The University of Nottingham | School of Computer Science | MSc Computer Science

StockVis: Visualisation of Investment Transactions

Submitted in September 2023, in partial fulfilment of the degree MSc Computer Science.

–

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Mingrui Li – 20468731

Submitted Date: 08/09/2023

Supervisor: Robert S. Laramee

Abstract

Investors typically adopt diversified strategies, distributing resources among different stocks to optimise returns while mitigating risks. Contemporary stock market platforms and research instruments provide diverse stock data visualisations. However, such data is frequently siloed, precluding an integrated perspective for thorough inquiry and assessment, thereby limiting the full potential of portfolio management. In response to this identified gap, this paper introduces StockVis, a web platform based on a B/S architecture to facilitate integrated visual analysis for investors to optimize their portfolios. StockVis draws upon a comprehensive dataset derived from an anonymous investor's transactions between 30th April 2020 and 21st July 2023, complemented by specific metadata centred on the invested companies. This dataset forms the basis for comparing capital gains and dividends across stocks from 76 pertinent firms. The anonymous investor can actively differentiate between high-calibre and high-risk investment opportunities through tailored and interactive visualisations. The platform offers a comprehensive suite of tools, paramount among which is the capacity for inter-investment comparisons. Such ability extends beyond mere archiving, storage, and retrieval functions. Additionally, it features advanced visual analytic displays that allow for customization and the generation of more intricate individual views. These functionalities facilitate the generation of detailed personal perspectives and provide users with an enriched interactive interface. In conclusion, this dissertation establishes a seminal framework in stock trade visualisation, furnishing prospective investors with an innovative comparative instrument and theoretical approach.

Keywords: Stock Transaction Visualization; Investment Visualization; Financial Visualization;Investment Comparison; Visual Analysis; Transaction Data; Stock Markets.

Acknowledgments

I would like to express my profound gratitude to my supervisor, Dr. Robert Laramee. Throughout this project, he has provided invaluable guidance and unwavering support. I deeply appreciate his patience, enthusiasm, passion, and extensive expertise.He has consistently encouraged me to forge ahead and has provided invaluable suggestions for my project.

Additionally, I would like to thank my family for their continuous support and understanding throughout my academic journey.

Contents

1 Introduction

This thesis delves into an investment visualisation platform's research, design, and implementation. The platform offers investors an effective way to analyse and compare the performance of their stocks in stock portfolios. While multiple tools facilitate stock analysis, few specifically evaluate the performance of individual stocks or their portfolios. Throughout this thesis, insights are provided to assist potential investors in gaining a deeper understanding of stock performance, thereby facilitating informed investment decisions.

1.1 Motivation

Despite the surge in electronic stock transactions on a global scale and their escalating significance, delving deep into and analysing these transactions poses challenges due to privacy constraints. Stock investment data can provide insights into market trends, investor strategy preferences, asset valuations and potential risks. An informed interpretation of stock transaction data can guide pivotal decisions concerning portfolio optimisation, investment opportunities, and risk assessment. Consequently, a visualisation and visual analysis platform can provide invaluable insights to investors, enabling them to formulate data-driven investment strategies, promote optimal portfolio arrangements, and uncover latent market opportunities.

1.2 Field Challenges

There are two major difficult challenges in this field:

1) **Barriers to Individual Stock Investment Data:** There are multiple impediments to accessing genuine individual stock investment data. Foremost among these are concerns related to data privacy and security. The sensitivity surrounding individual investor behaviours and decisions can prevent institutions from sharing this data openly. Regulatory restrictions further compound the issue, as stringent guidelines protect the dissemination of such data to avoid market manipulations and insider trading. Additionally, the need for more standardization across different trading platforms and data repositories can pose a significant challenge. Various investment institutions might employ varied data recording and storage systems, making collating and gaining a holistic understanding of an individual's investment journey cumbersome. Moreover, while online platforms offer datasets, many restrict direct access, and some may question the authenticity of their content.

2) Challenges in Interactive Visualization of Investment Performance:
Interactive visualization tools aiming to highlight investment performance or portfolio dynamics face their own set of obstacles. The diverse nature of investment strategies, ranging from short-term trading to long-term holdings, requires comprehensive and adaptable visualization methodologies. A tool designed for one might not be appropriate for another. Furthermore, given the dynamic nature of stock markets, ensuring real-time or near-real-time data visualization is a technical challenge. Additionally, stock data is multifaceted, encompassing daily trades, capital gains, dividends, and more; presenting all this information intuitively and user-friendly can be daunting. Lastly, there's the challenge of balancing granularity with clarity. While investors seek detailed insights, an overload of information can hinder decision-making, defeating the tool's purpose.

1.3 Aims and Objectives

This project aims to provide the first authentic, publicly available dataset dedicated to individual stock investment transactions exclusively for analysis and research purposes. Furthermore, it aspires to produce an investment transaction visualization and analysis system capable of probing company metadata and comparing investment performance. The objectives of the project are:

- Provide the first open investment transaction data set from an anonymized investor.
- Develop a web-based interface to visualise and analyse extensive stock transaction data, enabling a detailed comparison of individual investment performances.
- The system facilitates the visualization of company-centric metadata associated with the user's investments, thereby providing a comprehensive perspective on the performance of individual stocks.
- The system comprehensively represents aggregate and individual stock investment performances, subsequently ranking them based on their respective investment outcomes.
- Provide users with the ability to construct and execute queries within the programme to help them locate particular stocks.
- To research the stock transactions of individual investors, give users options relating to the visual display of stock transaction data.

The paramount intent of this research initiative is to enable investors to differentiate between high-quality and high-risk investment opportunities by meticulously assessing the performance metrics of individual investments.

1.4 Key Results

The visualisation interface displays an overview of both stock and portfolio performances. Users can select, apply filters, and access detailed information as needed. Insights from two case studies demonstrate the effectiveness of the proposed solution in comparing and visualising stock investment performance metrics, including its capability for dynamic ranking. The application adeptly retrieves and displays data in response to user queries. For investors to gain a broader understanding of the investment performance of various stocks, obtaining a more comprehensive set of transaction data is crucial. Additionally, it's essential to focus on parallel coordinate charts that use company data as metadata and bar charts that rank investment performance. Future research should prioritise adopting advanced visualisation techniques, using clear and intuitive graphics to analyse specific investment datasets comprehensively.

1.5 Thesis Structure

The thesis is structured following the recommendations from Bob's Project Guidelines [Lar11]. The rest of this paper is as follows: Section 2 reviews past studies focusing on the visual analysis of investment transaction data. Section 3 details the data preparation and organisation processes, followed by the project specification in Section 4. A complete project plan is described in section 5. Section 6 elaborates on the design of the project.

In contrast, Section 7 introduces the system's implementation, emphasising the display of primary performance and enhancements, accompanied by a brief overview. Section 8 presents the evaluation of the results from case studies. Finally, Section 9 offers conclusions, with Section 10 discussing broader implications and potential directions for future research.

2 Background

This section first provides a comprehensive literature review, emphasizing works and techniques relevant to the project. The second part describes the systems relevant to this project. Through an exhaustive examination of the literature and an analysis of prevalent visualization platforms, a deeper understanding of the visualization and analysis of individual investment transaction data emerges, establishing a firm foundation for further research endeavours.

2.1 Related Work

In the rapidly evolving domain of financial visualization, a multitude of visualization and financial analysis tools have been extensively integrated into business intelligence and market analysis software. These advanced instruments offer traders in-depth market insights, facilitating informed and timely decisions [ID17; PM17].

The increasing reliance of investors on technological innovations reflects the growing demands of traders and the dynamic nature of markets. Recent academic research underscores a heightened interest in investment visualization, especially studies on stock portfolios. This section aims to provide a comprehensive understanding of the topic, starting with exploring financial visualization surveys and delving into a detailed analysis of relevant research papers from academic databases.

2.1.1 Overviews and Surveys

McNabb and Laramee [ML17] introduced the inaugural Survey of Surveys (SoS) in information visualization, devising a unique taxonomy for survey literature in the field. This categorization acts as a guide for budding researchers and a reference for experienced professionals. They detail their method for curating literature, outline the scope of the survey, present a classification methodology, and underscore challenges identified in each study. Moreover, they provide summarized survey documents spanning various categories, such as data-focused surveys, multivariate and hierarchical assessments, graph and network analyses, geospatial and temporal dimensions, coordinated multiple views, and real-world applications and scenarios. Using SurVis, they identify prevalent research themes, offering perspectives on the appropriateness of a method, the effectiveness of diverse approaches, and takeaways from visualization design choices.

Roberts et al. [RL18] thoroughly analyse emerging trends within business visualisation and visual analytics, addressing prevalent data complications and elucidating how various industries leverage graphic design to comprehend the business milieu better. Roberts et al. [RL18] segment the literature into three main domains: business intelligence, the business ecosystem, and a customer-oriented focus, further sub-dividing based on the data source. The authors delineate their research approach and scope, categorising the reviewed articles under these domains. They observe a prevalent reliance on existing databases within the business ecosystem due to their ease of access and relevance. In contrast, customer-oriented visualisation studies have transitioned from prioritising feedback to emphasising behaviour, as seen in Figure 1, potentially influenced by the growing accessibility of GPS data. Although web scraping often emerges as a cost-effective data collection method, concerns regarding data integrity persist. Business visualisation is exploratory, with its potential becoming more pronounced as its efficacy undergoes further validation.

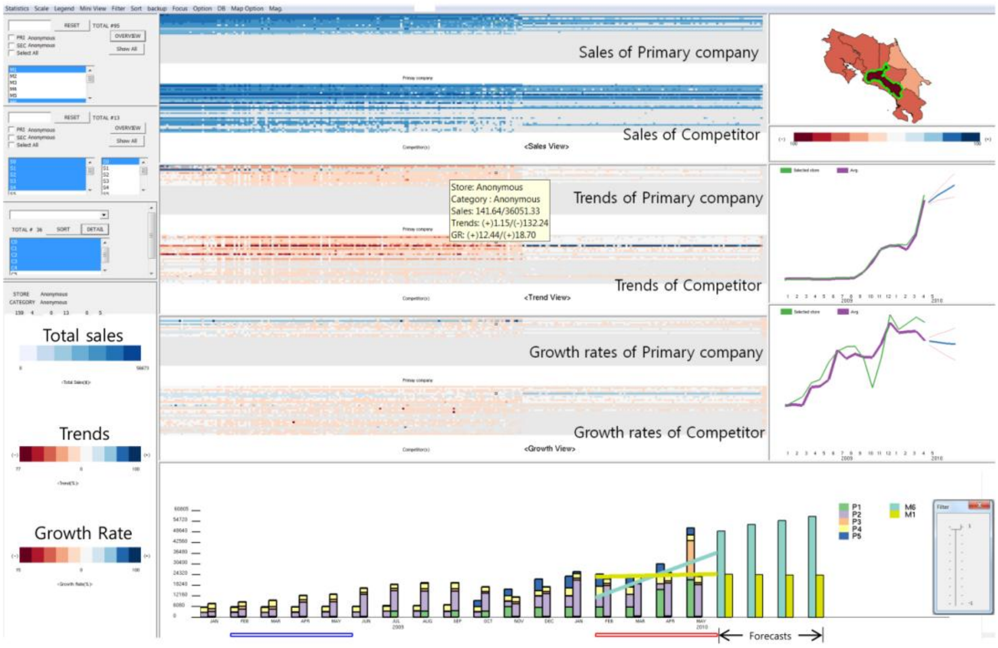


Figure 1. A screenshot of the customer interface: the left panel showcases filtering choices and colour legends, while the central area presents pixel-based comparisons. The right panel offers a geographical perspective, with the bottom panel featuring the stacked bar visualization and time sliders [RL18]**.**

Rees, D., & Laramee, R. S. [RL19] introduce the inaugural literature review on information visualization books, utilizing an innovative bifurcated classification centered on both the book and chapter themes. This framework aids readers in swiftly ascertaining the depth of coverage pertinent to their topics within specific titles, facilitating the selection of the most relevant book from the burgeoning collection. The authors elucidate challenges faced during the survey, the strategy for book discovery, and delineate the scope of the examined books. The categorization structure is detailed, where the primary tier is audience-centric, and the subsequent tier focuses on chapter themes. The books under review are segmented into six distinct groups based on their target readership, with concise summaries accompanying each chapter. An evaluative and suggestive segment is incorporated, juxtaposing each title, emphasizing its merits, and proffering reading suggestions. Their scrutiny discerns that among the books reviewed, "Visual Design" emerges as the predominant theme. Rees, D., & Laramee, R. S. [RL19] envision delving into additional facets of data visualization in upcoming studies.

Ko et al. [KC+16] systematically review visualization and visual analytics within the financial domain. Employing established classification systems, they categorize financial tools based on data sources, automation procedures, visualization approaches, interactive elements, and assessment methods. Their study details the survey's methodology, pertinent tasks, and system prerequisites. They expound upon the data sources suited for visualization in financial analysis and discuss automated methods, pinpointing emerging prospects in anomaly detection and predictive modelling. Further, they probe the transformation of financial visualization techniques, an evolution propelled by the intricacies of financial data and the imperative for proficiently interpreting extensive, multidimensional, and hierarchical datasets. Figure 2 proposes a crucial technique for preventing the blocking of various line graph views. The research also sheds light on different interactive strategies utilized in visualization. Concluding their findings, Ko et al.[KC+16] emphasize the widespread adoption of case studies as the primary means for assessing the effectiveness of visualization techniques in financial data analysis.

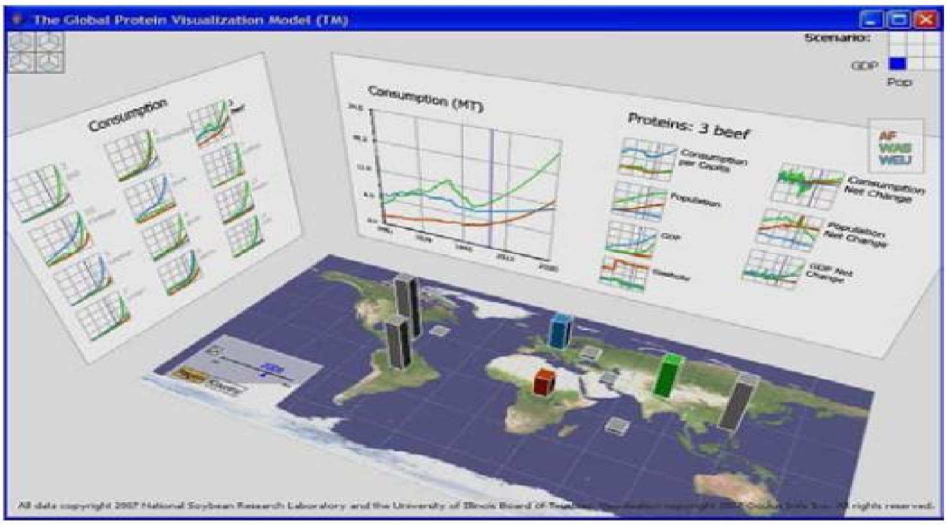


Figure 2. A display capture of the analysis interface showcasing a commodity market in 3D [KC+16]. This workspace melds 2D line diagrams with 3D bar (or column) representations.

The study by [KO+16] presents a compilation of commercial and academic articles focusing on stock trading and fund management paradigms. This research highlights analytical methods and visualisation techniques, notably spotlighting visualisation methods in section 7. The analysis indicates that considerable research heavily leans towards conventional 2D representation tools, including line charts, pie diagrams, and their ilk [KO+16]. Tobias Schreck and team [Sch+09] introduced an innovative utilisation of line charts, delving into the depiction of grouped trajectory data via interactive Kohonen maps. Their work allows users to engage with a self-orchestrating map interactively, providing the latitude to manipulate the underlying algorithm's specifics [Sch+09]. Such generated visual illustrations from clustering endeavours are available for selection and juxtaposition, thus facilitating exhaustive scrutiny of the algorithmic methods. Other scholars, including Dao and associates [Dao+08], propose techniques leveraging wedges designated to individual stocks to decipher force dynamics in the NASDAQ financial market.

2.1.2 Visualisation of Investment Transactions

Firat et al. proposed "MoneyVis"[FVS+23], a meticulously anonymised bank transaction dataset for research and analysis. Drawing from comprehensive reviews and research in visual analytics within the financial sector, as well as established visualisation methodologies for financial transactions, this study acknowledges the current spectrum of accessible transaction data. In the practical implementation, the research delves into a dataset spanning seven years, encompassing more than 6,500 anonymised retail banking activities. These transactions have been systematically categorised through manual, semi-automated, and fully automated techniques. Firat et al. [FVS+23] undertook a comprehensive analysis using sophisticated interactive visualization tools to validate the dataset's relevance and utility. They implemented techniques like stacked bar charts (Figure 3), location-centric pie charts (Figure 4), and sunburst diagrams to discern intricate spending patterns and evolving transaction trends. Notably, the utilized data, characterized by its abstract nature, temporal relevance, structural integrity, and multidimensionality, enriches the depth of understanding about transactional data.

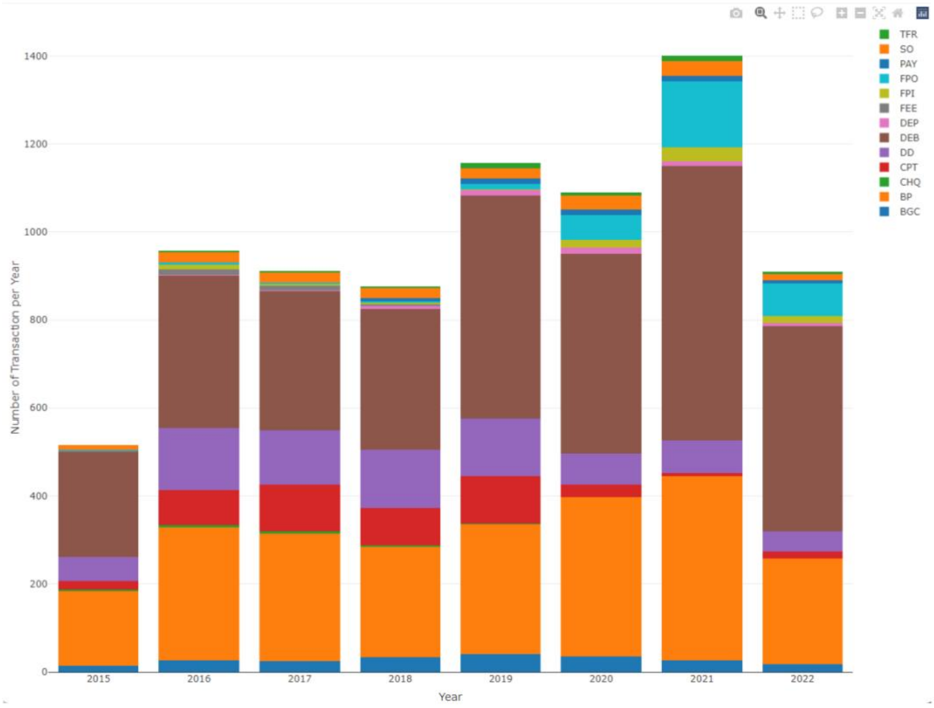


Figure 3. A screenshot of the transaction count (y-axis) plotted against bank transaction categories spanning 2015-2022[FVS+2023].

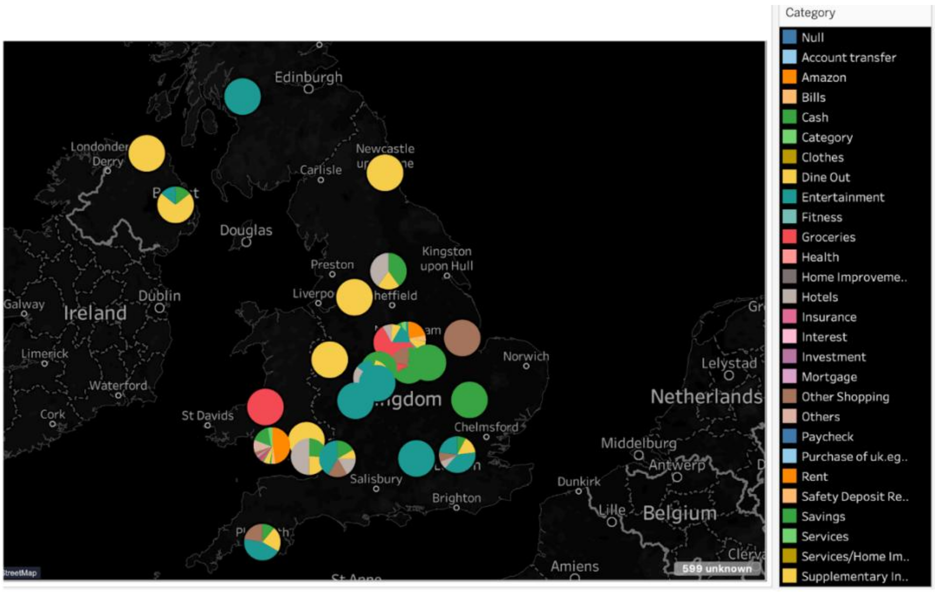


Figure 4. A screenshot illustrates the distribution of money spent across UK categories, with a detailed view of specific city expenditures [FVS+2023].

E.Sorenson and R. Brath [SB13] present an innovative visualization technique, as illustrated in Figure 5, which effectively merges continuous time-series data, exemplified by stock prices, with discrete events such as news bulletins, earnings announcements, and other notable occurrences. Their methodology stems from foundational concepts in event-based and time-series visualization, enriched by interactive features. The project's execution encompassed stages: categorizing event types, data collection and preprocessing, crafting the visualization design, and embedding interactive capabilities. The data chosen for this venture uniquely merges time series elements with discrete occurrences. Evaluation metrics prioritize user performance, highlighting user contentment in terms of visual appeal and the breadth of event representation.

Figure 5. A Screenshot of visualization merging discrete events with price time series by Eugene Sorenson et al [SB13].

Hua J. et al. [HW+18] propose a distinctive approach that blends the force-directed algorithm with time-series charts, striving for an integrated perspective on relevant stock structures and intricate interlock links. This approach, which juxtaposes weighted graphs with time-series visualizations for selected equities, is anchored in established methods, including FolioMap, SelfOrganizing Map, FundExplorer, Ordered Treemap, Parallel Coordinates, and Principal Distribution of Chart Shapes.The methodology unfolds in four phases. Initially, they extract raw stock data from the ASX, which is then meticulously formatted and filtered, resulting in a streamlined dataset and corresponding graph models. These models are subsequently manipulated using the force-directed algorithm and time-series representations. The culmination of this process is an analysis of stock interconnections derived from the juxtaposition of graph layouts and time-series visualizations. The dataset is characterized by its structured, bi-dimensional, time-sequential, and multivariate nature. Hua J. and the team employ a quantitative lens for evaluation, amassing a substantial volume of stock data to investigate stock correlations rigorously. They use their method to craft connected and undirected graphs, subsequently examining network characteristics such as node dimensions, edge width, and span.

Arleo, A. et al. [AT+23] established Sabrina 2.0, seen in Figure 6, an enhanced Visual Analytics (VA) system adept at navigating financial data across various magnitudes, from specific business-centric information to encompassing national-level summaries. This innovation builds upon existing methodologies, enhancing areas such as financial data portrayal, geospatial visualization across scales, representation of transactional networks, and the depiction of transactions as flows. The realization of Sabrina 2.0 involves sequential steps: (1) Crafting the core blueprint of Sabrina, which integrates diverse views and interconnected components; (2) Choreographing the synergy between the visualization interface and the core model dedicated to analyzing trade nuances;(3) They are employing JavaScript within the React framework and the deck.GL platform for application development, sourcing data from CSV files, and integrating model parameters through JSON configurations. The data under analysis showcases characteristics such as spatial orientation, temporal elements, structured layouts, three-dimensional aspects, and multivariate data points. To gauge the robustness of Sabrina 2.0's visualization and user interaction mechanisms, Arleo, A. et al. undertake a comparative case study examination.

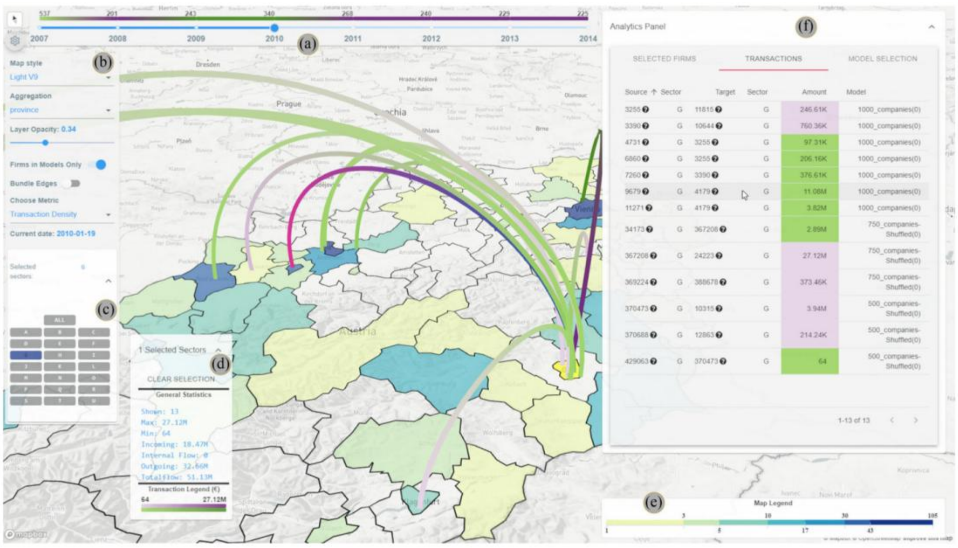


Figure 6. A screenshot of the Sabrina 2.0 system showing the timeline, display panel, details box and transaction table [AT+23].

Chan and Qu [CQ16] put forth advanced visualization techniques tailored for deciphering financial news narratives. Their approach aids investors in dispassionately grasping market volatilities and understanding the influence of various economic indicators. The foundational theories that underpin their work span the visualization of news articles, the interplay between narratives and financial events, and visual depictions of relational and keyword-based data. The proposed narrative visualization strategy unfolds over three distinct phases:(1) Architecting the overarching narrative framework; (2) Mining pertinent details from financial news, followed by its categorization and topic modelling; (3) Handpicking suitable visual aids like word clouds, interconnected graphics, summative plots, and spherical charts.The data processed in their work displays a variety of facets: it's multidimensional, temporally sequenced, lacks a uniform structure, and is predominantly text-based. To substantiate the practical utility of their approach, Chan and Qu [CQ16] delve into two illustrative scenarios: the 2014 oil price trajectory and post-2008 financial downturn unemployment trends in Europe.

Lei and Zhang [LZ10] introduce a visual analytics framework designed for a nuanced exploration of stock market data, employing a unique ring-centric visualization approach seen in Figure 7. This methodology grants investors a holistic view of specific market data within a broader context. The ring-centric visualization, as conceptualized, employs categorical variables to craft arcs symbolizing the market capitalization scale. Colors within the visualization are strategically used to demarcate distinct market sectors, while the radius offers insights into the stock's performance metrics. In tandem with this visualization, the researchers also present a clustering mechanism for stock price analysis, underpinned by the k-means clustering technique [LZ10]. This mechanism amalgamates both temporal factors and quantitative data, facilitating the portrayal of price dynamics relative to a stock's inherent value, aptly illustrated using scatter plot representations.

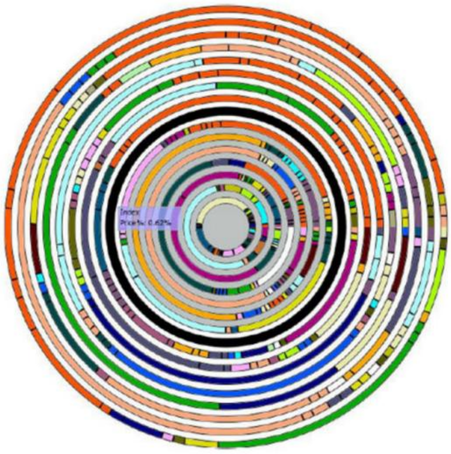


Figure 7. A screenshot of the ring visualisation for the stock market [LZ10].

Javad and Vincent have developed "HFTViz,"[YG22] a dedicated visualization tool (seen in Figure 8) for high-frequency trading data to tackle the complexities inherent in data from trading machines, enabling financial researchers to gain lucid insights into market dynamics. HFtTViz draws from principles of multi-dimensional financial visualization and incorporates time-oriented visualization techniques. D3.js was instrumental in configuring overarching and granular visual perspectives in the developmental phase. There was a concerted effort to fine-tune colour palettes and data visualization modalities to suit a range of display screens. React.js was incorporated to enhance user interactivity. Integrating HTML, CSS, and Javascript in the final iterations ensured that HFTViz proficiently managed data characterized by high frequency, multi-dimensionality, time sequencing, and multiple variables. To evaluate the utility and effectiveness of HFTViz, Javad and Vincent embarked on a comprehensive case study involving five experts specializing in financial data analytics. These intensive discussions aimed to gather insights and identify any challenges or hurdles users face. Based on these findings, subsequent tool iterations were enhanced to proactively address potential user concerns, ensuring an intuitive and user-friendly experience.

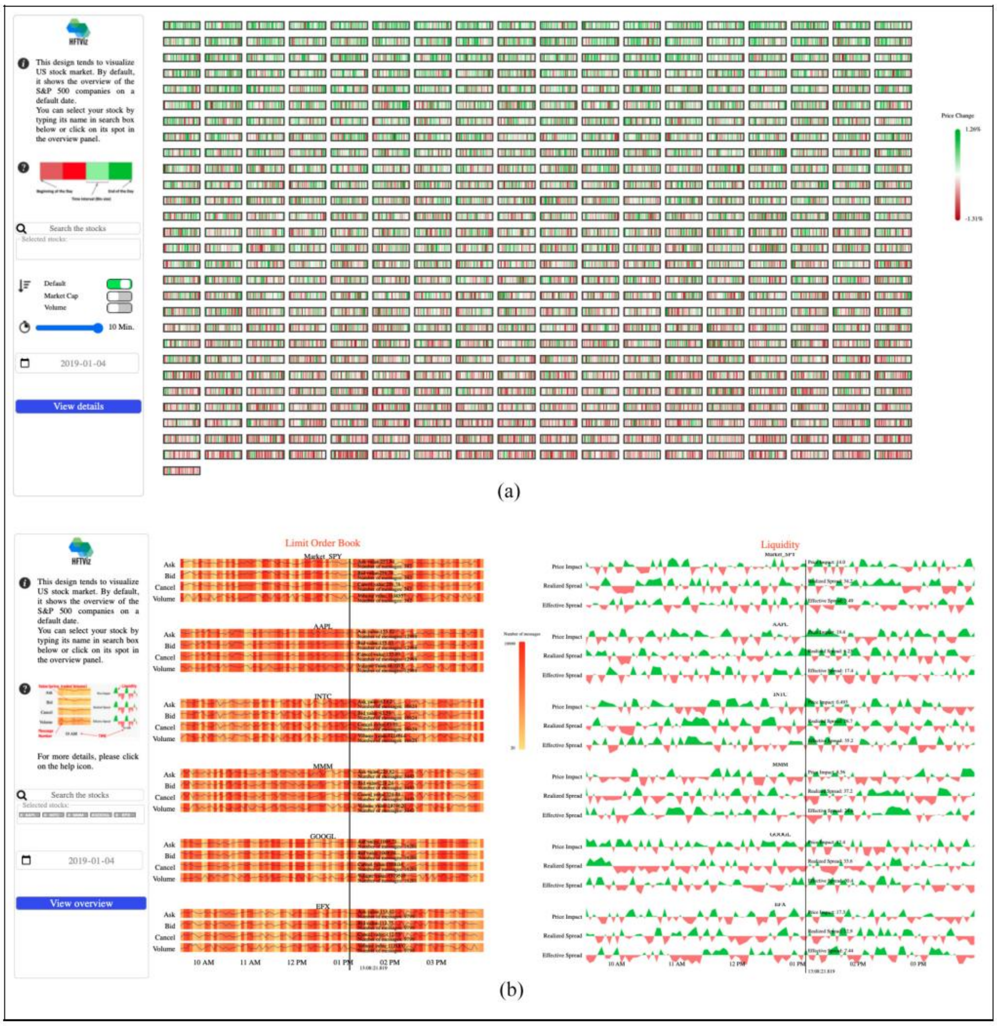


Figure 8. A screenshot of the HFTViz system [YG22] shows market movement for the selected date and detailed information on surveyed stocks.

Roger A. et al. presented EVA [LG+18], seen in Figure 9, a visualization technique to bolster fraud investigations and enhance fraud detection systems' accuracy. This approach integrates technologies such as anomaly detection, neural networks, and data mining. For its implementation, the team used time-dimensional views for daily transactions, parallel coordinate plots for scores, and scatter plots to correlate transaction counts with amounts. They aggregated amounts and frequencies with two bar graphs and integrated an account selector with dynamic data pivot charts. The data in EVA is multi-dimensional, geographical, and time sequential. To measure the effectiveness of their proposed method, the team undertook a qualitative evaluation, soliciting feedback from three users. This feedback, juxtaposed against other tools, illuminated EVA's strengths and weaknesses, insights garnered during its use, and any potential areas for enhancement or missing functionalities.

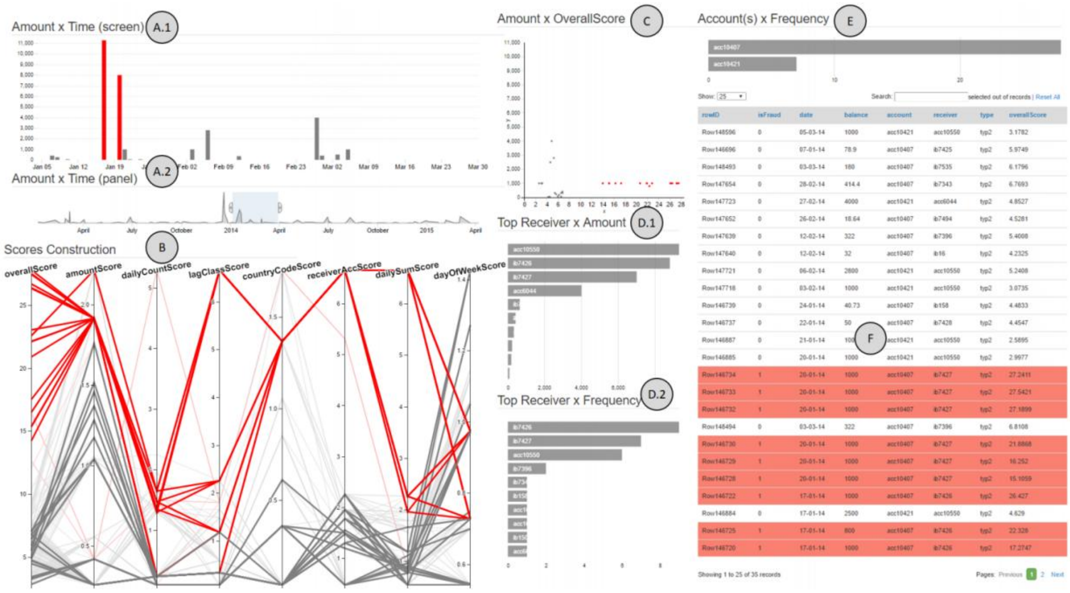


Figure 9. The screenshot of the EVA system [LG+18]. Suspicious data are highlighted in red.

Yue et al. invented "sPortfolio,"[Yue+19] an innovative visualization tool in Figure 10 tailored for factor investing. It offers a holistic view of factor data, streamlining analysis at three levels: the broad market dynamics (Risk Factor Level), portfolio strategies (Multiple Portfolio Level), and specific trades (Single Portfolio Level). Built on the Barra Risk Model, sPortfolio merges portfolio and financial time-series visualization techniques. As a web-based full-stack application, it integrates data via RQdata, processes it with pandas, and offers users four insightful views to explore: Portfolio Cluster, Factor Correlation, Comparative, and Single Portfolio. The tool effectively manages multi-dimensional, multi-variable, and time-sequential data. Its effectiveness was gauged through case studies, touching on portfolio comparisons, performance, factor crowdedness, and model measurement.

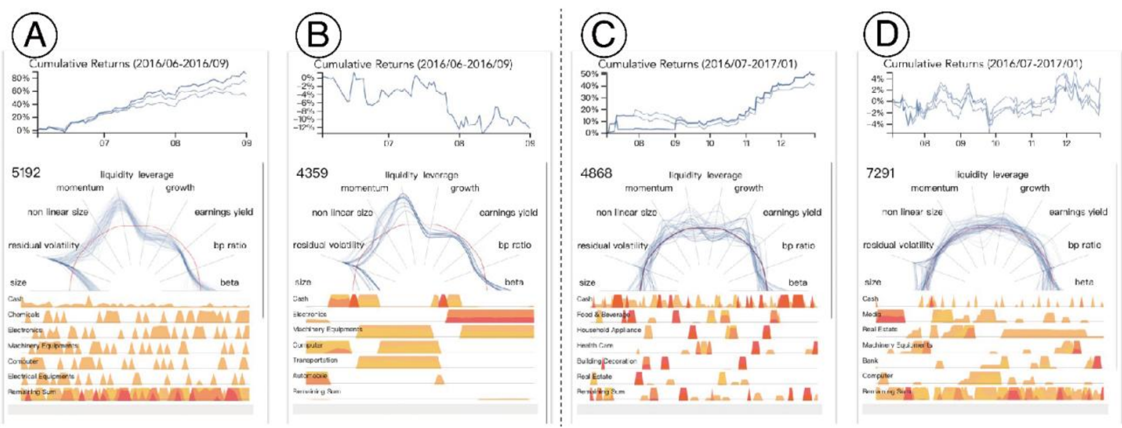


Figure 10. Screenshots of the various portfolios are displayed in the comparison view, with a line graph at the top of each cluster showing the cumulative return for the selected collection [Yue+19].

Yujing et al. introduced the BitVis system [Sun+19], an interactive platform designed to elucidate the relationships between Bitcoin accounts. This system allows users to filter transactions, engage with the underlying transactional network, and conduct in-depth analyses of Bitcoin behaviours. In the section discussing related work, the focus begins on visualization tools that primarily serve demonstrative purposes. It then transitions to sophisticated tools emphasizing detecting anomalous activities on the blockchain and interactions between accounts.The BitVis system stands out by comprehensively examining Bitcoin behaviours through two pivotal components: a backend module responsible for data acquisition and preprocessing and a frontend module dedicated to interactive visualization. The system extracts data from wallet resource managers and subsequently offers an array of visualization techniques to explore Bitcoin addresses, entities, and transactions in detail. Moreover, BitVis facilitates user engagement with the visual representations, allowing data filtering and access to granular account details. The system uses Python, PyQt, relational, and graphic databases to process abstract, two-dimensional, temporally-influenced multivariate data. The evaluation segment illuminates the system's proficiency in Bitcoin behaviour analysis through a case study focusing on the Silk Road entity.

2.2 Previous Systems

This section delves into various stock visualization and analysis software tools, examining how these systems achieve their visualization features. By scrutinizing these tools' data and visual displays, one can discern which data points and trends are most valuable when considering investments. Furthermore, observing current software visualizations can guide effective strategies for representing investment data.

2.2.1 Google Finance

Google Finance [Goo20] is an online portal for financial data developed by Google as part of its suite of search engine products. This free service permits users to directly search for "finance" or specific stocks across all Google-supported clients, including Android, iOS, and the Web. As illustrated in Figure 11, the application showcases relevant news sources and can offer comparisons with other stock data.

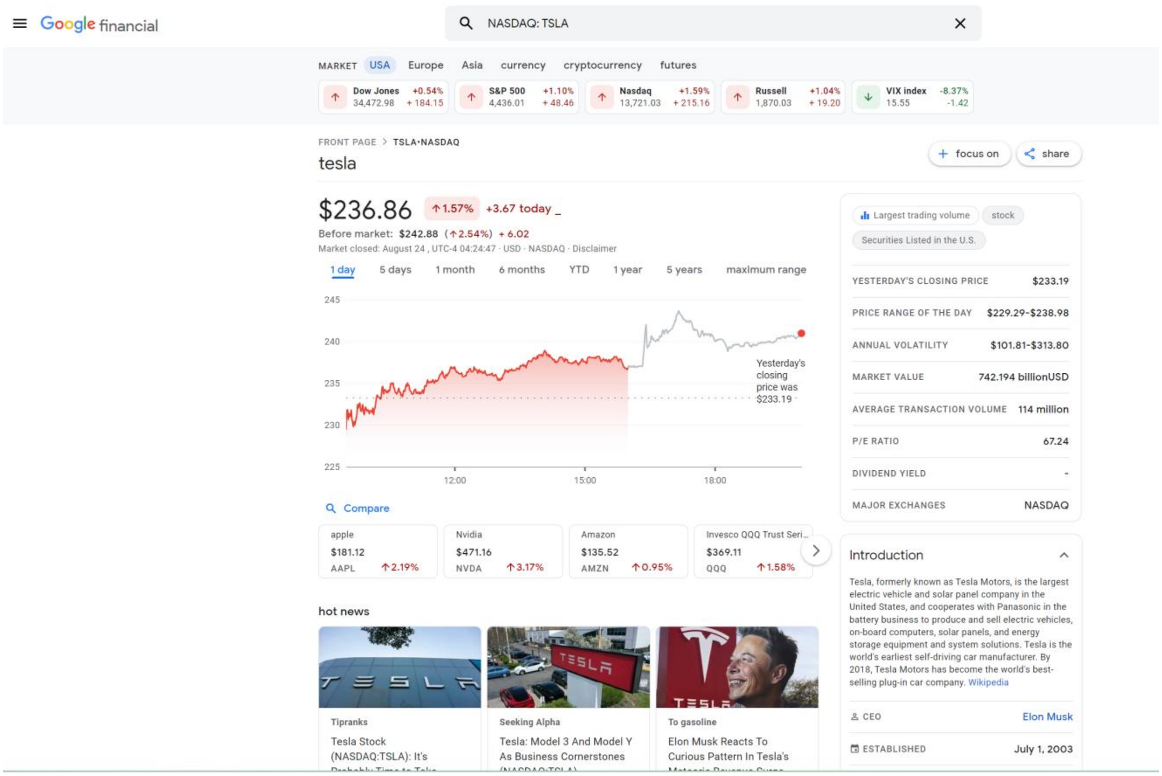


Figure 11. A visual capture of the TSLA stock from the Google Finance search interface [Goo20], highlighting time-series charts and related news articles.

Users input the stock or company of interest in Google Finance and swiftly receive pertinent information. This platform predominantly attracts novice financial traders or those desiring a cursory glance at market trends, particularly those not pursuing in-depth data analytics. Google Finance emphasizes stock charts and data aggregation, focusing on a company's financial health rather than an expansive market analysis. Beyond real-time data, it provides historical market data to users. Additionally, alongside the line graphs, the market capitalization, the previous day's closing price, and the annual price fluctuation range are presented textually.

2.2.2 Yahoo Finance

Yahoo Finance [Yah20] is an online financial data portal by Yahoo, offering free, comprehensive tools for detailed financial analysis and research. Designed as a web-based application, it runs seamlessly across all contemporary browsers, offering real-time financial data that spans major exchanges to cryptocurrencies. Figure 12 illustrates that the platform presents stock information textually, complemented by time-series charts, facilitating in-depth exploration and swift browsing.

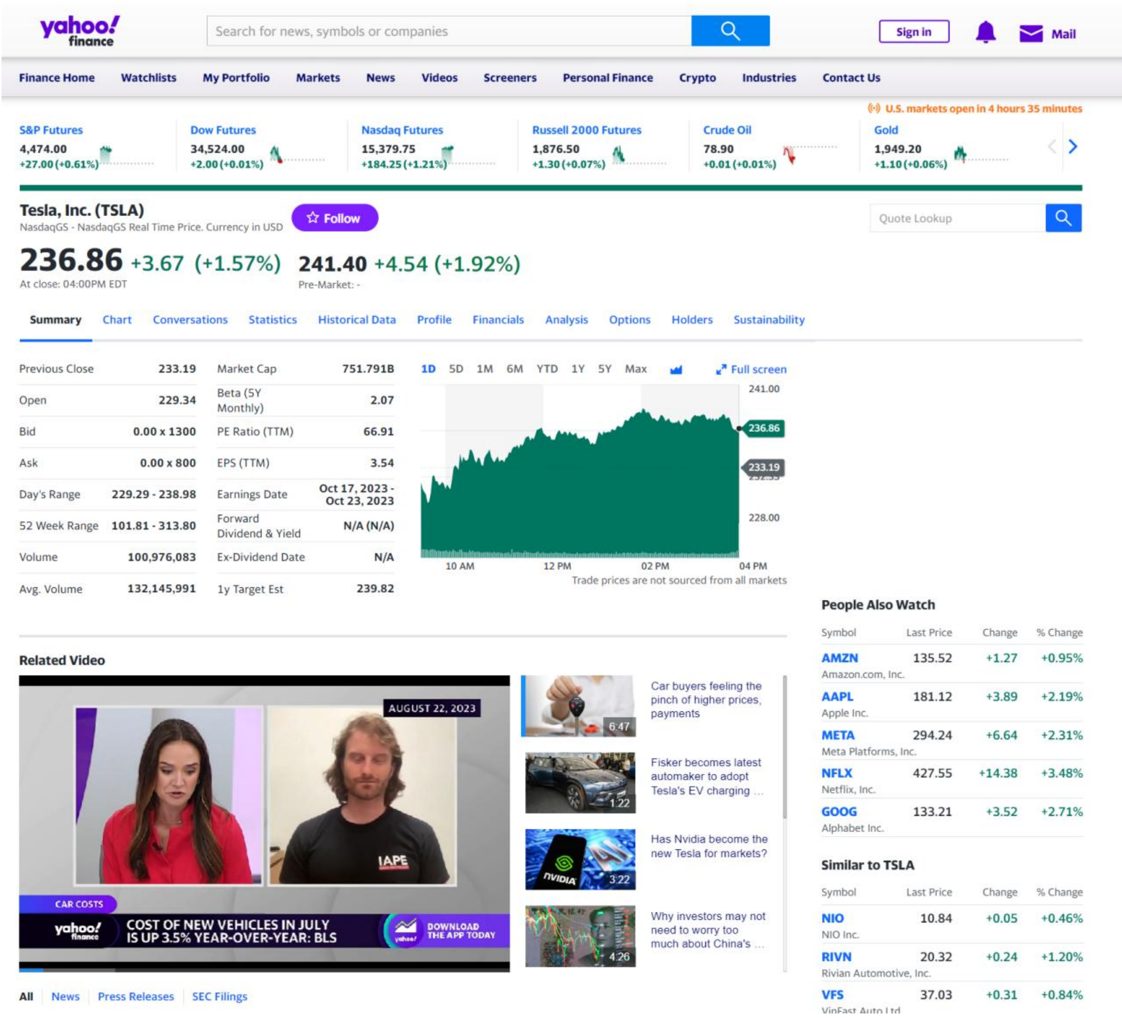


Figure 12. A capture of the TSLA stock on the Yahoo finance website [Yah20].

While the platform allows for data display customization, it doesn't support data comparison beyond its predefined organizational lists and search queries. Yahoo Finance is primarily tailored for financial traders and those interested in economic news, with news content sourced and aggregated through Yahoo's search engine. In essence, Yahoo Finance emerges predominantly as an analytical and data-driven portal with limited real-time data visualization functionalities.

2.2.3 Seeking Alpha

Seeking Alpha [Alp20], positioned as an online stock market insight application, offers users profound insights and knowledge by delving into corporate financial data. It has a limited free version and introduces an advanced data and service subscription priced at $199 per year. Not only can users access a vast dataset, but they can also import their stock portfolios and obtain consolidated news and advice on the platform seen in Figure 13. Like Google Finance and Yahoo Finance, Seeking Alpha presents users with information related to specific stocks. However, it differentiates by providing more comprehensive financial data, such as dividend payout rates.

Furthermore, Seeking Alpha has two paid modules: advanced features and a professional version priced at $2400 annually. The paid modules are essential for those aiming for in-depth examinations of a company's financial health, such as profitability. The inclusion of both primary and professional subscription modules suggests that Seeking Alpha caters to both financial novices and professionals in the finance sector. To provide potential customers with a trial experience, Seeking Alpha offers a 7-day trial period for its advanced feature version. Content-wise, it combines in-depth analytical articles with basic investment strategies tailored for beginners, aiming to meet the needs of a diverse investor audience.

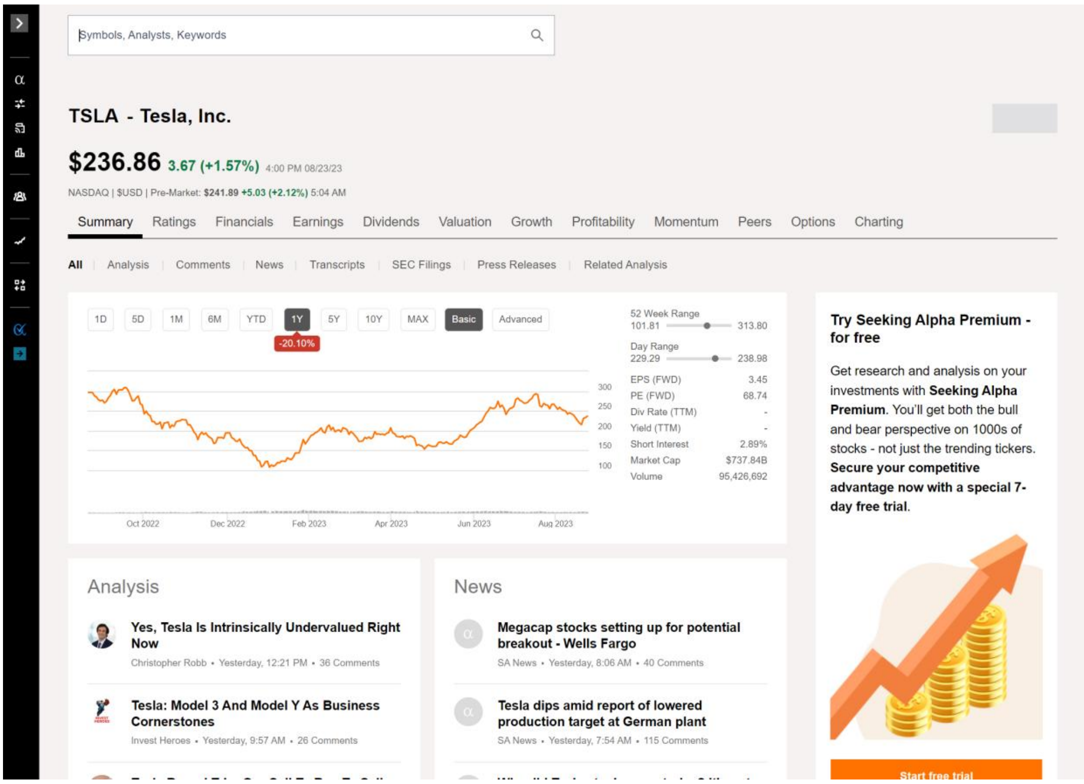


Figure 13. A capture of the TSLA stock on Seeking Alpha [Alp20].

2.3 Potential Contribution

The literature review and examination of previous systems have revealed various visualisation methods for financial markets. Modern web applications such as Google Finance [Goo20], Seeking Alpha [Alp20], and Yahoo Finance [Yah20], although proficient in presenting extensive financial information, need more capability to compare investment portfolios across diverse stocks. As a result, the primary objective of this research is to introduce the first open investing transaction dataset and develop a visual interface, providing an overview of the transactions and enabling users to explore the performance of investments.

3 Data Preparation

Data is an integral component of all visualizations, acting as a significant factor alongside user experience in guiding the selection and attributes of visualization methods.This section introduces the data used for this project, encompassing its source, type, size, format, and characteristics.

3.1 Data Sources and Description

This study focuses on two distinct types of data related to investment activities:

- **Raw Transaction Data**: This encompasses investment records directly downloaded from the Trading 212 platform, spanning from April 30, 2020, to July 21, 2023.
- **Derived Metadata**: Based on the raw transaction data, the research has computed and generated company-centred metadata to provide further insights and statistics about investment patterns and behaviours.

To the best of available knowledge, this dataset represents the inaugural release of publicly available investment transaction data for scholarly research.

3.1.1 Original Trading Data from Trading 212

The investment data for this study is exclusively sourced from the Trading 212 platform, covering the period from April 30, 2020, to July 21, 2023. Initially downloaded in CSV format, these records were meticulously organized and processed to produce a comprehensive dataset that encapsulates all trading activities of investors during the specified period, as despict in Figure 14. After organization, the dataset comprises 2,255 entries. By conducting an in-depth analysis of this data, the research aims to gain a holistic understanding of various patterns of individual investment behaviours, such as daily trading frequency, engagement with specific stocks, and the amount of each transaction. Each investment entry encompasses 22 distinct features, of which the most critical include:

- Transaction type,
- Transaction date,
- Ticker

- Company name,
- The number of stocks traded,
- Price per stock,
- Transaction result,
- The transaction amount.

It is noteworthy to mention that the "transaction type" feature is automatically generated and annotated by the investment platform.

3.1.2 Company-Centred Metadata

This research developed a company-centred metadata dataset based on the original transactional data about 76 stocks, as seen in Figure 15. The principal objective is to analyze the investment performance of these stocks comprehensively. The inception and the latest stock transaction dates were recorded upon detailed analysis of the data. This research calculated various metrics, including the total volume of stocks purchased, the cumulative investment, the average purchase price, and the net volume of stocks retained post-transaction. The price of the most recent stock sale was also factored into the analysis. Specifically, as of August 3, 2023, stock valuations were utilised to analyse fundamental returns and dividend distributions. Additionally, background information for each stock was investigated, including stock splits and past company acquisitions. The dataset comprises 17 distinct attributes, with the most critical attributes ：

- Company name,
- Ticker,
- Average purchase price per share,
- Stock price on the last trading day,
- Net number of shares held,
- Stock price as of August 3, 2023,
- Cumulative dividends up to July 21, 2023,
- Realized capital gains or losses,
- Unrealized capital gains or losses.

After meticulous organization, all datasets were archived in spreadsheet format and uploaded to Google Sheets, a cloud-based spreadsheet platform. Several factors influenced the decision to utilize this particular storage solution. It ensures reliable data storage, significantly reduces the risk of data loss, and facilitates seamless data sharing among researchers, ensuring continuous accessibility from any location.

Analyzing these datasets offers a comprehensive perspective on the trading patterns of individual investors over the past three years. Furthermore, it enables a detailed assessment of the performance of distinct stocks within their respective investment portfolios.

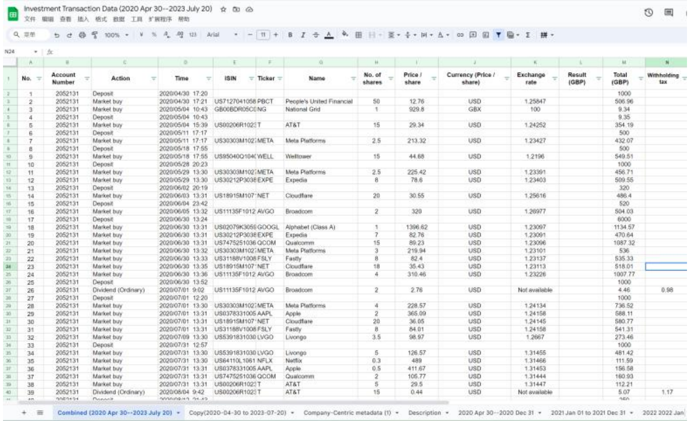


Figure 14. The screenshot of Investment dataset.

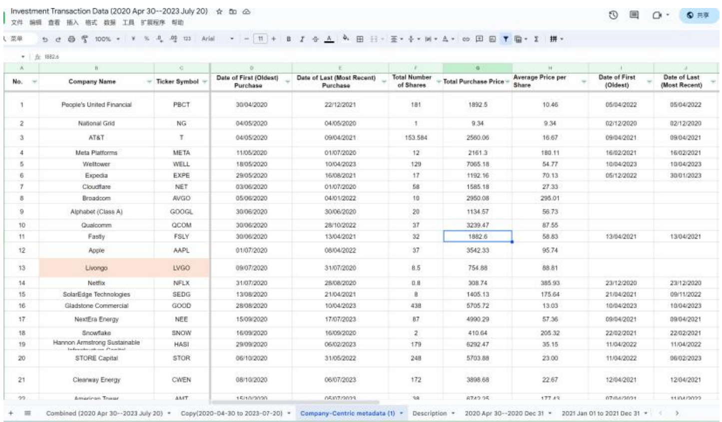


Figure 15. The screenshot of company-centred metadata

3.2 Accessing Data

To ensure the anonymity of the investor and safeguard their privacy, all information that might disclose the investor's identity, such as account names and other relevant personal details, has been removed from the data. Furthermore, to assist researchers in gaining a deeper understanding and analysis of the dataset, the dataset includes a detailed feature explanation table that elucidates the content of each column. The following URL provides read access to the open investment transaction data set: https://tinyurl.com/4aedp524

4 Project Specification

4.1 Feature Specification

Established on insights from Bob's Project Guidelines [Lar11], a feature specification has been formulated to comprehend the essential functionalities and potential improvements for the system. This structured list of features serves as a guideline towards the successful completion of the project.

4.1.1 Basic Features

Essential features are crucial for the system's core functionality and for the project's successful completion. The criteria were formulated based on Ben Shneiderman's Visualization Mantra' [Shn03]. This mantra outlines the principle for creating the visualisation systems: beginning with an overview, then filtering and selection, and concluding with details on demand. The fundamental features encompass the scope of the initial product and the planned application to be developed.

- Provide an interactive visualisation system.
- Construct an assortment of tools for user selection.
- Data is initially stored in a .xlsx format and converted to JSON for more efficient data processing and retrieval.
- Provide a parallel coordinates plot to provide an overview of the transaction data and compare investors' investment metadata across different companies.
- Provide a feature for users to remove undesired axes in the parallel coordinate plot.
- Generate a stacked bar chart to compare and rank the investment performance of various invested companies.
- Offer a user option for filtering by company name or stock ticker.
- Provide users an option to toggle between the parallel coordinates plot and the bar chart.
- Generate a calendar chart to exhibit the timeline of all investment activities for the anonymous investor.
- Generate a pie chart to represent anonymous investors' transaction volume and associated funds. In this chart, the sector angle for each company directly corresponds to its transaction amount: the larger the investment, the wider the angle.
- Integrate pie charts into calendar charts. On the calendar representation, the pie chart radius for a particular day is scaled according to the transaction amount for that day. Larger transaction amounts correspond to pie charts with larger radius.
- Generate a stacked bar chart to exhibit the daily transaction amounts for different companies, differentiated by their respective actions. In this visualization, the X-axis represents the transaction amounts, while the Y-axis depicts the stock tickers of the traded companies.
- Construct a comprehensive table to list the specific data and details of daily transactions.
- Construct a user option to filter the date of the calendar view.
- Distinct categories like companies are distinguished and represented through

colour mapping.

4.1.2 Enhancements

Enhancements are integral components that build upon and refine previously delineated foundational features. These augmentations are typified by integrating additional functionalities into either the visualization component or the broader application framework to bolster the efficacy of outcomes. These enhancements are mentioned in Dr. Robert Laramee's paper [Lar11]. Implementing such enhancements optimizes tasks under certain conditions and enhances the system's comprehensiveness and efficiency.

- The parallel coordinate view renders selected companies with colour emphasis as the focal point while representing unselected entities transparently to provide a clear focus on the chosen companies.
- The stacked bar chart introduces two fundamental categories: Dividends and Capital Gains/Losses. The initial display presents overall performance rankings and their monetary values. Users can filter between dividends and capital gains/losses, and the system dynamically adjusts rankings, displaying specific amounts accordingly.
- The system incorporates a user option: when users select to aggregate data by month or year, the calendar view transitions to a superpositioned calendar graphic.
- Generate the sunburst chart to better delineate distinct transaction behaviours while offering a detailed understanding of the associated companies and their respective transaction amounts.
- Integrate the calendar view with other graphical representations. Upon selecting a particular date within the calendar interface, corresponding visualization charts are dynamically updated to reflect data pertinent to the chosen date.
- All visualization components dynamically adjust to accommodate varying screen resolutions.
- In the visualization interface, as the cursor is positioned over any graphical element, an informative tooltip emerges, providing users with precise data insights and elucidations, consequently amplifying the interpretability of the depicted data.

4.2 Technology Choices

According to the project specifications and delineated features, the ensuing chapter details the selection of technological choices.

4.2.1 Programming Languages

Ensuring that visualisation tools align with distinct requirements is paramount in the rapidly evolving technological landscape. While Bob's project guidelines [Lar11] advocate for the Java Swing Library, this recommendation may only be partially relevant to the needs of modern, cross-platform web applications. Despite its proficiency in desktop application development, Java Swing may not cater to the broader accessibility demands of today's web platforms.

Python and R have established their significance in data analysis and visualisation. However, it's vital to distinguish their primary objectives. Ceyhun Ozgur et al. [Ozg+17] argue that both languages are fundamentally statistical in orientation, often prioritising data computation over interactivity. R, designed primarily for statistical processing, typically produces static visualisations, even with powerful libraries like ggplot2. Although integrating the Shiny library introduces interactive aspects to R, it does not alter its core identity as a statistical tool. With its Plotly library, Python offers interactivity. However, it's more acknowledged as a specialised graphical tool within academic and professional domains rather than a complete UI platform. Strategies beyond solely employing tools like Plotly might be necessary for dynamic web applications. Given the complexities of managing extensive data sets and executing complicated algorithms, Python and R might face performance challenges. The primary design objectives of Python and R focus on data manipulation and analysis rather than addressing the rigorous performance requirements of contemporary web applications.

Analyses of systems mentioned in [Goo20; Alp20] show a dominant trend towards JavaScript and related web-centric technologies. Given this trend and specific project objectives, TypeScript has been chosen as the primary programming language. Integrating TypeScript with the Vue 3 framework, known for its component-based architecture, enhances modularity and extensibility. Examples showcasing Vue 3's capabilities can be found in Figure 16. Additionally, Vite has been selected as the build tool to streamline and optimise development. The combination of TypeScript, Vue 3, and Vite collectively represents the optimal solution for the project's scope.

Regarding data storage, information initially stored in Excel is converted to JSON format. This choice stems from JSON's lightweight properties and capability for straightforward machine interpretation. Moreover, its considerable compatibility with various modern programming languages and its superior data structuring capability position JSON as an ideal format for data exchange.

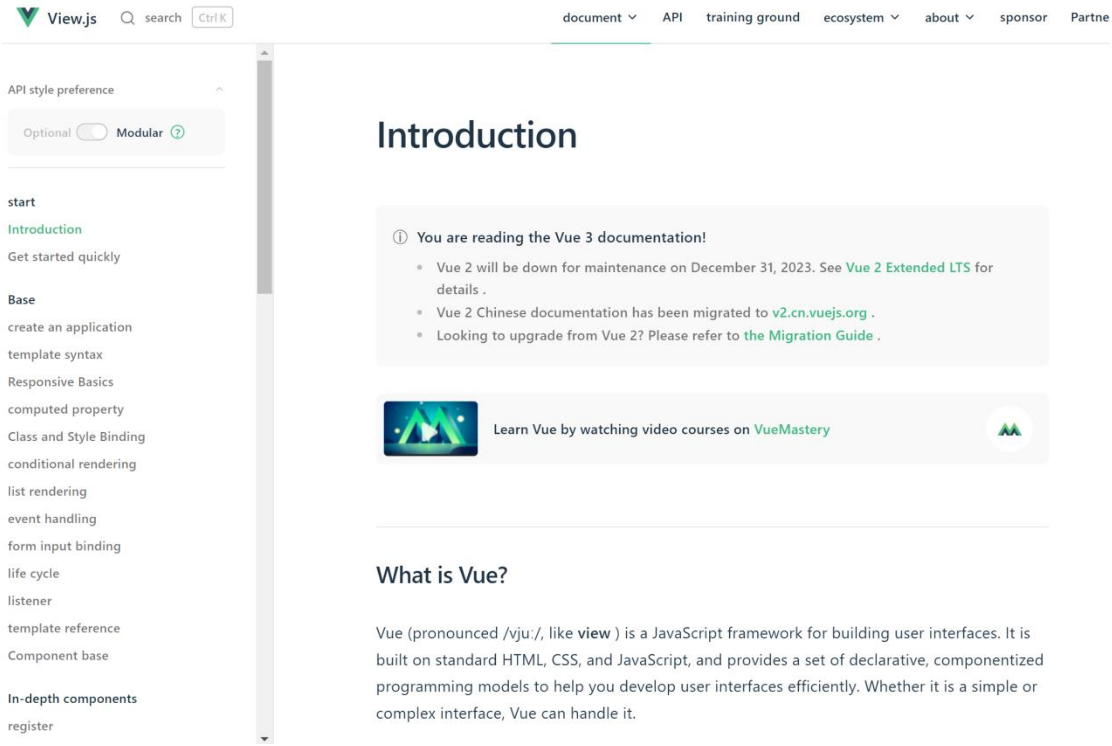


Figure 16. A screenshot of the homepage of Vue 3 showcasing the components included in the framework. Taken from: https://cn.vuejs.org/.

4.2.2 Libraries

In the diverse landscape of TypeScript visualization libraries, the most popular is ECharts.js. It is highly regarded for its comprehensive chart selection, ranging from simple graphical representations to more complex visual structures. A defining feature of ECharts.js is its modular design, which emphasizes efficiency and optimizes application performance. The library's API is designed to enhance the developer experience, offering straightforward integration and easy customization. Furthermore, ECharts.js provides interactive features, such as dynamic zooming capabilities, and is well-equipped to process multi-dimensional datasets.

Within the landscape of Vue 3-oriented libraries, Element Plus distinguishes itself by offering high-quality components optimally designed for efficient user interface development. Its modular architecture permits tailored integration, leading to improved application efficiency. The project selected Element Plus due to its adaptability and the strong backing from the Vue community.

In the domain of state management libraries, Pinia stands out due to its tailored design for Vue applications. Its seamless compatibility with Vue 3 and intuitive API facilitates precise data dynamics and state transitions management. These attributes determined this project's choice to utilize Pinia. Furthermore, using Element Plus and Pinia amplifies the development environment, effectively handling user interface components and state management intricacies.

4.2.3 Other Technologies

- **GitHub**

The project utilizes GitHub for source code backup and version control, as demonstrated in Figure 17. This approach leverages GitHub's interactive user interface to streamline committing changes. As a comprehensive version control system, GitHub supports the software development workflow, ensuring that all software versions are preserved throughout the project lifecycle. The complete code for this project is available on GitHub.The direct link to the repository is provided: https://github.com/AbibLi/StockVis.git

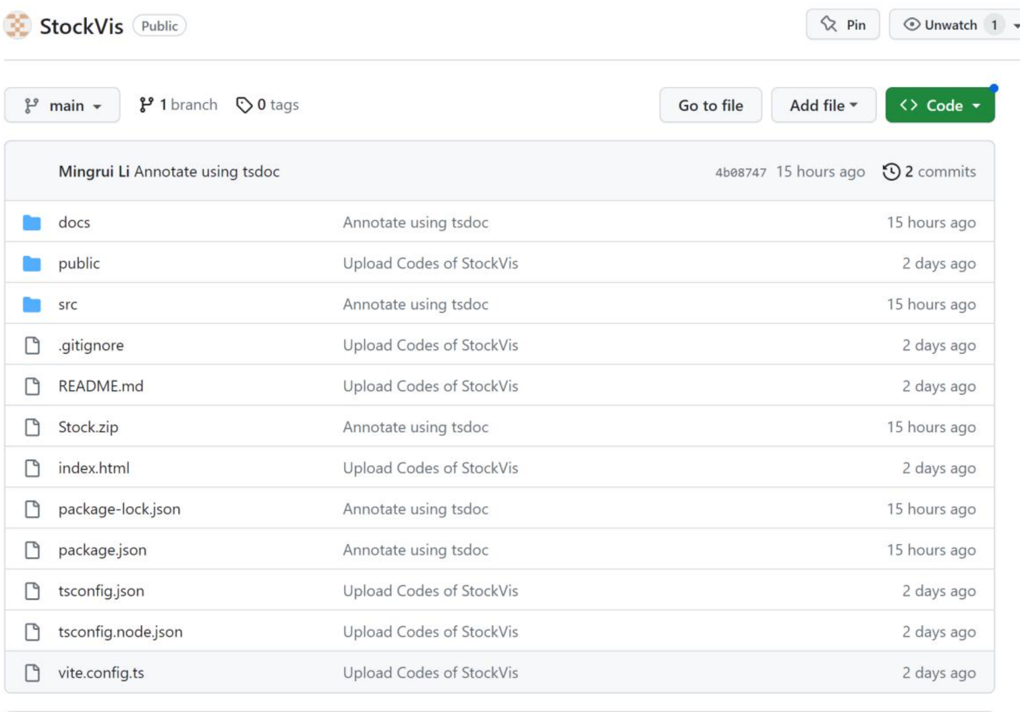


Figure 17. The screenshot displays the code uploaded to GitHub.

- **NPM**

The project utilises the Node Package Manager (NPM) to streamline and secure dependencies. Renowned for its vast library, NPM facilitates effortless installations through its command-line interface. In this Vue 3 project integrated with TypeScript, incorporating the Vue Composition API requires a npm install command within the pre-configured node.js environment. It is pivotal to highlight that the project's development environment is already optimised with node.js, ensuring efficient package management.

- **Visual Studio Code**

For the StockVis project, Visual Studio Code was employed for code compilation, as depicted in Figure 18. This platform has features supporting most web-based languages and offers downloadable extensions. Furthermore, it integrates inherent Git and console support, seamlessly collaborating with the project above and package management tools.

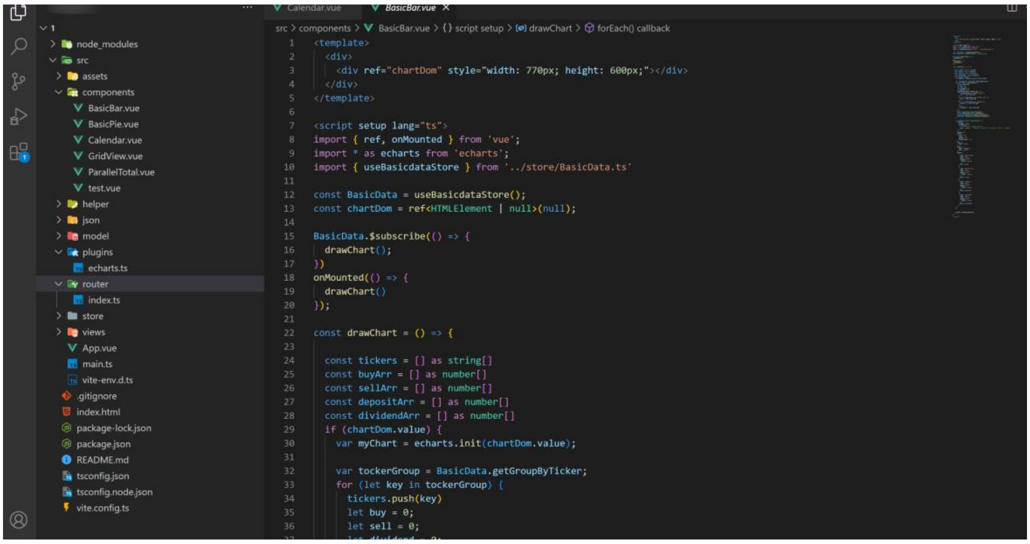


Figure 18. The screenshot of Visual Studio Code.

5 Project Plan and Timetable

For the successful execution of each phase, the project involved formulating a detailed plan, maintaining adherence to the software development lifecycle, and establishing tracking mechanisms. The associated project planning chart was designed in alignment with Bob's project guidelines [Lar11].

5.1 Project Plan

To ensure the smooth progression of the project, the project supervisor meticulously documented each weekly meeting and stored the records securely in

Google Docs. As Bob's project guidelines [Lar11] outlined, the primary objective of formulating a project plan was to ensure the timely completion of all critical tasks. Considering the project spanned over three months, each study indicated the expected completion date. Table 1, listing the projected completion dates (corresponding to each project meeting) and the actual dates achieved, provides a detailed overview of the project's progress.

Task	Expected Completion	Actual Completion	Additional Notes
1. Data preperation	July 12th	July 21th	Completed two versions of data and sorted out 2 data sets.
2. Explore Data	Aug 11th	Aug 11th	Use Tableau to explore two datasets while learning Typescript, Vue 3 and Echarts.
3. Import Data	Aug 16th	Aug 16th	Data is converted from .xlsx format to JSON format and imported.
4. Web-Based Visualization	Aug 16th	Aug 16th	
5. Parallel Coordinates View	Aug 16th	Aug 16th	Use Company-centric metadata.
6. Remove an Axis	Aug 18th	Aug 18th	
7. Stacked Bar Chart	Aug 16th	Aug 16th	Use Investment transaction data.
8. Company name or Ticker Queries	Aug 16th	Aug 16th	
9. Calendar View	Aug 16th	Aug 16th	Use Investment transaction data.
10. Pie Chart	Aug 16th	Aug 16th	Use Investment transaction data.
11. Integrate Pie charts into Calendar View	Aug 18th	Aug 18th	
12. Table	Aug 16th	Aug 16th	
13. Stacked Bar Chart	Aug 18th	Aug 18th	Use Company-centric metadata, to show Investment Performance.
14. Add categories and map them with color in Stacked Barchart	Aug 18th	Aug 18th	
15. Hot update and interaction	Aug 18th	Aug 18th	
16. Tooltip	Aug 22th	Aug 22th	
17. User Option and Color Map	Aug 18th	Aug 18th	
1. Focus elements are colored while the context appears in greyscale in parallel coordinates view	Aug 18th	Aug 18th	
2. Dynamic ranking in Stacked Bar Chart	Aug 18th	Aug 18th	Use Company-centric metadata to show the Investment Performance.
3. Sunburst chart	Aug 20th	Aug 20th	
4. Date filter in Calendar View	Aug 22th	Aug 22th	
5. Superpositioned calendar view	Aug 22th	Aug 22th	
6. Visual components are dynamically adjusted to fit the screen	Aug 23th	Aug 23th	

Table 1. The expected completion date for the application's core features and the time planning for its enhanced functions.

5.2 Development Approach

The Waterfall model has traditionally been widely adopted as a management strategy for numerous projects. This model delineates each project stage through a linear process, enabling individuals to discern the subsequent phase after completing the current one. Such an approach follows a strictly sequential order, meaning that once developers progress to later stages, it's often not feasible to go back and make changes to earlier steps. As described in [AA13], the Waterfall model typically encompasses five primary stages: requirement gathering, design, implementation, testing, and maintenance (see Figure 19). However, it's noteworthy that when project timelines are constrained, the testing phase often sees reduced time allocation, potentially compromising the final product's quality. Understanding earlier design decisions becomes crucial as developers advance toward the final stages. This necessity can make any late-stage code modifications challenging before the project concludes.

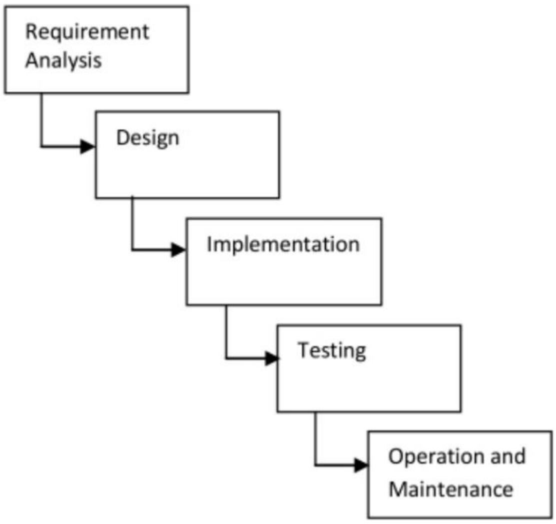


Figure 19: Five phases of the waterfall Module [AA13].

Contrary to the Waterfall model, which structures the project into continuous stages executed sequentially, the Agile methodology divides the project into multiple tasks, each accomplished over several short phases. When developers identify mistakes, this approach ensures they can make timely adjustments, addressing the quality and clarity of the development process concerns associated with the Waterfall model. The iterative nature of Agile development facilitates continuous communication. During regular discussions, project supervisors can suggest modifications to the core functionalities of the visualisation system based on primary requirements. Given these advantages, the Agile model was chosen as the developmental methodology for this project.

6 Project Design

As emphasised by Bob in his project guide [Lar11], the design phase is pivotal, driving developers to address challenges in implementation. An initial design was laid out in the early phase of developing the visualisation platform. However, due to the iterative nature of Agile methodologies, the design underwent several updates and optimisations. After multiple iterations, the final design fulfilled the project's core specifications and included advanced features.

This section presents an overview of the visualisation process, subsequently detailing the various design components integral to the project. Diagrams offered adhere to the guidelines stipulated in Bob's project manual [Lar11], each accompanied by a concise explanation of its associated system or functionality.

6.1 Pipeline in Visualisation

As depicted in Figure 20 [ERT+17], the visualisation pipeline delineates the visual representation process, transitioning from the source data to the final view. This pipeline can be adapted to cater to requirements pertinent to domain-specific data, thereby enabling deeper comprehension and polished visual representations.

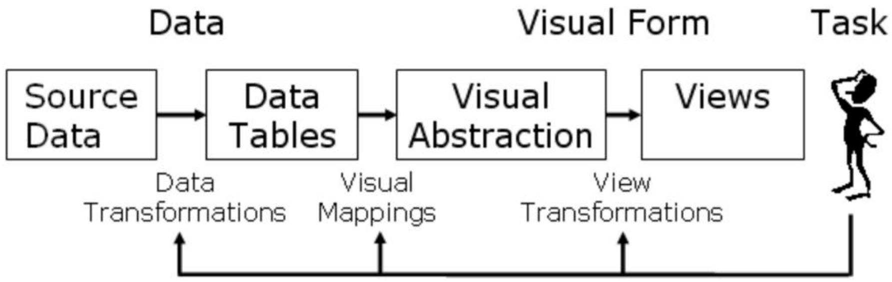


Figure 20. The visualization pipeline presented by Endert et al. [ERT+17] outlines the process of formulating visual data representations.

As depicted in Figure 21 [EFN12], the semantic interaction pipeline highlights how user interactions within spatial visualisations integrate into the computational framework of a visual analytic system, emphasising its adaptability and precision in processing user feedback.

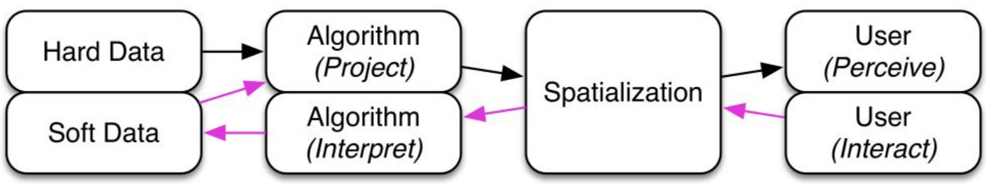


Figure 21. The semantic interaction pipeline [EFN12] integrates user interactions in spatial visualisations into the analytic system's computation.

6.2 Process Diagram

Figure 22 provides a detailed overview of the application's workflow. In the initial phase, an anonymous investor provided comprehensive investment records. The systematic compilation of these records produced two primary datasets. The data was transformed into Excel format, optimizing storage and facilitating future analyses before secure storage in Google Sheets. A designated tool was employed to convert the Excel (.xlsx) files to JSON format, enhancing efficiency and optimizing storage within the system. Utilizing Vue 3 as its foundational framework, the application integrated with Echarts.js. This collaboration facilitated the creation of advanced visualizations, including parallel coordinate plots and stacked bar charts. The advanced rendering capabilities of ECharts.js guarantee that these visualizations represent investment data accurately and adapt to diverse datasets dynamically. The application prioritizes user experience by ensuring that charts are responsive across varied screen dimensions.

Additionally, the system has introduced interactive features to the charts—such as company filtering, date selection, axis removal, and view switch—to improve data readability and cater to the distinctive needs of potential investors. While the process diagram provides a comprehensive overview of a user's interaction with the system, a broader range of subsystems exists. Subsequently, the 'System Diagram' will describe these complexities.

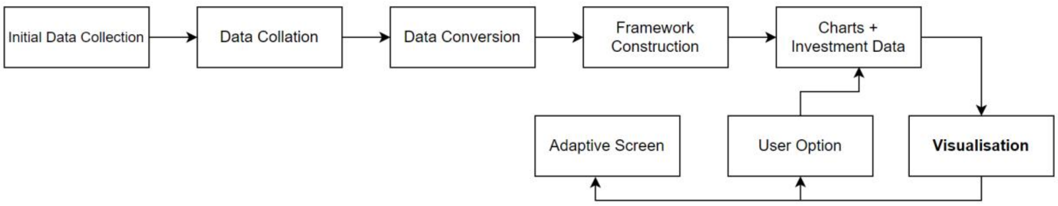


Figure 22. A process diagram produced for the Interactive web visualization tools.

6.3 System Diagram

Figure 23 demonstrates the system relationship diagram as part of the application design, illustrating various developed visualizations and their interactions. This software subsystem diagram presents potential users with an overview of the system's architecture, helping them comprehend how different application components collaborate. Additionally, each chart marks key attributes and functions, aiding in a better understanding of various elements in the code.

The detailed design of the StockVis system displayed in Figure 23 elaborates on each visualisation chart's functions, the represented data, and the interaction mechanisms among them. The design centres around a web interface consolidating various visualisation components, such as parallel coordinate plots, stacked bar charts, calendar views, sunburst diagrams, and data tables. These visualisation elements aim to present users with comprehensive and transparent information and offer an interactive data exploration method, making data interpretation more intuitive and insightful.

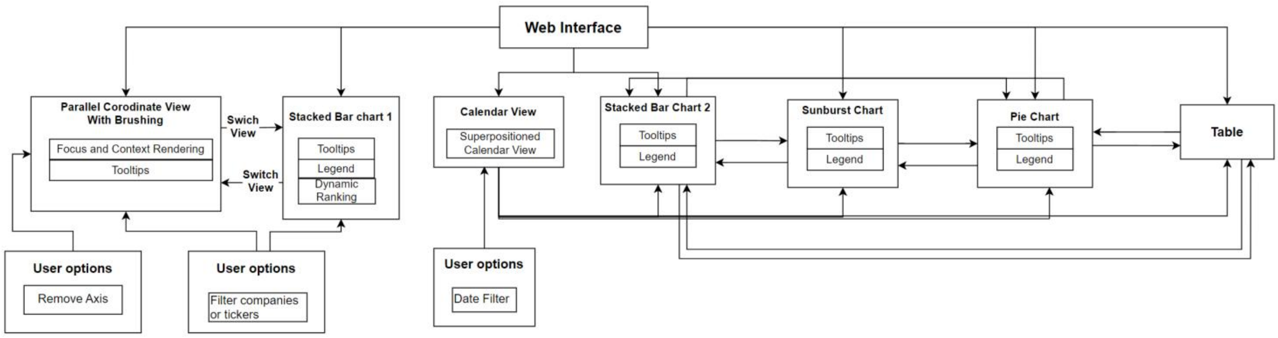


Figure 23. A visual representation of the system highlights the primary subsystems and their interactions.

In the design, the parallel coordinate plot primarily employs metadata associated with individual companies, providing potential investors with a comprehensive overview of investment data and trends. Furthermore, the system facilitates a seamless transition between the parallel coordinate plot and the stacked bar chart, visualising investment performance through designated controls. This interactivity, denoted by double arrows in the design, emphasises the two charts' shared data and communicative nature. The parallel coordinate plot permits filtering by company or excluding superfluous axes to assist users in obtaining the specific information they seek. Similarly, the stacked bar chart facilitates data filtering based on company names and Tickers.

The calendar view, sunburst diagram, pie chart, stacked bar chart displaying investment records and table all employ thorough investment data from anonymous investors. In the interactive design of these visual tools, users can pinpoint specific dates through the calendar view. Subsequently, the sunburst diagram, pie chart, stacked bar chart, and table dynamically adjust to represent the detailed transactional information corresponding to the selected date. Additionally, the system proposes a group of controls, allowing users to switch between these visualisations seamlessly. The system employs colour mapping with corresponding legends to enhance data readability when delineating diverse transaction actions or companies. The calendar view authorises users to select individual day data and data over extended periods. In such cases, the system presents a superpositioned calendar visual. The bidirectional arrows intuitively signify the interactive linkage among the sunburst diagram, pie chart, stacked bar chart, and table.

Overall, the visual representation of the StockVis system and user interaction are aptly depicted in Figure 23. It is evident from the illustration that users engage with various charts on the platform. Additionally, it displays how users engage with various visual charts within the application and emphasises the deliberate design decisions made to ensure efficient interactivity. Concerning the current design framework, considerable scope remains for subsequent optimisation and expansion. The following adjustments and refinements will systematically address evolving user requirements and feedback.

7 Implementation

This section provides a detailed exploration of the project's implementation process. Throughout the execution phase, web-based technologies were adopted, complemented by tools delineated in the technology selection section. Bob's project guideline [Lar11] describes an overview of the essential steps and knowledge required for reconstructing the visualisation platform. For clarity and illustration, screenshots documenting the implementation of each element listed in Table 1 are incorporated.

Figure 24 provides a detailed view of the application through specific screenshots. Within these, Figure 24(a) demonstrates the parallel coordinates plot, which has been filtered to display stocks SBNY, BYND, and HASI while excluding the purchase and sale dates. In this visualization, each axis corresponds to dimensions centred around company-specific metadata. By highlighting the three companies mentioned above and rendering the background transparent, users can identify data points more precisely. Meanwhile, Figure 24(b) displays a screenshot of the calendar visualization integrated with its associated charts. This visualization shows the investment transactions for July 2022, giving a succinct overview of daily transactions via embedded pie charts. These pie charts enable users to zoom into specific dates for a granular view of transactions. Unique colours discern stocks, each paired with an appropriate legend. Notably, variations in the segments of the pie charts reflect transaction volumes, providing users with an intuitive grasp of daily trades and their associated values.

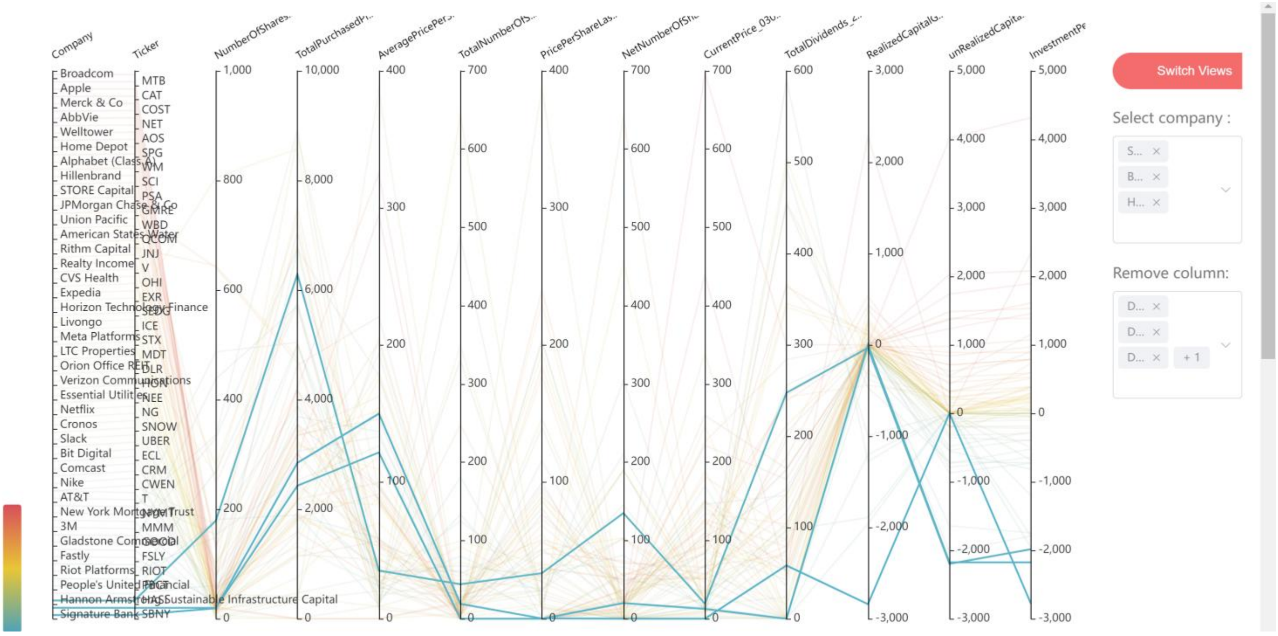


Figure 24 (a): The screenshot displays a parallel coordinates plot for stocks SBNY, BYND, and HASI, with purchase and sale dates excluded.

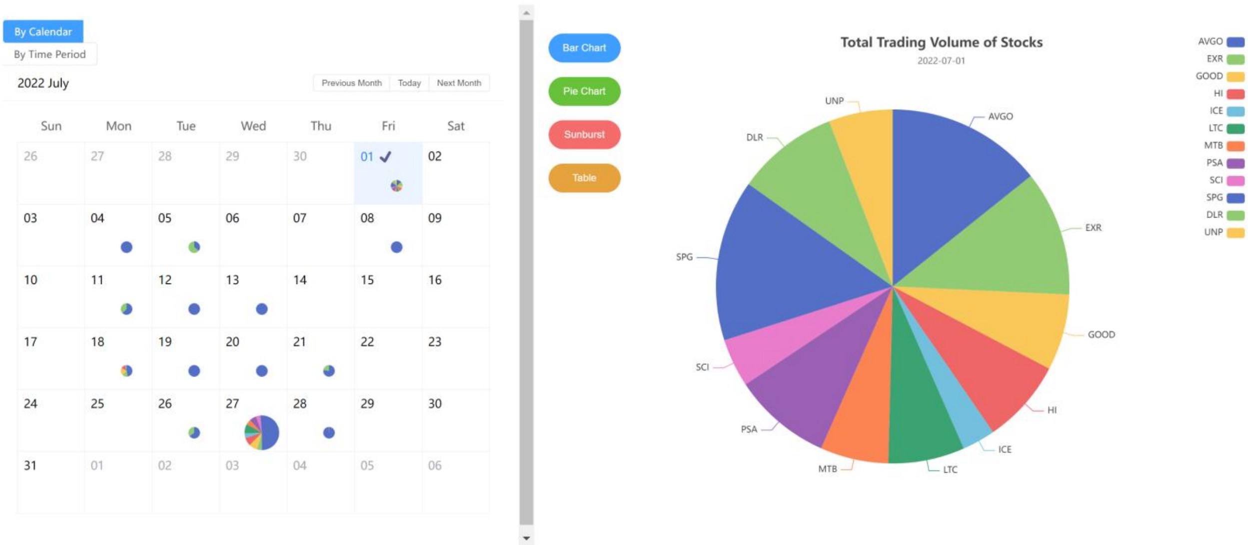


Figure 24 (b): The screenshot depicts a calendar visualization for July 2022 investment transactions, highlighted by daily embedded pie charts. The enlarged pie chart specifically represents transactions on July 1, 2022.

7.1 Basic Implementation

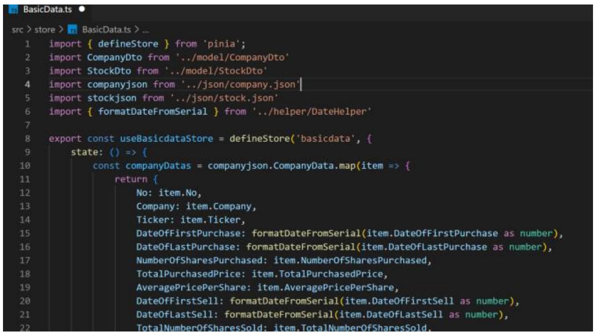
In the foundational implementation phase of this project, all core functionalities have been successfully covered. Each constituent element is crucial, ensuring the project's initial objectives are accurately realized.

7.1.1 Data Management Process

In the initial phase of the study, a primary emphasis was on the preparation, exploration, and importing of data. Comprehensive investment data, sourced from an anonymous investor, was collated into two distinct datasets: the overall investment dataset and a company-centred metadata set. These datasets were compiled into Excel formats to ease data entry and preliminary analysis. The visualisation software, specifically Tableau, was initially employed to explore the data. Subsequently, these datasets were transformed into the JSON format, facilitating their integration within the system. Additionally, the datasets were uploaded to Google Sheets to augment data accessibility and promote collaborative sharing.

One significant challenge faced was understanding and interpreting the data procured from the anonymous investor. Considering that the Trading 212 platform only provides cumulative data monthly, the data directly downloaded by the anonymous investor was the primary source for this study. The Investment Transaction Data presents all transactional data systematically arranged by transaction dates. This dataset incorporates columns such as 'Trading Account', 'Transaction Type', 'Transaction Date', 'Ticker', and 'Transaction Amount', providing detailed information for the investor's trading activity. Furthermore, Company-Centred Metadata displays the "Company-centred Metadata" table, organised based on the initial purchase date of the stocks. It details columns including the stocks' first purchase date, quantity purchased, total purchase price, average buying price, and additional data. This table aggregates all the stocks acquired by the investor, offering users a comprehensive view of the investment portfolio. Consequently, the dataset equips researchers with the foundation to analyse investors' strategic choices and investment patterns deeply.

After successfully converting the data into JSON format, the study undertook technical preparations that involved importing necessary modules, data models, JSON files, and auxiliary functions, as seen in Figure 25. The project utilised the "defineStore" functionality of the Pinia library to integrate the two JSON datasets into the system. The study first defined the system's state and then extracted vital information from both datasets. Subsequently, this information was organised into a data model tailored to the project's specifications. A point of emphasis was standardising data containing transaction dates to cater to impending visualisation needs.

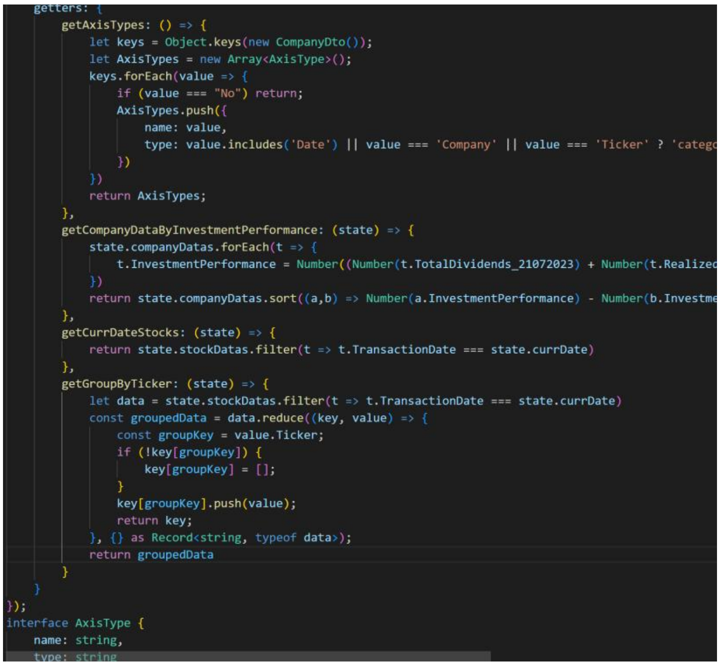


```
BasicData.ts
src > store > BasicData.ts > ...
import { defineStore } from 'pinia';
import CompanyDto from '../model/CompanyDto'
import StockDto from '../model/StockDto'
import companyjson from '../json/company.json'
import stockjson from '../json/stock.json'
import { formatDateFromSerial } from '../helper/DateHelper'

export const useBasicdataStore = defineStore('basicdata', {
    state: () => {
        const companyDatas = companyjson.CompanyData.map(item => {
            return {
                No: item.No,
                Company: item.Company,
                Ticker: item.Ticker,
                DateOfFirstPurchase: formatDateFromSerial(item.DateOfFirstPurchase as number),
                DateOfLastPurchase: formatDateFromSerial(item.DateOfLastPurchase as number),
                NumberOfSharesPurchased: item.NumberOfSharesPurchased,
                TotalPurchasedPrice: item.TotalPurchasedPrice,
                AveragePricePerShare: item.AveragePricePerShare,
                DateOfFirstSell: formatDateFromSerial(item.DateOfFirstSell as number),
                DateOfLastSell: formatDateFromSerial(item.DateOfLastSell as number),
                TotalNumberOfSharesSold: item.TotalNumberOfSharesSold,
```

Figure 25. The screenshot illustrates the integration of necessary modules, data models, JSON files, auxiliary functions, and the utilisation of the 'defineStore' functionality in the project.

In the data acquisition phase, a series of "getters" functions (seen in Figure 26) were delineated within the project to address specific research requisites. Among these functions are "getAxisTypes", tasked with extracting columnar attributes from the metadata set; "getCompanyDataByInvestmentPerformance", which quantitatively evaluates and hierarchically ranks the investment performance of individual companies; "getCurrentDateStocks", devised to acquire stock data contingent upon designated dates; and "getGroupByTicker", facilitating the categorisation of data based on stock ticker symbols. An interface, denominated as "AxisType", was methodically instituted to process foundational stock data about companies. These "getter" functions, formulated in congruence with the precise exigencies of the research, are in diverse segments of the application for the procurement of either primary or derivative data.



```
    getters: {
        getAxisTypes: () => {
            let keys = Object.keys(new CompanyDto());
            let AxisTypes = new Array<AxisType>();
            keys.forEach(value => {
                if (value === "No") return;
                AxisTypes.push({
                    name: value,
                    type: value.includes('Date') || value === 'Company' || value === 'Ticker' ? 'catego
                })
            })
            return AxisTypes;
        },
        getCompanyDataByInvestmentPerformance: (state) => {
            state.companyDatas.forEach(t => {
                t.InvestmentPerformance = Number((Number(t.TotalDividends_21072023) + Number(t.Realized
            })
            return state.companyDatas.sort((a,b) => Number(a.InvestmentPerformance) - Number(b.Investme
        },
        getCurrDateStocks: (state) => {
            return state.stockDatas.filter(t => t.TransactionDate === state.currDate)
        },
        getGroupByTicker: (state) => {
            let data = state.stockDatas.filter(t => t.TransactionDate === state.currDate)
            const groupedData = data.reduce((key, value) => {
                const groupKey = value.Ticker;
                if (!key[groupKey]) {
                    key[groupKey] = [];
                }
                key[groupKey].push(value);
                return key;
            }, {} as Record<string, typeof data>);
            return groupedData
        }
    }
});
interface AxisType {
    name: string,
    type: string
```

Figure 26. The realization of "getters" function.

7.1.2 Web-Based Visualisation

One of the salient features of this research project is its cross-browser compatibility, guaranteeing smooth functionality across different modern web browsers. Additionally, the system has now been configured on a server and can be accessed online at http://47.94.152.99:6688/. Furthermore, the complete code for this project is available on GitHub.The direct link to the repository is provided: https://github.com/AbibLi/StockVis.git

From a design perspective, the upper section of this system primarily features two charts: a parallel coordinate plot and a stacked bar chart. Both are based on company-centred metadata. Users can toggle between views with the 'Switch Views' button.Figure 27(a) displays the parallel coordinate chart of the system's upper section. Within this chart, each axis represents a dimension from the dataset, encompassing aspects such as company name, ticker, initial purchase time, and the average price of each stock. Through colour mapping, where different colours represent distinct companies, the system offers users a comprehensive overview of the data and facilitates the identification and comparison of multi-dimensional information. Users can discern underlying patterns and relationships by observing the data distribution across different axes.Figure 27(b) illustrates the stacked bar chart in the system's upper section, purposefully designed to visually convey the performance of all stocks held by the investor, subsequently ranked based on these performances. This chart includes two notable legends delineated by unique colours, symbolising capital gains and dividends. With this visual design, the diagram displays each stock's investment performance, thereby assisting users in effective portfolio management.

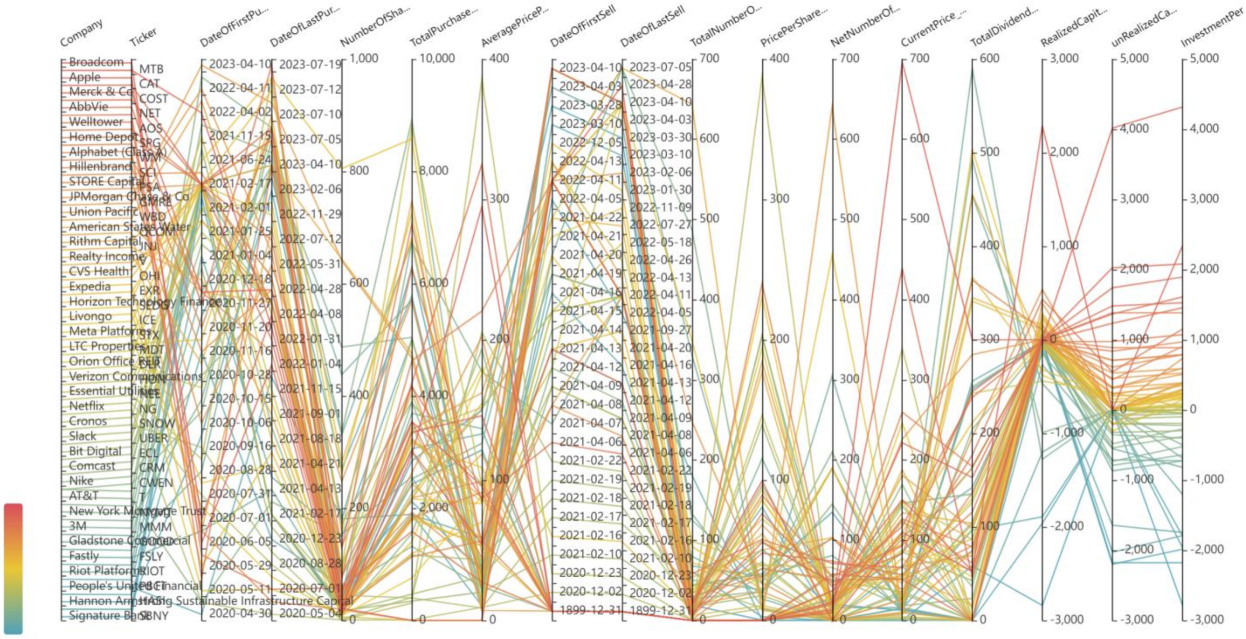


Figure 27 (a): The screen shot of the parallel coordinate chart of the system's upper section.

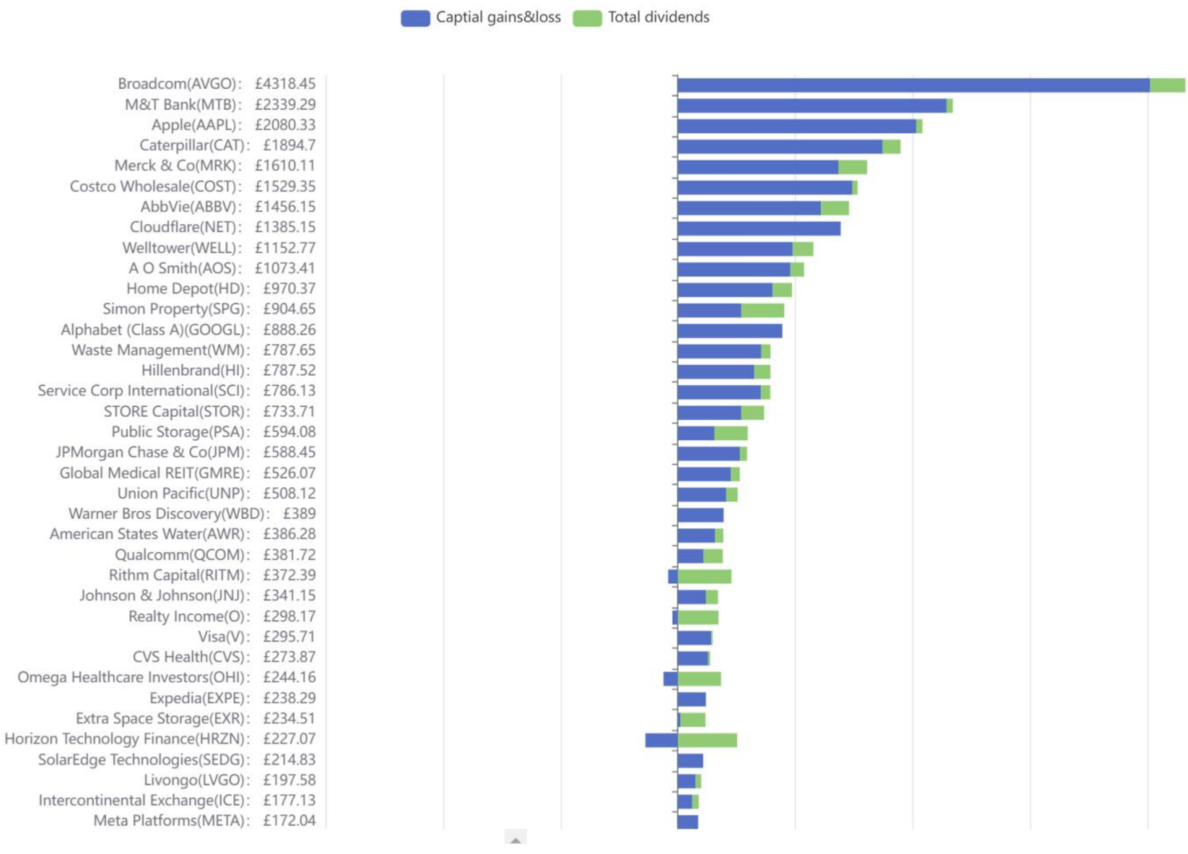


Figure 27(b): The screen shot of the stacked bar chart of the system's upper section.

Figure 27(c) displays that the system's lower section comprises various charts, including a calendar view, pie chart, sunburst chart, and table, all based on the investor's complete investment data. Within the calendar chart, each date-specific pie chart provides a detailed representation of that day's trading activities, encapsulating the trading companies and the respective transaction amounts. A more considerable transaction amount corresponds to the pie chart's more significant sector angle and radius. The system features four distinct controls, each representing a unique view. Users can click these controls to enlarge trading data for specific dates and seamlessly toggle between views to detailed analysis. Each traded stock is differentiated by a unique colour in the enlarged pie chart, and its Ticker is annotated within the corresponding sector. An accompanying legend is provided beside the pie chart to further aid user identification, displaying these ticker symbols. Such a design ensures that users can rapidly identify the trading volume for each stock, providing a clear overview of the distribution and relative proportions of daily trading activities.

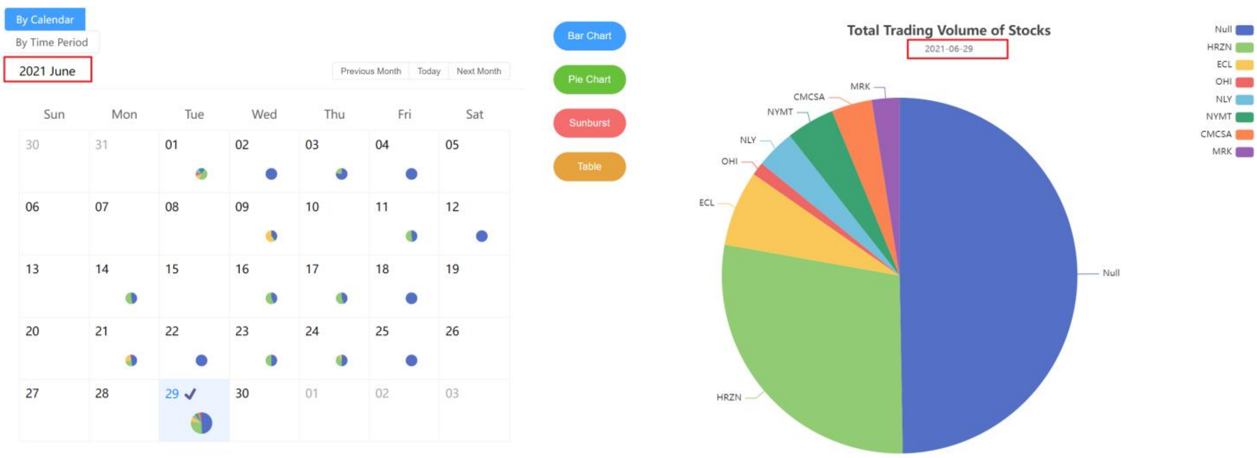


Figure 27 (c): The screen shot of the lower part of the system including calendar view and pie chart.

7.1.3 Parallel Coordinates View

The system utilises a parallel coordinates plot to visualise the company-centred metadata. The primary reason for selecting this chart type is its distinguished capacity to represent multi-dimensional data. This metadata set emphasises several critical attributes of the invested companies, such as the ticker, the average price of stocks, and the current prices. The parallel coordinates plot clearly and simultaneously displays the aforementioned multi-dimensional information.Additionally, the parallel coordinates chart design enables users to directly observe and analyse relationships among various dimensions, aiding in identifying data patterns and trends. The integrated interactive features allow users to tailor their viewing experience. For instance, they can remove unnecessary measurements, highlight specific companies or tickers, or select a data range on a designated axis to focus on their areas of interest. Such design elements enhance the adaptability of data analysis, deepening user comprehension.

The parallel coordinates plot is primarily constructed using the ECharts.js library. This library provides interactive capabilities for canvas-based elements, delivering a detailed representation of company-centric metadata. The system extracts all stock investment data, centred around companies, for anonymous investors through the "useBasicdataStore()" function. Moreover, The "el-container" component effectively integrates the primary content with the sidebar.

For configuring the plot, the system retrieves axis names using the "AxisTypes.map()" method, which subsequently forms the "schema" array for specific axis settings. Data for each company is compiled based on the chosen axes, denoted as "curAxisSelects", and this data is then integrated into the two-dimensional "BindData" array.

In the interaction design, two "<el-select>" components from Element were employed to cater to company data filtering and axis selection requirements. When users opt for different companies, the "drawChart" function is activated by the identifier bound to "v-model='comSelects'". The variable "comSelects" is designated to store companies selected within the filter box, facilitating the company selection process. Similarly, upon updating the axis selection, the system invokes the "drawParallel" function, removing unnecessary axes. The specifics of the code implementation are demonstrated in Figure 28. Correspondingly, as illustrated in Figure 29 demonstrates that the parallel coordinates chart offers a clear view detailing the associated data by filtering for a specific company and removing extraneous axes.

```
const drawParallel = () => {
    let curAxisSelects = AxisTypes.map(t => t.name)
    if (axisSelects.value.length > 0) {
        curAxisSelects = curAxisSelects.filter(t => !axisSelects.value.includes(t))
    }
    const schema: any = [];
    let index = 0;
    AxisTypes.forEach(item => {
        if (curAxisSelects.includes(item.name)) {
            if (item.name.includes("Date")) {
                schema.push({
                    dim: index++,
                    name: item.name,
                    type: item.type,
                    data: Array.from(new Set(CompanyData.map(t => t[item.name as keyof CompanyDto]))).sort()
                })
            }
            else {
                schema.push({
                    dim: index++,
                    name: item.name,
                    type: item.type,
                })
            }
        }
    })
    let BindData: any = []
```

Figure 28. Part of the code for parallel coordinate chart implementation

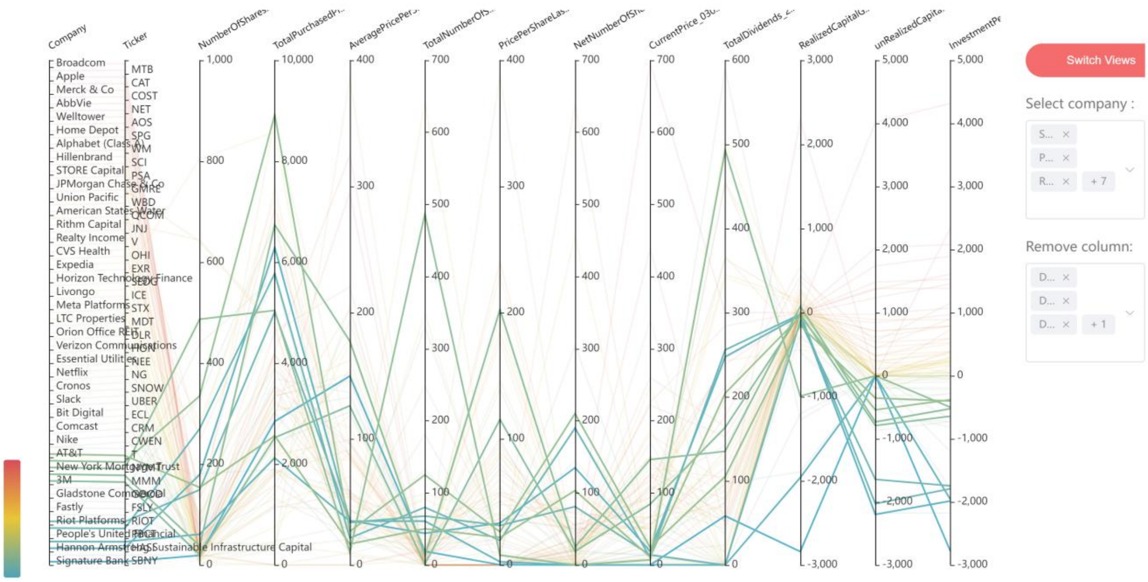


Figure 29. The screenshot of the parallel coordinate chart illustrates the filtering of stocks such as T, VICI, and NYMT, with purchase and sale dates removed for clarity.

7.1.4 Stacked Bar Chart I

Figure 30 presents a stacked bar chart depicting the investment performance of various companies. In the displayed chart, the X-axis represents the amount of investment performance, while the Y-axis indicates the invested companies. Notably, the investment performance consists of both capital gains and dividends. Two separate colours in the chart distinguish these categories, with their meanings explicitly annotated in the legend. Through interactive controls, users can dynamically view stock performance by category or filter specific companies for in-depth analysis of their investment returns. The chart updates in response to user selections and dynamically ranks the investment data. The primary aim of this chart is to provide users with an intuitive insight into the investment performance of each stock and facilitate a deep analysis of the stocks' performance across different categories and in aggregate.

Analogous to the parallel coordinate chart, this visualisation is rendered using the Echarts.js library. The system retrieves the required data from the database using the "useBasicdataStore()" function.

A constant documented "drawStackedBar" was designated for graphical manipulation to construct this stacked bar chart. The system integrated the Element UI's <el-select> component to enhance user interactivity, allowing users to filter data for specific companies conveniently. Within the chart's configuration, the "dividend" and "capital gain and loss" were explicitly defined as legend indicators through the series attribute. These legends provide a distinct visual cue for data categorisation using varied colours and possess interactive capabilities: Users can click on these legend indicators to selectively display or conceal the corresponding data segments. Additionally, "viewchange()" and "drawchart()" functions were employed to facilitate the switch between the parallel coordinate chart and the stacked bar chart.

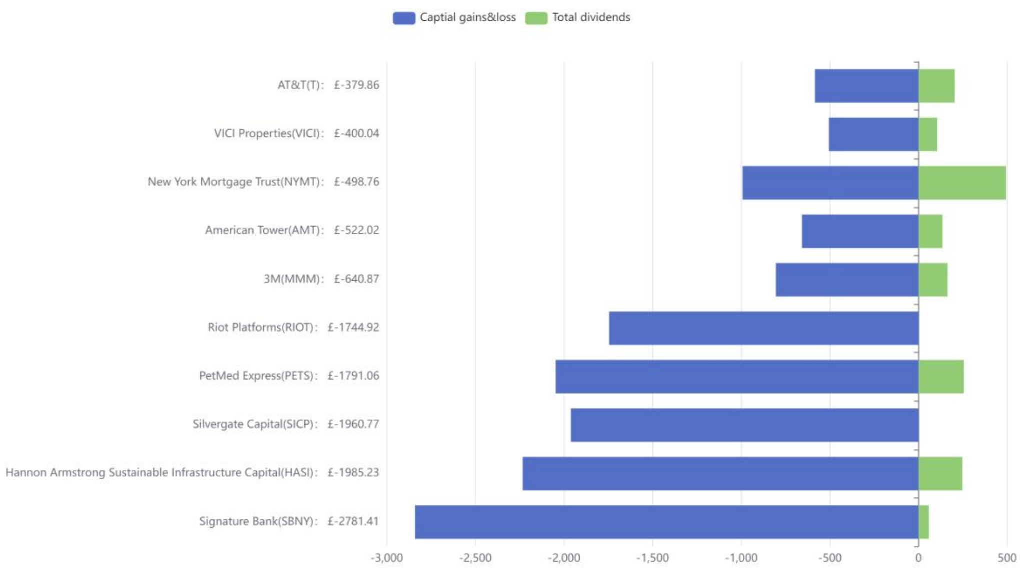


Figure 30. A screenshot of the investment performance for stocks such as T, VICI, and NYMT.

7.1.5 Calendar View

The application chose the el-calendar component from the Element Plus library to showcase a fully-featured calendar interface. This design facilitates date-specific data retrieval by users and incorporates daily enhancements to enrich user experience. A calendar chart was chosen to provide a comprehensive visual representation of an investor's entire trading activities, with each date cell embedding a pie chart,as depicted in Figure 31. Within these pie charts, different colours distinguish the separate companies engaged in transactions on the respective days, with the segment occupied by each company reflective of its transaction volume for that day. Notably, within each month, the radius of the pie charts correlates directly with the daily transaction volume. Specifically, days with larger transactional volumes display a expanded chart radius. The combined use of calendar and pie charts visually represents daily trading volumes and the count of engaged corporations. Additionally, it provides an overview of monthly trading activities.

Upon date selection, the system promptly verifies the presence of matching stock trade data. The corresponding pie chart is displayed on the calendar interface if detected. The application employs the “GroupByTicker” function to streamline data presentation, ensuring transactions are categorized by their stock code, or 'Ticker', and cumulatively computed.

Taking advantage of Vue 3's capabilities, responsive variables, such as “currentDate” and “echartsContainers”, are declared using ref. All stock data resides within the “stockDatas” array, and the reduce method, in conjunction with object data structures, categorizes data by stock code.

Considering Vue's environment, it's crucial to initialize ECharts post-DOM update. The "nextTick" function ensures this precision. Bridging the interaction between the calendar and the pie chart presented a significant challenge, mandating accurate and real-time data display when merging the Element Plus calendar component with ECharts pie charts. Notably, the calendar component of Element Plus offers a date-cell slot, permitting daily content customization. Utilizing this feature, ECharts pie charts are seamlessly incorporated into calendar cells. The "shouldShowChart" function determines the display eligibility of the pie chart for a chosen date based on its relevance to the current month. Lastly, to guard against potential memory leak issues, all ECharts instances are destroyed using the "onBeforeUnmount()" lifecycle hook before the component is unmounted.

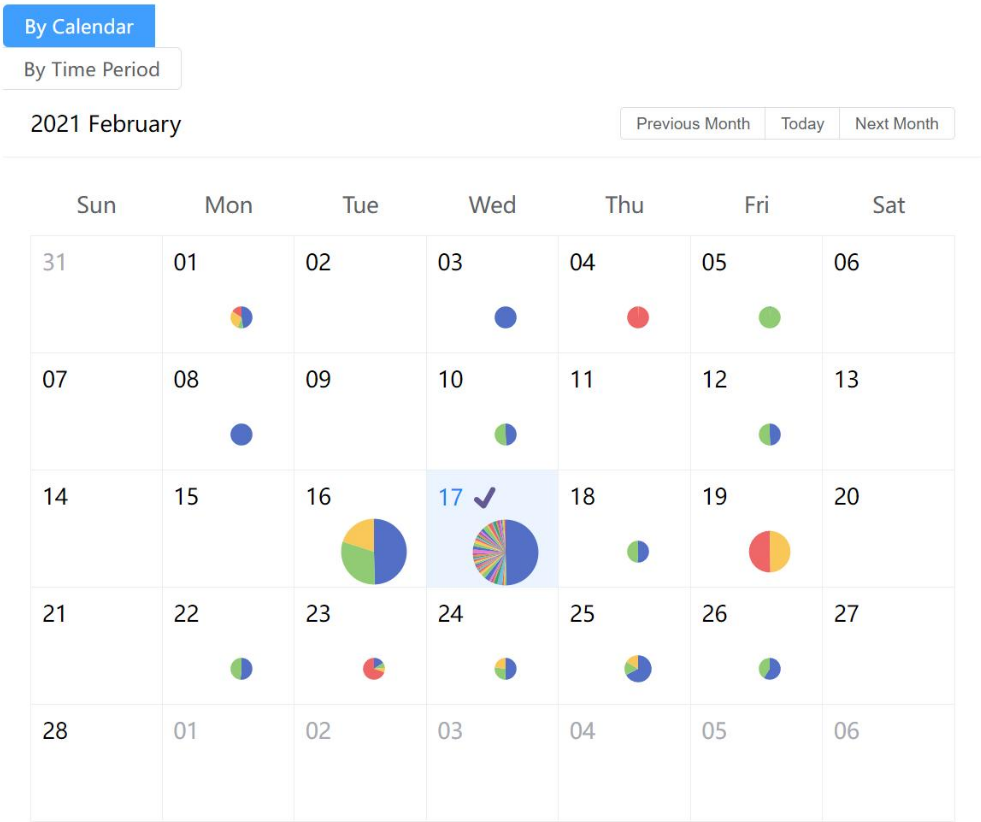


Figure 31. The screen shot of Feb, 2021 trading data calendar: Pie chart radius indicates transaction volume; segments show company trading proportions.

7.1.6 Stacked Bar Chart II

The current study developed the second stacked bar chart, as shown in Figure 32, to display investment data from anonymous investors. In this chart, the x-axis represents the monetary amount in this chart, while the y-axis delineates various Tickers. The chart employs distinct colours representing the four pivotal transactional categories: market purchases, sales, dividends, and deposits. Meanwhile, An accompanying legend is provided, clearly indicating the categories associated with each colour. This colour differentiation, supported by the legend, allows for a swift interpretation of transaction types and their corresponding monetary values for each stock on specific days. Significantly, the chart is closely linked to a calendar graphic; selecting a specific date within the calendar prompts the bar chart to update, showcasing the transaction data for the chosen date.

To facilitate this visualisation, the system incorporated the Echarts.js library. A bespoke "drawChart" function generated this stacked bar chart. For data management, arrays named "tickers", "buyArr", and "sellArr" were implemented to house processed data. By leveraging the "BasicData.getGroupByTicker" method, data are organised by stock codes and relevant transactional data are automatically computed.

A primary challenge during this process was ensuring the fidelity of data arrays, particularly "buyArr" and "sellArr", to the underlying transaction information. To address this and ensure real-time responsiveness to updates in BasicData, the $subscribe method was introduced. This method actively tracks data modifications, triggering the drawChart function to refresh the chart upon any changes. Within the drawChart process, data from BasicData are first retrieved and subsequently categorised by stock code. The system then aggregates the total for each transaction type and stores the data in appropriate categories.

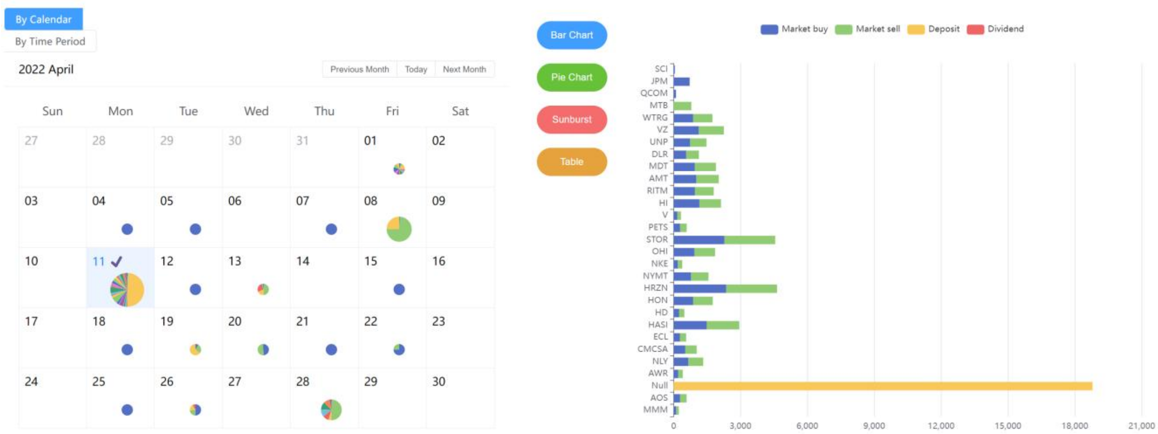


Figure 32. The screenshot of stock trading data for 11/04/2022, detailing the specific transaction actions and their respective monetary values for each stock.

7.1.7 Pie Chart

Figure 33 presents a pie chart illustrating the transaction volumes of various stocks within the investment data. Each stock is distinctly represented in the chart by a colour, with a corresponding legend provided for identification. The pie chart segments are proportioned based on the total transaction amount of individual companies. Companies with larger transaction volumes occupy a more significant portion of the pie chart. When users hover over a specific section, the system instantly provides information about the stock's name, aggregated transaction amount, and total transactions. This design provides users with an intuitive means to visualize the relative transaction volumes of individual stocks within a specified timeframe.

This pie chart strongly correlates with the calendar graphic, similar to the stacked bar chart discussed earlier. When a user selects a date within the calendar, the pie chart dynamically updates to reflect the transaction data pertinent to that date.

In terms of its creation, the methodology employed for the pie chart closely mirrors that of the stacked bar chart. However, navigating through the entire dataset to categorise based on stock identifiers, followed by summarising the total transaction amounts and counts for each stock, undeniably constituted the most challenging part of this endeavour.

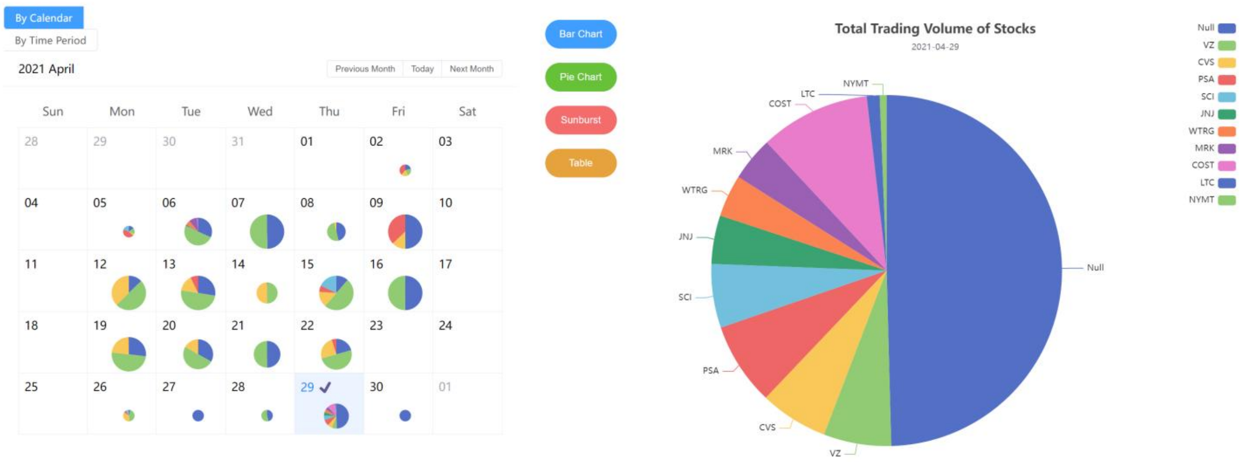


Figure 33. The screen shot of stock transaction data for 29/04/2021, showing the proportion of trading volume for each stock.

7.1.8 Table

As illustrated in Figure 34, the diagram showcases a table detailing specific investment data. The table encompasses the entirety of the transactional information, including elements such as Account, Ticker, Company Name, Action, and Total Transaction Amount. There exists a strong correlation between this table and the calendar graphic. Specifically, when users select a particular date within the calendar, the table automatically updates, presenting the detailed transaction data pertinent to the chosen date.

The program retrieves the stock transaction data for the current date from the BasicData storage using the 'useBasicdataStore' function and subsequently binds this data to the 'el-table' component. When there are alterations in the data within the BasicData storage, the '$subscribe' function is invoked, leading to the refreshment of the table's data. Among the implementation challenges, ensuring data reactivity and effective binding with the 'el-table' component is the most demanding.

To enhance user accessibility to pivotal data, the columns "Account Number" and "Ticker" are configured as fixed columns. This configuration ensures that these essential details persist in visibility, irrespective of table scrolling.

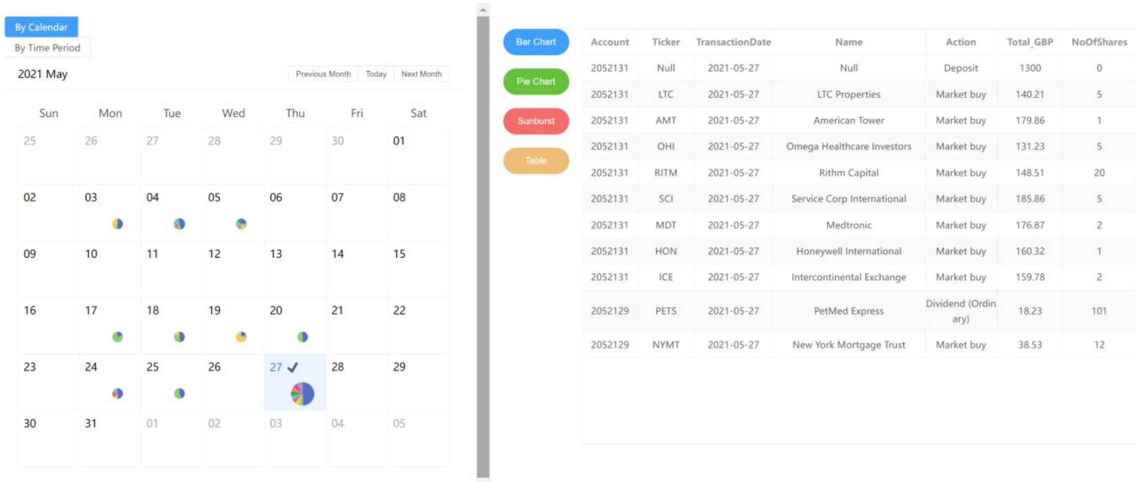


Figure 34. The screen shot of stock transaction data for 27/05/2022, showing the detailed transaction.

7.1.9 User Option

Figure 35 displays four view-switching buttons on the user interface. These buttons enhance the data analysis experience by offering multiple viewing options, allowing users to examine investment data from various perspectives. Specifically, the options provided include:

- Stacked Bar Chart: Enables data visualization in a stacked bar chart;
- Pie Chart: Enables data visualization in a pie chart;
- Sunburst Chart: Enables data visualization in a Sunburst chart;
- Table: Enables data visualization in a table;

Such a design approach has facilitated the creation of an efficient, intuitive, and comprehensive data analysis interface for users.To further demonstrate the interactive nature of these user options, a video demonstration is accessible through the following link: https://www.youtube.com/watch?v=UZ60MKiWuRc

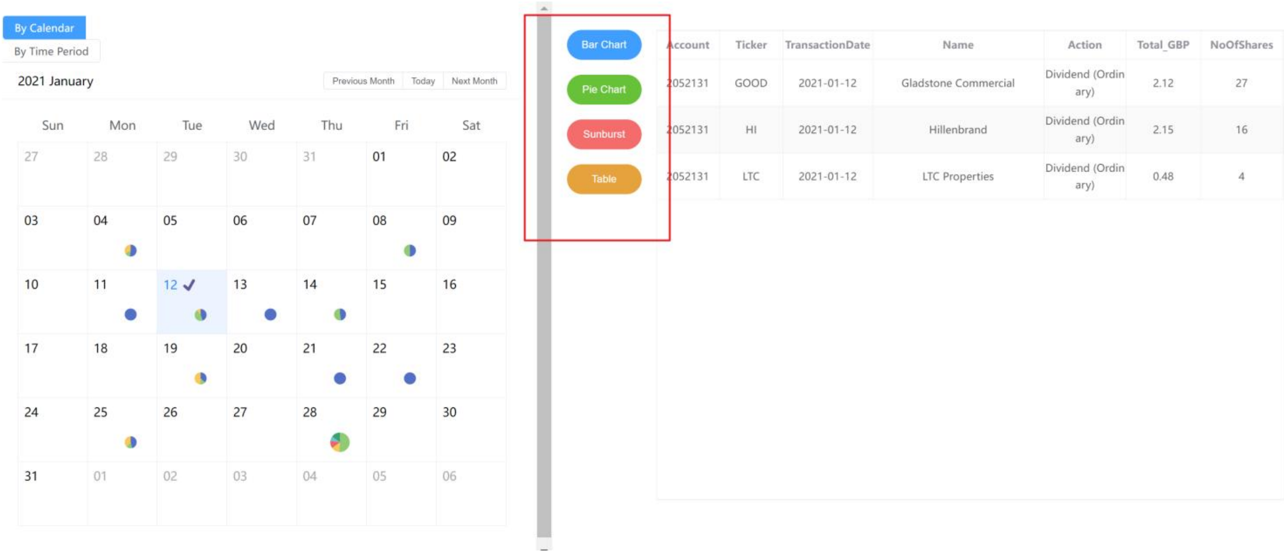


Figure 35. The screen shot of four view-switching buttons.

To seamlessly implement the switching between the three views, the routing mechanism within Vue 3 was employed. Within the modern context of engineering-centric front-end development, particularly for Vue projects, routing enables the effortless rendering and navigation of multiple pages using a single HTML file on the same platform. This routing mechanism provides efficient navigation management for single-page applications (SPA) and enhances the user experience's consistency and fluidity.

Figure 36 displays the specific code snippets related to routing configuration, while Figure 37 illustrates the practical implementation of routing.

Figure 36. The screen shot of the specific code snippets related to routing configuration.

```
        <RouterLink to="/Home/BasicBar"><el-button type="primary" round size='large' style="width: 100px
            Chart</el-button>
        </RouterLink>
      </li>
      <li>
        <RouterLink to="/Home/BasicPie"><el-button type="success" round size='large' style="width: 100px
            Chart</el-button>
        </RouterLink>
      </li>
      <li>
        <RouterLink to= "><el-button type="warning" round size='large'
            style="width: 100px;">Table</el-button>
        </RouterLink>
      </li>
    </ul>
  </el-aside>
  <el-main id="ccontent">
    <router-view></router-view>
```

Figure 37. The Screen shot of the code for the specific implementation of the function.

7.1.10 Tool

The system incorporates a series of display tools to enhance the readability and comprehensibility of the data. For instance, detailed information is revealed in the stacked bar chart when the user hovers over a specific bar, given the presence of distinct legends for different data categories. This display feature has been integrated across all charts within the system to ensure a thorough understanding of the presented data. Figure 38 provides an illustrative example of this functionality.

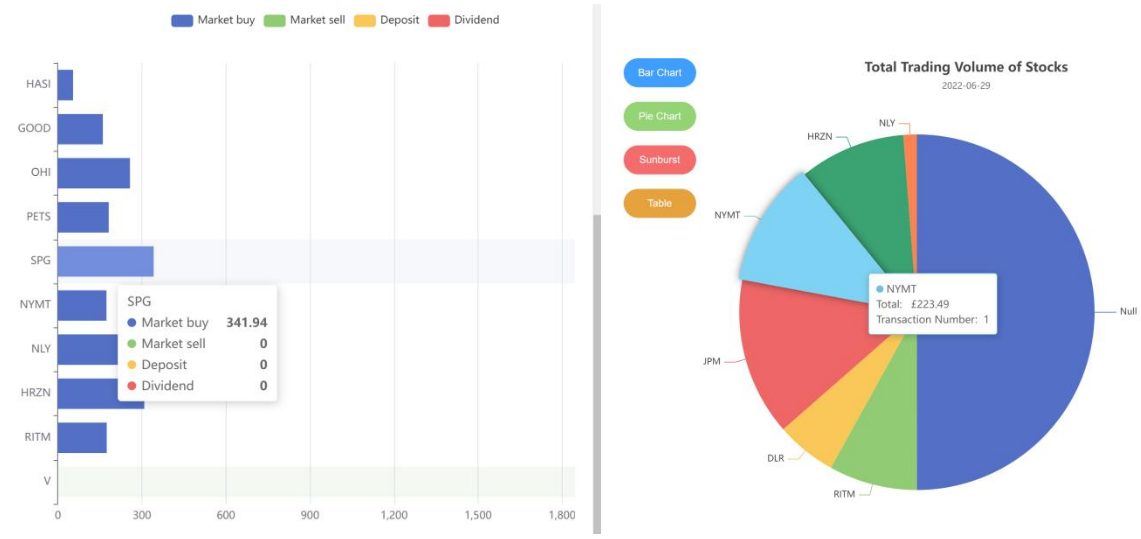


Figure 38. The screen shot of showing tool.

7.2 Enhancements

All enhancements were built upon the foundation of existing work to refine the project further and enhance its functionality.

7.2.1 Focus-Context Visualization Approach in Parallel Coordinates

In situations involving intricate data visualisation, especially when handling multivariate data from various companies and dimensions, delivering clarity for the viewer is crucial. Figure 39 demonstrates the utility of providing clarity, as HASI and

PETS are distinguishable from the other entities. The focus-context visualisation technique was integrated into the parallel coordinates plot to address the challenges posed by such multifaceted visual displays.

Conventional visualisation methods risk overwhelming the user with abundant overlapping data points. By employing the focus-context approach, the system highlights specific entities, like HASI and PETS, through colour differentiation. Simultaneously, to avoid overwhelming the viewer with too much simultaneous information, the non-essential or non-selected data is rendered in grayscale, allowing it to fade into the background. The technique is designed to direct users' attention specifically to selected entities, facilitating clearer comprehension and differentiation in information-rich environments.

Figure 39. The screenshot displays a parallel coordinate plot implementing the focus-context technique, with selected companies being HASI and PETS.

7.2.2 Interactive Visualization of Dynamically Ranked Stacked Bar Charts

Integrating interactive capabilities into stacked bar charts with specific category legends is essential. This visualization method effectively displays investors' performance across various stocks, considering capital gains, losses, and dividend returns. While users can discern the overall investment performance through a comprehensive ranking, there's a pronounced need for interactive and dynamic ranking functionalities for in-depth data analysis about specific stocks or categories. In response to this need, the chart design enables dynamic ranking, sorting, and vividly displaying the exact monetary amounts on the Y-axis. To further ascertain the efficacy of this approach, the subsequent sections on testing and evaluation delve into two predefined case studies, with Figure 40 providing a visual demonstration.

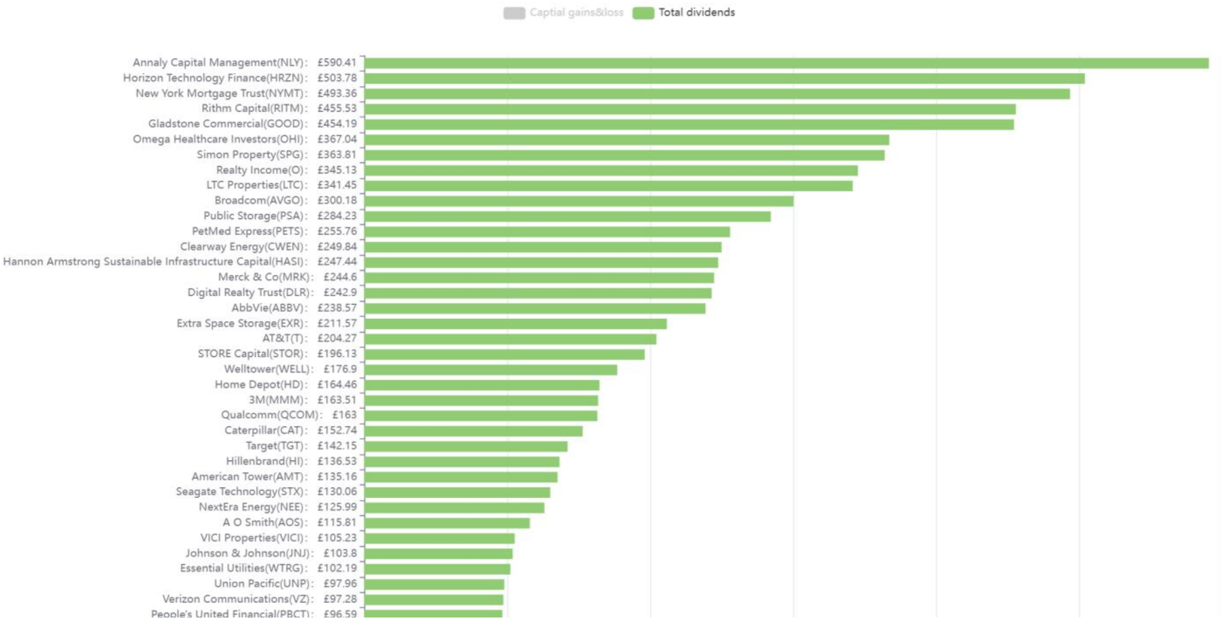


Figure 40(a). The screenshot illustrates dynamic rankings based on dividends with investment returns displayed on the Y-axis. It is evident from the graph that NLY has the highest dividend payout at ￡590.41.

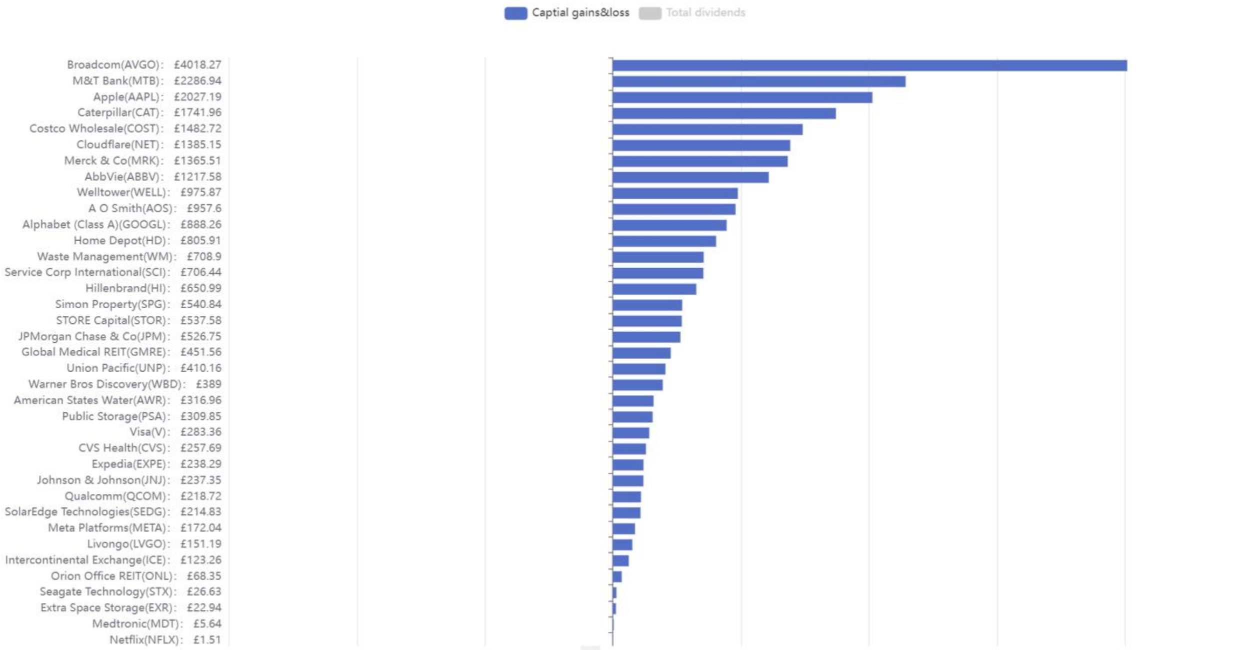


Figure 40(b). The screenshot illustrates dynamic rankings based on Capital gains with investment returns displayed on the Y-axis. It is apparent from the graph that AVGO exhibits the highest capital gains, amounting to £4018.27.

7.2.3 Sunburst Chart

In analyzing the comprehensive dataset of investors' activities, relying solely on date-based associations and pie charts detailing daily transaction amounts, stocks, and frequencies might not achieve optimal clarity and intuitiveness. The insufficiency can be attributed to the potential variation in daily transaction types. For instance, the investor could purchase and sell a company's shares on a specific day. Accordingly, the sunburst diagram was introduced to address these visualization challenges and enhance clarity. The sunburst chart presents the data hierarchically: initially categorized by actions and subdivided within these action categories to display the associated stock trades. This tiered structure enables users to progress from a broad overview based on actions to the intricate details of specific stock transactions, simplifying the data exploration while enhancing clarity and comprehensibility.

The system constructed the chart using Echarts.js. In the sunburst visualization, data is first segregated based on transaction types (such as purchases or sales). Subsequently, each segment displays the respective stock transactions and their associated values. Larger transaction values correspond to proportionally larger pieces in the diagram. Figure 41 depicts a sunburst chart representing the 11th of April 2022 trades. This chart reveals that investors engaged in actions such as 'market buy,' 'market sell,' and 'deposit' on that particular day. Further granularity is provided by showcasing the specific stocks traded under each action category, with the segmented angles of the chart indicating the proportion of each stock transaction.

Figure 41. The screenshot of Sunburst chart of trades on April 11, 2022. It details investor actions ('market buy', 'market sell', 'deposit') and associated stocks, with segment angles indicating trade proportions.

7.2.4 Date Selection in Calendar View

In calendar visualisations, the date selection feature is paramount and is a fundamental attribute of the calendar interface. This feature facilitates users in conducting a comprehensive analysis of investor data based on specific years, months, or days. In the preliminary design phase of the calendar interface, the selection functionality predominantly relied on the date widget provided by Element Plus, enabling a month-wise navigation within the system's default month. The system still employs the Element Plus date control for date selection currently. However, its functionality has undergone significant enhancements compared to previous versions. Figure 42 illustrates the date search functionality spanning from 1st April 2022 to 31st May 2022. When a specific date is selected, it is highlighted in blue, allowing users to identify their chosen dates. The current month and year are also prominently displayed at the top of the calendar search window. Users can conveniently access data for desired dates through this calendar search feature.

When filtering, the system first processes and retrieves the daily data corresponding to the selected date, ensuring seamless integration with the date widget and delivering accurate query results to users. The calendar view is synchronised with other visualisation components such as stacked bar charts, pie charts, sunburst diagrams, and tables. The data linkage between the calendar view and other visualisation components implies that whether users select a single day, an entire month, or a specific date range, these components will display data accordingly. Figure 42 indicates the interface for date selection.

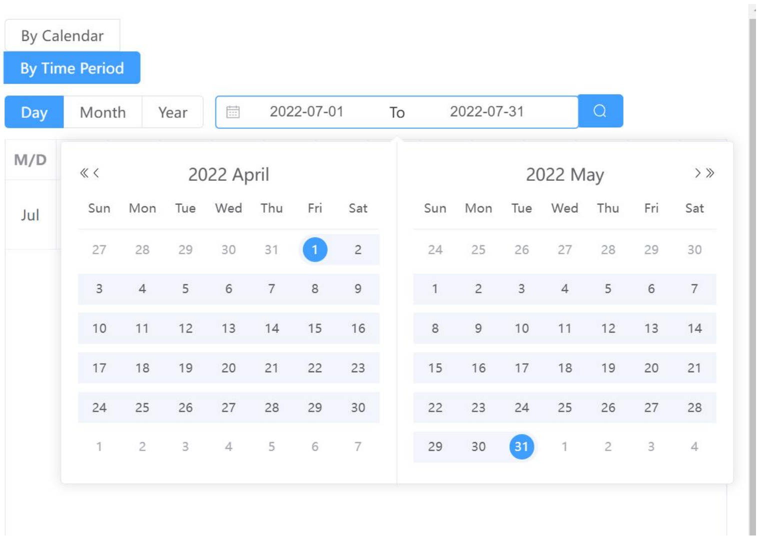


Figure 42. The screenshot of Date search functionality from 1st April 2022 to 31st May 2022. Selected dates are highlighted in blue for easy user identification.

7.2.5 Superpositioned Calendar View

The calendar view displays daily transaction amounts and related stock trading details in stock investment analysis. However, relying solely on the data from a specific date might not provide comprehensive insight for investors. The reason for this perspective is that holding and investing in stocks is typically a long-term strategy. Consequently, to cater to the requirements of long-term investment analysis, a superimposed calendar view has been integrated into the existing calendar chart, as depicted in Figure 43. When users specify a particular query period, the superimposed calendar view systematically aggregates all the data from that selected interval. Simultaneously, this visualisation displays pie charts based on daily transaction amounts and associated shares. Compared to the standard calendar view, the superimposed calendar view demonstrates superior performance in data summarisation and visualisation, providing investors with a more detailed and holistic interpretation of the data.

Figure 43(a) presents a combined monthly and daily view of the trading data for 2022. This visualisation format allows users to visualise trading activities on a day-to-day basis comprehensively. In contrast, Figure 43(b) displays a layered monthly view of the 2022 trading data, enabling users to visualise transactions monthly.

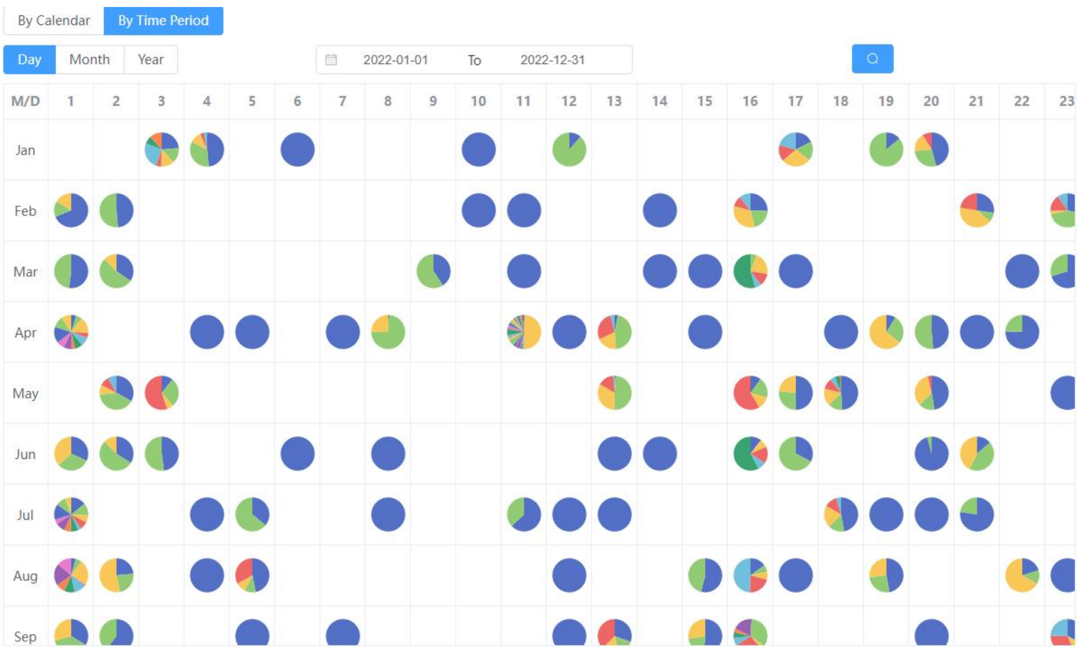


Figure 43(a). The screenshot of the superpositioned calendar view of 2022,visualized on a daily basis

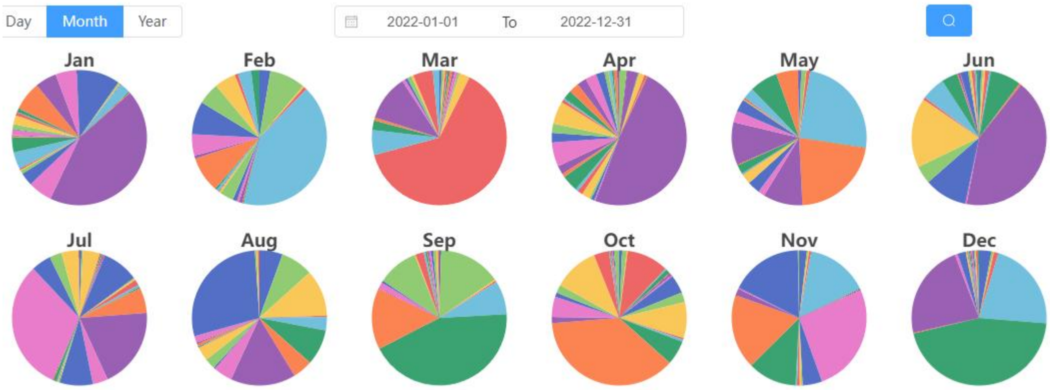


Figure 43(b). The screenshot of the superpositioned calendar view of 2022, visualised monthly.

7.2.6 Adaptive Screen Resolution Techniques

Given the diverse screen resolutions across computers, there can be inconsistencies in the presentation of the system's charts on different devices. In response to this inconsistency, the code underwent refinement and optimisation, explicitly targeting these disparities in chart displays. As a result of this optimisation, the system is designed to provide a uniform viewing experience, regardless of the computer's resolution. Additionally, integrating adaptive screen resolution functionality into the charts aims to enhance both the quality of visuals and the accuracy of data representation, further improving user-friendliness and adaptability.

7.3 Code Commenting

The source code for this project was annotated and documented utilising TSDoc.Despite the recommendation from Bob's Project Guidelines [Lar10] to employ a Java documentation or C document library, the development language used in the project is TypeScript. Given TSDoc's compatibility with TypeScript documentation, it was implemented throughout. This project employs VS Code for code compilation. TSDoc can be installed within VS Code using the "NPM i tsdoc" command. With TSDoc, it’s feasible to generate and create documentation for each TypeScript element of the website rapidly. Detailed guidance on the use of TSDoc is available on its official website.

An HTML website is included in the "docs" folder of the submission, forming an integral part of the portfolio accompanying this document. The website assists potential future researchers or developers understand the parameters, namespaces, and functionalities used throughout the project.

8 Evaluation

The evaluation section presents the results of the case studies. Throughout the implementation phase, rigorous testing targeted each visual chart and its corresponding interactive components, such as user selection interactions. The results of these tests are evident in the final visual representations.

8.1 Results

The core objective of this project is to showcase the performance of stocks associated with anonymous investors. Accordingly, the themes of the case studies have been specifically designed to emphasize and align with this central purpose.

8.1.1 Case Study A Comparison of Capital Gains/Loss with Dividend Gains for mREITs

1) Introduction

NLY, NYMT, and RITM are all mortgage Real Estate Investment Trusts (mREITs), according to Pellerin et al.[PW13], mREITs or Mortgage Real Estate Investment Trusts primarily invest in mortgage-backed securities. Their income is generated mainly from the interest on these mortgage loans, which then gets passed on to shareholders as dividends. Their model is sensitive to interest rate fluctuations: when rates rise, the price of existing mortgage-backed securities generally drops; conversely when rates fall, the prices rise.

2) Background on U.S. Mortgage Rates

Recent data on U.S. MBA 30-year fixed mortgage rates show a significant uptick:

- From 2021 to 2022: 2.86% increased to 3.27%;
- From 2022 to 2023: A sharp rise from 3.33% to 6.34%;
- During 2023: A further increase from 6.58% to 7.09%;

This rapid escalation in rates suggests that the cost of borrowing has substantially increased for homebuyers, making housing less affordable. As Nielsen points out[Nie23], Such an environment could slow down the housing market and potentially increase the default rate on new mortgages.

3) Analysis of Investment Performance

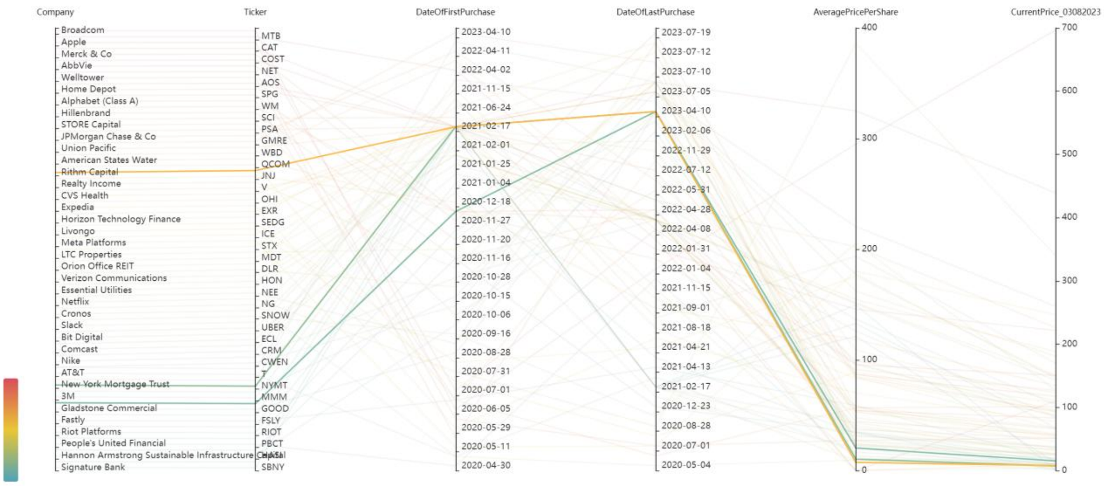


Figure 44. The screenshot of Parallel Coordinate View about NLY, NYMT, and RITM.

A comprehensive overview of the investment associated with Annaly Capital Management (NLY), New York Mortgage Trust (NYMT), and Rithm Capital (RITM) shares can be found in Figure 44, which visualises the trends using a parallel coordinate chart.

- Shares in Annaly Capital Management (NLY) were first purchased on 14 December 2020, each valued at £6.24. A subsequent transaction on 10 April 2023 registered a discernible uptick in value to £15.47 per unit. Yet, when averaging the expenditure over the entirety of the purchased period, the mean investment stands at £20.41 per share. This average is in stark contrast to the market rate of £15.38 as of 3 August 2023. This differential signifies a downturn in capital performance. To quantify this depreciation: with a portfolio of 330.5 NLY shares, there is an aggregate capital deficit of £1,276.42, marking a deleterious effect on the overall investment.
- Shares in New York Mortgage Trust (NYMT) were first acquired on 17 February 2021, each at a cost of £2.91. By 10 April 2023, there was a significant increase in their market price, reaching £8.26 per share. Yet, an average purchase price across all transactions stands at £10.36 per share, which, when compared to the market valuation of £7.77 as of 3 August 2023, indicates a decline in capital performance. Holding 487 NYMT shares, this translates to an aggregate capital loss of £992.12, underscoring a detrimental impact on the investment.
- Rithm Capital (RITM) shares were initially acquired on 17 February 2021 at £7.12 each. However, by 10 April 2023, the purchase price had decreased to £6.36 per share. Taking into account all transactions, the average cost per share is

calculated at £7.39. Compared to its market price of £7.28 on 3 August 2023, a modest capital loss is evident. Given a holding of 644 RITM shares, this results in a collective loss of £83.4.

4) Dividend Analysis

Dividend distributions for Annaly Capital Management (NLY), New York Mortgage Trust (NYMT), and Rithm Capital (RITM) during the respective holding periods are illustrated in Figure 45. Throughout this period, NLY disbursed dividends amounting to £590.41, marking it as the leading distributor among the three entities. For comparison, NYMT allocated dividends of £493.36, while RITM's distributions totaled £455.53. This information provides insight into the dividend performance of each stock, with NLY being the highest, followed by NYMT and then RITM.

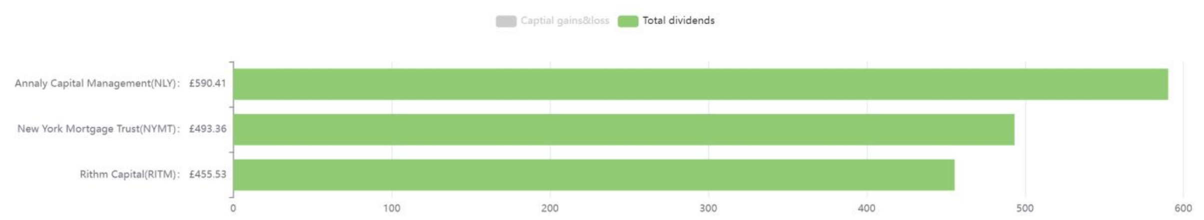


Figure 45. The screenshot to display the dividend about NLY, NYMT, and RITM.

5) Overall Performance Investment

Figure 46 provides a comprehensive graphical representation of the investment performance for NLY,NYMT and RITM. The details are provided below:

- Annaly Capital Management (NLY): Despite a capital loss of £1,276.42, NLY paid out dividends totaling £590.41. Thus, the net loss for this investment stands at £686.01.
- New York Mortgage Trust (NYMT): NYMT posted a capital loss of £992.12 but compensated with dividends of £493.36, resulting in a net loss of £498.76.
- Rithm Capital (RITM): With a capital loss of £83.4 and dividends amounting to £455.53, RITM's overall financial impact was a net gain of £372.13.

Overall, as observed from Figure 45, Only Rithm Capital (RITM) has dividend income sufficient to offset its capital loss. In contrast, Annaly Capital Management (NLY) and New York Mortgage Trust (NYMT) cannot fully compensate for their respective capital losses through dividends.

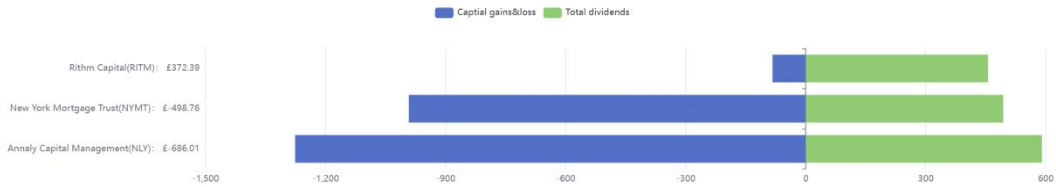


Figure 46. The screenshot displays the Investment Performance of NLY, NYMT, and RITM. While Rithm Capital (RITM) offsets its capital loss with dividends, both Annaly Capital Management (NLY) and New York Mortgage Trust (NYMT) have dividends that are insufficient to cover their capital losses.

8.1.2 Case Study B Analysis of Capital Gains and Dividend Across a Portfolio of 76 Stocks

1) Introduction

Diversified investment primarily involves the strategic allocation of resources across various stocks to maximise returns while minimising risk. In this case study, we explore an individual investor's spread across more than 70 companies. By examining both capital gains or losses and dividend returns for each company, our goal is to identify the top and bottom performers. Specifically, we highlight the top 10 and bottom 10 companies, providing a comprehensive overview of the investor's portfolio performance. This analysis serves not just as a review but also aims to offer valuable insights for future investment decisions.

2) Methodology

- Data Collection:
  1) Company Identification: Using the investor's trading records, all 76 invested companies were pinpointed.
  2) Trade Details Extraction: For each company, the following specific elements from the investor's stock transaction history were gathered:
     - Initial Purchase Details:
       a) Time of the first stock purchase.
       b) Price at which the stock was first acquired.
     - Most Recent Purchase Details:
       a) Time of the latest stock purchase.
       b) Price during the most recent acquisition.
     - Quantity Details:
       a) Total number of shares purchased for each company.
     - Realized Capital Gains/Losses:

    a) Results from stock sales, as explicitly documented in the transaction records.
  - Dividend Record:
    a) Total dividends received for each stock during the holding period.

- Analysis:
  1) Average Purchase Price Determination: For each stock, compute the average acquisition cost. This entails consolidating all individual transactions and their associated prices.
  2) Unrealized Capital Gain or Loss:
     - Utilise the stock price as of 3 August 2023.
     - Subtract the average purchase price from the current price.
     - Multiply the difference by the total number of shares currently held..
  3) Overall Capital Gain or Loss: To determine the total capital gain or loss for each stock, combine both the realized and unrealized capital gains or losses.
  4) Total Return Analysis: Integrate the capital gains/losses with dividends for a complete performance review of each stock.

3) Analysis of Investment Performance

As of 21st July 2023, the stock market showcased a diverse range of performances. As depicted in Figure 47, Broadcom (AVGO) emerged as the top performer, boasting a capital gain of £4,018.27. In contrast, Signature Bank (SBNY) experienced the most significant decline, recording a capital loss of £2,839.71.

In the forefront of gains, Broadcom (AVGO) took the lead with a capital appreciation of £4,018.27. Following closely were M&T Bank (MTB) at £2,286.94, Apple (AAPL) with £2,027.19, Caterpillar (CAT) at £1,741.96, and Costco Wholesale (COST) which saw gains of £1,482.72. Completing the top ten were Cloudflare (NET) at £1,385.15, Merck & Co (MRK) with £1,365.51, AbbVie (ABBV) registering £1,217.58, Welltower (WELL) at £975.87, and A O Smith (AOS) which posted £957.60 in capital gains.

Conversely, the stocks witnessing the most significant dips were led by Signature Bank (SBNY) which recorded a considerable capital decrease of £2,839.71. Accompanying it in this downturn were Hannon Armstrong Sustainable Infrastructure Capital (HASI) with -£2,232.67, Beyond Meat (BYND) at -£2,176.98, PetMed Express (PETS) which declined by -£2,046.82, and Silvergate Capital (SICP) with -£1,960.77.

The list of these bottom performers also includes People's United Financial (PBCT) with a loss of £1,891.72, Riot Platforms (RIOT) down by -£1,744.92, Annaly Capital Management (NLY) at -£1,276.42, Walt Disney (DIS) registering -£1,201.33, and lastly, Gladstone Commercial (GOOD) which fell by -£1,176.55.

Some stocks, like M&T Bank (MTB) and Warner Bros Discovery (WBD), were purchased at £0.00. This could be due to stock splits, bonuses, or other corporate actions where additional shares might be awarded without an associated cost.

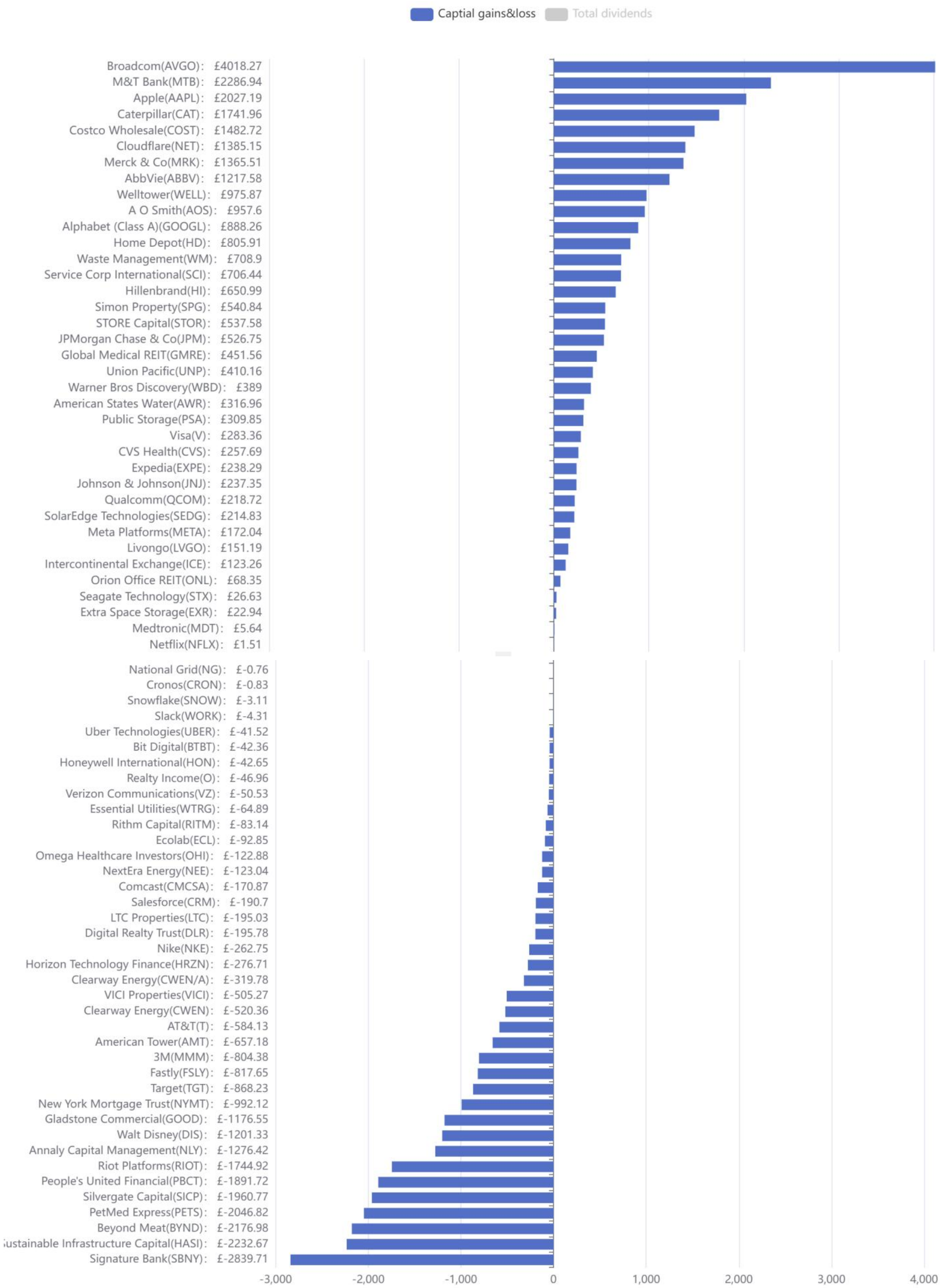


Figure 47. The screen shot of capital gains about 76 stocks.Among 76 stocks, Broadcom (AVGO) showed the highest capital gain at £4,018.27, while Signature Bank (SBNY) had the largest decline with a capital loss of £2,839.71.

4) Dividend Analysis

From the data on 21st July 2023, it's clear that dividends across stocks vary significantly,which is displayed in Figure 48. As observed in Figure 48, Orion Office REIT (ONL) takes the top spot with a dividend of £590.41. In contrast, Riot Platforms (RIOT) appears at the other end, distributing a mere £0.42.

Looking at the top dividend payouts on 21st July 2023, Orion Office REIT (ONL) leads with £590.41. Following this, we see Walt Disney (DIS) at £503.78, Horizon Technology Finance (HRZN) with £493.36, Clearway Energy (CWEN) at £455.53, and Simon Property (SPG) with £454.19. Completing the top ten, there's Clearway Energy (CWEN/A) at £367.04, SolarEdge Technologies (SEDG) with £363.81, CVS Health (CVS) at £345.13, Qualcomm (QCOM) with £341.45, and AbbVie (ABBV) at £300.18.

Several companies chose not to give dividends. Notable names on this list include Caterpillar (CAT), Cloudflare (NET), Merck & Co (MRK), Welltower (WELL), and Alphabet (Class A) (GOOGL). Others like Service Corp International (SCI), Hillenbrand (HI), JPMorgan Chase & Co (JPM), Johnson & Johnson (JNJ), and Netflix (NFLX) also kept their dividend tally at zero. The list continues with National Grid (NG), Ecolab (ECL), Comcast (CMCSA), American Tower (AMT), Fastly (FSLY), New York Mortgage Trust (NYMT), Annaly Capital Management (NLY), and PetMed Express (PETS), all of whom decided against dividend payouts.

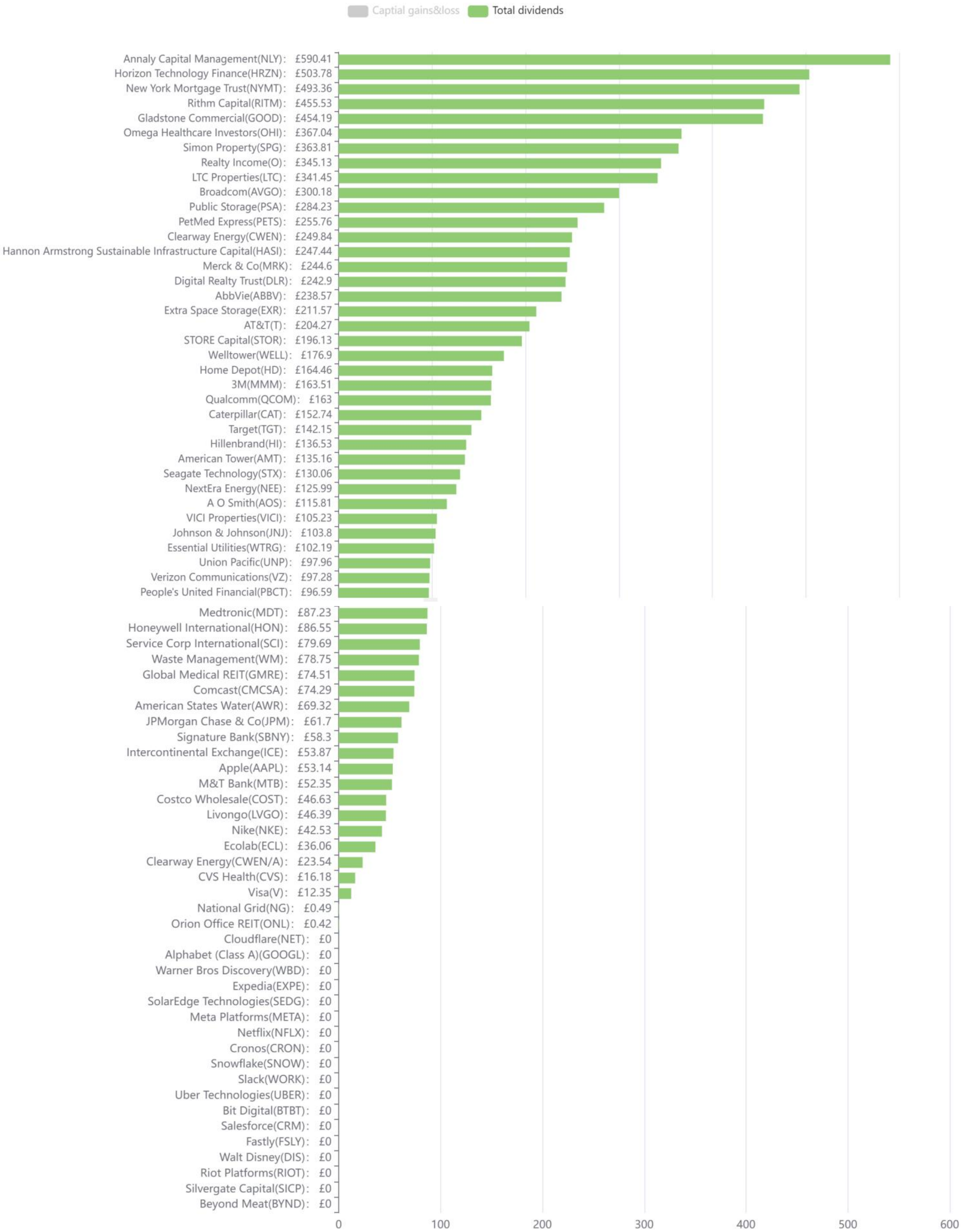


Figure 48. The screenshot of dividends for 76 stocks reveals Annaly Capital Management (NLY) at the forefront with a dividend of £590.41.

5) Overall Performance Analysis

Upon combining capital gains with dividends to evaluate the overall performance as of 21st July 2023, a more holistic perspective on stock performances emerged(seen in Figure 49). Leading the charge is Broadcom (AVGO) with an impressive £4,114.86, whereas Signature Bank (SBNY) finds itself at the opposite end of the spectrum, recording a performance of -£2,765.20.

When we narrow down the top performers, Broadcom (AVGO) takes the crown with £4,114.86. Following closely are M&T Bank (MTB) at £2,287.43, Apple (AAPL) with £2,231.46, and Caterpillar (CAT) holding strong at £1,741.96. Other companies making up the top ten include Costco Wholesale (COST) with £1,659.62, AbbVie (ABBV) at £1,517.76, Cloudflare (NET) with £1,385.15, Merck & Co (MRK) at £1,365.51, A O Smith (AOS) posting £1,120.60, and Simon Property (SPG) rounding it off with £995.03.

On the other hand, some stocks didn't perform as well. Signature Bank (SBNY) posted the lowest performance of -£2,765.20. Other stocks in this lower bracket include Hannon Armstrong Sustainable Infrastructure Capital (HASI) at -£2,127.44, and Beyond Meat (BYND) recording -£2,115.28, Others in this list include PetMed Express (PETS) with -£2,046.82, Silvergate Capital (SICP) posting -£1,908.42, People's United Financial (PBCT) at -£1,868.18, Riot Platforms (RIOT) with -£1,744.50, Annaly Capital Management (NLY) at -£1,276.42, and Gladstone Commercial (GOOD) with -£1,096.86.

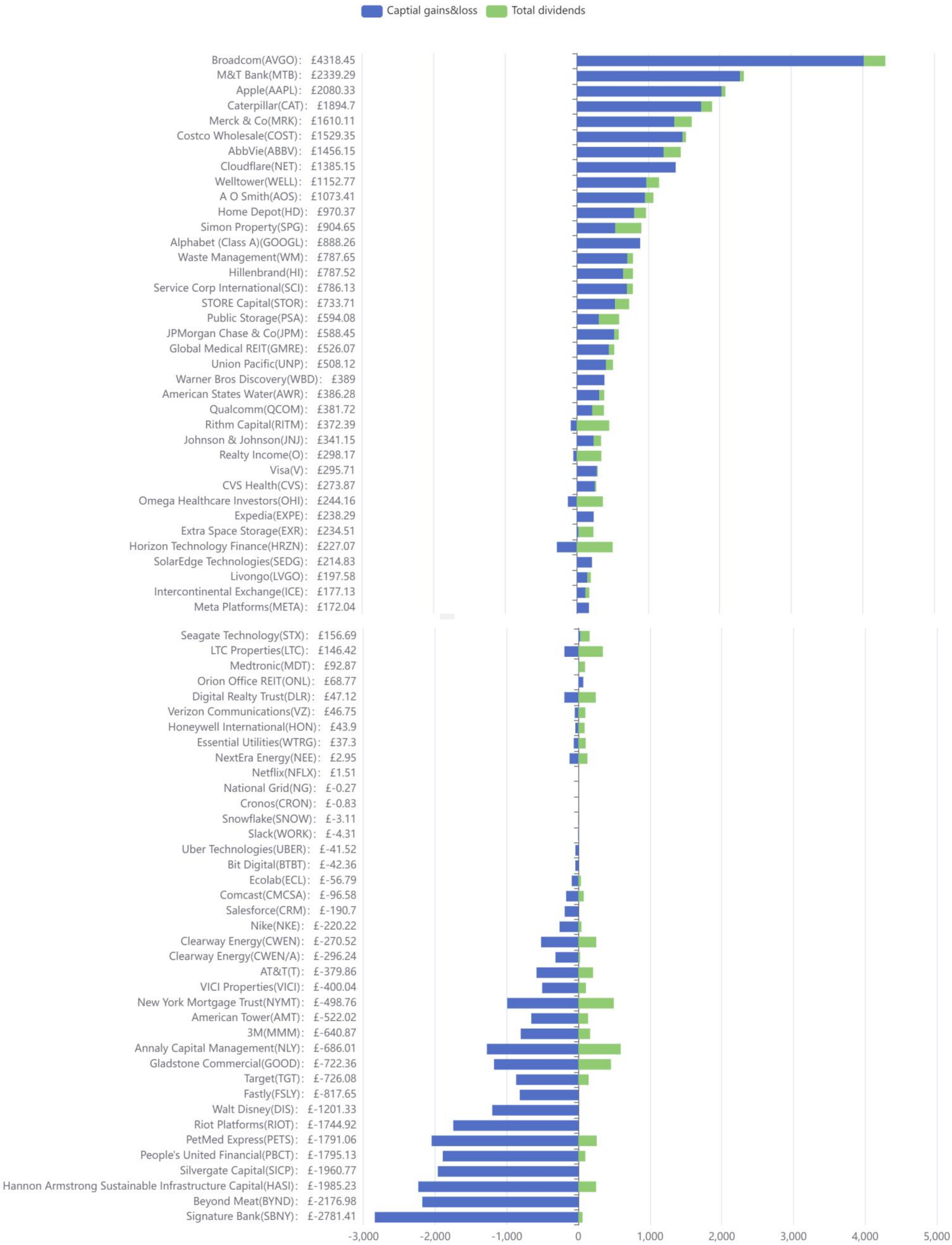


Figure 49. Investment performance screenshot of 76 stocks: Broadcom (AVGO) leads with £4,114.86, while Signature Bank (SBNY) lags at -£2,765.20.

8.1.3 Demonstration Video

This section includes a demonstration video that intuitively showcases the features and functionalities of this research project. The video covers the project's interface, key features, and user interactions to provide a complete understanding of its capabilities. The link to the video is provided below:
https://www.youtube.com/watch?v=NYceDKyfQus

9 Conclusion

The thesis and development work aimed to establish a visualization analysis system tailored explicitly for analyzing investment data from anonymous investors. The system provides the following significant features:

Robust Querying and Statistical Analysis: The tool facilitates user querying, data statistics, and the generation of exploratory visualisations.

Diverse Data Visualisation: The application displays various visualisation charts, such as parallel coordinate plots and stacked bar charts.

Specialised Stock Investment Analysis: Through case studies, the system analyses the investment performance of various stocks, serving as an effective analytical tool for potential investors.

Distinctive Interactive Features:

- The system creates parallel coordinate plots to elucidate the characteristics of the 76 stocks in which anonymous investors have invested.
- The platform visualises the investment outcomes for these 76 stocks, distinctly categorised by capital gains and dividends.
- In the calendar visualisation, the system incorporates a date-filtering feature. Upon selecting a specific date range, the system generates a superpositioned calendar representation corresponding to the chosen interval.
- The system allows users to magnify data specific to certain dates and offers a choice of visualisation formats, including stacked bar charts, pie charts, sunburst diagrams, and tables.

Furthermore, the system is enriched with extensive interactive capabilities and tooltips to enhance the user's experience.

Compared to established platforms like Google Finance or Seeking Alpha, This application exhibits certain limitations, especially concerning the comprehensiveness of its stock information. However, its visualisations possess distinctive features unavailable in these established platforms.

Overall, this research presents a pioneering approach to visualizing investment data and introduces the first publicly accessible dataset of transactions from anonymous investors stored on Google Sheets.

10 Future Work

The optimisation of the system and the enrichment of the dataset remain primary concerns. The strategy aims to enhance its capacity to deliver essential information to users precisely and efficiently through consistent refinements. The current dataset may encompass the stock trading activities of only a select number of firms. However, as the research advances, the goal is to capture an expanded range of investment data, which industry sectors will then categorise.

Secondly, delving deeply into the dataset's potential information is paramount. For instance, the data might encompass detailed insights about transaction fees, such as those levied based on specific monetary thresholds or within particular time intervals. Additionally, concealed within the data could be pivotal information regarding market trends, seasonal influences, or the impacts of distinct event s on stock prices. Only through rigorous analysis can these insights be extracted, subsequently offering investors a more lucid and profound understanding of stock data.

Considering future data challenges, the ever-increasing volume of data is a critical issue. With advancements in data collection methods and the continuous evolution of big data technology, there is a pressing imperative to process and analyse extensive data volumes swiftly and efficiently. Furthermore, the interactivity between visual representations across various datasets is an essential research focus. For example, by integrating parallel coordinate plots with calendar charts, selecting a specific company within the parallel coordinate plot should prompt the calendar chart to automatically illustrate the respective company's trading activities across different dates. Such interactivity enhances the user experience, provides a more intuitive data interpretation, and substantially boosts system practicality.

Reference

[AA13] Adenowo, A. A. A. and Adenowo, B. A. "Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach." In: International Journal of Scientific & Engineering Research 4.7. 2013, pp. 427-434.

[Alp20] Seeking Alpha. Stock Picks, Stock Market Investing. 2020. URL: https://seekingalpha.com/symbol/AAPL.

[AT+23] Arleo, A., Tsigkanos, C., Leite, R. A., Dustdar, S., Miksch, S., & Sorger, J. Visual Exploration of Financial Data with Incremental Domain Knowledge. Computer Graphics Forum, 2023,42, pp. 101-116.

[CQ16] Chan, Y. -Y., & Qu, H. (2016). FinaVistory: Using Narrative Visualization to explain social and Economic relationships in financial news. 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 2016, pp. 32-39.

[Dao+08] Huyen Tue Dao et al. "NASDAQ Velocity and Forces: An Interactive Visualization of Activity and Change." In: J. UCS 14.9 ,2008, pp.1391–1410.

[EFN12] Endert A., Fiaux P., North C.: Semantic interaction for visual text analytics. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2012, ACM, pp. 473–482.

[ERT+17] A. Endert, W. Ribarsky, C. Turkay et al. "The state of the art in integrating machine learning into visual analytics". In: Computer Graphics Forum. Vol. 36. 8. Wiley Online Library. 2017,pp. 458–486.

[FVS+23] Firat, E. E., Vytia, D., Vasudeva, N., Jiang, Z., & Laramee, R. S. MoneyVis: Open Bank Transaction Data for Visualization and Beyond. Eurovis Short Papers, Eurovis 2023, Leipzig, Germany, 2023.pp. 12-16.

[Goo20] Google. Finance - Google Search. 2020. URL: https://www.google.co.uk/finance.

[HW+18] Hua, J., Huang, M. L., Wang, G., & Zreika, M. Applying data visualization techniques for stock relationship analysis. Filomat, 2018, 32, pp. 1931-1936.

[ID17] Vaishali Ingle and Sachin Deshmukh. "Live news streams extraction for visualization of stock market trends". In: Proceedings of the International Conference on Signal, Networks, Computing, and Systems. Springer. 2017, pp. 297–301.

[KC+16] Ko, S., Cho, I., Afzal, S., Yau, C., Chae, J., Malik, A., Beck, K., Jang, Y., Ribarsky, W., & Ebert, D. S. A Survey on Visual Analysis Approaches for Financial Data. Computer Graphics Forum,2016, 35, pp.599-617.

[Ko+16] Sungahn Ko et al. “A survey on visual analysis approaches for financial data”. In: Computer Graphics Forum. Vol. 35. 3. Wiley Online Library. 2016, pp. 599–617.

[Lar10] Robert S Laramee. “Bob’s concise coding conventions (c3)”. In: Advances in Computer Science and Engineering (ACSE) (2010).

[LZ10] Su Te Lei and Kang Zhang. “A visual analytics system for financial time-series data”. In: Proceedings of the 3rd International Symposium on Visual Information Communication. 2010, pp. 1–9.

[Lar11] Laramee, Bob, and S. Robert. "Bob's Project Guidelines: Writing a Dissertation for a BSc. in Computer Science." Innovation in Teaching and Learning in Information and Computer Sciences 10.1, 2011, pp. 43-54.

[LG+18] Leite R A, Gschwandtner T, Miksch S, et al. Eva: Visual analytics to identify fraudulent events[J]. IEEE transactions on visualization and computer graphics, 2017, 24(1): pp. 330-339.

[ML17] Liam McNabb and Robert S Laramee. “Survey of Surveys (SoS)-Mapping The Landscape of Survey Papers in Information Visualization”. In: Computer Graphics Forum. Vol. 36. 3. Wiley Online Library. 2017, pp. 589–617.

[Ozg+17] Ceyhun Ozgur et al. “MatLab vs. Python vs. R”. In: Journal of Data Science 15.3 , 2017, pp. 355–372.

[PM17] Shyam Prabhakar and Larry Maves. “Big Data Analytics and Visualization: Finance”. In: Big Data and Visual Analytics. Springer, 2017, pp. 219–229.

[RL18] Roberts, R. C., & Laramee, R. S. (2018). Visualising Business Data: A Survey. Information, 2018, 9, 285.

[RL19] Rees, D., & Laramee, R. S. A Survey of Information Visualization Books. Computer Graphics Forum, 2019, 38, pp.610-646.

[Shn03] Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations. In: Bederson B, Shneiderman B (eds) The craft of information visualization. Elsevier, 2003, pp.364 – 371.

[Sch+09] Tobias Schreck et al. “Visual cluster analysis of trajectory data with interactive kohonen maps”. In: Information Visualization 8.1, 2009, pp. 14–29.

[SB13] Sorenson, E., & Brath, R. Financial Visualization Case Study: Correlating Financial

Timeseries and Discrete Events to Support Investment Decisions. 2013 17th International Conference on Information Visualisation, London, UK,2013. pp.232-238.

[Sun+19] Sun Y, Xiong H, Yiu S M, et al. Bitvis: An interactive visualization system for bitcoin accounts analysis[C]//2019 Crypto Valley conference on blockchain technology (CVCBT). IEEE, 2019: 21-25.

[Yue+19] Xuanwu Yue et al. “sPortfolio: Stratified Visual Analysis of Stock Portfolios”. In: IEEE Transactions on Visualization and Computer Graphics 26.1, 2019, pp. 601–610.

[Yah20] Yahoo. Yahoo Finance – stock market live, quotes, business finance news. 2020. URL: https://uk.finance.yahoo.com/.

[YG22] Yaali J, Grégoire V, Hurtut T. HFTViz: Visualization for the exploration of high frequency trading data[J]. Information Visualization, 2022, 21(2): pp. 182-193.