1. Tell us the differences between uncontrolled and controlled components.

Controlled components manage and update their state through props passed down from a parent component, while uncontrolled components manage their state internally without relying on props from a parent component.

2. How to validate React props using PropTypes

React provides a way to validate props passed to a component using PropTypes. PropTypes is a type-checking library that can be used to ensure that the props passed to a component are of the expected type. To use PropTypes, you first need to import it from the 'prop-types' package. Then, you can define the expected type of each prop in the component's propTypes object

3. Tell us the difference between nodejs and express js.

Node.js is a runtime environment that allows JavaScript to be run on the server-side. It provides the ability to execute JavaScript outside of a web browser. Express.js, on the other hand, is a web application framework built on top of Node.js. It provides a set of tools and features that make it easier to build web applications using Node.js. Node.js provides the basic building blocks for server-side JavaScript programming, such as file I/O, network communication, and event handling. Express.js builds on top of Node.js by providing a layer of abstraction for common web application functionality, such as routing, middleware, and template engines. In short, Node.js is a runtime environment for JavaScript, while Express.js is a web application framework that runs on top of Node.js. Express.js simplifies web application development by providing a set of tools and features that are not included in Node.js out of the box.

4. What is a custom hook, and why will you create a custom hook?

A custom hook is a reusable function in React that encapsulates stateful logic across different components. They can reduce code duplication, improve code readability, and make your code more modular. You might create a custom hook to extract and reuse shared logic across multiple components.