Question 1

(Answer)

Lists are used to display data in an ordered format and mainly used to display menus on websites. In React, Lists can be created in a similar way as we create lists in JavaScript. It is helpful in developing the user interface of any website. List items in unordered List should have a unique key. A “key” is a special attribute we need to include when creating lists of elements in React.

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

function ViewList(props) {
  const funList = props.funList;
  const itemsList = funList.map((dumList) =>
    <li>{dumList}</li>
  );
  return (
    <div>
        <h2>Rendering Lists inside component</h2>
              <ul>{itemsList}</ul>
    </div>
  );
}
const funList = ['Rahul', 'Rohit', 'Nikhil', 'Jass', 'Vipul'];
ReactDOM.render(
  <ViewList funList={funList} />,
  document.getElementById('app')
);
export default App;
```

Question 2

(Answer)

There is a missing dependency setSelectedIndex in React Hook useEffect. It should either be included or remove the dependency array.

Question 3.

(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;
```