Name:- Shivam Kumar

Frontend

Reg no:12012376

1.   Explain what the simple List component does.

The simple List component is a React functional component that displays a list of items. Each item in the list is represented by a single list item, and when an item is clicked, its index is passed to a callback function.

2. What problems / warnings are there with code?

There are a few issues with the code:

- Giving the key attribute in map to distinguish the different elements of items array.
- I make isSelected to the number so that I get the index of clicked element which I used to change the color of clicked elements.
- In the WrappedListComponent, the setSelectedIndex is not being called correctly so I write the selectedIndex, and setSelectedIndex at its correct place and passing null as initial value.
- The PropTypes for the items prop in the WrappedListComponent are not defined correctly. PropTypes.array should be PropTypes.arrayOf, and PropTypes.shapeOf should be PropTypes.shape.

3. Please fix, optimize, and/or modify the component as much as you think is necessary.

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

import PropTypes from 'prop-types';


// Single List Item
const WrappedSingleListItem = ({

  index,

  isSelected,

  onClickHandler,

  text,

}) => {
```

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


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


const SingleListItem = memo(WrappedSingleListItem);


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

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

```jsx
  const handleClick = index => {

    setSelectedIndex(index);

  };


  return (

    <ul style={{ textAlign: 'left' }}>

      {items.map((item, index) => (

        <SingleListItem

          key={index}

          onClickHandler={() => handleClick(index)}

          text={item.text}

          index={index}

          isSelected={selectedIndex}

        />

      ))}

    </ul>

  )

};


WrappedListComponent.propTypes = {

  items: PropTypes.arrayOf(PropTypes.shape({

    text: PropTypes.string.isRequired,

  })).isRequired,

};


const List= memo(WrappedListComponent);


export default List;
```
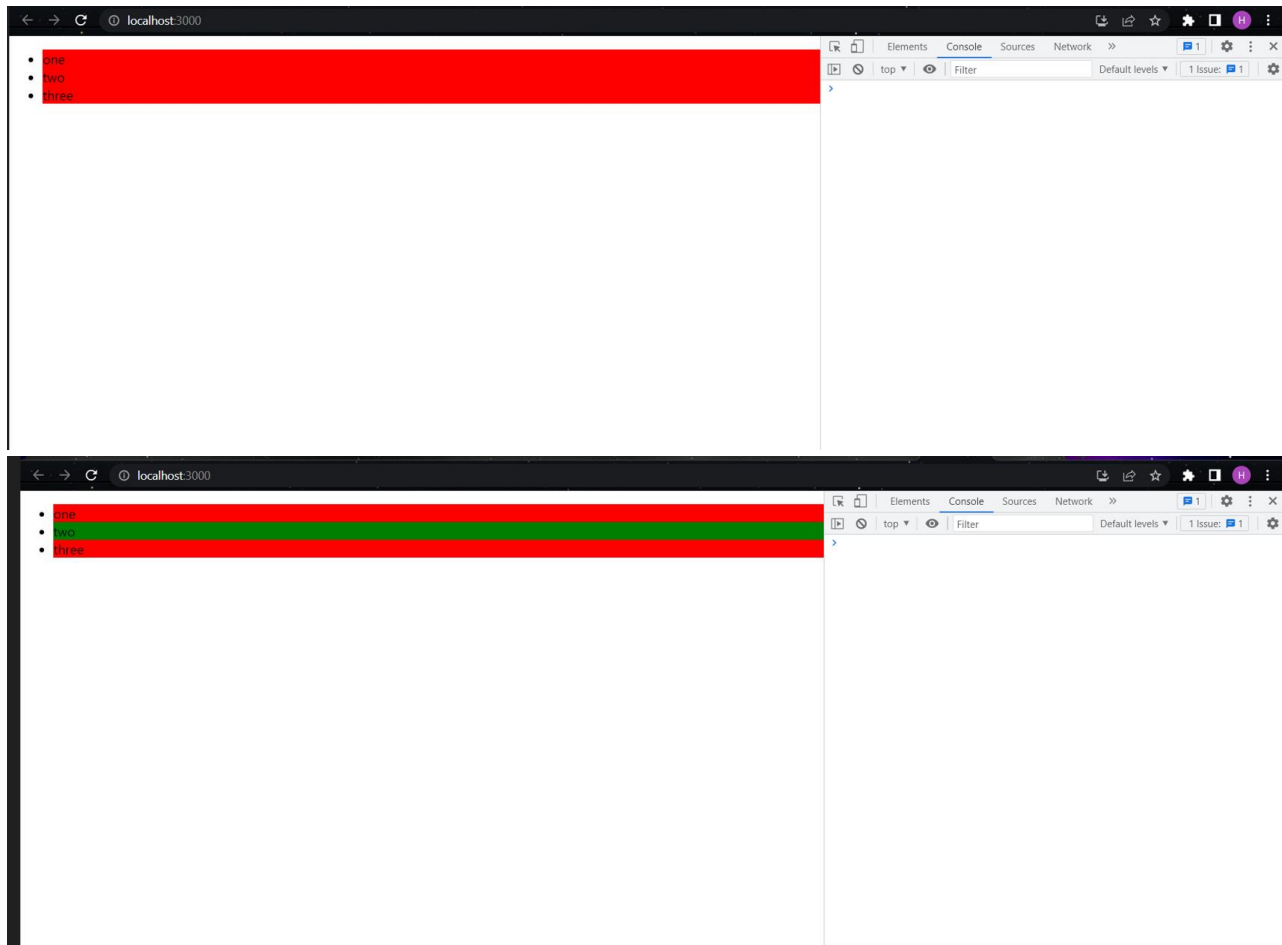
this is how it looks after correction of code

1) Problem-shapeOf is not a function

To fix this issue, , PropTypes.shape should be used instead of PropTypes.shapeOf and PropTypes.arrayOf

should be used instead of PropTypes.array. WrappedListComponent.propTypes = {

 items: PropTypes.arrayOf(PropTypes.shape({

```
    text: PropTypes.string.isRequired,

 })),

};
```

Compiled with problems:

ERROR

```
prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
TypeError: prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
    at ./src/List.js (http://localhost:3000/static/js/bundle.js:175:116)
    at options.factory (http://localhost:3000/static/js/bundle.js:42453:31)
    at __webpack_require__ (http://localhost:3000/static/js/bundle.js:41877:33)
    at fn (http://localhost:3000/static/js/bundle.js:42110:21)
    at ./src/App.js (http://localhost:3000/static/js/bundle.js:15:63)
    at options.factory (http://localhost:3000/static/js/bundle.js:42453:31)
    at __webpack_require__ (http://localhost:3000/static/js/bundle.js:41877:33)
    at fn (http://localhost:3000/static/js/bundle.js:42110:21)
    at ./src/index.js (http://localhost:3000/static/js/bundle.js:304:62)
    at options.factory (http://localhost:3000/static/js/bundle.js:42453:31)
```

2) Problem-setSelectedIndex is not a function

The error message suggests that setSelectedIndex is not a function.
The issue is with this line:

const [setSelectedIndex, selectedIndex] = useState();

To fix this issue, we need to provide an initial value for the state. Since setSelectedIndex is a function that updates the selectedIndex state, we can initialize setSelectedIndex to a function that sets the initial state value:

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

3) Problem-child in a list should have unique property

Each child element in a list should have a unique "key" prop. When rendering a list of elements in React, each item should have a unique key so that React can identify which item has changed, been added, or been removed when the list is updated. To resolve this warning, you can add a unique "key" prop to each item in the list

4)Problem-error with isSelected

I change isSelected as number so that I get the current clicked index .