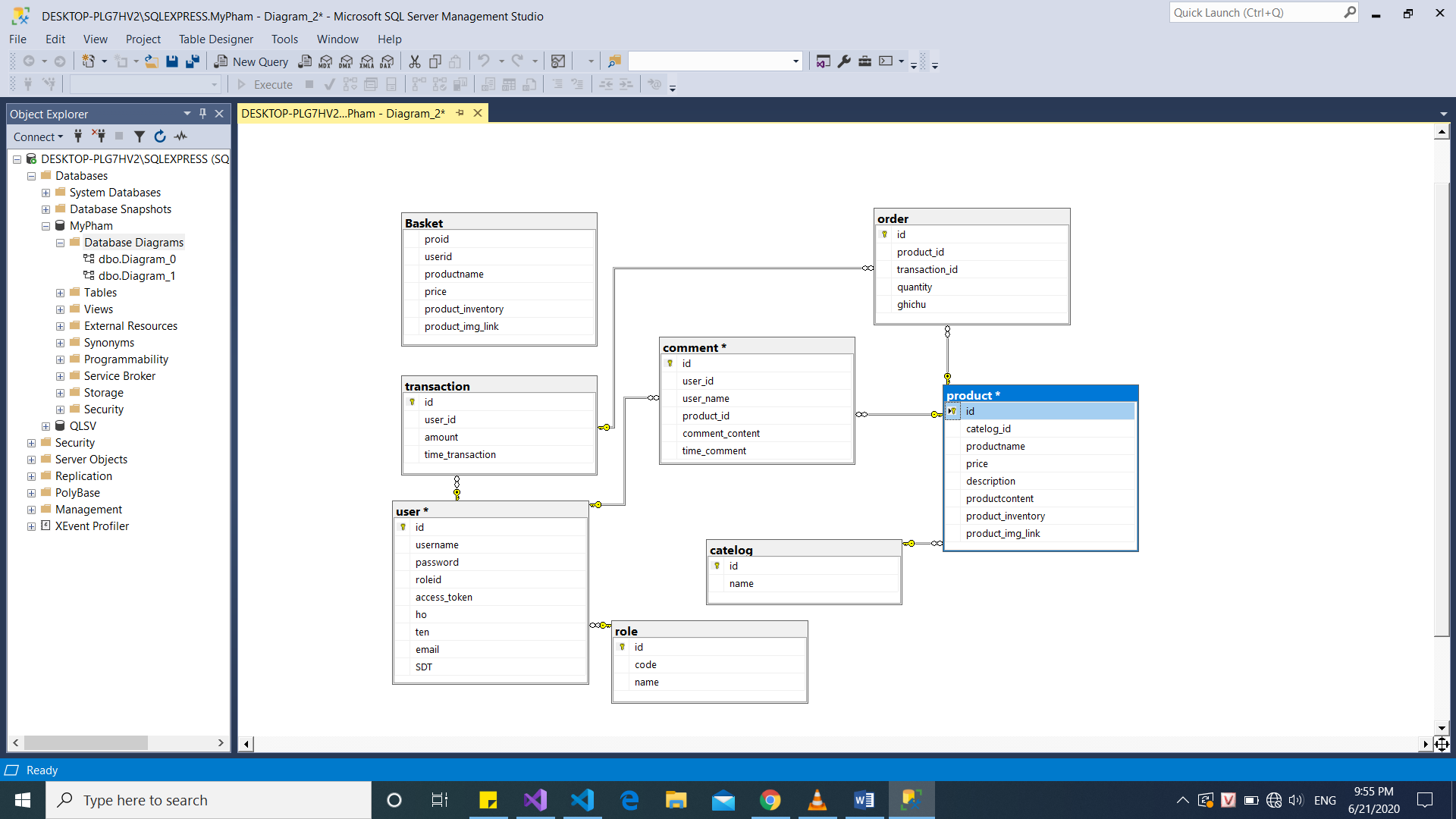
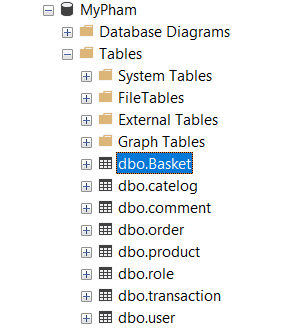
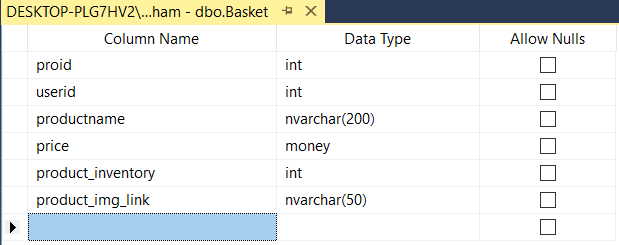
**Mô Tả Nghiệp Vụ**

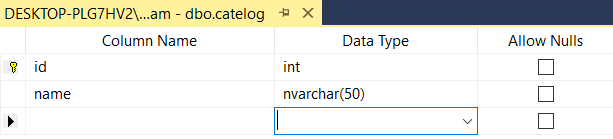
**Database**(Sử dụng SQL Server 2017)

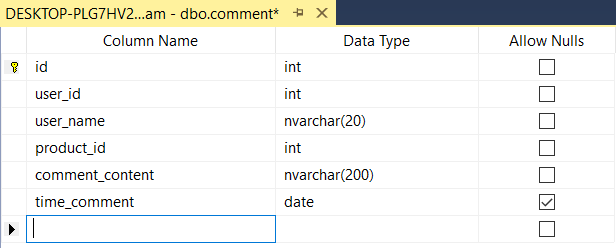
* Diagram

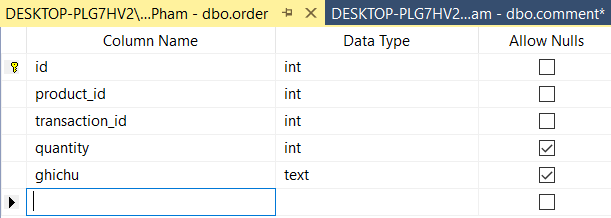


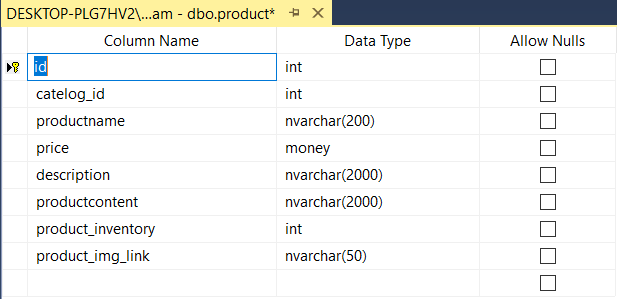


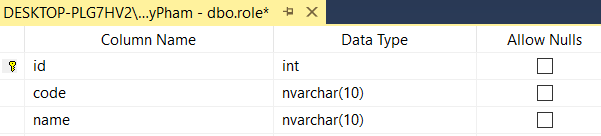


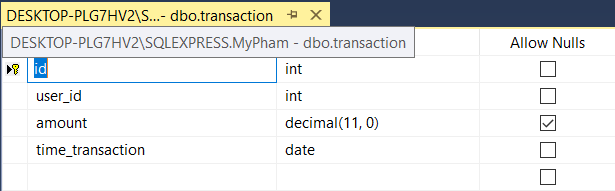


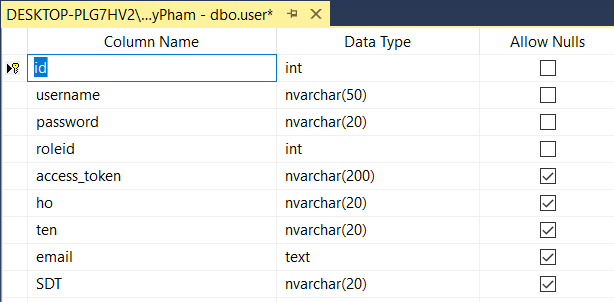












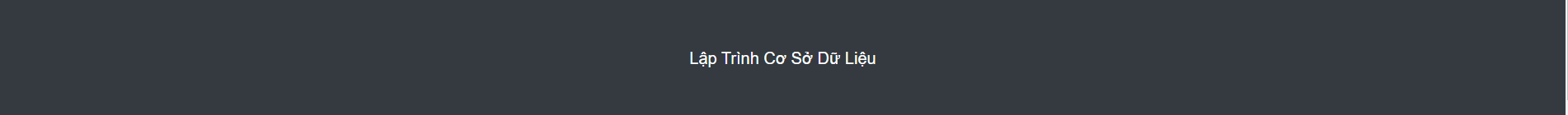
**Trang Chủ**

* Folder **TrangChu**(trangchu.component.html, trangchu.component.css, trangchu.component.ts, trangchu.service.ts)
* Các trang trong trang chủ sẽ có 3 phần chung là: header, menu, footer

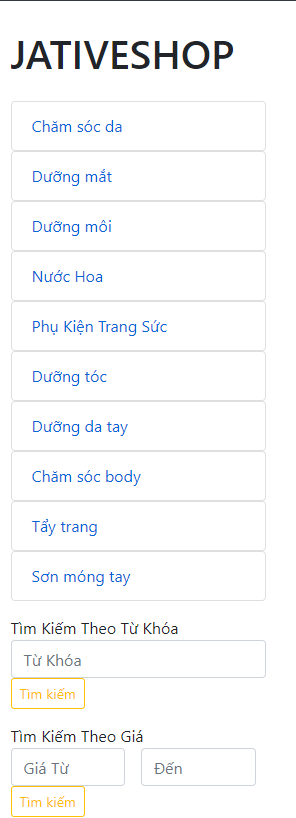
Header

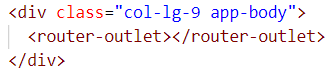


Footer



Menu

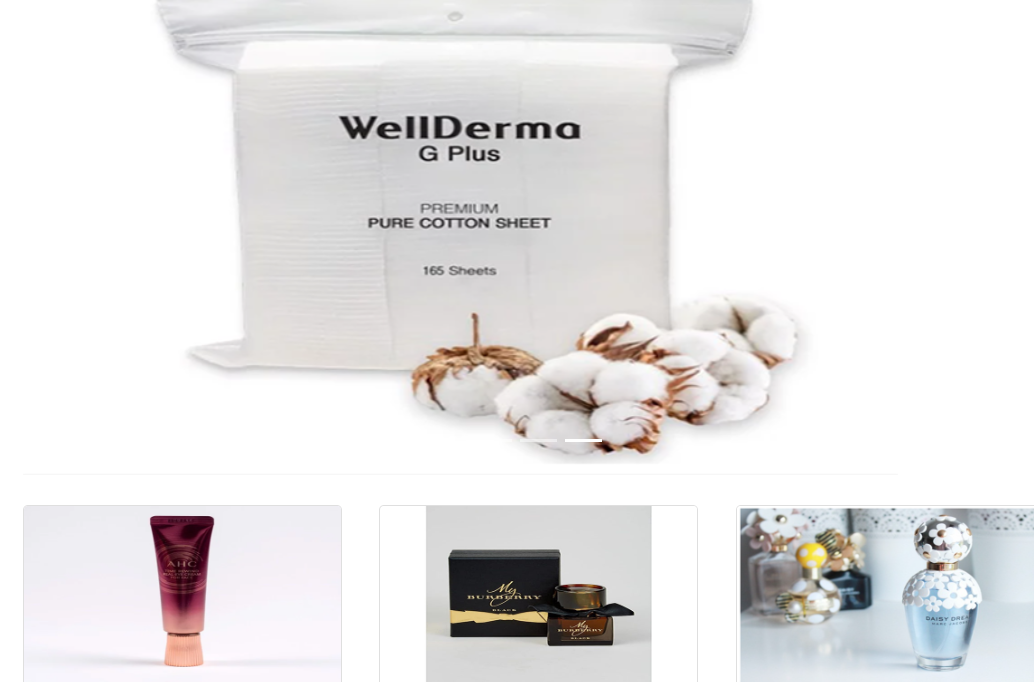


* Các phần sử dụng chung sẽ được viết trong file **trangchu.component..html**
* Và các phần hiển thị nội dung khác sẽ được viết trong folder khác là con của folder cha là TrangChu và để các phần có nội dung khác có thể sử dụng được phần chung là **header menu footer** thì ta sẽ định nghĩa 1 thuộc tính trong file **trangchu.component..html** là “<router-outlet>”

Phần chung **menu** trong body có width là 3/12, còn những phần có nội dung riêng trong body sẽ có width là 9/12

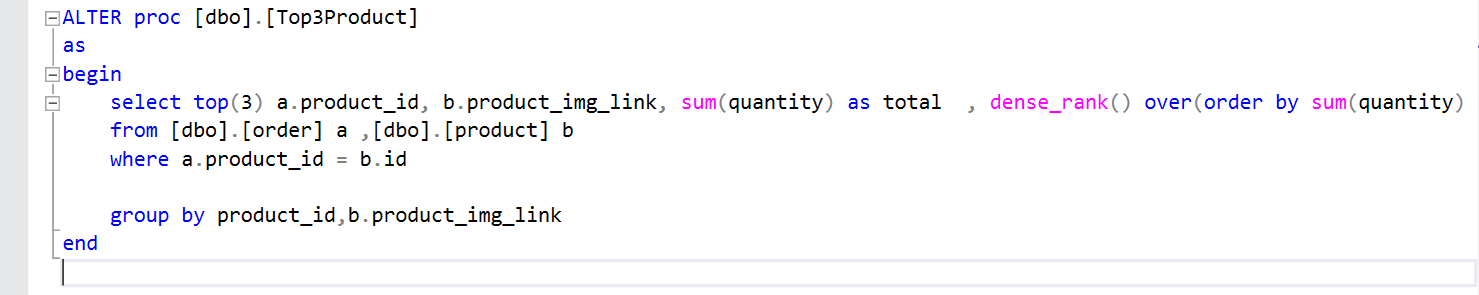
**Body trang chủ**

* Như trình ở trên, phần body chia làm 12 cột, phần **menu** có width 3 cột, còn phần hiển thị nội dung sản phẩm có diện tích là 9 cột
* **Phần hiển thị nội dung sản phẩm(**Folder **home** trongangular**)**:
* Sản phẩm có ảnh lớn là 1 trong 3 sản phẩm có lượng đặt hàng nhiều nhất
* Còn những ảnh nhỏ là lấy danh sách các sản phẩm có trong shop sắp xếp theo tên sản phẩm tăng dần và có phân trang



* API top 3 sản phẩm có nhiều người đặt hàng nhất

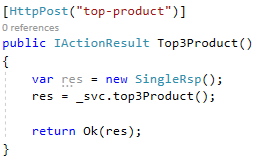
Stored Procedures



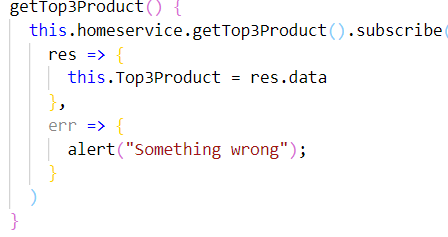
ProductRep



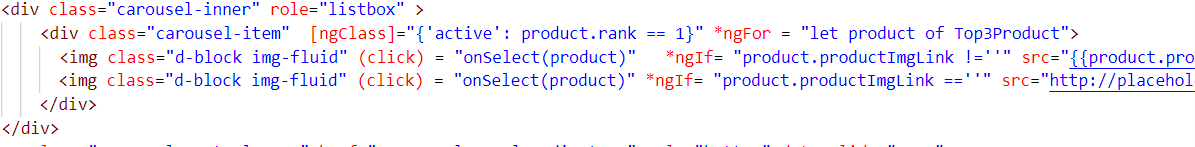
Controller



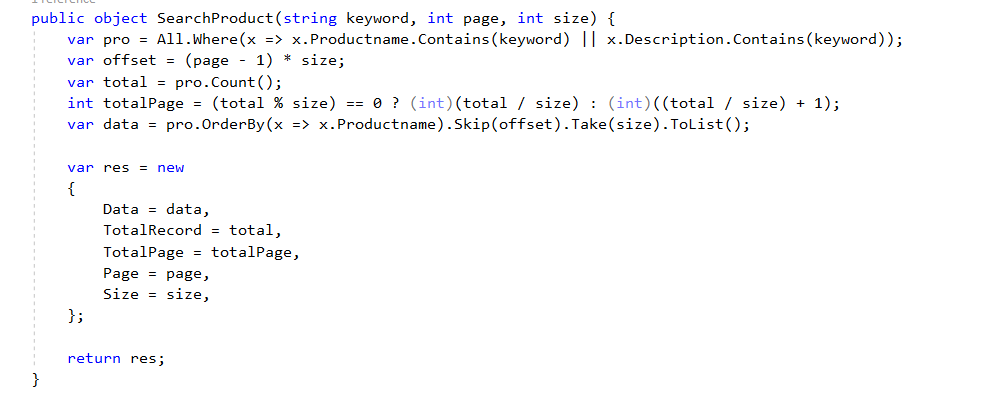
* Trong **home.component.ts**, chúng ta sẽ gọi API “top-product” là lưu 3 object này vào 1 array Top3Product

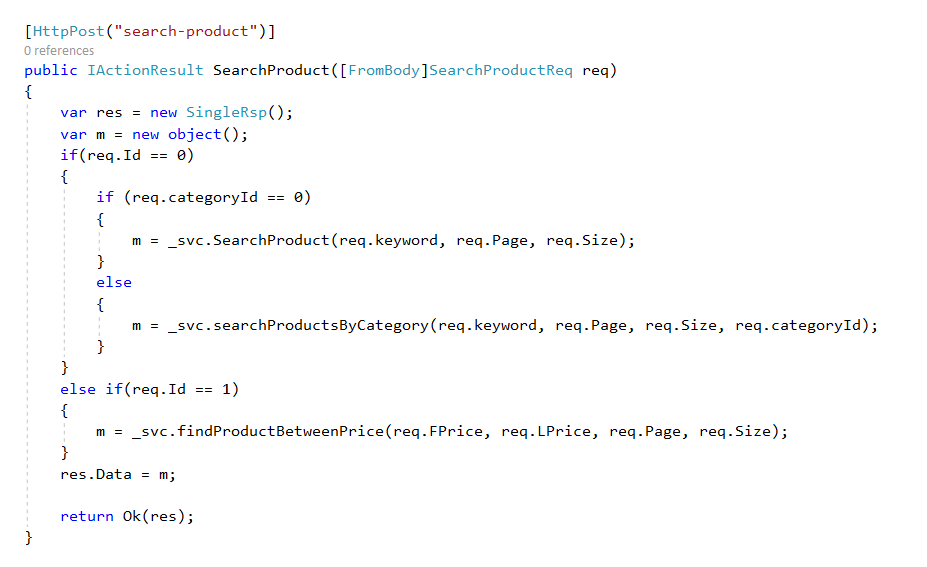


* Gọi vòng lặp 3 trong [**home.component.html**](http://home.component.html) để hiển thị top 3 sản phẩm, sản phẩm vào có **rank = 1** sẽ được hiển thị đầu tiên

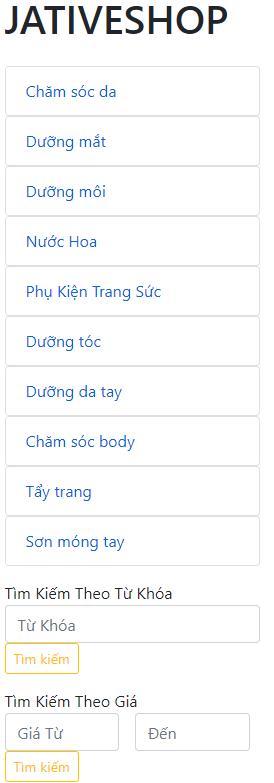


* Hiển thị danh sách sản phẩm, phân trang



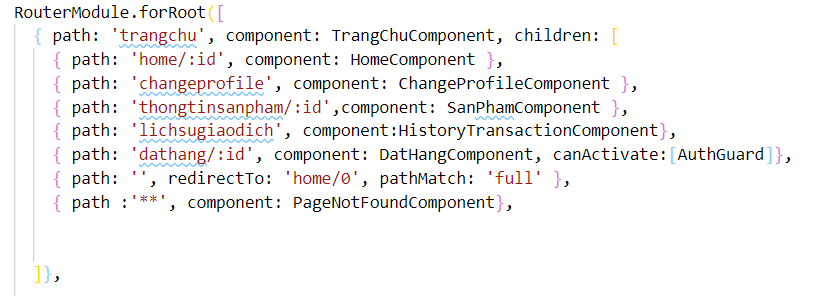


* **Menu**:

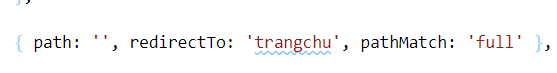


* Trên thanh menu sẽ chưa thông tin về các loại sản phẩm và hỗ trợ tìm kiếm
* Mỗi loại sản phẩm sẽ có một **id**(id > 0), để khi bấm vào loại sản phẩm đó sẽ tiến hành phân trang theo loại sản phẩm, còn mặc định khi vào **Trang chủ,** **id = 0,** để load thông tin tất cả sản phẩm
* Để gắn được id cho url (/id) thì trong **app.module.ts** ta set thuộc tính **enableTracing:** true





* Khi vào link trang chủ, nó sẽ tự route link đến <http://localhost:4200/trangchu/home/0> vì ở đây em đã thực hiện gắn liên kết khi router link là ‘’ nó sẽ tự redirect tới trang <http://localhost:4200/trangchu/home/0>

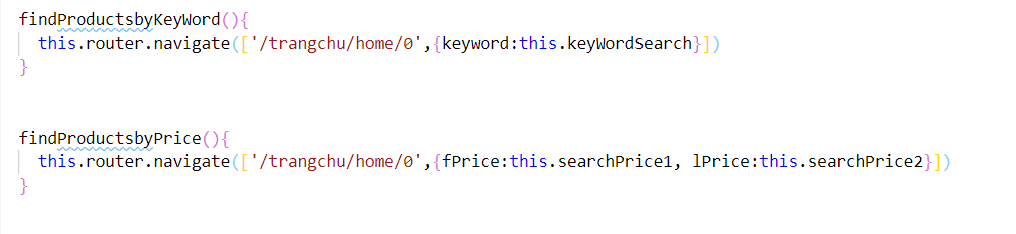




* Và mỗi lần ta bấm vào loại sản phẩm nó sẽ redirect tới trang home có id theo loại sản phẩm

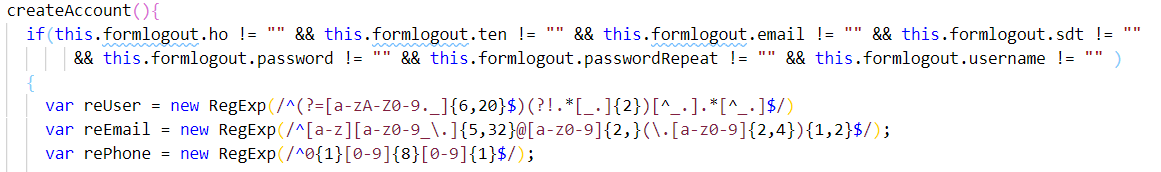


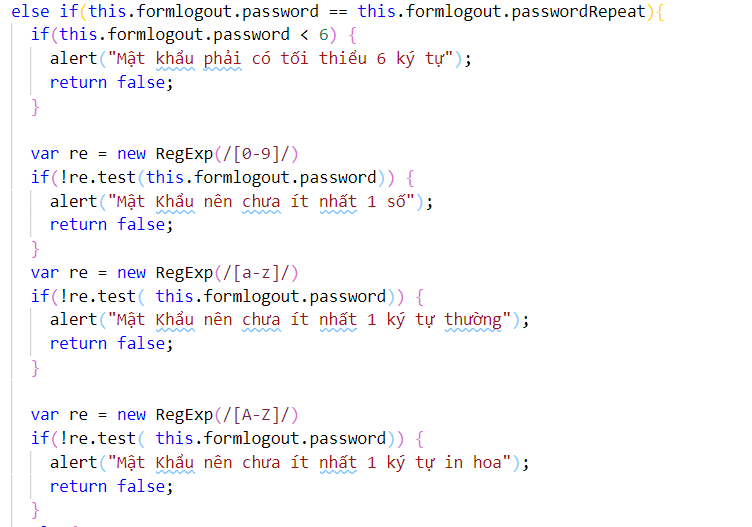
* **Tìm kiếm sản phẩm trang chủ**:
* Tương tự như là lấy các sản phẩm theo loại sản phẩm, tìm kiếm sản phẩm sẽ redirect tới trang **home/0,** bên cạnh đó còn gửi thêm keyword tìm kiếm đến trang home



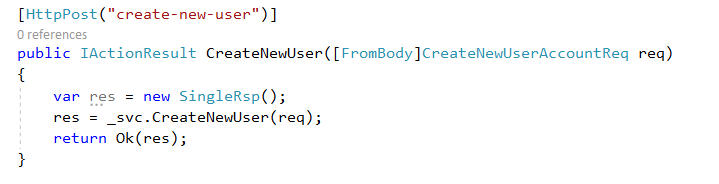
**Header Trang Chủ**

* **Đăng Ký:**
* Dùng **Regress Expression** để kiểm tra tính hợp lệ username, password,email, phonenumber
* Kiểm tra các giá trị khác rỗng, và password và repeat password phải giống nhau



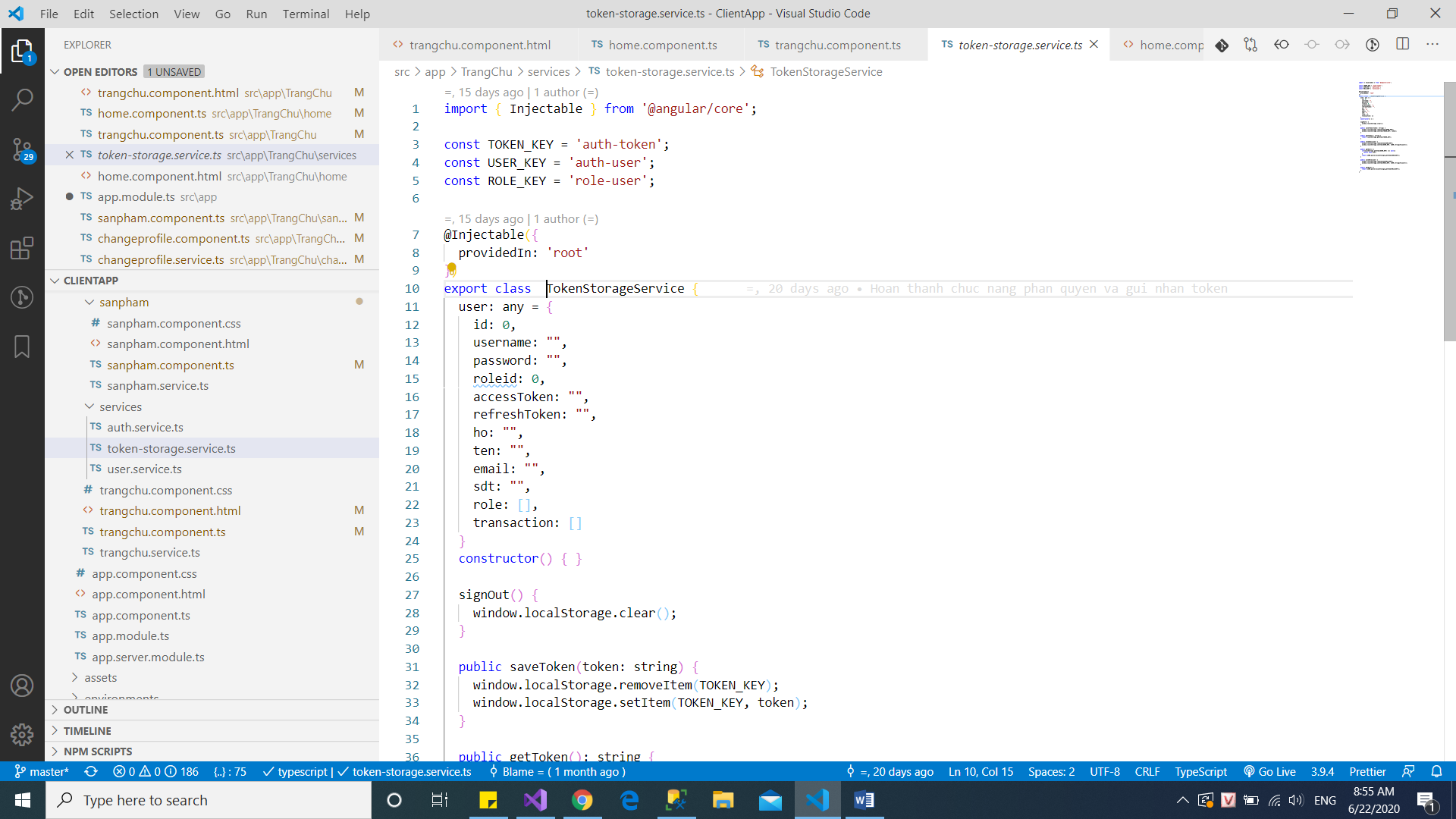


* API:



* **Đăng Nhập**
* **Phân Quyền**:
* Mỗi User sẽ có 1 thuộc tính Role để kiểm tra người dùng là **ADMIN** hay **USER**
* Khi người dùng đăng nhập thành công vào vào hệ thống thì sẽ dùng **window localStorage** để lưu lại thông tin ngươi dùng

****



****

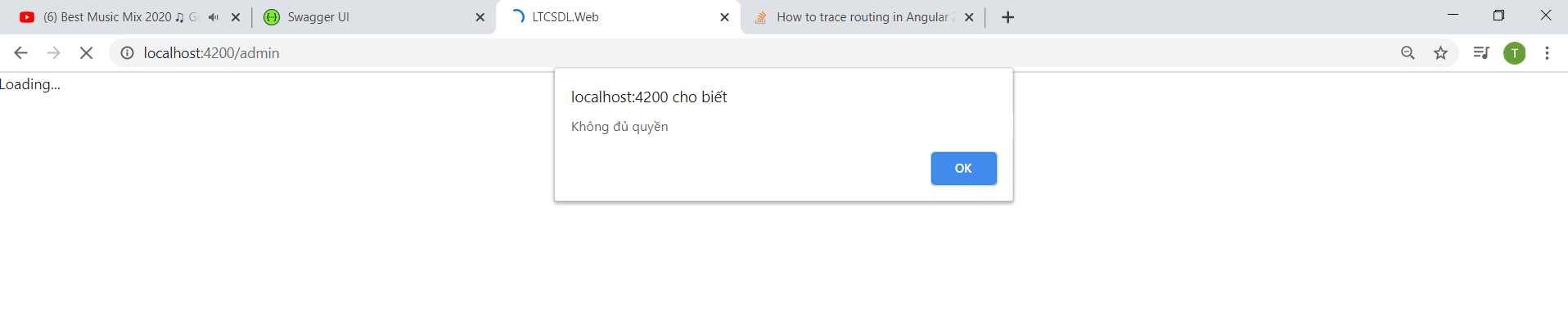
* Sử dụng **Guard** để phân quyền người dùng
* Từ local storage ta sẽ lấy Role để set quyền, nếu role là ADMIN thì cho phép vào trang admin, còn không sẽ thông báo không đủ quyền
* Kiểm tra khi có User nào đó muốn vào trang admin

****

* Trong app.component.ts, chúng ta sẽ set thêm thuộc tính **canActive** để kiểm tra quyền mỗi khi có người muốn vào trang ADMIN
* canActivate:[AuthAdminGuard] gọi đến class AuthAdminGuard để kiểm tra có thể activate vào trang ADMIN hay không



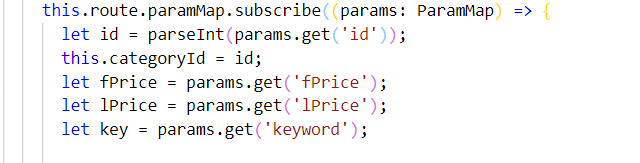
* Khi dùng tài khoản có quyền USER mà cố gắng truy cập vào trang ADMIN



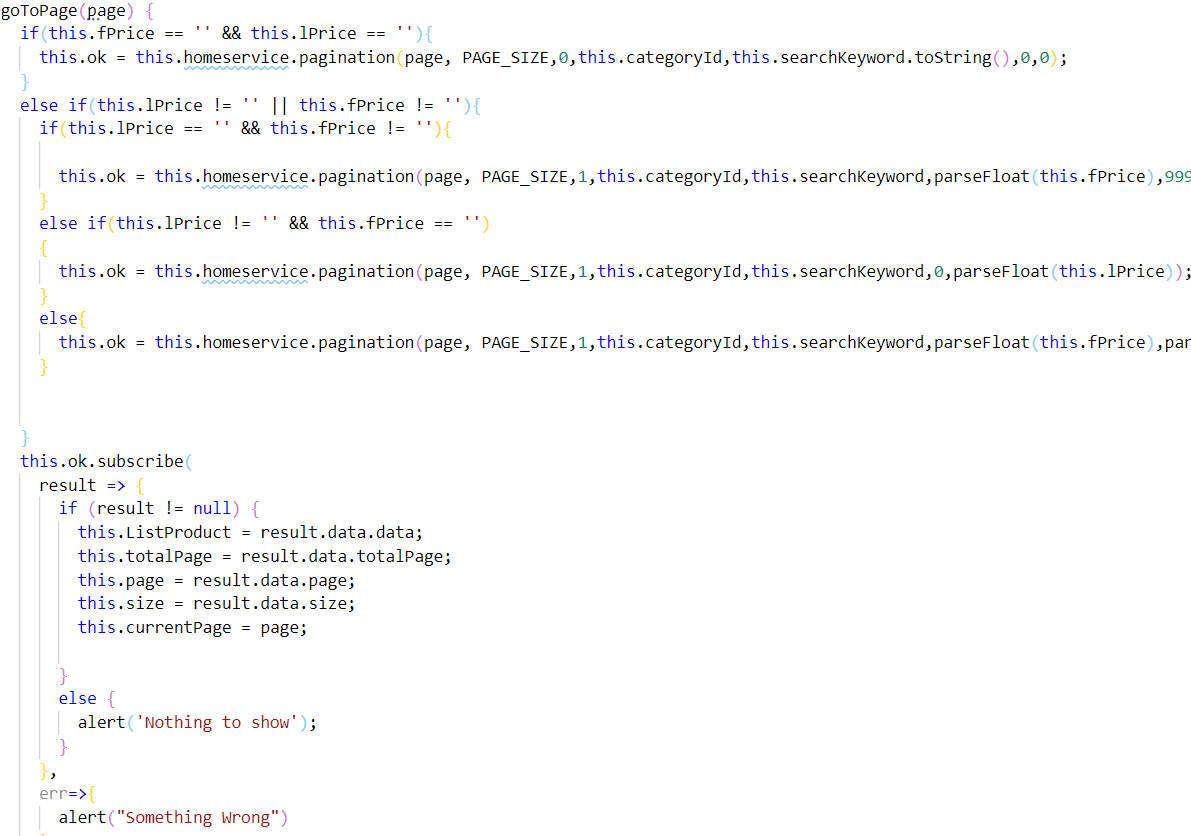
* **Danh sách sản phẩm(Folder Home)**
* Như đã nói ở phần body, khi vào trang chủ t sẽ gán thuộc tính gửi tới url: /home/0, khi muốn lấy danh sách sản phẩm theo loại sản phẩm url: /home/id,

Khi tìm kiếm theo giá hoặc theo mô tả sản phẩm, thì tới url: /home/0;keyword = ? hoặc /home/0;fPrice=?;lPrice=?

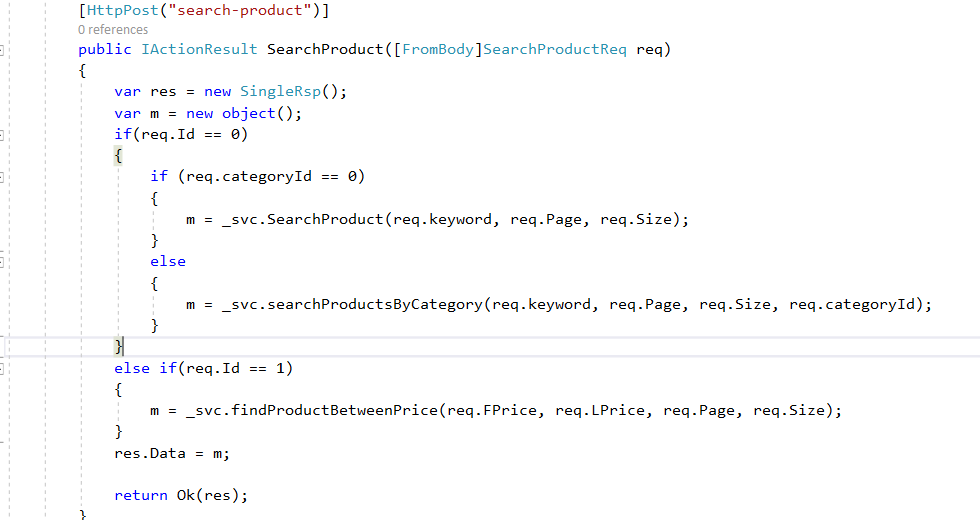
* Vậy để có thể lấy được yêu cầu trang chủ, thì trong [home.component.ts](http://home.component.ts) ta phải lấy được các yêu cầu từ phía trang chủ để hiển thị phân trang
* Trong constructor [home.component.ts](http://home.component.ts), sử dụng paramMap để lấy yêu cầu



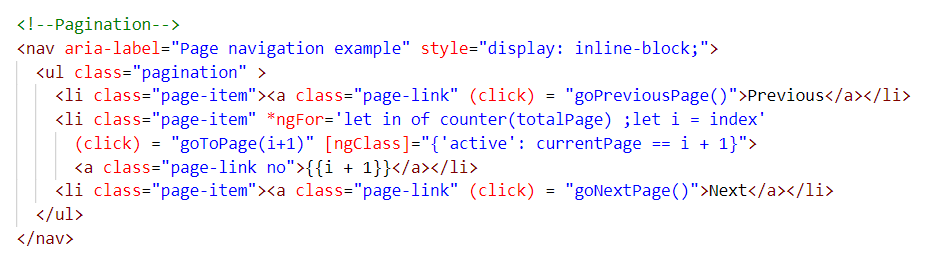
* Để có đầy đủ giá trị, bây giờ chỉ cần phân trang theo yêu của
* Viết chung phần phân trang của tất cả yêu cầu vào method gotoPage
* Nếu tìm kiếm theo giá thì cho id = 0, nếu tìm kiếm theo giá thì cho id = 1

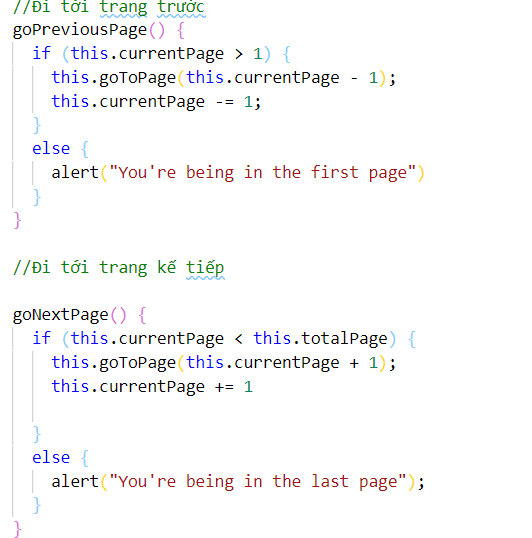


API

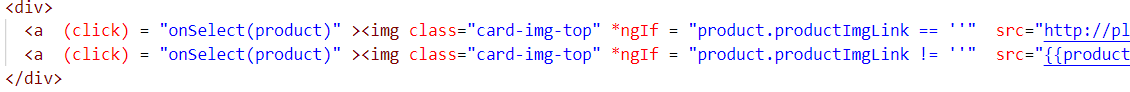


* Phân trang:

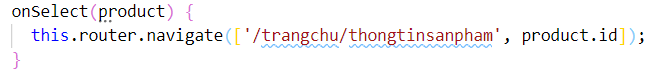




* Mỗi sản phẩm đều có một ID, Để xem thông tin sản phẩm, click vào sản phẩm , và trang home sẽ gửi yêu cầu đến **/trangchu/thongtinsanpham/id**

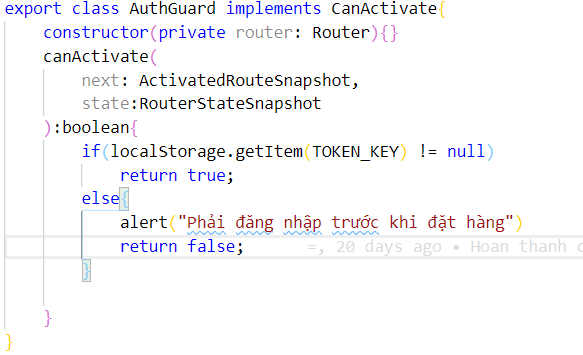






**Trang Thông Tin Sản Phẩm**

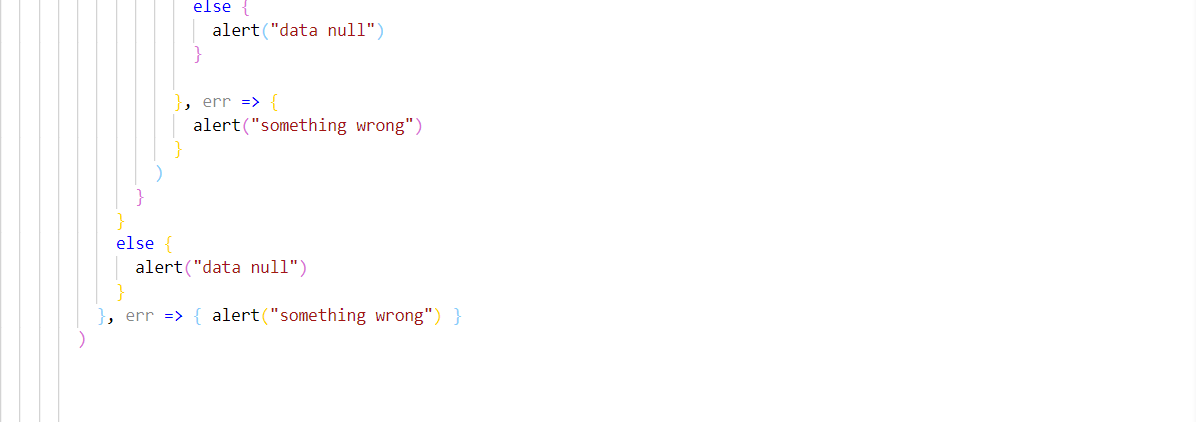
* Folder **sanpham(**sanpham.component.html, sanpham.component.css, sanpham.component.ts, sanpham.service.ts**)**
* Đặt Hàng: để đặt được hàng thì chúng ta cần phải đăng nhập trước, và để kiểm tra đã có đăng nhập hay chưa thì chỉ cần xem trong **localStorage** có lưu thông tin USER hay không(Khi đăng nhập window localStorage lưu thông tin người dùng) tương tự như việc kiểm tra khi vào trang ADMIN



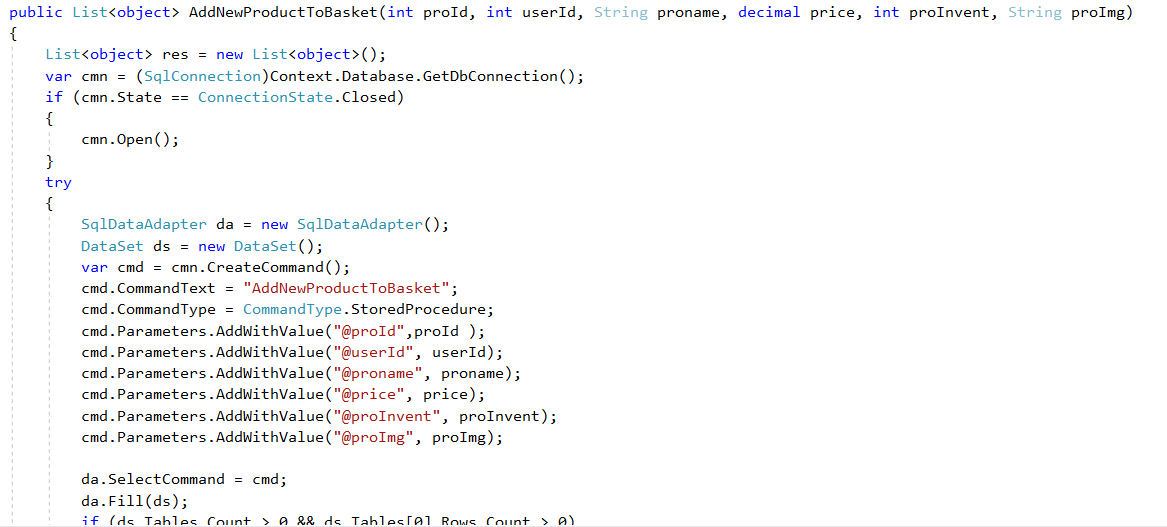


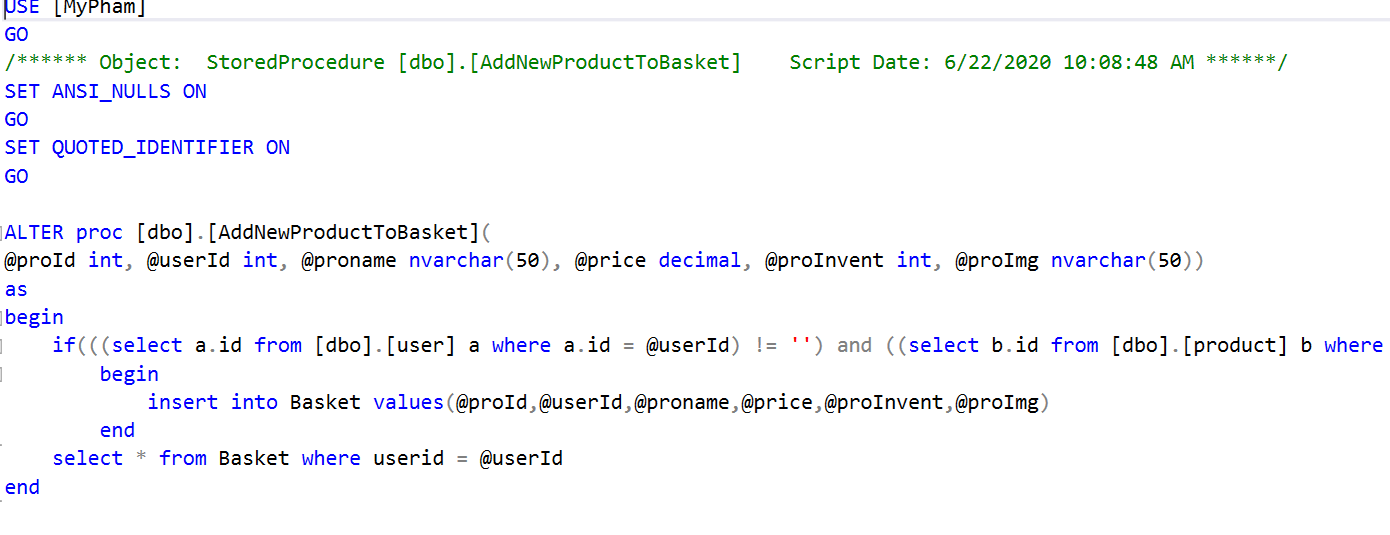
* Các thông tin về sản phẩm được đặt sẽ lưu trong Giỏ Hàng(Basket)
* Khi đặt hàng kiểm tra trong giỏ hàng đã có tồn tài sản phẩm này chưa, nếu có tồn tại ta chỉ cần cập nhập lại số lượng sản phẩm đó còn nếu không thì tạo mới sản phẩm đó trong giỏ hàng



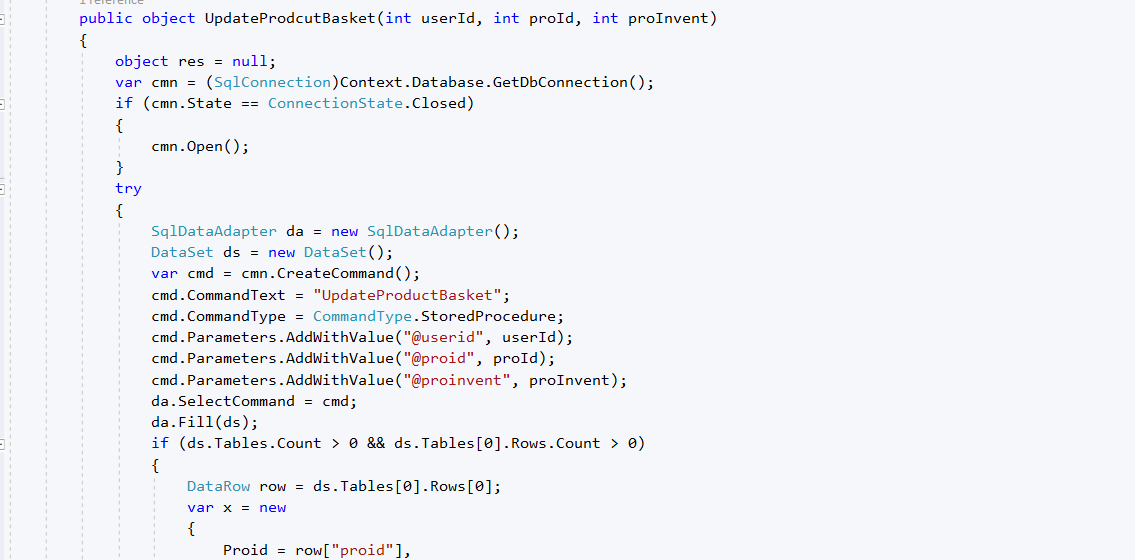


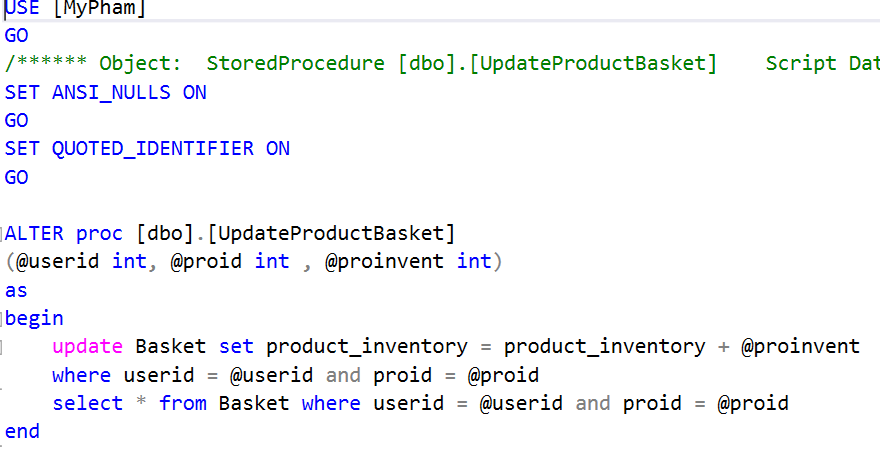
API create product basket



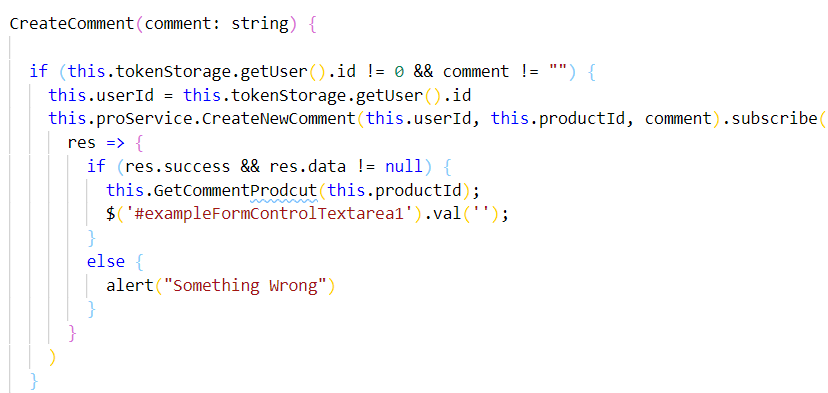


API Update product basket

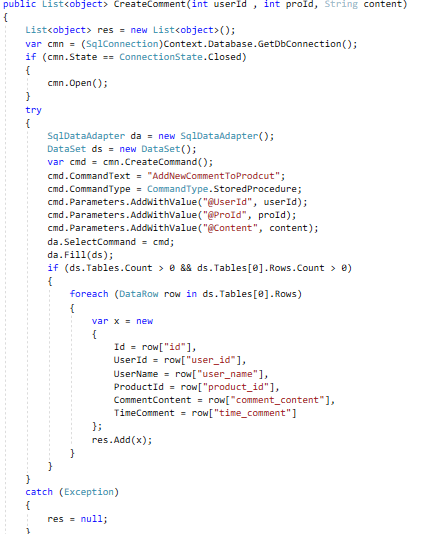




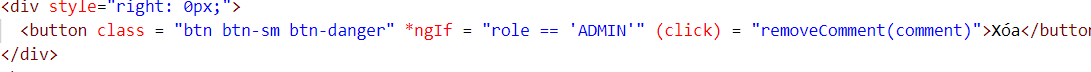
* **Comment bài viết**: tương tự như đặt hàng ta cũng phải kiểm tra xem đã có Authentication trước

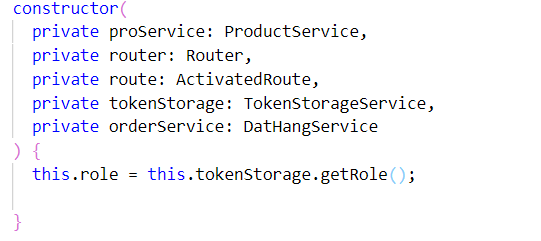


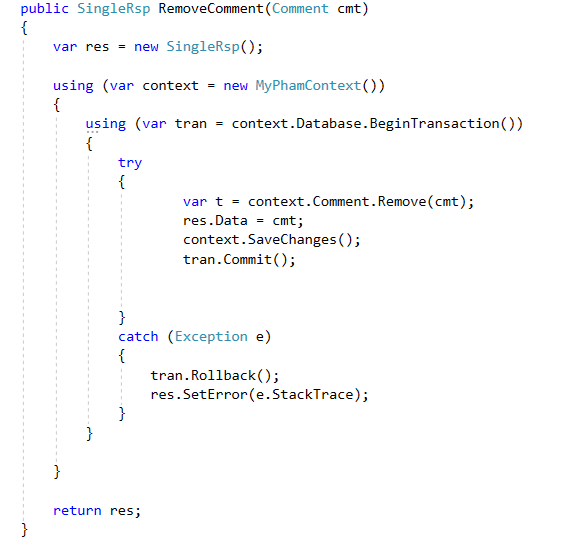
API comment



* Xóa comment: Kiểm tra nếu user là ADMIN thì tạo nút xóa trên mỗi comment

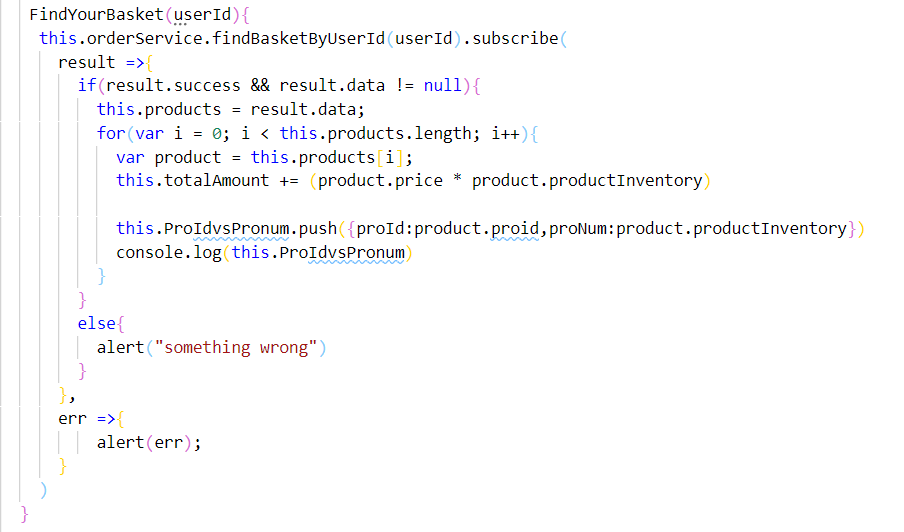




API Xóa Comment

**Đặt Hàng:**

* Folder **dathang(**dathang.component.html, dathang.component.css, dathang.service.ts, dathang.component.ts**)**
* Tương tự như trong [home.component.ts](http://home.component.ts), trong dathang.component.ts cũng sử dụng paramMap để hứng userId từ home để hiển thị Giỏ Hàng của user đó
* Để có được tổng giá tiền của đơn hàng thì ta sẽ chạy vòng lặp các sản phẩm trong Giỏ hàng và cộng tổng giá trị của từng sản phẩm
* Tương tự như phần Comment, để có thể xóa sản phẩm ra khỏa Giỏ Hàng chỉ cần gọi api xóa sản phẩm trong giỏ hàng

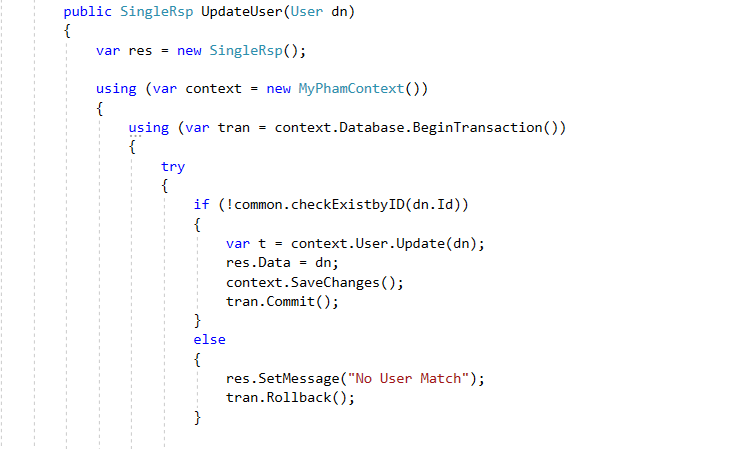


**LỊCH SỬ GIAO DỊCH:**

* Folder **HistoryTransaction**(historytransaction.component.html, historytransaction.component.css, historytransaction.component.ts, historytransaction.service.ts)
* Xóa các Giao dịch có giá trị trong ngày(Xóa tương tự)
* Sửa các giao dịch có giá trị trong ngày
* Xóa các sản phẩm trong giao dịch(Xóa tương tự)

**Sửa Thông Tin Tài Khoản:**

* Folder **changeprofile(**changeprofile.component.html, changeprofile.component.css, changeprofile.component.ts, changeprofile.service.ts**)**
* Lấy thông tin input từ form qua [(ngModel)] và cập nhập thông tin người dùng bằng API update-user



**ADMIN-PRODUCT**

* Folder **admin-product**(admin-product.component.html, admin-product.component.css, admin-product.component.ts, admin-product.service.ts)
* Phân trang, tìm kiếm sản phẩm tương tự ở phần home, và trang chủ
* Xóa sản phẩm tương tự như các phần trước
* Sửa và thêm sản phẩm cũng tương tự gọi API create hoặc update(có sử dụng **ckeditor 4** để sửa, thêm nội dung)

**ADMIN-CATELOG**

* Folder **admin-catelog**(admin-catelog.component.html, admin-catelog.component.css, admin-catelog.component.ts, admin-catelog.service.ts)
* Chức năng thêm sửa xóa tương tự như ADMIN-PRODUCT

**ADMIN-USER**

* Folder **admin-user**(admin-user.component.html, admin-user.component.css, admin-user.component.ts, admin-user.service.ts)
* Chức năng thêm sửa xóa tương tự như ADMIN-PRODUCT

**ADMIN-TRANSACTION**

* Folder **admin-transaction**(admin-transaction.component.html, admin- transaction.component.css, admin- transaction.component.ts, admin- transaction.service.ts)
* Chức năng thêm xóa, phân trang tương tự như ADMIN-PRODUCT
* Tìm kiếm theo ngày tương tự như tìm kiếm theo giá ở trang chủ

**ADMIN-STATISTICAL**

* Folder **statistical**(statistical.component.html, statistical.component.ts)
* Tìm kiếm theo ngày tương tự như ADMIN-TRANSACTION
* Sử dụng **GOOGLE CHART** để vẽ biểu đồ hình tròn thống kê doanh thu của shop