TRADE ALERT PRO

1st Abhisekh Pandey, 2nd Vinay S,3 3rd Shivaditya Pandey, 4rd Shreyas R

School of Computer Science Engineering & Information Science, Presidency University,
Itgalpura, Yelahanka, Bengaluru, India

Abstract - Trade Alert Pro stands as an innovative financial tool, designed to provide traders with real-time market insights and intelligent alerts, transforming their approach to navigating dynamic financial landscapes. This platform harnesses advanced algorithms and machine learning to analyze market trends, news, and historical data, offering users timely and actionable trading signals. Trade Alert Pro boasts an intuitive interface and a customizable alert system, allowing traders to stay ahead of market movements, capitalize on opportunities, and effectively manage risks. With a focus on precision and speed, this tool delivers instantaneous notifications, ensuring users are informed of critical developments as they unfold. Whether one is a novice or seasoned trader, Trade Alert Pro serves as a valuable companion, improving decision-making processes and cultivating a competitive edge in today's fast-paced financial markets. Stay informed, make educated decisions, and optimize trading strategies with Trade Alert Pro.

Index Terms – Trading App, API connection, Firebase, Node JS, Json

I. Introduction

The debut of Trade Alert Pro marks a pivotal leap in the ever-evolving realm of financial technology, heralding a new era by amalgamating the formidable capabilities of React.js in frontend development with the robust prowess of Node.js for backend operations. In this epoch where instantaneous access to market insights stands as the linchpin, Trade Alert Pro emerges as an allencompassing web application meticulously crafted to endow traders and investors alike with not just timely, but meticulously accurate and profoundly actionable information. The deliberate choice to synergize React.js and Node.js is rooted in a strategic mandate, one that aspires to birth a platform that isn't just dynamic and responsive but is imbued with a user-centric ethos, surmounting the constraints that have historically bound traditional trading systems. This integration isn't merely a technical decision but a forward-looking move aimed at revolutionizing how market data is harnessed and presented, ensuring that users can navigate the turbulent waters of finance with confidence and agility.

II. Literature survey

This research paper discusses about why React JS is the most preferable tool for developing trading applications. Trading applications have frequent changes in their data. The data involved in trading websites is huge and it should display the changes on the screen much faster so that when the cost of a stock price from a particular company increases or decreases, the user is able to buy the latest share. React JS is one of the most popular libraries of JavaScript developed by Facebook. A practical example has been implemented to show the practical use of these libraries. The concepts discussed in this research paper discuss in detail the advantages of using React JS. One of them is about its cross-platform development. It can be used to build web apps, android apps and even iOS apps. The cross platform syncing helps to save both money and time in the long run. The case study of the trading web App consists of in-depth analysis of the latest library versions available till date for creating forms, navigation bars and adding validations. We will be discussing the frontend framework for the development framework. Git also plays an important role as one of the tools for development of the product. The research discusses the various tools and languages that were used to build the website like HTML5, CSS, Bootstrap 5, ReactJS, JavaScript, Redux, React-redux and many libraries in react. Along with react, there are findings about the languages used for designing the user interface. There is comparison between the functionalities available in the old version compared with the new version. One of such examples include React MUI (Material User Interface) which has undeniably made a developer's life easier. It has many predefined switch tabs and even exceeds Bootstrap in ease of implementation in grids. React Mui grids are very easy to use to develop an interactive user interface. Research consists quite a lot of things that might not fit into this research paper but efforts have been put so that all the learnings can be summed up without leaving any stone unturned.[1]

In the direction of computer globalization and digitization, India is rapidly developing education and information technology. People are taught how to invest in deposits, postal investments, government bonds, gold systems and bonds, and the private sector. The world in which we now live has been completely transformed by technology. The study indicates that there are more than 4 billion active Internet users worldwide, or nearly half of the world's population. Our lives are now faster, easier to manage, and more enjoyable thanks to modern technology. This paper focuses in experiencing and developing a stock application using PHP, React JS, NodeJS and CSS . All the stock data is stored in a MYSQL database. On the other side for developing machine language application python code is used to convert the data into csv format for machine learning algorithms. The investor is presented with a login screen in the python environment where they must enter their user name and password. The stock dashboard shows the investor's current stock holdings, as well as online stocks' their current price, percentage change in stocks, sensex, nifty, bonus, rights, IPO's, annual report etc. statistical methods are used as software modules for the investor, and with a single click of a button, they can compare and contrast their own stocks with online stocks, as well as the trend in the stock market position in order to decide whether to buy, hold, or sell the stocks. Data visualization component is used for comparison of various stocks, and by clicking of a button, stock prediction are displayed whether to hold, buy or sell in future according to the market trend. The trader must log in using their user name and password. The trader will browse the client current market price of all stocks, buying and selling stocks, contract note, client margin, e-off market transactions, ledgers, journals, commission of buying and selling stocks, and so on. In future strategy the stock application programming is converted by a portable mobile application by using python packages like Kivy, PyQt, or even Beeware's Toga library [2].

InvestingIQ is an online platform designed as a one-stop destination where retail investors can get information on the stock market and various stocks in an easy-to-understand manner. The platform provides information about the stock and helps investors perform quantitative analysis with ease. Investors get current as well as past stock price information, the latest news details about the company and other relevant info on the company activity. Apart from the details of the company and its stock, the InvestingIQ platform also houses 2 machine learning models – which are Facebook's Prophet and Random Forest. These two models provide independent predictions of the future stock price of the companies and help investors get an idea of how the stock would perform in the future. Each stock profile also has a section displaying information on how the company is being run by the current management. Metrics like Profitability, Liquidity, Asset Turnover, etc. are formulated from the latest quarter balance sheet of the company. This help investors judge the performance of the company easily. Also, a sentiment analysis model is used to analyse the company's sentiment on various social network sites and news sites and more to get the overall confidence in the company by the general public.All this information together provides investors ample figures to judge the company and decide whether to invest in that company or not in a hassle-free manner [3].

The decentralized application (DApp) is an application that has a backend operation on the distributed computing nodes system. DApp has been built on decentralized technology such as blockchain. The advantages of DApp are security, transparency, and reliability. There are several use cases of DApp for many aspects such as prediction potential trading gains on Augur, sharing economy of computing power by Golem and browsing, chatting, and payment on Status. However, these DApps are utilized without any details about how to implement it. This paper address this issue by presenting the implementation of solar energy trading. Ethereum Blockchain – an open-source platform for DApp has been proposed and applied for solar energy trading. The token is created by using the ERC20 token for trading. The wallet is deployed by Metamask. Transactions, assets, and participants are made by Ganache and tested by Truffle. Moreover, the trading algorithm has been shown to check the correction between seller and buyer with the smart contract on Solidity. Lastly, React- a javascript library for building user interfaces has been deployed as a front- end to make users interactive in solar energy trading [4].

Whilst trading (buying/selling shares) in the stock market, the number of data streams that can be monitored has never been greater. Consequently, traders who do not have years of experience can become overwhelmed and miss multiple opportunities due to visual fatigue. This study investigates using sonification to utilise the auditory channel so that the user can focus primarily on technical analysis whilst still receiving information in a serendipitous peripheral fashion. A custom sonification tool was designed and built to enable NYSE index data to be sonified in real time. One pilot study was conducted during the testing phase with an additional user study of 5-

10 participants to take place after. Through this study it was found that having a continuous stream of data allowed for the trader to have knowledge of the NYSE whilst focusing on cognitively-demanding tasks such as technical analysis and research into further opportunities in different stock exchanges [5].

The project aims to provide retail investors with a third-party investment mobile application to navigate through the stock market. This is achieved through the use of machine learning and mobile web technologies. Several stock price prediction approaches and models are developed including dense, feedforward neural networks, recurrent neural networks, simple linear regressions, and linear interpolations. Model architectures and hyperparameters are optimized and automatically searched by evolution algorithm. Promising results are found for trend prediction. The project serves as a foundation for democratizing machine learning technologies to the general public in the context of discovering investment opportunities. It paves the way for extending and testing out new models, and developing AutoML in the financial context in the future [6].

III. Proposed Methodology

The development of a Trade Alert Website Application involves a comprehensive methodology that encompasses various stages from conceptualization to deployment. Below is a structured approach outlining the proposed methodology for creating a robust and effective Trade Alert Website Application. The proposed methodology for developing the Trade Alert Pro web application, powered by the dynamic duo of React.js and Node.js, represents a strategic and meticulous approach aimed at crafting a cutting-edge platform that revolutionizes the way traders and investors interact with financial data. This methodology is intricately designed to leverage the strengths of React.js for frontend development and Node.js for backend operations, creating a seamless, real-time, and user-centric experience. The journey begins with a comprehensive exploration of the functionalities and features required to meet the diverse needs of users in the dynamic landscape of financial markets. The initial phase of the methodology involves a detailed requirements analysis, where the specific needs and expectations of traders and investors are meticulously outlined. This analysis serves as the foundation for defining the scope of the Trade Alert Pro application, encompassing aspects such as real-time data updates, customization options, user-friendly interfaces, and scalability considerations. Stakeholder consultations and feedback sessions play a crucial role in refining these requirements, ensuring that the application aligns with the practical needs and preferences of its end users. Following the requirements analysis, the design phase unfolds, where the conceptual vision of Trade Alert Pro is translated into a tangible blueprint. React.js takes center stage during this phase, as the modular and component-based architecture facilitates the creation of an intuitive and visually appealing frontend interface. Wireframing and prototyping activities provide a tangible representation of the user interface, allowing for iterative feedback and refinement. The design phase prioritizes not only aesthetic considerations but also user experience, ensuring that the platform is not only functional but also engaging and accessible. Simultaneously, the backend architecture takes shape with the adoption of Node.js. The proposed methodology emphasizes the eventdriven and non-blocking I/O nature of Node.js, positioning it as the backbone of Trade Alert Pro's real-time data processing capabilities. This phase involves establishing secure connections, implementing WebSocket communication for instant data updates, and laying the groundwork for a scalable and efficient backend. The integration of security measures, including SSL/TLS encryption, further solidifies the foundation, acknowledging the paramount importance of safeguarding sensitive financial data. As the frontend and backend components evolve in parallel, the development phase initiates. React.js components are brought to life, providing traders with an interactive and responsive interface. Concurrently, Node.js handles the backend logic, ensuring that real-time data is captured, processed, and disseminated seamlessly. This phase is iterative, with continuous testing and feedback loops to address any discrepancies and refine the features.

IV. CONCLUSION

The development and implementation of the Trade Alert Pro web application, powered by React.js for frontend interactions and Node.js for backend operations, represent a significant milestone in the landscape of financial technology. This comprehensive web platform has been meticulously crafted to provide traders and investors with real-time insights, an intuitive user experience, and robust performance capabilities. In this extensive conclusion, we delve into the overarching achievements, lessons learned, potential areas for future exploration, and the enduring impact of Trade Alert Pro in the dynamic world of financial markets. Trade Alert Pro's journey from conceptualization to realization has been marked by several noteworthy achievements. The decision to adopt React.js for frontend development and Node.js for backend operations was strategic, influenced by the need for a responsive, modular, and scalable solution in the context of financial market dynamics. One of the key achievements lies in the realization of a user-friendly interface that seamlessly combines the power of React.js's component-based architecture with intuitive design principles. Traders and investors can now navigate through complex financial data with ease, customize alerts according to their preferences, and access realtime information effortlessly. React.js's virtual DOM has played a pivotal role in ensuring that the user interface remains responsive and dynamic, even under high data loads. The integration of real-time data processing has been a standout achievement, providing traders with instantaneous updates and insights. Leveraging WebSocket communication, Trade Alert Pro bridges the gap between users and the rapidly changing market conditions, offering a comprehensive and up-to-date view of relevant data. The platform's adherence to security best practices, including SSL/TLS encryption and robust authentication mechanisms, instills confidence in users regarding the protection of sensitive financial information. In the realm of user-centric features and customization, Trade Alert Pro has excelled in providing a tailored experience for individual traders. The platform's flexibility allows users to set preferences, customize alerts, and interact with market data in a way that aligns with their unique trading strategies. React.js's modular design has facilitated the addition of new features and enhancements, contributing to Trade Alert Pro's adaptability to evolving user needs. The development of Trade Alert Pro has not been without its challenges, and these challenges have yielded valuable lessons that contribute to the platform's maturity and resilience. One notable challenge was balancing the need for real-time updates with ensuring the security and accuracy of the data being processed. The integration of WebSocket communication and backend processing mechanisms required careful optimization to prevent information overload and potential inaccuracies. Through iterative testing and optimization, Trade Alert Pro successfully overcame this challenge, striking a harmonious balance between real-time responsiveness and data integrity. Another challenge involved ensuring cross-platform compatibility, given the diversity of devices and browsers used by traders. React.js's responsive design principles and compatibility testing across various platforms have addressed this challenge, creating a consistent user experience regardless of the device or browser being utilized. Scalability considerations posed a unique set of challenges, especially during peak market hours. Implementing load balancing and auto-scaling mechanisms required meticulous fine-tuning to ensure optimal performance. The iterative testing and refinement of these mechanisms have resulted in a platform that remains responsive and efficient even under the strain of increased user activity. Security considerations, given the sensitivity of financial data, were paramount. Ensuring secure data transmission and robust authentication required constant vigilance and adherence to evolving security standards. Regular security audits and updates have been instrumental in fortifying Trade Alert Pro against potential vulnerabilities. The iterative development and testing processes have underscored the importance of continuous improvement and adaptability. User feedback has been a valuable resource, guiding the refinement of features and the introduction of new functionalities. This iterative approach aligns with the dynamic nature of financial markets and ensures that Trade Alert Pro remains at the forefront of technological advancements. Trade Alert Pro, while a robust and feature-rich platform, is poised for continuous evolution and expansion. The journey does not conclude with the platform's current state; instead, it sets the stage for future enhancements and potential areas of exploration. One avenue for future development involves the integration of machine learning algorithms for advanced analytics and predictive insights. The application of machine learning models can further enhance the platform's ability to recognize patterns, analyze historical data, and provide intelligent forecasts. Predictive analytics can empower traders with foresight, aiding them in making more informed and strategic decisions. The potential integration of additional financial instruments and markets is another area for future exploration. Expanding the scope of Trade Alert Pro to include a broader range of assets and markets would cater to the diverse needs of traders engaged in various financial instruments. Enhancing the platform's collaboration capabilities is another potential direction. The incorporation of social trading features, where users can share insights, strategies, and alerts, could foster a collaborative community within the platform. This communal aspect can contribute to knowledgesharing and a more engaged user base.

V. REFERENCES

[1] Dwivedi P, Joshi A. ReactJS For Trading Applications. In 2022 International Conference on Cyber Resilience (ICCR) 2022 Oct 6 (pp. 01-07). IEEE.

[2] AI, D. P. B. (2023). Investing IQ–Algorithmic Trading Dashboard Powered by AI & ML Built Using ReactJS & Python.

[3] Jakúbek, T., Valovič, R., & Přichystal, J. Summary of Financial Articles.

[4] Thupphae, P., & Ongsakul, W. (2021). Laboratory-Scale Implementation of Ethereum Based Decentralized Application for Solar Energy Trading. In Intelligent Computing and Optimization: Proceedings of the 3rd International Conference on Intelligent Computing and Optimization 2020 (ICO 2020) (pp. 53-62). Springer International Publishing.

[5] Storey, M., & Vickers, P. ADSM: AUDIBLE DISPLAY OF THE STOCK MARKET IN REAL TIME.

[6] Man, C. T. (2019). Stock Price Prediction App using Machine Learning Models Optimized by Evolution (Doctoral dissertation, Department of Computer Science, The Hong Kong University of Science and Technology).