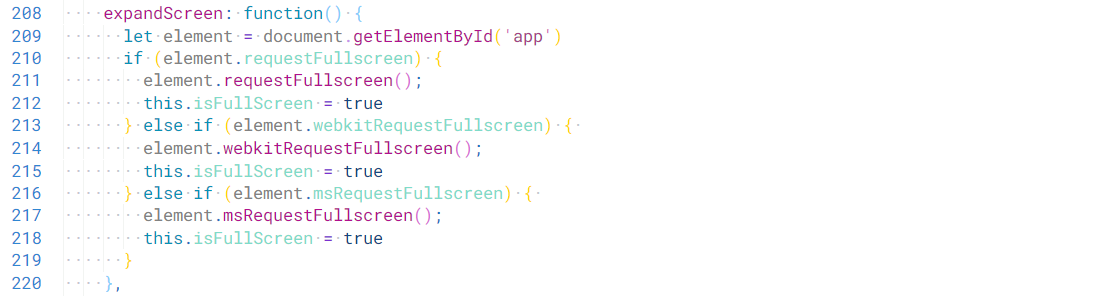
# Method Ref trong Vue

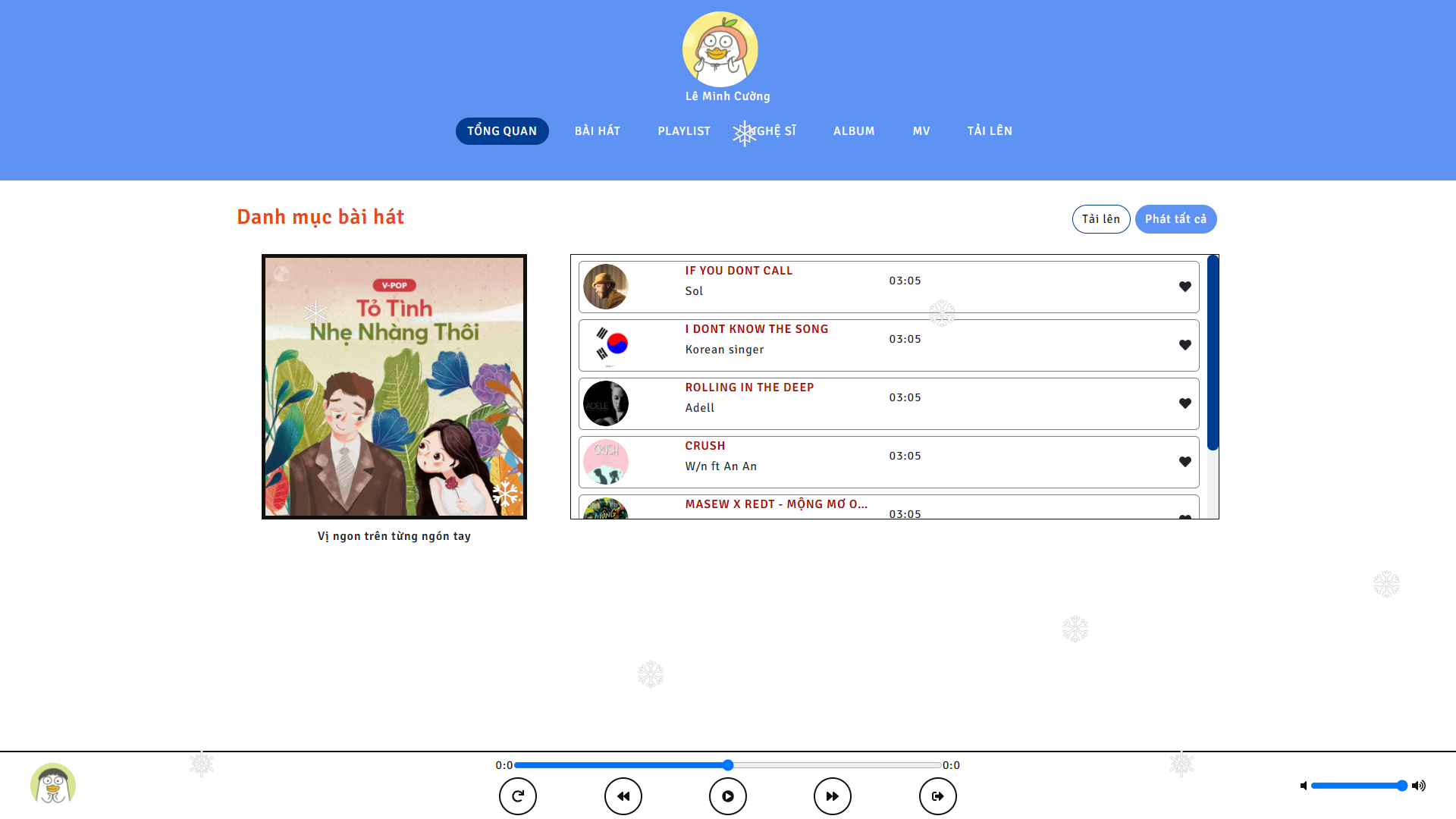
Trong Vue 3 chúng ta có thể tạo biến hoặc giá trị bất kì có thể reactive được (nghĩa là tương tác được). Bởi vì một biến tạo đơn thuần sẽ không một khi đã render thì sẽ không thể cập nhật giá trị của chúng. Vậy nên Vue 3 hỗ trợ cho chúng ta một method là ref.



Như vậy thì chúng ta sử dụng các biến này trong methods thì hoàn toàn có thể cập nhật giá trị của chúng.

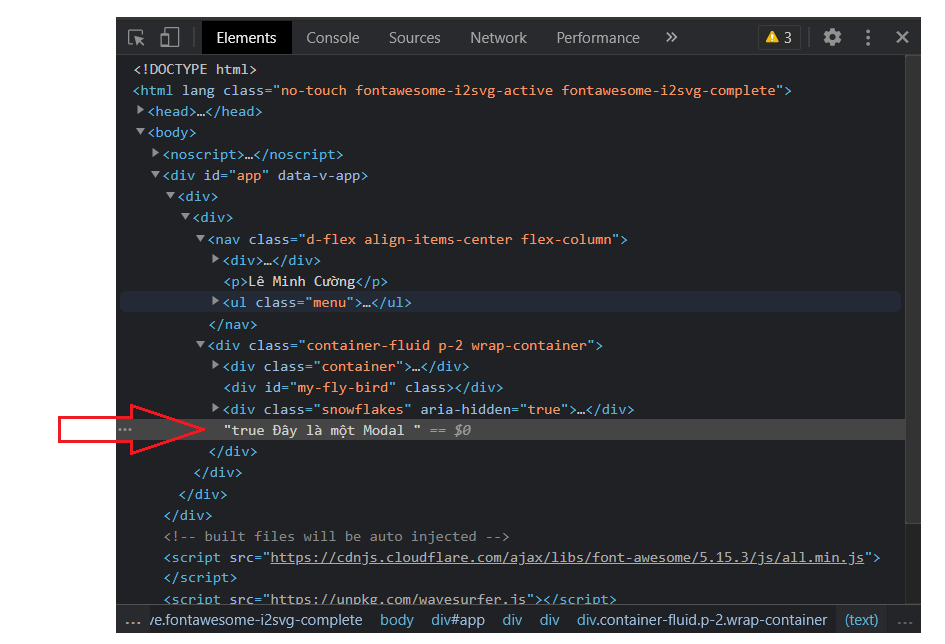


Ở đây, chúng ta muốn thay đổi giá trị của biến isFullScreen thì chỉ cần ghi giá trị true hoặc false là đã hoàn tất quá trình update. Rất đơn giản và tiện lợi. Kết quả như hình dưới đây.



# Teleport

Vue khuyến khích chúng ta xây dựng UI bằng việc đóng gói UI and những hành động liên quan thành một components. Chúng ta có thể đặt 1 component trong những component khác để xây dựng một cây cấu trúc để tạo nên application UI như thế này.

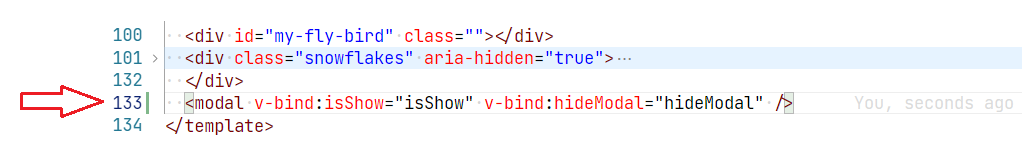


Chúng ta đặt Modal Component nested vào những Component khác. Kết quả để hiển thị cho biết rằng Modal được nằm trong những components khác.

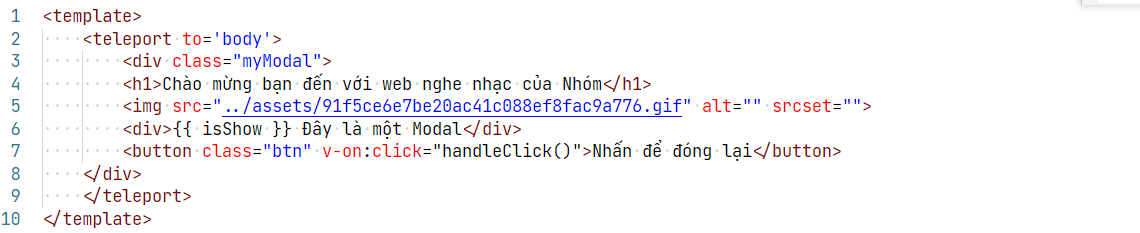
Tuy nhiên, thỉnh thoảng, một phần template của component thì thuộc bản than component đó, một phần khác nằm ở bất cứ nơi khác trong DOM, ngoài Vue app. Nếu như vậy thì dòng chữ “true Đây là một Modal” phải nằm ở thẻ body. Tách biệt hoàn toàn với Vue App.

Trong hầu hết các trường hợp thì chúng ta muốn logic Modal sẽ nằm trong cùng Component, nhưng chúng ta muốn code lại vị trí của modal bằng position: absolute điều đó nhanh chóng trở nên khó khăn để mà code CSS. Như chúng ta đã biết để định vị được một thẻ child position absolute như ý thì chúng ta phải xác định được thẻ parent là một position relative. Như vậy càng nested Modal nhiều vào thì sẽ có rất rất nhiều thẻ div có thuộc tính position: relative. Làm cho chúng ta khó có thể đặt child vào đúng vị trí mong muốn. Như vậy cần có một cách đưa Modal nằm ngoài body.

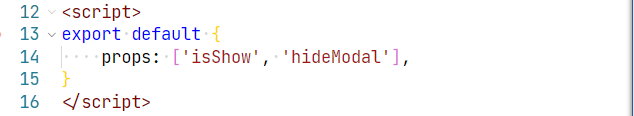
Bậy giờ chúng ta sẽ xem lại đoạn code ở phần Modal đó



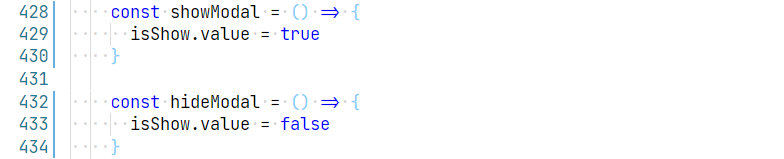
Nội dung trong file Modal Component



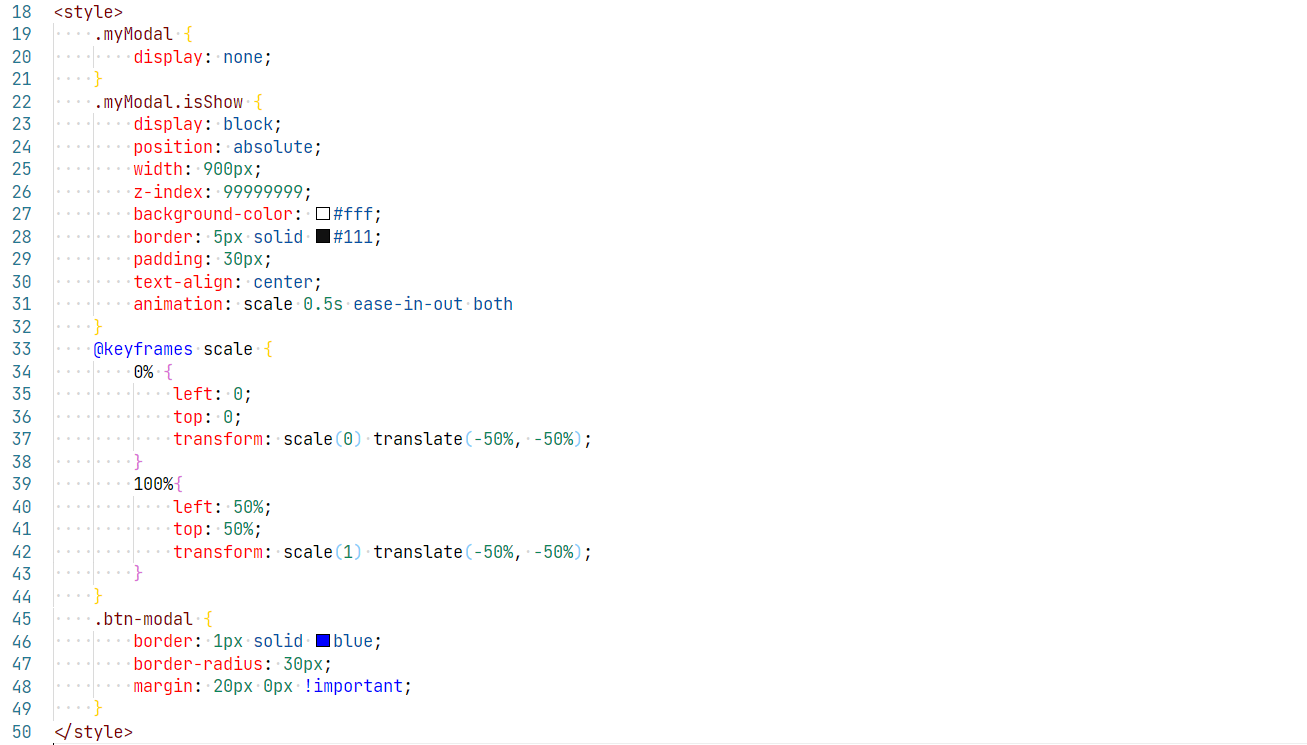
Truyền props từ Homepage.vue vào Modal.vue để component cha luôn luôn quản lý state của con như vậy sẽ dễ dàng sử dụng props hơn



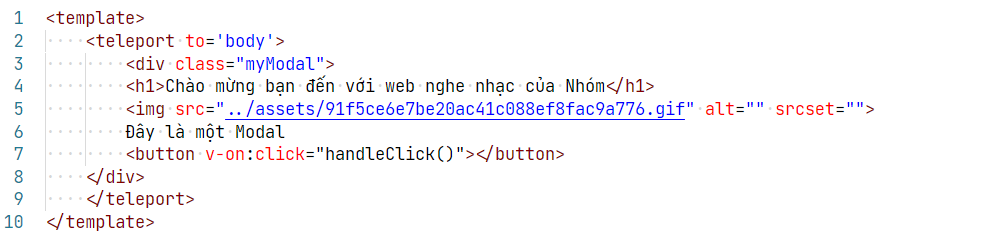
Code bật tắt Modal nằm trong Homepage.vue



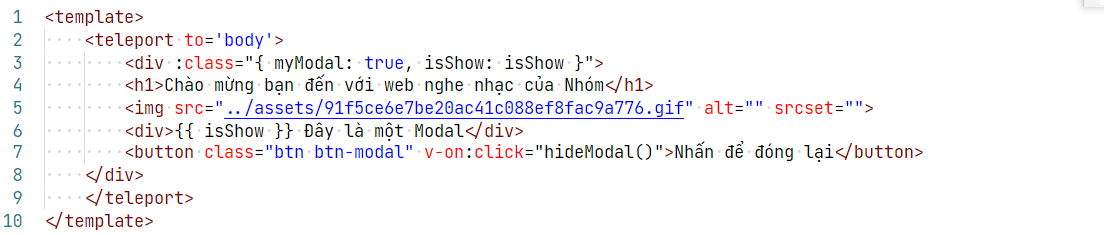
Bổ sung CSS vào cho Modal



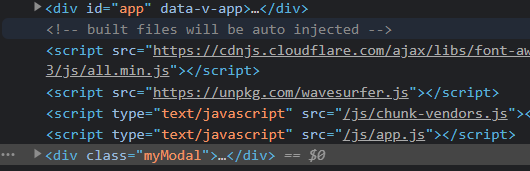
Bây giờ chúng ta sẽ thực hiện bọc cả modal component bằng thẻ teleport



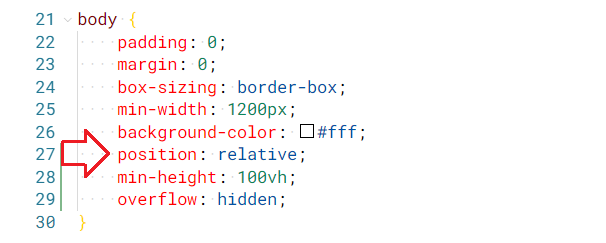
Thêm chút điều kiện để khi isShow là true thì hiện và isShow là false thì tắt. Ở class thì bind chúng lại. Còn Button thì add thêm props function hideModal được truyền từ component cha Homepage.vue vào Modal.vue.



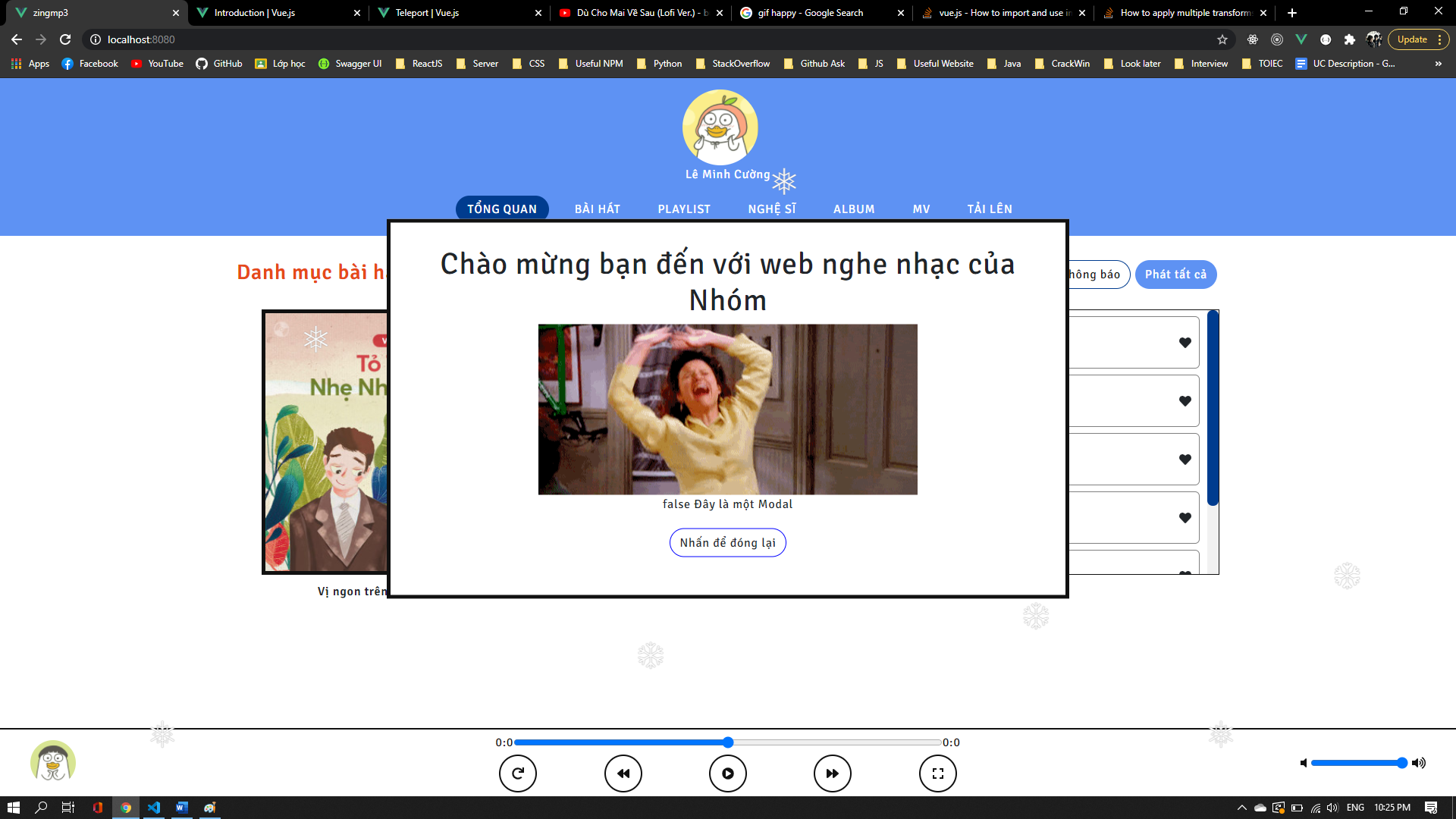
Kết quả DOM ngay lập tức di chuyển Modal Component nằm trong thẻ body và Ngoài app Vue



Sau đó chúng ta sẽ thêm CSS vào thẻ body là position relative do là Modal đã được dời ra ngoài + position relative nữa.



Kết quả là:

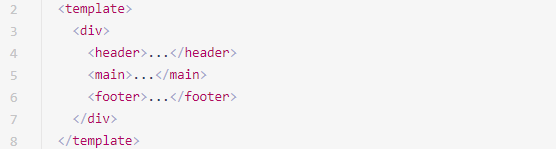


Thật tuyệt vời. cuối cùng chúng ta cũng sử dụng được Modal

# Fragment

Trong Vue 3, template tag hỗ trợ gói nhiều component root với nhau

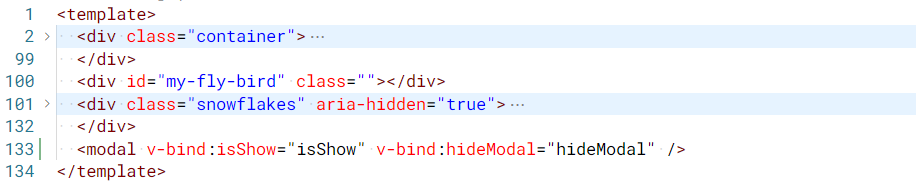
Đối với phiên bản 2.x



Nếu chúng ta muốn sử dụng những component như là header, main, footer nằm trong template tag thì buộc lòng chúng ta phải bọc thêm thẻ div nữa nếu không sẽ có cảnh báo. Như vậy là nhiều components sẽ phải được bọc trong thẻ div để mà fix lỗi này

Đối với phiên bản 3.x

Components bây giờ có thể có nhiều root nodes.



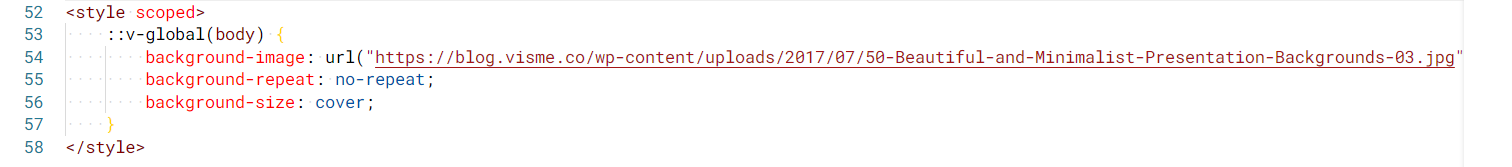
# Style Scope

Một tính năng mới trong Vue 3 đó là Style trong single file. Nếu trong Vue 2 chúng ta có thể code CSS nằm trong cùng một Component. Nhưng bây giờ Vue 3 đã hỗ trợ nhiều hơn thế, chúng ta có thể code global rule CSS hoặc là CSS cho slot.

Thứ nhất, sau đây là ví dụ như trong file Modal.vue chúng ta muốn background là một hình ảnh gì đó mới mẻ mà không trực tiếp dùng file CSS để code. Vì đây chỉ là demo về sự mạnh mẽ của style scope nên chúng ta sẽ không bàn về ý nghĩa của đoạn code trên đối với cả ứng dụng

**Cú pháp:**

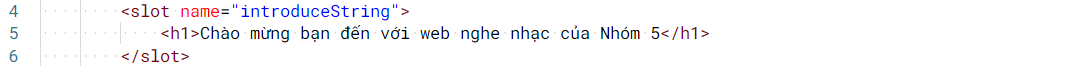
**Ví dụ:**



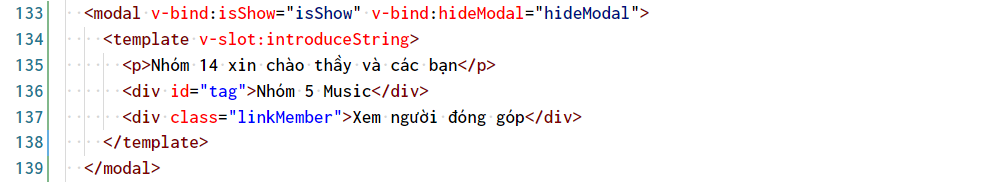
Thứ hai, chúng ta muốn tác động vào slot content của chính Component đó. Bây giờ chúng ta muốn là chúng ta có thể thay thế dòng trong Modal.vue thành một chuỗi nào đó được Component cha sử dụng thêm vào

<h1>Chào mừng bạn đến với web nghe nhạc của Nhóm</h1>

Giờ chúng ta sẽ đặt đoạn code đó trong slot với name là introduceString.



OK vậy giờ chúng ta sẽ thêm đoạn giới thiệu mới tại Homepage.vue nơi mà sử dụng Modal.vue



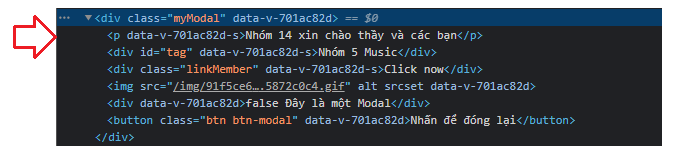
Bây giờ, chúng ta sẽ thực code css cho slot introduceString đó

**Cú pháp:**

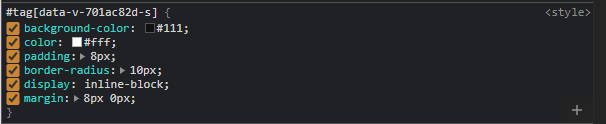
**Ví dụ:**



Tác động vào thẻ id là tag. Khi đó Vue sẽ tự động render cho chúng ta class và hiện kết quả như thế này



Slot sẽ được thêm 1 attribute là data-v-[xxxxxxxxxxxx] và khi đó các thẻ mà chúng ta muốn style CSS vào sẽ có dạng

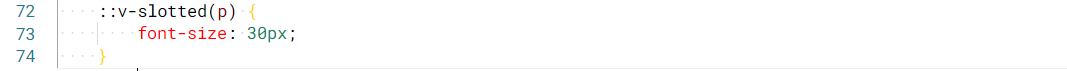


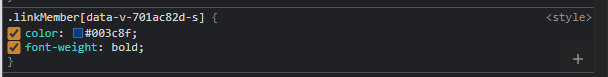
Tương tự với class linkMember





Tương tự với tag bình thường





Nếu chúng ta để ý thì tất cả các css strong slot sẽ có dạng

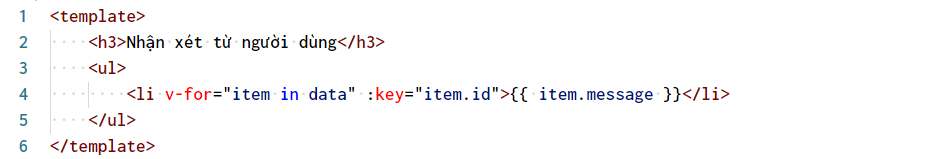
# Suspe**nse**

Suspense là một tính năng mới thử nghiệm và API về nó có thể thay đổi bất cứ lúc nào. Tài liệu được viết ở đây để cung cấp cho cộng đồng đóng góp phản hồi.

Thông thường đối components cần tiến hành một số asynchorous request trước khi mà chúng có thể được render. Những component này thường handle các request này bên trong chính chúng, và trong nhiều trường hợp thì nó đáp ứng một cách rất tốt.

Một cách dùng khác trong là Vue 3 cung cấp cho chúng ta suspense component

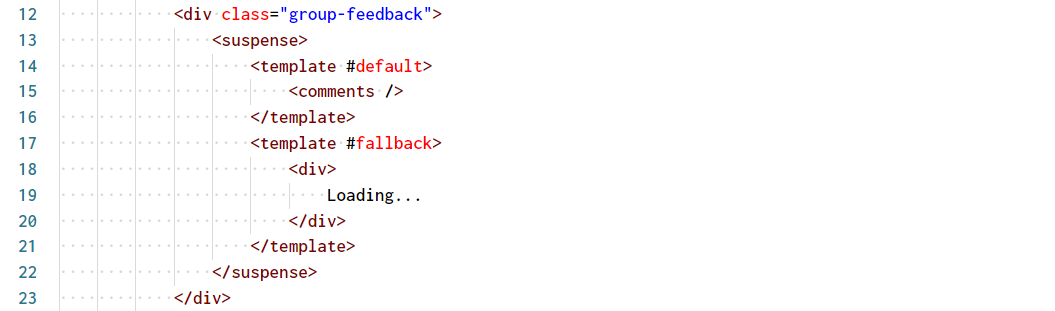
Bây giờ hãy sử dụng nó nhé. Giả dụ chúng ta muốn load một thêm những comment từ API về và nó nằm trong 1 Component có tên là Comments.vue

Chúng ta thêm async vào setup thì Vue mặc định sẽ hiểu là nguyên component của chúng ta là async component.

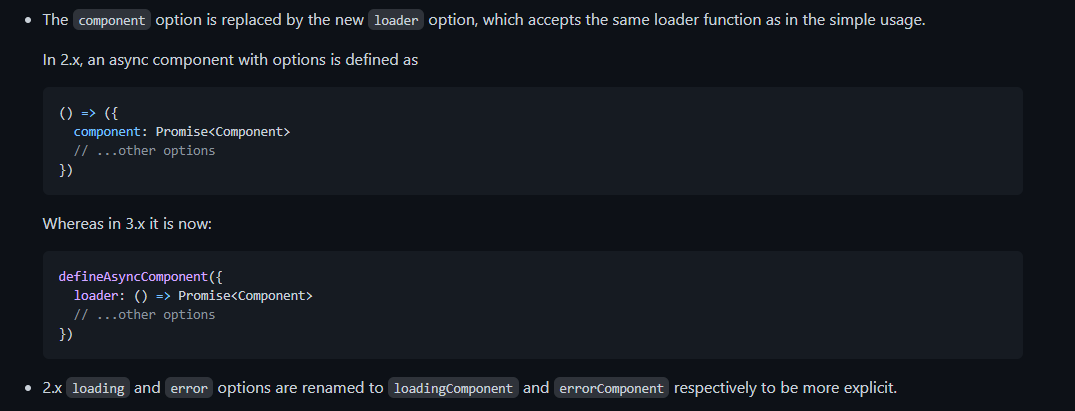
Ở ngoài Modal.vue



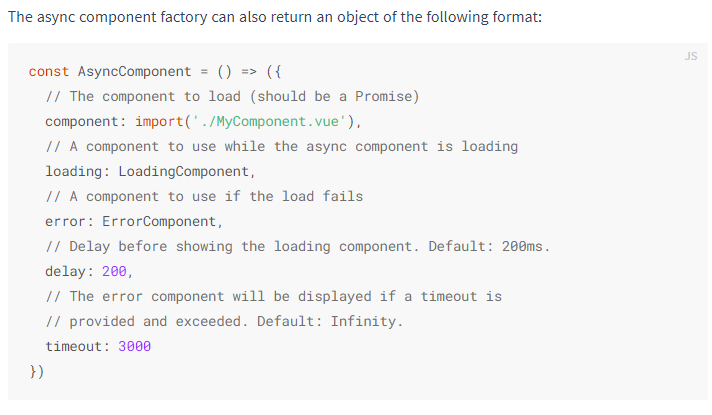


Như vậy thì khi mà component Comments của chúng ta chưa load xong thì nó sẽ hiện Loading mà thôi và sau khi nó load xong thì chúng ta sẽ hiện những Component. Tiết kiệm thời gian thời vì phải đợi tất cả các component được load xong.

Thay đổi trong Vue 2 và Vue 3 của defineAsyncComponent



Như vậy chúng ta có thể thấy rằng ở Vue 3 có thêm defineAsyncComponent tăng tính tiện lời khi sử dụng hơn và chúng ta có thể định nghĩa async component một cách rõ ràng.



<https://vuejs.org/v2/guide/components-dynamic-async.html>



<https://v3.vuejs.org/api/global-api.html#defineasynccomponent>.

# DefineAsyncComponent

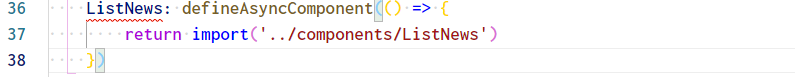
DefineAsyncComponent thì chúng ta không thể tủy chỉnh được props hoặc là slot cho loading component. Đó là một điều khá bất tiện. Vì vậy mà Suspense được tạo ra có mục đích làm mạnh mẽ hơn Async Components.

Ở trong Vue 2 để định nghĩa Async Component thì nó thực sự không rõ ràng nên Vue 3 đã thể hiện 1 cách rất dễ dàng bây giờ chúng ta hay Viết 1 trang News.vue sau đó bỏ 2 component ListNews.vue, ListNews2.vue vào cả 2 đều là Async Component

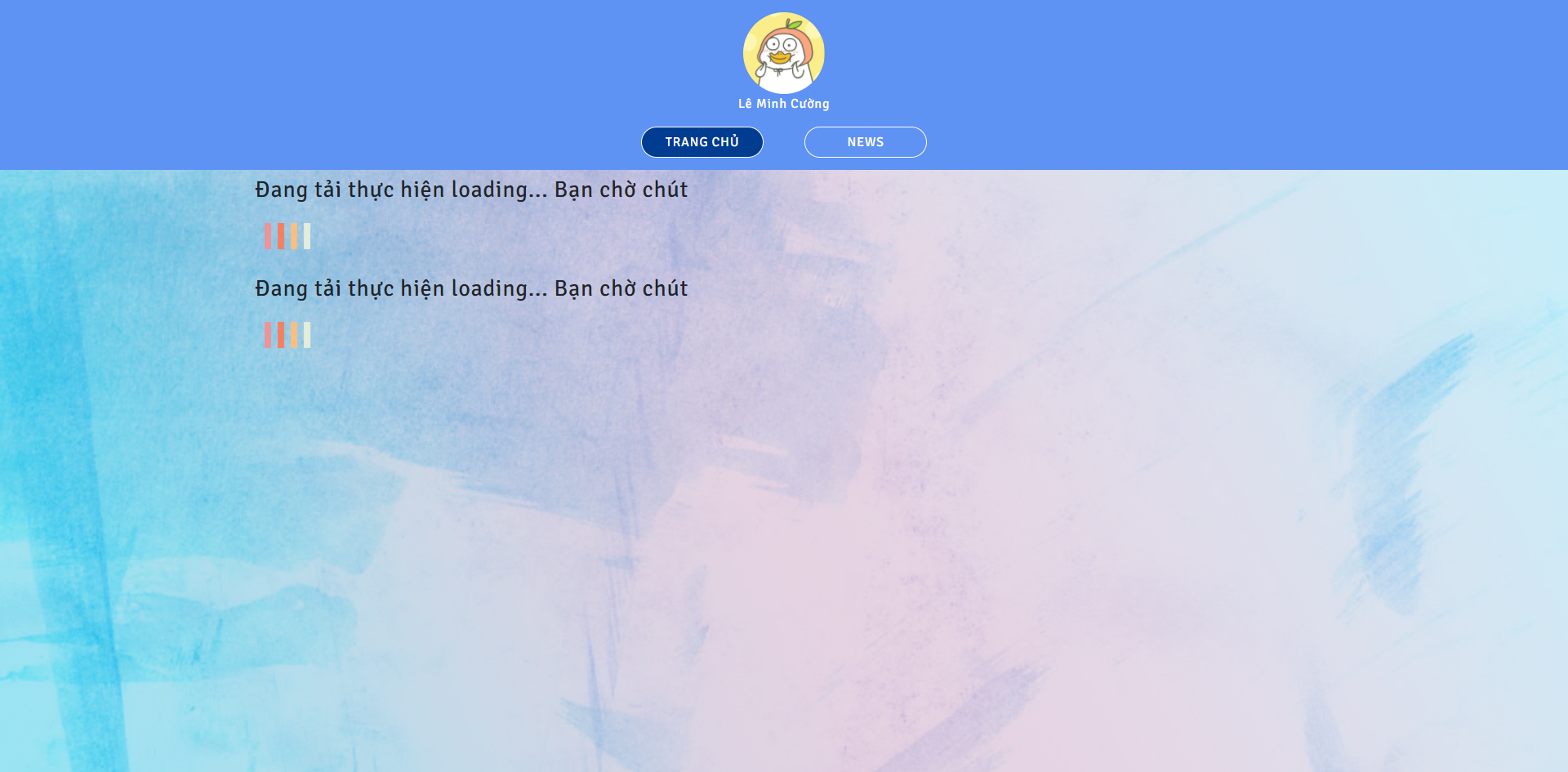


Chúng ta thêm tạo một component Async Component bằng cách return về 1 Promise và resolve import Component đó. Ở đây chúng ta thêm loadingComponent để có thể hiển thị thông báo khi mà Component kia chưa load xong.

Chúng ta có 2 cách, 1 truyền như hình trên 2 là



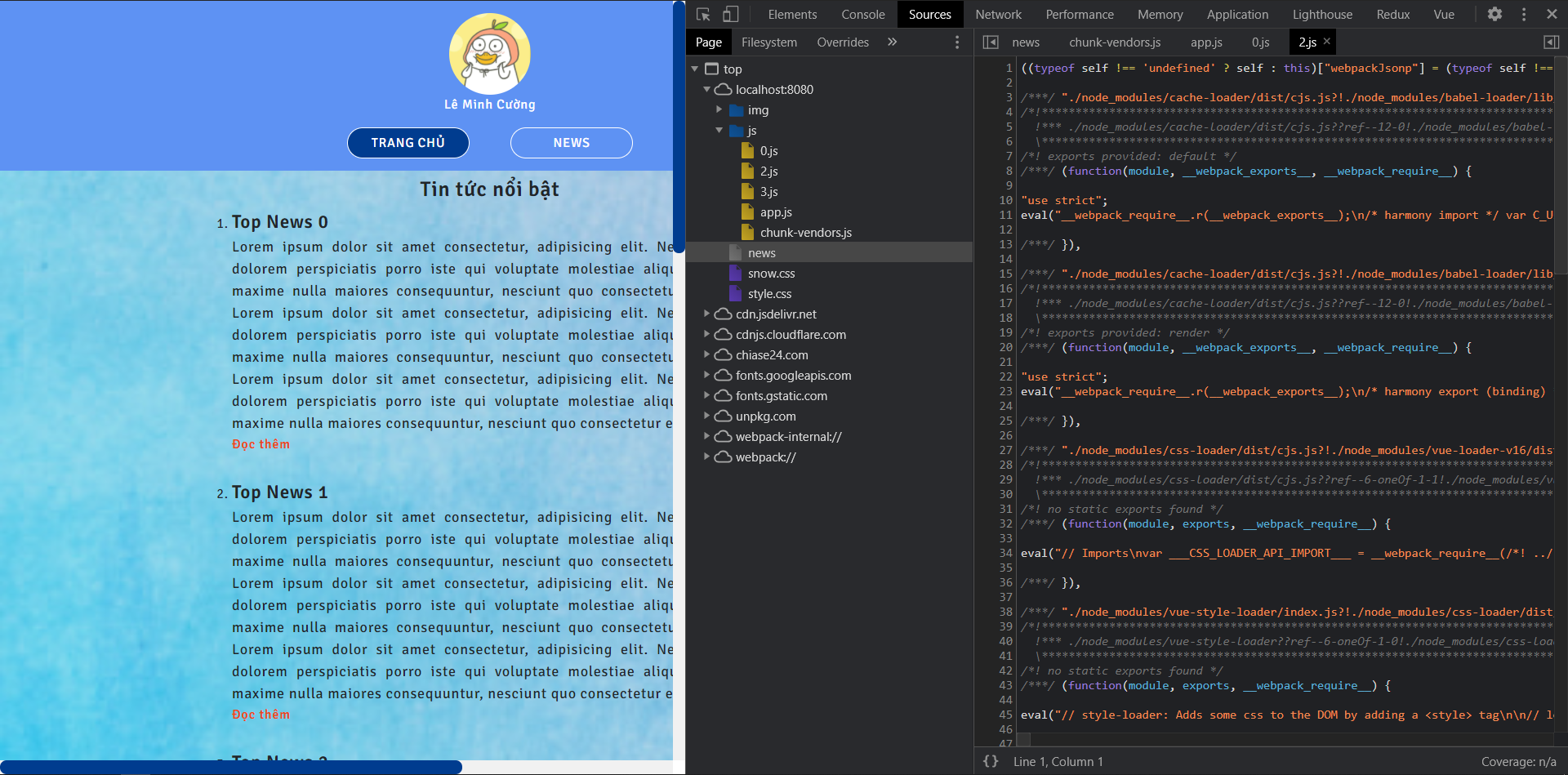
Cách 1 thì sẽ mở rộng được nhiều config hơn. Không quan trọng ListNews.vue, ListNews2.vue có gì giờ chúng ta sẽ thấy hiệu quả của chúng ở chỗ. Khi mà chúng ta vào trang News.vue thì 2 Component ListNews.vue, ListNews2.vue sẽ không load cùng với News.vue như vậy News.vue sẽ load nhanh hơn. Còn 2 Component ListNews.vue, ListNews2.vue sẽ trở thành dòng Loading



Tab source

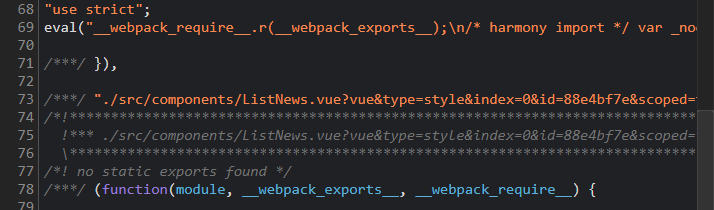
Như chúng ta có thể thấy lúc mà load xong trang News.vue thì chỉ có 2 file là app.js và chunk.

Một lúc sau:



Có thêm 3 file mới chúng ta quan tâm là 2.js 3.js

2.js



3.js



Đúng vậy nó được load sau khi mà News.vue được load xong

Bây giờ chúng ta hãy xem network của 2 file đó nhé

2.js sẽ được load trước khi mà 3.js load xong.