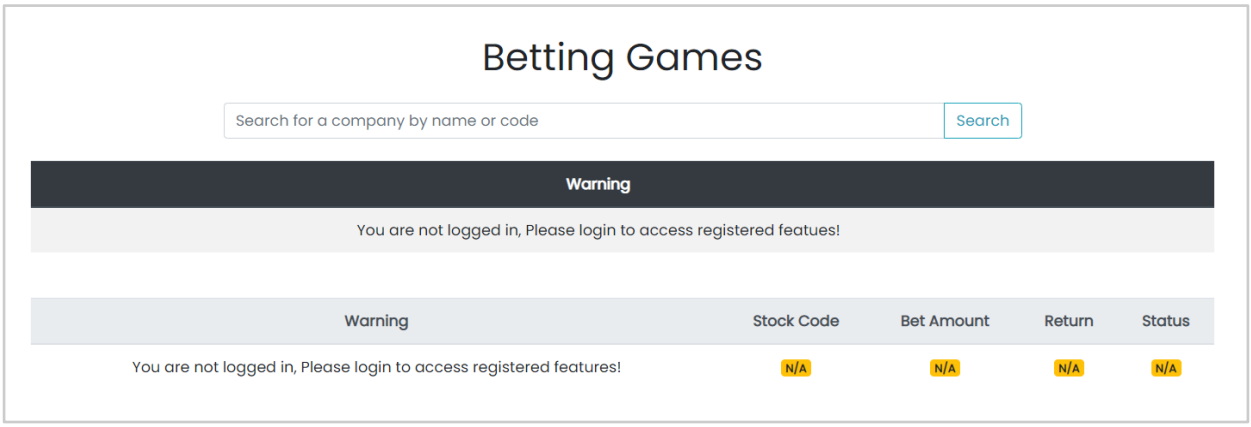
1	ADALTA LIMITED	1AD	Bet
2	AMAERO INTERNATIONAL LTD	3DA	Bet
3	POINTERRA LIMITED	3DP	Bet
4	3P LEARNING LIMITED..	3PL	Bet
5	4DMEDICAL LIMITED	4DX	Bet
6	9 SPOKES INTERNATIONAL LIMITED	9SP	Bet
7	AFRICAN GOLD LTD.	A1G	Bet
8	AIC MINES LIMITED	A1M	Bet
9	A2B AUSTRALIA LIMITED	A2B	Bet
10	THE A2 MILK COMPANY LIMITED	A2M	Bet

#	Stock Code	Bet Amount	Return	Status
1	CBA	$50000	+$0	active
2	ARX	$12000	+$48000	inactive
3	AUZ	$30000	-$120000	inactive

*Please note it takes a while for to show recently made bets

If the user tries to place a bet with an invalid amount of money or the loss of the bet exceeds the user's capital. The system does not process the bet and will display the specific error message to the user notifying them that their bet amount was invalid.

In the event that the user is not logged in, the system or interface will display a message to prompt the user to register or login to access the feature.



4. Third Party Functionalities

4.1. React

React is a JavaScript library for building frontend user-interfaces (Facebook, 2020). React is used throughout our code to construct components based on the route requested by the user, to provide a dynamic user experience without requiring full-page refreshes. React is provided on the permissive MIT license.

4.1.1. React-router

React-router is a npm library that contains a collection of navigational components and functions that compose declaratively with a web application. It is used to set up navigation between different pages in this React web application project. React-router is part of React and thus also provided under the MIT license.

4.1.2. React-router-dom

React-router-dom acts as a supporting library for react-router that exports DOM-aware components such as <BrowserRouter> and <Link> to use. Since react-router-dom re-exports all of react-router's exports, therefore we only need to import react-router-dom in our projects. React-router-dom is part of React and thus also provided under the MIT license.

4.2. Universal-Cookie

Universal cookie is a library that creates HTTP cookies to remember stateful information (Reactive Stack, 2020). A cookie is used to track whether a user is logged in or not, and if they are what their username is. It also gets posted in HTTP requests, is created when a user logs in and is deleted when a user logs out. Universal-cookie is provided on the permissive MIT license that allows for free use for non-commercial and commercial projects.

4.3. Node.js

NodeJS is a JavaScript runtime environment based on Chrome's v8 JavaScript runtime engine (Node.js Foundation, 2020). NodeJS uses non-blocking, event-driven I/O to remain lightweight and efficient, and serves to be the first point of contact for users connecting to our website. NodeJS is provided on the permissive MIT license that allows for free use for non-commercial and commercial projects.

4.4. Flask

Flask is a lightweight web microframework designed to make setup and routing quick and easy (Ronacher, 2020). Flask is used as a lightweight router to and from the JavaScript based frontend to the Python based backend. This allows quick deployment of the web application due to the team's familiarity with Python and Flask. Flask is provided on a permissive custom license similar to MIT where redistribution and use in source and binary forms, with or without modification, are permitted for free.

4.5. Bootstrap

Bootstrap is a popular framework for building responsive, mobile-friendly web application frontends (Otto et al., 2020). Bootstrap requires jQuery to function, and many of the frontend features utilises Bootstrap to format the page, e.g. the list of stocks displayed on the Market page is limited to 75% of the page width. Bootstrap is provided on the permissive MIT license.

4.6. React-bootstrap

React-Bootstrap is a replacement of the Bootstrap framework rebuilt specifically for use in React projects, and does not require jQuery to function (Schmitt, 2020). Due to the discovery of this framework late in the project, it was deemed not necessary to convert all existing code that relies on Bootstrap to instead of React-Bootstrap. React-Bootstrap was used to create modal popups on the page to support the buy/sell functionalities without requiring separate pages, improving the user experience. React-Bootstrap is licensed on the permissive MIT license.

4.7. jQuery

jQuery is a JavaScript library that simplifies HTML document manipulation, event handling and Ajax (Resig, 2020). jQuery is a requirement for Bootstrap and was used explicitly to support the Bootstrap hamburger menu on the navigation bar for mobile displays. jQuery is provided on the permissive MIT license that places almost no restrictions on what can be done with the software.

4.8. Canvas.js

CanvasJS is a HTML5 charting library that utilises a JavaScript API (Canvas.js, 2020). CanvasJS is necessary to provide the candlestick charts that provide a visual indication of stock performance in TradeHub. CanvasJS provides their code free for evaluation purposes for up to 30 days, after which it requires the purchase of a license before any project using CanvasJS can be deployed for commercial use. However, CanvasJS provides free licenses for projects deployed for non-commercial use, and as this project is not intended for commercial use, requesting a free non-commercial license will allow the continued use of CanvasJS in the future.

4.9. Bcrypt

Bcrypt is a popular password hashing function based on the Blowfish cipher (Python Cryptographic Authority, 2020). Password hashing is necessary to provide a layer of security to our database, by only storing the hashed password and its accompanying salt in the database, rather than storing the passwords directly encrypted or in plaintext. This means the release of our database will not compromise the security of our user’s accounts. The version used is provided by the Python Cryptographic Authority, and the code is provided on the permissive Apache 2.0 License.

4.10. SQLite

SQLite is a lightweight, fast, self-contained public domain relational database engine that allows storing user-specific information (Hwaci, 2020). The following functionalities must be tracked against user accounts, thus require interaction with the database:

- Account creation/registration + Account deletion
- Logging In
- Password resetting
- Viewing account details/portfolio/watchlist
- Stock trading
- Investment advice
- Betting games

The core SQLite database software is in the public domain, thus no licensing is required to use SQLite as the relational database management system for TradeHub.

4.11. yfinance

In order to retrieve information on stocks listed on the ASX, yfinance is used. yfinance is a third-party API that provides a Pythonic way of downloading historical market data from Yahoo! Finance (Aroussi, 2019). This allows users to view current and historical information of stock performance, and trade based on the current market price. yfinance is released on the permissive Apache 2.0 License which allows for free use. yfinance also provides a level of abstraction that will allow easier deployment of the web application by placing less focus on learning an API.

5. Implementation Challenges

5.1. Gateway between Frontend and Backend

In the beginning, the main challenge is setting up connections between frontend and backend. The option was to pick between Flask or Django as our main web framework. We eventually chose Flask as our web framework due to all group members having relevant experience with Flask. Additionally, due to the time constraint imposed the group needed something that is easier to learn and work. While Django is a much more preferable web framework, the learning curve was steep therefore it was not picked.

5.2. Frontend Inexperience

Most group members have not experienced React even javascript programming. Before our group started on coding, all of us followed the react tutorial and understood basic concepts in React while another more experienced group member set our basic architecture of the website. Our group learns from necessary functionalities including updating of state, fetching data, and rendering on the page. Next, we organize every element on the page using bootstrap to format our page for the purpose of improving the user experience.

5.3. Preprocessing Company Directory

Our group downloaded the original company list from ASX (Australian Securities Exchange,2020). However, Yahoo Finance (yfinance) API is not able to search some company codes due to the company being bankrupt or changing the name. Another dilemma is yfinance API considers company code as valid even company code without any historical data. In order to prevent above situations, we wrote a filter program that filters out invalid company code and company code without data.

To begin with, try and except statements were used to filter out the delisted companies. Next, we called the yfinance API to retrieve the company's latest unit price and historical data for the last five working days. The filter function removed any companies that have zero volumes which indicates that no investor has been trading the company's stock for at least the past five working days or if there is no valid historical data for the past five days. The resulting company list or directory after filtering is our company list in which all the companies are still in market and has valid historical data.

5.4. Handling Raw Data from API

Our website calls from yfinance to get original data of company code. One of the main challenges is preprocessing the retrieved raw data. In order to avoid calculating cumbersome trading days and non trading days, we handled by fetching data over a longer period of time and formatting it. If any daily data is missing for the last five days, we push the day ahead so it makes sure there will be five latest trading days' data presented in stock history.

5.5. Testing Betting System

There was a challenge in testing that a bet finishes properly and that it is reflected correctly in the frontend. Each bet in the betting system lasts for a week and it was impractical to wait a week for testing purposes.

We successfully tested the backend functionality through unit tests, however we needed to test that the front end changes state appropriately when a bet finishes. We would start the workflow for making a bet through the frontend, and once we've made a bet we would force it to close by directly manipulating the 'end_date' value in the 'bet_history' table and the 'start_date' in the 'betting' table.

This was done by running a small python script that performs UPDATE statements directly on the SQLite database to change the end date to to 2 days previous, and the start date to 9 days previous. The frontend sends a request every 10 second intervals to keep the page updated. This allowed us to check for any bugs.

6. User documentation/Manual

6.1. First-time Setup and Run

1. Open a terminal and clone the directory using git, or unpack the zip file:

```
$> git clone git@github.com:unsw-cse-capstone-project/capstone-project-comp3900-h11a-undefined.git h11a-undefined
```

2. Navigate to the new folder via the terminal, or open a terminal in the new folder:

```
$> cd h11a-undefined
```

3. Execute setup.sh. Please note that this process may take a few minutes to download and install the requirements, and fully initialise the web application.

```
$> ./setup.sh
```

Setup.sh is only needed for first time deployment of the web application, afterwards you only need to execute run.sh:

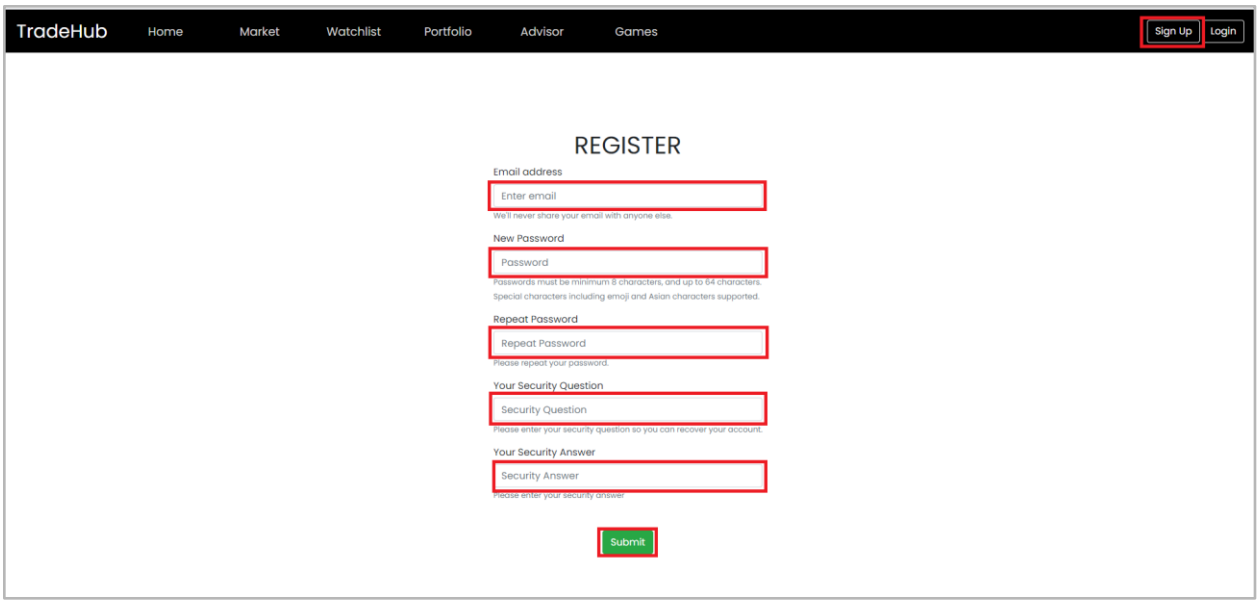
```
$> ./run.sh
```

The default web browser will automatically open and navigate to the deployed web application. If this does not occur, navigate to address localhost:3000. Ensure port 5000 is not blocked as the flask server will run on this port.

4. To close the web application, simply type CTRL-C in the terminal.

6.2. Create Account

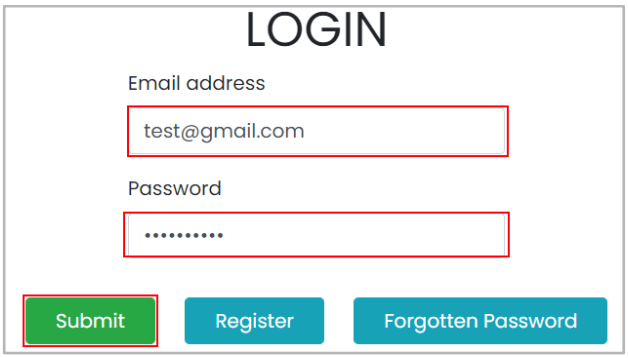
1. If you want to create your personal account,you need to go to the registration page by clicking the signup button at the navigation bar. Credentials need to meet following criterias:
    - A valid email address
    - The length of password between eight to sixty-four
    - New password and repeated password are identical.
    - Any fields can’t be void. After entering credentials, you could click the “submit” button. If any of criteria haven’t met, the corresponding error message will be displayed. You can change the details based on message(s).



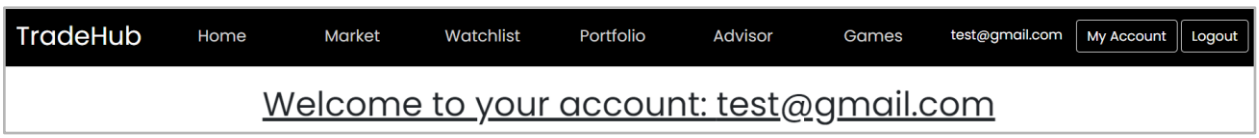
6.3. Logging In/Out

1. You log into your account by first clicking on ‘Login’ on the navigation bar.

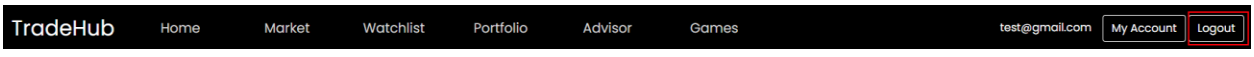
2. On the login page, enter your credentials then click ‘Submit’.



3. Successfully logging in will take you to your account homepage. Unsuccessful attempts will lead to the relevant error message being displayed on the page.



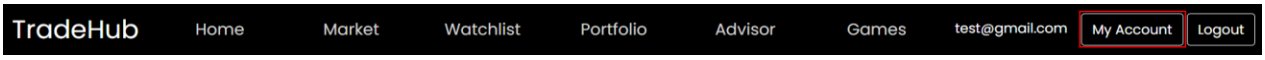
4. You can logout at any time via the navbar.



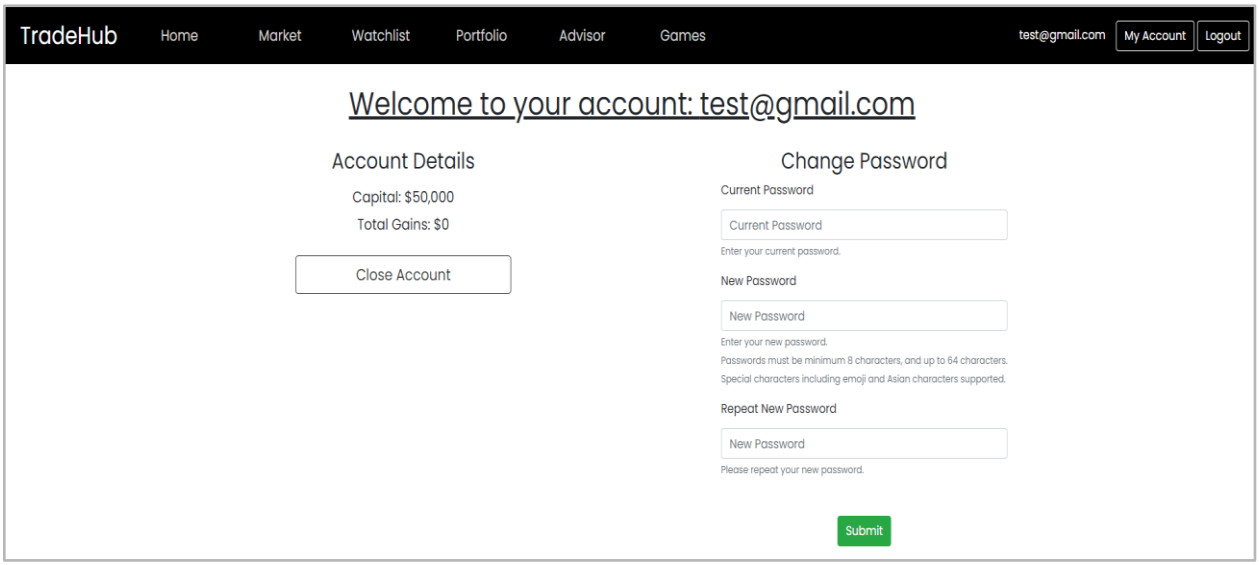
6.4. Account Details

1. To view your account details, you must first be logged in. This is described in section 6.3.

2. Once logged in the navbar changes and the ‘My Account’ button will appear. Click on this button to navigate to your account page



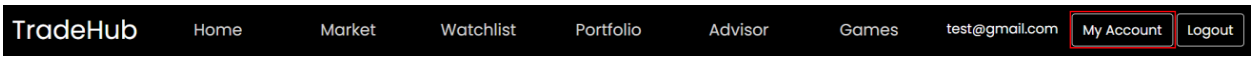
3. The account page will be displayed. It contains relevant details as well as close account functionality and change password functionality.



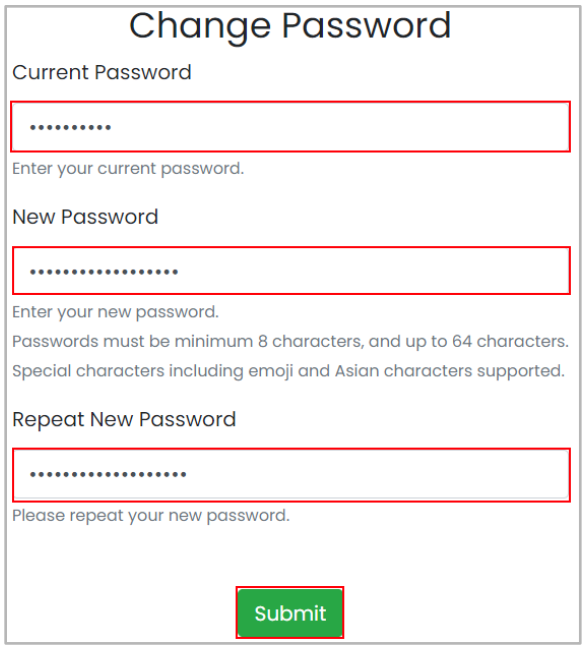
6.5. Password Reset

6.5.1. Logged In

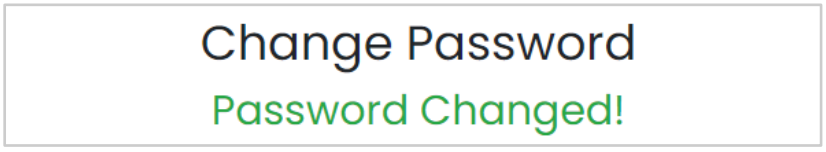
1. Navigate to your account homepage by clicking ‘My Account’



2. On the right side of the screen you can change your password by entering your current password, your new password, and a repeat of your new password. Once you have entered these details, click ‘Submit’



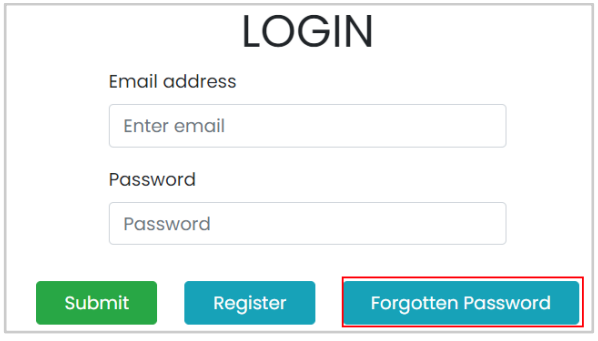
3. After submitting, messages will appear to notify of a successful or unsuccessful password change. If unsuccessful, the relevant error messages will be displayed to tell you why it failed.



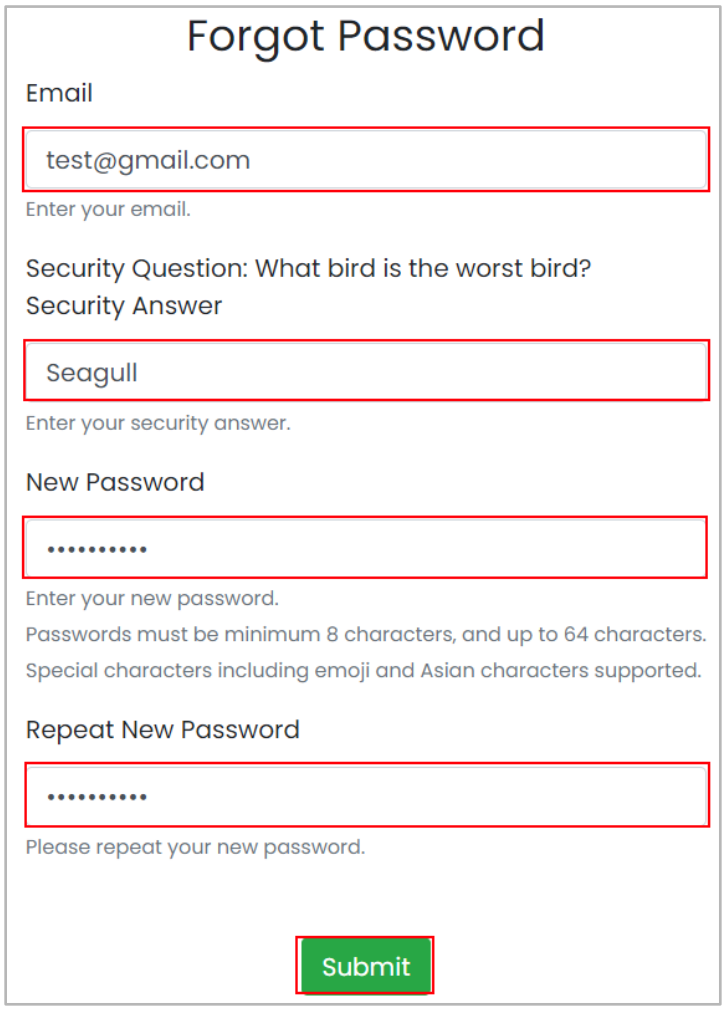
6.5.2. Forgotten Password

1. If you have forgotten your password, you can recover your account by answering the security question you created when you registered. Click on ‘Login’ in the navigation bar

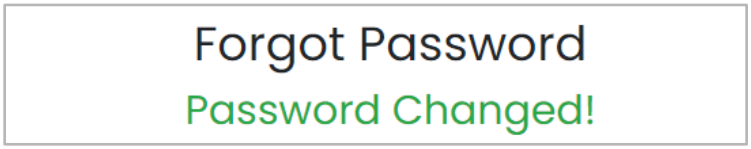
2. Click on ‘Forgotten password’



3. Enter your email, security answer, your new password and a repeat of your new password, then click on ‘Submit’

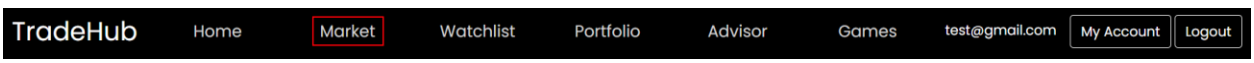


4. On a successful password change a confirmation message will appear. If unsuccessful, the relevant error messages will be displayed.

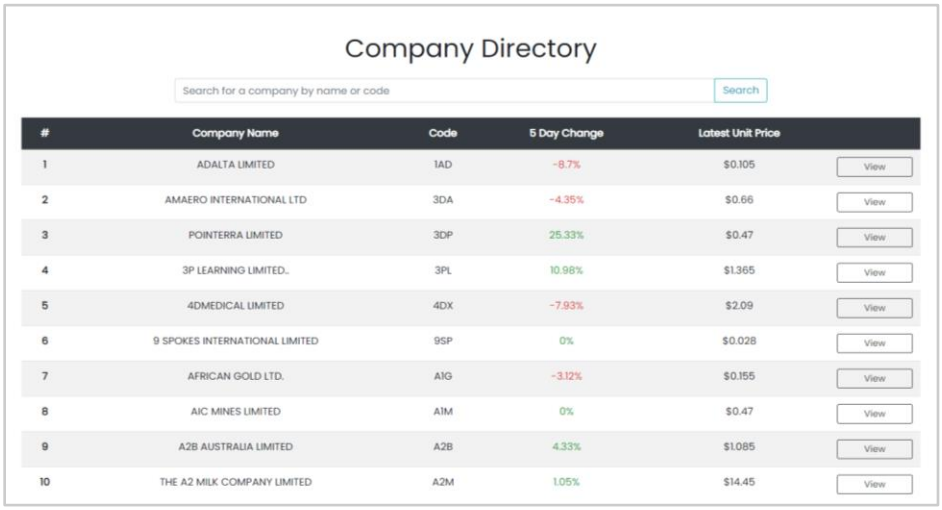


6.6. Search Stocks

1. To search for stocks, navigate to the ‘Market’ page via the navbar. You do not need to be logged in for this function.

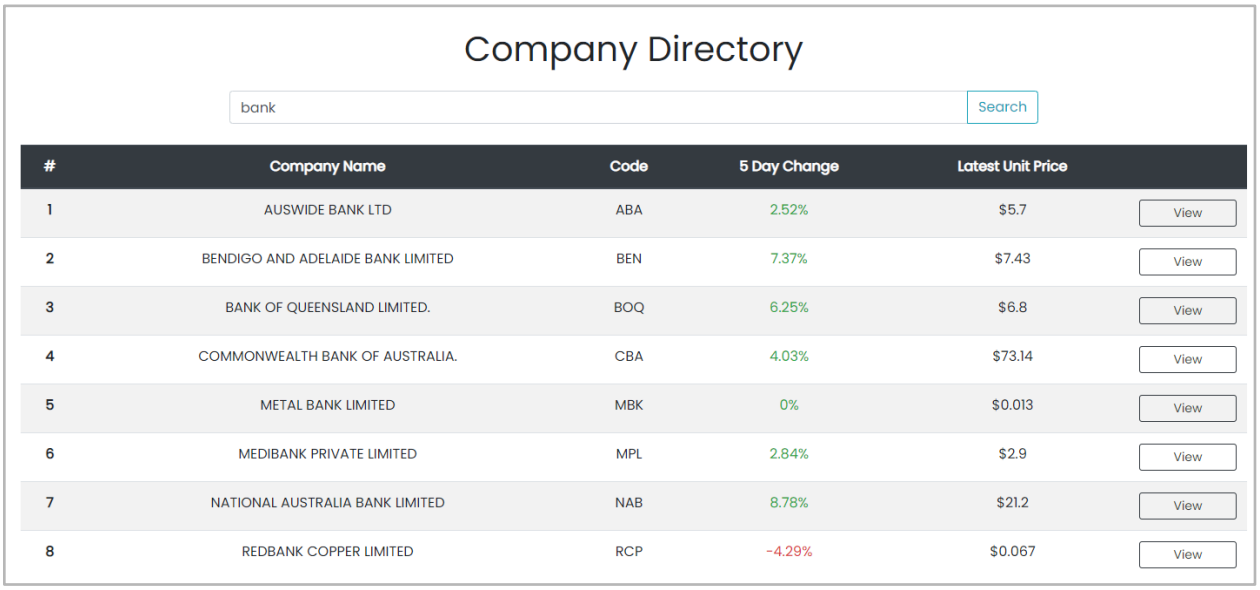


2. You will be redirected to the company directory.



3. You can search for stocks based on a company's name, or their associated stock code. Type in your search query and click the 'search' button.

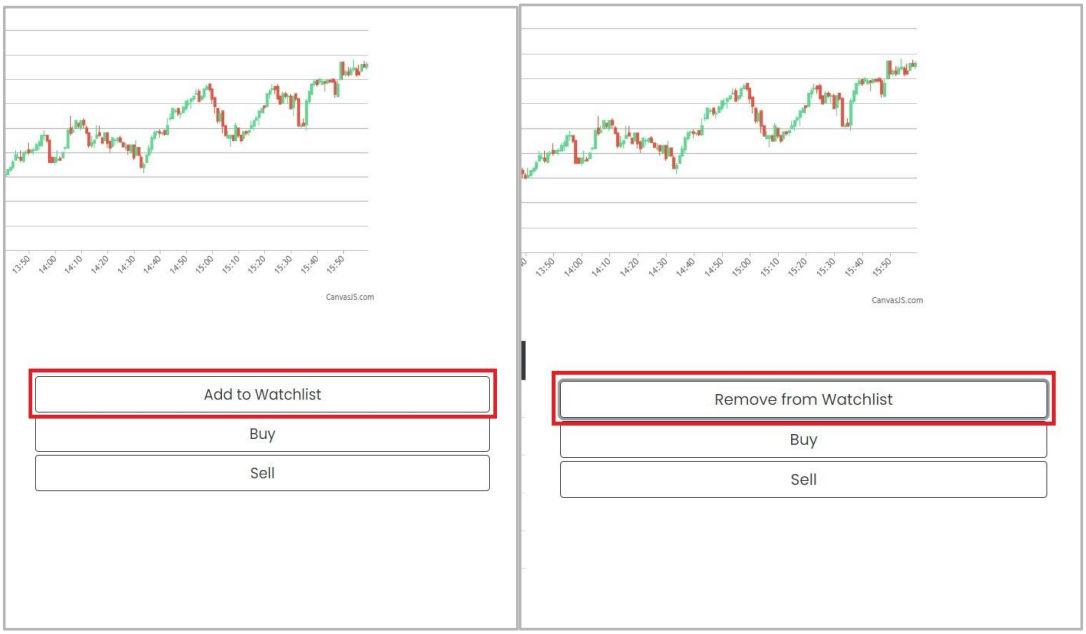
4. Your search results will be displayed. The results are sorted alphabetically, based on stock code. You can also see the latest stock closing price, and the stock percentage change from 5 days previous



5. You can then navigate to a specific stock's details page by clicking on 'view'

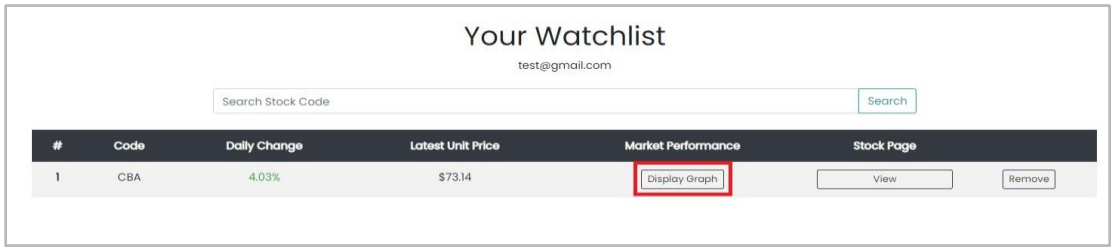
6.7. Watchlist

1. In order to add stock to your watchlist, you need to go to the stock detail page and click the ‘Add to Watchlist’ button. If you already have this stock in your watchlist, you can remove it by clicking the ‘Remove from Watchlist’ button.

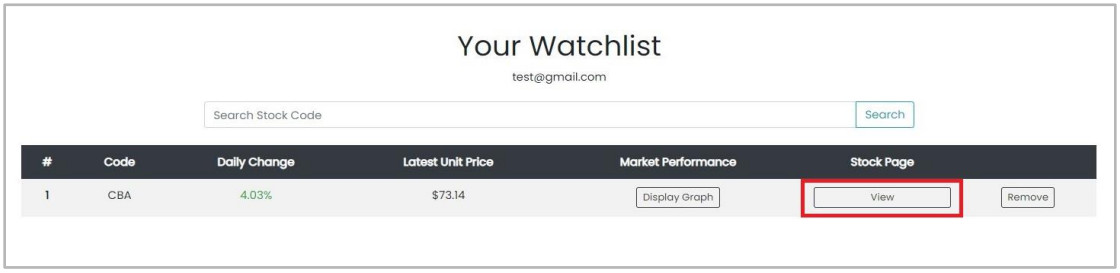


2. You can access the watchlist by clicking the Watchlist button on the navigation bar.

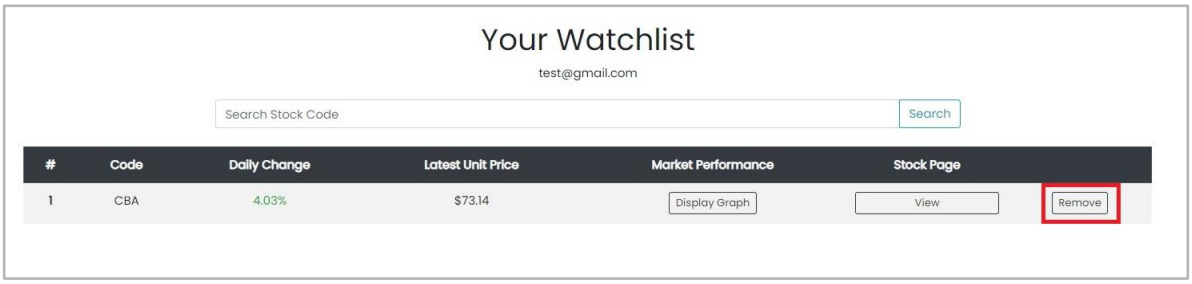
3. You can click the ‘Display Graph’ button and view the historical graph.



4. You can click the ‘View’ button and redirect to the detail page of stock.

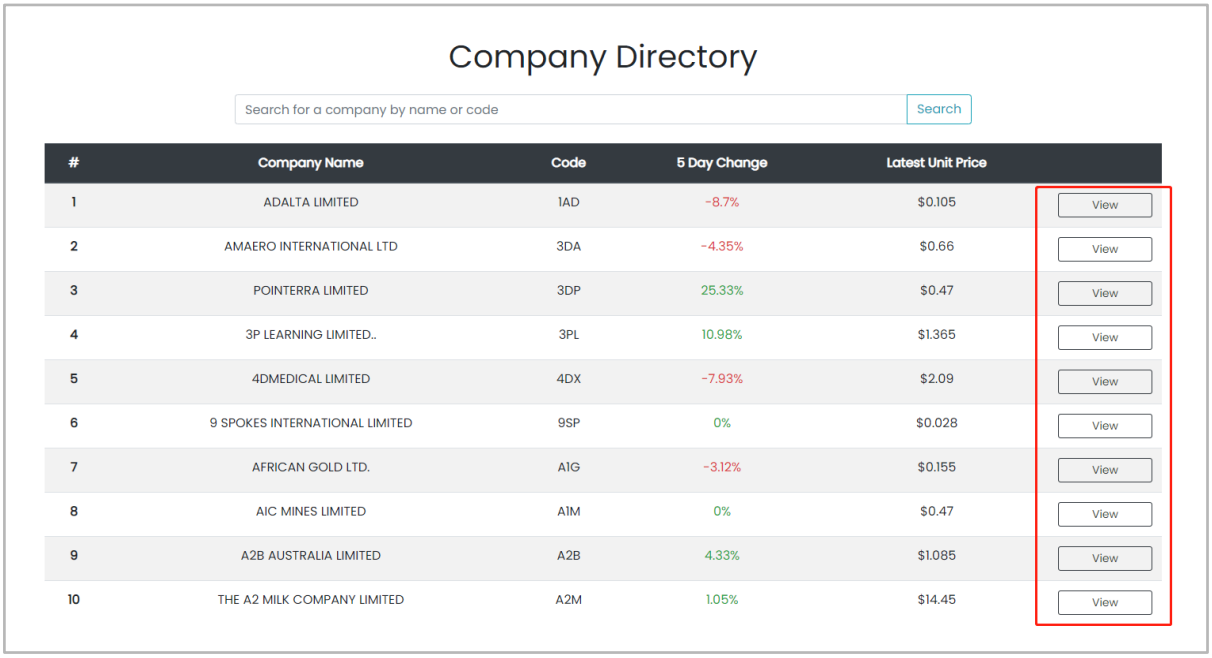


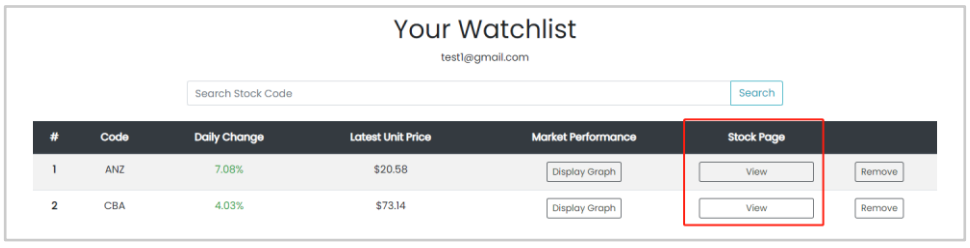
5. You can remove the stock by clicking the ‘Remove’ button.



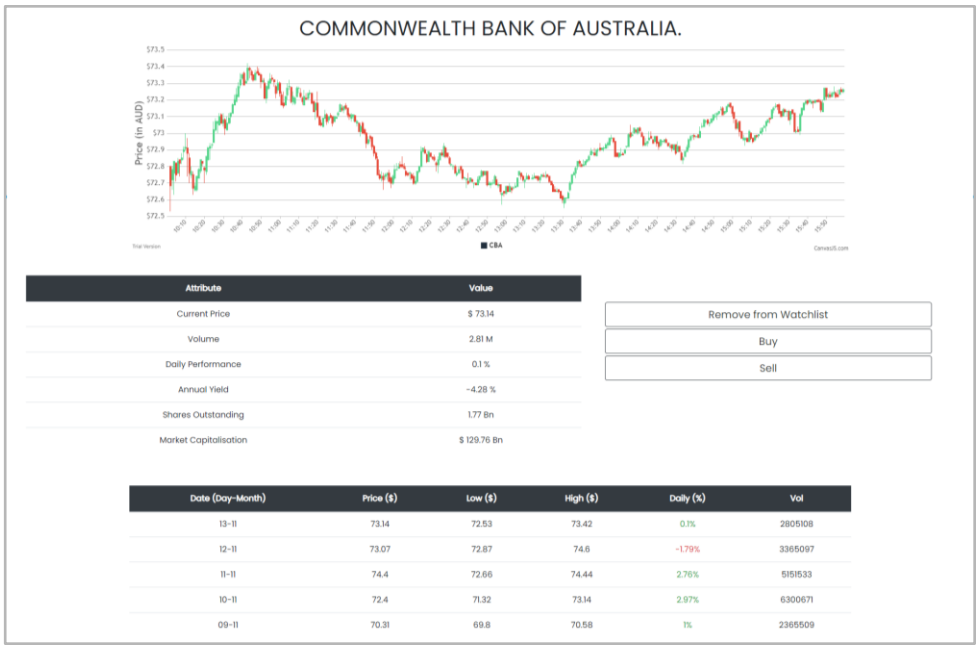
6.8. Stock Details Page

1. By clicking the view button on the market search page or the user’s watchlist, you can go to the stock details page.





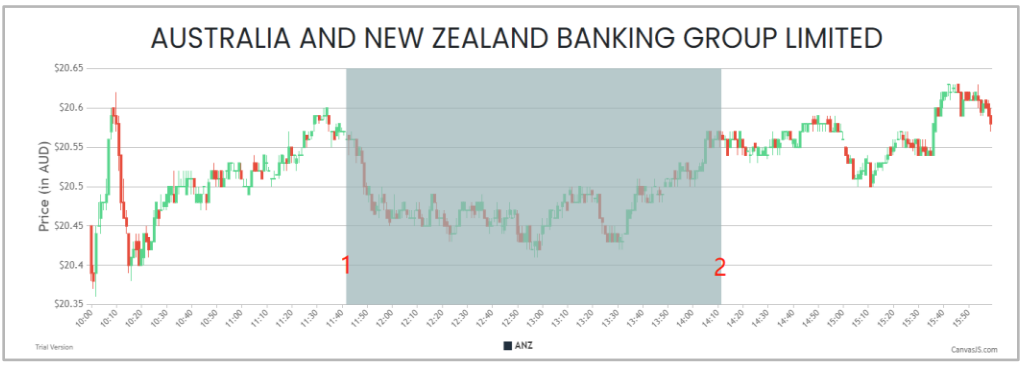
2. There are two different tables, the upper one displays Volume, Daily Performance, Annual Yield, Shares Outstanding and Market Capitalisation for the latest trading day. The bottom one displays closing price, highest price, lowest price, daily change and volume for the latest five trading days.



6.9. Graph Navigation

1. There is a candlestick chart in the stock details page displaying an overview for the latest trading day. If it's trading time, it will update by minute automatically. By putting the mouse on to different bars, you can see specific information for each minute.

2. You can zoom by clicking at the start time and dragging to the end time and release



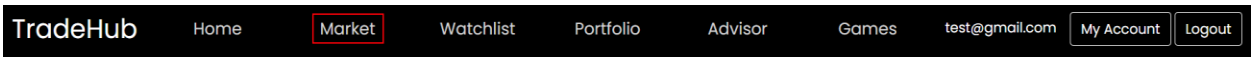
3. After zooming in, two helper buttons will show up. button 2 is the reset button, you can go back to overview by clicking button 2. By clicking the button 1, the chart will change to pan mode.

4. In pan mode, the cross arrow becomes a magnifier symbol. You are able to drag the chart horizontally by the left button of the mouse. To go back to zoom-in mode, you can click on the magnifier symbol.

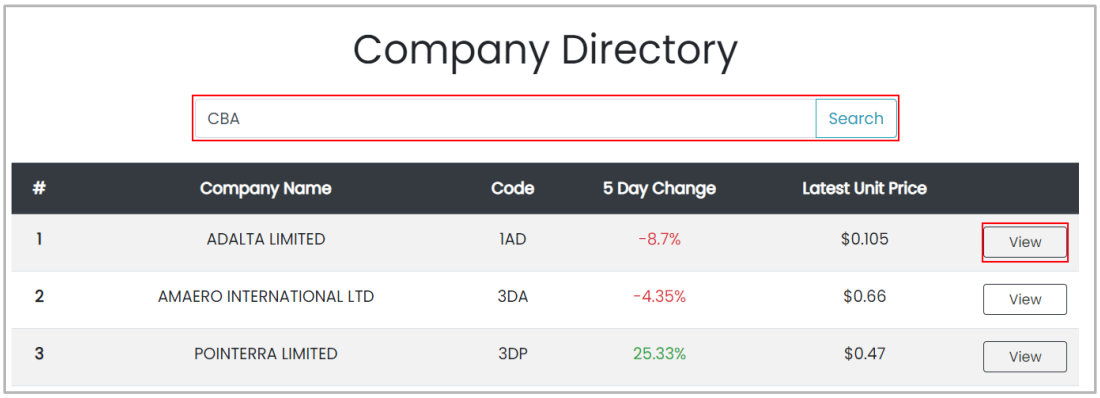
6.10. Stock Trade

1. To trade stocks, you must first be logged into your account

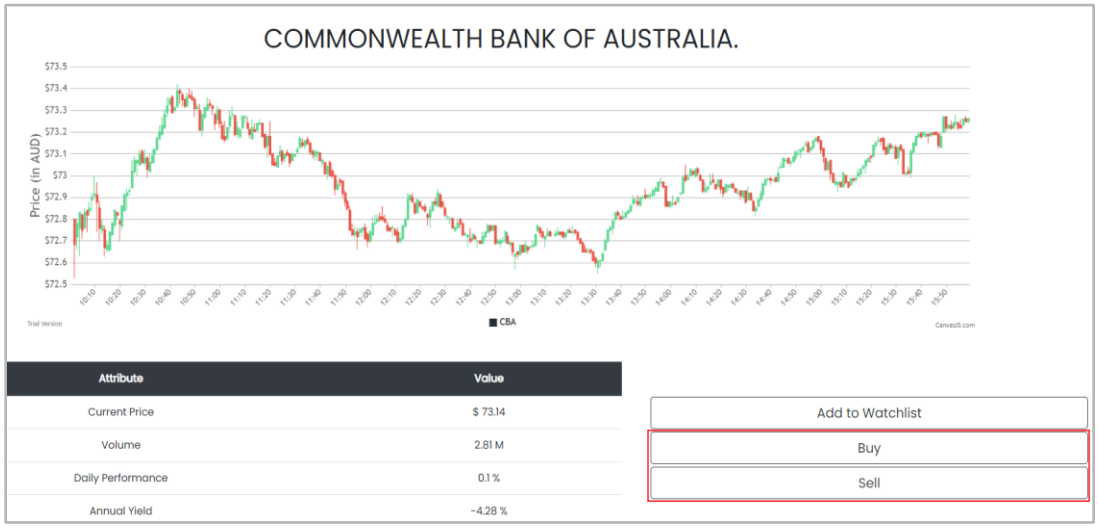
2. Once logged in, navigate to the market



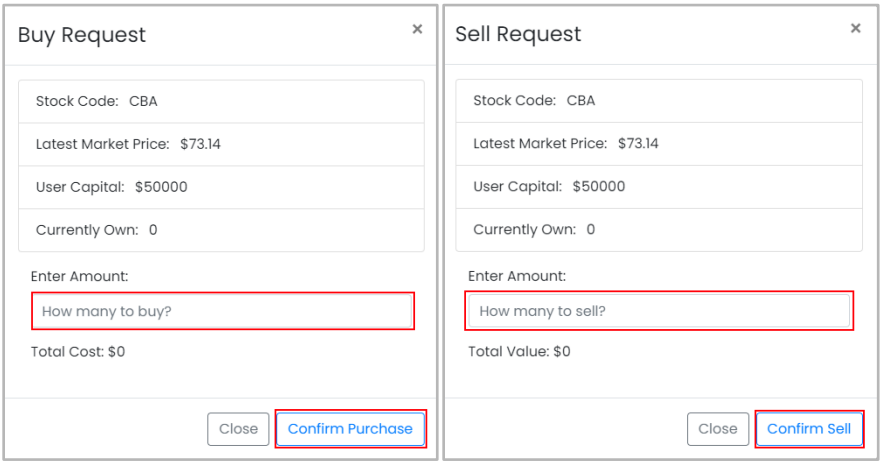
3. A company directory will appear, along with a search function. Search the stock you wish to trade if it is not displayed in the list, then click ‘View’ of the stock you wish to trade.



4. On the stock’s details page you can trade with the ‘Buy’ or ‘Sell’ buttons



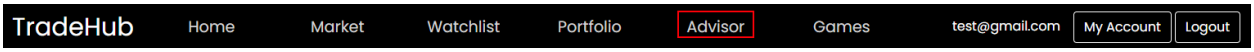
5. Clicking either of these will bring up a pop-up will appear where you can choose how many stocks to buy or sell. Click ‘Confirm’ to confirm your trade.



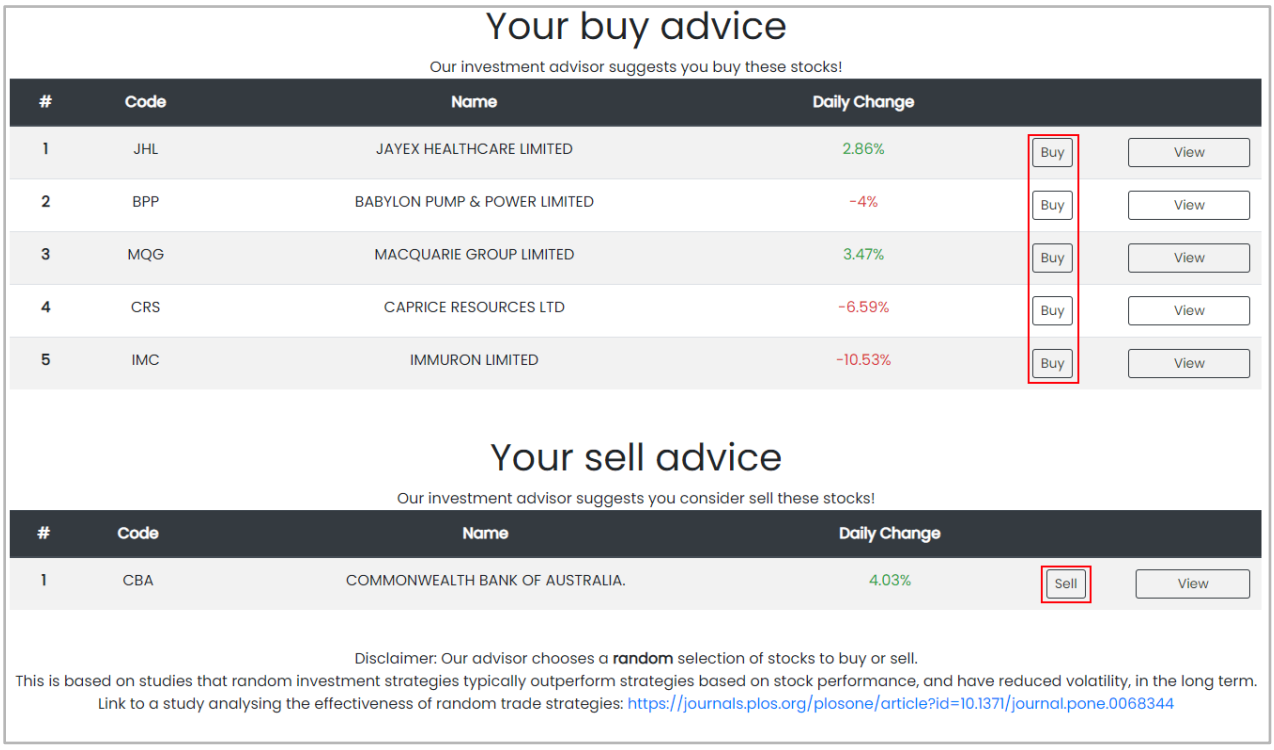
6.11. Investment Advisor

1. The investment advisor provides a selected list of stocks that you can trade, however to access the full features you must first login.

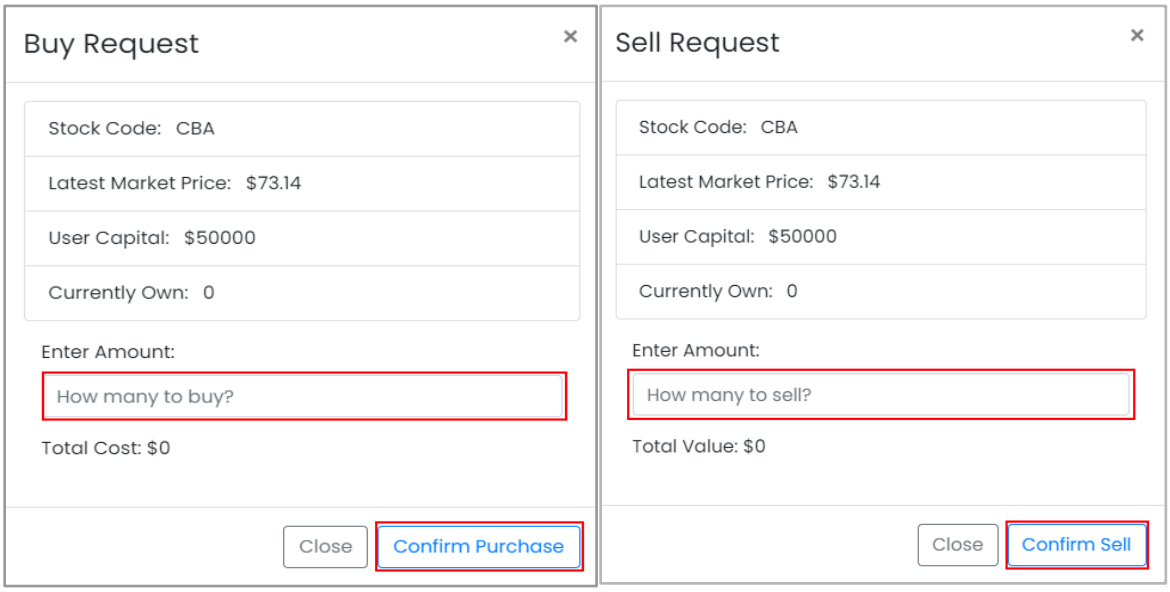
2. Navigate to the ‘Advisor’ page via the navigation bar



3. On the advisor page you will be presented 5 stocks to buy, and possibly some stocks that you currently hold that could be sold. Click on the ‘Buy’ or ‘Sell’ buttons to trade stocks without leaving the page, or click on ‘View’ to navigate to the stock’s details page.

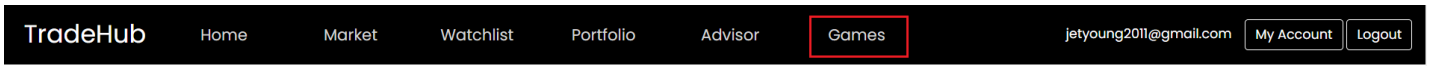


4. Clicking ‘Buy’ or ‘Sell’ will bring up the same interface for trading stocks as doing so in a stock’s details page.

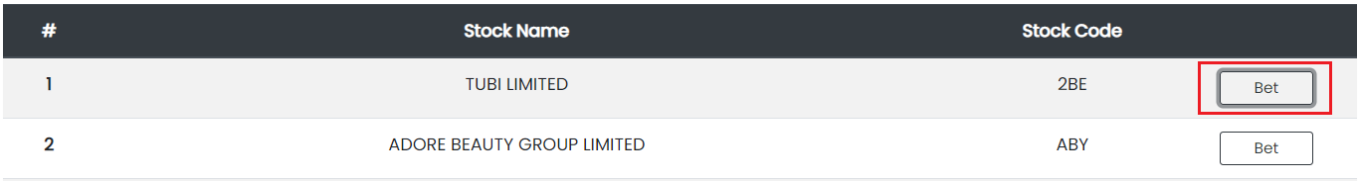


6.12. Betting System Interface

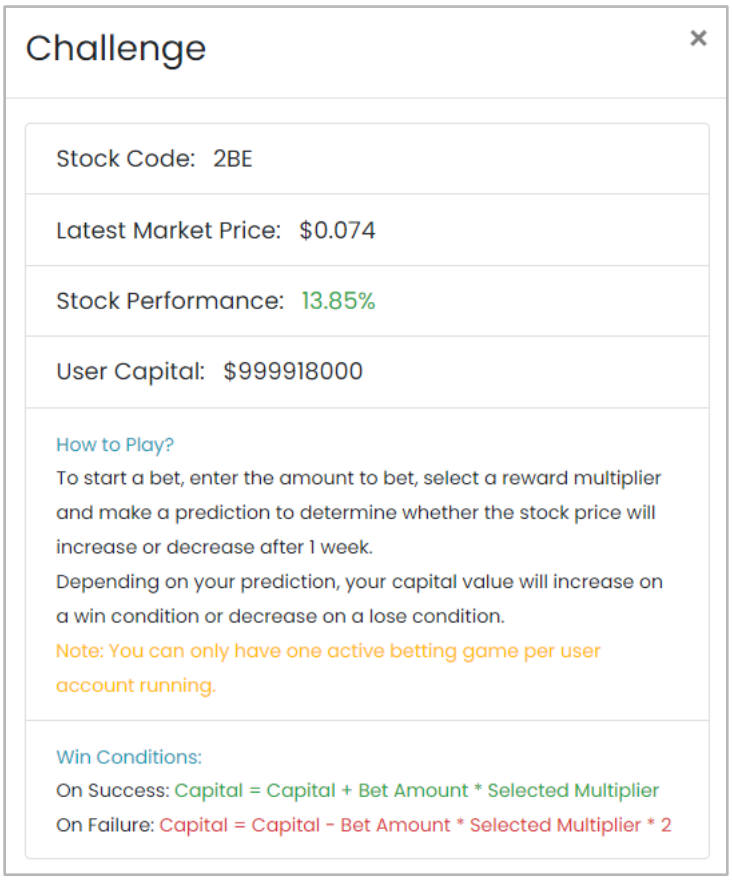
1. To access the betting game, the user must click the “Game” hyperlink in the navigation bar at the top of the page.



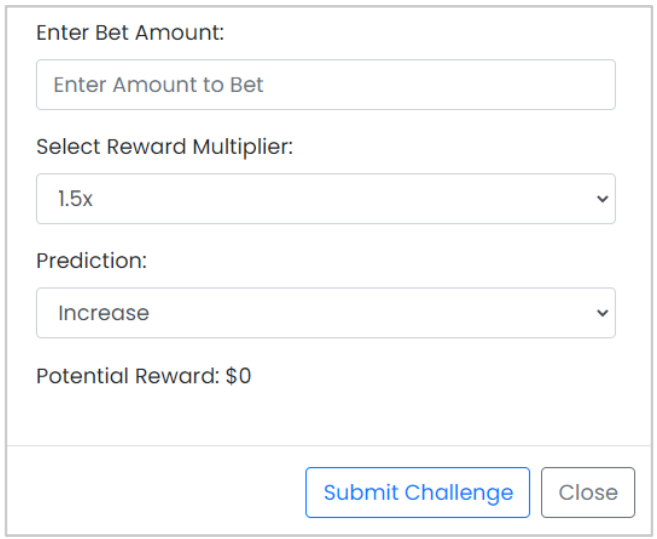
2. Once the user is at the game or betting page, the user can place a bet by clicking on the bet button for a specific stock that is located at the end of each row.



3. Upon clicking the bet button, a popup will appear with the stock details, your capital, and an introduction on the betting system.



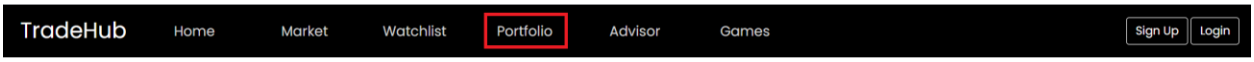
4. You can customise your bet's bet amount, their prediction, and the reward multiplier they desire.



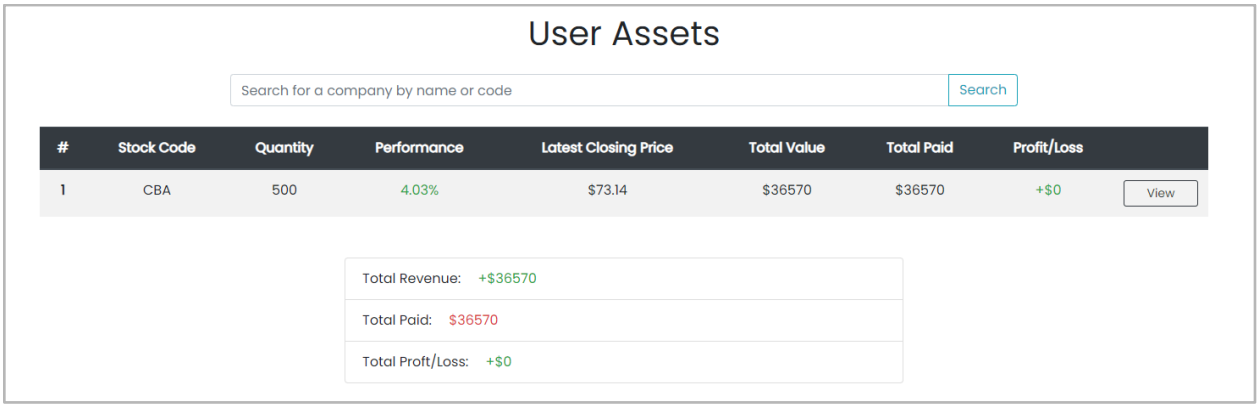
5. Once the user finishes customising their bet, they can click on the “Submit Challenge” button to place their bet. Optionally, if the user decides they do not want to bet they can click on the “Close” button to cancel the bet.
6. The user is also able to search for stocks associated with a given search word or stock code. The search functionality is identical to searching for stocks on the Market page.

6.13. User Portfolio Interface

1. To access the portfolio page, the user must click the “Portfolio” hyperlink located in the navigation bar at the top of the page.



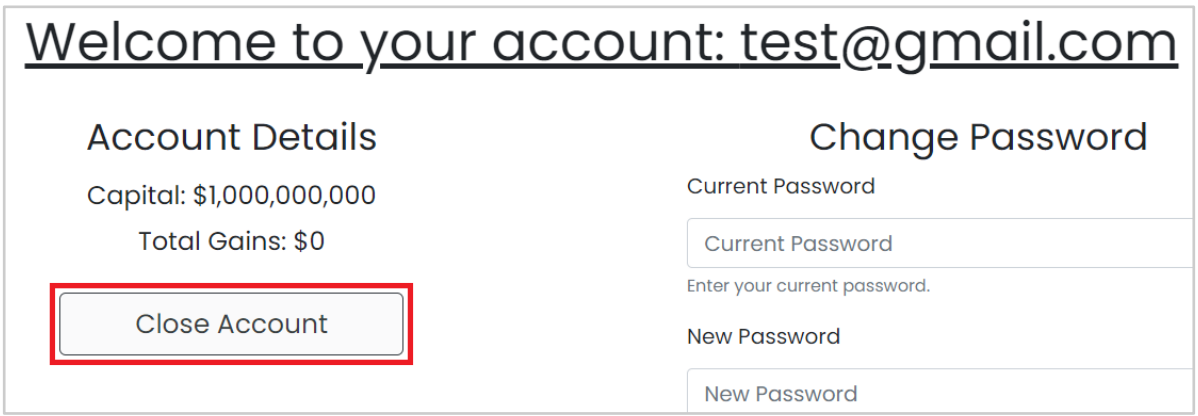
2. The user can now see the new stock they purchased along with other useful information for investing in the table.



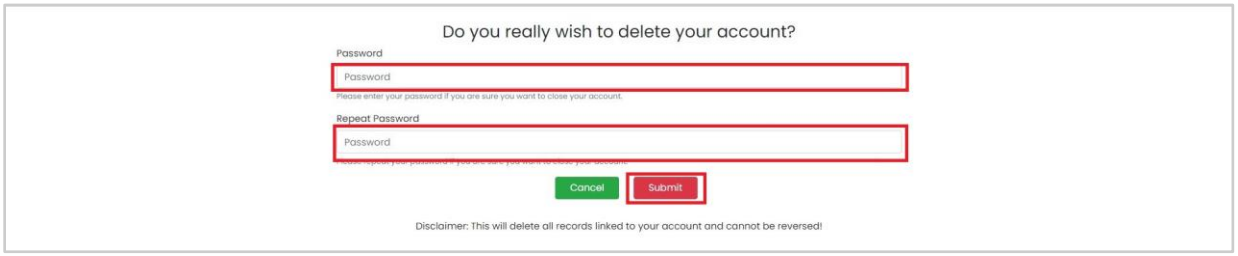
3. Users can enter search words into the search bar to shorten and find specific stocks they own associated with the search word. The search functionality is identical to searching for stocks on the Market page.

6.14. Delete Account

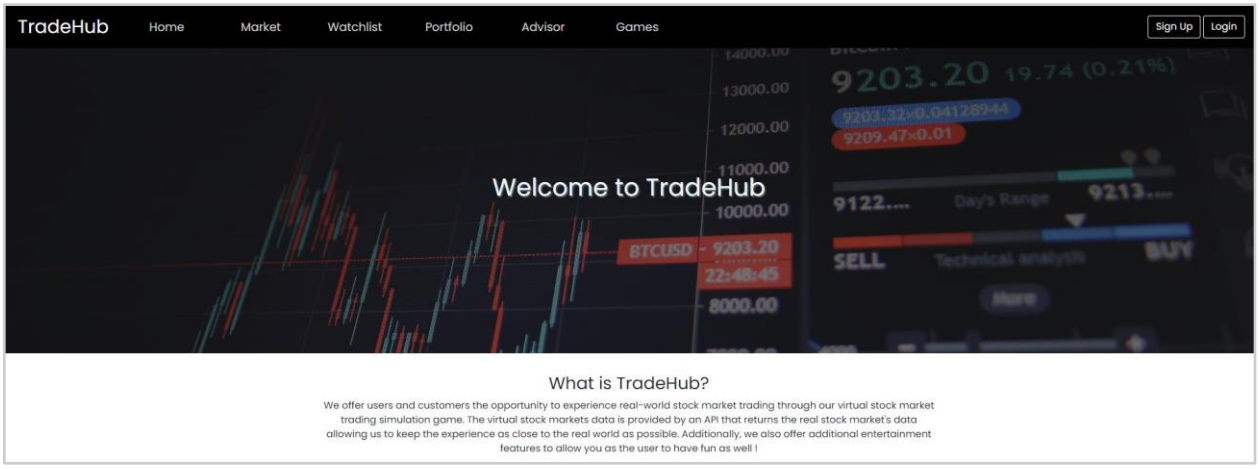
1. If you want to close your account,you need to go to your account page and click the close account button.



2. Then you are redirected to the Close Account Page. If you don't want to close your account, you could click the cancel button and redirect to your account page. If you still want to close your account, then you need to input the correct password twice and then click the Submit button. If unsuccessful, the respective error messages will appear to indicate what went wrong.



3. You will be redirected to the main page after five seconds without login.



7. References

1. Apache.org. 2020. *Apache License, Version 2.0*. [online] Available at: https://www.apache.org/licenses/LICENSE-2.0 [Accessed 16 November 2020].

2. Aroussi, R. (2019). yfinance (0.1.55) [software] (16 Nov 2020) Available at: https://github.com/ranaroussi/yfinance

3. Australian Securities Exchange. 2020. *Company Directory*. [online] Available at: https://www2.asx.com.au/markets/trade-our-cash-market/directory [Accessed 16 November 2020].

4. Biondo, A., Pluchino, A., Rapisarda, A. and Helbing, D., 2013. Are Random Trading Strategies More Successful than Technical Ones?. *PLoS ONE*, 8(7), p.e68344.

5. CanvasJS. (2020). CanvasJS StockChart (1.2.2) [software] (16 Nov 2020) Available at: https://canvasjs.com/download-html5-charting-graphing-library/

6. CanvasJS. 2013. *Canvasjs Licenses | Canvasjs*. [online] Available at: https://canvasjs.com/license/ [Accessed 16 November 2020].

7. Facebook (2020). React (16.13.1) [software] (16 Nov 2020) Available at: https://reactjs.org/

8. Hwaci. (2020). SQLite (3.33.0) [software] (16 Nov 2020) Available at: https://www.sqlite.org/index.html

9. Node.js Foundation (2020). Node.js.(14.15.0) [software] (16 Nov 2020) Available at: https://nodejs.org/en/

10. Opensource.org. 1984. *The MIT License | Open Source Initiative*. [online] Available at: https://opensource.org/licenses/MIT [Accessed 16 November 2020].

11. Otto M, Thornton J. (2020). bootstrap (4.5.2) [software] (16 Nov 2020) Available at: https://getbootstrap.com/

12. Python Cryptographic Authority. (2020). Bcrypt. (3.2.0) [software] (16 Nov 2020) Available at: https://github.com/pyca/bcrypt/

13. Reactive Stack.(2020). Universal-cookie. (4.0.4) [software] (16 Nov 2020) Available at: https://github.com/reactivestack/cookies/tree/master/packages/universal-cookie

14. Resig, J. (2020). jquery (3.5.1) [software] (16 Nov 2020) Available at: https://jquery.com/

15. Ronacher, A. (2020). Flask (1.1.2) [software] (16 Nov 2020) Available at: https://flask.palletsprojects.com/en/1.1.x/

16. Schmitt, M et al. (2020). react-bootstrap (1.4.0) [software] (16 Nov 2020) Available at: https://react-bootstrap.github.io/