

Question -1)

### **React's components list creation work:**

In order to create lists of pieces, students must use the character characteristic "key," as specified in the React documentation. To determine whether items in a list had already been modified, added, or destroyed, React need so many keys. An individual identifier is required for each object. The id about an entity usually works nicely for all of that.

### **Collection but instead attributes presume:**

Whenever making lists of components in Javascript, users must use a special word attribute called "key". React uses keys to indicate whether additional burdens have been modified, removed, or altered. Or, to put it another way, designers may say that keywords are applied to identify the components in collections.

Question-2) errors and problems in code

```
1) WrappedListComponent.propTypes = {  
  items: PropTypes.array(PropTypes.shapeOf({  
    text: PropTypes.string.isRequired,  
  })),  
};
```

In this **PropTypes.array(PropTypes.shapeOf())** is creating the error.

2)

Second error I have faced that is we are using maps on items so it becomes necessary to check that map is not empty.

3)

Third and last problem I found in this is ->

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

this was original code =>

here the setter and state are replaced the general format to declare a state is like this =>

```
const[stateValue, stateValueSetter] = useState(initial value);
```

But in original code **setter is written before and state later** that's why it caused error.

Question3) fixed code with no erros or warning

```
import './styles.css';
```

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

```

import PropTypes from 'prop-types';

// Single List Item
const WrappedSingleListItem = ({
  index,
  isSelected,
  onClickHandler,
  text,
}) => {
  return (
    <li
      style={{ backgroundColor: isSelected ? 'green' : 'red'}}
      onClick={onClickHandler(index)}
    >
      {text}
    </li>
  );
};

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

const SingleListItem = memo(WrappedSingleListItem);

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

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

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

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

```

```

        />
      )}}
    </ul>
  )
};

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

// items: PropTypes.array(PropTypes.shape({
//   text: PropTypes.string.isRequired,
// })),

WrappedListComponent.defaultProps = {
  items: null,
};

const List = memo(WrappedListComponent);

export default List;

```

Thank you 😊