|  |
| --- |
| **LocalStorage & SessionStorage**   * Giúp lưu trữ dữ liệu tạm thời ở trình duyệt (Tab Application). * Tương tự cookie. Lưu ở trình duyệt. * Lưu ~5MB * Mất khi xóa lịch sử hoặc dùng lệnh js (removeItem). * Session storage mất đi khi đóng trình duyệt. * Các phương thức chính:   setItem(‘key’,’value’);  getItem(‘key’);  removeItem(‘key’);   * Khi lưu trữ object trong localStorage nên chuyển thành string như sau:     thay vì |
| **Array Destructuring và Object Destructuring** |
| **componentWillMount**  - Chỉ được gọi duy nhất một lần trước thời điểm render đầu tiên.  - Demo mỗi lần F5 lại thì load data từ localStorage vào state như bên dưới: |
| **componentWillReceiveProps(nextProps)**  - Khi props được nhận vào thì gọi. |
| **Làm việc với Form trong ReactJS**  - Tạo 1 state lưu trữ thông tin các element tương ứng trong form. Các element thường gặp: Text, TextArea, Select, Radio, Checkbox (đặc biệt).  - Element phải có attribute name trùng với tên được lưu trong state và attribute value để hiển thị data.  - Gắn attribute onChange vào các element và gọi hàm xử lí thay đổi (onHandleChange).  - Trong hàm onHandleChange thường lấy data trong form như sau:    - Gắn attribute onSubmit cho form, gọi hàm onSubmit và xử lí chặn load lại trang và business theo yêu cầu. Demo dưới là 1 ví dụ:    - Trong trường hợp muốn truyền giá trị ngược lại component cha, ta sẽ thêm một attribute (chẳng hạn như onSubmit) vào phần render ra component con ở component cha và sẽ vào component con để truyền cái state của con lên cha. Sau đó, qua component cha xử lí. |
| **Dynamic setState in react** |
| Hoisting trong JS, JavaScript Closures, curry function |
| **splice() và unshift()** |
| **Chức năng lọc dữ liệu:**  **a. Luồng thực hiện**  - Trong component **TaskList** tạo state lưu **filterName** và **filterStatus**  - Lấy data trong state ra bỏ vào biến trong hàm render.  - **TaskList** component: Ở 2 ô input lọc dữ liệu thiết lập attribute **name** và **value** và **onChange**  - Ở **TaskList** component, viết hàm **onChange(e)** xử lí việc lấy data từ input và set lại state và gọi hàm **onFilter** trong props thằng cha **App** truyền xuống để truyền dữ liệu ra ngoài thằng cha **App.**  - Tại component **App** tạo cho thằng component **TaskList** một cái prop **onFilter** để nhận data truyền từ thằng con **TaskList** ra thằng cha **App** và cái function **onChange** nhận vô 2 tham số **filterName, filterStatus** để xử lí.  - Bổ sung vào state của **App** attribute object **filter** chứa **filterName** và **filterStatus.**  - Ở **onFilter** trong **App** set lại state cho **filter.**  - Xử lí lọc trước khi render.    **b. Tech liên quan**  Skill search:    **c. Hỏi đáp**    Tại sao ở **this.props.onFilter** lại phải truyền tham số như vậy? |
| **Chức năng tìm kiếm**  **a. Luồng thực hiện**  - Thêm thuộc tính keyword vào state của App để lưu trữ data cần tìm kiếm.  - Tạo state trong Search component và thêm thuộc tính name và onChange cho ô input trong component này.  - Xử lí lấy dữ liệu trong hàm onChange.  - Xử lí sự kiện click của button.  - Truyền dữ liệu cần tìm ra ngoài App và xử lí lọc |
| **Chức năng sắp xếp:**  **a. Luồng thực hiện**  - Thêm object sort chứa by (tiêu chí sort) và value (1 tăng dần -1 giảm dần) vào state App.  - Lấy dữ liệu truyền ra từ component Search.  - Xử lí sắp xếp  **b. Skill**  - Truyền dữ liệu từ dropdown list ra ngoài component cha.  - Gọi hàm có truyền tham số trong bắt sự kiện onClick  - Tùy chỉnh css bằng cách sử dụng toán tử 3 ngôi cho className của tag.  - Sử dụng thư viện lodash (xử lí array và object) |