

1 Understand the lifecycle of a React component.

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

```
const SimpleLifecycleDemo = () => {  
  const [counter, setCounter] = useState(0);
```

```
  // Equivalent to componentDidMount  
  useEffect(() => {  
    console.log('Component mounted');
```

```
  // Cleanup function, equivalent to componentWillUnmount  
  return () => {  
    console.log('Component will unmount');
```

```
  }, []); // Empty dependency array ensures this runs only once
```

```
  // Equivalent to componentDidUpdate  
  useEffect(() => {  
    console.log('Component did update');
```

```
  // No need for cleanup in this case, but you can return a function to clean up  
  }, [counter]); // Runs whenever 'counter' changes
```

```
  const increment = () => {  
    setCounter(counter + 1);  
  };
```

```
  return (  
    <div>  
      <p>Counter: {counter}</p>  
      <button onClick={increment}>Increment</button>  
    </div>  
  );  
};
```

```
export default SimpleLifecycleDemo;
```

2 Handle events using the React event system.

```
import React from 'react';

function List(){
  function handleClick (event) {
    // Check the target element to determine which button was clicked
    if (event.target.tagName === 'LI' | event.target.tagName === 'BUTTON') {
      alert( event.target.innerText + ' is Selected!' );
    }
  }

  return (
    <div>
      <ul onClick = {handleClick}>
        <li>Full stack Course</li>
        <li>UI/UX</li>
        <li>Data Science</li>
      </ul>

      <div onClick={handleClick}>
        <button >FYCS</button>
        <button>SYCS</button>
        <button>TYCS</button>
      </div>
    </div>
  );
}

export default List;
```

Routing with React Router:

1. Install and set up React Router for navigation in single-page applications.
2. Create dynamic routes using route parameters.

\$ npm install react-router-dom

1. Install and set up React Router for navigation in single-page applications.

Step-1: In our project, we will create two more components along with **App.js**, which is already present.

About.js

```
import React from 'react'
class About extends React.Component {
  render() {
    return <h1>About</h1>
  }
}
export default About
```

Contact.js

```
import React from 'react'
class Contact extends React.Component {
  render() {
    return <h1>Contact</h1>
  }
}
export default Contact
```

App.js

```
import React from 'react'
class App extends React.Component {
  render() {
    return (
      <div>
        <h1>Home</h1>
      </div>
    )
  }
}
export default App
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Route, Link, BrowserRouter as Router } from 'react-router-dom'
import './index.css';
import App from './App';
import About from './about'
import Contact from './contact'

const routing = (
```

```

<Router>
  <div>
    <h1>React Router Example</h1>
    <ul>
      <li>
        <Link to="/">Home</Link>
      </li>
      <li>
        <Link to="/about">About</Link>
      </li>
      <li>
        <Link to="/contact">Contact</Link>
      </li>
    </ul>
    <Route exact path="/" component={App} />
    <Route path="/about" component={About} />
    <Route path="/contact" component={Contact} />
  </div>
</Router>
)
ReactDOM.render(routing, document.getElementById('root'));

```

2. Create dynamic routes using route parameters.

```

import React from 'react';
import ReactDOM from 'react-dom';
import { BrowserRouter as Router, Route, Link, NavLink } from 'react-router-dom'
import './index.css';
import App from './App';
import About from './about'
import Contact from './contact'

const routing = (
  <Router>
    <div>
      <h1>React Router Example</h1>
      <ul>
        <li>
          <NavLink to="/" exact activeStyle={
            {color:'red'}
          }>Home</NavLink>
        </li>

```

```
    <li>
      <NavLink to="/about" exact activeStyle={
        {color:'green'}}
      >About</NavLink>
    </li>
    <li>
      <NavLink to="/contact" exact activeStyle={
        {color:'magenta'}}
      >Contact</NavLink>
    </li>
  </ul>
  <Route exact path="/" component={App} />
  <Route path="/about" component={About} />
  <Route path="/contact" component={Contact} />
</div>
</Router>
)
ReactDOM.render(routing, document.getElementById('root'));
```