1. Explain what the simple List component does ?

Ans:-

The simple List component is a React component that displays a list of items with the ability to select one item from the list. It uses the state hook to keep track of the selected index and the useEffect hook to reset the selected index when the items prop changes.

2.There are a few problems with the code:-

1. The setSelectedIndex state hook should be initialized with an initial value to prevent it from being undefined. For example,` useState(null)`.

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

2. In the SingleListItem component, the onClickHandler function should be wrapped in an arrow function to prevent it from being immediately invoked. For example, `onClick={() => onClickHandler(index)}`.

```
// Single List Item
const SingleListItem = memo(({ index, isSelected, onClickHandler, text }) => {
  return (
    <li
      style={{ backgroundColor: isSelected ? "green" : "red" }}
      onClick={() => onClickHandler(index)}
    >
      {text}
    </li>
  );
});
```

3. The isSelected prop passed to the SingleListItem component should be a boolean indicating whether the current item is selected or not. For example, `isSelected={selectedIndex === index}`.

```
SingleListItem.propTypes = {
  index: PropTypes.number.isRequired,
  isSelected: PropTypes.bool.isRequired,
  onClickHandler: PropTypes.func.isRequired,
  text: PropTypes.string.isRequired,
};
```

4. In the WrappedListComponent propTypes, `items` should be defined as an array of objects, so it should be items: `PropTypes.arrayOf(PropTypes.shape({ text: PropTypes.string.isRequired }))`.

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

Overall

In this modified code, the `setSelectedIndex` state hook is initialized with an initial value of `null`. The onClickHandler function in the SingleListItem component is wrapped in an arrow function to prevent it from being immediately invoked. The isSelected prop passed to the SingleListItem component is now a boolean indicating whether the current item is selected or not. Finally, the items prop is defined as an array of objects in the propTypes, and a key prop is added to each SingleListItem component to prevent React from giving a warning.

Given 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 [setSelectedIndex, selectedIndex] = useState();

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

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

  return (
    <ul style={{ textAlign: "left" }}>
      {items.map((item, index) => (
        <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;
```

After Some Change

```
import React, { useState, useEffect, memo } from "react";
```

```jsx
import PropTypes from "prop-types";

// Single List Item
const SingleListItem = memo(({ index, isSelected, onClickHandler, text }) => {
  return (
    <li
      style={{ backgroundColor: isSelected ? "green" : "red" }}
      onClick={() => onClickHandler(index)}
    >
      {text}
    </li>
  );
});

SingleListItem.propTypes = {
  index: PropTypes.number.isRequired,
  isSelected: PropTypes.bool.isRequired,
  onClickHandler: PropTypes.func.isRequired,
  text: PropTypes.string.isRequired,
};

// List Component
const List = memo(({ items }) => {
  const [selectedIndex, setSelectedIndex] = useState(null);

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

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

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

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

export default List;
```