import React, {useEffect,useState} from "react";

export function UseEffectExample() {

  const [users, setUsers] = useState([]);

  const [loading, setLoading] = useState(true);

// React hook: runs after the component renders

  useEffect(() => {

  async function fetchUsers() {   // Define an async function to fetch data

    try {

      // Send HTTP request to the given API endpoint, wait for response

      const res = await fetch("https://jsonplaceholder.typicode.com/users");

     // Parse response as JSON, wait until finished, store in `data`

      const data = await res.json();

     // Update component state with the fetched users

      setUsers(data);

    } catch (err) {

      console.error("Error fetching users:", err);

    } finally {

      // No matter success or error: stop the "loading" indicator

      setLoading(false);

    }

  }

 // Call the async function immediately

  fetchUsers();

 // Empty dependency array: run only once, when component mounts

}, []);

  if (loading) return <p>Loading users...</p>;

  return (

    <div>

      <h3>useEffect Example (Data Fetching)</h3>

3      <ul>

        {users.map(user => <li key={user.id}>{user.name}</li>)}

      </ul>

    </div>

  );

}