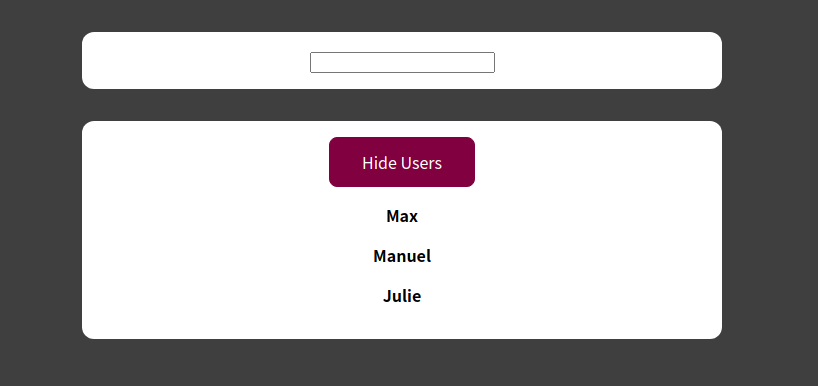
Lab 11.1. Vận dụng các Lifecycle Methods

Ở bài Lab này, bạn sẽ thực hành vận dụng các Lifecycle Methods trong Class Based Component để tạo một trang Web tìm kiếm tên đơn giản. Đồng thời bạn cũng sẽ áp dụng các kiến thức về ErrorBoundary và Content để hoàn thành dự án.



Trong bài Lab này, bạn sẽ xây dựng các Component theo Class-based Component và sửa dụng các life cycle method thay cho các hook như thường dùng ở Functional-based Component. Bạn sẽ cần tải các Resource ở [**link sau**](https://drive.google.com/file/d/1onj02BXYUN1ozS83KeA2KzLVN_LTSSyI/view?usp=sharing).

# 1. Hiển thị danh sách tên người dùng.

Bạn sẽ tạo một Component là UserFinder, Component này sẽ hiển thị Form tìm kiếm cùng với một component con là UserList.js để hiển thị ra danh sách tên những người dùng hiện có.

Để quản lý danh sách người dùng, bạn cũng sẽ tạo một UserContext để xử lý vấn đề này.

Bạn sẽ sử dụng Dummy data như sau để tượng trưng cho dữ liệu về người dùng.

| const DUMMY\_USERS = [  { id: 'u1', name: 'Max' },  { id: 'u2', name: 'Manuel' },  { id: 'u3', name: 'Julie' }, ]; |
| --- |

# 2. Ẩn/hiện danh sách

Tiếp theo, bạn sẽ tạo một State mới là showUsers để tượng trưng cho việc có hiển thị danh sách hay không. Sau đó, bạn hãy thêm sự kiện khi click vào nút “Hide User" thì sẽ đặt showUsers = false.

# 3. Tìm kiếm người dùng

Tiếp theo, bạn sẽ thêm event mỗi khi người dùng nhập vào form tìm kiếm, bạn sẽ lọc danh sách các user và trả về danh sách các người dùng có tên chứa đoạn chuỗi đã nhập vào.

Để hoàn thành yêu cầu này, ở Component **Users.js** bạn cần truyền vào 1 Props là userList tượng trưng cho các User sẽ được hiển thị. Sau đó ở Component **UserFinder.js** bạn sẽ cần tạo 2 state mới như sau:

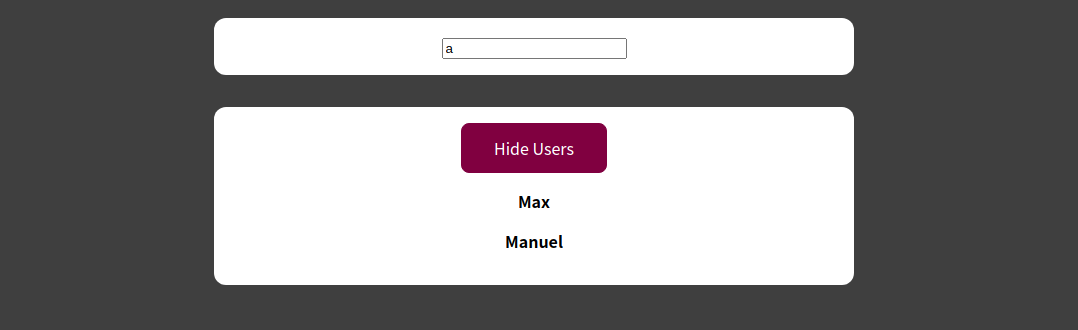
* **filteredUsers**: Danh sách các người dùng sau khi lọc theo từ khóa (mặc định state này ban đầu có giá trị bằng sanh sách User trong Context).
* **searchTerm**: State này tượng trưng cho từ khóa mà người dùng nhập vào form.

Bạn sẽ cần sử dụng componentDidUpdate để bắt sự kiện mỗi khi searchTerm thay đổi, thì sẽ cập nhật lại filteredUsers và truyền xuống Component **User.js**

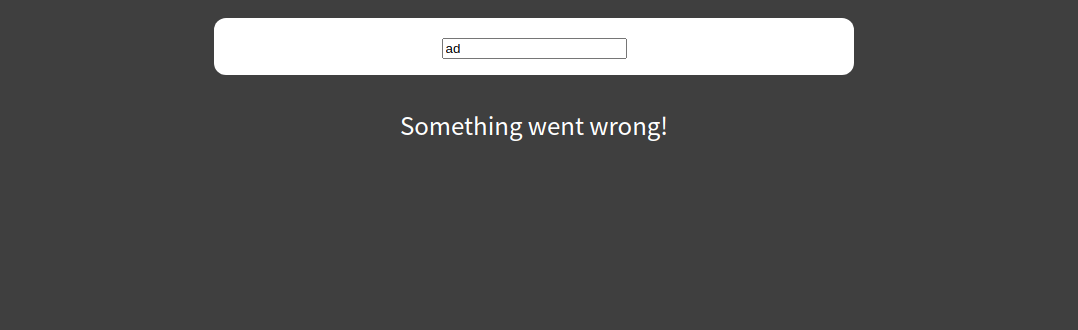
# 4. Hiển thị lỗi

Cuối cùng, bạn sẽ tạo một ErrorBoundary để hiển thị mỗi khi người dùng nhập vào từ khóa và không có User nào thỏa mãn.

Ví dụ nếu có User thỏa mãn:



Nếu không có User nào thỏa mãn:



Để hoàn thành yêu cầu này, ở Component **User.js** bạn sẽ cần dùng **componentDidUpdate** để bắt sự kiện khi Props userList thay đổi, nếu userList mới trống thì sẽ throw ra lỗi.

Tiếp đó, bạn sẽ viết một Component mới bao quanh danh sách User là **ErrorBoundary.js** và sử dụng **componentDidCatch** để kiểm tra xem có lỗi hay không, nếu có thì sẽ hiển thị thông điệp như màn hình.

Ngoài ra, bạn có thể tham khảo video hướng dẫn làm bài Lab tại [link sau](https://funix.udemy.com/course/react-the-complete-guide-incl-redux/learn/lecture/25599736#overview) trong trường hợp chưa hoàn thành được.