

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KỲ
THỰC TẬP CƠ SỞ

ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN HÀNG
SỬ DỤNG CÔNG NGHỆ ASP.NET STACK

Giảng viên hướng dẫn: Kim Ngọc Bách

Sinh viên thực hiện: Vũ Văn Hậu

Mã sinh viên: B21DCCN334

Nhóm: D21CQCN04-B

Hà Nội - 2024

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI	5
1.1. Giới thiệu đề tài	5
1.2. Mục đích của đề tài	6
CHƯƠNG 2: KIẾN THỨC, CÔNG NGHỆ, NGÔN NGỮ SỬ DỤNG	7
2.1. FrontEnd	7
2.1.1. HTML, CSS, JavaScript	7
2.1.2. ReactJS	7
2.2. BackEnd	10
2.2.1. Một số khái niệm cơ bản	10
2.2.2. ASP.Net Web Core API	12
2.2.3. SQL Server	13
CHƯƠNG 3: BÁO CÁO QUẢN LÝ DỰ ÁN	14
3.1. Phân Tích Yêu Cầu Chức Năng	14
3.2. Phân Tích Thiết Kế Dự Án	17
3.2.1. Thiết kế tổng quan	17
3.2.1.1. Sơ đồ UseCase	17
3.2.1.2. Sơ đồ lớp	20
3.2.2. Thiết kế chi tiết	21
3.2.2.1. Thiết kế cơ sở dữ liệu	21
3.2.2.2. Viết API	24
3.2.2.3. Thiết kế giao diện	25
3.3. Quản Lý Tiến Độ	30
3.4. Kết Quả	31
CHƯƠNG 4: KẾT LUẬN	32
TÀI LIỆU THAM KHẢO	34

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến “Học viện Công nghệ Bưu chính viễn thông” đã đưa môn học Thực tập cơ sở vào chương trình giảng dạy.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giáo viên bộ môn – Kim Ngọc Bách đã miệt mài dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia học Thực tập cơ sở, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn Thực tập cơ sở là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ. Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung đó. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Lập trình web là một lĩnh vực tập trung vào việc xây dựng và phát triển các trang web và ứng dụng web. Với sự phát triển vượt bậc của Internet, Lập trình web đã trở thành một trong những kỹ năng quan trọng và được săn đón nhiều nhất trong thị trường lao động hiện nay.

Bằng việc lựa chọn và thực hiện đề tài “Xây dựng website bán hàng điện tử H-Mart”, em muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty. Website mang lại cho công ty rất nhiều lợi ích như: khả năng quảng cáo, phổ biến tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh sẽ mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, quảng cáo, dễ dàng nhận phản hồi từ phía khách hàng, cơ hội mở rộng liên kết và hợp tác ở phạm vi quốc tế...v.v..

Website bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được thứ mình cần. Ngoài ra, hệ thống mang lại sự tiện dụng trong việc cập nhật và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dàng với khách hàng.

Nội dung báo cáo được trình bày thành 4 chương:

Chương I: Giới thiệu đề tài

Chương II: Tìm hiểu chung

Chương III: Phân tích và thiết kế hệ thống

Chương IV: Kết luận

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu đề tài

Hiện nay, việc bán hàng qua mạng đã trở thành một xu hướng phổ biến và không còn xa lạ đối với mọi người. Không thể phủ nhận rằng việc kinh doanh trực tuyến mang lại nhiều lợi ích đáng kể. Một trong những cách để tiến xa hơn trong việc bán hàng trực tuyến là xây dựng một website bán hàng chuyên nghiệp và hiệu quả.

Mặc dù ngôn ngữ lập trình như C và C++ có khả năng xây dựng website, nhưng trình biên dịch của chúng lại phụ thuộc vào từng loại CPU. Điều này đồng nghĩa rằng để có một trình biên dịch tương thích, ta cần tốn nhiều thời gian và kinh phí để phát triển và duy trì cho từng loại CPU riêng biệt.

Tuy nhiên, việc xây dựng một website bán hàng mang lại nhiều lợi ích đáng kể. Người mua chỉ cần truy cập trang web bán hàng để lựa chọn và mua những sản phẩm cần thiết mà không cần phải đến tận cửa hàng. Điều này giúp tiết kiệm rất nhiều thời gian và công sức cho khách hàng. Hơn nữa, việc thanh toán cũng được thực hiện trực tuyến và nhanh chóng hơn, mang lại sự thuận tiện và tiện ích cho người mua hàng. Một điểm đáng chú ý nữa là khách hàng đặt hàng sẽ được giao hàng đến tận nơi trong khoảng thời gian ngắn sau khi đặt hàng.

Web HMart được xây dựng nhằm cung cấp một môi trường tương tác và mua sắm trực tuyến cho người dùng, đáp ứng nhu cầu mua sắm đa dạng của khách hàng và cung cấp những thông tin mới nhất về các sản phẩm. Để đạt được điều này, HMart chú trọng đến các tiêu chí sau:

- **Giao diện:** Giao diện đơn giản, đẹp mắt và dễ thao tác trên màn hình. Giao diện được thiết kế sao cho phù hợp với mọi đối tượng khách hàng.
- **Sản phẩm:** Cung cấp một loạt các mẫu mã sản phẩm đa dạng, kèm theo hình ảnh chân thực và chi tiết.

1.2. Mục đích của đề tài

Web HMart được xây dựng với mục tiêu cung cấp cho khách hàng một môi trường tương tác và mua sắm trực tuyến tốt nhất, đáp ứng nhu cầu mua sắm đa dạng của họ. Dưới đây là những mục đích cụ thể của web HMart:

- Cung cấp trải nghiệm mua sắm trực tuyến tốt nhất: Web HMart tập trung vào việc tạo ra một trải nghiệm mua sắm trực tuyến dễ dàng, thuận tiện và tiện ích cho khách hàng. Giao diện web được thiết kế đẹp mắt, dễ sử dụng và tối ưu hóa trải nghiệm người dùng.
- Đáp ứng nhu cầu mua sắm đa dạng của khách hàng: HMart cung cấp một loạt các sản phẩm từ nhiều danh mục khác nhau nhằm đáp ứng nhu cầu mua sắm đa dạng của khách hàng.
- Tạo dựng lòng tin và đáng tin cậy từ khách hàng: Bằng cách cung cấp thông tin chính xác và đáng tin cậy về sản phẩm, đảm bảo chất lượng và sự an toàn của giao dịch mua hàng trực tuyến.
- Tăng doanh số bán hàng: Một mục đích quan trọng của việc xây dựng website bán hàng trực tuyến là tăng doanh số bán hàng. Bằng cách tạo ra một giao diện hấp dẫn và tiện lợi, khách hàng có thể dễ dàng duyệt qua các sản phẩm, thực hiện giao dịch mua hàng và thanh toán trực tuyến.
- Mở rộng thị trường và tiếp cận khách hàng mới: Website bán hàng trực tuyến mang lại lợi ích vượt trội về khả năng tiếp cận khách hàng từ khắp mọi nơi trên thế giới. Điều này mở ra cơ hội mở rộng thị trường kinh doanh, không chỉ giới hạn trong khu vực địa lý cố định.
- Nâng cao hiệu quả quản lý và tối đa hóa quy trình kinh doanh: Xây dựng một website bán hàng trực tuyến đồng nghĩa với việc tạo ra một hệ thống quản lý đơn hàng, hệ thống quản lý kho hàng và quy trình kinh doanh tự động.

CHƯƠNG 2: KIẾN THỨC, CÔNG NGHỆ, NGÔN NGỮ SỬ DỤNG

2.1. FrontEnd

2.1.1. HTML, CSS, JavaScript

- *Khái niệm cơ bản về HTML:*

HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu để xây dựng và cấu trúc nội dung trên một trang web. Nó đóng vai trò như khung xương cơ bản cho trang web, cho phép xác định cách các phần tử được hiển thị và tổ chức trên trang. Thông qua các thẻ HTML, chúng ta sẽ định hình được hình hài của 1 trang web sẽ trông như thế nào và tổ chức sao cho hợp lý. Nếu không có HTML, trang web của chúng ta sẽ không thể hoạt động, HTML đóng vai trò là nền móng cho một trang web.[1]

- *Khái niệm cơ bản về CSS:*

Nếu HTML là nền móng, là khung xương cho một trang web thì CSS (Cascading Style Sheets) là ngôn ngữ định dạng để trình bày và tạo kiểu cho các phần tử HTML trên trang web, là da thịt và là quần áo khiến cho trang web hiển thị một cách dễ nhìn và giúp người dùng dễ dàng thao tác cũng như sử dụng website hơn. CSS đóng một vai trò cũng hết sức quan trọng để giúp tối ưu trải nghiệm của người dùng và giúp trang web của chúng ta dễ nhìn hơn.[2]

- *Khái niệm cơ bản về JavaScript:*

Javascript là ngôn ngữ chuyên dùng cho lập trình một trang web. Nếu CSS cung cấp cho HTML những thuộc tính về màu sắc, về cách hiển thị, Thì JavaScript cung cấp cho HTML dom những hành động. Giả sử khi chúng ta ấn vào 1 nút bất kì thì sẽ xảy ra hành động gì? Javascript sẽ cung cấp những hành động cho trang web của chúng ta. Nó là một phần quan trọng để làm cho trang web trở nên sống động và tương tác.[3]

2.1.2. ReactJS

- *Khái niệm về ReactJS:*

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Với react là một

library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.[4]

- *Khái niệm về JSX:*

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. 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.[4]

- *Khái niệm về Components:*

React xây dựng trang web bằng cách sử dụng các thành phần (component) nhỏ thay vì sử dụng template như các framework khác. Mỗi component có trạng thái riêng và có thể thay đổi. React sẽ tự động cập nhật component dựa trên các thay đổi của trạng thái. Component có thể được tái sử dụng ở nhiều nơi với các trạng thái và thuộc tính khác nhau và có thể chứa các component khác bên trong. React xem mọi thứ là component, giúp duy trì mã code trong các dự án lớn. Một component đơn giản của React chỉ cần có một phương thức render, và cũng có nhiều phương thức khác nhưng render là phương thức chủ đạo.[4]

- *Khái niệm về Virtual DOM:*

Công nghệ Virtual DOM trong ReactJS giúp cải thiện hiệu năng của ứng dụng. Khi một phần tử gốc có trạng thái thay đổi, Thay vì cập nhật trực tiếp lên DOM thật, ReactJS sử dụng một DOM ảo để tính toán sự thay đổi và chỉ cập nhật những phần tử DOM thật bị ảnh hưởng. Điều này giúp tối ưu hóa quá trình render DOM và cải thiện tốc độ xử lý của ứng dụng. Nó sử dụng cơ chế one-way data binding - luồng dữ liệu một chiều để dễ dàng quản lý và truyền dữ liệu, giúp tối ưu hóa quá trình render và giảm thiểu lỗi. Điều này làm cho ReactJS trở thành công cụ hiệu quả cho các ứng dụng web động và phức tạp.[4]

- *Khái niệm về Redux:*

Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.[4]

- *Ưu điểm của ReactJS:*

- Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng.[5]
- Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.
- Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.
- Hỗ trợ tốt cho SEO: ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.
- Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.[5]

2.2. BackEnd

2.2.1. Một số khái niệm cơ bản

- *Khái niệm về API:*

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

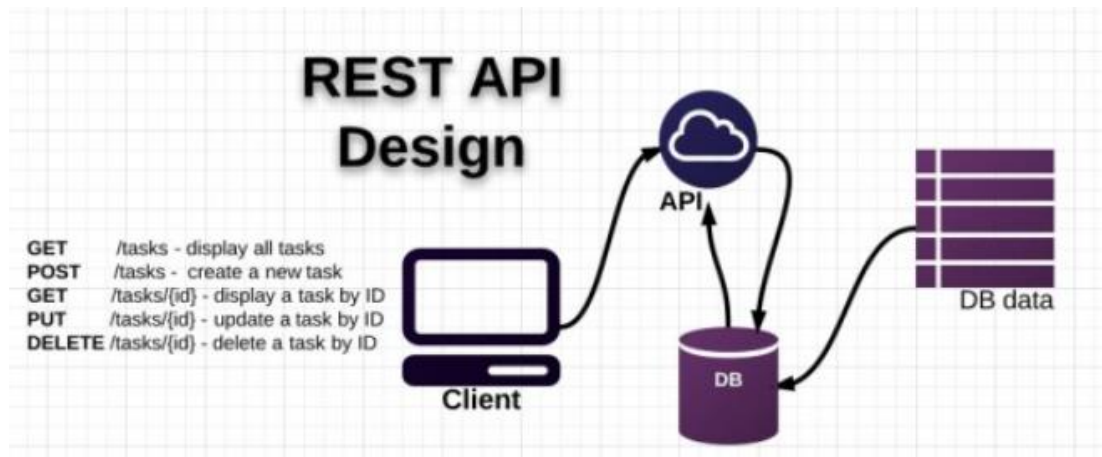
Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.[6]

- *Khái niệm về REST:*

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE,... đến một URL để xử lý dữ liệu.[6]

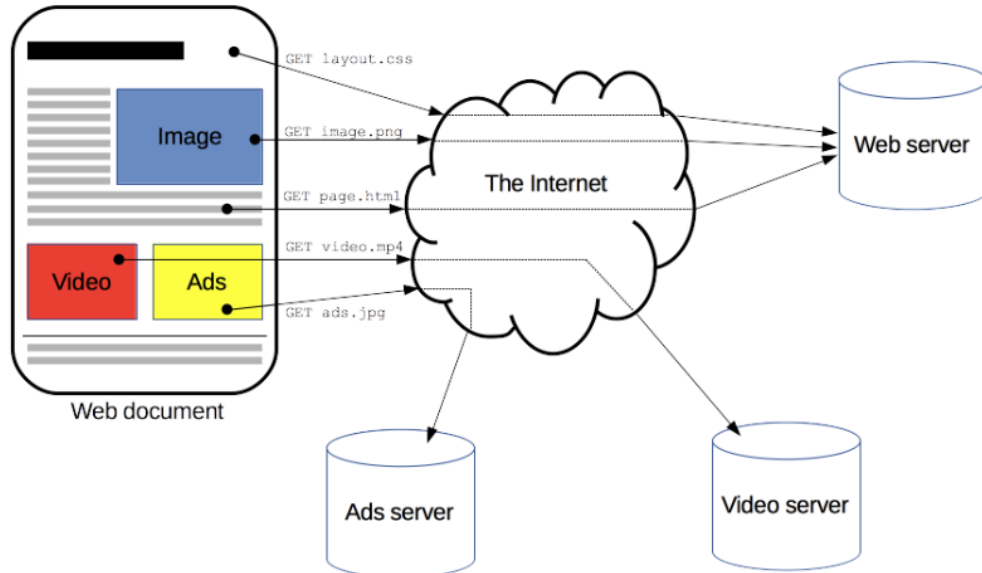
- *Khái niệm về RESTful API:*

RESTful API là một kiểu thiết kế API dựa trên nguyên tắc của REST. REST (Representational State Transfer) là một kiến trúc phần mềm định nghĩa các tiêu chuẩn và quy tắc cho việc xây dựng dịch vụ web. RESTful API sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để thao tác với các tài nguyên (resources) trên server. Nó cho phép truy cập và thay đổi trạng thái của tài nguyên thông qua các yêu cầu HTTP.[6]



- *Khái niệm về giao thức HTTP:*

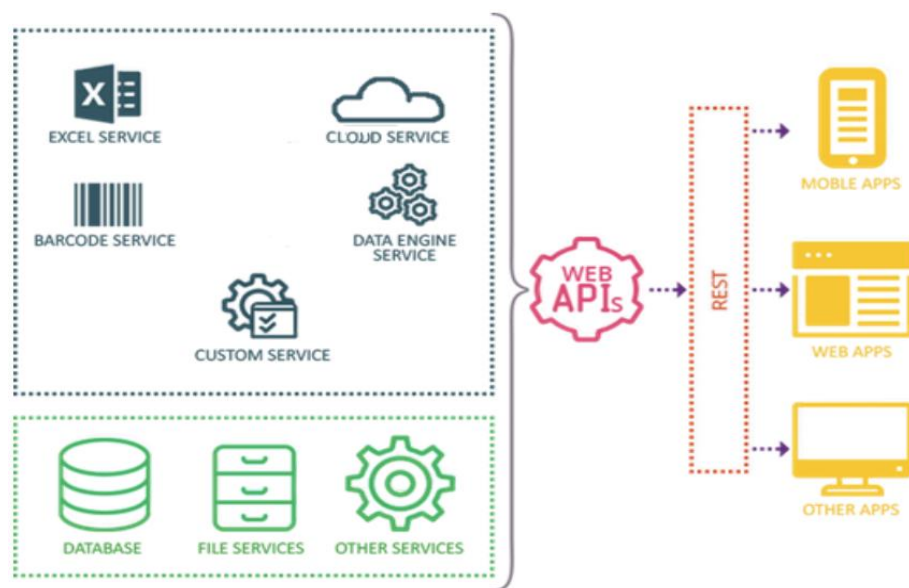
HTTP (Hyper Text Transfer Protocol) nghĩa là Giao thức Truyền tải Siêu Văn Bản được sử dụng trong www. HTTP là 1 giao thức cho phép tìm nạp tài nguyên, chẳng hạn như HTML doc. Nó là nền tảng của bất kỳ sự trao đổi dữ liệu nào trên Web và cũng là giao thức giữa client (thường là các trình duyệt hay bất kỳ loại thiết bị, chương trình nào) và server (thường là các máy tính trên đám mây). 1 doc hoàn chỉnh được tái tạo từ các doc con khác nhau được fetch – tìm nạp, chẳng hạn như văn bản, mô tả layout, hình ảnh, video, script...[6]



2.2.2. ASP.Net Web Core API

ASP.NET Core Web API là một framework được sử dụng để xây dựng các dịch vụ web API. Nó được phát triển bởi Microsoft và hoạt động trên nền tảng .NET Core. ASP.NET Core Web API cung cấp các công cụ, tính năng để tạo ra các dịch vụ web API đáp ứng nhu cầu của các ứng dụng web hiện đại.

Các ứng dụng web thường cần liên lạc với nhiều hệ thống khác nhau và trao đổi dữ liệu. Các dịch vụ web API là một cách để thực hiện việc này, cho phép các ứng dụng trao đổi dữ liệu thông qua một giao diện lập trình ứng dụng. ASP.NET Core Web API cung cấp các tính năng để xử lý các yêu cầu HTTP từ các ứng dụng khác, bao gồm định dạng dữ liệu, bảo mật và lỗi xử lý.[7]



- Ưu điểm của ASP.NET Core Web API:[8]
 - Độ tin cậy cao: ASP.NET Core Web API được phát triển bởi Microsoft, là một công ty công nghệ hàng đầu của thế giới, nên độ tin cậy của nó rất cao.
 - Thiết kế dựa trên quy tắc RESTful: ASP.NET Core Web API được thiết kế dựa trên quy tắc RESTful, cho phép phản hồi nhanh chóng với các yêu cầu HTTP và truy cập vào các tài nguyên dễ dàng.
 - Hiệu suất tối ưu: Với việc sử dụng công nghệ mới và tối ưu, ASP.NET Core Web API có hiệu suất tốt hơn so với các khuôn khổ khác.
 - Linh hoạt và dễ dàng mở rộng: Với việc hỗ trợ đa nền tảng và sử dụng các thư viện mã nguồn mở, ASP.NET Core Web API có khả năng linh hoạt và dễ dàng mở rộng hơn.

- Hỗ trợ tính năng mới nhất: Với việc đến từ Microsoft và được đóng góp từ cộng đồng, ASP.NET Core Web API được cập nhật và hỗ trợ các tính năng mới nhất.

2.2.3. SQL Server

SQL Server (viết tắt của cụm từ Structured Query Language) là một hệ thống quản lý cơ sở dữ liệu quan hệ (Relational Database Management System, viết tắt là RDBMS). SQL Server có khả năng hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty hoạt động trong lĩnh vực IT.[9]

Cũng giống như các hệ thống quản lý cơ sở dữ liệu qua hệ khác, SQL Server được xây dựng trên lớp SQL – là ngôn ngữ lập trình tiêu chuẩn hoá được quản trị viên cơ sở dữ liệu (DBAs) và các chuyên gia IT sử dụng để quản lý cơ sở dữ liệu và truy vấn các dữ liệu nằm bên trong.[9]

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập. Nó là thành phần với vai trò ngôn ngữ làm công cụ giao tiếp của cơ sở dữ liệu và người dùng. Bởi vậy, nó được sử dụng trong các dịch vụ thiết kế web được tích hợp tính năng giao tiếp với người dùng với các vai trò sau:

- SQL là một ngôn ngữ đòi hỏi sự tương tác cao. Người dùng có thể dễ dàng trao đổi với tiện ích bằng câu lệnh SQL đến cơ sở dữ liệu và nhận lại kết quả từ đó.
- SQL là một ngôn ngữ lập trình cơ sở dữ liệu. Bằng cách nhúng các câu lệnh SQL trong ngôn ngữ lập trình, các lập trình viên có thể xây dựng được các chương trình ứng dụng giao tiếp với cơ sở dữ liệu
- SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu. Quản trị viên cơ sở dữ liệu có thể quản lý và điều khiển các truy cập tới cơ sở dữ liệu thông qua SQL
- SQL được sử dụng như một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách – chủ.
- SQL được sử dụng với vai trò tương tác với các dữ liệu trong các máy chủ web và máy chủ Internet
- SQL có vai trò giao tiếp với các hệ thống trên mạng, làm nhiệm vụ gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.[9]

CHƯƠNG 3: BÁO CÁO QUẢN LÝ DỰ ÁN

3.1. Phân Tích Yêu Cầu Chức Năng

- **Phía admin:**

1. Đăng nhập

- Form gồm 2 ô nhập thông tin email và password.
- Nếu nhập sai email hoặc password sẽ hiện thông báo “Sai tài khoản hoặc mật khẩu”.

2. Đăng xuất

- Nhấp vào nút logout sẽ đăng xuất khỏi tài khoản, điều hướng về trang đăng nhập.

3. Xem danh sách các sản phẩm

- Hiện thị toàn bộ danh sách sản phẩm dưới dạng bảng gồm các thông tin của sản phẩm: Name, Manufacturer, UnitPrice, Discount, Quantity, ExpDate, Image
- Mỗi sản phẩm có 2 nút Edit và Delete

4. Thêm sản phẩm mới

- Form nhập thông tin sản phẩm mới gồm các trường: Name, Manufacturer, UnitPrice, Discount, Quantity, ExpDate, ImageUrl
- Các trường không được để trống
- Có tích hợp tải hình ảnh sản phẩm từ máy tính
- Sau khi bấm Add thông báo thêm sản phẩm thành công

5. Chỉnh sửa thông tin sản phẩm

- Form nhập sửa sản phẩm mới gồm các trường giống với thêm sản phẩm mới
- Sau khi bấm Update thông báo hiển thị Cập nhật thành công

6. Xóa sản phẩm

- Sau khi bấm Delete thông báo hiển thị Xóa thành công sản phẩm

7. Xem danh sách khách hàng

- Danh sách các khách hàng hiển thị theo dạng bảng biểu gồm các trường: FirstName, LastName, Password, Email, Fund, Type, Status, CreatedOn

8. Chỉnh sửa thông tin khách hàng

- Gồm form chỉnh sửa số tiền của khách hàng, gồm 1 trường Fund

9. Xem danh sách các đơn hàng

- Danh sách đơn hàng hiển thị dưới dạng bảng

- Gồm các trường: CustomerName, OrderNo, OrderTotal, OrderStatus, CreatedOn
 - Và 1 nút Update Status
10. Xem chi tiết các mặt hàng của đơn hàng
- Khi bấm vào đơn hàng bất kỳ, hiển thị chi tiết các mặt hàng của đơn hàng theo dạng bảng, gồm các trường: ProductName, Manufacturer, UnitPrice, Quantity, OrderDate
11. Chỉnh sửa trạng thái đơn hàng
- Bấm Update, giao diện chỉnh sửa trạng thái đơn hàng hiện ra

• **Phía Client:**

1. Đăng ký
 - Form đăng ký gồm 4 ô nhập firstname, lastname, email, password
 - Nếu thiếu 1 trường nào đó hiển thị thông báo yêu cầu điền đầy đủ các thông tin
 - Nếu nhập đủ thông tin bấm Register, hiển thị thông báo Đăng ký tài khoản thành công, Chuyển hướng về lại trang đăng nhập
2. Đăng nhập
 - Form gồm 2 ô nhập thông tin email và password.
 - Nếu nhập sai email hoặc password sẽ hiện thông báo “Sai tài khoản hoặc mật khẩu”.
3. Đăng xuất
 - Nhấp vào nút logout sẽ đăng xuất khỏi tài khoản, điều hướng về trang đăng nhập.
4. Xem thông tin tài khoản
 - Thông tin tài khoản hiển thị gồm các trường: FirstName, LastName, Password, Email, nút Update và Reset
5. Chỉnh sửa thông tin tài khoản
 - Các trường hiển thị thông tin khách hàng như phần xem thông tin
 - Sau khi bấm update hiển thị thông báo chỉnh sửa thông tin thành công
6. Xem danh sách sản phẩm
 - Danh sách sản phẩm hiển thị gồm hình ảnh, giá và tên sản phẩm
7. Xem danh sách sản phẩm theo phân loại
 - Hiển thị các loại sản phẩm, theo dạng nút

- Khách hàng chọn 1 phân loại thì hiển thị danh mục các sản phẩm của danh mục
- 8. Tìm kiếm sản phẩm
 - Form tìm kiếm sản phẩm hiển thị gồm trường: Name, Manufacturer, UnitPrice và nút Find
 - Khi bấm Find, hiển thị danh sách các sản phẩm tồn tại
 - Nếu không có sản phẩm hợp lệ hiện không tồn tại sản phẩm hợp lệ
- 9. Thêm sản phẩm vào giỏ hàng
 - Thêm sản phẩm gồm 1 trường: Quantity, chọn số lượng sản phẩm
 - Sau khi chọn Quantity bấm Add Cart, sản phẩm được thêm vào giỏ hàng, thông báo thêm thành công
- 10. Xóa sản phẩm khỏi giỏ hàng
 - Sau khi bấm nút Remove, thông báo xóa thành công sản phẩm
- 11. Xem các đơn hàng đã đặt
 - Danh sách đơn hàng hiển thị dưới dạng bảng
 - Gồm các trường: CustomerName, OrderNo, OrderTotal, OrderStatus, CreatedOn, và 1 nút xuất hóa đơn
- 12. Xem chi tiết các mặt hàng của đơn hàng
 - Khi bấm vào đơn hàng bất kỳ, hiển thị chi tiết các mặt hàng của đơn hàng theo dạng bảng, gồm các trường: ProductName, Manufacturer, UnitPrice, Quantity, OrderDate
- 13. Xuất hóa đơn
 - Sau khi bấm xuất hóa đơn, giao diện hóa đơn hiển thị gồm các mặt hàng trong hóa đơn và tổng tiền
- 14. Thanh toán
 - Hiển thị đơn hàng cần thanh toán gồm sản phẩm và tổng tiền đơn hàng.
 - Thanh toán thành công sẽ hiển thị thông báo thanh toán thành công

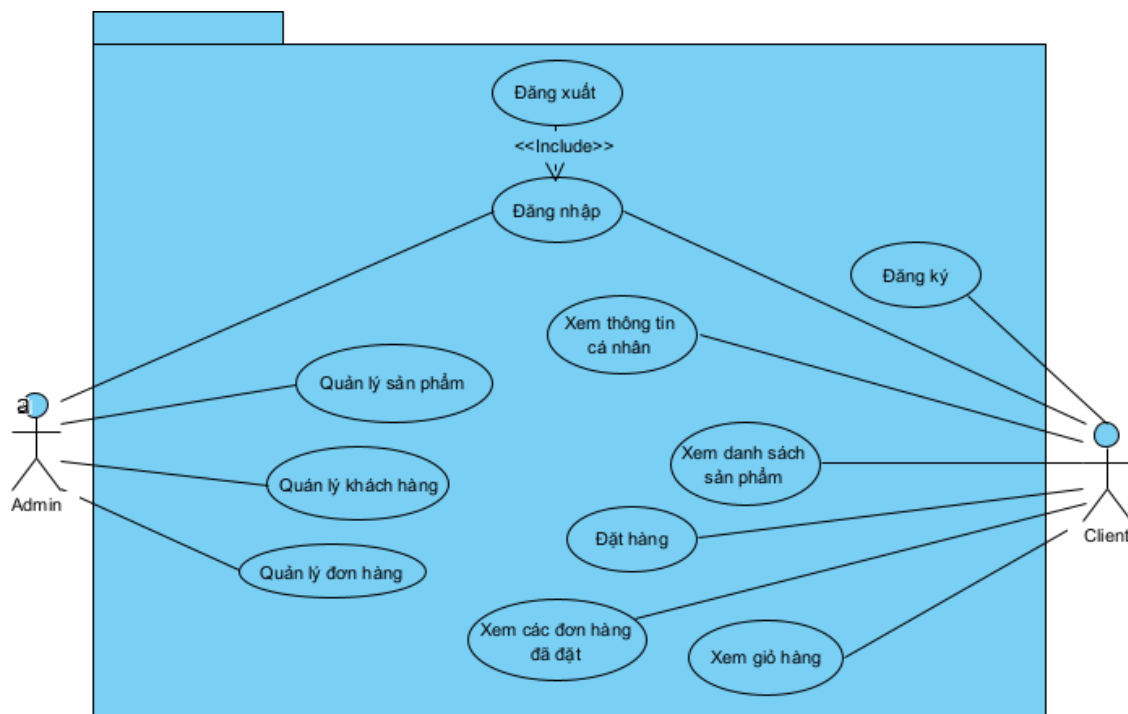
3.2. Phân Tích Thiết Kế Dự Án

3.2.1. Thiết kế tổng quan

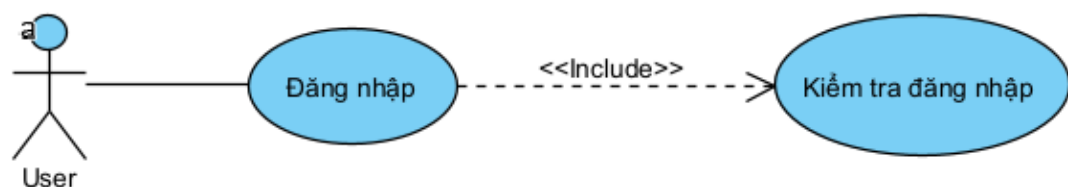
3.2.1.1. Sơ đồ UseCase

a) *Use case tổng quan toàn bộ hệ thống:*

- Có thể đề xuất được các actor chính của hệ thống – những người tương tác trực tiếp với hệ thống: Quản trị viên(Admin), Khách hàng(Client)
- Các chức năng tương ứng với từng actor:
 - Client: Đăng ký, Đăng nhập, Chỉnh sửa thông tin cá nhân, Thêm sản phẩm vào giỏ hàng, Đặt hàng, Xuất hóa đơn
 - Admin: Đăng nhập, Quản lý sản phẩm, Quản lý khách hàng, Quản lý đơn hàng

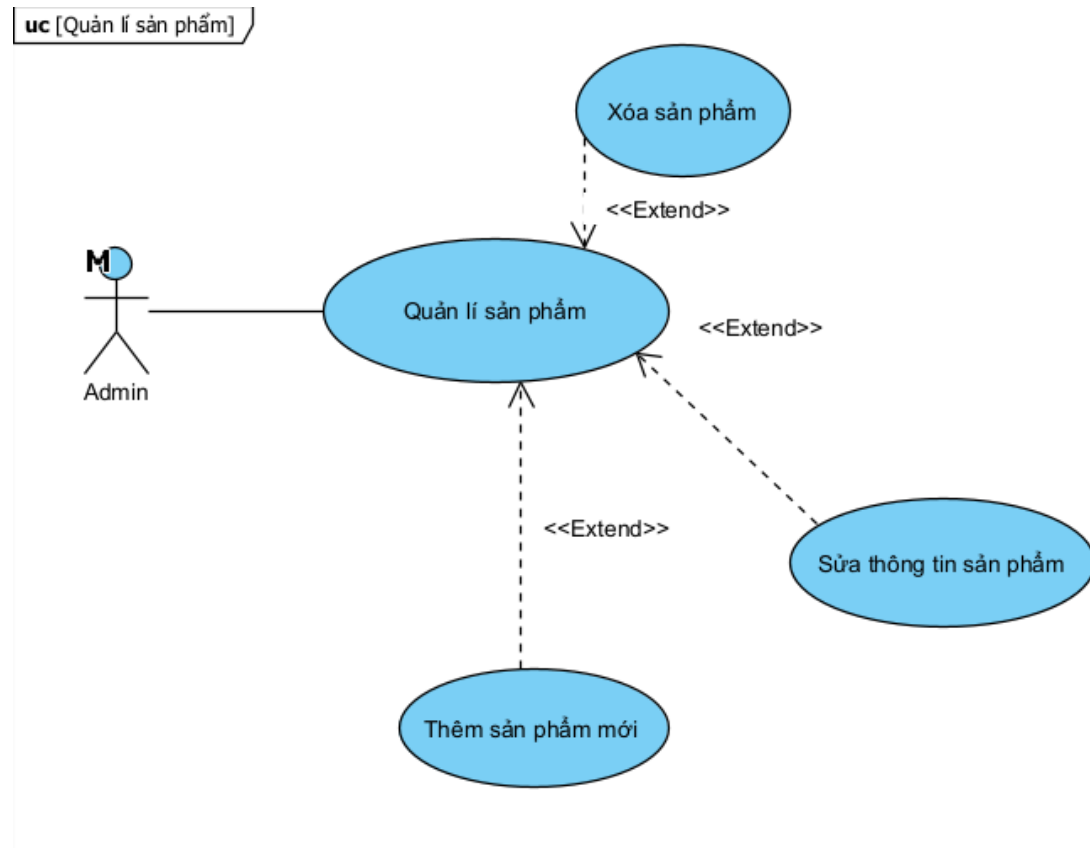


b) *Use case đăng nhập*



Mô tả: Use case cho phép User (Admin và Client) đăng nhập vào hệ thống

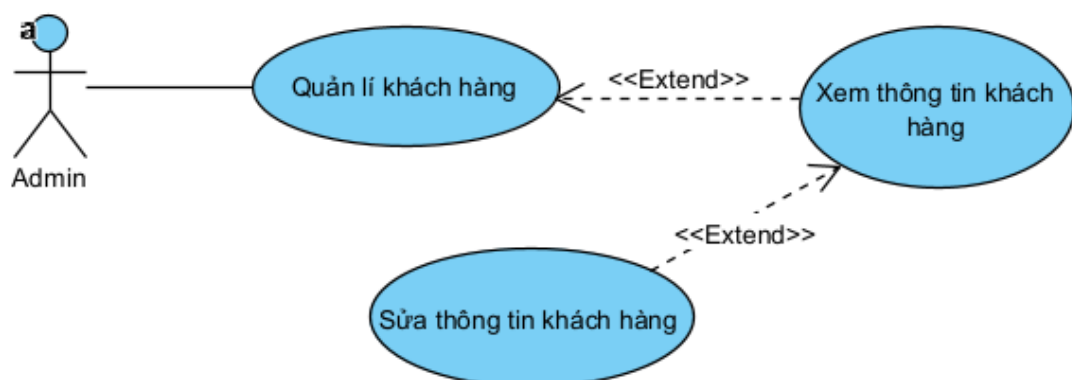
c) Use case quản lí sản phẩm



Mô tả: Use case cho phép Admin xem, sửa, xóa thông tin của sản phẩm trong hệ thống

Điều kiện trước: Admin đã đăng nhập vào hệ thống

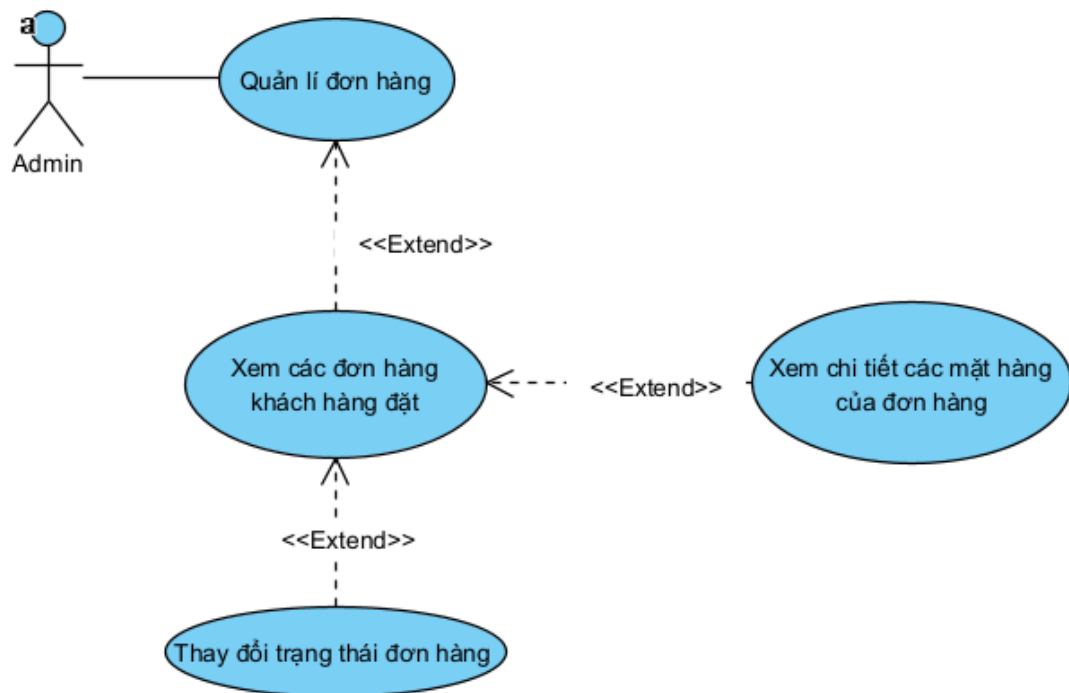
d) Use case quản lí khách hàng



Mô tả: Use case cho phép Admin xem, sửa thông tin của các khách hàng trong hệ thống

Điều kiện trước: Admin đã đăng nhập vào hệ thống

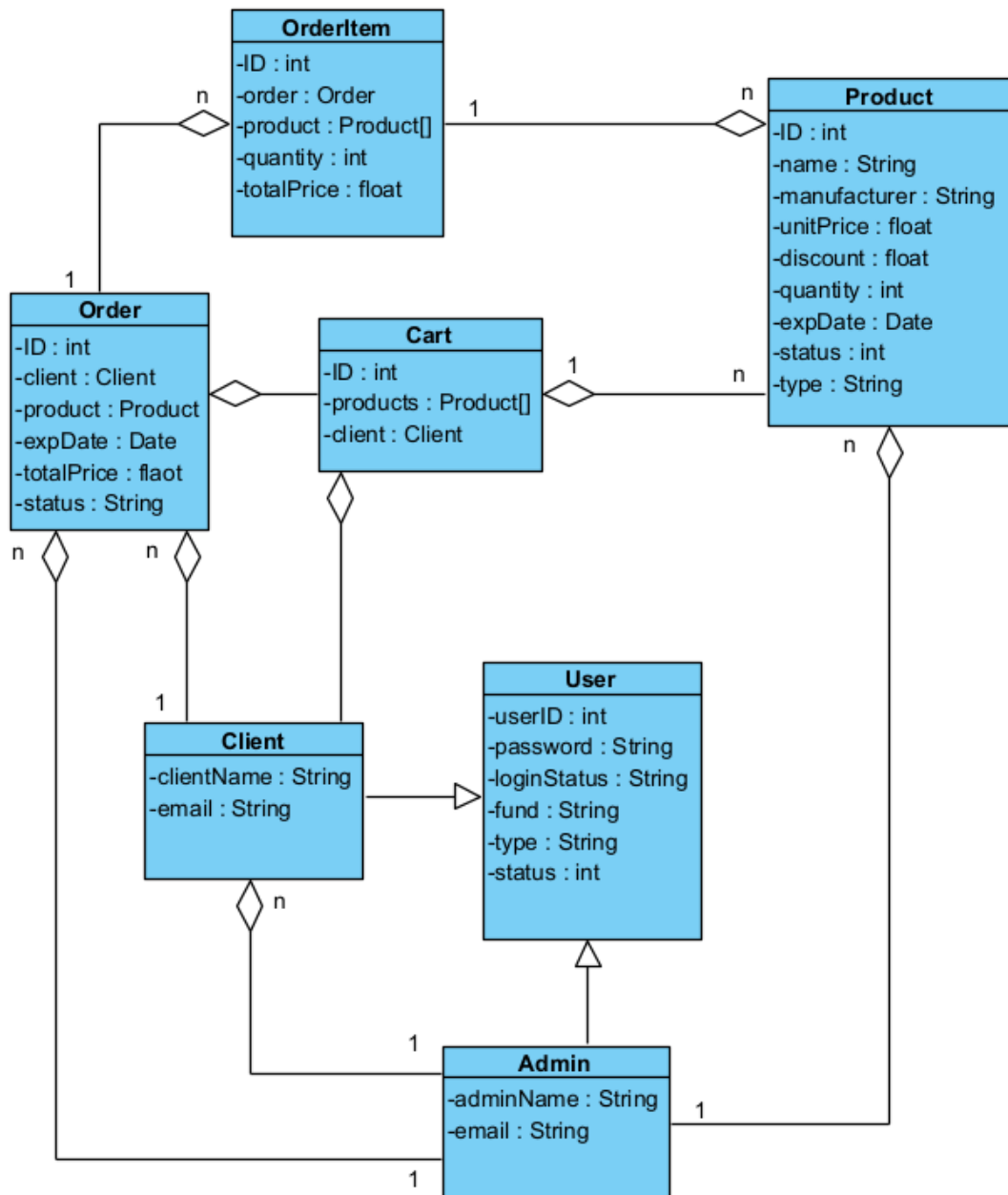
e) *Use case quản lí đơn hàng*



Mô tả: Use case này cho phép Admin xem, thay đổi trạng thái các đơn hàng của khách hàng, cho phép xem chi tiết các mặt hàng của từng đơn hàng

Điều kiện trước: Admin đã đăng nhập vào hệ thống

3.2.1.2. Sơ đồ lớp



3.2.2.Thiết kế chi tiết

3.2.2.1. Thiết kế cơ sở dữ liệu

- **Danh sách các bảng cơ sở dữ liệu:**

STT	Tên thực thể	Mô tả
1	Users	Chứa thông tin khách hàng
2	Products	Chứa thông tin của các sản phẩm
3	Carts	Chứa thông tin giỏ hàng của khách hàng
4	Orders	Chứa thông đơn hàng của khách hàng
5	OrderItems	Chứa thông tin các mặt hàng của đơn hàng

- **Bảng Users:**

STT	Tên thực thể	Kiểu dữ liệu	Mô tả
1	ID	Int(10)	Khóa chính
2	FirstName	Nvarchar(100)	Tên của người dùng
3	LastName	Nvarchar(100)	Họ của người dùng
4	Password	Nvarchar(100)	Mật khẩu của người dùng
5	Email	Nvarchar(100)	Địa chỉ email của người dùng
6	Fund	Float(18)	Số tiền tài khoản của người dùng
7	Type	Nvarchar(100)	Thông tin về loại người dùng
8	Status	Int(10)	Trạng thái của người dùng
9	CreatedOn	Datetime	Thông tin về thời gian tạo người dùng

- **Bảng Products:**

STT	Tên thực thể	Kiểu dữ liệu	Mô tả
1	ID	Int(10)	Khóa chính
2	Name	Nvarchar(100)	Tên sản phẩm
3	Manufaturer	Nvarchar(100)	Tên nhà sản

4	UnitPrice	Float(18)	Giá của sản phẩm
5	Discount	Float(18)	Giảm giá của sản phẩm
6	Quantity	Int(10)	Số lượng sản phẩm có sẵn
7	ExpDate	Datetime	Ngày hết hạn của sản phẩm
8	ImageUrl	Nvarchar(100)	Đường dẫn của hình ảnh sản phẩm
9	Status	Int(10)	Trạng thái của sản phẩm

• **Bảng Carts:**

STT	Tên thực thể	Kiểu dữ liệu	Mô tả
1	ID	Int(10)	Khóa chính
2	UserID	Int(10)	Khóa ngoại tham chiếu đến bảng Users
3	ProductID	Int(10)	Khóa ngoại tham chiếu đến bảng Products
4	UnitPrice	Nvarchar(100)	Giá của sản phẩm
5	Discount	Nvarchar(100)	Giảm giá của sản phẩm
6	Quantity	Nvarchar(100)	Số lượng sản phẩm trong giỏ hàng
7	TotalPrice	Float(18)	Tổng giá của sản phẩm trong giỏ hàng

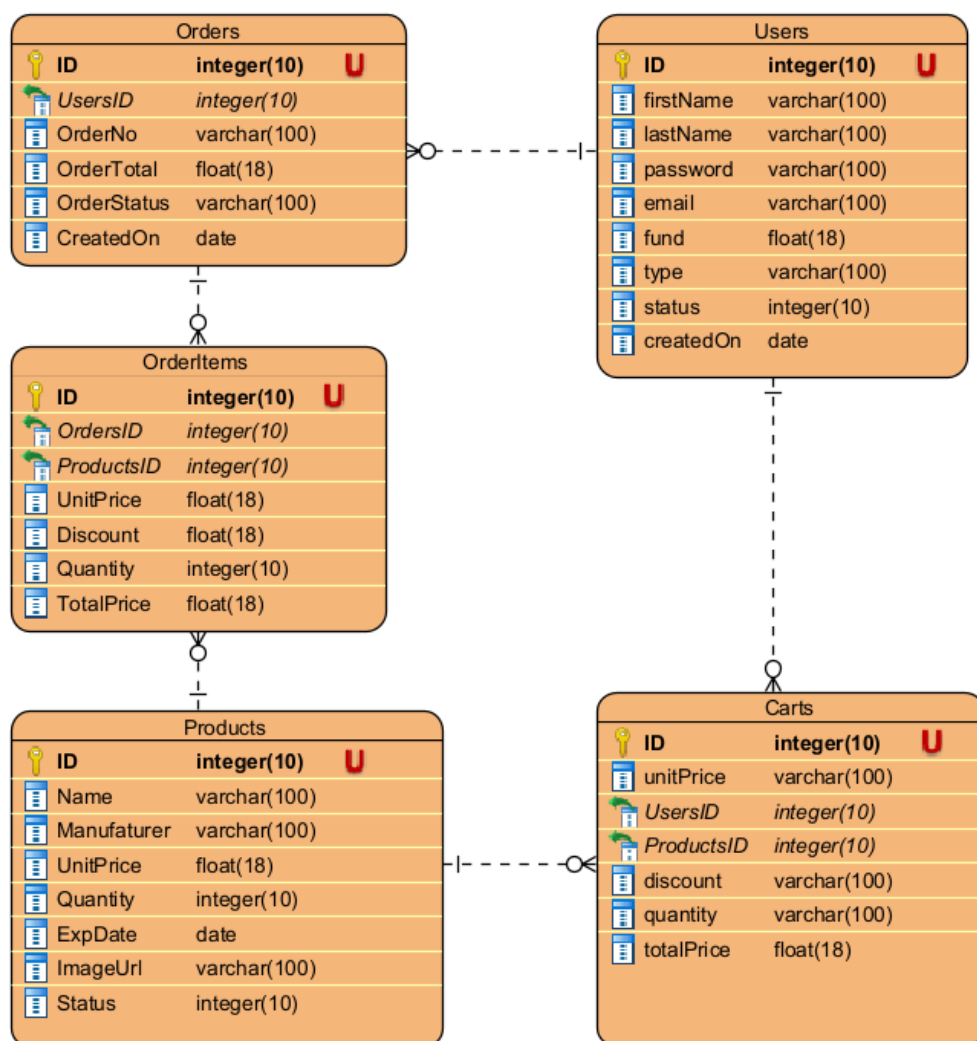
• **Bảng Orders:**

STT	Tên thực thể	Kiểu dữ liệu	Mô tả
1	ID	Int(10)	Khóa chính
2	UserID	Int(10)	Khóa ngoại tham chiếu đến bảng Users
3	OrderNo	Nvarchar(100)	Số thứ tự của đơn hàng
4	OrderTotal	Float(18)	Tổng giá trị của đơn hàng
5	OrderStatus	Nvarchar(100)	Trạng thái của đơn hàng
6	CreatedOn	Datetime	Thời gian tạo đơn hàng

• **Bảng OrderItems:**

STT	Tên thực thể	Kiểu dữ liệu	Mô tả
1	ID	Int(10)	Khóa chính
2	OrderID	Int(10)	Khóa ngoại tham chiếu đến bảng Orders
3	ProductID	Int(10)	Khóa ngoại tham chiếu đến bảng Products
4	UnitPrice	Float(18)	Giá của sản phẩm
5	Discount	Float(18)	Giảm giá của sản phẩm
6	Quantity	Int(10)	Số lượng sản phẩm trong đơn hàng
7	TotalPrice	Float(18)	Tổng giá của sản phẩm trong đơn hàng

- **Biểu đồ CSDL:**



3.2.2.2. Viết API

- Admin

Phương thức	Route	Mô tả
POST	/api/Admin/addUpdateProduct	Chỉnh sửa thông tin sản phẩm
GET	/api/Admin/userList	Xem danh sách khách hàng
POST	/api/Admin/updateFund	Chỉnh sửa tiền khách hàng
POST	/api/Admin/updateOrderStatus	Thay đổi trạng thái đơn hàng
POST	/api/Admin/UploadFile	Thêm ảnh sản phẩm
POST	/api/Admin/cartList	Xem danh sách đơn hàng

- User:

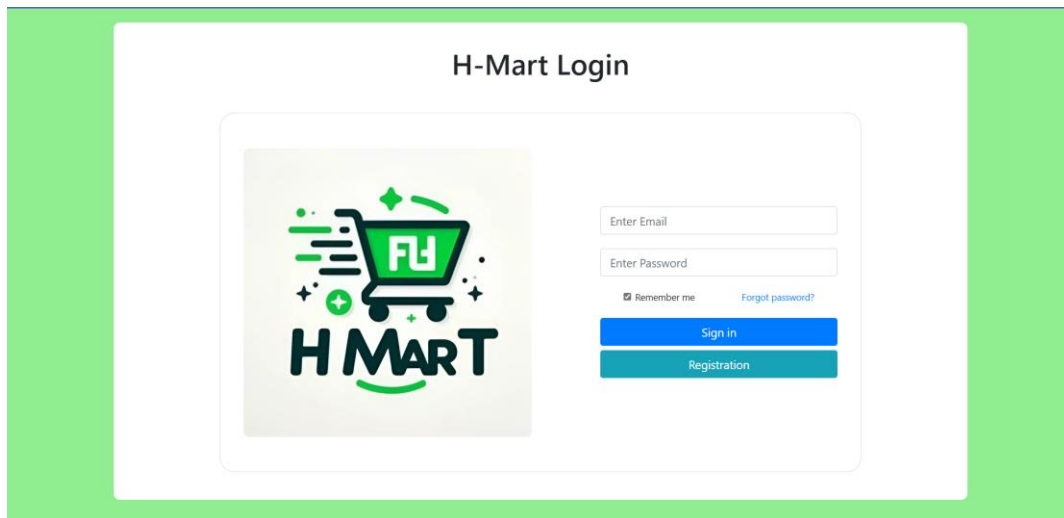
Phương thức	Route	Mô tả
POST	/api/Users/registration	Đăng ký
POST	/api/Users/login	Đăng nhập
POST	/api/Users/viewUser	Xem danh thông tin khách hàng
POST	/api/Users/updateProfile	Chỉnh sửa thông tin cá nhân

- Product:

Phương thức	Route	Mô tả
POST	/api/Products/addToCart	Thêm sản phẩm vào giỏ hàng
POST	/api/Products/removeFromCart	Xóa sản phẩm khỏi giỏ hàng
POST	/api/Products/placeOrder	Xem danh sách các đơn hàng
POST	/api/Products/orderList	Xem danh sách các mặt hàng của đơn hàng

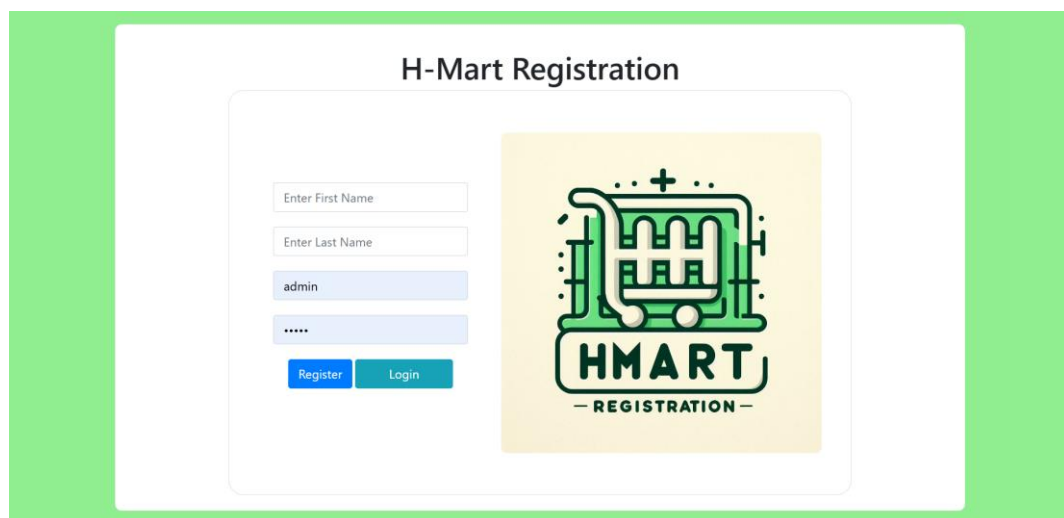
3.2.2.3. Thiết kế giao diện

- Giao diện đăng nhập:



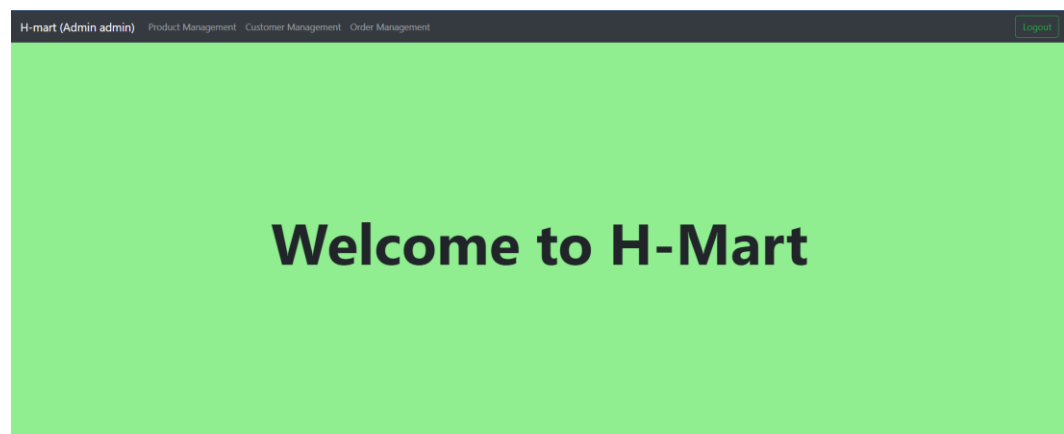
The login interface is titled "H-Mart Login". It features a large logo on the left showing a shopping cart with the letters "FM" inside, and the text "H MART" below it. On the right, there are two input fields: "Enter Email" and "Enter Password". Below the password field, there is a checkbox labeled "Remember me" and a link "Forgot password?". At the bottom, there are two buttons: a blue "Sign in" button and a teal "Registration" button.

- Giao diện đăng ký:

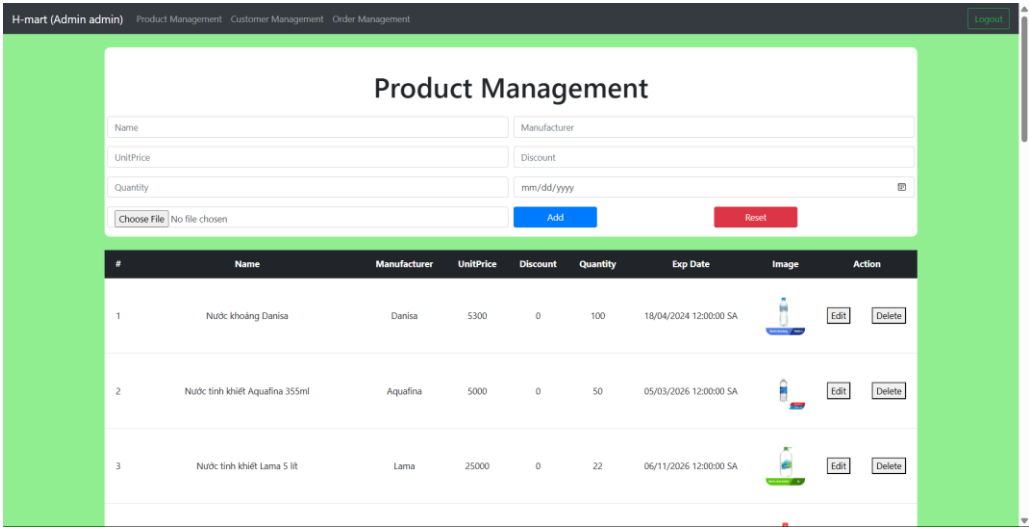


The registration interface is titled "H-Mart Registration". It features a large logo on the right showing a shopping cart with the letters "HM" inside, and the text "HMART" below it, with "— REGISTRATION —" underneath. On the left, there are four input fields: "Enter First Name", "Enter Last Name", a field containing "admin", and a field with four dots. At the bottom, there are two buttons: a blue "Register" button and a teal "Login" button.

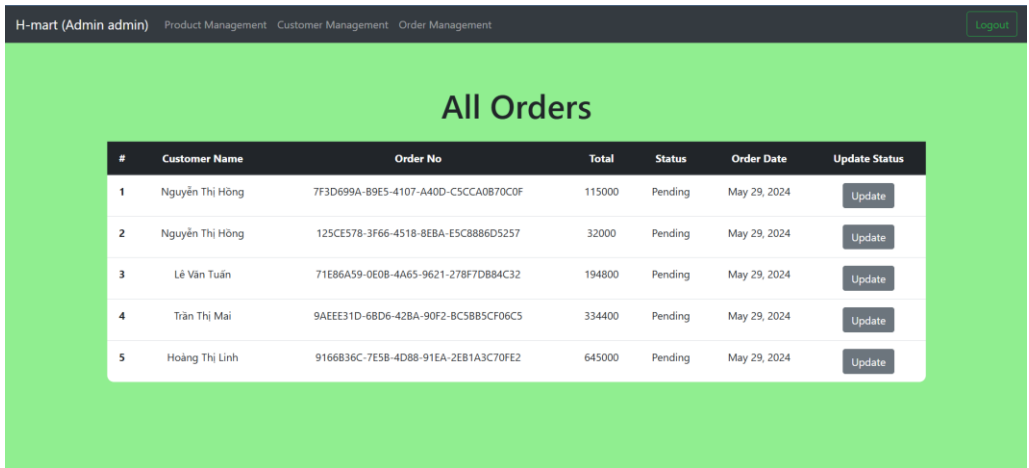
- Giao diện trang chủ Admin:



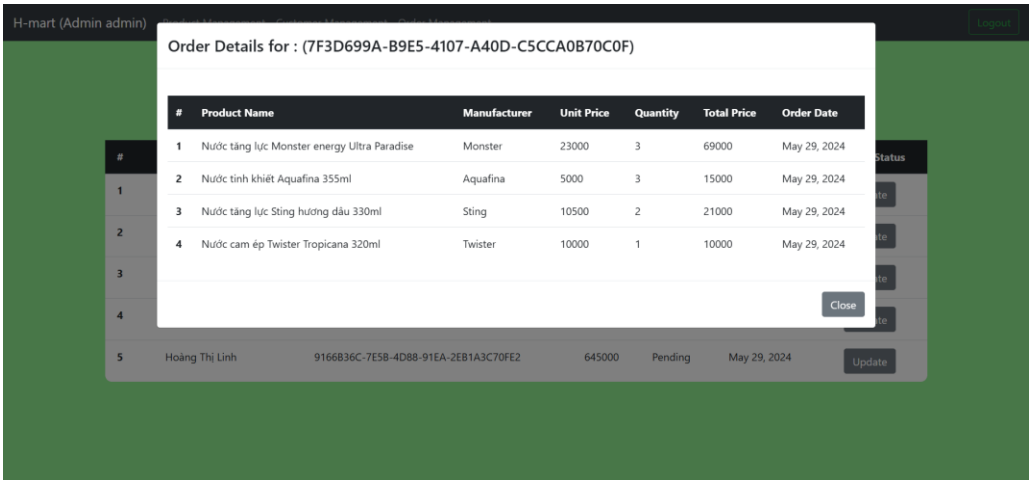
• **Giao diện quản lí sản phẩm:**



• **Giao diện quản lí đơn hàng:**



• **Giao diện hiển thị chi tiết các sản phẩm của đơn hàng:**



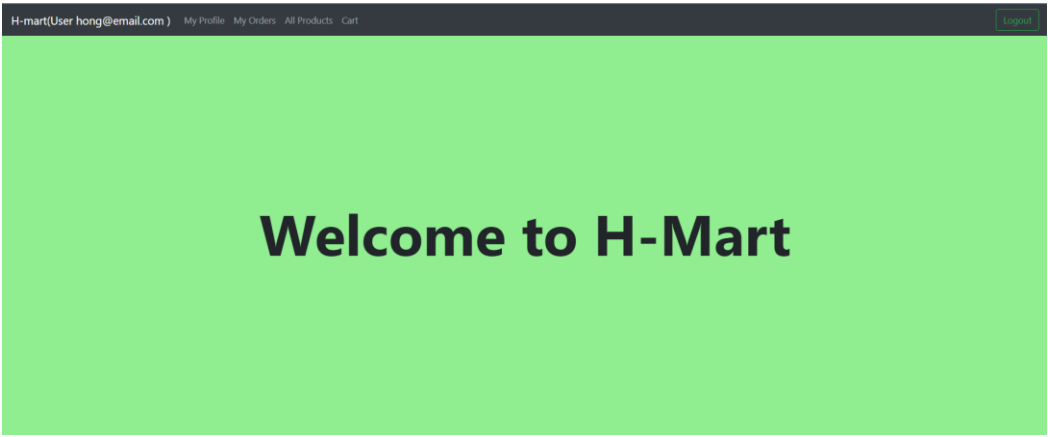
- **Giao diện quản lí khách hàng:**

H-mart (Admin admin)Product ManagementCustomer ManagementOrder ManagementLogout

Customer List

#	First Name	Last Name	Email	Password	Funds	Status	Registration Date	Manage Funds
1	Nguyễn Thị	Hồng	hong@email.com	123	1500000.00	1	May 25, 2024	Update
2	Trần Thị	Mai	mai@email.com	123	2000000.00	1	May 25, 2024	Update
3	Lê Văn	Tuân	tuai@email.com	123	1800000.00	1	May 25, 2024	Update
4	Võ Thị	Thu	thu@email.com	123	2500000.00	1	May 25, 2024	Update
5	Phạm Văn	Anh	anh@email.com	123	3000000.00	1	May 25, 2024	Update
6	Hoàng Thị	Linh	linh@email.com	123	1700000.00	1	May 25, 2024	Update

- **Giao diện trang chủ khách hàng:**



- **Giao diện thông tin khách hàng:**

H-mart(User hong@email.com)My ProfileMy OrdersAll ProductsCartLogout

My Profile

Nguyễn Thị

hong@email.com

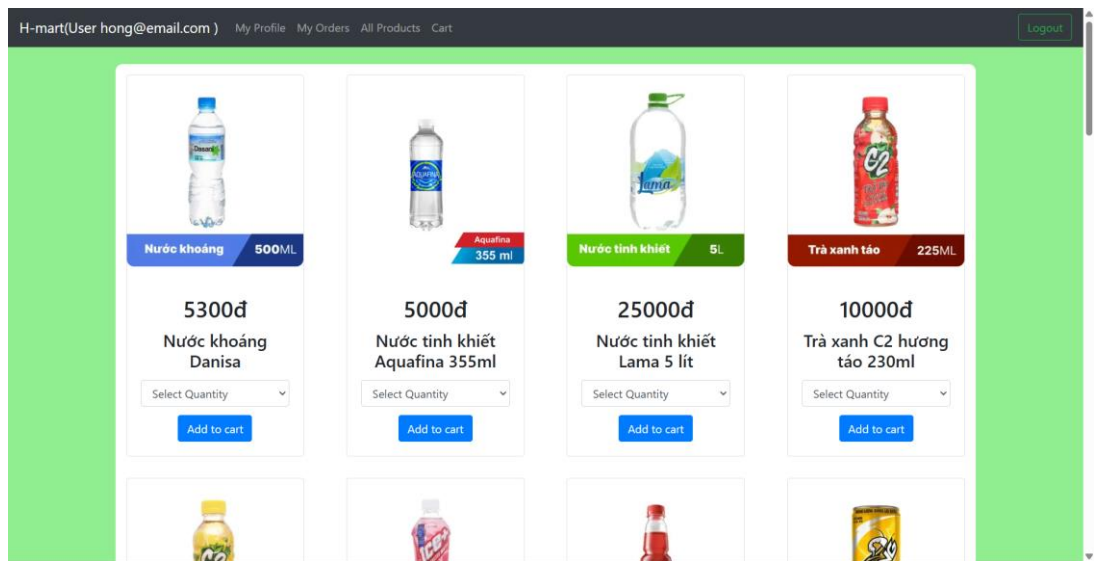
Update

Hồng

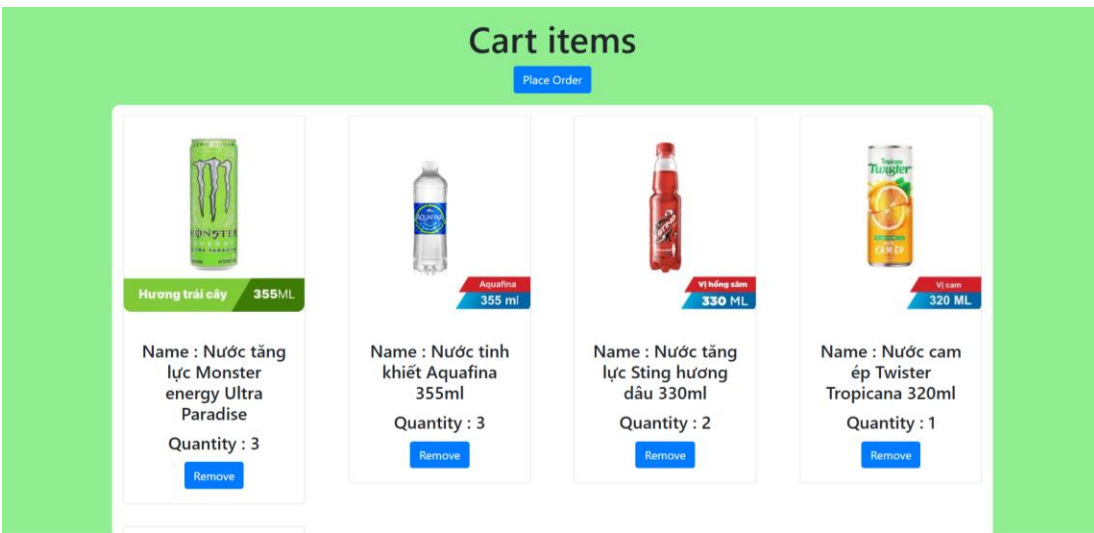
123

Reset

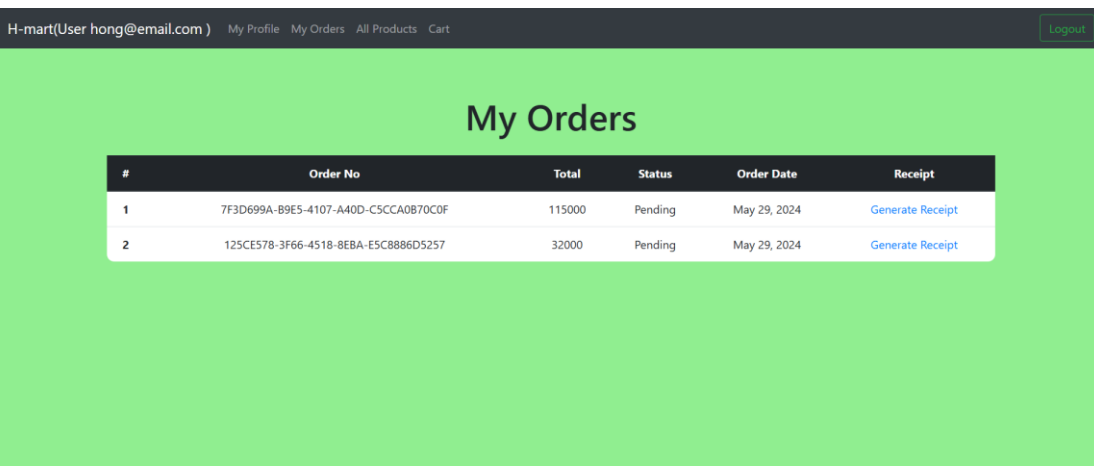
- **Giao diện hiển thị sản phẩm:**



- **Giao diện giỏ hàng:**



- **Giao diện các đơn hàng đã đặt:**

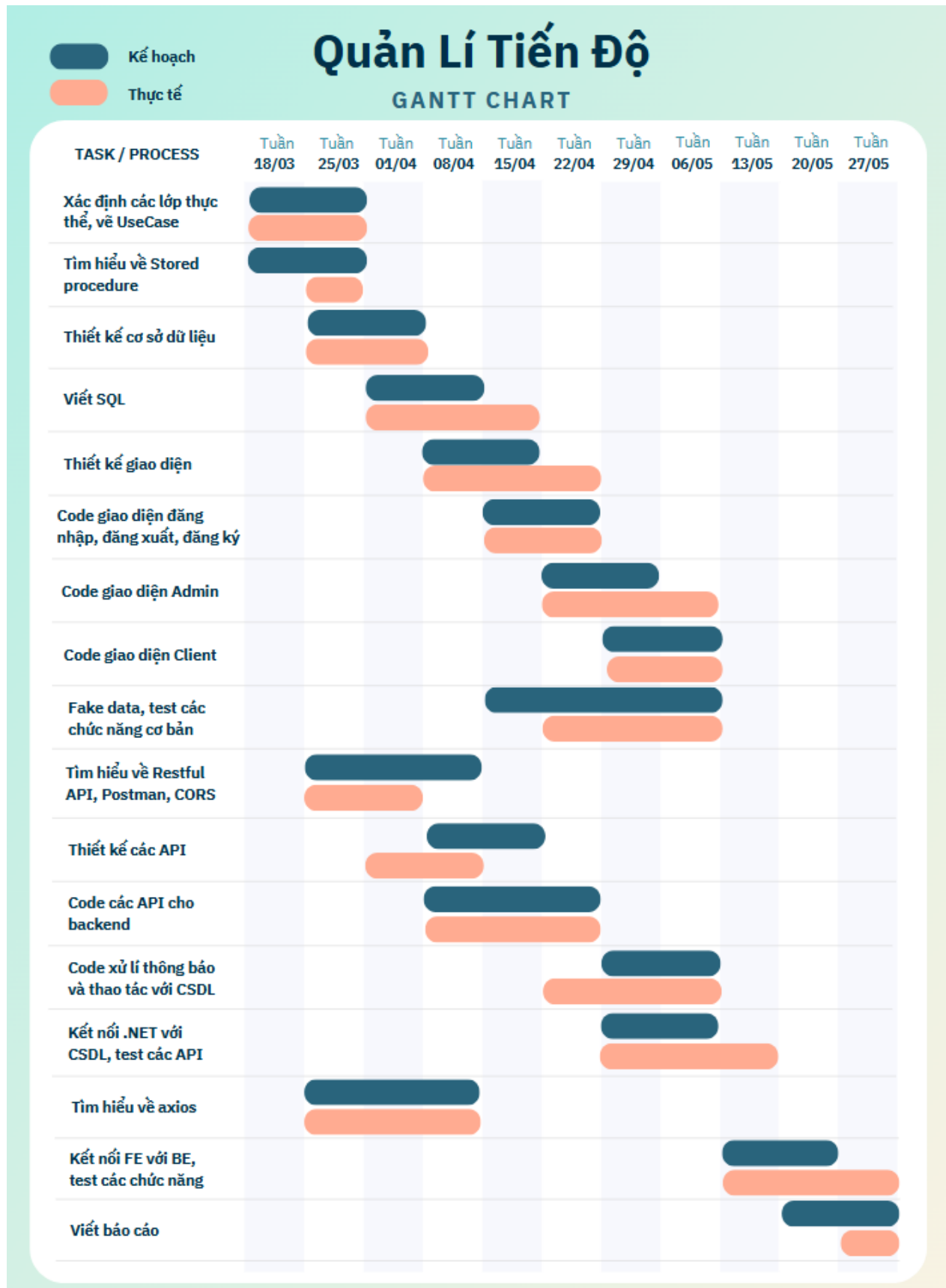


- **Giao diện xuất hóa đơn:**

Download PDF

Order NO : 7F3D699A-B9E5-4107-A40D-C5CCA0B70C0F						
Order Total : 115000						
#	Name	Manufacturer	Unit Price	Quantity	Total Price	Order Date
1	Nước tăng lực Monster energy Ultra Paradise	Monster	23000	3	69000	May 29, 2024
2	Nước tinh khiết Aquafina 355ml	Aquafina	5000	3	15000	May 29, 2024
3	Nước tăng lực Sting hương dâu 330ml	Sting	10500	2	21000	May 29, 2024
4	Nước cam ép Twister Tropicana 320ml	Twister	10000	1	10000	May 29, 2024

3.3. Quản Lý Tiến Độ



3.4. Kết Quả

- **Các tính năng đã hoàn thiện:**

- **Phía admin:**
 1. Đăng nhập
 2. Đăng xuất
 3. Xem danh sách các sản phẩm
 4. Thêm sản phẩm mới
 5. Chỉnh sửa thông tin sản phẩm
 6. Xóa sản phẩm
 7. Xem danh sách khách hàng
 8. Chỉnh sửa thông tin khách hàng
 9. Xem danh sách các đơn hàng
 10. Xem chi tiết các mặt hàng của đơn hàng
 11. Chỉnh sửa trạng thái đơn hàng
- **Phía Client:**
 1. Đăng ký
 2. Đăng nhập
 3. Đăng xuất
 4. Xem thông tin tài khoản
 5. Chỉnh sửa thông tin tài khoản
 6. Xem danh sách sản phẩm
 7. Thêm sản phẩm vào giỏ hàng
 8. Xóa sản phẩm khỏi giỏ hàng
 9. Xem các đơn hàng đã đặt
 10. Xem chi tiết các mặt hàng của đơn hàng
 11. Xuất hóa đơn

- **Các tính năng chưa toàn thiện:**

1. Xem danh sách sản phẩm theo danh mục
2. Tìm kiếm sản phẩm
3. Thanh toán

➔ **Mức độ hoàn thiện dự án: 88%**

Link Source Code: [haumkit/HMart \(github.com\)](https://github.com/haumkit/HMart)

CHƯƠNG 4: KẾT LUẬN

Sau một thời gian tập trung triển khai đề tài, em đã hoàn thành được Xây dựng website bán hàng điện tử H-Mart với giao diện, thao tác sử dụng dễ dàng, thân thiện với người dùng.

Sau khi hoàn thành xong được Website bán hàng điện tử Online, em có thêm những kỹ năng nền tảng để xây dựng được một website bằng ngôn ngữ ReactJS, ASP.Net Core Web API và SQL Server, áp dụng các kiến thức HTML, CSS và Javascript đã học vào việc phát triển ứng dụng web. Ngoài ra, em cũng đã có hiểu biết sâu hơn về quy trình phát triển phần mềm và trải qua các giai đoạn quan trọng như lập kế hoạch, thiết kế, triển khai, kiểm thử và bảo trì. Điều này giúp em có cái nhìn tổng quan về quy trình phát triển phần mềm và hiểu rõ hơn về vai trò của mỗi giai đoạn. Nhờ vậy, kỹ năng tự học và vận dụng các kỹ năng mới của em được nâng cao hơn.

Trong quá trình phát triển dự án, em đã gặp và giải quyết các vấn đề thực tế liên quan đến hiệu năng, bảo mật và trải nghiệm người dùng. Việc giải quyết những vấn đề này đã giúp em nâng cao kỹ năng của mình trong việc xử lý các vấn đề thực tế và tạo ra một sản phẩm chất lượng. Đồng thời, em cũng đã hình thành kỹ năng cải thiện và nâng cao mã nguồn, kỹ năng fix bug và clean code. Việc áp dụng các phương pháp này đã giúp em nâng cao chất lượng và độ tin cậy của mã nguồn trong quá trình phát triển dự án.

Thành quả của việc xây dựng website này cũng giúp em tạo dựng một hồ sơ cá nhân mạnh mẽ. Việc có sản phẩm của riêng mình từ giai đoạn học tại trường là một lợi thế lớn và có thể góp phần nâng cao khả năng cạnh tranh của em khi xin việc trong tương lai. Đồng thời, em đã học cách trình bày báo cáo một cách hiệu quả và khoa học. Việc thực hiện dự án này đã giúp em rèn luyện kỹ năng trình bày thông tin một cách rõ ràng, logic và hấp dẫn. Bên cạnh những kết quả đã đạt được, em nhận thấy còn tồn tại một số hạn chế như tốc độ xử lý chưa cao, giao diện chưa thực sự đẹp mắt,...

Hướng phát triển tiếp theo trong tương lai:

Về hướng phát triển tiếp theo của website, em sẽ tiếp tục hoàn thiện, bổ sung những chức năng chưa được triển khai. Liên tục theo dõi và giải quyết các vấn đề phát sinh từ người dùng trong quá trình sử dụng hệ thống, thường xuyên cập nhật và bảo trì hệ thống, sửa lỗi và cải thiện tính ổn định của website. Tăng cường trải nghiệm người dùng bằng cách tập trung vào thiết kế giao diện thân thiện, dễ sử dụng và hấp dẫn, đồng thời tăng tính tương tác và thu hút người

dùng. Liên tục cập nhật các chính sách bảo mật mới nhất và áp dụng các công nghệ bảo mật tiên tiến để đảm bảo an toàn thông tin của người dùng và dữ liệu.

Về hướng phát triển của bản thân, em sẽ tiếp tục rèn luyện và cải thiện kỹ năng đã có, bao gồm kỹ năng lập trình, quản lý dự án và giao tiếp. Tiếp tục học hỏi và nắm bắt những xu hướng mới trong lĩnh vực công nghệ thông tin. Điều này đảm bảo rằng chúng ta không bị lạc hậu và có khả năng áp dụng những công nghệ mới vào dự án của mình. Trau dồi kỹ năng mềm: phát triển kỹ năng giao tiếp, làm việc nhóm, quản lý thời gian và giải quyết vấn đề. Tìm cơ hội để tham gia vào các dự án thực tế hoặc làm việc với các doanh nghiệp. Cải thiện kỹ năng bảo trì, sửa lỗi và clean code: Tiếp tục nâng cao kỹ năng bảo trì hệ thống, tìm hiểu và áp dụng các phương pháp sửa lỗi hiệu quả. Đồng thời, tuân thủ các quy tắc clean code để tạo ra mã nguồn dễ đọc, dễ bảo trì và hiệu suất cao.

TÀI LIỆU THAM KHẢO

- [1] "HTML Tutorial," W3Schools. [Online]. Từ: <https://www.w3schools.com/html/>.
- [2] "CSS Tutorial," W3Schools. [Online]. Từ: <https://www.w3schools.com/css/>.
- [3] "JavaScript Tutorial," W3Schools. [Online]. Từ: <https://www.w3schools.com/js/>.
- [4] "React Tutorial," Viblo. [Online]. Từ: <https://viblo.asia/p/tim-hieu-reactjs-co-ban-va-cach-su-dung-BYjv4ke0GxpV/>.
- [5] "ReactJS dùng để làm được gì? Ưu nhược điểm của ReactJS," Lotusacademy. [Online]. Từ: <https://lotusacademy.edu.vn/blog/reactjs-dung-de-lam-duoc-gi-uu-nhuoc-diem-cua-reactjs-288/>.
- [6] "RESTful API là gì" Topdev. [Online]. Từ: <https://topdev.vn/blog/restful-api-la-gi/>.
- [7] "Giới thiệu về ASP.NET Web API" Codelean, 16 April 2024. [Online]. Từ: <https://www.codelean.vn/2020/02/gioi-thieu-ve-aspnet-web-api.html/>.
- [8] "Tìm hiểu về asp.net core web api" Memart. [Online]. Từ: <https://memart.vn/tin-tuc/blog/tim-hieu-ve-aspnet-core-web-api-la-gi-va-cach-su-dung-trong-web-development-vi-cb.html/>.
- [9] "What is SQL Server", GeeksforGeeks. [Online]. Từ: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/>.