

Q1) Explain what the simple **List** component does.

Use case of List Component

When creating interfaces, you may need to display multiple instances of the same component with varying data, such as lists of comments or galleries of profile images. The items in these lists are distinguished only by their contents. To achieve this, you can use JavaScript objects and arrays to store the data and utilize methods like `map()` and `filter()` to generate lists of components

Definition of List Component

The List component is commonly used in React to display a list of elements that can be populated with data from a variety of sources, including the parent component, an API, or other data sources.

First Div contains hard coded list item while second contains using list component

```
src > components > JS ListComponent.js > ListComponent
1
2 import React from 'react'
3
4
5 function ListComponent() {
6   const people = [
7     'Creola Katherine Johnson: mathematician',
8     'Mario José Molina-Pasquel Henríquez: chemist',
9     'Mohammad Abdus Salam: physicist',
10    'Percy Lavon Julian: chemist',
11    'Subrahmanyan Chandrasekhar: astrophysicist'
12  ];
13
14  const listItems = people.map(data =>
15    <li><List data={data} /></li>
16  );
17
18
19  return (
20    <div>
21
22      <div> <h4> list without Any loops : </h4>
23      <ul>
24
25        <li>Creola Katherine Johnson: mathematician</li>
26        <li>Mario José Molina-Pasquel Henríquez: chemist</li>
27        <li>Mohammad Abdus Salam: physicist</li>
28        <li>Percy Lavon Julian: chemist</li>
29        <li>Subrahmanyan Chandrasekhar: astrophysicist</li>
30      </ul>
31    </div>
32
33    <div>
```

```
src > components > JS ListComponent.js > ListComponent
28    <li>Percy Lavon Julian: chemist</li>
29    </ul>
30  </div>
31
32  <div>
33
34    <h4> List using List Component with loops </h4>
35
36    {listItems}
37  </div>
38
39 </div>
40
41 }
42
43 const List = ({ data }) => {
44
45   return (
46     <div>
47       <li> {data} </li>
48     </div>
49   )
50 }
51
52 export default ListComponent
```

Q2) What problems / warnings are there with code?

Solution)

First Error that comes immediately after when program is compiled is :

Compiled with problems:

ERROR

```
prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
TypeError: prop_types__WEBPACK_IMPORTED_MODULE_2___default(...).shapeOf is not a function
    at ./src/components/wrappedListComponent.js (http://localhost:3000/static/js/bundle.js:166:116)
    at options.factory (http://localhost:3000/static/js/bundle.js:42376:31)
    at __webpack_require__ (http://localhost:3000/static/js/bundle.js:41800:33)
    at fn (http://localhost:3000/static/js/bundle.js:42033:21)
    at ./src/App.js (http://localhost:3000/static/js/bundle.js:15:90)
    at options.factory (http://localhost:3000/static/js/bundle.js:42376:31)
    at __webpack_require__ (http://localhost:3000/static/js/bundle.js:41800:33)
    at fn (http://localhost:3000/static/js/bundle.js:42033:21)
    at ./src/index.js (http://localhost:3000/static/js/bundle.js:227:62)
    at options.factory (http://localhost:3000/static/js/bundle.js:42376:31)
```

In code it should be around here :

```
57   };
58
59   WrappedListComponent.propTypes = {
60     items: PropTypes.array(PropTypes.shapeOf({
61       text: PropTypes.string.isRequired,
62     })),
63   };
64
65   WrappedListComponent.defaultProps = {
66     items: null,
```

The PropTypes validation for the items prop of WrappedListComponent is incorrect. The array type should be defined as `PropTypes.arrayOf`, and the shape of each item should be defined using `PropTypes.shape`.

Correction :

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

Next Bug :

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

Correction :

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

Next Bug :

OnClickHandler should be passed in prop as reference but it is immediately called ,when the component is rendered .

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

Correction :

```
<SingleListItem
  onClickHandler={handleClick}
  text={item.text}
  index={index}
  isSelected={selectedIndex}
/>
```

Next Bug :

```
✖ ▶ Uncaught TypeError: Cannot read properties of wrappedListComponent.js:46
null (reading 'map')
```

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

Correction :

```
WrappedListComponent.defaultProps = {  
  items: [],  
};
```

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

Modifications :

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

The setSelectedIndex hook is initialized with undefined instead of null.

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

The isSelected prop passed to SingleListItem is now a boolean that checks whether the item's index matches the selectedIndex state value.

```
<SingleListItem  
  onClickHandler={handleClick}  
  text={item.text}  
  index={index}  
  key={index}  
  isSelected={selectedIndex===index}  
>
```

Added key prop to SingleListItem component to prevent React warnings when rendering a list of items.

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)}
    >
      {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(undefined);

  useEffect(() => {
    setSelectedIndex(null);
  });
```

```
    }, [items]));

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

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

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

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

  WrappedListComponent.defaultProps = {
    items: [],
  };

  const List = memo(WrappedListComponent);

  export default List;
```

