

Ans 1:

The list component has the WrappedListComponent passes inside in memo. The WrappedListComponent contains an unordered list which contains a map function which loops through the items array passed as a props from App.js. For every item in the items array The SingleListComponent is called. The SingleListComponent has a onClick event which calls the handleClick function. The handleClick function is then sets the selected index state.

Ans 2:

The first error which we encounter on running the command npm start in the terminal is:

```
✖ Uncaught TypeError: prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
    at ./src/List2.js (List2.js:59:1)
    at options.factory (react_refresh:6:1)
    at __webpack_require__ (bootstrap:24:1)
    at fn (hot module replacement:62:1)
    at ./src/App.js (logo.svg:34:1)
    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:70:1)
    at options.factory (react_refresh:6:1)
```

```
WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(
    PropTypes.shape({
      // use .arrayOf instead use .array
      // .shapeOf is not func, use .shape
      text: PropTypes.string.isRequired,
    })
  ),
};
```

2<sup>nd</sup> error:

```
✖ ▶ Uncaught TypeError: Cannot read properties of null (reading 'map')
    at WrappedListComponent (List2.js:45:1)
    at renderWithHooks (react-dom.development.js:16305:1)
    at mountIndeterminateComponent (react-dom.development.js:20074:1)
    at beginWork (react-dom.development.js:21587:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
    at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
    at invokeGuardedCallback (react-dom.development.js:4277:1)
    at beginWork$1 (react-dom.development.js:27451:1)
    at performUnitOfWork (react-dom.development.js:26557:1)
    at workLoopSync (react-dom.development.js:26466:1)
```

```
✖ ▶ 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
```

```

return (
  <ul style={{ textAlign: "left" }}>
    {items?.map(
      (
        item,
        index // use optional chaining to prevent the code
        crash when items value is not provided.
      ) => (
        <SingleListItem
          onClickHandler={() => handleClick(index)}
          text={item.text || ""} // if text field is not
provided in json
          index={index}
          isSelected={selectedIndex === index} // correct
the condition isSelected for the list item
          key={index} // unique key prop
        />
      )
    )}
  </ul>
);
};

```

3<sup>rd</sup> error:

```

✖ ▶ Uncaught TypeError: setSelectedIndex is not a function
    at List2.js:37:1
    at commitHookEffectListMount (react-dom.development.js:23150:1)
    at commitPassiveMountOnFiber (react-dom.development.js:24926:1)
    at commitPassiveMountEffects_complete (react-dom.development.js:24891:1)
    at commitPassiveMountEffects_begin (react-dom.development.js:24878:1)
    at commitPassiveMountEffects (react-dom.development.js:24866:1)
    at flushPassiveEffectsImpl (react-dom.development.js:27039:1)
    at flushPassiveEffects (react-dom.development.js:26984:1)
    at react-dom.development.js:26769:1
    at workLoop (scheduler.development.js:266:1)

```

```

const [selectedIndex, setSelectedIndex] = useState();
// error setSelectedIndex is not a func

```

Ans 3:

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

// List Component
const WrappedListComponent = ({ items }) => {
  const [selectedIndex, setSelectedIndex] = useState(); //
  error setSelectedIndex is not a func

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

  const handleClick = (index) => setSelectedIndex(index); //
  code refactor

  return (
    <ul style={{ textAlign: "left" }}>
      {items?.map(
        (
          item,
          index // use optional chaining to prevent the code
          crash when items value is not provided.
        ) => (
          <li
            style={{
              backgroundColor: selectedIndex === index ?
"green" : "red",
              cursor: "pointer", // add css style to look
the list item is clickable
            }}
            onClick={() => handleClick(index)} // wrong way
of calling function
            key={index}
          >
            {item.text || ``}
          </li>
        )
      )
    </ul>
  )
}
```

```

    })
  </ul>
);
};

WrappedListComponent.propTypes = {
  items: PropTypes.arrayOf(
    PropTypes.shape({
      // use .arrayOf instead .array      // .shapeOf is not
      func, use .shape
      text: PropTypes.string.isRequired,
    })
  ),
};

const List = memo(WrappedListComponent);

export default List;

```

**Note:** we can use react virtualization if list items are way too many and re-render can cause slow process processing, we can also use a npm package called "react-window"