1) What is Javascript

Ans JavaScript is A Server Side as well as client Side scripting language.Javascript is a single threded . javascript is synchronous.

In the help of javascript we can enhanced the interest of user.

Javascript is the capability of convert html into Dhtml.

Javascript is object based,prototype based multiparadigm scripting language.

javascript is a loosely type

Javascript is Developed by Branden Each in 1995.

Javascript is a user programmer friendly.

Java script maintain by ECMA script

2) How to execute javascript code on the browser.

Ans. Javscript Engine is Responsible to execute the javascript Code. Call Stack is a part of the javascript Engine . Everything is Happen in the call stack. Firstly in the Callstack global execution context is created.all script is loaded into the global execution context. For the every function called Separate global exexution context.

Each and Every global execution context have two component

Memory Component

Code Component

Now When the execution of f1 function is completed. The Global execution is created by the f1 function will be popped .

After the script is completed the global execution context will also be destroyed..

3) What are features of Javascript

JavaScript is A Server Side as well as client Side scripting language.Javascript is a single threded . javascript is synchronous.

In the help of javascript we can enhanced the interest of user.

Javascript is the capability of convert html into Dhtml.

Javascript is object based,prototype based multiparadigm scripting language.

javascript is a loosely type

Javascript is Developed by Branden Each in 1995.

Javascript is a user programmer friendly.

Java script maintain by ECMA script interpreted language

4) What is DOM ?

Dom is stands for Document Object Model . Dom is basically belong to the html Document.It is the root element that represent the Html Document.

The Document object model is programming interface in web development.which represent the structure of html or xml as a tree like structure. it has properties and methods and with the help of that We can add Dynamic Content in our pages.
If we want to access any dom element in js provide document object and it provide multiple method like document.getElementById , document.getElementByClassName , document.querySelector, document.querySelectorAll.

5) What is BOM ?

Ans Bom stands for browser Object Model. Bom is basically Belong to the window. The default object of a browser is a window. So, you can call all the functions of the window by specifying the window or directly. The window object provides various properties like document, history, screen, navigator, location, inner Height, inner Width.

6) How to perform click event on Dom Using Javascript

With the help of addEventListener method we can aply any event on any dom , first access dom element by id and class querySelector and we apply event.

What is alert() and confirm() ?

Alert and Confirm are javascript function to used to display dialog box to the user in a web browser.
Alert: The Alert method display an alert box with a message and a ok Button.Thealert method is used when you want to information through to the user.

Confirm: The Confirm Method Display a dialog message with a two button ok or cancle. It is often ask to user for confirmation before performing an action. If the user click ok the confirm return true. If the user click cancle the confirm return false

7) What is difference between undefined and null

Ans Undefined : undefined means If a variable is declared but not initialized.
2.Type of undefined is undefined.
Undefined can not be used in Json
Using undefined in a calculation will result NaN (20+undefined);

Null :
If you want to create a reference which point to nothing.
Type of null is Object.

Null can be used in Json
using null in a calculation return value

8) What is difference between == and ===
Ans The == operator is check the content nd === operator check the content as well as type.

9) What is function . Can we overload function in javascript
Function is a Block of code. when perform a Specific task of code. We can not overload the javascript Function. If we try to overload the javascript function. We will execute last definition of function

Advantage of Function
Code reuseability
Remove the code Redutancy and complexity of code.
Trouble shouting and Error handling is easy.
Encapsulation.
.

10) What is callback ? Explain with suitable example
Ans callback is a function which passed as argument into another function is called callback. A callback function can run after another function has finished.

For the Asynchronous Programming we can use Callback.

High order Function :
A Higher order function is a function that takes one or more function as a Argument or return as its result.

```
Ex : function fi(a,b){
  Console.log(“f1 called”);
     a();
}
Fi(function f2(){
   Cosole.log(“collback called”);
}, 100)
```

11) What is promise?
.

Ans. Promise is a Object. Which will be executed in future. Either it will be resolved or it will be rejected .Promise handle the multiple asynchronous operation.

Promise Has four stage in the promise.
Resolved.
Rejected.
Pending
allsettled

12) What are advantage of promise over callback
Promise provide multiple method to handel asynchronous opration .
In the help of promise error handling is easy;

13) What is callback hell
The phenomenon which happen when we nesting of multiple function within a function is called callback hell. The shape of the result is resemble as pyramid.The callback hell is also called pyramid of the droom.It makes the code very difficult to understand and maintain. The solution of callback hell is promise .

14) What is closure ?
A closure is the combination of a fuction bundled together with references to its surrounding state (the lexical environment)
(parant memory is called lexical environmant);
In other word a closure gives you access to Inner function is access to the outer function.

Advantage :

1. Closure give you access inner function can access outer function.
2. For the eccapsulate code with the help of closure.
3. Closure can also be used to create an currying in js.

Disadvantage : it will consume some extra memory

15) What is IIFEs(Immediately Invoked Function expression)
Function that are executed as soon as they are mounted these function are know as immediately invoked function

```
  (Function (x){
   Cosole.log("iife callsed")
})(2);
```

16) What is annonymous function ?
Ans Which Function that have any name

17) What is hoisting in javascript

Moving all the variable declaration on the top of script is called hoisting.
It is a default behaviour of java script.

18) What is difference between var, let and const keyword

Var are hoisted but let and const is not hoisted .
Var can be redefine but let and const are not ;
Var are can be access outside side the scope but let can not .
Var take memory in globle but let take memory in scop area.

19) Explain use strict

Ans it is a literal expression ,which comes in a ECMA script version 5,it is use for strict checking ,in that we can't initialize variable without declaration.

For example:-
"use strict"
X=3.14; //this will be occur an error because x is not declared

20) What is event bubbling and event capturing

With bubbling the event is first captured and handled by the innermost element and then propagate to outer element .

Capturing : with capturing the event is first captured by the outermost element and propogate to the inner element it called event trickling and capturing

21) What are the primitive datatype in javascript

Ans The predefined Data types provided by javascript language are known as a Primitive Data Type. Primitive data types are also known as a in build datatype.
1 Number 2. Null 3.BigInt 4. Symbol 5. String 6. Undefined 7. boolean

22) What are different types of popup boxes available in javascript

Java script has three type of popop box alert box , confirm box , prompt box

23) What will happen if an infinite while loop is run in javascript

Ans An infinite loop is a looping construct that does not terminate the loop and executes the loop forever. It is also called an indefinite loop or an endless loop. It either produces a continuous output or no output.

24) List HTML DOM mouse events

onclick , ondbclick , onmouseover , onmouseout , onmouseup , onmousemove , onmouseleve ,onmouseenter

25) How to get the last index of a string in javascript

by using lastIndexOf() method return the index of last occurance of a specified value in a string

26) Describe negative infinity in javascript

A number in javascript which can be derived by dividing the negative number by zero.

-7/0 = nagative infinitive

27) Explain await and aync ? How to use await and async?

Async await both are the operator and async is use only function exprestion async function

It return by default promise object (promise is an asynchronous non blocking code)

Await is use basiclly wait for the promise

28) How to handle the excption in javascript.

Ans In javascript we can handle the Exception with the help of try Catch

In try block we write that code in which we think to chance of occur error. Nd catch block which we write error handling code .If there is no exception in the code catch block is skipped.

29) What is Function expression nd Function Statement?

Ans Function statement:The Function Statement declared is a initial way of function.

Function statement are hoisted by default

```
 Function f1(){}
```

Function expression : A Function expression decalred as a expression . a function expression can be stored in the variable

Function expression are not hoisted by default.

```
Var inner = function f1(){}
   Inner();
```

30) What is generator function

Ans A generator Functionis a function that returns a Generator object, and is defined by the function keyword followed by an asterisk

```
function* generatorFunction() {}
```

A generator function, however, does not return a value immediately, and instead returns an iterable Generator object.

```
// Assign the Generator object to generator
```

```
const generator = generatorFunction();
```

The Generator object returned by the function is an iterator. An iterator is an object that has a next() method available, which is used for iterating through a sequence of values. The next() method returns an object with value and done properties. value represent the returned value, and done indicates whether the iterator has run through all its values or not.

Knowing this, let’s call next() on our generator and get the current value and state of the iterator:

```
// Call the next method on the Generator object
generator.next()
```

Generators introduce a new keyword to JavaScript: yield. yield can pause a generator function and return the value that follows yield, providing a lightweight way to iterate through values.

```
// Create a generator function with multiple yields
function* generatorFunction() {
  yield 'Neo'
  yield 'Morpheus'
  yield 'Trinity'

  return 'The Oracle'
}

const generator = generatorFunction()
```

Now, when we call next() on the generator function, it will pause every time it encounters yield. done will be set to false after each yield, indicating that the generator has not finished. Once it encounters a return, or there are no more yields encountered in the function, done will flip to true, and the generator will be finished.

Output

```
{value: "Neo", done: false}
{value: "Morpheus", done: false}
{value: "Trinity", done: false}
{value: "The Oracle", done: true}
```

31)What is Spread Operator and Rest Operator

Ans A function can be called with any number of arguments, no matter how it is defined..

```
function sum(…a, b) {
  return a + b;
}
```

```
alert( sum(1, 2, 3, 4, 5) );
```

spread operator: The JavaScript spread operator (...) allows us to quickly copy all or part of an existing array or object into another array or object.

31) What is rest parameter.
Ans The Rest parameter is an improve way to handle function argument.allowing us to more easily handle various input as a function.The rest parameter of the syntax is allow infinite argument in the array.with the help of rest parameter
We can call any number of argument.no matter how it was defined.

```
Function f1(…input){
Let Sum=0;
For(let x in input){
  Sum +=input
}
Return sum;
}
Console.log(f1(1,2));
Console.log(f2(1,2,3);
```

32) What is Currying
Ans Chaining of multiple Function is called javascript Currying.

33)What is Json
Ans Json is a web technique.which is used to exchange the data between two parties.

34)what is setinterval , settimeout,process.nexttick();
Ans SetImmediate:use SetImmediate when you want to execute some function asynchronously.as soon as possible. after execute the current function

setTimeout=SetTimeOut when you want to execute some function asynchronously. After a some specific time.

nextTick : process.nextTick() is a method in Node.js that schedules a function to be executed on the next pass of the event loop. It allows you to defer the execution of a function until the current call stack has completed, which can be useful for ensuring that certain code runs asynchronously or for breaking up long-running tasks.

35) what is the use of higher order function.
Ans It is use to manage the code reductancy they provide simplicity the code.

36)How we can compare two object.

Ans Json.stringify(a)===json.strigify(b)

37) what is Recursion
Ans Recursion is **a process of calling itself**. A function that calls itself is called a recursive function.

```
function recurse() {
   // function code
   recurse();
   // function code
}

recurse();
```

38)what is array inflating .
Ans Array inflating in javascript refer to a process of creating a new array by duplicating or repeating the value of an existing array multiple times .The resulting array is typically larger than the original array ,with the repeated values filling in the additional space.
There are many ways to inflate an array in javascript but one common approach is to use array.prototype.concat()method in combination with the array.prototype.fill()method.

```
Const originalarray=[1,2,3]
Const inflatarray = originalarray.concat(originalarray,originalarray)
Console.lo(inflatarray)
```

39) what is call(),bind(),apply().
Ans
    Call method: with call(),an object can use a method belonging to another object.

```
Const youtuber1 = {
   Name:”geekforgreeks”,
   Content :”bank”,
   Features:”Function(rating){
     Console.log(“name”+this.name+ “concat”+this.concat+”rating”+this.rating);
}
}

Const youtuber2 = {
   Name:”javatpoint”,
   Content :”java”,
}

Youtuber1.feature.call(youtuber2,5);
```

Apply() = method takes argument as an array.

```
Const youtuber1 = {
   Name:”geekforgreeks”,
   Content :”bank”,
   Features:”Function(rating,title){
     Console.log(“name”+this.name+
“concat”+this.concat+”rating”+this.rating,+”title”+this.title);
}
}

Const youtuber2 = {
   Name:”javatpoint”,
   Content :”java”,
}

Youtuber1.feature.apply(youtuber2,[6,”hello”]);
```

Bind():The Bind Method Takes an object as an First Argument and create a new Function.

```
Const student1 = {
Name:”Tanisha”,
Class:”11”
}

Const student2={
Name:”Sneha”,
Class:”12”

}



Const student3 ={

Name:”shradha”

Class:”10”

}
```

```
Function subject(){

Console.log(“name”+this.name+”class”+this.class)

}


Let s1= subject.bind(student1);

S1();
```

1)What is Node.js? Explaing the advantage of Node.js Over java and php

Ans Node js is Developeed By Ryan Dahl.

It is Not A Programmin Language.

It is a Open Source Cross Platform Server Environment.

Node js is Javascript Runtime Environment built on chrome v8 engine.

Node js is used Synchronous And Asynchronous Blocking liberary Both .

Node js is Single threded from Event Loop.

In the Help of Node js We cn run javascript code on the browser.

Node js is superfast and highly scalable.

Node js is used for Io itensive application.

2) What are the limitation of Node.Js

Ans

3)How Node.Js Works

Ans According to the above diagram, the clients send requests to the webserver to interact with the web application. These requests can be non-blocking or blocking and used for querying the data, deleting data, or updating the data.

Js receives the incoming requests and adds those to the Event Queue.

After this step, the requests are passed one by one through the Event Loop. It checks if the requests are simple enough not to require any external resources.

The event loop then processes the simple requests (non-blocking operations), such as I/O Polling, and returns the responses to the corresponding clients.

A single thread from the Thread Pool is assigned to a single complex request. This thread is responsible for completing a particular blocking request by accessing external resources, such as computation, database, file system, etc.

Once the task is completed, the response is sent to the Event Loop that sends that response back to the client.

4)How Node.js is single threaded

Ans NodeJS is a single-threaded platform. This means that it can only process one request at a time.

Example : Took order from table 1 and give it to kitchen then goes to table 2 while taking the order from table 2 table 1's food is ready but now waiter can not immediately come and give the food to table 1 , waiter must finish the order from table 1 and then hand over it to kitchen after that only he should give the prepared meals to table 1.

NodeJS Web Server maintains a **limited Thread Pool** to provide services to client requests. Multiple clients make multiple requests to the NodeJS server. NodeJS receives these requests and places them into the **EventQueue .**
NodeJS server has an internal component referred to as the **EventLoop** which is an infinite loop that receives requests and processes them. **This EventLoop is single threaded**. In other words, **EventLoop is the listener for the EventQueue**.

5)Explain any five built-in package/Dependency name in node.

Ans fs,http,url,path,event

6)What is module in node.js

Ans. A simple javascript file is called Module. Modules are basically encapsulated blocks of code that can be imported and used anywhere in the application.

There are three types of modules in Node.js:

1. **Core modules :** These are a set of built-in modules that are a part of Node.js. They can be imported and used without any installation requirements.

   Example - file system (fs) module, and HTTP module.

2. **Local modules :** These are the modules that users can create to improve the code organization. We can create a simple module as shown below:

   ```
   Exports.multiply=function(a,b){

   Return a*b;
   ```

```
}
```

3. Third party Module: These are the modules required the installation of node package manager.After installation they exports and used
   Ex mongoose,express

7)How to create server in node.js.

Ans

```
Import http from ‘http’


var server = http.createServer((request,response)=>{

  response.write(“welcome in Node js”);

})


Server.listen(3000,()=>{

Console.log(“server Started”);
```

```
})
```

First we have to import http we don’t need to install http it is core module of node,then we create server with the help of http reference ,then listen for any incoming request.

8_40what is Module.exports

Ans
when we want to export single class function/class/ variable from one module to another module   we use Module.exports .

The module.exports is a special object which is included in every JavaScript file in the Node.js application by default. The module is a   variable that represents the current module, and exports is an object that will be exposed as a module. So, whatever you assign to module.exports will be exposed as a module.

9) How node.js handle multiple request

10)How to use url module in node.js
Ans The url module is core part of the node js.The url module is used to split web address into readable parts.

Example : import url from “url”

```
Var adr=”"http://localhost:8080/default.htm?year=2017&month=february"”
var     q  =url.parse(adr,true);
console.log(q.host)
console.log(q.query);
console.log(q.search)
var qdata=q.query
console.log(qdata.month);
```

11)What is setInterval, setTimeout
Ans . The setTimeout function is used to call a function after certain amount of time.
        The setTimeout function take a two argument first is callback function and second is time in millisecond.

```
setTimeout(()=>{
console.log(“hello”),3000})
```

SetInterval : The setInterval is used to call a function repeatedly at a specified interval of time

```
setInterval(()=>{
console.log(“hello”),3000})
```

12)What is _dirname and __filename
Ans The __dirname in a node script returns the path of the folder where the current JavaScript file resides. __filename and __dirname are used to get the filename and directory name of the currently executing file.Filename return the file name of the current module.including the absolute path of the module file

13)What is synchronous/Blocking and Asynchronous/Non-blocking code in node.js
Ans   Synchronous Blocking : The code which runs via line by line execution is called synchronous blocking.It means the line of code written first will be executed first.

Asynchronous Blocking:A block of code line by line execution is not guareentee
Is called Asynchronous Bocking.

14)What is file system in node.js
Ans The Node.js file system module allows you to work with the file system on your computer.

Common use for the File System module:

- Read files
- Create files
- Update files
- Delete files
- Rename files

Every method in the fs module has synchronous as well as asynchronous forms. Asynchronous methods take the last parameter as the completion function callback and the first parameter of the callback function as error. It is better to use an asynchronous method instead of a synchronous method, as the former never blocks a program during its execution

14)	What are the differenct type of flag using in node.js
And	Feature flags help us turn a feature on or off without the need to deploy any code. Node. js feature flags also enable the same.

15)	What is stream in node.js? Explain the types of stream
Ans Streams are objects that let you read data from a source or write data to a destination in continuous fashion. In Node.js, there are four types of streams −

Readable − Stream which is used for read operation.
Writable − Stream which is used for write operation.
Duplex − Stream which can be used for both read and write operation.
Transform − A type of duplex stream where the output is computed based on input.

16) How to pipe stream in node.js

Ans we can use pipe stream to connect two file and transfor data from one file to Another file.

```
var fs = require("fs");//step 1
var readStream = fs.createReadStream("D://datainput.txt"); //step 2
var writeStream = fs.createWriteStream("D://dataOutput.txt"); //step 3
readStream.pipe(writeStream); //step 4
```

in step 1 we require fs and in 2nd step we read file and in 3rd we point second file and in 4th step we transfer data from one file to another.

17) What is request and response in node.js

Ans Request and Response object both are parameter of the callback function. The express. js request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on. Request are come from the user and accoding to the user request.we give the response to user from server.

18) What is package.json and package.lock

Ans

package.json	package.lock.json
It contains basic information about the project.	It describes the exact tree that was generated to allow subsequent installs to have the identical tree.
It is mandatory for every project.	It is automatically generated for those operations where npm modifies either node_modules tree or package.json.
It records important metadata about the project.	It allows future devs to install the same dependencies in the project.
It contains information such as name, description, author, script, and dependencies.	It contains the name, dependencies, and locked version of the project.

19) What is npm ? How to install dependency/module at application level and Environment elevel

Ans Node Package Manager (NPM) is a command line tool that installs, updates or uninstalls Node.js packages in your application. It is also an online repository for open-source Node.js packages. The node community around the world creates useful modules and publishes them as packages in this repository.

Application level add dependency: npm install <package name>
Enviromental level add dependency: npm install –g <package name>

20) How do you manage packages in your node.js project

21) How Node.js is better then other framework
Ans JavaScript is used for both front-end and back-end development, making the language more consistent across the entire application. This is in contrast to most applications, which use different languages for the front-end (like HTML, CSS, and JavaScript) and back-end (like PHP, Ruby on Rails, or Java).

Js makes development simple because of its non-blocking I/O and event-based model. This simplicity results in short response time and concurrent processing, unlike other frameworks where developers use thread management.

Js runs on a chrome V8 engine which is written in C++. It enhances its performance highly with constant improvement.

With Node.js, we will use JavaScript in both the frontend and backend development that will be much faster.

Js provides sample libraries so that we don't need to reinvent the wheel.

What are the some commonly used timing features of Node.js

Ans The setTimeout function is used to call a function after certain amount of time.
The setTimeout function take a two argument first is callback function and second is time in millisecond.

```
setTimeout(()=>{
console.log(“hello”),3000})
```

SetInterval : The setInterval is used to call a function repeatedly at a specified interval of time

```
setInterval(()=>{
console.log(“hello”),3000})
```

setImmediate /clearImmediate: This timing feature of Node.js is used to set the execution of the code at the end of the event loop cycle.

nextTick: This timing feature sets the execution of code at the beginning of the next event loop cycle.

22) What is fork in node.js

ANS In Node.js, fork is a method that allows you to spawn a new process, using the child_process module. It is similar to the spawn method, but with some additional features.

When you call the fork method, a new Node.js process is created, and the current Node.js process becomes the parent process. The new process that is created is known as the child process. The child process can communicate with the parent process by sending messages using a built-in IPC (Inter-Process Communication) channel.

Here's an example of how you can use the fork method to create a child process:

javascript

```
const { fork } = require('child_process');

const childProcess = fork('child.js');

childProcess.on('message', message => {
  console.log('Received message from child:', message);
});

childProcess.send('Hello from parent!');
```

In this example, we're creating a new child process by calling fork('child.js'). We're also setting up an event listener for the message event, so that we can receive messages from the child process. Finally, we're sending a message to the child process by calling childProcess.send('Hello from parent!').

In the child process (child.js), you can set up an event listener for the message event, and send messages back to the parent process using the process.send() method.

javascript

```
process.on('message', message => {
  console.log('Received message from parent:', message);
  process.send('Hello from child!');
});
```

In this example, the child process is set up to receive messages from the parent process by listening for the message event. When a message is received, it logs the message to the console, and sends a message back to the parent process by calling process.send('Hello from child!').

23) How do you create a simple server in node.js that return hello world

Ans Import http from ‘http’

```
var server = http.createServer((request,response)=>{

  response.write(“welcome in Node js”);

})



Server.listen(3000,()=>{

Console.log(“server Started”);

})
```

First we have to import http we don’t need to install http it is core module of node,then we create server with the help of http reference ,then listen for any incoming request.

24) How many types of API functions are there in node.js
Ans API stands for Application Programming Interface. It consists of various communication protocols and subroutines that can be used by programs for inter-communication.

Types of API functions in Node.js:

Asynchronous, Non-blocking functions
Synchronous, Blocking functions

25) What is REPL and how to use it?
Ans A Read-Eval-Print Loop, or REPL, is a computer environment where user inputs are read and evaluated, and then the results are returned to the user.
We can use repl -->it is esay to use comand line arguments

First we can write node on our terminal than press Enter start .
5 types of express perform in repl
operator and operand
Multiline / and loop

26) What is purpose of module.exports

Ans The main purpose of module.exports is to achieve **modular programming**. Modular programming refers to separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. By not using the module.exports it becomes difficult to write a large program without modular/reusable code.

- Using module.exports we can separate business logic from other modules. In other terms, we can achieve **abstraction** using it.

- By using it becomes easy to **maintain and manage** the code base in different modules.

27) What is an event-loop in Node.js

Ans

28) If Node.js is single threaded then how does it handle multiple request/ concurrency

Ans NodeJS receives multiple client requests and places them into EventQueue. NodeJS is built with the concept of event-driven architecture. NodeJS has its own EventLoop which is an infinite loop that receives requests and processes them. EventLoop is the listener for the EventQueue.

29) Differenctiate between process.nextTick() and setImmediate()

Ans set Immediate adds call backs to the event queue that are executed during the check phase, whereas process. nextTick executes callbacks immediately after the current phase.

30) What is middleware

Ans *Middleware* functions are functions that have access to the request object (req), the response object (res), and the next function in the application's request-response cycle.The next function is a function in the Express router which, when invoked, executes the middleware succeeding the current middleware.

Middleware functions can perform the following tasks:
Execute any code.
Make changes to the request and the response objects.
End the request-response cycle.
Call the next middleware in the stack.

If the current middleware function does not end the request-response cycle, it must call next() to pass control to the next middleware function. Otherwise, the request will be left hanging.

31) Explain what a reactor pattern in node.js

Ans **Reactor Pattern** is used to avoid the blocking of the Input/Output operations. It provides us with a handler that is associated with I/O operations. When the I/O requests are to be generated, they get submitted to a demultiplexer, which handles concurrency in avoiding the blocking of the I/O mode and collects the requests in form of an event and queues those events.

32) Describes the exit code of Node.js

Ans

33) What is an EventEmitter in node.js

Ans To use the EventEmitter module, you need to first create an instance of the EventEmitter class. You can then use the instance's methods to emit events and register listeners for those events.

```
const EventEmitter = require('events');

// Create an instance of EventEmitter
const myEmitter = new EventEmitter();

// Register an event listener
myEmitter.on('myEvent', () => {
  console.log('Event occurred!');
});

// Emit the event
myEmitter.emit('myEvent');
```

34) What is a thread pool and which library handles it in node.js

Ans **Java Thread pool** represents a group of worker threads that are waiting for the job and reused many times.

In the case of a thread pool, a group of fixed-size threads is created. A thread from the thread pool is pulled out and assigned a job by the service provider. After completion of the job, the thread is contained in the thread pool again.

35) What is purpose of NODE_ENV

36) How would you connection mongodb database to node application

Ans Const mongoose=require(“mongoose”);’

```
Mongoose.connect(‘mongodb://localhost:27017/<database_name>’)
.then(()=>{
    //here we can write our routes
}).catch(()=>{
    //handle error
});
```

37) What are different type of http request

1. GET: Used to retrieve a resource from a server. GET requests are read-only and should not modify any data on the server.
2. POST: Used to submit data to a server to create a new resource or trigger a specific action. POST requests often involve sending data in the request body.
3. PUT: Used to update or replace an existing resource on the server. PUT requests send the entire updated representation of the resource.
4. PATCH: Similar to PUT, PATCH is used to update an existing resource, but only sends the modified fields or changes instead of the entire representation.
5. DELETE: Used to delete a specified resource on the server.
6. HEAD: Similar to a GET request, but only retrieves the response headers without the actual content. It is useful for retrieving metadata or checking the status of a resource.
7. OPTIONS: Used to retrieve the supported HTTP methods, headers, and other capabilities for a specified resource

38) What is difference between get and post

Ans

1) In case of Get request, only limited amount of data can be sent because data is sent in header.	In case of post request, large amount of data can be sent because data is sent in body.
2) Get request is not secured because data is exposed in URL bar.	Post request is secured because data is not exposed in URL bar.
3) Get request can be bookmarked.	Post request cannot be bookmarked.
4) Get request is idempotent . It means second request will be ignored until response of first request is delivered	Post request is non-idempotent.
5) Get request is more efficient and used more than Post.	Post request is less efficient and used less than get.

39) What is query string and how to send the data in get request

The "query string" is defined as a question mark followed by the parameters and their values.

40) What is the use body parser

Ans The body-parser middleware is used to parse this request body and make it available as a JavaScript object in req.body. The body-parser middleware supports different types of request body data, including JSON, URL-encoded data, and raw text data.

41) How to set the path of static file in express?

Ans

42) What are types of Middleware in express ? Explain with suitable example

Ans Type of Middleware

Application-level Middleware
Router-level Middleware
Error-handling Middleware
Built-in Middleware
Third-party Middleware

Application-level middleware: The application-level middleware method is used to bind to the app object using app.use() method. It applies on all routes.

```
//This middleware will execute for each route.
app.use(function (req, res, next) {
  console.log('Current Time:', Date.now())
 next()
})
```

Router-level Middleware: The router-level Middleware is used to bind to a specific instance of express.Router().

Built-in Middleware: The built-in Middleware was introduced with version 4.x. It ends the dependency on Connect.

There are the following built-in middleware functions in Express.js:

static: It is used to serve static assets such as HTML files, images, etc.
json: It is used to parse the incoming requests with JSON payloads. It is available with Express 4.16.0+
urlencoded: It is used to parse the incoming requests with URL-encoded payloads. It is available with Express 4.16.0+

Third-party Middleware: There are many third-party middleware available such as:

Body-parser
Cookie-parser
Mongoose
Sequelize
Cors
Express-validator

43) Does order of middleware matters in express.

Ans Order matters in middleware because middleware functions are executed in a sequence and you'd need to execute one before the other in order to utilize its functionality.

44) What is express.js

Ans Express.js is a small and flexible framework of Node.js. . It is used to build a single page, multipage, and hybrid .
web application. It provides a set of features and tools to build web applications and APIs easily.It simplifies the process of handling HTTP requests, routing, middleware integration, and serving static files. It is widely used for building server-side applications in JavaScript.

45) What are some distinctive features of Express

46) is Express.js fron-end or backend framework?

Ans Express.js or Express is a JavaScript backend framework. It is mainly designed to develop complete web applications (single-page, multi-page, and hybrid web applications) and APIs. Express is the backend component of the **MEAN stack** where **M stands for MongoDB**, which handles database; **E stands for Express,** which handles backend; **A stands for AngularJS**, which is for the front-end, and **N stands for Node**.

47) Why do we use express.js

Ans

48) What is difference between express.js and node.js

Ans

Feature	**Express.js**	**Node.js**
Definition	Express.js is a lightweight and fast backend web application framework for Node.js.	Node.js is an open-source and cross-platform that is used to execute JavaScript code outside of a browser.
Usage	Express.js is used to develop complete web applications such as single-page, multi-page, and hybrid web applications and APIs. It uses approaches and principles of Node.js.	Node.js is used to build server-side, input-output, event-driven apps.
Features	Express has more features than Node.js.	Node.js has fewer features as compared to Express.js.
Building Block	Express.js is built on Node.js.	Node.js is built on Google's V8 engine.

Written in	Express.js is written in JavaScript only.	Node.js is written in C, C++, and JavaScript language.
Framework/Platform	Express.js is a framework of Node.js based on its functionalities.	Node.js is a run-time platform or environment designed for server-side execution of JavaScript.
Controllers	Express.js is assigned with controllers.	Node.js is assigned with controllers.
Routing	Routing is provided in Express.js.	Routing is not provided in Node.js.
Middleware	Express.js uses middleware to arrange the functions systematically on the server-side.	Node.js doesn't use any such provision of middleware.
Coding	Express is easy to code and requires less coding time.	Node.js requires more coding time as compare to Express.js.

49) What do you understand by Scaffolding in Express.Js

Ans Scaffolding is a technique used for creating the skeleton structure of an application. It facilitates users to easily create their public directories, routes, views, etc., or a web application skeleton. Generally, users manually create their public directory, add middleware, create separate route files, etc. Using a scaffolding tool, they can set up all these things to directly get started with building their application.

50) Which are the argument available to an Express.Js route handler function

Ans Following are the arguments that are available to an Express.js route handler-function:

Req: the request object

Res: the response object

Next (optional): It is a function employed to pass management to one of the above route handlers.

51) How can you allow CORS in Express.jS?

Ans when two process are running at two different port ,then at the time send request one port to another port then it will generate error cross origin request ,so that save from this problem we can use this dependency in our application.

First we have to require :-const cors=require("cors");

Then we use it:-app.use(cors());

52) How can you deal with Error handling in Express.js? Explain with an example

Ans In Express.js, errors can occur during the request-response cycle for various reasons, such as database errors, network failures, validation errors, or exceptions thrown in your application's code. To handle errors effectively, Express provides multiple mechanisms:

1. **Error-handling middleware**: You can define a middleware function with four parameters (`err`, `req`, `res`, `next`) to handle errors. This middleware should be defined after all other routes and middleware in your application. It can catch errors passed through the `next()` function and perform error-specific actions, such as logging, sending error responses, or executing error recovery logic.

Example:

```javascript
app.use((err, req, res, next) => {
  // Handle the error
  console.error(err);

  // Send an error response to the client
  res.status(500).json({ error: 'Internal Server Error' });
});
```

2. **Async/await error handling**: When using async/await syntax for handling asynchronous operations, you can wrap your route handlers or middleware functions in a `try-catch` block. Any errors thrown within the `try` block can be caught in the `catch` block and processed accordingly.

Example:

```javascript
app.get('/example', async (req, res, next) => {
  try {
    // Perform asynchronous operations
const result = await someAsyncOperation();
    res.json(result);
  } catch (err) {
    // Handle the error
    console.error(err);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});
```

3. **Custom error classes**: You can define custom error classes that extend the built-in `Error` class or any other predefined error class. These custom error classes can carry additional information or be used for specific types of errors in your application.

Example:

~~~
        ```javascript
        class CustomError extends Error {
          constructor(message, statusCode) {
            super(message);
  this.statusCode = statusCode;
          }
        }

        app.get('/example', (req, res, next) => {
          try {
            // Perform operations
            if (someCondition) {
              throw new CustomError('Custom error occurred', 400);
            }
            res.json(result);
          } catch (err) {
            // Handle the custom error
            console.error(err);
            res.status(err.statusCode || 500).json({ error: err.message });
          }
        });
~~~

53) Write a code to start serving static file Express.JS

Ans To serve static files such as images, CSS files, and JavaScript files, use the express.static built-in middleware function in Express.

Signature:-

```
App.use(express.static(‘public’));
app.use(express.static(path.join(__dirname, 'public')))
```

54) How can we render plain HTML in express.

Ans simply we can render htlm plain file in express by using sendFile() method using response object.

```
// Importing modules
const express = require('express');
const path = require('path');
const app = express();
```

```
 app.get('/', (req, res) => {
res.sendFile(path.join(__dirname, '/public/index.html'));
 });
 app.get('/products', (req, res) => {
   res.sendFile(path.join(__dirname, '/public/products.html'));
 });
 app.listen(3000, () => {
   console.log('Server is up on port 3000');
 });
```

55) How can we send the data while rendering page in express

Ans To send data while rendering a page in Express, you can use a template engine like EJS or Pug. Template engines allow you to combine HTML with dynamic data to generate dynamic web pages.

To send data while rendering a page in Express, you can use the res.render() method. This method allows you to pass data to your view templates and render them dynamically.

```
   const express = require('express');
const app = express();

app.set('view engine', 'ejs');
app.set('views', 'views');

   app.get('/', (req, res) => {
  const name = 'John Doe';
  res.render('index', { name });
});
```

56) How to enable debugging in express app?

57) What is routing and How routing works in express

Ans Routing in Express refers to the process of determining how an application responds to a client request for a specific URL (or route) and HTTP method.

In Express, routing is achieved using the app.METHOD(path, handler) functions, where METHOD is an HTTP method (such as GET, POST, PUT, DELETE, etc.), path is a URL pattern, and handler is a function that gets executed when a request matching the defined method and path is received.

58) How dynamic routing works in express.js

Database :-

1) What do you understand by term database ?

Ans A **database** is an structured collection of data, so that it can be easily accessed and managed.

A database typically consists of tables, which are composed of rows and columns. Each row represents a record or an instance of data, while each column represents a specific attribute or characteristic of that data. The structure and organization of the data in a database are defined by a database schema, which specifies the tables, their columns, data types, relationships, and constraints.

The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

There are many **databases available** like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

2) Define DBMS

Ans DBMS stands for Database Management System. It refers to a software system that allows users to define, create, manipulate, and manage databases. DBMS provides an interface for interacting with the database, enabling users to perform various operations such as storing, retrieving, updating, and deleting data.

3)Define RDBMS

Ans RDBMS stands for Relational Database Management System. It is a specific type of Database Management System (DBMS) that is based on the relational model of data. RDBMS is designed to store, manage, and manipulate data organized in tables with predefined relationships between them.

4)What are various type of relationships in database? define them

Ans In a relational database, various types of relationships can exist between tables. These relationships define how data in one table is related to data in another table. The most common types of relationships in a database are:

1)One-to-One
2)One-to-Many
3)Many-to-one
4)Many-to-Many

1)one-to-one : It is used to create a relationship between two tables in which a single row of the first table can only be related to one and only one records of a second table. Similarly, the row of a second table can also be related to anyone row of the first table.

2)One to Many Relationship: It is used to create a relationship between two tables. Any single rows of the first table can be related to one or more rows of the second tables, but the rows of second tables can only relate to the only row in the first table. It is also known as a **many to one** relationship.

3)Many to Many Relationship: It is **many to many** relationships that create a relationship between two tables. Each record of the first table can relate to any records (or no records) in the second table. Similarly, each record of the second table can also relate to more than one record of the first table. It is also represented an **N:N** relationship.

3) Explain normalization

Ans The process of converting complex data structure into simple data structure is called normalization.

In this process the relations with anomalies can be decomposed to produce smaller,well defned structure manner.

Normalization is a process in database design that aims to eliminate data redundancy and maintain data integrity.

4) What are different types of normalization?

Ans 1NF : A relation is in 1NF if every attribute is a single-valued attribute or it does not contain any multi-valued or composite attribute, i.e., every attribute is an atomic attribute. If there is a composite or multi-valued attribute, it violates the 1NF. To solve this, we can create a new row for each of the values of the multi-valued attribute to convert the table into the 1NF.

Let's take an example of a relational table <EmployeeDetail> that contains the details of the employees of the company.

<EmployeeDetail>

Employee Code	Employee Name	Employee Phone Number
101	John	98765623,998234123
101	John	89023467
102	Ryan	76213908
103	Stephanie	98132452

Here, the Employee Phone Number is a multi-valued attribute. So, this relation is not in 1NF.

To convert this table into 1NF, we make new rows with each Employee Phone Number as a new row as shown below:

<EmployeeDetail>

Employee Code	Employee Name	Employee Phone Number
101	John	998234123
101	John	98765623
101	John	89023467
102	Ryan	76213908

Employee Code	Employee Name	Employee Phone Number
103	Stephanie	98132452

2NF:

5) What is SQL ?

Ans SQL stands for Structured Query Language.SQL lets you access and manipulate databases.It maintain structure Data. It store the data in the form of tables.

It is a standardized programming language used for managing and manipulating relational databases. SQL provides a set of commands and syntax for interacting with databases, performing various operations such as querying data, inserting new data, updating existing data, and deleting data.

6) How many SQL statement used ? Define them

Ans SQL (Structured Query Language) provides several types of statements for interacting with relational databases. The main categories of SQL statements are as follows:

Data Querying Statements:

SELECT: Retrieves data from one or more tables based on specified conditions.
FROM: Specifies the table(s) from which to retrieve data.
WHERE: Filters the data based on specified conditions.
GROUP BY: Groups the data based on specified columns.
HAVING: Filters the grouped data based on specified conditions.
ORDER BY: Sorts the data in ascending or descending order based on specified columns.
Data Manipulation Statements:

INSERT INTO: Inserts new data into a table.
UPDATE: Modifies existing data in a table.
DELETE: Deletes data from a table.
MERGE: Combines INSERT, UPDATE, and DELETE operations into a single statement for conditional data manipulation.
Data Definition Statements:

CREATE TABLE: Creates a new table with specified columns, data types, and constraints.
ALTER TABLE: Modifies an existing table structure, such as adding or dropping columns, altering column definitions, or adding constraints.
DROP TABLE: Deletes an entire table and its data.

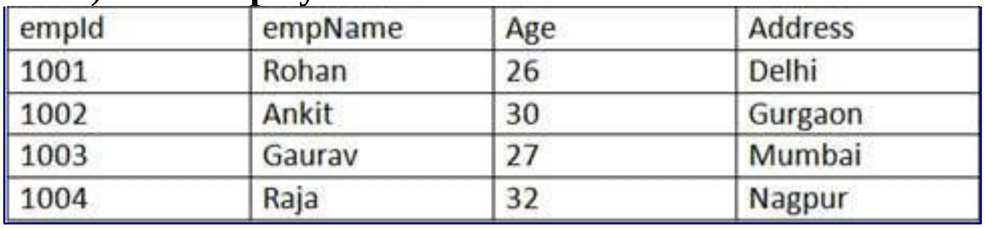
CREATE INDEX: Creates an index on one or more columns to improve query performance.
DROP INDEX: Removes an index from a table.
Data Control Statements:

GRANT: Grants specific privileges to database users.
REVOKE: Revokes previously granted privileges from database users.
Transaction Control Statements:

COMMIT: Saves all the changes made in a transaction.
ROLLBACK: Discards all the changes made in a transaction.
SAVEPOINT: Sets a savepoint within a transaction to allow partial rollbacks.
Data Manipulation Language (DML) Statements:

SET: Assigns values to variables.
SELECT INTO: Inserts the result of a SELECT query into a new table.
UNION: Combines the results of two or more SELECT queries.
INTERSECT: Retrieves the common rows between the results of two or more SELECT queries.
EXCEPT: Retrieves the rows from the first SELECT query that do not exist in the result of the second SELECT query.
These are some of the commonly used SQL statements. The specific capabilities and syntax of SQL may vary slightly between different database management systems, but the core functionality remains consistent across most implementations.

7) Enlist some commands of DDL , DML and DCL
8) Explain the term record, field and table in database
9) What is view in database
10) What are advantage and disadvantage of view?
11) Define Entity , Entity Type and Entity Set
12) Define a weak entity set
13) What do you undestand by join ? Explain different types of Join
14) What is self join
15) Explain Primary key, Foreign key, unique key, composite key, super key
16) What is projection in database?
17) What is difference between delete, truncate and drop command
18) Based on given table, solve the following query ?
19) **Employee table**



empId	empName	Age	Address
1001	Rohan	26	Delhi
1002	Ankit	30	Gurgaon
1003	Gaurav	27	Mumbai
1004	Raja	32	Nagpur

a) Write the SELECT command to display the details of the employee with empid as 1004.
b) Write the SELECT command to display all the records of table Employees.

c) Write the SELECT command to display all the records of the employee whose name starts with the character ‘R’.

d) Write a SELECT command to display id, age and name of the employees with their age in both ascending and descending order.
e) Write the SELECT command to calculate the total amount of salary on each employee from the below Emp table.

22.What is mongodb ? What are advantage of mongodb

Mongodb is open source document oriented database , it is designed to store large scale of data and it allows you to work with that data very efficientl it store the data structure or not structure both , it store the data in the form of bson (binary json)

Advantage : there is no need to sql injection
It dose not support to join opration
It also support multipal transition
It provide high performance and scalability

23.What is difference between SQL and NoSQL database

Sql	NoSql
1 it store structure data	It store structured and unstracture data
2 in sql data stored in the form or table	in store data in the form of collaction
3 in terms of performance it is slower Then Nosql	in terms of it is faster then RDBMS
4 it has a predefine schema	it has dynamic schema

24.What is document in mongodb?

Document : is a way to organize and store the data as a set of field -value (key-value) pairs

25.What is collection in mongodb

Collactions just like a tables in relational database . they also store the data , but in the form of document . a single database is allowed to store multiple collactions.

26.What is mongo shell

MongoShell is a command-line interface for MongoDB, It is a powerful tool that allows developers to interact with MongoDB databases using JavaScript commands.

27.What are some features of mongodb

Supports map reduce and aggregation tools.
Uses JavaScript instead of Procedures.
It is a schema-less database written in C++.
Provides high performance.

1. Document-Oriented: MongoDB stores data in flexible and self-descriptive documents.
2. Scalability: MongoDB can handle large amounts of data by distributing it across multiple servers or clusters.
3. High Performance: MongoDB is optimized for fast data read and write operations.
4. Flexible Schema: MongoDB allows you to store data without predefined structures, making it easy to adapt to changing requirements.
5. Rich Query Language: MongoDB provides a powerful and easy-to-use language for retrieving and manipulating data.
6. Replication and High Availability: MongoDB ensures that your data is always accessible even if some servers fail.

28.How to add data in mongodb ?

1 To insert a single document, use db. collection. insertOne() .
2 To insert multiple documents, use db. collection. insertMany() .

29.How do you update data in mongodb

1. To update a single document, use db. collection. updateOne() .
2. To update multiple documents, use db. collection. updateMany() .
3. To replace a document, use db. collection. replaceOne() .

30.How do you delete document

1. To delete multiple documents, use db. collection. deleteMany() .
2. To delete a single document, use db. collection. deleteOne() .

31.How to perform query in mongodb

1. Connect to MongoDB: Establish a connection to your MongoDB server using a client or driver.

2. Choose a Collection: Select the collection where you want to perform the query.
3. Specify Query Criteria: Define the conditions for your query, such as finding documents with a specific value in a field.
4. Perform the Query: Use the find() method on the collection, passing in the query criteria. This returns a set of matching documents.
5. Retrieve Results: Iterate over the results to access the matching documents and process them as needed.

```
  const MongoClient = require('mongodb').MongoClient;

// Connection URL
const url = 'mongodb://localhost:27017';

// Database Name
const dbName = 'mydatabase';

// Connect to MongoDB
MongoClient.connect(url, function(err, client) {
  if (err) {
    console.error('Error connecting to MongoDB:', err);
    return;
  }
     console.log('Connected to MongoDB');
      // Select a collection
  const db = client.db(dbName);
  const collection = db.collection('users');
  // Specify query criteria
  const query = { age: 25 };
  // Perform the query
    collection.find(query).toArray(function(err, documents) {
    if (err) {
      console.error('Error executing query:', err);
      return;
    }
          console.log('Query results:', documents);
          // Close the connection
    client.close();
  });
});
```

32.What are datatypes in mongodb

String , Integer , Double ,Boolean , Null ,Array , Object , Object Id , Undefined , Binary Data , Date, Timestamp

33.What is index and how to create index in mongodb

34.Explain the set modifier in mongodb

35.Does mongodb support primary key and foreign key relationship ?
MongoDB does not have built-in support for primary key and foreign key relationships like traditional relational databases. MongoDB is a NoSQL database that follows a document-oriented data model, which is different from the table-based structure of relational databases.

36.Explain the structure of ObjectId in mongodb
Ans :-ObjectID is a 12-byte BSON type. These are:
4 bytes value representing seconds
- 3 byte machine identifier
- 2 byte process id
- 3 byte counter

Is it true that MongoDB uses BSON to represent document structure?
37.What are Indexes in MongoDB?
In MondoDB, Indexes are used to execute query efficiently. Without indexes, MongoDB must perform a collection scan, i.e. scan every document in a collection, to select those documents that match the query statement. If an appropriate index exists for a query, MongoDB can use the index to limit the number of documents it must inspect.
38.By default, which index is created by MongoDB for every collection?
In-which language mongodb is writtern ?
MongoDB is primarily written in C++. The core components of MongoDB, including the database server (mongod), query engine, storage engine, and networking stack, are implemented in C++. This choice of language allows MongoDB to achieve high performance and efficient data processing.

39.What will happen when you remove a document from database in MongoDB? Does MongoDB remove it from disk?
40.What is capped and uncapped collection in mongodb
41.What is lookup in mongodb
42.How to get record of second max age record from mongodb collection
43.What is mongoose ?
44.What is mongoose Schmea
45.How to set reference in mongoose Schema
46.How populate works in mongoose ?
47.List the name of the mongoose method to perform CRUD operation
48.What is default connction pool size in mongoose
49.How to set connection pool size in mongoose
50.What is connection pool and what are advantage of using connection pool size

Event-driven = event-driven programming It allows applications to efficiently handle concurrent and asynchronous operations by utilizing a single-threaded event loop.