1. Why you choose iT job?

I've always been interested in computer science and coding, even though I studied mechanical engineering in college. I took a few computer science courses in college, and I also taught myself to code in my free time. I've always enjoyed the problem-solving and creative aspects of coding, and I'm excited to use my skills to build user-facing interfaces.

Why do you want to change jobs?

I'm currently working as a frontend developer at TCS, and I'm grateful for the opportunities that I've had there. However, I'm feeling ready for a new challenge. I'm eager to take on more responsibility and to work on more complex projects. I'm also interested in working in a more fast-paced and innovative environment which I can get in Mid-mnc and startups.

How MBA can help frontend developer?

* Develop a better understanding of the business side of things. This would help me to develop and design user interfaces that are aligned with the business goals of the company.
* Improve my communication and collaboration skills. This would help me to work more effectively with other members of the development team, as well as with stakeholders outside of the development team.
* Learn about new technologies and trends in the industry. This would help me to build and design user interfaces that are both user-friendly and technically advanced.
* Be more likely to be considered for promotions or management positions.
* How aws can help frontend developer?

Scalability

Cost effective

Security

Reliable

Global reach

1. What is React?

React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta and a community of individual developers and companies. React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js.

React is declarative, efficient, and flexible. Components are reusable React also uses a virtual DOM to efficiently update the UI when data changes.

What is JSX?

Jsx is syntax extension to Javascript. It is used with react to describe how user interface should look like .By using jsx, we can write HTML in same file in which javascript is written.

Features of React ?

1)JSX

2)compoenent based architecture

3)one way data binding – parent to child- helps to debug easily

4)virtual DOM

5)high performance

Can browser read jsx directly?

No, browser can read only regular JS so we use babel for this.

1. What are the benefits of using React?

Some of the benefits of using React include:

* Declarative: React code is declarative, which means that it describes what you want the UI to look like, rather than how to get there. This makes React code easier to read and understand. This is in contrast to imperative programming, where you tell the library how to get to the final result,
* Efficient: React uses a Virtual DOM, which means that it only updates the parts of the UI that have changed. This makes React applications more efficient than other frameworks like angular that update the entire UI every time there is a change.
* Flexible: React is a very flexible framework. It can be used to build a wide variety of UIs, from simple to complex.
* **Easy to read and understand:** Declarative code is easier to read and understand because it describes the desired result, rather than the steps involved in achieving it.
* Maintainable: Declarative code is more maintainable because it is easier to update and refactor. This is because the code is not tied to the implementation details of the UI.
* Less error-prone: Declarative code is less error-prone because it is easier to write correctly. This is because the code does not need to worry about the implementation details of the UI.
* One way data binding –
* Reusable component
* View oriented
* Easy to build new application
* Dedicated tool for debugging

Difference between ES5 vs ES6?

ES5 | ES6

Regular | Arrow function

Use regular function | introduce Class component

Var | let const and var

Require var React = require(‘root’) | import root from root

What is event in react?

Event is action trigger by user or system like button click,alert()

Using jsx ,function can be use as event handler rather than string

Synthetic event?

* Synthetic events combine the response of different browser's native events into one API, ensuring that the events are consistent across different browsers.

Why keys are important in list ?

* A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists
* It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

React js vs React Native –

React utilizes HTML, CSS, and JavaScript to create interactive user interfaces, whereas React Native utilizes APIs and native UI components to build mobile applications. React. js used a virtual DOM to render browser code in React, whereas React Native uses Native API to render components for mobile applications.

React vs angular

React | angular | vue

Virtual dom real dom

Facebook google

Language jsx typescript javascript

Type library framework framework

Data binding unidirectional bidirectional bidirectional

1. What are the drawbacks of using React?

Some of the drawbacks of using React include:

* Learning curve: React has a bit of a learning curve. It is not as easy to learn as some other frameworks.
* Performance: React applications can be slow if they are not optimized properly.
* State management: React does not have a built-in state management solution. This means that you need to use a third-party library or implement your own state management solution.
* Large ecosystem: React has a large ecosystem of third-party libraries and tools. This can be overwhelming for beginners, and it can be difficult to know which libraries and tools to use.
* High pace of development: React is constantly evolving, and new features and changes are being released all the time. This can make it difficult for developers to keep up, and it can also lead to breaking changes in existing applications.

1. What are the different components in React?

A React component is a reusable piece of code that renders a UI element. Components can be nested to create complex UIs.

1. What is the difference between state and props in React?

State is the internal data of a component. It is used to store data that changes over time.

Props are the external data of a component. They are passed to a component from its parent component.

1. What are the different lifecycle methods in React?

React components have a lifecycle. This means that they have a set of methods that are called at different stages in their lifecycle. The most important lifecycle methods are:

* constructor(): This method is called when a component is first created.
* render(): This method is called to render the component's UI.
* componentDidMount(): This method is called after the component has been rendered for the first time.
* componentDidUpdate(): This method is called after the component has been updated.
* componentWillUnmount(): This method is called before the component is removed from the DOM.

Hooks and lifecycle methods comparison –

1. Mounting phase: where the component is created and added to the DOM.
2. Updating phase: where the component is updated with new data or props.
3. Unmounting phase: where the component is removed from the DOM.

React Hooks are used to manage state and logic within functional components, whereas Lifecycle Methods are associated with class component lifecycle phases

componentDidMount would be called after the component was mounted, whereas in a functional component, the equivalent functionality would be achieved by using useEffect.

1. What are hooks in React?

Hooks are a new feature in React that allow you to use state and other React features without writing a class component. Hooks are a great way to make your React code more concise and reusable.

Link - <https://www.freecodecamp.org/news/react-hooks-cheatsheet/>

1. Usestatehook- to update state of variable
2. Useeffect hook – 1)to perform sideeffect i.e action outside world– 1) to fetch api 2) to manipulate dom
3. useRef -

1)to directly manipulate Dom

2) to store mutable values that does not rerender

3) tracking state changes

1. usereducer
2. useContext
3. use

What is DOM?

DOM stands for ‘Document Object Model’. In simple terms, it is a structured representation of the HTML elements that are present in a webpage or web app. DOM represents the entire UI of your application. The DOM is represented as a tree data structure. It contains a node for each UI element present in the web document. It is very useful as it allows web developers to modify content through JavaScript, also it being in structured format helps a lot as we can choose specific targets and all the code becomes much easier to work with.

1. What is the Virtual DOM?

Link - <https://www.geeksforgeeks.org/reactjs-virtual-dom/>

The Virtual DOM is a representation of the real DOM that is stored in memory. React uses the Virtual DOM to compare the current state of the UI with the desired state of the UI. If there is a difference, React will only update the parts of the real DOM that need to be updated. This makes React applications more efficient than other frameworks that update the entire DOM every time there is a change.

React uses 2 virtual dom – preupdated dom and updated dom

This process of comparing the current Virtual DOM tree with the previous one is known as[**‘diffing’**](https://www.geeksforgeeks.org/explain-dom-diffing/). Once React finds out what exactly has changed then it updates those objects only, on real DOM. This entire process of transforming changes to the real DOM is called [**Reconciliation**](https://www.geeksforgeeks.org/reactjs-reconciliation/)**.**

1. How does React compare to other JavaScript frameworks?
2. What are some best practices for writing React code?

Global execution context –

The global execution context is the environment in which all JavaScript code executes. It contains the global variables and functions, as well as the object prototype chain. The global execution context is created when the JavaScript engine starts up, and it exists until the JavaScript engine shuts down.

Event loop

JavaScript is a single-threaded asynchronous programming language

Single threaded – single thing at a time

Inheritance—

Inheritance in JavaScript is a mechanism that allows a new class to inherit the properties and methods of an existing class. This is a powerful feature that can be used to reduce code duplication and improve code readability.

Use like - prototype

Prototype—to use inheritance in JS

Prototypes are the mechanism by which JavaScript objects inherit features from one another. Every object in JavaScript has a built-in property, which is called its **prototype**. The prototype is itself an object, so the prototype will have its own prototype, making what's called a **prototype chain**. **Prototype property enables other objects to inherit all the properties and methods of function constructor. When a certain method(or property) is called, it first checks inside the object but when it doesn’t find, then search moves on Object’s prototype.**

**Example – make component and import it In app.js**

Hoisting—

Link - https://www.freecodecamp.org/news/javascript-let-and-const-hoisting

Hoisting is a JavaScript feature that moves all variable and function declarations to the top of their scope, even if they are declared later in the code. This means that you can refer to a variable or function before it is declared, but the value will be undefined.

Hoisting is for var only.

Let and const can not access before initialization.

Console.log(a)

Var a = 10

Promises –

promise is an object that represents the eventual completion (or failure) of an asynchronous operation. It is a more modern way of handling asynchronous code in JavaScript than using callbacks.

Async await—

Async and await are keywords in JavaScript that allow you to write asynchronous code in a more concise and readable way.Asynchronous code is code that runs without blocking the main thread of execution. This is important for web applications, because it allows the UI to remain responsive while the application is performing tasks such as fetching data from a server.

Asynchronous—

Asynchronous code is code that performs operations that do not block the main thread of execution

A higher-order function (HOF)

is a function that does at least one of the following:

* Takes one or more functions as arguments.
* Returns a function as its result.

All other functions are first-order functions.

A callback function

is a function that is passed as an argument to another function

Callback function is part of HOF only

Memoization

Memorization in JavaScript is an optimization technique that stores the results of function calls in a temporary data structure and then retrieves the results whenever the stored result is needed in the program. This can be used to speed up the execution of a program by avoiding the need to recompute the same results multiple times.

Usememo(), react.memo(), usecallback()

Drawbacks - If the function being memoized is not called very often, the overhead of memoization may outweigh the benefits.

A white background with black text

Description automatically generated

Event bubbling--

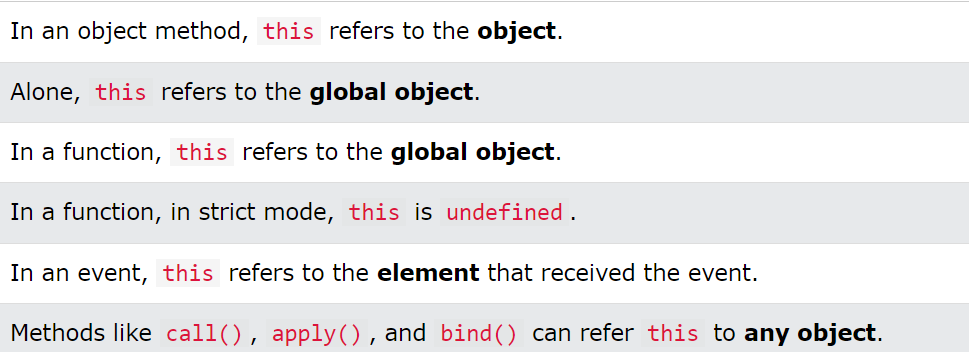
Event bubbling is a mechanism in JavaScript that allows an event to propagate from the element where it occurred to its parent elements, and so on, up to the document object. This means that when an event occurs on an element, all of its parent elements will also receive the event.

Return

The return statement is a useful tool for controlling the flow of execution of a JavaScript function. It can be used to stop the execution of a function early, to return a value from a function, and to avoid errors. The return statement in JavaScript is used to return a value from a function. The function will stop executing when the return statement is called and the value will be returned to the caller of the function

This –

In JavaScript, the this keyword refers to an **object**.

**Which** object depends on how this is being invoked (used or called).

React Routing –

React Router is a [JavaScript framework](https://www.simplilearn.com/javascript-frameworks-what-are-they-how-do-they-work-article) that lets us handle client and server-side routing in React applications. It enables the creation of single-page web or mobile apps that allow navigating without refreshing the page. It also allows us to use browser history features while preserving the right application view.

Hooks –

Usestate-

useState is React Hook that allows you to add state to a [functional component](https://blog.logrocket.com/fundamentals-functional-programming-react/). It returns an array with two values: the current state and a function to update it. The Hook takes an initial state value as an argument and returns an updated state value whenever the setter function is called

useeffect hook –

The useEffect hook is a React hook that allows you to perform side effects in your functional components. Side effects are any actions that have an impact on the outside world, such as fetching data, updating the DOM, or setting up timers.

Useeffect( function(),[dependency])

Usecontext hook –

The useContext hook is a React hook that allows function components to access the context value for a context object.The useContext hook is useful for sharing data between components without having to pass props down manually through each nested component

Props –

In React, props are short for "properties". They are the data that is passed from a parent component to a child component. Props are read-only, which means that the child component cannot change them. Props are used to pass data between components in React

Props drilling—

Prop drilling is the practice of passing data down through multiple levels of a component hierarchy. This can be done by passing the data as a prop to each child component, until it reaches the component that needs it.Prop drilling can be a necessary and effective way to manage application state, but it can also become a problem when the hierarchy of components becomes too deep or complex.

Problems – Increased code complexity: Performance problems Difficult to test

Solution –

Use the React Context API:

Use composition

Use state management libraries,redux

Closure-

Link - https://www.tutorialsteacher.com/javascript/closure-in-javascript

A [closure](https://www.geeksforgeeks.org/closure-in-javascript/) is simply a function inside another function that has access to the outer function variable. The inner function (closure) can access the variable defined in its scope (variables defined between its curly brackets), in the scope of its parent function, and in the global variables.

Now the question is why do we need to learn closures? What’s the use of it? Closures are used when you want to extend behavior such as passing variables, methods, or arrays from an outer function to an inner function.

A computer code with black text

Description automatically generated

Redux vs usecontext

The React Context API and Redux are both state management libraries that can be used to share data between components in React. The React Context API is a lightweight solution that is ideal for sharing small amounts of data between components. It is easy to use and does not require any third-party libraries. Redux is a more robust state management library that is ideal for sharing large amounts of data or for complex applications. It provides a more structured way to manage state and can be used to implement complex features, such as undo/redo or logging.

What is Redux –

Redux is an open-source JavaScript library for managing and centralizing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Redux follows the unidirectional data flow pattern, which means that data can only flow in one direction through the application: from the store to the components. This helps to prevent bugs and makes it easier to reason about the state of the application.

Benefits -

**Predictable state--** This makes it easier to reason about the state of your application and to debug problems.

**Centralized state --**Redux centralizes the state of your application in a single store.

**Testable—**

**Improvement in react-**

* Reducing the number of unnecessary renders.
* Improving the performance of third-party libraries that are used with React.
* Optimizing React for different devices and browsers.

Next js framework --

### Building Blocks of a Web Application

There are a few things you need to consider when building modern applications. Such as:

* **User Interface** - how users will consume and interact with your application.
* **Routing** - how users navigate between different parts of your application.
* **Data Fetching** - where your data lives and how to get it.
* **Rendering** - when and where you render static or dynamic content.
* **Integrations** - what third-party services you use (CMS, auth, payments, etc) and how you connect to them.
* **Infrastructure** - where you deploy, store, and run your application code (Serverless, CDN, Edge, etc).
* **Performance** - how to optimize your application for end-users.
* **Scalability** - how your application adapts as your team, data, and traffic grow.
* **Developer Experience** - your team’s experience building and maintaining your application.

For each part of your application, you will need to decide whether you will build a solution yourself or use other tools such as libraries and frameworks.

### What is React?

[React](https://react.dev/learn) is a JavaScript **library** for building interactive **user interfaces**.

By user interfaces, we mean the elements that users see and interact with on-screen.

A screenshot of a computer

Description automatically generated

By library, we mean React provides helpful functions to build UI, but leaves it up to the developer where to use those functions in their application.

Part of React’s success is that it is relatively unopinionated about the other aspects of building applications. This has resulted in a flourishing ecosystem of third-party tools and solutions.

It also means, however, that building a complete React application from the ground up requires some effort. Developers need to spend time configuring tools and reinventing solutions for common application requirements.

### What is Next.js?

Next.js is a React **framework** that gives you building blocks to create web applications.

By framework, we mean Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application.

.

A screenshot of a computer screen

Description automatically generated

You can use React to build your UI, then incrementally adopt Next.js features to solve common application requirements such as routing, data fetching, integrations - all while improving the developer and end-user experience.

Whether you’re an individual developer or part of a larger team, you can leverage React and Next.js to build fully interactive, highly dynamic, and performant web applications.

Uncontrolled and controlled component

| **Features** | **Controlled Component** | **Uncontrolled Component** |
| --- | --- | --- |
| Value Management | Managed by React state | Managed by component's own internal state |
| User Interaction | Parent component updates state on user interaction | Component updates own internal state on user interaction |
| Data Flow | Data flows from parent component to component | Data flows within the component |
| Debugging | Easier to debug | More difficult to debug |
| Performance | Generally faster as there's less state management | Generally slower as there's more state management |
| Code Complexity | Less complex code | More complex code |
| Best Practices | Considered a best practice | Considered an alternate approach |

Optimization of react –

Optimization of React is the process of making a React application run faster and more efficiently. There are many ways to optimize a React application, some of which are more effective than others.

Here are some of the most common ways to optimize React applications:

* Use lazy loading: Lazy loading is a technique that defers the loading of components until they are needed. This can improve the performance of your application by reducing the amount of JavaScript that needs to be loaded initially.
* Use memoization: Memoization is a technique that caches the results of expensive calculations. This can improve the performance of your application by preventing the same calculation from being performed multiple times.
* Use code splitting: Code splitting is a technique that divides your application code into smaller chunks that can be loaded independently. This can improve the performance of your application by reducing the amount of JavaScript that needs to be loaded initially.
* Use functional components: Functional components are a newer type of React component that are more efficient than class components. This is because functional components do not have a state or lifecycle, which can improve the performance of your application.
* Use performance tools: There are a number of performance tools available that can help you identify and fix performance problems in your React application. These tools can help you to optimize your application for different devices and browsers.

Here are some additional tips for optimizing React applications:

* Use a profiler: A profiler is a tool that can help you to identify performance bottlenecks in your application. This can help you to focus your optimization efforts on the areas that will have the biggest impact.
* Use a caching mechanism: A caching mechanism can help you to store frequently used data in memory. This can improve the performance of your application by reducing the number of times that data needs to be fetched from the server.
* Use a content delivery network (CDN): A CDN is a network of servers that can be used to deliver static content, such as images and JavaScript files, to users. This can improve the performance of your application by reducing the distance that content needs to travel to reach users.

usememeo usecallback

Error boundary

Challenges while working

If not have any solution

Any suggestions

React Fiber

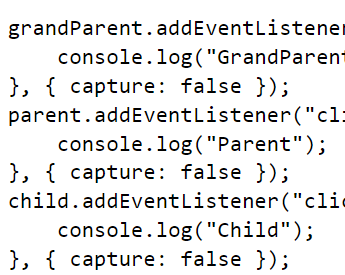
-----------------------\*\*\*\*\*\*\*\*\*\*\*

---------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------------------------------------------

Javascript interview question

Event bubbling and event capturing –

* Event capturing means propagation of event is done from parent elements to child element in the DOM while event bubbling means propagation is done from child element to parent elements in the DOM.
* The event capturing occurs followed by event bubbling.
* If {capture: true} ,event capturing will occur else event bubbling will occur.
* Both can be prevented by using the **stopPropagation()** method.

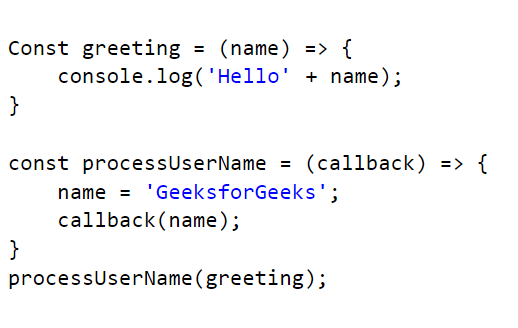


What is scope –

* **Local Scope** allows access to everything within the boundaries (inside the box)
* **Global Scope** is everything outside the boundaries (outside the box). A global scope can not access a variable defined in the local scope because it is enclosed from the outer world, except if you return it.
* **Block Scope**is everything inside the boundaries but it works only for let and const keywords. It does not work with the var keyword.

Immediately invoked Function element

* As the name suggests[IIFE](https://www.geeksforgeeks.org/immediately-invoked-function-expressions-iife-in-javascript/)is a function in Javascript which immediately invoked and executed as soon as it is defined. Variables declared within the IIFE cannot be accessed by the outside world and this way you can avoid the global scope from getting polluted. So the primary reason to use IIFE is to immediately execute the code and obtain data privacy.
* Just like callback function.
* A computer screen shot of a code

  Description automatically generated
* Callback function –
* In javascript, a [callback](https://www.geeksforgeeks.org/javascript-callbacks/) is simply a function that is passed to another function as a parameter and is invoked or executed inside the other function. Here a function needs to wait for another function to execute or return a value and this makes the chain of the functionalities (when X is completed, then Y is executed, and it goes on.). This is the reason callbackis generally used in the asynchronous operation of javascript to provide the synchronous capability.
* 

Closures –

To maintain privacy

Consider a counter function

If we use global variables to access in different function , then it can be change by any function

If we use local function it will not update as required

So we use function inside function as return .

Let var const ==

All three are hoisted but var initialize with undefined value and let and const are initialize with any value ,they are in temporal dead zone.

---------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------------------------------------------

HR interview question

Why do want to work for us?

I am excited about the potential experts in the field and contribute to the innovative work this company is doing,I have always been passionate about this industry and when I heard about this opportunity, I knew it was the perfect fit for my skills and interest.

What makes you uniquely qualified for this position ?

I have a combination of experience and qualities that makes me a perfect fit for this role.I am highly organized and have excellent communication skills,which I believe are essential for this role.Additionally my previous experience in a similar role has allowed me to develop a strong understanding of the industry and how to effectively navigate it.

What whould you say is your greatest strength?

My greatest strength is my ability to think analytically . I am able to identify potential issues and come up with efficient solutions quickly.I am also great at organizing and planning which allows me to stay on top of tasks and meet tight deadlines.