„ALEXANDRU IOAN CUZA “UNIVERSITY OF IAȘI

FACULTY OF COMPUTER SCIENCE

BACHELOR THESIS

Proposed by

Sandu Tozlovanu

Session: July 2023

Scientific coordinator

Arusoaie Andrei

„ALEXANDRU IOAN CUZA “UNIVERSITY OF IAȘI

FACULTY OF COMPUTER SCIENCE

PortfoPal - a web tool for investment portfolio management and stock market analysis

Proposed by

Sandu Tozlovanu

Session: July 2023

Scientific coordinator

Arusoaie Andrei

Contents

Table of Contents

Introduction

Nowadays, such words as “stock market,” “investment,” “finances” are very popular among people all around the world. Everyone with internet access has heard or even used these words at least once in their lives. The reason behind this is amazingly simple - people are constantly looking for more and more ways to earn money. The stock market is seen as a way to increase a person’s income without having to work a full-time job, and that is also a reason why it is so popular these days. But not many people know that the history of the stock market is lengthy and fascinating, spanning many centuries. To summarise it in a few words, it started out as a tool for businesses to raise money and for people to invest their money in companies they trusted. People began buying stocks more frequently as they became aware of the possibility for wealth accumulation and financial progress. The stock market promoted economic growth, innovation, and wealth accumulation over time, eventually becoming a crucial part of the world economy.

But why do people decide to trust their money and time investing in the stock market? One of the many reasons is that it offers numerous advantages. We will discuss some of them based on the book „The Little Book of Common Sense Investing” by John C. Bogle. First of all, according to the book, stocks have provided significant long-term growth, outperforming other asset classes over extended periods. This implies a reduced risk level, thus a higher trust in investing significant amounts of money. Moreover, John C. Bogle underscores that stock investing allows individuals to become partial owners of profitable companies and participate in their success through dividends and capital appreciation. Therefore, a personal connection is formed, and the trust feeling is enhanced. Thirdly, stocks are highly liquid investments, offering the ability to buy and sell shares easily. In this way, investors have enough flexibility to adjust their holdings based on changing market conditions or personal financial goals. In addition, by investing in stocks, individuals gain exposure to the overall market returns, which have historically been positive, allowing investors to benefit from the growth and progress of the economy on the long term.

This being said, it is not a surprise that in the past decade, there has been a significant increase in the total market capitalisation globally. More and more people from all over the world have started to acknowledge the benefits of stock market investing. According to data depicted in Figure 1 and Figure 2, the market capitalisation of both the United States and Romania has grown by over 60%. This substantial rise highlights the positive trend and growth potential in the respective stock markets. The increased market capitalisation indicates the growing investor confidence, economic

development, and overall stability of the financial markets in both countries. These figures also indicate the attractiveness and potential for investment opportunities in the US and Romanian markets.

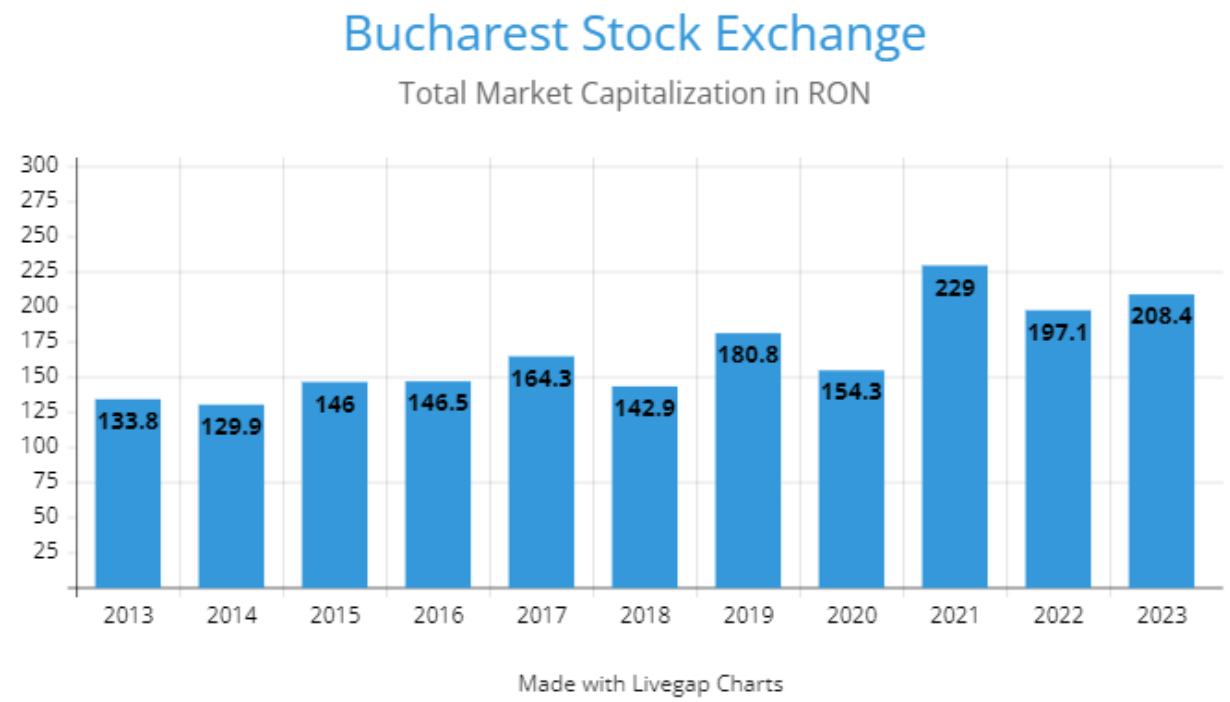


Figure 1 - „Bucharest Stock Exchange Evolution" processed by the author using data from https://bvb.ro/TradingAndStatistics/Statistics/MarketEvolution

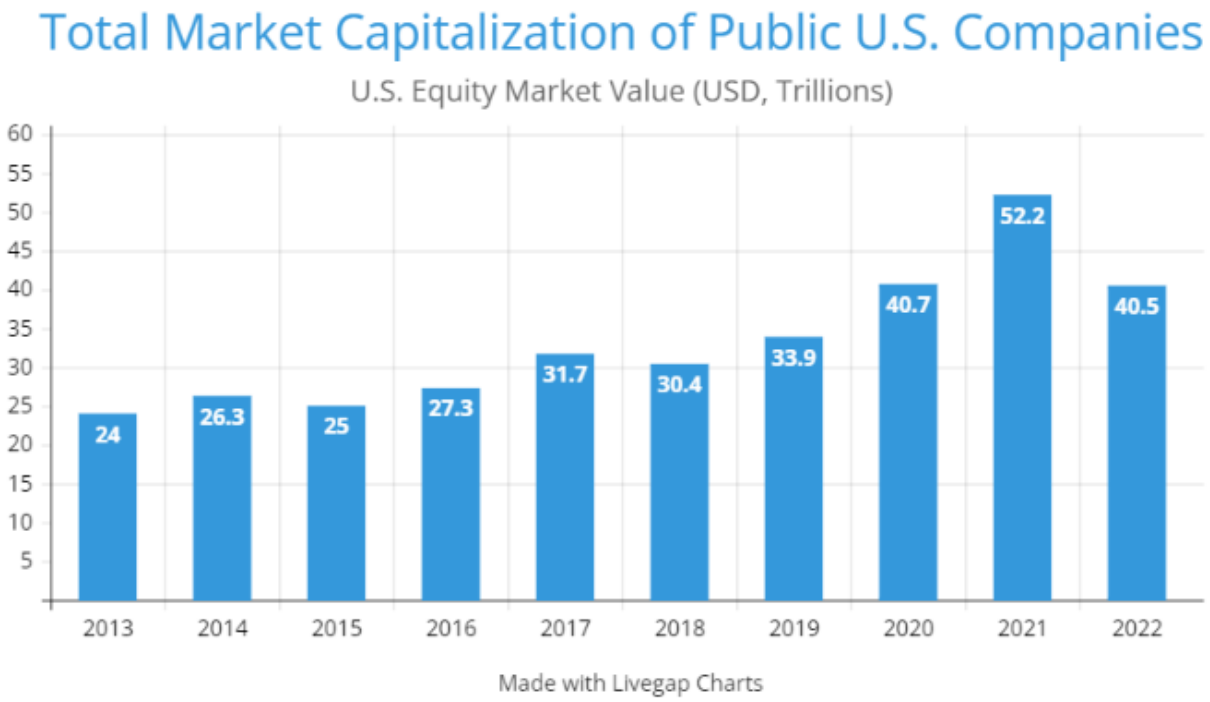


Figure 2 - "Total Market Capitalization of Public U.S. Companies" processed by the author using data from https://siblisresearch.com/data/us-stock-market-value/.

Nowadays, the stock market has become even more prominent due to technological advancements and online platforms that have made investing more accessible to people from all backgrounds. Just an internet connection and some capital is enough for anyone to enter the world of investing. This accessibility has democratised investing, opening up avenues for wealth creation and financial independence and allowing individuals to take charge of their financial future.

Just like other young people, I also saw the incredible opportunities that the stock market offers and decided to create a web tool that would help investors to manage their portfolio easier and to help them in the stock market analysis. Several factors drove my motivation to start doing this project, including:

1- Personal Interest, because investing has captivated my attention since I turned eighteen. I have actively built my investment portfolio, gaining first-hand experience in the market. This passion has led me to continuously research and analyse stocks, always searching for valuable investment opportunities. Through my own journey, I understand the challenges and complexities that investors face. I believe that developing a web tool can simplify and enhance the investment process for myself and others.

2- Long-Term Financial Independence, since my ultimate goal is to achieve financial independence by the time I reach the age of 50 years. In my opinion, long-term investing in stocks is a key pathway to achieving this dream. The concept of compound interest, where returns are reinvested and grow exponentially over time, is a strong reason to create a lifetime habit that will provide both joy and financial stability. Through this project that facilitates portfolio management and stock analysis, I aim to provide myself and others with the necessary tools to help with in the journey towards financial independence.

3- Lack of Proper Tools, which I have often encountered, just like other investors. While there are various resources available, there is a need for a user-friendly web tool that combines powerful screening capabilities with an algorithmic approach. Such a tool would help investors make informed decisions and optimise their investment strategies. By addressing this gap, I strive to provide investors with a reliable and efficient tool that enhances their ability to identify promising stocks and make smarter investment choices.

I wanted to contribute to the way investors manage their portfolios and analyse the stock market, so I developed an advanced web application called "PortfoPal." This application uses state-of-the-art technologies like the React framework for the user-friendly frontend and the Flask framework for the backend. What makes PortfoPal special is its modular architecture, consisting of three microservices designed to perform specific tasks. Thanks to the Data Collector Module included

in one of these microservices, it is ensured that the database remains constantly updated with the latest market data.

Another significant contribution is the development of algorithms that enable the creation of investment portfolios based on proven investing strategies. What sets my algorithms apart is their user-friendly nature. Unlike other tools that primarily rely on pre-packaged investment options like Exchange-Traded Funds[1]. PortfoPal allows users to understand and visualise their created portfolios in a meaningful way. This hands-on approach empowers users to take control of their investments and gain a deeper understanding of the underlying investment strategies.

Additionally, PortfoPal offers a unique feature through its individual stocks profile page. This page presents users with easy-to-understand fundamental stock information and statistics. Going beyond typical stock profiling tools, PortfoPal incorporates a Price Prediction feature for each stock. This prediction uses the Prophet statistical forecasting algorithm, providing users with insights into potential future price movements. This valuable information assists users in making informed investment decisions.

Furthermore, PortfoPal delivers personalised stock recommendations and news to each user. By analysing the stocks already held by a user, the platform customises recommendations to their specific preferences and interests. This personalised approach ensures that users receive relevant and engaging stock suggestions. Moreover, PortfoPal chooses and organises news articles and information specifically related to the user's current stock holdings, keeping them well-informed about the latest developments that may impact their investments.

Now that we have discussed a little about the stock market history, my motivation to start this project and the contributions I provided, we can go through the structure of the following project and what will be presented in each one of the four chapters. Chapter 1, where I delve into the addressed problem of my thesis and present the solution that was developed. Shortly, chapter 1 is about the comprehensive analysis of the challenges faced by investors in managing their portfolios and analysing the stock market, highlighting the importance of having a user-friendly and efficient tool for portfolio management. Additionally, existing tools in the market that attempt to solve similar problems are discussed, so that we can examine their features, limitations, and user experiences. By comparing these existing tools, the unique value proposition of this project's solution is emphasised. Furthermore, we can see an in-depth overview of the features of this web application, including

[1] (ETF) Type of investment fund and exchange-traded product, i.e., they are traded on stock exchanges.

authentication, stock screening functionality, portfolio creation using investing strategies, and general portfolio management and monitoring capabilities.

Chapter 2 focuses on the application architecture, where a detailed explanation of its different components is provided. The general aspects of the architecture are discussed, highlighting the design choices made to ensure the efficiency and scalability of the application. I delve into the back-end and front-end implementation, discussing the technologies and frameworks used in each layer. Additionally, the networking considerations are addressed, such as data communication protocols and APIs used to interact with external services. It also presents the data management process, including data collection, storage, and retrieval mechanisms. Furthermore, the security concerns are highlighted, presenting the measures implemented to protect user data and ensure the integrity of the application.

In chapter 3 we explore the various use cases of the application. One significant use case is positioning the application as an investment education platform. It argues how PortfoPal promotes fundamental analysis, allowing users to gain insights into the financial performance and valuation of different stocks. Moreover, it emphasises the way the application provides users with the opportunity to explore diverse investment strategies, facilitating their understanding of various approaches to portfolio management. Additionally, it presents the application as an investment portfolio experimenting tool, enabling users to simulate and evaluate the performance of different investment portfolios based on their selected strategies.

Chapter 1: Description of the application

In this chapter, we will first see what the addressed problem is, proposing potential solutions for the it while analysing the existing tools on the market. Secondly, we will go through some of the application functionalities, such as the authentication process, stock screening, stock profiling and others. The goal is to understand the context in which we are working, such as the problem and the existing solutions and to explain our own solution with all its functionalities.

1.1. Addressed problem

Investing in the stock market can be an exciting and potentially rewarding experience for anyone who dares to try. However, there are several challenges that individuals face when they want to start this journey. These challenges can often discourage or confuse aspiring investors, hindering their progress and success in the market.

The lack of educational resources for beginner investors is one common challenge that almost every person encounters when they want to begin investing. Many people have limited knowledge about investing concepts, how the stock market works, or even how to get started. Therefore, understanding investment strategies, the role of index funds, and other fundamental concepts can be overwhelming for newcomers. Without proper education and guidance, individuals may struggle to make informed decisions and navigate the complexities of the stock market.

According to "The Investor's Manifesto: Preparing for Prosperity, Armageddon, and Everything in Between" by William J. Bernstein, another significant problem is the overwhelming amount of information that investors need to analyse. When considering specific stocks, investors are bombarded with data and characteristics that need careful evaluation. Comparing stocks and identifying the most suitable investment opportunities can be a time-consuming and daunting task. The absence of user-friendly tools and data visualisation capabilities makes this process even more challenging, often resulting in confusion and inefficiency.

Managing and tracking investment portfolios is another hurdle that investors must overcome. With multiple holdings and numerous transactions, keeping track of portfolio performance, asset allocation, and transaction history can be overwhelming. Without a centralised platform or system, investors may struggle to stay organised and may miss important opportunities. A streamlined portfolio management solution is crucial for investors to monitor their performance, track transactions, and make informed adjustments seamlessly.

Combining the difficulties of data search and analysis, investors also face challenges when trying to apply investing strategies. Implementing an investing strategy often requires gathering data from multiple sources, including stock data and specific data relevant to the chosen strategy. Analysing and assigning scores to various factors can be a time-consuming and complex task for investors, but with the help of a smart tool this process can be streamlined and automated.

By leveraging advanced algorithms and data processing capabilities, a smart tool could efficiently collect and analyse the necessary data to apply an investing strategy. It could swiftly evaluate multiple stocks based on predetermined criteria, saving investors valuable time and effort. Through automation, investors could access actionable insights and scores that align with their investing strategy, empowering them to make informed decisions without the burden of manual data analysis.

This automation would not only enhance the efficiency of applying investing strategies, but also minimise the potential for human bias or error. A smart tool would be able to objectively process data and provide consistent results, ensuring a more reliable and robust approach to strategy implementation. Investors can have confidence in the tool's ability to process vast amounts of information and generate accurate scores, allowing them to focus on other aspects of their investment decision-making process.

1.2. Solution and similar existing tools

Luckily for the investors, the problems mentioned above are solvable, and tools that are designed to deal with some of these problems already exist on the market nowadays. The goal of this chapter is to analyse the most popular tools and the solutions that are implemented for the problems enounced above.

As we said, there are various solutions available in the market for investors. Among them, two dominant categories emerge: robo-portfolio advisors and comprehensive online platforms. These categories differ in their primary focus and the tools they provide.

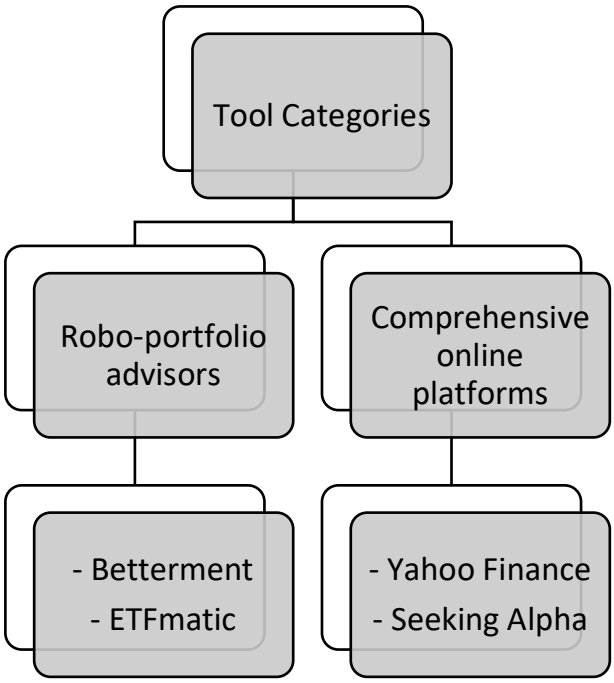


Figure 3 - Main investment tool categories existing on the market, source: personal redaction.

Robo-portfolio advisors primarily concentrate on managing portfolios that have been created by algorithms and experts. They offer a range of portfolio management tools and portfolio trackers, enabling users to monitor the performance of their investments. However, these advisors often limit users to a selection of ETFs or predefined strategies, excluding individual stock holdings. Consequently, they lack stock screening and profiling features that allow users to analyse and assess specific stocks.

On the other hand, comprehensive online platforms provide a wide range of financial tools for analysing stocks. These platforms offer various charts, visualisations, stock screeners, and stock profiles, allowing users to delve into the details of individual stocks. Additionally, they typically provide up-to-date and relevant financial news to keep investors informed. However, comprehensive online platforms may not have dedicated tools for creating portfolios using algorithms. Their focus is primarily on providing investors with extensive financial analysis and tools for stock research and evaluation.

Betterment [2]is a robo-advisor platform available exclusively in the United States. It leverages algorithms and expert guidance to create and manage investment portfolios based on individual preferences and risk profiles. As a portfolio creator, Betterment offers various strategies built by professionals for investors to choose from. It allows users to track portfolio performance over time

[2] https://www.betterment.com/

and offers options for manual or automatic portfolio adjustments. However, Betterment does not support individual stock holdings, which means it lacks stock screening and profiling features. It can also be used only in the United States, which makes it limited to a narrow audience of investors.

ETFmatic[3] is a sophisticated online investment platform available in select European countries. It simplifies the process of building and managing portfolios using Exchange-Traded Funds (ETFs) through algorithmic techniques. By considering individual risk profiles, financial goals, and time horizons, ETFmatic creates customised investment portfolios. However, ETFmatic focuses exclusively on ETF-based investing and does not support individual stock holdings. Consequently, it does not offer stock screening or profiling features.

Another tool worth mentioning is SeekingAlpha [4] - a comprehensive online platform catering to investors seeking in-depth financial analysis and insights, with a primary focus on stock market investments. It offers a vast array of information, research, and analysis from a diverse community of contributors, including individual investors and industry experts. While SeekingAlpha provides portfolio management and performance tracking features, it lacks data visualisation in the form of charts.

Yahoo Finance[5] is a comprehensive online platform that provides a wide range of financial tools and resources. It offers real-time stock quotes, historical price data, charts, and a robust news platform with extensive coverage from reputable sources. Moreover, the data mentioned above is also available through an open-source and robust API. Although Yahoo Finance offers portfolio management and performance tracking features, they are very primitive and lack visualisation.

In the landscape of existing tools, I propose PortfoPal as a smart solution that combines the functionalities of robo-advisors and comprehensive financial platforms. PortfoPal aims to empower users by offering them the ability to create personalised portfolios using different investing strategies. By leveraging the stock analysing and profiling features, PortfoPal enables users to make informed decisions based on efficient and smart stock analysis.

With PortfoPal, users have access to a diverse range of tools and features that will help them shape their portfolios according to their preferences and assumptions about the market. The Portfolio Creator functionality allows users to filter and select holdings based on their investment goals and

[3] https://etfmatic.com/

[4] https://seekingalpha.com/

[5] https://finance.yahoo.com/

strategies. They can use the robust stock profiling, analysis, and advertising capabilities of PortfoPal to gain insights into the potential growth of specific stocks.

By combining the best aspects of robo-advisors and comprehensive financial platforms, PortfoPal offers users a powerful arsenal of tools to optimise their investment portfolios. This approach ensures that users have the flexibility to customise their portfolios while benefiting from advanced stock analysis and profiling features. With PortfoPal, investors can make more informed and strategic investment decisions, leading to potentially better outcomes in their investment journey.

1.3. Application functionalities

1.3.1. Authentication

PortfoPal offers a range of personalised functionalities that enhance the user experience but accessing them requires authentication. To log in, users are required to provide their email and password inside the login form (Fig 3). However, even without logging in, users can still access certain features such as the News Articles page, Stock Screening and Profiling functionality (Fig 4).

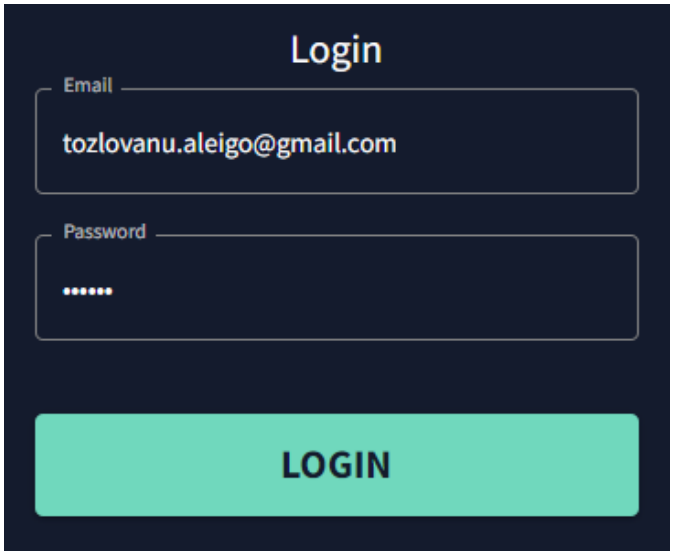


Figure 4 - Login Form

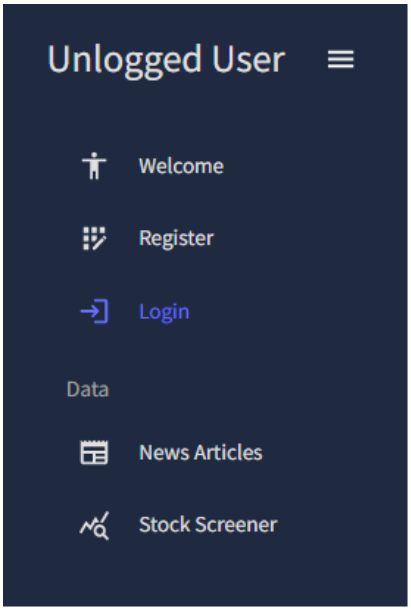


Figure 5 - Unlogged Sidebar

In cases where users do not have an account, they can easily navigate to the register page, where a form awaits their information. Like the login form, the register form validates all the input data and provides friendly messages in case of any incorrect or missing information (Fig 5). Once the register form is correctly filled, users will receive a confirmation code via email. They need to input this code in the following confirmation form (Fig 6).

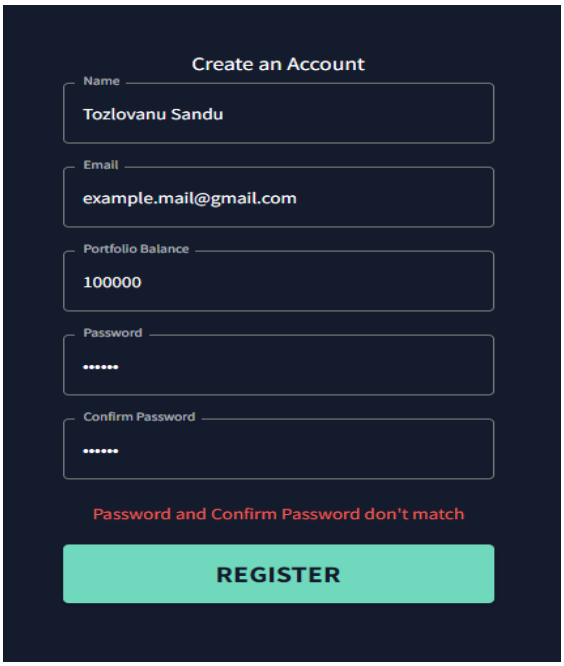


Figure 6 - Register form

In situations where users do not receive the confirmation code, a repeated code can be sent after a request is made. After a successful email confirmation, users are redirected back to the login form. If login process is completed, they will receive a JWT[6], which will be stored in the local storage of their device. This token will serve as authentication for future API calls, ensuring secure access to PortfoPal's features and data.

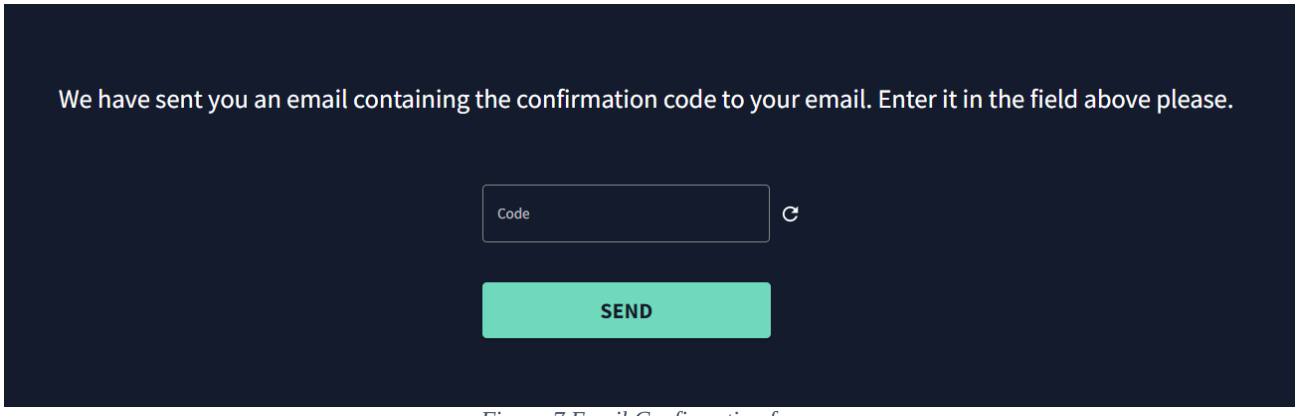


Figure 7 Email Confirmation form

1.3.2. Stock Screening

The Stock Screener page in PortfoPal offers users a convenient way to filter stocks based on various fundamental criteria. Users can access this page either by navigating to the "*/stockScreener*" route or by selecting it from the sidebar menu. Once on the Stock Screener page, users will find six fundamental criteria for filtering stocks: Market Cap, Price, Beta, Volume ranges, and the desired sector.

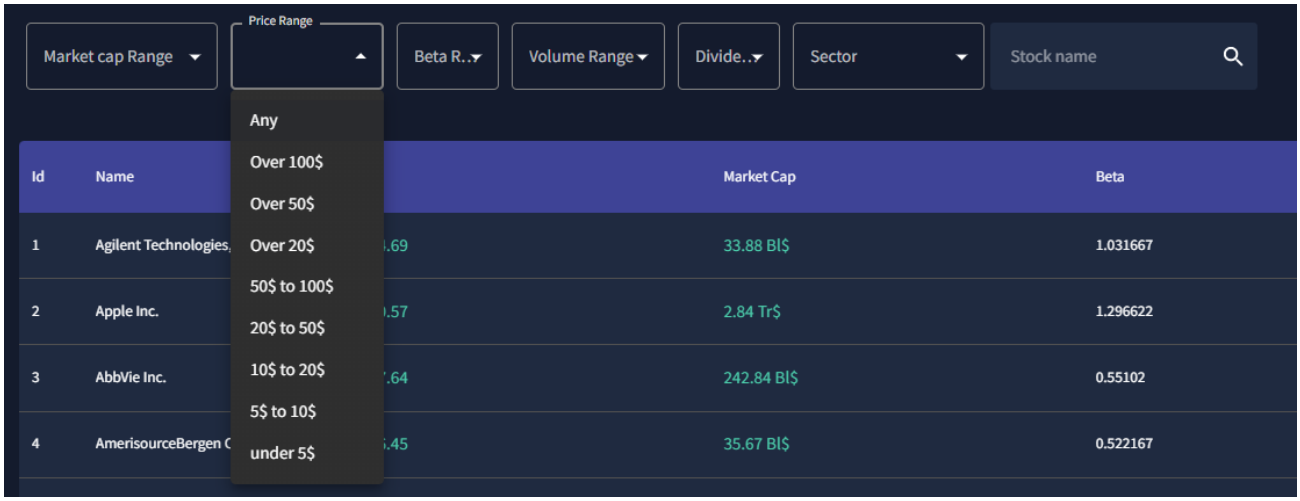


Figure 8 - Filter Dropdown

[6] JSON Web Token - defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

These criteria are presented as dropdown menus, allowing users to select their preferred values. Upon selecting the desired filters, an API request is sent to the backend, which retrieves a list of stocks along with their corresponding statistics. The stocks and their statistics are then displayed in a table format. The table offers convenient features such as pagination, allowing users to navigate through multiple pages of stock results. Users can also further filter the stocks by specific parameters such as price, market cap, beta, or sector (Fig 7).

Additionally, there is an input field near the filter dropdowns where users can enter a stock name (Fig 8).

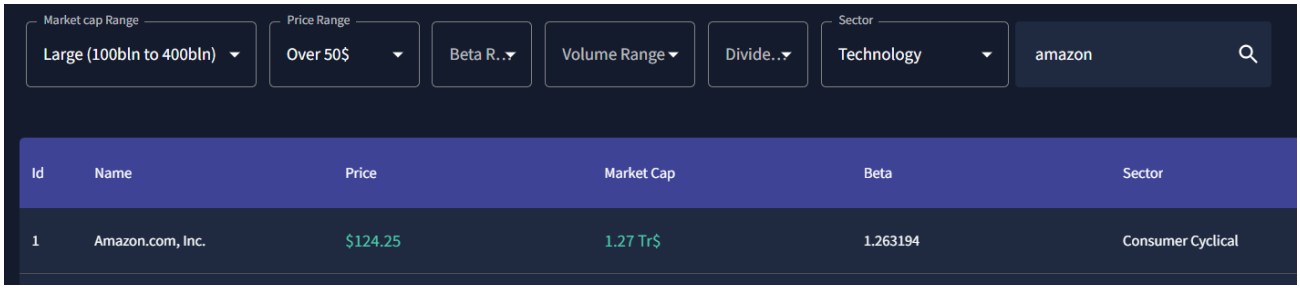


Figure 8- Stock Screener search

This triggers an API request to search for the company whose stock name or ticker is most like the input value. The backend implements a Sequence Matcher component from the "*difflib*[7]" Python library to perform this similarity search. Once users find the desired stock, they can click on the corresponding table row, which redirects them to the *"/stockProfile/{ticker_name}"* route. This route leads to the Stock Profiling page, where users can access in-depth information and analysis specific to the selected stock.

1.3.3. Stock Profiling

Upon accessing the stock profile page in PortfoPal, users are presented with comprehensive data and insights about the selected stock. The page begins with key financial statistics of the stock, including its current stock price, market capitalisation, average trading volume, beta, number of

[7] https://docs.python.org/3/library/difflib.html

employees, and the stock market where it is listed. A concise company description, highlighting the most relevant aspects of the stock is also provided (Fig 9).

Figure 9 - Stock profiling stats

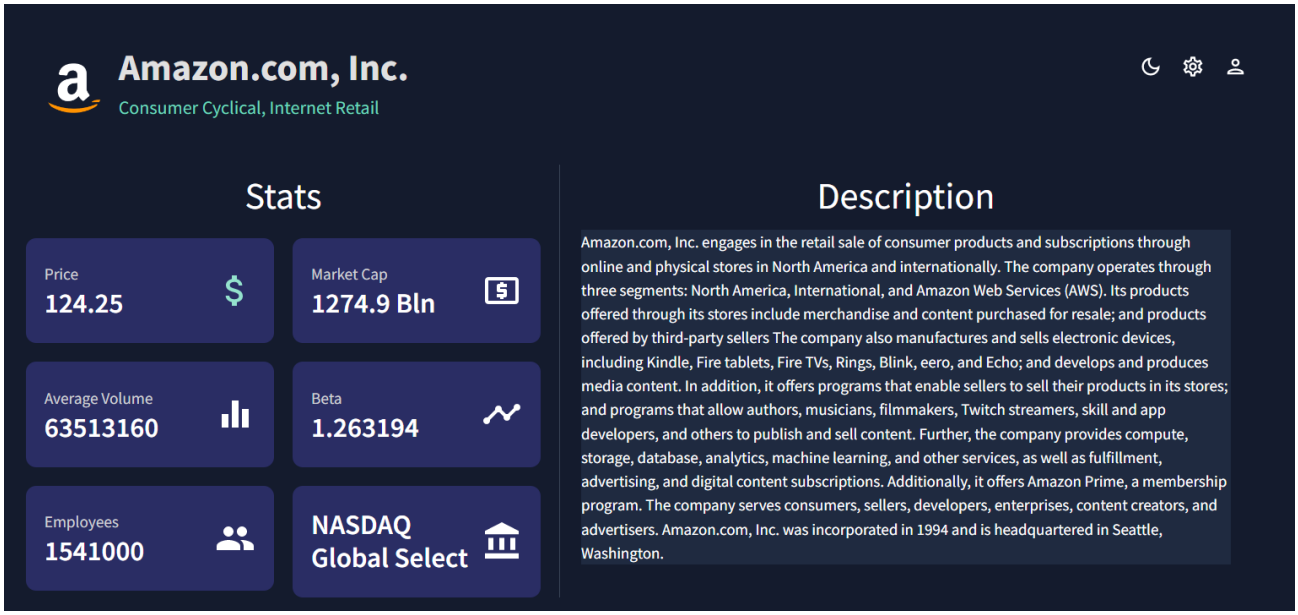


Figure 10 - Financial Statement stats

Beneath this information, users will find the main statistics from the stock's most recent financial statement. This section covers various aspects such as valuation, income and expenses, margins and quality, and balance sheet items (Fig 10), providing a comprehensive summary of the stock's financial performance.

A series of charts at the bottom of the page offer further details about the development and future of the company. The first chart (Fig 11) is the Price Chart, which is an interactive chart displaying the stock market price from two thousand year to the present day (Fig 10). Users can analyse historical price trends and fluctuations to gain a deeper understanding of the stock's performance.

Figure 11- Stock Profiling price chart

The second chart is the Revenue Bar Chart, representing the company's revenue as reported in its last ten quarterly financial statements (Fig 12). This chart visually displays the revenue figures over time, allowing users to assess the company's financial growth or decline.

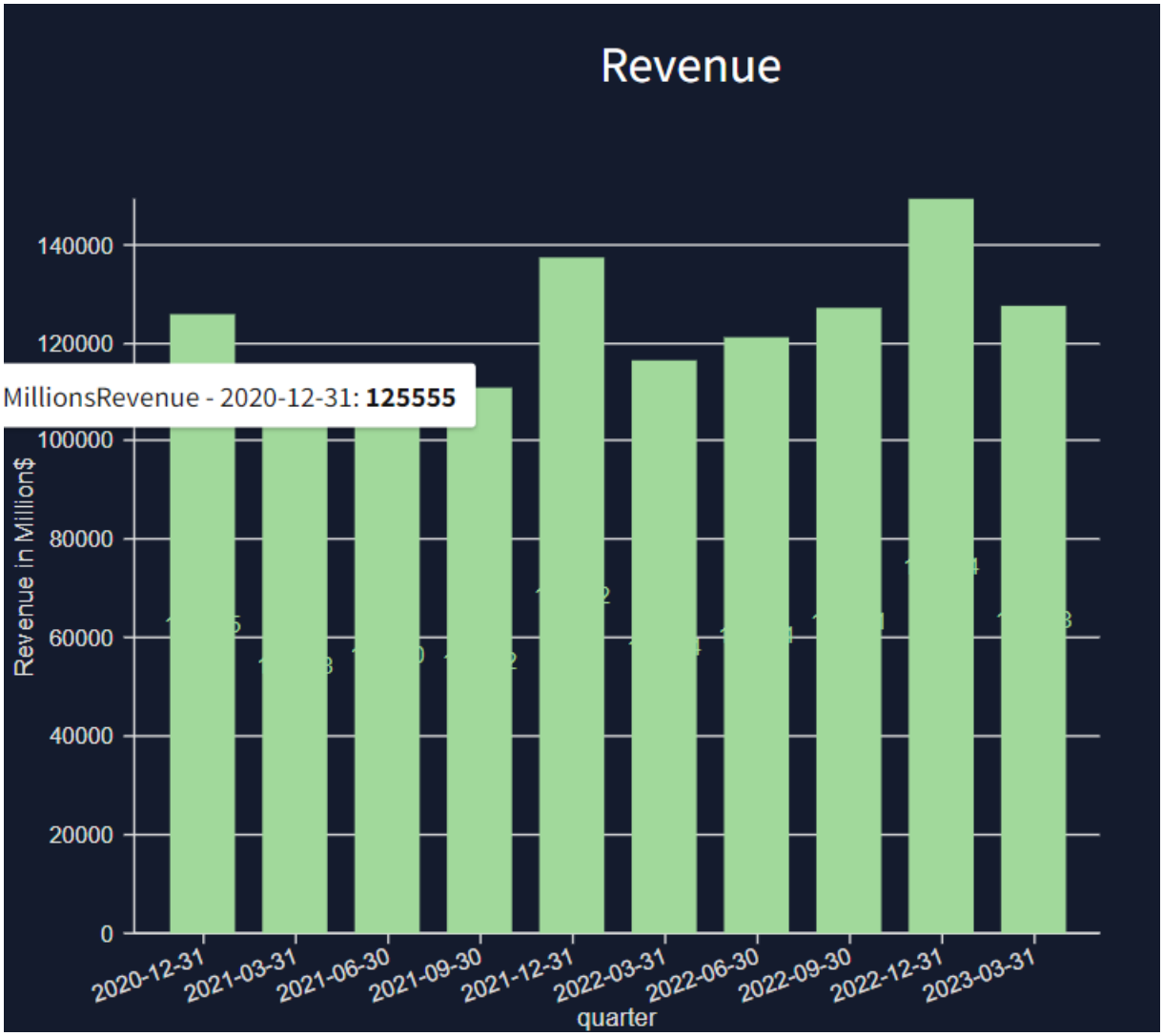


Figure 12 - Stock Profiling Revenue chart

Lastly, there is a Price Prediction Chart generated using the Prophet[8] statistical forecasting algorithm. This chart offers a prediction of the stock's price evolution for the next five years (Fig 13). It uses data collected from the Yahoo Finance API, providing users with a forecasted outlook to assist in their investment decisions.

Collectively, these charts and financial data empower users to gain valuable insights into the selected stock's historical performance, current financial standing, and potential future trajectory.

[8] Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

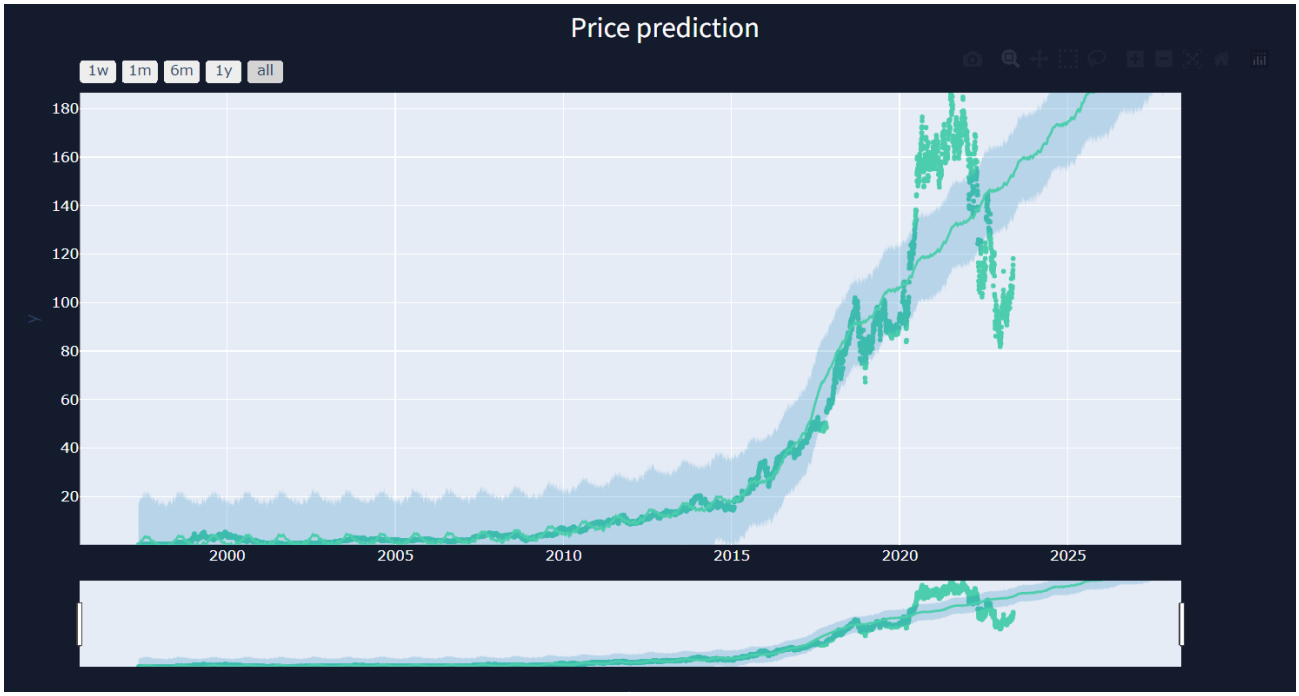


Figure 13 - Stock Profiling price prediction chart

1.3.4. Portfolio creation using investing strategies

To access the Portfolio Creator page in PortfoPal, users can either use the *"/portfolioCreator"* route or select it from the sidebar. However, both techniques require the user to be logged in. Once on the page, users are given the option to build their own portfolios using a certain investing strategy. There are five available strategies to choose from:

1. Equal-Weight S&P 500 Index Fund Investing (Fig 14)
2. Weighted S&P 500 Index Fund Investing (Fig 14)
3. Quantitative Momentum Investing (Fig 15)
4. Quantitative Value Investing (Fig 15)
5. Quantitative Momentum & Value Investing (Fig 16)

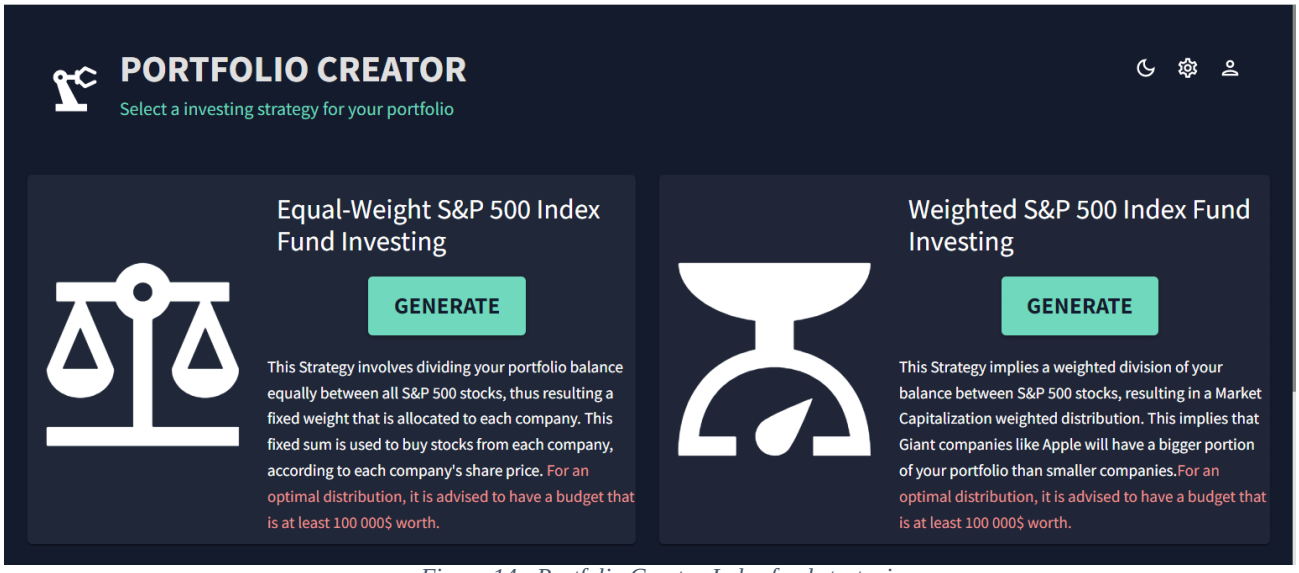


Figure 14 - Portfolio Creator Index fund strategies

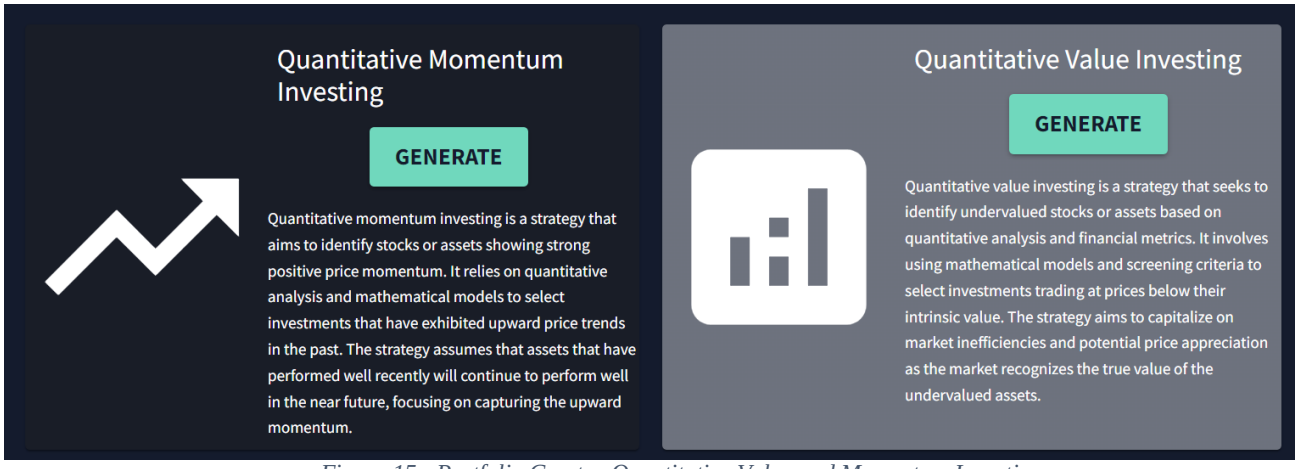


Figure 15 - Portfolio Creator Quantitative Value and Momentum Investing

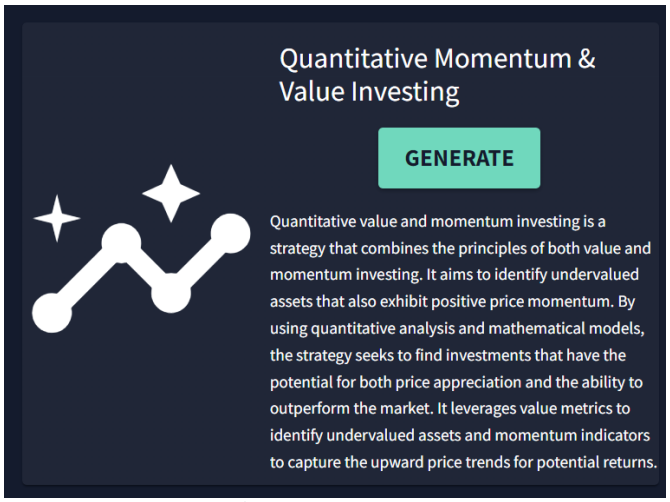
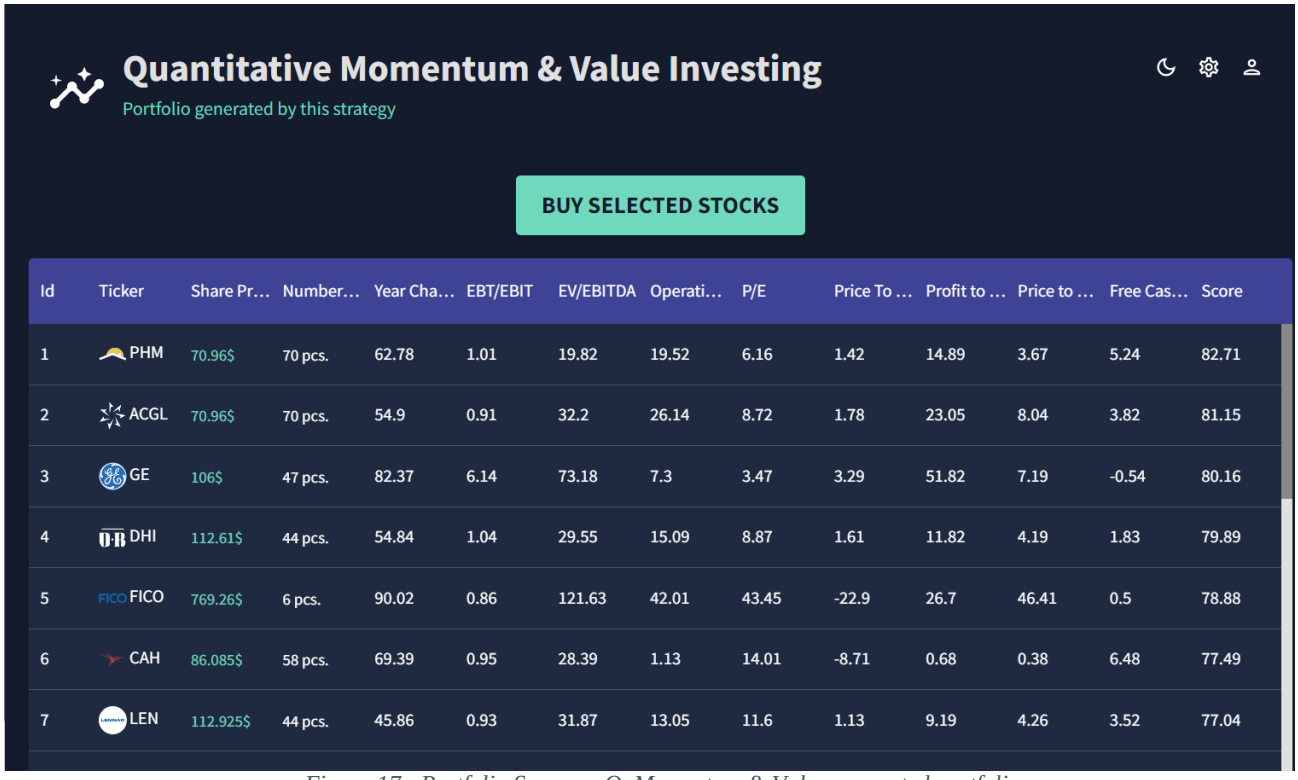


Figure 16 - Portfolio Creator Strategy Combination

Each strategy uses different heuristics, which are detailed in Chapter 3. After selecting one of the five strategies and clicking the Generate button, an API call is made to the backend to create a portfolio based on the chosen strategy. If the API call was successful, the user is redirected to the *"/portfolioScreener/{strategy_name}"* route.

The Portfolio Screener page consists of two main components. The first component is the "Buy Selected Stocks" button, which allows the user to purchase the entire generated portfolio. Upon clicking this button, the user is also routed to the *"/myPortfolio"* route to view their newly created portfolio.

The second component is a table displaying all the stocks within the portfolio generated by the algorithm. This table is interactive, allowing users to switch between pages and filter the table by various columns. The columns in the table display statistics relevant to the selected strategy. For example, a value investing strategy will include fundamental stock statistics, while a momentum strategy will focus on price change statistics (Fig 17).



Id	Ticker	Share Pr…	Number…	Year Cha…	EBT/EBIT	EV/EBITDA	Operati…	P/E	Price To …	Profit to …	Price to …	Free Cas…	Score
1	PHM	70.96$	70 pcs.	62.78	1.01	19.82	19.52	6.16	1.42	14.89	3.67	5.24	82.71
2	ACGL	70.96$	70 pcs.	54.9	0.91	32.2	26.14	8.72	1.78	23.05	8.04	3.82	81.15
3	GE	106$	47 pcs.	82.37	6.14	73.18	7.3	3.47	3.29	51.82	7.19	-0.54	80.16
4	DHI	112.61$	44 pcs.	54.84	1.04	29.55	15.09	8.87	1.61	11.82	4.19	1.83	79.89
5	FICO	769.26$	6 pcs.	90.02	0.86	121.63	42.01	43.45	-22.9	26.7	46.41	0.5	78.88
6	CAH	86.085$	58 pcs.	69.39	0.95	28.39	1.13	14.01	-8.71	0.68	0.38	6.48	77.49
7	LEN	112.925$	44 pcs.	45.86	0.93	31.87	13.05	11.6	1.13	9.19	4.26	3.52	77.04

Figure 17 - Portfolio Screener Q. Momentum & Value generated portfolio

By providing users with a choice of investing strategies and an intuitive interface to create and manage their portfolios, PortfoPal empowers investors to implement personalised investment approaches based on their preferred strategies.

1.3.5. Portfolio management and performance monitoring

The My Portfolio page in PortfoPal can be accessed by logged in users through the "/myPortfolio" route or by selecting it from the sidebar. Users can manage their stock portfolios on this page by buying and selling equities.

To buy stocks, users can click on the Buy Stocks button. A form appears after pressing this button and within it, users can select the desired stocks and specify the number of shares they wish to purchase. For selling stocks there is a similar procedure, but users can only select the stocks they already own (Fig 18).

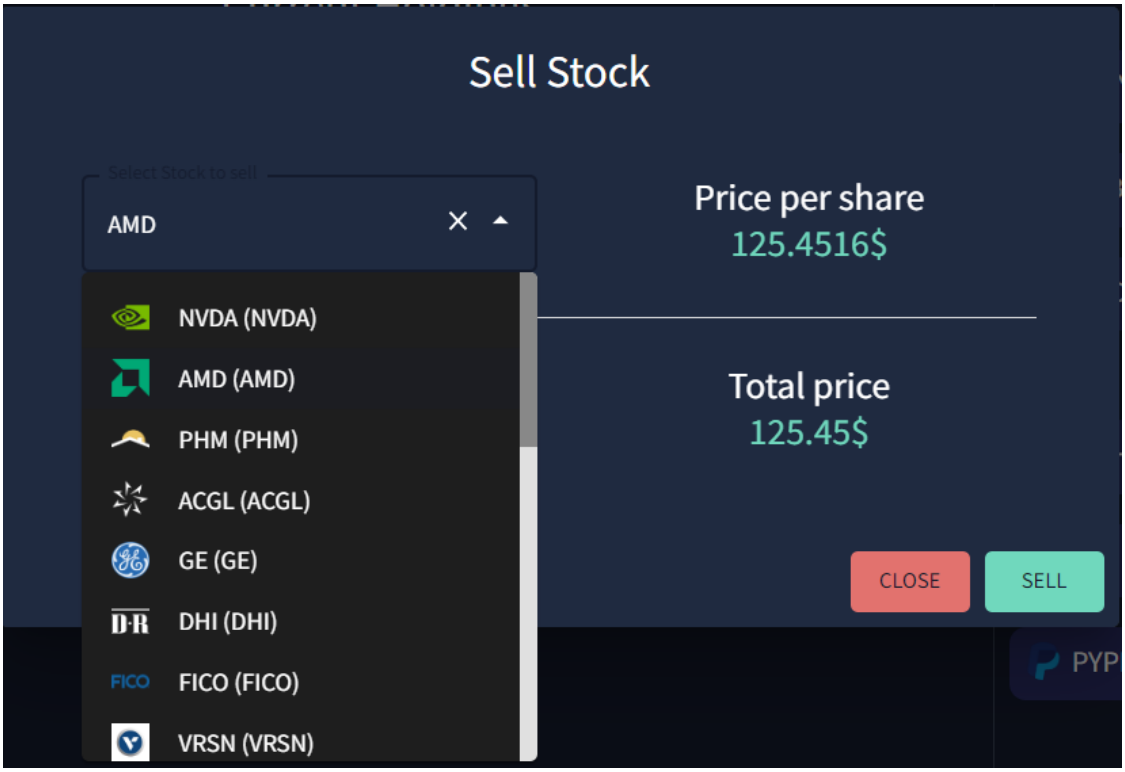


Figure 18 - My Portfolio Sell stock form.

In addition to facilitating transactions, the My Portfolio page provides tools for tracking and monitoring stock performance. One such feature is the interactive Pie Chart (Fig 19), which visually represents the user's total holdings and their respective proportions within the portfolio.

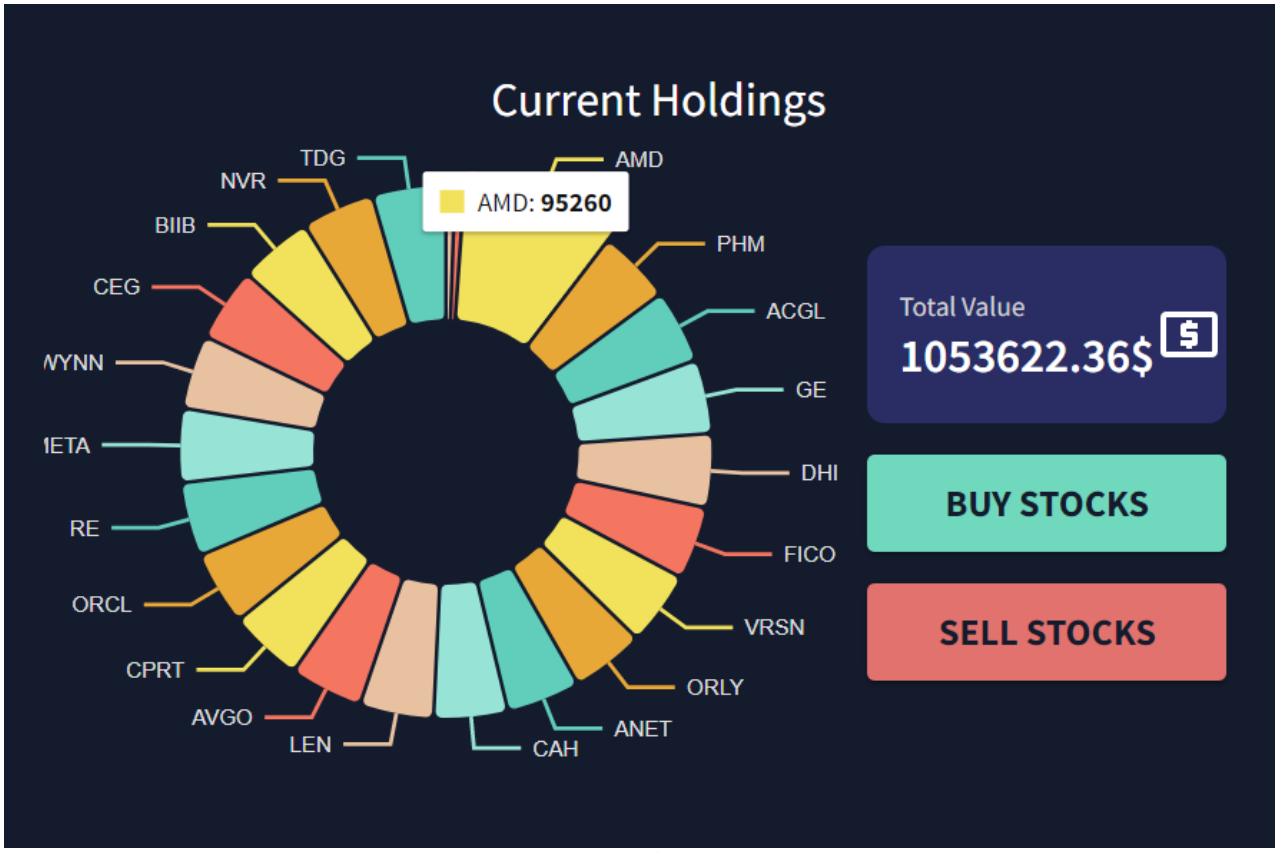


Figure 19 - My Portfolio pie chart

The page also includes a Portfolio Stats component, offering important metrics such as the portfolio's overall value, records, and a comparison to a benchmark value (Fig 20).

Figure 20 - My Portfolio metrics

For detailed tracking purposes, the page displays the performance of individual holdings. This feature presents the evolution of each stock's price within the portfolio, allowing users to monitor their investments in a user-friendly manner (Fig 21).

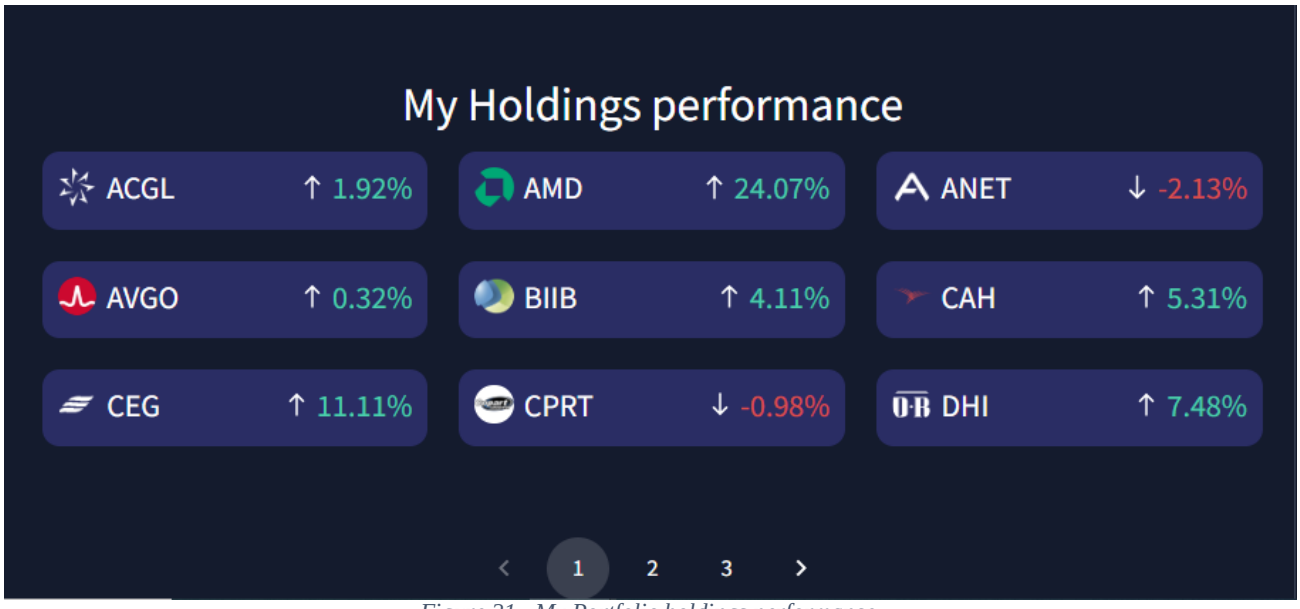


Figure 21 - My Portfolio holdings performance

Perhaps the most impressive aspect of the My Portfolio page is the Portfolio Performance Chart. This chart illustrates the changes in the portfolio's value from its inception until the present day (Fig 22). To accomplish this, the platform implements a "Portfolio Value Tracking Algorithm" on the backend, providing users with valuable insights into the overall performance of their portfolio over time.

By combining portfolio management functionalities, stock performance tracking tools, and informative visualisations, PortfoPal empowers users to effectively manage their investments and make informed decisions based on their portfolio's performance.

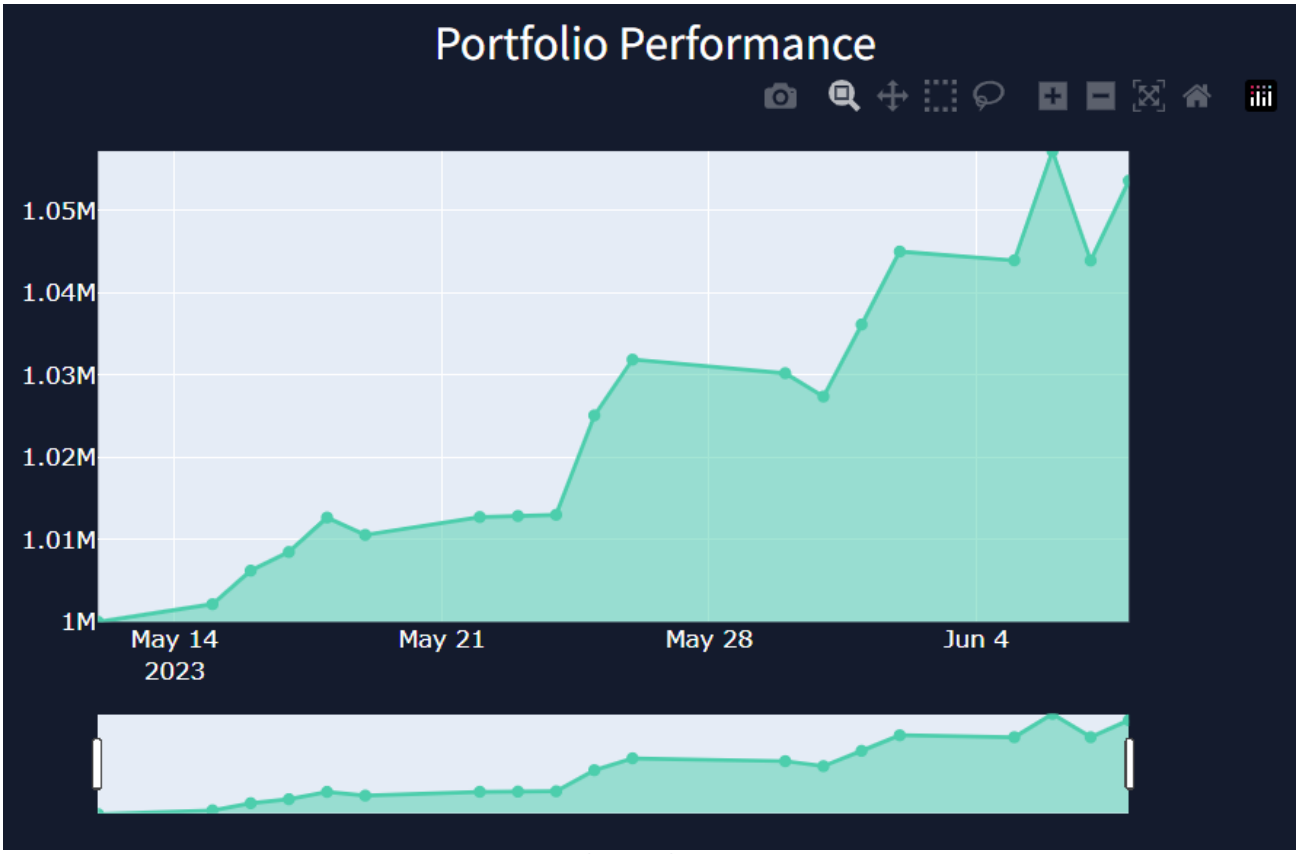


Figure 22- My Portfolio performance chart

The last but certainly not least important component of the portfolio tracking feature is the History Transaction page, which can be accessed through the "/transactions" route or by selecting "My Transactions" from the sidebar. This page offers users a comprehensive list of all the transactions conducted within their account.

Presented in the form of a table (Fig 23), the History Transaction page provides users with crucial details related to each transaction. These details include:

- the ticker symbol of the stock involved in the transaction,
- the type of operation (whether it was a buy or sell),
- the price per share at the time of the transaction,
- the date on which it took place.

Users can also easily review and analyse their past actions using the transaction history, gaining insights into their trading patterns, investment decisions, and overall portfolio activity. This

historical data allows users to track their progress, evaluate the success of their investment strategies, and make informed adjustments to their portfolio when necessary.

The inclusion of the History Transaction page in PortfoPal's platform ensures transparency and accountability, providing users with a comprehensive overview of their past transactions and enabling them to maintain a well-documented record of their investment activities.

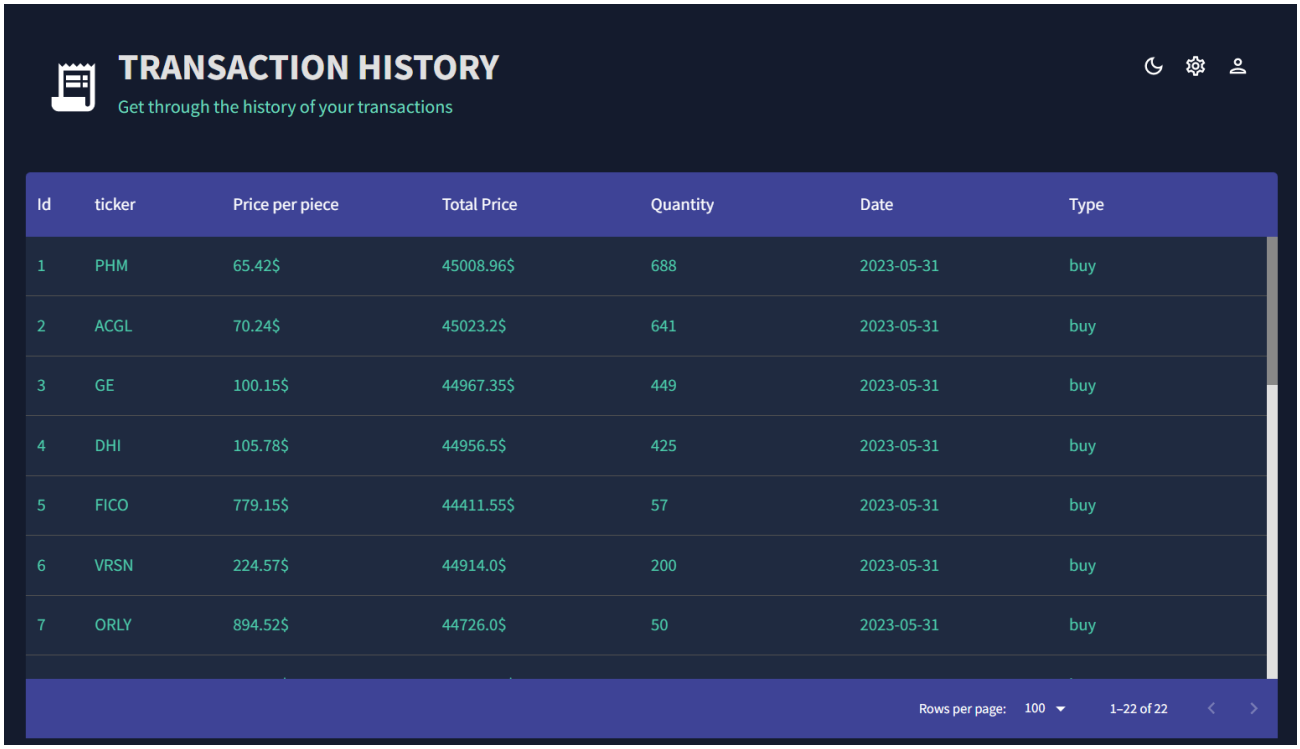


Figure 23 – My Transactions transaction table

1.3.6. Smart stock recommendations and news

The News Articles page is accessible through the "/news" route or by selecting it from the sidebar within the PortfoPal platform. This page presents articles in order of relevance to the user's interests to keep readers up to date with the most recent stock market news. (Fig 24).

To ensure a personalised experience, the news articles are divided into two categories, each comprising 50% of the content. The first category focuses on the stocks that the user already holds in their portfolio, in order to provide valuable insights and updates on the market, helping users stay informed about the current status and developments of their investments. The second category includes news articles related to other stocks that might be of interest to the user, covering stocks that are potential candidates for investment, aligning with the user's investment preferences, industries, and sectors.

It is important to note that all news articles are arranged in chronological order, with the most recent articles displayed first. This allows users to stay updated on the latest market trends and developments as they occur.

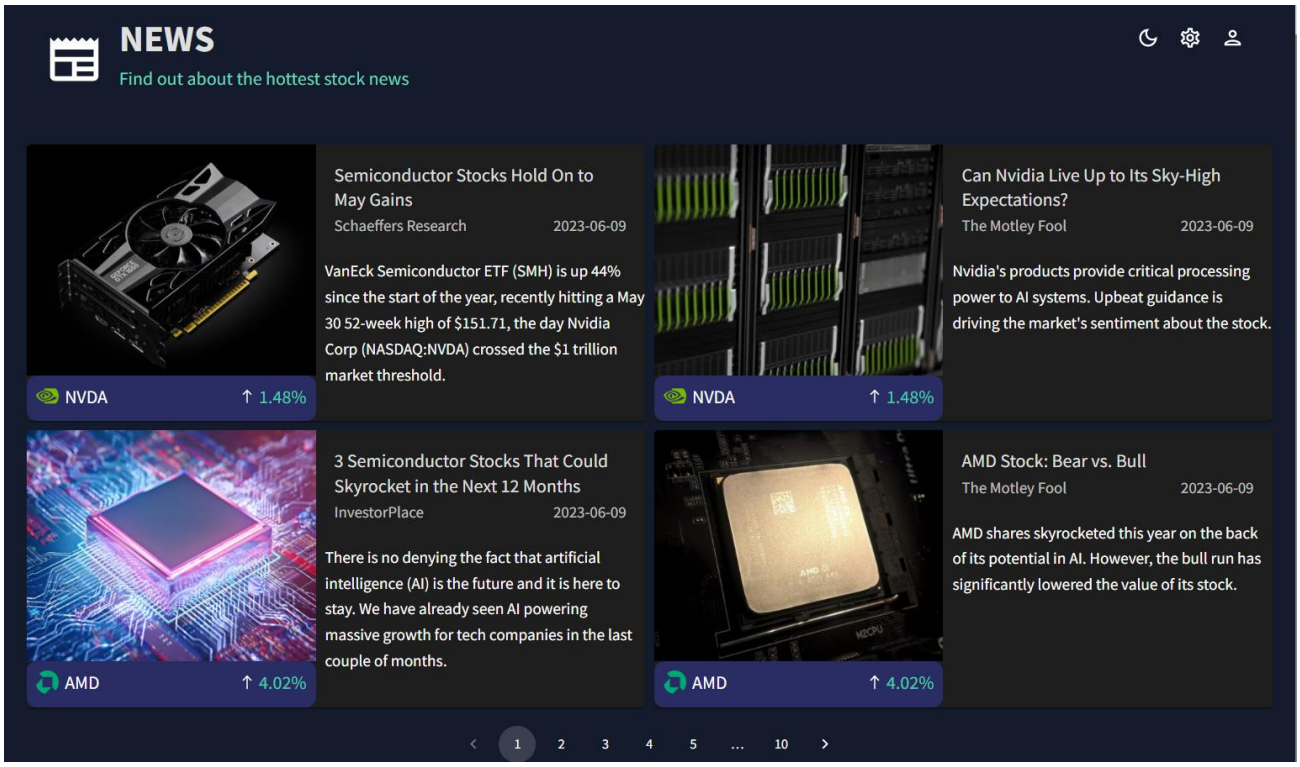


Figure 24- News Articles

In addition to the News Articles page, the My Portfolio page also features the right section populated with stock tickers from three distinct categories (Fig 25).

The first category highlights market gainers, showcasing stocks that have experienced the most significant value increases during the current day. This information helps users identify stocks that are performing well in the market.

The second category focuses on market losers, featuring stocks that have witnessed the most significant declines in value for the day. This data enables users to stay informed about stocks that are experiencing downward trends.

Lastly, the “Stocks for You” category presents stocks that belong to the same industries and sectors as the user's current portfolio holdings. By suggesting stocks within the user's area of expertise, this feature helps users explore potential investment opportunities that align with their investment strategy and knowledge.

The inclusion of the News Articles page and the stock ticker categories within the My Portfolio page enhances the user's overall experience on the PortfoPal platform, providing them with relevant news updates and valuable insights into the market and potential investment options.

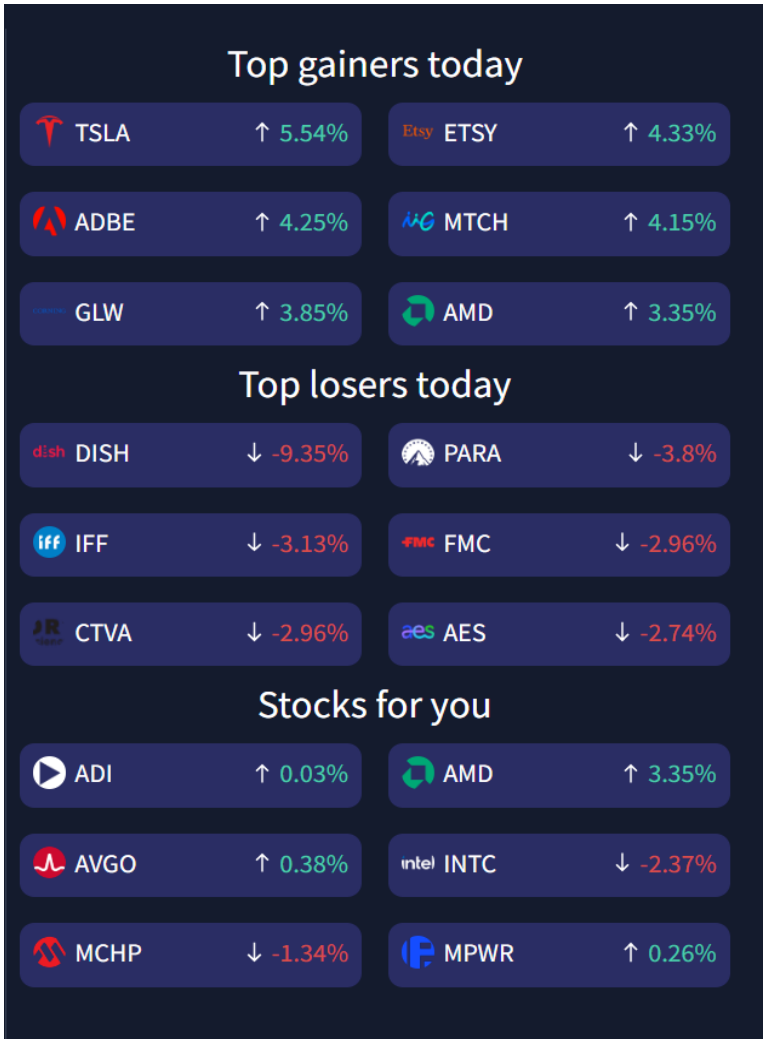


Figure 25 - My Portfolio Stock Recommendations

Chapter 2: Application architecture

Chapter 2 is all about our application: the general aspects of it, the back-end and the front-end implementation, the data management process and concerns about the security and privacy. Here we get to know the Portfopal application in many details and see all the strengths it has.

2.1. General Aspects

PortfoPal is an extensive client-server application that combines a user-friendly front-end interface with a reliable back-end server. The front-end application presents thousands of stock data in simple-to-read charts and features for a smooth and intuitive user experience. It adheres to essential principles of UI/UX design and responsiveness, ensuring a visually appealing and accessible platform.

On the other hand, the back-end server oversees complex data processing and management tasks to serve user requests efficiently. The architecture of PortfoPal involves the logical division into two servers: the front-end server and the back-end server. Within the back-end server, there are three microservices: private microservice, public microservice, and portfolio creator microservice (Fig 26). This microservice architecture allows the separation of concerns and encapsulation of specific functionalities within each service. It enables easy deployment, scalability, and independent management of different components of the system.

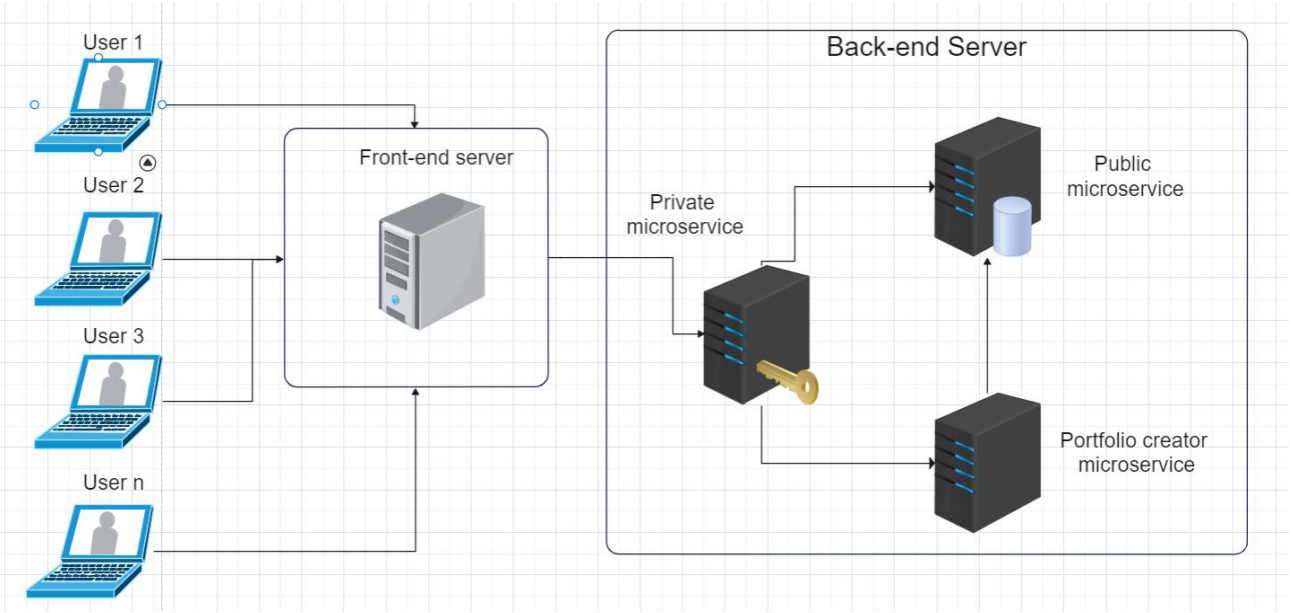


Figure 26 – System architecture diagram

It is worth mentioning that all microservices are provided with a Swagger/OpenAPI documentation generated using “flassger” library that enhances development team collaboration and interoperability, thus improving developer experience.

Additionally, all backend services use DTOs to convey API responses, allowing the efficient and structured data exchange, while contributing to enhanced security by excluding sensitive or confidential information from API response.

Git is employed as the version control system for the project, facilitating collaboration among developers and ensuring project stability and backup through repositories. The Flask framework is chosen for all the microservices due to its flexibility, compatibility with the MVC design pattern, and extensive library support. This decision is particularly significant as the project relies on various Python libraries and modules.

For the front-end development, the React framework is used, leveraging its component-based architecture approach. This approach promotes code reusability, maintainability, and scalability, which are crucial aspects of the PortfoPal project. The use of React allows the efficient development and management of the user interface.

In terms of data management, SQLite is the chosen database system. It is favoured for its ease of use, ACID compliance, and high performance. These attributes make it an ideal choice for building fast and reliable APIs that can handle the substantial amount of data involved in the PortfoPal platform.

2.2. Back-end implementation

2.2.1 Overall back-end design decisions

To achieve a scalable, flexible, and modular architecture, all the microservices within PortfoPal implement the Model-View-Controller (MVC) design pattern. This design pattern offers essential features that contribute to its effectiveness as an architectural approach. It enables the application to be divided into three distinct components: the Model, View, and Controller.

The Model component is responsible for handling data and implementing business logic. It ensures data integrity, manipulation, and storage, serving as the backbone of the application that manages data-related operations and maintains consistency.

The View component, on the other hand, focuses on presenting data to users and providing an intuitive user interface. It encompasses the visual elements and user interaction components that are designed to deliver a seamless and user-friendly experience, ensuring that users can easily access and navigate through the application. The last component, the Controller, acts as a mediator between the Model and View components. It manages user input, processes requests, and orchestrates the interaction

between the Model and View. The Controller ensures synchronisation between the data and the user interface, handling user actions and updating the Model and View accordingly.

PortfoPal adheres to the Separation of Concerns principle by mixing microservices with the MVC design, resulting in a well-organized code structure at each application level. This separation allows for easier development, maintenance, and scalability of the application. Each microservice can focus on its specific functionality while maintaining a clear division of responsibilities.

In order to improve system performance, reduce latency, and enhance the user experience, PortfoPal implements a caching mechanism using the "*flask caching*" library. This mechanism involves setting cache timeouts for endpoints based on their data change rate. By caching frequently accessed endpoints, such as those in the public and portfolio creator microservices, the application can deliver faster responses and reduce the load on the backend, resulting in a more efficient and responsive user experience.

2.2.2 Private microservice

The Private Microservice within PortfoPal fulfils the responsibilities of two types of microservices. Firstly, it acts as a business logic provider, offering access to private user portfolio data and management functionalities. Secondly, it serves as a proxy, providing caching capabilities, request authentication, and an API rate limiter. This microservice acts as an intermediary between the frontend and the other two microservices, in order to ensure safe communication and secure the backend services against unauthorized access.

This architectural decision has been made to add an additional layer of security and protection to the application. By centralising access to private user data within the Private Microservice, the other backend services are shielded from direct exposure. This helps to safeguard sensitive information and mitigate potential security risks. Moreover, combining these functionalities into a single microservice simplifies the general development and maintenance of the project.

However, this approach has some negative consequences. It breaks the modularity principle to some extent, as the Private Microservice takes on multiple responsibilities that would typically be divided among separate microservices. While this decision simplifies certain aspects of the application, it may introduce some complexities in terms of scalability and separation of concerns.

To adhere to the MVC architecture, the Private Microservice provides four controllers to the views (Fig 27):

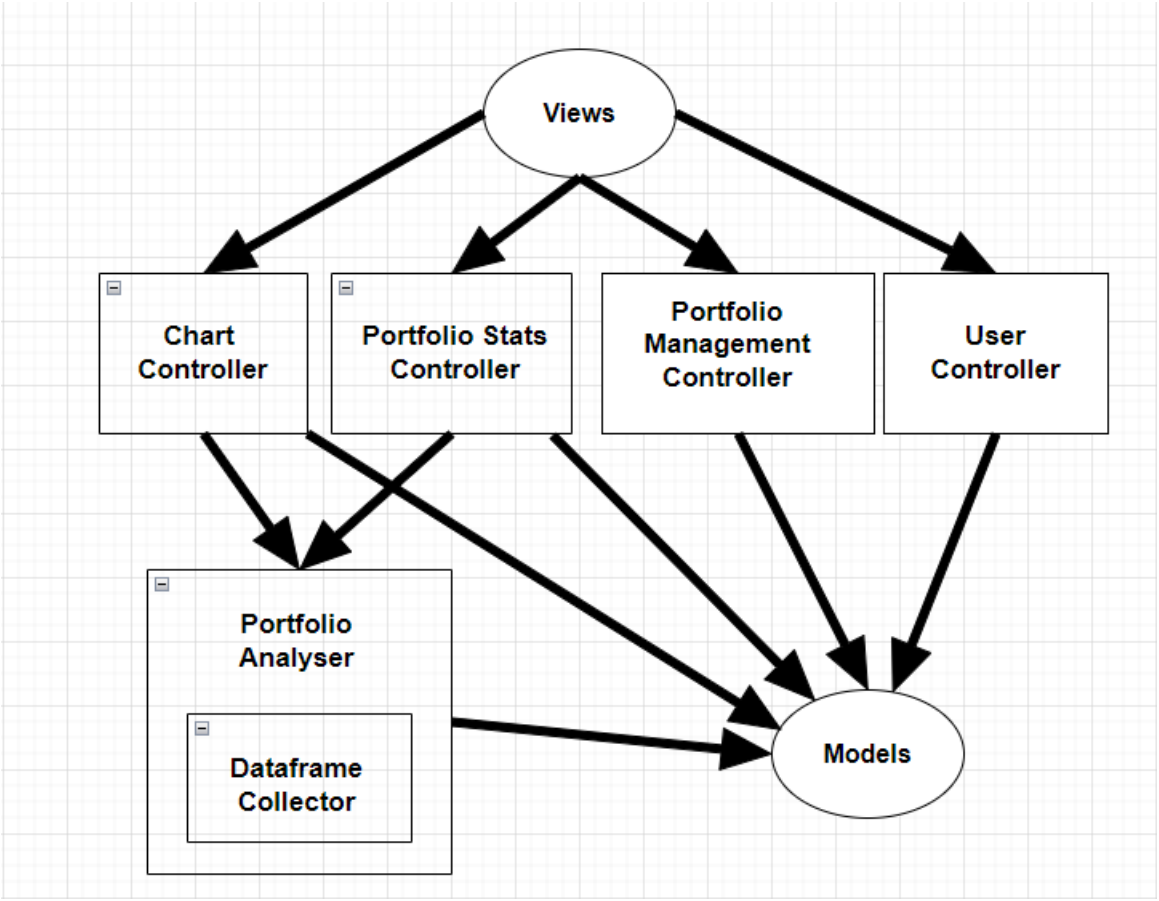


Figure 27 – Private Microservice MVC diagram

The Portfolio Management Controller implements the business logic for stock buying and selling features. It handles the processes related to managing the user's portfolio, ensuring proper execution of transactions.

The User Controller handles user authentication, registration, and JWT (JSON Web Token) handling. It is responsible for managing user login and registration processes, validating credentials, and generating and handling JWTs. Additionally, it manages email confirmation logic, ensuring the security and integrity of user accounts.

The Chart Controller generates chart data from the database for the frontend. It retrieves relevant data and formats it in a way that can be easily visualised on the frontend, enabling the display of various charts and graphs related to stock market data and portfolio performance.

The Portfolio Stats Controller manages portfolio statistics for each user. It calculates and provides essential metrics and insights regarding the user's portfolio, such as overall value, historical performance, and benchmark comparisons, ensuring that users have access to relevant and up-to-date information about their portfolios.

The last two Controllers use an additional module called Portfolio Analyser, which implements the Portfolio Value Tracking Algorithm. Its primary objective is to generate a data frame consisting of two columns: Date and portfolio value on that specific date. The algorithm achieves this by creating a comprehensive record of the portfolio's value from its inception date until the last available trading day.

Two main data structures play significant roles in the algorithm:

- **Transaction:** This class represents a transaction made on a specific stock. It includes attributes such as the stock name, transaction date, price per share, number of shares, transaction type (buy or sell), and the portfolio's cash balance after the transaction.
- **Hold Period:** This structure represents a period during which a particular stock and its corresponding quantity of shares were held in the portfolio. Hold periods track the duration for which a stock remained in the portfolio, regardless of any subsequent transactions.

The Portfolio Value Tracking Algorithm operates as follows:

Data frame Initialisation: The algorithm begins by populating a data frame with the dates of an arbitrary stock (for example "AAPL") and setting the portfolio value for each date as 'nan.'[9] This step ensures that the data frame includes all the relevant dates for tracking portfolio performance.

Initialising the data frame with Residual Cash: The algorithm then updates the value column of the data frame created in the previous step (which is currently 'nan') with the portfolio cash balance after each transaction. This step establishes the data frame's initial values, considering only the remaining cash balance after all transactions, without accounting for the value of stocks held.

Generating Hold Periods: The algorithm calls the "generateHoldPeriods" method (Fig 28), which creates a dictionary with the stock ticker[10] as the key and a list of transactions for that stock sorted in ascending order by date. For each stock in this dictionary, the "generateTickerHoldPeriod" method (Fig 28) is invoked. This method creates a Hold period for each transaction, except the first one. The first transaction is always a purchase, indicating when the stock was acquired. Two variables, last_date and current_quantity, are initialised based on the first transaction and subsequently updated for each following transaction. Each transaction generates a new Hold period, updating the current_quantity and last_date values. At the end of this process, if the current_quantity is nonzero, it

[9] Nan – Not a Number

[10] Ticker – "A stock symbol or ticker is a unique series of letters assigned to a security for trading purposes" - https://www.investopedia.com/terms/s/stocksymbol.asp

indicates that there are remaining shares of the stock in the portfolio, and an additional Hold period is added to account for these shares.

```
def __generateTickerHoldPeriods(transactions: list[Transaction]) -> list[HoldPeriod]:
    hold_periods_to_return: list[HoldPeriod] = []
    ticker: str = transactions[0].ticker
    current_quantity: int = transactions[0].quantity
    last_date: datetime = transactions[0].date
    transactions.remove(transactions[0])
    for transaction in transactions:
        hold_periods_to_return.append(
            HoldPeriod(ticker=ticker, quantity=current_quantity,
                       start_date=last_date, end_date=transaction.date))
        last_date = transaction.date
        if transaction.is_buy:
            current_quantity += transaction.quantity
        else:
            current_quantity -= transaction.quantity
    if current_quantity != 0:
        hold_periods_to_return.append(
            HoldPeriod(ticker=ticker, quantity=current_quantity, start_date=last_date,
                       end_date=None))

    return hold_periods_to_return
```

```
Tozlovanu Sandu
@staticmethod
def __generateHoldPeriods(transaction_list: list[Transaction]) -> list[HoldPeriod]:
    transaction_dict: dict[str, list[Transaction]] = {}
    for transaction in transaction_list:
        if transaction.ticker not in transaction_dict:
            transaction_dict[transaction.ticker] = [transaction]
        else:
            transaction_dict[transaction.ticker].append(transaction)
    hold_periods_to_return = []
    for ticker in transaction_dict:
        hold_periods_to_return += PortfolioAnalyser.__generateTickerHoldPeriods(transaction_dict[ticker])
    return hold_periods_to_return
```

Figure 28 – Portfolio Analyser generateTickerHoldPeriods and generateHoldPeriods methods code snippet

Populating the Data frame: For each Hold period, the algorithm retrieves the data frame containing the price per share for each date in the portfolio data frame. It then multiplies the price per share by the quantity of shares held during that period and adds this value to the portfolio data frame, thereby populating it with the corresponding portfolio value for each date (Fig 29).

```
def portfolio_value_tracking_algorithm(self, transaction_list: list[Transaction],
                    creation_time: datetime, money: float):
    df = self.dataframe_collector.get("AAPL", start=creation_time.strftime("%Y-%m-%d"),
                                      end=self.dataframe_collector.last_date)
    df["Close"] = np.nan
    df = df.rename(columns={'Close': 'Value'})
    transaction_list = self.filter_transaction_list(transaction_list)
    hold_periods = PortfolioAnalyser.__generateHoldPeriods(transaction_list)
    PortfolioAnalyser.__initDataframeCash(df, transaction_list, money)

    for hold_period in hold_periods:
        if hold_period.end_date is None:
            df2 = self.dataframe_collector.get(hold_period.ticker, start=hold_period.start_date)
        else:
            df2 = self.dataframe_collector.get(hold_period.ticker, start=hold_period.start_date,
                                               end=hold_period.end_date)
        if df2.shape[0] < 2 and df2['Date'][0].strftime("%Y-%m-%d") != hold_period.start_date.strftime(
                "%Y-%m-%d"):
            continue
        if df2['Date'].iloc[-1] != df['Date'].iloc[-1]:
            last_row = df2.iloc[-1]
            df2.loc[df.index.max() + 1] = last_row
            df2.reset_index(drop=True, inplace=True)
            df2['Date'].iloc[-1] = df['Date'].iloc[-1]

        df = pd.merge(df, df2, on='Date', how='left')
        df['Value_multiplied'] = df['Close'].fillna(0) * hold_period.quantity
        df['Value'] = df['Value'] + df['Value_multiplied']
        df = df[['Date', 'Value']]

    df['Value'] = df['Value'].round(3)
    df['Date'] = df['Date'].dt.strftime('%Y-%m-%d')
    return df
```

Figure 29 – Portfolio Analyser portfolio_value_tracking_algorithm method code snippet

To sum up, the Portfolio Value Tracking Algorithm effectively tracks the performance of the portfolio over time, providing valuable insights and data for portfolio analysis and monitoring. This proves, once again, why the Private Microservice serves as a critical component within the PortfoPal application, offering a wide range of features through its API endpoints. It stands out as one of the most important microservices due to its ability to provide highly personalised and user-related data compared to other microservices.

2.2.3 Public microservice

The Public Microservice is a critical component of the back-end architecture in PortfoPal. It serves as the central hub for storing, collecting, and providing all the stock information required by other microservices. To fulfil this role effectively, the Public Microservice operates two threads: the Data Collector thread and the Flask application.

The Data Collector module is responsible for retrieving relevant data from the "financialmodelingprep" API. It continuously collects up-to-date stock information and populates the database with the latest data. This ensures that the application has access to accurate and timely information. [11]

The Flask application module is the core of the Public Microservice. It implements the business logic of the microservice, handling all the requests received by extracting data from the database populated by the Data Collector. The Flask application processes the data according to the specific business requirements and responds to the requests accordingly.

In line with the MVC (Model View Controller) architecture, the Public Microservice provides views with six controllers:

1. The Stock Price Controller**,** which manages information related to stock tickers' prices and price changes over time. It retrieves data on stocks that have gained or lost the most within a specific period and provides current pricing information.
2. The Stock Profile Controller, that handles the business logic for retrieving company profile information. It includes details such as company descriptions, logo links, and various statistics.
3. Stock Financials Controller, which provides data on company ratios and financial statements, enabling users to access key financial information about specific stocks.
4. The Search Controller, that enables users to search for stocks based on different criteria. It implements an algorithm that finds the most similar stocks or tickers to a given search string.
5. The News Controller, responsible for providing the most relevant news to the user. It ensures that the latest stock market news is presented based on the user's interests and holdings.

[11] A more detailed description of the Data Collector module is provided in subchapter 2.4.1.

6. The Chart Controller, which generates and provides various charts, such as revenue charts, price prediction charts, and company price charts. These charts offer visual representations of important data for better understanding and analysis.

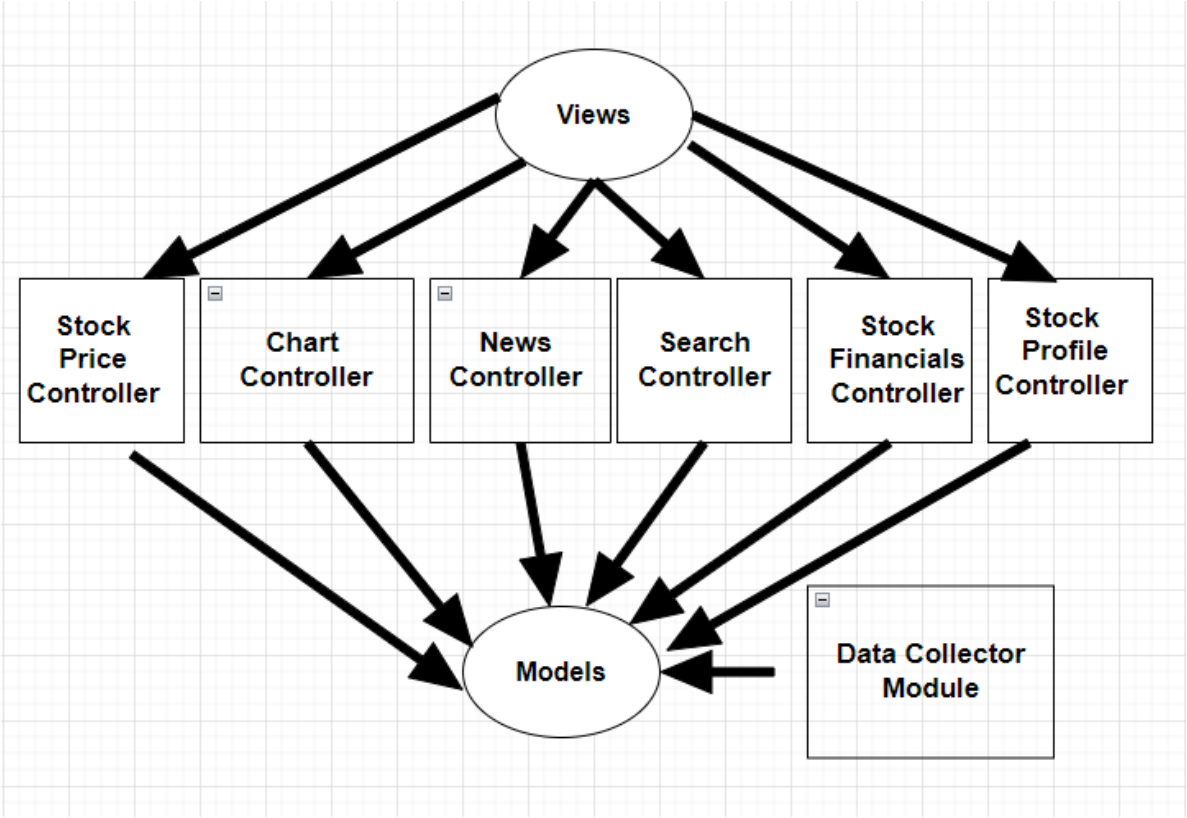


Figure 30 - Public Server MVC diagram

By leveraging the capabilities of these controllers, the Public Microservice offers a wide range of functionalities related to stock information, enabling users to make informed investment decisions and stay updated on the latest market trends.

2.2.4 Portfolio creator microservice

The Portfolio Creator microservice plays a crucial role in the development of investment portfolios for PortfoPal users. It serves as the central hub for implementing algorithmic techniques and investment strategies. This microservice heavily relies on consuming large amounts of financial data from the Public Microservice and utilises this data in various algorithms.

Unlike other microservices, the Portfolio Creator microservice does not have its own database. Instead, the Model component of the MVC pattern is supplied by retrieving data from external sources, specifically from the Public Microservice. The controller part of the MVC architecture is provided by the Portfolio Creator Module, which is responsible for handling complex computations and data manipulation using algorithms.

The Portfolio Creator Module consists of five algorithms, each generating a different investment portfolio based on a specific investing strategy. These strategies include:

- Index fund S&P 500 investing: This strategy involves creating a portfolio that invests in the S&P 500 index fund, using either a weighted or equal distribution approach.
- Quantitative momentum strategy: This strategy utilises quantitative analysis to identify stocks with positive momentum and includes them in the portfolio.
- Quantitative value strategy: This strategy focuses on identifying undervalued stocks based on quantitative analysis of their financial data.
- Quantitative momentum and value strategy: This strategy combines the momentum and value strategies mentioned above, seeking to leverage both factors in the portfolio construction process.

The logic of these algorithms follows a common pattern. They start by gathering relevant data based on the chosen strategy. This data is used to create a data frame containing information for all five hundred companies in the S&P index. Each company is then assigned a score based on their financial data. The algorithm extracts the top-performing companies based on their scores and distributes the portfolio's value among them.[12]

2.3. Front-end implementation

2.3.1. Architecture and Design

The front-end application's architecture uses a component-based design approach, which is known for its many advantages such as reusability, modularity, and maintainability. The development process is made more effective and codebase administration is made easier by dividing the application into smaller, independent components, each in charge of a certain functionality or aesthetic element.

The architecture of the application further extends to the concept of scenes, where components are combined to form complete pages or views (Fig 31). These scenes encapsulate the specific

[12] Subchapter 3.2 provides comprehensive information on each strategy's logic and methodology for a more in-depth understanding of these investment strategies and their implementation details.

functionality and layout required for each page, providing a clear separation of concerns, and promoting code organisation.

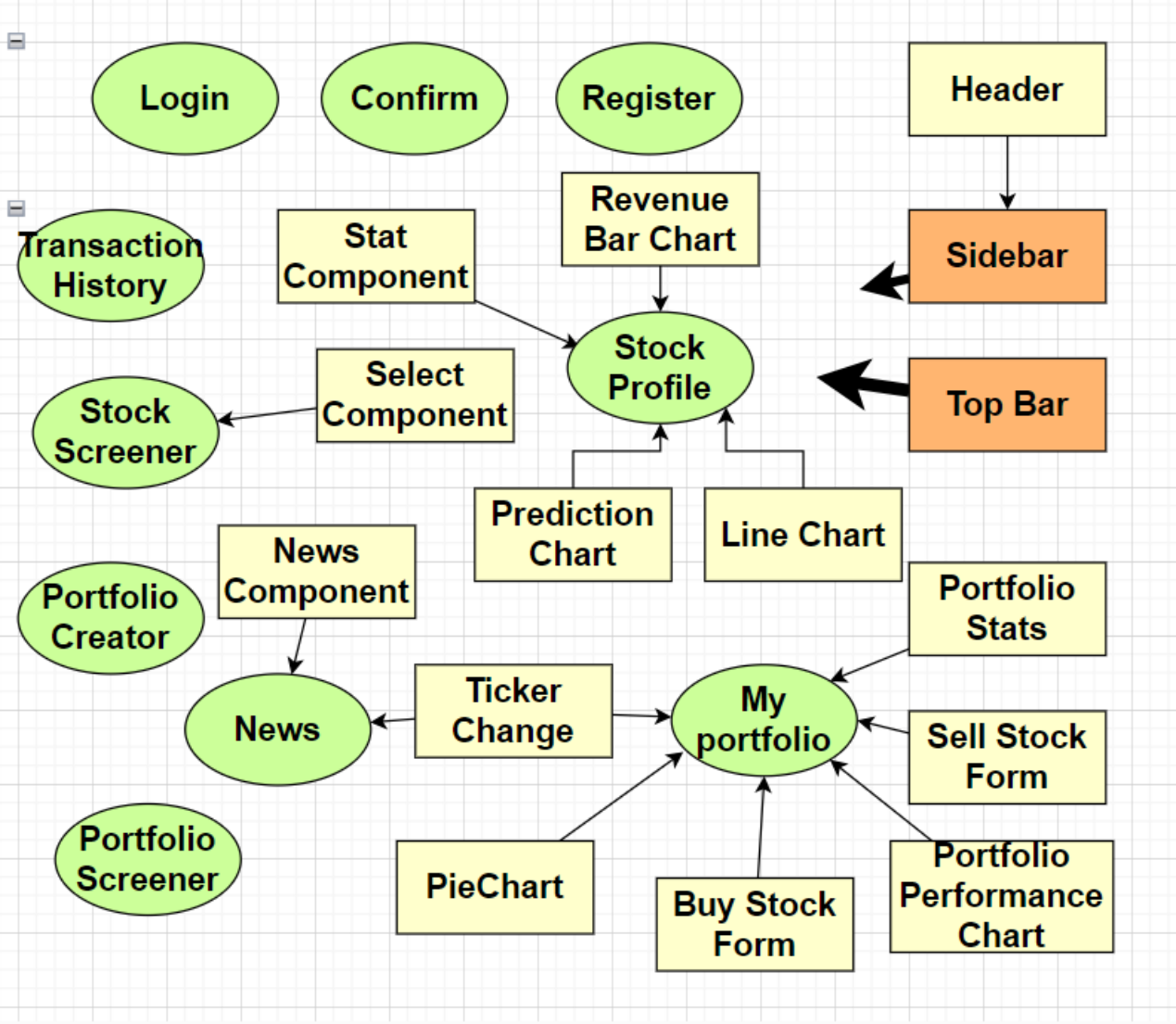


Figure 31 – Frontend component-scene diagram (scenes are green and components are yellowh)

2.3.2. User Interface

The user interface (UI) of a web application refers to the visual elements that users interact with. It plays a crucial role in improving user experience and facilitating user interactions. A well-designed UI engages users, improves usability, and helps them navigate the application efficiently. It also focuses on visual appeal, accessibility, branding consistency, providing feedback, and guiding users through the system. A good UI contributes to user satisfaction, retention, and positive perception of the application.

To achieve a good user interface (UI) experience, PortfoPal uses Material-UI which is a UI framework for React that implements Google's Material Design principles. It offers a vast gamma of tools and elements that are highly customisable, such as forms, different buttons, tables, icons, typographical units and other. Along with this UI-arsenal, PortfoPal implements a list of strategies to enhance the UI experience:

- Interaction and Feedback: PortfoPal ensures user-friendly interactions by providing meaningful feedback. For instance, when users fill out the registration form with incorrect input, the application promptly responds with error messages and provides guidance on how to rectify the issues. Similarly, for critical actions like portfolio reset, PortfoPal presents a confirmation form to ensure that users are certain about their decision. Loading screens are also implemented for components or pages that are not yet available, indicating progress and setting appropriate user expectations.

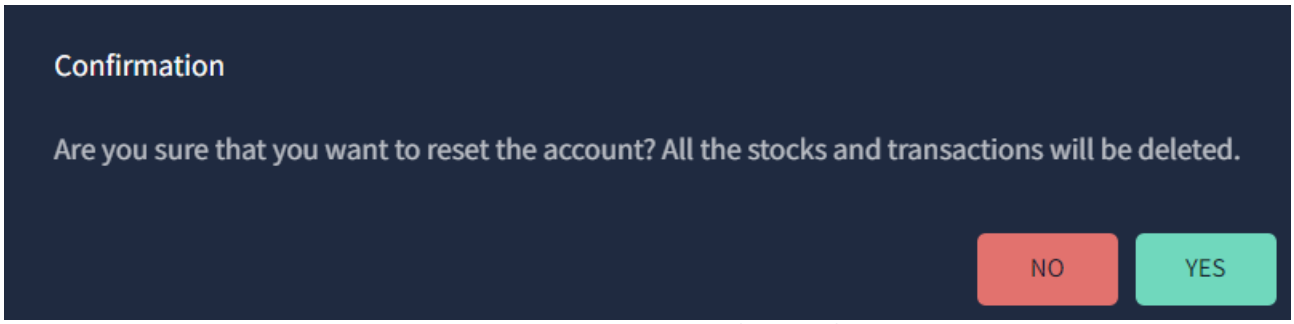


Figure 32 – Reset Account confirmation form.

- Visual Design Elements: Consistency and aesthetic appeal are key considerations in PortfoPal's UI design. The application employs a single font style, "Source Sans Pro," throughout its interface, ensuring uniformity in typography. Additionally, a carefully chosen colour theme is applied across the application, contributing to a visually pleasing and harmonious design across all pages. The use of icons is also prevalent throughout the application, providing visual representations and enhancing user understanding and engagement.
- User Engagement: PortfoPal employs various visual elements, such as charts and indicators, to actively engage users. These elements are strategically placed on almost every page, providing valuable information and insights in an easily digestible format. By incorporating dynamic visual representations, PortfoPal aims to captivate users and encourage active participation with the application.

- Usability and Efficiency: The design of PortfoPal's pages prioritises usability and efficiency. Clear and intuitive page and functionality names are used, ensuring users can easily navigate and understand the purpose of each section. The content within the application is well-organised, allowing users to locate information quickly and efficiently. Additionally, PortfoPal incorporates an easy-to-navigate menu, providing users with seamless access to different sections and functionalities, further enhancing the overall usability of the application. This menu in made in form of a Sidebar component that is present on absolutely all pages of the application (Fig 34). Besides a list of available for navigation pages, all the options are categorised in one of the three categories: data, pages, and charts.

By focusing on interaction, visual design, user engagement, and usability, PortfoPal aims to provide a user interface that prioritises usability, efficiency, and user satisfaction, becoming an intuitive and efficient platform for managing investment portfolios.

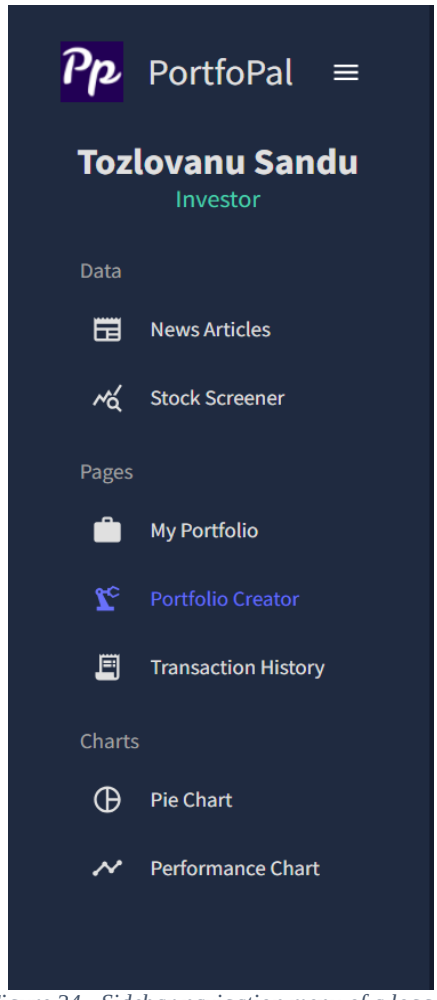


Figure 34 - Sidebar navigation menu of a logged user

2.3.3. Interaction and User Experience

Interaction and user experience (UX) play an important role in the context of front-end development. They directly impact the way users perceive and engage with a website or application, ultimately influencing their overall satisfaction with the application.

Several actions have been taken to enhance the user experience:

- Performance optimisation - To reduce maximum loading times and delays, PortfoPal uses effective caching techniques and effective algorithms.
- To provide the user with natural and efficient interactions, recurrent design solutions called interaction patterns have been implemented. These remedies consist of:
    1. Hover – In my Portfolio page you can hove over the pie chart to find out how much value in dollars each stock has.
    2. Navigation – Intuitive pagination features for stock screener page, news page and portfolio screener page.
    3. Form Input – Dropdown selections and text input on the stock screener page.
    4. Click/Tap – Stocks price change components navigate the user to the stock's profile page.

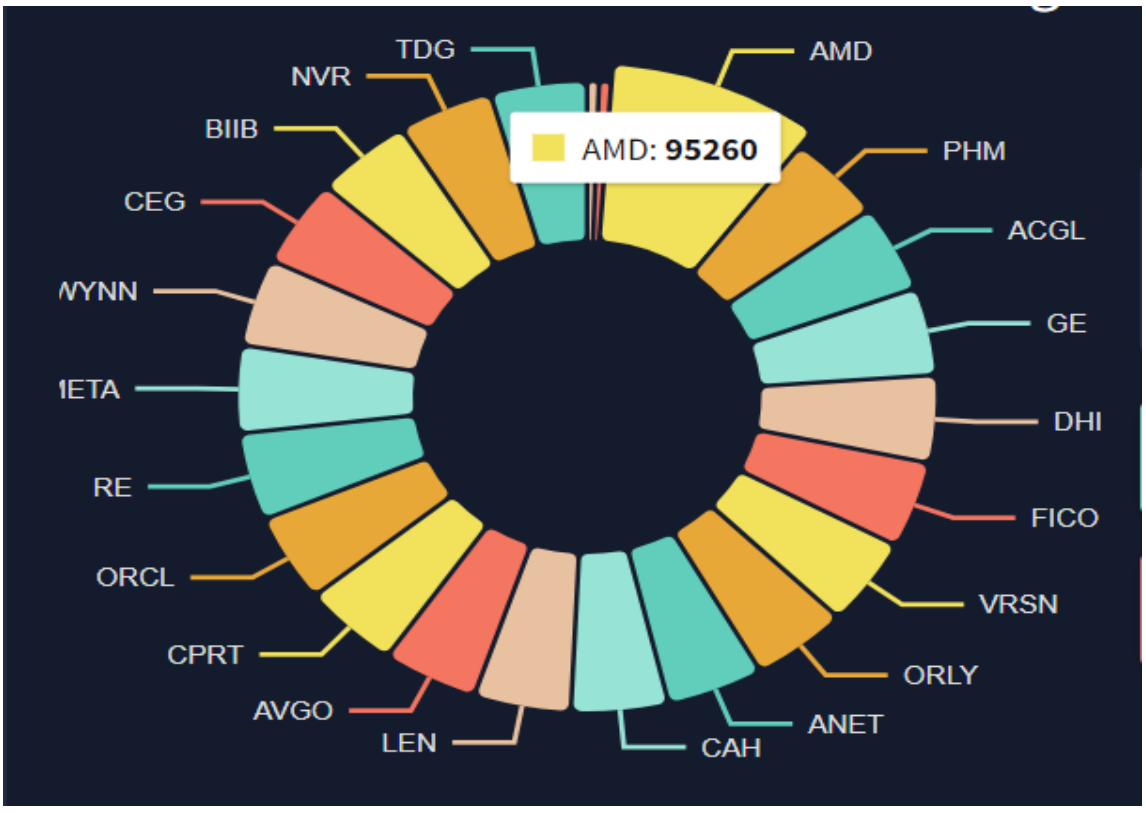


Figure 35 – My Portfolio Pie Chart hover

2.3.4. Data Presentation

Data presentation plays a crucial role in the development of the PortfoPal application. The platform aims to simplify complex financial data by presenting it in visually appealing graphs and charts. To achieve this goal, two graphical libraries have been used:

- The first library is Plotly.js, a JavaScript library known for its high customisability and interactivity. It is particularly suitable for handling large amounts of data, making it ideal for creating charts such as the Stock Price chart, Stock Price Prediction chart, and Portfolio Value Performance chart. Plotly.js supports Big Data and can be used with various programming languages, including Python. This allows the backend to generate the charts and send them to the front-end in an efficient manner using JSON format.
- The second library used is Nivo, a data visualisation library for React. Nivo is known for its ease of use and provides advanced visual effects like animations, tooltips, and hovers. While it excels at creating stunning visualisations for smaller datasets, it may not be as suitable for handling Big Data representations efficiently. Nivo has been employed for creating charts such as the Stock Revenue Chart and Portfolio Holdings Pie Chart.

In addition to charts and graphs, the MUI Data Grid Component is utilised for displaying tables within the application. This component is responsible for tables like the Stock Screening table, Transaction History table, and Portfolio Screening table. The MUI Data Grid Component enables the creation of responsive and interactive tables with built-in pagination functionality. It also offers filtering capabilities, allowing users to quickly find the desired data.

Overall, the combination of Plotly.js, Nivo, and the MUI Data Grid Component ensures that the PortfoPal application presents financial data in a visually appealing and user-friendly manner, catering to both novice and experienced investors.

2.4. Data management

2.4.1 Data collection

Real-time and current data are now essential in the quickly developing world of web apps for giving users a comprehensive and dynamic experience. To ensure the availability of such data, a reliable and constantly updated database is essential. Both the Private and Public microservices incorporate data collecting modules tasked with regularly updating the database using information

obtained from external APIs. These modules are known as the Data Frame Collector for the private microservice and the Data Collector for the public microservice.

In the context of the public microservice, Data Collector module functions independently on a separate thread and has the task of continuously gathering data from the financialmodelingprep[13] API. The financialmodelingprep API is used as a financial data provider, offering a wide range of financial information and calculation capabilities. It was selected for its thorough coverage of stock-related topics, including stock data, stock market data, economic data, and recent financial stocks. Another reason why financialmodelingprep.com was chosen is due to its ability to provide extensive information about many stocks, allowing PortfoPal to access and use relevant data for its functionalities. This API is a valuable resource for the application to access up-to-date financial data.

It is important to note that PortfoPal offers a paid plan, which grants users access to an API key. This key is used to authenticate and authorise API calls to financialmodelingprep.com. The API key provided with the paid plan has an API rate limit of three hundred calls per minute. Considering the current context and requirements, this rate limit is deemed sufficient to meet the data collection needs of the application.

As architecture, Data Collector implements the Command Design Pattern[14]. Two classes, namely APIInvoker and Command, work together to achieve successful data collection (Fig 36).

1. The Command class serves as an abstract base class and contains several important members. The "interval" member specifies how often the command should be executed(for example, once a minute, once a day, or once a week). The "priority" member is an identifier that determines the execution order of specific commands. Additionally, the "rate cost" member indicates the API consumption or cost associated with executing the command. This member is crucial in limiting the system's API usage within the permitted limits to avoid potential API key bans imposed by the financial data provider.
2. The “APIInvoker” class is responsible for registering commands and executing them sequentially in accordance with the command intervals. It manages the API limits and prioritises commands accordingly.

[13] https://site.financialmodelingprep.com/

[14] “The Command pattern is a behavioural design pattern, in which an abstraction exists between an object that invokes a command, and the object that performs it.” - https://medium.com/design-patterns-in-python/command-design-pattern-in-python-2f15b09f3774

Each API endpoint can have a unique command implemented by using these classes. This involves receiving data from the API, extracting relevant information, processing it, and storing it in the database. The modular approach allows for efficient data collection and integration within the application's database, ensuring that users have access to the most up-to-date information.

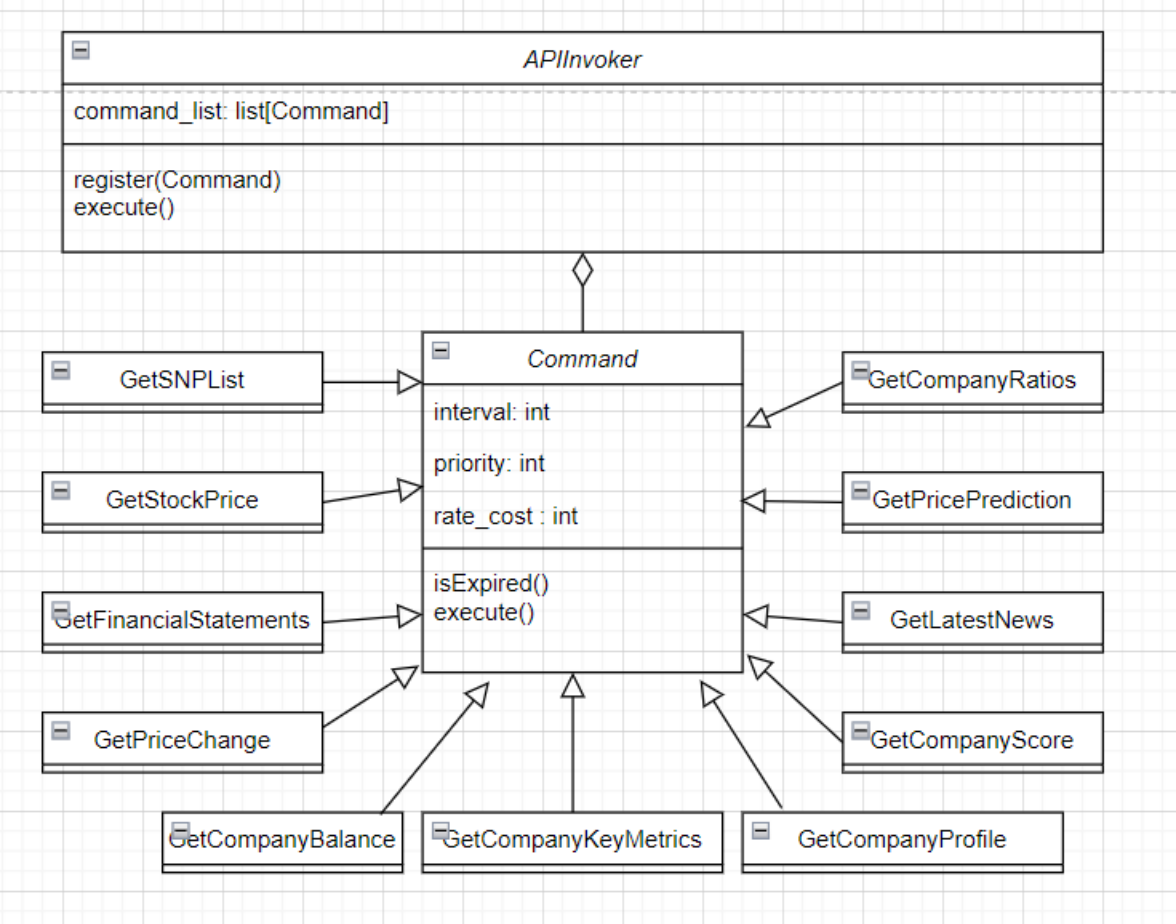


Figure 36 – Class diagram of the Data Collector Module

The Data Frame Collector module, which operates within the private microservice, is responsible for gathering stock price data for each trading day starting from 2022 up until the present day. This is accomplished through the utilisation of the “yfinance” Python library, which retrieves data from the Yahoo Finance API.

To collect the stock price updates, the Data Frame Collector module makes an API call every hour. This ensures that the module obtains the latest stock price information for each date and for every stock. The collected data is then used to update the application's database.

The module is named Data Frame Collector because the data provided by the “yfinance” library is structured as a pandas data frame. The obtained data frame is then converted to JSON format using the pandas “to_json” method before being stored in the database. This allows for efficient

storage and retrieval of the stock price data within the application, which is lately used for creating portfolio performance chart and stats.

2.4.2 Data storing and database structure

PortfoPal uses two databases, one in the public microservice and another in the private microservice. Both databases employ SQLalchemy as the Object-Relational Mapping (ORM) tool. SQLalchemy was chosen for its several advantages such as its object-oriented approach, which aligns naturally with the manipulation and creation of tables. This object-oriented paradigm allows for seamless integration between the database and the application's code, simplifying the development process. Secondly, SQLalchemy provides a simplified and intuitive interface for interacting with the database. With its Python-based interface, developers can easily manage and manipulate data using familiar programming techniques. This streamlined interaction enhances productivity and reduces the complexity associated with database operations. Another significant advantage of SQLalchemy is its reliable transaction management. Transactions enable the grouping of multiple operations into a single atomic unit. This feature ensures data consistency and integrity by either committing all changes or rolling back the entire transaction if an error occurs. In the context of this project, transaction management plays a vital role in data synchronisation among threads. While the Data Collector thread continually updates the database, the Flask application can safely manipulate the data without the risk of conflicts or inconsistent states.

In the context of PortfoPal's public microservice database, a logical organisation is implemented with separate tables corresponding to each financialmodelingprep API endpoint. The diagram below illustrates this structure (Fig 37), showcasing a total of nine tables, each capturing specific data fetched from the API.

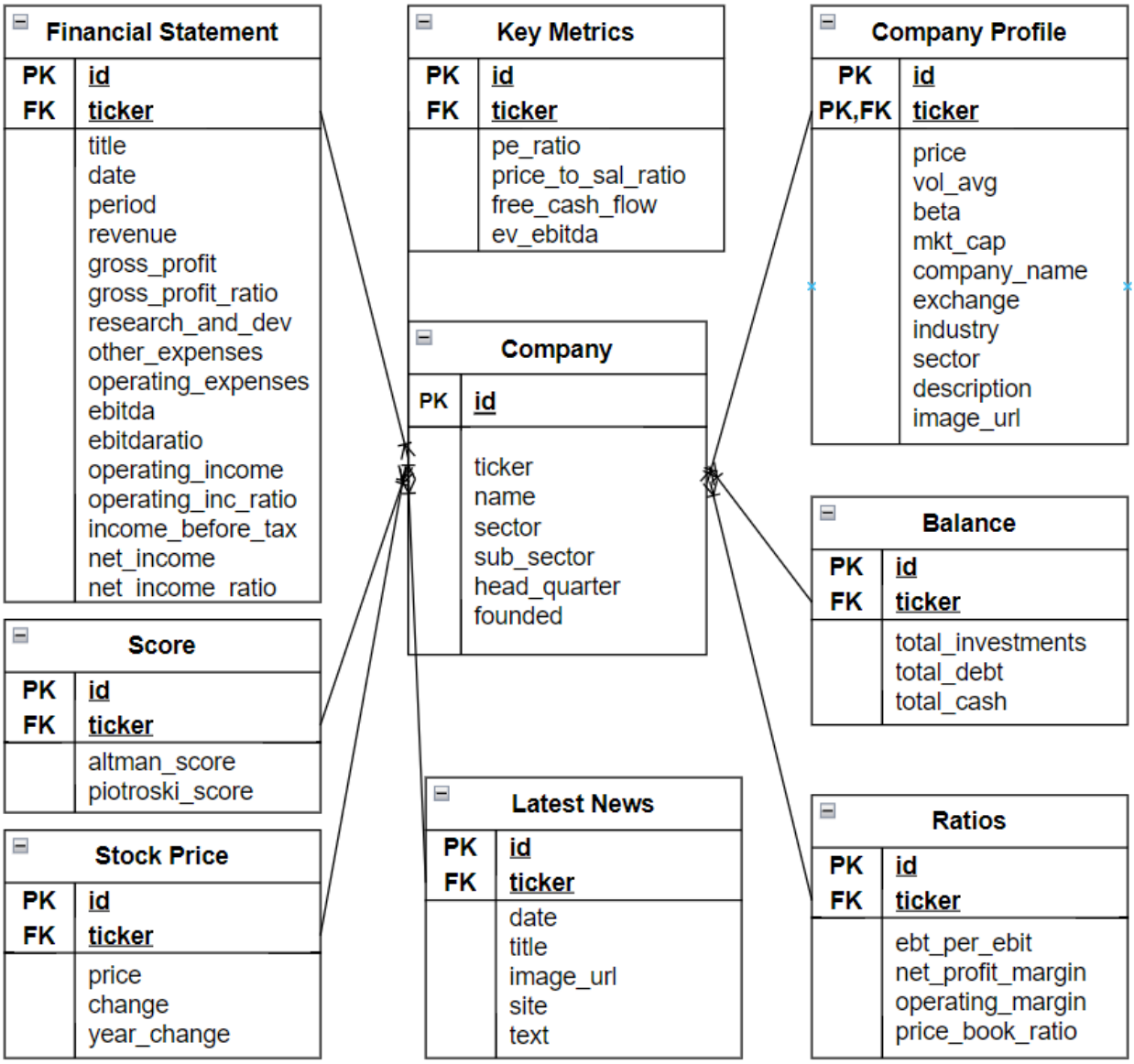


Figure 37 – Public microservice database diagram

Among these tables, the Company table stands as the central component of the entire database. It serves as the core entity, housing the most fundamental stock data. As the primary table, it establishes relationships with other tables, facilitating the retrieval and management of related data.

By structuring the database in this manner, the organisation aligns with the API endpoints, allowing for a clear and coherent representation of the data. Each table corresponds to a specific set of information obtained from the API, enabling efficient querying and manipulation based on distinct data categories.

This logical division of tables based on API endpoints promotes modularity and facilitates the management of data. It provides a structured foundation that allows for focused data access and retrieval, enhancing the overall functionality and effectiveness of the system.

Within PortfoPal's private microservice database, sensitive and confidential user information is stored. This database encompasses crucial tables such as user portfolios, transactions, stock holdings, and user credentials, ensuring comprehensive data management (Fig 38).

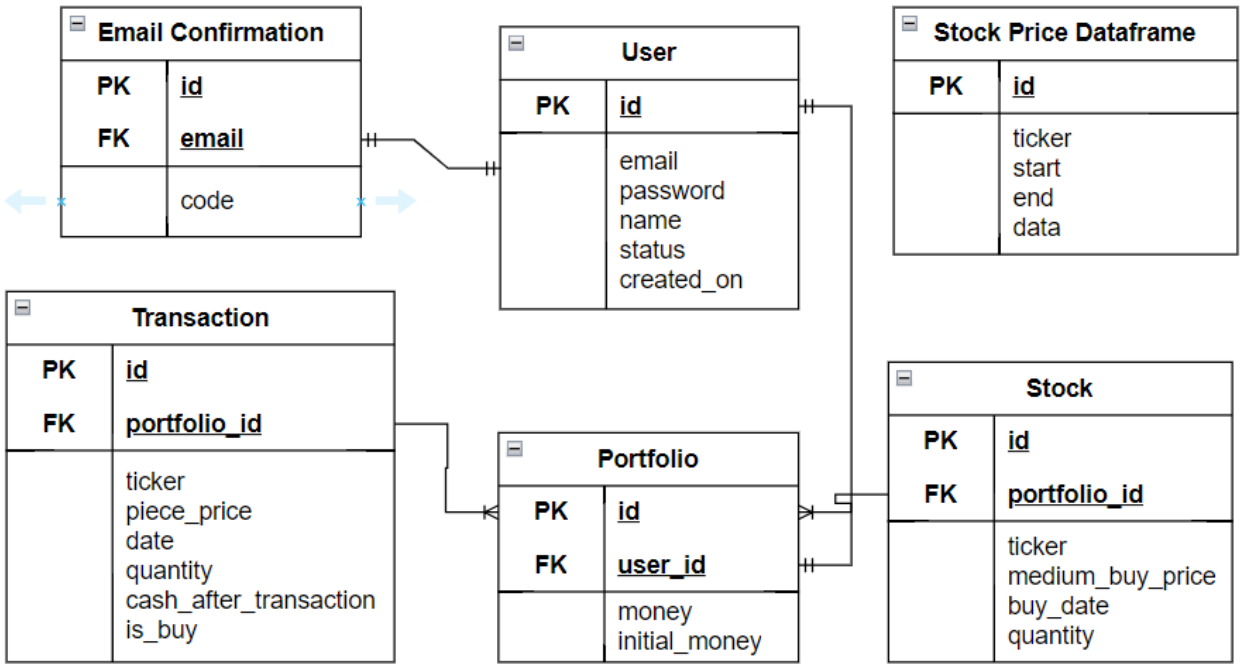


Figure 38 – Private microservice database diagram

The Stock Price Data frame table plays a significant role in this database, holding a wealth of information regarding stock prices over time. By storing this data in the form of data frames, it enables efficient analysis and manipulation of stock price information within the system.

The Portfolio table serves as a pivotal component, capturing essential details about each user's portfolio. It includes the unique user ID, the current cash amount within the portfolio, and the initial cash amount when the portfolio was created. This table acts as a central reference point for tracking and managing individual portfolios within the system.

The Stock table complements the Portfolio table by storing information about individual stocks. It records details such as the stock ticker, the date of purchase, the quantity of shares held, and the average price at which these shares were acquired. This table facilitates the tracking and management of individual stock holdings for each user, forming a crucial aspect of their portfolio.

Ultimately the User and Email confirmation tables, which hold the main user credentials and the email confirmation code respectively, being used for logging, registering and email confirmation functionalities.

2.5 Security and Privacy

Security and privacy are of utmost importance to PortfoPal, and the platform implements several measures to ensure user privacy, system security, and protection against breaches. These measures include:

- Authentication and Authorisation: PortfoPal enforces strong user authentication and authorisation procedures that users are required go through, including email confirmation. This procedure secures the application's user data from unauthorized access. JSON Web Tokens (JWT) implementation also makes sure that users have the proper access privileges to the various services, maintaining the system's integrity generally.
- Protection against Web Scraping, Brute Force, and DoS Attacks: PortfoPal incorporates safeguards to protect against common malicious activities such as web scraping, brute force attacks, and Denial of Service[15] (DoS) attacks. All API endpoints are equipped with a well-defined API rate limit per IP, which helps prevent excessive requests from a single source and mitigates the risk of these attacks.
- XSS[16] and SQL Injection [17]Protection: To mitigate the risks associated with Cross-Site Scripting (XSS) attacks, PortfoPal includes a content security policy (CSP) header in all API responses. The probability of XSS vulnerabilities is reduced by this header, which sets standards and limitations on the types of content that may be loaded. Additionally, using parameterised queries and prepared statements, SQLAlchemy's Object-Relational Mapping (ORM) tool offers built-in defence against SQL injection threats.

By putting these security measures in place, PortfoPal proactively protects user privacy and guarantees the system's general security. Strong authentication, rate limiting, content security policies,

[15] A denial-of-service (DoS) attack occurs when legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor - https://www.cisa.gov/news-events/news/understanding-denial-service-attacks.

[16] Cross site scripting (XSS) is a type of attack in which the attacker injects malicious scripts into web-pages belonging to legitimate web-sites. - https://www.enisa.europa.eu/topics/incident-response/glossary/cross-site-scripting-xss

[17] SQL injection attacks attempt to exploit web application vulnerabilities by concatenating user input with SQL queries. - https://www.ibm.com/docs/en/guardium/11.2?topic=analytics-characteristics-sql-injection-attack

and SQL injection prevention mechanisms work together to bolster the platform's security posture and provide protection against typical web vulnerabilities.

Chapter 3: Use cases

In this last chapter we first see our app as an investment education platform and how it helps people learn about the stock market and initializes them into the world of investing. Secondly, we experiment with four strategies and see how each of them influences the user's portfolio, in order to learn and see in what circumstances each strategy is best to be used.

3.1. Investment education platform

In this section of the thesis, we will examine PortfoPal from the perspective of an investment education platform. Given the current political, social, and financial environment, investment education has gained significant importance. The volatile and uncertain nature of markets, driven by various economic and geopolitical factors, highlights the need for individuals to have a solid understanding of investment strategies and principles. These principles and strategies are not the easiest to handle, but PortfoPal makes its impact into promoting these by providing the required tools for the learners.

3.1.1. Fundamental analysis promotion

In the investment landscape, where various strategies and types of analysis exist, PortfoPal promotes a particular approach that focuses on assessing key factors influencing a company's performance and growth - Fundamental analysis. This methodology involves evaluating a company's intrinsic value by examining its financial statements, competitive position, and overall industry trends. Fundamental analysis emphasises long-term investments as it assesses the company's financial health and growth prospects, thus mitigating the impact of short-term market volatility and considering long-term trends.

PortfoPal actively encourages and supports this investment approach by providing users with a range of tools and resources. The platform offers comprehensive company profiles and reports that facilitate in-depth analysis. Users can access screening and analysis tools that help them identify companies that align with their fundamental analysis criteria. Furthermore, PortfoPal offers a vast amount of data, enabling users to conduct thorough research and perform detailed fundamental analysis on companies of interest.

By promoting fundamental analysis, PortfoPal empowers users to make informed investment decisions based on a deep understanding of a company's financial position, competitive advantages,

and industry outlook. The platform equips users with the necessary tools to conduct research, analyse data, and gain insights into companies, facilitating the application of fundamental analysis principles.

By emphasising fundamental analysis and providing users with company profiles, reports, screening and analysis tools, and extensive data, PortfoPal enables users to effectively apply this investment strategy. This approach encourages a thorough examination of a company's fundamental factors, empowering users to make investment decisions aligned with their long-term goals and enhancing their overall investment success.

3.1.2. Exploration of diverse investment strategies

In addition to promoting fundamental analysis, PortfoPal offers users the opportunity to explore different investment strategies that have been proven to be effective. The platform provides explanations and visual representations of portfolios created using these strategies, allowing users to gain insights into their workings.

PortfoPal presents five distinct investment strategies that enable users to observe their performance over time. By providing a risk-free environment, users can explore these strategies without any financial consequences. This feature allows individuals to delve into the portfolios generated by these strategies and gain a deeper understanding of their composition and underlying principles.

Through these strategies, PortfoPal aims to introduce users to basic concepts of financial investing. By offering explanations and visual representations, the platform simplifies complex investment strategies, making them more accessible to individuals who may be new to the field.

This approach enables users to observe the performance and composition of portfolios created using these investment strategies. Users can study how the strategies evolve over time and gain valuable insights into their potential effectiveness. By offering this educational opportunity, PortfoPal equips users with knowledge and a hands-on experience that can serve as a foundation for their own investment decisions.

In this way, PortfoPal’s provision of investment strategies, accompanied by explanations and visual representations of portfolios, empowers users to explore and comprehend the fundamental concepts of financial investing. It serves as a valuable educational resource, enabling individuals to gain insights into proven strategies and develop their investment knowledge and skills.

3.2. Investment portfolio experimenting

In this section of the last chapter, practical examples will be presented to showcase the performance of the five investment strategies on real portfolios. The focus will be on demonstrating how these strategies perform over time and providing insights into the process of creating and analysing these portfolios using the algorithms.

The examples will shed light on how each investment strategy constructs portfolios and the underlying principles employed. The performance of these portfolios will be assessed across different time periods, allowing for a comprehensive evaluation of their effectiveness.

By conducting experiments and analysing the results, we will make several conclusions on why these strategies performed in this way and analyse the overall stock market context of the experiment.

3.2.1. Index fund S&P 500 investing

Index fund investing has long been regarded as one of the most reliable investment strategies. Index funds offer portfolio diversification, effectively spreading risks across various companies and industries. Over an extended period, index funds have proven to deliver stable returns, primarily due to their broad exposure to the growth of the overall economy. Recognising the benefits of this strategy, PortfoPal has developed its own index fund designed to track the S&P 500 index. This is achieved through two approaches: the Weighted S&P 500 index fund and the Equal-weight S&P 500 index fund.

Creating the Equal-weight S&P 500 index fund required obtaining the price per share for all five hundred stocks in the S&P 500 index. This strategy's algorithm ensures that each stock receives an equal weighting in the portfolio, with each stock receiving 0.2% of the portfolio's overall value. To illustrate this, a simulation was conducted in PortfoPal, starting with an initial cash value of $1 million on June 13, 2022. It is important to note that the portfolio is not adjusted to accommodate new companies entering the S&P index.

In the chart below, the performance of the simulated portfolio is depicted. Over the course of one year, the portfolio achieved a growth of 11.21%. While it slightly trailed the S&P 500 index benchmark by 2.25%, it demonstrates a commendable performance that closely tracks the overall market.

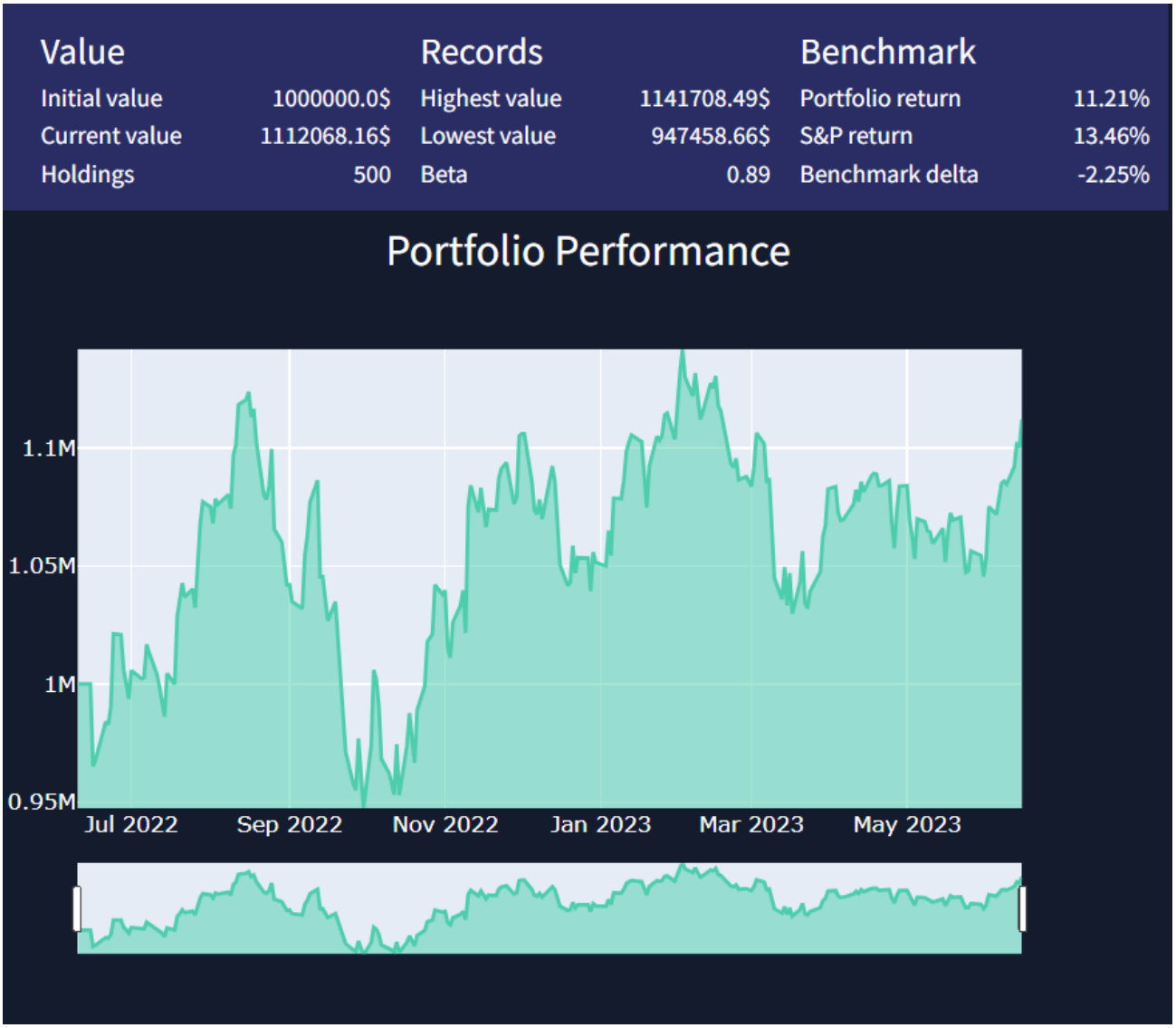


Figure 39 – Equal-weighted S&P 500 index fund strategy generated portfolio performance chart and stats

Alongside the Equal-weight S&P 500 index fund, PortfoPal also offers the Weighted S&P 500 index fund, which follows a similar strategy but with one key distinction. In this situation, the shares are allocated based on market capitalization. As a result, larger market capitalization companies, like Apple, will receive a larger proportion in the portfolio than the smaller businesses will. In the chart below is displayed a portfolio worth $1 million, created using the Weighted S&P 500 index fund. Is remarkable the portfolio's performance over the course of one year, showing a tremendous growth of 23.09%. This approach has produced outstanding returns, outperforming the performance of the S&P 500 index by more than 9.5%. It is important to note that, unlike the S&P 500 index, the experiment's portfolio was not modified to account for changes in market capitalization.

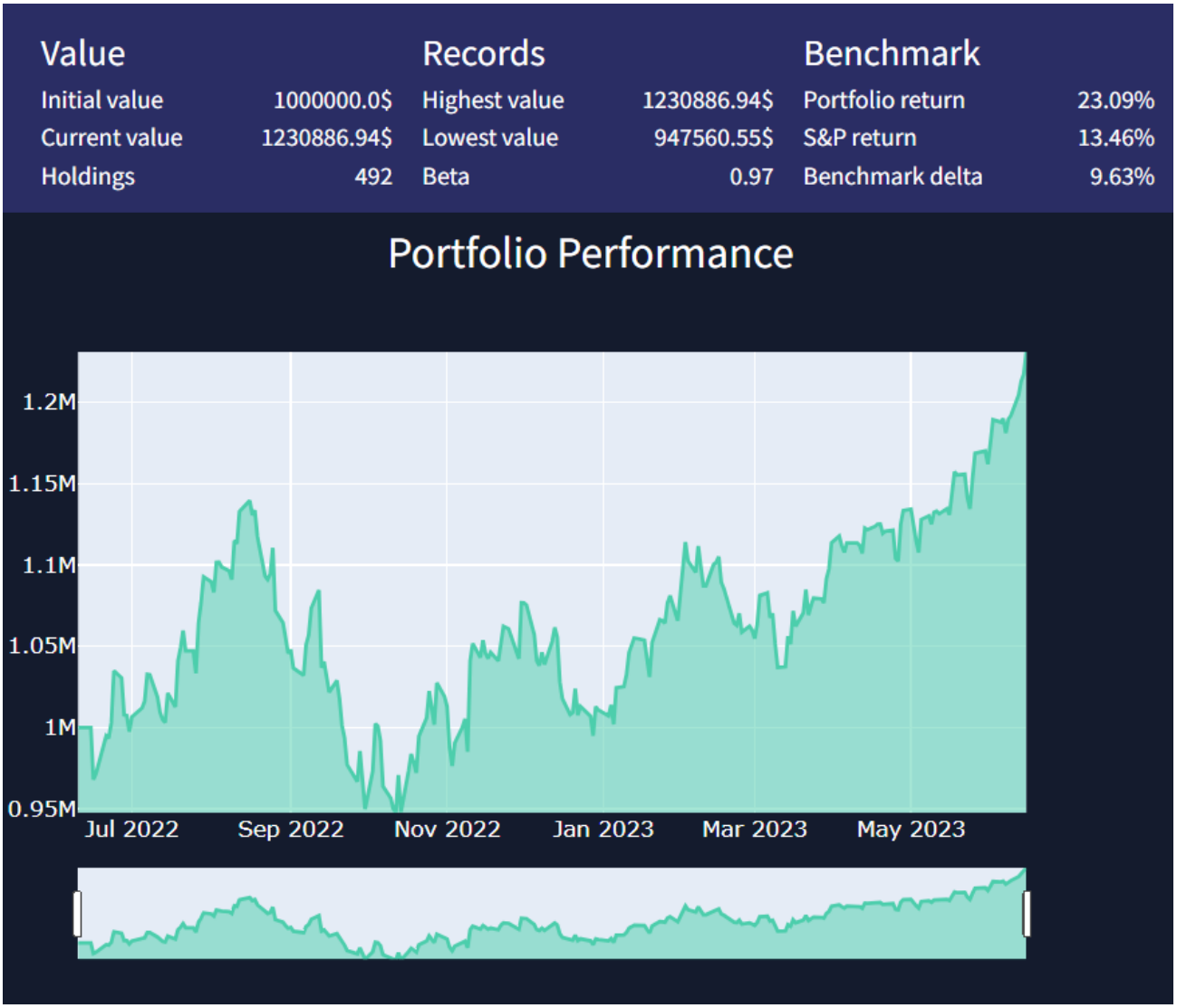


Figure 40 – Weighted S&P 500 index fund strategy generated portfolio performance chart and stats

The results of this experiment highlight a significant concentration of stock value among a select few companies. This concentration indicates that a small number of companies hold most of the market capitalisation, essentially monopolising the market. This phenomenon can have implications for portfolio diversification and risk management. Additionally, it is worth mentioning that by providing the Weighted and Equal-Weighted S&P 500 index fund, PortfoPal allows users to explore different weighting methodologies and observe the impact of market capitalisation on portfolio performance. This strategy offers insights into the market dynamics and the influence of individual companies with larger market capitalisations on overall portfolio returns.

3.2.2. Quantitative momentum strategy

Quantitative Momentum Investing is a widely embraced investment strategy that involves analysing large amounts of historical financial data, typically focusing on a company's stock prices, in order to project its future trend trajectory. The underlying concept is simple: if a stock has demonstrated strong returns over a certain period, it is likely to continue performing well in the subsequent years, and vice versa. PortfoPal leverages this strategy by collecting the price change data from the previous year and constructing a portfolio consisting exclusively of the twenty stocks with the highest momentum.

In the graph below, the attributes of this portfolio created using the Quantitative Momentum strategy are presented. It is worth noting that the portfolio experienced a 6.27% drop in value, while the S&P index grew by 13.46%, indicating a poor result. Upon examining the composition of the portfolio, it becomes evident that fifteen out of the twenty stocks belong to the Energy and Oil sector. This sector witnessed a substantial increase in stock values during the period of 2021-2022, driven by rising oil prices. However, when the oil prices subsequently dropped, the stock prices of these energy companies also experienced a significant decline.

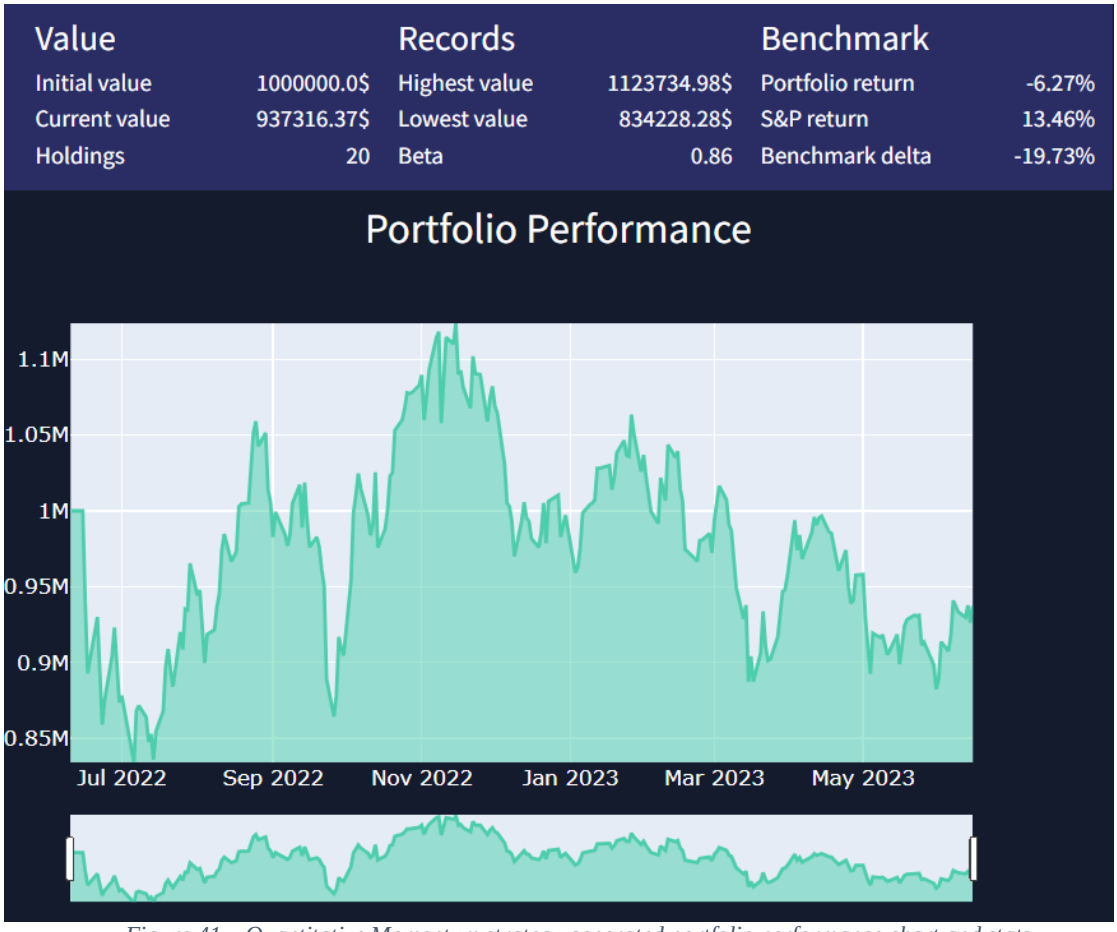


Figure 41 – Quantitative Momentum strategy generated portfolio performance chart and stats

This portfolio serves as a powerful illustration of the importance of diversification. It highlights the significance of diversifying not only across a range of companies but also across various sectors. By incorporating diversity into a portfolio, investors can mitigate the risks associated with concentrated exposure to a single sector or industry.

The result of this experiment emphasizes how important it is to take portfolio variety and the potential effects of sector-specific trends into account. While the Quantitative Momentum method may have had success in the past, it is important to realize that performance in the future might be affected by a variety of variables, such as market dynamics and sector-specific dynamics. Engaging in thorough research, analysis, and maintaining a well-diversified portfolio are key aspects of successful investing.

PortfoPal's incorporation of the Quantitative Momentum strategy provides users with insights into this popular investment approach. By observing the performance and composition of portfolios constructed using this strategy, investors can gain valuable knowledge regarding the potential benefits and risks associated with momentum-based investing.

3.2.3. Quantitative value strategy

Quantitative Value investing operates within the same analysis category as the previous strategy discussed. However, it diverges in its focus on value investing principles, using different financial metrics, ratios, and historical financial data. PortfoPal implements this strategy by evaluating various financial parameters, including free cashflow yield, operating margin, P/E ratio, price-to-sales ratio, price-to-book ratio, and profit margin of each stock. The algorithm assigns a score to each of these financial characteristics based on the mean value across all stocks. The top twenty stocks with the highest scores are then selected to form the portfolio.

In the graph below, we can observe a portfolio created using the Quantitative Value investing strategy, employing the same parameters as before. Over the course of one year, the portfolio experienced a notable growth of 10.68%. While it slightly trailed behind the benchmark value by approximately 3%, the results can still be considered very positive. It is worth noting that this portfolio exhibits an intriguing composition, as it comprises stocks from various sectors and industries, indicating a high level of diversification.

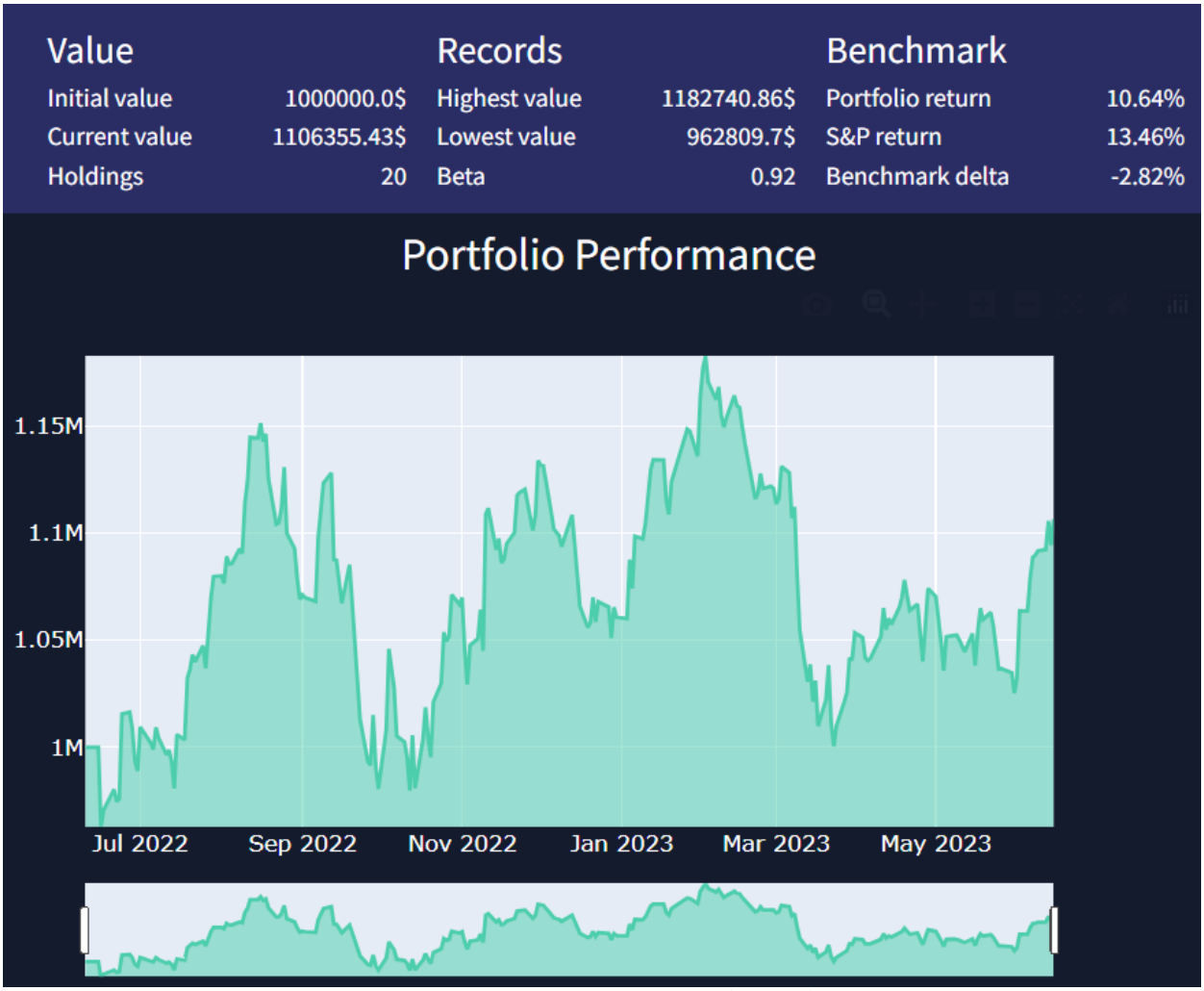


Figure 42 – Quantitative Value strategy generated portfolio performance chart and stats

The Quantitative Value investing strategy is particularly favoured by investors who rely on fundamental analysis and seek companies with strong financial health that are often undervalued in the market. By incorporating financial metrics and ratios, this approach aims to identify stocks that are potentially undervalued and may offer attractive investment opportunities.

The success of this strategy, as demonstrated by the performance of the portfolio, reinforces the importance of conducting thorough fundamental analysis and considering a company's financial health when making investment decisions. By selecting stocks with favourable financial characteristics, investors aim to capitalise on potential market inefficiencies and generate attractive returns over time.

3.2.4. Quantitative value & momentum strategy

Quantitative Value & Momentum investing combines both the concepts of stock momentum and fundamental financial data analysis into a single strategy. This approach serves as a powerful tool

for investors who aim to follow existing trends while selecting stocks that exhibit strong financial health to support sustained growth. PortfoPal applies this strategy by conducting the necessary computations to generate value and momentum scores, as explained in the previous subchapters. These scores are then combined to produce a final score, which is used to identify the top 20 companies that are to be included in the portfolio.

In the chart below, we can notice the performance of a portfolio created using the Quantitative Value & Momentum investing strategy over the course of one year. The portfolio experienced a growth of 3.77%, falling slightly behind the benchmark by nearly 10%.

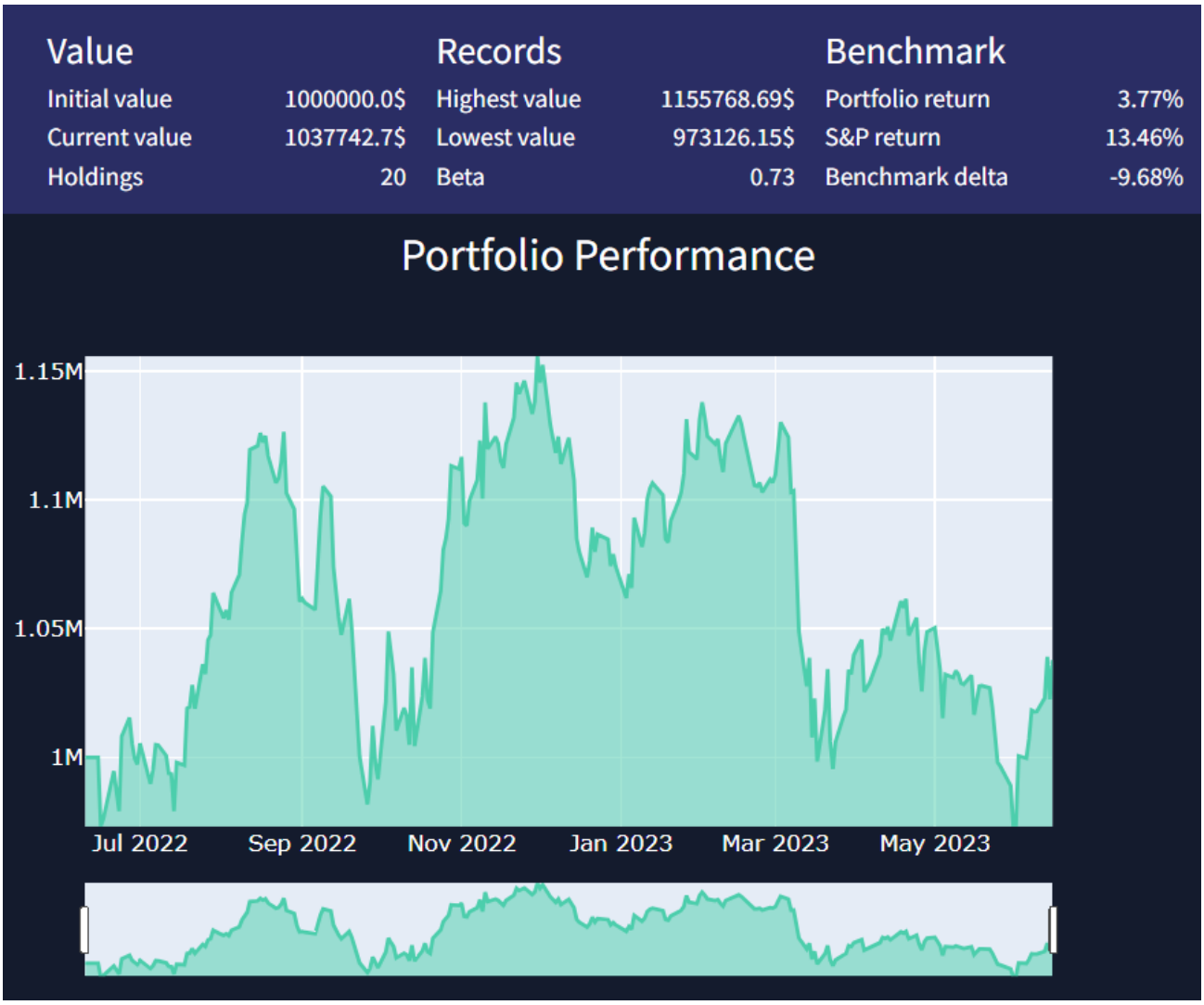


Figure 43 – Quantitative momentum & value strategy generated portfolio performance chart and stats

It is worth noting that despite incorporating momentum analysis, the portfolio contains a minimal number of Energy and Oil sector companies. This suggests that the fundamentals of these energy companies may not be as robust as those found within the portfolio. From a personal perspective, this strategy is highly regarded due to its focus on real trends and reliance on sound

financial data. It avoids speculative bubbles and prioritises companies with healthy financial indicators.

Additionally, in the graph below, we present a portfolio generated with the same strategy approximately two months ago. It is interesting to observe an 8.6% growth, surpassing the benchmark by 1.28%. This highlights the significant impact of market context on investment strategies. Investing during a bullish[18] or bearish[19] market can yield vastly different results for a given strategy.

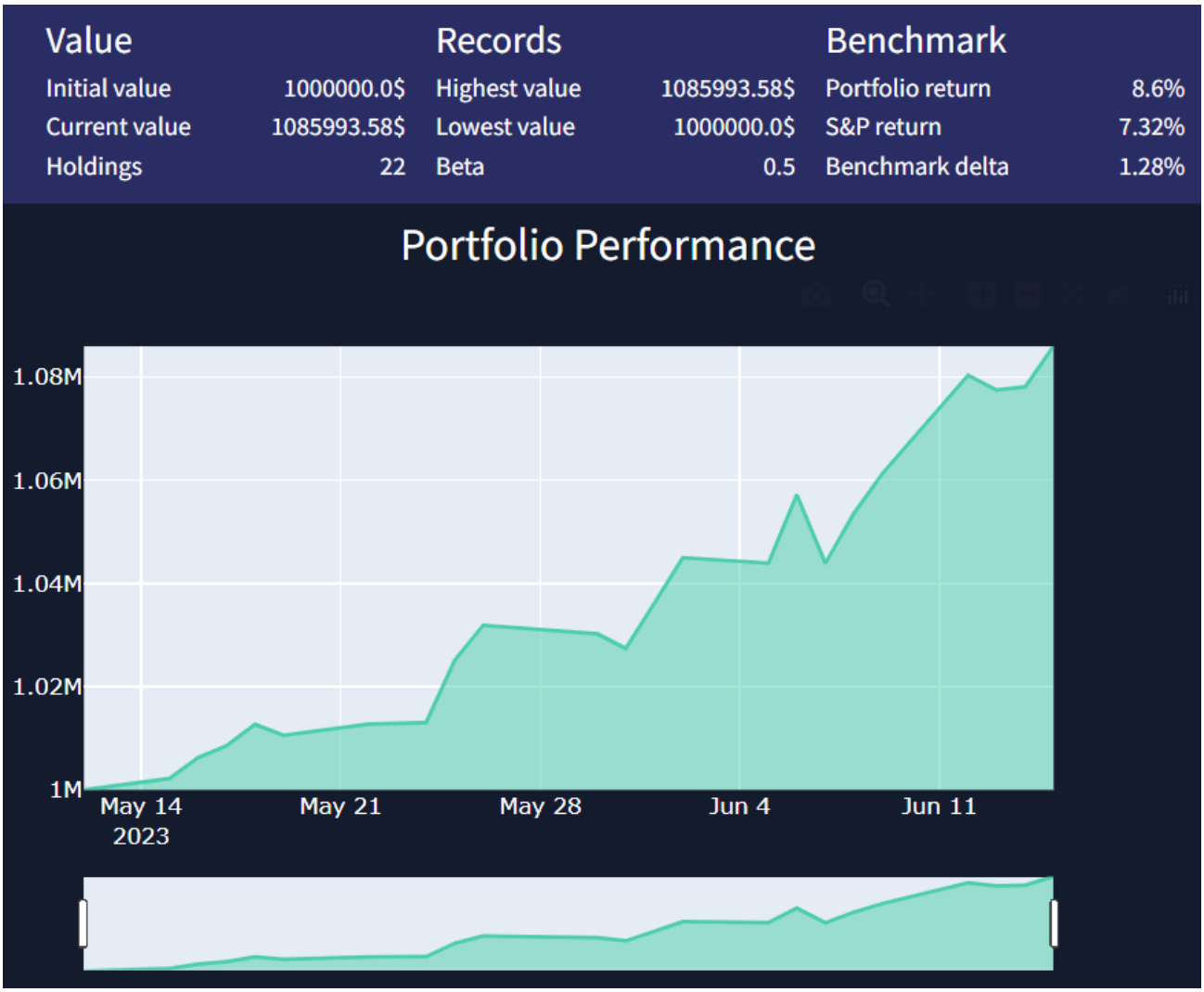


Figure 44 – Quantitative momentum & value generated portfolio performance chart for 2 months

[18] Bull market – "the condition of a financial market in which prices are rising or are expected to rise" https://www.investopedia.com/terms/b/bullmarket.asp

[19] Bear market – "The opposite of a bull market is a bear market, which is characterised by falling prices and typically shrouded in pessimism." https://www.investopedia.com/terms/b/bullmarket.asp

The context in which investments are made plays a crucial role in determining the success of a chosen strategy. By combining both value and momentum analysis, the Quantitative Value & Momentum strategy aims to capture the advantages of following trends while considering the financial health of selected companies. PortfoPal provides users with the opportunity to explore this strategy and observe its performance over time. By studying the generated portfolios and analysing their results, investors can gain valuable insights into the effectiveness of this combined approach and make informed investment decisions based on real-world trends and solid financial fundamentals.

Conclusions and future prospects

Throughout the project, we have observed the benefits of investing and how the goal of reaching financial independence could be achieved if we created and developed this habit, thus explaining why this precise field of study was chosen. Motivated by personal interest, current lack of proper investment tools and the dream of long-term financial independence, the creation of Portfopal began, which evolved into a user-friendly investing tool that helps people with the stock market analysis and offers them a starting point for their own journey of investing. We have seen what similar apps the market offers and what strengths and weaknesses they have, preparing the background to explain what other contributions Portfopal brings, such as its modular architecture, the user-friendly nature of the algorithms and the personalized stock and news recommendations. We have explained in details all the important features of the app's architecture for both the back- and the front-end. Moreover, we discussed the security concerns and what methods were implemented in order to protect user data and app's integrity. Nonetheless, we have presented various use cases of this application and experimented to see how different strategies influence the portfolio's performance.

The biggest challenge encountered was the process of documentation about the stock market. In order to create a tool that would really help investors, there was the need to thoroughly search for relevant information and find the best way to capitalize all the information into a truly useful project that would help investors all the way from zero until financial independence. It required time and a long exercise of finding reliable sources, comparing answers and thinking about where would a specific information be best suitable. But the enthusiasm of being, myself, an investor helped overcome the challenge and the mentioned exercise was enjoyable rather than exhausting.

Another challenge was presenting the project in written form, with letters. After the code was completed and the idea got the shape of a website, the next step was explaining everything on paper and for a programmer this is the most difficult part. Choosing the best way to formulate a sentence in order to put the idea that made me so enthusiastic in the best light was a big challenge, but the solution was stepping out of my comfort zone and just do it. One page a day in the beginning, then two pages a day, then next thing I know I was doing it without thinking and created a skill that will definitely be useful for me in the future – presenting my ideas for others, explaining the code I wrote and putting it into words.

Overall, creating Portfopal was an unforgettable experience that taught me many things. Of course, there is always room for improvement, so here are several key areas that should be considered for further enhancements:

Back-end:

1)Splitting the private microservice into two separate microservices, which involves creating a proxy microservice that handles routing, caching, rate limiting, and load balancing, while the private microservice focuses on authorisation and private features like portfolio management. This separation can improve performance and maintainability.

2)Introducing an "Import Portfolio" functionality, feature that would allow users to import external portfolios from other applications by integrating them into PortfoPal. This would provide users with more flexibility and convenience in managing their investments.

3)Implementing a Machine Learning Module, allowing PortfoPal to offer advanced stock prediction techniques that are independent of the Prophet technology. This can enhance the accuracy of investment recommendations and provide users with valuable insights.

4)Improving overall response time through the optimisation of algorithms, efficient data querying, and storage techniques can significantly reduce response waiting times. This optimisation will improve user experience and ensure a smooth and efficient application performance.

5)Testing-including integration tests, unit tests, and load tests, is crucial to ensure the stability and reliability of the application. This will help identify and address any hidden bugs or issues that may impact the functionality of PortfoPal.

6)Expanding the range of investment strategies: Adding additional investment strategies, such as contrarian investing and growth investing, will provide users with a wider array of options and cater to different investment preferences and goals.

Front-end:

1)Improving responsiveness by adapting PortfoPal from a desktop web application to a mobile-friendly one will expand its potential user base. This involves UI refactoring and optimisations to ensure a seamless and intuitive experience for users accessing the application on smartphones and tablets.

2)Portfolio exports, which would enable users to export their portfolio performance in formats like CSV or Excel can facilitate data analysis and reporting, allowing for further insights and customisation.

3)Implementing front-end caching, which could enhance performance by reducing server requests and improving loading times. This can lead to a smoother and more responsive user experience.

To sum up, the stock market world is a huge world that is worth to be explored. I believe that Portfopal's main goal – to be truly helpful for investors even if they are beginners – is achieved, even though there are many things to still be improved. If there were more tools as Portfopal and with even better features, achieving financial independence would not be just a dream, but a reality come true for much more people.

Bibliography

[1] „The Intelligent Investor” by Benjamin Graham (2003)

[2] „The Little Book of Common Sense Investing” by John C. Bogle

[3] "The Investor's Manifesto: Preparing for Prosperity, Armageddon, and Everything in Between" by William J. Bernstein

[4] “6 Basic Financial Ratios and What They Reveal” - https://www.investopedia.com/financial-edge/0910/6-basic-financial-ratios-and-what-they-tell-you.aspx

[5] Financial modelling Prep API Documentation - https://site.financialmodelingprep.com/developer/docs/

[6] React framework official documentation - https://react.dev/

[7] Mui Components official documentation - https://mui.com/components/

[8] Flask framework official documentation - https://flask.palletsprojects.com/

[9] Plotly library official documentation - https://plotly.com/graphing-libraries/

[10] Nivo library official website - https://nivo.rocks/