

**ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Họ và tên sinh viên: Nguyễn Bảo Tân**

## **BÁO CÁO**

# **THỰC TẬP TỐT NGHIỆP**

**Công ty thực tập : CÔNG TY TNHH MTV HỆ THỐNG  
APPVITY VIỆT NAM**

**Chuyên gia hướng dẫn : Trần Đức Vinh**

**Giảng viên hướng dẫn : TS. Cao Thái Phương Thanh**

***TP. Hồ Chí Minh, tháng 11 năm 2024***

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

Họ và tên sinh viên: Nguyễn Bảo Tân

Mã số sinh viên: 3120410464

**BÁO CÁO**  
**THỰC TẬP TỐT NGHIỆP**

Công ty thực tập : CÔNG TY TNHH MTV HỆ THỐNG  
APPVITY VIỆT NAM

Chuyên gia hướng dẫn : Trần Đức Vinh

Giảng viên hướng dẫn : TS. Cao Thái Phương Thanh

*TP. Hồ Chí Minh, tháng 11 năm 2024*

## MỤC LỤC

DANH MỤC HÌNH ẢNH.....	iii
NHẬN XÉT CỦA CHUYÊN GIA DOANH NGHIỆP .....	v
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....	vi
LỜI MỞ ĐẦU .....	1
CHƯƠNG 1: GIỚI THIỆU .....	3
1.1.    Giới thiệu về công ty thực tập. ....	3
1.1.1.    Thông tin chung của APPVITY .....	3
1.1.2.    Sơ lược về sự hình thành và phát triển của doanh nghiệp.....	3
1.1.3.    Cơ sở vật chất của công ty .....	4
1.1.4.    Sản phẩm được triển khai bằng Trusted Technologies Worldwide .....	4
1.1.5.    Lĩnh Vực Hoạt Động Của Doanh Nghiệp.....	6
1.1.6.    Các sản phẩm quan trọng .....	7
1.1.7.    Một số đối tác của doanh nghiệp.....	10
1.1.8.    Vị trí thực tập .....	10
1.2.    Nhiệm vụ thực tập.....	10
Tổng kết chương 1. ....	12
CHƯƠNG 2: NỘI DUNG THỰC TẬP .....	13
2.1.    Tìm hiểu và cài đặt môi trường, các tool và các framework công ty sử dụng.....	13
2.1.1.    Sơ lược về TypeScript.....	13
2.1.2.    Sơ lược về SCSS .....	17
2.1.3.    Sơ lược về Tailwind CSS .....	18
2.1.4.    Sơ lược về Javascript.....	20
2.1.5.    Cài đặt và dựng source .....	22
2.1.6.    Sơ lược về Angular.....	22

2.2.	Thực hành code website THEVOUX .....	26
2.2.1.	Tìm hiểu source FE .....	26
2.2.2.	Tạo trang Home .....	27
2.2.3.	Tạo chức năng admin gửi dữ liệu về Trending News lên firebase .....	28
2.2.4.	Tạo chức năng hiển thị Trending News trên trang Home .....	34
2.2.5.	Trang admin đẩy products lên firebase .....	36
2.2.6.	Hiển thị trang shopping .....	41
2.2.7.	Hiển thị Product theo Category .....	42
2.2.8.	Hiển thị Product Detail của Product .....	45
2.2.9.	Chức năng addToCart từ product, product details .....	47
2.2.10.	Hiển thị product vào trang cart .....	49
2.2.11.	Giao diện trang checkout .....	52
	Tổng kết chương 2 .....	53
	CHƯƠNG 3: KẾT QUẢ THỰC TẬP .....	54
	Tổng kết chương 3 .....	63
	CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ .....	64
	TÀI LIỆU THAM KHẢO .....	65

# DANH MỤC HÌNH ẢNH

Hình 1. 1 Logo công ty .....	3
Hình 1. 2 Microsoft SharePoint.....	4
Hình 1. 3 Azure App Service .....	5
Hình 1. 4 MEAN Stack .....	5
Hình 1. 5 .NET Core.....	5
Hình 1. 6 Microsoft Power Automate .....	5
Hình 1. 7 Nintex Workflow.....	6
Hình 1. 8 Chu kì phát triển (Development Cycle ) .....	9
Hình 1. 9 Appvity eProducts Architecture .....	9
Hình 1. 10 Các tác doanh nghiệp của công ty .....	10
Hình 2. 1 Logo Typescript.....	13
Hình 2. 2 Logo Tailwind CSS .....	18
Hình 2. 3 Logo Framework Angular .....	23
Hình 2. 4 Cấu trúc typescript trong dự án Angular .....	27
Hình 2. 5 Giao diện trang Home .....	27
Hình 2. 6 Giao diện trang Beauty .....	28
Hình 2. 7 Giao diện dashboard nút button Add New Post .....	29
Hình 2. 8 Giao diện form điền add new post.....	29
Hình 2. 9 Giao diện table Trending News của dashboard admin.....	30
Hình 2. 10 Giao diện của thành phần Trending News ở trang home .....	34
Hình 2. 11 Giao diện dashboard có nút add product .....	36
Hình 2. 12 Form điền product .....	36
Hình 2. 13 Giao diện table product của trang dashboard admin .....	37
Hình 2. 14 Giao diện trang shopping của users.....	41
Hình 2. 15 Giao diện trang shoes của shopping .....	43
Hình 2. 16 Giao diện trang Accessories của shopping .....	43
Hình 2. 17 Giao diện trang product details.....	45
Hình 2. 18 Giao diện trang men shirts có nút add to cart.....	47
Hình 2. 19 Giao diện trang product details có hiển thị nút add to cart .....	48

Hình 2. 20 Giao diện khi nhấn nút add to cart thì hiển thị thông báo .....	48
Hình 2. 21 Giao diện trang cart .....	52
Hình 2. 22 Giao diện trang checkout.....	52

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

(Ký tên và đóng dấu)

(Ký tên và đóng dấu)

[illegible]

(Ký tên và đóng dấu)



# LỜI MỞ ĐẦU

Ngày nay, hầu như tất cả mọi lĩnh vực đều cần đến công nghệ thông tin để phát triển và thuận tiện về nhiều mặt. Từ trường học cho đến các công ty, xí nghiệp đều tuyển dụng các vị trí liên quan đến ngành nghề này.

Công nghệ thông tin là một lĩnh vực không thể thiếu đối với nhiều ngành nghề khác, nhất là trong bối cảnh của ngành công nghiệp 4.0 hiện nay, khi mà công nghệ phần mềm được ứng dụng trong rất nhiều lĩnh vực của đời sống con người.

Là một sinh viên ngành Công nghệ thông tin của trường Đại học Sài Gòn, em hiểu rất rõ tầm quan trọng của ngành nghề mà mình đã chọn và theo học. Ngoài việc học ở trường, em còn mong muốn được trải nghiệm, trau dồi và học hỏi thông qua các dự án thực tế. Thông qua cơ hội thực tập này, bản thân em mong muốn được tiếp cận môi trường doanh nghiệp để hiểu rõ về ngành nghề mà mình đã đang và sẽ theo đuổi, cũng như cách làm việc trong một môi trường chuyên nghiệp là như thế nào. Sẽ là một trải nghiệm tuyệt vời để em thử sức, cọ xát và nhìn nhận khả năng của bản thân một cách khách quan, hiểu rõ hơn về môi trường làm việc chuyên nghiệp của công ty, doanh nghiệp.

Xin chân thành cảm ơn nhà trường, các thầy cô giảng viên đã giảng dạy những kiến thức nền tảng. Đặc biệt, em xin chân thành gửi lời cảm ơn đến thầy Cao Thái Phương Thanh đã hướng dẫn và góp ý cho em trong suốt quá trình thực tập. Một lần nữa em xin gửi lời cảm ơn chân thành đến phía nhà trường cũng như doanh nghiệp là Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam đã cho em cơ hội được thực tập trong một môi trường làm việc chuyên nghiệp. Tuy thời gian thực tập không lâu nhưng công ty và đặc biệt hơn là các anh chị trong công ty đã tạo điều kiện tốt nhất để em cũng như những bạn thực tập sinh khác được phát huy khả năng của mình và phát triển hơn nữa. Sau khi kết thúc đợt thực tập này, em đã tích lũy được rất nhiều kinh nghiệm và rút ra được nhiều bài học, giải đáp được những thắc mắc về làm việc trong một doanh nghiệp và trang bị cho bản thân vốn kiến thức chắc chắn, song song với đó là một tác phong làm việc chuyên nghiệp và hết mình với công việc cho sau này.

Nội dung của cuốn báo cáo thực tập tốt nghiệp này bao gồm:

Chương 1: Giới thiệu về đơn vị thực tập.

Chương 2: Nội dung thực tập

Chương 3: Kết quả thực tập.

Chương 4: Kết luận và kiến nghị

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Giới thiệu về công ty thực tập.

### 1.1.1. Thông tin chung của APPVITY



*Hình 1. 1 Logo công ty*

- Mã số thuế: 0312081276
- Tên công ty : CÔNG TY TNHH MỘT THÀNH VIÊN HỆ THỐNG APPVITY VIỆT NAM
- Tên quốc tế: APPVITY SYSTEM CO.,LTD
- Địa chỉ : Lầu 8, Toà Nhà Kicotrans, Số 46, Đường Bạch Đằng, Phường 2, Quận Tân Bình, Thành phố Hồ Chí Minh, Việt Nam
- Loại hình pháp lý: Công ty TNHH một thành viên
- Ngày cấp: 07/12/2012
- Website : <https://www.appvity.com>
- Điện thoại : 02835474562

### 1.1.2. Sơ lược về sự hình thành và phát triển của doanh nghiệp

Công ty chuyên kinh doanh về lĩnh vực công nghệ thông tin. APPVITY được thành lập từ năm 2012 là công ty chuyên về lĩnh vực phát triển phần mềm với đội ngũ nhân sự chất lượng cao, chuyên nghiệp và nhiều năm kinh nghiệm trong lĩnh vực phần mềm. Với tầm nhìn trở thành một trong những công ty hàng đầu trong lĩnh vực công nghệ, Appvity đã không ngừng phát triển và mở rộng quy mô, đồng thời nâng cao chất lượng sản phẩm và dịch vụ của mình.

Appvity bao gồm một nhóm các chuyên gia công nghệ Microsoft mong muốn xây dựng các ứng dụng cho phép các chuyên gia thông tin làm việc hiệu quả khi ở bất kỳ đâu trên thế giới. Chúng tôi xây dựng các sản phẩm tận dụng nền tảng đám mây Azure và Office 365 và được tích hợp tốt với Microsoft Teams. Trao quyền kết nối trong doanh nghiệp và cộng tác giữa các tổ chức.

Tại Appvity, chúng tôi coi trọng sự cộng tác là nhanh chóng, rõ ràng và đơn giản. Các sản phẩm của chúng tôi được xây dựng dựa trên các nguyên tắc cải thiện độ tin cậy của dữ liệu và xử lý tự động hóa để tăng cường kết nối giữa mọi người và tiết kiệm thời gian. Appvity eProducts hoạt động trên nhiều thiết bị để đảm bảo người dùng có thể theo dõi công việc của mình mọi lúc, mọi nơi.

### **1.1.3. Cơ sở vật chất của công ty**

Văn phòng được thiết kế và bố trí chuyên nghiệp, thân thiện, với đầy đủ các thiết bị cần thiết hỗ trợ công việc hàng ngày của nhân viên, bao gồm cả máy pha cà phê và lò vi sóng. Không gian làm việc thoáng mát, ánh sáng đầy đủ mang lại cảm giác dễ chịu, nâng cao năng suất làm việc. Nhân viên không bị gò bó về thời gian và hiếm khi phải làm việc ngoài giờ. Môi trường làm việc luôn được giữ sạch sẽ nhờ vệ sinh hàng ngày.

### **1.1.4. Sản phẩm được triển khai bằng Trusted Technologies Worldwide**



*Hình 1. 2 Microsoft SharePoint*

Chúng tôi triển khai và hỗ trợ Sản phẩm và Công nghệ Microsoft SharePoint trong tổ chức của bạn, cung cấp các giải pháp cộng tác có khả năng mở rộng cao. Các giải pháp của chúng tôi được xây dựng trên SharePoint Framework, PnP, Microsoft Graph và SharePoint Add-ins. Chúng tôi cung cấp hỗ trợ cho cả hai phiên bản SharePoint, bao gồm SharePoint tại chỗ và SharePoint trong Microsoft 365.



*Hình 1. 3 Azure App Service*

Xây dựng ứng dụng mạnh mẽ bằng nền tảng Microsoft Azure. Các dịch vụ Nền tảng dưới dạng Dịch vụ (PaaS) của Azure cung cấp các dịch vụ được quản lý hoàn toàn cần thiết để hỗ trợ ứng dụng của bạn bao gồm VM, Dịch vụ ứng dụng, Chức năng... Azure cung cấp độ tin cậy cần thiết cho các ứng dụng có tính khả dụng cao, thậm chí bao gồm cả chuyển đổi dự phòng giữa các vùng khác nhau.



*Hình 1. 4 MEAN Stack*

MEAN, một giải pháp mã nguồn mở, miễn phí, toàn diện cho các ứng dụng MEAN. MEAN kết hợp MongoDB, Express.js, AngularJS và Node.js thành một giải pháp toàn diện duy nhất để phát triển JavaScript.



*Hình 1. 5 .NET Core*

.NET Framework is a development platform developed by Microsoft for building apps for web, Windows, Windows Phone, Windows Server, and Microsoft Azure.



*Hình 1. 6 Microsoft Power Automate*

Power Platform bao gồm bốn sản phẩm chính: Power Apps, Power Automate, Power BI và Power Virtual Agents. Power Platform cho phép doanh nghiệp của bạn tạo ra các giải pháp đồng thời trao quyền cho bạn để kết hợp công nghệ tùy chỉnh nhằm giúp mọi người, từ CEO đến nhân viên tuyến đầu, thúc đẩy doanh nghiệp bằng dữ liệu.



*Hình 1. 7 Nintex Workflow*

Nền tảng quy trình tự động hóa hàng đầu thế giới. Nintex tích hợp với các hệ sinh thái lớn nhất thế giới bao gồm Microsoft SharePoint, Office 365, K2... Nintex giúp quản lý, tự động hóa và tối ưu hóa quy trình kinh doanh của bạn nhanh chóng và dễ dàng. Chúng tôi là đối tác được chứng nhận của nhà cung cấp giải pháp Nintex.

#### **1.1.5. Lĩnh Vực Hoạt Động Của Doanh Nghiệp**

Appvity chuyên về tư vấn toàn chu kỳ từ phân tích kinh doanh, thiết kế, phát triển và hỗ trợ. Dịch vụ phát triển của chúng tôi tập trung vào nền tảng Office 365, Azure và SharePoint. Chúng tôi đã phát triển quy trình làm việc tự động hóa liên kết dữ liệu, tài liệu và quy trình kinh doanh của bạn, chạy trên cả đám mây và tại chỗ. Ngoài ra, chúng tôi cũng cung cấp đào tạo, chuyển giao kiến thức và tài liệu -- như một phần mở rộng cho bộ phận CNTT nội bộ của bạn.

Lĩnh vực hoạt động của APPVITY :

Lập trình vi tính (CPC 842) bao gồm các hoạt động liên quan đến sản xuất, nghiên cứu và phát triển phần mềm máy tính. Đây là một trong những lĩnh vực chính của ngành công nghệ thông tin, đóng vai trò quan trọng trong việc xây dựng các ứng dụng và hệ thống thông tin hiện đại. Cụ thể :

- Sản xuất phần mềm máy tính: Bao gồm quá trình lập trình và phát triển phần mềm ứng dụng và hệ thống, từ các ứng dụng di động đơn giản đến các hệ thống quản lý doanh nghiệp phức tạp.

- Nghiên cứu và phát triển: Bao gồm hoạt động nghiên cứu để khám phá và áp dụng các công nghệ mới, các phương pháp tiên tiến và giải pháp sáng tạo vào phát triển phần mềm. Đây là yếu tố quan trọng giúp nâng cao tính cạnh tranh và sự khác biệt của các sản phẩm phần mềm.
- Công nghệ mới và xu hướng: Lĩnh vực này liên tục tiếp nhận các công nghệ mới như trí tuệ nhân tạo (AI), học máy (Machine Learning), blockchain, IoT (Internet of Things), và cloud computing để áp dụng vào các dự án phát triển phần mềm.
- Ứng dụng rộng rãi: Các sản phẩm và dịch vụ phát triển từ lĩnh vực này có thể áp dụng trong nhiều lĩnh vực khác nhau như y tế, giáo dục, tài chính, du lịch, và sản xuất.
- Lập trình máy vi tính không chỉ đóng vai trò quan trọng trong việc cung cấp các giải pháp công nghệ hiện đại mà còn mở ra những cơ hội nghề nghiệp hấp dẫn cho các nhà phát triển phần mềm và chuyên gia công nghệ thông tin.

APPVITY luôn tạo ra một môi trường năng động, sáng tạo và phát triển toàn diện, vượt trội cho từng thành viên. Luôn đem đến cho các thành viên những giá trị về vật chất cũng nhưng tinh thần tương ứng với những nỗ lực đóng góp của bản thân.

#### **1.1.6. Các sản phẩm quan trọng**

- eFiles
- eXsync
- eRequest
- eTask
- eHelpdesk

#### **Chu kì phát triển (Development Cycle )**

##### *Discovery (khám phá)*

Phân tích và chuyển đổi các yêu cầu kinh doanh của bạn thành tài liệu kỹ thuật để đánh giá chi tiết.

##### *Design (thiết kế)*

Đơn giản hóa tài liệu kỹ thuật dưới dạng wireframe (khung thiết kế giao diện) và mô hình kiến trúc để đảm bảo hệ thống đáp ứng đầy đủ nhu cầu của doanh nghiệp ở mọi khía cạnh.

#### *Development (phát triển)*

Phát triển hệ thống một cách nghiêm ngặt, đảm bảo tuân thủ theo thiết kế và đáp ứng các yêu cầu kinh doanh.

#### *Quality Assurance (đảm bảo chất lượng)*

Chuẩn bị kế hoạch kiểm thử, các trường hợp kiểm thử, và thực hiện kiểm thử hệ thống theo mô hình lặp để loại bỏ rủi ro.

#### *Deployment (triển khai)*

Thực hiện quy trình triển khai qua các môi trường:

- Môi trường phát triển.
- Môi trường thử nghiệm [UAT] (User Acceptance Testing).
- Triển khai môi trường sản xuất với quy trình kiểm soát mã nguồn an toàn.

#### *Training & Support (đào tạo và hỗ trợ)*

Thực hiện đào tạo trực tiếp khi hệ thống sẵn sàng hoạt động. Hỗ trợ thông qua email hoặc hệ thống quản lý yêu cầu (ticketing system).





Hình 1. 8 Chu kỳ phát triển (Development Cycle )



Hình 1. 9 Appvity eProducts Architecture

### 1.1.7. Một số đối tác của doanh nghiệp



*Hình 1. 10 Các tác doanh nghiệp của công ty*

- Ngân hàng thế giới
- Công ty dầu khí quốc tế PremierOil
- Công ty bảo hiểm nhân thọ Prudential
- Công ty cổ phần hàng không VIETJET
- Ngân hàng Á Châu - ACB
- Tập đoàn Vingroup
- Công ty cổ phần sữa Việt Nam
- Công ty TNHH MTV Dịch vụ mặt đất sân bay Việt Nam

### 1.1.8. Vị trí thực tập

Vị trí thực tập trong công ty: Thực tập sinh Frontend Developer

## 1.2. Nhiệm vụ thực tập.

**Nội dung chính:** Làm quen với môi trường làm việc thực tế của ngành. Tìm hiểu về cách công cụ, quy trình làm việc được sử dụng trong công ty. Tham gia vào việc phát triển một dự án trong công ty.

#### **Kiến thức chung:**

- Công cụ làm việc cho các dự án
- Kỹ năng giao tiếp, trình bày ý tưởng
- Kỹ năng làm việc nhóm
- Kỹ năng quản lý thời gian
- Kỹ năng tìm tài liệu, giải quyết vấn đề

#### **Kiến thức chuyên môn:**

- Kiến thức về Javascript ES6
- Kiến thức về Tailwind CSS
- Kiến thức về TypeScript, Angular và các thư viện liên quan
- Kiến thức về quản lý phiên bản phần mềm
- Cách sử dụng Git trong dự án thực tế
- Kiến thức về Database

## **Tổng kết chương 1.**

Trong quá trình thực tập tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam, em đã có cơ hội được giới thiệu và làm quen với môi trường làm việc chuyên nghiệp của một công ty phát triển phần mềm. Phần “Giới thiệu công ty thực tập” đã giúp em tìm hiểu rõ hơn về môi trường, văn hoá làm việc, lĩnh vực hoạt động, cơ cấu tổ chức và đối tác của công ty. Appvity thực sự là một nơi để em tích lũy kinh nghiệm đối với một sinh viên còn chưa vững kiến thức thực tế như em và cũng là nơi để em rèn luyện các kỹ năng kỹ thuật và nghiệp vụ cho các quá trình phát triển phần mềm.

Phần “Nhiệm vụ thực tập” đã đề cập đến các công việc cụ thể mà em đã được giao. Trong quá trình thực tập, em đã có cơ hội làm quen với môi trường làm việc thực tế của ngành công nghệ thông tin, nơi em đã học hỏi và nắm bắt các công cụ cũng như quy trình làm việc được sử dụng trong công ty. Em đã tham gia vào phát triển một dự án cụ thể, từ đó củng cố kiến thức về các công cụ, rèn luyện kỹ năng giao tiếp, làm việc nhóm, quản lý thời gian, cũng như khả năng tìm tài liệu và giải quyết vấn đề. Về mặt chuyên môn, em đã nâng cao hiểu biết về JavaScript ES6, TypeScript, Angular và các thư viện liên quan. Em xin cảm ơn sự hỗ trợ và hướng dẫn tận tâm từ anh Trần Đức Vinh đã giúp đỡ em rất nhiều trong quá trình thực tập tại công ty. Thực tập tại công ty đã giúp em xây dựng nền tảng cho sự phát triển cá nhân và sự nghiệp tương lai của mình trong lĩnh vực phát triển phần mềm.

APPVITY đang là đối tác tin tưởng của nhiều khách hàng, tạo ra được những sản phẩm phục vụ cho nhiều lĩnh vực và là công ty tiên phong trong phát triển công nghệ mới, tạo điều kiện thuận lợi cho những thực tập sinh như em có thể tiếp cận với công việc trong lĩnh vực phần mềm một cách chuyên nghiệp hơn và hiểu được quy trình làm việc.

Trong tương lai em hi vọng là bản thân mình đủ năng lực để có cơ hội đóng góp cho ngành công nghệ thông tin và phần mềm cùng các anh chị đồng nghiệp tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam hoặc các công ty phần mềm khác.

## CHƯƠNG 2: NỘI DUNG THỰC TẬP

### 2.1. Tìm hiểu và cài đặt môi trường, các tool và các framework công ty sử dụng

#### 2.1.1. Sơ lược về TypeScript

TypeScript là một ngôn ngữ lập trình mã nguồn mở, được phát triển bởi Microsoft, và ra mắt lần đầu tiên vào năm 2012. TypeScript là một sự mở rộng của JavaScript, bổ sung các tính năng như hệ thống kiểu tĩnh (static typing), giúp phát hiện lỗi ngay trong quá trình viết mã thay vì chờ đến khi thực thi. TypeScript biên dịch xuống JavaScript thuần túy, do đó có thể chạy trên bất kỳ môi trường nào hỗ trợ JavaScript.



*Hình 2. 1 Logo Typescript*

Một trong những đặc điểm nổi bật của TypeScript là khả năng hỗ trợ lập trình hướng đối tượng (Object-Oriented Programming - OOP). Điều này bao gồm các khái niệm như lớp (class), kế thừa (inheritance), giao diện (interface), và module, giúp các nhà phát triển viết mã theo cấu trúc và dễ bảo trì hơn.

Trên cơ sở ngôn ngữ JavaScript mạnh mẽ, TypeScript giúp đảm bảo tính chính xác và độ tin cậy cao trong quá trình phát triển phần mềm. Bằng cách hỗ trợ kiểu tĩnh, TypeScript cho phép xác định kiểu dữ liệu tường minh cho biến, tham số và giá trị trả về. Điều này giúp phát hiện lỗi sớm và cung cấp thông tin gợi ý mã nhanh chóng, giảm thiểu lỗi gây ra bởi kiểu dữ liệu không chính xác.

Tuy nhiên, TypeScript không chỉ dừng lại ở việc cung cấp kiểu tĩnh. Nó cũng đi kèm với một loạt các tính năng mạnh mẽ khác như interfaces, generics và modules, giúp tổ chức mã một cách rõ ràng và tạo điều kiện cho việc tái sử dụng mã. TypeScript tích hợp tốt với các công cụ phát triển và thư viện hiện có của JavaScript, và cung cấp hỗ trợ cho các tính năng mới của JavaScript ngay khi chúng được giới thiệu.

TypeScript được sử dụng rộng rãi trong các dự án phát triển web và ứng dụng di động. Nó tích hợp tốt với các framework phổ biến như Angular, React và Node.js, và có một cộng đồng lớn và năng động.

### **(\*) Một số dữ liệu phổ biến của TypeScript**

Dưới đây là một số dữ liệu phổ biến trong TypeScript:

#### **1. Kiểu dữ liệu cơ bản:**

- number: số
- string: chuỗi ký tự
- boolean: giá trị logic (true hoặc false)
- null: giá trị null
- undefined: giá trị chưa được định nghĩa
- any: kiểu dữ liệu bất kỳ

#### **2. Kiểu dữ liệu mảng:**

- Array: mảng chứa các phần tử có cùng kiểu dữ liệu
- T[]: mảng chứa các phần tử có kiểu dữ liệu T (thay thế T bằng kiểu dữ liệu cụ thể)

#### **3. Kiểu dữ liệu đối tượng:**

- object: đối tượng không xác định cụ thể
- { key: type }: đối tượng có thuộc tính key với kiểu dữ liệu type

#### **4. Kiểu dữ liệu hợp nhất (Union types):**

- number | string: kiểu dữ liệu có thể là số hoặc chuỗi
- type1 | type2 | type3: kiểu dữ liệu có thể là một trong các kiểu được liệt kê



3. **Modules và Namespaces:** TypeScript hỗ trợ modules và namespaces để tổ chức mã thành các đơn vị độc lập và phân chia mã thành các phạm vi khác nhau. Điều này giúp quản lý mã dễ dàng, tăng tính mô-đun và tái sử dụng mã.
4. **Generics:** TypeScript hỗ trợ generics, cho phép tạo ra các hàm và lớp có thể làm việc với nhiều kiểu dữ liệu khác nhau. Generics giúp tăng tính linh hoạt và tái sử dụng mã.
5. **Type Annotations và Type Inference:** TypeScript cho phép bạn gán kiểu dữ liệu cho biến và phép gán, hoặc cho phép TypeScript suy luận kiểu dữ liệu dựa trên ngữ cảnh. Điều này giúp xác định rõ ràng kiểu dữ liệu và đảm bảo tính chính xác của mã.
6. **Enumerations:** TypeScript hỗ trợ kiểu dữ liệu enum, cho phép định nghĩa một tập hợp các giá trị có thể cho một biến hoặc thuộc tính.
7. **Decorators:** TypeScript hỗ trợ decorators, là một tính năng mạnh mẽ để thay đổi hoặc mở rộng hành vi của các lớp, phương thức, thuộc tính và tham số hàm.
8. **Tích hợp với công cụ phát triển:** TypeScript tích hợp tốt với các công cụ phát triển như trình biên dịch TypeScript (TypeScript compiler) và trình soạn thảo mã (code editor) như Visual Studio Code, cung cấp các tính năng gợi ý mã, kiểm tra kiểu thông minh, debug và tự động hoàn thành mã.

Tổng thể, TypeScript cung cấp các tính năng và chức năng mạnh mẽ để phát triển ứng dụng JavaScript chất lượng cao, đáng tin cậy và dễ bảo trì.

### **(\*) TypeScript và Angular**

Angular, bắt đầu từ phiên bản 2 trở đi, đã được xây dựng dựa trên TypeScript. Điều này mang lại nhiều lợi ích cho cả các nhà phát triển mới và có kinh nghiệm. TypeScript cung cấp một nền tảng mạnh mẽ để phát triển các ứng dụng Angular có cấu trúc tốt và dễ bảo trì. Nhờ có TypeScript, các nhà phát triển Angular có thể tận dụng tối đa các tính năng như tự động hoàn thành mã (auto-completion), kiểm tra kiểu (type checking), và hỗ trợ lập trình hướng đối tượng.



TypeScript cũng giúp tăng cường tính rõ ràng và an toàn của mã nguồn trong các dự án Angular, nhờ vào hệ thống kiểu tĩnh. Điều này giúp giảm thiểu lỗi và làm cho mã nguồn trở nên dễ hiểu hơn, đặc biệt là khi làm việc trong các nhóm phát triển lớn.

### **2.1.2. Sơ lược về SCSS**

SCSS (Sassy CSS) là một trong những phần mở rộng phổ biến nhất của CSS, và là một phần của bộ công cụ Sass (Syntactically Awesome Stylesheets). SCSS bổ sung các tính năng nâng cao cho CSS, chẳng hạn như biến (variables), phép toán (operations), hàm (functions), và kế thừa (inheritance), giúp viết mã CSS trở nên dễ dàng và hiệu quả hơn. SCSS có cú pháp tương thích với CSS truyền thống, điều này giúp việc chuyển đổi từ CSS sang SCSS trở nên đơn giản.

SCSS cho phép người dùng sử dụng các khái niệm lập trình cơ bản như biến và hàm trong CSS, giúp mã nguồn trở nên rõ ràng, có cấu trúc và dễ bảo trì hơn. Ngoài ra, SCSS còn hỗ trợ việc chia nhỏ các tệp tin CSS thành các module, giúp quản lý và bảo trì mã nguồn dễ dàng hơn trong các dự án lớn.

### **(\*) Ứng dụng của Angular, TypeScript và SCSS trong phát triển web hiện đại**

#### **➤ Sự kết hợp hoàn hảo**

Angular, TypeScript và SCSS là một sự kết hợp hoàn hảo cho việc phát triển các ứng dụng web hiện đại. Angular cung cấp một nền tảng mạnh mẽ để xây dựng các ứng dụng web phức tạp, trong khi TypeScript cung cấp các tính năng nâng cao giúp tăng tính rõ ràng và an toàn cho mã nguồn. SCSS, với khả năng mở rộng của CSS, giúp việc thiết kế giao diện người dùng trở nên dễ dàng và linh hoạt hơn.

Sự kết hợp này không chỉ giúp tăng cường hiệu suất phát triển mà còn giúp giảm thiểu lỗi và tăng cường khả năng bảo trì mã nguồn. Các nhà phát triển có thể tận dụng tối đa các tính năng của cả ba công nghệ này để tạo ra những ứng dụng web có chất lượng cao, với giao diện người dùng đẹp mắt và hiệu suất hoạt động tốt.

#### **➤ Ví dụ thực tế**

Một ví dụ điển hình cho sự kết hợp này là việc phát triển một ứng dụng quản lý vận chuyển. Trong dự án này, Angular có thể được sử dụng để quản lý các thành phần giao diện phức tạp và đồng bộ hóa dữ liệu giữa client và server. TypeScript giúp tăng cường

tính rõ ràng và an toàn của mã nguồn, giúp các nhà phát triển dễ dàng kiểm soát và mở rộng ứng dụng. SCSS giúp tạo ra một giao diện người dùng trực quan, với các phong cách CSS được tổ chức tốt và dễ bảo trì.

### (\*) Cấu trúc thư mục

to-do-app/

```
|
|
|— src/
|   |— app/
|   |   |— app.component.html
|   |   |— app.component.scss
|   |   |— app.component.ts
|   |   |— app.module.ts
|   |— index.html
|— angular.json
|— package.json
|— tsconfig.json
```

### 2.1.3. Sơ lược về Tailwind CSS



*Hình 2. 2 Logo Tailwind CSS*

Tailwind CSS là một framework CSS "utility-first" được phát triển bởi Adam Wathan và đội ngũ của anh ấy. Khác biệt chính giữa Tailwind CSS và các framework khác như Bootstrap hay Foundation là Tailwind không cung cấp các thành phần giao diện người dùng (UI components) cố định. Thay vào đó, nó cung cấp một bộ sưu tập lớn các lớp

tiện ích (utility classes) mà bạn có thể sử dụng để xây dựng các thiết kế tùy chỉnh mà không cần phải viết CSS từ đầu.

## **Lợi ích của Tailwind CSS**

### *Tính linh hoạt cao*

Tailwind CSS có thể sử dụng cho bất kỳ loại dự án nào, từ trang web đơn giản đến các ứng dụng web phức tạp. Nó tương thích với nhiều công cụ xây dựng và phát triển như Webpack, PostCSS, và các framework như React, Vue, Angular, v.

### *Tối ưu hiệu suất*

Tailwind CSS sử dụng một công cụ xây dựng để loại bỏ các lớp tiện ích không sử dụng trước khi sản xuất, giúp giảm kích thước tệp CSS cuối cùng. Điều này đảm bảo rằng chỉ có những lớp tiện ích cần thiết mới được tải vào trang web, giúp tăng tốc độ tải trang.

### *Phát triển nhanh chóng*

Với Tailwind CSS, bạn có thể nhanh chóng tạo ra các giao diện người dùng mà không cần phải chuyển qua lại giữa các tệp HTML và CSS. Điều này làm cho quá trình phát triển nhanh hơn và hiệu quả hơn.

## **(\*) Cấu trúc thư mục**

Khi sử dụng **Tailwind CSS** trong một dự án, cấu trúc thư mục có thể thay đổi tùy theo yêu cầu của dự án và cách bạn tổ chức mã nguồn. Tuy nhiên, dưới đây là một cấu trúc thư mục điển hình khi sử dụng **Tailwind CSS** với dự án web thông thường:

### **Cấu Trúc Thư Mục Cơ Bản với Tailwind CSS**

To-do-app/

```
|
|
| — public/
|   | — index.html
|   | — ...
|
```

```

|— src/
|   |— assets/
|   |— css/
|       |— styles.css
|       |— tailwind.css
|   |— js/
|       |— app.js
|   |— index.html
|   |— ...
|
|— tailwind.config.js
|— postcss.config.js
|— package.json
|— README.md

```

#### 2.1.4. Sơ lược về Javascript

(\*) Lịch sử phát triển của Javascript

- Brendan Eich chính là người đã phát triển JS tại Netscape với tiền thân là Mocha. Sau đó, Mocha được đổi thành LiveScript và cuối cùng mới đổi thành JavaScript.
- **Năm 1998**, JavaScript với phiên bản mới nhất là ECMAScript 2 phát hành và đến năm 1999 thì ECMAScript 3 được ra mắt.
- **Năm 2016**, ứng dụng JavaScript đã đạt kỷ lục lên tới 92% website sử dụng, đồng thời cũng được đánh giá là một công cụ cực kỳ quan trọng đối với lập trình viên

(\*) Javascript Framework

Là thư viện được xây dựng dựa trên JS phục vụ cho từng mục đích và lĩnh vực khác nhau

**JavaScript Framework** là thư viện được xây dựng dựa vào ngôn ngữ lập trình JavaScript. Từ đó, mỗi framework được tạo ra để phục cho từng lĩnh vực khác nhau.

Bạn có thể tìm hiểu kỹ hơn về framework là gì, sẽ giúp bạn có thêm nhiều thông tin rõ ràng hơn. Hiện nay, có rất nhiều JavaScript Framework thông dụng như:

- **Reactjs:** Thư viện dùng cho ứng dụng mobile.
- **Node.js:** Dùng để xây dựng và phát triển ứng dụng realtime từ phía máy chủ.
- **Angular:** Dùng để xây dựng ứng dụng Single Page

(\*) Ưu và nhược điểm

Ưu điểm :

- Chương trình rất dễ học.
- Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.
- Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
- JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
- Được các chuyên gia đánh giá là một loại ngôn ngữ lập trình nhẹ và nhanh hơn nhiều so với các ngôn ngữ lập trình khác.
- JS còn có thể được gắn trên một số các element hoặc những events của các trang web.
- Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.
- Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.

Nhược điểm :

- JS Code Snippet khá lớn.
- JS dễ bị các hacker và scammer khai thác hơn.
- JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.
- Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.

- Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.
- Vì tính bảo mật và an toàn nên các Client-Side **Javascript** sẽ không cho phép đọc hoặc ghi các file.
- JS không được hỗ trợ khi bạn sử dụng ở trong tình trạng thiết bị được kết nối mạng.

### 2.1.5. Cài đặt và dựng source

- Ngôn ngữ: Angular v16
- Node: version v18.20.4
- Firebase
- Cài đặt IDE

### 2.1.6. Sơ lược về Angular

Angular là một framework JavaScript mã nguồn mở, được phát triển và duy trì bởi Google, với mục tiêu đơn giản hóa việc phát triển các ứng dụng web một trang (Single Page Applications - SPAs).

Ra đời vào năm 2010, Angular đã nhanh chóng trở thành một trong những framework phổ biến nhất cho việc xây dựng các ứng dụng web. Với khả năng giúp quản lý và đồng bộ hóa dữ liệu, Angular cho phép các nhà phát triển tạo ra những ứng dụng phức tạp một cách dễ dàng hơn.

Hoạt động của nó dựa theo các thuộc tính mở rộng HTML (các attributes tuân theo quy tắc của Angular). Đây là một Framework mã nguồn mở được sử dụng hoàn toàn miễn phí và được ưa chuộng bởi hàng ngàn các lập trình viên trên thế giới. Ở những nước ngoài khác, Framework này được thế hệ Web 2.0 phát triển khá mạnh, tại Việt Nam đã được các công ty, doanh nghiệp, tập đoàn lớn áp dụng rộng rãi cho việc phát triển phần mềm và Angular là một open source, nó hoàn toàn miễn phí và được phát triển bởi hàng ngàn các lập trình viên giỏi trên thế giới.



*Hình 2. 3 Logo Framework Angular*

### **(\*) Các đặc tính của Angular**

Angular sử dụng mô hình MVC (Model-View-Controller) để tách biệt dữ liệu (Model) khỏi giao diện người dùng (View), và giúp quản lý luồng công việc logic (Controller). Cách tiếp cận này giúp mã nguồn trở nên gọn gàng, dễ quản lý và bảo trì hơn. Ngoài ra, Angular cũng cung cấp một hệ thống ràng buộc dữ liệu hai chiều (two-way data binding), cho phép tự động đồng bộ hóa dữ liệu giữa model và view, giúp loại bỏ nhiều thao tác thủ công trong việc cập nhật giao diện người dùng.

### **(\*) MVC (Model View Controller)**

Framework này được xây dựng dựa theo khuôn mẫu nổi tiếng MVC (Model-View-Controller). Đây là một mẫu thiết kế mà ngày nay được sử dụng hầu hết các ứng dụng web. Mô hình này dựa trên việc chia lớp logic nghiệp vụ, lớp dữ liệu và lớp trình bày thành các phần riêng biệt khác nhau. Việc phân chia như vậy được thực hiện để mỗi phần có thể được quản lý dễ dàng hơn.

### **(\*) Data Model Binding**

Không nhất thiết phải viết mã đặc biệt để liên kết dữ liệu với HTML controls. Công đoạn này có thể thực hiện bởi Angular chỉ bằng cách thêm một vài đoạn mã.

### **(\*) Có sẵn Unit Testing**

Thêm vào đó, Angular cung cấp một bộ công cụ phong phú để kiểm tra mã nguồn (testing), giúp đảm bảo chất lượng của ứng dụng. Angular hỗ trợ cả Unit Testing (kiểm

thử đơn vị) và End-to-End Testing (kiểm thử từ đầu đến cuối), giúp các nhà phát triển dễ dàng phát hiện và khắc phục lỗi trong quá trình phát triển.

Các lập trình viên tại Google không chỉ phát triển Angular mà còn phát triển một khung kiểm thử có tên “Karma” phục vụ cho việc thiết kế các Unit Testing cho các ứng dụng Angular.

### **(\*) Các tính năng quan trọng của Angular**

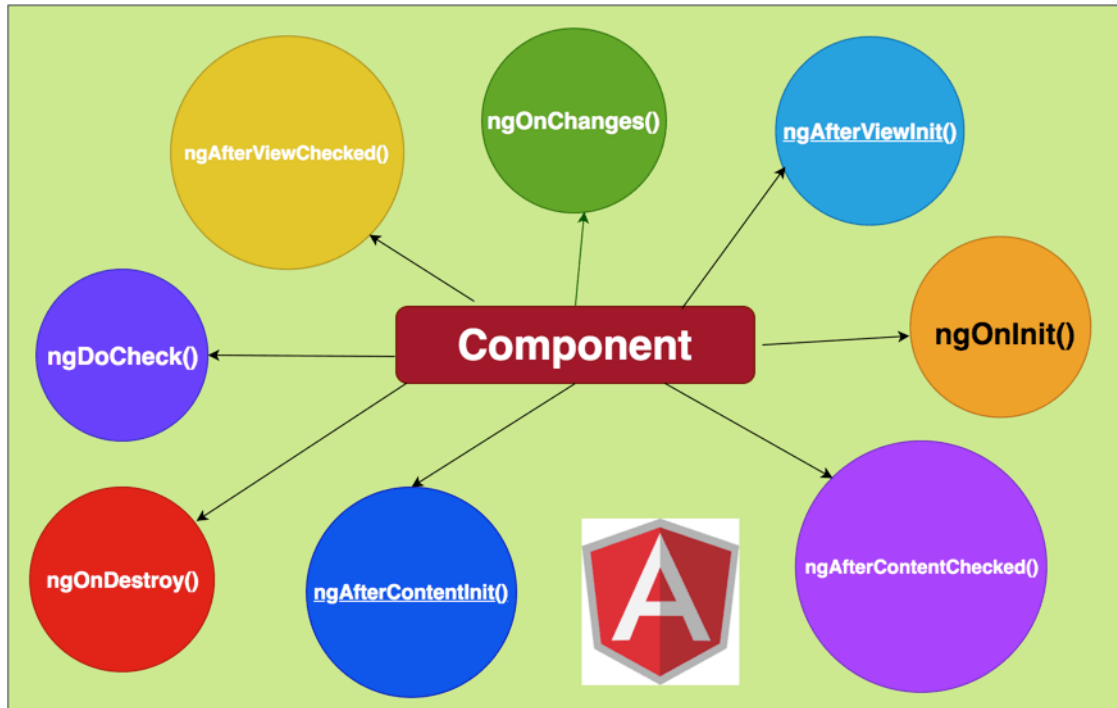
Dưới đây là các tính năng cốt lõi quan trọng trong Angular:

- Data-binding: (liên kết dữ liệu) sẽ có chức năng tự động đồng bộ dữ liệu giữa model và view.
- Scope: (Phạm vi) Đây là những liên kết giữa Controller và View.
- Controller: Đó là những hàm JavaScript xử lý kết hợp với bộ điều khiển Scope.
- Service: Như đã đề cập ở trên, Angular sử dụng chủ yếu là các API được tạo dựng từ các web service (PHP, ASP) để thao tác với DB.
- Filters: Bộ lọc chọn lọc ra các thành phần của một mảng và sau đó trả về mảng mới.
- Directives: đánh dấu các yếu tố của DOM, nghĩa là sẽ tạo ra các thẻ HTML tùy chỉnh.
- Templates: thông tin được hiển thị từ controller, đây là một thành phần của views.
- Routing: tạo sự chuyển đổi giữa các action trong controller.
- MVC: Mô hình chia ra các thành phần riêng biệt thành Model, View, Controller. Đây được xem là một mô hình khá hay nhưng trong Angular thì nó được “cải biên” lại một chút gần giống với MVVM.
- Deep Linking: Liên kết sâu, nó cho phép mã hóa trạng thái của ứng dụng trong các URL để có thể đánh dấu được với công cụ tìm kiếm.
- Dependency Injection: Angular giúp các nhà phát triển dễ dàng hơn trong việc tạo ứng dụng để phát triển, hiệu và thử nghiệm dễ dàng.



### (\*) Các components chính

- ng-app: định nghĩa và liên kết ứng dụng Angular tới HTML.
- ng-model: gắn kết các giá trị của dữ liệu ứng dụng Angular đến các điều khiển đầu vào HTML.
- ng-bind: liên kết dữ liệu ứng dụng Angular đến các thẻ HTML.



Hình 2. 4 Angular Lifecycle Hook

### (\*) Cài đặt Angular

#### Tải và cài đặt Nodejs

```
node -v
```

```
npm -v
```

#### Cài đặt Angular CLI

```
# npm install -g @angular/cli
```

#### Tạo dự án Angular mới

```
ng new project-name
```

#### Chạy ứng dụng Angular

ng serve
----------

## 2.2. Thực hành code website THEVOUX

### 2.2.1. Tìm hiểu source FE

Theo **package.json** src sử dụng các công nghệ sau:

Angular Framework:

*Phiên bản Angular: Dự án này sử dụng Angular phiên bản ^16.2.0. Điều này bao gồm các module cốt lõi của Angular:*

@angular/animations: ^16.2.0

@angular/common: ^16.2.0

@angular/compiler: ^16.2.0

@angular/core: ^16.2.0

@angular/forms: ^16.2.0

@angular/platform-browser: ^16.2.0

@angular/platform-browser-dynamic: ^16.2.0

@angular/router: ^16.2.0

*Công cụ phát triển Angular: Để hỗ trợ phát triển và xây dựng ứng dụng, dự án sử dụng các công cụ liên quan đến Angular:*

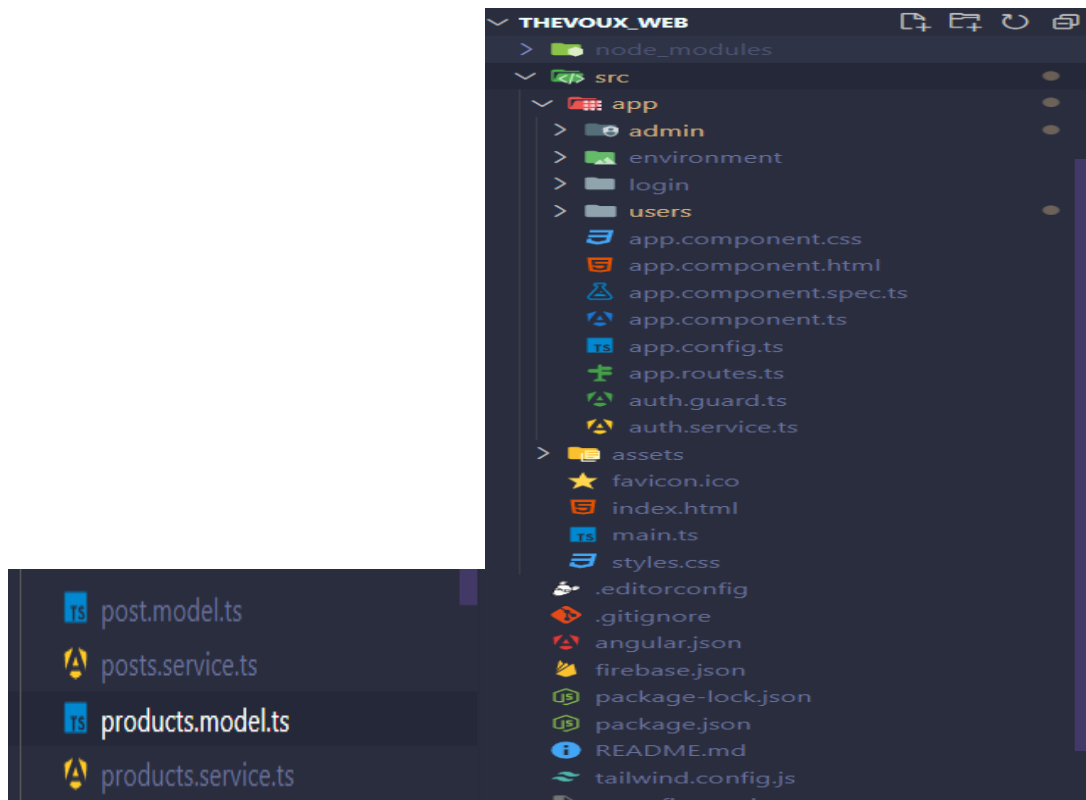
@angular-devkit/build-angular: ^16.2.0

@angular/cli: ~16.2.0

@angular/compiler-cli: ^16.2.0

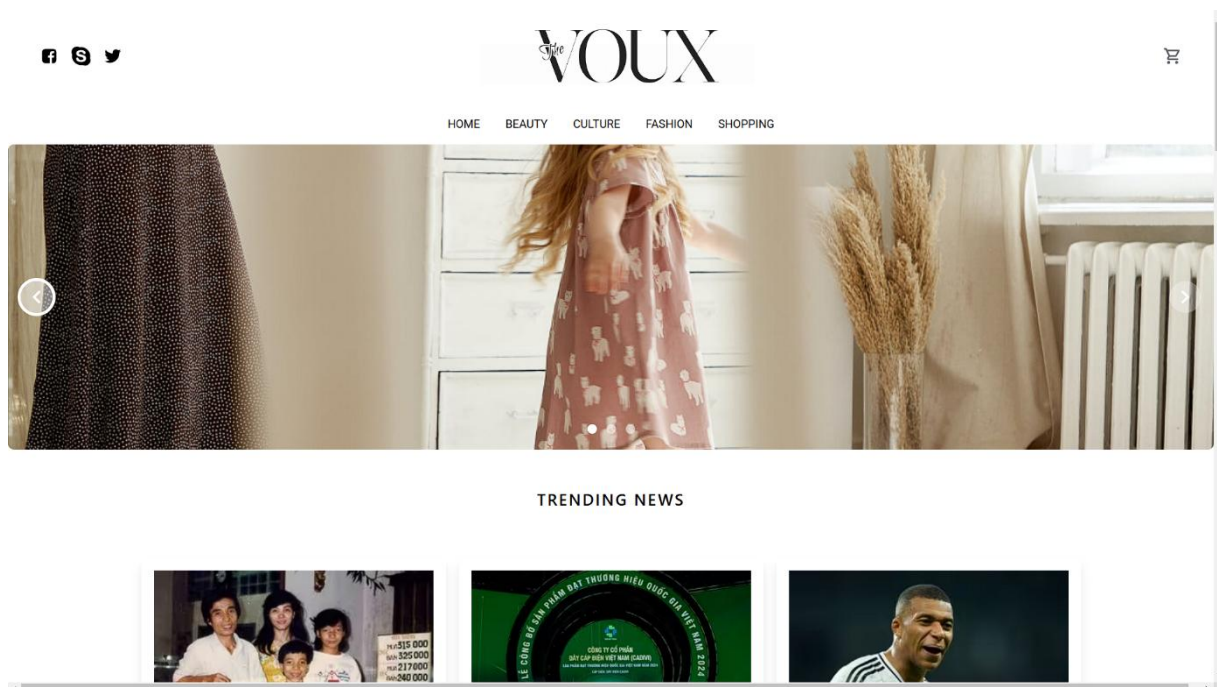
TypeScript:

Phiên bản TypeScript: Dự án sử dụng TypeScript phiên bản ~4.8.3

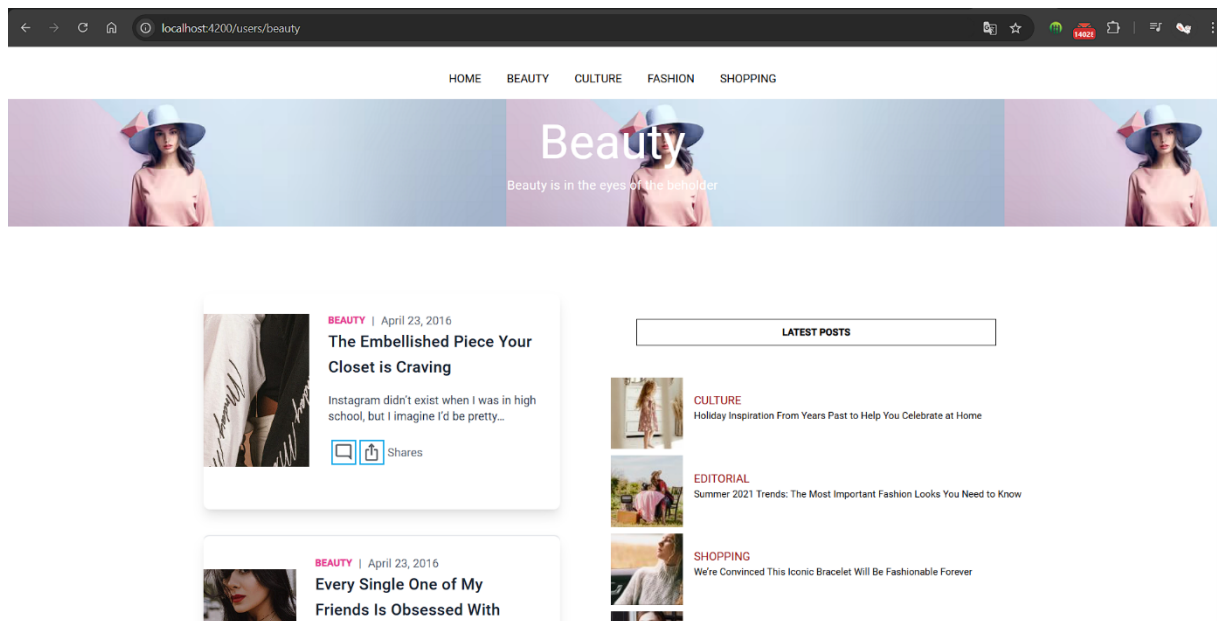


Hình 2. 5 Cấu trúc src và service FE

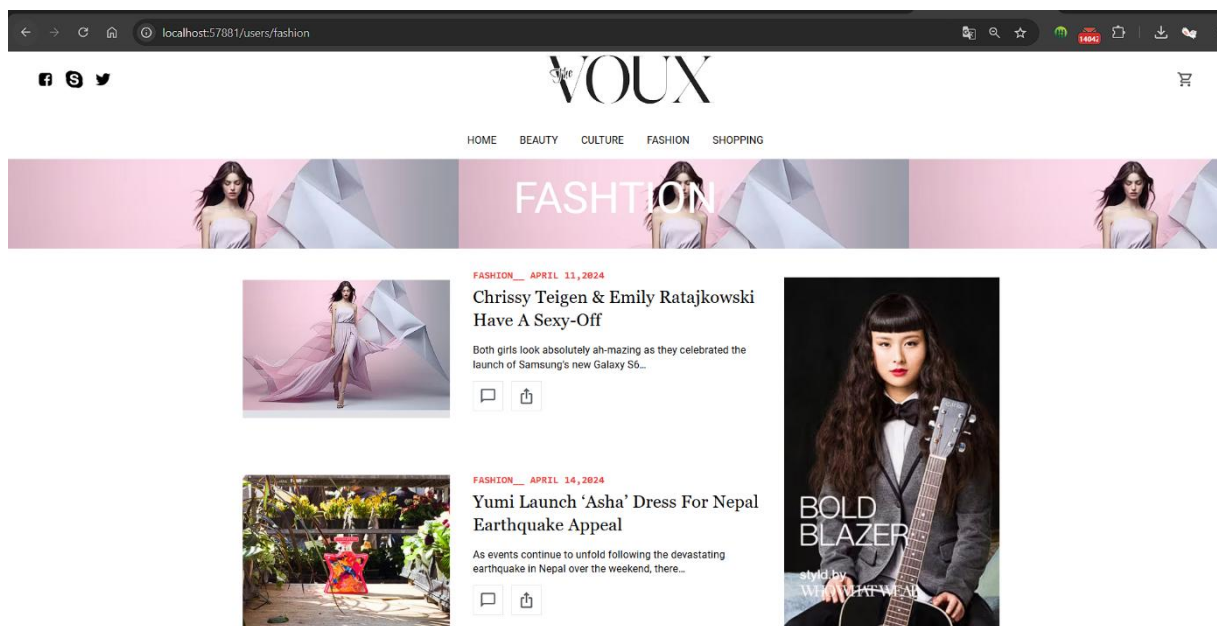
## 2.2.2. Tạo trang Home



Hình 2. 6 Giao diện trang Home

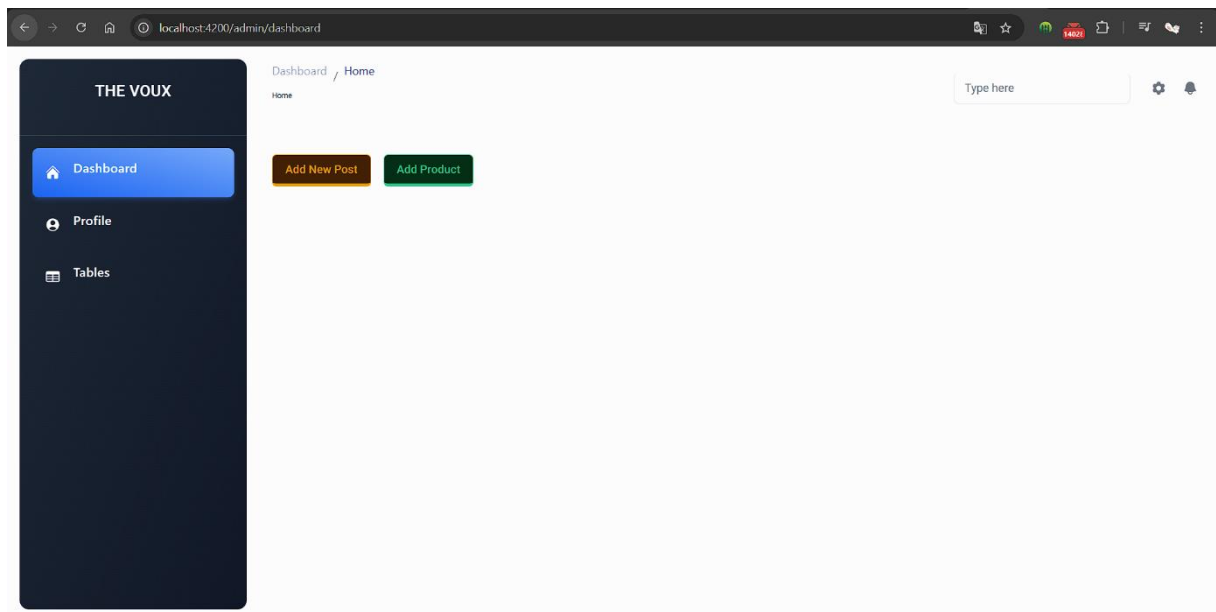


Hình 2. 7 Giao diện trang Beauty



Hình 2. 8 Giao diện trang fashion

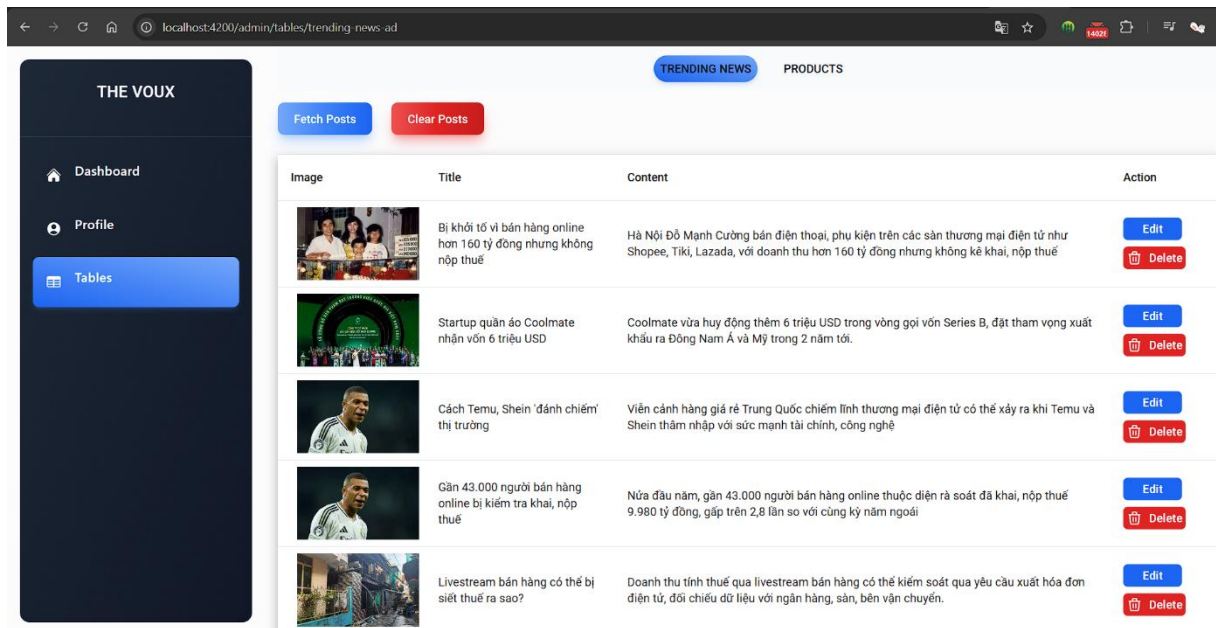
### 2.2.3. Tạo chức năng admin gửi dữ liệu về Trending News lên firebase



*Hình 2. 9 Giao diện dashboard nút button Add New Post*

A screenshot of a form for adding a new post. The form is contained within a light gray border. It has three input fields: 'Nhập URL image\*' (with the example 'https://example.com/image.jpg'), 'Title\*', and 'Content\*'. The 'Content\*' field is a text area with a small icon in the bottom right corner. At the bottom of the form are two buttons: a red 'Cancel' button and a gray 'Send Post' button.

*Hình 2. 10 Giao diện form điền add new post*



Hình 2. 11 Giao diện table Trending News của dashboard admin

(\*)Code chức năng sửa và tạo đẩy dữ liệu từ form lên firebase

Phương thức onSubmit() trong Angular được sử dụng trong một form để xử lý sự kiện khi người dùng gửi (submit) form. Nó thực hiện một trong hai hành động:

- Cập nhật bài viết nếu form đang trong chế độ chỉnh sửa.
- Tạo mới bài viết nếu form không trong chế độ chỉnh sửa.

Kiểm tra tính hợp lệ của form dùng if (this.postForm.valid).

Cập nhật bài viết nếu form đang ở chế độ chỉnh sửa. Nếu this.isEditMode là true, hàm sẽ cập nhật một bài viết hiện có bằng cách gọi this.postsService.updatePost(). Phương thức updatePost nhận ID của bài viết cần cập nhật (this.data.id) và dữ liệu từ form (this.postForm.value) để gửi yêu cầu cập nhật đến firebase. Sau khi cập nhật thành công, hàm sẽ đóng dialog (form) bằng cách gọi this.dialogRef.close().

Tạo bài viết mới nếu form không ở chế độ chỉnh sửa. Nếu this.isEditMode là false, hàm sẽ tạo mới một bài viết bằng cách gọi this.postsService.createAndStorePost(). Dữ liệu từ form (this.postForm.value) sẽ được gửi đến firebase dưới dạng các tham số để tạo

một bài viết mới (bao gồm imageUrl, title, content). Sau khi tạo mới bài viết thành công, hàm đóng dialog và trả về thông tin bài viết mới được tạo qua `this.dialogRef.close()`.

Đóng dialog sau khi thao tác hoàn tất : Sau mỗi hành động (cập nhật hoặc tạo mới), phương thức sử dụng `subscribe()` để xử lý phản hồi từ firebase. Trong trường hợp này, chỉ đơn giản là đóng dialog sau khi có phản hồi thành công.

`pipe(take(1))`: Phương thức này đảm bảo rằng observable chỉ phát hành một lần và sau đó tự động hoàn thành (ngừng phát hành dữ liệu).

```
onSubmit(): void {
  if (this.postForm.valid) {
    if (this.isEditMode) {
      // Update the post
      this.postsService
        .updatePost(this.data.id, this.postForm.value)
        .pipe(take(1))
        .subscribe((responseData) => {
          this.dialogRef.close();
        });
    } else {
      // Create a new post
      this.postsService
        .createAndStorePost(
          this.postForm.value.imageUrl,
          this.postForm.value.title,
          this.postForm.value.content
        )
        .pipe(take(1))
        .subscribe((responseData) => {
          this.dialogRef.close({ action: 'add', data: responseData });
        });
    }
  }
}
```

```
}
```

(\*)Code chức năng update post

```
updatePost(postId: string, postData: { title: string; content:
string }): Observable<Post> {
    return this.http.put<Post>(`https://httpclient-a4bf8-default-
rtdb.firebaseio.com/posts/${postId}.json`, postData)
    .pipe(
        map(responseData => {
            const updatedPosts = this.postsSubject.value.map(post =>
                post.id === postId ? { ...responseData, id: postId } : post
            );
            this.postsSubject.next(updatedPosts);
            return responseData;
        })
    );
}
```

(\*)Code chức năng delete post

Phương thức deletePost() được sử dụng để xóa một bài đăng (post) khỏi hệ thống bằng cách gửi yêu cầu HTTP đến firebase, trong đó URL của bài đăng được xác định bởi ID của bài đăng. Sau khi xóa thành công, firebase sẽ trả về một phản hồi, xác nhận việc xóa bài đăng đã được thực hiện.

```
// Delete a post by ID
deletePost(postId: string): Observable<void> {
    this.http.delete<void>(`${this.firebaseUrl}/${postId}.json`);
}
```

(\*)Code chức năng edit



Phương thức `onEdit()` được sử dụng để mở một hộp thoại chỉnh sửa bài đăng, cho phép người dùng thay đổi thông tin bài đăng, sau đó cập nhật bài đăng đó lên server và làm mới danh sách bài đăng trong giao diện người dùng.

Sử dụng dịch vụ `MatDialog` của `Angular Material` để mở một hộp thoại (dialog). Trong trường hợp này, hộp thoại chứa `DashboardInfoComponent`, có thể là một component dùng để hiển thị và chỉnh sửa thông tin bài đăng.

`if (result) { ... }`: Kiểm tra nếu có thay đổi từ hộp thoại, gửi yêu cầu cập nhật bài đăng lên server thông qua dịch vụ `postsService.updatePost(post.id!, result)`.

`post.id!`: Đảm bảo rằng ID bài đăng không bị null hoặc undefined.

Khi bài đăng được cập nhật thành công, hàm sẽ nhận lại dữ liệu bài đăng đã được cập nhật (`responseData`).

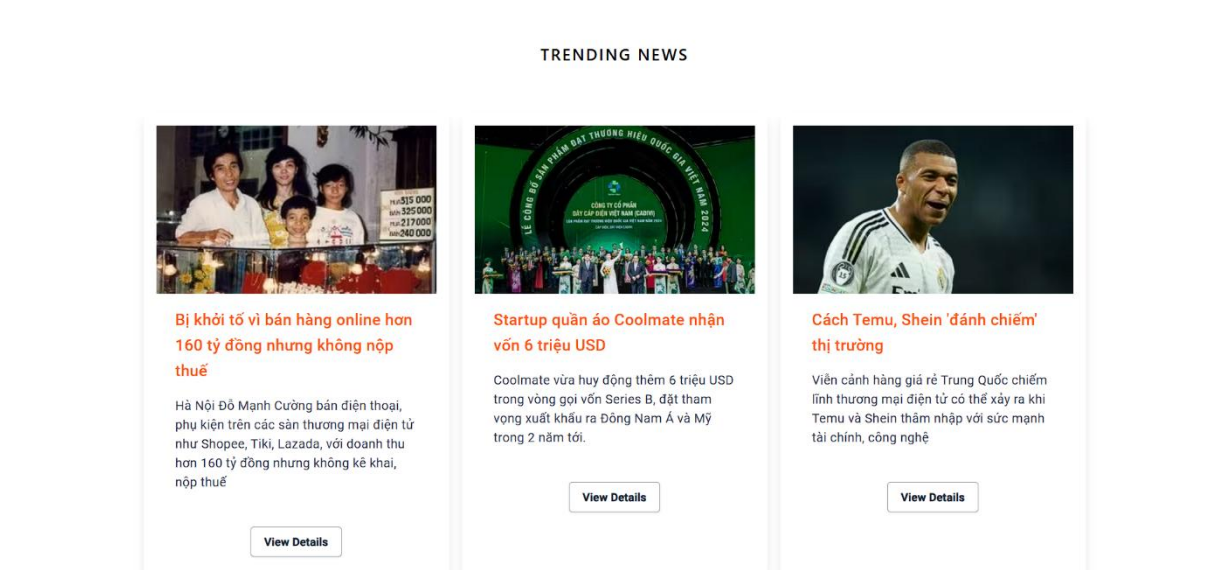
Cập nhật danh sách các bài đăng trong `this.posts` bằng cách thay thế bài đăng cũ với bài đăng mới đã được trả về từ server (`responseData`). Sử dụng phương thức `map` để duyệt qua tất cả bài đăng và thay thế bài đăng có ID tương ứng.

```
onEdit(post: Post): void {
  const dialogRef = this.dialog.open(DashboardInfoComponent, {
    width: '300px',
    data: { ...post }, // Pass a copy of the post to the dialog
  });

  dialogRef.afterClosed().subscribe((result) => {
    if (result) {
      this.postsService
        .updatePost(post.id!, result)
        .subscribe((responseData) => {
          this._snackBar.open('Post updated successfully', 'Close',
            {
              duration: 3000,
            });
          this.posts = this.posts.map((p: { id: string | undefined })
```

```
=>
        p.id === post.id ? { ...responseData } : p
    );
    // Refresh the table after editing
    console.log('createAndStorePost.....',
responseData);
    });
    }
    this.onFetchPosts();
  });
}
```

#### 2.2.4. Tạo chức năng hiển thị Trending News trên trang Home



Hình 2. 12 Giao diện của thành phần Trending News ở trang home

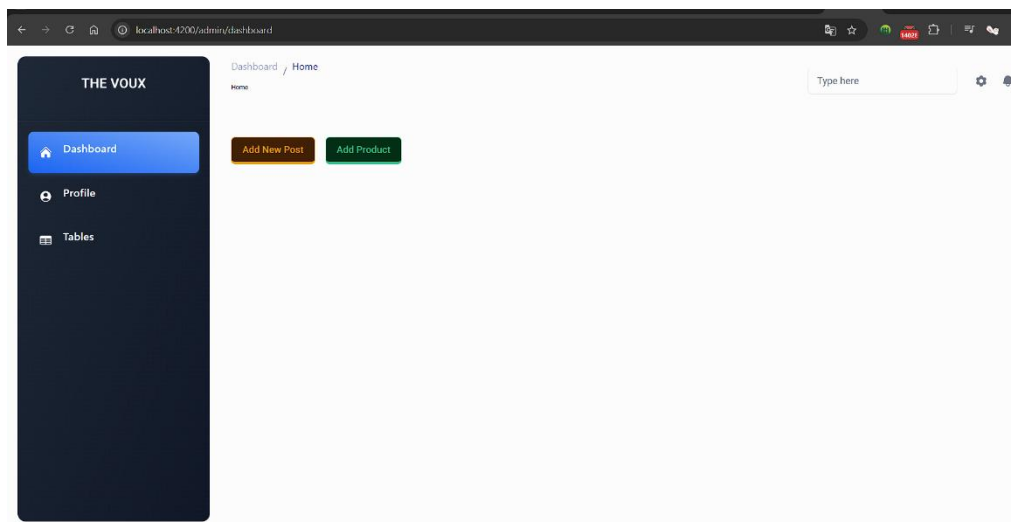
(\*)Code chức năng lấy dữ liệu Trending News từ firebase về cho users

Đoạn code này định nghĩa giao diện người dùng cho danh sách bài đăng, nơi mỗi bài đăng được hiển thị dưới dạng một thẻ riêng biệt với các thông tin cơ bản như hình ảnh, tiêu đề và nội dung tóm tắt.

Danh sách bài đăng: sử dụng vòng lặp \*ngFor để hiển thị một loạt bài đăng từ dữ liệu posts.

```
<section class="bg-gray-2 dark:bg-dark pt-10 pb-6 lg:pt-[40px] lg:pb-2">
  <div class="container mx-auto">
    <div class="flex flex-wrap gap-4 -mx-4 justify-center">
      <div
        *ngFor="let post of posts"
        class="max-w-sm rounded overflow-hidden shadow-lg bg-white p-4 mb-4"
      >
        <img
          [src]="post.imageUrl"
          alt="{{ post.title }}"
          class="w-full h-auto"
        />
        <div class="px-6 py-4">
          <h2 class="font-bold text-lg text-orange-500 mb-2">
            {{ post.title }}
          </h2>
          <p class="text-gray-700 text-base">{{ post.content }}</p>
        </div>
        <div class="px-6 pt-4 pb-2 text-center">
          <button
            class="bg-white text-gray-800 font-semibold py-2 px-4 border border-gray-400 rounded shadow"
            View Details
          </button>
        </div>
      </div>
    </div>
  </div>
</section>
```

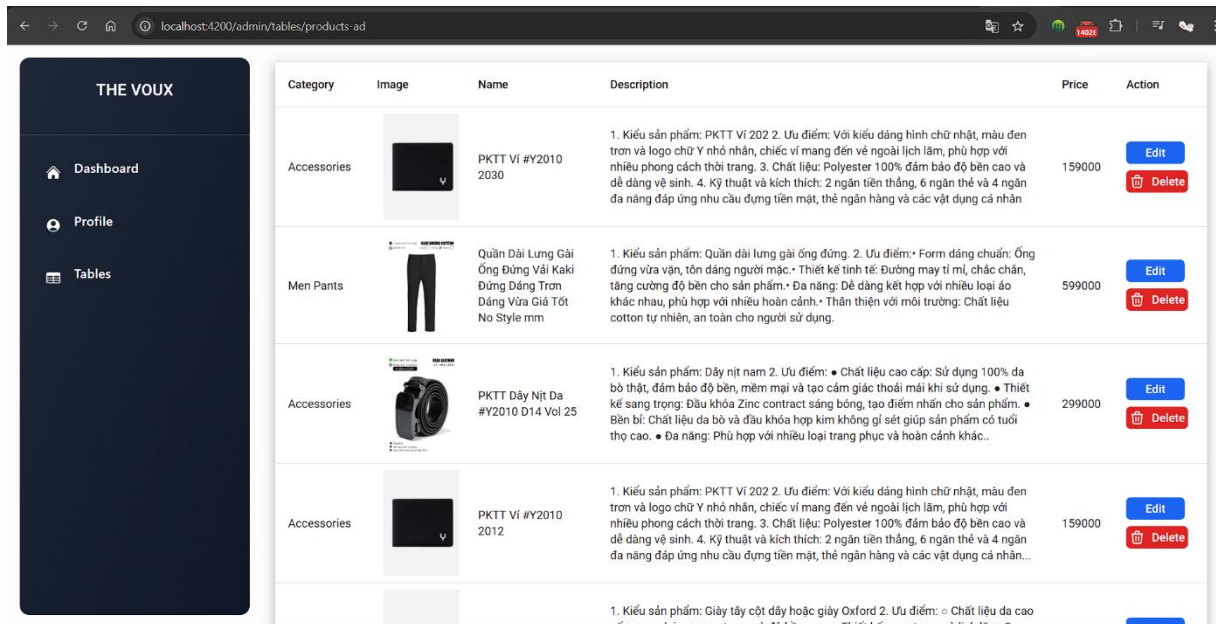
### 2.2.5. Trang admin đẩy products lên firebase



Hình 2. 13 Giao diện dashboard có nút add product

A screenshot of a form for adding a product. The form is enclosed in a light gray border. It starts with a dropdown menu labeled 'Select\*' with a downward arrow. Below this are four input fields: 'Nhập URL image\*', 'Name\*', 'Description\*' (with a text area icon), and 'Price\*'. At the bottom of the form are two buttons: a red 'Cancel' button and a blue 'Submit' button.

Hình 2. 14 Form điền product



Hình 2. 15 Giao diện table product của trang dashboard admin

(\*)Code chức năng tạo or edit products lên firebase

Phương thức onSubmitPr() dùng để quản lý hành động gửi biểu mẫu trong việc tạo mới hoặc chỉnh sửa sản phẩm. Đoạn code sẽ kiểm tra tính hợp lệ của biểu mẫu và thực hiện các hành động tương ứng bao gồm gửi yêu cầu tạo hoặc cập nhật sản phẩm lên server.

Chức năng chỉnh sửa (isEditMode): Nếu chỉnh sửa, mã gọi id product có sẵn lên firebase để cập nhật sản phẩm.

Chức năng tạo mới: Nếu tạo mới, code sẽ gọi id để tạo sản phẩm mới.

Xử lý biểu mẫu hợp lệ: kiểm tra tính hợp lệ của biểu mẫu trước khi thực hiện hành động.

Đóng hộp thoại sau khi hành động thành công: hộp thoại được đóng lại với kết quả phù hợp.

```
onSubmitPr(): void {
  if (this.productsForm.valid) {
    if (this.isEditMode) {
      this.productsService
        .updateProducts(this.dataItem.id, this.productsForm.value)
    }
  }
}
```

```

        .pipe(take(1))
        .subscribe((responseData) => {
            // this.dialogRef.close();
            this.dialogRef.close('ok');
        });
    } else {
        // Create a new post
        let prd = this.productsForm.value
        this.productsService
            .createAndStoreProducts(
                prd.category,
                prd.imageUrl,
                prd.name,
                prd.description,
                prd.price
            )
            .pipe(take(1))
            .subscribe((responseData) => {
                this.dialogRef.close({ action: 'add', data: responseData });
            });
    }
    console.log('thanh cong');
} else {
    console.log('Form không hợp lệ');
}
}

```

(\*)Code chức năng delete

Phương thức onDelete() cung cấp chức năng xóa sản phẩm từ hệ thống. Sẽ yêu cầu người dùng xác nhận trước khi xóa và làm mới danh sách sản phẩm sau khi xóa thành công.

Kiểm tra productId: Đảm bảo rằng có giá trị productId trước khi thực hiện xóa.

Xác nhận xóa: Hiển thị hộp thoại xác nhận (confirm) yêu cầu người dùng xác nhận trước khi xóa sản phẩm.

Gửi yêu cầu xóa đến server: Gửi yêu cầu xóa sản phẩm và làm mới danh sách sau khi xóa thành công.

```
onDelete(productId: string) {  
  if (productId){  
    if (confirm('Bạn có chắc chắn muốn xóa ?')) {  
      this.productsService.deleteProduct(productId).subscribe(() => {  
        console.log('xóa thành công');  
        this.onFetchProducts(); // Refresh the table after deletion  
      });  
    }  
  } else {  
    console.log('Không có id');  
  }  
}
```

*(\*)Code chức năng edit product*

Phương thức onEdit() được sử dụng để mở một dialog chỉnh sửa thông tin sản phẩm và cập nhật thông tin sản phẩm sau khi thay đổi.

Sử dụng MatDialog để mở dialog chỉnh sửa sản phẩm.

Cập nhật sản phẩm trên server qua productsService.updateProducts.

Cập nhật danh sách sản phẩm sau khi thay đổi.

Thông báo thành công qua Snackbar.

```
onEdit(product: Products): void {  
  const dialogConfig = new MatDialogConfig();
```

```

dialogConfig.data = {
  data: { ...product },
  isEditMode: true,
};
dialogConfig.width = '300px';

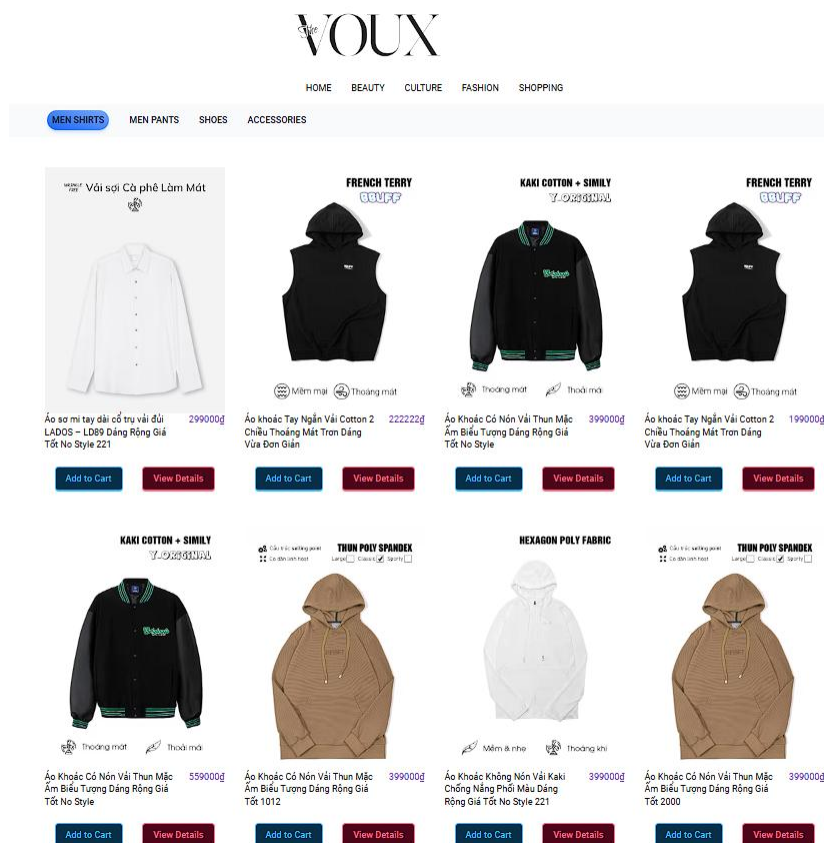
this.dialog
  .open(AddProductComponent, dialogConfig)
  .afterClosed()
  .subscribe((result) => {
    if (result) {
      this.productsService
        .updateProducts(product.id!, result)
        .subscribe((responseData) => {
          this._snackBar.open('Post updated successfully', 'Close',
{
          duration: 3000,
        });
          this.product = this.product.map((p: { id: string |
undefined }) =>
            p.id === product.id ? { ...responseData } : p
          );

          console.log('createAndStoreProducts.....', responseData);
          console.log('Editing Product ID:', product.id);
          console.log('Updated Product ID from Response:', responseData.id);
        });
    }
    this.onFetchProducts();
  });
}

```



## 2.2.6. Hiện thị trang shopping



Hình 2. 16 Giao diện trang shopping của users

(\*)Code chức năng lấy dữ liệu từ firebase về hiển thị trang shopping

Đoạn code được sử dụng để hiển thị danh sách các sản phẩm đã lọc, bao gồm thông tin như hình ảnh, tên và giá sản phẩm. Sử dụng \*ngFor để tạo danh sách sản phẩm dựa trên dữ liệu từ filteredProducts.

```
<div *ngFor="let product of filteredProducts">
  <div class="product-small box m-2">
    <div class="box-image max-w-72">
      <div class="image-none">
        <img
          [src]="product.imageUrl"
          alt="{{ product.name }}"
          class="w-full h-auto"
        >
      </div>
    </div>
  </div>
</div>
```

```

        />
    </div>
</div>
<div class="flex flex-row justify-between max-w-72">
    <div class="flex flex-wrap title-wrapper">
        <a>{{ product.name }}</a>
    </div>
    <div class="price-wrapper text-violet-800 ml-5">
        <span>{{ product.price }}đ</span>
    </div>
</div>
</div>

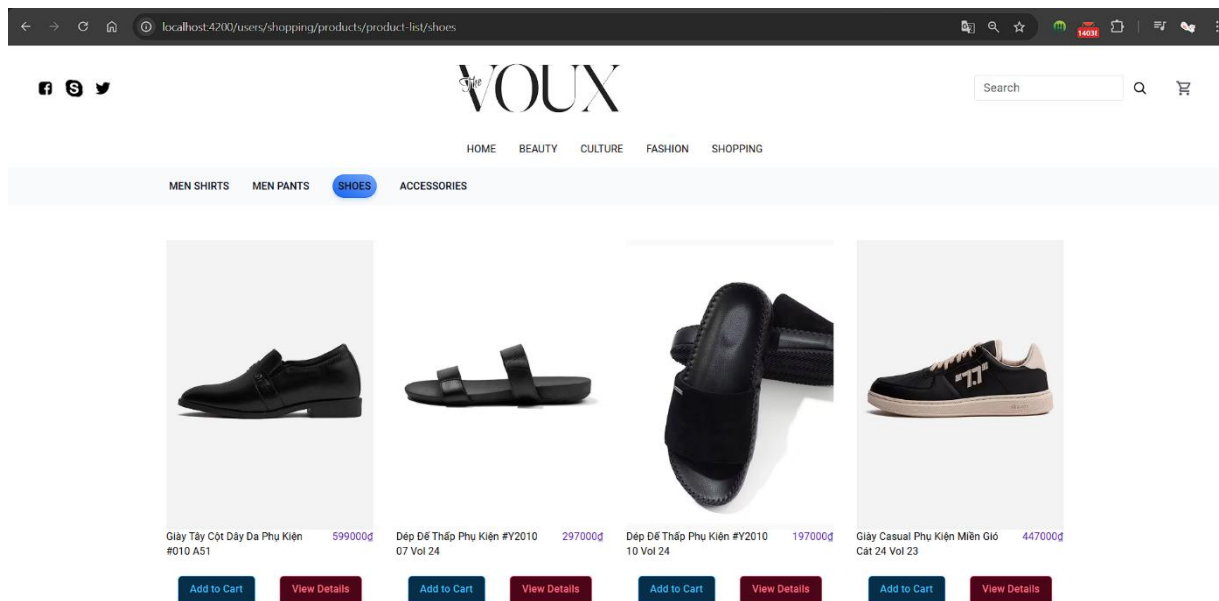
```

```

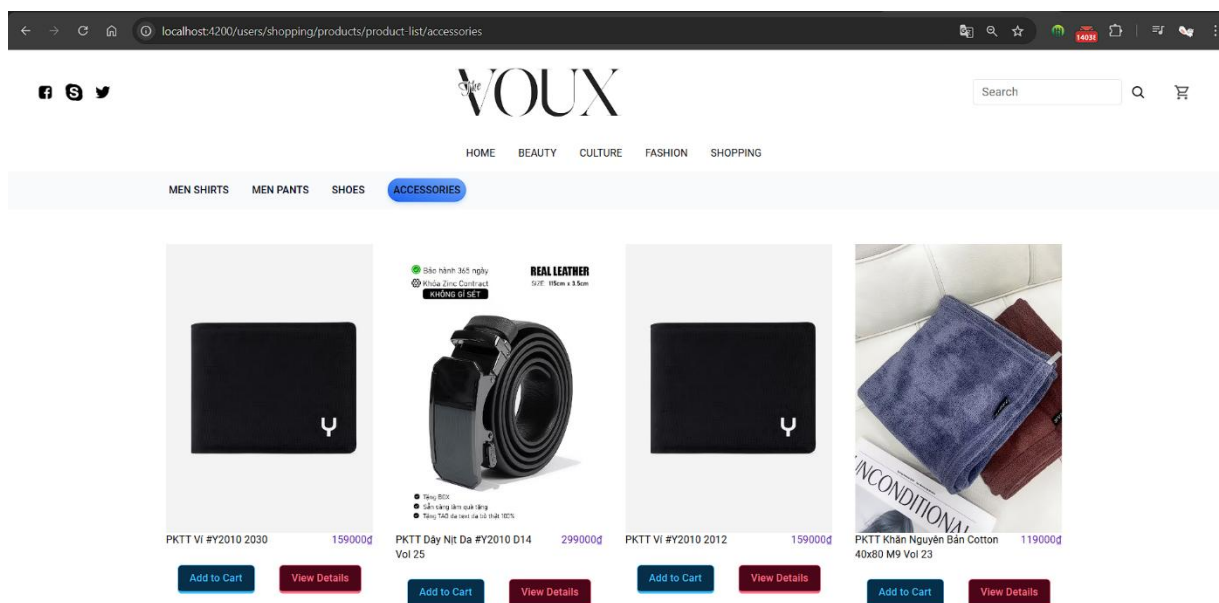
filterProductsByCategory() {
    this.filteredProducts = this.products.filter(
        (product) => product.category === this.category
    );
}

```

### 2.2.7. <sup>2</sup>Hiển thị Product theo Category



Hình 2. 17 Giao diện trang shoes của shopping



Hình 2. 18 Giao diện trang Accessories của shopping

(\*)Code lọc danh sách sản phẩm có thuộc tính Category

Phương thức `filterProductsByCategory` được thiết kế để lọc danh sách sản phẩm theo danh mục cụ thể, giúp hiển thị các sản phẩm phù hợp với nhu cầu người dùng.

Sử dụng phương thức `filter()` để duyệt qua danh sách sản phẩm (`this.products`).

Kiểm tra từng sản phẩm: nếu `product.category` trùng với `this.category`, sản phẩm sẽ được đưa vào danh sách kết quả.

Lưu danh sách kết quả vào biến `filteredProducts`.

```
filterProductsByCategory() {  
  this.filteredProducts = this.products.filter(  
    (product) => product.category === this.category  
  );  
}
```

(\*)Code hiển thị sản phẩm phân theo loại sản phẩm

Đoạn code này hiển thị danh sách các sản phẩm đã lọc, sử dụng `*ngFor` để lặp danh sách `filteredProducts`. Giúp hiển thị danh sách sản phẩm động, dễ duy trì và mở rộng hơn.

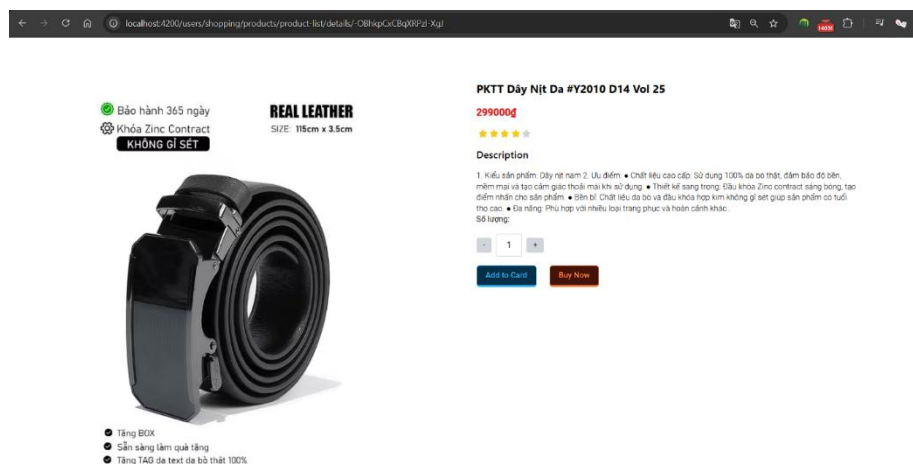
```
<div *ngFor="let product of filteredProducts">  
  <div class="product-small box m-2">  
    <div class="box-image max-w-72">  
      <div class="image-none">  
        <img  
          [src]="product.imageUrl"  
          alt="{{ product.name }}"  
          class="w-full h-auto"  
        />  
      </div>  
    </div>  
    <div class="flex flex-row justify-between max-w-72">  
      <div class="flex flex-wrap title-wrapper">  
        <a>{{ product.name }}</a>  
      </div>  
      <div class="price-wrapper text-violet-800 ml-5">  
        <span>{{ product.price }}đ</span>  
      </div>  
    </div>  
  </div>  
</div>
```

```

    </div>
  </div>
</div>

```

## 2.2.8. Hiển thị Product Detail của Product



Hình 2. 19 Giao diện trang product details

Khi nhấn vào nút view details của sản phẩm nào thì router sẽ điều hướng đúng tới chi tiết sản phẩm đó

```

<button
  routerLink="/users/shopping/products/product-list/details/{{
    product.id
  }}"
  class="bg-rose-950 text-rose-400 border border-rose-400
border-b-4 font-medium overflow-hidden relative px-4 py-2 rounded-md
hover:brightness-150 hover:border-t-4 hover:border-b active:opacity-75
outline-none duration-300 group"
  >
  <span

```

```

class="bg-rose-400 shadow-rose-400 absolute -top-[150%]
left-0 inline-flex w-80 h-[5px] rounded-md opacity-50 group-hover:top-
[150%] duration-500 shadow-[0_0_10px_rgba(0,0,0,0.3)]"
></span>
View Details
</button>

```

Phương thức `filterProductsByCategory` được thiết kế để lọc danh sách sản phẩm theo danh mục cụ thể, giúp hiển thị các sản phẩm phù hợp với nhu cầu người dùng.

Sử dụng phương thức `filter()` để duyệt qua danh sách sản phẩm (`this.products`).

Kiểm tra từng sản phẩm: nếu `product.category` trùng với `this.category`, sản phẩm sẽ được đưa vào danh sách kết quả.

Lưu danh sách kết quả vào biến `filteredProducts`.

```

filterProductsByCategory() {
  this.filteredProducts = this.products.filter(
    (product) => product.category === this.category
  );
}

```

Đoạn code này được dùng để hiển thị thông tin chi tiết sản phẩm bao gồm tên, giá, hình ảnh nhằm cung cấp thông tin trực quan nhất cho người dùng.

Kiểm tra điều kiện để được hiển thị: sử dụng `*ngIf` để đảm bảo chỉ hiển thị khi sản phẩm (`product`) tồn tại. Hiển thị thông tin chi tiết hình ảnh sản phẩm được liên kết với `product.imageUrl`, tên và giá được liên kết với `product.name` và `product.price`...

```

<div class="">
  <div *ngIf="let product"></div>

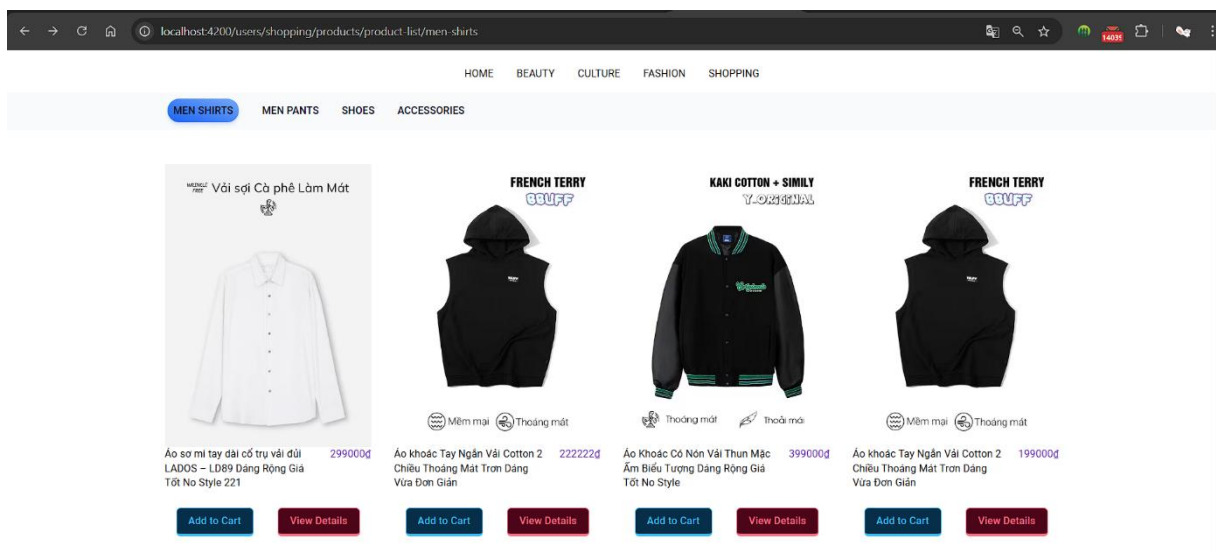
```

```

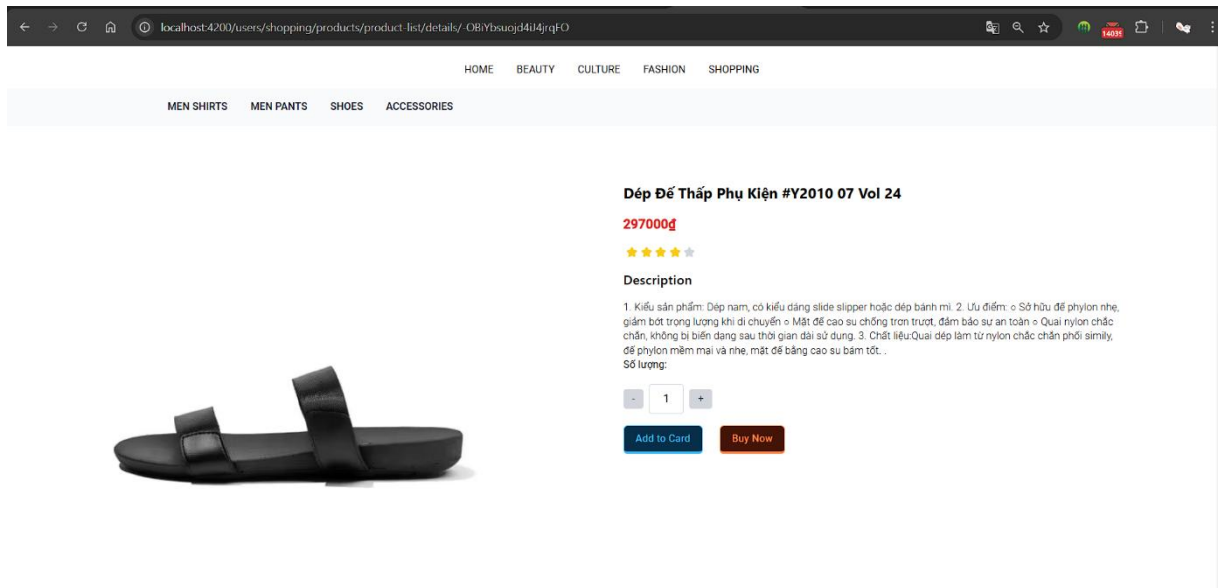
<div class="flex flex-row justify-center px-16 m-12">
  <div class="col-md-6 w-1/2 p-4">
    <img [src]="product?.imageUrl" alt="{{ product?.name }}" />
  </div>
  <div class="col-md-6 w-1/2 p-4 items-center">
    <div class="row">
      <!-- <div class='col-md-3'>Name:</div> -->
      <div class="col-md-6 text-xl font-sans font-bold mt-4">
        {{ product?.name }}
      </div>
    </div>
    <div class="row">
      <div class="col-md-3"></div>
      <div class="col-md-6 text-red-600 text-lg font-bold mt-4">
        {{ product?.price }}đ
      </div>
    </div>
  </div>

```

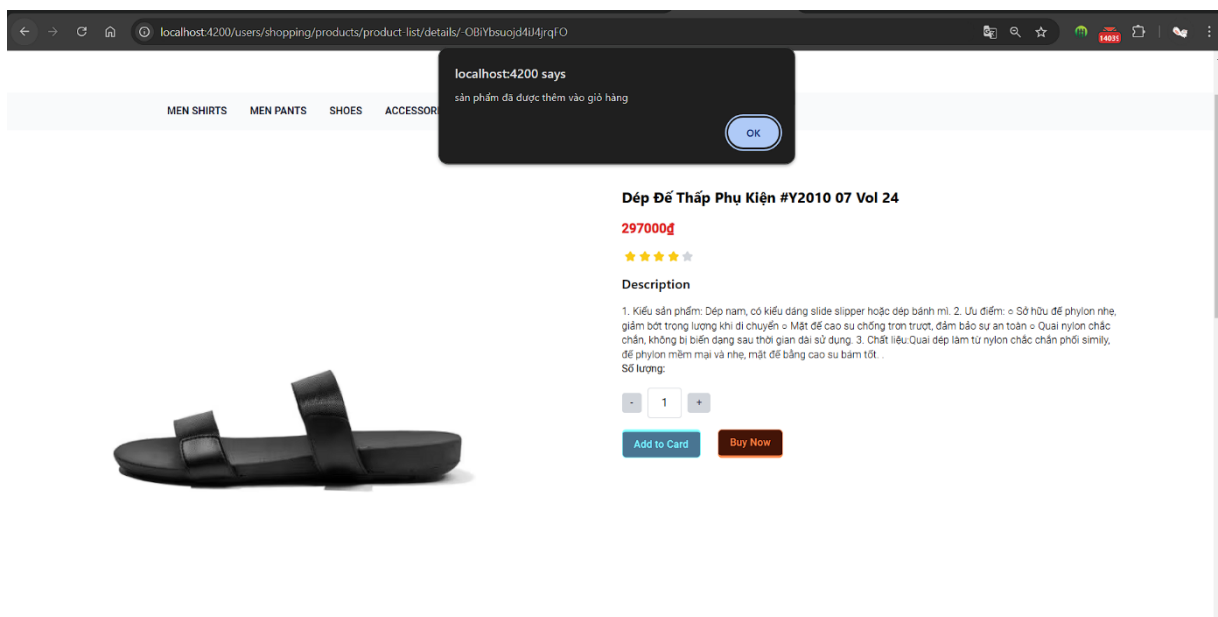
### 2.2.9. Chức năng addToCart từ product, product details



Hình 2. 20 Giao diện trang men shirts có nút add to cart



Hình 2. 21 Giao diện trang product details có hiển thị nút add to cart



Hình 2. 22 Giao diện khi nhấn nút add to cart thì hiển thị thông báo

(\*)Code chức năng add to cart

### product-details.ts

Đoạn code sẽ kiểm tra nếu sản phẩm hợp lệ (this.product tồn tại), tạo đối tượng productToAdd với các thuộc tính của sản phẩm và số lượng. Gọi cartService để thêm sản phẩm vào giỏ. Nếu đã thêm thành công thì sẽ hiển thị thông báo qua Alert.



```

addToCart(){
  if(this.product){
    const productToAdd =
    { ...this.product, quantity: this.productQuantity };
    //this.cartItems.push(productToAdd);
    this.cartService.addToCart(productToAdd);
    alert('sản phẩm đã được thêm vào giỏ hàng');
  }
}

```

## 2.2.10. Hiển thị product vào trang cart

### cart.service.ts

Code này để quản lí trạng thái giỏ hàng bao gồm lưu trữ, cập nhật, đồng bộ dữ liệu với localStorage.

Đoạn code có chức năng lấy dữ liệu giỏ hàng từ localStorage, lưu dữ liệu khi cập nhật trạng thái giỏ hàng, thêm mục mới vào giỏ hàng hoặc cập nhật số lượng nếu đã tồn tại. Đoạn code có sử dụng BehaviorSubject đảm bảo tính đồng bộ và dễ tái sử dụng.

```

private storageKey = "cartItems" ;

private cartSubject = new BehaviorSubject<any[]>(this.getCartItems());
cartItems$ = this.cartSubject.asObservable();
constructor() {}

getCartItems(): any[] {
  const items = localStorage.getItem(this.storageKey);
  return items ? JSON.parse(items) : [];
}

setCartItems(items: any[]): void {
  localStorage.setItem(this.storageKey, JSON.stringify(items));
}

```

```

    }

    addToCart(item: any): void {
        const items = this.getCartItems();
        const existItem = items.find((cartItem) => cartItem.id === item.id
&& cartItem.name === item.name);
        if (existItem) {
            existItem.quantity = (existItem.quantity || 0) + item.quantity;
        } else {
            // item.quantity = 1;
            //const newItem = { ...item, quantity: 1 };
            items.push(item);
        }
        this.setCartItems(items);
        this.cartSubject.next([...items]);
    }

```

Phương thức addItemToCart được tạo ra để thêm một mục hàng vào giỏ và đồng bộ trạng thái giỏ hàng với giao diện người dùng.

Gọi cartService.addToCart(item) để thêm mục hàng vào giỏ.

Gọi loadCart() để tải lại trạng thái giỏ hàng và đảm bảo giao diện luôn hiển thị dữ liệu cập nhật.

```

addItemToCart(item:any){
    this.cartService.addToCart(item);
    this.loadCart();
}

```

Phương thức loadCart() được sử dụng để tải dữ liệu giỏ hàng từ dịch vụ cartService và đồng bộ trạng thái với thuộc tính cartItems.

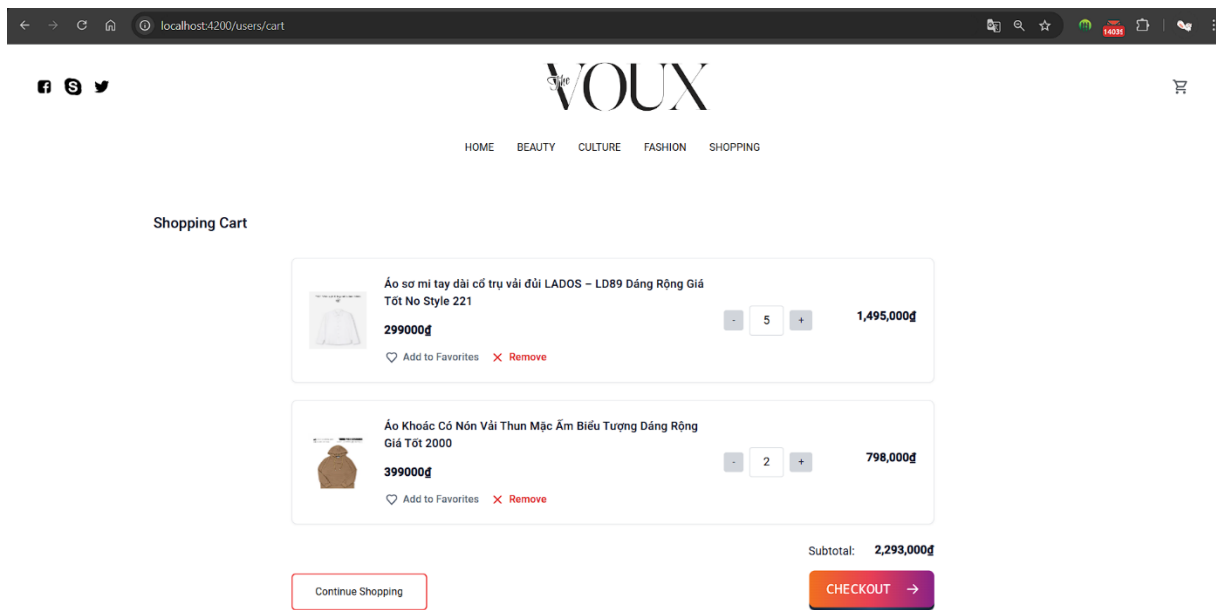
Chức năng của đoạn code là gọi phương thức `getCartItems()` từ `cartService` để lấy danh sách các mục trong giỏ hàng, lưu dữ liệu trả về vào biến `cartItems` cho phép các thành phần khác sử dụng.

```
loadCart(): void {  
  this.cartItems = this.cartService.getCartItems();  
}
```

(\*) Chức năng cập nhật số lượng

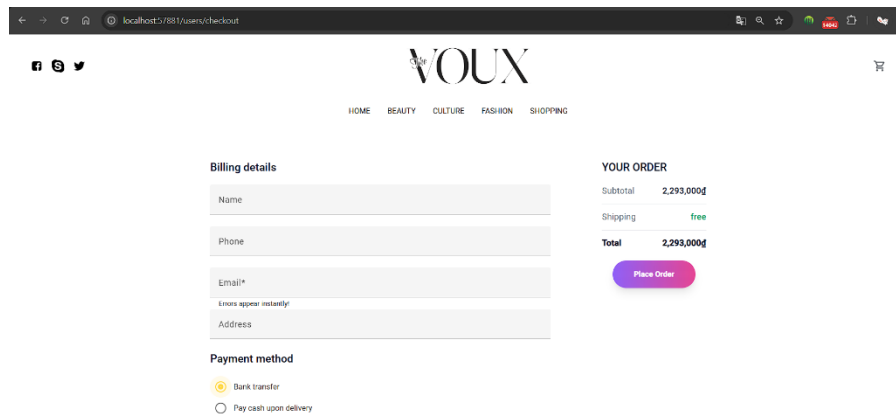
Lấy danh sách các sản phẩm hiện tại có trong giỏ hàng, sau đó sẽ kiểm tra sự tồn tại của sản phẩm. Giá trị cũ của sản phẩm đó sẽ được thay thế bằng giá trị mới. Cuối cùng là lưu lại danh sách đã cập nhật.

```
updateQuantity(index: number, quantity: number): void {  
  const items = this.getCartItems();  
  if (items[index]) {  
    items[index].quantity = quantity;  
    this.setCartItems(items);  
    this.cartSubject.next(items);  
  }  
}
```



Hình 2. 23 Giao diện trang cart

## 2.2.11. Giao diện trang checkout



Hình 2. 24 Giao diện trang checkout

## Tổng kết chương 2

Phía trên là những công việc mà em đã được giao và hoàn thành trong thời gian thực tập tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam. Qua 8 tuần vừa học hỏi vừa cống hiến, em đã tìm hiểu và cài đặt các công cụ, framework mà công ty sử dụng và các nền tảng như Angular, TypeScript, và Tailwind CSS. Em tự hào về những đóng góp của mình vào các dự án, cũng như khả năng vượt qua những thử thách trong môi trường làm việc đầy thách thức không chỉ trong chuyên môn mà qua cả kỹ năng mềm.

Trong thời gian thực tập tại công ty, người hướng dẫn (mentor) của em là anh Trần Đức Vinh đã giúp đỡ em rất nhiều trong việc trở thành một frontend developer của dự án. Không những tận tình chỉ bảo về mặt chuyên môn mà còn cho em nhiều lời khuyên ý nghĩa trong kỹ năng mềm – cách làm việc nhóm, góp phần quan trọng vào việc hoàn thiện bản thân và chuẩn bị cho những bước đi tiếp theo trên con đường sự nghiệp. Khi người mới như em có nhiều câu hỏi thắc mắc, anh vẫn dành thời gian giải đáp dù anh rất bận trong công việc và giúp em có cái nhìn tốt hơn trong việc code. Bên cạnh đó, em xin cảm ơn anh leader Đặng Thanh Phong đã luôn sẵn lòng chia sẻ kinh nghiệm, hỗ trợ em hòa nhập với môi trường làm việc và tạo nên một trải nghiệm thực tập đáng nhớ.

Em xin chân thành cảm ơn anh!

Em tin rằng những kinh nghiệm này sẽ là nền tảng vững chắc cho sự phát triển của bản thân trong lĩnh vực phát triển phần mềm. Việc thực tập mở ra cánh cửa cho những hành trình tiếp theo, nơi em sẽ tiếp tục đóng góp và học hỏi.

## CHƯƠNG 3: KẾT QUẢ THỰC TẬP

Trong thời gian thực tập tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam, tuy ban đầu còn bỡ ngỡ với môi trường doanh nghiệp nhưng sau vài tuần, qua những lần thực hiện Task của Project và những nhận xét của chuyên gia doanh nghiệp, em đã tích lũy được rất nhiều kinh nghiệm quý báu. Từ kỹ năng mềm trong giao tiếp, thái độ làm việc và cách ứng xử với mọi người, cho đến việc ứng dụng những kiến thức được học ở trường vào công việc, em học và nắm được những gì cần làm với vai trò là một DEV. Trên đây là tóm tắt những trải nghiệm của em sau khi được thực tập tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam và những kỹ năng đã học hỏi thêm:

- Làm việc nhóm có hiệu quả
- Sử dụng công cụ cho việc quản lý dự án
- Kỹ năng tự học, tìm tài liệu

Em xin cảm ơn Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam đã tạo điều kiện để em hoàn thành kỳ thực tập. Đặc biệt, em xin cảm ơn sự hỗ trợ, hướng dẫn tận tâm từ anh Trần Đức Vinh đã hướng dẫn tận tình, giúp đỡ em rất nhiều trong việc trở thành một frontend developer của dự án. Không những tận tình chỉ bảo về mặt chuyên môn mà còn cho em nhiều lời khuyên ý nghĩa trong kỹ năng làm việc. Khi người mới như em có nhiều câu hỏi thắc mắc, anh vẫn dành thời gian giải đáp dù anh rất bận trong công việc và giúp em có cái nhìn tốt hơn trong việc code. Em cũng xin cảm ơn các anh leader Đặng Thanh Phong đã hỗ trợ và tạo điều kiện để em học hỏi và phát triển bản thân tại công ty. Thực tập tại công ty đã giúp em xây dựng nền tảng cho sự phát triển cá nhân và sự nghiệp tương lai của mình trong lĩnh vực phát triển phần mềm.

Trong tương lai em hi vọng là bản thân mình đủ năng lực để có cơ hội đóng góp cho ngành công nghệ thông tin và phần mềm cùng các anh chị đồng nghiệp tại Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam hoặc các công ty phần mềm khác.

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập – Tự do – Hạnh phúc

*Tp. Hồ Chí Minh, ngày 22 tháng 11 năm 2024*

## **BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN**

### **Một số thông tin liên hệ**

Họ và tên: Nguyễn Bảo Tân

Ngày sinh: 02/12/2002

Mã số sinh viên: 3120410464

Lớp:DCT1202

Ngành học: Công nghệ thông tin

Email:nguyenbaotan02122002@gmail.com

Điện thoại:0369069809

Chuyên gia doanh nghiệp: Trần Đức Vinh

Email: vinhtran.appvity@gmail.com

Điện thoại:0987380002

Giảng viên hướng dẫn: TS. Cao Thái Phương Thanh

Email: thanh.ctp@sgu.edu.vn

Điện thoại: 0909022966

<b>Tuần</b>	<b>Nội dung thực tập (do chuyên gia của doanh nghiệp giao)</b>	<b>Kết quả thực tập (do chuyên gia của doanh nghiệp đánh giá)</b>
1  Từ ngày 30/ 09/2024  đến ngày 04/ 10/2024	<ul style="list-style-type: none"> <li>• Tìm hiểu về công ty , các nội quy, văn hóa của công ty</li> <li>• Tìm hiểu các hoạt động của các bộ phận doanh nghiệp.</li> <li>• Được hướng dẫn các công việc trong lúc thực tập.</li> <li>• Cài đặt các môi trường cần thiết cho FE.</li> <li>• Cài đặt thêm các tool giúp code nhanh hơn.</li> </ul>	Hoàn thành
2  Từ ngày 07/ 10/ 2024  đến ngày 11/10/2024	<ul style="list-style-type: none"> <li>• Ôn lại kiến thức Database</li> <li>• Đọc lại các kiến thức HTML, SCSS/SASS</li> <li>• Ôn lại về ngôn ngữ Javascript</li> <li>• Tìm hiểu về Framework Angular V16</li> <li>• Tìm hiểu về TypeScript</li> </ul>	Hoàn thành
3  Từ ngày 14/ 10/ 2024  đến ngày 18/ 10/ 2024	<ul style="list-style-type: none"> <li>• Tìm hiểu về firebase và cách sử dụng để liên kết với Angular</li> <li>• Cài đặt NodeJS , Angular V16</li> <li>• Tìm hiểu các component của Angular, hiểu được cách tương tác với nhau giữa các thành</li> </ul>	Thái độ làm việc của bạn có tích cực và mong muốn học hỏi. Hoàn thành đủ yêu cầu.



	<p>phần. Tạo ra một component cơ bản và thực hành</p> <ul style="list-style-type: none"> <li>• Tìm hiểu các loại data binding trong Angular (one-way, two-way) và cách sử dụng chúng để liên kết dữ liệu giữa component và template.</li> <li>• Tìm hiểu về template, service của Angular</li> <li>• Tìm hiểu và thực hành về modules trong angular</li> <li>• Tìm hiểu về Dependency Injection và cách sử dụng trong Angular</li> <li>• Xem tài liệu về Lifecycle Hooks trong Angular</li> </ul>	
<p>4</p> <p>Từ ngày 21/10/2024 đến ngày 25/10/2024</p>	<ul style="list-style-type: none"> <li>• Tìm hiểu về Pipe trong Angular</li> <li>• Tìm hiểu RxJS</li> <li>• Làm các task về Creation Operators, Transformation Operators, Filtering, Combination, Error Handling và conditional trong RxJS</li> <li>• Tìm hiểu về Router</li> <li>• Làm các task về lazy-load module, guards-resolvers</li> <li>• Convert file HTML sang Angular</li> </ul>	<p>Hoàn thành task đúng thời hạn. Tuy nhiên task có cần thời gian để trao đổi thêm.</p>
<p>5</p> <p>Từ ngày 28/10/2024 đến ngày 01/11/2024</p>	<ul style="list-style-type: none"> <li>• Tìm hiểu template driven forms, reactive forms</li> <li>• Sử dụng query, fragment trong route</li> <li>• HTTP Request &amp; Response</li> <li>• Thiết lập Firebase   Angular HTTP</li> <li>• Code 1 trang dashboard post bài lên firebase</li> </ul>	<p>Có tiến bộ, hoàn thành công việc.</p>

<p>6</p> <p>Từ ngày 04/11/2024 đến ngày 08/11/2024</p>	<ul style="list-style-type: none"> <li>• Authentication &amp; Authorization</li> <li>• Route Guard</li> <li>• Understanding Angular Modules</li> <li>• Understanding Signals in Angular</li> <li>• Module &amp; Lazy Loading</li> <li>• Code trang home-page, beauty dùng tailwind css, angular</li> <li>• Code trang products dùng router link để chuyển hướng</li> <li>• Code chức năng thêm, sửa, xóa trang products cho admin , đẩy dữ liệu lên firebase realtime database</li> <li>• Học quản lý trạng thái (State Management)</li> </ul>	<p>Hoàn thành</p>
<p>7</p> <p>Từ ngày 11/11/2024 đến ngày 15/11/2024</p>	<ul style="list-style-type: none"> <li>• Đọc tài liệu về Jira Angular, Rxjs Angular</li> <li>• Code giao diện, chức năng của trang product details</li> <li>• Code giao diện trang cart và chức năng thêm , hiển thị sản phẩm vào giỏ</li> <li>• Xử lý lại form , chức năng add, edit của products lên firebase</li> <li>• Code giao diện trang checkout</li> </ul>	<p>Hoàn thành công việc tuy nhiên vẫn phát sinh lỗi trong quá trình làm việc. Có khả năng học hỏi tốt và tích cực cống hiến.</p>
<p>8</p> <p>Từ ngày 18/11/2024</p>	<ul style="list-style-type: none"> <li>• Đọc và tham khảo thêm tài liệu angular nâng cao</li> <li>• Xử lý lại giao diện cho trang web</li> <li>• Hoàn thành báo cáo thực tập</li> </ul>	<p>Hoàn thành tốt</p>

đến ngày 22/ 11/ 2024		
-----------------------------	--	--

**Chuyên gia doanh nghiệp hướng dẫn thực tập**

(Ký tên và ghi họ tên)

**ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập – Tự do – Hạnh phúc**

**KHOA CNTT**

**BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP**  
**(do chuyên gia doanh nghiệp đánh giá).**

**Họ và tên sinh viên :** NGUYỄN BẢO TÂN

**Ngày sinh :** 02/12/2002

**Mã số sinh viên :** 3120410464

**Lớp :** DCT1202

**Thời gian thực tập :** 30/09/2024 – 23/11/2024

**Doanh nghiệp thực tập :** Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam

**Địa chỉ doanh nghiệp :** Lầu 8, Tòa Nhà Kicotrans, Số 46, Đường Bạch Đằng 2, Phường 2, Quận Tân Bình, Thành phố Hồ Chí Minh, Việt Nam

**Chuyên gia doanh nghiệp hướng dẫn :** TRẦN ĐỨC VINH

**I. ĐÁNH GIÁ VỀ QUÁ TRÌNH THỰC TẬP**

<b>S T T</b>	<b>Nội dung đánh giá</b>	<b>ĐIỂM</b>		
		<b>0</b>	<b>0.5</b>	<b>1</b>
1	Khả năng thực hành			
2	Khả năng làm việc nhóm			
3	Tính thân thiện			
4	Tính năng động			
5	Tính thần sáng tạo			
6	Chấp hành nội quy cơ quan			
7	Giờ giấc làm việc			
8	Phương pháp làm việc			
9	Khối lượng công việc			
10	Báo cáo thực tập tốt nghiệp			

(theo thang điểm 10).

**II. CÁC ĐÁNH GIÁ KHÁC:**

.....  
 .....

.....  
.....  
**III. KẾT QUẢ TỔNG HỢP:**

**Điểm tổng cộng :**

**XÁC NHẬN CỦA DOANH NGHIỆP**  
*(đóng mộc tròn của doanh nghiệp, họ tên, ký tên)*

**Chuyên gia hướng dẫn**  
*(Ký và ghi họ tên)*

KHOA CNTT

**PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP**  
(do giảng viên hướng dẫn đánh giá)

**Họ và tên sinh viên :** NGUYỄN BẢO TÂN

**Ngày sinh :** 02/12/2002

**Mã số sinh viên :** 3120410464

**Lớp :** DCT1202

**Thời gian thực tập :** 30/09/2024 – 23/11/2024

**Doanh nghiệp thực tập :** Công Ty TNHH MTV Hệ Thống APPVITY Việt Nam

**Địa chỉ doanh nghiệp :** Lầu 8, Tòa Nhà Kicotrans, Số 46, Đường Bạch Đằng 2,  
Phường 2, Quận Tân Bình, Thành phố Hồ Chí Minh, Việt Nam

**Chuyên gia doanh nghiệp hướng dẫn :** TRẦN ĐỨC VINH

**I. ĐIỂM CỦA CHUYÊN GIA DOANH NGHIỆP :**

(thang điểm 10)

**II. ĐIỂM CỦA GIẢNG VIÊN HƯỚNG DẪN:**

(thang điểm 10)

**III. ĐIỂM TỔNG KẾT:**

(trung bình cộng 2 cột điểm trên, thang điểm 10)

**Xếp loại :**

TP Hồ Chí Minh, ngày      tháng      năm

**Giảng viên hướng dẫn**

### **Tổng kết chương 3**

Qua những đánh giá khách quan từ phía doanh nghiệp và của giảng viên hướng dẫn, em sẽ ghi nhớ để cải thiện cũng như phát triển bản thân hơn sau này. Những nhận xét trên giúp em nhìn nhận rõ hơn được những ưu khuyết điểm mà mình mắc phải khi làm việc trong một môi trường chuyên nghiệp, định hướng bản thân được rõ ràng hơn.

## **CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ**

### **(\*) Kết luận**

Sau thời gian được thực tập, dưới sự chỉ dẫn và hỗ trợ của các anh chuyên gia doanh nghiệp APPVITY, em đã học hỏi được rất nhiều kinh nghiệm và kiến thức chuyên môn. Thứ nhất, em đã hiểu được quy trình phát triển phần mềm của doanh nghiệp, cách doanh nghiệp hoạt động như thế nào và biết được cách làm việc trong môi trường doanh nghiệp ra sao. Thứ hai, em đã rèn luyện được thêm kỹ năng chuyên môn rất cần thiết để làm việc trong hiện tại và cả tương lai. Ngoài ra em cũng được rèn luyện về kỹ năng làm việc nhóm, xử lý vấn đề, tương tác giữa các thành viên trong nhóm khi làm project và họp team. Qua đó, sau những mặt đạt được, em cũng cần cải thiện nhiều về kỹ năng mềm, kỹ năng chuyên môn, như là phải nâng cao chất lượng sản phẩm làm ra nhiều hơn nữa để đáp ứng nhu cầu khách hàng.

Ngoài ra một điều quan trọng hơn nữa khi làm việc tại công ty là một thái độ làm việc nghiêm túc và luôn luôn trên tinh thần phát triển bản thân. Quy tắc ứng xử đối với đồng nghiệp, khả năng làm việc nhóm và kỹ năng giao tiếp cũng là một chìa khóa để thành công trong công việc.



# TÀI LIỆU THAM KHẢO

[1] Angular, "Angular Documentation," [Online].

Available at : <https://v16.angular.io/docs> (Accessed: Oct-01-2024).

[2] Angular University, "Angular University," [Online]. Available at: <https://angular-university.io/> (Accessed: Oct-10-2024).

[3] Angular Vietnam, "100 Days of Angular," [Online]. Available: <https://github.com/angular-vietnam/100-days-of-angular/tree/master> (Accessed: Nov-01-2024).

[4] Tailwind CSS, "Tailwind CSS - Rapidly build modern websites without ever leaving your HTML," [Online]. Available: <https://tailwindcss.com/> (Accessed: Sep-30-2024).

[5] W3Schools, "AngularJS Tutorial," [Online]. Available: <https://www.w3schools.com/angular/default.asp> (Accessed: Oct-10-2024).

[6] A. Murray, Learning Angular - Fourth Edition: A no-nonsense guide to building web applications with Angular, 4th ed., Packt Publishing, 2023.

[7] A. Mevada and M. Gamboa, Angular Projects - Third Edition: Build modern web apps in Angular 16 with 10 different projects and cutting-edge technologies, 3rd ed., Packt Publishing, 2023.

[8] A. Murray, Learning Angular - Third Edition: A no-nonsense beginner's guide to building web applications with Angular 10 and TypeScript, 3rd ed., Packt Publishing, 2020.