

## REACT

1.

```
import React from 'react';
```

```
const SimpleButton = () => {  
  return (  
    <button>  
      Click me!  
    </button>  
  );  
}
```

```
export default SimpleButton;
```

2.

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

```
const Counter = () => {  
  const [count, setCount] = useState(0);  
  
  const increment = () => {  
    setCount(prevCount => prevCount + 1);  
  };  
  
  const decrement = () => {  
    setCount(prevCount => prevCount - 1);  
  };  
  
  return (  
    <div>  
      <h2>Counter: {count}</h2>  
      <button onClick={increment}>Increment</button>  
      <button onClick={decrement}>Decrement</button>  
    </div>  
  );  
}
```

```
export default Counter;
```

3.

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

```
const ConditionalComponent = () => {  
  const [isLoggedIn, setIsLoggedIn] = useState(false);  
  
  const handleLogin = () => {  
    setIsLoggedIn(true);  
  };  
  
  const handleLogout = () => {  
    setIsLoggedIn(false);  
  };  
  
  return (  
    <div>  
      {isLoggedIn ? (  
        <div>  
          <p>Welcome, User!</p>  
          <button onClick={handleLogout}>Logout</button>  
        </div>  
      ) : (  
        <p>Please login</p>  
        <button onClick={handleLogin}>Login</button>  
      )  
    </div>  
  );  
}
```

```

    </div>
  ): (
    <div>
      <p>Please log in to continue</p>
      <button onClick={handleLogin}>Login</button>
    </div>
  )}
</div>
);
};

```

export default ConditionalComponent;

4.

import React, { useState } from 'react';

const FormComponent = () => {

```

  const [formData, setFormData] = useState({
    name: "",
    email: "",
    age: "",
  });

```

```

  const handleInputChange = (event) => {
    const { name, value } = event.target;
    setFormData({
      ...formData,
      [name]: value,
    });
  };

```

```

  const handleSubmit = (event) => {
    event.preventDefault();
    // You can perform form validation here
    // For simplicity, let's just log the form data
    console.log(formData);
  };

```

```

  return (
    <div>
      <h2>Form</h2>
      <form onSubmit={handleSubmit}>
        <label>
          Name:
          <input
            type="text"
            name="name"
            value={formData.name}
            onChange={handleInputChange}
          />
        </label>
        <br />
        <label>
          Email:
          <input
            type="email"
            name="email"

```

```

        value={formData.email}
        onChange={handleInputChange}
      />
    </label>
    <br />
    <label>
      Age:
      <input
        type="number"
        name="age"
        value={formData.age}
        onChange={handleInputChange}
      />
    </label>
    <br />
    <button type="submit">Submit</button>
  </form>
</div>
);
};

```

```
export default FormComponent;
```

5.

6.

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

```
const SideEffectComponent = () => {
  const [count, setCount] = useState(0);

```

```

  useEffect(() => {
    console.log('Component is mounted or updated');
    (equivalent to componentWillUnmount for unmounting)
    return () => {
      console.log('Component is unmounted');
    };
  });

```

```

const increment = () => {
  setCount(prevCount => prevCount + 1);
};

```

```

const decrement = () => {
  setCount(prevCount => prevCount - 1);
};

```

```

return (
  <div>
    <h2>Count: {count}</h2>
    <button onClick={increment}>Increment</button>
    <button onClick={decrement}>Decrement</button>
  </div>
);
};

```

```
export default SideEffectComponent;
```

7.

8.

9.

10.

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

```
const ListComponent = ({ items }) => {  
  const expensiveCalculation = (item) => {
```

```
    console.log('Calculating...');  
    return item * 2;
```

```
  };
```

```
  const memoizedList = useMemo(() => {  
    console.log('Memoizing...');  
    return items.map(expensiveCalculation);  
  }, [items]);
```

```
  return (  
    <div>  
      <h2>List of Items</h2>  
      <ul>  
        {memoizedList.map((item, index) => (  
          <li key={index}>{item}</li>  
        ))}  
      </ul>  
    </div>  
  );  
};
```

```
export default ListComponent;
```

```
// App.js
```

```
import React from 'react';  
import { AppProvider } from './context';  
import Counter from './Counter';
```

```
const App = () => {  
  return (  
    <AppProvider>  
      <div>  
        <h1>State Management with Context API</h1>  
        <Counter />  
      </div>  
    </AppProvider>  
  );  
};
```

```
export default App;
```

11.

```
import React, { Component } from 'react';

class ErrorBoundary extends Component {
  constructor(props) {
    super(props);
    this.state = { hasError: false };
  }

  static getDerivedStateFromError(error) {

    return { hasError: true };
  }

  componentDidCatch(error, errorInfo) {

    console.error('Error Boundary caught an error:', error, errorInfo);
  }

  render() {
    if (this.state.hasError) {

      return (
        <div>
          <h2>Something went wrong!</h2>
          <p>Please try refreshing the page or contact support.</p>
        </div>
      )
    }

    return this.props.children;
  }
}

export default ErrorBoundary;

import React from 'react';
import ErrorBoundary from './ErrorBoundary';
import ComponentThatMightError from './ComponentThatMightError';

const App = () => {
  return (
    <div>
      <h1>My App</h1>
      <ErrorBoundary>
        <ComponentThatMightError />
      </ErrorBoundary>
    </div>
  );
};

export default App;
```

12.

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

```

import Login from './Login';
import Dashboard from './Dashboard';
import PrivateRoute from './PrivateRoute';

const App = () => {
  const [isAuthenticated, setIsAuthenticated] = useState(false);

  const login = () => {
    setIsAuthenticated(true);
  };

  const logout = () => {
    setIsAuthenticated(false)
  };

  return (
    <Router>
      <Switch>
        <Route path="/login">
          <Login login={login} isAuthenticated={isAuthenticated} />
        </Route>
        <PrivateRoute path="/dashboard" isAuthenticated={isAuthenticated}>
          <Dashboard logout={logout} />
        </PrivateRoute>
        <Redirect exact from="/" to="/dashboard" />
      </Switch>
    </Router>
  );
};

export default App;

```

14.

15.