**BỘ LAO ĐỘNG THƯƠNG BINH VÀ XÃ HỘI  
TRƯỜNG CAO ĐẲNG BÌNH MÌNH SÀI GÒN**

----o0o----



**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

WEB QUẢN LÝ KHÁCH SẠN

GVHD: ThS. Trương Bá Vĩnh

SVTH: Nguyễn Hữu Trọng

MSSV: 221ACNTT01

Lớp: Công nghệ thông tin

TP.HCM, Tháng 06 năm 2024

**NHẬN XÉT GIÁO VIÊN HƯỚNG DẪN**

1. **Họ và tên sinh viên:**
2. **Tên đề tài**:

1. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên:……………………………………………………………………

Điểm số: ……….…… Điểm chữ: …………………………………………

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20…*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

**Lời cảm ơn:**

Tôi xin cảm ơn chân thành đến với những người đã viết tài liệu đã được tham khảo trong bài báo cáo này cũng như giảng viên hướng dẫn của tôi, Nguyễn Quốc Ninh, vì sự chỉ dẫn.

**Cam kết:**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của giảng viên Nguyễn Quốc Ninh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các Bảng phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số khái niệm có trích dẫn và chú thích nguồn gốc. Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình.

MỤC LỤC

[CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU 1](#_Toc167830917)

[1.1. Giới thiệu sơ lược về đề tài: 1](#_Toc167830918)

[1.2. Mục đích và ý nghĩa 1](#_Toc167830919)

[1.2.1. Mục đích: 1](#_Toc167830920)

[1.2.2. Ý nghĩa: 1](#_Toc167830921)

[1.3. Lý do chọn đề tài: 2](#_Toc167830922)

[1.4. Mục tiêu của đề tài 2](#_Toc167830923)

[1.5. Nội dung chính của đề tài: 2](#_Toc167830924)

[1.6. Nhiệm vụ phải thực hiện: 3](#_Toc167830925)

[1.6.1 Mục tiêu: 3](#_Toc167830926)

[1.6.2 Kết quả: 3](#_Toc167830927)

[1.7. Phân công việc trong từng giai đoạn: 3](#_Toc167830928)

[1.7.1. Nội dung công việc: 3](#_Toc167830929)

[1.7.2. Lịch trình thực hiện: 3](#_Toc167830930)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 5](#_Toc167830931)

[2.1. Môi trường lập trình 5](#_Toc167830932)

[2.1.1. Giới thiệu về Visual Studio Code 5](#_Toc167830933)

[2.1.2. Giới thiệu về Intellij Idea 5](#_Toc167830934)

[2.2. Framework và thư viện 6](#_Toc167830935)

[2.2.1 Giới thiệu về Reactjs 6](#_Toc167830936)

[2.2.2. Giới thiệu về Spring boots 7](#_Toc167830937)

[2.2.3. Giới thiệu về Spring Security 8](#_Toc167830938)

[2.2.4. Giới thiệu về Hibernate 9](#_Toc167830939)

[2.3 Giới thiệu mô hình MVC 9](#_Toc167830940)

[2.4. Phân tích đề tài với UML 10](#_Toc167830941)

[2.4.1. Sơ đồ use case tổng quát 10](#_Toc167830942)

[2.4.2. Sơ đồ use case chức năng 10](#_Toc167830943)

[2.5. Mô hình quan hệ Database 10](#_Toc167830944)

[2.6. Database 10](#_Toc167830945)

[TỔNG KẾT 11](#_Toc167830946)

[TÀI LIỆU THAM KHẢO 12](#_Toc167830947)

# CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU

## 1.1. Giới thiệu sơ lược về đề tài:

Hiện nay, ngành khách sạn là một trong những ngành công nghiệp lớn và liên tục phát triển, đòi hỏi sự quản lý chuyên nghiệp cũng như quảng bá hiệu quả. Trong bối cảnh cạnh tranh ngày càng gay gắt, việc áp dụng công nghệ vào quản lý và vận hành khách sạn không chỉ giúp nâng cao hiệu quả kinh doanh mà còn đáp ứng tốt hơn nhu cầu của khách hàng. Một trang web quản lý khách sạn, với tính năng đa dạng và tiện lợi, chính là một công cụ vô cùng cần thiết để gia tăng hiệu quả quản lý cũng như nâng cao trải nghiệm của cả người dùng và người quản lý.

## 1.2. Mục đích và ý nghĩa

### 1.2.1. Mục đích:

Xây dựng Website phục vụ cho nhu cầu đặt phòng khách sạn cũng như là quản bá nó. Người đặt phòng chỉ cần tốn ít thời gian để có thể tìm một căn phòng ưng ý ở nơi mà mình muốn, và họ cũng có thể xem các đánh giá của người khác về phòng đó. Những chủ khách sạn có thể đăng phòng của mình lên trên Website để quản bá và quản lý. Ngoài những chức năng tiện lợi ra thì Website cũng có hệ thống bảo mật an toàn cho khách hàng cũng như là nhưng chủ quản lý khách sạn.

### 1.2.2. Ý nghĩa:

Đề tài xây dựng và thiết kế trang web quản lý khách sạn mang ý nghĩa to lớn không chỉ đối với việc học tập và nghiên cứu mà còn đối với ngành công nghiệp khách sạn nói chung.

Website quản lý khách sạn giúp nâng cao hiệu quả hoạt động kinh doanh và quản lý trong các khách sạn, việc áp dụng công nghệ vào quản lý khách sạn không chỉ giúp tiết kiệm chi phí và thời gian mà còn tăng cường hiệu quả cạnh tranh của các khách sạn trên thị trường

Tạo trải nghiệm tốt cho khách hàng với tính năng tìm và đặt phòng trực tuyến, khách hàng có thể dễ dàng tìm và đặt phong theo nhu cầu của mình một cách nhanh chóng và tiện lợi. Họ có thể xem trước thông tin một cách chi tiết và những đánh giá từ những khách hàng trước đó, giúp họ có quyết định đúng hơn.

Hệ thống trang Web còn được thiết kế với các biện pháp bảo mật, bảo vệ thông tin các nhân và dữ liệu giao dịch của khách hàng cũng như chủ khách sạn.

## 1.3. Lý do chọn đề tài:

Hiện nay, nhu cầu đặt phòng và quản bá khách sạn ngày càng lên cao cùng với sự phát triển của công nghệ, thế nên có một Website giúp việc tiếp cận với người dùng một cách dễ dàng và cũng như quản lý khách sạn là vô cùng cần thiết. Nhằm đáp ứng nhu cầu này, em đã chọn ra đề tài xây dựng và thiết kế trang Web CoutingStar Hotel.

## 1.4. Mục tiêu của đề tài

Xây dựng một hệ thống hỗ trợ cho việc đặt phòng và quản lý khách sạn của các chủ khách sạn trên Internet:

- Nhanh chóng, tiện lợi và hiệu quả.

- Đảm bảo chính xác và dễ dàng sử dụng cho khách hàng và người quản lý khách sạn.

- Dễ dàng trong việc sử dụng.

## 1.5. Nội dung chính của đề tài:

Xây dựng hệ thống đăng ký, đặt phòng, bình luận cho khách hàng và tìm kiếm phòng dựa trên nhu cầu của họ.

Xây dựng hệ thống đăng khách sạn, phòng cũng như quản lý của những chủ khách sạn.

Tạo một giao diện bắt mắt, phù hợp với thị hiếu cũng như cảm quan thẩm mỹ hiện nay.

## 1.6. Nhiệm vụ phải thực hiện:

### 1.6.1 Mục tiêu:

Nắm bắt được các yêu cầu cụ thể của khách hàng cũng như của những chủ khách sạn mà đưa ra những thiết kế hệ thống phù hợp.

Lên kế hoạch và phân bổ công việc một cách có hệ thống.

Hoàn thiện dự án trong thời gian 5 tuần.

### 1.6.2 Kết quả:

Một Website đặt phòng khách sạn vô cùng tiện lợi nhanh chóng, giúp người dùng có thể tìm được một căn phòng ưng ý trong một khoảng thời gian nhanh chóng trên nền giao diện đẹp mắt. Các chủ khách sạn có thể đăng khách sạn và các phòng của mình lên trên Website để quản bá cũng như thúc đẩy kinh doanh.

## 1.7. Phân công việc trong từng giai đoạn:

### 1.7.1. Nội dung công việc:

Công việc 1: Tiến hành khảo sát nhu cầu hiện nay của những đối tượng người dùng để liên kế hoạch thiết kế Database, phác thảo UI.

Công việc 2: Lựa ngôn ngữ lập trình và Framework dựa vào những ý tưởng đang có.

Công việc 3: Thống nhất ngôn ngữ lập trình, Framework và các ý tưởng để tiến hành xây dựng nền móng cơ bản cho Website.

Công việc 4: Viết Website và xây dựng những tính năng cốt lõi.

Công việc 5: Chỉnh sửa, nâng cấp và vá lỗi.

Công việc 6: Thử nghiệm và hoàn thiện Website

### 1.7.2. Lịch trình thực hiện:

Tuần 1: Tiến hành khảo sát, liên ý tưởng, tham khảo ý kiến. Lựa ngôn ngữ lập trình và Framework phù hợp, lên danh sách những loại công nghệ hỗ trợ trong quá trình làm.

Tuần 2: Tiến hành việc code tính năng cốt lõi đồng thời thử nghiệm, bổ xung thêm bảo mật và phân quyền tại thời điểm này.

Tuần 3: Bắt đầu trang trí UI và cải thiện UX (nếu như tuần 2 đã làm các UI cơ bản rồi). Lập trình thêm các tính năng nâng cao ở Back-end.

Tuần 4: Thử nghiệm API, cải thiện độ ổn định của Website, dò tìm các lỗi còn lại.

Tuần 5: thử nghiệm và nộp báo cáo.

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## 2.1. Môi trường lập trình (bỏ)

### 2.1.1. Giới thiệu về Visual Studio Code

Visual Studio Code hay còn được gọi là Vscode là một phần mềm cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế Website một cách nhanh chóng.

Ngoài việc đi kèm các hỗ trợ đa dạng các Extension (các tính năng mở rộng cài thêm) thì Vscode còn đi kèm với Git, có Syntax Highlighting rất thuận tiện cho việc viết code. Tuy đây là một trình soạn code tương đối nhẹ nhưng vô cùng mạnh mẽ.

Với bản chất gọn nhẹ, giao diện tối giản, thân thiện, Vscode còn cung cấp tích hợp hỗ trợ cho các ngôn gnwx lập trình Front-end như HTML, CSS, Javascript, các Frameworks và thư viện phổ biến như React, Angular, và Vue.js cũng được hỗ trợ vô cùng tốt thông qua các Extension mở rộng. Đây là một lựa chọn tuyệt vời cho việc phát triển Front-end.

### 2.1.2. Giới thiệu về Intellij Idea

Intellij IDEA là một trình soạn code rất mạnh mẽ, tuy nhiên nó có tính phí và ngốn nhiều phần cứng máy tính. Intellij IDEA cung cấp cho người dùng một giao diện tốt và nhiều tính năng tiên tiến giúp đáp ứng nhu cầu phát triển phần mềm đa dạng.

Với các lợi ích như: tập trung quản lý tác vụ, phất triển và gỡ lỗi mã một cách linh hoạt, hỗ trợ nhiều ngôn ngữ và công nghệ, duy trì chất lượng mã tự động, tạo mã sạch và nhanh chóng,...Thì đây là một lựa chọn vô cùng tuyệt vời.

Được thiết kế đặc biệt để hỗ trợ Java và các ngôn ngữ chạy trên JVM như Kotlin, Scala và Groovy. IntelliJ IDEA cung cấp các công cụ tích hợp để làm việc với cơ sở dữ liệu, bao gồm các tính năng như kết nối, truy vấn, và quản lý cơ sở dữ liệu. Điều này giúp dễ dàng làm việc với dữ liệu trong quá trình phát triển backend.

## 2.2. Framework và thư viện

### 2.2.1 Giới thiệu về Reactjs

ReactJS được phát triển bởi Facebook và được giới thiệu lần đầu tiên vào năm 2011 nhằm cải thiện tốc độ và hiệu suất của ứng dụng web.

Với những tiện ích cho việc phát triển ứng dụng web, bao gồm:

- Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.

- Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

- Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.

- Hỗ trợ tốt cho SEO: ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.

- Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

### 2.2.2. Giới thiệu về Spring boots

Spring Boot là một framework Java được sử dụng để xây dựng các ứng dụng và dịch vụ web dễ dàng và nhanh chóng. Nền tảng cung cấp các cấu hình mặc định cho một số thư viện và bộ công cụ hỗ trợ xây dựng, triển khai, quản lý ứng dụng Spring-based.

Cách Spring Boot hoạt động nhằm tối ưu hóa quy trình phát triển ứng dụng Java. Điều này sẽ giúp nhà phát triển tập trung vào việc xây dựng tính năng chính của ứng dụng mà không cần phải lo lắng về cấu hình phức tạp.

Một số ưu điểm dưới đây đã giúp Spring Boot trở thành sự lựa chọn phổ biến cho việc xây dựng các ứng dụng và dịch vụ web trong hệ sinh thái Java:

- Tối ưu hóa quá trình phát triển: Spring Boot cung cấp cấu hình mặc định thông minh và tự động, giúp giảm thiểu việc cấu hình thủ công và tối ưu quá trình phát triển ứng dụng Java.

- Tích hợp tốt: Spring Boot tích hợp tốt với nhiều công nghệ và thư viện khác trong hệ sinh thái Spring Framework. Nền tảng cho phép hệ thống dễ dàng tích hợp các module và dịch vụ khác nhau mà không cần phải lo lắng về cấu hình phức tạp.

- Embedded server: Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow. Đây là công cụ không thể thiếu trong việc triển khai ứng dụng một cách đơn giản mà không cần cấu hình thêm bất kỳ máy chủ nào khác.

- Tự động cấu hình: Spring Boot sử dụng cơ chế cấu hình tự động thông minh, cho phép ứng dụng tự cấu hình dựa trên các thư viện và module được sử dụng.

- Quản lý phụ thuộc: Spring Boot cung cấp các công cụ quản lý phụ thuộc mạnh mẽ như Maven hoặc Gradle, giúp quản lý các phụ thuộc của ứng dụng một cách hiệu quả.

- Monitoring và quản lý: Spring Boot cung cấp các công cụ hỗ trợ giám sát và quản lý ứng dụng dễ dàng, bao gồm Spring Boot Actuator cho việc giám sát và quản lý ứng dụng.

### 2.2.3. Giới thiệu về Spring Security

Spring Security được phát triển bởi SpringSource (hiện thuộc Pivotal) và được xem là một trong những framework bảo mật phổ biến nhất cho ứng dụng Java. Spring Security được thiết kế theo kiến trúc plugin, cho phép tùy biến linh hoạt và dễ dàng theo nhu cầu của ứng dụng và được tích hợp sẵn với các thành phần khác của Spring Framework, như Spring Boot, Spring MVC, Spring Data, Spring Cloud, và Spring WebFlux.

Spring Security hoạt động theo mô hình client-server. Khi một client gửi một request đến server, server sẽ xác thực người dùng và phân quyền để đảm bảo rằng người dùng chỉ có thể truy cập vào những tài nguyên mà họ được phép truy cập.

Cơ chế hoạt động của Spring Security dựa trên cơ chế lọc (filter) và sự kiện (event) để can thiệp vào quá trình xử lý yêu cầu (request) và phản hồi (response) của ứng dụng web, tức là khi một yêu cầu được gửi đến ứng dụng web, nó sẽ được chuyển qua một chuỗi các bộ lọc (filter chain) do Spring Security quản lý. Mỗi bộ lọc có một nhiệm vụ cụ thể, như kiểm tra xác thực, kiểm tra phân quyền, điều hướng đến trang đăng nhập hoặc đăng xuất, xử lý các lỗi bảo mật.

Nếu một yêu cầu không thỏa mãn các điều kiện bảo mật của ứng dụng, Spring Security sẽ sinh ra một sự kiện (event) để thông báo cho ứng dụng biết. Ứng dụng có thể lắng nghe và xử lý các sự kiện này theo ý muốn, ví dụ như ghi log, gửi email hoặc hiển thị thông báo lỗi.

Ngược lại, nếu một yêu cầu được chấp nhận bởi Spring Security, nó sẽ được tiếp tục xử lý bởi ứng dụng web như bình thường. Khi ứng dụng web trả về một phản hồi cho yêu cầu, nó cũng sẽ được chuyển qua lại chuỗi các bộ lọc của Spring Security để áp dụng các thiết lập bảo mật cho phản hồi.

### 2.2.4. Giới thiệu về Hibernate

Hibernate là một thư viện ORM (Object Relational Mapping) mã nguồn mở giúp lập trình viên viết ứng dụng Java có thể map các objects (pojo) với hệ quản trị cơ sở dữ liệu quan hệ, và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cớ dữ liệu quan hệ.

Hibernate cung cấp các câu lệnh truy vấn tương tự SQL, HQL của Hibernate hỗ trợ đầy đủ các truy vấn đa hình như, HQL “hiểu” các khái niệm như kế thừa (inheritance), đa hình (polymorphysm), và liên kết (association).

Code sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là ta không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác… Do đó rất dễ để ta thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

Với những ứng dụng Java làm việc với cơ sở dữ liệu lớn hàng trăm triệu bản ghi, việc có sử dụng Lazy loading trong truy xuất dữ liệu từ database mang lại lợi ích rất lớn. Nó giống như việc ta có thể bẻ từng chiếc đũa của bó đũa to thay vì bẻ cả bó đũa.

## 2.3 Giới thiệu mô hình MVC

MVC là viết tắt của cụm từ “Model-View-Controller“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

Model (dữ liệu): Quản lí xử lí các dữ liệu.

View (giao diện): Nới hiển thị dữ liệu cho người dùng.

Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.

Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

MVC hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một mô hình MVC được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên View và một developer khác có thể làm việc với Controller để tạo logic nghiệp vụ cho ứng dụng web đó.

## 2.4. Phân tích đề tài với UML

### 2.4.1. Sơ đồ use case tổng quát

### 2.4.2. Sơ đồ use case chức năng

## 2.5. Mô hình quan hệ Database

## 

## 2.6. Database

## 

# TỔNG KẾT

**Kết quả nghiên cứu, tìm hiểu:**

Báo cáo được thực hiện trong một thời gian vừa đủ nhưng vì độ rộng nên vẫn bị giới hạn trong một phạm vi nhất định, song nó vẫn đạt được một số kết quả như sau:

Biết được các requirements.

Có các model use case.

Mô tả system architect một cách rõ ràng.

Trình bày Data model

Thể hiện rõ System user interface

**Các hạn chế và hướng phát triển:**

Hạn chế:

Trong khuôn khổ của bài báo cáo này, việc phân tích hệ thống thông tin bị hạn chế lại ở một quy mô nhất định để tránh quá tải.

Do khối lượng kiến thức lẫn kỹ năng cần thiết để thực hiện báo cáo này là rất nhiều nên báo cáo này sẽ dừng lại ở những thứ cơ bản.

Hiện tại kiến thức và trình độ bản thân còn hạn chế nên báo cáo không thể tránh khỏi những thiếu sót. Em mong rằng sẽ nhận được những ý kiến và đóng góp của thầy để bài báo cáo hoàn thiện hơn, có ích hơn trong thực tế.

Hướng phát triển:

Hướng phát triển nghiên cứu trong tương lai:

Phát triển đồ án ở quy mô rộng hơn kèm theo đó là sự hoàn chỉnh lẫn kiến thức chưa được thể hiện trong báo cáo này.

Phát triển báo cáo thành một phần mềm hoàn chỉnh.

# TÀI LIỆU THAM KHẢO

1. Trang Vũ. (2024). ReactJS là gì? Tất tần tật những điều căn bản về ReactJS. <https://stringee.com/vi/blog/post/reactJS-la-gi>
2. Lê Trí Dũng. (Không rõ thời điểm đăng). Hibernate là gì? Sao phải dùng nó thay JDBC? <https://topdev.vn/blog/hibernate-la-gi-sao-phai-dung-no-thay-jdbc/>
3. Nam Thang. (2023). Spring Security là gì? Cơ chế hoạt động của Spring Security. <https://200lab.io/blog/co-che-hoat-dong-cua-spring-security/>
4. Cao Lê Viết Tiến. (không rõ thời điểm đăng). Tìm hiểu mô hình MVC là gì? Ví dụ về cách sử dụng mô hình MVC. <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>