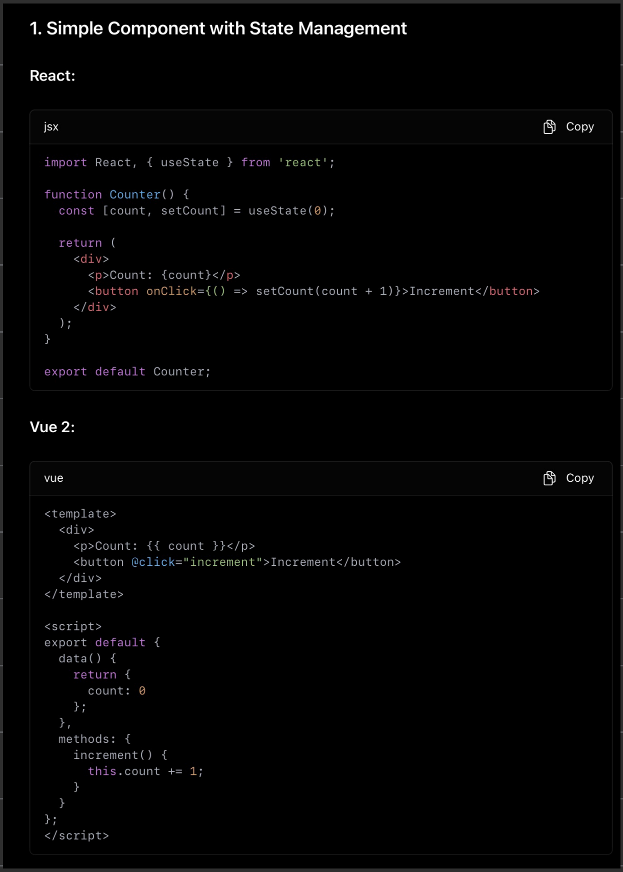
In Vue.js, the equivalent of 'usestate' in React is using the data option within a Vue component to manage local state.

1. Simple component with State Management



1. Simple Component with State Management

React:

jsx Copy

```jsx
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}

export default Counter;
```

Vue 2:

vue Copy

```vue
<template>
  <div>
    <p>Count: {{ count }}</p>
    <button @click="increment">Increment</button>
  </div>
</template>

<script>
export default {
  data() {
    return {
      count: 0
    };
  },
  methods: {
    increment() {
      this.count += 1;
    }
  }
};
</script>
```

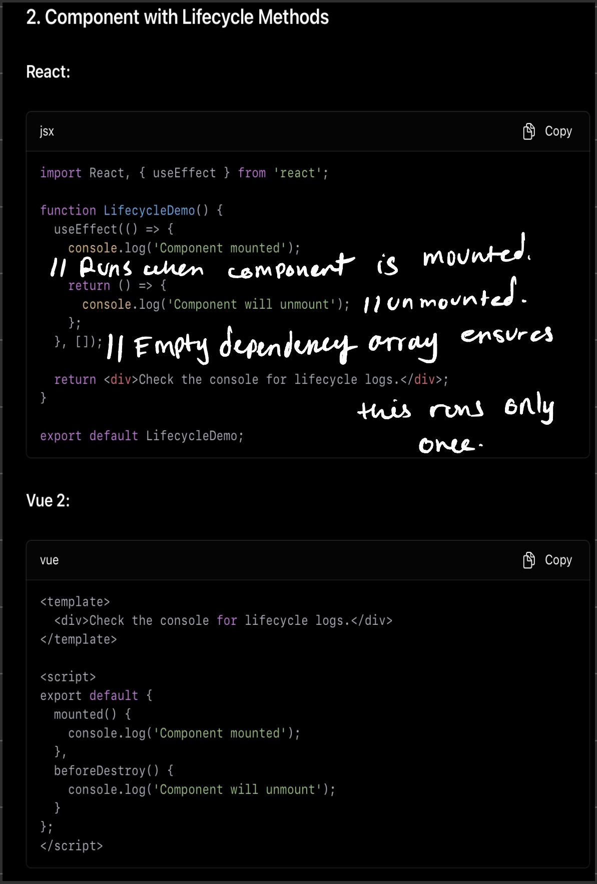
Explanation :-

'usestate' is a React Hook that adds state to Functional components

'count' is the state variable, and setcount is the function to update it.

clicking the button increments the count by calling 'setcount (count + 1)'

2. Component with Lifecycle Methods



2. Component with Lifecycle Methods

React:

```jsx
import React, { useEffect } from 'react';

function LifecycleDemo() {
  useEffect(() => {
    console.log('Component mounted');
// Runs when component is mounted.
    return () => {
      console.log('Component will unmount'); //unmounted.
    };
  }, []); // Empty dependency array ensures
                                    this runs only
                                        once.
  return <div>Check the console for lifecycle logs.</div>;
}

export default LifecycleDemo;
```

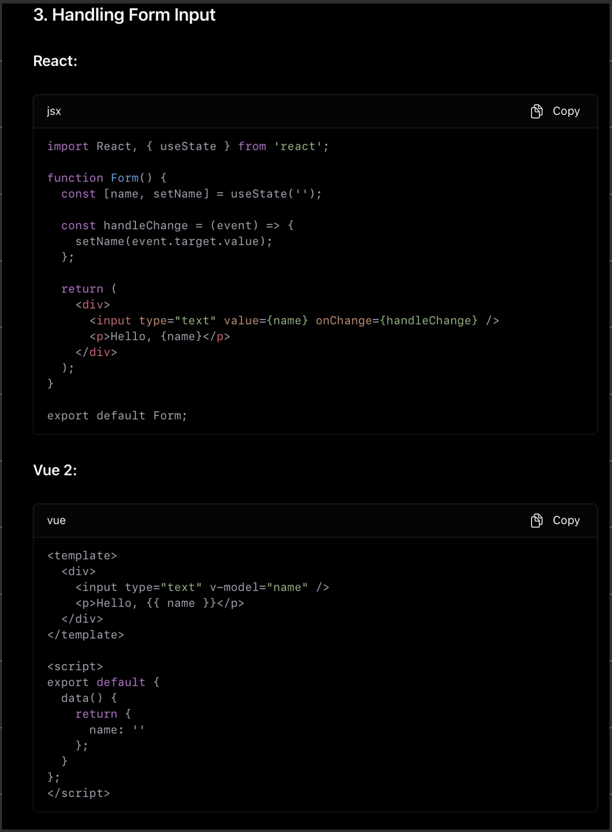
Vue 2:

```vue
<template>
  <div>Check the console for lifecycle logs.</div>
</template>

<script>
export default {
  mounted() {
    console.log('Component mounted');
  },
  beforeDestroy() {
    console.log('Component will unmount');
  }
};
</script>
```

'useEffect' is a hook for managing side effects, similar to lifecycle methods.

3. Handling Form input



3. Handling Form Input

React:

jsx

```jsx
import React, { useState } from 'react';

function Form() {
  const [name, setName] = useState('');

  const handleChange = (event) => {
    setName(event.target.value);
  };

  return (
    <div>
      <input type="text" value={name} onChange={handleChange} />
      <p>Hello, {name}</p>
    </div>
  );
}

export default Form;
```

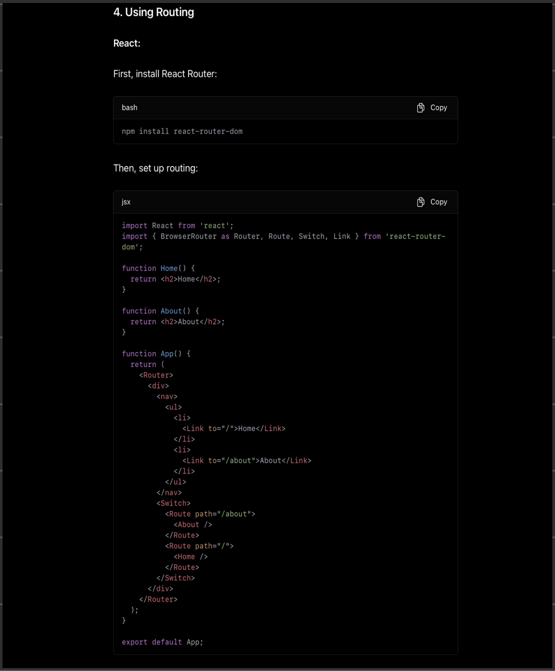
Vue 2:

vue

```vue
<template>
  <div>
    <input type="text" v-model="name" />
    <p>Hello, {{ name }}</p>
  </div>
</template>

<script>
export default {
  data() {
    return {
      name: ''
    };
  }
};
</script>
```

4. Using Routing



4. Using Routing

React:

First, install React Router:

```bash
npm install react-router-dom
```

Then, set up routing:

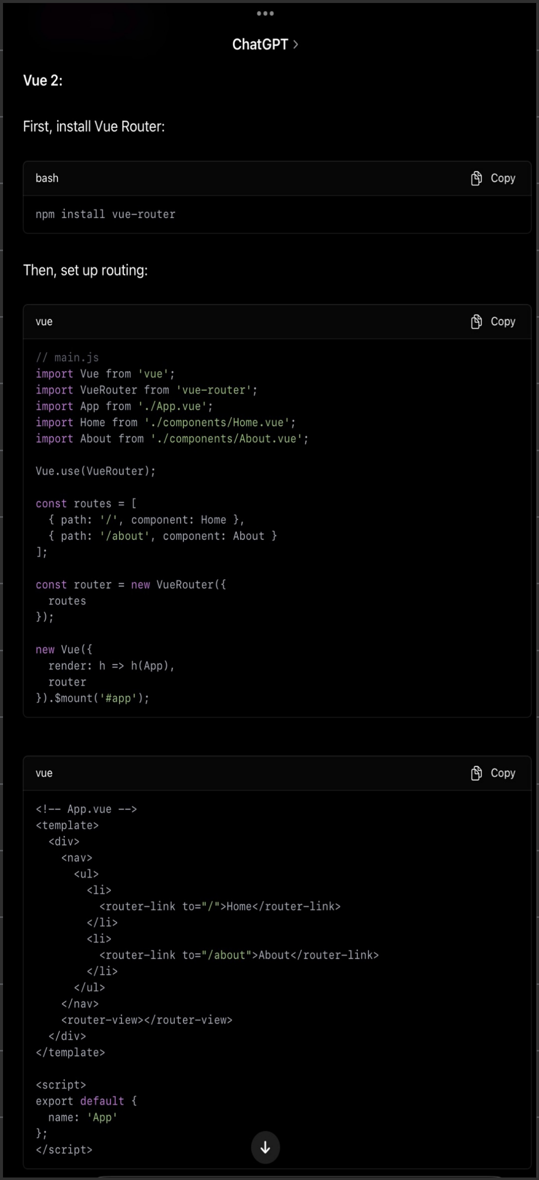
```jsx
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';

function Home() {
  return <h2>Home</h2>;
}

function About() {
  return <h2>About</h2>;
}

function App() {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li>
              <Link to="/">Home</Link>
            </li>
            <li>
              <Link to="/about">About</Link>
            </li>
          </ul>
        </nav>
        <Switch>
          <Route path="/about">
            <About />
          </Route>
          <Route path="/">
            <Home />
          </Route>
        </Switch>
      </div>
    </Router>
  );
}

export default App;
```



Vue 2:

First, install Vue Router:

```bash
npm install vue-router
```

Then, set up routing:

```vue
// main.js
import Vue from 'vue';
import VueRouter from 'vue-router';
import App from './App.vue';
import Home from './components/Home.vue';
import About from './components/About.vue';

Vue.use(VueRouter);

const routes = [
  { path: '/', component: Home },
  { path: '/about', component: About }
];

const router = new VueRouter({
  routes
});

new Vue({
  render: h => h(App),
  router
}).$mount('#app');
```

```vue
<!-- App.vue -->
<template>
  <div>
    <nav>
      <ul>
        <li>
          <router-link to="/">Home</router-link>
        </li>
        <li>
          <router-link to="/about">About</router-link>
        </li>
      </ul>
    </nav>
    <router-view></router-view>
  </div>
</template>

<script>
export default {
  name: 'App'
};
</script>
```