

Q (1) Explain what the simple List component does.

- ➔ List component are used to display data in an ordered format and mainly used to display menus on website. There is also a handleClick function which we pass to the Single ListItem component by which the background color of that particular item change. React reuses the memoized content from the samereference. By memoized, it states that it doesn't re-render unnecessarily. There is also used map() function for traversing the list

Example

```
Const List = ({list}) => (
  <ul>
    {( list || [] ).map( item => (
      < ListItem key = { item.id } item = { item } />
    ))}
  </ul>
);
```

Q (2) What problems/warnings are there with the code?

- ➔ Syntax errors in the following code

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

- ➔ Unique key prop is missing for List items

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

- ➔ Defining a default prop as null is not recommended

- ➔ onClick events should have a function reference instead of function call

```
<li style={{ backgroundColor: isSelected ? "green" : "red" }}
  onClick={onClickHandler(index)}>
```

```
        {text}  
      </li>
```

- ➔ useState variables being misplaced
- ➔ Passing a number selectedIndex to isSelected which should be a bool
- ➔ Using Single Quotes instead of Back-ticks
- ➔ Not Using the right version of what a tutorial is using
- ➔ Confusing function with class
- ➔ Confusing curly brackets with parenthesis

Q (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(Boolean(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(true);  
  }, [items]);  
  
  const handleClick = (index) => {  
    setSelectedIndex(Boolean(index));  
    console.log(index);  
  };  
  
  return (  
    <ul style={{ textAlign: "left" }}>  
      {items.map((item, index) => (  

```

```

        <SingleListItem
        onClickHandler={() => handleClick(index)}
        text={item.text}
        index={index}
        isSelected={selectedIndex}
        key={index}
        />
    )}
  );
};

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

WrappedListComponent.defaultProps = {
  items: [
    { text: "Niketan", index: false },
    { text: "Wadaskar", index: false }
  ]
};

const List = memo(WrappedListComponent);

export default List;`

```