

Full-Stack Web Development for Intelligent User Interfaces: Integration of MERN and Rails

Abid Hussain*

Abstract

In the modern era of web development, by providing users with flexible, responsive, and interactive designs, Intelligent User Interfaces (UIs) greatly improve the user experience. A hybrid method for constructing intelligent, scalable, as well as efficient online apps is discussed in this review work, which focuses on the integration of Ruby on Rails with the MERN stack (MongoDB, React.js, Express.js, Node.js). The MERN stack excels in dynamic frontend development, while Ruby on Rails provides a robust backend framework, offering a powerful combination for building comprehensive full-stack solutions. The study explores the architectural benefits, integration techniques, and real-world applications of these technologies in the context of UIs. Presenting a literature analysis of current works, this study emphasizes important trends and developments in intelligent interface full-stack web development. The study concludes by highlighting how the combination of MERN and Rails might propel user-centric, intelligent web applications.

Keywords: Full-stack web development, intelligent user interfaces (UI), MERN stack, ruby on rails, frontend development, backend APIs, web application architecture, user experience, scalability, real-time interaction

INTRODUCTION

Developing a website in a "full stack" entails working on both the back and front ends of the application. People working in web development are the ones most likely to use this word. As for the back end, which deals with the logic of the appliance, the developers have extensive knowledge of a very specific programming language and experience with front-end programming and user competence. The front-end and back-end parts of a program are referred to as the "full stack" in software or web development. The front-end of a program is its visible and interactive parts. The logic, server, database, as well as other components that users cannot see make up the program's back-end [1]. "Full stack web developer" refers to a person who is proficient in both back-end as well as front-end web development [2]. Full Stack Developers have extensive knowledge of every facet of software development, from the bottom up, like the front end, back end, and internal workings of various operating systems, and the tools and environments needed to code.

*Author for Correspondence

Abid Hussain

E-mail: abid.hussain@cpur.edu.in

Professor, School of Computer Application & Technology &
Dean, Research Career Point University, Kota, Rajasthan, India

Received Date: June 19, 2025

Accepted Date: August 02, 2025

Published Date: September 12, 2025

Citation: Abid Hussain. Full-Stack Web Development for Intelligent User Interfaces: Integration of MERN and Rails. Journal of Web Engineering & Technology. 2025; 12(3): 15–25p.

A comprehensive mapping research that has been conducted in the area of UI during the last 10 years [3]. The light is shed on the present tendencies in the assessment of intelligent interfaces. Some of the topics covered by the current body of research on intelligent UI include adaptive UI, intelligent human-computer interaction, smart, context-sensitive, multimodal UIs, as well as adaptive and adaptable UIs. This research is deemed relevant to the work done as it focuses on UI solutions. However, studies presented at the ACM

International Conference on Intelligent User Interfaces were the only ones considered in the meta-analysis. This work is notable for its systematic assessment and mapping of recent research on IUI, specifically focusing on methodologies for evaluating IUI usability and user experience (UX). It is one of the first of its kind, as far as the author is aware.

Creating a website from the ground up, including the front end, and then deploying it. Every piece of foundational technology used in their capstone project is covered. It is all in the dark when it comes to full-stack development [4]. It has no idea what full-stack developers learn or how the process works. Those who are ignorant of all this may benefit from even a little investigation. Currently, they are familiar with the process and have a good grasp of the technology and how full-stack developers operate. This technology is also the foundation of the project. Their major emphasis is on the JavaScript-based stack, and they are collecting information from many sources while thinking and investigating all these things. One subset of web development, known as "full-stack", includes every step of making a website, whether it is for an internal company network or the public web.

Rapid adoption of agile process techniques and service-oriented designs has resulted from advancements in software engineering; these approaches are well-suited to the creation of Web applications [5]. This study compares as well as contrasts conventional software engineering with the process of creating web applications. It investigates the development of approaches and tools for developing web apps, assess current patterns, as well as provide a rough prediction of future paths.

Structure of the Paper

This study is organized as follows: The next Section covers full-stack technologies like MERN and Rails for intelligent UI. The Section after that explains their integration for scalable and efficient UI development; which is followed by the Section which compares MERN and Rails in IUI development. Then the next Section reviews related literature, highlighting key findings and challenges, while the study concludes with the last Section with insights and future research directions.

OVERVIEW OF FULL-STACK TECHNOLOGIES FOR INTELLIGENT USER INTERFACES

The purpose of implementing a full-stack development architecture is to standardize the technologies that are used to construct an application's front end as well as back end. These frameworks provide developers a suite of tools and libraries to accomplish just that. The benefits of creating end-to-end apps with fewer dependencies became apparent to developers in the early 2000s, and this change started to gather momentum. Developers need these frameworks to be more flexible to create contemporary, dynamic web applications because of their fairly inflexible structure in the beginning. As the complexity of web applications rose along with the desire for interactive, real-time experiences, full-stack solutions emerged as a response to the need for more advanced and flexible frameworks [6]. The emergence of Node.js and the subsequent meteoric climb of JavaScript to the position of dominant programming language hastened this transition, unifying the development process and establishing a framework where both the front- and back-ends could collaborate without any hitches. With Node.js, programmers could use JavaScript in two places: the client as well as the server. A watershed moment in full-stack development's evolution occurred with this union. As seen in Figure 1, outlined the tools and frameworks used by full-stack developers:

The key web development tools: frontend frameworks (React, Angular, Vue.js, Svelte Kit, Nuxt.js, Strapi), backend frameworks (Node.js, Express, Next.js, Gatsby, Sails.js, Meteor), and databases (MongoDB, MySQL, SQLite), as shown in Figure 1, represent a full-stack development ecosystem.

Traditional Technologies in Web Development

Technologies seen on the web are often categorized based on the setting in which they manifest. So, are they more concerned with the server side or the client side of web applications? Familiarity with the main lean engineering categories is essential for anybody hoping to contribute to lean development. As a result, below is a list of web development techniques organized by category [7, 8].



Figure 1. Overview of technologies used in full-stack development.

Using a Browser

A browser (e.g., Chrome, Firefox, Edge) is used to access and view websites by interpreting HTML, CSS, and JavaScript files from web servers.

HTML and CSS

Web pages are often created using Hypertext Markup Language (HTML), which organizes elements such as text, graphics, and links. To design and arrange HTML components (such as colors, fonts, spacing, and responsiveness), one uses CSS, which stands for Cascading Style Sheets. Programming language: Designed to enhance online applications with interaction, logic, as well as server-side processing. Common languages include: JavaScript (client-side), Python, PHP, Ruby, Java, and Node.js (server-side).

Profile

- *User Profile:* A personal account on a website, typically contains user-specific data (e.g., name, email, preferences).
- *Developer Profile:* Information about a web developer, such as skills, portfolio, and experience.

Web Server

A system (hardware/software) that hosts websites and delivers web content to browsers via HTTP/HTTPS.

MERN Stack Components

Data in MongoDB is stored in a document-oriented manner, making it a NoSQL database [9]. Express.js, a back-end web application framework, provides capabilities for creating RESTful APIs. To build interactive UI, developers may utilize the React library, which is located on the front end. An environment for running JavaScript, using Node.js, developers may create server-side JavaScript code [10].

Express.js, a popular Node.js web framework, simplifies the method of developing online applications. Some key considerations while dealing with Express.js are as follows:

- *Routing:* The Express.js application programming interface (API) makes it simple to define routes that handle individual HTTP requests. The URL, HTTP method, and other parameters may be used to tailor answers to individual requests.
- *Middleware:* Middleware methods in Express.js allow to modify the request as well as response objects prior to their processing by the route handler. Tasks like error management, logging, and authentication may be accomplished using this.

- *Template engines:* A range of template engines are supported by Express.js, enabling to display dynamic HTML pages using server-side data. Express.js supports a number of well-known template engines, such as EJS, Handlebars, as well as Pug.
- *Error handling:* Express.js has an inbuilt error handling system that lets handle problems that come up while processing requests. When an issue occurs, it may configure error-handling middleware to be called, enabling to record the issue, provide the client with an error message, or take other actions.

Ruby on Rails Framework

One framework for creating web apps is Ruby on Rails. This framework is popular among developers for backend development, and it is written in the Ruby programming language. Software engineers find online application development more laborious; to make it less laborious and more adaptable, web development specialists use ROR technology. One of the most effective and well-liked frameworks for creating dynamic web applications is Ruby on Rails [11]. The Ruby on Rails platform is one open-source tool for developing web applications. The course is structured on the Model-Controller-View paradigm. With ROR, building a database-backed web application using the Ruby programming language is a breeze.

INTEGRATION OF MERN AND RAILS FOR INTELLIGENT USER INTERFACES

A hybrid approach to full-stack web development is shown via the combination of the MERN stack (MongoDB, Node.js, React.js, Express.js) with Ruby on Rails, particularly beneficial for building IUI. While MERN is highly efficient for creating dynamic and responsive frontend experiences, Rails excels in rapid backend development with its convention-over-configuration philosophy. By leveraging Rails for building robust APIs and handling server-side logic, and using React from the MERN stack for developing rich, component-based UI, developers can harness the best of both ecosystems [12]. This integration enables scalable data handling, real-time user interactions, and seamless state management, essential for intelligent applications that rely on AI-driven personalization, recommendation systems, and user adaptability. Furthermore, using Rails as a backend API service with MERN frontend promotes modularity, making applications easier to maintain and extend. However, achieving efficient communication between these technologies requires careful consideration of architectural patterns, API design, and performance optimization. The combined use of MERN and Rails offers a flexible and powerful solution for developing modern IUI in complex web applications.

MERN Stacks: MongoDB, Express.js, React, and Node.js

The MERN stack is the most popular and capable technology stack for constructing web applications. M is the MongoDB MER form. R stands for ReactJS in Express, and N for NodeJS. One example of a database system that does not use SQL is MongoDB, as seen in Figure 2. The primary use is in the realm of record data gathering and planning. The letter "E" represents Express, a popular framework for developing Node.js web applications [13].

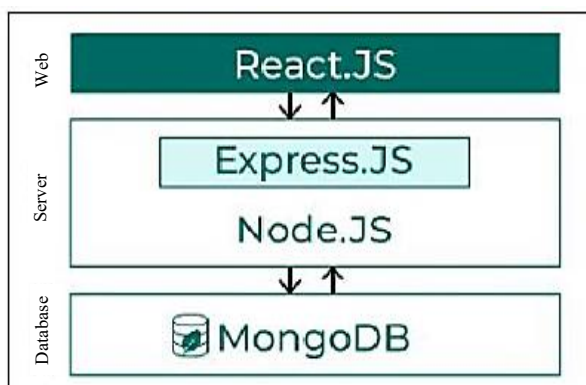


Figure 2. MERN Stack works.

R stands for React, a popular library for client-side JavaScript systems. When it comes to N, the main JavaScript web server is where it is at. A MERN's architectural structure consists of three tiers. The web is at the top of a stack, servers are in the center, and the databases are at the bottom.

These are the stacks of MERN, including Node.js, Express.js, MongoDB, and React, as follows:

Node.js

JavaScript scripts may run in an open-source, platform-neutral environment using the Node.js framework. It is widely recognized as the preferred tool for almost all tasks. The V8 JavaScript engine, which is necessary for Google Chrome, is executed remotely via Node.js [14]. Blocking Behavior is uncommon since most Node.js modules are intended to be non-obstructive [15]. There are many nonconcurrent I/O natives in the Node.js standard library that stop JavaScript code from interfering. By skipping Node.js's input/output operations as well as restarting when the reaction is complete, the proposed solution reduces needless string processing. Node.js has the advantage of removing the need for front-end programmers to learn new techniques while developing server-side code, whether the work is meant for the application or the client.

MongoDB

A NoSQL database known as MongoDB stores data in a document-oriented fashion. Express.js is a framework for back-end web applications that provides a set of tools for creating RESTful APIs [16]. Coders may create dynamic UI with the help of the front-end library React. A framework for building server-based JavaScript applications is called Node.js.

- MongoDB stores data in documents that resemble JSON since it is a document-oriented database. This system is incredibly flexible and does not have any schemas.
- MongoDB is schema-less, meaning that data may be entered without a schema and modified as necessary.
- MongoDB's powerful querying tools and proprietary query language are especially useful for data saved in documents.
- NoSQL databases designed for horizontal scalability, such as MongoDB, make it easier to distribute data across several servers.

Express.js

The optimized and regular nature of server-side programming facilitates the development of APIs as well as the processing of requests and responses. This is very advantageous for developing and executing web server-based applications using Node.js. It has several useful features for handling web development tasks, including handling requests, responses, as well as URLs.

- *Routing*: A straightforward API for building routes that would function with certain HTTP requests is also provided by the Express.js framework. Certain circumstances, such as the URL and HTTP method, may determine how to handle requests and respond.
- *Middleware*: In the Express.js framework, middleware methods may be used to alter both requests as well as responses objects before the route handler can execute them.
- *Template engines*: Making updates to HTML webpages in real time could be possible using one of the many template engines that work with Express.js.
- *Error handling*: Issues that occur during request processing may be handled by using Express.js built-in error handling mechanism.

Server-side Dependencies

The libraries, tools, frameworks, and services that the backend of a web application needs in order to operate correctly are known as server-side dependencies. These dependencies help with routing, database interactions, authentication, API integration, and more.

- *MongoDB*: The Node.js driver enables asynchronous data access and processing in Node.js applications by connecting to MongoDB, API that uses traditional callbacks or promises for communication [17].

- *Mongoose*: As an object-oriented JavaScript development tool, Mongoose bridges the gap between MongoDB and the Node.js runtime environment for JavaScript.
- *Node mailer*: Node mailer is a Node.js utility that simplifies email sending. Enabling embedding Unicode attachments and including SMTP as well as OAuth2 authentication for safety, this single-module solution does not rely on any other components.
- *No demon*: The node application is automatically restarted anytime a directory file change is detected, making it easier to create Node.js-based applications. The node wrapper is replaced by it.
- *JSON Web Token*: The JSON Web Token (JWT) is a concise and secure method of encoding claims that are necessary for a two-party transaction to take place. It may think of the claims in a JWT as JSON objects. This may be used as the payload for a JSON Web Encryption (JWE) structure or a JSON Web Signature (JWS) framework, which enables the claims to be digitally signed, encrypted, or integrity secured using a Message Authentication Code (MAC).
- *Google APIs*: A library called (Google APIs Node.js Client) allows Node.js to use Google APIs. The endpoints of the API are automatically created, JWT tokens, API keys, as well as OAuth 2.0 are supported for authorization and authentication. They utilize Google APIs and OAuth2 playground to send emails to recently enrolled users for verification purposes.

Ruby on Rails

The main component of this stack is Ruby on Rails, or simply Rails. The Ruby web application framework known as Rails follows the CoC principle. A methodology for creating software suggests using a predetermined set of rules and assumptions to save time, as well as the DRY concept promotes simplicity and productivity above repetition in coding principles [18, 19]. Database interaction is made easier with Active Record, which acts as an Object-Relational Mapping tool. Frontend development uses JavaScript, HTML, and CSS. The Rails framework is praised for its convention-driven development as well as developer-friendly methodology.

COMPARATIVE ANALYSIS AND APPLICATIONS IN INTELLIGENT UI DEVELOPMENT

Figure 3 displays the consumption rate of each popular UI design tool, as measured by a questionnaire administered before to the experiment. Designers, both professional and amateur, rely on Sketch, Adobe XD, and Figma as their go-to user interface design programs [20]. These three applications serve as the experimental case study [21]. As the first point of contact between the program and its users, the interface has a major influence on the likelihood that users will continue to use and buy the product. As a result, interface design is a crucial part of software development.

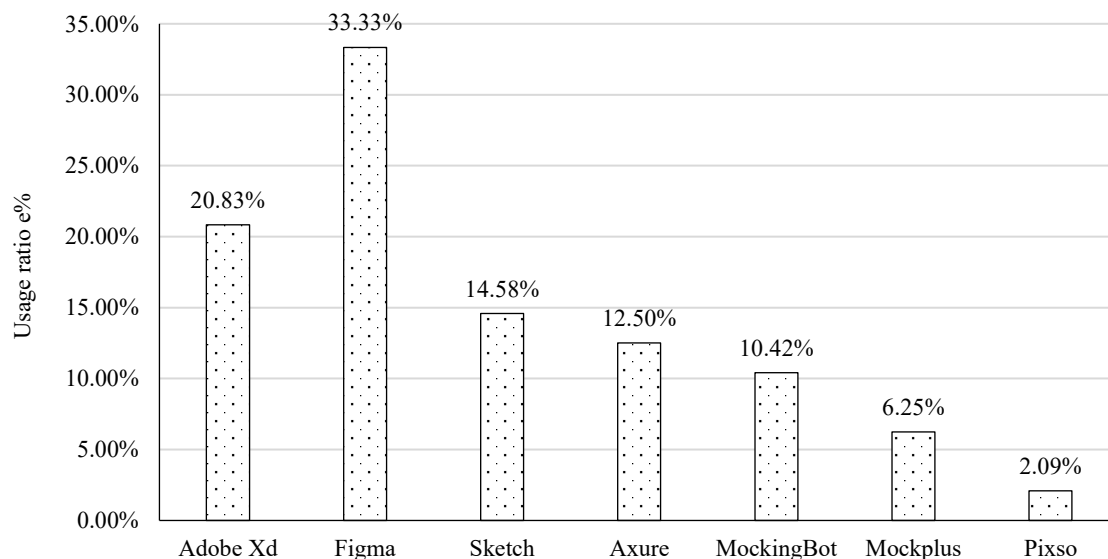


Figure 3. Usage Rate of UI Design Software.

The goal of this research is to analyze all three programs side-by-side, compare their layout and interaction, and draw conclusions on how each affects the software's usability and the user's overall experience.

Tools for Creating User Interfaces and Other Essentials

Here are the main tools for user-intelligent interfaces as given below:

- *Figma*: Figma is a prototype tool and online vector graphics editor; its main selling point is that it is accessible from any computer with an internet connection. Figma is a powerful tool for creating vector graphics, UI for websites, and more. Because of its adaptability and simplicity, Figma is a great tool for beginners. Working together in real-time is a breeze with Figma. The best thing about Figma is that it is a free UI tool for creation, collaboration, handoff, as well as prototyping. "The Figma Mirror" companion app, which is compatible with both iOS and Android, is one of its greatest advantages. It allows users to see Figma prototypes on their phones in real time [22].
- *Adobe XD*: A tool for creating user experiences that is entirely vector-based is Adobe XD. Adobe XD facilitates the design process for both mobile and online applications. Adobe Inc. designed and published it entirely, from designing the tiny features of the interface, such as the typeface and iconography, to achieving the ideal layout. Adobe XD claims to be the greatest tool available for all UI designers. Additionally, Adobe XD makes it simple to retrieve and arrange elements with other Creative Cloud applications, such as Photoshop as well as Illustrator [23].
- *Sketch*: Sketch is another vector graphics tool that helps designers create interfaces and designs for current UIs as well as user experiences. However, Sketch is limited to macOS. Sketch received a 2012 Apple Design Award and was made public on September 7. Designing UI and user experiences for websites and mobile apps is the primary usage of Sketch. Due to the usage of macOS-specific frameworks as well as methods, it is only accessible on macOS.

Challenges in Developing Intelligent User Interfaces

Making IUIs functional, practical, and reliable is the primary objective. The three primary issues: presentation, competence, and trust, are in line with this. The presentation will focus on the IUIs' human-computer interface, while the competency will cover the relevant AI [24]. Since users will not give tasks to an IUI they do not trust, IUI creation must focus especially on trust. Otherwise, the IUI would be useless [25]. IUIs, on the other hand, are far more difficult to make users trust the system than traditional UI because they use AI methods that often have unpredictable outcomes, making users feel less in control of the system. They list the difficulties that must be overcome when creating user-adaptive IUIs for each of these problems and provide potential solutions in Table 1. Moreover, Table 1 provides a summary of the issues that have been found. The issues should only highlight a few key areas for developing user-adaptive IUI since they are interconnected as well as not disjunctive. It should be noted that not all IUIs will face all of the obstacles listed below in the Table 1. For instance, Collaborative filtering systems are not always tasked with handling limited-use data.

Analysis of AI-Powered UX/UI Design Platforms

To truly evaluate the real-world effectiveness of the various AI-assisted UX/UI tools, a comparative analysis of leading tools is warranted [26]. This section does just that, comparing leading platforms according to a selection of core criteria critical to design professionals [27]. These criteria are: automation features, ease of use, customization options, and integration capabilities. Factors like these provide valuable insight into the strengths and weaknesses between platforms and allow the determination of the most efficient platforms for design workflows. The platforms selected for comparison encompass a rich variety of approaches to AI-assisted UX/UI design that cater to distinct user segments and design workflows. These are: Figma (AI-powered), Adobe Sensei, Uizard, Framer, and Canva (AI-driven). Each platform presents its unique array of features, advantages, and limitations in this evolving landscape of AI for design.

Table 1. Challenges in developing user-adaptive UIs.

Presentation	Designing interactions Unobtrusiveness Adaptivity
Competence	Limited use data and accuracy in modifying user behavior
Trust	Controllable behavior. Intelligibility Privacy

LITERATURE REVIEW

This literature examines developments in web-based applications, UI design, and full-stack development, with a focus on security models, performance optimization, emotion-aware interfaces, as well as data-driven decision-making for improved system stability and user experience.

Singh [7]

Singh explored developments in full-stack development frameworks, focusing on their security and compliance models to tackle contemporary issues, including regulatory non-compliance, data breaches, and unauthorized access. The research examines well-known full-stack frameworks and highlights their built-in security features, like Django, MERN (Express.js, Node.js, React, MongoDB), as well as MEAN (Angular, Node.js, MongoDB, Express.js). These include strong encryption techniques, authentication procedures, along with safeguarding against common vulnerabilities such as SQL injection, cross-site scripting, as well as cross-site request forgery. Additionally, the analysis explores compliance models that are included in these frameworks, emphasizing how well they conform to international requirements that regulate data security and privacy, such as GDPR, HIPAA, and PCI DSS [7].

Ajah *et al.* [28]

Full-stack web development technology can be applied to achieve this purpose. This will show that the load flow analysis of any electrical power system can be done online and in real time, with a user friendly, multi-platform portable environment for improved electrical power system management. The electrical power system is essential for sustainable development. The management and maintenance of quality electrical power delivery is very important and requires constant system data analysis and application of analytical results [28].

Supriya *et al.* [29]

The goal of the project is to create an all-inclusive platform that will make organizing assignments and learning resources easier, as well as student cooperation, ensuring scalability and performance at the same time. This project makes extensive use of software engineering techniques to improve code efficiency, security, and maintainability. Testing and iterative system modification based on end users' feedback comprise the whole software development cycle. In the end, the LMS is enhanced via extensive testing and user input in a progressive and iterative manner. The results of this study are relevant to upcoming advancements in educational technology that aim to provide cutting-edge, feature-rich, and intuitive learning management systems [29].

Duan *et al.* [30]

The increasing demand involves implementing a sophisticated emotion identification technology to greatly enhance UI's emotional reactivity. The system reads complex emotional indicators in real-time by combining textual data, facial expressions, and voice using a multi-branch Transformer model. This allows UIs to communicate with users more efficiently and sympathetically. Since their model outperforms conventional techniques in terms of emotion identification accuracy and F1 scores using the publicly available MELD dataset for validation, the significance of user interface (UI) design cannot be emphasized. In addition to improving usability, a well-designed UI encourages more organic, intuitive, and emotionally appealing experiences, expanding technology's accessibility as well as impact on everyday life [30].

Chaudhary and Mittal [31]

The growing popularity of the MERN stack in web development, the performance, scalability, security and flexibility become important when selecting a database. The database must be secure, fast

and scalable. MongoDB has a schemaless architecture. Its document-oriented NoSQL database fits nicely with the dynamic and fluid design that is needed in the MERN applications. The goal of the work is to show through examinations how MongoDB offers the best solution in MERN-based applications and how it is the best possible database for efficient data storage [31].

Stefanidi *et al.* [32]

UIs are meant to increase users' Situational Awareness (SA) by exploiting the current context as well as presenting the most useful data in the most efficient manner feasible. Combinatorial optimization, ontology modelling, and reasoning are used to ascertain what information should be provided, when it should be presented, and where it should be visualized on the display, all while keeping contextual factors and placement constraints in mind. In order to prevent stress and data overload, the main goal of the suggested strategy is to optimize the SA related to the user interface that is displayed during runtime [32].

Table 2 provides an overview of the literature on Full-Stack Web Development for IUI, including the research, methodology, main conclusions, difficulties, and possibilities for the future.

Table 2. Literature review on full-stack web development for intelligent user interfaces.

Author	Study On	Approach	Key Findings	Challenges	Future Directions
Singh (2020) [7]	Developments in full-stack development frameworks that prioritize compliance and security concepts	Comparative review of MEAN, MERN, and Django frameworks, analyzing built-in security and compliance	Identified robust features like authentication, encryption, SQL injection, and XSS/CSRF defense; highlighted compliance with GDPR, HIPAA, and PCI DSS	Handling dynamic and evolving threat landscapes, ensuring real-time compliance with changing regulations	Enhance AI-driven threat detection in frameworks; integrate automated compliance auditing mechanisms
Ajah <i>et al.</i> (2024) [28]	Real-time Load Flow Analysis of Electrical Power Systems	Full-Stack Web-based system for online and real-time power system analysis	Provides a user-friendly, multi-platform environment for efficient power system management	Real-time large-scale data processing and analysis	Enhance system sustainability and management through web technologies
Radhakrishnan <i>et al.</i> (2024) [29]	Development of Learning Management System (LMS)	Iterative development with continuous user feedback using Full-Stack technologies	Improves LMS usability, scalability, and performance through feedback-driven enhancements	Maintaining scalability, security, and code quality	Design modern LMS platforms with advanced features and user-centric design
Duan <i>et al.</i> (2024) [30]	Emotion Recognition for IUI	Multi-branch Transformer integrating facial, speech, and text data for real-time analysis	Significantly improves emotion recognition accuracy and enhances user experience	Handling complex multi-modal emotional data in real-time	Expand real-time emotion-aware UIs for broader applications
Chaudhary and Mittal (2024) [31]	MongoDB's Role in MERN Applications	Evaluation of MongoDB's performance, scalability, and flexibility in MERN stack	MongoDB's schemaless NoSQL database offers best-fit solution for dynamic MERN applications	Ensuring database security, performance under high loads	Optimize MongoDB for large-scale MERN implementations
Stefanidi <i>et al.</i> (2022) [32]	Situational Awareness in UI	Ontology modeling and combinatorial optimization for context-aware UI design	Enhances user Situational Awareness by optimizing information presentation	Managing information overload and contextual placement constraints	Develop adaptive, context-aware UIs for stress-free interaction

CONCLUSION AND FUTURE WORK

This review work has investigated full stack web development in the context of creating UIs by integrating Ruby on Rails with the MERN stack. By analyzing the strengths of MERN in building dynamic frontends and Rails in robust backend API development, their combination clearly provides a strong and adaptable solution for contemporary web applications. The synergy of these technologies enables scalable, modular, and user-centric systems that enhance interactivity and user experience. Furthermore, the study highlights the relevance of UIs in improving usability, adaptability, and engagement across various applications, making full-stack development an essential approach in the evolving digital landscape. However, the integration of two distinct technology stacks may lead to increased system complexity and potential performance bottlenecks. Additionally, ensuring seamless compatibility and managing version conflicts between MERN and Rails components remains a challenging limitation.

Future studies should concentrate on improving MERN as well as Rails' data synchronization and communication in large-scale systems, with a focus on AI-driven user adaptation as well as real-time responsiveness. To further improve intelligent user interface capabilities, it is also necessary to investigate the integration of cutting-edge technologies like Edge Computing, ML, as well as Progressive Web Apps (PWAs) inside this hybrid stack. Addressing security, scalability, and cross-platform compatibility will also be critical in advancing the development of more intuitive and intelligent web applications.

REFERENCES

1. Singh G, Javed M, Dhaliwal BK. Full stack web development: Vision challenges and future scope. *Int Res J Eng Technol*. 2022 Apr; 9(04): 3083–3089.
2. Sacks M. Web Testing Practices. In *Pro Website Development and Operations: Streamlining DevOps for Large-Scale Websites*. Berkeley, CA: Apress; 2012; 27–43.
3. Brdnik S, Heričko T, Šumak B. Intelligent user interfaces and their evaluation: a systematic mapping study. *Sensors*. 2022 Aug 4; 22(15): 5830.
4. Baiskar Y, Paulzagade P, Koradia K, Ingole P, Shirbhate D. MERN: A Full-Stack Development. *Int J Res Appl Sci Eng Technol*. 2022 Jan; 10(1): 1029–35.
5. Jazayeri M. Some trends in web application development. In *IEEE Future of Software Engineering (FOSE'07)*. 2007 May 23; 199–213.
6. Shethiya AS. Scalability and Performance Optimization in Web Application Development. *Integrated Journal of Science and Technology (IJST)*. 2025 Jan 22; 2(1): 1–7.
7. Singh B. Automating Security Testing in CI/CD Pipelines using DevSecOps Tools a Comprehensive Study. *Science, Technology and Development*. 2020 Dec; 9(12): 631–651.
8. Petrikoglou A, Kaskalis TH. Full Stack Web Development Teaching: Current Status and a New Proposal. In *WEBIST*. 2019 Sep 18; 218–225.
9. Soni M, Jyotinagar V. Sql vs nosql databases for the microservices: A comparative survey. In *2023 IEEE 2nd International Conference on Edge Computing and Applications (ICECAA)*. 2023 Jul 19; 17–22.
10. Kadam P, Gopani A, Mattoo S, Gupta S, Amrutkar D, Dhanke J, Kadam Y. Introduction to MERN stack & comparison with previous technologies. *Eur Chem Bull*. 2023 Jun; 12(4): 14382–6.
11. Nittur K, Chippagiri S, Zhidko M. Evolving Web Application Development Frameworks: A Survey of Ruby on Rails, Python, and Cloud-Based Architectures. *International Journal of New Media Studies (IJNMS)*. 2020; 7(1): 28–34.
12. Patel D. The Role of Amazon Web Services in Modern Cloud Architecture: Key Strategies for Scalable Deployment and Integration. *Asian J Comput Sci Eng*. 2024; 9(4): 1–9.
13. Thokala VS. Enhancing user experience with dynamic forms and real-time feedback in web applications using MERN and Rails. *Int J Res Anal Rev*. 2023; 10(3): 87–93.
14. Gupta S, Agrawal N, Gupta S. A review on search engine optimization: Basics. *Int J Hybrid Inf Technol*. 2016 May; 9(5): 381–90.

15. Jadhav G, Gonsalves F. Role of Node. js in modern web application development. *Int Res J Eng Technol*. 2020 Jun; 7(06): 6145–50.
16. Györödi CA, Dumșe-Burescu DV, Zmaranda DR, Györödi RŞ. A comparative study of MongoDB and document-based MySQL for big data application data management. *Big Data Cogn Comput*. 2022 May 5; 6(2): 49.
17. Desai K, Fiaidhi J. Developing a Social Platform using MERN Stack. *TechRxiv Preprints*. 2023 Oct 30; 1–10.
18. Aggarwal S, Verma J. Comparative analysis of MEAN stack and MERN stack. *Int J Recent Res Asp*. 2018 Mar 1; 5(1): 127–132.
19. Kalava SP. Enhancing Software Development with AI-Driven Code Reviews. *North American Journal of Engineering Research (NAJER)*. 2024 Apr 13; 5(2): 1–4.
20. Soni A, Kumar A, Arora R, Garine R. Integrating AI into the Software Development Life Cycle: Best Practices, Tools, and Impact Analysis. *Tools, and Impact Analysis*. Available at SSRN. 2023 Jun 10.
21. Wang J, Xu Z, Wang X, Lu J. A comparative research on usability and user experience of user interface design software. *Int J Adv Comput Sci Appl*. 2022; 13(8): 21–29.
22. Sharma V, Tiwari AK. A study on user interface and user experience designs and its tools. *World J Res Rev*. 2021 Jun; 12(6): 41–5.
23. Murri S, Chinta S, Jain S, Adimulam T. Advancing Cloud Data Architectures: A Deep Dive into Scalability, Security, and Intelligent Data Management for Next-Generation Applications. *Well Testing Journal*. 2024 Dec 29; 33(S2): 619–44.
24. Hartmann M. Challenges in developing user-adaptive intelligent user interfaces. In *LWA*. 2009 Jan; ABIS-6.
25. Goyal A. Optimising Software Lifecycle Management through Predictive Maintenance: Insights and Best Practices. *Int J Sci Res Arch*. 2022; 7(02): 693–702.
26. Modalavalasa G, Bheri SP. Next-Generation AI-Powered Automation for Streamlining Business Processes and Improving Operational Efficiency. *Journal of Computing Technologies (JCT)*. 2023 Dec; 12(12): 1–7.
27. Fountain L. *Ethics and the Internal Auditor's Political Dilemma: Tools and Techniques to Evaluate a Company's Ethical Culture*. New York: Auerbach Publications; 2016 Dec 12.
28. Ajah NG, Uchenyi K, Kesandu-Uchenyi O, Ejiogu EC. Application of Full Stack Web Development Technology in Load Flow Analysis for Real Time Electrical Power System Management. In *2024 IEEE International Conference on Power and Energy (PECon)*. 2024 Nov 4; 128–131.
29. Supriya M, Rathinavel K, Radhakrishnan D. Development of a Comprehensive Learning Management System using Full Stack Technology: A Modern Approach to Educational Platforms. In *IEEE 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. 2024 Nov 18; 1426–1431.
30. Duan S, Wang Z, Wang S, Chen M, Zhang R. Emotion-aware interaction design in intelligent user interface using multi-modal deep learning. In *2024 IEEE 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*. 2024 Nov 8; 110–114.
31. Chaudhary N, Mittal N. Leveraging Mongo DB for Efficient Data storage in MERN. In *2024 IEEE 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 2024 Mar 14; 1–6.
32. Stefanidi Z, Margetis G, Ntoa S, Papagiannakis G. Real-time adaptation of context-aware intelligent user interfaces, for enhanced situational awareness. *IEEE Access*. 2022 Feb 18; 10: 23367–93.