**Mã sinh viên: 175A071069**

**Họ và tên: Nguyễn Quang Tú**

**BÀI LUẬN SỐ 01: NHỮNG KIẾN THỨC CHUNG VỀ**

**PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN WEB**

**1. Networks**

**What is the Web?**

\_ Web là một mạng lưới trải dài trên toàn cầu kết nốivô số thiết bị và cho phép chúng giao tiếp với nhau.

**How Does It Work?**

\_Nhiệm vụ căn bản của mạng Internet là tạo phương tiện cho thông tin điện tử di chuyển từ nơi phát xuất tới nơi đến, theo một tuyến đường phù hợp với một hình thức chuyên chở thích ứng.

How Data Finds its Way?

\_ Các trang web trên Internet được lưu trữ trên các thiết bị được gọi là máy chủ và khi chúng ta tương tác với một trang web trên Internet, thực chất là trao đổi dữ liệu với máy chủ mà trang web được lưu trữ trên.

\_Internet hoạt động thông qua mạng định tuyến gói theo ***Giao thức Internet (IP), Giao thức điều khiển vận chuyển (TCP)*** và các giao thức khác.

**Life of a Packet**

\_ Dữ liệu được gửi qua Internet được gọi là mesage. Trước khi message được gửi ,đầu tiên nó được chia thành nhiều phần được gọi là packets. Các gói này được gửi độc lập với nhau. Kích thước tối đa thường từ 1000 đến 3000 ký tự. Giao thức internet chỉ định cách các message nên được đóng gói.

\_ Internet được tạo thành từ mạng lưới lớn các máy tính chuyên dụng gọi là bộ định tuyến. Mỗi công việc của bộ định tuyến là biết cách di chuyển từ các gói nguồn đến đích. một gói sẽ di chuyển qua nhiều bộ đinh tuyến trong suốt quá trình của nó.

**2. Databases**

Databases in Web Development

\_Nếu trang web của chúng ta cần cập nhật số lượng lớn thông tin qua web, chúng ta sẽ cần một cơ sở dữ liệu để lưu trữ thông tin của mình. Có nhiều hệ thống cơ sở dữ liệu khác nhau để lưu trữ web. Phổ biến nhất là MySQL/MariaDB, SQL Server, Oracle và MS Access.

SQL Databases

\_SQL là ngôn ngữ để truy cập cơ sở dữ liệu. Nếu chúng ta muốn trang web của bạn có thể lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu, máy chủ web của chúng ta sẽ có quyền truy cập vào hệ thống cơ sở dữ liệu sử dụng ngôn ngữ SQL.

NoSQL Databases

\_Cơ sở dữ liệu NoSQL là Cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn.

\_ Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

Web Caching

\_ Web cache (hay HTTP cache) là một ứng dụng công nghệ thông tin cho bộ nhớ tạm thời (caching) của các dữ liệu từ trang Web, ví dụ các trang HTML và các hình ảnh nhằm mục đích giảm lag khi server hoạt động. Một hệ thống bộ nhớ lưu trữ nhiều dữ liệu mà nó cho phép; các yêu cầu xuất hiện thêm như cần lưu trữ thêm sẽ được thực hiện nếu hệ thống này cho phép.

**3. The Front End**

**The Server-side and The Client-side**

\_ Phía Khách hàng (Client – side) : Mã nguồn của một trang web có thể được chia thành hai phần: phía máy kháchvà phía máy chủ. Mã ở phía 'máy khách' chạy trên máy tính của khách hàngtrình duyệt và xử lý trang web trông như thế nào, cách trang web yêu cầu dữ liệu từmáy chủ và cách nó tương tác với bộ nhớ tạm thời và cục bộ trên máy kháchmáy móc. Điều này bao gồm nhưng không giới hạn ở việc chọn và tạo kiểu giao diện người dùngcác thành phần, tạo bố cục, điều hướng, xác thực biểu mẫu và bộ nhớ cachesự điều khiển. Các ngôn ngữ phía máy khách bao gồm HTML , CSS và JavaScript

\_ Phía Máy chủ (Sever – side ) : Lập trình phía máy chủ được sử dụng để cung cấp thông tin được truy vấn từ trang web cho khách hàng. Lập trình phía máy chủ bao gồm: *Cung cấp và lưu trữ thông tin một cách hiệu quả, Cung cấp trải nghiệm tùy chỉnh cho mỗi người dùng, Kiểm soát quyền truy cập vào nội dung, Lưu trữ thông tin về phiên / trạng thái, Thông báo và giao tiếp, Phân tích dữ liệu*

**HTML**

\_ HTML là chữ viết tắt của Hypertext Markup Language. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes,…

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web

**CSS**

\_ CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

**JavaScript**

\_ JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web:

HTML: Giúp bạn thêm nội dung cho trang web.

CSS: Định dạng thiết kế, bố cục, phong cách, canh lề của trang web.

JavaScript: Cải thiện cách hoạt động của trang web.

\_ JavaScript thường được nhúng trực tiếp vào một trang web hoặc được tham chiếu qua file .js riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

**4. The Back End**

**What is the Back End?**

\_ Back-end (phần lập trình trên server) gồm có các thành phần để xử lý các thông tin từ Front-end. Thông thường ám chỉ tới việc tương tác với DBMS (hệ quản trị dữ liệu).

**Microservice Architecture**

\_ Microservice là một kiếu kiến trúc phần mềm. Các module trong phần mềm này được chia thành các service rất nhỏ (microservice). Mỗi service sẽ được đặt trên một server riêng -> dễ dàng để nâng cấp và scale ứng dụng.

**Back-end programming**

\_ Backend engineers : Lập trình viên Backend có nhiệm vụ lập trình máy chủ để xử lý yêu cầu của người dùng và phản hồivới tài nguyên được yêu cầu một cách chính xác. Họ cũng viết mã để xử lý và lưu trữ dữ liệu người dùng.

\_ Back-end web developer tập trung vào việc xây dựng và xử lý logic bằng cách sử dụng ngôn ngữ lập trình dành cho server như PHP hay Ruby- những thứ đã quá quen thuộc với các bạn trên giảng đường đại học. Là một back-end web developer tại BSS, mình làm việc với PHP hàng ngày nên ngôn ngữ này sẽ được mình sử dụng làm ví dụ trong suốt cả bài viết.

\_ Ngoài làm ra các chức năng cho trang web, back-end web developer còn có trách nhiệm phải tìm cách tối ưu chức năng của mình, đảm bảo về tốc độ xử lý và hiệu suất của toàn bộ trang web.

\_ Hơn nữa, các back-end web developer cũng là những người phải làm việc với cơ sở dữ liệu (csdl). Đây là một phần tất yếu của một trang web để lưu trữ các thông tin về người sử dụng, sản phẩm hàng hóa, … Những csdl đã quen thuộc với các back-end web developer là MySQL, MongoDB hay PostgreSQL.

**NodeJS**

\_ Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

\_ Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

\_ Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

\_ Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

\_ Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

**5. Web Development Frameworks**

**What are the Frameworks?**

\_ Các framework ứng dụng web là các framework phần mềm được sử dụng để sắp xếp hợp lý các ứng dụng web và phát triển trang web, các dịch vụ web và các tài nguyên web. Một loại framework ứng dụng web phổ biến là kiến trúc Model-View-Controller (MVC). Như chúng ta đã biết để lập trình một website chúng ta sử dụng tới 3 ngôn ngữ thành phần là HTML, CSS và Javascript sau đó kết hợp với một ngôn ngữ kịch bản máy chủ như PHP, ASP.NET, JAVA… với mỗi thành phần này chúng ta có các framework khác nhau.

**Front-end Frameworks**

5 frameworks phổ biến cho lập trình Front-end :

\_ Angular : là một framework JavaScript do Google tạo ra đã được thiết kếđặc biệt để tạo các ứng dụng web động. AngularJS giải quyếthạn chế của chế độ xem động HTML trong các ứng dụng web và cho phép bạnmở rộng vốn từ vựng HTML cho ứng dụng của bạn. Môi trường kết quả làdiễn đạt cao, dễ đọc và nhanh chóng phát triển. Angular được sử dụng chủ yếuđể cập nhật giao diện người dùng trong thời gian thực và cung cấp tính tương tác cao.

\_ Bootstrap: có thể dễ dàng được phân loại là nguồn mở được sử dụng nhiều nhấttrên thế giới, được tạo ra bởi các nhà phát triển Twitter và chủ yếuphục vụ để dễ dàng quá trình thêm CSS vào HTML. Bootstrap, giống như bất kỳ loại nào khácfront-end framework, bao gồm các thành phần CSS, HTML và JavaScript. Nótuân thủ các tiêu chuẩn thiết kế web đáp ứng và do đó, cho phép người dùng phát triểncác trang web đáp ứng thuộc mọi mức độ phức tạp và quy mô.

\_ React.js: là một thư viện JavaScript để xây dựng giao diện người dùng. Trong khi có nghi ngờ về việc liệu nó có phải là một khuôn khổ thực tế, nó là một thư viện rất phổ biếnđược sử dụng để xây dựng lớp Xem của ứng dụng MVC và đảm bảothảo luận trong bối cảnh phát triển front-end. React là một thành phần-thư viện dựa trên cho phép người dùng xây dựng các thành phần được đóng gói, mỗi thành phầnquản lý trạng thái của chính nó, và sau đó soạn chúng để làm cho Người dùng phức tạpCác giao diện. Lợi ích của việc này là khi dữ liệu thay đổi, React đảm bảorằng chỉ các thành phần liên quan được cập nhật.

\_ Backbone.js: là một framework công tác cực kỳ nhẹ cho phép bạn cấu trúcMã JavaScript ở dạng MVC (Model, View, Controller). Xương sống làm chosử dụng các mô hình , khung nhìn và bộ sưu tập để đảm bảo rằng chương trình của bạn khôngbị vướng vào vô số các lệnh gọi lại và các phần không liên quan khác củamã làm cho các thay đổi phức tạp. Thay vào đó, các mô hình đại diện cho dữ liệucủa ứng dụng và Backbone đảm bảo rằng bất kỳ thay đổi nào đối với các mô hình nàytự động kích hoạt các thay đổi đối với bất kỳ chế độ xem nào hiển thị các mô hình này.

\_ Semantic – UI : Mặc dù Semantic-UI là một framework tương đối mới, nhưng nó nổi bật ở một sốcác cách. Sự khác biệt chính mà khung cụ thể này thích làsự đơn giản. Semantic-UI sử dụng ngôn ngữ tự nhiên và do đó, mã làphần lớn là tự giải thích, do đó rất mong muốn cho người mới bắt đầu,đặc biệt là những người có ít hoặc không có kinh nghiệm viết mã. Ngoài điều này,Semantic-UI cũng được tích hợp với vô số thư viện của bên thứ ba. Điều nàycó nghĩa là quá trình phát triển trở nên dễ dàng hơn nhiều bởi vì,các ứng dụng đơn giản hơn, tất cả các thư viện bạn có thể yêu cầu có thể đãtích hợp với khuôn khổ. Do đó, Semantic-UI là một khởi đầu tuyệt vờiđiểm để phát triển giao diện người dùng của các trang web cấp mới bắt đầu. Tuy nhiên, nókích thước gói lớn hơn đáng kể so với Foundation và Bootstrap,và nó có thể không phải là một lựa chọn khả thi khi phát triển các trang web với nhiềucấu trúc phức tạp

**Back-end Frameworks**

5 frameworks cho lập trình Back-end :

\_ Express (JavaScript) : thường được gọi là "framework máy chủ tiêu chuẩn trên thực tế" cho Node.js, là một khung ứng dụng web Node.js tối thiểu và rất linh hoạtcung cấp một bộ tính năng mạnh mẽ cho các ứng dụng web. Ý tưởng đằng sau Express là đơn giản hóa quy trình phát triển back-end đủ để giảm các tính năng back-end cơ bản như tạo một máy chủ HTTP từ nhiều các dòng mã Node.js khó hiểu, cho một lệnh duy nhất, trong khiđồng thời giữ các đặc điểm cú pháp và logic cốt lõi của Node.js còn nguyên vẹn.

\_ Symfony : là một framework phát triển web cho PHP. Ý tưởng chính đằng sau Symfony về cơ bản là một tập hợp các thành phần có thể tái sử dụng. Những lợi ích củađiều này là hiển nhiên; với hơn 30 thành phần độc lập theo ý của bạn, quy trình viết đơn trở nên đơn giản hóa đáng kể và, do đó, dễ dàng hơn và hiệu quả hơn nhiều. Symfony, giống như phiên bản trướccác khuôn khổ chúng tôi đã thảo luận, cho phép nhỏ gọn hơn, dễ đọc hơn và mã đơn giản. Điều này có lợi vì nó cho phép bạn tập trung vàochức năng cấp mà bạn muốn triển khai khi các nhiệm vụ đơn giản hơn trở nên giới hạn trong các đoạn mã nhỏ hơn nhiều mà không cần phải suy nghĩ về từ đầu.

\_ Django (Python) : là một framework công tác web Python cấp cao cho phép phát triển các trang web an toàn và có thể bảo trì. Django, giống như Express, mất quan tâm đến việc giảm mã cho các chức năng cơ bản thành các hướng dẫn đơn giản, vì vậy bạn có thể tập trung vào việc viết ứng dụng của bạn và triển khai cụ thể hơnchức năng mà không cần phải phát minh lại bánh xe và bắt kịp những đoạn mã dài không cần thiết để làm như vậy.

\_ Ruby on Rails (Ruby) : còn được gọi là Rails, là một ứng dụng web phía máy chủkhung viết bằng Ruby. Rails được xây dựng trên Model View Controller kiến trúc mà chúng tôi đã nghiên cứu trước đây và nó cung cấp cấu trúc cho mọi thứ đi vào từng thành phần, bao gồm cả cơ sở dữ liệu,một số dịch vụ web phổ biến cũng như chính các trang web. Viên ngọc trên tay vịn,do đó, tách quá trình phát triển web thành các thành phần và cung cấp cấu trúc cơ bản cho từng phần tử bất kỳ thành phần bao gồm, do đó vô cùng thuận tiện cho người dùng xây dựng chức năng cụ thể của riêng chúng trên đầu các cấu trúc tạm thời này mà không cần phải lo lắng về việc bắt đầu từ những điều rất cơ bản.

\_ASO .NET (C#) : là một framework mã nguồn mở cho xây dựng các ứng dụng và dịch vụ web hiện đại sử dụng .NET hoặc bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET. ASP .NET được phát triển bởi Microsoft để tạo điều kiện cho quá trình phát triển các trang web động và do đó, đáng kể đơn giản hóa quá trình tạo trang web dựa trên HTML5, CSS và JavaScript. Do đó, người dùng có thể tạo các trang web không chỉ đơn giản và nhanh chóng mà còn có khả năng mở rộng lên hàng triệu người dùng.

**Isomorphic Frameworks**

\_ Meteor JS : Meteor, hoặc Meteor JS, là một framework JavaScript máy chủ-máy khách mã nguồn mở đã được viết trong Node.js. Ngoài việc có lợi ích là khuôn khổ độc lập cho phép phát triển cả mặt trước và mặt sau của ứng dụng, Meteor cũng cho phép tạo mẫu, có nghĩa lànó cho phép các phiên bản của cùng một ứng dụng được lưu trữ và thử nghiệm. Ngoài ra, Meteor cũng sản xuất mã có thể mở rộng nhiều nền tảng bao gồm cả Android, IOS và chính web.Ở phía máy chủ, Meteor tích hợp với MongoDB và tuân theo các giao thức để tạo cơ chế quản lý cơ sở dữ liệu cho phép thay đổi dữ liệu được truyền trực tiếp đến phía khách hàng mà không cần phải viết bất kỳ mã đồng bộ hóa ở cả hai đầu một cách rõ ràng.Về phía máy khách, Meteor có công cụ tạo khuôn mẫu của riêng mình được gọi là Blaze.cho phép người dùng chọn từ các mẫu cho giao diện người dùng để tùy chỉnh, do đó làm cho việc phát triển giao diện người dùng dễ dàng hơn nhiều. Ngoài ra, Meteor có thể cũng được sử dụng với một số khuôn khổ giao diện người dùng mà chúng tôi đã nghiên cứu. Nó có thể được tích hợp với các khuôn khổ Angular hoặc React để liền mạch sản xuất các giao diện người dùng không yêu cầu quá nhiều công việc.

\_ Lazo.js :là một framework phát triển web máy khách-máy chủ được xây dựng trên Node.js có lợi ích là cung cấp cho các nhà phát triển front-end một cách đơn giản và tương đối cấu trúc cú pháp quen thuộc mà họ có thể sử dụng để tạo web có cấu trúc MVC các ứng dụng được phân tách thành nhiều thành phần phức tạp kết hợp với nhau để tạo thành giao diện người dùng hoàn chỉnh. LazoJS có thêm lợi ích là cung cấp tải trang đầu tiên được tối ưu hóa, một thứ thường là chìa khóa yếu tố quyết định trải nghiệm người dùng. Những lợi ích này đạt được bằng cách sử dụng sự kết hợp của các công cụ bao gồm Backbone.js, RequiJS và jQuery.Ban đầu, Lazo được tạo ra để giải quyết các vấn đề xảy ra từ việc tạo ra một toàn bộ giao diện trang web kết thúc dưới dạng một đơn vị hoặc các ứng dụng trang đơn thay vìtách giao diện người dùng thành các thành phần loại trừ lẫn nhau. Lazo, trong biến, không chỉ cung cấp cho các kỹ sư front-end một môi trường quen thuộc đểtạo các ứng dụng web hoàn chỉnh nhưng cũng đảm bảo rằng các thành phầntách biệt như vậy để tránh logic ứng dụng quá phức tạp. Các trang là được xây dựng thông qua các thành phần có thể tái sử dụng, có thể kiểm tra có vòng đời riêng của chúng,do đó cho phép các nhà phát triển dễ dàng tạo các khung nhìn phức tạp cho giao diện đồng thời cung cấp khả năng đóng gói và phân tách các mối quan tâm một cách tuyệt vời.

\_ Rendr : Rendr là một thư viện nhỏ cho phép bạn chạy các ứng dụng Backbone.js của mình trên cả máy khách và máy chủ, tùy thuộc vào nhu cầu cụ thể của web ứng dụng bạn đang phát triển. Điều này có nghĩa là Rendr cung cấp cho người dùng cơ chế xây dựng toàn bộ ứng dụng trên máy khách hoặc máy chủ kết thúc bằng ngoài khả năng phân phối mã trên cả máy khách và máy chủ và sử dụng Rendr ở cả hai đầu. Về cơ bản, Rendr cho phép máy chủ web của bạn phục vụcác trang HTML được định dạng đầy đủ đến bất kỳ liên kết nào trong ứng dụng của bạn, trong khi đồng thời duy trì cảm giác của phía máy khách Backbone.js truyền thốngỨng dụng MVC

**Software Stacks**

\_ Software Stack (Ngăn xếp phần mềm) là một tập hợp các chương trình, công nghệ được sử dụng cùng nhau để tạo ra một kết quả cụ thể.

\_Nó bao gồm cả một hệ điều hành và ứng dụng của nó :

**LAMP STCK** : LAMP là một stack được sử dụng rộng rãi bởi lập trình viên web Full stack. Tên của nó 'LAMP' là một từ viết tắt của bốn thành phần nguồn mở:

* L = Linux: Hệ điều hành nguồn mở Linux
* A = Apache: Phần mềm máy chủ web được sử dụng rộng rãi
* M = MySQL: Cơ sở dữ liệu nguồn mở phổ biến
* P = PHP: Ngôn ngữ kịch bản mã nguồn mở phía máy chủ

Các thành phần ở trên hỗ trợ lẫn nhau. Và có nhiều trang web và ứng dụng web phổ biến chạy trên LAMP stack, kể cả trang web lớn, ví dụ: Facebook.

**MERN Stack :** MERN là tập hợp các công nghệ dựa trên JavaScript, bao gồm:

* M = MongoBD: Cơ sở dữ liệu NoSQL phổ biến
* E = Express: Web Framework nhẹ và di động
* R = React: Thư viện Javascript để xây dựng giao diện người dùng
* N = Node.js: Môi trường thực thi Server - side Javascript

Với sự phổ biến của Javascript, Stack này đang có nhu cầu rất lớn và được sử dụng rộng rãi để lập trình website.

**MEAN Stack** : MEAN Stack là một bộ công nghệ dựa trên Javascript nhưng hơi khác với MERN đang chứng kiến ​​một xu hướng sử dụng ngày càng tăng. MEAN là tên viết tắt của:

* M = MongoDB: Cơ sở dữ liệu NoSQL
* E = Express: Web Framework nhẹ và di động
* A = Angular.js: Web Framework mạnh mẽ để phát triển HTML5 và Javascript
* N = Node.js: Môi trường thực thi Server - side Javascript