Equites

A project delivered under the requirements of MIE479

Available at: http://ec2-35-183-96-226.ca-central-1.compute.amazonaws.com:5000/

Data Repository: https://github.com/lluo5779/Equites/tree/master/server/models/portfolio/data

Code Repository: https://github.com/lluo5779/Equites

Members:

Aidan Kehoe

Qingyang Li

Louis Luo

Mike Travis

Matthew Reiter

Instructors:

Prof. R. Kwon

G. Costa

Table of Contents

0. Introduction

“Equites” is a culmination of the engineering design requirements defined in the Capstone course MIE479. The project offers a scalable, Muli-Period Portfolio optimization engine, comparable to that of a “Robo-Advisor”. The delivery is framed to that of a professional quantitative asset management company and the design is meant to accommodate a range of users, from those who are new to investing to those who are financially literate.

Equites offers a baseline of three functions:

1) Track: Displays backtest results for a specified user-inputed portfolio. Additionally, basic statistics, risk profiles and related metrics are displayed. The user may select various investment horizons to view the portfolio.
2) Enhance: Given a specified user-inputed portfolio as a benchmark, a portfolio will be constructed with the goal of beating the corresponding return benchmark, on a risk-adjusted basis. The portfolio will consider a diverse range of assets from which the user can choose inclusions or specify exclusions.
3) Build: For a user with a certain investment goal, the optimization engine will deliver an optimal, risk-adjusted portfolio to keep the user on track when it comes to goal planning.

To satisfy the aforementioned requirements, a suite of quantitative solutions are packaged together to deliver an economically-just framework. To this effect, modern technological approaches are adopted in order to provide answers to technical challenges and mitigate undue risks in the investment planning process. It is the goal of the project to integrate the service offerings in such a way that clearly distinguishes efficient and value-added services to the user.

Development for this project began on September 9th, 2019 with the team first engaging in a conceptualization phase. Following a period of best-practices review, design consultations and feasibility analysis, the development and coding began. The project will be delivered in final on December 4th, 2019.

1. Executive Summary

In defining the problem, as mentioned in the Executive Summary, the objective of this project is to deliver a scalable, Portfolio Optimization Service for both novice and financially literate investors. This service is to be hosted on a public domain and provide financial investment insights which are easy to consume. Elaborating further on the three target functions:

Track a Portfolio

- Provide an end-user with the option to input a portfolio of assets. A portfolio is specified by the collection of tupled-pairs, consisting of a Ticker and a Dollar amount invested.
  - A comprehensive and easy guide will handle error catching, in the event of a misspecified ticker or unfeasible dollar investment.
- Given a portfolio, the Track service will display the historical performance of that investment.
- A guided format will follow the user through the portfolio statistics and help inform how to make investment insights from the displayed results.
- Experienced users will be able to interact with the portfolio to a greater and more comprehensive extent.

Enhance a Portfolio

- Similar to the portfolio input mechanism, a user will choose the asset allocations. The intended goal for implementation is to deliver a portfolio which will dominate the risk-return characteristics of the inputted portfolio.
  - The risk-return characteristics will be determined through quantitative solutioning and optimization, involving maximizing the risk-adjusted returns using a Sharpe optimization.
- To facilitate the optimization, a balanced risk-profile will be fed as input parameters.
- The dominant portfolio will be selected from a list of curated assets maintained by the portfolio optimization service. The list of assets will be detailed in the following sections of this report, but the curation is selected so as to provide a solid parameter estimation basis.
  - Due to scalability concerns, the universe of considered assets is reduced to 18. As bandwidth would increase however, the feasibility of increasing the scope will be demonstrated as within the realm of possibilities.
- Similar to the Track function, a guided format will allow the user to better understand the recommended portfolio by answering the question “So, why this portfolio and what’s so special?”

Build a Portfolio

- Yet to be the most comprehensive feature, the Build option allows a user to specify their exact investment goals and be matched with a portfolio which will best suit their needs.

- From a functional perspective, this involves an understanding of goal planning and how this relates to the user's specified comfort with managing risk.
- The portfolio's Build function will display a stylistically simple and intuitive questionnaire, to understand the needs of the user. The questionnaire will follow three different streams:
    - Accumulating Wealth: Targeted at the user who does not have a specific goal in mind but is interested in investing and would like to see his or her earnings accumulate. This user is likely new to investing, yet to realize the potential for compounded returns. The main goal of this stream is to therefore demonstrate the power of financial growth.
    - Large Purchase: The user who is saving for a large purchase understands the growth phenomenon and wishes to use the portfolio optimization service to assist with their goals ready in mind. The service will leverage this understanding to recommend an investment strategy which is best able to deliver on the return targets.
    - Retirement Planning: The user who is saving for retirement is able to clearly define their financial goals and consistently put away money to realize the goals. Given the likely longer investment horizon, the build function will leverage the trajectory on equities to compound wealth for a critical financial period in the user's life.
- The portfolio solution will be generated through a multi-period framework. This is designed to ensure that the user is protected against transaction costs, unnecessary and unfocused rebalancing and demonstrate some sensitivity to parameter estimates.

Packaged together, the definition of the problem at hand is to deliver an investment service which is best able to accommodate the financial planning requirements of the user.

2. Design Engineering

In the content to follow, there are a countless number of design elements which are carefully chosen before their attribution into the final product. To assist in communication these design considerations, three tools will be used:

1) **Design Matrices:** The design matrices will list, column-wise, all the alternatives considered. Row-wise, the main decision features will be tabled. Included in the table will then be a comparison of each of the design alternatives. Meaning, if there are *n alternatives* to choose from, each row corresponding to a decision feature will have *n* rankings for each alternative.

2) **Alternatives Analysis:** Such an element will table leading design alternatives, listing the relative advantages and disadvantages of each solution. Such a display will make it clear to the user why a design was chosen and what factors led to the decision. An example of a design matrix is as follows, to help answer the question:

What to eat for Breakfast?

Decision: Poached Eggs		
Alternative	**Advantages**	**Disadvantages**
Toast	• Quick to Make • Can be spread with jam	• Low Protein content

3) **Design Q&A's:** For other considerations which perhaps have a limited solution space or are refined in their technical approaches, the question will be posed ("Why solution X for problem Y?") followed by a brief supporting reason. In some cases, due to the scope and limitations of the project, the reasoning may be a referral to a literature body of work or best practices guide. In this case, the content in the Literature Review will be of great relevance and support for sufficient background information.

   In other cases, a reasonable display of justification will supplement the analysis and decision. The extent of this application will pertain to the limitations of resources and the ability to perform a comprehensive review. It is the opinion of the design team that a poorly supported justification offers less value than a strong referral to an Expert's account.

3. Project Resources

This section overviews the technological resources used to facilitate group work and process streams. Please note that references have been provided, with links to each resources detailed below.

Table 1: Consolidation of project links and resources

Resource Type	Resource Name	Link
Online Website	Hosted on AWS	http://ec2-35-183-96-226.ca-central-1.compute.amazonaws.com:5000/
Code Repo	Github	https://github.com/lluo5779/Robo-Adviser
Hosting and database	AWS Developer Account	https://273323797504.signin.aws.amazon.com/console (username and password available upon request)
Project documentations; miscellaneous	Google Drive	https://drive.google.com/drive/folders/1lARuYpCQ7F9kxTdevJ4V_BNWcnDCA79v?usp=sharing
Internal Communication	WhatsApp channel	For internal use only.

3.1 Project Costs

The only incurred cost now belongs to the Tiingo API Premium subscription at $10. The remaining resources are all eligible for a free tier use, which is leveraged to the full extent in delivering good quality services. During production, the AWS RDS instance charges on I/O throughput exceeding the free tier range.

4. Literature Review

The next section is meant to provide an overview of literature and best practice reviews to inform the design decisions that go into the business logic. Note that not all topics discussed in this section will be implemented, nor will all implemented solutions be discussed here.

Modern Portfolio Theory was introduced by Harry Markowitz in 1952 (Markowitz, 1952). Markowitz's seminal paper, which the 1990 Noble prize in Economics was issued for, developed an optimization framework by either minimizing the variance of a portfolio of correlated assets or maximizing the expected return of the portfolio. The asset allocations in a "Markowitz Optimization" is based upon the expected returns and risk profiles of assets; for instance, a high returning asset with low risk is favored over a lower returning asset with higher risk

The concepts and frameworks introduced by Markowitz has since motivated comprehensive research in multidisciplinary financial engineering fields such as computer science, statistics and operations research. The purpose of this section is to provide a literature review of several applications of portfolio optimization, with a supporting analysis on the areas of improvement to the model.

Multi-period Optimization

The premise behind a multi-period optimization framework is clearly spelled out in (Boyd et al, 2017), which casts the objective function over the investment horizon. In defining a generalized risk utility function, in addition to definition functions for the portfolio's transactional and holding costs, the consideration of the multi-period framework balances immediate investment outlooks with longer term views. The benefit of a model adapted as such is a strategic alignment which reduces the executional risks, in terms of transaction costs and exciting position, associated with rebalancing. The main challenge with a multi-period investment model is forecasting parameter estimates, that is expected returns and covariances, over the defined investment horizon.

One natural way of dealing with this uncertainty is by taking an expectation, thus introducing the concept of Multi-stage Stochastic Optimization. For review, the paper entitled "Multi-stage Stochastic Mean-Variance Portfolio Analysis with Transaction Costs" (Gulpmar et al, 2004). In this paper, MSO is applied to a Markowitz Optimization to minimize the variance of a portfolio. Specifically, the authors elect to model the expected returns of the assets with a multivariate normal distribution and define a scenario tree based on the possible realizations of asset returns. What this leads to is setting up the optimization problem whereby you invest at time zero taking into consideration different possible asset return paths {at time zero, if an asset returns 10% then at time-step one the asset might return 8% or 12% which some probability. MSO as developed by (Gulpmar et al, 2004) provides a robust and scalable addition to Markowitz Optimization for which the solution remains tractable. The benefit to MSP is that it treats the asset returns as the random variables that they prove to be. This is largely ignored in a basic Markowitz Optimization and can largely be viewed as a risk mitigation technique by optimization against unfavorable outcomes. The paper also outlines the inclusion of transaction costs, which act as optimization constraints which penalize frequent and large-reshuffling of asset allocations which would otherwise incur large fees. In all, (Gulpmar et al, 2004) provides a quick and elegant guide for MSP with portfolio optimization.

Where in the previous text, MSP is applied to Markowitz Optimization, there are other possible objective functions which specifically target risk exposures. Consider (Mahmutoğulları et al, 2017), an entry to the European Journal of Operations Research entitled "Bounds on risk-averse mixed-integer multi-stage stochastic programming problems with mean-CVaR". This paper applies a mixed-integer variant of MSP to minimize Conditional Value at Risk (CVaR). CVaR is a risk measure which provides insight into the maximum likely loss of a portfolio over a certain investment horizon given a level of confidence. This paper adopts the same MSP framework from (Gulpmar et al, 2004) in terms of handling uncertainty with finite scenarios for asset returns, however, applies this to a different objective function. The interests and applications of (Mahmutoğulları et al, 2017) come from the regulatory requirements regarding CVaR reporting, which has become required under Basel Accords after the 2008 subprime mortgage crisis.

Factor Models

A factor model is one that attributes and associates asset returns with underlying risk drivers. Perhaps the most commonly known factor model is the "Capital Asset Pricing Model", or CAPM for short, introduced by Wiliam Sharpe (Sharpe, 1964). The CAPM equation goes hand-in-hand with Modern Portfolio Theory, justifying the existence of a tangency portfolio which captures the behaviour and generalized performance of the market. Indeed, it is from CAPM that the concept of "beta" gained traction in the investment space. A stock's beta is the factor associated with CAPM which drives asset returns and is a parameter that Sharpe calculated by taking the ratio of a stock's covariance with the market weighted against the market's volatility.

The CAPM equation represents a very simple model, which contributes to its popular use. However, it has been shown that the model fails to capture additional complexities and drivers of asset returns. This has prompted great development in the field of factor-based investing, where money manager's seek to define robust parameterization of factors which may explain asset returns and generate excess alpha. An important stepping stone in development come from Eugene Fama and Kenneth French, in a 1992 paper entitled "The Cross-Section of Expected Stock Returns" (Fama et al, 1992). The 3-factor model introduced builds on the Market Risk factor from CAPM but introduces two additional factors:

- "Small Minus Big" (SMB) factorizes the market capitalization of a stock;
- "High Minus Low" (HML) is based on capturing the explanatory power of any outperformance for growth stocks versus value stocks.

The Fama-French 3 factor model is commonly cited due its offering of improvement over the CAPM baseline. From the pair of researchers, the most recent development in the factor modelling is a 5 factor model which builds on the previous factors with the introduction of 2 more:

- "Robust Minus Weak" (RMW) measures excess returns driven by the health of profilitibility;
- "Conservative Minus Aggressive" (CMA) reflects the differences in conservative portfolios versus aggressive ones.

In addition to Fama and French, Mark Carhartt is widely attributed with the development of a successful factor model which incorporates momentum as a risk factor (Carhartt, 1997). While the possibilities are vast for the application of factor expression, the general process for incorporating factors into a model is characterized by the "Fama-MacBeth" process, which involves data collection on asset returns followed by a regression on the determined factors. For a modern treatment of this process, refer to "A weighted Fama-MacBeth two-step panel regression procedure" (Yoon et al, 2019).

Volatility Modeling

In addition to providing insights about asset returns, factor models provide guidance towards the modeling of volatility among assets in consideration. Particularly, asset volatilities and covariances can be attributed to the factor loadings. Where as this approach carries the simplicity and conveniences typically found in factor modelling, another more rigorous approach is to define stochastic differential equations (SDE) for the volatilities of assets. In the work entitled "Portfolio Optimization & Stochastic Volatility Asymptotics" (Forque et al, 2017) build upon the Merton Portfolio under a multiscale stochastic volatility model, specifically by considering a fast and slow volatility factor. The resulting approach advances from a theoretical lense, built upon the notion that stock market volatility contains high-frequency and low-frequency components. Inevitably under the complex framework, the solutioning reduces down to solving the associated Hamilton-Jacobi-Bellman equations.

In addition to stochastic volatility modeling derived from SDEs, it is possible to consider implied volatilities, as done in "A Forward-Looking Factor Model for Volatility: Estimation and Implications for Predicting Disasters" (Kadan et al, 2017). The approach adopted under this framework derives from the functional form of the factor-regression model, estimating variances based on the implied volatilities from related options. The associated limitations are that to covering a universe of assets with a sufficient volatility surface, but the researchers identify a relevant application under forward looking guidance.

Yet another approach is defined by taking the stock price under a regime-switching Heston model, as done with (Liu et al, 2017) in “Portfolio Optimization Using Regime-Switching Stochastic Interest Rate and Stochastic Volatility Models”. Consideration under a regime-switching model captures macro-level changes observed across the business cycle. The result is a modeling framework that demonstrates greater robustness and carries more descriptive power.

Regime Switching Models

Regime-switching models carry greater descriptive potential by offering flexibility and consideration to macro level market regimes. Due to the intuitive approach, the application of regime-switching in regards to factor modeling is a common approach taken in asset management. Economic models incorporating Markov regime switching approaches date back to 1989, with James Hamilton’s paper entitled “A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle” (Hamiton, 1989).

Issued in the whitepaper entitled “Markov Switching Models and the Volatility Factor: An MCMC Approach”, the French Quantitative Asset Manager Ossiam provides an overview of a Fama-French model cast under a Markov-switching model. What’s informative about this review is the treatment given to regime selection as well as Gibbs sampling for transition state probabilities. In regards to the former, the number of free parameters associated with a K-state regime environment associates with it a “BIC” score, which reflects the integral of the likelihood function times the prior probability distribution over the parameters of the exponential model. This BIC scores allows one to arrive at an optimal number of states, which will correspond to the minimum BIC score. In regards to the Gibbs sampling over, this is a sampling-technique belonging to the class of Markov Chain Monte Carlo MCMC simulations. Gibbs sampling provides access to a hidden Markov chain only through the realization of the process’s outcomes. In what is otherwise a simple construct, the MCMC techniques will be further explored below.

Further insights into regime switching models allow for the study of macro level changes, as reviewed in “The Effect of Market Regimes on Style Allocation” by (Ammann and Verhofen, 2009). What is to be taken away from this review is that the observed regime will govern the most effective investment allocation style. For instance, Ammann and Verhofen find that value investing is a rationalized strategy under regimes characterized by high volatility, and momentum investing is well suited for low volatility regimes.

Markov Chain Monte Carlo Sampling

Due to the nature of the Markov Regime switching approach, transition probabilities need to be inferred so as to appropriately define the markov chain. One approach owes to the frequentist views of maximum likelihood estimation, as done with (Guidolin and Timmermann, 2007). However, Ossiam notes complexities in this approach under a multivariate model, specifically under the presence of many factors governing asset returns. This is where Markov Chain Monte Carlo (MCMC) offers efficient and simple relief.

One such iterative algorithm to MCMC is referred to as Gibbs Sampling. Summarized in the paper by (van Ravenzwaaij et al, 2018), Gibbs sampling is a non-rejectory iterative process that samples from the conditional distributions of each parameter, using sequential updates. The result of this process is a limiting distribution which is the joint posterior density. Applied to regime switching, the resulting density informs the transition state probabilities. While Gibbs sampling is but one example of an MCMC algorithm, others are explores and compared in the work by (Chauveau and Vandekerkhove, 2007).

Black Litterman and Generalized Bayesian Models

The famed Black Litterman model is an application of a Bayesian Model. Characterized by a prior distribution coming from a normal market centered on implied market-equilibrium returns, the Black Litterman model allows for robust views which blends investor's subjective views and opinions (Satchell and Scowscroft, 2000). For this reason, popularity in the model comes from the intuitive and investor-reinforcing outcomes. However, the model is not without its shortcomings. Owing to the closed-form simplicity, the model assumes that the distribution of asset returns is normal, which is an assumption that is demonstratively false under real conditions. Accounting for this, it is possible to extend the Black Litterman Model beyond Normal markets, into the realm of generalized Ellipsoidal Distributions (under which, a Normal distribution exists as an example of).

The work outlined in "A Black–Litterman asset allocation model under Elliptical distributions" (Xiao and Valdez, 2015) provide the mathematical framework to make such modifications. Owing to favourable properties of Ellipsoidal distributions, Xiao and Valdez are able to recover closed-form solutions for distributions including the T-distributions.

Applied more generally, Attilio Meucci provides a recipe for a general Bayesian model (Meucci, 2006). Under such a model, the prior may be arbitrarily assigned and is not anchored to the equilibrium returns. To deal with such generality, a Copula Opinion Pooling (COP) approach is taken and results are empirically gathered. The concept of COP is analogous to that of distribution mixtures.

Discovering Market Views with Machine Learning

As discussed above, the key advantage provided by the Black-Litterman framework is the ability to leverage the views held by an investor regarding the market and the respective confidence in those views. While this can be very useful in practical settings, clearly the strength of the model is proportional to the quality of the views held by the investor. Finding robust and accurate ways to discover appropriate views of the market will greatly increase the effectiveness of the Black-Litterman framework. In particular, it is of interest to leverage recent advances in machine learning to help inform these aspects of the framework.

The low-volatility anomaly is an observation that stocks with lower risk tend to have higher returns than their high-volatility counterparts (Haugen et al, 1991). It is important to note that this is not a proven phenomena and contradicts most financial theories showing that taking on higher risk rewards the investor with higher rewards. However, if this is a view the investor is willing to adopt, machine learning can be used to discover the assets that are best complemented by this view. Predictive models can be used to classify the volatility of stocks as high or low based on historical time-series data. The views of the investor regarding the low-volatility stocks that will overperform can then be easily integrated into the Black-

Litterman framework. In fact, a study of the Korean stock market showed that this method of extracting views dominates the market portfolio and increases profitability (Pyo et al, 2018).

While the above strategy is an intelligent way of leveraging machine learning to identify stocks that could complement an investor's specific view, the overall strategy for discovering views is static and relies on the knowledgeability of the individual investor. Thus, it will not be able to adapt to scenarios in which the low-volatility anomaly is not observed. Financial articles, releases, and online posts are clearly dynamic and while they do not directly affect the market, the "mood" of the investors they represent could affect it indirectly. For instance, the action of market participants are in part influenced by their own views and the views of others. It is safe to assume that at least a portion of these views will be indicated by the online data stream from the participants. Recent strides in Natural Language Processing (NLP) have provided ways to extract sentiment from text documents. Applying these strategies to financial text data could provide a method for discovering dynamic views directly linked to public mood. Injecting the discovered sentiment into the Black-Litterman framework has been shown to increase the effectiveness of the model (Xing et al, 2018). Despite the empirical evidence in favour of this strategy, sentiment may not be the best way of discovering views. Sentiment can be very noisy and is not a very expressive ontology to inject into existing frameworks. Instead of predicting sentiment, one could simply use the text data and machine learning methods to generate a matrix of market views directly. While this would require end-to-end or segmented training involving the Bayesian asset allocation model itself, it may also provide a more robust and powerful set of market views.

Deep Portfolio Theory

Identifying market factors that best explain the expected return of a portfolio is a very important part of financial modelling. Methods of analyzing historical data to identify the most important factors, such as PCA, are very popular. While it can be a powerful technique, PCA and other similar methods are only scratching the surface of what could be very predictive factors. These factors can be thought of as latent, unobserved, or deep factors and the fundamental idea behind them is very popular in the machine learning community. Compressing data into a lower-dimensional yet representative latent space has been shown to increase predictive power across a wide variety of tasks and models. One of the most popular models that discover an efficient encoding or latent space is known as an autoencoder. Very simply, the autoencoder encodes data into a low-dimensional space, decodes the space into the original dimensionality and then compares the decoded data with the original data. It seeks to minimize the differences between the two instances of data. Autoencoders have been proposed as a way to create a latent portfolio and market map based on historical data that captures all the important factors attributed to the performance of the market. Extending this idea to portfolio optimization involves three major steps. First, autoencode the data into a market map (low-dimensional representation of the data). Secondly, find a portfolio map that optimizes for a given objective based on the market map. Finally, optimize the two objectives simultaneously to find solutions that offer a balance between the errors present in both optimization schemes (Heaton et al, 2016). Finding these maps could be an effective way to discover latent factors hidden to most methods of portfolio optimization.

Intelligent Design of Surveys

As discussed above, a Black-Litterman model is dependent on the quality of the views provided by the investor. To an investor that is not financially literate, expressing their views in a way that is easily integrated into the asset allocation model is difficult. Designing a questionnaire that best captures the views of an investor could therefore greatly improve a Black-Litterman model but also the entire strategy associated with generating a portfolio for the investor. Verbal Decision Analysis (VDA) can provide a methodology for creating these surveys. Examples of such questionnaires have been presented in (Silva et al, 2016).

5. Business Logic

In this section, a comprehensive account will be detailed for the solutions implemented to address the business logic of the project. Therefore, under the following headings, there will be two characteristic steams: (1) a description of the design implementation and (2) a justification for the design implementation. Where appropriate, design alternatives will be presented, followed by a decision analysis weighing all relevant benefits and costs.

5.1 Summary and Overview

The table below provides a brief overview of the implementation for each aspect of the project's business logic. The titles in the first column refer identically to headings which will subsequently follow the content in this section. Please note that keywords will be embolden.

Table 2: Summary of business logic aspects.

Aspect	Solution
Data Inputs	Financial Data is sourced from: Tiingo (API service) and Yahoo Finance (custom built web-scraping service).
Parameter Estimates	Parameters for two periods are estimated: the immediately upcoming period (P1), and the one-period forward extension (P2). For P1, a Regime-Switching Factor Model is cast from the Fama-French 5 Factor Model. For P2, a Machine Learning pipeline makes use of Natural Language Processing to inform views. Both P1 and P2 estimate are fed into a Black-Litterman model for the final output of parameter estimates, which are fed into the portfolio optimization.
Multi-Period Framework	The objective target is to maximize the trade-off between Mean and CVaR. Transaction costs are modeled using a square-root-based Volume estimate, risk preferences are adjusted to user preference levels, and various constraints are further applied such as cardinality and diversity.
Backtesting Functionality	A function-based service which returns the portfolio risk-return characteristic over a given time interval.
Portfolio Enhancement	Portfolio optimization service which is derived to beat, on a risk-adjusted basis, the returns of a benchmarked portfolio.
Scheduled Routines	Prices are updated on a daily basis which then updates the values of any portfolio's saved by the user base. Parameter estimates are updated on a monthly-basis, in accordance with the publishing of factor data from the Fama/French repository.

5.1.1 Initially Proposed Optimization Framework

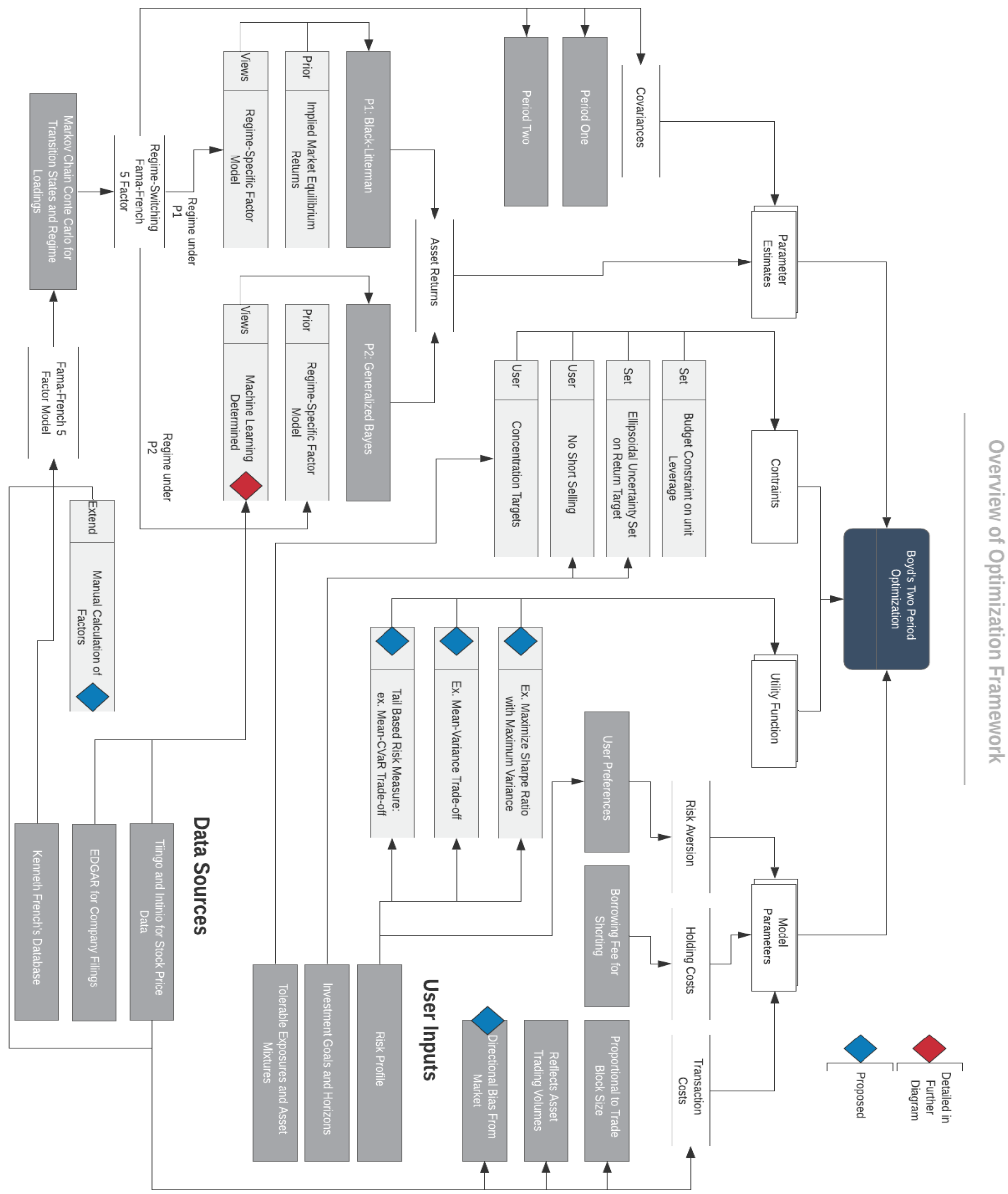


Figure 1: An overview of the portfolio optimization model.

5.1.2 Final Optimization Framework

Whereas previously, we have shown the proposed optimization framework, the final implementation is presented in the following figure. Note that the selection is focused only on the parameter estimations feeding into the optimization.

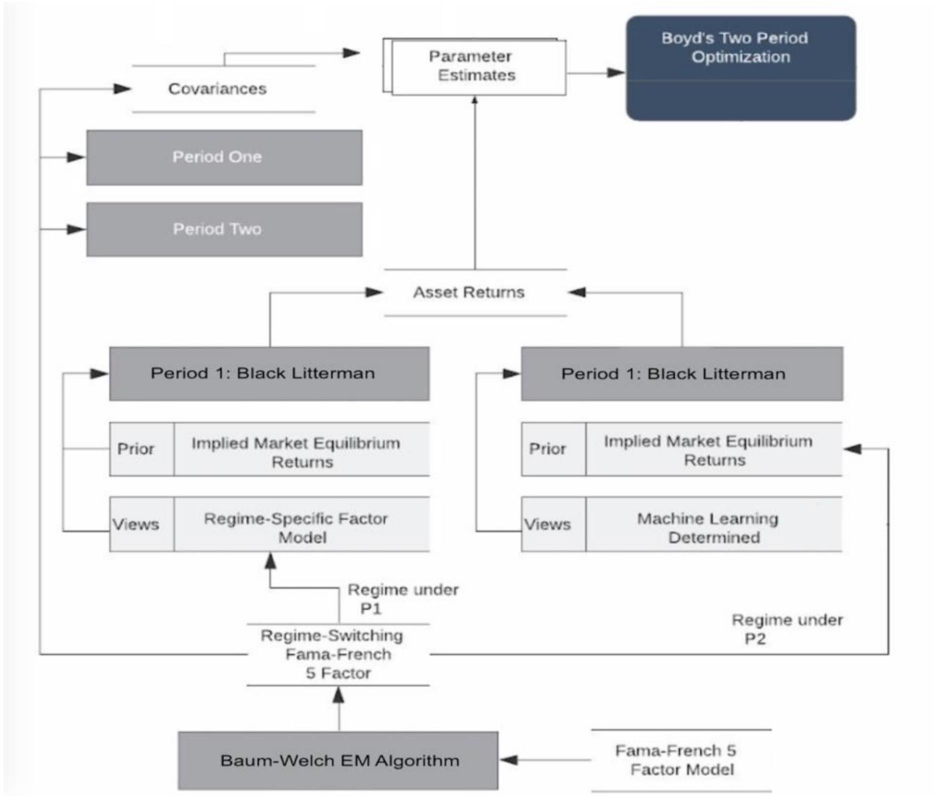


Figure 2: Summary of Finalized Business Logic Framework.

5.2 Data Space

5.2.1 Sources of Financial Data

Financial data, that is asset prices and volumes, are sourced from the free Tiingo client. This service provides end-of-day adjusted closing prices for a large universe of assets. While the service is limited to a certain number of calls (per hour is 500), the service provides a great enough support in order to make the project feasible. The live data is pulled on a daily basis in order to calculate the values of portfolio and perform on-the-fly parameter estimates as required.

Additionally, Market Capitalization data is consumed from Yahoo Finance. While a ready-to-use API service from Yahoo Finance has long since been discontinued, a custom built web-scraping script was designed to extra HTML code from the live web pages and infer relevant financial data. This proved crucial in sourcing data, such as historical market capitalizations, for the assets under consideration.

5.2.2 Universe of Tracked Assets

Indices and ETFs are selected to provide sufficient coverage of sectors and exposures. As discussed in the design considerations section, indices and ETFs are chosen due to the adherence in the value of passive investment strategies. Index investment has become a highly-recommended and easy to enter space for investment. These conditions fit the requirements for this project ideally, provided a complete offering in terms of exposures with mitigated and diversified requirements.

Additionally, addressing the scalability, the inclusion of assets also extends to 10 single name stocks. These assets are directed to users who specify a greater tolerance for risk and accordingly, the chosen assets are ones that, from a balanced perspective, deliver strong risk adjusted returns.

The selection of the Universe of Tracked assets is a major design consideration which is explained further in the design considerations section of this report. To this effect, 18 indices and ETFs are chosen for scrutiny, during the parameter estimation phase and the portfolio optimization output. These assets are listed below in the table.

Table 2: List of ETFs and Indices under consideration for portfolio optimization.

Tracks	ETF	Ticker
Market-Cap Equity		
S&P Total Market Index	iShares Core S&P Total US Stock Market ETF	IVV
Dow Jones U.S. Index	SPDR Dow Jones Industrial Average ETF	SPDW
S&P 500	SPDR S&P 500	SPY
S&P 500 Top 50	Invesco S&P Top 50 ETF	XLG
S&P Asia 50	iShares Asia 50 ETF	AIA
S&P China BMI	SPDR S&P China ETF	GXC
Sector Solutions		
S&P Select Sector Consumer Discretionary	SPDR - Consumer Discretionary Select Sector Fund	XLY
S&P Select Sector Energy	SPDR - Energy Select Sector Fund	XLE
S&P Select Sector Financials	SPDR - Financials Select Sector Fund	XLF
S&P Select Sector Health Care	SPDR - Health Care Select Sector Fund	XLV
S&P Select Sector Industrials	SPDR - Industrials Select Sector Fund	XLI
S&P Select Sector Materials	SPDR - Materials Select Sector Fund	XLB
S&P Select Sector Technology	SPDR - Technology Select Sector Fund	XLK
S&P Select Sector Utilities	SPDR - Utilities Select Sector Fund	XLU
Real Asset Solutions, ESG and Thematic		
S&P Global Clean Energy Index	iShares Global Clean Energy ETF	ICLN
S&P Global Water Index	Invesco S&P Global Water Index ETF	GCW
S&P Global Timber & Forestry Industry	iShares Global Timber & Forestry ETF	WOOD
Dow Jones U.S. Real Estate Index	iShares US Real Estate ETF	IYR

In addition to the 18 Indices and ETFs, there are 10 single-name stocked for which parameter estimates are provided and portfolios may include. These assets are all chosen from the SP500 based on their historical demonstration of strong return profiles.

Table 3: List of Single Name stocks used.

Ticker	Company Name
F	Ford Motor Company
Dis	Walt Disney Co
MCD	McDonald's Corp
KO	Coca-Cola Co
PEP	PepsiCo, Inc
JPM	JPMorgan Chase & Co
AAPL	Apple Inc
PFE	Pfizer Inc
JNJ	Johnson & Johnson
ED	Consolidated Edison, Inc

5.2.3 Data Inputs Design Engineering

Design Question: Why focus on Market-Cap Equity, Section Solutions and Thematic?

Across the thematic groups, the mixture of the three different classes allows for a portfolio optimization service to capture trends that may be specific, or shared. The exposures selected are comprehensive carry large market capitalizations and Net Asset Values, including:

ITOT	$23,914,129,408.00	XLG	$842,611,072.00	DIA	$21,614,039,040.00
GXC	$1,167,455,616.00	AIA	$1,077,156,480.00	SPY	$275,287,670,784.00
XLY	$14,352,367,616.00	XLE	$9,978,409,984.00	XLF	$23,675,326,464.00
XLV	$18,043,512,832.00	XLI	$10,040,603,648.00	XLB	$3,403,495,424.00
XLK	$23,339,634,688.00	XLU	$11,234,997,248.00	ICLN	$369,268,032.00
CGW	$741,918,016.00	WOOD	$240,967,632.00	IYR	$4,701,537,280.00

Design Question: why include ETFs which track the same class, for instance IVV and SPY with regards to the S&P 500?

It is true that the indices are tracking the same base, in this case, the S&P 500, but the assets are not identical. They differ based on their compositions, expense and management fees and inception. Specifically, SPY is the larger ETF, but its associates large expenses, compared to ITOT. Conversely, SPY has greater liquidity leading to lower transaction costs. The portfolio optimization is thus primed to deal with the trade-offs and deliver a mitigated solution which tracks the S&P 500 target to the most accurate level.

5.3 Parameter Estimation

Calling to the diagram from section 5.1.2, the parameter estimation is accomplished through 2 stages. Each stage provides estimates for the two periods being optimized under the Boyd model. The first period estimates result from the expected value from a Regime-Switching Factor model. The second period estimates come from a Machine Learning Recurrent Neural Network. These estimates, separately are fed into a Black-Litterman model, with the final output harmonizing the period estimates. The remaining content of this section addresses the parameter estimation process in detail.

5.3.1 Implementation of the Regime-Switching Factor Model

In an attempt to capture market insights pertaining to different business cycles, a Markov-Regime Switching model is implemented. A regime switching model is a powerful way of enhancing the performance of an underlying factor model, and works by imposing a stochastic framework on the calibration of factor loadings. By parameterizing the number of regimes, an Expectation-Maximization algorithm can be run to determine the best set of factor loadings that represents the clustered market.

For the application to this project, it is assumed that the number of regimes is fixed at two. This has a convenient and intuitive significance when it comes to the financial industry, corresponding to a BULL and BEAR state respectively. This also is well supported in leading academic papers, justifying the two state model from a data-driven approach, particularly, it can be shown that in most applications, a two-state regime settings minimizes the Bayesian Information Criterion. What’s more, choosing two regimes is convenient from a computational perspective, a benefit for this project due to the finite resources and limited computing power available. The two state regime also minimizes the chances of overfitting data.

To summarize the EM algorithm performed, the adaption of the Baum Welch algorithm is presented below:

- The Expectation-step: Set initial parameter estimates for factor loadings and then calculate the conditional distribution for regimes, given the loadings. From this measure, the expected log-likelihoods of the data with respect to the regimes are known.
- The Maximization-step: Maximize the expected log-likelihood with respect to the conditional distribution of the hidden regimes. The result is an improved estimate for the loadings.

The emissions from this algorithm give estimated transition probabilities and the factor loadings, covariances. It is assumed that the factors following a Gaussian Distribution, are stationary, and all contribute to regime switching. To demonstrate the performance of the model, the following figure presents smoothed probabilities predicting the incidence of each state. For reference, the daily returns for the SPY index (tracking the SP500) are also shown in alignment with the timeseries.

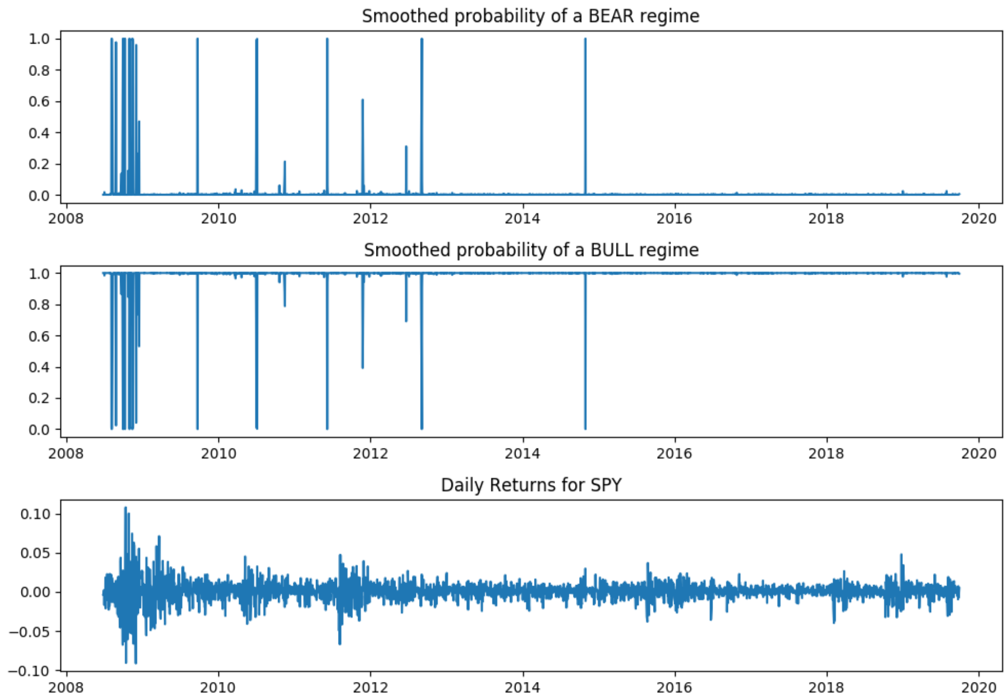


Figure 3: Output of Smoothed probabilities of regime state, back tested over an 11 year period.

From the figure above, it is clear that the model is able to accurately react to the Global Financial Crisis of 2008, where the probability of being in a BEAR state rose to a certainty. What's more, the model captures the volatile characteristics of several regimes thereafter. Curiously, from 2015 onwards, the model settles into almost perpetual BULL market. While on a larger trend this is accurate and representative of the past five years, the regime switching model does not trigger a regime change during the volatile end of 2018.

This limitation may in fact be the result of the calibration process. A regime switching model is powerful when it is able to perform the Expectation-Maximization over a long time period. This is to ensure that the model captures representative data from all possible market states. Due to computational restrictions for the project, a scalable service could only be provided when the regime switching model was calibrated and trained using 10 years of prior data. For any training period exceeding 10 years, the computation expense of performing the backwards-forwards pass of the algorithm made wait-times too long.

5.3.2 Validation of Regime-Switching Factor Model

The baseline for the Regime-Switching Model is the Fama-French. This was noted due to its comprehensive ability to track close to realized asset returns. The remainder of this section is dedicated

In choosing the factor model, two alternatives were closely explored. That being the Capital Asset Pricing Model (CAPM) and the Fama-French 3 Factor Model. As such, the validation of the implemented factor model relied on a calibration period consisting of 20 periods. During each period, a collection of in-sample data was used to fit the model, followed by a corresponding testing period of the same duration.

To inform the decision over which factor model should be pursued, the following plots for the r-squared value, measuring goodness-of-fit, and Root-Mean-Squared-Error provide useful information. Factors data are consumed directed from the Kenneth-French library.

Table 4: Comparison of Factor Models in terms of statistical fit.

	Goodness of Fit (R-squared)			Root-Mean-Squared Error		
	CAPM	**FF3**	**FF5**	**CAPM**	**FF3**	**FF5**
count	*20*	*20*	*20*	*20*	*20*	*20*
mean	*0.930*	*0.941*	*0.942*	*0.00246*	*0.00225*	*0.00226*
std	*0.071*	*0.067*	*0.069*	*0.00196*	*0.00192*	*0.00193*
min	*0.760*	*0.766*	*0.757*	*0.00080*	*0.00053*	*0.00052*
25%	*0.908*	*0.911*	*0.908*	*0.00112*	*0.00076*	*0.00075*
50%	*0.926*	*0.969*	*0.969*	*0.00185*	*0.00175*	*0.00175*
75%	*0.965*	*0.993*	*0.993*	*0.00290*	*0.00288*	*0.00290*
max	*0.983*	*0.997*	*0.997*	*0.00769*	*0.00759*	*0.00774*

It was noted a near identical performance to the Fama-French 3 Factor model, but characteristically with a higher degree of volatility. Particularly from the Factor model calibration, it is apparent that the Fama-French 5 Factor model has the potential to offer strong results when calibrated in an appropriate regime or state. The drawback from this consideration is that the model suffers performance-wise when the regime or state is not accurately represented by that during the calibration period.

One reason for the greater degree of variability with the Fama-French 5 Factor model deals with an issue of over-fitting. Indeed, more factors are not always ideal and may hurt performance. Under a view of regime switching states, it is clear to see that during a period of a regime switch, a model with greater feature count may suffer from the overfitting issue. It was this issue that motivated, in part, the adoption of a regime-switching approach.

The Baum-Welch algorithm was used to determine the regime-switching nature. It was assumed that factors followed a gaussian distribution and no assumption on the driving factors for the regime switches were made. This means that any of the five factors may pose a latent effect on the loadings under each regime. Strictly speaking from a data-driven approach, this was convenient. However, as discussed in the limitations section, this poses concerns from a noise perspective.

Two regimes were assumed, corresponding to a bull market and a bear market. This follows a strong industry perception of the benefits of modeling two period financial applications as such. It can be justified further by minimizing the Bayesian Information Criterion under k-regimes. From this perspective, it can be shown that under some application 2 regimes is sufficient for this minimization.

For this assumption of two regimes, an initial loadings were calibrated using Ordinary Least Squares Estimates on Factor loadings, and K-Means with 2 clusters on the realized returns. From this point, a forward-backward Expectations Maximization algorithm was run where emissions were granted as a transition matrix and factor loadings and covariances. Once the loadings were calibrated, using 10 years as a basis, the expected returns were calculated as the probability weighted sum of being under each regime. Likewise, the covariances were calculated noting additional terms due to cross variances.

5.3.3 Machine Learning

Portfolio optimization is not a new application of machine learning (Heaton et al.) (Oh et al.). However, there have not been any significant breakthroughs of note in the field. This may stem from the ambition of the optimization task defined in most of the models. Most commonly, machine learning tools are used to construct portfolios that optimize returns (or some function of returns) for the investor. In supervised learning cases, this is usually cast as a binary classification problem (since continuous labels are in general more difficult to optimize for) in which positive returns of stocks are given a positive label and negative returns are given a negative label. Binary discretization of continuous values clearly limits the information available to the algorithm. Thus, the machine learning model is being asked to approximate the probability distribution governing the returns of a portfolio using artificially restricted information.

Methods to help solve this difficult problem such as using Natural Language Processing (NLP) or Time Series models are a step in the right direction since they incorporate trends and exogenous sources of data to help converge on the optimal model parameters. However, these methods alone are still relatively unsatisfactory. A potential way to incorporate the power of machine learning models into portfolio optimization more effectively could be simplifying the problem given to the algorithm. For instance, asking the model to create an optimal set of views for the portfolio and measuring the effectiveness of those views when incorporated into a larger optimization framework is a much simpler task then creating an optimal portfolio from scratch. Furthermore, the ability of the larger optimization framework to include constraints greatly increases flexibility (as well as reduces the search space) of the generated portfolios. It should be noted that from the universal approximation theorem, a feedforward neural network with a single hidden layer of finite size can be trained to approximate any function (Hornik 1991), presumably including the most effective portfolio optimization technique. However, this is limited by training time and the amount of data available, making universal approximation practically infeasible.

The problem setup injects the machine learning model into the middle of the optimization framework. The output dimensions of the model are constrained to the necessary size of the views. The outputted views are then fed into the generalized Bayesian optimization to find the estimated returns. The training scheme is set up as a classification problem where the machine learning views are trained to outperform the classic views. If the machine learning views outperform the classic views, they are assigned a target value of 1, otherwise it is set to 0.

5.3.4 Natural Language Processing

As the optimization framework time horizon increases, so too does the uncertainty and noise in the predictions. Methods of increasing the robustness of predictions include injecting a broader scope of data that may capture trends in the market more completely, and using more powerful modelling techniques to identify these trends. The use of machine learning is an obvious candidate for incorporating these methods. More specifically, Natural Language Processing techniques can be applied to text data as a method from learning from related to yet distinct from the numerical financial data used for classic optimization. By using financial news data from Tiingo filtered by assets in the portfolio, the machine learning model will be trained to extract information from the text that will give indications about the future trends in the market.

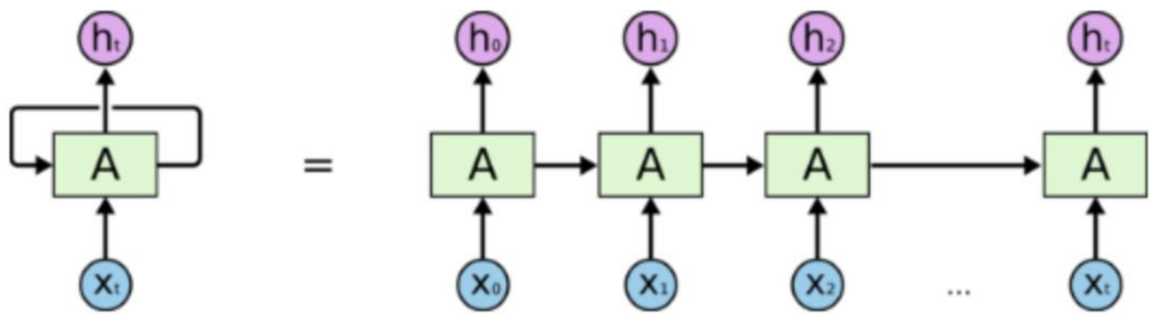


Figure 4: Recurrent Neural Network Architecture.

Recurrent Neural Networks (RNNs) are the standard models for text classification. RNNs make use of internal memory to help learn from input data by feeding the state of the hidden layers in the model back to the previous layer (see figure above). The cyclic nature of this model makes them very well suited for learning from sequential information, as information from one time step is used to process the previous timestep. Vanilla RNNs do suffer from some practical problems during implementation. During training, the weights (usually values less than 1) associated with the recurrent cycles are updated by multiplication with the connection weights, causing a small number to become smaller at an exponential rate. Since the gradient of a very small number is close to 0, the units associated with the start of a long sequence do not learn very much from the gradient during training. This is known as the vanishing gradient problem, where the neural network "forgets" the information learned from units near the start of the sequence. The state-of-the-art variant of an RNN is known as a Gated Recurrent Unit (GRU) that seeks to address the vanishing gradient problem (Chung et al). It uses a "reset gate" (r) that lets a certain amount of information that should not be forgotten flow freely through the network as shown in the figure below.

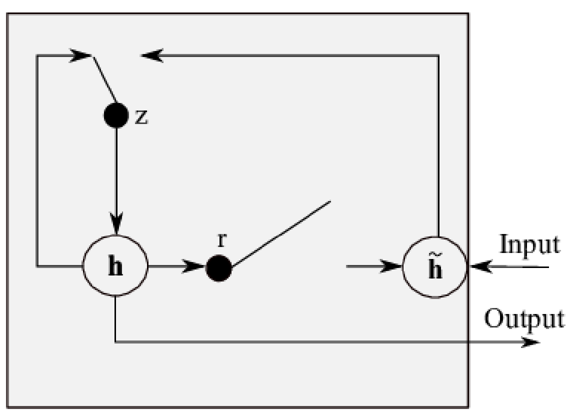


Figure 5: GRU Architecture.

There are other, simple machine learning methods for predicting from text documents, although they are not able to fully make use of the sequential nature of the data. The following table shows a decision matrix used to decide on the type of machine learning model to be implemented. Accuracy is given the highest weight, as it is most crucial for generating meaningful portfolios. Ease of implementation is also important given the time constraints of the project. Since re-training only needs to be done periodically, a quick training time is not the most important feature required of the model. Finally, while interpretability of how the views were generated is a bonus, it is not a strict requirement of the optimization framework. Therefore, it is given the lowest weight.

Table 5: Machine Learning model decision matrix

Feature / Model	Random Forest	MLP	RNN	GRU
Training Time (weight = 1)	5	4	3	2
Accuracy (weight = 3)	1	1	3	5
Ease of Implementation (weight = 2)	5	5	3	2
Intuitive Results (weight = 0.5)	1	2	3	4
Total (weighted)	18.5	18	19.5	**23**

From the table above, the GRU is the most appropriate model. Financial news data relating to all relevant assets in the portfolio is fed into the GRU. Six years of historical data is used to pre-train the model using the loss function outlined in the previous section. The weights of the model are then saved to be used for prediction during the portfolio optimization routine.

5.3.5 Time Series Prediction

Using a sequence of historical prices as an input to a GRU is another method of estimating views. While historical prices do not contain the same richness of data as text, the GRU may still be able to uncover outlying trends that are missed in classic approaches of estimating views. The problem setup for this method is exactly the same as with the text data, except the sequential nature of the input data is a reflection of evolution through time as opposed to sequences of words. As a result, the signal picked up from the GRU will be from recent price trends as opposed to the latent market views hidden in news articles.

5.3.6 Black-Litterman

Black-Litterman view modification is a power tool which may be used to augment outstanding market equilibrium positions. Using the market equilibrium as a guide, inputs from the Machine Learning model and Regime Switching Factor model learning served as “views” which augmented the market equilibrium returns. The result was a new set of returns and covariances, whereby the blend of the data from the Machine Learning and Regime Model was imposed with the nature of the market.

To summarize,
Period 1 ...

- Prior Distribution, parameterized by Market Cap. Equilibrium Returns and Empirical Covariances
- Likelihood parameterized by the expected returns from the Regime Switching Model, at a confidence level given by the regime-adjusted covariance matrix.

Period 2 ...

- Prior Distribution, parameterized by Market Cap. Equilibrium Returns and Empirical Covariances **but augmented by predicted future regime state**
- Likelihood parameterized by the expected returns and covariances from the Machine Learning Model.

5.4 Multi-Period Framework

Shown in the formulation below is the optimization framework adopted for the portfolio optimization service.

$$\begin{aligned} \max_{x_1, x_2} \quad & \left(\mu_1^T x_1 - \gamma_r CVaR_{\alpha,1}\right) + \left(\mu_2^T x_2 - \gamma_r CVaR_{\alpha,2}\right) \\ & - \gamma_t \phi_t(x_1, x_2) - \gamma_h \phi_h(x_1, x_2) \\ st. \quad & 1^T x_1 = 1, \quad 1^T x_2 = 1 \\ & \mu_1^T x_1 - \epsilon_2 \sqrt{x_1^T \Theta_1 x_1^T} \geq R_t \\ & \mu_2^T x_1 - \epsilon_2 \sqrt{x_2^T \Theta_2 x_2^T} \geq R_t \\ & x_1 \in B_r, \quad x_2 \in B_r \end{aligned}$$

The optimization model, as previously mentioned, performs a two-period optimization based on the Boyd framework. The objective target is to maximize the trade-off between expected returns and expected shortfall. In other words, the optimization performs a maximization of Mean-CVaR. Additionally, transaction costs as well as holding costs are included in the objective statement as penalty factors, which leads to a forward-looking model that is able to balance the risk performance with respect to multiple dimensions. The remaining content in this section will address the optimization model, including the validation of its target as well as a general discussion in regards to each constraint applied.

5.4.1 Validation of Mean-CVaR Objective

The base portfolio engine makes use of a Mean-CVaR Maximization. In the preceding analysis, the backtested performance of the optimization will be summarized. Furthermore, the performance of the portfolio will be compared against two other portfolio optimizations: the first, following a Risk-Parity Optimization and the second, under a Risk-Parity Optimization.

In setting up the following experiments and backtest values, it is necessary to note that all portfolios are calibrated under the same parameter estimates and risk profiles. This is to ensure that the lift in performance can be appropriately compared. The backtest began on November 29th, 2004 and culminated on September 26, 2019. The dates in this session where chosen as a design consideration (refer to the appropriate section) so as to ensure that the results were relevant over current market conditions. What’s more, the lengthy period of time allows for a good opportunity to calibrate the Regime Switching Factor Models, including information on the Global Financial Crisis and the boom that followed thereafter.

The following table presents a comprehensive overview of the backtesting validation procedure that justifies the decision to perform Mean-CVaR optimization. In and of itself, the selection of the two baseline portfolios was a choice. Selecting Sharpe optimization and Risk-Parity optimization provided a sufficient view of the application capabilities, in the Sharpe optimization focused on risk-adjusted returns whereas Risk-Parity Optimization focused on an equal distribution of risk amongst the assets.

Portfolio Returns: Comparison Between Different Optimization Schemes

Mean-CVaR compared to Risk-Parity: Cumulative Returns

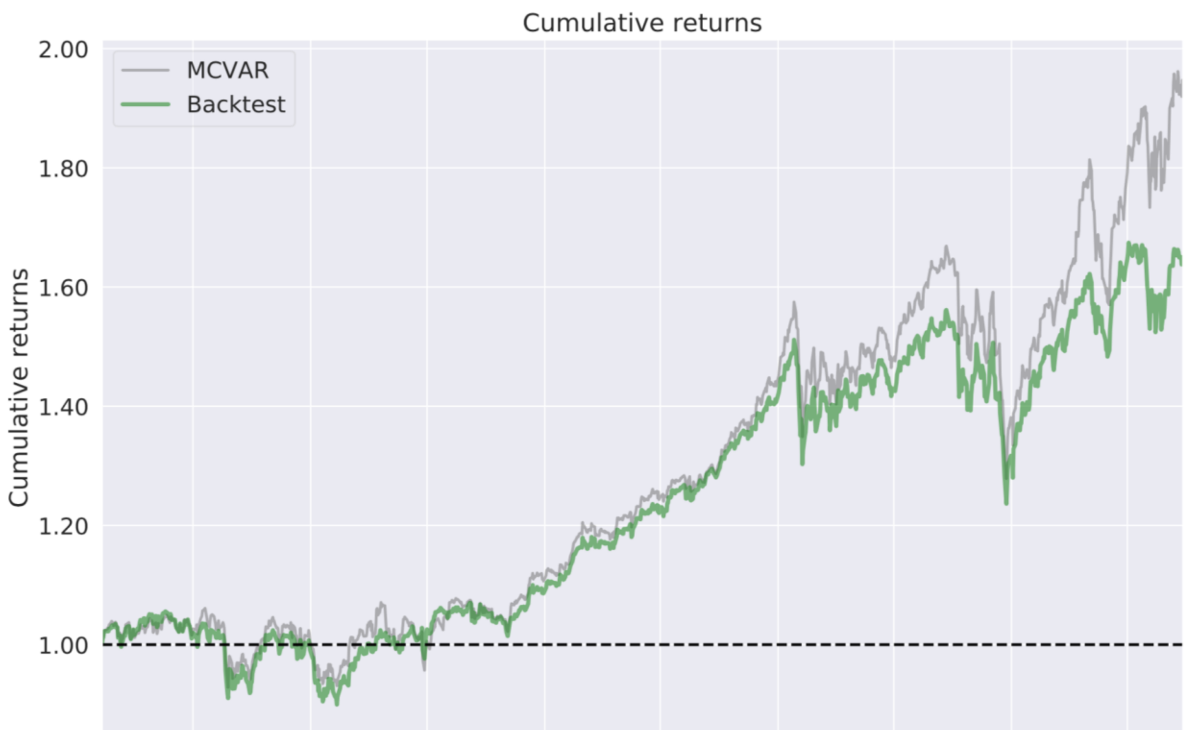


Figure 6: Cumulative Returns comparison for the Mean-CVaR Portfolio and the Risk Parity Portfolio.

The performance of both the MCVaR portfolio and the Risk Parity portfolio were, all things considered, nearly identical. It is apparent that the risk-sharing characteristics of the Parity portfolio performed similarly well as the drawdown-limiting features of the MCVaR portfolio. This signals, more so than anything, an importance of risk-mitigation when it comes to investment strategy, and that sometimes the best offense is a solid defence.

Mean-CVaR compared to the Sharpe Portfolio: Cumulative Returns

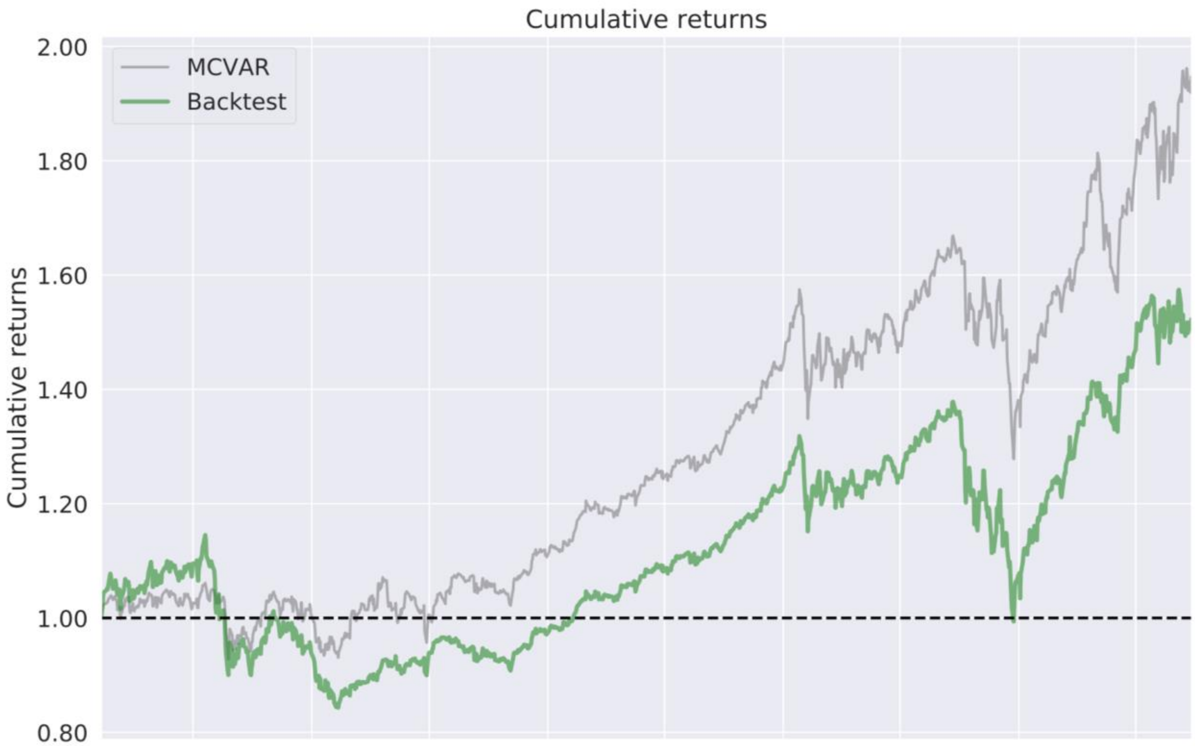


Figure 7: Cumulative Returns comparison for the Mean-CVaR Portfolio and the Sharpe Portfolio.

Comparatively, the returns from the MCVaR Optimization dominated that of the Sharpe Portfolio. Despite the two portfolio's trending the same way, it appears that the Sharpe Portfolio suffered a greater comparative loss at the beginning and failed to recover. This ultimately unearths an important characteristic of the MCVaR optimization process, specifically that drawdowns are limited, allowing the portfolio to realize greater cumulative returns in the long run.

Rolling Volatility: Mean-CVaR

The rolling volatility of the portfolio was primarily representative of the underlying market conditions. This much is shown by how closely all portfolios, Sharpe and Risk-Parity included, trended together. To this effect, it comes as no surprise that the MCVaR portfolio demonstrated muted volatility during quiet periods in the market, and exaggerated volatility when markets are more volatile.

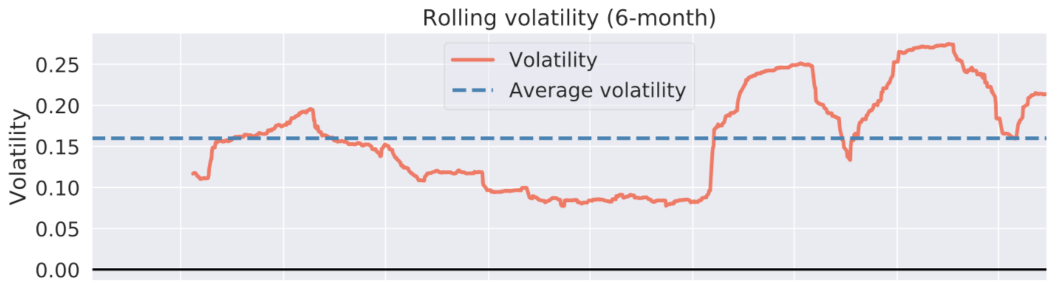


Figure 8: Rolling Volatility for the Mean-CVaR Portfolio

Rolling Sharpe Ratio: Mean-CVaR

Similarly, the rolling Sharpe ratio of the portfolio speaks more so to the tracking of the overall market than the unique characteristics of the optimization. Most noticeably however are the periods when the Sharpe ratio exceeds 1, indicating a period of strong risk-adjusted returns.

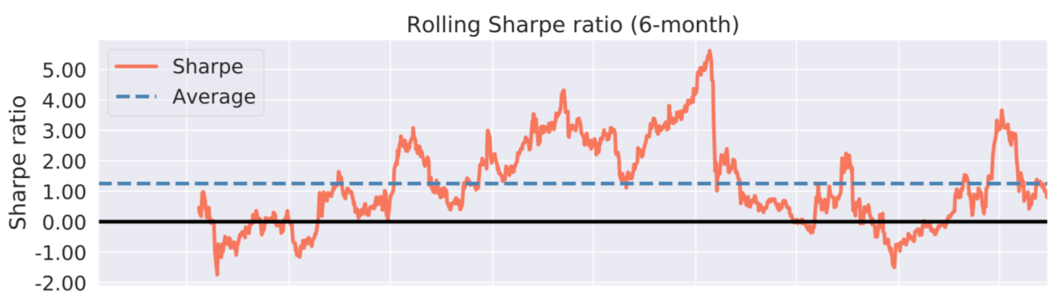


Figure 9: Rolling Sharpe Ratio for the Mean-CVaR Portfolio.

Drawdown Periods: Mean-CVaR

Among the main features of the MCVaR optimization is downside protection that it enables. This is generally seen in the Underwater plot, tracking the periods of drawdown from relative peaks in performance. For reference, if the portfolio had returns that were monotonically increasing, then there would be no drawdowns observed.

Drawdowns for Mean-CVaR

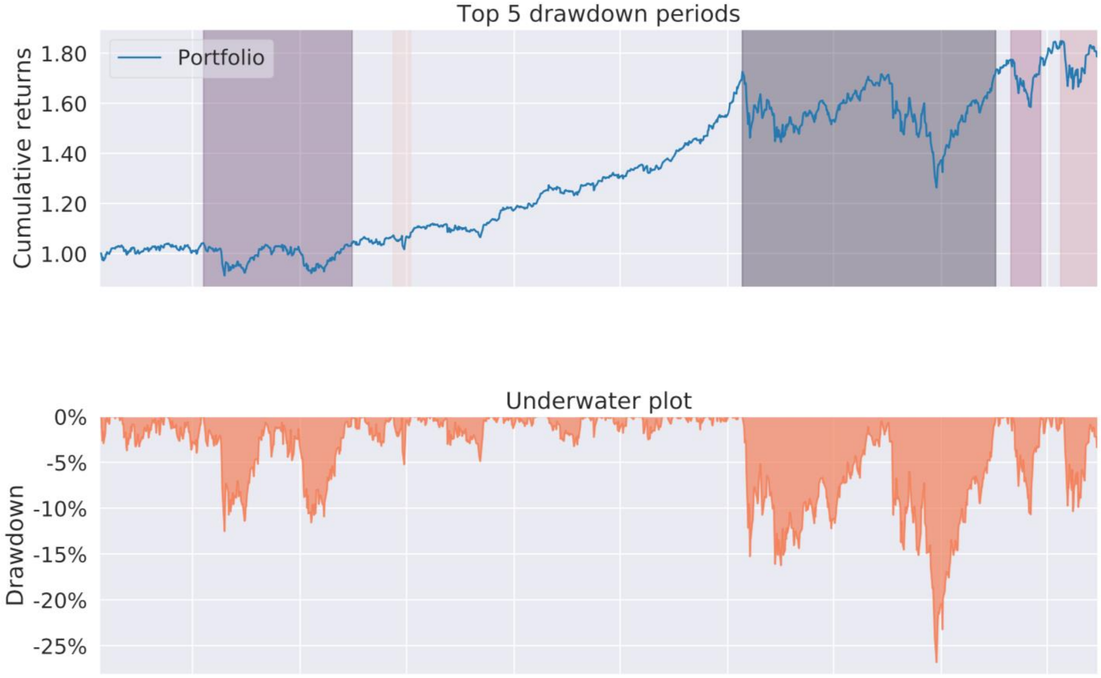


Figure 10: Drawdown in the Mean-CVaR Portfolio.

Drawdowns in the portfolio remained relatively muted, despite two large periods of continued loss towards the end of the backtest. It would not due the MCVaR Optimization scheme justice without showing the *comparative* performance to that of another portfolio. For this purpose, consider the same drawdown plots tracked for the Sharpe Optimization.

The following plot reveals just how successful the MCVaR portfolio was at reducing large downward swings. Whereas the Sharpe portfolio suffered greater drawdown throughout the beginning of the backtest, the MCVaR portfolio emerged relatively unscathed. It was this strength that allowed the MCVaR portfolio to dominate the Sharpe portfolio in terms of cumulative returns.

Drawdowns for Sharpe

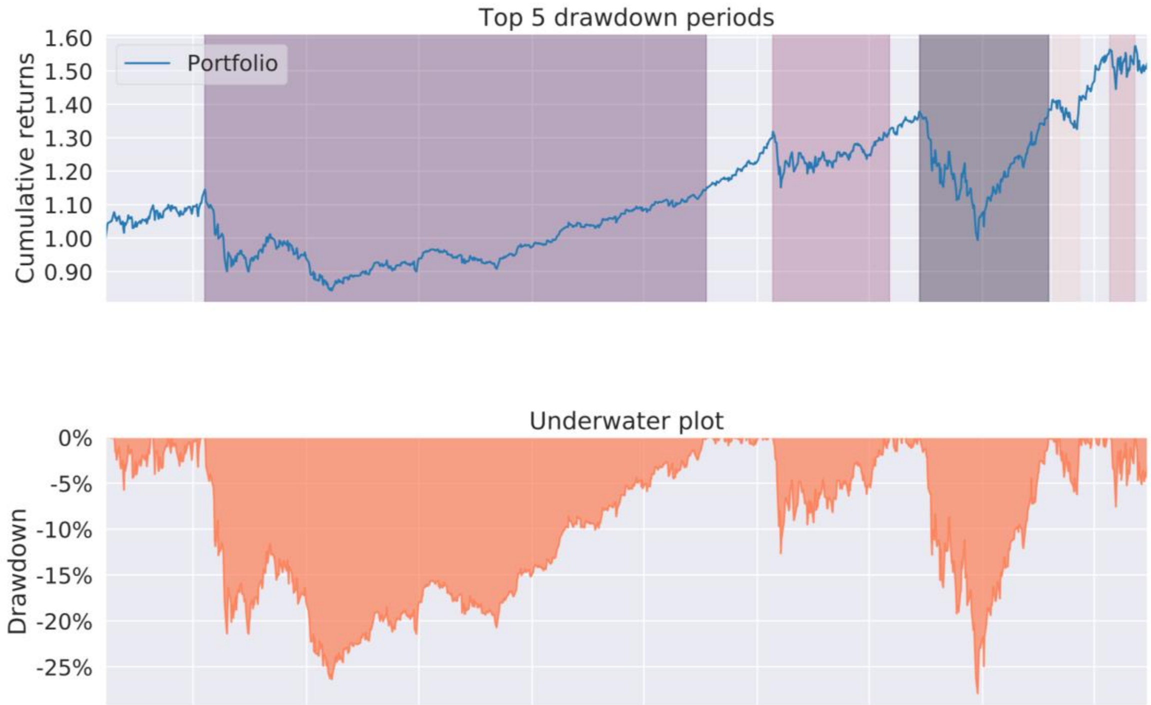


Figure 11: Drawdown in the Sharpe Portfolio.

Distributional Returns: Mean-CVaR

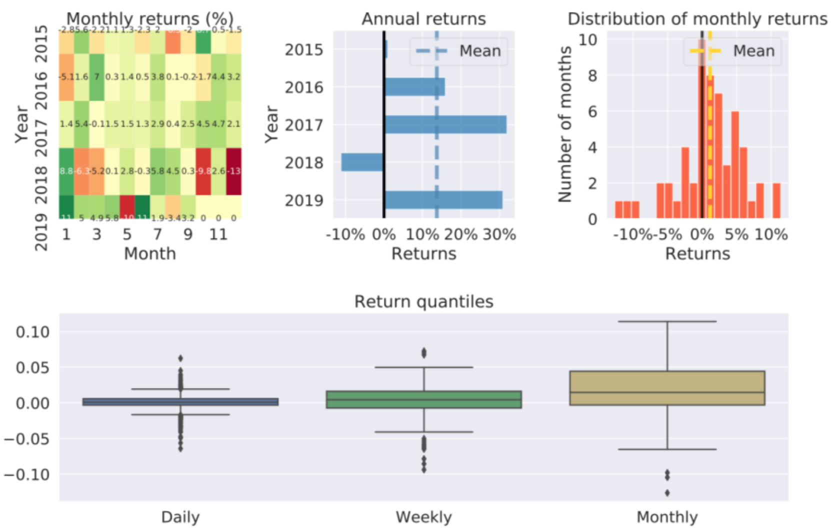


Figure 12: Distributional Characteristics of Returns for trailing end of backtest.

Finally, consider the distributional characteristics of the portfolio returns. From this view, it is clear that returns are not well-fitted to a normal distribution, with a left-skew noticed, thus favouring larger returns. This observation agrees well with

5.4.1.1 Design Considerations for an Optimization Scheme

When considering between what optimization scheme to follow through on, the content above provided a throughout justification. Tabled below is a synthesis of this discussion, where each salient feature is ranked across each alternative. Note that in some cases, the alternatives are given the same ranking, for instance, when it comes to the integration with returns estimates. A ranking of 1 is the lowest rank, whereas 3 is the highest. Therefore, the total row should be interpreted as an increasing measure, whereby the alternative with the highest column-wise ranking proves superior.

Table 5: Decision Matrix for Optimization Scheme

Feature / Model	MCVaR	Sharpe	Risk-Parity
Growth Characteristics	3	1	2
Mitigated Losss	3	1	2
Computation Ease in Implementation	1	3	2
Integration With Estimates	3	3	3
Total	**10**	8	9

Based on the validation, it should come as no surprise that the MCVaR portfolio comes out on top. The main limitation with the MCVaR portfolio is the computation might that is involved with generating this risk measure. This will be explained later on, and a method to solve CVaR analytically will be shown. However, this will involve imposing a normal distribution on portfolio returns.

5.4.2 Two-Period Horizon and Rebalancing Methodology

It is recognized that the largest source of parameter error comes from the estimated returns. For this reason, it was decided that to discount the forecasting inaccuracies it would be appropriate to limit consideration of the multi-period model to only two periods. In this sense, the portfolio remains primed to deal with the rebalance account trade-offs, while not sacrificing immediate returns in favour of noisy and uncertain future estimates. Note that the consideration of a two period model is akin to applying a strong (infinitely-so) discounting factors on estimates that extend beyond one forward looking principle. From this perspective

By restriction focus to two periods, there is the added benefit of simplicity. To a novice investor, the prospect of optimization a portfolio over 10 periods seems daunting - how could returns 10 years from now possibly alter my decision today? To this effect, a two-period implementation has a simple interpretation: the optimization is focused on the immediate decision, while ensuring that the portfolio is well-positioned to handle adverse and unexpected shocks. All investors understand that markets are uncertain, so a framework that deals with this uncertainty is clearly a benefit. However, extending the considering too far into the future lends itself to a degree of abstraction that may not be immediately obvious. In this sense, the focus is on providing the user a portfolio which they can make sense of.

Rebalancing for each saved portfolio will occur every 6 months, with back tested data performing also on a rolling basis.

5.4.3 Transaction and Holding Cost Modeling

Transaction costs are modelled using a model based on the bid-ask spread, period volatility and root volumes. Specifically, for each asset, the transaction costs are taken as:

$$\phi_t(Q) = \, Spread\ Cost \, + \, \alpha\, \sigma \sqrt{Q/V}$$

Where Spread cost is taken as the latest quoted bid-ask spread on the asset, alpha is a sensitivity parameter empirically set to scalar of order-one (ex. α=5), Q is share-lot size to be traded, σ is the period volatility and V is the period volume. From this perspective, it is clear to see how this framework lends itself to the multi-period approach. On a daily basis, the bid-ask spread is updated for each asset and so too are daily volatility and volumes, data which is readily available. When comes time for the optimization, the transaction costs act as an aversion penalty to large changes in portfolio asset holdings. It therefore can be interpreted as applying less of a penalty on highly liquid assets, where spreads are tight, volatilities are managed and volumes on the limit order book are large.

As it applied to holding costs, the spread cost was taken.

5.4.4 Risk Preferences, Cardinality and Diversity Considerations

In facilitating the optimization, there are several conditions that alter the performance, including the constraints and optimization coefficients. The risk preferences inform the coefficients on the risk penalty functions and the robust, cardinality and diversity constraint limit the set of feasible solutions. The risk preference, specifically the risk aversion levels, map-in directly from inputs and confidences of the investor. Additionally, both cardinality and diversity constraints are applied on the portfolio. These act, once again, as a function of the investor's risk levels. As it is applied in this project, a more risky investor is willing to accept a less-diversified portfolio with fewer asset restrictions (cardinalities). On the other hand, a prudent investor is one which focuses their portfolio on ones that spreads risks more evenly.

To handle the mapping of risk preferences to optimization constraints, there are three aspects of consideration:

1) **Time Horizon**: a longer time horizon means that the investor is better able to "ride-out-the-storm" and benefit from last-standing positive drift in equities. Generally speaking, the longer the time horizon, the greater affordance to risk an investor has.
2) **Target Returns**: is an obvious constituent that comprises the constraint level risk considerations. Note only does this function to robustness, it also gives a hint of insight into the risk-competence of the investor. A very aggressive risk target (ex. 15% in 6 months!) would require a very loose aversion to risk. As it pertains to robustness, constraint is formulated using an ellipsoidal uncertainty set with a

confidence level of 95%. The robustness helps to combat the noise and parameter uncertainty that associate the largest source of errors.

3) **User Assigned Risk Tolerance**: In what is a direct mapping of risk, a user who understands their own personal financial goals will take on a risk exposure that they are comfortable with.

Speaking now in terms of the implementation, the following pseudo code sheds light to how this mapping is accomplished. Please note that the assignment of risk into three categories is accomplished through the front-end facing questionnaire. The following logic demonstrates how this translates to the optimization parameters dealing with risk.

Constraint Level Risk Preference
>> Inputs(Investment Horizon, Aversion=(High, Med, Low), Init_dollars, Target_dollars, Market_Coefficient, Parameter Return and Covariances Estimates)

If No Horizon (user is just building wealth) Then Set Horizon = 10

Define a bath exposure, or diversity constraint, per asset of (MIN=0.05, MAX=0.20)

Define a base, confidence level, alpha = 0.05

Based on Target_dollars, determine:
return_target = (target_dollars / initial_dollars) ** (1 / (2 * horizon)) - 1

The “Safe” levels of return is the average of the period one and two estimates
safe_target = (ret_1 + ret_1) / 2

Initialize a Risk Multiplier, penalizing large risky positions and a Transaction Cost Multiplier, penalizing large portfolio fluctuations between periods one and two
Risk Mult = Market_Coefficient
Cost Mult = 1

Part 1: Horizon Analysis
If short term horizon (<= 1):
- Select for cardinality the 12 assets with the lowest variances, as per parameter covariance estimates
- Lower upper bound of exposure: exposures = (0.05, 0.15)
- Double the Risk Mult: risk_mul *= 2
- Reduce the Cost Mult by a Fourth: cost_mul*= 0.25 (why? The portfolio won't be needed in the long term!)

If medium term horizon (>1 and <=5 years)
- Select for cardinality the 12 assets with the highest sharpe ratios, that is, the best risk-adjusted performance offerings
- Keep the target exposures.
- Adjust Risk Mult by 0.75.
- Keep Cost Mult.

If long term horizon (>5 years)
- Select for cardinality the 12 assets with the highest expected returns, that is, those that will bring the highest returns

- Raise the upper bound on exposures and lower the lower bound on exposures: exposures = (0.02, 0.35)
- Adjust Risk Mult by a fourth: Risk Mult *= 0.25
- Double Cost Mult: Cost Mult *= 2 (why? The portfolio is being held for longer durations, so it is going to be more sensitives to transaction costs and rebalancing effects)

Part 2: Target Returns

If the return_target exceeds safe_target:

- The portfolio needs to be more aggressive, so halve the Risk Mul: Risk Mul *= 0.5

Else: no adjustments

Part 3: Stated Risk Aversion

Adjust Risk Mul directly on aversion, where Aversion=(High, Med, Low) <-> (1, 2, 3)

- risk_mul *= aversion
- *If Low risk, exclude any single-name stocks*

<< Return: (Risk Mul, Cost Mul), exposures, list(cardinality)

5.4.5 Robust Considerations

The parameter estimation yields estimates may still be noisy. To develop a framework to account for this uncertainty, consider the following robust optimization model presented. For this task, define an ellipsoidal uncertainty set

$$\mu_{true} \in U(\mu) = \{\mu_{true} \in R^n : (\mu_{true} - \mu)^T \Theta^{-1}(\mu_{true} - \mu) \le \varepsilon^2\}$$

Note that $\Theta \in R^{n\times n}$ represents the standard variance derived from the covariance matrix Q and $\varepsilon^2 \in R$ is the uncertainty tolerance of U. Specifically, $\varepsilon^2 := \chi^2_n(1 - \alpha)$, in which $\chi^2_n(1 - \alpha)$ is the inverse cumulative distribution function for a chi- squared distribution with n degrees of freedom. The degrees of freedom for this application will be 28, since there are 18 ETFs and 10 single name assets considered. As a baseline, the percentile estimate at a 95% confidence level. When optimizing under a robust framework, we penalize our expected returns by the amount that we expect the returns to vary around their true value.

The application of robust optimization with an ellipsoidal uncertainty set involves taking the difference between the portfolio's expected return and the penalty term shown above. These are applied on a constraint wise basis and help the optimization account for noisy return estimates.

5.4.5.1 Tracking and Enhancing a Portfolio

Whereas the previous content in this section focused on the portfolio building processing, this project also offers a portfolio tracing and enhancing service. The portfolio tracking service is simply the back-tested results of a user-inputted portfolio, while the portfolio enhancement service is a Sharpe-Optimization of a user-inputted portfolio.

Speaking specifically to the latter enhancement service, a Sharpe-Optimization is chosen because of the strong risk-adjusted returns performance it offers. Overall, the optimization scheme remains the same as what was previously described for MCVaR, with the only difference coming from the objective target.

What should be known is that the risk preferences maps to an aggressive investor in terms of exposures and diversity constraints.

Running through an example, consider a benchmark Portfolio consisting solely of stock in Apple. Over the past six months, Apple has performed exceptionally well, giving investors a 45% return!

Figure 13: 6 month cumulative returns for AAPL

Surely benchmarking against Apple is doomed for failure, and this indeed is the case with the result from the Sharpe Portfolio Enhancement. The following pie graph depicts the portfolio generated, under a cardinality constraint of 17, that is trying to enhance the portfolio consisting solely of shares of Apple:

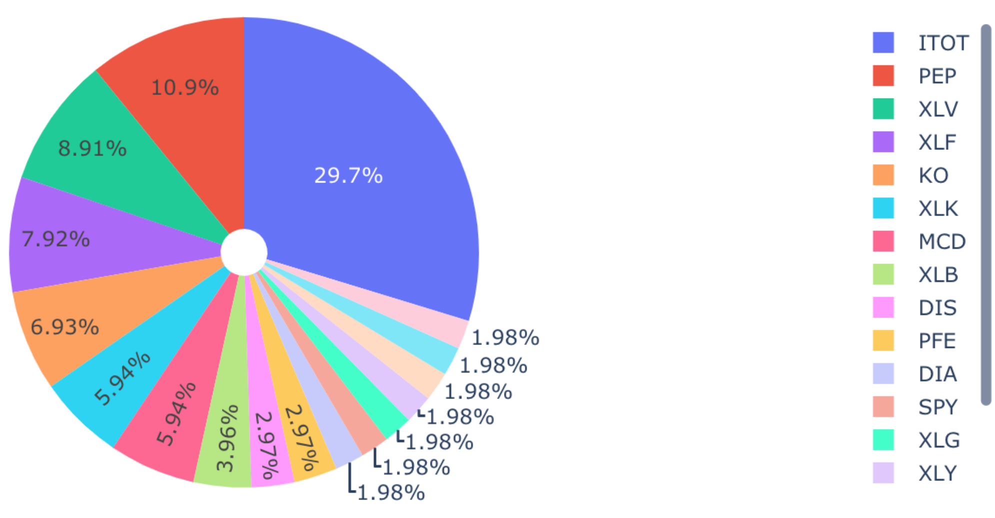


Figure 14: Enhanced Portfolio Optimized for Sharpe, benchmarked against Apple Shares.

Indeed, the ‘enhanced’ portfolio does not dominate Apple when it comes to returns over the last-six months:

Figure 15: Comparison of Cumulative Returns for the Sharpe-Optimized portfolio against Apple.

However, where the enhanced portfolio was designed to out-perform Apple was in regards to its Sharpe-Ratio, now shown in the figure below:

Figure 16: Comparison of Rolling Sharpe Ratio for the Sharpe-Optimized portfolio against Apple.

While it is clear that the enhanced portfolio does not dominate Apple by any means (Apple in its own right offers strong risk-adjusted returns), the comparison shifts in favour of the enhanced portfolio. Over the course of the backtest period, the enhanced portfolio tracked very closely with Apple when it comes to the Sharpe ratio measure, in fact beating Apple by a decent margin at the very end. This ultimately demonstrates the power of the Sharpe-enhanced portfolio, in agreement with the disciplined mission that the project attempts to achieve. The enhanced portfolio offers risk-adjusted characteristics comparable to that of a top performing stock like Apple.

5.5 Enhancements and Limitations

5.5.1 Noise in NLP Estimates

After testing the portfolio optimization framework using views generated by NLP, the returns were not very promising. While the method itself is promising, the limitations most likely lie with the data. Tiingo provides continuously updating financial news, however it only gives a description of the article instead of the entire document. This is still more information than other APIs, but just a short description limits the information the GRU can learn from. Furthermore, if the description itself does not accurately depict the content of the entire article, this can add undesirable noise in the training procedure. Due to the subpar performance of the NLP model, the Time Series model was implemented into the optimization framework which yielded more positive results.

Despite the underwhelming results of the NLP model, future iterations may be able to outperform the Time-Series model. Using an API that is able to provide the entire article text may improve the results of the GRU. Furthermore, since the article text contains a lot of latent information that may not be encoded in the historical price data, this additional data source could greatly outperform a Time-Series price analysis.

5.5.2 Period Calibration of Regime Switching

To alleviate the computational burden, the regime-switching model was calibrated using 10 years of historical data. On a 6 month rolling rebalancing basis, this means that the model is seeing new data for each phase in the updating of parameter estimates. This design choice was given so as to ensure that the computational burden was not too expense. The following summary shows the average compute time for various time periods under the Expectations Maximization algorithm:

Table 6: Summary of computation expense running the Regime Switching Model:

Backtest length (Years)	Average Processing Time (s)
1	12.067484
3	62.158764
5	150.730198
10	285.042863
15	532.35231

Additionally, another constraint beyond the 15 years was the limited price history for some of the tracked ETFs. This was why testing was capped at 15 years.

1 year

	BULL	BEAR
BULL	0.71973	0.28028
BEAR	0.76539	0.23461

3 years

	BULL	BEAR
BULL	0.7402	0.2598
BEAR	0.828177	0.171823

5 years

	BULL	BEAR
BULL	0.71973	0.28028
BEAR	0.83462	0.16538

10 years

	BULL	BEAR
BULL	0.895663	0.104337
BEAR	0.642359	0.357641

The trade-off is clear: a longer period of training leads to a better calibrated regime model, one that is better able to predict regime states and react to transitions. Judging by the transition matrix presented at the 10-year mark, this accurately captures market volatility from the global financial crisis. This is why the probabilities of staying in a BEAR state remain higher. Over a longer period, one would expect the probabilities of being in a BULL state to approach 1, given the overall upward trend of markets.

This observation is supported by the model 'seeing' a longer range of possible market returns. When this period is truncated, or taken too short, then it is likely that the model is under-trained. The impact of the under-trained model is apparent in its shortcomings in reacting to different regimes. Referring to the smoothed probability plot, the regime-switching model failed to capture an extended period of market volatility during the latter months of 2018. While this period did not correspond to a BEAR state per say, the model confidently predicted a BULL state. In hindsight this may have been correct, given that record-high market returns were subsequently achieved only months after the troubled period; however, it remains puzzling that the model did not record a then probability of a BEAR state.

5.5.2.1 Design Considerations for Period Calibration of Regime Switching Model

Table 7: Design Matrix for period calibrations.

Feature / Model	1 Year	3 Years	5 Years	10 Years	> 10 Years
Compute Time	5	4	3	2	1
Accuracy	1	2	3	4	5
Capture Price History of Assets Tracked	5	5	5	5	0
Intuitive Results	1	2	3	4	5
Total	12	13	14	**15**	11

5.5.3 Normality of Returns

One of the implications of using the Black-Litterman is the imposition of a Normal distribution on asset returns. From the distributional plots of monthly returns over the backtest period, it is clear that there is some skew presented. While the normality of returns leads to a closed-form solution for CVAR, shown below, it ultimately fails to capture the true nature of asset returns from a distribution perspective.

When asset returns $r \sim N(\mu, \Sigma)$, then the portfolio returns are also normal with $r_p \sim N(\mu^T w, w^T \Sigma w)$. From the interpretation that CVaR is an average of VaR values over it's support, the closed-form analytical solution is arrived at:

$$CVaR = -\mu^T w + \frac{\phi(\Phi^{-1}(\alpha))}{\alpha} \sqrt{w^T \Sigma w}$$

Where $\phi(.)$ represents the probability density function for a standard normal, $\Phi^{-1}(.)$ is the quantile function for the standard normal, and $\alpha$is the confidence level. It is clear that from this perspective, computing CVaR is quite simple and becomes similar to that of an MVO problem. Indeed, when running a comparison on the back-tested results between CVaR and MVO, the cumulative returns are nearly identical. Note that the backtest period runs 5 years, starting on 2014-11-30 and ending 2019-11-30.

Figure 17: Comparison of MCVaR Portfolio Optimization with MVO Portfolio Optimization

The normality limitation can be overcome by using a generalized Bayesian model rather than a Black-Litterman model. The Black-Litterman model in and of itself is a Bayseian model, but it one that uses a Normal distribution for both the prior and the likelihood estimates. As initially proposed, a Copula Opinion Pooling approach can be adopted with an underlying T-Distribution to augment views as accomplished in this model. The reason that this was not implemented in this project was due to technical difficulties in the implementation as well as the computation expense. The method of Copula Opinion Pooling involves sampling from a large panel of historical results to generate copulas. As the computational burden may prove to be too expensive, this method was avoided.

5.5.4 Expansion of the Universe of Assets

Once again limited by the inclusion of curated assets, a better optimization service would be able to pick from a large universe of assets. This is to ensure that the portfolio analytics and optimization tracks the best suited assets for a given risk tolerance. Right now, given a simple mapping of risk tolerances, the generated portfolio's do not differ too much from one another. The consistency in the generated assets ultimately comes from the reduced asset universe. Potentially increasing the consideration of assets would open the door for more custom portfolios that accurately represent given risk levels.

5.5.5 Modular Objective Targets

Initially proposed was the idea of using modular objective targets, rather than prescribing a MCVaR Optimization to all investments. Specifically, for an investor who is more suited towards aggressive risk targets, without much consideration to drawdowns, the preferred optimization service is arguably a Sharpe-based target. In this scenario, rather than putting consideration towards minimizing the drawdowns with respect to target returns, the portfolio generated would offer the greatest chance at achieving strong returns. While this runs contrary to what the backtest results indicated, where a MCVaR optimization beat out the Sharpe Optimization, this trend is not expected to continue during periods of strong market growth (a BULL state).

6. Front-End Design

The front-end uses the standard HTML, CSS and JavaScript technologies accompanied with Jinja2 templating engine. The user interface focuses on a minimalistic and modern design that is accessible to both advanced and novice users and easy to use.

6.1 Summary and Overview

The front-end considers three major design factors:

- **Aesthetics:** studies have shown that people form a first impression within a tenth of a second, giving only time for the user to visually experience the website before forming an opinion (Todorov & Willis, 2006).
- **Accessibility**: the goal of the website is to be accessible to both the novice and financially literate user, thereby reaching a larger target market, as stated in the Mission Statement.
- **Ease of Use:** the power of Equites comes from the functionality and backend and a difficult to use interface would distract the user from such features.

Upon entering the webpage the user is brought to the home page, at which they must login or register to access the rest of the website. The user is then able to navigate and use the features in a flow as seen below.

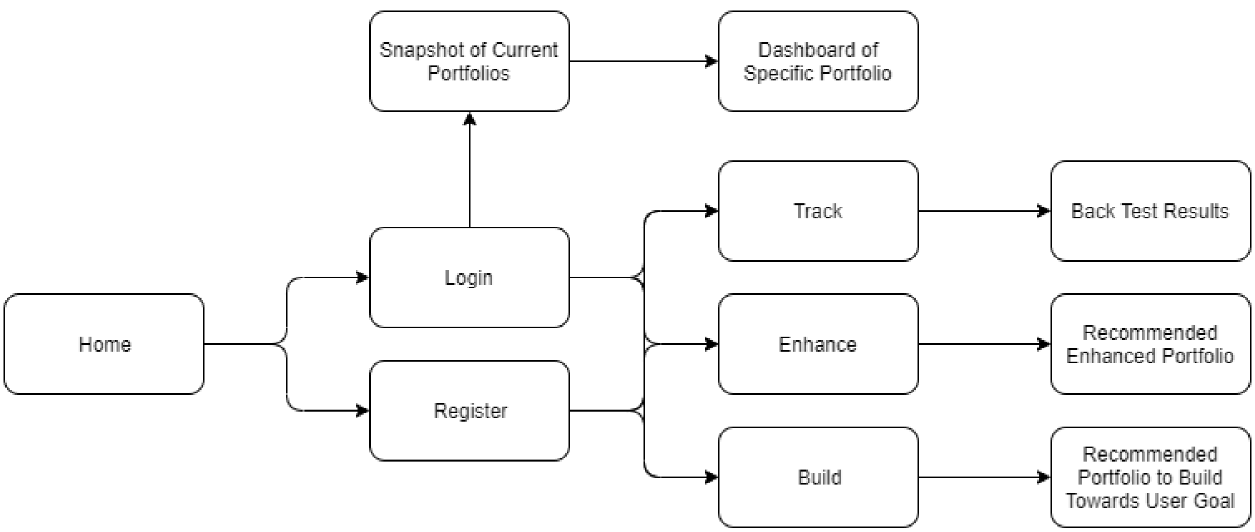


Figure 18: Diagram of User Flow

6.2 User Interface

The goal of the user facing aspect of the application is to provide an easy to understand and aesthetically pleasing experience for the user. This will be done through a design that guides the user through the application by focusing on logical interface guided steps with a minimalistic appearance rather than bombarding the user with questions and details all at once.

6.1.1 User Experience

6.1.1.1 Home Page

Our logo is up-centered in the homepage with a scroll-down button. Users can choose from three options to “Track”, “Enhance” or “Build” their portfolios by clicking “Start” button. Brief introductions of each option are given above the button. For “Track”, users are enabled to back test their own portfolios; For “Enhance”, an enhanced portfolio will be created for users. For “Build”, users will be provided personalized portfolios according to their specific needs and goals.

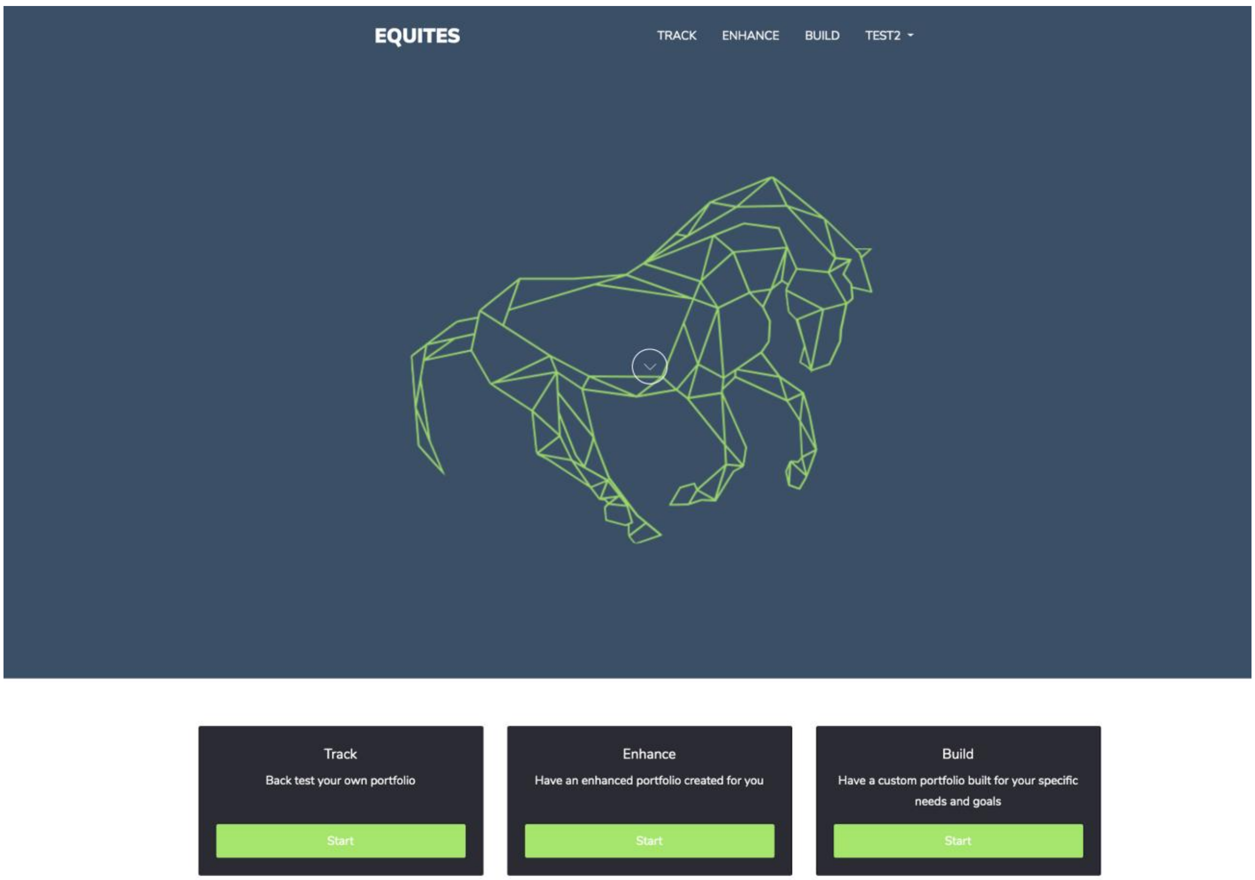


Figure 19: Display of the Home Page

6.1.1.2 Sign-up Page

To create an account, users need to type in a username and their email address. Password and repeated password are also required.

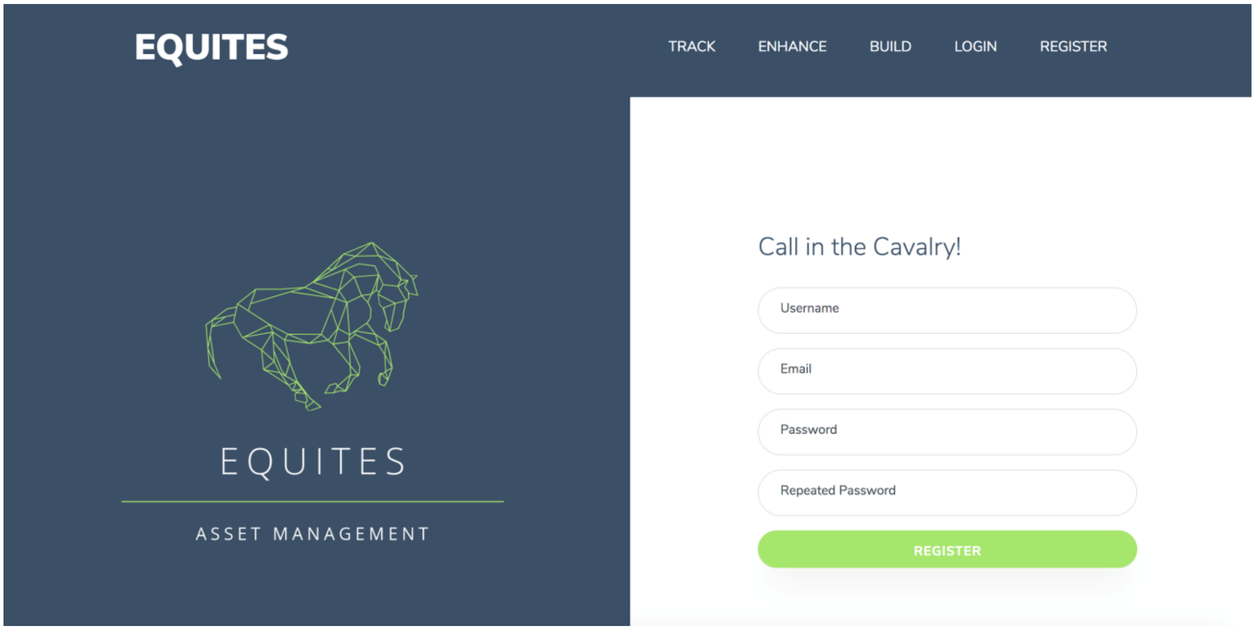


Figure 20: Display of Signup Page

6.1.1.3 Log-in Page

In the login page, users are required to type in the correct username and password. Option to remember password is offered. There is also a link to register below the button "sign-in" in case users don't have an account.

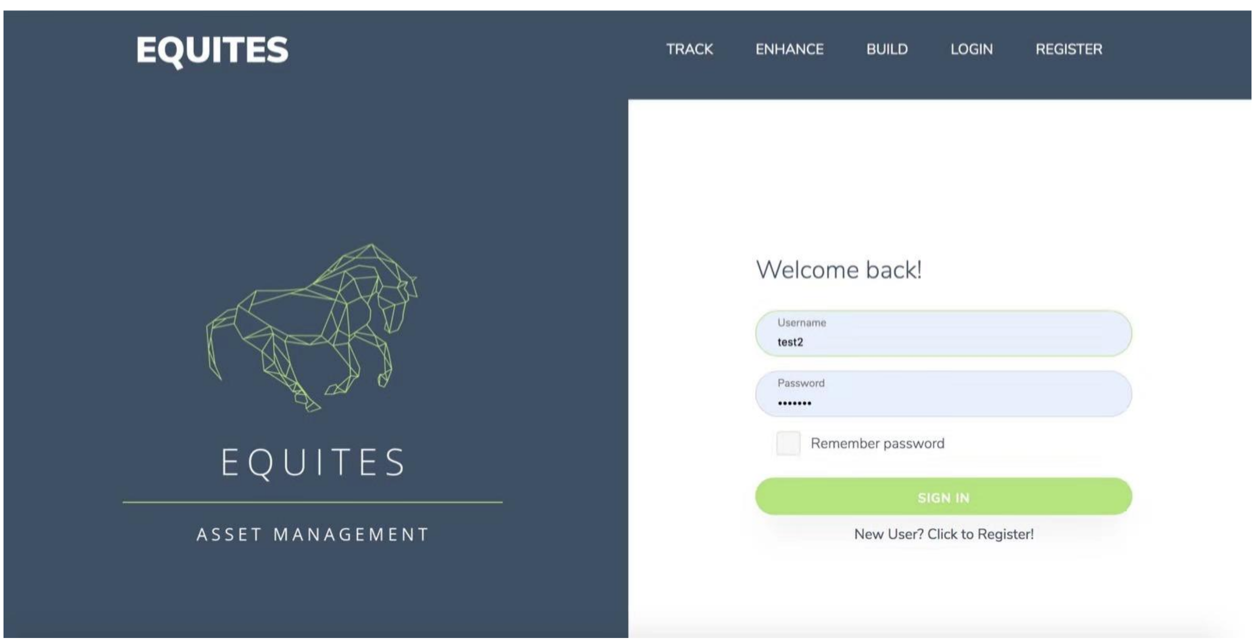


Figure 21: Login Page

6.1.1.4 Track Option

On the page of Track Option, users are required to type-in the “Start Date” and “End Date” and then “Ticker” and “Dollar Amount” for the portfolio they hoping to track. Users can easily delete the components by the red “Delete” buttons right after each component or add a new one through the green “Add New” button.

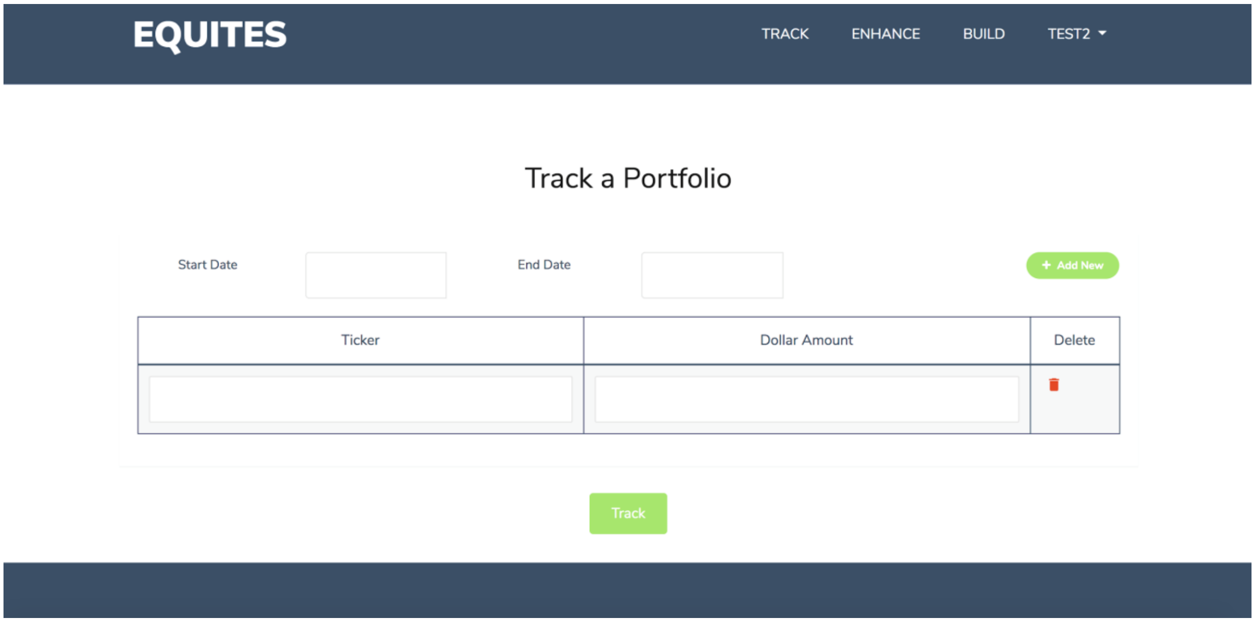
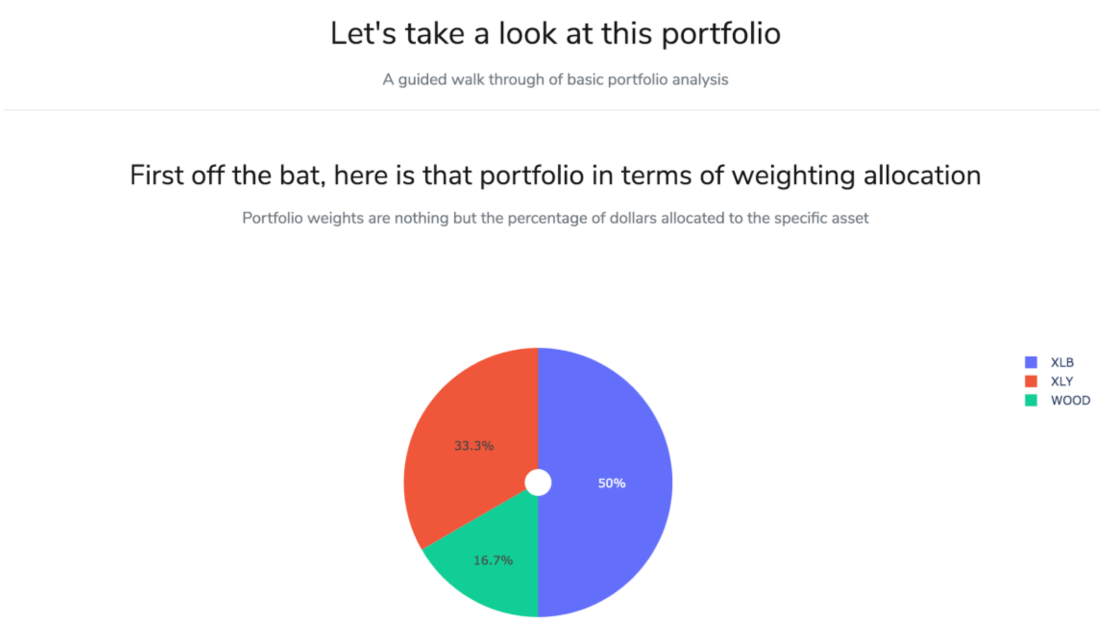


Figure 22: Track Option User input table

After clicking “Submit” button, a portfolio allocation will be generated in a pie-chart on the same page below users’ input-table as well as the total period returns, minimum portfolio value and maximum portfolio value. The change of return according to date will also be demonstrated in a line chart.



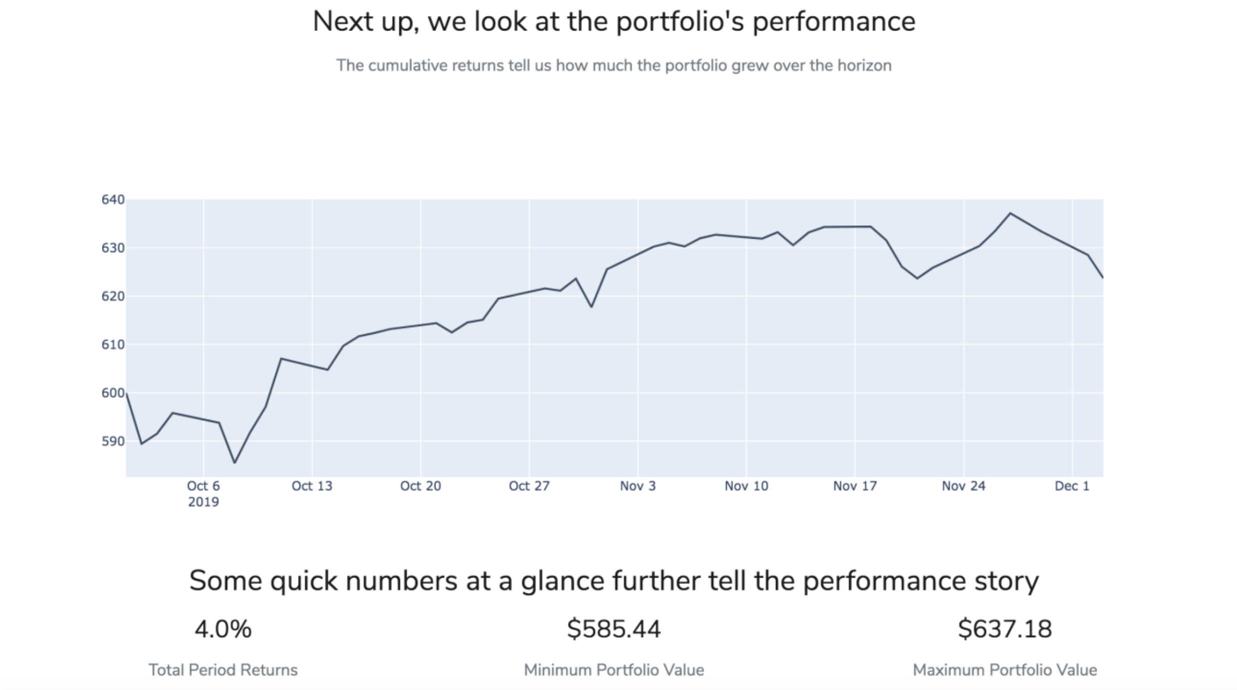
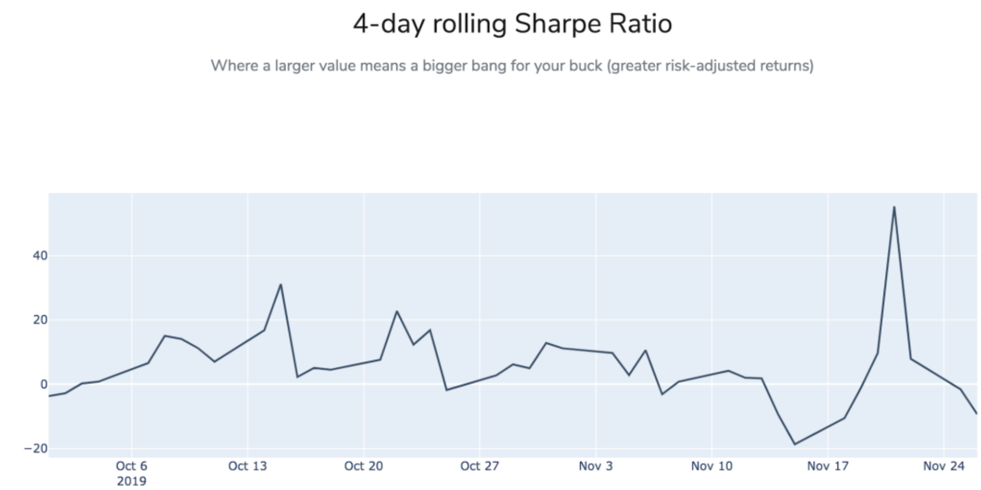
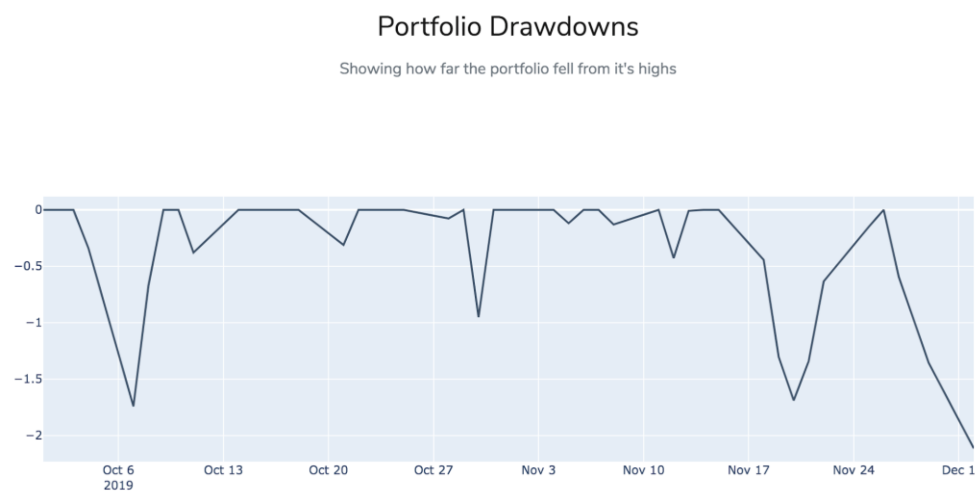


Figure 23: Track Option Result1

Rolling volatility, sharp performance and drawdowns will be displayed if users slide the button under prompt "I would like to see more detailed statistics on the portfolio"





Net drawdown in %	Peak date	Valley date	Recovery date	Duration
2.1135504117153197	2019-11-27T00:00:00	2019-12-03T00:00:00	null	null
1.7415837168097799	2019-10-04T00:00:00	2019-10-08T00:00:00	2019-10-10T00:00:00	5
1.6892700317040037	2019-11-18T00:00:00	2019-11-21T00:00:00	2019-11-27T00:00:00	8
0.951276180453298	2019-10-30T00:00:00	2019-10-31T00:00:00	2019-11-01T00:00:00	3
0.4277880521257905	2019-11-12T00:00:00	2019-11-13T00:00:00	2019-11-15T00:00:00	4
0.37872673112295147	2019-10-11T00:00:00	2019-10-14T00:00:00	2019-10-15T00:00:00	3
0.3116805233342089	2019-10-21T00:00:00	2019-10-22T00:00:00	2019-10-23T00:00:00	3
0.13020756180114795	2019-11-08T00:00:00	2019-11-11T00:00:00	2019-11-12T00:00:00	3
0.11927624031992	2019-11-05T00:00:00	2019-11-06T00:00:00	2019-11-07T00:00:00	3
0.07672322811399873	2019-10-28T00:00:00	2019-10-29T00:00:00	2019-10-30T00:00:00	3

Figure 24: Track Option Result2

6.1.1.5 Enhance Option

The page of “Enhance” option has a consistent format with that of “Track” option. Compared to “Track” option, users are only required to type in “Ticker” and “Dollar Amount” for each component of the portfolio they hoping to enhance. A pie-chart showing the optimal allocation will be demonstrated after users submitting their portfolio. After pie-chart, there displays

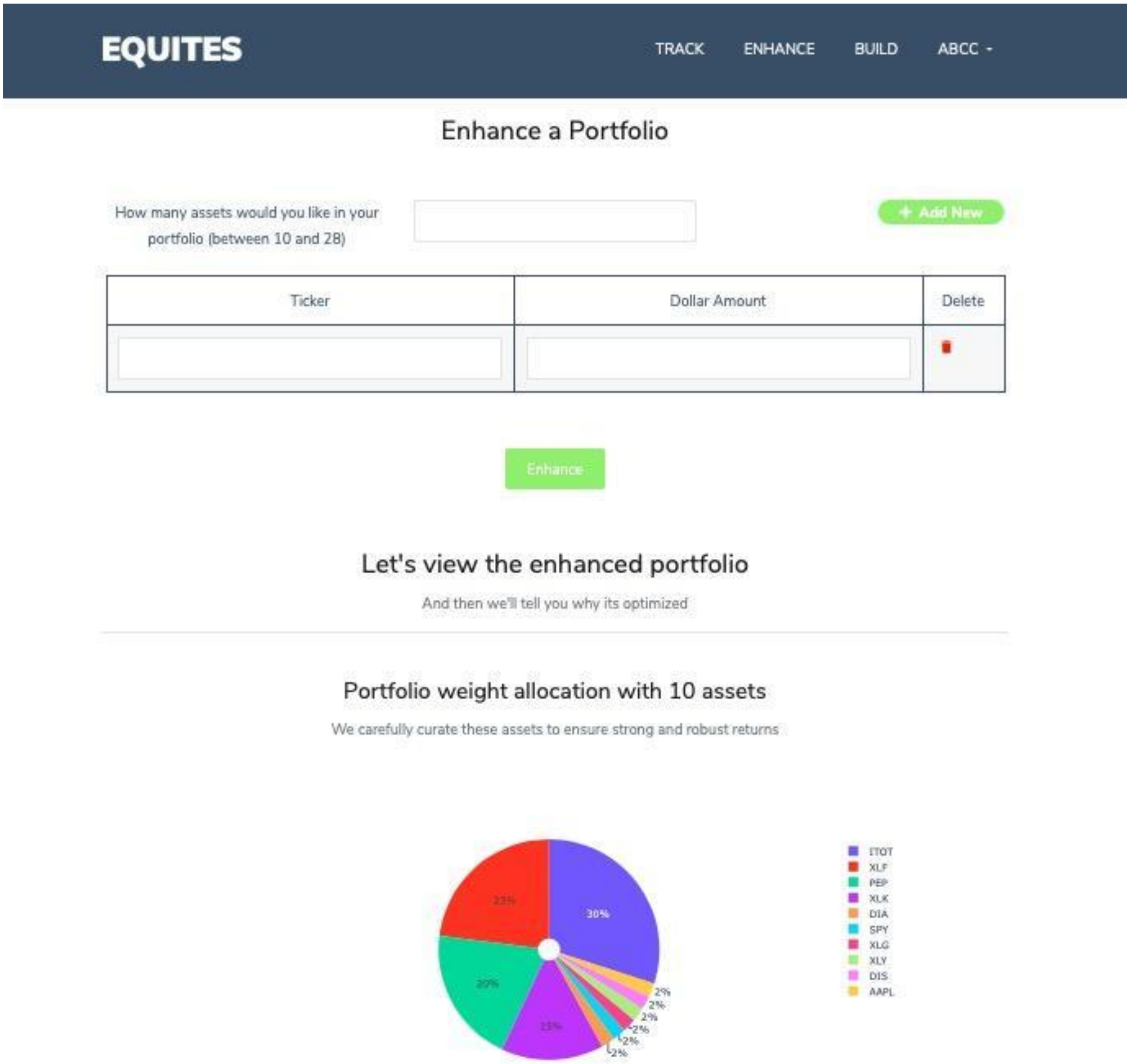


Figure 25: Enhance Option

6.1.1.6 Build Option

Three options under “Build” is offered for users to choose their investment purpose from “Build Wealth”, “Fund a Large Purchase”,“Retire”.For each option, users need to answer a few questions to continue. After the questionnaire, personalized results will be generated for the users. Users can view the asset allocation, backtested portfolio performance and risk profile. Risk profile includes portfolio volatility, sharpe ratio and drawdowns in the past six months. When the portfolio is generated, users can save it with a name or edit it to rebuild a portfolio.

6.1.1.6.1 Questionnaire for “Build Wealth”

To “Build Wealth”, users are asked “How much are you going to invest”,“What is your current annual income”, and as well as the risk appetite. An interactive return graph is used to demonstrate users a basic knowledge of relation between risk and return. Users can use the slider to choose a level of risk that they are comfortable with.

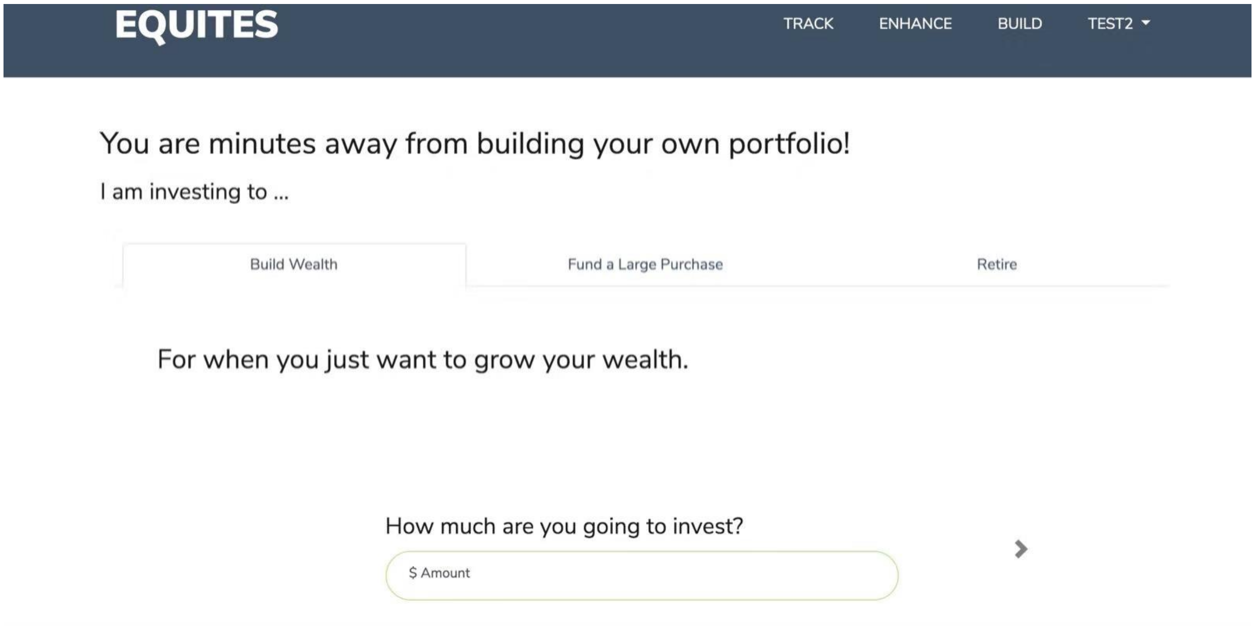


Figure 26 : Build Wealth - Questionnaire page1

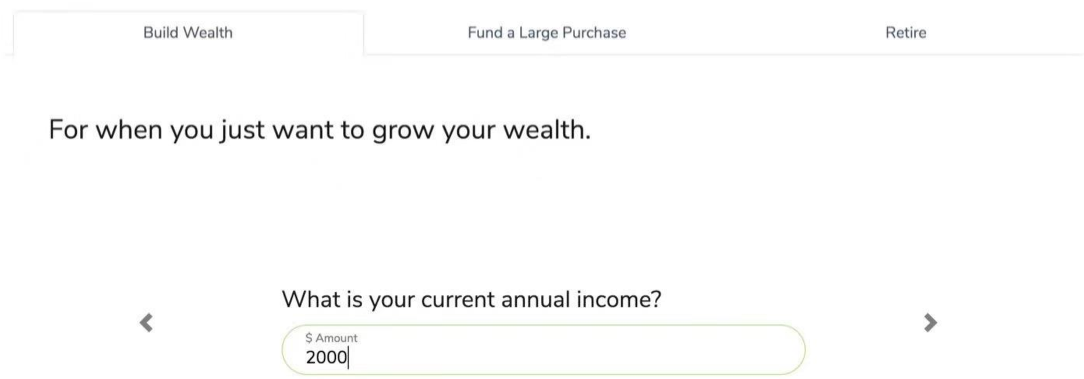


Figure 27: Build Wealth -Questionnaire page2

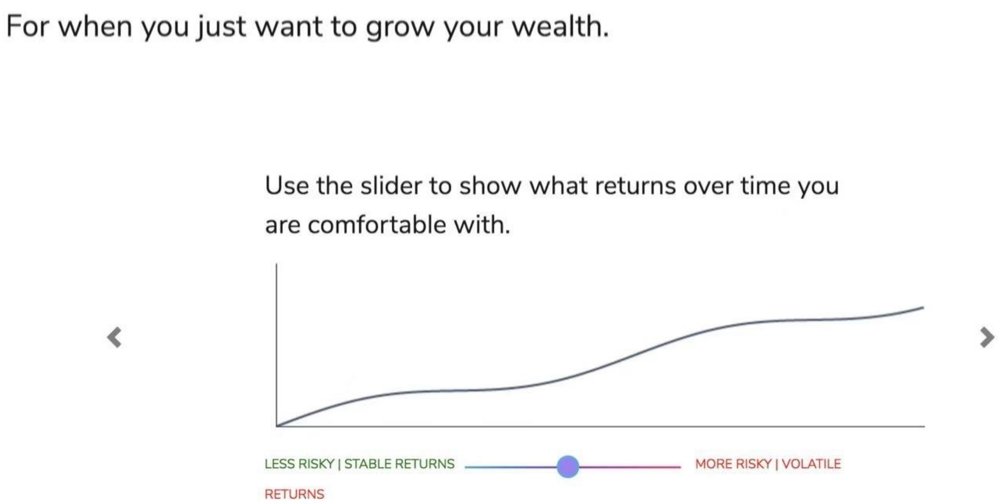
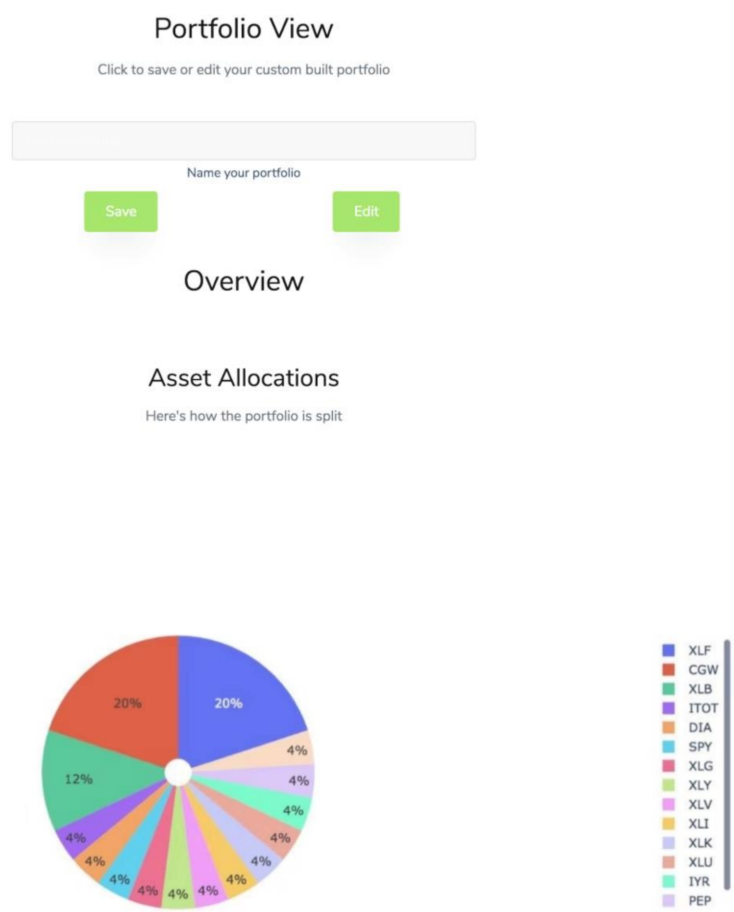


Figure 28: Build Wealth - Questionnaire page3



Backtested Portfolio Performance

Review what your returns could have been over the last few months

Risk Profile

Portfolio Volatility

Here is how stable returns are

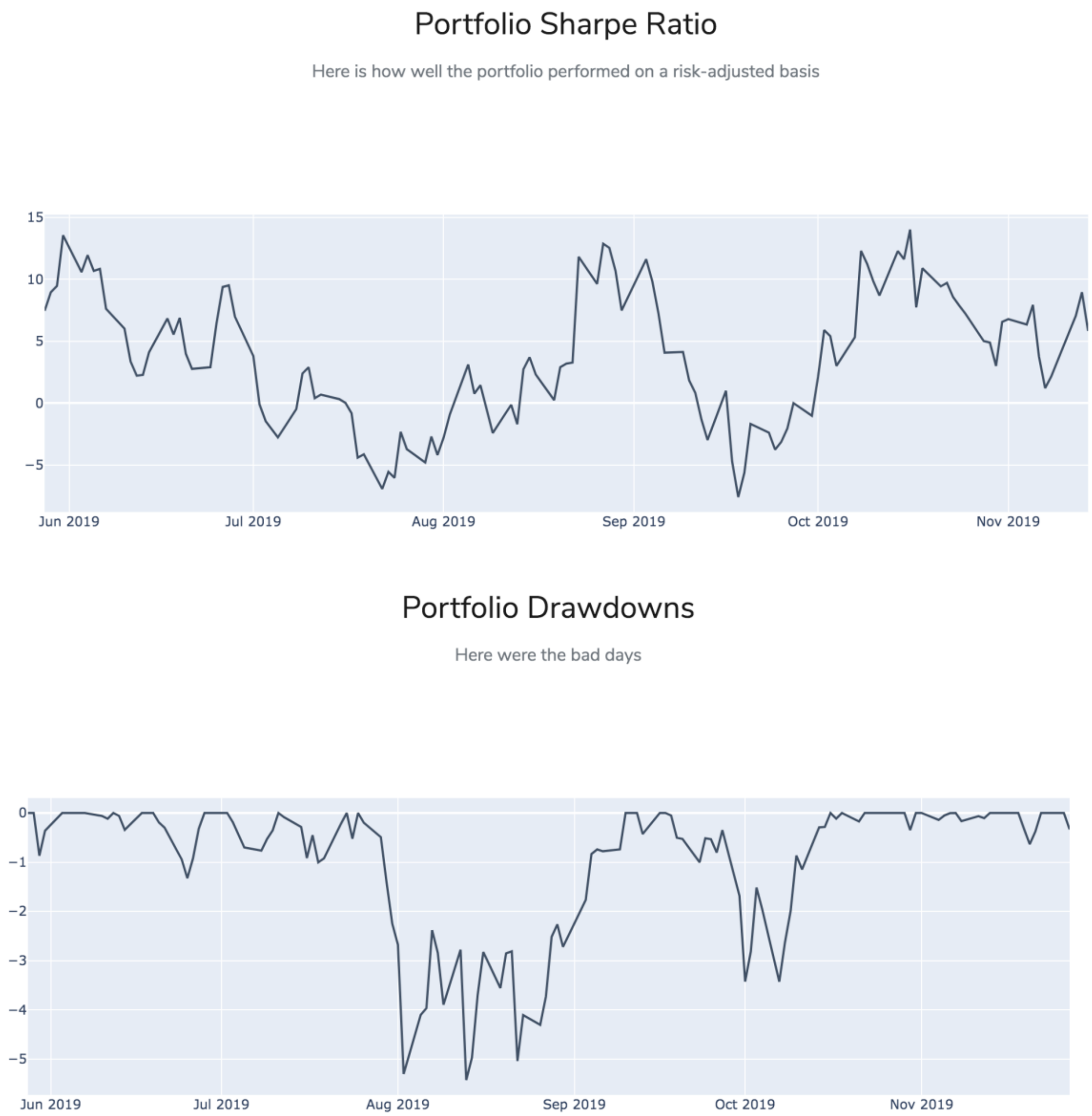


Figure 29 : Build Wealth Result Page

6.1.1.6.2 Questionnaire for "Fund a Large Purchase"

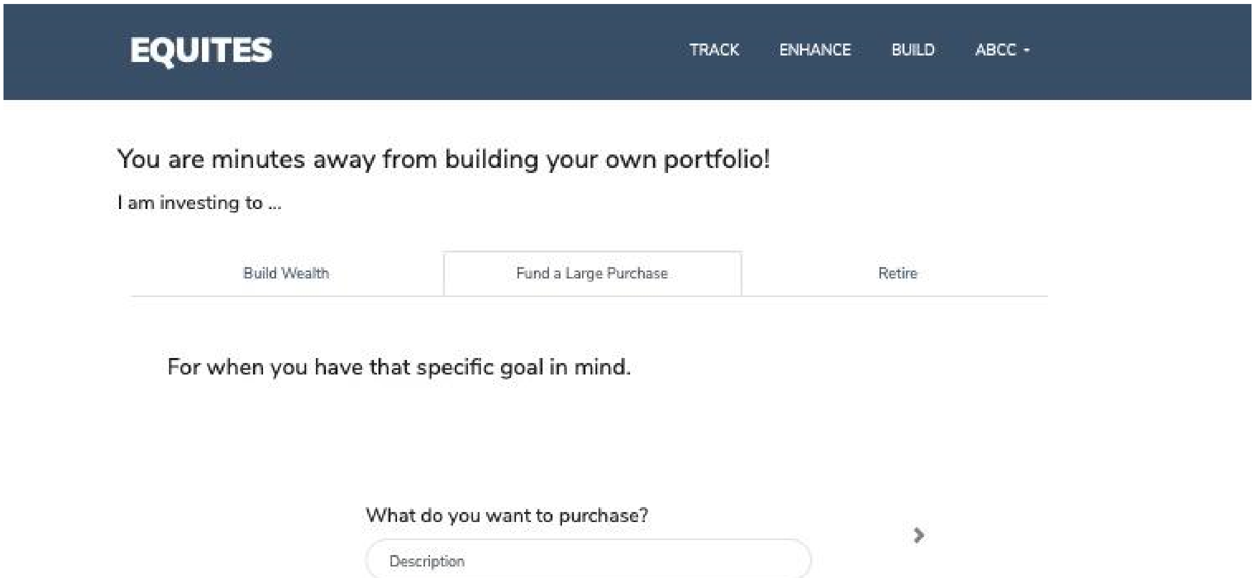


Figure 30 : Large Purchase- Questionnaire page1

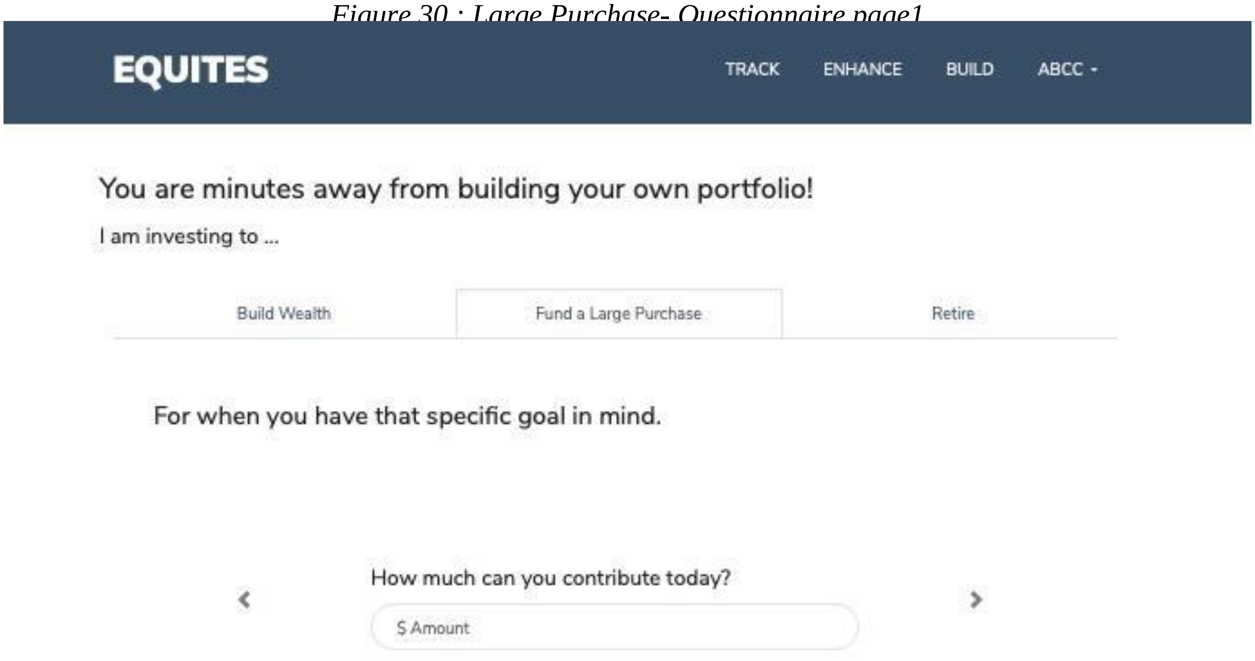


Figure 31 : Large Purchase- Questionnaire page2

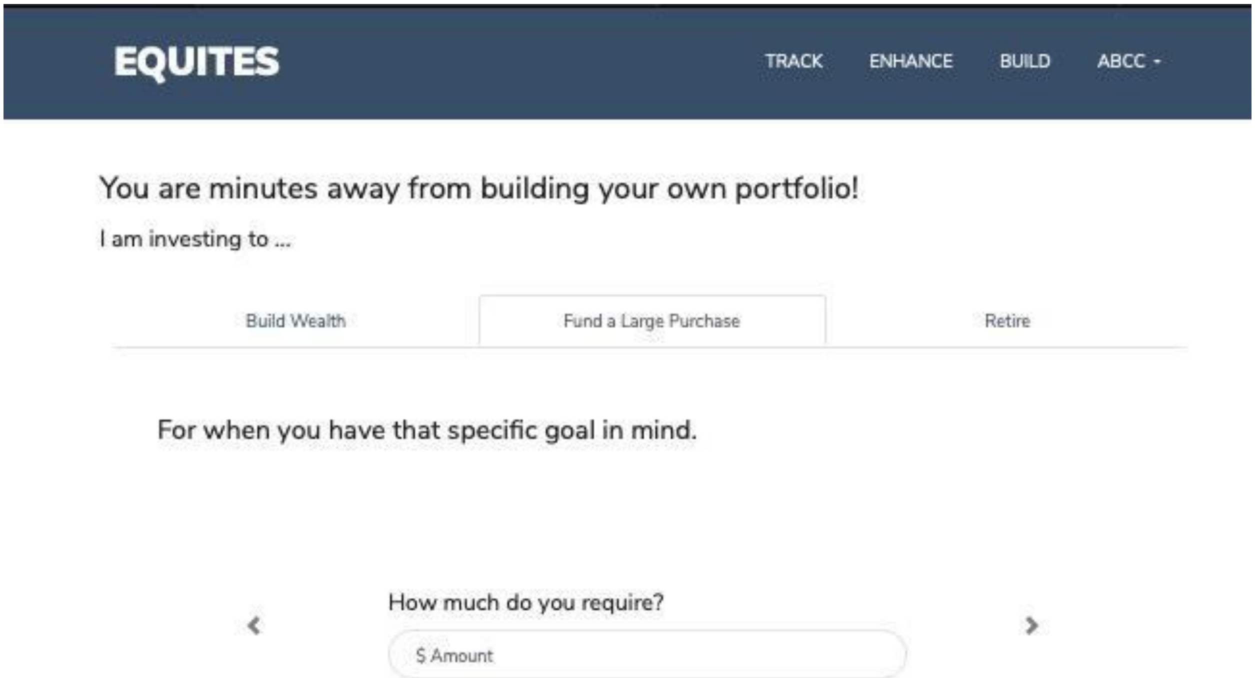


Figure 32 : Large Purchase- Questionnaire page3

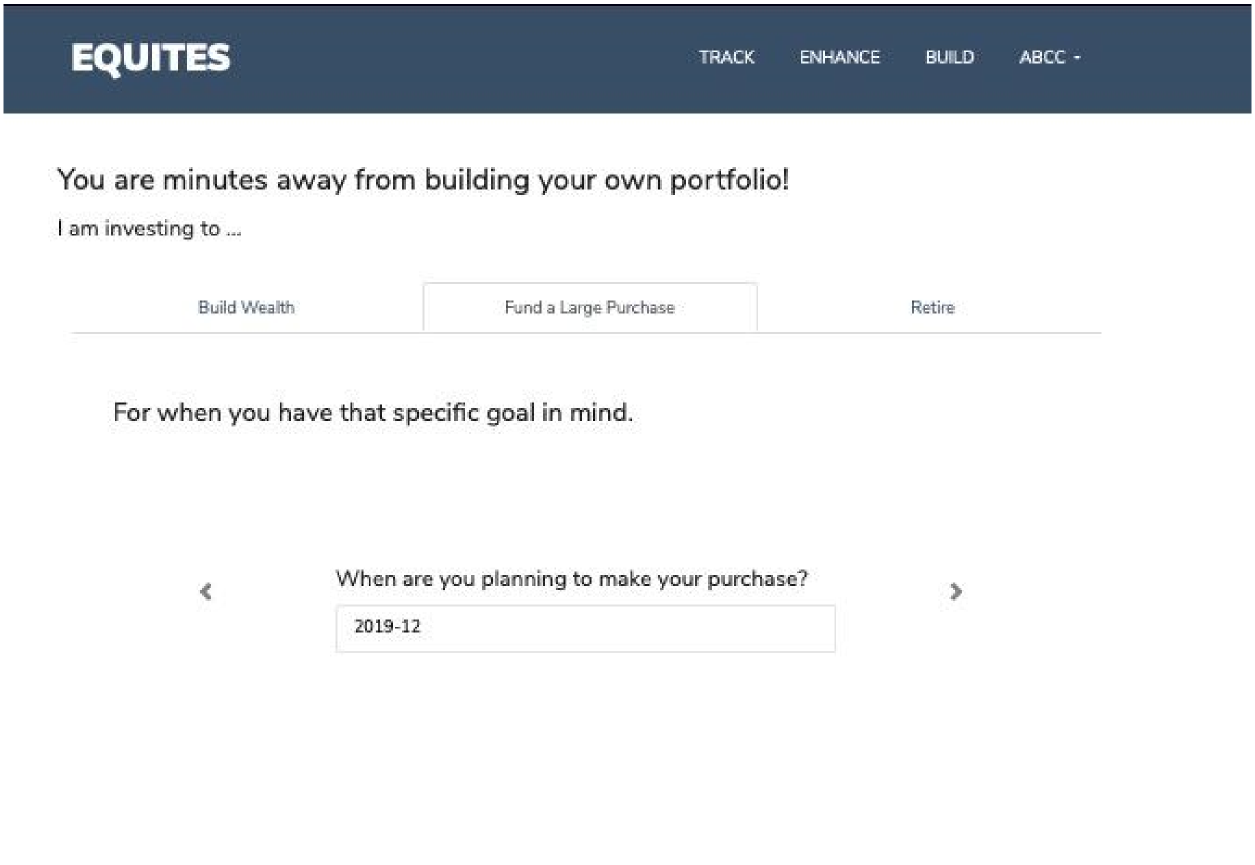


Figure 33 : Large Purchase- Questionnaire page4

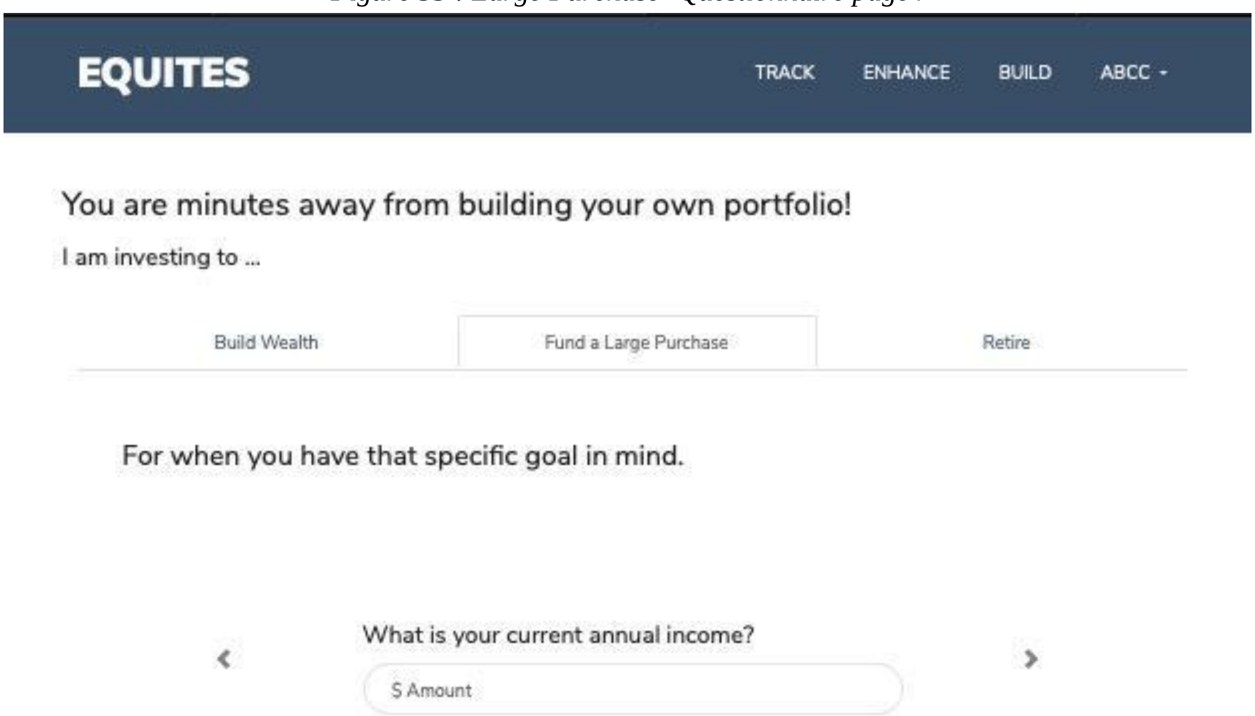


Figure 34 : Large Purchase- Questionnaire page5

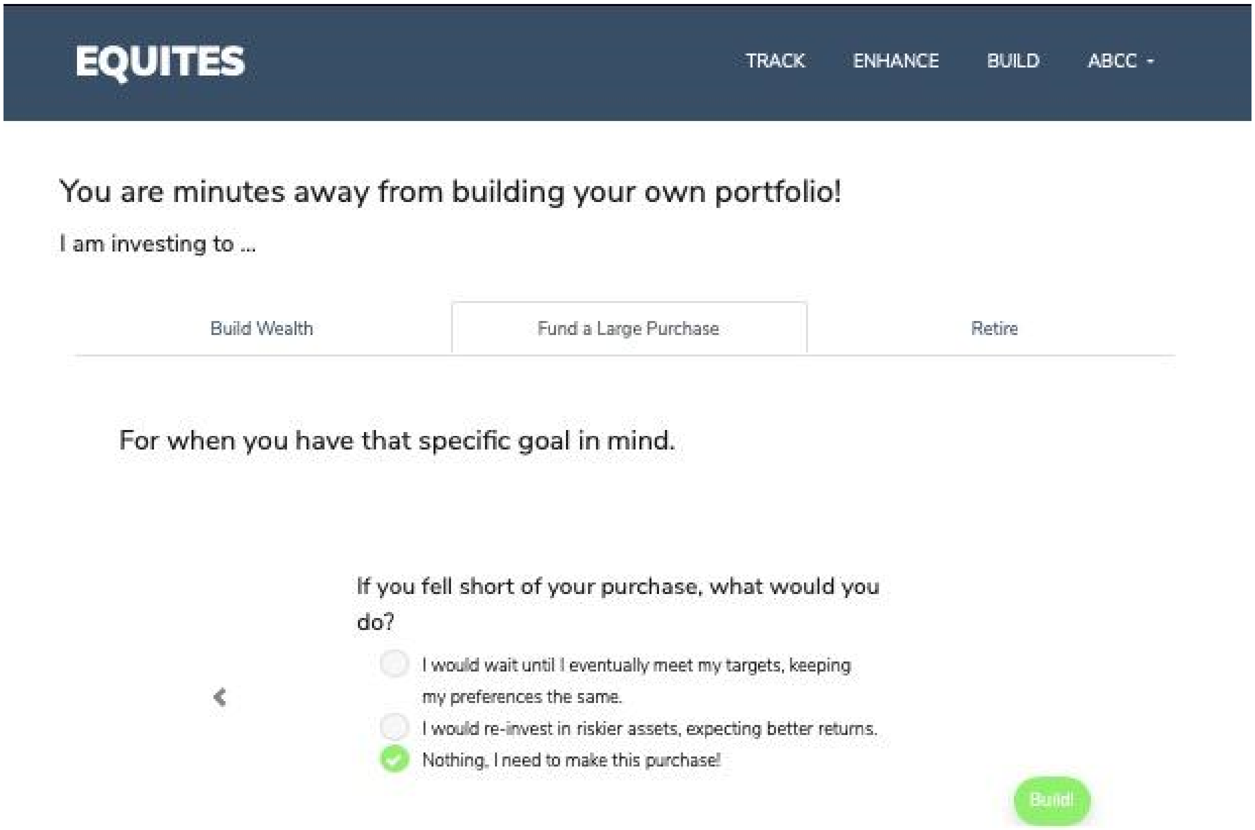


Figure 35 : Large Purchase- Questionnaire page6

6.1.1.6.3 Questionnaire for “Retire”

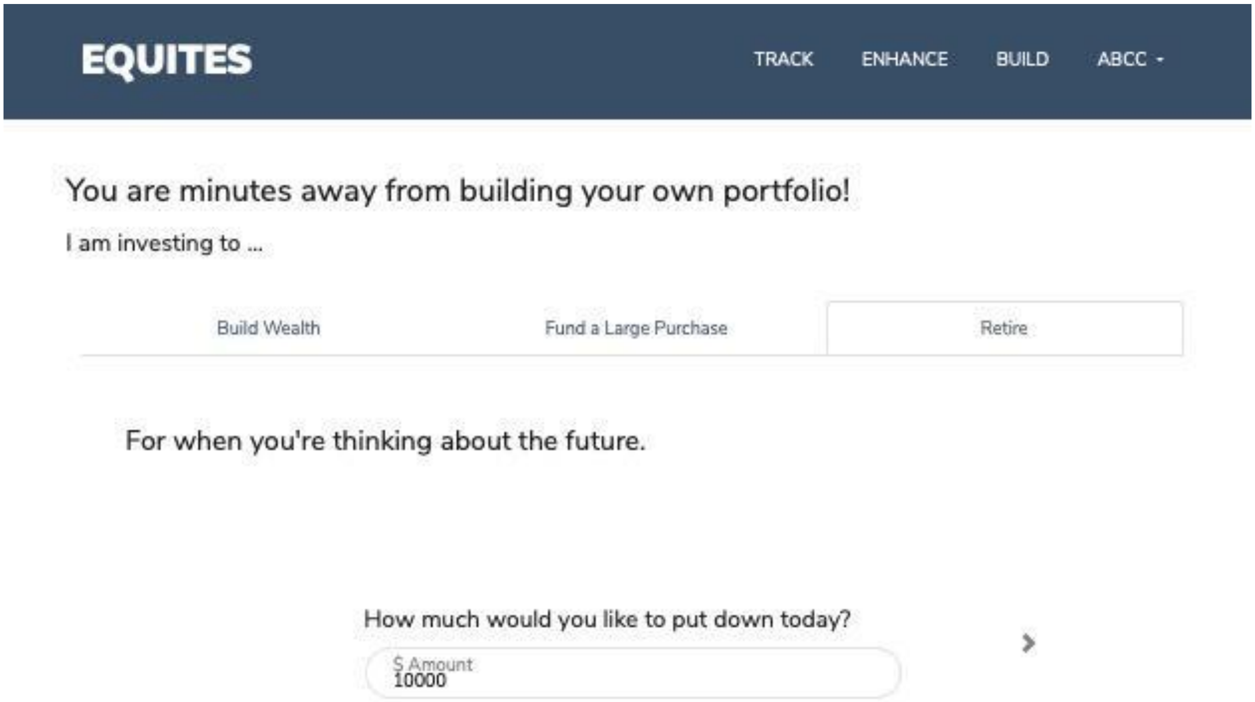


Figure 36 : Retire- Questionnaire page1

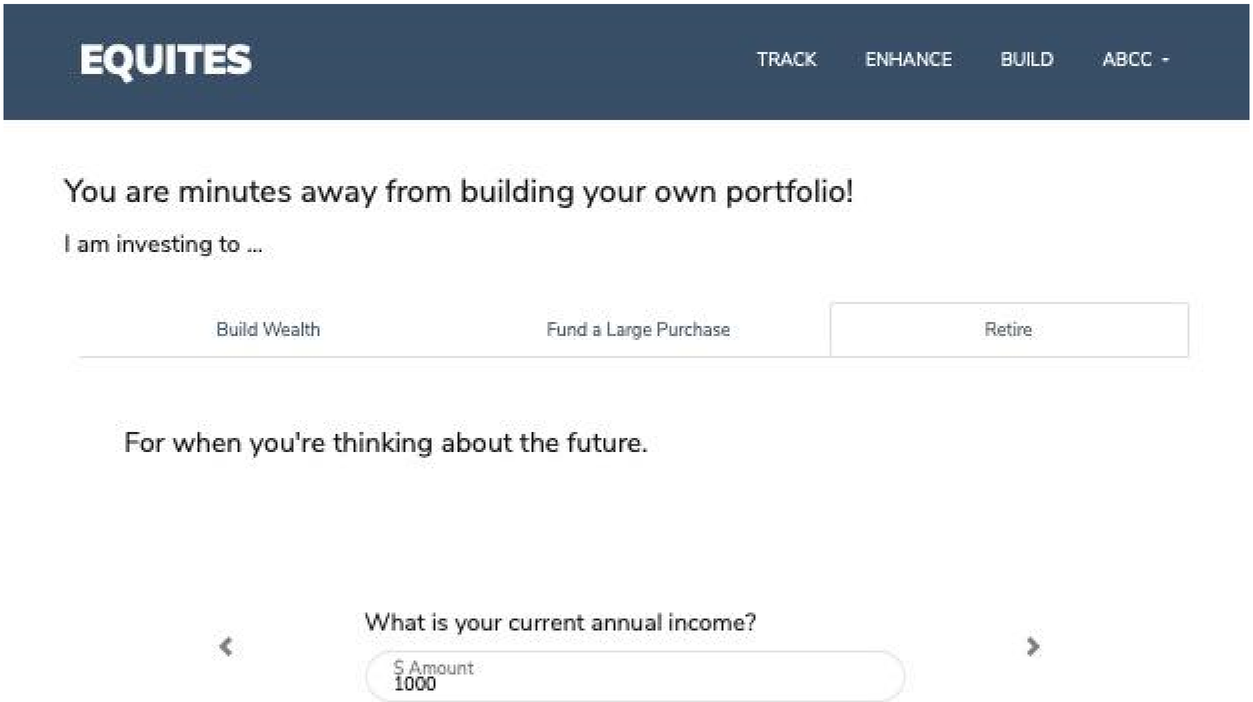


Figure 37 : Retire- Questionnaire page2

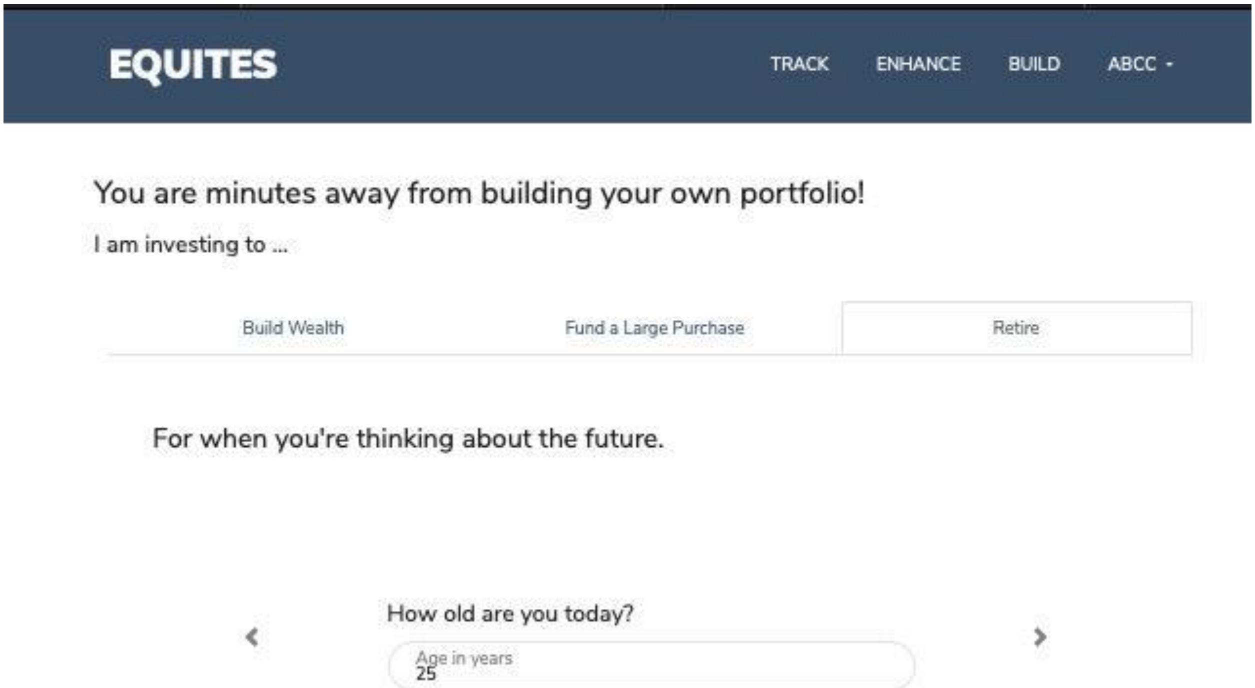

EQUITES
TRACK
ENHANCE
BUILD
ABCC
You are minutes away from building your own portfolio!
I am investing to ...
Build Wealth
Fund a Large Purchase
Retire
For when you're thinking about the future.
How old are you today?
Age in years
25

Figure 38 : Retire- Questionnaire page3

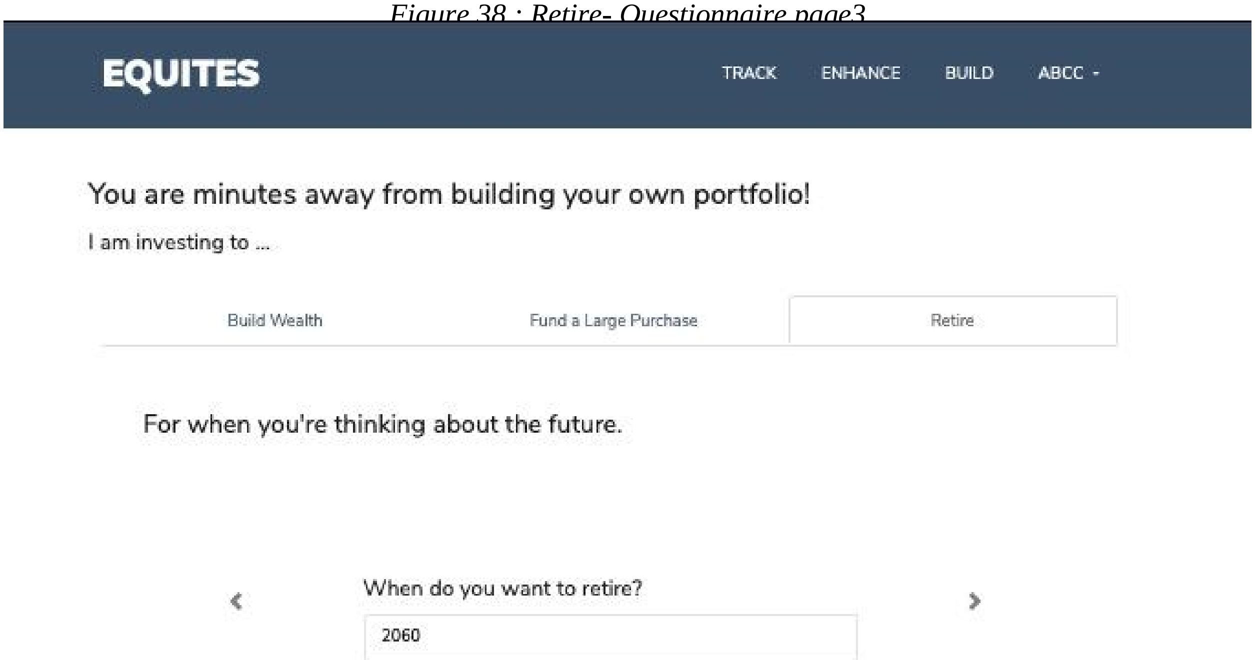


Figure 39 : Retire- Questionnaire page4

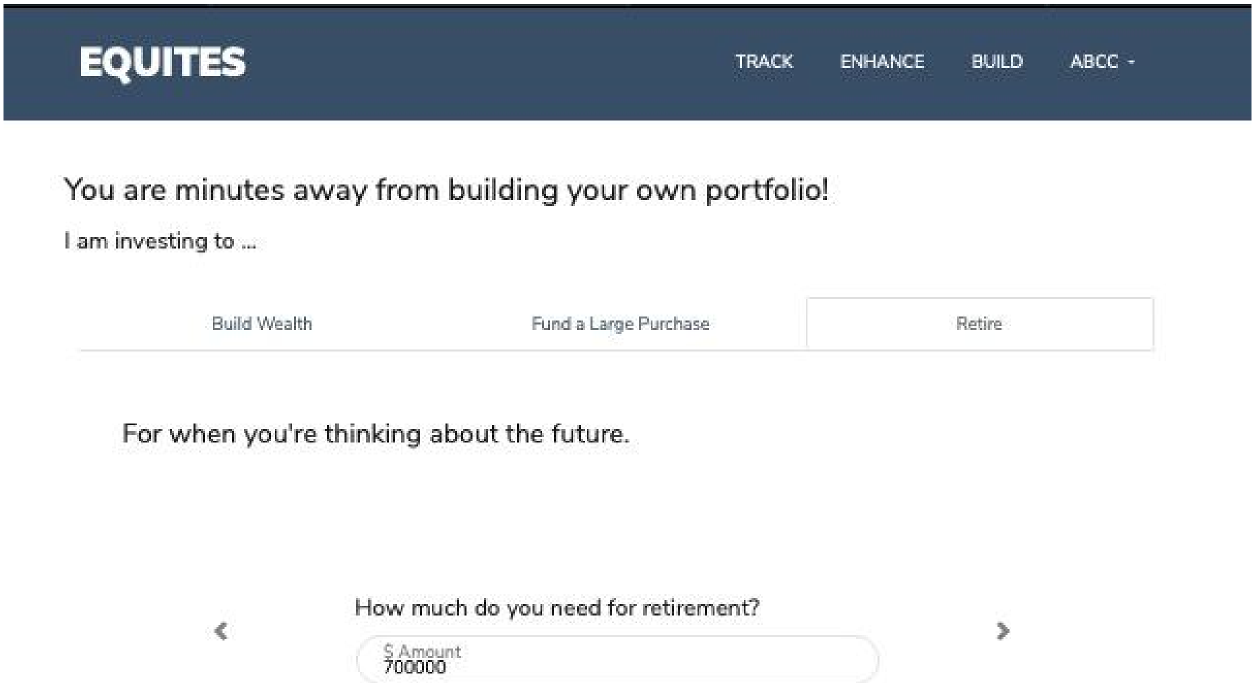

EQUITES
TRACK
ENHANCE
BUILD
ABCC
You are minutes away from building your own portfolio!
I am investing to ...
Build Wealth
Fund a Large Purchase
Retire
For when you're thinking about the future.
How much do you need for retirement?
$ Amount
700000

Figure 40 : Retire- Questionnaire page5

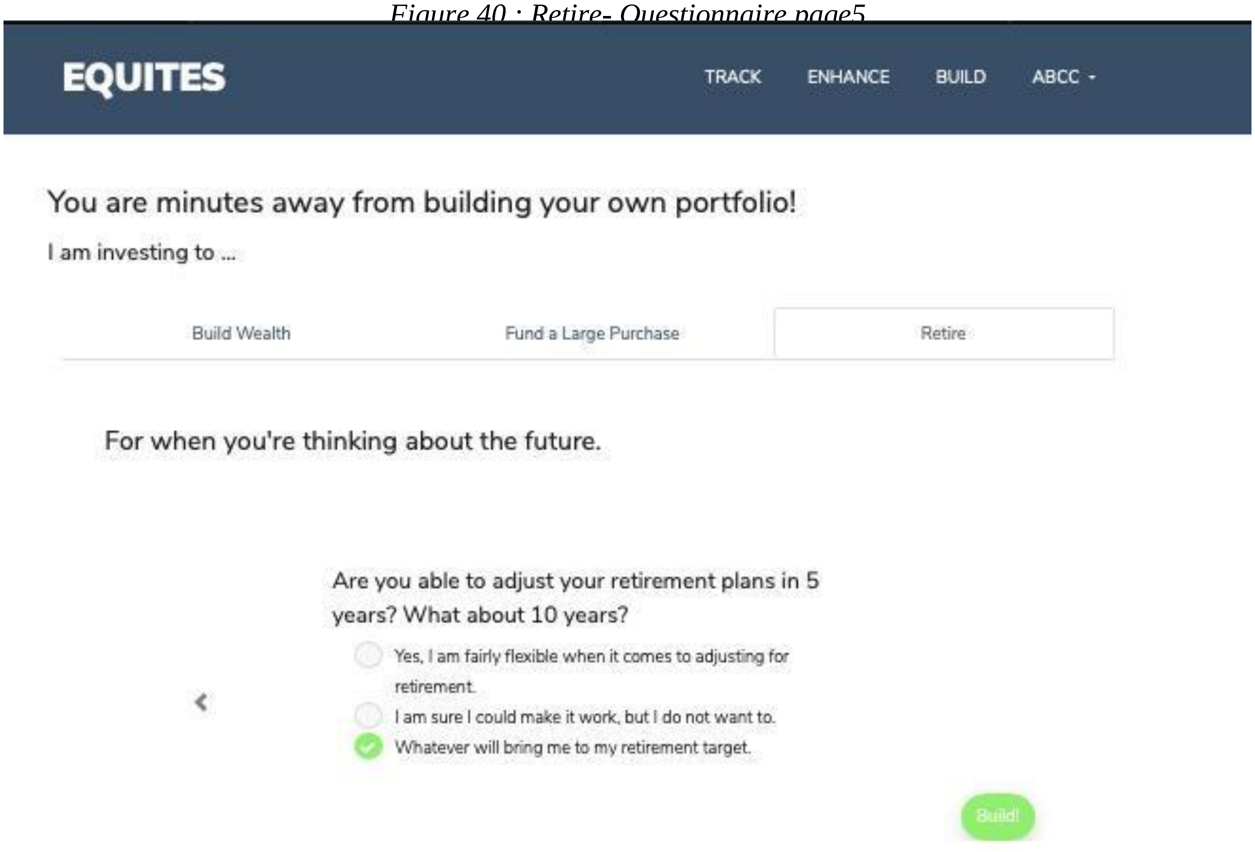


Figure 41 : Retire- Questionnaire page6

6.1.1.7 Snapshot and Dashboard

When a portfolio generated, users are able to save it and view it later. By clicking “MY PORTFOLIOS” through the dropdown menu under username on navigation bar, users can view a snapshot of all the saved portfolios. It is also available to see the details of each portfolio in dashboard by clicking the portfolio.

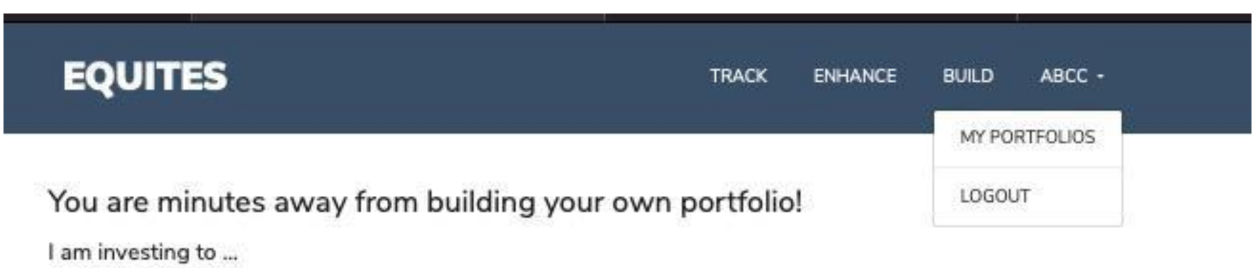


Figure 42 : Drop-down Menu

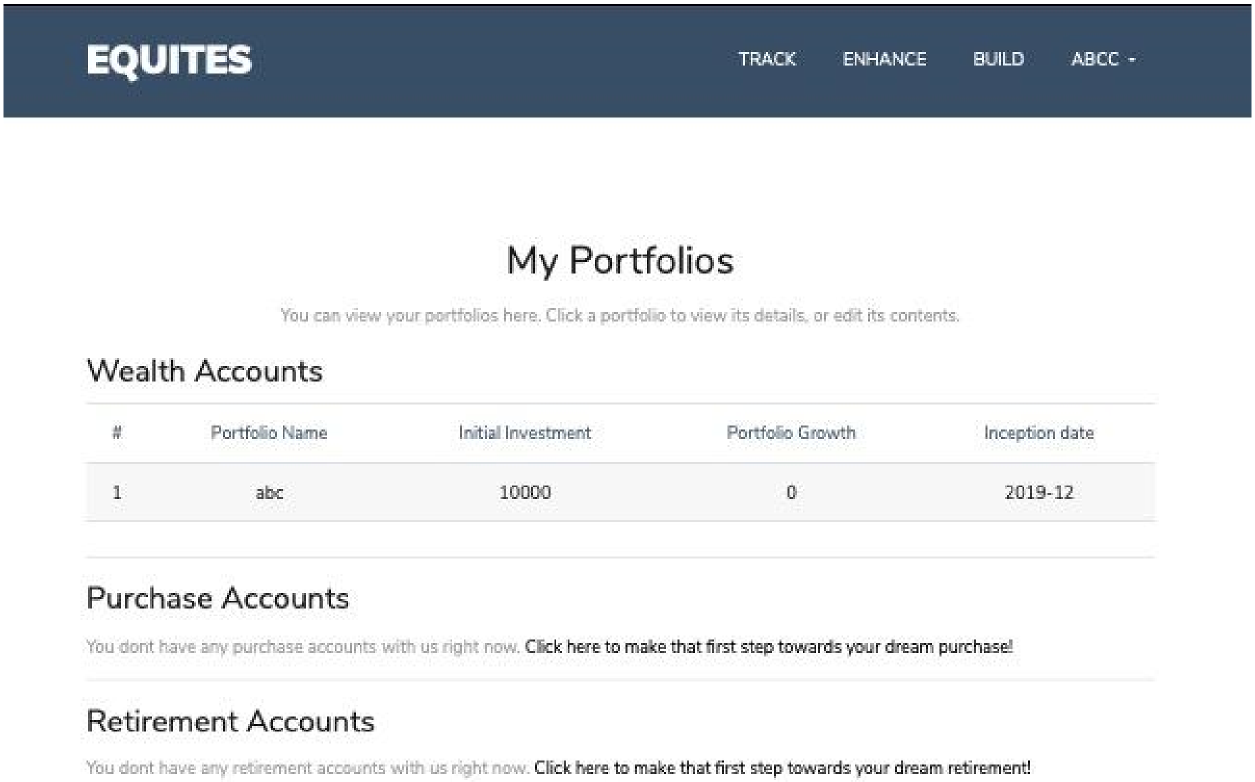


Figure 43 : Snapshot

6.2 Elements of Design

A modern approach to aesthetic design goal is taken as over time modern web design has accepted graphic design elements based on colour theory, colour psychology and empirical studies and has neglected simple designer preference. These elements include minimal designs with plenty of white space, a strong but limited colour palette and smooth and rounded edges.

A recent study by EyeQuant found that clean designs resulted in lower bounce rates, thereby capturing and retaining more users. Additionally, minimalism has brought in faster page loading times and better compatibility between screen sizes. To incorporate these minimalistic design ideas white space is taken advantage of and pages only contained necessary text and information.

To further the implementation of minimalistic and modern design, the popular 60-30-10 graphic design rule is used for the strong but limited colour palette. The 60-30-10 rule states that 60% of colour should be the primary colour, 30% should be a muted secondary colour and 10% should be a high chromatic accent colour and should use an analogous colour palette. After considering multiple analogous colour palettes the green and blue palette was chosen. This is due to colour theory stating that the palette provides the subject a calming and relaxing experience as well as portrays a sense of professionalism.

Table 8: Comparison of Colour Palettes

Colour Palette	Advantages	Disadvantages
Warm Colours (Red, Orange, Yellow)	- Energetic and passionate	- Associated with anger, fire, violence & warfare
Cool Colours (Blue, Green, Purple)	- Calm and relaxing, professionalism	- Green is associated with jealousy and blue with sadness
Neutral Colours (Black, Brown, Grey, White)	- Not distracting	- Meant to serve as backdrop and combined with brighter accent colours

The largest impact of calming and relaxation comes from the blue portion of the palette, with green adding the representation of new beginnings and abundance. New beginnings symbolizes the users new beginning of building wealth through smart equity allocation, for the future abundance they will experience from such a decision. Such a combination of feelings and portrayals is desirable as ideally the user would be providing Equites with a large portion of their savings which can be a stressful situation based on trust, so any method of providing the user some comfort is highly sought after. Blue (#3B4F66) was chosen as the 60% primary colour as it provides subjects the highest sense of relaxation from the palette, white (#FFFFFF) was chosen as the 30% muted colour to help establish the goal of maximizing white space and green (#A7E66E) was chosen as the accent. The accent is of high importance, because although relaxation was the attribute most sought after, the webpage contains an objective of being memorable and according to colour psychology high chromatic and intense colours provide the highest success of memorability.

To get the most out of the palette decision the home, login and registration pages, being the pages the user first engages with, used blue as the background to provide a sense of invitation and the questionnaire and portfolio display pages used a white background to capitalize on the less distracting white space to allow the user to focus on the monetary decisions they were making.

Another element of design incorporated, is the use of rounded edges for buttons and user input spaces. Rounded edges are heavily used in modern design and for good reason, according to Barrow Neurological Institute "perceived salience of a corner varies linearly with the angle of the corner. Sharp angles generated stronger illusory salience than shallow angles". In other words, the sharper the corner, the brighter it appears and the brighter a corner appears, the harder it is to look at. Another reasoning presented by the FMC Visualization guide rounded corners "suits better to the natural movement of the head and eyes".

6.3 User Accessibility

The main goal for accessibility is to make the application suitable and user friendly for both the experienced and novice user. This is mainly done in how the flow of the three options track, enhance and build are constructed. Track and enhance are both created such that all inputs and outputs are on the same page, by doing so gives more autonomy to the more experienced users giving them full control of the tickers and dates used for optimization and seeing the output directly on the page, allowing them to compare their input and output side by side and allowing them to make adjustments based on their own judgement. In addition, the user is initially shown basic portfolio statistics after portfolio generation but are given the option to view more complex statistics such as rolling volatility and sharpe ratio, waterfalls, and drawdowns by toggling the switch as seen below. By giving the user this option it doesn't overwhelm the novice user but gives the more experienced user the ability to analyze their portfolio in more depth.

Figure 44: Example of toggle switched on

The build option is geared more towards the novice user who may not fully understand the ins and outs of investing. Therefore it is constructed as a step by step questionnaire, guiding them through the process.

Another feature includes the applications to user input error. The front-end only accepts the allowed inputs for the models, therefore avoiding breaking of the model as well as inputs that result in non-sensical outputs.

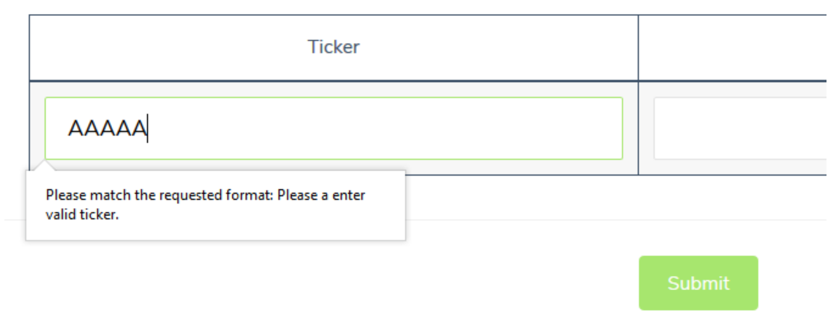


Figure 45: Example of front-end recognizing invalid ticker

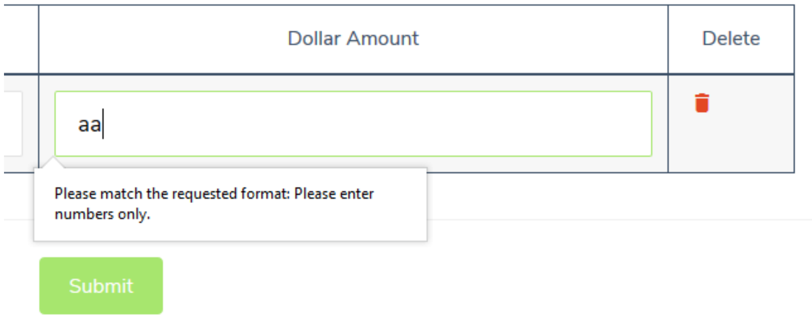


Figure 46: Example of front-end recognizing invalid dollar amount

6.4 Technologies

The front-end uses the standard HTML, CSS and Javascript technologies. The application also made use the Jinja2 templating engine. Jinja2 was compared to another popular templating engine, Mako, as seen below. Both templating functionalities are extremely similar therefore only compatibility with flask and syntax are compared.

Table 9: Comparison of Templating Engines

	Jinja2	Mako
Flask Compatibility	Default templating engine for flask	Compatible
Syntax	Subjectivity more readable within HTML	Subjectively difficult to read within HTML

To detail the subjective syntax comparison both Mako and Jinja2 examples can be seen below. Subjectively, the single “%” at the beginning of the code makes the location of the code less noticeable and clean compared to Jinja2’s “{{ }}” and “{% %}”.

```
<html>
  <head>
    <title>My Python articles</title>
  </head>
  <body>
    <p>These are some of the things I have written about Python:</p>
    <ul>
      %for topic in topics:
      <li>${topic}</li>
      %endfor
    </ul>
  </body>
</html>
```

Figure 47: Example of Mako

```
<body>
  <div class="container">
    <p>My string: {{my_string}}</p>
    <p>Value from the list: {{my_list[3]}}</p>
    <p>Loop through the list:</p>
    <ul>
      {% for n in my_list %}
      <li>{{n}}</li>
      {% endfor %}
    </ul>
  </div>
  <script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
  <script src="http://netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js">
</body>
```

Figure 48: Example of Jinja2

7. Back-End Design

The backend logic will be written in Python via Flask framework. Flask is a microframework for web applications that emphasize on quick development and easy extension. It supports API routing, and integration with the Jinja framework. Flask, however, does not include a database abstraction layer. This meant that a separate database will need to be maintained.

7.1 Summary and Overview

The backend logic is implemented in Python via the Flask framework. In order to support the rest of the application, the backend is divided into 4 main partitions: the API routing, Database solutions, User authentication, and hosting. In addition, the relationship between the parts can be summarized as the following:

7.1.1 Code flow

The code for the project can be described as the following flowchart. Note that the solution is designed to be modular so that each subsystem can be developed independently.

A few notable data flows are illustrated below:

7.1.1.1 Initializing the Application

Figure 49: Flowchart for starting the application

7.1.1.2 Getting to the Questionnaires

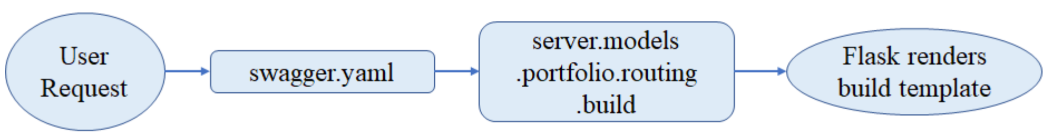


Figure 50: Flowchart for getting to the questionnaires

7.1.1.2 Generating portfolio snapshot page

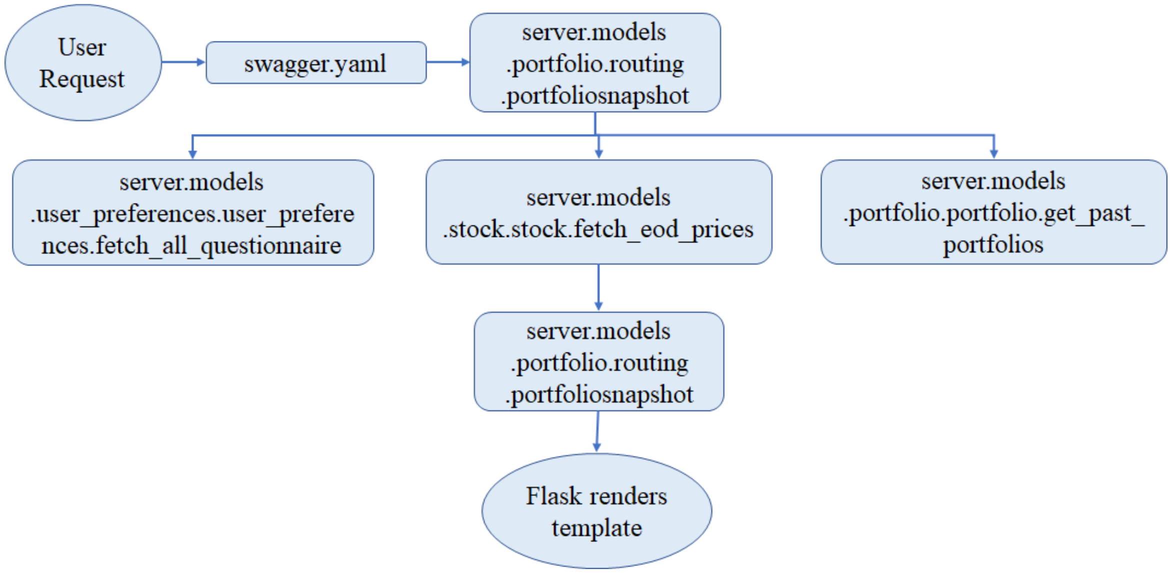


Figure 51: Flowchart for getting to the portfolio snapshot

7.1.1.2 Navigating the questionnaires and saving portfolio

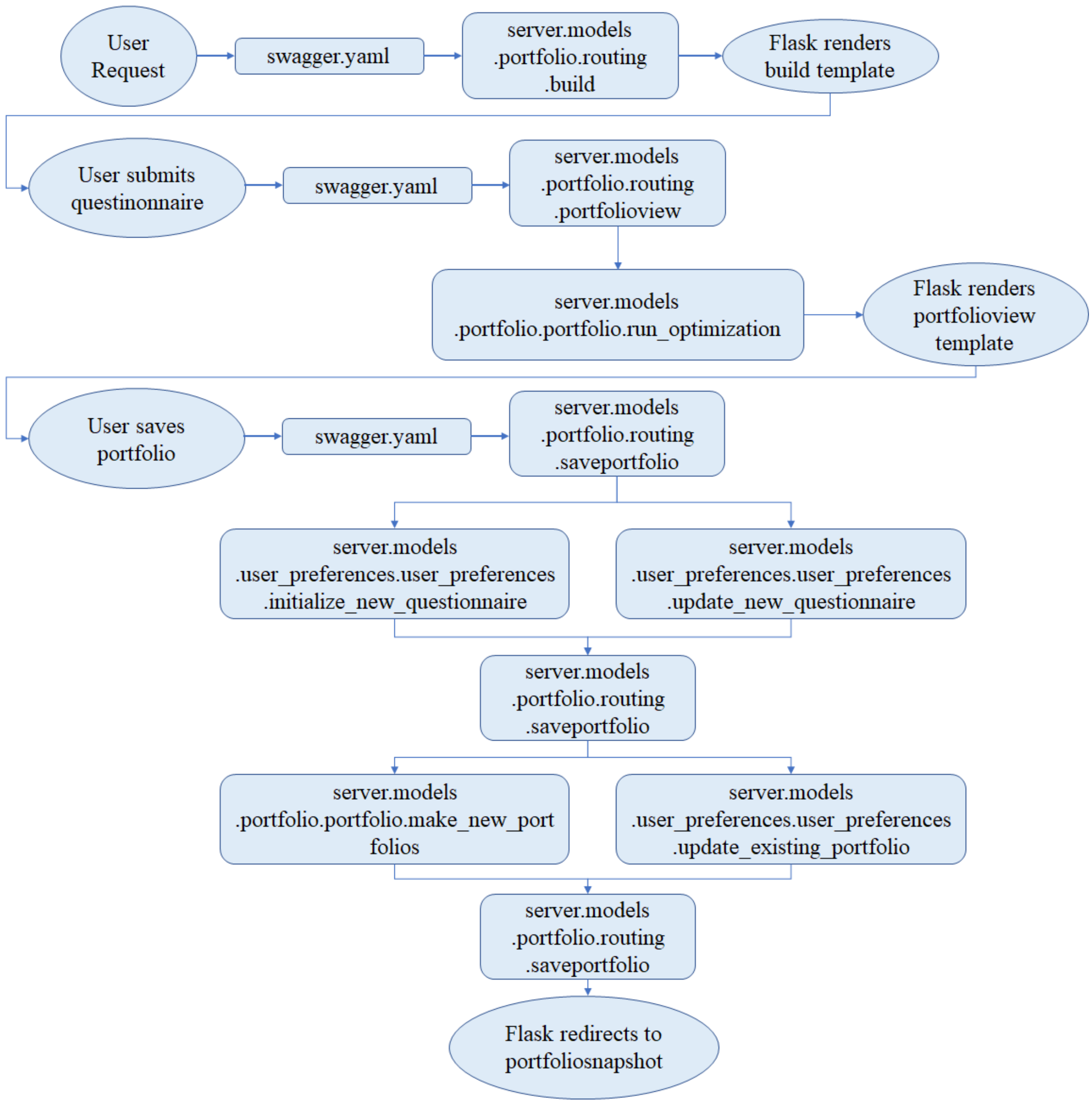


Figure 52: End-to-end flow for a user who goes through the questionnaire and saves a portfolio

7.1.1.4 Updating the database

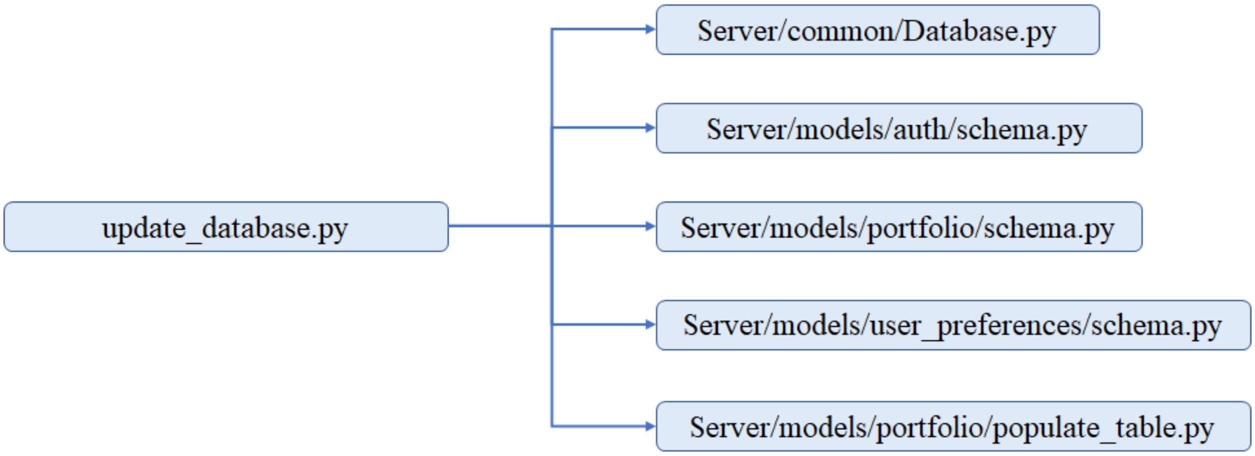


Figure 53: Flowchart for routine updating the database

7.2 API Routing

The resources in the frontend require communication from the server to provide information and direction via Application Programming Interfaces, or API. Upon reception of HTTP requests, API routes the incoming request to its corresponding unique endpoint. These API endpoints are the only means to accessing the server for security reasons.

7.2.1 Implementation

All APIs are designed to be RESTful, as recommended by RFC7230. While communication between the client and server is possible, through a tradeoff comparison shown below, RESTful APIs ultimately offered the most flexibility and lightweightness for the application.

Table 10: Comparison between API: SOAP vs REST

	SOAP (Simple Object Access Protocol)	REST (Representational State Transfer)
Function	Function-driven; transfer structured information	Data-driven: access a resource for data
Data fromat	XML only	Various types (plain text, HTML, XML, and JSON)
Security	Websocket and SSL	SSL and HTTPS
Bandwidth	Requires more resources and bandwidth	Requires fewer resources and lightweight
Payload handling	Requires knowledge of communication before any interaction	Needs no knowledge of API

Currently, 15 API endpoints are present, and as more functionalities are added on, more endpoints are needed. To require scalability and maintainability, the APIs are defined in a swagger contract, in lieu of the usual Flask Blueprint approach. Swagger contracts allow the APIs to be maintained, and adjusted without writing detailed code, and a comprehensive documentation for scaling or hooking up to other servers or clients.

7.2.2 API List

Table 11: API paths, query parameters, and general description

Path	Query Parameters	Description
/	N/A	Home page. Allows user to navigate to various solutions offered.
/track	N/A	Allows user to input their own portfolio and returns relevant information about that portfolio.
/enhance	N/A	Allow user to generate a portfolio that has a higher return than one inputted by the user.
/build	N/A	Allows user to generate a new portfolio based on behavioural and quantitative questions.
/portfolioview	- timeHorizon - investmentGoal - riskAppetite - initialInvestment -optionType	Proceeding build. Allows user to save or edit the portfolio generated from the build survey.
/portfoliosnapshot	- histValues - returnSinceInception	Allows user to see his/her portfolios in one place.
/portfoliodashboard	N/A	Allow user to see details of a portfolio.
/saveportfolio	- initialInvestment - riskAppetite - retirementDate - purchaseDate - retirementAmount - purchaseAmount - option - portfolioName	Internal endpoint used to save some portfolio defined in build.
/editportfolio	- initialInvestment - riskAppetite - retirementDate - purchaseDate - retirementAmount - purchaseAmount - option - portfolioName	Internal endpoint used to edit the portfolio. Redirects user back to build page.
/aboutus	N/A	A page that describes Equites
/contact	N/A	A page that allows user to send questions to the developer.

7.2.3 Flask Connexion

Integration between the Flask app and Swagger contract is established through Connexion, a wrapper around the flask app. The swagger contract is stored in: “server/models/portfolio/swagger.yaml”

With the swagger contract, API parameters become strictly regulated on incoming requests. This will not only prevent malicious attacks from hackers, but also enforce strict confirmation to OpenAPI standards and REST protocol.

7.3 Database Solutions

7.3.1 Database Selection

A good, fast, and reliable database is required to store various types of data used in the business logic. Databases breakdown into two main models: non-relational (noSQL) and relational (SQL) databases. Each model of database can then be divided into various implementations.

A relational database management system is one that employs the relational data model. In this model, data are organized as relations (aka tables). Contained within each relation is a set of tuples (aka rows), with each tuple sharing a set of attributes (aka columns). Navigation and querying among the database is performed via structured query language (SQL). One of the prominent features of a SQL database is the predefined schema. This meant that the database is restricted in its contents upon the declaration of its existence. The trade-off to this constraint is that SQL databases are generally faster and more powerful in its querying prowess for complex operations.

A non-relational database management system is one that provides storage and retrieval of data through means other than the relational data model. The NoSQL model employs a collection of key-value stores to represent data. As a result, NoSQL is easy to simple to understand and easy to use.

Given the transaction load, as well as the potential need for relations in the data, we intend to use a relational database. A more detailed pros and cons analysis for the type can be found in the table below:

Table 12: Overview of Database considerations.

	Relational	Non-Relational
Language	SQL	NoSQL
Stored data format	Tables in forms of rows and columns	Key-value pairs, graphs or columns
Data flexibility	No flexibility. Requires predefined schema	Dynamic schema provides absolute flexibility
Scalability Cost	Costly	Cheaper in comparison
Scalability	Vertically scalable (more RAM)	Queries not powerful
ACID properties (Atomicity, Consistency, Isolation, Durability	Yes	Not always
Heavy transaction load	Excels	Not suitable

Various implementations of the relational database exists, including the popular MySQL, lightweight SQLite, and less popular but powerful PostgreSQL. In this project, we intend to use a PostgreSQL database. We feel that its ability to perform complex operations on the querying is something to be desired, considering the various data sources we intend to use.

A more detailed comparison between popular implementations of SQL is as follows:

Table 13: Comparison of SQL versions.

	Pros	Cons
SQLite	- Lightweight library that takes up less than 60 KB of space - Does not require starting or stopping. Runs as if it is a readable object. - Database stored in a single file	- Only one process of database can be run at any time. No concurrency. - Databases that uses servers can be more secure than those without. SQLite does not run on a server.
MySQL	- Secure installation as it is triggered via a script - Very fast in data retrieval - Supports database replication for failure backups	- Does not support full SQL protocol - Some features not available in the open-community version of MySQL
PostgreSQL	- SQL fully compliant - Easily Extensible - Capability to incorporate user code into database system.	- Each client connection will fork up new process - Not as popular as MySQL

7.3.2 Database architecture

The database's tables were schematized as follows. A common unique identifier connects the questionnaire with the portfolio weighting. The username allows only authenticated user to have registered portfolio. These relationships allow fetching and updating data simple and straightforward.

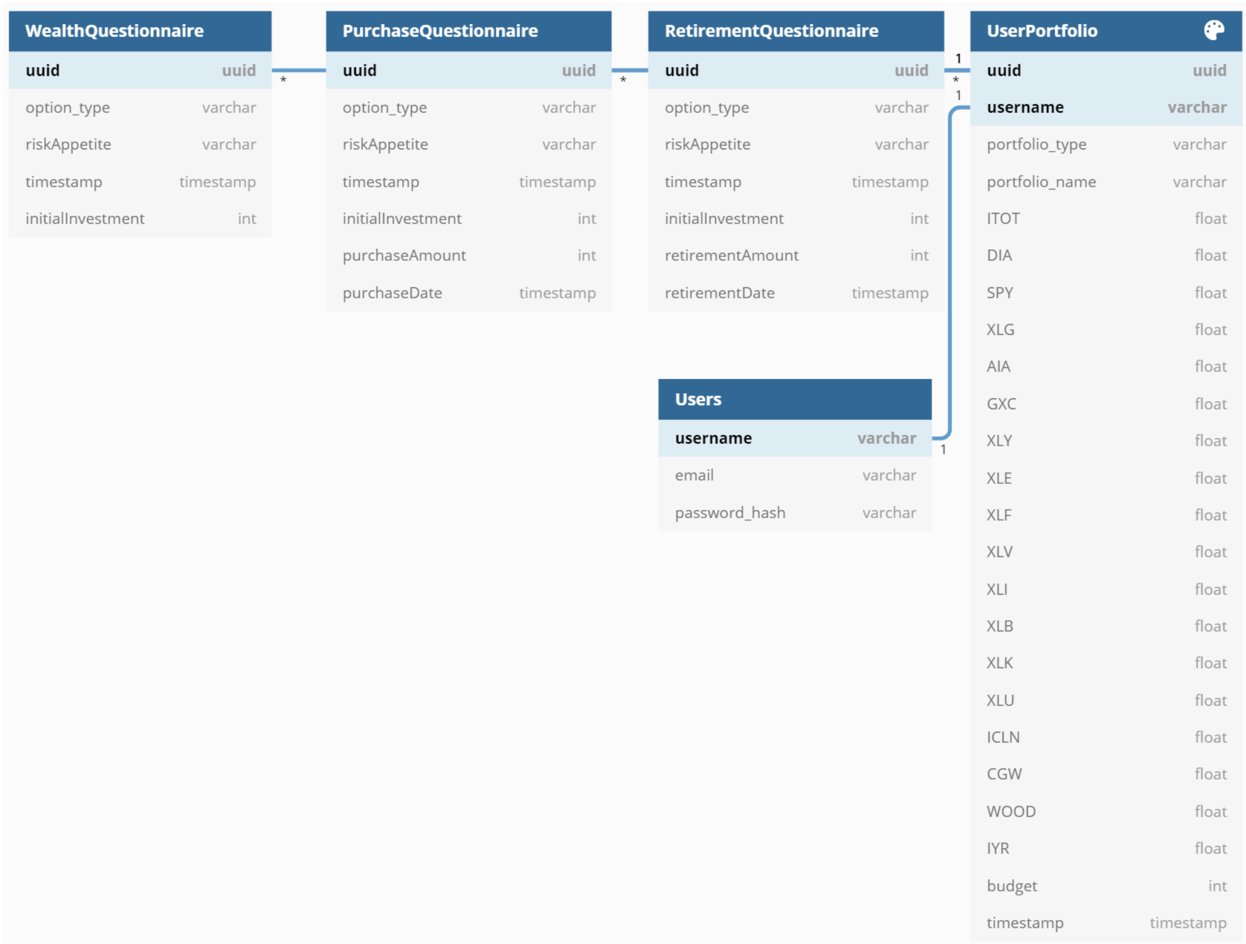


Figure 54: Relationship of tables

7.3.3 Database handling

The database are accessed for the following purposes: fetch and upsert questionnaire, fetch and upsert portfolio and fetch and upsert users. The object relational mapping is done via SQLAlchemy. The specific data post-processing is performed by Pandas.

7.3.4 Updating Price Data

The price data and other immediate important variables essential to the optimization framework are also stored in the database. The rationale is that by storing them in the database, any redundant computation is avoided. Updating price data can be performed by running the file "update_databases.py", which reinitializes the SQL database by respecting corresponding schemas. This file can then be called by scheduled routines to update the databases daily.

7.4 User Authentication

7.4.1 User flow

A good authorization framework allows the application to be secure against malicious hackers.

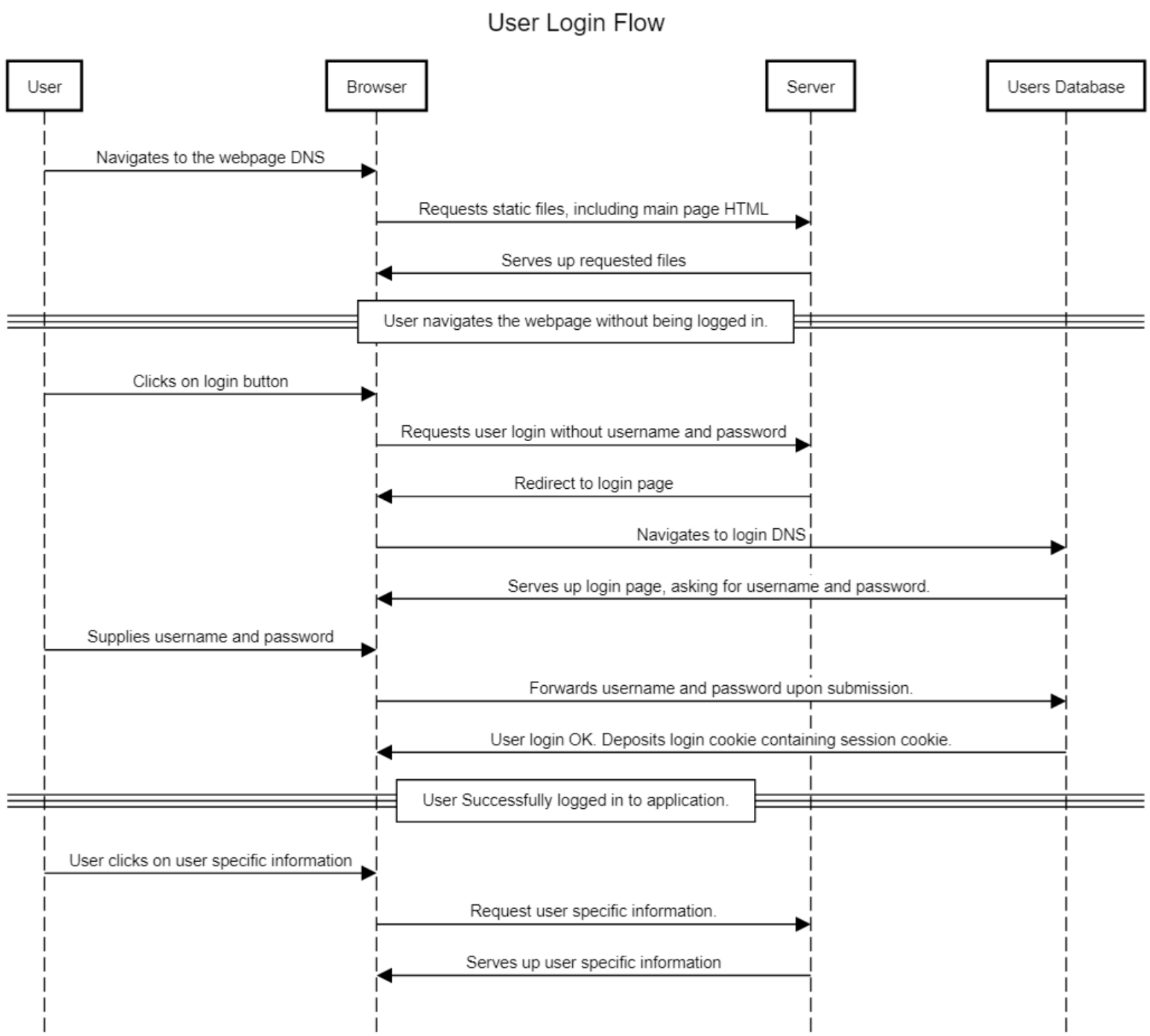


Figure 55: User Login flow.

7.4.2 Session management

Login session is maintained via the login manager via flask-login package. A login session should know when a user is logged in, and permit or restrict permission and access. The login manager accomplishes these by depositing a HTTP only session cookie, which is sent along with all HTTP requests. An advantage of a HTTP-only cookie is that the browser/client cannot edit its content, which makes the authentication secure, and prevents cross-site forgery. This session cookie is removed when the user logs out.

7.5 Hosting

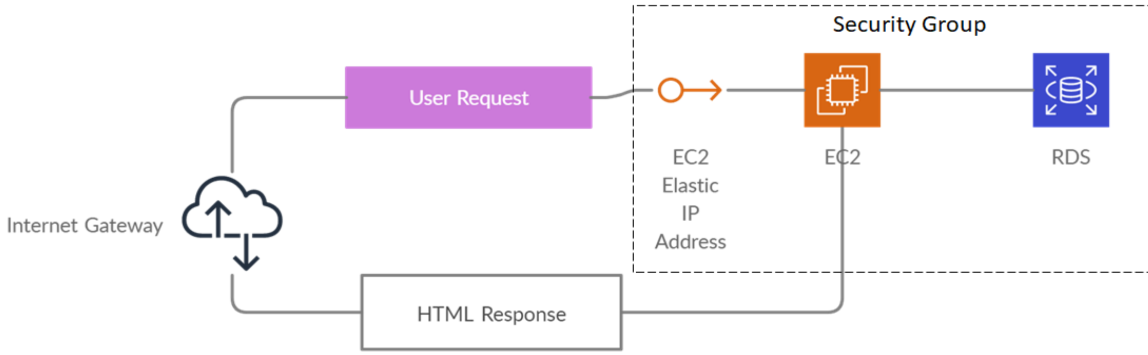


Figure 56: Cloud Architecture

7.5.1 Service Provider Selection

Hosting is one of the most important aspects of any applications. A good hosting service provides reliability, scalability, and security to the end to end flow for a functional user. In industry, a few established players have dominated the service provider market.

While each service providers have their own perks, we have decided to continue the venture with Azure for its lowered cost and the free credits available with GitHub Student Pack. In the table below we will compare the pros and cons in more detail:

Table 14: A Comparison and Contrast of Hosting Services and promotions offered.

	GitHub Student Pack	Pros	Cons
AWS	$110 in bonus AWS credits for a total of $75-$150.	- Flexibility: supports multiple OS's, programming language, database types, and application platform. - Reliability: Multiple availability zones worldwide. - Scalability: Inherent capability to expand. - Security: All data stored in security and data at rest and in transit encryption available.	- Hard to set up for first time users
Digital Ocean	$50 in platform credit for new users.	- Quick to set up - Affordable price	- Linux-based servers only. - Infrastructure as a service, so user requires management of services.
Heroku	N/A	- Quick to set up - Similar to AWS Elastic Beanstalk	- Not for performance heavy applications - Slow deployment
Google Cloud Platform	N/A	- Low cost: 25% cheaper than AWS - Unlimited free trial. Limited in space - Better UX	- Less service than AWS - Less market popularity than AWS.
Azure	$100 credit with your free Azure for Students account—no credit card required.	- Lower cost than AWS - 12 month of free usage - 25+ services forever free - Allow complete configuration of the infrastructure - Much better documentation than AWS - All perks from AWS	- Less market popularity than AWS.

Given team's experience in AWS, the application is decided to the cloud instance on AWS.

7.5.2 AWS EC2

AWS offers a variety of cloud computing services, one of the most prominent is its virtual machine instances, EC2 (Elastic Compute Cloud), a platform-as-a-service that provides secure, resizable compute capacity in the AWS cloud.

To most the main logic for the applications, a linux virtual machine is selected with a T2.micro instance. The T2 instance is perfect for general purpose web applications, which does not require a high I/O rate or UDP streaming. A micro instance is selected despite having only 1 GB of RAM for its free tier eligibility. Within development, this instance type has empirically been sufficient for all intended purposes. However, at production, a larger instance type, such as T2.medium or T2.large can be selected to handle user requests.

The EC2 instance can be accessed at the following URL:

ec2-35-182-226-124.ca-central-1.compute.amazonaws.com

7.5.3 RDS

Relational Database Service, or RDS, provides a cost-efficient and resizable capacity to host databases on the cloud, automating time-consuming administration and provisioning tasks such as setting up and backing up the database. RDS offers a variety of database engines, including inhouse Aurora, MySQL, and PostgreSQL. Remaining faithful to the initial proposal, a PostgreSQL engine is linked to the EC2 instance via security groups described in the following section.

In addition to the data hosting and querying, RDS also offers backup services such as Multi-AZ and Read Replicas. The former allows backup across multiple AWS regions in the rare case the RDS instance in Canada region goes down. The latter refers to replicas of the database that are automatically generated when I/O rate becomes very high.

The RDS instance can be accessed at the following URL:

database-equites.cfdzgfxqopom.ca-central-1.rds.amazonaws.com

7.5.4 Security Groups

Security groups acts as a virtual firewall for the EC2 and RDS instances to control inbound and outbound traffic. Currently, the attached security group allows all HTTP and TCP inbound traffic, and all types of outbound traffic.

7.5.5 IAM roles

IAM (Identity Access Management) roles are used to delegate access to users, applications and services that don’t have access to the resources. IAM roles are an important aspect of the security and vulnerability protection, by allowing users, applications and services alike only the minimum necessary set of permissions, or in AWS terms, rules, in their access privileges. This is in accordance with AWS’s best practice guideline: Grant Least Privilege.

7.5.6 Instances

Grant Least Privilege imposes that a EC2 instance which only requires access to an RDS instance nothing else, do not have permission to access other AWS services. This prevents potential security threats and minimizes any potential damage in the event of a vulnerability exposure.

Specifically, the following rules are added to the security groups associated with the EC2 and RDS instance:

- RDS Full Access (including CloudWatch, and various auto-scaling features)

7.5.7. Developer accounts

To facilitate development process, developer accounts under a single root account are created and distributed to all team members. Developer accounts have IAM roles such that they can access the EC2 instances and the RDS instances.

7.5.8 Monitoring

AWS offers CloudWatch, a logging service that provides detailed logs of all AWS instances, and a monitoring platform for all functional users’ activities. The monitoring aspect can also be leveraged once in production to generate key insights on user preferences and application performance.

8. Conclusion

The primary goal of this project was to deliver a portfolio optimization engine with an easily accessible user experience. The product must describe and produce high performing portfolios, while remaining flexible enough to accommodate a wide variety of users with different levels of financial literacy.

These requirements were met by seamlessly integrating the different aspects of the project. The front end of the application guides users through the process of setting up their experience while storing all the information required for the optimization procedures in the backend. The business logic framework applies state-of-the-art financial optimization tools to produce the highest performing portfolios possible. Machine learning was be leveraged in a limited sense within the business logic framework. It enhances aspects of classic financial optimization and will mitigate against future risk by incorporating continuously updating representations of public sentiment. The integration of these components will provide a robust framework for portfolio optimization. It will also allow for users to select their desired level of involvement based on financial literacy

9. References

Xing, F., Cambria, E. and Welsch, R. (2018). Intelligent Asset Allocation via Market Sentiment Views. *IEEE Computational Intelligence Magazine*, 13(4), pp.25-34.

Heaton, J., Polson, N. and Witte, J. (2016). Deep Portfolio Theory. *Arxiv*.

Haugen, R. and Baker, N. (1991). The efficient market inefficiency of capitalization–weighted stock portfolios. *The Journal of Portfolio Management*, 17(3), pp.35-40.

Pyo, S. and Lee, J. (2018). Exploiting the low-risk anomaly using machine learning to enhance the Black–Litterman framework: Evidence from South Korea. *Pacific-Basin Finance Journal*, 51, pp.1-12.

Silva, T., Pinheiro, P. and Poggi, M. (2017). A more human-like portfolio optimization approach. *European Journal of Operational Research*, 256(1), pp.252-260.

Markowitz, H. (1952). Portfolio Selection. The Journal of Finance, 7(1), 77-91.doi:10.1111/j.1540-6261.1952.tb01525.x

Stephen Boyd, Enzo Busseti, Steve Diamond, Ronald N. Kahn, Kwangmoo Koh, Peter Nystrup and Jan Speth (2017), "Multi-Period Trading via Convex Optimization", Foundations and Trends® in Optimization: Vol. 3: No. 1, pp 1-76. http://dx.doi.org/10.1561/2400000023

Gulpmar, N., Rustem, B., & Settergren, R. (2004, May 17). Multistage Stochastic Mean-Variance Portfolio Analysis with Transaction Costs [Scholarly project]. Retrieved March 30, 2018, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.3969&rep=rep1&type=pdf

Mahmutoğulları, A. I., Çavuş, Ö, & Aktürk, M. S. (2017). Bounds on risk-averse mixed-integer multi-stage stochastic programming problems with mean-CVaR. European Journal of Operational Research, 266(2), 595-608. doi:10.1016/j.ejor.2017.10.038

Sharpe, William F. “Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk.” The Journal of Finance, vol. 19, no. 3, 1964, pp. 425–442. JSTOR, www.jstor.org/stable/2977928.

Fama, Eugene F., and Kenneth R. French. “The Cross-Section of Expected Stock Returns.” The Journal of Finance, vol. 47, no. 2, 1992, pp. 427–465. JSTOR, www.jstor.org/stable/2329112.

Fama, Eugene F. and French, Kenneth R., A Five-Factor Asset Pricing Model (September 2014). Fama-Miller Working Paper. Available at SSRN: https://ssrn.com/abstract=2287202 or http://dx.doi.org/10.2139/ssrn.2287202

Carhart, Mark M. “On Persistence in Mutual Fund Performance.” The Journal of Finance, vol. 52, no. 1, 1997, pp. 57–82. JSTOR, www.jstor.org/stable/2329556.

Ho-Jung Yoon & Kyuseok Lee (2019) A weighted Fama-MacBeth two-step panel regression procedure, Applied Economics Letters, 26:8, 677-683, DOI: 10.1080/13504851.2018.1489492

Fouque, Jean-Pierre and Sircar, Ronnie and Zariphopoulou, Thaleia, Portfolio Optimization and Stochastic Volatility Asymptotics (July 2017). Mathematical Finance, Vol. 27, Issue 3, pp. 704-745, 2017. Available at SSRN: https://ssrn.com/abstract=2985814 or http://dx.doi.org/10.1111/mafi.12109

Kadan, O., Liu, F., & Tang, X. (2017). A forward-looking factor model for volatility: Estimation and implications for predicting disasters [Electronic version]. http://scholarship.sha.cornell.edu/workingpapers/39

Liu R.H., Ren D. (2019) Portfolio Optimization Using Regime-Switching Stochastic Interest Rate and Stochastic Volatility Models. In: Yin G., Zhang Q. (eds) Modeling, Stochastic Control, Optimization, and Applications. The IMA Volumes in Mathematics and its Applications, vol 164. Springer, Cham
Hamilton, James D. "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." Econometrica, vol. 57, no. 2, 1989, pp. 357–384. JSTOR, www.jstor.org/stable/1912559.

Ossiam. "Markov Switching Models and the Volatility Factor: A MCMC Approach". https://www.ossiam.com/files/research_papers/1479203133_Markov_Switching_Models_and_the_Volatility_Factor_A_MCMC_Approach..pdf

Ammann, Manuel and Verhofen, Michael, The Effect of Market Regimes on Style Allocation (January 1, 2009). Available at SSRN: https://ssrn.com/abstract=1322278 or http://dx.doi.org/10.2139/ssrn.1322278

Guidolin, Massimo & Timmermann, Allan, 2007. "Asset allocation under multivariate regime switching," Journal of Economic Dynamics and Control, Elsevier, vol. 31(11), pages 3503-3544, November.

van Ravenzwaaij, D., Cassey, P. & Brown, S.D. Psychon Bull Rev (2018) 25: 143. https://doi.org/10.3758/s13423-016-1015-8

Didier Chauveau, Pierre Vandekerkhove. How to compare MCMC simulation strategies?. 2007. ffhal00019174v3f

Satchell, S. & Scowcroft, A. "A demystification of the Black–Litterman model: Managing quantitative and traditional portfolio construction".J Asset Manag (2000) 1: 138. https://doi.org/10.1057/palgrave.jam.2240011

Yugu Xiao & Emiliano A. Valdez (2015) A Black–Litterman asset allocation model under Elliptical distributions, Quantitative Finance, 15:3, 509-519, DOI: 10.1080/14697688.2013.836283

Meucci, Attilio, Beyond Black-Litterman in Practice: A Five-Step Recipe to Input Views on Non-Normal Markets (May 2006). Available at SSRN: https://ssrn.com/abstract=872577 or http://dx.doi.org/10.2139/ssrn.872577

Willis, J., & Todorov, A. (2006). First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face. Psychological Science, 17(7), 592–598. https://doi.org/10.1111/j.1467-9280.2006.01750.x

Schwab, Katharine (2017). “Study: Clean Design Really is Better” Fast Company. Available at: https://www.fastcompany.com/90144106/stop-cluttering-up-your-website-study-suggests-its-bad-for-business

FMC Research Staff (2003) “Visualiztion Guidelines” Fundamental Modeling Concepts. Available at: http://www.fmc-modeling.org/visualization_guidelines

Macknik, S., Martinez-Conde, S. & Troncoso X. “Corner salience varies linearly with corner angle during flicker-augmented contrast: a general principle of corner perception based on Vasarely’s artworks” Spatial Vision 22 (3).
https://pdfs.semanticscholar.org/5353/31efbca891d601304a472c81850c85cab232.pdf

“Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing.” IETF Tools, tools.ietf.org/html/rfc7230.https://www.google.com/url?q=https://tools.ietf.org/html/rfc7230&sa=D&ust=1575490020308000&usg=AFQjCNGV5IAXFpw7SdVVxhzE-3eZkZSw4A