# Quy tắc code Front End

## Tổ chức thư mục:

### Tổ chức thư mục vue

#### Cách đặt tên tuân theo : **Pascal Case** (CpListUser.vue)

Đối với tên thư mục ta sẽ đặt tên theo dạng

Graphical user interface, application

Description automatically generated

Page sẽ được cấu trúc từ 2 view chính là Admin và user

Admin: sẽ là toàn bộ phần thiết lập cơ sở bao gồm các quyền thiết lập của admin và giáo viên

User: sẽ là page người dùng, nơi tháo tác các chức năng học và thi

Tên thư mục page sẽ được viết dưới dạng [page-descript]

Tên file sẽ được viết theo **Pascal Case**

Component page:

Cách đặt tên:

Tên thư mục page sẽ được viết dưới dạng [page-descript]

Đặt tên file theo quy tắc **Pascal Case** nhưng đính kèm Cp=> [CpName]

Component common:

Tên thư mục common sẽ được viết dưới dạng [page-descript]

Đặt tên file theo quy tắc **Pascal Case** nhưng đính kèm Cm=> [CmName]

\*\*Ngoài ra ta còn có thêm 2 thư mục trong file common-extention là

Handling: sẽ được đặt tên theo kiểu Cmh=> [CmhName]

* Là nơi đang xây dựng hoặc đang chỉnh sửa component common đã tồn tại và chưa được qua tích hợp
* Sau khi tích hợp được sẽ được đẩy lên common

Integrate: sẽ được đặt tên theo kiểu Cmi=> [CmiName]

* Là nơi để các componnet common sau khi xử lý và qua tích hợp nhưng không thể tích hợp thêm vào component common

Ta có thêm Component layout: đặt tên theo kiểu Cl=> [ClName]

* Là nơi cấu hình các common cho view layout

#### Cách xây dựng:

Một VuePage sẽ được xây dựng dựa trên các component page và tương tự các component page sẽ được xây dựng nên từ các component common

### Tổ chức thư mục style

Cách đặt tên:

Thư mục: Tên thư mục styles sẽ được viết dưới dạng [folder-descript]

Thư mục file: Tên thư file styles sẽ được viết dưới dạng [file-descript.scss]

Tên file biến:

Common: Tên thư file variable common sẽ được viết dưới dạng [\_file-name.cm.scss]

Config: Tên thư file variable config sẽ được viết dưới dạng [file-name.scss]

\*\*Lưu ý:

Các biến trong thư mục common sẽ được dùng để config cho common component

Các biến trong thư mục config dùng để config cho global

Các biến trong thư mục lib dùng để config cho các thư viện

Sau khi tạo các file variable ta khai báo vào trong file variable global: \_global.scss



Khi ta muốn sử dụng các biến config ở các file vue ta chỉ cần sử dụng gọi file \_global.scss và sử dụng thuộc tính @use

Text, logo

Description automatically generated

Đối với các file css ta cũng sẽ có thư mục global tương tự là style-global.scss

Text

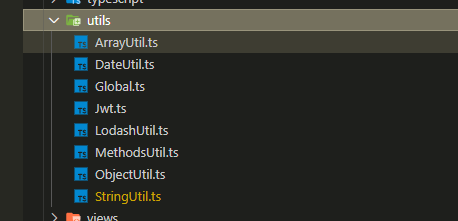
Description automatically generated

Để sử dụng cả biến cả scss ta có thể sử dụng file này bằng từ khóa @use thay cho \_global.scss

@use "@/styles/style-global.scss" as \*;

### Tổ chức Store

### Tổ chức function utils



Trong các file ta xây dựng theo hướng class và các thuộc tính hay phương thức muốn cung cấp thì thông qua biến static như sau

Text

Description automatically generated

## Api

A screenshot of a computer

Description automatically generated with medium confidence

Api sẽ được đặt tên thư mục và tên file như trên

Thư mục api sẽ được chia thành các thư mục con tuân theo các service trên BE

Trong từng file ta sẽ khai báo Tên service như sau

Tuân theo kiểu pascal và <Tên thư mục>Service

Ex: Trong thư mục service user ta sẽ khai báo service: UserService

Các key đường dẫn Api cũng được đặt tuân theo kiểu pascal case như sau

Text

Description automatically generated

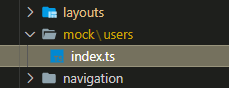
Muốn sử dụng khi call api ta làm như sau

*const* res = await MethodsUtil.requestApiCustom(ComboboxService.GroupProficiency, TYPE\_REQUEST.GET).then((*value*: *any*) *=>* *value*)

Sử dụng phương thức requestApiCustom

ComboboxService.GroupProficiency=> Chính là đường dẫn mà ta đã cài đặt ở service ComboboxService

\*\*Mock Api



Ta vẫn tổ chức thư mục mockApi như thư mục api

Text

Description automatically generated

Data mock sẽ được trả về thuông qua phương thức fetchData được cài đặt trong file mock của từng service

Nó sẽ check đường dẫn để trả về data như mình đã cài đặt