

## 1<sup>st</sup> Answer

Lists are very useful when it comes to developing the UI of any website. We can create lists in React in a similar manner as we do in regular JavaScript. Lists are mainly used for displaying menus in a website, for example, the navbar menu. List items in our unordered list should have a unique key. A “key” is a special string attribute you need to include when creating lists of elements in React.

```
import React from 'react';
import ReactDOM from 'react-dom';

function Navmenu(props)
{
  const list = props.menuitems;

  const updatedList = list.map((listItems)=>{
    return(
      <li key={listItems.toString()}>
        {listItems}
      </li>
    );
  });

  return(
    <ul>{updatedList}</ul>
  );
}

const menuItems = [1,2,3,4,5];

ReactDOM.render(
  <Navmenu menuitems = {menuItems} />,
  document.getElementById('root')
);
```

## 2<sup>nd</sup> Answer

React Hook `useEffect` has a missing dependency: `'setSelectedIndex'`. Either include it or remove the dependency array

## 3<sup>rd</sup> answer

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

// Single List Item
const WrappedSingleListItem = ({ index, isSelected, onClickHandler, text }: any) => {
  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 }: any) => {
  const [selectedIndex, setSelectedIndex, ] = useState();

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

  const handleClick = (index: number | null | undefined) => {
    setSelectedIndex(index);
  };

  return (
    <ul style={{ textAlign: 'left' }}>
      {items.map((item: { text: string; }, index: number | null | undefined) => (
        <SingleListItem onClickHandler={() => handleClick(index)} text={item.text} index=
        {index} isSelected={selectedIndex} />
      ))}
    </ul>
  );
};

WrappedListComponent.propTypes = {
  items: PropTypes.array(
    PropTypes.shapeOf({
      text: PropTypes.string.isRequired
    })
  )
};

WrappedListComponent.defaultProps = {
  items: null
};

const List = memo(WrappedListComponent);

export default List;
```