**REACTJS-04**

**src/App.js**

// src/App.js

import React from 'react';

import Posts from './Posts';

import './App.css';

function App() {

return (

<div className="App">

<header className="App-header">

[cite\_start]{/\* Add the Posts component to the App component [cite: 31] \*/}

<Posts />

</header>

</div>

);

}

export default App;

**src/Post.js**

// src/Post.js

class Post {

constructor(id, title, body) {

this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

**src/Posts.js**

// src/Posts.js

import React from 'react';

class Posts extends React.Component {

constructor(props) {

super(props);

//[cite\_start]// Initialize state with an empty list of posts [cite: 21]

this.state = {

posts: [],

error: null,

};

}

//[cite\_start]// Method to fetch posts from the API [cite: 22]

async loadPosts() {

try {

const response = await fetch('https://jsonplaceholder.typicode.com/posts'); //[cite\_start]// [cite: 23]

if (!response.ok) {

throw new Error('Something went wrong!');

}

const data = await response.json();

this.setState({ posts: data });

} catch (error) {

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

}

}

//[cite\_start]// Call loadPosts() after the component mounts [cite: 25]

componentDidMount() {

this.loadPosts(); //[cite\_start]// [cite: 25]

}

//[cite\_start]// Error boundary to catch errors in child components [cite: 29]

componentDidCatch(error, info) {

alert(`Error: ${error.message}\n\n${info.componentStack}`); //[cite\_start]// [cite: 29]

}

//[cite\_start]// Render the list of posts [cite: 27]

render() {

if (this.state.error) {

return <h1>{this.state.error}</h1>;

}

return (

<div>

<h1>Blog Posts</h1>

{this.state.posts.map(post => (

<div key={post.id}>

<h2>{post.title}</h2>

<p>{post.body}</p>

</div>

))}

</div>

);

}

}

export default Posts;

**OUTPUT**

