**Thành viên nhóm :**

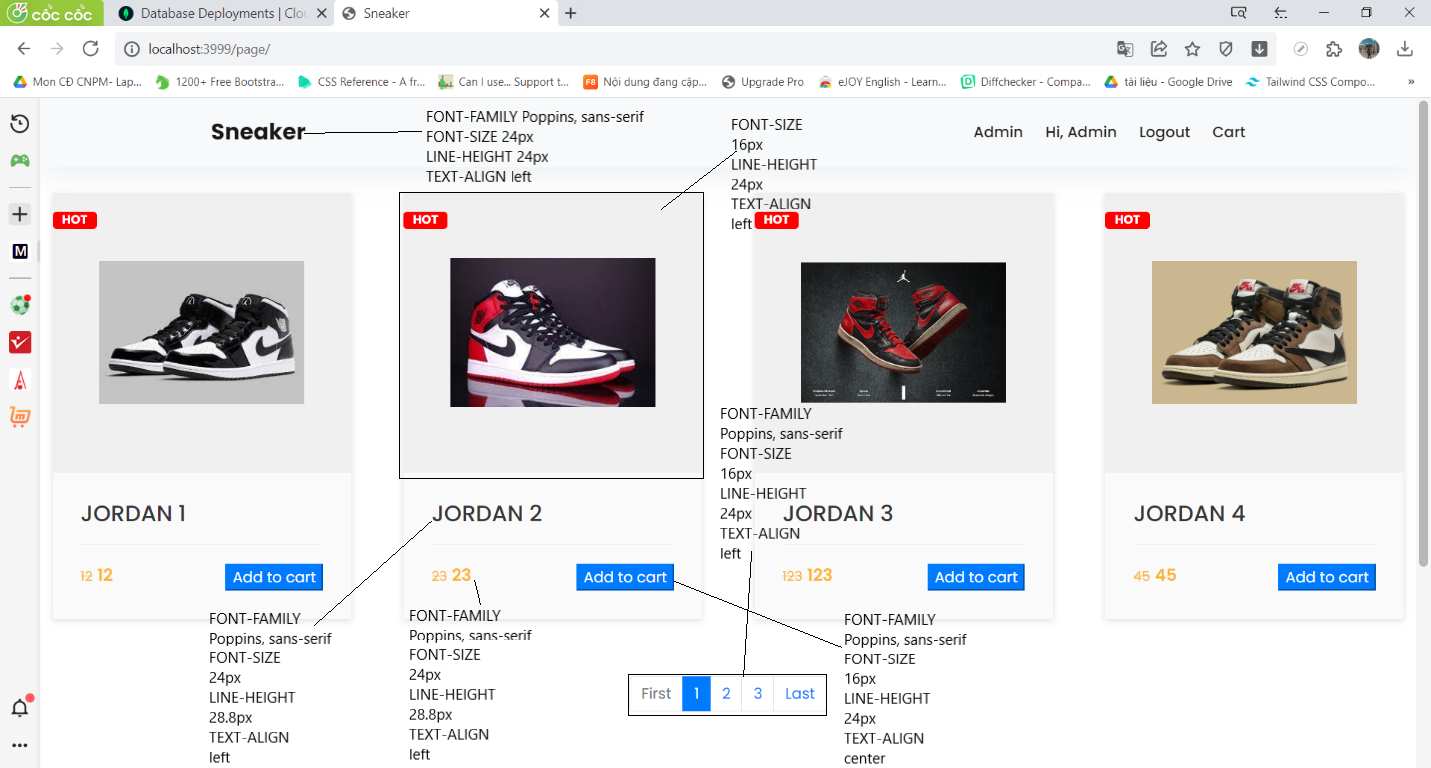
|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Lớp** |
| **Nguyễn Văn Tấn Phát** | **1911062071** | **19DTHA3** |
| **Phạm Hiệp Thành** | **1911061440** | **19DTHA3** |

**US01-Phân Trang - Là một admin tôi muốn phân trang cho trang sản phẩm**

**Mô tả**

* What: Phân trang cho sản phẩm
* Who: Admin
* Why: Để phân trang sản phẩm dễ quản lý và xem sản phẩm dễ dàng hơn

**Mô tả giao diện:**

* 

**Lập trình FrontEnd**

* Hiện thực hoá giao diện như trong file đính kèm
* Phải Login tài khoản thành thì mới bấm được Page trên navbar để điều hướng trang đến trang Page
* Và chọn trang 1 đến trang cuối để kiểm tra 4 sản phẩm trên trang có hoạt động giống với CSDL
* Bấm nút Page ở navbar kết quả trả về là kết quả truy suất CSDL thì hiển thị sản phẩm lên màn hình

**Lập trình BackEnd**

* Sử dụng req.params đó là lấy thông số các page.
* app.get('/news/:page', (req, res, next) => {} ) thì page được gọi là params ngoài các này các bạn
* cũng có thể sử dụng req.query để truy vấn vào các trang được pagination.
* Thay vì res.send để trả về dữ liệu theo dạng json thì bây giờ chúng ta sẽ thay đổi thành
* res.render để trả về dữ liệu dưới dạng html
* Hiển thị phân trang nếu có trang
* Nếu thông số http://localhost:3999/page/1/ là 1 thì First sẽ bị disabled
* Nếu thông số http://localhost:3999/page/2/ từ 2 trở lên thì First sẽ bỏ disabled
* Toán tử điều kiện 3 ngôi syntax: < điều\_kiện ? giá\_trị\_1 : giá\_trị\_2 >
* Nếu điều\_kiện trả về true, toán tử có giá trị giá\_trị\_1. Ngược lại toán tử có giá trị giá\_trị\_2.
* Sử dụng vòng lặp để lặp qua từng trang và tăng số trang
* Trong vòng lặp for có ba biểu thức tùy chọn nên chỉ cần các bạn tách mỗi biểu thức bằng dấu
* “ ; ”
* Nếu số current( chỉ số phân trang ) là 4 và tổng số page là 4 thì Last bị disabled
* Ngược lại, thì Last là trang cuối cùng pages ở đây là 4 đồng nghĩa với Last là trang số 4

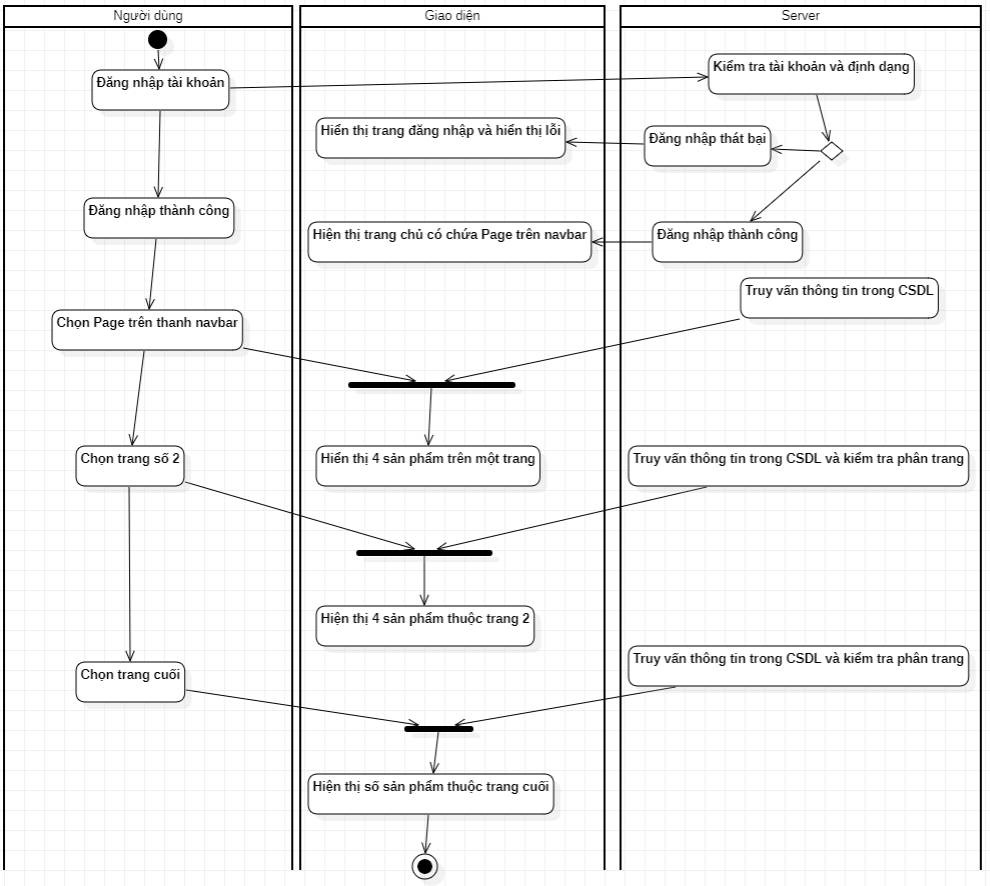
**Kiểm thử**

* Phải Login tài khoản thành công thì mới hiện ra navbar chứa Page
* Người dùng bấm Page ở trên thanh navbar
* Truy vấn thông tin trong CSDL để hiện thị 4 sản phẩm trên một trang thì chuyển hướng qua
* Page và xử dụng các chức năng
* Chọn số trang 2 để kiểm tra 4 sản phẩm có được phân trang
* Và chọn đến trang cuối để kiểm tra 4 sản phẩm trên trang có hoạt động giống với CSDL

**Luồng hoạt động**

* Phải Login tài khoản
* Người dùng bấm Page ở trên thanh navbar
* Truy vấn thông tin trong CSDL để hiện thị 4 sản phẩm trên một trang thì chuyển hướng qua
* Page và xử dụng các chức năng
* Chọn số trang 2 để kiểm tra 4 sản phẩm có được phân trang
* Và chọn đến trang cuối để kiểm tra 4 sản phẩm trên trang có hoạt động giống với CSDL.

**Activity phân trang:**

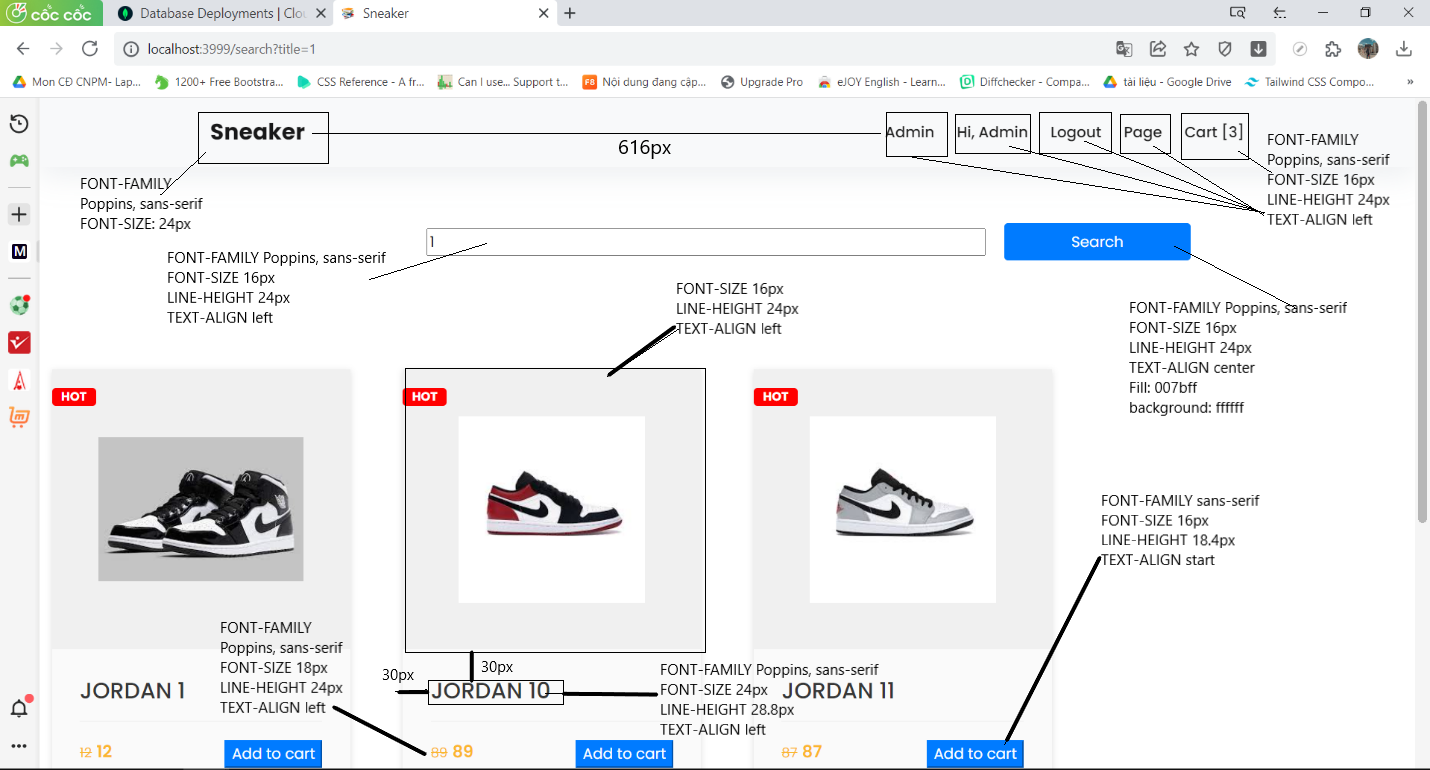
****

**US02-Tìm Kiếm-Là một admin tôi muốn tìm kiếm sản phẩm**

**Mô tả**

* What: Tìm kiếm sản phẩm
* Who: Admin
* Why: Để tìm kiếm sản phẩm thuận tiện hơn

**Mô tả giao diện:**



**Lập trình FrontEnd**

* Khi bấm nút Search nếu textbox Search trống thì hiển thị thông báo vui lòng nhập thông tin
* trường này
* Hiện thực hoá giao diện như trong file đính kèm
* Bấm nút Search nếu kết quả trả về dựa vào biểu thức chính quy của title sản phẩm

**Lập trình BackEnd**

* Viết api thực hiện kiểm tra đã đăng nhập chưa
* Tìm kiếm sản phẩm và xử lý tìm kiếm sản phẩm sử dụng biểu thức chính quy để tìm kiếm title
* trong sản phẩm ở CSDL
* Nếu có title giống với CSDL thì in tất cả sản phẩm giống title ra màn hình

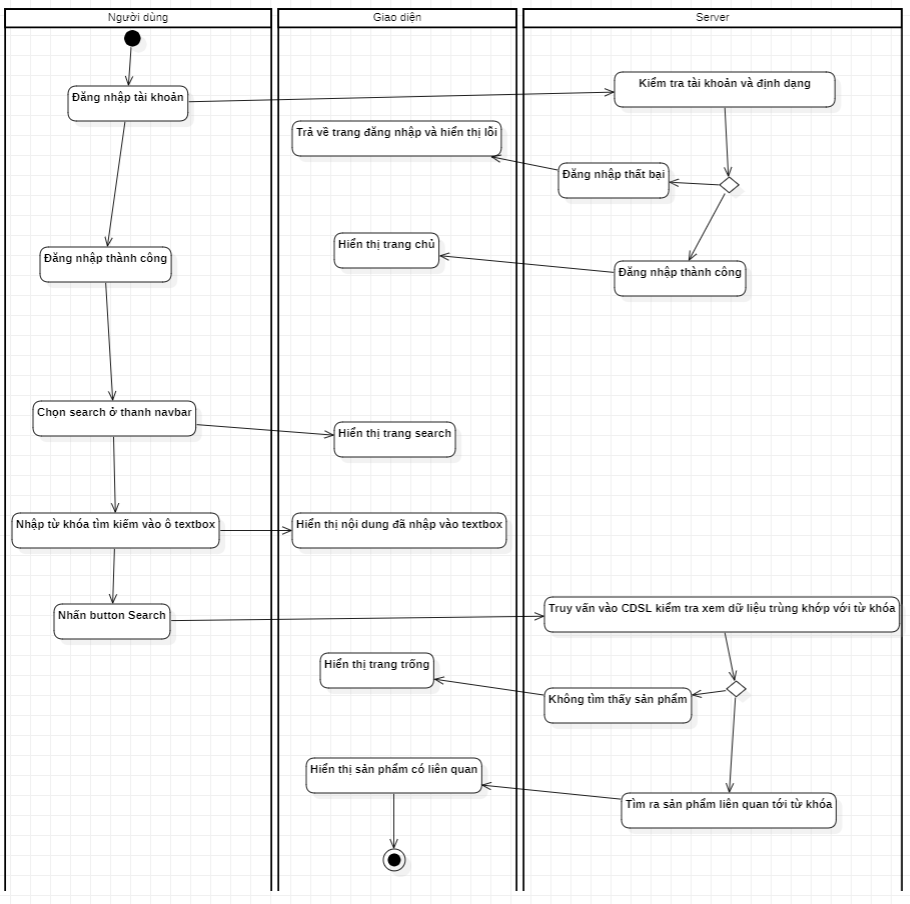
**Kiểm thử**

* Search theo toán tử or hoặc not (search thông tin với riêng từng giá trị hoặc phủ định của giá
* trị trong đó)
* Search theo toán tử and (tạo ra các bộ tương ứng để thực hiện search)
* Phân biệt chữ hoa chữ thường, tiếng việt, tiếng anh, .. Khi thực hiện test

**Luồng hoạt động**

* Người dùng đăng nhập
* Chọn Search ở thanh navbar
* Nhập từ khoá search vào ô textbox search rồi ấn button search

**Activity tìm kiếm:**

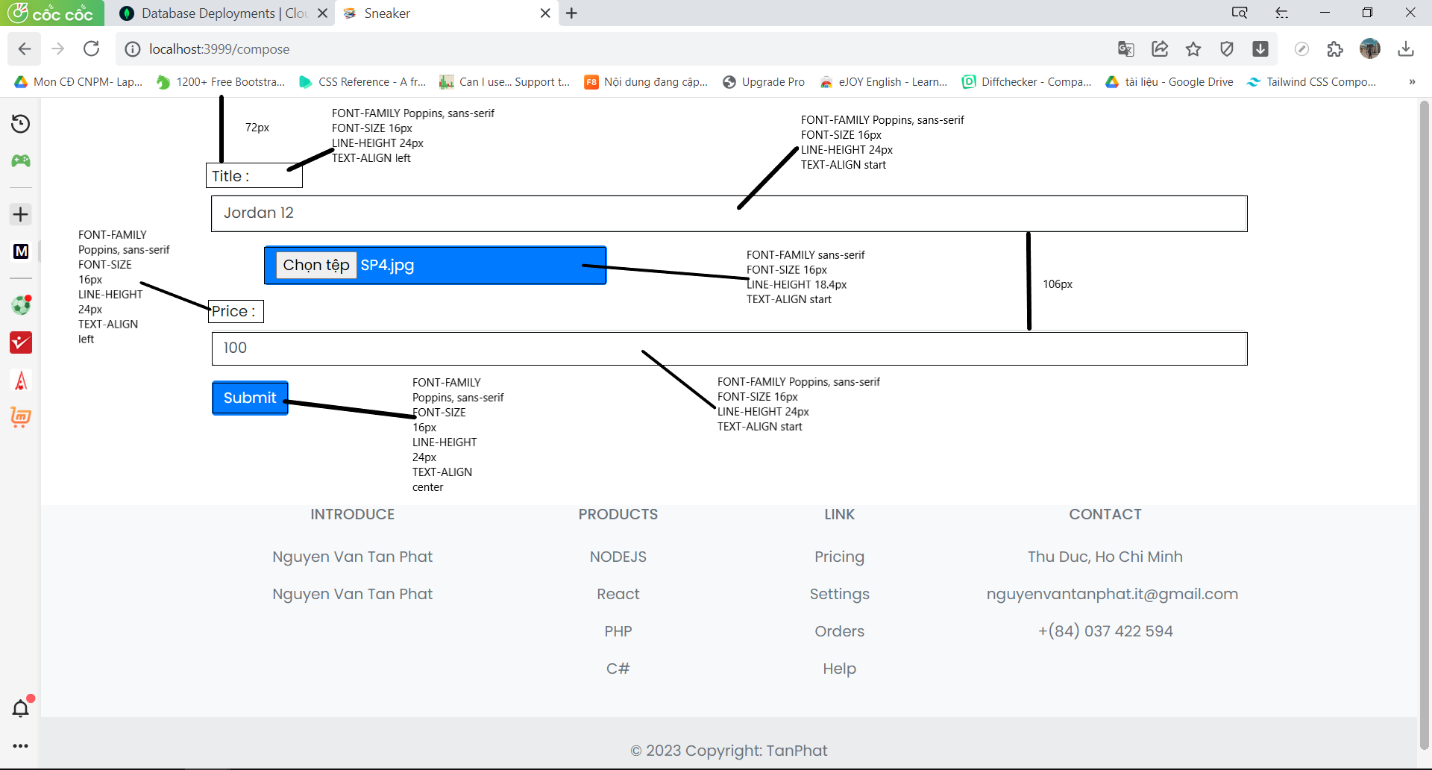


**US03-Thêm sản phẩm-Là một admin tôi muốn thêm sản phẩm**

**Mô tả**

* What: Thêm sản phẩm vào CSDL để bán hàng
* Who: Admin
* Why: Thêm sản phẩm để bán giày cho khách hàng

**Mô tả giao diện:**



**Lập trình FrontEnd**

* Hiện thực hoá giao diện như trong file đính kèm
* Khi bấm nút Submit nếu textbox title trống, file ảnh trống, price trống thì hiển thị thông báo vui
* lòng điền thông tin trường này
* Khi bấm nút Submit nếu textbox title trống hoặc file ảnh trống hoặc price trống thì hiển thị
* thông báo vui lòng điền thông tin trường này
* Khi bấm nút Submit nếu file ảnh không đúng định dạng hổ trỡ thì hiển thị thông báo vui lòng
* chọn file ảnh đúng
* Bấm nút Submit nếu các trường hợp lệ kết quả trả về điều hướng đến trang chủ

**Lập trình BackEnd**

* Viết api thực hiện kiểm tra nếu title hoặc file ảnh hoặc price để trống thì trả về thông báo vui
* lòng điền thông tin trường này
* Viết api thực hiện kiểm file ảnh đúng định dạng chưa nếu chưa thì trả về thông báo vui lòng
* điền thông tin trường này
* Viết api thực hiện thêm sản phẩm thành công và lưu thông tin sản phẩm vào CSDL

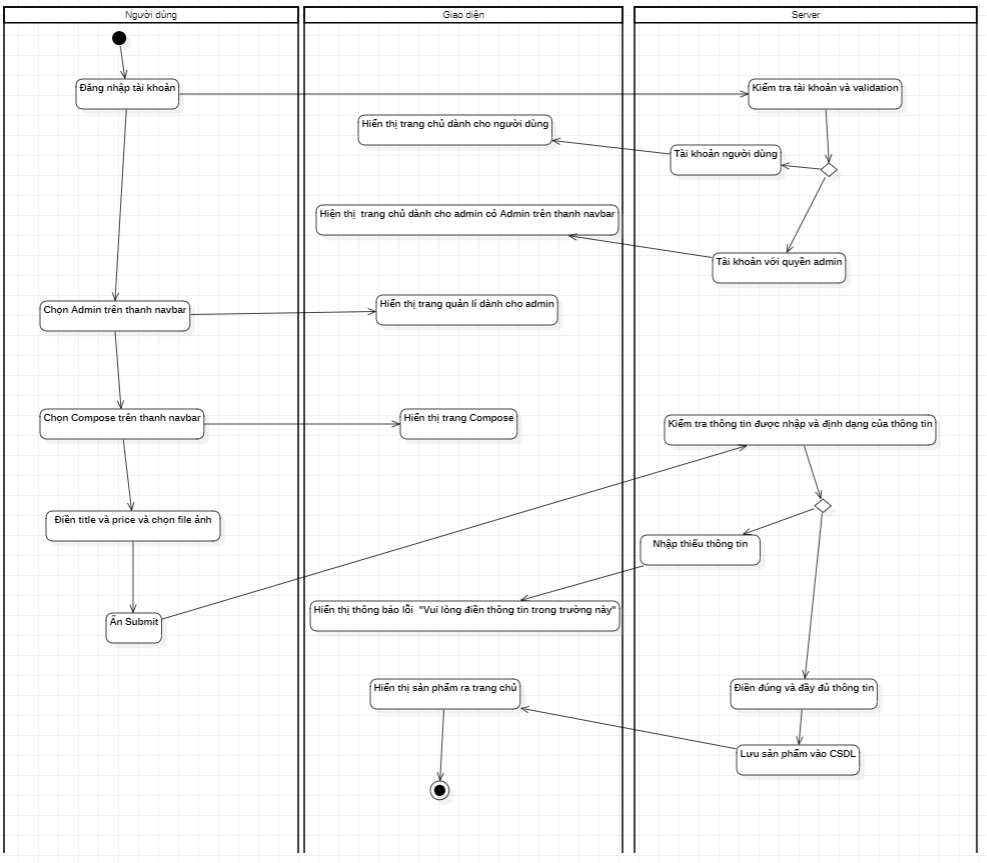
**Kiểm thử**

* Kiểm tra nếu thiếu title thì có xuất hiện thông báo vui lòng điền thông tin trường này
* Kiểm tra nếu thiếu file ảnh thì có xuất hiện thông báo vui lòng điền thông tin trường này
* Kiểm tra nếu file ảnh không đúng định dạng hỗ trợ thì có xuất hiện thông báo vui lòng điền
* thông tin trường này
* Kiểm tra nếu thiếu pricethì có xuất hiện thông báo vui lòng điền thông tin trường này

**Luồng hoạt động**

* Đăng nhập với quyền là admin
* Chọn admin ở trên thanh navbar
* Chọn tiếp Compose ở trên thanh navbar
* Admin điền title và price và chọn file ảnh vào form thêm sản phẩm
* Admin bấm nút Login
* Kiểm tra nếu thiếu title thì xuất hiện thông báo vui lòng điền thông tin trường này
* Kiểm tra nếu thiếu price thì xuất hiện thông báo vui lòng điền thông tin trường này
* Kiểm tra nếu thiếu file ảnh thì xuất hiện thông báo vui lòng điền thông tin trường này
* Nếu nhập đủ 3 trường đúng định dạng thì thêm vào CSDL và hiện sản phẩm ra trang chủ

**Acyivity thêm sản phẩm:**

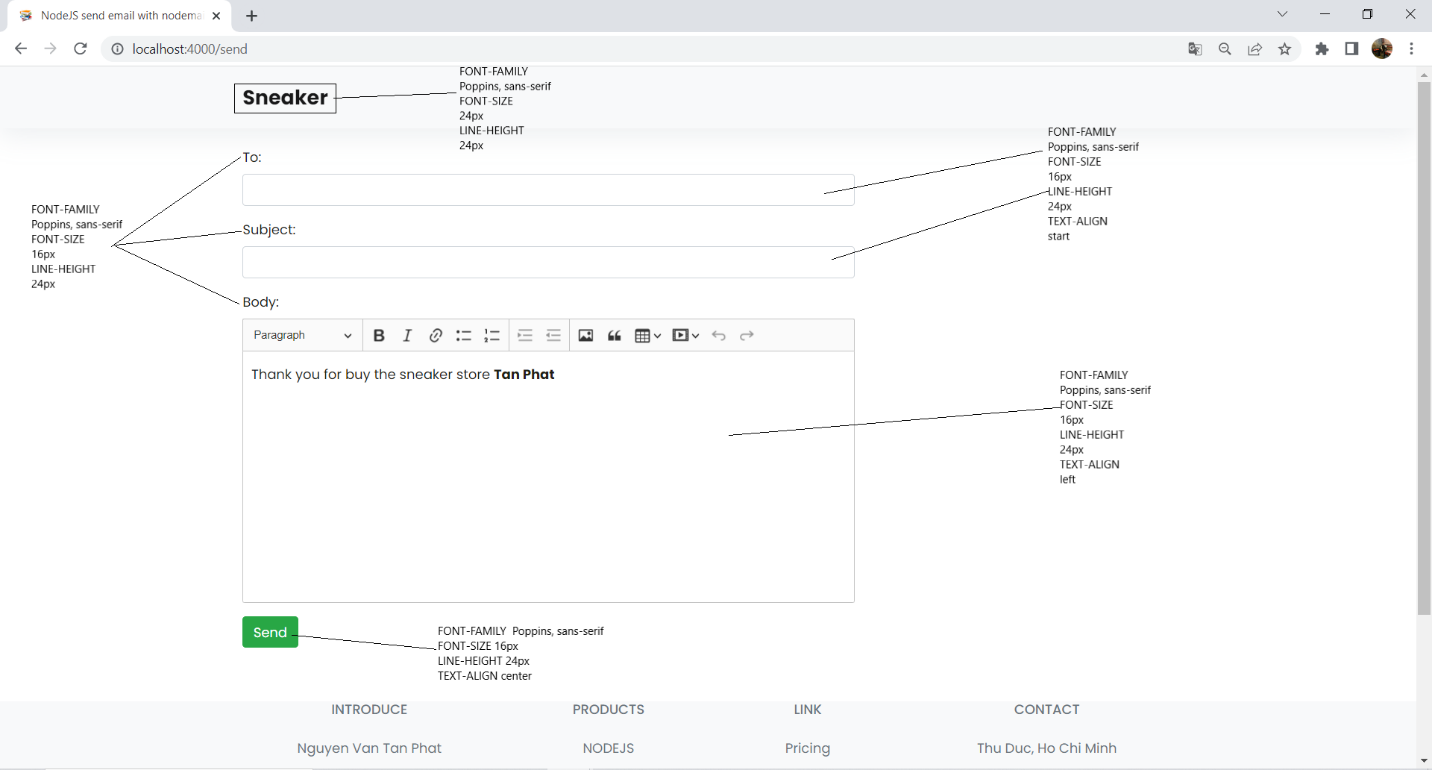
****

**US04-Gửi Mail-Là một admin tôi muốn gửi mail cho khách hàng**

**Mô tả**

* What: Gửi gmail cho khách hàng
* Who: Admin
* Why: Gửi gmail để trao đổi, tư vấn về sản phẩm giá cả hay phản hồi từ khách hàng

**Mô tả giao diện:**

****

**Lập trình FrontEnd**

* Hiện thực hoá giao diện như trong file đính kèm
* Kiểm tra nếu thiếu To thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Kiểm tra nếu thiếu Subject thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Kiểm tra nếu thiếu Body thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Bấm nút send gửi gmail thành công thì điều hướng về trang chủ

**Lập trình BackEnd**

* Trong file mailer.js này mình sẽ sử dụng thư viện nodemailer để khởi tạo một thằng transporter với những cấu hình cụ thể trong function sendMail và sau đó sử dụng nó để gửi email đến điểm cuối mà chúng ta mong muốn.
* File mailer.js này lát nữa chúng ta sẽ sử dụng ở emailController.js sau khi nhận được request gửi
* email từ form phía client.
* Khởi tạo một thằng transporter object sử dụng chuẩn giao thức truyền tải SMTP với các thông
* tin cấu hình ở trên.
* Hàm transporter.sendMail() này sẽ trả về cho chúng ta một Promise
* File homeController.js có một function getHome trả về cái màn hình cho chúng ta nhập thông tin cho email, chính là sử dụng module path để lấy ra file giao diện.
* Trong emailController này chúng ta sẽ import module mailer vừa viết ở trên kia, sau đó kết hợp
* cùng với những thông tin request từ phía client trong req.body, cuối cùng là gọi hành động gửi email. Nếu có lỗi hoặc thành công trong quá trình thì đều sẽ gửi về cho client biết.
* Trong file app.js là nơi chúng ta cấu hình routers cho ứng dụng, sẽ có 2 routes chính, một là để gọi ra màn hình nhập thông tin email để gửi đi, có method là GET, route còn lại là gọi đến emailController ở ngay phía trên để thực hiện hành động gửi email, method là POST.

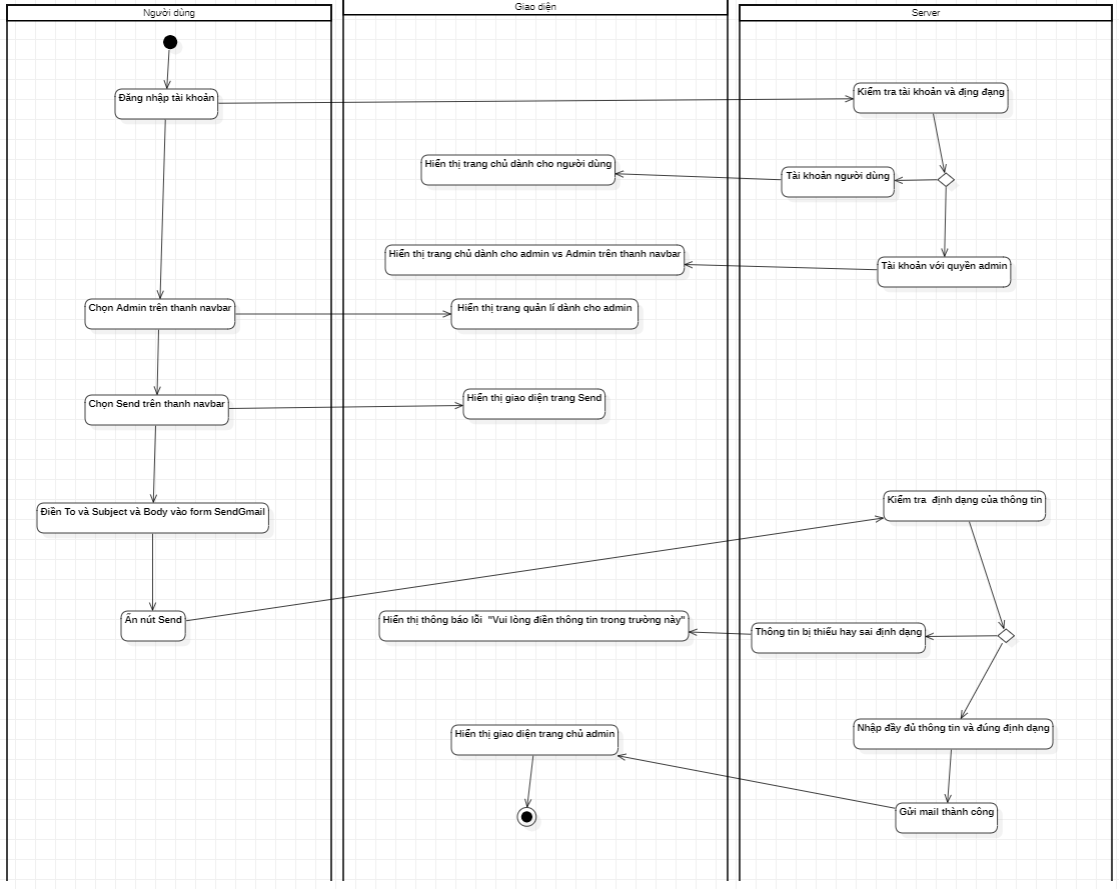
**Kiểm thử**

* Gửi email mẫu nên sử dụng CSS tiêu chuẩn cho tất cả các email .
* Địa chỉ Email phải được xác nhận trước khi gửi email.
* Ký tự đặc biệt trong trường email nên được xử lý đúng cách.
* Ký tự ngôn ngữ cụ thể (ví dụ như ký tự ngôn ngữ Nga, Trung, Đức) phải được xử lý đúng cách
* trong trường email.
* Tiêu đề email không nên để trống.
* Trường placehoder được sử dụng trong mẫu email phải thay thế bằng giá trị thực tế ví dụ như
* {FirstName} {Lastname} nên được thay thế với các cá thể đầu tiên và cuối cùng tên đúng cho tất
* cả người nhận.
* Nếu báo cáo với các giá trị động mới có trong nội dung email, dữ liệu báo cáo phải được tính
* toán một cách chính xác.
* Email tên người gửi không nên được trống.
* Email nên được kiểm tra trong các khách hàng email khác như Outlook, Gmail, Hotmail, Yahoo!
* Mail, …
* Kiểm tra chức năng gửi email sử dụng TO, CC và BCC.
* Kiểm tra văn bản email thô.
* Kiểm tra HTML định dạng email.
* Kiểm tra tiêu đề email và footer cho logo của công ty, chính sách bảo mật và các liên kết khác.
* Kiểm tra email với file đính kèm.
* Kiểm tra chức năng gửi email tới đơn, đa hoặc phân phối người nhận danh sách.
* Kiểm tra nếu trả lời vào địa chỉ email là chính xác.
* Kiểm tra gửi khối lượng lớn các email.

**Luồng hoạt động**

* Login với tài khoản admin
* Chọn Admin ở thanh navbar
* Chọn Send ở thanh navbar
* Người dùng điền To và Subject và Body vào form SendGmail
* Người dùng bấm nút Send
* Kiểm tra nếu thiếu To thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Kiểm tra nếu thiếu Subject thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Kiểm tra nếu thiếu Body thì xuất hiện thông báo vui lòng nhập thông tin trường này
* Nếu nhập đủ 3 trường đúng định dạng thì gửi gmail thành công

**Activity Gửi Mail:**

****