**Solution:**

**Post.js:**

class Post {

  constructor(id, title, body) {

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

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: [],

    };

  }

  // Step 6: Fetch posts

  loadPosts() {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then((response) => {

        if (!response.ok) {

          throw new Error('Failed to fetch posts');

        }

        return response.json();

      })

      .then((data) => {

        // Optional: convert to Post class objects

        // const postObjects = data.map(p => new Post(p.id, p.title, p.body));

        this.setState({ posts: data }); // or postObjects

      })

      .catch((error) => {

        throw error; // Will be caught by componentDidCatch

      });

  }

  // Step 7: Call loadPosts after component mounts

  componentDidMount() {

    this.loadPosts();

  }

  // Step 9: Error handling

  componentDidCatch(error, info) {

    alert("An error occurred: " + error.message);

  }

  // Step 8: Render method

  render() {

    return (

      <div>

        <h2>Blog Posts</h2>

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

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

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

OUTPUT:

