

# TÀI LIỆU HƯỚNG DẪN LẬP TRÌNH REACT CƠ BẢN (CRUD)

## I. MỤC TIÊU BÀI HỌC

- Hiểu cấu trúc ứng dụng React gồm nhiều Functional Component.
- Sử dụng Hooks (useState, useEffect) để quản lý trạng thái và vòng đời.
- Áp dụng mô hình CRUD (Create, Read, Update, Delete).
- Thực hành Luồng dữ liệu một chiều giữa Component cha và con
- Thiết kế Form thêm/sửa, giao diện thân thiện.

## II. TRÌNH TỰ THỰC HIỆN

### BƯỚC 1. THIẾT LẬP CẤU TRÚC REACT CƠ BẢN

1. Tạo file HTML với khung cơ bản và thẻ `<div id='root'></div>`
2. Thêm các thư viện cần thiết (React, ReactDOM, Babel – JS Compiler) qua CDN

```
//trong head  
  
<script  
src='https://unpkg.com/react@18/umd/react.development.js'></script>  
  
<script src='https://unpkg.com/react-dom@18/umd/react-  
dom.development.js'></script>  
  
<script  
src='https://unpkg.com/@babel/standalone/babel.min.js'></script>
```

3. Trong body, viết mã React trong thẻ `<script type= "text/babel">`
4. Tạo component gốc App và render bằng ReactDOM:

```
function App() {  
  return <div><h1>Quản lý người dùng</h1></div>;  
}  
const root =  
ReactDOM.createRoot(document.getElementById('root'));  
root.render(<App />);
```

**Kết quả:**

## Quản lý người dùng

### BUỚC 2. TỔ CHỨC COMPONENT VÀ STATE TẬP TRUNG

- Ứng dụng gồm các component: App, SearchForm, AddUser, ResultTable.

Component	Vai trò
App	Quản lý toàn bộ state và truyền props xuống con
SearchForm	Nhận input từ người dùng để tìm kiếm
AddUser	Form thêm người dùng mới
ResultTable	Hiển thị, lọc, sửa, xóa người dùng

**Nguyên tắc:** Dữ liệu cần chia sẻ (Từ khóa tìm kiếm, Người dùng mới) phải được lưu trữ ở component cha (App) và truyền xuống qua props. Các hàm cập nhật state được truyền xuống component con thông qua Props (State Lifting).

### 2. Mã minh họa của Component App

```
function App () {  
  const [kw, setKeyword] = React.useState("");  
  const [newUser, setNewUser] = React.useState(null);  
  return (  
    <div>  
      <h1>Quản lý người dùng</h1>  
      <SearchForm onChangeValue={setKeyword} />  
      <AddUser onAdd={setNewUser} />  
      <ResultTable keyword={kw} user={newUser}  
        onAdded={ ()=>setNewUser(null) } />  
    </div>  
  );  
}
```

### BUỚC 3. CHỨC NĂNG TÌM KIẾM (SEARCHFORM)

- Mục tiêu:** Truyền dữ liệu tìm kiếm từ con → cha → bảng kết quả.

**Logic:**

- SearchForm nhận hàm onChangeValue (tức là setKeyword từ App).
- Sử dụng sự kiện onChange để kích hoạt hàm callback, cập nhật kw ở App.

### 2. Mã minh họa Component SearchForm

```

function SearchForm({ onChangeValue }) {
  return (
    <input
      type="text"
      placeholder="Tìm theo name, username"
      onChange={(e) => onChangeValue(e.target.value)}
    />
  );
}

```

Khi người dùng nhập: SearchFrom gọi hàm onChangeValue để App cập nhật kw, sau đó App truyền kw cho ResultTable để lọc lại danh sách qua keyword.

## BƯỚC 4. HIỂN THỊ DANH SÁCH (RESULTTABLE)

1. **Mục tiêu:** Tải dữ liệu từ API, lưu trong state, và render ra bảng.
2. **Logic**

- Tải dữ liệu: Dùng React.useEffect (với dependency array []) để gọi API **một lần** khi component được mount.
- Lọc dữ liệu: Dùng Array.filter() để lọc danh sách users theo keyword nhận được từ props.

### 3. Mô hình họa Component ResultTable

```

function ResultTable({ keyword, user, onAdded }) {
  const [users, setUsers] = React.useState([]);
  const [loading, setLoading] = React.useState(true);

  // Tải dữ liệu 1 lần khi component mount
  React.useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/users")
      .then(res => res.json())
      .then(data => { setUsers(data); setLoading(false); });
  }, []);
}

```

### 4. Lọc danh sách theo keyword

```

const filteredUsers = users.filter(
  (u) =>
    u.name.toLowerCase().includes(keyword.toLowerCase()) ||
    u.username.toLowerCase().includes(keyword.toLowerCase())
);

```

### 5. Hiển thị dữ liệu bằng map trong return

```

<tbody>
  {filteredUsers.map((u) => (

```

```

<tr key={u.id}>
  <td>{u.id}</td>
  <td>{u.name}</td>
  <td>{u.username}</td>
  <td>{u.email}</td>
  <td>
    {u.address.street}, {u.address.suite}, {u.address.city}
  </td>
  <td>{u.phone}</td>
  <td>
    <button onClick={() => editUser(u)}>Sửa</button>
    <button onClick={() => removeUser(u.id)}>Xóa</button>
  </td>
</tr>
) )
</tbody>

```

## 6. Khai báo Component AddUser, sau đó chạy ứng dụng

```
function AddUser({ onAdd }) { }
```

### Quản lý người dùng

Tim theo name, username				
ID Họ tên	Username	Email	Địa chỉ	Số điện thoại
1 Leanne Graham	Bret	Sincere@april.biz	Kulas Light, Apt. 556, Gwenborough	1-770-736-8031 x56442
2 Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains, Suite 879, Wisokyburgh	010-692-6593 x09125
3 Clementine Bauch	Samantha	Nathan@yesenia.net	Douglas Extension, Suite 847, McKenziehaven	1-463-123-4447
4 Patricia Lebsack	Karianne	Julianne.OConner@kory.org	Hoefer Mall, Apt. 692, South Elvis	493-170-9623 x156
5 Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	Skiles Walks, Suite 351, Roscoeview	(254)954-1289
6 Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info	Norberto Crossing, Apt. 950, South Christy	1-477-935-8478 x6430
7 Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz	Rex Trail, Suite 280, Howemouth	210.067.6132
8 Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me	Ellsworth Summit, Suite 729, Aliyaview	586.493.6943 x140
9 Glenna Reichert	Delphine	Chaim_McDermott@dana.io	Dayna Park, Suite 449, Bartholomebury	(775)976-6794 x41206
10 Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	Kattie Turnpike, Suite 198, Lebsackbury	024-648-3804

## + BƯỚC 5. THÊM NGƯỜI DÙNG (ADDUSER)

- Mục tiêu:** Tạo form thêm người dùng mới, xử lý thuộc tính lồng (address), và gửi dữ liệu lên App.
- Logic**
  - Dùng useState để quản lý các trường nhập liệu.
  - Xử lý thuộc tính lồng: Khi cập nhật các trường như street (nằm trong address), phải sử dụng Spread Operator (...) để sao chép cả đối tượng cha (address) để tránh mất dữ liệu cũ và đảm bảo React nhận ra sự thay đổi.

- Thao tác Thêm: Gọi onAdd(user) để truyền dữ liệu lên App.
- Cập nhật danh sách: Trong ResultTable, dùng useEffect([user]) để lắng nghe người dùng mới và thêm vào danh sách users.

### 3. Mã minh họa

#### • Component con - AddUser

```

function AddUser({ onAdd }) {
    const [adding, setAdding] = React.useState(false);
    const [user, setUser] = React.useState({
        name: "", username: "", email: "",
        address: { street: "", suite: "", city: "" },
        phone: ""
    });

    const handleChange = (e) => {
        const { id, value } = e.target;
        if (["street", "suite", "city"].includes(id)) {
            setUser({ ...user, address: { ...user.address, [id]: value
} });
        } else {
            setUser({ ...user, [id]: value });
        }
    };

    const handleAdd = () => {
        if (user.name === "" || user.username === "") {
            alert("Vui lòng nhập Name và Username!");
            return;
        }
        onAdd(user);
        setUser({ name: "", username: "", email: "", address: {
street: "", suite: "", city: "" }, phone: "" });
        setAdding(false);
    };

    ---- UI ----
    <div>
        <button onClick={() => setAdding(true)}>Thêm</button>
        {adding && (
            <div>
                <div>
                    <h4>Thêm người dùng</h4>
                    <label htmlFor="name"> Name: </label>
                    <input id="name" type="text" value={user.name}
                        onChange={handleChange}>
                    ...
                    <button onClick={handleAdd}>Thêm</button>
                    <button onClick={()=>setAdding(false)}>Hủy</button>
                </div>
            </div>
        )}
    </div>
}

```

```

        </div>
    ) }

```

4. Khi prop user thay đổi → thêm vào danh sách trong ResultTable:

### **Component ResultTable**

```

React.useEffect(() => {
    if (user) {
        setUsers((prev) => [...prev, { ...user, id: prev.length + 1
    }] );
        onAdded();
    }
}, [user]);

```

## Quản lý người dùng

Thêm

**Thêm người dùng**

Name	UserName	Email	Địa chỉ	Số điện thoại
Phone				
<input type="button" value="Thêm"/> <input type="button" value="Hủy"/>				

ID Họ tên	Username	Email	Địa chỉ	Số điện thoại	Sửa	Xóa
1 Leanne Graham	Bret	Sincere@april.biz	Kulas Light, Apt. 556, Gwenborough	1-770-736-8031 x56442	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
2 Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains, Suite 879, Wisokyburgh	010-692-6593 x09125	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
3 Clementine Bauch	Samantha	Nathan@yesenia.net	Douglas Extension, Suite 847, McKenziehaven	1-463-123-4447	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
4 Patricia Lebsack	Karianne	Julianne.OConner@kory.org	Hoeger Mall, Apt. 692, South Elvis	493-170-9623 x156	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
5 Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	Skiles Walks, Suite 351, Roscoeview	(254)954-1289	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
6 Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info	Norberto Crossing, Apt. 950, South Christy	1-477-935-8478 x6430	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
7 Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz	Rex Trail, Suite 280, Howemouth	210.067.6132	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
8 Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me	Ellsworth Summit, Suite 729, Aliyaview	586.493.6943 x140	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
9 Glenna Reichert	Delphine	Chaim_McDermott@dana.io	Dayna Park, Suite 449, Bartholomebury	(775)976-6794 x41206	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>
10 Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	Kattie Turnpike, Suite 198, Lebsackbury	024-648-3804	<input type="button" value="Sửa"/>	<input type="button" value="Xóa"/>



## BƯỚC 6. SỬA NGƯỜI DÙNG (EDIT – COMPONENT RESULTTABLE)

- Mục tiêu:** Tạo form để người dùng sửa dữ liệu user

- Logic:**

- Kích hoạt form Sửa: Khi nhấn nút **Sửa**, lưu dữ liệu người dùng đó vào state **editing**.
- Kỹ thuật quan trọng: deep copy. Phải sao chép đối tượng user (trong đó có thuộc tính lồng address) vào state **editing** để tránh việc thay đổi dữ liệu trong form làm thay đổi dữ liệu gốc trong bảng (do tham chiếu).
- Lưu dữ liệu: Dùng Array.map() để tìm id tương ứng và thay thế bằng đối tượng editing đã được cập nhật.

- Khai báo THÊM state trong Component ResultTable**

```
const [editing, setEditing] = React.useState(null);
```

- Hàm sao chép dữ liệu user sang editing khi người dùng ấn nút Sửa**

```

function editUser(user) {
  setEditing({ ...user, address: { ...user.address } });
}

```

## 5. Giao diện bổ sung thêm Form Sửa khi ấn nút Sửa.

- o Bao bọc phần bảng (table) kết quả hiện có và phần Form Sửa trong một thành phần cha (ví dụ div)
- o Mỗi khi người dùng sửa trên form thì cập nhật dữ liệu

```

return (
  <div>
    <table>...
    </table>

    {editing && (
      <div>
        <div>
          <h4>Sửa người dùng</h4>
          <label htmlFor="name">Name</label>
          <input id="name" type="text" value={editing.name} onChange={handleChange} />
          ...
          <button onClick={handleSave}>Lưu</button>
          <button onClick={()=>setEditing(null)}>Hủy</button>
        </div>
      </div>
    ) }
  </div>
)

```

## 6. Hàm handleChange để cập nhật thay đổi vào state editing

```

const handleChange = (e) => {
  const {id, value} = e.target;
  if(["street", "suite", "city"].includes(id)){
    setEditing({...editing, address:{...editing.address,
[id]: value}});
  }else{
    setEditing({...editing, [id]:value});
  }
}

```

## 7. Hàm handleSave khi người dùng ấn nút Lưu

```

function handleSave () {
  setUsers(prev => prev.map(u => u.id === editing.id ? editing : u));
  setEditing(null);
}

```

}

## Quản lý người dùng

Tim theo name, userna					
<input type="button" value="Thêm"/>					
ID Họ tên	Username	Email	Địa chỉ	Số điện thoại	
1 Leanne Graham	Bret	Sincere@april.biz	Kulas Light, Apt. 556, Gwenborough	1-770-736-8031 x56442	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
2 Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains, Suite 879, Wisokburgh	010-692-6593 x09125	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
3 Clementine Bauch	Samantha	Nathan@yesenia.net	Douglas Extension, Suite 847, McKenzienhaven	1-463-123-4447	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
4 Patricia Lebsack	Karianne	Julianne.OConner@kory.org	Hoeger Mall, Apt. 692, South Elvis	493-170-9623 x156	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
5 Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	Skiles Walks, Suite 351, Roscoeview	(254)954-1289	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
6 Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info	Norberto Crossing, Apt. 950, South Christy	1-477-935-8478 x6430	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
7 Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz	Rex Trail, Suite 280, Howemouth	210.067.6132	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
8 Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me	Ellsworth Summit, Suite 729, Aliyaview	586.493.6943 x140	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
9 Glenna Reichert	Delphine	Chaim_McDermott@dana.io	Dayna Park, Suite 449, Bartholomewbury	(775)976-6794 x41206	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
10 Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	Kattie Turnpike, Suite 198, Lebsackbury	024-648-3804	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>

Name: <input type="text" value="Clementina DuBuque"/>	UserName: <input type="text" value="Moriah.Stanton"/>	Email: <input type="text" value="Rey.Padberg@karina.b"/>	Street: <input type="text" value="Kattie Turnpike"/>	Suite: <input type="text" value="Suite 198"/>	City: <input type="text" value="Lebsackbury"/>
Phone: <input type="text" value="024-648-3804"/>					
<input type="button" value="Lưu"/>	<input type="button" value="Hủy"/>				

## ✖ BƯỚC 7. XÓA NGƯỜI DÙNG (DELETE)

- Mục tiêu:** Xóa người dùng trực tiếp khỏi danh sách, không cần xác nhận.
- Trong component ResultTable, việc xóa người dùng được thực hiện bằng cách sử dụng hàm **filter()** để tạo ra một mảng mới mà không chứa phần tử có id trùng với người cần xóa.

```
function removeUser(id) {
    // Giữ lại tất cả người dùng có id khác với id cần xóa
    setUsers((prev) => prev.filter((u) => u.id != id));
}
```

## 3. Lưu ý

- React không cho phép chúng ta chỉnh sửa trực tiếp state; do đó, cần tạo **một mảng mới bằng filter()**.
- Hàm **setUsers((prev) => ...)** đảm bảo luôn dùng danh sách users mới nhất, tránh lỗi đồng bộ hóa.

## 🎨 BƯỚC 8. GIAO DIỆN (CSS) VÀ MODAL FORM

- Mục tiêu:** Tạo giao diện bảng dữ liệu, form nhập liệu và modal thân thiện với người dùng, sử dụng CSS cơ bản.

### 2. Mã tham khảo – Modal khi thêm hoặc chỉnh sửa user

Modal: là một cửa sổ nổi lên trên giao diện, yêu cầu người dùng tương tác.

```
/* Modal Overlay */
.modal-overlay {
    position: fixed;
```

```

        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background: rgba(0, 0, 0, 0.4);
        display: flex;
        justify-content: center;
        align-items: center;
        z-index: 999;
    }
    /* Modal Content */
    .modal-content {
        background: #fff;
        padding: 6px;
        width: 300px;
    }

```

### 3. Thêm CSS cho form Add và Edit và thêm <br/> sau mỗi input

```

{adding && (
    <div class="modal-overlay">
        <div class="modal-content">
            <h4>Thêm người dùng</h4>
            ...<input id="name" type="text" value={user.name} onChange={handleChange} /><br/>

{editing && (
    <div class="modal-overlay">
        <div class="modal-content">
            <h4>Sửa người dùng</h4>

#root {
    max-width: 1200px;
    margin: 0 auto;
    background: white;
    border-radius: 12px;
    padding: 30px;
    box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
}

.modal-content input {
    width: 100%;
    margin: 5px 0;
    padding: 6px;
    box-sizing: border-box;
}

```

**Thêm người dùng**

  
  
  
  
  
  


Thêm
Hủy

**Sửa người dùng**

  
  
  
  
  
  


Lưu
Hủy

4. Thêm CSS cho bảng (table, tr, td) và cho button (Xóa, Sửa, Thêm, Lưu, Hủy)

## BUỚC 9. KIỂM THỬ VÀ TỔNG KẾT

Kiểm tra đầy đủ CRUD: Thêm, Sửa, Xóa, Tìm kiếm.

Luồng dữ liệu chuẩn React:

- AddUser → App → ResultTable
- SearchForm → App → ResultTable

Nắm vững useState, useEffect, State Lifting, Controlled Components.

**Quản lý người dùng**

Thêm

ID	Họ tên	Username	Email	Địa chỉ	Số điện thoại	Thao tác
1	Leanne Graham	Bret	Sincere@april.biz	Kulas Light, Apt. 556, Gwenborough	1-770-736-8031 x56442	<span>Sửa</span> <span>Xóa</span>
2	Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains, Suite 879, Wisokyburgh	010-692-6593 x09125	<span>Sửa</span> <span>Xóa</span>
3	Clementine Bauch	Samantha	Nathan@yesenia.net	Douglas Extension, Suite 847, McKenziehaven	1-463-123-4447	<span>Sửa</span> <span>Xóa</span>
4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org	Hoeger Mall, Apt. 692, South Elvis	493-170-9623 x156	<span>Sửa</span> <span>Xóa</span>
5	Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	Skiles Walks, Suite 351, Roscoeview	(254)954-1289	<span>Sửa</span> <span>Xóa</span>
6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info	Norberto Crossing, Apt. 950, South Christy	1-477-935-8478 x6430	<span>Sửa</span> <span>Xóa</span>
7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz	Rex Trail, Suite 280, Howemouth	210.067.6132	<span>Sửa</span> <span>Xóa</span>
8	Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me	Ellsworth Summit, Suite 729, Aliyaview	586.493.6943 x140	<span>Sửa</span> <span>Xóa</span>
9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io	Dayna Park, Suite 449, Bartholomew	(775)976-6794 x41206	<span>Sửa</span> <span>Xóa</span>
10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	Kattie Turnpike, Suite 198, Lebsackbury	024-648-3804	<span>Sửa</span> <span>Xóa</span>