

# ENTRY LEVEL MERN QUESTIONS

Kaylin khalal



# BASIC JS QUESTIONS

- **Synchronous JS vs asynchronous**

promises, async wait, setTimeout, api/network calls

- **Event loop**

callstack, single thread vs multi thread concept, callback queue, data structures, web apis

- **Hoisting**

var keyword, ES6, var vs const vs let, scope, initialization, functions hoisting, global vs block scope

- **Closures**

Encapsulation in JS, nested functions, scope of nested functions

- **Functions**

arrow functions, function declaration, and executions, higher-order functions (map, filter reduce), callback functions, IIFE, anonymous functions, for in vs for of, for each vs map

- **Array and Objects**

data structures, Array and objects count, deletion, push, pop, looping, replacing values

- **Others**

debouncing, throttling, basic differences between javascript and typescript, Classes, constructor, this keyword, bind and apply method in Javascript

# BASIC REACT QUESTIONS

- **About react**

what is react? who built? why popular?  
alternatives?

- **React fundamentals**

JSX, Css, State, refs and props, basic hooks  
like useEffect and useState

- **Data flow**

sending data from child component to parent  
component and vice versa, props drilling

- **Managing global state**

Context API, Redux, reducers, state vs global  
state, actions, payload, initial state,  
dispatching an action, useDispatch,  
useSelector, use of Redux middlewares like  
persist and logger

- **React ecosystem**

Popular libraries. next.js vs gatsby vs create-  
react app difference? routing libraries,  
webpack, design libraries like antd or  
material, chart libraries like chart.js, or D3.

- **Others**

DOM vs virtual DOM, performance issues scenarios, useCallback vs useMemo hooks,  
useRefs, lazy loading, code splitting, infinite scrolling, pagination, Class vs functional  
components, lifecycle methods in react class based components and use of methods  
like componentDidMount, componentDidUpdate, and shouldComponentUpdate and  
how useEffect in functional component handles these effects.

- **Practice Some Live Tasks**

Build a simple counter using useState

Build a simple counter using redux

Build a todo app using react and state and perform CRUD operation using only react

Given a list of array, display it in a list, and delete the item from that list that gets clicked

# BASIC NODE.JS QUESTIONS

- **Server basics**

http, express, js vs nodejs, is node js single-threaded? Where can nodejs be used? examples of asynchronous tasks, REPL, backend validation, error handling, module.exports use, Authentications and encryptions. Folder structure, routing, middlewares, MVC achitecture

- **Asynchronicity**

is node.js asynchronous? What are callbacks? async-await? promises?

- **APIs**

HTTP methods: GET, PUT, DELETE, POST, and PATCH, request, response, params, body ,query string, form data, statusCodes

- **noSql databases**

what is collection? schema? what is noSql format? mongoose? basic mongoose queries to edit, create, find, delete, populate, aggregation, query operators, \$set vs \$push

- **SQL databases**

examples of popular SQL databases. What is an ER diagram, primary key, foreign key? What are joins? queries for selection, update, filter, order, deletion(drop)?

- **Other**

Inner working of nodejs, file handling, streams and buffers, real-time communication (web sockets, socket io, emit, connect, disconnect, sockets, etc), dotenv, Top companies using nodejs, security concerns(sql injection, dDos attack etc.)

