

Steel Eye Assignment

What List component does ?

1. List component accepts items (an array of objects) as a props and destructured in the parameters.
2. Inside the component, a **useState** [selectedIndex, setSelectedIndex] is created to track the index of selected list item.
3. Then **useEffect** is used to set the “**selectedIndex**” value to “**null**” when the page renders for first time and adding the dependency array with “**items**” [warning 1].
4. Later, **handleClick** function is created to set the “current” selected index of list item in **selectedIndex** value.
5. Finally, **List** component returns an “**ul**” in which “**items**” was iterated using **map** function (higher order function) and passing the items and some other props while rendering another component “**SingleListItem**”.

Props that are passed:

- onClickHandler => passing **handleClick** function using anonymous function in this variable
- text => passing the data that is present in the object “**item**” using the key “**text**”.
- Index => index (unique number) is passed for tracking the id while using “**map**” function

Later, the props are destructured by “**SingleListItem**” component. By using those props, it sets the **backgroundColor** of “**li**” elements using condition (**Ternary Operator**).

Warnings/Errors:

Warning 1:

```
WARNING in [eslint]
src/List.jsx
  Line 39:6:  React Hook useEffect has a missing dependency: 'setSelectedIndex'. Either include it or remove the dependency array  react-hooks/exhaustive-deps
webpack compiled with 1 warning
```

useEffect dependency array, in the given code, the dependency array was “items” but its not the correct dependency.

To overcome this, we need to pass “setSelectedIndex” because this is modifying in the useEffect.

Solution: check commit “**error 1 resolved**”

Error 1:

```
Uncaught TypeError: prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function bootstrap:27
    at ./src/List.jsx (List.jsx: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:16:63)
    at options.factory (react_refresh:6:1)
    at __webpack_require__ (bootstrap:24:1)
    at fn (hot_module_replacement:62:1)
    at ./src/index.js (List.jsx:71:1)
    at options.factory (react_refresh:6:1)
```

shapeOf is not a function is an error, I encountered and the reason behind is the improper syntax of prototype. To resolve this, I have modified the syntax, refer to “**error 1 resolved**” commit.

Error 2:

ERROR

```
setSelectedIndex is not a function
TypeError: setSelectedIndex is not a function
    at handleClick (http://localhost:3000/main.a30b99f827f55ff6a798.hot-update.js:70:5)
    at onClickHandler (http://localhost:3000/main.a30b99f827f55ff6a798.hot-update.js:77:29)
    at WrappedSingleListItem (http://localhost:3000/main.a30b99f827f55ff6a798.hot-update.js:40:14)
    at renderWithHooks (http://localhost:3000/static/js/bundle.js:22286:22)
    at updateFunctionComponent (http://localhost:3000/static/js/bundle.js:25168:24)
    at updateSimpleMemoComponent (http://localhost:3000/static/js/bundle.js:25030:14)
    at updateMemoComponent (http://localhost:3000/static/js/bundle.js:24903:18)
    at beginWork (http://localhost:3000/static/js/bundle.js:26933:20)
    at HTMLUnknownElement.callCallback (http://localhost:3000/static/js/bundle.js:11878:18)
    at Object.invokeGuardedCallbackDev (http://localhost:3000/static/js/bundle.js:11922:20)
```

setSelectedIndex is not a function, It is due to in correct destructing of `useState()`. As the first parameter must be an variable/state and the second parameter must be an updating function. To remove this error, I swapped the parameters of `useState()` from the given code.

Solution: check “**Error 2 resolved**”.

Warning 2:

```
Warning: Each child in a list should have a unique "key" prop. react-jsx-dev-runtime.development.js:87
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:109:5)
    at WrappedListComponent (http://localhost:3000/static/js/bundle.js:140:5)
    at div
    at App
```

Key prop (unique) is missing while iterating the list of items as it is recommended to add a unique key to each “li” item to avoid uncertainty behaviour of rendering.

Solution: check “warning 2 resolved” commit.

Warning 3:

```
Warning: Failed prop type: Invalid prop `isSelected` of type `number` supplied to
`WrappedSingleListItem`, expected `boolean`.
    at WrappedSingleListItem (http://localhost:3000/static/js/bundle.js:111:5)
    at ul
    at WrappedListComponent (http://localhost:3000/static/js/bundle.js:155:5)
    at div
    at App
```

Invalid prototype of isSelected prop is an warning encountered by me, As “isSelected” holds the value of type “number”. So, I changed the propTypes from “bool” to “number” to resolve this issue.

Solution: check “**warning 3 resolved**”.

Warning 4:

```
Warning: Cannot update a component (`WrappedListComponent`) while rendering a different component
(`WrappedSingleListItem`). To locate the bad setState() call inside `WrappedSingleListItem`, follow
the stack trace as described in https://reactjs.org/link/setstate-in-render
    at WrappedSingleListItem (http://localhost:3000/static/js/bundle.js:109:5)
    at ul
    at WrappedListComponent (http://localhost:3000/static/js/bundle.js:140:5)
    at div
    at App
```

Updating a component while rendering another component is the error occurred in the given code, due to the inconsistency of calling the “handleClick” function when triggering “onClick” event.


To avoid this problem, and to maintain the stack tree, call “handleClick” function in an anonymous function.

Check the proper code here : “**warning 4 resolved**” commit.

Modifications and Optimizations:

- I have added a new state [buffer, setBuffer] to hold the Boolean values for conditional rendering in list component.
- When the list index of isSelected and the index of “li” item is equal, I made buffer to negate the previous value.
- Which means Initially buffer is “false”, then on satisfying the condition, Buffer becomes “false” and vice versa.
- Later, I have added CSS to both “ul” and “li” items for better UI.
- Modification/optimization done in commits : “**added the toggle of bg color of li item**” and “**added CSS for better UI**”. (Refer: Fig1.1 and Fig.1.2)

Output:



item 1
item 2
item 3
item 4
item 5

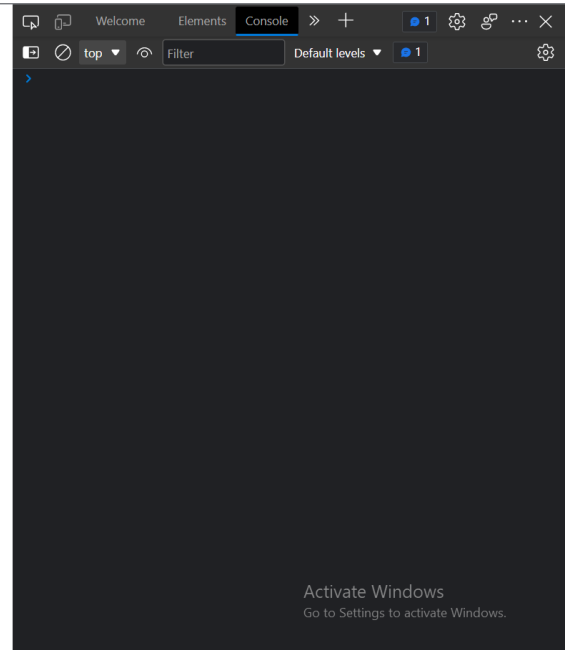



Fig 1.1: Default user Interface (UI) after first time rendering of web page.



item 1
item 2
item 3
item 4
item 5

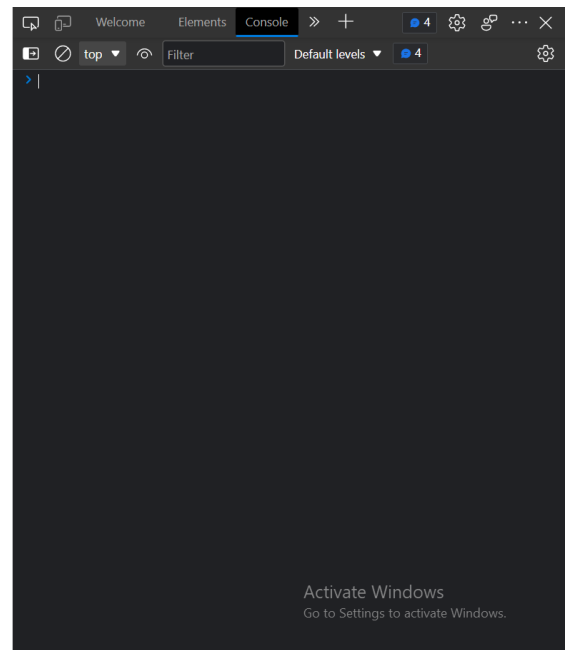


Fig 1.2: User Interface (UI) after selecting an li item.