

Paper Title: A Survey on Current Technologies for Web Development

Authors: A. V. Akhil Krishna, Dr. Padmashree T

Published In: *International Journal of Engineering Research & Technology (IJERT)*, Volume 9, Issue 6, June 2020

1. Introduction

- The paper emphasizes how web development remains a critical area despite the dominance of mobile platforms.
 - It highlights the importance
 - Web development continues to evolve rapidly with modern tools that enhance user experience, scalability of Progressive Web Apps (PWAs), which combine the best of mobile and web applications., and performance.
 - Selecting the right technology stack is crucial — the wrong combination can jeopardize the entire project.
-

2. Problem Statement

- There is no universally accepted model for choosing and implementing web development technologies.
 - Developers are often overwhelmed by the variety of tools and frameworks, making the decision process confusing.
 - The paper aims to provide a comparative overview of major technologies to aid better decision-making.
-

3. Objective of the Study

- To compare relational and non-relational databases in terms of performance, scalability, and flexibility.
 - To analyze frontend and backend frameworks for building efficient web apps.
 - To highlight performance issues, runtime behavior, and security concerns in widely used platforms like Node.js and Angular.
 - To recommend suitable tech stacks based on different application needs.
-

4. Technologies Discussed in the Paper

✓ Databases: MongoDB vs. MySQL

- MongoDB (NoSQL) is shown to be more flexible and scalable for modern, data-heavy applications.
- MySQL (RDBMS) is reliable but can struggle with unstructured or large-scale datasets.
- Use case examples where MongoDB's schema-less nature provides more customization per user.

✓ Frontend Technologies: AngularJS & Angular 2

- Angular 2 (with TypeScript) supports Aspect-Oriented Programming for better modularity and code maintenance.
- Dynamic form rendering, type safety, and context-aware interfaces make Angular 2 more robust for enterprise-level apps.

✓ Backend Technology: Node.js

- Node.js is based on an asynchronous, non-blocking I/O model, ideal for real-time applications.
- Provides a single language (JavaScript) environment across client and server.
- Runtime analysis shows high CPU usage due to the V8 engine, especially when handling C++-based operations.

✓ Security Considerations in Node.js

- The paper highlights two denial-of-service (DoS) vulnerabilities.
- It advises developers to avoid Node.js in security-critical environments.
- Suggests the importance of experience and best practices in asynchronous server-side JavaScript programming.

5. Key Observations & Insights

- **Performance Metrics:** Node.js runtime behavior can lead to performance bottlenecks if not managed properly.
 - **Tool Ecosystem:** Developers must be cautious about dependencies and third-party libraries (common in JavaScript projects).
 - **Developer Experience:** Tools like Angular improve development ease but require a steep learning curve.
 - **Scalability Needs:** NoSQL databases like MongoDB outperform relational databases for applications needing horizontal scaling.
-

6. Real-world Implications

- Businesses must tailor their tech stack based on their application's complexity, user base, and data structure.
 - Developers must stay updated with evolving frameworks and library versions to avoid deprecated practices.
 - The paper indirectly encourages adopting DevOps and automated testing strategies to handle dynamic application behavior.
-

7. Conclusion

- There is no one-size-fits-all technology stack.
 - The paper successfully highlights strengths and limitations of the most widely-used tools in web development.
 - Developers and businesses should take a modular, test-driven, and security-aware approach to choosing technologies.
-

8. Critical Review / Personal Opinion

- The paper offers a very practical guide for beginners and intermediate developers.
- Covers a wide range of current tools and best practices.
- A few limitations:
 - Could include case studies or performance benchmarks for real-world applications.
 - Would benefit from updated data or mention of newer frameworks like React, Vue.js, or Next.js.
- Still, the review is informative and relevant for understanding core technologies in modern web development.