**Danh mục các ký hiệu, các chữ cái viết tắt:**

1. API: Application Programming Interface
2. CNTT: Công nghệ thông tin
3. CSS: Cascading Style Sheets
4. DOM: Document Object Model
5. JS: JavaScript
6. HTML: Hypertext Markup Language
7. MVC: Model-View-Controller
8. SEO: Search Engine Optimization
9. TMDT: Thương mại điện tử

**Danh mục các sơ đồ, bảng biểu, hình ảnh**

[Sơ đồ 2. 1. Database Diagram 22](#_heading=h.3u2rp3q)

[Sơ đồ 2. 2. Cơ sở dữ liệu quan hệ 26](#_heading=h.47hxl2r)

[Hình ảnh 2. 1. Javascript 10](#_heading=h.3oy7u29)

[Hình ảnh 2. 2. React 12](#_heading=h.243i4a2)

[Hình ảnh 2. 3. Redux 13](#_heading=h.j8sehv)

[Hình ảnh 2. 4. Redux 14](#_heading=h.338fx5o)

[Hình ảnh 2. 5. React Router 14](#_heading=h.1idq7dh)

[Hình ảnh 2. 6. Axios 15](#_heading=h.42ddq1a)

[Hình ảnh 2. 7. HTML5 17](#_heading=h.2hio093)

[Hình ảnh 2. 8. CSS 17](#_heading=h.wnyagw)

[Hình ảnh 2. 9. ExpressJS 18](#_heading=h.3gnlt4p)

[Hình ảnh 2. 10. Github 20](#_heading=h.2uxtw84)

[Bảng 2. 1. Mô tả chi tiết chức năng website phía người dùng 22](#_heading=h.38czs75)

[Bảng 2. 2. Mô tả chi tiết chức năng phía người quản lý (admin) 24](#_heading=h.1nia2ey)

[Bảng 2. 3. Ký hiệu các ràng buộc 27](#_heading=h.2mn7vak)

[Bảng 2. 4. Mô tả chi tiết đối tượng người quản trị (admins) 27](#_heading=h.11si5id)

[Bảng 2. 5. Mô tả chi tiết đối tượng nhân viên (users) 28](#_heading=h.3q5sasy)

[Bảng 2. 6. Mô tả chi tiết đối tượng thiết bị (items) 28](#_heading=h.25b2l0r)

[Bảng 2. 7. Mô tả chi tiết đối tượng loại thiết bị (item\_types) 28](#_heading=h.3ls5o66)

[Bảng 2. 8. Mô tả chi tiết đối tượng danh mục thiết bị (categories) 29](#_heading=h.20xfydz)

[Bảng 2. 9. Mô tả chi tiết đối tượng quyền hạn (permissions) 29](#_heading=h.4kx3h1s)

[Bảng 2. 10. Mô tả chi tiết đối tượng trạng thái (statuses) 29](#_heading=h.302dr9l)

[Bảng 2. 11. Mô tả chi tiết đối tượng kho hàng (stocks) 29](#_heading=h.1f7o1he)

[Bảng 2. 12. Mô tả chi tiết đối tượng loại kho hàng (stock\_types) 30](#_heading=h.3z7bk57)

[Bảng 2. 13. Mô tả chi tiết đối tượng yêu cầu nhân viên (staff\_requests) 30](#_heading=h.2eclud0)

[Hình ảnh 3. 1. Giao diện đăng nhập 31](#_heading=h.3dhjn8m)

[Hình ảnh 3. 2. Giao diện bảng điều khiển danh sách thiết bị 32](#_heading=h.1smtxgf)

[Hình ảnh 3. 3. Giao diện thêm thiết bị 33](#_heading=h.4cmhg48)

[Hình ảnh 3. 4. Giao diện gửi thông báo qua email 33](#_heading=h.2rrrqc1)

[Hình ảnh 3. 5. Giao diện gửi email thành công 34](#_heading=h.16x20ju)

[Hình ảnh 3. 6. Giao diện nội dung email 34](#_heading=h.3qwpj7n)

[Hình ảnh 3. 7. Giao diện chỉnh sửa thiết bị 35](#_heading=h.261ztfg)

[Hình ảnh 3. 8. Giao diện cảnh báo khi xoá thiết bị 35](#_heading=h.l7a3n9)

[Hình ảnh 3. 9. Giao diện quản lý loại thiết bị 36](#_heading=h.356xmb2)

[Hình ảnh 3. 10. Giao diện thêm loại thiết bị 36](#_heading=h.1kc7wiv)

[Hình ảnh 3. 11. Giao diện chỉnh sửa loại thiết bị 37](#_heading=h.44bvf6o)

[Hình ảnh 3. 12. Giao diện trang quản lý danh mục thiết bị 37](#_heading=h.2jh5peh)

[Hình ảnh 3. 13. Giao diện trang yêu cầu duyệt phía nhân viên 38](#_heading=h.ymfzma)

[Hình ảnh 3. 14. Giao diện tạo mới yêu cầu duyệt 38](#_heading=h.3im3ia3)

[Hình ảnh 3. 15. Chi tiết đơn yêu cầu duyệt 39](#_heading=h.1xrdshw)

[Hình ảnh 3. 16. Giao diện quản lý nhân viên 40](#_heading=h.4hr1b5p)

[Hình ảnh 3. 17. Giao diện thêm nhân viên 40](#_heading=h.2wwbldi)

[Hình ảnh 3. 18. Giao diện sửa thông tin nhân viên 41](#_heading=h.1c1lvlb)

[Hình ảnh 3. 19. Giao diện cảnh báo xoá tài khoản nhân viên 41](#_heading=h.3w19e94)

[Hình ảnh 3. 20. Giao diện xác nhận đặt lại mật khẩu cho nhân viên 42](#_heading=h.2b6jogx)

[Hình ảnh 3. 21. Nội dung email nhận mật khẩu 42](#_heading=h.qbtyoq)

[Hình ảnh 3. 22. Giao diện chức năng quản lý danh sách yêu cầu duyệt phía admin 43](#_heading=h.3abhhcj)

[Hình ảnh 3. 23. Chi tiết yêu cầu duyệt với nút xử lý 43](#_heading=h.1pgrrkc)

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Giới thiệu đề tài.

### 1.1.1. Tình hình sử dụng internet ở Việt Nam.

Trong vòng 1 năm qua, đại dịch Covid-19 đã tác động mạnh mẽ tới mọi mặt của đời sống xã hội. Bên cạnh những ảnh hưởng tiêu cực; việc ứng phó với dịch bệnh cũng đã tạo ra nhiều động lực cho sự phát triển, đặc biệt là thúc đẩy quá trình chuyển đổi số và ứng dụng khoa học công nghệ vào mọi mặt của đời sống xã hội.

Năm 2020, Việt Nam bắt đầu khởi động chuyển đổi số quốc gia với 3 trụ cột là Chính phủ số, Kinh tế số và Xã hội số. Một trong những yếu tố cốt lõi đảm bảo cho sự thành công của công cuộc chuyển đổi số chính là hạ tầng viễn thông và thói quen sử dụng internet của người dân và doanh nghiệp.

### 1.1.2. Thực trạng chuyển đổi số tại các doanh nghiệp ở Việt Nam.

Trong những năm gần đây cùng với sự phát triển nhanh chóng của công nghệ đã đưa doanh nghiệp vào một cuộc chuyển mình lớn mang tên chuyển đổi số. Với điều kiện là một nước đang phát triển với khả năng tiếp cận công nghệ nhanh chóng và linh hoạt, các doanh nghiệp Việt Nam đứng trước cơ hội bắt kịp xu thế chung của nền kinh tế thế giới cũng như nguy cơ tụt hậu xa hơn nữa nếu thất bại trong cuộc đua này. Chính vì vậy, chuyển đổi số không còn là tầm nhìn mà đã trở thành nhu cầu hiện tại, thực tế mà các doanh nghiệp Việt Nam cần thực hiện và thực hiện thành công.

## 1.2. Lý do hình thành đề tài.

Nắm bắt được tình hình chung của nền kinh tế Việt Nam, đi sâu vào nhu cầu của các cá nhân, doanh nghiệp tôi nhận thấy chuyển đổi việc quản lý các kho hàng từ cách truyền thống (sổ sách hoặc các phần mềm chưa chuyên dụng như excel) sang một phần mềm quản lý là một vấn đề cấp thiết. Vì vậy tôi đã thiết kế một phần mềm chuyên dụng giúp việc quản lý kho trở nên nhanh hơn, gọn gàng, linh động, người dùng với chiếc laptop và kết nối internet có thể quản lý kho của mình ở bất cứ đâu.

## 1.3. Cấu trúc đồ án.

### 1.3.1. Chương 1: Giới thiệu đề tài.

* Thực trạng sử dụng internet và chuyển đổi số ở Việt Nam.
* Lý do hình thành đề tài.
* Cấu trúc đồ án.

### 1.3.2. Chương 2: Cơ sở lý thuyết.

* Ngôn ngữ, thư viện: JavaScript, ReactJS, React-Redux, React-Router,Axios, HTLM, MySQL, ExpressJS,…
* Thiết kế giao diện: CSS, Material UI.
* API hỗ trợ: Nodemailer

### 1.3.3. Chương 3: Kết quả thực nghiệm

Mô tả website, các kết quả nghiên cứu lý thuyết, thực nghiệp đã đạt được

### 1.3.4. Chương 4: Kết luận và kiến nghị

Những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất mới và kiến nghị.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Ngôn ngữ JavaScript.

### 2.1.1. Định nghĩa JavaScript.



Hình ảnh 2. 1. Javascript

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 thành phần chính của lập trình web:

* HTML: giúp thêm nội dung cho trang web
* CSS: Định dạng thiết kế, bố cục, phong cách, canh lề của trang web
* JavaScript: Cải thiện cách hoạt động của trang web

JavaScript được tạo trong mười ngày bởi Brandan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thật sự trở thành JavaScript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.

Trong năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lục đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web. Nếu bạn đang dùng internet, vậy chắc chắn bạn đã từng sử dụng JavaScript rồi.

### 2.1.2. Ưu điểm của JavaScript.

JavaScript có rất nhiều ưu điểm khiến nó vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

* Bạn không cần một compiler vì web browser có thể biên dịch nó bằng HTML.
* Nó dễ học hơn các ngôn ngữ lập trình khác.
* Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn.
* Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới.
* JS hoạt động trên nhiều trình duyệt, nền tảng,…
* Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database.
* Nó giúp website tương tác tốt hơn với khách truy cập.
* Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

### 2.1.3. Khuyết điểm của JavaScript.

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần là vì ngôn ngữ đó khi phát triển đến một mức độ như JavaScript, nó cũng sẽ thu hút lượng lớn hacker, scammer, và những người có ác tâm luôn tìm kiếm những lỗ hổng và các lỗi bảo mật để lợi dụng nó. Một số khuyết điểm có thể kể đến là:

* Dễ bị khai thác;
* Có thể được dùng để thực thi mã độc trên máy tính của người dùng;
* Nhiều khi không được hỗ trợ trên mọi trình duyệt;
* JS code snippets lớn;
* Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

### 2.1.4. React.



Hình ảnh 2. 2. React

#### Định nghĩa:

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.

Components của công cụ này được phát triển bởi Facebook. Nó được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất thời bấy giờ.

#### Ưu điểm của React:

Dễ sử dụng: React là một thư viện GUI mã nguồn mở JavaScript tập trung vào một điều cụ thể; hoàn thành nhiệm vụ UI hiệu quả. Nó được phân loại thành kiểu “V” trong mô hình MVC. Là lập trình viên JavaScript, bạn sẽ dễ dàng hiểu được những điều cơ bản về React. Bạn thậm chí có thể bắt đầu phát triển các ứng dụng dựa trên web bằng cách sử dụng react chỉ trong vài ngày.

Hỗ trợ Reusable Component trong java: React cho phép bạn sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng. Tính năng tái sử dụng component là một lợi thế khác biệt cho các lập trình viên.

Viết component dễ dàng: React component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép bạn kết hợp HTML với JavaScript. JSX là một sự pha trộn tuyệt vời của JavaScript và HTML. Nó làm rõ toàn bộ quá trình viết cấu trúc trang web. Ngoài ra, phần mở rộng cũng giúp render nhiều lựa chọn dễ dàng hơn. JSX có thể không là phần mở rộng cú pháp phổ biến nhất, nhưng nó được chứng minh là hiệu quả trong việc phát triển components đặc biệt hoặc các ứng dụng có khối lượng lớn.

Hiệu suất tốt hơn với Virtual DOM: React sẽ cập nhật hiệu quả quá trình DOM. Như bạn có thể biết, quá trình này có thể gây ra nhiều thất vọng trong các dự án ứng dụng dựa trên web. May mắn là React sử dụng virtual DOMs, vì vậy bạn có thể tránh được vấn đề này. Công cụ cho phép bạn xây dựng các virtual DOMs và host chúng trong bộ nhớ. Nhờ vậy, mỗi khi có sự thay đổi trong DOM thực tế, thì virtual sẽ thay đổi ngay lập tức. Hệ thống này sẽ ngăn DOM thực tế để buộc các bản cập nhật được liên tục. Do đó, tốc độ của ứng dụng sẽ không bị gián đoạn.

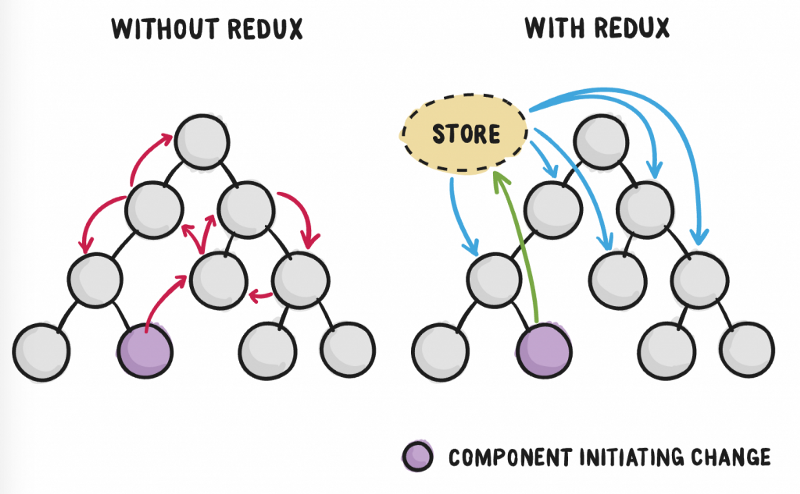
Thân thiện với SEO: React cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau. Tính năng này là một lợi thế rất lớn vì không phải tất cả các khung JavaScript đều thân thiện với SEO.

### 2.1.5. React redux.



Hình ảnh 2. 3. Redux

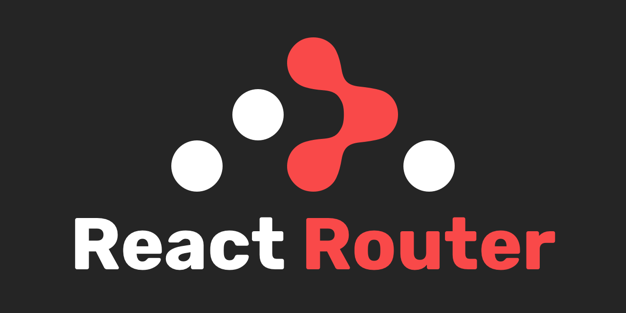
Redux là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ ELM và kiến trúc FLUX của Facebook.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



Hình ảnh 2. 4. Redux

Redux Thunk là một Middleware cho phép bạn viết các Action trả về một function thay vì một plain javascript object bằng cách trì hoãn việc đưa action đến reducer. 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.

### 2.1.6. React Router



Hình ảnh 2. 5. React Router

#### Định nghĩa:

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

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.

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 đương render trên giao diện.

Link: trong HTLM thì cặp thẻ để chuyển hướng đó là thẻ <a> <a/> thì trong react chúng ta sử dụng cặp thẻ <Link></Link> được import từ React-Router.

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

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.

Đối tượng prompt – xác nhận trước khi chuyển trang: Giả sử khi bạn đang nhập liệu ở form nào đó mà không may click nút back hay chuyển trang thì thôi xong dữ liệu bạn nhập sẽ mất hết để khác phục điều đó ta có đối tượng prompt nó sẽ giúp chúng ta trước khi back hay chuyển trang nó sẽ xác nhận xem là chúng ta có chắc chắn muốn back hay chuyển trang không!

Redirect: Chức năng dùng để chuyển trang, có thể truy xuất thông tin trang trước đó thông qua đối tượng location.

### 2.1.7. Axios



Hình ảnh 2. 6. Axios

#### Định nghĩa

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.

#### Transforms

Axios cho phép bạn cung cấp các function để biến đổi dữ liệu đến và đi, dưới dạng 2 config option mà bạn có thể set khi thực hiện một request: transformRequest và transformResponse. Cả 2 thuộc tính đều là các mảng, giúp bạn có thể tạo một luồng các function mà dữ liệu có thể đi qua. Bất kỳ function nào được truyền vào transformRequest đều được ứng dụng vào cho các request PUT, POST và PATCH. Chúng nhận vào dữ liệu cần request và object header dưới dạng tham số và phải trả về một object đã được chỉnh sửa thay đổi.

Các function cũng có thể được thêm vào transformResponse tương tự như vậy, nhưng chỉ được gọi khi có response data từ server, và trước khi response được truyền qua bất cứ hàm then() nào.

Vậy ta có thể sử dụng transform vào mục đích gì? Một trường hợp sử dụng đó là khi API mong chờ dữ liệu trả về dưới format cụ thể khác như XML hay kể cả CSV. Bạn có thể set up một vài hàm transform để convert dữ liệu gửi đi và trả về. Mặc định, hàm transformRequest và transformResponse biến đổi dữ liệu thành JSON.

#### Interceptors

Trong khi transform cho phép bạn điều chỉnh dữ liệu gửi đi và trả về, **Axios** cũng cho phép bạn thêm các function gọi là interceptors. Giống transform, những function này có thể được đính vào để gọi đến khi một request được tạo ra, hay một response được trả về.

#### Tổng kết

Có rất nhiều lý do để ta chọn sử dụng Axios. Thư viện này có một API rõ ràng, với các method hữu dụng quen thuộc với bất kì ai đã từng sử dụng jQuery trước đây. Sự phổ biến của nó và việc có sẵn các thư viện bên thứ 3 khiến Axios trở thành lựa chọn chắc chắn cho ứng dụng của bạn cho dù là front-end hay back-end hay cả hai.

## 2.2. HTML 5



Hình ảnh 2. 7. HTML5

HTML là chữ viết tắt của Hypertext Markup Language. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, vâng vâng.

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản <p> và </p>.

Tổng quan, HTML là ngôn ngữ markup, nó rất trực tiếp dễ hiểu, dễ học, và tất cả mọi người mới đều có thể bắt đầu học nó để xây dựng website.

## 2.3. CSS

### 2.3.1. Định nghĩa CSS



Hình ảnh 2. 8. CSS

Cascading Style Sheets, hay còn gọi tắt là CSS, là những file hỗ trợ cho trình duyệt Web trong việc hiển thị một trang HTML. Về cơ bản, các trang HTML như bộ xương – khuôn khổ cơ bản – của một trang web – trong khi các file CSS sẽ cụ thể hóa các thành phần của một trang nên được hiển thị như thế nào. CSS cho phép bạn kiểm soát phông chữ, màu chữ, kiểu nền…, của một trang HTML.

### 2.3.2. Ưu điểm CSS

CSS làm cho việc thiết kế và xây dựng một trang Web trở nên dễ dàng hơn. Trước khi có CSS, bạn phải sử dụng HTML để thực hiện tất cả các tiêu đề cũng như các thành phần của một trang Web. Mặc dù đã có một số cách khác nhưng HTML vẫn thường xuyên được sử dụng và các nhà thiết kế Web sẽ phải thay đổi từng thành phần riêng lẻ trên mỗi trang. CSS đã giúp cho việc thay đổi phong cách của một trang Web đơn giản hơn và đỡ tốn thời gian hơn, bạn có thể thực hiện sự thay đổi trên một mục nào đó và sau đó apply sự thay đổi này trên toàn bộ trang Web thay vì phải thực hiện trên từng mục riêng lẻ như trước đây.

## 2.4. ExpressJS

### 2.4.1. ExpressJS là gì



Hình ảnh 2. 9. ExpressJS

Express là một framework giành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ rợ các phương thức HTTP và midleware tạo ra môt API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:

* Thiết lập các lớp trung gian để trả về các HTTP request
* Định nghĩa router cho phép 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.

## 2.5. MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

## 2.6. SQL

MySQL và SQL không giống nhau.MySQL là một trong các phần mềm RDBMS, hoạt động theo mô hình client-server. Nhưng, làm thế nào client và server liên lạc với nhau trong môi trường của RDBMS? Chúng sử dụng ngôn ngữ truy vấn có cấu trúc chung – Structured Query Language (SQL). Nếu bắt gặp một thương hiệu nào đó có SQL kèm theo, như PostgreSQL, Microsoft SQL server, chúng thường là những thương hiệu sử dụng syntax của SQL. RDBMS software lại được viết bằng ngôn ngữ lập trình, nhưng luôn sử dụng SQL làm ngôn ngữ chính để tương tác với database. Bản thân MySQL được viết bằng C và C++. Hãy tưởng tượng các nước Nam Mỹ, chúng có vị trí địa lý và lịch sử khác nhau, nhưng đều nói tiếng Tây Ban Nha.

Nhà khoa học máy tính Ted Codd tạo ra SQL vào đầu thập niên 1970s với IBM theo mô hình liên hệ. Nó được sử dụng rộng khắp vào năm 1974 và nhanh chóng thay thế các ngôn ngữ lỗi thời khác như ISAM và VISAM. SQL nói server phải làm gì với dữ liệu. Giống khi bạn nhập mật khẩu để đăng nhập vào trang quản lý của WordPress, trong trường hợp này, lệnh SQL sẽ nói cho server biết cần phải thực hiện một số hành động:

* Truy vấn dữ liệu – data query: yêu cầu một thông tin cụ thể trong database có sẵn.
* Điều hành dữ liệu – data manipulation: thêm, xóa, sửa, sắp xếp, và các tác vụ điều hành khác để chỉnh sửa dữ liệu, giá trị, và cách hiển thị.
* Loại dữ liệu – data identity: định nghĩa loại dữ liệu, như là đổi từ dữ liệu số thành số nguyên. Nó cũng bao gồm định nghĩa một schema hoặc định nghĩa mối quan hệ của từng bảng trong cơ sở dữ liệu
* Quyền truy cập data – data access control: cung cấp biện pháp bảo mật để bảo vệ dữ liệu, bao gồm việc quyết định ai có thể xem thông tin dữ liệu nào trong database

## 2.5. Công cụ quản lý phiên bản

### 2.5.1. Github



Hình ảnh 2. 10. Github

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

* Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.
* Giúp đồng bộ source code của team lên 1 server.
* Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, …)

## 2.6. Xác định yêu cầu

### 2.6.1. Yêu cầu hệ thống

* Dễ dàng sử dụng và quản trị các thiết bị và nhân viên
* Phục vụ công tác quản lý kho trở nên tự động, chính xác và nhanh chóng hơn
* Thông qua phần mềm, các hoạt động nhập xuất tồn sẽ được thể hiện chi tiết và cụ thể.
* Nhà quản lý có thể nắm bắt được các thông tin hoạt động kho ở mọi lúc, mọi nơi
* Quản lý cùng lúc nhiều loại hàng hóa một cách khoa học

### 2.6.2. Yêu cầu chức năng

* Quản trị danh sách thiết bị
* Quản trị loại thiết bị
* Quản trị danh mục thiết bị
* Quản trị nhân viên
* Quản trị và xử lý yêu cầu duyệt từ nhân viên
* Tự động hoặc bán tự động gửi email cho nhân viên những thiết bị gần hết hạn sử dụng

## 2.7. Phân tích hệ thống

### 2.7.1. Phân tích chức năng website

#### Quản trị danh mục thiết bị:

Đặc trưng nổi bật trong tính năng quản trị thiết bị của Stock-Manager là hỗ trợ phân loại thiết bị rành mạch, rõ ràng giúp người dùng có thể nhanh chóng tìm kiếm các thiết bị mong muốn.

Phân biệt rành mạch từng thiết bị, trạng thái, vị trí,… sắp xếp thiết bị theo đặc trưng.

Bao gồm các hành động: thêm, xoá, sửa, sắp xếp danh sách thiết bị theo mã, tên, trạng thái, ngày nhập, ngày xuất, ngày hết hạn, tăng dần hoặc giảm dần các loại thuộc tính.

#### Quản trị loại thiết bị:

Quản trị danh sách các loại thiết bị: cho phép hiển thị danh sách loại thiết bị, có thể chỉnh sửa tên, danh mục, đơn vị, mô tả.

Bao gồm các hành động: thêm, xoá, sửa, sắp xếp danh sách hiển thị theo mã, tên, danh mục, tăng dần hoặc giảm dần các loại thuộc tính.

#### Quản trị danh mục thiết bị:

Quản trị danh sách danh mục thiết bị: hiển thị, thêm, xoá, sửa từng danh mục.

Bao gồm các hành động: thêm, xoá, sửa.

#### Quản trị tài khoản nhân viên:

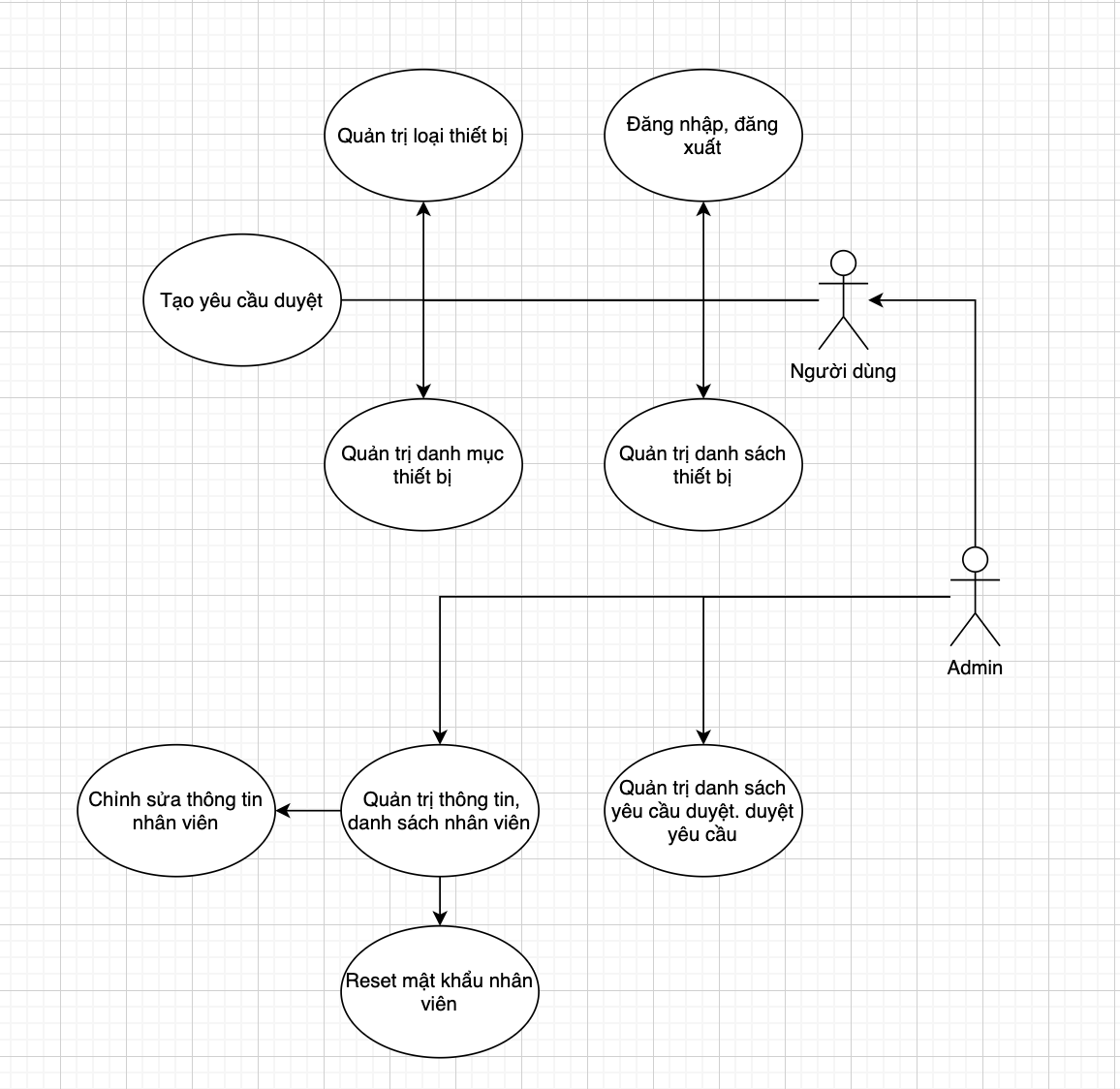
Cho phép admin có thể quản lý danh sách nhân viên.

Bao gồm các hành động: thêm, xoá, sửa tài khoản nhân viên, admin có thể đặt lại mật khẩu cho nhân viên, mật khẩu mới sẽ được tự động gửi qua email cho nhân viên.

#### Quản trị yêu cầu duyệt:

Tài khoản loại nhân viên có thể tạo một yêu cầu để thay đổi trạng thái của thiết bị

Tài khoản của admin có thể xem và duyệt yêu cầu, sau khi duyệt, trạng thái của thiết bị sẽ được thay đổi.



Sơ đồ 2. 1. Database Diagram

### 2.7.2. Chi tiết chức năng website

Bảng 2. 1. Mô tả chi tiết chức năng website phía người dùng

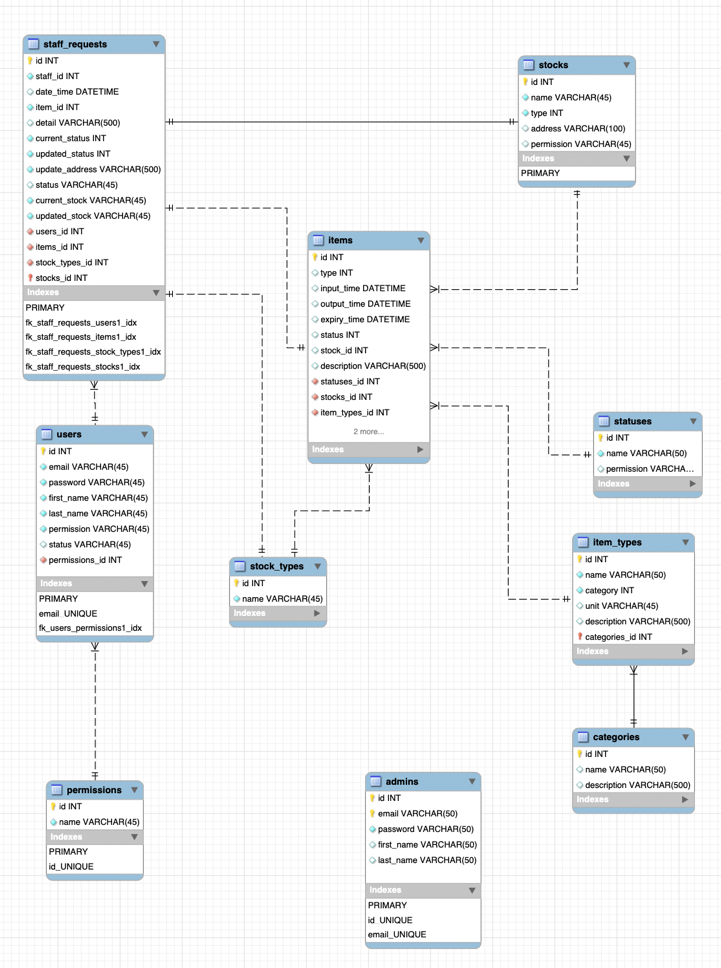
| Stt | Chức năng | Mô tả |
| --- | --- | --- |
| 1 | Xem danh sách thiết bị | Người dùng có thể xem mã, tên, trạng thái, vị trí, ngày nhập, ngày xuất, ngày hết hạn, mô tả, sắp xếp danh sách thiết bị theo các thuộc tính như: mã, tên, trạng thái, ngày nhập, ngày xuất, ngày hết hạn theo thứ tự tăng hoặc giảm dần. |
| 2 | Chỉnh sửa thông tin thiết bị | Người dùng có thể chỉnh sửa các thông tin của thiết bị như: Tên, trạng thái, kho, ngày nhập, ngày hết hạn, mô tả (không bao gồm “ngày xuất”).  Các trạng thái có thể chỉnh sửa: mới, đã sử dụng, đang sử dụng, cần thanh lý (không bao gồm “đã thanh lý”) |
| 3 | Thêm thiết bị | Cho phép người dùng thêm mới một thiết bị với các thông tin như: tên, trạng thái, kho, ngày nhập, ngày hết hạn, mô tả (không bao gồm ngày xuất). |
| 4 | Xoá thiết bị | Người dùng có thể xoá thiết bị từ danh sách thiết bị. |
| 5 | Thông báo thiết bị gần hết hạn | Người dùng có thể thông báo những sản phẩm hết hạn vào ngày tiếp theo. Khi nhấn vào nút thông báo, hệ thống sẽ tự động gửi email với nội dung bao gồm thông tin những sản phẩm sẽ hết hạn sử dụng vào ngày tiếp theo vào email của nhân viên. |
| 6 | Xem danh sách loại thiết bị | Người dùng có thể xem mã, tên, danh mục, đơn vị, mô tả, sắp xếp danh sách thiết bị theo các thuộc tính như: mã, tên, danh mục theo thứ tự tăng hoặc giảm dần. |
| 7 | Chỉnh sửa loại thiết bị | Người dùng có thể chỉnh sửa các thông tin của loại thiết bị như: tên, danh mục, đơn vị, mô tả. |
| 8 | Xoá loại thiết bị | Người dùng có thể xoá loại thiết bị từ danh sách loại thiết bị. |
| 9 | Thêm loại thiết bị | Cho phép người dùng thêm mới một loại thiết bị với các thông tin như: tên, danh mục, đơn vị, mô tả. |
| 10 | Xem danh mục thiết bị | Người dùng có thể xem mã, tên, mô tả của từng danh mục. |
| 11 | Chỉnh sửa danh mục thiết bị | Người dùng có thể chỉnh sửa các thông tin của danh mục thiết bị như: tên, mô tả. |
| 12 | Xoá danh mục thiết bị | Người dùng có thể xoá danh mục thiết bị |
| 13 | Xem yêu cầu duyệt | Người dùng có thể xem mã, tên, trạng thái hiện tại, trạng thái sau khi duyệt, kho hiện tại, kho sau khi duyệt, tên nhân viên, trạng thái báo cáo. |
| 14 | Tạo yêu cầu duyệt | Cho phép người dùng thêm mới một loại thiết bị với các thông tin như: loại thiết bị, mã thiết bị, trạng thái hiện tại, kho hiện tại, kho hiện tại, kho sau khi duyệt, địa chỉ sau khi duyệt, chi tiết.  Sau khi người dùng chọn được mã thiết bị, trạng thái hiện tại và kho hiện tại sẽ được tự động hiển thị. |

Tài khoản có chức năng quản trị viên có toàn bộ quyền truy cập vào các chức năng của tài khoản nhân viên và có thêm các chức năng như mô tả ở bảng dưới.

Bảng 2. 2. Mô tả chi tiết chức năng phía người quản lý (admin)

| Stt | Chức năng | Mô tả |
| --- | --- | --- |
| 1 | Chỉnh sửa thông tin thiết bị | Admin được chỉnh sửa tất cả các thông tin của thiết bị (Không bao gồm ngày xuất) |
| 2 | Xem danh sách nhân viên | Admin có thể xem danh sách nhân viên bao gồm các thông tin như: mã, họ và tên, email, chức năng, trạng thái |
| 3 | Chỉnh sửa thông tin nhân viên | Admin có thể chỉnh sửa các thông tin của nhân viên: email, họ, tên, chức năng, trạng thái (không bao gồm mật khẩu) |
| 4 | Xoá tài khoản | Admin có thể xoá tài khoản của nhân viên ra khỏi danh sách nhân viên. Tài khoản sau khi bị khoá sẽ không thể đăng nhập vào hệ thống. |
| 5 | Đặt lại mật khẩu | Admin có thể đặt lại mật khẩu của nhân viên, tài khoản được đặt lại mật khẩu sẽ tự động cập nhật mật khẩu mới, mật khẩu mới sẽ được tự động gửi cho nhân viên thông qua email, admin sẽ không biết được mật khẩu của nhân viên. |
| 6 | Duyệt yêu cầu duyệt | Đối với những yêu cầu có trạng thái “Đang chờ duyệt” admin có thể “Duyệt” hoặc “Không duyệt”, sau khi duyệt yêu cầu, trạng thái của thiết bị sẽ được thay đổi hoặc không (nếu duyệt trạng thái của thiết bị được duyệt sẽ thay đổi) admin sẽ không thể thay đổi trạng thái của yêu cầu đó nữa. |

### 2.7.3. Phân tích cơ sở dữ liệu



Sơ đồ 2. 2. Cơ sở dữ liệu quan hệ

Bảng 2. 3. Ký hiệu các ràng buộc

| **Ràng buộc** | **Ý nghĩa** |
| --- | --- |
| NOT NULL | Đảm bảo dữ liệu của cột không được nhận giá trị NULL. |
| DEFAULT | Cung cấp một giá trị mặc định cho một cột trong trường hợp dữ liệu của cột không được nhập vào hay không được xác định. |
| UNIQUE | Đảm bảo rằng dữ liệu của cột là duy nhất, tất cả các giá trị trong một cột là khác nhau không trùng lặp. |
| PRIMARY Key | Dùng để thiết lập khóa chính trên bảng, giá trị của cột làm khóa chính phải là duy nhất, không được trùng lặp. Việc khai báo ràng buộc khóa chính yêu cầu các cột phải NOT NULL. |
| FOREIGN Key | Dùng để thiết lập khóa ngoại trên bảng, tham chiếu đến bảng khác thông qua giá trị của cột được liên kết. Giá trị của cột được liên kết phải là duy nhất trong bảng kia. |
| CHECK | Đảm bảo tất cả các giá trị trong một cột thỏa mãn một số điều kiện. |
| INDEX | Dùng để tạo và lấy dữ liệu từ cơ sở dữ liệu một cách nhanh chóng. |

Bảng 2. 4. Mô tả chi tiết đối tượng người quản trị (admins)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | id | INT | Phân biệt các admin | Primary key, not null, unique |
| 2 | email | VARCHAR | Email của admin | Not null, unique |
| 3 | password | VARCHAR | Mật khẩu của admin | Not null |
| 7 | first\_name | VARCHAR | Tên | Not null |
| 8 | last\_name | VARCHAR | Họ | Not null |

Bảng 2. 5. Mô tả chi tiết đối tượng nhân viên (users)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | id | Int | Phân biệt các user | Primary key, not null, unique |
| 2 | email | VARCHAR | Email của admin | Not null, unique |
| 3 | password | VARCHAR | Mật khẩu của admin | Not null |
| 7 | first\_name | VARCHAR | Tên | Not null |
| 8 | last\_name | VARCHAR | Họ | Not null |
| 9 | permission | VARCHAR | Quyền hạn | Not null |
| 10 | status | VARCHAR | Trạng thái | Default: “active” |

Bảng 2. 6. Mô tả chi tiết đối tượng thiết bị (items)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | id | INT | Phân biệt các thiết bị | Primary key, not null, unique |
| 2 | type | INT | Loại thiết bị | Foreign key |
| 3 | input\_time | DATETIME | Ngày nhập |  |
| 4 | output\_time | DATETIME | Ngày xuất |  |
| 5 | expiry\_time | DATETIME | Ngày hết hạn |  |
| 6 | status | INT | Tình trạng thiết bị | Foreign key |
| 7 | stock\_id | INT | Kho | Foreign key |
| 8 | description | VARCHAR(500) | Mô tả thiết bị |  |

Bảng 2. 7. Mô tả chi tiết đối tượng loại thiết bị (item\_types)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | Phân biệt các loại thiết bị | Primary key, not null, unique |
| 2 | name | VARCHAR(50) | Tên của loại thiết bị | Not null |
| 3 | category | INT | Danh mục | Foreign key |
| 4 | unit | VARCHAR(45) | Đơn bị | Not null |

Bảng 2. 8. Mô tả chi tiết đối tượng danh mục thiết bị (categories)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | id | INT | Phân biệt các danh mục | Primary key, not null, unique |
| 2 | name | VARCHAR(50) | Tên danh mục | Default: null |
| 3 | description | VARCHAR(500) | Mô tả danh mục | Default: null |

Bảng 2. 9. Mô tả chi tiết đối tượng quyền hạn (permissions)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | Phân biệt các quyền | Primary key, not null, unique |
| 2 | name | VARCHAR(45) | Tên quyền hạn | Not null |

Bảng 2. 10. Mô tả chi tiết đối tượng trạng thái (statuses)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | Phân biệt trạng thái | Primary key, not null, unique |
| 2 | name | VARCHAR(50) | Tên của trạng thái | Not null |
| 3 | permission | VARCHAR(45) | Quyền truy cập | Foreign key |

Bảng 2. 11. Mô tả chi tiết đối tượng kho hàng (stocks)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | Phân biệt các kho | Primary key, not null, unique |
| 2 | name | VARCHAR(45) | Tên của trạng thái | Not null |
| 3 | type | VARCHAR(45) | Loại kho | Foreign key |
| 4 | address | VARCHAR(100) | Địa chỉ của kho |  |
| 5 | permission | VARCHAR(45) | Quyền truy cập | Not null |

Bảng 2. 12. Mô tả chi tiết đối tượng loại kho hàng (stock\_types)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | Phân biệt các loại kho | Primary key, not null, unique |
| 2 | name | VARCHAR(45) | Tên của trạng thái | Not null |

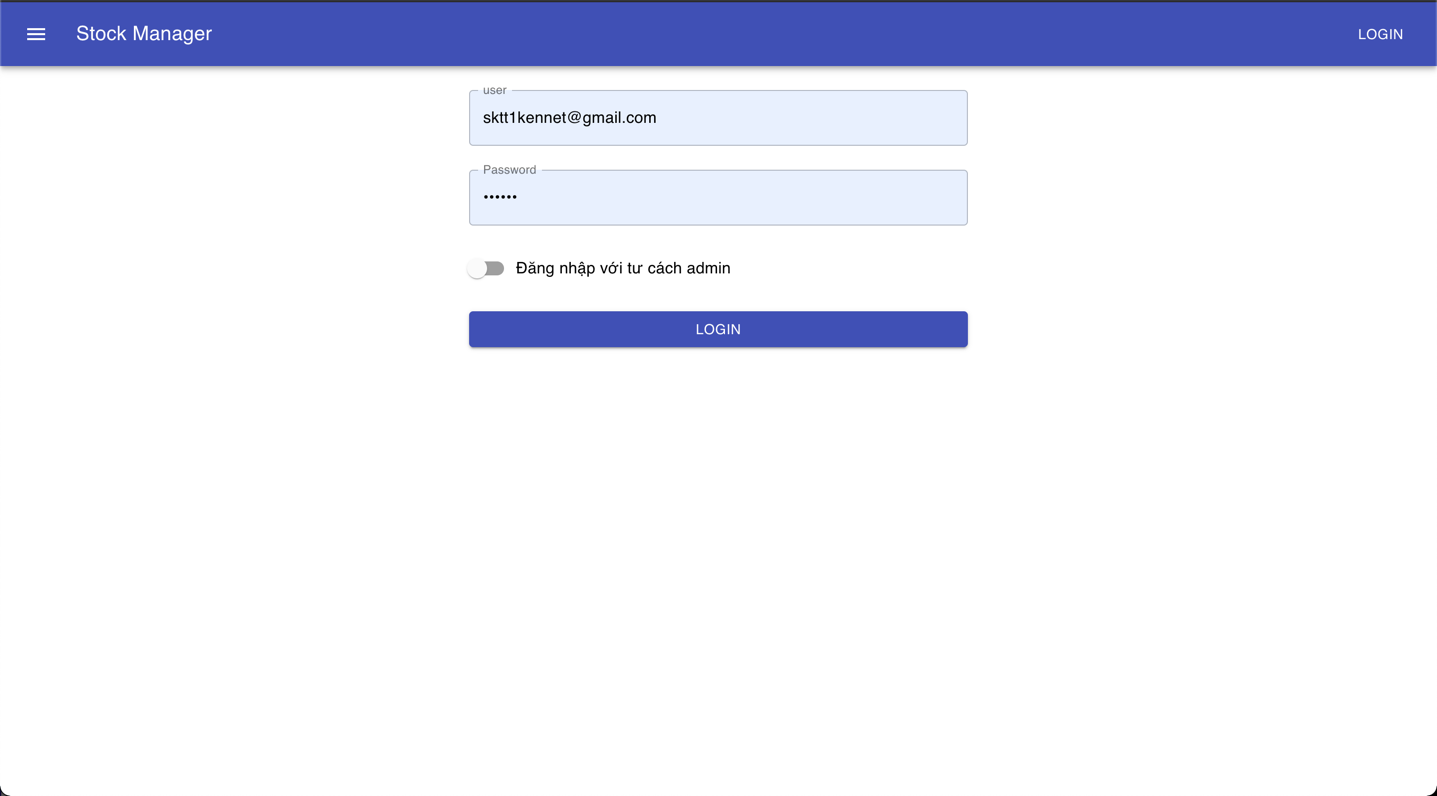
Bảng 2. 13. Mô tả chi tiết đối tượng yêu cầu nhân viên (staff\_requests)

| **Stt** | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| --- | --- | --- | --- | --- |
| 1 | id | INT | Phân biệt các thiết bị | Primary key, not null, unique |
| 2 | staff\_id | INT | Loại thiết bị | Foreign key |
| 3 | date\_time | DATETIME | Ngày nhập |  |
| 4 | item\_id | INT | Ngày xuất |  |
| 5 | detail | VARCHAR(500) | Ngày hết hạn |  |
| 6 | current\_status | INT | Tình trạng thiết bị | Foreign key |
| 7 | updated\_status | INT | Kho | Foreign key |
| 8 | update\_address | VARCHAR(500) | Mô tả thiết bị |  |
| 9 | status | VARCHAR(45) | Trạng thái | Default: “waiting” |
| 10 | current\_stock | VARCHAR(45) | Kho hiện tại | Foreign key |
| 11 | updated\_stock | VARCHAR(45) | Kho sau khi được duyệt | Foreign key |

# CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM

## 3.1. Các thành phần chức năng hệ thống phía nhân viên

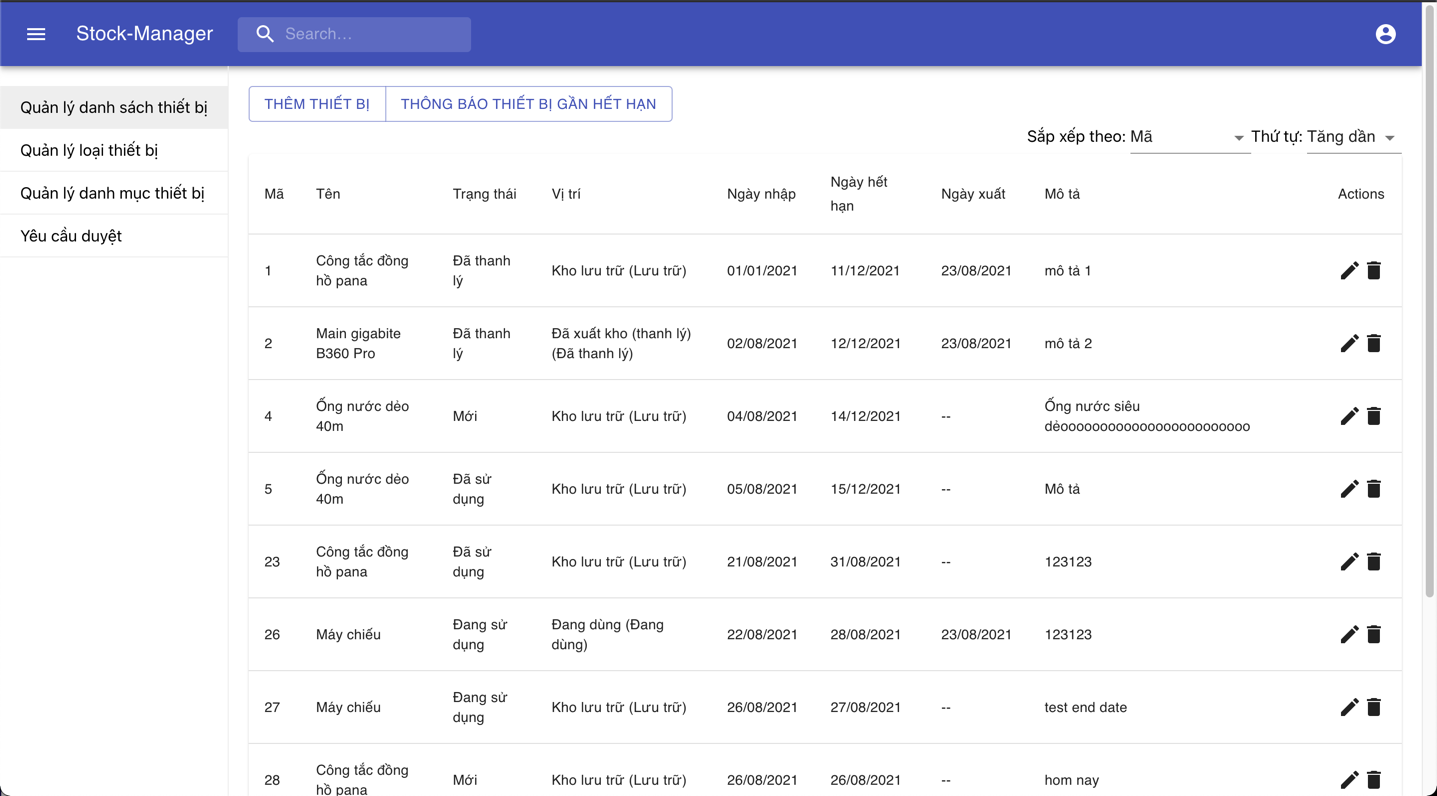
### 3.1.1. Giao diện đăng nhập



Hình ảnh 3. 1. Giao diện đăng nhập

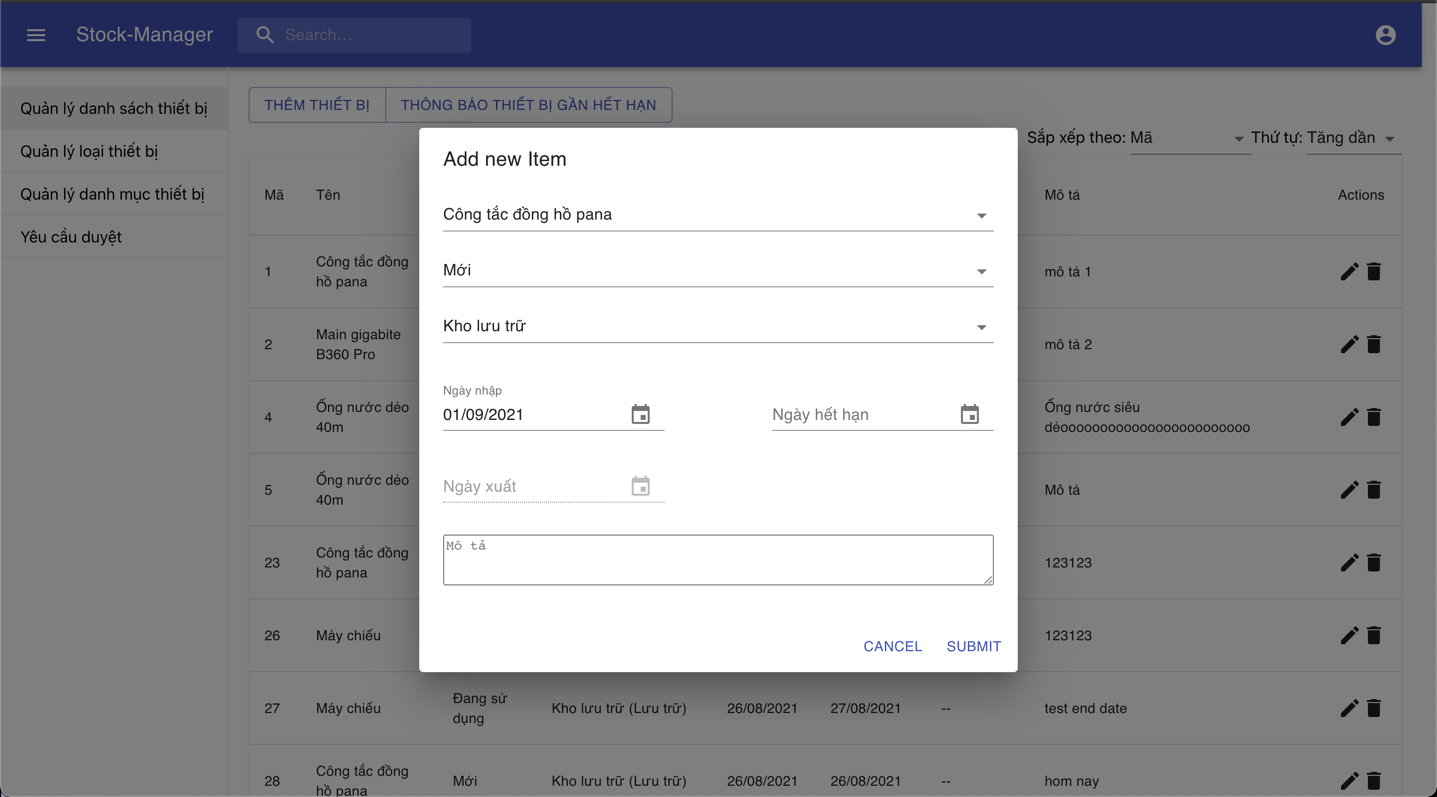
Ở trang đăng nhập, người dùng có thể nhập tài khoản và mật khẩu, nếu muốn đăng nhập bằng tài khoản người quản lý người dùng đánh dấu vào nút “Đăng nhập với tư cách admin”. Nếu tài khoản và mật khẩu đúng, người dùng sẽ được thấy giao diện bảng điều khiển (dashboard).

### 3.1.2. Giao diện bảng điều khiển quản lý danh sách thiết bị



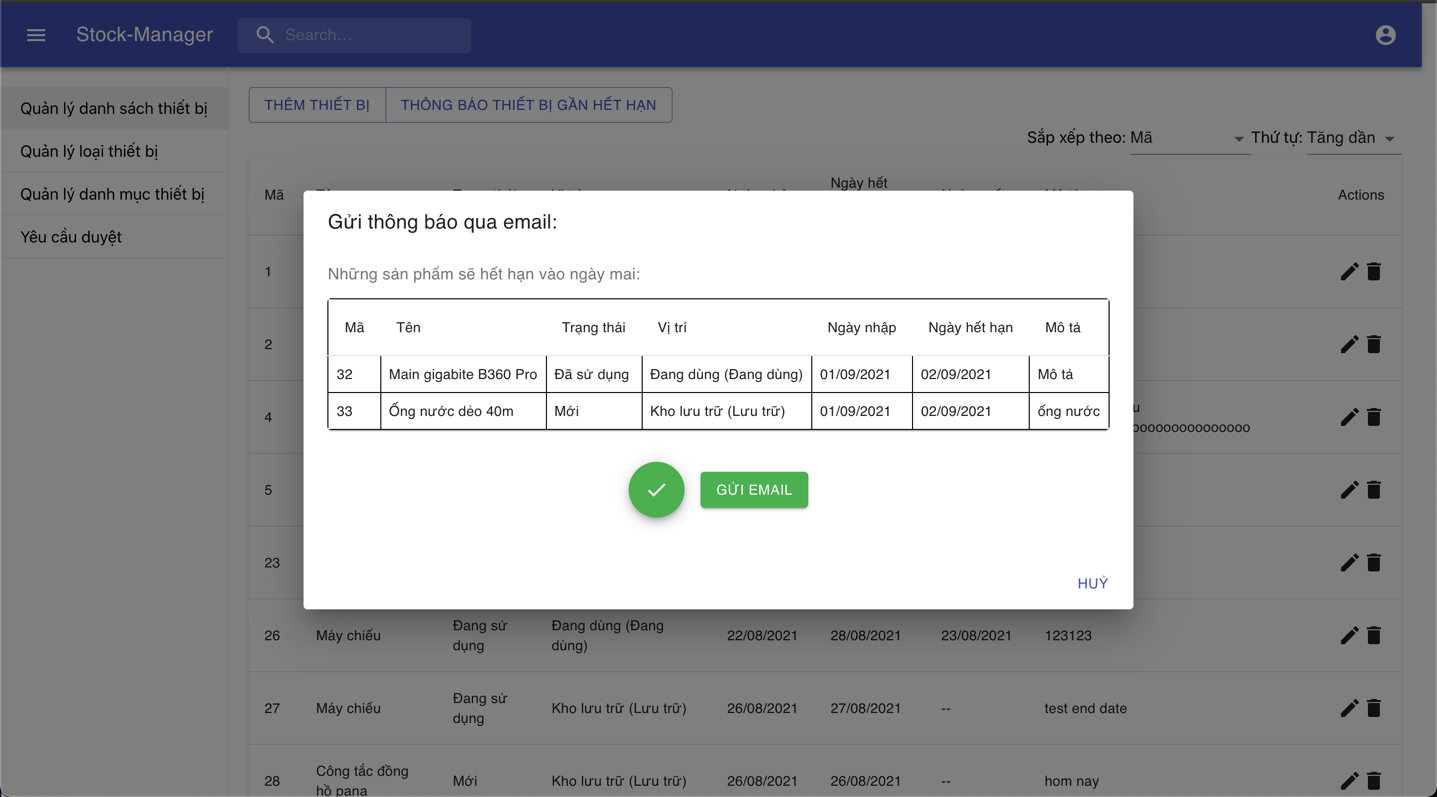
Hình ảnh 3. 2. Giao diện bảng điều khiển danh sách thiết bị

Với các nút điều khiển như: thêm thiết bị, thông báo thiết bị gần hết hạn, và bảng danh sách các thiết bị. Người dùng có thể sắp xếp các thiết bị theo thuộc tính mong muốn.



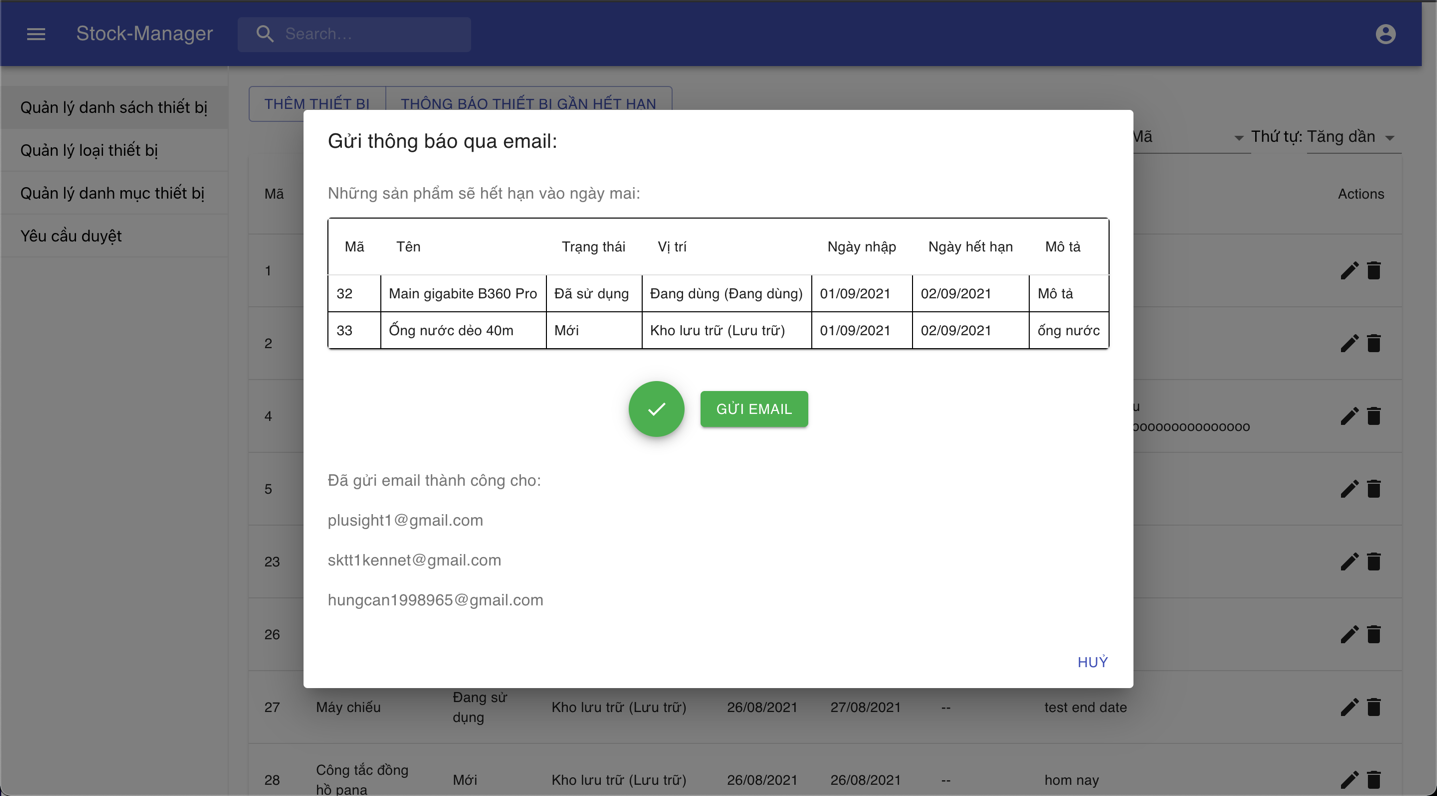
Hình ảnh 3. 3. Giao diện thêm thiết bị

Khi nhấn vào nút “thêm thiết bị”: bảng điều khiển thêm thiết bị xuất hiện người dùng nhập thông tin vào form và nhấn xác nhận để thêm thiết bị.

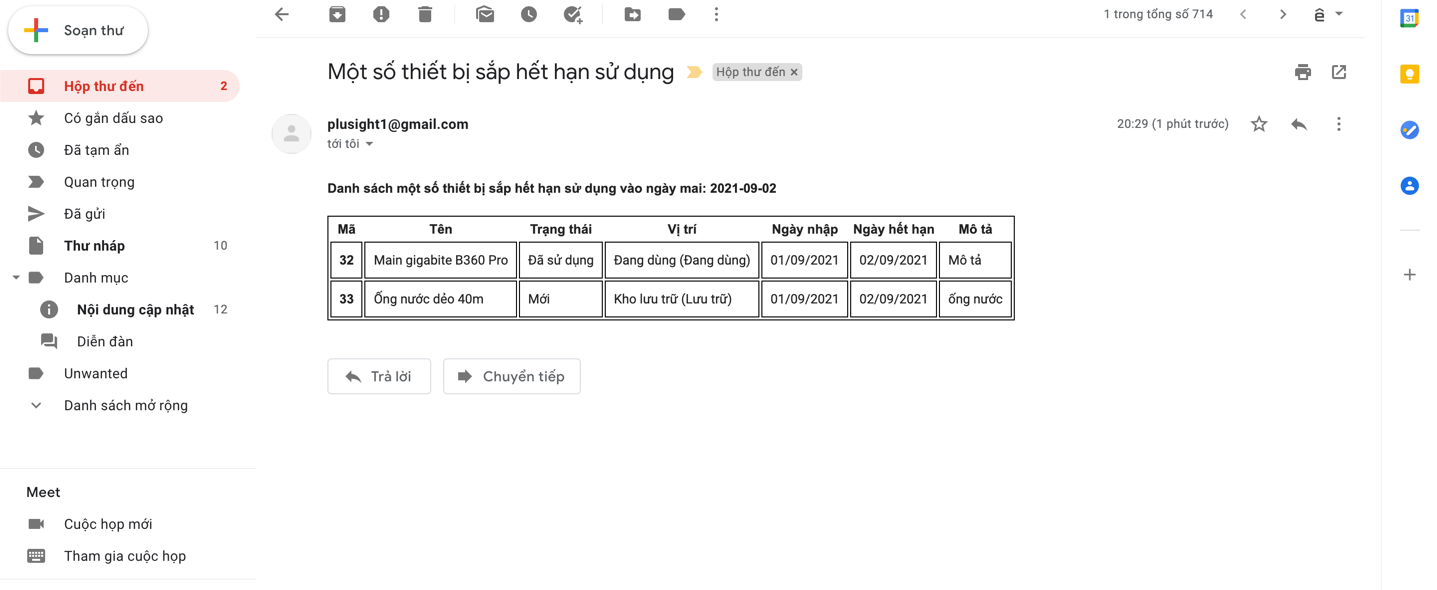


Hình ảnh 3. 4. Giao diện gửi thông báo qua email

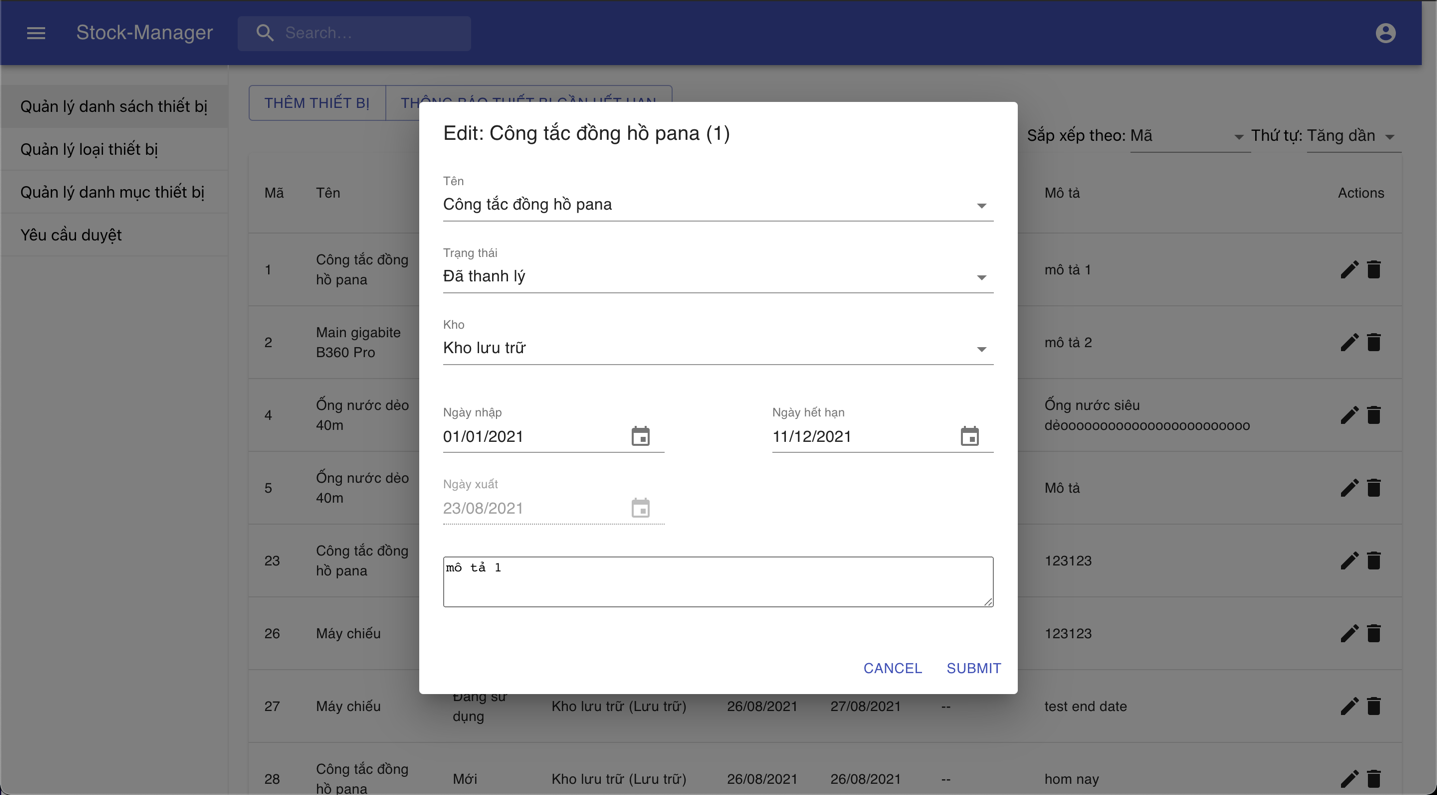
Khi nhấn vào nút “thông báo thiết bị gần hết hạn” danh sách thiết bị hết hạn vào ngày hôm sau sẽ xuất hiện, người dùng nhấn vào nút gửi email danh sách thiết bị sắp hết hạn sẽ được tự động gửi vào email của tất cả nhân viên.



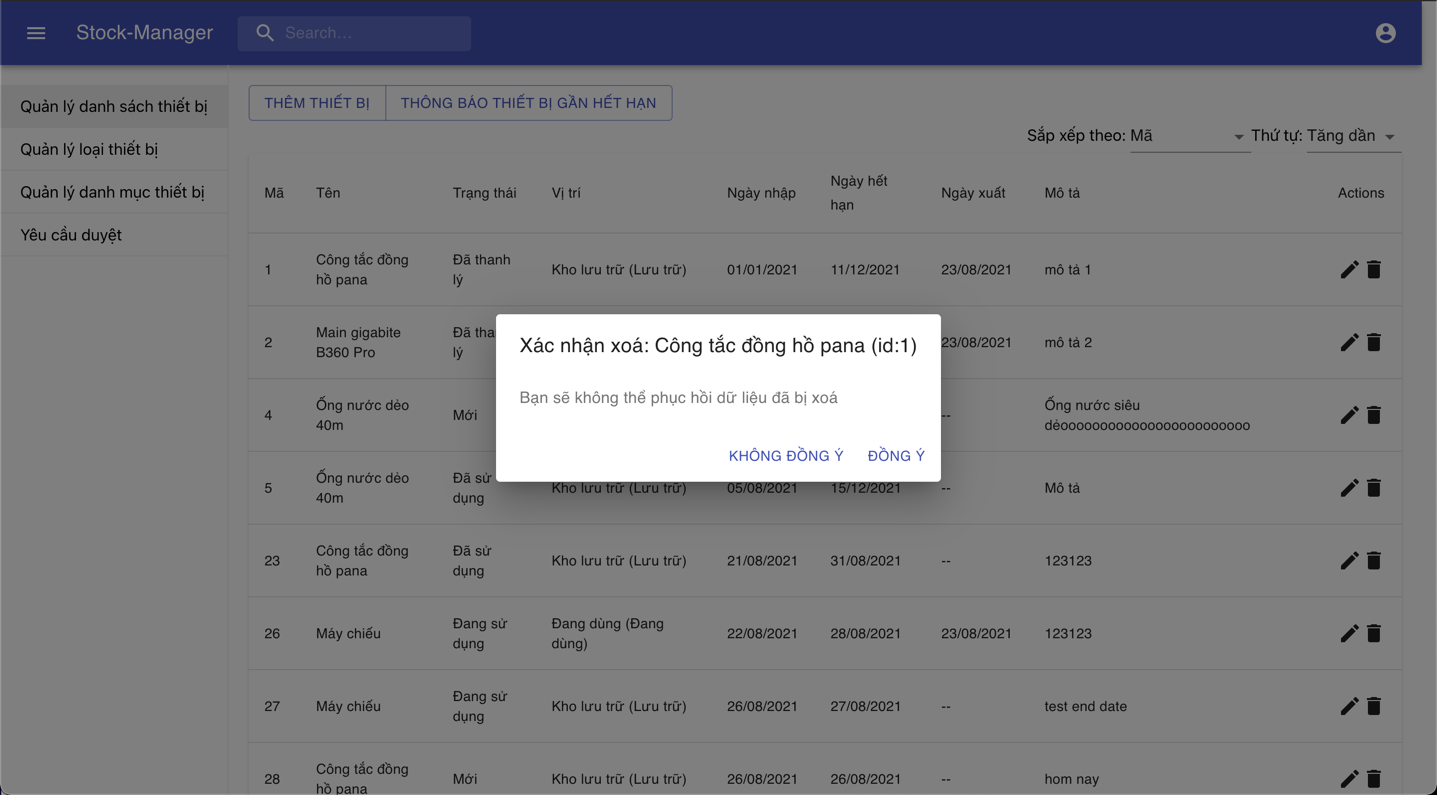
Hình ảnh 3. 5. Giao diện gửi email thành công



Hình ảnh 3. 6. Giao diện nội dung email

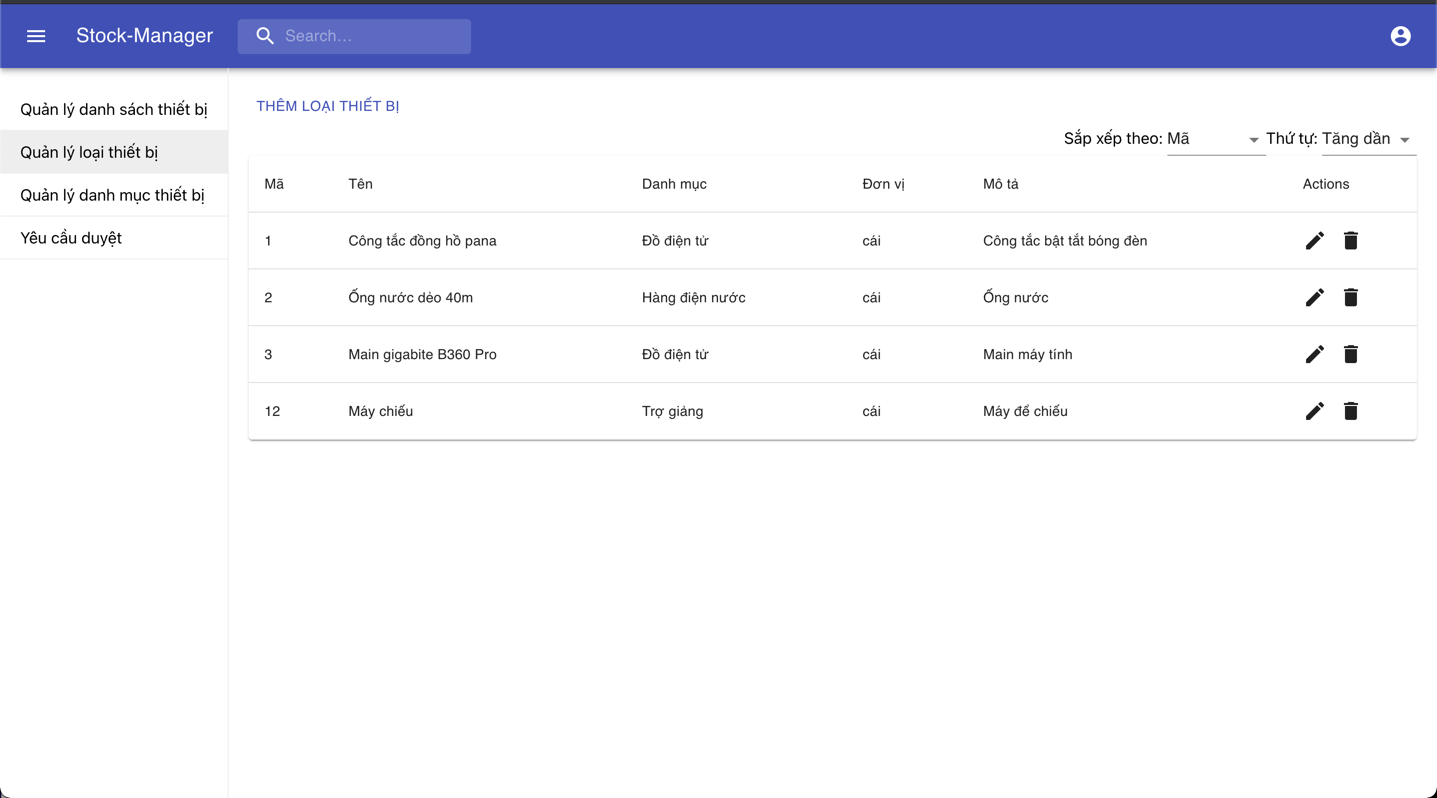


Hình ảnh 3. 7. Giao diện chỉnh sửa thiết bị



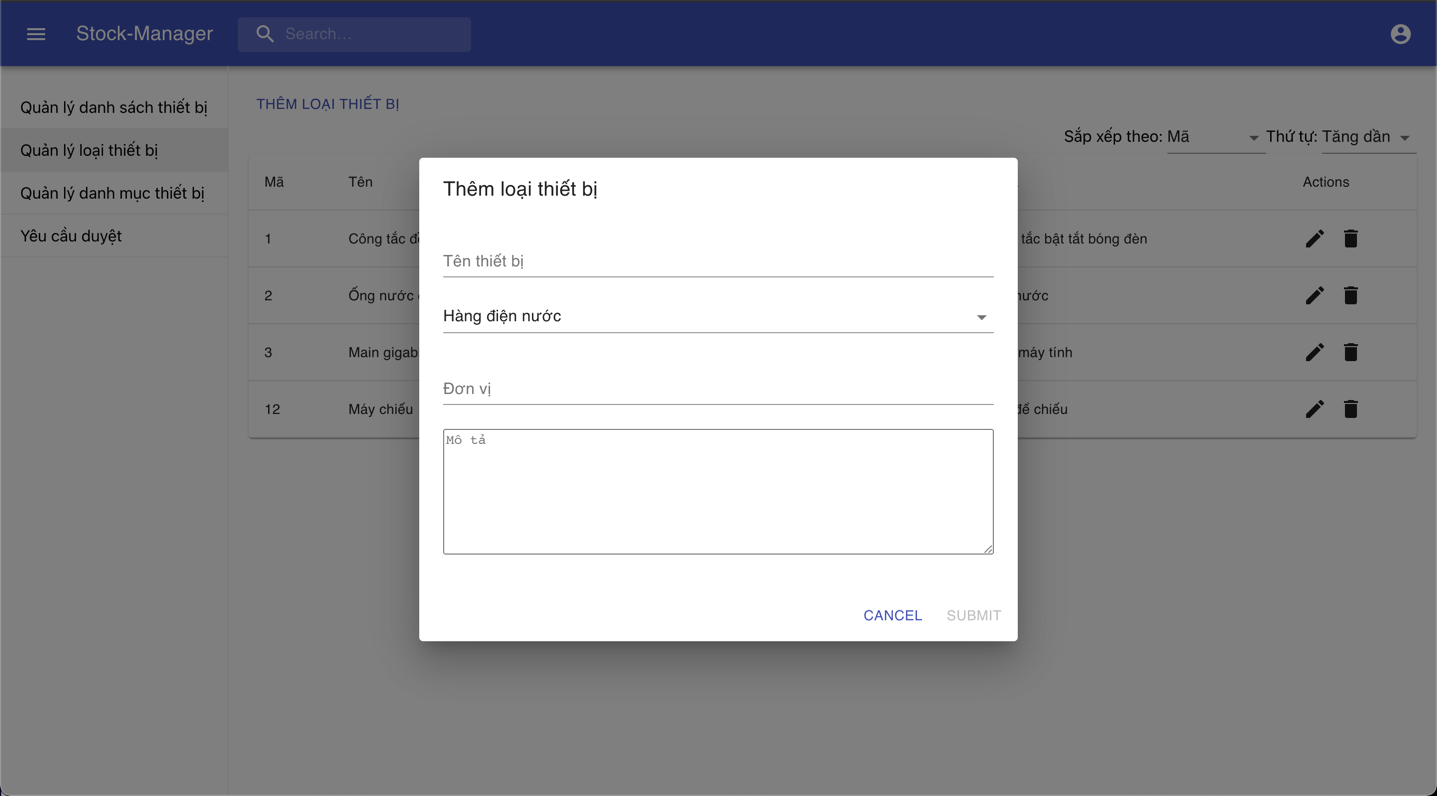
Hình ảnh 3. 8. Giao diện cảnh báo khi xoá thiết bị

### 3.1.3. Giao diện bảng điều khiển quản lý loại thiết bị

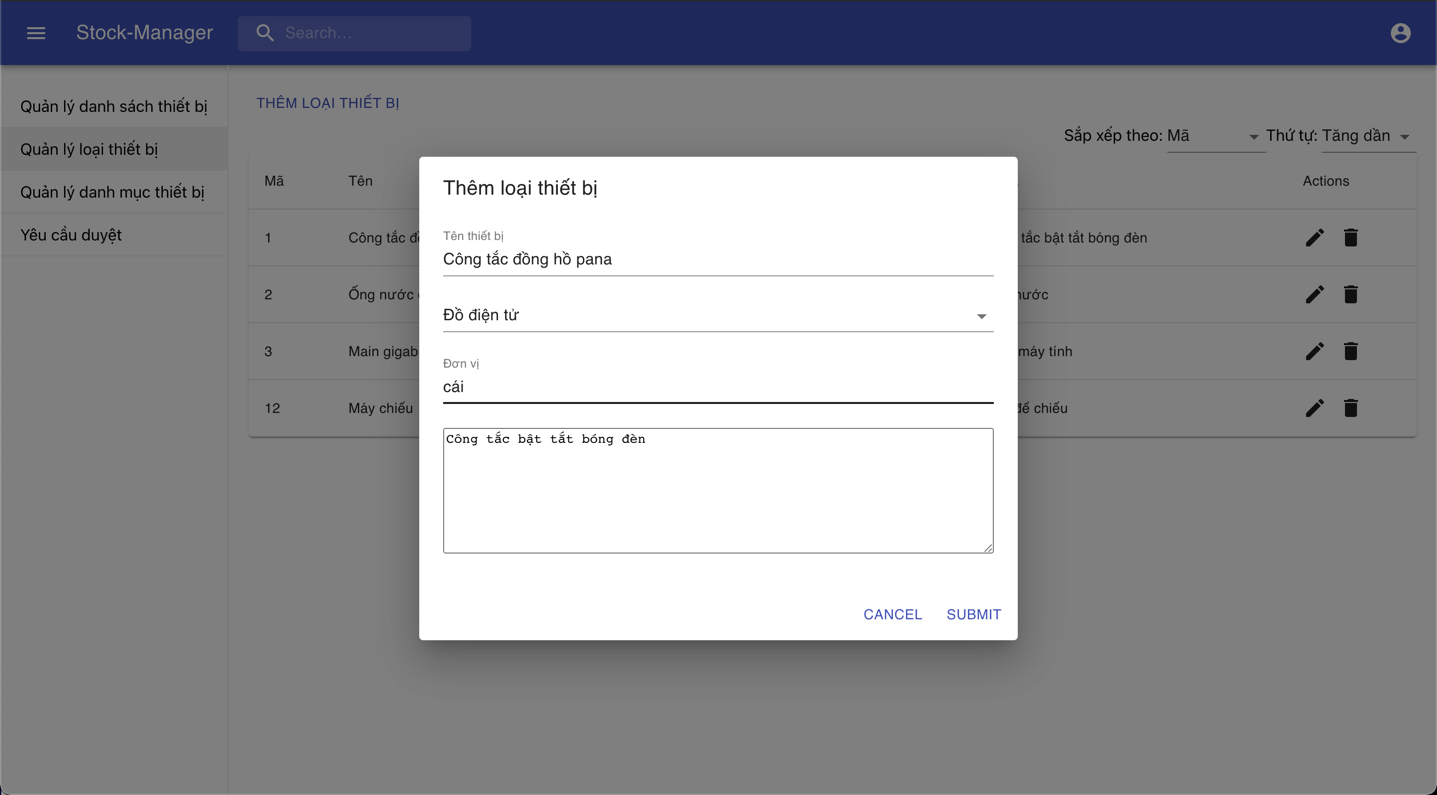


Hình ảnh 3. 9. Giao diện quản lý loại thiết bị

Giao diện quản lý loại thiết bị cho phép người dùng quản lý danh sách các loại thiết bị.

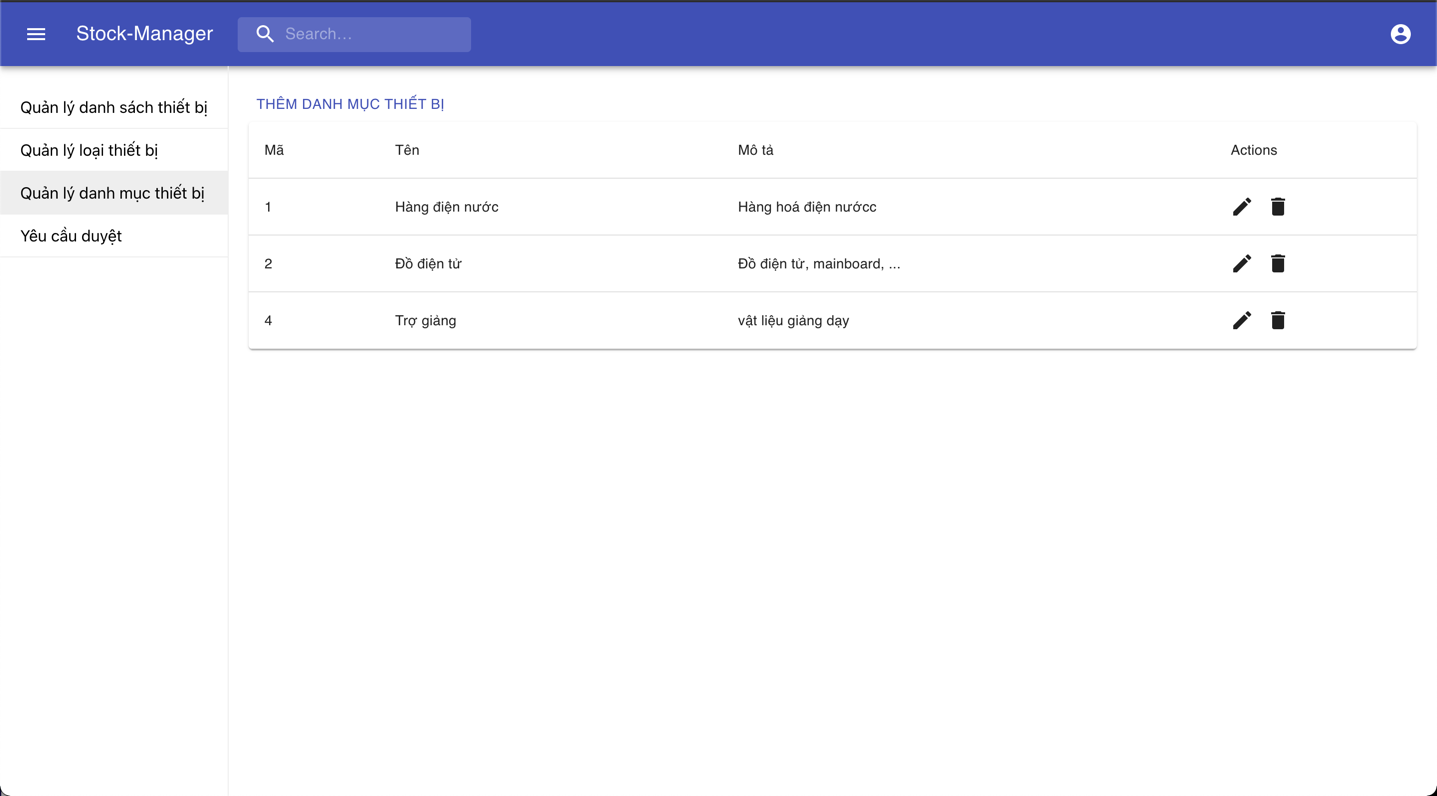


Hình ảnh 3. 10. Giao diện thêm loại thiết bị



Hình ảnh 3. 11. Giao diện chỉnh sửa loại thiết bị

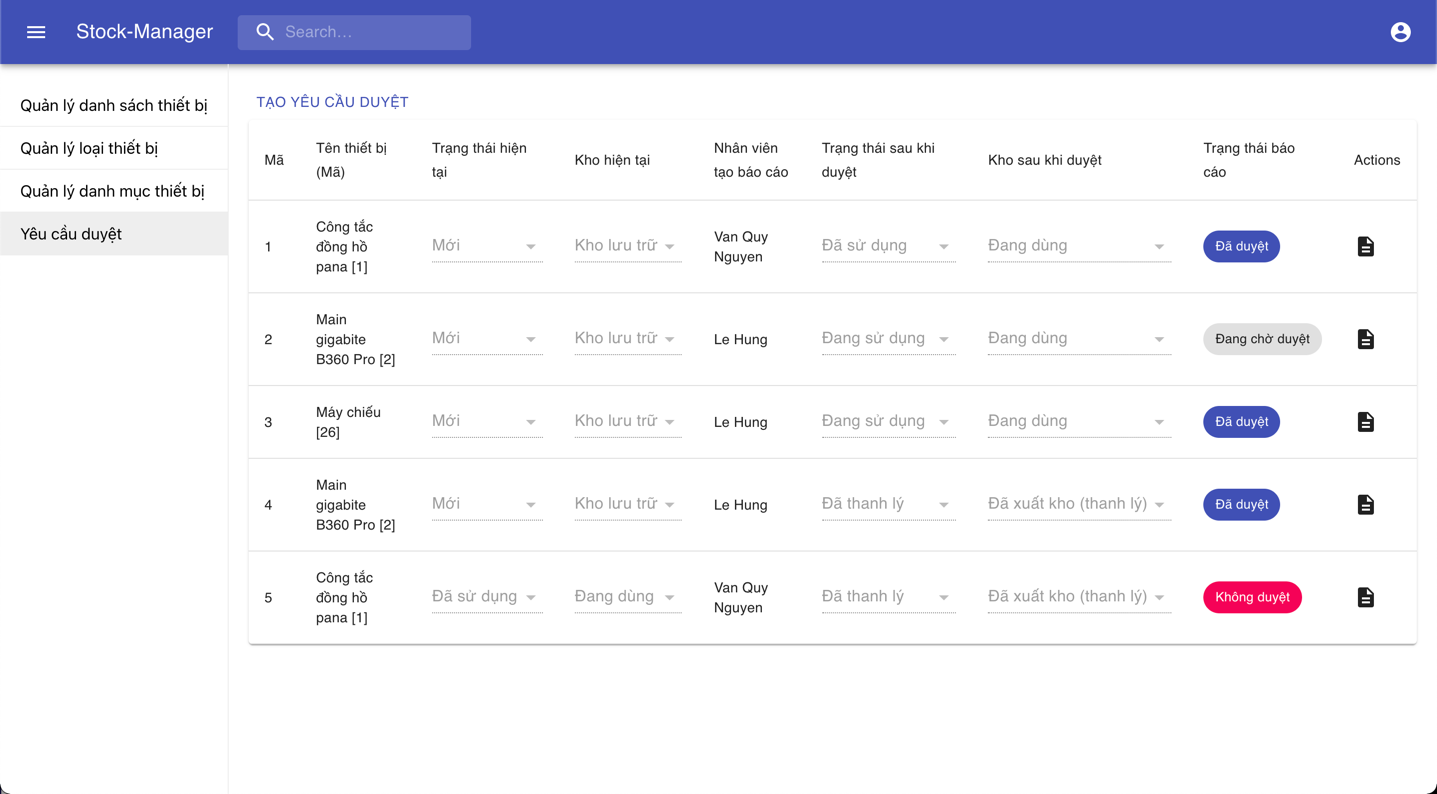
### 3.1.4. Giao diện trang quản lý danh mục thiết bị



Hình ảnh 3. 12. Giao diện trang quản lý danh mục thiết bị

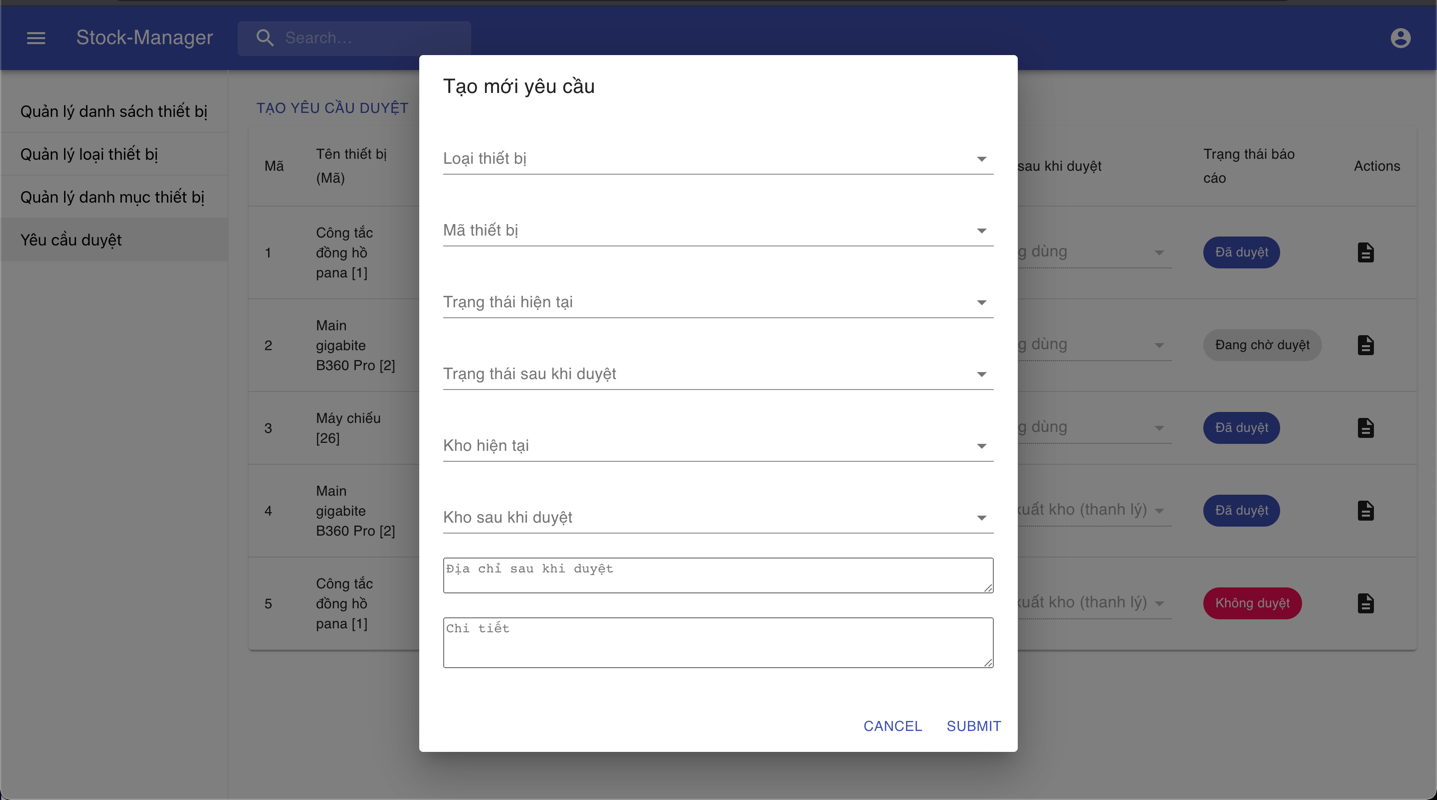
Cho phép người dùng quản lý danh sách danh mục các loại thiết bị.

### 3.1.5. Giao diện trang yêu cầu duyệt

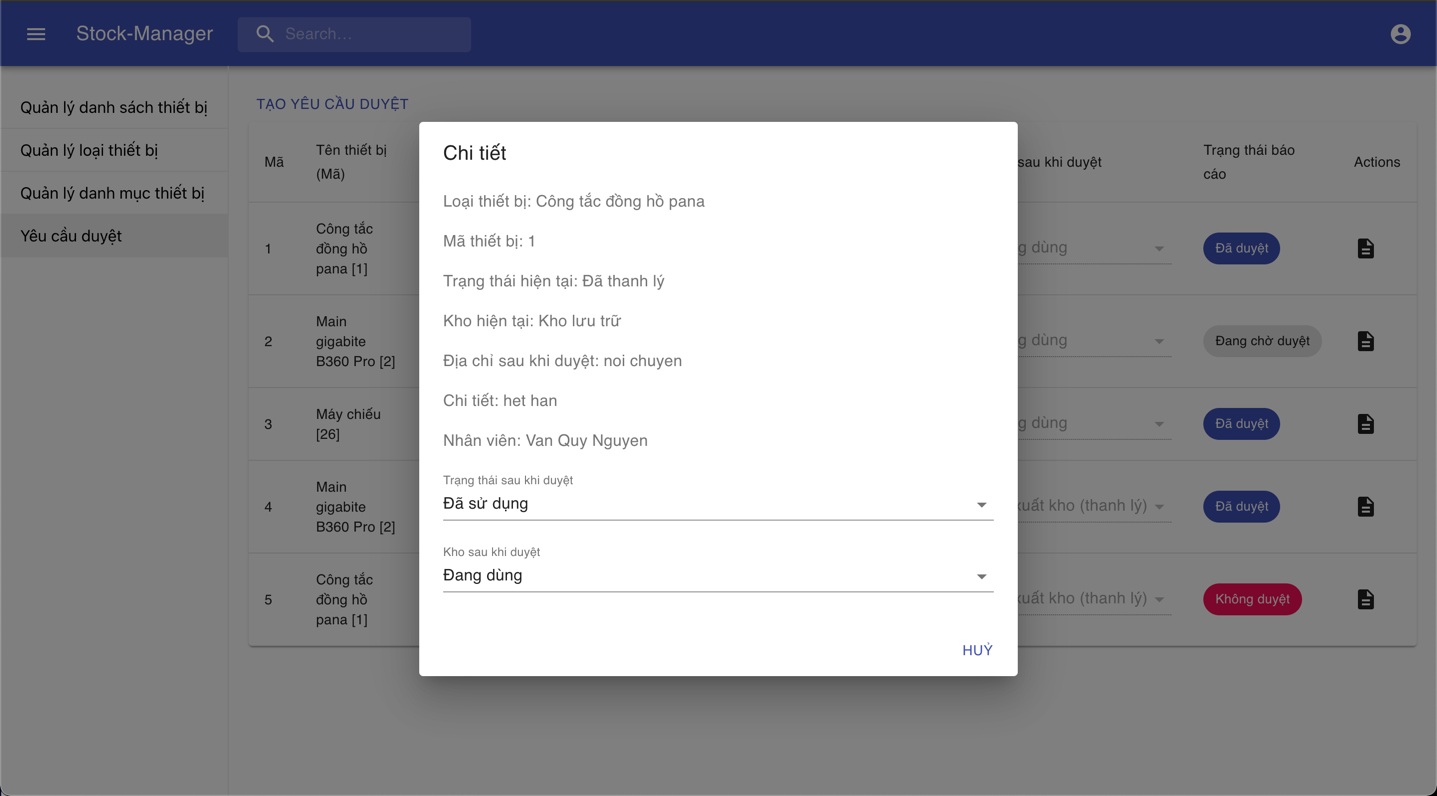


Hình ảnh 3. 13. Giao diện trang yêu cầu duyệt phía nhân viên

Nhân viên có thể xem danh sách yêu cầu duyệt và chi tiết từng yêu cầu.



Hình ảnh 3. 14. Giao diện tạo mới yêu cầu duyệt

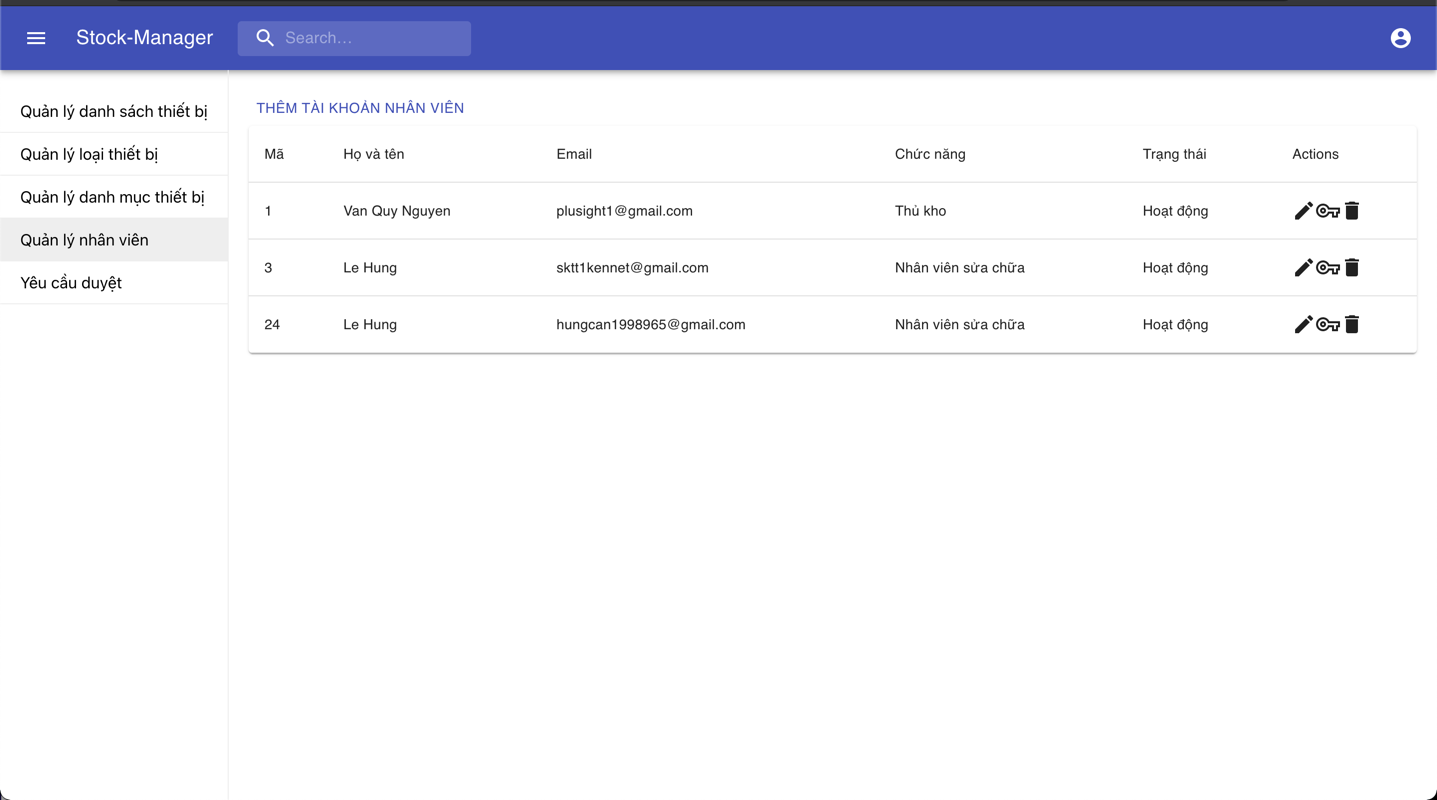


Hình ảnh 3. 15. Chi tiết đơn yêu cầu duyệt

## 3.2. Các thành phần chức năng hệ thống phía admin

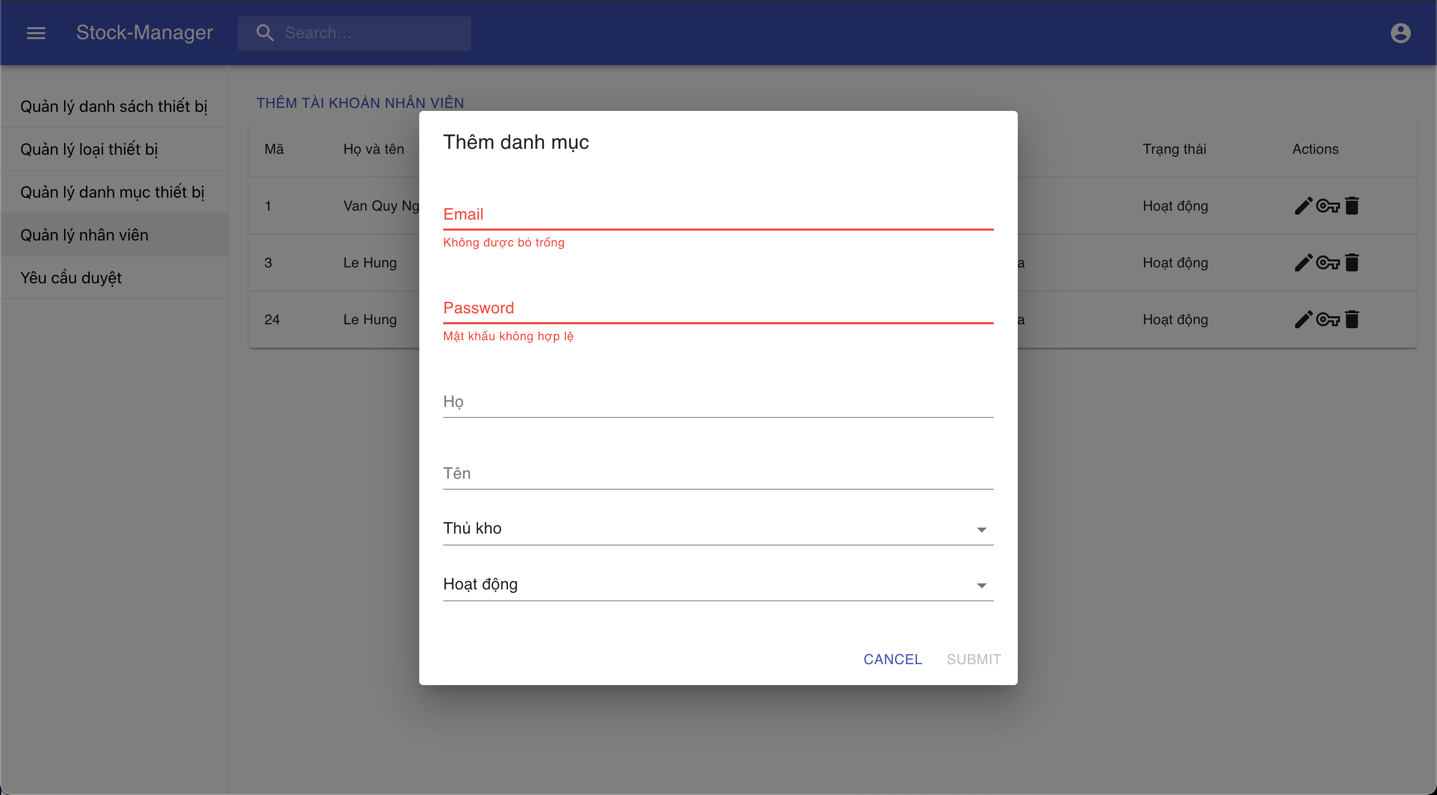
Giao diện admin cũng sẽ có các chức năng của nhân viên và thêm các tính năng sau

### 3.2.1. Giao diện chức năng quản lý danh sách nhân viên

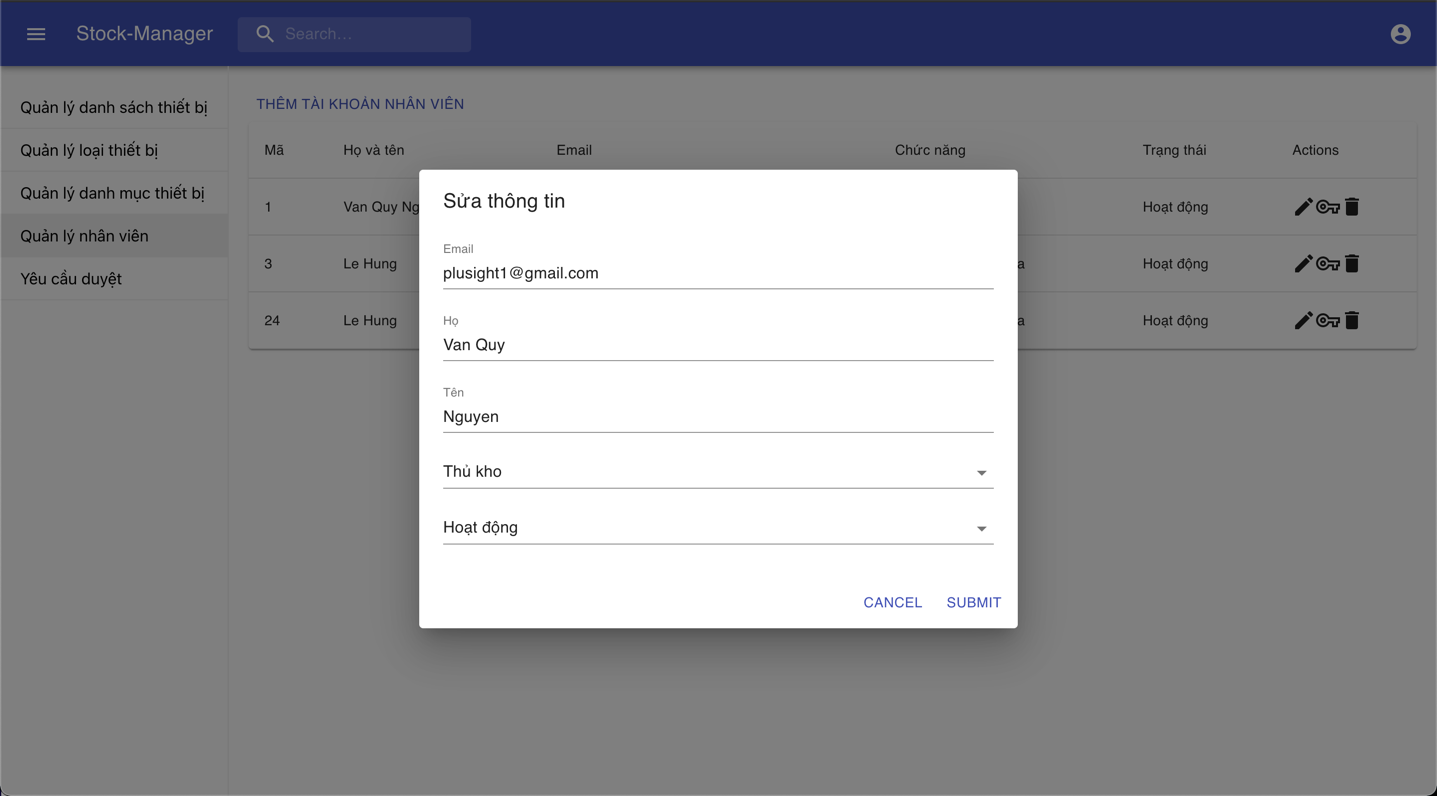


Hình ảnh 3. 16. Giao diện quản lý nhân viên

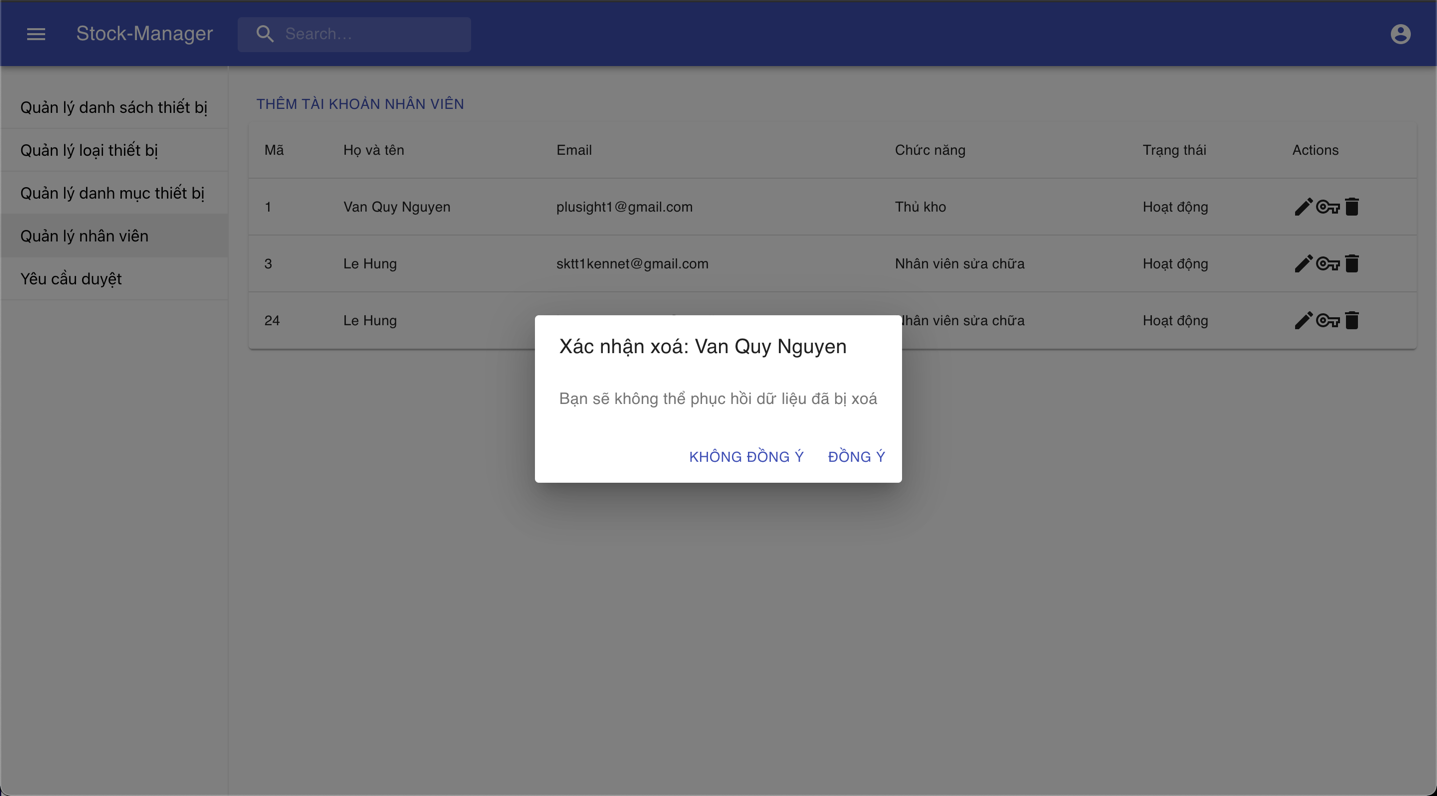
Chỉ có admin mới có quyền truy cập danh sách quản lý nhân viên.



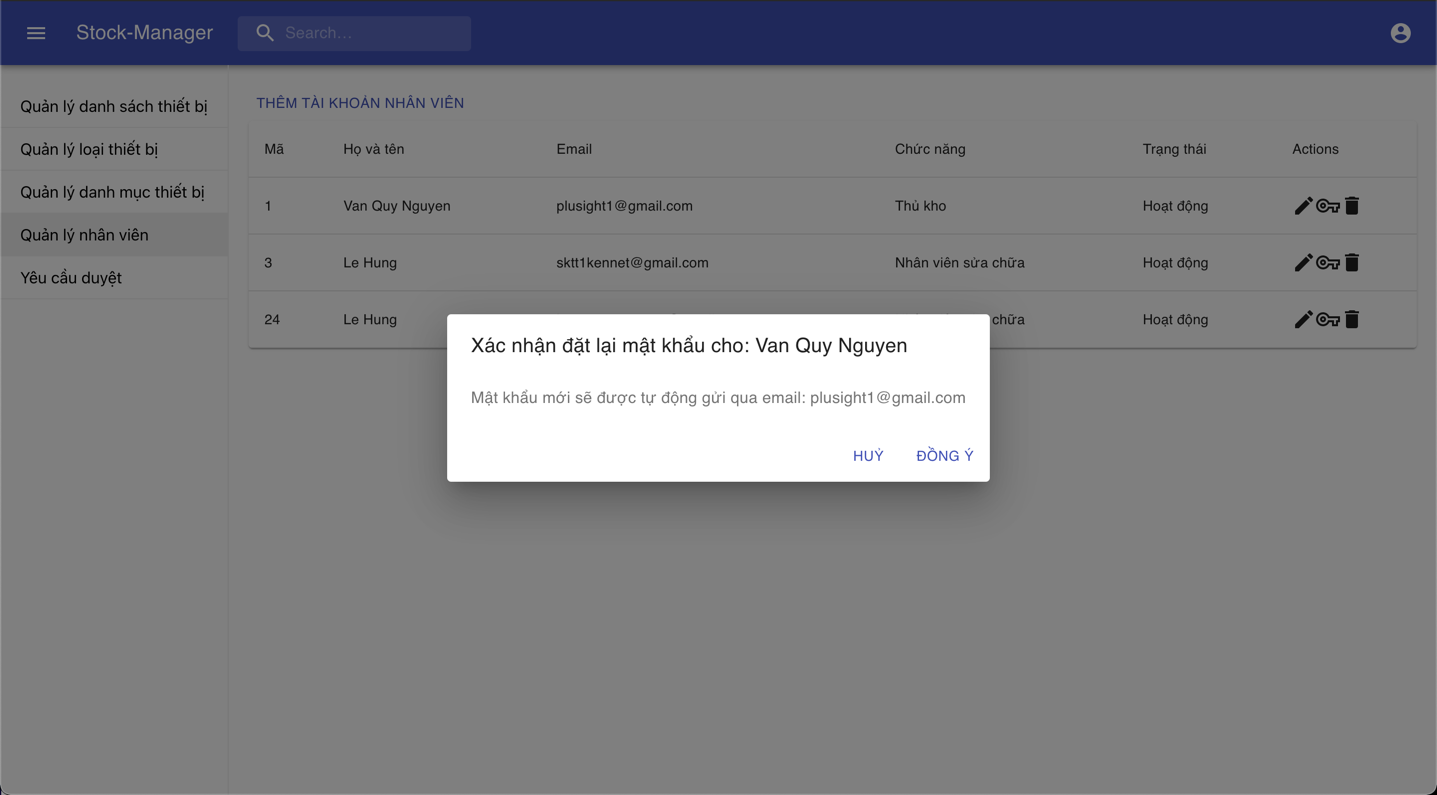
Hình ảnh 3. 17. Giao diện thêm nhân viên



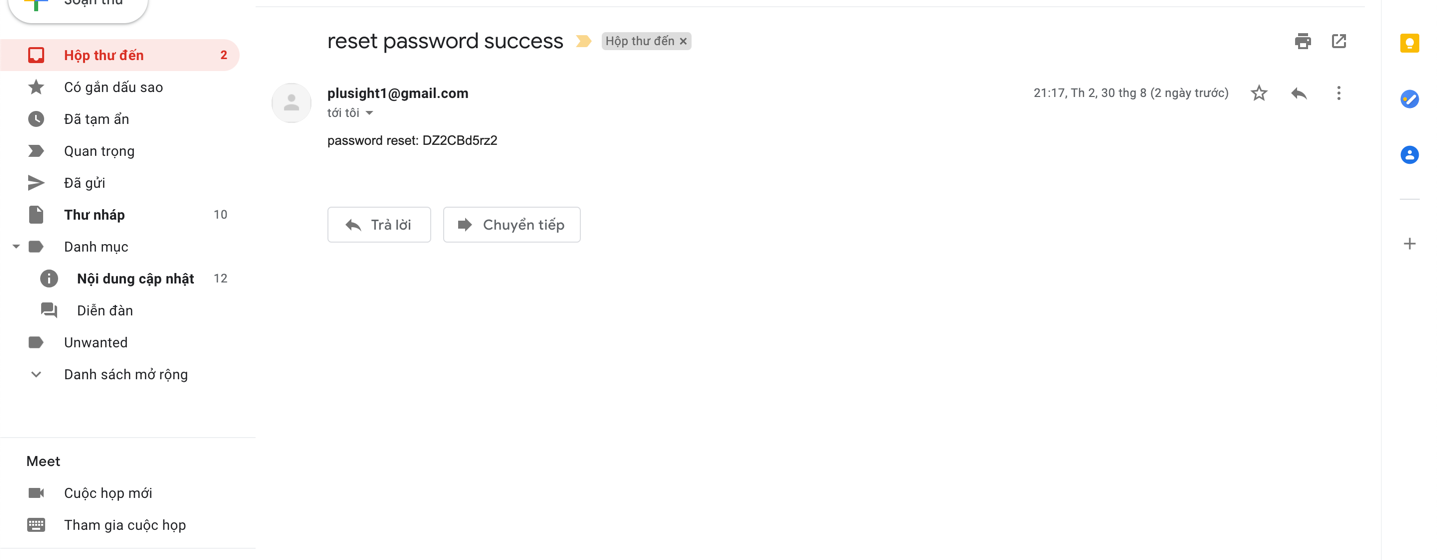
Hình ảnh 3. 18. Giao diện sửa thông tin nhân viên



Hình ảnh 3. 19. Giao diện cảnh báo xoá tài khoản nhân viên

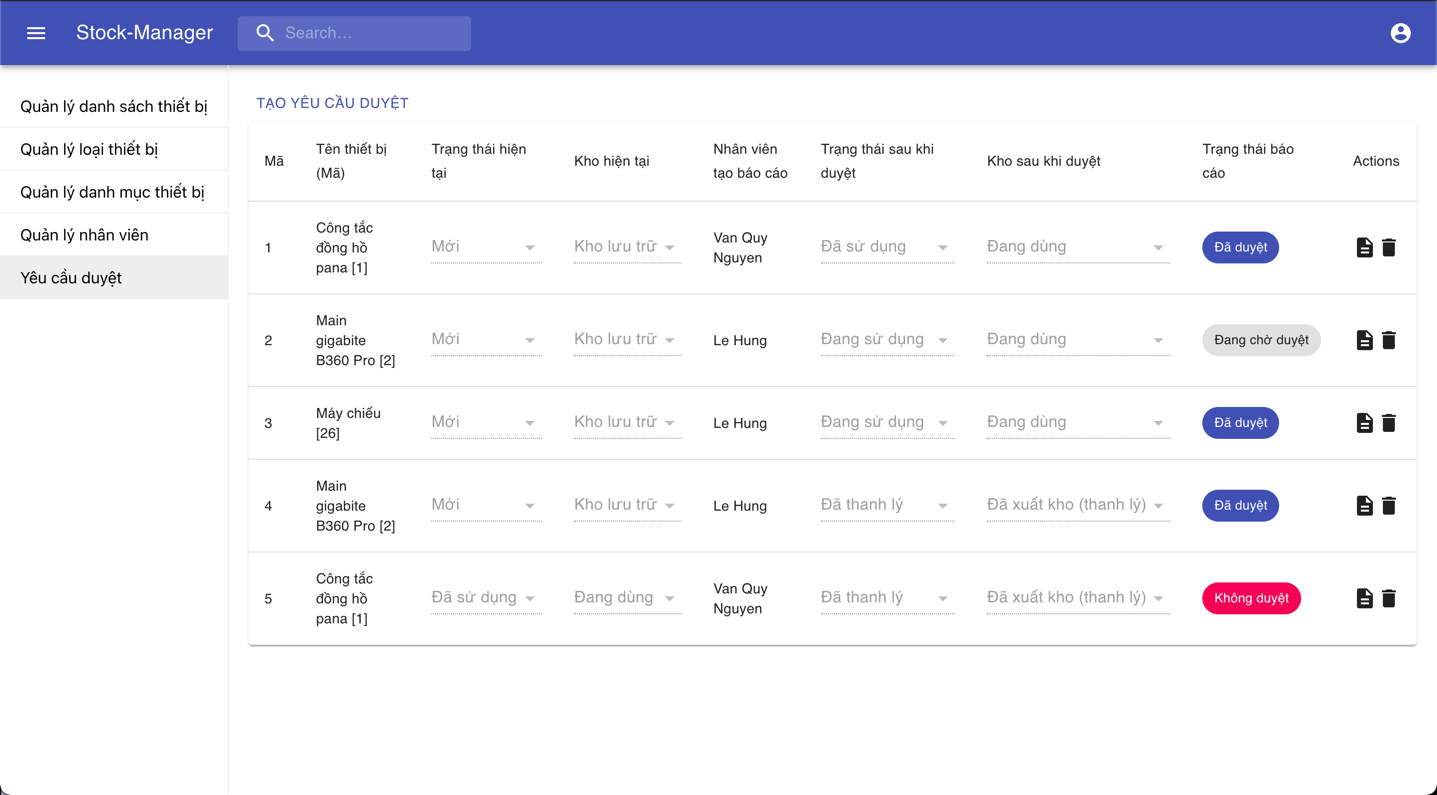


Hình ảnh 3. 20. Giao diện xác nhận đặt lại mật khẩu cho nhân viên

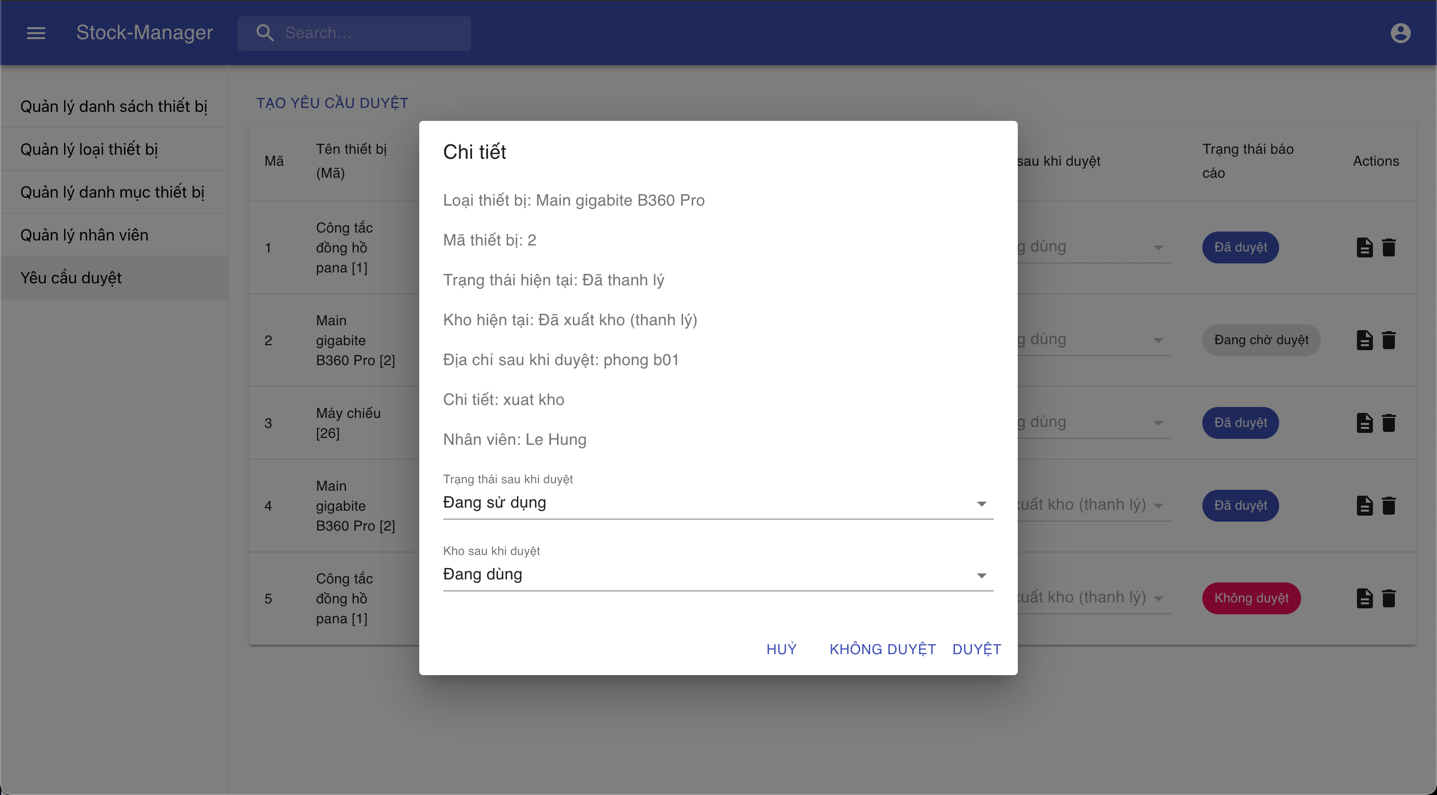


Hình ảnh 3. 21. Nội dung email nhận mật khẩu

### 3.2.2. Giao diện chức năng quản lý danh sách yêu cầu duyệt



Hình ảnh 3. 22. Giao diện chức năng quản lý danh sách yêu cầu duyệt phía admin



Hình ảnh 3. 23. Chi tiết yêu cầu duyệt với nút xử lý

# CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ

## 4.1. Kết quả đạt được

Xây dựng website quản lý kho không chỉ là việc xây dựng đơn thuần mà đòi hỏi trước đó phải tiến hành một cách có hệ thống giai đoạn khảo sát, phân tích thiết kế.

Qua đề tài trên, tôi đã cũng cố được kỹ năng lập trình webapp với ReactJS, SQL, ExpressJS ngoài ra tôi cũng học được các thành phần cơ bản của nhiều công nghệ nỗi bật hiện nay như Redux, Axios, cách hoạt động và sử dụng API. Qua đó cũng cố và mở rộng được kiến thức lập trình, tạo nền tảng vững chắc cho tương lai sau này.

## 4.2. Đánh giá ưu nhược điểm

### 4.2.1. Ưu điểm

* Rút ngắn thời gian xử lý kho
* Lọc các thông tin nhanh chóng
* Dữ liệu được cập nhật tức thời, không tải lại trang
* Giao diện chức năng đơn giản, dễ sử dụng
* UX mượt mà, nâng cao trải nghiệm của khách hàng

### 4.2.2. Nhược điểm

* Một số chức năng chưa hoàn thiện
* Mã nguồn chưa clean, gọn gàng
* UI chưa đẹp, màu sắc, bố cục trang web chưa hợp lý

## 4.3. Hướng phát triển và mở rộng đề tài

* Tối giản thao tác admin
* Nâng cao trải nghiệm người dùng thông qua việc nâng cấp UX, UI
* Sử dụng các thuật toán thông minh giúp admin kiểm duyệt thiết bị nhanh chóng
* Nâng cấp server, nâng cao khả năng lưu trữ, xử lý

**Tài liệu tham khảo**

1. Báo cáo EBI 2017
2. https://www.webico.vn/tong-quan-thuc-trang-tinh-hinh-thuong-mai-dien-tu-o-viet-nam-hien-nay-nam/
3. http://startup.vitv.vn/tin-chu/23-05-2017/thuong-mai-dien-tu-2017-xu-huong-thuc-trang-va-giai-phap/8393
4. https://mona-media.com/tin-tuc/thuc-trang-tinh-hinh-phat-trien-thuong-mai-dien-tu-viet-nam-2017/
5. http://enternews.vn/thuong-mai-dien-tu-viet-nam-tiem-nang-va-thach-thuc-107099.html
6. http://tapchitaichinh.vn/nghien-cuu--trao-doi/trao-doi-binh-luan/phat-trien-thuong-mai-dien-tu-tai-viet-nam-142802.html
7. https://www.hostinger.vn/huong-dan/javascript-la-gi/
8. https://gtvseo.com/seo-la-gi/
9. https://topdev.vn/blog/redux-la-gi/
10. https://topdev.vn/blog/gioi-thieu-ve-axios-mot-http-client-dua-tren-promise- cua-javascript/
11. https://o7planning.org/vi/12139/
12. https://techblog.vn/co-ban-ve-router-trong-reactjs
13. <https://www.dammio.com/2018/06/17/bootstrap-4-phan-1-gioi-thieu-ve-> bootstrap-4
14. <https://www.youtube.com/watch?v=gL5HBA_1bDQ&list=PLJ5qtRQovuE> OoKffoCBzTfvzMTTORnoyp
15. Group:

* Kteam - Lập Trình C/C++, C#, SQL, Unity3D, Java, Android, PHP, ASP.NET

https://www.howkteam.vn/

https://www.facebook.com/groups/howkteam

* Cộng đồng Front-end(HTML/CSS/JS) Việt Nam

https://www.facebook.com/groups/1065116420221723