**Đồ án I**

**Đề tài: web bán quần áo, giày dép**

**Giới thiệu về công nghệ Vuejs  
Vue js là gì?**Gọi tắt là Vue (phát âm là /vjuː/, giống như view trong tiếng Anh), Vue.js là một framework linh động (nguyên bản tiếng Anh: progressive – tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao hơn nhiều.  
Nếu bạn muốn tìm hiểu thêm về Vue, chúng tôi đã tạo một video clip về những nguyên tắc cốt lõi và một dự án mẫu.  
Nếu bạn là một lập trình viên front-end giàu kinh nghiệm và muốn hiểu hơn về tương quan giữa vue và các thư viện hay framework khác, hãy xem phần So sánh với các framework khác.

**Giới thiệu Asp.net core(Web API)  
ASP.NET Core là gì?**

ASP.NET Core là một tập hợp các thư viện chuẩn như một framework để xây  
dựngứng dụng web. ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET. Nó là một cái tên mới được xây dựng từ đầu. Nó có một sự thay đổi lớn về kiến trúc và kết quả là nó gọn hơn, phân chia module tốt hơn. ASP.NET Core có thể chạy trên cả .NET Core hoặc full .NET Framework.

.Net Core core là gì?

.NET Core là môi trường thực thi. Nó được thiết kế lại hoàn toàn của .NET Framework. Mục tiêu chính của .NET Core là hỗ trợ phát triển ứng dụng đa nền tảng cho ứng dụng .NET. Nó được hỗ trợ trên Windows, Mac OS và Linux. .NET Core là một framework mã nguồn mở được xây dựng và phát triển bởi Microsoft và cộng đồng .NET trên [Github](https://github.com/dotnet/core)

NET Core là một tập con của Full .NET Framwork. WebForms, Windows Forms, WPF không phải là một phần của .NET Core. Nó cũng triển khai đặc điểm của .NET Standard.

.NET Standard là gì?

.NET Standard là một đặc tả chuẩn của .NET API hướng tới hỗ trợ trên tất cả các triển khai của nền tảng .NET. Nó định nghĩa một tập các quy tắc thống nhất cần thiết để hỗ trợ tất cả các ứng dụng trên nền .NET.

Sự khác nhau giữa .NET Core và .NET Framework

.NET Framwork là môi trường cũ hơn và đã tồn tại trên Windows nhiều năm nay. .NET Core hỗ trợ các tính năng trong tập con của .NET Framwork. Các tính năng như WebForms, Windows Forms, WPF chưa được đưa lên .NET Core. .NET Framwork chỉ chạy trên Windows trong khi .NET Core có thể chạy trên bất cứ nền tảng nào.

ASP.NET Core

Từ Core trong ASP.NET Core rất dễ nhầm. Tên đó gợi nhớ việc ứng dụng ASP.NET Core chỉ có thể được build trên .NET Core, điều đó là sai.Ứng dụng ASP.NET Core có thể được phát triển sử dụng .NET Core hoặc .NET Framework. Ứng dụng ASP.NET Core được xây dựng sử dụng .NET Core có thể chạy trên bất cứ hệ điều hành nào trong khi nếu được xây dựng trên .NET Framework chỉ có thể chạy trên Windows.

Lựa chọn công nghệ: Các giải pháp công nghệ được lựa chọn để thực hiện bài toán: Hệ điều hành (nếu có đặc thù), hạ tầng mạng, ngôn ngữ lập trình, Hệ quản trị CSDL, các công cụ hỗ trợ lập trình (front-end/back-end) và quản lý tài nguyên, giao tiếp nhóm….

Các đặc tính quan trọng của ASP.NET Core

Bạn có thể xây dựng và chạy ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux (mã nguồn mở và cộng đồng phát triển)

ASP.NET Core hợp nhất ASP.NET MVC và ASP.NET Web API.

Có thể host trên IIS hoặc tự host.

Có sẵn Dependency Injection.

Dễ dàng tích hợp với các framework frontend như Angular, Knockout...

Hỗ trợ cấu hình cho nhiều môi trường.

Cơ chế HTTP Request pipeline mới.

Hỗ trợ quản lý phiên bản

Dùng chung toàn bộ Nuget Package.

Các nhánh của ASP.NET

Có hai nhánh của ASP.NET cho đến hiện tại

ASP.NET Đây là phiên bản hiện tại của ASP.NET và nó cần .NET Framwork để chạy

ASP.NET Core. Là cách mới để xây dựng ứng dụng web. Nó có thể chạy cả trên .NET Framework và .NET Core.

Web API là gì?

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.

Những điểm nổi bật của Web API Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Khả năng tích hợp linh động API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

Cập nhật thông tin thời gian thực API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Có tiêu chuẩn chung dễ sử dụng bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng.Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Web API hoạt động như thế nào?

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động , dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu…

Ưu và nhược điểm của Web API

Mỗi một ứng dụng bất kỳ đều có những ưu nhược điểm riêng, hỗ trợ tốt cho các ứng dụng. Vì vậy mà web API cũng không ngoại lệ:

Ưu điểm

· Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.

· Linh hoạt với các định dạng dữ liệu khi trả về client: Json, XML hay định dạng khác.

· Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.

· Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, Json.

· Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

· Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

*Nhược điểm*

· Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST

· Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt

· Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành

· Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

### Library Dapper trong C#

Dapper là một Micro ORM, (micro để phân biệt với các ORM được xây xây dựng bới các hãng lớn và có nhiều chức năng) được phát triển bới các developer của StackExchange (nếu các bạn chưa rõ thì họ sở hữu Stack Overflow) dưới dạng phần mềm mã nguồn mở. Họ thiết kế sao cho nó có hiệu năng cao được xác nhận bới Sam Saffron Là một trong những yêu cầu cơ bản bới phần mềm truy cập cơ sở dữ liệu quan hệ nếu chúng ta sử dụng kiến trúc 3 tầng với Data Access Layer (DAL) để lấy dữ liệu từ database và thực hiện các thao tác CRUD (create, read, update, delete) nếu chúng ra sử dụng Entity Framework (EF) của Microsoft và sau đó sử dụng DBContext để tạo kết nối và lấy dữ liệu từ database giờ chúng ta sẽ làm quen với một cách đơn giản hơn để truy cập database sử dụng open source orm được gọi là Dapper

Lý do sử dụng dapper ?

Tốc độ và hiệu suất nhanh chóng. Ít dòng mã hơn. Bản đồ đối tượng.

Ràng buộc đối tượng tĩnh.Ràng buộc đối tượng động.

Xử lý dễ dàng truy vấn SQL. Xử lý dễ dàng các thủ tục đã lưu trữ.

Hoạt động trực tiếp đến lớp IDBConnection cung cấp sự mượt mà và chạy truy vấn trực tiếp đến cơ sở dữ liệu thay vì truyền dữ liệu bằng các đối tượng khác nhau như chúng ta làm trong EF và ADO.NET.

Hỗ trợ nhiều truy vấn. Hỗ trợ cho thủ tục được lưu trữ.Chức năng chèn dữ liệu hàng loạt.Dapper cũng cho phép tìm nạp nhiều dữ liệu dựa trên nhiều đầu vào.

### Kiến trúc sử dụng