AGENTIC AI FOR SMART PORTFOLIO MANAGEMENT: A FUSION OF ML AND NATURE INSPIRED ALGORITHMS

A PROJECT REPORT

Major Project – II (01CE0807)

Submitted by

HELI HATHI
92100103341

HET BUCH
92100103196

BACHELOR OF TECHNOLOGY
in
Computer Engineering

Faculty of Technology
Marwadi University, Rajkot

April, 2025

Marwadi University
Marwadi Chandarana Group
NAAC A+

Major Project-II (01CE0807)

Department of Computer Engineering

Faculty of Technology

Marwadi University

A.Y. 2024-25

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Agentic AI for Smart Portfolio Management: A Fusion of ML and Nature Inspired Algorithms** has been carried out by **Heli Hathi** (92100103341), **Het Buch** (92100103196) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 8th Semester of Marwadi University, Rajkot during the academic year 2024-25.

Prof. Ravikumar R. Natarajan	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project-II (01CE0807)

Department of Computer Engineering

Faculty of Technology

Marwadi University

A.Y. 2024-25

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Agentic AI for Smart Portfolio Management: A Fusion of ML and Nature Inspired Algorithms** has been carried out by **Heli Hathi** (92100103341) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 8th Semester of Marwadi University, Rajkot during the academic year 2024-25.

Prof. Ravikumar R Natarajan	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Major Project-II (01CE0807)

Department of Computer Engineering

Faculty of Technology

Marwadi University

A.Y. 2024-25

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Agentic AI for Smart Portfolio Management: A Fusion of ML and Nature Inspired Algorithms** has been carried out by **Het Buch** (92100103196) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 8th Semester of Marwadi University, Rajkot during the academic year 2024-25.

Prof. Ravikumar R Natarajan	Dr. Krunal Vaghela
Assistant Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

Marwadi University
Marwadi Chandarana Group
NAAC A+

Major Project-II (01CE0807)

Department of Computer Engineering

Faculty of Technology

Marwadi University

A.Y. 2024-25

DECLARATION

We hereby declare that the **Major Project-II (01CE0807)** report submitted along with the Project entitled **Agentic AI for Smart Portfolio Management: A Fusion of ML and Nature Inspired Algorithms** submitted in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering to Marwadi University, Rajkot, is a bonafide record of original project work carried out by us at Marwadi University under the supervision of **Prof. Ravikumar R. Natarajan** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

	Name of the Student	Sign of Student
1	Heli. Hathi	Heli
2	Het Buch	H.J. Buch

Acknowledgement

We would like to express our sincere gratitude to our mentor, Prof. Ravikumar R. Natarajan, for his invaluable guidance, insightful suggestions, and continuous support throughout the course of this project. His feedbacks have played an important role in polishing our work. The successful completion of this project would not have been possible without his invaluable insights.

We are also deeply grateful to Dr. Krunal Vaghela, Professor and Head of the Department of Computer Engineering at Marwadi University, for his constant support and for creating an environment of academic collaboration.

We also extend our appreciation to our friends whose constant feedbacks and discussions helped our understanding and refine our project.

Heli Hathi

Het Buch

Abstract

In financial markets, investors want reliable solutions to optimize their portfolios with minimum risk and maximum returns. In this project, we present an intelligent portfolio optimization system that uses machine learning, nature-inspired algorithms, and Agentic AI to improve investment decision-making. The proposed system includes stock price prediction, sentiment analysis, and risk assessment, to generate optimized investment portfolios according to user preferences. For predicting stock prices and identifying investment opportunities, we used regression algorithms like Ridge regression, XGBoost, etc. These algorithms predicted stock prices with minimum error by understanding complex patterns from historical data. Then optimization algorithms like PSO (Particle Swarm Optimization), GWO (Grey Wolf Optimization), and the Bat Algorithm, were used to create portfolios that balances risks and returns by measuring Sharpe Ratio and expected risk percentage. Among these, PSO delivered the highest returns, a high Sharpe Ratio, with low risks. The use of Agentic AI in portfolio optimization provides autonomous decision-making based on nature of markets. By utilizing two intelligent agents, the system continuously analyses financial data, assess portfolio performance, and suggest real-time adjustments to maximize returns according to user-defined risk parameters. This approach enhanced system to perform well in volatile market conditions. To enhance user experience and accessibility, we developed a web-based application using Streamlit. This dashboard supports automated portfolio rebalancing and real-time analytics making investment strategies more accessible.

List of Figures

List of Tables

Abbreviations

AI	**Artificial Intelligence**
ML	**Machine Learning**
XGBoost	**Extreme Gradient Boosting**
LightGBM	**Light Gradient Boosting Machine**
PSO	**Particle Swarm Optimization**
GWO	**Grey Wolf Optimization**
SVR	**Support Vector Regression**
LSTM	**Long Short-Term Memory**
CART	**Classification And Regression Techniques**
DNN	**Deep feed-forward Neural Network**
GA	**Genetic Algorithms**
GRU	**Gated Recurrent Unit**
LLM	**Large Language Models**
CNN	**Convolutional Neural Network**
MV	**Mean-Variance**
RL	**Reinforcement Learning**
DLEF-SM	**Deep Learning based Expert Frame-work for Stock Market forecasting**
DL	**Deep Learning**
BiGRU	**Bi-directional Gated Recurrent Unit**
BiLSTM	**Bi-directional Long Short-Term Memory**
MSE	**Mean Squared Error**
HGA-MT	**Heterogeneous Graph Attention network with Multi-Tasking model**
WOA	**Whale Optimization Algorithm**
SE	**Stock Exchange**
IBWO	**Improved Black Widow Optimization**
Nifty50	**National Stock Exchange Fifty**
Sensex	**Stock Exchange Sensitive Index**
TA-Lib	**Technical Analysis Library**
ER	**Entity – Relationship**
JSON	**JavaScript Object Notation**

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 PROJECT SUMMARY

In this project, we present an intelligent portfolio optimization framework that uses machine learning and nature-inspired algorithms along with Agentic AI to enhance investment decision-making. Our approach combines stock price forecasting, sentiment analysis from news websites along with portfolio report and risk assessment to generate optimized portfolios. Firstly, we applied regression algorithms such as Gradient Boosting, Ridge Regression, XGBoost, and LightGBM to model stock price trends and identify potential investment opportunities. Following this, we used optimization techniques such as PSO, GWO and Bat Algorithm. The proposed system enables users to review AI-generated portfolio suggestions, and make informed investment decisions. Finally, a web-based application was developed using Streamlit for seamless user interaction and automated portfolio rebalancing.

1.2 PURPOSE

The purpose of this project is to develop an intelligent portfolio management system to help investors make data-driven investment decisions. Traditional strategies struggle with market volatility, and risk management, making it difficult for investors. By integrating stock price forecasting, sentiment analysis, and risk assessment, this project provides an AI-based investment system. The goal is to enable users to set risk preferences, receive AI-generated portfolio recommendations, and automate portfolio rebalancing, thereby making investment strategies adaptive and accessible.

1.3 OBJECTIVE

Traditionally, investors analyse factors like economic events and historical trends to forecast stock returns. But with the rise of AI, techniques based on AI have slowly becoming more effective than traditional methods [1]. Because of nonlinear nature of financial markets, selecting stocks with higher returns remains a major challenge [4]. While stock predictions help in identifying profitable opportunities, traditional techniques mostly prioritize strategic performance over asset selection, even though good stock selection is crucial for long-term success [9][11]. In this project, we compare 25 ML algorithms for stock price prediction, then the best performing algorithm will be used for price prediction. The nature-inspired optimization techniques are used to balance risk and return in portfolio optimization. This project uses Agentic AI, which enables the system to autonomously analyse market trends, optimize investment strategies in real time, and adapt to fluctuations. Unlike traditional methods, Agentic AI continuously learns, adjusts portfolios, and suggests investment decisions using sentiment analysis from financial news for a comprehensive market understanding.

1.4 SCOPE

This scope of this project is to optimize investment portfolios, and assist investors in making informed decisions based on AI-driven insights. It allows users to set their risk preferences, analyse AI-generated portfolio suggestions, and track market trends through an interactive web-app. The system continuously analyses financial data, assess portfolios, and suggest optimal adjustments based on market conditions. While the system provides accurate stock predictions and optimized portfolio recommendations, it does not guarantee profits, as financial markets remain unpredictable. The system also does not execute trades automatically – users must manually implement the recommended strategies. Additionally, while sentiment analysis helps incorporate market reactions, the system cannot predict unforeseen global events such as financial crises or political disruptions.

1.5 LITERATURE REVIEW

Traditional investment strategies relied on fundamental analysis and models like Markowitz's Efficient Frontier, but with the rise of deep learning and reinforcement learning, demand for AI-based approaches have increased in portfolio optimization. Table 1.1 shows the detailed review on the recent works in this field. Ref. [1] showed that LSTM and Transformer models are great at capturing long-term dependencies in financial data, leading to better portfolio performance. Ref. [2] introduced a loss function to optimize portfolios by eliminating negative profits, but performance dropped while adjusting risk levels. Ref. [3] combined SVR, CART, DNN and GA for stock prediction and portfolio optimization, outperforming traditional methods despite challenges in tuning. Similarly, Ref. [4] found that GRU models did well in predicting stock returns but were not effective in assessing risk. On the other hand, Ref. [5] used

LLMs and DINNs for optimization, but using in real-time investment scenarios remains a challenge. Ref. [6] applied PSO, CNN, and MV forecasting to improve stock selection accuracy, but was computationally high. Ref. [7] found LSTM and RL models were well-suited for volatile markets, while traditional methods struggled with large datasets. Ref. [8] introduced the DLEF-SM, combining DL and RL for better portfolio optimization, though interpretation of model's decisions was difficult. Ref. [9] discovered that BiGRU in volatile markets achieved a lower MSE, but further testing across different conditions was needed. Ref. [10] proposed a CNN-BiLSTM model that improved stock trend prediction while removing risk, but meta-heuristic optimizations struggled with handling multiple objectives same time. Ref. [11] used GWO for parameter selection, while Ref. [12] introduced the HGA-MT model, which improved risk-adjusted returns. Ref. [13] combined CNN and LSTM to enhance stock selection, but the results were limited to the dataset used. Ref. [14] found that GWO helped optimize asset allocation in emerging markets, while Ref. [15] showed that WOA outperformed PSO in portfolio efficiency. Despite their advantages, these methods often struggled with computational complexity and generalizability. More recent research, such as Ref. [16], demonstrated that LSTMs achieve high forecasting accuracy, while Ref. [17] improved portfolio optimization by combining GWO with the Markowitz model. Ref. [18] introduced a BiLSTM+MV strategy that optimized risk-return trade-offs, and Ref. [19] showed strong cumulative returns in Brazilian stocks. Finally, Ref. [20] applied a hybrid meta-heuristic algorithm for improved risk-adjusted portfolio optimization, but real-world application and market variability were still concerns.

We used a novel approach of Agentic AI, which takes portfolio optimization systems to autonomously learn from market trends and adapt in real time. Unlike traditional AI models that follow predefined rules, Agentic AI actively refines strategies based on present market conditions, making investment decisions more dynamic. By using ML and nature-inspired optimization techniques within this framework, we also improved computational efficiency. Agentic AI enables self-learning agents to analyse patterns, react dynamically to financial fluctuations, with minimal human intervention, making portfolio management more effective in real-world financial applications.

Table 1.1 Literature Review

Ref. No.	Year	Dataset	Key Findings	Merits	Demerits
[1]	2025	Bloomberg, yFinance, Quandl, Kaggle	LSTM, Transformers excel at long-term dependencies.	LSTMs excelled in time-series data, with low error rates.	Real-time data boosts model sensitivity.
[2]	2025	Stocks from S&P 500 index	The loss function performs better by ignoring negative profits.	Proposed method allows for easier interpretation of negative profits.	Sharpe loss doesn't adjust to the portfolio's risk level.
[3]	2025	Stocks from Chinese SE A-share index	Uses ensemble & GA to optimize portfolio weights.	GA optimize portfolio, better than traditional methods.	Constraints in hyper-parameter optimization.
[4]	2024	Yahoo Finance	GRU showed 60% similarity to stocks with the highest actual returns.	The hybrid method achieves better accuracy than conventional methods.	Accurate predictions but failed to assess stock risk effectively.
[5]	2025	Stocks from DOW 30 index	LLMs with DINN has better portfolio optimization.	DINN outperforms DL models indicating better decisions, providing reliable performance.	Portfolio selection methods might not function in real-world scenarios.
[6]	2023	21 stocks from the NYSE	Integrates PSO, CNN and MVF, making efficient portfolios.	Proposed model improves returns compared to other DL models.	Requires high computational resources.
[7]	2022	Bloomberg, yFinance, and Alpha Vantage	LSTM and RL for volatile situations, while SVM and RF for stable markets.	Enhances accuracy, risk management, and returns compared to old methods.	Could not handle large, multi-variate financial data effectively.
[8]	2024	Stocks from S&P 500, and DAX indices	Achieved good accuracy by combining DL and RL.	The IBWO enhances optimization tasks.	DL models struggle with price movements in real-time.
[9]	2024	yFinance	BiGRU achieved MSE of 0.14, in volatile markets.	DL-based optimization outcome improves decision-making.	More testing required across a variety of market scenarios.
[10]	2024	RESSET Database	CNN-BiLSTM outperformed traditional models	Proposed system mitigates risks	Meta-heuristic algorithms struggle with

			in identifying trends.	while enhancing returns.	optimization problems.
[11]	2024	Stocks from Nifty50	GWO-based model improves by 5-8% than traditional models.	GWO minimizes forecasting errors by finding best parameters.	Increased computational complexities.
[12]	2023	Stocks from CSI100 index	Outperforms state-of-the-art models by 11.09% in precision.	The HGA-MT model reduces risk and increased profits.	Explore more relations between stocks to enhance robustness.
[13]	2024	NSE Stocks	Proposed model outperformed other models, by using MV optimization.	Hybrid approach outperforms in performance enhancing selection and optimization.	The model's generalizability is limited to the dataset used.
[14]	2023	Stocks from KSE-30 index	GWO enhances asset allocation proving better investment options.	TOPSIS with Eigenvector helps in ranking alternatives.	Difficulty in making predictions of when investing in new stocks.
[15]	2024	Stocks from S&P/BMV IPC index	WOA outperformed PSO in objective function value.	WOA shows faster convergence enhancing efficiency in optimal portfolios.	The performance is based on mathematical modelling, and not in real-time scenarios.
[16]	2024	yFinance	The model obtained accuracy of 90% in forecasting.	LSTMs handle long-term dependencies overcoming the vanishing gradient problem.	The portfolio testing does not reflect real-world conditions, affecting performance.
[17]	2024	698 Stocks from Tehran SE	GWO has superior performance and optimal solutions.	Combination of GWO & modified Markowitz model results in a better portfolio optimization.	May limit the performance due to high sensitivity of stock market behaviour.
[18]	2024	Stocks from LQ45 index	The BiLSTM outperformed with MAPE of 2.1 and MAE of 104.05.	Highest Sharpe Ratio, indicating optimal risk-return trade-off.	The nonlinearity of stock market might result in inaccurate predictions.
[19]	2024	Stocks from Brazilian SE	The portfolios outperformed the	Practical applications	Absence of fundamental data

			IBOV showing strong returns.	helping in optimal investments.	may compromise results.
[20]	2023	All stocks from Tehran SE	The hybrid algorithm improves adjusted risk in portfolio optimization.	The hybrid model helps in selecting optimal stock across various time series.	The data is limited to 7 years which may not capture long-term market trends.

1.6 PROJECT PLANNING

The planning phase of our project involved breaking the entire project into small modules. We first created a workflow of the project and divided tasks based on our individual strengths. Since Agentic AI is relatively a new concept and we initially lacked familiarity with many stock market terminologies, good amount of time was given to understanding the fundamentals. The steps below outline the breakdown of our project planning process.

1.6.1 Project Development Approach

Instead of working on everything simultaneously, each module was developed independently. Also, the work division allowed us to work in parallel manner rather than sequential, thereby increasing the efficiency and time management.

1.6.2 Project Effort and Time, Cost Estimation

We dedicated first few weeks for research, learning and understanding of the concepts which are to be used. This was followed by design, coding, and deployment. Regular updates and feedbacks were taken from our internal guide to complete the modules in the specified time. In total, around 3–4 person-months were spent on the entire project. No monetary investment was required during making of this project.

1.6.3 Roles and Responsibilities

Our project work was evenly divided based on individual strengths and areas of expertise. The conceptualization of the project was done by Heli. After properly understanding the project, Het carried out a literature survey focusing on the recent studies and challenges in portfolio optimization. He also designed a proper methodology by combining the key aspects of stock markets and machine learning. Heli created detailed diagrams (Class Diagram, Sequence Diagram, etc.) required at each stage of the project. She was also responsible for designing database schemas and establishing proper relationships between each collection. Meanwhile, Het focused on creating ready-to-use web-app, starting with wireframes for each page followed by the actual frontend. Due to limited computational resources with Het, optimization algorithms on top of regression algorithms and Agentic AI with Sentiment Analysis were implemented by Heli. The backend of the web-app, presentations of the reviews and project report were managed by both. Fig. 1.1 shows the work division between each member involved in the project.

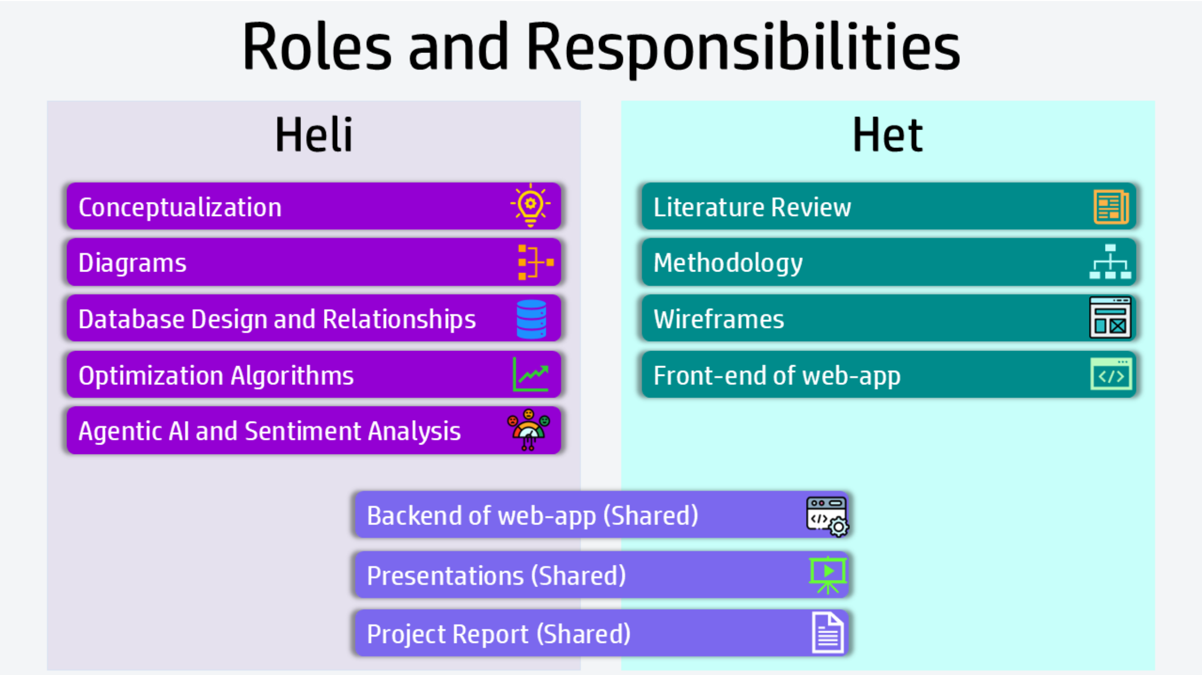

Fig. 1.1 Roles and Responsibilities of each project member

1.6.4 Group Dependencies

Our project was majorly group dependent that required continuous co-ordination between both team members. As the project was divided into distinct components based on individual strengths, completion of one module affected execution of others. For example, Het's work on literature review and methodology design laid the groundwork for the implementation phase, helping in implementation of machine learning models and optimization logic. Heli's database schema design and system architecture made sure smooth integration between backend and frontend. Further, Het's frontend development was dependent on the robust backend created by Heli, showing the interdependence. The implementation of Agentic AI and sentiment analysis by Heli was dependent on the prior completion of model training and regression-based analysis.

While the modules were developed independently, their integration required continuous collaboration.

1.7 PROJECT SCHEDULING

Our project scheduling ensured proper execution and on time completion. We organized project phases using Gantt chart, starting from understanding Agentic AI, learning fundamentals of stock markets, understanding nature inspired optimization algorithms to creating a fully functional deployed web-app. This provided clear visualization and use of resources, resulting in effective teamwork. In addition, regular team meetings with our project guide allowed open communication, cooperation, and timely solution of any potential problems Fig. 1.1 shows the Gantt chart of our project scheduling.

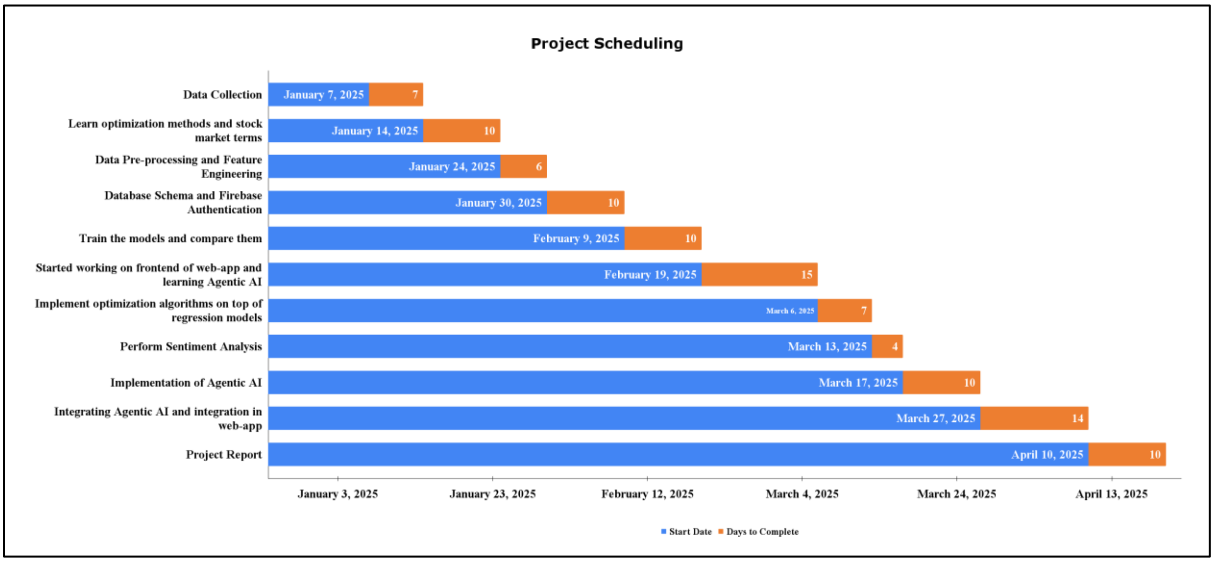

Fig. 1.2 Gantt Chart showing the scheduling of the project

CHAPTER 2

SYSTEM ANALYSIS

2.1 Study of Current System

The current studies show that traditional optimization techniques were largely dependent on statistical methods such as Markowitz's mean-variance framework which uses historical returns and covariance matrices to create portfolios. These methods were then enhanced with machine learning and deep learning techniques. Several studies shows that regression algorithms, LSTM, GRU, and even Transformer models were used to forecast stock prices and make asset allocation decisions. Also, nature-inspired metaheuristic algorithms like GA, PSO, and GWO have been used to refine the process of selecting optimal portfolio weights.

2.2 Problems and Weaknesses of Current System

Despite the advancements, there are still many challenges in current optimization systems which needs to be overcome. Though deep learning models captures complex market patterns and long-term dependencies, they suffer from issues of overfitting and lack interpretability, making it difficult to understand the underlying risk. Furthermore, the computational complexity associated with processing large, multivariate datasets and tuning hyperparameters remains a significant hurdle. Metaheuristic algorithms, although promising in finding optimal solutions, sometimes find difficulties with convergence speed and scalability when applied in real-time. This makes it challenging to quickly adjust to market fluctuations and accurately measure risks timely.

2.3 Requirements of Current System

To overcome these challenges, an enhanced portfolio optimization system must be built that uses modern techniques with traditional ways. The system should handle real-time data while effectively assessing and managing risk. Key requirements to be included are improved forecasting accuracy, specially under volatile market conditions, and the ability to process live data feeds to update predictions dynamically. Furthermore, the system must use hybrid approaches that merge ML with advanced optimization methods. In addition, less computational resources are necessary for timely adjustments providing relevant investment recommendations.

2.4 System Feasibility

The project is highly feasible due to the use of open-source tools, libraries, and frameworks essential for implementing ML algorithms, sentiment analysis, and Agentic AI-based decision-making. Additionally, the project had no financial costs, as all development and deployment were carried out using free platforms and tools.

2.4.1 Does the system contribute to the overall objectives of the organization?

Yes, the system contributes to the overall objectives. The primary goal was to provide data-driven investment insights and optimize portfolio performance for investors. It has been achieved that by using real-time stock market data, AI-based risk assessment, and portfolio optimization techniques to help users make informed investment decisions. The use of machine learning models and financial analytics ensures that investors receive dynamic recommendations according to market conditions. Further, a web-based application where user can create their portfolios, review the recommendation, and make investment strategies supports the overall objective of the organization.

2.4.2 Can the system be implemented using the current technology and within the given cost and schedule constraints?

Yes, the project was successfully implemented within the available resources and within the provided time. We utilized Firebase for database management and Google Colab for model training utilizing built-in services of GPUs. This project used data from two major Indian stock market indices Nifty50 and Sensex, making sure real-time financial insights are generated efficiently. Additionally, the use of open-source tools and pre-trained models using PyCaret has maintained high computational performance.

2.4.3 Can the system be integrated with other systems which are already in place?

Yes, it is possible to integrate seamlessly with existing platforms. As it uses Firebase in backend, it can easily connect with other cloud-based applications, including stock market APIs (Moneycontrol, Tickertape), trading platforms (Zerodha, Groww), and external financial databases. This can make it adaptable to future enhancements and expansions for large scale applications.

2.5 Activity in Proposed System

The proposed method contains various activities carried out by both the user and database manager to achieve efficient portfolio management. The user interacts with the system to receive investment recommendations based on sentiment analysis and machine learning models. Database Manager handles backend to maintain data integrity, reliability, and monitor the performance. Fig. 2.1 provides the full flow from data fetching, sentiment analysis, ML predictions, to portfolio optimization and user interactions, which helps in explaining the activities of the proposed system. Fig. 2.2 shows the sequence diagram involving activities of database manager. It begins when the user logs in, after which their request is verified by the system. If the user is not already verified, the system alerts the Database Manager. The Database Manager then approves the account manually, after which the user is granted access. In parallel, the system also checks if the user has reached the limit for the number of companies they can track. If the limit is exceeded, the Database Manager is notified, and they may choose to increase the limit based on eligibility. Also, if a user requests to update privacy settings, the request is reviewed and modified by the Database Manager through backend access.

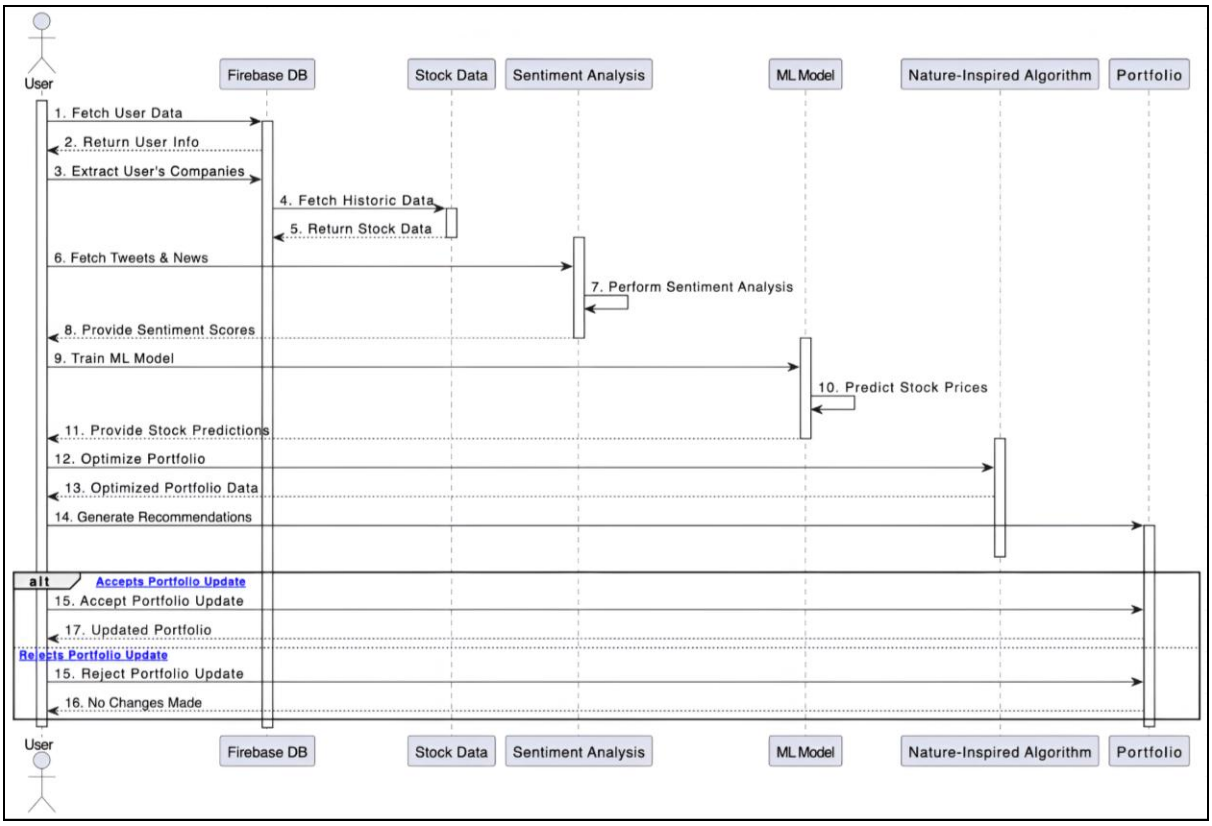

Fig. 2.1 Sequence Diagram showing the full flow of the project – User side

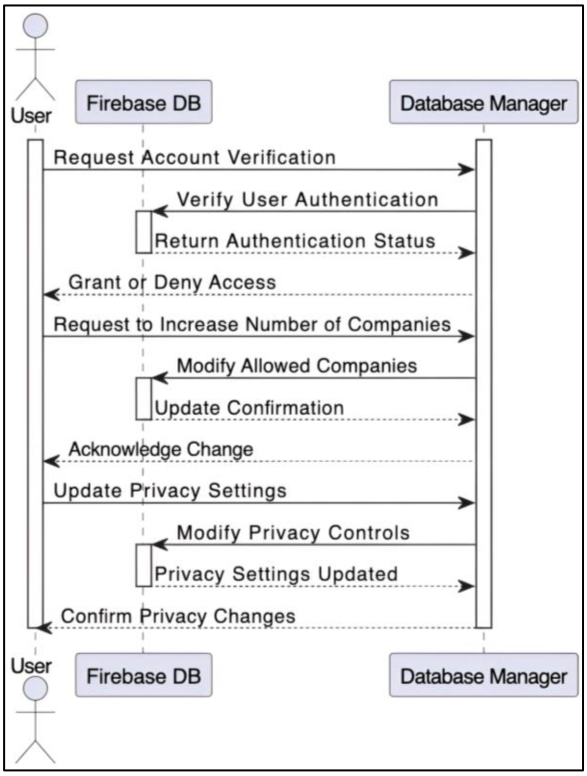

Fig. 2.2 Sequence Diagram showing database manager activities

Fig. 2.3 demonstrates the step-by-step process followed when a user interacts with the system (shown on left side). Initially, the system fetches the user's data from Firebase and identifies the companies they are interested in. Then it collects historic stock data and news articles relevant to that companies. After this, a ML model is trained, and sentiment analysis is performed on the data to generate sentiment scores. Using these inputs, model predicts future stock prices and optimizes the user's portfolio using a nature-inspired algorithm. Based on this, investment recommendations are generated. The user then has the option to accept or reject these recommendations. If user wants to accept, the portfolio is updated accordingly; otherwise, the process ends. The manager monitors the system operations and analyse market trends (shown on the right side). Based on this analysis, they review the recommendations. They can approve or modify these recommendations as needed. The database is also backed up to ensure data preservation. Following this, performance reports are generated to evaluate the system's outcomes. Finally, the manager ensures that all operations work within the regulations and risk is effectively managed.

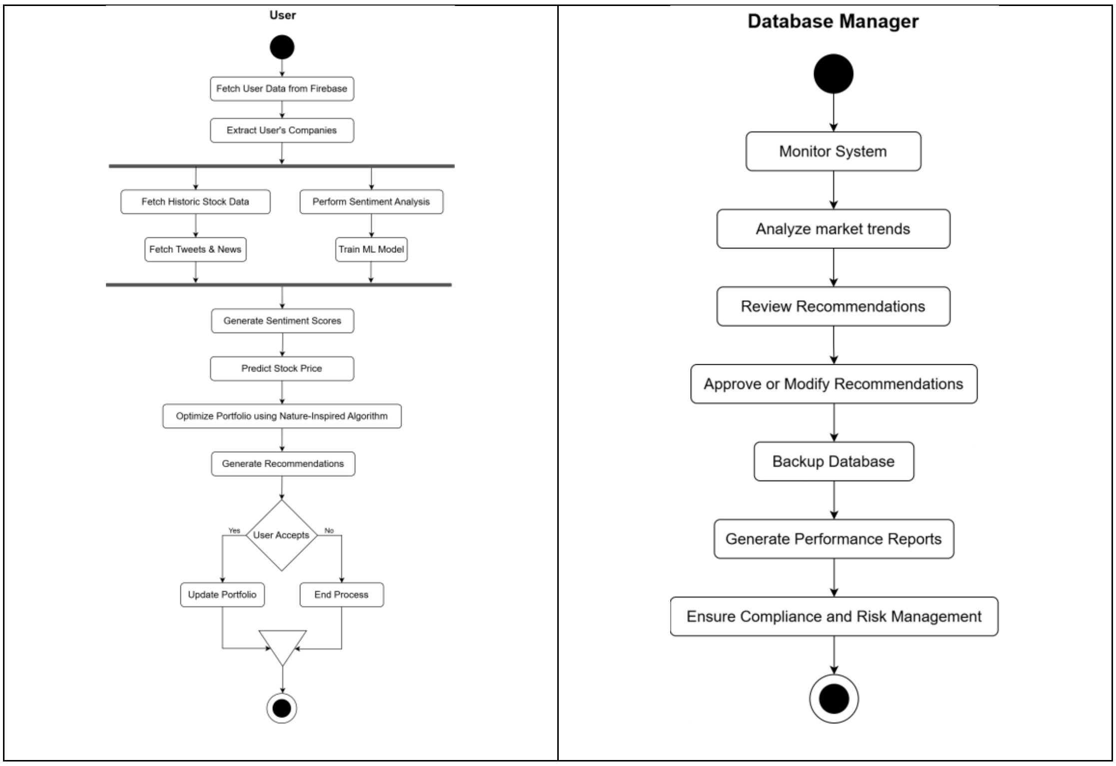

Fig. 2.3 Activity Diagrams showing interaction of user (left) and Database Manager (right) with the system

2.6 Features of Proposed System

The proposed system consists of a lot of features that assist users in making intelligent and optimized financial decisions. Fig. 2.4 shows the features of this project which are:

- It supports real-time data fetching from financial APIs and social media platforms.
- It offers automated stock price prediction using advanced machine learning models.
- It includes sentiment analysis from financial news and tweets, allowing the system to factor in market mood and public opinion before making predictions.
- It offers an optimal portfolio based on each user’s risk profile and investment goals using optimization algorithms.
- With the integration of Agentic AI, the system dynamically interacts with the user’s portfolio and provides personalized, intelligent recommendations.
- A user-friendly web-app allows users to access insights in real-time.
- Firebase ensures secure user management using its built-in authentication methods and portfolio storage.

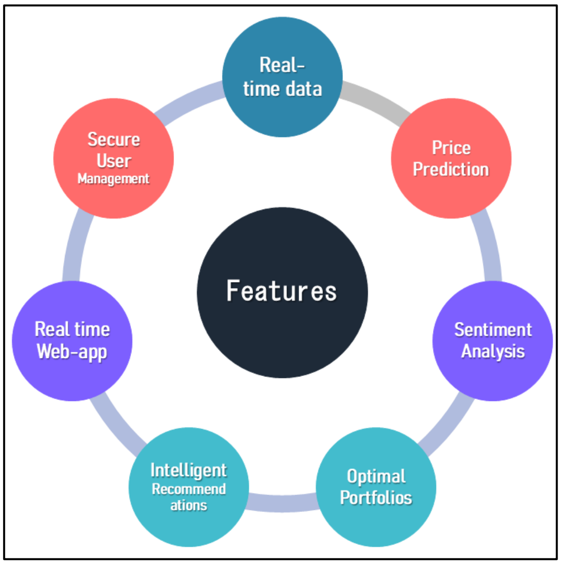

Fig. 2.4 Features of the proposed system

2.7 Main Modules of Proposed System

The proposed system is divided into various modules. User Authentication & Data Management enables secure login and portfolio storage using Firebase. Data Collection gathers both historical stock data and real-time textual data like news. To enhance decision-making, Sentiment Analysis & Feature Engineering module processes the collected text to derive sentiment scores, with the help of LangChain and other NLP tools. The Stock Price Prediction module uses machine learning libraries like PyCaret and Scikit-learn to identify the most accurate forecasting models. Optimized portfolio suggestions are generated by the Portfolio Optimization module, which uses nature-inspired algorithms and Optuna for hyperparameter tuning. Furthermore, an Agentic AI Recommendation System is integrated using LangChain and Groq to provide automated, intelligent suggestions to users. Finally, the Web-app built using Streamlit helps user interact with the whole system.

2.8 Selection of Softwares and Libraries

A variety of libraries were used during this project. Python was chosen as the core programming language due to its simplicity, vast ecosystem, and strong support for data science and machine learning. For data collection and analysis, libraries such as yFinance were used to fetch historical stock data. Firebase was used as database to store user's information and portfolio details. The applications of these libraries are mentioned as:

- PyCaret: Low-code ML library used to train list of prediction models efficiently.
- NumPy: Fundamental library for numerical computing
- Statsmodels: Used for statistical analysis to validate stock trends and relationships.
- Pandas: Used for data manipulation and to structure and preprocess data.
- Scikit-learn: Provided traditional ML algorithms and tools for preprocessing, model training, and evaluation.
- Streamlit: Framework for building and deploying real-time web app with an interactive interface for users.
- LangChain: Enables integration of Agentic AI by managing chains of LLM calls and tools to automate recommendations.
- TA-Lib: Generate technical indicators (e.g., RSI, MACD) from historical stock prices.
- BeautifulSoup: Scrape financial news headlines to extract text for sentiment analysis.
- Flask: Lightweight backend server for serving APIs behind Streamlit UI.
- Firebase: Manage real-time user data, authentication, and portfolio storage.
- Optuna: Hyperparameter optimization library used to fine-tune ML and optimization models for better accuracy.
- Groq: AI accelerator for high-speed computation and Agentic processing.
- Matplotlib: Create static charts for data analysis and performance visualization.
- Plotly: Built interactive visualizations of portfolio performance and predictions.

Fig. 2.5 List of libraries used

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM DESIGN & METHODOLOGY

Methodology is a series of step that simplifies the flow of activities of a project. It helps in visualizing the flow of process in a project. The proposed methodology consists of six steps: data collection, data preprocessing and feature engineering, data visualization, model selection, prediction and recommendations and user validation. Fig. 3.1 shows the proposed methodology of our project.

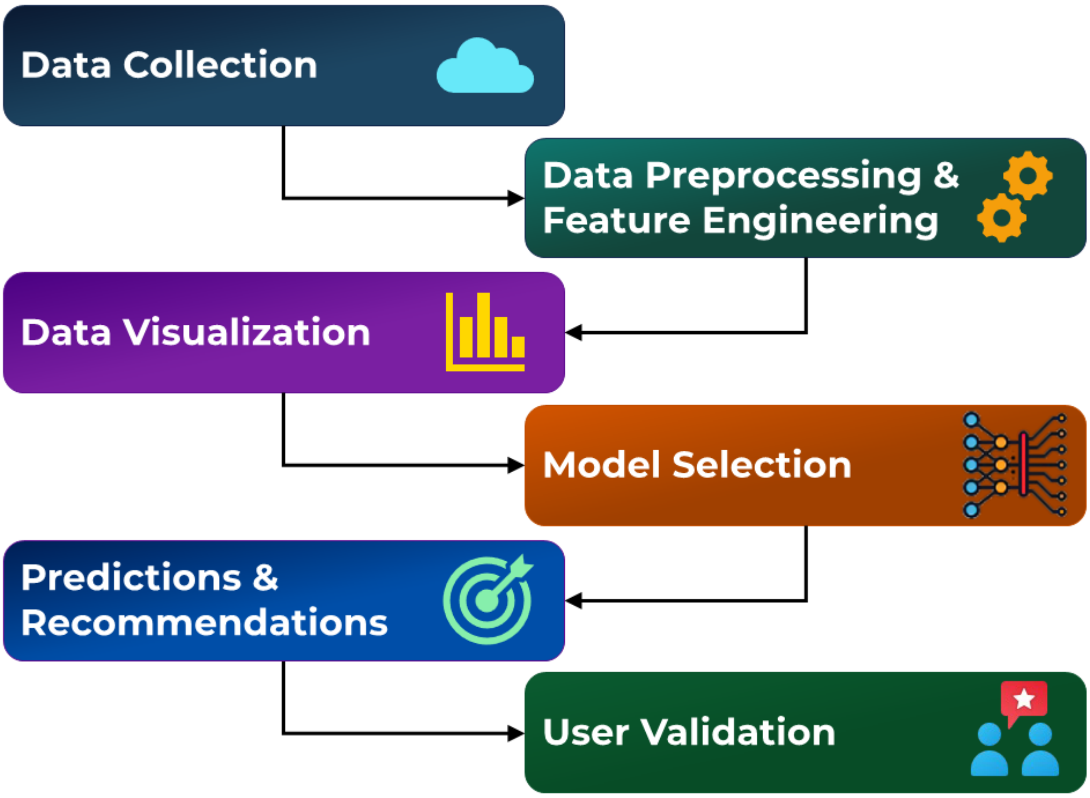

Fig. 3.1 Proposed Methodology

1. Data Collection

In this project, we utilized two types of data: Stock Market Data and News Article Data. Stock market data was fetched using yFinance, a Python library that fetches real-time and historical market data from the Yahoo Finance. We collected 15 years of historical data, starting from January 1, 2010, up to the latest available records. For news article data, we sourced real-time financial news from Moneycontrol and Livemint, ensuring accuracy and relevance in market sentiment analysis. We have used two indices for this project: Nifty50 which consists of top 50 most liquid and large cap stocks along with Sensex consisting of the largest and most traded stocks. Fig. 3.2 shows the list of companies used in this project. Table 3.1 shows the sample of the data that was fetched from yFinance which consists of date, closing price of the stock at EOD, opening price of the stock on that particular day, day's highest price achieved by stock, day's lowest price achieved by the stock and volume (total number of stocks) traded in a day.

Company Name	Ticker	Index	Company Name	Ticker	Index
NTPC Ltd	NTPC	Nifty 50	Bharat Petroleum Corporation Ltd	BPCL	Nifty 50
Trent Ltd	TRENT	Nifty 50	Tata Motors Ltd	TATAMOTORS	Nifty 50
Power Grid Corporation of India Ltd	POWERGRID	Nifty 50	Tata Consumer Products Ltd	TATACONSUM	Nifty 50
Kotak Mahindra Bank Ltd Fully Paid Ord. Shrs	KOTAKBANK	Nifty 50	HDFC Life Insurance Company Ltd	HDFCLIFE	Nifty 50
Maruti Suzuki India Ltd	MARUTI	Nifty 50	Sbi Life Insurance Company Ltd	SBILIFE	Nifty 50
Tech Mahindra Ltd	TECHM	Nifty 50	Shriram Finance Ltd	SHRIRAMFIN	Nifty 50
Coal India Ltd	COALINDIA	Nifty 50	Bajaj Finance Ltd	BAJFINANCE	Nifty 50
Larsen and Toubro Ltd	LT	Nifty 50	Bajaj Finserv Ltd	BAJAJFINSV	Nifty 50
Eicher Motors Ltd	EICHERMOT	Nifty 50	Axis Bank Ltd	AXISBANK	Nifty 50
Reliance Industries Ltd	RELIANCE	Nifty 50	Mahindra And Mahindra Ltd	M&M	Nifty 50
Adani Ports and Special Economic Zone Ld	ADANIPORTS	Nifty 50	Axis Bank Ltd	AXISBANK	Sensex
State Bank of India	SBIN	Nifty 50	Zomato Ltd	ZOMATO	Sensex
HCL Technologies Ltd	HCLTECH	Nifty 50	Tata Motors Ltd	TATAMOTORS	Sensex
Infosys Ltd	INFY	Nifty 50	Tata Consultancy Services	TCS	Sensex
UltraTech Cement Ltd	ULTRACEMCO	Nifty 50	Reliance Industries Ltd	RELIANCE	Sensex
Bharti Airtel Ltd	BHARTIARTL	Nifty 50	NTPC Ltd	NTPC	Sensex
Tata Consultancy Services	TCS	Nifty 50	Titan NV	TITAN	Sensex
Hindustan Unilever Ltd	HINDUNILVR	Nifty 50	Adani Ports and Special Economic Zone Ld	ADANIPORTS	Sensex
Hero Motocorp Ltd	HEROMOTOCO	Nifty 50	Bajaj Finserv Ltd	BAJAJFINSV	Sensex
Grasim Industries Ltd	GRASIM	Nifty 50	Tata Steel Ltd	TATASTEEL	Sensex
Apollo Hospitals Enterprise Ltd	APOLLOHOSP	Nifty 50	ITC limited	ITC	Sensex
Britannia Industries Ltd	BRITANNIA	Nifty 50	Maruti Suzuki India Ltd	MARUTI	Sensex
JSW Steel Ltd	JSWSTEEL	Nifty 50	HCL Technologies Ltd	HCLTECH	Sensex
ITC limited	ITC	Nifty 50	State Bank of India	SBIN	Sensex
Hindalco Industries Ltd	HINDALCO	Nifty 50	HDFC Bank Ltd	HDFCBANK	Sensex
Indusind Bank Ltd	INDUSINDBK	Nifty 50	Bharti Airtel Ltd	BHARTIARTL	Sensex
Cipla Ltd	CIPLA	Nifty 50	Mahindra And Mahindra Ltd	M&M	Sensex
Titan NV	TITAN	Nifty 50	Hindustan Unilever Ltd	HINDUNILVR	Sensex
HDFC Bank Ltd	HDFCBANK	Nifty 50	Larsen and Toubro Ltd	LT	Sensex
Wipro Ltd	WIPRO	Nifty 50	Asian Paints Ltd	ASIANPAINT	Sensex
ICICI Bank Ltd	ICICIBANK	Nifty 50	Nestle India Ltd	NESTLEIND	Sensex
Adani Enterprises Ltd	ADANIENT	Nifty 50	Infosys Ltd	INFY	Sensex
Tata Steel Ltd	TATASTEEL	Nifty 50	UltraTech Cement Ltd	ULTRACEMCO	Sensex
Dr Reddy's Laboratories Ltd	DRREDDY	Nifty 50	Bajaj Finance Ltd	BAJFINANCE	Sensex
Nestle India Ltd	NESTLEIND	Nifty 50	Indusind Bank Ltd	INDUSINDBK	Sensex
Asian Paints Ltd	ASIANPAINT	Nifty 50	ICICI Bank Ltd	ICICIBANK	Sensex
Bajaj Auto Limited	BAJAJ-AUTO	Nifty 50	Power Grid Corporation of India Ltd	POWERGRID	Sensex
Sun Pharmaceutical Industries Ltd	SUNPHARMA	Nifty 50	Kotak Mahindra Bank Ltd Fully Paid Ord. Shrs	KOTAKBANK	Sensex
Bentley Capital Ltd	BEL	Nifty 50	Tech Mahindra Ltd	TECHM	Sensex
Oil and Natural Gas Corporation Ltd	ONGC	Nifty 50	Sun Pharmaceutical Industries Ltd	SUNPHARMA	Sensex

Fig. 3.2 List of companies used

Table 3.1 Sample data from yFinance

Date	Close	High	Low	Open	Volume
2010-01-04	109.1127	112.4306	108.5715	111.7246	6,309,685

2010-01-05	107.9126	110.5245	107.3714	110.2186	8,199,537
2010-01-06	107.395	110.5481	107.1361	108.3833	8,325,996
2010-01-07	107.4184	108.1479	106.0066	108.1479	6,505,911
2010-01-08	108.7597	109.9363	107.8655	108.1479	7,441,676

2. Data Preprocessing and Feature Engineering

The data fetched from yFinance was not sufficient to train the model as it contained very simple features (as mentioned in Table 3.1) which are not sufficient to train the models. To enhance the dataset, we performed feature engineering, generating more complex technical indicators to help models gain deeper understanding of market trends. These features were created using a python library TA-Lib (Technical Analysis Library), a python library used for technical analysis of financial markets. The final dataset consisted of 25 technical features covering every aspect of stock market. Fig. 3.3 shows the steps of how feature engineering was implemented on the original dataset. The steps include:

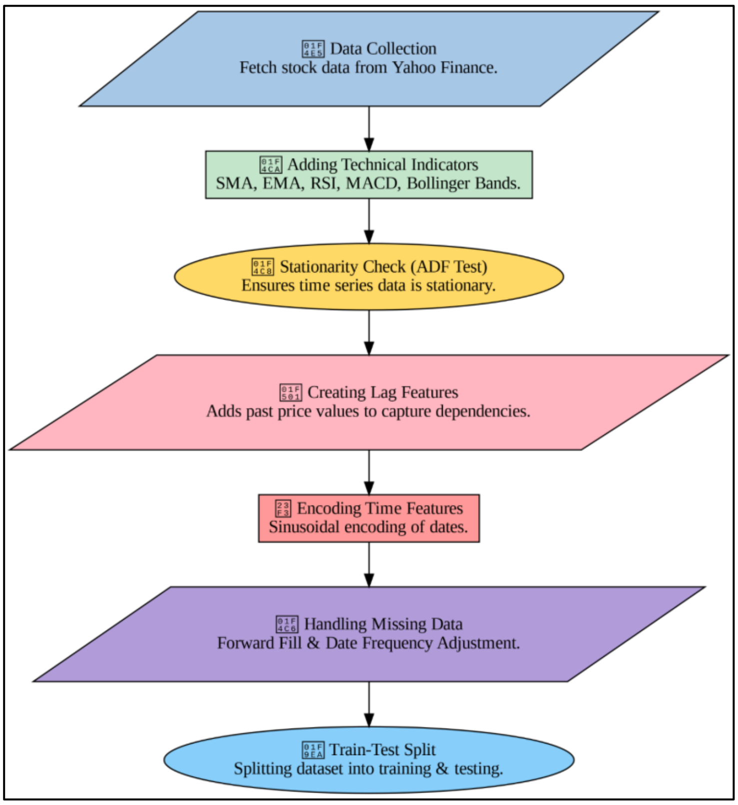

Fig. 3.3 Feature Engineering Process for Stock Market Data

- Data Collection: Fetches historical stock market data from Yahoo Finance using the yFinance library.
- Add Technical Indicators: Calculate financial indicators such as SMA, EMA, RSI, MACD and Bollinger Bands to capture trends and momentum.
- Stationary Check (ADF Test): This step uses Augmented Dickey-Fuller (ADF) test to check whether the stock price data follows a pattern over time. (For ex. If a stock price consistently increases over time, the model may struggle to identify patterns). Machine learning models work better with stationary (data that fluctuates but stays around a constant mean and do not follow a continuous trend) data because they assume that patterns do not change over time. A stock that always increases or decreases over time is considered non-stationary, and that kind of data is harder for machine learning models to handle directly.
- Encoding Time Features: Time-based features like days and months repeat in a cycle. If we just use numbers (1 to 31 for days or 1 to 12 for months), the model would not understand that Day 1 and Day 31 are actually close together. To solve this, we convert time into sine and cosine values, which naturally repeat in a smooth wave-like pattern.
- Handling Missing Data: This step ensures that any missing data points in the stock market dataset are filled appropriately to maintain continuity.
- Train-Test Split: Splits the final dataset into training and testing sets for model evaluation.

A sample of the newly generated features is shown in Table 3.2, which includes both newly created and existing features. These features improve the training process of ML algorithms, allowing them to understand and capture complex patterns in stock market data. Also, Fig. 3.4 provides a detailed description of each feature with its importance helping understanding each feature and its role in different cases.

Table 3.2 Dataset after performing Feature Engineering

Date	Close	High	Low	Open	Volume	SMA_10	SMA_30	EMA_10	EMA_30	Return	Volatility	RSI	MACD	BB_ High	BB_ Low	Close_lag_1	Close_lag_2	Close_lag_3	Close_lag_4	Close_lag_5	Day_sin	Day_cos	Month_sin	Month_cos
16-02-2010	95.84	96.00	94.84	95.39	12,19,861	96.492	103.31	97.058	101.64	0.0065	0.0147	30.56	-3.329	109.974	91.122	95.221	96.025	94.725	96.379	95.623	0.20	0.98	0.50	0.87
17-02-2010	95.34	96.71	95.03	96.62	23,31,838	96.294	102.84	96.746	101.23	-0.0052	0.0127	29.49	-3.265	108.638	91.067	95.836	95.221	96.025	94.725	96.379	0.20	0.98	0.50	0.87
18-02-2010	96.90	97.28	94.96	95.34	26,94,157	96.069	102.46	96.773	100.95	0.0164	0.0123	36.98	-3.053	107.434	91.203	95.339	95.836	95.221	96.025	94.725	0.20	0.98	0.50	0.87
19-02-2010	95.34	96.78	93.95	95.43	53,07,267	95.791	102.04	96.513	100.59	-0.0161	0.0128	33.18	-2.977	105.802	91.532	96.899	95.339	95.836	95.221	96.025	0.20	0.98	0.50	0.87
22-02-2010	95.55	96.80	95.03	96.80	50,54,958	95.694	101.63	96.338	100.27	0.0022	0.0120	34.17	-2.866	104.665	91.665	95.339	96.899	95.339	95.836	95.221	0.20	0.98	0.50	0.87
23-02-2010	95.27	95.72	95.08	95.43	42,51,614	95.658	101.17	96.144	99.944	-0.0030	0.0117	33.46	-2.769	103.470	91.883	95.552	95.339	96.899	95.339	95.836	0.20	0.98	0.50	0.87
24-02-2010	96.57	96.83	93.07	93.07	64,11,043	95.677	100.72	96.221	99.726	0.0136	0.0122	39.68	-2.558	101.690	92.748	95.269	95.552	95.339	96.899	95.339	0.20	0.98	0.50	0.87
25-02-2010	94.54	96.80	94.21	96.80	81,63,273	95.658	100.26	95.914	99.391	-0.0210	0.0129	34.28	-2.526	100.793	92.869	96.568	95.269	95.552	95.339	96.899	0.20	0.98	0.50	0.87
26-02-2010	95.93	96.62	94.68	94.82	1,30,10,078	95.649	99.842	95.917	99.168	0.0147	0.0130	40.28	-2.361	100.021	93.122	94.536	96.568	95.269	95.552	95.339	0.20	0.98	0.50	0.87
01-03-2010	95.79	96.85	95.55	96.10	26,42,516	95.706	99.397	95.894	98.950	-0.0015	0.0127	39.88	-2.215	99.010	93.587	95.930	94.536	96.568	95.269	95.552	0.20	0.98	0.50	0.87
02-03-2010	97.94	98.34	95.79	95.81	80,25,309	95.916	99.004	96.265	98.884	0.0224	0.0144	48.25	-1.905	98.520	93.888	95.788	95.930	94.536	96.568	95.269	0.20	0.98	0.50	0.87
03-03-2010	97.73	98.43	97.42	98.27	56,07,044	96.154	98.620	96.531	98.810	-0.0022	0.0143	47.54	-1.657	98.585	93.863	97.938	95.788	95.930	94.536	96.568	0.20	0.98	0.50	0.87
04-03-2010	97.30	98.36	97.04	98.22	41,50,608	96.195	98.277	96.671	98.712	-0.0044	0.0135	46.09	-1.478	98.149	94.115	97.725	97.938	95.788	95.930	94.536	0.20	0.98	0.50	0.87
08-03-2010	96.92	98.22	96.24	96.87	38,69,247	96.353	97.895	96.716	98.597	-0.0039	0.0124	44.78	-1.351	97.911	94.232	97.300	97.725	97.938	95.788	95.930	0.20	0.98	0.50	0.87

Features	Full Form	Description	Importance
Date	Trading Date	Represents the specific trading day.	Helps in tracking stock price trends over time.
Close	Closing Price	Final price at which the stock was traded.	Used for analyzing price trends and returns.
High	Highest Price	Maximum price of the stock on a trading day.	Indicates the highest value reached, useful for volatility analysis.
Low	Lowest Price	Minimum price of the stock on a trading day.	Helps in identifying the price range and market fluctuations.
Open	Opening Price	First price at which the stock was traded.	Important for gap analysis between opening and closing prices.
Volume	Trading Volume	Number of shares traded in a day.	Indicates market activity and liquidity of the stock.
SMA_10	10-day Simple Moving Average	Average of closing prices over the last 10 days.	Smooths out price fluctuations to identify short-term trends.
SMA_30	30-day Simple Moving Average	Average of closing prices over the last 30 days.	Helps in understanding long-term trends.
EMA_10	10-day Exponential Moving Average	Weighted average of the last 10 closing prices.	Reacts more quickly to price changes than SMA, useful for trading signals.
EMA_30	30-day Exponential Moving Average	Weighted average of the last 30 closing prices.	Identifies longer-term trends with less lag than SMA.
Return	Daily Return	Percentage change in closing price from the previous day.	Measures stock performance and helps in risk assessment.
Volatility	Price Volatility	Standard deviation of stock returns over a period.	Important for risk analysis and option pricing.
RSI	Relative Strength Index	Momentum indicator that measures overbought or oversold conditions.	Helps in identifying potential reversal points.
MACD	Moving Average Convergence Divergence	Trend-following momentum indicator.	Used to identify bullish and bearish trends.
BB_High	Upper Bollinger Band	Upper limit of the Bollinger Bands.	Shows potential resistance levels and overbought conditions.
BB_Low	Lower Bollinger Band	Lower limit of the Bollinger Bands.	Shows potential support levels and overbought conditions.
Close_lag_1	1-Day Lagged Close Price	Closing price of the stock 1 day ago.	Useful for time-series modeling and predicting future prices.
Close_lag_2	2-Day Lagged Close Price	Closing price of the stock 2 days ago.	Provides historical context for stock price movements.
Close_lag_3	3-Day Lagged Close Price	Closing price of the stock 3 days ago.	Helps in short-term trend analysis.
Close_lag_4	4-Day Lagged Close Price	Closing price of the stock 4 days ago.	Useful in technical analysis and feature engineering.
Close_lag_5	5-Day Lagged Close Price	Closing price of the stock 5 days ago.	Helps in modeling and identifying price patterns.
Day_sin	Sine value of the Day	Encodes the cyclical pattern of days using sine transformation.	Helps ML models understand seasonal patterns in stock prices.
Day_cos	Cosine value of the Day	Encodes the cyclical pattern of days using cosine transformation.	Complements Day_sin to represent cyclic nature effectively.
Month_sin	Sine value of the Month	Encodes the cyclical pattern of months using sine transformation.	Useful for detecting seasonal effects on stock prices.
Month_cos	Cosine value of the Month	Encodes the cyclical pattern of months using cosine transformation.	Helps in preserving seasonality in stock market data.

Fig. 3.4 Feature Descriptions

3. Data Visualization

Data visualization helps in understanding the patterns in the data by converting them into various types of graphs and charts. It makes complex data easier to understand and helps in spotting trends. Fig. 3.5 shows the RSI and MACD indicators from 2010 to 2025. RSI shows if a stock is overbought (RSI >= 70) or oversold (RSI <= 30). In the graph the stock was majorly between 30–70 while sometimes was overbought (Year 2023) or oversold (Year 2025). For MACD, if the line goes above zero, it indicates bullish signal. On the other hand, when the line goes below zero, it suggests bearish trend. In the chart, we can observe a significant increase in volatility around between 2023–2025, with sharp spikes followed by steep drops helping to understand when the stock might be gaining or losing strength. Only handful of stocks are taken for visualization for better understanding.

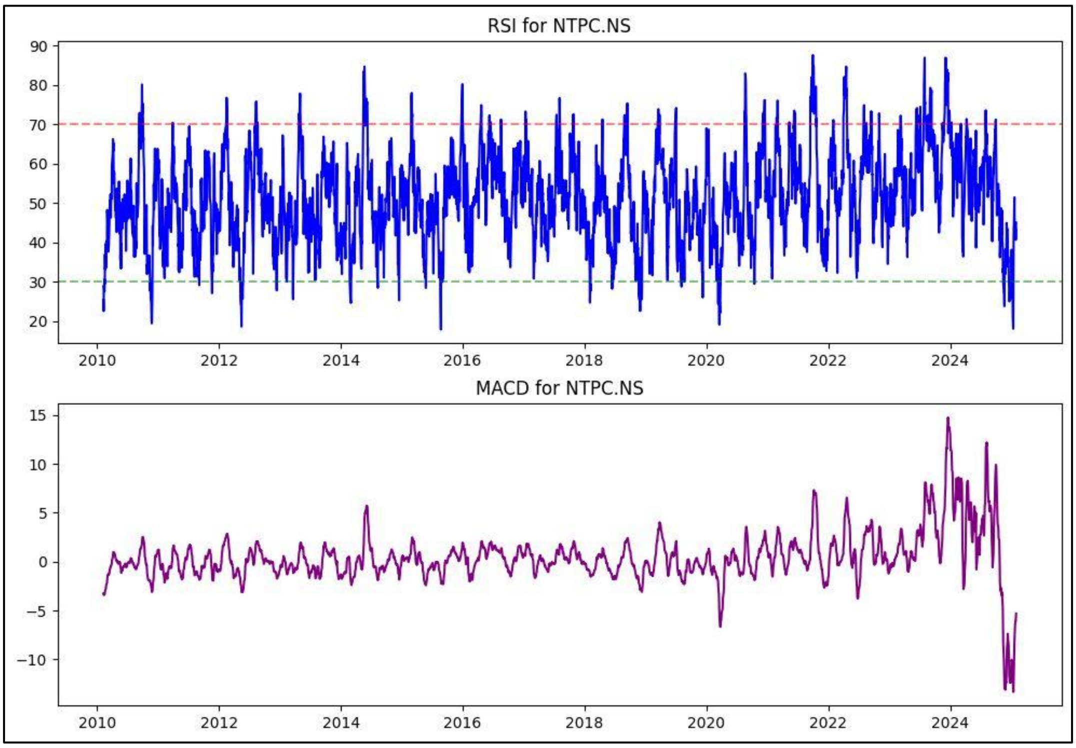

Fig. 3.5 Visualization of RSI and MACD indicators on NTPC stock

Fig. 3.6 shows the correlation heatmap indicating how stocks are correlated with one another. As the stocks are from one index (Nifty50), if the index is down, the stocks too go down which indicates strong correlation between the stocks. Fig. 3.7 shows the comparison of stocks based on volatility. It shows Zomato is the most volatile followed by ADANI ENTERPRISE while HDFC Bank being least volatile.

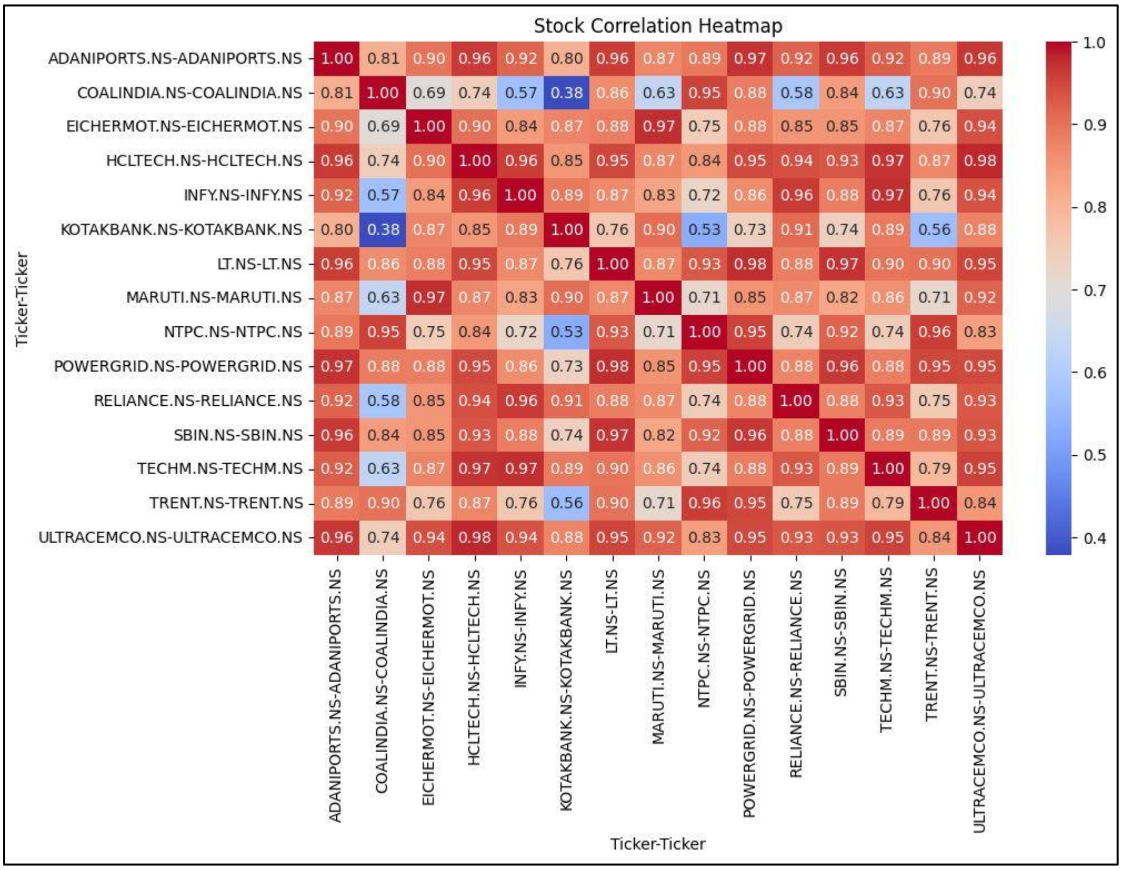

Fig. 3.6 Stocks Correlation Heatmap

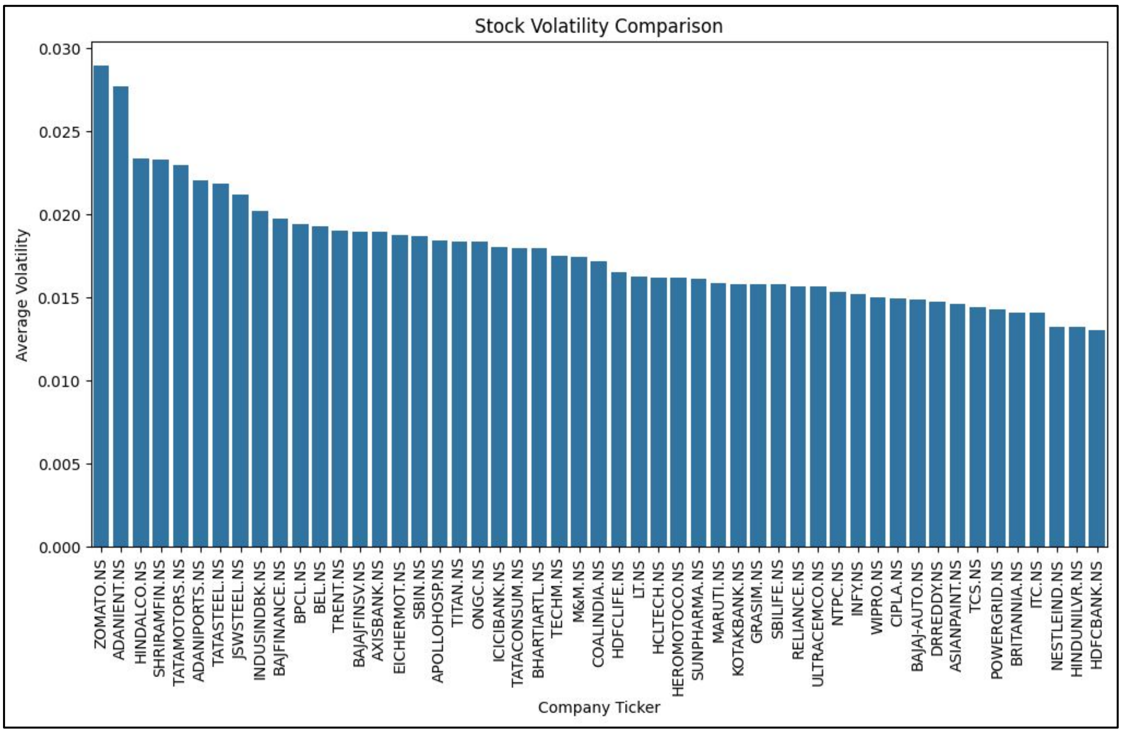

Fig. 3.7 Volatility comparison between sample of stocks

Fig. 3.8 shows the returns provide by sample of stocks over the period of time (2010–2025). From the figure we can observe that EICHER MOTORS had the highest cumulative returns roughly 5x to that of initial price. Fig. 3.9 shows a box plot showing the closing price range where box represents the spread of prices, the middle line indicates the median, and the whiskers extend to the lower and upper values. Outliers (dots) represent unusual price movements that deviate significantly from the normal range. These can occur due to sudden market events (budget announcement), etc.

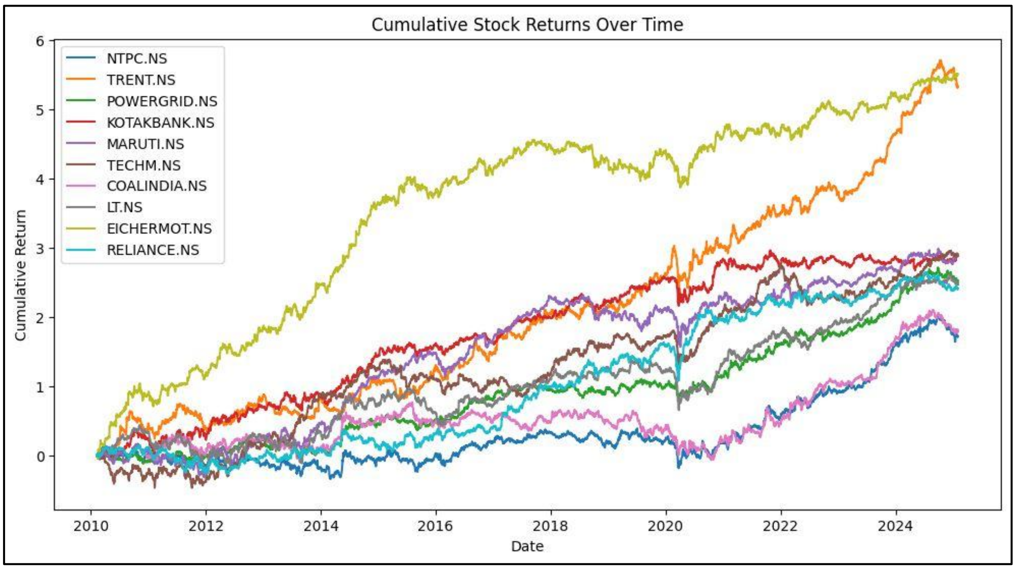

Fig. 3.8 Graph showing returns from a list of companies

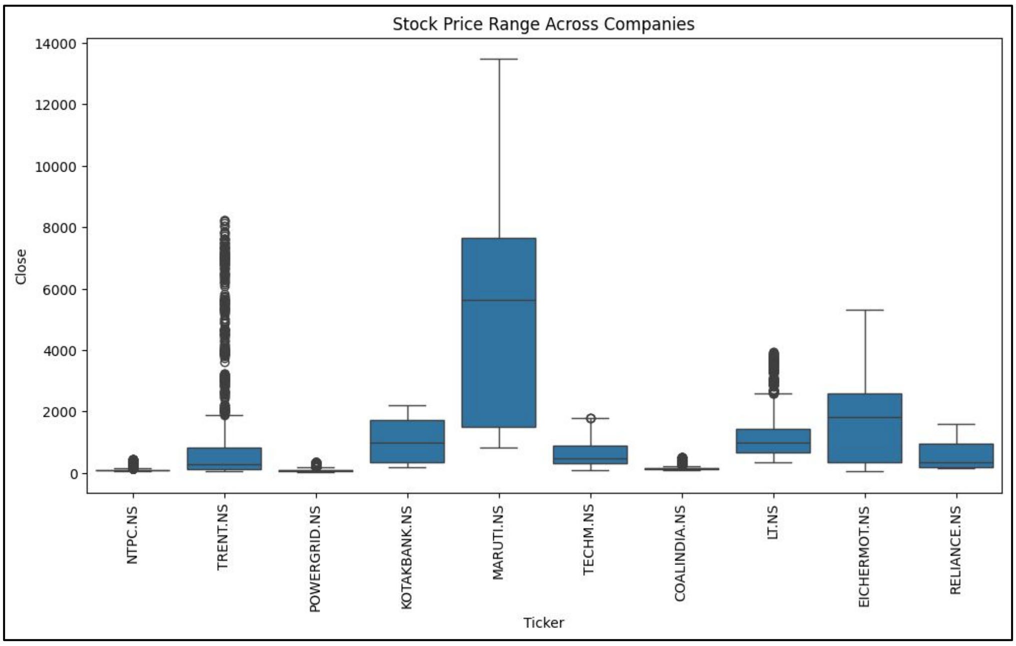

Fig. 3.9 Box plot showing closing prices from sample of stocks

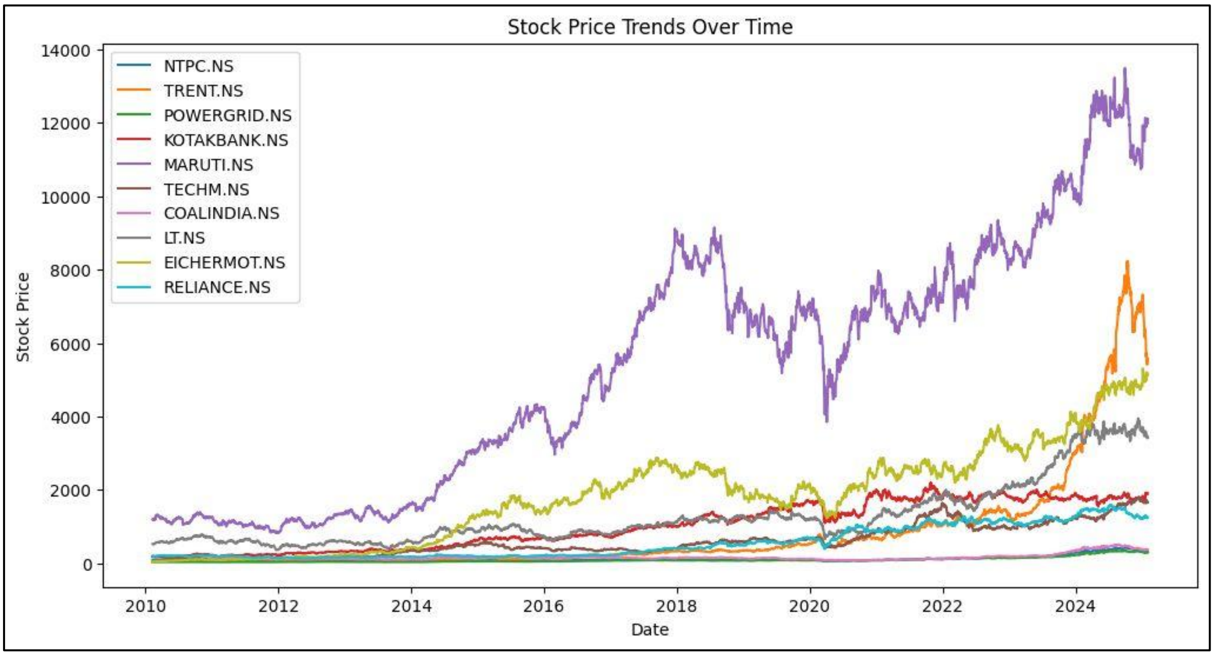

Fig. 3.10 Sample graph showing price of stocks from a list of companies

Fig. 3.10 shows the prices of stock over the course of 15 years. Sample of stock prices of 10 companies are shown in the above figure. From the figure MARUTI.NS had stock price of Rs. 1,500 in 2010 which has increased to Rs. 11,000 in 2025. TRENT.NS had a stable growth but showed an exponential growth in price around 2024.

4. **Model Selection**

Selection of appropriate models was an important step in this project because we needed an algorithm that gave minimum error in stock price prediction. Various machine learning algorithms were evaluated. After finding the best ML algorithm, it was integrated with nature-inspired optimization algorithms. Agentic AI was integrated to provide automated suggestions based on user's portfolio.

- **Regression Algorithms**

  A total of 25 machine learning algorithms were used in initial step of stock price prediction. These algorithms included Linear Regression, Lasso Regression (L1 Regularization), Ridge Regression (L2 Regularization), etc. More complex and ensemble algorithms like Gradient Boosting, LightGBM, etc. were also used in stock price prediction. These algorithms were trained simultaneously and in computationally efficient way using Python’s PyCaret library. Table 3.3 shows the list of ML algorithms used in training.

Table 3.3 List of Machine Learning Algorithms used

Linear Regression (LR)	Kernel Ridge Regression (KR)
Huber Regression (Huber)	Support Vector Regression (SVR)
Ridge Regression (Ridge)	K–Nearest Neighbors (KNN)
Elastic Net Regression (EN)	Decision Trees (DT)
Least Angle Regression (LAR)	Random Forest (RF)
Lasso Least Angle Regression (LLAR)	Extra Trees Regression (ET)
Orthogonal Matching Pursuit (OPM)	AdaBoost Regression (Ada)
Bayesian Ridge Regression (BR)	Gradient Boosting Regression (GBR)
Automatic Relevance Determination (ARD)	Multi–Layer Perceptron (MLP)
Passive Aggressive Regression (PAR)	Extreme Gradient Boosting (XGBoost)
Least Absolute Shrinkage and Selection Operator (Lasso)	Light Gradient Boosting Machine (LightGBM)
Theil–Sen Estimator (TR)	Categorical Boosting (CatBoost)
Random Sample Consensus (RANSAC)	

- **Optimization Algorithms**

  Out of all the available nature-inspired optimization algorithms, we found three algorithms that were relevant to this project and would provide optimal results. These algorithms are: Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO) and Bat Optimization. These algorithms were individually implemented as well as hybrid and ensemble approaches were also implemented to find best possible algorithm combination for portfolio optimization. Fig. 3.11 shows the list of optimization algorithms. A brief explanation of these algorithms is mentioned in later section (explained in Section 4.2).

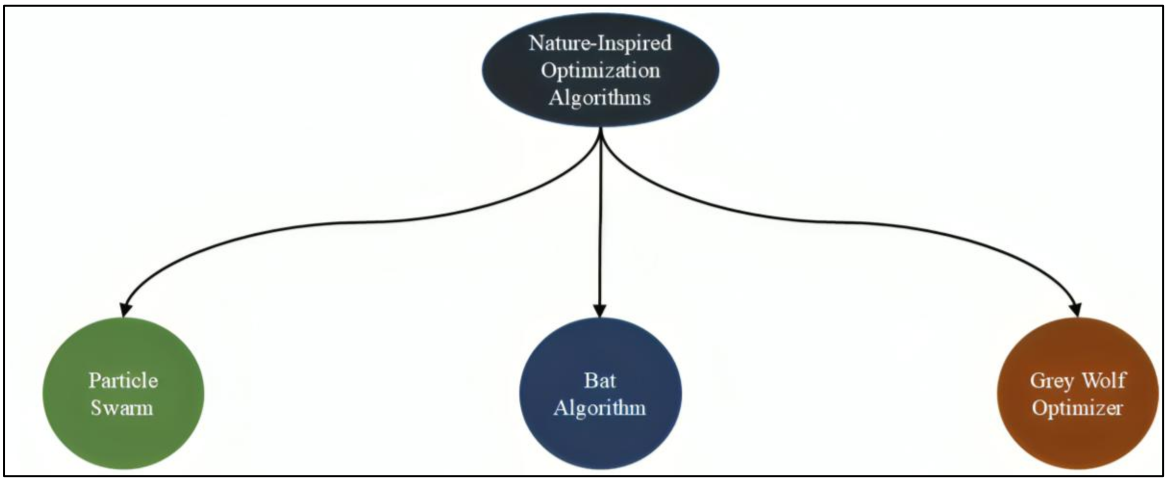

Fig. 3.11 Nature-Inspired Optimization Algorithms used in Portfolio Optimization

- **Agentic AI**

  Agentic AI is a new field in AI enabling intelligent, adaptive decision-making. In this project. In this project, Agentic AI is integrated to monitor user portfolios, track real-time market data, and autonomously adjust strategies based on user risk profiles and market dynamics. Unlike static models, it not only reacts to changes but also initiates actions such as rebalancing stocks or optimizing allocations without external commands. Agents created for portfolio optimization are:

  - Market Research Agent: For continuously gathering and analysing current market conditions.
  - Portfolio Optimization Agent: It focuses on analysing changes made to the user's portfolio.

5. **Predictions and Recommendations**

   First step was stock price prediction. We implemented 25 ML models for stock price prediction. The best performing model was used stock price prediction. The sentiment analysis was also simultaneously implemented to find the sentiment scores based on the news. After predicting stock prices, optimization algorithms were used to find the optimal weights of the portfolios. Combinations of optimization as well as ensemble approaches were also tried to find the optimal portfolios with minimum risk. Using best optimization algorithm, along with agentic AI, recommendations and portfolio rebalancing was implemented which user can accept or reject based on their preferences. The results are discussed further in Section 4.3 and 4.4.

6. **User Validation**

   In this step, personalized investment recommendations based on selected companies, risk profile, and current market conditions are provided to the users. These recommendations are generated by combining stock price predictions, sentiment analysis, and optimization results. The Agentic AI evaluates recent market trends and validates each suggestion before presenting it to the user. The final output includes portfolio rebalancing, confidence scores and sentiment score with detailed justification for rebalancing allowing the user to make informed decisions with less effort.

3.2 DATABASE DESIGN AND ER DIAGRAM

In our project, the database is used to handle user details and portfolios. To ensure real-time access, we have used Firebase as the primary database service for our project.

Firebase offers a cloud-based, NoSQL database services which enables faster access and retrieval. The database schema includes collections for Users, Stocks, Holdings, and Transactions, allowing dynamic tracking of investment activities. Firebase Authentication ensures secure access control to the database. Table 3.4 to 3.7 describes the schemas for each collection in database. Fig. 3.12 shows the class diagram where an investor manages their portfolio by adding or removing stocks based on their risk tolerance and investment goals. The portfolio contains multiple stocks, each with attributes like price, volatility, and expected return, and can be optimized using different algorithms to find the best asset allocation. The system includes machine learning models which analyse historical stock data to predict future prices. Optimization techniques like help improve investment decisions by balancing risk and return. Fig. 3.13 shows the ER diagram of our database describing the relationship between each collection with other collections. It shows how well the data is database is organized and relationships are defined.

Table 3.4 Database Schema for User Table

Field Name	**Data type**	**Field Length**	**Constraint**
User_ID	Varchar2	10	Primary Key, Auto Increment
Email_ID	Varchar2	30	Not Null
Phone_No	Varchar2	10, 0	Not Null
Password	Varchar2	30	Not Null
Name	Varchar2	30	Not Null
Address	Varchar2	100	Not Null
Blocked	Boolean	–	Not Null
First_Login_Date	Date	–	Not Null
Last_Login_Date	Date	–	Not Null
Modified_On	Date	–	Not Null
Modified_By	Varchar2	10	Foreign Key

Table 3.5 Database Schema for Stocks Table

Field Name	**Data type**	**Field Length**	**Constraint**
Stock_ID	Varchar	10	Primary Key, Auto Increment
Ticker_Symbol	Varchar2	30	Not null
Company_name	Varchar2	30	Not null
Exchange	Varchar2	30	Not null
Is_Active	Boolean		Not null

Listed_on	Varchar2	30	Not null
Listed_by	Number	10,0	Foreign key
Modified_on	Date		Not null
Modified_by	Varchar	10	Foreign key

Table 3.6 Database Schema for Holding Table

Field Name	Data type	Field Length	Constraint
Holding_ID	Varchar2	10	Primary Key, Auto Increment
Stock_ID	Number	10,0	Foreign key
User_ID	Number	10,0	Foreign key
Quantity	Number	10,0	Not null
Purchase_date	Date		Not null
Purchased_price	Number	10,3	Not null
Notes	Varchar2	100	Not null
Modified_by	Varchar2	10	Foreign key
Modified_on	Date		Not null

Table 3.7 Database Schema for Manager Table

Field Name	Data type	Field Length	Constraint
Emp_ID	Varchar2	10	Primary Key
Email_ID	Varchar2	30	Not null
Name	Varchar2	30	Not null
Password	Varchar2	30	Not null
Holding_ID	Varchar2	10	Foreign key
Stock_ID	Varchar2	10	Foreign key
User_ID	Varchar2	10	Foreign key

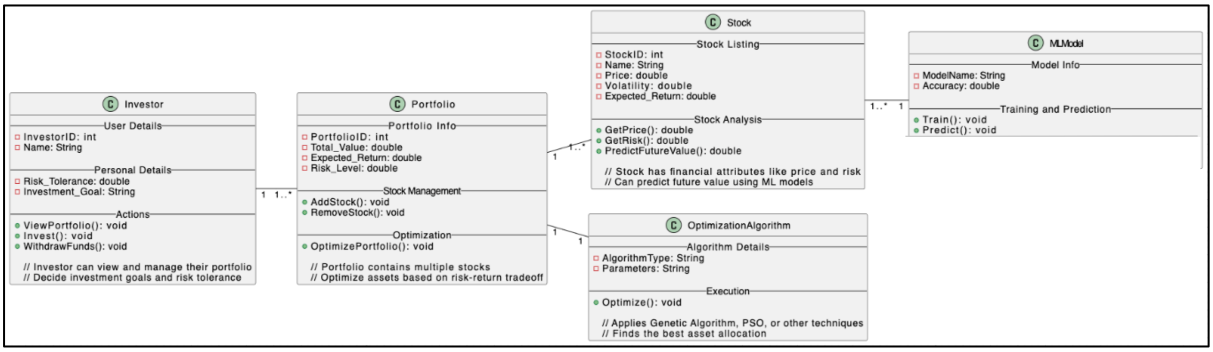

Fig. 3.12 Class Diagram

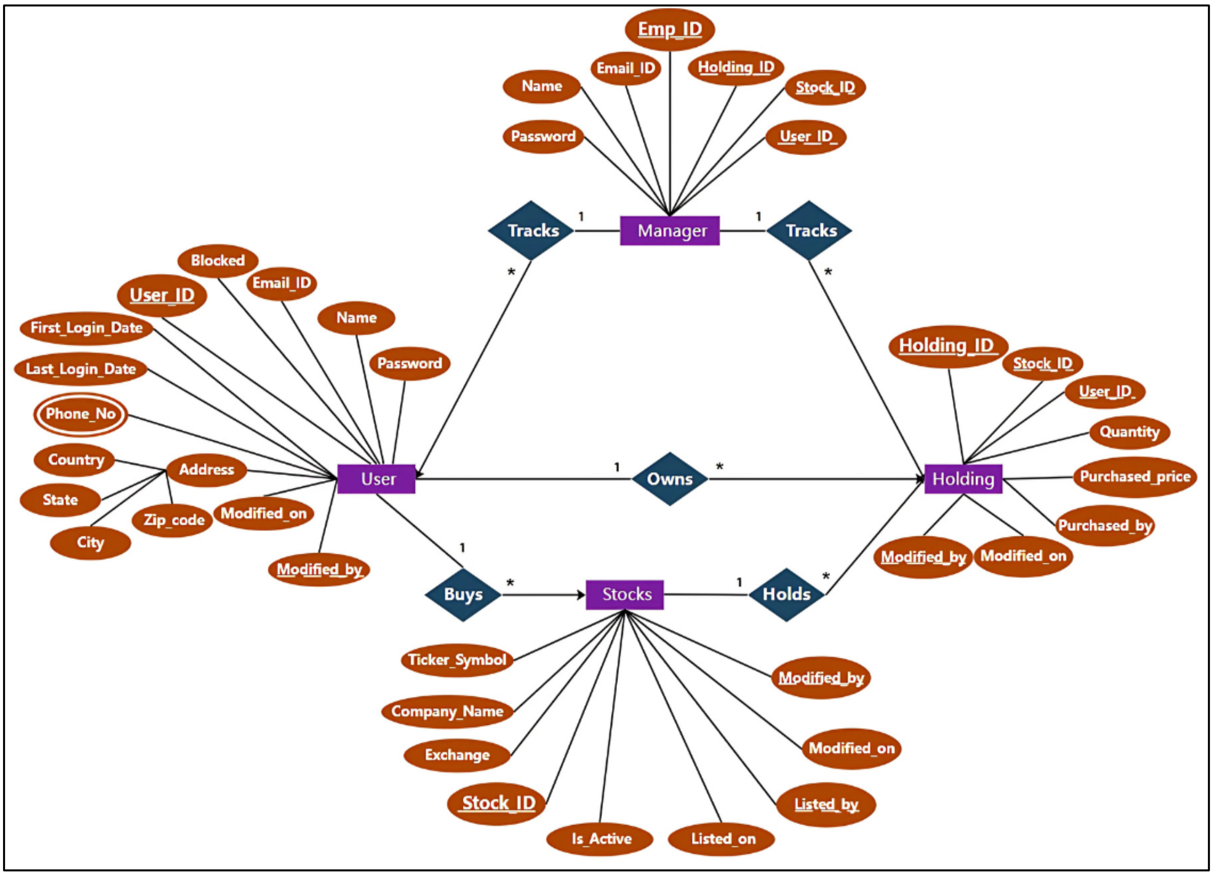

Fig. 3.13 ER Diagram for Portfolio Optimization

3.3 OUTPUT INTERFACE DESIGN

The output interfaces were designed to be user-friendly. Our primary goal was to display recommendation results in a way that is easy to understand. We used simple yet responsive UI elements using Streamlit, ensuring layouts remain consistent. Further sections describe how user interacts with the system (Section 3.3.1), wireframes of the web-app (Section 3.3.2) and authentication and role-based access (Section 3.3.3).

3.3.1 State Transition Diagram

The state transition diagram shows how a user interacts with the system. Fig. 3.14 shows the state transition diagram of the project. It starts by fetching user data from Firebase, followed by extracting the list of companies in the user's portfolio. Parallel processes are executed: one fetches historical stock data for those companies and performs preprocessing along with feature engineering to train an ML model for price prediction and the other system scrapes news articles relevant to the portfolio, performs sentiment analysis, and calculate sentiment scores. These predictions and scores are passed to optimization algorithms, which generate personalized investment recommendations. The user is then presented with these insights, and based on user's review, they can either accept and update the portfolio or reject the recommendations.

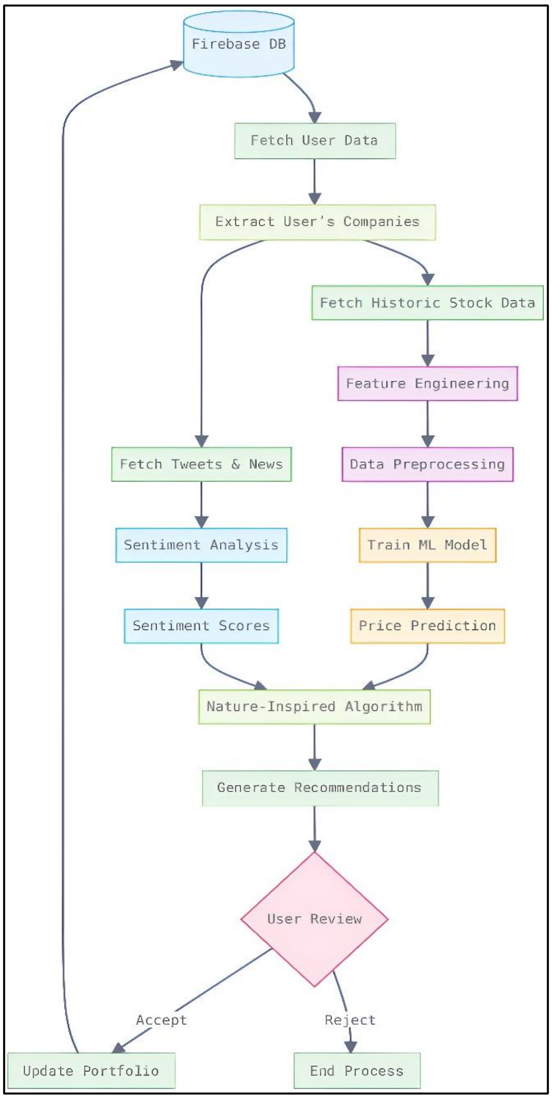

Fig. 3.14 State Transition Diagram

3.3.2 Sample of Forms and Interface

Each page was built with validation rules to prevent errors. Wireframes of these pages are included below. Fig. 3.15 – 3.21 shows the wireframes of each web pages. Fig. 3.15 shows sign in page, Fig. 3.16 shows the registration page, Fig. 3.17 shows profile page, Fig. 3.18 shows the user's home page, Fig. 3.19 shows buying page, Fig. 3.20 shows the edit stocks page where user can edit the quantity of the stocks, Fig. 3.21 shows the optimize page where optimized portfolio with detailed report is shown.

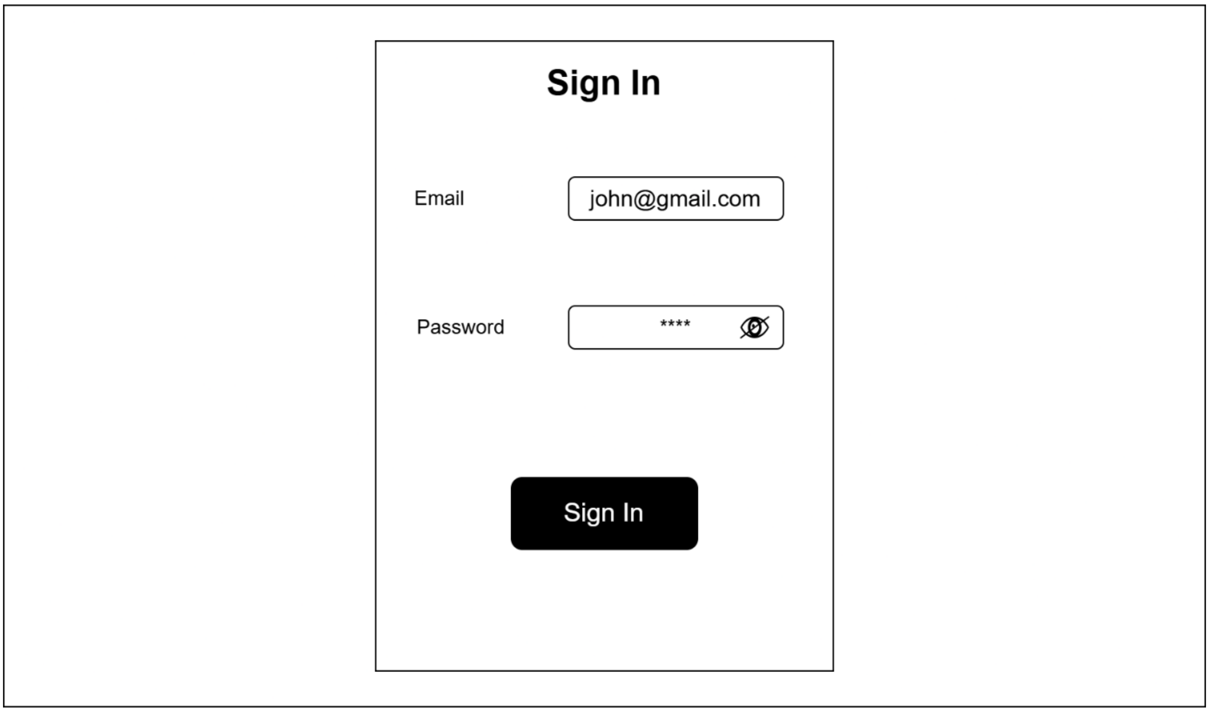

Fig. 3.15 Wireframe of Sign In Page

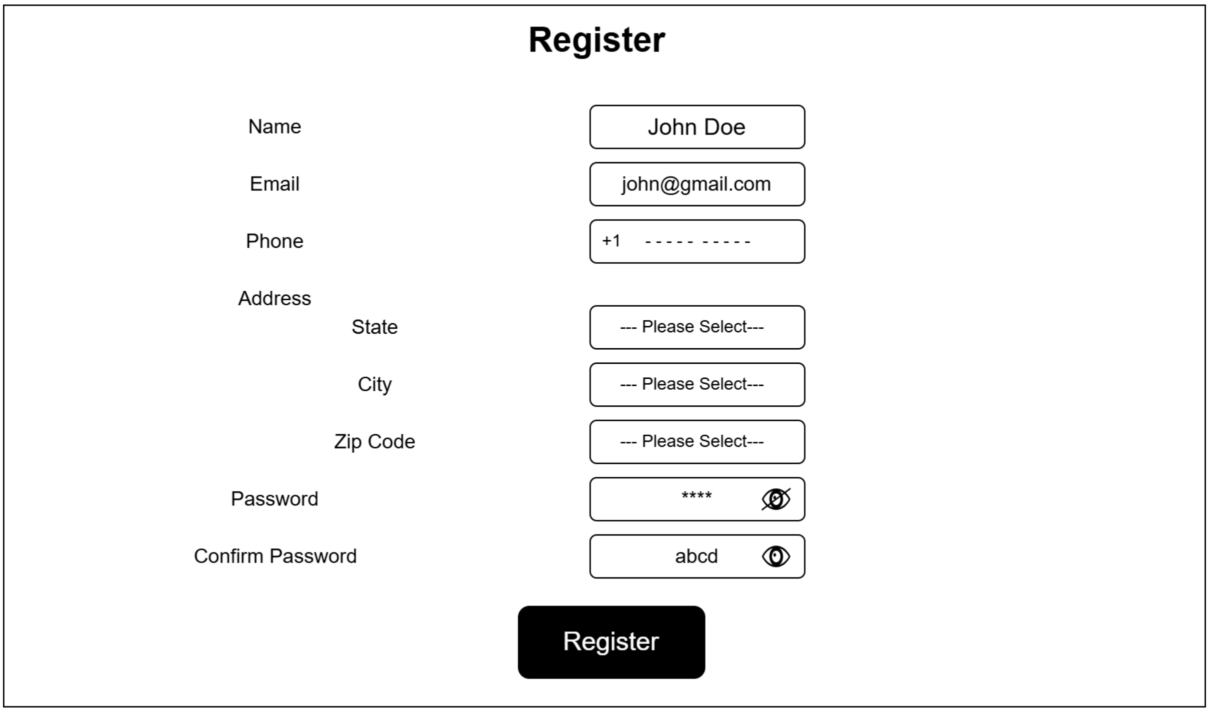

Fig. 3.16 Wireframe of Register Page

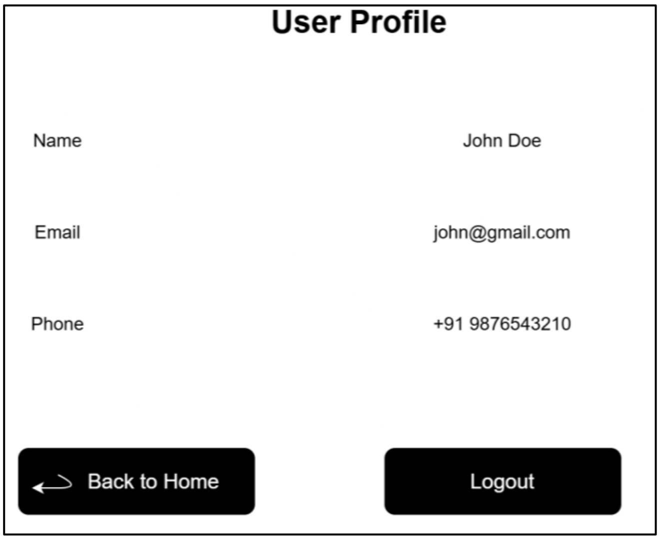

Fig. 3.17 Wireframe of Profile page of User

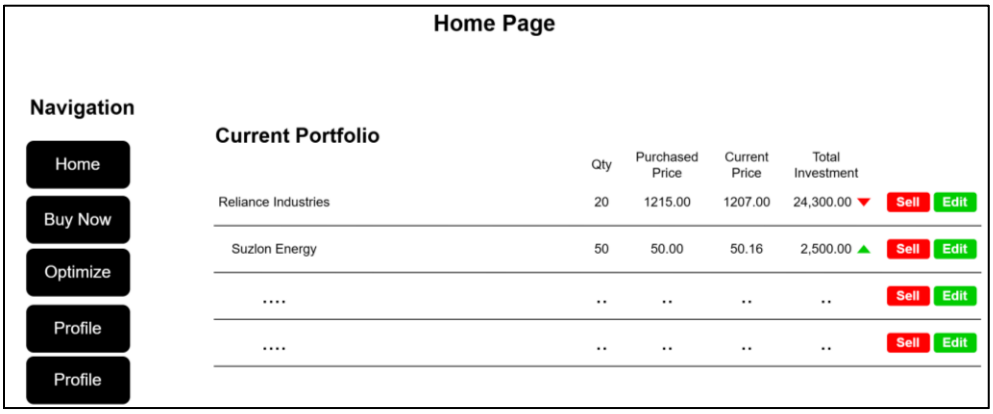

Fig. 3.18 Wireframe of Home Page of Web-App

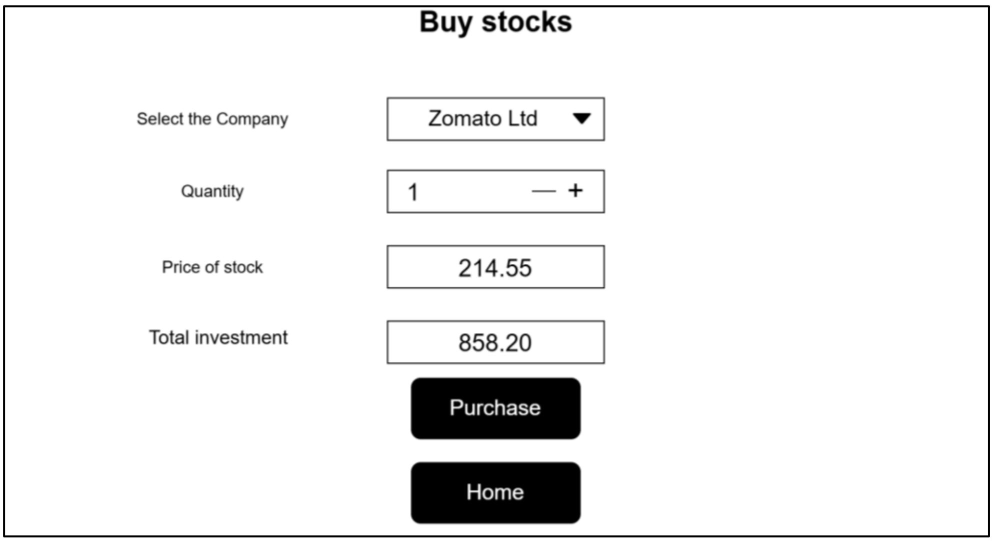

Fig. 3.19 Wireframe of stocks buying page

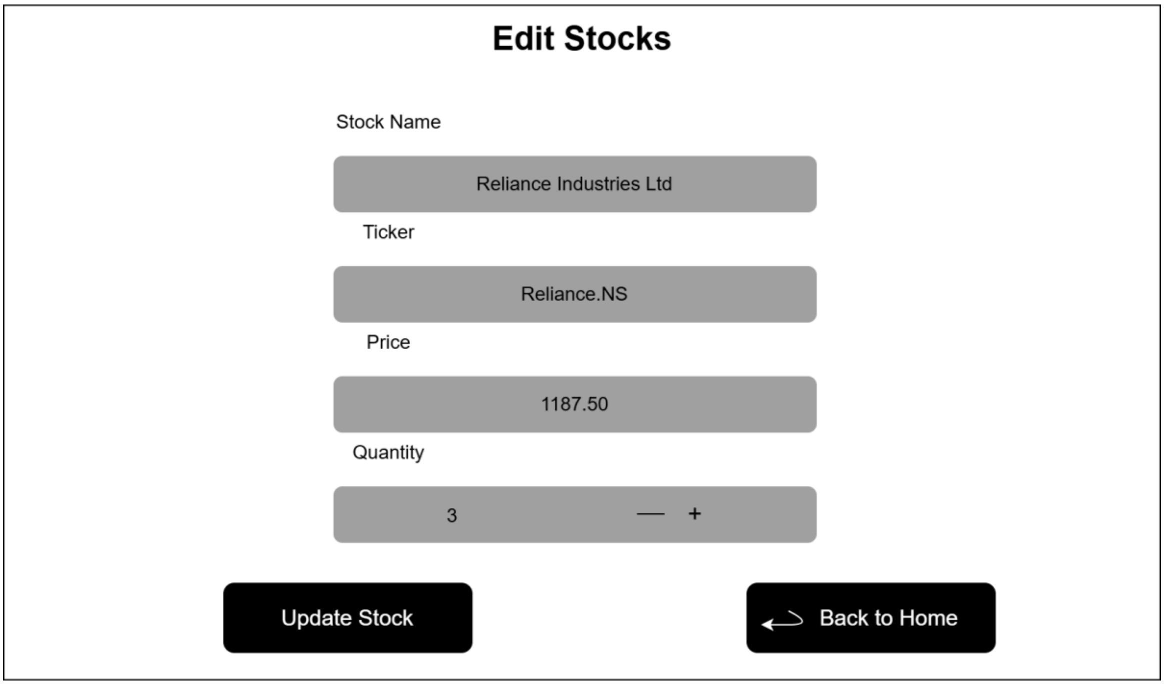

Fig. 3.20 Wireframe of Edit Stocks

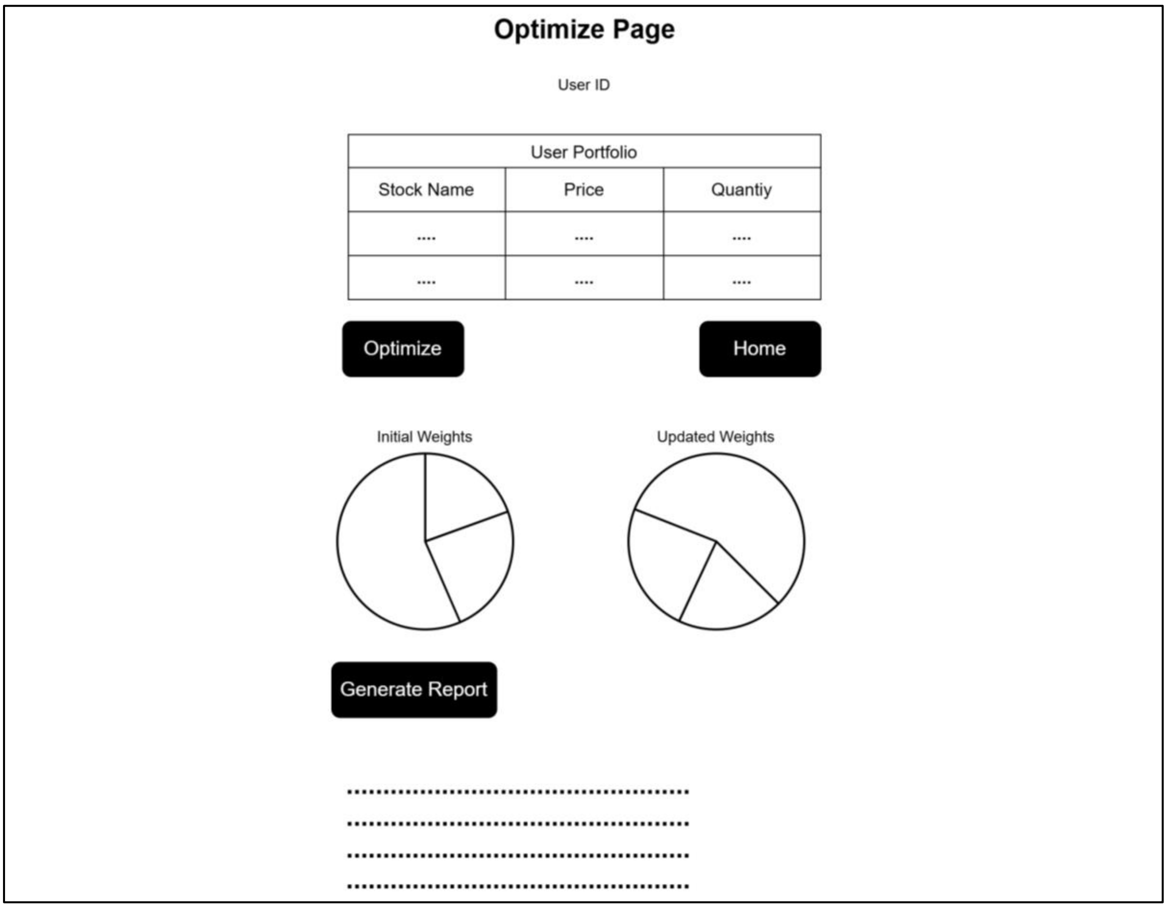

Fig. 3.21 Wireframe of Optimization Page

Fig. 3.22 – Fig. 3.25 shows the wireframes of the manager side. Fig. 3.22 shows the manager dashboard which contains analytics, Fig. 3.23 which shows all the registered users, Fig. 3.24 which shows the stocks that are added for user to transact, Fig. 3.25 shows the page to add new stocks.

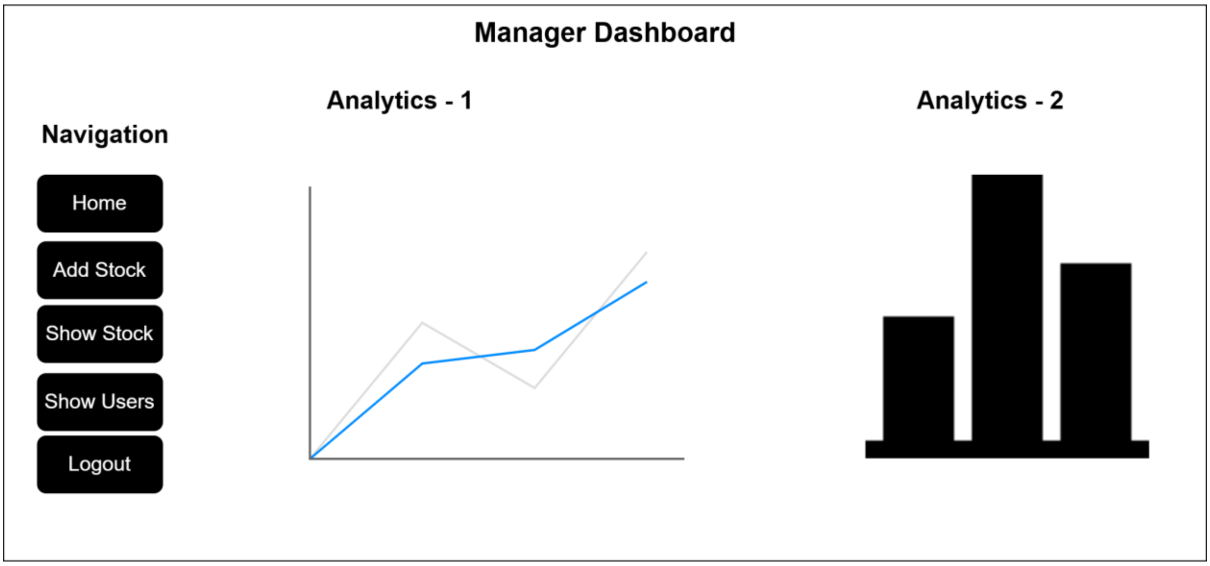

Fig. 3.22 Wireframe of Home Page – Manager Side

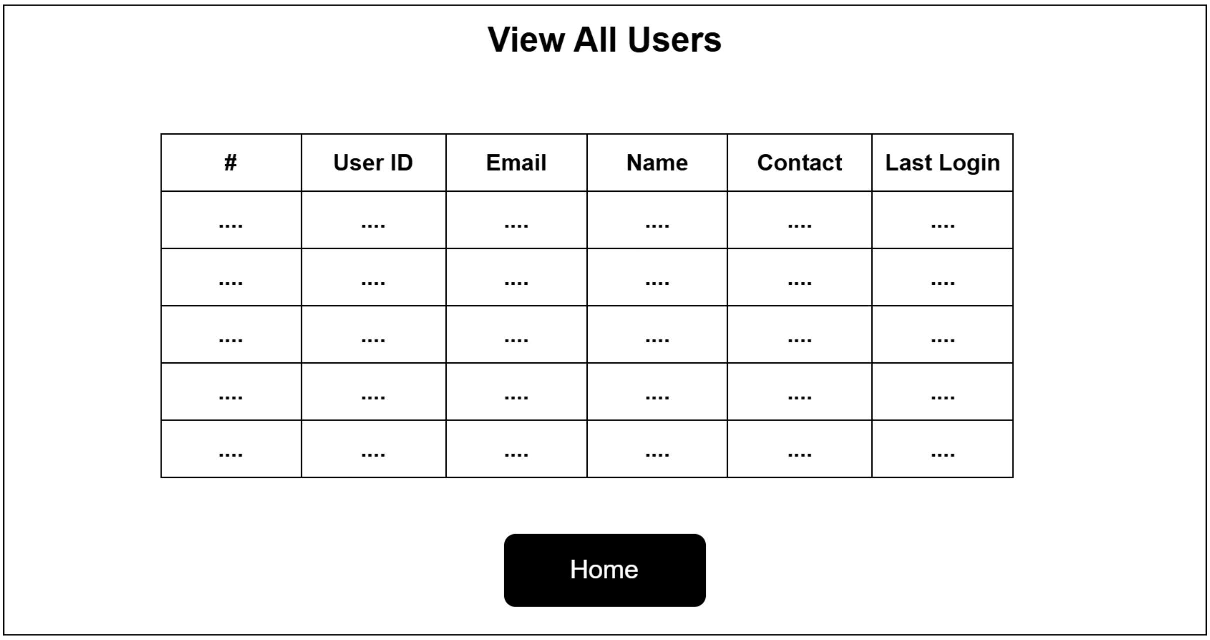

Fig. 3.23 Wireframe of list of Registered Users page

Fig. 3.24 Wireframe of list of added stocks page

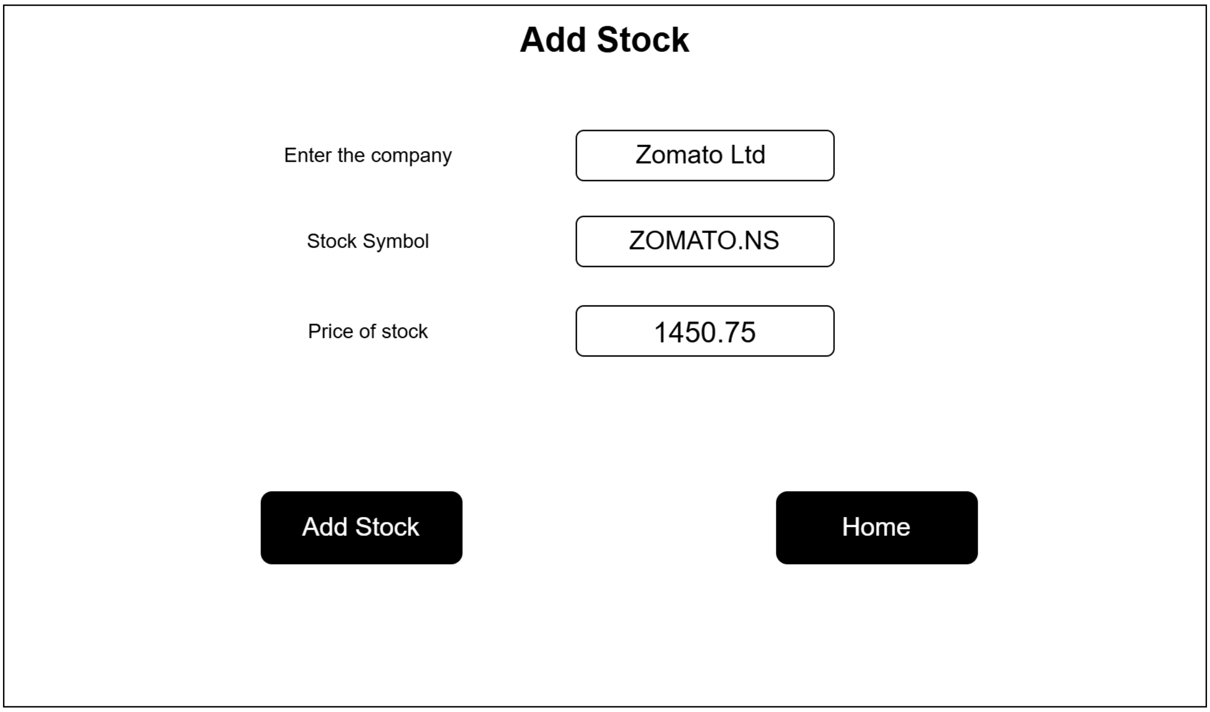

Fig. 3.25 Wireframe of add new stocks page

Fig. 3.26 shows the default page when user visits the website for the first time. It has three options: Register for new users, Login for existing users and Manger login.

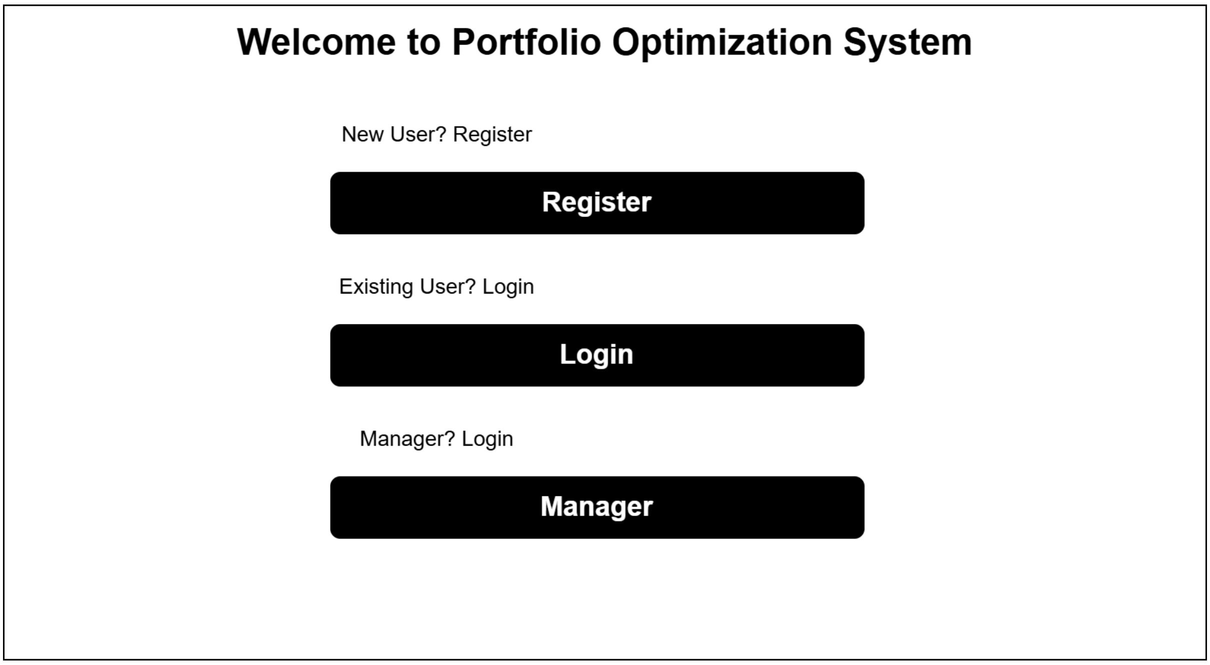

Fig. 3.26 Wireframe of the Landing Page

3.3.3 Access Control

For access control, we have implemented a role-based access control to make sure that user can access only those functionalities that are relevant to their roles. As shown in Fig. 3.22, two primary user roles exist: Investor and Database Manager. Investors manage their portfolios, optimize it, review the recommendations, buy and sell stocks, and update their profile. Also, they can accept or reject recommendations generated by the system based on their investment strategies. On the other hand, the Database Manager has higher privileges, including managing users, backup the database, monitor security, and modify stocks by adding or removing stocks. As we have Firebase as the database, we used Firebase Authentication to manage user roles. Firebase provides built-in authentication, Google Sign-In, and role-based authorization. This makes sure that investors can only access and modify their own data, Database Manager has high level permissions to manage data and security rules in Firebase Firestore uses access restrictions at the database level, preventing unauthorized access to sensitive information.

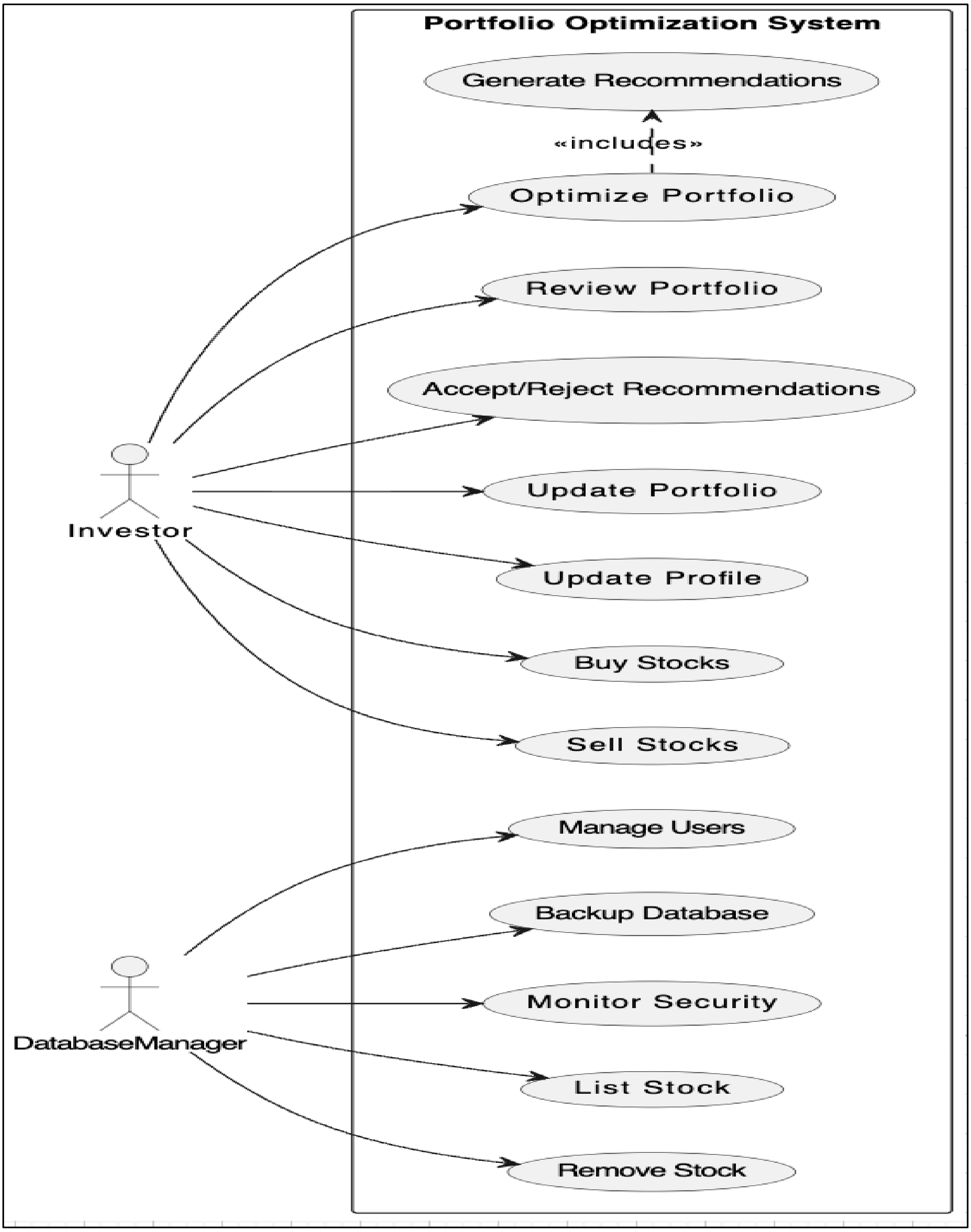

Fig. 3.27 Use Case Diagram for managing access

CHAPTER 4

IMPLEMENTATION & TESTING

4.1 Implementation Platforms

This project was implemented on free and open-source platforms. We used Jupyter Notebook for pre-processing, model training, optimization algorithms, sentiment analysis, agent creation and utilization. After the completion of model training and agents, Visual Studio Code was used in creating web-app using Streamlit and integration of models. Firebase was used as a database for user and portfolio data. Along with this, GitHub was used for managing the project allowing to work remotely. Additionally, Streamlit Cloud was used to host the web-app for real-time access from any device through the given URL.

4.2 Technologies and Modules Specifications

There were six major modules that were implemented. Each module is explained below:

- **ML Models**

  This module was responsible for stock price prediction. In this module, 25 ML models were directly trained using PyCaret library. The best performing models were then rebuilt from scratch using Scikit-Learn library and then these models were fine-tuned with new hyperparameters using Optuna. The best model was selected for stock price prediction.

- **Optimization Algorithms**

  For creating optimal portfolios, 3 nature inspired optimization algorithms were used namely PSO, GWO and BAT. All the hybrid as well as ensemble approaches were implemented for producing optimal portfolios. The algorithms are explained below:

  - **Particle Swarm Optimization (PSO):** It efficiently searches for the best asset allocation by balancing exploration (trying new solutions) and exploitation (refining known good solutions).
  - **Grey Wolf Optimization (GWO):** GWO is useful for finding the optimal trade-off between risk and reward by simulating how wolves encircle the best solution. The leader (alpha) represents the best portfolio configuration, and

others adjust their positions accordingly, leading to faster convergence to an optimal allocation.

- **BAT Algorithm:** Helps in exploring a wide range of portfolio allocations in the early stages and fine-tuning them as it converges. The frequency-based movements allow adaptive diversification, ensuring portfolios do not get stuck in local optima.

➢ **Agentic AI**

Two agents were used for portfolio optimization: Market Research Agent and Portfolio Optimization Agent which are explained below:

- **Market Research Agent:** It is responsible for continuously gathering and analysing data, current market conditions, recent news, industry trends, sentiment, and risk factors. It provides insights covering recent performance, developments, and market conditions. It validates portfolio decisions by comparing them with ongoing market movements and offers risk assessments, balancing analysis, and recommendations based on market trends. It uses LangChain for intelligent data retrieval and reasoning.
- **Portfolio Optimization Agent:** It focuses on analysing changes made to the user's portfolio. It provides a summary of allocation changes, the reason behind these changes, and evaluates risk and return. It also ensures that each recommendation aligns with real-time market context and sentiment. This agent compares optimization results with market insights. It helps validate the logic behind recommendations using risk exposure, recent events, etc. It used Groq to speed up complex calculations and generate recommendations faster.

➢ **Sentiment Analysis**

This module analyses news headlines related to a user’s selected companies to assess market sentiment. It uses BeautifulSoup for data scraping, and sentiment models like VaderSentimentAnalyser built using NLP to classify sentiments as positive, negative, or neutral.

➢ **Database**

Firebase was used as database for this project. It’s built-in authentication services and cloud-based database services helps in storing the and retrieving the data efficiently.

➢ **Web-app**

A web-application using Streamlit was created which users can access in real-time. Streamlit is a Python framework enabling quick building of interactive web applications. Streamlit simplifies interface creation by allowing developers to use Python functions to add widgets like sliders, buttons, and text inputs. Its easy connection with data science libraries and simple deployment choices makes it a popular choice for developing interactive applications.

4.3 Results

For stock price prediction, we used 25 ML regression models using PyCaret library. Out of these, top 8 best models were selected for further step and the rest were discarded because of inappropriate results. The result of the 8 models is shown in Fig. 4.1. From the figure, XGBoost, GradientBoosting, LightGBM and Ridge Regression were found optimal with minimal errors evaluated through regression metrics although some models like CatBoost, ElasticNet had less error but were not effective in real-time prediction. These 4 models were then built from scratch and tuned with hyperparameters for predicting stock prices with least errors (discussed in Section 4.4).

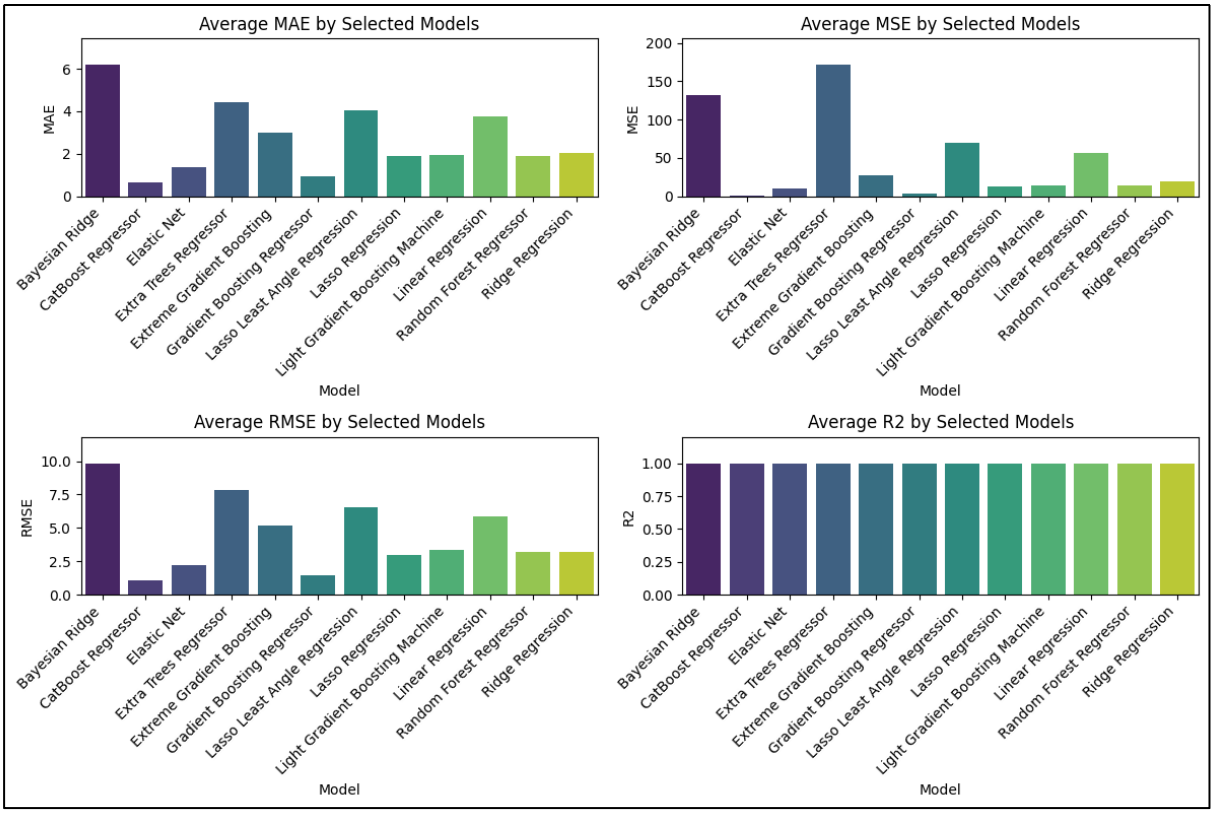

Fig. 4.1 Best performing models in stock price predictions

After predicting stock prices, the optimization algorithms were implemented using the sentiment scores and stock prices. Individual as well hybrid and ensemble approaches were considered. Fig. 4.2 shows the approximate returns obtained from each approach. From the figure, we can see that every algorithm achieved very similar returns while BAT algorithm achieved the highest returns while GWO→BAT Hybrid approach involving training with both algorithms simultaneously achieved the least returns. Fig. 4.3 shows the comparison based on the risk by each algorithm after rebalancing. GWO→BAT Hybrid achieved very less risk while BAT had the highest risk. As these two algorithms had contrasting returns, they were not preferred for the actual use.

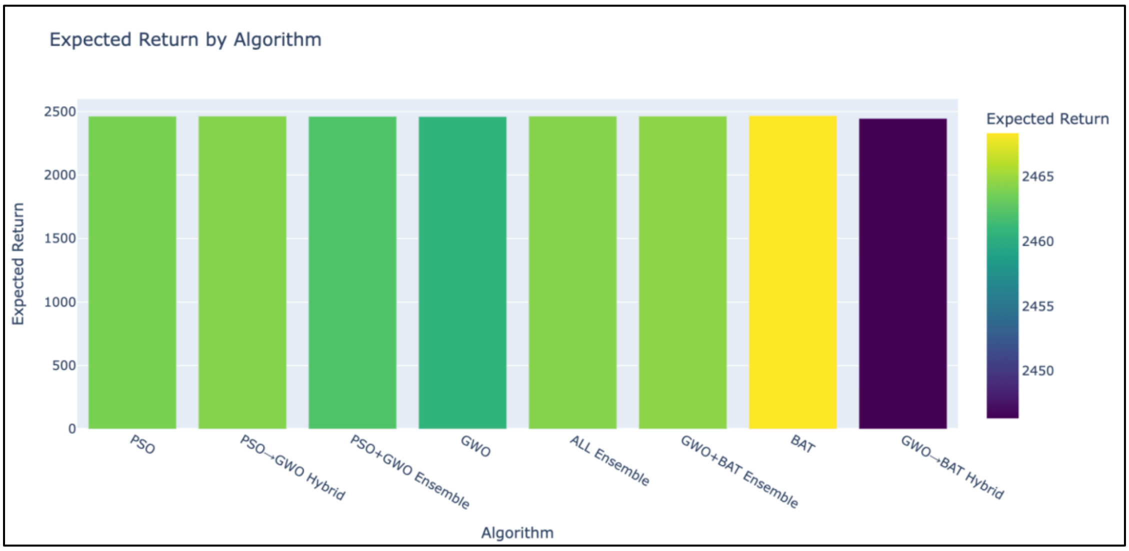

Fig. 4.2 Optimization algorithms comparison based on expected returns

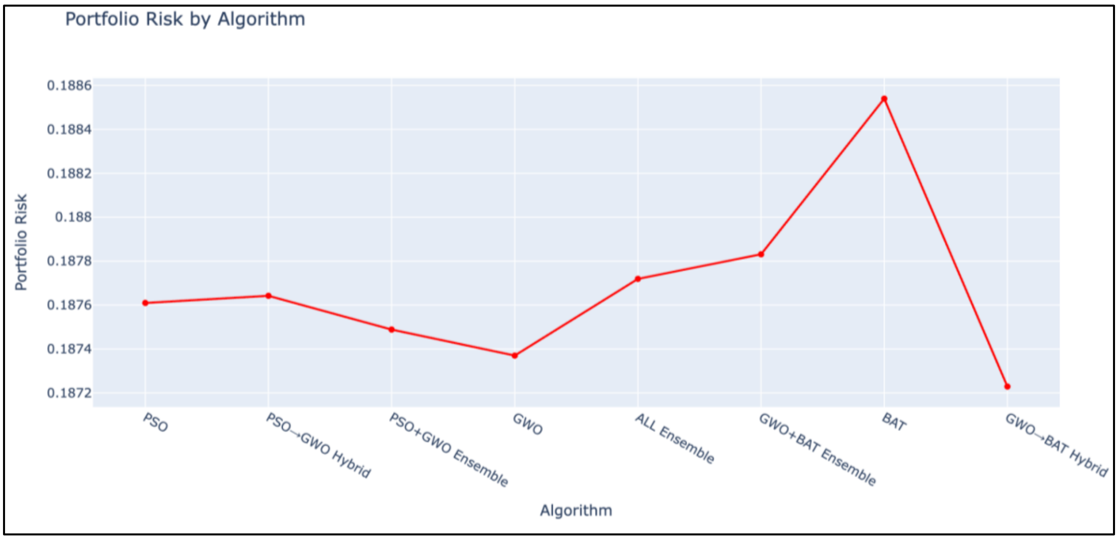

Fig. 4.3 Optimization algorithms comparison based on risk

Fig. 4.4 shows the Sharpe Ratio vs Expected Returns bubble chart. Sharpe Ratio is used to evaluate the performance of an investment by comparing its return to the level of risk involved. It helps investors understand whether they are being have sufficient returns for the risk they are taking. A higher Sharpe Ratio indicates a more favourable risk-return trade-off, making it a useful tool for comparing different portfolios or investment strategies. From the figure, we can see the ideal returns are between 2,460 – 2,465 and Sharpe Ratio is around 13,130 – 13,140. Fig. 4.5 shows the optimized portfolios by each algorithm for a sample portfolio consisting of 5 companies. As the dummy data was considered, the major difference is not visible due to minor variations (decimal points changes) between initial and final weights.

Fig. 4.4 Algorithm comparison based on Sharpe ratio vs returns

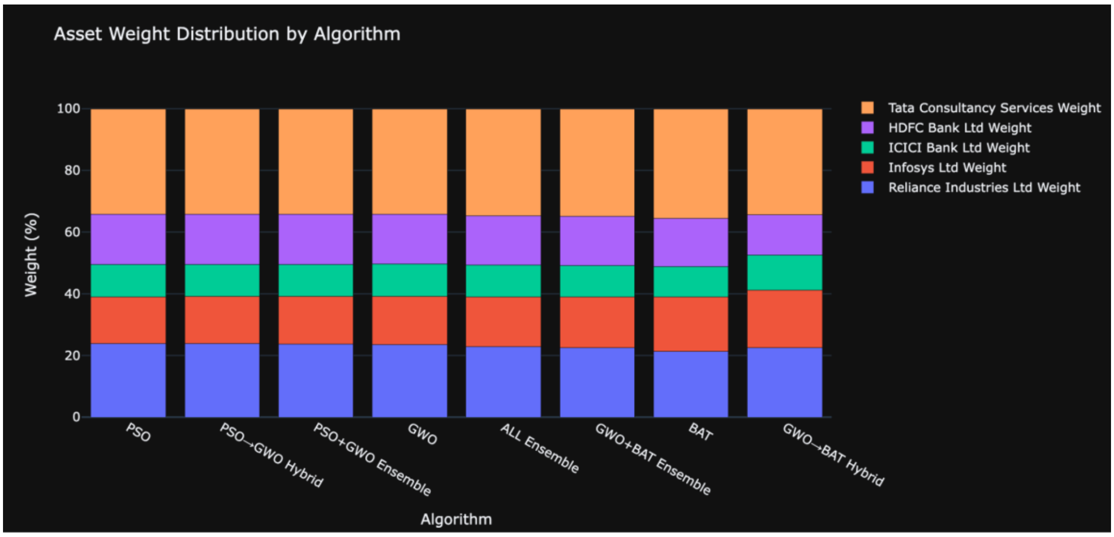

Fig. 4.5 Algorithm comparison based on portfolio weight distribution

4.4 Result Analysis and Model Comparisons

The best models i.e. XGBoost, LightGBM, Gradient Boosting and Ridge were fine tuned using new and best hyperparameters. The difference in results is shown in Fig. 4.6. From the figure, the errors are significantly decreased for every algorithm. Among these algorithms, Ridge Regression had the least error in predicting stock prices then all the other algorithms which helped in predicting stock prices with less errors.

Among all the optimization approaches, PSO had the most balanced optimization evaluation based on risks, returns and Sharpe Ratio. The comparison of optimization algorithm results is shown from Fig. 4.7 – 4.14. It consists of sample portfolio distribution (percentage of the stocks in portfolio). The initial portfolio is taken as the input and the optimization algorithm is applied on this portfolio. These algorithms use the agents (market and portfolio) created along with sentiment analysis and updates the weights of the stocks based on all the available information available on internet. This helps the user to decide to rebalance the portfolio or not. Fig. 4.7 shows the updated weights of the portfolio using PSO algorithm. From the figure, it shows 4% decrease in ICICI Bank stocks compared to initial portfolio suggesting user to sell the stocks while 5% increase in TCS stocks indicating to buy while rest to hold. Fig. 4.8 – 4.14 provides similar insights using different algorithms as well as hybrid and ensemble approaches.

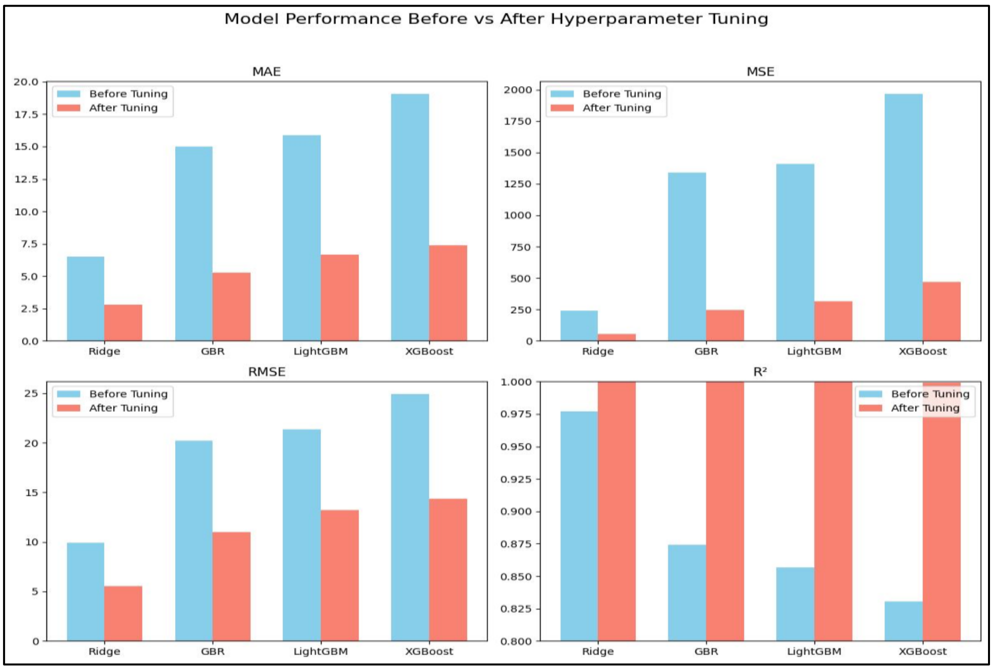

Fig. 4.6 Model comparison before and hyperparameter tuning

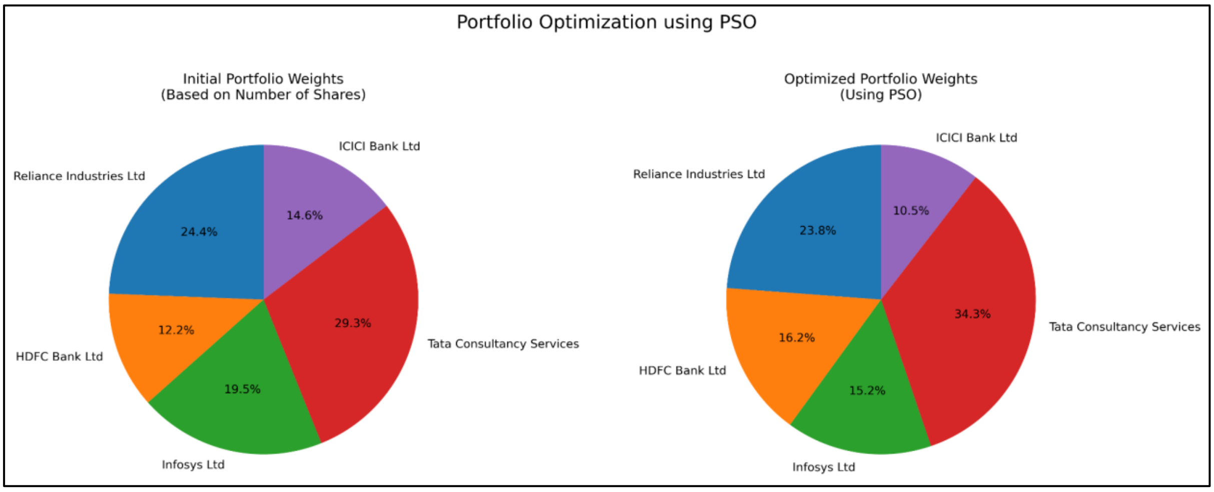

Fig. 4.7 Portfolio Optimization using PSO (Initial and Optimized)

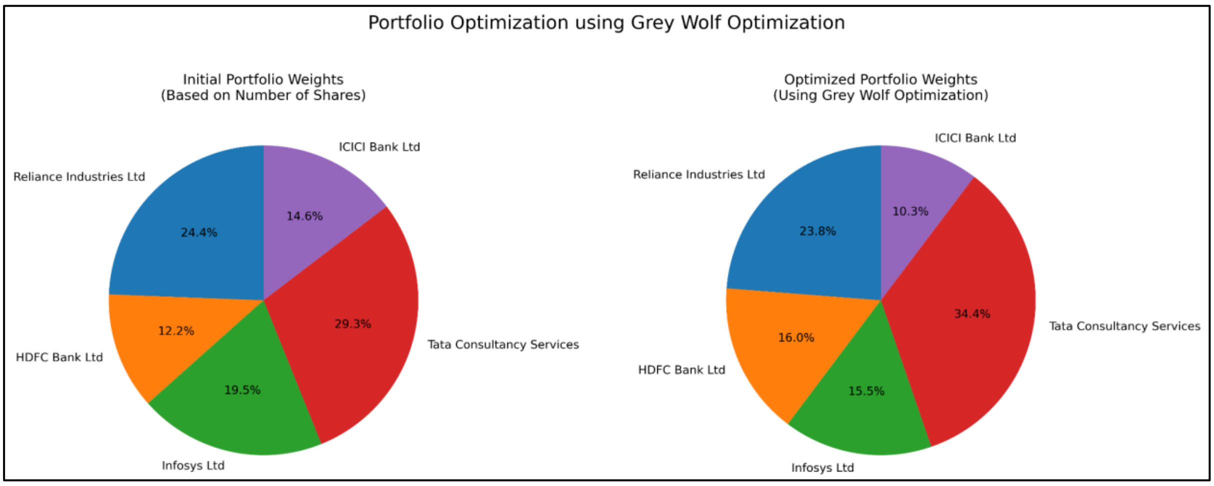

Fig. 4.8 Portfolio Optimization using GWO (Initial and Optimized)

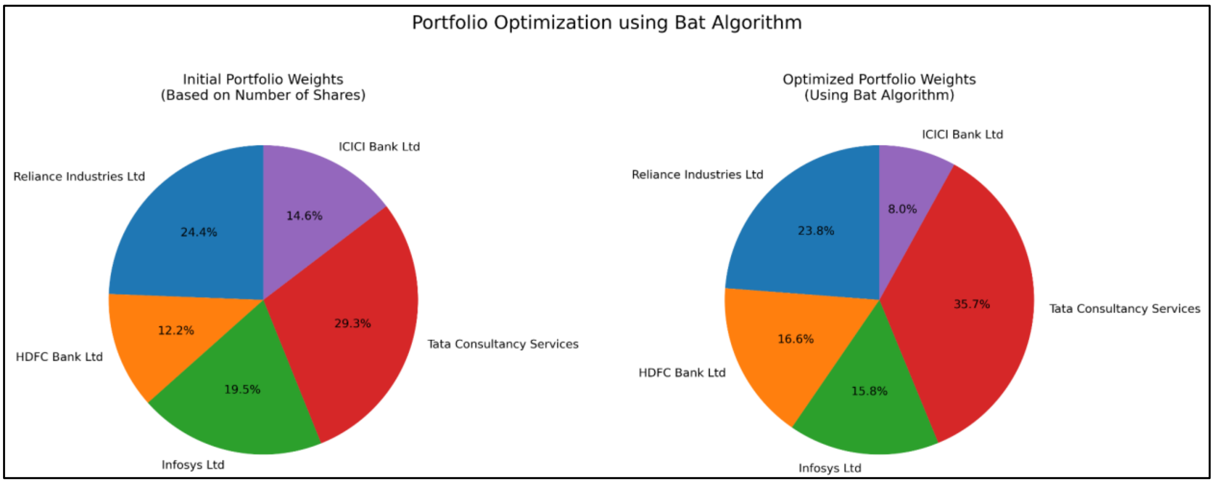

Fig. 4.9 Portfolio Optimization using Bat Algorithm (Initial and Optimized)

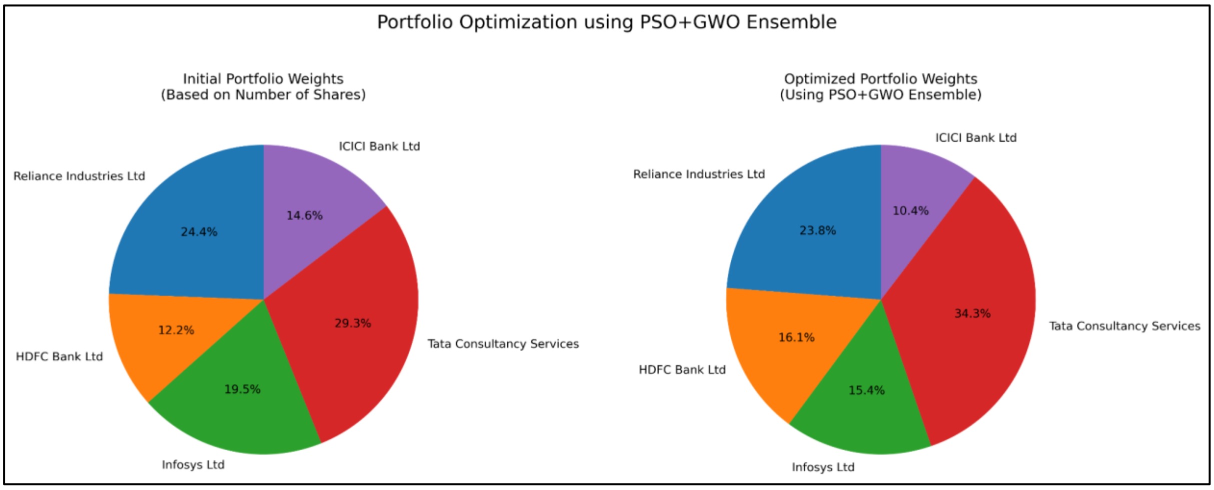

Fig. 4.10 Portfolio Optimization using PSO + GWO Ensemble (Initial and Optimized)

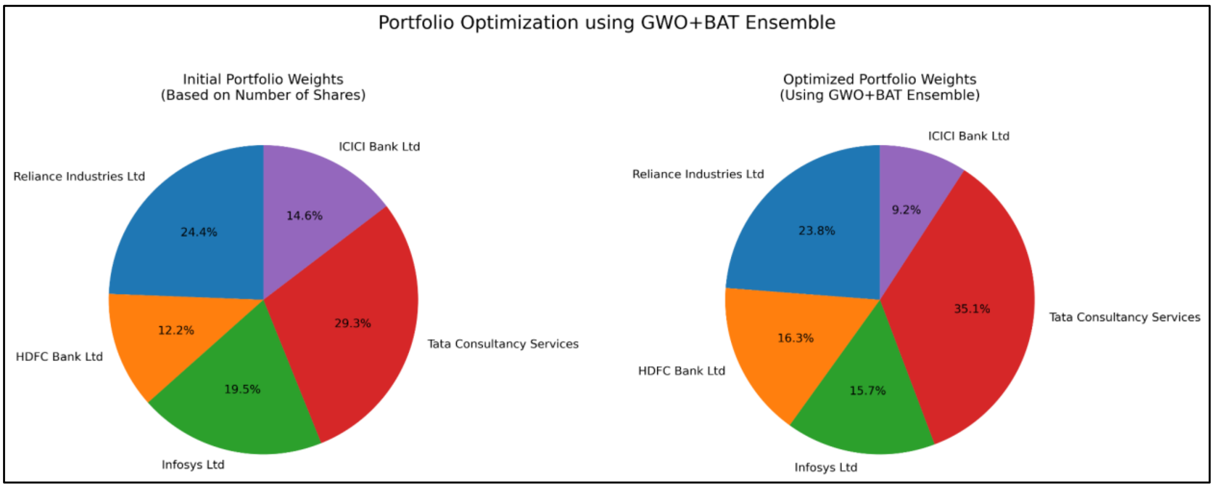

Fig. 4.11 Portfolio Optimization using GWO + BAT Ensemble (Initial and Optimized)

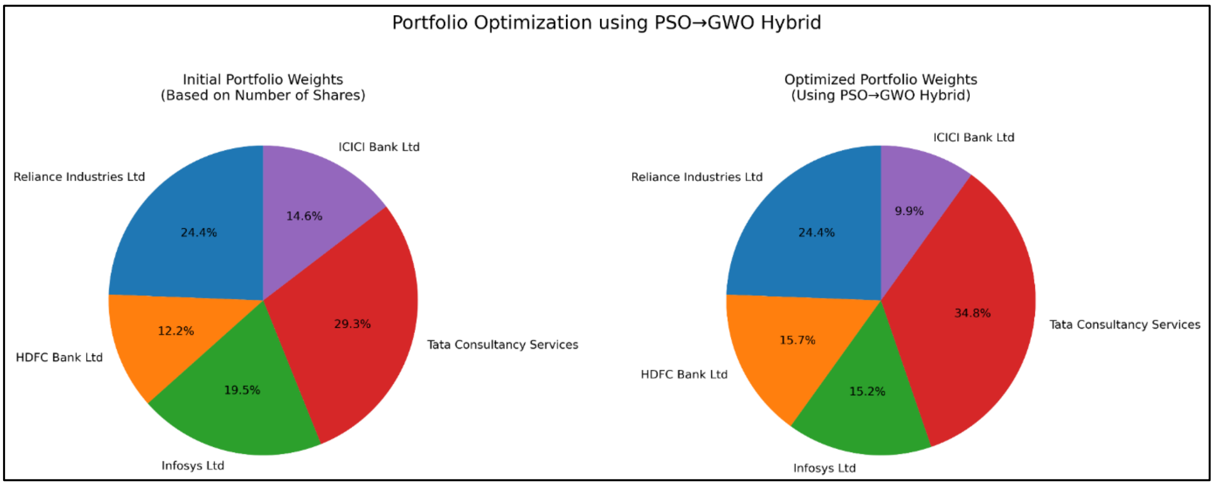

Fig. 4.12 Portfolio Optimization using PSO → GWO Hybrid (Initial and Optimized)

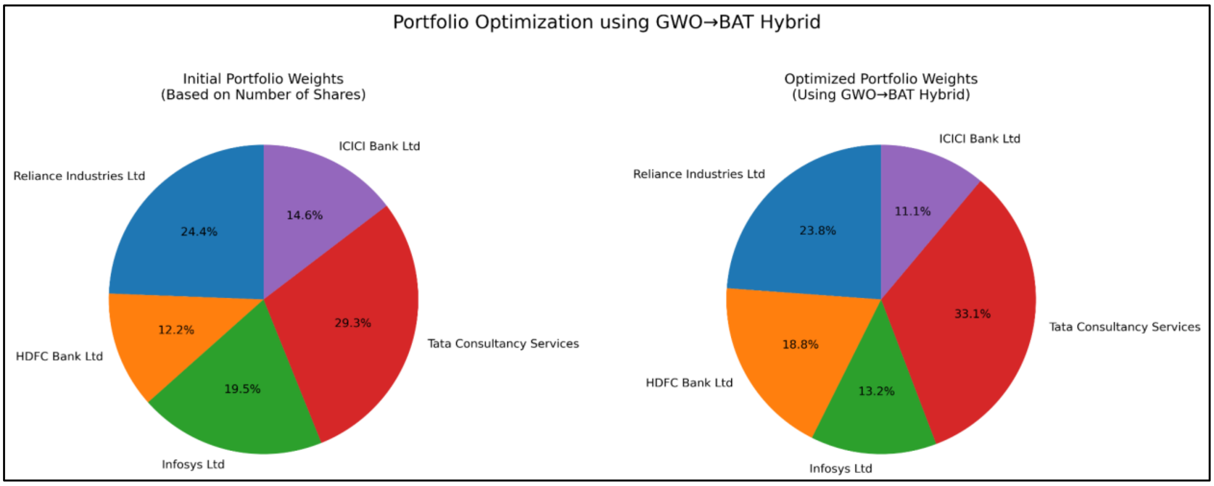

Fig. 4.13 Portfolio Optimization using GWO → BAT Hybrid (Initial and Optimized)

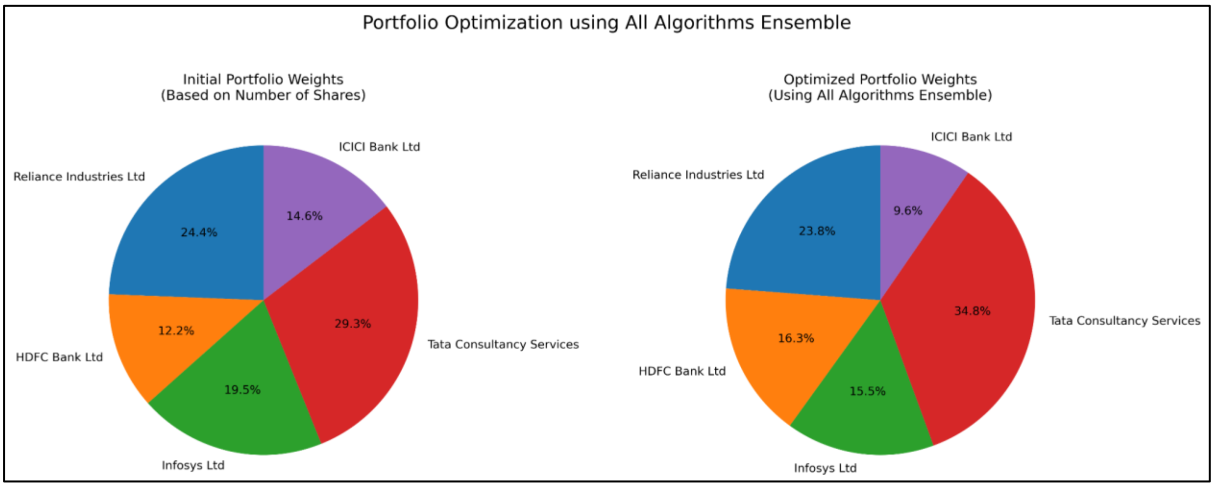

Fig. 4.14 Portfolio Optimization using All Algorithms Ensemble (Initial and Optimized)

Fig. 4.15 shows the sample of report generated by Agentic AI based on the user portfolio. It shows the detailed insights of the market, provides sentiment scores, analyses closing prices, last 1 – year return trends. Fig. 4.16 shows the suggested changes of the portfolio which includes justification of why the changes are recommended, initial and optimized weights, risk analysis. This in general provides highly detailed optimization reports which helps the investors to make the investments after considering every aspect of the market.

Validation Report: Tata Consultancy Services Ltd Portfolio Recommendation

Current Market Conditions:

Indicator	Value
Market Index (NIFTY)	18,450.50
Sector Index (IT)	5,530.25
Tata Consultancy Services Ltd Stock Price	₹3,535.00

Given the IT sector's strong performance (up 10% YTD) and Tata Consultancy Services Ltd's consistent earnings growth, the recommended weight increase seems aligned with the current market conditions.

Recent News and Sentiment:

News Sentiment	Value
Sentiment Score	0.05 (Neutral)

Despite the lack of recent news, the neutral sentiment could be due to the absence of significant market-moving events. However, this should not significantly impact the recommendation.

Potential Risks or Opportunities:

1. **Over-reliance on a single sector:** As the IT sector is a significant contributor to the Indian economy, its performance can have a substantial impact on the overall market. A higher weight in Tata Consultancy Services Ltd may increase the portfolio's sector concentration risk.
2. **Competition and Market Saturation:** The IT services market is highly competitive, and market saturation could lead to reduced growth prospects for Tata Consultancy Services Ltd.
3. **Geopolitical Risks:** The company's reliance on international clients exposes it to geopolitical risks, such as trade wars or economic sanctions.

Reasonableness of the Change:

Initial Weight	Recommended Weight	Change
33.3%	35.8%	2.5%

The recommended weight increase of 2.5% seems reasonable given the company's strong performance and consistent earnings growth. However, the increase is not drastic, suggesting that the portfolio manager is cautious and considering the potential risks.

Validation Conclusion: The recommended weight increase in Tata Consultancy Services Ltd seems aligned with the current market conditions, but it is essential to monitor the company's performance and the IT sector's growth prospects. The neutral sentiment and lack of recent news do not significantly impact the recommendation. Potential risks, such as over-reliance on a single sector and competition, should be carefully considered before implementing the recommended change." "Mahindra And Mahindra Ltd":"**Validation of Portfolio Recommendation**

Company Overview: Mahindra And Mahindra Ltd

Financial Metric	Current Value	1-Year Change	5-Year Change
Market Capitalization	₹1,04,111 Cr	-12.5%	51.4%
Revenue	₹39,444 Cr	-9.2%	24.1%
Net Profit	₹1,350 Cr	-25.6%	17.3%
PE Ratio	17.6	-34.5%	36.4%

Recent Market Performance

Index	1-Year Change	5-Year Change
Nifty 50	-8.2%	44.1%
Sensex	-7.4%	45.5%

News and Sentiment Analysis

According to the provided data, there is no recent news found for Mahindra And Mahindra Ltd, and the sentiment score is 0.05, which is very low (minimum value is 0).

Analysis

1. **Alignment with current market conditions**: The recommended weight decrease of -18.1% is higher than the market's 1-year change of -12.5% and -9.2% in revenue. The company's performance is also lagging behind the market indices (Nifty 50 and Sensex) over the past year. This suggests that the market may be moving away from Mahindra And Mahindra Ltd, making the recommended weight decrease seem reasonable.
2. **Support or contradiction from recent news and sentiment**: The lack of recent news and a very low sentiment score indicates a negative sentiment towards the company. This supports the recommended weight decrease, as the market may be pricing in a potential decline in the company's performance.
3. **Potential risks or opportunities**: The company's decreasing revenue and net profit, along with a high PE ratio, may indicate a potential risk in the short term. However, the company's 5-year revenue growth and market capitalization increase suggest opportunities in the long term.
4. **Reasonableness of the change**: Given the company's current performance, the recommended weight decrease of -18.1% seems reasonable. However, investors should consider the potential risks and opportunities before making any investment decisions.

Conclusion: Based on the analysis, the recommended weight decrease of -18.1% for Mahindra And Mahindra Ltd seems reasonable, considering the company's current market performance and the lack of recent news. However, investors should carefully consider the potential risks and opportunities before making any investment decisions." "Titan Company Ltd":"**Validation of Portfolio Recommendation for Titan Company Ltd**

1. Alignment with Current Market Conditions

To determine if the recommended change aligns with current market conditions, let's analyze the company's recent stock performance.

Date	Close Price
01/01/2024	2,434.75
01/15/2024	2,581.50
02/01/2024	2,699.25
02/15/2024	2,765.00
03/01/2024	2,822.50
03/15/2024	2,935.75

As shown in the table, the stock price has been increasing steadily over the past few months. This positive trend suggests that the market is favoring the company, which could justify an increase in the recommended weight.

2. Recent News and Sentiment

The provided news and sentiment analysis indicate a neutral sentiment score of 0.05. This score suggests that there is no significant positive or negative sentiment associated with the company. Given the lack of recent news, we cannot draw conclusions based on news events.

3. Potential Risks or Opportunities to Consider

1. **Market Volatility**: The Indian stock market has been experiencing high volatility in recent months. This could impact the company's stock price and affect the recommended weight.
2. **Industry Trends**: The watch and jewelry industry is highly competitive, and Titan Company Ltd faces intense competition from other players. Any changes in consumer preferences or market trends could impact the company's performance.
3. **Economic Factors**: The Indian economy has been facing challenges in recent times, including a slowdown in growth and high inflation. This could impact the company's sales and profitability.

4. Reasonableness of the Change

Given the company's positive stock performance and the lack of recent news, the recommended change of 15.7% seems reasonable. However, considering the potential risks and opportunities mentioned above, it is essential to monitor the company's performance closely and adjust the recommended weight accordingly.

Validation Conclusion

The recommended change in the portfolio weight for Titan Company Ltd seems reasonable based on the company's recent stock performance. However, it is essential to consider the potential risks and opportunities associated with the company and the market. A closer monitoring of the company's performance and market conditions is necessary to adjust the recommended weight accordingly.

Fig. 4.15 Sample of the market report

Portfolio Optimization Analysis Report

Summary of Major Changes in Portfolio Allocation

Company	Initial Weight	Optimized Weight	Change
Tata Consultancy Services Ltd	0.3333	0.3579	+0.0246 (+7.38%)
Mahindra And Mahindra Ltd	0.4667	0.2854	-0.1813 (-38.85%)
Titan Company Ltd	0.2000	0.3567	+0.1567 (+78.35%)

The optimized portfolio allocation recommends a significant shift in the weighting of the three companies. Tata Consultancy Services Ltd (TCS) is increased by 7.38%, Mahindra And Mahindra Ltd (M&M) is decreased by 38.85%, and Titan Company Ltd (TITAN) is increased by 78.35%.

Explanation of Why These Changes Were Recommended

The optimization algorithm has adjusted the portfolio weights based on the expected returns and risks associated with each company. The changes suggest that the algorithm has downgraded the importance of M&M, possibly due to its relatively lower expected return and higher risk compared to TCS and TITAN.

The increased weight on TCS is likely due to its strong expected return and relatively lower risk. TITAN's increased weight is consistent with its high expected return and moderate risk.

Analysis of Risk and Return Implications

Company	Expected Return	Portfolio Risk
Tata Consultancy Services Ltd	0.1042	0.1185
Mahindra And Mahindra Ltd	0.0755	0.2309
Titan Company Ltd	0.1239	0.1873

The optimized portfolio metrics indicate an expected return of 30.44% and a portfolio risk of 22.48%. The Sharpe ratio, a measure of excess return per unit of risk, is 13544.38, suggesting a high-risk, high-reward portfolio.

Market Context and Validation of Recommendations

As of 2025-04-15, the Indian stock market is experiencing a strong bull run, with the Nifty 50 index reaching a new high. TCS, a technology giant, has been a major contributor to the market's growth, driven by its strong performance in the IT sector.

M&M, a diversified conglomerate, has faced challenges in recent quarters due to supply chain disruptions and increased competition in its automotive business. TITAN, a luxury goods company, has benefited from the growing demand for premium products in India.

Given the current market context, the optimized portfolio allocation appears to be aligned with the market trends. However, it is essential to monitor the portfolio's performance and adjust the allocation as needed to ensure that it remains aligned with the investor's risk tolerance and investment objectives.

Potential Risks or Concerns to Consider

1. **Concentration risk**: The optimized portfolio allocation is heavily concentrated in the IT sector, which may increase the risk of a sector-specific downturn.
2. **Valuation risk**: TCS and TITAN are trading at high valuations, which may lead to a decline in their stock prices if the market becomes more cautious.
3. **Economic risk**: The Indian economy is experiencing a slowdown, which may impact the performance of companies in the portfolio.

In conclusion, the optimized portfolio allocation appears to be a well-diversified and risk-managed portfolio that takes into account the current market trends and company performance. However, it is essential to monitor the portfolio's performance and adjust the allocation as needed to ensure that it remains aligned with the investor's risk tolerance and investment objectives.

Fig. 4.16 Sample of Portfolio Optimization Report

4.5 Testing Results

After all the model training, optimizing portfolios and utilizing agents, a web-based application was created using Streamlit and Flask. Streamlit is a Python framework used for creating interactive web-apps while Flask for handling backend. Streamlit is implemented because of its ease of use where frontend components can be created using simple Python functions. Fig. 4.17 – 4.29 show the all the pages of the web-app. Fig. 4.17 is the default sign in page where user can choose to register (first time users) or login directly. The Database Manager can login through appropriate credentials. Fig. 4.18 shows the login page. Here user can login through their email and password into the website. Fig. 4.19 shows the register pages where new user register for the 1st time by entering the all the necessary details. After registration, user can login through the credentials. Fig. 4.20 shows the home page of the user after logging into the website. Main area consists of the current portfolio of the user, which they can edit or sell the stocks. On the left, are the various options navigation buttons to navigate to different pages like optimize page, buy page, sell page, profile page.

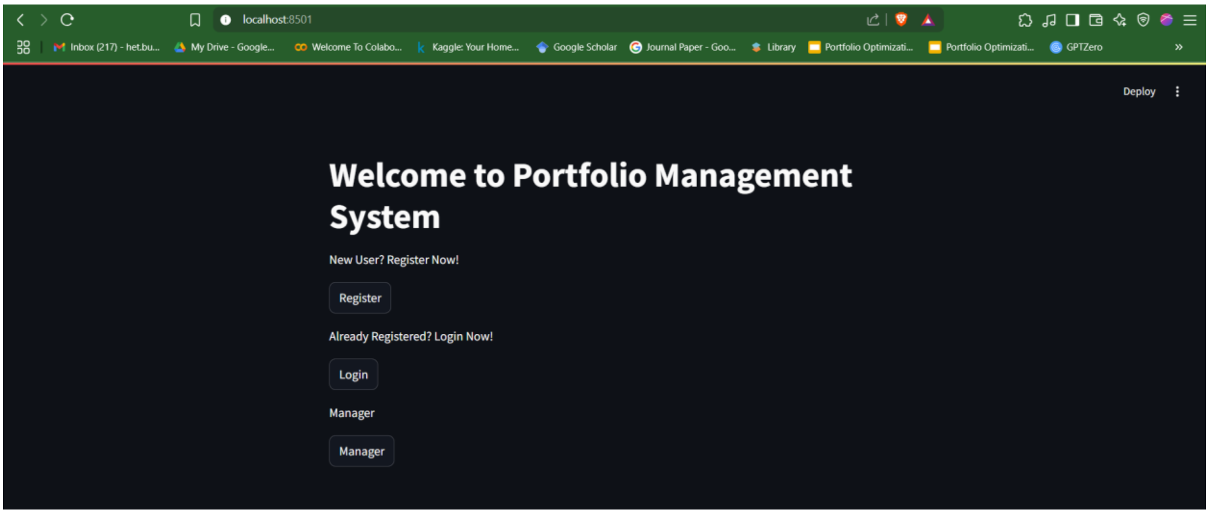

Fig. 4.17 Home page of the web-app

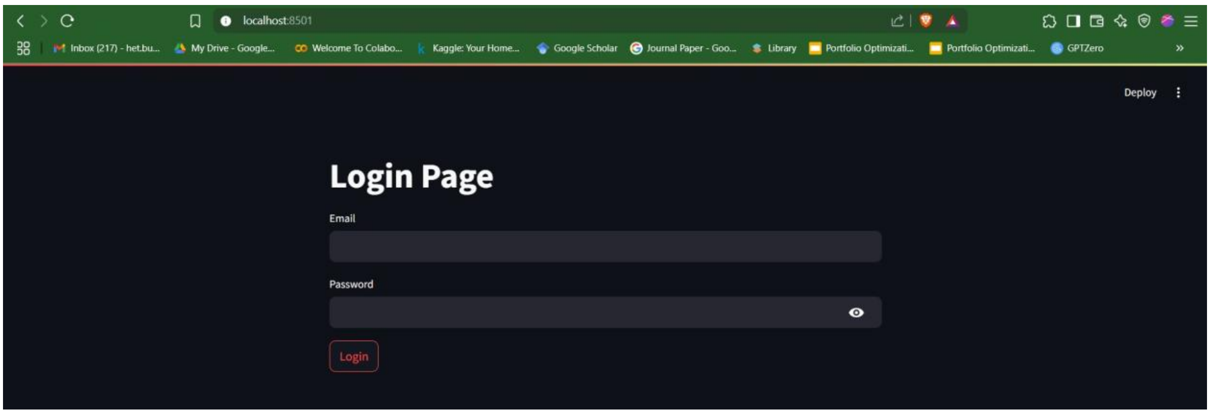

Fig. 4.18 Login Page

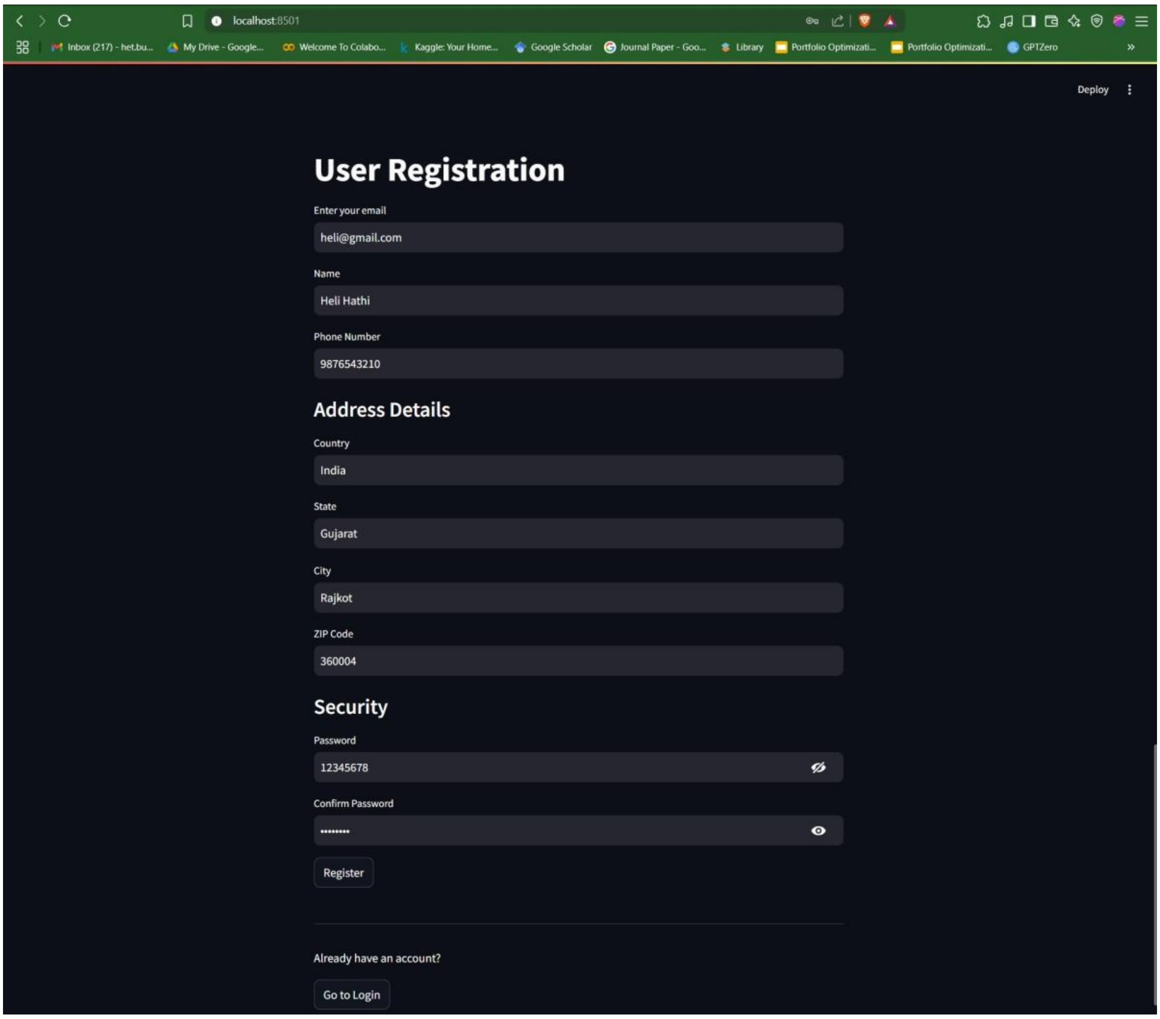

Fig. 4.19 User Registration Page

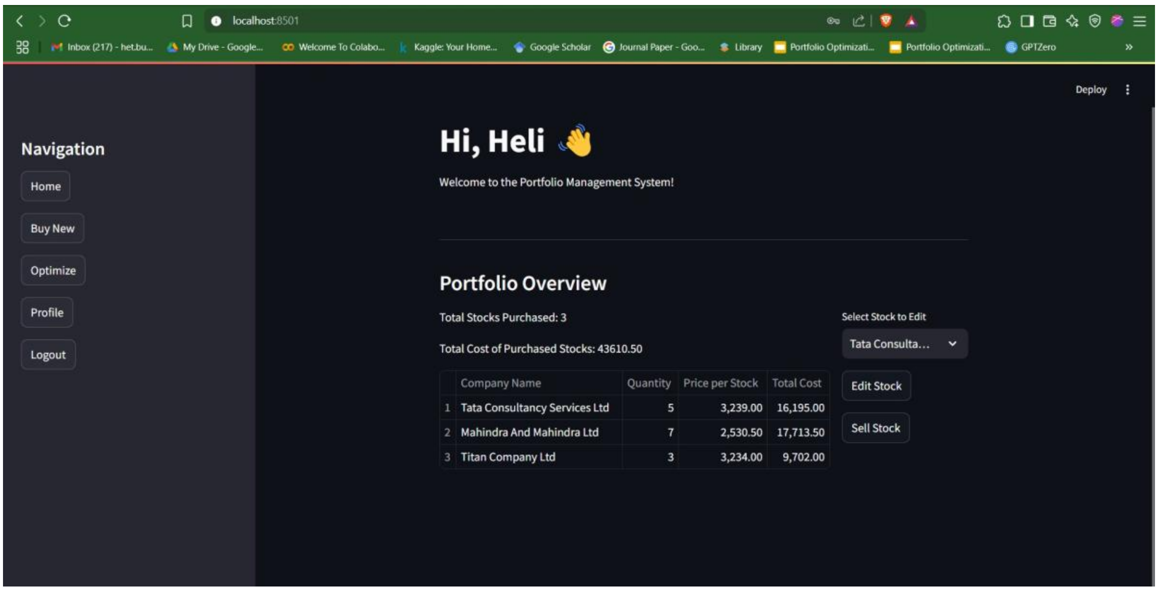

Fig. 4.20 Home page of the user

Fig. 4.21 shows the stocks buying page where you can buy the stocks of any quantity listed by the manager. Fig. 4.22 shows the edit page where user can modify their existing holdings. User can increase or decrease the company's shares from their portfolios.

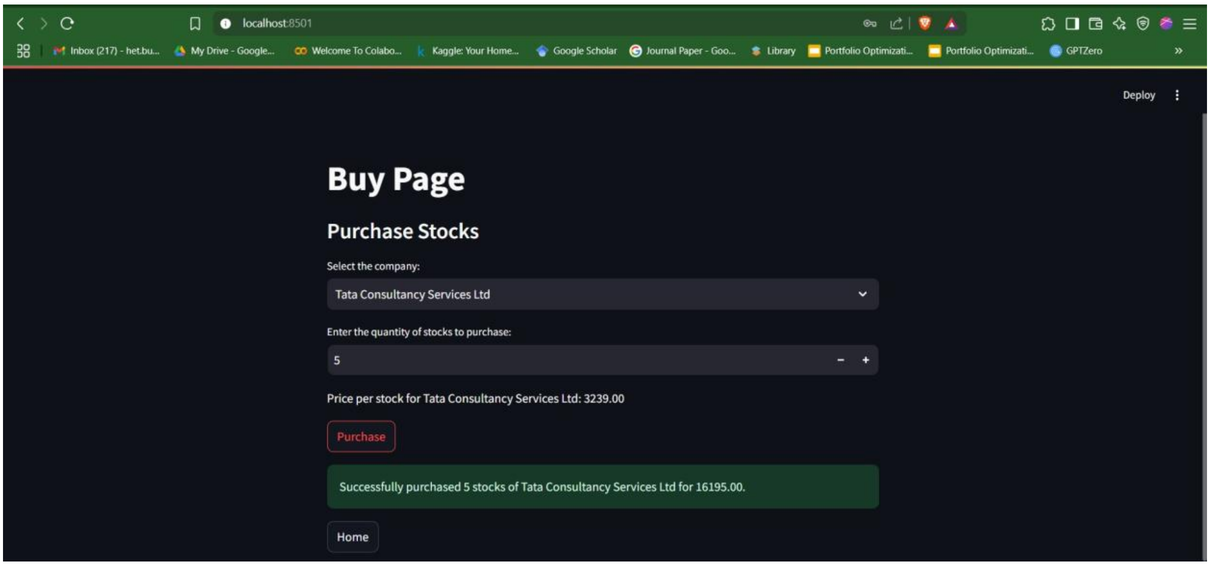

Fig. 4.21 Buy Stocks Page

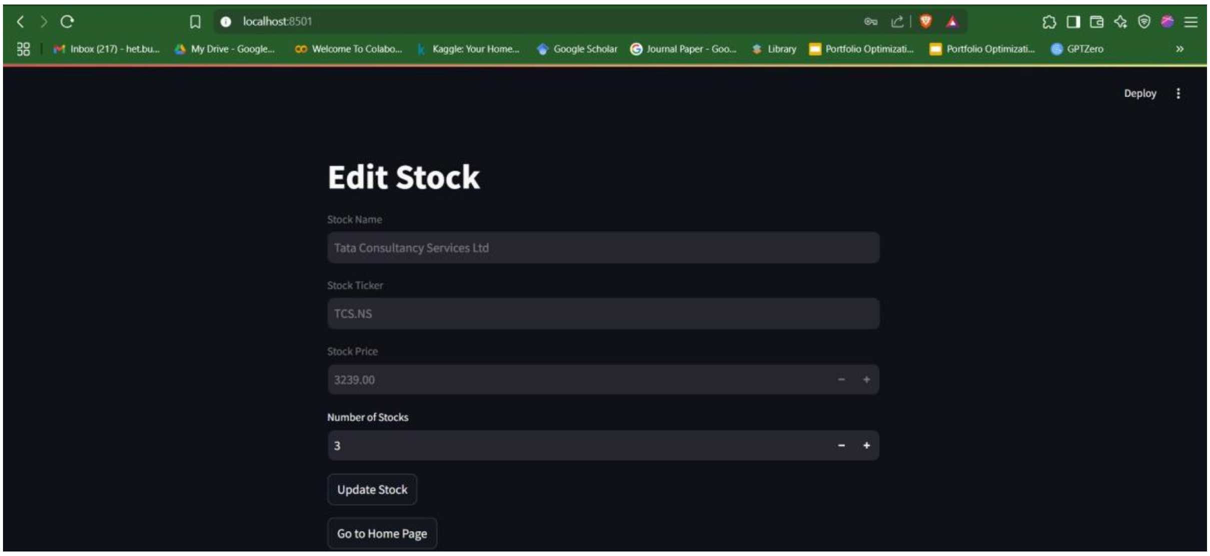

Fig. 4.22 Edit Stocks Page

Fig. 4.23 shows the profile page of user. It shows the personal details of the user. Fig. 4.24 is the main page of the website. It shows the optimization page where the suggestions are provided based on market conditions and user's portfolios. It provides the updated portfolio chart with performance metrics (Fig. 4.24) along with the detailed report and justifications for the changes in the portfolio (Fig. 4.25) and company specific report (Fig. 4.26).

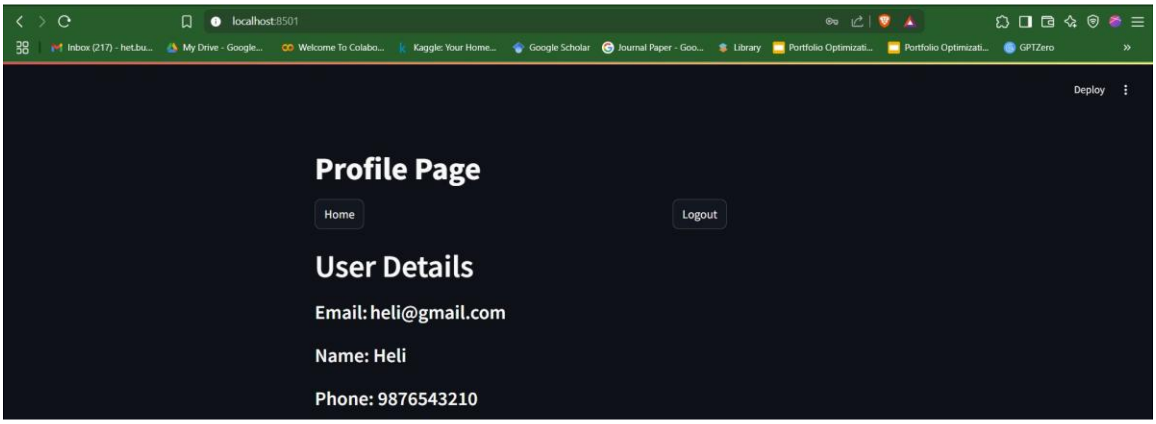

Fig. 4.23 Profile Page

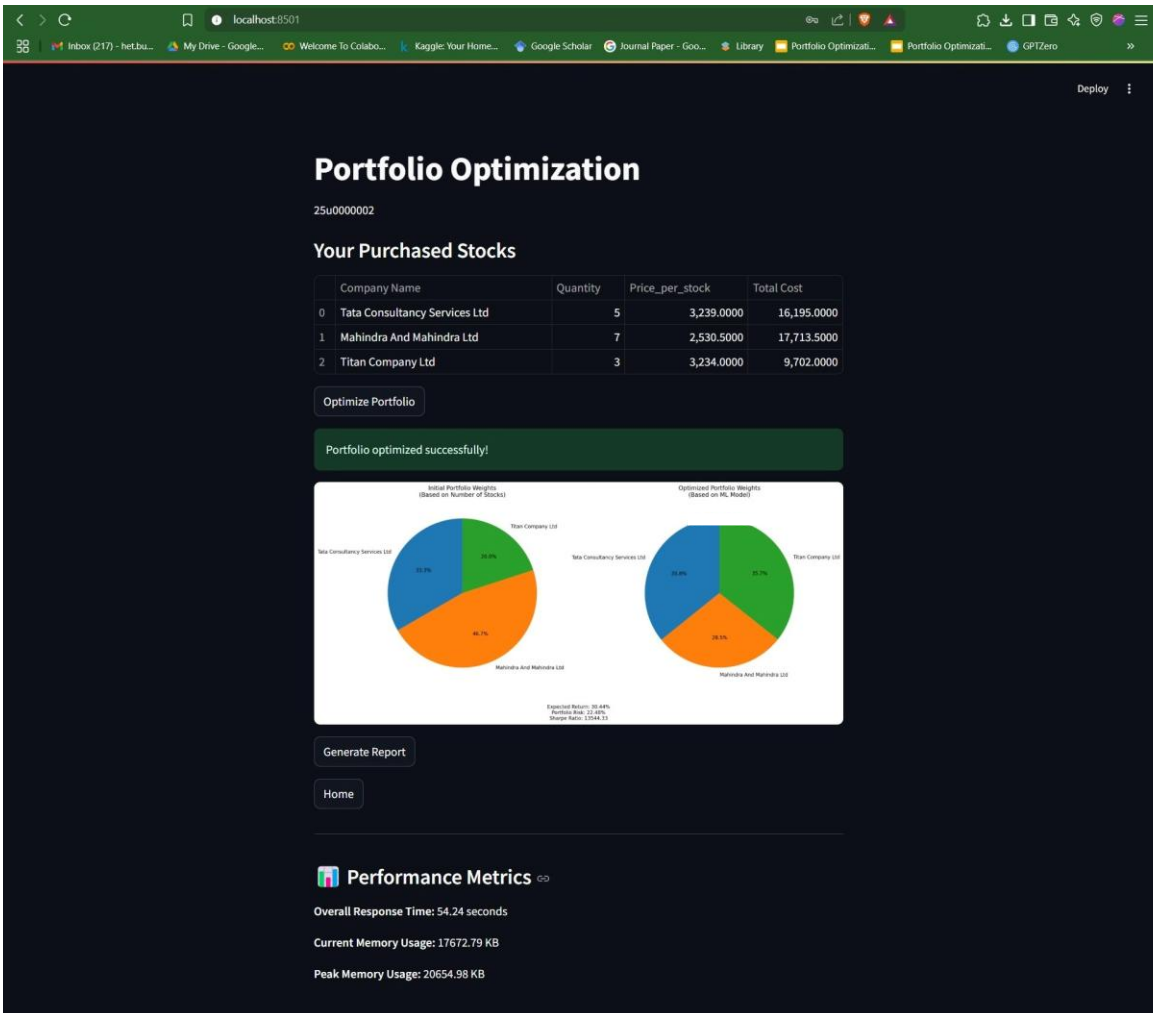

Fig. 4.24 Optimization Page

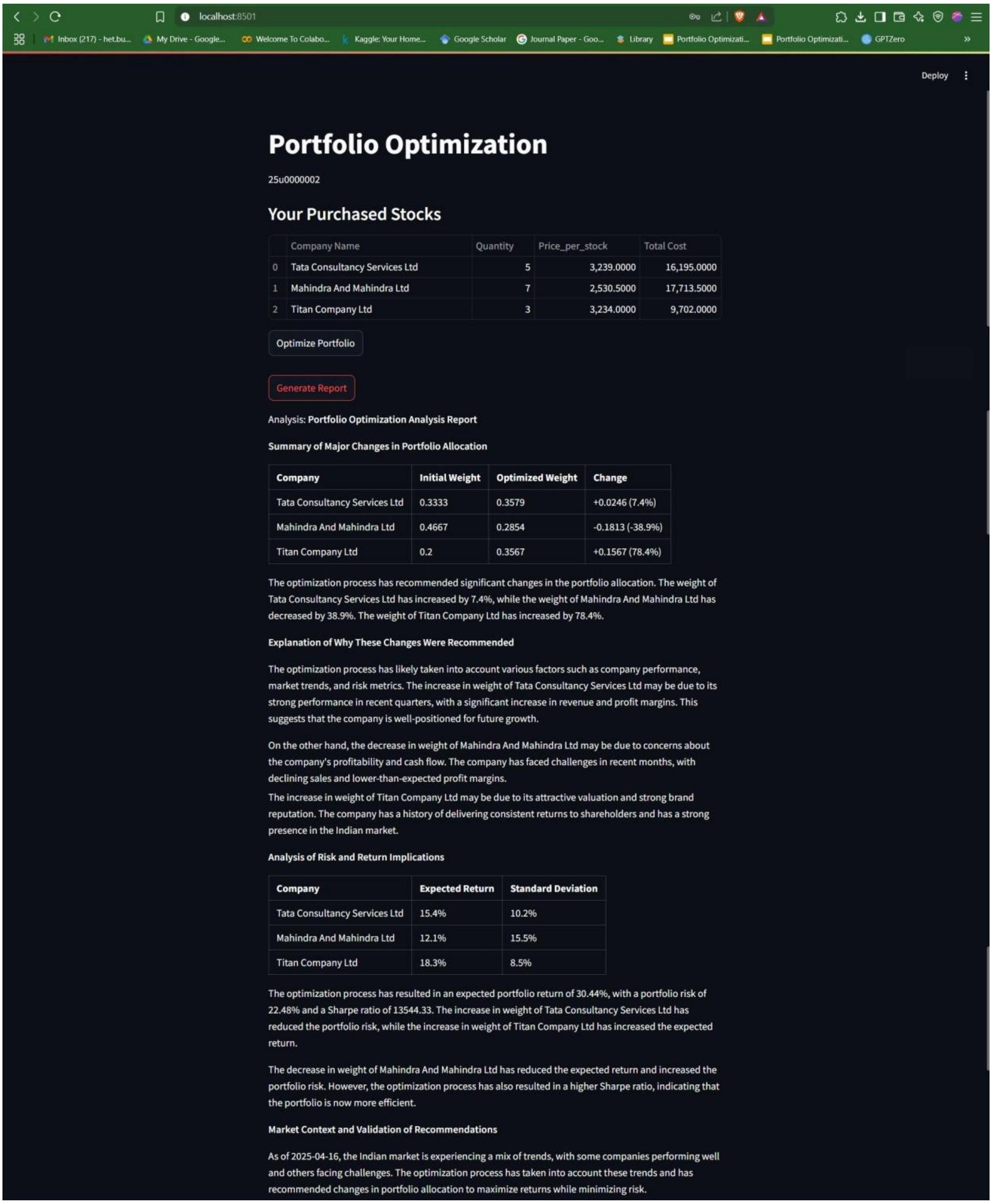

Fig. 4.25 Optimization Report of the portfolio

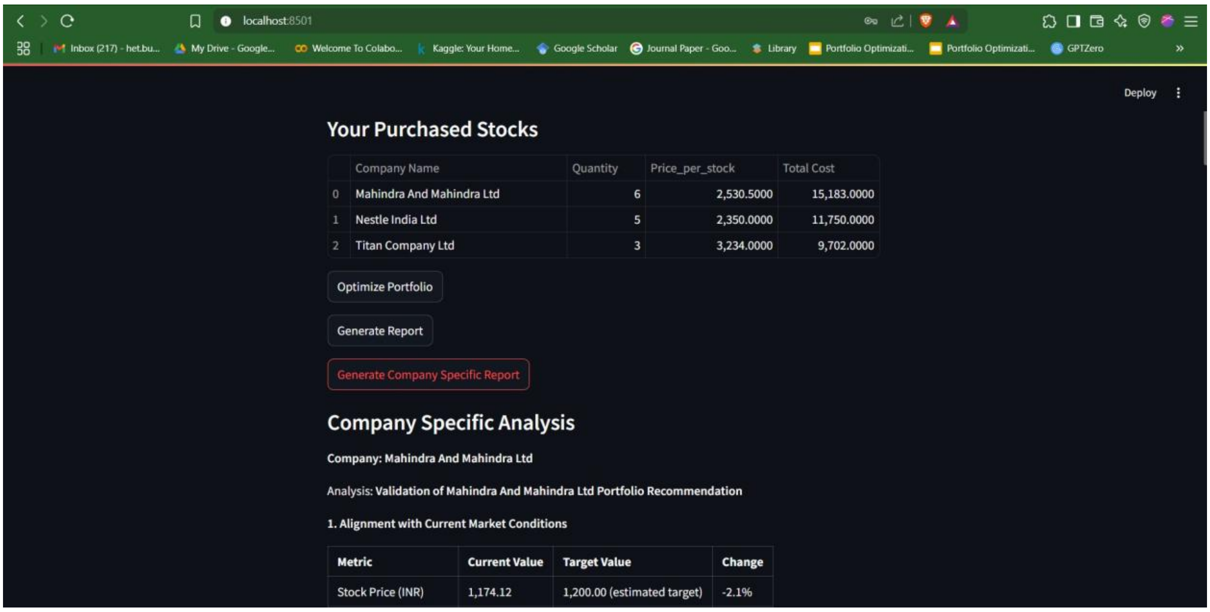

Fig. 4.26 Company Specific Analysis

Fig. 4.27 shows the list of stocks added/removed by the manager. Fig. 4.28 shows the Manager Dashboard. It consists of 2 analytics of the website: Increase in users and Purchases over time. Fig. 4.29 shows the page where the manager can add new stocks and define its price. Fig. 4.30 shows all the registered users in the website along with its personal information. Manager can block the users from logging in if anything suspicious is found.

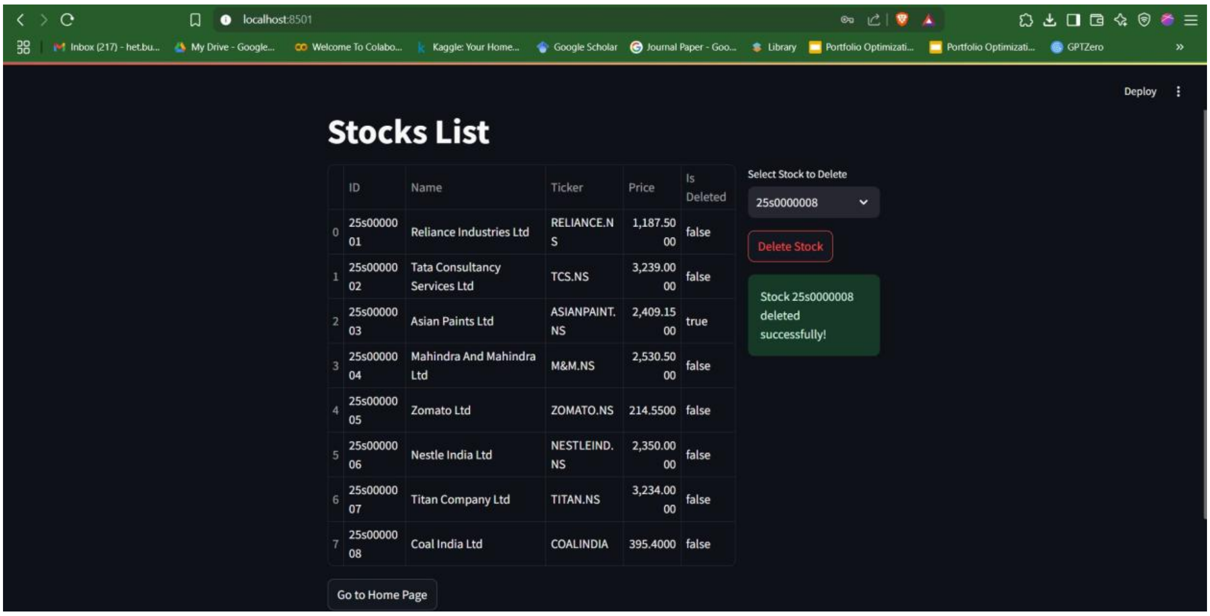

Fig. 4.27 View All Stocks

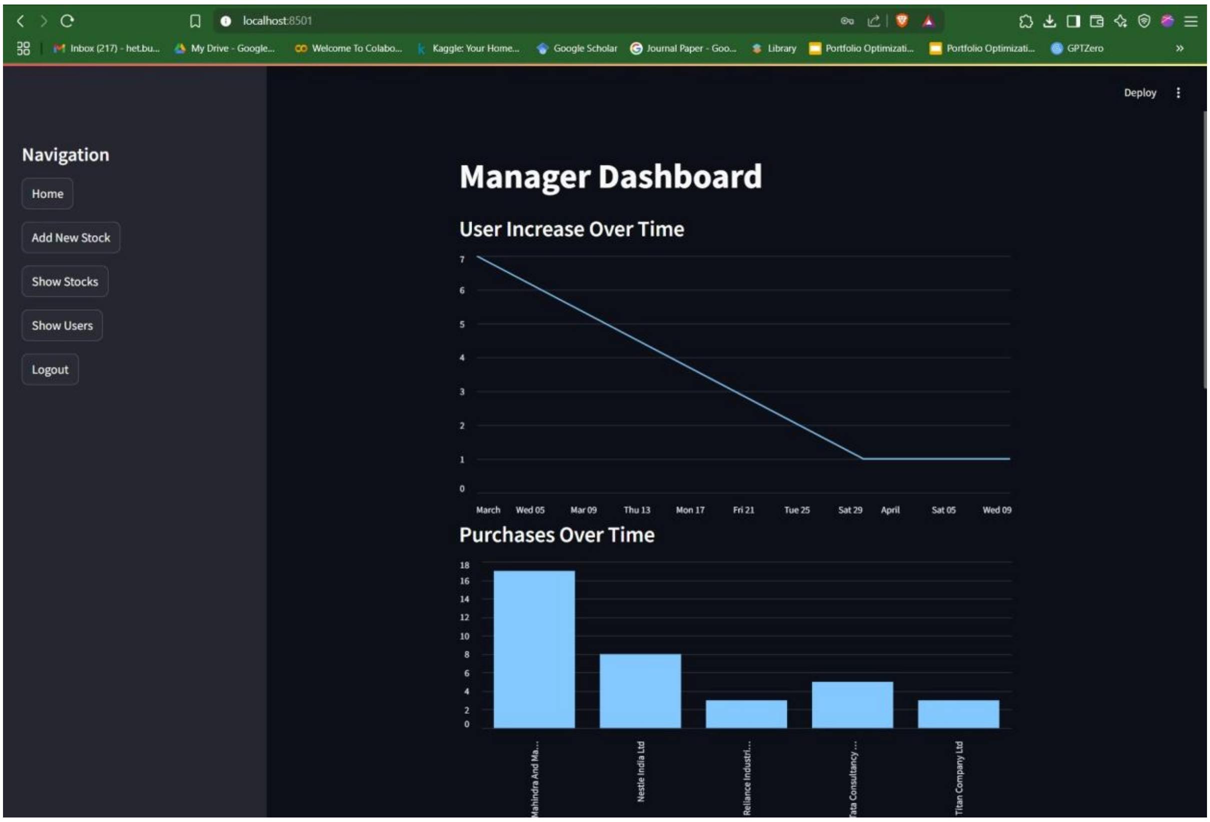

Fig. 4.28 Manager Dashboard

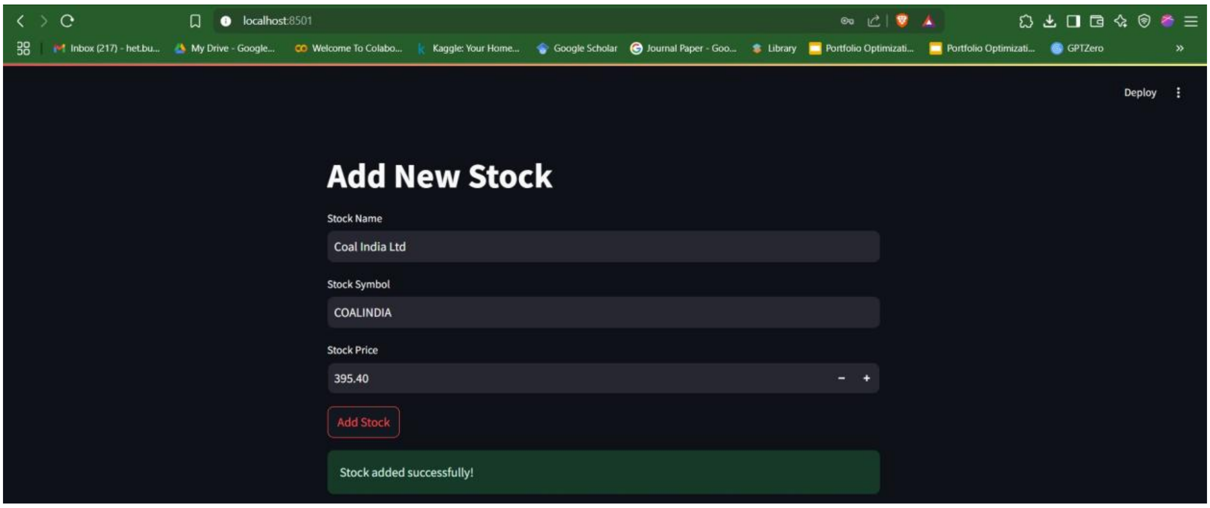

Fig. 4.29 Add new stock

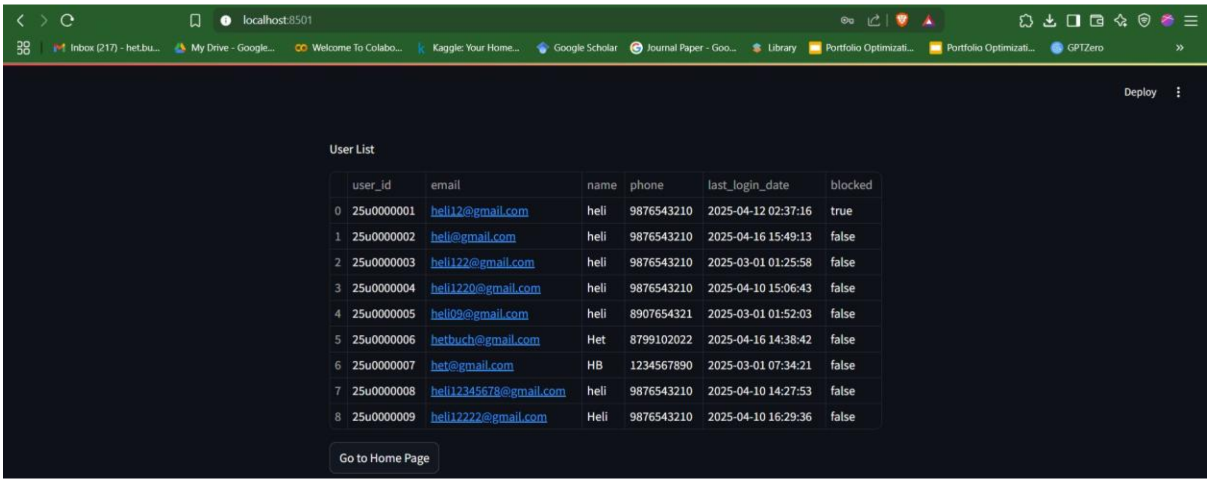

Fig. 4.30 List of all registered users

The database consists of three collections: Users, Purchases and Stocks as shown in Fig. 4.31. Each collection consists of various documents holding different types of data in JSON format. Fig. 4.32 shows a sample of document holding user information. The User document is nested in three different documents i.e. Address, Login Info and Personal Details to store and filter different types of information collectively.

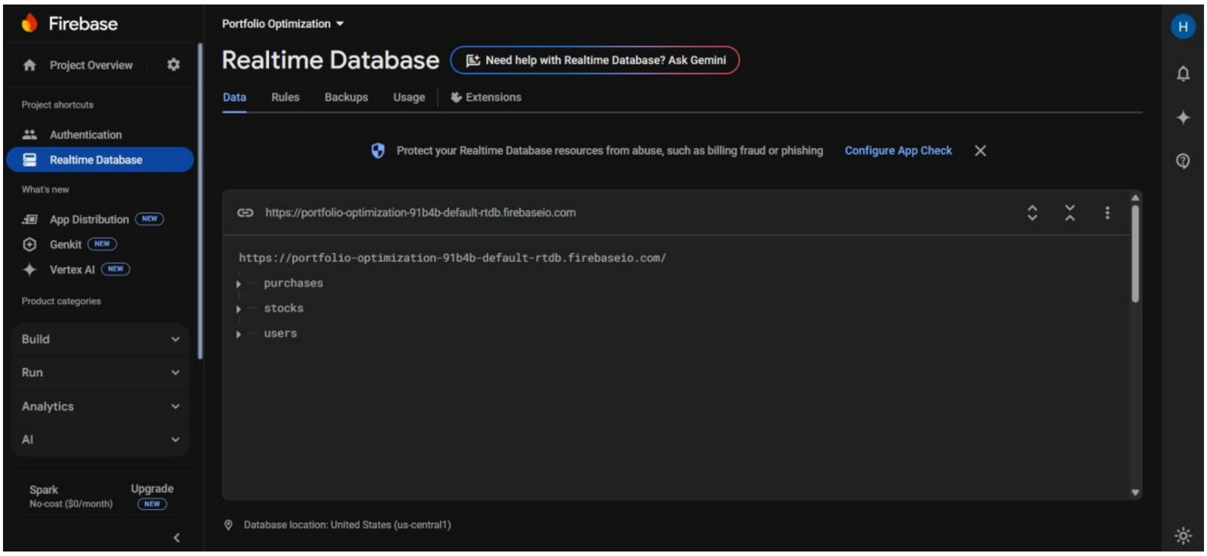

Fig. 4.31 Image showing different collections in the database

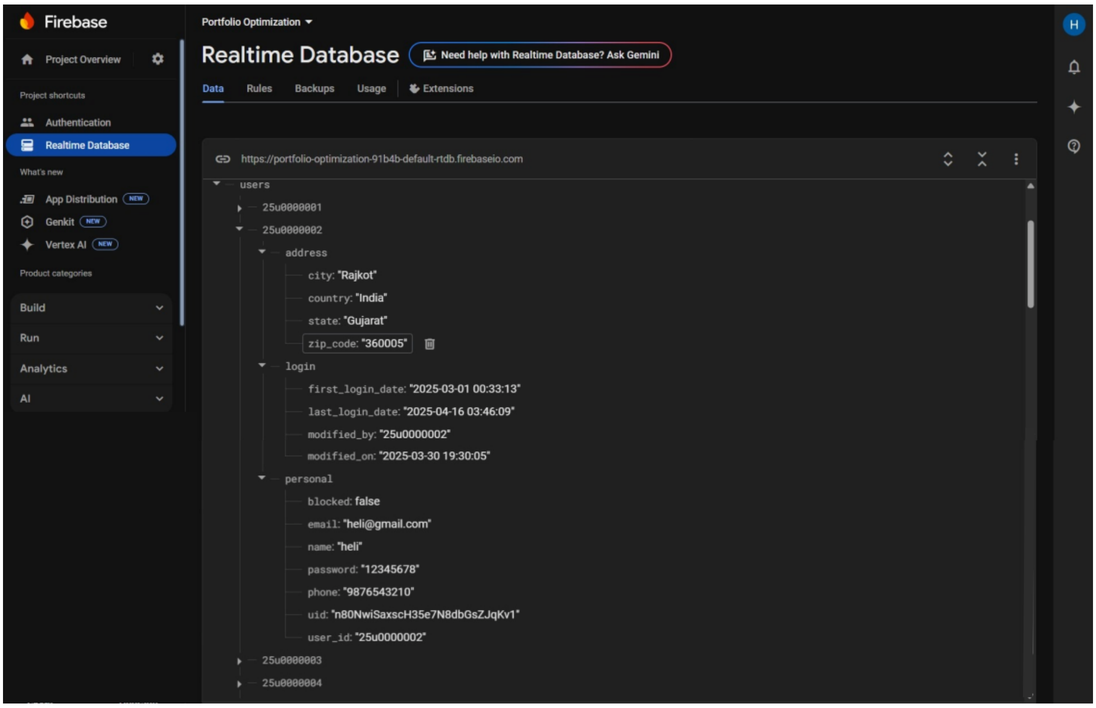

Fig. 4.32 Sample of User Collection

Fig. 4.33 shows the sample of the stocks collection. It consists of various information like who added the stock, timestamps when it was added, which company, ticker, Stock ID, etc. It is basically managed by the database manager and is not accessible to the user. The edition in stocks is done by manager which is visible to the user in buy page whenever a stock is added or removed. Fig. 4.34 shows the sample of purchase collection. It is used to the purchases done by the users. It stores information like which stock was bought, price of the stock, which User ID bought the stock, quantity, was sold or not, etc. Fig. 4.35 shows the stock that was bought was also sold.

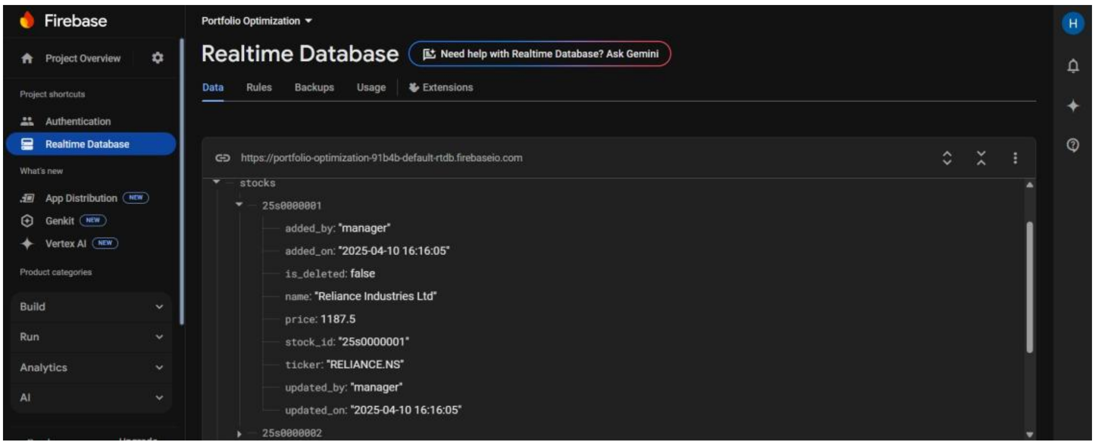

Fig. 4.33 Sample of Stocks Collection

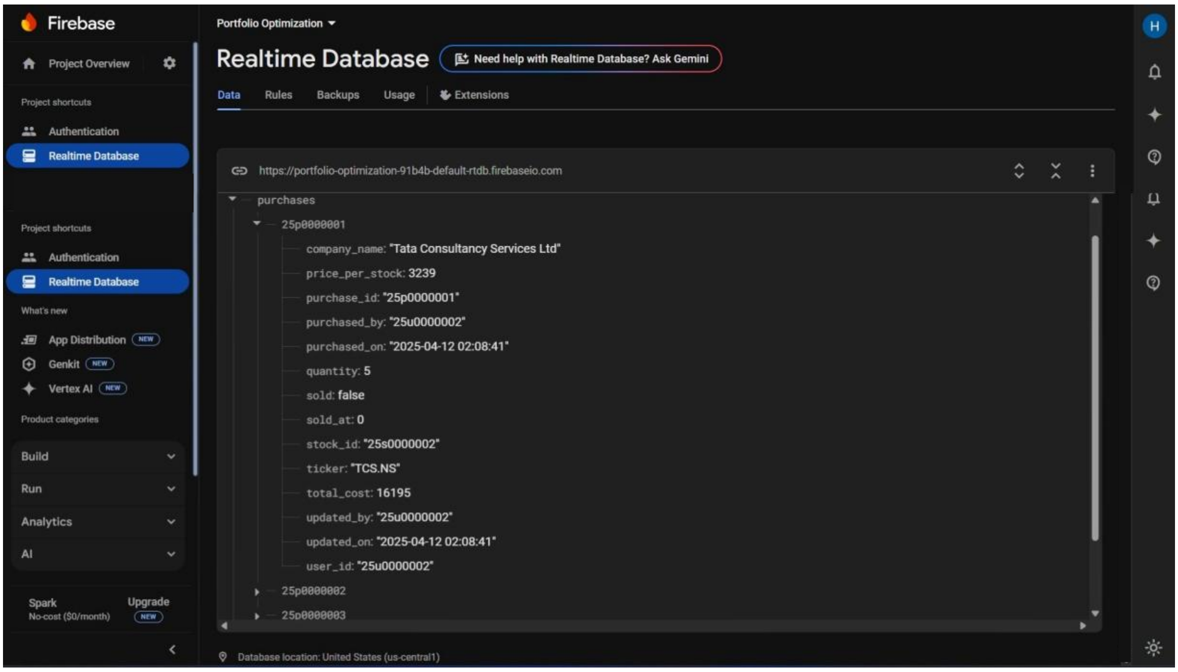

Fig. 4.34 Sample of Purchase Collection

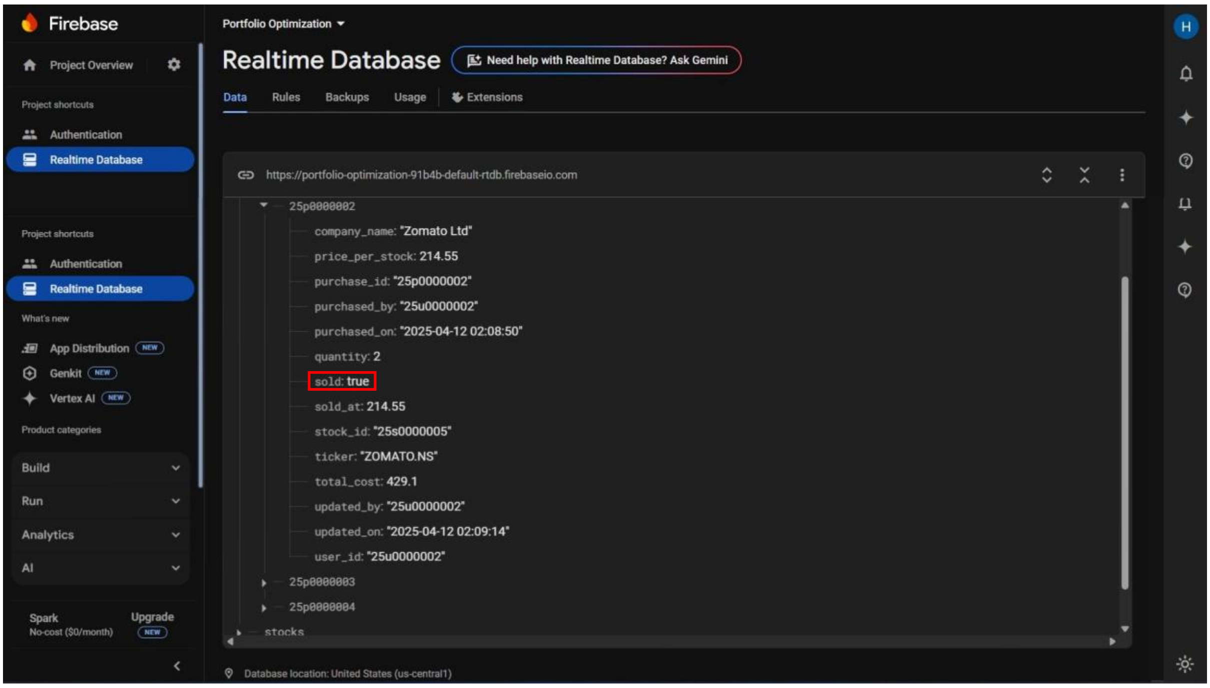

Fig. 4.35 Sample of Purchase Collection – Stock sold

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Overall Analysis of Project Viabilities

This project presents a novel approach in portfolio optimization. It was implemented using open-source tools and platforms, which eliminated financial constraints. Technologies like Firebase, Streamlit, PyCaret, and Groq allowed seamless integration of real-time data processing, machine learning, and Agentic AI. Its ability to provide data-driven investment recommendations to individual users makes it highly relevant in today's market. The system is designed to handle real-time financial data, ensuring accurate predictions even in volatile market conditions.

5.2 Problem Encountered and Possible Solutions

Some minor problems were encountered during the project especially related to computational limitations. Training machine learning models and optimization algorithms was time-consuming due to limited hardware. This was addressed by using platforms like Google Colab with GPU acceleration. Integrating multiple components into a single Streamlit web-app required significant coding and modifications in the interfaces which was collaboratively handled using meetings and regular testing of the web-app. Also, while data fetching, headers were changed in API from the provider which in turn stopped the data availability. Though, some tweaks in the code and small changes helped in restarting the execution stages.

5.3 Summary of Project

In this project, our aim was to build an AI-based portfolio optimization solution that combines real-time data, machine learning, and Agentic AI to help users in making data-driven financial decisions. It starts by collecting historical stock data using yFinance and extracting financial news and social media sentiment through tools like BeautifulSoup and NewsAPI. The collected data is further enhanced using technical indicators calculated using TA-Lib, and sentiment scores are calculated using NLP models such as VADER. Machine learning models are trained and evaluated to predict future stock prices. The sentiment scores are calculated, and fed into a nature-inspired optimization algorithm to generate a portfolio specific to the user's profile. The project uses two agents: Market Research Agent, which analyses trends, risks, and news to assess market conditions, and the Portfolio Optimization Agent, which recommends adjustments and validates portfolio changes based on user's portfolio and current market situations. All these components are integrated into a responsive and interactive web application built using Streamlit, with Firebase used for user authentication and data storage. This project helps users to get personalized investment insights in real-time.

5.4 Limitations

Despite many advantages, the project has few limitations. It does not execute trades automatically; users must manually act on recommendations. The reliance on third-party APIs also means data availability are subject to external sources. Small changes in these APIs might halt the data fetching thereby interruptions in the application. Also, the user interface is simple, it does provide a proper user experience. The recommendations are provided in a user-friendly way (more technical details which are hard to understand to new users) which could hinder the user experience. Also, the web-app does not use any dummy wallet or payment services, which might make it confusing for users to understand its applications in real-time.

5.5 Future Enhancements

To overcome the limitations and improve user experience, dependency on third-party APIs, error handling and fallback mechanisms can be introduced to ensure uninterrupted data flow even when changes in API occur. The user interface can be redesigned with a more modern layouts and simplified visual elements to improve accessibility for non-technical users. Additionally, recommendations can include simple summaries with technical insights, making it easy to understand. Finally, implementing a wallet or transaction service can help users visualize how the portfolio would perform in real-time, thereby increasing usability for learning and practice purposes remains the future work.

References

[1] A. Bathia and J. Padhy, “Deep learning and machine learning models enabled mean-variance portfolio optimisation in the Indian stock market.”

[2] A. Mosazadeh, J. Ramezani, M. Aliakbari, M. S. Geraiely, and R. Rezaeian, “Evaluation and comparison of portfolio optimization with the degree of stock risk adjustment based on the performance measurement model based on the hybrid metaheuristic algorithm and gray wolf optimization algorithm,” *Int. J. Nonlinear Anal. Appl.*, vol. 15, no. 6, pp. 63–70, 2024.

[3] A. Sebastian and V. Tantia, “Deep learning for stock price prediction and portfolio optimization,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 9, 2024.

[4] A. Uddin, M. A. H. Pabel, M. I. Alam, F. Kamruzzaman, M. S. U. Haque, M. M. Hosen, A. Sajal, M. R. Miah, and S. K. Ghosh, “Advancing financial risk prediction and portfolio optimization using machine learning techniques,” *Am. J. Manage. Econ. Innov.*, vol. 7, no. 1, pp. 5–20, 2025.

[5] F. Jahaniana, A. Mohammadia, S. A. P. Oskooeb, and A. Mottaghia, “Portfolio optimization using gray wolf algorithm and modified Markowitz model based on CO-GARCH modeling,” *Adv. Math. Finance Appl.*, vol. 9, no. 1, pp. 305–319, 2024.

[6] F. Jeribi, R. J. Martin, R. Mittal, H. Jari, A. H. Alhazmi, V. Malik, S. L. Swapna, S. B. Goyal, M. Kumar, and S. V. Singh, “A deep learning-based expert framework for portfolio prediction and forecasting,” *IEEE Access*, 2024.

[7] J. Das, S. Bowala, R. Thulasiram, and A. Thavaneswaran, “Hybrid data-driven and deep learning-based portfolio optimization,” *J. Math. Finance*, vol. 14, pp. 271–310, 2024.

[8] J. E. Silva, L. P. G. Santos, S. C. de Freitas, C. V. de O. Carvalho Junior, and R. de M. Santos, “Optimizing investment portfolios in the Brazilian market with integrated neural networks: CNN, LSTM, and GRU.

[9] J. F. G. Mejía, P. E. L. Margulis, E. E. G. Gutiérrez, Y. M. Garduño, A. A. F. Fuentes, and L. L. L. Martínez, “Design of an investment portfolio through machine learning and swarm intelligence,” *Rev. Aristas*, vol. 11, no. 19, pp. 79–84, 2024.

[10] J. Iqbal, A. Mehmood, A. Mirza, and A. Khaliq, “Asset allocation through grey wolf optimization: A case of KSE-30 index,” *Pak. J. Humanit. Soc. Sci.*, vol. 11, no. 1, pp. 670–681, 2023.

[11] K. Kubo and K. Nakagawa, “Portfolio optimization using deep learning with risk aversion utility function,” *Finance Res. Lett.*, vol. 74, p. 106761, 2025.

[12] M. Ashrafzadeh, H. M. Taheri, M. Gharehgozlou, and S. H. Zolfani, “Clustering-based return prediction model for stock pre-selection in portfolio optimization using PSO-CNN+ MVF,” *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 9, p. 101737, 2023.

[13] M. Faheem, M. Aslam, and S. Kakolu, “Artificial intelligence in investment portfolio optimization: A comparative study of machine learning algorithms,” *Int. J. Sci. Res. Arch.*, vol. 6, no. 1, pp. 335–342, 2022.

[14] M. Lv, J. Wang, S. Wang, J. Gao, and H. Guo, “Developing a hybrid system for stock selection and portfolio optimization with many-objective optimization based on deep learning and improved NSGA-III,” *Inf. Sci.*, vol. 670, p. 120549, 2024.

[15] P. Singh, M. Jha, M. Sharaf, M. A. El-Meligy, and T. R. Gadekallu, “Harnessing a hybrid CNN-LSTM model for portfolio performance: A case study on stock selection and optimization,” *IEEE Access*, vol. 11, pp. 104000–104015, 2023.

[16] R. A. Putra and E. Nurmawati, “Prediction-based stock portfolio optimization using bidirectional long short-term memory (BiLSTM) and LSTM,” *Sci. J. Informatics*, vol. 11, no. 3, pp. 609–620, 2024.

[17] R. Wang, “Two-stage portfolio optimization model based on ensemble learning and genetic algorithm,” *Available at SSRN 5098159*.

[18] S. S. Bagalkot and N. Naik, “Novel grey wolf optimizer-based parameters selection for GARCH and ARIMA models for stock price prediction,” *PeerJ Comput. Sci.*, vol. 10, p. e1735, 2024.

[19] Y. Hwang, Y. Kong, S. Zohren, and Y. Lee, “Decision-informed neural networks with large language model integration for portfolio optimization,” *arXiv preprint arXiv:2502.00828, 2025*.

[20] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, “Quantitative stock portfolio optimization by multi-task learning risk and return,” *Inf. Fusion*, vol. 104, p. 102165, 2024.

Regular Report Diary

Marwadi University
Marwadi Chandarana Group
NAAC A+

Department of Computer Engineering

Marwadi University

Academic Year : 2024-25

Semester : 8

Major Project-II (01CE0807)

Weekly Progress Report Diary (Project)

Team ID : 8CE_P009

Project Title : Agentic AI for Smart Portfolio Management : A Fusion of ML and Nature-Inspired Algorithms.

Sr. No.	Student Full Name	Student En. No.	Class
1	Heli. Hathi	92100103341	8-TC6
2	Het. Buch	92100103196	8-TC6
3			

Internal Guide Name: Prof. Ravikumar Natarajan

1

Marwadi University
Marwadi Chandarana Group

NAAC A+

FACULTY OF ENGINEERING & TECHNOLOGY
Department of Computer Engineering
Progress Report Diary- 01CE0807 - Major Project-II
A.Y. 2024-25

Weekly Project Progress Report Diary – January

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	→ Find data sources for finance and news data. (collect Raw data).	Dataset finalized.	7/1/25 9:15	
2	→ Perform Data pre processing on raw data. → Start Literature Review.	preprocessing done. Download 15 papers + Review it.	14/1/25	
3	→ Perform feature engineering, prepare workflow. → Continue Literature Review.	Done.	24/1/25	
4	→ Complete Literature Review. → Perform model comparision.	LR Completed. Prepare result table for ML algorithms.	27/1/25	

1

Marwadi University
Marwadi Chandarana Group

NAAC A+

FACULTY OF ENGINEERING & TECHNOLOGY
Department of Computer Engineering
Progress Report Diary- 01CE0807 - Major Project-II
A.Y. 2024-25

Weekly Project Report Diary – FEBRUARY

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	-> Done Model Comparison. -> Learn Nature Inspired optimiza-tion Techniques.	Explore optimization algorithms. Check how to use it for parameter tuning.	7/2/25	Ry
2	-> Optimize the model. -> Continue Research on Optimization Techniques.	Needs more Experimentation.	14/2/25	Ry
3	-> Connect database through Firebase. (Perform operation in database).	Done.	17/2/25	Ry
4	-> Make ppt for Review-1 and complete it.	prepare PPT & Send it to me.	24/2/25	Ry

Marwadi University
Marwadi Chandarana Group

NAAC A+

FACULTY OF ENGINEERING & TECHNOLOGY
Department of Computer Engineering
Progress Report Diary- 01CE0807 - Major Project-II
A.Y. 2024-25

Weekly Project Report Diary – MARCH

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	→ Learn different Nature- Inspire Optimization Techniques.	Explored NIO algorithms	2/3/25	Ry
2	→ Apply PSO, GWO, BAT Algorithm individually.	PSO ~~provides~~ Good results. try rest	7/3/25	Ry
3	→ Apply ensemble and Hybrid combination of Algorithm. → Learn about Agentic AI.	Hybrid algorithms done.	14/3/25	Ry
4	→ Apply Agentic AI concepts. → Complete Backend firebase connection.	Backend Connection done	23/3/25	Ry.

3

Marwadi University
Marwadi Chandarana Group
NAAC A+

FACULTY OF ENGINEERING & TECHNOLOGY
Department of Computer Engineering
Progress Report Diary- 01CE0807 - Major Project-II
A.Y. 2024-25

Weekly Project Report Diary – APRIL

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	→ Make webapp using Streamlit. → Integrate Backend and Frontend.	Web app done.	1/4/25	Ry.
2	→ Make PPT for Review-2. → Ch-1 to 3 of Report Completed.	Start report & PPT.	10/4/25	Ry
3	→ complete Report.	Done. Start Journal Paper Work	19/4/25	Ry
4	→ Start working on Journal writing.	Complete Paper work. & Patent form.	21/4/25	Ry.

4

Consent Letter

Marwadi University
Marwadi Chandarana Group
NAAC A+

FACULTY OF TECHNOLOGY
Department of Computer Engineering
Consent letter for Patent
A.Y. 2024-25

Consent for Filing Patent/Research Publication Application

We, **Prof. Ravikumar R. Natarajan**, **Heli Hathi, Het Buch** hereby give our full consent and authorization for the filing of a patent/research publication application for the project titled **"Agentic AI for Smart Portfolio Management: A Fusion of ML and Nature Inspired Algorithms**".

We hereby authorize Marwadi University and/or its legal representatives to file the patent/research publication application and act on our behalf regarding any matters related to this filing.

Date: 17/04/2025
Name: Prof. Ravikumar R. Natarajan
Signature:

Date: 17/04/2025
Name: Heli Hathi
Signature: Heli

Date: 17/04/2025
Name: Het Buch
Signature: H. J. Buch