**Profit Maximization**

Submitted in partial fulfillment of the requirements

of the degree of

**T. E. Computer Engineering**

By

**John Aishwarya Mathai 37**

**182044**

**Kurien Anashwara Shaji 44**

**182051**

**Lobo Sherwin Robinson 48**

**182055**

Guide (s):

**Mrs. Vincy Joseph**

Assistant Professor



Department of Computer Engineering

St. Francis Institute of Technology

(Engineering College)

University of Mumbai

2020 - 2021

**CERTIFICATE**

This is to certify that the project entitled **“Profit Maximization”** is a bonafide work of **“John Aishwarya Mathai” (37),“Kurien Anashwara Shaji” (44) and “Lobo Sherwin Robinson” (48)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of T.E. in Computer Engineering

****

**Mrs. Vincy Joseph**

**Guide**

**Dr. Kavita Sonawane**

**Head of Department**

**Project Report Approval for T.E.**

This project report entitled **Profit Maximization** by **John Aishwarya Mathai, Kurien Anashwara Shaji and Lobo Sherwin Robinson**  is approved for the degree of ***T.E. in Computer Engineering.***

Examiners

1.Mr. Mahendra Mehra



2.Mrs. Vincy Joseph

Date: 28-05-2021

Place: Mumbai

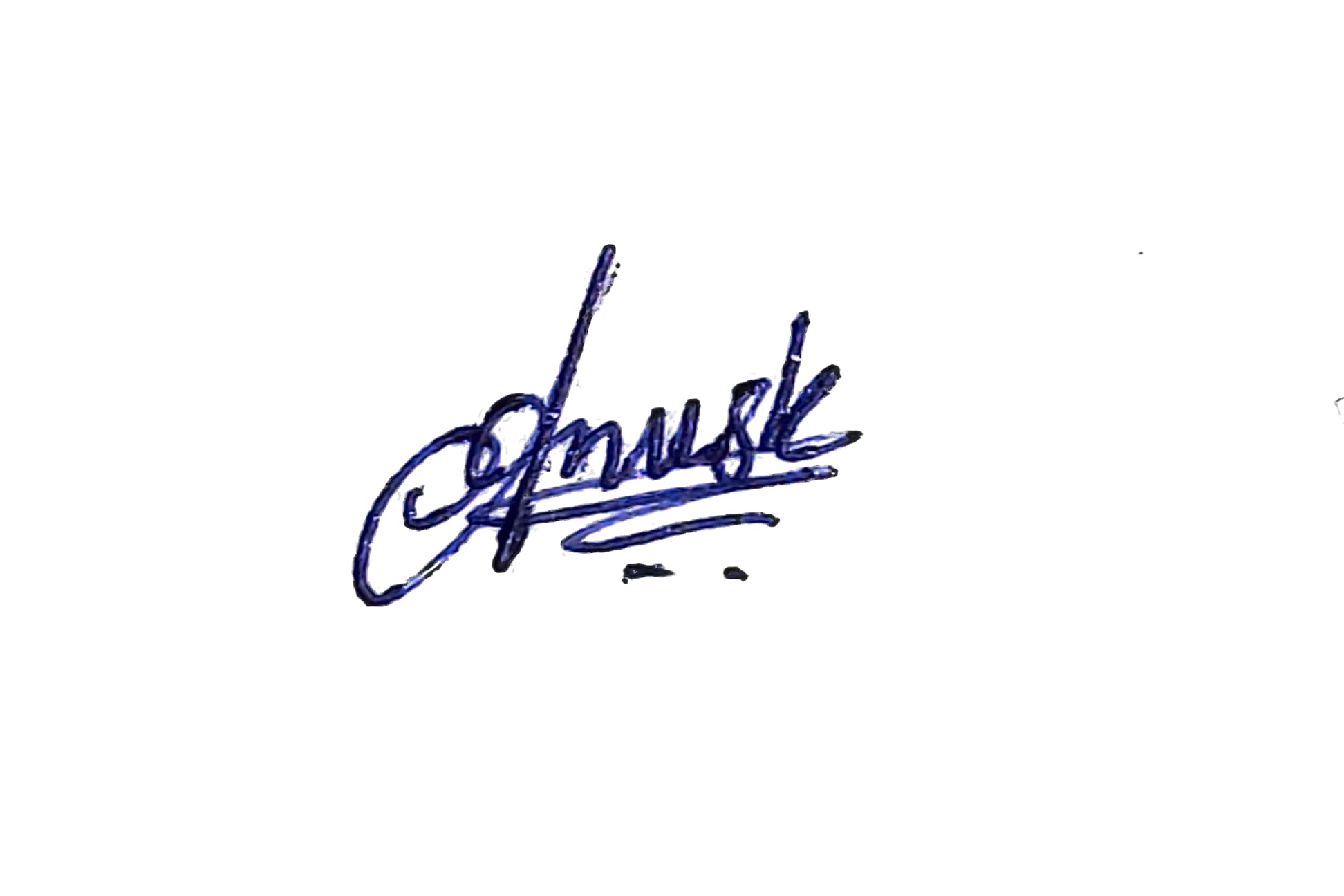
Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



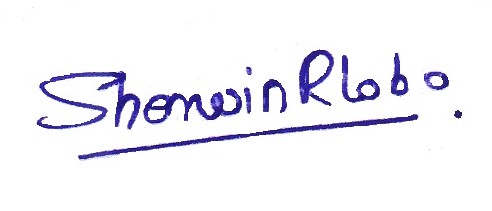
-----------------------------------------

**John Aishwarya Mathai 37**

****

-----------------------------------------

**Kurien Anashwara Shaji 44**



-----------------------------------------

**Lobo Sherwin Robinson 48**

Date:28-05-2021

**Abstract**

Modern Portfolio Theory (MPT) or mean-variance analysis is a mathematical model/study for developing and creating a portfolio which aims to maximize the return for a given amount of risk. The math is largely based on the assumption and experience that an average human prefers a less risky portfolio. The risk mitigation can be done by either investing in traditional safe havens or by diversification — a cause championed by the MPT. The theory was introduced by Henry Markowitz in the 1950s, for which he was awarded the Nobel prize. While the MPT has had its fair share of criticisms, partly due to its backward looking tendencies and inability to factor in force majeures/trends in business and economy, it is found that the tool is valuable to gauge the risk of one’s portfolio holdings by measuring the volatility as a proxy. The concepts of expected return, mean-variance and Sharpe ratio are explored to automate optimization and save valuable time in the process of doing so. Thus the profits can be maximised by making an optimized portfolio with one’s valuable investments.

**Contents**

| **Chapter** | | **Contents** | **Page No.** |
| --- | --- | --- | --- |
| **1** |  | [**INTRODUCTION**](#dkuzl6kq14qh) | **1-4** |
| **1.1** | [**Description**](#sms88tle9eb7) | **1** |
| **1.2** | [**Problem Formulation**](#32peccjfy89x) | **1** |
| **1.3** | [**Motivation**](#qsk3ixhfv567) | **1** |
| **1.4** | [**Proposed Solution**](#reyo6y13s69b) | **1** |
| **1.5** | [**Scope of the project**](#cia18zkyzx9n) | **4** |
| **2** |  | [**REVIEW OF LITERATURE**](#bseya2ux5shi) | **5-7** |
| **3** |  | [**SYSTEM ANALYSIS**](#8lykmt50eu9) | **8-11** |
| **3.1** | [**Functional Requirements**](#5a4ygp66alt3) | **8** |
| **3.2** | [**Non Functional Requirements**](#qt4iw6lwnqy8) | **9** |
| **3.3** | [**Specific Requirements**](#ubai7ww81r2c) | **9** |
| **3.4** | [**Use-Case Diagrams and description**](#mozw531rz04o) | **11** |
| **4** |  | [**ANALYSIS MODELING**](#9lhjfgsg61sv) | **12-13** |
| **4.1** | [**Activity Diagrams**](#tz8pzjclklbn) | **12** |
| **4.2** | [**Functional Modeling**](#j914cp6u9vcv) | **13** |
| **5** |  | [**DESIGN**](#ohg36w31ky5q) | **14-17** |
| **5.1** | [**Architectural Design**](#sk0swjcq73o) | **14** |
| **5.2** | [**User Interface Design**](#ae53t0jk556) | **15** |
| **6** |  | [**IMPLEMENTATION**](#wxlbywrmer7) | **18-21** |
| **6.1** | [**Algorithms / Methods Used**](#xdvfeghaavz3) | **18** |
| **6.2** | [**Working of the project**](#o3z8brsy22jp) | **20** |
| **7** |  | [**CONCLUSIONS**](#jf5nah8b75z) | **22** |

References

Acknowledgements

**List of Figures**

| **Fig. No.** | **Figure Caption** | **Page No.** |
| --- | --- | --- |
| 3.1 | [Use-Case Diagram for Portfolio Optimization](#nxl0thsurav4) | 10 |
| 4.1 | [Activity Diagram for Portfolio Optimization](#wr8s9xsmbvtn) | 11 |
| 4.2 | [Data Flow Diagram for Portfolio Optimization](#4tuoquiu0ql1) | 12 |
| 5.1 | [Block Diagram for Portfolio Optimization](#md1johr1juhi) | 13 |
| 5.2.1 | [View all stock](#npdbotllkp3e) | 14 |
| 5.2.2 | [Portfolio optimization for Nasdaq](#z66ggwbd5ciu) | 14 |
| 5.2.3 | [Portfolio optimization for Large Cap](#n98uuahv9umj) | 15 |
| 5.2.4 | [Portfolio optimization for Mid Cap](#big5x26dj5kq) | 15 |
| 5.2.5 | [Stock Overview](#hyiu6ocq8ws0) | 16 |
| 5.2.6 | [Stock Latest News](#7xfvb47vo5y4) | 16 |
| 6.1.1 | [Weight Calculation](#o0f8t5z2tc66) | 17 |
| 6.1.2 | [Sharpe Ratio](#ipujcs7ksfrp) | 18 |

**List of Abbreviations**

| **Sr. No.** | **Abbreviation** | **Expanded form** |
| --- | --- | --- |
| i | API | Application Programming Interface |
| ii | GUI | Graphical User Interface |
| iii | MVO | Mean-Variance Optimization |
| iv | MPT | Modern Portfolio Theory |
| v | DFD | Data Flow Diagram |
| vi | BL | Black-Litterman |
| vii | CNN | Convolutional Neural Network |
| viii | DNN | Deep Neural network |
| ix | DMLP | Deep Multilayer Perceptron |
| x | LSTM | Long Short Memory |

**Chapter 1**

**Introduction**

* 1. **Description**

Profit maximization is the process of selecting the best portfolio (asset distribution), out of the set of all portfolios being considered, according to some objective in order to maximize profit. The objective typically maximizes factors such as expected return and minimizes costs like financial risk. The main functionality of this project is to find the correlation among the components of the portfolio and assist the customer in optimizing his/her portfolio in order to receive maximum profit. This project considers the stock data of various US based companies over a time span of five years. It takes into account attributes such as opening price and closing price of the stock, the trading volume, and the changes in stock prices measured over a two-day span. Then, by applying certain machine learning algorithms, an optimal portfolio is built with reduced risks and increased returns.

* 1. **Problem Formulation**

The problem at hand is to maximize profits and at the same time reduce risks during investment. Profit maximization problem is specified as a constrained utility-maximization problem. Often, during investments the classic ideology is that in order to maximize returns, risks taken should be high. Hence, if the risk taken is low, the returns will be low. If the components of the portfolio selected are positively correlated, then the portfolio is considered to be comparatively weak. Positive correlation suggests that returns of all companies will be correlated to each other. Hence, a portfolio can be optimized by having a negative correlation among its components.

* 1. **Motivation**

Portfolio optimization is the process of selecting the best portfolio (asset distribution), out of the set of all portfolios being considered, according to some objective. The objective typically maximizes factors such as expected return, and minimizes costs like financial risk. The basic idea is for a given level of risk, it has to suggest investments assuring that maximum return is achieved. So the optimal portfolio is the one that combines the candidate assets in such a way that, for a given level of risk, the probability of the portfolio earning a positive return is maximized. Different approaches to portfolio optimization measure risk differently.

Portfolio optimization is based on Modern Portfolio Theory (MPT). The MPT is based on the principle that investors want the highest return for the lowest risk. To achieve this, assets in a portfolio should be selected after considering how they perform relative to each other, i.e.; they should have a low correlation. Any optimal portfolio based on the MPT is well-diversified in order to avoid a crash when a particular asset or asset class underperforms.

The standard portfolio optimization problem model is known as the Markowitz’s Mean-Variance portfolio optimization model. The Markowitz model assumes that investors make their decision in portfolio construction by choosing assets that maximize their portfolio returns at the end of the investment period (expected returns). By assuming that investors are risk averse, the simplest model with a number of unrealistic assumptions, namely perfect market without taxes, no transaction costs, no short sales and assets are infinitely divisible. The general methodology for the mean-variance optimization, as discussed by Attilio Meucci, is as follows:

1. Determine market invariants.

2. Estimate distribution of market invariants.

3. Estimate the mean and covariance of the portfolio of assets.

4. Find optimal weights for given objective function.

Market invariants are quantities that do not evolve over time. Market invariants for assets other than stocks are much harder to determine and acquire.Once the data for the market invariants is available, statistical estimation can be performed to find out the distribution and its parameters that best suits the data. This is done usually by maximum likelihood estimation or shrinkage methods. The issue with estimating the distribution is the amount of data available to fit the distribution tables as datasets can be very large and diverse in formats. A way to get around this constraint is to develop novel methods of estimating the distribution of the market invariants. Once an estimate of the distribution of the market invariants is achieved, it is needed to find the mean and covariance of the portfolio of assets. In the case of a portfolio with just stocks, it is trivial as the distribution of the market invariants already give us the mean and covariance of the portfolio.Next step is to find the optimal weights for the portfolio that maximizes some utility function that takes into account the estimates means and covariances.

The following are the limitations of this methodology.Optimal portfolios as outcomes are sensitive to the inputs. Wrongly estimating expected asset returns and volatilities will cause the outcomes to be suboptimal. The optimal portfolios, which use expected returns and volatilities forecasted by models based on historical samples, usually perform poorly out of sample,

The Markowitz model is a simplified model to focus only on a theoretical point of view. In the practicality of investment management, portfolio managers face a number of realistic constraints arising from normal business practices, practical matters and industry regulations. The realistic constraints that are of practical importance include (not exhaustively) integer constraints, cardinality constraints, floor and ceiling constraints, turnover constraints, trading constraints, buy-in threshold, and transaction cost inclusions. Integer constraints require that the number of any asset included in the portfolio must be an integer or indivisible (i.e. cannot be in any fraction of normal trading lot). Cardinality constraints are the maximum number and minimum number of assets that a portfolio manager wishes to include in the portfolio due to monitoring reasons or diversification reasons or transaction cost control reasons. Floor and ceiling constraints define lower and upper limits on the proportion of each asset, which can be held in a portfolio. These constraints may result from institutional policy in order to diversify portfolio and to rule out negligible holding of assets for ease of control. Turnover constraints impose an upper bound for variations of the asset holding from one period to the next. The constraints are a means to curb the transaction costs therefore they can be modelled indirectly by incorporating transaction costs. Trading constraints impose limits on buying and selling tiny quantities of assets due to practical reasons. Through this project we will enable the users to make better choices on investments and thus help the business sector to grow.

* 1. **Proposed Solution**

For optimizing the portfolio, optimization methods used includes classical mean-variance optimization techniques and Black-Litterman allocation, as well as more recent developments in the field like shrinkage and Hierarchical Risk Parity, along with some novel experimental features like exponentially-weighted covariance matrices.Then the expected\_return and sample\_covariance is computed. The expected\_returns module provides functions for estimating the expected returns of returns for each asset. This must be positive semidefinite, otherwise optimization will fail. Next the Sharpe ratio is computed. It Maximises the Sharpe Ratio. The result is also referred to as the tangency portfolio, as it is the portfolio for which the capital market line is tangent to the efficient frontier. Once the sharpe ratio is calculated, stocks are displayed to the user. The user then enters the amount to be invested in stocks along with choice of market capitalization. Then an optimal portfolio will be created for the user automatically.

* 1. **Scope of the Project**

This project focuses on maximizing profits by portfolio optimization. It will benefit the business sector greatly and it will help the users to make wise investment choices. To whichever field applied like finance, stock market, small/large businesses, companies with or without chains etc profits can be maximized depending on their attributes. The goal of portfolio optimization is to help organizations deliver maximum business value. Processes such as prioritization and managing resource capacity help us to “make the best or most effective use of human resources”. In this way, organizations can increase business value delivery.

**Chapter 2**

**Review of Literature**

**Paper 1: - Prediction-based portfolio optimization models using deep neural networks.**

According to this paper in order to optimize your portfolio three deep neural network (DNN) methods have been used-

* Deep multilayer Perceptron (DMLP)
* Long Short memory (LSTM) neural network
* Convolutional Neural Network (CNN)

Freitas et al proposed a novel portfolio optimization model, which used an auto regressive neural network to predict expected returns and applied predictive errors for portfolio optimization. Experimental results showed that this approach outperformed the traditional mean-variance model. It was also mentioned that prediction of future stock return gave better performance of their model.

This particular approach uses DMLP, LSTM and CNN to build prediction-based portfolio optimization models which have the advantage of both deep learning technology and modern portfolio theory (MPT). DNNS is first used to predict each stock’s future returns. The predictive errors obtained from DNNS are given as input in order to calculate the risks associated with each stock. Then the portfolio optimization models are built by integrating predictive returns and deviation calculated from predictive errors. The stocks selection is done by DMLP, LSTM, CNN and then compared with a few equal weighted portfolios.

The paper integrated components of the China security 100 index in Chinese stock market as the experimental dataset. Experimental results suggest that the model built using DMLP performs the best among all the models under different desired portfolio returns and a highly desired portfolio return can further improve the performance of this model.

**Paper 2: - Portfolio Optimization using Machine Learning.**

This paper has suggested the use of standard MPT along with the use of Black-Litterman model to optimize your portfolio.

The Black-Litterman asset allocation model, created by Fischer Black and Robert Litterman is a sophisticated portfolio construction method that overcomes the problem of unintuitive, highly-concentrated portfolios, input-error and estimation error maximization. These three related and well-documented problems with mean-variance optimization are the most likely reasons that more practitioners do not use the Markowitz paradigm, in which return is maximized for a given level of risk. The Black-Litterman model uses a Bayesian approach to combine the subjective views of an investor regarding the expected returns of one or more assets with the market equilibrium vector of expected returns (the prior distribution) to form a new, mixed estimate of expected returns.

The BL model starts from a neural position using MPT, and then takes additional input from investors’ views to determine how the ultimate asset allocation should deviate from the initial portfolio weights. It undergoes a process of mean-variance optimization (MVO) to maximize expected return given one’s objective risk tolerance.

The BL model was designed to improve on the MPT model as one of the limitations of MPT is that it assumes the past expected returns will continue into the future.

**Paper 3: - Deep Learning for Portfolio Optimization.**

In this paper, a portfolio management framework is developed based on a deep reinforcement learning framework called DeepBreath. The DeepBreath methodology combines a restricted stacked autoencoder and a convolutional neural network (CNN) into an integrated framework. The restricted stacked autoencoder is employed in order to conduct dimensionality reduction and features selection, thus ensuring that only the most informative abstract features are retained.

The CNN is used to learn and enforce the investment policy which consists of reallocating the various assets in order to increase the expected return on investment. The framework consists of both offline and online learning strategies: the former is required to train the CNN while the latter handles concept drifts i.e. a change in the data distribution resulting from unforeseen circumstances. These are based on passive concept drift detection and online stochastic batching. Settlement risk may occur as a result of a delay in between the acquisition of an asset and its payment failing to deliver the terms of a contract. In order to tackle this challenging issue, a blockchain is employed.

Finally, the performance of the DeepBreath framework is tested with four test sets over three distinct investment periods. The results show that the return of investment achieved by our approach outperforms current expert investment strategies while minimizing the market risk.

**Chapter 3**

**System Analysis**

**3.1** **Functional Requirements**

The various functional requirements of the system can be summarized as: -

* A homepage that is user friendly and descriptive.
* Clients can add portfolios.
* Clients can edit portfolios.
* Clients can delete portfolios.
* Clients can view the available stocks.
* Clients can check returns and balance amount if any, after investment.
* Clients can view stock details.
* The Web will provide us with a dataset of all companies and stocks.
* System can display stock information.
* System can optimize portfolios.

**Dataset:**

Investment firms nowadays are in the race of developing sophisticated algorithms for stocks trading. Whether it is about stock price prediction, stock market sentiment analysis or Equity research, they need a large volume of accurate data. It is often the case that they have the capital to hire a troop of developers. For independent researchers to predict the stock market, web scraping dataset is an affordable method to obtain the data at scale effortlessly. For accurate and viable information we web scrapped from Yahoo Finance api.

Libraries in python:

* Numpy
* Pandas
* Streamlit
* PyPortfolioOpt
* Plotly
* DiscreteAllocation
* EfficientFrontier
* IEXStock

**3.2** **Non-Functional Requirements**

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements.

• The processing of each request should be done within 10 seconds.

• The site should load in 5 seconds when the number of simultaneous users are > 10000

• The system should provide better accuracy and optimized results.

• Easy to use and user friendly.

* 1. **Specific Requirements**

**3.3.1. User Interfaces.**

The new system shall provide a very intuitive and simple interface to the user so they can easily navigate through software. The Client can Add Portfolio, Edit Portfolio, Delete Portfolio, View the available stock, view stock details, Check returns & Balance amount. The system can Display Stock information and Give an Optimized Portfolio.

**3.3.2. Hardware Interfaces.**

Hardware

The hardware environment consists of the following:

CPU : Intel Pentium IV 600MHz or above

Motherboard : Intel 810 or above

Hard disk space : 20GB or more

Display : Color Monitor

Memory : 128 MB RAM

Other Devices : Keyboard, mouse.

a) Server side

The web application will be hosted on a web server which is listening on the web standard port, port 80.

b) Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse

buttons. The mouse shall activate areas for data input, command buttons and select options from menus.

Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

**3.3.3. Software interfaces.**

Development Tools:

Front End : Steamlit

Back End : Python

Connection : Localhost

Operating System : Windows 10

The actual program that will perform the operations is written in Python.

**3.3.4. Communications interfaces**

The HTTP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

**3.4** **Use-Case Diagrams and Description**

**Use-Case diagram:**

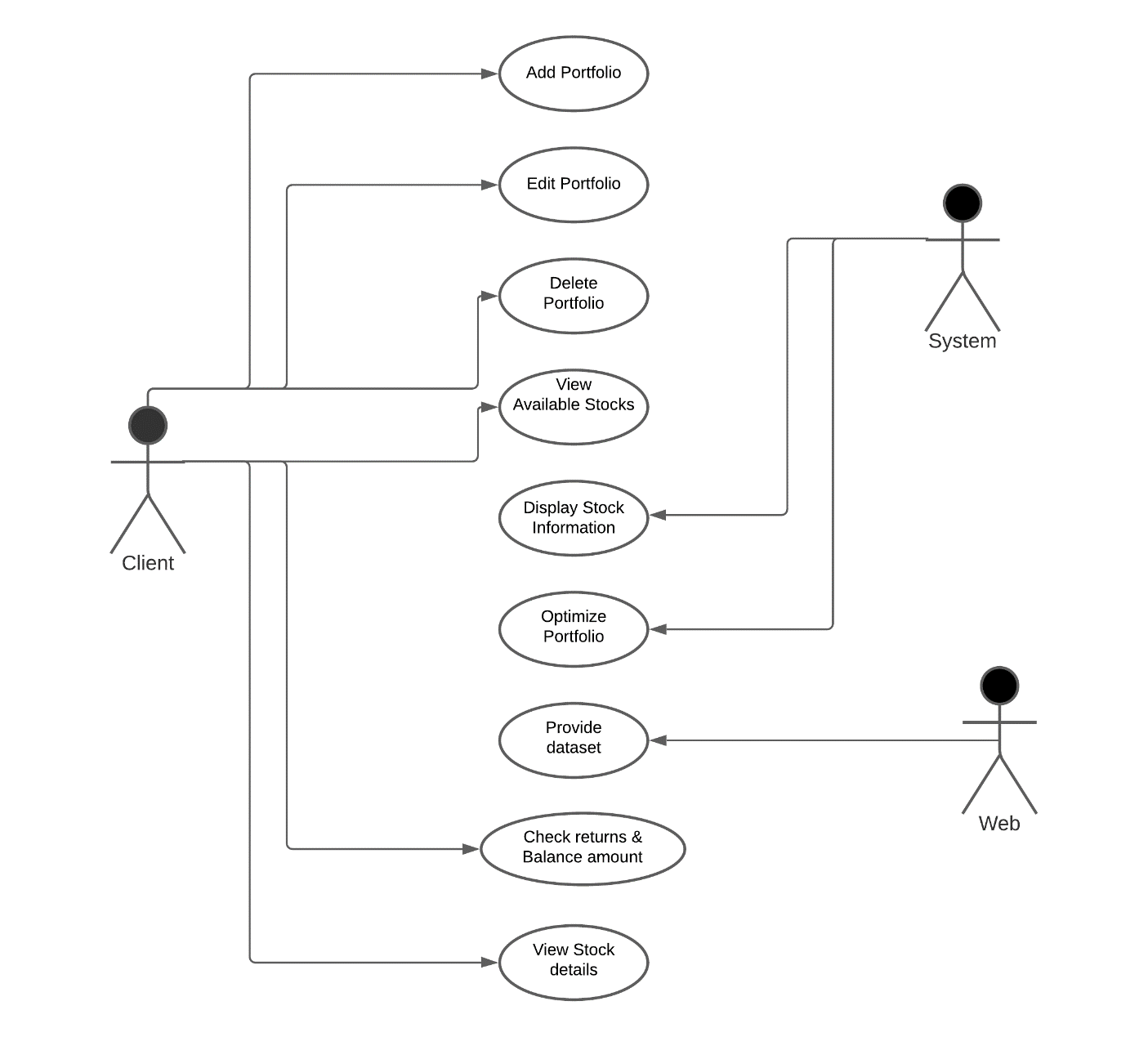


Fig 3.1 Use-Case Diagram for Portfolio Optimization

**Use-Case Description**:

In this Use Case diagram, we have three actors Client, Web and System. The web provides us with the dataset of the companies and stock details. The Client can Add Portfolio, Edit Portfolio, Delete Portfolio, View the available stock, view stock details, Check returns & Balance amount. The system can Display Stock information and Give an Optimized Portfolio.

**Chapter 4**

**Analysis Modeling**

**4.1** **Activity Diagram**

Activity diagram is flow of functions without trigger mechanism, state machine is consisting of triggered states.

Following Activity diagram created for use cases like Add Portfolio,Manage portfolio, Delete Portfolio, view available stock, view news related to all stocks and gain optimised portfolio.

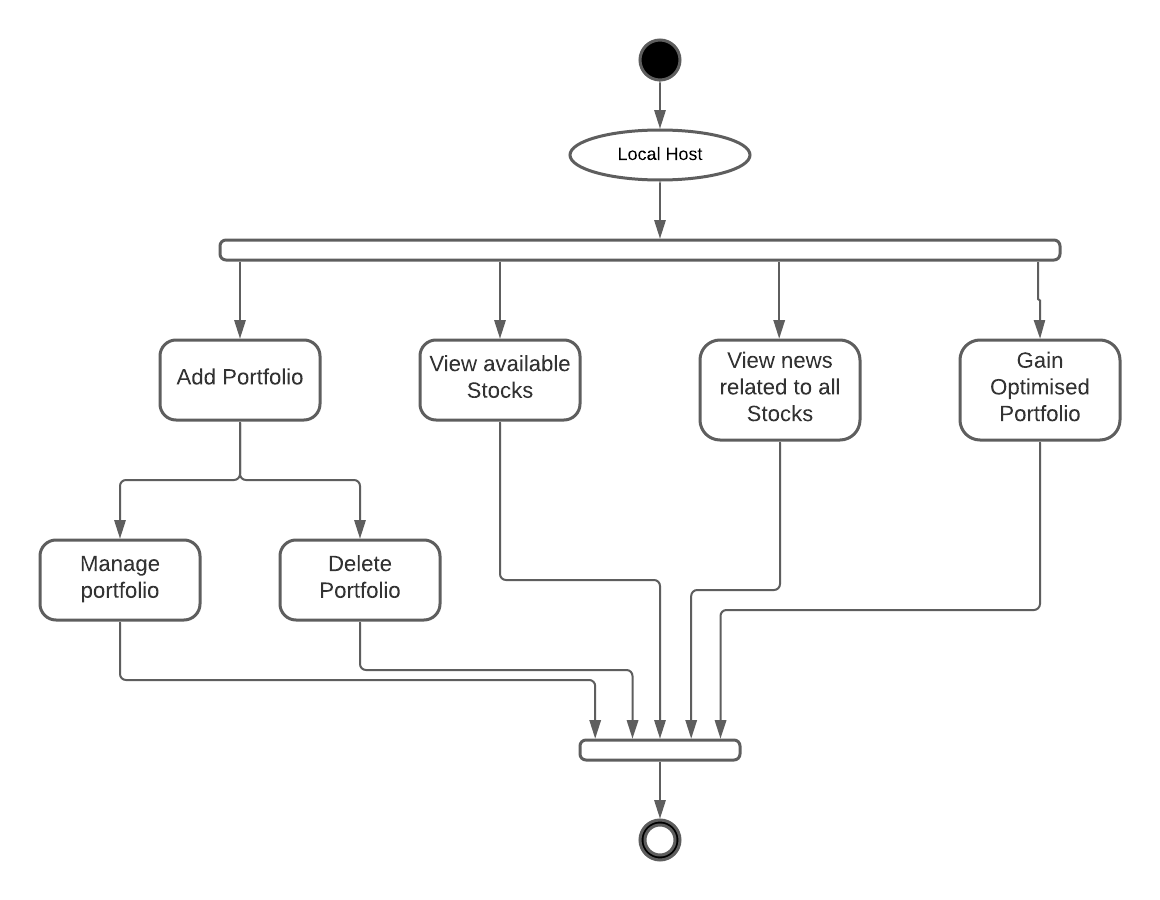
****

Fig 4.1 Activity Diagram for Portfolio Optimization

**4.2** **Functional Modelling**

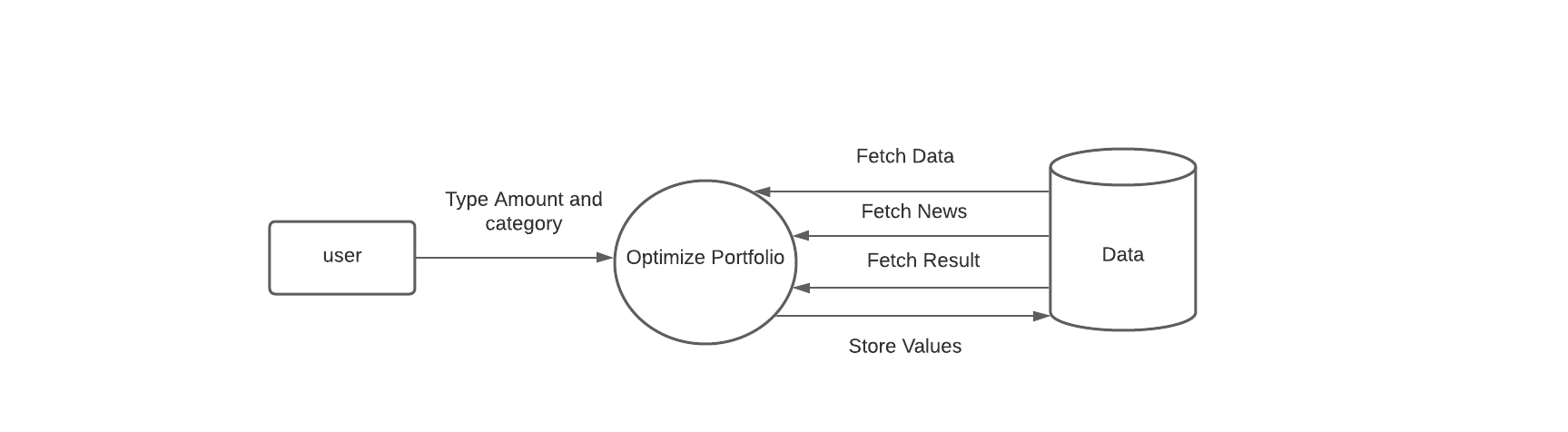
Portfolio optimization.****

Fig 4.2 Data Flow Diagram for Portfolio Optimization

**Chapter 5**

**Design**

**5.1** **Architectural Design**

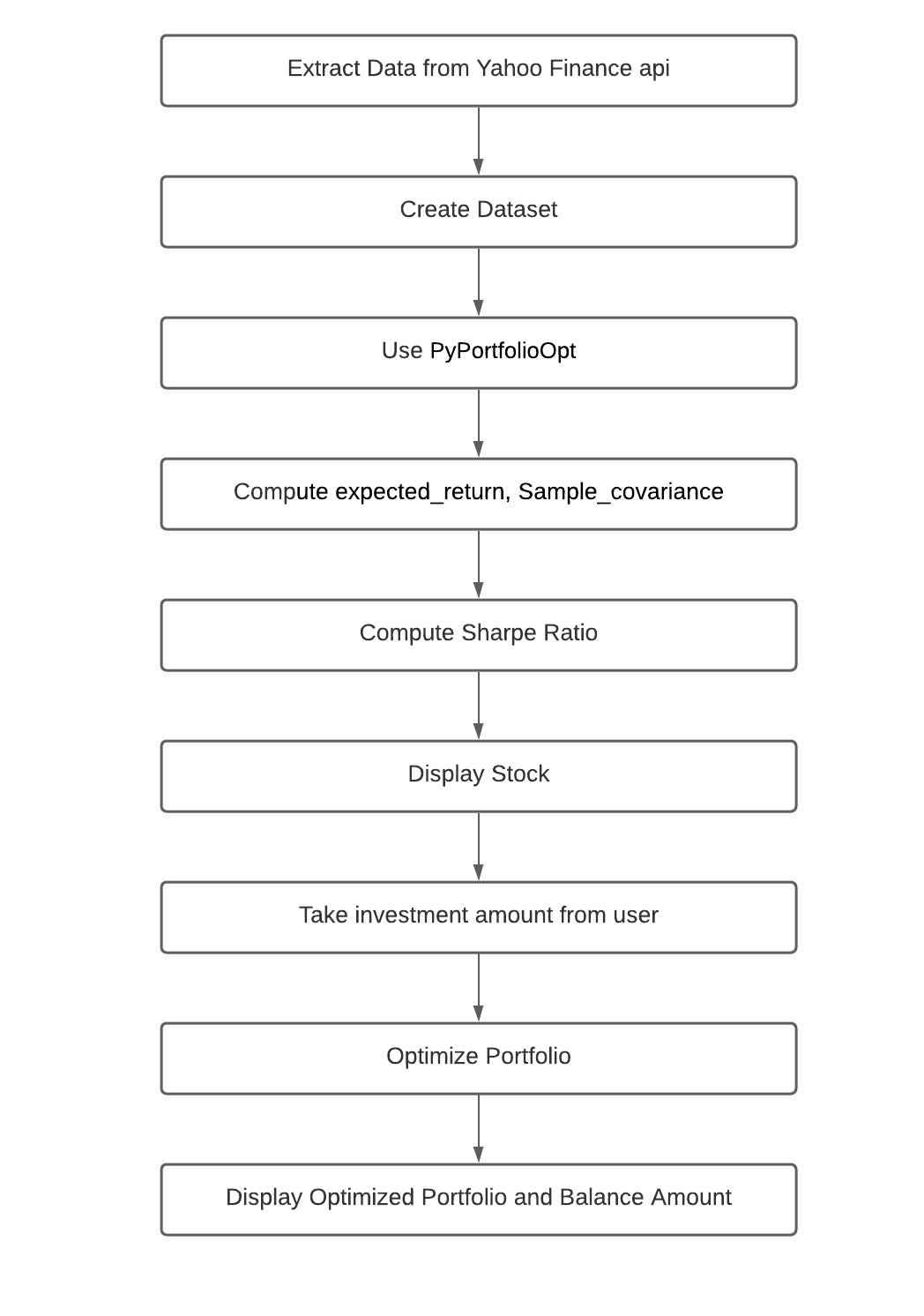
****

Fig 5.1 Block Diagram for Portfolio Optimization

The project flow starts with web scraping data from Yahoo Finance api to create a dataset. Then once the dataset is created we use PyPortfolioOpt library in Python to implement portfolio optimization. PyPortfolioOpt is a library that implements widely-used classical portfolio optimisation techniques, with a number of experimental features. Then we compute the expected\_return and sample\_covariance.The expected\_returns module provides functions for estimating the expected returns of returns for each asset. This must be positive semidefinite, otherwise optimization will fail. Next the Sharpe ratio is computed. It Maximises the Sharpe Ratio. The result is also referred to as the tangency portfolio, as it is the portfolio for which the capital market line is tangent to the efficient frontier. Once the sharpe ratio is calculated, stocks are displayed to the user. The user then enters the amount to be invested in stocks along with choice of market capitalization. Then an optimal portfolio will be created for the user automatically.

**5.2** **User Interface Design**

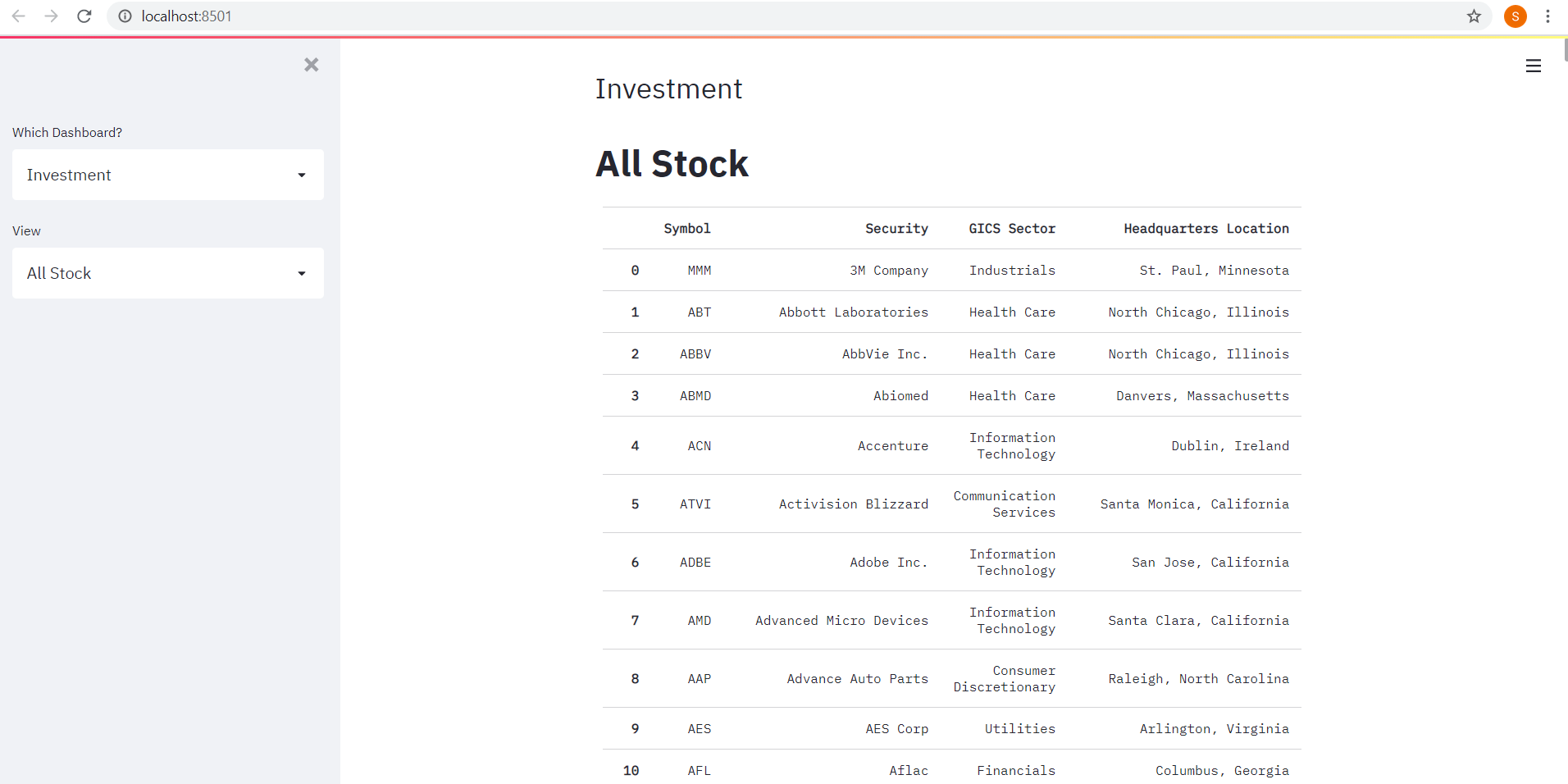
****

Fig. 5.2.1 View all stock

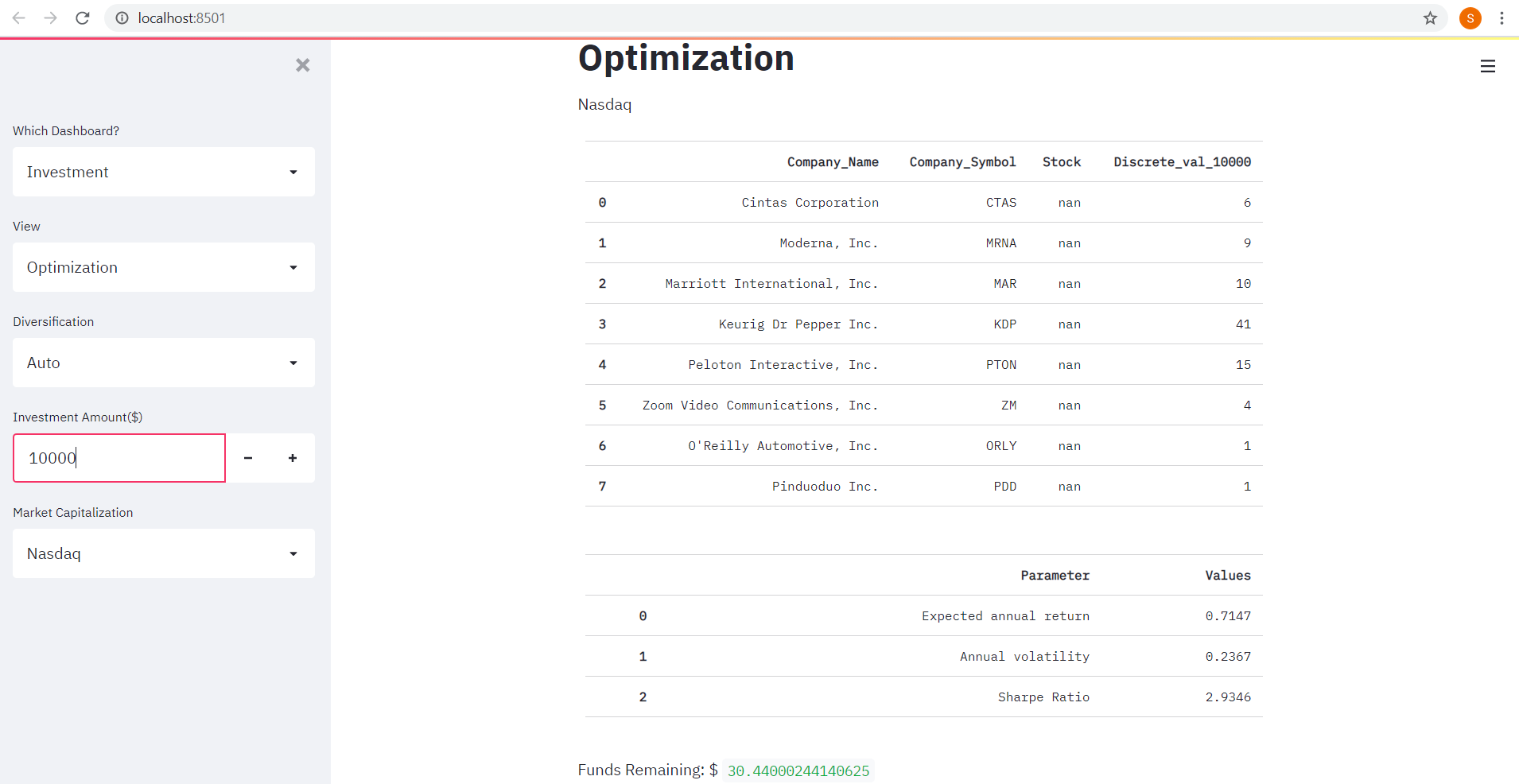
****

Fig. 5.2.2 Portfolio optimization for Nasdaq

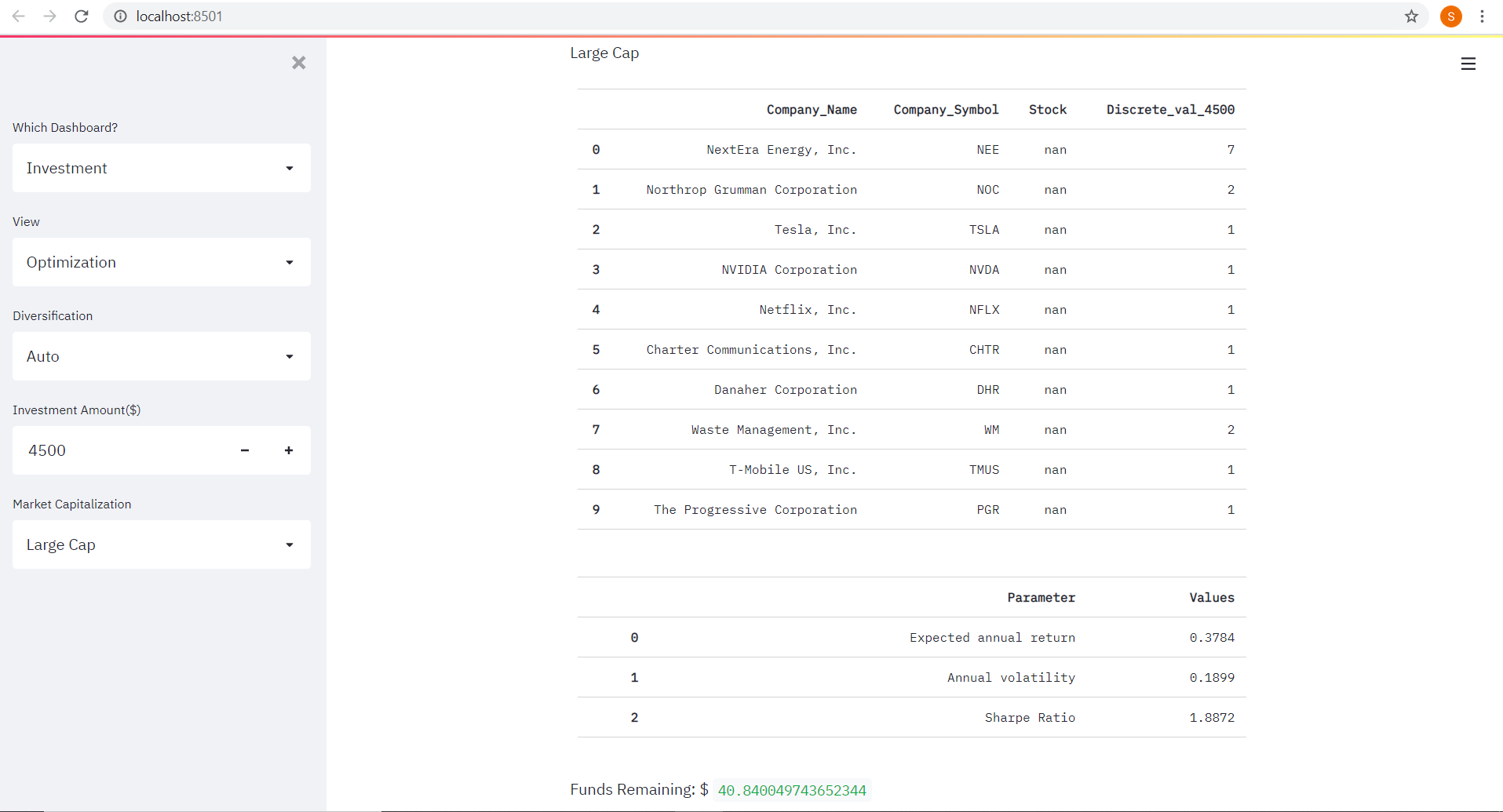
****

Fig. 5.2.3 Portfolio optimization for Large Cap

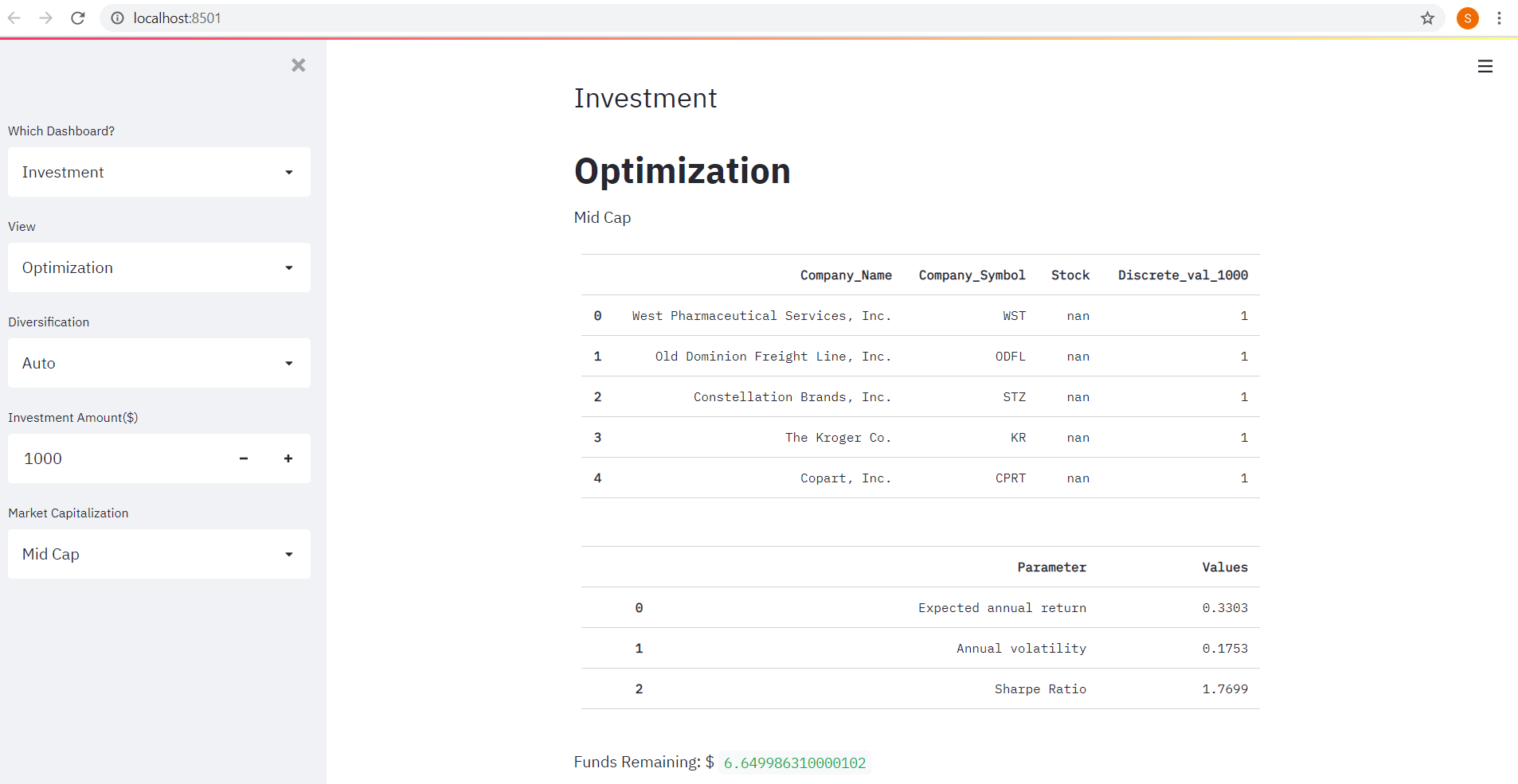
****

Fig. 5.2.4 Portfolio optimization for Mid Cap

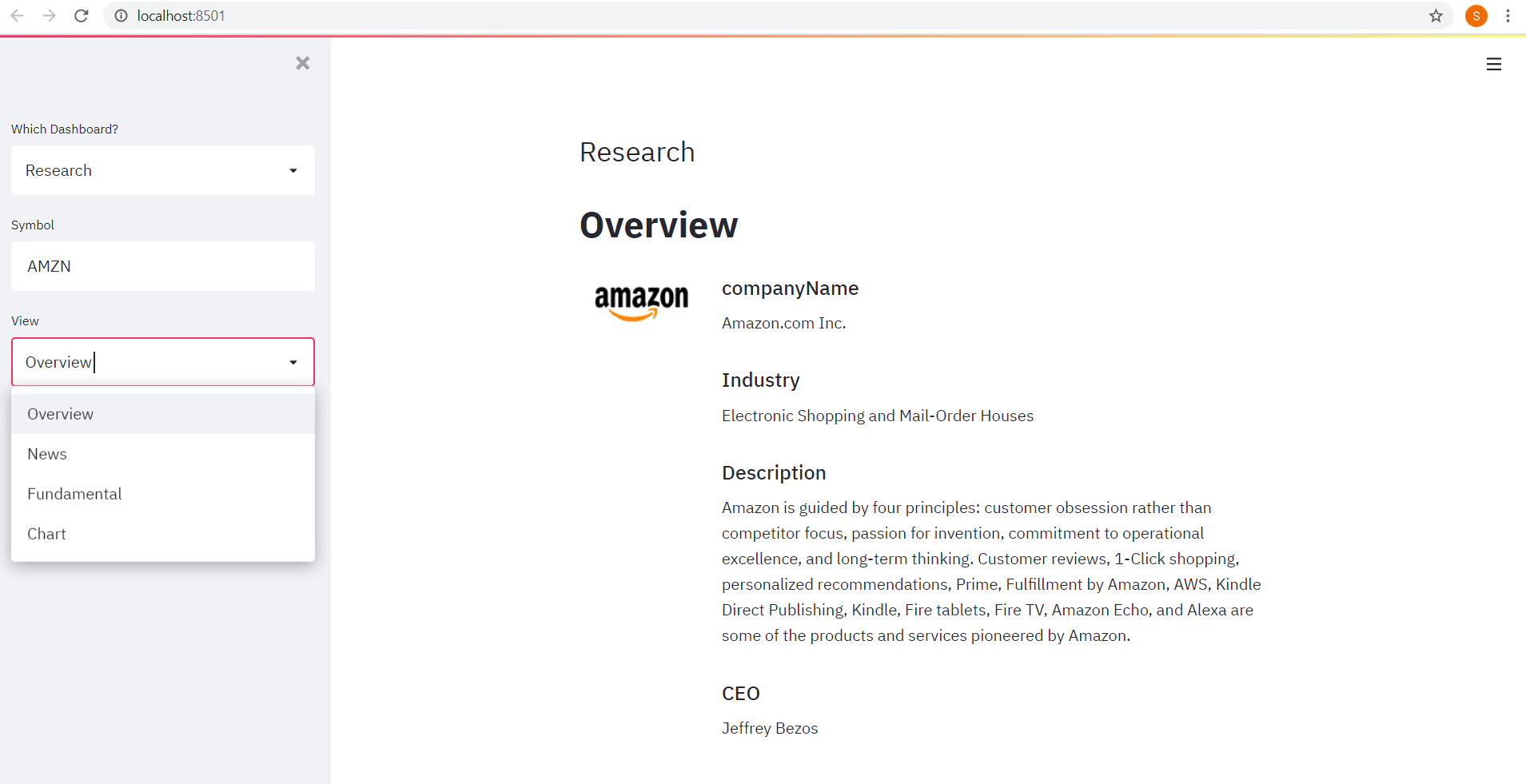
****

Fig. 5.2.5 Stock Overview

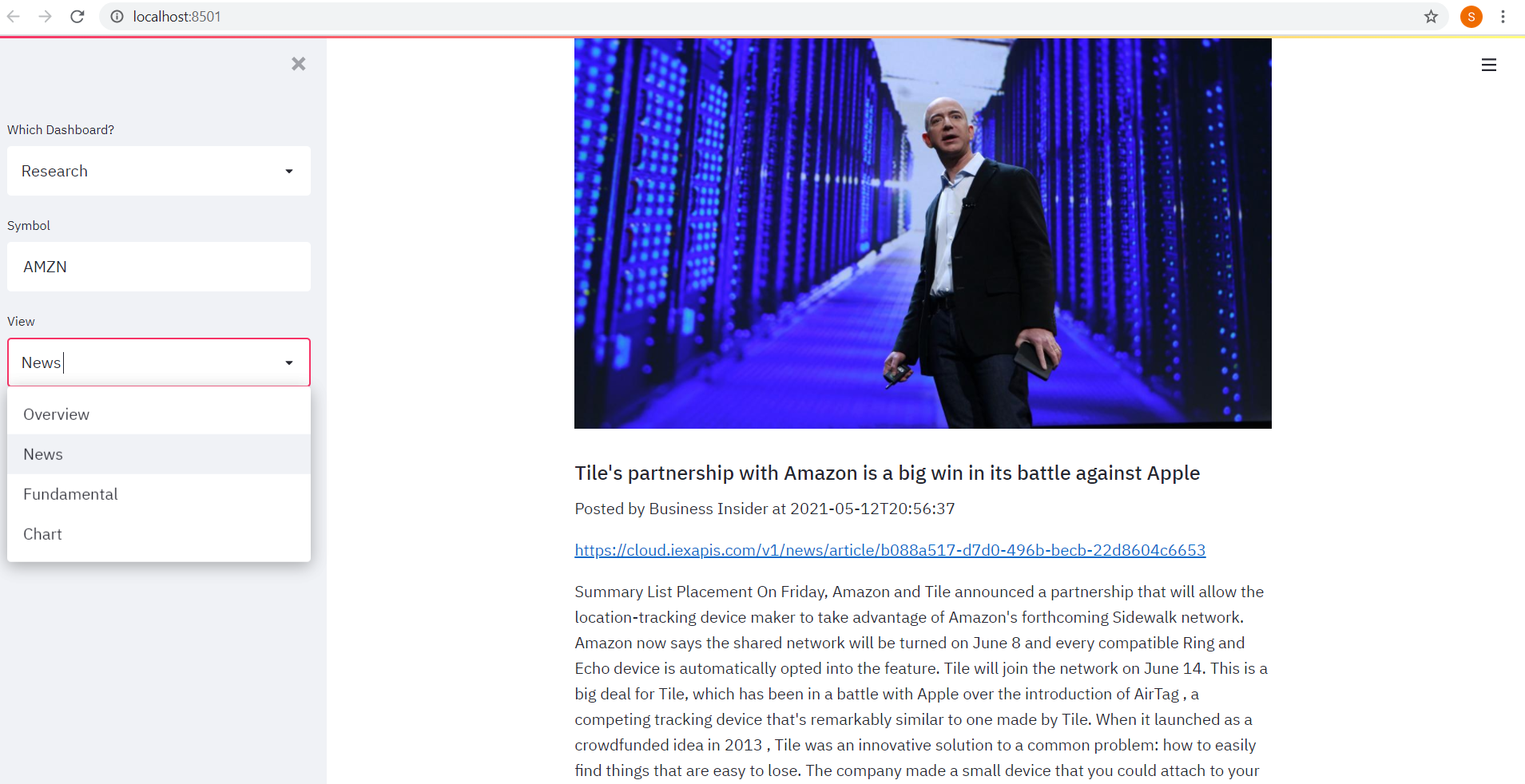
****

Fig. 5.2.6 Stock latest news

**Chapter 6**

**Implementation**

**6.1** **Algorithms/Methods used**

* For optimizing the portfolio, an inbuilt function called PyPortfolioOpt was used.
* PyPortfolioOpt is a library that implements portfolio optimization methods, including classical mean-variance optimization techniques and Black-Litterman allocation, as well as more recent developments in the field like shrinkage and Hierarchical Risk Parity, along with some novel experimental features like exponentially-weighted covariance matrices.
* It first computes the mean- variance (MV) and the sample covariance.
* Mean-variance optimization spearheaded the transformation of portfolio management.
* If w is the weight vector of stocks with expected returns μ, then the portfolio return is equal to each stock’s weight multiplied by its return, i.e wTμ. The portfolio risk in terms of the covariance matrix Σ is given by wTΣw. Portfolio optimization can then be regarded as a convex optimization problem, and a solution can be found using quadratic programming. If we denote the target return as μ∗, the precise statement of the long-only portfolio optimization problem is as follows:

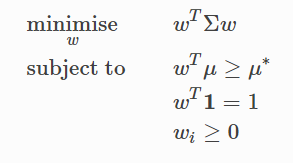


Fig 6.1.1 Weight Calculation

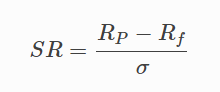
* If we vary the target return, we will get a different set of weights (i.e a different portfolio) – the set of all these optimal portfolios is referred to as the efficient frontier.
* The sharpe ratio is calculated and based on all these parameters the portfolio is optimized.
* The Sharpe ratio is the portfolio’s return in excess of the risk-free rate, per unit risk (volatility).
* 

Fig 6.1.2 Sharpe Ratio

* It is particularly important because it measures the portfolio returns, adjusted for risk. So in practice, rather than trying to minimise volatility for a given target return, it often makes more sense to just find the portfolio that maximises the Sharpe ratio. This is implemented as the max\_sharpe() method in the EfficientFrontier class.
* Features of PyPortfolioOpt: -

1. It is extensive yet easily extensible, and can be useful for both the casual investor and the serious practitioner.
2. Exponentially weighted mean historical returns
3. Minimum Covariance Determinant
4. Sample covariance matrix

Python and its libraries allow us to automate optimization and save valuable time in the process of doing so.

**6.2** **Working of the project:**

**Portfolio maximization working**

**def black\_litterman(df, cash):**

**assets = df.columns**

**#calculate annualized returns**

**mu = expected\_returns.mean\_historical\_return(df)**

**S = risk\_models.sample\_cov(df)**

**#optimize for the maximum sharpe ratio**

**ef = EfficientFrontier(mu, S)**

**weight = ef.max\_sharpe()**

**#Round off values**

**cleaned\_weights = ef.clean\_weights()**

**#Calculate the portfolio performance**

**returns = ef.portfolio\_performance(verbose=True)**

**r\_df = pd.DataFrame(columns=['Parameter', 'Values'])**

**r\_df['Parameter'] = ['Expected annual return','Annual volatility','Sharpe Ratio']**

**r\_df['Values'] = returns**

**portfolio\_val=cash**

**latest\_prices=get\_latest\_prices(df)**

**weights = cleaned\_weights**

**#Discrete Allocation using greedy\_portfolio method**

**da = DiscreteAllocation(weights, latest\_prices, total\_portfolio\_value = portfolio\_val)**

**allocation, leftover = da.greedy\_portfolio()**

**#Store the company name into list**

**company\_name = []**

**for symbol in allocation:**

**company\_name.append( get\_company\_name(symbol))**

**#Get the discrete allocation list**

**discrete\_allocation\_list = []**

**for symbol in allocation:**

**discrete\_allocation\_list.append( allocation.get(symbol))**

**#create dataframe for portfolio**

**portfolio\_df = pd.DataFrame(columns=['Company\_Name', 'Company\_Symbol', 'Allocation\_for\_$'+str(portfolio\_val)])**

**portfolio\_df['Company\_Name'] = company\_name**

**portfolio\_df['Company\_Symbol'] = allocation**

**portfolio\_df['Allocation\_for\_$'+str(portfolio\_val)] = discrete\_allocation\_list**

**#Display the output**

**st.table(portfolio\_df)**

**st.table(r\_df)**

**st.write('Funds Remaining: $', leftover)**

**Chapter 7**

**Conclusion**

Through this project, portfolio diversification and allocation are explored. Background study was done on all the different approaches. The project flow starts with web scraping data from Yahoo Finance API to create a dataset. Then once the dataset is created portfolio optimization was implemented using classical portfolio optimisation techniques, with a number of experimental features. Then the expected\_return and sample\_covariance were computed.The expected\_returns module provides functions for estimating the expected returns of returns for each asset. This must be positive semidefinite, otherwise optimization will fail. Next the Sharpe ratio is computed. It maximises the Sharpe Ratio. The result is also referred to as the tangency portfolio, as it is the portfolio for which the capital market line is tangent to the efficient frontier. Once the sharpe ratio is calculated, stocks are displayed to the user. The user then enters the amount to be invested in stocks along with choice of market capitalization. Then an optimal portfolio will be created for the user automatically.The frontend for this project was created using Streamlit. This project focuses on maximizing profits by portfolio optimization. It will benefit the business sector greatly and it will help the users to make wise investment choices. Portfolio Optimization finds its application in many fields like banking sector, finance, healthcare, IT, forign exchange, etc.This method is highly efficient and can be used in various fields like finance, healthcare, business, etc.

**References**

[1] R.A. Haugen, N.L. Baker, Dedicated stock portfolios, Journal of Portfolio Management, 16 (1990), 17-22.

[2] T-J Chang, Nigel Meade, John E Beasley, and Yazid M Sharaiha. 2000. Heuristics for Cardinality Constrained Portfolio Optimisation. *Computers & Operations Research* 27, 13, 1271--1302.

[3] P. Xidonas, G. Mavrotas, J. Psarras, Portfolio construction on the Athens Stock Exchange: A multiobjective optimization approach, Optimization, 59 (2010), 1211-1229.

[4] Francesco Cesarone, Andrea Scozzari, and Fabio Tardella. 2009. Efficient Algorithms for Mean-Variance Portfolio Optimization With Hard Real-World Constraints. *Giornale dell'Istituto Italiano degli Attuari* 72, 37--56

[5] R. Mansini, W. Ogryczak, M.G. Speranza, Portfolio optimization and transaction costs. In Quantitative financial risk management: Theory and practice, Oxford: Wiley (2015), 212-241.

**Acknowledgements**

We express our deep sense of gratitude to our project guide Mrs. Vincy Joseph for encouraging us and guiding us throughout this project. We were able to successfully complete this project with the help of her deep insights into the subject and constant help.

We are very much thankful to Dr. .Kavita Sonawane, HOD of the Computer Engineering Department at St. Francis Institute of Technology for providing us with the opportunity of undertaking this project which has led to us learning so much in the domain of Machine Learning.

Last but not the least we would like to thank all our peers who greatly contributed to the completion of this project with their constant support and help.