**NAME**-JALAJ MAHESHWARI

**REGISTRATION NO.** : 12015163

**ROLE** : FRONTEND

1. Github Link : <https://github.com/jalajmaheshwari/Jalaj-Frontend-assignment.git>
2. Live Link : <https://jalaj-frontend-assignment.netlify.app/>

Question 1) Explain what the simple List component does.

Ans) The List component is a React functional component that renders an unordered list of items. It accepts a single prop items which is an array of objects. Each object in the items array should have a text property that is a string.

The List component then maps through each item in the items array and renders a SingleListItem component for each item. The SingleListItem component is also a functional component that renders a single list item.

The SingleListItem component takes four props:

index: the index of the item in the items array

isSelected: a boolean value that indicates whether or not the item is selected

onClickHandler: a function that is called when the list item is clicked

text: the text to display in the list item

The SingleListItem component is wrapped in the memo HOC, which means that it will only re-render if its props have changed.

The List component uses the useState hook to keep track of the currently selected item's index in the selectedIndex state variable. It also uses the useEffect hook to reset the selectedIndex to null whenever the items prop changes.

When a list item is clicked, the handleClick function is called with the index of the clicked item. This updates the selectedIndex state variable, which causes the corresponding SingleListItem component to re-render with the isSelected prop set to true.

Overall, the List component provides a simple and reusable way to render a list of items with the ability to select one item at a time.

Question 2) What problems / warnings are there with code?

There are a few problems/warnings with the code:

The useState hook is not used correctly. The setSelectedIndex function should be called to update the selectedIndex state variable, not passed as the first element of the destructured array. To fix this, the WrappedListComponent function should use the following line instead: const [selectedIndex, setSelectedIndex] = useState();

The PropTypes declaration for the items prop in the WrappedListComponent is not correct. Instead of PropTypes.array(PropTypes.shapeOf({...})), it should be PropTypes.arrayOf(PropTypes.shape({...})).

The onClick event handler in the SingleListItem component is not defined correctly. It should be a function that is passed as a callback to the onClick prop, like this: onClick={() => onClickHandler(index)}.

To fix these issues, the code can be updated as follows:

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.map((item, index) => (

<SingleListItem

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

text={item.text}

index={index}

isSelected={selectedIndex === index}

key={index}

/>

))}

</ul>

);

};

WrappedListComponent.propTypes = {

items: PropTypes.arrayOf(

PropTypes.shape({

text: PropTypes.string.isRequired,

})

),

};

WrappedListComponent.defaultProps = {

items: null,

};

const List = memo(WrappedListComponent);

export default List;

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

**List.js**

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

import PropTypes from 'prop-types';

import axios from 'axios';

// Single List Item

const WrappedSingleListItem = ({

  index,

  isSelected,

  onClickHandler,

  id,

  body,

  email,

  name

}) => {

  return (

    <div style={{display:'flex',justifyContent:'center',alignItems:'center',overflowY:'scroll'}}>

      <div class="card" style={{ width: '50rem'}}>

        <div class="card-header" style={{color:'blue'}}>

          <b>Comment :</b>&nbsp;{id}

        </div>

        <div class="card-body">

          <h5 class="card-title">{email}</h5>

          <p class="card-text">{body}</p>

          <a href="#" class="btn btn-primary">Go somewhere</a>

        </div>

      </div>

    </div>

  );

};

WrappedSingleListItem.propTypes = {

  index: PropTypes.number,

  isSelected: PropTypes.bool,

  onClickHandler: PropTypes.func.isRequired,

  id: PropTypes.number,

  name: PropTypes.string.isRequired,

  email: PropTypes.string.isRequired,

  body: PropTypes.string.isRequired,

};

const SingleListItem = memo(WrappedSingleListItem);

// List Component

const WrappedListComponent = () => {

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

  const [items, setItems] = useState([]);

  const [searchQuery, setSearchQuery] = useState('');

  const getData = async () => {

    try {

      const response = await axios.get('https://jsonplaceholder.typicode.com/comments');

      console.log(response.data)

      setItems(response.data);

    } catch (error) {

      console.log(error);

    }

  };

  useEffect(() => {

    getData();

  }, []);

  const handleClick = useCallback((index) => {

    setSelectedIndex(index);

  }, []);

  const handleSearch = useCallback((event) => {

    setSearchQuery(event.target.value);

  }, []);

  const filteredItems = items.filter((item) => {

    return (

      item.name.toLowerCase().includes(searchQuery.toLowerCase()) ||

      item.email.toLowerCase().includes(searchQuery.toLowerCase()) ||

      item.body.toLowerCase().includes(searchQuery.toLowerCase()) ||

      item.id.toString().toLowerCase().includes(searchQuery.toLowerCase())

    );

  });

  return (

    <div >

      <input type="text" value={searchQuery} placeholder='Search User.....'  onChange={handleSearch}style={{  width: '300px', marginBottom: '50px', marginleft: '50px',borderRadius:'5px' }} />

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

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

          <SingleListItem

            key={index}

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

            index={index}

            id={item.id}

            name={item.name}

            email={item.email}

            body={item.body}

            isSelected={index === selectedIndex}

          />

        ))}

      </div>

    </div>

  );

};

WrappedListComponent.propTypes = {

  items: PropTypes.arrayOf(

    PropTypes.shape({

      text: PropTypes.string.isRequired,

    })

  ),

};

WrappedListComponent.defaultProps = {

  items: null,

};

const List = memo(WrappedListComponent);

export default List;

**Navbar.js**

import React from 'react'

const Navbar = () => {

    return (

        <nav class="navbar navbar-expand-lg navbar-light bg-light">

            <div class="container-fluid">

                <a class="navbar-brand" href="#">UserDetails</a>

                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">

                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                            <li class="nav-item">

                                <a class="nav-link active" aria-current="page" href='/'>Home</a>

                            </li>

                            <li class="nav-item">

                                <a class="nav-link" href='/userlist'>UsersList</a>

                            </li>

                    </ul>

                </div>

            </div>

        </nav>

    )

}

export default Navbar