1. **Is JSX Mandatory for React?**

ANS: Jsx is not mandatory we can create any application only by using React.createElement

But it leads lot’s of effort by developer because creating complex nested component using

createElement it bit tough. JSX is developer friendly way to create components.

1. **Is ES6 Mandatory for React?**

ANS: ES6 is easier way to write code in Javascript it provides lot’s built in helpers like Array higher order functions , arrow functions etc we can use these to code out complex functionalities with short line of code.

1. **How can I write comments in JSX?**

ANS: Any Javascript code in JSX must be enclosed {} so writing comments also similar to that i.e {/\* code \*/}

1. **What is <React.Fragment></React.Fragment/> vs <></>?**

ANS: In React the component must have only one parent so return statement should include only one Parent element it enforces that all the nested elements must be enclosed inside the div but in some scenarious you don’t want unnecessary div to resolve this issue React offers the Frgament element to wrap nested elements inside that during the rendering phase it makes the React to compile without any errors but at the time of displaying actual dom tree in browser it will be empty only nested elements will appear. So it is better way to avoid unnecessary div’s

1. **What is Virtual Dom?**

ANS: Virtual Dom is virtual tree which resembles the HTML dom tree using this we can make perform manipulations to this tree rather then the original tree and we can compare two instance of the virtual dom based on that only the updated section is flushed into actual dom tree it also has all properties attached as same as original dom object.

1. **What is Reconcilation in React?**

ANS: Reconciliation is the process where the React recursively check each nodes of the current dom tree for the updated node with the previous instance of the virtual dom tree and compares it and decides what to change/add in the original dome tree it happens on every state change,props change, Lifecycle methods , Initial rendering.

1. **What is React Fiber?**

ANS: From React version 16 React team changed the Reconcilation algorithm to Fiber because recursive reconciliation algorithm was synchronous and it is causing higher frame rate (Frame Rate: the rate of no of frames displayed per millisecond) other then this multiple asynchronous side effects takes more time it leads to cause user experience . To resolve this React team divided the rendering into asynchronous and commit phase which is synchronous and they converted react elements into react fiber nodes which indicates the piece of work it is also tree structure with all properties attached it don’t have children instead it has child and other child nodes are treated as sibling nodes of that child node it easy to traverse using parent-first and depth-first-traversal the main goal of creating react fiber is .

* Assign priority to different types of work
* Pause work and come back to it later
* Abort work if it’s no longer needed
* Reuse previously completed work.

Using fiber we can directly mutate the tree not like react element tree., since fiber nodes can mutate, React doesn’t need to recreate every node for updates; it can simply clone and update the node when there is an update.In the case of a fiber tree, React doesn’t perform recursive traversal.Once the render phase completes, React moves on to the commit phase, where it basically swaps the root pointers of the current tree and workInProgress tree,And what about the 16ms frame time? React effectively runs an internal timer for each unit of work being performed and constantly monitors this time limit while performing the work.The moment the time runs out, React pauses the current unit of work, hands the control back to the main thread, and lets the browser render whatever is finished at that point.Then, in the next frame, React picks up where it left off and continues building the tree. Then, when it has enough time, it commits the workInProgress tree and completes the render.

1. **Why we need keys in React?When we need keys in React?**

ANS: In React when we render the array of elements/component then react need the unique identifier for each element or component i.e key . it needs key because when we tried to add element to an array if the array are uniquely separated then React don’t which element / component to rerender so it traverse whole array and rerenders with all elements in to prevent this React needs the Unique ID then only that element will rerender . Keys are needed only at the time when we are rendering array of elements/components.

1. **Can we use index as keys in React?**

Ans: React team don’t recommend to use index as keys because in React APP there will be scenarios with multiple arrays of elements so if we use the index as key then it will effect the rerendering phase key must be unique.

1. **What is props in React? Ways to use**

ANS: Props are the custom properties that can be passed from parent to child components . It mainly used to create reusable components like card component. Props can be accessed through props.property\_name or destructure the props {property\_name}

1. **What is config driven UI?**

ANS: Config drive UI means all the data which will be dispalyed in the frontend is coming from backend as a config file (In most of the case JSON file). Ex: Whether to display the carousel is decided by config,config file will not be same it can be differ like corresponding to location it is modern way to create scaled web application with config data.