Steeleye Frontend Assignment

Name: Prince Kr Deka

Reg No: 12006624

Lovely Professional University

Q1. Explain what the simple List component does.

This code creates the "List" React component, which displays a list of things with selectable options. "WrappedSingleListItem" and "WrappedListComponent" are the two sub-components that make up the component.

A single item in the list is represented by the "WrappedSingleListItem" component, a memoized functional component. It requests the four arguments "index," "isSelected," "onClickHandler," and "text" and, depending on the selection status, delivers a list item element with a background colour.

Additionally a memoized functional component, the "WrappedListComponent" produces a list of items by utilising the "WrappedSingleListItem" component. It accepts the "items" parameter, which is an array of objects with the "text" and "key" attributes. The component keeps track of the index of the currently selected item using the "useState" hook and the "useEffect" hook.

Q2. What problems / warnings are there with code?

Errors

Error 1

PropTypes is a type-checking library in React that helps validate props passed to components and throws errors if the prop types don't match the expected type.

The code bellow has a syntax error. The “shapeOf” should be replaced with “shape”.

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

Fixed code:

```
WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(
    PropTypes.shape({
      text: PropTypes.string.isRequired,
```

```
    })
  ),
};
```

Error 2:

The code bellow has a syntax error. The “array” should be replaced with “arrayOf”. As there is no keyword named “array” in “prop-type” in rather it is “arrayOf”

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

Fixed code:

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

Error 3:

In the code below the “**selectedIndex**” is a useState variable and “**selectedIndex”** is a useState function which is used to update the value of “**selectedIndex**” variable. In a useStatate function variable the first parameter should be the variable and the second parameter should be the function used to update the variable. In the code the places of the two parameters were interchanged in a improper way.

Two fix the code the two parameters were interchanged.

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

Fixed code:

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

Error 4:

To initialize a variable with an empty array we should use “[]”. But in the code the variable items is initialized with “null” . This is an anti-pattern in Reactjs development, Using “null” is not recommended.

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

Fixed code:

```
WrappedListComponent.defaultProps = {
  items: [],
};
```

Warnings

Warning 1:

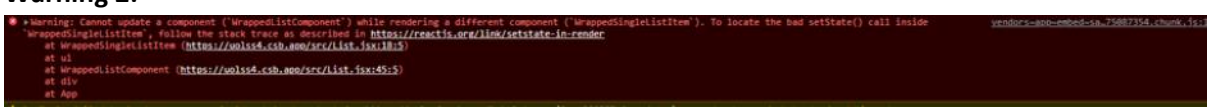
While mapping the elements of an array using array “map” function a unique “key” is required to distinguish and keep a track of the elements traversed. The key was missing earlier which was fixed in my code. Refer to “**fixed code".**


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

Fixed Error:

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

Warning 2:



The problem is that calling those setters while rendering is never a good idea. The result will be a warning message. To get rid from this warning we have to use an “**arrow function**”.

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

Warning 3:

Here the "**isSelected** " is expecting a bool value as parameter and validates using react "**prop-types" .** To fix this error we have to replace **{selectedIndex} to === {selectedIndex === index}.**

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

Fixed code:

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

Q3. Fixed, optimize, and/or modify the component.

Ans:

Here is the complete code of "List" component with appropriate changes

```
import React, { useState, useEffect, memo } from 'react';
import PropTypes from 'prop-types';
```

```
// Single List Item
const WrappedSingleListItem = memo(({
  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(null);

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

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

  return (
    <ul style={{ textAlign: 'left' }}>
      {items.map(({text, key}, index) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={text}
          index={index}
          isSelected={selectedIndex === index}
          key = {key}
        />
      ))}
      <button onClick={()=>setSelectedIndex(null)}>clear</button>

    </ul>
```

```
  )
};

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

WrappedListComponent.defaultProps = {
  items: [],
};

const List = memo(WrappedListComponent);

export default List;
```

Output