REFERENCES

[1] "ECMAScript." [Online]. Available: http://www.ecmascript.org/.

[2] L. Clark, "A crash course in just-in-time (JIT) compilers," Mozilla Org, 2017. [Online]. Available: https://hacks.mozilla.org/2017/02/a-crash-course-in-just-in-time-jit-compilers/.

[3] S. SOUDERS, High Performance Web sites. 2007.

[4] G. Grisogono, "JavaScript Performance Tips & Tricks," 2012. [Online]. Available: https://moduscreate.com/blog/javascript-performance-tips-tricks/.

[5] C. C. Charles D. Garrett, Jeffrey Dean, David Grove, "Measurement and Application of Dynamic Receiver Class Distributions," in Department of Computer Science and Engineering, FR-35, University of Washington, Seattle, Washington 98195 USA, 1994.

[6] A. Gal, B. Eich, M. Shaver, and D. Anderson, "Trace-based just-in-time type specialization for dynamic languages," ACM SIGPLAN Notices, Volume 44, Issue 6, pp. 465–478, 2009.

[7] "Kraken Benchmark." [Online]. Available: http://krakenbenchmark.mozilla.org/.

[8] "Octane Benchmarks." [Online]. Available: https://developers.google.com/octane.

[9] "SunSpider Benchmarks." [Online]. Available: http://www.webkit.org/perf/sunspider/sunspider.html.

[10] W. Ahn, J. Choi, T. Shull, M. J. Garzarán, and J. Torrellas, "Improving JavaScript Performance by Deconstructing the Type System." in PLDI '14: Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation, Jun. 2014, pp. 496–507.

[11] Jeff Nelson, "Which language has the best future prospects: Python, Java, or JavaScript?," 2016. [Online]. Available: https://www.quora.com/Which-language-has-the-best-future-prospects-Python-Java-or-JavaScript. [Accessed: 31-Jan-2022].

[12] “V8 JavaScript Engine.” [Online]. Available: https://developers.google.com/v8/.

[13] “SpiderMonkey Project.” [Online]. Available: https://developer.mozilla.org/en- US/docs/SpiderMonkey.

[14] B. Eich, “New JavaScript Engine Module Owner.” [Online]. Available: https://brendaneich.com/2011/06/new-javascript-engine-module-owner/. [Accessed: 30-Dec-2021].

[15] “A Short History of JavaScript.” [Online]. Available: https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript. [Accessed: 13-Dec-2021].

[16] “Chakra.” [Online]. Available: http://blogs.msdn.com/b/ie/archive/2021/03/18/the-new-javascript-engine-in-internet-explorer-9.aspx.

[17] J. K. Martinsen, H. Grahn, and A. Isberg, “A Comparative Evaluation of JavaScript Execution Behavior,” in Web Engineering: 11th International Conference, ICWE 2011, Paphos, Cyprus, June 20-24, 2011, S. Auer, O. Díaz, and G. A. Papadopoulos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 399–402.

[18] and M. K. M. S. L. Graham, P. B. Kessler, “Gprof: A call graph execution profiler,” in ACM Sigplan Notices, vol. 17, ACM, 1982, pp. 120–126.

[19] M. Fowler, “CodeSmell,” 2006. [Online]. Available: https://martinfowler.com/bliki/CodeSmell.html. [Accessed: 09-Jan-2022].

[20] R. Ashton, “you have ruined javascript,” 2014. [Online]. Available: http://codeofrob.com/entries/you-have-ruined-javascript.html. [Accessed: 09-Nov-2021].

[21] M. Selakovic and M. Pradel, “Performance issues and optimizations in JavaScript,” Proc. 38th Int. Conf. Softw. Eng. - ICSE ’16, pp. 61–72, 2016.