

ERP Web-Application using NodeJs, Angular

More Aadesh, Dhiwar Kavita, Malge Dhruvdeep, Bedse Mrugnayani

Matoshri College of Engineering and Research Centre (MCOERC), Nashik, Maharashtra, India

Abstract: *Implementing an Enterprise Resource Planning (ERP) system in a company can bring many benefits. However, ERP implementations have been high spending projects with low success rates. International ERP vendors are now trying to extend their market to companies in developing countries. However, little research has been conducted on ERP implementation in developing countries. So, understanding the critical success factors (CSFs) involved in ERP implementation would be of benefit to implementing companies and ERP software vendors. Thus, this research developed and empirically tested a model for ERP implementation success in the context of a developing country, namely Iran. A survey questionnaire was employed to collect data for this research. The structural equation modelling (SEM) was utilized to test the relationships hypothesized. It was found that the relationships between enterprise-wide communication, project management, ERP team composition and competence, ERP system quality, and ERP vendor support and ERP implementation success were significantly positive. Furthermore, organizational culture indicated a moderating effect on the above-mentioned relationships. However, the study did not find significant relationship between business process reengineering and ERP implementation success. This study has contributed to academic research by creating the empirical evidence to support the theories of CSFs and ERP implementation success. In addition, the outcomes of this study are useful in making ERP vendors and consultants familiar with the difficulties of implementation in developing countries and preparing some strategies to overcome the barriers. The results also present the managers of adopting companies with the ability to classify strategies, evaluation guides, checkpoints, and measure requirements that offer them a far greater likelihood of ERP implementation success*

Keywords: Critical Success Factors

I. INTRODUCTION

Node.js is a non-blocking (non-blocking), event-driven (Event-based) I/O platform built on Google Chrome's v8 engine, with event-driven as its core. Its biggest feature is the use of asynchronous I/O and event-driven architecture design. For high-concurrency solutions, the traditional architecture is a multi-threaded model, that is, a system thread is provided for each business logic, and system thread switching is used to make up for the time overhead of synchronous I/O calls. Node.js uses a single-threaded model and uses asynchronous request methods for all I/O, avoiding frequent context switching. Its unique event loop mechanism uses the processing advantages of call back functions to maximize the efficiency of task access. These unique construction advantages can be that node.js can handle more than 40,000 user connections at the same time on a server with 8GB memory. In traditional server-side languages such as PHP, Java or .NET, a new thread will be used for each client in a conventional way. A significant challenge for web programmers is to ensure data can be correctly understood across many different platforms. For example, are different databases using terms like "name" or "employee" in the same way—and, if not, how best to translate between them? One solution to this problem is Linked Data, which provides context to a term like "name" by attaching to "name" a web address that points to a formal definition of "name," so the term's meaning is unambiguous. JSON-LD is a W3C standard that allows developers to attach linked data to JSON objects. (JSON is a data interchange format that is often used in web programming.) The JSON-LD specification provides algorithms for several operations on JSON data with respect to a linked data context. There are official software libraries for these algorithms available in several programming languages, including JavaScript. However, this JavaScript library is written in "common" JavaScript, and does not take advantage of powerful libraries and techniques that have recently become available. Over the last five years, the open-source community has put in a tremendous amount of effort to make JavaScript a safer, more mature language. Two

open-source projects in that effort are Typescript and in versify. Typescript, backed by Microsoft, adds static types to JavaScript. Type script is now extremely popular among web developers, because type checking helps push runtime errors back to compile. time, and makes maintaining large projects easier. In versify is less-known than Typescript, but used by many companies with large-scale Java script projects. Perhaps In versify greatest strength is that it provides Java script and Type script programmers with a mechanism for Inversion of Control. Inversion of Control has become a popular design principle in web programming, in part because it focuses on the importance of contracts between modules, without having to know anything about the inner workings of a module.

II. EASE OF USE

EASE OF USE of ERP system using NodeJS and angular

The combination of Node.js and Angular can provide a powerful and user-friendly ERP system. Here are some of the key factors that contribute to the ease of use of an ERP system built with Node.js and Angular:

- User-friendly interface: Angular provides a powerful framework for developing user-friendly and intuitive interfaces. The use of components and templates can make it easy to create a consistent and intuitive user experience.
- Real-time updates: Node.js is well-suited for real-time applications, which can provide users with up-to-date information on business processes and operations. Real-time updates can improve user productivity and help them make more informed decisions.
- Cross-platform compatibility: Both Node.js and Angular are cross-platform technologies, which means that the ERP system can be accessed from any device or platform with a web browser. This can improve the accessibility and usability of the system for users who work from multiple devices.
- Scalability: Node.js is designed to be scalable, which means that the ERP system can grow and adapt to changing business needs without sacrificing performance. This can make it easier for users to work with the system as their organization grows and evolves.
- Modular design: Angular provides a modular design, which can make it easy to add or remove features and functionality as needed. This can make the system more flexible and adaptable to specific business needs.

Overall, the combination of Node.js and Angular can provide a powerful and user-friendly ERP system that can streamline business processes, improve productivity, and enhance overall operations.

III. OBSEVATIONS

Developing an ERP system using Node.js and Angular can provide a range of outcomes and observations, including:

- Increased performance: Node.js allows for high-performance server-side applications, while Angular's reactive approach enables fast rendering and efficient updates of user interfaces. This can result in a faster and more responsive ERP system.
- Improved scalability: Node.js provides a scalable architecture that can handle a large number of concurrent requests, making it ideal for ERP systems that need to handle a high volume of data and users.
- Modular design: Both Node.js and Angular have a modular design, which allows developers to break down an ERP system into smaller, more manageable components. This makes it easier to maintain and update the system over time.
- Enhanced user experience: Angular's template-based approach allows developers to create dynamic and responsive user interfaces, making it easier for users to interact with the ERP system. Additionally, Node.js allows for real-time communication, which can improve collaboration and decision-making within an organization.
- Simplified development: Using Node.js and Angular together can simplify the development process by enabling developers to write both the server-side and client-side code in a single language. This reduces the need for context switching between different technologies and can speed up development time.

Overall, developing an ERP system using Node.js and Angular can provide a range of benefits, including improved performance, scalability, modular design, enhanced user experience, and simplified development. However, it is

important to consider the specific needs of the organization and the resources available when deciding whether to use these technologies in an ERP system

IV. ANALYSIS

Using Node.js and Angular in an ERP system can offer several advantages, such as:

- **Scalability:** Node.js can handle high levels of concurrent requests, making it an ideal choice for building scalable applications. In an ERP system, where multiple users may be accessing the system at the same time, Node.js can help ensure smooth performance.
- **Speed:** Both Node.js and Angular have a reputation for being fast and efficient technologies. By using these technologies, an ERP system can deliver a responsive, high-performing user experience.
- **Modular Design:** Angular's modular design, combined with Node.js's ability to handle packages and dependencies, can help make an ERP system more manageable and easier to maintain. Developers can break down the system into smaller components that can be updated and improved separately.
- **Real-Time Communication:** Node.js has built-in support for real-time communication, which can be useful in an ERP system where multiple users may need to collaborate on tasks or communicate with each other. This can help to streamline workflows and reduce the need for manual data entry.
- **Easy to Learn:** Both Node.js and Angular are relatively easy to learn and can be quickly adopted by developers. This can be particularly beneficial for organizations that may need to onboard new developers to the project.

However, there are some potential challenges to using Node.js and Angular in an ERP system, including:

- **Complexity:** Building an ERP system can be a complex undertaking, and using Node.js and Angular together can add an extra layer of complexity. Developers must have a strong understanding of both technologies and how they interact.
- **Security:** As with any web application, security is a significant concern. Developers must take extra care to ensure that the system is secure, particularly when handling sensitive data such as financial or personal information.
- **Integration:** An ERP system must integrate with various other systems, such as accounting software, human resources systems, and inventory management software. Developers must ensure that the system can effectively integrate with these other systems to provide a seamless experience for users.

Overall, using Node.js and Angular in an ERP system can offer many benefits, but it is crucial to weigh the advantages against the potential challenges and ensure that the chosen technologies meet the specific needs of the organization.

V. RESULTS AND DISCUSSION

The outcomes of an ERP system web application can vary depending on the specific goals and objectives of the project. However, here are some common outcomes that organizations aim to achieve:

- **Improved Efficiency:** ERP system web applications can help organizations streamline their business processes by automating tasks and reducing the need for manual intervention. This can result in improved productivity and faster turnaround times.
- **Increased Transparency:** With an ERP system web application, all stakeholders can access the same data in real-time, which can lead to improved decision-making and greater transparency.
- **Better Resource Management:** ERP systems can help organizations optimize their use of resources, such as inventory, personnel, and equipment. This can result in cost savings and improved resource utilization.
- **Enhanced Customer Service:** With a centralized system for managing customer data, organizations can improve their ability to provide personalized and efficient customer service.
- **Improved Data Security:** An ERP system web application can provide robust security features to protect sensitive business data from unauthorized access or cyber-attacks.

- **Better Financial Management:** With an integrated financial management module, an ERP system web application can help organizations manage their finances more effectively, including budgeting, forecasting, and accounting.

Overall, the outcomes of an ERP system web application can have a significant impact on an

VI. SUMMARY AND FUTURE SCOPE

The future scope of ERP system web applications is vast, with new technologies and trends constantly emerging. Here are some potential future developments for ERP system web applications:

- **Artificial Intelligence (AI):** Integration of AI technology with ERP systems could provide predictive analytics and insights, automating routine tasks, and improving decision-making capabilities.
- **Blockchain:** The use of blockchain technology in ERP systems could provide greater security, transparency, and trust in data and transactions.
- **Internet of Things (IoT):** The integration of IoT devices with ERP systems could provide real-time monitoring of equipment and assets, leading to improved maintenance and efficiency.
- **Cloud Computing:** The adoption of cloud-based ERP systems is likely to increase as more organizations shift towards remote work and require access to their systems from anywhere.
- **Mobile Access:** The demand for mobile access to ERP systems is expected to grow as more employees work remotely or require access to data on-the-go.
- **Personalization:** ERP systems may become more personalized to meet the unique needs of individual users or departments within an organization.

Overall, the future scope of ERP system web applications is promising, with continued advancements in technology and the potential for greater integration with other systems and processes. As organizations increasingly rely on digital tools to manage their business processes, the role of ERP systems is likely to become even more critical in ensuring their success

REFERENCES

- [1] Node.js; Available from: <https://nodejs.org/en/>.
- [2] Tilkov S, Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing. 2010;14(6):80–83.
- [3] LozinskiL. The Uber Engineering Tech Stack, Part I: The Foundation; 2016. Available from: <https://eng.uber.com/tech-stack-part-one/>.
- [4] HarrellJ. PayPal Engineering, editor. Node.js at PayPal; 2013. Available from: <https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/>.
- [5] XiaoY. Node.js in Flames; 2014. Available from: techblog.netflix.com/2014/11/nodejs-in-flames.html.
- [6] Manchar A, Chouhan A. Sales force CRM: A new way of managing customer relationship in cloud environment. In: 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE; 2017. p.1–4.
- [7] “NodeJS,” <https://nodejs.org>, accessed 30 September 2018.
- [8] “Angular,” <https://angular.io/>, September 2016, accessed 28 September 2018.
- [9] D. Lehn, “json ld-cli,” <https://github.com/digitalbazaar/jsonld-cli>, September 2015, accessed 30 September 2018.
- [10] “Commander.js,” <https://github.com/tj/commander.js>, accessed 30 September 2018.
- [11] “chalk,” <https://github.com/chalk/chalk>, accessed 30 September 2018.
- [12] “Inquirer.js,” <https://github.com/SBoudrias/Inquirer.js/>, accessed 30 September 2018.
- [13] “ECMAScript 2015: Promise objects,” <https://www.ecma-international.org/ecma-262/6.0/#sec-promise-objects>, accessed 30 September 2018.
- [14] Zhang Zhaoyuan. A preliminary study on the back-end technology of webNode.js [J]. Small and Medium Enterprise Management and Technology 2020, Issue 22: 193- 194.



- [15] Wang Jijie. Design and implementation of a highly concurrent network application architecture based on Node.js technology [J]. Journal of Tonghua Teachers College, 2020, Issue 7: 106-109
- [16] Zhu Xiaoyang, Design and Implementation of Backend System of Learning Platform Based on Node.js[J]. Computer Knowledge and Technology: Academic Edition, 2019, Issue 5: 116-118.
- [17] Zhou Anhui. Discussion on Node.js asynchronous programming mode [J]. Journal of Sichuan Vocational and Technical College, 2018 Issue 4:149-154.