**XÂY DỰNG ỨNG DỤNG WEB HỖ TRỢ GIÁM THỊ COI THI DỰA TRÊN DỊCH VỤ NHẬN DẠNG KHUÔN MẶT CỦA AWS**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Mạng máy tính và truyền thông dữ liệu**

**TP.HỒ CHÍ MINH - 2024**

**XÂY DỰNG ỨNG DỤNG WEB HỖ TRỢ GIÁM THỊ COI THI DỰA TRÊN DỊCH VỤ NHẬN DẠNG KHUÔN MẶT CỦA AWS**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công Nghệ Thông Tin**

**Cán bộ hướng dẫn: PGS.TS. ................**

**TP.HỒ CHÍ MINH - 2024**

**LỜI CẢM ƠN**

Tôi xin bày tỏ lòng biết ơn sâu sắc đến những người đã giúp đỡ và hỗ trợ tôi trong quá trình thực hiện đồ án này.

Tôi muốn gửi lời cảm ơn chân thành đến PGS.TS ....................................(Bộ môn Công nghệ phần mềm, khoa Công nghệ thông tin, ....................................) đã trực tiếp định hướng, chỉ dẫn tôi trong việc nghiên cứu và hoàn thành đồ án.

Tôi cũng muốn gửi lời cảm ơn đến các giảng viên và cán bộ của trường Đại học Công nghệ đã tạo điều kiện cho tôi học tập và nghiên cứu, cung cấp môi trường tốt để tiếp thu và tích lũy kiến thức về lĩnh vực Công nghệ thông tin.

Cuối cùng, tôi xin gửi lời cảm ơn đến gia đình, bạn bè, và những người luôn ủng hộ và động viên tôi trong suốt quá trình này. Sự khích lệ và quan tâm của mọi người là nguồn động lực giúp tôi vượt qua khó khăn và hoàn thành đồ án này.

**LỜI CAM ĐOAN**

Tôi xin cam đoan đồ án là kết quả nghiên cứu của tôi, dưới sự hướng dẫn của PGS.TS ....................................(trường ....................................TP.Hồ Chí Minh).

Tài liệu tham khảo phục vụ nghiên cứu đồ án được thu thập từ các nguồn khác nhau đều có xuất xứ rõ ràng và trích dẫn hợp pháp trong phần tài liệu tham khảo.

Tôi xin chịu hoàn toàn trách chịu trách nhiệm và nhận hình thức kỷ luật của trường ....................................TP.Hồ Chí Minh trong trường hợp vi phạm những điều đã cam kết.

TP.Hồ Chí Minh, ngày … tháng … năm 2024

Sinh viên

**MỤC LỤC**

[CHƯƠNG 1: MỞ ĐẦU 10](#_Toc169379383)

[1.1. Mục đích 10](#_Toc169379384)

[1.2. Mục tiêu của dự án 11](#_Toc169379385)

[1.3. Nội dung đồ án 12](#_Toc169379386)

[1.4. Bố cục đồ án 13](#_Toc169379387)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 14](#_Toc169379388)

[2.1. React JS 14](#_Toc169379389)

[2.1.1. Giới thiệu về React JS 14](#_Toc169379390)

[2.1.2. Ưu điểm của React JS 14](#_Toc169379391)

[2.1.3. Nhược điểm của React JS 15](#_Toc169379392)

[2.2. NodeJS 16](#_Toc169379393)

[2.2.1. Giới thiệu về NodeJS 16](#_Toc169379394)

[2.2.2. Ưu điểm của NodeJS 17](#_Toc169379395)

[2.2.3. Nhược điểm của NodeJS 18](#_Toc169379396)

[2.3. Mô hình 19](#_Toc169379397)

[2.3.1. Giới thiệu về mô hình MVC 19](#_Toc169379398)

[2.3.2. Ưu điểm của mô hình MVC 20](#_Toc169379399)

[2.3.3. Nhược điểm của mô hình MVC 20](#_Toc169379400)

[2.4. Cơ sở dữ liệu MySQL 21](#_Toc169379401)

[2.4.3. Giới thiệu Mysql 21](#_Toc169379402)

[2.4.4. Ưu điểm của Mysql 22](#_Toc169379403)

[2.4.5. Nhược điểm của Mysql 23](#_Toc169379404)

[2.5 Amazon Rekognition 23](#_Toc169379405)

[2.3.3. Giới thiệu Amazon Rekognition 23](#_Toc169379406)

[2.3.4. Ưu điểm của Amazon Rekognition 24](#_Toc169379407)

[2.3.5. Nhược điểm của Amazon Rekognition 25](#_Toc169379408)

[2.6 Google sheet API 25](#_Toc169379409)

[2.6.1 Giới thiệu Google Sheet API 25](#_Toc169379410)

[2.6.2 Ưu điểm của Google Sheet API 25](#_Toc169379411)

[2.6.3 Nhược điểm của Google Sheet API 25](#_Toc169379412)

[CHƯƠNG 3: THU THẬP VÀ PHÂN TÍCH YÊU CẦU 26](#_Toc169379413)

[2.7 Mô tả bài toán 26](#_Toc169379414)

[2.8 Đặc tả yêu cầu 27](#_Toc169379415)

[2.8.1 Yêu cầu chức năng 27](#_Toc169379416)

[2.8.2 Yêu cầu phi chức năng 29](#_Toc169379417)

[2.9 Sơ đồ usecase 30](#_Toc169379418)

[2.10 Đặc tả usecase 31](#_Toc169379419)

[2.11 Biểu đồ hoạt động các ca sử dụng 31](#_Toc169379420)

[CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG 31](#_Toc169379421)

[2.12 Biểu đồ tuần tự các ca sử dụng 31](#_Toc169379422)

[2.13 Kiến trúc hệ thống backend 31](#_Toc169379423)

[2.14 Thiết kế cơ sở dữ liệu 32](#_Toc169379424)

[2.15 Thiết kế giao diện 32](#_Toc169379425)

[CHƯƠNG 5: KẾT LUẬN 32](#_Toc169379426)

[2.16 Kết quả đạt được 32](#_Toc169379427)

[2.17 Định hướng phát triển 32](#_Toc169379428)

# CHƯƠNG 1: MỞ ĐẦU

## Mục đích

Trong bối cảnh giáo dục ngày càng phát triển, việc giám sát và đảm bảo tính trung thực trong các kỳ thi trở thành một yếu tố quan trọng. Truyền thống, việc coi thi thường dựa vào sự hiện diện vật lý của giám thị trong phòng thi để theo dõi và kiểm tra thí sinh. Tuy nhiên, phương pháp này không chỉ tốn nhiều nguồn lực mà còn gặp nhiều hạn chế trong việc phát hiện các hành vi gian lận tinh vi. Với sự tiến bộ của công nghệ, đặc biệt là trong lĩnh vực trí tuệ nhân tạo và nhận dạng khuôn mặt, có thể cải thiện hiệu quả và tính chính xác của quá trình giám sát thi cử.

Mục đích của đề tài "Xây Dựng Ứng Dụng Web Hỗ Trợ Giám Thị Coi Thi Dựa Trên Dịch Vụ Nhận Dạng Khuôn Mặt Của AWS" là nghiên cứu và phát triển một ứng dụng web sử dụng dịch vụ nhận dạng khuôn mặt của Amazon Web Services (AWS) để hỗ trợ quá trình coi thi. Ứng dụng này không chỉ giúp giám thị dễ dàng theo dõi và quản lý thí sinh trong thời gian thực, mà còn tự động phát hiện các hành vi đáng ngờ như việc thay đổi thí sinh, hay hành vi gian lận trong phòng thi. Bằng cách ứng dụng công nghệ nhận dạng khuôn mặt, chúng tôi mong muốn nâng cao tính chính xác, hiệu quả và giảm thiểu sự phụ thuộc vào nguồn lực con người trong quá trình giám sát thi cử.

Hơn nữa, việc sử dụng dịch vụ nhận dạng khuôn mặt của AWS mang lại nhiều lợi ích về mặt công nghệ như khả năng mở rộng, bảo mật và độ tin cậy cao. AWS cung cấp một nền tảng mạnh mẽ và linh hoạt, cho phép tích hợp dễ dàng với các hệ thống hiện có và hỗ trợ việc phát triển các tính năng nâng cao trong tương lai. Qua đó, ứng dụng này không chỉ có ý nghĩa trong việc nâng cao chất lượng và sự công bằng trong giáo dục, mà còn góp phần vào xu hướng số hóa và tự động hóa trong quản lý giáo dục hiện đại.

## Mục tiêu của dự án

Dự án "Xây Dựng Ứng Dụng Web Hỗ Trợ Giám Thị Coi Thi Dựa Trên Dịch Vụ Nhận Dạng Khuôn Mặt Của AWS" hướng tới mục tiêu nâng cao chất lượng và hiệu quả của quá trình giám sát thi cử thông qua việc áp dụng công nghệ tiên tiến. Trước hết, dự án nhằm tăng cường hiệu quả giám sát bằng cách tự động hóa quy trình nhận diện và theo dõi thí sinh. Nhờ tích hợp công nghệ nhận dạng khuôn mặt của AWS, ứng dụng sẽ giúp giám thị dễ dàng kiểm tra danh tính thí sinh một cách nhanh chóng và chính xác, giảm bớt sự phụ thuộc vào kiểm tra thủ công và cho phép quản lý hiệu quả số lượng lớn thí sinh.

Bên cạnh đó, một trong những mục tiêu quan trọng nhất của dự án là phát hiện và ngăn chặn các hành vi gian lận. Hệ thống sẽ giám sát liên tục và kịp thời nhận diện các hành vi bất thường trong phòng thi, chẳng hạn như việc thay đổi thí sinh hoặc hành vi lừa đảo khác. Khi phát hiện vấn đề, hệ thống sẽ tự động gửi cảnh báo tới giám thị để có biện pháp can thiệp kịp thời, đảm bảo tính minh bạch và công bằng trong kỳ thi.

Đồng thời, dự án cũng tập trung vào việc tăng cường tính linh hoạt và khả năng mở rộng của ứng dụng. Với thiết kế mở, ứng dụng có thể dễ dàng tích hợp vào các hệ thống quản lý thi cử hiện có và mở rộng quy mô để phù hợp với các tổ chức giáo dục có số lượng thí sinh lớn. Điều này cho phép ứng dụng không chỉ phục vụ các đơn vị nhỏ lẻ mà còn đáp ứng được nhu cầu của các tổ chức lớn với hàng ngàn thí sinh tham gia thi cử.

Ngoài ra, đảm bảo bảo mật và quyền riêng tư của thí sinh là ưu tiên hàng đầu của dự án. Hệ thống sẽ được xây dựng để tuân thủ các tiêu chuẩn bảo mật cao nhất, bao gồm việc mã hóa dữ liệu và hạn chế quyền truy cập, nhằm bảo vệ thông tin cá nhân và hình ảnh của thí sinh khỏi các nguy cơ xâm phạm.

Cuối cùng, dự án hướng tới mục tiêu đơn giản hóa và cải thiện trải nghiệm người dùng. Ứng dụng sẽ được thiết kế với giao diện trực quan, dễ sử dụng, giúp giám thị có thể dễ dàng theo dõi và quản lý kỳ thi, đồng thời tạo điều kiện thuận lợi để thí sinh tham gia thi cử một cách suôn sẻ mà không gặp trở ngại từ việc sử dụng công nghệ mới. Bằng việc đạt được các mục tiêu này, dự án không chỉ giúp nâng cao hiệu quả quản lý thi cử mà còn thúc đẩy sự phát triển của công nghệ hiện đại trong lĩnh vực giáo dục, đáp ứng nhu cầu ngày càng cao về chất lượng và sự công bằng.

## Nội dung đồ án

Đồ án "Xây Dựng Ứng Dụng Web Hỗ Trợ Giám Thị Coi Thi Dựa Trên Dịch Vụ Nhận Dạng Khuôn Mặt Của AWS" sẽ tập trung vào việc phát triển và triển khai một hệ thống web giúp giám sát và quản lý kỳ thi một cách hiệu quả, tích hợp công nghệ nhận dạng khuôn mặt tiên tiến. Nội dung đồ án được chia thành hai phần chính, bao gồm các chức năng dành cho người quản trị (Admin) và người dùng (Client).

Phần đầu tiên của đồ án sẽ chi tiết hóa các chức năng dành cho Admin, bao gồm việc đăng ký, đăng nhập và quản lý tài khoản. Admin có thể thực hiện các thao tác như quên mật khẩu, thay đổi mật khẩu và cập nhật thông tin cá nhân để đảm bảo rằng tài khoản luôn được bảo mật và cập nhật. Ngoài ra, Admin còn có khả năng quản lý tài khoản của các người dùng khác, giúp duy trì hệ thống và điều chỉnh quyền truy cập một cách hiệu quả. Một khía cạnh quan trọng khác là việc quản lý môn học, từ việc thêm mới, cập nhật thông tin đến việc quản lý học viên của từng môn học. Đặc biệt, Admin có thể tạo lịch thi cho các môn học, đảm bảo rằng kỳ thi được tổ chức một cách suôn sẻ và có tổ chức.

Phần tiếp theo của đồ án sẽ tập trung vào các chức năng dành cho Client, chủ yếu là giám thị và thí sinh. Một trong những chức năng quan trọng là việc điểm danh sinh viên cho các môn học sử dụng công nghệ nhận dạng khuôn mặt của AWS. Thông qua ứng dụng, giám thị có thể dễ dàng xác minh danh tính của sinh viên tham gia thi cử, đảm bảo rằng chỉ những sinh viên đã được xác thực mới có thể tham gia kỳ thi. Sau khi điểm danh, hệ thống sẽ tự động cập nhật danh sách sinh viên đã điểm danh trên Google Sheets, giúp giám thị có thể theo dõi và quản lý thông tin một cách thuận tiện và chính xác.

Để hiện thực hóa các chức năng này, đồ án sẽ đi qua các giai đoạn từ phân tích yêu cầu hệ thống, thiết kế kiến trúc và giao diện, cho đến triển khai và kiểm thử hệ thống. Cụ thể, phần thiết kế hệ thống sẽ mô tả chi tiết cấu trúc của ứng dụng web, bao gồm các thành phần frontend và backend, và cách chúng tích hợp với các dịch vụ của AWS như Amazon Rekognition cho nhận dạng khuôn mặt. Việc triển khai sẽ bao gồm cài đặt và cấu hình các dịch vụ cần thiết, đồng thời đảm bảo rằng hệ thống có thể hoạt động ổn định và bảo mật.

## Bố cục đồ án

Nội dung của đồ án gồm các phần chính như sau:

Chương 1: Mở đầu: Lý do chọn đề tài, mục tiêu của dự án, nội dung nghiên cứu và phạm vi ứng dụng của đề tài.

Chương 2: Cơ sở lý thuyết: Nội dung chính của chương là đưa ra được những cơ sở lý thuyết phục vụ cho quá trình tìm hiểu và xây dựng hệ thống khi đã có yêu cầu bài toán.

Chương 3: Thu thập và phân tích yêu cầu: Nội dung của chương là xác định yêu cầu, đặc tả yêu cầu và các ca sử dụng của người dùng.

Chương 4: Phân tích thiết kế hệ thống và xây dựng hệ thống: Mô tả kiến trúc của hệ thống, cơ sở dữ liệu dùng trong hệ thống. Cùng với kết quả đạt được trên ứng dụng thật sau khi đã phân tích thiết kế.

Chương 5: Kết luận và hướng phát triển: Tổng kết những kết quả đã đạt được trong quá trình thực hiện đồ án và hướng phát triển sau này.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## React JS

### Giới thiệu về React JS

React.js, thường gọi tắt là React, là một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook và cộng đồng open-source. Nó được sử dụng rộng rãi trong việc xây dựng giao diện người dùng hiện đại trên web và ứng dụng di động. Đặc trưng nổi bật của React chính là việc quản lý giao diện người dùng một cách hiệu quả và linh hoạt [1].

React tập trung vào việc xây dựng các thành phần (components) riêng lẻ, mỗi thành phần đại diện cho một phần của giao diện người dùng. Điều này giúp tạo ra mã nguồn dễ đọc, tái sử dụng và bảo trì. React sử dụng JSX (JavaScript XML) để định nghĩa giao diện người dùng, cho phép lập trình viên kết hợp mã JavaScript và HTML một cách tự nhiên.

Một trong những điểm đặc biệt của React là sử dụng Virtual DOM (Document Object Model ảo) để cải thiện hiệu suất ứng dụng. Thay vì cập nhật toàn bộ DOM mỗi khi có sự thay đổi, React tạo một bản sao ảo của DOM và so sánh nó với DOM thực tế, chỉ cập nhật những phần thay đổi thực sự. Điều này làm giảm tải cho trình duyệt và tăng tốc độ render, giúp ứng dụng chạy nhanh hơn và mượt mà hơn.

React cung cấp cách quản lý trạng thái (state) của ứng dụng một cách hiệu quả thông qua props và state của các thành phần. Điều này giúp xây dựng các ứng dụng tương tác và phản hồi nhanh chóng.

Ngoài ra, React được hỗ trợ mạnh mẽ từ cộng đồng, với hàng ngàn các thư viện và công cụ bổ sung, cũng như tài liệu phong phú và ví dụ thực tế. Điều này làm cho việc học và sử dụng React trở nên dễ dàng và có thể áp dụng trong nhiều loại dự án khác nhau.

### Ưu điểm của React JS

React.js, một thư viện JavaScript phổ biến cho việc xây dựng giao diện người dùng, mang đến một loạt ưu điểm đáng chú ý. Dưới đây là một đoạn văn mô tả các ưu điểm của React.js:

React.js được biết đến với tính linh hoạt và hiệu suất cao. Một trong những ưu điểm lớn nhất của React là khả năng tái sử dụng thành phần. Thư viện này cho phép bạn xây dựng các thành phần (components) riêng lẻ có khả năng tái sử dụng cao, từ các button đơn giản đến các thành phần phức tạp như bảng điều khiển. Việc này giúp tối ưu hóa quá trình phát triển và duy trì mã nguồn, giảm thiểu lặp lại và tăng cường tính nhất quán trong giao diện người dùng.

Khả năng làm việc chặt chẽ với Virtual DOM cũng là một ưu điểm nổi bật của React. Virtual DOM giúp cải thiện hiệu suất ứng dụng bằng cách chỉ cập nhật những phần thay đổi thực sự trong giao diện, giảm tải cho trình duyệt và tăng tốc độ render. Điều này đồng nghĩa với việc ứng dụng React có thể chạy nhanh hơn và mượt mà hơn, đặc biệt là trên các ứng dụng web lớn và phức tạp.

React cung cấp sự quản lý trạng thái (state) mạnh mẽ thông qua props và state của các thành phần. Điều này giúp dễ dàng theo dõi và kiểm soát dữ liệu trong ứng dụng, từ đó tạo ra các ứng dụng có khả năng tương tác cao và phản hồi nhanh chóng.

Sự hỗ trợ mạnh mẽ từ cộng đồng là một điểm mạnh khác của React. Cộng đồng React lớn mạnh, và có sẵn hàng ngàn các thư viện và công cụ bổ sung, giúp bạn giải quyết các vấn đề phát triển một cách nhanh chóng và hiệu quả. React cũng có tài liệu đầy đủ và các ví dụ thực tế, giúp bạn nắm bắt nhanh chóng cách sử dụng nó trong dự án của mình.

### Nhược điểm của React JS

React.js, một thư viện JavaScript phổ biến dùng để xây dựng giao diện người dùng, mang lại nhiều ưu điểm đáng chú ý. Tuy nhiên, với những lợi ích này, React.js cũng đi kèm với một số nhược điểm cần được xem xét.

Trước hết, việc học và làm quen với ReactJjs có thể đầy thách thức, đặc biệt đối với những người mới bắt đầu trong lĩnh vực phát triển web hoặc không có kinh nghiệm với JavaScript. JSX (JavaScript XML) và việc làm việc với các component là những khái niệm mới mẻ có thể gây khó khăn ban đầu.

Một nhược điểm khác là bạn thường cần sử dụng nhiều công cụ và thư viện bổ sung như Webpack, Babel, Redux, hoặc React Router để phát triển ứng dụng ReactJs hoàn chỉnh. Điều này có thể làm tăng độ phức tạp của dự án và đòi hỏi sự hiểu biết rộng rãi về hệ sinh thái của React.

Mặc dù React.js được thiết kế để cải thiện hiệu suất ứng dụng web, nhưng hiệu suất của ứng dụng React có thể không đủ trong các trường hợp của ứng dụng lớn hoặc khi không được tối ưu hóa một cách đúng cách. Điều này có thể dẫn đến tình trạng ứng dụng chạy chậm trên các trình duyệt cũ hoặc khi có nhiều component phức tạp.

Hơn nữa, việc quản lý trạng thái (state) trong các ứng dụng React lớn có thể trở nên phức tạp và khó kiểm soát. Dù có Redux để hỗ trợ quản lý trạng thái, nhưng điều này đòi hỏi sự hiểu biết và kỹ năng sử dụng một cách hiệu quả.

Ngoài ra, SEO (Search Engine Optimization) trong các ứng dụng React cũng đặt ra thách thức. Mặc dù có cải tiến, trình duyệt không thể hiểu các dữ liệu được tạo ra bằng JavaScript, điều này có thể ảnh hưởng đến khả năng tìm kiếm của ứng dụng trên các công cụ tìm kiếm.

## NodeJS

### Giới thiệu về NodeJS

NodeJS là một nền tảng được xây dựng dựa trên thời gian chạy JavaScript của Chrome để dễ dàng xây dựng các ứng dụng mạng nhanh và có thể mở rộng. NodeJS sử dụng kiến trúc hướng sự kiện event-driver, không chặn (non-blocking) làm cho nó nhẹ nhàng hơn, hoàn hảo cho các ứng dụng real-time sử dụng nhiều dữ liệu chạy trên các thiết bị phân tán [2].

NodeJS là môi trường chạy mã JavaScript phía máy chủ.Nó được xây dựng dựa trên JavaScript engine V8 của Google Chrome, cho phép chạy mã JavaScript nhanh và hiệu quả trên máy chủ.

Với NodeJS bạn thể xây dựng ứng dụng web và back-end bằng JavaScript, một ngôn ngữ phổ biến và linh hoạt. NodeJS cho phép bạn xử lý các ứng dụng khách yêu cầu và tạo API, kết nối với cơ sở dữ liệu, thao tác với các tệp và thực hiện các tác vụ máy chủ khác một cách đơn giản và hiệu quả.

NodeJS cũng chú trọng đến việc xử lý không đồng bộ (bất đồng bộ), cho phép ứng dụng xử lý nhiều yêu cầu cùng một lúc và không bị chặn bởi các tác vụ đang chờ đợi. Điều này làm cho NodeJS trở thành một lựa chọn phù hợp cho việc xây dựng ứng dụng chạy thời gian thực và có khả năng xử lý tải cao.

NodeJS cũng đi kèm với mô-đun thư viện phong phú, cho phép lập trình viên sử dụng các chức năng có sẵn và không cần viết lại từ đầu. Nó còn có một cộng đồng rộng lớn và hỗ trợ mạnh mẽ từ cộng đồng lập trình viên, chắc chắn rằng bạn có thể tìm thấy các tài liệu hướng dẫn và các mô-đun hữu ích để phát triển ứng dụng NodeJS của riêng bạn.

### Ưu điểm của NodeJS

NodeJS có nhiều ưu điểm khi sử dụng trong việc phát triển ứng dụng web và back-end. Dưới đây là một số điểm ưu tiên của NodeJS:

**+** **Hiệu suất cao:** NodeJS được xây dựng trên JavaScript engine V8, đảm bảo hiệu suất cao với việc xử lý mã JavaScript nhanh chóng. Nó cũng hỗ trợ xử lý không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không gây gián đoạn.

**+** **Xử lý không đồng bộ:** NodeJS được xây dựng với kiến trúc không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không bị chặn bởi các nhiệm vụ chờ đợi. Điều này làm cho ứng dụng trở nên rất nhạy bén và có thể đáp ứng tải cao.

**+ Sử dụng cùng một ngôn ngữ:** Với NodeJS, bạn có thể sử dụng cùng một ngôn ngữ JavaScript cho phía máy chủ và phía khách hàng. Điều này giúp giảm múc độ phức tạp trong quá trình phát triển và duy trì ứng dụng.

**+ Hệ sinh thái mô-đun phong phú:** NodeJS có một mô-đun hệ sinh thái phông phú, với hàng sát mô-đun có sẵn để tái sử dụng. Bạn có thể tìm thấy các mô-đun hữu ích trong cộng đồng NodeJS và sử dụng chúng để giảm thời gian phát triển và tăng tính linh hoạt của ứng dụng.

**+ Cộng đồng lập trình thành viên mạnh mẽ:** NodeJS có một cộng đồng lập trình thành viên lớn và nổi. Điều này đảm bảo rằng bạn co thể tìm thấy các tài nguyên, hướng dẫn và hổ trợ từ những người khác trong cộng đồng để giải quyết vấn đề và nâng cao kỹ năng lập trình của mình.

**+ Thích hợp cho ứng dụng thời gian thực:** Với việc hỗ trợ xử lý không đồng bộ và hiệu suất cao. NodeJS là một lựa chọn phù hợp cho việc xây dựng ứng dụng thời gian thực như ứng dụng trò chuyện trực tuyến,máy chủ trò chơi và ứng dụng theo thời gian thực khác.

### Nhược điểm của NodeJS

Mặc dù NodeJS có nhiều ưu điểm, nhưng cũng có một số nhược điểm bạn cần lưu ý:

**+ Đơn luồng:** NodeJS dựa trên mô hình đơn luồng, điều này có nghĩa là có một luồng xử lý trong Node.js. Điều này có thể gây ra vấn đề khi có các tác vụ tốn nhiều thời gian xử lý, làm cho ứng dụng hoạt động chậm.

**+ Khả năng xử lý CPU hạn chế:** NodeJS phù hợp cho việc xử lý các tác vụ I/O (ví dụ: yêu cầu mạng, truy vấn cơ sở dữ liệu), nhưng không phải là lựa chọn tốt cho các tác vụ tính toán mất nhiều thời gian. Do đó, nếu ứng dụng của bạn có yêu cầu xử lý CPU nặng, Node.js có thể không phải là lựa chọn tốt.

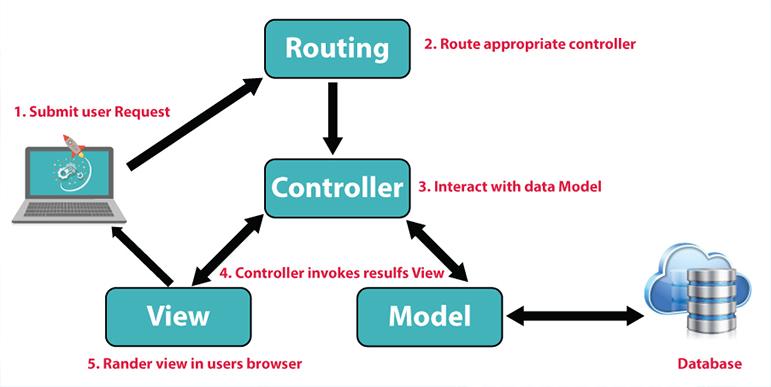
**+ Thư viện thứ ba không ổn định:** Mặc dù NodeJS có một hệ sinh thái module phong phú, nhung không phải tất cả các module bên thứ ba đều đảm bảo ổn định và chất lượng. Có thể gặp phải các vấn đề như sự không tương thích, lỗi hoặc sự thiếu hỗ trợ từ các module bên thứ ba.

**+ Quản lý phiên bản module:** Khi sử dụng NodeJS, việc quản lý phiên bản module có thể trở nên phức tạp. Vì NodeJS sử dụng mô hình xử lý module phiên bản, việc kiểm soát và đồng bộ các phiên bản module có thể trở nên rắc rối, đặc biệt khi có nhiều module sử dụng các phiên bản khác nhiêu.

**+ Scalability (tính mở rộng):** Scalability là một thách thức khi sử dụng NodeJS trong một môi trường có tải cao hoặc cần xây dựng ứng dụng lớn. Điều này đòi hỏi việc kiến trúc ứng dụng và các giải pháp phân tán được quản lý cẩn thận để đảm bảo tính mở rộng và hiệu xuất ổn định.

## Mô hình

### Giới thiệu về mô hình MVC



*Hình 2. 1: Mô hình MVC*

MVC là viết tắt của cụm từ "Model-View-Controller," là một mô hình kiến trúc phần mềm được sử dụng rộng rãi trong lĩnh vực công nghệ phần mềm. Mô hình này được thiết kế để quản lý và xây dựng giao diện người dùng trên các ứng dụng máy tính. MVC phân chia ứng dụng thành ba thành phần chính, và mỗi thành phần có vai trò và nhiệm vụ riêng biệt và độc lập với nhau.

Trong mô hình MVC, có ba thành phần không thể thiếu:

**+ Model:** Đây là thành phần chịu trách nhiệm lưu trữ toàn bộ dữ liệu của ứng dụng và đóng vai trò là giao diện giữa hai thành phần khác là View và Controller. Model thường đại diện cho dữ liệu sử dụng bởi ứng dụng, có thể là cơ sở dữ liệu, các file dữ liệu như XML, hoặc thậm chí là các đối tượng đơn giản như biểu tượng hoặc nhân vật trong một trò chơi.

**+ View:** Đây là thành phần giao diện dành cho người sử dụng, thể hiện thông tin và tương tác giữa ứng dụng và người dùng. View bao gồm các thành phần như cửa sổ, nút bấm, văn bản, và hiển thị dữ liệu để người dùng có thể nhìn thấy và tương tác với chúng.

+ **Controller:** Thành phần này có nhiệm vụ xử lý các yêu cầu được người dùng gửi qua View. Controller bao gồm cả Model và View, và nó nhận các dữ liệu đầu vào từ người dùng và thực hiện các cập nhật và xử lý tương ứng. Controller là bộ não của ứng dụng, điều hướng và quản lý luồng dữ liệu giữa Model và View.

### Ưu điểm của mô hình MVC

Mô hình MVC (Model-View-Controller) mang đến nhiều ưu điểm quan trọng trong phát triển ứng dụng. Trước hết, mô hình này giúp tiết kiệm băng thông bởi vì không sử dụng viewstate, từ đó giúp giảm tải cho server và cải thiện tính ổn định của website. Khả năng này không chỉ giúp giảm khả năng gặp sự cố kết nối mạng mà còn cải thiện trải nghiệm người dùng.

Việc kiểm tra ứng dụng cũng trở nên dễ dàng hơn trong mô hình MVC. Nhờ sự phân chia rõ ràng giữa các thành phần Model, Controller và View, việc kiểm tra lỗi phần mềm trước khi ứng dụng được bàn giao cho người dùng trở nên đơn giản và hiệu quả hơn. Điều này giúp đảm bảo rằng ứng dụng làm việc ổn định và cung cấp chất lượng tốt.

Mô hình MVC tách biệt các thành phần trong ứng dụng, bao gồm Model (mô hình dữ liệu), Controller (bộ điều khiển), và View (giao diện). Sự tách biệt này giúp quản lý mã nguồn dễ dàng hơn, làm cho mã trở nên sạch sẽ và dễ đọc. Điều này giúp tôi phát triển làm việc hiệu quả hơn và dễ dàng theo dõi và duy trì mã nguồn.

Controller, một phần quan trọng của mô hình MVC, được tối ưu hóa cho nhiều nền tảng ngôn ngữ khác nhau, từ đó tạo ra tính linh hoạt cho ứng dụng. Nó giúp ứng dụng hoạt động tốt trên nhiều môi trường và thiết bị khác nhau.

Duy trì ứng dụng trong mô hình MVC trở nên dễ dàng hơn do sự tách biệt giữa các thành phần. Bạn có thể chỉnh sửa hoặc nâng cấp một phần mà không ảnh hưởng đến các phần khác.

### Nhược điểm của mô hình MVC

Mô hình MVC (Model-View-Controller) là một kiến trúc phát triển phần mềm mạnh mẽ, nhưng cũng đi kèm với một số nhược điểm cần được xem xét. Ban đầu, việc hiểu và triển khai mô hình này có thể phức tạp, đặc biệt đối với người mới bắt đầu. Sự tách biệt chặt chẽ giữa Model (mô hình), View (giao diện) và Controller (điều khiển) có thể dẫn đến hiệu suất kém trong trường hợp các ứng dụng yêu cầu nhiều tương tác giữa các thành phần này.

Một trong những thách thức khác là việc quản lý mã trở nên phức tạp, đặc biệt là trong các dự án lớn. Cấu trúc MVC có thể tạo ra nhiều lớp và tầng lớp mã, làm cho mã trở nên khó quản lý và khó hiểu. Thời gian phát triển dự án cũng có thể kéo dài hơn so với các mô hình khác, đặc biệt khi yêu cầu của dự án thay đổi thường xuyên.

Đối với các dự án nhỏ và đơn giản, triển khai mô hình MVC có thể cảm thấy quá phức tạp và không cần thiết, làm cho mã trở nên quá nặng nề và khó quản lý. Cuối cùng, việc sử dụng mô hình MVC đòi hỏi kiến thức và kỹ năng cao cho phát triển và bảo trì hiệu quả, và có thể không phù hợp cho những dự án với đội ngũ phát triển không có kinh nghiệm trong việc làm việc với kiến trúc này.

## Cơ sở dữ liệu MySQL

### Giới thiệu Mysql

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến nhất hiện nay. Được phát triển vào giữa những năm 1990 bởi công ty Thụy Điển MySQL AB, MySQL nhanh chóng trở thành lựa chọn hàng đầu cho các ứng dụng web, nhờ vào tính linh hoạt, hiệu suất cao và khả năng mở rộng. Tên gọi "MySQL" xuất phát từ sự kết hợp của "My" - tên con gái của một trong những người sáng lập, và "SQL" - ngôn ngữ truy vấn có cấu trúc (Structured Query Language), là ngôn ngữ chuẩn được sử dụng để quản lý và thao tác với cơ sở dữ liệu.

MySQL được thiết kế với mục tiêu cung cấp một hệ quản trị cơ sở dữ liệu dễ sử dụng nhưng mạnh mẽ, phù hợp cho các dự án từ nhỏ đến lớn. Với khả năng hỗ trợ nhiều hệ điều hành như Linux, Windows và macOS, MySQL đã trở thành nền tảng dữ liệu của nhiều ứng dụng web phổ biến như WordPress, Joomla, và các hệ thống thương mại điện tử lớn. Sự phổ biến của MySQL còn được củng cố bởi sự hỗ trợ của một cộng đồng phát triển mạnh mẽ và sự tài trợ từ các công ty công nghệ lớn, bao gồm Oracle Corporation, công ty hiện đang sở hữu và phát triển MySQL.

### Ưu điểm của Mysql

MySQL nổi bật như một trong những hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến nhất thế giới nhờ vào nhiều ưu điểm vượt trội. Trước hết, MySQL được đánh giá cao về hiệu suất và tốc độ xử lý, đặc biệt là trong các ứng dụng web. Với khả năng xử lý hàng triệu truy vấn mỗi ngày, MySQL trở thành nền tảng lý tưởng cho các trang web có lượng truy cập lớn, giúp duy trì hiệu quả và sự ổn định ngay cả dưới áp lực cao. Là phần mềm mã nguồn mở, MySQL cung cấp sự linh hoạt tối đa cho người dùng và các nhà phát triển, cho phép họ tự do tùy chỉnh và mở rộng theo nhu cầu cụ thể của dự án mà không gặp rào cản về chi phí bản quyền.

Một trong những lý do chính khiến MySQL được ưa chuộng là tính linh hoạt cao trong việc hỗ trợ nhiều hệ điều hành và ngôn ngữ lập trình khác nhau, bao gồm PHP, Java, Python, và .NET. Điều này làm cho MySQL dễ dàng tích hợp vào các môi trường phát triển đa dạng, đáp ứng tốt cho cả các dự án nhỏ và lớn. Về mặt bảo mật, MySQL không hề thua kém bất kỳ hệ quản trị cơ sở dữ liệu nào khác. Nó cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, quản lý quyền truy cập chi tiết, và mã hóa dữ liệu, đảm bảo rằng thông tin của người dùng luôn được bảo vệ an toàn trước các mối đe dọa.

Khả năng mở rộng của MySQL cũng là một điểm mạnh đáng chú ý. MySQL có thể dễ dàng xử lý các cơ sở dữ liệu từ nhỏ đến rất lớn nhờ vào các tính năng như replication (sao chép dữ liệu) và clustering (cụm máy chủ), giúp hệ thống luôn sẵn sàng đáp ứng nhu cầu ngày càng tăng của các tổ chức lớn. Hơn nữa, sự hỗ trợ từ một cộng đồng phát triển toàn cầu mạnh mẽ là một tài sản quý giá của MySQL. Với hàng ngàn nhà phát triển và người dùng đóng góp vào việc cải thiện và hỗ trợ, người dùng MySQL luôn có sẵn nhiều tài liệu, công cụ, và giải pháp để giúp họ giải quyết mọi thách thức mà họ có thể gặp phải.

Những ưu điểm này giúp MySQL không chỉ là một lựa chọn phổ biến cho các nhà phát triển web và các doanh nghiệp mà còn là một nền tảng tin cậy để xây dựng các hệ thống dữ liệu mạnh mẽ, ổn định và linh hoạt.

### Nhược điểm của Mysql

Mặc dù MySQL nổi bật với nhiều ưu điểm và sự phổ biến rộng rãi, nhưng nó cũng tồn tại một số nhược điểm cần lưu ý khi lựa chọn hệ quản trị cơ sở dữ liệu cho các dự án phức tạp. Trước hết, MySQL có những giới hạn nhất định về tính năng so với các hệ quản trị cơ sở dữ liệu tiên tiến khác như PostgreSQL hay Oracle. MySQL thiếu một số tính năng cao cấp như hỗ trợ đầy đủ cho Common Table Expressions (CTEs), khả năng xử lý các truy vấn phức tạp và các giao dịch có độ phức tạp cao. Điều này có thể làm giảm hiệu quả khi cần thao tác với các dữ liệu lớn hoặc thực hiện các phép tính toán và phân tích phức tạp.

Một vấn đề khác là hiệu suất của MySQL có thể giảm đáng kể khi xử lý các cơ sở dữ liệu rất lớn hoặc trong các ứng dụng yêu cầu tốc độ truy cập rất cao. Mặc dù MySQL có khả năng mở rộng, nhưng việc tối ưu hóa hiệu suất cho các ứng dụng có lượng dữ liệu khổng lồ hoặc yêu cầu truy vấn thời gian thực phức tạp có thể trở thành thách thức lớn, đòi hỏi sự hiểu biết sâu rộng về cơ sở dữ liệu và các phương pháp tối ưu hóa cụ thể của MySQL.

Bên cạnh đó, mặc dù MySQL là một hệ quản trị cơ sở dữ liệu tuân thủ chuẩn SQL, nó không hoàn toàn tương thích với tất cả các tiêu chuẩn SQL, và có thể dẫn đến các vấn đề tương thích khi di chuyển dữ liệu hoặc tích hợp với các hệ quản trị cơ sở dữ liệu khác. Điều này đặc biệt quan trọng khi phải làm việc trong môi trường có sự tương tác đa dạng với nhiều loại cơ sở dữ liệu khác nhau.

Việc phát triển và cập nhật của MySQL, dù có một cộng đồng phát triển mạnh mẽ, đôi khi cũng diễn ra chậm hơn so với một số hệ quản trị cơ sở dữ liệu thương mại khác. Điều này có thể ảnh hưởng đến khả năng tiếp cận các tính năng mới hoặc các cải tiến về hiệu suất kịp thời, làm cho MySQL đôi khi trở thành một lựa chọn kém hấp dẫn hơn đối với các dự án cần các giải pháp hiện đại và nhanh chóng cập nhật.

## Amazon Rekognition

### Giới thiệu Amazon Rekognition

Amazon Rekognition là một dịch vụ mạnh mẽ do Amazon Web Services (AWS) cung cấp, chuyên cung cấp các tính năng nhận dạng hình ảnh và video bằng trí tuệ nhân tạo (AI). Dịch vụ này cho phép các nhà phát triển tích hợp dễ dàng các khả năng nhận dạng khuôn mặt, phát hiện vật thể và hành vi, phân tích video và ảnh động, cũng như đánh giá nội dung không phù hợp.

Amazon Rekognition cung cấp một loạt các tính năng nhận dạng khuôn mặt và phân tích hành vi, từ việc nhận dạng các đặc điểm khuôn mặt cụ thể đến phân tích hành vi trong video. Nó cũng hỗ trợ các tính năng phát hiện vật thể và đánh giá nội dung, giúp các tổ chức xây dựng và triển khai các ứng dụng thông minh và an toàn.

Đặc biệt, Amazon Rekognition tích hợp sâu vào hạ tầng AWS, cho phép tích hợp mượt mà với các dịch vụ khác như AWS Lambda, Amazon S3 và Amazon Kinesis Video Streams. Nhờ vào việc cung cấp API đơn giản và SDK hỗ trợ nhiều ngôn ngữ lập trình, Amazon Rekognition giúp các nhà phát triển nhanh chóng phát triển và triển khai các ứng dụng AI phức tạp mà không cần có sự chuyên sâu về machine learning.

Với những khả năng nổi bật và tích hợp linh hoạt, Amazon Rekognition là một lựa chọn lý tưởng cho các tổ chức muốn tận dụng sức mạnh của trí tuệ nhân tạo để tăng cường an ninh, quản lý dữ liệu và cải thiện trải nghiệm người dùng trong các ứng dụng của họ.

### Ưu điểm của Amazon Rekognition

Amazon Rekognition là một dịch vụ trí tuệ nhân tạo (AI) do Amazon Web Services (AWS) cung cấp, chuyên cung cấp các tính năng mạnh mẽ cho việc nhận dạng hình ảnh và video. Với khả năng phát hiện khuôn mặt, vật thể và hành vi, Amazon Rekognition đem lại cho các tổ chức và nhà phát triển những lợi ích đáng kể trong việc tự động hóa các quy trình phân tích dữ liệu và nâng cao trải nghiệm người dùng.

Một trong những ưu điểm nổi bật của Amazon Rekognition là tính linh hoạt và dễ sử dụng. Dịch vụ này cung cấp API đơn giản và SDK hỗ trợ nhiều ngôn ngữ lập trình, giúp các nhà phát triển có thể tích hợp các tính năng AI vào ứng dụng của mình một cách nhanh chóng và hiệu quả. Điều này không chỉ giảm thiểu thời gian phát triển mà còn giúp tăng tốc độ triển khai các giải pháp trí tuệ nhân tạo.

Amazon Rekognition cũng được đánh giá cao về hiệu suất và khả năng mở rộng. Với hạ tầng mạnh mẽ của AWS, dịch vụ này có thể xử lý các tình huống phức tạp và đồng thời đáp ứng được nhu cầu của các ứng dụng có lượng người dùng lớn một cách mượt mà. Điều này đặc biệt quan trọng đối với các ứng dụng yêu cầu độ chính xác cao và thời gian phản hồi nhanh.

Thêm vào đó, Amazon Rekognition còn cam kết đảm bảo bảo mật dữ liệu và tin cậy của người dùng. Dịch vụ này tuân thủ các tiêu chuẩn bảo mật nghiêm ngặt của ngành công nghiệp, đảm bảo rằng thông tin cá nhân và dữ liệu quan trọng được bảo vệ an toàn.

### Nhược điểm của Amazon Rekognition

Tuy nhiên, như mọi công nghệ và dịch vụ, Amazon Rekognition cũng có một số nhược điểm cần được xem xét trước khi triển khai:

Một trong những vấn đề đối mặt của Amazon Rekognition là độ chính xác và độ tin cậy của các tính năng nhận dạng. Dù đã được cải thiện đáng kể, các thuật toán AI vẫn có thể gặp phải các trường hợp nhận dạng sai lầm hoặc không chính xác, đặc biệt khi đối mặt với điều kiện ánh sáng yếu, góc chụp khó hoặc các tình huống phức tạp.

Bên cạnh đó, chi phí sử dụng Amazon Rekognition cũng là một yếu tố cần xem xét. Dịch vụ này tính phí theo số lượng yêu cầu xử lý và lưu trữ dữ liệu, do đó việc tính toán chi phí và tối ưu hóa tài nguyên là điều cần thiết để tránh gánh nặng tài chính cho các tổ chức và doanh nghiệp.

Thêm vào đó, việc bảo mật và quản lý dữ liệu cũng là một vấn đề quan trọng. Mặc dù Amazon Rekognition cam kết đảm bảo bảo mật thông tin, việc lưu trữ và xử lý dữ liệu nhạy cảm vẫn đòi hỏi các biện pháp bảo mật chặt chẽ để đảm bảo tuân thủ các quy định pháp luật và sự tin tưởng của người dùng.

## Google sheet API

### Giới thiệu Google Sheet API

### Ưu điểm của Google Sheet API

### Nhược điểm của Google Sheet API

# CHƯƠNG 3: THU THẬP VÀ PHÂN TÍCH YÊU CẦU

## Mô tả bài toán

Trong bối cảnh phát triển và hiện đại hóa lĩnh vực giáo dục, việc quản lý thi cử và điểm danh sinh viên một cách hiệu quả và chính xác là vô cùng quan trọng. Đồ án "Xây Dựng Ứng Dụng Web Hỗ Trợ Giám Thị Coi Thi Dựa Trên Dịch Vụ Nhận Dạng Khuôn Mặt Của AWS" được thiết kế nhằm giải quyết các thách thức trong việc giám sát và quản lý kỳ thi, đồng thời tối ưu hóa quy trình điểm danh sinh viên cho các môn học. Hệ thống này sẽ bao gồm hai nhóm chức năng chính, phục vụ cho các đối tượng người dùng khác nhau: Admin và Client.

**Chức Năng Của Admin**

**Đăng Ký, Đăng Nhập:**

* Hệ thống cung cấp giao diện cho Admin đăng ký tài khoản mới và đăng nhập vào hệ thống. Việc này đảm bảo rằng chỉ những người có quyền hạn mới có thể truy cập vào các chức năng quản trị của hệ thống.

**Quên Mật Khẩu, Thay Đổi Mật Khẩu:**

* Admin có thể sử dụng chức năng quên mật khẩu để khôi phục lại quyền truy cập trong trường hợp mất mật khẩu. Ngoài ra, hệ thống cho phép Admin thay đổi mật khẩu định kỳ hoặc khi cần thiết để tăng cường bảo mật.

**Thay Đổi Thông Tin Cá Nhân:**

* Admin có thể cập nhật các thông tin cá nhân của mình như tên, email, hoặc số điện thoại. Điều này giúp đảm bảo rằng các thông tin liên lạc của Admin luôn được cập nhật và chính xác.

**Quản Lý Tài Khoản:**

* Admin có khả năng quản lý các tài khoản người dùng khác trong hệ thống. Điều này bao gồm việc thêm mới, xóa bỏ hoặc cập nhật thông tin của các tài khoản này, giúp duy trì kiểm soát chặt chẽ và tổ chức các quyền truy cập.

**Quản Lý Môn Học:**

* Hệ thống cho phép Admin tạo, chỉnh sửa, và xóa các môn học trong hệ thống. Admin có thể thêm các thông tin chi tiết về từng môn học, giúp việc quản lý nội dung giáo dục trở nên dễ dàng và chính xác hơn.

**Quản Lý Học Viên Của Môn Học:**

* Admin có thể quản lý danh sách học viên tham gia từng môn học, bao gồm việc thêm mới, cập nhật hoặc loại bỏ học viên. Điều này giúp đảm bảo rằng thông tin về học viên luôn được duy trì một cách chính xác và kịp thời.

**Tạo Lịch Thi Cho Môn Học:**

* Hệ thống cho phép Admin lên lịch thi cho các môn học, bao gồm việc xác định ngày, giờ và địa điểm thi. Chức năng này giúp tổ chức kỳ thi một cách hiệu quả và thông báo kịp thời cho sinh viên.

**Chức Năng Của Client**

**Điểm Danh Sinh Viên Cho Môn Học:**

* Giám thị có thể sử dụng hệ thống để điểm danh sinh viên bằng công nghệ nhận dạng khuôn mặt của AWS. Việc này giúp xác minh danh tính của sinh viên một cách tự động và nhanh chóng, đảm bảo rằng chỉ những sinh viên đã đăng ký mới được tham gia vào lớp học hoặc kỳ thi.

**Xem Danh Sách Sinh Viên Đã Điểm Danh Trên Google Sheets:**

* Hệ thống sẽ tự động cập nhật danh sách sinh viên đã được điểm danh lên Google Sheets. Giám thị có thể dễ dàng truy cập và xem danh sách này để theo dõi và quản lý sự có mặt của sinh viên trong lớp học hoặc kỳ thi.

## Đặc tả yêu cầu

### Yêu cầu chức năng

Hệ thống website tôi hướng đến 2 đối tượng là: Admin, Client

*Bảng 3. 1: Yêu cầu chức năng*

|  |  |
| --- | --- |
| Đăng Ký | Admin có thể tạo tài khoản mới bằng cách điền các thông tin cá nhân cần thiết và gửi yêu cầu đăng ký. |
| Đăng Nhập | Admin nhập tên đăng nhập và mật khẩu để truy cập vào hệ thống quản trị. |
| Quên Mật Khẩu | Cung cấp tùy chọn cho Admin để yêu cầu đặt lại mật khẩu thông qua email trong trường hợp quên mật khẩu. |
| Thay Đổi Mật Khẩu | Admin có thể cập nhật mật khẩu mới cho tài khoản sau khi đăng nhập thành công hoặc khôi phục mật khẩu. |
| Thay Đổi Thông Tin Cá Nhân | Admin có thể cập nhật các thông tin cá nhân như tên, email, và số điện thoại thông qua trang hồ sơ cá nhân. |
| Quản Lý Tài Khoản | Admin có thể xem danh sách, tạo mới, chỉnh sửa hoặc xóa các tài khoản người dùng khác trong hệ thống. |
| Quản Lý Môn Học | Admin có thể thêm mới, chỉnh sửa hoặc xóa các thông tin chi tiết về các môn học trong hệ thống. |
| Quản Lý Học Viên Của Môn Học | Admin quản lý danh sách học viên cho từng môn học, bao gồm việc thêm, cập nhật hoặc xóa học viên. |
| Tạo Lịch Thi Cho Môn Học | Admin có thể thiết lập ngày, giờ, và địa điểm thi cho các môn học. |
| Điểm Danh Sinh Viên Cho Môn Học | Giám thị sử dụng công nghệ nhận dạng khuôn mặt để tự động điểm danh sinh viên tham dự môn học hoặc kỳ thi. |
| Xem Danh Sách Sinh Viên Trên Sheets | Giám thị có thể truy cập và xem danh sách sinh viên đã được điểm danh được cập nhật trên Google Sheets. |

### Yêu cầu phi chức năng

Hệ thống cần đáp ứng một số yêu cầu phi chức năng để đảm bảo hiệu suất, bảo mật và trải nghiệm người dùng tốt. Dưới đây là mô tả chi tiết về các yêu cầu phi chức năng:

*Bảng 3. 2: Yêu cầu phi chức năng*

|  |  |
| --- | --- |
| Hiệu Suất | Thời gian đáp ứng: Hệ thống cần đảm bảo thời gian đáp ứng nhanh chóng để người dùng trải nghiệm mượt mà và không bị gián đoạn. |
| Khả năng Mở Rộng: Hệ thống phải có khả năng mở rộng linh hoạt để chịu được tăng cường người dùng và dữ liệu. |
| Bảo Mật | Quản lý Quyền Truy Cập: Hệ thống phải cung cấp quản lý quyền truy cập chi tiết để đảm bảo mỗi người dùng chỉ có quyền truy cập vào thông tin và chức năng cần thiết. |
| Bảo Mật Dữ Liệu: Dữ liệu cá nhân của nhân viên phải được bảo vệ một cách chặt chẽ, bao gồm cả quá trình truyền và lưu trữ. |
| Tương Thích Diverse | Hệ thống cần tương thích với nhiều loại trình duyệt web để đảm bảo sự tiện lợi và sẵn sàng sử dụng cho người dùng. |
| Dễ Dàng Sử Dụng | Giao Diện Người Dùng Thân Thiện: Giao diện người dùng phải được thiết kế sao cho dễ sử dụng và hiểu quả, giúp người dùng tìm kiếm thông tin một cách thuận tiện. |
| Hỗ Trợ Ngôn Ngữ Đa Dạng: Hệ thống cần hỗ trợ nhiều ngôn ngữ để phục vụ người dùng có ngôn ngữ gốc khác nhau. |
| Bảo Trì và Hỗ Trợ | Dễ Dàng Bảo Trì: Hệ thống cần được thiết kế sao cho việc bảo trì và cập nhật có thể thực hiện một cách dễ dàng mà không làm ảnh hưởng đến hoạt động hàng ngày. |
| Hỗ Trợ Người Dùng: Cần có một hệ thống hỗ trợ trực tuyến và tài liệu chi tiết để người dùng có thể tìm kiếm và giải quyết vấn đề một cách tự nhiên. |
| Sao Lưu và Phục Hồi Dữ Liệu | Hệ thống cần có quy trình tự động sao lưu và khả năng phục hồi dữ liệu để đảm bảo tính ổn định và an toàn của thông tin. |

## Sơ đồ usecase

*Hình 3. 1: Sơ đồ đặc tả ca sử dụng*

## Đặc tả usecase

## Biểu đồ hoạt động các ca sử dụng

# CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

## Biểu đồ tuần tự các ca sử dụng

## Kiến trúc hệ thống backend

Kiến trúc hệ thống MERN là một giải pháp phần mềm mạnh mẽ dựa trên sự kết hợp của bốn công nghệ chủ chốt: MySQL, Express.js, Node.js và React.js. Đây là một trong những mô hình phát triển ứng dụng web được ưa chuộng nhất hiện nay, nhờ vào tính linh hoạt, hiệu suất và khả năng mở rộng mà từng công nghệ mang lại.

MySQL đóng vai trò là hệ quản trị cơ sở dữ liệu quan hệ chủ đạo trong kiến trúc này. Với cấu trúc bảng và quan hệ giữa chúng, MySQL cung cấp khả năng lưu trữ dữ liệu an toàn và có tính nhất quán cao. Điểm mạnh của MySQL nằm ở sự ổn định và khả năng xử lý các giao transactioan tốt, làm cho nó trở thành lựa chọn phổ biến cho các ứng dụng yêu cầu độ tin cậy cao và tính nhất quán dữ liệu.

Express.js, một framework backend cho Node.js, đảm nhận vai trò xử lý logic của ứng dụng. Với Express.js, việc xây dựng các API và quản lý các yêu cầu HTTP trở nên đơn giản hơn, nhờ vào các tính năng middleware mạnh mẽ và khả năng định tuyến linh hoạt.

Node.js là một môi trường thực thi mã nguồn mở cho phép chạy JavaScript trên server, giúp ứng dụng có thể xử lý hàng ngàn kết nối đồng thời một cách hiệu quả. Với khả năng xử lý bất đồng bộ, Node.js cho phép ứng dụng thực hiện các hoạt động không đồng bộ mà không làm chậm hệ thống.

React.js, là một thư viện JavaScript phát triển bởi Facebook, được sử dụng cho phần giao diện người dùng của ứng dụng. Với React.js, việc xây dựng các giao diện người dùng đơn trang (SPA) trở nên đơn giản hơn bao giờ hết, nhờ vào các thành phần UI tái sử dụng và khả năng quản lý trạng thái ứng dụng một cách hiệu quả.

## Thiết kế cơ sở dữ liệu

## Thiết kế giao diện

# CHƯƠNG 5: KẾT LUẬN

## Kết quả đạt được

## Định hướng phát triển

**TÀI LIỆU THAM KHẢO**

[1] <https://reactjs.org/docs/deployment.html>

[2] <https://nodejs.org/en/docs/guides/nodejs-docker-webapp/>

[3] "Mysql: The Definitive Guide" by Shannon Bradshaw, O'Reilly Media, 2022.

[4] Christopher Buecheler, "Node.js, Mysql, and React.js Full Stack Web Development," Packt Publishing, 2015.

[5] Brad Traversy, "MERN Stack Front To Back: Full Stack React, Redux & Node.js," Udemy, 2021.