TRƯỜNG ĐẠI HỌC TRÀ VINH

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

****

**ISO 9001:2015**

**BÁO CÁO KẾT THÚC MÔN**

**CÔNG NGHỆ PHẦN MỀM**  
**ĐỀ TÀI:**

**QUẢN LÝ CỬA HÀNG THIẾT BỊ ĐIỆN TỬ**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn**  **NGUYỄN BẢO ÂN** | **Nhóm Sinh viên thực hiện**  **NGÔ TẤN LỢI - 110120166**  **TRẦN PHÚC VĨ - 110120084**  **KIM THỊ SÔ PHI - 110120060**  Mã Lớp DA20TTA  Khóa 2020-2024 |

**TRÀ VINH, THÁNG 7 NĂM 2023**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn Thầy Nguyễn Bảo Ân. Nhờ sự giúp đỡ tận tình và những chỉ bảo của Thầy từ lúc bắt đầu học cho đến kết thúc môn học mà chúng em đã hoàn thành đúng thời hạn quy định và tích lũy được cho mình một lượng nền tảng kiến thức quý báu.

Thầy Bảo Ân đã không ngừng truyền đạt kiến thức một cách rõ ràng và chi tiết, giúp chúng em hiểu sâu về các khái niệm và vấn đề trong môn học. Thầy luôn sẵn lòng giải đáp những thắc mắc của chúng em và tạo điều kiện để chúng em phát triển tư duy và khả năng giải quyết vấn đề.

Chúng em chắc chắn rằng những kiến thức và kỹ năng mà chúng em đã học được từ Thầy Nguyễn Bảo Ân sẽ là nền tảng vững chắc cho hành trình phát triển sau này. Chúng em sẽ luôn ghi nhớ những bài học và lời khuyên của Thầy để áp dụng vào cuộc sống và công việc của mình.

Chúng em xin chân thành cảm ơn.

*Trà Vinh, ngày tháng năm*

Nhóm thực hiện:

Ngô Tấn Lợi - 110120166

Trần Phúc Vĩ - 110120084

Kim Thị Sô Phi - 110120060

**MỤC LỤC**

[CHƯƠNG 1 GIỚI THIỆU 8](#_Toc140004106)

[1.1 Tổng quan về dự án 8](#_Toc140004107)

[1.2 Mục tiêu báo cáo 8](#_Toc140004108)

[CHƯƠNG 2 PHÂN TÍCH YÊU CẦU 10](#_Toc140004109)

[2.1 Yêu cầu của dự án 10](#_Toc140004110)

[2.2 Các chức năng và tính năng của dự án 10](#_Toc140004111)

[2.2.1 User stories 10](#_Toc140004112)

[2.2.2 Features 11](#_Toc140004113)

[CHƯƠNG 3 THIẾT KẾ KIẾN TRÚC 13](#_Toc140004114)

[3.1 Kiến trúc hệ thống 13](#_Toc140004115)

[3.1.1 Microservices 13](#_Toc140004116)

[3.1.2 SOA (Service-Oriented Architecture) 13](#_Toc140004117)

[3.2 Thiết kế giao diện người dùng 13](#_Toc140004118)

[3.3 Thiết kế cơ sở dữ liệu 15](#_Toc140004119)

[3.3.1 Lược đồ mức vật lý 15](#_Toc140004120)

[3.3.2 Vai trò các bảng 15](#_Toc140004121)

[CHƯƠNG 4 QUY TRÌNH PHÁT TRIỂN 17](#_Toc140004122)

[4.1 Phương pháp Agile 17](#_Toc140004123)

[4.1.1 Aglie là gì? 17](#_Toc140004124)

[4.1.2 Đặc điểm của Agile 17](#_Toc140004125)

[4.2 Phương pháp Scrum 20](#_Toc140004126)

[4.2.1 Scrum là gì? 20](#_Toc140004127)

[4.2.2 Đặc điểm của Scrum 20](#_Toc140004128)

[4.3 Sơ đồ quy trình phát triển 20](#_Toc140004129)

[4.3.1 Lập Product backlog 20](#_Toc140004130)

[4.3.1 Lập Sprint 21](#_Toc140004131)

[4.4 Công cụ hỗ trợ phát triển 22](#_Toc140004132)

[CHƯƠNG 5 TRIỂN KHAI 24](#_Toc140004133)

[5.1 Môi trường triển khai 24](#_Toc140004134)

[5.1.1 Docker 24](#_Toc140004135)

[5.1.2 Cloud 24](#_Toc140004136)

[5.2 Quy trình 24](#_Toc140004137)

[5.2.1 Quy trình đưa lên Docker 24](#_Toc140004138)

[5.2.2 Quy trình đưa lên CloudFly 25](#_Toc140004139)

[5.3 Các bước triển khai 25](#_Toc140004140)

[5.3.1 Các bước triển khai website lên Docker 25](#_Toc140004141)

[5.3.2 Các bước triển khai website lên CloudFly 29](#_Toc140004142)

[CHƯƠNG 6 KẾT LUẬN 37](#_Toc140004143)

[6.1 Kết quả đạt được 37](#_Toc140004144)

[6.2 Khó khăn và bài học 37](#_Toc140004145)

[6.3 Hướng phát triển 38](#_Toc140004146)

**DANH MỤC HÌNH ẢNH**

[Hình 3‑1 Lược đồ mức vật lý 15](#_Toc140007851)

[Hình 4‑1 Quy trình Agile 17](#_Toc140007852)

[Hình 4‑2 Product backlog 21](#_Toc140007853)

[Hình 4‑3 Sprint Backlog 21](#_Toc140007854)

[Hình 4‑4 Sprint Planning 21](#_Toc140007855)

[Hình 4‑5 Sprint Review 22](#_Toc140007856)

[Hình 4‑6 Sprint Retrospective 22](#_Toc140007857)

[Hình 4‑7 Product Increment 22](#_Toc140007858)

[Hình 5‑1 Docker-compose.yml 26](#_Toc140007859)

[Hình 5‑2 Dockerfile.mydb 26](#_Toc140007860)

[Hình 5‑3 Dockerfile.apache 27](#_Toc140007861)

[Hình 5‑4 Mở terminal trên Dockerfile.mydb 27](#_Toc140007862)

[Hình 5‑5 Chạy lệnh docker build -t mysql -f Dockerfile.mydb . 27](#_Toc140007863)

[Hình 5‑6 Chạy lệnh docker build -t selling-computer -f Dockerfile.apache . 28](#_Toc140007864)

[Hình 5‑7 Chạy hai lệnh yêu cầu 28](#_Toc140007865)

[Hình 5‑8 Giao diện docker desktop sau khi chạy xong 29](#_Toc140007866)

[Hình 5‑9 Website hoạt động sau khi truy cập vào địa chỉ IP trên docker desktop 29](#_Toc140007867)

[Hình 5‑10 Nơi lưu dự án 30](#_Toc140007868)

[Hình 5‑11 Giao diện DockerHub khi push thành công 30](#_Toc140007869)

[Hình 5‑12 Giao diện trang chủ CloudFly 31](#_Toc140007870)

[Hình 5‑13 Giao diện Instances 31](#_Toc140007871)

[Hình 5‑14 Các bước tạo Cloud 32](#_Toc140007872)

[Hình 5‑15 Giao diện khi tạo thành công 33](#_Toc140007873)

[Hình 5‑16 Giao diện Putty 33](#_Toc140007874)

[Hình 5‑17 Đăng nhập Putty 34](#_Toc140007875)

[Hình 5‑18 Cài đặt Docker thành công 35](#_Toc140007876)

[Hình 5‑19 Đăng nhập DockerHub thành công 35](#_Toc140007877)

[Hình 5‑20 Images đã được kéo về 35](#_Toc140007878)

[Hình 5‑21 Truy cập thành công 36](#_Toc140007879)

**NHẬN XÉT CỦA GIẢNG VIÊN BỘ MÔN**

**LỜI MỞ ĐẦU**

Trong thập kỷ gần đây, công nghệ thông tin đã bùng nổ và phát triển mạnh mẽ tại Việt Nam. Sự phát triển nhanh chóng của khoa học và công nghệ đã tạo ra nhiều sản phẩm công nghệ mới, mang lại nhiều tiện ích cho cuộc sống của con người. Công nghệ thông tin đóng vai trò quan trọng trong nhiều ngành nghề và cải thiện chất lượng cuộc sống.

Trong lĩnh vực quản lý, công nghệ thông tin đã giúp tăng hiệu quả và giảm thiểu sự phụ thuộc vào lao động và thời gian. Thay vì sử dụng phương pháp truyền thống với giấy tờ, công nghệ thông tin giúp quản lý dữ liệu một cách chính xác và tránh sai sót. Ngoài ra, nó còn giúp tìm kiếm và tra cứu thông tin nhanh chóng và chính xác.

Trong bối cảnh này, việc xây dựng chương trình quản lý cho ngành quản lý trở nên cần thiết để nâng cao hiệu quả kinh doanh và chất lượng phục vụ khách hàng. Đời sống ngày càng được cải thiện, và nhu cầu của con người cũng ngày càng tăng cao. Điều này đã thúc đẩy sự phát triển của lĩnh vực vận tải. Áp dụng công nghệ thông tin vào quản lý, đặc biệt là xây dựng một phần mềm quản lý bán hàng, trở nên cấp thiết để cạnh tranh trong kinh doanh và đáp ứng nhanh chóng nhu cầu của khách hàng khó tính nhất. Với mục tiêu tìm hiểu các vấn đề trên, nhóm chúng tôi đã chọn đề tài **“Quản lý cửa hàng thiết bị điện tử”.**

Qua quá trình nghiên cứu và thực hiện đề tài, chúng tôi đã tiếp cận và áp dụng các kiến thức về cơ sở dữ liệu để xây dựng hệ thống quản lý thư viện hiệu quả. Chúng tôi đã tìm hiểu và áp dụng các phương pháp thiết kế cơ sở dữ liệu, xây dựng các bảng dữ liệu, quan hệ giữa chúng, và các câu truy vấn để truy xuất thông tin một cách nhanh chóng và chính xác.

Đề tài này không chỉ giúp chúng tôi có kiến thức vững chắc về cơ sở dữ liệu, mà còn cho phép chúng tôi áp dụng những kiến thức đó vào thực tế, từ đó cải thiện quy trình quản lý thư viện và nâng cao hiệu quả công việc.

Thông qua quá trình nghiên cứu và thực hiện dự án, chúng tôi đã tiếp cận và áp dụng kiến thức về công nghệ phần mềm để xây dựng một hệ thống quản lý hiệu quả. Chúng tôi đã tìm hiểu và áp dụng các phương pháp thiết kế và phát triển phần mềm, xây dựng các module, quan hệ giữa chúng, và các chức năng để đáp ứng nhu cầu của người dùng một cách tốt nhất.

Dự án này không chỉ giúp chúng tôi có kiến thức vững chắc về công nghệ phần mềm, mà còn cho phép chúng tôi áp dụng những kiến thức đó vào thực tế, từ đó cải thiện quy trình quản lý và nâng cao hiệu quả công việc. Chúng tôi đã sử dụng các phương pháp phát triển phần mềm như quy trình Agile, kiểm thử và triển khai để đảm bảo sản phẩm đáp ứng yêu cầu chất lượng và thời gian.

# GIỚI THIỆU

## 1.1 Tổng quan về dự án

Dự án "Quản lý cửa hàng thiết bị điện tử" là một dự án website được triển khai trên môi trường Docker và đám mây (cloud). Mục tiêu chính của dự án là cung cấp một hệ thống quản lý hoàn chỉnh cho cửa hàng thiết bị điện tử, bao gồm quản lý sản phẩm, quản lý đơn hàng và quản lý khách hàng.

Với việc triển khai dự án lên môi trường Docker, nó mang lại sự linh hoạt và dễ dàng trong việc triển khai và quản lý ứng dụng. Docker cho phép đóng gói ứng dụng và các phụ thuộc của nó vào các container độc lập, giúp tạo ra một môi trường cô lập và đáng tin cậy. Điều này giúp dự án dễ dàng di chuyển và triển khai trên các máy chủ khác nhau mà không gặp vấn đề về tương thích.

Bằng cách triển khai trên đám mây, dự án có thể tận dụng các dịch vụ đám mây như việc lưu trữ dữ liệu, quản lý cơ sở dữ liệu, phân phối tải và mở rộng hệ thống dễ dàng. Điều này giúp tăng tính sẵn sàng, độ tin cậy và khả năng mở rộng của ứng dụng.

Website "Quản lý cửa hàng thiết bị điện tử" cung cấp nhiều tính năng quan trọng để quản lý hoạt động kinh doanh của cửa hàng. Dưới đây là một số tính năng chính của dự án:

- Quản lý sản phẩm: Hệ thống cho phép quản lý danh mục sản phẩm, thông tin sản phẩm, giá cả. Người dùng có thể thêm sản phầm vào giỏ hàng.

- Quản lý đơn hàng: Hệ thống cho phép ghi nhận các đơn hàng từ khách hàng, xử lý đơn hàng và theo dõi quá trình vận chuyển. Quản lý đơn hàng giúp tăng cường khả năng phục vụ khách hàng và cải thiện quy trình bán hàng.

- Quản lý khách hàng: Dự án cung cấp chức năng quản lý thông tin khách hàng, bao gồm việc ghi nhận thông tin cá nhân, lịch sử mua hàng và tương tác với khách hàng. Điều này giúp tạo và duy trì mối quan hệ tốt với khách hàng và cung cấp dịch vụ tốt hơn.

## 1.2 Mục tiêu báo cáo

Mục tiêu báo cáo cho dự án "Quản lý cửa hàng thiết bị điện tử" là cung cấp một tổng quan về dự án, từ các khía cạnh chính đến các thành phần cơ bản của nó. Báo cáo này nhằm giúp người đọc hiểu rõ về dự án, mục tiêu của nó và những thành tựu đã đạt được. Các mục tiêu chính của báo cáo bao gồm:

- Giới thiệu dự án: Trình bày mục tiêu và phạm vi của dự án "Quản lý cửa hàng thiết bị điện tử". Giới thiệu tổng quan về ngành công nghiệp và lợi ích của việc triển khai hệ thống quản lý cửa hàng hiện đại.

- Kiến trúc và công nghệ: Mô tả kiến trúc tổng quan của dự án, bao gồm việc sử dụng môi trường Docker và triển khai trên đám mây. Giới thiệu các công nghệ và framework được sử dụng trong việc phát triển dự án.

- Tính năng chính: Trình bày các tính năng quan trọng của dự án như quản lý sản phẩm, quản lý đơn hàng và quản lý khách hàng. Mô tả chi tiết về từng tính năng, cung cấp các ví dụ và hình ảnh minh họa để giúp người đọc hiểu rõ hơn về chức năng và ưu điểm của dự án.

- Lợi ích và giá trị: Đánh giá lợi ích và giá trị mà dự án mang lại cho cửa hàng thiết bị điện tử. Nêu rõ các lợi ích kinh doanh như tăng cường quản lý, cải thiện hiệu suất và nâng cao trải nghiệm khách hàng.

- Triển khai và kết quả: Đánh giá quá trình triển khai dự án lên môi trường Docker và đám mây, bao gồm các thách thức và giải pháp đã được áp dụng. Trình bày kết quả đạt được sau khi triển khai dự án, bao gồm sự ổn định, khả năng mở rộng và sự hài lòng của người dùng.

-Tương lai và đề xuất: Đề xuất các phát triển tiếp theo cho dự án, bao gồm việc nâng cấp tính năng, tối ưu hóa hiệu suất và mở rộng quy mô. Đưa ra những lợi ích và cơ hội tiềm năng khi phát triển dự án trong tương lai.

# PHÂN TÍCH YÊU CẦU

## 2.1 Yêu cầu của dự án

**Giao diện người dùng:**

- Thiết kế giao diện hấp dẫn, thân thiện với người dùng và dễ sử dụng.

- Cung cấp chức năng tìm kiếm sản phẩm dựa trên tên, loại, giá, và các tiêu chí khác.

- Hiển thị thông tin chi tiết của sản phẩm, bao gồm hình ảnh, mô tả, giá, đánh giá, và các thuộc tính khác.

- Cung cấp chức năng đánh giá và nhận xét sản phẩm từ phía người dùng.

- Cho phép người dùng thêm sản phẩm vào giỏ hàng và thực hiện thanh toán trực tuyến.

**Quản lý sản phẩm:**

- Hỗ trợ quản lý danh mục sản phẩm và danh sách sản phẩm.

- Cho phép quản lý thông tin chi tiết của sản phẩm, bao gồm hình ảnh, mô tả, giá và các thuộc tính khác.

- Hỗ trợ quản lý danh mục và thuộc tính sản phẩm (ví dụ: loại sản phẩm, nhà sản xuất, màu sắc, kích thước, v.v.).

- Tích hợp công cụ tìm kiếm và lọc để dễ dàng tìm kiếm và quản lý sản phẩm.

**Quản lý đơn hàng:**

- Hỗ trợ quản lý đơn hàng từ khi khách hàng đặt hàng cho đến khi giao hàng.

- Cung cấp thông tin chi tiết về đơn hàng, bao gồm sản phẩm, số lượng, giá, thông tin khách hàng, địa chỉ giao hàng, v.v.

- Cho phép quản lý viên cập nhật trạng thái đơn hàng (đã tiếp nhận, đang xử lý, đã giao hàng, v.v.).

- Tích hợp tính năng gửi thông báo cho khách hàng về trạng thái đơn hàng.

**Hệ thống thanh toán:**

Cung cấp chức năng giỏ hàng để khách hàng có thể xem và chỉnh sửa các sản phẩm trong giỏ hàng trước khi thanh toán.

## 2.2 Các chức năng và tính năng của dự án

### 2.2.1 User stories

**Người dùng muốn tìm kiếm sản phẩm:**

- Chức năng tìm kiếm sản phẩm dựa trên tên, thương hiệu, danh mục, giá và tính năng.

- Hiển thị kết quả tìm kiếm phù hợp với yêu cầu của người dùng.

**Người dùng muốn xem danh sách sản phẩm:**

- Hiển thị danh sách sản phẩm theo danh mục, thương hiệu hoặc các tiêu chí khác.

- Cho phép người dùng sắp xếp danh sách sản phẩm theo giá, đánh giá hoặc các tiêu chí khác.

**Người dùng muốn xem chi tiết sản phẩm:**

- Hiển thị thông tin chi tiết của sản phẩm bao gồm hình ảnh, mô tả, giá, đánh giá, thông số kỹ thuật và các thuộc tính khác.

- Cho phép người dùng đánh giá sản phẩm và viết nhận xét.

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

- Cho phép người dùng thêm sản phẩm vào giỏ hàng từ trang danh sách sản phẩm hoặc trang chi tiết sản phẩm.

- Hiển thị thông tin về số lượng sản phẩm trong giỏ hàng và tổng giá trị của giỏ hàng.

**Người dùng muốn thanh toán và hoàn tất đơn hàng:**

- Cung cấp giao diện thanh toán an toàn và tiện lợi cho người dùng.

- Xác nhận thông tin đơn hàng, phương thức thanh toán và địa chỉ giao hàng.

- Gửi thông báo xác nhận đơn hàng đến người dùng.

### 2.2.2 Features

**Tính năng tìm kiếm:**

- Người dùng có thể tìm kiếm sản phẩm dựa trên tên, thương hiệu, danh mục, giá cả và các tiêu chí khác.

- Hiển thị kết quả tìm kiếm phù hợp với yêu cầu của người dùng.

**Tính năng danh mục sản phẩm:**

- Hiển thị danh mục sản phẩm để người dùng có thể duyệt và lựa chọn sản phẩm dễ dàng hơn.

- Cho phép người dùng lọc sản phẩm theo danh mục, thương hiệu hoặc các tiêu chí khác.

**Tính năng xem chi tiết sản phẩm:**

- Hiển thị thông tin chi tiết của sản phẩm bao gồm hình ảnh, mô tả, giá, đánh giá, thông số kỹ thuật và các thuộc tính khác.

- Cho phép người dùng đánh giá và viết nhận xét về sản phẩm.

**Tính năng giỏ hàng:**

- Cho phép người dùng thêm sản phẩm vào giỏ hàng từ trang danh sách sản phẩm hoặc trang chi tiết sản phẩm.

- Hiển thị thông tin về số lượng sản phẩm trong giỏ hàng và tổng giá trị của giỏ hàng.

- Cho phép người dùng chỉnh sửa số lượng sản phẩm trong giỏ hàng.

**Tính năng thanh toán và đặt hàng:**

- Cung cấp giao diện thanh toán an toàn và tiện lợi cho người dùng.

- Xác nhận thông tin đơn hàng, phương thức thanh toán và địa chỉ giao hàng.

- Gửi thông báo xác nhận đơn hàng đến người dùng sau khi đặt hàng thành công.

**Tính năng quản lý sản phẩm:**

- Cho phép quản trị viên thêm, sửa, xóa sản phẩm từ giao diện quản trị.

- Quản lý danh mục sản phẩm và các thuộc tính sản phẩm.

- Tích hợp công cụ tải lên hình ảnh và quản lý thư viện hình ảnh.

**Tính năng quản lý đơn hàng:**

- Hiển thị danh sách đơn hàng và chi tiết đơn hàng cho quản trị viên.

- Cập nhật trạng thái đơn hàng (đã tiếp nhận, đang xử lý, đã giao hàng, v.v.).

- Gửi thông báo đến người dùng về trạng thái đơn hàng.

**Tính năng quản lý người dùng:**

- Đăng ký tài khoản cho người dùng mới.

- Đăng nhập và xác thực người dùng.

- Quản lý thông tin cá nhân của người dùng, bao gồm thông tin liên hệ, địa chỉ giao hàng và lịch sử mua hàng.

**Tính năng đánh giá và nhận xét sản phẩm:**

- Cho phép người dùng đánh giá và viết nhận xét về sản phẩm đã mua.

- Hiển thị đánh giá và nhận xét của người dùng khác để giúp người dùng có quyết định mua hàng tốt hơn.

# THIẾT KẾ KIẾN TRÚC

## 3.1 Kiến trúc hệ thống

### 3.1.1 Microservices

Kiến trúc Microservices là một mô hình phát triển ứng dụng phân tán, trong đó ứng dụng được chia thành các thành phần nhỏ độc lập gọi là microservices.

Mỗi microservice đảm nhận một nhiệm vụ cụ thể và có thể được phát triển, triển khai và quản lý độc lập với các microservice khác.

Các microservice giao tiếp với nhau thông qua các giao thức như HTTP, gRPC hoặc message queues để hoàn thành các quy trình kinh doanh phức tạp hơn.

Kiến trúc Microservices giúp tăng tính linh hoạt, khả năng mở rộng và độc lập của các thành phần hệ thống, giúp dễ dàng triển khai và mở rộng mỗi microservice riêng biệt.

### 3.1.2 SOA (Service-Oriented Architecture)

Kiến trúc SOA tập trung vào việc tổ chức ứng dụng thành các dịch vụ độc lập, được cung cấp thông qua giao diện chuẩn như API.

Mỗi dịch vụ (service) đảm nhận một nhiệm vụ cụ thể và thực hiện các chức năng kinh doanh liên quan.

Các dịch vụ giao tiếp với nhau qua các giao thức tiêu chuẩn như SOAP hoặc REST, đồng thời họ có thể được tái sử dụng và kết hợp để xây dựng các ứng dụng lớn hơn.

Kiến trúc SOA giúp tách rời các chức năng kinh doanh, tăng khả năng mở rộng, và tạo ra môi trường linh hoạt cho việc triển khai và mở rộng hệ thống.

## 3.2 Thiết kế giao diện người dùng

**User Story: Tìm kiếm sản phẩm**

Giao diện có ô tìm kiếm nơi người dùng có thể nhập từ khóa tìm kiếm.

Kết quả tìm kiếm được hiển thị dưới dạng danh sách sản phẩm với các thông tin cơ bản như tên, giá, và đánh giá.

Người dùng có thể sử dụng bộ lọc để tìm kiếm theo thương hiệu, danh mục, giá và tính năng.

Giao diện cung cấp sắp xếp kết quả tìm kiếm theo giá, đánh giá hoặc các tiêu chí khác.

**User Story: Xem danh sách sản phẩm**

Giao diện hiển thị danh sách sản phẩm được phân loại theo danh mục hoặc thương hiệu.

Mỗi mục sản phẩm hiển thị thông tin cơ bản như tên, giá và đánh giá.

Người dùng có thể sử dụng các tiêu chí sắp xếp để thay đổi thứ tự hiển thị.

**User Story: Xem chi tiết sản phẩm**

Khi người dùng nhấp vào một sản phẩm, giao diện hiển thị trang chi tiết sản phẩm.

Trang chi tiết sản phẩm chứa thông tin đầy đủ về sản phẩm bao gồm hình ảnh, mô tả, giá, đánh giá, và thông số kỹ thuật.

Người dùng có thể đánh giá sản phẩm và viết nhận xét.

**User Story: Thêm sản phẩm vào giỏ hàng**

Giao diện hiển thị nút "Thêm vào giỏ hàng" trên trang danh sách sản phẩm và trang chi tiết sản phẩm.

Khi người dùng nhấp vào nút "Thêm vào giỏ hàng", sản phẩm được thêm vào giỏ hàng.

Giao diện hiển thị thông tin về số lượng sản phẩm trong giỏ hàng và tổng giá trị của giỏ hàng.

**User Story: Thanh toán và hoàn tất đơn hàng**

Khi người dùng muốn thanh toán, giao diện chuyển hướng đến trang thanh toán.

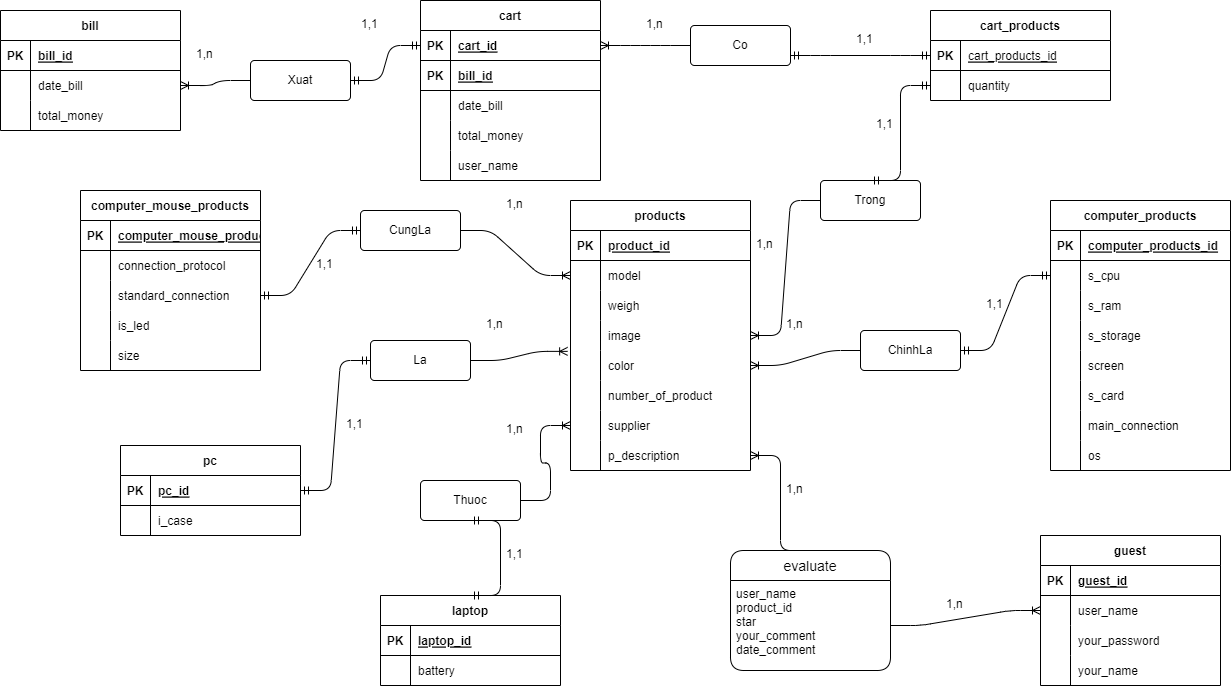
Giao diện hiển thị thông tin đơn hàng, phương thức thanh toán và địa chỉ giao hàng.

Người dùng điền thông tin cần thiết và xác nhận đơn hàng.

Sau khi xác nhận, giao diện hiển thị thông báo xác nhận đơn hàng và gửi email xác nhận đến người dùng.

## 3.3 Thiết kế cơ sở dữ liệu

### 3.3.1 Lược đồ mức vật lý



Hình 3‑1 Lược đồ mức vật lý

### 3.3.2 Vai trò các bảng

- **Bảng "bill":**

Vai trò: Lưu thông tin về hóa đơn (bill) trong quản lý cửa hàng thiết bị điện tử.

Các cột: bill\_id (khóa chính), cart\_id (khóa ngoại đến bảng "cart"), date\_bill, total\_money.

- **Bảng "cart":**

Vai trò: Lưu thông tin về giỏ hàng (cart) của người dùng.

Các cột: cart\_id (khóa chính), user\_name.

- **Bảng "cart\_products":**

Vai trò: Lưu thông tin về các sản phẩm trong giỏ hàng.

Các cột: cart\_id (khóa ngoại đến bảng "cart"), product\_id (khóa ngoại đến bảng "products"), quantity.

- **Bảng "computer\_mouse\_products":**

Vai trò: Lưu thông tin về các sản phẩm chuột máy tính.

Các cột: computer\_mouse\_product\_id (khóa chính), product\_id (khóa ngoại đến bảng "products"), standard\_connection, connection\_protocol, is\_led, size.

- **Bảng "computer\_products":**

Vai trò: Lưu thông tin về các sản phẩm máy tính.

Các cột: computer\_product\_id (khóa chính), product\_id (khóa ngoại đến bảng "products"), s\_cpu, s\_ram, s\_storage, screen, s\_card, main\_connection, os.

- **Bảng "evaluate":**

Vai trò: Lưu thông tin về đánh giá sản phẩm từ người dùng.

Các cột: user\_name (khóa ngoại đến bảng "guest"), product\_id (khóa ngoại đến bảng "products"), star, your\_comment, date\_comment.

- **Bảng "guest":**

Vai trò: Lưu thông tin về người dùng (khách hàng).

Các cột: guest\_id (khóa chính), user\_name, your\_password, your\_name.

- **Bảng "laptop":**

Vai trò: Lưu thông tin về các sản phẩm laptop.

Các cột: laptop\_id (khóa chính), product\_id (khóa ngoại đến bảng "products"), battery.

- **Bảng "pc":**

Vai trò: Lưu thông tin về các sản phẩm máy tính để bàn.

Các cột: pc\_id (khóa chính), product\_id (khóa ngoại đến bảng "products"), i\_case.

- **Bảng "products":**

Vai trò: Lưu thông tin về các sản phẩm trong cửa hàng.

Các cột: product\_id (khóa chính), model, image, weigh, color, number\_of\_product, supplier, p\_description.

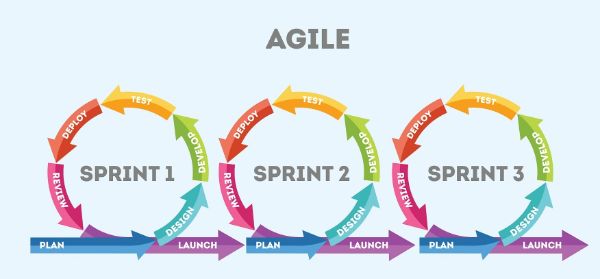
# QUY TRÌNH PHÁT TRIỂN

## 4.1 Phương pháp Agile

### 4.1.1 Aglie 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.

Phát triển phần mềm linh hoạt (agile software development – gọi tắt là Agile) là một triết lí cùng với nhóm các phương pháp và phương pháp luận phát triển phần mềm dựa trên các nguyên tắc phát triển phân đoạn lặp và tăng trưởng, theo đó nhu cầu và giải pháp tiến hóa thông qua sự hợp tác giữa các nhóm tự quản và liên chức năng. Agile thường sử dụng cách lập kế hoạch thích ứng, việc phát triển và chuyển giao theo hướng tiến hóa; sử dụng các khung thời gian ngắn và linh hoạt để dễ dàng phản hồi lại với các thay đổi trong quá trình phát triển.



Hình 4‑1 Quy trình Agile

### 4.1.2 Đặc điểm của Agile

- Tính lặp (Iterative): Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại. Các phân đoạn (được gọi là Iteration hoặc Sprint) này thường có khung thời gian ngắn (từ một đến bốn tuần). Trong mỗi phân đoạn này, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử (với các mức độ khác nhau) để cho ra các phần nhỏ của sản phẩm. Các phương pháp agile thường phân chia mục tiêu thành các phần nhỏ với quá trình lập kế hoạch đơn giản và gọn nhẹ nhất có thể, và không thực hiện việc lập kế hoạch dài hạn.

- Tính tiệm tiến (Incremental) và tiến hóa (Evolutionary): Cuối các phân đoạn, nhóm phát triển thường cho ra các phần nhỏ của sản phẩm cuối cùng. Các phần nhỏ này thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng ngay. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn. Khác với mô hình phát triển Thác nước – vốn chỉ cho phép nhìn thấy toàn bộ các chức năng tại thời điểm kết thúc dự án, sản phẩm trong các dự án agile lớn dần lên theo thời gian, tiến hóa cho tới khi đạt được trạng thái đủ để phát hành.

- Tính thích ứng (adaptive): Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn, và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển đều có thể được đáp ứng theo cách thích hợp . Ví dụ, trong Scrum – phương pháp phổ biến nhất hiện nay – trong khi nhóm phát triển sản xuất ra các gói phần mềm, khách hàng có thể đưa thêm các yêu cầu mới, chủ sản phẩm (Product Owner) có thể đánh giá các yêu cầu này và có thể đưa vào làm việc trong phân đoạn (được gọi là Sprint trong Scrum) tiếp theo. Theo đó, các quy trình agile thường thích ứng rất tốt với các thay đổi.

- Tính tự tổ chức (self-organizing) và liên chức năng (cross-functionality): Cấu trúc nhóm agile thường là liên chức năngvà tự tổ chứcTheo đó, các nhóm này tự thực hiện lấy việc phân công công việc mà không dựa trên các mô tả cứng về chức danh hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Các nhóm này cộng tác với nhau để ra quyết định, theo dõi tiến độ, giải quyết các vấn đề mà không chờ mệnh lệnh của các cấp quản lý. Họ không làm việc theo cơ chế “mệnh lệnh và kiểm soát”.

- Quản lý tiến trình thực tiễn (Empirical Process Control): Các nhóm agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiền giả định. Việc phân nhỏ dự án thành các phân đoạn ngắn góp phần gia tăng các điểm mốc để nhóm phát triển thu thập dữ kiện cho phép điều chỉnh các chiến lược phát triển của mình. Nói cách khác, Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt. Theo thời gian, các chiến lược này sẽ tiến gần đến trạng thái tối ưu, nhờ đó nhóm có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

- Giao tiếp trực diện (face-to-face communication): Về yêu cầu của khách hàng, agile khuyến khích nhóm phát triển trực tiếp nói chuyện với khách hàng để hiểu rõ hơn về cái khách hàng thực sự cần, thay vì phụ thuộc nhiều vào các loại văn bản. Trong giao tiếp giữa nội bộ nhóm phát triển với nhau, thay vì một lập trình viên (thực hiện việc code) và một kĩ sư (thực hiện việc thiết kế) giao tiếp với nhau thông qua bản thiết kế, agile khuyến khích hai người này trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

- Phát triển dựa trên giá trị (value-based development): Để vận hành được cơ chế “làm việc dựa trên giá trị”, nhóm agile thường làm việc trực tiếp và thường xuyên với khách hàng (hay đại diện của khách hàng), cộng tác trực tiếp với họ để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án. Nhờ đó các dự án agile thường giúp khách hàng tối ưu hóa được giá trị của dự án. Một cách gần như trực tiếp, agile gia tăng đáng kể độ hài lòng của khách hàng.

**\* 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.

## 4.2 Phương pháp Scrum

### 4.2.1 Scrum là gì?

Scrum là một khung quản lý được các đội ngũ sử dụng để tự tổ chức và hoạt động vì một mục tiêu chung. Khung này mô tả một loạt các cuộc họp, công cụ và vai trò để bàn giao dự án hiệu quả. Cũng giống như một đội thể thao tập dượt cho trận đấu quan trọng, thực hành Scrum tạo điều kiện cho đội ngũ tự quản lý, học hỏi từ kinh nghiệm và thích nghi với sự thay đổi. Các đội ngũ phát triển phần mềm sử dụng Scrum để giải quyết những vấn đề phức tạp một cách tiết kiệm và bền vững.

### 4.2.2 Đặc điểm của Scrum

- Tính minh bạch: Các khía cạnh quan trọng của quy trình được xác định bởi một quy chuẩn chung và phải được hiển thị minh bạch, rõ ràng cho những người chịu trách nhiệm về kết quả.

- Tính kiểm tra: Để phát hiện những sai sót không mong muốn, việc kiểm tra thường xuyên là vô cùng quan trọng.

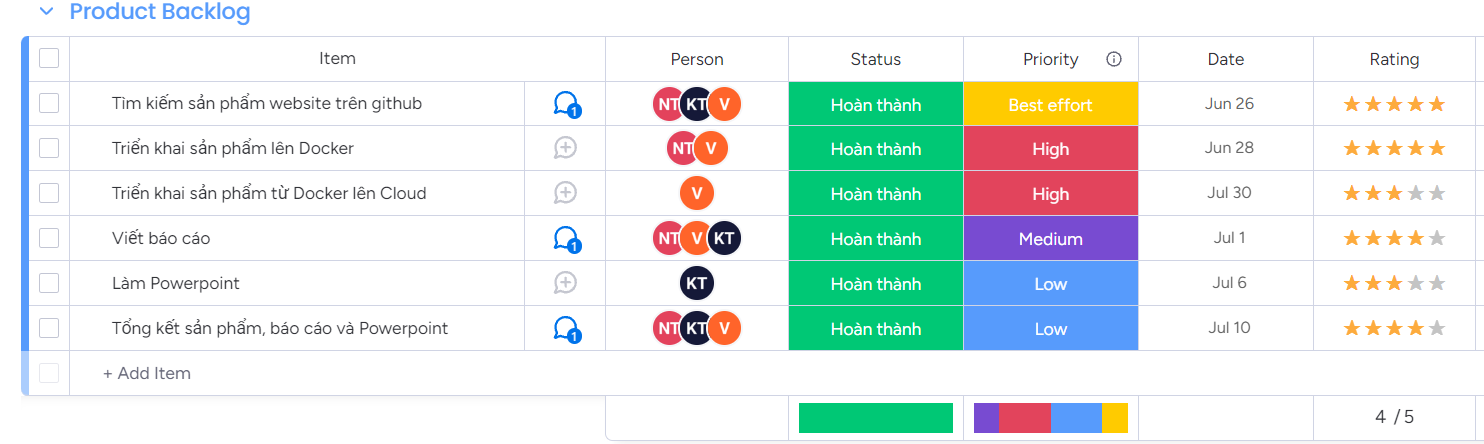
- Tính thích ứng: Sau khi kiểm tra, nếu một hoặc nhiều khía cạnh được xác định là sai, đi chệch hướng so với mục tiêu của sprint và có thể sản phẩm không được chấp nhận thì các khía cạnh cần được thay đổi càng sớm càng tốt.

## 4.3 Sơ đồ quy trình phát triển

### 4.3.1 Lập Product backlog

- Mục đích: Quản lý các yêu cầu và tính năng của dự án.

- Bảng:



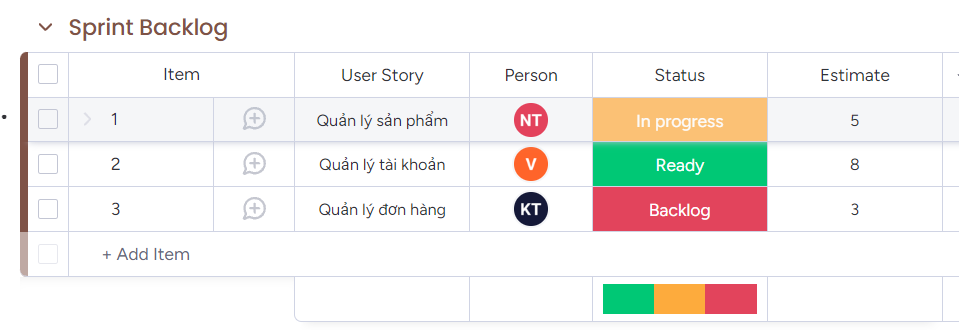
Hình 4‑2 Product backlog

### 4.3.1 Lập Sprint

**- Sprint Backlog.**

- Mục đích: Quản lý các User Story và công việc trong mỗi Sprint.

- Bảng:

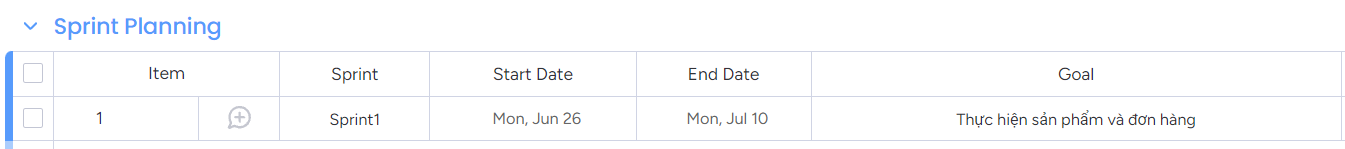


Hình 4‑3 Sprint Backlog

**- Sprint Planning.**

- Mục đích: Lên kế hoạch cho mỗi Sprint.

- Bảng:

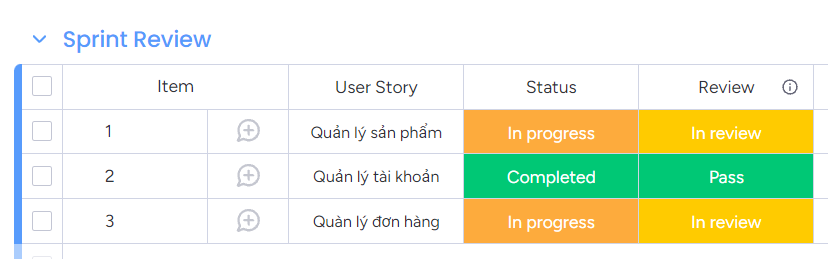


Hình 4‑4 Sprint Planning

**- Sprint Review.**

- Mục đích: Xem xét và đánh giá kết quả của mỗi Sprint.

- Bảng:

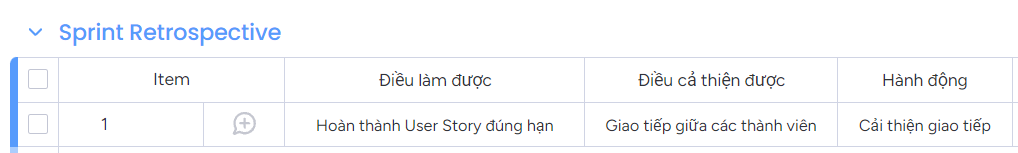


Hình 4‑5 Sprint Review

**- Sprint Retrospective.**

- Mục đích: Đánh giá và rút ra bài học từ mỗi Sprint.

- Bảng:

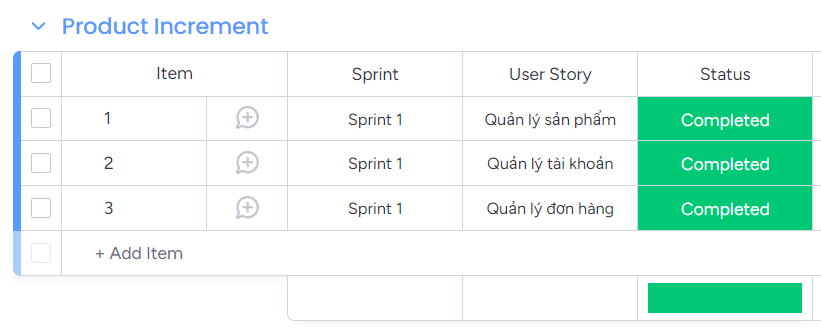


Hình 4‑6 Sprint Retrospective

**- Product Increment.**

- Mục đích: Kiểm tra và xác nhận các công việc đã hoàn thành trong mỗi Sprint.

- Bảng:



Hình 4‑7 Product Increment

## 4.4 Công cụ hỗ trợ phát triển

- **Công cụ lập trình:**

VSCode (Visual Studio Code): Đây là một trình chỉnh sửa mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các tính năng mạnh mẽ như gợi ý mã, gỡ lỗi, kiểm tra cú pháp, và tích hợp Git.

- **Cơ sở dữ liệu:**

Cơ sở dữ liệu có thể được xây dựng bằng sử dụng hệ quản trị cơ sở dữ liệu quan hệ như MySQL, PostgreSQL hoặc SQL Server. CSDL này sẽ được sử dụng để lưu trữ thông tin về sản phẩm, đơn hàng, người dùng và các dữ liệu khác liên quan.

- **Ngôn ngữ và công nghệ:**

Ngôn ngữ phía máy chủ: Sử dụng ngôn ngữ như JavaScript, PHP để phát triển phía máy chủ.

Ngôn ngữ phía máy khách: Sử dụng HTML, CSS và JavaScript để xây dựng giao diện người dùng phía máy khách.

Framework: Có thể sử dụng các framework như Laravel (cho PHP) để tăng cường hiệu suất và tốc độ phát triển.

- **Công cụ hỗ trợ SCM và làm việc cộng tác:**

Git: Đây là hệ thống quản lý phiên bản phổ biến được sử dụng để quản lý mã nguồn và kiểm soát phiên bản.

GitHub: Cung cấp nền tảng để lưu trữ mã nguồn, quản lý dự án và làm việc cộng tác với đội ngũ phát triển.

Công cụ quản lý dự án: Monday.com là một công cụ quản lý dự án trực tuyến mạnh mẽ và linh hoạt. Nó cung cấp các tính năng hỗ trợ quản lý công việc, lịch trình, giao tiếp và cộng tác trong một giao diện trực quan.

# TRIỂN KHAI

## 5.1 Môi trường triển khai

### 5.1.1 Docker

Docker là một nền tảng cho developers và sysadmin để develop, deploy và run application với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.

### 5.1.2 Cloud

Dịch vụ Cloud là cách sử dụng phần mềm, lưu trữ, cơ sở dữ liệu, máy chủ, mạng và các dịch vụ máy tính khác. Thông qua các máy chủ có thể được truy cập thông qua Internet.

## 5.2 Quy trình

### 5.2.1 Quy trình đưa lên Docker

1. Chuẩn bị Docker: Đầu tiên, cần cài đặt Docker trên máy tính của mình. Truy cập trang web chính thức của Docker (https://www.docker.com/) và tải xuống phiên bản phù hợp với hệ điều hành của máy tính. Tiến hành cài đặt Docker theo hướng dẫn đi kèm.

2. Xây dựng Dockerfile: Tạo một tệp Dockerfile để định nghĩa cấu trúc của container. Dockerfile chứa các hướng dẫn để cài đặt các thành phần cần thiết và cấu hình môi trường để chạy website. Cần đảm bảo rằng Dockerfile của bạn đúng cú pháp và đáp ứng yêu cầu của website.

3. Xây dựng ảnh Docker: Sử dụng lệnh docker build để xây dựng ảnh Docker từ Dockerfile. Lệnh này sẽ thực thi các hướng dẫn trong Dockerfile và tạo ra một ảnh Docker hoàn chỉnh.

4. Chạy container từ ảnh Docker: Sử dụng lệnh docker run để chạy container từ ảnh Docker đã được xây dựng.

5. Kiểm tra website: Sau khi container chạy thành công, có thể truy cập vào website bằng cách sử dụng địa chỉ IP và cổng đã chỉ định.

### 5.2.2 Quy trình đưa lên CloudFly

Bước 1: Truy cập vào my.cloudfly.vn để đăng nhập vào tài khoản đã tạo tại CloudFly

Bước 2: Tại Dashboard - Bảng điều khiển, click vào “Dùng thử” để có mức nạp hỗ trợ dùng thử tối thiểu là 20,000VNĐ vào ví CloudFly

Bước 3: Sau khi thao tác thanh thanh toán và chờ cho hệ thống cập nhật số dư, tiếp tục bấm vào “Dùng thử” để bắt đầu tạo máy dùng thử (mỗi tài khoản chỉ duy nhất 01 lần tạo dùng thử) hoặc Public Cloud - Instances => tạo máy chủ tính phí.

## 5.3 Các bước triển khai

### 5.3.1 Các bước triển khai website lên Docker

**Bước 1: Chuẩn bị môi trương**

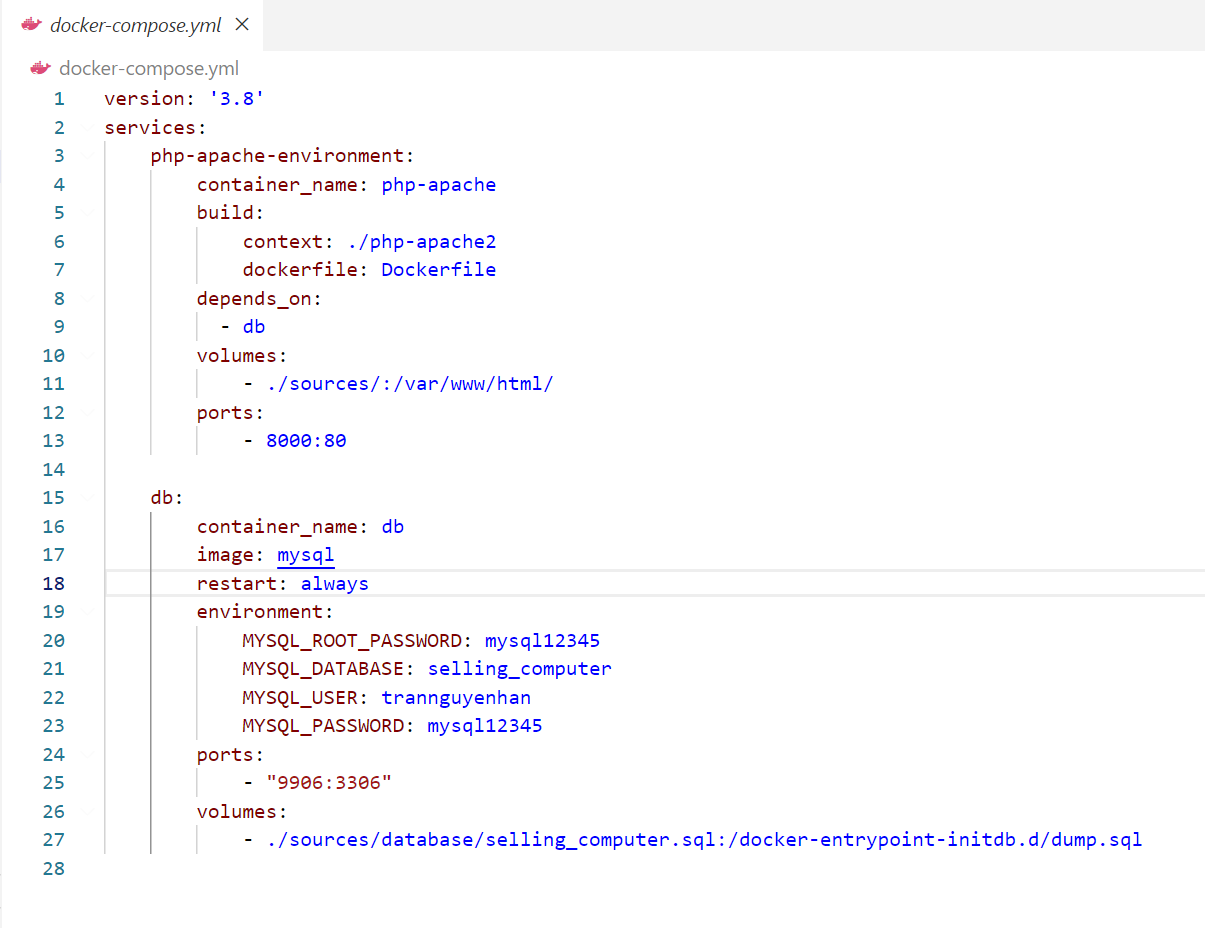
- Cài đặt Docker Desktop

- Đảm bảo rằng mã nguồn của website đã được chuẩn bị và sẵn sàng để triển khai

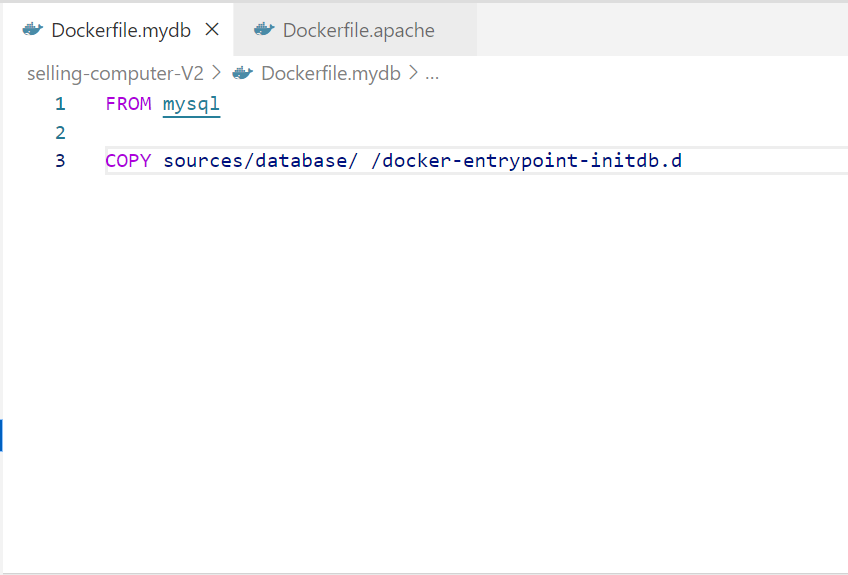
**Bước 2: Tạo Dockerfile hoặc docker-compose**

- Tạo một tệp Dockerfile hoặc docker-compose.yml để định nghĩa cấu trúc của container Docker. Dockerfile hoặc docker-compose.yml chứa các hướng dẫn để cài đặt các thành phần cần thiết và cấu hình môi trường để chạy website của bạn.

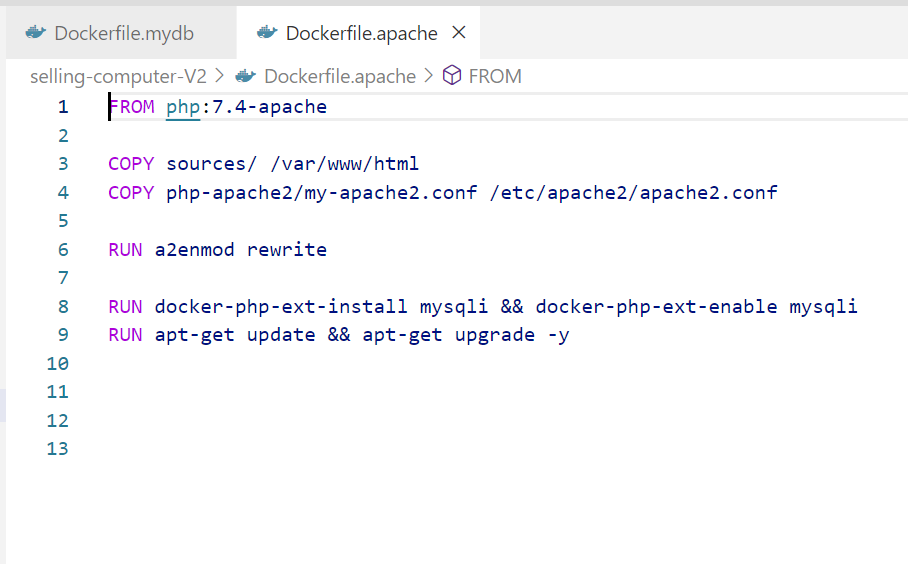
- Đảm bảo rằng Dockerfile bao gồm các bước cần thiết như cài đặt hệ điều hành, cài đặt ngôn ngữ lập trình và các thư viện phụ thuộc, cấu hình môi trường, và sao chép mã nguồn vào container.

****

Hình 5‑1 Docker-compose.yml

****

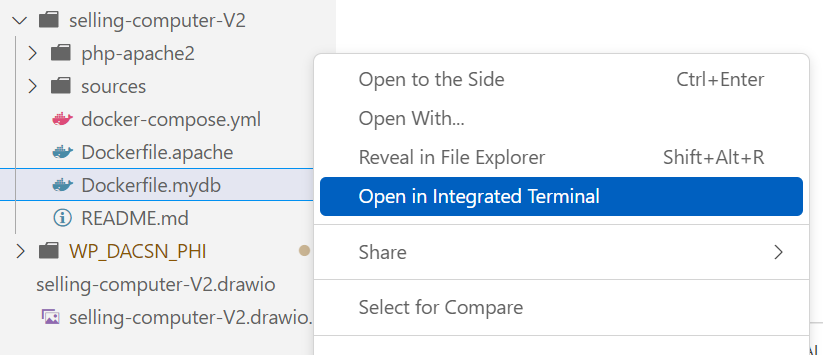
Hình 5‑2 Dockerfile.mydb

****

Hình 5‑3 Dockerfile.apache

**Bước 3: Tiến hành cài đặt**

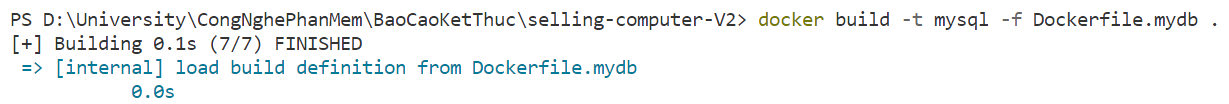
- Click chuột phải vào Dockerfile.mydb mở terminal.



Hình 5‑4 Mở terminal trên Dockerfile.mydb

- Ở terminal của Dockerfile.mydb tiến hành chạy lệnh:

+ docker build -t mysql -f Dockerfile.mydb .

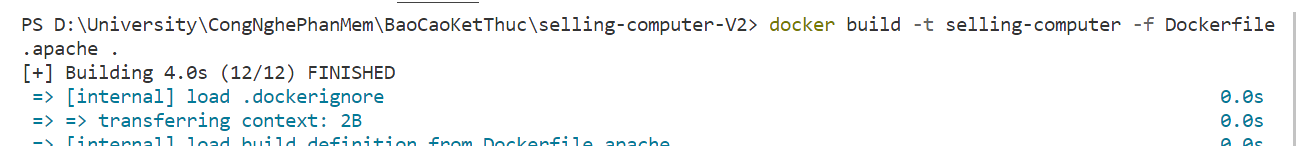


Hình 5‑5 Chạy lệnh docker build -t mysql -f Dockerfile.mydb .

- Click chuột phải vào Dockerfile.apache mở Terminal.

- Ở terminal của Dockerfile.apache chạy lệnh:

+ docker build -t selling-computer -f Dockerfile.apache .

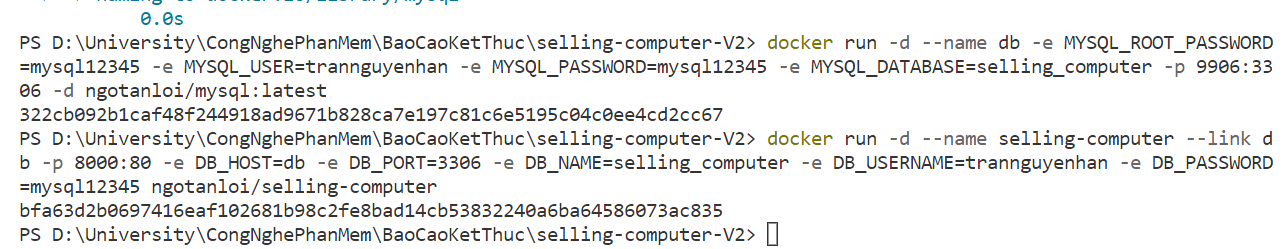


Hình 5‑6 Chạy lệnh docker build -t selling-computer -f Dockerfile.apache .

- Sau đó lần lượt chạy 2 lệnh:

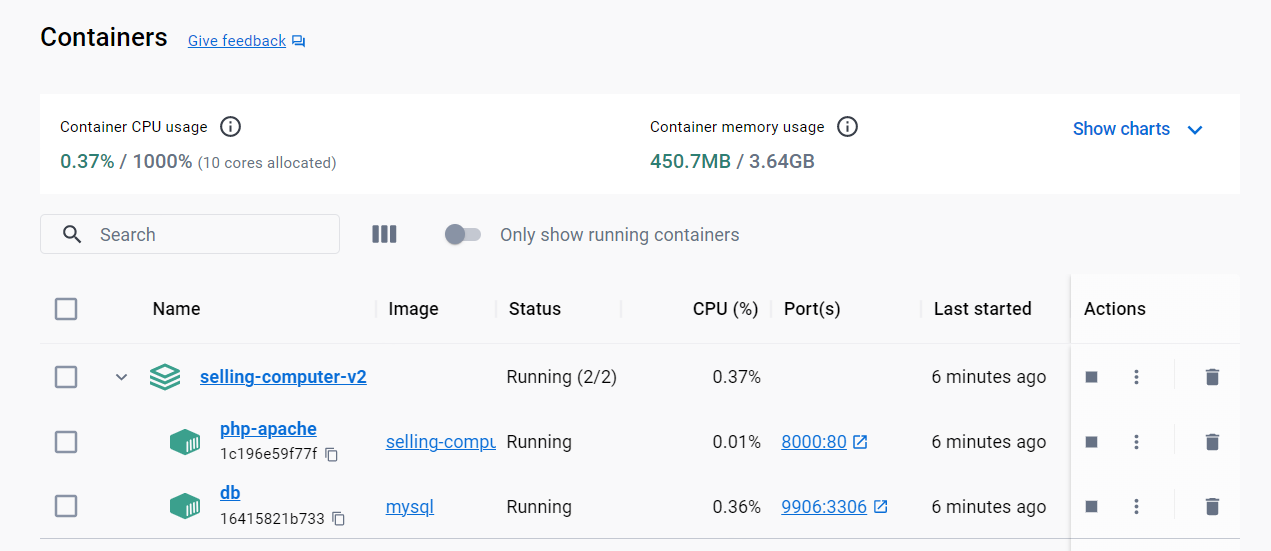
+ docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=mysql12345 -e MYSQL\_USER=trannguyenhan -e MYSQL\_PASSWORD=mysql12345 -e MYSQL\_DATABASE=selling\_computer -p 9906:3306 -d ngotanloi/mysql:latest

+ docker run -d --name selling-computer --link db -p 8000:80 -e DB\_HOST=db -e DB\_PORT=3306 -e DB\_NAME=selling\_computer -e DB\_USERNAME=trannguyenhan -e DB\_PASSWORD=mysql12345 ngotanloi/selling-computer



Hình 5‑7 Chạy hai lệnh yêu cầu

- Đợi một khoảng thời gian, sau đó giao diện docker desktop sẽ xuất hiện:



Hình 5‑8 Giao diện docker desktop sau khi chạy xong

**Bước 4: Kiểm tra website:**

- Truy cập vào địa chỉ IP hoặc tên miền của máy chủ Docker để kiểm tra website đã được triển khai thành công.

- Đảm bảo rằng các chức năng và tính năng của website hoạt động đúng trong môi trường Docker.



Hình 5‑9 Website hoạt động sau khi truy cập vào địa chỉ IP trên docker desktop

### 5.3.2 Các bước triển khai website lên CloudFly

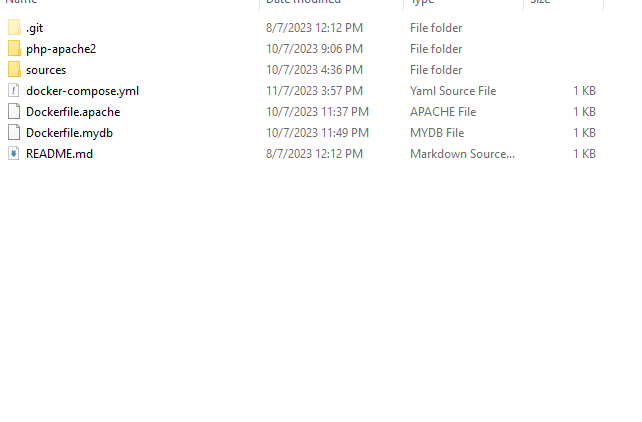
* **Các bước build dự án thành docker images:**

- Vào terminal, chuyển đến nơi lưu trữ dữ án

- Build images: sử dụng lệnh:

+ docker build -t mysql -f Dockerfile.mydb .

+ docker build -t selling-computer -f Dockerfile.apache .



Hình 5‑10 Nơi lưu dự án

- Đẩy image lên dockerhub:

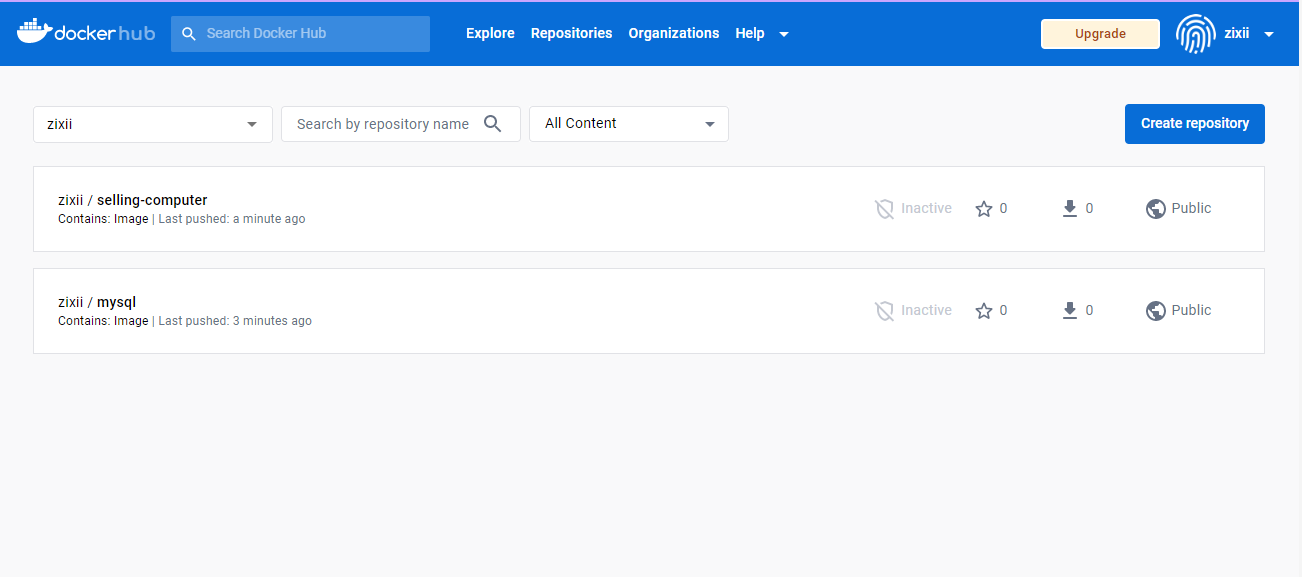
+ docker tag mysql zixii/mysql

+ docker tag selling-computer zixii/selling-computer

+ docker push zixii/mysql

+ docker push zixii/selling-computer

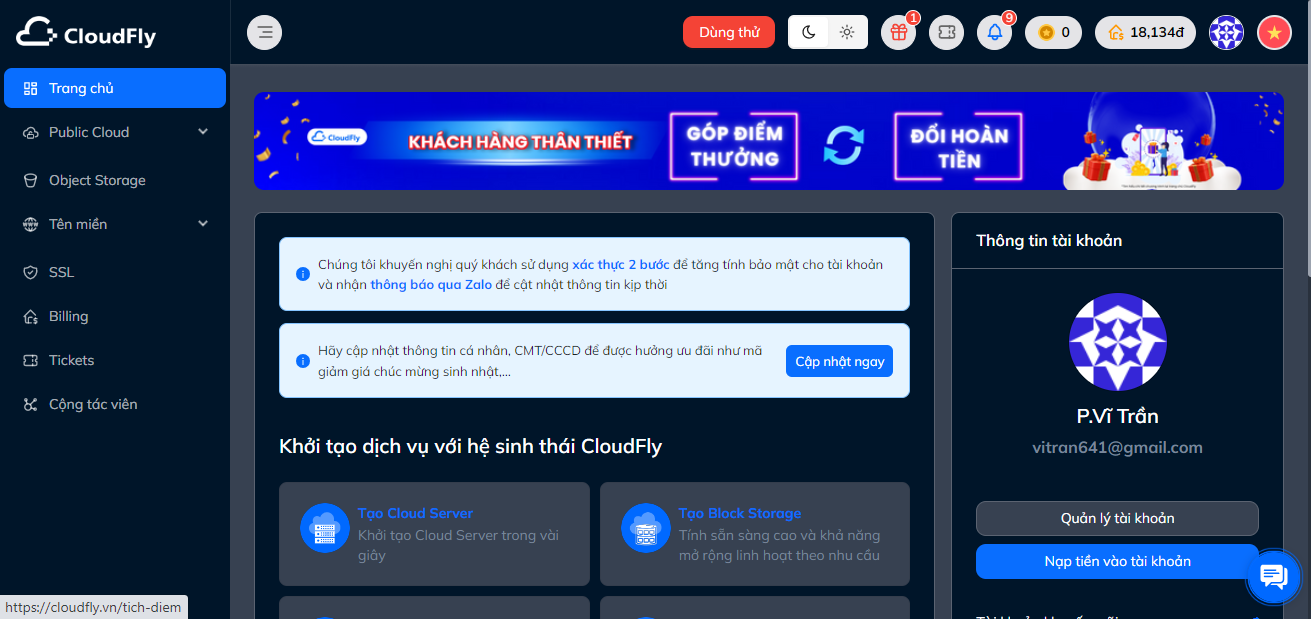
- Truy cập vào https://hub.docker.com/ để kiểm tra



Hình 5‑11 Giao diện DockerHub khi push thành công

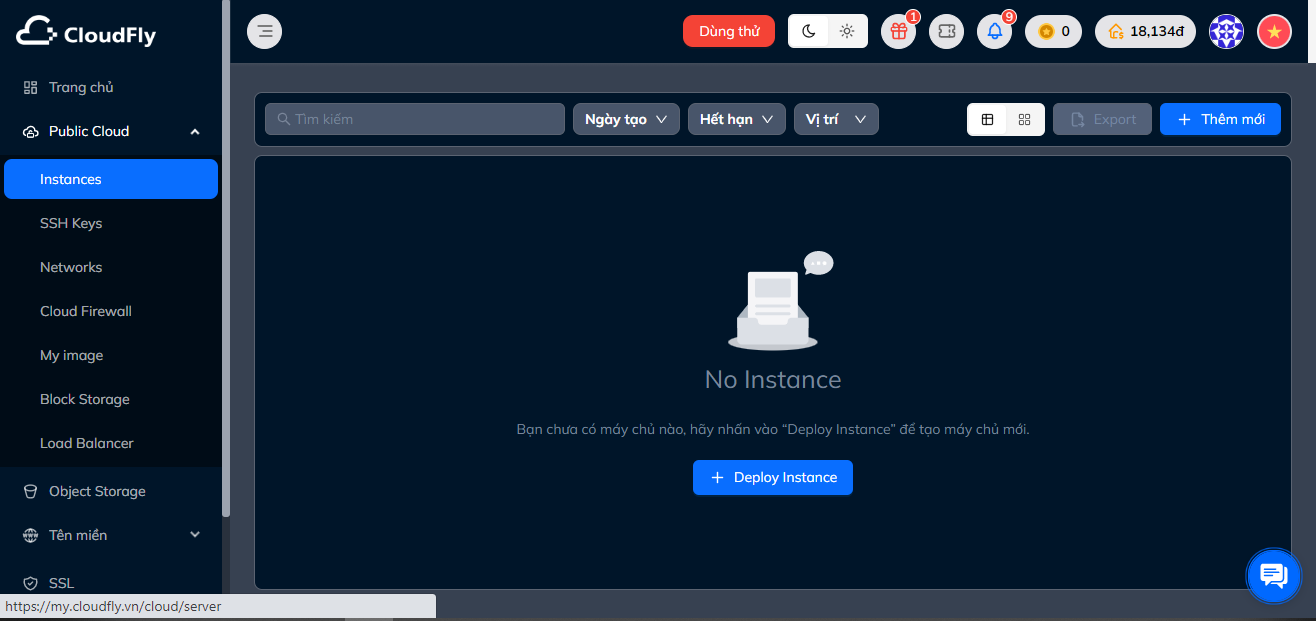
* **Các bước triển khai lên CloudFly**

**Bước 1:** Đăng ký tài khoản tại <https://my.cloudfly.vn/>



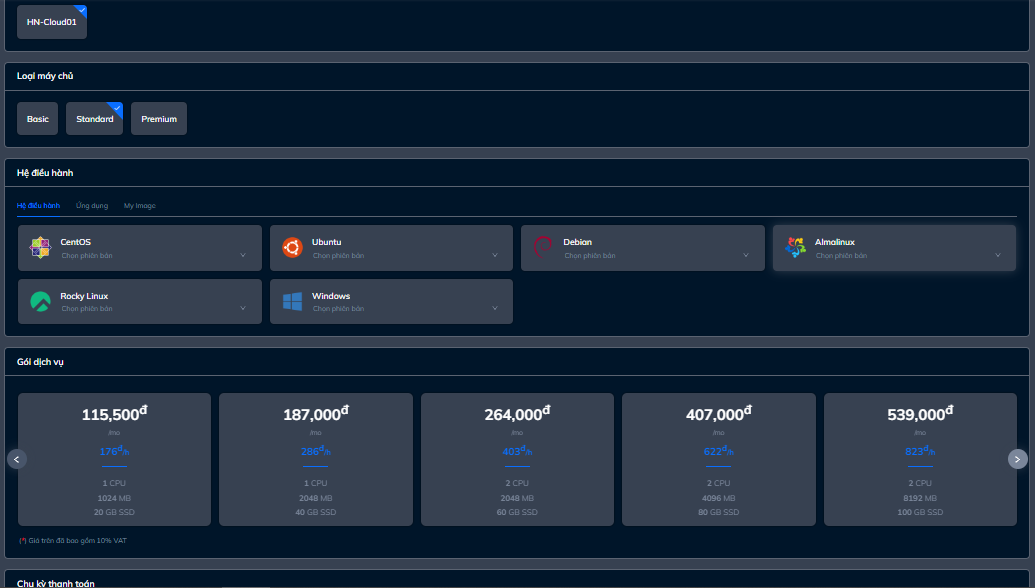
Hình 5‑12 Giao diện trang chủ CloudFly

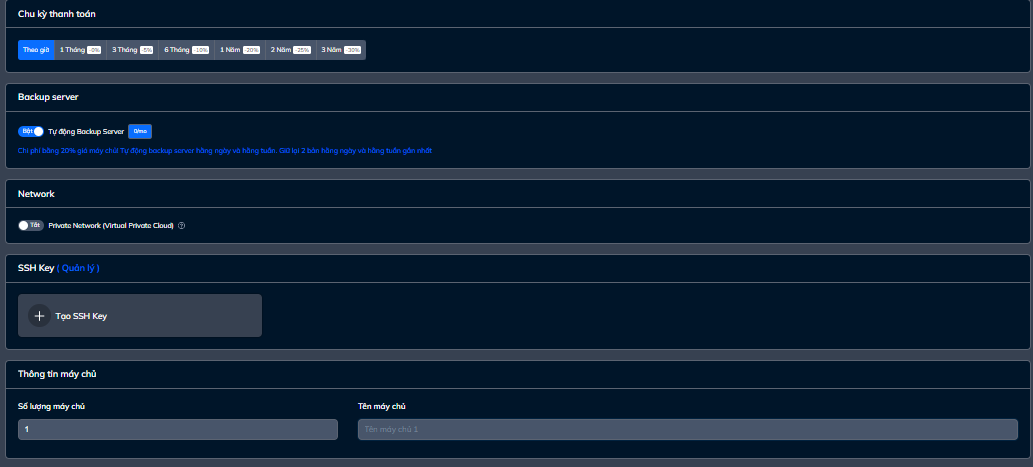
**Bước 2:** Chọn Public Cloud -> chọn Instances -> chọn thêm mới



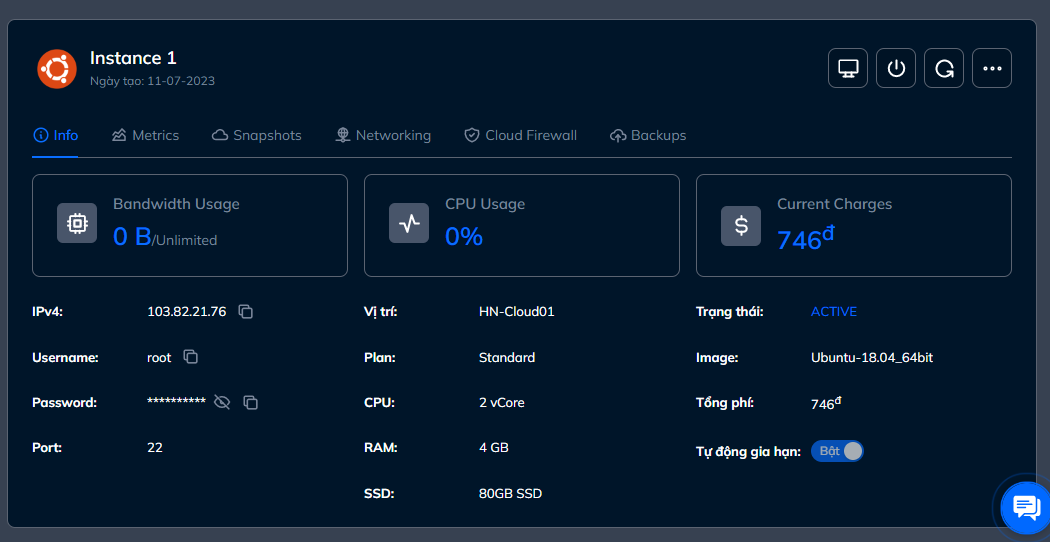
Hình 5‑13 Giao diện Instances

**Bước 3:** Chọn hệ điều hành Ubuntu 18.04, Gói dịch vụ tuỳ theo website muốn triển khai, chọn chu kỳ thanh toán theo giờ, nhập tên máy chủ vào và nhấn tạo máy chủ.





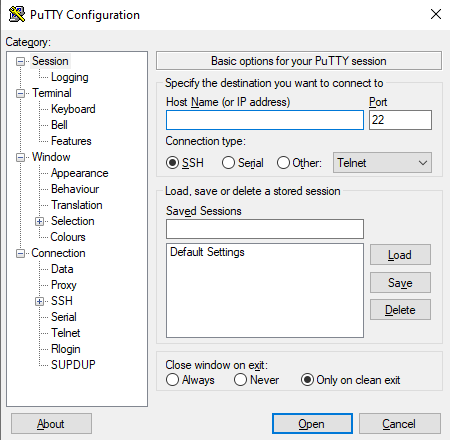
Hình 5‑14 Các bước tạo Cloud



Hình 5‑15 Giao diện khi tạo thành công

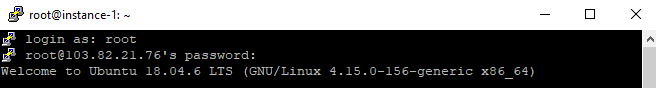
**Bước 4:** Tải Putty theo đường dẫn https://www.putty.org/

Khởi động Putty lên và nhập địa chỉ IPv4 vào Host Name và nhân Open



Hình 5‑16 Giao diện Putty

**Bước 5:** Đăng nhập Username và Password mà Cloud cung cấp



Hình 5‑17 Đăng nhập Putty

**Bước 6:** Cập nhật các gói phần mềm, bằng lệnh sau:

- sudo apt update

- sudo apt upgrade

**Bước 7:** Cài đặt các gói phụ thuộc cần thiết để cài đặt Docker:

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

**Bước 8:** 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

**Bước 9:** Thêm kho lưu trữ Docker vào hệ thống:

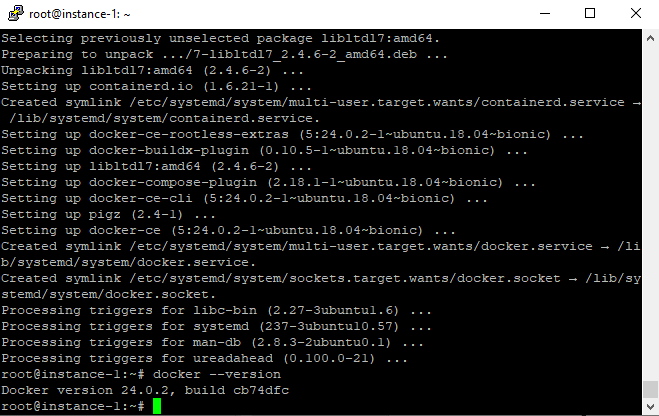
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**Bước 10:** 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: docker –version

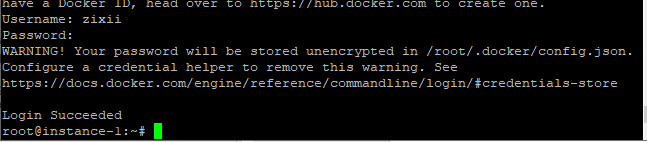


Hình 5‑18 Cài đặt Docker thành công

**Bước 11:** Sau khi chạy xong đăng nhập vào DockerHub:

- docker login

- Nhập Username và Password của DockerHub vào



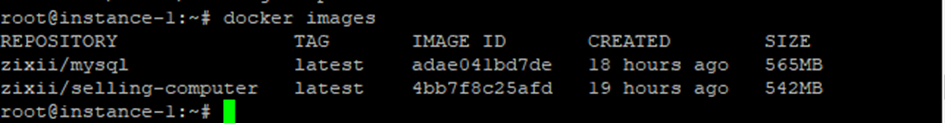
Hình 5‑19 Đăng nhập DockerHub thành công

**Bước 12:** Kéo các images đã đẩy lên dockerhub về Cloud bằng lệnh:

- sudo docker pull zixii/mysql

- sudo docker pull zixii/selling-computer

- Kiểm tra xem images đã được kéo về chưa: docker images



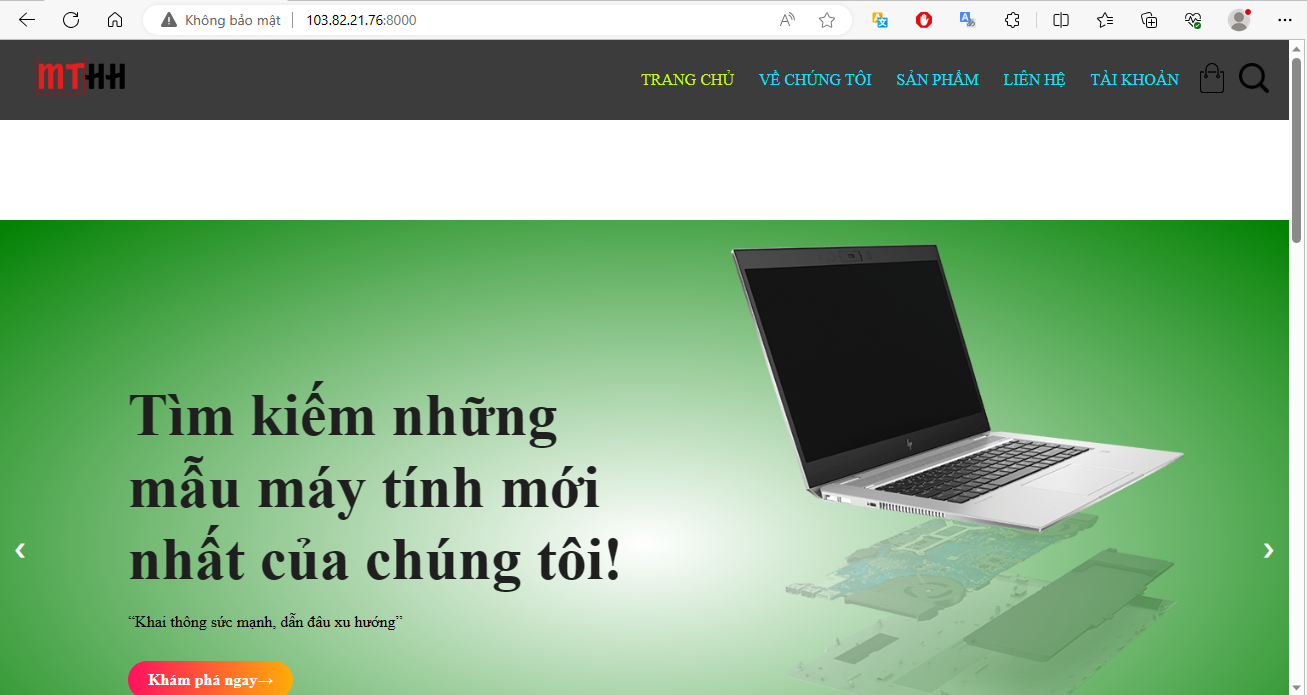
Hình 5‑20 Images đã được kéo về

**Bước 13:** chạy các container của các images đã kéo về:

- sudo docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=mysql12345 -e MYSQL\_USER=trannguyenhan -e MYSQL\_PASSWORD=mysql12345 -e MYSQL\_DATABASE=selling\_computer -p 9906:3306 -d zixii/mysql:latest

- sudo docker run -d --name selling-computer --link db -p 8000:80 -e DB\_HOST=db -e DB\_PORT=3306 -e DB\_NAME=selling\_computer -e DB\_USERNAME=trannguyenhan -e DB\_PASSWORD=mysql12345 zixii/selling-computer

**Bước 14:** Sau khi đã chạy lệnh bên trên, dùng địa chỉ IPv4 của Cloud + port để truy cập vào Web.



Hình 5‑21 Truy cập thành công

# KẾT LUẬN

## 6.1 Kết quả đạt được

- Nắm vững được kiến thức lý thuyết cơ bản phục vụ cho việc thiết kế phần mềm.

- Triển khai thành công một website lên docker, hiểu được các thành phần cấu trúc của một dockerfile và docker-compose.

- Triển khai thành công một website lên cloud, hiểu được các bước triển khai.

- Áp dụng thành công quy trình Agile và Scrum trong quá trình phát triển phần mềm. Phân chia dự án thành các sprint ngắn và linh hoạt, tập trung vào việc xác định rõ vai trò, nhiệm vụ và quy trình làm việc của từng thành viên trong nhóm. Điều này giúp tăng khả năng thích ứng và linh hoạt trong việc phản hồi yêu cầu và cải thiện hiệu suất làm việc.

- Sử dụng các công cụ hỗ trợ phát triển phần mềm như draw.io để xây dựng cơ sở dữ liệu và monday.com để lập Product backlog và lập Sprint.

## 6.2 Khó khăn và bài học

* **Khó khăn, thách thức trong quá trình phát triển:**

- Việc định nghĩa và cấu hình Dockerfile đòi hỏi kiến thức về Docker và các khái niệm liên quan như image, container, volumes, ports và environment variables. Nhóm mất khá nhiều thời gian để học các kiến thức này.

- Việc sử dụng các công cụ hỗ trợ như Draw.io để thiết kế cơ sở dữ liệu, Monday.com để lập các Product backlog và các Sprint cũng gây khó khăn cho nhóm. Mỗi công cụ hỗ trợ có giao diện và tính năng riêng, việc học cách sử dụng và làm quen với các công cụ này gây mất thời gian và gây khó khăn ban đầu cho các thành viên trong nhóm.

* **Bài học rút ra được từ dự án:**

- Nhóm đánh giá được kiến thức về Docker và các công cụ hỗ trợ phát triển phần mềm là rất quan trọng. Nên cần đầu tư thời gian và nỗ lực để học tập, nhất là khi sử dụng các công nghệ mới. Điều này giúp tăng cường năng lực và hiệu quả làm việc của nhóm.

- Trong quá trình học và sử dụng các công cụ mới, việc hỗ trợ và chia sẻ kiến thức trong nhóm là rất quan trọng. Các thành viên luôn sẵn lòng giúp đỡ và chia sẻ kinh nghiệm của mình để giúp các thành viên trong nhóm vượt qua khó khăn ban đầu. Điều này cũng tạo ra một môi trường học tập và phát triển tích cực trong nhóm.

## 6.3 Hướng phát triển

Trong tương lại nhóm sẽ xác định các chức năng chưa hoàn thiện trong website và xác định ưu tiên của từng chức năng đó. Lập kế hoạch và phân công công việc để hoàn thiện các chức năng này, sử dụng quy trình Agile và phân chia thành các sprint ngắn. Thực hiện lập trình, kiểm thử và đánh giá chức năng để đảm bảo chất lượng và hiệu suất của website. Đánh giá và sửa lỗi trong quá trình triển khai website trên.

**TÀI LIỆU THAM KHẢO**

**[1]** <https://www.tanca.io/blog/mo-hinh-agile-la-gi-cac-phuong-phap-agile-hieu-qua-trong-quan-ly-du-an>(07/2023)

**[2]** <https://viblo.asia/p/scrum-la-gi-va-tai-sao-nen-dung-scrum-1Je5Ed6LlnL> (07/2023)

**[3]** <https://blog.haposoft.com/tich-hop-draw-io-visual-studio-code/>(07/2023)

**[4]** <https://tigosoftware.com/vi/quan-ly-cong-viec-va-van-hanh-remote-team-voi-mondaycom>(07/2023)