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

The List component allows you to display a list of pages or links. You can build list of links by defining a parent page, manually choosing pages, or defining search criteria.

List get an array of objects which are data to display and renders children SingleListItem components and then pass them the data that they need to display and also it the click handler to update the state to know which list item is selected.

1. **Problems/ warnings with the code.**

1] setSelectedIndex is missing

2] Got one warning

React Hook useEffect has a missing dependency: 'setSelectedIndex'. Either include it or remove the dependency array react-hooks/exhaustive-deps

3] Parameters are misplaced in const[setSelectedIndex, selectedIndex] = useState(null). We need to swap them

4] index needs to be removed from these

WrappedSingleListItem

WrappedSingleListItem.propTypes

5] Error in return function

Index and key needs to be replaced

6] onClick ={onclickHandler}

1. **Final Code**

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

import PropTypes from 'prop-types';

// Single List Item

*const* WrappedSingleListItem = ({

*isSelected*,

*onClickHandler*,

*text*,

}) *=>* {

  return (

    <li

      style={{ backgroundColor: *isSelected* ? 'green' : 'red'}}

      onClick={*onClickHandler*}

    >

      {*text*}

    </li>

  );

};

WrappedSingleListItem.propTypes = {

  isSelected: PropTypes.bool,

  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*]);

*const* handleClick = *index* *=>* {

    setSelectedIndex(*index*);

  };

  return (

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

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

        <*SingleListItem*

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

          text={*item*.text}

          key={*index*}

          isSelected={selectedIndex === *index*}

        />

      ))}

    </ul>

  )

};

WrappedListComponent.propTypes = {

  items: PropTypes.array(PropTypes.shape({

    text: PropTypes.string.isRequired,

  })),

};

WrappedListComponent.defaultProps = {

  items: null,

};

*const* List = memo(WrappedListComponent);

export default List;

**Output:**

****

****