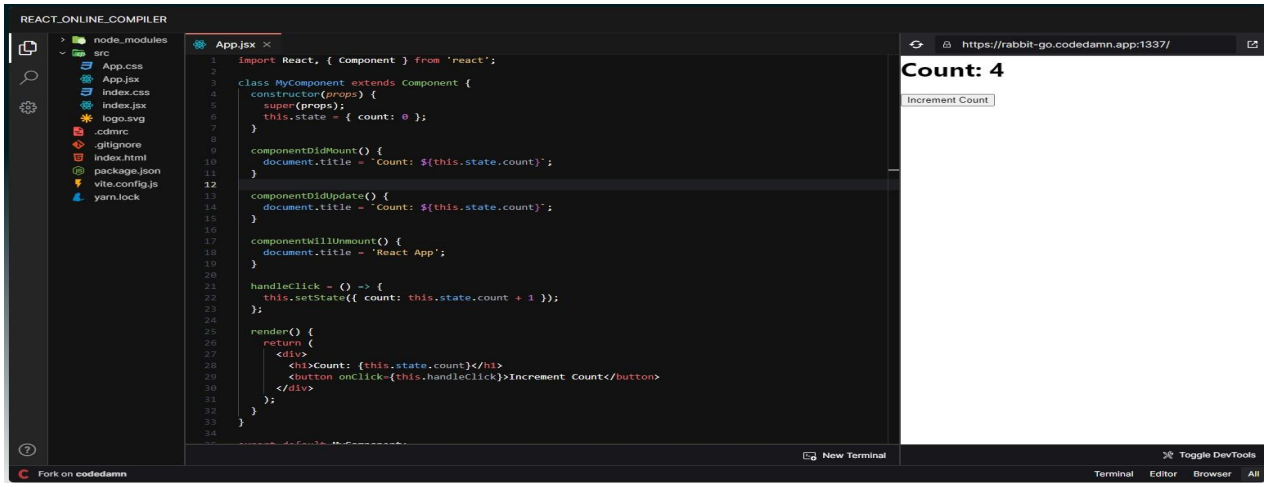
Const Assignment = (Jainam) => { React "Life cycle & Hooks" }

There are two types of components in React == Class components & Functional components. Both types of components have a lifecycle, which is a series of methods that are called in a specific order during the component's lifetime;

{/* Class Component Lifecycle: */}

Mounting == When an instance of a component is created and inserted into the DOM, called mounting;

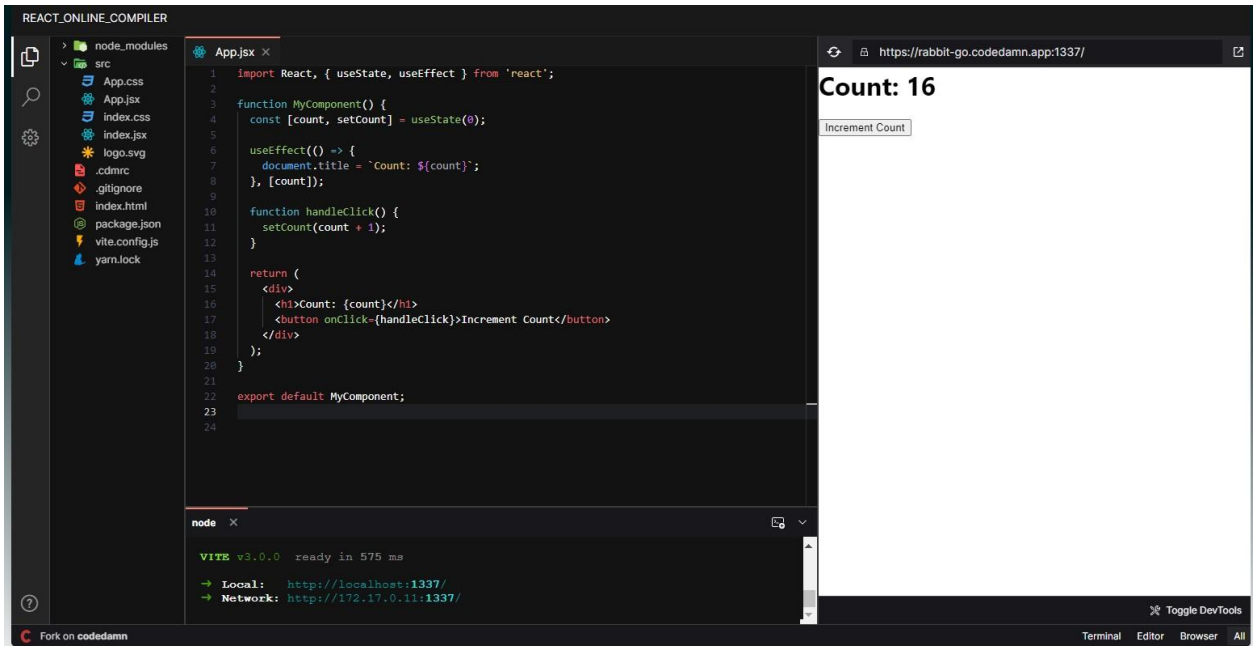
- ➔ {/* constructor (props) */} *== This method is called first, and is used to initialize the component's state and bind event handlers.*
- ➔ {/* render( ) */} *== This method is required and returns the JSX code that represents the component's UI.*
- ➔ {/* componentDidMount( ) */} *== This method is called after the component is rendered for the first time, and is used for side-effects like fetching data from an API.*



The componentDidMount method sets the document title to the current count. The componentDidUpdate method updates the document title every time the count changes. The componentWillUnmount method sets the document title back to the default when the component is unmounted.

{/* Functional Component Lifecycle with Hooks: */}

React Hooks provide a way to useState and other React features in Functional components. The most commonly used Hooks for lifecycle methods are useEffect and useState.



The useEffect Hook is used to update the document title whenever the state changes. The function that we put will update the state when the button is clicked. The dependency array passed as the second argument to useEffect tells React to only call the effect function when the state changes.

The lifecycle methods are used to manage the component's state and to interact with the DOM or external APIs. Whether you're using a Class or a Functional component with Hooks, the lifecycle methods follow the same order of execution.

export default THE_END