

Important AFS Questions:

NODE & Vanilla JavaScript:

- 1) Write JS code to remove duplicate values from an array?
- 2) Define the underlying Architecture of Node?
- 3) What is middleware? Write a function to explain middleware?
- 4) What is Event Loop?
- 5) What is spread and rest operator?
- 6) How to Fetch and print out all keys from an object?
- 7) What is Node Package Manager?
- 8) What is the importance of package.json file?
- 9) How we can create package.json through terminal?
- 10) Why Global installation of Modules requires?
- 11) Ques: What is the output of below code:

```
var a = function(){  
    console.log(a);  
    var a = 10;
```

```
}  
console.log(a());  
var b = function(){  
    console.log(a);  
    let a = 10;  
}  
console.log(b());
```

12) What is the difference between var, let and const?

13) What is callback function and higher order function?

14) what is arrow function? Can you write the syntax of arrow function

15) In any JS file I have two variables a and b. I want to use both these variable in any other js file, how can I do this?

16) What is deep copy and shallow copy?

17) What is prototype?

18) What is worker in node.js?

19) What is hoisting?

20) What is closure?

21) Write code to connect node.js with MySQL database?

22) What is Http status code?

23) Write code to count Duplicate elements from an array

```
let arr = ["one","two","three","one","two","one"];
let count = {};
arr.forEach((element)=>{
    //The expression count[element] || 0 returns the value of count[element] if it is
    set, otherwise 0.
    //Then just add one and set it again in the object.
    count[element] = (count[element] || 0) + 1;
})
```

24) What is 'in' and 'of' keyword in JS?

25) Explain below array functions

- a) join()
- b) split()
- c) slice()
- d) splice()
- e) indexOf()

26) How the event loop works internally?

Event loop continuously runs in the following sequence

Timers -> Pending callbacks -> Idle, Prepare -> Poll -> Check -> close callbacks

After completing one cycle, it starts again if there are more tasks.

Node.js uses **libuv**, a C++ library that provides the **event loop** and manages asynchronous operations.

The event loop consists of **six phases** that execute in a fixed order, handling various types of callbacks.

1. **Timer Phase:** It checks if any timer has expired and executes its callback.
2. **Pending Callbacks phase:** This phase executes I/O callbacks that were deferred (like errors or TCP errors)
3. **Idle, Prepare Phase:** This phase is used only internally by libuv.
4. **Poll Phase:** This is the heart of Event Loop. This phase can pause if no timers are pending. It handles:
 - a. Incoming I/O events (like reading files or network requests)
 - b. Blocking I/O operations (if none are pending, it waits here)
 - c. Executes I/O callbacks immediately if available.
5. **Check Phase:** This phase executes `setImmediate()` callbacks. Callbacks from `setImmediate()` always executes after the poll phase
6. **Close callbacks phase:** This phase executes
 - a. Callbacks from `process.exit()`
 - b. `Socket.on('close')`
 - c. `Stream.on('close')`

27) What is the output of below code:-

```
setTimeout(() => console.log('Timeout'), 0);  
  
setImmediate(() => console.log('Immediate'));  
  
process.nextTick(() => console.log('Next Tick'));  
  
Promise.resolve().then(() => console.log('Promise'));  
  
console.log('Sync');
```

Output:

Sync
Next Tick
Promise
Timeout
Immediate

NOTE: Microtasks always execute **before** the next phase.

Summary

Task	Type	Phase
setTimeout()	Macrotask	Timers
setImmediate()	Macrotask	Check
process.nextTick()	Microtask	Before next phase
Promise.then()	Microtask	Before next phase
fs.readFile() callback	Macrotask	Poll

28) Difference between put and patch Http request method?

29) Difference between post and GET http request method?

30) What is SQL injection?

31) Why should we use ORM like sequelize?

32) how many types of modules or packages are there in node js?

- Core Modules (built-in)
- Local Modules (user-created)
- Third Party modules (installed via NPM)

33) What is Debouncing?

34) What is throttling?

35) What is MVC? Draw its architecture diagram?

36) What are the different ways to create an object in JavaScript?

37) explain event bubbling and event capturing.

38) What are Streams in Node.js?

39) How would you handle error handling in Node.js applications?

40) Explain the concept of a "callback hell" in Node.js and how can it be avoided?

41) Explain the concept of CORS (Cross-Origin Resource Sharing) and how would you handle it in a Node.js app?

42) What are the differences between SQL and NoSQL databases? When would you use each in a Node.js application?

- 43) Implement a function to read and write files asynchronously using fs module.
- 44) What are some best practices for securing a Node.js application
- 45) Write a simple HTTP server using Node.js.
- 46) Explain the use of app.use() in Express.js for middleware and static files
- 47) How would you build a RESTful API using Express.js? Explain routing, middleware usage, and error handling
- 48) How would you set up session management and authentication in an Express.js application?
- 49) What is dynamic routing in Express.js, and how does it work?
- 50) What are the unique features of Express.js that make it suitable for scalable web applications?

MYSQL:

- 51) What is difference between char and varchar?
- 52) What is difference between Inner Join and Outer Join?
- 53) What is difference between where and having clauses?
- 54) What are indexes and why it is important?
- 55) Which table engine supports foreign key?
- 56) What is AUTO_INCREMENT?
- 57) Explain difference between Innodb and Myisam?
- 58) What is normalization?
- 59) What is ACID properties?
- 60) What is stored procedure?
- 61) What is cascade?
- 62) What is Full Text Index?
- 63) Explain difference between DELETE and TRUNCATE?
- 64) What are foreign key constraints, and how do they affect the design of a MySQL database?
- 65) What are views in MySQL? How are they different from tables?

66) How do you use stored functions in MySQL, and what are their benefits?

67) How do you find the second-highest salary from an employee table?

```
SELECT MAX(salary) AS SecondHighestSalary  
FROM Employee  
WHERE salary < (SELECT MAX(salary) FROM Employee);
```

68) How do you count the number of records in a table?

69) What is the command to create view?

70) How to remove duplicate values of a column from result set?

71) What is drop command?

72) What is the maximum length of varchar?

73) Which function returns today's date and current time?

74) Which data-type can store image in mysql?

75) What is the SQL to check list of tables?

76) What is the SQL to execute a stored procedure?

77) What is the significance of HAVING clause in MySQL?

78) What is the SQL to fetch details of all users who registered in year 2024?

79) What type of join returns all records from the left table and matched records from the right table in MySQL?

80) Write a query to find the maximum, minimum, and average salary of employees.

81) Write a SQL query to find the total number of orders placed by each customer in an Orders table.

82) What is a Composite Key in SQL?

83) Explain the difference between BETWEEN and IN in SQL.

84) What are NULL values in SQL, and how are they handled?

85) What are aggregate functions in SQL? Can you name some?

86) Difference between COUNT(*) and COUNT(Column_nme)?

GIT:

- 87) Difference between git clone and git pull?
- 88) What is git fetch?
- 89) Explain what is the conflict in git?
- 90) what is git add?
- 91) What is git commit?
- 92) What is git push?
- 93) What is branch in git?
- 94) What is git merge?
- 95) How will you create a git repository?
- 96) What is git stash?
- 97) How to check log of commits?
- 98) Difference between GIT and GITHUB?
- 99) How to check list of branch ?
- 100) What is git revert?
- 101) Which command defines the author email to be used for all commits by the current user.
 - git config --global user.email
- 102) What command creates an empty Git repository?
 - git init

TypeScript:

- 103) What is typescript?
- 104) What is any in typescript?
- 105) How do we define the type of Array?
- 106) what is tsconfig.json?
- 107) What are the advantages of using TypeScript over JavaScript?
- 108) What are singleton class?

- 109) What are the basic types in TypeScript?
- 110) What is interface in type script?
- 111) How do you define a class in TypeScript?

React

- 112) What is JSX in React?
- 113) What are the different types of components in React?
- 114) What are props in React?
- 115) What are state and its uses in React?
- 116) What is useState hook?
- 117) What is the virtual DOM, and how does React use it to render the DOM?
- 118) What is useEffect hook?
- 119) Explain component life cycle in React
- 120) What is SSR in react?
- 121) Explain Controlled Vs Uncontrolled Components
- 122) What is Higher Order Component?
- 123) What are keys in React?
- 124) What is error boundary?
- 125) How to pass data between react components?
- 126) What is React Router?
- 127) What is Lazy loading?
- 128) What is useMemo()?
- 129) Explain Strict Mode in React?
- 130) Is HTML used in React?
- 131) Difference between Real DOM and Virtual DOM?
- 132) What is Redux?

- 133) What are stateless components?
- 134) What is web pack?
- 135) What is Babel?
- 136) Difference between Props and state?
- 137) What is Axios?
- 138) Can browser understand JSX?
- 139) How can we pass numbers to react component?
- 140) Create a simple counter using React?
- 141) How to display a list in React?
- 142) Explain the concept of "lifting state up" in React.

- 143) Where we should use MySQL and where we should use MongoDB?
- 144) What is CI/CD?
- 145) What is Jenkins?
- 146) What is Agile and Scrum?
- 147) What is sprint in Agile?
- 148) What is cluster in Node.js?
- 149) Explain promise.all vs promise.allsettled
- 150) Explain securities in web application.
- 151) Explain bind(), apply() & call()