

Q1. Create a class component "Animals" that returns a list of options (the options are "select" input options) of animals and a class component "Birds" that returns a list of options ("select" input options) of birds. Create another component "PetSelector" that renders a usable select type of input with both animals and birds as options.

Pet Selector

Select

Q2. Create a class component that renders an input and focuses the input on mount.

Q3. Create a class component Form which renders controlled text inputs for name, address and age. Create another class component FormParent which renders Form and a "Clear" button. Clicking this button should clear out the values inside Form.

Name:

Address:

Age:

Clear Form

Q4. Create an class component that renders its children and catches error throw inside the children. If any error is present, it renders the error message instead of the children.

Cricket

Football

Children is broken!!!

Q5. What can be used to return more than one component from a react component?

In React, we can return more than one component from a component using a technique called "Fragments".

It allow us to group a list of children elements without adding an extra node to the DOM.

Fragments can be declared using a shorthand syntax with the angle brackets `<>` and `</>`, or using the explicit `<React.Fragment>` and `</React.Fragment>` tags.

Q6. What is the difference between a React Component and a React PureComponent?

A PureComponent only re-renders when its props or state have actually changed, while a regular component may re-render even if there are no actual changes.

Q7. What can be used to access the methods and properties of a child component from a component?

In React, you can access the methods and properties of a child component from a parent component by passing them down as props.

Q8. What is used to access the methods and properties of a nested child component from a component?

To access the methods and properties of a nested child component from a component in React, we can use two technique:

Prop drilling: It is the process of passing down a prop through multiple levels of nested components to reach a component that needs access to that prop. While this technique can work for simple cases, it can become unwieldy and difficult to maintain as the number of levels of nesting increases.

React context:- Context allows you to pass data through the component tree without having to pass props down manually at every level .To avoid prop drilling and make it easier to access nested components, we can use react context.

Q9. Name at least one method that is used to create an Error Boundary in React.

The **static `getDerivedStateFromError()`** method is used to create an Error Boundary in React.