TRƯỜNG ĐẠI HỌC TRÀ VINH

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**BÁO CÁO KẾT THÚC HỌC PHẦN**

**CÔNG NGHỆ PHẦN MỀM**

**NĂM HỌC: 2022-2023**

**LỚP DA20TTA**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn:  Nguyễn Bảo Ân | Nhóm sinh viên thực hiện:  Trần Quang Tiến - 110120077  Trần Trọng Nhân - 11020051  Lâm Ngọc Hân - 110120026 |

*Trà Vinh, tháng 7 năm 2023*

TRƯỜNG ĐẠI HỌC TRÀ VINH

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**ĐỀ TÀI**

**SpringBoot-Angular7-Online-Shopping-Store**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn:  Nguyễn Bảo Ân | Nhóm sinh viên thực hiện:  Trần Quang Tiến - 110120077  Trần Trọng Nhân - 11020051  Lâm Ngọc Hân - 110120026 |

*Trà Vinh, tháng 7 năm 2023*

**LỜI CẢM ƠN**

Chúng tôi xin cảm ơn thầy Nguyễn Bảo Ân. Nhờ sự giúp đỡ và hướng dẫn tận tình của thầy từ đầu đến cuối đồ án, chúng tôi đã hoàn thành đồ án đúng thời hạn và tích luỹ được một vốn kiến thức quý báu.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài này nhưng do hạn chế về trình độ cũng như kiến thức chuyên môn nên trong quá trình nghiên cứu, đánh giá và trình bày đề tài, chúng tôi còn nhiều thiếu sót. Chúng tôi rất mong được sự quan tâm, góp ý của các thầy cô bộ môn để khóa luận của tôi được hoàn thiện và hoàn thiện hơn.

Chúng tôi xin chân thành cảm ơn.

**Sinh viên thực hiện**

Trần Trọng Nhân

Lâm Ngọc Hân

Trần Quang Tiến

MỤC LỤC

[LỜI MỞ ĐẦU 7](#_Toc140180324)

[CHƯƠNG 1: GIỚI THIỆU 8](#_Toc140180325)

[1.1 Công nghệ phần mềm 8](#_Toc140180326)

[1.2 Tổng quan về dự án 8](#_Toc140180327)

[1.3 Mục tiêu 9](#_Toc140180328)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU 10](#_Toc140180329)

[2.1 Miêu tả yêu cầu dự án 10](#_Toc140180330)

[2.2 Các chức năng và tính năng 11](#_Toc140180331)

[CHƯƠNG 3: THIẾT KẾ KIẾN TRÚC 18](#_Toc140180332)

[3.1 Kiến trúc hệ thống 18](#_Toc140180333)

[3.1.1 Microservices? 18](#_Toc140180334)

[3.2 Thiết kế giao diện người dùng 20](#_Toc140180335)

[3.3 Thiết kế cơ sở dữ liệu 20](#_Toc140180336)

[CHƯƠNG 4: QUY TRÌNH PHÁT TRIỂN 22](#_Toc140180337)

[4.1 Agile 22](#_Toc140180338)

[4.1.1 Agile là gì? 22](#_Toc140180339)

[4.1.2 Các giá trị và nguyên tắc của Agile 22](#_Toc140180340)

[4.1.3 Phương pháp Agile 23](#_Toc140180341)

[4.2 Scrum 24](#_Toc140180342)

[4.2.1 Scrum là gì? 24](#_Toc140180343)

[4.2.2 Khái niệm cơ bản của Scrum 24](#_Toc140180344)

[4.2 Sơ đồ quy trình phát triển 26](#_Toc140180345)

[CHƯƠNG 5: TRIỂN KHAI 27](#_Toc140180346)

[5.1 Docker 27](#_Toc140180347)

[5.2 Các bước triển khai 28](#_Toc140180348)

[5.2.1 Triển khai ứng dụng lên Localhost 28](#_Toc140180349)

[5.2.2 Triển khai ứng dụng trên docker 30](#_Toc140180350)

[5.2.3 Triển khai lên cloud 35](#_Toc140180351)

[CHƯƠNG 6: KẾT LUẬN 41](#_Toc140180352)

[6.1 Kết quả đạt được 41](#_Toc140180353)

[6.2 Khó khăn và bài học 42](#_Toc140180354)

[6.3 Hướng phát triển 42](#_Toc140180355)

[CHƯƠNG 7: TÀI LIỆU THAM KHẢO 43](#_Toc140180356)

DANH MỤC HÌNH ẢNH

[Hình 1. Danh sách sản phẩm 12](#_Toc140171155)

[Hình 2. Lọc sản phẩm theo loại 12](#_Toc140171156)

[Hình 3. Chi tiết sản phẩm 13](#_Toc140171157)

[Hình 4. Thêm sản phẩm vào giỏ hàng 13](#_Toc140171158)

[Hình 5 Điều chỉnh số lượng sản phẩm 14](#_Toc140171159)

[Hình 6. Điều chỉnh thông tin cá nhân 14](#_Toc140171160)

[Hình 7. Điều chỉnh thông tin sản phẩm 15](#_Toc140171161)

[Hình 8. Thanh toán sản phẩm 15](#_Toc140171162)

[Hình 9. Quản lý thông tin sản phẩm 16](#_Toc140171163)

[Hình 10. Cập nhật trạng thái đơn hàng 16](#_Toc140171164)

[Hình 11. Giao diện Quản trị viên quản lý đơn hàng 17](#_Toc140171165)

[Hình 12. Thiết kế giao diện người dùng 21](#_Toc140171166)

[Hình 13. Thiết kế mô hình dữ liệu mức vật lý 21](#_Toc140171167)

[Hình 14 Quy trình phát triển của Scrum 27](#_Toc140171168)

[Hình 15. Sơ đồ quy trình phát triển 27](#_Toc140171169)

[Hình 16. Cài đặt một dự án Maven vào kho lưu trữ Maven cục bộ hoặc từ xa 29](#_Toc140171170)

[Hình 17. Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm 30](#_Toc140171171)

[Hình 18. Chạy ứng dụng Angular trong môi trường phát triển 30](#_Toc140171172)

[Hình 19. Ứng dụng sau khi triển khai trên Localhost 31](#_Toc140171173)

[Hình 20. Đóng gói một dự án thành một file artifact 31](#_Toc140171174)

[Hình 21. Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm 32](#_Toc140171175)

[Hình 22. Xây dựng (build) một phiên bản sản phẩm (production build) 32](#_Toc140171176)

[Hình 23. Các images có trong máy 33](#_Toc140171177)

[Hình 24. Các Container có trong máy 33](#_Toc140171178)

[Hình 25. Đăng nhập vào dockerhub 34](#_Toc140171179)

[Hình 26. Xem các image có trên máy cục bộ 34](#_Toc140171180)

[Hình 27. Đặt lại tag cho các image 34](#_Toc140171181)

[Hình 28. Xem lại các images 34](#_Toc140171182)

[Hình 29. Đẩy image postgres lên dockerhub 35](#_Toc140171183)

[Hình 30. Đẩy image frontend lên dockerhub 35](#_Toc140171184)

[Hình 31. Đẩy các image backend lên dockerhub 35](#_Toc140171185)

[Hình 32. Sau khi triển khai trên DockerHub 35](#_Toc140171186)

[Hình 33. Đăng ký tài khoản tại cloudfly.vn 36](#_Toc140171187)

[Hình 34. Dùng putty để kết nối với máy chủ sau đó đăng nhập vào máy chủ 36](#_Toc140171188)

[Hình 35. Cập nhật và thêm kho ảnh 37](#_Toc140171189)

[Hình 36. Kéo các images đã đẩy lên dockerhub về 38](#_Toc140171190)

[Hình 37. Kéo các images đã đẩy lên dockerhub về 38](#_Toc140171191)

[Hình 38. Kéo các images đã đẩy lên dockerhub về 39](#_Toc140171192)

[Hình 39. Chạy các container của các images 39](#_Toc140171193)

[Hình 40. Chạy các container của các images 40](#_Toc140171194)

[Hình 41. Chạy các container của các images 40](#_Toc140171195)

[Hình 42. Xem các container đang chạy 41](#_Toc140171196)

[Hình 43. Frontend đang chạy ở cổng 8008 41](#_Toc140171197)

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Trà Vinh, ngày …, tháng …, năm 2023

Giảng viên hướng dẫn

**Nguyễn Bảo Ân**

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

# LỜI MỞ ĐẦU

Trong quá trình chuyển đổi số đang diễn ra mạnh mẽ ở Việt Nam và thế giới, nhu cầu nhân lực ngành Công nghệ thông tin cho thị trường toàn cầu và trong nước đều tăng cao. Công nghệ Phần mềm đang đóng vai trò vô cùng to lớn trong đời sống của con người.

Hiện nay Việt Nam ngày càng nổi lên là một trong những quốc gia xuất khẩu công nghệ thông tin ra thế giới. Với nguồn nhân lực chất lượng cao, nếu được đào tạo các kiến thức, kỹ năng toàn cầu sẽ giúp Việt Nam nắm bắt được cơ hội của cuộc cách mạng công nghệ thông tin đang diễn ra.

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 Công nghệ phần mềm

Công nghệ phần mềm là chuyên ngành nghiên cứu về hệ thống kỹ thuật, phần mềm máy tính. Cụ thể, ngành học này tập trung nghiên cứu về các hạ tầng phần mềm, cơ sở dữ liệu cũng như sự phát triển của các ứng dụng và hệ thống. Ngành tập trung vào việc xây dựng các ứng dụng giúp nâng cao hiệu quả hoạt động của các doanh nghiệp và chất lượng đời sống con người.

Là sự áp dụng một các tiếp cận hệ thống, có kỹ luật. Và định lượng được cho việc phát triển, sử dụng và bảo trì phần mềm. Nó bao gồm các kiến thức, công cụ và các phương pháp cho việc định nghĩa các phần mềm. Ngoài ra còn có các nhiệm vụ như xây dựng, kiểm tra,…phần mềm.

Ngoài ra, công nghệ phần mềm còn yêu cầu các kiến thức của các lĩnh vực như kỹ thuật máy tính, khoa học máy tính, toán học, quản lý dự án, quản lý chất lượng…

## 1.2 Tổng quan về dự án

Dự án "SpringBoot-Angular7-Online-Shopping-Store" là một ứng dụng web thương mại điện tử được phát triển bằng Spring Boot (backend) và Angular 7 (frontend). Dự án này cung cấp một giao diện người dùng tương tác để mua sắm trực tuyến.

Dự án có một số tính năng cơ bản như:

1. Đăng nhập và đăng ký người dùng: Người dùng có thể tạo tài khoản mới hoặc đăng nhập vào hệ thống.
2. Xem danh sách sản phẩm: Người dùng có thể xem danh sách sản phẩm có sẵn để mua hàng.
3. Xem chi tiết sản phẩm: Người dùng có thể xem thông tin chi tiết của sản phẩm, bao gồm hình ảnh, giá cả, mô tả, v.v.
4. Thêm sản phẩm vào giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng để tiến hành thanh toán.
5. Quản lý đơn hàng: Người dùng có thể xem và quản lý các đơn hàng đã đặt.
6. Quản lý sản phẩm và danh mục: Quản trị viên có thể thêm, sửa đổi hoặc xóa sản phẩm và danh mục.

## 1.3 Mục tiêu

\* Giao diện người dùng:

- Người dùng có thể xem danh sách sản phẩm có sẵn để mua hàng.

- Người dùng có thể xem thông tin chi tiết của sản phẩm, bao gồm hình ảnh, giá cả, mô tả và đánh giá.

- Người dùng có thể thêm sản phẩm vào giỏ hàng và cập nhật số lượng sản phẩm trong giỏ hàng.

- Người dùng có thể xem giỏ hàng và tiến hành thanh toán các sản phẩm.

\* Quản lý sản phẩm:

- Quản trị viên có thể thêm, sửa đổi và xóa thông tin về sản phẩm, bao gồm danh mục, mô tả và hình ảnh.

- Quản trị viên có thể xem danh sách sản phẩm hiện có và chỉnh sửa thông tin của chúng.

\* Quản lý đơn hàng:

- Quản trị viên có thể xem danh sách đơn hàng và trạng thái của chúng.

- Quản trị viên có thể cập nhật trạng thái đơn hàng, ví dụ như xác nhận, vận chuyển hoặc hoàn thành.

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

## 2.1 Miêu tả yêu cầu dự án

\* Giao diện người dùng:

- Người dùng có thể xem danh sách sản phẩm có sẵn để mua hàng.

- Người dùng có thể xem chi tiết sản phẩm, bao gồm hình ảnh, giá cả, mô tả, v.v.

- Người dùng có thể thêm sản phẩm vào giỏ hàng để tiến hành thanh toán.

- Người dùng có thể xem và quản lý các đơn hàng đã đặt.

\* Quản lý sản phẩm và danh mục:

- Quản trị viên có thể thêm, sửa đổi và xóa thông tin sản phẩm, bao gồm danh mục, chi tiết và hình ảnh.

- Quản trị viên có thể quản lý danh mục sản phẩm để phân loại sản phẩm một cách hợp lý.

\* Quản lý người dùng:

- Người dùng có thể đăng ký tài khoản mới và đăng nhập vào hệ thống để thực hiện mua sắm và quản lý tài khoản cá nhân.

\* Giỏ hàng và thanh toán:

- Người dùng có thể thêm sản phẩm vào giỏ hàng, chỉnh sửa số lượng sản phẩm và xóa sản phẩm trong giỏ hàng.

- Người dùng có thể thanh toán các sản phẩm trong giỏ hàng.

\* Quản lý đơn hàng và lịch sử giao dịch:

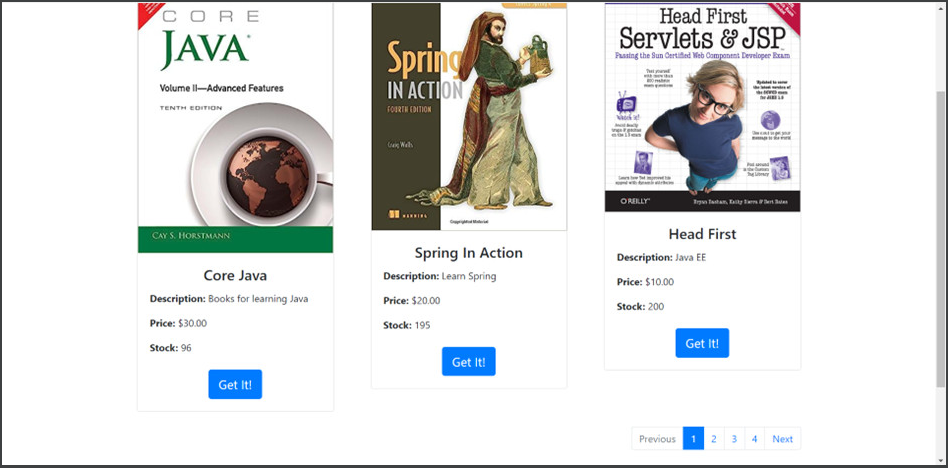
- Người dùng có thể xem trạng thái đơn hàng và lịch sử giao dịch của mình.

- Quản trị viên có thể xem và quản lý các đơn hàng, bao gồm xác nhận, hủy đơn hàng.

## 2.2 Các chức năng và tính năng

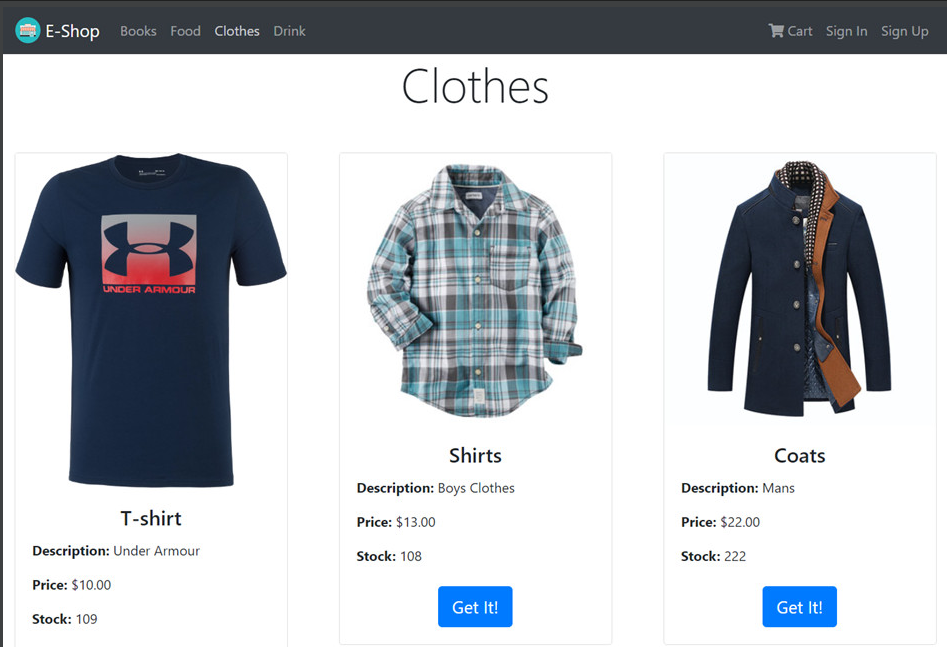
**User stories**

\* Người dùng muốn xem danh sách sản phẩm:



Hình : Danh sách sản phẩm

\* Người dùng muốn lọc sản phẩm theo loại (Book, Food, Clothes, Drink):



Hình 2: Lọc sản phẩm theo loại

\* Người dùng muốn xem chi tiết sản phẩm:



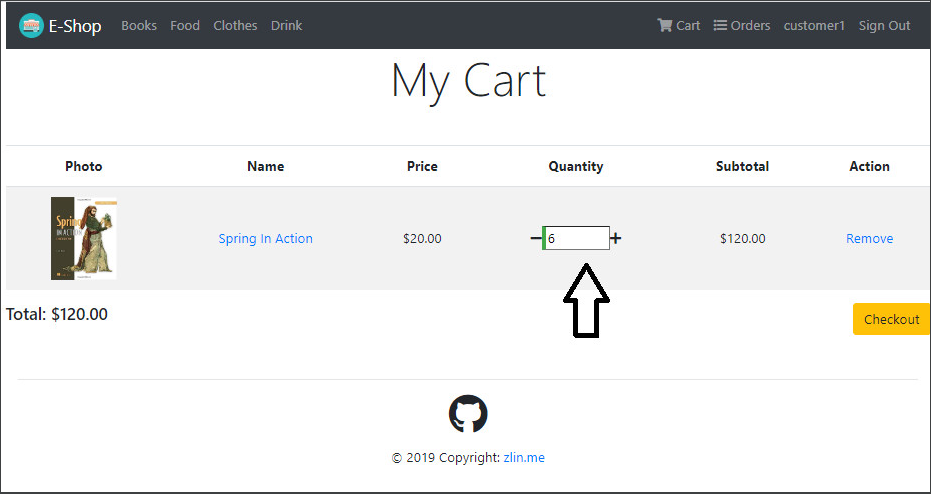
Hình 3. Chi tiết sản phẩm

\* Người dùng muốn thêm sản phẩm vào giỏ hàng:



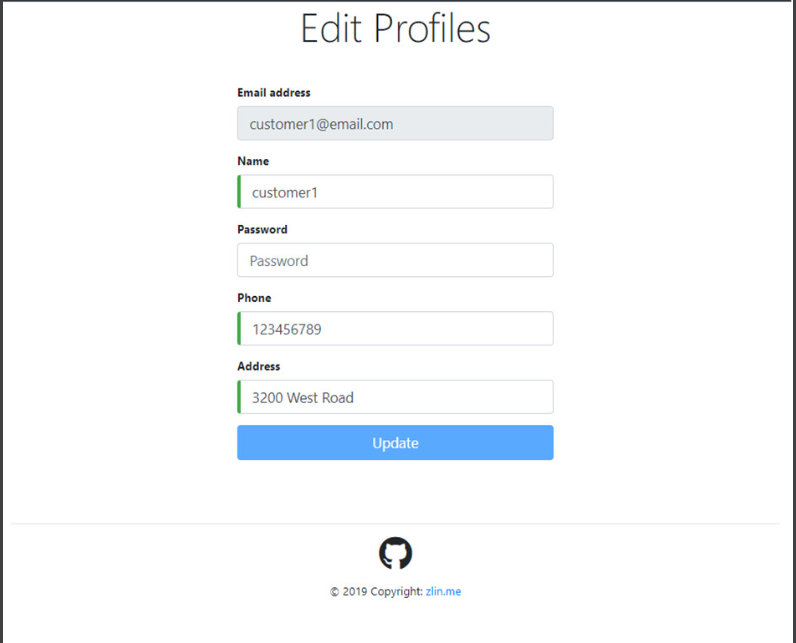
Hình 4. Thêm sản phẩm vào giỏ hàng

\* Người dùng muốn điều chỉnh số lượng sản phẩm:



Hình 5 Điều chỉnh số lượng sản phẩm

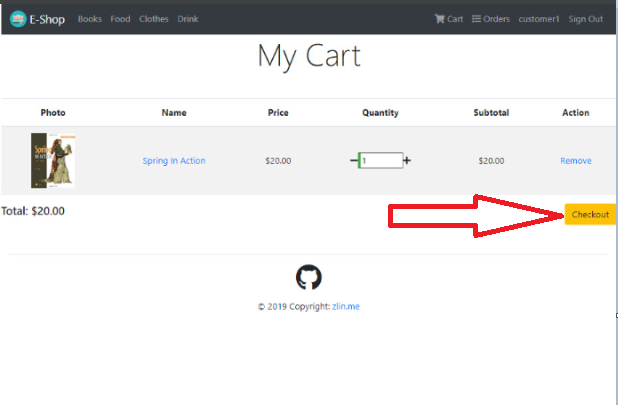
\* Người dùng muốn điều chỉnh thông tin cá nhân trước khi thanh toán:



Hình 6. Điều chỉnh thông tin cá nhân

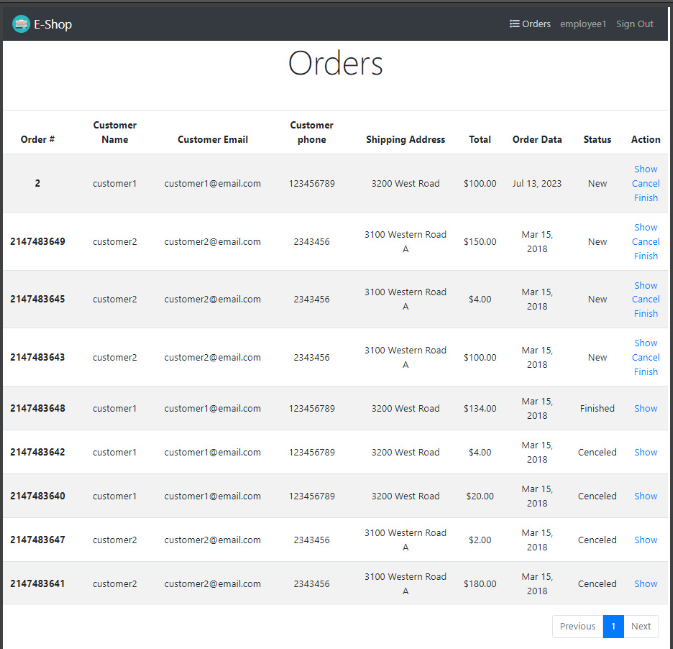
\* Người dùng điều chỉnh thông tin sản phẩm:

\* Người dùng muốn thanh toán sản phẩm:



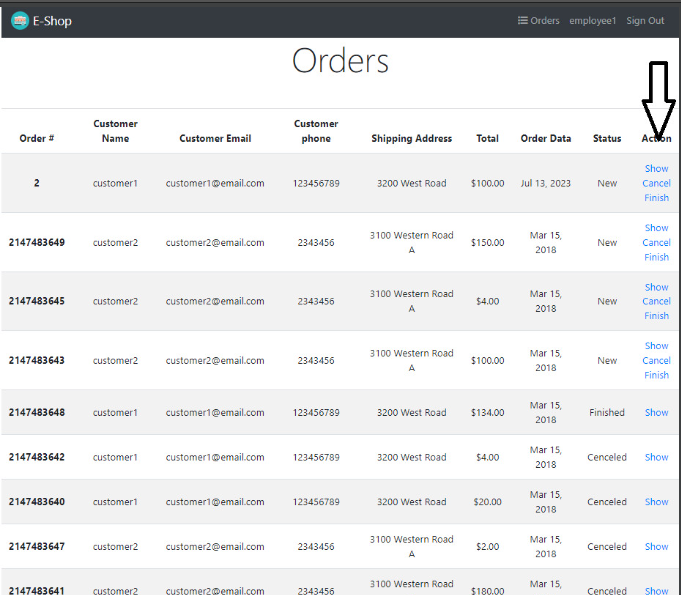
Hình 7. Thanh toán sản phẩm

\* Quản trị viên muốn quản lý thông tin sản phẩm:



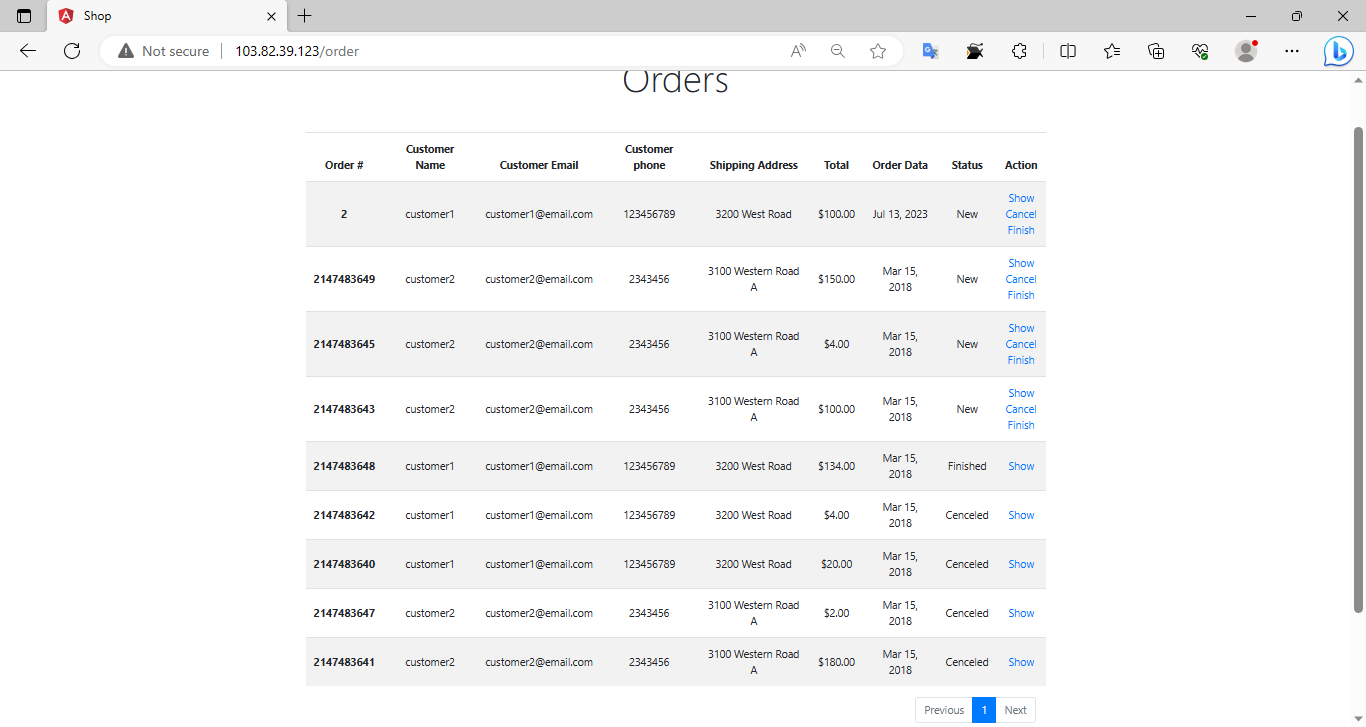
Hình 8. Quản lý thông tin sản phẩm

\* Người dùng muốn cập nhật trạng thái đơn hàng:



Hình 9. Cập nhật trạng thái đơn hàng

\* Quản trị viên muốn quản lý đơn hàng:



Hình 10. Giao diện Quản trị viên quản lý đơn hàng

**Features**

\* Quản lý sản phẩm:

- Thêm sản phẩm mới: Quản trị viên có thể thêm thông tin về sản phẩm mới, bao gồm tên, danh mục, giá, mô tả và hình ảnh.

- Sửa đổi sản phẩm: Quản trị viên có thể chỉnh sửa thông tin của sản phẩm hiện có như giá, mô tả, hình ảnh, v.v.

- Xóa sản phẩm: Quản trị viên có khả năng xóa sản phẩm khỏi danh sách sản phẩm.

\* Xem danh sách sản phẩm:

- Hiển thị danh sách sản phẩm: Người dùng có thể xem danh sách các sản phẩm có sẵn để mua hàng.

- Lọc sản phẩm: Người dùng có thể lọc sản phẩm theo danh mục, giá, đánh giá, v.v.

\* Xem chi tiết sản phẩm:

- Hiển thị thông tin chi tiết sản phẩm: Người dùng có thể xem thông tin chi tiết của sản phẩm, bao gồm hình ảnh, giá, mô tả và thông tin liên hệ.

\* Quản lý giỏ hàng:

- Thêm sản phẩm vào giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng để chuẩn bị cho quá trình thanh toán.

- Xóa sản phẩm khỏi giỏ hàng: Người dùng có khả năng xóa sản phẩm khỏi giỏ hàng.

\* Cập nhật số lượng sản phẩm:

- Người dùng có thể điều chỉnh số lượng sản phẩm trong giỏ hàng.

\* Quản lý đơn hàng:

- Xem danh sách đơn hàng: Quản trị viên có thể xem danh sách đơn hàng và thông tin chi tiết của chúng.

- Cập nhật trạng thái đơn hàng: Quản trị viên có khả năng cập nhật trạng thái đơn hàng, ví dụ như xác nhận, vận chuyển và hoàn tất.

# CHƯƠNG 3: THIẾT KẾ KIẾN TRÚC

## 3.1 Kiến trúc hệ thống

### 3.1.1 Microservices?

Microservices là một kiến trúc phần mềm cho việc phát triển ứng dụng dựa trên việc chia nhỏ thành các dịch vụ nhỏ, kết nối lỏng lẻo và có thể triển khai độc lập. Trong phương pháp này, một ứng dụng được chia thành nhiều dịch vụ nhỏ, tự đủ và tự chứa có thể được phát triển, triển khai và mở rộng độc lập.

Mỗi microservice tập trung vào một khả năng kinh doanh cụ thể và chịu trách nhiệm thực hiện một chức năng xác định. Những dịch vụ này giao tiếp với nhau thông qua các giao thức nhẹ như HTTP hoặc hệ thống tin nhắn để thực hiện chức năng tổng thể của ứng dụng.

Dưới đây là một số đặc điểm và nguyên tắc chính của microservices:

**1. Phân tán và độc lập**: Microservices được thiết kế để độc lập và tự chứa, mỗi dịch vụ có cơ sở dữ liệu và logic riêng. Điều này cho phép các nhóm làm việc độc lập với nhau trên các dịch vụ khác nhau mà không gây xung đột.

**2. Liên kết lỏng lẻo:** Các dịch vụ trong microservices có liên kết lỏng lẻo, có nghĩa là chúng có thể được phát triển, triển khai và mở rộng độc lập. Sự thay đổi trong một dịch vụ không ảnh hưởng đến toàn bộ hệ thống, giúp việc bảo trì và cập nhật dễ dàng hơn.

**3. Khả năng mở rộng và linh hoạt:** Microservices cho phép mở rộng theo chiều ngang bằng cách cho phép các dịch vụ cá nhân được mở rộng độc lập dựa trên nhu cầu. Nó cũng cung cấp tính linh hoạt trong việc lựa chọn các công nghệ và ngôn ngữ lập trình khác nhau cho các dịch vụ khác nhau, tùy thuộc vào yêu cầu cụ thể của chúng.

**4. Khả năng chịu lỗi và cô lập lỗi:** Vì mỗi microservice chạy độc lập, một lỗi trong một dịch vụ không làm hỏng toàn bộ ứng dụng. Hệ thống có thể được thiết kế để chịu lỗi, với cơ chế cô lập lỗi và các phương pháp dự phòng.

**5. Giao tiếp dựa trên API:** Các microservice giao tiếp với nhau thông qua các API được xác định rõ ràng. Điều này cho phép các dịch vụ phát triển độc lập, miễn là các hợp đồng API được duy trì, và cho phép các dịch vụ khác nhau được phát triển bằng các công nghệ khác nhau.

**6. DevOps và Continuous Delivery:** Microservices phù hợp với các phương pháp DevOps và quy trình Continuous Delivery, vì mỗi dịch vụ có thể được phát triển, kiểm thử và triển khai độc lập, tạo điều kiện cho chu kỳ phát hành nhanh hơn.

**7. Giám sát và khả năng quan sát:** Microservices yêu cầu các cơ chế giám sát và khả năng quan sát toàn diện để theo dõi sức khỏe, hiệu suất và tương tác giữa các dịch vụ. Điều này rất quan trọng để xác định và giải quyết các vấn đề trong một hệ thống phân tán.

Mặc dù microservices mang lại lợi ích như khả năng mở rộng, linh hoạt và phát triển độc lập, nhưng cũng đưa ra những khó khăn như quản lý giao tiếp giữa các dịch vụ, sự nhất quán dữ liệu và triển khai. Tổ chức cần cân nhắc kỹ yêu cầu của ứng dụng và sự đánh đổi kiến trúc trước khi áp dụng microservices.

Dự án SpringBoot-Angular7-Online-Shopping-Store sử dụng Spring Boot và Angular 7 để xây dựng một cửa hàng mua sắm trực tuyến. Cấu trúc tổ chức microservice của dự án này có thể bao gồm các thành phần như:

Service Discovery: Dự án có thể sử

dụng một service registry như Netflix Eureka hoặc Consul để quản lý và phát hiện các microservice. Điều này cho phép các microservice tìm thấy và giao tiếp với nhau một cách tự động.

Product Service: Đây là một microservice chịu trách nhiệm quản lý thông tin về sản phẩm. Nó có thể cung cấp các API để lấy thông tin sản phẩm, tạo mới sản phẩm, cập nhật hoặc xóa sản phẩm.

Cart Service: Microservice này quản lý giỏ hàng của người dùng. Nó có thể cung cấp các API để thêm hoặc xóa sản phẩm từ giỏ hàng, tính tổng giá trị giỏ hàng và lưu trữ thông tin giỏ hàng cho người dùng.

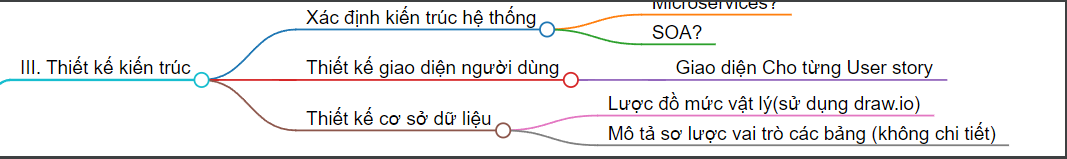
Order Service: Microservice này xử lý việc đặt hàng. Nó có thể xác nhận đơn hàng, lưu trữ thông tin đơn hàng và cung cấp các API để truy vấn thông tin đơn hàng.

User Service: Microservice này quản lý thông tin người dùng và xác thực. Nó có thể xử lý việc đăng ký người dùng mới, đăng nhập, quản lý thông tin người dùng và cung cấp các API liên quan đến người dùng.

Mỗi microservice trong dự án này được triển khai và quản lý độc lập, có thể có cơ sở dữ liệu riêng và có thể được mở rộng một cách độc lập. Chúng giao tiếp với nhau thông qua các giao thức như REST hoặc messaging, cho phép các microservice tương tác và hoạt động cùng nhau để cung cấp các chức năng toàn diện cho ứng dụng mua sắm trực tuyến.

Kiến trúc microservice cho phép dự án này phân chia ứng dụng thành các phần nhỏ hơn, giúp cải thiện khả năng mở rộng, quản lý và phát triển. Nó cũng cho phép các phần của hệ thống được triển khai và phát triển độc lập nhau, đồng thời giảm thiểu tác động khi một phần của hệ thống gặp sự cố hoặc cần được thay đổi.

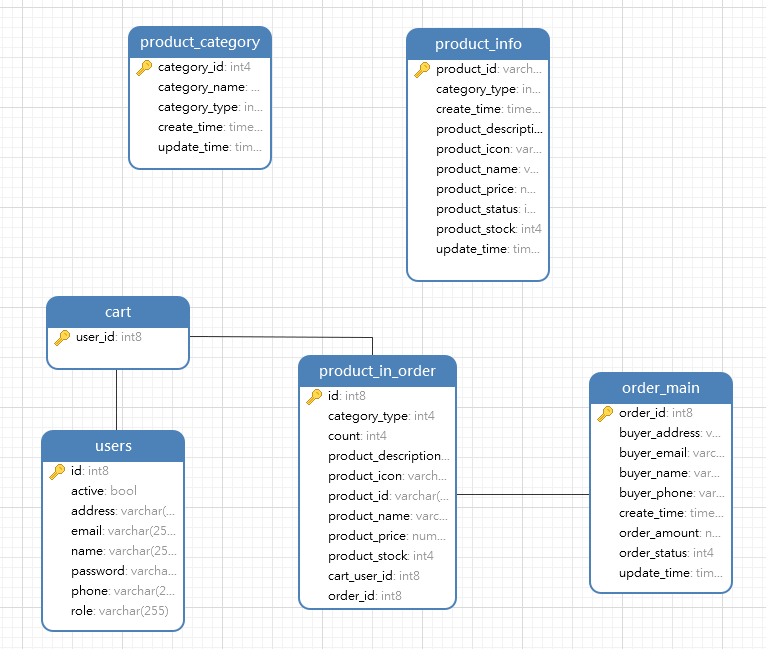
## 3.2 Thiết kế giao diện người dùng



Hình 11. Thiết kế giao diện người dùng

## 3.3 Thiết kế cơ sở dữ liệu

\*Lược đồ mức vật lý



Hình 12. Thiết kế mô hình dữ liệu mức vật lý

\* Mô tả sơ lược vai trò các bảng

1. **Bảng "product\_category" (Danh mục sản phẩm):**

- Đây là bảng chứa thông tin về các danh mục sản phẩm.

- Mỗi hàng trong bảng đại diện cho một danh mục sản phẩm cụ thể với các thuộc tính như ID danh mục và tên danh mục.

1. **Bảng "product\_info" (Thông tin sản phẩm):**

- Đây là bảng chứa thông tin về các sản phẩm.

- Mỗi hàng trong bảng đại diện cho một sản phẩm cụ thể với các thuộc tính như ID sản phẩm, tên sản phẩm, giá, mô tả và ID danh mục liên quan.

1. **Bảng "cart" (Giỏ hàng):**

- Đây là bảng lưu trữ thông tin về giỏ hàng của người dùng.

- Mỗi hàng trong bảng đại diện cho một mục trong giỏ hàng với các thuộc tính như ID mục giỏ hàng, ID sản phẩm, số lượng và ID người dùng.

1. **Bảng "user" (Người dùng):**

- Đây là bảng lưu trữ thông tin về người dùng của hệ thống.

- Mỗi hàng trong bảng đại diện cho một người dùng cụ thể với các thuộc tính như ID người dùng, tên người dùng và địa chỉ email.

1. **Bảng "product\_in\_order" (Sản phẩm trong đơn hàng):**

- Đây là bảng liên kết các sản phẩm cụ thể với các đơn hàng.

- Mỗi hàng trong bảng đại diện cho một mục sản phẩm trong đơn hàng với các thuộc tính như ID mục sản phẩm đơn hàng, ID sản phẩm, số lượng và ID đơn hàng.

1. **Bảng "order\_main" (Đơn hàng chính):**

- Đây là bảng lưu trữ thông tin về các đơn hàng chính.

- Mỗi hàng trong bảng đại diện cho một đơn hàng cụ thể với các thuộc tính như ID đơn hàng, ID người dùng, ngày đặt hàng và trạng thái đơn hàng.

# CHƯƠNG 4: QUY TRÌNH PHÁT TRIỂN

## 4.1 Agile

### 4.1.1 Agile là gì?

Agile được ra đời vào năm 2001 là một thuật ngữ mô tả các phương pháp tiếp cận phát triển phần mềm nhấn mạnh vào phân phối gia tăng, hợp tác nhóm. Phương pháp này được duy trì thông qua việc lên kế hoạch và học hỏi liên tục. trong Tuyên ngôn Agile (Agile Manifesto). Tuyên ngôn đề ra việc thiết lập các nguyên tắc để hướng dẫn một cách tiếp cận tốt hơn để phát triển phần mềm.

Trong các quy trình Agile, việc nhận phản hồi liên tục cho phép các thành viên trong nhóm điều chỉnh với các thách thức khi chúng phát sinh và các bên liên quan có cơ hội giao tiếp một cách nhất quán. Mặc dù ban đầu được tạo ra để phát triển phần mềm, phương pháp Agile hiện được sử dụng rộng rãi trong việc thực hiện nhiều loại dự án khác nhau. Nó còn được sử dụng như một phương pháp điều hành và quản trị ở nhiều tổ chức.

### 4.1.2 Các giá trị và nguyên tắc của Agile

Phương pháp quản lý dự án Agile được thiết lập dựa trên bốn giá trị và mười hai nguyên tắc. Những giá trị và nguyên tắc này bắt nguồn từ Tuyên ngôn Agile, được tạo ra vào năm 2001 bởi mười bảy nhà quản lý phát triển phần mềm uy tín trên toàn thế giới. Phần lớn các triết lý nền tảng cho Tuyên ngôn Agile ra đời nhằm phản ứng lại những gì mọi người coi là điểm yếu của các quy trình phát triển phần mềm vào thời điểm đó.

**\* Các giá trị của phương pháp Agile bao gồm:**

**1. Các cá nhân và sự tương tác đối với các quy trình và công cụ**: Trong khi hầu hết mọi người đều hiểu tầm quan trọng của các công cụ và quy trình, thì Tuyên ngôn Agile lại ưu tiên những người đứng sau chúng. Sử dụng đúng người và cho phép họ tương tác nhịp nhàng với nhau có thể dẫn đến những thành công mà bản thân các công cụ không thể làm được.

**2. Làm việc với phần mềm dựa trên tài liệu toàn diện**: Những người tạo ra Agile tin rằng hoàn thành công việc phải là mục tiêu cốt lõi của mọi dự án. Chúng ta phải tập trung vào nó hơn là sa lầy vào các giai đoạn lập kế hoạch và tài liệu.

**3. Hợp tác với khách hàng trong quá trình đàm phán hợp đồng**: Thay vì ngăn cản các bên liên quan rời khỏi dự án, Agile hướng tới việc duy trì liên hệ với họ trong suốt quá trình hợp tác.

**4. Đáp ứng sự thay đổi so với việc tuân theo một kế hoạch**: Việc tuân theo một kế hoạch không còn hợp lý có thể phản tác dụng. Thích ứng và đổi mới là trọng tâm của triết lý Agile.

**\* Tuyên ngôn Agile liệt kê 12 nguyên tắc mà các nhà phát triển phần mềm phải tuân theo:**

**1.** Ưu tiên cao nhất của là làm hài lòng khách hàng thông qua việc phân phối sớm và liên tục các phần mềm có giá trị.

**2.** Hoan nghênh các yêu cầu thay đổi, ngay cả khi phát triển muộn. Các quy trình nhanh nhẹn khai thác sự thay đổi vì lợi thế cạnh tranh của khách hàng.

**3.** Cung cấp phần mềm hoạt động thường xuyên, từ vài tuần đến vài tháng, với ưu tiên khoảng thời gian ngắn hơn.

**4.** Người kinh doanh và nhà phát triển phải làm việc cùng nhau trong suốt dự án.

**5.** Xây dựng các dự án xung quanh những cá nhân có động lực. Cung cấp cho họ môi trường và sự hỗ trợ họ cần, và tin tưởng để họ hoàn thành công việc.

**6.** Phương pháp hiệu quả nhất để truyền tải thông tin đến và trong nhóm phát triển là trò chuyện trực tiếp.

**7.** Phần mềm làm việc là thước đo chính của sự tiến bộ.

**8.** Các quy trình nhanh nhẹn thúc đẩy sự phát triển bền vững. Các nhà tài trợ, nhà phát triển và người dùng sẽ có thể duy trì tốc độ liên tục vô thời hạn.

**9.** Liên tục chú ý đến sự cải tiến về kỹ thuật và thiết kế tốt giúp tăng cường sự nhanh nhẹn.

**10.** Sự đơn giản – nghệ thuật tối đa hóa khối lượng công việc chưa hoàn thành – là điều cần thiết.

**11.** Các kiến ​​trúc, yêu cầu và thiết kế tốt nhất xuất hiện từ các nhóm tự tổ chức.

**12.** Theo định kỳ, nhóm phản ánh về cách trở nên hiệu quả hơn, sau đó điều chỉnh và điều chỉnh hành vi của mình cho phù hợp.

### 4.1.3 Phương pháp Agile

Phương pháp Agile (Agile methodology) là một phương pháp quản lý dự án linh hoạt và phát triển phần mềm dựa trên việc làm việc theo các vòng lặp ngắn gọi là "sprints". Agile tập trung vào việc thích ứng với sự thay đổi và tạo ra giá trị sớm cho khách hàng bằng cách tăng cường sự linh hoạt, tương tác và phản hồi trong quá trình phát triển.

Dưới đây là một số phương pháp Agile phổ biến:

**1. Scrum:** Scrum là một phương pháp Agile phổ biến, trong đó dự án được chia thành các "sprints" có độ dài cố định, thường là 2-4 tuần. Mỗi sprint tập trung vào việc phát triển một phần của sản phẩm và kết thúc bằng việc sản xuất một phiên bản có thể triển khai. Scrum sử dụng các cuộc họp ngắn như Daily Scrum và Sprint Review để tăng cường tương tác và quản lý tiến độ.

**2. Kanban:** Kanban là một phương pháp Agile tập trung vào quản lý và theo dõi quy trình làm việc. Bảng Kanban được sử dụng để hình dung quy trình làm việc và theo dõi tiến độ từng công việc. Các công việc được đưa vào bảng Kanban và di chuyển qua các cột khác nhau, đại diện cho các giai đoạn khác nhau trong quy trình.

**3. Extreme Programming (XP):** XP là một phương pháp Agile tập trung vào việc cải thiện chất lượng và tăng cường phản hồi trong quá trình phát triển. Nó sử dụng các phương pháp như lập trình theo cặp (pair programming), kiểm thử tự động (automated testing), quy trình phát triển ngắn và liên tục, và phản hồi thường xuyên từ khách hàng.

**4. Lean Software Development:** Lean Software Development áp dụng các nguyên tắc Lean từ ngành công nghiệp sản xuất vào phát triển phần mềm. Nó tập trung vào việc loại bỏ lãng phí, tối ưu hóa quy trình và cung cấp giá trị nhanh chóng cho khách hàng. Các nguyên tắc như "tạo đều (flow)", "rút gọn (eliminate waste)", và "tìm hiểu liên tục (continuous learning)" được áp dụng trong quá trình phát triển.

Các phương pháp Agile đều nhấn mạnh sự linh hoạt, tương tác, và tập trung vào việc đáp ứng nhanh chóng cho yêu cầu của khách hàng. Chúng giúp tăng cường khả năng thích ứng và cải tiến liên tục trong quá trình phát triển phần mềm.

## 4.2 Scrum

### 4.2.1 Scrum là gì?

Scrum là một phương pháp quản lý dự án và phát triển phần mềm trong Agile. Nó tập trung vào việc tổ chức công việc thành các đợt phát triển ngắn gọi là "sprints" để tăng cường tương tác và sản xuất giá trị nhanh chóng. Scrum được xây dựng trên ba phần chính: vai trò, sự kiện và các sản phẩm.

### 4.2.2 Khái niệm cơ bản của Scrum

**1. Vai trò:**

- Scrum Master: Người đảm nhận vai trò lãnh đạo và hướng dẫn nhóm Scrum. Scrum Master giúp đảm bảo quá trình Scrum diễn ra một cách hiệu quả và loại bỏ các rào cản gây trở ngại cho nhóm.

- Product Owner: Người đại diện cho khách hàng hoặc người sử dụng cuối trong dự án. Product Owner xác định yêu cầu, ưu tiên và quản lý Product Backlog - danh sách các yêu cầu cho sản phẩm.

- Nhóm Scrum: Nhóm phát triển chịu trách nhiệm thực hiện công việc và sản xuất sản phẩm. Nhóm Scrum thường gồm các thành viên đa chuyên ngành và tự tổ chức.

**2. Sự kiện:**

- Sprint: Một khoảng thời gian cố định, thường từ 1 đến 4 tuần, trong đó nhóm Scrum phát triển một phần của sản phẩm và tạo ra một phiên bản có thể triển khai.

- Sprint Planning: Cuộc họp để xác định và lập kế hoạch cho công việc trong một Sprint. Product Owner và nhóm Scrum cùng nhau định rõ mục tiêu và lựa chọn các công việc từ Product Backlog.

- Daily Scrum: Một cuộc họp ngắn hàng ngày, không quá 15 phút, trong đó các thành viên nhóm Scrum chia sẻ tiến độ công việc, thảo luận về những rào cản và đồng bộ hóa hoạt động của mình.

- Sprint Review: Cuộc họp cuối Sprint để xem xét và đánh giá kết quả của công việc. Product Owner và khách hàng được mời tham gia để xem phiên bản sản phẩm mới và cung cấp phản hồi.

- Sprint Retrospective: Cuộc họp sau Sprint Review để nhóm Scrum đánh giá quá trình làm việc của mình và tìm kiếm cách cải thiện. Nhóm xác định những điểm mạnh và điểm yếu, đề xuất biện pháp cải tiến và lên kế hoạch cho Sprint tiếp theo.

**3. Quy trình phát triển:**

Xác định Sản phẩm (Product Backlog): Xác định và liệt kê tất cả các yêu cầu và chức năng của sản phẩm trong danh sách "Product Backlog". Các yêu cầu được sắp xếp theo mức độ ưu tiên.

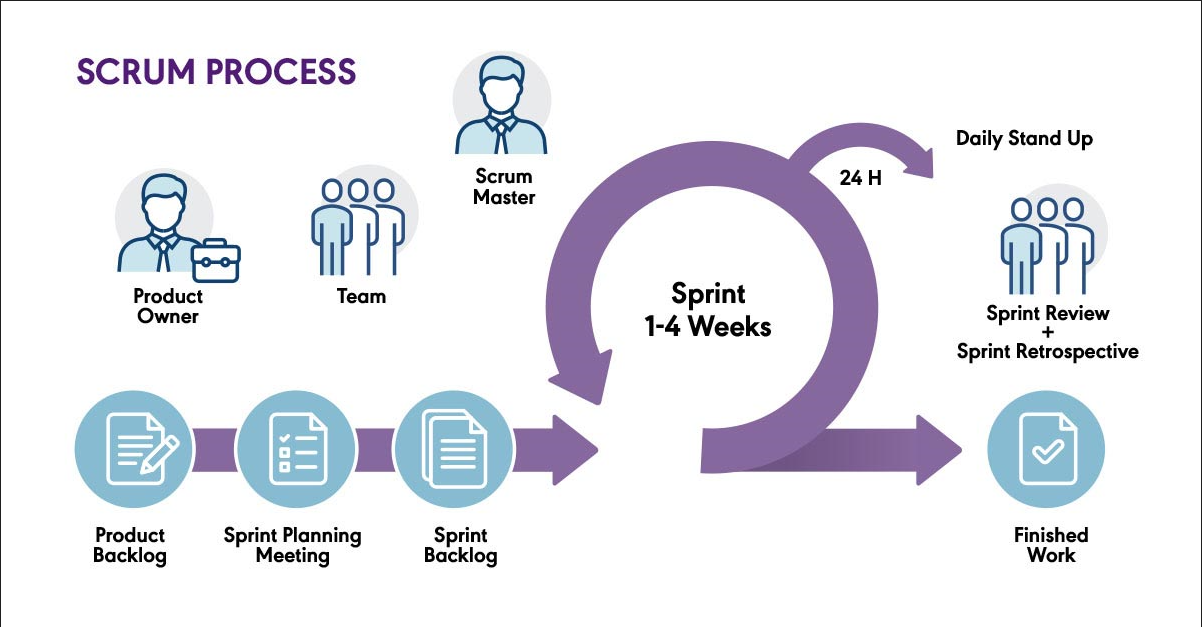
Lập kế hoạch Sprint (Sprint Planning): Chọn một tập hợp các yêu cầu từ Product Backlog để thực hiện trong một đợt phát triển gọi là "Sprint". Xác định mục tiêu của Sprint và tạo ra một kế hoạch chi tiết về cách thực hiện các yêu cầu đó.

Thực hiện Sprint (Sprint Execution): Các yêu cầu được phát triển trong suốt thời gian của Sprint, thông qua các hoạt động như lập trình, kiểm thử và tích hợp.

Daily Scrum (Daily Stand-up): Mỗi ngày, các thành viên trong nhóm Scrum tổ chức một cuộc họp ngắn gọi là Daily Scrum để trao đổi về tiến độ công việc, thảo luận về các vấn đề gặp phải và định kế hoạch cho ngày làm việc tiếp theo.

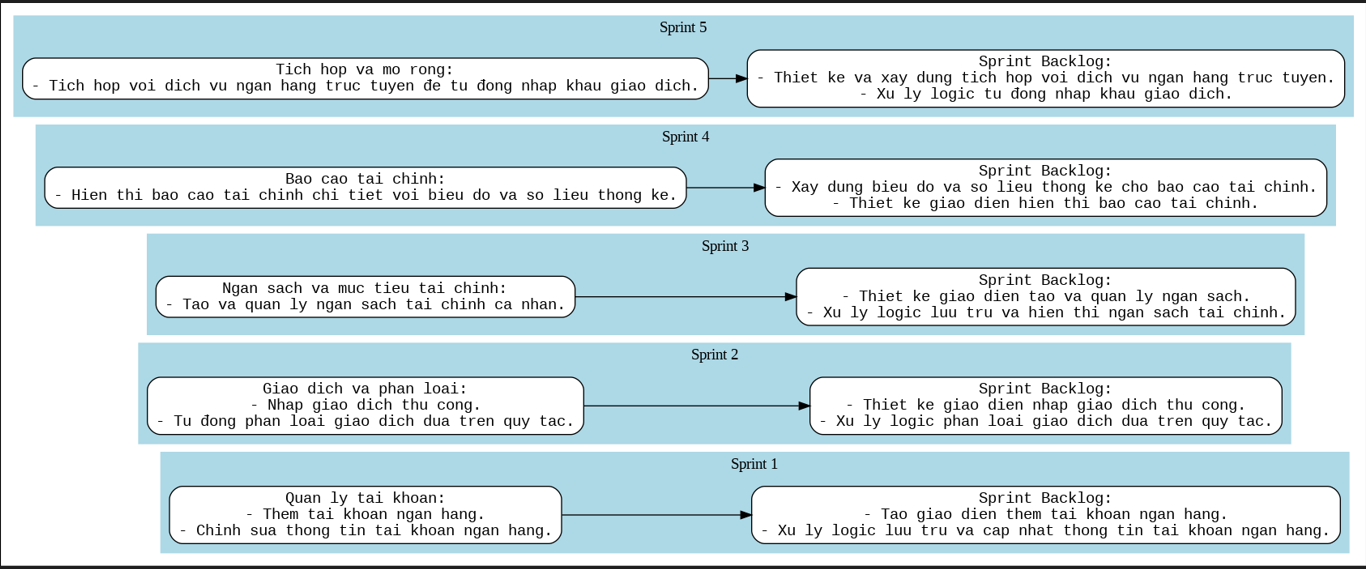
Kiểm tra Sprint (Sprint Review): Sau khi hoàn thành một Sprint, nhóm Scrum tổ chức một buổi họp gọi là Sprint Review để đánh giá sản phẩm đạt được và thu thập phản hồi từ khách hàng hoặc người sử dụng. Dựa trên phản hồi này, Product Backlog có thể được điều chỉnh.

Retrospective (Sprint Retrospective): Sau cuộc họp Sprint Review, nhóm Scrum tiến hành cuộc họp Retrospective để đánh giá và cải thiện quy trình làm việc của họ. Các thành viên nhóm đánh giá các khía cạnh tích cực và tiêu cực của dự án và đề xuất các biện pháp cải thiện.



Hình 13 Quy trình phát triển của Scrum

## 4.2 Sơ đồ quy trình phát triển



Hình 14. Sơ đồ quy trình phát triển

# CHƯƠNG 5: TRIỂN KHAI

## 5.1 Docker

Docker là một nền tảng mã nguồn mở cho việc tạo, triển khai và quản lý các ứng dụng trong một môi trường được gọi là container. Docker cho phép đóng gói ứng dụng và tất cả các phụ thuộc của nó vào một container độc lập, mang lại sự di động, tin cậy và hiệu suất cao cho quá trình triển khai ứng dụng.

**\* Một số khái niệm và thuật ngữ cơ bản trong Docker:**

**1. Container:** Container là một đơn vị chứa chứa tất cả các thành phần cần thiết để chạy một ứng dụng, bao gồm mã nguồn, thư viện, biến môi trường và tài nguyên. Containers cung cấp sự cô lập và độc lập giữa các ứng dụng, giúp đảm bảo tính nhất quán và di động.

**2. Docker Image:** Docker Image là một gói chứa các thành phần cần thiết để tạo một container. Nó bao gồm mã nguồn, thư viện, cấu hình và tài nguyên cần thiết để chạy ứng dụng. Docker Image có thể được xây dựng từ Dockerfile hoặc có thể được tải xuống từ Docker Hub, một kho lưu trữ chia sẻ các Docker Image đã được xây dựng sẵn.

**3. Dockerfile:** Dockerfile là một tập tin văn bản chứa các chỉ thị và lệnh để xây dựng một Docker Image. Nó cho phép định nghĩa môi trường, cài đặt ứng dụng và cấu hình khác để tạo một Image.

**4. Docker Registry:** Docker Registry là một kho lưu trữ chứa các Docker Image. Docker Hub là một Registry công cộng, trong khi Docker Enterprise Edition cung cấp các lựa chọn để triển khai các Registry riêng tư.

**5. Docker Compose:** Docker Compose là một công cụ cho phép định nghĩa và quản lý nhiều container như một ứng dụng duy nhất. Nó cho phép xác định các dịch vụ, mạng và cấu hình liên quan giữa các container.

Docker giúp đơn giản hóa việc triển khai ứng dụng và tạo ra môi trường di động và nhất quán trên nhiều nền tảng. Nó cho phép ứng dụng chạy độc lập trong các container cô lập và cung cấp khả năng mở rộng và quản lý dễ dàng. Docker cũng giúp tăng cường sự đồng nhất và khả năng phát triển phần mềm trong môi trường DevOps.

## 5.2 Các bước triển khai

**A. Chuẩn bị môi trường**

\* Đảm bảo rằng máy tính đã cài đặt các ứng dụng:

- PostgreSQL với username là postgres, password là root

- java jdk 11

- maven tương ứng với bản java jdk 11

- nodejs nvm 8.9.4

### 5.2.1 Triển khai ứng dụng lên Localhost

Clone dự án từ github về:  
$ https://github.com/zhulinn/SpringBoot-Angular7-Online-Shopping-Store.git

Chuyển đến thư mục chứa dự án, chuyển đến thư mục backend  
$ cd SpringBoot-Angular7-Online-Shopping-Store/backend

Cài đặt một dự án Maven vào kho lưu trữ Maven cục bộ hoặc từ xa

$ mvn install

A screenshot of a computer program

Description automatically generated

Hình 15. Cài đặt một dự án Maven vào kho lưu trữ Maven cục bộ hoặc từ xa

Chạy ứng dụng Spring Boot từ Maven

$ mvn spring-boot:run

-> Spring Boot sẽ tự động nhập dữ liệu mẫu vào cơ sở dữ liệu bằng cách thực thi tệp import.sql

Chuyển đến thư mục frontend

$ cd ..

$ cd frontend

Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm

$ npm install

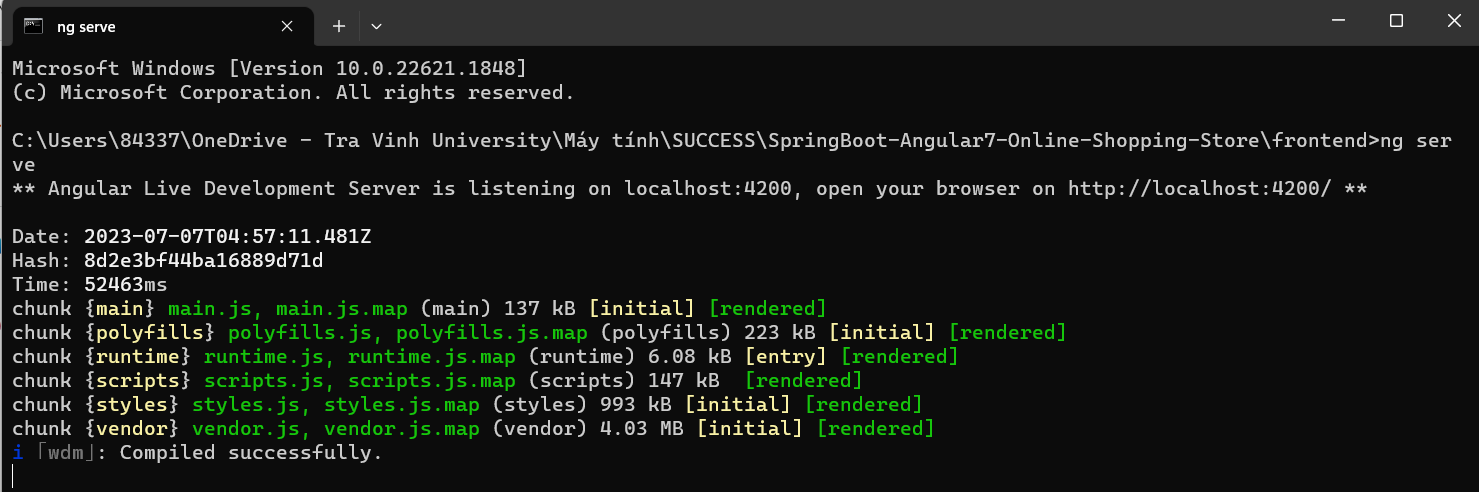
A screenshot of a computer

Description automatically generated

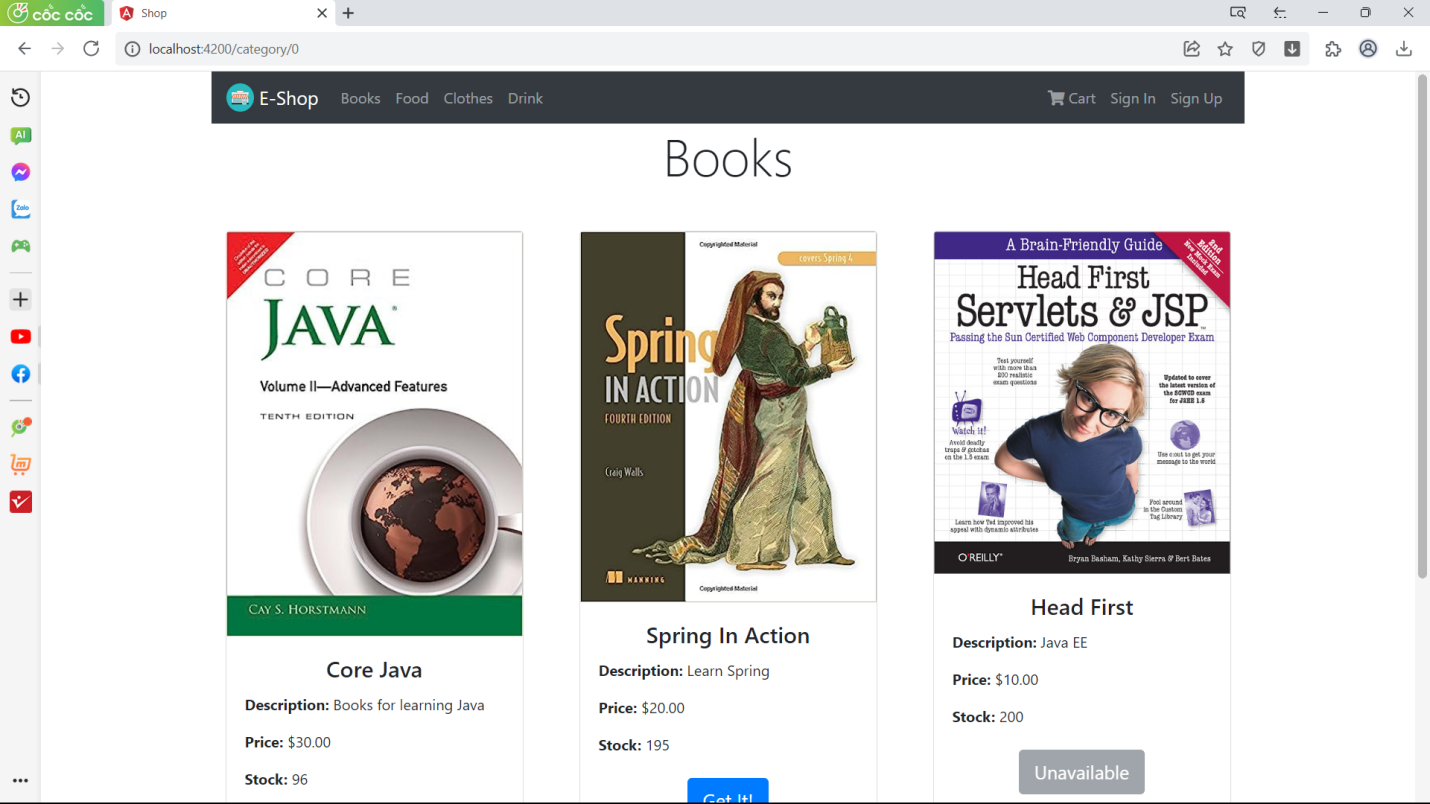
Hình 16. Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm

Chạy ứng dụng Angular trong môi trường phát triển

$ ng serve



Hình 17. Chạy ứng dụng Angular trong môi trường phát triển



Hình 18. Ứng dụng sau khi triển khai trên Localhost

### 5.2.2 Triển khai ứng dụng trên docker

**A. Xây dựng dự án backend**

Chuyển đến thư mục backend

$ cd backend

Đóng gói một dự án thành một file artifact

$ mvn package

A screenshot of a computer screen

Description automatically generated

Hình 19. Đóng gói một dự án thành một file artifact

**B. Xây dựng dự án frontend**

Chuyển đến thư mục frontend

$ cd frontend

Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm

$ npm install

A screenshot of a computer program

Description automatically generated

Hình 20. Cài đặt các gói (packages) JavaScript từ kho lưu trữ npm

Xây dựng (build) một phiên bản sản phẩm (production build) của ứng dụng Angular, tạo ra các tệp mã nguồn và tài nguyên tối ưu hóa và sẵn sàng để triển khai trên môi trường sản phẩm.

$ ng build --prod

A screenshot of a computer program

Description automatically generated

Hình 21. Xây dựng (build) một phiên bản sản phẩm (production build)

Chuyển đến thư mục chứa docker-compose.yml để build image và run container

$ cd ..

$ docker-compose up --build

A screenshot of a computer

Description automatically generated

Hình 22. Các images có trong máy

A screenshot of a computer

Description automatically generated

Hình 23. Các Container có trong máy

1. **Push (đẩy) các image lên docker hub**

Đăng nhập vào dockerhub

$ docker login

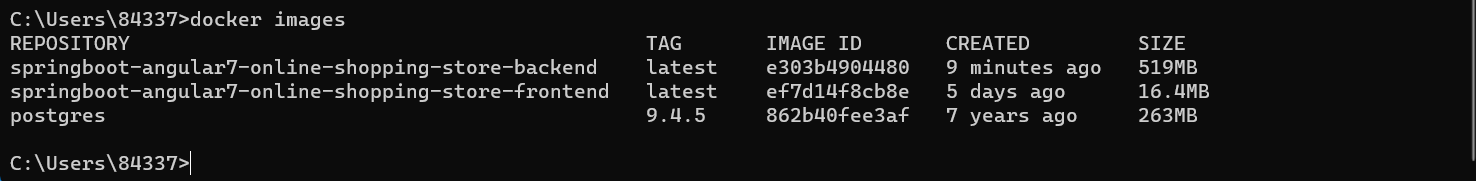
A screen shot of a computer

Description automatically generated

Hình 24. Đăng nhập vào dockerhub

Xem các image có trên máy cục bộ

$ docker images



Hình 25. Xem các image có trên máy cục bộ

Đặt lại tag cho các image

$ docker tag postgres:9.4.5 sostien0409/postgres:9.4.5

$ docker tag springboot-angular7-online-shopping-store-backend sostien0409/springboot-angular7-online-shopping-store-backend

$ docker tag springboot-angular7-online-shopping-store-frontend sostien0409/springboot-angular7-online-shopping-store-frontend

A screen shot of a computer

Description automatically generated

Hình 26. Đặt lại tag cho các image

Xem lại các images

$ docker images

A screen shot of a computer

Description automatically generated

Hình 27. Xem lại các images

Đẩy các image lên dockerhub

$ docker push sostien0409/postgres:9.4.5

A screen shot of a computer

Description automatically generated

Hình 28. Đẩy image postgres lên dockerhub

$ docker push sostien0409/springboot-angular7-online-shopping-store-frontend

A black screen with white text

Description automatically generated

Hình 29. Đẩy image frontend lên dockerhub

$ docker push sostien0409/springboot-angular7-online-shopping-store-backend

A screen shot of a computer

Description automatically generated

Hình 30. Đẩy các image backend lên dockerhub

A screenshot of a computer

Description automatically generated

Hình 31. Sau khi triển khai trên DockerHub

### 5.2.3 Triển khai lên cloud

Đăng ký tài khoản tại cloudfly.vn

Tạo máy chủ tên là TestCloud chạy hệ điều hành ubuntu 18.04

A screenshot of a computer

Description automatically generated

Hình 32. Đăng ký tài khoản tại cloudfly.vn

Dùng putty để kết nối với máy chủ sau đó đăng nhập vào máy chủ

A screenshot of a computer

Description automatically generated

Hình 33. Dùng putty để kết nối với máy chủ sau đó đăng nhập vào máy chủ

Cập nhật các gói phần mềm

$ sudo apt update

$ sudo apt upgrade

Cài đặt các gói phụ thuộc cần thiết để cài đặt Docker:

$ sudo apt install apt-transport-https ca-certificates curl software-properties-common

Thêm kho lưu trữ Docker GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Thêm kho lưu trữ Docker vào hệ thống:

$ echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Cập nhật lại danh sách gói phần mềm và cài đặt Docker:

$ sudo apt update

$ sudo apt install docker-ce docker-ce-cli containerd.io

Kiểm tra xem Docker đã được cài đặt thành công hay chưa:

$ sudo docker run hello-world

A screenshot of a computer

Description automatically generated

Hình 34. Cập nhật và thêm kho ảnh

Nếu nhìn thấy thông báo "Hello from Docker!" có nghĩa là Docker đã được cài đặt thành công trên máy chủ

Kéo các images đã đẩy lên dockerhub về

$ sudo docker pull sostien0409/postgres:9.4.5

A screenshot of a computer

Description automatically generated

Hình 35. Kéo các images đã đẩy lên dockerhub về

$ sudo docker pull sostien0409/springboot-angular7-online-shopping-store-backend

A screenshot of a computer

Description automatically generated

Hình 36. Kéo các images đã đẩy lên dockerhub về

$ sudo docker pull sostien0409/springboot-angular7-online-shopping-store-frontend

A screenshot of a computer

Description automatically generated

Hình 37. Kéo các images đã đẩy lên dockerhub về

Chạy các container của các images

$ docker run -d --name my-postgres -p 5432:5432 -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=root sostien0409/postgres:9.4.5

A screenshot of a computer

Description automatically generated

Hình 38. Chạy các container của các images

$ docker run -d --name my-backend -p 8080:8080 --link my-postgres:db sostien0409/springboot-angular7-online-shopping-store-backend:latest

A screenshot of a computer

Description automatically generated

Hình 39. Chạy các container của các images

$ docker run -d --name my-frontend -p 8008:80 --link my-backend:backend sostien0409/springboot-angular7-online-shopping-store-frontend:latest

A screenshot of a computer

Description automatically generated

Hình 40. Chạy các container của các images

Xem các container đang chạy

$ docker ps -a

A screenshot of a computer screen

Description automatically generated

Hình 41. Xem các container đang chạy

Chúng ta có thể thấy, frontend đang chạy ở cổng 8008

Truy cập vào địa chỉ <IP cloud server>:<port>, cụ thể ở đây là 103.82.24.128:8080

A screenshot of a login page

Description automatically generated

Hình 42. Frontend đang chạy ở cổng 8008

# CHƯƠNG 6: KẾT LUẬN

## 6.1 Kết quả đạt được

Tính linh hoạt và đáp ứng nhanh: Agile giúp nhóm của bạn làm việc theo các vòng lặp ngắn, linh hoạt và có khả năng thích ứng nhanh với thay đổi. Điều này giúp bạn chủ động điều chỉnh và cải thiện dự án theo phản hồi từ khách hàng và yêu cầu thay đổi.

Tăng tốc độ triển khai: Bằng cách triển khai dự án thành Docker image, bạn có thể tạo ra một môi trường đồng nhất và xây dựng hệ thống dễ dàng hơn. Docker cho phép đóng gói ứng dụng và các phụ thuộc của nó thành các container di động và cung cấp môi trường phát triển và triển khai đồng nhất trên mọi máy chủ.

Tính di động và mở rộng: Docker cho phép triển khai ứng dụng và các microservice của bạn trên môi trường cloud linh hoạt như AWS, Google Cloud hay Microsoft Azure. Bằng cách triển khai lên cloud, bạn có thể tận dụng các tài nguyên mạnh mẽ, linh hoạt và tự động hóa để mở rộng và quản lý ứng dụng một cách hiệu quả.

Tăng tính nhất quán và khả năng tái sử dụng: Docker image cung cấp một môi trường cố định và đáng tin cậy cho ứng dụng. Bạn có thể tái sử dụng Docker image để triển khai và chạy ứng dụng trên nhiều môi trường, đảm bảo tính nhất quán và giảm rủi ro do khác biệt giữa các môi trường.

Tính bảo mật và kiểm soát: Docker cung cấp các tính năng bảo mật để cô lập và kiểm soát các container. Bạn có thể thiết lập các quyền truy cập và giới hạn tài nguyên cho từng container, tăng cường bảo mật và đảm bảo an toàn cho ứng dụng của bạn.

Hướng phát triển

Mở rộng tính năng của ứng dụng: Xem xét mở rộng chức năng của ứng dụng mua sắm trực tuyến để cung cấp trải nghiệm người dùng tốt hơn. Bạn có thể thêm tính năng như thanh toán trực tuyến, bảo mật giao dịch, quản lý đánh giá và nhận xét sản phẩm, tích hợp với các dịch vụ vận chuyển và theo dõi đơn hàng.

Tối ưu hóa hiệu suất và khả năng mở rộng: Tiếp tục nghiên cứu và tối ưu hóa hiệu suất của các microservice và hệ thống dự án. Sử dụng công cụ và kỹ thuật giám sát và theo dõi để phát hiện và giải quyết các vấn đề về hiệu suất. Nếu ứng dụng của bạn phát triển, đảm bảo rằng hệ thống có khả năng mở rộng bằng cách thêm các phiên bản mới của microservice hoặc sử dụng các công nghệ như Kubernetes để quản lý việc mở rộng tự động.

Tích hợp và liên kết với các dịch vụ bên ngoài: Nếu ứng dụng của bạn cần tích hợp với các dịch vụ bên ngoài như thanh toán trực tuyến, dịch vụ xác thực, dịch vụ gửi email, v.v., bạn có thể tiếp tục phát triển tích hợp với các API của các dịch vụ này để mở rộng khả năng của ứng dụng.

Tăng cường bảo mật và quản lý xác thực: Đảm bảo rằng hệ thống của bạn có các biện pháp bảo mật vững chắc và kiểm soát truy cập an toàn. Xem xét sử dụng các công nghệ như JWT (JSON Web Token) để quản lý xác thực và ủy quyền người dùng.

Thích ứng với thiết bị di động: Nếu chưa có, bạn có thể phát triển ứng dụng di động (ví dụ: ứng dụng Android hoặc iOS) để cho phép người dùng truy cập và mua hàng từ thiết bị di động của họ. Điều này mở rộng phạm vi khách hàng và cung cấp trải nghiệm tiện lợi hơn.

Tự động hóa quy trình triển khai: Nâng cao quy trình triển khai bằng cách xây dựng quy trình triển khai tự động sử dụng các công cụ như Jenkins hoặc GitLab CI/CD. Điều này giúp giảm thời gian triển khai và đảm bảo tính nhất quán trong việc triển khai các phiên bản mới của ứng dụng.

## 6.2 Khó khăn và bài học

Quá trình tìm kiếm đề tài mất nhiều thời gian

Cài đặt nhiều môi trường phát triển như Nodejs, Java, Angular, ...

## 6.3 Hướng phát triển

Tạo các Docker Images với các trang web phức tạp hơn

# CHƯƠNG 7: TÀI LIỆU THAM KHẢO

[1] https://docs.npmjs.com/getting-started

[2] https://docs.oracle.com/javase/tutorial/getStarted/index.html

[3] https://docs.docker.com/get-started/resources/

[4] https://www.w3schools.com/aws/aws\_cloudessentials\_cloudcomputing.php