**Lesson 1 Summary: Building Rich Front-End Applications with React and ES6**

Congratulations! You have completed this lesson. At this point in the course, you know that:

* Libraries offer reusable code components, while frameworks provide a comprehensive structure for building applications.
* React features include component-based architecture, declarative syntax, virtual DOM optimization, one-way data binding, JSX integration, and React Hooks employment.
* You can create React projects using Create React App (CRA) and use Vite as an alternative tool. Vite optimizes web project development by leveraging modern browser capabilities.
* New features introduced in JavaScript as a part of ES6 are let, const, arrow functions, promise, and class.
* JSX is an extension to JavaScript that transforms HTML-like text found in JavaScript files into standard JavaScript objects.

**Lesson 2 Summary: Introduction to Class Components**

* React components enable you to break up the user interface into separate parts that can be reused and handled independently.
* The four types of React components are functional, class and high-order components.
* A class component in React is a JavaScript class that extends the React Component class from the React library.
* Class components manage state, handle lifecycle events, and define component methods.
* In React, props, or properties, send data from a parent component to a child component.
* Event handling is the process of responding to user interactions, such as clicks, mouseovers, mouseouts, form submissions, and so on, within a React application.
* State is a plain JavaScript object that React uses to represent information about the component’s current situation.
* Any changes in the state data cause the re-rendering of the component to reflect the change.
* Components can create and manage their data with states, whereas they receive data from outside with props.
* The lifecycle of each React component encompasses three key phases: Mounting, updating, and unmounting.
* In component lifecycle, components are created or mounted on the DOM, they grow by updating, and then die or are unmounted on the DOM.