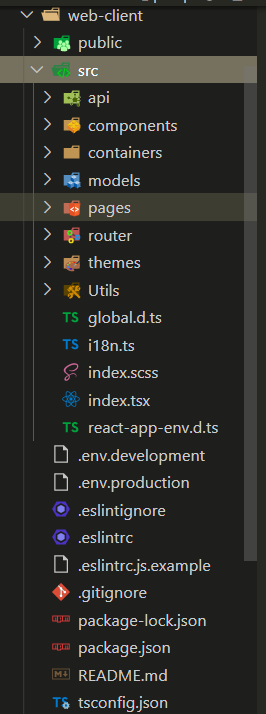
**Phân tích về structure của một project React**

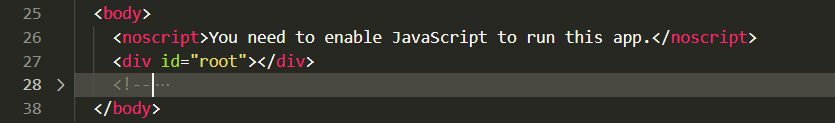
**(Web-client)**

* ****Api : Thư mục API là nơi để chứa những phương thức liên kết giúp trao đổi, xử lý dữ liệu giữa client với server như: GET, POST, PUT, PATCH, DELETE
* Component : Là thưu mục chứa các thành phần của React có thể liên kết với nhau hoạc tái sử dụng tạo nên giao diện người dùng
* Containers : Là nơi lưu trữ các tài liệu chính của website có chức năng tiếp nhận và xử lý dữ liệu bên phía server và client rồi hiển thị ra thông tin cho người dùng thông qua thư mục Pages
* Models : Là thành phần lưu trữ định dạng dữ liệu của các đối tượng được sử dụng trong trang web
* Pages : Là nơi hiển thị chính ra cho người dùng được cấu tạo từ các thành phần containers
* Router : Là thành phần xử lý điều hướng phía client, là thành phần để xác định xem trang nào sẽ được hiển thị bên phía người dùng
* Themes : Là thư mục chứa các mã màu để custom giao diện người dùng
* Utils : Là nơi chứa thông tin về các kiểu dữ liệu đã được fomat
* Package.js : Chứa thông tin dữ liệu về các phiên bản các thư viện
* Public : Là thư mục tĩnh chứa các thông tin về hình ảnh, ngôn ngữ và các tài liệu HTML chính cho người dùng
* Node-module : Là nơi chứa các thư viện của các bên thứ 3 cung cấp được sử dụng trong trang web

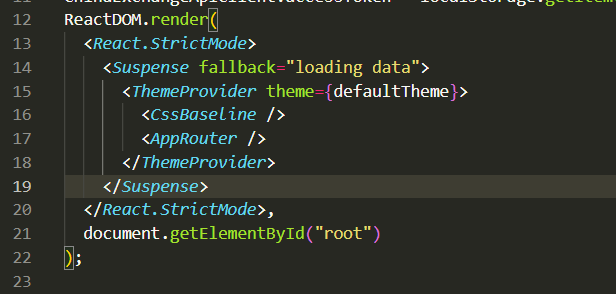
**Các phần nghiên cứu về project.**

1. **Rendering Elements**

\_ Tất cả các elements đều được render vào 1 thẻ <div>…</div>



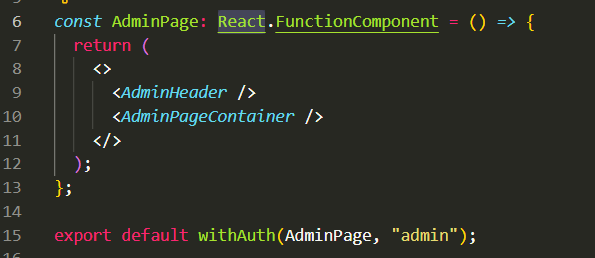
và được quản lý bằng ReactDOM



1. **Components**

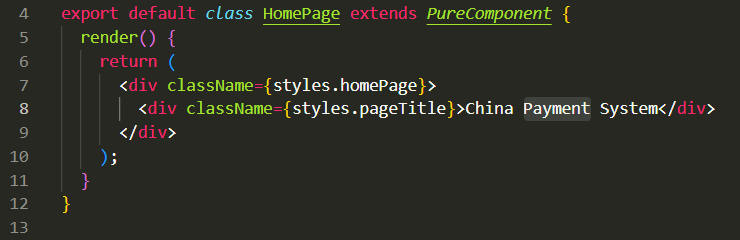
**\_** Component là các khối UI độc lập của trang web có chức năng xử lý những tác vụ riêng biệt.

\_ Các component có thể được tái sử dụng hoặc kết hợp tạo thành 1 component lớn hơn

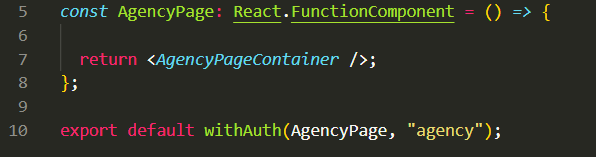


\_ Component được chia làm 2 dạng:

* *Class component*



* *Function component*

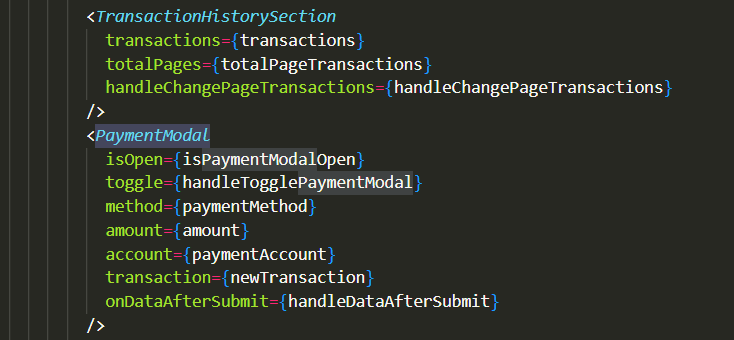


1. **Props**

\_ Props là nơi lưu trữ giá trị của thuộc tính thẻ

\_ Giá trị của props khi được truyền vào sẽ không thể thay đổi

\_ Props còn dùng để truyền dữ liệu từ component cha đến các component con của nó

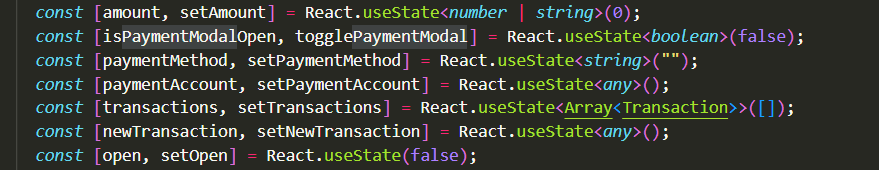


1. **State**

**\_** State được dùng để chứa dữ liệu của 1 component và chỉ tồn tại trong phạm vi của component

\_ State có thể được thay đổi bằng setState

\_ Khi state bị thay đổi thì component sẽ được Re-render



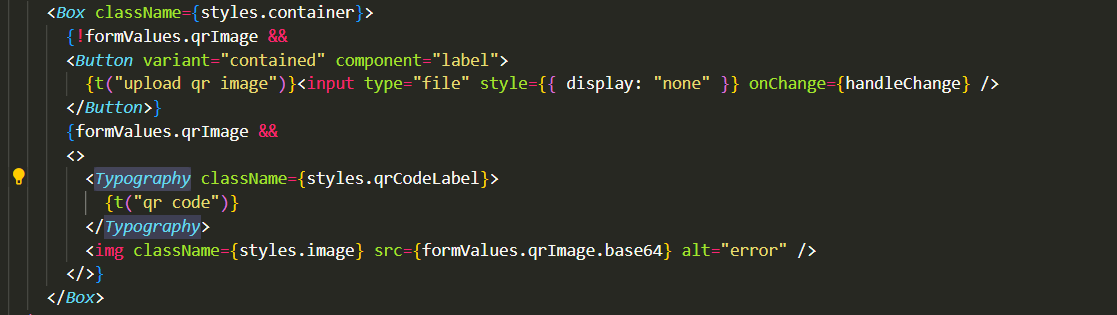
1. **Handling Events**

\_ Là các function để bắt sự kiện sảy ra trong component như : click,submit,hover,change,press,…



1. **Conditional Rendering**

\_ Là hoạt động render có kết hợp sử lý điều kiện (if else/ && / ? : ) để cập nhật giao diện người dùng

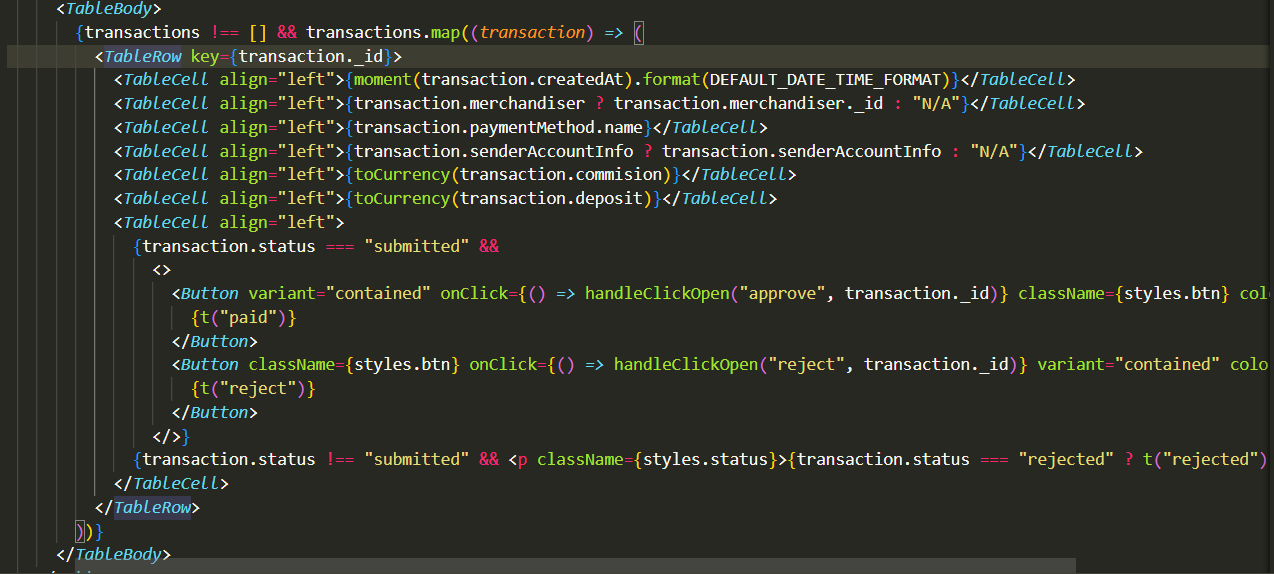


1. **Lists and Keys**

\_ Khi render 1 danh sách các phần tử trong mảng (map) ta có thể dùng key để đánh dấu

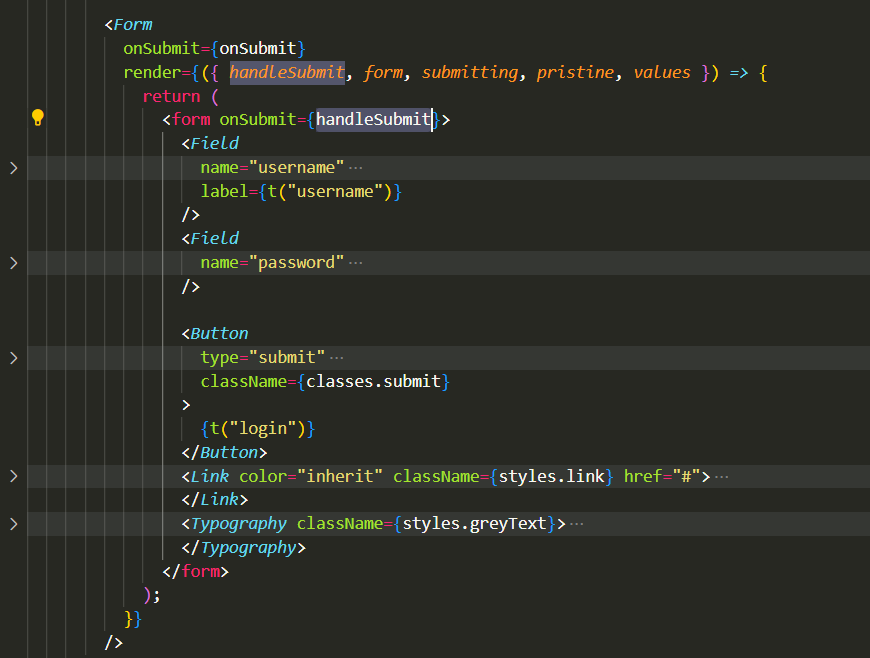
\_ Key giúp react dễ dàng theo dõi các phần tử nhanh chóng xác định được những phần tử nào được thay đổi , thêm hoặc xóa trong mảng

\_ Key thường được gán bằng index hoặc chính \_id của phần tử được render

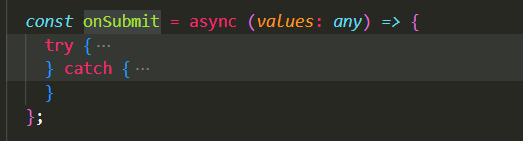


1. **Forms**

\_Thường là nơi tiếp nhận các dữ liệu từ input



\_Dữ liệu nhận vào sẽ được sử lý dữ liệu thông qua các hàm



1. **Lifting State Up**

\_ Là việc chuyển state từ component con lên component cha để các component con khác có thể dễ dàng truy cập và sử dụng

\_ Khi cần thay đổi component con sẽ dùng callback từ component cha qua props để thay đổi state của component cha

1. **Composition**

\_ Là việc tạo ra 1 component cha bằng cách sử dụng kết hợp các component con lại với nhau việc này cho phép quản lý các component trở nên dễ dàng hơn

**Giải pháp cho project**

1. Không nên lưu accessToken ở dưới loaclStorage

