**Họ và Tên : Lê Sỹ Anh Tấn**

**MSSV : B2113342**

**Bài Thực Hành Số 3\_4**

**Phát triển Ứng dụng ContactBook**

**Phía FrontEnd sử dụng VueJS**

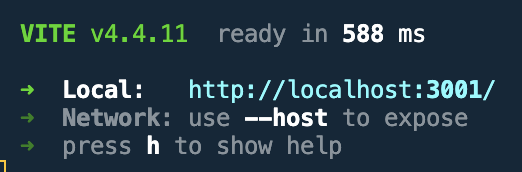
## 1.Tạo Ứng Dụng Vue

Bao gồm npm init vue@3 và chọn các tuỳ chọn là No trừ Vue-Router

Sau đó vào file vite.config.js để tuỳ chỉnh lại cổng đầu ra cho vueServer như sau :

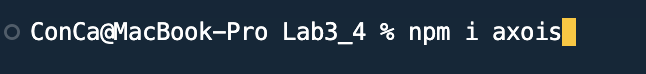


Sau khi đã config server port ta có thể gọi npm run dev để bắt đầu chạy server và lúc này server chạy ở cổng 3001 theo như ta config :



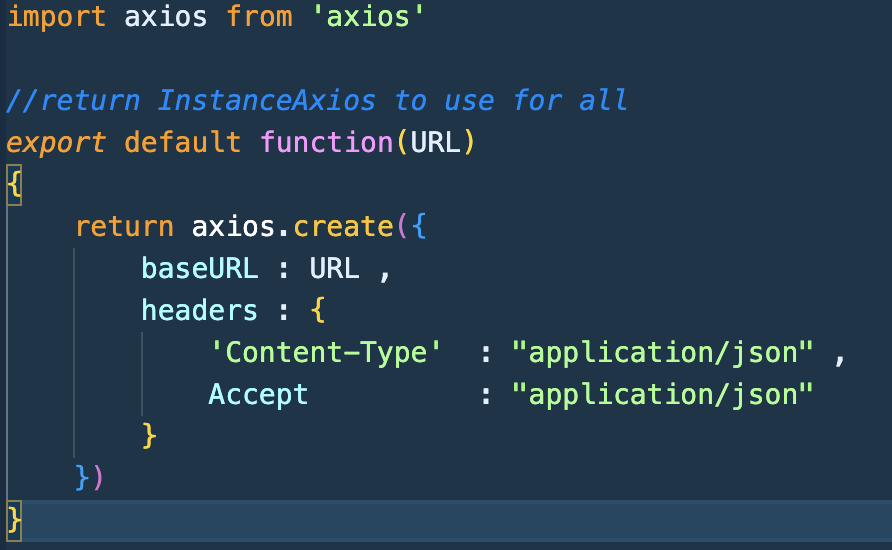
## 2 . Cài Đặt Server gọi dữ liệu về cho clinet bằng Axios

Đầu tiên cài đặt bằng :



Sau đó tạo thư mục services và trong đó tạo file : api.axios.js

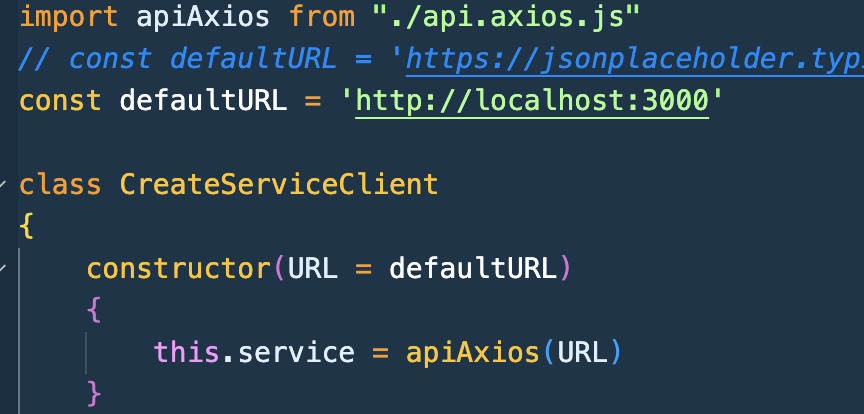
File này sẽ tạo ra một thực thể axios dựa theo URL được cấp vào

Cài đặt như sau : 

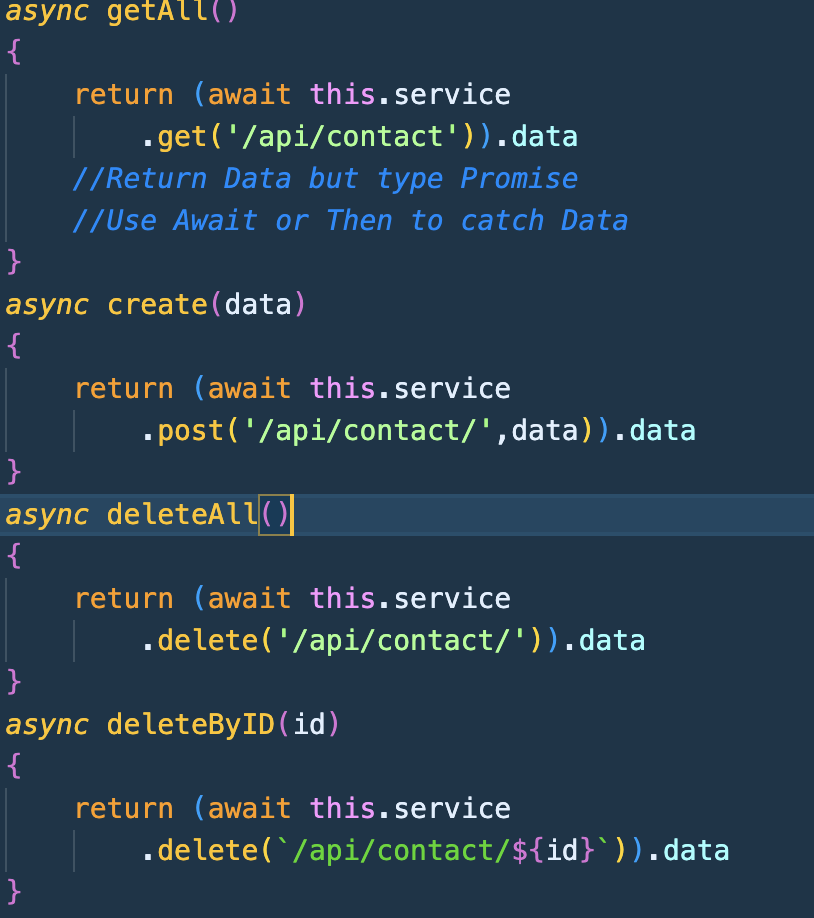
Hiệu chỉnh Header của instance luôn nhận và trả về kiểu Json

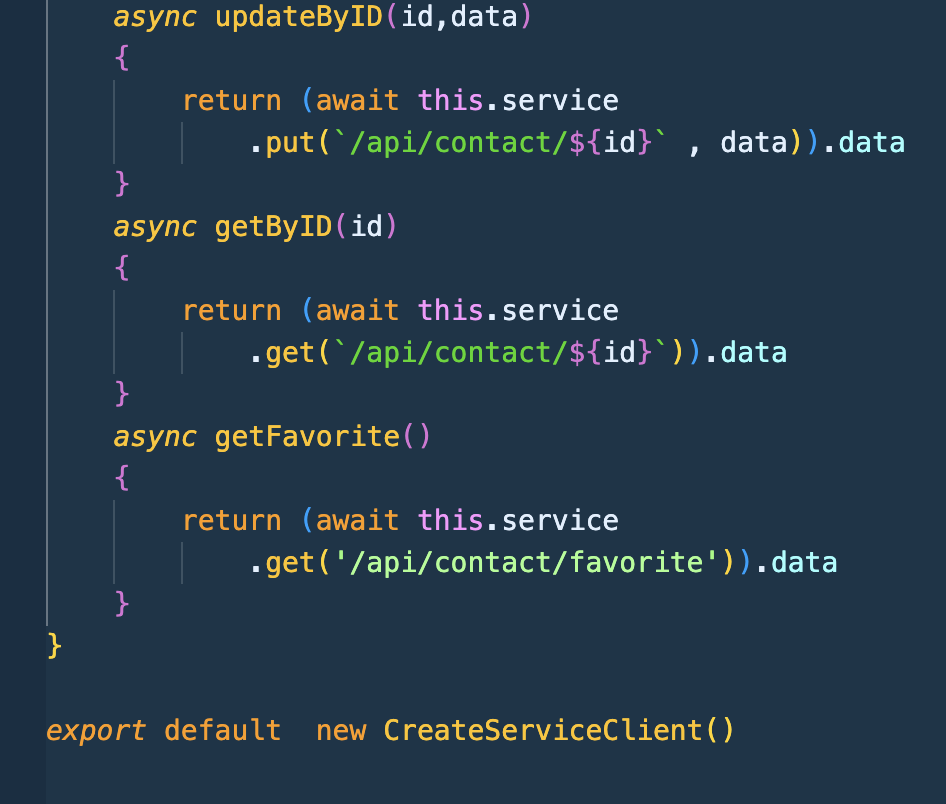
Tiếp đến là tạo thêm một file client.service.js trong service để định nghĩa các service của Phía Client như sau :

Tạo một class để chứa các method của service và có hàm xây dựng (constructor) là khởi tạo một service ứng với một axios function



Tiếp đến là cài đặt các service ứng với từng route được thiết kế ở phía Server :



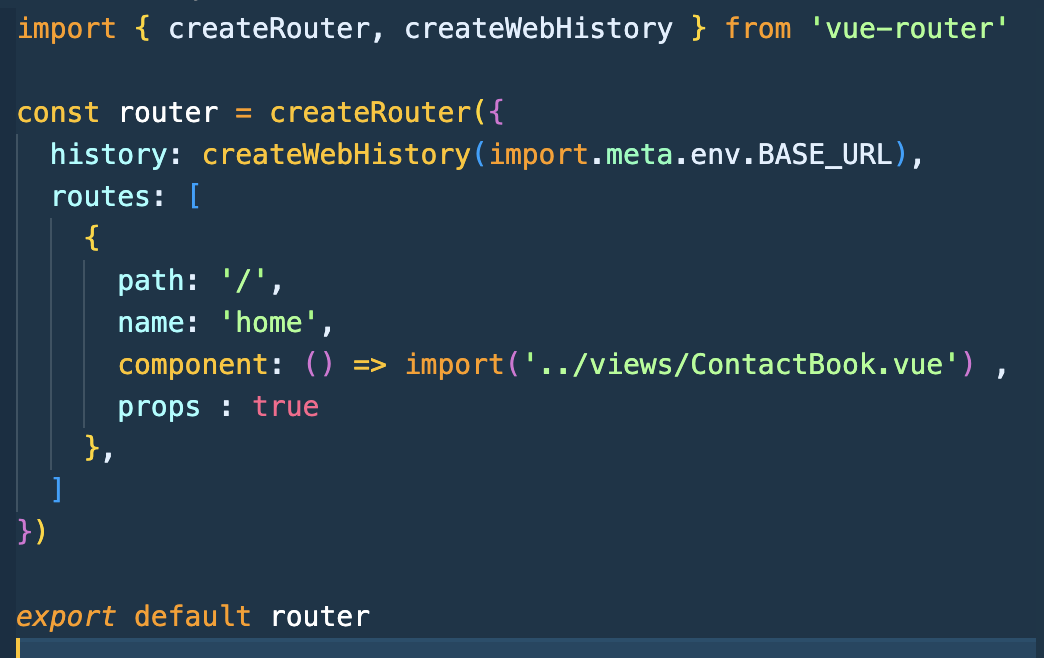


Cuối cùng xuất ra một đối tượng ứng với class vừa tạo đối với module client.service.js

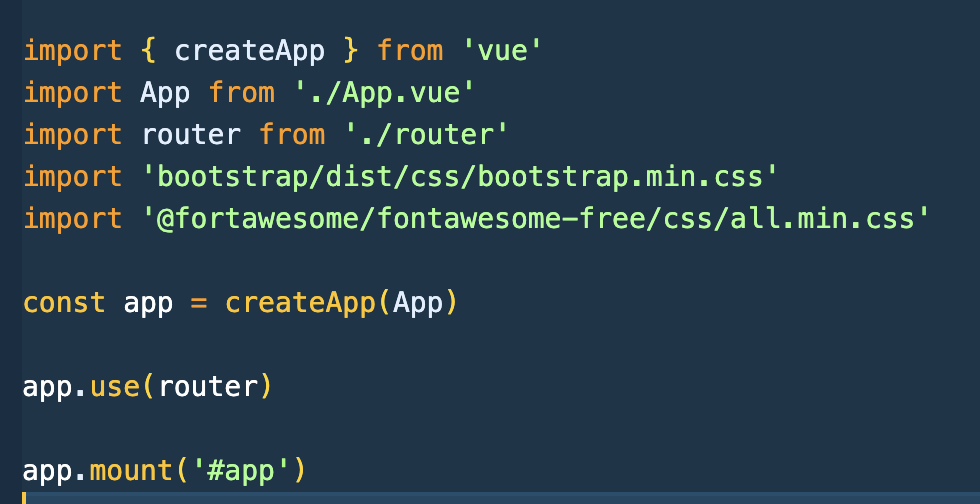
## 3. Cài đặt Trang để hiển thị danh sách

Đầu tiên cài và thêm vào các thư viện cần thiết như Bootstrap , Vue-Router , ..

Thêm và định nghĩa các path mặc định ứng với Component ContactBook như sau :



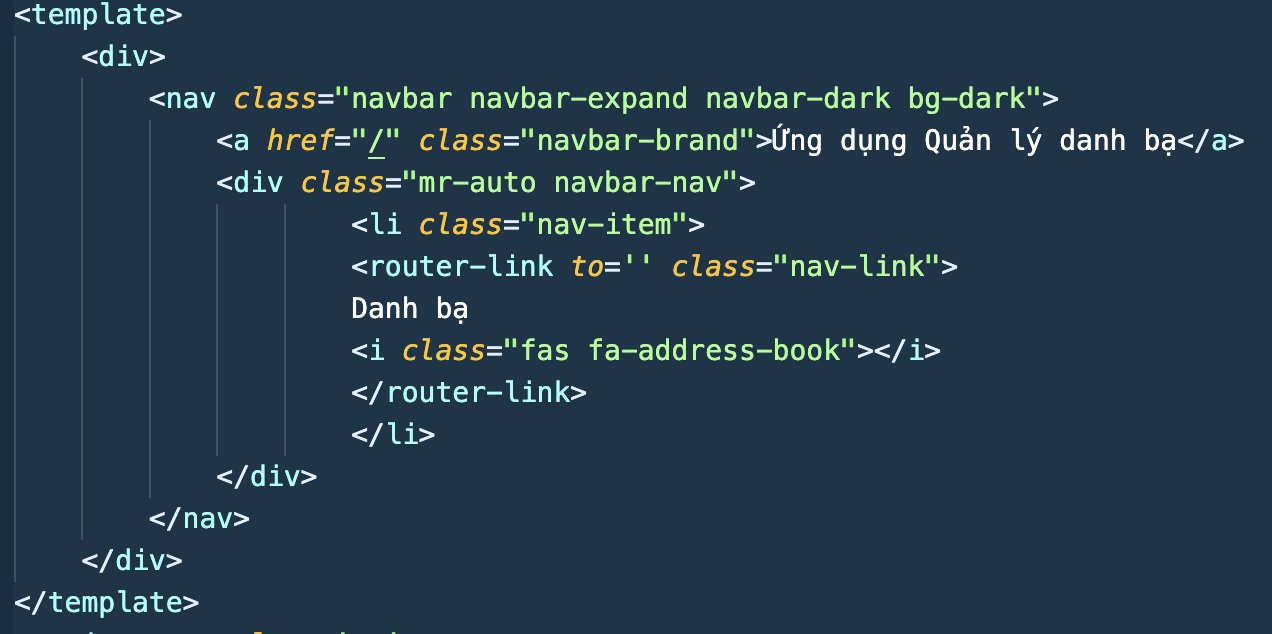
Tiếp đến là import các thư viện và áp dụng router vừa định nghĩa ở trên vào ứng dụng như sau :



Thiết Kế phần NavBar để sử dụng chung cho tất cả trang web như sau



Thẻ nav sẽ được thiết kế như sau :



Gồm tên ứng dụng và phần danh bạ khi ấn vào sẽ đưa trở về trang ban đầu

Ngoài Ra Nội dung trang chủ sẽ bao gồm như sau :

* Components Search
* Conponents ContactList
* Conponets ContactCard

### 3.1 Components Search

Một Componets Search sẽ bao gồm:

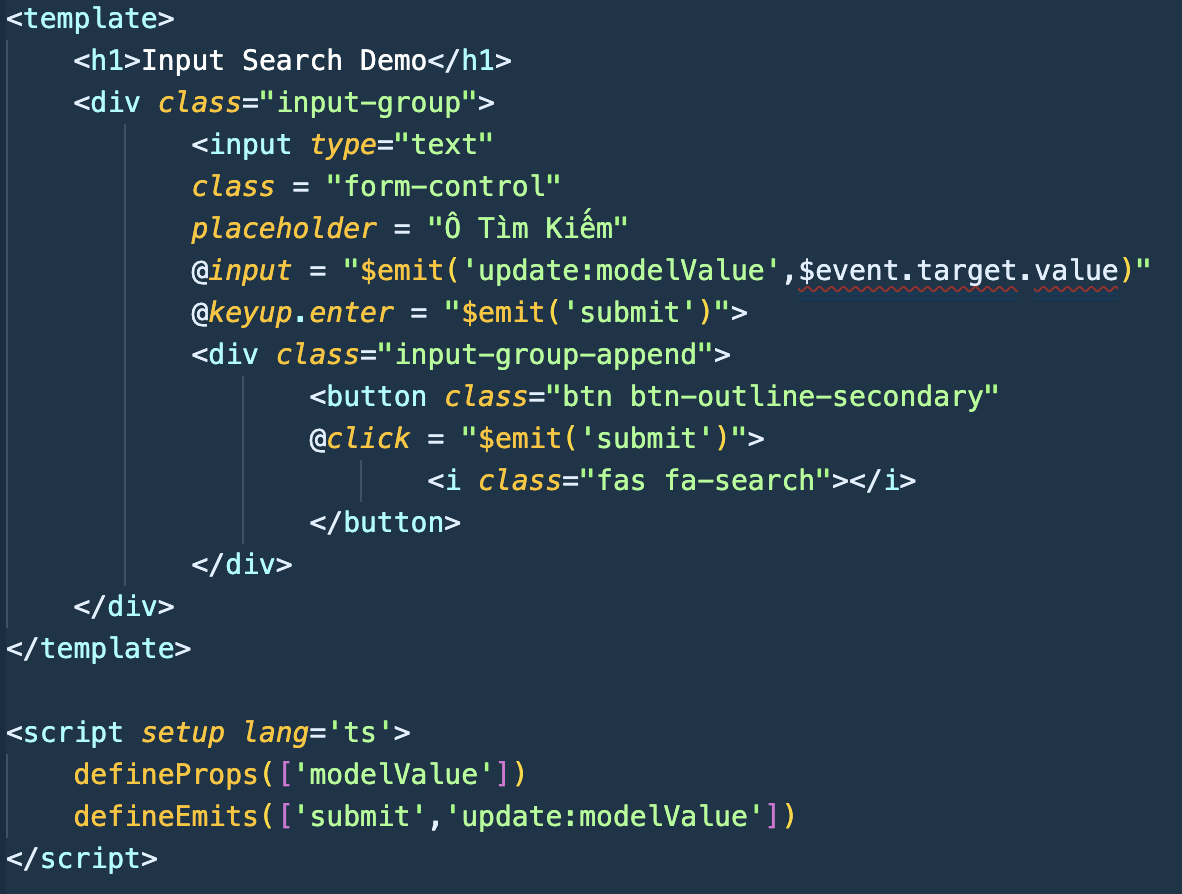


Sẽ bao gồm một input:text và button có icon là search

Khi nhập dữ liệu vào ô tìm kiếm thì sẽ trả dữ liệu đang nhập trực tiếp về cho trang Home

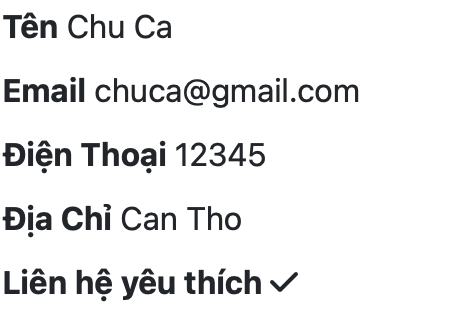
Khi ấn enter khi đang nhập dữ liệu hoặc nhấn vào button search thì sẽ gửi lên cho home thực hiện event submit

Ta thiết kế như sau :



### 3.2 Componets ContactCard

Một contactCard sẽ nhận một selectedContact ứng với một khách hàng đang được lựa chọn và sẽ hiển thị các thông tin của khách hàng đó như sau :



Ta sẽ triển khai Code như sau :



### 3.3 Componets ContactList

ContactList sẽ nhận vào 2 biến là

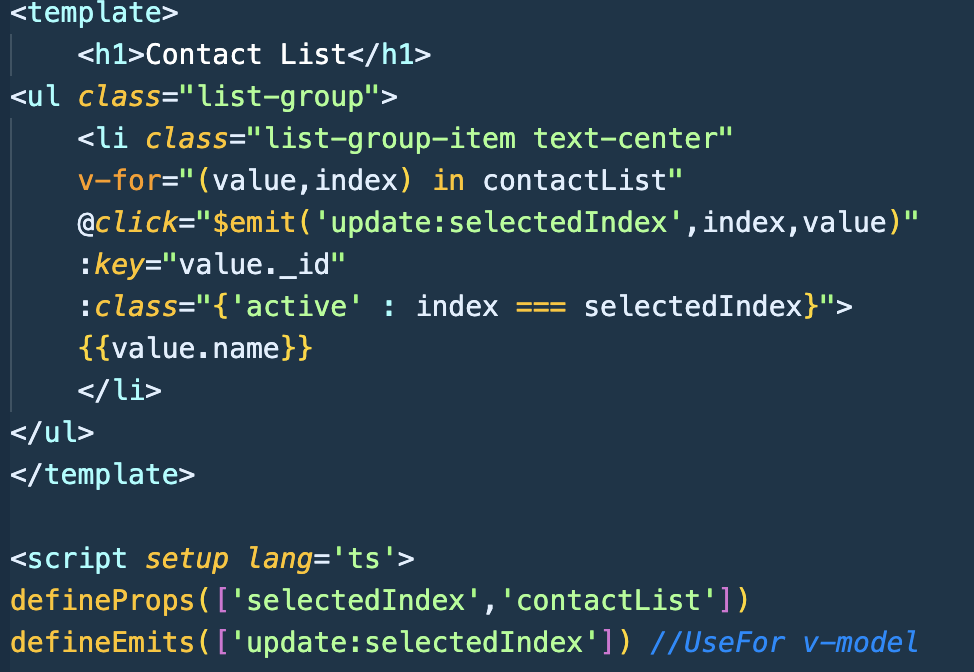
* **selectedIndex** ứng với vị trí index của danh sách đang được chọn và mặc định là -1
* **contactList** ứng với danh sách tất cả khách hàng

Ở đây dữ liệu contactList có 4 khách hàng và được hiển thị như sau :

Ở Trạng Thái chưa chọn ai Khi Chọn vào một khách



Phần Code sẽ được thiết kế như sau :



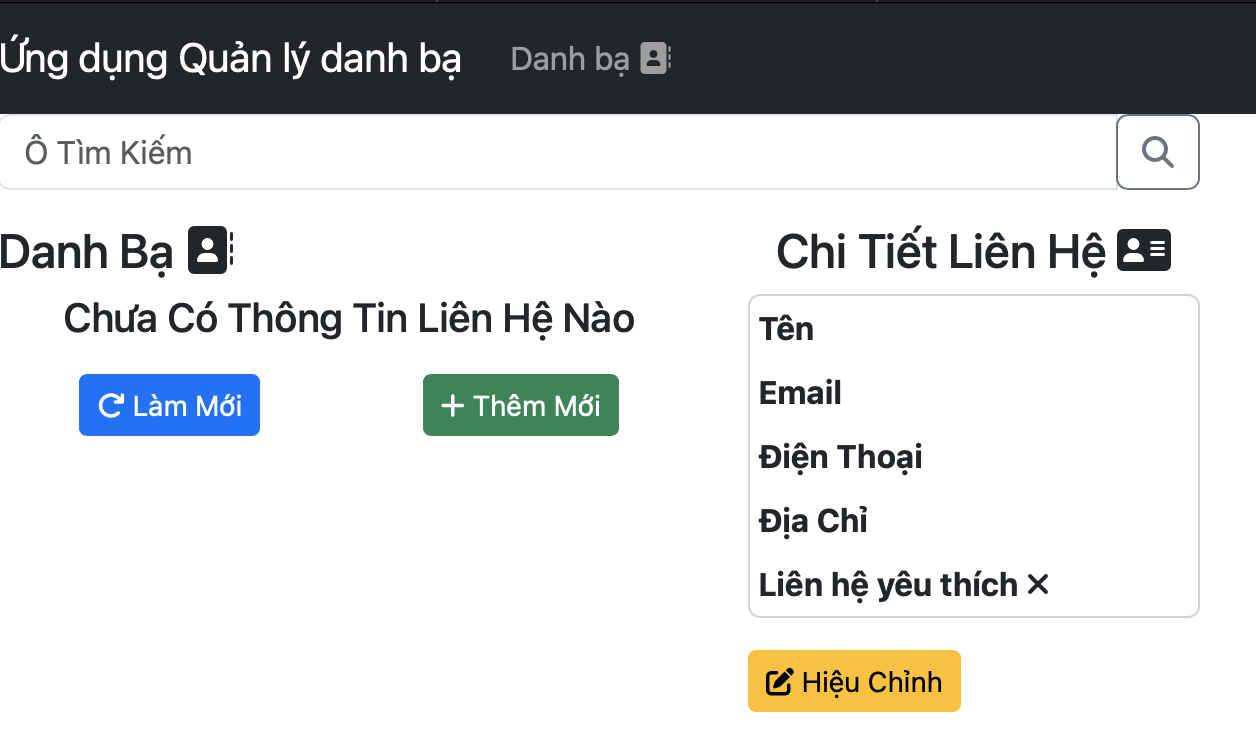
Chạy vòng lặp for để hiển thị ra hết mọi phần tử trong contactList

Khi click vào thì sẽ truyền lên 2 đối số đó là index và thông tin của user đang được chọn về cho Home

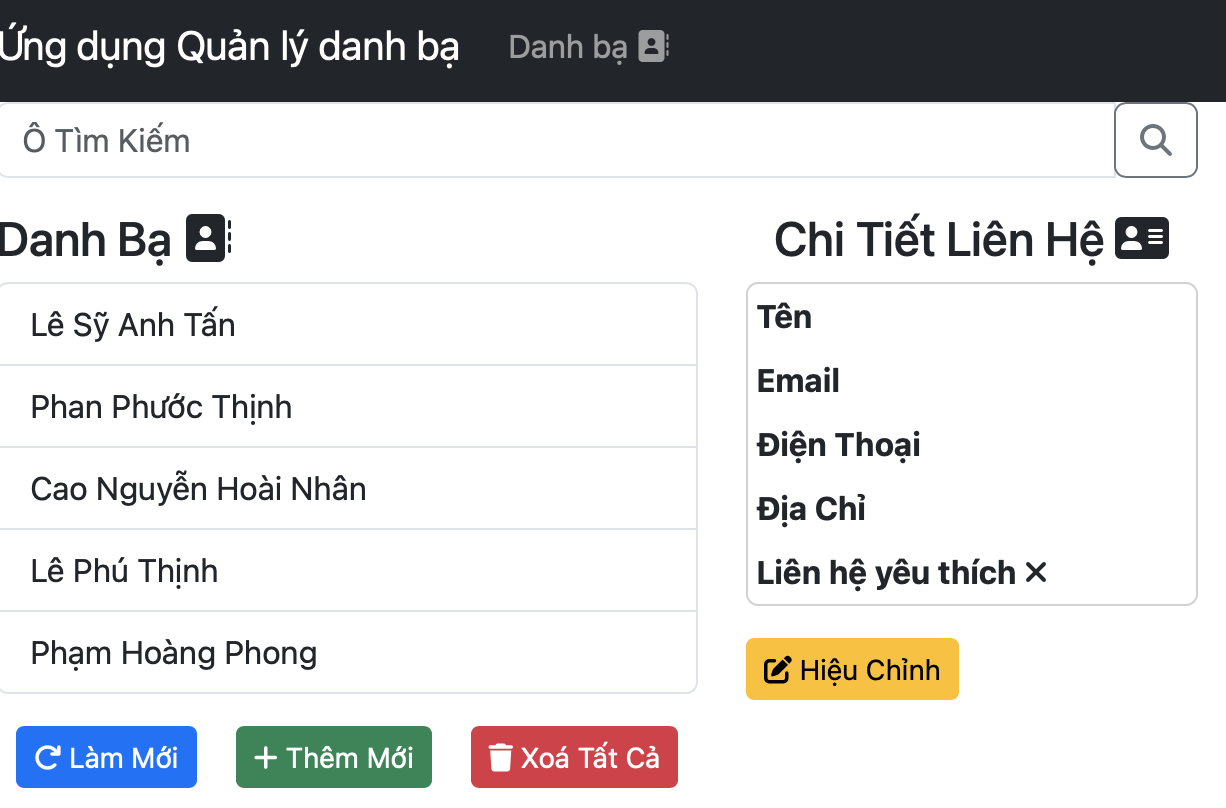
## 3.4 Cài Đặt Trang HomeView

Đây sẽ là phần hiển thị và kết hợp các components như InputSearch – ContactList – ContactCard

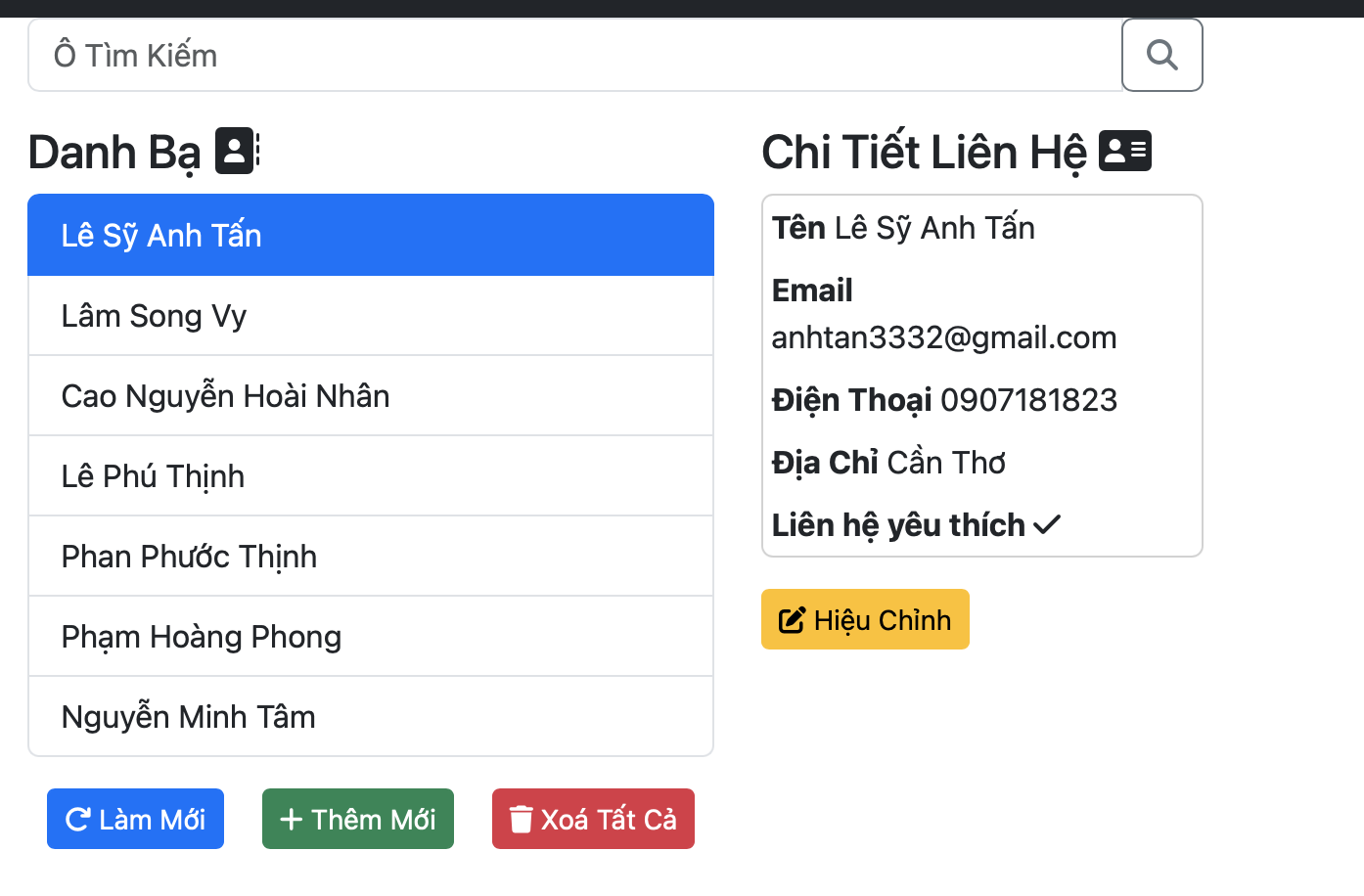
Đây là Trang chủ khi vừa load vào và chưa có dữ liệu :



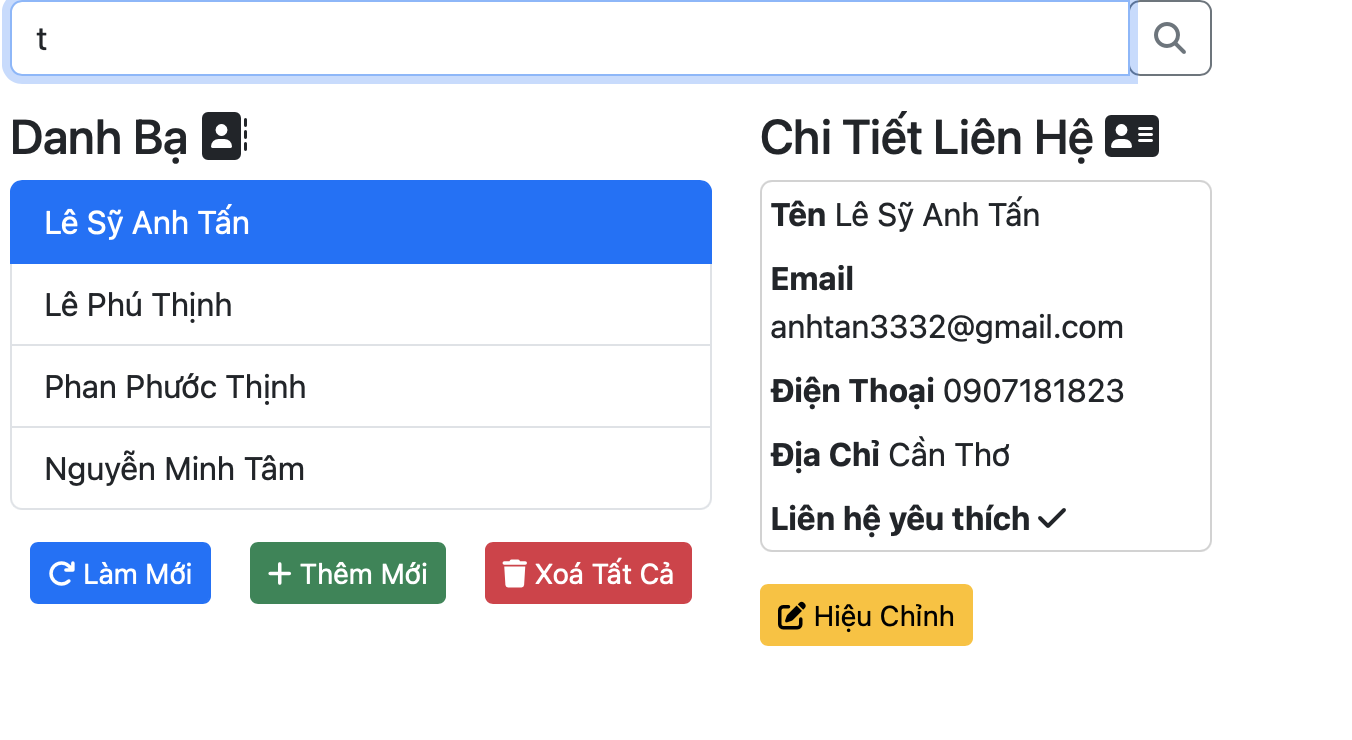
Và Khi dữ liệu người dùng được lấy lên :

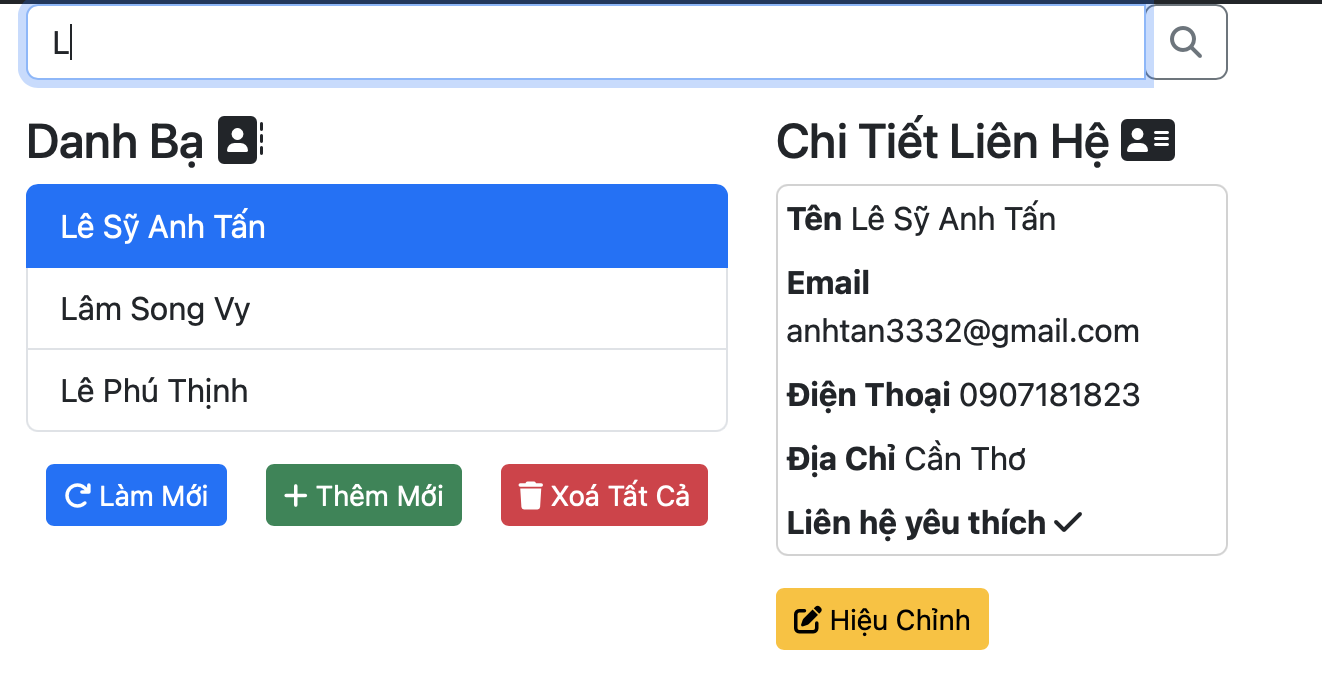


Và đây là thực hiện chọn vào một ngừoi dùng để hiển thị như sau :



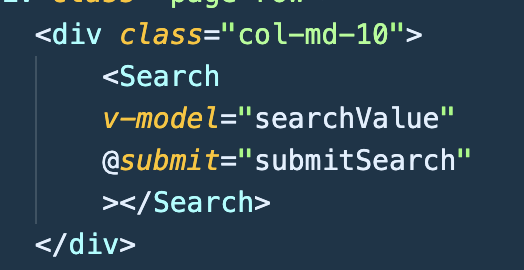
Đây là khi ta tiến hành Tìm kiếm :



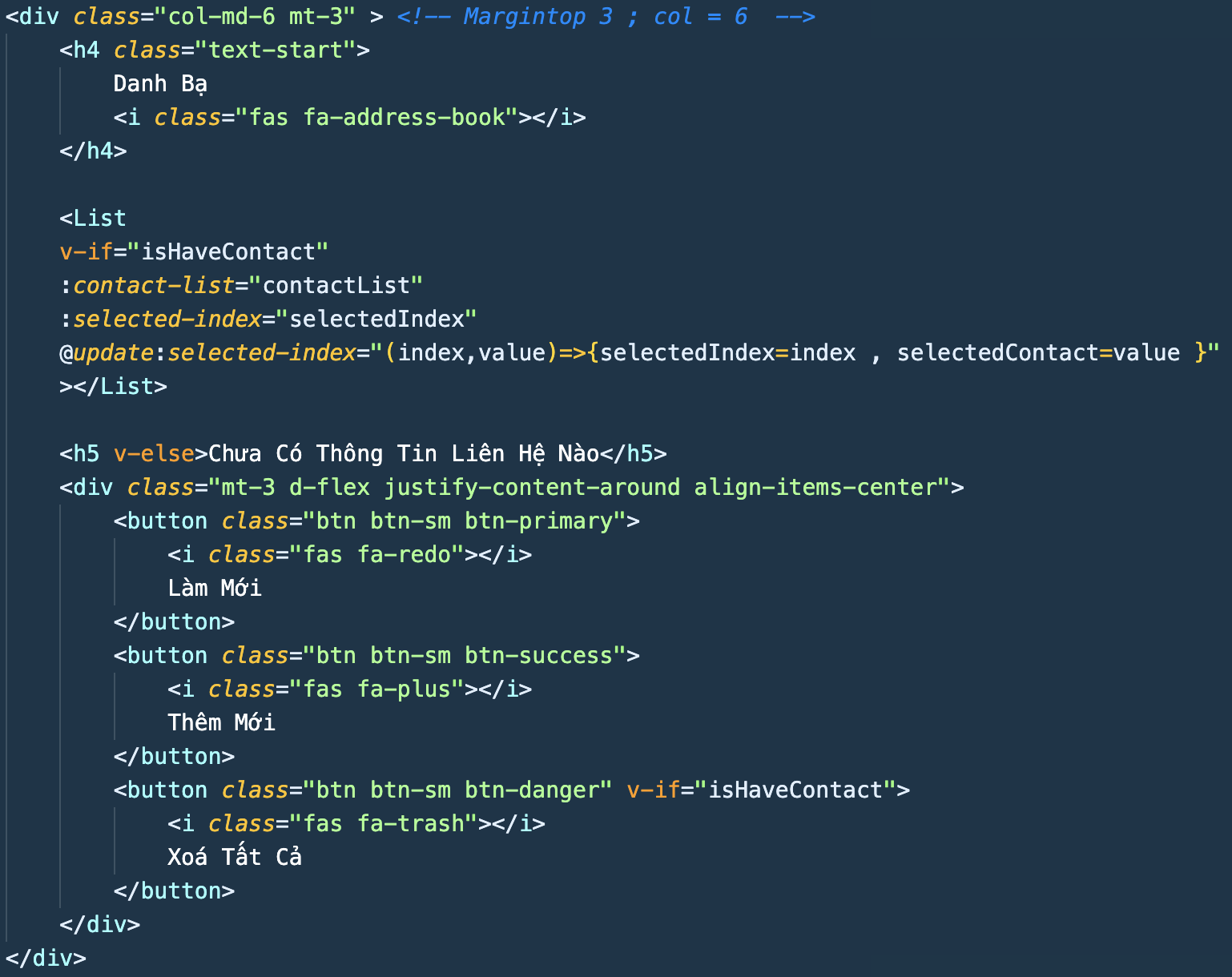


Tiếp tục tiến hành gom các components lại và để vào Trang Chủ (HomeView)

Thêm vào InputSearch như sau :



Cài đặt Phần ListContact như sau :



Cài đặt phần ContactCard :



Dưới đây là phần cài đặt ở thẻ Script :

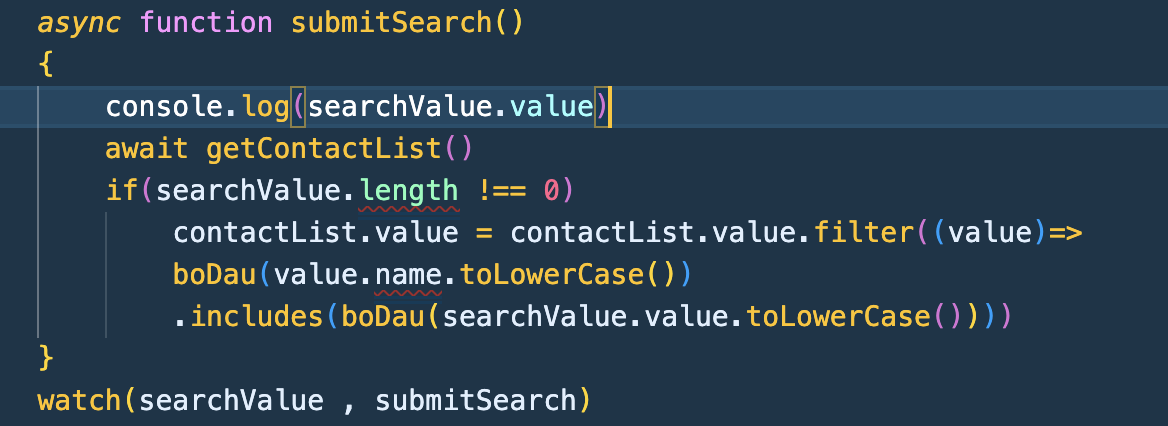
Thêm Các components , module và các biến cần thiết vào :



Cài đặt các hàm để lấy dữ liệu lên thông qua ServiceClient và có thể xoá tất cả



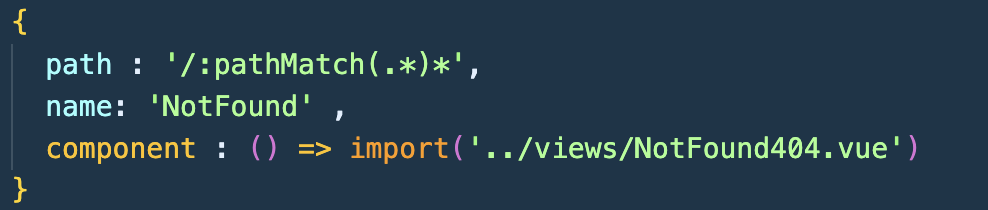
Đây là hàm để lọc danh sách ở Ô Tìm Kiếm :



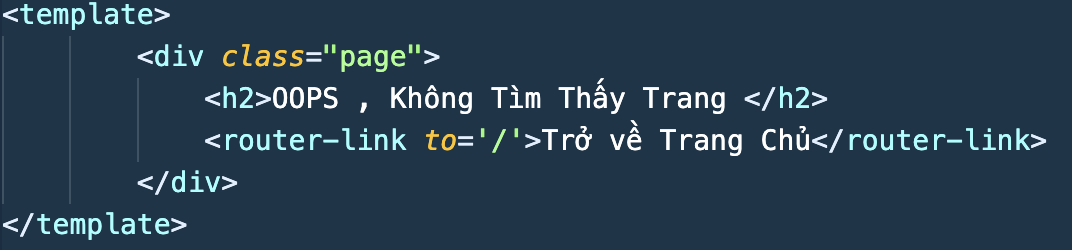
sử dụng watch để ô liên tục cập nhật để tìm kiếm

## 4. Cài đặt trang 404 Not Found

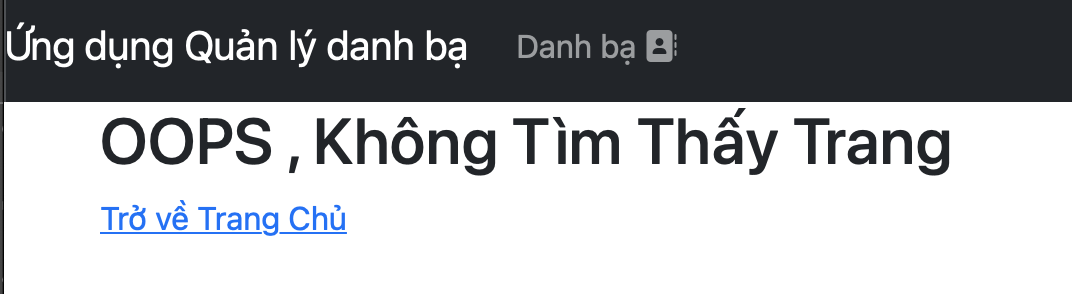
Đầu tiên định nghĩa các route không được định nghĩa sẽ mở components như sau :



Tiếp đến cài đặt trang NotFound404.vue như sau :

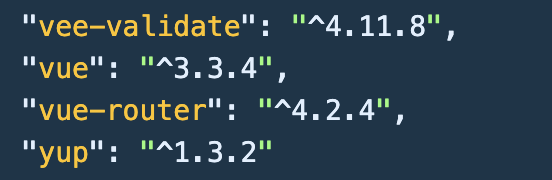


Đây sẽ là trang khi không tìm thấy :

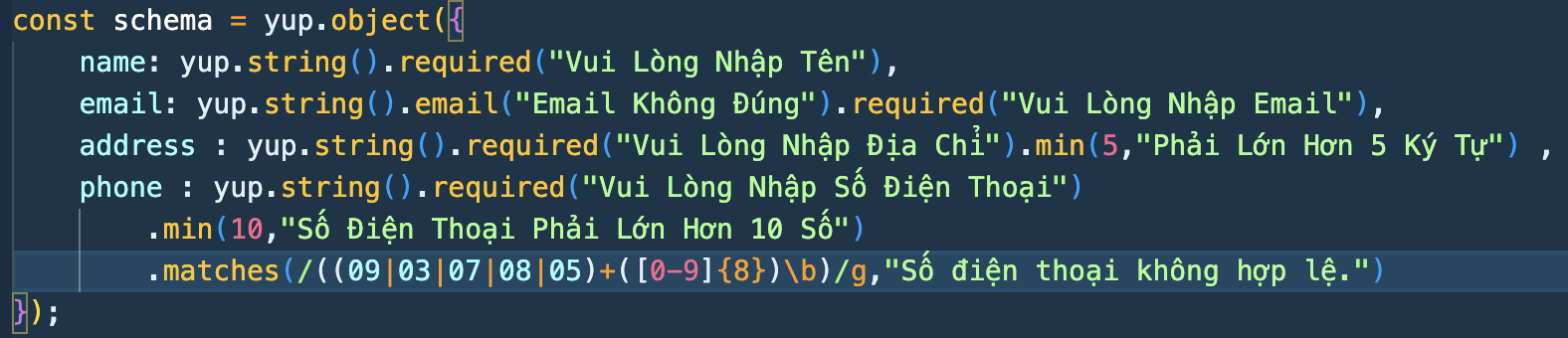


## 5 . Cài đặt Components Form

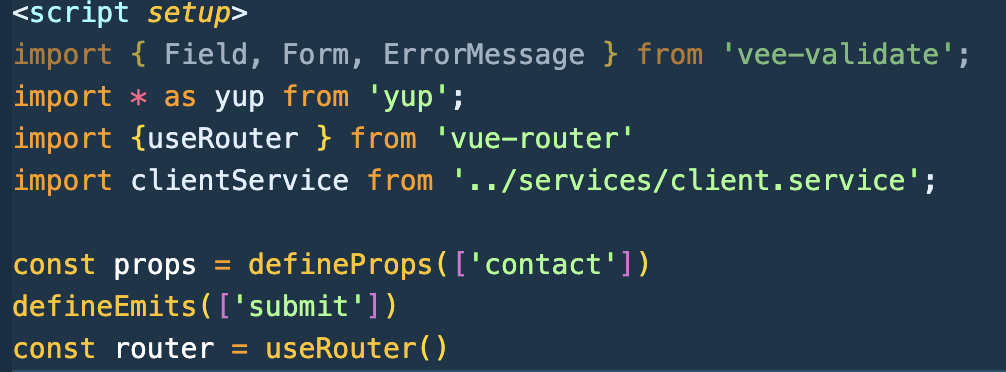
Cài đặt vee-validation và yup để có thể tạo form dễ dàng hơn :



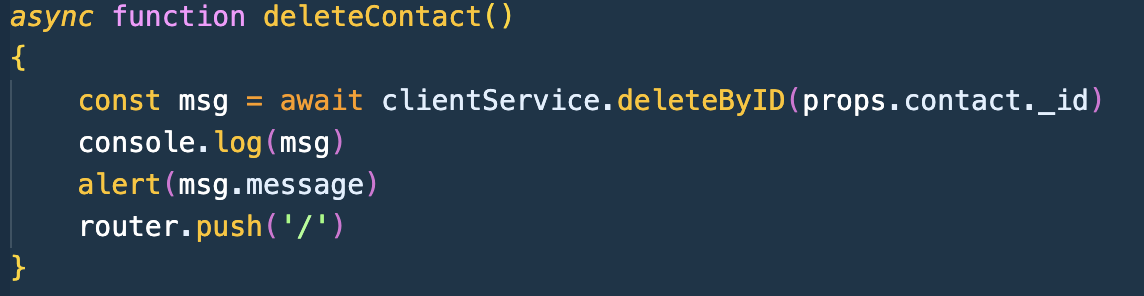
Tiếp đến ta tiến hành định dạng kiểu yup object theo ý muốn như sau :



Tiến hành cài đặt và thêm vào các hàm , các components cần thiết sau đây :



Và cài đặt phần hàm để xoá một contact như sau :



Và đây là phần code ta thêm vào templet để sử dụng như sau :

<template>

<div *class*="card">

<div *class*="card-header text-center">

<h3 >

ở đây sẽ gửi lên emits để componts cha xử lý

<slot></slot>

</h3>

</div>

<div *class*="card-body">

<Form :*validation-schema*="schema" @*submit*="$emit('submit',$props.contact)">

<div *class*="form-group text-center row mb-3">

<label *for*="name" *class*="col-2 text-start">Tên</label>

<Field *name*="name" *type*="text"

*v-model*="$props.contact.name"

*class*="from-control col-5"

/>

<ErrorMessage *name*="name" *class*="error-feedback col-auto" />

</div>

<div *class*="form-group text-center row mb-3">

<label *for*="email" *class*='col-2 text-start'>Email</label>

Các dữ liệu nhận vào từ props contact sẽ được thay đổi qua các input

<Field *name*="email" *type*="email"

*v-model*="$props.contact.email"

*class*="from-control col-5"

/>

<ErrorMessage *name*="email" *class*="error-feedback col-auto" />

</div>

<div *class*="form-group text-center row mb-3">

<label *for*="phone" *class*='col-2 text-start'>Số Điện Thoại </label>

<Field *name*="phone" *type*='tel'

*v-model*="$props.contact.phone"

*class*="from-control col-5"

/>

<ErrorMessage *name*="phone" *class*="error-feedback col-auto" />

</div>

<div *class*="form-group row mb-3">

<label *for*="address" *class*='text-start col-2'>Địa Chỉ</label>

<Field *name*="address"

*v-model*="$props.contact.address"

*class*="from-control col-5"

/>

<ErrorMessage *name*="address" *class*="error-feedback col-auto" />

</div>

<div *class*="form-group row mb-3">

<label *for*="favorite" *class*='text-start col-auto'>

<strong>

Liên Hệ Yêu Thích

</strong>

</label>

<input *name*="favorite"

*class*="from-control col-5"

*type*='checkbox'

*v-model*="$props.contact.favorite"

/>

</div>

Khi ấn Lưu thì @submit sẽ thực hiện gửi emit lên

<div *class*="form-group d-flex justify-content-around">

<button *class*="btn btn-outline-primary">Lưu</button>

<button

*v-show*="$props.contact.name !== undefined"

*type*="button"

*class*="ml-2 btn btn-outline-danger"

@*click*="deleteContact"

>

Xóa

</button>

<router-link *to*="/">

<button *class*="btn btn-outline-warning">Quay Về</button>

</router-link>

</div>

</Form>

</div>

## 6. Thiết Kế Trang Chỉnh Sửa và Thêm mới

Đâu tiên thêm đường path cho hai trang như sau :



Sau đó tiến hành cài đặt theo nguyên lý sau :

Thẻ **Form**  sẽ nhận vào props là : contact ứng với một object selectedContact

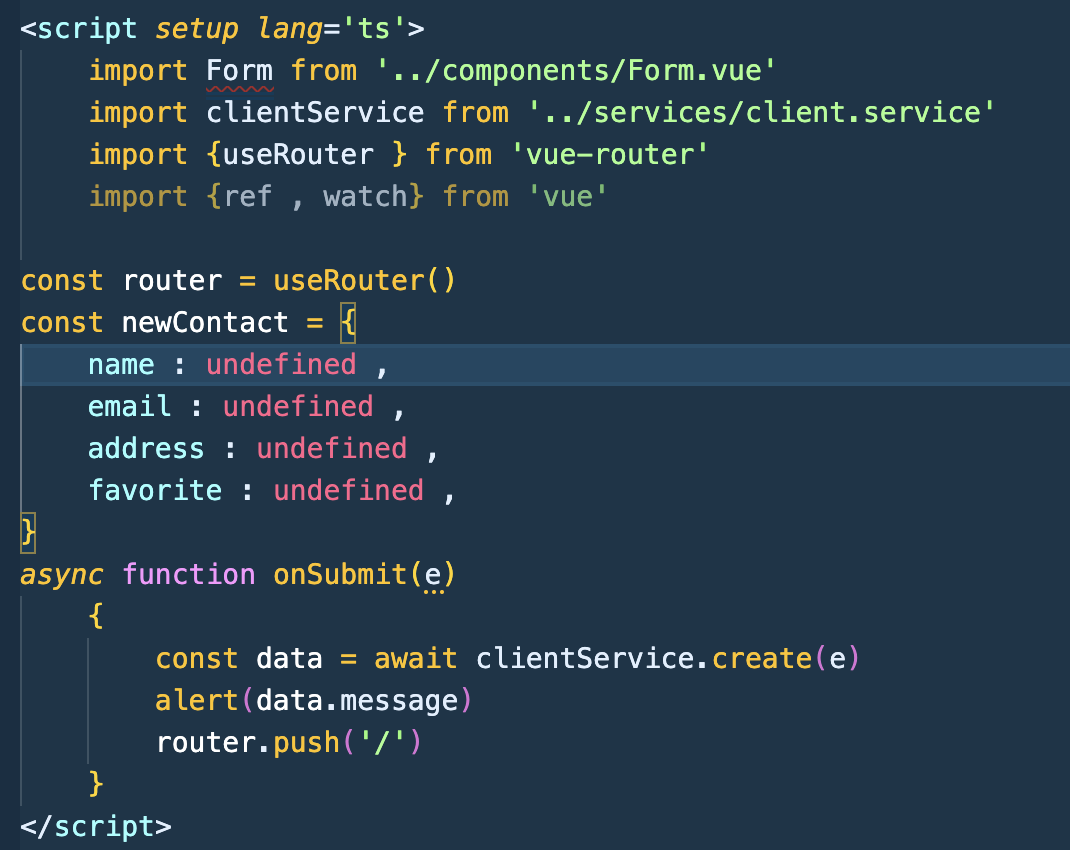
Nếu contact là rỗng thì chúng ta sẽ tiến hành sử dụng để thêm mới một liên hệ vào

Nếu contact là có giá trị thì ta sẽ tiến hành cập nhật giá trị mới

Ta tiến hành cài đặt như sau :

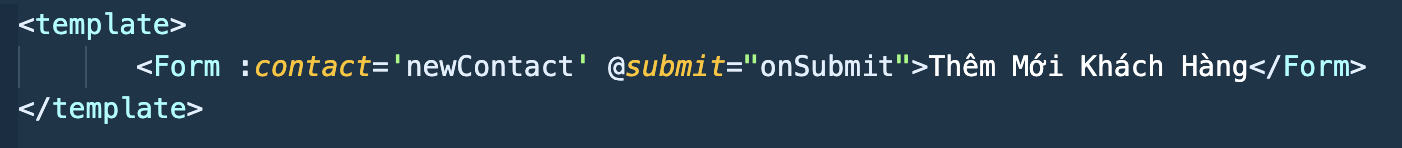
## 6.1 Trang Thêm Mới

Cài đặt và định nghĩa các hằng và hàm như sau :

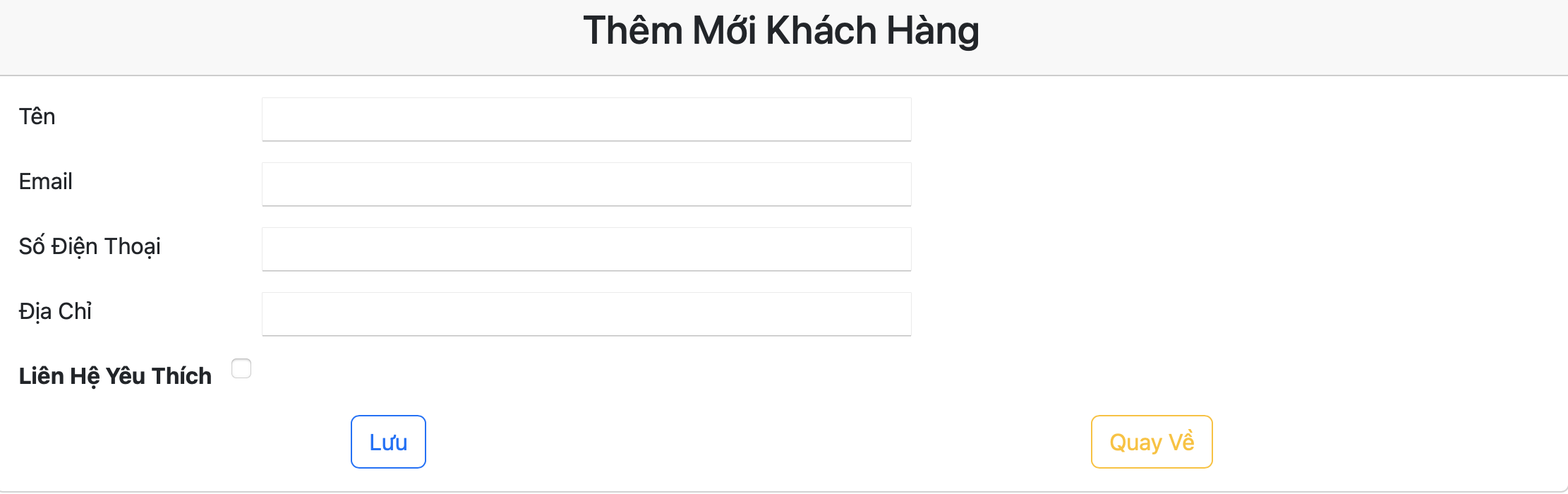


Khi ấn submit ở Form hàm onSubmit sẽ được gọi và tiến hành tạo mới một liên hệ

Tiến hành cài đặt Form để thêm mới như sau :

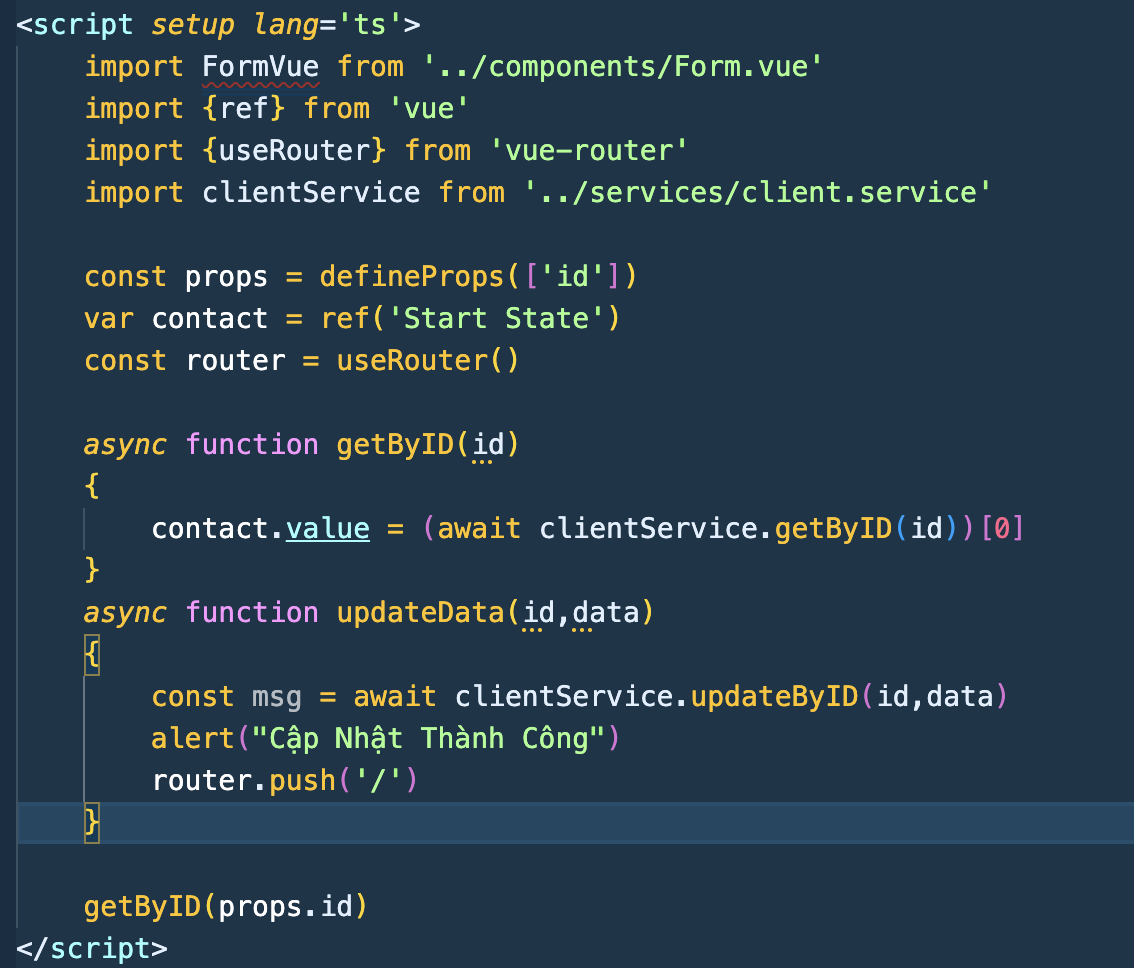


Đây là trang khi ta tiến hành thêm mới :



## 6.2 Trang Cập Nhật

Cài đặt các hàm và các hằng cần thiết như sau :



Ta sẽ có hàm để lấy dữ liệu dựa vào id nhận từ props của URL

Hàm updateData sẽ được gọi khi xác nhận là thay đổi thông tin

Tiến hành cài đặt thẻ Form như sau :



Sẽ nhận vào liên hệ dựa theo ID đang truy vấn tới và tiến hành thao tác chỉnh sửa sau đó cập nhật lại .

Đây Sẽ là trang khi ta tiến hành cập nhật – chỉnh sửa thông tin :



Link Github Chứa SourceCode :

https://github.com/anhtan2223/B2113342\_LeSyAnhTan-frontend34