

STEELEYE ASSIGNMENT

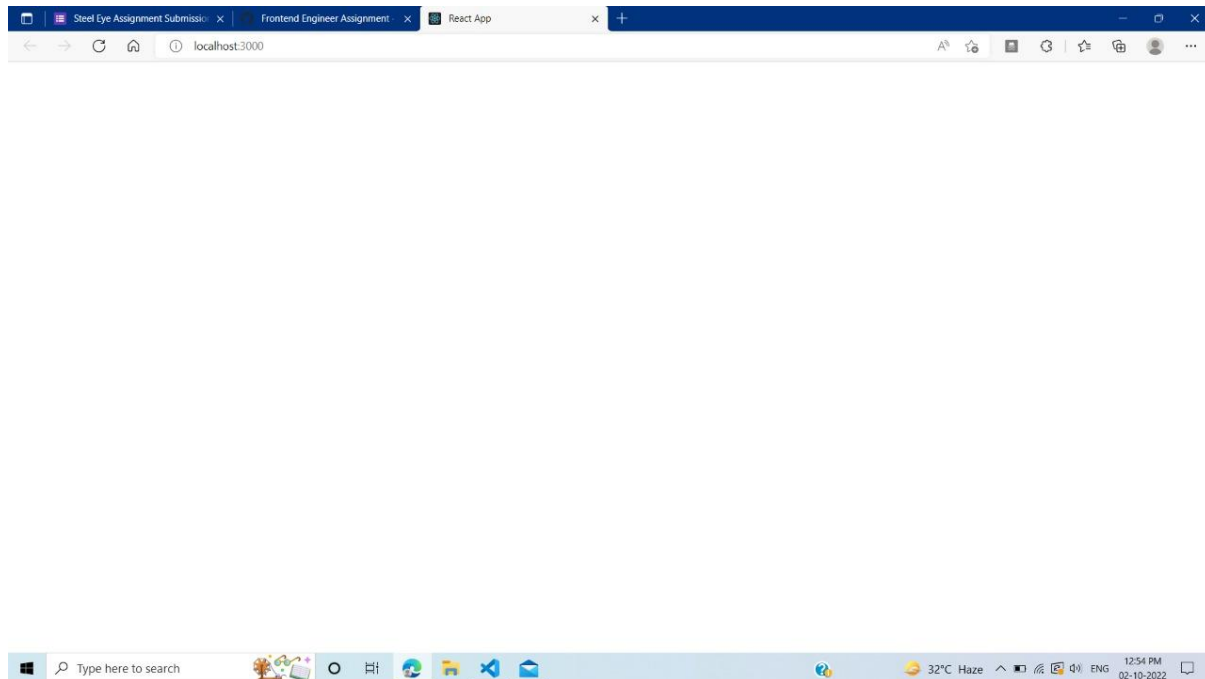
```
const mylist = ['a', 'b', 'c'];

const App = () => (
  <SimpleList list={mylist} />
);

const SimpleList = ({ list }) => (
  <ul>
    {list.map(item => (
      <li key={item}>{item}</li>
    ))}
  </ul>
);
```

That's a simple list component example for React. We only have a list of JavaScript primitives, like strings or integers, map over it, and render a HTML list item for each item in our array.

2. The program runs but shows no output



STEELEYE ASSIGNMENT

3.

```
1 import React, { useState, useEffect, memo } from 'react';
2 import PropTypes from 'prop-types';
3
4 // Single List Item
5 const WrappedSingleListItem = ({
6   index,
7   isSelected,
8   onClickHandler,
9   text,
10 }) => {
11   return (
12     <li
13       style={{ backgroundColor: isSelected ? 'green' : 'red'}}
14       onClick={onClickHandler(index)}
15     >
16       {text}
17     </li>
18   );
19 };
20
21 WrappedSingleListItem.propTypes = {
22   index: PropTypes.number,
23   isSelected: PropTypes.bool,
24   onClickHandler: PropTypes.func.isRequired,
25   text: PropTypes.string.isRequired,
26 };
27
28 const SingleListItem = memo(WrappedSingleListItem);
29
30 // List Component
31 const WrappedListComponent = ({
32   items,
33 }) => {
34   const [selectedIndex, setSelectedIndex] = useState();
35
36   useEffect(() => {
37     setSelectedIndex(null);
38
39     const [selectedIndex, setSelectedIndex] = useState(),
40
41     useEffect(() => {
42       setSelectedIndex(null);
43     }, [items]);
44
45     const handleClick = index => {
46       setSelectedIndex(index);
47     };
48
49     return (
50       <ul style={{ textAlign: 'left' }}>
51         {items.map((item, index) => (
52           <SingleListItem
53             onClickHandler={() => handleClick(index)}
54             text={item.text}
55             index={index}
56             isSelected={selectedIndex}
57           >/>
58         ))}
59       </ul>
60     );
61   });
62
63   WrappedListComponent.propTypes = {
64     items: PropTypes.array(PropTypes.shapeOf({
65       text: PropTypes.string.isRequired,
66     })),
67   };
68
69   WrappedListComponent.defaultProps = {
70     items: null,
71   };
72
73   const List = memo(WrappedListComponent);
74
75   export default List;
```