

**ĐẠI HỌC THÁI NGUYÊN  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

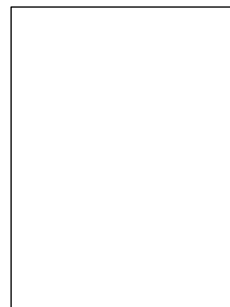
**ĐÀO NGUYỄN KHÁNH**

**XÂY DỰNG ỨNG DỤNG PHÁT TRIỂN KINH DOANH CHO  
CỬA HÀNG MÁY TÍNH HNKEM**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
CHUYÊN NGÀNH KỸ THUẬT PHẦN MỀM**

**THÁI NGUYÊN, NĂM 2024**

ĐẠI HỌC THÁI NGUYÊN  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**ĐỒ ÁN**  
**TỐT NGHIỆP ĐẠI HỌC**  
**CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài:

**XÂY DỰNG ỨNG DỤNG PHÁT TRIỂN KINH DOANH  
CHO CỬA HÀNG MÁY TÍNH HNKEM**

Sinh viên thực hiện: Đào Nguyễn Khánh  
Lớp: KTPM K17B  
Giảng viên hướng dẫn: TS. Trần Quang Quý

**Thái Nguyên, năm 2024**

## **LỜI CAM ĐOAN**

Em xin cam đoan về nội dung đề án tốt nghiệp với tên đề tài “**XÂY DỰNG ỨNG DỤNG PHÁT TRIỂN KINH DOANH CHO CỬA HÀNG MÁY TÍNH HNKEM**” không sao chép nội dung từ các đề án khác, hay các sản phẩm tương tự mà không phải em làm ra. Sản phẩm của đề án do chính bản thân em nghiên cứu và xây dựng nên. Nếu có gì sai em xin chịu mọi hình thức kỉ luật của Trường Đại học Công Nghệ Thông Tin và Truyền Thông – Đại học Thái Nguyên.

Thái Nguyên, 3 tháng 5 năm 2024  
**SINH VIÊN**

**Đào Nguyễn Khánh**

## LỜI CẢM ƠN

Đầu tiên cho em gửi lời biết ơn sâu sắc đến thầy Trần Quang Quý đã định hướng, cung cấp tài liệu và tận tình giúp đỡ, động viên em trong quá trình thực hiện đề tài.

Thầy đã dành nhiều thời gian và tâm huyết theo sát, chỉ bảo em những mặt được và chưa được để em có thể hoàn thành đề tài một cách tốt nhất. Em xin chân thành cảm ơn thầy.

Em cũng xin chân thành cảm ơn các thầy/cô trong trường Đại học Công nghệ thông tin và Truyền thông đã dạy dỗ giúp em có được một nền tảng kiến thức căn bản hỗ trợ cho việc em làm đồ án tốt nghiệp. Mặc dù, bản thân đã thực sự nỗ lực, cố gắng thực hiện đồ án tốt nghiệp nhưng không tránh khỏi nhiều thiếu sót. Do vậy, em rất mong nhận được sự đóng góp quý báu của toàn thể thầy cô và các bạn!

Em xin chân thành cảm ơn!

Thái Nguyên, 3 tháng 5 năm 2024  
**SINH VIÊN**

**Đào Nguyễn Khánh**

# MỤC LỤC

## LỜI MỞ ĐẦU

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Công nghệ Vue.js .....	2
1.2. Công nghệ ASP.NET Core .....	2
1.3. Hệ quản trị cơ sở dữ liệu PostgreSQL .....	4
1.4. MicroServices .....	5
1.5. Docker .....	5
1.6. DNS .....	6
1.7. Hyper V .....	7
1.8. VMWare .....	8
1.9. Proxmox .....	9
1.10. DDNS .....	9

## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Khảo sát quy trình bán hàng .....	11
2.2. Giải pháp xử lý .....	11
2.3. Phân tích tài nguyên .....	12
2.4. Yêu cầu chức năng .....	12
2.5. Mô tả tổng quan ứng dụng .....	13
2.7. Đặc tả các yêu cầu chức năng .....	17
2.8. Mô hình lớp .....	70
2.9. Cơ sở dữ liệu .....	70

## CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM CHƯƠNG TRÌNH

3.1. Triển khai phần Back-end lên Docker .....	72
3.2. Triển khai phần Front-end .....	74
3.3. Giới thiệu giao diện các chức năng .....	75
3.4. Hướng dẫn tạo máy ảo VMWare .....	79
3.5. Hướng dẫn cài đặt Proxmox .....	82
3.6. Setting port trên model .....	87
3.7. Hướng dẫn đặt ip tĩnh .....	89
KẾT LUẬN .....	90
TÀI LIỆU THAM KHẢO .....	91

## LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin, việc có một ứng dụng để quảng bá công ty hay một ứng dụng cá nhân không còn là điều gì xa xỉ nữa. Thông qua ứng dụng, khách hàng có thể lựa chọn sản phẩm mà mình cần một cách nhanh chóng và hiệu quả.

Hiện tại, vấn đề kinh doanh máy tính, laptop được quản lý trên sổ sách, giấy tờ hay trên excel vẫn tốn rất nhiều công sức lẫn thời gian và khách hàng cũng rất khó có thể xem các mặt hàng hiện có và mức giá cụ thể.

Với sự bùng nổ của các cổng thanh toán, dịch vụ thanh toán trực tuyến, hiện nay khách hàng có thể dễ dàng mua thứ mình muốn thông qua Internet.

Xuất phát từ nhu cầu thực tế, các cửa hàng, cá nhân kinh doanh máy tính, laptop đang rất cần một ứng dụng có thể giúp cho khách hàng đặt hàng và thanh toán trực tuyến, đồng thời cũng quản lý được đơn hàng và sản phẩm.

Với lý do đó, được sự hướng dẫn và giúp đỡ của thầy **Trần Quang Quý**, em đã chọn đề tài “**XÂY DỰNG ỨNG DỤNG PHÁT TRIỂN KINH DOANH CHO CỬA HÀNG MÁY TÍNH HNKEM**” làm đề tài cho đề án tốt nghiệp của mình.

Tuy nhiên do thời gian hạn hẹp, đề án khó tránh khỏi thiếu sót, em rất mong nhận được sự thông cảm, những lời góp ý và chỉ bảo tận tình của quý thầy cô và các bạn.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1. Công nghệ Vue.js

### 1.1.1. Vue.js là gì?

- Vue.js là một framework JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng và ứng dụng web đơn trang (SPA - Single Page Applications). Nó được thiết kế để tối giản việc phát triển ứng dụng web, cung cấp một cách tiếp cận linh hoạt và dễ dàng để xây dựng các thành phần giao diện và tương tác trên các trang web.

### 1.1.2. Các tính năng chính của Vue.js

Vue.js có nhiều tính năng hữu ích để xây dựng các ứng dụng web động và tạo các giao diện người dùng (UI) đẹp và dễ sử dụng. Sau đây là một số tính năng chính của Vue.js:

- **Binding dữ liệu (Data Binding):** Vue.js cho phép bạn liên kết dữ liệu của ứng dụng với DOM, tức là khi dữ liệu thay đổi, giao diện người dùng sẽ được cập nhật tự động và ngược lại.
- **Directive:** Vue.js cung cấp các directive (ví dụ: v-if, v-for, v-bind) để thêm các hành vi động lực vào DOM của bạn một cách đơn giản và linh hoạt.
- **Components (Thành phần):** Vue.js cho phép bạn tạo ra các thành phần tái sử dụng, giúp tạo ra cấu trúc ứng dụng có tổ chức và dễ bảo trì.
- **Routing:** Vue.js cung cấp một gói router chính thức để quản lý việc chuyển đổi giữa các trang hoặc các chế độ xem trong ứng dụng SPA.
- **Vuex (Quản lý trạng thái):** Vuex là một thư viện quản lý trạng thái dành cho các ứng dụng Vue.js lớn. Nó giúp quản lý và chia sẻ trạng thái của ứng dụng qua nhiều thành phần khác nhau một cách hiệu quả.
- **Reactivity (Tính phản ứng):** Vue.js tự động theo dõi các thay đổi trong dữ liệu và cập nhật DOM một cách hiệu quả, giúp tối ưu hiệu suất của ứng dụng.

Nhờ vào những tính năng này, Vue.js đã trở thành một trong những framework phổ biến và được sử dụng rộng rãi trong cộng đồng phát triển web.

### 1.1.4. DevExtreme là gì?

- DevExtreme là một tập hợp các thư viện và công cụ cho phát triển ứng dụng web và di động. Nó bao gồm các thành phần giao diện người dùng và công cụ quản lý dữ liệu, và hỗ trợ cho các framework JavaScript như Angular, React, và Vue.js.
- DevExtreme có thể dễ dàng tích hợp với các framework JavaScript khác nhau và có thể dễ dàng sử dụng với các công cụ và thư viện hỗ trợ phát triển.

## 1.2. Công nghệ ASP.NET Core

### 1.2.1. ASP.NET Core là gì?

- ASP.NET Core là một framework phát triển ứng dụng web mã nguồn mở được phát triển bởi Microsoft. Nó là phiên bản tiếp theo của ASP.NET Framework, được thiết kế để hoạt động trên nền tảng nhiều hơn và linh hoạt hơn. Dưới đây là một số điểm chính về ASP.NET Core:

- **Đa nền tảng (Cross-Platform):** ASP.NET Core cho phép bạn xây dựng ứng dụng trên nhiều hệ điều hành khác nhau, bao gồm Windows, Mac và Linux. Điều này giúp bạn phát triển một lần và triển khai trên nhiều môi trường khác nhau.
- **Open Source:** ASP.NET Core là mã nguồn mở, cho phép cộng đồng lập trình viên tham gia đóng góp và cải thiện framework.
- **Tối ưu cho Cloud:** ASP.NET Core được thiết kế để phát triển các ứng dụng web liên quan đến đám mây (cloud), chẳng hạn như web apps, IoT và backend cho mobile. Bạn có thể chạy ứng dụng ASP.NET Core trên .NET Core hoặc phiên bản đầy đủ của .NET Framework.
- **Module hóa và Tối ưu tài nguyên:** ASP.NET Core sử dụng các gói (NuGet packages) để tối ưu hóa ứng dụng. Bạn chỉ bao gồm những packages cần thiết, giúp ứng dụng nhỏ hơn, bảo mật tốt hơn và tối ưu hiệu suất.
- **Hỗ trợ MVC và Web API:** ASP.NET Core bao gồm MVC framework, kết hợp các tính năng của MVC và Web API.

### 1.2.2. Ưu điểm và nhược điểm

- Ưu điểm:

- **Tốc độ nhanh:** ASP.NET Core cung cấp tốc độ xử lý nhanh hơn so với các phiên bản trước của ASP.NET vì nó sử dụng bộ xử lý Kestrel mới.
- **Nền tảng đa nền tảng:** ASP.NET Core có thể chạy trên nhiều nền tảng, bao gồm Windows, macOS và Linux.
- **Mã nguồn mở:** ASP.NET Core là mã nguồn mở, cho phép nhà phát triển có thể tùy chỉnh và tối ưu hóa nền tảng theo nhu cầu của họ.
- **Tương thích với các kiến trúc mới:** ASP.NET Core hỗ trợ các kiến trúc mới như microservices (kiến trúc dịch vụ nhỏ) và cloud computing (điện toán đám mây).
- **Dễ dàng mở rộng:** ASP.NET Core có một hệ thống plugin mạnh mẽ, cho phép nhà phát triển mở rộng nền tảng mà không cần phải thay đổi mã gốc.
- **Tích hợp với các công cụ mã nguồn mở:** ASP.NET Core tích hợp với các công cụ mã nguồn mở như Git, Nginx và Docker.

- Nhược điểm:

- **Tương thích với các phần mềm cũ:** ASP.NET Core có thể không tương thích với một số phần mềm cũ hoặc không được hỗ trợ bởi cộng đồng.
- **Ít cộng đồng:** Mặc dù ASP.NET Core đang ngày càng phổ biến nhưng nó vẫn có một cộng đồng nhỏ hơn so với các framework phổ biến khác, điều này có thể gây khó khăn cho việc tìm ra câu trả lời cho các vấn đề phức tạp.



## 1.3. Hệ quản trị cơ sở dữ liệu PostgreSQL

### 1.3.1. PostgreSQL là gì?

- PostgreSQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở mạnh mẽ và linh hoạt, được phát triển và duy trì bởi cộng đồng rộng lớn. Nó hỗ trợ nhiều hệ điều hành và tuân thủ các tiêu chuẩn SQL, cung cấp tính ACID để đảm bảo tính nhất quán và an toàn của dữ liệu. PostgreSQL linh hoạt trong việc xử lý nhiều loại dữ liệu và cung cấp các tính năng mở rộng như truy vấn phức tạp và trigger. Nó cũng đáng tin cậy với hiệu suất cao và tính ổn định trong môi trường sản xuất, cùng với các tính năng bảo mật như phân quyền và mã hóa dữ liệu.

### 1.3.2. Ưu điểm và nhược điểm

- Ưu điểm:

- Hỗ trợ đa dạng dữ liệu: bao gồm các dạng dữ liệu phức tạp như dữ liệu đồ họa, âm thanh và video.
- Tính năng mạnh mẽ: Hỗ trợ rất nhiều tính năng như quản lý dữ liệu, lưu trữ lịch sử, quản lý bảo mật và quản lý đa ngôn ngữ.
- Tốc độ xử lý tốt: Có thể xử lý lượng dữ liệu lớn với tốc độ xử lý cao.
- Tự động quản lý bản sao lưu và khôi phục dữ liệu: giúp cho việc quản lý dữ liệu trở nên dễ dàng và an toàn hơn.
- Tính linh hoạt: Có thể dễ dàng tùy biến và mở rộng theo nhu cầu của người dùng.

- Nhược điểm:

Khó sử dụng cho người mới: PostgreSQL có một học tập và làm quen khá cao đối với người mới bắt đầu. Cú pháp của các truy vấn và cấu hình có thể phức tạp, đặc biệt đối với những người không có kinh nghiệm với hệ quản trị cơ sở dữ liệu.

Hiệu suất có thể bị ảnh hưởng bởi cấu hình không tối ưu: Để đạt được hiệu suất tốt nhất, PostgreSQL cần được cấu hình đúng cách. Cấu hình không đúng có thể dẫn đến hiệu suất kém hoặc sự cố về bộ nhớ hoặc tài nguyên.

Thời gian sao lưu và khôi phục có thể mất nhiều thời gian: Quá trình sao lưu và khôi phục cơ sở dữ liệu trong PostgreSQL có thể tốn nhiều thời gian, đặc biệt đối với các cơ sở dữ liệu lớn.

Quản lý và cấu hình phức tạp: PostgreSQL yêu cầu một số kiến thức kỹ thuật để quản lý và cấu hình một cách hiệu quả. Việc hiểu rõ về các tùy chọn cấu hình và các tính năng của PostgreSQL có thể là một thách thức cho những người không có kinh nghiệm.

Hạn chế về quản lý tự động và công cụ hỗ trợ: So với một số hệ quản trị cơ sở dữ liệu khác, PostgreSQL có ít công cụ quản lý tự động và thiếu một số tính năng quản trị tiện ích.

## 1.4. MicroServices

### 1.4.1. MicroServices là gì?

- Microservices là một kiểu kiến trúc phần mềm, trong đó một hệ thống phức tạp được chia thành nhiều dịch vụ nhỏ, đơn lẻ và hoạt động riêng biệt nhau. Mỗi dịch vụ này có chức năng và dữ liệu riêng biệt, có thể được phát triển, bảo trì và triển khai riêng biệt.

### 1.4.2. Ưu điểm và nhược điểm

- Ưu điểm:

- Dễ bảo trì phần mềm: các dịch vụ được triển khai độc lập nên có thể dễ dàng bảo trì mà không ảnh hưởng đến các dịch vụ khác.
- Dễ mở rộng phần mềm: vì các dịch vụ được triển khai độc lập nên có thể dễ dàng mở rộng phần mềm.
- Giảm rủi ro: nếu một service gặp sự cố, chỉ có service đó sẽ bị ảnh hưởng, giúp giảm rủi ro toàn bộ hệ thống.
- Giảm sự phụ thuộc giữa các dịch vụ: mỗi dịch vụ riêng biệt có thể hoạt động mà không phụ thuộc vào các dịch vụ khác.

- Nhược điểm:

- Phức tạp hơn so với monolithic (kiến trúc nguyên khối), khó kiểm soát đa dạng hơn, yêu cầu mô hình hệ thống phức tạp hơn.
- Tăng chi phí phát triển và bảo trì, vì cần phải xây dựng và bảo trì nhiều dịch vụ riêng biệt.
- Không phù hợp cho các hệ thống nhỏ
- Khó khăn trong việc quản lý và gỡ lỗi hệ thống
- Phức tạp trong việc tích hợp các dịch vụ với nhau

### 1.4.3. Cách phân tích một yêu cầu phần mềm thành microservices

1. Phân tích yêu cầu của bài toán và tìm ra các tác vụ chính.
2. Xác định rõ ràng rồi phân chia mỗi tác vụ thành một dịch vụ riêng biệt.
3. Xác định các liên kết giữa các dịch vụ và xác định các giao diện API cần thiết để trao đổi dữ liệu giữa các dịch vụ.
4. Xác định các công nghệ phù hợp cho mỗi dịch vụ.
5. Triển khai và vận hành các dịch vụ.

## 1.5. Docker

### 1.5.1. Docker là gì?

- Docker là một công cụ ảo hóa, cho phép bạn đóng gói lại một ứng dụng với tất cả các thành phần cần thiết vào một container (thùng chứa), để có thể chạy một cách độc lập và dễ dàng triển khai trên bất kỳ hệ thống nào có cài đặt Docker. Container giúp đảm