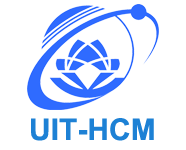
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**Môn Đồ án 1 – SE121.K21**

*Đồ án: Nghiên cứu Reactjs và xây dựng ứng dụng  
(Trang web order nước uống trực tuyến AwakeCup)*

Sinh viên thực hiện:

HUỲNH TRẦM BẢO CHẤN – 17520035

**Giảng viên hướng dẫn: Nguyễn Tấn Toàn**

MỤC LỤC

[DANH MỤC HÌNH VẼ 4](#_Toc47352139)

[DANH MỤC BẢNG 5](#_Toc47352140)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc47352141)

[Chương 1. MỞ ĐẦU 8](#_Toc47352142)

[1.1. Giới thiệu đề tài 8](#_Toc47352143)

[1.2. Lý do chọn đề tài 9](#_Toc47352144)

[Chương 2. GIỚI THIỆU TỔNG QUAN 10](#_Toc47352145)

[2.1. Mục tiêu 10](#_Toc47352146)

[2.2. Phạm vi 10](#_Toc47352147)

[2.3. Phương pháp 10](#_Toc47352148)

[2.4. Ý nghĩa thực tiễn 11](#_Toc47352149)

[2.5. Kết quả dự kiến 11](#_Toc47352150)

[Chương 3. CÔNG NGHỆ SỬ DỤNG 12](#_Toc47352151)

[3.1. ASP.NET Core 12](#_Toc47352152)

[3.1.1. Tại sao chọn ASP.NET Core [5] 12](#_Toc47352153)

[3.1.2. ASP.NET Core web APIs 13](#_Toc47352154)

[3.2. Hệ quản trị cơ sở dữ liệu MySQL/MariaDb 15](#_Toc47352155)

[3.2.1. Vì sao chọn sử dụng MySQL [9] 15](#_Toc47352156)

[3.2.2. MariaDB 16](#_Toc47352157)

[3.2.3. Sự lựa chọn của cá nhân 16](#_Toc47352158)

[3.3. ReactJs framework [10] [11] 17](#_Toc47352159)

[3.4. Apache2 web server [12] 19](#_Toc47352160)

[3.4.1. Web server 19](#_Toc47352161)

[3.4.2. Apache 19](#_Toc47352162)

[3.4.3. Một số vấn đề khác 20](#_Toc47352163)

[Chương 4. XÂY DỰNG ỨNG DỤNG 21](#_Toc47352164)

[4.1.1. User stories 21](#_Toc47352165)

[4.1.2. Danh sách User stories 21](#_Toc47352166)

[4.1.3. Mô tả chi tiết các User stories 22](#_Toc47352167)

[4.2. Thiết kế cơ sở dữ liệu 37](#_Toc47352168)

[4.2.1. Sơ đồ thực thể kết hợp ERD 37](#_Toc47352169)

[4.2.2. Danh sách các Table 37](#_Toc47352170)

[4.2.3. Chi tiết table trên MySQL 38](#_Toc47352171)

[4.3. Thiết kế kiến trúc phần mềm 42](#_Toc47352172)

[4.3.1. Kiến trúc phía backend 42](#_Toc47352173)

[4.3.2. Kiến trúc phía frontend 43](#_Toc47352174)

[Chương 5. ĐÁNH GIÁ KẾT QUẢ 46](#_Toc47352175)

[Chương 6. KẾT LUẬN 47](#_Toc47352176)

[6.1. Ưu điểm: 47](#_Toc47352177)

[6.2. Nhược điểm 47](#_Toc47352178)

[Chương 7. HƯỚNG PHÁT TRIỂN 48](#_Toc47352179)

[TÀI LIỆU THAM KHẢO 49](#_Toc47352180)

[PHỤ LỤC 1: 51](#_Toc47352181)

# DANH MỤC HÌNH VẼ

[Hình 4‑1: Sơ đồ thực thể kết hợp ERD 41](#_Toc43270493)

[Hình 4‑2: Mô hình kiến trúc phía backend 46](#_Toc43270494)

[Hình 4‑3: Mô hình kiến trúc phía frontend 48](#_Toc43270495)

# DANH MỤC BẢNG

[Bảng 4‑1: Danh sách User Stories 25](#_Toc43270496)

[Bảng 4‑2: US-03: Visitor: Xem danh sách các sản phẩm 26](#_Toc43270497)

[Bảng 4‑3: US-05: Visitor: Xem chi tiết sản phẩm 26](#_Toc43270498)

[Bảng 4‑4: US-06: Visitor: Thêm sản phẩm vào giỏ hàng 27](#_Toc43270499)

[Bảng 4‑5: US-07: Visitor: Quản lý giỏ hàng hiện tại 28](#_Toc43270500)

[Bảng 4‑6: US-08: Visitor: Xác nhận đặt hàng 29](#_Toc43270501)

[Bảng 4‑7: US-09: Visitor: Đăng nhập & Admin: Đăng xuất 31](#_Toc43270502)

[Bảng 4‑8: US-14: Admin: Quản lý danh sách sản phẩm 32](#_Toc43270503)

[Bảng 4‑9: US-14-1: Admin: Tìm kiếm sản phẩm 33](#_Toc43270504)

[Bảng 4‑10: US-14-2: Admin: Xóa sản phẩm 34](#_Toc43270505)

[Bảng 4‑11: US-14-3: Admin: Thêm sản phẩm mới 34](#_Toc43270506)

[Bảng 4‑12: US-14-4: Admin: Cập nhật sản phẩm 36](#_Toc43270507)

[Bảng 4‑13: US-18: Admin: Quản lý danh sách đơn hàng 37](#_Toc43270508)

[Bảng 4‑14: US-18-1: Admin: Tìm kiếm đơn hàng 38](#_Toc43270509)

[Bảng 4‑15: US-18-2: Admin: Cập nhật trạng thái đơn hàng 39](#_Toc43270510)

[Bảng 4‑16: US-18-3: Admin: Xem chi tiết đơn hàng 39](#_Toc43270511)

[Bảng 4‑17: Danh sách các table trên MySQL 41](#_Toc43270512)

[Bảng 4‑18: Chi tiết category table trên MySQL 42](#_Toc43270513)

[Bảng 4‑19: Chi tiết product table trên MySQL 42](#_Toc43270514)

[Bảng 4‑20: Chi tiết account table trên MySQL 43](#_Toc43270515)

[Bảng 4‑21: Chi tiết administrative\_division table trên MySQL 43](#_Toc43270516)

[Bảng 4‑22: Chi tiết order\_status table trên MySQL 44](#_Toc43270517)

[Bảng 4‑23: Chi tiết cart table trên MySQL 44](#_Toc43270518)

[Bảng 4‑24: Chi tiết cart\_detail table trên MySQL 44](#_Toc43270519)

[Bảng 4‑25: Chi tiết order table trên MySQL 45](#_Toc43270520)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Từ viết đầy đủ** | **Diễn giải** |
| DBMS | Database Management System | Hệ quản trị cơ sở dữ liệu |
| RDBMS | Relational Database Management System | Hệ quản trị cơ sở dữ liệu kiểu quan hệ |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# MỞ ĐẦU

## Giới thiệu đề tài

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, đặc biệt là sự phát triển nhanh chóng của lĩnh vực công nghệ thông tin, công nghệ thông tin ngày càng đi vào đời sống và được con người khai thác một cách rất hiệu quả biến nó thành công cụ lao động hữu ích và đóng vai trò rất quan trọng trong đời sống xã hội. công cụ lao động. Và đặt biệt là sự phát triển cùa internet, nó ngày càng phổ biến vì sự tiện dụng, tính linh động của nó mang lại khả năng kết nối mọi lúc mọi nơi.

Đối với những cửa hàng nhỏ nho do cá nhân kinh doanh, đôi khi họ muốn cửa hàng mình được nhiều người biết tới. Có thể, xuất phát từ việc họ muốn nhiều người biết tới để họ có thể bán được nhiều hơn. Hay chỉ đơn thuần là họ muốn cửa hàng yêu quý của mình được biết đến. Cá nhân hay những nhóm nhỏ quản lý một cửa hàng cho riêng mình, họ xuất phát không đơn thuần là vì lợi nhuận mà nó còn là cả sở thích, niềm đam mê. Và caffee nói riêng hay những sản phẩm liên quan nói chung có một giá trị đặc biệt riêng đối với những người thích chúng.

Chúng ta cứ nghĩ làm chủ một cửa hàng nhỏ cũng không có gì khó khăn. Nhưng thực tế có rất nhiều vấn đề xoay quanh nó. Từ việc quản lý các sản phẩm của cửa hàng, cách mà cửa hàng hoạt động như mua hàng, bán hàng,... Đặc biệt trong thời buổi thương mại điện tử và giao hàng phát triển, nhiều người không thích ra đường mua. Vì thế cũng cần những phương pháp mua bán “hợp thời” hơn.

Ứng dụng web/website cho cửa hàng sẽ là giải pháp thích hợp để giải quyết toàn phần/một phần các vấn đề nêu trên. Website sẽ là công cụ hữu dụng nhằm quảng bá thương hiệu/cửa hàng, nó được xem như hình ảnh đại diện cho cửa hàng. Ngoài ra, những chức năng quản lý và hỗ trợ quản lý mà nó đem lại cũng giúp ích không nhỏ cho chủ cửa hàng.

Qua quá trình tìm hiểu, nghiên cứu và thực nghiệm, cá nhân em sẽ trau dồi thêm kiến thức, kỹ năng phát triển ứng dụng web nói chung và từng công nghệ được áp dựng trong đồ án nói riêng.

## Lý do chọn đề tài

Sự phát triển của thương mại điện tử ngày càng tăng lên với tốc độ cao. Đặt biệt là trong tình hình dịch bên COVID19 xảy ra trên toàn cầu. Nhu cầu đi lại giảm sút, khiến cho nhiều người ra đường giảm mạnh. Nắm bắt được điều đó, em quyết định chọn lựa đề tài “Xây dựng ứng dụng bán hàng” trên nền tảng web. Đề tài là sự kết hợp giữa thời sự và sở thích cá nhân em.

Và để giúp đỡ cho việc phát triển lĩnh vực này với kiến thức, hiểu biết về công nghệ mà em đã học và tìm hiểu được. Em quyết định chọn đề tài “Xây dựng ứng dụng bán hàng với ReactJs và ASP.NET Core” nhằm áp dụng toàn bộ kỹ năng, kiến thức mình có để đem ra được một sản phẩm hoàn chỉnh.

Với mong muốn sẽ làm chủ một tiệm cafe nhỏ cho riêng mình. Bản thân em chọn lựa đề tài và sẽ cố gắng hết sức để hoàn thiện sản phẩm.

# GIỚI THIỆU TỔNG QUAN

## Mục tiêu

Để xây dựng một hệ thống như mong muốn em cần đạt được các mục tiêu sau:

* Xây dựng một hệ thống website cửa hàng cafe ổn định
* Tìm hiểu nghiệp vụ bán hàng online ở mức cơ bản để hiện thực các chức năng
* Tìm hiểu về quy trình nghiệp vụ thanh toán, mua bán sản phẩm thức uống nói chung và cafe nói riêng
* Tìm hiểu cách hiện thực web api bằng công nghệ ASP.NET Core
* Tìm hiểu cách hiện thực web front-end bằng công nghệ ReactJS
* Nâng cao kĩ năng chuyên môn về phát triển ứng dụng web (front-end và back-end)

## Phạm vi

Như vậy, để đạt được mục tiêu nghiên cứu, em cần xác định rõ phạm vi của đề tài. Đề tài của em là xây dựng trang web bán hàng (cafe) bằng ReactJS và ASP.NET Core, cho nên sẽ có các tính năng cơ bản như sau:

* Quản lý các sản phẩm của cửa hàng trên hệ thống
* Cung cấp các tính năng mua sắm và đặt hàng
* Quản lý các đơn hàng để hỗ trợ mua bán hàng hóa
* Và một vài tính năng khác về phân quyền, xác thực người dùng

## Phương pháp

* Quy trình:
  + Phân tích đề tài
  + Liên hệ những người có kiến thức về bán hàng / bán hàng online
  + Tham khảo các trang web/hệ thống tương tự hiện có ở Việt Nam
  + Thu thập các yêu cầu cho hệ thống
  + Nghiên cứu công nghệ để áp dụng
* Môi trường phát triển:
  + Backend:
    - Web API với ASP.NET Core 3.1
    - Database với MySQL 8.0
  + Frontend:
    - NodeJS 12.16.3
    - ReactJs framework
    - Redux
    - Các thư viện JavaScript với Yarn package manager

## Ý nghĩa thực tiễn

* Cung cấp website quảng bá, đại diện cho cửa hàng
* Cung cấp sự nhanh chóng, tiện lợi cho người sử dụng
* Giúp tiết kiệm thời gian trong việc tìm kiếm và mua sản phẩm thức uống
* Áp dụng công nghệ thông tin vào lĩnh vực quảng cáo và mua bán

## Kết quả dự kiến

* Hiểu về nghiệp vụ mua bán sản phẩm online
* Tiếp cận và sử dụng được các công nghệ phổ biến hiện nay

# CÔNG NGHỆ SỬ DỤNG

## ASP.NET Core

Trước khi nói về ASP.NET Core, chúng ta hãy cùng bàn luận về .NET Core trước. Bởi vì .NET Core được xem là platform còn ASP.NET Core là framework.

.NET Có một lịch sử rất lâu đời, nhưng .NET Core thì lại còn rất non trẻ. .NET Core lần đầu xuất hiện với phiên bản 1.0 vào ngày 27-06-2016 [1]. .NET Core là mã nguồn mở, nền tải phát triển vì mục đích chung. Chúng ta có thể tạo các ứng dụng .NET Core cho Windows, macOS, Linux cho các proccessor kiến trúc x64, x86, ARM32 và ARM 64 bằng nhiều ngôn ngữ lập trình khác nhau [2]. .NET Core hỗ trợ 4 loại hình đa nền tảng gồm: ASP.NET Core web apps; command-line apps; các thư viện libraries; Universal Windows Platform apps [3].

ASP.NET là một framework để phát triển web rất phổ biến, nó dùng để xây dựng các ứng dụng web trên nền tảng .NET. ASP.NET Core là phiên bản mã nguồn mở của ASP.NET, nó có thể chạy trên macOS, Linux và Windows [4].

[5]platform là mới nhất tại thời điểm thực hiện đồ án này.

### Tại sao chọn ASP.NET Core [5]

ASP.NET Core hỗ trợ đa nền tảng, hiệu năng cao, và là framework mã nguồn mở phục vụ cho việc xây dựng ứng dụng hiện đại, cho phép kết nối cloud và internet. Với ASP.NET Core, chúng ta có thể:

* Xây dựng ứng dụng web và các dịch vụ, ứng dụng IoT, backend cho mobile app
* Có thể sử dụng chung với các công cụ khác trên Windows, macOS, Linux
* Triển khai trên cloud hoặc tự triển khai trên server riêng
* Chạy trên platform .NET Core

ASP.NET Core cung cấp một số lợi ích sau:

* Thống nhất cách để xây dựng Web UI và Web APIs
* Có thể phát triển và chạy trên Windows, macOS, Linux
* Mã nguồn mã và được cộng đồng quan tâm
* Tích hợp các framework hiện đại phía client như Angular, React, Redux,...
* Xây dựng sẵn pattern cho Dependency Injection
* Có thể host trên nhiều ứng dụng làm web server như:
  + Kestrel
  + IIS
  + Nginx
  + Apache
  + Docker

Ngoài những lý do về tính năng của ASP.NET trên thì lý do cá nhân mà bản thân em chọn lựa đó là:

* Có sẵn nền tảng kiến thức về ngôn ngữ lập trình C# và .NET của Microsoft
* Môi trường phát triển hiện đang sử dụng là Windows và đã quen thuộc với các công cụ phát triển đến tự Microsoft. Cho nên việc chọn lựa .NET cũng góp phần tăng thêm hiệu quả
* Thực hiện việc tự triển khai trên server riêng chạy Linux OS cần đến .NET để có thể chạy trên Linux và host trên Apache đã cài đặt ở server.

### ASP.NET Core web APIs

Trong phạm vi xây dựng ứng dụng web của đồ án này, em chọn lựa cách thức dùng ASP.NET Core web APIs để xây dựng phần backend cho hệ thống.

#### API và Web API [6]

API là các phương thức/giao thức kết nối một ứng dụng này với các thư viện/ứng dụng khác. Nó là viết tắt của từ tiếng anh Application Programming Interface. API cung cấp khả năng truy xuất đến một hoặc một tài các hàm mà ứng dụng đó muốn cung cấp API cho ứng dụng khác có thể sử dụng. Nhờ đó các ứng dụng có thể trao đổi dữ liệu với nhau.

Web API là một phương pháp trao đổi dữ liệu như API nhưng nó được thông qua giao thức HTTP hoặc HTTPS của mạng máy tính. Dữ liệu được API trả về thương là JSON hoặc XML, còn dữ liệu nhận thì có nhiều hình thức tùy thuộc vào phương thức HTTP của API đó.

Web API hoạt động như sau:

1. Xây dựng URL API hay còn được gọi là Endpoint bên các đó xác định phương thức HTTP của API (GET/POST/PUT/DELETE/...)
2. Các ứng dụng khác (bên thứ ba) có thể gửi request đến server cung cấp nội dung (API) thông qua giao thức HTTP/HTTPS
3. Tại web server cung cấp nội dung, thực hiện kiểm tra và xử lý sau đó trả về response tương ứng thông qua giao thức HTTP/HTTPS
4. Tại nơi yêu cầu ban đầu, sau khi nhận được dữ liệu sẽ tiến hành xử lý theo nhu cầu

#### Những tính năng của Web API [7]

ASP.NET Core web APIs khiến cho mọi thứ dễ dàng hơn khi mà chúng ta muốn xây dựng một service có khả năng cung cấp cho nhiều client như là các thiết bị mobile và browser. Với ASP.NET Core MVC chúng ta có thể sử dụng cùng framework, pattern để xây dựng cả 2 loại hình là web page và web APIs trên cùng một project.

Một số tính năng mà chúng ta sẽ sử dụng để xây dựng Web APIs:

* Serialization: ASP.NET được thiết kế cho việc trải nghiệm ứng dụng web theo kiểu hiện đại. Các endpoint sẽ tự động serialize các class thành định dạng JSON.
* Authentication & Authorization: xác thực và phân quyền là cách mà chúng ta có thể bảo vệ API endpoint. ASP.NET có tích hợp sẵn một số phương pháp xác thực như JWT, Role based, Policy based
* Routing alongside your code: ASP.NET cho phép chúng ta xác định route và các hành động (HTTP method) trực tiếp trong code bằng cách sử dụng các attribute. Quy định dữ liệu được đặt ở đâu: request path, query string, request body, form data từ đó tự động chuyển đổi thành các parameter.

## Hệ quản trị cơ sở dữ liệu MySQL/MariaDb

MySQL là một *hệ quản trị cơ sở dữ liệu kiểu quan hệ* gọi tắt là RDBMS – Relational Database Management System hoạt động theo mô hình Client-Server. MySQL quản lý dữ liệu thông qua các database, mỗi database có nhiều relational table chứa dữ liệu. MySQL có cách truy vấn dữ liệu thông qua ngôn ngữ SQL. MySQL được phát hành từ những năm 90s và có lịch sử lâu đời [8].

MySQL là open-source, nó cho phép bất kì ai đều có thể sử dụng và chỉnh sửa phần mềm. Bất kì ai đều có thể tải và sử dụng nó mà không cần trả phí. Nếu như muốn, có thể học cách chỉnh sửa nó cho phù hợp với nhu cầu. Tuy nhiên, nó không có nghĩa là hoàn toàn tự do. Open source cũng có những license (giấy phép) quy định những điều bạn có thể và không thể làm với phần mềm Open Source [9].

### Vì sao chọn sử dụng MySQL [9]

Mặc dù được học và làm quen về cơ sở dữ liệu bằng Microsoft SQL Server, tuy nhiên MySQL có một số lý do khách quan và chủ quan khiến em chọn lựa sử dụng MySQL làm hệ quản trị CSDL cho hệ thống như sau:

* Linh hoạt và dễ dùng: quá trình cài đặt MySQL rất đơn giản dễ hiểu và thường không quá 15 phút. Trong khi SQL Server lại quá nặng nề.
* Hiệu năng cao: MySQL có thể đáp ứng được các nhu cầu từ cơ bản đến nâng cao với tốc độ cao, hiệu quả. Và có thể chạy được trên các thiết bị có cấu hình yếu.
* Tiêu chuẩn trong ngành: MySQL cũng được xem là một trong những DBMS tiêu chuẩn của ngành công nghệ thông tin
* An toàn: An toàn dữ liệu luôn được lựa chọn đối với các phần mềm DBMS. Với hệ thống phân quyền truy cập và quản lý tài khoản, MySQL đặt tiêu chuẩn bảo mật rất cao. Mã hóa thông tin đăng nhập và chứng thực từ host đều khả dụng.

### MariaDB

MariaDB là RDBMS được phát triển từ MySQL. Nguyên do là khi MySQL bị mua lại bởi Oracle, Michael “Monty” Widenius – developer hàng đầu của MySQL lo sợ MySQL sẽ bị thương mại hóa không còn giữ được tính miễn phí của nó. Nên ông quyết định phát triển MariaDB nhằm thay thế MySQL và nó hoàn toàn miễn phí. Mặc dù vẫn có phiên bản trả phí của MariaDB, tuy nhiên phiên bản miễn phí của MariaDB được xem là vượt trội hơn so với MySQL miễn phí.

Không phải ngẫu nhiên mà MariaDB được nhiều người yêu thích hơn sơ với MySQL. Những ưu điểm lớn nhất của DBMS này phải kể đến bao gồm:

* Hoàn toàn miễn phí: Đây là một hệ quản trị sử dụng mã nguồn mở hoàn toàn miễn phí. Do đó, người dùng không cần phải bỏ tiền mua bản quyền và vẫn có thể sử dụng đầy đủ những tính năng của phần mềm này.
* Khắc phục hạn chế MySQL: những hạn chế của MySQL, khi chuyển qua MariaDB đều sẽ được khắc phục một cách triệt để nhất, thậm chí có thể tăng tốc độ hơn so với MySQL từ 3-5%. Ngoài ra, hệ quản trị này còn cải thiện hiệu năng và có thêm nhiều chức năng mới hơn so với MySQL.
* Tương thích với MySQL: chúng ta hoàn toàn có thể sử dụng những kiến thức đối với việc sử dụng MySQL mà áp dụng cho MariaDB. Ví dụ, phát triển đoạn code giao tiếp với MySQL thì hoàn toàn có thể sử dụng nói trên MariaDB

### Sự lựa chọn của cá nhân

Ban đầu vì bản thân chưa biết đến sự tồn tại của MariaDB và đã chọn lựa MySQL làm DBMS cho môi trường phát triển. Sau này, khi tiến hành cài đặt MySQL thì được biết MariaDB là DBMS mặc định trên các hệ điều hành Linux.

Nhận thấy được cả 2 đều có khả năng tương thích với nhau nên quyết định giữ lại MySQL trên môi trường phát triển vì MySQL Workbench khá là tiện lợi cũng như đã sử dụng quen thuộc. Trong khi trên server sẽ cài đặt MariaDB chỉ nhằm mục đích cung cấp một Database server chung để triển khai và lấy dự liệu cho quá trình phát triển.

Phiên bản được cài đặt trên môi trường phát triển là MySQL 8.0 Community Server trên hệ điều hành Windows 10. Và phiên bản được cài đặt trên môi trường triển khai là MariaDB 10.3 trên hệ điều hành Raspbian 10.

## ReactJs framework [10] [11]

React hay React.js hoặc ReactJS là một thư viện JavaScript mã nguồn mở cho việc xây dựng giao diện người dùng (UI). Nó được bảo trì bởi Facebook và cộng đồng các developer/công ty độc lập. Tuy nhiên, React chỉ quan tâm đến việc thực hiện render dữ liệu lên DOM. Vì thế để tạo ứng dụng React thường cần sử dụng thêm các thư viện bổ sung như Redux và React Router để quản lý state và định tuyến và nhiều thư viện khác nữa.

Một số tính năng đáng chú ý của React:

1. Virtual DOM

Công nghệ Virtual DOM giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này.Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

1. JSX – JavaScrip XML

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

1. Components

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi. Mỗi component có khả năng render thành một element cụ thể trên DOM bằng cách sử dụng thư viện DOM của React. Chúng ta có thể truyền dữ liệu cho component bằng “props”.

* Functional component: là cách khai báo một component bằng việc sử dụng nó như một hàm và phải trả về JSX
* Class-based component: sử dụng việc khai báo class theo chuẩn ES6. Chúng được biết như là các “statefull” component, bởi vì state của class sẽ giữ các giá trị và truyền xuống cho các component con thông qua props
* Props: là viết tắt cho từ properties và chúng được dùng để truyền dữ liệu vào bên trong component. Nó gồm dữ liệu người dùng tự định nghĩa và các thuộc tính liên quan đến việc render của component. Props là không đổi.
* State: đại diện cho trạng thái của component. Khi state thay đổi component sẽ thực hiện việc render lại và nhờ Virtual DOM cập nhật lên UI.

## Apache2 web server [12]

### Web server

Nhiệm vụ của web server là đưa website lên internet. Để làm được điều đó, nó hoạt động giống như là một người đứng giữa server và client. Nó sẽ kéo nội dung từ server về cho mỗi một request xuất phát từ client để hiển thị kết quả tương ứng dưới hình thức là một website.

Web server xử lý các file lưu trên server vật lý dưới ngôn ngữ lập trình như là PHP, Python, Java, vâng vâng. Những ngôn ngữ này biến chúng thành file HTML và file trên trình duyệt cho người dùng web thấy được. Khi bạn nghe tới cụm từ “web server”, hãy hiểu rằng nó là công cụ chịu trách nhiệm giao tiếp giữa server-client.

### Apache

Apache là phần mềm web server miễn phí mã nguồn mở. Nó đang chiếm đến khoảng 46% thị phần websites trên toàn thế giới. Tên chính thức của Apache là Apache HTTP Server, được điều hành và phát triển bởi Apache Software Foundation.

Mặc dù chúng ta gọi Apache là web server, nhưng nó lại không phải là server vật lý, nó là một phần mềm chạy trên server đó. Công việc của nó là thiết lập kết nối giữa server và trình duyệt người dùng (Firefox, Google Chrome, Safari, vâng vâng.) rồi chuyển file tới và lui giữa gchúng (cấu trúc 2 chiều dạng client-server). Apache là một phần mềm đa nền tảng, nó hoạt động tốt với cả server Linux và Windows.

Apache web server là lựa chọn ưu việc để vận hành một website ổn định và có thể tùy chỉnh linh hoạt. Tuy nhiên, nó cũng có một số điểm bất lợi mà bạn nên biết:

* Ưu điểm:
  + Phần mềm mã nguồn mở và miễn phí, kể cả cho mục đích thương mại.
  + Phần mềm đáng tin cậy, ổn định.
  + Được cập nhật thường xuyên, nhiều bản vá lỗi bảo mật liên tục.
  + Linh hoạt vì có cấu trúc module.
  + Dễ cấu hình, thân thiện với người mới bắt đầu
  + Đa nền tảng (hoạt động được cả với server Unix và Windows).
  + Hoạt động cực kỳ hiệu quả với WordPress sites.
  + Có cộng đồng lớn và sẵn sàng hỗ trợ với bất kỳ vấn đề nào.
  + Virtual Host cho phép nhiều domain trên một server, và tùy chọn khác
* Nhược điểm:
  + Gặp vấn đề hiệu năng nếu website có lượng truy cập cực lớn.
  + Quá nhiều lựa chọn thiết lập có thể gây ra các điểm yếu bảo mật

### Một số vấn đề khác

Vì ở đây chúng ta thực hiện cả công việc triển khai một ứng dụng lên server và có thể kết nối từ phía ngoài mạng localhost. Cho nên chúng ta cần một số kiến thức liên quan đến mạng máy tính:

* Làm chủ được internet network của server: chúng ta cần thực hiện một số cấu hình trên mạng mà server kết nối đến, cho nên chúng ta cần có quyền thực hiện những điều đó. Nói đơn giản thì chúng ta cần kết nối vào modem.
* Static IP: cần phải cố định địa chỉ IP của thiết bị làm server để đảm bảo rằng các kết nối đến server được chính xác.
* Port forwarding: Modem chúng ta được kết nối với Internet bên ngoài sẽ có một địa chỉ gọi là Public IP. Internet bên ngoài không biết gì về kết nối bên trong của Modem chúng ta mà chỉ biết duy nhất Public IP của Modem. Vậy để bên ngoài có thể vào đúng server của chúng ta thì Port forwarding, nó sẽ cho phép chúng ta có thể chuyển tiếp từ port của Public sang port của một địa chỉ cụ thể trong mạng nội bộ

# XÂY DỰNG ỨNG DỤNG

### User stories

### Danh sách User stories

Bảng ‑: Danh sách User Stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mã US** | **Tiêu đề** | **Miêu tả** | **Độ ưu tiên** |
|  | US-03 | Visitor: Xem danh sách các sản phẩm | Là một Visitor, tôi muốn trang sản phẩm phải có khả năng xem toàn bộ sản phẩm hoặc theo một bộ lọc nào đó | 1\_Highest |
|  | US-06 | Visitor: Thêm sản phẩm vào giỏ hàng | Là một Visitor, tôi muốn chọn một sản phẩm để thêm nó vào trong giỏ hàng hiện tại của tôi | 1\_Highest |
|  | US-07 | Visitor: Quản lý giỏ hàng hiện tại | Là một Visitor, tôi muốn thực hiện một số thao tác quản lý trên giỏ hàng của mình. Vì tôi muốn kiểm tra cũng như là thay đổi thông tin trong giỏ hàng của mình. | 1\_Highest |
|  | US-08 | Visitor: Xác nhận đặt hàng | Là một Visitor, sau khi kiểm tra giỏ hàng, tôi muốn xác nhận mua các sản phẩm trong giỏ với cửa hàng từ đó tạo thành đơn hàng | 1\_Highest |
|  | US-09 | Visitor: Đăng nhập & Admin: Đăng xuất | Là một Visitor, tôi muốn đăng nhập vào tài khoản để xác thực và phân quyền đối với hệ thống. Là một Admin, sau khi đăng nhập tôi có thể thực hiện chức năng đăng xuất để xác thực lại và chuyển đổi phân quyền | 1\_Highest |
|  | US-14-1 | Admin: Tìm kiếm sản phẩm | Là một Admin, tôi muốn tìm kiếm các sản phẩm theo bộ lọc dành cho người quản trị. Để tôi có thể thực hiện tiếp tục các chức năng quản lý | 1\_Highest |
|  | US-18-1 | Admin: Tìm kiếm đơn hàng | Là một Admin, tôi muốn tìm kiếm các đơn hàng theo bộ lọc dành cho người quản trị. Để tôi có thể thực hiện tiếp tục các chức năng quản lý | 1\_Highest |
|  | US-05 | Visitor: Xem chi tiết sản phẩm | Là một Visitor, tôi một page trình bày chi tiết thông tin về sản phẩm đó | 2\_High |
|  | US-14-2 | Admin: Xóa sản phẩm | Là một Admin, tôi muốn xóa sản phẩm khỏi cửa hàng. Vì tôi không còn cần sản phẩm đó trong cửa hàng | 2\_High |
|  | US-14-3 | Admin: Thêm sản phẩm mới | Là một Admin, tôi muốn tạo sản phẩm mới cho cửa hàng. Vì tôi muốn cửa hàng của mình có thêm nhiều sản phẩm. | 2\_High |
|  | US-14-4 | Admin: Cập nhật sản phẩm | Là một Admin, tôi muốn thay đổi và cập nhật thay đổi của thông tin sản phẩm mới cho cửa hàng. | 2\_High |
|  | US-18-2 | Admin: Cập nhật trạng thái đơn hàng | Là một Admin, tôi không thể thay đổi hoàn toàn đơn hàng. Nhưng tôi cần thay đổi trạng thái của đơn hàng. | 2\_High |
|  | US-18-3 | Admin: Xem chi tiết đơn hàng | Là một Admin, tôi có thể xem chi tiết các sản phẩm trong đơn hàng để chuẩn bị cho khách. | 2\_High |

### Mô tả chi tiết các User stories

#### US-03: Visitor: Xem danh sách các sản phẩm

Bảng ‑: US-03: Visitor: Xem danh sách các sản phẩm

|  |  |
| --- | --- |
| **Title** | Visitor: Xem danh sách các sản phẩm |
| **Value Statement** | Là một Visitor, tôi muốn page sản phẩm phải có khả năng xem toàn bộ hoặc một phần danh sách sản phẩm |
| **Acceptance Criteria** | Thông tin của SẢN PHẨM bao gồm: Tiêu đề (tên); Giá bán; Mô tả; Phân loại; Hình ảnh;  **Scenario:** Visitor xem danh sách sản phẩm thành công  Given: Tôi có thể chọn xem danh sách sản phẩm  When: Tôi nhấp chọn xem danh sách sản phẩm  Then: Tôi thấy được các sản phẩm trong page sản phẩm  Có một bộ lọc đơn giản cho danh sách sản phẩm dựa trên phân loại sản phẩm |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-05: Visitor: Xem chi tiết sản phẩm

Bảng ‑: US-05: Visitor: Xem chi tiết sản phẩm

|  |  |
| --- | --- |
| **Title** | Visitor: Xem chi tiết sản phẩm |
| **Value Statement** | Là một Visitor, tôi một page trình bày chi tiết thông tin về sản phẩm đó |
| **Acceptance Criteria** | Thông tin của SẢN PHẨM bao gồm: Mã SP; Tiêu đề (tên); Giá bán; Mô tả; Phân loại; Hình ảnh;  **Scenario:** Visitor xem chi tiết sản phẩm thành công  Given: Tôi có thể chọn xem chi tiết sản phẩm  When: Tôi nhấp chọn xem chi tiết sản phẩm  Then: Tôi thấy được các thông tin chi tiết sản phẩm |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-06: Visitor: Thêm sản phẩm vào giỏ hàng

Bảng ‑: US-06: Visitor: Thêm sản phẩm vào giỏ hàng

|  |  |
| --- | --- |
| **Title** | Visitor: Thêm sản phẩm vào giỏ hàng |
| **Value Statement** | Là một Visitor, tôi muốn chọn một sản phẩm để thêm nó vào trong giỏ hàng hiện tại của tôi vì tôi muốn mua nó. |
| **Acceptance Criteria** | **Scenario:** Visitor thêm sản phẩm vào giỏ hàng thành công  Given: Tôi đang trong page danh sách sản phẩm  When: Tôi nhấp chọn mua sản phẩm  Then: Sản phẩm được chọn thêm vào trong giỏ hàng của tôi |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-07: Visitor: Quản lý giỏ hàng hiện tại

Bảng ‑: US-07: Visitor: Quản lý giỏ hàng hiện tại

|  |  |
| --- | --- |
| **Title** | Visitor: Quản lý giỏ hàng hiện tại |
| **Value Statement** | Là một Visitor, tôi muốn thực hiện một số thao tác quản lý trên giỏ hàng của mình. Vì tôi muốn kiểm tra cũng như là thay đổi thông tin trong giỏ hàng của mình. |
| **Acceptance Criteria** | Thông tin của CHI TIẾT GIỎ HÀNG: Hình ảnh; Tiêu đề (tên); Giá bán hiện tại; Số lượng; Thành tiền;  Thông tin của GIỎ HÀNG: Danh sách CT Giỏ hàng; Giá tạm tính; Phí vận chuyển; Giảm giá; Tổng cộng;  **Scenario**: Visitor xem thông tin giỏ hàng  Given: Tôi có thể chọn xem giỏ hàng của mình  When: Tôi nhấp chọn xem giỏ hàng của mình  Then: Tôi có thể thấy được thông tin giỏ hàng của mình  **Scenario:** Visitor xóa sản phẩm trong giỏ hàng  Given: Tôi đang trong page xem giỏ hàng của mình  When: Tôi nhấp chọn xóa một sản phẩm trong giỏ hàng  Then: Sản phẩm được chọn sẽ bị xóa khỏi giỏ hàng của tôi  **Scenario:** Visitor thay đổi số lượng sản phẩm trong giỏ hàng  Given: Tôi đang trong page xem giỏ hàng của mình  When: Tôi thay đổi số lượng của sản phẩm được chọn có trong giỏ hàng của tôi  Then: Số lượng của sản phẩm được chọn cập nhật thành giá trị tôi muốn thay đổi |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-08: Visitor: Xác nhận đặt hàng

Bảng ‑: US-08: Visitor: Xác nhận đặt hàng

|  |  |
| --- | --- |
| **Title** | Visitor: Xác nhận đặt hàng |
| **Value Statement** | Là một Visitor, sau khi kiểm tra giỏ hàng, tôi muốn xác nhận mua các sản phẩm trong giỏ với cửa hàng từ đó tạo thành đơn hàng |
| **Acceptance Criteria** | Thông tin của ĐƠN HÀNG bao gồm: Mã đơn; Firstname khách đặt hàng; Lastname khách đặt hàng; Tỉnh thành; Quận/Huyện/Thị xã; Phường/Xã; Số địa chỉ; Ghi chú; Số điện thoại; Giỏ hàng;  **Scenario**: Visitor cập nhật thông tin giao hàng  Given: Tôi đang trong page xem giỏ hàng của mình  And: Giỏ hàng của tôi có sản phẩm trong đó  When: Tôi nhấp chọn đặt hàng  And: Tôi đã nhập thông tin giao hàng  And: Tôi chọn xác nhận  Then: Thông tin giao hàng được cập nhật vào trong đơn hàng  **Scenario:** Visitor xác nhận đặt hàng  Given: Tôi đã cập nhật và kiểm tra thông tin giao hàng  When: Tôi nhấp chọn đặt hàng  Then: Đơn hàng của tôi được thiết lập trên hệ thống  And: Tôi có thể thấy đơn hàng trên màn hình kết quả  **Scenario:** Visitor hủy thao tác đặt hàng  Given: Tôi đang trong thao tác đặt hàng  And: Trước khi đơn hàng được xác nhận đã đặt  When: Tôi nhấp chọn hủy  Then: Thông tin giao hàng đã nhập được cập nhật  And: Tôi quay về page giỏ hàng của mình |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-09: Visitor: Đăng nhập & Admin: Đăng xuất

Bảng ‑: US-09: Visitor: Đăng nhập & Admin: Đăng xuất

|  |  |
| --- | --- |
| **Title** | Visitor: Đăng nhập & Admin: Đăng xuất |
| **Value Statement** | Là một Visitor, tôi muốn đăng nhập vào tài khoản để xác thực và phân quyền đối với hệ thống. Là một Admin, sau khi đăng nhập tôi có thể thực hiện chức năng đăng xuất để xác thực lại và chuyển đổi phân quyền |
| **Acceptance Criteria** | Thông tin TÀI KHOẢN bao gồm: Username; Password;  **Scenario**: Visitor đăng nhập thành công  Given: Tôi chưa đăng nhập hoặc đã đăng xuất  And: Tôi đã nhập thông tin đăng nhập  When: Tôi nhấp chọn đăng nhập  And: Hệ thống xác thực tài khoản thành công  Then: Tôi có thể thấy mình đã đăng nhập vào hệ thống  **Scenario:** Visitor đăng nhập thất bại  Given: Tôi đã nhập thông tin đăng nhập  When: Tôi nhấp chọn đăng nhập  And: Hệ thống xác thực tài khoản không thành công  Then: Tôi có thể thấy thông báo lỗ  And: Tôi có thể thực hiện đăng nhập lại  **Scenario**: Admin đăng xuất thành công  Given: Tôi đã đăng nhập vào hệ thống  When: Tôi nhấp chọn đăng xuất  Then: Tôi có thể thấy mình chưa đăng nhập và hệ thống |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-14: Admin: Quản lý danh sách sản phẩm

Bảng ‑: US-14: Admin: Quản lý danh sách sản phẩm

|  |  |
| --- | --- |
| **Title** | Admin: Quản lý danh sách sản phẩm |
| **Value Statement** | Là một Admin, tôi muốn thực hiện một số thao tác quản lý trên danh sách các sản phẩm. Vì tôi muốn kiểm tra cũng như là thay đổi thông tin các sản phẩm của cửa hàng |
| **Acceptance Criteria** | Thông tin của SẢN PHẨM bao gồm: Mã SP; Tiêu đề (tên); Giá bán; Mô tả; Phân loại; Hình ảnh;  **Scenario:** Admin tìm kiếm sản phẩm  Xem US-14-1  **Scenario:** Admin xóa sản phẩm  Xem US-14-2  **Scenario:** Admin thêm sản phẩm mới  Xem US-14-3  **Scenario:** Admin cập nhật sản phẩm  Xem US-14-4 |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-14-1: Admin: Tìm kiếm sản phẩm

Bảng ‑: US-14-1: Admin: Tìm kiếm sản phẩm

|  |  |
| --- | --- |
| **Title** | Admin: Tìm kiếm sản phẩm |
| **Value Statement** | Là một Admin, tôi muốn tìm kiếm các sản phẩm theo bộ lọc dành cho người quản trị. Để tôi có thể thực hiện tiếp tục các chức năng quản lý |
| **Acceptance Criteria** | Thuộc tính của SẢN PHẨM cho phép tìm kiếm: Mã SP; Tiêu đề (tên); Giá bán; Phân loại;  **Scenario:** Admin tìm kiếm sản phẩm thành công  Given: Tôi đang trong page quản lý sản phẩm  And: Tôi đã nhập các thông tin tìm kiếm  When: Tôi nhấp chọn tìm kiếm  Then: Tôi có thể thấy được danh sách sản phẩm đã lọc |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-14-2: Admin: Xóa sản phẩm

Bảng ‑: US-14-2: Admin: Xóa sản phẩm

|  |  |
| --- | --- |
| **Title** | Admin: Xóa sản phẩm |
| **Value Statement** | Là một Admin, tôi muốn xóa sản phẩm khỏi cửa hàng. Vì tôi không còn cần sản phẩm đó trong cửa hàng |
| **Acceptance Criteria** | **Scenario:** Admin xóa sản phẩm thành công  Given: Tôi đang trong page xem danh sách sản phẩm  When: Tôi nhấp chọn xóa một sản phẩm trong danh sách  Then: Sản phẩm được chọn sẽ bị xóa khỏi danh sách |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-14-3: Admin: Thêm sản phẩm mới

Bảng ‑: US-14-3: Admin: Thêm sản phẩm mới

|  |  |
| --- | --- |
| **Title** | Admin: Thêm sản phẩm mới |
| **Value Statement** | Là một Admin, tôi muốn tạo sản phẩm mới cho cửa hàng. Vì tôi muốn cửa hàng của mình có thêm nhiều sản phẩm. |
| **Acceptance Criteria** | Thông tin của SẢN PHẨM bao gồm: Mã SP; Tiêu đề (tên); Giá bán; Mô tả; Phân loại; Hình ảnh;  **Scenario:** Admin nhập thông tin sản phẩm mới  Given: Tôi đang trong page quản lý sản phẩm  When: Tôi nhấp chọn thêm sản phẩm mới  Then: Tôi có thể thấy được màn hình nhập thông tin sản phẩm mới  **Scenario:** Admin thêm sản phẩm mới thành công  Given: Tôi đang trong màn hình nhập thông tin sản phẩm mới  And: Tôi đã nhập các thông tin của sản phẩm mới  When: Tôi nhấp chọn thêm sản phẩm mới  Then: Tôi có thể thấy sản phẩm mới đã được thêm  **Scenario**: Admin hủy thêm sản phẩm mới  Given: Tôi đang trong màn hình nhập thông tin sản phẩm mới  And: Trước khi sản phẩm được xác nhận thêm  When: Tôi nhấp chọn hủy  Then: Các thông tin đã nhập bị hủy bỏ  And: Tôi quay về page quản lý sản phẩm |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-14-4: Admin: Cập nhật sản phẩm

Bảng ‑: US-14-4: Admin: Cập nhật sản phẩm

|  |  |
| --- | --- |
| **Title** | Admin: Cập nhật sản phẩm |
| **Value Statement** | Là một Admin, tôi muốn thay đổi và cập nhật thay đổi của thông tin sản phẩm mới cho cửa hàng. |
| **Acceptance Criteria** | Thông tin của SẢN PHẨM bao gồm: Mã SP; Tiêu đề (tên); Giá bán; Mô tả; Phân loại; Hình ảnh;  **Scenario:** Admin nhập thông tin sản phẩm đã chọn  Given: Tôi đang trong page quản lý sản phẩm  When: Tôi nhấp chọn cập nhật một sản phẩm trong danh sách  Then: Tôi có thể thấy được màn hình nhập thông tin sản phẩm  **Scenario:** Admin cập nhật sản phẩm thành công  Given: Tôi đang trong màn hình nhập thông tin sản phẩm  And: Tôi đã nhập các thông tin mới của sản phẩm  When: Tôi nhấp chọn cập nhật sản phẩm  Then: Tôi có thể thấy sản phẩm đã được cập nhật thông tin  **Scenario**: Admin hủy cập nhật sản phẩm  Given: Tôi đang trong màn hình nhập thông tin sản phẩm  And: Trước khi sản phẩm được xác nhận cập nhật  When: Tôi nhấp chọn hủy  Then: Các thông tin đã nhập bị hủy bỏ  And: Tôi quay về page quản lý sản phẩm |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-18: Admin: Quản lý danh sách đơn hàng

Bảng ‑: US-18: Admin: Quản lý danh sách đơn hàng

|  |  |
| --- | --- |
| **Title** | Admin: Quản lý danh sách đơn hàng |
| **Value Statement** | Là một Admin, tôi muốn thực hiện một số thao tác quản lý trên danh sách các đơn hàng. Vì tôi muốn kiểm tra cũng như là thay đổi thông tin các sản phẩm của cửa hàng |
| **Acceptance Criteria** | Thông tin của ĐƠN HÀNG bao gồm: Mã đơn; Firstname khách đặt hàng; Lastname khách đặt hàng; Tỉnh thành; Quận/Huyện/Thị xã; Phường/Xã; Số địa chỉ; Ghi chú; Số điện thoại; Giỏ hàng;  **Scenario:** Admin tìm kiếm đơn hàng  Xem US-18-1  **Scenario:** Admin cập nhật trạng thái đơn hàng  Xem US-18-2 |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-18-1: Admin: Tìm kiếm đơn hàng

Bảng ‑: US-18-1: Admin: Tìm kiếm đơn hàng

|  |  |
| --- | --- |
| **Title** | Admin: Tìm kiếm đơn hàng |
| **Value Statement** | Là một Admin, tôi muốn tìm kiếm các đơn hàng theo bộ lọc dành cho người quản trị. Để tôi có thể thực hiện tiếp tục các chức năng quản lý |
| **Acceptance Criteria** | Thuộc tính của ĐƠN HÀNG cho phép tìm kiếm: Mã đơn; Họ & tên khách đặt hàng; Số điện thoại; Trạng thái; Tổng tiền;  **Scenario:** Admin tìm kiếm đơn hàng thành công  Given: Tôi đang trong page quản lý đơn hàng  And: Tôi đã nhập các thông tin tìm kiếm  When: Tôi nhấp chọn tìm kiếm  Then: Tôi có thể thấy được danh sách đơn hàng đã lọc |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-18-2: Admin: Cập nhật trạng thái đơn hàng

Bảng ‑: US-18-2: Admin: Cập nhật trạng thái đơn hàng

|  |  |
| --- | --- |
| **Title** | Admin: Cập nhật trạng thái đơn hàng |
| **Value Statement** | Là một Admin, tôi không thể thay đổi hoàn toàn đơn hàng. Nhưng tôi cần thay đổi trạng thái của đơn hàng. |
| **Acceptance Criteria** | Thông tin TRẠNG THÁI của ĐƠN HÀNG bao gồm: Đang xử lý; Đang chuẩn bị; Đang giao hàng; Đã giao hàng;  **Scenario:** Admin cập nhật trạng thái đơn hàng thành công  Given: Tôi đang trong page quản lý đơn hàng  When: Tôi nhấp chọn cập nhật trạng thái một đơn hàng trong danh sách đơn hàng  And: Tôi chọn trạng thái cần cập nhật  Then: Tôi có thể thấy được trạng thái của đơn hàng đã được đổi |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

#### US-18-3: Admin: Xem chi tiết đơn hàng

Bảng ‑: US-18-3: Admin: Xem chi tiết đơn hàng

|  |  |
| --- | --- |
| **Title** | Admin: Xem chi tiết đơn hàng |
| **Value Statement** | Là một Admin, tôi có thể xem chi tiết các sản phẩm trong đơn hàng để chuẩn bị cho khách. |
| **Acceptance Criteria** | Thông tin của ĐƠN HÀNG bao gồm: Mã đơn; Firstname khách đặt hàng; Lastname khách đặt hàng; Tỉnh thành; Quận/Huyện/Thị xã; Phường/Xã; Số địa chỉ; Ghi chú; Số điện thoại; Giỏ hàng;  **Scenario:** Admin xem chi tiết đơn hàng thành công  Given: Tôi đang trong page quản lý đơn hàng  When: Tôi nhấp chọn xem chi tiết một đơn hàng trong danh sách đơn hàng  Then: Tôi có thể thấy được chi tiết đơn hàng đã được chọn |
| **Definition of Done** | * Đạt được Acceptance Criteria * Implement thành code * Hoàn thành Unit test * Build được |

## Thiết kế cơ sở dữ liệu

### Sơ đồ thực thể kết hợp ERD

A screenshot of a cell phone

Description automatically generated

Hình ‑: Sơ đồ thực thể kết hợp ERD

### Danh sách các Table

Bảng ‑: Danh sách các table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trong ERD** | **Tên trong DB** | **Ghi chú/Ý nghĩa** |
|  | Category | category | Danh mục phân loại sản phẩm |
|  | Product | product | Danh mục sản phẩm |
|  | Account | account | Danh mục tài khoản người dùng |
|  | AdministrativeDivision | administrative\_division | Danh mục đơn vị hành chánh Việt Nam |
|  | OrderStatus | order\_status | Danh mục trạng thái đơn hàng |
|  | Cart | cart | Danh mục giỏ hàng |
|  | CartDetail | cart\_detail | Danh mục chi tiết giỏ hàng |
|  | Order | order | Danh mục đơn hàng |

### Chi tiết table trên MySQL

#### Category Table

Bảng ‑: Chi tiết category table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (AI, PK) | Khóa chính, tự động tăng |
|  | Title | VARCHAR(32) (NN, UQ) | Không null, duy nhất |

#### Product Table

Bảng ‑: Chi tiết product table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (AI, PK) | Khóa chính, tự động tăng |
|  | Code | VARCHAR(8) (NN, UQ) | Mã sản phẩm người dùng tự nhập  Không null, duy nhất. |
|  | Title | VARCHAR(32) (NN, UQ) | Không null, duy nhất |
|  | Description | VARCHAR(1024) (NN) | Không null, giá trị mặc định:  “No Description” |
|  | CategoryID | INT (NN, FK) | Khóa ngoại Category |
|  | Price | INT (NN) | Không null |
|  | ImageURL | VARCHAR(256) (NN) | Không null, giá trị mặc định:  “default.png” |
|  | RecordStatus | TINYINT (NN) | Không null. RecordStatus:   * 0: Đã xóa * 1: Không xóa |

#### Account Table

Bảng ‑: Chi tiết account table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (AI, PK) | Khóa chính, tự động tăng |
|  | Username | VARCHAR(32) (NN, UQ) | Không null, duy nhất |
|  | Password | BINARY(20) (NN) | Không null, lưu ở dạng SHA1 |

#### AdministrativeDivision Table

Bảng ‑: Chi tiết administrative\_division table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (AI, PK) | Khóa chính, tự động tăng |
|  | FatherId | INT | Khóa ngoại, đơn vị cha |
|  | Name | VARCHAR(32) (NN, UQ) | Không null, duy nhất |
|  | Type | VARCHAR(32) (NN) | Không null, loại đơn vị |
|  | Level | VARCHAR(32) (NN) | Không null, phân cấp đơn vị:   * “Tỉnh/Thành” * “Quận/Huyện” * “Phường/Xã” |

#### OrderStatus Table

Bảng ‑: Chi tiết order\_status table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (AI, PK) | Khóa chính, tự động tăng |
|  | Status | VARCHAR(32) (NN, UQ) | Không null, duy nhất |

#### Cart Table

Bảng ‑: Chi tiết cart table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | OrderID | INT (PK, FK) | Khóa chính, khóa ngoại Order |
|  | Subtotal | INT (NN) | Không null. Giá tạm tính |
|  | Delivery | INT (NN) | Không null. |
|  | Discount | INT (NN) | Không null. |
|  | Total | INT (NN) | Không null. |

#### CartDetail Table

Bảng ‑: Chi tiết cart\_detail table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | OrderID | INT (PK, FK) | Khóa chính, khóa ngoại Cart |
|  | ProductID | INT (PK, FK) | Khóa chính, khóa ngoại Product |
|  | Price | INT (NN) | Không null. Giá thời điểm mua |
|  | Quantity | INT (NN) | Không null. |
|  | Total | INT (NN) | Không null. |

#### Order Table

Bảng ‑: Chi tiết order table trên MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **KDL trong DB** | **Ghi chú/Ý nghĩa** |
|  | ID | INT (PK, AI) | Khóa chính, tự động tăng |
|  | Firstname | VARCHAR(32) (NN) | Không null |
|  | Lastname | VARCHAR(32) (NN) | Không null |
|  | StatusID | INT (NN, FK) | Khóa ngoại OrderStatus |
|  | Phone | VARCHAR(16) (NN) | Không null |
|  | ProvinceID | INT (NN) | Khóa ngoại AdministrativeDivision |
|  | DistrictID | INT (NN) | Khóa ngoại AdministrativeDivision |
|  | CommuneID | INT (NN) | Khóa ngoại AdministrativeDivision |
|  | Address | VARCHAR(256) (NN) | Không null. Số nhà và đường |
|  | Note | VARCHAR(256) (NN) | Không null. Ghi chú thêm |

## Thiết kế kiến trúc phần mềm

Hệ thống web site bán hàng của em bao gồm 2 phần tách biệt là frontend và backend. Trong đó frontend được xây dựng trên công nghệ ReactJS và backend được xây dựng bằng công nghệ ASP.NET Core.

### Kiến trúc phía backend

Phía backend được thiết kế theo mô hình Layer. Trong đó có 3 layer chính là Controler, Services và Repositories; bên cạnh có một module Helper. Chi tiết các thành phần như sau:

A screenshot of a social media post

Description automatically generated

Hình ‑: Mô hình kiến trúc phía backend

* **Controllers**: được dùng để cấu hình và xử lý các request, response giữa frontend và backend. Controller là nơi định nghĩa và xử lý các web api. Trong khi đó, Resources sẽ là nơi định nghĩa các loại data nằm trong request và response, cùng với một số phương thức chuyển đổi giữa resources và models.
* **Services**: được dùng để định nghĩa các class để thực hiện xử lý nghiệp vụ bao gồm cả repository giúp thao tác database. Service sẽ sử dụng interface để thực hiện thao tác database khi cần thiết. Các xử lý nghiệp vụ khác sẽ được thực hiện tại đây. Ví dụ như lưu file vào bộ nhớ. Mặc dù việc lưu file không phải là của service, nhưng nó sẽ là nơi để quyết định việc lưu file.
* **Repositories**: Trong Repositories sẽ có ProcedureHelper là nhiệm vụ gọi stored procedure để thao tác với database. ProcedureHelper sử dụng thư viện Dapper giúp cho việc thao tác với database thông qua store procedure và trả về DTO. Các DTO là các class được định nghĩa tương đồng với kết quả trả về từ procedure.
* **Helpers**: Nơi định nghĩa các class có nhiệm vụ hỗ trợ, xử lý, thực hiện các công việc nhỏ lẻ. Chẳng hạng như xử lý kết quả trả về, xử lý thao tác với tập tin...

### Kiến trúc phía frontend

Phía frontend được xây dựng dựa trên quy chuẩn của ReactJS. Trong đó có 4 thành phẩn gồm có: Components, Assets, Redux, Utils và chi tiết như sau:

A screenshot of a cell phone

Description automatically generated

Hình ‑: Mô hình kiến trúc phía frontend

* **Components/Layouts/Views**: thành phần này gồm nhiều thư mục trong source code. Nhưng chung quy thì vẫn là thực hiện nhiệm vụ định nghĩa các React component cho trang web
* **Assets**: Thư mục chứa mọi vấn đề liên quan đề styles, font, image
* **Redux**: Thư viện React Redux đã quá quen thuộc cũng như là cần thiết cho việc quản lý state trên React. Và đây là nơi triển khai source code cho Redux
* **Utils**: Giống với Helper ở backend, đây là module dùng để chứa những hàm hỗ trợ xử lý cho phía frontend. Nó có thể là việc gọi API, xử lý lỗi,...

# ĐÁNH GIÁ KẾT QUẢ

# KẾT LUẬN

## Ưu điểm:

- Trang web đáp ứng được nhu cầu cơ bản cho việc đặt hàng, Giao diện đẹp, rõ ràng, dễ dàng sử dụng.

- Tốc độ phản hồi nhanh.

## Nhược điểm

- Độ bảo mật chưa cao.

- Chưa tích hợp chức năng giao hàng

# HƯỚNG PHÁT TRIỂN

* Mở rộng và hoàn thiện trang web.
* Tăng tính bảo mật người dùng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | R. Lander, "Announcing .NET Core 1.0," Microsoft Blog, 27 June 2016. [Online]. Available: https://devblogs.microsoft.com/dotnet/announcing-net-core-1-0/. [Accessed 24 May 2020]. |
| [2] | R. Lander, ".NET Core intro and overview," Microsoft Docs, 26 March 2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/core/introduction. [Accessed 24 May 2020]. |
| [3] | ".NET Core," Wikipedia, 23 May 2020. [Online]. Available: https://en.wikipedia.org/wiki/.NET\_Core. [Accessed 24 May 2020]. |
| [4] | "What is ASP.NET Core," Microsoft .NET, [Online]. Available: https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core. [Accessed 24 May 2020]. |
| [5] | R. Daniel, A. Rick and L. Shaun, "Introduction to ASP.NET Core," Microsoft Docs, 17 April 2020. [Online]. Available: https://docs.microsoft.com/vi-vn/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-3.1. [Accessed 24 May 2020]. |
| [6] | "API là gì? Những đặc điểm nổi bật của API," TopDev, [Online]. Available: https://topdev.vn/blog/api-la-gi/. [Accessed 24 May 2020]. |
| [7] | "ASP.NET Web APIs," Microsoft .NET, [Online]. Available: https://dotnet.microsoft.com/apps/aspnet/apis. [Accessed 24 May 2020]. |
| [8] | Đ. Đạt, "MySQL là gì? Hướng dẫn toàn tập về MySQL," MatBao, 22 May 2020. [Online]. Available: https://wiki.matbao.net/mysql-la-gi-huong-dan-toan-tap-ve-mysql/. [Accessed 24 May 2020]. |
| [9] | G. Hai, "MySQL là gì: giải thích tường tận MySQL cho người mới bắt đầu," Hostinger, 07 May 2019. [Online]. Available: https://www.hostinger.vn/huong-dan/mysql-la-gi/. [Accessed 24 May 2020]. |
| [10] | "React (web framework)," Wikipedia, 23 May 2020. [Online]. Available: https://en.wikipedia.org/wiki/React\_(web\_framework). [Accessed 24 May 2020]. |
| [11] | Đ. K. Toàn, "Giới thiệu về ReactJS - Phần I (Các khái niệm cơ bản)," Viblo, 30 May 2017. [Online]. Available: https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7. [Accessed 24 May 2020]. |
| [12] | G. Hai, "Apache là gì? Giải thích cho người mới bắt đầu hiểu kỹ về Apache Web Server," Hostinger, 22 Jan 2019. [Online]. Available: https://www.hostinger.vn/huong-dan/apache-la-gi-giai-thich-cho-nguoi-moi-bat-dau-hieu-ve-apache-web-server/. [Accessed 24 May 2020]. |

# PHỤ LỤC 1: