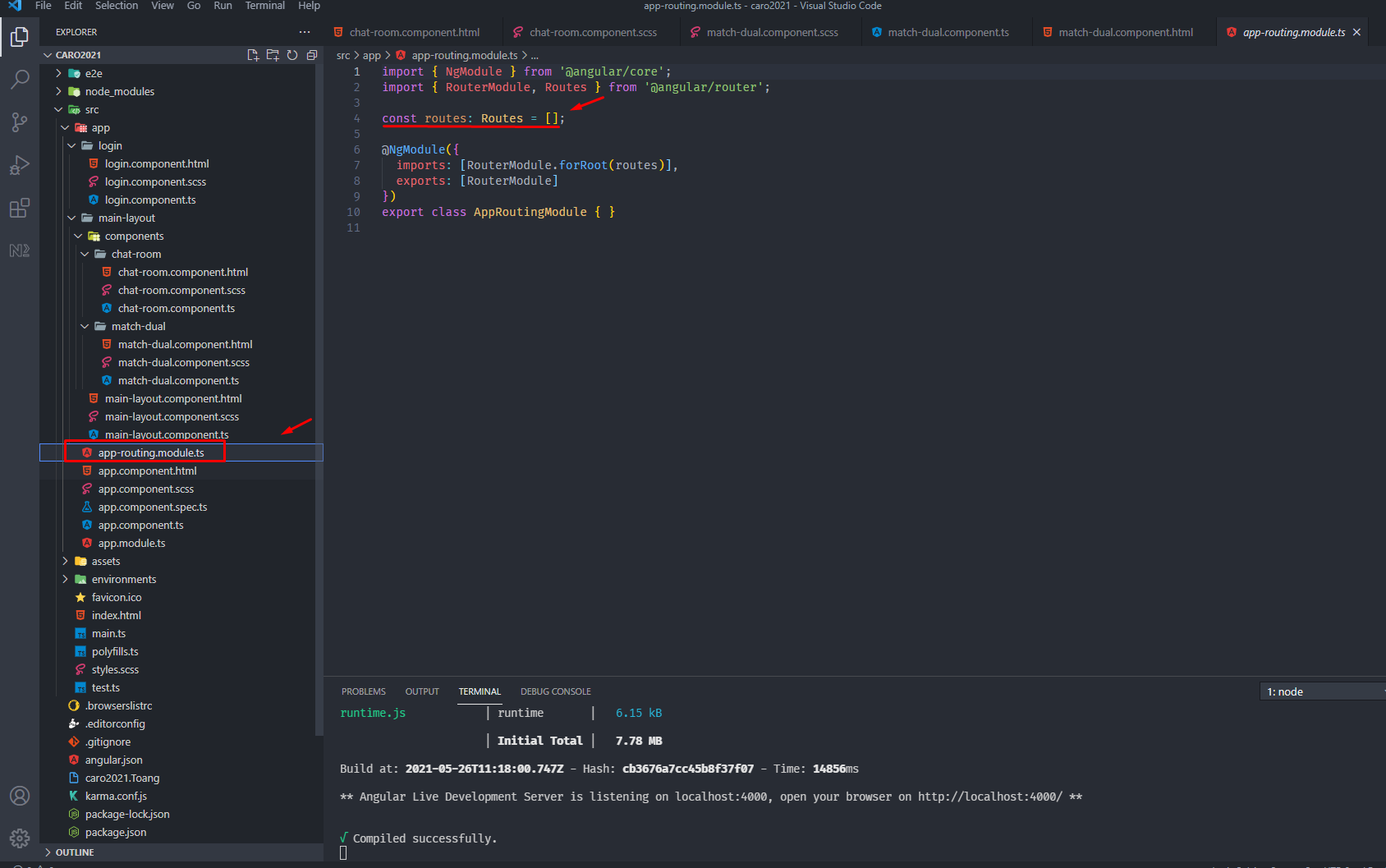
Thực hiện tích hợp api login và register vào front-end – caro online

Luồng dữ liệu chúng ta:

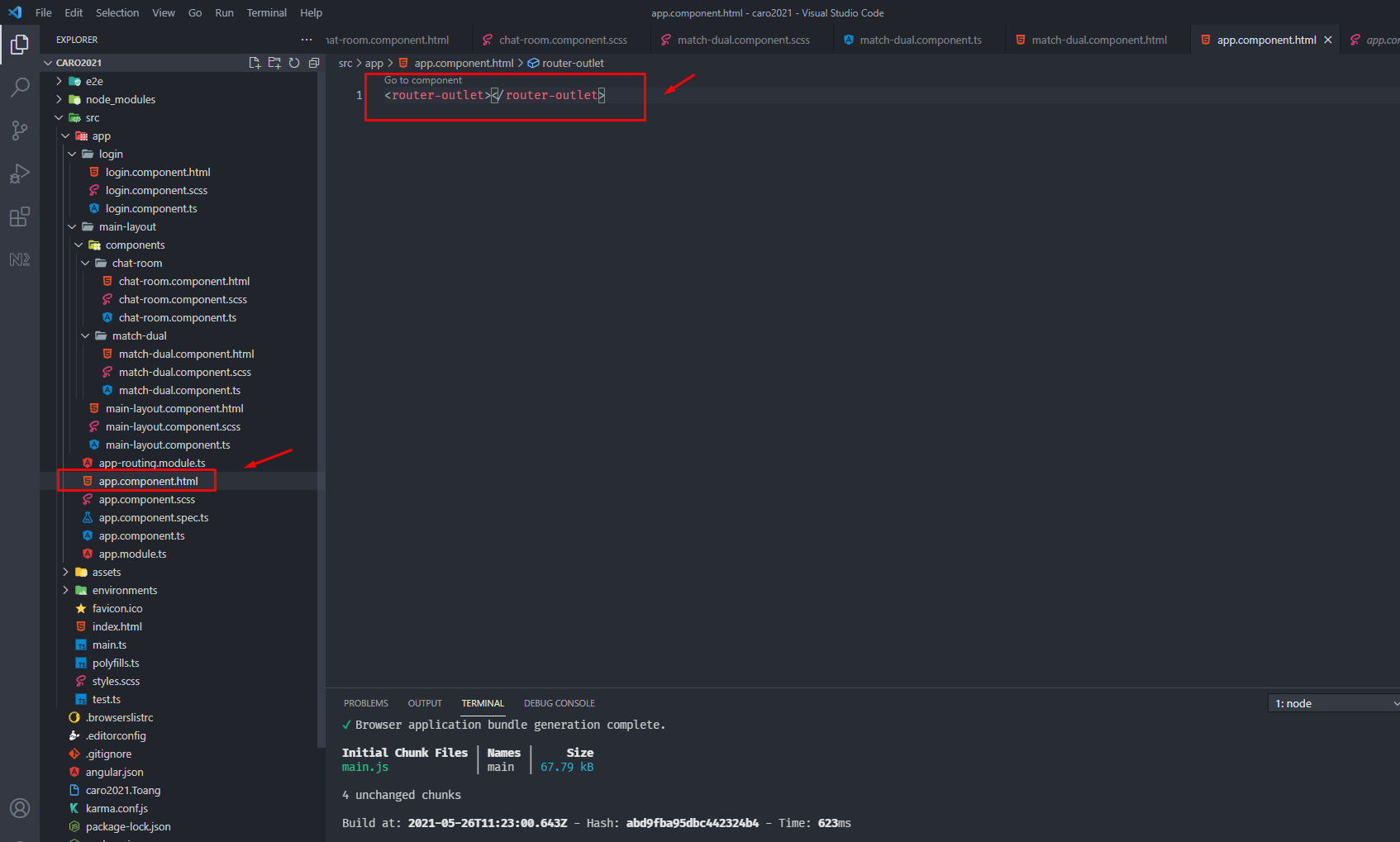
* Front-end: thực hiện có một màn hình đăng kí điền các thông tin vào trong form đăng kí đó và call 1 api có URL là register user bên back-end (local host) thì back-end sẽ trả về một chuỗi json thông tin của user mà chúng ta vừa đăng kí
* Login: ở front-end sẽ có 1 màn hình có 2 trường thông tin là username và password và khi mà chúng ta click vào nút login(đăng nhập) thì sẽ call 1 api là login bên back-end và được trả về một chuỗi (string) là :token của user đó.
* Nếu như có token trả về thì cho phép nhảy vào màn hình phòng chờ.

-Font-end sửa sao cho các có thể nhận các respond của back-end và thực hiện router sang một màn hình khác.

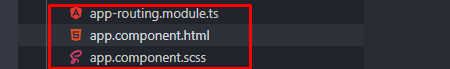


Routes: được hiểu là các tuyến đường mà hệ thống có. (có thể nhảy tới cái tuyến đường mà chứa màn hình là khác nhau)

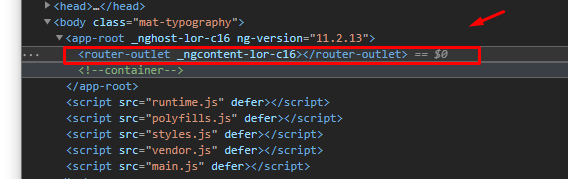
Đầu tiên sửa cái màn hình



Nhận thấy <app-root></app-root> nó chính component gốc được tạo thành từ 3 thành phần html, css, ts



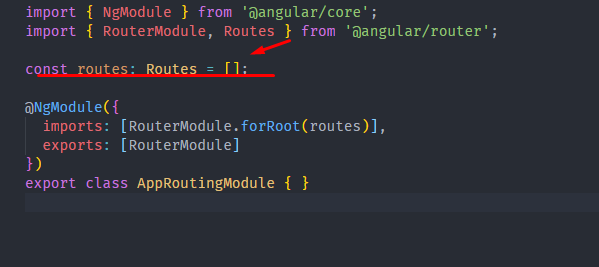
Thì đây là component cha ( thằng to nhất chứa những component khác)



Thế thì cái <router-outlet></router-outlet>

Đùng để làm gì ?

* Dùng để chuuyển đổi thành các component được khai báo trong app.routing trong biến

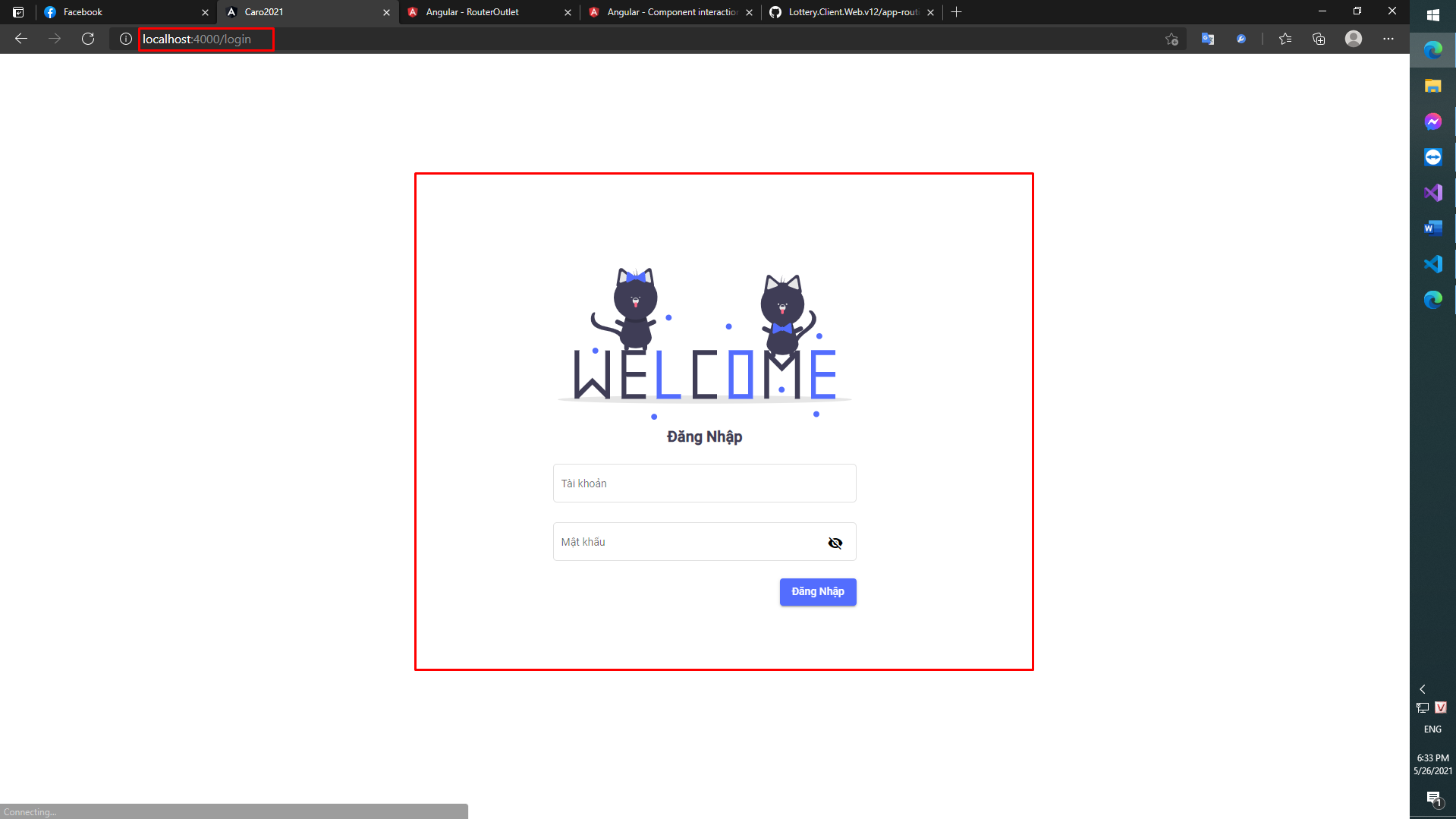




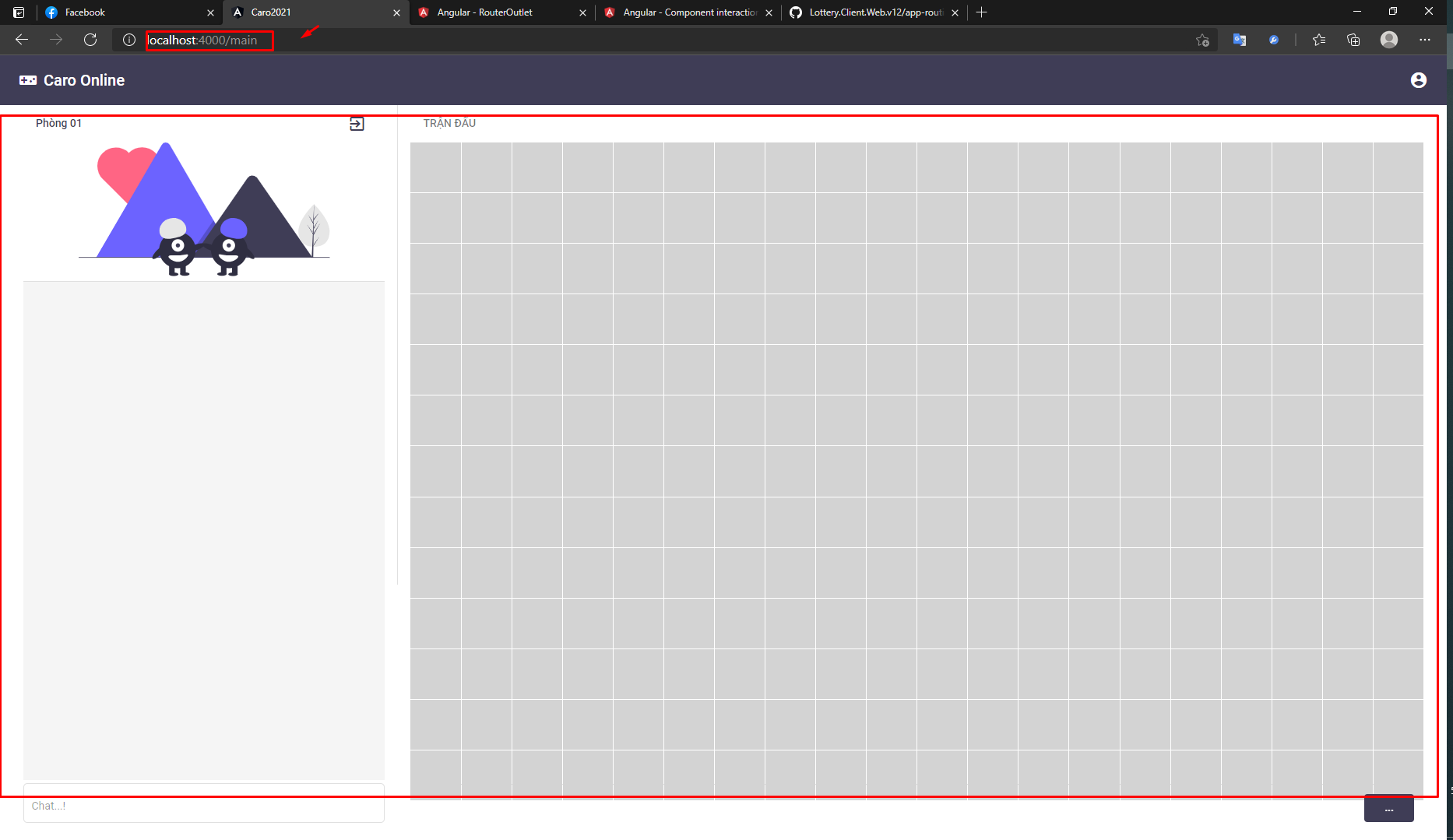
ở đây chúng ta hiểu là chúng ta định nghĩa 2 đường dẫn trỏ tới 2 components tương ứng

ví dụ:

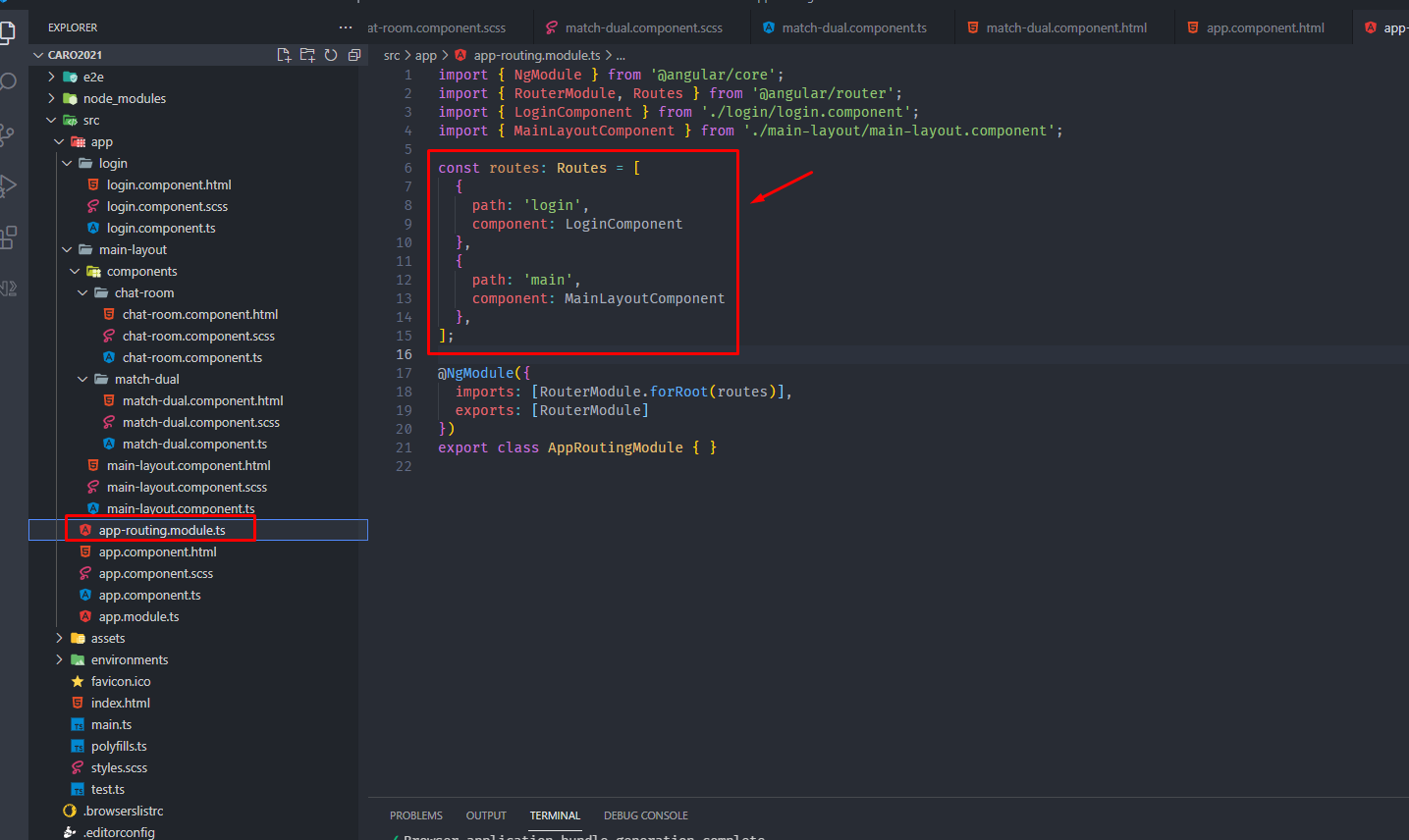
<http://localhost:4000/login> thì sẽ ra màn hình là LoginComponent



Và <http://localhost:4000/main> thì sẽ tương ứng vói MainLayoutComponent



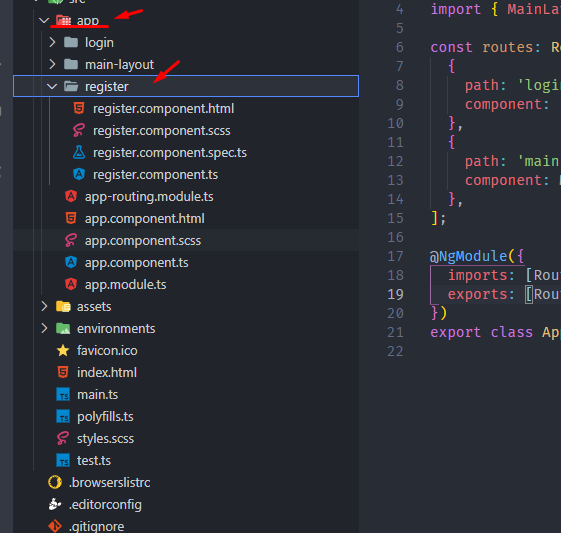
Tóm lại <router-outlet></router-outlet>

Thì đọc các đường dẫn từ 

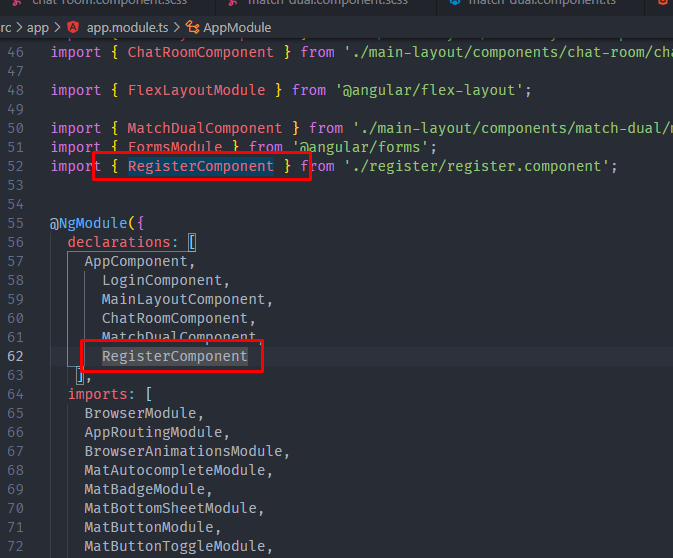
Thế bây giờ chúng ta cần một màn hình về đăng kí ( tại vì chúng ta chưa làm )

Bằng cách tạo 1 component đặt tên là register …

Sau đó là thêm 1 đường dần vào file app-routing.module.ts và trong mảng Routes. Với đường dẫn là tùy các bạn

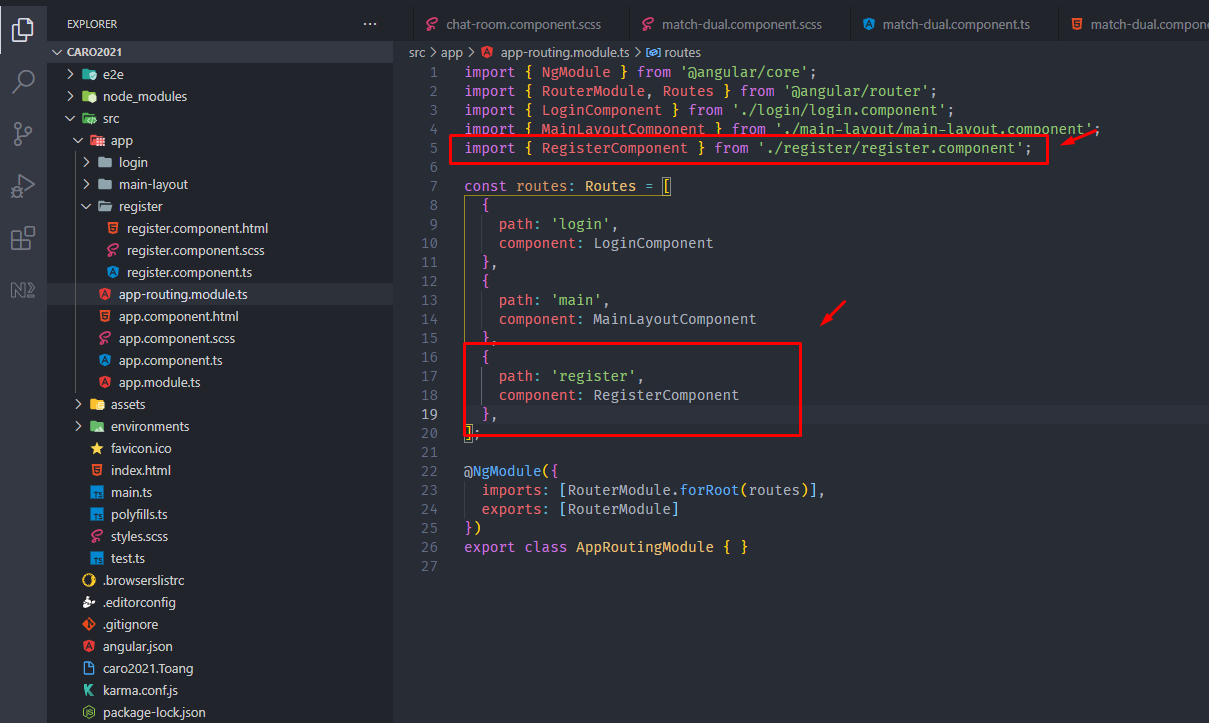


Sau đó thực hiện bước quen thuộc nhúng Component này vào app module để khai báo rằng là tôi sẽ được sử dụng nó trong con app của mình.

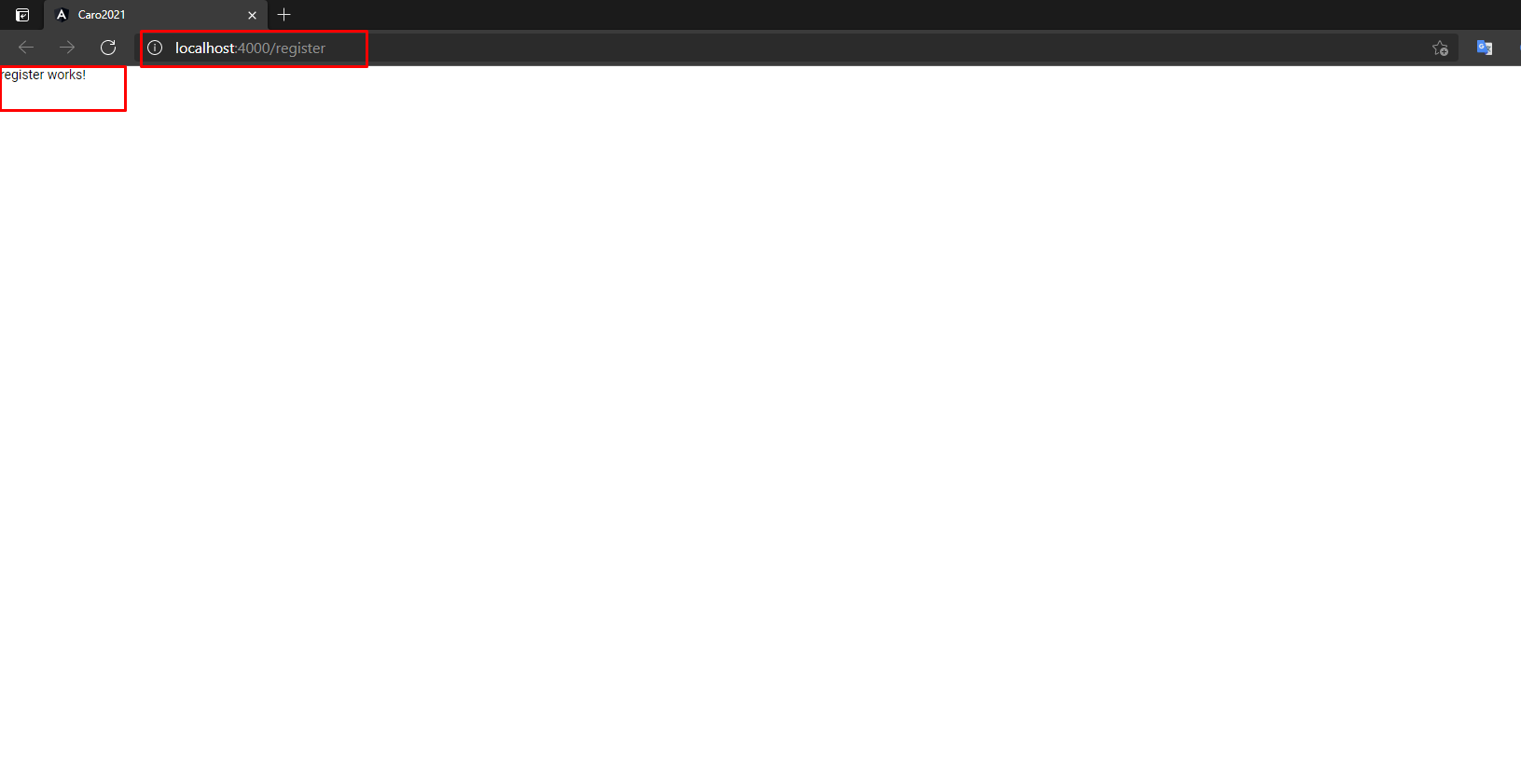


(bước này luôn phải nhớ)

Lập tức chúng ta đăng kí nó với đường dẫn là “register” trong app.routing.module.ts



Và thử nghiệm trên UI luôn xem có gọi đc k ?



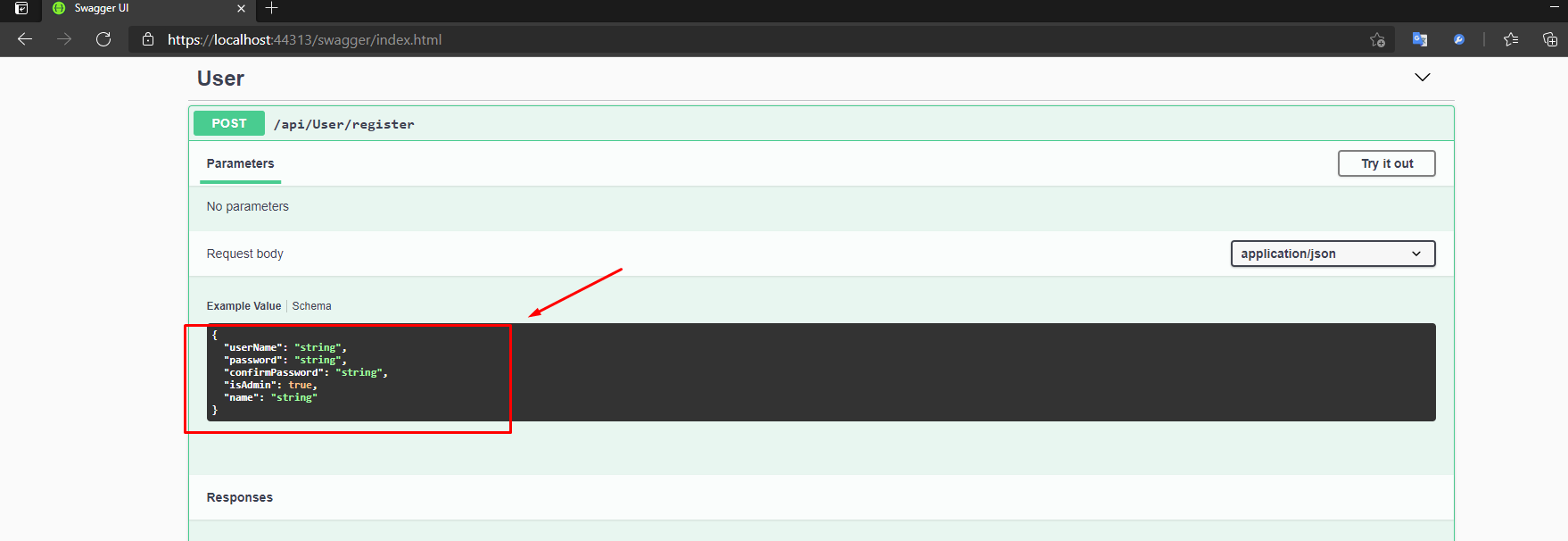
Như này đã chạy.

Dựa trên trang login để thực hiện sửa đổi html, css để tạo thành một giao diện mới dùng cho việc đăng kí tài khoản.

Chúng ta sẽ chép tương ứng các file của login với file của register

Html => chép html, css => chép csss tương ứng, ts cũng vậy….

Sau đó thực hiện sửa đổi text và các class css để sao cho nó giống 1 trang đăng kí.



Dựa vào api mà chúng ta định nghĩa ra các trường cần thiết để bắn lên server (local host)

Và tạo field tương ứng trên giao diện ở đây cần 5 component nhỏ.

Username, password, confirmpassword, name => thì sử dụng thẻ input

isAdmin => sử dụng checkbox

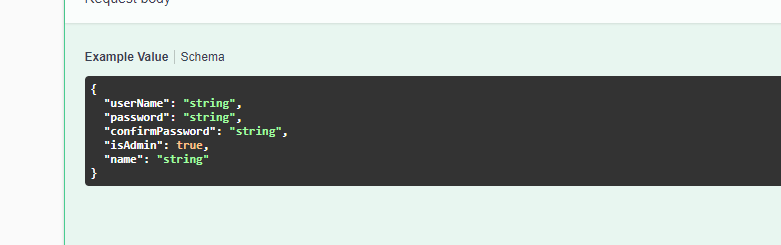
Chú ý: muốn sử dụng UI component của material angular thì phải import trong app.module (bài 1, 2) – nếu quên thì phải tự xem lại.

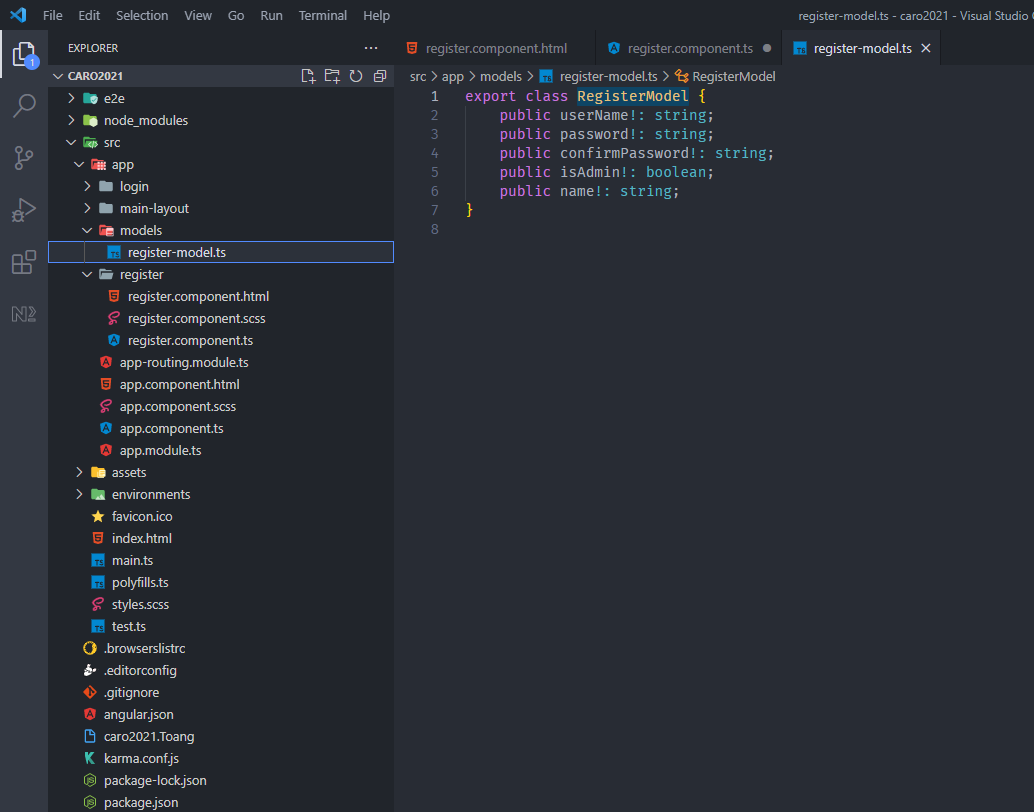
Đầu tiên chúng ta liên kết được các biến giữa ts và html mà chúng ta nhập từ thẻ input

ở sử dụng đơn giản chỉ dùng two way binding – tức là khi dữ liệu ở html thay đổi thì dữ liệu của file ts tương ứng cũng sẽ thay đổi và ngược lại

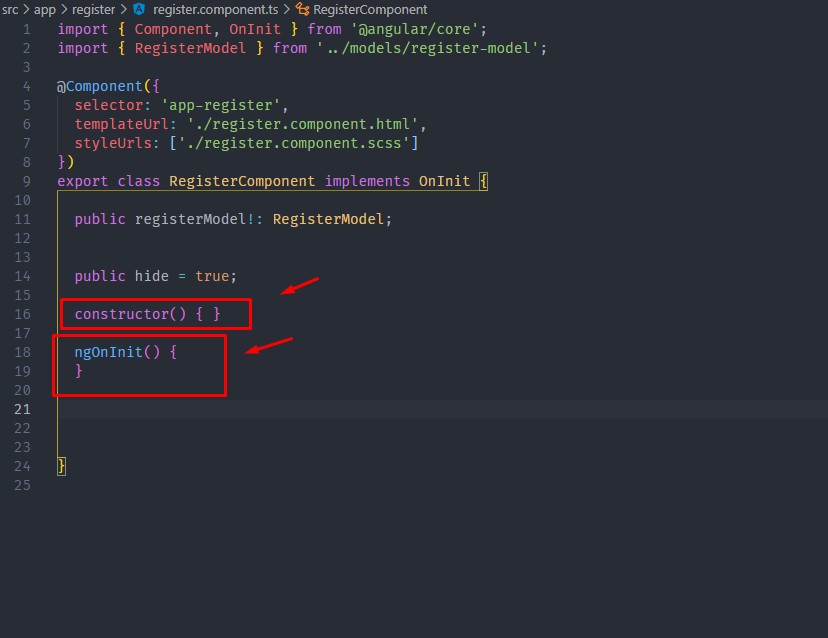
dùng ngModel để binding giữa biến ở .ts và .html với nhau.

Đầu tiên chúng ta định nghĩa cái class mà chứa các thông tin mà API cần.





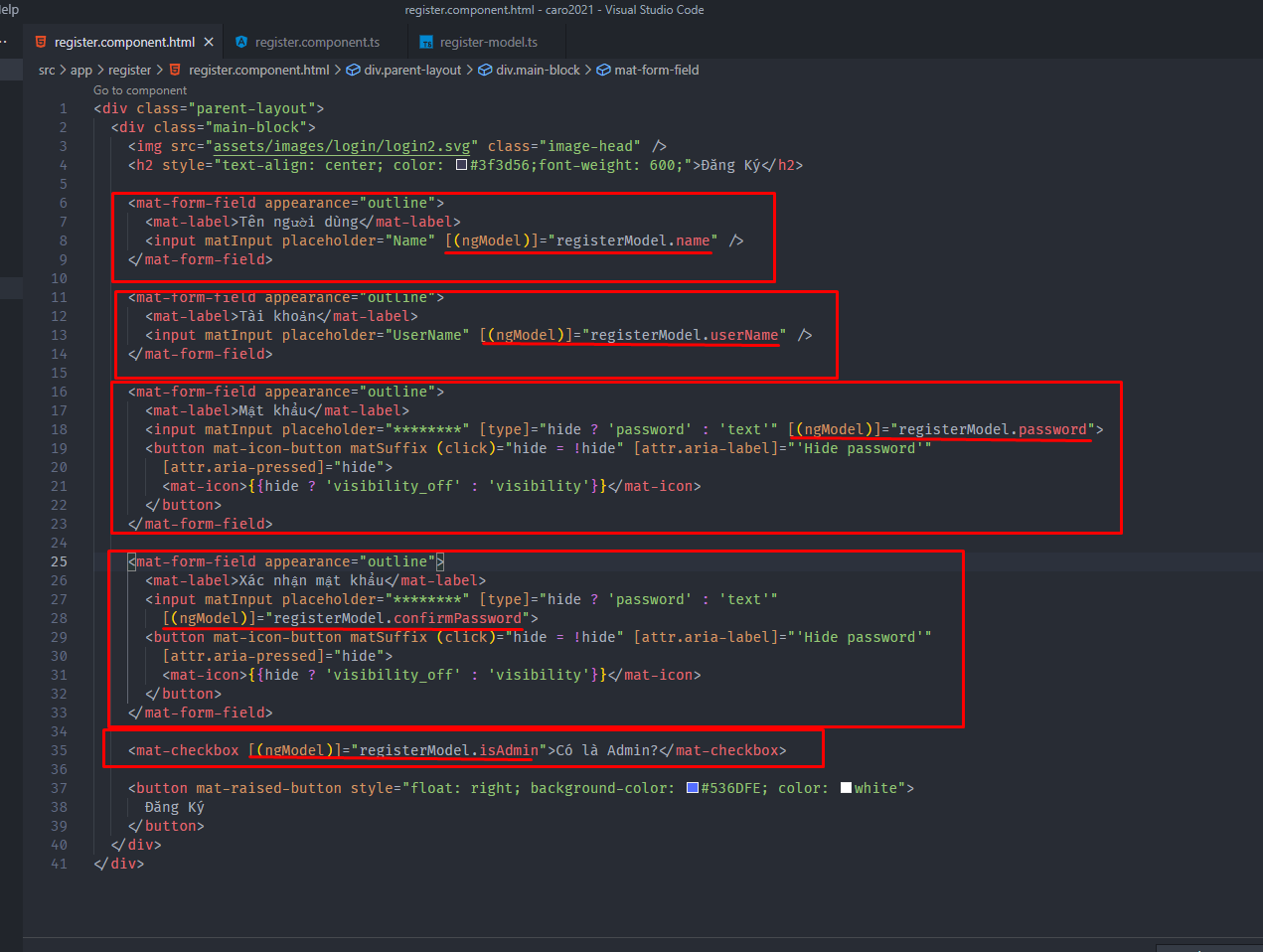
Thực hiện khởi tạo 1 biến có kiểu vừa class vừa khởi tạo để nhận các giá trị mà chúng ta nhập từ bàn phím.



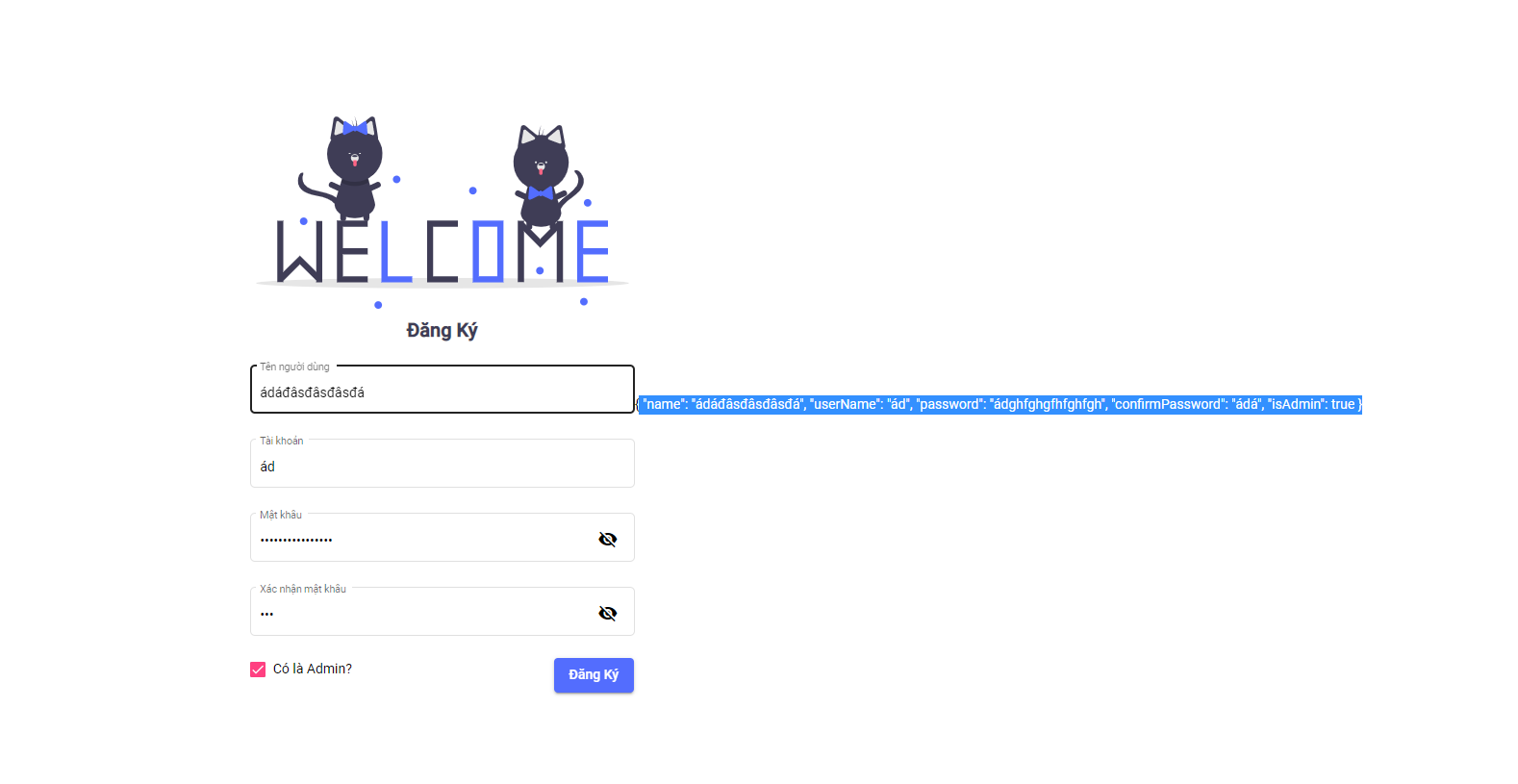
Trong một component chúng ta chú tới 2 thằng này constructor() và ngOnInit()

2 thằng là 2 func sẽ chạy đầu tiên khi component này được khởi tạo.

Thứ tự là construtor() sau đó là ngOnInit()



Thực hiện binding với cái biến mà chúng ta đã khai báo registerModel trong file .ts tương ứng.



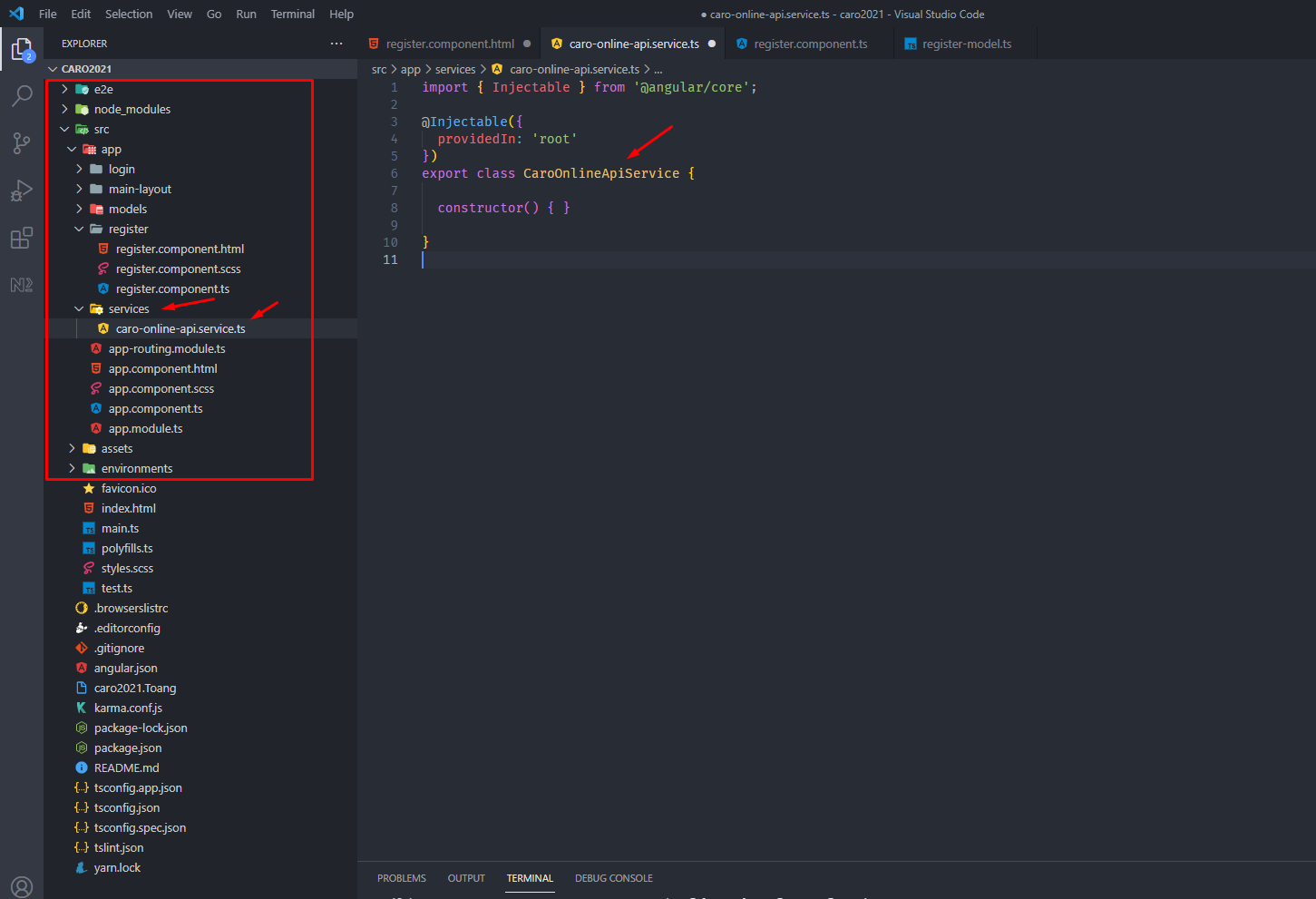
Chứng minh sử dụng ngModel thì các biến được binding với nhau

......

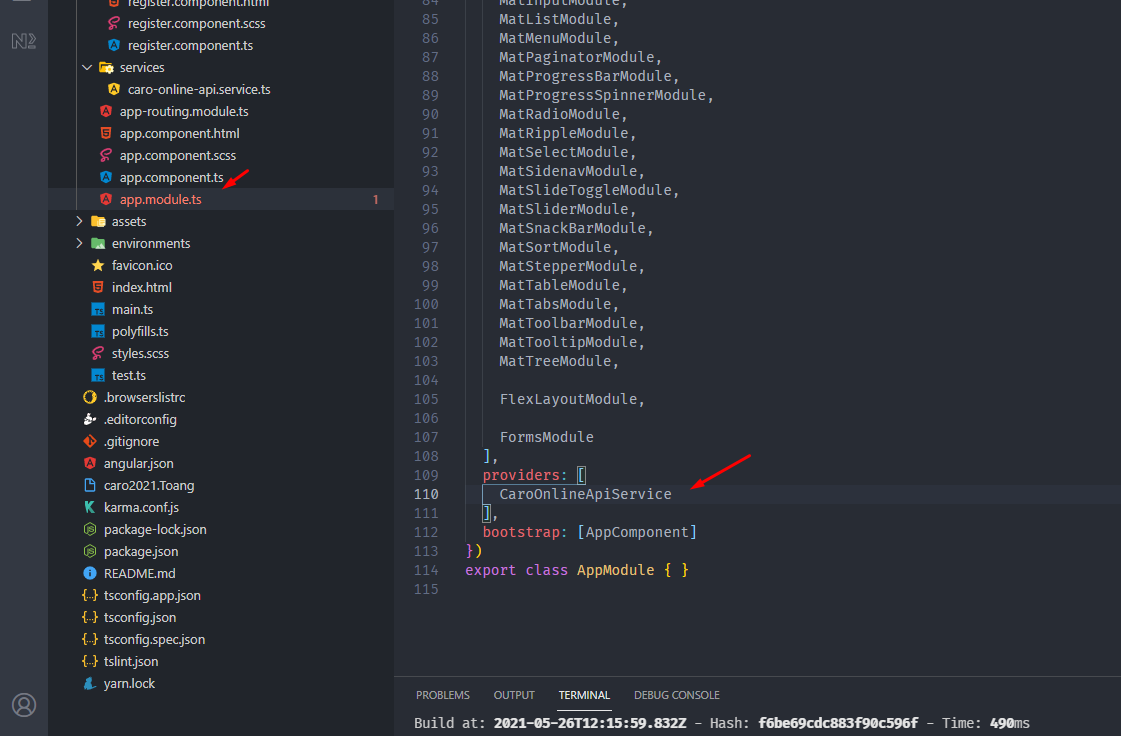
Câu chuyện tiếp theo là làm thế nào để gọi api tới server (back-end) từ front-end ???

Chúng ta viết services để thực hiện call thông qua cái thằng services này.

Với angular nó là file có đuôi là .service.ts



Giống như tất các component và module chúng ta từng sử dụng thì service cũng phải được import trong app.module.ts thì mới chạy được. ( do là để khai báo cho hệ thống biết rằng tôi có sử dụng services này trong con app của mình).



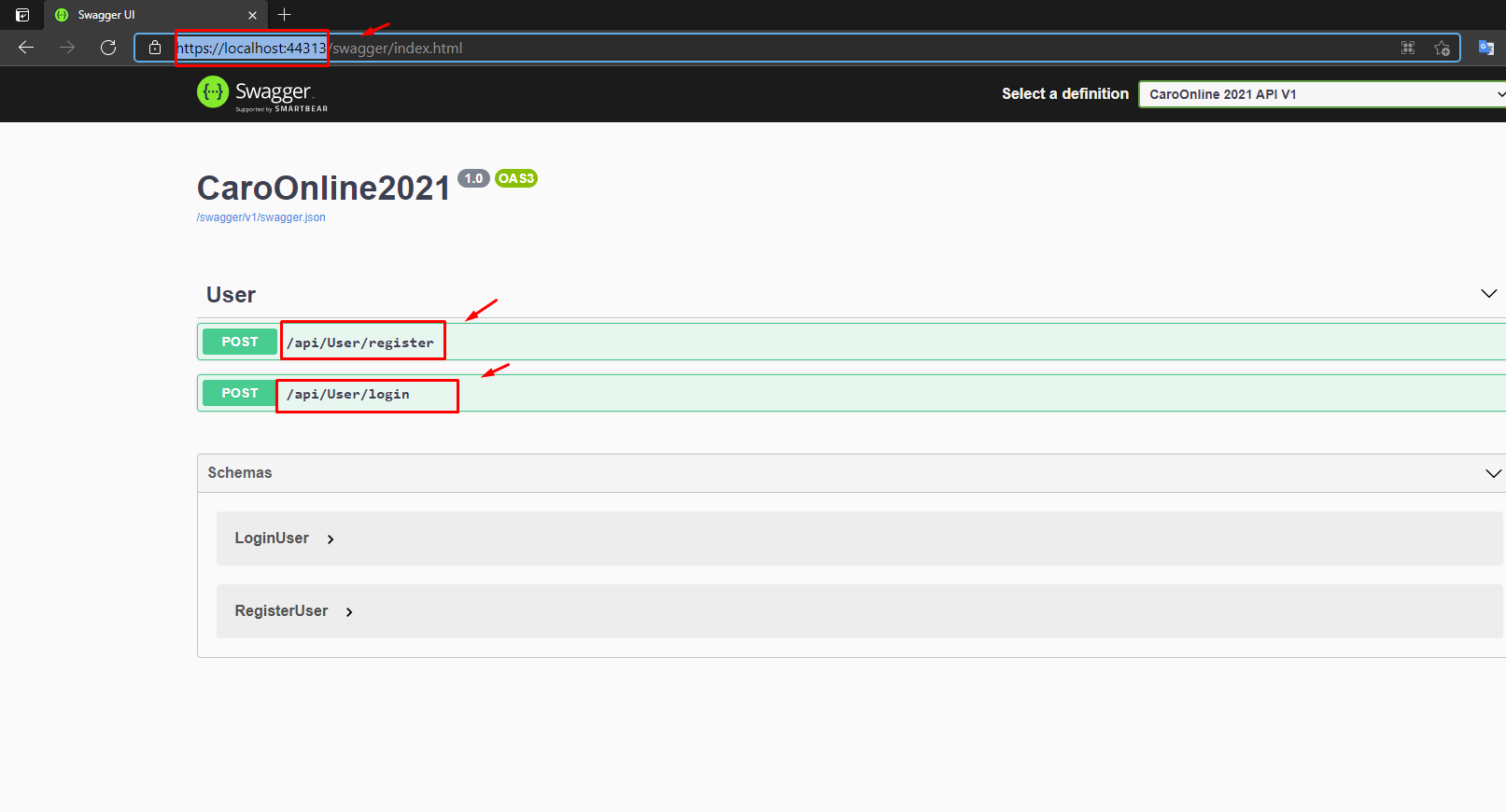
Được import trong mảng providers[] trong file app.module.ts

Muốn call được api chúng ta phải biết cái URL của api đó là gì?

Và phương thức của api đó là gì?

Và dữ liệu bắn lên cho api đó là gì?

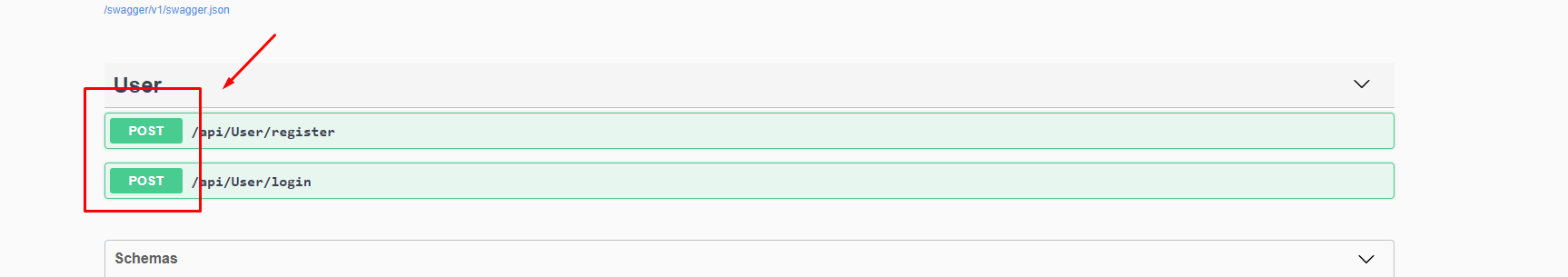
URL = <domain của server cần gọi> + <path của api trên con server đó>



url của api đăng kí là : <https://localhost:44313/api/User/register>

và login là: https://localhost:44313/api/User/login

(chú ý)



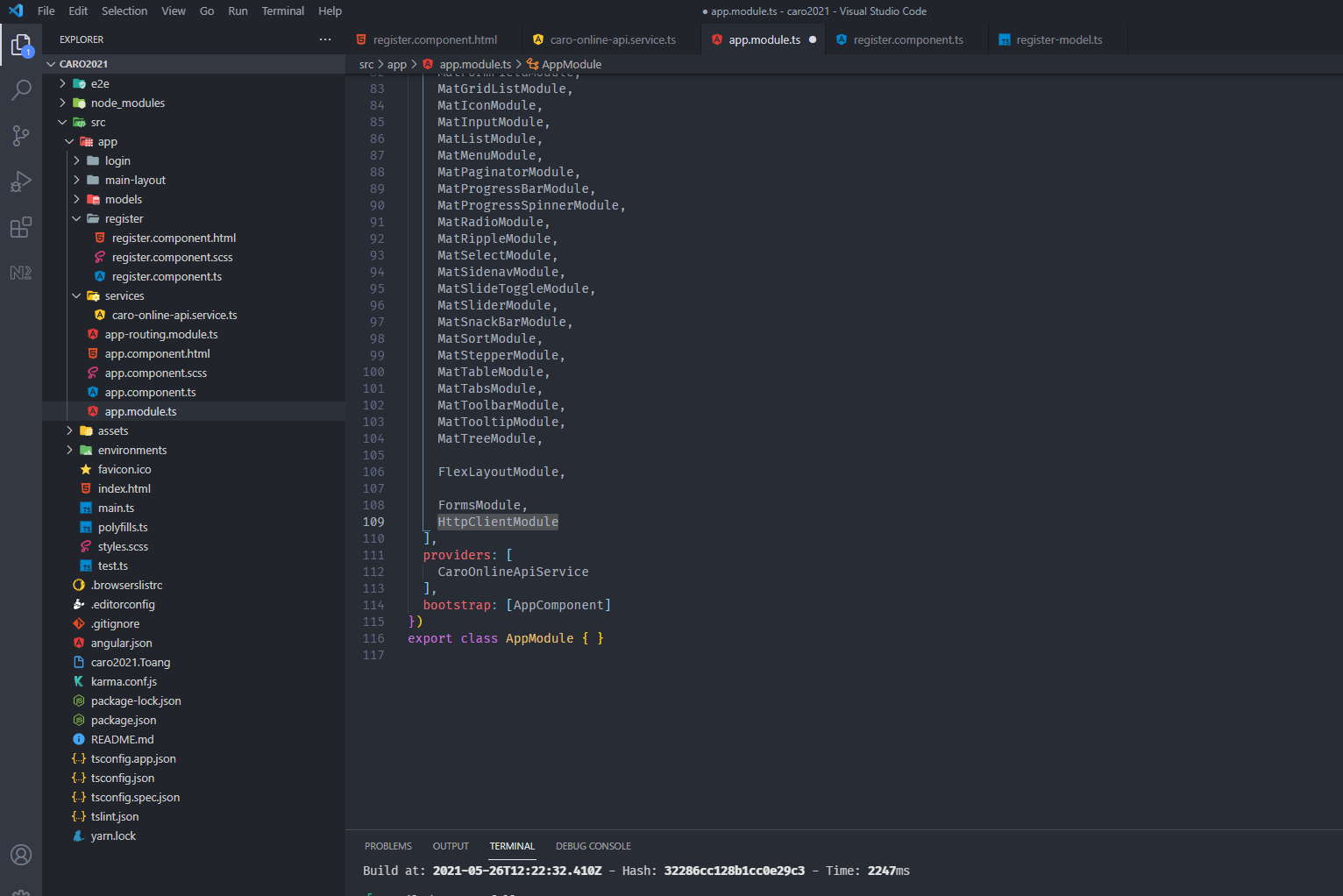
Đây chính là phương thức mà chúng cần khai báo khi gọi api này.

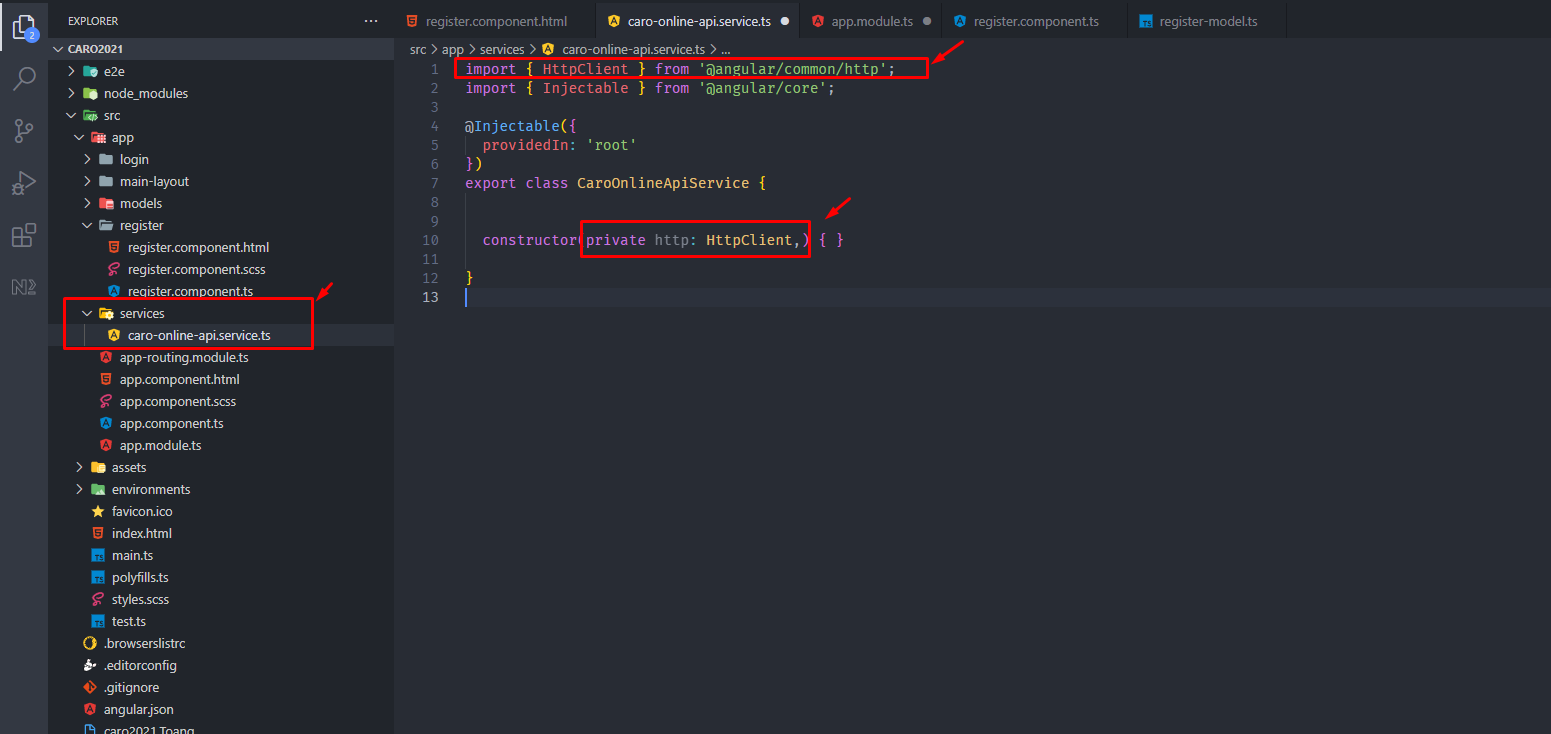
Phải đúng url và phương thức

Ở trong Angular cần sử dụng 1 module để thực hiện việc call này là HttpClientModule

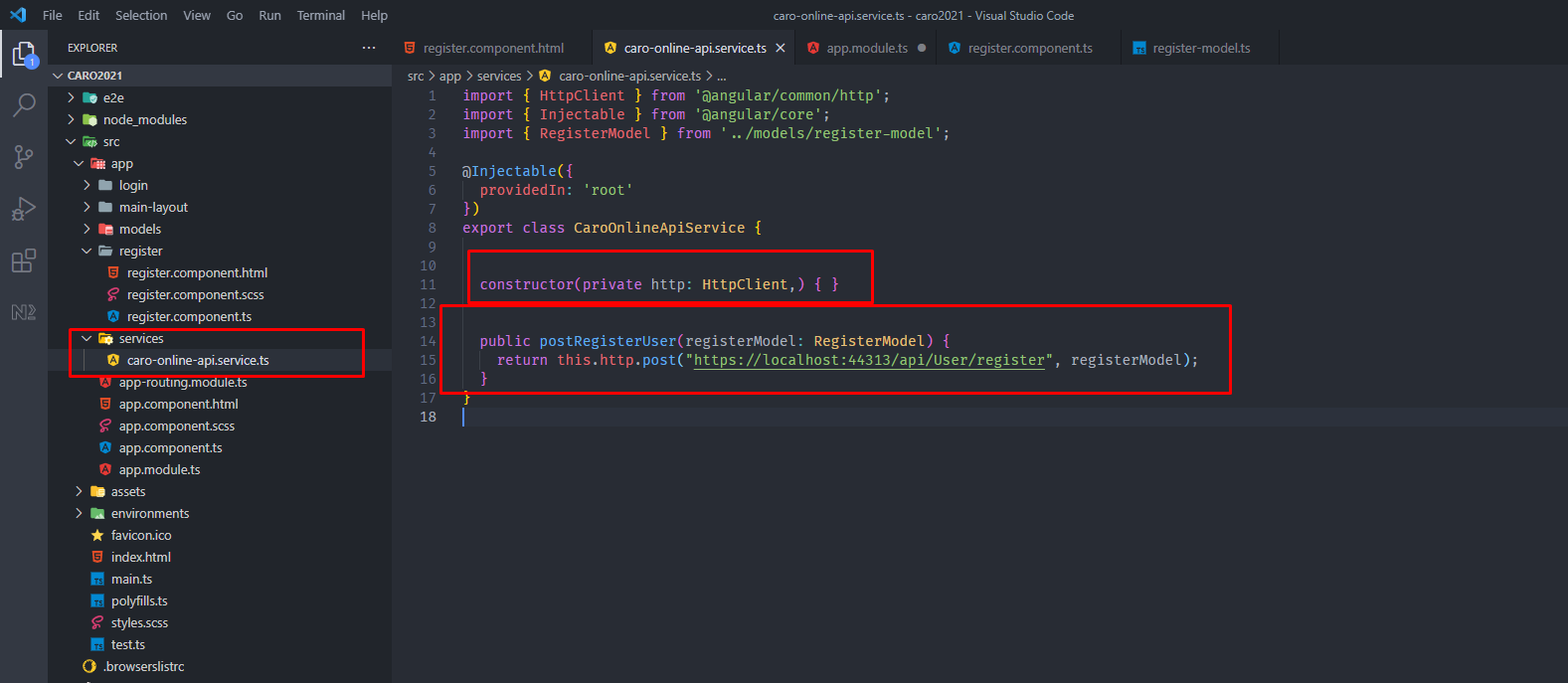
Muốn call api thì chúng ta phải import nó vào đâu ?- app.module.ts

HttpClientModule => vào mảng imports[] trong app.module.ts

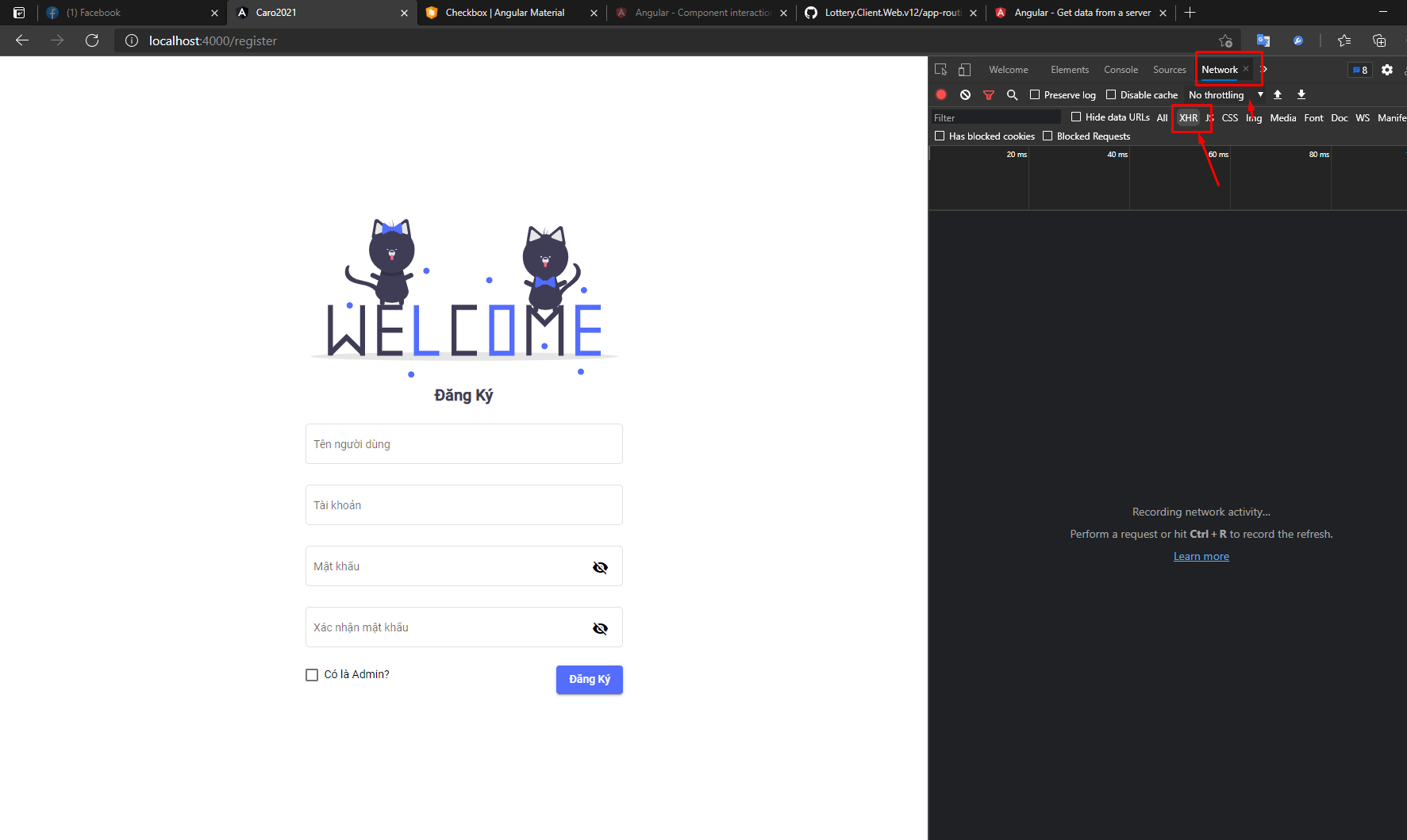


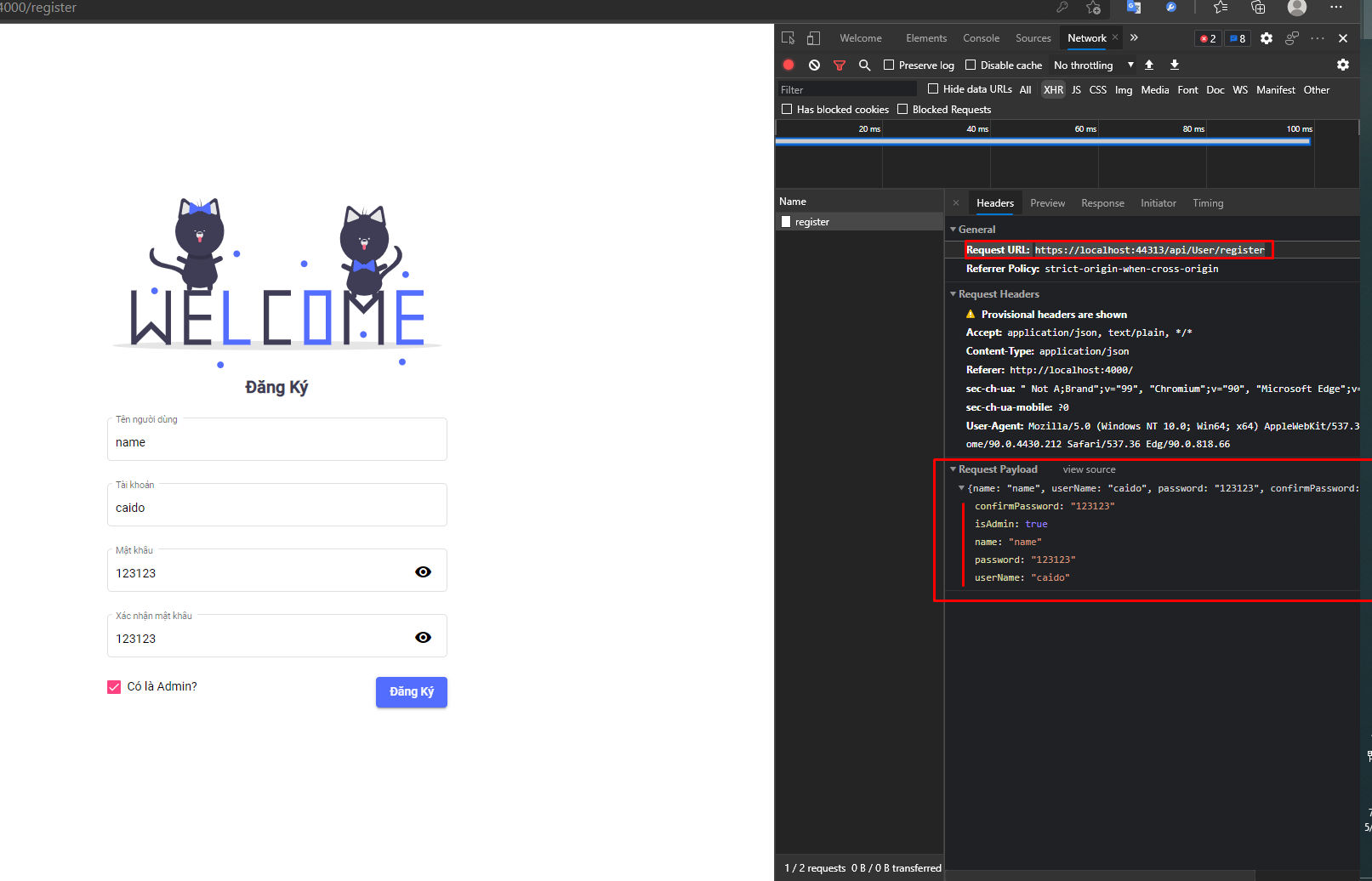


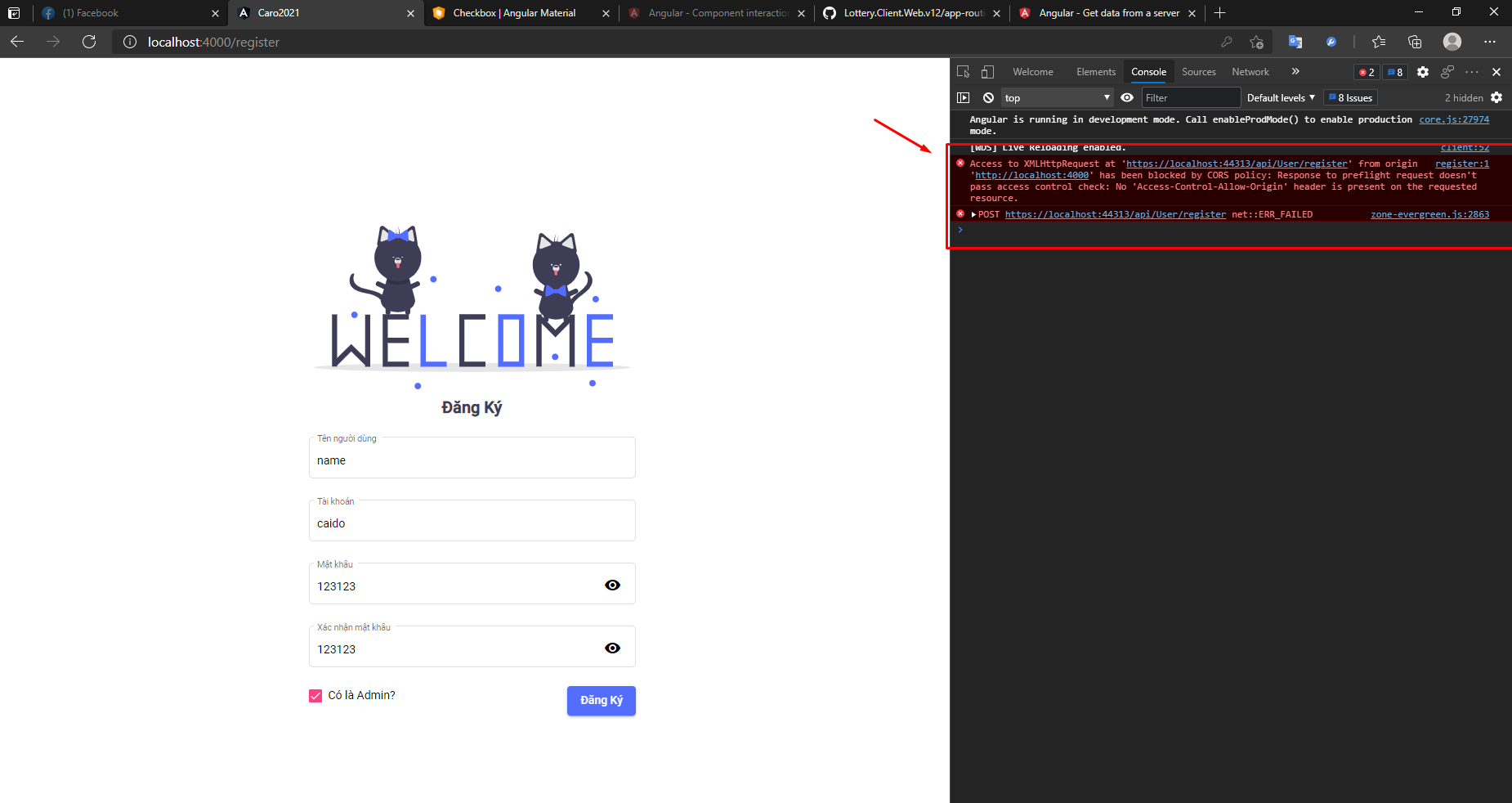
Thực hiện inject( tiêm) httpClientModule mà chúng ta vừa import trong app.module vào trong service để sử dụng.



Viết api đăng kí cho user trên front-end







Như này là chúng ta đã call api thành công (bắn 1 request đi tới server) nhưng bị lỗi

Lỗi ở đây là đang bị chặn CORS

Mà CORS là gì?

* Hiểu nôm na ở phía server nó chỉ tiếp nhận các request là từ các domain mà nó quen biết ( được khai báo từ trước)
* Giống như cái việc là bạn có 1 thẻ sinh viên để đi vào trường dhhp – hoặc bạn quen biết ô bảy vợ - thì chúng ta được phép đi vào vào trong và yêu cầu nhà trường làm cái gì đó( server phục vụ bạn 1 cái gì đó)

Lỗi này bắt buộc chúng ta phải sửa ở server chứ k thể sửa ở front-end đc

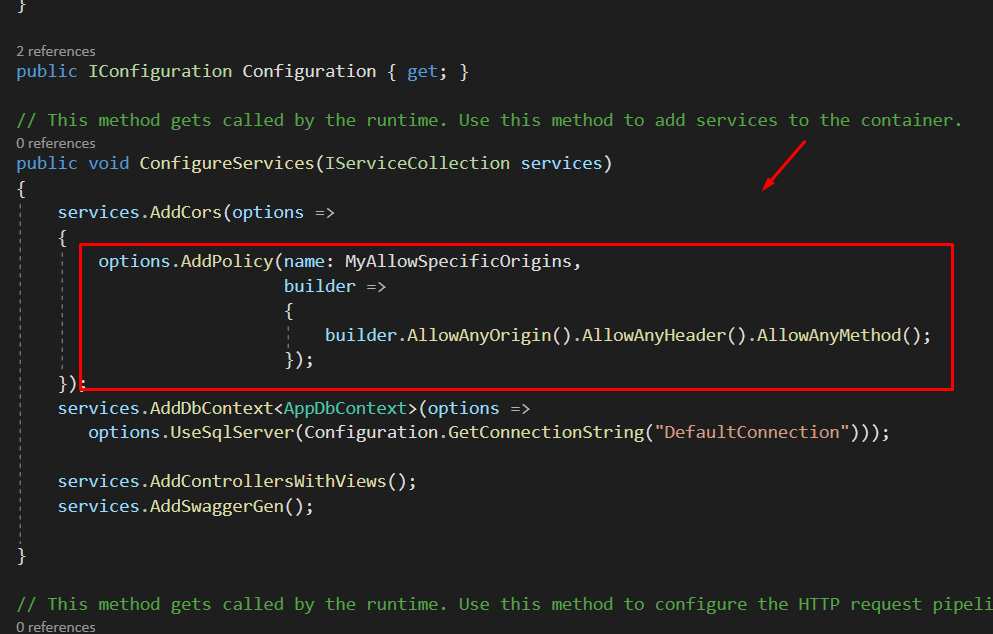
Thực hiện các bước sau để sửa

Coi như ai vào yêu cầu cũng đc, bất chấp method, bất chấp domain...

ở đây chúng ta k chú tâm tính bảo mật... ( mì ăn liền)

làm sao đó để front-end nó call 1 cách là thuận lợi nhất.

Link hướng dẫn của MS: <https://docs.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-5.0>



Làm y hệt trong doc của MS và sửa phần cấu hình như trên. Để chấp nhận toàn bộ request.