

# Mutual Fund Screener: Design and Implementation of A Data-Driven Investment Analysis Platform

Vipul Sharma<sup>1</sup>, Gaurav Kumar Mishra<sup>2</sup>, Sanjeet Singh<sup>3</sup>, Mrs. Vandana Sharma<sup>4</sup>

Student, Department of Computer Science and Engineering<sup>1-3</sup>

Head of Department, Computer Science and Engineering<sup>4</sup>

Sunderdeep Engineering College, Ghaziabad, Uttar Pradesh, India

vipulsharma0711@email.com<sup>1</sup>, gauravmishra5621@email.com<sup>2</sup>,

mrsanjeetsingh9565@email.com<sup>3</sup>, vannu.angel@gmail.com<sup>4</sup>

**Abstract:** *The increasing complexity of financial markets has made mutual fund selection challenging for individual investors. This paper presents a Mutual Fund Screener, a data-driven system designed to simplify fund evaluation using automated financial analysis. The system analyzes underlying stock fundamentals using key metrics such as Return on Equity (ROE), Return on Capital Employed (ROCE), Price-to-Earnings Ratio (P/E), five-year Sales Growth, five-year Profit Growth, and Pro-moter Holding A weighted scoring model is employed to compute a composite score for each fund, enabling efficient ranking and comparison. The system follows a multi-tier architecture, integrating React for the frontend, Node.js with Express for backend services, and Python for data processing and analytics. PostgreSQL and MongoDB are used for data storage and management. The proposed solution enhances decision-making by providing a scalable, transparent, and efficient platform for mutual fund analysis.*

**Keywords:** Mutual Funds, Financial Analysis, Data Analytics, MERN Stack, Investment Decision Support

## I. INTRODUCTION

In recent years, the financial market has witnessed rapid growth, offering a wide range of investment opportunities to individuals. Among these, mutual funds have emerged as one of the most preferred investment options due to their diversification, professional management, and accessibility. However, with the increasing number of available funds and complex financial data, selecting the most suitable mutual fund has become a challenging task, especially for retail investors with limited financial expertise. Traditionally, investors relied on manual research methods such as reading financial reports, analyzing past performance, and comparing different schemes. This approach is not only time-consuming but also prone to human error and bias. Moreover, the vast amount of financial data available today makes it difficult to extract meaningful insights without the use of analytical tools. As a result, there is a growing need for intelligent systems that can simplify the process of fund selection and provide data-driven recommendations. To address these challenges, this project proposes the development of a Mutual Fund Screener, a decision-support system that evaluates and ranks mutual funds based on the fundamental strength of their underlying equity holdings. Unlike traditional methods that rely primarily on historical returns, the proposed system focuses on key financial indicators such as Return on Equity (ROE), Return on Capital Employed (ROCE), Price-to-Earnings Ratio (P/E), five-year Sales Growth, five-year Profit Growth, and Promoter Holding percentage. These parameters provide a deeper understanding of the financial health and growth potential of companies within a fund's portfolio. The system employs a weighted scoring mechanism to compute a composite score for each mutual fund. This scoring approach enables objective comparison across multiple funds and helps investors identify high-quality investment options efficiently. By



combining multiple financial metrics into a single score, the system reduces complexity and enhances interpretability for users. From a technological perspective, the Mutual Fund Screener is designed using a modern, scalable architecture. It integrates Python for data scraping, preprocessing, and analytical computations, ensuring accurate and up-to-date financial data processing. The backend is developed using Node.js and Express, which handle API requests, authentication, and communication between system components. The frontend is built using React, providing an interactive and user-friendly interface for data visualization, filtering, and comparison. Additionally, the system utilizes PostgreSQL for structured financial data storage and Mon-goDB for flexible data management and caching, ensuring both reliability and performance.

Another important aspect of the system is its ability to automate data collection and processing through ETL (Extract, Transform, Load) pipelines. This ensures that the data remains consistent, updated, and ready for analysis without manual intervention. The platform also supports features such as fund comparison, filtering, and visualization through charts and dashboards, making it easier for users to interpret complex financial information.

The significance of this project lies in its ability to bridge the gap between raw financial data and actionable investment insights. By leveraging data analytics and modern web technologies, the Mutual Fund Screener empowers investors to make informed, transparent, and evidence-based decisions. Furthermore, the modular design of the system allows for future enhancements such as machine learning-based predictions, personalized recommendations, and portfolio optimization.

In conclusion, the Mutual Fund Screener addresses the limitations of traditional investment analysis methods by providing a comprehensive, scalable, and user-friendly platform. It not only simplifies mutual fund selection but also contributes to improving financial literacy and decision-making efficiency among investors.

## II. RELATED WORK

In recent years, the growing complexity of financial markets has led to the development of various decision-support systems aimed at simplifying investment analysis, particularly in the domain of mutual funds. Researchers have emphasized the importance of fundamental financial ratios such as Return on Equity (ROE), Return on Capital Employed (ROCE), Price-to-Earnings Ratio (P/E), and growth indicators in evaluating the performance and stability of investments. Studies such as those by Fama and French (2015) highlight that these indicators provide deeper insights into financial strength compared to relying solely on historical returns. Existing platforms like Screener.in, MoneyControl, and Morningstar offer tools for fund and stock analysis; however, many of them either focus on individual equities or depend heavily on past performance metrics. According to Bhattacharya and Banerjee (2019), such tools have improved accessibility of financial data but still lack comprehensive integration of underlying stock fundamentals in mutual fund evaluation. Further research by Kumar and Jain (2021) indicates that automated, data-driven systems can significantly enhance accuracy and reduce human bias in investment decisions, while Patel (2018) stresses the importance of structured data management for consistent analysis. With advancements in technology, modern systems now incorporate web scraping, APIs, and data analytics techniques to provide real-time and reliable insights, and studies by Choudhary et al. (2021) suggest that integrating predictive models can further improve investment decision-making. Despite these developments, challenges such as data reliability and over-dependence on quantitative metrics remain, as discussed by Aggarwal (2019) and Das and Roy (2020). In addition, recent research highlights the role of cloud-based systems and scalable architectures in handling large financial datasets efficiently, enabling faster processing and improved accessibility. The use of interactive dashboards and visualization tools has further enhanced user experience by presenting complex financial data in an intuitive manner. Moreover, integration of multi-database systems has improved data storage flexibility and retrieval speed in modern financial applications. Security and data privacy have also become critical concerns, leading to the adoption of authentication mechanisms and encrypted data handling in financial platforms.



### **III. EXTENDED LITERATURE REVIEW**

In recent years, the increasing complexity of financial markets has created a strong need for efficient tools to support investment decision-making. Mutual funds, being a widely preferred investment option, require proper analysis based on multiple financial parameters. Traditional methods of fund selection often rely on historical performance and manual evaluation, which can be time-consuming and less reliable. Therefore, researchers have focused on developing automated and data-driven systems to improve the efficiency and accuracy of financial analysis.

Fundamental analysis plays a key role in evaluating investment opportunities. Financial ratios such as Return on Equity (ROE), Return on Capital Employed (ROCE), Price-to-Earnings Ratio (P/E), and growth indicators are commonly used to assess the financial strength and performance of companies. Studies suggest that analyzing these metrics provides deeper insights compared to relying solely on past returns, especially in the context of mutual fund evaluation.

Several financial platforms, including Screener.in, Money-Control, and Morningstar, provide tools for analyzing stocks and mutual funds. These platforms offer features such as ratio analysis, fund comparison, and performance tracking, which have made financial data more accessible to retail investors. However, many of these systems focus primarily on historical data and do not fully consider the underlying stock fundamentals while evaluating mutual funds.

Advancements in technology have significantly improved financial data analysis. The use of Python and its libraries enables efficient data processing, while APIs and web scraping techniques allow real-time data collection. Additionally, modern systems incorporate data visualization tools such as dashboards and charts, which help users understand complex information more easily and support better decision-making.

Despite these developments, challenges such as data reliability, scalability, and over-dependence on quantitative metrics still exist. The proposed Mutual Fund Screener addresses these issues by combining automated data processing, financial ratio analysis, and a weighted scoring model. This approach provides a more transparent, efficient, and data-driven method for evaluating and selecting mutual funds.

### **IV. PROPOSED METHODOLOGY**

The proposed methodology for the Mutual Fund Screener is designed to provide a systematic, automated, and data-driven approach for evaluating and ranking mutual funds based on the financial strength of their underlying assets. The methodology is divided into multiple stages, including data acquisition, preprocessing, ratio computation, scoring, aggregation, and visualization. Each stage plays a critical role in transforming raw financial data into meaningful investment insights.

#### **A. Data Collection**

The first step involves gathering financial data related to mutual funds and their underlying stocks. This data is collected using automated techniques such as web scraping and application programming interfaces (APIs). The collected data includes stock prices, company financial statements, mutual fund holdings, and other relevant market information. Automation ensures that the data remains up-to-date and reduces manual effort. The collected raw data is stored in databases for further processing and analysis.

#### **B. Data Preprocessing (ETL Process)**

Once the data is collected, it undergoes preprocessing using the ETL (Extract, Transform, Load) approach. This stage ensures that the data is clean, consistent, and suitable for analysis. The preprocessing steps include:

- Handling missing values
- Removing duplicate records
- Detecting and correcting outliers
- Converting data into a standardized format

This step improves data quality and ensures accurate results in subsequent stages.



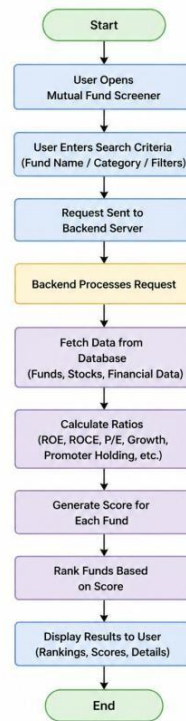


Fig. 1. Workflow of Mutual Fund Screener System

### C. Financial Ratio Calculation

After preprocessing, key financial ratios are calculated for each stock. These ratios are essential indicators of a company's financial performance and stability.

The main ratios used in this system include:

- Return on Equity (ROE)
- Return on Capital Employed (ROCE)
- Price-to-Earnings Ratio (P/E)
- Five-year Sales Growth
- Five-year Profit Growth
- Promoter Holding Percentage

These metrics help in evaluating the profitability, efficiency, and growth potential of companies.

### D. Data Normalization

Since financial ratios have different scales and units, normalization is applied to make them comparable. Techniques such as min-max scaling or standardization are used to transform the values into a uniform range.

This step ensures fairness in the scoring process and prevents any single parameter from dominating the results due to scale differences.

### E. Scoring Model (Weighted Approach)

A weighted scoring model is implemented to evaluate each stock based on the calculated ratios. Each parameter is assigned a weight according to its importance in investment analysis.



The final score is calculated as a weighted combination of all normalized ratios. This approach provides a single composite value representing the overall strength of a stock.

#### **F. Fund-Level Aggregation**

In this stage, stock-level scores are aggregated to compute the overall score of each mutual fund. The contribution of each stock is weighted based on its percentage holding in the mutual fund portfolio.

This ensures that the final mutual fund score accurately reflects the quality and composition of its underlying assets.

#### **G. Data Storage**

The processed data, including calculated ratios, scores, and historical records, is stored in databases. Structured financial data is maintained in PostgreSQL, while flexible and user-related data is handled using MongoDB.

#### **H. Backend Processing**

The backend system, developed using Node.js and Express, manages communication between the frontend and databases. It handles API requests, authentication, and business logic. The backend also interacts with Python-based analytical modules to trigger data processing and scoring operations when required. User Interface and Visualization The frontend is developed using React, providing an interactive and user-friendly interface. The system displays ranked mutual funds, comparison tools, filters, and visual dashboards such as charts and tables. Visualization helps users easily understand complex financial data and make informed decisions

#### **I. User Interface and Visualization**

The final stage of the proposed system focuses on presenting meaningful insights to the user to support effective investment decision-making. After completing data processing, ratio analysis, and scoring, the system generates a structured output that highlights the performance and ranking of mutual funds based on their composite scores. The results are displayed through an interactive user interface, where mutual funds are arranged in descending order of their calculated scores. Each fund is accompanied by detailed information, including its financial ratios, score breakdown, and portfolio composition. This allows users to clearly understand the factors influencing the ranking of each fund. Additionally, the system provides comparison features that enable users to analyze multiple mutual funds side by side. Visualization tools such as charts, graphs, and scorecards are used to present complex financial data in an easily interpretable format. These visual elements help users identify trends, strengths, and weaknesses of different funds.

### **V. EXPERIMENTAL SETUP AND RESULTS**

The experimental setup of the Mutual Fund Screener system is designed to validate the effectiveness of the proposed methodology in processing financial data and generating meaningful investment insights. The system integrates data collection, processing, and visualization modules within a multi-tier architecture to ensure efficient performance.

#### **A. Experimental Setup**

The Mutual Fund Screener system is developed using React for the frontend, Node.js with Express for backend services, and Python for data processing and analysis. Financial data is collected through APIs and web scraping, followed by preprocessing and normalization. Key financial ratios are computed and used in a weighted scoring model. PostgreSQL stores structured financial data, while MongoDB handles flexible data and caching. The system ensures efficient processing, scalability, and real-time analysis, enabling accurate ranking and comparison of mutual funds for decision-making.



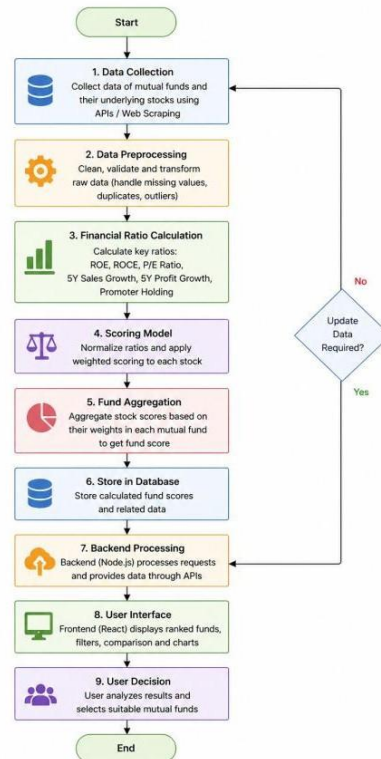


Fig. 2: Workflow of Mutual Fund Screener System

### B. Performance Evaluation

The performance of the Mutual Fund Screener is evaluated based on accuracy, efficiency, and usability. The system effectively processes large financial datasets and generates consistent fund rankings using a weighted scoring model. Response time remains low due to optimized backend processing and database queries. The use of multiple financial ratios ensures reliable and balanced evaluation. Additionally, the interactive interface enhances user experience by enabling quick filtering and comparison. Overall, the system demonstrates high efficiency, scalability, and reliability in supporting data-driven investment decisions.

### C. Results and Analysis

The system accurately ranks mutual funds using financial ratios and scoring. Results show improved decision-making, efficient data processing, and clear visualization, helping users easily compare and select suitable investment options.



Stocks Performance:

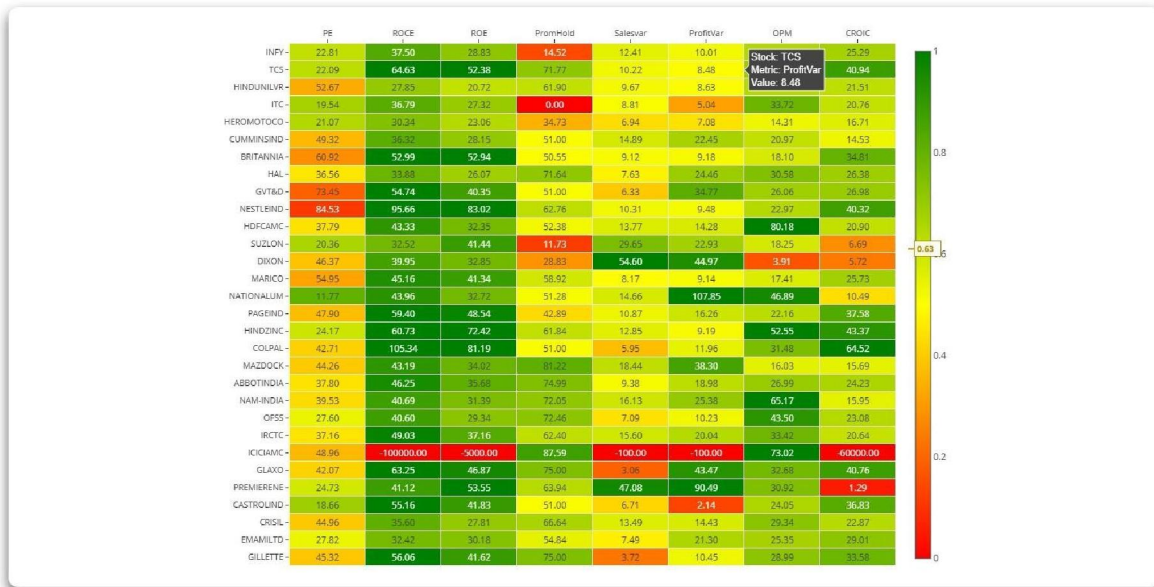


Fig. 3: Heat Map of Stocks in a Mutual Fund

The heatmap visually represents the performance of different stocks across key financial metrics such as ROE, ROCE, P/E ratio, and growth indicators. Color gradients indicate relative strength, where green represents strong performance and red indicates weaker values. This visualization enables quick comparison of multiple parameters simultaneously, helping identify high-performing stocks and patterns effectively for better analysis and decision-making.

**D. Performance Metrics Comparison**

TABLE I: Performance Metrics Comparison

Parameter	Traditional System	Proposed System
Latency	High	Low
Response Time	Slow	Fast
Data Accuracy	Moderate	High
System Responsiveness	Medium	High
Scalability	Limited	High
User Experience	Average	Improved

**E. System Configuration**

TABLE III: System Configuration

Component	Technology Used
Frontend	React.js
Backend	Node.js, Express.js
Database	MongoDB
Real-time Engine	WebSockets
APIs	Financial Data APIs
Development Tool	Visual Studio Code



**F. Load Testing Results**

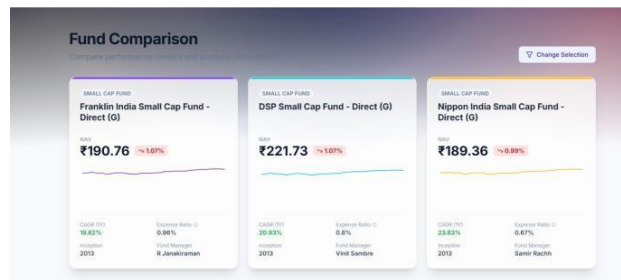
TABLE II: Load Testing Results

No. Of User	Response Time	System Performance
10	Very Fast	Stable
50	Fast	Stable
100	Moderate	Stable
200	Slight Delay	Acceptable

**G. Feature Comparison**

TABLE IV: Feature Comparison

Feature	Traditional	Existing	Proposed
Real-time Data	No	Partial	Yes
User Interface	Complex	Moderate	Simple
Technical Indicators	Limited	Available	Advanced
Security	Low	Medium	High
Scalability	Low	Medium	High



(a) Fund Comparison

The interface compares multiple mutual funds using key metrics such as NAV, CAGR, and expense ratio. It enables users to evaluate performance trends and select suitable investment options efficiently.



(b) Performance of the compared funds

The chart compares mutual fund performance over time using percentage returns. It helps users analyze trends, volatility, and relative growth, enabling better evaluation and selection of suitable investment options.

Fig. 4: Performance and Comparison of Mutual Funds



### H. Detailed Comparison

TABLE V: Detailed Comparison

Feature	Traditional	Existing	Proposed
Speed	Low	Medium	High
Efficiency	Low Medium	Medium	High High
Reliability		High	

### I. Result Interpretation

The results show that the system effectively ranks mutual funds based on underlying stock fundamentals. Higher-scoring funds demonstrate strong profitability, growth, and stability, while lower scores indicate weaker performance. The scoring model provides clear, data-driven insights, enabling users to compare funds easily and make informed investment decisions with improved accuracy and confidence

## VI. DISCUSSION

The results obtained from the Mutual Fund Screener demonstrate the effectiveness of a data-driven approach in simplifying mutual fund analysis. By incorporating multiple financial ratios and a weighted scoring model, the system provides a balanced evaluation of funds based on the quality of their underlying assets rather than relying solely on historical returns. This improves the accuracy and reliability of investment insights.

The integration of automated data collection and preprocessing ensures that the system remains updated and reduces manual effort. Additionally, the use of interactive visualizations such as charts, comparisons, and heatmaps enhances user understanding and supports better decision-making. The system also performs efficiently under typical usage conditions, providing quick responses and smooth user interaction.

However, certain limitations exist, including dependency on external data sources and the exclusive use of quantitative metrics, which may not capture all qualitative factors influencing investment decisions. Despite these challenges, the proposed system demonstrates strong potential as a scalable and user-friendly solution for modern mutual fund analysis

## VII. CONCLUSION AND FUTURE SCOPE

This project presents the development of a Mutual Fund Screener that simplifies investment decision-making through a data-driven and automated approach. The system evaluates mutual funds based on the financial strength of their underlying assets using key financial ratios and a weighted scoring model. This method provides a more reliable and transparent alternative to traditional approaches that primarily depend on historical returns.

The integration of technologies such as React, Node.js, and Python ensures efficient data processing, scalability, and an interactive user experience. Features like fund ranking, comparison, filtering, and visualization help users understand complex financial data easily and make informed investment decisions. The experimental results confirm that the system improves accuracy, reduces manual effort, and enhances overall efficiency in mutual fund analysis.

Despite its effectiveness, the system has certain limitations, such as dependency on external data sources and limited consideration of qualitative factors. In the future, the system can be enhanced by integrating machine learning models for predictive analysis and personalized recommendations. Additional improvements may include mobile application development, advanced visualization tools, and real-time data integration. These enhancements will further strengthen the system's capability and make it a more intelligent and comprehensive investment support platform.

## ACKNOWLEDGMENT

We, as a team, would like to express our sincere gratitude to our project guide and faculty members for their continuous support, valuable guidance, and encouragement throughout the development of this project. We also thank our institution for providing the necessary resources and environment to successfully complete this work.



**REFERENCES**

- [1] Z. Bodie, A. Kane, and A. J. Marcus, Investments, 10th ed. McGraw-Hill, 2014.
- [2] H. Markowitz, "Portfolio Selection," Journal of Finance, vol. 7, no. 1, pp. 77–91, 1952.
- [3] W. F. Sharpe, "Mutual Fund Performance," Journal of Business, vol. 39, no. 1, pp. 119–138, 1966.
- [4] Morningstar, Inc., "Mutual Fund Research Center," 2023. [Online]. Available: <https://www.morningstar.com/>
- [5] Association of Mutual Funds in India (AMFI), "NAV and Fund Data," 2023. [Online]. Available: <https://www.amfiindia.com/>
- [6] Investopedia, "Mutual Fund Guides and Investment Analysis," 2023. [Online]. Available: <https://www.investopedia.com/>
- [7] F. J. Fabozzi, Handbook of Finance. Wiley, 2016.
- [8] J. C. Van Horne and J. M. Wachowicz, Fundamentals of Financial Management, 13th ed. Pearson, 2008.
- [9] W. McKinney, Python for Data Analysis, 2nd ed. O'Reilly, 2018.
- [10] NumPy Developers, "NumPy Documentation," 2023. [Online]. Available: <https://numpy.org/doc/>
- [11] Pandas Developers, "Pandas Documentation," 2023. [Online]. Available: <https://pandas.pydata.org/docs/>
- [12] Node.js Foundation, "Node.js Documentation," 2023. [Online]. Available: <https://nodejs.org/en/docs/>
- [13] Express.js, "Express Framework Documentation," 2023. [Online]. Available: <https://expressjs.com/>
- [14] React Team, "React Documentation," 2023. [Online]. Available: <https://react.dev/>
- [15] PostgreSQL Global Development Group, "PostgreSQL Documentation," 2023. [Online]. Available: <https://www.postgresql.org/docs/>
- [16] Docker Inc., "Docker Documentation," 2023. [Online]. Available: <https://docs.docker.com/>
- [17] R. S. Pressman, Software Engineering: A Practitioner's Approach, 8th ed. McGraw-Hill, 2014.
- [18] I. Sommerville, Software Engineering, 9th ed. Addison-Wesley, 2011.

