

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



BÁO CÁO THỰC TẬP

Ngành: Khoa học Máy tính

Khoa: Công nghệ Thông tin

Dự án 1: “*Xây dựng module quản lý tài khoản người dùng.*”

Dự án 2: “*Giao diện UI cho Product Listing.*”

Người hướng dẫn: Nguyễn Phúc Lợi

Giảng viên đánh giá: PGS.TS Hà Quang Thụy

Sinh viên: Nguyễn Đức Anh

Mã sinh viên: 16020196

Lớp: K61-CA-CLC1

Hà Nội, tháng 9 năm 2019

MỤC LỤC

Lời cảm ơn	3
I. Giới thiệu chung	4
1. Giới thiệu công ty	4
2. Giới thiệu công việc.....	5
II. Kiến thức, lý thuyết	5
1. Web API	5
2. Backend	6
3. Frontend	7
III. Dự án 1	8
1. Giới thiệu	8
2. Yêu cầu chức năng.....	9
3. Cụ thể dự án.....	11
IV. Dự án 2.....	19
1. Giới thiệu	19
2. Yêu cầu chức năng.....	19
3. Cụ thể dự án.....	22
V. Tổng kết	25
Ý kiến đánh giá của công ty.....	26
Ý kiến đánh giá của giảng viên.....	27

LỜI CẢM ƠN

Kính gửi các thầy cô trong nhà trường và toàn thể thành viên tại Công ty Cổ phần Công nghệ Teko Việt nam.

Trong đợt thực tập vừa qua, em đã được tham gia vào môi trường làm việc thực tế. Nhờ đó, em đã được mở rộng tầm nhìn và trau dồi thêm những kiến thức hữu ích từ kinh nghiệm làm việc của các anh chị đi trước. Đó là nguồn thông tin vô cùng quý giá và có ích đối với em trong suốt quá trình học tập cho tới sau này.

Em xin chân thành cảm ơn các thầy cô trong khoa Công nghệ Thông tin đã tổ chức đợt thực tập bổ ích này.

Em xin gửi lời cảm ơn tới thầy Hà Quang Thụy, là người trực tiếp, nhiệt tình hướng dẫn, có những góp ý bổ sung, tài liệu rất hữu ích không những giúp em hoàn thành thực tập theo đúng quy định của nhà trường mà còn cung cấp cho em nhiều kiến thức cơ bản cần thiết.

Em xin cũng xin gửi lời cảm ơn tới ban lãnh đạo công ty cùng các anh chị trong công ty đã tạo điều kiện cho em có một môi trường thực tập vô cùng tốt, giúp em phát triển bản thân.

Em xin cảm ơn anh Nguyễn Phúc Lợi, anh Nguyễn Văn Linh là người trực tiếp hướng dẫn và hỗ trợ rất nhiệt tình em trong đợt thực tập tại công ty dù công việc bận rộn và áp lực.

Xin gửi lời cảm ơn tới 5 thành viên còn lại trong nhóm đã luôn hợp tác, giúp đỡ, khắc phục dần những khuyết điểm để cùng tiến bộ.

I. Giới thiệu chung

1. Giới thiệu công ty

- Công ty Cổ phần Công nghệ Teko Việt Nam được thành lập đầu năm 2017. Trên thực tế, đội ngũ của Teko đã đồng hành cùng nhau hơn 7 năm trong lĩnh vực công nghệ phục vụ cho giải trí – cụ thể là Thẻ thao điện tử. Teko Việt Nam đã xây dựng theo mô hình trực tuyến kết hợp với cửa hàng. Tekshop hiện được định vị là kênh bán hàng công nghệ chính hãng giá tốt nhất thị trường và cung cấp cả gói cho vay trả góp với đối tác là Công ty Tài chính ACS Việt Nam. Song song đó, Teko Việt Nam còn cung cấp dịch vụ lắp đặt phòng net bao gồm từ tư vấn, lắp đặt, nâng cấp dưới tên gọi là Gcafe. Đơn vị này hiện có khoảng 26.000 phòng máy là khách hàng trên toàn quốc. Và để củng cố thêm sức mạnh cho hệ sinh thái, Teko Việt Nam cần một đơn vị bán lẻ có thương hiệu lâu năm và đó là Phong Vũ, đơn vị có gần 20 năm trong lĩnh vực bán lẻ phần cứng, linh kiện công nghệ ở Thành phố Hồ Chí Minh.
- Tầm nhìn: Chia sẻ và đồng hành cùng nhau trong mục tiêu mang tới những sản phẩm tiện lợi, hữu ích cho xã hội dựa trên thể mạnh cốt lõi mà Teko lựa chọn: Công nghệ Thông tin.
- Giá trị cốt lõi:



Hình 1: Giá trị cốt lõi của Teko.

- Văn hóa công ty: Tầm nhìn rõ ràng, thành viên hòa đồng, môi trường làm việc mở.
- Lĩnh vực hoạt động: Công nghệ Thông tin.
- Địa chỉ: Tầng 16, TNR Tower, 54A Nguyễn Chí Thanh, quận Đống Đa, Hà Nội.

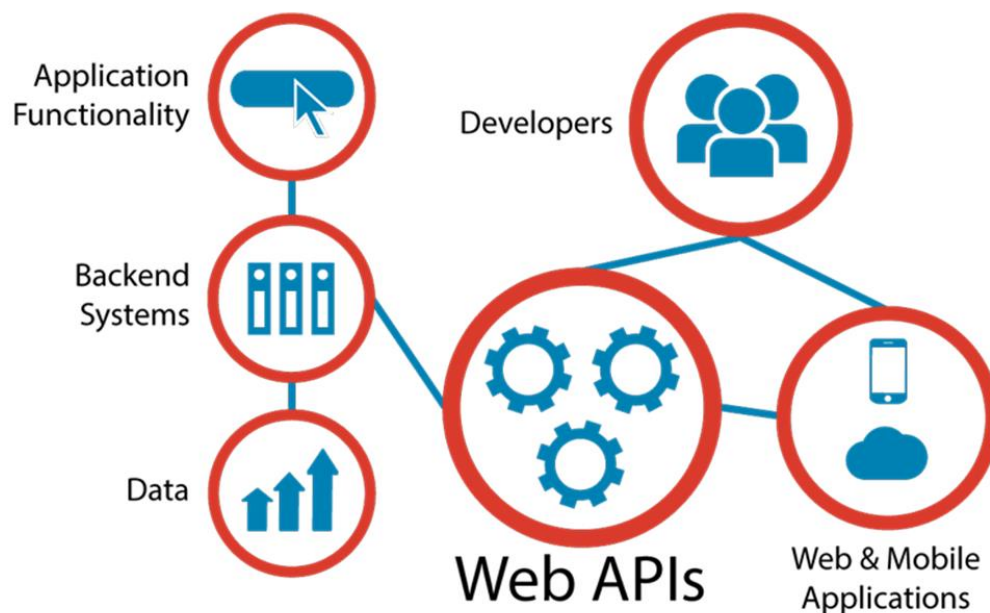
2. Giới thiệu công việc

- Tìm hiểu kiến thức lý thuyết, nắm rõ cách hoạt động của các framework, library, ngôn ngữ lập trình cần sử dụng trong công việc.
- Phát triển phần mềm theo các yêu cầu đã đưa ra.
- Học các kỹ năng xây dựng dự án theo nhóm.

II. Kiến thức, lý thuyết

1. API, Web API

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.



Hình 2: Cấu trúc của Web API.

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Phương thức hoạt động:

- Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.
- Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.
- Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
- Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Database, hiển thị dữ liệu...

2. Backend

Phần backend của một trang web bao gồm một máy chủ, một ứng dụng, và một cơ sở dữ liệu. Các lập trình viên backend xây dựng và duy trì công nghệ mà sức mạnh của những thành phần đó, cho phép phần giao diện người dùng của trang web có thể tồn tại được.

Backend tập trung vào các nội dung:

- Các ngôn ngữ lập trình backend như php, Python, Java, Ruby,...
- Auto-test cho ngôn ngữ được sử dụng.
- Truy cập, quản lí database.

- Logic.
- Khả năng mở rộng.
- Tính sẵn sàng cao.
- Bảo mật, xác thực và ủy quyền.
- Kiến trúc phần mềm.
- Chuyển đổi dữ liệu.
- Phương pháp và phần mềm sao lưu.

* Hệ thống yêu cầu sử dụng framework Flask của Python.

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các API nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

3. Frontend

Phần frontend của một trang web là phần tương tác với người dùng. Tất cả mọi thứ được hiển thị khi điều hướng trên Internet, từ các font chữ, màu sắc cho tới các menu xổ xuống và các thanh trượt, là một sự kết hợp của HTML, CSS, và JavaScript được điều khiển bởi trình duyệt máy tính. Frontend lấy dữ liệu từ backend, xử lý (nếu cần), sau đó hiển thị lên cho người dùng; người dùng có thể thao tác với frontend để gọi request tới backend của hệ thống. Frontend cũng có thể thực hiện các phép logic giúp giảm tải cho hệ thống backend, đồng thời giúp trải nghiệm người dùng nhanh và tiện lợi hơn như việc xác định tính đúng đắn của các form nhập dữ liệu...

Frontend tập trung vào các nội dung:

- Markup và các ngôn ngữ HTML, CSS, JavaScript và các thư viện phụ trợ như SASS hay JQuery.
- Xử lý request bất đồng bộ, AJAX.
- Các ứng dụng một trang (với React, AngularJS, Vue.js).
- Hiệu suất trang web.
- Responsive web design.
- Search engine
- Giao diện trực quan.

* Hệ thống đã làm yêu cầu sử dụng library React của JavaScript.

React (còn được gọi là React.js hoặc ReactJS) là một thư viện JavaScript để xây dựng giao diện người dùng. Nó được duy trì bởi Facebook và một cộng đồng gồm các nhà phát triển và công ty cá nhân.

React có thể được sử dụng làm cơ sở trong việc phát triển các ứng dụng một trang (single page app) hoặc di động, vì nó giúp tối ưu để tìm nạp dữ liệu thay đổi nhanh chóng cần được ghi lại. Các ứng dụng React phức tạp thường yêu cầu sử dụng các thư viện bổ sung để quản lý trạng thái, định tuyến và tương tác với API.

III. Dự án 1

1. Giới thiệu

a. Mô tả

Xây dựng hệ thống quản lý tài khoản người dùng cơ bản, cho phép người dùng tham gia và sử dụng hệ thống.

b. Yêu cầu hệ thống

- Phát triển backend dựa trên Flask – Python.
- Phát triển frontend dựa trên React – JavaScript.
- Xây dựng hệ thống web API.

2. Yêu cầu chức năng

a. Chức năng cho người dùng cuối

- Đăng ký tài khoản, trong đó:
 - Thông tin tài khoản gồm:
 - + Tên đăng nhập (*): Tối thiểu 6 kí tự gồm chữ, số viết liền không dấu.
 - + Mật khẩu (*): Tối thiểu 8 kí tự gồm chữ hoa, chữ thường, số.
 - + Email (*): Đúng định dạng email.
 - + Nút Đăng ký.
 - Khi Đăng ký:
 - + Nếu có thông tin không hợp lệ, thông báo rõ ràng cho người dùng.
 - + Nếu mọi thông tin hợp lệ:
 - Thông báo người dùng đã đăng ký thành công, người dùng có thể đăng nhập.
 - Gửi email xác nhận tài khoản đã tạo thành công.
- Đăng nhập:
 - Form đăng nhập gồm:
 - + Tên đăng nhập (*)
 - + Mật khẩu (*)
 - + Nút đăng nhập | Đăng ký | Quên mật khẩu
 - Khi đăng nhập:
 - + Nếu có thông tin không hợp lệ, thông báo rõ ràng cho người dùng.
 - + Nếu mọi thông tin hợp lệ:
 - Chuyển người dùng sang trang Chào mừng đăng nhập thành công, có nút Đổi mật khẩu | Đăng xuất.
 - Lưu trạng thái đăng nhập của người dùng trong vòng 30 phút:

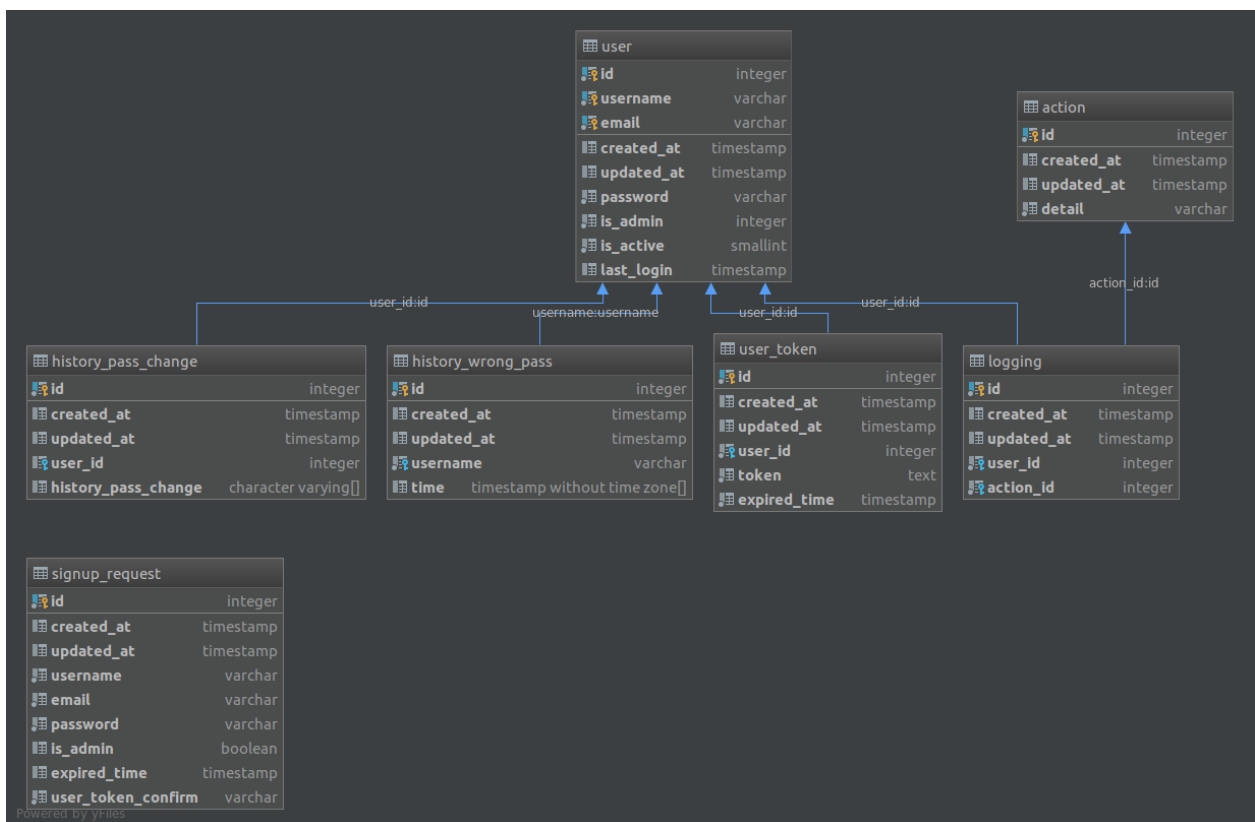
- Trong 30 phút người này tắt trình duyệt mở lại hệ thống vẫn duy trì đăng nhập và vào thẳng trang Chào mừng, không yêu cầu Đăng nhập lại.
 - Quá 30 phút hệ thống tự động Đăng xuất.
- Quên mật khẩu:
 - Form Quên mật khẩu gồm:
 - + Tên người dùng (*)
 - + Email (*)
 - + Nút Reset mật khẩu
 - Khi nhấn Reset, hệ thống kiểm tra đúng tên người dùng có email tương ứng và gửi mật khẩu mới (sinh ra ngẫu nhiên theo quy tắt đăng ký) cho người dùng vào email. Người dùng có thể đăng nhập lại bằng mật khẩu mới này.
 - Đổi mật khẩu:
 - Form đổi mật khẩu gồm:
 - + Mật khẩu hiện tại(*)
 - + Mật khẩu mới (*)
 - Khi nhấn Đổi mật khẩu, hệ thống kiểm tra đúng mật khẩu hiện tại, mật khẩu mới không trùng mật khẩu hiện tại sẽ đổi mật khẩu mới cho người dùng.
 - Đăng xuất
 - Khi nhấn Đăng xuất, người dùng sẽ thoát khỏi hệ thống và phải đăng nhập lại nếu muốn truy cập lại.
 - Lưu ý:
 - Mỗi tên đăng nhập chỉ được đăng ký hoặc tạo ra 1 lần duy nhất trên hệ thống.
 - Các email không được trùng nhau trên hệ thống.
 - Tất cả thao tác thành công, thất bại đều có thông báo rõ ràng cho người dùng.

b. Yêu cầu khác

- Hệ thống có giao diện đơn giản, dễ phát triển các chức năng về sau.
- Hệ thống cần đảm bảo hiệu năng tốt.
- Nhân lực: cá nhân.

3. Cụ thể dự án

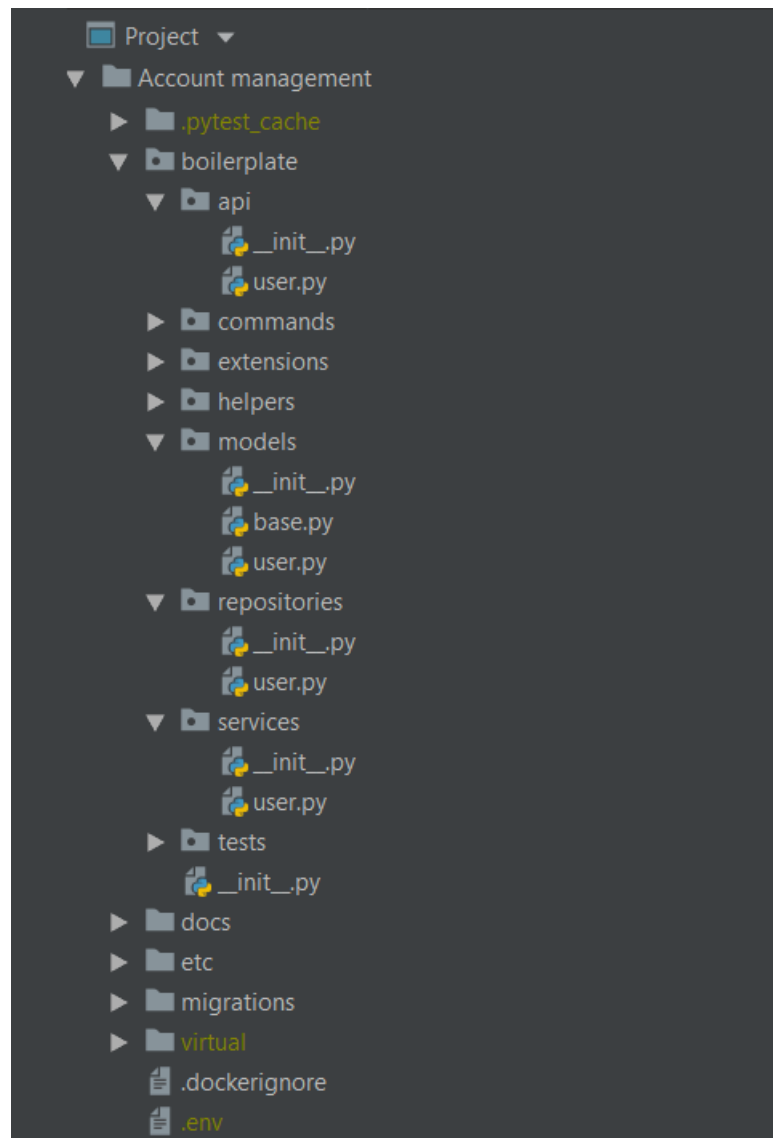
a. Phân tích thiết kế database



Hình 3: Thiết kế database của hệ thống quản lý tài khoản.

b. Backend

➤ Cấu trúc



Hình 4: Cấu trúc của hệ thống backend.

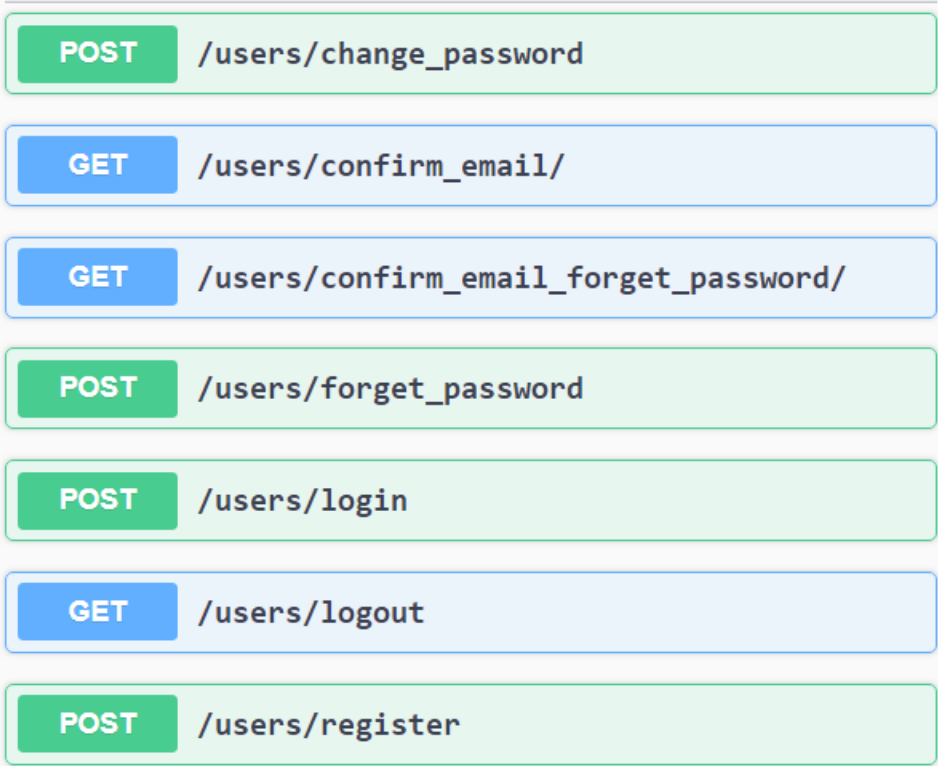
➤ Một vài thành phần của hệ thống

- Thư mục “api”: Nơi khai báo và quản lí các route API, các route này thực hiện lệnh gọi về các component trong “services”.
- Thư mục “services”: Nơi chứa các phép logic, xử lí dữ liệu.
- Thư mục “repositories”: Nơi chứa các đoạn mã xử lí ra vào database.

- Thư mục “models”: Nơi thực hiện các lệnh trực tiếp tới database.
- Thư mục “migration”: Nơi chứa các phiên bản database đã tạo và sử dụng.
- Thư mục “virtual”: Nơi tạo môi trường phát triển.

➤ **Kết quả**

- Đây là danh sách các API backend được tạo ra để thực hiện gọi các request.

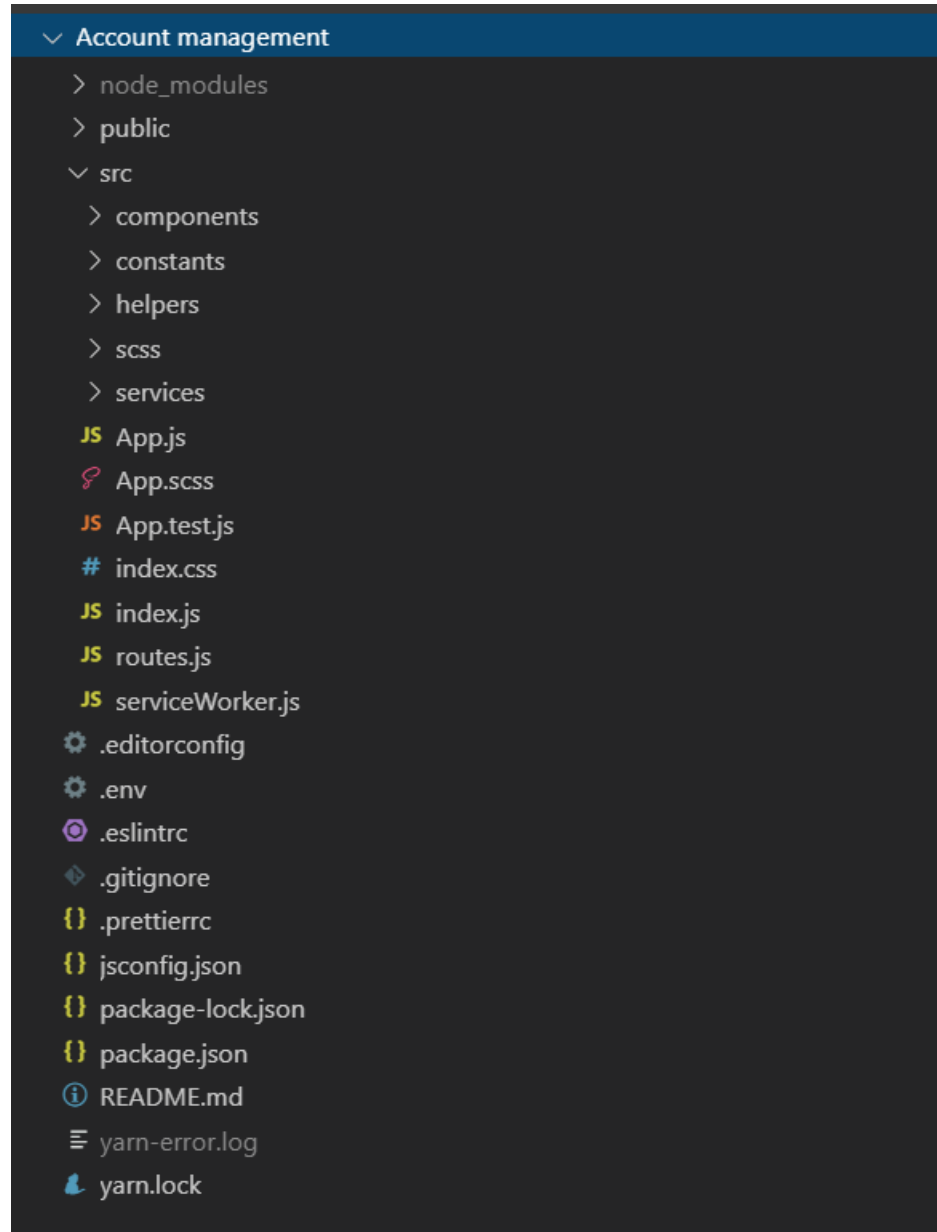


POST	/users/change_password
GET	/users/confirm_email/
GET	/users/confirm_email_forget_password/
POST	/users/forget_password
POST	/users/login
GET	/users/logout
POST	/users/register

Hình 5: Danh sách API tạo ra từ backend.

c. Frontend

➤ Cấu trúc



Hình 6: Cấu trúc hệ thống frontend.

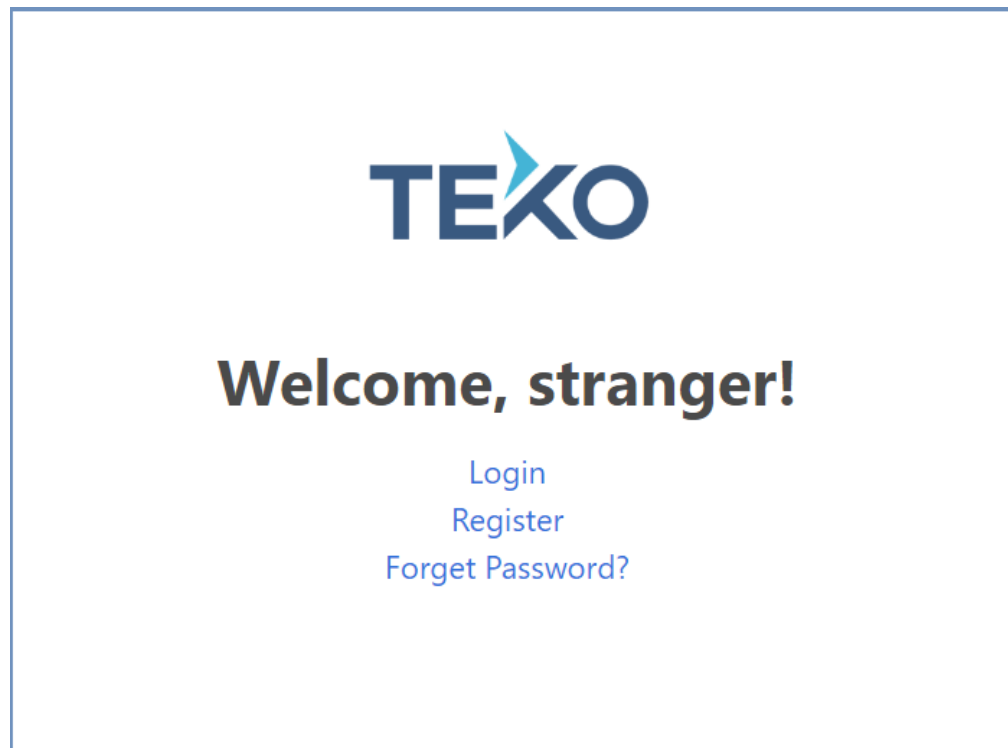
➤ Thành phần cơ bản

- Thư mục “node_modules”: Chứa các thư viện được lập trình sẵn để xây dựng hệ thống.

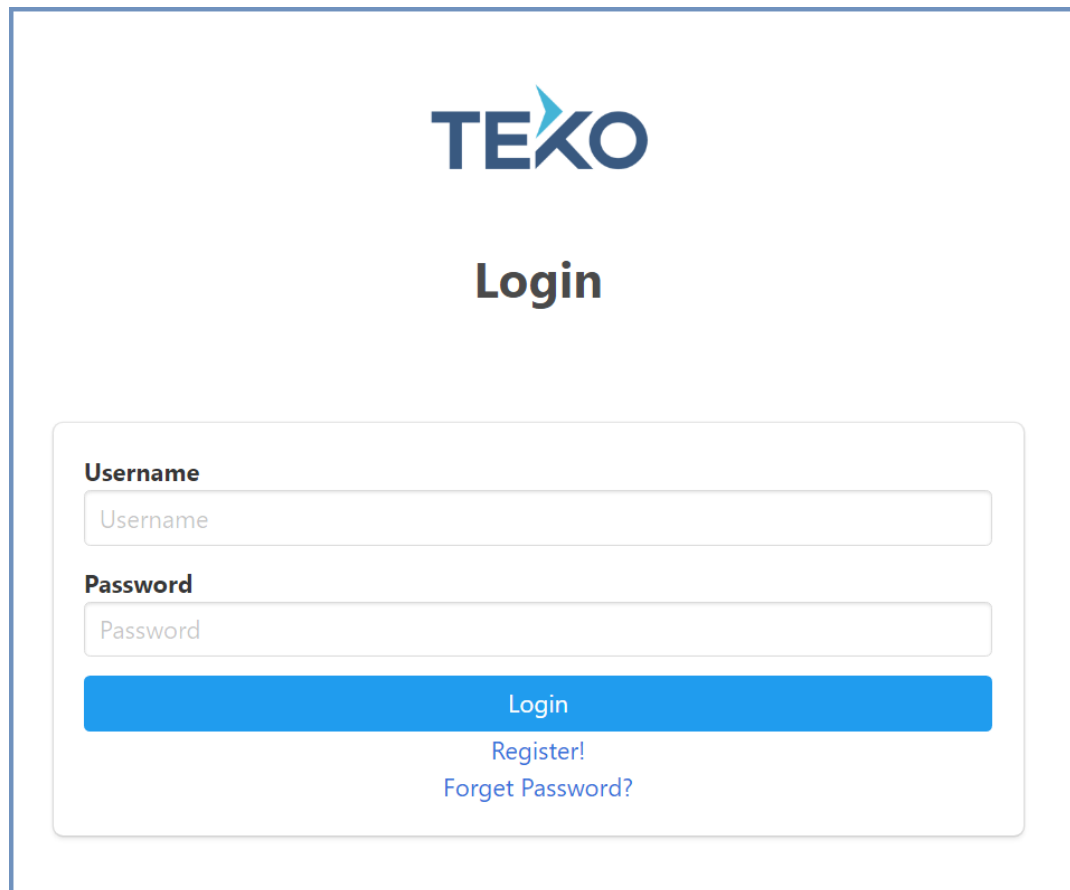
- Thư mục “components”: Chứa các thành phần tạo nên trang web, các component được gọi sẽ xử lý logic và hiển thị của trang web.
- Thư mục “services”: Thực hiện việc gọi và gửi request tới url API của backend.
- Tập tin “App.js”: Nơi thực hiện việc cấu hình các định tuyến và gọi component qua định tuyến.

➤ **Kết quả**

Dưới đây là một vài giao diện của hệ thống:

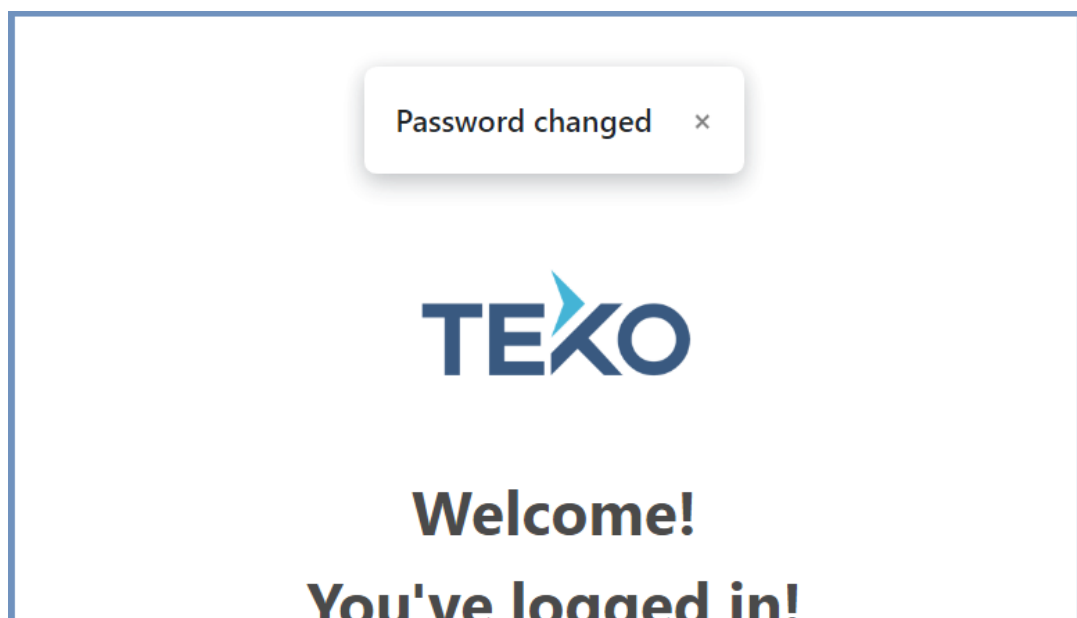


Hình 7: Giao diện trang khởi đầu.



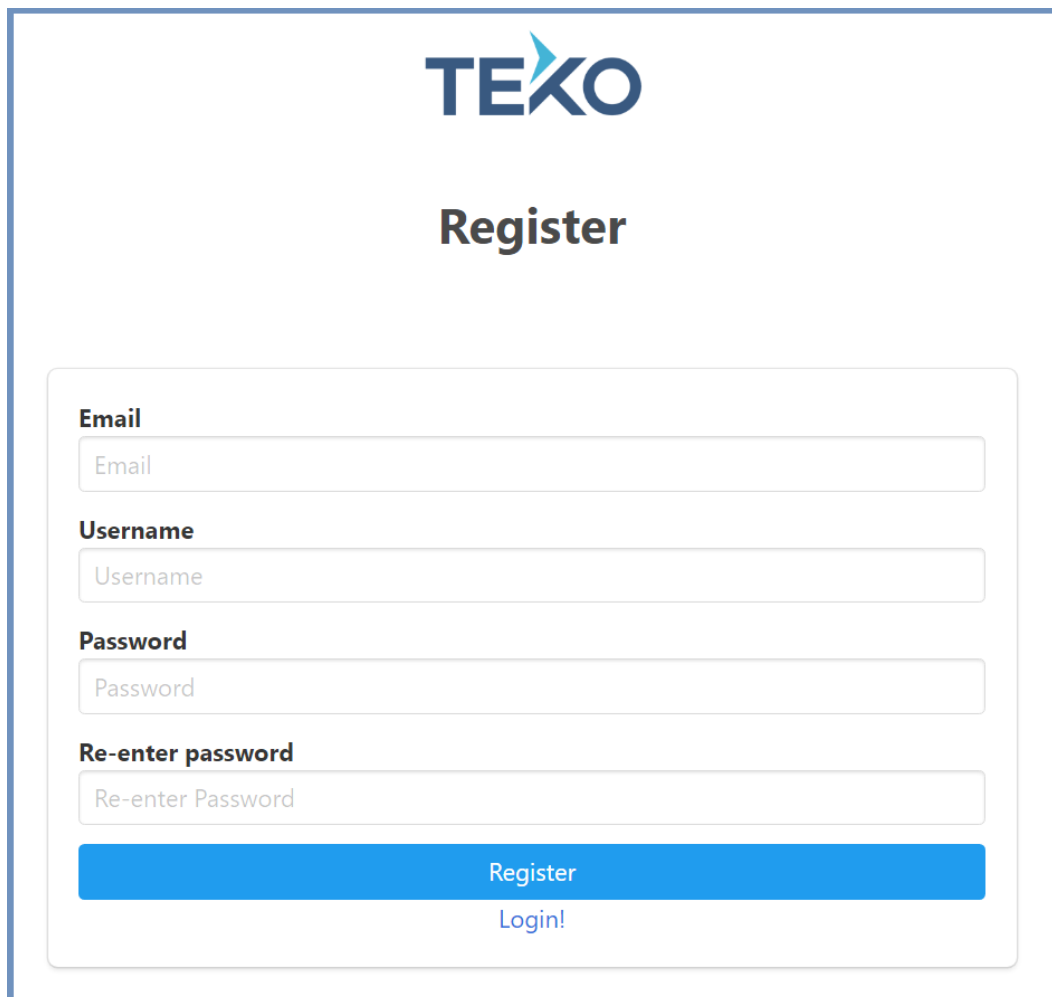
The screenshot shows the TEKO Login interface. At the top center is the TEKO logo, which consists of the word "TEKO" in a bold, dark blue sans-serif font, with a light blue stylized arrow pointing upwards and to the right integrated into the letter 'K'. Below the logo is the word "Login" in a bold, dark grey sans-serif font. Underneath "Login" is a light grey rounded rectangular box containing the login form. Inside this box, the label "Username" is in bold dark grey, followed by a text input field with the placeholder text "Username". Below that, the label "Password" is in bold dark grey, followed by a text input field with the placeholder text "Password". At the bottom of the box is a solid blue rectangular button with the word "Login" in white. Below the button are two links: "Register!" and "Forget Password?", both in a smaller, blue, sans-serif font.

Hình 8: Giao diện Đăng nhập



The screenshot shows the TEKO Welcome page. At the top center is a white notification box with a thin grey border and a soft drop shadow. It contains the text "Password changed" in a dark grey sans-serif font, followed by a small grey "x" icon for closing the notification. Below the notification is the TEKO logo, identical to the one in the previous image. Underneath the logo, the word "Welcome!" is displayed in a large, bold, dark grey sans-serif font. At the bottom, the phrase "You've loaded in!" is displayed in a large, bold, dark grey sans-serif font.

Hình 9: Thông báo trạng thái.



TEKO

Register

Email

Username

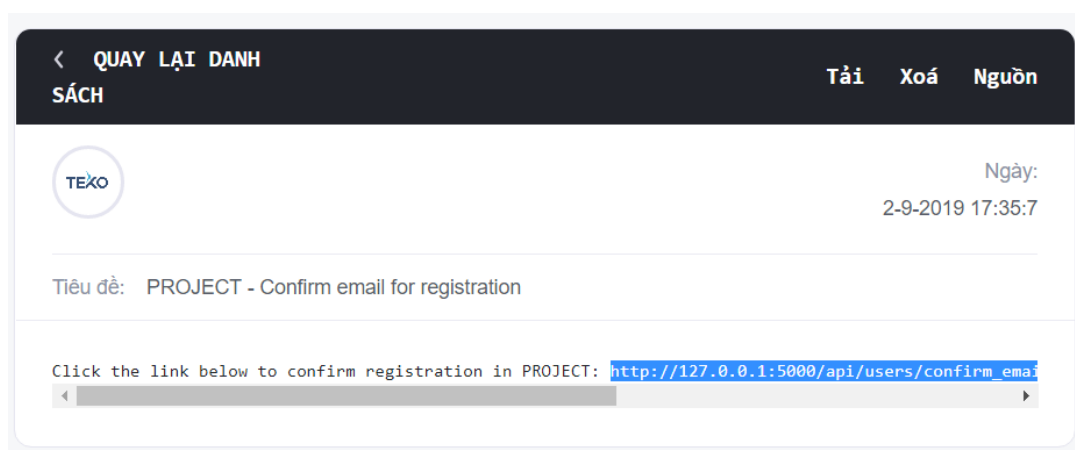
Password

Re-enter password


[Register](#)

[Login!](#)

Hình 10: Giao diện Đăng kí



Hình 11: Email người dùng nhận được khi đăng kí/quên mật khẩu.



Change Password

Current password

Password must be 6 or more characters

Enter new password

Password must be 6 or more characters

Re-enter password

Confirm password does not match

Change password

Hình 12: Giao diện đổi mật khẩu và nhập sai thông tin.

IV. Dự án 2

1. Giới thiệu

a. Mô tả

Hiện tại, dự án Product Listing là nguồn cung cấp API trả về các thông tin sản phẩm để cho các hệ thống bán hàng (website phongvu.vn, app đại lý, app showroom của Phong Vũ, vnshop,...) cũng như các backoffice (PPM, sale chat tool,...) sử dụng để nhằm mục đích có được thông tin sản phẩm chính xác, đầy đủ và nhanh nhất. API cũng cấp các tham số cần thiết, hỗ trợ sắp xếp, lọc sản phẩm theo nhiều field, nhiều giá trị khác nhau. Tuy nhiên, do đặc thù là API, trong quá trình làm việc, đôi khi cần có cái nhìn trực quan hơn về thông tin sản phẩm được trả ra trong API để có thể kiểm tra tính chính xác, độ hiệu quả của câu query tìm kiếm.

Do đó, cần thiết có một giao diện client có thể hỗ trợ việc hiển thị kết quả của API cũng như có thể support cho nhiều điều kiện lọc, sắp xếp khác nhau.

b. Yêu cầu hệ thống

- Backend: Đã có service riêng cung cấp API.
- Frontend: Sử dụng React thiết kế giao diện phù hợp với yêu cầu bài toán. Sử dụng các kỹ thuật lập trình, 3rd party nếu cần thiết để tối ưu. Giao diện cần đầy đủ chức năng, UI/UX hợp lý.

2. Yêu cầu chức năng

a. Chức năng cho người dùng

Business Rule	Mô tả
Tổng quát	Cho phép người dùng tìm kiếm và xem danh sách, thông tin sản phẩm theo mong muốn
1	Cho phép tìm kiếm kết hợp (AND) của các tiêu chí đã chọn (từ khóa tìm kiếm + filter).
2	Bộ Filter : <ul style="list-style-type: none"> • Bắt buộc phải có lựa chọn channel và terminal.

	<ul style="list-style-type: none"> • Thương hiệu (brands), tập thuộc tính (attributeSets), danh mục (categories) , Các giá trị này được lấy tại API extra-data của Catalog. Với brands và categories có thêm lựa chọn Bao gồm/ Không bao gồm (brands_ne, productLines_ne). • Khoảng giá (price_gte, price_lte)... Có thể hiện ô nhập hoặc slider cho thuận tiện. • Filter khác: Có KM (hasPromotions), Là bundle (isBundle), Có gán danh mục (hasCategories). • Chỉ hiển thị 10 giá trị trong các filter này và có nút Xem thêm để hiển thị toàn bộ giá trị còn lại, VD: (có thể làm khác, VD xổ ra 1 popup con bên cạnh...). <p>Cho phép chọn nhiều giá trị trong 1 filter, khi đó sẽ tìm kiếm sp có thuộc tính có giá trị IN các lựa chọn này</p>
3	Các giá trị đã chọn trong filter được hiển thị riêng ở khối, ngoài ra các giá trị này vẫn được giữ nguyên tick chọn trong danh sách Filter.
4	<p>Input text: cho phép nhập từ khóa tìm kiếm bất kỳ.</p> <p>Lưu lại trên browser 10 từ khóa gần nhất và hiển thị suggestion khi user nhập , ngoài ra hiển thị các Search đã lưu lại, cho phép Clear all hoặc Clear từng bản ghi.</p>
5	<p>Khối lựa chọn sắp xếp:</p> <ul style="list-style-type: none"> • Cho phép lựa chọn sắp xếp theo nhiều tiêu chí: giá (price), tên (name), tổng tồn kho (stock), Mới về (new), SKU (sku). • Với mỗi tiêu chí có phép chọn sort asc hoặc desc. • Cho phép chọn và sắp xếp đồng thời theo nhiều tiêu chí trên , tiêu chí nào chọn trước ưu tiên sắp xếp trước.

6	<p>Khi tìm kiếm, hiển thị danh sách sản phẩm thỏa mãn điều kiện tìm kiếm, trong đó:</p> <ul style="list-style-type: none"> • Khối hiển thị thông tin sản phẩm: Sẽ hiển thị 1 ảnh làm ảnh đại diện, tên sản phẩm, giá của sản phẩm, tooltip nếu có các khuyến mại, rating... • Cần có phân trang, mặc định 20 sp/ trang, có thể thay đổi số lượng này từ 10-100 sp/trang. • Có progress bar trong lúc chờ load. • Cho phép save lại bộ filter/sort tương ứng để sử dụng lại về sau (save trên browser).
7	<p>Logic hiển thị thông tin sản phẩm:</p> <ul style="list-style-type: none"> • Tên: Ưu tiên field displayName. Trong trường hợp displayName rỗng hoặc null sẽ lấy tên sản phẩm (field name) • Giá: Ưu tiên giá theo thứ tự sau: <ul style="list-style-type: none"> + Trong khối promotionPrices, nếu object có channel và terminal giống với channel và terminal truyền lên khi gọi api thì sẽ lấy giá finalPrice, hiển thị là giá khuyến mại, giá bị gạch hiển thị sellPrice. + Trong khối promotionPrices, nếu không thỏa mãn điều kiện a, nhưng có object mà channel = 'all' và terminal = null thì lấy giá finalPrice, hiển thị là giá khuyến mại, giá bị gạch hiển thị sellPrice. + Nếu không thỏa mãn cả a và b, chọn hiển thị giá bán sellPrice và giá bị gạch hiển thị giá niêm yết supplierSalePrice.

b. Yêu cầu khác

- Hệ thống hiển thị rõ ràng, chính xác.
- Hiệu năng tốt.
- Làm theo nhóm, phân chia công việc theo các thành phần.
 - Trần Quang Tuấn: Cấu trúc trang web, gọi API từ backend và ghép nối các thành phần riêng rẽ vào chung trang web (wrapper).
 - Hoàng Xuân Trường, Nguyễn Đức Anh: Phát triển thành phần filter, trả về giá trị cho wrapper.
 - Nguyễn Hữu Nghĩa, Hoàng Văn Thắng: Phát triển phần hiển thị danh sách thông tin sản phẩm.
 - Nguyễn Đình Nam Thịnh: Phát triển phần Search bar và Sorter.

3. Cụ thể dự án

a. Thiết kế cơ bản

The diagram illustrates the basic layout of the system interface. It consists of the following components:

- Logo TEK0**: Located in the top left corner.
- Tu khoa tìm kiếm**: A search bar located at the top right.
- khởi filter**: A sidebar on the left containing filter categories: Others, brands, price, attribute_set, categories, and Điều kiện lọc. Below these are input fields for **Gia trị lọc** and an **Áp dụng** button.
- Khởi lựa chọn sắp xếp**: A sorting section with five radio buttons and arrows indicating sort order.
- Khởi filter đã chọn**: A section for selected filters.
- Khởi hiển thị thông tin sản phẩm**: The main area for displaying product information.
- Pagination**: A pagination bar at the bottom right.

Hình 13: Thiết kế cơ bản giao diện hệ thống.

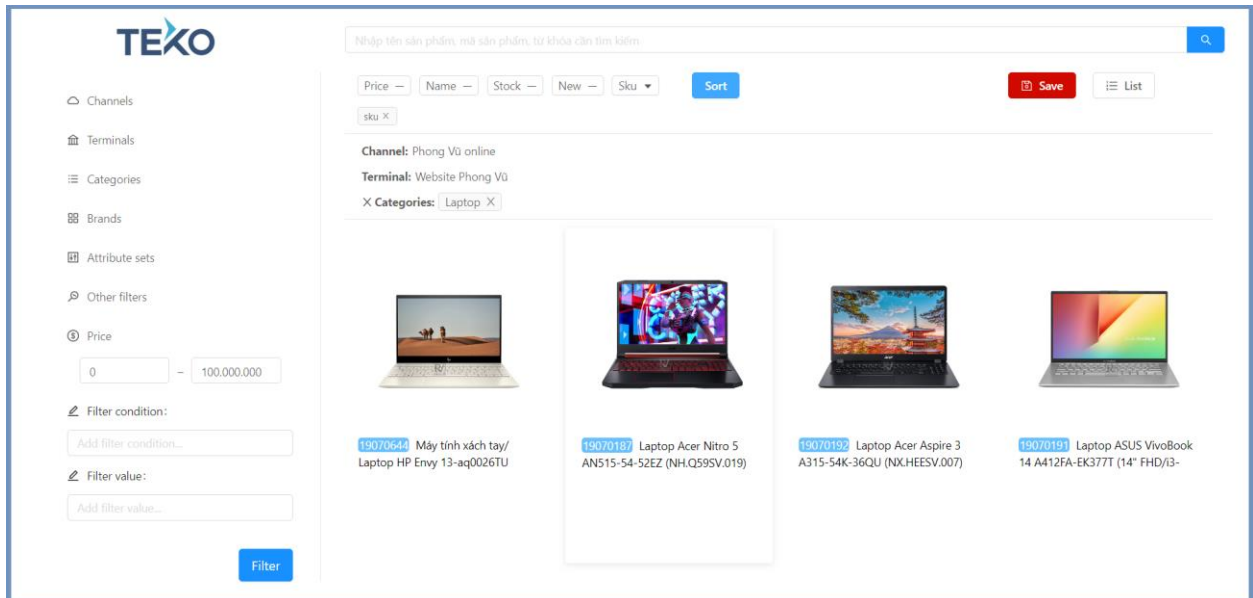
b. Cấu trúc

Cấu trúc của hệ thống giống y hệt như frontend của dự án 1, khác nhau ở chỗ hệ thống lớn hơn và có nhiều thành phần hơn trong các file code.

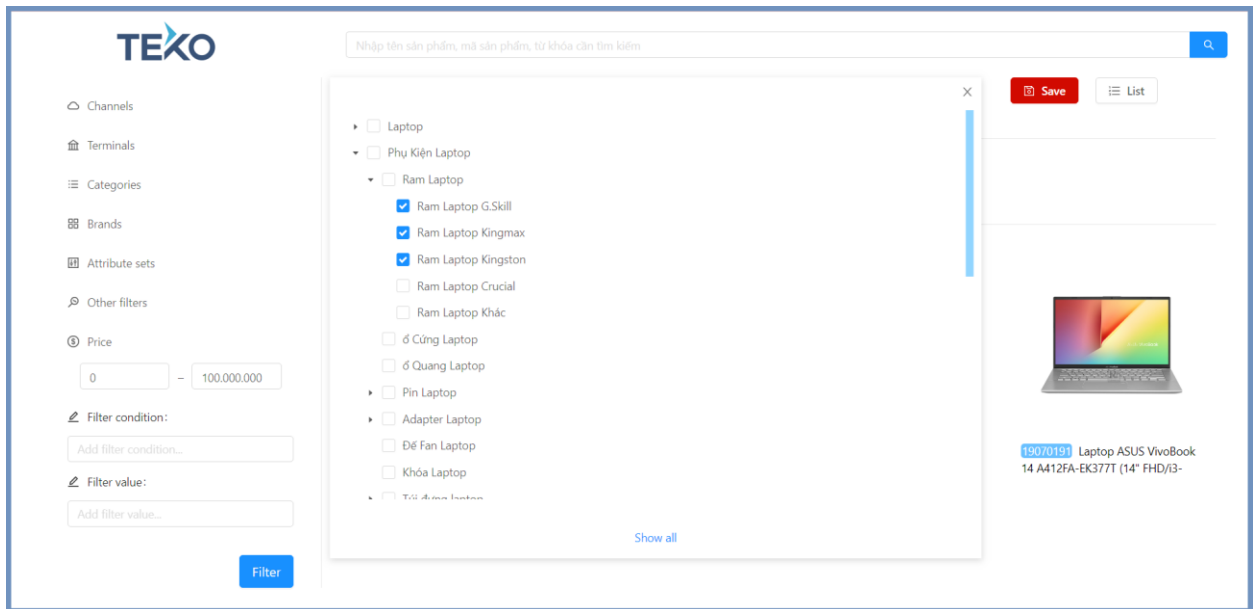
Các thành viên thực hiện riêng công việc sau đó gộp các thành phần với nhau, sử dụng Git để quản lý phiên bản.

c. Kết quả

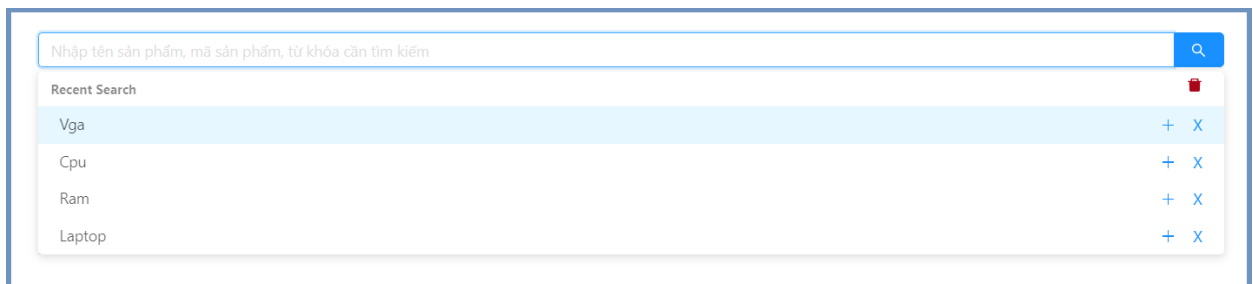
Dưới đây là một số giao diện của hệ thống:



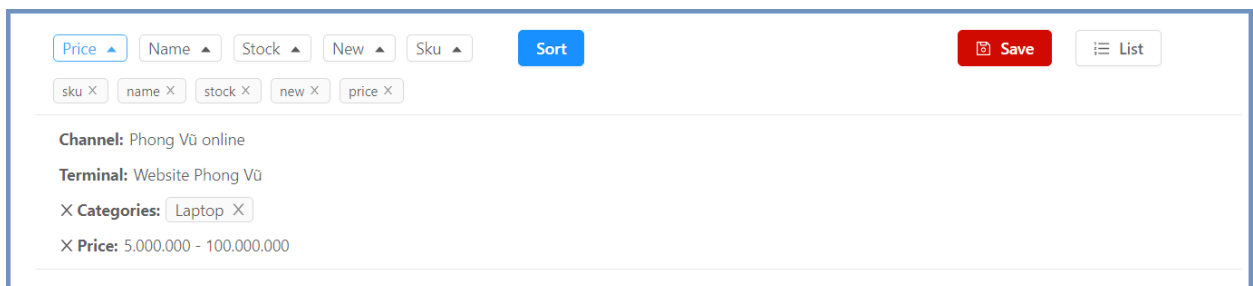
Hình 14: Giao diện hiển thị kết quả.



Hình 15: Giao diện Filter.



Hình 16: Thanh tìm kiếm.



Hình 17: Giao diện Sort và thông số Filter, lưu danh sách tìm kiếm

V. Tổng kết

- Qua 2 dự án đã làm, em đã có thêm rất nhiều kiến thức và kỹ năng làm việc cá nhân cũng như làm việc nhóm.
- Các hệ thống được phát triển đạt đủ các yêu cầu đặt ra.
- Các hệ thống đã hoàn thành và chạy ổn định trên môi trường phát triển.

* Cấu trúc báo cáo tham khảo từ: LinhPV__BCTK_190917

* Các nguồn tham khảo lý thuyết:

<https://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html>

<https://www.mulesoft.com/resources/api/what-is-an-api>

...và một số đoạn trích khác.



Ý kiến đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

Hà Nội, ngày _ tháng 9 năm 2019.
Người hướng dẫn
(Ký, ghi rõ họ tên & dấu công ty)

Nguyễn Phúc Lợi



Ý kiến đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

Điểm số: Điểm chữ:

Hà Nội, ngày _ tháng 9 năm 2019.

Giảng viên đánh giá
(Ký, ghi rõ họ tên)

Hà Quang Thụy