In this hands-on lab, you will learn how to:

* Implement componentDidMount() hook
* Implementing componentDidCatch() life cycle hook.

Post.js

class Post {

  constructor(id, title, body){

    this.id=id;

    this.title=title;

    this.body=body;

  }

}

export default Post;

Posts.js

import Post from "./Post"

import React  from "react";

class Posts extends React.Component {

  constructor(props){

    super(props);

    this.state = {

      posts: [],

      error: null,

    };

    this.isloading = true;

  }

  async loadPosts(){

    const response = await fetch('https://jsonplaceholder.typicode.com/posts');

    console.log(response);

    if (!response.ok) {

      throw new Error('Network response was not ok');

    }

    const data = await response.json();

    // Map JSON to Post instances

    return data.map(item => new Post(item.id, item.title, item.body));

  }

  async componentDidMount(){

    try{

      const posts = await this.loadPosts();

      this.setState({ posts: posts });

      this.isloading=false;

      console.log(posts);

    }catch(error){

      this.componentDidCatch(error, "Failed to fetch posts:");

      this.setState({ error: error.message });

      this.isloading=true;

    }

  }

  render(){

    const { posts, error } = this.state;

    if (error) return <div>Error: {error}</div>;

    if (this.isloading)

      return (

        <div>

          <h2>Posts</h2>

          <p>loading...</p>

        </div>

    );

    return (

      <div>

      <h2>Posts</h2>

      <hr/>

        {posts.map((post) => (

          <div key={post.id} style={{ marginBottom: '20px' }}>

            <h3>{post.title}</h3>

            <p>{post.body}</p>

            <hr />

          </div>

        ))}

      </div>

    );

  }

  componentDidCatch(error, info){

    console.error(info, error);

  }

}

export default Posts;

App.js

import './App.css';

import Posts from './Components/Posts';

function App() {

  return (

    <div className="App">

      <Posts/>

    </div>

  );

}

export default App;

**Output**

 