**Webpack :**

* Giúp tải nhanh hơn và giảm tải phía server
* Dành cho dự án lớn, dễ phát triển , quản lý , custum
* Phân thành các module để quản lí và khi tk nào cần thì nó mới load thôi
* Đóng gói file
* Biên dịch file
* Biến các tài nguyên tĩnh hình ảnh , trở thành module
* Biến đổi các mã nguồn js ,less, sass ,css, ES6->ES5
* Đóng gói thành các module => giải quyết vấn đề đường dẫn
* Browser k hiểu JSX (biến đổi JSX => JS)
* Áp dụng CSS Preprocessor để xây dựng stylesheet

**Phân tách Module**

Ch 01

Vong đời của ứng dụng :

Contructor

UNSAFE\_componentWillMount

render

componentDidMount : thực hiện những thư viện Jquery , đưa ra những tương tác ở phần return

Ch 02

Npm uuid : packet sinh ra id ngẫu nhiên

Lodash : thư viện để xử lý với chuỗi

Js check tring content a substring : kiểm tra 1 chuỗi có chứa 1 chuỗi con ko

-Chức năng Form Togo :

Khi vừa vào thì form k cần phải hiển thị nó chỉ hiển thị kh người dùng click vào button add và nếu nó hiện r click phát nữa thì nó sẽ tắt

* Thiết lập trạng thái isShowForm : false => form dựa vào trạng thái đó để hiển thị
* Thực hiện nút ẩn hiện thì truyền isShowForm vào cho tk control
* Thực hiện chức năng tìm kiếm
  + Thì phải thay đổi giá trị state gì đó
  + Muốn lấy nôi dung trong ô input thì đặt ref cho nó
* Bài toán đưa về tìm kiếm phần tử trong item mà name có chứa tk này

Npm lodash : cung cấp rất nhiều method

Lọc chuỗi : js check string contains

Sắp xếp : orderby

* Thực hiện chức năng delete :
  + Khi click vào nút delete thì cần gửi các cái id về tập tin app.js

Sử dụng hàm remote ở lodash để xóa tại tập tin app là ok

* Thực hiện chức năng thêm mới :
  + Lấy được nội dung của form ra đã rồi đẩy nó về cho tk app và tại app sẽ có một method thêm mới 1 công việc : handleSubmit : sẽ nhận vào tên công việc , level công việc thay bằng nó nhận 1 đối tượng công việc
  + Khi gọi form thì phải truyền về cho app nghĩa là phải truyền thông qua cái props
  + Phía form cần tạo ra 1 cái item => gửi về app
  + Và thêm mới vào this.state.item là có 1 công việc mới
  + Thực hiện ẩn form thông qua isShowForm = false là xong
* Thực hiện chức năng edit

App -> list ->item

Khi ckicl thì tên sẽ được chuyển về form

* Cần có 1 cái gì đó lưu trữ tên công việc đó là công việc nào => app cần có 1 cái itemSelectted : phần tử được chọn là phần tử nào => đẩy cho form
* Khi ấn vào edit thì lấy phần tử hiện tại cập nhập lại cho item selected nghĩa là lấy nó ra rồi đẩy về cho app từ app mới cập nhập lại cho itemSelected
* App : handleEdit nhận lại cái item mà người dùng muốn là item nào

LoDash : update object in arr : cập nhập

Redux

* Chạy câu lệnh : npm install --save react-redux
* Import vào tập tin index.js: import { Provider } from ‘react-redux’; => khai báo provider và cung cấp cái Store cho tk app
* Import cả kho chứa (Store) vào nữa
* List kết nối với kho chứa để lấy ra các item => import {connect} from react-redux
* Học 1 tham số trong phương thức connect
* Từ Toggle kết nối tới Store để lấy ra cái isShowForm:
  + Import {connect}
  + Quan sát ở tập tin reduser sẽ quản lí các trạng thái
  + Để lấy được isShowForm thì phải connetc đến strore để lấy
* Chức năng Search :
  + Chả có cái gì quản lí thì cần tạo 1 cái reducer để quản lí nó

Chương 04 : Redux với Project Shopping Cart

B1: cài đặt redux và redux react

B2 : chuẩn bị sẵn thêm 1 số thư mục để làm việc với tk redux

* + - Constants : Các hằng số :
      * Lấy ra các sản phẩm
      * Mua 1 sản phẩm nào đó
      * Cập nhập sản nhâp
      * Xóa sản phẩm ra khỏi giỏ hàng
      * Thay đổi câu thông báo
      * Ngoài ra sẽ phải tạo 1 tập tin mới để định nghĩa các câu thông báo
    - Action : định nghĩa các hành động
    - Reduxer: phân tích ra xem nó có những cái state gì để quản lí và thấy những giá trị có thể bị thay đổi => tạo ra 3 reduser để cung cấp các thứ đó => tạo 1 tập tin index.js và kéo 3 tập tin kia vào là xong
      * Có danh sách các sản phẩm
      * Những sản phẩm nằm trong giỏ hàng
      * Câu thông báo
* Cần tạo 1 store.js (chính là cái cửa hàng quản lí) => import no vao tap tin index, va import proviceder cho no

Công việc tiếp theo là lấy tất cả các sản phẩm từ reduser ra và đổ vào ProductlistItem.js : thực hiện connect

* Chức năng thêm sản phẩm vào giỏ hàng thì cần kết nối cart với store
* Chức năng mua sản phẩm => tập tin productItem => click vào thẻ a và có sự kiện onClick cho nó
* Chức năng câu thông báo ấy thì làm bằng cách gửi
* Chú ý phân biệt trạng thái khi connect
  + Connect(null,mapDispathchToProps) : lấy Dispath về làm cái props và truyền cho reduser (store ) để xử lí
  + Connect(mapDispathchToProps,null) : lấy state về làm props
* Chức năng thống kê nhìn xem nó nằm ở đâu nằm trong method nào => in ra tk item đó => thấy có quality rồi cộng vào và duyệt qua mảng lấy giá tiền sp 1 nhân số sản phẩm rồi cộng với giá tiền sản phẩm 2 nhân với số sản phẩm => sử dụng thư viện lodash : từ khóa lodash sum value of key => sử dụng sumBy
* Chức năng xóa sp :
  + Đầu tiên kiểm tra xem khi click vào button delete thì in ra sản phẩm đó
  + Kiểm tra hàm khi click vào delete : thì đã in ra được thông tin sản phẩm đó qua action
  + Vấn đề trở thành xóa sản phẩm qua cái arr danh sách hiển thị sp => hay nói cách khác xóa cái action.product ra khỏi state của chúng ta
* Chức năng update :
  + Thực hiện chức năng cho button update sự kiện click vào cần cho biết ban update sản phẩn nào , số lượng là bn

Nâng Cao Redux

Components -> kết nối tới : Containers -> kết nối tới redux để lấy dữ liệu ra

PropType : giúp kiểm tra cái propertive nhận vào có đúng theo kiểu dữ liêu gì đó hay ko

Ch 04 :

01 : REACT Router

Cài đặt router : npm install –save react-router-dom

import { BrowserRouter as Router, Route, Link } from "react-router-dom";

Đọc theo tài liệu thấy Router phải đặt ở phía ngoài

Sau đó có thấy:

Khi đường dẫn / thì truy cập Home component

Khi đường dẫn /about truy cập About component nhưng thấy /about có bao bọc cả / => lỗi vì thế cần exact để là duy nhất

<Route exact path="/" component={Home} />

<Route path="/about" component={About} />

Chú ý vì thẻ a sẽ làm nó load lại trang và k đúng với tinh thần của react vì thế k chơi thẻ a mà chơi theo kiểu : <Link to="/about">About</Link> => giao diện k còn được như khi dùng thẻ a nghĩa là vấn đề cần là đặt class vào cho Link => gg react router link add class => sử dụng className như bt thôi

* Vấn đề xảy ra là khi ấn vào phần menu thì nó phải có thêm class active cho nó mà phần link k giải quyết được => đọc tiếp phần NavLink sẽ thấy class activeClassName : sẽ đưa thêm được 1 class vào khi active
* Dùng thêm tk Switch để tránh trường hợp các menu trùng nhau

Ch 05 :

Thực hiện người dùng truy cập vào trang Blog thì connect tới cái tk store

Kiểm tra xem người dùng đăng nhập thành công hay chưa nếu rồi thì ok cho đăng nhập bình thường còn nếu chưa thì đẩy họ sang trang login để làm việc

Thực hiện kiểm tra nếu người dùng đăng nhập rồi thì hiện button logout để họ out ra khỏi hệ thống còn nếu chưa thì người dùng nhập thông tin thì cho họ đăng nhập: kết nối trang login với store

Logout : thì khi click vào nó thì chỉ cần kết nối tới store và set cho cái isLogin = false là xong

Reactjs

05: JSX

Lesson : 08 :

* truyền dữ liệu từ cha vào con thì dùng 1 cái props là xong
* truyền dữ liệu từ con ra cho cha thì dùng 1 cái props và tk con sử dụng cái props đó và gửi dữ liệu ra cho tk cha thông qua 1 cái function

Lesson 10 : todo list