Create a .md file for Lab 2, write a paper about how JS engine optimizes JavaScript code.

A JavaScript engine is a program or an interpreter which executes JavaScript code.

JavaScript is a high-level language, and it can’t interact with machine lower-level logic. The first step is JavaScript engine parses the source code and turns into Abstract Syntax Tree (AST) producing bytecode. AST (Abstract Syntax Tree) is a data-structure that represents what this code really means.

The optimizing compiler makes certain assumptions based on the profiling data it has, and then produces highly optimized machine code.

V8 compilers take care of rest of the work. So, the first step is interpreter will interpret the code. The interpreter generates unoptimized bytecode quickly, and the optimizing compiler takes a little longer but eventually produces highly optimized machine code.

Any code that can be optimized then goes to the optimizing compiler. Then the optimizing compiler analyzes the code and make assumptions to make it even faster. The optimizing compiler generates highly optimized machine code, but sometimes it has to de-optimize on runtime and change back to the byte code.

The interpreter which generates the bytecode is called Ignition, while the optimizing compiler is called TurboFan.

The Profiler monitors and watches code to optimize it. it'll detect which parts of your code are being used the most, and then it'll send them over to be compiled and stored.

The compiler works ahead of time and creates a translation of the code that has been written and compiles down to a lower-level language that machines can read.

REFERENCES:

- https://mathiasbynens.be/notes/shapes-ics
- https://www.digitalocean.com/community/tutorials/js-v8-engine
- https://codeburst.io/javascript-compiler-optimization-techniques-only-for-experts-58d6f5f958ca
- Class power point