1) Explain what the simple List component does.

- The 'WrappedListComponent' takes in an array of objects called 'items' as props, where each object in the array must have a parameter named "text" of type string. Using the 'map' function, the component iterates through the 'items' array and renders a 'SingleListItem' component for each object. The 'SingleListItem' component receives props such as onClickHandler, text, index, and isSelected from the 'WrappedListComponent'.

- The 'index' prop is unique for each 'SingleListItem' component and helps to identify the clicked list item. Initially, the 'selectedIndex' value is null, but changes to the respective index value of the clicked list item when the user interacts with it.

- The 'isSelected' prop is used to highlight the selected list item by changing its background color to green. If 'isSelected' is false, the background color will be red. Initially, all list items have a red background.

- To avoid unnecessary re-rendering, the 'WrappedSingleListItem' component is memoized and only re-renders when the value of the 'isSelected' prop changes. This ensures that the component will not re-render unnecessarily even if the user double-clicks on a list item.

- In summary, the 'WrappedListComponent' creates a list of items using the 'WrappedSingleListItem' component, which is also a memoized functional component.

2) What problems / warnings are there with code?

Ans: There are multiple problems/warnings with the code that are to be rectified to render the component correctly.

I. **Uncaught TypeError: PropTypes.shapeOf is not a function**, the propTypes declaration for the items prop in the WrappedListComponent component is wrong

```
Code Given :
WrappedListComponent.propTypes = {
  items: PropTypes.array(PropTypes.shapeOf({//ShapeOf is not a Valid Function
    text: PropTypes.string.isRequired,
  })),
};

Edited Code :
WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(PropTypes.shape({//Correct Function is ArrayOf
    text: PropTypes.string.isRequired,
  })),
};
```

II. **The onClickHandler property in WrappedSingleList Item is called incorrectly**; it should be called as a callback to be triggered when the list item is clicked.

```
Code Given :
const WrappedSingleListItem = ({
  index,
  isSelected,
  onClickHandler,
  text,
}) => {
  return (
    <li
      style={{ backgroundColor: isSelected ? 'green' : 'red'}}
      onClick={onClickHandler(index)} //callback is not defined
    >
      {text}
    </li>
  );
};

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

III. **The default value of null** for the items prop in WrappedListComponent can lead to issues when attempting to perform mapping operations on it.

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

Modified:
WrappedListComponent.defaultProps = {
  items:  [
    {text : "L Chetan"},
    {text : "12012893"},
    {text : "LPU"},
    {text : "B-Tech"},
    {text : "Cse"},

  ],
};
```

IV. **To ensure that only the clicked option changes color and not all the content present in a list**, we can modify the isSelected property to isSelected = (SelectedIndex === Index). This will allow us to update the selected option's color based on its index in the list.

```
Given:
 return (
    <ul style={{ textAlign: 'left' }}>
      {items.map((item, index) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={item.text}
          index={index}
          isSelected={selectedIndex}// using this syntax, all available
colours are selected and changed.
        />
      ))}
    </ul>
  )
};

Modified:
 return (
    <ul style={{ textAlign: 'left' }}>
      {items.map((item, index) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={item.text}
          index={index}
          isSelected={selectedIndex=== index}// Whenever a user selects an
option, only that option's colour changes.        />
      ))}
    </ul>
  )
};
```

The modification to the isSelected property ensures that only the option selected by the user changes its color, while the rest of the content in the list remains unaffected.

V. **Uncaught TypeError: setSelectedIndex is not a function in the code given**. The useState function used in the WrappedListComponent component initialises the selectedIndex state, but the initial value (null) should appear first in the returned array, not the setter function.

```
Given:
const [setSelectedIndex, selectedIndex] = useState(); //incorrect sequence

Modified:
const [ selectedIndex, setSelectedIndex] = useState(); //corrected code
```

VI. In the map function, **the value of the unique key is not defined as a prop** for each child value.

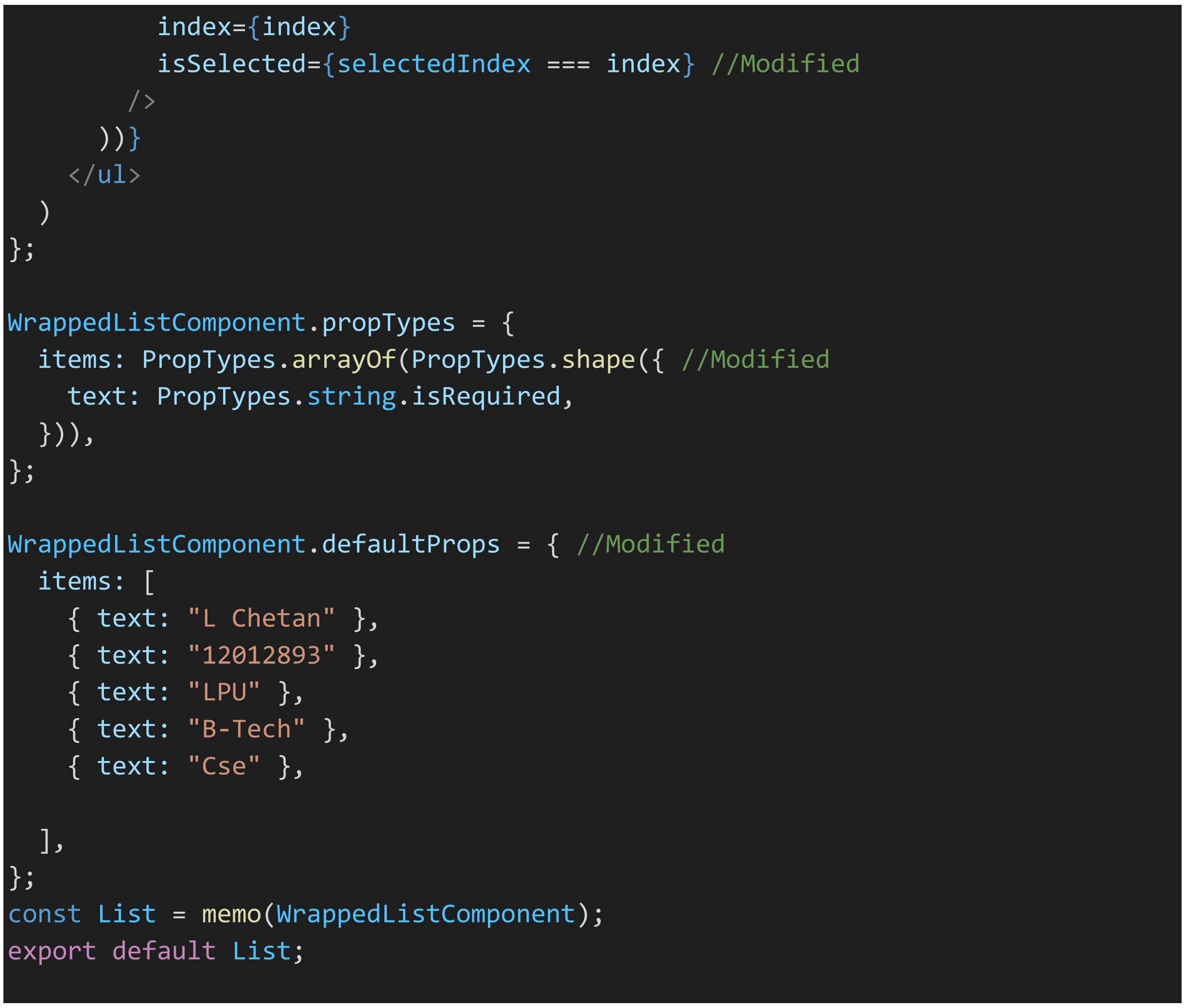
```
Given:
return (
    <ul style={{ textAlign: 'left' }}>
      {items.map((item, index) => (
        <SingleListItem //key is not present
          onClickHandler={() => handleClick(index)}
          text={item.text}
          index={index}
          isSelected={selectedIndex=== index}
        />
      ))}
    </ul>
  )
};

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

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

Ans: The Working Modified Code:

```
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)} //Modified
    >
      {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(); //Modified

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

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

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



```
          index={index}
          isSelected={selectedIndex === index} //Modified
        />
      ))}
    </ul>
  )
};

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

WrappedListComponent.defaultProps = { //Modified
  items: [
    { text: "L Chetan" },
    { text: "12012893" },
    { text: "LPU" },
    { text: "B-Tech" },
    { text: "Cse" },

  ],
};
const List = memo(WrappedListComponent);
export default List;
```