

- Explain what the simple List component does.
- What problems / warnings are there with code?
- Please fix, optimize, and/or modify the component as much as you think is necessary.

```
1.
2. import React, { useState, useEffect, memo } from 'react';
3. import PropTypes from 'prop-types';
4.
5. // Single List Item
6. const WrappedSingleListItem = ({
7.   index,
8.   isSelected,
9.   onClickHandler,
10.  text,
11. }) => {
12.  return (
13.    <li
14.      style={{ backgroundColor: isSelected ? 'green' : 'red' }}
15.      onClick={onClickHandler(index)}
16.    >
17.      {text}
18.    </li>
19.  );
20. };
21.
22. WrappedSingleListItem.propTypes = {
23.  index: PropTypes.number,
24.  isSelected: PropTypes.bool,
25.  onClickHandler: PropTypes.func.isRequired,
26.  text: PropTypes.string.isRequired,
27. };
28.
29. const SingleListItem = memo(WrappedSingleListItem);
30.
31. // List Component
32. const WrappedListComponent = ({
33.  items,
34. }) => {
35.  const [setSelectedIndex, selectedIndex] = useState();
36.
37.  useEffect(() => {
38.    setSelectedIndex(null);
```

```

39.   }, [items]);
40.
41.   const handleClick = index => {
42.     setSelectedIndex(index);
43.   };
44.
45.   return (
46.     <ul style={{ textAlign: 'left' }}>
47.       {items.map((item, index) => (
48.         <SingleListItem
49.           onClickHandler={() => handleClick(index)}
50.           text={item.text}
51.           index={index}
52.           isSelected={selectedIndex}
53.         />
54.       ))}
55.     </ul>
56.   )
57. };
58.
59. WrappedListComponent.propTypes = {
60.   items: PropTypes.array(PropTypes.shapeOf({
61.     text: PropTypes.string.isRequired,
62.   })),
63. };
64.
65. WrappedListComponent.defaultProps = {
66.   items: null,
67. };
68.
69. const List = memo(WrappedListComponent);
70.
71. export default List;
72.

```

Answer01: The code defines a simple React List component that displays a list of items, where each item is rendered as a list item. The component receives an array of items as a prop and maps through the array to render each item using the `SingleListItem` component.

The `SingleListItem` component is a functional component that receives some props related to the item being rendered, such as `isSelected`, `onClickHandler`, and `text`. It renders the item as a list item, with a background color that is set based on the `isSelected` prop. When the list item is clicked, it calls the `onClickHandler` function with the item's index.

The `List` component is another functional component that receives an array of items and renders them using the `SingleListItem` component. It also manages the state of the currently selected item using the `useState` hook, and resets the selected index when the `items` prop changes using the `useEffect` hook.

Answer02:

`array` is not a valid `propType`. It should be `PropTypes.arrayOf()`.

The initial state of the `selectedIndex` state variable is not set. It should be initialized with a default value of `null`.

The `onClick` event handler is not passed properly. It should be wrapped in an arrow function to prevent it from being called immediately.

Answer03:

```
import React, { useState, useEffect, memo } from 'react';
import PropTypes from 'prop-types';

// Single List Item
const SingleListItem = memo(({ index, isSelected, onClickHandler, text }) => {
  const handleClick = () => onClickHandler(index);

  return (
    <li
      style={{ backgroundColor: isSelected ? 'green' : 'red'}}
      onClick={handleClick}
    >
      {text}
    </li>
  );
});

SingleListItem.propTypes = {
```

```

    index: PropTypes.number.isRequired,
    isSelected: PropTypes.bool.isRequired,
    onClickHandler: PropTypes.func.isRequired,
    text: PropTypes.string.isRequired,
  };

  // List Component
  const List = memo(({ items }) => {
    const [selectedIndex, setSelectedIndex] = useState(null);

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

    if (!items || items.length === 0) {
      return <p>No items found</p>;
    }

    const handleClick = index => {
      setSelectedIndex(index);
    };

    return (
      <ul style={{ textAlign: 'left' }}>
        {items.map((item, index) => (
          <SingleListItem
            key={index}
            onClickHandler={handleClick}
            text={item.text}
            index={index}
            isSelected={selectedIndex === index}
          />
        ))}
      </ul>
    );
  });

  List.propTypes = {
    items: PropTypes.arrayOf(PropTypes.shape({
      text: PropTypes.string.isRequired,
    })),
  };

  List.defaultProps = {
    items: null,
  };

```

```
};
```

```
export default List;
```