| **4. ReactJS-HOL** |
| --- |

**1. Need and Benefits of Component Lifecycle**

The Component Lifecycle in React refers to the series of methods that are automatically called at different stages of a component's existence — from creation (mounting), to updates, and finally to removal (unmounting).

**Why is it needed?**

* React applications are dynamic and require managing side effects, such as data fetching, subscriptions, DOM updates, etc.
* Lifecycle methods allow us to hook into the internal phases of a component, enabling controlled and predictable behavior.

**2. Lifecycle Hook Methods**

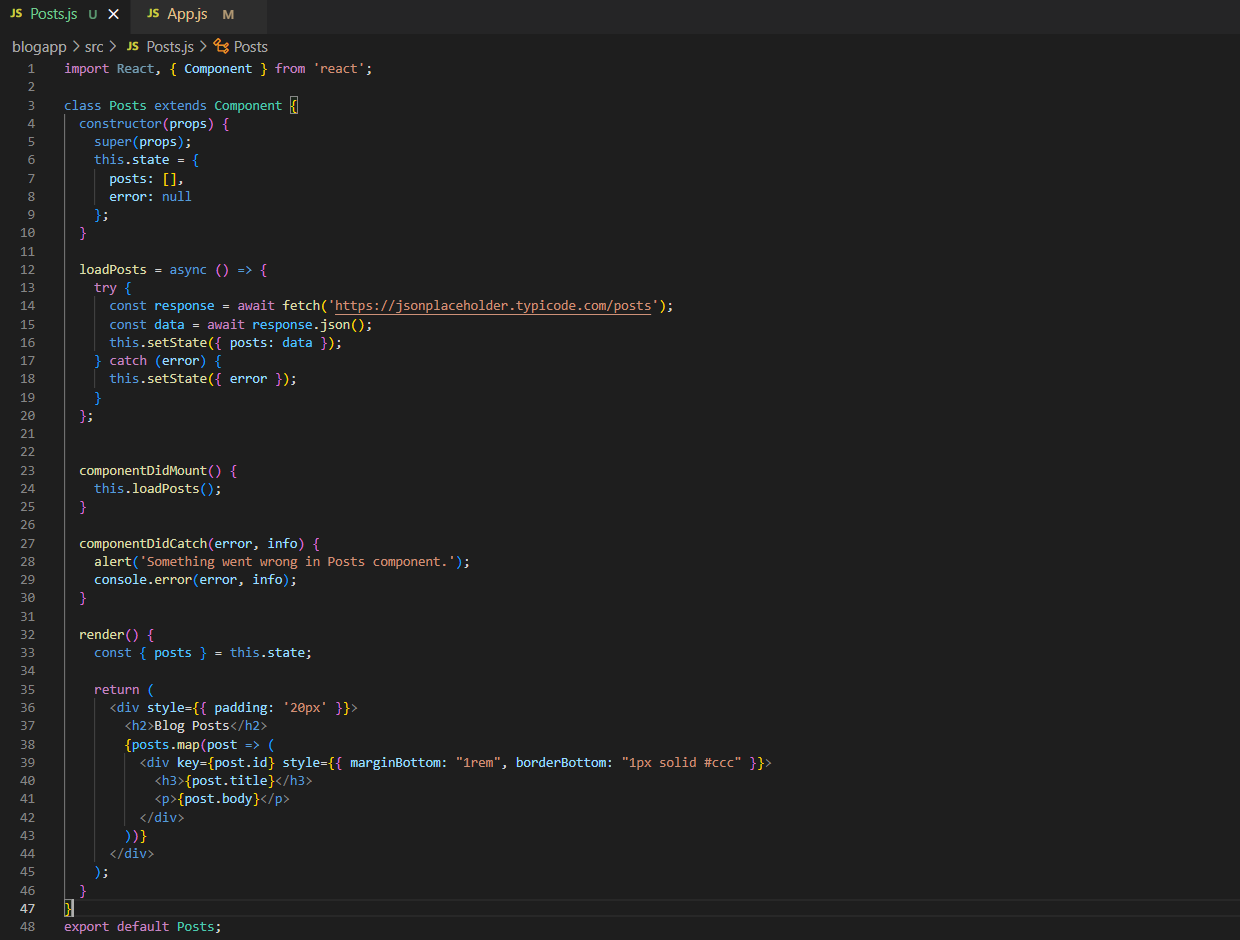
React Class Components provide several built-in lifecycle methods categorized by the phase:

**Mounting Phase (When the component is inserted into the DOM):**

1. **constructor()** – Initializes state and binds methods.
2. **static getDerivedStateFromProps()** – Updates state based on props before rendering.
3. **render()** – Returns the JSX to render the component.
4. **componentDidMount()** – Invoked after the component is mounted; ideal for API calls and side effects.

**Updating Phase (When props/state changes):**

1. **static getDerivedStateFromProps()**
2. **shouldComponentUpdate()** – Determines whether the component should re-render.
3. **render()**
4. **getSnapshotBeforeUpdate()** – Captures some information before the DOM updates.
5. **componentDidUpdate()** – Called after the component updates.



**Unmounting Phase (When the component is removed from DOM):**

* **componentWillUnmount()** – Cleanup tasks such as cancelling network requests, clearing timers, etc.

**Error Handling Phase:**

* **componentDidCatch(error, info)** – Catches and handles errors during rendering or in lifecycle methods.

