HỌC VIỆN NÔNG NGHIỆP VIỆT NAM

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO THỰC TẬP CHUYÊN NGÀNH**

**TÊN ĐỀ TÀI:**

**PHÁT TRIỂN ỨNG DỤNG CMS QUẢN LÝ THƯ VIỆN VỚI REACTJS**

Họ và tên sinh viên: **Lê Ngọc Hà- 621126 – K62CNTTA**

**Trần Đình Cừ - 621179 – K62CNTTA**

Giảng viên hướng dẫn**: ThS.Nguyễn Văn Hoàng**

Bộ môn**: Khoa học máy tính**

Tên cơ sở thực tập**: ­­­­­­ Công ty Cổ Phần Đào tạo và Phát triển Công nghệ cao Việt Nam**

**Hà Nội-2021**

**LỜI CẢM ƠN**

*Trong lời đầu tiên của báo cáo thực tập về đề tài “****phát triển ứng dụng với React****” này, chúng em muốn gửi lời cảm ơn và biết ơn chân thành nhất của mình tời tất cả những người đã hỗ trợ, giúp đỡ chúng em về kiến thực và tinh thần trong lần đầu chúng em được thực tập thực hiện đề tài.*

*Trước hết chúng em xin chân thành cảm ơn thầy Nguyễn Văn Hoàng, thạc sĩ và là giảng viên khoa Công Nghệ Thông Tin, Học Viện Nông Nghiệp Việt Nam, người đã trực tiếp hướng dẫn, nhận xét và giúp đỡ chúng em trong suốt quá trình thực hiện đề tài thực tập.*

*Xin chân thành cảm ơn ban đào tạo Học Viện đã tạo điều kiện tốt nhất cho chúng em cũng như các bạn khác trong suốt thời gian học tập và làm đề tài.*

*Cuối cùng chúng em xin gửi lời cảm ơn đến đội ngũ các anh chị của công ty Cổ Phần Đào tạo và Phát triển Công nghệ cao Việt Nam đã giúp đỡ chúng em rất nhiều trong quá trình thực tập cũng như thực hiện đề tài. Do thời gian thực tập và hoàn thành đề tài có hạn, kiến thức vẫn còn nhiều hạn chế. Đề tài chúng em thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định.*

*Em mong nhận được ý kiến đóng góp của các thầy cô để chúng em có thêm kinh nghiệm để tiếp tục trong các đợt thực tập và khóa luận sau này.*

*Em xin chân thành cảm ơn!*

Mục Lục

[Phần I,Mở đầu. 4](#_Toc78114302)

[1,Tên Đề Tài. 4](#_Toc78114303)

[2, Đặt vấn đề. 4](#_Toc78114304)

[3, Mục đích. 5](#_Toc78114305)

[4, Yêu cầu. 5](#_Toc78114306)

[Phần II,Tổng quan tình hình nghiên cứu trong và ngoài nước. 5](#_Toc78114307)

[2.1,Tình hình nghiên cứu trong nước. 5](#_Toc78114308)

[2.2.Tình hình nghiên cứu trên thế giới 6](#_Toc78114309)

[2.3.Tên đề tài và tính thời sự của đề tài. 6](#_Toc78114310)

[Phần III,Nội dung và phương pháp nghiên cứu. 6](#_Toc78114311)

[3.1.Nội dung. 6](#_Toc78114312)

[3.2.Phương pháp tiến hành. 7](#_Toc78114313)

[Phần IV.Kết quả và Thảo luận 7](#_Toc78114314)

[4.1,Giới thiệu Reactjs. 7](#_Toc78114315)

[4.1.1.Tổng quan. 7](#_Toc78114316)

[4.1.2.Đặc trưng. 7](#_Toc78114317)

[4.1.3.Component. 9](#_Toc78114318)

[4.1.4.State và Props. 9](#_Toc78114319)

[4.1.5,Cài đặt dự án với Reactjs. 9](#_Toc78114320)

[4.2,Các thư viện áp dụng. 10](#_Toc78114321)

[4.2.1.Ant-design. 10](#_Toc78114322)

[4.2.2.Redux. 11](#_Toc78114323)

[4.2.3.Redux Saga. 13](#_Toc78114324)

[Generator 13](#_Toc78114325)

[4.3.Back-End. 14](#_Toc78114326)

[4.3.1.NodeJs. 14](#_Toc78114327)

[4.3.2.Express. 15](#_Toc78114328)

[4.3.3.MongoDB. 16](#_Toc78114329)

[4.4,Ứng dụng. 17](#_Toc78114330)

[4.4.1.Các chức năng chính 17](#_Toc78114331)

[Phần V: Kết luận và đề nghị 25](#_Toc78114332)

[5.1. Kết luận 25](#_Toc78114333)

[5.2. Đề nghị 26](#_Toc78114334)

[Tài liệu tham khảo: 26](#_Toc78114335)

*Danh Mục Hình*

[Hình 1.Quản lý State theo cách thông thường 10](#_Toc78114783)

[Hình 2.Giao diện bắt đầu dự án với React. 11](#_Toc78114784)

[Hình 3.Các component được sử dụng ở trang chủ. 12](#_Toc78114785)

[Hình 4.Sơ đồ liên kết dữ liệu. 18](#_Toc78114786)

[Hình 5.Tài Liệu 19](#_Toc78114787)

[Hình 6,Giao diện phần danh sách tài liệu. 20](#_Toc78114788)

[Hình 7,Giao diện phần danh sách quy trình. 21](#_Toc78114789)

[Hình 8,Giao diện danh sách người dùng. 22](#_Toc78114790)

[Hình 9,Giao diện Hồ sơ. 23](#_Toc78114791)

[Hình 10,Modal thêm mới tài liệu. 24](#_Toc78114792)

[Hình 11,Modal thêm quy trình 25](#_Toc78114793)

[Hình 12,Modal thêm mới người dùng. 26](#_Toc78114794)

[Hình 13,Biểu đồ thống kê dữ hiệu bằng HightChart. 27](#_Toc78114795)

[Hình 14,Giao diện đăng nhập. 27](#_Toc78114796)

# Phần I,Mở đầu.

## 1,Tên Đề Tài.

Phát triển ứng dụng web CMS bằng ReactJS quản lý thư viện theo chuẩn ISO.

## 2, Đặt vấn đề.

Trong thời đại công nghệ hóa hiện nay dưới sự phát triển vượt bậc của ngành công nghệ điện tử kéo theo những ngành nghề liên quan cũng phát triển trong đó có cả việc sử dụng những thư viện JavaScript cho các trang website. Vì thế ReactJS đã ra đời và trở nên phổ biến bởi tính năng linh hoạt đơn giản mà nó đem lại.Để ứng dụng điều này,nhóm đã quyết định xây dựng một ứng dụng CMS quản lý thư viện.

## 3, Mục đích.

Xây dựng một ứng dụng với React quản lý thư viện thuận tiện cho việc sử dụng ,cung cấp đầy đủ các chức năng cơ bản cho người dùng,đồng thời đem lại giao diện với độ thẩm mỹ cao.

## 4, Yêu cầu.

* Tìm hiểu các cấu trúc,thành phần của React.
* Xây dựng một ứng dụng web quản lý thư viện.
* Quản lý các loại tài liệu,quy trình.
* Kiểm tra các thông tin tài khoản.
* Giao diện dễ sử dụng.
* Cho phép khách hàng đăng kí và đăng nhập vào hệ thống.
* Cho phép khách hàng lưu trữ,thêm,sửa,xoá tài liệu.
* Hiện thị các thông tin một cách chính xác.

# Phần II,Tổng quan tình hình nghiên cứu trong và ngoài nước.

## 2.1,Tình hình nghiên cứu trong nước.

Cùng với sự tiến bộ của khoa học kỹ thuật,đặc biệt là ngành công nghệ thông tin.Việc đáp ứng nhu cầu thay thế các phương pháp lưu trữ thủ công và phát triển các ứng dụng nhằm lưu trữ Tài liệu đang ngày càng khẳng định vai trò không thể thiếu trong đời sống.

Đi đôi với sự phổ biến của Công nghệ,nhu cầu lưu trữ dữ liệu của con người ngày càng tăng.Những phương pháp lưu trữ thủ công được áp dụng rộng rãi hàng trăm năm qua không còn đáp ứng đầy đủ những nhu cầu lưu trữ dữ liệu của con người.

Cùng với sự phát triển mạnh mẽ của React,đã đem tới một môi trường lập trình web mạnh mẽ,dễ tiếp cận hơn,vì vậy React nói chung và ReactJS nói riêng,đã và đang được áp dụng ngày càng rộng rãi trong việc phát triển ứng dụng trên nền web.

## 2.2.Tình hình nghiên cứu trên thế giới

Ngày nay, [ReactJS](https://topdev.vn/viec-lam-it?q=Reactjs) đã trở nên rất phổ biến bởi những tính năng linh hoạt và đơn giản với hơn 1,300 developer và hơn 94,000 trang web đang sử dụng ReactJS. Nhiều người ám chỉ rằng ReactJS sẽ là tương lai của việc phát triển web.Sự phổ biến của ReactJS một phần nhờ vào các ông trùm về công nghệ như Facebook, PayPal, Uber, Instagram và Airbnb, sử dụng để giải quyết các vấn đề liên quan đến UI. Từ đó kéo theo làn sóng rất nhiều người và công ty tiếp tục tin tưởng sử dụng ReactJS.

Tại các nước phát triển,việc xây dựng và ứng dụng các phần mềm CMS vào quản lý tài liệu đã được áp dụng từ lâu với sự ra đời của các hệ quản trị CMS nổi tiếng như WordPress.

## 2.3.Tên đề tài và tính thời sự của đề tài.

Tên đề tài: Phát triển ứng dụng web CMS quản lý tài liệu với ReactJs.

Tính thời sự:Dựa trên những dữ liệu chúng em tham khảo ở cả trong và ngoài nước như trên,chúng em đã đặt ra đề tài xây dựng ứng dụng web quản lý tài liệu với Reactjs để cung cấp cho người dùng có được trải nghiệm tốt,với các chức năng cơ bản của một website CMS hỗ trợ cho việc lưu trữ,quản lý tài liệu hiệu quả với thời gian phát triển ngắn.

# Phần III,Nội dung và phương pháp nghiên cứu.

## 3.1.Nội dung.

Tìm hiểu về các thành phần trong React.

Xây dựng ứng dụng web với React cho phép người dùng quản lý các tài liệu,quy trình,các chức năng thêm,sửa,xoá.

## 3.2.Phương pháp tiến hành.

Khảo sát yêu cầu thực tế.

Tham khảo lời góp ý từ giảng viên và cán bộ hướng dẫn ở cơ sở thực tập.

Sử dụng kiến thức hiện có về lập trình.

Tìm hiểu và sử dụng thêm các thư viện bổ trợ.

# Phần IV.Kết quả và Thảo luận

## 4.1,Giới thiệu Reactjs.

### 4.1.1.Tổng quan.

ReactJs là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác.

### 4.1.2.Đặc trưng.

Về cơ bản, các tính năng của ReactJS thường xuất phát từ việc chia các thành phần riêng lẻ, cho phép các developer có chức năng chia nhỏ giao diện của người dùng từ một cách phức tạp và biến nó trở thành các thành phần đơn giản. Hiểu đơn giản thì các render dữ liệu không chỉ được thực hiện ở vị trí server mà còn ở vị trí client khi sử dụng ReactJS.

Các thế mạnh của React:

* Đầu tiên phải nói đến tính phổ biến của nó: Theo đó vào tháng 10,năm 2017 có tới hơn 7 triệu lượt tải về trên npm ,do mức độ phổ biến của nó,nên luôn có một cộng đồng Developer đông đảo luôn sẵn sàng để hỗ trợ.
* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
* 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. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
* Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
* Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Tuy nhiên React cũng còn nhiều vấn đề còn tồn tại :

Nhược điểm chính của reactjs đó là:

* 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

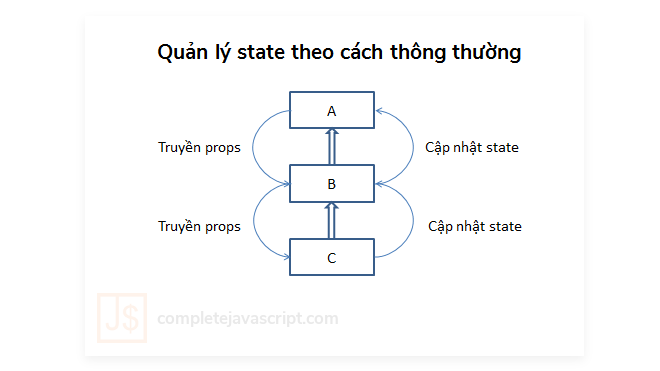
### 4.1.3.Component.

Mỗi component là một phần của website chúng có chức năng đặc biệt và được gọi ra từ các chương trình khác, được liên kết và sử dụng như một bộ phận của chương trình. Mỗi component sẽ là một file riêng biệt dưới dạng file .js.

### 4.1.4.State và Props.

Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render.

State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.



Hình 1.Quản lý State theo cách thông thường

### 4.1.5,Cài đặt dự án với Reactjs.

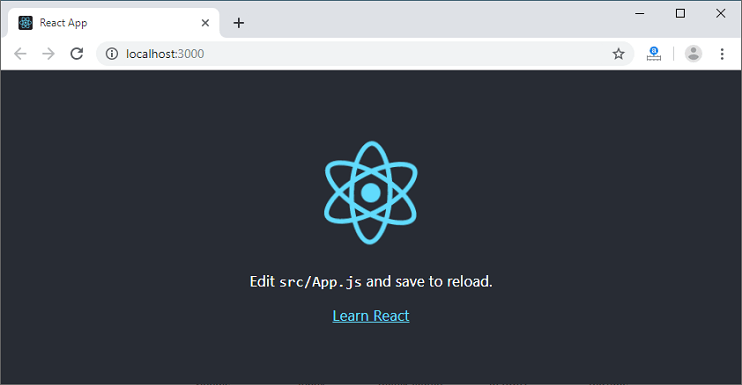
Mở cửa sổ lệnh trên cmd lần lượt chạy các lệnh:

npx create-react-app my-app

cd my-app

npm start

Chúng ta sẽ có giao diện sau:



Hình 2.Giao diện bắt đầu dự án với React.

## 4.2,Các thư viện áp dụng.

### 4.2.1.Ant-design.

Ant là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Tương tự như chuẩn Material Design, Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicket, v.v…Bên cạnh đó Ant cũng có những component riêng thú vị, như LocaleProvider cho phép bạn thay đổi ngôn ngữ trên toàn ứng dụng.

Có thể coi Ant Design cho React là tập hợp của hầu hết các thư viện về React. Nó đáp ứng được hầu hết các yêu cầu của project của bạn mà ban không phải cài thêm bất cứ thư viện nào nữa.

Cài đặt :

npm install antd --save

# or

yarn add antd

Các component Ant Design  :

-Genaral :Button, Icon, Typography.

-Layout :Divider,Grid,Layout,…

-Navigation :Affix,Breadcrumb,Dropdown,…

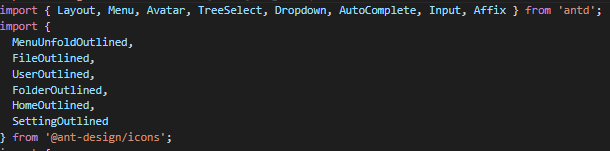
-Data Entry :Autocomplete,Checkbox,…

-Data Display :Avatar,Badge,…

-Feedback :Alert,Message,…

-Other :Backtop,Anchor,….

Các component Ant Design được sử dụng trong trang chủ.



Hình 3.Các component được sử dụng ở trang chủ.

### 4.2.2.Redux.

Redux được tạo ra bởi Dan Abramov, nó được lấy cảm hứng từ Flux của Facebook và ngôn ngữ lập trình hướng chức năng Elm.

Nói một cách đơn giản, Redux là một công cụ quản lý trạng thái. Mặc dù nó được sử dụng chủ yếu với React, nhưng nó có thể được sử dụng với bất kỳ khung hoặc thư viện JavaScript nào khác. Nó rất nhẹ ở mức 2KB (bao gồm cả phụ thuộc), vì vậy bạn không phải lo lắng về việc nó làm cho ứng dụng của bạn có kích thước lớn.

Với Redux, trạng thái ứng dụng của bạn được giữ trong một "store" và mỗi thành phần có thể truy cập bất kỳ trạng thái nào mà nó cần từ "store"này.

Có 3 thành phần của Redux: Actions, Store,Reducer:

Action: Action là các sự kiện, Action gửi dữ liệu từ ứng dụng tới store. Store sẽ đảm nhiệm việc lấy thông tin từ actions. Bản thân action là những simple JS object có thuộc tính type. Một action thì phải có type để chỉ ra action nào đang được thực thi.

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

* Trong một ứng dụng phức tạp sử dụng hàm combineReducers() để combine tất cả các reducers trong application thành một single index reducer. Mỗi một reducers đại diện cho một phần của application state, và Mỗi một reducer thì có state parameter là khác nhau.
* combineReducers() utinity làm cho cấu trúc file trở nên dễ maintain hơn.
* Nếu một một object state chỉ thay đổi một vài giá trị, Redux sẽ tạo ra một object mới nhưng những giá trị không thay đổi sẽ vẫn tham chiếu tới object cũ và chỉ những giá trị mới mới được tạo ra trong object mới. Điều đó làm tăng hiệu quả cho perfonrmance.

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listeners thông qua helper methods.

Bởi vì Redux được xem như một thư viện và nó cho phép bạn có thể quản lý state có trong ứng dụng javascript. Vì vậy, nếu bạn kết hợp nó trong Reactjs sẽ mang lại sự hỗ trợ vô cùng tốt. Ngoài ra, một số lý do giúp cho redux được dùng ngày càng phổ biến là:

* Khả năng dự báo về kết quả. (Predictability of outcome)
* Khả năng maintain bảo trì: Nghiêm ngặt trong cấu trúc làm cho code trở nên dễ dàng bảo trì hơn.
* Khả năng tổ chức: Redux chặt chẽ hơn về việc tổ chức code, điều đó làm cho code trở nên nhất quán và dễ dàng hơn khi làm việc nhóm.
* Server rendering: Nó rất hữu ích đặc biệt là cho việc khởi tạo render, người dùng sẽ có trải nghiệm tốt hơn hoặc cho việc tối ưu bộ mấy tìm kiếm.
* Developer tools: Developer có thể theo dõi mọi thứ xảy ra trong ứng dụng từ các actions đến sự thay đổi của state
* Cộng đồng và hệ sinh thái: Cộng đồng chia sẽ lớn nhiều article, example

### 4.2.3.Redux Saga.

* Redux-Saga là một thư viện redux middleware , giúp chúng ta quản lý những side effect trong ứng dụng redux trở nên một cách đơn giản hơn và dễ kiểm soát hơn. Sagas được triển khai dưới dạng các hàm khởi tạo (generator) cung cấp các đối tượng cho middleware redux-saga.

### Generator

generator là khái niệm cốt lõi trong Redux-Saga, vì vậy trước khi nói về Redux-Saga, chúng ta phải hiểu **generator** là gì, cách chúng hoạt động và cách sử dụng chúng. Nếu bạn đã quen thuộc với nó, bạn có thể bỏ qua phần setup Redux-Saga.

* Một hàm generator được định nghĩa bởi khai báo function\* (từ khóa function được theo sau bởi một dấu hoa thị).
* Các functions có thể được thoát và sau đó nhập lại. Các ràng buộc biến của chúng sẽ được lưu lại qua các lần truy cập lại.
* Trả về một đối tượng Generator, là một loại Iterator.

Redux saga cung cấp cho chúng ta một số method gọi là effect như sau:

* Fork(): sử dụng cơ chế non - blocking call trên function.
* Call(): Gọi function. Nếu nó return về một promise, tạm dừng saga cho đến khi promise được giải quyết.
* Take(): tạm dừng cho đến khi nhận được action.
* Put(): Dùng để dispatch một action.
* takeEvery(): Theo dõi một action nào đó thay đổi thì gọi một saga nào đó.
* takeLastest() : Có nghĩa là nếu chúng ta thực hiện một loạt các actions, nó sẽ chỉ thực thi và trả lại kết quả của của actions cuối cùng.
* yield(): Có nghĩa là chạy tuần tự khi nào trả ra kết quả mới thực thi tiếp.
* Select(): Chạy một selector function để lấy data từ state.

## 4.3.Back-End.

Hiện nay bên cạnh các cơ sở dữ liệu SQL dạng bảng đang rất phổ biến như MySQL, SQLServer,… thì cơ sở dữ liệu NoSQL dạng JSON cũng đang rất được ưa chuộm với những đặc điểm như không yêu cầu phép nối, không cần lược đồ cố định và dễ dàng mở rộng.Nó phù hợp với các cơ sở dữ liệu thời gian thực và dữ liệu khổng lồ đang được các ông lớn tin dùng như: FaceBook, Twitter,….

Và chúng tôi sẽ sử dụng Nodejs, Expressjs, MongoDB để xây dựng nó.

### 4.3.1.NodeJs.

NodeJS (hay Node) không phải là một ngôn ngữ, thư viện hay một framework, Node là môi trường thực thi (Runtime Environment) cho· JavaScript. Mặc dù JS thường cần trình duyệt để hoạt động, nhưng Node.js tạo các cài đặt phù hợp để JS chạy bên ngoài trình duyệt. Nó được xây dựng trên công cụ JavaScript V8 trong Chrome.

Gần đây, các công ty đã tích cực chuyển từ công nghệ backend của họ sang Node.js. LinkedIn đã chọn Node.js thay vì Ruby on Rails vì nó xử lý khối lượng công việc ngày càng tăng tốt hơn và giúp giảm số lượng máy chủ xuống 10 lần.

Và đây là một số ưu điểm của NodeJs:

* Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Bạn có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất.
* Giúp bạn dễ dàng mở rộng khi có nhu cầu phát triển website.
* Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử Nodejs lý nhanh hơn rất nhiều.
* Có khả năng xử lý nhiều Request/s cùng một lúc trong thời gian ngắn nhất.
* Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.
* Phù hợp để xây dựng những ứng dụng thời gian

### 4.3.2.Express.

Express là một web application framework for nodejs, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web. Express rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn. Khi mới tiếp cận với Express mình thực sự bị cuốn hút bởi các API của nó, từ cách sử dụng route, template, đều khá dễ tùy chọn và làm việc. Các tính năng của Express framework phải kể đến như:

* Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
* Định nghĩa routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

### 4.3.3.MongoDB.

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

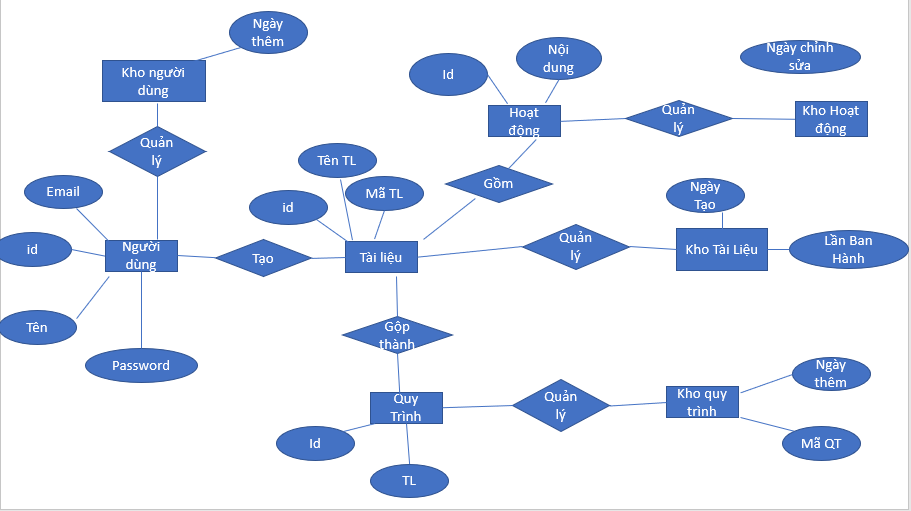
MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Ưu điểm:

* Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.
* Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.
* MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:
* Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh gấp 100 lần so với MySQL

## 4.4,Ứng dụng.

### 4.4.1.Các chức năng chính



Hình 4.Sơ đồ liên kết dữ liệu.

#### 4.4.1.1,Tài Liệu.

Tài liệu là thành phần chính tạo nên dữ liệu cho ứng dụng,một tài liệu có các phần chính như sau,các chức năng gồm:

+Thêm tài liệu.

+Xoá tài liệu.

+Chỉnh sửa thông tin tài liệu.

+Tải xuống tài liệu.

+Xem lịch sử chỉnh sửa tài liệu.

Một tài liệu gồm:

Tên Tài liệu,

Mã tài liệu,

File tài liệu

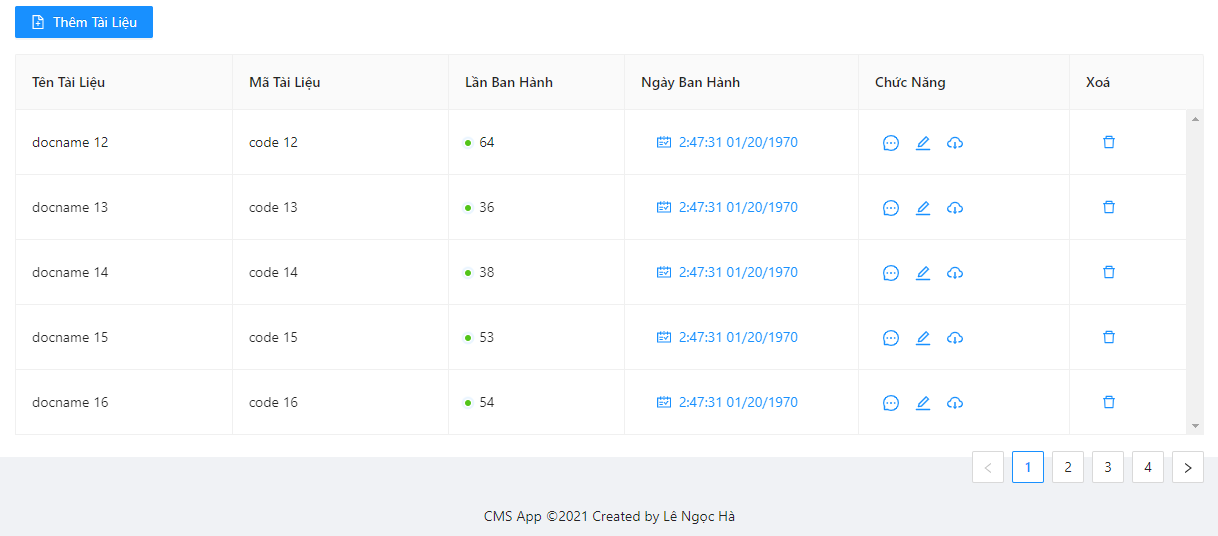


Hình 5.Tài Liệu

Kho tài liệu.

Các cột tài liệu được thêm sẽ gồm:

* Tên Tài Liệu.
* Mã Tài Liệu.
* Lần Ban Hành.
* Ngày Ban Hành.
* Chức Năng.
* Xoá



Hình 6,Giao diện phần danh sách tài liệu.

#### 4..4.1.2,Quy trình

Chức năng thứ hai đó là cho phép người dùng gộp các tài liệu thành quy trình,quy trình gồm:

-Mã quy trình.

-Các tài liệu có trong quy trình.

Các chức năng cho quy trình gồm:

+Thêm ,sửa,xoá quy trình.

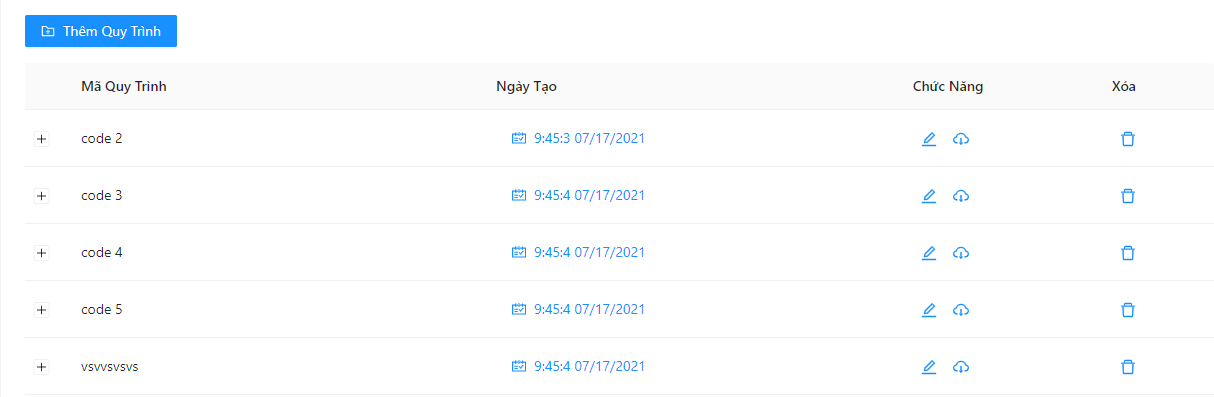
+Thêm mới tài liệu vào quy trình.

+Tải quy trình.

Thêm quy trình cho phép chúng ta gộp nhiều tài liệu vào thành một quy trình,giúp dễ dàng quản lý,tìm kiếm các tài liệu.

Các trường dữ liệu trong quy trình gồm:

* Mã quy trình.
* Ngày tạo.
* Chức năng.
* Xoá quy trình.



Hình 7,Giao diện phần danh sách quy trình.

#### 4.4.1.3.Người dùng.

Quản lý người dùng

Cho phép người quản trị có thể quản lý danh sách người dùng.

Chức năng gồm:

+Thêm người dùng.

+Sửa thông tin.

+Xoá người dùng.

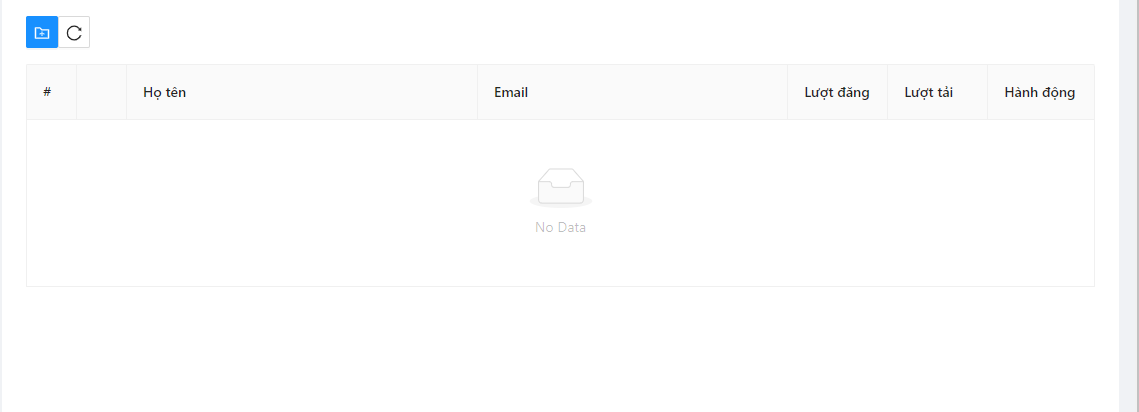
+Thống kê các hoạt động của người dùng trên ứng dụng.

Bảng người dùng cho phép quản trị viên có quyền quản trị đối với người dùng,

Sửa thông tin và thống kê hoạt động của người dùng.

Bảng người dùng sẽ gồm:

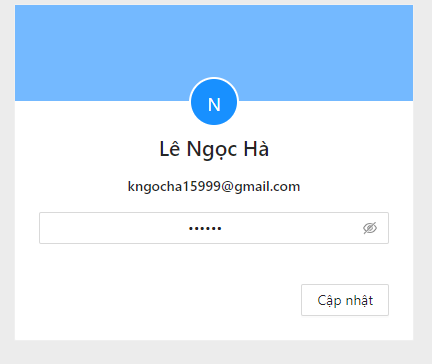
* Họ tên.
* Email
* Lượt đăng.
* Lượt Tải.



Hình 8,Giao diện danh sách người dùng.

* Profile :

Cho phép người dùng thay đổi password.



Hình 9,Giao diện Hồ sơ.

#### 4.4.1.4.Các hoạt động trên component.

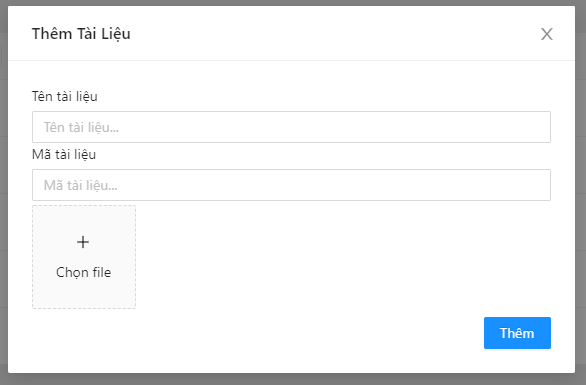
* Thêm mới tài liệu:

Người dùng cần nhập đầy đủ thông tin gồm:

Tên tài liệu

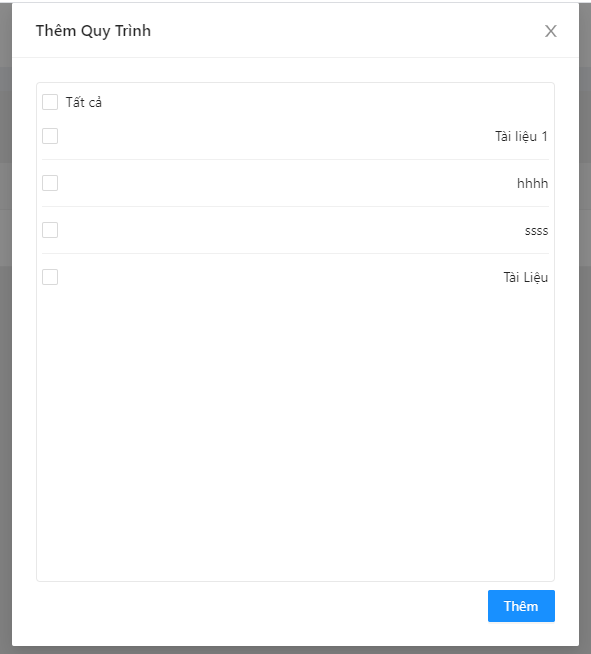
Mã tài liệu

Upload file:file tài liệu.



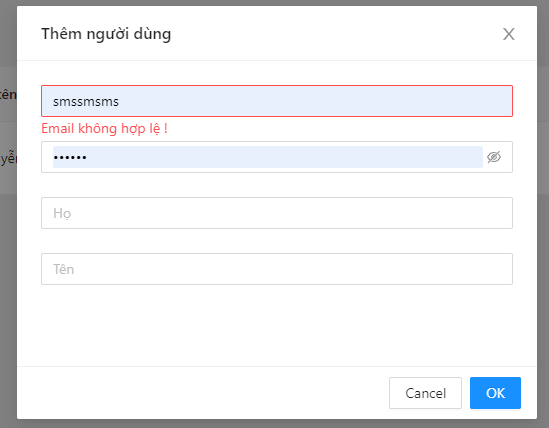
Hình 10,Modal thêm mới tài liệu.

* Thêm quy trình bằng cách sử dụng modal thêm quy trình,người dùng sẽ sử dụng các checkbox để xác định các tài liệu sẽ được gộp lại trong quy trình.



Hình 11,Modal thêm quy trình

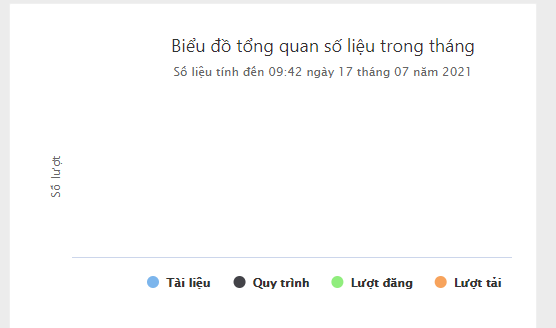
* Thêm người dùng:



Hình 12,Modal thêm mới người dùng.

* Biểu đồ thống kê số liệu.

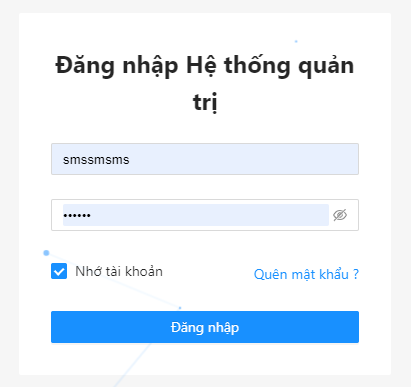
Biểu đồ thống kê các chỉ số cơ bản trong ứng dụng trong một tháng.



Hình 13,Biểu đồ thống kê dữ hiệu bằng HightChart.

#### 4.4.1.3.Đăng nhập.

Giao diện đăng nhập:



Hình 14,Giao diện đăng nhập.

# Phần V: Kết luận và đề nghị

## 5.1. Kết luận

Sau một thời gian thực tế tại công ty cũng như trong quá trình thực hiện đề tài thực tập, nhóm đã đạt được một số kết quả như sau:

- Kiến thức chuyên môn:

+ Nắm được các kiến thức về Front-end

+ Học được kiến thức về ReactJs

- Kinh nghiệm thực tế

+ Tiếp xúc và tham gia môi trường làm việc tại doanh nghiêp.

+ Vận dụng kiến thức đã học vào trong thực tế.

- Kỹ năng mềm

+ Tự tin hơn trong giao tiếp.

+ Cải thiện kỹ năng làm việc nhóm.

## 5.2. Đề nghị

Trong quá trình thực hiện đề tài nhóm đã cố gắng hết sức hoàn thiện chương trình demo nhưng do điều kiện thời gian có hạn và kinh nghiệm thực thế, kiến thức còn hạn hẹp nên chương trình chưa được hoàn thiện tốt nhất và vẫn còn nhiều thiếu sót. Vì vậy hướng phát triển tiếp theo dự kiến của nhóm là:

- Cải thiện thêm về phần giao diện và chức năng.

- Xây dựng thêm hệ thống quản lý một cách tối ưu, hoàn chỉnh hơn.

## Tài liệu tham khảo:

ReactJS: <https://reactjs.org>.

AntDesign: [https://ant.design](https://ant.design/docs/react/introduce).

Redux: [https://redux.js.org](https://redux.js.org/).

Redux Saga: [https://redux-saga.js.org](https://redux-saga.js.org/).

NodeJs: <https://nodejs.org>.

Express: <https://expressjs.com>.

MongoDB: <https://www.mongodb.com>.