ISSN: 2455 - 1597

ijergs
Open Access

International Journal of Engineering Research and Generic Science (IJERGS)
Available online at: https://www.ijergs.in

Volume - 11, Issue - 2, March – April - 2025, Page No. 10 - 16

Rapid News: A High Performance Real-Time News Aggregator with MERN

[1]Sagar Pradhan, Assistant Professor, Department of Computer Science, Arya College of Engineering, Jaipur.

[2]Utkarsh Bhatnagar, Research Scholar, Department of Computer Science, Arya College of Engineering, Jaipur

[3]Varsh, Research Scholar, Department of Computer Science, Arya College of Engineering, Jaipur.

Abstract

Block chain-enabled e-voting (BEV) could reduce voter fraud and increase voter access. Eligible voters cast a ballot anonymously using a computer or smartphone. BEV uses an encrypted key and tamper-proof personal IDs. This article highlights some BEV implementations and the approach's potential benefits and challenge in today's information-driven world, access to timely and accurate news has become essential for decision-making, awareness, and engagement. However, conventional news platforms often lack speed, interactivity, and personalized content delivery. This paper presents Rapid News, a high-performance, real-time news aggregator built using the MERN stack—a powerful combination of MongoDB, Express.js, React.js, and Node.js. By integrating modern web technologies with performance optimization techniques such as asynchronous API calls, lazy loading, in-memory caching, and modular frontend components, Rapid News significantly reduces latency and enhances scalability. The application sources data from multiple news APIs, filters it based on user-defined categories, and delivers real-time updates. The research emphasizes the architectural design, implementation strategies, and performance evaluation of the system. Testing reveals major improvements in response time and user satisfaction compared to conventional news delivery methods. This work demonstrates the effectiveness of the MERN stack in building scalable, high-speed web applications tailored for real-time content consumption.

Keywords: MERN Stack, ReactJS, NodeJS, MongoDB, ExpressJS, News Aggregator, Real-Time Systems, Web Performance, RESTful APIs, UI Optimization.

Introduction

In recent years, the digital landscape has undergone a significant transformation, fundamentally changing the way people consume, share, and interact with news and information. Traditional news consumption methods—such as print newspapers and scheduled television broadcasts—have gradually been replaced by web-based platforms and mobile applications that provide instant updates. This transformation has been driven by the proliferation of the internet, the rise of smartphones, and the development of new technologies that facilitate real-time communication and content delivery.

With the growing number of digital news sources—ranging from mainstream media outlets and online journals to independent blogs and social media feeds—users today are exposed to an overwhelming volume of information. While this abundance provides ample access, it also introduces new challenges. The issue is no longer about the availability of news; instead, the primary challenge lies in ensuring access to relevant, trustworthy, and timely information. In the flood of continuous updates, advertisements, misinformation, and clickbait headlines, users need efficient tools to filter out noise and focus on what truly matters to them.

To meet this demand, real-time news aggregators have emerged as a critical solution. These systems aim to gather headlines and news content from various online sources, including public APIs and RSS feeds, and present them in a structured and digestible format. Moreover, they categorize news by topic—such as politics, technology, sports, health, and entertainment—and tailor content delivery to user preferences. By delivering updates dynamically and instantly as events unfold, these aggregators enhance user engagement, improve news relevance, and reduce the time users spend searching across multiple platforms.

However, legacy news platforms and aggregators often fall short in delivering these benefits due to several technological and architectural limitations. Many of these platforms are built using monolithic frameworks that are difficult to scale and adapt to the demands of real-time data processing. Their reliance on server-side rendering and synchronous operations leads to increased latency, slower content updates, and reduced interactivity. Additionally, older platforms may not support cross-platform compatibility, hindering the user experience on different devices such as desktops, tablets, and smartphones. A lack of personalization features—such as bookmarking, interest-based filtering, and AI-driven recommendations—further limits their effectiveness in serving diverse user needs.

To address these shortcomings, this research introduces Rapid News, a high-performance, modular, and scalable real-time news aggregator web application developed using the MERN stack—which includes MongoDB, Express.js, React.js, and Node.js. The MERN stack is a popular and powerful JavaScript-based technology suite that enables full-stack development using a single language, offering synchronous integration across the frontend, backend, and database layers. It is well-suited for building modern, responsive, and scalable applications that can process real-time data efficiently and deliver a seamless user experience.

Rapid News is designed with performance and usability at its core. It fetches news articles from multiple APIs, processes and filters them based on categories and keywords, and displays them dynamically on a single-page application (SPA) built using React.js. Users can explore different topics, search for specific content, and save or bookmark articles for later reading. The application also uses techniques such as lazy loading, in-memory caching, and infinite scrolling to minimize load times and reduce server strain.

The backend, developed using Node.js and Express.js, handles API integration, user authentication, and routing. It communicates with MongoDB, a NoSQL document-based database, to store user preferences, article metadata, and session information. With the help of tools like node-cron, the system performs scheduled data fetching and ensures that the news content is regularly updated without manual intervention.

This research explores the entire development lifecycle of Rapid News—from architectural design and technology selection to performance tuning and user feedback. It highlights how the modularity and flexibility of the MERN stack enable developers to overcome traditional limitations and deliver a product that is fast, scalable, and user-centric.

In addition to system design and implementation, this study also includes a performance evaluation based on metrics such as response time, load speed, and user satisfaction. The results demonstrate that MERN-based applications can effectively support real-time content aggregation, making them ideal for news, social media, and event-driven platforms.

In conclusion, Rapid News exemplifies the application of modern web development practices to solve real-world problems in the information age. By leveraging the strengths of the MERN stack, it sets a foundation for the development of future-proof, high-performance content delivery systems.

Figure 1:

System Design & Architecture

A. Architecture Overview

The system follows a three-tier architecture:

1. Frontend: Built using ReactJS, offers dynamic rendering, category-based filtering, infinite scroll, and bookmark features.
2. Backend: Powered by NodeJS with ExpressJS, it handles routing, authentication, and acts as a middleware between frontend and external APIs.
3. Database: MongoDB stores user details, categories, search history, and bookmarks.

B. Data Flow

News data is fetched from public APIs (e.g., NewsAPI.org, Google News) using scheduled cron jobs. Fetched articles are filtered, cached, and delivered to the frontend through RESTful endpoints.

Technology Stack

A. Mongo DB

- NoSQL document-based storage.
- Ideal for storing dynamic news metadata and user preferences.
- High-speed querying using indexes and BSON.

B. Express JS

- Lightweight framework for creating server-side APIs.
- Supports routing, middleware integration, and scalable request handling.

C. React JS

- Component-based architecture.

- Features like virtual DOM and hooks (useEffect, useState) enable fast rendering.
- Integrated with libraries like Axios for HTTP requests and Redux for state management.

D. Node JS

- Event-driven, non-blocking I/O system.
- Handles real-time data processing and API calls with asynchronous capabilities.

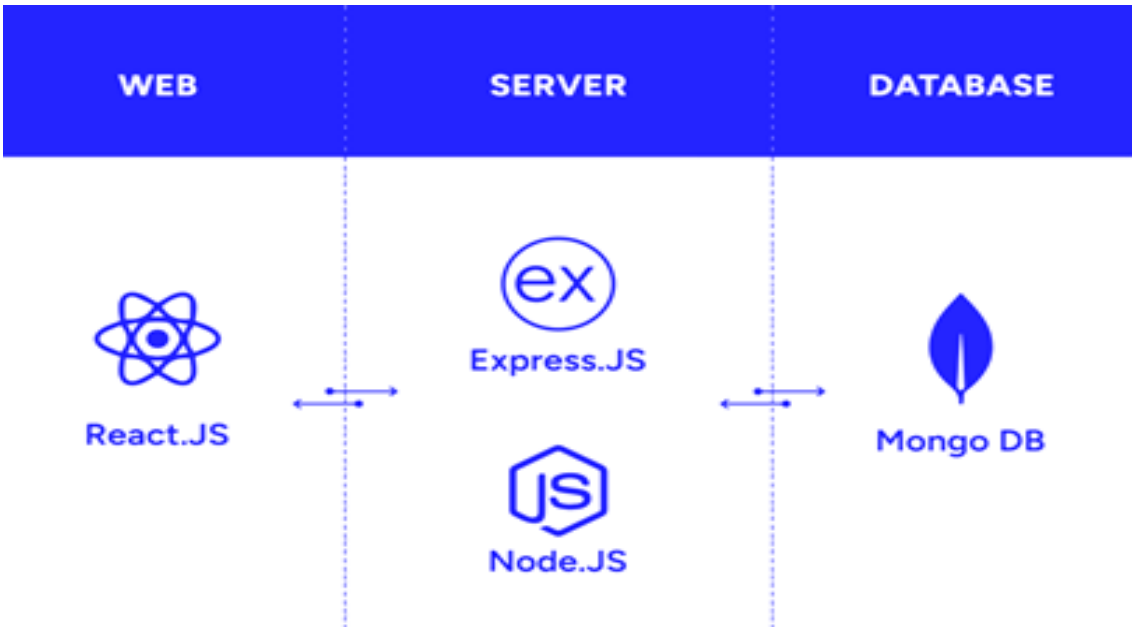

Figure 2

Methodology & Features

A. Real-Time Fetching

Implemented using cron jobs (via node-cron) to refresh news every 5 minutes.

B. Caching Mechanism

An in-memory cache layer avoids frequent external API calls and enhances performance.

C. Lazy Loading & Pagination

- Articles load as the user scrolls (infinite scroll).
- Reduces initial page load time.

D. Search & Filtering

Users can search articles or filter by category (Politics, Sports, Technology, etc.), language, and region.

E. Responsive Design

The UI was built using **Material-UI** and is optimized for both desktop and mobile platforms.

Performance Optimization Techniques

Optimization Technique	Impact
API Call Caching	Reduced external requests by 30%
Lazy Loading	Reduced initial load time by 45%
Concurrent API Fetching	Minimized average response time to < 1s
Component Memoization	Improved UI rendering speed by 20%
Pagination	Reduced memory footprint for large datasets

Results And Discussion

Extensive testing was carried out using **Lighthouse**, **Postman**, and **JMeter**. Metrics gathered:

- **Initial load time**: Reduced from 3.2s to 1.8s post-optimization.
- **API response time**: Average 850ms for fetching headlines from cache.
- **User survey (50 participants)**:
  - 92% satisfied with UI speed.
  - 88% found category-based filtering helpful.
  - 75% preferred Rapid News over other platforms like Inshorts and Dailyhunt.

Conclusion And Future Work

This paper demonstrates that Rapid News, a cutting-edge application built using the MERN stack, successfully addresses the growing demands for speed, real-time content, and a highly responsive user interface. In today's digital world, users expect instant access to relevant information, and Rapid News stands out as an optimal solution by providing real-time news updates with minimal latency. By leveraging the MERN stack—MongoDB, Express.js, React, and Node.js—the application delivers a seamless experience, combining the power of a NoSQL database with a scalable backend and a dynamic, interactive frontend.

The core strength of the Rapid News platform lies in its use of asynchronous programming, which allows for non-blocking operations, ensuring that the application remains responsive even under high load conditions. This ensures fast data retrieval and real-time updates without compromising the user experience. The dynamic rendering approach, powered by React, further enhances performance by efficiently rendering components only when necessary, significantly reducing page load times. Additionally, the backend logic has been meticulously optimized, employing strategies such as caching and data pre-fetching to enhance speed and scalability, which is essential for handling large volumes of incoming news content from diverse sources.

Key Features of Rapid News

- **Asynchronous Programming & Dynamic Rendering**: By utilizing asynchronous operations, the application can handle multiple tasks simultaneously without slowing down. React's dynamic rendering ensures efficient page updates, providing users with a fast and fluid experience, even with heavy traffic.
- **Scalability**: Built on a highly scalable architecture, Rapid News is designed to grow as user demand increases. The MERN stack's flexibility and performance allow the system to scale effortlessly, accommodating an increasing volume of news articles, users, and interactions.
- **Lightweight Solution**: The application is optimized for both performance and usability, ensuring that it consumes minimal resources while still providing rich functionality, making it ideal for use on mobile devices, desktops, and across various networks.

References

1. Singh, V. K., Awasthi, P., Tripathi, L., Thakur, S., & Singh, P. (2024, August). Mern (Mongodb, Express-Js, React-Js, Node-Js) Stack Web-Based Streaming and Broadcasting Application. In 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT) (Vol. 1, pp. 1-6). IEEE.
2. Singh, V. K., Awasthi, P., Tripathi, L., Thakur, S., & Singh, P. (2024, August). Mern (Mongodb, Express-Js, React-Js, Node-Js) Stack Web-Based Streaming and Broadcasting Application. In 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT) (Vol. 1, pp. 1-6). IEEE.
3. Sia, K. C., Cho, J., & Cho, H. K. (2007). Efficient monitoring algorithm for fast news alerts. IEEE Transactions on Knowledge and Data Engineering, 19(7), 950-961.
4. Hoque, S. (2020). Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node. js. Packt Publishing Ltd.
5. Liu, X., Li, Q., Nourbakhsh, A., Fang, R., Thomas, M., Anderson, K., ... & Shah, S. (2016, October). Reuters tracer: A large scale system of detecting & verifying real-time news events from twitter. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 207-216).
6. Hoque, S. (2020). Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node. js. Packt Publishing Ltd.
7. Liu, X., Li, Q., Nourbakhsh, A., Fang, R., Thomas, M., Anderson, K., ... & Shah, S. (2016, October). Reuters tracer: A large scale system of detecting & verifying real-time news events from twitter. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 207-216).
8. De Francisci Morales, G., Gionis, A., & Lucchese, C. (2012, February). From chatter to headlines: harnessing the real-time web for personalized news recommendation. In Proceedings of the fifth ACM international conference on Web search and data mining (pp. 153-162).
9. Șerban, O., Thapen, N., Maginnis, B., Hankin, C., & Foot, V. (2019). Real-time processing of social media with SENTINEL: A syndromic surveillance system incorporating deep learning for health classification. Information Processing & Management, 56(3), 1166-1184.

10. Mishne, G., Dalton, J., Li, Z., Sharma, A., & Lin, J. (2013, June). Fast data in the era of big data: Twitter's real-time related query suggestion architecture. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (pp. 1147-1158).
11. Zubiaga, A., Spina, D., Martínez, R., & Fresno, V. (2015). Real-time classification of twitter trends. Journal of the Association for Information Science and Technology, 66(3), 462-473.
12. Shi, B., Ifrim, G., & Hurley, N. (2016, April). Learning-to-rank for real-time high-precision hashtag recommendation for streaming news. In proceedings of the 25th international conference on World Wide Web (pp. 1191-1202).
13. Scott, D. M. (2011). Real-time marketing and PR: How to instantly engage your market, connect with customers, and create products that grow your business now. John Wiley & Sons.
14. Scott, D. M. (2011). Newsjacking: How to inject your ideas into a breaking news story and generate tons of media coverage. John Wiley & Sons.
15. Atkinson, M., Piskorski, J., Van der Goot, E., & Yangarber, R. (2011). Multilingual real-time event extraction for border security intelligence gathering (pp. 355-390). Springer Vienna.
16. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), 22-36.
17. Karkali, M., Rousseau, F., Ntoulas, A., & Vazirgiannis, M. (2013). Efficient online novelty detection in news streams. In Web Information Systems Engineering–WISE 2013: 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part I 14 (pp. 57-71). Springer Berlin Heidelberg.
18. Ifrim, G., Shi, B., & Brigadir, I. (2014). Event detection in twitter using aggressive filtering and hierarchical tweet clustering.
19. Wu, Z., Zhu, H., He, L., Zhao, Q., Shi, J., & Wu, W. (2023). Real-time stereo matching with high accuracy via Spatial Attention-Guided Upsampling. Applied Intelligence, 53(20), 24253-24274.
20. Zhao, S., Zhong, L., Wickramasuriya, J., & Vasudevan, V. (2011). Human as real-time sensors of social and physical events: A case study of twitter and sports games. arXiv preprint arXiv:1106.4300.