steeleye_frontend_assignment

AKASH NAGINENI - 12008727

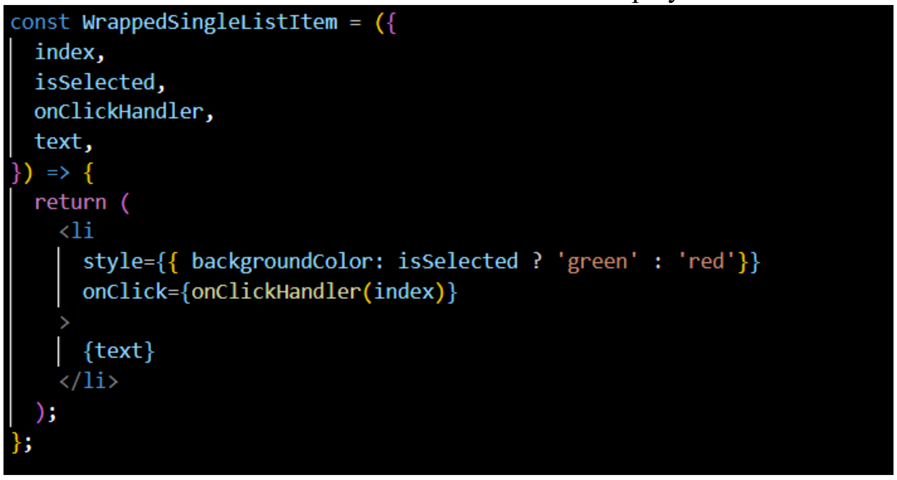
1. **What the List Component does?**

   Here, the WrappedListComponent passed inside memo () to reduce the unnecessary renders and to increase the optimization. When a component is wrapped in memo (), React renders the component and memoizes the result. Before the next render, if the new props are the same, react reuses the memoized result, skipping the next rendering. So, the List is actually the WrappedListComponnet Component. Now, if we talk about the work done by List Component, it takes an object of array as props and simply iterate over thar array. We are iterating on an array using in-built map () method. And for every iteration we are passing the data of each object into SingleListItem as a prop.
   In the List Component, we are also checking and defining the prop Types. So, it checks whether the receiving prop is an array of objects or not and the text in those is string form or not. If not is throws a warning.

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

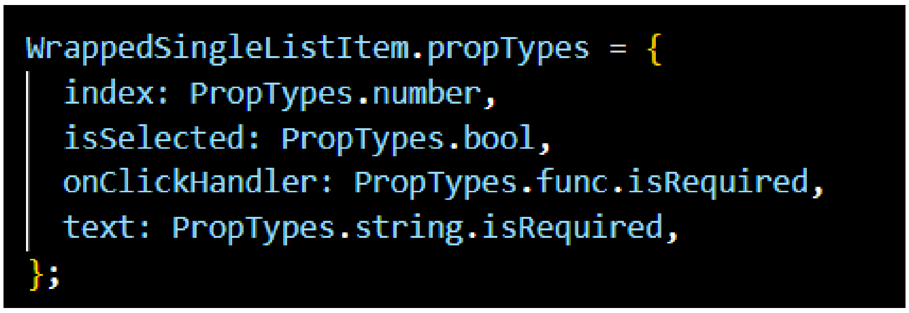
   There are few errors/warnings in the code with which the component is not rendering. The errors I have noticed are.

   1. Here in onClick function the onClickHandler () function is getting called without a call back function because of which the results are not displayed.



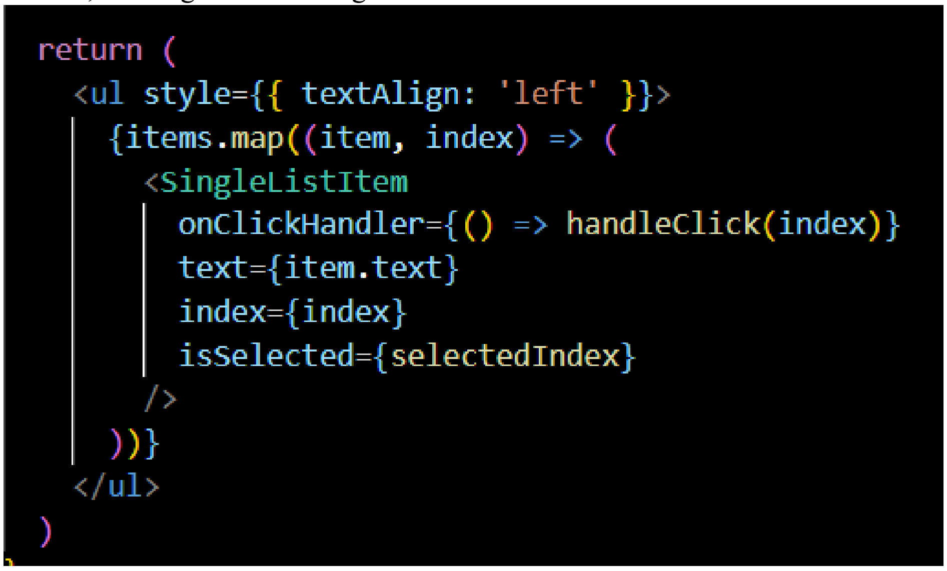
```
const WrappedSingleListItem = ({
  index,
  isSelected,
  onClickHandler,
  text,
}) => {
  return (
    <li
      style={{ backgroundColor: isSelected ? 'green' : 'red'}}
      onClick={onClickHandler(index)}
    >
      {text}
    </li>
  );
};
```

   2. While doing prop checking using prop type, it is mentioned that isSelected should be of a Boolean type. But from List Component we are passing isSelected with an integer value. So, this going to give a warning.



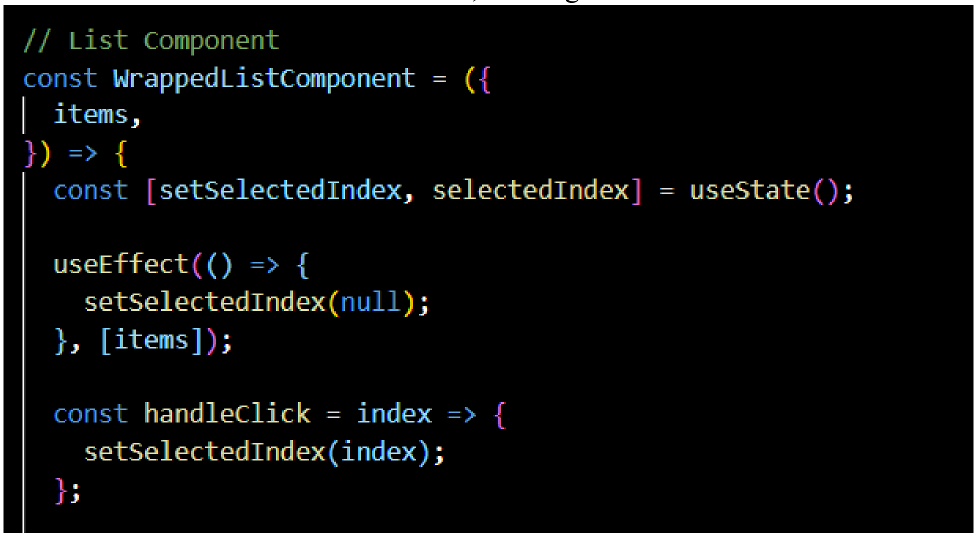
```
WrappedSingleListItem.propTypes = {
  index: PropTypes.number,
  isSelected: PropTypes.bool,
  onClickHandler: PropTypes.func.isRequired,
  text: PropTypes.string.isRequired,
};
```

3. To increase the optimization and reduce error occurring chance it is a good practice to pass key value, while iterating the array of objects using map. But in the code, it is not written, which gives a warning.



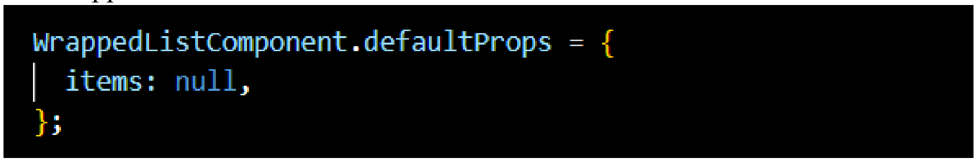
```
  return (
    <ul style={{ textAlign: 'left' }}>
      {items.map((item, index) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={item.text}
          index={index}
          isSelected={selectedIndex}
        />
      ))}
    </ul>
  )
```

4. In react useState () is a predefined Hook to use variables and to main the state globally in a component. It actually returns an array of length 2, in which the first element is a variable which stores data, and the second element is a function that is used to update the state. In the given code, the syntax of the useState () is written in wrong way. First a function is written and then the variable, which gives an error.



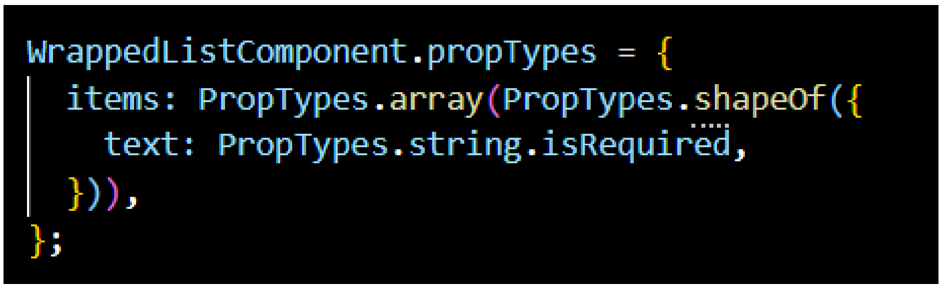
```
// List Component
const WrappedListComponent = ({
  items,
}) => {
  const [setSelectedIndex, selectedIndex] = useState();

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

  const handleClick = index => {
    setSelectedIndex(index);
  };
```

5. Giving null as default props is not a good practice. This may sometimes lead to crashing of the app.



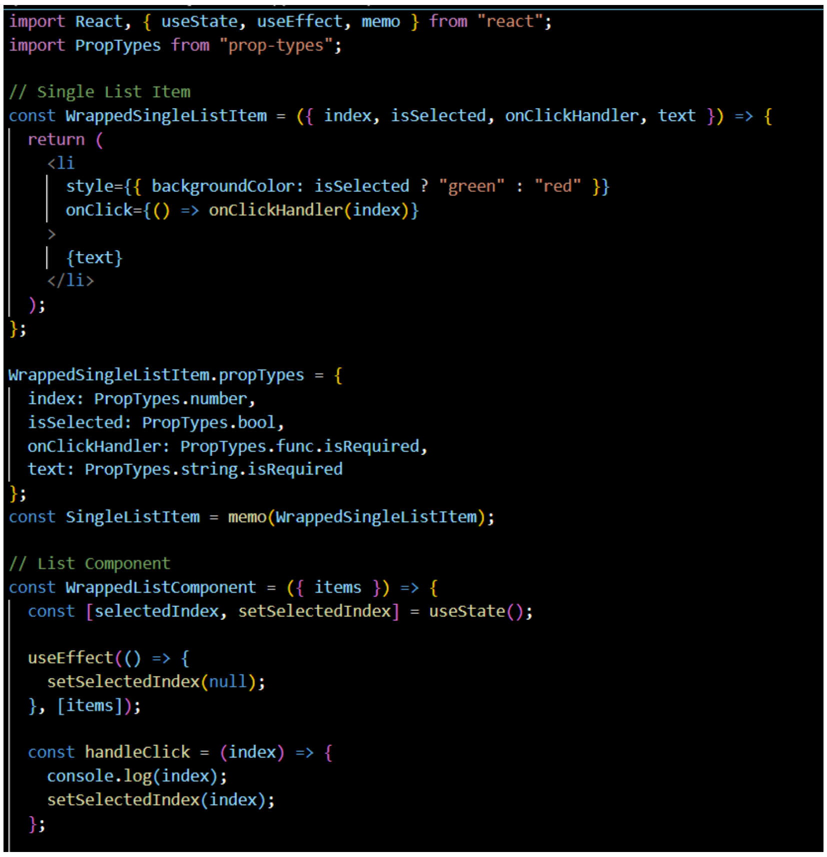
```
WrappedListComponent.defaultProps = {
  items: null,
};
```

6. An error in the prop Type checking syntax in the List Component. Here the code should be arrayOf () and .shape (). But in given code it is wrong syntax which gives an error, and it is mandatory to write .Requires after the propType is initialized. This is also missing in the code.

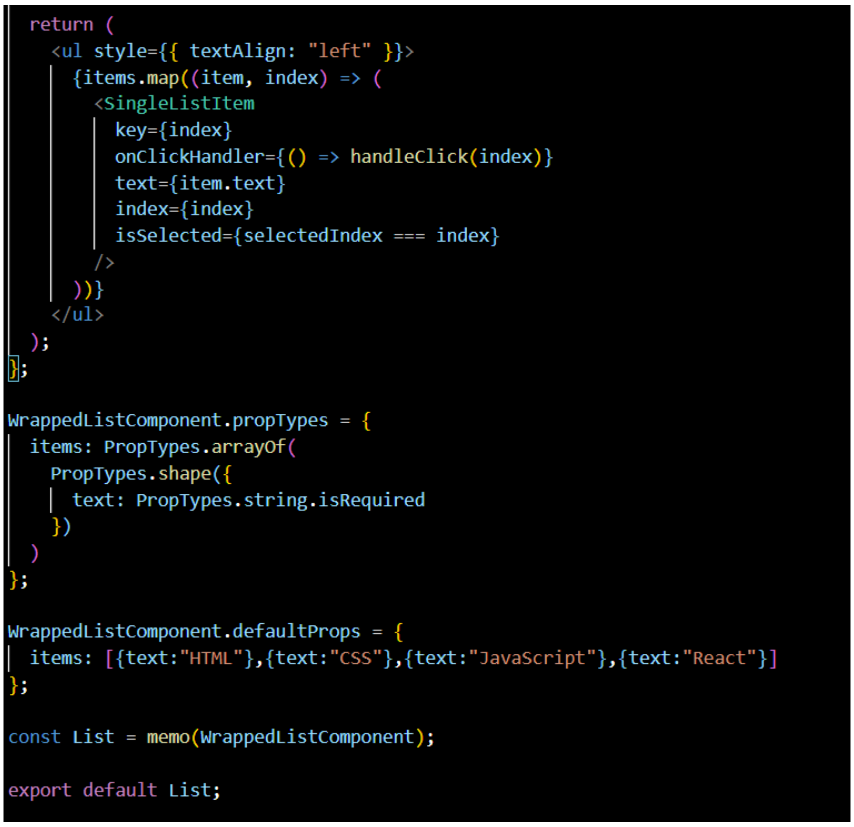


```
WrappedListComponent.propTypes = {
  items: PropTypes.array(PropTypes.shapeOf({
    text: PropTypes.string.isRequired,
  })),
};
```

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 }) => {
  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) => {
    console.log(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 === index}
        />
      ))}
    </ul>
  );
};

WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(
    PropTypes.shape({
      text: PropTypes.string.isRequired
    })
  )
};

WrappedListComponent.defaultProps = {
  items: [{text:"HTML"},{text:"CSS"},{text:"JavaScript"},{text:"React"}]
};

const List = memo(WrappedListComponent);

export default List;
```

Explanation:

1. First, I'm passing default props as an array of objects and getting through object destructing as props into the List Component. As I have to pass this data further individually, I'm iterating over this data using map (). As we are iterating, I'm passing the key value as the index of the array, which is a good practice. And I'm passing the data into SingleListItem Component as a prop.
2. Here, according to my approach as the code already have a useState () Hook with selectedIndex and setSelectedIndex as elements in List component and passing isSelected, onClickHandler which is passing a callback function handleClick.
3. In the SingleListItem Component, I'm receiving the index, text and all the remaining props which I have passed as props previously.
4. Now when I click on particular element, using callback I'm passing that index of that particular element to the parent component, that is list component, where I'm checking whether the isSelected and index is equal or not. If equal, then a Boolean value is passed to the child component through isSelected prop and change the color accordingly as I'm setting a condition that isSelected true color will be green else the colour will be red.

THANK YOU
AKASH NAGINENI