|  |
| --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  KHOA CÔNG NGHỆ THÔNG TIN |
| **NGÀNH CÔNG NGHỆ THÔNG TIN**  **XÂY DỰNG WEBSITE BÁN ĐỒ TENNIS BẰNG ASP.NET MVC**  Sinh viên thực hiện:  Mã sinh viên:  Lớp:  Giảng viên hướng dẫn: |
| HÀ NỘI - 2024 |

**MỤC** **LỤC**

[DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT ii](#_Toc163078115)

[DANH MỤC HÌNH ẢNH iv](#_Toc163078116)

[**LỜI CẢM ƠN** v](#_Toc163078117)

[LỜI NÓI ĐẦU vi](#_Toc163078118)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1](#_Toc163078119)

[**1.1.** **Giới thiệu chung về hệ thống** 1](#_Toc163078120)

[**1.2.** **Đặt vấn đề** 2](#_Toc163078121)

[**1.3.** **Phạm vi đề tài** 2](#_Toc163078122)

[**1.4.** **Phương pháp nghiên cứu/Hướng giải quyết đề tài** 3](#_Toc163078123)

[**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG** 6](#_Toc163078124)

[**2.1. Hệ quản trị cơ sở dữ liệu SQL Server** 6](#_Toc163078126)

[**2.2. Tổng quan về ASP.NET MVC** 8](#_Toc163078127)

[**2.3. Tổng quan về Web API** 10](#_Toc163078128)

[**2.4. Tổng quan kiến trúc của Entity Framework** 15](#_Toc163078129)

[**2.5. Tổng quan về Entity Framework Code First** 16](#_Toc163078130)

[**2.6. HTML, CSS VÀ JAVASCRIPT** 18](#_Toc163078131)

# DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Thuật ngữ/ Từ viết tắt** | **Cụm từ đầy đủ** | **Ý nghĩa** |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| APK | Android Package Kit | Gói ứng dụng Android |
| DOM | Document Object Model | Giao diện lập trình ứng dụng |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| CI | Continuous Integration | Tích hợp liên tục |
| DOM | Document Object Model | Mô hình các đối tượng trong tài liệu HTML |
| Framework | Framework | Framework là phần mềm được phát triển và sử dụng bởi các nhà phát triển để xây dựng  các ứng dụng |
| GUI | Graphical User Interface | Giao diện đồ họa người dùng |
| HTML | HyperText Markup  Language | Ngôn ngữ đánh dấu siêu văn bản |
| IDE | Integrated Development  Environment | Môi trường phát triển tích hợp |
| IE | Internet Explorer | Một trình duyệt web |
| Module | Module | Một thành phần phần mềm hoặc một phần của chương trình |
| QA | Quality Assurance | Những công việc nhằm đảm bảo chất lượng của quy trình phát triển một hệ thống phần  mềm |
| TCP | Transmission Control Protocol | Giao thức điều khiển truyền tin |
| UDP | User Datagram Protocol | Giao thức giao vận không kết nối |
| SSL | Secure Socket Layer | Giao thức cổng an toàn |

# DANH MỤC HÌNH ẢNH

[Hình 2.1: 3](#_Toc106882228)

[Hình 2.2: 11](#_Toc106882228)

**LỜI CẢM ƠN**

Để hoàn thành đồ án, trước hết em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy..............., người đã tận tình hướng dẫn em trong suốt quá trình nghiên cứu vừa qua. Trong thời gian được thầy hướng dẫn, em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần, thái độ làm việc nghiêm túc, hiệu quả. Đây là điều rất cần thiết cho em trong quá trình học tập và làm việc sau này.

Em chân thành cảm ơn các thầy cô trong khoa công nghệ thông tin, …….. đã tâm huyết dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt 4 năm học đại học. Những kiến thức đó không chỉ giúp em hoàn thành đồ án tốt nghiệp này mà còn là thứ hành trang quý báu để em có thể tự tin trên quá trình theo đuổi thành công của mình.

Em cũng xin cảm ơn gia đình, bạn bè, người thân và đặc biệt là tập thể lớp ........ đã hỗ trợ em hết mình trong những năm tháng sinh viên.

Xin kính chúc tất cả mọi người dồi dào sức khỏe và tràn đầy nhiệt huyết trong cuộc sống cũng như công việc của mình.

**Sinh viên thực hiện đồ án**

# LỜI NÓI ĐẦU

Công nghệ thông tin nói chung và internet nói riêng đang là một nhân tố cần thiết trong tất cả các lĩnh vực của đời sống, trong các tất cả các hoạt động…Và trong đấy web là một ứng dụng phổ biến nhất của Internet hiện nay trên thế giới. Web đang ngày một phát triển mạnh mẽ và có nhiều ứng dụng thực tiễn trong cuộc sống.

Internet không có giới hạn về không gian địa lý, các thông tin trên Internet được cập nhật liên tục, nhanh chóng và chính xác. Mà trong đời sống hiện nay các yếu tố này là rất cần thiết, sự nhanh chóng và chính xác là hết sức cần thiết và luôn đươc ưu tiên trong nền kinh tế thị trường.

Người tiêu dùng Việt Nam đã xem Internet là một công cụ để tìm kiếm thông tin về những sản phẩm, mặt hàng, những thông tin về người bán và người mua. Hiện nay trên thực tế có nhiều sản phẩm người này cấn bán và người mua. Vì vậy em cần có một nơi để kết nối người mua và người bán. Do vậy em chọn đề tài của mình là “ **Xây dựng website bán đồ tennis bằng ASP.NET MVC**“

Website bán sản phẩm bán đồ tennis của em không chỉ là nơi liên kết người cần bán và cần mua, mà còn là công cụ hỗ trợ cho người sử dụng tìm hiểu về sản phẩm mình cần và có thể liên hệ, để biết thêm thông tin về sản phẩm mà mình cần mua. Mà còn là nơi để mọi người có thể quảng cáo, đưa các thông tin một cách cụ thể và chính xác nhất. Trong quá trình làm đồ án môn học, do còn hạn chế về thời gian và kinh nghiệm thực tế, em mong nhận được những góp ý chân thành từ thầy cô và các bạn.

**SINH VIÊN THỰC HIỆN ĐỒ ÁN**

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## **Giới thiệu chung về hệ thống**

* Hệ thống bán sản phẩm bán đồ tennis qua mạng chạy trên nền web giúp công việc trao đổi hàng hóa được tiến hành một cách nhanh chóng, tiện lợi và ít tốn kém nhất. Cung cấp những dịch vụ và giải pháp quảng cáo bán sản phẩm bán đồ tennishàng đầu trên internet. Hệ thống Bán sản phẩm bán đồ tennis ra đời nhằm đáp ứng mọi nhu cầu mua bán trao đổi những sản phẩm, mặt hàng và cả dịch vụ cho những tổ chức công ty và cá nhân có nhu cầu.
* Thông tin sẽ được truyền tải tới người sử dụng một cách nhanh chóng và hiệu quả nhất giúp người dùng dễ dàng tìm kiếm thông tin sản phẩm mình cần và giúp quảng bá sản phẩm tới tay người tiêu dùng một cách rộng rãi nhất.
* Website cho phép mọi người đăng kí làm thành viên và đăng tải tin bán sản phẩm bán đồ tennis của mình. Tin bán sản phẩm bán đồ tennis có thể là mua, bán, trao đổi hoặc xem các thông tin về công nghệ … Website sẽ giúp người dùng phát tán tin bán sản phẩm bán đồ tennislên các website khác và giúp quảng bá tin bán sản phẩm bán đồ tennis tới tất cả mọi người ghé thăm website. Tin bán sản phẩm bán đồ tenniscủa người dùng sẽ có cơ hội rất cao để được một số lượng người dùng lớn đang sử dụng các dịch vụ mua bán trực tuyến biết đến và theo dõi.
* Hệ thống website còn được quản lý chuyện nghiệp cùng với hệ thống đánh giá bài viết, đánh giá khách hàng giúp tạo niềm tin vào các thông tin bán sản phẩm bán đồ tennis trên internet.
* Hệ thống gồm 3 phân hệ: Phân hệ khách, phân hệ thành viên (người dùng) thông thường, và phân hệ Administrator.

## **Đặt vấn đề**

* Để quản lý quy trình mua hàng của khách nhân viên kỹ thuật phải ghi chép các biểu mẫu hằng ngày và lưu trữ dưới dạng file thô, khả năng lưu trữ kém và có thể sai sót và gây nhiều bất lợi cho nhân viên kỹ thuật về việc quản lý mua hàng của khách hàng. Ngoài ra khi các bộ phận, nhà máy đặt hàng với số lượng lớn và các mặt hàng khác nhau thì việc giải quyết các nhu cầu mua hàng đó vô cùng phức tạp và mất quá nhiều thời gian nhưng hiệu quả công việc vẫn còn rất thấp.
* Vấn đề đặt ra cho bài toán là làm thế nào xây dựng một website cho phép nhân viên kỹ thuật có thể quản lý quy trình mua hàng của từng khách riêng biệt nhờ vậy mà công ty dễ dàng quản lý được các đơn đặt hàng, thống kê được doanh số bán hàng và lợi nhuận của từng tháng.
* Bên cạnh đó đề tài cần đảm bảo được các yêu cầu sau:
* Tính tiện dụng: Hệ thống phải có tính thân thiện với người dùng, đảm bảo khả năng hỗ trợ tối đa cho người dùng trong việc sử dụng và khai thác hệ thống.
* Tính bảo mật: Hệ thống phải đảm bảo khả năng bảo mật cho thông tin trong quá trình truy xuất thông tin từ phía người dùng. Mọi truy xuất của người dùng phải được kiểm soát cả từ phía trình khách lẫn trình chủ nhầm đảm bảo khả năng vận hành tốt nhất cho hệ thống.
* Tính tùy biến: Hệ thống phải hỗ trợ khả năng tùy biến nhằm giúp cho người quản trị dễ dàng kiểm soát các nguồn thông tin của mình tùy thuộc vào nhu cầu của họ.

## **Phạm vi đề tài**

Đề tài nghiên cứu xoay quanh vấn đề quản lý quy trình mua hàng của khách hàng và hỗ trợ cho nhân viên kỹ thuật để giúp công ty quản lý tốt quy trình mua hàng của của khách hàng được hiệu quả hơn.

* Hệ thống được xây dựng trên nền website và người dùng là khách hàng có nhu cầu mua hàng trực tuyến trên hệ thống website của công ty. Hệ thống sẽ gồm các chức năng chính như sau:
* Quản trị viên (admin) quản lý sản phẩm, đơn hàng, thành viên người dùng
* Khách hàng xem, tìm kiếm và mua sản phẩm
* Khách hàng có thể đăng nhập và xem danh sách đơn hàng đã mua
* Khách hàng có thể gửi liên hệ và xem thông tin công ty
* Có 2 view cho khách hàng và cho quản trị, quản trị yêu cầu đăng nhập còn khách hàng thì mặc định là không.

## **Phương pháp nghiên cứu/Hướng giải quyết đề tài**

* Cấu trúc dự án
* Dự án được xây dựng theo sự kết hợp của mô hình 3 lớp và MVC.
* Mục đích là để có thể sử dụng lại được toàn bộ code ở phần server side vì AngularJS cũng là MVC.
* Một số khái niệm:
* Backend: Toàn bộ code c# xử lý WebAPI, EF, MVC5
* Frontend: Phần code giao diện bên ngoài như AngularJS, Bootstrap,HTML, CSS, JavaScript
* Trang admin: Trang dành cho Admin
* Trang customer: Trang dành cho khách hàng Database -> Repository -> Service -> Web API, Controller-> ASP.NET MVC
* Yêu cầu cài đặt:
* Microsoft SQL Server 2022:

https://www.microsoft.com/en-us/download/details.aspx?id=30438

* Visual Studio 2022 Community:

https://www.visualstudio.com/downloads/download-visual-studio-vs

* Node Package Manage: https://nodejs.org/en/
* Bower: https://bower.io/
* Công nghệ sử dụng
* Truy xuất dữ liệu: Entity Framework
* Resfull API: Web API
* Hiển thị cho khách hàng: ASP.NET MVC thuần
* Phần quản trị:
* Phần chứng thực: ASP.NET Identity
* Định hướng công nghệ:
* Có thể sử dụng ADO.NET nếu cần tốc độ nhưng lại vất vả trong khi codevà cần có kỹ năng quản lý code tốt.
* Có thể dùng Dapper hỗ trợ bạn quản lý kết nối và mapping data.
* Ngoài ra có thể dùng các thư viện khác.
* Tất cả các thư viện này đều phải base trên ADO.NET Provider. Nên ADO.NET tốc độ vẫn là vô địch vì nó nằm gần SQL nhất.
* Dựng cấu trúc solution cho dự án
* Các thành phần của Solution
* Demo.Common: Chứa các lớp tiện ích dùng chung cho dự án
* Demo.Model: Chứa các lớp Domain Entities của dự án
* Shop.Data: Chứa tầng truy cập dữ liệu sủ dụng Entity Framework
* Codefirst
* Demo.Service: Chứa các service xử lý Business logic
* Demo.Web: Project chính dùng để hiển thị giao diện và tương tác
* người dùng.
* Demo.UnitTest: Chứa các class Text sử dụng cho việc Unit Test
* Các package cần cài đặt
* Shop.Common
* Shop.Model
* Entity Framework
* Microsoft.AspNet.Identity.Core
* Microsoft.AspNet.Identity.EntityFramework
* Shop.Data
* Entity Framework
* Shop.Service
* Shop.Web
* Autofac, AutoMapper, Entity Framework...
* Shop.UnitTest
* Moq, Entity Framwork

**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG**



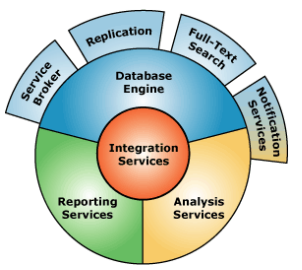
## **2.1.** **Hệ quản trị cơ sở dữ liệu SQL Server**

****

*Hình 2.1: Hệ quản trị CSDL SQL Server*

* Giới Thiệu SQL Server
* SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.
* SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server....
* Các thành cơ bản trong SQL Server

SQL Server được cấu tạo bởi nhiều thành phần như Database Engine, Reporting Services, Notification Services, Integration Services, Full-Text Search Service.... Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng.

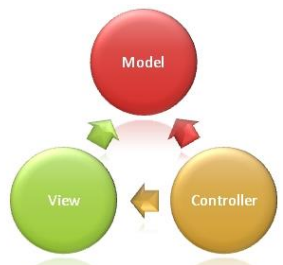


*Hình 2.2: Các thành phần cơ bản trong SQL Server*

* Database Engine: Đây là mộtengine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.
* Integration Services: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,...và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.
* Analysis Services: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.
* Notification Services: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.
* Reporting Services: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.
* Full Text Search Service: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.
* Service Broker: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

## **2.2. Tổng quan về ASP.NET MVC**

* Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.
* MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.
* Nền tảng MVC bao gồm các thành phần dưới đây:



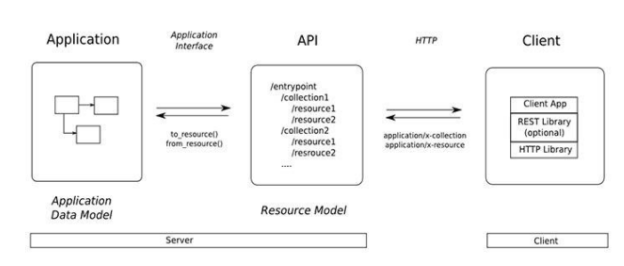
*Hình 2.3: Mô hình ASP.NET MVC*

* Models: Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.
* Views: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.
* Controllers: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.
* Nền tảng ASP.NET MVC mang lại những lợi ích sau:
* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller.
* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến. Để có nhiều thông tin hơn, bạn nên xem phần Front Controller trên web site MSDN.
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
* Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng
* Lợi ích của ứng dụng được xây dựng trên nền tảng Web Forms
* Nó hỗ trợ cách lập trình hướng sự kiện, quản lý trạng thái trên giao thức HTTP, tiện dụng cho việc phát triển các ứng dụng Web phục vụ kinh doanh. Các ứng dụng trên nền tảng Web Forms cung cấp hàng tá các sự kiện được hỗ trợ bởi hàng trăm các server controls.
* Sử dụng mẫu Page Controller. Xem thêm ở mục Page Controller trên MSDN
* Mô hình này sử dụng view state hoặc server-based form, nhờ đó sẽ giúp cho việc quản lý trạng thái các trang web dễ dàng.
* Nó rất phù hợp với các nhóm lập trình viên quy mô nhỏ và các thiết kế, những người muốn tận dụng các thành phần giúp xây dựng ứng dụng một cách nhanh chóng.
* Nói tóm lại, áp dụng Web Forms giúp giảm bớt sự phức tạp trong xây dựng ứng dụng, bởi vì các thành phần (lớp Page, controls,...) được tích hợp chặc chẽ và thường thì giúp bạn viết ít code hơn là áp dụng theo mô hình MVC.

## **2.3. Tổng quan về Web API**

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành.

* Những điểm nổi bật của API:
* Đây là một trong những framework mới sẽ giúp ít cho bạn trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng.
* Mã nguồn mở nên bạn có thể được sử dụng bởi bất kì một client nào hỗ trợ XML, JSON.
* Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma.
* Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.
* Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị băng thông giới hạn như smartphone, tablet.
* Thường nó có định dạng dữ liệu là JSON, XML hoặc một kiểu dữ liệu bất kỳ.
* REST For Web Developers
* Hiện nguyên lý REST và cấu trúc dữ liệu RESTful được biết đến rất rộng rãi trong giới lập trình, nhưng vẫn có nhiều người hay nhầm lẫn. Bản thân REST không phải là một loại công nghệ, mà là phương thức tạo API với nguyên lý tổ chức nhất định. Những nguyên lý này nhằm hướng dẫn lập trình viên tạo môi trường xử lý API request được toàn diện hơn.
* REST là viết tắt của Representational State Transfer. Giải thích đơn giản, REST là một loạt hướng dẫn và dạng cấu trúc dùng cho việc chuyển đổi dữ liệu. Thông thường, REST hay được dùng cho ứng dụng web, nhưng cũng có thể làm việc được với dữ liệu phần mềm.
* Tóm lại, RESTful API là các API theo cấu trúc REST.
* Đặc điểm
* Sự nhất quán trong cả API
* Tồn tại không trang thái (ví dụ, không có server-side session)
* Sử dụng HTTP status code khi cần thiết
* Sử dụng URL endpoint với logical hierarchy



*Hình 2.4: Cấu trúc REST*

* Versioning trong URL chứ không phải trong HTTP header
* REST là một phương thức nhỏ gọn. Nên rất được ưa chuộng cho dữ liệu HTTP. Cũng vì vậy nên REST dần phổ biến trên web, và được xem là lựa chọn “số một” cho phát triển API.
* Có 4 lệnh dùng để truy cập RESTful API:
* GET để truy vấn object
* POST để tạo object mới
* PUT để sửa đổi hoặc thay thế một object
* DELETE để loại bỏ một object
* Mỗi phương thức trên phải được API call thông qua để gửi chỉ thị cho server phải làm gì.
* Sự khác nhau giữa Web Service, WCF, WCF REST, Web API

Hiện nay trong kỷ nguyên hậu PC, smartphone và tablet lên ngôi, nhu cầu xây dựng các ứng dụng mobile trên Windows Phone, iOS, Android đồng bộ dữ liệu đến server theo mô hình điện toán đám mây đang trở nên rất hot. Nếu quyết định xây dựng phía server sử dụng .NET Framework (với phiên bản mới nhất 4.5.3) thì lập trình viên có thể tạo ra các HTTP service bằng rất nhiều công nghệ khác nhau như là: Web Service (ASMX), WCF Service, WCF REST Service,

* Web API Service
* Web Service
* Đây là công nghệ cũ nhất của .NET Framework
* Nó dựa trên SOAP (Simple Object Access protocol) dữ liệu trả về dạng XML
* Chỉ hỗ trợ giao thức HTTP
* Không phải Open Source nhưng có thể sử dụng được với bất cứ client
* nào hỗ trợ XML
* Chỉ có thể host trên IIS

Ưu điểm:

* Code và Test đơn giản

Nhược điểm:

* Chỉ hỗ trợ giao thức SOAP để truyền nhận dữ liệu nên performance
* không cao
* Không thể tạo ra service dạng REST hỗ trợ định dạng dữ liệu JSON
* WCF (.NET 3.0 trở lên)
* Là bản nâng cấp đáng giá của WCF với việc trên .NET 3.5 Microsoft
* bổ sung webHttpBinding để hỗ trợ RESTful service
* Hỗ trợ 2 HTTP verb GET, POST để truyền nhận dữ liệu với 2 thuộc
* tính tương ứng là WebGet và WebInvoke
* Muốn sử dụng các HTTP verb khác như PUT, DELETE cần cấu hình
* thêm trên IIS
* Hỗ trợ các định dạng dữ liệu XML, ATOM, JSON

Ưu điểm:

* Bổ sung hỗ trợ RESTful service với định dạng dữ liệu JSON nhẹ hơn SOAP với dữ liệu XML rất nhiều
* Cho phép cấu hình tham số WebGet qua URI sử dụng UriTemplate

Nhược điểm:

* Chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST
* Cấu hình khó nhớ (cố hữu của WCF)
* Web API (.NET 4 trở lên)
* Đây là một framework mới giúp cho việc xây dựng các HTTP service rất đơn giản và nhanh chóng
* Open Source và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats
* Có thể host trong ứng dụng hoặc trên IIS
* Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như smartphone, tablet
* Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ

Ưu điểm:

* Cấu hình hết sức đơn giản khi so với WCF
* Performance cao
* Hỗ trợ RESTful đầy đủ
* Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test
* Open Source

Nhược điểm:

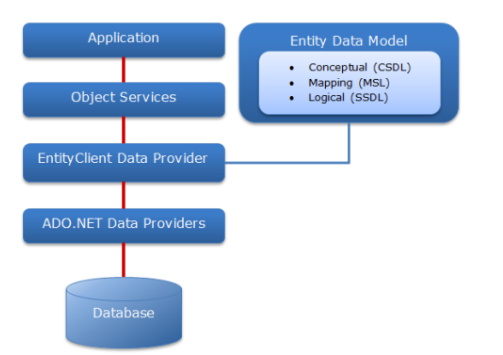
* Còn rất mới nên chưa có nhiều đánh giá về nhược điểm của Web API

- Tại sao dùng Web API:

* Service RESTful hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats
* Service cung cấp dữ liệu cho nhiều client khác nhau với băng thông giới hạn như: browser, mobile, tablet...

## **2.4. Tổng quan kiến trúc của Entity Framework**

ADO.NET Entity Framework là một nền tảng được sử dụng để làm việc với database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, bạn có thể truy vấn, thao tác với database gián tiếp thông qua các đối tượng lập trình.



*Hình 2.5: Kiến trúc của Entity Framework*

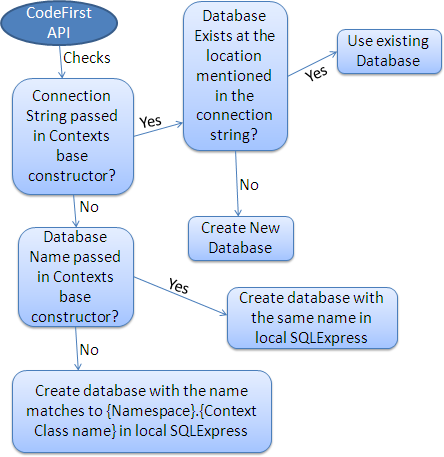
* Kiến trúc của Entity Framework được minh họa như sau:
* Object Services

Đây là các class tự động sinh ra tương ứng với mô hình dữ liệu. Các class này bao gồm:

* ObjectContext đại diện cho một database. ObjectContext có chức năng quản lý các kết nối, định nghĩa mô hình dữ liệu với metadata và thao tác với database. Lớp này cũng có thể thêm vào các phương thức đại diện cho các stored procedure trong database.
* ObjectSet<TEntity> là một một tập hợp các entity. Mỗi đối tượng này tương ứng với một table. Có thể lấy được các đối tượng này thông qua các property tương ứng của ObjectContext.
* EntityObject, ComplexObject là các lớp tương ứng cho một dòng dữ liệu của table trong database. Khác biệt chính giữa hai loại này là ComplexObject không chứa primary key.
* EntityCollection<TEntity> và EntityReference<TEntity>: là các đối tượng thể hiện mối quan hệ (relationship) giữa hai entity class. Mỗi đối tượng này có thể được truy xuất thông qua các property của entity class.

## **2.5. Tổng quan về Entity Framework Code First**

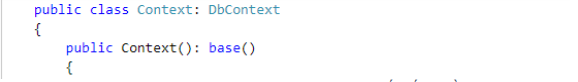
* Entity Framework Code First được giới thiệu từ Entity Framework 4.1. Code First chủ yếu là hữu ích trong Domain Driven Design (DDD). Trong cách tiếp cận Code First, bạn có thể tập trung vào việc thiết kế Domain và bắt đầu tạo ra các lớp theo yêu cầu của Domain của bạn chứ không phải thiết kế cơ sở dữ liệu trước rồi sau đó tạo ra các lớp phù hợp với thiết kế cơ sở dữ liệu đó. Code First API sẽ tạo ra cơ sở dữ liệu dựa trên các lớp thực thể và lớp cấu hình của bạn.
* Vì vậy, đầu tiên bạn bắt đầu viết các lớp thay vì tập trung vào thiết kế cơ sở dữ liệu, sau đó khi bạn chạy ứng dụng, Code First API sẽ tạo ra cơ sở dữ liệu mới hoặc ánh xạ các lớp của bạn vào cơ sở dữ liệu đã tồn tại trước khi chạy ứng dụng của bạn.
* Vì vậy, các công việc cơ bản sẽ là: Viết các lớp → Nhấn F5 để chạy ứng dụng → Code First API tạo ra cơ sở dữ liệu mới hoặc ánh xạ các lớp với cơ sở dữ liệu đã tồn tại → Thêm dữ liệu mặc định vào cơ sở dữ liệu hoặc kiểm tra dữ liệu trong cơ sở dữ liệu → Cuối cùng khởi chạy ứng dụng.
* Hình dưới đây thể hiện luồng khởi tạo database dựa trên đối số truyền vào phương thức khởi tạo của lớp context kế thừa từ lớp DbContext:



*Hình 2.6: Luồng khởi tạo database*

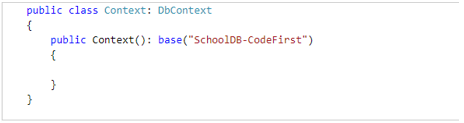
Có thể truyền các đối số vào phương thức khởi tạo của lớp context như sau:

* Không có đối số truyền vào: Nếu bạn không truyền đối số cho phương thức khởi tạo thì Code First API sẽ tạo database của bạn trong SQLEXPRESS với tên được đặt theo quy tắc sau {Namespace}.{Tên\_lớp\_Context}. Ví dụ: database sẽ có tên là SchoolDataLayer.Context nếu chúng ta định nghĩa lớp Context như sau:

**

*Hình 2.7: Định nghĩa lớp Context không có đối số truyền vào*

* Truyền vào tên của database: Nếu bạn truyền tên của database vào đối số của phương thức khởi tạo, Code First API sẽ tạo database có tên như vậy trong SQLEXPRESS. Ví dụ: lớp Context dưới đây sẽ tạo database tên là "SchoolDB-CodeFirst" trong SQLEXPRESS.



*Hình 2.8: Định nghĩa lớp Context truyền vào tên của database*

**2.6. HTML, CSS VÀ JAVASCRIPT**

* **HTML (Hypertext Markup Language)**

HTML là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

* **CSS (Cascading Style Sheets)**

CSS là từ viết tắt của Cascading Style Sheets: công dụng dùng để trang trí trang web và thường được gắn với các ngôn ngữ như HTML, PHP dùng làm nổi bật trang web và hình ảnh trang web. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi câu lệnh CSS sẽ định dạng một phần nhất định của HTML, ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh,…

* **JAVASCRIPT**
* JavaScript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu được tạo ra vào năm 1995 bởi Brendan Eich tại Netscape. Mục đích ban đầu để nâng cao khả năng tương tác của trang web với người sử dụng.
* JavaScript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome,... thậm chí các trình duyệt trên thiết bị di động cũng hỗ trợ.
* JavaScript là một ngôn ngữ lập trình dự trên nguyên mẫu với cú pháp phát triển từ C. JavaScript có khái niệm từ khóa, do đó gần như JavaScript không được mở rộng. Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.
* Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dòng JavaScript để tương tác với DOM bao gồm DHTML, Ajax và SPA.