|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN TỔNG HỢP NGÀNH**

**TRANG WEB BÁN THIẾT BỊ DI ĐỘNG**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn :

Sinh viên thực hiện :

MSSV :

Lớp :

TP. Hồ Chí Minh, 2020

MỤC LỤC

[LỜI NÓI ĐẦU 6](#_Toc77795149)

[LỜI CẢM ƠN 7](#_Toc77795150)

[CHƯƠNG I. TỔNG QUAN 8](#_Toc77795151)

[1.1. Sơ lược về sự ra đời và phát triển của website 8](#_Toc77795152)

[**1.1.1.** **Tốc độ phát triển** 8](#_Toc77795153)

[**1.1.2.** **Thị phần** 8](#_Toc77795154)

[1.2. Nhu cầu bán các thiết bị điện tử 8](#_Toc77795155)

[1.3. Mục đích nghiên cứu 8](#_Toc77795156)

[1.4. Phương pháp nghiên cứu 8](#_Toc77795157)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 9](#_Toc77795158)

[2.1. Các khái niệm và cơ chế hoạt động 9](#_Toc77795159)

[2.1.1. Tổng quan về ASP.NET MVC 9](#_Toc77795160)

[2.1.2. Tổng quan về SQL Server 9](#_Toc77795161)

[2.2. Xác định yêu cầu 9](#_Toc77795162)

[2.2.1. Mô tả bài toán: 9](#_Toc77795163)

[2.2.2. Môi trường thiết kế ứng dụng: 9](#_Toc77795164)

[2.3. Giải quyết vấn đề 10](#_Toc77795165)

[2.3.1. Lược đồ Usecase 10](#_Toc77795166)

[2.3.1.1. Usecase Admin 10](#_Toc77795167)

[2.3.1.2. Usecase Client 10](#_Toc77795168)

[2.3.1.3. Mô tả Actors 10](#_Toc77795169)

[2.3.1.4. Mô tả Usecase 10](#_Toc77795170)

[2.3.1.5. Đặc tả Usecase 10](#_Toc77795171)

[2.3.2. Mô hình tổng thể 10](#_Toc77795172)

[2.3.3. Mô hình phân rã chức năng 10](#_Toc77795173)

[2.3.4. Mô hình thực thể kết hợp 10](#_Toc77795174)

[CHƯƠNG III. KẾT QUẢ THỰC NGHIỆM 11](#_Toc77795175)

[3.1. Giao diện & Chức năng dành cho Người quản trị (Web) 11](#_Toc77795176)

[3.2. Giao diện & Chức năng dành cho Sinh viên (Mobile) 11](#_Toc77795177)

[CHƯƠNG IV. KẾT LUẬN VÀ ĐÁNH GIÁ 11](#_Toc77795178)

[4.1. Kết quả đạt được 11](#_Toc77795179)

[4.2. Đánh giá ứng dụng 11](#_Toc77795180)

[4.2.1. Ưu điểm 11](#_Toc77795181)

[4.2.2. Nhược điểm 11](#_Toc77795182)

[4.2.3. Những thuận lợi và khó khăn 11](#_Toc77795183)

[4.3. Hướng phát triển 11](#_Toc77795184)

[Tài liệu tham khảo 12](#_Toc77795185)

# LỜI NÓI ĐẦU

🙞 ★ 🙜

# LỜI CẢM ƠN

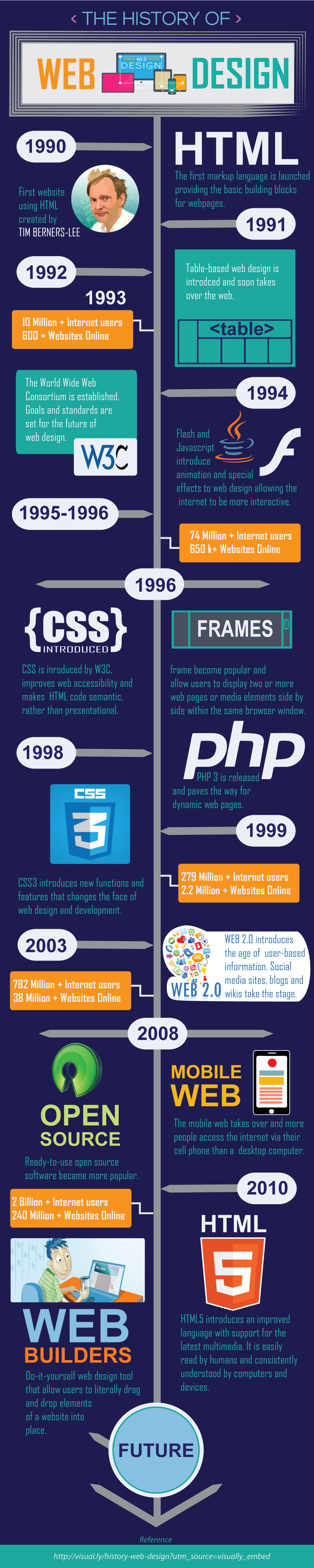
🙞 ★ 🙜

# CHƯƠNG I. TỔNG QUAN

## Sơ lược về sự ra đời và phát triển của website

Website là một tập hợp các trang thông tin có chứa nội dung dạng văn bản, chữ số, âm thanh, hình ảnh, video, v.v... được lưu trữ trên máy chủ (web server) và có thể truy cập từ xa thông qua mạng Internet.

* + 1. **Tốc độ phát triển**Ngày 6-8-1991, Tim Berners Lee công bố chương trình web. Dự án “World Wide Web” được Tim giới thiệu tại newsgroup alt.hypertext.Tháng 12 năm 1990, Tim đã gia nhập nhóm với một nhà vật lý Robert Cailliau, người viết lại đề xuất ban đầu của Tim. Đó là tầm nhìn của họ để kết hợp siêu văn bản với internet để tạo ra các website, nhưng không có ai vào thời điểm đó có thể đánh giá cao ý tưởng này có thể thành công như thế nào. Mặc dù ít quan tâm đến, Tim tiếp tục phát triển ba thành phần chính cho các trang web, HTTP, HTML và trình duyệt web đầu tiên thế giới. Trình duyệt này cũng được gọi là “World Wide Web” và nó cũng tương tự như một trình soạn thảo. Ngày 06 tháng 8 năm 1991, trang web đầu tiên trên thế giới đã được đưa ra. Đây cũng là năm mà HTML được sinh ra và công bố công khai đầu tiên diện mạo của HTML đã được phát hành. Một số các thẻ vẫn còn sử dụng ngày nay, chẳng hạn như thẻ h1-h6, thẻ paragraph và các thẻ anchor.



## Mục đích nghiên cứu

* Củng cố kiến thử về mô ASP.Net và mô hình MVC
* Tạo ra trang web ứng dụng thực tế
* Phương pháp để xây dựng một website
* Biết cách tiếp cận một ngôn ngữ mới
* Hiểu được quy trình tạo môt trang web
* Nắm được những kiến thức cơ bản về Visual Studio, Visual Code, Sql Server

## Phương pháp nghiên cứu

• Thu thập tham khảo các tài liệu trên các trang web.

• Tham khảo ý kiến của giảng viên để hoàn thiện trang web.

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

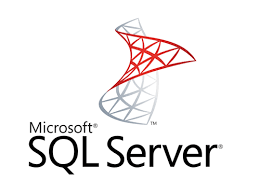
## Các khái niệm và cơ chế hoạt động

### Tổng quan về ASP.NET MVC



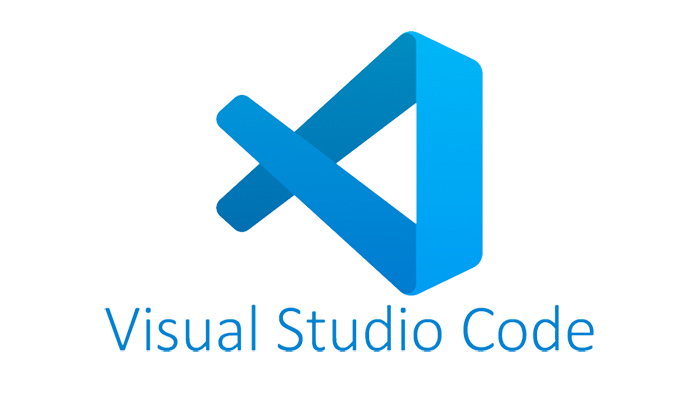
* Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm 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.
* 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.
* Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp lien quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).
* 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.

### Tổng quan về SQL Server



* SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.
* SQL server chính là cụm từ viết tắt của Structure Query Language được sử dụng nhiều trong các lĩnh vực bởi chức năng quản lý dữ liệu. Các ngôn ngữ cấp cao như: Visual C, Oracle, Visual Basic,… đều có trình hỗ trợ là SQL. Những ứng dụng khi chạy phải sử dụng SQL khi người dùng truy cập tới cơ sở dữ liệu thì không cần sử dụng trực tiếp SQL.

### 2.1.3 Tổng quan về Visual Code



* Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

## Xác định yêu cầu

### Mô tả bài toán:

### Môi trường thiết kế ứng dụng:

* Sử dụng ASP.Net MVC để quản lý và lập trình source code trang web
* Sử dụng Sql Server để quản lý cơ sở dữ liệu
* Sử dụng Visual code chỉnh tạo giao diện fontend cho web

## Giải quyết vấn đề

### Lược đồ Usecase

#### Usecase Admin

#### Usecase Client

### Mô hình tổng thể

### Mô hình phân rã chức năng

### Mô hình thực thể kết hợp

# CHƯƠNG III. KẾT QUẢ THỰC NGHIỆM

## Giao diện & Chức năng dành cho Người quản trị (Admin)

## Giao diện & Chức năng dành cho Khách hàng (Client)

# CHƯƠNG IV. KẾT LUẬN VÀ ĐÁNH GIÁ

## Kết quả đạt được

## Đánh giá ứng dụng

### Ưu điểm

### Nhược điểm

### Những thuận lợi và khó khăn

## Hướng phát triển

# Tài liệu tham khảo