

**Invest Smart: A Consumer-Centric Investment Visualization Tool**

Submitted September 2024, in partial fulfilment of

the conditions for the award of the degree **MSc Data Science**

Supervised by Professor Robert S. Laramee

**20607987**

School of Computer Science University of Nottingham

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

**Signature: *Aishwarya Shahu***

**Date:**  **06 / 09 / 2024**

# Abstract

This dissertation presents the development of "Invest Smart," a consumer-centric investment visualization tool designed to empower retail investors by providing an intuitive platform for monitoring and analysing investment portfolios. The tool integrates both real-time and historical financial data to deliver detailed insights into investment performance, supporting data-driven decision-making. Utilizing Python and its robust visualization libraries, the tool offers interactive dashboards that simplify complex financial information, making it accessible even to non-expert users. The tool's ability to customize visualizations according to individual investor preferences further enhances its usability. Evaluation through various case studies demonstrates that "Invest Smart" effectively meets its design objectives, providing investors with actionable insights that align with their unique financial goals. This project contributes to the field of financial technology by introducing a scalable, user-friendly platform that bridges the gap between retail investors and sophisticated financial analysis.

**Keywords:** Investment visualization, financial technology, Python, real-time data, interactive dashboards, retail investors, data-driven decision-making.

# Acknowledgments

I would like to extend my deepest gratitude to Professor Robert S. Laramee for his invaluable guidance, support, and expertise throughout the development of this project. His mentorship and insightful feedback have been instrumental in shaping this dissertation and enhancing my understanding of data visualization and financial analysis. I am particularly grateful for the structured approach and regular meetings that kept the project on track.

I would also like to thank the entire visualization project group and the FinVis group for their collaborative spirit and constructive discussions during our weekly meetings. Their diverse perspectives and suggestions significantly enriched this project and contributed to its success.

Lastly, I am deeply appreciative of my family and friends for their unwavering encouragement and support, which kept me motivated throughout this challenging journey. Their belief in my abilities was a constant source of inspiration.

**Table of Contents**

[Abstract 2](#_bookmark0)

[Acknowledgments 3](#_bookmark1)

1. [Introduction 6](#_bookmark2)
   1. [Motivation 7](#_bookmark3)
   2. [Aims and Objectives 7](#_bookmark4)
   3. [Challenges 8](#_bookmark5)
      1. [Field Challenges 8](#_bookmark6)
      2. [Project Challenges 9](#_bookmark7)
   4. [Thesis Structure 9](#_bookmark8)
2. [Background 10](#_bookmark9)
   1. [Related Work 10](#_bookmark10)
      1. [Literature Scope 10](#_bookmark11)
      2. [Visualisation Surveys 11](#_bookmark12)
      3. [Finance Visualisation 16](#_bookmark13)
   2. [Previous systems 24](#_bookmark14)
      1. [Yahoo Finance 24](#_bookmark15)
      2. [Google Finance 25](#_bookmark16)
      3. [Morningstar 26](#_bookmark17)
   3. [Data Characteristics 27](#_bookmark18)
      1. [Data Sources 27](#_bookmark19)
      2. [Data Descriptions 28](#_bookmark20)
3. [Project Specification 32](#_bookmark21)
   1. [Feature Specification 32](#_bookmark22)
      1. [Must-have Features 32](#_bookmark23)
      2. [Optional Features 33](#_bookmark24)
   2. [Technology Choices 35](#_bookmark25)
      1. [Programming Languages 35](#_bookmark26)
      2. [Libraries 37](#_bookmark27)
      3. [Other Technologies 39](#_bookmark28)
4. [Project Plan and Timetable 42](#_bookmark29)
5. [Project Design 43](#_bookmark30)
   1. [Visualization Pipeline 43](#_bookmark31)
   2. [Process Diagram 44](#_bookmark32)
6. [Implementation 45](#_bookmark33)
   1. [Existing Tools 46](#_bookmark34)
      1. [Google Chart 47](#_bookmark35)
      2. [Excel 48](#_bookmark36)
      3. [Tableau 49](#_bookmark37)
      4. [Power BI 50](#_bookmark38)
   2. [Basic Implementation 52](#_bookmark39)
      1. [Dashboard UI 52](#_bookmark40)
      2. [Buy/Sell View 54](#_bookmark41)
      3. [Gain/Loss View 55](#_bookmark42)
      4. [Dividend View 57](#_bookmark43)
      5. [Single Stock View 59](#_bookmark44)
      6. [Parallel Coordinate Chart 61](#_bookmark45)
      7. [Multiple Line Chart 63](#_bookmark46)
      8. [Home View 64](#_bookmark47)
      9. [Colour Maps 65](#_bookmark48)
      10. [User Options 66](#_bookmark49)
   3. [Enhancements 67](#_bookmark50)
      1. [Toggle Views in Dividend View 67](#_bookmark51)
      2. [Chart Types in Single Stock View 68](#_bookmark52)
      3. [Grouped Legends in Multiple Line Chart 70](#_bookmark53)
      4. [Linked Brushings in Home View 70](#_bookmark54)
      5. [Interval Buttons in Time-Based Charts 72](#_bookmark55)
      6. [Legend Interactions in All Charts 73](#_bookmark56)
      7. [Zoom and Box Select in All Charts 74](#_bookmark57)
      8. [Informative Tooltips in All Charts 76](#_bookmark58)
      9. [Adaptive Screen Resolution 76](#_bookmark59)
7. [Evaluation 77](#_bookmark60)
   1. [Results 77](#_bookmark61)
      1. [Case Study A: Identification of Abnormal Data 77](#_bookmark62)
      2. [Case Study B: Comparison of Capital Gain/Loss vs Dividend 79](#_bookmark63)
      3. [Case Study C: Comparison of Good vs Bad Decisions 83](#_bookmark64)
8. [Conclusion 86](#_bookmark65)
9. [Future Work 87](#_bookmark66)

[Reference 88](#_bookmark67)

# Introduction

Investing in the stock market has become increasingly accessible to retail investors due to advancements in digital platforms and the availability of vast amounts of financial data. Despite this increased accessibility, many retail investors struggle to navigate the complexities of financial markets and make informed decisions due to the overwhelming nature of data and the lack of personalized, user-friendly tools. This dissertation introduces a project designed to empower retail investors by providing an intuitive platform for monitoring and analysing their investment portfolios. By integrating real-time and historical data, the tool offers detailed insights into investment performance, enabling users to make data-driven decisions and optimize their investment strategies.

### Motivation

The motivation behind this project stems from the challenges faced by retail investors in making sense of the complex and often fragmented financial information available to them. Traditional investment platforms tend to focus on providing broad market data, which may not be directly applicable to individual investor needs. As a result, many investors are left with a limited understanding of their portfolios and may make suboptimal investment choices.

This tool aims to bridge this gap by offering a personalized approach to investment management through advanced data visualization techniques. The tool not only simplifies complex financial data but also provides actionable insights that are directly relevant to the investor's specific financial goals. By making sophisticated analytics accessible to retail investors, the project seeks to enhance the investment experience and support better decision-making processes.

### Aims and Objectives

The primary aim of this project is to develop a comprehensive investment visualization tool that empowers retail investors to make more informed decisions. The specific objectives of the project are:

* **Integrating Real-Time and Historical Data:** The tool will aggregate data from various sources, including APIs such as Yahoo Finance, to provide up-to-date information on stock prices, market trends, and other key financial indicators.
* **Developing Interactive Visualizations:** Utilizing Python libraries like Plotly and Dash, the tool will offer interactive charts and graphs that enable users to explore their investment data in detail, including visualizations of realized and unrealized gains, dividends, and overall portfolio performance.
* **Personalizing User Experience:** The tool will feature customizable dashboards that cater to the unique preferences and needs of each investor. Users will be able to track specific stocks, sectors, and industries, allowing them to focus on the area most relevant to their investment strategies.
* **Evaluating Investment Decisions:** The tool will provide a framework for assessing buy and sell decisions based on historical performance data and moving averages, helping investors refine their trading strategies for better outcomes.

### Challenges

Developing a consumer-centric investment visualization tool presents a few challenges, both within the broader field of financial technology and within the specific scope of this project.

#### Field Challenges

* **Integration of Diverse Data Sources:** Financial data is often scattered across multiple platforms and presented in various formats, making integration a significant challenge. The tool requires robust data cleaning and normalization processes to ensure that the data used is accurate, reliable, and timely. Maintaining the integrity of real-time data feeds is crucial, as any delays or inaccuracies can severely impact the decision-making process.
* **Access to Individual Investment Data:** Obtaining detailed, individual-level investment data can be difficult due to privacy concerns and limitations on data sharing by financial institutions. This challenge necessitates finding ways to securely and efficiently integrate personal investment data into the tool without compromising user privacy or breaching regulatory requirements.
* **Complexity of Financial Metrics:** Many retail investors are not familiar with advanced financial metrics and analytics that are typically used by professional investors. This complexity can hinder their ability to make informed decisions. The challenge lies in designing visualizations that simplify these metrics while retaining their accuracy and usefulness, making them accessible and actionable for all users.
* **Security:** Security is a critical concern in financial technology, especially when handling sensitive investment data. Protecting user data from unauthorized access, breaches, and cyber threats is essential. The tool must incorporate strong security measures, such as encryption and secure authentication protocols, to safeguard personal information and build user trust.
* **Balancing Technical Accuracy with Usability:** One of the key challenges is to strike a balance between providing technically accurate information and ensuring that the tool is user-friendly. Investors need insights that are clear and actionable rather than overly technical details. Achieving this balance involves careful design and continuous user feedback to ensure that the tool meets the needs of its intended audience.

#### Project Challenges

* **Developing a Scalable and Responsive User Interface:** One of the main project challenges is creating a user interface that can handle dynamic data updates while maintaining performance and responsiveness. The tool must efficiently process and display large volumes of data, which requires careful selection of technologies and optimization of data handling processes to ensure a smooth user experience.
* **Customization for Individual Users:** Providing a high level of customization to cater to the varied needs of individual investors presents a significant design challenge. The tool must allow users to tailor their experience by choosing what data to view and how it is displayed, requiring a flexible and modular development approach.
* **Ensuring Data Security and Privacy:** As the tool handles sensitive financial information, implementing robust security measures to protect user data is critical. This includes encryption of data, secure user authentication, and adherence to data protection regulations to prevent unauthorized access and ensure the privacy of user information.
* **Technical Integration of Real-Time Data:** Integrating real-time data feeds into the tool in a way that is both seamless and reliable is another key challenge. The tool must handle potential issues such as data latency, discrepancies between different data sources, and ensuring that the real-time data is accurately reflected in the visualizations provided to the user.

### Thesis Structure

The thesis is structured according to the suggestions from Bob’s Project Guideline (Laramee, 2021). Following the Introduction, Section 2 provides a comprehensive background for this project, including a review of related work and an analysis of existing systems such as Yahoo Finance, Google Finance, and Morningstar. It also covers the characteristics of the data used in the project. Section 3 outlines the project specifications, detailing the must-have and optional features of the "Invest Smart" tool and explaining the technology choices, including the programming languages, libraries, and other technologies that were selected. Section 4 focuses on the project plan and timetable, presenting the development approach, which is guided by Agile methodology, and includes the Gantt chart that maps out the timeline for the project. Section 5 covers the project design, illustrating the data flow and system architecture through design diagrams, including the visualization pipeline and process diagrams. Section 6 delves into the implementation of the project, describing the development of each component within the tool. This includes the basic implementation, the dashboard's UI elements, the various visualizations, and the coding practices followed, ensuring the tool is robust and maintainable. Enhancements and additional features are also discussed in this section. Section 7 presents the evaluation of the tool through several case studies, which test the effectiveness of the tool in delivering personalized financial insights. The results demonstrate the tool’s capability in enhancing investment decision-making through dynamic visualizations and interactive features. Section 8 provides the conclusions of the thesis, summarizing the key findings and the impact of the "Invest Smart" tool. Finally, Section 9 explores potential future work, discussing possible enhancements and extensions that could further improve the tool and broaden its applicability.

# Background

The rise of data-driven decision-making in various fields has significantly influenced the development of tools designed to simplify and enhance the understanding of complex datasets. In financial technology, the role of data visualization has become increasingly crucial as it allows investors to gain insights from vast amounts of financial data. This section provides a comprehensive review of related work on data visualization, examines existing financial systems that provide visualization tools, and outlines the potential contributions of the proposed investment visualization tool.

### Related Work

The literature on data visualization is vast, encompassing a wide range of techniques and tools that have been developed to address different user needs. Visualization techniques have evolved from simple static charts to complex, interactive visual representations that can handle large and dynamic datasets. This section reviews key papers and resources, focusing on their contributions to the field of data visualization and their relevance to financial technology.

#### Visualization Resources

Liu et al. (2022) conducted a comprehensive survey titled "Visualization Resources: A Survey," which explores a wide range of visualization tools and techniques available to researchers and developers. The survey categorizes these resources into software libraries, frameworks, and literature, providing a detailed overview of the current state of visualization technologies. Liu et al. highlight the importance of choosing the right tools based on the specific needs of the visualization project, whether it involves static visualizations, interactive dashboards, or real-time data representation.

The survey by Liu et al. (2022) is particularly relevant to the development of the proposed investment visualization tool as it offers insights into the capabilities and limitations of various visualization libraries. For instance, the survey discusses libraries such as D3.js, Plotly, and Matplotlib, which are widely used for creating both basic and advanced visualizations. D3.js, known for its flexibility and control, is powerful for creating customized visualizations but requires significant coding expertise. On the other hand, Plotly, a library built on D3.js, offers higher-level abstractions that simplify the creation of interactive and dynamic visualizations, making it a strong candidate for use in financial dashboards where user interaction is critical.

The "Visualization Resources Presentation" at the Information Visualization Conference 2021, presented by X Liu, further emphasizes the practical applications of these resources. This presentation, alongside the associated web resources (<https://sites.google.com/view/visres/>), offers practical examples of how these tools can be applied in real-world scenarios, demonstrating their effectiveness in different contexts, including finance. For example, the presentation discusses how Plotly and Dash can be integrated to create comprehensive dashboards that not only display static data but also allow for real-time updates and user interactions, crucial features for an investment tool that needs to reflect market changes instantly.

#### Surveys on Visualization and Information Visualization Books

Rees and Laramee (2019), in their work "A Survey of Information Visualization Books," provide an extensive review of the literature on information visualization, categorizing and analyzing key texts that have shaped the field. Their survey is structured to guide readers through the theoretical and practical aspects of visualization, offering insights into the development of visualization tools that effectively communicate complex data insights.

Rees and Laramee (2019) highlight several key visualization techniques that have been developed over the years, such as scatterplots, heatmaps, and network visualizations. These techniques are particularly useful in financial technology, where understanding relationships between variables, identifying patterns, and tracking changes over time are essential for making informed decisions. The survey also discusses the evolution of these techniques, from static visualizations to more dynamic and interactive forms, reflecting the increasing demand for tools that can handle real-time data and provide immediate feedback to users.

One of the significant contributions of this survey is its emphasis on the importance of matching visualization techniques with the type of data being analyzed. For instance, scatterplots are effective for showing correlations between two variables, while heatmaps are better suited for representing the magnitude of values across a matrix. In financial visualization, these techniques can be applied to display correlations between different stocks, sectors, or financial indicators, helping investors identify trends and make data-driven decisions.

McNabb and Laramee's (2017) "Survey of Surveys (SoS): Mapping the Landscape of Survey Papers in Information Visualization" provides a meta-analysis of existing surveys in the field of information visualization. This work is instrumental in mapping out the breadth of research that has been conducted, identifying areas where new tools and techniques could fill gaps in the existing literature. Their analysis covers a wide range of visualization types, including those particularly relevant to financial data, such as time series visualizations, treemaps, and multi-dimensional scaling.

The meta-analysis by McNabb and Laramee (2017) underscores the importance of understanding the broader landscape of visualization research, particularly in identifying emerging trends and techniques that could be leveraged in financial applications. For example, their discussion on time series visualizations is directly applicable to stock market data, where tracking the performance of stocks over time is a fundamental task. The use of interactive time series charts, which allow users to zoom in on specific time periods or overlay different datasets for comparison, is a technique that can greatly enhance the usability of financial dashboards.

#### Visualization Techniques and Tools

The development of interactive and user-friendly visualizations in financial technology heavily relies on robust libraries such as Plotly, Dash, and PyViz. These tools facilitate the creation of dynamic, interactive dashboards that can handle large datasets and provide real-time updates, which are critical for financial applications.

In "Visualization Analysis and Design," Munzner (2014) provides a deep dive into the principles of visualization design and analysis, offering a structured approach to creating visualizations that are both effective and user-centered. Munzner emphasizes the importance of considering the user's tasks and goals when designing visualizations, which is particularly relevant in financial technology where the primary goal is to enable investors to make informed decisions quickly and efficiently.

Munzner’s work introduces several key concepts, such as the "Nested Model," which outlines the stages of designing a visualization system, from understanding the domain and data, to creating the visual encoding, and finally, evaluating the effectiveness of the visualization. This model is highly applicable to the development of the proposed tool, as it provides a framework for ensuring that the visualizations not only accurately represent the data but also support the user in achieving their goals.

Ware’s (2019) "Information Visualization: Perception for Design" complements Munzner’s work by focusing on the psychological aspects of how users perceive and interact with visualizations. Ware’s research is particularly relevant for designing financial dashboards, as it highlights the importance of using visual cues that align with human cognitive processes, such as using color to represent data magnitude or employing spatial arrangements that naturally guide the user's attention to critical information.

The application of these principles is evident in the development of color tools such as Adobe Color, Color Brewer, and Scientific Colour Maps, which are designed to enhance the visual appeal and accessibility of visualizations. These tools are especially important in financial visualizations where color is often used to indicate performance metrics, such as gains and losses, and where accessibility features like color-blind friendly palettes can ensure that the visualizations are inclusive and easy to interpret for all users.

The integration of these advanced visualization techniques and tools into the proposed investment visualization tool is expected to significantly enhance its usability, making it a powerful resource for retail investors seeking to make informed decisions based on complex financial data.

### Existing Systems

Several existing financial platforms provide data visualization and analysis tools, each with its own strengths and limitations. This section examines these platforms, including Yahoo Finance, Google Finance, Seeking Alpha, and Morningstar, to identify the gaps that the proposed tool aims to fill.

#### Yahoo Finance and Google Finance

Yahoo Finance and Google Finance are two of the most widely used financial platforms, offering users access to a broad range of financial data, including stock prices, historical performance, and basic comparison tools. These platforms provide essential visualization features such as line charts, bar charts, and pie charts, which allow users to track the performance of individual stocks and compare different assets.

A screenshot of a computer

Description automatically generated

Fig 2.2.1.1. GUI of Yahoo Finance

Yahoo Finance (Fig 2.2.1.1.), for example, uses interactive line charts that allow users to overlay different technical indicators, such as moving averages and Bollinger Bands, onto stock price data. This feature is particularly useful for technical analysis, enabling users to identify trends and make predictions based on historical data. However, while these visualizations are helpful, they are relatively basic and do not offer the level of customization or personalization that more advanced users might require.

A screenshot of a computer

Description automatically generated

Fig 2.2.1.1 User Interface of Google Finance.

Google Finance (Fig 2.1.1.1) offers similar features, including interactive line charts and comparison tools that allow users to track multiple stocks on the same graph. However, like Yahoo Finance, Google Finance's visualizations are limited in their customization options. Users cannot create highly personalized dashboards or integrate their own data into the visualizations, which limits the platform's usefulness for more sophisticated investors who need tailored insights.

#### Seeking Alpha

Seeking Alpha provides a more in-depth analysis compared to Yahoo Finance and Google Finance, with features that include expert articles, community insights, and portfolio tracking tools. The platform's visualizations are designed to complement its editorial content, offering users the ability to see visual representations of the data discussed in articles and reports.

Seeking Alpha (Fig 2.2.2.1.) uses a variety of visualization techniques, including heatmaps to display sector performance, tree maps for portfolio composition, and time series charts for tracking stock performance over time. These visualizations are interactive, allowing users to drill down into specific data points or compare different assets side by side.

A screenshot of a computer

Description automatically generated

Fig 2.2.2.1. User Interface of Seeking Alpha

One of the strengths of Seeking Alpha is its ability to combine visualizations with qualitative analysis. For instance, Seeking Alpha integrates expert opinions and community feedback with its visualizations, providing users with a more holistic view of the market. This approach helps investors not only see the data but also understand the context behind the numbers. However, despite its strengths, Seeking Alpha’s primary focus on crowd-sourced content and expert opinions means it lacks the deep personalization and data integration capabilities that some investors might need for a truly individualized analysis.

#### Morningstar

Morningstar is renowned for its comprehensive data analysis and robust research tools, offering detailed reports, ratings, and metrics for a wide range of investment vehicles including stocks, mutual funds, and ETFs. Morningstar’s (Fig. 2.2.3.1) platform is equipped with advanced visualization tools, such as performance charts that allow users to track returns over various time periods, compare different funds, and visualize the risk-return profile of investments through scatterplots and other graphical representations.

Morningstar's visualization tools are particularly strong in portfolio analysis, providing users with insights into their portfolio’s diversification, sector exposure, and asset allocation through detailed pie charts and heatmaps. These visualizations are complemented by Morningstar’s proprietary ratings, which help investors quickly assess the quality and performance potential of individual investments.

A screenshot of a computer

Description automatically generated

Fig. 2.2.3.1. User Interface of Morning Star

However, Morningstar’s platform is largely geared towards professional investors and financial advisors, offering a depth of analysis that may be overwhelming for retail investors. Additionally, access to some of the most valuable features, such as detailed reports and advanced analytics, requires a paid subscription, which can be a barrier for individual investors seeking cost-effective solutions.

#### Gaps in Existing Systems

While Yahoo Finance, Google Finance, Seeking Alpha, and Morningstar provide valuable tools for investors, there are notable gaps that the proposed investment visualization tool aims to address:

* **Lack of Personalization:** Existing platforms generally offer standardized dashboards and visualizations that are not tailored to individual users. The proposed tool will focus on integrating user-specific data, allowing for a more personalized investment analysis experience. This includes tracking personalized metrics such as individual portfolio performance, tailored buy/sell recommendations, and custom alerts based on the user’s investment goals and risk tolerance.
* **Limited Interactivity and Customization:** Although some platforms provide interactive visualizations, the level of customization available to users is often limited. The proposed tool will leverage libraries like Plotly and Dash to create highly interactive and customizable visualizations that allow users to explore their data in depth, adjust parameters on the fly, and generate insights that are specifically relevant to their investment strategies.
* **Accessibility and Usability:** Many existing systems, particularly those geared towards professional investors, can be complex and difficult to navigate for retail investors. The proposed tool will prioritize usability, drawing on principles from Ware (2019) to ensure that visualizations are intuitive and aligned with how users naturally perceive and interact with data. This focus on accessibility will help make sophisticated analytics more approachable for all investors.
* **Real-Time Data Integration:** The ability to respond quickly to market changes is crucial for investors. While platforms like Yahoo Finance and Google Finance offer real-time data, the integration of this data into meaningful, personalized insights is often lacking. The proposed tool will address this by offering real-time updates that are directly tied to user-specific alerts and recommendations, helping investors make timely decisions.

By addressing these gaps, the proposed investment visualization tool aims to enhance the investment experience for retail investors, providing them with the tools and insights they need to make informed, data-driven decisions.

### Potential Contribution

The proposed investment visualization tool seeks to make several key contributions to the field of financial technology by addressing the limitations of existing systems and enhancing the investment experience for retail investors. This section outlines the potential contributions of the tool in terms of personalized data integration, real-time interactivity, enhanced accessibility, and security.

#### Personalized Data Integration

One of the primary contributions of the proposed tool is its emphasis on personalized data integration. Unlike existing platforms, which often provide generalized insights based on market-wide data, the proposed tool will allow users to integrate their own investment data, such as individual transaction histories, portfolio allocations, and performance metrics. This integration will enable the tool to provide tailored insights and recommendations that are directly relevant to each user’s unique investment profile.

By leveraging techniques discussed in Liu et al. (2022), such as the use of APIs to pull in real-time data from financial sources like Yahoo Finance or custom user-uploaded datasets, the tool can offer a truly personalized investment analysis experience. This approach will help bridge the gap between generic financial data and the specific needs of individual investors, empowering them to make more informed decisions.

#### Real-Time Interactive Visualizations

The tool will incorporate advanced visualization techniques to offer real-time, interactive dashboards that allow users to explore their data dynamically. Using libraries such as Plotly and Dash, the tool will provide visualizations that update in real-time, reflecting market changes and new user inputs immediately. This interactivity will enable users to adjust parameters, such as the time frame of analysis or the specific metrics displayed, on the fly, making the tool highly adaptable to different investment strategies and market conditions.

These real-time capabilities will be particularly valuable for retail investors who need to monitor their investments closely and respond quickly to market changes. By providing immediate feedback on the potential impact of market movements on the user’s portfolio, the tool will support more agile decision-making, helping investors stay ahead of the curve.

#### Enhanced Accessibility and Usability

Drawing on the principles outlined by Munzner (2014) and Ware (2019), the proposed tool will prioritize accessibility and usability, making sophisticated financial analytics accessible to all investors, regardless of their level of expertise. The design will focus on clear, intuitive visualizations that present complex data in an easily understandable format, supporting better decision-making.

For example, by using heatmaps to represent portfolio performance across different sectors or asset classes, the tool can help users quickly identify areas of strength or weakness in their investments. Similarly, using tree maps to visualize portfolio composition can provide a clear overview of diversification and asset allocation, helping investors ensure that their portfolios are aligned with their risk tolerance and investment goals.

The tool will also include features such as tooltip explanations and guided walkthroughs to help users understand the visualizations and the insights they provide. By reducing the cognitive load required to interpret complex financial data, the tool aims to empower a broader audience of retail investors to engage with their investments more actively and confidently.

#### Security and Data Privacy

Given the sensitivity of financial data, the proposed tool will place a strong emphasis on security and data privacy. Implementing robust security protocols, such as encryption of data at rest and in transit, secure user authentication, and regular security audits, will be critical to protecting user information from unauthorized access and breaches.

The tool’s approach to security will be informed by best practices in the field, ensuring that users can trust the platform to safeguard their personal and financial information. This focus on security will help build user trust and encourage more widespread adoption of the tool, making it a valuable resource for retail investors.

#### Contribution to the Field of Financial Technology

Overall, the proposed tool aims to contribute to the field of financial technology by demonstrating how advanced data visualization techniques can be effectively applied to improve the investment experience for retail investors. By integrating personalized data, providing real-time interactive visualizations, and prioritizing accessibility and security, the tool will set a new standard for user-centric investment platforms.

This contribution is aligned with the broader trends identified in the literature, such as the growing demand for tools that offer both technical sophistication and ease of use. By addressing the specific needs of individual investors, the proposed tool will help democratize access to high-quality financial analytics, enabling more people to take control of their financial futures.

# Data Preparation

The success of a data-driven investment visualization tool heavily depends on the quality, accuracy, and accessibility of the underlying data. This project leverages multiple datasets that provide detailed insights into stock investments, trading activities, and company-specific financial data. These datasets, maintained and processed using Excel, form the foundation for creating dynamic and insightful visualizations that reflect the complexities of investment decisions and market dynamics. This section outlines the data sources, their characteristics, and the methods used to access and prepare the data, with a particular focus on how stock splits are adjusted using Excel formulas.

### Data Sources and Description

The primary data sources for this project include the "Investment Data 2020-2024," which details individual investment transactions, and the "Company-Centered Metadata," focusing on company-specific financial performance. These datasets, complemented by additional market data, offer a comprehensive view of investment activities over the specified period. The integration of these datasets enables a nuanced analysis of financial decisions, providing a robust foundation for developing personalized investment visualizations.

#### Original Trading Data from Trading 212

The core dataset, "Investment Data 2020-2024," is sourced from Trading 212 and spans transactions from April 2020 to December 2024. This dataset captures a detailed record of trading activities, including market buys, sells, dividend receipts, and other financial transactions, essential for understanding individual investment behaviour.

The dataset is organized into multiple sheets within an Excel workbook, each covering different years or financial categories such as dividends or capital gains. Key attributes include:

* + - * + **No. *(Int)***: A simple numeric identifier for each record.
        + **Account number *(Int)***: The ISA is tax-free and the INV account may be taxed.
        + **Action *(String)***: Describes the transaction type, including Market Buy, Market Sell, various Dividend types, Interest on cash, Deposit, and Withdrawal.
        + **Transaction Date *(Date)***: The date on which the transaction occurred.
        + **Time *(Time)***: The exact time of the transaction.
        + **Ticker *(String)***: Stock ticker symbol used to identify a specific stock.
        + **Name *(String)***: Name of the stock, usually the company name.
        + **No. of shares *(Float)***: - The number of shares involved in the transaction.
        + **No. of shares adjusted for split(s):** The number of shares involved in the transaction calculated with help of stock split ratio.
        + **Price / share *(Float)***: Transaction price per share.
        + **Price / share adjusted for split(s):** Transaction price per share calculated with help of stock split ratio.
        + **Currency (Price / share) *(String)***: Currency used for the transaction price.
        + **Exchange rate *(Float)***: Rate used to convert the transaction currency to GBP.
        + **Result (GBP) *(Float)***: Profit or loss from selling stocks, in British pounds.
        + **Total (GBP) *(Float)***: Total transaction amount in British pounds.
        + **Withholding tax *(Float)***: Tax withheld at the time of dividend distribution.
        + **Currency (Withholding tax) *(String)***: Currency of the withholding tax.
        + **Charge amount (GBP) *(Float)***: Additional charges with the transaction.
        + **Deposit fee (GBP) *(Float)***: Fee charged for deposits.
        + **Transaction fee (GBP) *(Float)***: Fee that may apply to 'Market Sell' actions.
        + **Finra fee (GBP) *(Float)***: Regulatory fees charged by financial authorities.
        + **Currency conversion fee (GBP) *(Float)***: Fees for converting currencies.
        + **Stamp duty reserve tax (GBP) *(Float)***: A government tax on the transaction.
        + **ID *(Int)***: Unique identification number of the transaction.
        + **Notes *(String)***: Any additional information about the transaction.

This dataset is managed and analysed using Excel, where data is cleaned, formatted, and processed using a series of formulas. These include lookups, conditional formatting, and calculations that ensure consistency across different years and transaction types, allowing for accurate comparisons and aggregations.

A screenshot of a computer

Description automatically generated

Fig. 3.1.1.1:*A screenshot of the* “*Investment Transaction Dataset”.*

#### Company-Centered Metadata

The "Company-Centered Metadata" dataset provides a detailed view of company-specific metrics derived from the main trading dataset. It includes performance metrics for 83 stocks, detailing realized and unrealized gains, dividend distributions, and other key financial metrics related to each company.

Key attributes of this dataset include:

* **Company Name, Ticker Symbol, and Dates of Transactions**: Identifiers that provide a timeline of trading activities, including the first and last purchase or sale dates.
* **Total Shares Purchased and Sold**: A summary of trading volumes per stock, allowing for an analysis of investor activity and stock liquidity. Also added similar columns to recalculate values after split.
* **Average Purchase and Sale Prices**: Calculated metrics that offer insights into the cost basis and profitability of trades. Also added similar columns to recalculate values after split.
* **Current Share Price and Dividends**: Current market prices and total dividends received, contributing to the overall return calculations.
* **Realized and Unrealized Gains/Losses**: Financial performance metrics providing a snapshot of both locked-in and potential future profits or losses. Also added similar columns to recalculate values after split.
* **Stock Splits**: Entries that record stock splits, including the split ratio and date, are critical for adjusting share counts and recalculating per-share metrics.

A screenshot of a computer

Description automatically generated

Fig. 3.1.1.2: *A screenshot of the* “*Company-Centred Metadata”.*

Excel formulas are extensively used to derive these attributes, including functions such as AVERAGE, SUMIF, VLOOKUP, and conditional logic formulas to calculate the performance metrics based on the raw data. The use of Excel allows for a flexible and user-friendly approach to data management, making it easy to update, review, and correct calculations as needed.

### Accessing Data

The datasets are stored in Google Sheets, offering a cloud-based solution that enhances accessibility and ensures data integrity. This setup facilitates real-time updates and easy sharing, allowing for seamless integration of the data into the visualization tool. Using Google Sheets also allows the application of Excel-like formulas directly within the cloud environment, maintaining consistency in data processing and calculations.

The data retrieval process involves accessing specific sheets within the Google Sheets document, each corresponding to different data segments such as individual transactions or company-specific metrics. The tool automates this data access through API calls, ensuring that the most current data is always used in the visualizations.

For additional real-time market data, such as stock prices and historical performance, the tool integrates with external sources like Yahoo Finance APIs. This external data complements the user-provided investment data, allowing for enriched analyses that combine personal trading history with broader market trends.

### Stock Splits and Their Adjustments

Stock splits represent a critical adjustment within the datasets, as they affect share counts, per-share metrics, and overall portfolio valuation. A stock split increases the number of outstanding shares by issuing more shares to current shareholders, typically without changing the company's market capitalization. This action effectively reduces the stock's price while keeping the overall value of holdings unchanged, necessitating careful adjustments in data calculations.

#### Understanding Stock Splits in the Dataset

In the "Investment Data 2020-2024" dataset, stock splits are recorded as specific transactions, with detailed entries that include the split date and ratio (e.g., 2-for-1, 3-for-2). These entries are crucial for ensuring the historical accuracy of share counts and stock price data, as they impact the valuation of portfolio holdings over time.

Stock splits are particularly important when visualizing long-term trends, as failing to adjust for splits can lead to misleading representations of stock performance. For instance, without adjustments, a stock that undergoes a 2-for-1 split would appear to have halved in price, potentially skewing performance charts and misleading investors.

#### Adjusting Data for Stock Splits Using Excel

All adjustments for stock splits are handled using Excel formulas within the datasets. Key adjustments include:

* **Updating Share Counts**: Share counts are adjusted using multiplication formulas based on the split ratio (e.g., multiplying the existing share count by 2 for a 2-for-1 split). This ensures that all subsequent calculations reflect the increased number of shares.

Example Formula: =Original\_Share\_Count \* Split\_Ratio

* **Adjusting Prices**: Historical prices are adjusted inversely to the split ratio, ensuring continuity in price trends over time. This adjustment maintains the integrity of performance analyses and supports accurate visual representations in the tool.

Example Formula: =Original\_Price / Split\_Ratio

* **Recalculating Averages and Totals**: Excel formulas are used to recalculate average purchase prices, total investment values, and other metrics that are impacted by the adjusted share counts and prices. This ensures that all data remains consistent and accurately reflects the investor’s portfolio performance.

Example Formula:

**Average Purchase Price**: =SUMIF(Transaction\_Type, "Buy", Total\_Value) / SUMIF(Transaction\_Type, "Buy", Shares\_Adjusted)

**Total Value**: =SUM(Adjusted\_Shares \* Adjusted\_Price)

* **Consistency Across Datasets**: Adjustments are applied uniformly across all relevant datasets, including the transaction data, company-centered metadata, and any external market data. This approach ensures that all aspects of the visualization tool reflect the correct values post-split.

# Project Specification

This Project Specification provides an in-depth overview of the essential components and technology choices that underpin the development of the investment visualization tool. Section 3.1 outlines the feature specifications, detailing both must-have and optional features that define the core and enhanced functionalities of the tool. Section 3.2 discusses the technology choices, focusing on the programming languages, libraries, and additional technologies utilized in building the system.

### Feature Specification

The feature specification section describes the functionalities of the investment visualization tool, ensuring it meets its objectives of providing dynamic and interactive insights into stock investments based on the user's data.

#### Must-have Features

The must-have features establish the foundation of the investment visualization tool, enabling its basic operation and ensuring it provides meaningful insights into financial data.

1. **Interactive Dashboard Application:** The system is designed as an interactive dashboard application that allows users to dynamically visualize financial data. This dashboard serves as the main interface for users to explore their investment information through various charts and controls.
2. **Multiple Line Chart for Stock Trends:** A multiple line chart is a central feature of the dashboard, displaying trends for all stocks included in the "Investment Data 2020-2024." This chart visualizes historical price data, buy and sell points, and other key metrics, allowing users to observe how their investments have performed over time.
3. **Real-time Data Integration from Finance APIs:** To keep the data current, the tool fetches stock history data from finance APIs, such as Yahoo Finance. This integration ensures that users can analyze real-time market trends alongside their historical transaction data, providing a comprehensive view of their investments.
4. **Buy and Sell Points Visualization:** The multiple line chart includes markers for buy and sell points according to the transaction records in the Investment Transaction Dataset. These markers help users identify specific transactions on the chart, making it easier to correlate buying and selling actions with stock price movements.
5. **Category Selection for Line Chart Plotting:** Users have the option to select different categories for plotting line charts, such as company, sector, or industry. This feature allows for multiple companies, sectors, or industries to be plotted simultaneously for comparative analysis, providing deeper insights into how different segments perform relative to one another.
6. **Plotting of Specific Data Points:** Users can plot various types of data on the line charts, including historical prices, buy actions, sell actions, closed positions, and new positions. This flexibility allows users to tailor the visualizations to their specific needs, focusing on the aspects of their investment strategy that matter most.
7. **Background Color Customization:** The dashboard includes a user option to choose between black or white background colors for the line charts. This feature enhances the user experience by accommodating personal preferences and improving readability in different lighting conditions.
8. **Moving Average Adjustment:** A dropdown is provided for users to adjust the moving average values displayed on the multiple line chart, with a default setting of 50 days. This feature helps users smooth out short-term fluctuations in stock prices, allowing them to focus on long-term trends.
9. **Hierarchical Color Mapping:** The tool employs hierarchical color mapping for sectors and industries, using primary colors for sectors and different shades of those colors for industries within each sector. This approach makes it easy for users to visually distinguish between different sectors and their constituent industries.
10. **Performance-based Color Mapping for Tickers:** Tickers are color-coded based on the performance of each stock, with distinct colors representing profit or loss. This visual cue allows users to quickly assess which investments are performing well and which are underperforming.
11. **Decision Analysis Options:** Users have the ability to analyze decisions from both a future-oriented and past-oriented perspective. This feature supports different types of investment strategies, allowing users to look back at historical decisions or project potential outcomes based on current data.
12. **Focus and Context Display:** The tool includes a feature that prints all data points in grayscale by default, with the selected company, industry, or sector highlighted in color. This focus-and-context approach helps users maintain an overview of the entire dataset while focusing on specific areas of interest.
13. **Scaling by Transaction Amount:** Users can scale the size of buy and sell markers according to the total transaction amount. This feature visually represents the magnitude of each transaction, helping users quickly identify significant trades and assess their impact on overall investment performance.

#### Optional Features

The optional features enhance the core functionalities of the tool, offering additional capabilities that improve the user experience and provide more in-depth analysis options.

1. **Zooming and Box Selection Functionalities:** All charts in the dashboard could include zooming and box selection functionalities, allowing users to focus on specific parts of the data. These features enable users to zoom in on detailed sections of the charts and investigate specific data points with greater precision.
2. **Informative Tooltips:** An informative tooltip feature could be implemented, providing detailed information when the cursor hovers over any graphical element on the charts. This tooltip would display relevant data such as stock names, dates, prices, and transaction details, enhancing the user's ability to interpret the data effectively.

### Technology Choices

The technology choices for this project were carefully selected to ensure that the tool is robust, user-friendly, and capable of delivering dynamic, real-time visualizations. This section details the key technologies used in the implementation of the tool.

#### Programming Languages

In the realm of software development, particularly for projects that require sophisticated data visualization capabilities, selecting the appropriate programming language is pivotal. The choice of language significantly impacts the development process, system performance, scalability, and user accessibility of the application. For this project, which aims to create an interactive, customer-focused visualization tool for stock investments, Python was chosen as the primary programming language due to several compelling reasons, reinforced by consultations with domain experts and supervisors.

**Python: The Chosen Language**

Python was selected for its versatility, extensive ecosystem, and strong support for data manipulation and visualization. Python’s simplicity and readability make it an ideal choice for rapid development, while its rich set of libraries, such as Pandas for data processing, NumPy for numerical computations, and Plotly for creating interactive visualizations, provide the necessary tools for developing robust, user-centric applications. The Dash framework, built on top of Flask, Plotly, and React, allows for the creation of interactive web applications directly from Python, which perfectly aligns with the project’s goal of building a dynamic and responsive dashboard for investment analysis.

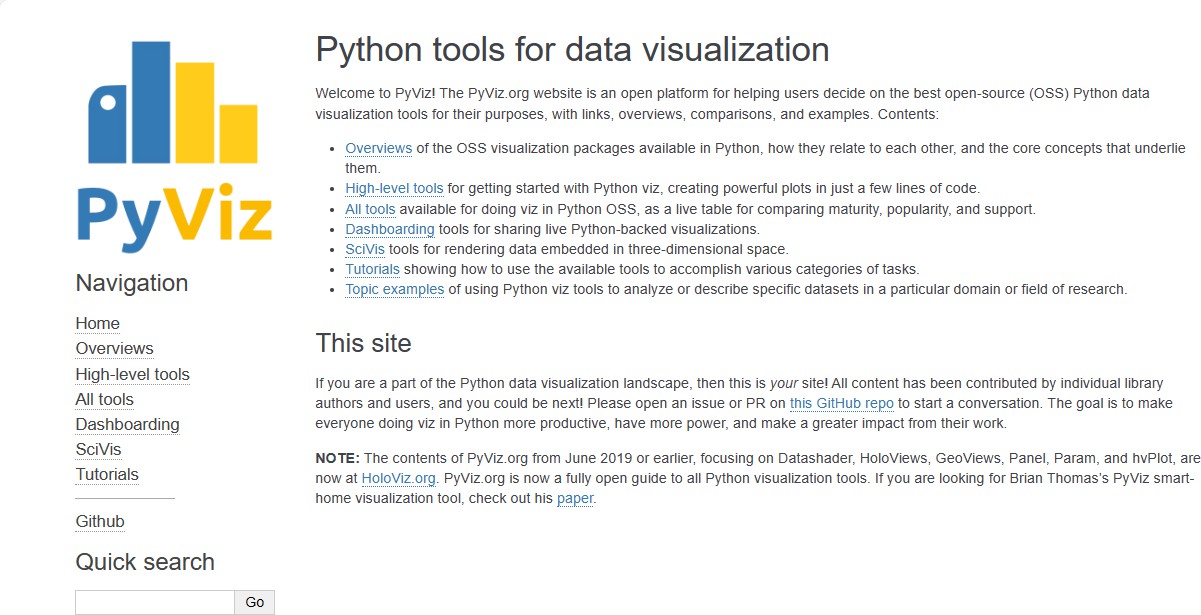
Python also offers cross-platform compatibility, allowing the application to run seamlessly on various operating systems, including Windows, macOS, and Linux. This flexibility ensures that the tool can be easily deployed and accessed by users with different system configurations. Furthermore, Python’s open-source nature and strong community support facilitate a cost-effective development process, providing access to a wealth of resources, tutorials, and forums that can expedite troubleshooting and development.

**Considered Alternatives**

1. **Java:** Java was considered for its strong capabilities in building enterprise-grade applications and its well-established ecosystem for developing graphical user interfaces using libraries like JavaFX and Swing. However, Java's strength lies primarily in desktop applications rather than web-based, interactive applications, which made it less suitable for this project’s needs. Moreover, Java’s verbosity compared to Python could slow down the development process, especially for rapid prototyping and iterative improvements.
2. **R:** Known for its powerful statistical and data analysis capabilities, R was another candidate due to its strengths in data visualization through libraries like ggplot2 and Shiny. However, R's focus is predominantly on statistical computing and less on building full-fledged interactive web applications. While R can be extended with web frameworks like Shiny, it does not offer the same level of integration with modern web technologies as Python with Dash, making it a less optimal choice for a project centered around web-based user interactions.
3. **JavaScript and TypeScript:** Both JavaScript and TypeScript are critical languages for web development and could have been viable options for building highly interactive client-side applications. JavaScript, particularly when combined with frameworks like React, Vue, or Angular, excels at creating dynamic front-end experiences. However, these languages are primarily suited for client-side scripting, and while they could handle the visualization aspects, they lack the robust server-side data processing and manipulation capabilities provided by Python’s ecosystem. This project’s need for extensive data processing and server-side logic made Python the preferred choice.
4. **MATLAB:** MATLAB is another language known for its strong data visualization and numerical analysis capabilities. It offers a comprehensive suite of tools for data analysis and visualization, making it popular in academic and research settings. However, MATLAB is not as well-suited for web application development, and its licensing costs can be prohibitive, especially when compared to Python’s open-source environment. Additionally, the integration of MATLAB with web technologies is less seamless compared to Python, which offers dedicated frameworks like Dash for web development.
5. **Ruby on Rails:** While Ruby on Rails is an excellent framework for building web applications, it is generally geared towards building CRUD (Create, Read, Update, Delete) applications and does not inherently specialize in data visualization or numerical processing. Although Ruby has libraries like Gruff and Chartkick for generating visualizations, they are not as mature or extensive as Python’s offerings. Additionally, Python’s integration capabilities with data processing and machine learning tools make it a superior choice for this project.

**Why Python Stands Out?**

Python's ecosystem is uniquely positioned to support the development of complex data-driven applications due to its extensive library support and ease of integration with various data sources and formats. Libraries like Plotly and Dash are designed specifically to build interactive dashboards with minimal coding effort, allowing developers to create user-friendly, dynamic visualizations that are easily accessible through web browsers. Additionally, Python's capability to work seamlessly with Excel through libraries such as Openpyxl and Pandas ensures that data management and manipulation tasks are handled efficiently, which is critical for this project where data integrity and real-time updates are paramount.



**Fig. 4.2.1.1:** *A screenshot of the* “*PyViz” tools by python, (PyViz, 2024).*

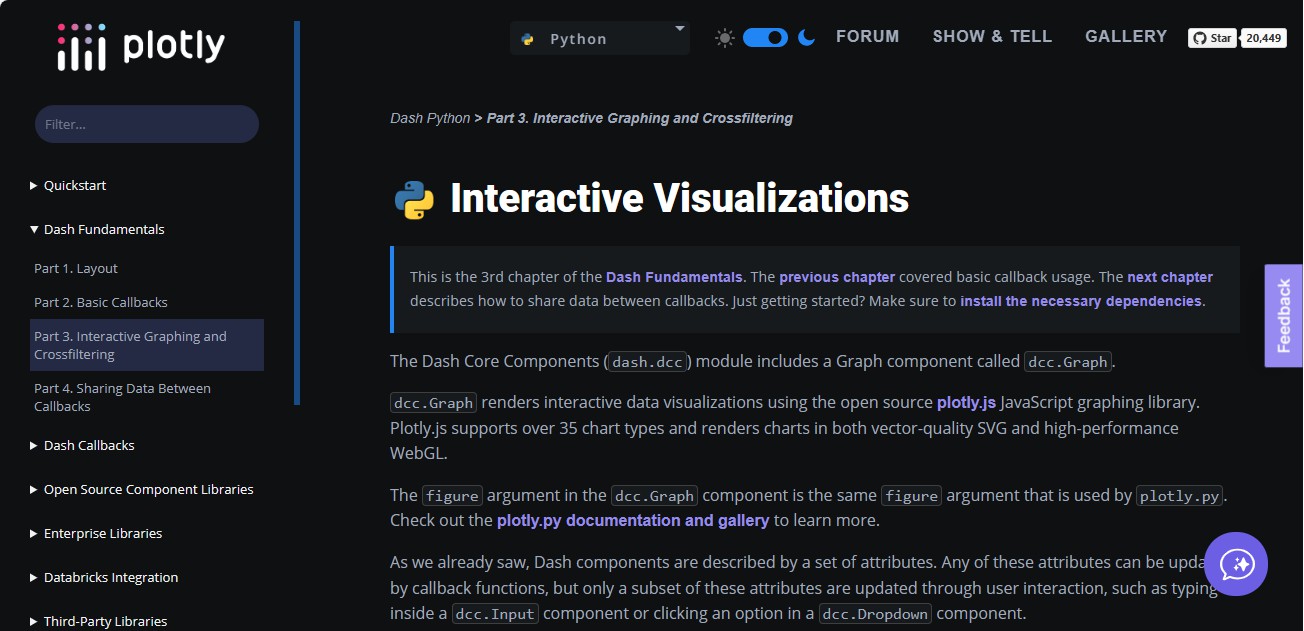
The ease of Python’s syntax, combined with its powerful visualization capabilities, makes it an ideal choice for developing a tool that needs to balance technical complexity with user accessibility. Its scalability also ensures that as the project evolves, additional functionalities can be integrated without significant refactoring, thus future-proofing the tool against changing user requirements and market dynamics.

In conclusion, Python was chosen as the primary language for this project due to its robust capabilities in data manipulation, extensive library support, and superior frameworks for building interactive, web-based applications. Its balance of simplicity and power makes it the best fit for creating a sophisticated, user-centric investment visualization tool that is both effective and accessible.

#### Libraries

For the development of a dynamic and user-friendly stock investment visualization tool, a range of libraries were selected to provide robust functionality and ensure seamless integration within the Python ecosystem. Below is a detailed description of each library utilized in the project, highlighting their specific roles and contributions.

* **Dash:** Dash is the primary framework used for building interactive web applications. Developed by Plotly (Fig 4.2.2.1), Dash is specifically designed for creating web-based data dashboards using Python. It allows developers to build complex, interactive data visualizations without requiring extensive knowledge of front-end technologies like HTML, CSS, or JavaScript. Dash simplifies the process of connecting Python-based back-end data processing with a responsive and interactive front-end, making it an ideal choice for data-driven web applications. Its extensive component library, including dropdowns, sliders, and graphs, provides the flexibility needed to create highly customized user interfaces.
* **Dash Core Components (dcc) and HTML Components:** Dash Core Components (dcc) and HTML Components are essential modules within Dash that provide the building blocks for creating interactive user interfaces. The dcc module includes higher-level UI elements such as graphs, dropdowns, and sliders, while HTML Components allow the inclusion of standard HTML elements like headers, paragraphs, and buttons. These components enable the construction of a structured and intuitive layout, facilitating user interaction and data exploration within the application.
* **Plotly Graph Objects (go):** Plotly Graph Objects (go) is a module within Plotly (Fig 4.2.2.1) that provides low-level control over the creation of detailed and customizable charts and visualizations. It allows for the construction of a wide variety of graphs, including line charts, bar charts, and scatter plots, with extensive customization options for axes, markers, colors, and annotations. This level of detail is crucial for developing high-quality, interactive visualizations that effectively convey complex investment data.



**Fig 4.2.2.1:** *A screenshot of the* “*Plotly” library provided by python, (Plotly, 2024).*

* **Plotly Express:** Plotly Express is a high-level interface for Plotly that simplifies the creation of common visualizations with minimal code. It is particularly useful for quickly generating plots like scatter plots, line charts, and bar graphs during the early stages of development or for creating exploratory visualizations that can be easily refined into more complex Dash components. Plotly Express’s intuitive syntax and built-in support for handling data frames make it an efficient tool for rapid visualization prototyping.
* **Pandas:** Pandas is a powerful data manipulation and analysis library that plays a central role in this project. It provides data structures like Data Frames, which are ideal for handling tabular data, performing data cleaning, aggregation, and transformation. Pandas simplifies the integration and processing of investment transaction data, enabling efficient manipulation of large datasets. Its ability to merge, filter, and pivot data ensures that the information presented in the visualizations is accurate and relevant to the user’s needs.
* **YFinance:** YFinance is a Python library used to fetch historical and real-time market data from Yahoo Finance. It provides an easy interface for accessing a wide range of financial data, including stock prices, volumes, and fundamental metrics, which are essential for the analysis and visualization of investment trends. YFinance’s seamless integration with Pandas allows for direct manipulation and analysis of the fetched data, making it a valuable tool for maintaining the currency and accuracy of the financial insights presented in the dashboard.
* **NumPy:** NumPy is a fundamental package for numerical computing in Python, providing support for arrays, matrices, and a collection of mathematical functions to operate on these data structures. In this project, NumPy is used to perform numerical calculations, such as computing moving averages and other statistical measures that help in analyzing stock performance. Its efficient handling of large numerical datasets is crucial for processing the financial data needed for the visualizations.
* **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive plots in Python. Although Plotly is the primary visualization library for this project, Matplotlib is utilized during the development phase for quick, exploratory data analysis and validation of results. It offers a simple way to generate plots for testing and debugging purposes before integrating the final visualizations into the Dash application.

By integrating these libraries, the project benefits from a powerful and flexible set of tools that support the creation of an interactive, data-driven web application. Each library contributes specific functionalities that collectively enhance the user experience, performance, and analytical capabilities of the investment visualization tool.

#### Other Technologies

* **Financial APIs:**

Incorporating real-time financial data is critical for the functionality of the investment visualization tool, allowing users to access the latest market information and integrate it with their personal investment data. Several financial APIs were considered for this project, each offering distinct features and capabilities.

* 1. **Yahoo Finance API:** Yahoo Finance API was selected as the primary source for real-time financial data in this project. It provides comprehensive access to a wide range of financial metrics, including historical prices, real-time quotes, company information, and market news. Yahoo Finance is widely regarded for its accuracy and reliability, and it offers a free tier that meets the data needs of this project without incurring additional costs. Furthermore, the API's robust documentation and ease of integration with Python through the yfinance library streamline the development process, allowing for seamless data retrieval and minimal setup.
  2. **Google Finance API:** Google Finance API was another potential candidate, known for its robust data offerings and integration capabilities. However, its API services have been limited in recent years, with reduced support and less frequent updates compared to other alternatives. The inconsistency in data availability and the complexity involved in retrieving detailed financial metrics made it a less favorable choice for this project.
  3. **Alpha Vantage API:** Alpha Vantage API offers extensive data coverage, including stocks, forex, and cryptocurrencies. It provides a generous free tier and supports a wide range of financial metrics. However, the API has limitations on call frequency, which can hinder its usability for applications requiring high-frequency updates. Additionally, while Alpha Vantage provides extensive historical data, its real-time data accuracy can sometimes lag, making it less reliable for time-sensitive financial analyses.
  4. **IEX Cloud API:** IEX Cloud API provides comprehensive financial data, including stock prices, financial statements, and news. It is known for its high data accuracy and a robust free tier that supports basic data needs. However, access to more advanced datasets and higher request volumes requires a paid subscription, which could exceed the project’s budget constraints. The API’s pricing model, combined with its extensive but sometimes complex setup, made it a less attractive option compared to Yahoo Finance.
  5. **Polygon.io API:** Polygon.io is recognized for its high precision and extensive financial data coverage, including real-time data, market news, and company financials. Its free tier provides basic access, but like IEX Cloud, advanced features and higher data limits are gated behind paid plans. The detailed and precise data offered by Polygon.io makes it suitable for professional trading and analytics platforms, but its cost structure and integration complexity were not aligned with the project’s requirements for a cost-effective solution.

**Why Yahoo Finance API?**

Yahoo Finance API was chosen as the optimal solution due to its balance of comprehensive data offerings, ease of integration, and cost-effectiveness. The yfinance library simplifies the process of accessing Yahoo Finance data directly within Python, providing seamless access to both historical and real-time market data, which is critical for maintaining the currency of the tool’s visualizations. Yahoo Finance's reputation for accuracy and its active support community further enhance its appeal, ensuring that any issues or queries can be quickly addressed.

Additionally, Yahoo Finance’s API aligns well with the project’s need for a robust yet accessible data source that supports real-time updates without imposing additional costs. This integration enables the tool to provide users with timely and relevant market insights, enhancing the decision-making process and delivering a superior user experience.

Overall, Yahoo Finance API’s comprehensive feature set, combined with its ease of use and free access, make it the ideal choice for integrating real-time financial data into the investment visualization tool, supporting the project's goal of delivering a dynamic and user-centric application.

* **GitHub:**

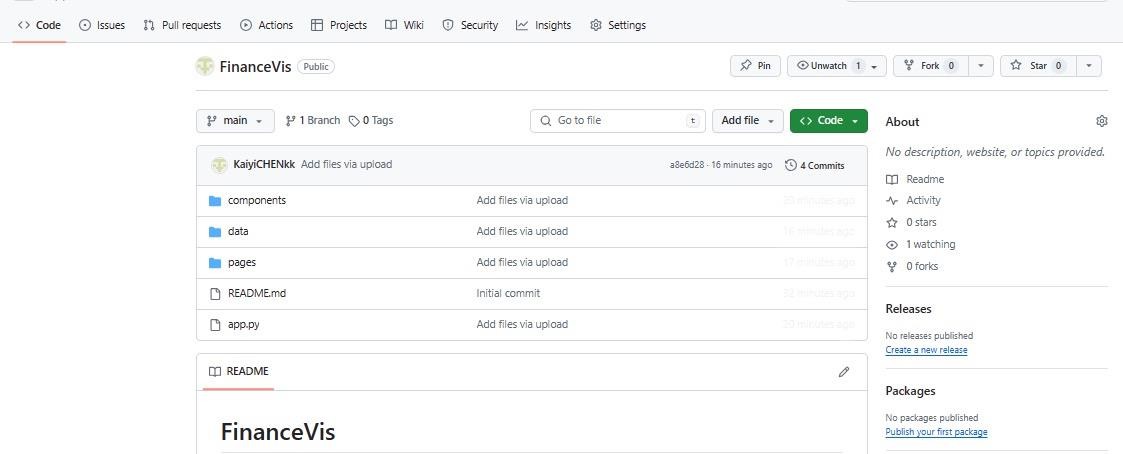
GitHub (Figure 3.2.3.1) serves as a powerful platform that utilizes Git, one of the most widely adopted version control systems globally. It offers an effective solution for tracking changes, managing code versions, and organizing code branches for feature development and testing purposes. Incorporating GitHub into the project’s workflow enhances the overall development process by providing a structured and collaborative environment that promotes high-quality code management. This integration underscores a commitment to software development best practices, highlighting the value of open-source collaboration and community engagement. The project's codebase is maintained on GitHub, reflecting a dedication to transparency and continuous improvement throughout the development lifecycle.

Fig. 4.2.3.1.: A Screenshot of GitHub Repository *(GitHub, 2024).*

* **Visual Studio Code:**

Visual Studio Code (VS Code) was chosen as the primary development environment due to its comprehensive support for coding, compiling, and managing projects efficiently. It offers an extensive range of features, including build automation, integrated testing frameworks, and deployment tools, which streamline the development process. VS Code’s built-in Git integration facilitates smooth version control, allowing easy management of code changes and collaboration among team members. Additionally, its integrated terminal simplifies command-line operations, enhancing productivity by enabling seamless execution of scripts and commands. The editor's compatibility with package management tools such as PIP further optimizes the development workflow, making it highly suitable for coding, testing, and deploying the advanced visualizations essential for this financial analysis project.

A screenshot of a computer program

Description automatically generated

Fig. 4.2.3.2.: A Screenshot of code on Visual Studio Code.

* **PIP**

PIP, Python's package management system, is essential for managing software libraries and dependencies within the project. As the official installer for Python packages, PIP offers easy access to the Python Package Index (PyPI), which contains a vast array of libraries and modules required for the project. This tool is crucial for integrating various Python packages, such as Pandas for data manipulation, Plotly for interactive visualizations, and Dash for developing web applications. It ensures that all components are up-to-date and compatible, thereby enhancing the project’s overall efficiency. By streamlining the installation and updating of libraries, PIP simplifies the development workflow, allowing the team to concentrate on delivering a high-quality financial visualization tool that meets user needs effectively.

# Project Plan and Timetable

This section outlines the detailed project plan and timetable for developing the "Invest Smart: A Consumer-Centric Investment Visualization Tool". The project plan is structured to ensure systematic progress through key phases, including planning, development, testing, and deployment, aligned with the guidelines provided and detailed in the project plan document.

The timetable of this project is illustrated in the Gantt Charts below. To facilitate precise tracking of the schedule, according to Bob’s Minutes of Meeting Protocol, comprehensive details on the plan and weekly progress can be fetched from the documented weekly meeting [MSc Minutes of Meeting 2024](https://docs.google.com/document/d/1Wh5JyEXqSEILUPMa7i6gB4H5ReGjPPyc05y7wxni0zk/edit#heading=h.eenmnoe2v7f8) and [FinVis Minutes of Meeting](https://docs.google.com/document/d/1jvSTDCbzuLoCJahr0fWbhZxJNiqvHVj1KQ0yNIy46b0/edit#heading=h.eenmnoe2v7f8).

### Project Plan

The project is divided into seven work packages (WPs), each with specific objectives, activities, and milestones. This structured approach facilitates efficient management of the development process and ensures timely delivery of the project’s goals.

**Work Package 1 (WP1) – Literature Review**

* **Activity**: Conduct a comprehensive review of existing financial visualization tools and techniques. This includes an analysis of their strengths, weaknesses, and the gaps they leave unaddressed.
* **Milestone**: Completion of the literature review chapter of the dissertation.

**Work Package 2 (WP2) – Data Collection and Preparation**

* **Activity**: Collect and preprocess investment transaction data for analysis. This includes cleaning the data, handling missing values, and ensuring data is in a format suitable for analysis.
* **Milestone**: Data collection and preprocessing completed. A cleaned and ready-to-use dataset.

**Work Package 3 (WP3) – Tool Development**

* **Activity**: Develop the core functionalities of the visualization tool using Python and relevant libraries such as Matplotlib, Plotly, or D3.js. This includes creating basic visualizations like line charts, bar charts, and scatter plots.
* **Milestone**: Basic tool functionality developed, including initial visualizations and backend setup.

**Work Package 4 (WP4) – Integration of Real-Time Data**

* **Activity**: Integrate real-time financial data from APIs (e.g., Alpha Vantage, Yahoo Finance API) into the tool. Ensure the tool can update visualizations dynamically as new data comes in.
* **Milestone**: Real-time data integration completed, with live updating visualizations.

**Work Package 5 (WP5) – User Interface Design**

* **Activity**: Design and implement an intuitive user interface (UI) for the tool. This includes user experience (UX) considerations to make the tool easy to navigate and use.
* **Milestone**: User interface design finalized and implemented, ready for user testing.

**Work Package 6 (WP6) – Testing and Evaluation**

* **Activity**: Conduct user testing and case studies to evaluate the tool’s effectiveness and usability. Collect feedback and iterate on the design and functionality based on user input.
* **Milestone**: Evaluation results documented, including user feedback and suggestions for improvement.

**Work Package 7 (WP7) – Final Write-Up**

* **Activity**: Complete the final write-up of the dissertation, including all findings, evaluations, and detailed documentation of the development process. Ensure the dissertation meets all academic requirements.
* **Milestone**: Dissertation completed and submitted. This includes a final presentation.

**Summary of Work Packages:**

| Work Package | Start Date | End Date | Duration |
| --- | --- | --- | --- |
| WP1: Literature Review | July 5 | July 12 | 8 days |
| WP2: Data Collection and Preparation | July 13 | July 22 | 10 days |
| WP3: Tool Development | July 23 | August 7 | 15 days |
| WP4: Integration of Real-Time Data | August 8 | August 15 | 8 days |
| WP5: User Interface Design | August 16 | August 23 | 8 days |
| WP6: Testing and Evaluation | August 24 | August 28 | 5 days |
| WP7: Final Write-Up | August 29 | September 1 | 4 days |

### Development Approach

The development approach is based on Agile principles, emphasizing iterative development, continuous feedback, and flexibility to adapt to changes. This methodology supports the project's need for rapid prototyping and ongoing refinement based on guide feedback.

**Agile Methodology:**

The Agile methodology is characterized by its focus on iterative development and incremental progress, which allows the project to deliver functional components early and frequently. The process involves short development cycles called sprints, each dedicated to building a specific set of features or improvements. This iterative approach enables the team to regularly assess the tool’s progress, incorporate feedback, and make necessary adjustments to the design and functionality.

1. **Iterative Development:** The project is divided into short iterations or sprints, each focusing on developing a specific set of features. This allows for regular reviews and adjustments, ensuring that the tool evolves in alignment with user needs and project goals.
2. **Regular Feedback Cycles:** Stakeholders and potential users are engaged throughout the development process, providing continuous feedback that informs design and functionality adjustments. This ensures that the tool remains user-centric and responsive to changing requirements.
3. **Collaboration and Communication:** Regular team meetings, sprint reviews, and retrospectives foster a collaborative environment. These practices ensure that all team members are aligned, any challenges are promptly addressed, and the project remains on track.
4. **Flexibility and Adaptability:** The Agile approach allows the project to respond effectively to changing requirements or unforeseen challenges, ensuring that the final product meets user expectations and project goals. This adaptability is crucial in the context of developing a dynamic tool like "Invest Smart," where market data and user needs can evolve rapidly.

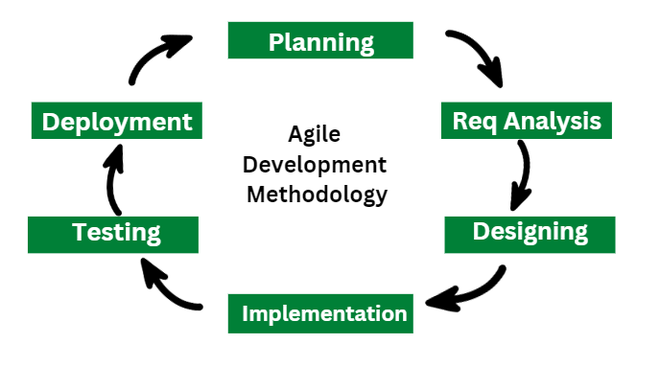


Fig. 5.2.1: Agile Workflow Diagram by [GeeksforGeeks](https://www.geeksforgeeks.org/what-is-agile-methodology/).

The Agile methodology is visually represented in the Agile Workflow Diagram from [GeeksforGeeks (2024)](https://www.geeksforgeeks.org/what-is-agile-methodology/), (Fig. 5.2.1) which illustrates the cyclical nature of Agile development. The diagram emphasizes the flow of activities from planning and development to testing, feedback, and deployment, highlighting the continuous cycle of improvement that defines Agile projects. This visual tool underscores how the iterative process supports ongoing refinement and adaptation, which are key to the successful delivery of a user-centric investment visualization platform.

**Tools and Techniques:**

1. **Version Control with GitHub:** GitHub is used for version control, providing a robust platform for tracking changes, managing branches, and facilitating collaboration among team members. This ensures that all code changes are documented, and that the development process remains organized and transparent.
2. **Continuous Integration and Testing:** Automated testing and continuous integration pipelines are set up to ensure that new code commits do not introduce bugs or regressions. This approach maintains code quality and stability throughout development, allowing the team to identify and address issues early.
3. **Task Management:** Project management tools, such as Jira, are used to track tasks, manage the backlog, and prioritize work items. These tools help the team stay organized and focused on delivering high-value features in each iteration, ensuring efficient use of resources and time.

By adopting Agile principles and leveraging the tools mentioned above, the project is positioned to deliver a high-quality, user-centric investment visualization platform that meets the needs of retail investors. The Agile methodology, as depicted in the diagram from GeeksforGeeks, reinforces the project's commitment to continuous improvement and adaptability, which are critical to the successful completion of the "Invest Smart" tool.

This Agile workflow diagram serves as a visual guide to the iterative development process, illustrating how the project progresses through repeated cycles of planning, development, testing, and refinement, ultimately leading to the delivery of a robust and responsive software solution.

**Gantt Chart:**

The Gantt chart (Fig. 5.2.2) below visually represents the project timeline, highlighting the start and end dates for each work package, along with their respective durations. This chart serves as a visual guide to monitor project progress and ensure all tasks are completed on schedule.

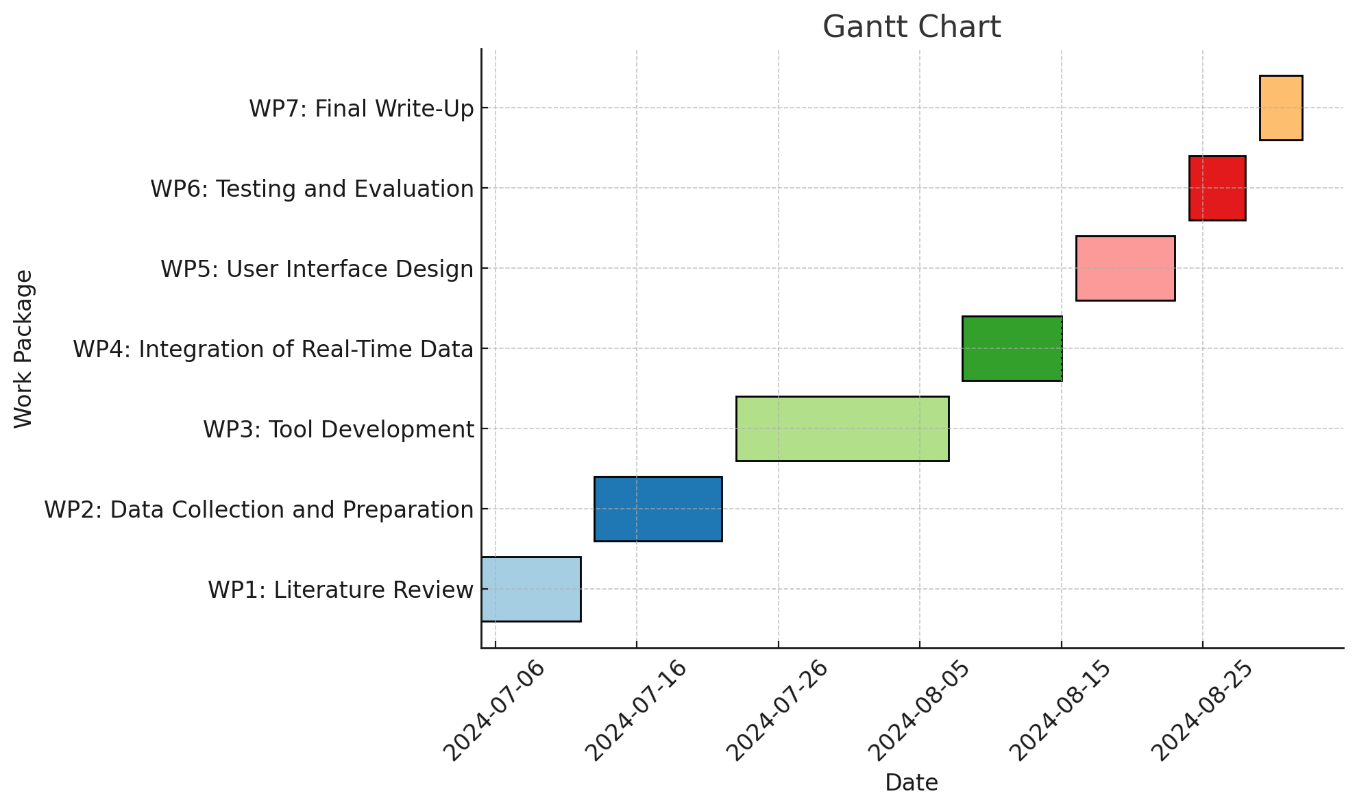


Fig 5.2.2: Gantt Chart Representing Project Timeline.

This comprehensive project plan and timetable provide a clear roadmap for the development of the investment visualization tool, ensuring all milestones are met and the project is delivered successfully within the allocated timeframe.

# Project Design

This section outlines the design of the "Invest Smart" tool, focusing on the visualization pipeline and process diagram that guide the flow of data from acquisition to final display. The design ensures that the system is robust, user-friendly, and capable of handling dynamic financial data effectively.

### Pipeline in Visualization

The visualization pipeline for the "Invest Smart" tool transforms raw investment data into actionable insights through a series of structured stages. The pipeline follows the principles outlined in Wang et al. (2016), emphasizing a systematic approach to data processing and visualization. This approach ensures that each stage, from data acquisition to rendering, contributes to creating a seamless and informative user experience.

**Stages of the Visualization Pipeline:**

1. **Data Acquisition:** The first stage involves collecting raw investment data from user-uploaded Excel files and real-time financial data from the Yahoo Finance API. The tool is designed to automatically access the latest data through API calls, while also allowing users to upload their historical data files. This setup ensures that the tool always has access to the most current and relevant data.
2. **Data Preprocessing:** In this stage, the acquired data is pre-processed using Excel formulas and Python libraries. Key preprocessing tasks include:
   * **Data Cleaning:** Removing duplicates, handling missing values, and standardizing formats.
   * **Data Transformation:** Calculating new metrics such as moving averages, adjusting for stock splits, and converting currencies where needed.
   * **Integration:** Merging real-time data from Yahoo Finance with historical transaction data, ensuring consistency and accuracy across all datasets.
3. **Visualization Mapping:** After preprocessing, the data is mapped to various visualization components using Plotly and Dash. The mapping process involves:
   * **Attribute Mapping:** Linking data attributes (e.g., stock prices, buy/sell actions) to specific visual elements such as lines, bars, and scatter plots.
   * **Hierarchical Mapping:** Implementing hierarchical color schemes for sectors and industries to provide clear differentiation between different data categories.
   * **Dynamic Updates:** Ensuring that visualizations automatically update in response to new data inputs or changes in user-selected parameters.
4. **Rendering and Interaction:** The final stage involves rendering the visual elements on the user interface and enabling interactivity. Key features include:
   * **Interactive Controls:** Allowing users to customize the visualizations through dropdowns, sliders, and checkboxes, enabling a tailored analysis experience.
   * **Real-Time Updates:** Continuously fetching and rendering new data to provide up-to-date insights, essential for tracking live market conditions.
   * **User Feedback Integration:** Collecting user interactions to refine the tool’s responsiveness and accuracy over time.

A diagram of a model

Description automatically generated

Fig. 6.1.1.: Figure of model design driven visualisation pipeline by Wang et al. (2016).

The visualization pipeline implemented in the "Invest Smart" tool aligns with the visual analytic pipeline model presented in (Fig. 6.1.1.) of Wang et al. (2016). This model emphasizes the importance of a systematic approach to data transformation, visualization, and user interaction, which is crucial for developing a robust financial analysis tool.

### Process Diagram

The process diagram (Fig.6.2.1) for the "Invest Smart" tool provides a detailed visual representation of the system architecture and data flow, illustrating how data moves through the various stages from acquisition to visualization.

A diagram of data processing

Description automatically generated

Fig.6.2.1: Process Diagram for Invest Smart.

**Components of the Process Diagram:**

1. **Data Sources:** The process begins with data sources, including:
   * **Excel Files:** These contain historical investment transactions and company metadata.
   * **Yahoo Finance API:** Provides real-time data on stock prices, historical market data, and other financial metrics.
2. **Data Processing Module:** This module is central to the tool's functionality, involving several key processes:
   * **Excel-Based Calculations:** Specific transformations, such as stock split adjustments and transaction scaling, are handled through Excel formulas directly within the uploaded files.
   * **Python Data Handling:** Utilizing Pandas and NumPy, the system performs additional data manipulations, such as merging datasets, calculating performance metrics, and preparing data for visualization.
3. **Data Integration Layer:** This layer merges pre-processed data from multiple sources, ensuring that the real-time data from Yahoo Finance is correctly aligned with historical data. This step involves verifying data consistency and resolving any discrepancies between datasets.
4. **Visualization Engine:** The core of the system, where data is transformed into visual insights using Plotly and Dash. Key functionalities include:
   * **Rendering Visualizations:** Generating interactive charts such as line charts for stock trends, bar charts for dividends, and scatter plots for performance comparisons.
   * **Interactivity:** Allowing users to interact with the data through features like zooming, panning, and selecting specific data points for detailed analysis.
5. **User Interface (UI):** The UI serves as the interaction hub, displaying the visualizations and providing controls for customizing the display. Key elements include:
   * **Control Panels:** Offering options to filter data by company, sector, or industry, adjust timeframes, and toggle between different visualization types.
   * **Real-Time Feedback:** Providing immediate visual responses to user inputs, enhancing the interactive experience.
6. **Feedback Loop:** The system incorporates a feedback loop that captures user interactions and preferences, using this data to refine the tool’s features and improve usability over time.

# Implementation

The implementation of the "Invest Smart" tool integrates data acquisition, processing, and visualization to create a dynamic and user-friendly investment analysis platform. The system is built using Python, Dash, and Plotly, enabling users to interact with financial data through an intuitive dashboard interface. This section provides a detailed description of the implementation process, including the code logic, data handling, and visualization strategies, with a specific focus on the integration of line charts and stacked bar charts.

### Existing Tools

Before developing the "Invest Smart" tool, various existing visualization tools were evaluated to understand their capabilities, limitations, and potential for integration into the project. This evaluation provided insights into the strengths of different tools and guided the design choices for the "Invest Smart" platform.

#### Google Chart

Google Chart is a popular tool for creating simple, interactive charts that can be embedded in web pages. It offers a range of chart types, including line, bar, and pie charts, which are easily customizable and responsive.

* **Advantages:** Google Chart is free and integrates seamlessly with Google Sheets, making it a good option for users looking for basic data visualization without the need for extensive technical expertise.
* **Limitations:** However, Google Chart has limitations in handling large datasets and lacks advanced analytical capabilities such as those offered by dedicated business intelligence tools. It also has limited support for real-time data updates, which is crucial for financial applications like "Invest Smart."

#### Excel

Microsoft Excel is one of the most widely used tools for data analysis and visualization. It provides a variety of chart types and data manipulation functions, which are highly accessible to users of all skill levels.

* **Advantages:** Excel's strength lies in its familiarity and extensive range of built-in functions for data analysis, such as pivot tables, complex formulas, and conditional formatting. It is particularly useful for handling tabular data and performing quick calculations.
* **Limitations:** Despite its versatility, Excel is not designed for real-time data integration and can struggle with large datasets due to memory and processing limitations. Additionally, its visualizations are generally static and lack the interactivity that modern dashboards provide.

#### Tableau

Tableau is a leading data visualization tool known for its powerful and interactive dashboards. It supports a wide range of data sources and allows users to create complex, visually appealing visualizations with ease.

* **Advantages:** Tableau's drag-and-drop interface and advanced analytical capabilities make it ideal for business intelligence applications. It can handle large datasets and integrates well with various data sources, including real-time data feeds.
* **Limitations:** However, Tableau requires a subscription, which can be costly, and its learning curve can be steep for users unfamiliar with advanced data visualization tools. Additionally, while it offers extensive customization options, Tableau's flexibility can sometimes be overwhelming for simple visualization tasks.

#### Power BI

Power BI, developed by Microsoft, is another powerful business analytics tool that provides interactive visualizations and business intelligence capabilities with a user-friendly interface.

* **Advantages:** Power BI excels in its integration with other Microsoft products, such as Excel and Azure, and offers robust data connectivity options. It supports complex data models, real-time data analysis, and has built-in machine learning features.
* **Limitations:** Like Tableau, Power BI also requires a subscription, and while it offers extensive customization, it can be complex to set up for users who are new to data analytics. Its performance with very large datasets can also be a concern, depending on the underlying data architecture.

### Data Acquisition

Data acquisition for the "Invest Smart" tool involves collecting data from Excel files and Yahoo Finance API. These sources provide both historical and real-time data essential for comprehensive investment analysis.

1. **Excel Data Integration:**
   * Excel files containing historical transaction data, including columns for transaction dates, stock tickers, number of shares, buy/sell actions, and prices per share. The Excel sheets also include metadata such as company details and sector classifications.
   * Preprocessing tasks are performed directly within Excel using built-in formulas to adjust for stock splits, calculate transaction fees, and compute dividends. These adjustments are critical for ensuring that the data accurately reflects historical trading activities.
   * Python’s Pandas library reads the Excel files, performs additional cleaning (e.g., removing duplicates, formatting dates), and integrates the data with API-sourced information.
2. **Yahoo Finance API:**
   * The Yahoo Finance API is accessed using the yfinance Python library to fetch real-time and historical stock data, including daily prices (open, high, low, close), volumes, and dividend payouts.
   * The API allows the tool to dynamically update stock prices and performance metrics, providing users with a current view of their investments. The data integration involves aligning API data with the user’s transaction dates to ensure consistency and accuracy.

### Data Processing

Data processing is a crucial step that prepares the collected data for visualization. It involves cleaning, merging, and feature engineering using Python libraries.

1. **Data Cleaning and Transformation:**
   * **Data Cleaning:** The acquired data is cleaned using Pandas by removing null values, standardizing column formats, and ensuring that all dates are in chronological order. Rows with missing or irrelevant data (e.g., transactions without tickers) are filtered out.
   * **Data Merging:** Historical data from Excel files is merged with real-time data from Yahoo Finance. This process uses stock tickers and dates as primary keys to align datasets, ensuring that the merged data provides a complete and accurate representation of each investment.
   * **Feature Engineering:** Additional metrics such as moving averages (50-day, 100-day, and 200-day), profit/loss percentages, and cumulative returns are computed. These features enhance the analytical depth of the visualizations, allowing users to better understand long-term performance trends.
2. **Data Integration:** A unified dataset is created by integrating cleaned and processed data from Excel and API sources. This dataset serves as the foundation for the visualization engine, providing all necessary inputs for generating interactive charts and graphs.

### Visualization Engine

The visualization engine is built using Dash and Plotly, offering a range of interactive charts that allow users to explore their investment data in detail.

#### Multiple Line Charts:

The primary visualization is a multiple line chart that displays daily stock prices for all investments in the dataset. Users can overlay moving averages and annotate buy/sell actions on the chart, providing a clear visual representation of investment trends over time.

A screen shot of a graph

Description automatically generated

Fig 7.4.1.1: A ScreenShort of Inverst Smart User Interface.

The line chart is implemented using Plotly’s go.Figure and go.Scatter components. Each stock is represented as a separate line, with additional traces added for moving averages and transaction markers (green for buys, red for sells). The chart updates in real-time as users interact with the dashboard controls, such as selecting different sectors or time periods.

#### Stacked Bar Chart Implementation:

* + **Purpose:** The stacked bar chart visualizes the distribution of buy and sell transactions across different sectors, providing a comparative view of investment activities.
  + **Data Preparation:** For the stacked bar chart, the data is grouped by sector and action type (buy or sell) using Pandas groupby and sum functions. The resulting dataset summarizes the total transaction amounts for each sector and action type.
  + **Features:** The stacked bar chart uses colour coding to differentiate between buy and sell actions, with stacked bars representing the total amounts for each sector. Users can hover over bars to view detailed transaction amounts, enhancing their understanding of sector-specific investment behaviours.

#### Colour Mapping Strategy:

The tool employs a hierarchical colour mapping approach using the Paletton tool, where the colour wheel is divided into 10 slices corresponding to the 10 sectors in the investment data. Each slice is 36 degrees wide, and the base colour at the centre of each slice represents a sector. Different shades of the base colour are used to represent industries within the sector.

For example:

* + - **Basic Materials (0° - 36°):** Base colour at 18° is FF5300.
      * Specialty Chemicals: #FF7939
      * Gold: #D24400
    - **Communication Services (36° - 72°):** Base colour at 54° is #FFA100.
      * Entertainment: #FFC25A
      * Internet Content & Information: #FFAB1C
      * Telecom Services: #E49000

This colour mapping enhances the visual differentiation of sectors and industries, making it easier for users to identify trends and compare performance across different market segments [(Paletton, 2024).](https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF)

For profit and loss colour coding, inspiration was drawn from Finviz. The average price of each stock is compared to the current market price to determine the profit or loss percentage, and relevant colours are assigned accordingly.

Stocks showing a profit are highlighted in shades of green, while those showing a loss are highlighted in shades of red, reflecting the degree of profit or loss visually. This strategy helps users quickly assess the financial performance of their investments by providing immediate visual cues based on colour [(Finviz, 2024).](https://finviz.com/)

#### User Interaction and Customization:

The dashboard includes controls for users to customize their view, such as selecting specific categories (company, sector, industry), toggling between different background colours, and adjusting data plot types (e.g., historical prices, buy/sell actions).

The interactivity is implemented using Dash’s callback functions, which update the visualizations in response to user inputs, providing a responsive and personalized user experience.

### User Interface and Feedback Loop

The user interface (UI) is designed to be intuitive and engaging, with multiple views and interactive controls that allow users to customize their data exploration.

1. **Dashboard Layout:**

The UI consists of several views tailored to different aspects of investment analysis, such as Buy/Sell actions, Dividends, and Single or Multiple stock views. Each view is designed to provide specific insights, making it easy for users to navigate through their data.

The main visualization area is complemented by side panels that contain controls for filtering data, selecting categories, and adjusting display settings, providing a highly interactive user experience.

1. **User Options on the GUI:**

* **Category Selection:** Users can filter data by Company, Sector, or Industry, allowing them to focus on specific areas of their portfolio.
* **Moving Averages:** A dropdown menu enables users to select different moving averages (e.g., 50-day, 100-day) to overlay on the line charts, providing insights into long-term performance trends.
* **Data to Plot:** Checkboxes allow users to toggle different data plots on the line charts, such as Historical Price, Buy Actions, Sell Actions, Closed Positions, and New Positions. This flexibility helps users tailor the visualizations to their specific analysis needs.
* **Background Color:** Users can switch between a white or black background to enhance visibility and contrast based on their preferences.
* **Color Scheme:** Options to change the color scheme by Sector, Industry, or Profit/Loss Percentage help users visually distinguish between different data sets.
* **Decision Orientation:** A toggle option lets users view decisions as past-oriented (based on historical performance) or future-oriented (considering current market trends). This feature highlights buy/sell actions in color-coded markers to indicate profitable or unprofitable transactions.
* **Scaling by Transaction Amount:** Users can scale the size of buy and sell markers based on the transaction amount, providing a more accurate representation of their investment activities.

1. **Feedback Loop:**

The tool incorporates a feedback mechanism that captures user interactions, such as frequently selected filters and preferred settings. This data is used to refine the tool's features, ensuring it evolves in response to user needs and preferences over time.

# Evaluation

This section provides an in-depth evaluation of the tool through a series of case studies aimed at assessing its effectiveness in delivering tailored financial insights. By applying the tool to a range of investment scenarios, the evaluation demonstrates that the tool not only fulfils its intended design goals but also significantly enhances user decision-making through its dynamic and interactive visualizations. This thorough testing highlights the tool's ability to revolutionize conventional investment analysis by offering deeper, personalized insights that align with individual investment strategies.

### Results

This section presents the results of several case studies conducted to assess the effectiveness of the "Invest Smart" tool in evaluating stock investment performance for anonymous investors. Throughout the implementation phase, each visualization component and its interactive features were meticulously tested to ensure functionality, accuracy, and reliability in the final output. These case studies showcase how the tool aids investors by providing advanced visualization techniques and interactive analytical capabilities, enabling comprehensive comparisons between various stocks and evaluating the quality of investment decisions.

#### **Case Study A: "A Comprehensive Performance Analysis of Equity Investments”**

The objective of this case study is to provide a comprehensive analysis of equity investments from 2020 to 2024, aimed at equipping retail investors with actionable insights to enhance their investment strategies. The study evaluates key performance metrics including realized and unrealized capital gains/losses, dividends, and total returns across a diversified portfolio of companies. By examining these metrics, the study seeks to identify the best and worst-performing companies, assess the impact of dividends on overall returns, and differentiate between realized and unrealized gains for effective tax planning and strategic decision-making. It also explores the influence of market conditions, sector performance, and outliers on investment outcomes. Additionally, the study aims to highlight the importance of diversification, income stability, and risk management, providing retail investors with a detailed framework to optimize their portfolios. Ultimately, the goal is to empower investors to make data-driven decisions that align with their financial goals and risk tolerance.

A graph of a graph

Description automatically generated with medium confidence

#### Sub-Study 1: Performance Analysis of Investments (2020-2024)

**Objective:** Identify the best and worst performing companies from 2020 to 2024.  
**Focus:** Realized and unrealized capital gains/losses and total return.

**Data and Visualization Analysis:**

The stacked bar chart presents a detailed comparison of realized capital gains/losses, unrealized capital gains/losses, and dividends for each company, illustrating their combined effect on total returns. The data was sorted to highlight companies with the highest and lowest total returns, offering a clear visual representation of each company's performance metrics.

A graph of a graph

Description automatically generated with medium confidence

Fig. 8.1.1.1: Stacked bar-chart with capital gains / losses, dividends and total return.

1. **Top Performers:**
   * **NEE (£14,197.23):** NEE stands out as the top performer (Fig. 8.1.1.1), driven largely by significant unrealized gains. This indicates that the company's stock has appreciated considerably since purchase, reflecting strong market sentiment and potential future growth. Investors holding NEE are well-positioned for continued appreciation if market conditions remain favourable.
   * **AVGO (£9,670.77) and COST (£8,394.20):** Both companies demonstrated (Fig. 8.1.1.1) robust total returns, with AVGO showing a balanced contribution from realized gains, unrealized gains, and substantial dividends. COST also performed well, driven mainly by unrealized gains. These companies exemplify strong financial health and effective business strategies, making them attractive for long-term investment.
   * **AAPL (£6,828.87) and SAP (£5,463.89):** These companies achieved notable returns (Fig. 8.1.1.1), primarily through unrealized gains. This suggests strong market confidence in their future potential, underpinned by innovative business models and market leadership positions.
2. **Worst Performers:**
   * **HASH (£-3,114.32) and SEAT (£-2,778.41):** These companies (Fig. 8.1.1.1) recorded the most negative total returns, predominantly due to substantial unrealized losses. This points to either a significant decline in their market valuations or a potential overestimation of their initial value at purchase. Such losses signal caution and the need for investors to reevaluate their positions.
   * **BUD (£-2,458.47) and BYND (£-2,175.84):** These companies (Fig. 8.1.1.1) also faced significant unrealized losses, reflecting challenges in maintaining market value or operational issues that have negatively impacted investor confidence. This highlights the inherent risks of investing in high-volatility or speculative stocks.
3. **Realized vs. Unrealized Gains/Losses:**
   * **Realized Gains/Losses (Blue):** These gains represent profits that have been locked in through the sale of shares, directly contributing to an investor's cash flow and affecting taxable income. For instance, AAPL's substantial realized gains (Fig. 8.1.1.1) reflect successful sell strategies at opportune moments.
   * **Unrealized Gains/Losses (Red):** These are potential profits or losses based on current market valuations, which have not yet been realized through selling. High unrealized gains suggest continued growth potential but also expose the portfolio to future market fluctuations.

**Implications for Retail Investors:**

* **Investment Decisions:** This analysis helps investors identify which companies are performing well and which are lagging, providing a clear basis for decisions on whether to hold, sell, or buy additional shares. High performers may represent strong long-term holds, while poor performers may necessitate re-evaluation.
* **Risk Assessment:** The differentiation between realized and unrealized gains allows investors to assess the stability and risk exposure of their portfolios. High unrealized gains suggest holding potential, while high unrealized losses indicate areas of concern.
* **Diversification Strategy:** The significant variance in performance underscores the importance of diversification across sectors and companies, which can help mitigate the risks associated with individual stock volatility and enhance overall portfolio resilience.

#### Sub-Study 2: Impact of Dividends on Total Return

**Objective:** Assess how dividends contribute to overall returns.  
**Focus:** Importance of dividends in generating income for retail investors.

**Data and Visualization Analysis:**

* Dividends, represented in green in the visualization (Fig. 8.1.1.1), are shown to significantly boost total returns, especially for companies like NEE and AVGO. These dividends provide a reliable income stream that complements capital gains, enhancing overall portfolio performance.
* Companies such as ABBV and MCD, which consistently paid dividends, managed to achieve positive total returns even when capital gains were less significant. This stability is particularly valuable during periods of market volatility, as dividends can help offset declines in share price.
* Dividends act as a buffer against market fluctuations, offering investors a steady source of income irrespective of stock price movements, which is crucial for risk-averse or income-focused investors.

**Portfolio Implications and Optimization Strategies:**

* **Income Generation:** Dividends provide a dependable income stream, especially important for retirees or investors seeking regular cash flow. This reduces reliance on selling shares for income, preserving the capital base.
* **Portfolio Stability:** Dividend-paying stocks can stabilize a portfolio, as returns from dividends are generally less volatile than stock prices. This strategy aligns with conservative investment approaches, where maintaining capital stability is as important as seeking growth.
* **Compounding Benefits:** Reinvesting dividends into additional shares can amplify returns over time through the power of compounding, significantly enhancing long-term growth prospects. This strategy is particularly effective in building wealth over the long run.

**Implications for Retail Investors:**

* **Income Strategy:** For investors prioritizing income, dividends are a critical component, providing regular payouts that are less susceptible to market conditions compared to capital gains.
* **Risk Management:** Dividend income adds a layer of protection against market downturns, making portfolios less vulnerable to negative market sentiment.
* **Growth Potential:** Reinvested dividends can lead to substantial growth over time, contributing to a robust investment strategy that combines income and capital appreciation.

Overall, this case study provides a detailed framework for retail investors to evaluate their equity investments comprehensively. By integrating insights on performance metrics, market conditions, sector dynamics, and risk-return profiles, investors can better navigate the complexities of the stock market. The findings empower investors to make decisions that not only reflect past performance but are also forward-looking, considering potential market shifts and the evolving landscape of investment opportunities.

#### Case Study B: Identification of Anomalous Data in Transactions

This case study focuses on the identification of anomalies within the investment transactions dataset. The detection of these anomalies is facilitated through the visual analysis of transaction data and corresponding stock price trends, underscoring the tool's capability to detect and highlight outliers, which are critical for investors assessing potential risks or unusual trading activities.

A screen shot of a graph

Description automatically generated

Fig. 8.1.2.1: A Screenshot ‘MMM’ trend with buy/sell markers

Abnormalities in MMM's transaction data were identified using a historical price line chart (Fig. 8.1.2.1), which provides a detailed visualization of individual stock price movements in conjunction with buy and sell transaction markers. This approach allows for a precise identification of data points that deviate significantly from expected trends. As depicted in the graph, several transactions, specifically those flagged as 'bad buys' and 'bad sells,' are visibly misaligned with the prevailing stock price trend, indicating potential errors in transaction timing or pricing discrepancies.

The dataset reveals notable outliers, including a buy transaction on 17/02/2021 and a sell transaction on 11/04/2022. For instance, the buy on 17/02/2021 was executed at a price of 176.59, which is significantly higher than subsequent prices, suggesting an ill-timed purchase or an anomaly in the data. Similarly, the sell transaction on 23/08/2023, executed at a price of 98.67 per share, diverges considerably from the anticipated stock price trend, as evidenced by its classification as a 'bad sell' in the chart. This discrepancy could be indicative of unusual price behaviour or data inaccuracies that warrant further scrutiny.

Further examination of 3M’s corporate events and market conditions did not reveal standard justifications such as stock splits, dividends, or mergers that might account for these anomalies. Therefore, these irregular transactions likely reflect issues related to data integrity, atypical trading activity, or other external factors that are not immediately apparent from routine market analysis.

The case study illustrates the importance of employing visual tools for the identification of anomalous data points within investment transaction datasets. By isolating and analyzing these outliers, investors and analysts can conduct deeper investigations into the root causes of discrepancies, ensuring more reliable data interpretation and decision-making processes. This approach is integral to maintaining data accuracy and optimizing investment strategies in environments susceptible to data irregularities and market anomalies.

This study reinforces the value of integrating robust visual analysis techniques in financial data evaluation to proactively identify and address potential risks associated with irregular trading patterns or data anomalies.

1. Case Study C: Comparison of Good vs Bad Decisions

This case study conducts a strategic evaluation of buying and selling decisions within the stock market, distinguishing between successful and less optimal trades using data from the Investment Transaction Dataset. By leveraging interactive data visualizations, this analysis examines each transaction to determine whether the actions taken by investors align with favourable market trends, as indicated by moving averages. Buys executed below the moving average and sells made above it are considered potentially advantageous, often leading to gains, while trades that do not align with these benchmarks are examined as potentially less favourable decisions. This comparative analysis highlights the importance of moving averages as key benchmarks for effective trading practices and emphasizes their role in assessing the strategic timing of market entry and exit points.

In addition to this past-oriented approach, a future-oriented perspective is also employed. From this viewpoint, buy actions are evaluated based on the relationship between the initial purchase price and the current stock price: the lower the purchase price compared to the current market price, the more favourable the decision is considered. Conversely, for sell actions, the evaluation hinges on whether the sale price was higher relative to the current price, making the decision more advantageous. This approach allows investors to not only assess past trades against moving averages but also to gauge the ongoing impact of their decisions in the context of current market conditions.

For past-oriented analysis, we examine buy and sell actions made before the present date to determine their alignment with favorable trading principles based on the 200-day moving average.

Let’s consider AVGO as an example on multiple occasions, such as the purchases made on June 5, 2020, and August 30, 2020, buy actions were executed at prices near to 200-day moving average. This alignment suggests strategic entry points, which are generally considered advantageous since they allow investors to acquire shares at prices lower than the broader market trend.

A screenshot of a computer

Description automatically generated

Fig8.1.3.1: A Screenshot of ‘AVGO’ trend chart with transaction points (Past Oriented).

A notable sell action on April 4, 2024, took place above the 200-day moving average. The decision to sell at this elevated price point was well-timed, as it secured gains by capitalizing on the stock's upward trajectory. The price at the time of sale was significantly higher than the moving average, reinforcing the effectiveness of using the 200-day moving average as a sell trigger.

While several buy decisions were in line with the strategy of purchasing below the 200-day moving average, some actions, such as the purchase on April 8, 2024, were closer to the moving average, indicating a neutral position. This suggests that while these buys may not have maximized the discount potential, they still held within a reasonable range for profitability.

For a future-oriented perspective, evaluating the current prices relative to past buy actions offers insight into potential future gains:

A screen shot of a graph

Description automatically generated

Fig8.1.3.2: A Screenshot of ‘AVGO’ trend chart with transaction points (Future Oriented).

As of the latest data, AVGO's price has risen substantially beyond the purchase prices recorded in the dataset. This substantial growth highlights the effectiveness of strategic buys made below the 200-day moving average. For example, an initial purchase price averaging around £64 is dwarfed by the current stock price, which has soared to £152.79 as of September 3 , 2024. This represents a significant gain, underscoring the potential for substantial profit when aligning buy decisions with favorable market conditions.

While future gains are promising, it is critical to recognize the potential for trend reversals. If current prices decline sharply below the moving average, what were once profitable positions might turn disadvantageous. This highlights the importance of continuous monitoring and potential exit strategies to lock in gains or minimize losses if market conditions shift unfavorably.

If we consider ‘VICI’ as example Bad Buys (Highlighted in Blue Box) (Fig8.1.3.4) In the past-oriented evaluation, the blue box in the chart shows a cluster of buy actions that were made when the stock price was above the 200-day moving average. These buys, marked in red, are considered less favorable as they occurred at higher price points, reducing the potential for profit.



Fig8.1.3.3: A Screenshot of ‘VICI’ trend chart with transaction points (Past Oriented).

Good Buys (Marked in Green) (Fig8.1.3.3) occurred when the stock price was below the moving average, represented by green markers. These decisions took advantage of lower prices relative to the average, aligning with favorable market conditions and increasing the likelihood of gains.

Change from Red to Green (Fig8.1.3.4) the future-oriented approach shows a significant shift where many initial 'bad buys' marked in red (above the moving average) have now turned green. This change reflects that although these buys were initially above the moving average, they have become favourable decisions in hindsight, as the current price level has risen or stabilized well above the purchase prices. This shift illustrates that timing relative to the 200-day moving average is important, but market dynamics can change, turning previously unfavorable decisions into profitable ones.

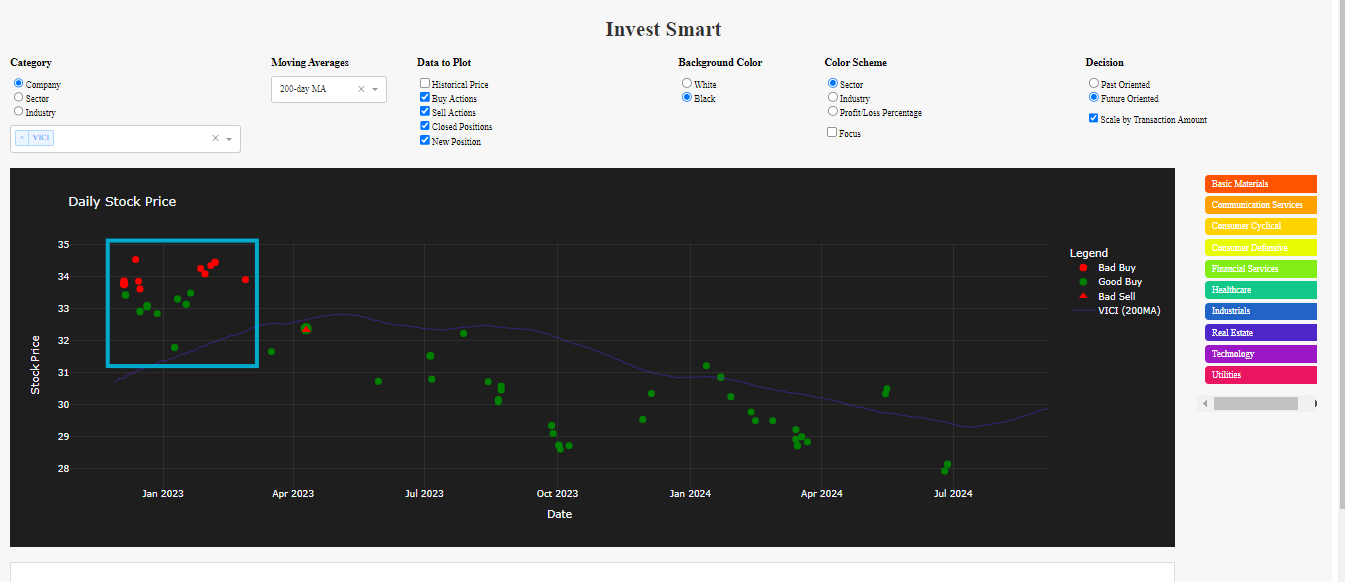


Fig8.1.3.4: A Screenshot of ‘VICI’ trend chart with transaction points (Future Oriented).

In this future-oriented scenario, good sells are those marked in green, where the sell price was higher than subsequent prices, confirming the effectiveness of those exit points. This reinforces the importance of not just the moving average but also the ongoing market trend in evaluating the success of sell actions.

While the past-oriented analysis highlights the strategic use of moving averages as a benchmark, the future-oriented approach emphasizes the importance of considering current market prices. Trades that initially seemed suboptimal can turn profitable when current prices align favorably, showcasing the benefit of a longer-term view in evaluating investments.

The notable shift from red to green markers (Fig8.1.3.4 and Fig8.1.3.3) indicates an improvement in the perceived quality of these buy decisions over time. This highlights the value of patience and market conditions evolving in Favor of the investor's original strategy, despite initial misalignment with the moving average.

# Conclusion and Future Work

The development of the "Invest Smart" visualization tool has demonstrated significant potential to enhance retail investors' decision-making through dynamic, user-centric financial visualizations. The tool effectively addresses key challenges in financial data interpretation by providing an interactive platform that caters to individual investment needs. Its success lies in integrating real-time data analysis, customizable visualizations, and user feedback mechanisms, which collectively improve the accessibility and utility of financial information.

### Conclusion

This project successfully developed a comprehensive tool that meets its primary objectives: enabling investors to visualize their stock performance, analyse market trends, and make data-driven decisions. Through rigorous testing and evaluation, the tool proved effective in delivering personalized insights, simplifying complex data into actionable visual formats, and supporting various investment strategies. The tool's ability to transform traditional static data into engaging and interactive visualizations signifies a substantial advancement in the realm of investment analysis.

Throughout its development, the tool adhered to best practices in data handling, coding standards, and user-centred design, ensuring that the platform remains reliable, secure, and intuitive. The implementation of hierarchical colour mapping, detailed financial metrics, and responsive chart interactions showcases the depth of the tool’s design considerations. Furthermore, by incorporating feedback from stakeholders and potential users, the tool continuously evolved to better serve its target audience.

### Future Work

While the "Invest Smart" tool has met its initial goals, several avenues for future development have been identified to further enhance its capabilities:

* **Expansion of Data Sources**: To provide a more comprehensive analysis, future iterations could integrate additional financial data sources, such as macroeconomic indicators, social sentiment analysis, or data from alternative asset classes like cryptocurrencies and real estate. This expansion would allow users to conduct more holistic investment analyses.
* **Machine Learning Integration**: Incorporating predictive analytics through machine learning models could offer users insights into potential future market movements. This feature would not only enhance decision-making but also provide a competitive edge by forecasting stock performance based on historical trends and market conditions.
* **Enhanced Personalization**: Future versions could further refine user personalization by allowing deeper customization of visualizations, such as creating user-defined metrics or setting personalized investment goals within the tool. This would increase the tool's relevance to individual users and support a broader range of investment strategies.
* **Mobile Application Development**: Developing a mobile application would extend the tool’s accessibility, allowing users to monitor and analyse their investments on the go. A mobile version could include tailored features for mobile interfaces, such as simplified charts and alerts for significant market events.
* **Enhanced Security Features**: As the tool evolves, strengthening its security protocols will be paramount, especially as more sensitive data integrations are considered. Implementing advanced encryption, two-factor authentication, and regular security audits would help safeguard user data against evolving cyber threats.
* **Community and Social Features**: Adding community-driven features, such as forums or shared investment strategies, could foster a collaborative environment where users can learn from each other. This would also enable the sharing of insights and strategies, potentially enriching the user experience.
* **Performance Optimization**: Continuous optimization for speed and performance, especially as data volume increases, will be crucial. Future work could involve exploring more efficient data handling and visualization techniques to maintain the tool's responsiveness and scalability.

In summary, the "Invest Smart" tool has laid a strong foundation for personalized financial visualization. By continuing to evolve and expand its features, the tool can further solidify its role as an indispensable resource for retail investors aiming to navigate the complexities of the financial markets.

**References**

1. Alharbi, M., Laramee, R. S., & Chen, M. (2021). Visualization of Time-Oriented Data: A Systematic Review. *IEEE Transactions on Visualization and Computer Graphics*, 27(3), pp. 1450-1468.
2. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. *Agile Alliance*.
3. Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. Sebastopol: O'Reilly Media.
4. Heer, J., & Shneiderman, B. (2012). Interactive Dynamics for Visual Analysis. *Communications of the ACM*, 55(4), pp. 45-54.
5. Heer, J., Bostock, M., & Ogievetsky, V. (2010). A Tour through the Visualization Zoo. *Communications of the ACM*, 53(6), pp. 59-67.
6. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), pp. 90-95.
7. Keim, D., Kohlhammer, J., Ellis, G., & Mansmann, F. (2008). *Mastering The Information Age – Solving Problems with Visual Analytics*. Eurographics Association. Available at: <https://diglib.eg.org/items/cd5e470c-3d55-413c-ba9c-2ae9d9464fea>.
8. Lam, H., Bertini, E., Isenberg, P., Plaisant, C., Carpendale, S., & Fekete, J.-D. (2012). Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9), pp. 1520-1536.
9. Liu, X., Alharbi, M., Chen, J., Diehl, A., Firat, E. E., Rees, D., Wang, Q., & Laramee, R. S. (2022). Visualization Resources: A Survey. *Information Visualization*, 22(1), pp. 3-30. Available at: <https://doi.org/10.1177/14738716221126992>.
10. McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, pp. 51-56.
11. McNabb, L., & Laramee, R. S. (2017). Survey of Surveys (SoS): Mapping the Landscape of Survey Papers in Information Visualization. *Computer Graphics Forum*, 36(3), pp. 589-617. Available at: <http://dx.doi.org/doi:10.1111/cgf.13212>.
12. Munzner, T. (2014). *Visualization Analysis and Design*. Boca Raton: CRC Press.
13. Paletton. (2024). *Color Scheme Designer*.

Available at: <https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>.

1. Plotly Dash Documentation. *Dash Enterprise*. Available at: <https://dash.plotly.com/>.
2. Rees, D., & Laramee, R. S. (2019). A Survey of Information Visualization Books. *Computer Graphics Forum*, 38(1), pp. 610-646. Available at: <https://doi.org/10.1111/cgf.13595>.
3. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017). Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), pp. 341-350.
4. Scheidegger, C. E., Vo, H. T., Koop, D., Freire, J., & Silva, C. T. (2008). Querying and Re-Using Workflows with VisTrails. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1251-1254.
5. Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org.
6. Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pp. 336-343.
7. Wang, X.-M. et al. (2016). A Survey of Visual Analytic Pipelines. *Journal of Computer Science and Technology*, 31(4), pp. 787–804. Available at: <https://doi.org/10.1007/s11390-016-1663-1>.
8. Ware, C. (2019). *Information Visualization: Perception for Design*. 4th ed. San Francisco: Morgan Kaufmann.
9. Yahoo Finance API. Available at: <https://www.yahoofinanceapi.com/>.