# 4. ReactJS HOL

***• Explain the need and benefits of component life cycle***

The component life cycle in React helps developers manage what happens at different stages of a component’s existence—like when it is created, updated, or destroyed. It enables better control over rendering, data fetching, and cleanup, improving performance and reliability. Using life cycle methods, we can trigger actions like fetching data, updating the DOM, or clearing resources efficiently.

***• Identify various life cycle hook methods***

React class components provide several life cycle methods, including:

* constructor() – for initializing state
* componentDidMount() – called after the component mounts
* shouldComponentUpdate() – to control re-rendering
* componentDidUpdate() – called after updates
* componentWillUnmount() – for cleanup before component is removed
* componentDidCatch() – for error handling

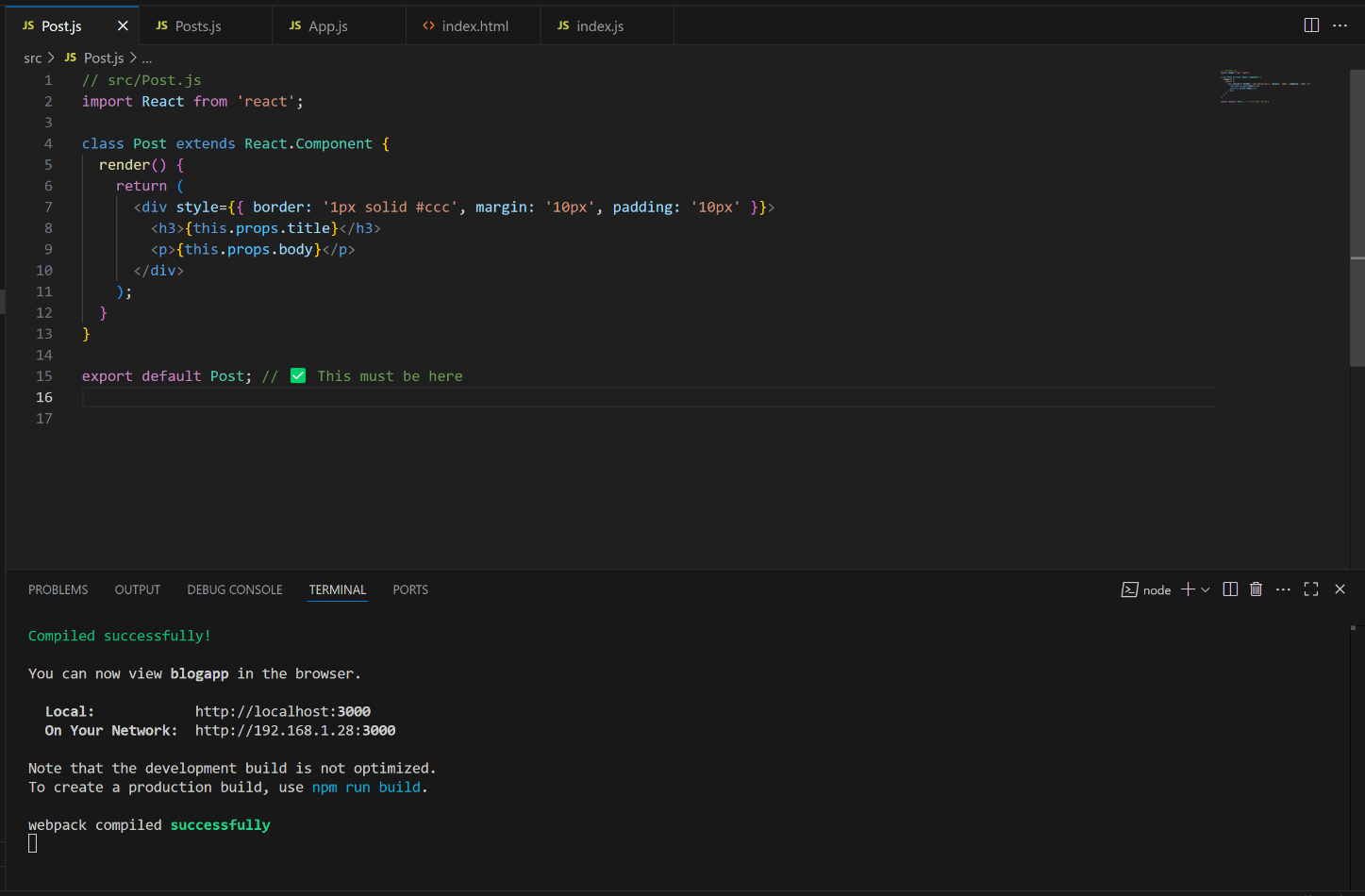
***• List the sequence of steps in rendering a component***

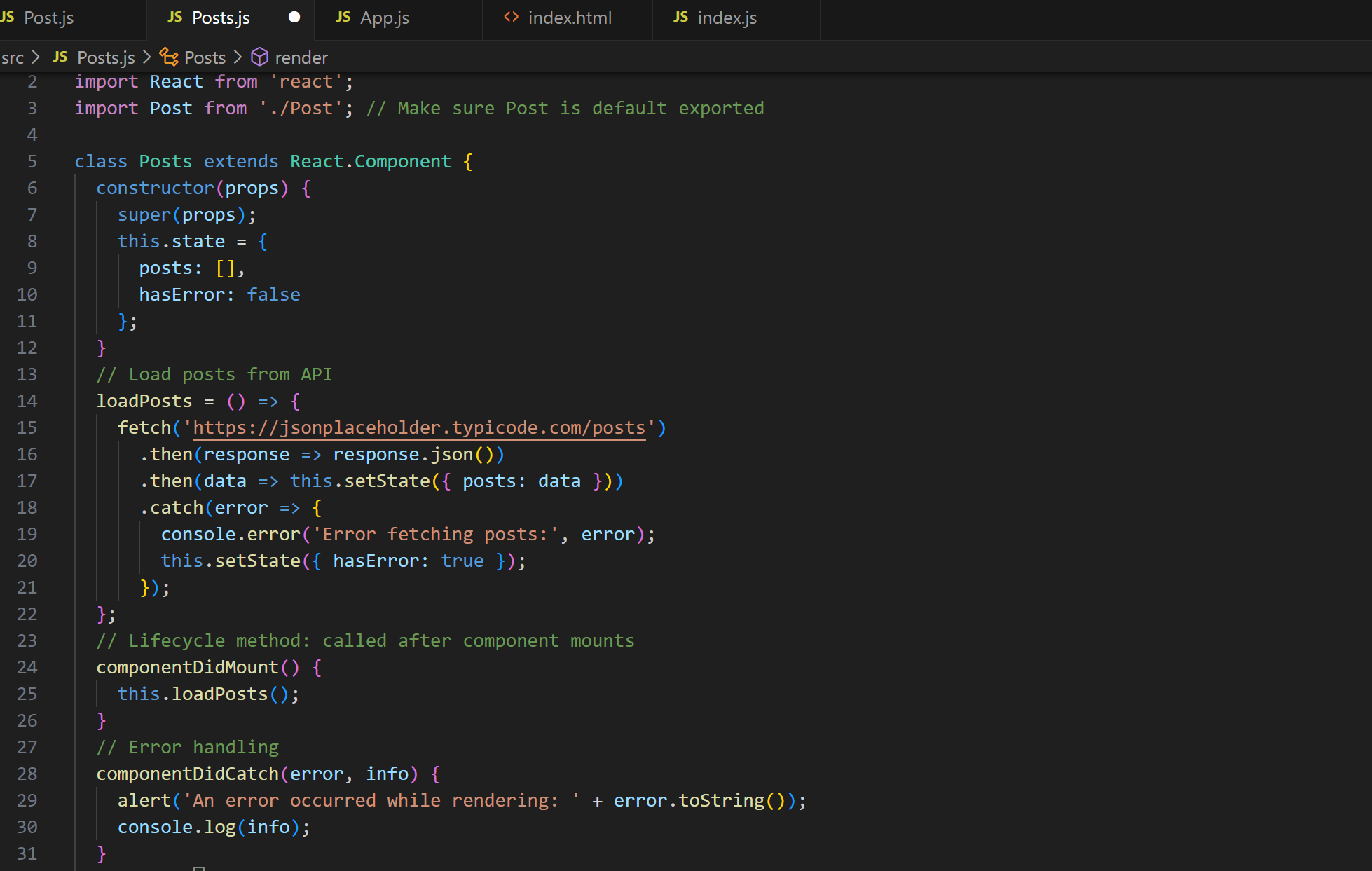
The typical sequence in rendering a React class component is:

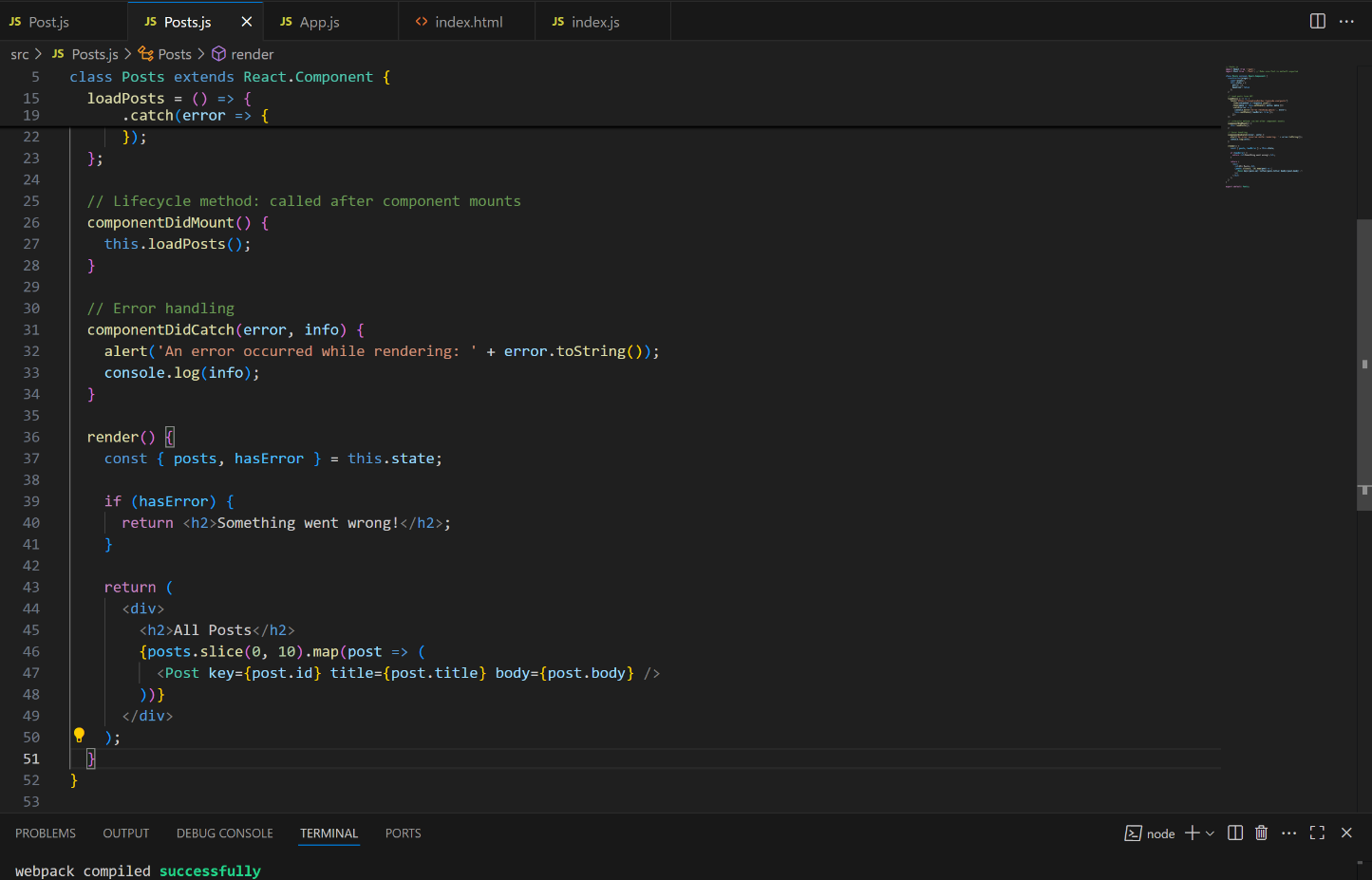
1. **constructor()**
2. **render()**
3. **componentDidMount()**
4. On updates: shouldComponentUpdate() → render() → componentDidUpdate()
5. On unmount: componentWillUnmount()

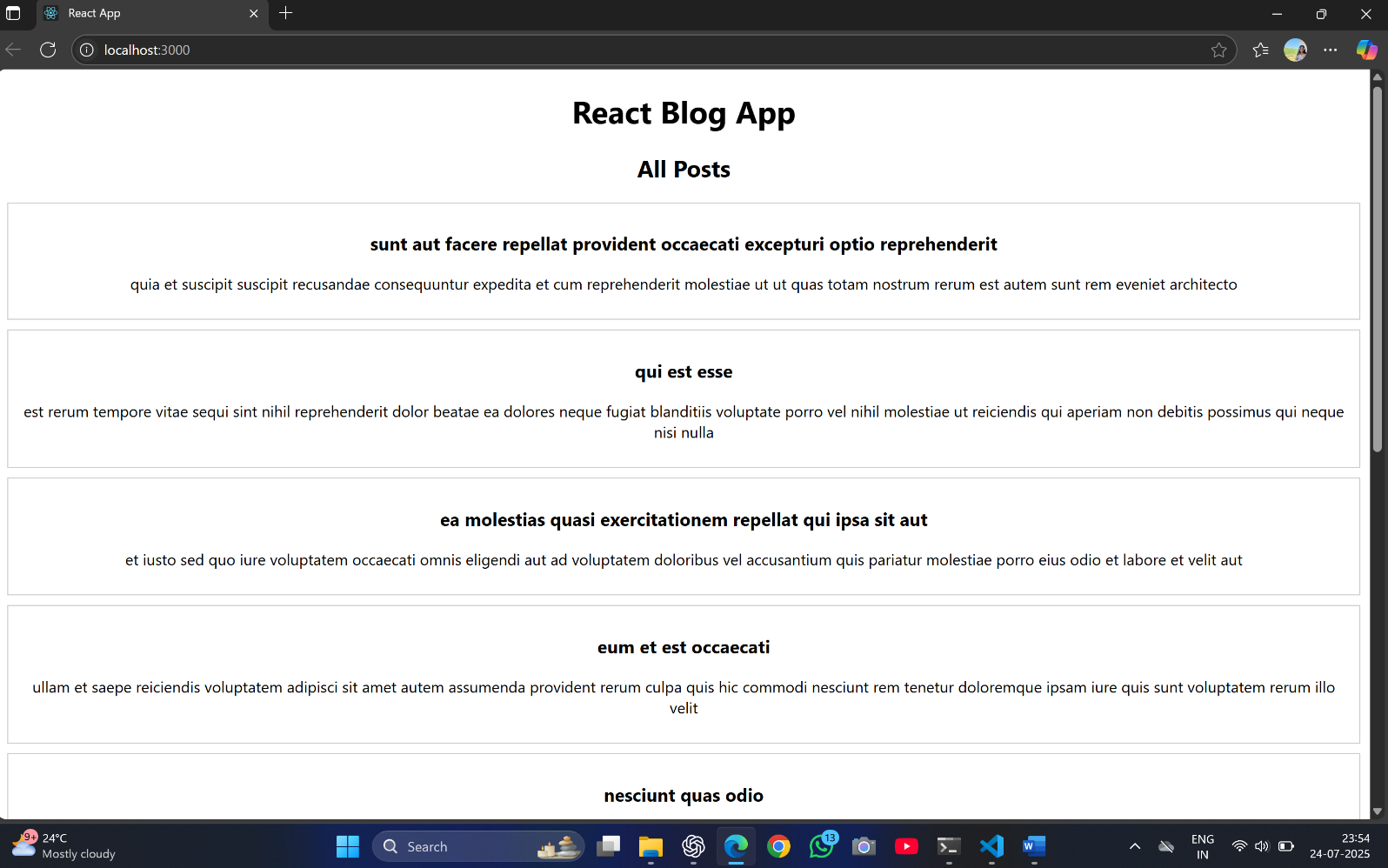
**Create a new react application using *create-react-app* tool with the name as “blogapp”**

**Create a new method in component with the name as loadPosts() which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (**[**https://jsonplaceholder.typicode.com/posts**](https://jsonplaceholder.typicode.com/posts)**)**









--------------------