

Name - Ajeet kumar singh
Reg. No. - 12006488
Email. - ajeetku4975@gmail.com
Phone no. - 7061955007
Solution of the Assignment

Q.1. Explain what the simple List component does?

=> The simple List component is a React component that renders a list of items. Here's what it does:

It has a single prop items which is an array of objects, each object having a text property.

It renders an unordered list () with each item from the items array as a list item ().

It uses a memoized SingleListItem component to render each list item.

Each list item can be clicked and will change color to green, indicating that it is selected.

When an item is selected, the component will remember its index and re-render with that item's background color set to green.

When the items prop changes, the component will reset the selected index to null. It uses the PropTypes library to define the expected types of its props.

Q.2. What problems / warnings are there with code ?

=> Here are some problems / warnings with the given code and their suggested solutions:

In order to pass a parameter in a function in the 'onClick' event, the attribute should be returned by an arrow function. For example, it should be `onClick={() => onClickHandler(index)}` instead of `onClick={onClickHandler(index)}`.

The 'index' and 'isSelected' props should be required, along with the 'onClickHandler' and 'text' props. You can set this requirement in the PropTypes validation, for example

```
WrappedSingleListItem.propTypes = { index: PropTypes.number.isRequired, isSelected:
PropTypes.bool.isRequired, onClickHandler: PropTypes.func.isRequired, text:
PropTypes.string.isRequired, };
```

The 'setSelectedIndex' function should come after the 'selectedIndex' state variable when creating the useState hook. This is because the first value returned from the useState hook is the current state value, and the second value is the function to update it. For example, it should be `const [selectedIndex, setSelectedIndex] = useState();` instead of `const [setSelectedIndex, selectedIndex] = useState();`.

The 'isSelected' prop should pass a boolean value by checking if the 'selectedIndex' is equal to the 'index'. For example, it should be `isSelected={selectedIndex === index}` instead of `isSelected={selectedIndex}`.

Each child in a list should have a unique "key" prop. This is important for React to efficiently update the list when changes occur. For example, it should be `key={index}` inside the map function.

The `PropTypes` validation for the 'items' prop should use 'arrayOf' instead of 'array' and 'shape' instead of 'shapeOf'. For example, it should be: `WrappedListComponent.propTypes = { items: PropTypes.arrayOf(PropTypes.shape({ text: PropTypes.string.isRequired, })), };`

The default props for the 'items' prop should be an empty object array instead of a null value. This is to avoid any issues with rendering the list when there are no items. For example, it should be `WrappedListComponent.defaultProps = { items: [{ text: 'No items' }] };.`

Q.3. Please fix, optimise, and/or modify the component as much as you think this is necessary?

=>

```
import React, { useState, useEffect, useMemo, useCallback } from 'react';
```

```
function List({ items, selectedIndex, handleItemClick }) {
  const renderedItems = useMemo(
    () =>
      items.map((item, index) => (
        <li
          key={index}
          onClick={() => handleItemClick(index)}
          style={{
            textAlign: 'left',
          }}
        >
          {item.text}
        </li>
      )),
    [items, selectedIndex, handleItemClick]
  );

  return <ul>{renderedItems}</ul>;
}
```

```
function AddItem({ onAddItem }) {
  const [text, setText] = useState("");

  const handleSubmit = e => {
    e.preventDefault();
    if (text.trim()) {
```

```

        onAddItem(text);
        setText("");
    }
};

return (
    <form onSubmit={handleSubmit}>
        <input
            type="text"
            value={text}
            onChange={e => setText(e.target.value)}
            placeholder="Enter item text"
        />
        <button type="submit">Add Item</button>
    </form>
);
}

function App() {
    const [items, setItems] = useState([
        { id: 1, text: 'Name - Ajeet Kumar Singh' },
        { id: 2, text: 'Stream - B.Tech(CSE)' },
        { id: 3, text: 'Location - India' },
        { id: 4, text: 'University - Lovely Professional University' },
        { id: 5, text: 'Year - 2024' },
    ]);

    const [selectedIndex, setSelectedIndex] = useState(null);

    useEffect(() => {
        setSelectedIndex(null);
    }, [items]);

    const handleItemClick = useCallback(index => {
        setSelectedIndex(index);
    }, []);

    const handleAddItem = useCallback(
        text => {
            const newItem = { id: items.length + 1, text };
            setItems([...items, newItem]);
        },
        [items]
    );
}

```

```
return (  
  <div>  
    <AddItem onAddItem={handleAddItem} />  
    <List  
      items={ [...items]}  
      selectedIndex={selectedIndex}  
      handleItemClick={handleItemClick}  
    />  
  </div>  
);  
}  
  
export default App;
```