1. **What is Javascript**

Ans : JS is a scripting language which is used to make web pages dynamic or interactive .

Scripting Language :- Scripting language means that is run directly on browser without needing any compilation.

1. **How Borwser Execute Javascript.**

Ans :- Every js code is executed by v8 and inside this there is a callstack which is responsible to execute the js code.

1. **What are features of Javascript.**

Ans : Js is dynamically typed language

It is function oriented language

Object based language

It is scripting language

It is cross platform

It is versetile.

It is lightweight.

It is a case sensitive language

1. **What is DOM ?**

Ans :- DOM is stand for Document Object Model it is a programming interface that represent structured documents like **HTML**  and **XML** as a tree of object. It define how to access , define and manuplate the document elements using scripting langauag like JavaScript.

1. **How to perform click event on Dom Using Javascript**

Ans :-

<!DOCTYPE html>

<html lang="en">

<body>

     <button><h1 onclick="changeText(this)">click</h1></button>

     <script>

        function changeText(ele){

            if(ele.innerHTML === 'hello')

                ele.innerHTML = 'click'

            else

                ele.innerHTML = 'hello'

        }

     </script>

</body>

</html>

1. **What is alert() and confirm() ?**

Ans : alert is used to show alert message with OK button and confirm is used to confirm the some the after clicking on button with OK and cancle button .­

**7 What is difference between undefined and null**

Ans :-

**Undefined** = when you use any variable without define it it show undefined .The type of undefined is undefined

**Null** = null is a special value that represents an empty or unknown value and the type of null is object.

**For example** let x = null ;

It show that the vriable is empty at that moment and have some value later .

1. **What is difference between == and === .**

Ans :-

== is compare the two variable without checking the variable type

=== is comapare two value and check the type of variable.

1. **What is function . Can we overload function in javascript.**

Ans : function is a block of code that is used to perform a specific task when it is called .

In js function overloading is not supporting .It support overriding ,so if you define the function with the same name it execute the last one and it override the previously defined function.

1. **What is callback ? Explain with suitable example.**

And : When a function is passed as an argument to another function is known as callback.It is executed after some some operation have complete their execution .

function print(msg,callback){

console.log(msg)

callback() ;

}

function hello(){

console.log(“ !Hello World ”) ;

}

print(“hii”,hello())

1. **What is promise? What are advantage of promise over callback.**

Ans : In js , promise is an object that produce a single value somtime in future.If the promise is successful it produce a resolve value and if the promise is unsuccessful it produce a rejected value and will produce a result that why promise is reject .

**The advantage of promise over callback is**

It resolve the callback hell problem .By using promise error handling is easy using catch block and it syntax is user friendly and easy to understand.

1. **What is callback hell**.

Ans :- Callback hell is referred to “Pyramid of doom” .In js it happens when you have multiple nested callback functions.This make code hard to read and maintain.

Function first(callback){

setTimeout(()=>{

console.log(‘first task done’)

callback()

},1000}

)

Function second(callback){

setTimeout(()=>{

console.log(‘second task done’)

callback()

},1000}

)

Function third(callback){

setTimeout(()=>{

console.log(‘third task done’)

callback()

},1000}

)

first(()=>{

second(()=>{

third(()=>{

console.log(“All task done”)

})

})

})

1. **What is closure ?**

Ans :- A clouser is the combination of function bundled together with references to its surrounding state.

And clouser give you access to an outer function scope from an inner function scope .

**Advantage of callback :**

Create funton factories

Maintain states

Data encapsulate

Callbacks

1. **What is IIFEs(Immediately Invoked Function expression).**

Ans : IIFE stands for immediately invoked function expression ,It is a function that are execute immediately after the define.IIFE is used to create local scope for variable to prevent them from polluting the global scope.

1. **What is annonymous function ?**

Ans :- the meaning of word ‘annonymous’ defining something that has no name or no identity.In js an anonymous function is the type of function that has no name or no identity.When we create anonymous function ,it is declared without any identifier.

For example :-

let x = function(){

return 10+10 ;

}

Console.log(x)

1. **What is hoisting in javascript.**

Ans :-Hoisting is the default behaviour of js where variable and function declaration are move on the top of their respective scope at the time of execution.

1. **What is difference between var, let and const keyword.**

Ans :- const is block scope ,it dosen’t allow redaclaration and reassingnment .

Let is block scope, it allow reassignment but dosen’t allow redeclaration.

Var is a function and it is hoisted and it allow redeclaration and reassignment the variable.

1. **Explain use strict**

Ans :- The purpose of “use strict” is to indicates that your code executed in strict mode .For example you can not use the undeclared variables or function or object in js .

1. **What is event bubbling and event capturing**

Ans :

event bubbling : so an event triggred specific element that start from that element and propagate to the outer most element and so on up to the root element.

Event capture :- Event capture is opposite to the event bubbling ,in this event start from the outer most element to the inner most element .

Event Target :- The event target is the element on which event originally occurred .This can be use access by ‘target’ property of event object.

Let ele = document.getElementById(‘#hello’)

Ele.addEventListener(‘click’,()=>{

Console.log(“hello from element”)

},true)

1. **What are the primitive datatype in javascript**

Ans : string ,number ,Boolean ,null , undefined ,bigint , symbol .

1. **What are different types of popup boxes available in javascript**

Ans : there are three types of pop up box in js

Alert , cofirm , prompt ;

1. **What will happen if an infinite while loop is run in javascript**

Ans :- Hang the browser : means the browser is unresponsive prevent any other option like rendering UI

high cpu usage : it slow down the entire system , affect the performance of another application ,

1. **List HTML DOM mouse events**

Ans :

onclick,ondblclick, onmouseover, onmouseout, onmouseup,onmousemove

1. **How to get the last index of a string in javascript**

Ans : by use of lastIndexOf() method

1. **Describe negative infinity in javascript**

Ans :

In js negative infinity is a special value that represent negative value . It is the global value of ‘Number’ object .It show the lower number value in comaper to another number in js.

1. **Explain await and aync ? How to use await and async?**

Ans :- async and await are the syntactic sugar in js that allow to write asynchronous code in readable and mangable way.

Async function is declared with async keyword. It always return promise and you can use await inside the async function to pause the execution .

1. **How to handle the excption in javascript.**

Ans :- By using the catch block we can handle the exception in js

**Why we use Symbol data type in js**

Symbol is a unique identifier and immutable datatype in js that is primarily used for unique identifier for object properties. Symbol data type is created using Symbol() function

Let sym1 = Symbol(‘description’)

Let sym2 = Symbol(‘description’)

Const myObj = {

[sym1] : “hello”,

[sym2] : “hello2”

}

1. **What is Node.js?**

Ans : Node js is runtime environment for js , that is used execute js code server side .It is single threaded by default and use event loop to handel request for client side .

1. **What are the limitation of Node.Js**

Ans :  it does not support rendering , limited standard library means dependencies install again and again ,security concerns means we install many dependencies if any on dependency is miss it throw an error .

 **Single-threaded**: Heavy CPU tasks se server slow ho sakta hai.

 **Callback Hell**: Nested callbacks se code complex ho jata hai.

 **Weak Typing**: Type-related bugs easily aa sakte hain.

 **Limited Standard Library**: External libraries ka zaroori use hota hai.

 **Not for CPU-bound Tasks**: Complex calculations me slow performance.

1. How Node.Js Works

Ans :

1. **How Node.js is single threaded**

Ans : node.js does not follow multi-threaded request/response model.It follow the Single-threaded with the event-loop model.This event-loop handle blocking and non-blocking request.Because of this nods.js handle more request with ease.When Node. js encounters an I/O operation (like reading a file or making a network request), it doesn't halt the entire application.

1. **Explain any five built-in package/Dependency name in node.js**

Ans :

Express

Body-parser

Cors

Multer

Bcrypt

Express-validator

Jwt

url

Path

Nodemon

1. **What is module in node.js**

Ans : module is a block of code that is used to communicate with external application on their related functionality.Module is used by requiring them in our application by using require() function.

1. **what is Module.exports**

Ans : In node.js module.exports is a special type of object and it is used in all js file of node.js by default . It is use to export functions,objects and primitive values from a module  and use in another file of js by importing them.

1. **How to create server in node.js**

Ans : To import http from http we can create server

 For ex : const http = require(‘http’) ;

Const server = http.createServer((req,res)=>{

// set header

res.writeHead(200,{‘content type’ : ‘application/JSON’});

//set response

res.end(‘hello world’)

}).listen(3000)

1. **How node.js handle multiple request**

Ans : Node js use event-driven , non-blocking i/o model to handle multiple request .

1. **How to use url module in node.js**

Ans : const url = require('url') ;

const parseUrl = url.parse('https://dummyjson.com/docs/products')

console.log(parseUrl);

const formatUrl = url.format(parseUrl)

console.log(formatUrl);

1. **What is setInterval, setTimeout**

Ans :

1. **What is \_\_dirname and \_\_filename**

Ans :- \_\_dirname is the variable in js that represent the directory in which multiple documents are together. It show the actual path of directory in which js file are store .

\_\_filename : is the variable that represent the current file in js. It return the current path of file including file name .

1. **What is synchronous/Blocking and Asynchronous/Non-blocking code in node.js**

Ans :

1. **What is file system in node.js**

Ans : in node js fs system is built-in module in node js which allow to access , write , delete or update the file in our operating system.Commonly used features of fs module is readFile() , writeFile() and replace the file if already exist .

1. **What are the differenct type of flag using in node.js**

Ans : Node js fs module use flags to indicate that how file is open it use flag to specify that file is open in writing mode or reading mode append mode and many more .

Some flage are : ‘r’ , ‘w’ ,’r+’,’a’,’w+’,

1. **What is stream in node.js? Explain the types of stream**

Ans :- Stream in node js is a data handling concept It allows you to process data piece-by-piece, which is especially useful for working with large files or incoming data from the network, without loading the entire data into memory at once.

There are four types of stream in node js

Readable stream :

Writable Stream :

Duplex Steam :

Transform Strem : It is a special case of duplex stream that can modify the data when you read it or write it.

1. How to pipe stream in node.js

Ans : In Node.js, you can pipe streams to pass data from one stream to another. This is commonly used to read data from a readable stream and write it to a writable stream. The pipe() method facilitates this operation.

Here's an example of how to pipe a readable stream (e.g., reading from a file) to a writable stream (e.g., writing to another file):

javascript

Copy code

const fs = require('fs');

// Create a readable stream

const readableStream = fs.createReadStream('source.txt');

// Create a writable stream

const writableStream = fs.createWriteStream('destination.txt');

// Pipe the readable stream to the writable stream

readableStream.pipe(writableStream);

// Optionally, handle the 'finish' event

writableStream.on('finish', () => {

console.log('Data has been successfully piped to the destination file.');

});

In this example:

1. fs.createReadStream('source.txt') creates a readable stream from source.txt.
2. fs.createWriteStream('destination.txt') creates a writable stream to destination.txt.
3. readableStream.pipe(writableStream) pipes the data from the readable stream to the writable stream.

The pipe() method is efficient and handles backpressure automatically, ensuring that data is read and written at the appropriate rates without overwhelming the writable stream.

**45. What is request and response in Node.js?**

* **Request**: An object representing the HTTP request and has properties for the request query string, parameters, body, HTTP headers, etc.
* **Response**: An object representing the HTTP response that an Express app sends when it gets an HTTP request.

**46. What is package.json and package-lock.json?**

* **package.json**: A file that holds various metadata relevant to the project and is used to give information to npm that allows it to identify the project as well as handle the project's dependencies.
* **package-lock.json**: A file generated automatically for any operations where npm modifies either the node\_modules tree or package.json. It ensures the same dependency tree across all installations.

**47. What is npm? How to install dependency/module at application level and Environment level?**

* **npm (Node Package Manager)**: A package manager for JavaScript that lets you install, share, and manage dependencies.
  + **Application Level**: npm install <package-name>
  + **Global Level**: npm install -g <package-name>

**48. How do you manage packages in your Node.js project?**

* Using npm or yarn to install, update, and remove packages listed in the package.json file.

**49. How Node.js is better than other frameworks?**

* Non-blocking I/O, single-threaded event loop, efficient for I/O-heavy operations, vast package ecosystem via npm, and active community support.

**50. What are some commonly used timing features of Node.js?**

* setTimeout(), setInterval(), setImmediate(), process.nextTick().

**51. What is fork in Node.js?**

* A method to create a new instance of V8 engine to run multiple workers to execute code simultaneously.

**52. How do you create a simple server in Node.js that returns "Hello World"?**

javascript

Copy code

const http = require('http');

const server = http.createServer((req, res) => {

res.statusCode = 200;

res.setHeader('Content-Type', 'text/plain');

res.end('Hello World');

});

server.listen(3000, () => {

console.log('Server running at http://localhost:3000/');

});

**53. How many types of API functions are there in Node.js?**

* **Asynchronous (non-blocking)** and **Synchronous (blocking)** functions.

**54. What is REPL and how to use it?**

* **REPL (Read-Eval-Print Loop)**: An interactive shell that processes Node.js expressions.
  + Start it by typing node in the command line.

**55. What is the purpose of module.exports?**

* To export functions, objects, or primitives from a module so they can be reused in other files using require().

**56. What is an event-loop in Node.js?**

* A mechanism that handles and processes external events and converts them into callback invocations.

**57. If Node.js is single-threaded, then how does it handle multiple requests/concurrency?**

* Through the event loop and non-blocking I/O operations.

**58. Differentiate between process.nextTick() and setImmediate()**

* process.nextTick(): Executes the callback immediately after the current operation completes.
* setImmediate(): Executes the callback on the next iteration of the event loop.

**59. What is Node.js stream?**

Ans : Stream is

* A way to handle reading/writing files or receiving/sending data over the network efficiently. Types: Readable, Writable, Duplex, and Transform streams.

**60. What is middleware?**

* Functions that execute during the lifecycle of a request to the server. Each function can modify the request or response objects, end the request-response cycle, or call the next middleware function.

**61. Explain what a reactor pattern in Node.js is.**

* A pattern used for non-blocking I/O operations. It registers callbacks and handles I/O events asynchronously through an event loop.

**62. Describe the exit code of Node.js.**

* **0**: Success
* **1**: Uncaught fatal exception
* **2**: Unused
* **3**: Internal JavaScript parse error
* **4**: Internal JavaScript evaluation failure
* **5**: Fatal error
* **6**: Non-function internal exception handler
* **7**: Internal exception handler run-time failure

**63. What is an EventEmitter in Node.js?**

* A class that facilitates communication or data exchange between objects in Node.js. It allows us to bind events and execute actions when those events are triggered.

**64. What is a thread pool and which library handles it in Node.js?**

* A group of pre-initialized threads that wait for tasks to be assigned. The libuv library handles it.

**65. What is the purpose of NODE\_ENV?**

* An environment variable that specifies the environment in which a Node.js app is running (e.g., development, production).

**66. How would you connect a MongoDB database to a Node application?**

* Using the mongoose library:

javascript

Copy code

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/mydatabase', { useNewUrlParser: true, useUnifiedTopology: true });

**67. What are different types of HTTP requests?**

* GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS.

**68. What is the difference between GET and POST?**

* **GET**: Requests data from a server. Parameters are appended in the URL.
* **POST**: Sends data to the server. Parameters are sent in the body of the request.

**69. What is query string and how to send the data in GET request?**

* A string included in the URL to pass parameters. Use ?key=value format:

javascript

Copy code

http://example.com/page?name=Abhimanyu&age=24

**70. What is the use of body-parser?**

* A middleware to parse incoming request bodies in a middleware before handlers, available under req.body.

**71. How to set the path of static files in Express?**

javascript

Copy code

app.use(express.static('public'));

**72. What are types of Middleware in Express? Explain with a suitable example.**

* **Application-level**: Defined using app.use() or app.METHOD().
* **Router-level**: Defined using router.use() or router.METHOD().
* **Error-handling**: Defined with four arguments (err, req, res, next).
* **Built-in**: Provided by Express (e.g., express.static).
* **Third-party**: Provided by third parties (e.g., body-parser).

**73. Does order of middleware matter in Express?**

* Yes, middleware is executed in the order it is defined.

**74. What is Express.js?**

* A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

**75. What are some distinctive features of Express?**

* Middleware support, robust routing, template engines, HTTP utility methods.

**76. Is Express.js front-end or back-end framework?**

* Back-end framework.

**77. Why do we use Express.js?**

* To build web applications and APIs with ease, handling requests and responses efficiently.

**78. What is the difference between Express.js and Node.js?**

* **Node.js**: A runtime environment for executing JavaScript on the server side.
* **Express.js**: A framework built on top of Node.js to simplify server-side programming.

**79. What do you understand by Scaffolding in Express.js?**

* Automatically generating the basic structure of an application.

**80. Which are the arguments available to an Express.js route handler function?**

* req (request), res (response), and next (next middleware function).

**81. How can you allow CORS in Express.js?**

* Using the cors middleware:

javascript

Copy code

const cors = require('cors');

app.use(cors());

**82. How can you deal with Error handling in Express.js? Explain with an example.**

* Using error-handling middleware:

javascript

Copy code

app.use((err, req, res, next) => {

console.error(err.stack);

res.status(500).send('Something broke!');

});

**83. Write a code to start serving static files in Express.js.**

javascript

Copy code

app.use(express.static('public'));

**84. How can we render plain HTML in Express?**

javascript

Copy code

app.get('/', (req, res) => {

res.sendFile(\_\_dirname + '/index.html');

});

**85. How can we send the data while rendering a page in Express?**

javascript

Copy code

app.get('/', (req, res) => {

res.render('index', { title: 'Express' });

});

**86. How to enable debugging in an Express app?**

* Set the DEBUG environment variable:

bash

Copy code

DEBUG=express:\* node app.js

**87. What is routing and how routing works in Express?**

* Defining application endpoints (URIs) and how they respond to client requests. Routes are created using app.METHOD(PATH, HANDLER).

**88. How does dynamic routing work in Express.js?**

* Using route parameters and wildcards:

javascript

Copy code

app.get('/user/:id', (req, res) => {

res.send('User ID: ' + req.params.id);

});

Database :-

1. **What do you understand by term database ?**

**Ans :** A database is an electronically stored, systematic collection of data. It can contain any type of data, including words, numbers, images, videos, and files. You can use software called a database management system (DBMS) to store, retrieve, and edit data.

1. **Define RDBMS**

Ans : RDBMS is stand for **relational database management system** it store data in the form of rows and columns .In this table relate to each other , by this relation we can easily retive ,create ,manulate and update the data of any table.

1. **What are various type of relationships in database? define them**

Ans: There are different different types of relatiship in database and these relationship are help to establish connection between the tables.there are three types of relationship in this

* **One-to-One**
* **Many-to-Many**
* **One-to-Many**

1. **Explain normalization**

Ans : normalization is the process to organize the data in database it reduce data redundancy and improve data integrity.It involves dividing tables into small table and manage relationship between them.

1. **What are different types of normalization?**

Ans: Types of normalization is

* 1NF
* 2NF
* 3NF
* BCNF
* 4NF
* 5NF

1. **What is SQL ?**

Ans : Sql stand for **structured query lang uage** it is use to access ,manage and store the data in database.SQL database is known for relational database.In the SQL database data store in the form of rows and columns .

1. **How many SQL statement used ? Define them**

Ans : There are 5 types of command used by SQL .

* DDL (Data definition language)

It is use to change the structure of table like create,alter,dropand truncate the table .It permanently save the changes in database .

Some commands under the DDL .

1 create ,2 alter , 3drop, 4 truncate

* DML(Data manipulate language)

DML is responsible for all the changes in database .Commands under the DML is

Insert,update and delete .

* DCL(Data Control language)

Data Control Language is used to give and take back authority from any users

Some DCL

grant and revoke

* TCL(Transaction Control Language)

Transaction control language are only used in DML command like insret,update and delete only.It use commit,rollback and save point

command

* DQL(Data Query Language)

It uses to get the data from the database it only use select command.

1. **Enlist some commands of DDL , DML and DCL**

DDL : create , truncate , alter and drop

DML : insert , update , delete

DCL : grant , revoke

1. **Explain the term record, field and table in database**

Ans : A table has records (rows) and fields (columns). Fields have different types of data, such as text, numbers, dates, and hyperlinks. A record: Contains specific data, like information about a particular employee or a product.

1. **What is view in database**

Ans : A view is a virtual table in database it provide the way to look at data from one or more table in a customize manner.It does not store data it self but it show the data store in the database .

1. **What are advantage and disadvantage of view?**

Ans :

**Advantages**: Views simplify complex queries, enhance security, provide data abstraction, allow custom data presentation, ensure reusability and consistency, and simplify reporting tasks.

 **Disadvantages**: Views can introduce performance overhead, have limited updatability, depend on base tables, may become outdated, add complexity in management, and consume database resources.

1. **Define Entity , Entity Type and Entity Set**

Ans : If we talk about **Entity** is a real world object like any pen or any object . And there is multiple **types of entiy** of an object like its firstName , LName , contact no so these are the types of entity .

And **Entity set** is a collection of multiple entity ,in the entity set different types of entity are available .

1. **Define a weak entity set**

Ans : Weak entity set are set which depend on another strong entity set to define their unique identity.Weak entity have not its own primary key but weak entity make its primary key with a strong entity.

1. **What do you undestand by join ? Explain different types of Join**

Ans : Joins are used to combine the rows from one or more tables in the database .

There are four types of join.

* Inner : inner join return the matching values in both tables based on their join conditions
* Left outer : Left join return all the rows form the left table and return matching rows from the right table
* Right outer : Right joins return all the rows from the right table and return matching values from the left table.
* Full outer : Return all the rows when there are either match in right or left table

1. **What is self join.**

Ans : Self join is a databse concept in this we join the same table with different aliases name :

Select e.name,e.lastName , m.firstName as ManagerFirstName ,m.lastName as ManagerLastName from employee e inner join employee m on m.employeeId = e.managerId ;

1. **Explain Primary key, Foreign key, unique key, composite key, super key**

Ans :

Primary Key : Primary key is the set of column in the table that is uniquely identified the rows in the table.It can we use to create relatoship between tables.

Foreign Key : foreign key is the column of primary key from another table of databse.And it is used to make relationship between two tables.and it can be null

Unique Key : Unique key is set to uniquely identified in the table and it is not duplicate same as primary key

Composite Key : A primary key consist multiple columns to uniquely identified a row in a table .

Supre Key : a super key is set of two or more columns in a tables that uniquely identified the rows of table .

1. **What is projection in database?**

Ans : Projection me hum kisi table k columns ko filter karte h suppose in a table there are five fields like in, name , address , contact , marks .

Isme hum projection dete hai ki uss table me se name and address hi chaiye h

Iske lia query hoti h

Select name , address from students ;

Select name ,address from students where address =’Indere’

1. **What is difference between delete, truncate and drop command**

Ans : delete is used to delete a particular rows and column while truncate is used to delete all the items from the table and drop is used to delete a database or a table from the database .

1. **Based on given table, solve the following query ?**
2. Employee table



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**

Ans: MongoDb is NoSQL database it is document based database,means it store data in json like document .Data in mongoDb store in document with key-value pair instead of row and columns .It is flexible and schema less.

Advantage of mongoDb : It is flexible , Schema less , Read and write operation are fast due to indexing ,Easy to use due to JSON like structure it helps developers to use it easy .

**23.What is difference between SQL and NoSQL database**

Ans :

1. In SQL there is a fixed schema and static but in NoSQL there is no fixed schema and it has dynamic schema .
2. In sql data store in rows and column but in NOSQL data store in document or in key,value pair.
3. In sql Data model are predefine but in NOSQL data model are ustructured and semi structured ,it is a dynamic anf flexible .

**24.What is document in mongodb?**

Ans.In mongoDB document is the primary data structure which is used to store data in json like formate .Its structure is flexible .In mongoDB the document is the collection of key value pair and each key is unique.

**25.What is collection in mongodb**

Ans : In mongoDb , a collection is a group of mongoDb documents.Documents within the collection can have different fields .

**26.What is mongo shell**

Ans : mongo shell is an interactive js interface , that is use to directly interact with mongoDB .It is used to manage collection and manuplate.

**27.What are some features of mongodb**

Ans:

1. dynamic schema means each documents have their own fields
2. it provide indexing by this you find a document on the basis of indexing without scanning the whole document .
3. easy to modify : you can add or remove any field of document without impacting old document.
4. File storage : by using gridFs you can store large file in document and retive it .
5. **Full-Text Search:** MongoDB built-in text search capabilities provide karta hai jo text indexes aur text-based queries ko enable karta hai.

**28.How to add data in mongodb ?**

Ans : By the use of insertOne and insertMany or save() method .

**29.How do you update data in mongodb**

Ans : BY the use of updateOne , updateMany method .

**30.How do you delete document**

Ans : by the use of deleteOne , deleteMany ,findByIdAndDelete

And there is another method remove that is deprecated in newer version of mongoDB.

**31.How to perform query in mongodb**

Ans : db.collection.find(<query>, <projection>)

// Find users with name "John Doe"

db.users.find({ name: "John Doe" })

// Find users older than 30

db.users.find({ age: { $gt: 30 } })

**32.What are datatypes in mongodb**

Ans : string , null , object , objectId , array , number , decimal , Boolean , date , timestamp , binary data .

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

Ans : In mongodb , index is the data structure that improve the retrieval operation on a collection.By the use of index you can retrieve the data on the basis of index instead scanning the whole document in a collection.

**how to create index in mongodb :**

by the use of createIndex method and when you define the schema

**34.Explain the set modifier in mongodb**

Ans : The set operator is used in update operation.It is used to modify or set any field of document in collection.

**35.Does mongodb support primary key and foreign key relationship ?**

Ans : Normally mongoDb doesn’t support primary key and foreign key

MongoDB support references between documents using objectId field.This act as foreign key in mongoDb.

**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?**

Yes it is true that mongoDB use BSON to represent document structure in mongoDB.It extends JSON to incuding data types and to be more efficient for encoding and decoding data .

### 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?

Ans : \_id index is created by mongoDB for every collection.It is create automatically in every document by default.Each document is uniquely identified by \_id .

**In-which language mongodb is writtern ?**

Ans :- The mongodb is written in c++ language . It support multi-threading , provide high performance , memory management .

### 39.What will happen when you remove a document from database in MongoDB? Does MongoDB remove it from disk?

Ans : When you remove a document from mongodb it remove from the disk .

When you perform deleteOne() and deleteMany() operation, mongoDb locates the specified document and remove from the collection.

**40.What is capped and uncapped collection in mongodb**

Ans : **Capped Collection :-** Capped collection are the collection that are fixed in memory size and a maximum no. of document is also fixed in this.Capped collection are used when you know the size and memory usage by the collection.If in case size limit is reached the older document is overwritten by the newer document .

**Uncapped Collection :** Uncapped collection are the collection that is dynamic in size and no document is automatic overwritten in case of uncapped.Full crud operation is perform easily in uncapped collection .

**41.What is lookup in mongodb**

Ans :

LookUp is an aggregation pipeline stage that allows to perform left outer join between two collection.

In mongoDb lookup is used in aggregation framework to perfom a left outer join to another collection in the same database .This allow to combine document from different collection on the related field .

Db.collection.aggregate([{

$lookup: {

from: "otherCollection",

localField: "localFieldName",

foreignField: "foreignFieldName",

as: "outputArrayField"

}

}])

**42.How to get record of second max age record from mongodb collection**

Ans : To get second maximum from the collection use the aggregate framework in mongoDb collection .

const result = await collection.aggregate([ { $sort: { age: -1 } }, { $skip: 1 }, { $limit: 1 } ]).toArray();

**43.What is mongoose ?**

Ans : mongoose is a Object Data Modeling(Object Data Modeling) library for mongoDb and node.js . It provides a schema based solution to model your data application . schema-based modeling, validation, middleware hooks, query building, and other essential features.

**44.What is mongoose Schmea**

Ans : in mongoose schema is the blueprint that define the structure of document in your collection .It represents what fields will be present , data type of field , validations for data .

**45.How to set reference in mongoose Schema**

Ans : By using the ‘ref’ option with in the schema .

**46.How populate works in mongoose ?**

Ans : In mongoose populate method is used to when you want to mix the one document with the another collection document .

First when you define schema set ‘ref’ field with ‘Schema.type.ObjectId’

After then create schema with reference field

And when we want populate the data from parent table db.collection.find().populate(‘userId’) it give the whole result from the parent table

**47.List the name of the mongoose method to perform CRUD operation**

**for create** : save() , create()

**for read** : find() , findOne() , findById()

**for update** : updateOne() , updateMany() , findByIdAndUpdate() , findOneAndUpdate()

**for delete** : deleteOne() , deleteMany() , findOneAndDelete() , findByIdAndDelete() .

**48.What is default connction pool size in mongoose.**

Ans : the default pool size connection is 100 . we can increase or decrease the pool size connection .

**49.How to set connection pool size in mongoose.**

Ans : by the use of poolSize method when we connect the mongoDb .

**50.What is connection pool and what are advantage of using connection pool size.**

Ans :

**How many ways to send the data in requsest object**

**How to serve static file**

**What is global in js**

**What is router function**

**If security is not matter to send the data which method you use get or post method**

**What is the use of app.use()**

**How to send data between two parties**

**1. Which of the following is NOT an OOP (Object-Oriented Programming) concept?**  
\*

Inheritance

Encapsulation

Polymorphism

Compilation

**2. Which of the following is NOT a type of software testing?**  
\*

Unit Testing

Integration Testing

Cloud Testing

System Testing

**3. Which of the following is an example of a web server?**  
\*

Apache

Google Chrome

Microsoft Word

MySQL

**4. Which of the following data structures uses LIFO (Last In, First Out) principle?**  
\*

Queue

Array

Linked List

Stack

**5. "In Python, what is the output of the following code snippet?  
list = [1, 2, 3, 4, 5]  
print(list[2:4])"**  
\*

[1, 2]

[3, 4]

[2, 3, 4]

[3, 4, 5]

**6. "What will be the output of the following C++ code snippet?  
#include <iostream>  
using namespace std;  
class Base {  
public:  
virtual void show() { cout << ""Base class"" << endl; }  
};  
class Derived : public Base {  
public:  
void show() { cout << ""Derived class"" << endl; }  
};  
int main() {  
Base \*b;  
Derived d;  
b = &d;  
b->show();  
return 0;  
}"**  
\*

Base class

Derived class

Compilation error

Runtime error

**7. Which of the following statements about pointers in C is true?**  
\*

A) The address of a variable can be obtained using the $ operator.

B) A pointer can only point to a single variable at a time.

C) Pointers can be used to access array elements.

D) Pointer arithmetic is not supported in C.

**8. "What is the output of the following C code snippet?  
#include <stdio.h>  
int main() {  
int arr[] = {10, 20, 30, 40, 50};  
int \*ptr = arr;  
printf(""%d %d"", \*(ptr + 2), ptr[2]);  
return 0;  
}"**  
\*

10 30

30 30

20 20

30 50

**9. "What will be the output of the following C++ code snippet?  
#include <iostream>  
using namespace std;  
class Test {  
public:  
Test() { cout << ""Constructor"" << endl; }  
~Test() { cout << ""Destructor"" << endl; }  
};  
void func() {  
Test t;  
}  
int main() {  
func();  
cout << ""End of main"" << endl;  
return 0;  
}"**  
\*

Constructor\nEnd of main\nDestructor

Constructor\nDestructor\nEnd of main

Destructor\nConstructor\nEnd of main

Compilation error

**10. Which of the following is true about the delete operator in C++?**  
\*

It is used to free memory allocated using malloc().

It can be used to deallocate memory allocated for an array.

It can only be used for objects created with new.

It can deallocate memory for both single objects and arrays created with new.

**11. Hari (H), Gita (G), Irfan (I) and Swati (S) are brothers and sisters. All were born on 1st January. The age difference between any two successive siblings (that is born one after the other) is less than 3 years. Given the following facts, find out the order in which they were born (eldest first): I. Hari's age + Gita's age > Irfan's age + Swati's age. II. The age difference between Geeta and Swati is 1 year. However, Geeta is not the eldest and Swati is not the youngest. III. There are no twins**  
\*

HSIG

IGSH

IHSG

SGHI

**12. A group of children went to play a game of marbles. Lata had 6 marbles. Maria had 6 marbles and the youngest Ariana had none. So they decided to share their marble quality among themselves. In return , Ariana offered to give them her 15 PokÃ©mon cards. She gave the cards in the same proportion in which she received marbles. How many cards Maria gets from Ariana?**  
\*

4

5

1

3

**13. A, B and C can do a piece of work in 20, 30 and 60 days respectively. In how many days can A do the work if he is assisted by B and C on every third day?**  
\*

12 days

15 days

16 days

18 days

**14. An error 2% in excess is made while measuring the side of a square. The percentage of error in the calculated area of the square is:**  
\*

0.02

0.0202

0.04

0.0404

**15. A boat running upstream takes 8 hours 48 minutes to cover a certain distance, while it takes 4 hours to cover the same distance running downstream. What is the ratio between the speed of the boat and speed of the water current respectively?**  
\*

02:01

03:02

08:03

Cannot be determined

**16. How do you approach learning a new technology or programming language?**  
\*

I wait for formal training or a course.

I start by reading documentation and practicing on my own.

I ask a colleague to teach me.

I avoid it if it is not immediately required for my job.

**17. How do you handle feedback from your supervisor?**  
\*

I take it personally and feel discouraged.

I listen carefully, understand the feedback, and make necessary improvements.

I ignore the feedback if I do not agree with it.

I argue my point of view to prove I am right.

**18. How do you handle a situation where you are given a task that you do not know how to complete?**  
\*

I try to figure it out on my own, even if it takes a long time.

I ask a colleague for help immediately.

I research and attempt to find a solution, then seek help if necessary.

I inform my supervisor that I cannot complete the task

**19. Choose the most effective way to confirm receipt of an email**\*

Got It

I have received your email and will review it shortly

Seen your email

Okay, noted.

**20. Select the most professional way to request a meeting with a colleague.**  
\*

Can we meet to talk about the project?

Let's chat about the project whenever you're free.

Could we schedule a meeting to discuss the project at your earliest convenience?

Need to talk. When are you free?

1. **What is request and response in node.js.**

Ans : Request and response object are the callback function parameter used in node js and express.js.You can get the request query , headers , body , params and cookies

1. **What is package.json and package.lock**

Ans : **Package.json** is a file that contain node js application information .It contain information like project name , version , dependencies and many more.

**Package.lock** is an automatic generated file that lock the exact version of dependencies.It ensure that project environment remains consistent and regarless which developer or machine is being used.

1. **What is npm ? How to install dependency/module at application level and Environment elevel**

**Ans :** npm is a package manager for node js that let you share , install and manage dependencies

Global level npm i –g <dependency name>

Application level : npm i <dependency name>

1. **How do you manage packages in your node.js project**

Ans : using npm and yarn to install ,remove and update the package tat listed in package.json .

1. **How Node.js is better then other framework**

Ans :

1. What are the some commonly used timing features of Node.js
2. What is fork in node.js
3. How do you create a simple server in node.js that return hello world
4. How many types of API functions are there in node.js
5. What is REPL and how to use it?
6. What is purpose of module.exports.
7. What is an event-loop in Node.js 57
8. If Node.js is single threaded then how does it handle multiple request/ concurrency 58
9. Differenctiate between process.nextTick() and setImmediate() 59
10. What is node.js stream 60
11. What is middleware 61
12. Explain what a reactor pattern in node.js 62
13. Describes the exit code of Node.js 63
14. What is an EventEmitter in node.js 64
15. What is a thread pool and which library handles it in node.js 65
16. What is purpose of NODE\_ENV
17. How would you connection mongodb database to node application
18. What are different type of http request
19. What is difference between get and post
20. What is query string and how to send the data in get request
21. What is the use body parser
22. How to set the path of static file in express
23. What are types of Middleware in express ? Explain with suitable example
24. Does order of middleware matters in express.
25. What is express.js
26. What are some distinctive features of Express
27. is Express.js fron-end or backend framework?
28. Why do we use express.js
29. What is difference between express.js and node.js
30. What do you understand by Scaffolding in Express.Js
31. Which are the argument available to an Express.Js route handler function
32. How can you allow CORS in Express.jS?
33. How can you deal with Error handling in Express.js? Explain with an example
34. Write a code to start serving static file Express.JS
35. How can we render plain HTML in express
36. How can we send the data while rendering page in express
37. How to enable debugging in express app?
38. What is routing and How routing works in express
39. How dynamic routing works in express.js