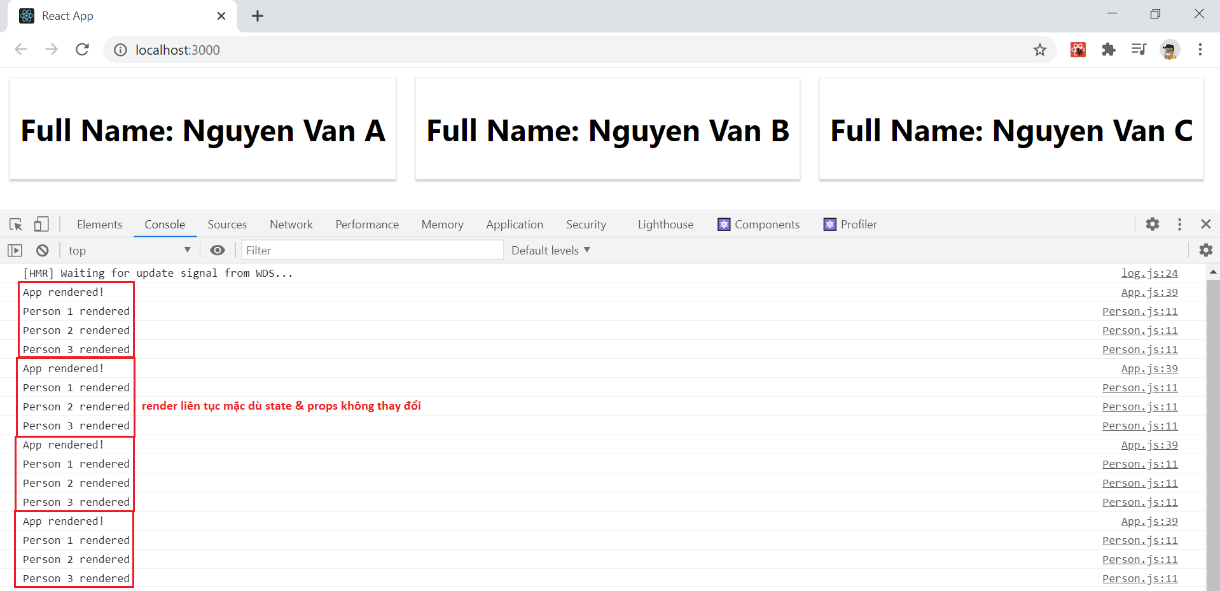
* Khác nhau ở cách triển khai method shouldComponentUpdate()
* **Component**
  + Mặc định thì Component sẽ return true cho method shouldComponentUpdate() 🡺 sẽ re-render bất cứ khi nào có yêu cầu

VD:

|  |  |
| --- | --- |
|  |  |

****

* **PureComponent**
  + Không có method shouldComponentUpdate()
  + So sánh shallow tất cả các thuộc tính của props và state trong method shouldComponentUpdate() 🡺 khi props hoặc state thay đổi mới re-render lại
    - Nếu có cấu trúc dữ liệu phức tạp thì có thể vẫn sẽ update (do sự khác biết nằm ở bên trong)

Có thể sử dụng immutable objects để so sánh complex object

<https://immutable-js.github.io/immutable-js/>

<https://formidable.com/open-source/react-fast-compare/>

|  |  |
| --- | --- |
| *Component* | *PureComponent* |

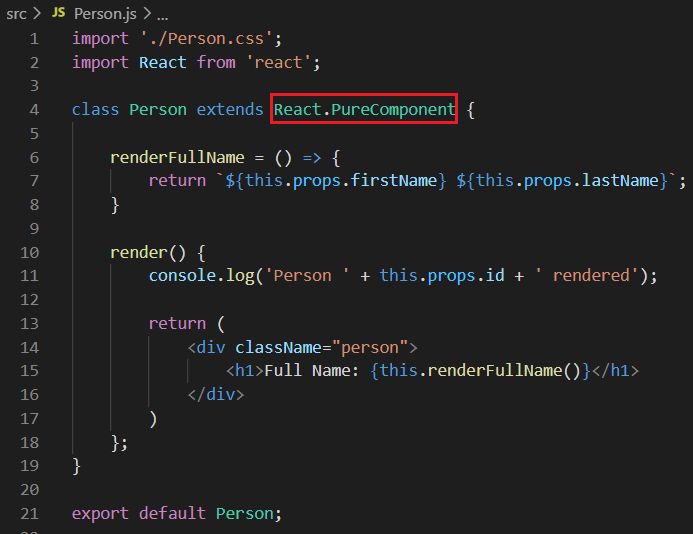
*2 đoạn code này là tương đương nhau*

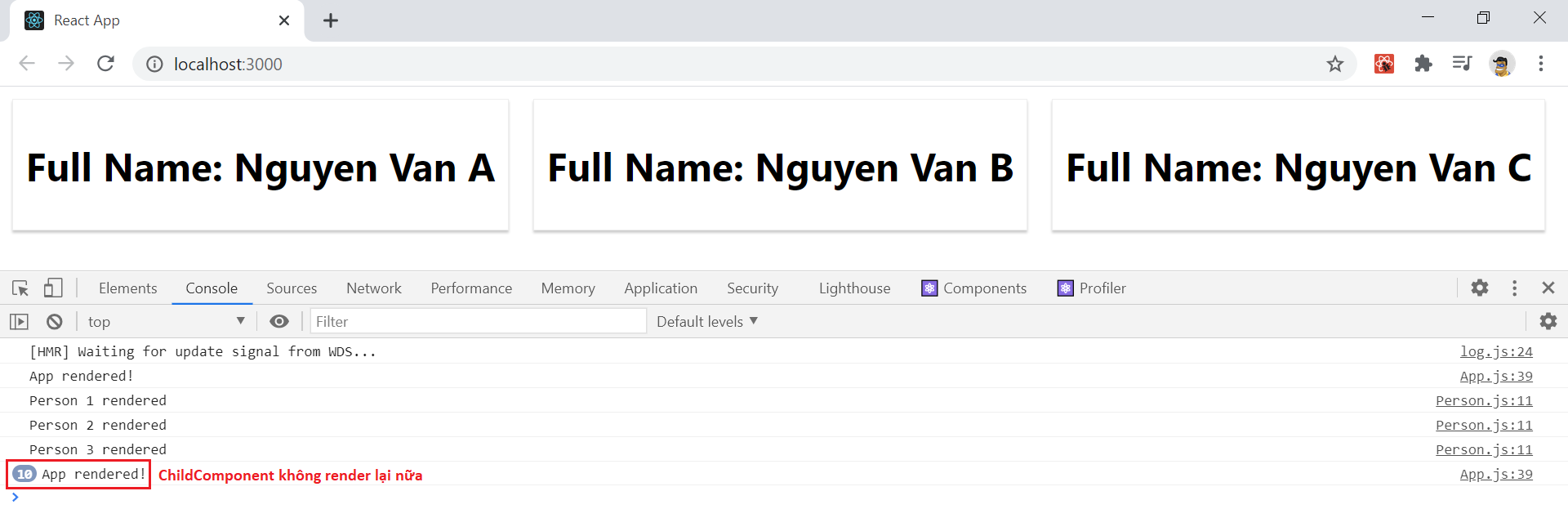
Bỏ qua việc re-render toàn bộ các component con khi component cha thay đổi (do đang so sánh shallow nên sẽ trỏ cùng tới địa chỉ của props 🡺 sẽ không re-render lại ở component con mặc đù component cha đã thay đổi giá trị)

🡺 Nên hãy thiết kế các component con đều là PureComponent

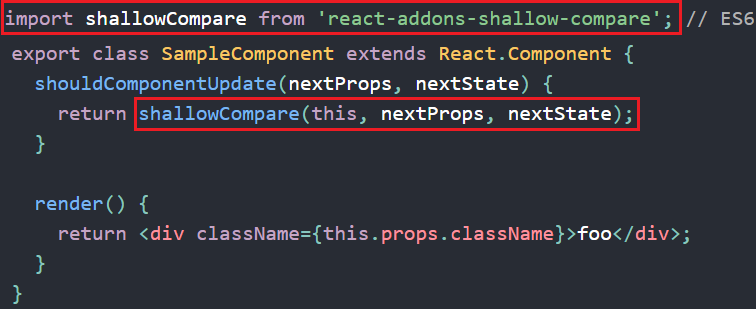
* + Ưu điểm
    - Để tăng hiệu suất (đỡ phải re-render nhiều khi props và state không thay đổi)
    - Là presentation component, tất cả data phụ thuộc vào parent component

VD:





* **Shallow compare**



* + returns true nếu props hoặc state thay đổi 🡺 sẽ re-render
  + return false nếu props và state không thay đổi 🡺 sẽ không re-render
* **Reference**

<https://reactjs.org/docs/react-api.html#reactpurecomponent>

<https://reactjs.org/docs/shallow-compare.html>