Question 1

Tell us differences between uncontrolled and controlled compon

Controlled and uncontrolled components are terms used to describe React components that render HTML form e
their value and state
A **controlled component** is one that takes its current value through props and notifies changes through callback
callback and managing its own state and passing the new values as props to the controlled component.
A **uncontrolled component** is one that stores its own state internally, and we query the DOM using a ref to find
HTML.

Question 2

How to validate React props using PropTypes ?

React props are attributes that you can pass from one component to another. They are read-only and can be of
arrays1. To validate React props using PropTypes, we need to do the following steps:

- Install the prop-types library using npm or yarn. This library provides a range of validators for checking the typ
- Import PropTypes from "prop-types" in your component file.
- Define the propTypes property on our component and assign it an object with the prop names and validators
- Optionally, you can also define the defaultProps property on your component and assign it an object with the

---

Question 3

Tell us the difference between nodejs and expressjs.

Node.js and **Express.js** are not comparable in terms of functionality, because they are different in terms of abstr
Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It is used to run JavaScript code outsid
Node.js provides a low-level API for working with HTTP requests, responses, streams, buffers, events, etc. **Nod**
developers to install and use thousands of third-party packages
Express.js is a web framework based on **Node.js** that simplifies the development of web applications using **Noc**
templating, error handling, static file serving, etc. that make it easier to organize and structure your code and har
popular frameworks in the **Node.js** ecosystem and has many extensions and plugins available
In summary , **Node.js** is a platform for running JavaScript code on servers, while **Express.js** is a framework for
without **Express.js**, but we cannot use **Express.js** without **Node.js**

Question 4

What is a custom hook, and why will we create a custom hook

A **custom hook** is a function that starts with the word “use” and can call other hooks inside it. **Custom hook**s all
changing their structure124. **Custom hook**s can return anything we want, such as values, functions, or effects.
We might create a **custom hook** when we want to extract some common logic from our components and share i
fetching data from an API, for accessing the local storage, for subscribing to an event, or for validating a form. Cr