**ASSIGNEMENT REPORT**

**(SteelEye Assignment : Front End Development)**

Submitted by

## Name:- Harsh Tripathi

## Registration Number:- 12007942

**Role :- Front End Developer**

Submitted To

## (SteelEye Company)

**School of Computer Science and Engineering**



GithubLink : <https://github.com/HarshTripathih/harsh_Frontend.git>

Deployed Link : <https://harsh-frontend.netlify.app/>

1. Question 1 : - Explain what the simple List component does.

The List component is a React component that displays a list of items. Each item is represented by a SingleListItem component, which renders an HTML list item element. The SingleListItem component receives four props: index, isSelected, onClickHandler, and text. The index prop is a number that represents the index of the item in the list. The isSelected prop is a boolean that indicates whether the item is currently selected. The onClickHandler prop is a function that is called when the item is clicked, and it is responsible for updating the state of the component to indicate which item is selected. The text prop is a string that represents the text of the item.

The List component receives a single prop, items, which is an array of objects that represent the items in the list. Each object has a single property, text, which is a string that represents the text of the item.

The List component maps over the items array and renders a SingleListItem component for each item in the array. The SingleListItem component receives the appropriate props, including the onClickHandler function, which is bound to the index of the current item in the map function. When an item is clicked, the onClickHandler function is called with the index of the clicked item as its argument, and it updates the state of the component to indicate which item is selected.

The List component uses the useState hook to maintain the state of the selected item, and it uses the useEffect hook to reset the selected item when the items prop changes. Finally, the List component is wrapped in the memo higher-order component to optimize performance by preventing unnecessary re-renders.

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

1. The state variable setSelectedIndex should be defined as const instead of let.
2. The selectedIndex state variable is not initialized with a value, which can cause issues when checking its truthiness in the isSelected prop of SingleListItem. It should be initialized with a value of -1.
3. In the onClick prop of SingleListItem, the onClickHandler function is being called immediately instead of being passed as a callback function. It should be wrapped in an arrow function to prevent this behavior: onClick={() => onClickHandler(index)}.
4. The propTypes definition for the items prop in WrappedListComponent should be PropTypes.arrayOf(PropTypes.shape(...)) instead of PropTypes.array(PropTypes.shape(...)).
5. The default value for the items prop in WrappedListComponent should be an empty array ([]) instead of null.
6. The memo HOC in SingleListItem and WrappedListComponent should be removed because there are no expensive calculations being performed in these components that would benefit from memoization.

**The correct code with this modification :**

1. : Fix the useState declaration in the WrappedListComponent:

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

1. :Update the PropTypes declaration for the items array in the WrappedListComponent:

WrappedListComponent.propTypes = {

items: PropTypes.arrayOf(PropTypes.shape({

text: PropTypes.string.isRequired,

})),

};

1. : Modify the onClickHandler prop passed to the SingleListItem component to prevent it from being called immediately

<SingleListItem

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

text={item.text}

index={index}

isSelected={selectedIndex === index}

/>

Q3). Please fix, optimize, and/or modify the component as much as you think is necessary.

For this question I have Modified This code Using API data fetching data and showed it in web page and I have implemented the search filter functionality also .

1. **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,

firstName,

username,

email,

phone,

image,

domain,

age,

gender,

maidenName,

lastName,

birthDate,

bloodGroup,

}) => {

return (

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

<div class="card" style={{ width: '30rem' }}>

<h4>{id}</h4>

<img src={image} class="card-img-top" alt="loading" />

<div class="card-body">

<h5 class="card-title">

{firstName} &nbsp;{maidenName}&nbsp;{lastName}

</h5>

<li class="list-group-item">

<b>UserName</b>:&nbsp;{username}

</li>

<p class="card-text">

Some quick example text to build on the card title and make up the bulk of the card's

content.

</p>

</div>

<ul class="list-group list-group-flush">

<li class="list-group-item">

<b>Age</b>:&nbsp;{age}

</li>

<li class="list-group-item">

<b>Gender</b>:&nbsp;{gender}

</li>

<li class="list-group-item">

<b>BirthDate</b>:&nbsp;{birthDate}

</li>

<li class="list-group-item">

<b>bloodGroup</b>:&nbsp;{bloodGroup}

</li>

<li class="list-group-item">

<b>Email</b>:&nbsp;{email}

</li>

<li class="list-group-item">

<b>Phone</b>:&nbsp;{phone}

</li>

<li class="list-group-item">

<b>Domain</b>:&nbsp;{domain}

</li>

</ul>

<div class="card-body">

<a href="#" class="btn btn-primary">

Higher Now

</a>

</div>

</div>

</div>

);

};

WrappedSingleListItem.propTypes = {

index: PropTypes.number,

isSelected: PropTypes.bool,

onClickHandler: PropTypes.func.isRequired,

id: PropTypes.number,

firstName: PropTypes.string.isRequired,

maidenName: PropTypes.string.isRequired,

lastName: PropTypes.string.isRequired,

age: PropTypes.number,

gender: PropTypes.number,

birthDate: PropTypes.number,

bloodGroup: PropTypes.string.isRequired,

username: PropTypes.string.isRequired,

email: PropTypes.string.isRequired,

phone: PropTypes.string.isRequired,

domain: PropTypes.string.isRequired,

image: PropTypes.any.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://dummyjson.com/users');

console.log(response.data.users);

setItems(response.data.users);

} catch (error) {

console.log(error);

}

};

1. **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