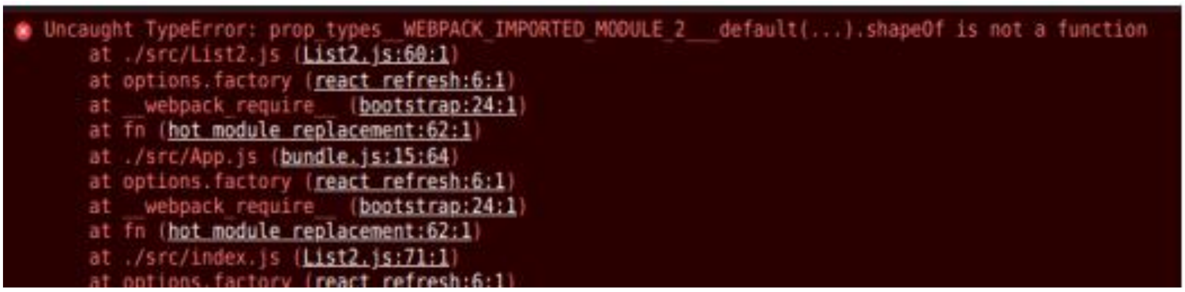
React Component Code Review – STEELEYE TASK

1. **Explain what the simple List component does.**

   Ans : The following function component shows how to render a list of items. It would work the same with a list of numbers instead of strings, we can use the built-in JavaScript array map method to iterate over our list items; and to map them from JavaScript primitive to HTML elements. The WrappedListComponent returns an unordered list on rendering. A SingleListComponent is used in the ul which is rendered for every item in the “items” which is passed as props.

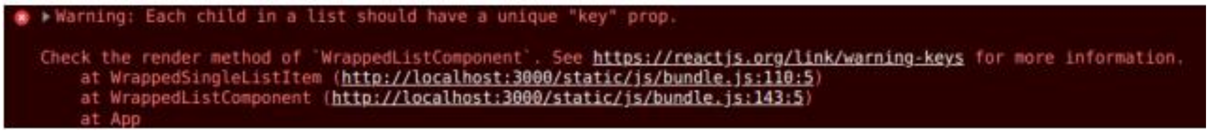
2. Errors

   Problem 1 :



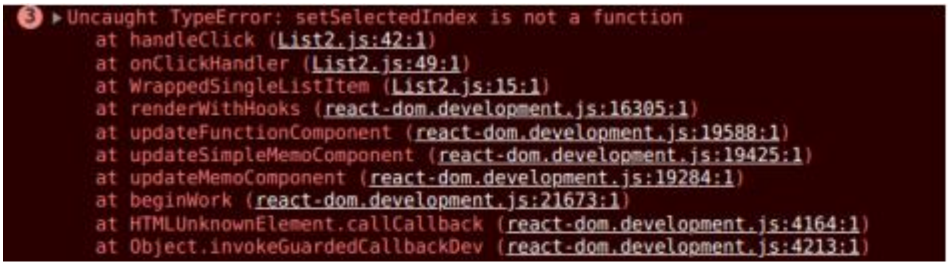
```
Uncaught TypeError: prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
    at ./src/List2.js (List2.js:60:1)
    at options.factory (react refresh:6:1)
    at __webpack_require__ (bootstrap:24:1)
    at fn (hot module replacement:62:1)
    at ./src/App.js (bundle.js:15:64)
    at options.factory (react refresh:6:1)
    at __webpack_require__ (bootstrap:24:1)
    at fn (hot module replacement:62:1)
    at ./src/index.js (List2.js:71:1)
    at options.factory (react refresh:6:1)
```

   Problem 2 :



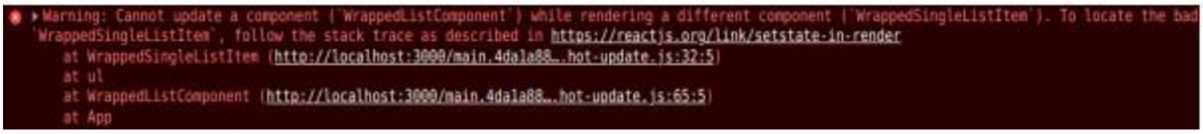
```
▸Warning: Each child in a list should have a unique "key" prop.

Check the render method of `WrappedListComponent`. See https://reactjs.org/link/warning-keys for more information.
    at WrappedSingleListItem (http://localhost:3000/static/js/bundle.js:110:5)
    at WrappedListComponent (http://localhost:3000/static/js/bundle.js:143:5)
    at App
```

   Problem 3 :



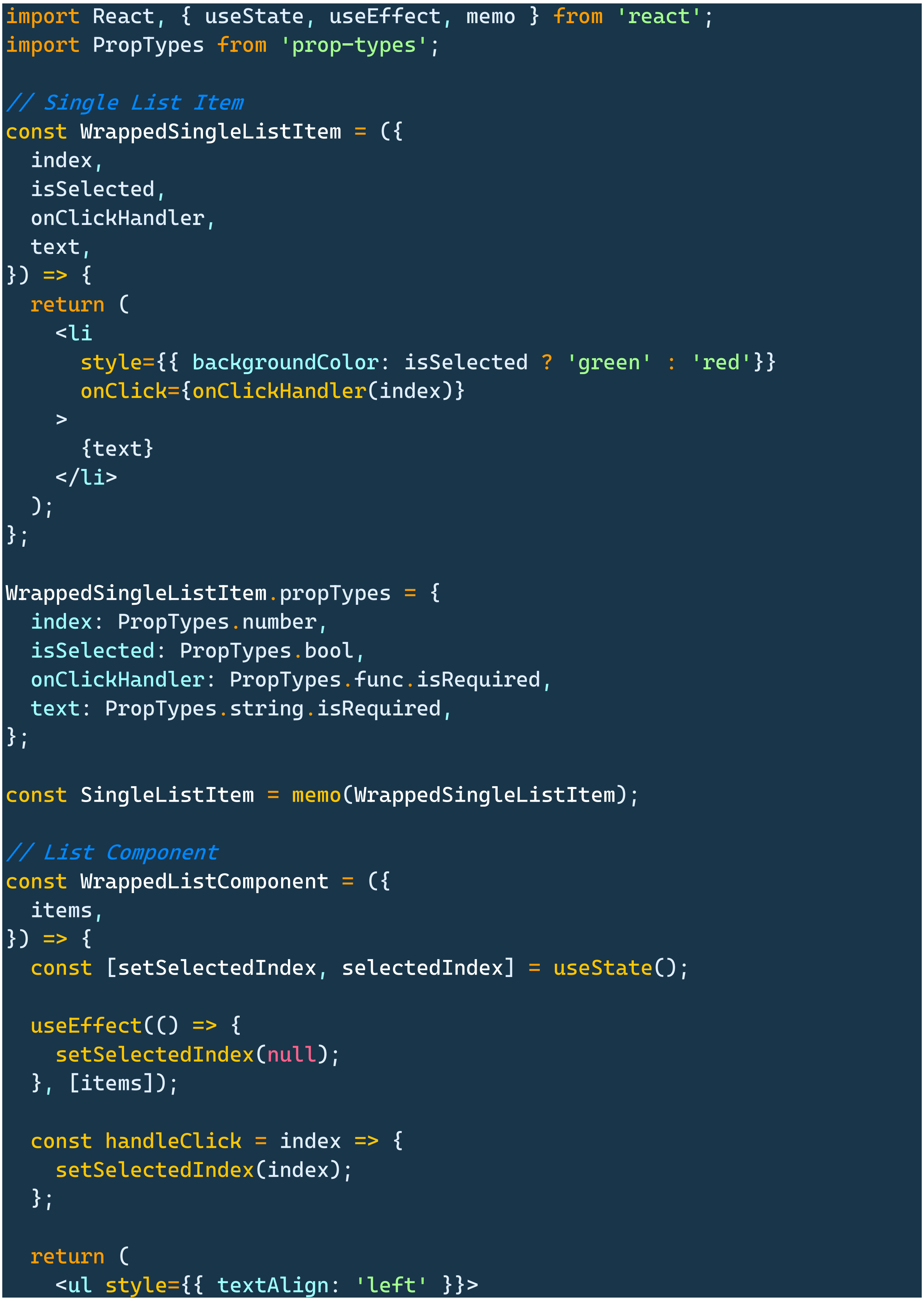
```
3 ▸Uncaught TypeError: setSelectedIndex is not a function
    at handleClick (List2.js:42:1)
    at onClickHandler (List2.js:49:1)
    at WrappedSingleListItem (List2.js:15:1)
    at renderWithHooks (react-dom.development.js:16305:1)
    at updateFunctionComponent (react-dom.development.js:19588:1)
    at updateSimpleMemoComponent (react-dom.development.js:19425:1)
    at updateMemoComponent (react-dom.development.js:19284:1)
    at beginWork (react-dom.development.js:21673:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
    at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
```

   Problem 4:

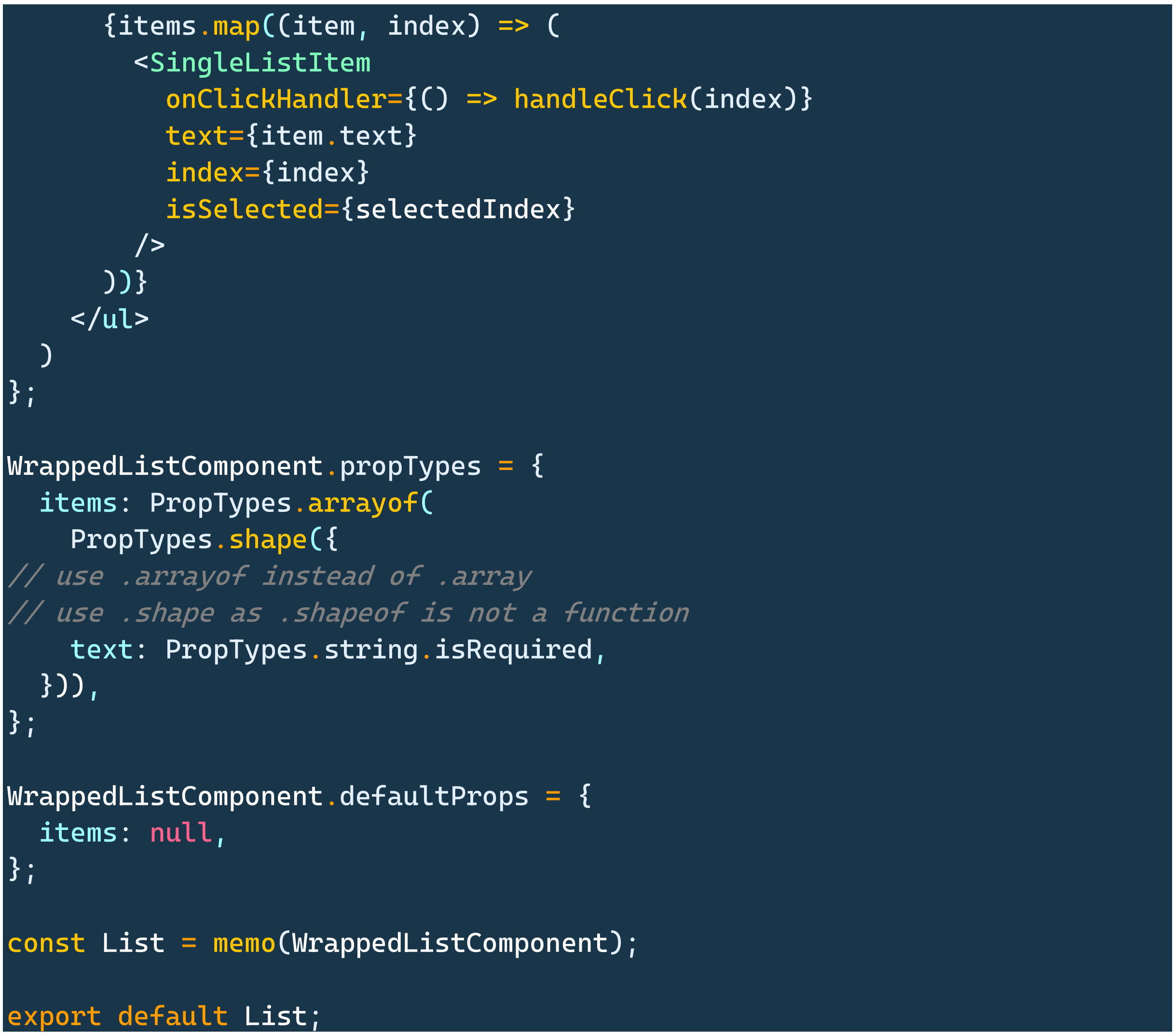


```
▸Warning: Cannot update a component (`WrappedListComponent`) while rendering a different component (`WrappedSingleListItem`). To locate the bad
 `WrappedSingleListItem`, follow the stack trace as described in https://reactjs.org/link/setstate-in-render
    at WrappedSingleListItem (http://localhost:3000/main.4da1a88….hot-update.js:32:5)
    at ul
    at WrappedListComponent (http://localhost:3000/main.4da1a88….hot-update.js:65:5)
    at App
```

3. Fixed Code / Optimized /modified



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

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

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

  return (
    <ul style={{ textAlign: 'left' }}>
```



```
      {items.map((item, index) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={item.text}
          index={index}
          isSelected={selectedIndex}
        />
      ))}
    </ul>
  )
};

WrappedListComponent.propTypes = {
  items: PropTypes.arrayof(
    PropTypes.shape({
// use .arrayof instead of .array
// use .shape as .shapeof is not a function
    text: PropTypes.string.isRequired,
  })),
};

WrappedListComponent.defaultProps = {
  items: null,
};

const List = memo(WrappedListComponent);

export default List;
```