**1.Explain what the simple List component does.**

In List component the information of array **items** sent to SingleListItem component.

The information of items will be presented as list in SingleListItem component.

In List component state variable **selectedIndex** is initialized as null.When we click a list then selectedIndex will store the value of index.

*PropTypes.arrayOf*property is used to validate the type of items to be an array of objects.  *PropTypes.shape* inside the *arrayOf*property is used to specify the object properties of the elements in the items array. isRequired property is used to make the individual object property to be present in the object, it will raise a warning if that property is not present in the object.

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

**i)** isSelected: PropTypes.bool this statement is wrong it will be isSelected: PropTypes.number because it store index number

ii) const [setSelectedIndex, selectedIndex] = useState() this statement is wrong it will be const [selectedIndex ,setSelectedIndex] = useState() because selectedIndex is state variable.

iii)  Keys should be given to the elements inside the array to give the elements a stable identity.

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

        <SingleListItem

          key={index}  //key is to use a string that uniquely identifies a list item among its siblings

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

          text={item.text}

          index={index}

          isSelected={selectedIndex}

        />

      ))}

iv) items: PropTypes.array(PropTypes.shapeOf({

text: PropTypes.string.isRequired,

It will be arrayOf instead of array and shape instead of shapeOf.

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

Below I have modified the code:

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.number,//here instead of bool will be number

  onClickHandler: PropTypes.func.isRequired,

  text: PropTypes.string.isRequired,

};

const SingleListItem = memo(WrappedSingleListItem);

// List Component

const WrappedListComponent = ({

  items,

}) => {

  const [selectedIndex,setSelectedIndex] = useState();//..here state variable will be selectedIndex

  useEffect(() => {

    setSelectedIndex(null);

  }, [items]);

  const handleClick = index => {

    setSelectedIndex(index);

  };

  return (

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

      {items.map((item, index) => (

        <SingleListItem

          key={index}  //key is to use a string that uniquely identifies a list item among its siblings

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

          text={item.text}

          index={index}

          isSelected={selectedIndex}

        />

      ))}

    </ul>

  )

};

WrappedListComponent.propTypes = {

  items: PropTypes.arrayOf(PropTypes.shape({//..Here will be arrayOf instead of array and shape instead of shapeOf

    text: PropTypes.string.isRequired,

  })),

};

WrappedListComponent.defaultProps = {

  items: null,

};

const List = memo(WrappedListComponent);

export default List;