Q1. Difference between let, var and const

If we have declared a variable using a **var** keyword that variable can be reassigned and redeclared using var. However, if the same variable is redeclared using const or let, it would result in a compile time error.
If we have declared a variable using a **let** keyword that variable can be reassigned but cannot be redeclared using any let, const or var again.

Q2. Hoisting in Javascript

Hoisting is a behavior in JavaScript where variable and function declarations are moved to the top of their containing scope during the compilation phase.

Variables declared with **var** are hoisted to the top of their scope and initialised with undefined. This means you can access the variable before its declaration, but its value will be undefined until the actual declaration in the source code.

Variables declared with **let** and **const** are also hoisted, but they are not initialised. If you try to access the variable before its declaration, you will get a ReferenceError.

Functions are also hoisted, but function expressions are not hoisted. When you assign a function to a variable it becomes function expression

Q3. What are higher order functions?

A higher-order function in JavaScript is a function that can take another function as an argument, return a function as a result, or both.

Q4. What is IIFE(Immediately Invoked Function Expression)

An Immediately Invoked Function Expression (IIFE) is a JavaScript design pattern where a function is defined and executed immediately after its creation. It is a self-invoking function that runs as soon as the browser encounters it. The primary purpose of using an IIFE is to create a private scope for variables and functions to avoid polluting the global scope. Variables declared inside the IIFE are not accessible from the outside, preventing them from polluting the global scope

Q5. What is a promise

A promise is an object that may produce a single value sometime in the future with either a resolved value or a reason that it's not resolved(for example, network error). It will be in one of the 3 possible states: **fulfilled**, **rejected**, or **pending**. Promises are used to handle asynchronous operations. They provide an alternative approach for callbacks by reducing the callback hell and writing the cleaner code.

Q6. What is a callback function?

Callbacks in JavaScript refer to functions that are passed as arguments to other functions and are then invoked within those functions. Callbacks are a way to achieve asynchronous behaviour, allowing certain code to run after an operation has completed.

Q7. What is async await

The await keyword can only be used inside an async function.
It is used to pause the execution of the asynchronous function until the awaited promise is resolved or rejected. Whereas, the async keyword is used to declare a function as asynchronous.

Q8. What is the difference between null and undefined

Type of null is object, type of undefined is undefined. Null is an actual value assigned to a variable, whereas undefined is nothing. Undefined Indicates the absence of the variable itself, whereas null indicates the absence of a value.

Q9. What are global variables

Global variables are those that are available throughout the length of the code without any scope. The var keyword is used to declare a local variable but if you omit it then it will become global variable

Q10. What is promise.all()

Promise.all() is a method in JavaScript that takes an iterable (such as an array) of promises and returns a single promise. This resulting promise is fulfilled with an array of the fulfilled values from the input promises in the same order as the promises passed to Promise.all(). If

any of the input promises are rejected, the resulting promise is rejected with the reason of the first rejected promise.

Use Cases:

Parallel Execution:

Promise.all() is commonly used when you want to execute multiple asynchronous operations in parallel and wait for all of them to complete.

Q11. Is Js a compiled language or an interpreted language?

JavaScript is often referred to as an interpreted language, but it is more accurately described as a "just-in-time compiled" language. When you run JavaScript code in a web browser, the browser's JavaScript engine (like V8 in Chrome or SpiderMonkey in Firefox) reads the JavaScript source code and converts it into an intermediate representation. This intermediate representation is then optimized and compiled into machine code by the engine, and the compiled code is executed.

So, the process involves both interpretation (parsing and execution of the source code) and compilation (generation and optimization of machine code). This combination of interpretation and compilation is why JavaScript is often called an "interpreted" language with a "just-in-time compiler" (JIT compiler).

Q12. What is the difference between for and for-each loop?

We cannot use break, continue in foreach loop. for loops are more flexible and generic, suitable for iterating over various data types with precise control. forEach loops are specifically designed for iterating over arrays,

Q13. Difference between the find and filter method

Filter:

Returns a new array containing all elements that satisfy a provided testing function.
Continues searching the entire array and includes all matching elements.
Returns an empty array if no matching elements are found.

Find:

Returns the first element in an array that satisfies a provided testing function.
Stops searching the array once it finds an element that matches the condition.
Returns undefined if no matching element is found.

Q14. Difference between arrow functions and normal functions

Arrow functions are not hoisted so we cannot access them before initialization whereas normal functions are hoisted. Secondly, we cannot access keywords like this in arrow functions whereas we can use them in normal functions

Q15. What is NodeJs

Node.js is a runtime environment that allows you to run JavaScript on the server side, while Express is a web application framework for Node.js. In other words, Node.js provides the runtime for executing JavaScript on the server, and Express is a framework that runs on top of Node.js, simplifying the process of building web applications. It works on a single-threaded event loop and a non-blocking I/O which provides high rate as it can handle a higher number of concurrent requests.

Q16. What is REPL in node js

In Node.js, *REPL* stands for Read-Eval-Print Loop. It is a simple, interactive programming environment that allows you to enter and execute JavaScript code directly in the terminal. The REPL provides a way to interact with the Node.js runtime, test small pieces of code, and experiment with JavaScript expressions and statements.

Q17. Event Loop in Node Js

The Node.js event loop is a crucial aspect of its non-blocking, asynchronous nature. The event loop in Node.js handles I/O operations asynchronously to ensure efficiency. When a task, like reading a file or making a network request, is initiated, it doesn't block the entire process. Instead, it's delegated to the event loop. The event loop continuously checks the task queue for completed tasks.
Here are the key components:
1. **Event Queue:** This is where events and tasks are queued up. Asynchronous operations, like reading a file or making an HTTP request, are initiated and placed in the queue.

2. **Event Loop:** It's a continuous process that constantly checks the event queue for any tasks that have been completed. If there's a completed task, it's pushed to the callback queue.

3. **Callback Queue:** Completed tasks, along with their callback functions, are placed here. These callbacks are functions that should be executed after the associated task is finished.

4. **Call Stack:** This is where function calls are tracked. When a function is called, it's pushed onto the stack. When it's done, it's popped off the stack.

The event loop's job is to move events from the queue to the call stack when the stack is empty. This ensures that the program doesn't wait for one task to complete before moving on to the next one, making Node.js highly scalable and efficient for handling a large number of concurrent connections.

An event loop in Node.js handles all the asynchronous callbacks in an application. It is one of the most important aspects of Node.js and the reason behind Node.js have non-blocking I/O. Since Node.js is an event-driven language, you can easily attach a listener to an event and when the event occurs the callback will be executed by the specific listener. Whenever functions like setTimeout, http.get, and fs.readFile are called, Node.js executed the event loop and then proceeds with the further code without waiting for the output. Once the entire operation is finished, Node.js receives the output and then executes the callback function. This is why all the callback functions are placed in a queue in a loop. Once the response is received, they are executed one by one.

Q18. Just a pro tip

when we compare two same values of primitive data type we get true, however, if we compare two same values e.g. [ ]===[ ] of non-primitive data type we get false because dono ka alag alag reference javascript ki memory me save hojata ha.

Q19. What is REST API/Framework

A REST API (Representational State Transfer - Application Programming Interface) is a type of web API that follows the principles and constraints of the REST architectural style. It is stateless because the server does not store client’s state between requests.

Q20. What does “low in coupling and high in cohesion” mean in backend development?

Low coupling means that the components or modules of the system are designed to be independent of each other. In other words, they have minimal or no dependencies on each other. This design principle makes the codebase easier to maintain, test, and modify because changes in one module do not affect other modules in the system. It also allows for better scalability as new features can be added without disturbing the existing system.

High cohesion, on the other hand, refers to the design principle of grouping related functionality together within a module or component. This means that the functions or methods within a module have a common purpose and work together to achieve that purpose. This design principle also makes the codebase easier to maintain, test, and modify, as changes to one function or method affect only the related functionality.

Q21. Explain the process of Sharding.

Sharding is the process of splitting data up across machines. We also use the term “partitioning” sometimes to describe this concept. We can store more data and handle more load without requiring larger or more powerful machines, by putting a subset of data on each machine.

Give an example of encapsulation.

The notion of data hiding is referred to as encapsulation. Protected and private members in C++ are examples.

How can you call a base class method without creating an instance?

you can call a base class method without creating an instance of the class by using the class name directly. This is typically possible when the method is defined as static or class-level.

What are static method?

Static methods belong to a class rather than an instance of the class. They are associated with the class itself rather than any particular object created from the class. In many programming languages, static methods are denoted by a keyword such as static or by using decorators.

Abstraction in OOP?

Abstraction is the process of hiding the implementation details and showing only the functionality to the user. An abstract class can have both concrete and abstract methods. Interfaces only have abstract methods. We cannot instantiate the object of abstract class.

What is an interface?

An interface declares a list of method signatures without specifying how those methods should be implemented. It defines what methods a class should have, leaving the details of the implementation to the classes that implement the interface

Define virtual functions.

The functions that help achieve runtime polymorphism are a part of functions present in the parent class and overridden by a subclass.

What is a Super Keyword?

You can use "super" to access members (fields or methods) of the superclass from within the subclass.

What is constructor chaining?

Constructor chaining is a method to call one constructor from another concerning a current object reference. It can be done in two ways: –

- Using the “this” keyword, the reference can be made to the constructor in the current class.
- To call the constructor from the base class “super” keyword will be used.

What is the difference between the RANK() and DENSE_RANK() functions?

The RANK() function in the result set defines the rank of each row within your ordered partition. If both rows have the same rank, the next number in the ranking will be the previous rank plus a number of duplicates. If we have three records at rank 4, for example, the next level indicated is 7.

The DENSE_RANK() function assigns a distinct rank to each row within a partition based on the provided column value, with no gaps. It always indicates a ranking in order of precedence. This function will assign the same rank to the two rows if they have the same rank, with the next rank being the next consecutive number. If we have three records at rank 4, for example, the next level indicated is 5.

Difference between union and union all?

UNION merges the contents of two structurally-compatible tables into a single combined table. The difference between UNION and UNION ALL is that UNION will omit duplicate records whereas UNION ALL will include duplicate record.

GROUP BY In SQL

In SQL, the GROUP BY clause is used to group rows that have the same values in specified columns into summary rows. It is often used in conjunction with aggregate functions like SUM, COUNT, AVG, MAX, or MIN to perform operations on each group of rows

What is the difference between BETWEEN and IN operators in SQL?

- BETWEEN is used for range comparisons, while IN is used for a list of discrete values.
- BETWEEN checks if a column value is within a specified range (inclusive), while IN checks if a column value matches any value in a specified list.

What is stored procedure?

A stored procedure is a precompiled collection of one or more SQL statements or procedural statements (such as if statements, loops, etc.) that is stored in a relational database management system (RDBMS).

What is the difference between CHAR and VARCHAR2 datatype in SQL?

Both of these data types are used for characters, but varchar2 is used for character strings of variable length, whereas char is used for character strings of fixed length. For example, if we specify the type as char(5) then we will not be allowed to store a string of any other length in this variable, but if we specify the type of this variable as varchar2(5) then we will be allowed to store strings of variable length. We can store a string of length 3 or 4 or 2 in this variable.

11. What is the difference between primary key and unique constraints?

The primary key cannot have NULL values, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constraints. The primary key creates the clustered index automatically but the unique key does not.

Difference between delete and truncate?

The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row.

TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.

How are the elements of a 2D array stored in the memory?

Row-Major Order: -In row-major ordering, all of the rows of a 2D array are stored in memory in a contiguous manner. First, the first row of the array is entirely stored in memory, followed by the second row of the array, and so on until the final row.

Column-Major Order: In column-major ordering, all of the columns of a 2D array are stored in memory in the same order. The first column of the array is entirely saved in memory, followed by the second row of the array, and so on until the last column of the array is wholly recorded in memory.

EXTRAS

Properties provide a way to access and manipulate the private fields (class variables) of a class. They encapsulate the internal state of an object and allow controlled access to it.

QuickSort algorithm is generally considered the fastest because it has the best performance for most inputs.

Balanced Binary Search Trees
Consider a Binary Search Tree with ‘m’ as the height of the left subtree and ‘n’ as the height of the right subtree. If the value of (m-n) is equal to 0,1 or -1, the tree is said to be a Balanced Binary Search Tree.

Heap is a special balanced binary tree data structure where root-node key is compared with its children and arranged accordingly. A min-heap, a parent node has key value less than its childs and a max-heap parent node has value greater than its childs.

Inorder Traversal:

Visit the left subtree.
Visit the root node.
Visit the right subtree.
In the case of binary search trees (BSTs), inorder traversal visits the nodes in ascending order.

Preorder Traversal:

Visit the root node.
Visit the left subtree.
Visit the right subtree.
Preorder traversal is useful for creating a copy of the tree.

Postorder Traversal:

Visit the left subtree.
Visit the right subtree.
Visit the root node.