

Basic Questions. 1.How can you make a network request with http module from the backend? Ans = // Importing https module const http = require("http");

```
// Setting the configuration for // the request const options = { hostname:
"jsonplaceholder.typicode.com", path: "/posts", method: "GET", };

// Sending the request const req = http .request(options, (res) => { let data
= "";

res.on("data", (chunk) => {
    data += chunk;
});

// Ending the response
res.on("end", () => {
    console.log("Body:", JSON.parse(data));
});

}) .on("error", (err) => { console.log("Error:", err); }) .end();
```

2.How can you create your own events? Ans = Node.js has a built-in module, called "Events" we can use it to create our own events. example => let events = require("events"); let EventEmitter = new events.EventEmitter();

```
let myEvent = function () { console.log("Hello World"); };

eventEmitter.on("hello", myEvent);

eventEmitter.emit("hello");
```

5.What is the difference between readFile and readFileSync? Ans = readFile is asynchronous while readFileSync is synchronous. They first read the file then deliver the content to us but the difference is that readFile not blocks the execution of the code. It takes a callback function and continue the execution and whenever the response is ready then it will send the response to us. readFileSync block the execution until it completes the operation then it returns the response.

6.What are CORS? How do you configure them? Why do you need them? Ans = CORS stands for Cross-Origin Resource Sharing. It allows you to make requests from one website to another website in the browser, which is normally prohibited by another browser policy called the Same-Origin Policy (SOP).

7.What is rate limiting? Ans= Rate limiting is a strategy for limiting network traffic. Rate limiting applies to the number of calls a user can make to an API within a set time frame. This is used to help control the load that's put on the system. Rate limiting helps prevent a user from exhausting the system's resources. Without rate limiting, it's easier for a malicious party to overwhelm the system.

8.How does middlewares work in express? Ans = The middleware in is a function that will have all the access for request and response object and moving to the

next middleware function in the application. These functions are used to modify req and res objects for tasks like parsing request bodies, adding response headers.

9.What is the difference between Encryption and Hashing? Ans= In the Encryption we can encrypt and decrypt the data with an access key. In Hashing data can be encrypted but not decrypted. For encryption we can use crypto module of node.js and for hashing we can use bcryptjs.

10.What is the difference between https and http? Ans = HTTP stands for HyperText Transfer Protocol and HTTPS stands for HyperText Transfer Protocol Secure. HTTPS uses TLS which is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet to encrypt normal HTTP requests and responses, and to digitally sign those requests and responses. As a result, HTTPS is far more secure than HTTP.

11.What is TLS? Ans = It is a cryptographic protocol that provides end-to-end security of data sent between applications.

13.What is JWT Token? Why do we need to use JWT? What are some pros and cons? Ans = A JSON web token(JWT) is JSON Object which is used to securely transfer information over the web(between two parties). It can be used for an authentication system and can also be used for information exchange.The token is mainly composed of header, payload, signature. A header in a JWT is mostly used to describe the cryptographic operations applied to the JWT like decryption technique used on it. It can also contain the data about the media/content type of the information we are sending. The payload is the part of the JWT where all the user data is actually added. This is the third part of JWT which is used to verify the securityKey of token.

14.What is salting? Where do we store salt? Ans = Salting is the process of adding unique random strings of characters to passwords in a database. This is done to make passwords more secure. The string of characters added to the password is called the salt. A salt can be added in front or behind a password.

15.What is the difference between authorisation and Authentication? Ans = Authentication is the process of verifying the identity of a user, while in the process of authorization user's authorities are checked for accessing the resources.

16.What is the difference between JS on the browser and node? Ans = Js is client side and Node is server side.

17.What is V8? Ans = V8 JavaScript engine is an open source JavaScript and WebAssembly engine that compiles JavaScript to optimized machine code before execution.V8 engine was initially developed for Google Chrome.

18.Explain in brief what is node js? Ans= Node.js is an open-source server side runtime environment built on Chrome's V8 JavaScript engine. It provides an event driven, non-blocking (asynchronous) I/O and cross-platform runtime environment for building highly scalable server-side applications using JavaScript.

19.How is node js non-blocking? Ans= Node.js works asynchronously by using the event loop and callback functions, to handle multiple requests coming in parallel. An Event Loop is a functionality which handles and processes all your external events and just converts them to a callback function. It invokes all the event handlers at a proper time. Thus, lots of work is done on the back-end, while processing a single request, so that the new incoming request doesn't have to wait if the processing is not complete.

While processing a request, Node.js attaches a callback function to it and moves it to the back-end. Now, whenever its response is ready, an event is called which triggers the associated callback function to send this response.

20.What is throughput? Ans = Basically throughput is the speed of a server at which something is processed in a specified time.

21.How is Node js having high I/O throughput? Ans = Because of the asynchronous feature of node js the i/o is very high. It makes the program fast and scalable without using as many resources as a multi-threaded application.This makes Node a good fit for data-intensive and I/O intensive programs.

22.What are CPU intensive tasks? Ans = The tasks which have complex calculations, sorting, traversing are mainly the CPU intensive task.CPU intensive tasks depends on how fast is their execution.

23.How can you end up blocking your main thread in node.js? Ans = Node js uses event loop to handle blocking operations. An Event Loop is a functionality which handles all your external events and just converts them to a callback function. After that event loop is constantly checks our main thread and once it is empty then event loop will push these callback functions to the main thread.

25.What are different phases in event loop?

26.What is process.tick? ans = Every time the event loop takes a full trip, we call it a tick. and in the process.nextTick() we can pass a callback function to instruct the engine to invoke this function at the end of the current operation, before the next event loop tick starts.

28.What is the difference between setTimeout and setInterval? Ans = setTimeout allows us to run a function once after the interval of time. setInterval allows us to run a function repeatedly, starting after the interval of time, then repeating continuously at that interval.

29.What is package.json? Ans= A package.json is a JSON file that exists at the root of a Node project. It holds metadata relevant to the project and it is used for managing the project's dependencies, scripts, version and a whole lot more.

Advance Questions. 7.What are the different types of databases? Ans = So there are mainly four types of database.relational, non-relational,key-value based and graph.

(1)relational(SQL) = A relational database is a type of database that stores

and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

(2)non-relational(noSQL) = A non-relational database stores data in a non-tabular form, and more flexible than the traditional, SQL-based, relational database structures. The non-relational databases are below.

(3)key-value = A key-value database (sometimes called a key-value store) uses a simple key-value method to store data. These databases contain a simple string (the key) that is always unique and an large data field (the value). They are easy to design and implement. The examples of these databases are DynamoDB, redis and our browser localStorage is also a key-value store.

(4)graph = A graph database stores nodes and relationships instead of tables, or documents. Data is stored just like you might sketch ideas on a whiteboard. Your data is stored without restricting it to a pre-defined model, allowing a very flexible way of thinking about and using it.