**ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**Khoa Công Nghệ Thông Tin**

A picture containing text, sign

Description automatically generated

**MÔN: CÔNG NGHỆ WEB**

**BÁO CÁO ĐỀ TÀI: MEAN STACK**

**GV hướng dẫn:** Th.S Lương Trần Hy Hiến

**Lớp học phần:** 2111COMP103101

**Nhóm sinh viên thực hiện: KTQ**

1. Trần Đoàn Đăng Khoa - 4501104114

2. Võ Phúc Thiện - 4501104225

3. Nguyễn Lê Anh Quốc – 4501104193

4. Trương Nguyễn Phúc – 4501104181

**Thành phố Hồ Chí Minh, ngày 05 tháng 12 năm 2021**

**ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

A picture containing text, sign

Description automatically generated

**Khoa Công Nghệ Thông Tin**

**MÔN: CÔNG NGHỆ WEB**

**BÁO CÁO ĐỀ TÀI: MEAN STACK**

**GV hướng dẫn:** Th.S Lương Trần Hy Hiến

**Lớp học phần:** 2111COMP103101

**Nhóm sinh viên thực hiện: KTQ**

1. Trần Đoàn Đăng Khoa - 4501104114

2. Võ Phúc Thiện - 4501104225

3. Nguyễn Lê Anh Quốc – 4501104193

4. Trương Nguyễn Phúc – 4501104181

**Thành phố Hồ Chí Minh, ngày 05 tháng 12 năm 2021**

**LỜI CẢM ƠN !**

“Đặc biệt, nhóm em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn - Thầy Lương Trần Hy Hiến đã quan tâm hướng dẫn, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp “ Công nghệ Web ”của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù, chúng em đã cố gắng hết sức nhưng chắc chắn khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong quý thầy xem xét và góp ý để bài báo cáo của nhóm chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn thầy !”

Người viết

Nhóm sinh viên: KTQ

TP.HCM, tháng 12 năm 2021

**Mục Lục**

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc88941920)

[1.1 giới thiệu Mean Stack 1](#_Toc88941922)

[1.2 Quy trình hoạt động của MEAN Stack 1](#_Toc88941922)

[CHƯƠNG 2. NỘI DUNG CHÍNH 3](#_Toc88941921)

[2.1 ANGULAR JS 3](#_Toc88941923)

[2.1.1 Khái niệm 3](#_Toc88941924)

[2.1.2 Tính năng AngularJS. 3](#_Toc88941925)

[2.1.3 Ưu – nhược điểm. 4](#_Toc88941926)

[2.2 MONGO DB 5](#_Toc88941927)

[2.2.1 Khái niệm 5](#_Toc88941928)

[2.2.2 Tính năng 5](#_Toc88941929)

[2.2.3 Ưu điểm của MongoDB 5](#_Toc88941930)

[2.3 NODE JS 7](#_Toc88941931)

[2.3.1 Định nghĩa NodeJS 7](#_Toc88941932)

[2.3.2 Những ứng dụng của NodeJS trong lập trình web 8](#_Toc88941933)

[2.3.3 Điểm mạnh và hạn chế 8](#_Toc88941934)

[2.4 EXPRESS JS 10](#_Toc88941935)

[2.4.1 Định nghĩa ExpressJS 10](#_Toc88941936)

[2.4.2 Công dụng của ExpressJS 10](#_Toc88941937)

[2.4.3 Các tính năng của ExpressJS 11](#_Toc88941938)

[2.4.4 Những lợi ích của ExpressJS 11](#_Toc88941939)

[CHƯƠNG 3: KẾT LUẬN 12](#_Toc88941940)

[CHƯƠNG 4: NGUỒN THAM KHẢO 12](#_Toc88941941)

# MỞ ĐẦU

**1.1 Giới thiệu về Mean Stack**

MEAN Stack là một tập hợp 4 công nghệ liên kết với JavaScript. Chúng sở hữu những tính năng ưu việt với mục đích tối ưu hóa hiệu suất xây dựng phần mềm. Điểm khác biệt của MEAN Stack là nó cho phép tạo nên sự kết nối giữa client với ngôn ngữ server và ngôn ngữ cơ sở dữ liệu. Nó được sử dụng để pháp triển những ứng dụng web full stack.

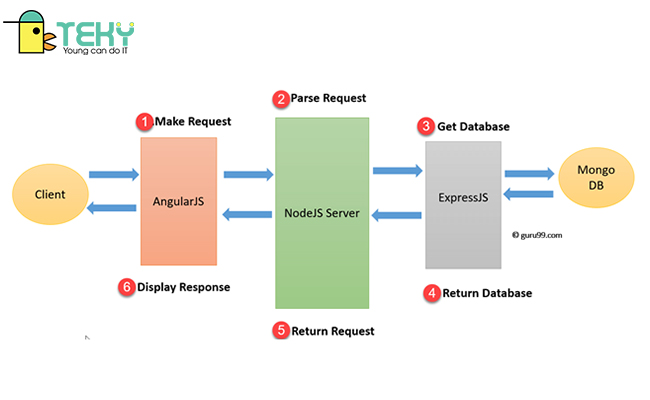
Thuật ngữ MEAN được viết tắt từ tên của 4 ứng dụng tạo thành nó:

* M là MongoDB
* E là ExpressJS
* A là AngularJS
* N là NodeJS

MEAN Stack được sử dụng phổ biến như vậy vì nó là giải pháp tốt nhất để tạo nên một mô hình SPA hoàn hảo nhất.

## Quy trình hoạt động của MEAN Stack

Quy trình hoạt động của MEAN Stack được chia thành 3 phần chủ chốt. Trong đó, ta sẽ có AngularJS chịu trách nhiệm với tương tác người dùng, NodeJS và ExpressJS làm việc với bên server, còn MongoDB sẽ đảm nhận phần cơ sở dữ liệu.



Hình 0.1 Quy trình hoạt động chi tiết

Khi bắt đầu muốn sử dụng MEAN Stack, một yêu cầu sẽ được phát đi thông qua AngularJS. Yêu cầu này được chuyển đến server. NodeJS sẽ tiếp xúc với yêu cầu đầu tiên và đưa ra các phân tích cú pháp. Sau khi quá trình phân tích kết thúc, ExpressJS nhận được kết quả về các phần dữ liệu cần thiết và đưa ra thông báo cho MongoDB. MongoDB sẽ lấy dữ liệu được yêu cầu từ Database trả về cho ExpressJS. ExpressJS lại truyền dữ liệu về cho NodeJS. Và cuối cùng, NodeJS gửi kết quả dữ liệu cho AngularJS. AngularJS sẵn sàng hiển thị thông tin mà khách hàng muốn có trên giao diện màn hình.

# NỘI DUNG CHÍNH

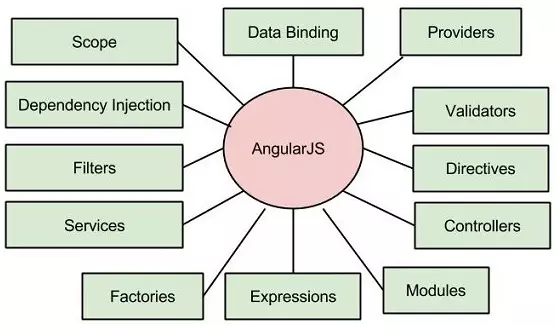
## ANGULAR JS



### Khái niệm

* Angular JS được xem là một open source (mã nguồn mở) được phát triển bởi Google.
* AngularJS là một Framework được viết bằng Javascript, dùng để tạo các ứng dụng trang web động.
* Sử dụng ngôn ngữ HTML làm ngôn ngữ mẫu và mở rộng khả năng của HTML bằng cách thêm các thuộc tính và thành phần dựng sẵn và cũng cung cấp khả năng tạo các thuộc tính tùy chỉnh bằng cách sử dụng JavaScript đơn giản.

### Tính năng AngularJS.



(**Data**-**binding**: Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.

Có hai cách để ràng buộc dữ liệu:

* Khai thác dữ liệu trong các hệ thống mẫu cổ điển.
* Ràng buộc dữ liệu trong các mẫu.

**Scope**: Scope đề cập đến mô hình ứng dụng, nó hoạt động như cầu nối giữa controller và view. Scope được sắp xếp theo cấu trúc phân cấp và cấu trúc DOM của ứng dụng. Nó có thể xem biểu thức và truyền các sự kiện.

**Controller**: Đây là những tính năng của AngularJS mà được giới hạn tới một scope cụ thể.

**Model View Whatever:** MVC là một mô hình thiết kế để phân chia các ứng dụng thành nhiều phần khác nhau (gọi là Model, View và Controller), một phần sử dụng với một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel. Nhóm phát triển AngularJS đã đặt tên vui cho mô hình này là Model View Whatever.

**Services**: Trong các services AngularJS là các đối tượng singleton hoặc các hàm được sử dụng để thực hiện các nhiệm vụ cụ thể. Nó nắm giữ một số logic và chức năng này có thể được gọi là bộ điều khiển, chỉ thị, bộ lọc.

**Directives**: Một chỉ thị trong AngularJS là một lệnh cung cấp chức năng HTML mới. Khi nhìn qua mã HTML, đầu tiên ta sẽ tìm thấy các chỉ thị trong trang và sau đó phân tích cú pháp trang HTML cho phù hợp. Lưu ý: Bạn có thể có mã góc cơ bản trong một trang HTML với chỉ thị ng-init, ng-repeat và ng-model mà không cần phải có Controllers. Logic cho các chỉ thị này nằm trong tệp Angular.js do Google cung cấp.

**Filters**: Để định dạng dữ liệu trước khi hiển thị, bạn có thể sử dụng bộ lọc, nó lựa chọn (hay là lọc) các tập con từ tập item trong các mảng và trả về các mảng mới.

**Deep Linking:** Cho phép bạn mã hóa trạng thái các ứng dụng trên địa chỉ URL để nó có thể được bookmark. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.

**Dependency Injection:** AngularJS có sẵn một hệ thống con dependency injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và kiểm tra.

### Ưu – nhược điểm.

**Ưu điểm của angularJS**

* Cung cấp khả năng tạo ra các Single Page Aplication dễ dàng.
* Cung cấp khả năng data binding tới HTML, khiến cho người dùng cảm giác linh hoạt, thân thiện.
* Dễ dàng Unit test
* Dễ dàng tái sử dụng component.
* Giúp lập trình viên viết code ít hơn với nhiều chức năng hơn.
* Chạy được trên các loại trình duyệt, trên cả PC lẫn mobile.

**Nhược điểm**

* Không an toàn : được phát triển từ javascript cho nên ứng dụng được viết bởi AngularJS không an toàn. Nên có sự bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng ứng dụng của vô hiệu hóa JavaScript thì sẽ chỉ nhìn thấy trang cơ bản.

## MONGO DB



### Khái niệm

* MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL.
* Hỗ trợ trên nhiều nền tảng khác nhau và được viết bằng ngôn ngữ C++ với mục đích hướng đến đối tượng.
* Mongo DB là một lựa chọn lý tưởng cho một hệ thống cơ sở dữ liệu, nơi bạn cần quản lý các bảng có kích thước lớn với hàng triệu dữ liệu.

### Tính năng

* **Truy vấn ad hoc**

Đây được xem là tính năng tốt nhất của MongoDB. Nó có tác dụng giúp các trường truy vấn phạm vi, kết hợp cùng việc tìm kiếm biểu thức nhằm trả về kết quả tài liệu cụ thể với kích thước nhất định. Ngoài ra, các trường dùng trong MongoDB còn được dùng để lập chỉ mục chính, phụ.

* **Nhân rộng**

MongoDB cung cấp Replica Set cho phép nhân 2 hoặc nhiều bản sao của dữ liệu. Đồng thời, mỗi bản sao lại đóng vai trò chính và phụ.

- Khi nhân rộng, toàn bộ dữ liệu khi ghi và đọc được thực hiện trên bản sao chính.

- Bản sao thứ cấp sẽ dùng bản sao tích hợp để có thể duy trì các bản sao dữ liệu.

Trong trường hợp có bất kỳ bản sao chính nào bị thất bại thì Replica set sẽ chọn một bản sao thứ cấp để thay thế làm bản sao chính tiếp theo. Trong quá trình nhân rộng, Replica thứ cấp được tùy ý chọn các hoạt động nhưng dữ liệu cuối cùng vẫn phải tuân theo mặc định.

* **Cân bằng tải**

MongoDB sử dụng Sharding nhằm chia tỷ lệ theo chiều ngang và xác định dữ liệu phân phối trong collection. Điều này giúp người dùng có thể chọn một Shard key.

Nói tóm lại, MongoDB cân bằng tải bằng cách dựa vào các Shard key để chia dữ liệu thành các phạm vi và phân phối đồng đều. Chúng có thể chạy trên nhiều máy chủ khác nhau và thực hiện chức năng sao chép dữ liệu hay cân bằng tải nhằm giữ hệ thống hoạt động liên tục trong trường hợp phát sinh lỗi về phần cứng.

* **Lưu trữ tệp**

Khi tìm hiểu hệ cơ sở dữ liệu MongoDB thì bạn sẽ thấy, tính năng lưu trữ tệp được dùng như một hệ thống tệp (gọi là GridFS) đóng vai trò cân bằng tải, đồng thời, sao chép dữ liệu trên nhiều máy tính. Cụ thể, GridFS chia một tệp ra làm nhiều phần và lưu trữ thành các tài liệu riêng biệt. Sau đó, người dùng dễ dàng truy cập GridFS thông qua Mongofiles hay các plugin sử dụng cho Nginx và Lighttpd.

* **Tập hợp**

Tính năng này chính là chương trình mang đến ba giải pháp để thực hiện tập hợp gồm Aggregation Pipeline, Mapreduce và Single-purpose Aggregation. Trong đó, Aggregation Pipeline được đánh giá là có hiệu suất tốt nhất.

* **Giới hạn kích thước collection**

Các collection được MongoDB hỗ trợ thường có kích thước cố định. Vì thế, người ta gọi chúng là collection giới hạn. Với kích cỡ cố định, kết hợp cùng việc theo sau thứ tự chèn giúp tăng hiệu suất của các hoạt động liên quan đến dữ liệu. Và khi dữ liệu vượt giới hạn thì những tài liệu cũ hơn sẽ tự động bị xóa mà bạn không cần thực hiện thao tác thêm bất kỳ dòng lệnh nào.

* **Giao dịch**

Bắt đầu từ phiên bản 4.0 ra mắt vào tháng 6 năm 2018, MongoDB được bổ sung thêm tính năng hỗ trợ giao dịch ACID đa tài liệu.

### Ưu điểm của MongoDB

* Linh hoạt trong lưu trứ các kích cỡ dữ liệu khác nhau. Lưu dưới dạng JSON.
* Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa, cập nhật dữ liệu.
* Không có sự ràng buộc trong một khuôn khổ, quy tắc nhất định nào.
* Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM.

**Nhược điểm**

* Vì dữ liệu không bị ràng buộc nên trong quá trình sử dụng, bạn cần cẩn thận trong mọi thao tác nhằm tránh xảy ra những điều không mong muốn, làm ảnh hưởng đến dữ liệu.
* Chương trình MongoDB tiêu tốn khá nhiều dung lượng bộ nhớ do dữ liệu được lưu dưới dạng key và value
* Thời gian để dữ liệu chuyển đổi từ RAM xuống ổ cứng khoảng 60s nên nguy cơ bị mất dữ liệu nếu xảy ra mất điện là điều có thể xảy ra.

## NODE JS

### Định nghĩa NodeJS

NodeJS là mã nguồn mở chạy trên môi trường V8 JavaScript runtime. NodeJS giúp các nhà phát triển xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS có thể được dùng để xây dựng các loại ứng dụng khác nhau như các ứng dụng dòng lệnh, ứng dụng web, ứng dụng trò chuyện theo thời gian thực, máy chủ [REST API](https://wiki.tino.org/restful-api-la-gi/),.. Tuy nhiên, NodeJS thường được dùng chủ yếu để xây dựng các chương trình mạng như máy chủ web, tương tự như PHP, Java hoặc ASP.NET.

Đặc điểm Node

NodeJS không cần đợi API trả dữ liệu về, do đó mọi APIs nằm trong thư viện NodeJS đều không được đồng bộ.

Đây là một Platform chứ không phải là một Framework. Do đó, NodeJS cho phép bạn có thể xây dựng các website một cách độc lập và nhanh chóng hơn.

NodeJS có thể chạy trên đa nền tảng gồm: Window, MacOS, Linux.

NodeJS được xem là một máy chủ đơn luồng và không thể hỗ trợ đa luồng.

NodeJS không được xem là một ngôn ngữ lập trình, nên những người mới phải nắm chắc kiến thức lập trình căn bản như: các giao thức, Javascript,… mới có thể sử dụng NodeJS. Tuy nhiên, cộng đồng NodeJS thường rất lớn, sẵn sàng support cho bạn mọi lúc mọi nơi.

Phần core của NodeJS thường được biết bằng ngôn ngữ C++ nên nó hiệu năng và tốc độ xử lý tương đối cao. Nhờ vậy, hầu hết các ứng dụng NodeJS đều có khả năng đáp ứng được thời gian thực chạy trên đa nền tảng, đa thiết bị,…

### Những ứng dụng của NodeJS trong lập trình web

NodeJS có thể tạo, mở, đọc, ghi, xóa và đóng các tệp ngay khi đang truy cập trên máy chủ.

Xây dựng nội dung cho các trang web động.

Thực hiện thu thập dữ liệu theo yêu cầu cụ thể.

Thực hiện truy vấn, sửa, xóa, thêm các dữ liệu trong các hệ quản trị cơ sở như: Microsoft SQL Server, [MySQL](https://wiki.tino.org/mysql-la-gi/), [MongoDB](https://wiki.tino.org/mongodb-la-gi/), PostgreSQL.

### Điểm mạnh và hạn chế

* ***Điểm mạnh:***

IO hướng sự kiện không đồng bộ giúp xử lý nhiều yêu cầu đồng thời.

Đáp ứng được những yêu cầu về thời gian thực.

Có tốc độ cực rất nhanh, đáp ứng được nhu cầu sử dụng của khách truy cập ‘khổng lồ’ trong thời gian ngắn.

Sử dụng JavaScript, một ngôn ngữ lập trình rất dễ học.

Chia sẻ cùng một đoạn mã với cả phía máy chủ và máy khách.

Npm và các module rất mạnh mẽ và vẫn đang tiếp tục phát triển.

Có một cộng đồng lớn mạnh, có nhiều mã được chia sẻ qua github

Tương thích với nhiều thiết bị, nhiều hệ điều hành như MacOS, Window, Linux,…

* ***Hạn chế:***

NodeJS không cung cấp khả năng mở rộng và không thể tận dụng lợi thế của nhiều lõi thường có trong phần cứng cấp máy chủ ngày nay.

Thao tác với cơ sở dữ liệu quan hệ là một khó khăn nếu bạn đang sử dụng NodeJS

Mỗi lần sử dụng lệnh gọi lại sẽ kết thúc với rất nhiều lệnh gọi lại lồng vào nhau.

Nếu không hiểu rõ về JavaScript, bạn sẽ gặp khó khăn với NodeJS

NodeJS không phù hợp với các tác vụ đòi hỏi nhiều CPU mà chỉ phù hợp với những I/O như máy chủ web.

Nếu bạn có một web hosting dùng chung, sẽ rất khó khăn nếu bạn tải lên một ứng dụng NodeJS. VPS và Dedicated server là một sự lựa chọn tốt hơn nhiều.

Những loại ứng dụng dưới đây nên được viết bằng NodeJS:

Fast File Upload: Những chương trình upload file với tốc độ cao.

Websocket server: Các máy chủ của web socket gồm các dạng như: Game server, online chat,…

Restful API: gồm các ứng dụng được dùng cho những ứng dụng khác thông qua API.

Any Real-time Data Application: Những ứng dụng có yêu cầu cao về tốc độ thực hiện trong thời gian thực.

Ad server: Các máy chủ quảng cáo.

Tại thời điểm này, chúng ta vẫn chưa thể cho rằng NodeJS đủ mạnh để thay thế hoàn toàn cho .NET, PHP hay Java. Nhưng trong tương lai gần, NodeJS hoàn toàn có thể trở thành một giải pháp tối ưu nhất cho web back-end. Hy vọng bài viết trên sẽ giúp bạn nắm được những thông tin cơ bản về NodeJS và nghiên cứu sâu hơn sau này.

## EXPRESS JS



### Định nghĩa ExpressJS

ExpressJS là một [framework](https://wiki.tino.org/framework-la-gi/) ứng dụng web có mã nguồn mở và miễn phí được xây dựng trên nền tảng Node.js. ExpressJS được sử dụng để thiết kế và phát triển các ứng dụng web một cách nhanh chóng. Để hiểu ExpressJS, người dùng chỉ cần phải biết [JavaScript](https://wiki.tino.org/html-css-javascript-la-gi/), do đó nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn đối với các lập trình viên và nhà phát triển đã thành thạo JavaScript trước đó.

Vì ExpressJS là một framework của Node.js nên hầu hết các mã đã được viết sẵn cho các lập trình viên làm việc. Bạn có thể tạo các ứng dụng web cho một trang, nhiều trang hoặc kết hợp lại bằng cách sử dụng ExpressJS. framework này khá nhẹ, giúp tổ chức các ứng dụng web ở phía máy chủ thành một kiến trúc [MVC](https://wiki.tino.org/mvc-la-gi/) hoàn hảo hơn.

ExpressJS hỗ trợ nâng cao các chức năng của NodeJS. Nếu bạn không sử dụng ExpressJS, bạn phải thực hiện rất nhiều lập trình phức tạp để xây dựng một API hiệu quả. ExpressJS đã giúp cho việc lập trình trong NodeJS trở nên dễ dàng hơn rất nhiều.

### Công dụng của ExpressJS

ExpressJS sẽ giúp bạn tổ chức kiến trúc [back-end](https://wiki.tino.org/frontend-la-gi-backend-la-gi/) của mình. Các nhà phát triển web thường sử dụng ExpressJS để triển khai kiến trúc MVC, điều này cho phép họ viết một codebase back-end bảo trì tương đối dễ dàng.

Bởi vì ExpressJS hoạt động trên back-end, nên bạn có thể coi công nghệ này như một “bộ não đằng sau một trang web”. Ví dụ: ExpressJS có thể xác định cách những trang được định tuyến trên một trang web. Hơn nữa, một nhà phát triển có thể sử dụng ExpressJS để quản lý xác thực trên một trang web.

Dưới đây là một số trường hợp sử dụng của ExpressJS:

* Sử dụng cookie trên một trang web
* Triển khai xác thực
* Thêm thanh tìm kiếm vào một trang web
* Cung cấp các tệp tĩnh như hình ảnh

### Các tính năng của ExpressJS

***Phát triển máy chủ nhanh hơn***

ExpressJS cung cấp cho bạn nhiều tính năng phổ biến của Node.js dưới dạng hàm có thể dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này sẽ giúp rút ngắn thời gian để viết code.

***Phần mềm trung gian***

Phần mềm trung gian là một phần trong chương trình cho phép truy cập vào cơ sở dữ liệu, xem xét yêu cầu của khách hàng và các phần mềm trung gian khác. Tính năng này chịu trách nhiệm chính cho việc tổ chức chức năng khác nhau của ExpressJS.

***Định tuyến***

ExpressJS cung cấp một cơ chế định tuyến nâng cao giúp duy trì trạng thái của trang web.

***Khuôn mẫu***

ExpressJS cung cấp các công cụ tạo khuôn mẫu cho phép các nhà phát triển tạo nội dung động trên các trang web bằng việc xây dựng các mẫu HTML ở phía máy chủ.

***Gỡ lỗi***

Gỡ lỗi là yếu tố quan trọng để phát triển các ứng dụng web. ExpressJS giúp gỡ lỗi dễ dàng hơn bằng cách cung cấp một cơ chế có khả năng xác định chính xác phần ứng dụng web có lỗi.

### Những lợi ích của ExpressJS

Rất dễ học, chỉ cần bạn biết JavaScript, bạn sẽ không cần phải học một ngôn ngữ mới để học ExpressJS

Giúp cho việc phát triển back-end dễ dàng hơn nhiều khi sử dụng ExpressJS

Mã JavaScript được diễn giải thông qua Google V8 JavaScript Engine của Node.js. Do đó, mã sẽ được thực hiện một cách nhanh chóng và dễ dàng.

ExpressJS rất đơn giản để tùy chỉnh và sử dụng theo nhu cầu.

Cung cấp một module phần mềm trung gian linh hoạt và rất hữu ích để thực hiện các tác vụ bổ sung theo phản hồi và yêu cầu.

Hỗ trợ phát triển ứng dụng theo mô hình MVC, đây là mô hình phổ biến cho việc lập trình web hiện nay.

# CHƯƠNG 3: KẾT LUẬN

MEAN Stack đang từng bước chinh phục người dùng toàn cầu nhờ cơ chế thông minh và nền tảng mạnh mẽ của nó. Với MEAN Stack hoạt động phát triển phần mềm và website diễn ra với tốc độ và cường độ cao hơn. Đó là lý do rất nhiều người muốn tìm học MEAN Stack là gì. Nếu bạn muốn tìm hiểu sâu hơn về lĩnh vực này, hãy tìm kiếm những tài liệu miễn phí khác trên Internets nhé. Mong rằng những thông tin mà Teky vừa mang đến cho bạn trong bài viết này sẽ phần nào giải đáp được những thắc mắc về MEAN Stack.

# CHƯƠNG 4: NGUỒN THAM KHẢO

1. Son Che Dinh, AngularJS là gì? [https://viblo.asia/p/angularjs](https://viblo.asia/p/angularjs-toan-tap-phan-1-angularjs-la-gi-MgNvWYZEeYx#_cac-tinh-nang-co-ban-3)
2. AngularJS là gì? Ưu, nhược điểm của framework AngularJS, <https://www.devpro.edu.vn/angularjs-la-gi>
3. MongoDB là gì? Tìm hiểu về MongoDB, <https://hostingviet.vn/mongodb-la-gi>
4. NodeJS là gì? Tại sao NodeJS lại quan trọng với lập trình web?, <https://wiki.tino.org/nodejs-la-gi/>
5. ExpressJS là gì? Tại sao nên sử dụng ExpressJS?, <https://wiki.tino.org/expressjs-la-gi/>
6. Node Document, <https://nodejs.org/en/docs/>

HẾT.