Code convention, Quy tắc code

* **Ident**

Tab với kích thước bằng 2 lần dấu cách (Space),

ví dụ trong Visual Code, đặt tab size = 2

Graphical user interface, text

Description automatically generated

* **Code trong 1 component function**

Code trong 1 component function chia thành các mục sau:

States and Variables: chứa khai báo

Processing: chứa các tính toán xử lý, hoặc React.useEffect

Sub Components: chứa các thành phần con, các hàm bắt đầu bằng từ render, ex: renderButtonOk

Event Handlers: chứa các xử lý sự kiện, các hàm bắt đầu bằng từ on, tiếp tới là hành động, tiếp tới là kiểu đối tượng, cuối cùng là tên đối tượng, ex: onClickButtonOk

Main Render: viết render cuối cùng của file.

EX:

const CalendarView = (props) => {

/\*\*\* States and Variables \*\*\*/

Const [currentDate, setCurrentDate] = React.useState(null);

/\*\*\* Processing \*\*\*/

React.useEffect(()=>{

}, [props.isOpen]);

/\*\*\* Sub Component \*\*\*/

const renderButtonOk = () => {

}

/\*\*\* Event Hanlders \*\*\*/

const onClickButtonOk = (e)=>{

}

/\*\*\* Main Render \*\*\*/

return <div />

}

* **Code trong 1 file**

Code trong 1 file phải được cấu trúc như sau

import : chứa tất cả import

propTypes: chứa định nghĩa properties

defaultProps: chứa giá trị mặc định của properties

component function: cài đặt component

liên kết component với định nghĩa trong propTypes, defaultProps

export: export component

EX

Text

Description automatically generated

Để đơn giản hoá, sửa code snippet của javascript (đối với visual code), như sau

Text

Description automatically generated

{

// Place your snippets for javascript here. Each snippet is defined under a snippet name and has a prefix, body and

// description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. Possible variables are:

// $1, $2 for tab stops, $0 for the final cursor position, and ${1:label}, ${2:another} for placeholders. Placeholders with the

// same ids are connected.

// Example:

// "Print to console": {

// "prefix": "log",

// "body": [

// "console.log('$1');",

// "$2"

// ],

// "description": "Log output to console"

// }

"Create Component": {

"prefix": "component",

"body": [

"import React from 'react';",

"import PropTypes from 'prop-types';",

"const propTypes={};",

"const defaultProps={};",

"const ${1:Component} = (props) => {",

"\t/\*\*\* States and Variables \*\*\*/",

"\t/\*\*\* Processing \*\*\*/",

"\t/\*\*\* Sub Components \*\*\*/",

"\t/\*\*\* Event Handlers \*\*\*/",

"\t/\*\*\* Main Render \*\*\*/",

"\treturn $0;",

"}",

"$1.propTypes = propTypes;",

"$1.defaultProps = defaultProps;",

"export default $1;"

]

}

}

* **Cấu trúc thư mục trong một project**

- Các component được dùng chung cho nhiều project được đặt trong thư mục src/components, chỉ được lấy các component ra sử dụng, không được sửa. (1)

- Các component có khả dùng cho các project khác hoặc các component (1) mà muốn sửa style css thì gọi là base componet được đặt trong thư mục src/base\_components (2)

Các components này chỉ được gọi lẫn nhau hoặc dùng lại (1)

- Các component đặc trưng cho 1 app, được đặt trong thư mục src/app\_components (3),

Các components này được phép truy cập cơ sở dữ liệu của app (nếu cần, các truy cập này nên đơn giản và không quá phức tạp), sử dụng lại (1), (2) hoặc gọi lẫn nhau

- Các trang của 1 web được đặt trong thư mục src/pages (4). Các trang này được sử dụng components (1), (2), (3). Hoặc có thể dùng lẫn nhau

* Khi có đề xuất thêm hoặc thay đổi component trong (1), liên lạc tới Việt. ^^
* Các component trong (1) được hiển thị trên trang localhost:3000/test, bổ sung thêm component mới thì hay viết thêm component đó vào trang test này, và ko deploy trang test này
* Lưu các biến CSS toàn cục trong file App.css
* ?