Crypto Trading Platform: MERN Stack with Real-Time Data and Social Features

Gurupriya M
Department of Computer Science & Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham, Bengaluru, India
m_gurupriya@blr.amrita.edu

Mouhitha A
Department of Computer Science & Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham, Bengaluru, India
bl.en.u4aie22075@bl.students.amrita.edu

Saranya Gujjula
Department of Computer Science & Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham, Bengaluru, India
bl.en.u4aie22077@bl.students.amrita.edu

Okesh Reddy Ankireddypalli
Department of Computer Science& Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham, Bangalore, India
bl.en.u4aie22044@bl.students.amrita.edu

Abstract—The increasing complexity of cryptocurrency markets necessitates the development of efficient portfolio management tools that provide real-time tracking, price updates, and market awareness. This paper focuses on an advanced concept of a cryptocurrency wallet service provision that provides a summary of available coins, their prices, available quantities, and customers' transaction histories. Developed for the 'frontend' using React.js and for the 'backend' using Node.js, which uses the CoinGecko API for receiving live market prices and collecting curated news blog articles from other APIs. The data related to the wallets are efficiently managed since the application is built based on the MongoDB schema, and in addition to this, the user authentication is also integrated. A responsive design of the user interface allows for showing both a portfolio summary and the latest news related to cryptocurrencies, thus suggesting an 'everything in one place' solution for users interested in cryptocurrencies. This system shows how real-time data integration could be implemented, how the system is friendly to users, and how the system manages the content. This system could be expanded in the future with predictive analytics and an AI recommendation system for the portfolio management and contents.

Index Terms—Cryptocurrency, Portfolio Management, Real-Time Data, News Integration, React.js, Node.js, MongoDB, CoinGecko API

I. INTRODUCTION

Cryptocurrencies are revolutionizing financial transactions, much like how the internet transformed global communication. As the demand for digital assets grows, traders—from casual investors to institutional players—seek platforms that offer not only real-time trading but also market insights and analytics. However, traditional centralized trading systems often fail to provide dynamic interactions, real-time data updates, and seamless user experiences essential in high-volatility markets.

Imagine a scenario where a trader wants to monitor Bitcoin's price fluctuations in real time while discussing strategies with fellow investors and simultaneously managing their digital portfolio. Existing platforms either lack the ability to integrate these features seamlessly or fail to scale efficiently under high trading volumes. This limitation creates a pressing need for a robust, user-friendly, and scalable solution.

The development of a Virtual CryptoTrading React Application presents a cryptocurrency trading platform built using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The MERN stack ensures high-speed performance, global scalability, and real-time data integration to enhance the trading experience. The backend efficiently handles secure transactions, user authentication, and portfolio management, while the frontend provides an interactive and responsive user interface with features such as a watchlist, market insights, and discussion forums. To power real-time stock and cryptocurrency price tracking, the platform integrates Tradeview API, offering live price updates and technical charting tools. Additionally, for historical data and general market analytics, the CoinGecko API is utilized. With React Router, the platform ensures seamless navigation, enabling traders to switch between portfolio tracking, discussions, and news feeds effortlessly. Beyond trading, the platform enhances user experience by introducing AI-driven sentiment analysis for market trends. For example, if news articles indicate regulatory changes in a major economy, the AI system can assess the potential impact on the market and provide traders with sentiment-based predictions. Features like personalized watchlists, automated alerts, and discussion forums further create an engaging ecosystem where traders can track, analyze, and strategize in one place.

By incorporating these functionalities, the Virtual Crypto-Trading React Application offers an "all-in-one" cryptocurrency trading and information hub, tailored to both novice and experienced traders. The platform aims to bridge the gap between real-time market data, user engagement, and advanced trading tools, providing an intelligent and scalable solution for modern cryptocurrency investors.To highlight the novelty and advantages of this platform compared to existing solutions, a comparative analysis is presented in Table I below.

Feature	Proposed System	Existing Systems
Technology Stack	MERN (MongoDB, Express.js, React.js, Node.js) with optimized real-time updates	Mostly MERN or Laravel-based with limited scalability
Real-Time Data	CoinGecko API + TradingView for interactive visualization	CryptoCompare, CoinGecko API used in other studies, but lacks interactivity
Portfolio Management	Personalized, interactive portfolio visualization with custom watchlists	Basic portfolio tracking with minimal user interaction
Wallet and Transactions	Integrated wallet with buy/sell transactions and secure authentication (JWT, 2FA)	Focuses on UI tracking, lacks secure wallet integration
News & Sentiment Analysis	AI-powered NLP sentiment analysis on cryptocurrency news	Provides basic crypto news without AI-based insights
Security Features	JWT-based authentication, 2FA, hashed passwords (bcrypt)	Basic security measures without two-factor authentication
Social Trading	Built-in discussion forum for user engagement	Lacks user-generated trading discussions
Subscription Model	Tiered access to premium features (news, analytics)	Most platforms are open-access with no differentiation
Performance Optimization	Optimized UI/UX with Lighthouse score (88%)	Slower performance and less optimization (74% average)

II. LITERATURE SURVEY

Anusha et al. [1] proposed an improved cryptocurrency tracking system with a feature of 'favorite list' to monitor preferred cryptocurrencies. Created with the help of MERN stack, the app features real-time and historical rates of various coins and tokens based on the CryptoCompare and CoinGecko APIs. The layout of the website and the design of the site and the contents are user friendly and can easily be accessed by the low and high investment users. Also, it has the concept of a user's balance in a given cryptocurrency, as well as a convenient dashboard for its representation.

Kumar et al. [2] utilizes React JS to design a cryptocurrency tracking application with the help of which the cryptocurrency markets can be tracked in real-time, and which is required in today's fast-moving world for tracking the cryptocurrency markets. It also features a widget to get notifications instantly of any news related to the cryptocurrencies that the user is tracking to make better decisions based on current trends.

Zhang et al. [3] offers an empirical study of cryptographic usages in Ethereum and points out that problematic cryptographic interfaces of smart contracts. The study aims at enhancing the cryptographic APIs and the tools in order to close the gap between the low level cryptographic processes and the high level implementation difficulties. Consequently, this research recommends that cryptographic methods can be incorporated into the MERN-stack application developments to cater for contemporary security demand.

Kumar et al. [4] reported on the creation of CryptoIntel, which focuses on user interface design for tracking cryptocurrencies for both novice and experienced traders. With the cryptocurrency data obtained from CoinGecko, the platform revolves around perpetual CI/CD as the primary development approach for enhanced feedback and bugs' resolution to create the best experience for users.

Zheng et al. [5] spoke about ways of building CryptocMal detectors that involve using machine learning to look for CryptocMal. The study done here points to the utilization of the MERN stack for secure backend processing as well as the processing of real time data with the use of machine learning algorithms for threat detection and user behavior analysis.

Mittal et al. [6] in their work was aimed to deliver a P2P decentralized cryptocurrency exchange platform that centers on security, openness, and user control by means of smart contracts and P2P networks. Emerging regulation and laws problems are being reviewed in the research, which would help to advance the DeFi platforms even further.

Bauriya et al. [7] developed a real time cryptocurrency trading system by using Ethereum blockchain platform and following the Laravel framework. The system enables secure and fast exchange of cryptocurrenices and solves such problems as slow transactions and changes in price.

Mahdavi-Damghani et al. [8] opted for cryptocurrency clustering based on sectors via clustering and web scraping. Using methods like k-means and hierarchical clustering, they classify cryptos into sectors, valuing risk and investment when doing so.

Shivsharan et al. [9] proposed Cryptohub, a tool designed to issue and analyse cryptocurrencies highlighting real time data and portfolio. The platform has a priority watchlist to help the users in making their choices and monitoring their cryptocurrencies.

Sakas et al. [10] employed statistical analysis and Fuzzy Cognitive Mapping to study 10 cryptocurrency related websites with 180 days, customer interaction. This study highlights the need for further improving the digital marketing

efforts in the cryptocurrency organizations.

Khanum et al. [11] evaluate the possible security risks in cryptography wallet exchanges' websites and revealed that they are vulnerable to such factors as strict content security policies and absence of anti CSFR token. The study provides an understanding of why vulnerability assessment should be incorporated for enhancing security measures for the cryptocurrency platforms.

Supriya et al. [12] analyzes a full-stack web application with real-time learning management features powered by React.js for an interactive user interface and Node.js running in the backend for scalability purposes. The authors use secure real-time data handling and authentication, which matches cryptocurrency trading platforms that demand secure transactions and dynamic dashboards and user profile management. Data persistence through MongoDB forms a central aspect of the study since it enables necessary portfolio tracking and historical trade examination in crypto platforms.

Asmitha et al. [13] developed a blockchain-based decentralized user wallet system that enhances transaction speed alongside improved financial security. The proposed design implements Node.js for its backend application programming interfaces while MongoDB manages transaction records similarly to crypto trading environments that require safe wallet management combined with instantaneous marketplace alerts. The study's research into minimizing third-party involvement combined with transparent transactions directly benefits the operations of peer-to-peer crypto networks.

Madhuja et al. [14] demonstrate cryptocurrency arbitrage strategies through the examination of real-time markets utilizing optimized data structures. Live price updates are obtained through integrated APIs, while the platform uses MongoDB as a structured data storage method, which matches the requirements of crypto trading platforms needing efficient price monitoring. The research details automated trading solutions built with Node.js and Express.js, which match requirements for high-speed operations within decentralized trading domains.

Valsan et al. [15] presents a blockchain-based asset trading platform that establishes secure transactions and operates decentralized markets. The system architecture combines React.js for user interface interactivity with Node.js for the quick updating of trading data, thereby enabling the application to cryptocurrency trading operations. The integration of smart contracts for automated transactions mirrors the way decentralized finance (DeFi) applications handle secure and transparent cryptocurrency trades.

Sai et al. [16] developed a MERN stack-based real-time chat application focused on instant data sharing and YouTube UI connectivity. The methodology provides essential functionality to crypto trading platforms that support live discussion groups together with user interaction and social trading capabilities. The authors employ WebSockets for live communication alongside MongoDB as a permanent storage solution to create performant digital crypto trading platforms.

III. METHODOLOGY

Developing the Virtual CryptoTrading React Application involves a structured methodology, covering system architecture, frontend and backend implementation, and API integrations. It has evolved into a platform where users can effectively trade, interact via blogs and chats, view real-time cryptocurrency quotes, and make purchases or store cryptocurrencies. The application has modern web technologies and follows a well-defined system architecture for both the frontend and backend, ensuring efficient performance. The welcome page of this application is shown in Fig. 1. To enhance user experience, the platform incorporates interactive dashboards, real-time market trend visualizations, and a responsive user interface optimized for various devices. Additionally, robust security measures, such as two-factor authentication and encrypted transactions, are implemented to ensure the safety of user data and assets.
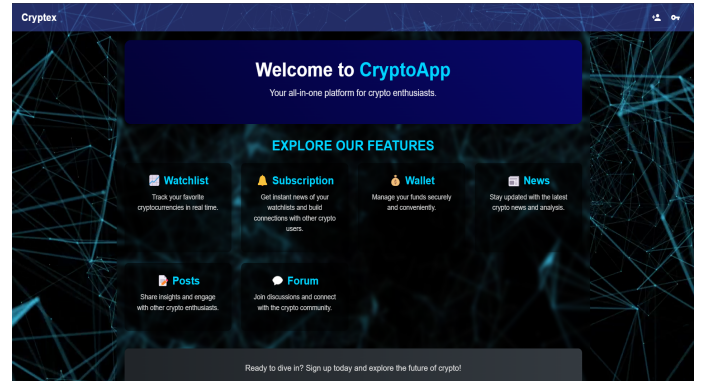

Fig. 1. *welcome page of the proposed Crypto Currency Application*

A. System Architecture

The concepts are built with the use of MERN stack, which stands for MongoDB, Express.js, React.js, and Node.js. MongoDB is used as the NoSQL database to store user data, portfolios, wallets balances, transactions history, and blog posts. The backend API is implemented utilizing Express.js and Node.js for user authentication, portfolio tasks, wallet operations, transaction history, blog creation and external API interaction. Frontend development is done in React.js, it allows to create vivid and responsive user interface. The use of these technologies enable the accomplishment of easy real time synchronizations and seamless user experience.

Algorithm 1 Virtual CryptoTrading System Implementation

Result: Real-time cryptocurrency trading with portfolio management and social features.

Input : User login credentials, cryptocurrency market data (via APIs), user portfolio and wallet data, buy/sell transaction requests, user discussions and news feeds.

Output: Real-time market updates, updated user portfolio and wallet balance, executed buy/sell transactions, secure authentication and user session management, sentiment analysis of cryptocurrency news.

System Initialization: Setup backend server (Node.js & Express.js), connect to MongoDB, configure frontend (React.js), and setup API endpoints.

User Authentication & Session Management: Validate user credentials, assign JWT token, implement secure access control, and enable Two-Factor Authentication (2FA).

Start

while *True* **do**

    Fetch Live Market Prices: Use CoinGecko API to retrieve cryptocurrency price, market cap, and volume.

    Display Real-Time Charts: Integrate TradingView API for price visualization and technical analysis.

    Portfolio and Wallet Management: Retrieve user holdings, update portfolio values, and manage virtual wallet transactions.

    if *User initiates buy request* **then**

        Process Buy Order: Verify sufficient balance, fetch latest price, deduct balance, and update holdings.

    end

    if *User initiates sell request* **then**

        Process Sell Order: Verify cryptocurrency balance, check real-times prices, and update wallet.

    end

    Manage Watchlist: Allow users to track specific cryptocurrencies and fetch live price updates.

    Enable Social Discussions: Users can create posts, comment, and engage in trading discussions.

    Sentiment Analysis: Fetch news via NewsAPI, analyze sentiment using NLP (BERT Model), and display structured insights.

    Subscription Model & Monetization: Implement free and premium tiers, integrate payment gateway, and validate user subscriptions for access to advanced features.

end

B. Frontend Implementation

The frontend of the application is created using React.js to develop solo and complex user interfaces. The major functionalities that would be available to the user under the frontend are: user authorization, portfolio management, wallet operations, traders and exchanges for buying and selling, real time tracking of the desired currencies, organized and Customer Relationship Management with information through the watchlist, and the blog/discussions which can be seen in Fig 2.
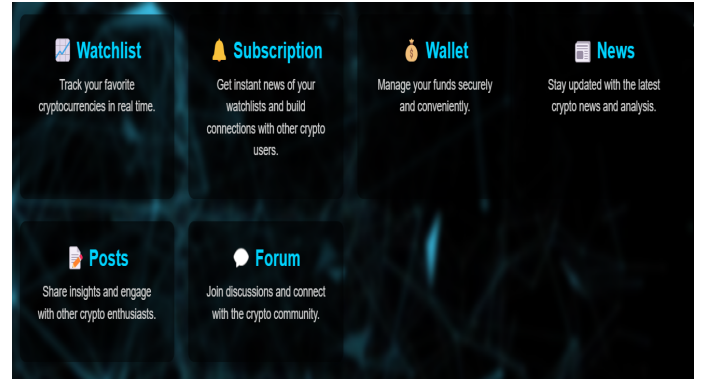

Fig. 2. *Available features of the proposed Crypto Currency Application*

1) User Authentication: JWTs are used to provide the mechanism of user authentication. When a user logs in or register their credentials are checked and if the authentication is successful a JWT token is created. This token is stored in the local storage of the user's browser and to the subsequent API request, the token is used to authenticate the user. If the token is invalid or missing the user is immediately redirected to the login page in a bid to guarantee maximum security to only the authorized persons in areas that require such a token.

2) Portfolio Management and Cryptocurrency Tracking: The portfolio management feature enables users get details of their wishlist in cryptos at a given time. There is the ability to manage the portfolio by including or omitting a particular type of cryptocurrency, and the value alters as the real rates will show. The market data for the cryptocurrencies are obtained from third-party API's namely the CoinGecko API in order to offer the users updated information in terms of price, market capitalization among others.

3) Wallet Functionality: The wallet feature has enabled the users to store, view, as well as manage their amount conveniently in the application. Every user has an option of a virtual wallet that keeps record of the balance of the various currencies that are held. The identified wallet information is saved in the back end database, which makes wallets more persistent and intact across multiple sessions.

The wallet allows users to:

- View their available balance.
- Deposit amount into their wallet.
- Withdraw amount to external addresses.
- Check transaction history for each transaction which came be either subscription or buy or sell cryptocurrencies.

This feature is implemented inside the backend API, where the user's wallet data are also kept and updated according to the performed activities (for instance, purchasing or selling cryptos).

4) Buying and Selling Cryptocurrencies: Crypto trading is done through requests made to external Cryptocurrency Exchange APIs for buying and selling. Wallet balance can be used to purchase or sell supported crypto currencies.

- **Buying Cryptocurrencies**: If a user wants to buy a certain cryptocurrency, he or she enters the quantity they want to buy. The current price of the cryptocurrency is gotten through external APIs such as CoinGecko or direct from a Cryptocurrency exchange API for instance the Binance API. In the case of the user, the application then subtracts the corresponding value of the fiat (or any other cryptocash) balance in the wallet and adjust the portfolio with the new possession.
- **Selling Cryptocurrencies**: In order to sell a cryptocurrency, you are to enter the quantity of cryptocurrencies you want to sell. The system ensures that the user has sufficient amount of that particular cryptocurrency in his/her wallet. When confirmed the application computes the current value on the current market price, automatically deducts the amount from the user's wallet balance and also registers the transaction in the transaction history. The user is then paid out fiat currency or stablecoins for their cryptocurrency of choice.

5) Watchlist and Real-Time Data: First, there is a watchlist option on the application where the users can follow specific cryptocurrencies' performance. This watchlist is saved in the backend database and, via the APIs used, the application draws real time data from external API such as Coin Gecko API which offers other features such as crypto market prices, volumes and market capitalisation as shown in Fig. 3. The time for updating the data being every few of few seconds, the user is provided with the most current information.

Fig. 3. *Real time visualization of crypto currency prices*

6) Cryptocurrency News Feed: The application must have the news feed from the field of cryptocurrencies to provide the user with all the necessary information about the current situation on the market. In the news feed, the articles are delivered by the `NewsAPI`, which returns the articles of the latest cryptocurrencies. The API delivers news articles based on the user's choice of, for instance, cryptocurrency markets or updates and general market trends. The articles appear with headlines, brief descriptions of the articles, and clickable links to the full article.

7) Data Visualization and Charting: The application is incorporated with real time Cryptocurrency price chart trough TradingView Charting Library. Through these charts the user is able to get a perspective of price trends in a given time frame from 24 hours to 7 days or 1 month. The charts are interactive in nature that collects real time data from external APIs while making the dynamics of the trends visually attractive for the users.

C. Backend and API Integration

It has also extended opportunities for backend and API integration because such a set of tools is ideal for using application programming interfaces for various purposes. For the backend of the application, programmers employ Node.js and Express.js to handle APIs and user data and to get information on cryptocurrencies and related essential info, including market data and news.

1) User Authentication and JWT Tokens: User authentication is implemented in the backend to use the JSON Web Token; JWT. When a user logs in their credentials are checked, then a JWT token is created by passing the user's data to it. This token is stored in the frontend side for convenient and applies in any consequent API call. The backend guarantees that only authenticated tokens can interact with secured resources as for example, the user's portfolios and watchlists, and any other form of data that a user deems private.

2) Wallet and Transaction Management API: The backend provides endpoints for managing the user's wallet, including:

- Firstly, the amount of each cryptocurrency in the user's wallet during the present time should be verified.
- Recording deposits and withdrawals of cryptocurrencies.
- Keeping track of the user's transaction record on every cryptocurrency regarding both purchase and selling as shown in Fig. 4
- Performing buy and sell operations based on the current market prices fetched from external APIs.

The backend communicates with an internal API to get the current listed prices of cryptocurrencies from different exchange platforms or CoinGecko when computing for the value or cost of cryptocurrencies during, and during transactions.

Fig. 4. *Transaction history of all the transactions performed by the user*

To track user transactions, the following methods are implemented:

- Detailed Transaction Logs: Each transaction is stored with timestamps, transaction IDs, and user details.
- Downloadable Transaction History: Users can export their transaction history for personal records and compliance.

These features allow users to monitor their transactions with complete transparency.

3) Cryptocurrency Market Data Integration: Realtime information about cryptocurrencies such as price, market capitalization, volume and price trends are obtained from the CoinGecko API. The backend also being an API interacts with CoinGecko by sending requests and the obtained data is then formatted and passed to the frontend. It is used to populate a user's portfolio or watchlist and also the charts you see on the site. This data is updated as often as possible in order to display the most current market information.

4) Enhancing Reliability: To enhance the reliability of the backend, the following strategies are implemented:

- Error Handling Mechanisms: Comprehensive exception handling prevents system crashes and ensures smooth operation.
- Failover Database Replicas: Redundant database instances provide continuous availability in case of failures.
- Regular Backups: Automated backups protect against data loss and enable disaster recovery which is provide through MongoDb Atlas.
- Transaction Consistency (ACID Compliance): Ensures financial transactions are executed reliably and securely.

These measures ensure that the backend remains robust, scalable, and fault-tolerant, allowing for uninterrupted trading and secure portfolio management.

5) Cryptocurrency News API Integration: The `NewsAPI` fetches real time news articles related to cryptocurrencies from various source. The backend uses the HTTP request to send the requests to the `NewsAPI` with certain parameters on cryptocurrency including 'bitcoin', 'ethereum' and 'crypto market'. The response is then calculated and sent back to the frontend where the articles are conveniently presented in lists to the user. The articles contain headlines, brief overviews and links to the rest of the material.

6) Sentiment Analysis of Financial News: In this platform, AI is primarily utilized for analyzing financial news articles using advanced Natural Language Processing (NLP) techniques. The platform integrates NewsAPI to fetch the latest news related to specific cryptocurrencies. AI-driven NLP algorithms process these articles to extract sentiment, identify key trends, and detect potential market-driving events. This sentiment analysis provides traders with valuable context, highlighting how current events might influence cryptocurrency market behavior. By evaluating the tone and sentiment of articles, AI helps traders understand public perception and the underlying factors driving market fluctuations.

D. Subscription Model

The application operates on a subscription-based model, where users can access additional features based on their membership level. Core functionalities, such as portfolio management and tracking a watchlist, are available to all users as shown in Fig. 5. However, premium options like commenting on the blog and accessing instant news articles are exclusive to users with a paid subscription. Users can manage their subscription services, including subscribing or canceling, directly from the backend. Payment processing and subscription validation are handled through a third-party service, ensuring that only subscribed users can access the premium features.
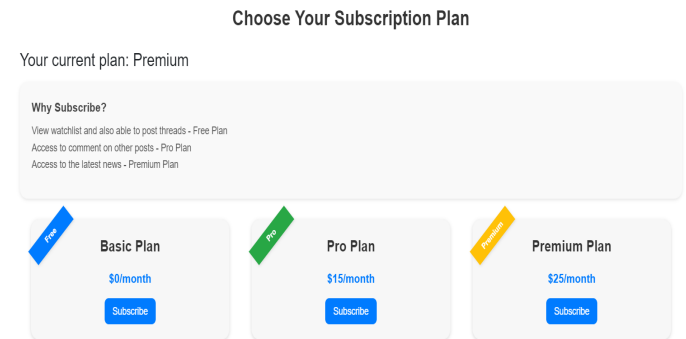

Fig. 5. *Type of subscription and its benefits*

To enhance the precision and security of cryptocurrency transactions, several improvements have been implemented. The application ensures accurate floating-point calculations to prevent rounding errors in financial transactions. Data validation techniques have been incorporated to verify correct user inputs, minimizing errors and enhancing reliability. Additionally, two-factor authentication (2FA) strengthens user security and reduces fraudulent activities, providing a more secure environment for users engaging with cryptocurrency features.

E. Testing and Quality Assurance

Ensuring the reliability and efficiency of the Crypto Trading Platform requires a structured approach to testing and performance optimization. The system has been evaluated through multiple testing methodologies, including frontend, backend, security, and performance testing.

1) Frontend Testing: For frontend testing, tools such as Jest and React Testing Library are used in order to automate test frameworks. These frameworks are utilized to vet out the appearance of components, the behavior of composed and accumulated user interactions, and how it interconnects with backend functioning. Tests check that all CRUD operations that imply updating and querying the portfolio, the wallet, the watchlist, and getting real-time data work fine.

2) Backend Testing: For testing the backend API, tool such as Postman and Mocha are used. They assist to ensure that the backend API routes are properly working as well as the response they are supplying. There are user authentication tests, portfolio management tests, wallet functionality tests, and tests pertinent to the integration of external APIs such as CoinGecko API and NewsAPI.

3) Performance and Security Testing:
- **Server-Side Caching (Redis):** Reduces API response time by storing frequently accessed data.
- **Optimized API Calls:** Redundant API requests are minimized, and batch processing is used for efficiency.
- **Database Indexing (MongoDB):** Indexing is applied to optimize read and write operations for large transaction data.

Testing for security is conducted for the reasons of ascertaining that users' sensitive details including the login details and the portfolio details do not end up in the wrong hands of malicious hackers. Further, in penetration testing, the specialists try to find weaknesses in the system.

IV. RESULTS

The Virtual CryptoTrading React Application effectively incorporates the fundamental modules in relation to cryptocurrencies such as tracking of portfolio, analysis of real time market data, updated news feeds on the specific crytocurrency and secure cryptocurrency wallet. Upon testing, the system proved to perform well in processing the real-time

data, implement user authentication using JWT securely, and allow the user to add the cryptocurrencies into the model and delete or watch the changes in the user's list effectively.

A. Performance and Usability

The application was tested under primary conditions such as multiple users connected to the application, real-time data retrieval feature, and portfolio management. The system worked fine and the response time of API calls as much as fewer numbers of cryptocurrencies as some of them are in the image below average less than 2 seconds. particularly was easy to navigate since the frontend part of the application was created using React Router, which made it easy to switch between the portfolio, the blog and the watchlist. Also allowing access to multiple blogs at a time as shown in Fig. 6. Bootstrap was used to keep the application layout responsive across devices which enhanced the overall use of the application.
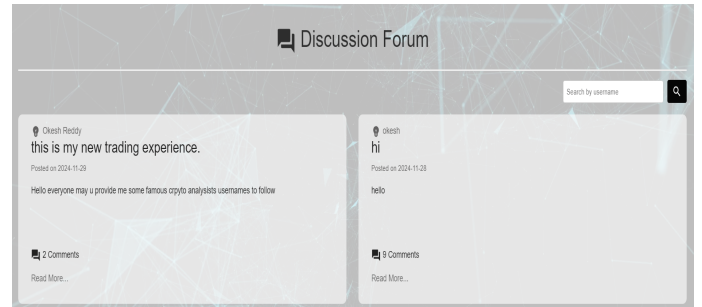

Fig. 6. *Discussion Forum to communicate with other crypto enthusiasts*

B. Transaction and Wallet Management

Incorporating wallet capabilities helped to develop realistic transactions during the course of the study. It was easy and fluid to add and delete cryptos to a wishlist. The buying and selling of cryptocurrencies was hitch-free. During the purchase of cryptocurrencies, users received cues such as the balance in their wallet and the price of the cryptocurrencies, which made the trading easy to use. The system made sure that every trade was safely completed; the wallet balance was also updated on a real-time basis.

C. Security and Authentication

The employed JWT based authentication system was effectively used to handle and secure user sessions and the application data. All the user interactions with the system were transmitted through a secure channel that required authentication each time the user wanted to access the restricted areas such as the watch list and creating personal blogs. Passwords were saved into hash using bcrypt in order to minimize the risk of the users' credentials Pin if the database disclosure happens some day. The usage of secure API routes that will be used for user data retrieval, transaction

processing, and wallet management secures the user data throughout the platform.

D. Market Data Integration

With the help of CoinGecko and NewsAPI, the application featured unrestricted real-time market data and the latest cryptocurrency news. Using the CoinGecko API, genuine and real-time information was obtained regarding the prices, market capitalization, and volume of cryptocurrencies, providing users with accurate market insights.

To further enhance the platform's functionality, predictive analytics was achieved by integrating Natural Language Processing (NLP) techniques for sentiment analysis of real-time financial news. The platform utilizes NewsAPI to fetch relevant news articles related to specific cryptocurrencies, and NLP algorithms, including the Bidirectional Encoder Representations from Transformers (BERT) model, analyze the sentiment and key insights from these articles.

By leveraging BERT, the platform captures contextual meaning and nuanced sentiment from financial news, improving the accuracy of sentiment analysis. The system analyzes news articles for nine major cryptocurrencies, ensuring that both positive and negative sentiment articles are equally displayed for each specific coin. The overall accuracy of sentiment classification using BERT reached 85%. However, the sentiment analysis results are not displayed directly on the front end; instead, they are used for grouping news articles in a structured manner. To ensure a balanced market perspective, the news feed is organized in an alternating format, where positive sentiment articles are followed by negative sentiment articles, allowing users to simultaneously view both bullish and bearish trends. This arrangement helps traders quickly assess both perspectives, providing a more comprehensive market outlook.

Unlike traditional approaches, this context-driven method allows the platform to provide informed predictions based on current events without relying on machine learning-based time-series forecasting models like LSTM.

Users were also presented with organized news related to cryptocurrency price trends through NewsAPI integration, as shown in Fig. 7. This integration aimed to provide users with up-to-date information, helping them make more intelligent and informed cryptocurrency investment decisions.

E. User Engagement and Interaction

The blog and the discussion that was made available for the users offered them a platform through which they could post their findings, queries and ideas as well as interact with other members in the world of cryptocurrency. It enhanced user engagement and provided means for users discuss on several cryptocurrencies. The system was also able to monitor and record engagement from the users including likes and comments on particular content and ensure that the engagement level was increased. This feature played a role in the development of the community environment within
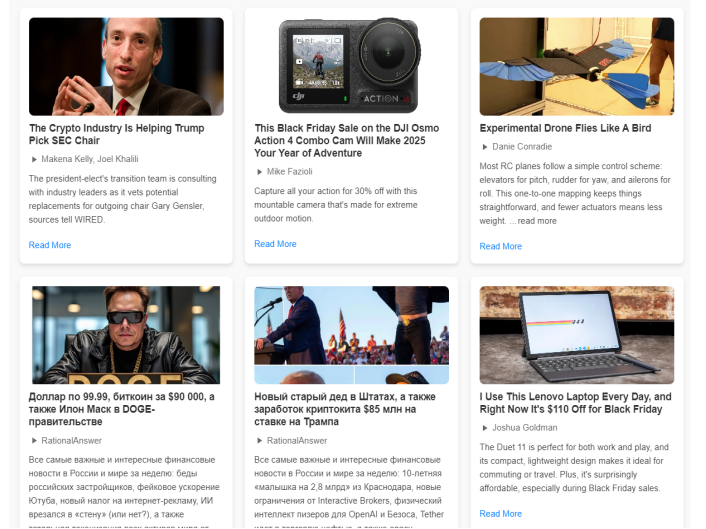

Fig. 7. *News fetched for the personalized watchlist through NewsAPI*

the given application, and made it possible to ensure that the end-users are engaged within the purview of the application, and not only with the portfolio.

F. Website Performance Comparison Using Lighthouse

The functionality and interactivity of the Virtual Crypto-Trading application were evaluated using Google Lighthouse: a tool for measuring a website's responsiveness, accessibility, and optimization given key parameters of necessity, performance, and other critical benchmarks.

TABLE II
WEBSITE PERFORMANCE AND METRICS COMPARISON

Website	Perf.	Access.	Best Prac.	SEO	FCP	LCP	TBT	CLS	SI
Website 1	74	72	74	92	1.0 s	1.4 s	390 ms	0.055	1.7 s
Website 2	33	93	70	100	1.9 s	5.2 s	710 ms	0.068	8.6 s
Our Website	**88**	**100**	**100**	**100**	**1.8 s**	**1.9 s**	**190 ms**	**0.006**	**2.0 s**

G. Explanation of Metrics

- **First Contentful Paint (FCP)**: Determines time it takes for the first pixel of content to be displayed on a screen. Faster FCP means that users are able to view content on the page and consequently, readers feel that their time to load a page has reduced.
- **Largest Contentful Paint (LCP)**: Monitors the size of the largest visible bitmap on the page, thus the loading time. A lower LCP value makes certain that central content refreshes rapidly, thus improving the usability of a website.
- **Total Blocking Time (TBT)**: Stands for the period during which the browser is inactive and cannot respond to any actions of the user caused by JavaScript. It is evident from the graph above that products with low TBT value offer better interactivity and responsiveness.
- **Cumulative Layout Shift (CLS)**: Measures the stability of the visual content by measuring layout shifts in terms

of variability. A lower CLS thus guarantees a more stable environment in the view of the viewer.

- **Speed Index (SI)**: Gives the time taken to load different areas of a page in view. While a lower value of SI is desirable as it is an indication that there is faster visual completeness and more user satisfaction.
- **Performance**: Describes the loading, rendering, and interactivity values at different scenarios and situation. Points that denote positive experiences take shorter times and are resource optimally utilized than those with lower scores.
- **Accessibility**: Determines ease of access for the disabled - color contrast as well as navigation without a mouse. It also means that people with disabilities have equal rights and adherence to equal opportunities for persons with disabilities Act in the companies.
- **Best Practices**: Checks on compliance with current trends in website designing and development and security measures. Greater values are indicative of the superiority of design, security and compatibility.
- **SEO**: As a result, it assesses optimization for organic search, tagging: meta data structured data, and mobile. The score increases the visibility and positions of the website in the search engine.

The findings as shown in Table II demonstrate that our website has better functionality than those evaluated and load considerably faster, are not restrictively interactive, easily accessible and optimised for SEO.

V. Conclusion

This application provides a comprehensive suite that enables users to monitor and track their virtual currencies, explore various data types, and engage in virtual currency purchases and sales. The system is built through utilizing the latest web technologies, with the choices of React.js for the front-end and Node.js with the help of Express.js for the back-end, which allows achieving high scalability, fast response, and security. Incorporation of external APIs of CoinGecko and News API helps users obtain real-time data on the cryptocurrency prices and general news to make rightful investment decisions. Moreover, the wallet and transaction management functionalities provide real comfort and security to facilitate a user's experience with a virtual environment for their assets.

The design of the application is lean and modular and this means that in future key enhancements such as multiple currencies support and even better analytical tools can be implemented. Through integration of the features of real time update, wallet, and discussion in the blog, this platform provides a complete solution for the media and those interested in virtual currency. One of the important advantages is that using the measures of automated testing and security will keep the application stable and safe and therefore can become helpful both for the beginners and for those people who are already experienced in alphanumeric manipulation. Future work will be devoted to the extension of the number of available cryptocurrencies and the integration of more exchange platforms to maintain platform efficiency in the quickly developing world of cryptocurrencies.

References

[1] Anusha, N., Vivek, A., Gullapally, A., Teja, P.R. and Rahul, R.S.R., 2024. "Crypto Tracking Web Application." In *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 2, pp. 1-6. IEEE.

[2] Kumar, R., Mishra, A., Mollick, M., Chaurasiya, H. and Roy, K., 2023. "Cryptocurrency Tracker for Real-Time Market Analysis and Monitoring Using React JS."

[3] Zhang, J., Chen, J., Wan, Z., Chen, T., Gao, J. and Chen, Z., 2024. "When Contracts Meet Crypto: Exploring Developers' Struggles with Ethereum Cryptographic APIs." In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1-13.

[4] Choudhary, S., Tripathi, S., Singh, T., Goutam, A. and Prajapati, Y.N., 2024. "Enhancing Crypto Insights Using Redux Toolkit for State Management." *Volume 6, Issue 9*.

[5] Zheng, R., Wang, Q., Lin, Z., Jiang, Z., Fu, J. and Peng, G., 2022. "Cryptocurrency Malware Detection in Real-World Environment: Based on Multi-Results Stacking Learning." *Applied Soft Computing*, 124.

[6] Mittal, H., Sharma, V., Jain, T., Jain, S. and Sharma, S., 2024. "A Decentralized Cryptocurrency Exchange Application." *International Journal of Engineering Trends and Applications (IJETA)*, 11(3), pp. 1-6.

[7] Bauriya, A., Tikone, A., Nandgaonkar, P. and Sakure, K.S., 2019. "Real-Time Cryptocurrency Trading System." *International Research Journal of Engineering and Technology (IRJET)*, 6(3).

[8] Mahdavi-Damghani, B., Fraser, R., Howell, J. and Halldorsson, J.S., 2022. "Cryptocurrency Sectorization through Clustering and Web-Scraping: Application to Systematic Trading." *Journal of Financial Data Science*.

[9] Shivsharan, N.M., Kambli, S.G., Nikharge, O.V. and Kudatarkar, K.U., 2023. "Facilitating Cryptocurrency Analysis and Watchlist Management with a Web-Based Platform." In *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, pp. 1-6.

[10] Sakas, D.P., Giannakopoulos, N.T., Kanellos, N. and Tryfonopoulos, C., 2022. "Digital Marketing Enhancement of Cryptocurrency Websites through Customer Innovative Data Process." *Processes*, 10, p.960.

[11] Khanum, S., Bisht, D., Kashyap, K. and Mehta, M., 2023. "Vulnerability Assessment of Cryptocurrency Wallet and Exchange Websites." In Marriwala, N., Tripathi, C., Jain, S. and Kumar, D. (eds) *Mobile Radio Communications and 5G Networks*. Lecture Notes in Networks and Systems, vol. 588. Springer, Singapore.

[12] Supriya, M., Rathinavel, K. and Radhakrishnan, D., 2024, November. Development of a Comprehensive Learning Management System using Full Stack Technology: A Modern Approach to Educational Platforms. In *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)* (pp. 1426-1431). IEEE.

[13] Asmitha, M. and Kavitha, C.R., 2024, October. Decentralized User Wallet: Transforming Digital Banking with Blockchain. In *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 566-571). IEEE.

[14] Madhuja, C., Yadav, S. and Sidhu, S., 2024, July. Analysis and Implementation of Cryptocurrency Arbitrage Strategies: A Data Structures Approach. In *2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)* (pp. 1-5). IEEE.

[15] Valsan, V., Kanakasabapathy, P., Ramesh, M.V., Varsha, L., Deeksha, M. and Ankith, S., 2023, September. Blockchain based Smart Energy Trading in a Sustainable Community: A Comparative Technological Analysis. In *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 6, pp. 561-567). IEEE.

[16] Sai, P.C., Karthik, K., Prasad, K.B., Pranav, V.S. and Ramasamy, G., 2025. Web-based real time chat application using MERN stack. In *Challenges in Information, Communication and Computing Technology* (pp. 195-199). CRC Press.