**JAVASCRIPT**

IIFE

Closures (benefits/purpose)

Higher-order functions (benefits/purpose)

First-class functions

Title case words in a string

Convert nested callbacks into promises

Type coercion

Prototype chain

Call, Apply, Bind

Inheritance in JavaScript

Instanceof

Pure functions

Type casting (explicit)

JSON methods

WeakMap, WeakRef

Object.seal

Eval()

Labels in loops

Escape sequences

Falsy values

ASCII vs Unicode

Byte vs Kibibyte

Yesterday’s date (generate programmatically)

6-digit random OTP generation

Find keys common in two objects

Reduce(): Sum of positive numbers in a mixed array

Prototype

Super (concept)

DOM manipulation methods

Examples for higher-order functions (HOF)

Examples for JavaScript engines

REPL

parseInt()

Event bubbling

Proxy object

Type casting

undefined and null

typeof operator

Exceptions

Recursion

Map vs Object

Promise methods

Node.js

File response

Cluster and Worker

Transactions

app.all

Pagination

Stream and buffer

Process and threads

Static folder

res.end()

Status codes

Port numbers

Rate limiting

Content negotiation

req.body

res.setHeader

ES Modules

EventEmitter

Memoization

Real-time example of closures

Polyfills

Rules for hooks

Synthetic events vs native events

HTML sanitization

fs.stat

Immutability of state

Running a callback when a functional component unmounts

ForwardRef

Preflight request

OPTIONS method

Browser storage types and IndexedDB

Git:

Staged vs unstaged

Stash

Closures (benefits/purpose)

Higher-order functions (benefits/purpose)

First-class functions

Title case words in a string

Convert nested callbacks into promises

Type coercion

Prototype chain

Call, Apply, Bind

Inheritance in JavaScript

Instanceof

Pure functions

Type casting (explicit)

JSON methods

WeakMap, WeakRef

Object.seal

Eval()

Labels in loops

Escape sequences

Falsy values

ASCII vs Unicode

Byte vs Kibibyte

Yesterday’s date (generate programmatically)

6-digit random OTP generation

Find keys common in two objects

Reduce(): Sum of positive numbers in a mixed array

Prototype

Super (concept)

DOM manipulation methods

Examples for higher-order functions (HOF)

Examples for JavaScript engines

REPL

Default parameters

Object methods practicals

Object.freeze, Object.seal

Object.prototype

Function composition

Generator functions

WeakSet, WeakMap

Nullish coalescing operator (??)

Optional chaining (?.)

Proxy objects

Truthy and falsy values

Labels in loops

Getters and setters

Mark and sweep garbage collection

Scoping (let vs var)

Split an object into two objects containing equal keys

Escape sequences

Static keyword

Properties of error objects

Countdown timer from 10 to 0

Throw error if an object passed is empty

Error types (custom errors)

Print date in dd-mm-yyyy format

Default parameters

Object methods: freeze, seal, prototype

Function composition and currying

Nullish operator, optional chaining

Proxy object

Promises: internals, callback queue, microtask queue, promisify

Generators

WeakMap, WeakSet, WeakRef

Truthy vs falsy values

Scoping: let vs var

StructuredClone

Memory: heap vs stack

Anagrams

Countdown timer from 10 to 0

Flat arrays and frequency using reduce

Remove last property from an object

Shuffle positions in an array

Pascal to camel case, snake case to Pascal

Find uncommon elements from two arrays

Print random value

Block scope vs function scope

Closures: drawbacks and deeper understanding

Null pointer exception

JS typed arrays

Byte

Labels

Properties of the error object

Static keyword

Escape sequences

Referential equality

Regular function vs arrow function (this, shallow copy, deep copy)

Application of closures

Event delegation

Object methods: freeze, seal, prototype

Promise combinators

Proxy object

Implicit vs explicit type casting

Map vs forEach

InstanceOf

Pure functions

Call, Apply, Bind

Inheritance in JS

Error objects

Process vs thread

Streams: duplex/transform

Content negotiation (e.g., accept header)

REST principles

Shallow vs deep copy

Async/Await and Promises syntax

Design patterns

Map, WeakMap, Set, WeakSet differences

StructuredClone

Scoping (let, const, var)

Proxy object

Event emitter

Process.nextTick vs SetImmediate

**NODE.JS**

Cron jobs

Worker threads

CommonJS

Promisify

Environment variables

fs.stat

Middleware

HTTP 400

httpOnly cookies

JWT claims

Reverse proxying

Default HTTP port

Content negotiation

Socket

CSRF

Cron jobs

Fork() vs Spawn()

Streams in Node.js

Stream and buffer

Delete a file using fs.unlink

Core modules in Node.js

Process vs Thread

Process.nextTick()

App.use

Error handling in Node.js

CORS (concept)

Transform streams

Process.nextTick

setImmediate

Event-driven programming

Thread pool

Child processes: spawn, fork, exec, execFile

Piping

Express middlewares: express.urlencoded, express.json

Environment variables

app.locals, app.set, app.engine

Rate limiting

Content negotiation

OPTIONS method and preflight requests

EventEmitter

JWT components and forgery protection

CSRF protection

HTTP codes: 403, 201, 400

Refresh tokens

HttpOnly cookies

Reverse proxy

DNS and subdomains

Default HTTP/HTTPS ports

FS module: fs.stat

Threadpool

Concurrency vs parallelism

Event-driven programming

Child processes: spawn, fork, exec, execFile

Streams, transform streams, piping

process.nextTick, setImmediate

Middleware: logging parameter names, express.urlencoded, express.json, app.locals, app.engine, app.use, app.all

JWT: components, forgery protection, refresh tokens

CSRF prevention

HTTP methods: OPTIONS, preflight request

HTTP statuses: 401, 403, 500, 201, 400

Reverse proxy (Nginx), HTTPS port

Environment variables

Rate limiting

Content negotiation

Crypto module, encryption vs hashing

PM2

Cron-job

fs.stat

Node.js internal working

Fork vs spawn

Middleware types (app.use, error handling middleware)

Duplex and transform streams

OPTIONS method

Journaling

Sharding and replication

Voting in replica sets

fs.unlink

updateMany syntax

Process vs thread

Axios interceptors and cancel tokens

Bulk write operations

Creating indexes for a collection

Dynamic routing and router chaining

Error handling middleware

Query selectors

App.locals usage

Covered queries in MongoDB

Transactions and materialized views

Dynamic routing

**MongoDB**

Isolation

Bulk write

$nin, $exists

$match, $group & $sum query

BSON

Timestamp

How to rename a collection

Upsert operation

Increase 10% of the salary for employees in the Finance department

Change employee names to lowercase

Change the first character of employee names to lowercase

Partition tolerance

Capped collection

$facet, $merge

$addToSet, $pullAll

$elemMatch

TTL indexing

Levels in isolation

Database profiler

Covered query

Clustered collection

Default index

Namespace

Explain() function

Journaling

TTL Index

UpdateMany syntax

Increment/Decrement syntax ($inc, $dec)

Count products in each category query

Create an index for a collection

$all

$addToSet

$elemMatch

Rename a collection

Developers who know at least 3 skills (query)

Count employees not referred (query)

Voting in replica sets

Partition tolerance (CAP theorem)

Capped collections

TTL index

Virtual collections

Covered queries

Durability, isolation (ACID properties)

$push vs $addToSet

Scaling options

Calculate total salary of developers who know both Java and HTML

Query developers who know more than 4 skills

Decrease salary by 10% if the developer isn’t full-stack

Rename collections

WiredTiger storage engine

Partition tolerance

Capped collection

$lookup and $addFields

Covered query, indexing, indexing drawbacks

Transactions

Materialized views, virtual collections

Query routing, batch sizing

$push vs $addToSet

Scalability options

Total salary of developers who know Java and HTML

Reduce salary by 10% if not a full-stack developer

Rename collection

Indexing types: single key, multikey, compound

Covered queries

$lookup, $facet, $addToSet, $expr

Query to find second largest marks

Upsert operation

Query routing and batch sizing

$all operator

Materialized views vs virtual collections

Sharding and partition tolerance

Aggregation framework operations

**DATA STRUCTURES AND ALGORITHMS**

Complete tree

Full tree

Perfect tree

DFS complexity

BFS complexity

Binary search with target element

Find unmatched elements from two arrays

Example:

Input: arr1 = [3,5,7,9,4,6], arr2 = [5,7,2,43,8,35]

Trees and Graphs

Load factor

Double hashing

Separate chaining

Priority queue

Dijkstra's algorithm

Tarjan's algorithm

Red-black tree

AVL tree

Write a function to find uncommon elements from two different arrays

Write a function to initialize a 2D array

Sum all elements of an array using recursion

Separate chaining (collision resolution)

Balanced tree

Complete binary tree

Heap sort

Backtracking

Divide and conquer technique

Collision resolution techniques in hashing

Head in a linked list

AVL tree

Red-black tree

Sparse arrays

Priority queue

Dijkstra's algorithm

Tarjan's algorithm

Disadvantages of recursion

Time complexity of binary search

Insertion sort

Stack vs queue

Sparse tree

Complete tree

Separate chaining (collision resolution technique)

Load factor in hashing

Tail recursion

Compressed tries

Backtracking

Anagrams

Insertion sort

Stack vs queue

Sparse vs complete trees

Binary tree: number of nodes, depth calculation

AVL trees

Tail recursion: advantages

Load factor, rehashing

Linear and quadratic probing in hash tables

Length of the longest repeating and non-repeating substring

Function to check if two strings are anagrams

Remove odd element nodes from a linked list

Asymptotic notations

Minimum spanning tree

Divide and conquer technique

Collision resolution techniques in hash tables

Complete binary tree vs sparse tree

Heap sort

Merge sort, bubble sort, selection sort

Binary search tree operations: search, traversal

Linked list operations: find middle element, remove odd nodes

Depth-first search (DFS) traversal

Trie serialization and deserialization

Tail recursion advantages

**REACT**

Class components

PropTypes

Browser router vs Hash router

useCallback

useMemo

Lazy loading

Pure components & React.memo

Thunk vs Saga

Drawbacks of CSR (Client-Side Rendering)

How to pass data from child to parent

Add CSS properties (e.g., to editbutton class)

Task completion with Redux Thunk

Reverse string to their own place (custom functionality)

Performance optimization

Achieving DOM manipulation in React

Custom hooks and common components

Throttling & debouncing

useCallback

React.memo

React Fiber

Uncontrolled components

Shadow DOM

Axios:

Interceptors

CancelToken

HTML sanitization in React

Synthetic events and event pooling

Getting query params and path params (useParams, useLocation)

Types of routers: BrowserRouter, HashRouter, etc.

Link props

Rules for hooks

Immutability of state

Create a list of countries

Getting values from context

Custom hooks implementation

Error boundaries

Render props

useMemo with a dependency list

Suspense and lazy loading implementation

Core principles of Redux

Redux store methods

Disadvantages of recursion

Prototype chain

React.memo

WeakMap and WeakRef in React

UseParams and UseLocation

Axios interceptors

Shadow DOM

Controlled vs Uncontrolled components

React.memo

useCallback

Axios interceptors and cancelToken

Controlled vs uncontrolled components

React Fiber

Shadow DOM

Synthetic events vs native events

Event pooling

Polyfills

HTML sanitization

useParams, useLocation

Immutability of state

Purpose of spread operator in updating state

Redirecting users

URL fragments

MemoryRouter

Downsides of localStorage

IndexedDB

Debouncing

useEffect vs useLayoutEffect

Lifecycle methods

State and immutability

React.memo and referential equality

Context and passing props from child to parent

Higher-order components

Dynamic routing, useParams, useLocation

Axios: interceptors and cancel tokens

Error boundaries

Lazy loading, custom hooks

useCallback and useRef

useEffect dependency array

React Fiber

Synthetic vs native events, event pooling

HTML sanitization

Shadow DOM

Redux, Redux Thunk

LocalStorage downsides, IndexedDB

Suspense

Counter app (with sibling components)

Disabling right-click events globally in useEffect

Error boundaries

React Profiler

ForwardRef and Proptypes

Lazy loading and Suspense

React.memo and referential equality

Pure components

Render props

Fragments

Flux vs Redux architecture

Custom hooks implementation

React Router concepts

Event pooling and delegation

useMemo and dependency lists

Redux thunk

Disabling right-click events globally

Child-to-parent communication

Error handling in React apps