Lab 7.1: Tạo ứng dụng quản lý người dùng

Ở bài Lab này, bạn sẽ được thực hành ôn tập lại toàn bộ các kiến thức đã được học ở phần 01, từ đó giúp bạn học các kiến thức ở phần 02 dễ dàng hơn. Bạn sẽ thực hành tạo một ứng dụng đơn giản để nhập liệu và hiển thị thông tin người dùng.

Ứng dụng này gồm các chức năng và giao diện như sau:

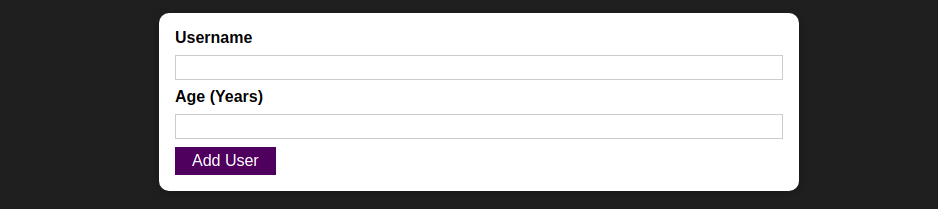
* Thêm mới một User.
* Hiển thị danh sách User.
* Hiển thị thông báo lỗi khi Validate dữ liệu nhập vào.

Để hoàn thiện được bài Lab này, bạn sẽ cần tải các Resource ở [**link sau**](https://drive.google.com/file/d/1mlLKjlaRunl1BzpiyAYBfk3CVb6mWhF7/view?usp=sharing). Các Resource này sẽ bao gồm 2 Component là Button và Card để bạn sử dụng cho việc tạo giao diện. Đồng thời Resource cũng cung cấp các file .css để tạo kiểu cho Website.

# 1. Tạo giao diện thêm User

Đầu tiên, bạn sẽ tạo các Component ban đầu và tạo giao diện form để thêm được một người dùng bao gồm tên và tuổi.

Bạn sẽ cần tạo các Component **AddUser** như sau:



Bạn sẽ tạo một form gồm 2 Input lần lượt cho Username và Age, đồng thời sẽ có một nút bấm để xử lý cho việc thêm User.

# 2. Xử lý sự kiện thêm User

TIếp theo, bạn sẽ xử lý sự kiện để thêm được User vào danh sách. Để lưu trữ danh sách các User đã được thêm vào, bạn sẽ tạo một State dạng mảng gọi là **userList** ở Component **App.js**, sau đó, bạn sẽ truyền vào **AddUser.js** một Props là hàm sẽ thêm dữ liệu User mới vào mảng userList.

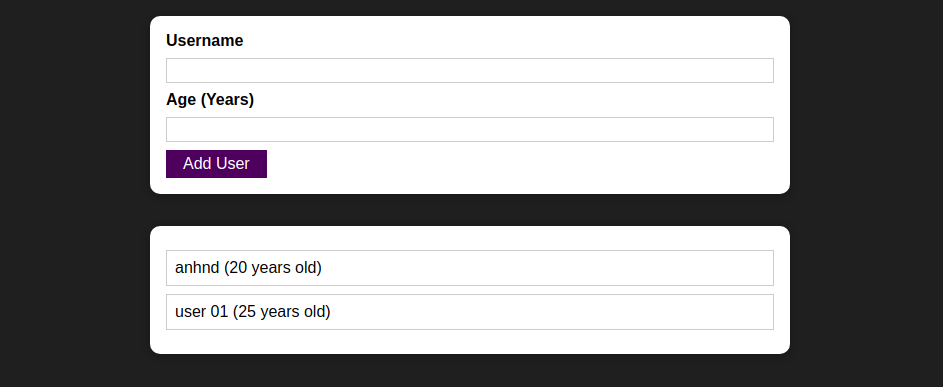
| <**AddUser** onAddUser={**addUserHandler}** /> |
| --- |

Hàm addUserHandler nhận vào fullName và age, sau đó cập nhật dữ liệu vào userList. Một User sẽ là một Object với các thuộc tính như sau:

* **id**: được sinh ra tự đông, bạn có thể sử dụng **Math.random().toString().**
* **fullName**
* **age**.

# 3. Hiển thị danh sách User

Tiếp theo, bạn sẽ tạo một Component là UsersList.js để hiển thị danh sách các người dùng đã được nhập vào. Component này sẽ có giao diện như sau:



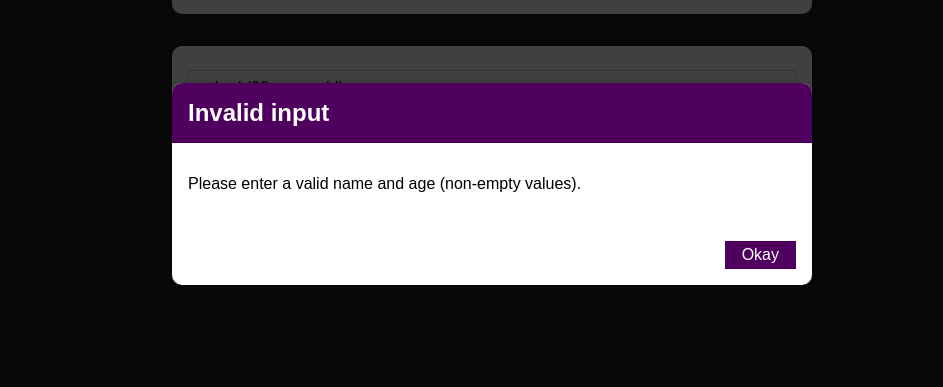
Component này sẽ nhận vào một Props là userList đươc truyền vào từ App.js. Sau đó sẽ hiển thị ra danh sách tương ứng. Bạn có thể sử dụng các file .css đã được cung cấp ở phần Resource để hoàn thành bước tạo kiểu cho Website.

| <**UsersList** users={usersList} /> |
| --- |

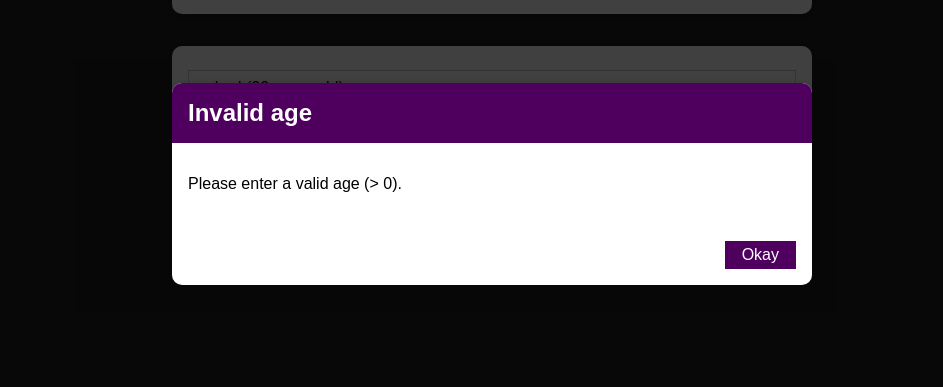
**4. Hiển thị lỗi**

Bạn sẽ cần thực hiện bước Validate dữ liệu nhập vào từ người dùng. Sẽ có 2 trường hợp như sau:

Người dùng nhập thiếu fullName hoặc age, bạn sẽ cần hiển thị lỗi như sau.



Người dùng nhập age <= 0, bạn sẽ cần hiển thị lỗi như sau:



Để hoàn thành yêu cầu này, bạn sẽ cần tạo thêm một Component mới là ErrorModal.js để hiển thị ra các lỗi. Component này sẽ nhận vào 3 Props như sau:

| <**ErrorModal**  title={error.title}  message={error.message}  onConfirm={errorHandler} /> |
| --- |

* **title**: Tiêu đề của Modal.
* **mesage**: Nội dung của Modal.
* **onConfirm**: function sẽ chạy khi nhấn nút đóng Modal (thường sẽ là để set error về rỗng.

Sau đó, ở component AddUser.js, bạn sẽ tạo 1 State mới là error tượng trưng cho lỗi, giá trị mặc định là null. Trước khi thêm mới một User, bạn sẽ validate dữ liệu đầu vào dựa theo các trường hợp bên trên, nếu có trường hợp nào thỏa mãn thì sẽ setState cho error theo lỗi tương ứng.

Đồng thời, bạn cũng sẽ cần sử dụng Conditional Rendering để kiểm tra nếu State error không phải null thì sẽ hiển thị Component **ErrorModal**.

Lưu ý: bạn có thể tham khảo cách làm ở **“Section 8: Time to Practice: A Complete Practice Project”** trên Udemy trong trường hợp chưa hoàn thành được bài Lab.