IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Design and Development of a Real-Time-Stock Trading Application

Mahi and Honey Choudhary
Department of Computer Science and Engineering
RD Engineering College Duhai, Ghaziabad
mahi122cs0064@rdec.in and hc1781741@gmail.com

Abstract: *The development process and evolution of financial technology, commonly referred to as FinTech, have made possible the existence of stock trading platforms through sophisticated technological applications for executing trades. The research paper presents the development of a stock trading platform that allows real-time monitoring of stocks during the trading process.*
Developers have employed the use of the MERN Stack to develop the application and establish systems for financial data analysis and the creation of web-based applications.
In the architecture of the application, React is used to design and develop user interfaces while Node.js creates scalable backend servers and MongoDB makes available database storage capabilities to enable real-time stock trading. The application acquires real-time stock data using the APIs offered by Alpha Vantage, allowing low latency data delivery and accuracy in data processing. The design of the system aims at creating a secure and usable system by incorporating security measures such as authentication and data handling systems created for peak performance.
The current stock trading platforms can process more than 1000 transactions per second with their various operations running simultaneously. This is achieved by ensuring that the system runs efficiently and optimally when handling these operations [1]. The research proves that web systems require efficiency and usability coupled with real-time data processing abilities to create an efficient stock trading platform. The study highlights the need for real-time systems and a good foundation for web applications when designing modern trading platforms.
The study focuses on stock trading through the application of the MERN stack with real-time systems.

Keywords: *MERN Stack*

I. INTRODUCTION

The development of digital technology has brought about the creation of online stock exchange platforms that have revolutionized how the financial markets work. Stock exchange transactions required brokers to be involved in the transaction processes that would lead to time lags and expensive transactions and thus limited accessibility by the general public. Electronic trading systems have transformed the stock market due to their capabilities of allowing investors to conduct transactions independently.

Zerodha and Robinhood Markets because of their ease of use and inexpensive trading facilities provided to the customers. FinTech has grown in recent years as it has become possible to conduct trades via the software and analysis the market using the software system. Stock market apps run Similar to high-frequency data streams, there is a technical aspect in that latency requirements must be met. The reason why the trading platforms remain able to process more than 1000 orders in a second lies in the effectiveness of their data structure and optimization of their backend processes that facilitate trading activity. The requirement for the system to synchronize data in real time between the client and server interfaces arises due to accuracy in price and trade performance results.

This is due to the integration of full-stack technologies like React, Node.js, and MongoDB, which form a framework for building trading systems that have scalability capabilities. This system can process information using asynchronous

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 18, April 2026 Impact Factor: 8.2

operations as well as non-blocking I/O actions and utilizes several data modelling methods that allow it to manage changes in dynamic data.

The simple finance models offer an effective mathematical approach that facilitates the analysis of the variations in the share prices and the portfolio valuation. Portfolio Value (PV) can be mathematically represented as follows:

$$PV = \sum_{i=1}^{n} (\mathrm{Pi} * Qi)$$

The The equation of P formula determines the stock price for stock i by multiplying Qi, which refers to the quantity of stocks owned by the investors. Trading systems apply these formulas as vital tools, enabling them to determine the real portfolio performance.

III. PROBLEM STATEMENT

The current development of internet trading platforms is hindered by several problems which influence the creation of efficient stock trading software. The primary issue is connected with the delay when dealing with the real-time market data handling process. The stock market involves drastic changes in prices which require up-to-date market data for correct trading decision making. There is a need for fast and reliable systems able to provide market data and perform trades but currently used approaches do not satisfy these requirements [4].

The primary task to solve is system scalability. The system should be capable of dealing with numerous users' requests along with other operations including database and API usage. It works poorly owing to the design of the system.

There are performance issues in the system that lead to slower operations and higher probabilities of transaction failures. Security is another important issue that trading systems have to tackle. The system deals with personal data of customers, such as their financial information and transaction records. There are threats to user security since authentication and transmission mechanisms are vulnerable to attacks by hackers [12].

The current systems lack user-centered design, hence making it hard for beginner investors to cope with such complex systems. Though expert traders might fancy sophisticated tools, novice traders need easy-to-use interfaces and processes to engage actively in the market. The software is not reliable because it uses third-party APIs that supply real-time data yet limit access and cause inconsistencies in the data and services.

IV. OBJECTIVE OF THE STUDY

Primary objective of this study involves designing an online stock trading application that can cater to several users as well as real-time stock market data. This is achieved by developing intuitive interfaces that interact with complex trading systems due to the adoption of existing web development technology in the project.

- Some of the particular objectives include:
- Design an enjoyable and user-friendly interface using React to provide a smooth user experience.
- Development of an Nginx reverse proxy implementation.
- This project demands a MongoDB database schema that should have efficient performance characteristics.

There is need to integrate live stock market data in the project through the use of Alpha Vantage APIs and other application programming interfaces.

There should be mechanisms for authentication and authorization of users which ensures that user information is safe and secure.

System performance will be assessed using three standards, namely; latency testing, scalability testing, and reliability testing.

One of the aims of the study is to make trading accessible by developing easy trading procedures as well as creating easily understood financial data. Multiple sources have been identified by the researcher for their research. This was achieved through combining their new findings with the already existing sources of material. One of the aims of the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

study is to make trading accessible by developing easy trading procedures as well as creating easily understood financial data. Multiple sources have been identified by the researcher for their research. This was achieved through combining their new findings with the already existing sources of material.

V. LITERATUE REVIEW

It has been extensively researched in various disciplines like finance and learning, and the design of applications for stock trading is also People studying distributed systems in conjunction with their learning about systems. It is evident from the literature that trading activities require three key components namely, real-time processing, predictive analytics, and system optimization.

Through their work in Kearns and Nevmyvaka [1], it is evident that machine learning methods can be utilized to study market microstructures in situations where automated processes are essential for executing trades at high frequencies. It has been observed that efficient algorithms contribute significantly towards improving trade execution and market prediction tasks.

Aldridge [2] researched the high-frequency trading system design including techniques for latency reduction and trading algorithms execution. It has been shown that trading interruption even in microseconds affects the market outcomes therefore the financial system requires high-performance computing capabilities.

Easley et al. [4] explored the financial market behaviour under extreme events that include a flash crash in order to understand what conditions are necessary for market stability which includes two aspects – liquidity and real-time information flow. Thus, the study results show that the system architecture is necessary to develop a stable system which would not fail at any point leading to cascading failure of the entire system.

Barberis and Thaler [5] explore how psychological factors form traders' behaviour from the viewpoint of behaviour research. It has been concluded that the factors such as user interface design elements and information presentation have considerable effects on investor's behaviour therefore the usability plays an important role in trading application development.

Zerodha and Robinhood Markets represent how these concepts could be implemented by offering their platforms that provide the opportunity to analyse markets in real-time using intuitive systems despite growing user traffic. Both systems use the most innovative technologies in order to offer smooth trading services to millions of their users.

JavaScript full-stack frameworks offer companies to create interactive applications that are able to increase users' base since they are based on advanced web development technology. The technologies of React and Node.js enable developing systems that allow handling events due to their non-blocking way of executing tasks. These operations require critical system functions to perform efficiently. However, this branch of research continues facing problems concerning delays and possible breaches and should solve them in regards to its external API problems. The research proposes a new approach for the solution by creating the system of an efficient architecture and data processing.

VI. SYSTEM ARCHITECTURE

The design of the system follows a tiered architecture approach, which provides a necessary foundation for implementing it. The architectural design of the system is depicted in Figure 1.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

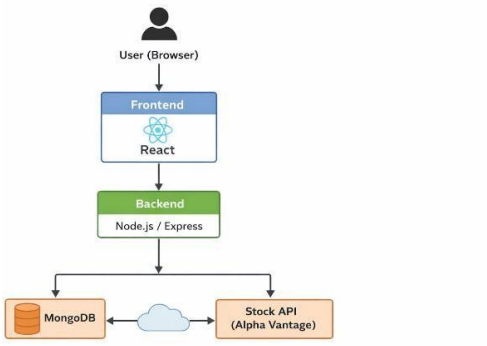

Fig. 1. System Architecture of the Stock Trading Application

Subsystems of the system include:

6.1 Presentation Layer (Frontend)

The frontend system employs React as its development framework, enabling the creation of dynamic content that ensures responsiveness. The system displays live stock through the connectivity with the backend services that work on RESTful APIs and WebSocket connections.

6.2 Application Layer (Backend)

In this case, the backend system employs the use of Node.js as an implementation platform due to its capacity to support event-driven programming and non-blocking input and output operations.

The system undertakes various activities using three key processes including user request processing, execution of buy/sell transaction activities, management of authentication and authorization tasks, and management of API handling processes.

6.3 Real-Time Data Handling

Constant updating of stock prices is needed for a stock trading system. It can be done using:

1) WebSocket to provide live data

2) API polling to make a periodic update

System latency L is calculated as a sum of turnaround time for requests T_request, processing time T_processing, and response time T_response

Where:

Time to send a request is calculated as T_request. Time of server processing is measured as T_processing. Response sending time is measured as T_response.

One of the most crucial parameters of reliability is whether traders are making their decision based on the most exact information.

6.4 The system has the following basic entities:

Designing of Database (Conceptual ER Model) There are two major elements in the system which accept users' requests and process data within a database.

User (UserID, Name, Email, Password) Stock (Stock ID, Name, Price)

Transaction (Transaction ID, Type, Quantity, Date) Portfolio (User ID, Holdings, Value)

Relationships:

Some clients perform many transactions

A portfolio involves different stocks which usually differ from one customer to another

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Each transaction influences a client's portfolio. Portfolio includes different stocks.
Each transaction changes portfolio

VII METHODOLOGY

The methodology discusses the process flow of the application with components that define the operational logic of the system. The process flow of the system has been represented through Fig. 2

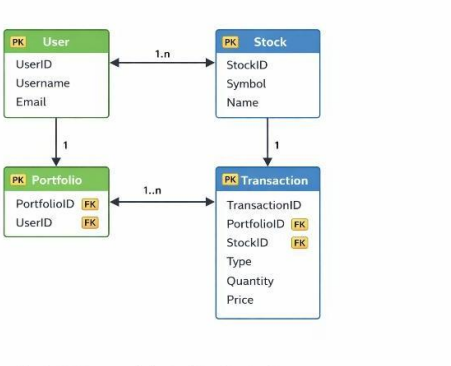

Fig. 2. ER Diagram of the Stock Trading Application

MongoDB is used as the database management system as it provides flexible schemas, which help in storing the user profile information and transaction histories. The system carries out user request activities using three stages starting from the user's request through processing up to responding to the user interface. The system allows requesting for three processes, which include user login, buying, and selling activities. The frontend system makes the user request to the backend application programming interface.

The backend carries out all its operational processes while getting information on stocks from Alpha Vantage. The database system stores all required data elements for the process.

8.1 User Authentication

The system allows users to register and login with their credentials that are encrypted for the purpose of securing access to their accounts. Session integrity is maintained by the use of authentication tokens which ensure no tampering or unauthorized access to the system [12].

8.2 Retrieval of Stock Prices

Stock price information is retrieved through APIs such as Alpha Vantage and integrated into the system in real-time. Information is processed and presented in the dashboard constantly.

8.3 Trading Logic

When a user initiates trading, Buying:
Cost=Price × Quantity
Selling:
Profit/Loss=(Selling" " Price - Buying" " Price) × Quantity
Checks made at the backend:
User’s account balance Stock availability Transaction legitimacy

8.4 Portfolio Management

The portfolio value is automatically updated by: Portfolio" " Value=Σ (Current" " Price $\times$ Quantity)

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Users can track their portfolio performance in real-time.

- User sees stock information
- User initiates trade
- Backend system processes trade
- Database system updates user's portfolio
- User interface receives updated data

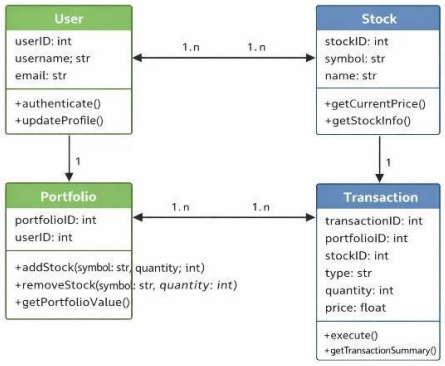

Fig. 3. UML Class Diagram of the Stock Trading Application

Fig. 3 highlights the operational process of the system.

IX IMPLEMENTATION

The suggested stock trading application is implemented using the full the operational processes of the system are illustrated using the visual representation presented in Fig. 4.

The Stock Trading Application Use Case Diagram is presented in Fig. 4. The system works based on three components which include frontend, backend, and database.

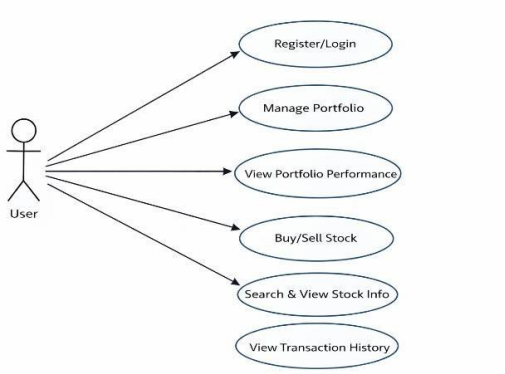

Fig. 4. Use Case Diagram of the Stock Trading Application

9.1 Implementation of Frontend

React framework is used in developing the frontend of the system to provide responsive and interactive user interface in response to any actions taken by users. These are some of the essential components of the system:

- Interactive dashboard that provides information about the prices of stocks currently on the market
- Easy to use interface where trading activities can be carried out by users
- The system provides visual representation of portfolios of users

State management is used to enable updating of the UI without the need for reloading the page,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

9.2 Backend Implementation

For system implementation, Node.js and Express.js are used as technologies in the system's backend implementation. There are three major roles handled by the system which are as follows:

- Processing of API requests
- Execution of business operations of trading system
- Management of user authentication and session

The system employs a RESTful server which uses asynchronous functionalities to handle several requests at once [7].

9.3 Database Implementation

For database operations in the system, MongoDB is employed to handle:

- Storage of user credentials
- Transaction records
- Portfolio details

The creation of collection is made in the system to facilitate fast access and storage of data since this capability is vital for any real-time application.

9.3 Database Implementation

For database operations in the system, MongoDB is employed to handle:

- Storage of user credentials
- Transaction records
- Portfolio details

The creation of collection is made in the system to facilitate fast access and storage of data since this capability is vital for any real-time application.

X RESULTS AND DISCUSSION

With considerations to performance, responsiveness and usability, the created system underwent testing.

10.1 Performance Analysis

In terms of non-blocking architecture built using Node.js, the application shows great ability in handling multiple users. Experimental observation revealed that:

- The system performs better in terms of response to users' API requests
- User Interface updates were done instantly
- The system makes use of database queries effectively.

Reduction of latency within the system came as a result of the adoption of API call optimization and asynchronous processing methods which have proven useful components in a trading system as per reference 2.

10.2 System Accuracy

The system has offered its users an easy-to-use interface allowing them to conduct trading activities in the way they do it at Zerodha. Users are able to:

- Navigate through different parts of the system
- Conduct their trades easily
- Check their portfolio performance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Through monitoring portfolio results in the system, trading operations conducted are verified to be executed accurately. The system offers accurate valuations of portfolio based on the live stock price update leading to exact financial assessments.

10.3 User Experience

The interface provides a trading system which works like any other used in Zerodha. Users are able to:

- Navigate through different parts of the system
- Conduct their trades instantaneously
- Track their portfolio results

10.4 Discussion

The findings emphasize that the use of current web technologies together with financial databases helps to build a scalable and effective trading platform. System performance is impaired as the system relies on external APIs and network availability.

XI CONCLUSION

This paper presented a practical example of a real-time stock trading system that was built with the help of the MERN stack. The system operates successfully as it incorporates frontend, backend, and database technologies creating an efficient trading platform that provides scalable capacity.

The implementation demonstrates that React and Node.js, being modern web technologies, allow working simultaneously with multiple users while analyzing real financial data. The software provides a convenient user experience enabling precise transactions and high system performance.

The system creates a good foundation for future upgrading despite the limitations it possesses, such as its need for external API integration and the lack of advanced forecasting capabilities. It can be argued that next-generation stock trading systems should provide real-time data processing capabilities which should combine with scalable web technologies for their development.

REFERENCES

[1] M. Kearns and Y. Nevmyvaka "Machine Learning for Market Microstructure and High Frequency Trading," Proc. IEEE Int. Conf. Data Mining, 2013.

[2] A. Aldridge High-Frequency Trading: The Practical Guide to Algorithmic Trading Strategies and Trading Systems is a book from the second edition published by Wiley in 2013 from Hoboken New Jersey United States.

[3] S. Russell and P. Norvig Artificial Intelligence is the name of the book written by authors S. Russell and P. Norvig that has been published in its third edition by Pearson publishing house located in Upper Saddle River in New Jersey.

[4] D. Easley, M. López de Prado, and M. O'Hara "The Microstructure of the 'Flash Crash': Flow Toxicity, Liquidity Crashes and the Probability of Informed Trading", J. Portfolio Manag., vol. 37, no. 2, pp. 118–128, 2011.

[5] N. Barberis and R. Thaler "A Survey of Behavioral Finance," in Handbook of the Economics of Finance, vol. 1, pp. 1053–1128, 2003.

[6] OpenJS Foundation, documentation for Node.js software is found in the website https://nodejs.org

[7] Meta Platforms, "The React Documentation," (Online), Available from: https://react.dev/

[8] Zerodha, "Zerodha Trading Platform Overview". [Online] Available at https://zerodha.com

[9] Robinhood Markets, "Robinhood Investing Platform". [Online] Available at https://robinhood.com

[10] The Alpha Vantage API is available in form of an online source that one can access through visiting https://www.alphavantage.com

[11] O. Wasice. "Security on the Web," IEEE Security & Privacy , vol.12, no.4, pp.84-87,2014.