hàng tồn kho,lượng xuất hàng,chi phí hoạt động.

rcjc : to fast create class components (trước đó phải cài plugin React code snippet).

&nbsp;dấu cách trắng trong HTML.

how to install bootstrap in ReactJS.

npm i --save bootstrap@version\_name

npm install --save bootstrap@4.3.1

sau đó,import vào trong file index.js dòng này :

import '../node\_modules/bootstrap/dist/css/bootstrap.min.css';

hoặc dòng này :

import 'bootstrap/dist/css/bootstrap.min.css';

IMPORT CSS to ReactJS:

trong components tạo 4 file: header.js;products.js;header.css;products.css

trong file header.js thêm dòng : import “./header.css”;

how to import font awesome-ReactJS

npm install --save font-awesome

sau đó import dòng này vào index.js

import “../node\_modules/font-awesome/css/font-awesome.min.css”;

or use CDN links

<link **href**="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" **rel**="stylesheet" />

vẫn chưa tìm dc icon fontawesome cho ReactJS.

link CDN cho jquery,nhúng 3 dòng này vào trong index.html

    <script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"></script>

đổi cổng localhost thành 4200 trên browser.

đầu tiên tắt npm start trong command line.

sau đó vào package.json,thêm “set port=4200 &&” vào dòng đầu tiên trong script, like this :

    "start": "set port=4200 && react-scripts start",

folder images,khi cho vào trong folder src thì link ko dc,nhưng khi cho sang folder public thì dùng link sau lại dc :

src="../images/block-chain.png"

trong khi component Product nằm trong folder Components.

link này lỗi nhưng vẫn hiện ảnh : src="../../images/block-chain.png"

trong React-book-store:

cho toàn bộ ảnh vào trong folder public.dùng link sau hiển thị toàn bộ ảnh

<img src="images/products/bi-vo.png" title="bỉ vỏ" alt />

trong react-minimo,cho ảnh vào trong cả public và src,dùng link sau hiển thị dc ảnh.

<img src="../images/banner.png" itle="girls on the hill" className="img-fluid" alt="" />

how to deploy wep-app to Heroku.

1 tạo 1 app mới trên heroku.

2 kết nối app với repo app trên github.

3 vào phần setting(app trên heroku),sau đó add buildpack với địa chỉ <https://github.com/mars/create-react-app-buildpack>.

4 vào deploy,tiến hành deploy như bình thường.

tìm hiểu try catch trong JavaScript.

cài đặt và sử dụng React Router:

npm install --save react-router-dom

handling event.

cách 1:sử dụng constructor.

constructor(props){

super(props);

this.onAddToCart = this.onAddToCart.bind(this);

}

onAddToCart(){  
 alert(this.props.children + ‘ ‘ + this.props.price + ‘ vnd’);  
}

sau đó gán sự kiện onClick vào hàm onAddToCart(ko cần truyền tham số);

onClick = {this.onAddToCart}

c2:ko cần sử dụng constructor.

onAddToCart(){  
 alert(this.props.children + ‘ ‘ + this.props.price + ‘ vnd’);  
}

sau đó gán sự kiện onClick với 1 arrow function.(có truyền tham số

onClick = { () => {this.onAddToCart()} }

c3:vẫn ko cần sử dụng constructor.

gán hàm onAddToCart với 1 arrow function.

onAddToCart = () => {

alert(this.props.children + ‘ ‘ + this.props.price);

}

sau đó trong gán sự kiện onClick với hàm onAddToCart.ko cần truyền tham số.

onClick = {this.onAddToCart}

Refs :dùng để lấy giá trị trong ô input,text area...

các cách lấy giá trị của Refs:

c1: dùng arrow function.

1:gán hàm onAddProduct vào trong event onClick.

2:viết hàm onAddProduct ngoài phần render.

onAddProduct = () => {

console.log(this.refs.name.value);

}

c2:dùng constructor.

1: như trong cách 1.

2:trong constructor.dùng hàm bind để trải dài các giá trị có trong input.

constructor(props){

super(props);

this.onAddProduct = this.onAddProduct.bind(this);

}

3:viết hàm onAddProduct như bình thường.

onAddProduct(){

console.log(this.refs.name.value);

}

lưu ý : trong ô input HTML phải có thuộc tính ref và giá trị của ref(ở đây là name),like this:

<input type="text" className="form-control" ref="name" />

props : truyền từ cha sang con.

State :

-tạo mới 1 state để lưu thông tin,giá trị của component.

-là trạng thái của component.

-khai bao những giá trị cần lưu giữ của riêng components đó.

-tạo state tại constructor.

-gọi state bằng cách : this.state.key.

18-project TodoList:

hiển thị danh sách công việc : dùng localStorage và SessionStorage để lưu giữ liệu tạm thời trên browser.(trong video này sử dụng localStorage).

21: trong video này vẫn fix cứng trạng thái của công việc trong constructor.

chưa làm dc như trong bài học(có thể chọn trạng thái ‘kích hoạt’ hoặc ‘ẩn’);

23: chức năng xóa

khóa thông minh sử dụng machine-learning,học hình ảnh thao tác người dùng để phân biệt chủ nhà hoặc kẻ trộm.

achieve : hoàn thành  
almost : hầu hết  
approach : tiếp cận  
appropriate : thích hợp  
argument : tranh luận  
articles : bài viết  
assign : chỉ định  
assigned : chuyển nhượng,chuyển giao

attach : đính kèm  
attribute : thuộc tính

avoid : tránh  
behavior : hành vi  
bind : trói buộc  
brief : tóm tắt  
certain : nhất định  
characteristics : nét đặc trưng  
clarify : làm rõ  
common : chung

Composer: nhà soạn nhạc

composing: sáng tác  
considered : xem xét

contains : lưu trữ(chứa đựng,bao bọc)

convenient : tiện lợi  
convention : quy ước  
correcsponding : tương ứng  
course : khóa học  
curly braces : dấu ngoặc nhọn { }

current : hiện hành  
define : định nghĩa  
demonstrated : chứng minh  
dependency : phụ thuộc  
describe : diễn tả  
dialog : hộp thoại

distinct : khác biệt

Encountered : đã gặp  
efficient : có hiệu quả  
elegant : thanh lịch  
embedding : nhúng  
encapsulated : đóng gói  
enroll : ghi danh  
essence : bản chất

Extract: trích xuất  
execute : thi hành

fieldset : lĩnh vực  
foundation : nền tảng

fork : nhánh  
frag : miếng  
fragments : những mảnh vỡ  
generate: tạo ra

generally : nói chung là

generic : chung  
hard-coded : mã hóa cứng  
hash : băm (làm nát,làm hỏng)  
immutable : bất biến

implement : hoàn thành,bổ sung  
increment : tăng  
independent : độc lập  
inefficiency : ko hiệu quả

inheritance : di sản  
initialize : khởi tạo

insert : chèn  
instantly : ngay lập tức  
instead : thay thế  
integer : số nguyên

intention : dự định  
interpret : thông dịch

inverse : nghịch đảo

isolation : sự cô lập  
issue : vấn đề

manual : nghịch đảo  
mention : đề cập đến  
modify : sửa đổi  
necessary : cần thiết  
nested : lồng nhau  
otherwise : nếu ko thì  
otherwise : nếu ko thì  
override : ghi đè

panic : hoảng loạn  
pending : đang chờ xử lý  
phase : giai đoạn  
prepare : chuẩn bị  
prevent : ngăn chặn  
probably : có lẽ  
prompt : lời nhắc  
props : thuộc tính  
provided : cung cấp

predictable : dự đoán  
publish : công bố  
puff : phun  
purely : hoàn toàn

rather : hơn  
readable : có thể đọc được dễ đọc

recap : tóm tắt  
receive : nhận được  
recommended : đề nghị  
redirect : chuyển hướng  
reference : tài liệu tham khảo  
register : ghi danh  
reordered : sắp xếp lại

reputation : uy tín

representing : đại diện  
require : chống đỡ  
resume : sơ yếu lí lịch  
retrieve : lấy lại  
reusable : tái sử dụng

sacrifice : sự hy sinh  
scenario : kịch bản  
schedule : lịch trình

sense : giác quan  
separate : tách rời

several : một số  
similar : giống(trông giống)

situation : tình hình  
solve : gỡ rối  
standalone : độc lập  
state : trạng thái

stranger : người lạ

Suspense : lưỡng lự,phân vân

such : như là  
summarize : tóm tắt

temperature : nhiệt độ  
terminate : chấm dứt  
terms : điều kiện  
throught : xuyên qua  
toggle : chuyển đổi

transfer : chuyển khoản  
unique : độc nhất  
unless : trừ khi

Verdict : phán quyết  
via : thông qua

variables : biến số  
within : trong  
worth : có giá trị

bug :

-Error: Maximum update depth exceeded. This can happen when a component repeatedly calls setState inside componentWillUpdate or componentDidUpdate. React limits the number of nested updates to prevent infinite loops.

fix : xóa dấu ngoặc đơn trong hàm,ví dụ

<button onClick={this.changeContent()}>Click Me</button>

sửa thành

<button onClick={this.changeContent}>Click Me</button>