**MYSQL Documentation**

**MySQL là gì?**

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở. là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có khả năng thay đổi mô hình sử dụng phù hợp với điều kiện công việc khả chuyển. MySQL hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

**Ưu điểm của MySQL**

* Sử dụng dễ dàng: MySQL là cơ sở dữ liệu tốc độ cao và ổn định, công cụ này dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích.
* Tính bảo mật cao: MySQL phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet vì nó sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao.
* Đa tính năng: MySQL có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.
* Khả năng mở rộng và mạnh mẽ: Công cụ MySQL có khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.
* Tương thích trên nhiều hệ điều hành
* Cho phép khôi phục: MySQL cho phép các transaction được khôi phục

**Nhược điểm của MySQL**

MySQL bị hạn chế dung lượng, cụ thể, khi số bản ghi của người dùng lớn dần, sẽ gây khó khăn cho việc truy xuất dữ liệu, khiến người dùng cần áp dụng nhiều biện pháp để tăng tốc độ chia sẻ dữ liệu như chia tải database ra nhiều server, hoặc tạo cache MySQL.

**Tại sao nên sử dụng MySQL?**

MySQL được tích hợp đầy đủ các tiện ích, dễ sử dụng, lại thêm khả năng giao dịch an toàn, chi phí thấp,... nên được xem là cơ sở dữ liệu mã nguồn mở phổ biến nhất trên thế giới.

MySQL là một công cụ có tốc độ cao cùng với tính ổn định tốt. Công cụ này dễ sử dụng, lại còn hoạt động được ở nhiều hệ điều hành. Ngoài ra, tính bảo mật của MySQL rất mạnh, có thể sử dụng được ở trên nhiều ứng dụng.

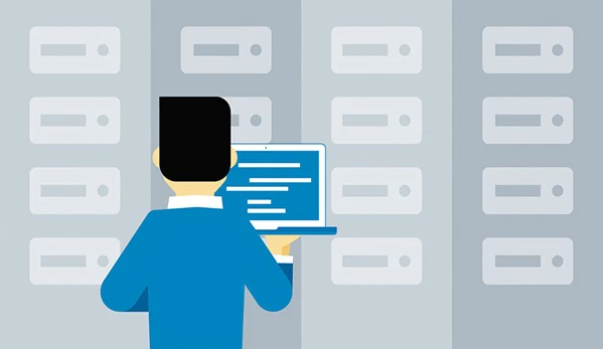
MySQL đã được sử dụng với mục đích nhằm bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác. Có thể nói MySQL là nơi giúp lưu trữ những thông tin của các trang web viết bằng NodeJs, PHP… và MYSQ công có phiên bản được sử dụng hoàn toàn miễn phí.

**Cách thức hoạt động của MySQL**

MySQL tạo ra các bảng với mục đích lưu trữ những dữ liệu, đồng thời có thể định nghĩa sự liên quan với các bảng đó.

Thông thường Client sẽ gửi yêu cầu cho SQL bằng một lệnh đặc biệt ở trên MySQL.

Ứng dụng ở trên Server sẽ phản hồi thông tin và trả về kết quả ở trên máy Client.



*Máy chủ MySQL được sử dụng ở trong môi trường riêng biệt mạng Client – Server*

Nguồn tham khảo:

<https://bizflycloud.vn/tin-tuc/mysql-la-gi-tai-sao-nen-su-dung-mysql-20200917180705499.htm>

**React**

**React là gì?**

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Nó được phát triển bởi Facebook

**Ưu điểm**

* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi.
* Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, chúng ta nên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp chúng ta debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, chúng ta sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

**Nhược điểm**

* Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax
* Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
* React khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
* Khó tiếp cận cho người mới học Web

**Cách thức hoạt động của React**

React hoạt động dựa trên sự kết hợp giữa HTML và JavaScrip. Hiểu đơn giản, bạn có thể chèn hoặc viết code vào thẳng HTML bằng JavaScrip với giải pháp này.

Hầu hết các phần trong React đều được viết bởi JSX (JavaScript XML) để giúp việc tạo React components dễ dàng hơn. Trong React, bạn có thể:

* Tạo đại diện cho DOM bằng hàm Element. Cú pháp HTML của code trong trường hợp này khá tương tự XML components, chỉ khác biệt ở chỗ bạn không dùng DOM class truyền thống mà sử dụng className.
* Là sự kết hợp giữa HTML và JavaScrip

Nguồn tham khảo:

<https://bizfly.vn/techblog/react-la-gi.html>

<https://www.hostinger.vn/huong-dan/react-la-gi-va-no-hoat-dong-nhu-the-nao>

<https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7>

**Redux**

**Redux là gì?**

Redux là một công cụ quản lý trạng thái cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ Elm và kiến trúc Flux của Facebook. Do vậy Redux thường dùng kết hợp với React.

**Tại sao nên sử dụng Redux**

Hầu hết các lib như React, Angular được xây dựng theo một cách sao cho các components đến việc quản lý nội bộ các trạng thái của chúng mà không cần bất kỳ một thư viện hoặc công cụ nào từ bên ngoài.

**Cách thức làm việc của Redux**

**1. Actions**

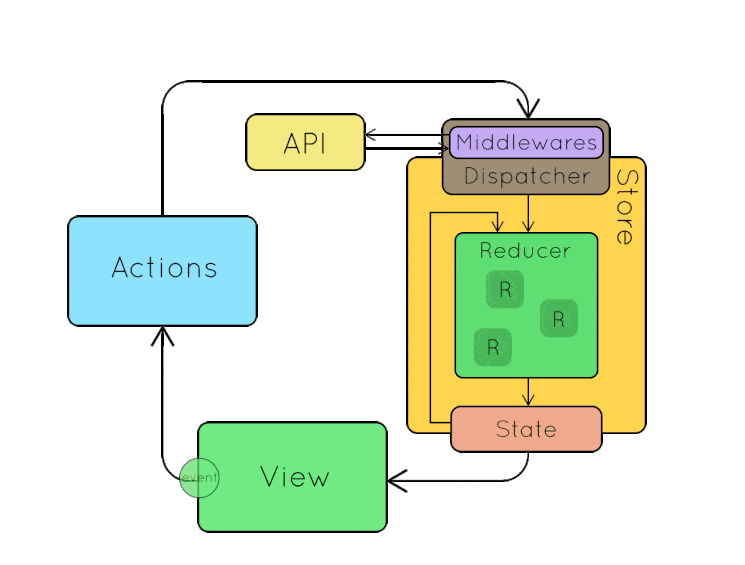
Actions đơn giản là các sự kiện. Chúng là cách mà chúng ta gửi dữ liệu từ app đến Redux store. Những dữ liệu này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

**2. Reducers**

Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một trạng thái mới. Những trạng thái này được lưu như những đối tượng và chúng định rõ cách trạng thái của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

**3. Store**

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể cho phép các trạng thái được lưu, cập nhật trạng thái, và đăng ký hoặc hủy đăng ký các listeners thông qua helper methods.



Nguồn: <https://topdev.vn/blog/redux-la-gi/>

**Redux**

**Redux thunk là gì?**

Redux-Thunk là một middleware phổ biến nhất được dùng để xử lý các action bất đồng bộ trong Redux.

Middleware là những đoạn mã trung gian nằm giữa các request và response. Nó nhận các request, thi hành các mệnh lệnh tương ứng trên request đó. Sau khi hoàn thành nó response (trả về) hoặc chuyển kết quả ủy thác cho một Middleware khác trong hàng đợi.

**Tại sao nên dùng redux Thunk?**

Redux Thunk được sử dụng để xử lý các logic bất đồng bộ phức tạp cần truy cập đến Store hoặc đơn giản là việc lấy dữ liệu như Ajax request.

Nguồn:

https://2kvn.com/redux-thunk-la-gi

<https://topdev.vn/blog/middleware-la-gi/>

**React-Router**

**React-Router là gì?**

React-Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khẳng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng.

**Các thành phần trong React-Router**

**1) BrowserRouter vs HashRouter**

React-Router cung cấp cho chúng 2 thành phần hay sử dụng đó là BrowserRouter & HashRouter. Hai thành phần này khác nhau ở kiểu URL mà chúng sẽ tạo ra và đồng bộ.

BrowserRouter: Được sử dụng phổ biến hơn, nó sử dụng History API có trong HTML5 để theo dõi lịch sử bộ định tuyến của bạn.

HashRouter: Sử dụng hash của URL (window.location.hash) để ghi nhớ mọi thứ.

**2) Route**

Route: Định nghĩa một ánh xạ (mapping) giữa một URL và một Component. Điều đó có nghĩa là khi người dùng truy cập theo một URL trên trình duyệt, một Component tương ứng sẽ được render trên giao diện.

Ví dụ:

<Router>

<div className="App">

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

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

<Route path="/contact" component={Contact} />

<Route component={NotFound}/>

</div>

</Router>

Trong đó:

**path:** Là đường dẫn trên URL.

**exact:** Liúp cho route này này chỉ hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.

**component:** Là component sẽ đươc load ra tương ứng với Route đó.

**3) Link**

Trong HTML thì cặp thẻ để chuyển hướng đó là thẻ <a></a> thì trong react chúng ta sẽ dử dụng cặp thẻ <Link></Link> được import từ React-Router.

**4) NavLink**

NavLink thì rất giống với Link về cách sử dụng, nhưng NavLink tốt hơn vì nó hỗ trợ thêm một số thuộc tính như là activeClassName và activeStyle 2 thuộc tính này giúp cho khi mà nó trùng khớp thì nó sẽ được active lên và chúng ta có thể style cho nó.

**6) Đối tượng Match**

Khi bạn muốn lấy một số thông tin ở trên URL thì bạn có thể dùng đối tượng match để lấy dữ liệu về. Tại cấu hình Router ta chỉ cần truyền thêm đối tượng match vào component mà cần sử dụng đối tượng match

**7) Redirect**

Chức năng dùng để chuyển trang.

Có thể truy xuất thông tin trang trước đó thông qua đối tượng location. Để sử dụng Redirect ta chỉ cần import nó từ react-router.

Nguồn tham khảo:

<https://viblo.asia/p/co-ban-ve-router-trong-reactjs-07LKXzAElV4>

**Axios**

**Axios là gì?**

Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt hay Node.js

**Các option cho Request**

Có rất nhiều các option thêm mà bạn có thể truyền vào khi tạo một request, nhưng dưới đây là những option phổ biến nhất:

* baseURL: nếu bạn chỉ định một base URL, nó sẽ được đính vào trước bất cứ một URL tương đối nào mà bạn sử dụng.
* headers: một object gồm các cặp key/value có thể gửi trong header của request.
* params: một object gồm các cặp key/value mà sẽ được serialize và đính vào URL dưới dạng một query string.
* responseType: nếu bạn mong chờ một format trả về khác ngoài JSON, thì bạn có thể set thuộc tính này thành arraybuffer, blob, document, text hay stream
* auth: truyền vào một object gồm 2 trường username và password được sử dụng để cung cấp quyền truy cập cho các request yêu cầu HTTP Basic auth.

**Trả về một đối tượng**

Khi một request được thực hiện thành công, thì hàm then() của bạn sẽ nhận được một response object với các thuộc tính như sau:

* data: Payload được trả về từ server. Mặc định, Axios mong chờ JSON và sẽ parse nó trở thành một Javascript Object cho bạn.
* status: mã HTTP được trả về từ server.
* statusText: message của HTTP status được trả về bởi server.
* headers: tất cả các header được gửi về từ server.
* config: thiết lập ban đầu của request.
* request: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.

**Trả về đối tượng thất bại**

Nếu có bất kỳ vấn đề gì xảy ra với request, promise sẽ bị reject với một error object chứa những thuộc tính sau đây.

message: message của error ở dạng text.

response: response object được trả về.

config: thiết lập ban đầu của request.

request: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.

Nguồn:

<https://viblo.asia/p/gioi-thieu-ve-axios-mot-http-client-dua-tren-promise-cua-javascript-maGK7MeOlj2>

**Bootstrap**

**Bootstrap là gì?**

Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…, các designer có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với framework này trong quá trình thiết kế giao diện website.

**Tại sao nên dùng Bootstrap**

1.Dễ dàng thao tác

Cơ chế hoạt động của Bootstrap là dựa trên xu hướng mã nguồn mở HTML, CSS và Javascript. Người dùng cần trang bị kiến thức cơ bản 3 mã này mới có thể sử dụng Bootstrap hiệu quả. Bên cạnh đó, các mã nguồn này cũng có thể dễ dàng thay đổi và chỉnh sửa tùy ý.

2.Tùy chỉnh dễ dàng

Bootstrap được tạo ra từ các mã nguồn mở cho phép designer linh hoạt hơn. Giờ đây có thể lựa chọn những thuộc tính, phần tử phù hợp với dự án họ đang theo đuổi. CDN Boostrap còn giúp bạn tiết kiệm dung lượng vì không cần tải mã nguồn về máy.

3.Chất lượng sản phẩm đầu ra hoàn hảo

Bootstrap là sáng tạo của các lập trình viên giỏi trên khắp thế giới. Bootstrap đã được nghiên cứu và thử nghiệm trên các thiết bị. Được kiểm tra nhiều lần trước khi đưa vào sử dụng. Do đó, khi chọn Bootstrap, bạn có thể tin rằng mình sẽ tạo nên những sản phẩm với chất lượng tốt nhất.

4.Độ tương thích cao

Điểm cộng lớn nhất của Bootstrap là khả năng tương thích với mọi trình duyệt và nền tảng. Đây là một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng. Sử dụng Grid System cùng với hai bộ tiền xử lý Less và Sass, Bootstrap mặc định hỗ trợ Responsive và ưu tiên cho các giao diện trên thiết bị di động hơn. Bootstrap có khả năng tự động điều chỉnh kích thước trang website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.

Nguồn tham khảo:

https://wiki.matbao.net/bootstrap-la-gi-cai-dat-bootstrap-web-chuan-responsive/#tai-sao-nen-dung-bootstrap

**Material**

**Material UI là gì?**

Material UI là một thư viện các React Component đã được tích hợp thêm cả Google's Material Design. Theo như giới thiệu trên trang chủ thì được xây dựng nhờ tình cảm với React và Google's Material Design.

**Sử dụng theme trong Material UI**

Có 2 theme cơ bản trong Material UI là light-theme và dark-theme. Light theme cho trang web của bạn phông nền trắng còn ngược lại dark-theme sẽ là phồng nền đen. Material UI cũng cung cấp cho bạn khả năng customize lại theme.

**Sử dụng các Component**

Sau khi khai báo ở trên thì việc sử dụng các Component rất đơn giản. Cần Component nào thì chỉ việc gọi var Component = mui.Component. Tất nhiên là tên Component thì bạn phải xem trên docs của Material UI rồi. Với trên 20 components thì có lẽ là đủ cho trang web của bạn.

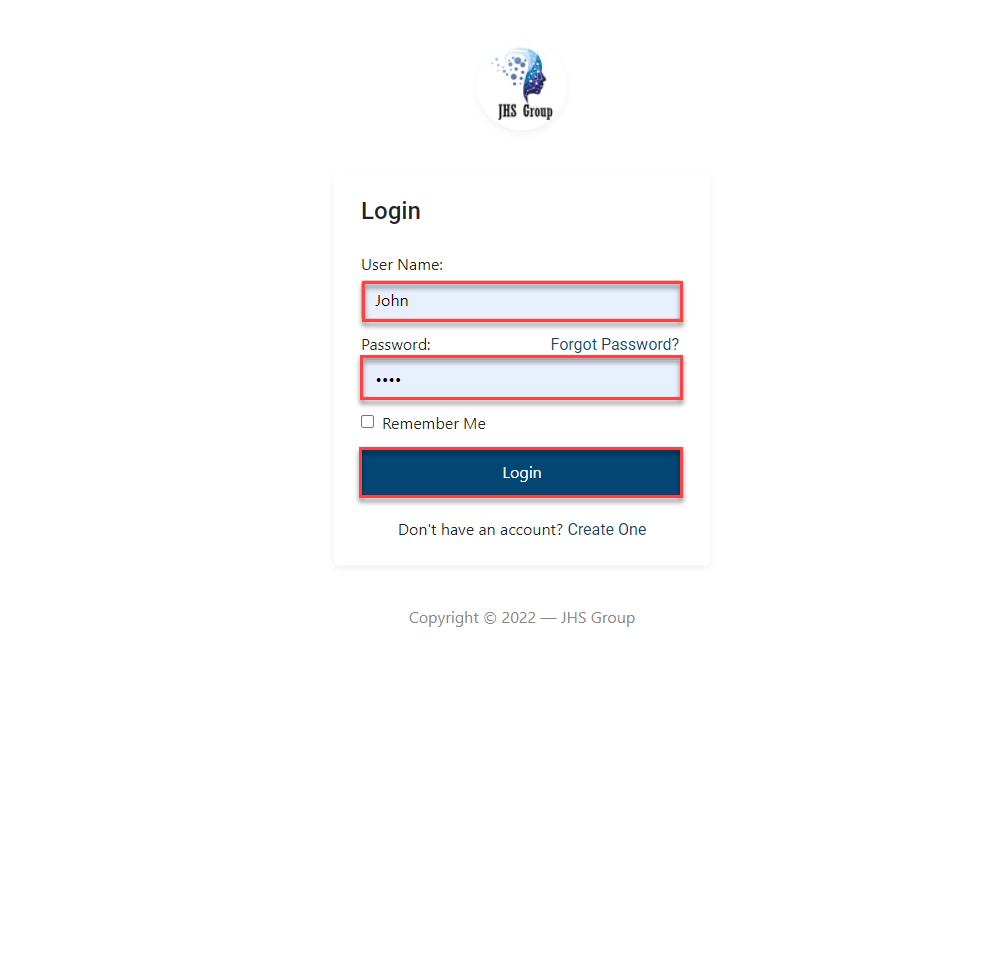
Nguồn tham khảo:

<https://viblo.asia/p/tim-hieu-ve-material-ui-jdWrvwg8vw38>

Login account: with url: <http://localhost:3000/>

User Name: John

Password: 1234



**Sau khi login xong, nó sẽ chuyển tới url:** [**http://localhost:3000/admin/dashboard**](http://localhost:3000/admin/dashboard)

**Ở đây sẽ có 3 cột:**

**+Cột to do: là những cột chứa những task chuẩn bị làm**

**+Cột Doing: những task mình đang làm**

**+Cột Done: nhưng task mình đã làm, trừ những task finished sẽ không có trong cột**

