React Component Lifecycle Lab

# Steps and Code

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

2. Open the application using VS Code

3. Create a new file named as Post.js in src folder with following content:

import React from 'react';  
  
class Post extends React.Component {  
 render() {  
 return (  
 <div>  
 <h2>{this.props.title}</h2>  
 <p>{this.props.body}</p>  
 </div>  
 );  
 }  
}  
  
export default Post;

4. Create a new class based component named as Posts inside Posts.js file:

import React from 'react';  
import Post from './Post';  
  
class Posts extends React.Component {  
 constructor(props) {  
 super(props);  
 this.state = { posts: [], hasError: false };  
 }  
  
 loadPosts() {  
 fetch('https://jsonplaceholder.typicode.com/posts')  
 .then(response => response.json())  
 .then(data => this.setState({ posts: data.slice(0, 10) }))  
 .catch(error => this.setState({ hasError: true }));  
 }  
  
 componentDidMount() {  
 this.loadPosts();  
 }  
  
 componentDidCatch(error, info) {  
 alert('An error occurred: ' + error);  
 }  
  
 render() {  
 return (  
 <div>  
 <h1>Blog Posts</h1>  
 {this.state.posts.map(post => (  
 <Post key={post.id} title={post.title} body={post.body} />  
 ))}  
 </div>  
 );  
 }  
}  
  
export default Posts;

10. Add the Posts component to App component:

import React from 'react';  
import Posts from './Posts';  
  
function App() {  
 return (  
 <div className="App">  
 <Posts />  
 </div>  
 );  
}  
  
export default App;

11. Build and Run the application using npm start command.

# Output

