

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP DOANH NGHIỆP**

NGÀNH HỆ THỐNG THÔNG TIN

**ĐỀ TÀI: TÌM HIỂU VỀ CÔNG TY FPT VÀ NỀN TẢNG LẬP TRÌNH .NET FRAMEWORK**

**Cán bộ hướng dẫn :** Đào Việt Linh

**Giảng viên hướng dẫn :** Đỗ Bảo Sơn

**Sinh viên thực hiện :** Lê Minh Đức

**Mã sinh viên :** 64DCMT3001

**Lớp :** 66DCHT21

**Hà Nội, tháng 12 năm 2018**

MỤC LỤC

[**DANH MỤC BẢNG**](#_heading=h.30j0zll)

[**DANH MỤC HÌNH VẼ**](#_heading=h.1fob9te)

[**DANH MỤC TỪ VIẾT TẮT**](#_heading=h.3znysh7)

[**LỜI CẢM ƠN**](#_heading=h.2et92p0)

[**CHƯƠNG 1. GIỚI THIỆU CHUNG** 1](#_heading=h.tyjcwt)

* 1. [**GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP** 1](#_heading=h.3dy6vkm)

[1.1.1. Lịch sử phát triển 2](#_heading=h.4d34og8)

[1.1.2. Một số sản phẩm nổi bật của công ty 5](#_heading=h.17dp8vu)

[**1.2.** **GIỚI THIỆU VỀ ĐỀ TÀI THỰC TẬP** 7](#_heading=h.35nkun2)

[1.2.1. Lý do chọn đề tài 7](#_heading=h.1ksv4uv)

[1.2.2. Mục tiêu 8](#_heading=h.44sinio)

[1.2.3. Nội dung công việc 8](#_heading=h.2jxsxqh)

[**CHƯƠNG 2. GIỚI THIỆU VỀ ASP.NET, FRAMEWORK MVC VÀ SQL SERVER** 9](#_heading=h.3j2qqm3)

[**2.1. GIỚI THIỆU ASP.NET** 9](#_heading=h.1y810tw)

[**2.2. TỔNG QUAN VỀ FRAMEWORK MVC** 12](#_heading=h.1ci93xb)

[2.2.1. Mô hình 2 lớp 12](#_heading=h.3whwml4)

[2.2.2. Mô hình 3 lớp 13](#_heading=h.qsh70q)

[2.2.3. Entity Framework và cách làm việc 13](#_heading=h.1pxezwc)

[2.2.4. Mô hình MVC 15](#_heading=h.2p2csry)

[**2.3. ASP.NET MVC** 17](#_heading=h.3o7alnk)

[2.3.1. Tổng quan về mô hình ASP.NET MVC 17](#_heading=h.23ckvvd)

[2.3.2. Cơ chế hoạt động của ASP.NET MVC 18](#_heading=h.32hioqz)

[2.3.3. Cấu trúc thư mục trong ứng dụng web ASP.NET MVC 18](#_heading=h.41mghml)

[2.3.4. Ưu điểm của ASP.NET MVC 20](#_heading=h.vx1227)

[**2.4. TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER** 20](#_heading=h.3fwokq0)

[2.4.1. Tổng quan về hệ quản trị cơ sở dữ liệu 20](#_heading=h.1v1yuxt)

[2.4.2. Hệ quản trị cơ sở dữ liệu SQL Server 22](#_heading=h.2u6wntf)

[**CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH** 24](#_heading=h.19c6y18)

[**3.1. PHÂN TÍCH HỆ THỐNG** 24](#_heading=h.3tbugp1)

[3.1.1. Tác nhân hệ thống 24](#_heading=h.28h4qwu)

[3.1.2. Biểu đồ Usecase 25](#_heading=h.37m2jsg)

[3.1.3. Biểu đồ lớp 26](#_heading=h.46r0co2)

[Hình 3.2. Biểu đồ lớp 26](#_heading=h.2lwamvv)

[**3.2. THIẾT KẾ CHƯƠNG TRÌNH** 26](#_heading=h.111kx3o)

[3.2.1. Thiết kế cơ sở dữ liệu 26](#_heading=h.3l18frh)

[3.2.2. Thiết kế giao diện 29](#_heading=h.4bvk7pj)

[**CHƯƠNG 4: TỔNG KẾT** 33](#_heading=h.1jlao46)

[**4.1. KẾT QUẢ ĐẠT ĐƯỢC** 33](#_heading=h.43ky6rz)

[**4.2. HẠN CHẾ** 33](#_heading=h.2iq8gzs)

[**4.3. HƯỚNG PHÁT TRIỂN** 33](#_heading=h.xvir7l)

[**TÀI LIỆU THAM KHẢO** 34](#_heading=h.3hv69ve)

# 

# **DANH MỤC BẢNG**

[Bảng 1.1. Lịch sử phát triển](#_heading=h.2s8eyo1)

[Bảng 1.2. Nội dung công việc](#_heading=h.z337ya)

[Bảng 3.1. Tác nhân hệ thống](#_heading=h.nmf14n)

[Bảng 3.2. Bảng User](#_heading=h.206ipza)

[Bảng 3.3. Bảng Category](#_heading=h.4k668n3)

[Bảng 3.4. Bảng Author](#_heading=h.2zbgiuw)

[Bảng 3.5. Bảng Publisher](#_heading=h.1egqt2p)

[Bảng 3.6. Bảng Book](#_heading=h.3ygebqi)

[Bảng 3.7. Bảng Comment](#_heading=h.2dlolyb)

[Bảng 3.8. Bảng Cart](#_heading=h.sqyw64)

[Bảng 3.9. Bảng OrderDetail](#_heading=h.3cqmetx)

[Bảng 3.10. Bảng Order](#_heading=h.1rvwp1q)

# 

# **DANH MỤC HÌNH VẼ**

[Hình 1.1. Logo FPT Software](#_heading=h.1t3h5sf)

[Hình 1.2. Ứng dụng kiểm tra tải đám mây](#_heading=h.3rdcrjn)

[Hình 1.3. Lễ Khai trương hệ thống thông tin chính quyền điện tử](#_heading=h.26in1rg)

[Hình 1.4. FPT.eClinic](#_heading=h.lnxbz9)

[Hình 2.1. .Net Platform](#_heading=h.4i7ojhp)

[Hình 2.2. Trình biên dịch trang web ASP.NET](#_heading=h.2xcytpi)

[Hình 2.3. Mô hình 2 lớp](#_heading=h.2bn6wsx)

[Hình 2.4. Mô hình 3 lớp](#_heading=h.3as4poj)

[Hình 2.5. Mô hình kiến trúc Entity Framework](#_heading=h.49x2ik5)

[Hình 2.6. Luồng đi trong MVC](#_heading=h.147n2zr)

[Hình 2.7. Mô hình ASP.NET MVC](#_heading=h.ihv636)

[Hình 2.8. Cơ chế hoạt động của ASP.NET MVC](#_heading=h.1hmsyys)

[Hình 2.9. Cấu trúc thư mục dự án ASP.NET MVC](#_heading=h.2grqrue)

[Hình 2.10. Các thành phần của hệ quản trị cơ sở dữ liệu](#_heading=h.4f1mdlm)

[Hình 3.3. Giao diện trang chủ](#_heading=h.2r0uhxc)

[Hình 3.4. Giao diện chi tiết sách](#_heading=h.1664s55)

[Hình 3.5. Giao diện giỏ hàng](#_heading=h.3q5sasy)

[Hình 3.6. Giao diện trang đăng nhập](#_heading=h.25b2l0r)

[Hình 3.7. Giao diện trang tạo mới sách](#_heading=h.kgcv8k)

[Hình 3.8. Giao diện trang quản lý tác giả](#_heading=h.34g0dwd)

**DANH MỤC TỪ VIẾT TẮT**

- FPT: Financing Promoting Technology

- ODC: Offshore Development Center

- SQL: Structured Query Language

- RDBMS: Relational Database Management System

- GUI: Graphical user interface

- MVC: Model – View – Controler

- GTVT: Giao Thông Vận Tải

- TNHH: Trách Nhiệm Hữu Hạn

# **LỜI CẢM ƠN**

Trong báo cáo này, em xin gửi lời cảm ơn đến thầy **Đỗ Bảo Sơn** - người đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình nghiên cứu, thực tập và hoàn thành đề tài.

Em xin chân thành cảm ơn các thầy, cô giáo khoa Công nghệ thông tin trường Đại học Công nghệ GTVT – những người đã hướng dẫn em trong suốt quá trình học tập tại trường.

Em xin chân thành cảm ơn công ty TNHH phần mềm FPT (FSOFT), cán bộ hướng dẫn anh Hà Khánh Toàn và chị Đào Việt Linh cùng với các anh/chị trong công ty đã tận tình hướng dẫn em trong suốt quá trình thực tập, tạo điều kiện để em hoàn thành kỳ thực tập này.

Em xin chân thành cảm ơn!

# **CHƯƠNG 1. GIỚI THIỆU CHUNG**

* 1. **GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP**

**Tìm hiểu về công ty**



*Hình 1.1. Logo FPT Software*

Công ty TNHH phần mềm FPT – FPT Software (FSOFT)

Website: <https://www.fpt-software.com>

Email: [fsoft.contact@fsoft.com.vn](mailto:fsoft.contact@fsoft.com.vn)

Số điện thoại: +84 (24) 3 768 9048

Địa chỉ:

Trụ sở chính: Cầu Giấy Building, đường Duy Tân, quận Cầu Giấy, Hà Nội, Việt Nam

Chi nhánh:

* Khu công nghệ cao Hòa Lạc, KM29, Đại lộ Thăng Long, Tân Xá, Thạch Thất, Hà Nội
* Đà Nẵng
* Hồ Chí Minh
* Nhật Bản: Tokyo, Osaka, Nagoya, Fukuoka. Shizuoka
* Mỹ: Texas, Washington, California, Colorado
* Châu Âu: Frankfurt & Essen (Germany), Paris (France), Košice (Slovakia)
* Đông Nam Á: Singapore, Kuala Lumpur (Malaysia), Cebu (Philippines), Yangon (Myanmar)
* Úc: Sydney
* Hàn Quốc: Seoul
* Trung Quốc: Thượng Hải

Công ty TNHH phần mềm FPT thành lập ngày 13 tháng 1 năm 1999, là công ty con trực thuộc tập đoàn FPT- tập đoàn công nghệ và dịch vụ CNTT hàng đầu toàn cầu, có trụ sở tại Việt Nam với doanh thu gần 2 tỷ USD và 32.000 nhân viên. Đạt tiêu chuẩn CMMI Cấp 5 & ISO 27001: 2013, ASPICE LEVEL 3, FPT Software cung cấp các dịch vụ đẳng cấp thế giới trong Nhà máy thông minh, Nền tảng kỹ thuật số, RPA, AI, IoT, Đám mây, AR / VR, Hệ thống nhúng, Dịch vụ được quản lý, Thử nghiệm, hiện đại hóa nền tảng, Ứng dụng kinh doanh, Dịch vụ ứng dụng, BPO và nhiều dịch vụ khác trên toàn cầu từ các trung tâm phân phối trên khắp Hoa Kỳ, Nhật Bản, Châu Âu, Úc, Việt Nam và Châu Á Thái Bình Dương.

Công ty TNHH phần mềm FPT là nhà máy phần mềm:

* Lớn nhất và quy mô nhất và phát triển nhanh nhất tại Việt Nam.
* Hơn 11.000 dự án gia công phần mềm trong suốt 19 năm qua.
* Có mặt rộng rãi trong các thị trường khác nhau trên toàn cầu.
  + 1. **Lịch sử phát triển**

Bảng 1.1. Lịch sử phát triển

| Thời gian | Sự kiện |
| --- | --- |
| 1999 | Thành lập công ty về lĩnh vực gia công phần mềm |
| 2000 | Thành lập Trung tâm phát triển ngoài khơi đầu tiên (ODC) cho một khách hàng ở Vương quốc Anh  Hợp tác với một khách hàng Nhật Bản đầu tiên |
| 2001 | Được IBM chọn làm nhà cung cấp dịch vụ phần mềm độc lập |
| 2004 | Trở thành công ty cổ phần phần mềm FPT  Mở chi nhánh thành phố Hồ Chí Minh và văn phòng đại diện tại Tokyo, Nhật Bản |
| 2005 | Mở chi nhánh tại Đà Nẵng và thành lập công ty phần mềm Nhật Bản |
| 2006 | Trở thành đối tác được chứng nhận vàng của Microsoft |
| 2007 | Thành lập công ty phần mềm FPT Châu Á Thái Bình Dương Pte. Ltd tại Singapore |
| 2008 | Thành lập công ty phần mềm FPT tại Úc, Hoa Kỳ, Malaysia, Paris |
| 2010 | Thành lập nhà máy di chuyển ứng dụng Lotus Notes toàn cầu đầu tiên của Microsoft |
| 2011 | Khánh thành trung tâm R & D Tòa nhà F-Town tại Khu công nghệ cao Sài Gòn, Quận 9 Thành phố Hồ Chí Minh Bắt đầu cung cấp dịch vụ điện toán đám mây như một dịch vụ cốt lõi |
| 2012 | Được quốc tế công nhận và giới thiệu trong Global Services 100 và Software 500  Khởi động dự án phát triển ứng dụng đầu tiên trên Amazon Web Services với một công ty điện tử hàng đầu Nhật Bản. |
| 2013 | Đạt doanh thu 100 triệu USD và đạt hơn 5.000 thành viên  Xếp hạng trong số 50 công ty niêm yết tốt nhất của Việt Nam |
| 2014 | Xếp thứ 57 trong 100 công ty gia công toàn cầu hàng đầu năm 2014 bởi IAOP  Đạt 7000 nhân viên  Mua lại RWE IT Slovakia và mở rộng phạm vi ở châu Âu  Mở thêm 3 văn phòng tại Việt Nam bao gồm F-Ville tại Hà Nội, F-Town 2 tại thành phố Hồ Chí Minh và FPT Cần Thơ tại thành phố Cần Thơ  Hợp tác với các công ty và hiệp hội công nghệ quan trọng: Đối tác tư vấn nâng cao AWS, Đối tác được chứng nhận của Microsoft Gold, Đối tác phát triển ứng dụng di động của SAP, Đối tác tư vấn ủy quyền của Xamarin, Liên minh TV thông minh, Liên minh GENIVI, v.v. |
| 2015 | Giữ vị trí dẫn đầu trong 100 công ty gia công toàn cầu hàng đầu năm 2015 của IAOP  Khai trương thêm 2 trung tâm phát triển tại Yangon và Cebu Được công nhận là đối tác chính của các nhà cung cấp công nghệ lớn trên thế giới cho Cloud và IoT |
| 2016 | Chào mừng nhân viên thứ 10.000  Mở rộng phạm vi với các văn phòng mới tại Chicago ở Mỹ, Fukuoka ở Nhật Bản, Thượng Hải ở Trung Quốc và Seoul ở Hàn Quốc.  Tập trung vào phát triển Số hóa như một dịch vụ theo chiến lược để trở thành người tiên phong trong xu hướng số hóa thông qua phát triển, ứng dụng và chuyển giao. |
| 2017 | Chào mừng nhân viên thứ 13.000  Đạt được CMMi Cấp 5 phiên bản 1.3  Khai trương văn phòng thứ 5 tại Mỹ Được đặt tên là Đối tác công nghệ AWS của năm Trung tâm R & D đã mở tại Okinawa, Nhật Bản. |
| 2018 | Trở thành đối tác hàng đầu của AWS có trụ sở tại ASEAN.  Thành lập văn phòng thứ 6 tại Shizuoka, Nhật Bản.  Trở thành cổ đông lớn của Intellinet. |

* + 1. **Một số sản phẩm nổi bật của công ty**

+ Kiểm tra tải đám mây - Citus Cloud Suite:



*Hình 1.2. Ứng dụng kiểm tra tải đám mây*

* Tăng tốc chuyển đổi, đơn giản hóa hoạt động.
* Chuyển đổi đám mây chuyên nghiệp và bộ công cụ quản lý đám mây.
* Tăng năng suất lên tới 70% thông qua giảm thiểu 60% nỗ lực phát triển.

+ Hệ thống thông tin chính quyền điện tử - FPT.eGov



*Hình 1.3. Lễ Khai trương hệ thống thông tin chính quyền điện tử*

+Hệ thống quản lý phòng khám eClinic: FPT.eClinic có 10 phân hệ chính, quản lý mọi hoạt động của phòng khám theo một quy trình khép kín, từ quản lý thông tin bệnh nhân, bệnh án điện tử, bệnh sử, các công cụ hỗ trợ bác sỹ chẩn đoán bệnh, ra toa thuốc đến các công cụ quản lý hoạt động chung của phòng khám như quản lý kho dược, nhà thuốc, quản lý báo cáo và tài chính.



*Hình 1.4. FPT.eClinic*

* 1. **GIỚI THIỆU VỀ ĐỀ TÀI THỰC TẬP**

Tên đề tài: Tìm hiểu và xây dựng ứng dụng website với ASP.NET MVC

Loại đề tài: nghiên cứu, phát triển

* + 1. **Lý do chọn đề tài**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các tổ chức, công ty và tạo ra những đột phá mạnh mẽ.

Với sự phát triển như vũ bão của mạng Internet và công nghệ thông tin, con người đã thực hiện được nhiều công việc với tốc độ nhanh và chi phí tối ưu hơn so với các phương pháp truyền thống. Chính điều này đã thúc đẩy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện tử trên toàn cầu giúp nâng cao chất lượng cuộc sống con người.

Hiện nay, để góp phần nâng cao chất lượng đời sống của con người, các nhà phát triển đã nghiên cứu và đưa ra các nền tảng công nghệ và phát triển chúng trên các ứng dụng, website,… Đặc biệt, việc phát triển nền tảng ASP.NET MVC đã góp phần xây dựng website ứng dụng hiệu năng cao, bảo mật và an toàn.

Trong hoạt động sản xuất và kinh doanh, thương mại điện tử đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của các doanh nghiệp. Đặc biệt, việc quảng bá, giới thiệu các sản phẩm đến với khách hàng được đẩy mạnh, mang lại doanh thu cao cho các doanh nghiệp, cửa hàng.

Chính vì các lý do trên, em chọn đề tài “Tìm hiểu các bước xây dựng ứng dụng website với ASP.NET MVC” để nghiên cứu và trình bày báo cáo thực tập.

Với sự hướng dẫn của thầy Đỗ Bảo Sơn cùng anh/chị công ty TNHH phần mềm FPT em đã hoàn thành bản báo cáo này.

Tuy nhiên, vì thời gian có hạn cũng như kinh nghiệm còn hạn chế nên không thể tránh khỏi những thiếu sót và chưa hoàn thiện. Em rất mong nhận được sự đóng góp quý giá từ các thầy cô. Em xin chân thành cảm ơn!

* + 1. **Mục tiêu**
* Tìm hiểu nội quy, quy chế hoạt động các lĩnh vực kinh doanh, sản xuất của công ty.
* Tìm hiểu từng bước xây dựng ứng dụng website với ASP.NET
* Nâng cao kỹ năng tin học, kỹ năng làm việc nhóm và làm việc độc lập.
* Xây dựng website bán và giới thiệu sách của của hàng, công ty đến người tiêu dùng thông qua mạng Internet với các chi tiết đầy đủ, giá cả chính xác. Website bán sách gồm các chức năng sau:
* Cho phép quản lý sách ( thêm, sửa, xóa) ; quản lý nhà cung cấp ( thêm, sửa, xóa) ; quản lý đơn hàng; quản lý khách hàng
* Hiển thị danh sách sách theo thể loại, theo tác giả.
* Hiển thị các đầu sách mới, đầu sách bán chạy, đầu sách khuyến mại.
* Hiển thị thông tin khách hàng.
* Xử lý đơn hàng.
  + 1. **Nội dung công việc**

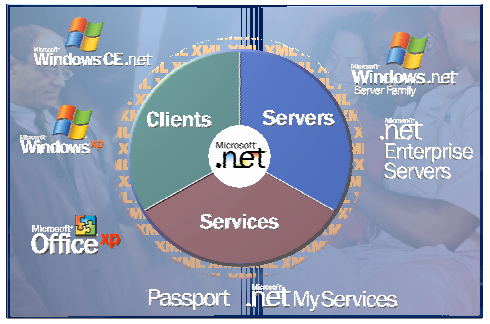
Bảng 1.2. Nội dung công việc

| STT | Thời gian | Nội dung công việc |
| --- | --- | --- |
| 1 | Tuần 1 đến 2 | Tìm hiểu nội quy, quy chế hoạt động của công ty. |
| 2 | Tuần 3 đến tuần 7 | Tìm hiểu từng bước xây dựng ứng dụng website với ASP.NET, SQL Server |
| 3 | Tuần 7 đến tuần 10 | Xây dựng website bán sách bằng ASP.NET MVC |

# **CHƯƠNG 2. GIỚI THIỆU VỀ ASP.NET, FRAMEWORK MVC VÀ SQL SERVER**

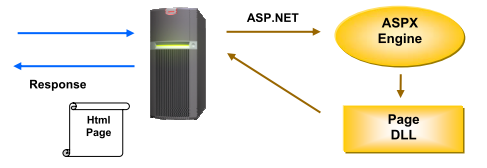
## **2.1. GIỚI THIỆU ASP.NET**

* Tìm hiểu về .Net Platform
* .Net Platform bao gồm .Net Framework và những công cụ được dùng để xây dựng, phát triển ứng dụng và dịch vụ ASP.Net.



*Hình 2.1. .Net Platform*

* Những sản phẩm công nghệ .Net của Microsoft bao gồm: MSN.Net, Office.Net, Visual Studio.Net và Windows Server 2003 được biết đến với tên gọi Windows .Net Server.
* Visual Studio .Net là bộ phần mềm được dùng để xây dựng và phát triển các ứng dụng bao gồm các ngôn ngữ lập trình: C++.Net, Visual Basic.Net, ASP.Net, C# và J#. Tất cả các ngôn ngữ này được xây dựng dựa trên nền .Net Framework, vì thế, bạn sẽ nhận thấy cú pháp cơ bản của những ngôn ngữ này tương tự nhau.
* Tìm hiểu về .Net Framework
* Hệ điều hành:
* Cung cấp các chức năng xây dựng ứng dụng.
* Với vai trò quản lý việc xây dựng và thi hành ứng dụng, .NET Framework cung cấp các lớp đối tượng (Class) để người dùng có thể gọi thi hành các chức năng mà đối tượng đó cung cấp.
* Các chức năng đơn giản như hiển thị một hộp thông báo (Messagebox) sẽ được .NET framework sử dụng các hàm API của Windows. Chức năng phức tạp hơn như sử dụng các COMponent sẽ yêu cầu Windows phải cài đặt Microsoft Transaction Server (MTS) hay các chức năng trên Web cần Windows phải cài đặt Internet Information Server (IIS).
* Common Language Runtime: Là thành phần "kết nối" giữa các phần khác trong .NET Framework với hệ điều hành.
* Common Language Runtime (CLR) giữ vai trò quản lý việc thi hành các ứng dụng viết bằng .NET trên Windows. CLR sẽ thông dịch các lời gọi từ chương trình cho Windows thi hành, đảm bảo ứng dụng không chiếm dụng và sử dụng tràn lan tài nguyên của hệ thống. Nó cũng không cho phép các lệnh "nguy hiểm" được thi hành. Các chức năng này được thực thi bởi các thành phần bên trong CLR như Class loader, Just In Time compiler, Garbage collector, Exception handler, COM marshaller, Security engine, …
* Trong các phiên bản hệ điều hành Windows mới như XP.NET và Windows 2003, CLR được gắn kèm với hệ điều hành. Điều này đảm bảo ứng dụng viết ra trên máy tính của chúng ta sẽ chạy trên máy tính khác mà không cần cài đặt, các bước thực hiện chỉ đơn giản là một lệnh xcopy của DOS!
* Bộ thư viện các lớp đối tượng: .NET Framework là bộ thư viện dành cho các lập trình viên .NET. Với hơn 5000 lớp đối tượng để gọi thực hiện đủ các loại dịch vụ từ hệ điều hành, chúng ta có thể bắt đầu xây dựng ứng dụng bằng Notepad.exe
* ADO.NET và XML: Bộ thư viện này gồm các lớp dùng để xử lý dữ liệu.
* ADO.NET thay thế ADO để trong việc thao tác với các dữ liệu thông thường. Các lớp đối tượng XML được cung cấp để bạn xử lý các dữ liệu theo định dạng mới: XML. Các ví dụ cho bộ thư viện này là SqlDataAdapter, SqlCommand, DataSet, XMLReader, XMLWriter, …
* ASP.NET: Bộ thư viện các lớp đối tượng dùng trong việc xây dựng các ứng dụng Web. ASP.NET.
* Ứng dụng web xây dựng bằng ASP.NET tận dụng được toàn bộ khả năng của .NET Framework.
* Sự xuất hiện của ASP.NET làm cân xứng giữa quá trình xây dựng ứng dụng trên Windows và Web.
* ASP.NET cung cấp một bộ các Server Control để lập trình viên bắt sự kiện và xử lý dữ liệu của ứng dụng như đang làm việc với ứng dụng Windows.
* Ưu điểm của ASP.NET
* ASP.NET cho phép lựa chọn một trong các ngôn ngữ lập trình: Visual Basic.Net, J#, C#.
* Trang ASP.Net được biên dịch trước: ASP.Net biên dịch những trang web động thành tập tin DLL mà Server có thể thi hành nhanh chóng về hiệu quả.

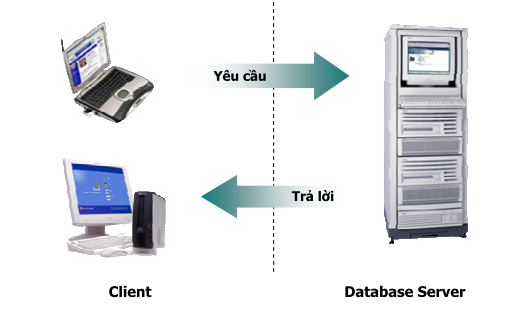


*Hình 2.2. Trình biên dịch trang web ASP.NET*

* ASP.Net hỗ trợ mạnh mẽ các thư viện phong phú và đa dạng của .Net Framework, làm việc với XML, Web service, truy cập cơ sở dữ liệu qua ADO.Net, …
* Tách code riêng, giao diện riêng nên dễ đọc, dễ quản lý và dễ bảo trì.
* Hỗ trợ quản lý trạng thái của các control.
* Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại trình duyệt.
* Hỗ trợ nhiều cơ chế cache.
* Hỗ trợ quản lý ứng dụng ở mức toàn cục.
* Quản lý session trên nhiều Server, không cần cookies.

## **2.2. TỔNG QUAN VỀ FRAMEWORK MVC**

### 2.2.1. Mô hình 2 lớp



*Hình 2.3. Mô hình 2 lớp*

Đây là một dạng mô hình đơn giản, khá phổ biến của một ứng dụng phân tán. Trong mô hình này, việc xử lý dữ liệu được thực hiện trên Database Server, việc nhận và hiển thị dữ liệu được thực hiện ở Client.

* Ưu điểm
* Dữ liệu tập trung giúp đảm bảo dữ liệu được nhất quán.
* Dữ liệu được chia sẻ cho nhiều người dùng.
* Nhược điểm
* Các xử lý tra cứu và cập nhật dữ liệu được thực hiện ở Database Server, việc nhận và hiển thị được thực hiện ở Client gây khó khăn cho quá trình bảo trì và nâng cấp.
* Khối dữ liệu truyền trên mạng lớn => chiếm dụng đường truyền, thêm gánh năng cho Database Server.

### 2.2.2. Mô hình 3 lớp



*Hình 2.4. Mô hình 3 lớp*

Mô hình 2 lớp phần nào đáp ứng được các yêu cầu khắc khe của một ứng dụng phân tán, tuy nhiên, khi khối lượng dữ liệu lớn, ứng dụng đòi hỏi nhiều xử lý phức tạp, số người dùng tăng, mô hình 2 lớp không thể đáp ứng được.

Mô hình 3 lớp sử dụng thêm Application Server giữ nhiệm vụ tương tác giữa Client và Database server, giảm bớt các xử lý trên Database server, tập trung các xử lý nhận và hiển thị dữ liệu tại Application server.

* Ưu điểm
* Hỗ trợ nhiều người dùng.
* Giảm bớt xử lý cho Client -> Không yêu cầu máy tính ở Client có cấu hình mạnh.
* Xử lý nhận và hiển thị dữ liệu tập trung tại Application Server -> dễ quản lý, bảo trì và nâng cấp.
* Xử lý truy cập dữ liệu tập trung tại Database Server.
* Nhược điểm
* Phải sử dụng thêm một Application Server -> Tăng chi phí.

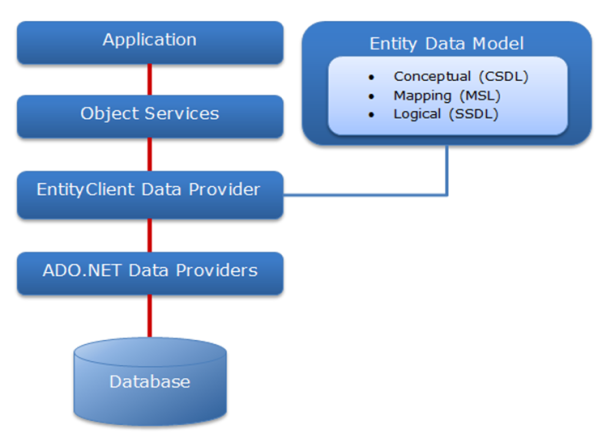
### 2.2.3. Entity Framework và cách làm việc

**a. Tổng quan Entity Framework**

Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web.

EF có vị trí trung gian, đóng vai trò kết nối giữa cơ sở dữ liệu và các thành phần khác của 1 dự án Web khi cần đến.

**b. Mô hình kiến trúc Entity Framework**



*Hình 2.5. Mô hình kiến trúc Entity Framework*

* Application: là tầng chứa giao diện trang Web (HTML, CSS, Javascript, hình ảnh, …) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.
* Object Services (tạm dịch là các dịch vụ đối tượng) là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.
* EntityClient Data Provider: đây là tầng cung cấp các kết nối, diễn dịch các truy vấn thực thể thành truy vấn nguồn dữ liệu (chuyển L2E – LINQ to Entity hay các truy vấn thực thể SQL thành truy vấn SQL), trả về data reader để EF dùng chuyển dữ liệu thực thể thành các đối tượng. Phần này kết nối ADO.NET Data Providers để gửi hoặc lấy dữ liệu từ database. Tầng này hoàn toàn khác với EDM (Entity Data Model) khi thực thi các truy vấn tương tự như cách thực hiện ở ADO.NET Provider. EntityClient Data Provider sử dụng ESQL (Entity SQL), một ngôn ngữ truy vấn độc lập dạng văn bản, tương tự như SQL.
* ADO.NET Data Providers: đây là tầng thấp nhấp để dịch các truy vấn L2E (LINQ to Entity) thông qua cây lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.
* EDM (Entity Data Model) (mô hình dữ liệu thực thể) chứa 3 phần chính: mô hình khái niệm (CSDL – Conceptual schema definition language), mô hình ánh xạ (MSL – mapping specification language) và mô hình lưu trữ (SSDL – store schema definition language). EDM khác với EntityClient Data Provider ở chỗ EDM sử dụng LINQ là ngôn ngữ truy vấn tương tác với database.

### 

### 2.2.4. Mô hình MVC

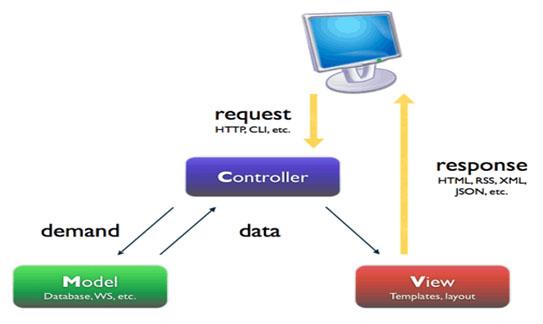
**a. Khái niệm**

MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nói cho dễ hiểu, nó là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

**b. Các thành phần trong mô hình MVC**

* Controller: giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng. Ví dụ Controller nhận request từ url và form để thao tác trực tiếp với Model.
* Model: Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý…
* View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images… Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

**c. Luồng đi trong MVC**



*Hình 2.6. Luồng đi trong MVC*

* Khi có một yêu cầu từ phía client gửi đến server, Bộ phận controller có nhiệm vụ nhận yêu cầu, xử lý yêu cầu đó. Và nếu cần, nó sẽ gọi đến phần model, vốn là bộ phần làm việc với Database..
* Sau khi xử lý xong, toàn bộ kết quả được đẩy về phần View. Tại View, sẽ gen ra mã Html tạo nên giao diện, và trả toàn bộ html về trình duyệt để hiển thị.

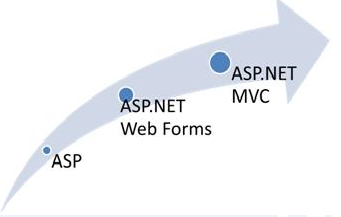
**d. Ưu và nhược điểm của MVC**

* Ưu điểm: Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì..
* Nhược điểm: Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

## **2.3. ASP.NET MVC**

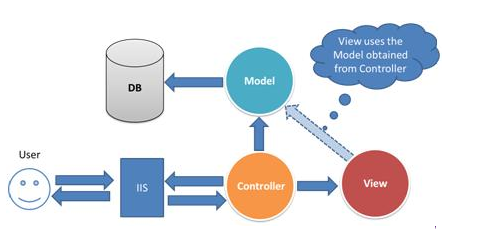
### 2.3.1. Tổng quan về mô hình ASP.NET MVC

ASP.NET MVC là một framework hỗ trợ pattern MVC cho ASP.NET. Framework này là khá nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.



*Hình 2.7. Mô hình ASP.NET MVC*

### 2.3.2. Cơ chế hoạt động của ASP.NET MVC

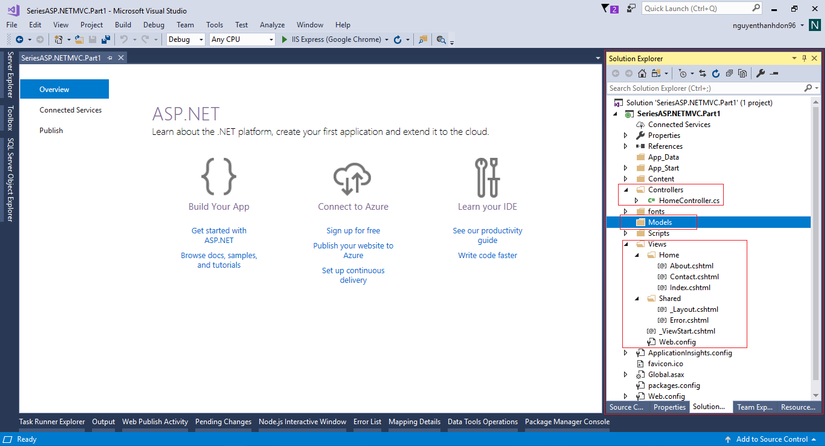


*Hình 2.8. Cơ chế hoạt động của ASP.NET MVC*

* B1: User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser.
* B2: Yêu cầu đó được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ chuyển qua tầng model.
* Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller.
* Controller sẽ giúp dữ liệu được chuyển từ model qua view.
* View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho User thông qua tầng View.

### 2.3.3. Cấu trúc thư mục trong ứng dụng web ASP.NET MVC

* Property: là file chứa các cấu hình chung cho project.
* References: là nơi chứa các DLL (Dynamic Link Library) mà ứng dụng sử dụng.
* App\_Data: là thư mục lưu dữ liệu dạng SQL, các files, file cache,...
* App\_Start: có chứa các files cấu hình.
* BundleConfig: đóng gói các files dạng javascript và stylesheet (các files cùng kiểu)
* FilterConfig: là nơi để cấu hình khi muốn thực hiện logic nào đó trước khi thực hiện hành động. Ví dụ: 1 trang web cho phép comment nhưng người dùng phải đăng nhập, trong trường hợp này trước khi thực hiện truyền đi đoạn comment đó thì phải kiểm tra xem họ đã đăng nhập chưa, thì cái đó được hiểu là 1 dạng của Filter trong ASP.NET MVC.
* RouteConfig: là file cấu hình Router được hiểu là cấu hình URL để gọi đến 1 hành động nào đó.
* Content: là thư mục chứa các files css.
* Controllers: là thư mục chứa các files controller. Trong mỗi controller có thể định nghĩa các actions.
* fonts: là thư mục chứa các fonts chữ cho trang web.
* Models: là thư mục chứa các models.
* Script: là thưc mục chứa các files javascript.
* Views: là thư mục chứa các files razor, ở ASP.NET MVC các files view sẽ là dạng file .cshtml chứ không phải ở dạng thuần html. Mỗi thư mục trong views thì sẽ tương ứng với 1 controller và các files view trong đó là 1 action nằm trong controller đó. Trong thư mục này có thư mục Shared chứa các views dùng chung như \_Layout.cshtml được coi như file khung tương tự như trang Master Page trong Web Forms để cho các views khác kế thừa và muốn các trang khác kế thừa được từ trang này thì chúng ta phải dùng đến file \_ViewStart.cshmlt.
* ApplicationInsights.config: là file cấu hình để theo dõi hiệu suất của ứng dụng web của bạn từ đó bạn có thể biện pháp thiết kế tốt nhất cho ứng dụng.
* favicon.icon: chứa các file dạng file .icon được cấu hình trong file \_Layout.cshtml.
* Global.asax: là file cấu hình chung cho project.
* packages.config: là file quản lý các packages đang sử dụng cho ứng dung và phiên bản sử dụng của các packages đó.
* Web.config: là file cấu hình cho ứng dụng web ở dạng xml. Cho phép thiết lập các thông số về bảo mật, kết nối, debug, ... Là file cấu hình có ảnh hướng đến ứng dụng web trên môi trường Server.



*Hình 2.9. Cấu trúc thư mục dự án ASP.NET MVC*

### 2.3.4. Ưu điểm của ASP.NET MVC

* Do sử dụng mô hình MVC nên trong ASP.Net MVC đã tách biệt được các tầng trong mô hình lập trình web vì vậy giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện.
* Giao diện trong ASP.Net MVC sử dụng công nghệ thiết kế web HTML, CSS nền việc thiết kế giao diện trở nên dễ dàng và giúp cho lập trình viên linh hoạt trong việc thiết kế ASP.Net.
* MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị bị giảm.

## **2.4. TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER**

### 2.4.1. Tổng quan về hệ quản trị cơ sở dữ liệu

**a. Định nghĩa**

Hệ quản trị cơ sở dữ liệu (Datebase Management System - DBMS) là một hệ thống phần mềm cho phép tạo lập cơ sở dữ liệu và điều khiển mọi truy nhập đối với cơ sở dữ liệu đó.

Trên thị trường phần mềm hiện nay ở Việt Nam đã xuất hiện khá nhiều phần mềm hệ quản trị cơ sở dữ liệu như: Microsoft Access, Foxpro, DB2, SQL Server, Oracle, …

Hệ quản trị cơ sở dữ liệu quan hệ (Relation Database Management System- RDBMS) là một hệ quản trị cơ sở dữ liệu theo mô hình quan hệ.

**b. Các khả năng của hệ quản trị cơ sở dữ liệu**

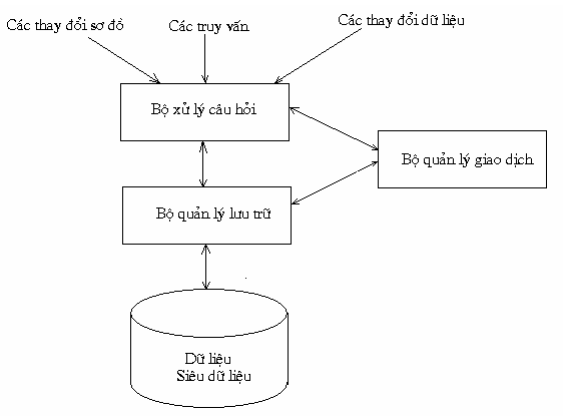
Có hai khả năng chính cho phép phân biệt các hệ quản trị cơ sở dữ liệu với các kiểu hệ thống lập trình khác:

* Khả năng quản lý dữ liệu tồn tại lâu dài: đặc điểm này chỉ ra rằng có một cơ sở dữ liệu tồn tại trong một thời gian dài, nội dung của cơ sở dữ liệu này là các dữ liệu mà hệ quản trị CSDL truy nhập và quản lý.
* Khả năng truy nhập các khối lượng dữ liệu lớn một cách hiệu quả.

Ngoài hai khả năng cơ bản trên, hệ quản trị CSDL còn có các khả năng khác mà có thể thấy trong hầu hết các hệ quản trị CSDL đó là:

* Hỗ trợ ít nhất một mô hình dữ liệu hay một sự trừu tượng toán học mà qua đó người sử dụng có thể quan sát dữ liệu.
* Ðảm bảo tính độc lập dữ liệu hay sự bất biến của chương trình ứng dụng đối với các thay đổi về cấu trúc trong mô hình dữ liệu.
* Hỗ trợ các ngôn ngữ cao cấp nhất định cho phép người sử dụng định nghĩa cấu trúc dữ liệu, truy nhập dữ liệu và thao tác dữ liệu.

**c. Các thành phần của hệ quản trị cơ sở dữ liệu**



*Hình 2.10. Các thành phần của hệ quản trị cơ sở dữ liệu*

### 2.4.2. Hệ quản trị cơ sở dữ liệu SQL Server

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relation Database Management System - RDBMS), cung cấp cách tổ chức dữ liệu bằng cách lưu chúng vào các bảng. Dữ liệu quan hệ được lưu trữ trong các bảng và các quan hệ đó được định nghĩa giữa các bảng với nhau.

Người dùng truy cập dữ liệu trên Server thông qua ứng dụng. Người quản trị CSDL truy cập Server trực tiếp để thực hiện các chức năng cấu hình, quản trị và thực hiện các thao tác bảo trì CSDL.

Ngoài ra, SQL Server là một CSDL có khả năng mở rộng, nghĩa là chúng có thể lưu một lượng lớn dữ liệu và hỗ trợ tính năng cho phép nhiều người dùng truy cập dữ liệu đồng thời.

* Ưu điểm:
* Cấu trúc dữ liệu dễ dùng, không cần hiểu biết sâu về kỹ thuật cài đặt.
* Cải thiện tính độc lập dữ liệu và chương trình.
* Cung cấp ngôn ngữ thao tác phi thủ tục.
* Tối ưu hoá cách truy xuất dữ liệu.
* Tăng tính bảo mật và toàn vẹn dữ liệu.
* Cung cấp các phương pháp thiết kế có hệ thống và mở ra cho nhiều loại ứng dụng (lớn và nhỏ).

## **CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH**

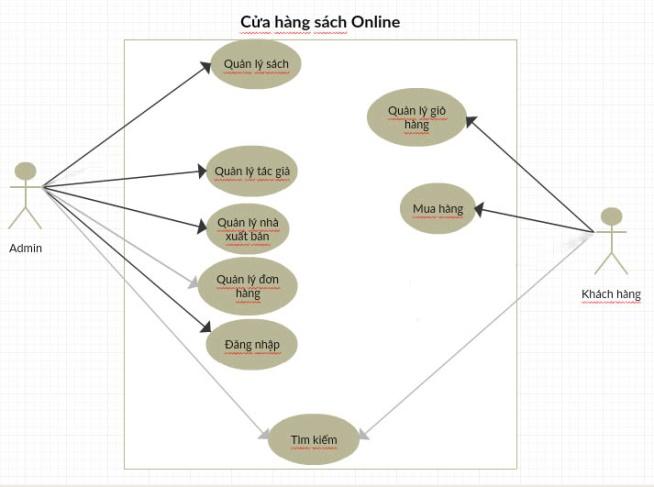
## **3.1. PHÂN TÍCH HỆ THỐNG**

### 3.1.1. Tác nhân hệ thống

Bảng 3.1. Tác nhân hệ thống

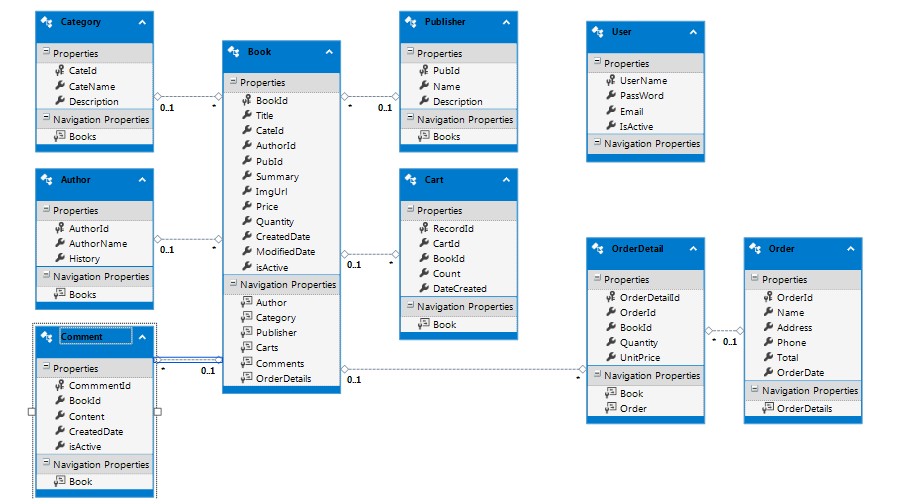
| Tác nhân | Hành động |
| --- | --- |
| Admin | Đăng nhập hệ thống  Thống kê, tìm kiếm sách  Quản lý sách  Quản lý tác giả  Quản lý nhà xuất bản  Quản lý đơn hàng |
| Khách hàng | Tìm kiếm sách  Xem thông tin sách  Quản lý giỏ hàng  Mua hàng |

### 3.1.2. Biểu đồ Usecase



*Hình 3.1. Biểu đồ Usecase*

### 3.1.3. Biểu đồ lớp



*Hình 3.2. Biểu đồ lớp*

## **3.2. THIẾT KẾ CHƯƠNG TRÌNH**

### 3.2.1. Thiết kế cơ sở dữ liệu

Bảng 3.2. Bảng User

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | UserName | Nvarchar(30) | Khóa chính |
| 2 | PassWord | Nvarchar(10) |  |
| 3 | Email | Nvarchar(150) |  |
| 4 | IsActive | Bit |  |

Bảng 3.3. Bảng Category

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | CateId | Int | Khóa chính |
| 2 | CateName | Nvarchar(50) |  |
| 3 | Description | Nvarchar(150) |  |

Bảng 3.4. Bảng Author

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | AuthorId | Int | Khóa chính |
| 2 | AuthorName | Nvarchar(50) |  |
| 3 | History | Nvarchar(150) |  |

Bảng 3.5. Bảng Publisher

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | PubId | Int | Khóa chính |
| 2 | Name | Nvarchar(100) |  |
| 3 | Description | Nvarchar(150) |  |

Bảng 3.6. Bảng Book

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | BookId | Int | Khóa chính |
| 2 | Title | Nvarchar(100) |  |
| 3 | CateId | Int | Khóa ngoại |
| 4 | AuthorId | Int | Khóa ngoại |
| 5 | PubId | Int | Khóa ngoại |
| 6 | Summary | Nvarchar(300) |  |
| 7 | ImgUrl | Nvarchar(350) |  |
| 8 | Quantity | Int |  |
| 9 | CreatedDate | Datetime |  |
| 10 | ModifiedDate | Datetime |  |
| 11 | IsActive | Bit |  |

Bảng 3.7. Bảng Comment

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | CommmentId | Int | Khóa chính |
| 2 | BookId | Int | Khóa ngoại |
| 3 | Content | Nvarchar(150) |  |
| 4 | CreatedDate | Datetime |  |
| 5 | isActive | Bit |  |

Bảng 3.8. Bảng Cart

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | RecordId | Int | Khóa chính |
| 2 | CartId | Int |  |
| 3 | BookId | Int | Khóa ngoại |
| 4 | Count | Int |  |
| 5 | DateCreated | Datetime |  |

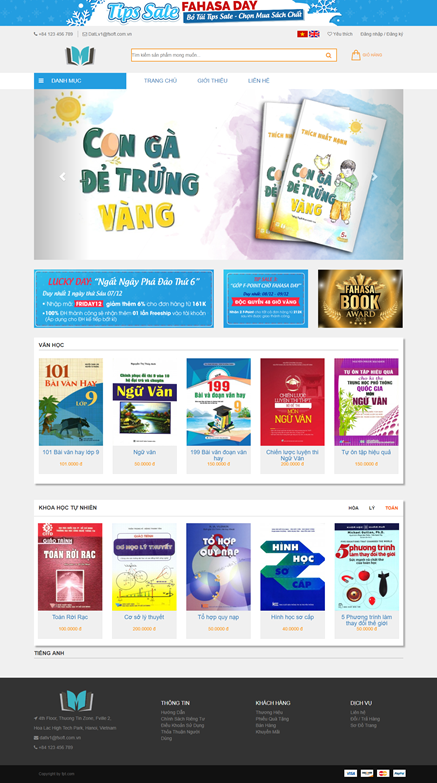
Bảng 3.9. Bảng OrderDetail

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | OrderDetailId | Int | Khóa chính |
| 2 | OrderId | Int | Khóa ngoại |
| 3 | BookId | Int | Khóa ngoại |
| 4 | Quantity | Int |  |
| 5 | UnitPrice | Money |  |

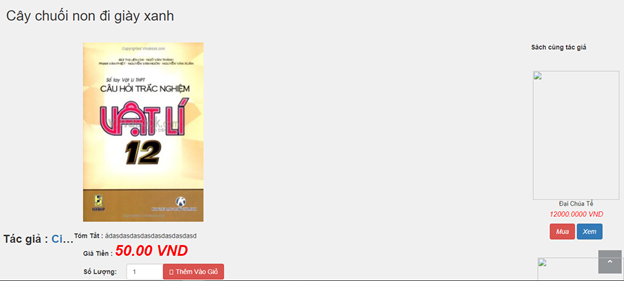
Bảng 3.10. Bảng Order

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
| --- | --- | --- | --- |
| 1 | OrderId | Int | Khóa chính |
| 2 | Name | Nvarchar(50) |  |
| 3 | Address | Nvarchar(150) |  |
| 4 | Phone | Nvarchar(10) |  |
| 5 | Total | Money |  |
| 6 | Orderdate | Datetime |  |

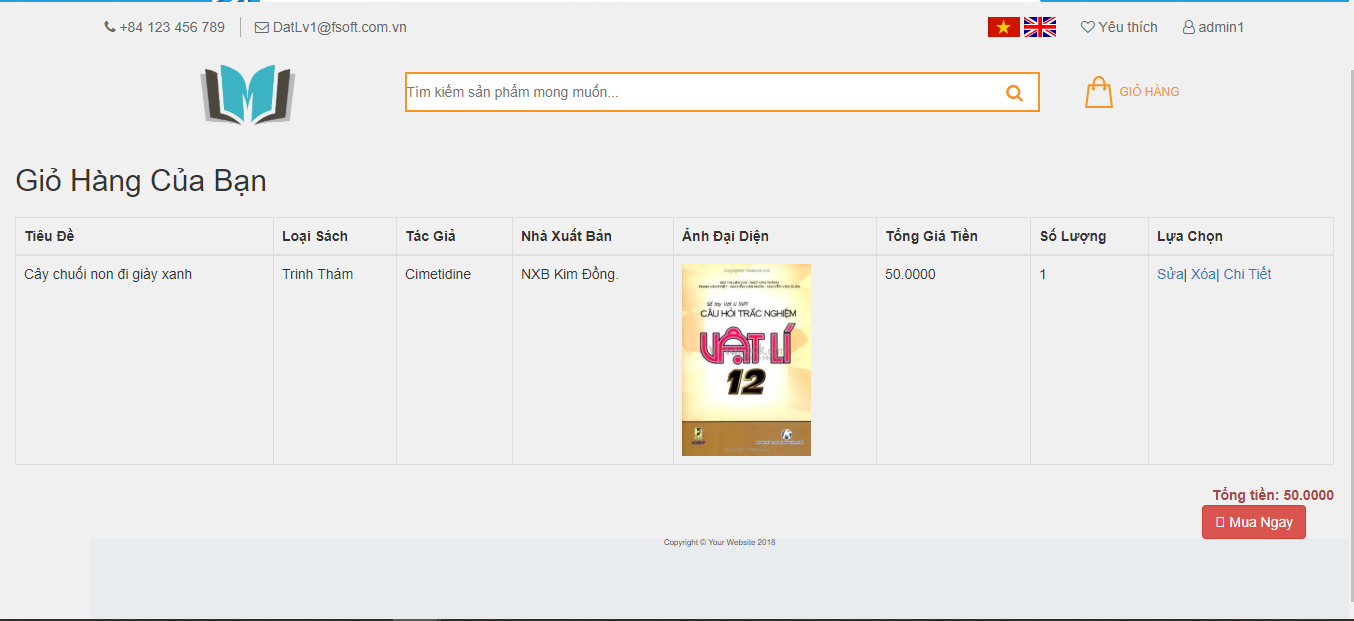
### 3.2.2. Thiết kế giao diện



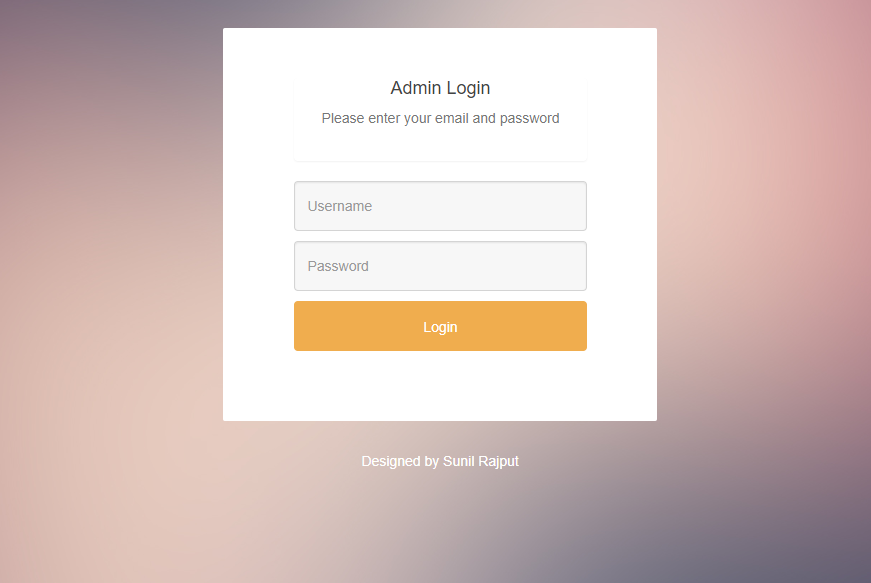
*Hình 3.3. Giao diện trang chủ*



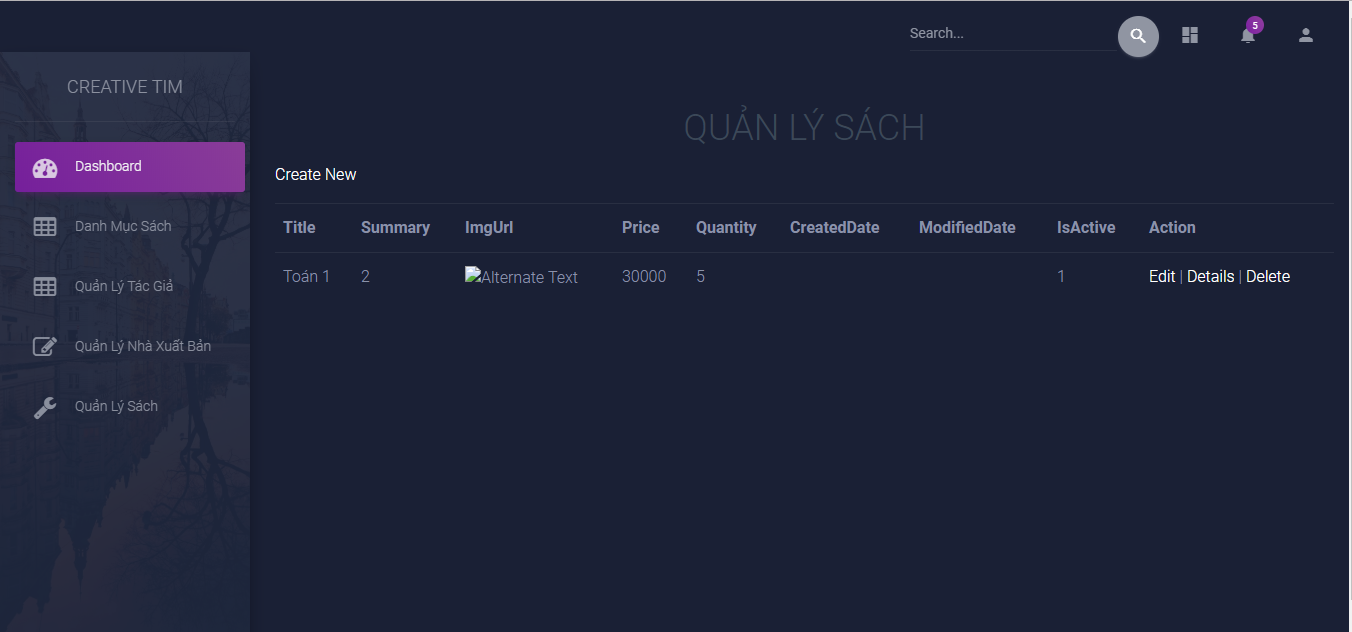
*Hình 3.4. Giao diện chi tiết sách*

**

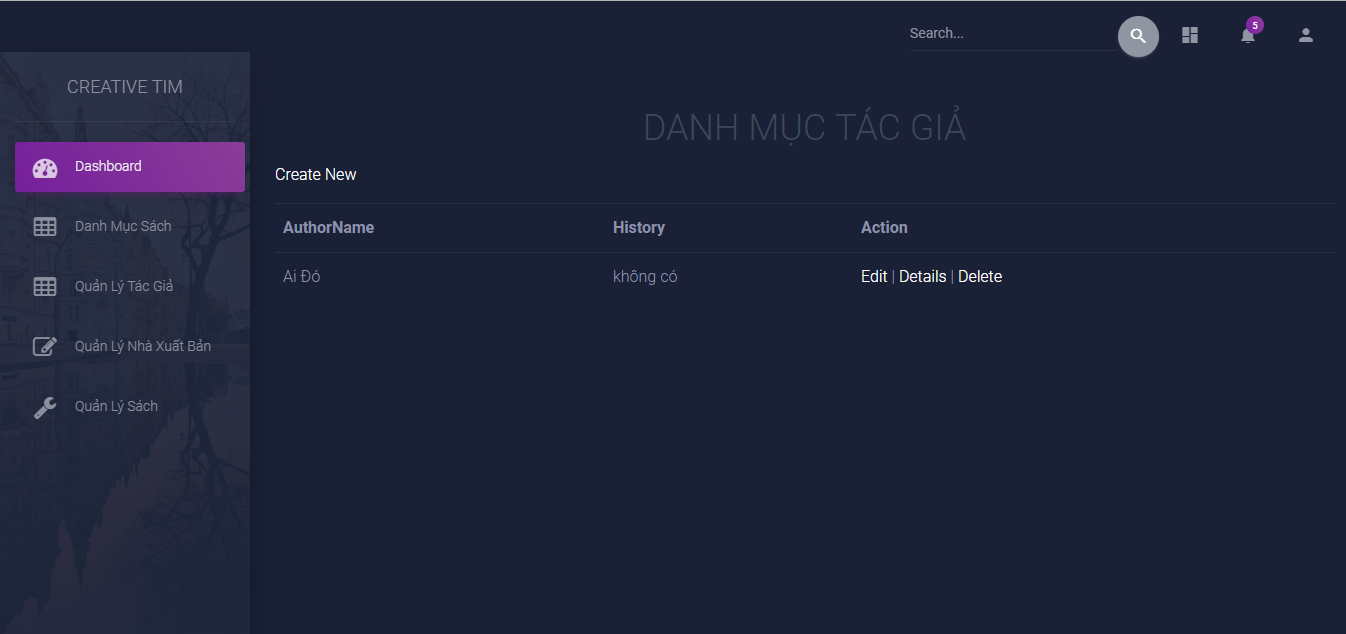
*Hình 3.5. Giao diện giỏ hàng*



*Hình 3.6. Giao diện trang đăng nhập*



*Hình 3.7. Giao diện trang tạo mới sách*



*Hình 3.8. Giao diện trang quản lý tác giả*

# **CHƯƠNG 4: TỔNG KẾT**

## **4.1. KẾT QUẢ ĐẠT ĐƯỢC**

Sau 10 tuần thực tập tại công ty TNHH phần mềm FPT em đã nhận được những kiến thức, kỹ năng như:

* Được tiếp cận với môi trường làm việc hiện đại, chuyên nghiệp tại công ty FPT Software.
* Được tìm hiểu thêm kiến thức về ASP.NET MVC.
* Học được cách tư duy đối với một bài toán.
* Phát triển về kỹ năng làm việc nhóm, kỹ năng làm việc độc lập.
* Sử dụng HTML, CSS và JavaScript, Bootstrap để thiết kế trang Web.
* Sử dụng ASP.NET MVC và SQL Server để xây dựng ứng dụng web động.

## **4.2. HẠN CHẾ**

Do thời gian có hạn cũng như kinh nghiệm còn ít nên không thể tránh khỏi những thiếu sót và hạn chế:

- Sự áp dụng kiến thức mới tìm hiểu được mới chỉ dừng lại ở một bài toán nhỏ, mang tính tổng quát.

- Chưa đáp ứng được toàn bộ yêu cầu bài toán cũng như yêu cầu của cán bộ, giáo viên hướng dẫn.

## **4.3. HƯỚNG PHÁT TRIỂN**

Vì những thiếu sót về kiến thức và kinh nghiệm nên đề tài chưa thể hoàn thiện một cách tốt nhất. Trong tương lai, em mong muốn website bán sách bằng ASP.NET MVC sẽ hoàn thiện về chức năng với giao diện đẹp mắt, thân thiện với người dùng. Một số chức năng cần phát triển thêm như:

* Độ tải trang và hiệu năng trang website nhanh, chính xác.
* Nâng cấp các chức năng giỏ hàng, chức năng thanh toán.
* Hoàn thiện chức năng thống kê, tìm kiếm nâng cao.

# **TÀI LIỆU THAM KHẢO**

* Trang youtube học trực tuyến của Fresher Academy FPT: <https://www.youtube.com/channel/UCydxL-Kg0hxeZ5V_V9eOMTQ>
* Trang học lập trình cơ bản online:

<https://www.w3schools.com>

**Ý kiến đánh giá của Cán bộ hướng dẫn tại đơn vị:**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

Điểm số: ……… Điểm chữ: ………..

Hà Nội, ngày tháng năm 20….

Người đánh giá

(Ký và ghi rõ họ tên)

**Ý kiến đánh giá của Giáo viên hướng dẫn:**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

Điểm số: ……… Điểm chữ: ………..

Hà Nội, ngày tháng năm 20….

Người đánh giá

(Ký và ghi rõ họ tên)