# Development of an Investment Research Dashboard with Machine Learning Integrations

**Cameron Morrison**

2831609

Dr Jason Adair

*April 2023*

**Dissertation submitted in partial fulfilment for the degree of**
**Bachelor of Science with Honours Software Development with Cyber Security**

Computing Science and Mathematics

University of Stirling

# Abstract

Investment research platforms are complex, often inaccessible, and sometimes hidden behind paywalls requiring investors to pay for access. They are designed to deliver excessive information on a single webpage. These factors result in new, dyslexic, and visually impaired investors potentially missing the free exchange of market knowledge. Even for experienced investors, the current investment research platforms have unintentionally been designed to negatively impact investor sentiment due to an overuse of fearmongering headlines, the colour red, and forums containing impulsive and misinformed statements from users.

The project aimed to design a fast, dynamic, and robust enterprise-level web application with the hopes of delivering the end user, regardless of age, experience, or net worth, a cost-free alternative and more inclusive investment research platform. This was achieved using modern technologies, tools, and User Experience (UX) design best practices. Sentiment analysis interprets headlines to generate a sentiment classification for the daily news articles, allowing the news feed to display positive headlines first. Time series forecasting dynamically predicts a particular stock's price trend, which could aid users in their research. The application provided a platform for usability testing with the addition of Artificial Intelligence (AI) to discover whether this addition could benefit amateur investors in their research and help them gain a healthier long-term mindset towards the markets.

The project's aim and objectives outlined in section 1.3 were all achieved. The application was made with industry-recognised languages and tooling. The application's performance, scalability, security, robustness, and customizability were demonstrated throughout. During usability testing, 80% of participants stated the proposed application was more accessible than its two compared competitors, Yahoo Finance and Google Finance. In addition, participants found sentiment analysis helpful, as they could see a benefit to new investors unaware of financial terminology.

Future work is recommended for time series forecasting, as participants were nervous about the perceived inaccuracies in market forecasts. However, the proof-of-concept integration is dynamic and performant. Furthermore, the current setup serves as a foundation to expand and use more complex models, as Microsoft announced their intentions to add native support for deep learning methods [37].

# Attestation

I understand the nature of plagiarism, and I am aware of the University's academic integrity policy.

I certify that this dissertation reports original work by me during my university project except for the following:

The market data responsible for the operation of server-side application identified in section 3 is dynamically provided by [68], [69] and [70].

The sentiment analysis model identified in section 3.2.7 was trained on a dataset taken from Kaggle [55].

A dataset used for the stock search bar functionality responsible for mapping ticker symbols to descriptive company names identified in section 3.4.2 was obtained from Kaggle [71].

I certify that the figures are my own, excluding figures 3, 8, 9, 13, 16, 17 & 25. References to the initial sources are included within each figure.

**Signature** *Cameron Morrison* **Date** 16/04/2023

## Acknowledgements

I want to express my gratitude to Dr Jason Adair for his patience, constructive criticism, and continued support throughout the project.

I would additionally like to thank the individuals that participated in usability testing.

# Table of Contents

# List of Figures

# List of tables

# 1 Introduction

This project aimed to design and implement a client-facing dashboard for investment research with conscious performance, proof-of-concept sentiment analysis and time series forecasting integrations. At its core, the project was to deliver a fast, dynamic, and robust enterprise-level progressive web application. This project aimed to provide the end user, regardless of age, experience, or net worth, with a cost-free alternative and more accessible web application. The application used modern technologies, tools, and User Experience (UX) design best practices. A sentiment analysis integration interprets headlines to generate a sentiment classification (positive or negative) for each daily news article. This results in quicker insight into the tone of the markets and delivers a more flexible user experience. Time series forecasting dynamically predicts a particular stock's price trend, which could aid users in their research. The application provided a platform for usability testing with the addition of Artificial Intelligence (AI) to discover whether this could benefit amateur investors in their research and help them gain a healthier long-term mindset towards the markets.

## 1.1 Background and Context

Investing in companies can benefit investors as it grants them more financial independence and, over time, can yield impressive returns. In periods of economic downturn where investing looks less appealing, less experienced investors begin to doubt their holdings and flee the markets as a reaction to news [1, 2]. The markets are a roller coaster of emotion as negative sentiment spreads from news articles to independent forums. With losses not experienced before by new investors, the unique experience could be frightening [10]. Fearmongering headlines and misinformation [1, 3] [Fig. 2], do not help investors make informed and calm decisions during market oscillations.

Often popular investment research and stock advisor websites offer guidance only accessible to investors who can afford costly subscriptions. The websites that do not require payments often lack information or are overcrowded with small-font technical terminology, which deters users from pursuing an interest in managing their money. The news articles tied to these websites can be time-consuming to analyse, contradictory and fearmongering [Fig. 2]. For this project, the application could not guarantee the articles offered are reliable when the content is provided from an external source. However, as the sentiment analysis classifies the article by summarising the headlines, the articles are sorted by positive first, which subtly dissuades fearmongering headlines from being displayed to the user.

Investors are more enticed to liquidate holdings or purchase expensive subscriptions to research websites due to the worry instilled by negative headlines [4]. According to a study published by Oxford University [5], there is a considerable correlation between stock price fluctuations, mental disorders, and hospitalizations. The study was conducted in Taiwan over four thousand days from 1998 to 2009 and was the most extensive known population study of its kind. The conclusion was the identification of a noticeable correlation between mental illness, hospitalizations and a daily decline or repeated price declines.

## 1.2 Problem analysis

Often, investment research platforms are hidden behind paywalls, requiring investors to pay for access and are often designed to deliver an excessive quantity of information on a single webpage [Fig. 3]. This project aimed to design and implement a client-facing web application to research stock investments, completely free and more accessible to a broader audience. In addition, the application sought to be competitive as an investment research tool and provide

expected functionality such as browsable stocks, charting views, market news, user logins and stock watchlists.

Colours are essential for accessibility. The application had to cater for the visually impaired with a higher contrast ratio. The application's choice of colour had to limit tones that impact users negatively, such as red. A medically verified study on cognitive psychology suggested red tends to be interpreted as urgent and dangerous, which can result in elevated blood pressure and increased heart rate [6]. It is essential to note this research was for Western markets. The psychological impact of red would need to be reconsidered when targeting foreign markets, as in China, red is considered a symbol of luck and fortune [7].

Most investment research websites turn their users' portfolios red to indicate losses, from the company name to the entire area chart [Fig. 1]. The overuse of red in financial platforms, a colour which has negative connotations associated with danger and urgency [6] in Western markets, is troubling. However, as stocks are volatile and risky, red was used minimally on the solution's User Interface (UI) to communicate pessimistic price predictions. Issues with the overuse of red can be found in competitor applications such as Yahoo Finance's area chart [Fig. 1]. When the daily performance is negative, the whole page creates a sense of doubt in the entire portfolio. Research [8] has identified that investment behaviour changes in correlation with how red is displayed. When a stock price is shown in red, risk appetite reduces, and investors lower their expectations for future gains.


**Figure 1. Yahoo Finance's area chart [9]**

Fearmongering is an issue with media globally as it attracts a reader's attention when telling people that their financial security is at risk [Fig. 2] and harms investor confidence. A study [1] identified that fake news strongly influences the stock market, especially during economic and political uncertainty. Research showed that COVID-19 misinformation negatively influenced investors' sentiment and returns.


**Figure 2. October 2022 Google news search "Stock market" [3]**

As discussed, fear-mongering is a concerning issue; this is where sentiment analysis comes in. The solution programmatically provides a sentiment classification on each market article, so the end user's feed is sorted by positive articles first, offering a friendlier experience. The platform gives users the option to avoid stressing over the headlines. Of course, this depends on the user's personality, as users often continue reading when presented with negative headlines.

The application additionally offers time series forecasting to predict the future movement of stock prices. The accuracy of time series forecasting and impression on the user must be

critically evaluated to understand better its viability of aiding the user. Having code estimate future stock movements is a much-researched topic rapidly growing in the fintech industry. AI has already played a massive role in anything from stock picking to high-frequency trading [11]. Additionally, according to Blackrock CEO Laurence Fink [11], the abundance of freely available information on companies has made active fund managers research further into AI and big data, as they are no longer the only investors with insights. The project continues the trend and contains performance-efficient price forecasting as a proof-of-concept integration yet to be widely publicised within the investment research space.

Ten anonymous participants engaged in usability testing involving qualitative and quantitative methodologies [Section 5]. They evaluated and interacted with the application to assess the design's effectiveness and identify improvement areas. Usability testing explored topics ranging from the design and accessibility to the psychological impact of trusting code to perform accurately in a dynamic and high-risk environment like finance.

## 1.3 Scope and Objectives

1. The overarching aim of the application was to create an enterprise-level progressive web application for investment research using best practices:

I. The application must be mobile-compatible to reach a broader audience, which can be accomplished using a component library (a collection of reusable UI elements).

II. The application must allow for extensive testing and the creation of code documentation. Swagger [12] is a toolset that meets the criteria for this objective. Swagger offers better testing and documentation of Application Programming Interface (API) controllers through a custom user interface.

III. The application can utilize feature flags to demonstrate industry-level configuration management. To achieve this, a software service ConfigCat [13], offers industry-recognised management.

IV. The application must be accessible to all. Therefore, more than one API is needed to retrieve relevant market data, as gathering from multiple sources minimises cost. Code should be used to limit the number of calls leaving the server side to gain data from external servers, decreasing the chances of external errors.

V. API rate-limiting could ensure robustness and aid the server side from being bombarded by malicious requests.

VI. Code must be secure when dealing with users' accounts and API keys linked to personal funding. Security can be demonstrated by utilizing a secrets manager on the server-side application to replace hardcoded API keys and database connections containing credentials.

VII. Following this, the login and registration system must be secure, never storing user passwords directly. A secure system would offer a user claim identification system like JSON web tokens. It would hash and salt passwords before storing them to minimise the risk of password cracking if the database were hacked.

2. The design must be accessible and up to recognised standards through research of the navigation, layout, colours, font styles and font sizing, resulting in an accessible, positive, and simplistic experience for all users.

I. The application's content must be dynamic and interactive and offer the choice of different charting views, such as candlestick and area charts.

II. The application must have up-to-date market news and searchable equities to ensure it is competitive and usable as an investment research tool.

3. AI must be integrated to evaluate its impact on the end-user:

I. Sentiment analysis must be integrated into the website to assess the sentiment of individual articles and provide positive news first sorting of the feed, unknown to the user, to create a healthier mindset.

II. Time series forecasting is needed to evaluate its probability of being a valuable tool to the user. For example, forecasted prices based on current historical trends could aid a user in avoiding potential losses.

4. The application must facilitate registration and users being able to log in to their accounts. Upon registration, users could browse and add stocks to their watchlist, resulting in the subsequent display of that stock on their dashboard.

5. Administrators will be able to log in to the oversight dashboard with logs of user actions, such as adding items to their watchlist, registering, logging in, signing out and more.

6. Interview participants' feedback must be critically evaluated and uncensored to deliver impactful changes and analysis of future work.

## 1.4 Achievements

The project achieved all the criteria listed in the scope and objectives [Section 1.3]. The application was implemented, including all the outlined code integrations, best practices, and security measures. The application is mobile-compatible and accessible. The accessibility was evaluated through Google Lighthouse scoring [Section 4.5] and extensive usability testing [Section 5], which identified that 80% of interview participants found the design more accessible and colour-contrasting than its competitors listed in section 2. Strengths in design were identified by participants in the charting component, colour theme and ease of navigation. Usability testing suggested that 80% of participants additionally approved of the sentiment analysis integration as they could see how the programmatic interpretation of terminology could benefit new investors. A dynamic and performant time series forecasting model predicts stock prices and displays the result to the user. Further work identified in section 6.3 is required to improve the models continuously.

## 1.5 Overview of Dissertation

- Section 2: State-of-The-Art critically evaluates existing investment research platforms. Three competitors Bloomberg Terminal [Section 2.2.1], Yahoo Finance [Section 2.2.2] and Google Finance [Section 2.2.3], are evaluated based on their strengths and weaknesses, identifying the need for a more accessible investment platform.

- Section 3: .NET server-side discusses the design and implementation of a server-side application which formats, stores, and retrieves the information required by the client side to offer its functionality. This section is highly technical, taking a developer's point of view as the user does not interact directly with the server side.

- Section 4: Angular client-side discusses the design and implementation of the client-side application, which is the interface users will interact with. This section has more of an emphasis on accessibility and user interaction. This section also evaluates the proposed solution's Google Lighthouse page metrics compared to its free competitor platforms, Google Finance and Yahoo Finance [Section 4.5].

- Section 5: Usability testing discusses the design and implementation of a research-administered interview in which ten anonymized participants were observed interacting with the application and two competitors. The interview was followed up by an anonymized survey to allow for statistical measurement. The results from these methods are then critically evaluated.
- Section 6: Conclusion summarises the project. The results were evaluated, limitations were stated, and suggestions for future development were proposed.

# 2 State-of-The-Art

This section aims to critically evaluate the strengths and weaknesses of popular investment research tools. Three leading research platforms were selected, namely Bloomberg Terminal, Yahoo Finance and Google Finance, based on the complexity of the interface, evaluated in descending order from most to least cluttered. The diverse sample offers a broader scope of the investment research tools available and a more effective insight to support the need for the proposed solution.

## 2.1 Investment research websites

### 2.1.1 Bloomberg Terminal

Bloomberg Terminal [14] is a famous investment research tool used by institutional finance practitioners and protected behind a $24,000-per-year paywall [15]. The common issue with accessing market data and insights is the cost. Bloomberg Terminal offers more than just investment research. However, the main benefits from a research perspective are that it covers everything, has live data on global markets, commodities, and analysts' actions, and facilitates contacting other investors, allowing for purchasing their research reports. The design is confusing [Fig. 3] yet consistent, which would be profoundly challenging to implement with the sheer volume of data in the system. Without years of experience in the finance industry and paying for the expensive license to access it, the platform is not a viable tool for most investors.


**Figure 3. Bloomberg Terminal [16]**

### 2.1.2 Yahoo Finance

Yahoo Finance [17] is a popular free tool that offers superb insights. Its strengths include customisable charting views, analyst opinions, market news and conversations, and a feature which acts as a forum for each stock. However, the 'conversations' feature damages Yahoo Finance's effectiveness as a reliable source of information. The conversations are filled with negative sentiment bots that purposefully spread fear to push prices down. As discussed in section 1.2, misinformation damages investors' sentiment. The issue is so widespread a user has published a guide on identifying falsified statements posted by other users [18]. From a design perspective, Yahoo Finance's navigation [Fig. 4] is cluttered as it is intertwined with Yahoo's other services. A good rule of thumb for designing websites is to limit the number of options on a navigation bar. A short navigation bar is faster to read, and if users have difficulty navigating, they are more likely to exit the website [19].

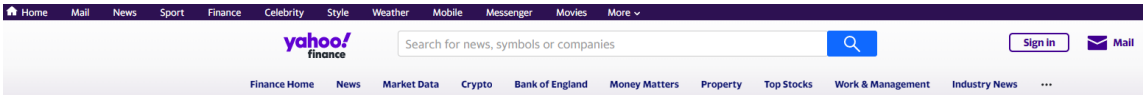



**Figure 4. Yahoo Finance's navigation bar [17]**

[Fig. 5] is an example of the full-screen charting view. It is excellent and allows for drawing trend lines. The page is covered in adverts, links, and metrics of other assets, such as stocks and cryptocurrencies, which is confusing. If a user searches for a specific stock, they could be frustrated being supplied with the performance of other assets in the same field of view. The recently viewed stocks are intended for easy navigation; however, users would probably expect links to other equities on the search page instead.

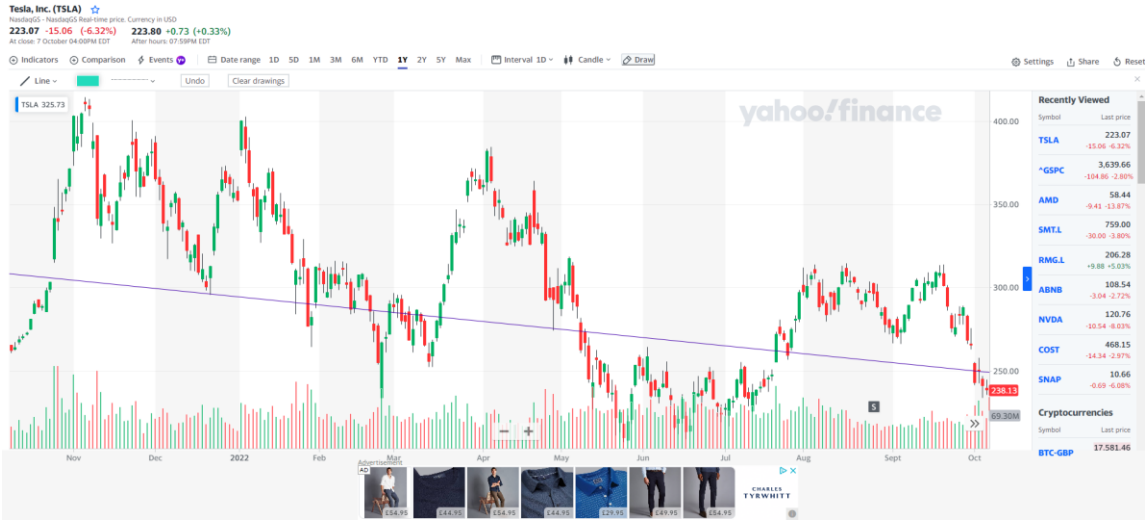



**Figure 5. Yahoo Finance's full-screen candle chart view [17]**

### 2.1.3 Google Finance

Google Finance's [20] design is simple, which promotes ease of use. Although it only boasts a line chart view which offers the user no flexibility, not having a candlestick chart is detrimental for advanced investors. The navigation feels more accessible than Yahoo Finance's navigation. There are fewer options, simplifying the user experience. Fewer options on a navigation bar improve user experience [19]. Google has markers on the stock graph with special dates and shows the reaction in price movement, a well-implemented and unique feature [Fig. 6]. The proposed solution aims to match or beat Google in ease of use. However, Google Finance's use of colour is its crucial weakness in accessibility. They overuse shades of light green and red [Fig. 6], which can be hard to identify for users affected by visual impairments, where the daily movement elements would appear identical until closer inspection [Fig. 7]. However, to negate issues caused by this, Google has implemented arrow icons to indicate the stock's movement direction. Another weakness of Google Finance is that it offers limited information compared to its competitors.

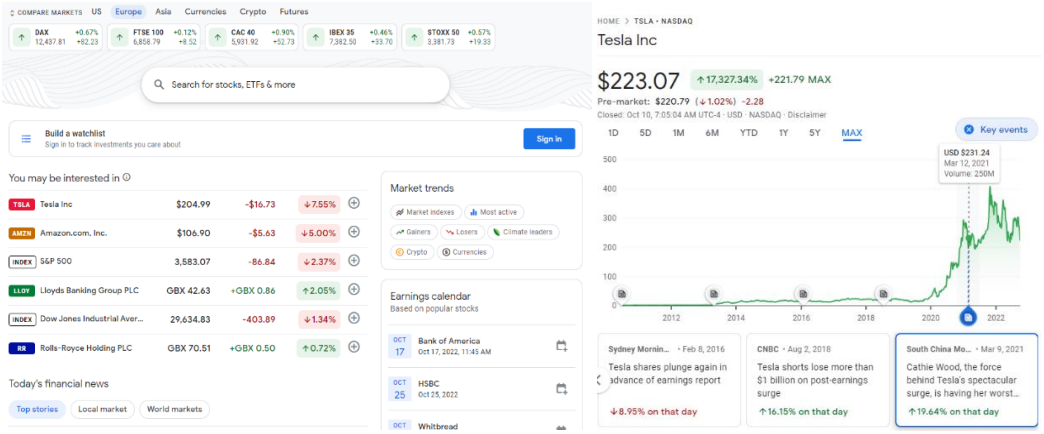



**Figure 6. Google Finance landing page [20] and chart markers on a stock page**

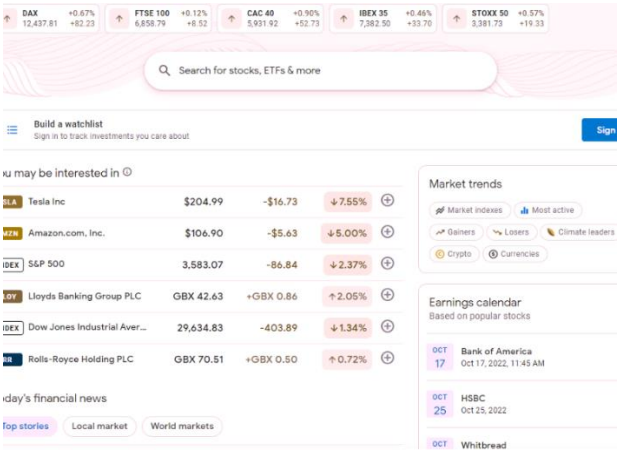



**Figure 7. Google Finance in colour-blindness (Green/red) simulator [21]**

### 2.1.4 Comparison

Compared to its competition, the proposed solution can be better represented in a table [Table 1]. The table only contains features where answers differ between platforms.

| | Bloomberg Terminal | Yahoo Finance | Google Finance | Proposed Solution |
|---|---|---|---|---|
| **AI assistant** | X | X | X | Sentiment analysis and time series forecasting. |
| **Ability to change chart view** | ✓ | ✓ | X | ✓ |
| **Free access** | X | ✓ | ✓ | ✓ |
| **Design ranked by simplicity** | 4th | 3rd | 2nd | Target: 1st or 2nd |
| **Use of colours** | Downloadable Colour Vision Deficiency (CVD) support [22]. | Good balance. A clear sense of primary colour (blue). | Overuse of lighter shades, minimalizing the contrast. | Utilizing contrasting colours and darker shades of greens and reds where they are needed. |

**Table 1. Comparison of investment research platforms**

Regarding the cost-free competitors, competing with Yahoo and Google Finance was not an easy task. Yahoo Finance is packed with information, and Google Finance has thought heavily about its accessible design. The proposed application took inspiration from Google Finance's design and improved upon its work. The application focuses on investor guidance and chart customisation like Yahoo Finance. In addition, using colours with better contrast scoring helped deliver a free service that caters to a broader audience.

## 2.2 Accessibility

The UK government has regulations [23] to ensure public sector bodies offer perceivable and operable services to all users regardless of disability. The web was created to be a free and open resource for everybody, with equal opportunity and access. An easily interpretable design allows those with disabilities to participate. Research [24] has identified that the average user has better task completion time and a higher satisfaction rating when using a website designed with accessibility in mind. Clever design benefits everyone, including companies that gain and retain a larger audience. According to research published by Harvard University [25], increasing retention by 5% can increase profits by up to 95%; this is how companies like Apple have performed well against competitors.


**Figure 8. Accessibility statistics from the UK Click-Away survey [26]**

Focusing on designing easy-to-learn and use products [27] increases the retention rate and, as a result, boosts profits in online e-commerce. The most significant accessibility issue is knowing whom to design for. Organisations such as The World Wide Web Consortium (W3C) [28] research and advocate for accessible best practices to be enforced as standard. For example, W3C states that colours must be a 3:1 contrast ratio on their background. In addition, a 4.5:1 ratio is recommended for users with letter contrast sensitivity. The British Dyslexia Association reiterates the contrast standard set by W3C. It offers a style guide [29] that states fonts that have generous spacing, such as Sans Serif, Arial or Calibri are suitable. Web pages should avoid underlining or setting text in italics as they make it appear more crowded.

## 2.3 Sentiment analysis for articles on markets and securities

Sentiment analysis, or opinion mining, is a Natural Language Processing (NLP) technique used to extract, analyse, categorise, and determine the attitude towards a topic. Among the best-

performing models are Long Short-Term Memory (LSTM) or other recurrent neural networks (RNNs) models [30]. Researchers using sentiment analysis have benefited from abundant accessible online data, such as Twitter opinions and facts in news articles. However, evidence suggests classification accuracies in multiclassification problems have dropped by as much as 50% when using Twitter posts as the dataset due to sentiment analysis's inability to recognise sarcasm [31] efficiently.

Sentiment analysis is an invaluable tool for market analysts [32] as it can help determine which direction investor sentiment is heading. The project extracts the positive or negative tone from news article headlines. According to Forbes, expensive systems created to analyse enormous quantities of tweets can still fall victim to misinformation [33]. Therefore, the application only analyses news article headlines as they are less likely to be an opinion and yield higher classification accuracy than Twitter sentiment analysis due to impartiality.

Utilising sentiment analysis in the application could deliver a healthier browsing experience, promoting positive articles first. The sentiment model was implemented using ML.NET Model Builder [54] [Section 3.2.7] and evaluated with performance metrics and user opinion. User opinion provides insight into the effectiveness of the feature and its positioning within the application.

## 2.4 Time series forecasting to predict the future movement

Time series forecasting is a machine learning technique that predicts future values by analysing data with associated time intervals [64]. The assumption is that future movements will trend analogously to historical movements. Analysts commonly adopt forecasting to estimate financial performance, resource management, infection spread, product demand, etc. Despite its importance, producing high-quality forecasts appears challenging. The main concerns of forecasting are whether the data and its source are reliable and how often the data is updated.

Predicting the market has been a widespread interest of research. Limitations in accuracy tend to be caused by the surprise of geopolitical tensions or negative company earnings. Research has indicated that adding additional factors to models, such as sentiment or political news, could combat limitations and increase stock price prediction accuracy [34]. Currently, popular models in the field are built using LSTM neural networks. LSTM is an improved version of popular recurrent neural networks due to their ability to store input data in memory. LSTM time series forecasting uses cells to predict sequential data by selectively remembering or forgetting information from earlier data points. The cells are typically organized in layers, each processing a particular aspect of the data. However, LSTM models are challenging to train due to their memory bandwidth. There are currently popular tools available for machine learning, notably Prophet by Meta. Prophet [35] is a Python/R-based library that makes creating forecasts easy.

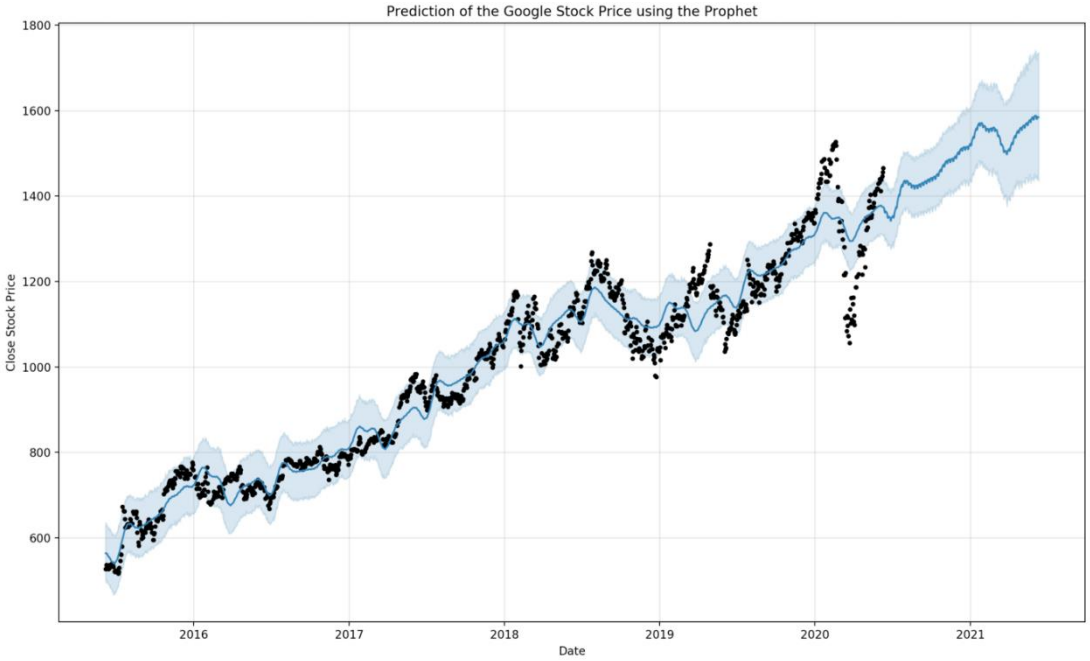



**Figure 9. An example of forecasting using the Prophet library by Meta [36]**

The proof-of-concept integration the application set out to solve is whether a prediction model can be integrated into a server-side API for dynamic and time-efficient forecasts. The application utilized Microsoft's performant ML.NET library. The project needed to demonstrate how the integration would be achieved. Therefore, linear regression was used as it is currently available within the ML.NET ecosystem. This is a foundation to expand and use more complex models in the future, as Microsoft announced their intentions to add native support for deep learning methods such as LSTM [37]. Microsoft plans to additionally improve the consumption of cognitive models through services such as Microsoft's cognitive toolkit [38] and TensorFlow [39]. Compared to the Python ecosystem, the lack of documentation and polished implementation (the frequency of modifications in design through new releases) is limited. Creating a deep learning model currently in .NET, in most cases, would require the consumption of pre-trained models over time inefficient APIs. The implemented regression model is quick and trained using the fed market data.

The model has been evaluated by metrics such as mean squared, absolute mean squared and observed price movement. If AI can correctly predict which way stock prices will trend even as a proof-of-concept regression model, its role in the application would benefit the user.

# 3 .NET server-side

This section aims to highlight and justify the design and implementation choices for the server-side application responsible for providing functionality to the user interface (later introduced in section 4). In addition, industry-recognised tooling to heighten security, robustness and efficiency is demonstrated to deliver a cost-effective and competitive solution. First, the .NET architecture advantages [Section 3.1] and code integrations that play a vital role in the configuration are discussed [Section 3.2]. Then the custom endpoints that provide a gateway to functionality are shown in a high-level overview [Section 3.3 – 3.6].

## 3.1 The .NET API implementation

The investment research solution was created as a modular client-server distributed application that splits the computation between the .NET server (backend), which acts as a provider of resources and the animated client-side (frontend) Angular progressive web application, which delivers a responsive user experience. The applications communicate with each other using RESTful application programming interfaces (API), which allows for the secure transfer of information across the web. The server-side application was constructed with Microsoft's modern .NET six framework [40], with sound reasoning. .NET six is the fastest full-stack web framework [40] that delivers competitive performance. Additionally, it has intelligent code editing and extensive documentation, and compared to its predecessors, it can be used to create more time-efficient APIs. These benefits were crucial in catering for the profuse creation of controllers and services the client-side application required to function as intended.

Furthermore, being developed by Microsoft, .NET is cross-platform compatible. The framework was designed to emphasise security, from encryption, access control, and protection against SQL injections and cross-site scripting to continuous rollouts and security patches. .NET delivers the native groundwork required by a financial research platform. The .NET framework thrives due to its simplicity and built-in functionality [41], creating a perfect community for developer productivity and an inflow of third-party code libraries.

## 3.2 Configuration and tooling code integrations

A high-level system diagram [Fig. 10] aims to communicate a simplistic overview of the system. The users and administrators can interact with the client-side application, which retrieves its functionality from the server-side (backend) application. Microsoft's Azure Key Vault secrets manager [Section 3.2.1] and ConfigCat [Section 3.2.3] secure the server. User information and audit logs are stored within a local MySQL database [Section 3.2.2]. External data providers provide market information to enhance the experience and provide up-to-date information to the user base [Section 3.4.1]. Responses are cached to limit resource usage and are occasionally updated with another request to the external servers by a scheduled code service. The market data is fed dynamically into the ML.NET machine learning models [Section 3.2.7] on the server side for enhanced insight. All these features are testable using a user interface provided by Swagger [Section 3.2.4].

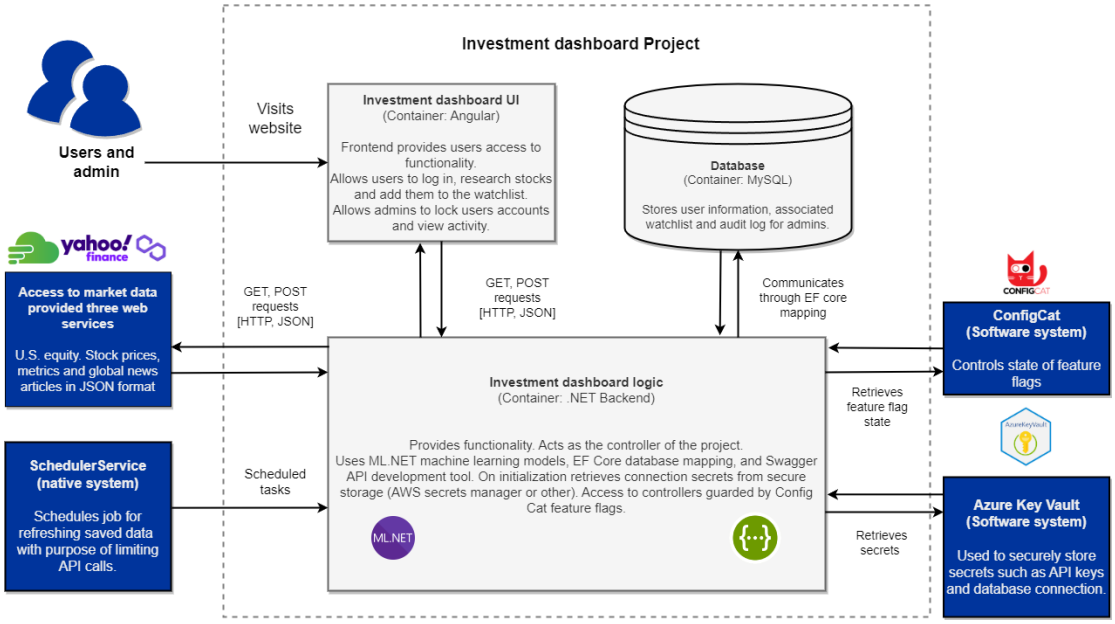



**Figure 10. High-level system overview diagram**

### 3.2.1 Secure secrets management

The server-side application needed to provide the client-side application with abundant market data and account-specific information. Providing an informative and data-heavy service, collecting partial information from third-party data providers was required. The providers use unique API “keys” to associate usage limits and resource costs with a particular identity. If the application’s identifying “keys” were hard coded in a JSON configuration file, that creates a vulnerability as an individual could come across the code’s repository and use the keys for malicious exploitation or an attack to increase resource costs [42] for the initial proprietor. Secrets management is a method of managing an application's credentials, keys, and tokens. It acts as a central source of truth where key values can be easily modified and updated across the application. The designed system uses Azure Key Vault [43] as the Microsoft development ecosystem was becoming familiar, and they offered a fantastic service in terms of cost, speed, and support.

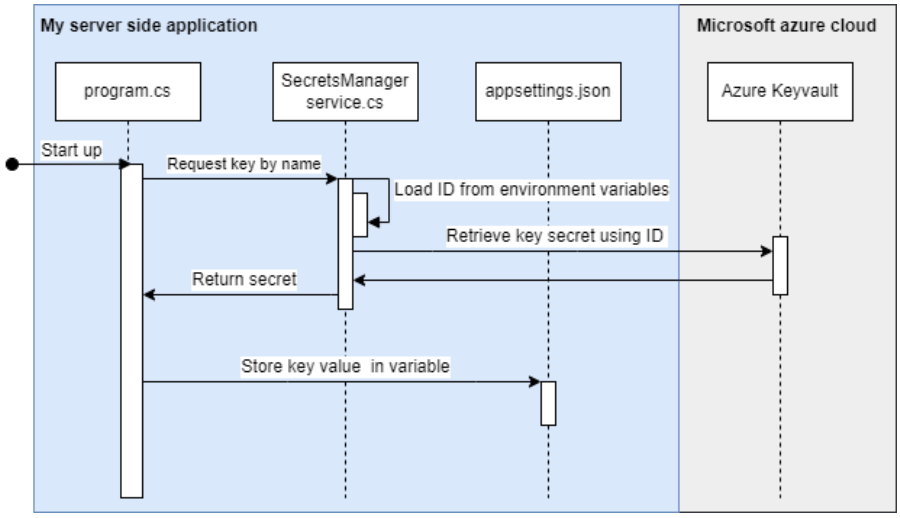



**Figure 11. Sequence diagram of Azure Key Vault operations**

The diagram [Fig. 11] shows that upon startup, the program file initialises a reference to a custom secrets management class, where code calls Azure Key Vault and retrieves a key secret

for the requested key name. Currently, the secrets needed to contact Azure's servers are vulnerable, but in future work, they could be securely stored as environment variables in a virtual container [Section 6.3]. The key values are retrieved from Azure and dynamically loaded into empty variables in the application settings JSON file. The market and news data services require the values to access data from third-party suppliers; therefore, the application retrieves secrets from the configuration file's variables upon the first usage of the services. The connection string needed for the database is additionally retrieved securely from Azure Key Vault. The keys remain secure as the values are not hard coded within any file and are removed from system memory if the application shuts down.

### 3.2.2 Database scaffolding

The proposed system required a database to store user credentials, audit logs, unique user watch lists and reading lists. Microsoft's Entity Framework [43] was the preferred choice. It is an object-relational mapper that enables code first creation of a database. Entity Framework creates tables from column and datatype information for each object and runs command lines to generate the mapping file. This allowed for the efficient creation of the MySQL database at initialisation if one does not already exist. This is ideal for scalability as Entity Framework is cross-platform and provides the ability to create "migrations" that can modify the existing database using a command line. The primary outcome is reduced development time through code-generated mapping and lowered resource costs as it performs fast CRUD (Create, read, update, and delete) operations in .NET-based systems [44].

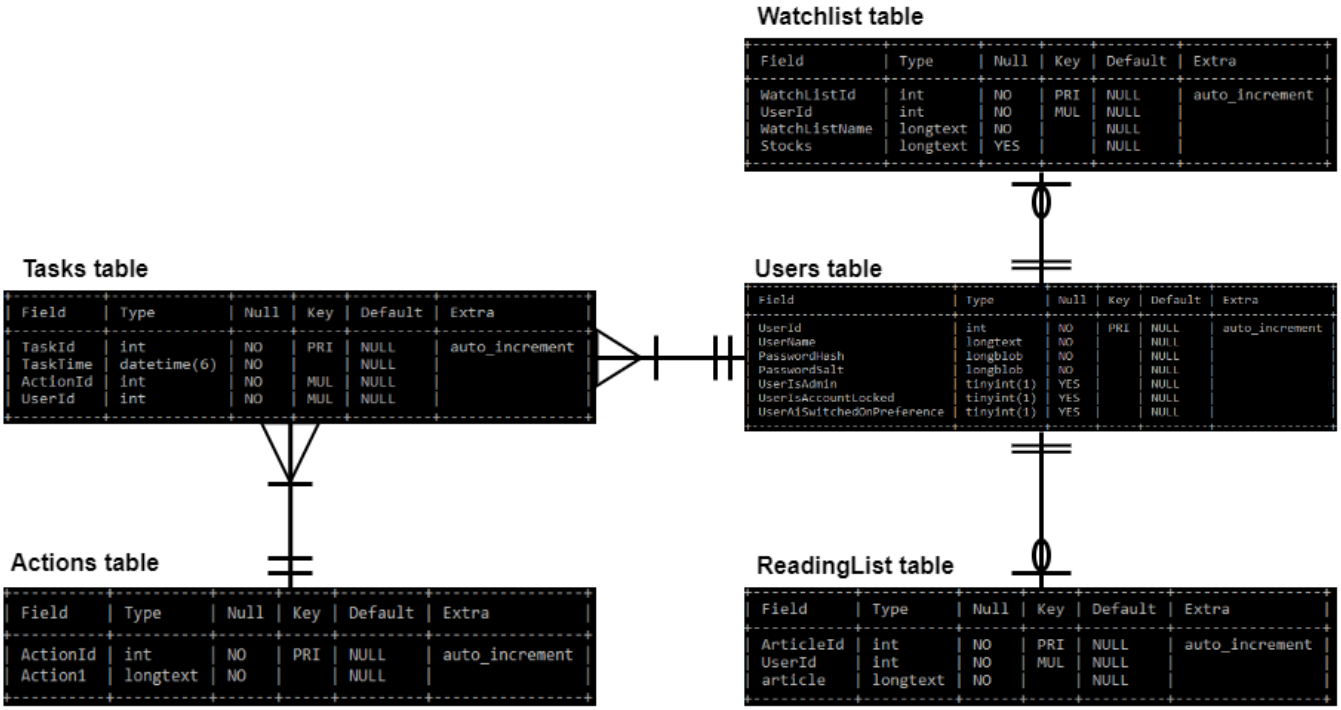



**Figure 12. Entity relationship diagram of the generated MySQL database**

The database consists of five tables [Fig. 12], which stores and logs user-related information. The code first migration generates the tables upon initialization. The user's table consists of an autoincrementing identifier, password hash and salt, and account-specific binary settings such as whether their account is locked or they have artificial intelligence features turned off. Each unique user has one optional watchlist and reading list. Once created, the lists cannot be deleted from the database, although if empty, it will appear as if the user doesn't have one on the interface. The user's table is additionally related to the tasks table where their actions

are logged. Activities are logged for each account, and the registration process is logged; therefore, the relationship is not optional. Each task has a unique identifier, user identifier, task time, and action id. The action id maps the user's action to descriptive text for the administration page [Section 4.4.5]. This is why there can only be one action per logged task.

### 3.2.3 Configuration management through feature flags

Feature flagging is a software engineering technique used heavily within the industry, as apparent by ConfigCat's cliental [13]. Feature flags enable the toggling of functionality (on and off) during runtime without forcing active users to experience any downtime [Fig. 13]. Instead of needing multiple code versions, a feature could be displayed to active users through one quick remote condition change. If a feature is causing a bug, it can be remotely deactivated instead of temporarily taking the whole website down. An online survey found that 63% of IT professionals reported better testing of features or higher-quality software because of feature flagging [45]. The server-side applications functionality endpoints are contained within feature flags provided by ConfigCat [13], an industry-praised hosted service that allows for environment (development/test/production) specific flags to be controlled remotely.

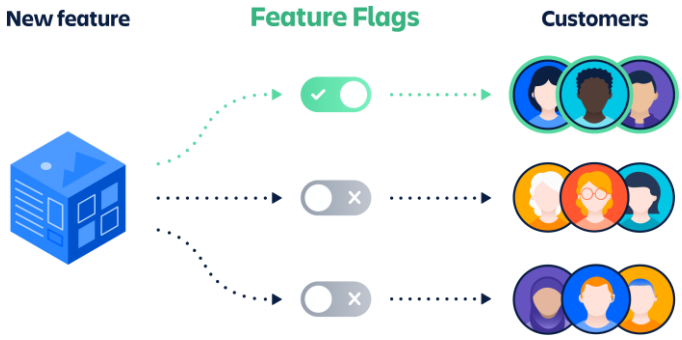



**Figure 13. Feature flag visualisation [45]**

The server-side application has a service folder for feature flagging where code is waiting to communicate to ConfigCat's servers to check the status of a requested flag. Every one of the controller methods that users can request data from is wrapped in a conditional statement containing a flag name. The flag name is created by the user and groups all endpoints using it, if the flag name was "LoginFunctionality" and it was also applied to registration, then they could both be turned off or on at once. This service is programmatically called, and the application asks ConfigCat for the flag's status in the application's current runtime environment. If false is returned, the user is blocked from accessing that feature. This integration is cost-free, helping keep development and future hosting costs low.

### 3.2.4 API development and testing tool integration

As the server side required various endpoints, implementing and documenting them became significant. The client side is heavily reliant on information from the API. Therefore, testing functionality was vital. This is where Swagger, an API development and testing tool, was adequate. Swagger [12] is a tool that creates interactive and readable REST API documentation. When integrated into the application, Swagger created a user-friendly interface to conduct improved testing of the application's controllers. In Industry, Swagger's interface is often used to demonstrate a product's capabilities and design early on to potential clients and product managers as it is easier to interpret than a code editor [46]. Swagger enables a developer to take a top-down design first approach, where documentation and design can be written first before finishing the functionalities coding.

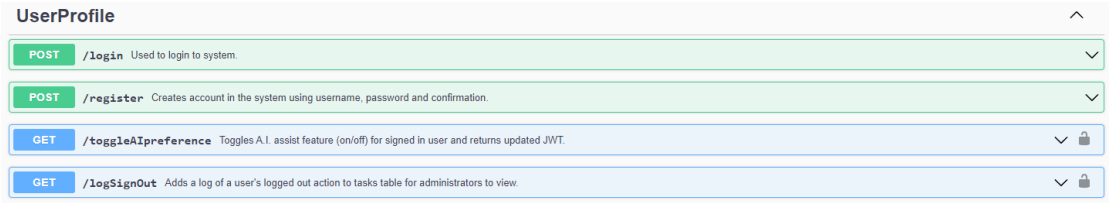



**Figure 14. Colour-coded endpoints on the Swagger user interface**

Endpoints are colour-coded [Fig. 14] by operation and can be categorised based on which controller file the methods are placed within. For example, the application has four controller classes, admin, market data, market news and user. All endpoints within the admin controller would appear under the admin category, so the platform's code is well-organised and friendly to future developers. The application's swagger-related instantiation code is wrapped within a conditional statement so that the user interface is only accessible in the development phase.

As the platform is based on users being able to have a role-based account (user or administrator), authentication is required in the .NET application so anonymous users cannot access endpoints as a default. This implementation required the addition of data annotations above controller methods accessible by anyone, such as the login and registration functionality. Accessing the authenticated endpoints requires the bearer token returned from a successful login, which must be pasted into the authorization popup [Fig. 15] added to the swagger UI. Users cannot access administrator-only endpoints as the login returns a role-based JSON web token. The authorization prevents attacks where malicious users frequently request insecure endpoints, often resulting in inflated usage costs, compliance issues, and reputation loss [47].

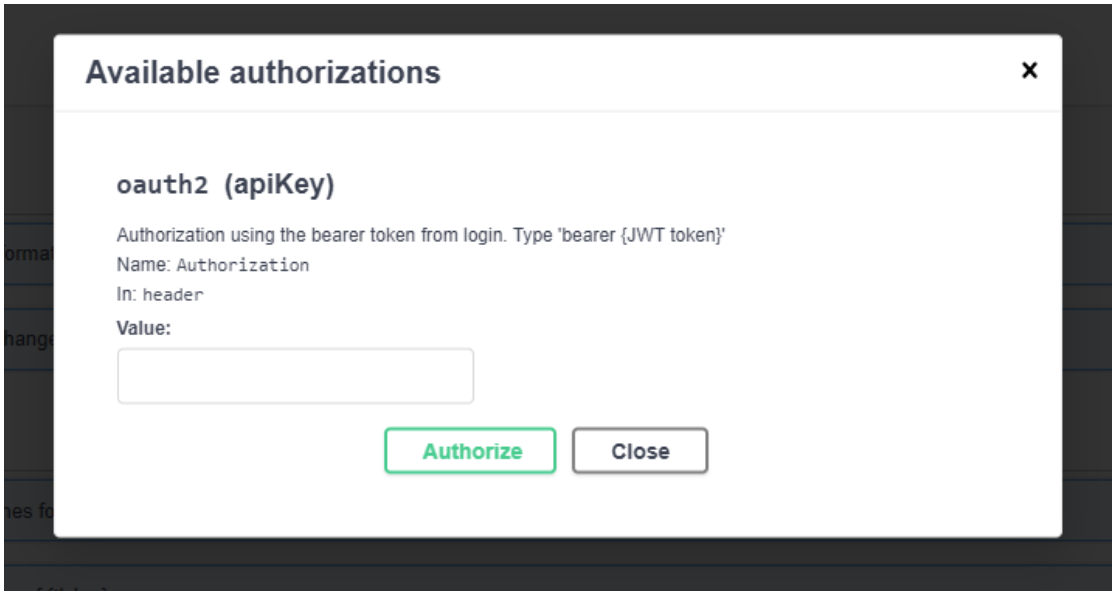



**Figure 15. Swagger user interface authorization popup**

### 3.2.5 JSON Web Token (JWT) authentication

A JSON web token (JWT) based system is a security method that exchanges information between a client and a server. JWTs contain encoded user claims, a name-value pair that identifies the requester. The JWT is digitally signed to ensure the user's claim's contents cannot be modified once issued. Once a user registers, a new JWT is generated with an expiration date, like the login process. The token is returned to the client to store as a cookie in local memory. Every request to the web server whilst the user is logged in will contain the JWT for the system to authenticate the user.

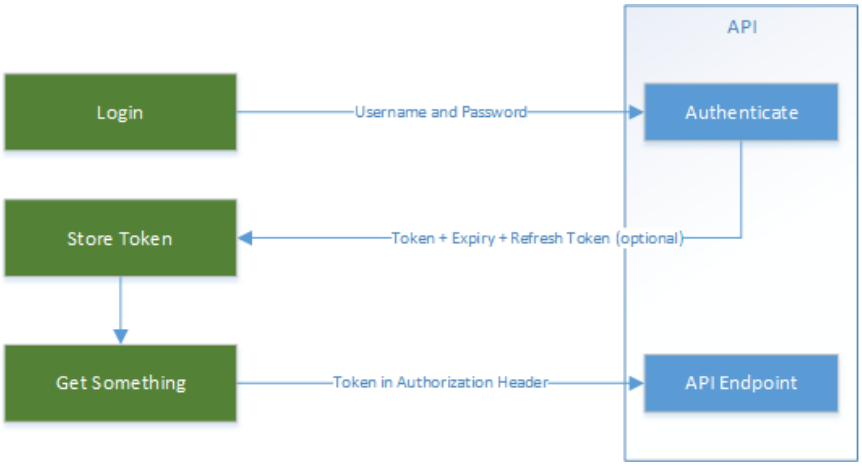



**Figure 16. Simple diagram of basic JWT operations [48]**

When performing endpoint testing using the Swagger user interface introduced in section 3.2.4, login and register are the only endpoints that allow unidentified requests. Upon completing either of these actions, the returned JSON web token can be entered into the authorization popup, which keeps the developer signed in and allows access to endpoints that will return account-specific information and market data. The main advantage of utilizing a JWT system is that there is no need to use a database to store or verify tokens, which results in faster development time. Additionally, having relevant information within the contents of each JWT request speeds up processing, as accessing a username and expiration date is quick and easy.

### 3.2.6 Endpoint Rate-limiting

The application sends and receives a large quantity of API requests, which presents a problem. When the prototype budget is a constraint upon API usage, it is worrying that all it could take is one malicious user spamming the refresh button on resource-heavy pages to cause disruption. Creating an API rate-limiting service was vital to negate the risk of such an attack. In networking, rate-limiting controls the “rate” of requests the server sends or receives [Fig. 17]. Rate-limiting helps prevent Denial of Service (DoS) [49] attacks where a malicious user overloads a server with web traffic and limits web scraping. Rate-limiting is necessary as the application contains multiple APIs with quotas for external market data providers. There is a fee for every request over the free or agreed-upon limit.

Due to an authorization-only request setting on the endpoints returning market data, which only registered users can take advantage of, there are limitations in the options an attacker could consider. Additionally, implementing Azure Key Vault effectively hides API access codes and connection strings. Therefore, the most significant remaining threat is web scraping [50]. Web scraping is frequent practice nowadays. A new competitor could use knowledge of the Python programming language [51] and make a quick script to syphon market data from the dynamically generated webpages once they understand the website’s URL structure. With rate-limiting, if the server is overloaded with requests in a specified period, an error is thrown, and the usual flow of execution is interrupted. The period and request threshold per user is defined in the app settings configuration file of the server-side application. A user cannot make more than five requests in eight seconds. This is enough for a prototype application as it gives room for a refresh, a miss click and the right navigation option. With the short delay in page loading, this is more reasonable than it sounds. If a user were to refresh maliciously to consume API usage, the system would block the request from executing.

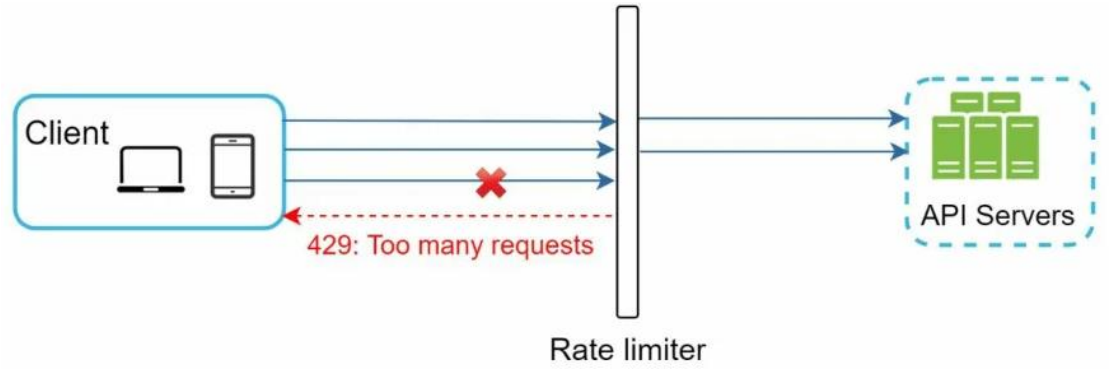



**Figure 17. Visualization of security benefit gained from rate-limiting [52]**

### 3.2.7 Machine learning integration

ML.NET [53] is an open-sourced, cross-platform machine learning framework. It prioritises time efficiency as being built with .NET allows for high-performance optimizations like parallel processing. The framework provides regression, classification, clustering, and anomaly detection. ML.NET offers Model Builder, a user interface [54] for limited use cases that allow for quick tuning of hyperparameters and evaluation metrics. Model Builder-created solutions can be exported as a zip file, and when placed within an application, they can be consumed through code. The proposed solution's sentiment analysis model was created with ML.NET's Model Builder and was trained using a dataset obtained from Kaggle [55]. This online community offers datasets and valuable tutorials for data science. The selected sentiment dataset contains 5791 rows of financial headlines with predetermined positive or negative values. In tests, there are two critical metrics. Micro-accuracy is a robust evaluation metric used to determine the average accuracy across multi-class classification problems where the dataset may be imbalanced. Micro-accuracy is a more indicative metric to measure performance than macro-accuracy, which computes the average accuracy using the arithmetic mean. Hence micro-accuracy is less likely to contain bias.

The final dataset was more accurate (>70% micro-accuracy) than other tested datasets. Other datasets included a wider variety of classifications such as "neutral", which resulted in negative and positive headlines being wrongly classified into neutral sentiment delivering the user the least helpful experience and yielding poor accuracy scores [Fig. 19]. The best-performing model discovered by the ML.NET Model Builder was Stochastic Gradient Descent (SGD), which gave a micro accuracy of 0.79 and a macro accuracy of 0.77 [Table. 2] for the binary classification dataset. The final model [Fig. 18] can be accurate in strongly-worded statements.

**Top 5 models explored**

| Rank by performance | Trainer | Micro Accuracy | Macro Accuracy | Duration (seconds) |
|---|---|---|---|---|
| 1 | SgdCalibratedOva | 0.7906 | 0.7717 | 25.0 |
| 2 | LbfgsMaximumEntropyMulti | 0.7899 | 0.7682 | 19.1 |
| 3 | LbfgsMaximumEntropyMulti | 0.7873 | 0.7697 | 14.8 |
| 4 | LbfgsLogisticRegressionOva | 0.7873 | 0.7679 | 23.3 |
| 5 | LbfgsMaximumEntropyMulti | 0.7872 | 0.7712 | 14.5 |

**Table 2. Model Builder Sentiment Analysis results**

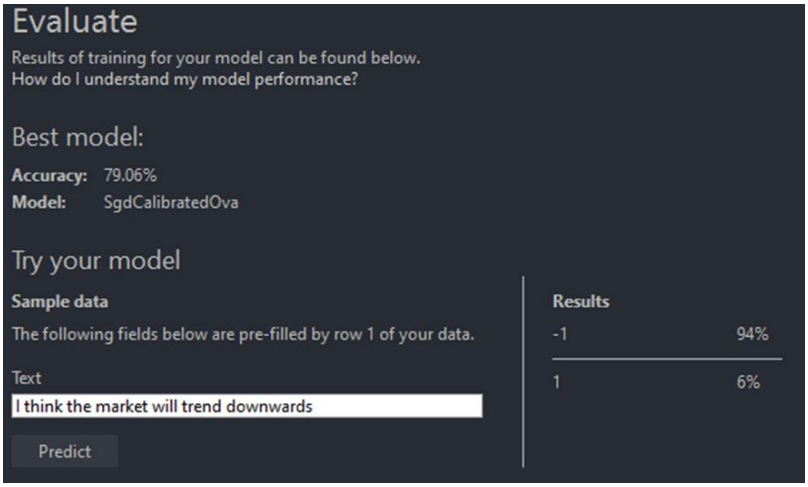



**Figure 18. Final sentiment analysis model**

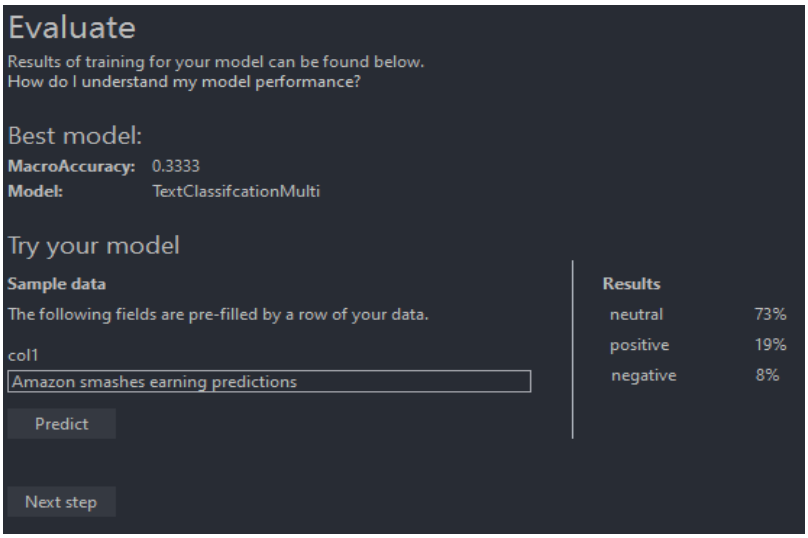



**Figure 19. A failed model that included "neutral" classification resulting in low accuracy**

Microsoft's ML.NET Model Builder offered an inadequate time series solution for the application's specific needs due to the parameter setup and pre-trained nature of the solution. Therefore, custom code developed created a suitable model which could train in real-time. JSON values taken from an API request are converted into an enumerable list of objects and then loaded into a data view fed into a tuned time-series forecasting pipeline. During development, a text file of predetermined JSON with correlated price and date values allowed for the creation and tuning of the model to mitigate the expenses of consuming further API requests. The resulting model is a proof-of-concept demonstration of a fast and dynamic training to-web integration using a Single Spectrum Analysis (SSA) model for univariate time series forecasting [56]. SSA is a flexible and robust way of making predictions, often applied to financial, meteorological, and industrial applications [57]. It groups historical values into components such as trends, noise, and periods of differential fluctuations in the mean of the time series (Oscillations), which affect the forecasted result.

The hyperparameters were manually tuned and tested. Window size, or length, ensures that the signal and noise components are separated. It is known as the most crucial parameter for the tuning of the model as it typically represents a season/period. Window sizes of twelve to ninety-one days were explored through manual tuning, and a window of twelve days was chosen to simulate the short-term emotion-based nature of the market. As part of the tuning

process, a window size of ninety-one days was explored as it represents the number of average active trading days in a quatre; surprisingly, this performed poorly in comparison. The series length is the number of data points used when performing predictions. Therefore, this is set to the total size of the input. Similarly, the train size is the same as the series length, which indicates the total available data points. A horizon indicates the number of periods to predict; it is set to thirty to predict one month. Finally, a confidence level percentage suggests the likelihood that the observed values will fall within the lower and upper prediction boundaries. Confidence percentage is vital as without a high value, predictions will vary more in price; therefore, a selected value of 95% decreases dramatic fluctuations.

The deployed model dynamically loads, trains, and predicts price movements in a recorded average of under two seconds which is impressive considering the model uses years of daily values passed through an API to the client-side application and displayed on a graph. However, the model's performance is another significant factor in assessing its viability. Therefore, cross-validation was not used, as it would be more time-consuming and delay the page load speed. However, this may be possible with additional resources a commercially deployed system provides. The model's performance was validated on live data during deployment. First, the data, which is unique per stock from the first recorded date to the most recent, is spit into training and testing data (80%:20%). The 80% training data forecasts the most recent 20% of prices. Next, evaluation metrics were calculated by comparing the difference between the predicted and testing dataset's values. Afterwards, the entire dataset (100%) is refit for training to get the final live prediction values of the future horizon (one-month). Finally, tests were conducted on live data by noting observations of one-day, one-week and one-month values and logging Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

**Predicted and observed valuations, forecasting metrics and accuracy**

| Predicted or observed? | 1-day (6/03) | 1-week (10/03) | 1-month (03/04) | MAE and RMSE metrics | Average difference | Correct direction? |
|---|---|---|---|---|---|---|
| **Rio Tinto (RIO)** | | | | | | |
| Predicted | $74.89 | $75.31 | $76.85 | 9.518, 11.132 | | |
| Observed | $72.71 | $67.54 | $68.09 | | 8.637% | No |
| **Advanced Micro Devices (AMD)** | | | | | | |
| Predicted | $86.33 | $72.34 | $82.98 | 16.189, 26.362 | | |
| Observed | $81.16 | $82.67 | $96.56 | | 11.925% | Yes |
| **Unilever PLC (UL)** | | | | | | |
| Predicted | $49.78 | $49.00 | $46.34 | 1.544, 2.100 | | |
| Observed | $49.50 | $48.74 | $52.70 | | 4.661% | No |
| **AbbVie (ABBV)** | | | | | | |
| Predicted | $152.81 | $168.48 | $174.48 | 10.075, 18.100 | | |
| Observed | $155.28 | $149.71 | $160.42 | | 7.265% | Yes |
| **Meta Platforms (META)** | | | | | | |
| Predicted | $194.89 | $216.89 | $243.63 | 40.737, 58.170 | | |
| Observed | $184.90 | $179.51 | $213.07 | | 12.5% | Yes |

**Table 3. Time series forecasting predicted and observed values**

The model's Mean Absolute Percentage Error (MAPE) was 8.998%, of which 60% (3/5) of stocks moved in the correct direction (positive or negative) [Table. 3]. MAPE measures prediction accuracy for statistical forecasting methods, highlighting the mean error percentage. These results were poor, considering the lack of risk appetite associated with personal finances. However, during the observation period, there were impactful headlines that negatively impacted the market. The Bank of England and the United States Federal

Reserve made statements [58][59] regarding raising interest rates, which scared investors and caused an adverse reaction in the stock market. An additional influencing factor that distilled panic in the global markets was the news of the Silicon Valley Bank's (SVB) collapse [60], resulting in a chain reaction of bank withdrawals where the Federal Reserve had to intervene with emergency funding [61]. Additional factors include Credit Suisse's bailout [62] and tensions escalating between the United States of America and Russia [63]. Unfortunately, these headlines are a strong example that it is challenging for a predictive model to predict short-term fluctuations caused by negative investor sentiment accurately. The research conducted in section 2.4 identified that this issue of predictive models' negligence towards human influence was common. Perhaps a longer horizon and observation period would have introduced more stability.

The proof-of-concept model lacks accuracy compared to an LSTM deep network. If, as mentioned in section 2.4, future support is to be released by ML.NET, a minor code modification would be required. However, the input and output formatting foundations are there, so the model is in a strong position for future refinement.

## 3.3 Freely accessible user endpoints

### 3.3.1 Registration functionality

To deliver a customisable user experience, each user must be unique. Therefore, to design a registration system, the following criteria were required:

1. The publicly displayed identifier used, in this system's case, a username, must be considered unique by the application, therefore requiring a restriction disabling duplicate usernames.
2. The account details needed to be stored on the server side using a database.
3. A password must only be known by its creator, therefore, cannot be stored in the database directly without a hashing algorithm and unique salt.
4. The system must prompt the user to confirm a new password to compare and ensure the user did not mistake their intended password.
5. The system requires basic standards to be met, such as a password's minimum and maximum length.
6. A login session must be limited by time constraints, meaning if a user has not accessed their account in a while, they must be prompted to reverify their credentials to ensure it is the user logging on from their device.

The registration endpoint allows for anonymous connections, meaning no authorization process blocks an unidentified user's API request. A ConfigCat feature flag for the registration process is checked when the endpoint is triggered. The feature flag allows developers to disable the registration process for unregistered users if an identified error exists within the database. The requested username, password and password confirmation are sent to the server as a custom data annotation class. Data annotations [65] have metadata for validation and formatting. Typically, required length, type and regular expressions are applied to the input fields. If the input passes these built-in rules, additional validation ensures the information complies with the expected format, passes a custom profanity filter, and the username does not exist. The three validated values are used to create a unique HMAC-SHA512 password salt and hash, which is stored in place of the entered password within the database.

HMAC-SHA512 [66] is a key hashing algorithm that mixes a secret key with the message contents and hashes the result. The resulting hash is scrambled with the secret key again and hashed again. The application's secret hashing key is loaded securely from the Azure Key Vault secrets manager for uncompromised security. The benefits of HMAC-SHA512 are that the messages are tough to decipher, even upon interception. Typically, this algorithm is used for passwords, personal financial data, and secure messaging [67]. After this process, a new user is stored within the database with the default parameters such as AI assistant turned on and user group set to user, not admin. Next, a user claim identifying the user in API requests is created and signed with the symmetric security key. The expiration date of the login session is set to one week, and username, user role and AI assistant preference are added to the payload so they can be used for later functionality. The payload is unmodifiable once created, and the entire message is used to create a JSON web token (JWT). The JWT is returned to the user so they can be redirected to the authorized pages and store a cookie allowing for uninterrupted site exploration, freely passing any authorization checkpoints. When the user's JWT expires, they are redirected to the login page and cannot access content until they reauthorise themselves.

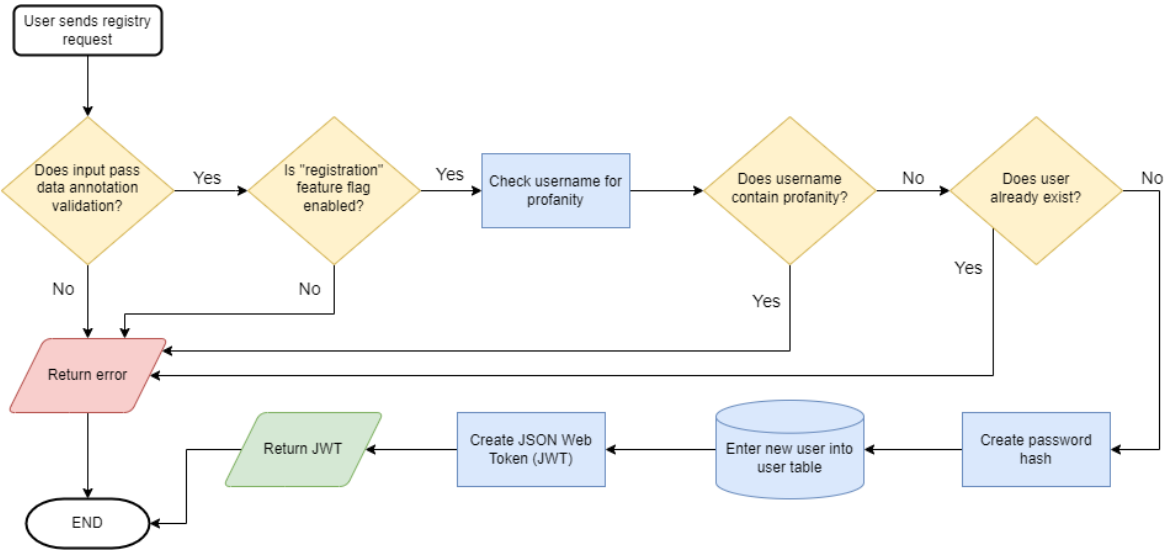



**Figure 20. Flowchart demonstrating the registration process**

### 3.3.2 Login functionality

Upon successful registration, a user is provided with a JSON web token (JWT) that grants access to content through the front end. Similarly, this is the result of logging in. Like the registration configuration addressed earlier, the login endpoint is open and accessible to all requests. The API request must pass feature flag and data annotation validations. Upon passing these checks and receiving a username and password, the system searches to determine whether the username exists in the database. If the user does exist, the entered password is salted and hashed with the stored salt and hashing key. The resulting hash is compared to the securely stored password hash. If the two hashes match, the user's account status is checked to see if an administrator has locked their account, and if not, a new JWT is generated and returned to the client.

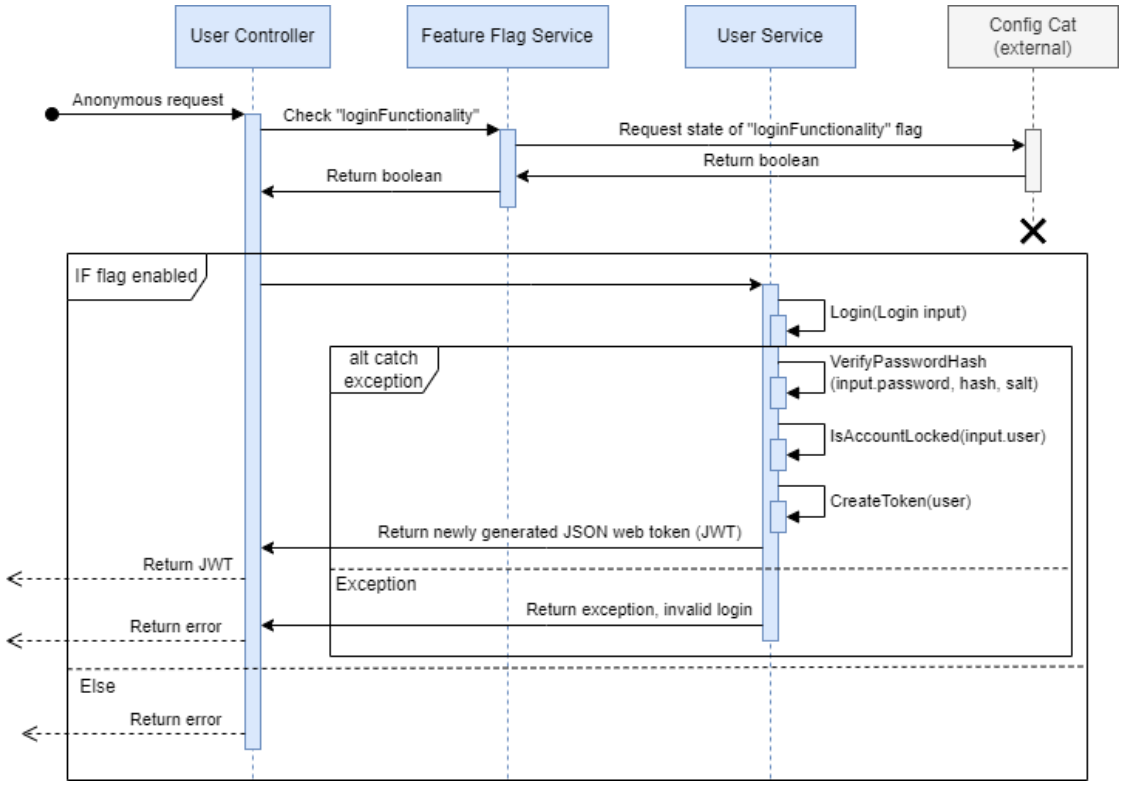



**Figure 21. Sequence diagram showing how a login request is managed**

## 3.4 Shared endpoints

### 3.4.1 How market data is retrieved cost efficiently

Market data is retrieved from three diverse providers to minimise costs and source reliable data for the application. The application has one paid subscription for a market data provider, which would be able to manage a more considerable amount of traffic and respond to customer queries with performance-conscious speeds. Yahoo Finance's API [68] was the best fit for the required use cases. Yahoo Finance's API was cheap ($9.95/month) for a monthly quota of ten thousand requests and delivered much of the functionality needed by the application. However, the limitations are that the service only offers U.S.-listed securities. The other two data providers, Finnhub [69] and Polygon.io [70] provide daily information such as global market news and snapshots of U.S. listed equities. Having data that only needs to be requested once every so often works well as the application can save the response and replay it to any users who require it within a relevant period. Effectively, the application could extract the offerings that delivered high quality from each service and merge them to provide the end user with a more satisfactory experience.

Initial responses are stored in static variables to minimise the requests for the two free providers. These responses get replayed or accessed throughout sections of the application. During the application's development, this resource reduction technique proved vital in cost reduction. Eventually, towards the production stage in the development cycle, to rigorously evaluate the usability of the front end and spot errors, the primary provider's (Yahoo Finance) plan had to be upgraded to a paid subscription. Therefore, minimising resource usage costs through caching responses is critical. However, the predominant issue with this is that the stored information relating to global news, company information and companies listed on the market frequently changes. Nevertheless, the efficiency of caching responses remains beneficial to the application thanks to a custom-implemented task scheduler dependency

injection. The task scheduling code executes automatically every twelve hours, going into the market and news data services to refresh the stored information.

### 3.4.2 Search bar functionality

One of the main requirements for the application is to provide search functionality so that a user can search for the company they are looking for. Without a search bar, there would be no logical way to navigate the website. The design of the search bar's backend operations uses zero external API calls once the market data service is initialised, as it would be one of the most interacted with features of the site; therefore, it could be too computationally expensive. A slightly delayed request (300ms) is sent to the market data controller when a user searches for a stock on the front-end application. Once authorization and feature flag checks are passed, the user's search value is cleaned to search a list of pre-initialised stocks created at startup. If the list contains the lookup value, one up to a maximum of five results are selected using an Excel spreadsheet (ticker to company mapping dataset) obtained from Kaggle [71]. The tickers are mapped to the company name for improved formatting and a more descriptive result set. If the ticker exists in the list of stocks but an associated company does not in the spreadsheet, the ticker is returned by itself. If the pre-initialised list of tickers does not contain the lookup, the mapping dataset is consulted directly, and one result or an error message is returned. See diagram [Fig. 22] for further explanation of how a company name search would return a more descriptive result.

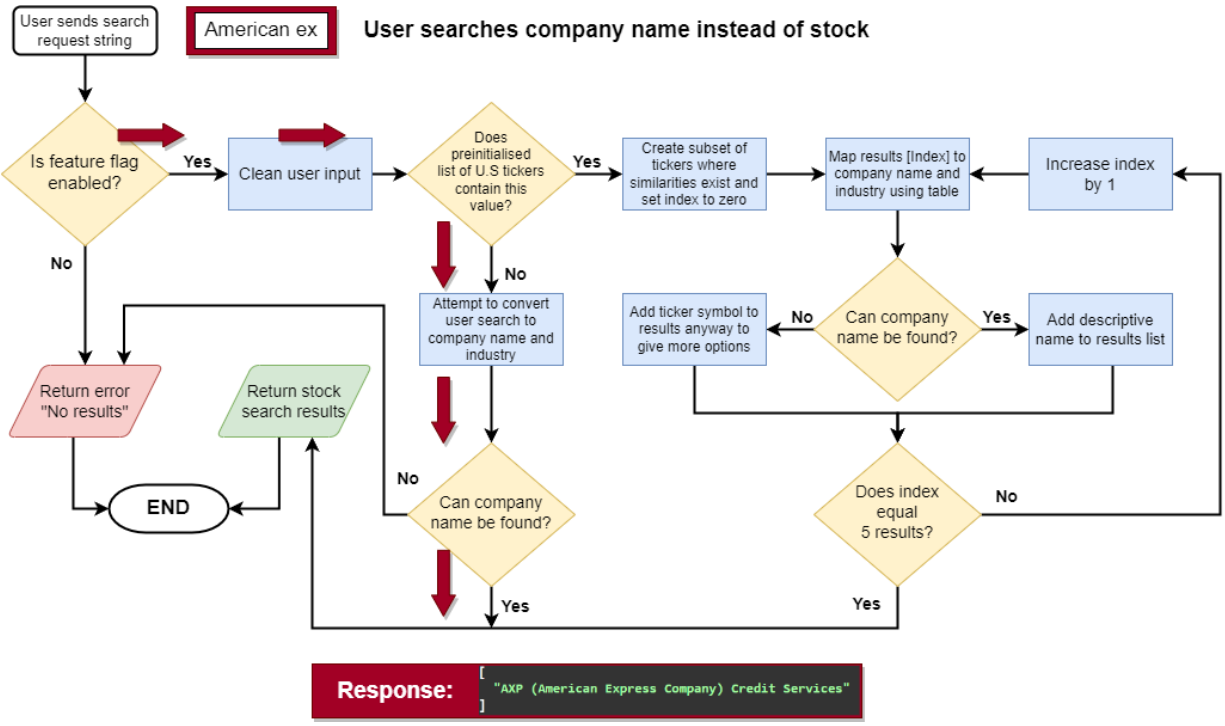



**Figure 22. Flowchart of search bar functionality, red arrows showing the flow of a company name search**

### 3.4.3 Stock details, history, and forecast future prices

The most significant consumption of resources in the application is the proposed stock details page. This is the page where more detail is shown on a specific stock. This page requires different charting views, company information, metrics and forecasted prices, so it requires multiple dataset sources.

First, the requested stock ticker is checked to see if it is the same ticker as the last request. If it is, the data is not loaded twice, and a cached response is returned. This further limits a malicious user's options to increase resource costs. Next, the stock's entire price history, including daily open, close, high, and low values, is requested from the Yahoo API. The retrieved dates and associated prices are stored as JSON objects and fed into a forecasting pipeline to gain price predictions for a one-month horizon. Then, the list of average predicted values is returned to the market data service. The values are then added to the JSON Object as a node with the key "predictions".

Additionally, two more nodes are added. They include company details and financial metrics, which are populated using methods that facilitate API calls to external data providers and response validations. Finally, the stock details JSON object is complete and saved along with the ticker name for limiting API calls if the user makes the same request consecutively, and the response is sent to the client side.

### 3.4.4 Daily news and sentiment analysis

An essential part of this project is the offering of news articles. To be competitive with other investment research tools, the progressive web application needed to offer market updates such as interest rate hikes, stock movement explanations, etc. The application uses an API from Finnhub [69] that provides a large variety of professional articles in a JSON format. The server side's responsibility is to perform sentiment analysis and allow the user to save an article for later reading before it disappears. Catering for these operations required a custom news controller with its endpoints wrapped in a unique news feature flag. The endpoint methods allow for the retrieval, addition, and removal of articles to a user's reading list.

Users who visit the news page will retrieve articles from the server application. Again, for efficiency and cost limiting, the previous responses are saved. If the news service has been initialised, it will return the daily articles without using a second request to the provider's servers as the response is cached. Upon the initial run of the news service, the response from the provider is iterated through, and each article is fed into a sentiment analysis model [Section 3.2.7], which returns a sentiment classification 'positive' or 'negative' and adds it along with a unique article ID to the JSON response. The classifications from the model sort articles in a positive to negative order, and the unique ID allows for adding and removing reading list operations.

## 3.5 Account specific endpoints

### 3.5.1 Article reading list

As previously mentioned, the unique ID attached to an article identifies which article should be removed or added to the user's associated reading list. The account-specific endpoints are all located within the user controller and service. Adding articles to the reading list checks the user's claim and identifies the user by their username stored within the unmodifiable JWT contents. Next, the selected article id the user requests is checked for validity. If the article exists, the user's reading list is selected. If the reading list does not exist, it is initialised upon adding an article. Otherwise, the article is inserted into the user's existing reading list. The removal of articles follows a similar process. Again, the system checks if the user's reading list exists and removes any article with the unique ID if it does.

### 3.5.2 Stock watchlist

The stock watchlist functionality is additionally positioned within the user controller with its functionality feature flag. When a user creates an account, they can add a stock to their

watchlist to see the individual daily price movement. Users can additionally remove the stock or change the watchlist's title. The user's watchlist is initialised the first time a new user saves a stock, only after the stock is validated to ensure it exists, and then the title of the watchlist is set by default as "My watchlist". Users can update their watchlist title to a name of their choice, assuming it complies with length validation and passes the profanity filter [Fig. 23] implemented using a library named Profanity Detector [72]. If the watchlist name is unprofessional or invalid, the user has an error displayed.

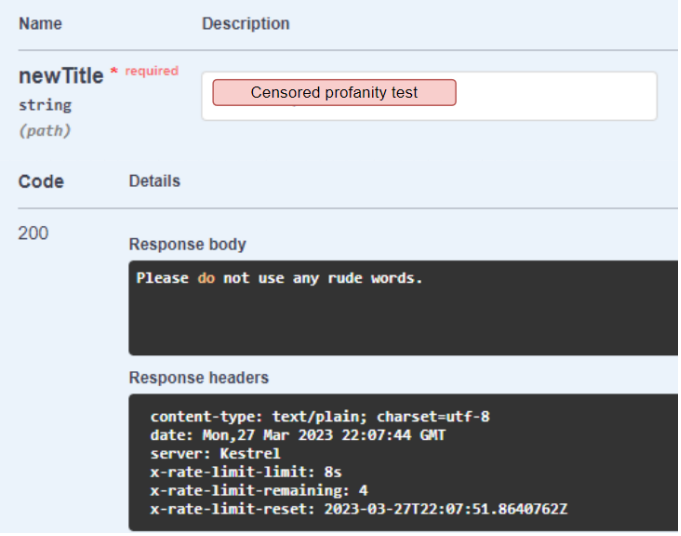



**Figure 23. Changing the watchlist title to profanity using the Swagger interface**

The stocks are stored as a comma-separated list within the watchlist table. When requested, each value is converted back into a list and returned to the user with price quotes retrieved from a method within the market data service. The results are temporarily saved to improve speeds and reduce costs. When a user removes an item from their watchlist, the list of stocks is retrieved from the database and searched for a requested ticker value. If the list contains the stock, it is removed if there is a match. Removing or adding tickers updates the watchlist so it is ready to return the correct information upon the subsequent request.

## 3.6 Administrator role-based endpoints

The user's table has a binary column to identify whether a user is an administrator. By default, this is false to indicate the user is not an admin. If a new user is to be added as an administrator, this must be manually fulfilled as it is the most secure way. An administrator can access more endpoints than a user, and the admin-only features are stored within a separate controller, service, and feature flag. This is so that the administrator-only functionality can be turned off remotely in the unexpected event of an exploit. The administrative functionality includes locking a user out of their account and retrieving audit logs of all the users' actions. Every time a user logs in, logs out, adds, or removes an article or stock from their lists or changes a watchlist title, it is logged to the database with a unique time stamp and the user identifier. The audit logs benefit administration as they allow for a hidden page on the front end where administrators can lock accounts committing suspicious activities, such as spamming actions, to find vulnerabilities. The administrator's actions are not logged in the tasks table.

# 4 Angular client side

This section introduces the Angular client-side application, its benefits, design, and implementation. The design focuses on user accessibility, offering an interactive interface to display the functionality provided by an API connection to the server-side application identified in section 3. The design of the solution has been critically evaluated against its free competitors, Google Finance and Yahoo Finance, on performance, accessibility, best practices, and Search Engine Optimisation (SEO) [Section 4.5].

## 4.1 The benefits of Angular

Angular [73] is a web development framework offering robust tools for building complex applications. Developed and maintained by Google, Angular provides continued version refinement, security, and reliability. Angular's two-way data binding enables changes in the UI to reflect in the underlying data model, while its dependency injection system simplifies development and saves time and resources. TypeScript enhances code organization, error checking, and editor support. Angular's modular architecture allows reusable components and modules to be shared across different projects, improving efficiency and consistency. Angular's comprehensive documentation and large developer community provide support and knowledge sharing. Third-party libraries and tools further extend Angular's capabilities. Widely used and often adopted by large organizations, Angular is a safe and reliable choice for enterprise-level applications. Its benefits [74] make it a strong preference for building reliable, scalable, and complex progressive web applications.

## 4.2 Structure and modules

In Angular, modules are crucial mechanisms that group components, directives, pipes, and services together in files. As more pages are added, these modules can route to each other for modular application growth. However, when modules have too many components and imports, the bundle size increases, which is how much content a user must download when visiting the application. As a result, slow performance and the possibility of the application crashing on mobile systems can occur. The industry standard is that anything over 5MB is unusable for mobile versions of Google Chrome, the most common browser.

To fix performance issues in an application that initially had only two main modules, extensive research into performance enhancements [75][76], including text compression, lazy loading modules, enabling ahead-of-time compilation, and other strategies, were conducted. Significant restructuring and reconfigurations were orchestrated throughout development to lower the bundle size and improve load performance. The structure was modified so that the main module had the smallest number of imports. In addition, the application was more modular so that only the required libraries were loaded on the user's device when they visited a webpage. This restructuring facilitated the lazy loading of future modules and components to improve performance when accessed.

Angular accomplishes this by splitting code into more bundles so that they are only used when required. Additionally, the application takes advantage of ahead-of-time compilation, optimization/output hashing, and build optimizer in the Angular configuration file to achieve competitive speeds.

## 4.3 Authorization guard, HTTP interceptor and router security

In Angular, a “router” enables navigation by interpreting a browser's URL as instructions to change the view. The router can identify active pages and navigate based on conditional statements. Delivering a user-specific experience meant accounts must be unique and secure. This required custom services that play vital roles in the security of the client-side application.

Two of these are guards, classes that contain conditional statements used to allow or deny access to the routing configurations they are added to. The main guard acts as authorization to check that a user is logged in. This guard is added to the entirety of the website, excluding the login page, meaning if an unauthorized user tries to visit a secure web address, they are instead redirected back to the login page. The second guard implemented is a role guard solely for the page that contains administrative functionality. The role guard loads the saved JWT cookie, decodes it and checks if the unmodifiable payload mentioned in section 3.2.5 includes a role variable set to administrator. If it does, the user is approved and allowed to proceed [Fig. 24].

The client-side web application heavily relies on content being supplied from the server. Due to this client-server architecture, security must be stressed, as a user cannot bypass client authentication to gain access to functionality designed for identified users and administrators. In section 3.2.4, the Swagger UI showcase demonstrated that an authentication bearer is required to test endpoints. Adding authentication bearer to requests required an HTTP interceptor. An HTTP interceptor is a mechanism that programmatically intercepts every HTTP request to the server to add various functions. In this application, a header of "Bearer", followed by the stored JWT retrieved from local storage, is used to authenticate the user. This way, communications with the server are secure, and requests will not get denied. The remarkable thing about this Angular mechanism is it only takes as little as three lines of code to add it to all API requests being sent from within a module, allowing a developer to concentrate more on managing the responses than implementing header requests individually.

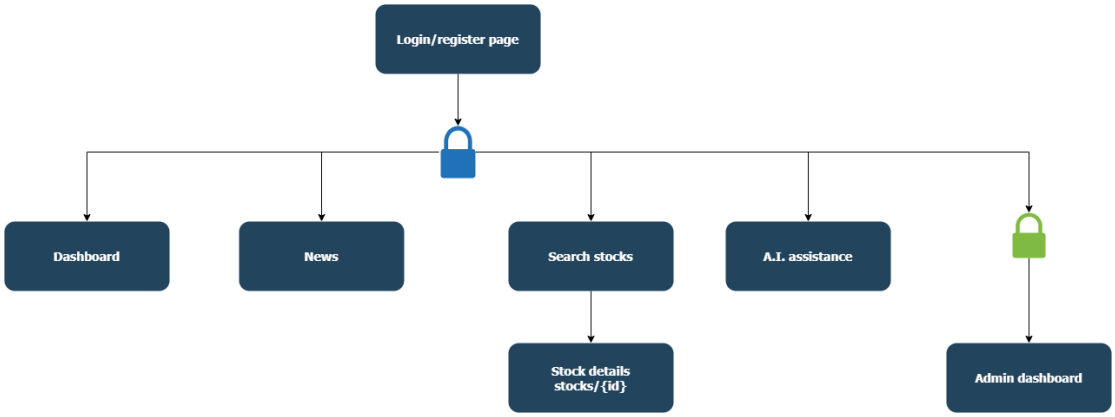



**Figure 24. Simplistic sitemap, the blue lock represents the authorization guard, and the green represents the role guard**

## 4.4 Angular Material, design, and accessibility

The client-side application uses Material [77], Google's official UI component library for Angular. Material is a user interface component library that prioritises consistency through interactive and accessible components such as cards, inputs, tables and more. Furthermore, it prioritises flexible mobile-first development, allowing developers to create an application that is accessible regardless of the user's device. Compared to other UI frameworks compatible with Angular, like Bootstrap [78], it is a popular opinion online that Angular Material has a more flexible and customisable layout [79].

To offer an accessible service to all age groups, dyslexic and visually impaired investors, the frontend design required a primary colour that would provide sufficient contrast for text, which would add appeal to the users. This is why the application's primary colour is navy blue. Blue is the most popular colour for male and female users globally, making up a 33% majority favourite in the United Kingdom [80] as it is often associated with feelings of calmness and

security. According to Adobe, the shade navy psychologically represents tradition, stability, and trust as it is less attention-seeking than a lighter shade [81]. This may sound poetic; however, extensive research has been done on the subconscious impact of colour in marketing, branding, and user interface design. Colour can evoke powerful emotional responses in the user. For example, as mentioned in the problem analysis [Section 1.2], the impact of red and how it can trigger a feeling of urgency. Research suggested [82] red, whilst occasionally being interpreted as a warming colour, is more controversial due to its association with love, anger, and danger. As addressed earlier, a study indicated [6] that exposure to red can cause an increase in heart rate, respiration and elevation of blood pressure, contradictory to the effects of blue, which was deemed calming. Colours are additionally crucial due to their ability to create a visual hierarchy [83], which helps distinguish elements on a page, such as a navigation bar. A study [84] found that a product's colour can influence how consumers perceive its quality.

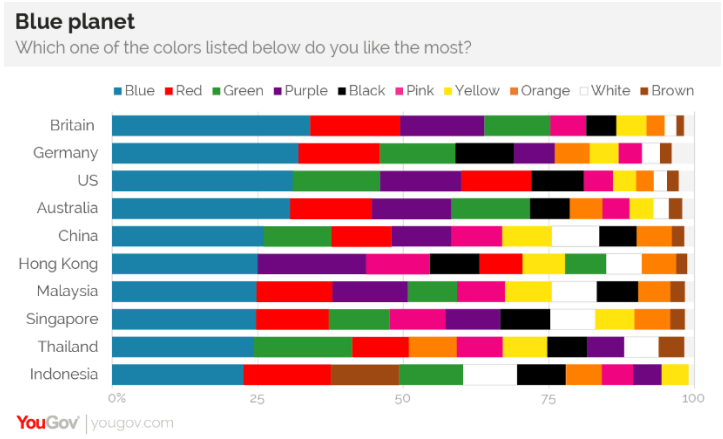



**Figure 25. Chart of favourite colours by nation [80]**

The application's primary colour theme for the navigation bar and other hierarchical areas of interest is a shade of navy (#07203D) featuring white text (#FFFFFF). In digital accessibility, sharp contrast colour combinations make it easier to distinguish text, leading to decreased eye strain, especially in high-glare environments [85]. As identified in the previous discussion of accessibility [Section 2.2], the W3C organisation emphasises the importance of a contrast rating of 4.5/1 when designing for visually impaired users. When entering the application's primary colour palette into a contrast checker website [86], it yields a superb score of 16.37/1.

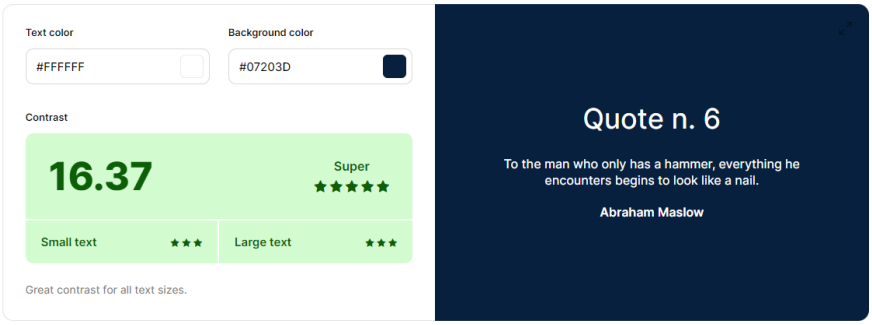



**Figure 26. Primary colour combination evaluated on contrast checker [86]**

Secondary areas with substantial content use the traditional black-on-white colour scheme. It provides the maximum level of contrast possible, which benefits users with visual impairments.

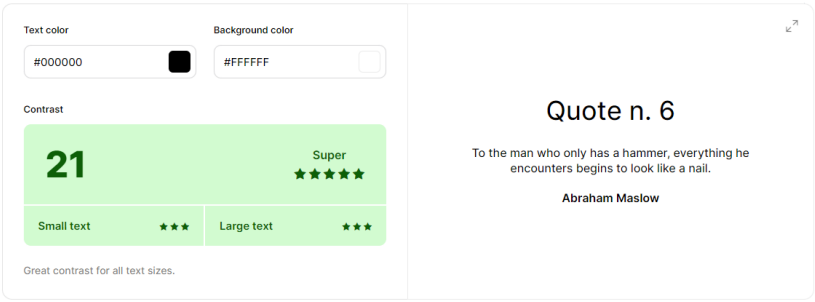



**Figure 27. Content (secondary) colours evaluated on contrast checker**

To further focus on accessibility, the application utilizes a readable and fun Google-designed font, Roboto [87]. As this front comes from the sans serif family, the characters have generous spacing and are easier to read for dyslexic and visually impaired users. Furthermore, the website has a minimum font size of 16px, the amount recommended by the UK government for accessibility [88]. The website makes headers bold instead of underlined for dyslexic users as underlining text can appear crowded, making it harder to read [89]. Additionally, buttons and navigation links are designed to provide further feedback so that when the user hovers over a link, the background changes to a light blue which helps the user navigate and makes the design more interactive. The navigation bar is horizontal for non-mobile screens as it gives more flexibility regarding the design of the page body. The left-to-right style of horizontal navigation bars appears more natural and easier to read to English speakers, which also aids familiarity as most websites prefer horizontal designs [90]. Icons are included in the navigation bar as research suggests icons improve navigation time due to the psychological impact on users and their ability to memorise navigation patterns [91].

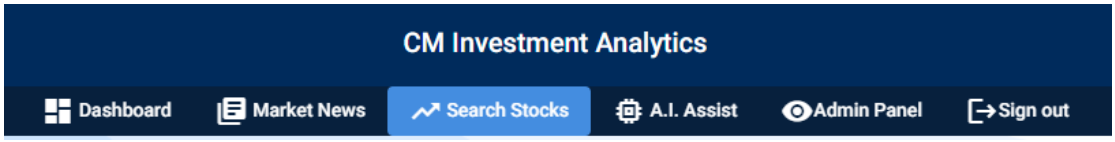



**Figure 28. The active page highlighted on the navigation bar**


**Figure 29. Highlighted table background on hover**

### 4.4.1 Login and registration page

The login page acts as the security wall of the website. The navigation bar keeps the links without the sign-out button using conditional statements. When trying to navigate with the navigation bar, a popup will tell the user to sign in to access these features. The web page's design shows the user what they could do if they registered an account. The background CSS wave pattern shared universally with the other pages provides themed visuals to fill excess space. However, an elevated card offers a high contrast area for content, making the user focus on a specific area.

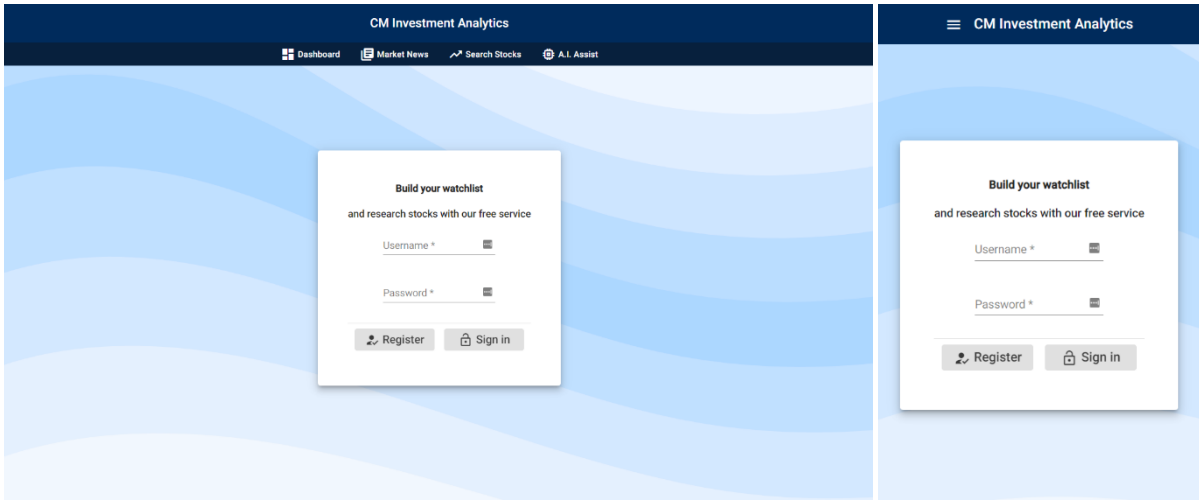



**Figure 30. Login and registration page in desktop and mobile view**

Using Angular Material allows the application to be mobile-friendly [Fig. 30]. It additionally allows for components that provide good user feedback and prompts. For example, if a user does not fill in fields with values that meet length constraints, the buttons are disabled, and the invalid fields will be red [Fig. 31]. If a user completes the input criteria, they can press the buttons. If the server responds with a JWT for the sign-in action, the client-side stores the cookie and redirects the user to the stock search page. Otherwise, users will be displayed an error message telling them the issue with their login details.

An issue resulting in a rejected login is typically due to an invalid username or password, and occasionally it can result from the user's account being locked by an administrator. When the user presses the register button, a confirm password dialog [Fig. 31] will be displayed. If the confirmed password input matches the initial password, the request is sent to the server, and any errors within the validation process are displayed back to the user. Errors sent back from the server vary from existing usernames to profanity filter restrictions on the name. If no errors are present, the redirection action resembles a successful login where the user is redirected and given a JWT to store as a valid session cookie.

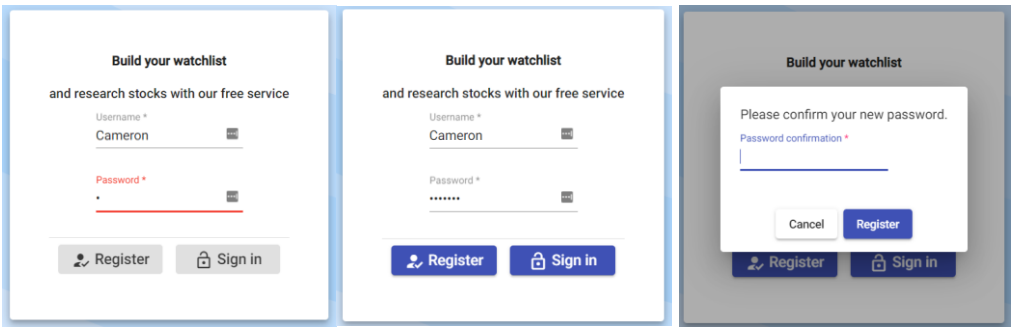



**Figure 31. Input feedback, valid format and registration confirmation dialog**

### 4.4.2 Stock search and details pages

The stock search page [Fig. 32] acts as the glue of the website, enabling the user to connect to additional content that the navigation bar does not offer. The top daily movers (winners) are displayed in a table with the company name, price-to-equity ratio, stock price, average analyst rating (sourced from Yahoo), and daily movement. The tooltips explain how the price-to-equity ratio and how the analyst rating column works. If any metrics are unavailable for the displayed winner, the cell is left blank to indicate no information. Adding a no-value message would make the table look more cluttered and could impair dyslexic users' speed in navigating

the application. Each row is highlighted when hovered and navigates to the dynamic stock detail page on click. The stock search page contains a search bar that returns descriptive results [Fig. 34].

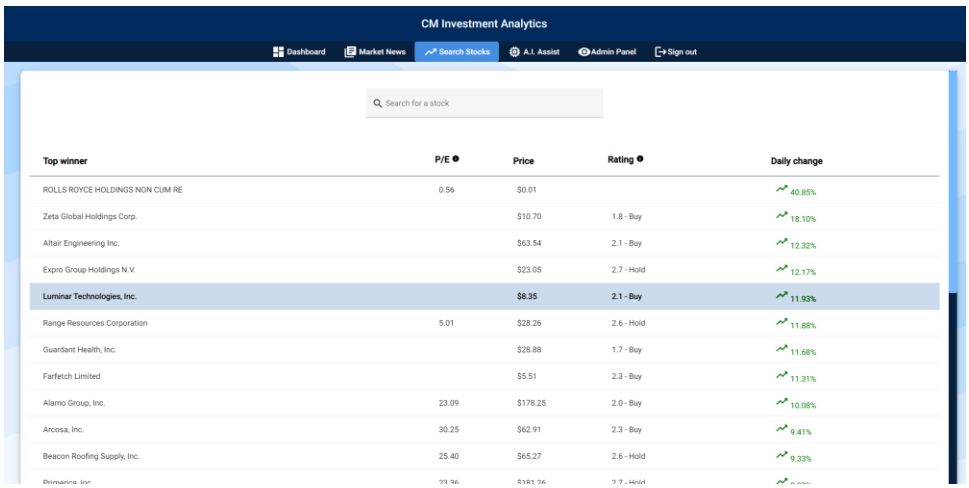



**Figure 32. Stock search page**

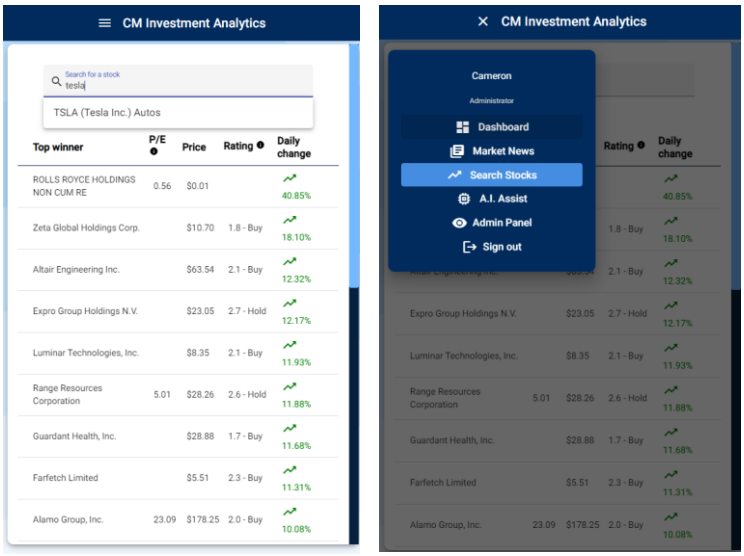



**Figure 33. Stock search page's mobile-friendly table and navigation bar**

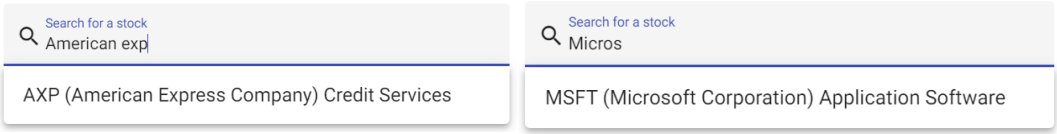



**Figure 34. Stock search bar working with examples**

The stock search page was designed to be minimal and well-spaced to ensure mobile compatibility (later altered in section 5). Daily movers are displayed to promote a healthier mindset towards the markets and to demonstrate that there is an investment opportunity. The design works nicely on mobile by increasing readability through greater line spacing [Fig. 33]. The navigation bar switches from a native horizontal to a vertical and collapsible popup on mobile, which displays the user's name and role. For example, an account with the default user role would have no "Admin panel" navigation button and, upon trying to navigate to the

page through a hyperlink, would be instantly denied and redirected as if the page did not exist. On the stock search page, the vertical scrollbar has been relocated from the default right position to further in with a custom emboldened design. The scrollbar featuring a light blue indicator on the contrasting dark blue background provides a more consistent theme with the rest of the application.

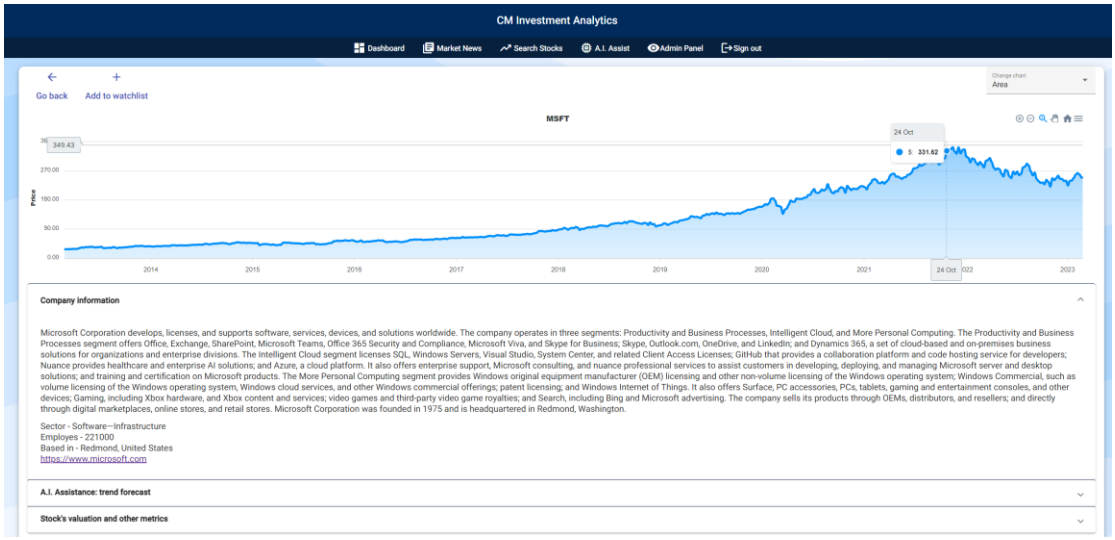



**Figure 35. Dynamic stock details page with default area chart view**

The stock detail page features an interactive and animated chart using the ApexChart library [Fig. 35]. A custom charting component in Angular allows for the injection of dynamic data retrieved from the server's response into the component to create an explorable and scalable chart. The chart can be magnified, zoomed, scrolled, switched to a candlestick view, reset to the initial view, or the user can even download the price history used in the chart as a spreadsheet or an image file free of charge. Allowing the user to download a spreadsheet to conduct any calculations of their own at no extra cost to the application is beneficial for aiding the free spread of information.

ApexCharts offered all this functionality which is part of the reason their charts were the primary choice. Additionally, custom tooltips were implemented to conveniently display the date and price at the location of the user's cursor. On a mobile device, the tooltip would appear when the user holds down the screen. In candlestick view [Fig. 36], the tooltips display each day's open, high, low, and close prices, allowing for a more precise valuation. The stock graph displays a loader animation whilst data is retrieved through an API request. Once loaded, the chart glides upwards using custom animation code.


**Figure 36. Candlestick chart view**

The performant and simple area chart is displayed first as it is more understandable for a wider audience. If the user chooses to see the candlestick chart [Fig. 36] they can switch views. For a stock with a long history of prices, the graph can feel less performant than the area chart due to having four recorded prices per active trading day. All ApexChart's features are compatible and easy to use on different display sizes, delivering a visually pleasing mobile

design [Fig. 37]. The Material selection dropdown on the top right of the page allows for switching between the primary area chart and the detailed candle stick chart.

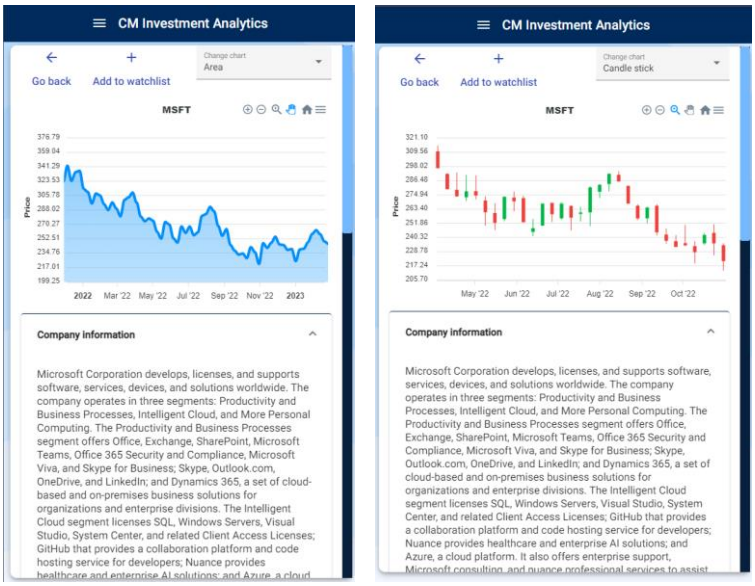



**Figure 37. Both charting views on mobile**

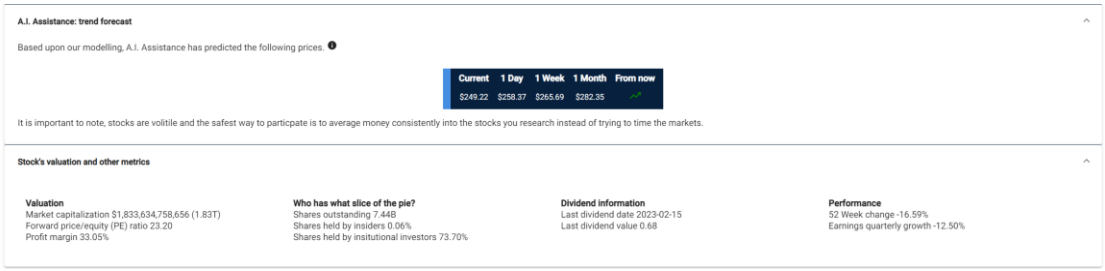



**Figure 38. Forecasted prices and stock valuation metrics**

Below the chart is company information that provides the industry, country of headquarters, number of employees and a business description [Fig. 37]. This is placed within a collapsible section that minimises to just the header, as shown in [Fig. 35]. There are other sections [Fig. 38], one of them contains dynamic time series forecasting results. AI assist displays predicted 1-day, 1-week and 1-month prices along with a colour-indicated chart trend icon to communicate the direction of prices. The other dropdown section contains stock valuations and other financial metrics for more advanced investors, including the percentage of the company held by insiders and retail investors. The stock details page has two buttons positioned at the top left. One adds the stock to the user's watchlist, which can only be used once, as it will return an error message as a popup if pressed twice. The other is a back button that navigates the user back to the search page for easy navigation.

### 4.4.3 News page

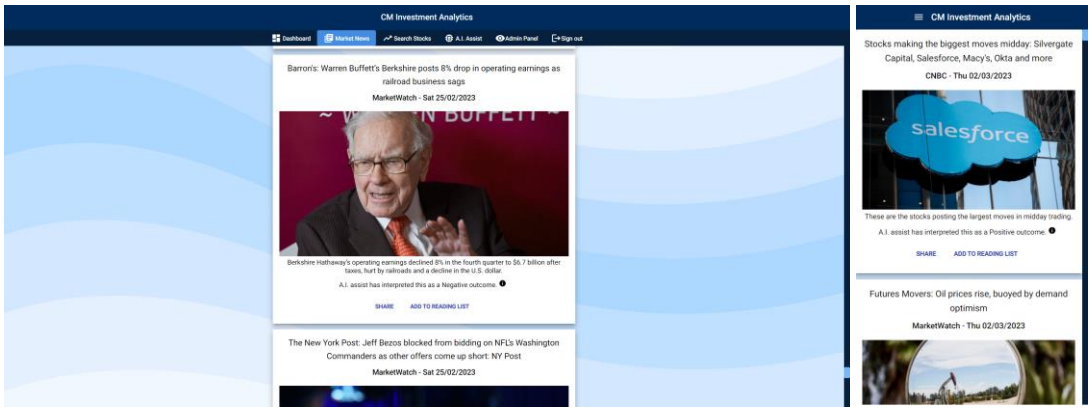



**Figure 39. News page in desktop and mobile view**

The news page [Fig. 39] is designed to be bold, eye-catching, and readable. Associated images are provided for each article; however, custom code excludes repetitive company logos from appearing and consuming more space than needed. When clicked by the user, articles open a new browser tab to the article on the source's domain for further reading. Each article has its predicted sentiment featured below the summary, in which positive or negative sentiment is concluded from the ML.NET sentiment model on the server-side application. All the articles are sorted and displayed in a positive to negative-sentiment order, helping to nurture a healthier mindset with long-term investing. The sentiment classification can be turned off to hide the predictive statement if the user finds it inaccurate. Each article has two buttons at the bottom, a share button to copy the link to the clipboard and an add to the reading list button, which adds the article to the user's dashboard. Most buttons on the website that do not provide navigation functionality provide feedback using Material snack bar popups to confirm to the user that their click was successful.

The news page required optimization due to having a high quantity of images. Ngx-picture [92] is an Angular library that facilitated the usage of appropriately sized images, improved lazy loading and automatic next-generation formatting, which improved page load performance.

### 4.4.4 Dashboard & A.I. assist page

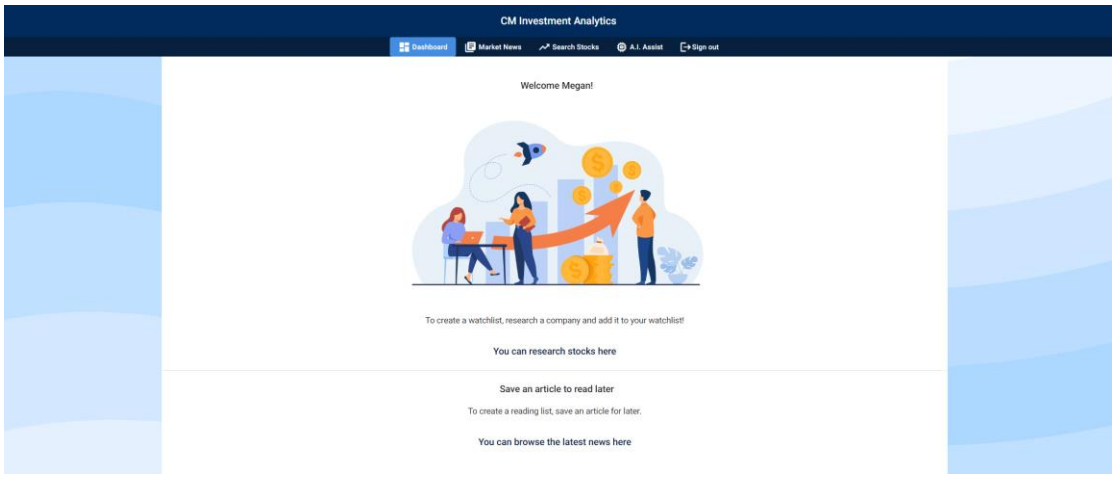



**Figure 40. Empty user dashboard**

The dashboard acts as the user's customisable page. When a new user registers, the page [Fig. 40] displays a welcome message and suggested instructions. The graphic is provided using

Cloudinary [93], a content delivery manager, offering a more performant method of delivering programmable media through an API. According to Cloudinary, they offer a 25% faster page load time on average with responsive image resizing. There are two hyperlinks to the news page and stock search page, with descriptive hyperlink text to improve accessibility. The default underlining of links was removed as it can impair dyslexic users [89].

Upon addition of a saved article or stock to the user's account, the dashboard updates [Fig. 41]. If a news article is the only content added to the user's account, the introduction section for the stock watchlist will remain as there is no stock within the list, and the same applies to the reading list. The design of the articles is an inverse of the news page, blue on a white parent container to stay consistent. The user can click the pen icon positioned right of the watchlist title, which will toggle an edit mode [Fig. 42]. The user can click to remove a listed stock or change the watchlist title if it conforms with product policies such as length validation and profanity checks. Articles on the dashboard do not have an edit mode. However, the "add to reading list" button is replaced with a removal option [Fig. 43]. The design is naturally mobile-friendly [Fig. 44] due to Angular Material's mobile-first design principle.

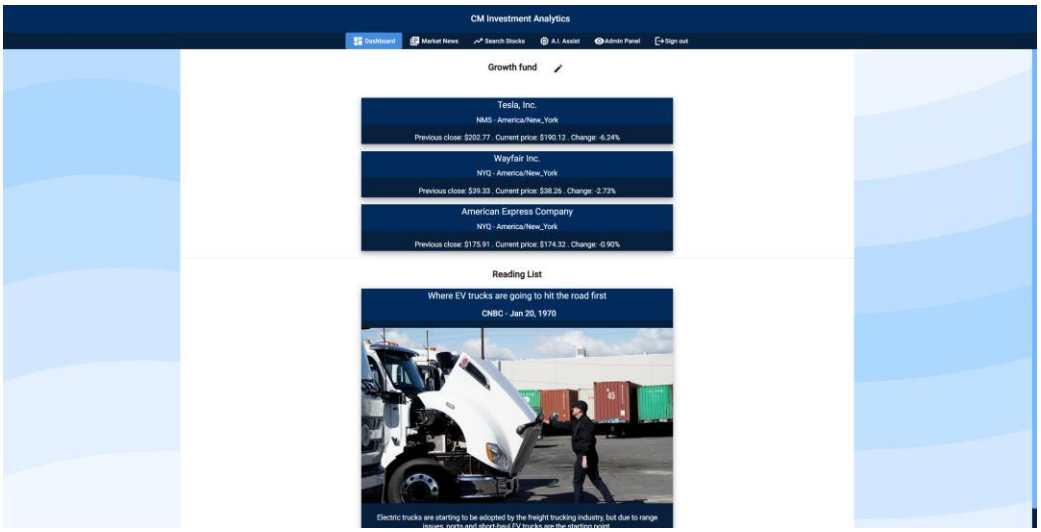



**Figure 41. Populated user dashboard**

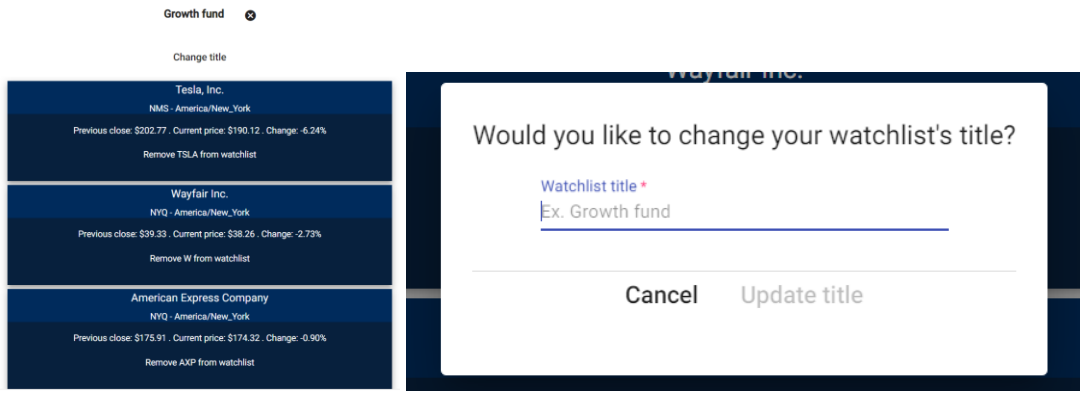



**Figure 42. The edit watchlist view and change title popup**

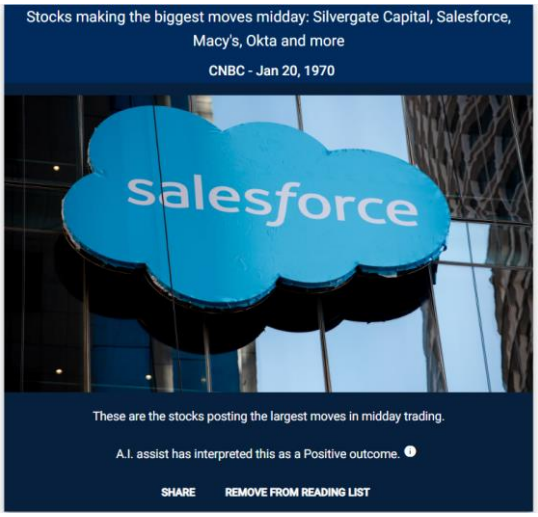



**Figure 43. Saved article showing sentiment analysis classification and removal button**

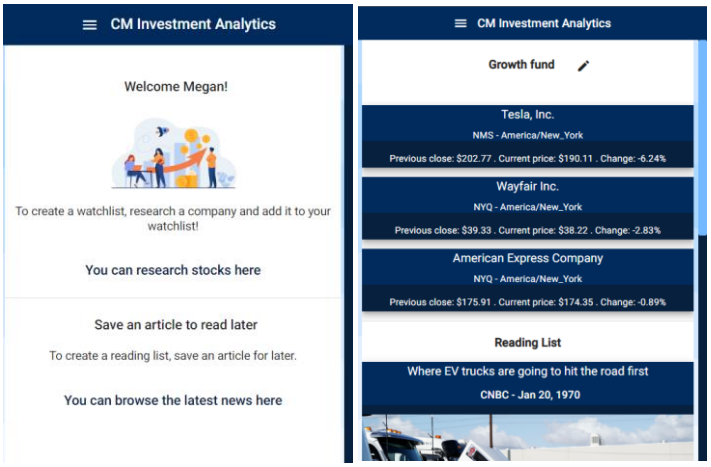



**Figure 44. Mobile view of an empty and populated dashboard**

The AI assist page is within the same module as the dashboard [Fig. 45]. The pages are grouped within the same module as the AI assist page has minimal imports, a minority of which are shared with the dashboard page. The AI assist page allows users to learn what features are on, what they do and even disable them. Changing the AI assistant to on or off is achieved using a radio button that communicates to the server-side application to update the user database table. This presented a significant issue; despite having a rate-limiting integration [Section 3.2.6], the application could still be disrupted as users could spam the radio button to overload the server. Therefore, a red error message appears upon a button toggle and a fifteen-second cooldown to mitigate the risk. Upon the timer's expiry, the button becomes enabled again.

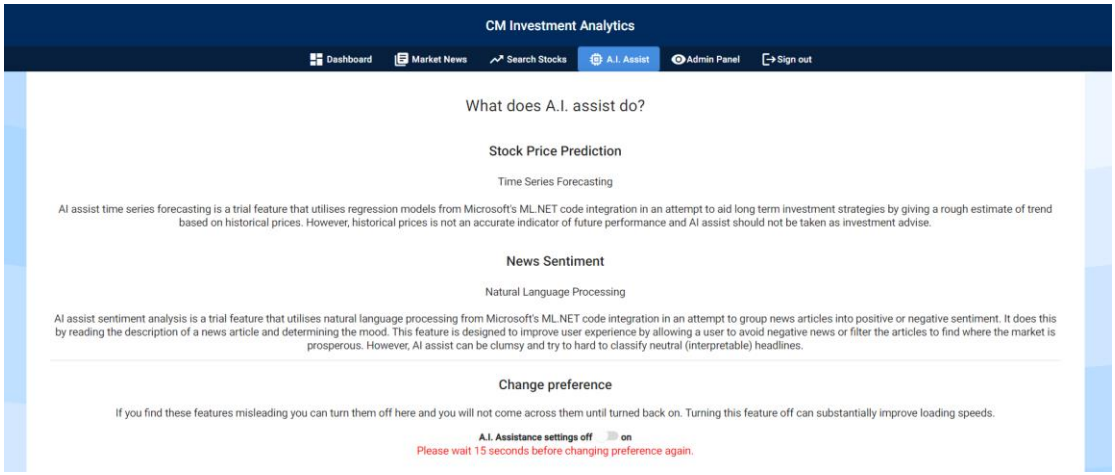



**Figure 45. AI Assist page**

### 4.4.5 Administration page

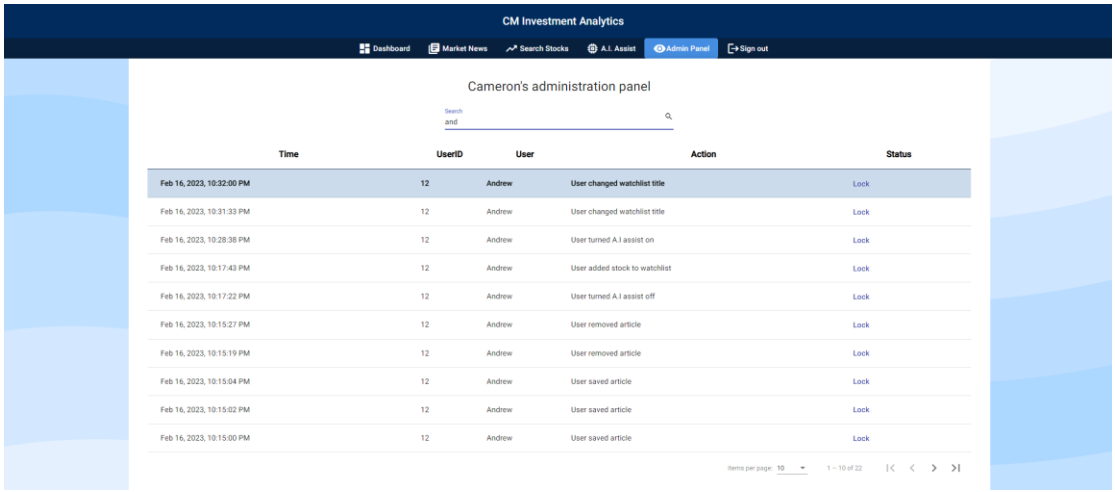



**Figure 46. Administration page featuring search bar and table of user actions**

Administrative users can view audit logs, search for users, actions, dates or identifiers and block accounts from using the application. Providing a management dashboard allows the administrator to view what is consuming the quotas given by data providers or even spot a user trying to maliciously spam actions assuming rate-limiting is not performing its task correctly. Setting up an administration page would additionally allow for greater expansion opportunities and potential monetisation in the long term. Regardless of the column, the administration page's search bar browses the entire table. The table is data heavy, so it requires pagination to limit the results per page to 10, 20 or 30 results [Fig. 46]. Using Angular Material and custom line spacing, the table is acutely readable in mobile or tablet views [Fig. 47]. The administrator's name is positioned at the top of the page to give a more personalised experience, and the JSON web token's payload involving role and username allows for the implementation of the role guard. Only administrators have a navigation button to the administration page. Therefore, the role guard will redirect normal users as if the page does not exist.

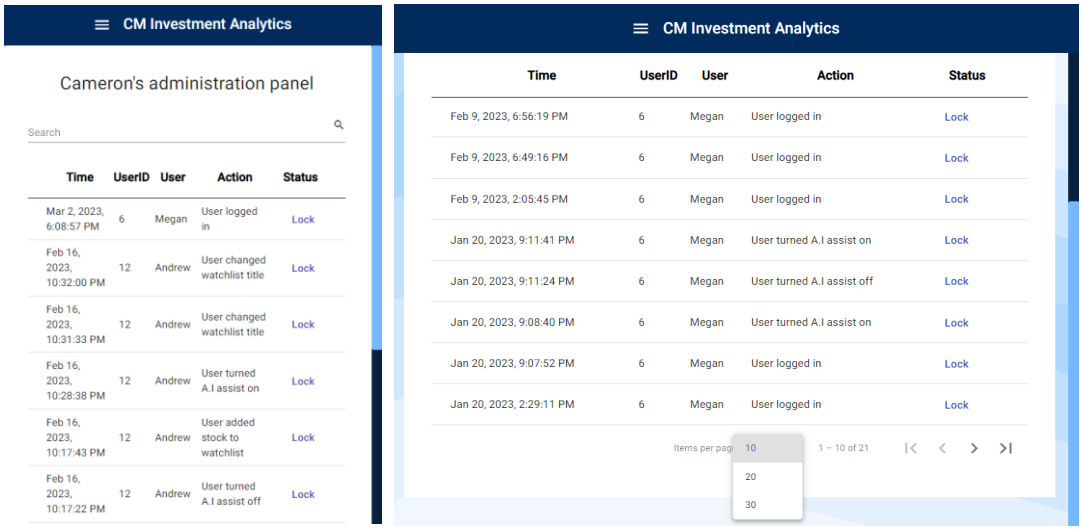



**Figure 47. Mobile and tablet view**

## 4.5 Performance results

As mentioned in section 4.2, the application underwent a major restructuring and reconfiguration to improve the load performance. The initial Lighthouse scores prompted this action. Google Lighthouse [94] is an open-source tool built into Google Chrome. It allows web developers to evaluate and improve their code by reviewing the performance, accessibility, best practices, and Search Engine Optimisation (SEO). The recommendations provided by Google Lighthouse are invaluable in aiding user experience and designing for impaired users.

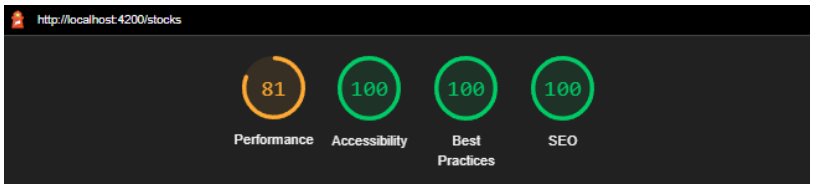



**Figure 48. Stock page Google Lighthouse score example**

The scores from performing Lighthouse tests on various pages from the proposed application and two of its larger competitors have been recorded in a tabular format for evaluation. As the competitor sites do not contain information on all the same pages except the stock lookup pages, two stocks, “Tesla” and “Twilio”, were selected to perform a fair evaluation. As mentioned in section 4.1, Angular’s common criticism is that the SEO optionality is limited. Therefore, the score of 100% obtained for SEO on the proposed solution cannot be compared with competitors as their platform’s SEO structure is not widely known, and their SEO could be page specific or globally identical. In the proposed application, the SEO is defined only once in the parent container as Angular page routing replaces the internal child contents of the parent container, meaning that it is difficult to have individual page SEO.

| Pages | Performance | Accessibility | Best practices | SEO |
|---|---|---|---|---|
| Login | 81 | 100 | 100 | 100 |
| Dashboard (populated) | 78 | 90 | 100 | 100 |
| News | 72 | 100 | 100 | 100 |
| Stocks | 81 | 100 | 100 | 100 |
| AI Assist | 83 | 92 | 100 | 100 |
| Tesla Stock details | 81 | 100 | 100 | 100 |
| Twilio Stock details | 83 | 100 | 100 | 100 |
| Admin | 83 | 100 | 100 | 100 |
| Average | 80.25 % | 97.75 % | 100 % | 100 % |

**Table 4. Proposed solution Lighthouse scores**

| Yahoo Finance | Performance | Accessibility | Best practices | SEO |
|---|---|---|---|---|
| Home | 59 | 91 | 75 | 92 |
| News | 62 | 87 | 75 | 92 |
| Tesla stock | 73 | 86 | 75 | 92 |
| Twilio stock | 71 | 84 | 75 | 92 |
| Average | 66.25 % | 87 % | 75 % | 92 % |

**Table 5. Yahoo Finance Lighthouse scores**

| Google Finance | Performance | Accessibility | Best practices | SEO |
|---|---|---|---|---|
| Home | 98 | 90 | 92 | 83 |
| Watchlist | 96 | 88 | 92 | 83 |
| Tesla stock | 96 | 79 | 92 | 75 |
| Twilio stock | 96 | 83 | 92 | 75 |
| Average | 96.5 % | 85 % | 92 % | 79 % |

**Table 6. Google Finance Lighthouse scores**

The results of performance testing identified strengths in competitors. Google Finance scores highly [Table. 6] in the performance category in comparison to Yahoo Finance [Table. 5] and the proposed solution. Yahoo Finance trails majorly in performance. However, this is due mainly to the number of financial metrics, navigation links, suggested articles, and other content Yahoo Finance displays. Surprisingly Yahoo Finance ranked lower in best practices. Lighthouse flagged Yahoo Finance for using multiple depreciated APIs and logging various error messages to the browser's web console, which slows performance. In terms of designing around simplicity and accessibility, the proper comparison lies between the proposed solution and Google Finance. The solution fits in a middle ground, offering more technical details than Google Finance but remaining more simplistic than Yahoo Finance. This reflects in the page performance score [Table. 4]. In terms of accessibility, the analysis of the platforms in section 2.1.4 was correct. The application has successfully scored higher in accessibility metrics by combining dyslexic and visual impairment design research [Section 2.2 & 4.4] to deliver a more inclusive platform. However, metrics do not outweigh human opinion; therefore, usability testing is in order.

# 5 Usability testing

This section discusses the methodology and methods used for usability testing, the process, and the results. Usability testing improves user experience by providing valuable insights into user behaviour, preferences, and website interaction. Issues identified by users may include complex navigation, confusing labelling, or other factors that affect the efficiency and effectiveness of the solution. The following sections discuss methods, sample composition, strategy, and data collection, contributing to delivering a user-centred and accessible application that can provide positive user experiences.

## 5.1 Mixed methodologies

A mixed-methodology evaluation plan was implemented to gain greater insight from limited user participation. Barnum [95] argues that utilizing quantitative and qualitative usability testing together can offer a better understanding of user experience. Qualitative usability testing is gaining insight by observing a user's interactions and behavioural reactions. Findings are typically based on the researcher's impressions or participants' comments. The flow of dialogue and depth of response increases the time required to conduct research and often reveals solutions to more complex problems hence the name "qualitative". This contrasts with quantitative user feedback, where the responses gained depend on the survey questions. Responses are limited by the researcher's imagination when designing the survey. However, the clear benefit of quantitative user feedback is that it provides measurable statistics, and due to the nature of distribution (i.e., a survey), it is less likely to incorporate bias in the responses as a researcher cannot lead the participants further than the wording of the questions.

For this project, usability testing involved a combination of mixed methodologies to gain the benefits of both approaches.

## 5.2 Mixed methods

Mixed method interviewing and surveying were deployed. The qualitative usability testing was conducted as researcher administered interviews. Ten anonymized participants were individually observed interacting with the proposed application and its free competitors, Yahoo Finance and Google Finance (excluding Bloomberg Terminal for licencing costs identified in section 2.1.1).

Quantitative usability testing is typically done by widespread survey distribution. However, to do this for the proposed application, it would have required it to be hosted, limiting the time to reflect and action changes. So instead, surveys were only distributed to the anonymized participants after the interviews to gain insights and deliver a consistent, unbiased, and statistical measurement of user feedback rather than acquire volumetric data.

## 5.3 Usability testing sample

The sampling strategy was based on convenience sampling [96], a method of recruiting participants that is easily accessible. This approach is time and cost-efficient and often results in a more willing and enthusiastic sample of participants hoping to gain more significant insights. However, the sample may not represent the views of the broader population and may increase bias. Therefore, the composition of the sample [Table. 7] was intentionally diverse as the proposed solution needed to fill a gap in the market. Designing a more inclusive platform means oversampling users from one level of investing experience would debilitate the broader usability. Individuals who were unaware of the project's aim and objectives were approached and selected primarily on age and investing experience.

**Anonymized participant profile**

Disclaimer: Participants consist of family, friends, and students.

| Participant | Age bracket | Dyslexia | Visually impaired | Investing experience |
|---|---|---|---|---|
| 1 | 18-21 | Yes | | None |
| 2 | 60+ | | Mildly colourblind | Experienced |
| 3 | 18-21 | | | Moderate |
| 4 | 18-21 | | | None |
| 5 | 60+ | | | Moderate |
| 6 | 41-60 | Yes | | Experienced |
| 7 | 22-40 | | | Moderate |
| 8 | 41-60 | | Mildly colourblind | Beginner |
| 9 | 18-21 | Yes | | Beginner |
| 10 | 22-40 | | | None |

**Table 7. Anonymized participant information**

## 5.4 Data collection strategy

During the interviews, the participants were asked to be critical of all three platforms and to speak of them comparatively. Participants were told to use dummy data (fake username and password) when registering accounts on the proposed solution. They were given access to one pre-existing account for both competitor services. Participants were occasionally prompted with open-ended questions to start a conversation on features to gain greater insight.

A researcher-administered interview can often introduce interviewer bias, where the response or presence of an interviewer can subconsciously influence the outcome of a participant's response [97]. The steps required to be an effective and unbiased interviewer when conducting an interview include acting professionally and avoiding interactions with the participants that can shift focus [98]. In addition, the interviewer's body language and choice of words can influence bias. Acting thankful for the feedback and criticism is important in helping a participant feel comfortable. Questions should be worded politely, starting with "what" or "how" instead of "why". Questions should additionally be balanced, asking how easy or difficult a task was instead of leading the response (i.e., "Why was that easy?") as it might influence the participant's answer.

The qualitative interview process involved improvisation of questions posed to the participants to gain the most appropriate insight for the situation as users could navigate freely with occasional tasks. The questioning themes were design, navigation, device compatibility [Section 5.5], element positioning, and readability. See [Appendix. 2] for examples of commonality present in each interview's questions.

Ten anonymized surveys were sent to the participants to measure user feedback. The survey was created using Google Forms after the interviews to keep the results anonymized. The follow-up questions posed to the participants can be identified in [Appendix. 1].

## 5.5 Results and user feedback

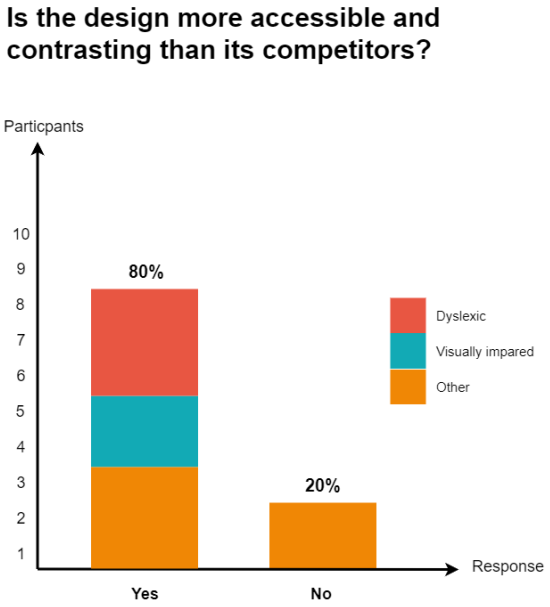
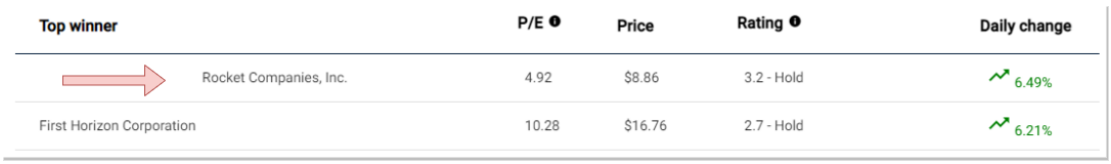



**Figure 49. Bar chart of survey Q5 response**

The results from the survey suggest 80% of participants found the proposed solution more accessible than its competitors [Appendix 1, Q5] [Fig. 49]. Buzzwords to describe the colour theme selected by participants [Appendix 1, Q8] were professional (40%), clear (30%), and calming (10%). Twenty per cent of users stated that the colour theme was basic.

According to a review of methodological and technical aspects of usability testing [99], conducting usability testing on applications intended for mobile viewing optionality is challenging due to the difference in mobile device dimensions [99]. Using Google Inspect [100], the participants were asked to select a mobile device of their choice, which simulated a mobile viewport for different viewing experiences. This is important as greater coverage of dimensions helps identify any potential issues with the design. Ninety per cent of participants selected that the application in mobile view worked seamlessly without interruption [Appendix 1, Q6]. Additionally, 90% of participants found the application easy to navigate as it followed a universal standard [Appendix 1, Q7]. However, qualitative questioning identified the user that did not enjoy navigating the mobile view and took longer than other participants to identify the popout navigation button.

Two users (participants 1, 4) suggested modification to the placement of company names within the top winner stock search page table [Fig. 50] [Section 4.4.2], stating there was wasted space in desktop orientation, and the positioning should be closer to the centre of the screen. This change has been acted upon following similar comments made by participants about spacing during the qualitative interviews [Fig. 51].



| Top winner | P/E | Price | Rating | Daily change |
|---|---|---|---|---|
| Rocket Companies, Inc. | 4.92 | $8.86 | 3.2 - Hold | 6.49% |
| First Horizon Corporation | 10.28 | $16.76 | 2.7 - Hold | 6.21% |

**Figure 50. Proposed change**

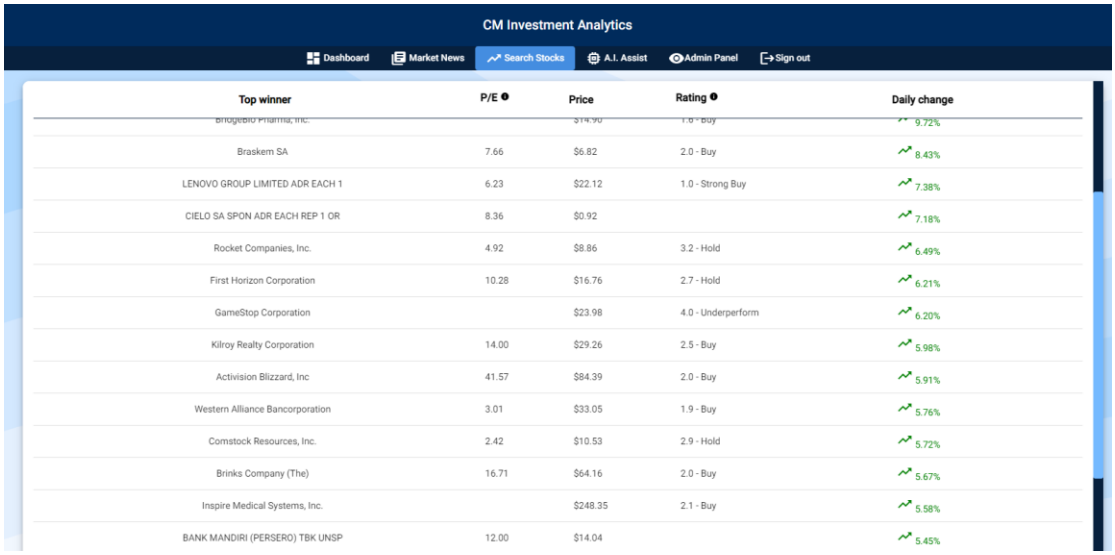



**Figure 51. Implemented change to the company name on the stock search page**

Seventy per cent of participants preferred the ApexChart integration over Google and Yahoo's offerings [Appendix 1, Q10]. Participants 2 and 6, with investing experience, chose Yahoo Finance due to its ability to draw trend lines. When asked [Appendix 1, Q11] if the proposed solution was more fun and less stressful to interact with compared to Yahoo Finance, 70% of participants agreed, while 80% identified Yahoo Finance as the most stressful to use [Appendix 1, Q9].

During observation, participants raised questions about the purpose and design of the sentiment analysis integration. Input from the researcher was needed to explain the dynamic nature of the classifications and how the positive or negative results are not guaranteed to be accurate. Barnum emphasizes the importance of keeping a neutral stance and not overselling features when communicating as an interviewer [98]. Eighty per cent of participants [Appendix 1, Q12] recognized the potential usefulness of the feature for new investors. However, participants 1 and 7 identified the need for further improvement in the model's accuracy.

Quantitative questioning [Appendix 1, Q13] revealed a growing interest and trust in the experimental finance sector, particularly among the 18-21 age group. Three out of ten participants all aged 18-21 claimed they would use time series forecasting for research. However, most participants (70%) expressed a lack of risk appetite in trusting a predictive model. The slow adoption of the latest ideas is often due to resistance to change. The perceived inaccuracy of the feature may be due to short-term stock declines caused by economic challenges and global financial strain identified in section 3.2.7. With a longer-term view and a more accurate model, more users may approve of the feature.

# 6 Conclusion

## 6.1 Summary

The project achieved all the criteria listed in the scope and objectives [Section 1.3]. The application was implemented, including all the outlined code integrations, best practices, and security measures. The application is mobile-compatible and accessible. The accessibility was evaluated through Google Lighthouse scoring [Section 4.5] and extensive usability testing [Section 5], which identified that 80% of interview participants found the design more accessible and colour-contrasting than its competitors listed in section 2. Strengths in design were identified by participants in the charting component, colour theme and ease of navigation. Usability testing suggested that 80% of participants additionally approved of the sentiment analysis integration as they could see how the programmatic interpretation of terminology could benefit new investors. A dynamic and performant time series forecasting model predicts stock prices and displays the result to the user. Further work identified in section 6.3 is required to improve the models continuously.

## 6.2 Evaluation of achievements

Section 1.3 outlined the scope and objectives, which are cross-referenced and evaluated below. All the content outlined was met to its full specification.

1. The project aimed to design and implement an enterprise-level progressive web application for investment research using best practices. Expectations with code integrations discussed in the objectives were fully adopted:

I. Angular Material was used to deliver a mobile-friendly experience [Section 4.4]. Usability testing found that 90% of participants had no issues navigating the mobile application [Section 5.5].

II. Swagger allowed for extensive endpoint testing and helped reduce the development time due to the provision of an interactive user interface early on [Section 3.2.4].

III. Feature flags demonstrated industry-level configuration management by utilising ConfigCat [Section 3.2.3].

IV. Multiple market data providers' services were used to limit costs and split the resource load. Requests were additionally limited through response caching [Section 3.4.1].

V. The application demonstrates robustness through API rate-limiting, helping limit server-side computation costs and blocking malicious users [Section 3.2.6].

VI. Necessary connection strings, application security codes and API keys have been removed from the application's configuration files and instead stored within the dynamic and secure Azure Key Vault secrets management [Section 3.2.1].

VII. Secure communication was implemented using JSON web tokens and a user claim system to identify users using the client-side [Section 3.2.5]. Additionally, passwords are never stored directly within the local database. Instead, passwords are hashed and salted using a unique user string and a HMAC-SHA512 hashing algorithm [Section 3.3.1].

2. The application's use of colours, fonts, styles, and layout was researched following guides from the World Wide Web Consortium, the British Dyslexia Association and UK

Government. A combination of quantitative and qualitative usability testing methods [Section 5] demonstrated the application's ability to appeal to a broader audience through accessibility and mobile compatibility. Google Lighthouse scoring [Section 4.5] additionally verified accessibility claims through scanning that identified 100% accessibility scores on all pages, suggesting there were no apparent improvements in code that the service could detect:

I. The application's dynamic content offers interactive charts with two viewing options: area or candlestick. In addition, stock quotes and price history used to populate the charting component are generated in real time.

II. Articles used to populate the market news page seen [Section 4.4.3] offers daily headlines. In addition, the application contains a table of daily top winners, stocks that outperform the market average and a search bar to browse equities using either the company name or ticker symbol [Section 4.4.2].

3. Artificial intelligence was present within the application [Section 3.2.7], and its popularity with users was evaluated through usability testing:

I. The sentiment analysis was well received. Participants were not informed of the automatic filtering applied to news articles to display positive sentiment articles before negative ones. However, 70% of participants agreed the proposed solution was less stressful than its competitors. Additionally, 80% of participants recognised the advantage a sentiment summary on each article could offer new investors.

II. Time series forecasting was integrated into the application to predict future prices based on price history dynamically. The outlined objective was achieved. However, there is room for improvement in accuracy. Participants did not well receive this feature due to the perceived inaccuracies and individual lack of risk appetite. This could be improved with further work on the model's accuracy and further beta testing.

4. The application required registration functionality to allow for user-specific content. This was met to the full extent and designed with an emphasis on security [Section 3.3.1].

5. An administration panel to monitor users' actions was proposed and delivered [Section 4.4.5]. This required tables and additional code to facilitate the logging of user actions. The administration panel is secure, only accessible to administrators, searchable, and facilitates banning malicious accounts.

6. Usability testing conducted mixed method research to gather unbiased user feedback. The insights and modifications identified during the process can be found in section 5.5.

## 6.3 Limitations and Future Work

The transition of the application from development to production requires three main steps:

1. Place the applications within containers to run on any platform seamlessly. This can be achieved using Docker [101], an open-sourced platform used to standardize executable components by combining code with operating system libraries needed to operate on any environment. Docker simplifies the deployment pipeline, cuts costs and mitigates compilation errors by acting similarly to a virtual machine. However, unlike the traditional definition, they are minimal in size and can be managed, shut down and started up quickly. Docker build and compose files have already been created. However, the Azure Key Vault connection parameters must be set as environment variables, and the containers need permission to communicate with the MySQL server. Additionally, the existing solution contains a health check on the server-side API. Container orchestrators typically use health checks to monitor the status of an application and are beneficial for debugging and monitoring a container.
2. Purchase a domain for the client-side application. Domain names can be purchased and set up from an online provider. The Domain Name System (DNS) needs to be set up to point to the hosting platform's IP address; this is how the DNS resolver establishes a connection.
3. Find a suitable hosting platform, such as Azure or AWS, to host the virtual container. An SSL certificate can be acquired to notify users the website is secure, granting HTTPS connections, which is frequently implemented through the hosting service's content management system.

The development requirements have been met, and the application has undergone production optimization and performance testing, resulting in a total expenditure of $19.9 from September 2022 to April 2023, thanks to cost-cutting measures such as caching responses and splitting resource usage between external services. Future expenses include $9.95 per month for market data, leading to a net loss. Therefore, monetizing the application through Google AdSense should be considered to keep the platform cost-free for users. Assessing the financial viability of AdSense and demand for the application would require it to be hosted. The application should be released in phases to live users via financial institutions, investment organizations, and various samples of users to gather disaggregated data for analysis.

The application could benefit from a content delivery management service in future development, allowing for snapshots of charted daily stock movement. For example, miniature charts appearing alongside each top mover on the stock search page shown in section 4.4.2 would improve design and user experience. However, using ApexCharts to deliver all chart snapshots on the same page would be too computationally expensive in API usage and could cause crashing in mobile devices.

Additionally, further development of both models would be recommended. For example, the sentiment analysis model could benefit from greater accuracy. This would require a more extensive training dataset. Replacing the time series forecasting model with a Long Short-Term Memory (LSTM) model as Microsoft expands its ML.NET offerings, as discussed in section 2.4, would benefit the user the most.

## 6.4 Critical Reflection

Overall, the project successfully managed its development timeline, met its aim and objectives, and usability testing improved its effectiveness. Consistent hours spent daily on the project, assisted by defining a realistic project scope and identifying limitations early, aided in delivering the prototype on time. Using Industry-recognized technologies, including Swagger, Azure Key Vault, and ConfigCat, helped ensure the deliverables created an enterprise-level application.

However, the development process overallocated time to the charting component, animations, and search bar, although these were critical for usability and were well-received in testing. ML.NET was disappointing due to time-consuming troubleshooting and the limited documentation and support from the .NET ecosystem compared to Python. Inconsistent method calls between versions added to the difficulty of learning the framework. Proof-of-concept speeds for the time series forecasting model were impressive, but the accuracy of the temporary model was poor. Geopolitical tensions and negative market sentiment limited the accuracy during observations, resulting in a mediocre performance, as predicted in section 2.4. Fortunately, the regression model provides the foundations to expand and use more complex models as Microsoft expands its native support for deep learning methods [37].

Regarding learning, performance optimization for Angular was an important lesson to take away from this project. The client-side application performed slowly without optimised modularity, production settings and lazy loading. However, the improvements worked, and the restructuring was quick and effective after precise planning. In conclusion, continuous learning to keep applications efficient, effective, and scalable is vital in delivering a robust application.

# References


[1] C.-O. Cepoi, “Asymmetric dependence between stock market returns and news during COVID-19 financial turmoil,” Finance research letters, Oct-2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7299872/

[2] T. I. Team, “How do recessions impact investors?” Investopedia, 07-Feb-2023. [Online]. Available: https://www.investopedia.com/insights/recession-what-does-it-mean-investors/

[3] Google search for “Stock market”, source of headlines identified in [Fig. 2]. [Online]. Available: https://www.google.com/search?q=stock+market

[4] C. C. Wu et al., “A study of network negative news based on behavioural finance analysis of abnormal fluctuation of stock price,” Discrete Dynamics in Nature and Society, 28-Jan-2022. [Online]. Available: https://www.hindawi.com/journals/ddns/2022/7952532/

[5] C.-L. Lin et al., “Do stock prices drive people crazy,” vol. 30, no. 2, pp. 206–214, Mar. 2015, doi: 10.1093/heapol/czu007. [Online]. Available: https://doi.org/10.1093/heapol/czu007

[6] A. AL-Ayash et al., “The influence of color on student emotion, heart rate, and performance in learning environments,” vol. 41, no. 2, pp. 196–205, 2016, doi: 10.1002/col.21949. [Online]. Available: https://doi.org/10.1002/col.21949

[7] H. Qiang, “A study on the metaphor of ‘red’ in Chinese culture,” vol. 1, no. 3, Nov-2011. [Online]. Available: https://www.aijcrnet.com/journals/Vol_1_No_3_November_2011/13.pdf

[8] W. Bazley et al., “Visual finance: The pervasive effects of red on investor behavior,” SSRN, 27-Jun-2017. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2992812

[9] “AMD Interactive Stock Chart | Advanced Micro Devices, Inc... stock,” Yahoo Finance. [Online]. Available: https://finance.yahoo.com/chart/AMD

[10] “Stock market declines and mental health - financial anxiety,” The Dunes East Hampton, 14-Dec-2022. [Online]. Available: https://theduneseasthampton.com/stock-market-declines-and-mental-health/

[11] S. Mishra, “Artificial Intelligence in the stock market: How did it happen?” Artificial intelligence in the stock market: how did it happen? | FIU Business. [Online]. Available: https://business.fiu.edu/graduate/insights/artificial-intelligence-in-the-stock-market.cfm

[12] “API Documentation & Design Tools for Teams | Swagger.” SMARTBEAR. [Online]. Available: https://swagger.io/

[13] “ConfigCat Feature Flag Service.” ConfigCat. [Online]. Available: https://configcat.com

[14] “Bloomberg Terminal.” Bloomberg. [Online]. Available: https://www.bloomberg.com/professional/solution/bloomberg-terminal/

[15] J. Martel et al., “The Best Bloomberg Terminal Alternatives of 2023,” 11-Apr-2023. [Online]. Available: https://www.investopedia.com/best-bloomberg-terminal-alternatives-5114517

[16] “Consistency: More than just a buzzword,” Bloomberg, 2020. [Online]. Available: https://www.bloomberg.com/ux/2020/08/11/consistency-more-than-just-a-buzzword/

[17] “Yahoo Finance – stock market live, quotes, business & finance news.”, Yahoo Finance. https://uk.finance.yahoo.com/

[18] D. Stone, “How to spot fake comments on stock boards,” GripRoom, 06-Jan-2023. [Online]. Available: https://www.griproom.com/fun/how-to-spot-fake-comments-on-stock-boards

[19] “5 tips for a perfect website navigation menu: Jimdo,” The Jimdo blog for small businesses, 18-Jun-2021. [Online]. Available: https://www.jimdo.com/blog/fixing-website-navigation-bar/

[20] “Google Finance - stock market prices, real-time Quotes & Business News,” Google. [Online]. Available: https://www.google.com/finance/

[21] “Coblis — Color Blindness Simulator – Colblindor.” [Online]. Available: https://www.color-blindness.com/coblis-color-blindness-simulator/

[22] “Financial Products UX.” [Online]. Available: https://www.bloomberg.com/ux/accessibility/

[23] “Making your service accessible: an introduction - Service Manual - GOV.UK.” [Online]. Available: https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction

[24] S. Schmutz et al., “Easy-to-read language in disability-friendly web sites: Effects on nondisabled users,” vol. 74, pp. 97–106, Jan. 2019, doi: 10.1016/j.apergo.2018.08.013

[25] F. F. Reichheld and P. Schefter, “The economics of E-loyalty,” HBS Working Knowledge, 07-Oct-2000. [Online]. Available: https://hbswk.hbs.edu/archive/the-economics-of-e-loyalty

[26] “The Importance of Accessible Web Applications,” 2020. [Online]. Available: https://www.nearform.com/blog/accessible-web-applications-overview/

[27] S. Aquino, “When it comes to accessibility, Apple continues to lead in awareness and Innovation,” TechCrunch, 19-May-2016. [Online]. Available: https://techcrunch.com/2016/05/19/when-it-comes-to-accessibility-apple-continues-to-lead-in-awareness-and-innovation/

[28] “Understanding SC 1.4.3:contrast (minimum) (level AA),” Understanding Success Criterion 1.4.3: Contrast (Minimum) | WAI | W3C, 24-Dec-2022. [Online]. Available: https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html

[29] British Dyslexia Association, “Dyslexia friendly style guide,” British Dyslexia Association, 2023. [Online]. Available: https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide

[30] “Using machine learning for sentiment analysis: A deep dive,” DataRobot AI Platform, 07-Mar-2019. [Online]. Available: https://www.datarobot.com/blog/using-machine-learning-for-sentiment-analysis-a-deep-dive

[31] M. Sykora et al., “A qualitative analysis of sarcasm, irony and related #hashtags on Twitter,” vol. 7, no. 2, p. 2053951720972735, Jul. 2020, doi: 10.1177/2053951720972735. [Online]. Available: https://doi.org/10.1177/2053951720972735

[32] B. Yılmaz, “Stock Market Sentiment Analysis in 2023,” 03-Apr-2023. [Online]. Available: https://research.aimultiple.com/sentiment-analysis-stock-market/

[33] K. Rapoza, “Can ‘Fake News’ Impact the Stock Market?” 26-Feb-2017. [Online]. Available: https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/

[34] R. Gupta and M. Chen, “Sentiment Analysis for Stock Price Prediction,” 2020, pp. 213–218, Doi: 10.1109/MIPR49039.2020.00051

[35] Prophet by Meta, “Forecasting at scale”. [Online]. Available: https://facebook.github.io/prophet/

[36] S. L. PhD, “Time-Series Forecasting: Predicting Stock Prices Using Facebook’s Prophet Model,” 2023. [Online]. Available: https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-facebooks-prophet-model-9ee1657132b5

[37] Microsoft, “Support for LSTM / RNN (time series) · issue #149 · dotnet/machinelearning,” GitHub, 14-May-2018. [Online]. Available: https://github.com/dotnet/machinelearning/issues/149

[38] “The Microsoft cognitive toolkit - cognitive toolkit - CNTK,” Cognitive Toolkit - CNTK | Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/cognitive-toolkit/

[39] “Create production-grade machine learning models with TensorFlow” TensorFlow. [Online]. Available: https://www.tensorflow.org/

[40] “What’s new in .NET 6,” 2022. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6

[41] A. Maciejek, “When .NET Framework is the Best Choice for App Development and Why?” 2021. [Online]. Available: https://cshark.com/blog/when-net-framework-is-the-best-choice-for-application-development-and-why/

[42] “API attack | Radware.” [Online]. Available: https://www.radware.com/cyberpedia/application-security/api-attack/

[43] “Key Vault | Microsoft Azure.” [Online]. Available: https://azure.microsoft.com/en-gb/products/key-vault

[43] “Entity Framework documentation.” [Online]. Available: https://learn.microsoft.com/en-us/ef/

[44] “Tutorial: Implement CRUD Functionality with the Entity Framework in ASP.NET MVC,” 2022. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/implementing-basic-crud-functionality-with-the-entity-framework-in-asp-net-mvc-application

[45] “Feature Flagging.” [Online]. Available: https://www.atlassian.com/software-development/practices/feature-flagging

[46] “What is Swagger and Why Does it Matter?” 2016. [Online]. Available: https://blog.readme.com/what-is-swagger-and-why-it-matters/

[47] Ankush, “Why and How to Secure API Endpoint?” 2018. [Online]. Available: https://geekflare.com/securing-api-endpoint/

[48] “How to perform JSON web token (JWT) authentication - 42 interactive,” How to Perform Json Web Token (JWT) Authentication - 42 Interactive, 22-Jul-2019. [Online]. Available: https://www.42interactive.com/blog/json-web-token-jwt-authentication-a-how-to-guide

[49] “Denial of service (DOS) guidance,” National Cyber Security Centre (NCSC). [Online]. Available: https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection

[50] R. Simon, “Defending Against Web Scraping Attacks,” 2021. [Online]. Available: https://www.darkreading.com/endpoint/defending-against-web-scraping-attacks/a/d-id/1340846

[51] “Welcome to Python.org.” [Online]. Available: https://www.python.org/

[52] “How to design a rate limiter,” 2021. [Online]. Available: https://devsday.ru/blog/details/59853

[53] “ML.NET | Machine learning made for .NET.” [Online]. Available: https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet

[54] “What is Model Builder and how does it work? - ML.NET,” 2022. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-model-builder

[55] Y. Chaudhary, “Stock-market sentiment dataset,” Kaggle, 05-Jun-2020. [Online]. Available: https://www.kaggle.com/datasets/yash612/stockmarket-sentiment-dataset

[56] N. Golyandina and A. Korobeynikov, “Basic Singular Spectrum Analysis and forecasting with R,” vol. 71, pp. 934–954, 2014, doi: 10.1016/j.csda.2013.04.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947313001394

[57] “Singular spectrum analysis,” Wikipedia, 08-Dec-2022. [Online]. Available: https://en.wikipedia.org/wiki/Singular_spectrum_analysis

[58] J. Tapsfield, “Bank of England set to hike interest rates again this month,” 08-Mar-2023. [Online]. Available: https://www.dailymail.co.uk/news/article-11834533/Bank-England-certain-hike-rates-month.html

[59] N. Sherman, “US interest rates could go higher than expected,” BBC News, 07-Mar-2023. [Online]. Available: https://www.bbc.co.uk/news/business-64881469

[60] J. Barrett, “Silicon Valley Bank: Why did it collapse and is this the start of a banking crisis?” The Guardian, 13-Mar-2023. [Online]. Available: https://www.theguardian.com/business/2023/mar/13/silicon-valley-bank-why-did-it-collapse-and-is-this-the-start-of-a-banking-crisis

[61] Z. Ma, “Ackman Says More Banks Will Likely Fail Despite Fed Intervention,” 13-Mar-2023 [Online]. Available: https://www.bloomberg.com/news/articles/2023-03-13/ackman-says-more-banks-will-likely-fail-despite-fed-intervention

[62] E. Nolsøe et al., “Lagarde vows to protect banking system from turmoil after $54bn lifeline for Credit Suisse,” 16-Mar-2023 [Online]. Available: https://www.telegraph.co.uk/business/2023/03/16/credit-suisse-news-latest-ftse-100-markets-live/

[63] J. Borger, “Russia plans to recover wreckage of US drone downed over Black Sea,” The Guardian, 15-Mar-2023. [Online]. Available: https://www.theguardian.com/us-news/2023/mar/15/russian-us-drone-jet-recovery

[64] “Time Series Forecasting: Definition, Applications, and Examples.” [Online]. Available: https://www.tableau.com/learn/articles/time-series-forecasting

[65] J. Galloway, “Part 6: Using Data Annotations for Model Validation,” 2023. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions/mvc-music-store/mvc-music-store-part-6

[66] “HMAC,” 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=HMAC

[67] “Why is sha512 hash considered secure?” Quora. [Online]. Available: https://www.quora.com/Why-is-SHA512-hash-considered-secure

[68] “Yahoo Finance API Documentation” RapidAPI. [Online]. Available: https://rapidapi.com/sparior/api/yahoo-finance15

[69] “Finnhub - Free realtime APIs for stock, forex and cryptocurrency.” [Online]. Available: https://finnhub.io/

[70] “Polygon.io - Stock Market Data APIs.” [Online]. Available: https://polygon.io

[71] Market Ahead, “US Stocks & etfs - tickers, company info, logos,” Kaggle, 06-Feb-2019. [Online]. Available: https://www.kaggle.com/datasets/marketahead/all-us-stocks-tickers-company-info-logos

[72] S. Haunts, “ProfanityDetector: This is a simple library for detecting profanities within a text string.” GitHub. [Online]. Available: https://github.com/stephenhaunts/ProfanityDetector

[73] Angular, “Deliver web apps with confidence” [Online]. Available: https://angular.io/

[74] A. Shybeko, “8 Benefits of Angular for Your Project.” [Online]. Available: https://light-it.net/blog/8-advantages-of-angular-for-businesses-and-developers/

[75] C. Lüdemann, “The Complete Guide to Angular Performance Tuning,” Christian Lüdemann, 06-Aug-2019. [Online]. Available: https://christianlydemann.com/the-complete-guide-to-angular-performance-tuning/

[76] M. Gechev, “angular-performance-checklist: Cheat sheet for developing lightning-fast progressive angular applications,” GitHub, 07-Aug-2022. [Online]. Available: https://github.com/mgechev/angular-performance-checklist

[77] Google, “Angular material,” Angular Material. [Online]. Available: https://material.angular.io/

[78] “Angular Powered Bootstrap,” Angular powered Bootstrap. [Online]. Available: https://ng-bootstrap.github.io/

[79] J. Patel, “Angular material vs bootstrap: Which is best to use & why?” Monocubed, 21-Mar-2022. [Online]. Available: https://www.monocubed.com/blog/angular-material-vs-bootstrap/

[80] W. Jordan, “Why is blue the world's favorite color?” YouGov, 12-May-2015. [Online]. Available: https://today.yougov.com/topics/international/articles-reports/2015/05/12/why-blue-worlds-favorite-color

[81] “Project stability with stately navy blue - adobe,” Adobe creative cloud. [Online]. Available: https://www.adobe.com/creativecloud/design/hub/guides/project-stability-with-navy-blue

[82] K. Cherry, “How does the color red impact your mood and behavior?” Verywell Mind, 16-Mar-2023. [Online]. Available: https://www.verywellmind.com/the-color-psychology-of-red-2795821

[83] R. Rodin, “A graphic designer's Guide to Visual Hierarchy,” ZevenDesign, 23-Sep-2016. [Online]. Available: https://zevendesign.com/designers-guide-to-visual-hierarchy/

[84] R. Rathee and P. Rajain, “Role Colour Plays in Influencing Consumer Behaviour,” vol. 12, pp. 209–222, Dec. 2019, doi: 10.21632/irjbs.12.3.209-222

[85] “Colour contrast for visual stress and why it's important to optimise it.” ScreenRisk, 06-Jan-2022. [Online]. Available: https://www.screenrisk.com/blog/colour-contrast-visual-stress-important-to-optimise-it/

[86] “Color contrast Checker - Coolors,” Coolors.co. [Online]. Available: https://coolors.co/contrast-checker/ffffff-07203d

[87] C. Robertson, “Roboto,” Google Fonts. [Online]. Available: https://fonts.google.com/specimen/Roboto

[88] “Accessible communication formats,” GOV.UK. [Online]. Available: https://www.gov.uk/government/publications/inclusive-communication/accessible-communication-formats#:~:text=A%20minimum%20size%20of%2016,size%20is%20suitable%20for%20everyone

[89] “Dyslexia friendly style guide,” British Dyslexia Association. [Online]. Available: https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide

[90] D. Dudeja, “Horizontal vs vertical website navigation menu: Which one should I choose?” LinkedIn, 08-Dec-2021. [Online]. Available: https://www.linkedin.com/pulse/horizontal-vs-vertical-website-navigation-menu-which-one-dudeja/

[91] A. Harley, “Icon usability,” Nielsen Norman Group, 27-Jul-2014. [Online]. Available: https://www.nngroup.com/articles/icon-usability/

[92] “Ngx-picture,” npm, Nov-2022. [Online]. Available: https://www.npmjs.com/package/ngx-picture

[93] “Programmable media experience,” Cloudinary, 14-Feb-2023. [Online]. Available: https://cloudinary.com/products/programmable_media

[94] “Lighthouse overview,” Chrome Developers, 27-Sep-2016. [Online]. Available: https://developer.chrome.com/docs/lighthouse/overview/

[95] C. M. Barnum, “Chapter 5: Planning for usability testing,” in Usability Testing Essentials: Ready, set... test !, Amsterdam: Elsevier, 2021, p. 136

[96] J. Dudovskiy, “Convenience Sampling - Research methodology,” Business research methodology. [Online]. Available: https://research-methodology.net/sampling-in-primary-data-collection/convenience-sampling/

[97] M. K. Salazar, “Interviewer Bias: How it Affects Survey Research,” vol. 38, no. 12, pp. 567–572, Dec. 1990, doi: 10.1177/216507999003801203. [Online]. Available: https://doi.org/10.1177/216507999003801203

[98] C. M. Barnum, “Chapter 7: Conducting a usability test,” in Usability Testing Essentials: Ready, set... test !, Amsterdam: Elsevier, 2021, p. 207

[99] J. M. C. Bastien, “Usability testing: a review of some methodological and technical aspects of the method,” vol. 79, no. 4, pp. e18–e23, 2010, doi: 10.1016/j.ijmedinf.2008.12.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386505608002098

[100] K. Basques and S. Emelianova, “Simulate mobile devices with Device Mode,” 13-Apr-2015. [Online]. Available: https://developer.chrome.com/docs/devtools/device-mode/

[101] “Docker: Accelerated, Containerized Application Development,” 2022. [Online]. Available: https://www.docker.com/

# Appendix 1 – Quantitative survey questions and answers

Quantitative questions were asked to all participants during the researcher-administered interview process to deliver measurable statistics. Answers from the survey are represented as percentages presented after the answering options.

1. What is your age range?
    I. 18-21 (40%)
    II. 22-40 (20%)
    III. 41-60 (20%)
    IV. Over 60 (20%)
2. Are you colour blind, and if so, to what extent?
    I. This does not apply to me (80%)
    II. Mildly (20%)
    III. Moderately
    IV. Severely
3. Are you dyslexic?
    I. Yes (30%)
    II. No
4. What is your investing experience?
    I. None (30%)
    II. Beginner (20%)
    III. Moderate (30%)
    IV. Experienced (20%)
5. In terms of accessibility and contrast of colour, do you believe the proposed platform to be easier than other investment research platforms to use?
    I. Yes (80%)
    II. No (20%)
6. Do you like the proposed solution on mobile? Does everything work as intended?
    I. Yes, it works well (90%)
    II. No, I found issues (10%)
7. Did you have difficulties navigating the proposed solution?
    I. Yes, I did not expect navigation links to be placed where they were (10%)
    II. No, it followed a universal standard (90%)
    III. It was not as easy as its competitors
8. Do any of these words describe what you thought about the colour theme? Pick the most prominent answer.
    I. Professional (40%)

II. Basic (20%)
III. Tacky
IV. Calming (10%)
V. Clear (30%)
VI. Stressful
VII. None listed here

9. Which of the three platforms was the most stressful to use?
    I. Proposed solution (10%)
    II. Google Finance (10%)
    III. Yahoo Finance (80%)
10. Which charting view do you prefer?
    I. Yahoo Finance (20%)
    II. Proposed solution (70%)
    III. Google Finance (10%)
11. Compared to Yahoo Finance, do you believe the proposed solution's news page will be more fun and less stressful?
    I. Yes (70%)
    II. No (30%)
    III. Unsure
12. How do you feel about sentiment analysis interpreting headlines?
    I. I can see why this would be useful to inexperienced users (80%)
    II. I do not see the point
    III. It seems inaccurate (20%)
13. How do you feel about time series forecasting (stock price predictions)? What quote best describes how you think?
    I. I like that they are a feature. I would use it for research (30%)
    II. I like that it is a feature, but I would not use it personally
    III. Finance is a sensitive area. I do not have that kind of risk appetite (70%)

# Appendix 2 – Qualitative interview questions

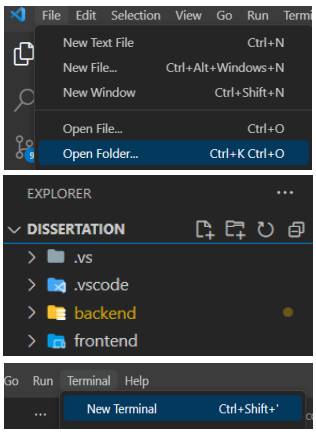
General questioning themes were explored in the situational interview. However, here are example questions of commonality posed to all users. Notice how the questions attempt to mitigate the risk of leading the response by balancing, as identified in section 5.4.

1. How easy or difficult is it to navigate the website?
2. Are there any features you like or dislike about the application? Any input would be appreciated.
3. How difficult or easy was registering/logging into the application?
4. Would you say the search bar interprets your input well or poorly? For example, did you get the company you searched for?
5. How do you feel about the layout of the application?
6. How difficult or easy was navigating the application in the selected mobile view?
7. Have you identified any issues with the application?
8. Would you say it is easy or difficult to read the text on the application?
9. Do the colours contrast enough?
10. Can you navigate the stock details page and change the charting view to candle stick?
    a. How easy or difficult was this task?
    b. What do you think of the interactive chart?
11. Please try your best to cause a bug within the application, whether entering special characters into forms or spam-clicking actions. Did you identify any potential errors?
12. Do you like the positioning of the button located at the top left of the stock search page?
13. Do buttons and form elements provide good feedback on click or hover? How could they be improved?

# Appendix 3 – Installation Guide

An Integrated Development Environment (IDE) software is required to run the code. The recommended application used for this project is Visual Studio Code, a lightweight and highly customisable editor. To download it, see the link.
https://code.visualstudio.com/
The Material Icon theme is recommended to help with the project's navigation, as it makes the distinct types of files stand out from each other. This is an extension for Visual Studio code, and you can find a quick install tutorial here.
https://youtu.be/XBO4rt3aKFE?t=37
To run the Node Package Manager (NPM), which is responsible for installing dependencies and running the front-end Angular application, you will need the Node JavaScript run time, including NPM.
https://nodejs.org/en/download
Next, you must install the .NET six run time and SDK. Here is a YouTube tutorial which demonstrates the steps required.
https://www.youtube.com/watch?v=AC5UWby16sg
You will need to install the MySQL server to install the environment for Entity Framework to scaffold a database. When running the MySQL installer, click “Developer default, " which only downloads the client command line. For ease of installation, set the port number to 3306 and management user to root if it is not already as default.
https://dev.mysql.com/downloads/installer/

Now, unzip the project folder and, using Visual Studio code at the top left corner, click “File” -> “Open folder”.




You should then see the following file structure. The backend contains the .NET server-side application, and Frontend contains the Angular client-side application.

Now you have opened the project, go to the top navigation bar of the editor, and click “Terminal” followed by “New Terminal”.
At the bottom of your editor, you should now see a command prompt terminal, click on it and type:

```
cd frontend
```

This will take you to the frontend folder where we will set up Angular, check its working and install dependencies for the project.
First, check that Node Package Manager has been installed correctly by running the following command. If a version is returned, then it has worked. If not, revisit the steps or search for the error code online.

```
npm --version
```

Now run this command to install Angular.

```
npm install -g @angular/cli
```

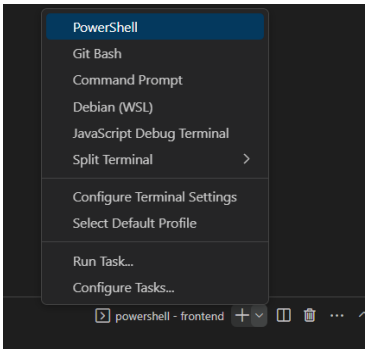
Now, run the following command to install dependencies. One dependency for injection may not be included in the install by default, so run the second command separately.

```
npm i
npm i ngx-window-token
```

Finally, after the dependencies are installed, start the front end by running this command.

```
npm start
```

The front end should now be running on the local device port 4200. Before navigating to the local host website. Start up the backend to give it the functionality. Create a new PowerShell window at the top right of the active terminal, as seen in the image.




Run the following commands individually to navigate to the server-side .NET application's directory, then check the version of the .NET install. If a number is not returned, there is an installation issue. The first number should be six. Finally, build the .NET application.

```
cd backend
dotnet --version
dotnet build backend.csproj
```

Now run the following commands one at a time. This will install the entity framework command for .NET and then run the database update command to create the tables.

```
dotnet tool install –-global dotnet-ef
dotnet ef migrations add setup --context backend.entity.dbcontext
dotnet ef database update --context backend.entity.dbcontext
dotnet run
```

You should now be able to access both the backend and frontend applications.
Angular app: http://localhost:4200/
Swagger user interface: https://localhost:7299/swagger/index.html
The Swagger endpoints will not work unless you register an account (or log in if you have an existing one) and copy the JWT response into the authorization popup located at the top right of the page with the word bearer, i.e., "bearer [TOKEN]".
Any accounts generated on the newly installed application version will be users by default. Creating an administrator account will require custom MySQL statements in the command line application.