

**TRƯỜNG ĐẠI HỌC LẠC HỒNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**BÁO CÁO**

**NGHIÊN CỨU KHOA HỌC**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE QUẢN LÝ CUỘC THI**

**<Tên\_SV>**

**…, THÁNG 5/2024**



**TRƯỜNG ĐẠI HỌC LẠC HỒNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**BÁO CÁO**

**NGHIÊN CỨU KHOA HỌC**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE QUẢN LÝ CUỘC THI**

**SVTH: <Tên\_SV>**

**GVHD: <Tên\_GV>**

**…, THÁNG 5/2024**

**MỤC LỤC**

**DANH MỤC HÌNH**

**DANH MỤC BẢNG**

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các *thầy cô của* *trường Đại học Lạc Hồng* và đặc biệt là các *thầy cô khoa Công nghệ thông tin của trường* đã tạo điều kiện cho em hoàn thành đồ án tốt nghiệp này.

Em cũng xin chân thành cảm ơn *cô …………* đã nhiệt tình hướng dẫn chỉ bảo em những kiến thưc bài học, cũng như những kiến thức thực tế trong suốt quá trình thực hiện đề tài.

Mặc dù đã cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiết sót. Em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý thầy cô và các bạn.

***Em xin chân thành cảm ơn!***

# CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI

* 1. **Lý do chọn đề tài:**

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc xây dựng một website quản lý cuộc thi trở thành một lựa chọn hết sức hợp lý và cần thiết một trong những lý do chính là nhu cầu thực tiễn. Các cuộc thi, đặc biệt là những cuộc thi có quy mô lớn, thường gặp khó khăn trong việc quản lý thông tin thí sinh, cập nhật kết quả, và duy trì liên lạc giữa ban tổ chức và các thí sinh. Website quản lý cuộc thi sẽ giúp tự động hóa và tối ưu hóa các quy trình này, tiết kiệm thời gian và nguồn lực, đồng thời nâng cao trải nghiệm người dùng.

Bên cạnh đó, việc sử dụng công nghệ giúp giảm bớt chi phí tổ chức và in ấn tài liệu, đồng thời đảm bảo tính minh bạch và công bằng trong quá trình chấm điểm và đánh giá. Điều này không chỉ giúp tăng cường uy tín của cuộc thi mà còn thúc đẩy sự tin tưởng từ phía thí sinh.

Hơn nữa, hệ thống quản lý cuộc thi trực tuyến có khả năng mở rộng và ứng dụng rộng rãi. Nó có thể được điều chỉnh để phù hợp với nhiều loại cuộc thi khác nhau, từ học thuật, nghệ thuật đến thể thao, đảm bảo tính bền vững và linh hoạt. Sau khi hoàn thành, sản phẩm có thể được áp dụng trực tiếp vào các cuộc thi của trường học, tổ chức hoặc các sự kiện cộng đồng, mang lại giá trị thực tiễn cao.

Đề tài này cũng là một cơ hội tuyệt vời để em có thể phát triển kỹ năng công nghệ của mình. Trong quá trình thực hiện, em sẽ được áp dụng và nâng cao các kiến thức về lập trình web, thiết kế giao diện người dùng, quản lý cơ sở dữ liệu, và bảo mật thông tin. Ngoài ra, việc xây dựng website còn giúp em nắm bắt và áp dụng các công nghệ mới, phù hợp với xu hướng chuyển đổi số đang diễn ra mạnh mẽ.

Tóm lại, chọn đề tài "Xây dựng website quản lý cuộc thi" không chỉ đáp ứng nhu cầu thực tế và mang lại nhiều lợi ích về kinh tế và xã hội mà còn giúp em phát triển kỹ năng công nghệ và mở rộng ứng dụng trong nhiều lĩnh vực khác nhau. Đây là một đề tài có tính khả thi cao và mang lại nhiều giá trị thiết thực cho cộng đồng.

* 1. **Khảo sát đề tài:**

### *Khảo sát thực trạng:*

Trong thời đại ngày nay, khi sự cạnh tranh và yêu cầu về tổ chức các cuộc thi ngày càng tăng, việc xây dựng một website quản lý cuộc thi trở nên ngày càng quan trọng và đặc biệt hữu ích. Để đáp ứng đúng những nhu cầu thực tế của cả thí sinh và ban tổ chức, các chức năng chính của nền tảng này cần phải được thiết kế và phát triển một cách linh hoạt và hiệu quả.

*Danh Sách Cuộc Thi:*

Chức năng này không chỉ đơn thuần là danh sách mà còn là bộ lọc thông tin quan trọng để thí sinh có thể nhanh chóng tìm kiếm và lựa chọn cuộc thi phù hợp với khả năng và sở thích của mình. Điều quan trọng là phải đảm bảo danh sách cuộc thi được cập nhật thường xuyên và cung cấp thông tin chính xác, giúp thí sinh có cái nhìn toàn diện về các cuộc thi khác nhau.

*Chi Tiết Cuộc Thi:*

Chức năng này đóng vai trò quyết định đối với thí sinh khi xem xét tham gia cuộc thi. Thông tin chi tiết về yêu cầu tham gia, mô tả cuộc thi, và giải thưởng không chỉ giúp thí sinh hiểu rõ về cuộc thi mà còn tạo niềm tin và sự hứng thú.

*Tìm Kiếm Cuộc Thi:*

Tìm kiếm cuộc thi là cầu nối giữa nhu cầu tìm kiếm của thí sinh và các cuộc thi được tổ chức. Các tiêu chí lọc như loại hình thi, địa điểm, và thời gian diễn ra là quan trọng để thí sinh có thể dễ dàng tìm kiếm và xác định cuộc thi phù hợp với nhu cầu của họ.

*Đăng Ký Tham Gia:*

Chức năng đăng ký tham gia được thiết kế một cách thuận lợi để thí sinh có thể nhanh chóng và dễ dàng đăng ký tham gia các cuộc thi mong muốn. Giao diện thân thiện và tiến trình đơn giản giúp tăng cường trải nghiệm người dùng và khuyến khích sự tương tác.

*Quản Lý Cuộc Thi Đã Tham Gia:*

Chức năng này là quan trọng để thí sinh theo dõi tiến trình tham gia của họ. Cập nhật trạng thái của cuộc thi và thông báo là những tính năng quan trọng để duy trì sự minh bạch và hiệu quả trong quá trình tổ chức cuộc thi.

*Thông Tin Cuộc Thi:*

Đây là chức năng cung cấp thông tin về cuộc thi, bao gồm mục đích, quy định, và các yêu cầu cụ thể. Thí sinh thường xem xét thông tin này để đảm bảo rằng họ hiểu rõ về cuộc thi và chuẩn bị tốt nhất cho phần tham gia của mình.

*Quản Lý Cuộc Thi và Thêm Cuộc Thi (Admin Duyệt):*

Chức năng này là quan trọng từ góc độ của ban tổ chức. Quản lý cuộc thi cung cấp cho họ khả năng theo dõi các cuộc thi đã đăng và tình trạng của chúng. Thêm cuộc thi (phải được admin duyệt) đảm bảo rằng thông tin về cuộc thi được kiểm duyệt trước khi được công bố, giúp đảm bảo chất lượng thông tin và tổ chức.

### *3.1.2. Yêu cầu của hệ thống:*

* Phân quyền người dùng: Quản trị viên, giảng viên và thí sinh
* Chức năng của thí sinh:
* Đăng nhập, đăng ký và cập nhật thông tin cá nhân của tài khoản
* Xem thông tin các cuộc thi
* Nộp bài thi lên hệ thống
* Xem kết quả các bài thi trong danh sách cuộc thi
* Chức năng của giảng viên:
  + Đăng nhập, đăng ký, cập nhật thông tin cá nhân
    - * Thêm và cập nhật các cuộc thi, cùng với tiêu chí của mỗi cuộc thi
      * Chấm điểm các bài thi của thí sinh gửi về
* Chức năng của quản trị viên:
* Đăng nhập và cập nhật thông tin tài khoản tài khoản
* Phân quyền và thay đổi thông tin các tài khoản trên hệ thống
* Có thể thêm và cập nhật các thông tin trên hệ thống
* Quản lý thông tin các tài khoản, trường, lớp, tiêu chí, cuộc thi, kết quả cuộc thi
  1. **Phương pháp thực hiện:**
  2. **Ý nghĩa của đề tài:**

# CHƯƠNG 2: KIẾN THỨC NỀN TẢNG

* 1. **Cơ sở lý thuyết:**
     1. *Ngôn ngữ lập trình:*

*Javascript* là một ngôn ngữ lập trình của HTML và ứng dụng Web. Nó được sử dụng phổ biến nhất, như một phần của các trang web. Chúng cho phép client-server script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

Một số ưu điểm của JavaScript:

* Tiêt kiệm băng thông máy chủ:

Javascript sử dụng trong thiết kế website có thể giúp tiết kiệm băng thông máy máy chủ vô cùng tốt bởi chúng được chạy trực tiếp trên máy của người dùng. Gia tăng hiệu suất, kéo dài tuổi thọ máy chủ và tiết kiệm diện tích băng thông.

* Linh hoạt vận hành, tương thích tốt:

Để thực thi, trình duyệt web chỉ cần tải file JavaScript về máy chủ từ một tên miền riêng biệt hoặc nhúng thẳng vào file HTML và chạy mà không cần tải thêm. Javascript có thể chạy được trên hầu hết các trình duyệt phổ biến hiện nay.

Hầu hết các trình duyệt web đều có tích hợp sẵn trình thông dịch ngôn ngữ JavaScript, đảm bảo người dùng có thể sử dụng ngay mà không mất công tải về trình duyệt, giảm tỷ lệ thực thi các chức năng của JavaScript.

* Dễ dàng xử lý vấn đề:

Với lợi thế có nhiều trình duyệt hỗ trợ nên việc kiểm tra và xử lý vấn đề dễ dàng hơn. Đa số các trình duyệt web hiện nay đều có cài sẵn các công cụ xử lý lỗi Javascript trong bảng điều khiển trình duyệt. Javascript sở hữu cấu trúc dễ đọc với các nguyên tắc rõ ràng nên việc xác định và gỡ lỗi trở nên đơn giản hơn.

* Nâng cao trải nghiệm người dùng:

Website sử dụng ngôn ngữ lập trình giúp thu hút người dùng bởi những tính năng ưu việt, hình ảnh sinh động hấp dẫn, chức năng lướt và tương tác nhanh nhạy. Nhờ những tính năng đó, người dùng cảm thấy thoải mái, thích thú và tin dùng trang web hơn.

* + 1. *Hệ quản trị cơ sở dữ liệu:*

*Cơ sở dữ liệu (Database)* là một tập hợp các dữ liệu có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

*Hệ quản trị cơ sở dữ liệu (Database Management System)* là một phần mềm để lưu trữ và truy xuất dữ liệu của người dùng trong khi xem xét các biện pháp bảo mật thích hợp. Nó bao gồm một nhóm các chương trình thao tác cơ sở dữ liệu. Hệ quản trị cơ sở dữ liệu chấp nhận yêu cầu dữ liệu từ một ứng dụng và hướng dẫn hệ điều hành cung cấp dữ liệu cụ thể. Trong các hệ thống lớn, nó giúp người dùng và phần mềm bên thứ ba lưu trữ và truy xuất dữ liệu.

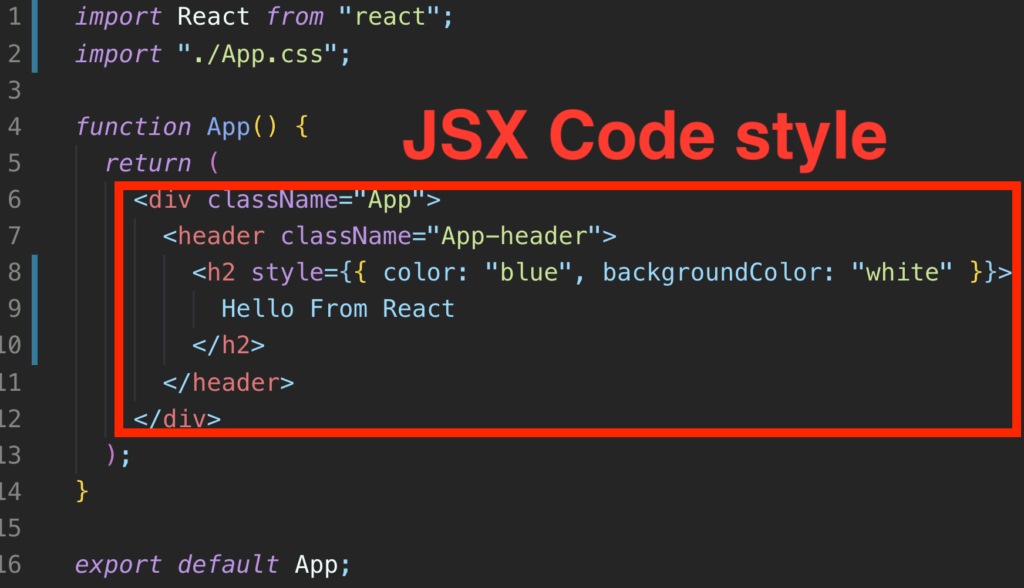
* 1. **Công cụ - công nghệ sử dụng:**
     1. *Ngôn ngữ lập trình ReactJS:*

*ReactJS* là một opensource được phát triển bởi Facebook, vào năm 2013. Đây là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần website. Một trong những điểm nổi bật của thư viện này là việc render dữ liệu không chỉ thực hiện phía Server, mà còn thực hiện được phía Client.

ReactJS là một thư hiện Javascript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập tình ứng dụng front-end, lập trình viên thường phải làm việc chính trên 2 thành phần: UI và xử lý tương tác người dùng. UI là tập hợp những thành phần nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ thanh tìm kiếm, nút nhấn… Trước khi có ReactJS, lập trình viên thường gặp khó khăn rất nhiều khi sử dụng Javascript thuần và Jquery để xây dựng UI.

Để tăng tốc quá trình phát triển và giảm thiểu rủi ro xảy ra trong khi coding, React còn cung cấp khả năng Reusable Code(tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng: JSX và Virtual DOM.

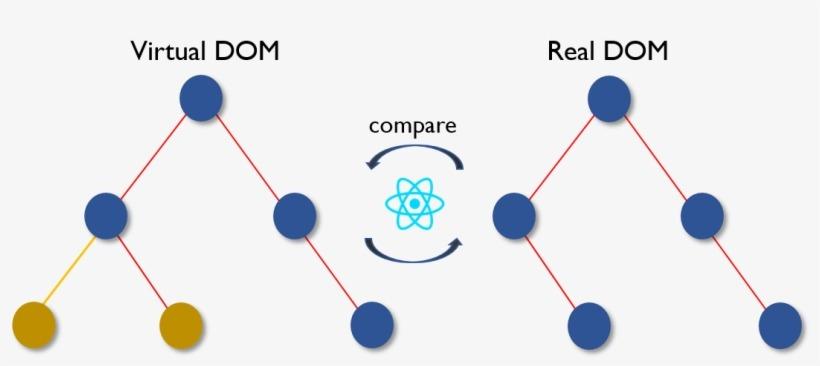
*JSX* *(nói ngắn gọn là Javascript extension)* là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và ReactJS browser hỗ trợ toàn bộ những trình duyệt web hiện đại, có thể sử dụng JSX trên bất kỳ trình duyệt nào đang làm việc.



*JSX Code Style*

Nếu không sử dụng ReactJS (và JSX), website sẽ sử dụng HTML để cập nhật lại cây DOM(quá trình thay đổi diễn ra tự nhiên trên trang mà người dùng không cần tải lại trang). Cách này sẽ ổn với các trang web nhỏ, đơn giản, static, nhưng với các website lớn, đặc biệt thiên về xử lý tương tác, điều này sẽ làm ảnh hưởng performance website nghiêm trọng vì toàn bộ cây DOM phải reload lại mỗi lần người dùng nhấn tính năng yêu cầu tải lại trang.

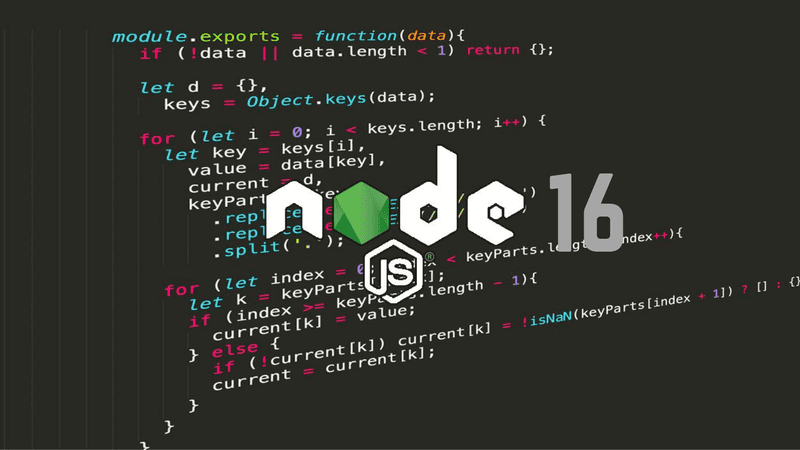
Tuy nhiên, nếu sử dụng JSX sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS đã khởi tạo Virtual DOM(DOM ảo). Virtual DOM là bản copy của Real DOM(DOM thật) trên trang đó, và ReactJS sử dụng bản copy để tìm kiếm đúng phần mà DOM thật cần cập nhật khi bất kỳ sự kiện nào khiến thành phần trong nó thay đổi(như việc người dùng nhấn vào một nút).



*So sánh Virtual DOM và Real DOM*

* + 1. *Ngôn ngữ lập trình NodeJS:*

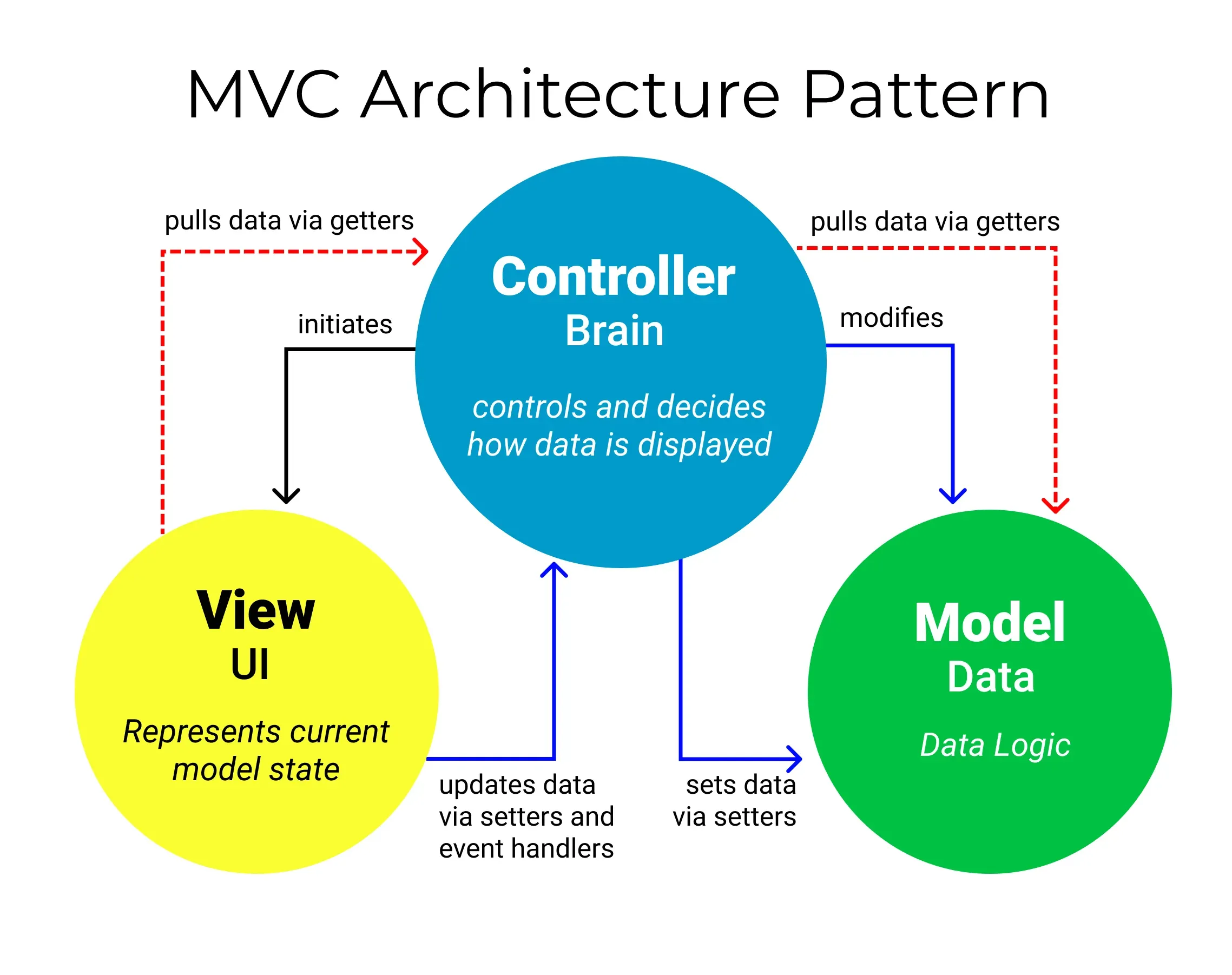
*NodeJS* được tạo ra bởi Ryan Dahl năm 2009, tạo nên môi trường hỗ trợ chạy JavaScript runtime, đa nền tảng và có mã nguồn mở. Đây là môi trường hoàn hảo để chạy các ứng dụng, web ngoài trình duyệt mà người dùng đang sử dụng. Đây cũng được coi là giải pháp thiết yếu giúp ứng dụng sử dụng dữ liệu hiệu quả nhờ mô hình event driven (mô hình hướng dự kiện) một cách không đồng bộ.



*NodeJS*

Cách hoạt động của NodeJS:

* Khi sử dụng Node JS các câu lệnh sẽ được đưa vào queue, hàng đợi. Đồng thời chạy từ trên xuống dưới.
* Lúc này Event loop lấy 1 tác vụ ở queue cho vào stack, và trước đó stack hoàn toàn trống. Điều này có thể hiểu rằng câu lệnh đầu tiên – console.log sẽ được stack xử lý và đưa ra “caulenh1”
* Sau khi stack xử lý xong, tác vụ đầu sẽ được lấy ra và tiếp tục đưa tác vụ tiếp theo vào để xử lý tiếp. Cụ thể có thể là setTimeout(function(){ console.log(“cau lenh 2”); },0);. Lúc này có thể thấy SetTimeout trở thành 1 hàm tính toán thời gian nằm trong web AIPS và đưa vào web AIPs để đợi tác vụ tiếp theo. Nó sẽ được đưa trở lại hàng đợi khi queue đã trống.
* Tương tự khi thực hiện tác vụ 1, tác vụ 3 sẽ in câu lệnh và hàng đợi ở trạng thái rỗng. Khi đó tác vụ 2 sẽ được đưa trở lại queue, tiếp đến sang stack để đi qua xử lý.
  + 1. *Mô hình MVC:*



*Mô hình MVC*

*Mô hình MVC (Model – View – Controller)* là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp tách ứng dụng thành 3 phần: Model (xử lý dữ liệu), View (hiển thị) và Controller (điều khiển). Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với nhau.

* View:

View là một phần của ứng dụng chịu trách nhiệm cho việc trình bày dữ liệu. Thành phần này được tạo bởi dữ liệu thu thập từ dữ liệu mô hình, và giúp người dùng có cái nhìn trực quan về trang web, cũng như ứng dụng.

View cũng đại diện cho dữ liệu từ các cuộc trò chuyện, sơ đồ và bảng. Ví dụ: bất kỳ View nào cũng sẽ có bao gồm tất cả các thành phần giao diện người dùng như nút bấm, menu, khung nhập...

* Model:

Model của kiến trúc MVC là thành phần chính đảm nhiệm chức năng lưu trữ dữ liệu và các bộ phận logic liên quan của toàn bộ ứng dụng.

Model chịu trách nhiệm cho các thao tác dữ liệu giữa Controller hoặc bất kỳ logic nghiệp vụ liên quan nào khác như cho phép xem, truy xuất dữ liệu...

* Controller:

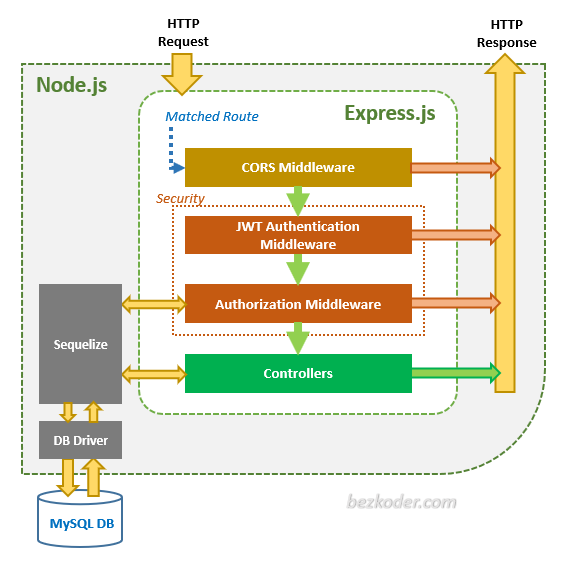
Controller xử lý tương tác người dùng của ứng dụng. Nó xử lý dữ liệu đầu vào từ bàn phím và chuột của người dùng, sau đó thông báo tới View và Model.

 Controller gửi các lệnh tới Model để thay đổi trạng thái của Model. Controller cũng gửi các lệnh tương tự tới View để thực hiện các thay đổi về giao diện.

Lý do nên áp dụng mô hình MVC:

* Cải thiện thời gian và quy trình lập trình: Nếu áp dụng mô hình MVC để sử dụng và lập trình trên bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên View, cùng thời gian với lập trình viên khác làm việc với Controller để xây dựng logic nghiệp vụ. Vì vậy, thời gian lập trình khi ứng dụng mô hình MVC có thể được hoàn thành gấp 3 lần so với các mô hình lập trình khác.
* Khả năng cung cấp nhiều chế độ View: Khi sử dụng mô hình MVC, người sử dụng có thể tạo nhiều View cho chỉ một mô hình.
* Những sửa đổi trên website không ảnh hưởng đến toàn bộ mô hình MVC: Với bất kỳ trang web nào, giao diện người dùng thường có xu hướng thay đổi theo thời gian, như bố cục, màu sắc, font chữ… Nên việc xuất hiện thêm kiểu view mới trong MVC sẽ trở nên đơn giản bởi phần Model không phụ thuộc vào phần View.
* Model trong mô hình MVC sẽ trả về dữ liệu mà không cần định dạng: MVC pattern có khả năng trả về các dữ liệu mà không cần áp dụng bất kỳ định dạng cụ thể nào.
* Nền tảng MVC thân thiện và cải thiện SEO: Nền tảng MVC giúp phát triển các trang web trở nên thân thiện hơn với các bộ tìm kiếm. Khi sử dụng nền tảng này, bạn có thể dễ dàng thay đổi và phát triển các URL để cải thiện SEO, từ đó tạo ra nhiều lượt truy cập hơn.
  + 1. *Xử lý đăng nhập với JWT (Json Web Token):*

[*JSON Web Token (JWT)*](https://jwt.io/) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.



*Xử lý authentication với JWT trong NodeJS*

Xử lý đăng nhập với JWT thường bao gồm các bước sau:

* Người dùng nhập thông tin đăng nhập (tên đăng nhập và mật khẩu) vào form đăng nhập của ứng dụng web.
* Ứng dụng web gửi thông tin đăng nhập của người dùng đến server để xác thực.
* Server kiểm tra thông tin đăng nhập của người dùng. Nếu thông tin đăng nhập chính xác, server tạo ra một JWT chứa thông tin về người dùng và trả về cho ứng dụng web.
* Ứng dụng web nhận JWT và lưu trữ nó ở phía client-side, thông thường là trong cookie hoặc local storage.
* Mỗi lần người dùng truy cập các trang được bảo vệ, ứng dụng web gửi JWT lưu trữ ở phía client-side đến server để xác thực. Server kiểm tra JWT và nếu hợp lệ, cho phép người dùng truy cập các trang được bảo vệ.

Ngoài việc xác thực người dùng, JWT cũng có thể được sử dụng để cấp quyền truy cập cho người dùng, cho phép họ truy cập vào các tài nguyên cụ thể trên server. Điều này giúp giảm thiểu việc lưu trữ thông tin đăng nhập trên server và giảm chi phí truy vấn cơ sở dữ liệu.

Để xử lý đăng nhập với JWT, có thể sử dụng các thư viện và công cụ hỗ trợ như JSON Web Token (JWT) của Node.js hoặc Django REST Framework của Python. Các thư viện này cho phép tạo, xác thực và quản lý các JWT, giúp giảm thiểu thời gian và công sức lập trình viên cần bỏ ra để xử lý đăng nhập và ủy quyền trong ứng dụng của mình.

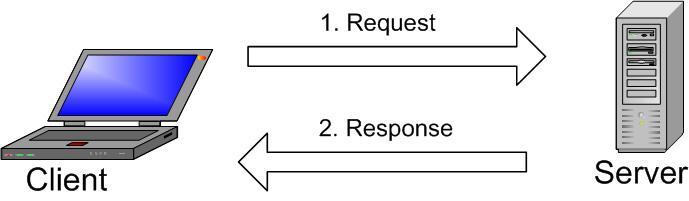
* + 1. *MySQL:*

*MySQL*là hệ quản trị cơ sở dữ liệu sử dụng mã nguồn mở phổ biến nhất thế giới và được ưa chuộng trong quá trình phát triển web, ứng dụng... Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có CSDL trên Internet.

Ưu điểm của MySQL:

* Dễ sử dụng: MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và có thể hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
* Đa tính năng: MySQL hỗ trợ rất nhiều chức năng được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
* Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
* Nhanh chóng: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.
* Độ bảo mật cao:  MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều tính năng bảo mật, ngay cả ở cấp cao.

Quy trình hoạt động khá đơn giản mỗi Client(máy khách) yêu cầu dữ liệu từ Server(máy chủ), từ Server sẽ phản hồi lại các dữ liệu đó. Ví dụ khi người dùng truy cập vào một trang web nào đó các Request sẽ gửi tới Server, lúc này Server sẽ nhận được Request và bắt đầu gửi dữ liệu quay lại máy người dùng để hiển thị thông tin trang web đó lên.



*Mô hình hoạt động của MySQL*

Một số lý do chính khiến MySQL trở nên quan trọng:

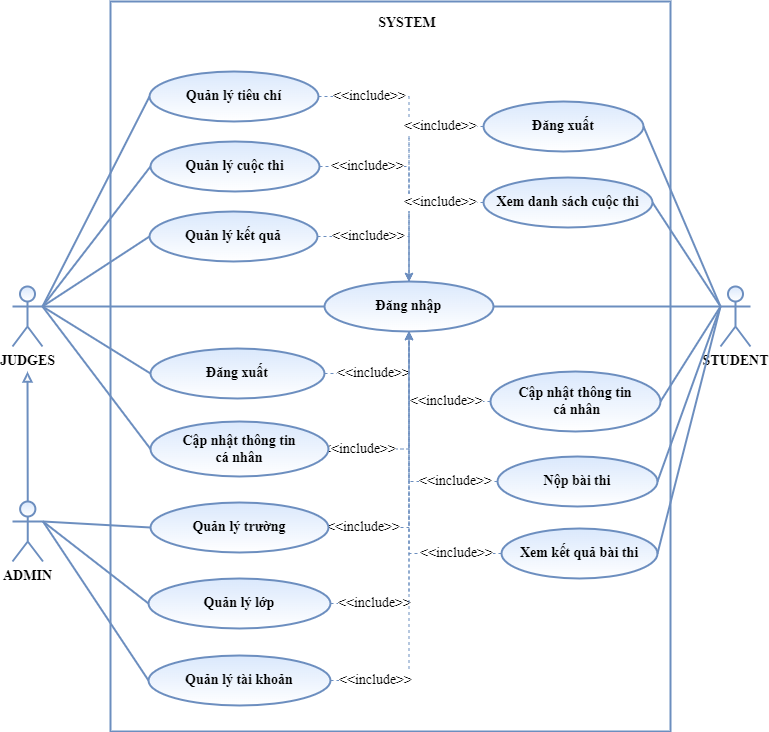
* Linh hoạt và dễ dùng: Có thể sửa source code để đáp ứng nhu cầu sử dụng mà không phải thanh toán thêm chí phí nào, quá trình cài đặt cũng đơn giản.
* Hiệu năng cao: Bất kể lưu trữ dữ liệu lớn của các trang thương mại điện tử hoặc những hoạt động kinh doanh lớn liên quan đến công nghệ thông tin, MySQL cũng có thể đáp ứng.
* An toàn: An toàn dữ liệu luôn là vấn đề quan trọng nhất khi chọn phần mềm RDBMS. Với hệ thống phân quyền truy cập và quản lý tài khoản, MySQL đặt tiêu chuẩn bảo mật rất cao.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU

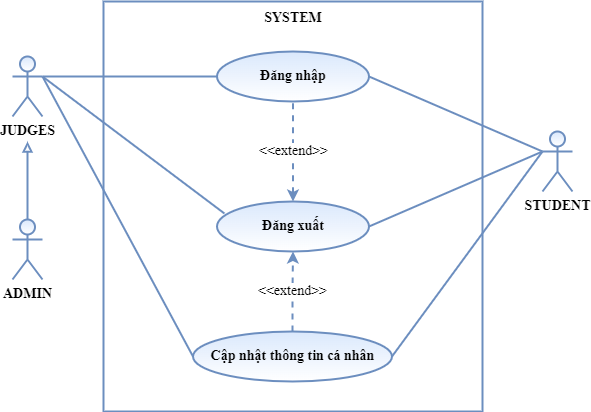
* 1. **Use Case của hệ thống:**
* *Các tác nhân của hệ thống:*

**

* *Use Case tổng quát:*

****

* *Use Case hệ thống:*

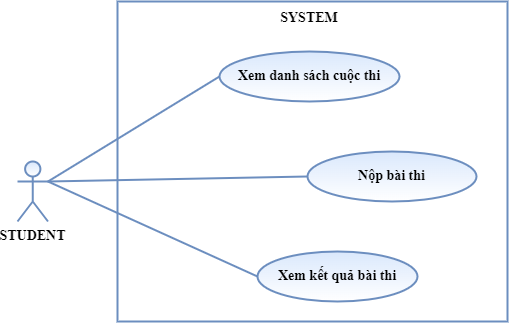
****

|  |  |
| --- | --- |
| Use Case | Đăng nhập |
| Tác nhân | Quản trị viên, giảng viên và thí sinh |
| Mô tả ngắn gọn | Người dùng đăng nhập vào hệ thống, truy cập hệ thống với đúng quyền của tài khoản |
| Tiền điều kiện | Use Case sử dụng khi người dùng muốn đăng nhập vào hệ thống |
| Chuỗi sự kiện chính | 1. Người dùng chọn đăng nhập 2. Hệ thống yêu cầu nhập tài khoản và mật khẩu 3. Người dùng nhập tài khoản và mật khẩu 4. Hệ thống kiểm tra tên và mật khẩu vừa nhập: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 3 5. Lấy quyền truy cập tài khoản 6. Hiển thị giao diện tương ứng với quyền của tài khoản |
| Ngoại lệ | 3. Người dùng chọn Hủy  4. Hệ thống hủy thao tác, và trả về trang chủ |
| Điều kiện đầu ra | Người dùng truy cập hệ thống với đúng quyền của tài khoản đã đăng nhập |

|  |  |
| --- | --- |
| Use Case | Đăng xuất |
| Tác nhân | Quản trị viên, giảng viên và thí sinh |
| Mô tả ngắn gọn | Người dùng muốn đăng xuất khỏi hệ thống |
| Tiền điều kiện | Use Case sử dụng khi người dùng muốn đăng xuất hệ thống |
| Chuỗi sự kiện chính | 1. Người dùng chọn đăng xuất 2. Hệ thống hủy lưu tài khoản 3. Hệ thống hiển thị giao diện trang đăng nhập |
| Ngoại lệ | Không có |
| Điều kiện đầu ra | Hủy lưu tài khoản người dùng, hiển thị trang đăng nhập |

|  |  |
| --- | --- |
| Use Case | Cập nhật thông tin cá nhân |
| Tác nhân | Quản trị viên, giảng viên và thí sinh |
| Mô tả ngắn gọn | Người dùng cập nhật thông tin tài khoản và thông tin cá nhân |
| Tiền điều kiện | Use Case sử dụng khi người dùng đăng nhập thành công hệ thống |
| Chuỗi sự kiện chính | 1. Người dùng chọn cập nhật thông tin cá nhân 2. Hệ thống lấy dữ liệu thông tin cá nhân của tài khoản theo ID 3. Hệ thống hiển thị form sửa 4. Nhà tuyển dụng nhập thông tin muốn sửa 5. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 3 6. Hệ thống cập nhật thông tin cá nhân vào cơ sở dữ liệu 7. Hệ thống thông báo cập nhật thông tin cá nhân thành công |
| Ngoại lệ | 3. Nhà tuyển dụng chọn Hủy  4. Hệ thống hủy thao tác, và trả về trang chủ |
| Điều kiện đầu ra | Thông tin cá nhân của người dùng được cập nhật vào cơ sở dữ liệu |

* *Use Case của tác nhân STUDENT:*

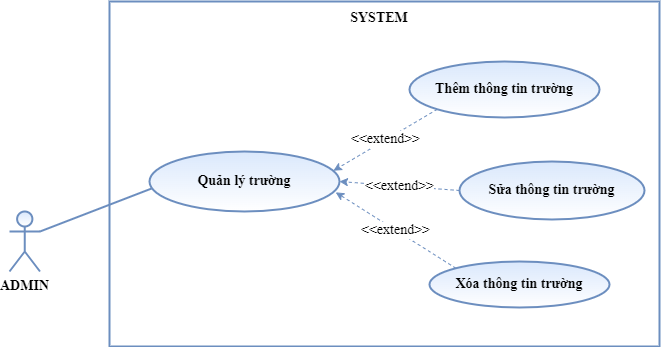
****

|  |  |
| --- | --- |
| Use Case | Xem danh sách cuộc thi |
| Tác nhân | Thí sinh |
| Mô tả ngắn gọn | Thí sinh muốn xem danh sách các cuộc thi của mình |
| Tiền điều kiện | Thí sinh đăng nhập thành công hệ thống |
| Chuỗi sự kiện chính | * 1. Thí sinh chọn xem danh sách cuộc thi   2. Hệ thống lấy danh sách cuộc thi theo mã thí sinh   3. Hệ thống hiển thị danh sách các cuộc thi |
| Ngoại lệ | Không có |
| Điều kiện đầu ra | Danh sách các cuộc thi của thí sinh hiển thị trên màn hình |

|  |  |
| --- | --- |
| Use Case | Nộp bài thi |
| Tác nhân | Thí sinh |
| Mô tả ngắn gọn | Thí sinh thực hiện nộp bài thi tương ứng với mỗi cuộc thi khác nhau |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền thí sinh |
| Chuỗi sự kiện chính | * 1. Thí sinh đăng nhập hệ thống, chọn cuộc thi muốn nộp bài   2. Hệ thống hiển thị form nộp bài   3. Thí sinh nhập các thông tin và thêm file đính kèm bài làm, nhấn nút Nộp bài   4. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 3.   5. Hệ thống lưu thông tin bài làm theo từng cuộc thi tương ứng vào cơ sở dữ liệu   6. Hệ thống thông báo thêm bài làm thành công |
| Ngoại lệ | 5. Thí sinh chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang danh sách cuộc thi |
| Điều kiện đầu ra | Thông tin nộp bài của thí sinh được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xem kết quả bài thi |
| Tác nhân | Thí sinh |
| Mô tả ngắn gọn | Thí sinh muốn xem chi tiết kết quả bài thi |
| Tiền điều kiện | Thí sinh đăng nhập thành công hệ thống |
| Chuỗi sự kiện chính | * 1. Thí sinh chọn bài thi muốn xem kết quả   2. Chọn xem chi tiết   3. Hệ thống lấy chi tiết kết quả bài thi theo ID   4. Hệ thống hiển thị trang chi tiết bài thi |
| Ngoại lệ | Không có |
| Điều kiện đầu ra | Chi tiết bài thi thí sinh muốn xem hiển thị trên màn hình |

* *Use Case quản lý trường:*

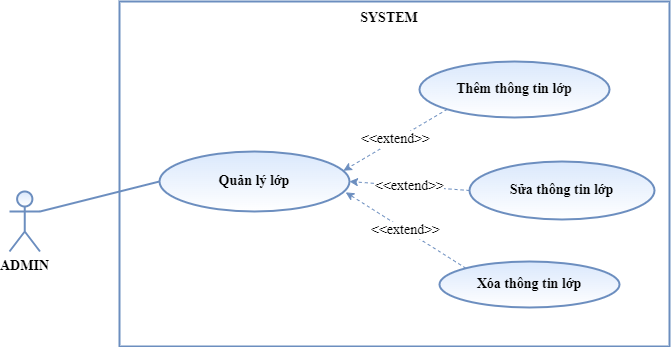
****

|  |  |
| --- | --- |
| Use Case | Thêm thông tin trường |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện thêm các trường mới |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | * 1. Quản lý đăng nhập hệ thống, chọn quản lý trường   2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý trường   3. Quản lý chọn nút Thêm trường   4. Hệ thống hiển thị form thêm trường   5. Quản lý nhập các trường dữ liệu, nhấn nút Lưu   6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5.   7. Hệ thống thông báo thêm trường thành công, và lưu thông tin trường trong cở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý trường |
| Điều kiện đầu ra | Các thông tin trường được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Sửa thông tin trường |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện sửa thông tin các trường |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý trường 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý trường 3. Quản lý chọn trường muốn sửa, chọn sửa 4. Hệ thống lấy dữ liệu trường theo ID, và hiển thị form sửa trường 5. Quản lý nhập các trường dữ liệu, nhấn nút Cập nhật 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo sửa trường thành công, và cập nhật thông tin trường trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý trường |
| Điều kiện đầu ra | Các thông tin trường được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xóa thông tin trường |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Người quản lý thực hiện xóa các trường không còn quản lý |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý trường 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý trường 3. Quản lý chọn trường muốn xóa, chọn xóa 4. Hệ thống hiển thị modal trước khi xóa 5. Quản lý chọn Xóa 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và hủy thao tác. 7. Hệ thống thông báo xóa trường thành công, và cập nhật trạng thái xóa trường trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý trường |
| Điều kiện đầu ra | Các thông tin trường được lưu vào cơ sở dữ liệu |

* *Use Case quản lý lớp:*

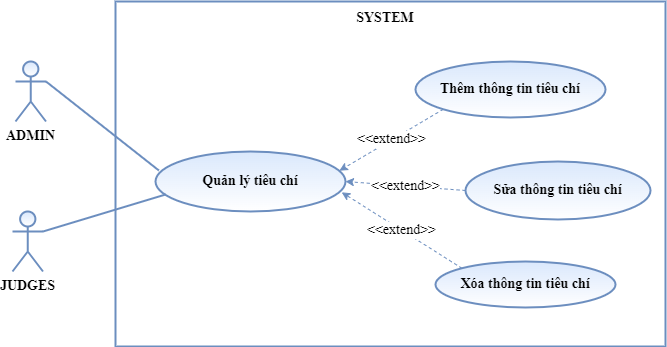
****

|  |  |
| --- | --- |
| Use Case | Thêm thông tin lớp |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện thêm các lớp mới |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | * 1. Quản lý đăng nhập hệ thống, chọn quản lý lớp   2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý lớp   3. Quản lý chọn nút Thêm lớp   4. Hệ thống hiển thị form thêm lớp   5. Quản lý nhập các lớp dữ liệu, nhấn nút Lưu   6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5.   7. Hệ thống thông báo thêm lớp thành công, và lưu thông tin lớp trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý lớp |
| Điều kiện đầu ra | Các thông tin lớp được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Sửa thông tin lớp |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện sửa thông tin các lớp |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý lớp 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý lớp 3. Quản lý chọn lớp muốn sửa, chọn sửa 4. Hệ thống lấy dữ liệu lớp theo ID, và hiển thị form sửa lớp 5. Quản lý nhập các lớp dữ liệu, nhấn nút Cập nhật 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo sửa lớp thành công, và cập nhật thông tin lớp trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý lớp |
| Điều kiện đầu ra | Các thông tin lớp được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xóa thông tin lớp |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Người quản lý thực hiện xóa các lớp không còn quản lý |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý lớp 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý lớp 3. Quản lý chọn lớp muốn xóa, chọn xóa 4. Hệ thống hiển thị modal trước khi xóa 5. Quản lý chọn Xóa 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và hủy thao tác. 7. Hệ thống thông báo xóa lớp thành công, và cập nhật trạng thái xóa lớp trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý lớp |
| Điều kiện đầu ra | Các thông tin lớp được lưu vào cơ sở dữ liệu |

* *Use Case quản lý tiêu chí:*

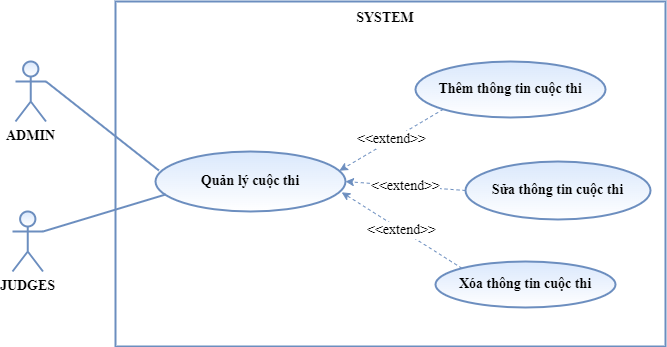
****

|  |  |
| --- | --- |
| Use Case | Thêm thông tin tiêu chí |
| Tác nhân | Quản trị viên và giảng viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện thêm các tiêu chí mới để bổ sung cho thông tin của cuộc thi |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tiêu chí 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tiêu chí 3. Quản lý chọn nút Thêm tiêu chí 4. Hệ thống hiển thị form thêm tiêu chí 5. Quản lý nhập các tiêu chí dữ liệu, nhấn nút Lưu 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo thêm tiêu chí thành công, và lưu thông tin tiêu chí trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tiêu chí |
| Điều kiện đầu ra | Các thông tin tiêu chí được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Sửa thông tin tiêu chí |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện sửa thông tin các tiêu chí |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tiêu chí 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tiêu chí 3. Quản lý chọn tiêu chí muốn sửa, chọn sửa 4. Hệ thống lấy dữ liệu tiêu chí theo ID, và hiển thị form sửa tiêu chí 5. Quản lý nhập các tiêu chí dữ liệu, nhấn nút Cập nhật 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo sửa tiêu chí thành công, và cập nhật thông tin tiêu chí trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tiêu chí |
| Điều kiện đầu ra | Các thông tin tiêu chí được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xóa thông tin tiêu chí |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Người quản lý thực hiện xóa các tiêu chí không còn quản lý |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tiêu chí 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tiêu chí 3. Quản lý chọn tiêu chí muốn xóa, chọn xóa 4. Hệ thống hiển thị modal trước khi xóa 5. Quản lý chọn Xóa 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và hủy thao tác. 7. Hệ thống thông báo xóa tiêu chí thành công, và cập nhật trạng thái xóa tiêu chí trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tiêu chí |
| Điều kiện đầu ra | Các thông tin tiêu chí được lưu vào cơ sở dữ liệu |

* *Use Case quản lý cuộc thi:*

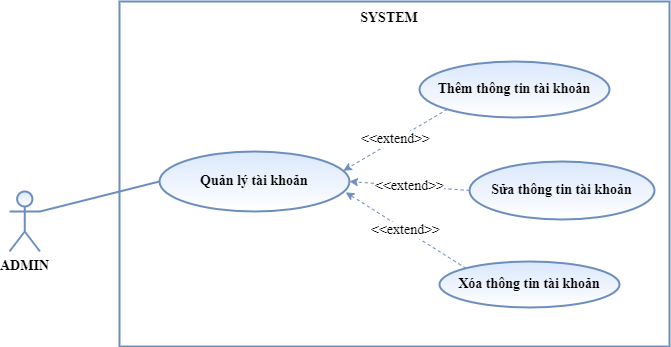
****

|  |  |
| --- | --- |
| Use Case | Thêm thông tin cuộc thi |
| Tác nhân | Quản trị viên và giảng viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện thêm các cuộc thi mới |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý cuộc thi 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý cuộc thi 3. Quản lý chọn nút Thêm cuộc thi 4. Hệ thống hiển thị form thêm cuộc thi 5. Quản lý nhập các cuộc thi dữ liệu, nhấn nút Lưu 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo thêm cuộc thi thành công, và lưu thông tin cuộc thi trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý cuộc thi |
| Điều kiện đầu ra | Các thông tin cuộc thi được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Sửa thông tin cuộc thi |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện sửa thông tin các cuộc thi |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý cuộc thi 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý cuộc thi 3. Quản lý chọn cuộc thi muốn sửa, chọn sửa 4. Hệ thống lấy dữ liệu cuộc thi theo ID, và hiển thị form sửa cuộc thi 5. Quản lý nhập các cuộc thi dữ liệu, nhấn nút Cập nhật 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo sửa cuộc thi thành công, và cập nhật thông tin cuộc thi trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý cuộc thi |
| Điều kiện đầu ra | Các thông tin cuộc thi được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xóa thông tin cuộc thi |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Người quản lý thực hiện xóa các cuộc thi không còn quản lý |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý cuộc thi 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý cuộc thi 3. Quản lý chọn cuộc thi muốn xóa, chọn xóa 4. Hệ thống hiển thị modal trước khi xóa 5. Quản lý chọn Xóa 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và hủy thao tác. 7. Hệ thống thông báo xóa cuộc thi thành công, và cập nhật trạng thái xóa cuộc thi trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý cuộc thi |
| Điều kiện đầu ra | Các thông tin cuộc thi được lưu vào cơ sở dữ liệu |

* *Use Case quản lý tài khoản:*

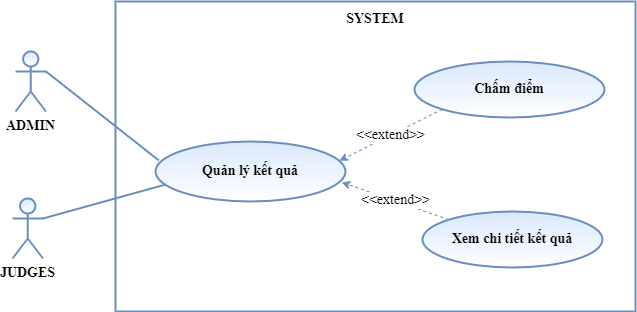
****

|  |  |
| --- | --- |
| Use Case | Thêm thông tin tài khoản |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện thêm các tài khoản mới |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tài khoản 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tài khoản 3. Quản lý chọn nút Thêm tài khoản 4. Hệ thống hiển thị form thêm tài khoản 5. Quản lý nhập các tài khoản dữ liệu, nhấn nút Lưu 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo thêm tài khoản thành công, và lưu thông tin tài khoản trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tài khoản |
| Điều kiện đầu ra | Các thông tin tài khoản được lưu vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Sửa thông tin tài khoản |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện sửa thông tin các tài khoản |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tài khoản 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tài khoản 3. Quản lý chọn tài khoản muốn sửa, chọn sửa 4. Hệ thống lấy dữ liệu tài khoản theo ID, và hiển thị form sửa tài khoản 5. Quản lý nhập các tài khoản dữ liệu, nhấn nút Cập nhật 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống thông báo sửa tài khoản thành công, và cập nhật thông tin tài khoản trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tài khoản |
| Điều kiện đầu ra | Các thông tin tài khoản được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xóa thông tin tài khoản |
| Tác nhân | Quản trị viên |
| Mô tả ngắn gọn | Người quản lý thực hiện xóa các tài khoản không còn quản lý |
| Tiền điều kiện | Người dùng đăng nhập thành công, với quyền quản trị |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý tài khoản 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý tài khoản 3. Quản lý chọn tài khoản muốn xóa, chọn xóa 4. Hệ thống hiển thị modal trước khi xóa 5. Quản lý chọn Xóa 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và hủy thao tác. 7. Hệ thống thông báo xóa tài khoản thành công, và cập nhật trạng thái xóa tài khoản trong cơ sở dữ liệu |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tài khoản |
| Điều kiện đầu ra | Các thông tin tài khoản được lưu vào cơ sở dữ liệu |

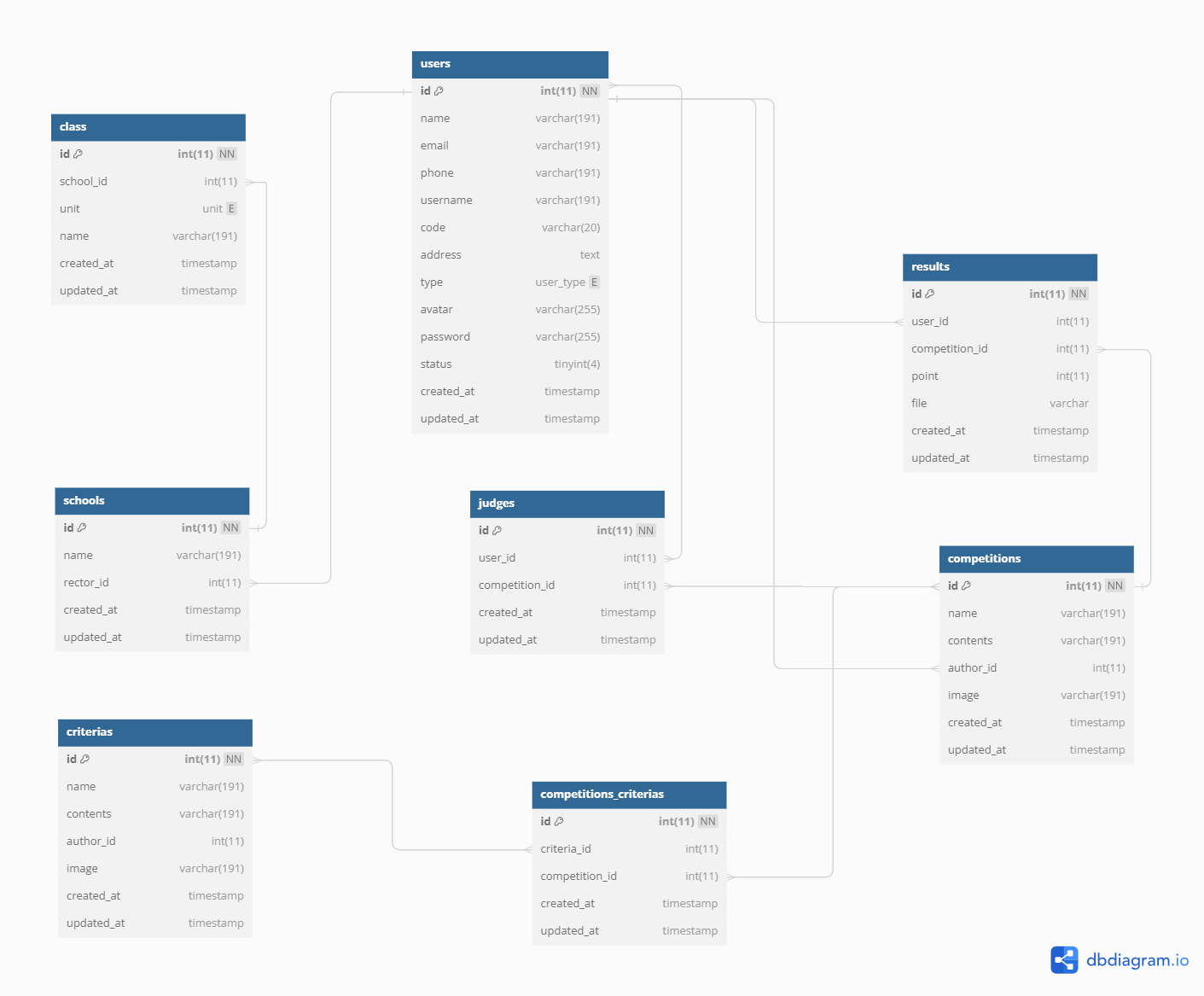
* *Use Case quản lý kết quả:*

****

|  |  |
| --- | --- |
| Use Case | Chấm điểm |
| Tác nhân | Quản trị viên và giảng viên |
| Mô tả ngắn gọn | Quản trị viên thực hiện chấm điểm các bài thi |
| Tiền điều kiện | Quản lý đăng nhập thành công hệ thống, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý đăng nhập hệ thống, chọn quản lý kết quả 2. Hệ thống kiểm tra tài khoản và quyền, nếu đúng thì truy cập hệ thống, chuyển hướng tới trang quản lý kết quả 3. Quản lý chọn bài thi muốn chấm, chọn chấm điểm 4. Hệ thống lấy dữ liệu bài thi theo ID, và hiển thị form sửa bài thi 5. Quản lý thực hiện chấm điểm, nhấn nút Xác nhận 6. Hệ thống kiểm tra dữ liệu: Nếu thỏa mãn thực hiện bước tiếp theo. Ngược lại, thông báo lỗi và quay lại bước 5. 7. Hệ thống cập nhật kết quả chấm thi trong cơ sở dữ liệu 8. Hệ thống thông báo cập nhật kết quả bài thi thành công |
| Ngoại lệ | 5. Quản lý chọn Hủy  6. Hệ thống hủy thao tác, và trả về trang quản lý tài khoản |
| Điều kiện đầu ra | Các kết quả chấm thi được cập nhật vào cơ sở dữ liệu |

|  |  |
| --- | --- |
| Use Case | Xem chi tiết kết quả |
| Tác nhân | Quản trị viên và giảng viên |
| Mô tả ngắn gọn | Quản lý muốn xem chi tiết kết quả bài thi |
| Tiền điều kiện | Quản lý đăng nhập thành công hệ thống, với quyền quản trị hoặc quyền giảng viên |
| Chuỗi sự kiện chính | 1. Quản lý chọn bài thi muốn xem kết quả 2. Chọn xem chi tiết 3. Hệ thống lấy chi tiết kết quả bài thi theo ID 4. Hệ thống hiển thị trang chi tiết bài thi |
| Ngoại lệ | Không có |
| Điều kiện đầu ra | Chi tiết bài thi quản lý muốn xem hiển thị trên màn hình |

* 1. **Các biểu đồ hoạt động:**
  2. **Các biểu đồ tuần tự:**
  3. **Sơ đồ quan hệ các thực thể trong cơ sở dữ liệu:**

****

Hình 3. 1. Sơ đồ quan hệ các thực thể trong cơ sở dữ liệu

* 1. **Xác định thuộc tính các bảng thực thể:**
* *Bảng class (cuộc thi):*

Bảng 3. 1. Bảng class (cuộc thi)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã cuộc thi |
| 2 | school\_id | int(11) | FK | Mã cuộc thi |
| 3 | unit | varchar(50) | Not null | Đơn vị |
| 4 | name | varchar(191) | Not null | Tên cuộc thi |
| 5 | created\_at | timestamp | Null | Ngày tạo |
| 6 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng schools (cuộc thi):*

Bảng 3. 2. Bảng schools (cuộc thi)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã cuộc thi |
| 2 | name | varchar(191) | Not null | Tên cuộc thi |
| 3 | rector\_id | int(11) | FK | Mã hiệu trưởng |
| 4 | created\_at | timestamp | Null | Ngày tạo |
| 5 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng criterias (cuộc thi):*

Bảng 3. 3. Bảng criterias (cuộc thi)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã cuộc thi |
| 2 | name | varchar(191) | Not null | Tên cuộc thi |
| 3 | contents | varchar(191) | Null | Nội dung |
| 4 | author\_id | int(11) | FK | Mã người tạo |
| 5 | image | varchar(191) | Null | Hình ảnh |
| 6 | created\_at | timestamp | Null | Ngày tạo |
| 7 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng users (người dùng):*

Bảng 3. 4. Bảng users (người dùng)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã người dùng |
| 2 | name | varchar(191) | Not null | Tên người dùng |
| 3 | email | varchar(191) | Not null | Email |
| 4 | phone | varchar(191) | Null | Số điện thoại |
| 5 | username | varchar(191) | Not null | Tên đăng nhập |
| 6 | code | varchar(20) | Null | Mã |
| 7 | address | text | Null | Địa chỉ |
| 8 | type | tinyint(4) | Not null | Phân loại |
| 9 | avatar | varchar(255) | Null | Hình đại diện |
| 10 | password | varchar(255) | Not null | Mật khẩu |
| 11 | status | tinyint(4) | Not null | Trạng thái |
| 12 | created\_at | timestamp | Null | Ngày tạo |
| 13 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng judges (giám khảo):*

Bảng 3. 5. Bảng judges (giám khảo)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã giám khảo |
| 2 | user\_id | int(11) | FK | Mã người dùng |
| 3 | competition\_id | int(11) | FK | Mã cuộc thi |
| 4 | created\_at | timestamp | Null | Ngày tạo |
| 5 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng results (kết quả):*

Bảng 3. 6. Bảng results (kết quả)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã kết quả |
| 2 | user\_id | int(11) | FK | Mã người dùng |
| 3 | competition\_id | int(11) | FK | Mã cuộc thi |
| 4 | point | int(11) | Not null | Điểm |
| 5 | file | varchar(max) | Not null | Tập tin đính kèm |
| 6 | created\_at | timestamp | Null | Ngày tạo |
| 7 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng competitions (cuộc thi):*

Bảng 3. 7. Bảng competitions (cuộc thi)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã cuộc thi |
| 2 | name | varchar(191) | Not null | Tên cuộc thi |
| 3 | contents | varchar(191) | Null | Nội dung |
| 4 | author\_id | int(11) | FK | Mã người tạo |
| 5 | image | varchar(191) | Null | Hình ảnh |
| 6 | created\_at | timestamp | Null | Ngày tạo |
| 7 | updated\_at | timestamp | Null | Ngày cập nhật |

* *Bảng competitions\_criterias (cuộc thi cuộc thi):*

Bảng 3. 8. Bảng competitions\_criterias (cuộc thi cuộc thi)

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | id | int(11) | PK | Mã cuộc thi cuộc thi |
| 2 | criteria\_id | int(11) | FK | Mã cuộc thi |
| 3 | competition\_id | int(11) | FK | Mã cuộc thi |
| 4 | created\_at | timestamp | Null | Ngày tạo |
| 5 | updated\_at | timestamp | Null | Ngày cập nhật |

**CHƯƠNG 4: GIAO DIỆN PHẦN MỀM**

**CHƯƠNG 5: TỔNG KẾT**

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| **(1)** | ***ReactJS là gì? Những điều bạn có thể chưa biết về ReactJS* [Online].** Nguồn: <https://200lab.io/blog/reactjs-la-gi> |
| **(2)** | *JSON Web Token (JWT) là gì ? Xác thực Nodejs với JWT* [Online]. Nguồn: <https://hocweb.vn/json-web-token-jwt-la-gi-xac-thuc-nodejs-voi-jwt> |
|  |  |