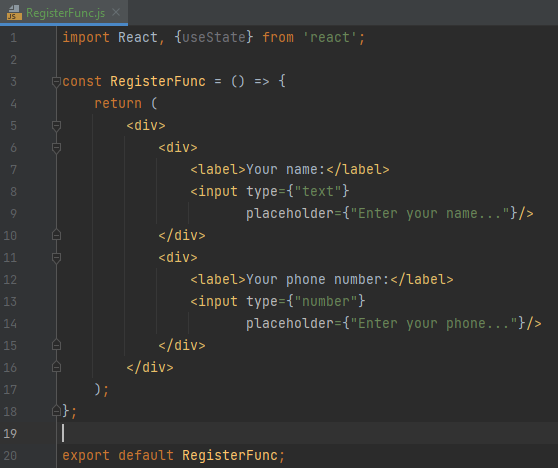
**Xử lý Form**

Trong quá trình xây dựng web với React, chúng ta cần phải làm việc với các form như đăng nhập, đăng ký, feedback,… Cách làm việc với form trong React rất đơn giản, hầu hết chỉ là các phần cơ bản mà chúng ta đã học ở phần trước như component, props, state,…

Ở phần này, chúng ta sẽ tìm hiểu những cách thức tương tác căn bản với form trong React.

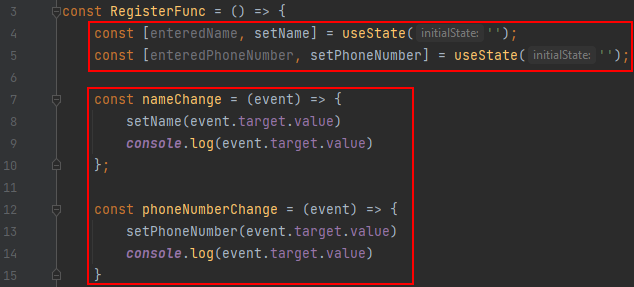
**Lấy giá trị của thẻ input**

Ta sẽ bắt đầu với arrow function RegisterFunc có JSX là 2 thẻ input.

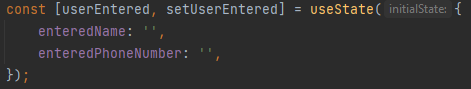


Tiếp theo, chúng ta nhiều cách làm để lấy dữ liệu từ 2 thẻ input trên.

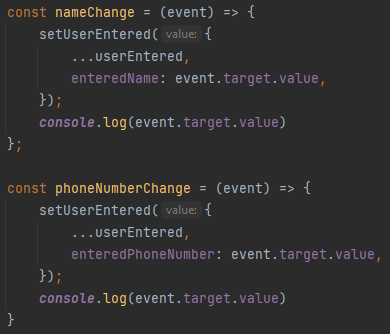
Trước tiên, chúng ta bắt đầu với cách làm phổ biến, khai báo 2 biến state cùng với 2 hàm dùng để cập nhật giá trị được nhập từ input vào biến state.



Một cách ít phổ biến hơn, thay vì khai báo từng biến state, chúng ta sẽ gom chúng lại như sau:

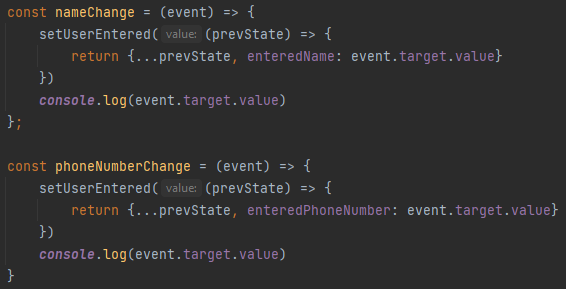


Lúc này, 2 hàm cập nhật của chúng ta cũng phải thay đổi:



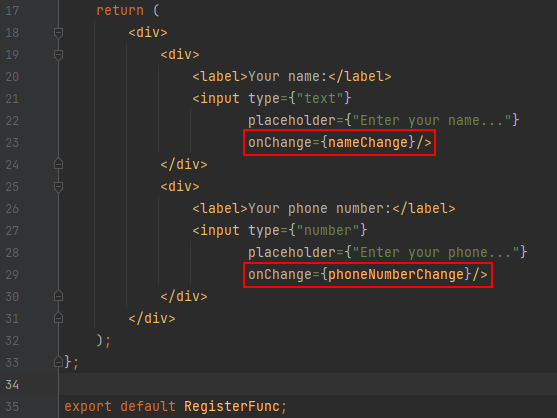
Phương thức set của state sẽ xóa trắng dữ liệu cũ và cập nhật vào giá trị mới. Trong ví dụ này, khi ta sử dụng setUserEntered trong hàm nameChange để cập nhật giá trị cho biến enteredName sẽ khiến giá trị của enteredPhoneNumber bị mất. Do đó, trước khi cập nhật dữ liệu mới vào, ta sử dụng …userEntered để sao chép lại dữ liệu của biến enteredPhoneNumber từ state trước khi cập nhật sang state sau khi cập nhật.

Tuy nhiên, với cách làm này nếu chúng ta thực thi cùng lúc nhiều lần sẽ dẫn đến bất đồng bộ dữ liệu của các state. Vì vậy, chúng ta có một cách khác để xử lý vấn đề này.



Trong setUserEntered, chúng ta viết một arrow function trong nó với đầu vào là prevState để snapshot dữ liệu lúc đó và cho trả về lại một snapshot state mang dữ liệu mới được cập nhật.

Tiếp theo, để có thể bắt được sự kiện nhập của thẻ input và lưu vào state, ta sử dùng onChange cùng với hàm đã viết bên trên vào mỗi thẻ input.



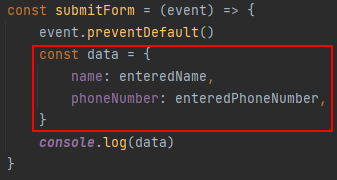
Và thế là xong, chúng ta đã dễ dàng lấy được giá trị từ thẻ input và lưu nó lại trong state.

**Submit Form**

Sau khi lấy được giá trị của input, form cần được submit để tiếp tục các công việc xử lý tiếp theo. Để submit form, chúng ta sẽ thêm sự kiện onSubmit trong form cùng với button submit.

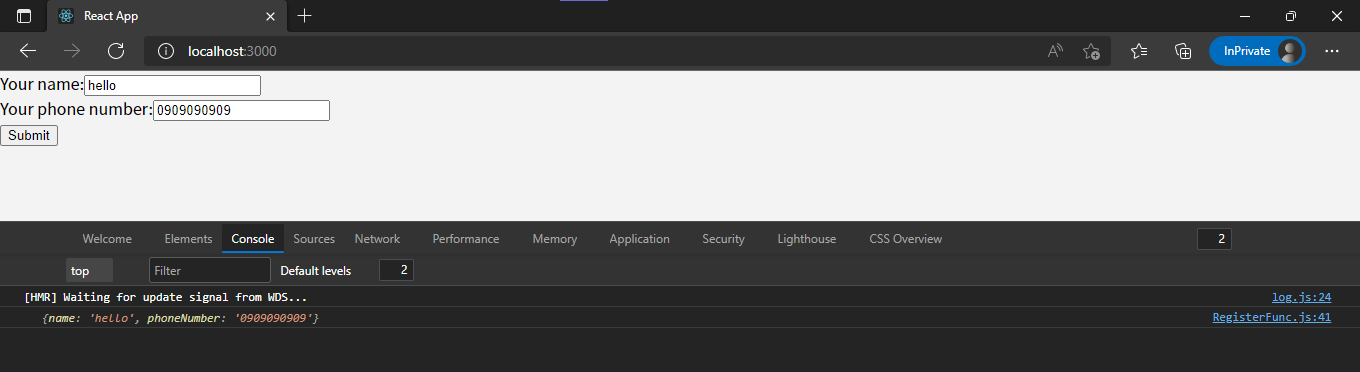


Tiếp theo, ta xây dựng hàm submitForm.



Với cách khai báo từng state thì lúc này chúng ta phải gom các dữ liệu lại thành một object duy nhất. Còn với cách khai báo state gom thì không cần gom nữa.

Kết quả chạy:



**Validation Form**

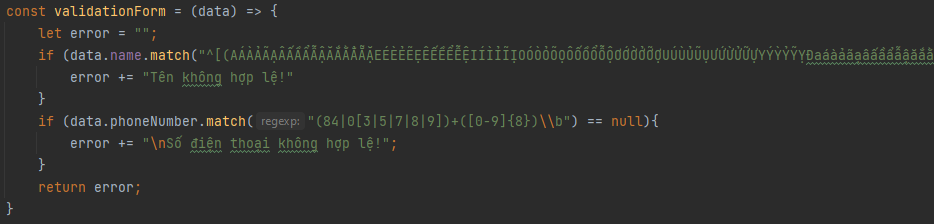
Ta có thể thêm validation trong form bằng cách xây dựng một hàm validation. Dưới đây là hàm validationForm để kiểm tra dữ liệu đầu vào của tên và số điện thoại Việt Nam.

- Regexp cho name:

^[(AÁÀẢÃẠÂẤẦẨẪẬĂẮẰẲẴẶEÉÈẺẼẸÊẾỀỂỄỆIÍÌỈĨỊOÓÒỎÕỌÔỐỒỔỖỘƠỚỜỞỠỢUÚÙỦŨỤƯỨỪỬỮỰYÝỲỶỸỴĐaáàảãạâấầẩẫậăắằẳẵặeéèẻẽẹêếềểễệiíìỉĩịoóòỏõọôốồổỗộơớờởỡợuúùủũụưứừửữựyýỳỷỹỵđ|a-z|A-Z|\s|'"]{1,}$

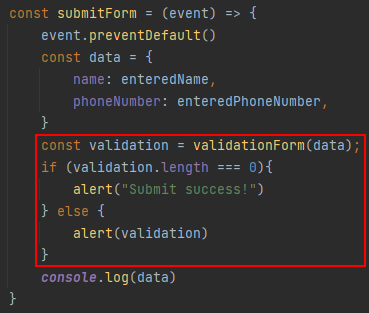
- Regexp cho số điện thoại Việt Nam (năm 2022):

(84|0[3|5|7|8|9])+([0-9]{8})\b



Với phương thức match, ta có thể dễ dàng kiểm tra tính chính xác của giá trị do người dùng nhập vào. Phương thức match trả về null nếu tìm được điểm trùng giữa regexp và giá trị nhập vào và ngược lại.

Tiếp theo, ta gọi hàm validationForm vào trong hàm submitForm được tạo từ trước.



Kết quả chạy:

