# React Blog App - Lifecycle Methods Lab

## Objectives

• Explain the need and Benefits of component life cycle  
• Identify various life cycle hook methods  
• List the sequence of steps in rendering a component  
  
In this hands-on lab, you will learn how to:  
• Implement componentDidMount() hook  
• Implement componentDidCatch() life cycle hook.

## Prerequisites

• Node.js  
• NPM  
• Visual Studio Code  
  
Estimated time to complete this lab: 60 minutes.

## Instructions

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

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

5. Initialize the component with a list of Post in state using constructor

6. Create a method loadPosts() that fetches posts from https://jsonplaceholder.typicode.com/posts

7. Call loadPosts() inside componentDidMount()

8. Render the posts using render() with title in heading and body in paragraph

9. Implement componentDidCatch() to catch and alert errors

10. Add the Posts component to App.js

11. Run the application using npm start

## Code

### Post.js

import React from 'react';  
  
function Post({ title, body }) {  
 return (  
 <div style={{ border: '1px solid gray', marginBottom: '1rem', padding: '1rem' }}>  
 <h3>{title}</h3>  
 <p>{body}</p>  
 </div>  
 );  
}  
  
export default Post;

### Posts.js

import React, { Component } from 'react';  
import Post from './Post';  
  
class Posts extends 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, 5) }))  
 .catch(error => {  
 console.error('Error fetching posts:', error);  
 this.setState({ hasError: true });  
 });  
 };  
  
 componentDidMount() {  
 this.loadPosts();  
 }  
  
 componentDidCatch(error, info) {  
 alert("An error occurred: " + error);  
 }  
  
 render() {  
 return (  
 <div>  
 <h2>Blog Posts</h2>  
 {this.state.posts.map(post => (  
 <Post key={post.id} title={post.title} body={post.body} />  
 ))}  
 </div>  
 );  
 }  
}  
  
export default Posts;

### App.js

import React from 'react';  
import Posts from './Posts';  
  
function App() {  
 return (  
 <div className="App">  
 <h1>Welcome to Blog App</h1>  
 <Posts />  
 </div>  
 );  
}  
  
export default App;

## Output Screenshot

