Node.js is a runtime environment for executing JavaScript code outside of a browser. Node is used to build back-end services such as an API (Application Programming Interface). Node is ideal for building highly-scalable, data-intensive, and real-time applications with back-end services. Node is great for prototyping and agile development, building superfast and highly scalable applications or services. Also used by Walmart and PayPal. PayPal found that using Node can build apps twice as fast with fewer people, 1/3 fewer lines of code, 40% fewer files, twice the requests per second, and down 35% faster response time. In node Applications, JavaScript can be used. Source code can also be cleaner and more consistent because it uses JavaScript. If a developer already knows JavaScripts, node lets them develop front end applications without learning a new language. Node also has a large ecosystem of open-source libraries through npm. Up until 2009, the only way to execute JavaScript was inside a browser. In 2009, Ryan Dahl took Google Chrome’s JavaScript Engine, v8, and imbedded it inside a C++ Program and called that program Node. Since Node is not in the browser, it provides an environment that contains its own objects such as a file system and works with server requests, not like documents from the browser and manipulating them. Node is basically a JavaScript Engine with additional functionality not available in browsers and can be ran in a terminal. Node is non-blocking, or asynchronous. Because it is asynchronous, it can use resources more efficiently by not waiting for threads to finish a single request. It can go on and do other requests while doing other requests while waiting for a response. Node is ideal for I/O-intensive apps. Node Should not be used for CPU-intensive applications such as video encoding or image manipulation service. Since Node is single threaded, it should not be used for CPU intensive applications. It should only be used for data intensive or real time applications.