**- Tránh chia nhỏ component không cần thiết**

**Về RequireInvalidInput Component**

+ Khai báo const và in hoa cho regex (EMAIL\_REGEX, CONTACT\_REGEX)

+ Func mailAndContactIsValid chỉ xử lý 2 condition

nên convert to if return (thay vì if-elseif)

+ Dùng return không đúng trong if else

**Nói chung**

- props nên đặt ở đầu component

- không cần thêm inputContent nên lấy value

- các hàm xử lý nên đặt trong **useEffect** với điều kiện state thay đổi

tránh việc call for each render hoặc TH các state khác không liên quan thay đổi

Mẫu:

**useEffect(() >=> {**

**nameisValid();**

**), [value]); //Điều kiện value thay đổi thì useEffect() được thực thi**

- Đặt các state value và hàm validate , change value tại một file riêng tạo sự thuận lợi khi muốn sử dụng các state value cho các mục đích khác

- Cách tiếp cận khác cho ẩn hiện (display: none/’ ’):

**<>**

**{**

**props.displayMode === ‘none’ && (chỉ render ra các component khi nhấn EDIT)**

**}**

**{**

**Props.displayMode !== ‘none’ && (chỉ render ra các component khi nhất SAVE)**

**}**

**{**

**Props.displayMode === ‘none’ ? <EDIT> : <SAVE>**

**}**

**</>**

- Các giá trị default nên khai báo chung tại một file

- Cách handle khác trong select (comboBoxComponent)

**const content = ['-Select-', 'Male', 'Female', 'Others', '...']**

**const content = {**

**101 : Invalid phone number //key: value**

**}**

**101 --- Invalid phone number**

**404 ---**

**setValue(content[e.target.value]**

- button component có thể gộp lại

**Cấu trúc chuẩn**

src

|\_\_images

|\_\_utility

| |\_\_validators.js

| |\_\_validationMessage.js

|\_\_components

| |\_\_Form

| |\_index.js

| |\_styles.scss

|\_\_shareComponents

|\_Button

| |\_\_index.js

| |\_\_styles.scss

|\_Header

| |\_\_index.js

| |\_\_styles.scss

|\_Input

|\_\_index.js

|\_\_styles.scss

Link project tham khảo: https://github.com/lhduyft15/react-training