Node.js

Node.js is a free and open source project available on Github.In the Node.js environment, a packages manager called npm is used. Typically, this programme is installed on the user's computer along with other Node.js components.

Advantage of Node.js:

The most advantageous features of Node.js include easy scaling, a welcoming environment for full stack development, and engaged community support. The event-based approach, asynchronous request handling, and built-in JS ON support are noteworthy technical features.

Disadvantages of Node.js

The Node.js community lists the following as its main drawbacks: poor optimization for demanding jobs, frequent changes to the actual APl, issues with documentation, and phenomena referred to as "callback hell" that occur from asynchronous execution.

Frequently asked questions (FAQs):-

What is Streams ?

Node.js uses streams to read and write data sequentially to files. Because streams don't open the entire file, they are best for reducing the use of device memory while working with huge files. Instead, it reads (or writes) files piecemeal in response to requests from developers for specific portions of the file.

What is Callback hell in Node.js ?

Due to Node.js's capabilities for asynchronous operations, problems with callbacks' execution sequences can arise. When a script launches an async callback that needs data from other callbacks that haven't ended execution, developers who utilise sophisticated callbacks must anticipate these circumstances and know how to detect and prevent them.

Difference between JavaScript and Node.js?

Node.js needs to be installed on the user's computer, whereas JavaScript can be used in any current browser. However, Node.js can also run JavaScript outside of the browser, which is sometimes more commodious. For instance, it aids in the creation of server-side components (back-end), whereas

JavaScript is effective in the creation of client-side components (front-end). Additionally, JS scripts can work with DOM and HTML because they are executed in the browser, whereas Node.js cannot.

What is REPL in the context of Node.js?

The name Read-Eval-Print-Lopp, or REPL, stands for an interactive shell that handles Node.js expression processing. Its name reveals how it operates: Reading the user's code in step one, evaluating the outcome of step two's interpretation, printing step three's result, and returning to step one in step four are the steps. In Node.js, it is mostly used for "on-the-fly" testing of JavaScript programmes.