IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Full Stack Stock Monitoring Tool

[1]Yash Khatdev, [2]Shreyash Shinde, [3]Jay Bhor , [4]Kalpesh Mane

Students Department of Computer Engineering, Bharat College of Engineering Badlapur,

Thane District, Maharashtra, India.

Prof. Dr. Radhika Nanda– Professor, Department of Computer Engineering,

Bharat College of Engineering, Badlapur, Maharashtra, India.

Abstract: This paper presents the design and development of a real-time stock monitoring web application built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The system enables users to track stock prices, view historical trends, and receive live updates through API integration. The backend is developed using Node.js and Express.js, providing RESTful APIs for data handling, while MongoDB is used for storing user preferences and watchlists. The frontend is built with React.js, offering a responsive and interactive user interface. Real-time stock data is fetched using third-party financial APIs and displayed dynamically using charts and dashboards. The system ensures scalability, responsiveness, and efficient data handling, making it suitable for investors and traders seeking real-time insights.

Keywords- MERN Stack, Stock Monitoring, Real-Time Data, React.js, Node.js, MongoDB, API Integration, Web Application.

I. Introduction

In the modern financial ecosystem, real-time access to stock market data is essential for informed decision-making. Investors require efficient tools to monitor stock prices, analyze trends, and react promptly to market fluctuations.

Traditional stock monitoring systems are often complex, expensive, or limited in accessibility. With the advancement of web technologies, full-stack development frameworks like MERN have enabled the creation of scalable and user-friendly applications.

This paper proposes a real-time stock monitoring tool that leverages the MERN stack to provide seamless data visualization, real-time updates, and personalized user experiences through a web-based platform.

I.I. Problem Definition

Despite the availability of various stock tracking platforms, many existing systems face several limitations that reduce their overall effectiveness and accessibility. A common issue is the lack of true real-time updates, which can hinder timely decision-making in fast-moving markets. Additionally, many platforms feature complex user interfaces that are difficult for beginners to understand and navigate. High subscription costs further limit access for a broader range of users, while limited customization options prevent users from tailoring the system to their specific needs. Moreover, several platforms fail to offer personalized dashboards or efficient watchlist management, reducing user convenience and engagement.
These challenges highlight the need for a cost-effective, scalable, and user-friendly stock monitoring system that delivers real-time data along with customizable features to enhance the overall user experience.

I.II. Objective

1. The primary objectives of this project are:
2. Real-Time Stock Tracking: Fetch and display live stock prices using APIs.
3. User Authentication: Secure login and registration system.
4. Watchlist Management: Allow users to add/remove stocks.
5. Data Visualization: Display stock trends using charts and graphs.
6. Responsive UI: Ensure smooth performance across devices.

II. LITERATURE SURVEY

Stock monitoring systems have evolved significantly with the advancement of web and cloud technologies. Earlier systems were primarily desktop-based and often provided delayed updates, limiting their effectiveness for real-time decision-making. In contrast, modern platforms leverage REST APIs and WebSockets to deliver real-time data streaming, enabling users to access up-to-date stock information instantly. Technologies such as React.js have further enhanced these systems by enabling dynamic and responsive user interfaces, improving overall user experience. Additionally, cloud-based databases like MongoDB offer scalable and flexible storage solutions, allowing applications to efficiently manage large volumes of financial data. Recent research also highlights the growing preference for full- stack JavaScript frameworks, which streamline development processes and improve performance by enabling seamless integration between frontend and backend components.

III. EXISTING SYSTEM

Existing stock monitoring systems, such as trading terminals and financial dashboards, provide powerful tools for tracking and analyzing market data, but they also come with several limitations. Many of these platforms require paid subscriptions, making them less accessible to a wider range of users. Additionally, they often offer limited customization options, restricting users from tailoring the system according to their preferences. The interfaces of such systems can be complex and difficult for beginners to navigate, creating a steep learning curve. Furthermore, some platforms depend on heavy software installations, which can consume significant system resources and reduce accessibility.

IV. METHODOLOGY

The system architecture is based on the MERN stack model, ensuring a structured and efficient flow of data between components. The frontend is developed using React.js, which provides a dynamic and responsive user interface through reusable components, making the design modular and easy to maintain. API requests are handled using Axios, allowing smooth communication with the backend, while data visualization is achieved through charting libraries such as Chart.js for interactive graphical representations. On the backend, Node.js and Express.js are used to build RESTful APIs that manage client-server communication efficiently. User authentication is implemented using JWT (JSON Web Tokens), ensuring secure access, and middleware is utilized to maintain safe and controlled data flow throughout the application.

For data storage, Supabase is used to manage user information and watchlists, providing reliable database services, while Mongoose is applied for schema modeling to organize the data effectively. The system also integrates external financial APIs to fetch real-time stock market data, which is then processed by the backend before being sent to the frontend. The overall workflow begins with the user logging into the system and selecting stocks to monitor. The backend then retrieves real-time data through APIs, processes it, and sends it to the frontend, where it is displayed using charts and tables. Additionally, the user's watchlist is stored in the database for future access, ensuring a seamless and personalized user experience.

V. APPLICATION AND RESULTS

The application and its results highlight a well-structured and efficient system for stock market monitoring. The interface includes a login and registration page for secure user authentication, a dashboard that displays real-time stock prices, a watchlist page for personalized tracking of selected stocks, and a charts section that visually represents stock trends for better understanding. The system offers important features such as real-time stock updates, interactive charts for analysis, user-specific watchlists, and secure authentication to ensure data protection.

In terms of performance, the application performs reliably under different scenarios. During normal load conditions, it provides fast response times with high accuracy and smooth operation. When handling a high number of API requests, the response time becomes moderate, but accuracy remains high with only slight delays observed. Under multiple user conditions, the system remains stable and demonstrates good scalability while maintaining high accuracy. Overall, the system shows efficient real-time performance with minimal latency, supported by React for quick and dynamic user

interface updates and Node.js for effectively managing concurrent requests. However, the system has certain limitations, such as its dependency on third-party APIs for stock data and the requirement of a stable internet connection for proper functioning.

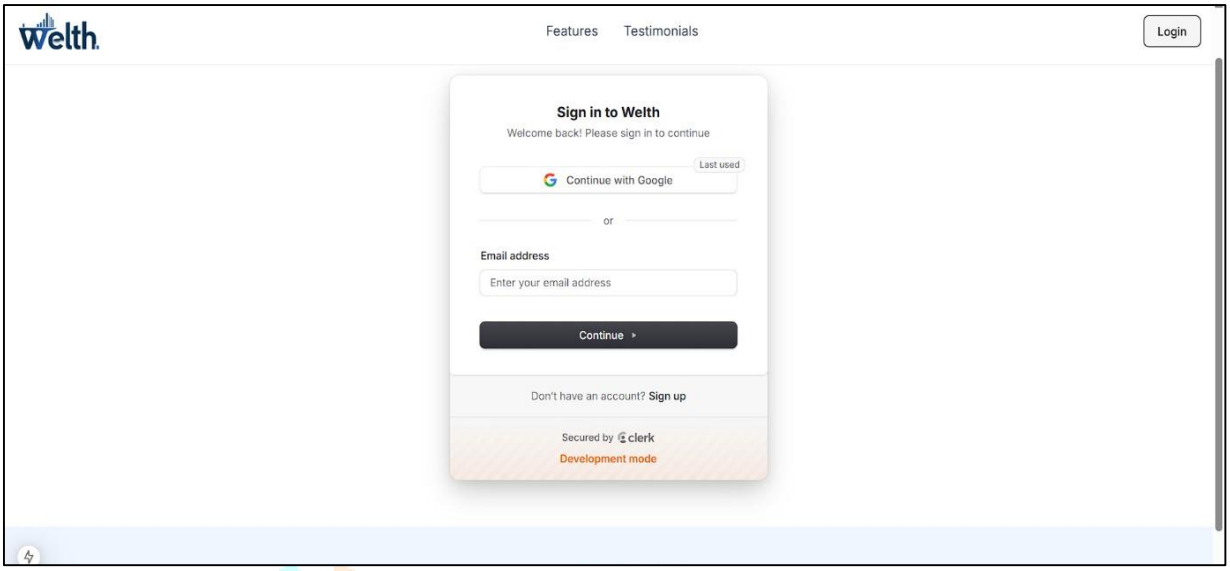

fig.1 login page

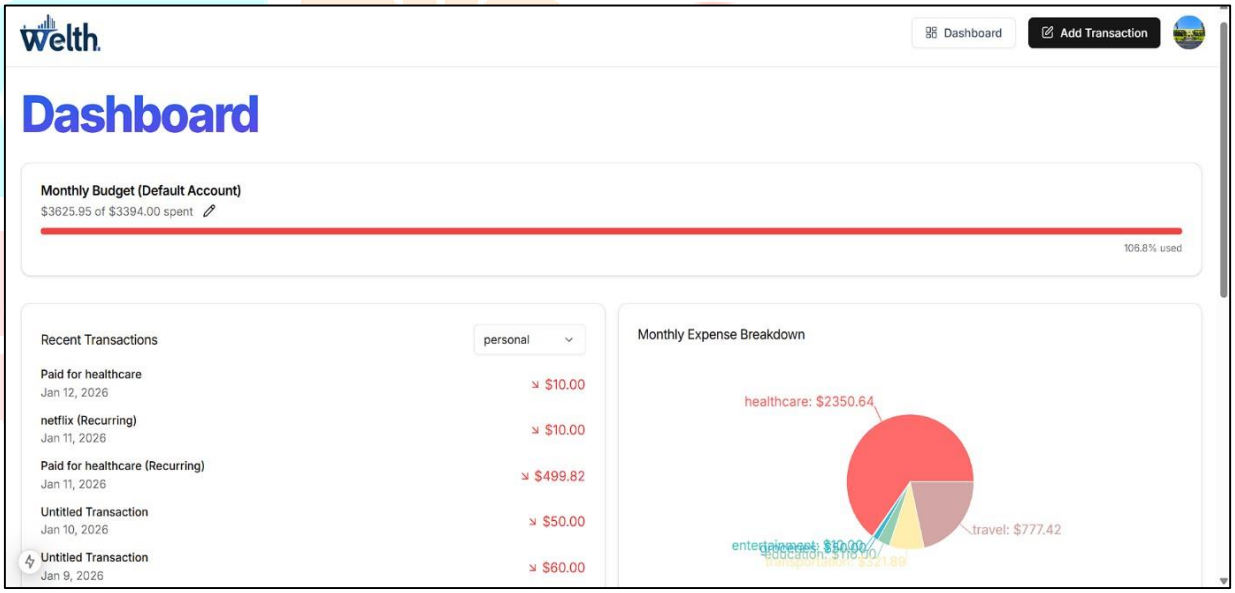

fig. 2 dashboard page

fig. 3 review page

VI. FUTURE SCOPE

1. Future improvements may include:
2. AI-based stock prediction
3. Push notifications for price alerts
4. Mobile application version
5. Integration with trading platforms
6. Advanced analytics using machine learning

VII. CONCLUSIONS

The MERN-based stock monitoring tool provides a scalable and efficient solution for real- time stock tracking. By integrating modern web technologies, the system delivers a responsive and user-friendly experience.

The project successfully demonstrates how full-stack development can be utilized to build powerful financial tools accessible to a wide range of users.

REFERENCE

[1] A. F. Khanbu, M. Shirisha, K. Sailaja, and G. Indraja, "Stock Pulse Monitoring: A Real- Time Stock Monitoring and Intelligent Alert System," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 12, no. 10, 2026.

[2] R. Kalra, "An Efficient Hybrid Approach for Forecasting Real-Time Stock Market Trends Using Deep Learning," *Journal of King Saud University – Computer and Information Sciences*, 2024.

[3] S. Singh, R. Kumar, and A. Badhoutiya, "Web-Based Inventory Stock Monitoring and Control System Powered by Local Encrypted Web Server," *International Journal of Research and Development*, 2023.

[4] S. R. Das, A. Dutta, and S. Bandyopadhyay,"Real-Time Stock Market Monitoring System Using Web Technologies,"International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 245–252, 2021.