**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**BÙI NGỌC TOÀN**

**KHÓA 16**

**HỆ ĐÀO TẠO: KỸ SƯ DÂN SỰ**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG HỆ THỐNG HỌC TRỰC TUYẾN, KIỂM TRA, ĐÁNH GIÁ CHẤT LƯỢNG ONLINE**

**NĂM 2022**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**BÙI NGỌC TOÀN**

**KHÓA 16**

**HỆ ĐÀO TẠO: KỸ SƯ DÂN SỰ**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**MÃ SỐ: 52480103**

**XÂY DỰNG HỆ THỐNG HỌC TRỰC TUYẾN, KIỂM TRA, ĐÁNH GIÁ CHẤT LƯỢNG ONLINE**

***Cán bộ hướng dẫn khoa học: GVC. TS Nguyễn Thị Hiền***

**NĂM 2022**

|  |  |
| --- | --- |
| HỌC VIỆN KỸ THUẬT QUÂN SỰ  **KHOA: CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN: CÔNG NGHỆ PHẦN MỀM** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên:Bùi Ngọc Toàn Lớp: CNPM16 Khóa: 16

Ngành: Công nghệ phần mềm Chuyên ngành: Công nghệ phần mềm

1. Tên đề tài: Xây dựng hệ thống học trực tuyến, kiểm tra, đánh giá chất lượng online

2. Các số liệu ban đầu:

- Quyết định giao nộp đồ án tốt nghiệp đại học – Học viện Kỹ thuật Quân sự

- Tài liệu tham khảo

3. Nội dung bản thuyết minh:

- Phần mở đầu

+ Đặt vấn đề: Nêu thực trạng, tầm ảnh hưởng của việc dạy học trực tuyến và kiểm tra, đánh giá chất lượng online.

+ Lý do chọn đề tài: Đưa ra những lý luận để thấy sự cần thiết của hệ thống học trực tuyến.

+ Mục tiêu nghiên cứu   
+ Đối tượng nghiên cứu

+ Nhiệm vụ nghiên cứu

+ Phương pháp nghiên cứu

+ Phạm vi nghiên cứu

- Phần nội dung

+ Chương 1: Mô tả hệ thống  
+ Chương 2: Đặc tả yêu cầu  
+ Chương 3: Thiết kế và xây dựng hệ thống

+ Chương 4: Cài đặt hệ thống

- Phần kết luận  
+ Kết quả đạt được   
+ Hướng phát triển

Phần danh mục tài liệu tham khảo

4. Số lượng, nội dung các bản vẽ:

Hình ảnh, bảng biểu được trải đều trong 3 chương.

5. Cán bộ hướng dẫn:

- Họ và tên: Nguyễn Thị Hiền

- Cấp bậc:

- Học hàm, học vị: GVC, TS

- Đơn vị:

- Hướng dẫn toàn bộ

Ngày giao: Ngày hoàn thành:

*Hà Nội, ngày ........ tháng ……..năm ……..*

**Chủ nhiệm bộ môn Cán bộ hướng dẫn**

(Ký, ghi rõ họ tên, học hàm, học vị)

**Học viên thực hiện**

Đã hoàn thành và nộp đồ án ngày tháng năm 2022

(Ký và ghi rõ họ tên)

**Toàn**

**Bùi Ngọc Toàn**

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Mô hình kiến trúc tổng thể 9](#_Toc103026281)

[Hình 2.2 Biểu đồ use case tổng quan 10](#_Toc103026282)

[Hình 2.3 Biểu đố use case khách ghé thăm hệ thống 11](#_Toc103026283)

[Hình 2.4 Biểu đồ usecase thành viên của hệ thống 12](#_Toc103026284)

[Hình 2.5 Biểu đố usecase người tạo khóa học 13](#_Toc103026285)

[Hình 2.6 Biểu đồ Usecase cho Admin 14](#_Toc103026286)

[Hình 3.1Mô hình kiến trúc hệ thống 29](#_Toc103026287)

[Hình 3.2 Cách thức hoạt động Restful 30](#_Toc103026288)

[Hình 3.3 Sơ đồ tổng quát 31](#_Toc103026289)

[Hình 3.4 Giao diện đăng nhập 46](#_Toc103026290)

[Hình 3.5 Giao diện đăng ký thành viên 47](#_Toc103026291)

[Hình 3.6 Giao diện trang chủ 48](#_Toc103026292)

[Hình 3.7 Giao diện xem chi tiết khóa học 48](#_Toc103026293)

[Hình 3.8 Giao diện danh sách khóa học đã tạo 49](#_Toc103026294)

[Hình 3.9 Giao diện tạo tên khóa học 49](#_Toc103026295)

[Hình 3.10 Giao diện lựa chọn thể loại khóa học 50](#_Toc103026296)

[Hình 3.11 Giao diện tạo mục đích của môn học 51](#_Toc103026297)

[Hình 3.12 Giao diện 52](#_Toc103026298)

[Hình 3.13 Giao diện nhập tiêu đề bài học 52](#_Toc103026299)

[Hình 3.14 Giao diện thêm video cho bài học 52](#_Toc103026300)

[Hình 3.15Giao diện tạo mô tả cho bài học 53](#_Toc103026301)

[Hình 3.16 Giao diện thêm file tài liệu cho bài học 53](#_Toc103026302)

[Hình 3.17Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài 54](#_Toc103026303)

[Hình 3.18Giao diện tạo câu hỏi cho bài tập trách nghiệm 54](#_Toc103026304)

[Hình 3.19Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài 55](#_Toc103026305)

[Hình 3.20Giao diện nhập nội dung cho bài tập trắc nghiệm 55](#_Toc103026306)

[Hình 3.21 Giao diện tạo học phí cho khóa học 56](#_Toc103026307)

[Hình 3.22 Giao diện tạo trang đích khóa học 57](#_Toc103026308)

[Hình 3.23 Giao diện thêm vào giỏ hàng 58](#_Toc103026309)

[Hình 3.24 Giao diện giỏ hàng 58](#_Toc103026310)

[Hình 3.25Giao diện cổng thanh toán 59](#_Toc103026311)

[Hình 4.1 Bảng so sánh hiệu năng giữa các framwork Python 60](#_Toc103026312)

[Hình 4.2 Bảng so sánh tốc độ truy vấn SQL và Mongo 64](#_Toc103026313)

[Hình 4.3 Sơ đồ Redis server 65](#_Toc103026314)

[Hình 4.4 Giao thức TCP 69](#_Toc103026315)

[Hình 4.5 Giao thức UDP 70](#_Toc103026316)

[Hình 4.6 Cấu trúc Websocket 71](#_Toc103026317)

[Hình 4.7 Docker 73](#_Toc103026318)

[Hình 4.8 Quy trình thực thi 74](#_Toc103026319)

[Hình 4.9 Cơ chế hoạt động của Angular 78](#_Toc103026320)

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc103024837)

[CHƯƠNG 1 KHẢO SÁT HỆ THỐNG 5](#_Toc103024838)

[1.1. Tổng quan về Dạy học trực tuyến (E-learning) 5](#_Toc103024839)

[1.2. Khảo sát hệ thống 6](#_Toc103024840)

[1.3. Mô tả nghiệp vụ hệ thống 7](#_Toc103024841)

[CHƯƠNG 2 ĐẶC TẢ HỆ THỐNG 9](#_Toc103024842)

[2.1. Mô hình kiến trúc tổng thể 9](#_Toc103024843)

[2.2. Mô hình use case 9](#_Toc103024844)

[CHƯƠNG 3 THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG 29](#_Toc103024845)

[3.1. Kiến trúc chung hệ thống 29](#_Toc103024847)

[3.2. Thiết kế các lớp 31](#_Toc103024848)

[3.3. Mối quan hệ các lớp 31](#_Toc103024849)

[3.4. Thiết kế cơ sở dữ liệu 31](#_Toc103024850)

[3.5. Thiết kế giao diện 46](#_Toc103024851)

[CHƯƠNG 4 60](#_Toc103024852)

[4.1 Công nghệ sử dụng 60](#_Toc103024853)

[4.2 Đánh giá 79](#_Toc103024854)

[KẾT LUẬN 80](#_Toc103024855)

[1. Kết quả đạt được 80](#_Toc103024856)

[2. Hướng phát triển 80](#_Toc103024857)

[TÀI LIỆU THAM KHẢO 81](#_Toc103024858)

# MỞ ĐẦU

Để hoàn thành đồ án này, em xin chân thành gửi lời cảm ơn đến Cô giáo Nguyễn Thị Hiền, giảng viên trực tiếp phụ trách hướng dẫn đề tài đồ án. Sự hướng dẫn tận tình của cô đã giúp em có thêm nhiều kiến thức, khắc phục được những lỗi em thường mắc phải để đồ án được hoàn thiện tốt hơn. Qua những hướng dẫn tận tâm của cô đã giúp chúng em hiểu sâu hơn về những kiến thức đã học, và trau dồi thêm những kiến thức mới bổ ích.

Em xin chân thành gửi lời cảm ơn quý thầy cô trong bộ môn cũng như tất cả các thầy cô trong khoa Công nghệ thông tin của Học Viện Kỹ Thuật Quân Sự đã tận tình giúp đỡ và trang bị cho em những kiến thức cơ bản, cần thiết để em có thể hoàn thành tốt đồ án này.

Ngoài ra, em xin gửi lời cảm ơn tới Ban Giám hiệu Học Viện Kỹ Thuật Quân Sự đã tạo điều kiện để em thực hiện tốt đồ án này.

Mặc dù có nhiều cố gắng để thực hiện đồ án tốt nghiệp một cách hoàn chỉnh nhất. Song do năng lực còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót nhất định.

Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

*Em xin chân thành cảm ơn!*

1. **Đặt vấn đề**

Với sự phát triển không ngừng của khoa học và công nghệ trong thời đại 4.0, việc sử dụng internet trở nên phổ biến và dễ tiếp cận hơn rất nhiều so với trước đó. Theo đó, chúng ta đang hướng đến việc sử dụng internet để tối ưu hóa mọi công việc trong cuộc sống. Bây giờ, chúng ta có thể “gặp mặt”, nói chuyện với nhau thông qua video call, không còn vấn đề về khoảng cách địa lý, chỉ cần có internet, mọi người có thể kết nối với nhau mọi lúc.

Với tình hình Covid-19 đã và đang xảy ra trên toàn cầu, việc đi lại trở nên khó khăn hơn. Đặc biệt là việc học sinh đến trường để học tập sẽ cần thời gian để chúng ta có thể kiểm soát được dịnh bệnh, đảm bảo an toàn cho giáo viên và học sinh khi đến trường. Trong tình hình đó, việc dạy và học online là lựa chọn phù hợp nhất. Và từ đó, chúng ta sẽ hướng tới việc dạy và học online có thể trở thành hình thức dạy học chính thức trong thời đại mới.

1. **Lý do chọn đề tài**

Như đã trình bày ở trên, việc dạy và học online đóng một vài trò quan trọng trong tình hình diễn biến dịch đang diễn ra rất căng thẳng. Các trung tâm dạy học hiện nay đã phải đóng cửa trong một thời gian dài, ảnh hưởng rất lớn đến việc duy trì và phát triển. Ngoài ra, việc học online có rất nhiều ưu điểm:

• **Tăng tính chủ động cho người học**: Dạy học trực tuyến giúp người học chủ động hơn trong việc tìm hiểu, lựa chọn môn học phù hợp, thời gian và địa điểm học tập linh động hơn.

• **Tiết kiệm chi phí và thời gian**: Các cơ sở đào tạo tiết kiệm được chi phí cho việc xây dựng cơ sở vật chất, trang thiết bị, thuê giáo viên… Nhờ đó các khóa học trực tuyến sẽ giảm chi phí học nên phù hợp với cả những người học không dư dả về tài chính. Ngoài ra, học trực tuyến cũng tránh làm hao tổn thời gian và chi phí đi lại.

• **Việc dạy và học trở nên tốt hơn**: Giảng viên có thể sử dụng hình ảnh, âm thanh và video để truyền đạt nội dung học tập đến người học thêm hấp dẫn và sinh động hơn. Học viên có thể xem lại các video để củng cố kiến thức và xem lại những nội dung chưa kịp nghe.

1. **Mục tiêu nghiên cứu**

* Đưa ra giải pháp giúp việc dạy học trực tuyến trở nên dễ dàng hơn và đem lại những hiệu quả to lớn hơn.
* Ứng dụng công nghệ thông tin vào việc xây dựng hệ thống dạy học trực tuyến, kiểm tra, đánh giá chất lượng online

1. **Đối tượng nghiên cứu**

* Nghiên cứu về các hệ thống dạy học trực tuyến
* Nghiên cứu phương pháp xây dựng hệ thống cho trung tâm đào tạo

1. **Nhiệm vụ nghiên cứu**

* Làm rõ vai trò của hệ thống đối với việc dạy học trực tuyến
* Xây dựng phần mềm dạy học trực tuyến

1. **Phương pháp nghiên cứu**

* Phương pháp phân tích

+ Tham khảo phần mềm hiện có Udemy

+ Tìm kiếm các thông tin về nghiệp vụ dạy học trực tuyến và các lý thuyết liên quan

+ Tìm hiểu các công nghệ sử dụng

* Phương pháp tổng hợp, quan sát:

+ Khảo sát trên các website dạy học trực truyến phổ biến như F8, udemy, … để có cái nhìn thực tế và hiểu rõ nghiệp vụ.

* Phương pháp thực hành

+ Lập trình thử nghiệm

1. **Phạm vi nghiên cứu**

* Tham khảo các phần mềm hiện có Udemy

1. **Phân công công việc**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | Nội dung công việc |
| 1 | Bùi Ngọc Toàn | Thực hiện xây dựng các chức năng:   * Xác thực User * Hiên thị khóa học * Chức năng Thanh Toán * Xem chi tiết nội dung khóa học, * … |
| 2 | Nguyễn Quang Dự | Thực hiện xây dựng các chức năng:   * Chức năng tạo khóa học của User * Chức năng kiểm tra đánh giá đối với học viên |

# KHẢO SÁT HỆ THỐNG

## Tổng quan về Dạy học trực tuyến (E-learning)

### Khái niệm về E-learning

Dạy học trực tuyến (E-learning) là hình thức giảng dạy và học tập ở các lớp học trên Internet. Người dạy và người học sẽ sử dụng phần mềm nền tảng học trực tuyến, ứng dụng truyền âm thanh, hình ảnh và các thiết bị thông minh (laptop, smartphone, máy tính bảng,...).

Các bài giảng, tài liệu (dưới dạng văn bản, hình ảnh, video…) được đưa lên các nền tảng và người dùng có thể dễ dàng truy cập và học mọi lúc mọi nơi. Bên cạnh đó còn có các khóa học cùng thời gian thực có sự tham gia và tương tác giữa giáo viên và học viên.

### Đặc điểm của phương pháp đào tạo qua E-learning

E-learning là phương thức đào tạo đang phát triển mạnh mẽ hiện nay. Công tác đào tạo có thể triển khai toàn bộ qua hệ thống này, từ khâu đăng kí khóa học đến khâu kiểm tra, đánh giá chất lượng cho học viên.

Ưu điểm của phương pháp đào tạo qua E-learning:

* Không giới hạn không gian và thời gian: Người học có thể học bất cứ nơi đâu và khi nào họ muốn vì nội dung học được chuyển tải tất cả qua mạng máy tính tới máy tính của người học cũng như các thiết bị thông minh khác như smart phone, Ipad,…bằng việc sử dụng hệ thống hỗ trợ E-learning.
* Tiết kiệm thời gian: Người học chủ động hơn về thời gian của mình để tham gia khóa học cũng như chủ động thời gian công tác.
* Tiết kiệm chi phí: Giúp giảm chi phí đi lại và các chi phí khác cho người học và người dạy.
* Chủ động trong kiểm tra đánh giá: Học viên có thể tham gia làm bài kiểm tra linh hoạt trong thời gian qui định, có thể thực hiện nhiều lần nhằm củng cố lượng kiến thức cần thiết qua hệ thống E-learning.

Yêu cầu đối với học viên:

* Trang bị các thiết bị có kết nối internet.
* Biết kết nối mạng internet và trình duyệt web.
* Kỹ năng đánh máy và sử dụng máy tính đạt trình A.
* Tính tự giác trong học tập.

Do thời gian người học tự học và tự nghiên cứu chiếm 50% tổng thời gian, cũng như việc thực hiện các đợt kiểm tra thông qua hệ thống E-learning. Vì vậy tính tự giác trong học tập là vô cùng cần thiết. Bên cạnh đó, người học cần phải tự tìm hiểu thêm các tài liệu liên quan đến khóa học ngoài các tài liệu mà trường cung cấp để nâng cao chất lượng học tập, đồng thời cần phải học hỏi thêm các kinh nghiệm từ những người học khác thông qua các diễn đàn.

### Mô hình hệ thống E-learning

## Khảo sát hệ thống

Website Udemy là website cung cấp các khóa học trực tuyến ở nhiều lĩnh vực như Phát triển phần mềm, marketing, design, fitness, … để giúp cho người dùng phát triển các kỹ năng chuyên môn phục vụ cho công việc và cuộc sống. Không giống với đa số các nền tảng học online tập trung vào những khóa học do các trường đại học, trung tâm lập nên, Udemy có tính mở, hướng đến việc cung cấp kiến thức và kinh nghiệm thực tiễn. Cũng vì thế mà chính các học viên của Udemy cũng có thể trở thành một giảng viên và xuất bản khóa học của chính mình nếu họ có được lượng kiến thức và kỹ năng đủ nhiều.

Các tính năng chính:

* Đăng ký
* Đăng nhập bằng email và password đã đăng ký trên hệ thống (Hoặc có thể đăng nhập bằng tài khoản gmail, facebook, apple)
* Xem danh sách các khóa học theo thể loại và chủ đề
* Xem thông tin chi tiết khóa học (giảng viên, đánh giá, lộ trình học, …)
* Giỏ hàng, thanh toán, danh sách ưa thích
* Quản lý khóa học
* Báo cáo thống kê

Ưu điểm:

* Số lượng khóa học khổng lồ với nhiều chủ đề khác nhau
* Ai cũng có thể tạo khóa học, các khóa học được đánh giá thông qua chất lượng của nội dung.
* Các khóa học hỗ trợ hơn 60 ngôn ngữ.

Nhược điểm:

* Số lượng khóa học quá nhiều nên việc lựa chọn khóa học gây mất nhiều thời gian.
* Bởi vì bất kỳ ai cũng có thể tạo khóa học dẫn đến nhiều khóa học có chất lượng không thật sự tốt.

## Mô tả nghiệp vụ hệ thống

### Đối tượng

Các trung tâm đào tạo có nhu cầu giảng dạy các khóa học trực tuyến và những người có nhu cầu học các khóa học trực tuyến.

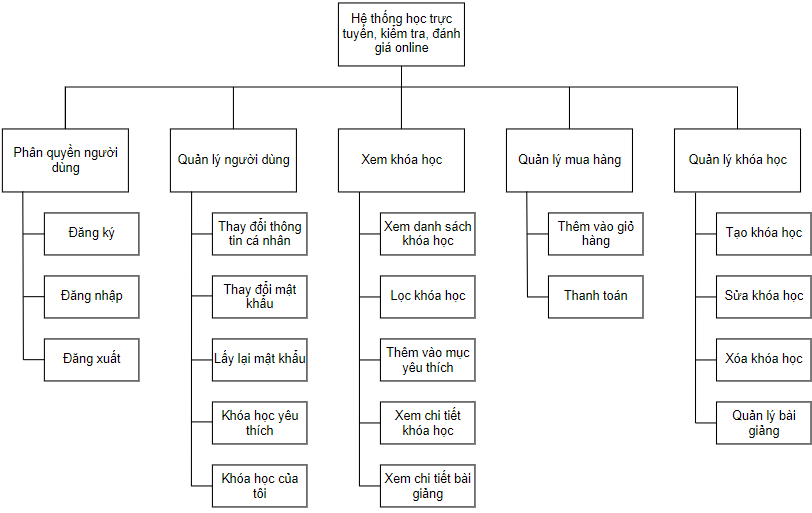
### Các chức năng

Bảng 1.1 Bảng các chức năng

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả** |
| 1 | Quản lý người dùng và phân quyền | * Tạo các quyền từ các chức năng hệ thống cung cấp * Tạo cáo user với quyền hạn khác nhau |
| 2 | Quản lý mua hàng | Người dùng xem thông tin giới thiệu về khóa học  Người dùng có thể thêm khóa học vào mục yêu thích hoặc thêm vào giỏ hàng  Sau khi người dùng thanh toán giỏ hàng có thể xem chi tiết khóa học |
| 3 | Quản lý khóa học | * Người dùng thêm/xem/sửa/xóa các khóa học. |
| 4 | Báo cáo thống kê | Người dùng xem những thống kê của hệ thống về số lượng khóa học đã bán, những khóa học được mua nhiều, … |

# ĐẶC TẢ HỆ THỐNG

## Mô hình kiến trúc tổng thể

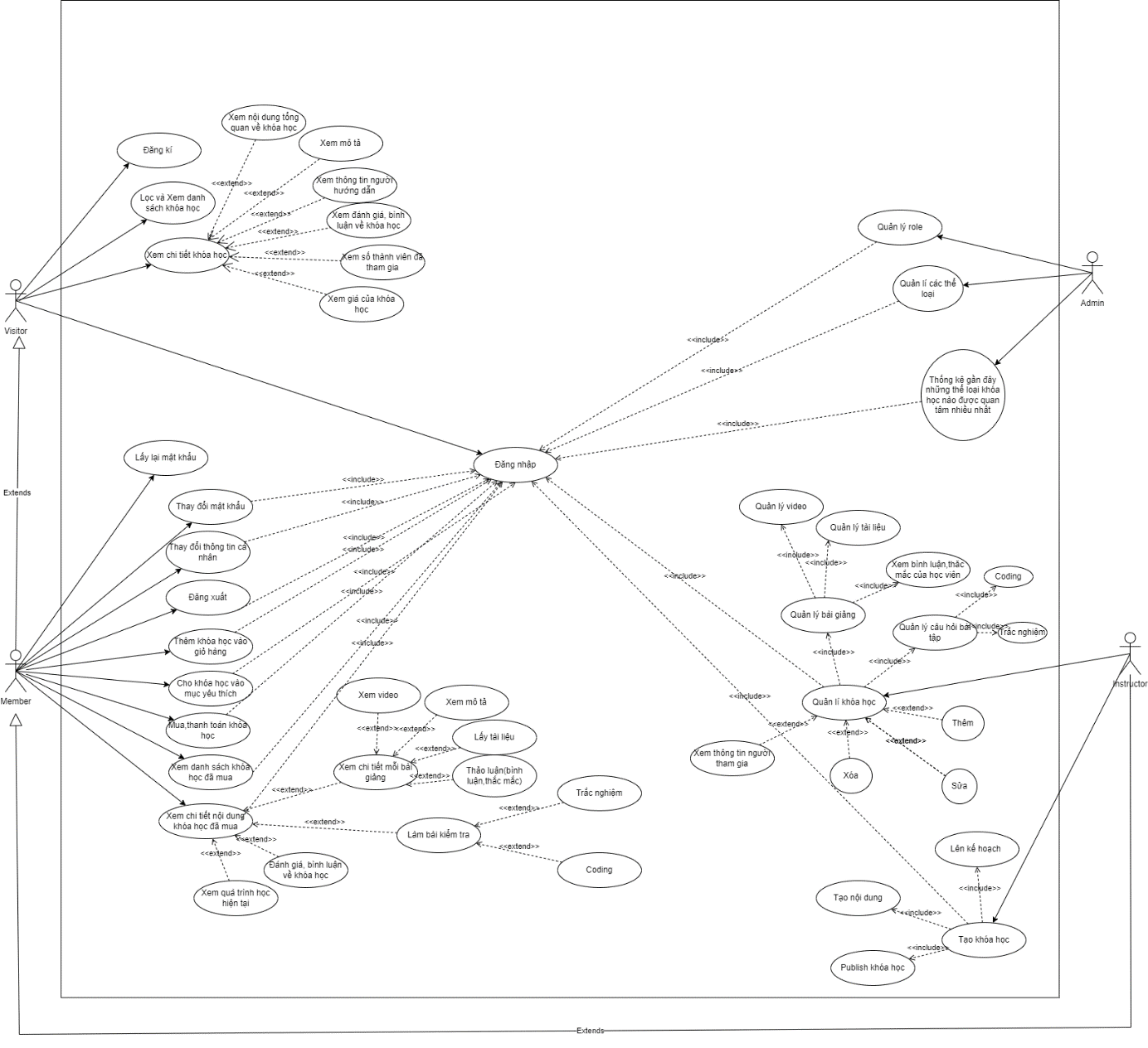


Hình 2.1 Mô hình kiến trúc tổng thể

## Mô hình use case

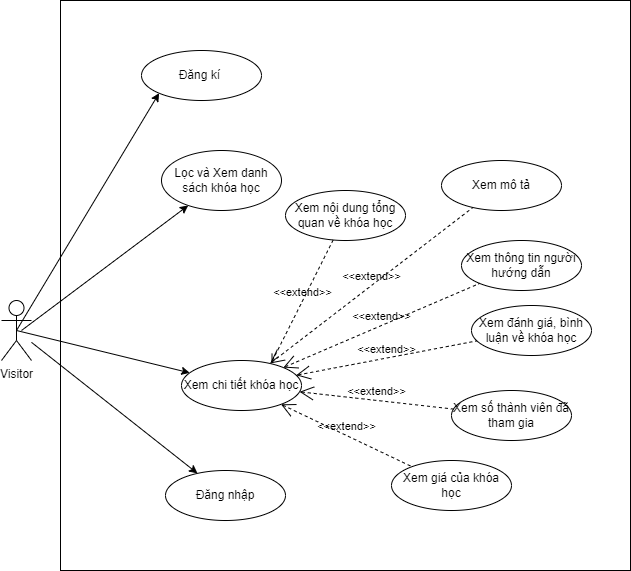
### ***Biểu đồ use case***

* Biểu đồ use case tổng quan

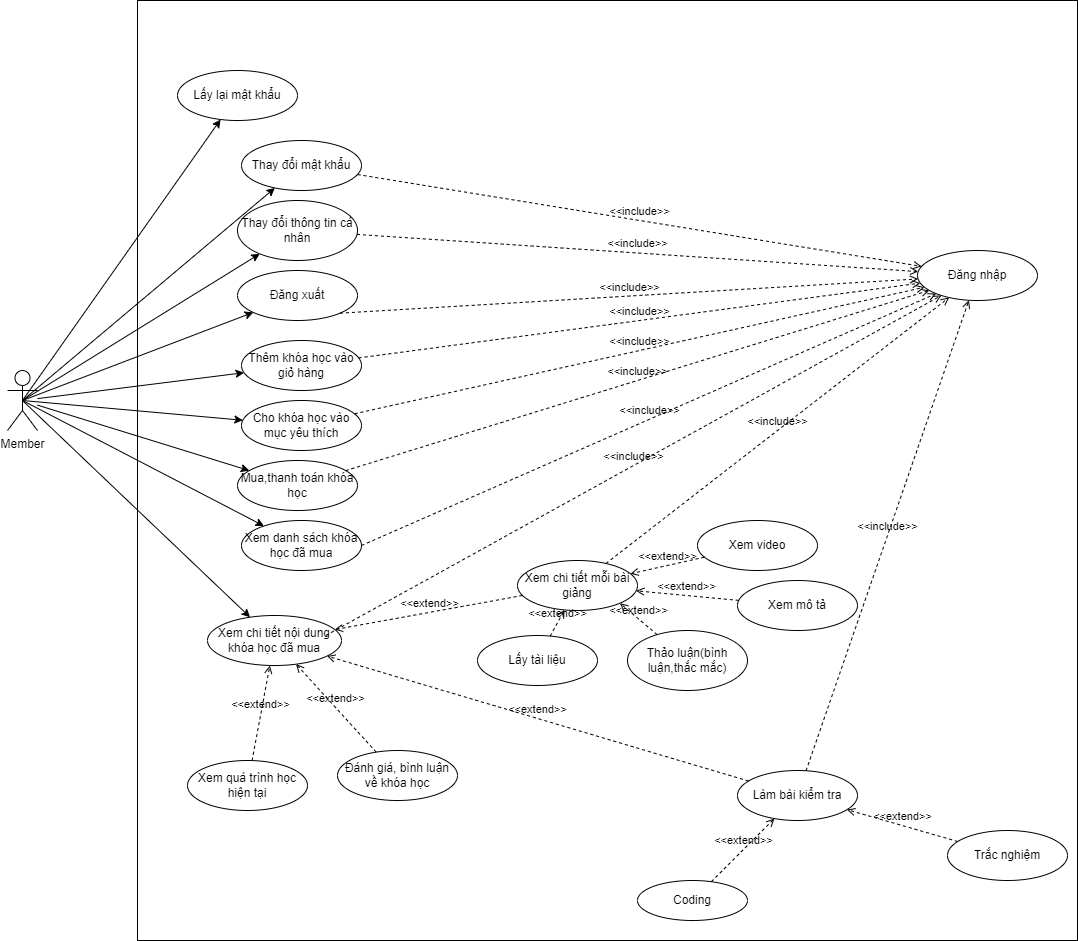


Hình 2.2 Biểu đồ use case tổng quan

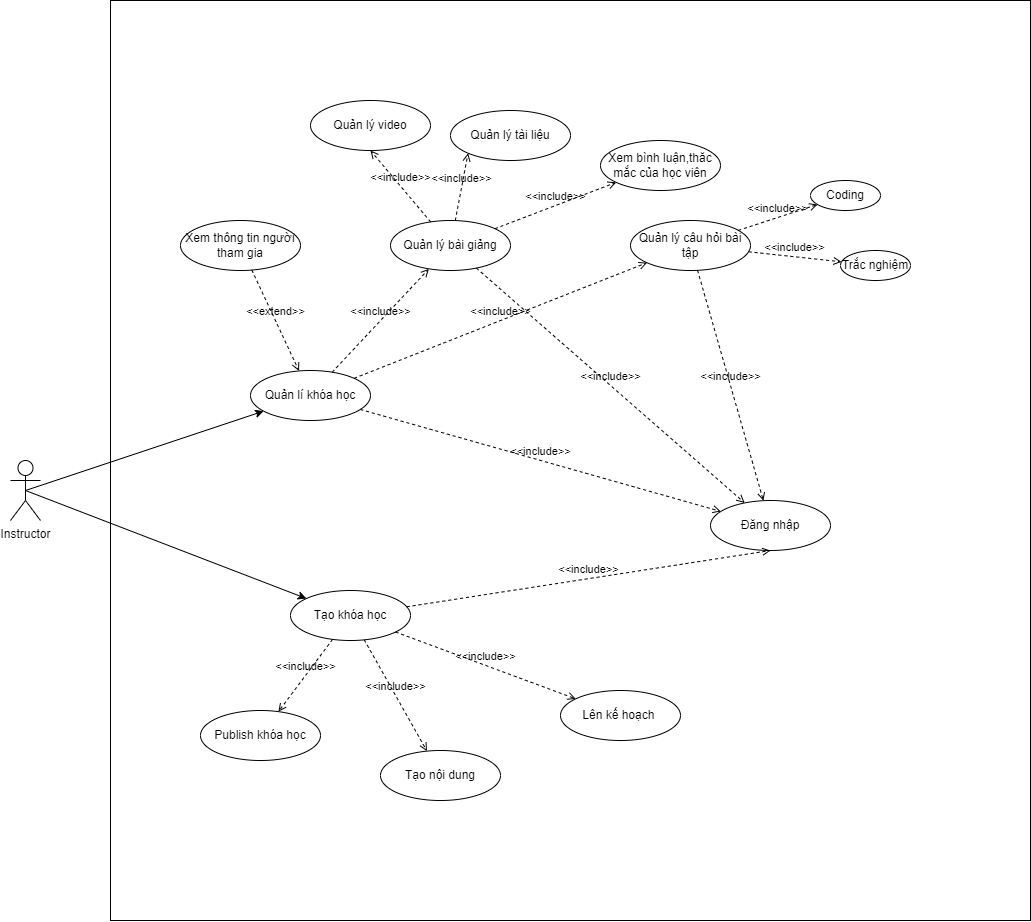
* Biểu đồ use case chi tiết



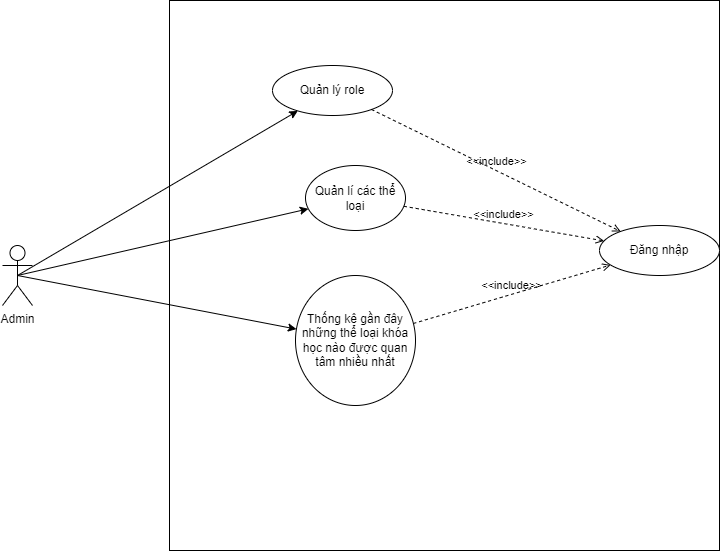
Hình 2.3 Biểu đố use case khách ghé thăm hệ thống



Hình 2.4 Biểu đồ usecase thành viên của hệ thống



Hình 2.5 Biểu đố usecase người tạo khóa học



Hình 2.6 Biểu đồ Usecase cho Admin

### ***Mô tả use case***

#### Usecase đăng ký

|  |  |
| --- | --- |
| Mục đích | Người dùng đăng ký tài khoản để trở thành thành viên của hệ thống |
| Mô tả | Cho phép người dùng đăng ký tài khoản để có thể sử dụng đầy đủ các chức năng dành cho thành viên của hệ thống |
| Tác nhân | Guest |
| Điều kiện | Chưa đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Nhập họ * Nhập tên * Nhập email * Nhập mật khẩu * Nhập mật khẩu xác nhận |
| Luồng sự kiện phụ | * Đăng ký không thành công vì chưa nhập đủ các trường bắt buộc * Đăng ký không thành công vì thông tin không đúng định dạng * Đăng ký không thành công vì email đã được sử dụng |
| Điều kiện sau | Đăng ký thành công và có thể đăng nhập vào hệ thống |

#### Usecase đăng nhập

|  |  |
| --- | --- |
| Mục đích | Người dùng đăng nhập thành công để sử dụng đầy đủ các chức năng của hệ thống |
| Mô tả | Cho phép người dùng đăng ký tài khoản để có thể sử dụng đầy đủ các chức năng của hệ thống |
| Tác nhân | Member, Instructor, Admin |
| Điều kiện | Truy cập vào phần mềm kiểm tra và đánh giá chất lượng |
| Luồng sự kiện chính | * Nhập email * Nhập mật khẩu |
| Luồng sự kiện phụ | * Đăng nhập không thành công vì email không tồn tại * Đăng nhập không thành công vì mật khẩu sai |
| Điều kiện sau | Đăng nhập thành công và bắt đầu sử dụng các chức năng của hệ thống |

#### Usecase đăng xuất

|  |  |
| --- | --- |
| Mục đích | Người dùng đăng xuất khỏi hệ thống |
| Mô tả | Cho phép người dùng đăng xuất tài khoản khỏi hệ thống |
| Tác nhân | Member, Instructor, Admin |
| Điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Luồng sự kiện chính | * Nhấn nút đăng xuất |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Đăng xuất thành công khỏi hệ thống |

#### Usecase thay đổi thông tin cá nhân

|  |  |
| --- | --- |
| Mục đích | Người dùng thay đổi thông tin cá nhân của tài khoản |
| Mô tả | Cho phép người dùng cập nhật thông tin cá nhân của tài khoản |
| Tác nhân | Member, Instructor, Admin |
| Điều kiện | * Người dùng đăng nhập thành công vào hệ thống * Truy cập tới trang thay đổi thông tin cá nhân |
| Luồng sự kiện chính | * Nhập họ * Nhập tên * Tải ảnh đại diện |
| Luồng sự kiện phụ | * Thay đổi không thành công do nhập không đúng định dạng yêu cầu * Thay đổi không thành công do nhập thiếu trường thông tin bắt buộc |
| Điều kiện sau | Thay đổi thông tin cá nhân thành công |

#### Usecase thay đổi mật khẩu

|  |  |
| --- | --- |
| Mục đích | Người dùng thay đổi mật khẩu đăng nhập để đảm bảo tính bảo mật của tài khoản |
| Mô tả | Cho phép người dùng thay đổi mật khẩu đăng nhập |
| Tác nhân | Member, Instructor, Admin |
| Điều kiện | * Người dùng đăng nhập thành công vào hệ thống * Truy cập tới trang thay đổi mật khẩu |
| Luồng sự kiện chính | * Nhập mật khẩu cũ * Nhập mật khẩu mới * Nhập lại mật khẩu mới |
| Luồng sự kiện phụ | * Đổi mật khẩu không thành công do nhập sai mật khẩu cũ * Đổi mật khẩu không thành công do mật khẩu mới không đúng định dạng * Đổi mật khẩu không thành công do mật khẩu mới nhập lại không trùng với mật khẩu mới |
| Điều kiện sau | Người dùng thay đổi mật khẩu thành công và đăng nhập thành công bằng mật khẩu mới |

#### Usecase lấy lại mật khẩu

|  |  |
| --- | --- |
| Mục đích | Người dùng lấy lại mất khẩu đăng nhập |
| Mô tả | Cho phép người dùng lấy lại mật khẩu đăng nhập |
| Tác nhân | Member, Instructor, Admin |
| Điều kiện | Truy cập vào trang đăng nhập |
| Luồng sự kiện chính | * Nhấn nút Quên mật khẩu * Nhập email đăng nhập * Mật khẩu mới sẽ được gửi vào email của người dùng |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Người dùng nhận được mật khẩu mới và đăng nhập thành công với mật khẩu mới |

#### Usecase khóa học yêu thích

|  |  |
| --- | --- |
| Mục đích | Người dùng xem danh sách các khóa học đã được lưu vào mục yêu thích |
| Mô tả | Hiển thị danh sách các khóa học đã lưu vào mục yêu thích |
| Tác nhân | Member |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập trang khóa học yêu thích |
| Luồng sự kiện chính | * Hiển thị danh sách các khóa học đã lưu vào mục yêu thích |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Hiển thị thành công danh sách các khóa học đã lưu vào mục yêu thích của người dùng |

#### Usecase khóa học của tôi

|  |  |
| --- | --- |
| Mục đích | Người dùng xem danh sách các khóa học đã mua thành công |
| Mô tả | Hiển thị danh sách các khóa học đã mua thành công của người dùng |
| Tác nhân | Member |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập trang khóa học của tôi |
| Luồng sự kiện chính | * Hiển thị danh sách các khóa học đã mua thành công của người dùng |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Hiển thị thành công danh sách các khóa học đã mua của người dùng |

#### Usecase xem danh sách khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng có thể xem danh sách các khóa học có trong hệ thống |
| Mô tả | Hiển thị tất cả danh sách khóa học |
| Tác nhân | Guest, Member |
| Điều kiện | Truy cập vào phần mềm kiểm tra và đánh giá chất lượng |
| Luồng sự kiện chính | * Truy cập vào trang chủ |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Hiển thị tất cả danh sách khóa học của hệ thống |

#### Usecase lọc khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng có thể lọc danh sách khóa học |
| Mô tả | Cho phép người dùng lọc danh sách khóa học theo các tiêu chí |
| Tác nhân | Guest, Member |
| Điều kiện | Truy cập vào trang chủ phần mềm kiểm tra và đánh giá chất lượng |
| Luồng sự kiện chính | * Nhập tên khóa học cần tìm kiếm * Chọn loại khóa học từ danh sách * Chọn sắp xếp danh sách theo thứ tự A-Z, Z-A, giá từ thấp-cao, giá từ cao-thấp |
| Luồng sự kiện phụ | * Không tìm thấy khóa học đã tìm kiếm |
| Điều kiện sau | Hiển thị danh sách khóa học đã lọc theo các tiêu chí đã chọn |

#### Usecase thêm vào mục yêu thích

|  |  |
| --- | --- |
| Mục đích | Người dùng có thể lưu những khóa học chưa mua để có thể tìm kiếm dễ dàng hơn vào những lần sau |
| Mô tả | Cho phép người dùng lưu khóa học vào mục khóa học yêu thích |
| Tác nhân | Member |
| Điều kiện | Truy cập vào trang chủ  Chọn xem chi tiết khóa học |
| Luồng sự kiện chính | * Nhấn nút thêm vào mục yêu thích |
| Luồng sự kiện phụ | * Không hiển thị nút thêm vào mục yêu thích |
| Điều kiện sau | Khóa học được hiển thị trong danh sách khóa học yêu thích |

#### Usecase xem chi tiết khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng có thể xem thông tin chi tiết của khóa học |
| Mô tả | Cho phép người dùng xem các thông tin chi tiết về khóa học |
| Tác nhân | Guest, Member |
| Điều kiện | * Truy cập vào trang chi tiết khóa học |
| Luồng sự kiện chính | * Nội dung tổng quan về khóa học * Mô tả khóa học * Thông tin người hướng dẫn * Xem số người đã mua khóa học * Xem giá của khóa học * Xem chi tiết bài giảng(video, mô tả, tài liệu, thảo luận) * Kiểm tra đánh giá(trắc nghiệm, code) * Đánh giá, bình luận về khóa học |
| Luồng sự kiện phụ | Người dùng chỉ có thể xem một số thông tin cơ bản của khóa học(nội dung tổng quan, mô tả, thông tin người hướng dẫn, xem đánh giá và bình luận, xem giá) nếu chưa mua khóa học |
| Điều kiện sau | Người dùng có thể xem thông tin chi tiết của khóa học |

#### Xem chi tiết bài giảng

|  |  |
| --- | --- |
| Mục đích | Người dùng có thể xem thông tin chi tiết của bài giảng trong khóa học |
| Mô tả | Cho phép người dùng xem các thông tin chi tiết về bài giảng |
| Tác nhân | Member |
| Điều kiện | * Người dùng đã mua khóa học * Truy cập vào bài giảng cụ thể trong chi tiết khóa học |
| Luồng sự kiện chính | * Xem video của bài giảng * Bình luận về bài giảng * Tải tài liệu của bài giảng * Làm bài tập với code và trắc nghiệm |
| Luồng sự kiện phụ | Người dùng không thể xem chi tiết bài giảng do chưa mua khóa học |
| Điều kiện sau | Người dùng có thể xem thông tin chi tiết của bài giảng |

#### Usecase thêm vào giỏ hàng

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm khóa học vào giỏ hàng |
| Mô tả | Cho phép người dùng thêm khóa học vào giỏ hàng |
| Tác nhân | Guest, Member |
| Điều kiện | Truy cập vào phần mềm kiểm tra và đánh giá |
| Luồng sự kiện chính | * Nhấn nút thêm vào giỏ hàng của khóa học cần thêm vào giỏ |
| Luồng sự kiện phụ | * Thêm giỏ hàng không thành công do khóa học đã có trong giỏ hàng |
| Điều kiện sau | Thêm khóa học vào giỏ hàng thành công |

#### Usecase thanh toán

|  |  |
| --- | --- |
| Mục đích | Người dùng thanh toán khóa học có trong giỏ hàng |
| Mô tả | Cho phép người dùng thanh toán trực tuyến khóa học trong giỏ hàng |
| Tác nhân | Member |
| Điều kiện | Chưa đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Nhập họ * Nhập tên * Nhập email * Nhập mật khẩu * Nhập mật khẩu xác nhận |
| Luồng sự kiện phụ | * Đăng ký không thành công vì chưa nhập đủ các trường bắt buộc * Đăng ký không thành công vì thông tin không đúng định dạng * Đăng ký không thành công vì email đã được sử dụng |
| Điều kiện sau | Đăng ký thành công và có thể đăng nhập vào hệ thống |

#### Usecase tạo khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm khóa học mới vào hệ thống |
| Mô tả | Cho phép người dùng thêm các khóa học mới vào hệ thống |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống |
| Luồng sự kiện chính | * Nhập tiêu đề * Chọn thể loại (frontend, backend, NLP, CV, Fullstack, Devops, …) * Thêm mô tả * Tạo nội dung khóa học(video, tài liệu, mô tả, câu hỏi trắc nghiệm,câu hỏi coding) * Hoàn thành các thông tin cơ bản như:ngôn ngữ, thể loại, tiêu đề,mô tả,ảnh,giá của khóa học và thông tin cá nhân cơ bản của instructor như tên ,ảnh đại diện,… * Review * Xuất bản khóa học |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Xuất bản khóa học thành công |

#### Usecase sửa khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng chỉnh sửa nội dung khóa học |
| Mô tả | Cho phép người dùng chỉnh sửa nội dung của khóa học |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang sửa khóa học |
| Luồng sự kiện chính | * Thay đổi nội dung cần chỉnh sửa(tiêu đề, thể loại,ngôn ngữ,giá,ảnh ,mô tả,…) * Thêm, sửa, xóa các bài giảng trong khóa học(video,tài liệu) * Thêm sửa xóa các câu hỏi trắc nghiệm của một đề trắc nghiệm : tên câu hỏi,đáp án ,… * Thêm sửa xóa câu hỏi coding : tên,mô tả, tên hàm,các tham số đâu vào,đầu ra và các giá trị của các testcase đối với hàm đó |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Những thay đổi của khóa học được lưu vào CSDL của hệ thống |

#### Usecase xóa khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng xóa khóa học ra khỏi hệ thống |
| Mô tả | Người dùng xóa khóa học ra khỏi hệ thống |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang danh sách khóa học |
| Luồng sự kiện chính | * Nhấn nút xóa khóa học * Hiển thị cảnh báo xác nhận xóa khóa học * Nhấn nút xác nhận |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Khóa học được xóa khỏi CSDL của hệ thống |

#### Usecase tạo bài giảng cho khóa học

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm bài giảng mới vào khóa học |
| Mô tả | Cho phép người dùng thêm bài giảng mới vào khóa học |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang tạo bài giảng |
| Luồng sự kiện chính | * Nhập tên bài giảng * Nhập mô tả * Thêm video bài giảng * Thêm tài liệu cần có |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Thêm mới bài giảng thành công |

#### Usecase sửa bài giảng

|  |  |
| --- | --- |
| Mục đích | Người dùng chỉnh sửa nội dung bài giảng |
| Mô tả | Cho phép người dùng chỉnh sửa nội dung của bài giảng |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang sửa bài giảng |
| Luồng sự kiện chính | * Thay đổi nội dung cần chỉnh sửa(tên, mô tả, video,tài liệu) |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Những thay đổi của bài giảng được lưu vào CSDL của hệ thống |

#### Usecase xóa bài giảng

|  |  |
| --- | --- |
| Mục đích | Người dùng xóa bài giảng ra khỏi khóa học |
| Mô tả | Người dùng xóa bài giảng ra khỏi khóa học |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học |
| Luồng sự kiện chính | * Nhấn nút xóa bài giảng * Hiển thị cảnh báo xác nhận xóa bài giảng * Nhấn nút xác nhận |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Bài giảng được xóa khỏi khóa học thành công |

#### Thêm kì thi trắc nghiệm

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm kì thi trắc nghiệm cho khóa học đang tạo |
| Mô tả | Nội dung khóa học có thể là:bài học,kì thi trắc nghiệm hoặc là kì thi coding. Việc tạo ra kì thi trắc nghiệm trong khóa học giúp người học tổng hợp được kiến thức của mình sau một vài bài học |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Truy cập vào khóa học muốn chỉnh sửa  Truy cập vào phần chương trình giảng dạy |
| Luồng sự kiện chính | * Nhấn nút thêm * Chọn thêm kì thi trắc nghiệm * Điền tên,mô tả về kì thi * Thêm câu hỏi trắc nghiêm * Thêm loại câu hỏi: 1 đáp án hoặc nhiều đáp án * Điền đáp án cho câu hỏi trắc nghiệm * Tích vào những đáp án đúng của câu hỏi đó * Tạo câu hỏi trắc nghiệm * Tiếp tục tạo câu hỏi mới với trình tự như trên |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Kì thi trắc nghiệm được tạo trong khóa học |

#### Thêm sửa xóa câu hỏi trong đề thi trắc nghiệm

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm sửa xóa câu hỏi trong đề thi trắc nghiệm |
| Mô tả | Một kì thi trắc nghiệm có nhiều câu hỏi .Một câu hỏi trắc nghiệm chứa các thông tin như:tên,danh sách đáp án,danh sách đáp án đúng .Người dùng có thể thêm câu hỏi,sửa một câu hỏi,hoặc xóa nó đi |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Truy cập vào khóa học muốn chỉnh sửa  Truy cập vào kì thi trắc nghiệm |
| Luồng sự kiện chính | * Thêm ,sửa ,xóa một câu hỏi trắc nghiệm |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Câu hỏi được thêm vào kì thi |

#### Thêm kì thi coding

|  |  |
| --- | --- |
| Mục đích | Thêm kì thi lập trình |
| Mô tả | Đối với những khóa học về lập trình ,bài tập có thể là những câu hỏi về lập trình để kiểm tra khả năng lập trình của học viện |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Chọn khóa học cần chỉnh sửa  Vào phần chương trình giảng dạy |
| Luồng sự kiện chính | * Nhấn nút thêm * Chọn câu hỏi coding * Nhập tên câu hỏi * Nhập mô tả về câu hỏi * Điền tên hàm,thời gian thực thi của hàm ,danh sách tham số đầu vào,kiểu dữ liệu đầu ra của hàm |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Kì thi coding đã được thêm vào khóa học |

#### Sửa kì thi coding

|  |  |
| --- | --- |
| Mục đích | Người dùng sửa kì thi coding |
| Mô tả | Người dùng có thể sửa các thông tin về kì thì coding như:tên kì thi,mô tả,tên hàm, danh sách tham số đầu vào,kiểu dữ liệu đầu ra,thời gian thực thi |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Truy cập vào khóa học muốn chỉnh sửa  Truy cập vào mục chương trình giảng dạy  Truy cập vào kì thi testcase muốn chỉnh sửa |
| Luồng sự kiện chính | * Chỉnh sửa thông tin như trên phân mô tả |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Dữ liệu sẽ được cập nhật trong cơ sở dữ liệu |

#### Xóa kì thi coding

|  |  |
| --- | --- |
| Mục đích | Người dùng xóa kỳ thi coding |
| Mô tả | Người dùng xóa kỳ thi coding ra khỏi nội dung chương trình giảng dạy của khóa học |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Truy cập vào khóa học muốn chỉnh sửa  Truy cập vào mục chương trình giảng dạy |
| Luồng sự kiện chính | * Đi đến kỳ thi coding muốn xóa * Nhấn vào icon xóa * Hiển thị cảnh báo xác nhận xóa bài giảng * Nhấn nút xác nhận |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Kỳ thi coding được xóa khỏi khóa học thành công |

#### Quản lý testcase của kì thi coding

|  |  |
| --- | --- |
| Mục đích | Người dùng thêm testcase vào trong kỳ thi coding |
| Mô tả | Testcase được thêm vào kỳ thi coding cho mục đích chấm tự động, code của học viên sẽ cho chạy qua những testcase này để kiểm tra kết quả đầu ra xem có khớp với nhau hay không |
| Tác nhân | Instructor |
| Điều kiện | Đăng nhập thành công vào hệ thống  Truy cập vào trang khóa học  Truy cập vào khóa học muốn chỉnh sửa  Truy cập vào mục chương trình giảng dạy  Truy cập đến kỳ thi coding muốn thêm testcase  Vào mục danh sách testcase |
| Luồng sự kiện chính | **Đối với thêm:**   * Nhấn nút thêm testcase * Ứng với mỗi tham số đầu vào và kiểu dữ liệu đầu ra ta điền thông tin phù hợp cho mỗi testcase * Nhấn nút thêm   **Đối với sửa:**   * Nhấn vào icon sửa với testcase muốn sửa * Sửa testcase   **Đối với xóa:**   * Nhấn vào icon xóa để xóa testcase muốn xóa ra khỏi kì thi |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Testcase đã được thao tác thành công |

#### Học viên xem chi tiết nội dung khóa học đã mua

|  |  |
| --- | --- |
| Mục đích | Học viên truy cập nội dung khóa học đã mua |
| Mô tả | Sau khi thanh toán thành công khóa học đăng kí học viên có thế xem nội dung chi tiết của khóa học như video,tài liệu của mỗi bài giảng,thảo luận thắc mắc của học viên sau mỗi bài giảng và có thể thi các bài thi trắc nghiệm hoặc coding |
| Tác nhân | Member |
| Điều kiện | Đăng nhập vào hệ thống  Được thêm vào khóa học sau khi thanh toán |
| Luồng sự kiện chính | * Vào khóa học đã thanh toán * Xem video, mô tả và các tài liệu liên quan của mỗi bài giảng * Sau mỗi bài giảng học viên có thể ý kiến đánh giá về chất lượng bài giảng, ý kiến thắc mắc về bài giảng * Theo lộ trình của tác giả khóa học mà học viên sẽ phải kiểm tra trình độ sau một vài buổi học nhất định:thi trắc nghiệm hoặc coding( đối với bài thi lập trình) |
| Luồng sự kiện phụ |  |
| Điều kiện sau |  |

#### Học viên thi kỳ thi trắc nghiệm

|  |  |
| --- | --- |
| Mục đích | Học viên kiểm tra trắc nghiệm trong khóa học |
| Mô tả | Trong mỗi khóa học hầu như sẽ có những bài thi trắc nghiệm để kiểm tra trình độ học viên hoặc tổng hợp kiến thức cho học viên sau một vài buổi học |
| Tác nhân | Member |
| Điều kiện | Đăng nhập vào hệ thống  Được thêm vào khóa học khi thanh toán |
| Luồng sự kiện chính | * Vào khóa học đã thanh toán * Nhấn vào bài thi trắc nghiệm trong lộ trình khóa học * Điền đáp án * Gửi đáp án * Xem kết quả |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Dữ liệu bài thi của học viên được lưu lai trong cơ sở dữ liệu hệ thống |

#### Học viên thi kỳ thi coding

|  |  |
| --- | --- |
| Mục đích | Học viên vào bài thi coding của khóa học |
| Mô tả | Đối với khóa học lập trình sẽ không thể thiếu những bài tập coding để kiểm tra tư duy lập trình của học viên |
| Tác nhân | Member |
| Điều kiện | Đăng nhập vào hệ thống  Được thêm vào khóa học khi thanh toán |
| Luồng sự kiện chính | * Vào khóa học đã thanh toán * Nhấn vào bài thi coding trong lộ trình khóa học * Điền code * Chạy code * Kết quả trả về |
| Luồng sự kiện phụ |  |
| Điều kiện sau | Kết quả bài thi được lưu trong cơ sở dữ liệu của hệ thống |

### ***Mô tả tác nhân***

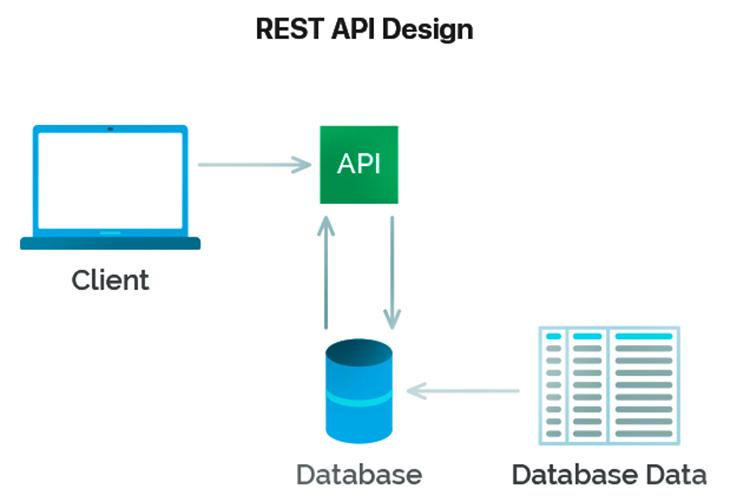
Bảng 2.1 Bảng mô tả tác nhân

|  |  |  |
| --- | --- | --- |
| STT | Actor | Ý nghĩa |
| 1 | Khách(Guest) | Người dùng hệ thống – là người có nhu cầu học các khóa học trực tuyến nhưng chưa đăng ký tài khoản của hệ thống nên chỉ có thể sử dụng những chức năng đơn giản của hệ thống |
| 2 | Thành viên(Member) | Người dùng hệ thống – là người có nhu cầu học các khóa học trực tuyến và đã đăng ký tài khoản thành viên của hệ thống, có thể sử dụng đầy đủ chức năng phía người dùng để học các khóa học có trên hệ thống |
| 3 | Giảng viên(Instructor) | Người dùng hệ thống – là người xây dựng và tạo các khóa học trên hệ thống |
| 4 | Quản trị(Admin) | Người dùng hệ thống – là người có quyền cao nhất, quản lý các số liệu thống kê của các khóa học để đưa ra những phương hướng phát triển mới cho hệ thống |

# THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG



## Kiến trúc chung hệ thống



Hình 3.1Mô hình kiến trúc hệ thống

***RESTful API?***

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

*Diễn giải các thành phần*

**API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

**REST** (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

*RESTful hoạt động như thế nào?*



Hình 3.2 Cách thức hoạt động Restful

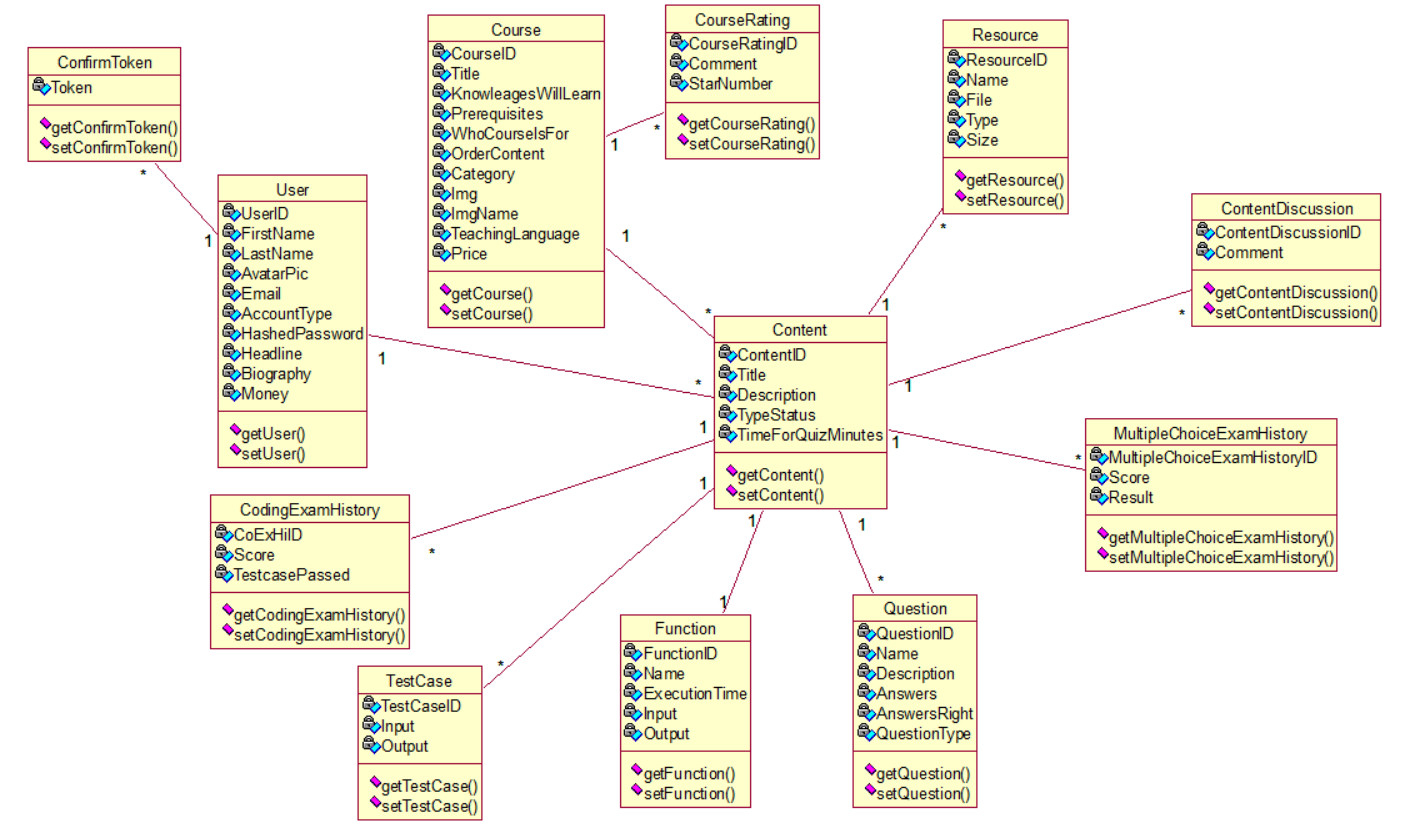
REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

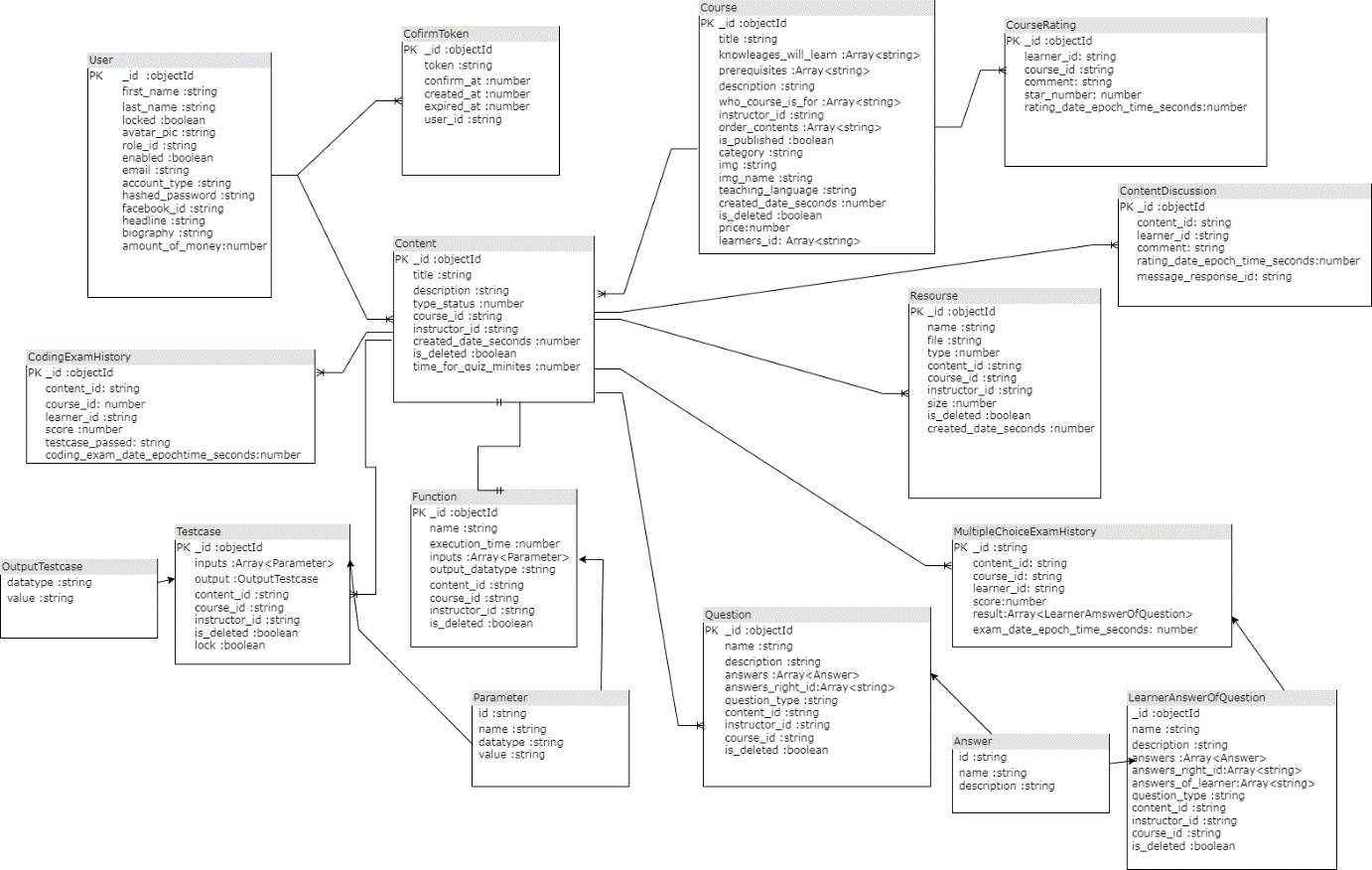
## Thiết kế các lớp

## Mối quan hệ các lớp



## Thiết kế cơ sở dữ liệu

* + 1. Sơ đồ tổng quát



Hình 3.3 Sơ đồ tổng quát

* + 1. Mô tả chi tiết
       1. User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin người dùng trong hệ thống | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã người dùng |  |
|  |  | First\_name | string | Tên người dùng | Dự |
|  |  | Last\_name | string | Họ | Nguyễn |
|  |  | locked | boolean | Tài khoản bị khóa hay không | false |
|  |  | Avatar-pic | string | Ảnh đại diện |  |
|  | X | Role\_id | string | Mã quyền |  |
|  |  | enabled | boolean | Tài khoản được kích hoạt chưa | true |
|  |  | email | String | email |  |
|  |  | Account\_type | String | Loại tài khoản:đăng nhập bằng facebook hay email | email |
|  |  | Hashed\_password | String | Mật khẩu được mã hóa |  |
|  | X | Facebook\_id | String | Mã facebook | 1312132123 |
|  |  | headline | String | Tiêu đề | Lập trình viên fullstack |
|  |  | biography | String | Tiểu sử | Có kinh nghiệm 10 năm trong lập trình web |
|  |  | Amount\_of\_money | number | Số tiền có trong tài khoản | 3000000 |

* + - 1. ConfirmToken

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng chứa token dùng để xác thực email | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | mã |  |
|  |  | token | string | Mã token | 121dsads121 |
|  |  | Confirm\_at | number | Thời điểm xác thực |  |
|  |  | Created\_at | number | Thời điểm tạo token |  |
|  |  | Expired\_at | number | Thời điểm hết hạn token |  |
|  | X | User\_id | string | Mã người dùng |  |

* + - 1. Content

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả nội dung chương trình đào tạo trong khóa học, nó có thể bài học,bài thi trắc nghiệm,bài thi coding.Nếu là bài thi sẽ có thêm một trường nữa là time\_for\_quiz\_minutes | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã bảng |  |
|  |  | title | string | Tiêu đề |  |
|  |  | Description | string | Mô tả |  |
|  |  | Type\_status | number | 0:bài học  1:bài tập trắc nghiệm  2:Bài tập coding |  |
|  | X | Course\_id | string | Mã khóa học |  |
|  | X | Instructor\_id | string | Mã tác giả |  |
|  |  | Created\_date\_seconds | number | Thời điểm tạo bản ghi |  |
|  |  | Is\_deleted | boolean | Bản ghi này đã xóa hay chưa |  |
|  |  | Time\_for\_quz\_minutes | number | Thời gian làm bài thi |  |

* + - 1. Parameter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Mô tả tham số trong 1 hàm, nó tượng trưng cho 1 kiểu dữ liệu | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | id | string | Mã tham số |  |
|  |  | name | string | Tên tham số | So\_thu\_nhat |
|  |  | datatype | string | Kiểu dữ liệu của tham số đó | integer |
|  |  | value | string | Giá trị của tham số | 12 |

* + - 1. Function

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin về hàm trong bài thi coding  Ví dụ:int add\_two\_number(int num1,int num2) | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | mã |  |
|  |  | name | string | Tên hàm | Add2number |
|  |  | Execution\_time | number | Thời gian thực thi | 12 |
|  |  | inputs | Array<Parameter> | Danh sách tham số đầu vào của hàm | [{id:1,name:num1,  Datatype:integer},  {id:2,name:num2,  Datatype:integer}] |
|  |  | Output\_datatype | string | Kiểu dữ liệu đầu ra | integer |
|  | X | Content\_id | string | Mã nội dung |  |
|  | X | Course\_id | string | Mã khóa học |  |
|  | X | Instructor\_id | string | Mã tác giả |  |
|  |  | Is\_deleted | boolean | Bản ghi này đã xóa hay chưa |  |

* + - 1. OutputTestcase

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin dữ liệu đầu ra của 1 testcase  Ví dụ: {datatype:integer,value:12} | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
|  |  | datatype | string | Kiểu dữ liệu | integer |
|  |  | value | string | Giá trị | 12 |

* + - 1. Testcase

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin testcase cho một hàm | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | mã |  |
|  |  | inputs | Array<Parameter> | Danh giá trị tham số đầu vào | [{datatype:integer,  name:num1.value:1},  {datatype:integer,  name:num2.value:2}] |
|  |  | output | OutpuTestcase | Giá trị đầu ra | {datatype:integer,  Value:3} |
|  | X | Content\_id | string | Mã nội dung |  |
|  | X | Course\_id | string | Mã khóa học |  |
|  | X | Instructor\_id | string | Mã tác giả |  |
|  |  | Is\_deleted | boolean | Bản ghi đã xóa hay chưa |  |
|  |  | lock | boolean | Tesecase hiển thị hay dấu |  |

* + - 1. CodingExamHistory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả lịch sử kì thi coding | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã |  |
|  | X | Content\_id | string | Mã nội dung |  |
|  | X | Course\_id | string | Mã khóa học |  |
|  | X | Leaner\_id | string | Mã học viên |  |
|  |  | score | string | Điểm | 5/10 |
|  |  | Testcase\_passed | string | Số testcase thành công | 6/12 |
|  |  | Coding\_exam\_date\_... | number | Ngày thi |  |

* + - 1. Course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin khóa học trong hệ thống | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã khóa học |  |
|  |  | title | string | Tên khóa học |  |
|  |  | knowleages\_will\_learn | Array<string> | Kết quả đạt được sau khi học |  |
|  |  | prerequisites | Array<string> | Điều kiện để tham gia khóa học |  |
|  |  | description | string | Mô tả về khóa học |  |
|  |  | who\_course\_is\_for | Array<string> | Khóa học dành cho ai |  |
|  | X | Instructor\_id | objectId | Mã người tạo khóa học |  |
|  |  | order\_content | Array<string> | Thứ tự nội dung bài học |  |
|  |  | is\_published | Boolean | Khóa học đã được công khai hay chưa |  |
|  |  | category | string | Thể loại của khóa học |  |
|  |  | img | string | Đường dẫn ảnh |  |
|  |  | img\_name | string | Tên ảnh |  |
|  |  | teaching\_language | string | Ngôn ngữ dùng để giảng dạy |  |
|  |  | created\_date\_seconds | number | Thời gian tạo dạng giây |  |
|  |  | Is\_deleted | boolean | Trạng thái đã xóa hoặc chưa |  |
|  |  | price | number | Giá của khóa học |  |
|  |  | learners\_id | Array<string> | Danh sách học viên |  |

* + - 1. CourseRating

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin đánh giá của học viên đối với khóa học | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã đánh giá |  |
|  | X | learner\_id | string | Mã người đánh giá |  |
|  | X | course\_id | string | Mã khóa học |  |
|  |  | comment | string | Bình luận của học viên về khóa học |  |
|  |  | start\_number | number | Số sao đánh giá |  |
|  |  | rating\_date\_epoch\_time\_seconds | number | Thời gian tạo đánh giá |  |

* + - 1. ContentDiscussion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin thảo luận của mỗi nội dung trong khóa học | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã thảo luận |  |
|  |  | content\_id | string | Mã nội dung |  |
|  | X | learner\_id | string | Mã học viên |  |
|  |  | comment | string | Bình luận của học viên về nội dung |  |
|  | X | message\_respone\_id | string | Mã phản hồi |  |
|  |  | rating\_date\_epoch\_time\_seconds | number | Thời gian tạo thảo luận |  |

* + - 1. Resourse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin tài liệu có trong hệ thống | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã tài liệu |  |
|  |  | name | string | Tên tài liệu |  |
|  |  | file | string | Đường dẫn của tài liệu |  |
|  |  | type | number | Loại file | 0 là video, 1 là file word,… |
|  | X | content\_id | string | Mã nội dung |  |
|  | X | course\_id | string | Mã khóa học |  |
|  | X | intructor\_id | string | Mã người tạo khóa học |  |
|  |  | size | number | Kích cỡ tài liệu |  |
|  |  | is\_deleted | boolean | Trạng thái xóa |  |
|  |  | created\_date\_seconds | number | Ngày tạo |  |

* + - 1. Answer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | mô tả kiểu dữ liệu câu trả lời | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
|  |  | id | objectId | Mã câu trả lời |  |
|  |  | Name | String | Tên câu trả lời |  |
|  |  | Description | String | Mô tả về đáp án vì sao đúng vì sao sai |  |

* + - 1. LeanerAnswerOfQuestion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Mô tả kiểu dữ liệu đáp án của học viên đối với câu hỏi trắc nghiệm | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
|  |  | id | objectId | Mã câu trả lời của học viên |  |
|  |  | Name | String | Tên câu trả lời |  |
|  |  | Description | String | Mô tả |  |
|  |  | answers | Array<Answer> | Danh sách câu trả lời |  |
|  |  | Answers\_right\_id | Array<string> | Mã câu trả lời đúng |  |
|  |  | Answer\_of\_learner | Array<string> | Câu trả lời của học viên |  |
|  |  | Question\_type | string | Loại câu hỏi |  |
|  |  | Content\_id | String | Mã nội dung |  |
|  |  | Instructor\_id | String | Mã người tạo |  |
|  |  | Course\_id | String | Mã khóa học |  |
|  |  | Is\_deleted | Boolean | Trạng thái xóa |  |

* + - 1. Question

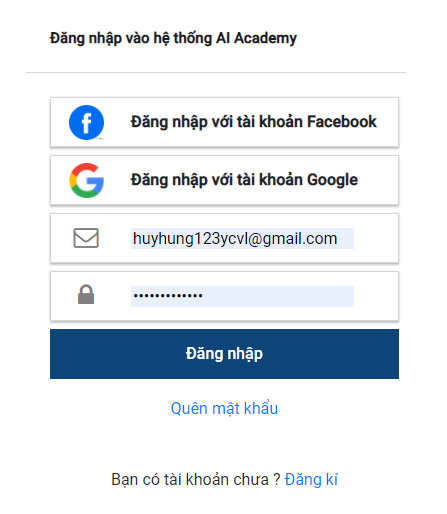
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả thông tin câu hỏi trong kỳ thi trắc nghiệm có trong khóa học | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã câu hỏi |  |
|  |  | Name | String | Tên câu trả lời |  |
|  |  | Description | String | Mô tả |  |
|  |  | answers | Array<Answer> | Danh sách câu trả lời |  |
|  |  | Answers\_right\_id | Array<string> | Mã câu trả lời đúng |  |
|  |  | Question\_type | string | Loại câu hỏi |  |
|  | X | Content\_id | String | Mã nội dung |  |
|  | X | Instructor\_id | String | Mã người tạo |  |
|  | X | Course\_id | String | Mã khóa học |  |
|  |  | Is\_deleted | Boolean | Trạng thái xóa |  |

* + - 1. MultipleChoiceExamHistory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô tả | Bảng mô tả lịch sử kết quả của học viên đối với bài thi trắc nghiệm | | | | |
| Khóa chính | Khóa ngoại | Tên trường | Kiểu dữ liệu | Mô tả | Ví dụ |
| X |  | \_id | objectId | Mã lịch sử |  |
|  | X | Content\_id | string | Mã nội dung |  |
|  | X | Course\_id | string | Mã khóa học |  |
|  | X | Learner\_id | String | Mã học viên |  |
|  |  | Score | number | Điểm |  |
|  |  | result | Array<LearnerAnswerOfQuestion> | Kết quả là một mảng câu trả lời của học viên |  |
|  |  | Exam\_date-epoch\_time-seconds | number | Thời gian thi |  |

## Thiết kế giao diện

### Giao diện đăng nhập

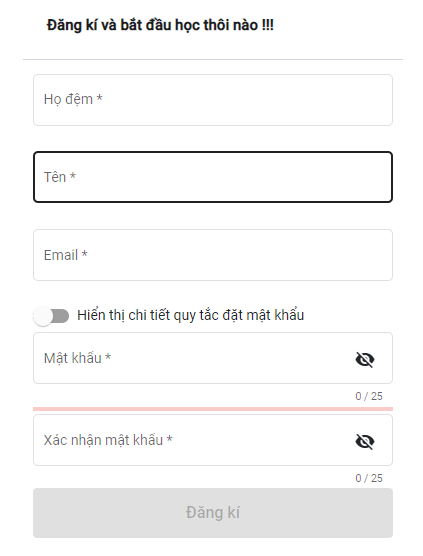


Hình 3.4 Giao diện đăng nhập

Chức năng đăng nhập

* Goals: Đăng nhập thành công vào hệ thống
* Methods: Người dùng truy cập vào link https://\*/login. Sau đó nhập [email], [Password], bấm nút “Login”, hoặc nhấn nút đăng nhập bằng Facebook. Hệ thống hiển thị giao diện trang chủ.

### Giao diện đăng ký thành viên

****

Hình 3.5 Giao diện đăng ký thành viên

Chức năng đăng ký

* Goals: Đăng ký thành viên thành công
* Methods: Người dùng truy cập vào link https://\*/sign-up hoặc thông qua “Đăng kí” ở màn hình [Login]. Nhập thông tin [email], [password], [Họ đệm], [Tên]. Chọn [Đăng kí] để đăng ký thành viên. Sau đó hệ thống sẽ gửi thông báo xác nhận về email của người dùng, người dùng cần vào email để xác nhận yêu cầu đăng kí.

### Giao diện trang chủ

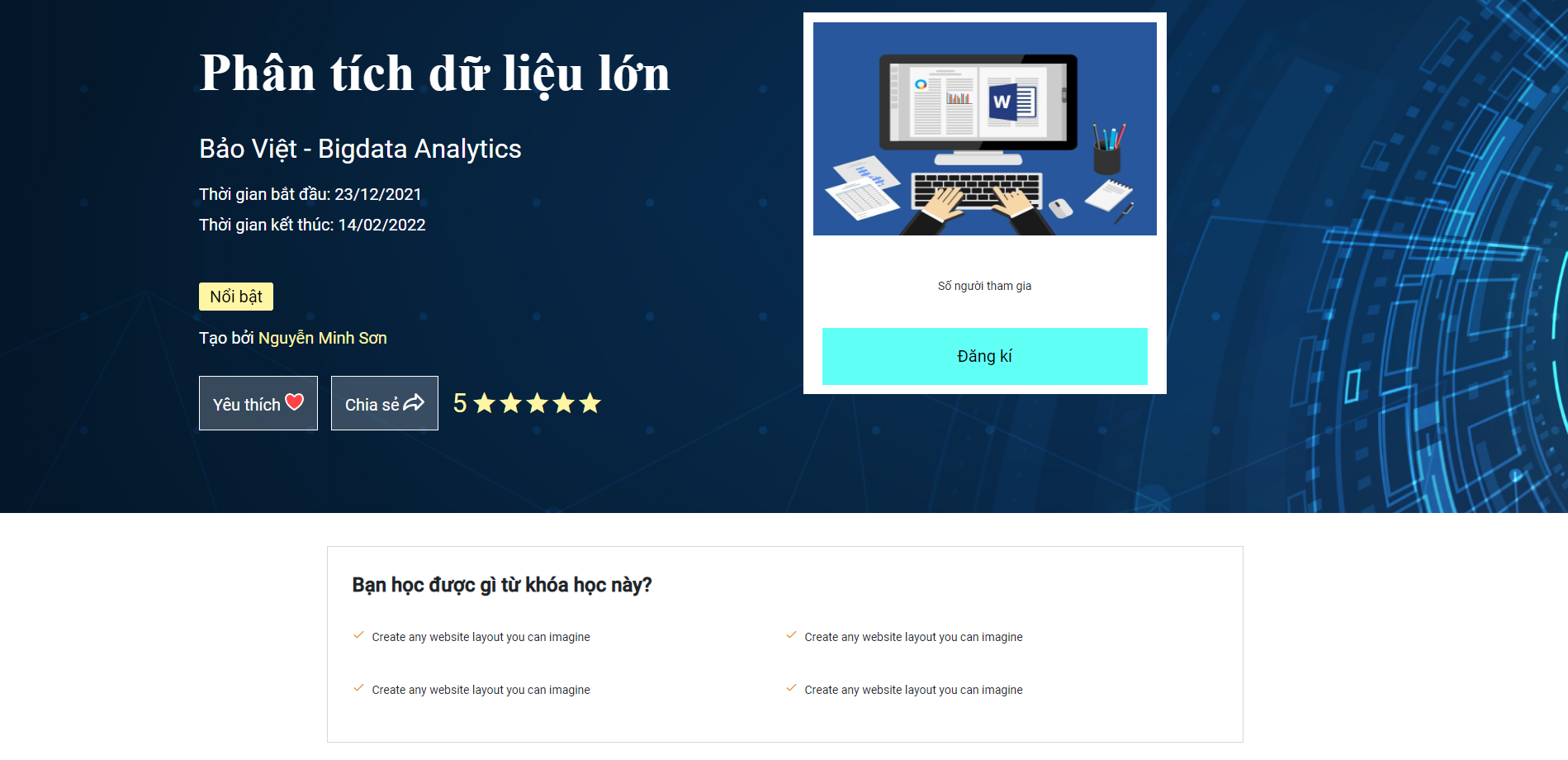


Hình 3.6 Giao diện trang chủ

Chức năng

* Goals: Xem danh sách khóa học
* Methods: Truy cập vào trang web

### Giao diện xem chi tiết khóa học

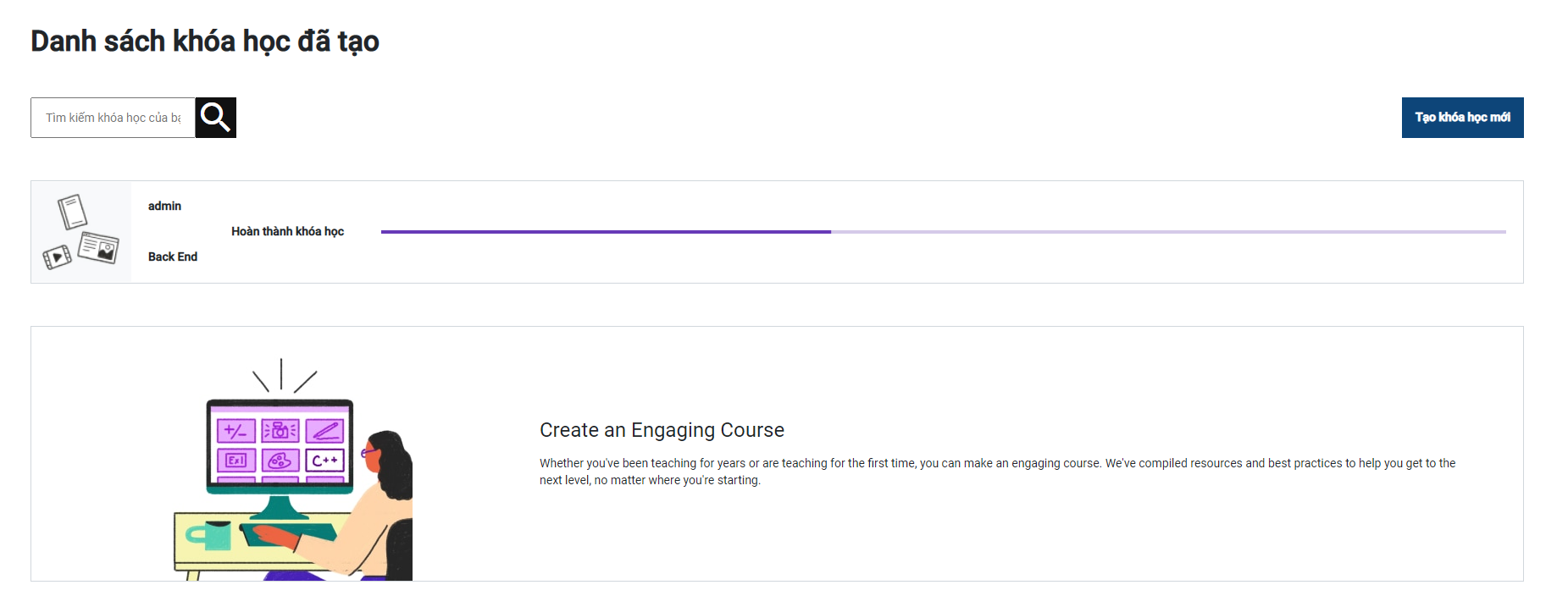


Hình 3.7 Giao diện xem chi tiết khóa học

Chức năng

* Goals: Xem chi tiết khóa học
* Methods: chọn một khóa học trong các khóa học ở trang chủ

### Giao diện danh sách khóa học đã tạo

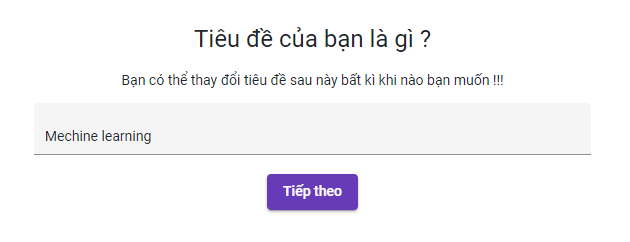
****

Hình 3.8 Giao diện danh sách khóa học đã tạo

Chức năng

* Goals: Xem danh sách khóa học đã tạo
* Methods: chọn Tạo khóa học ở trang chủ

### Giao diện tạo tên khóa học

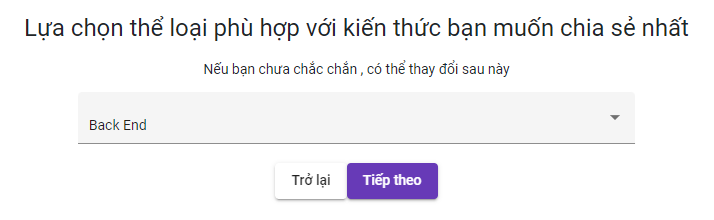


Hình 3.9 Giao diện tạo tên khóa học

Chức năng

* Goals: Lưu thông tin về tên khóa học.
* Methods: chọn tạo khóa học mới ở trang danh sách khóa học đã tạo.

### Giao diện lựa chọn thể loại khóa học

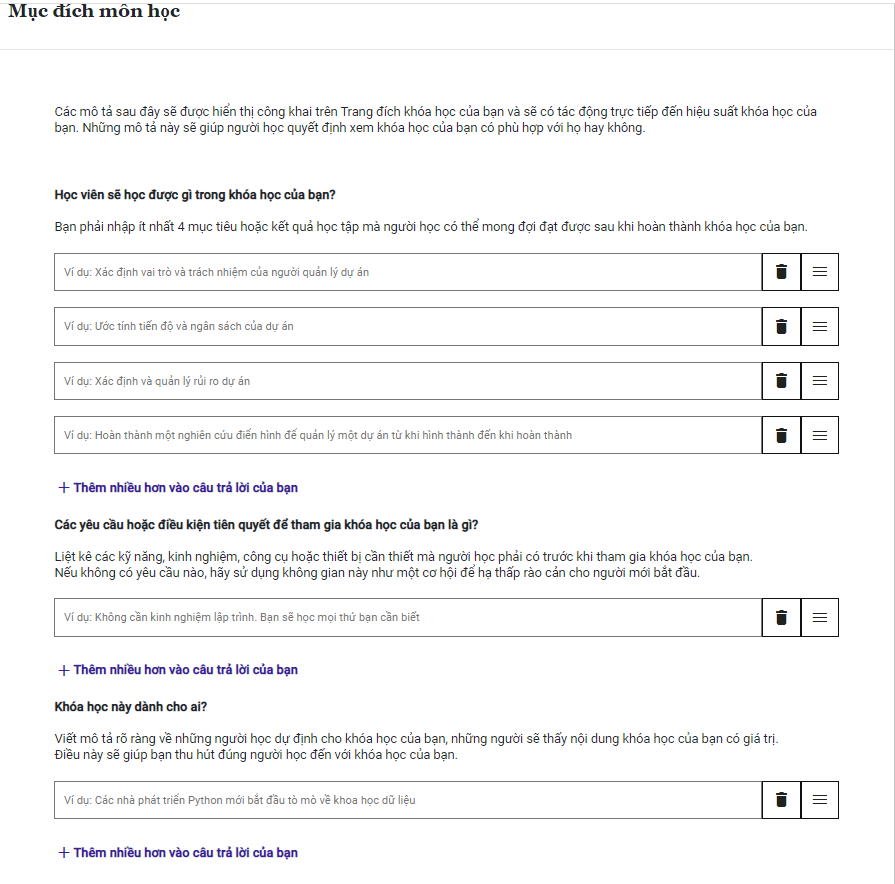


Hình 3.10 Giao diện lựa chọn thể loại khóa học

Chức năng

* Goals: Lưu thông tin về thể loại khóa học.
* Methods: Chọn Tiếp theo ở giao diện tạo tên khóa học.

### Giao diện tạo mục đích của môn học

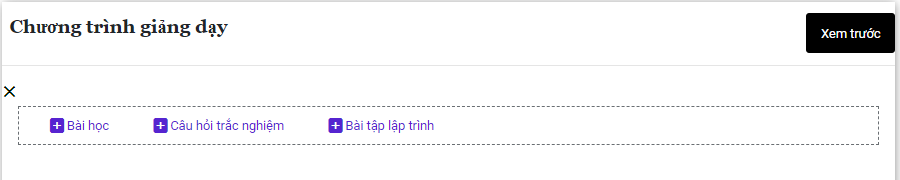


Hình 3.11 Giao diện tạo mục đích của môn học

Chức năng

* Goals: Lưu thông tin về mục đích môn học.
* Method: Nhập các thông tin cần thiết về mục đích môn học.

### Giao diện tạo nội dung khóa học

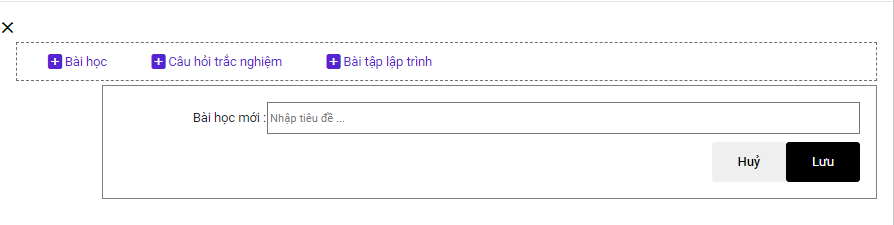
****

Hình 3.12 Giao diện

Chức năng thông tin

* Goals: Chọn thể loại nội dung người dùng muốn tạo cho khóa học.
* Methods: Chọn vào icon “+” trên màn hình.

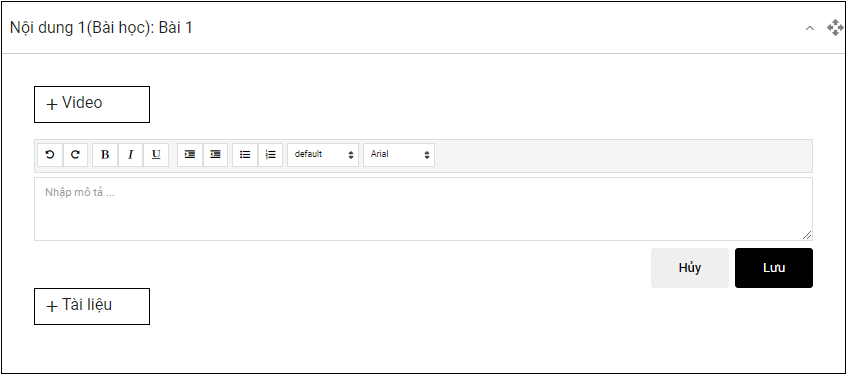
### Giao diện tạo bài học



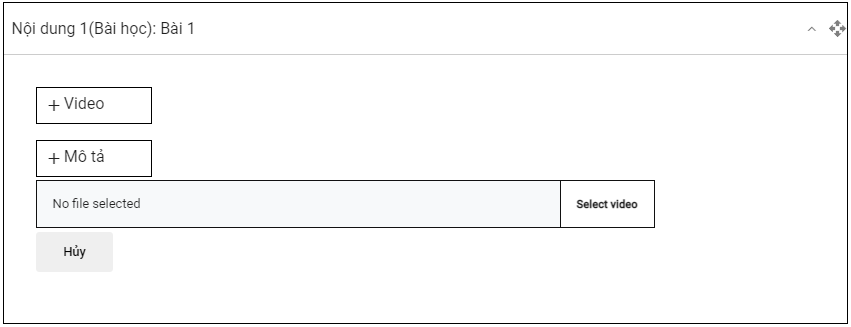
Hình 3.13 Giao diện nhập tiêu đề bài học



Hình 3.14 Giao diện thêm video cho bài học



Hình 3.15Giao diện tạo mô tả cho bài học



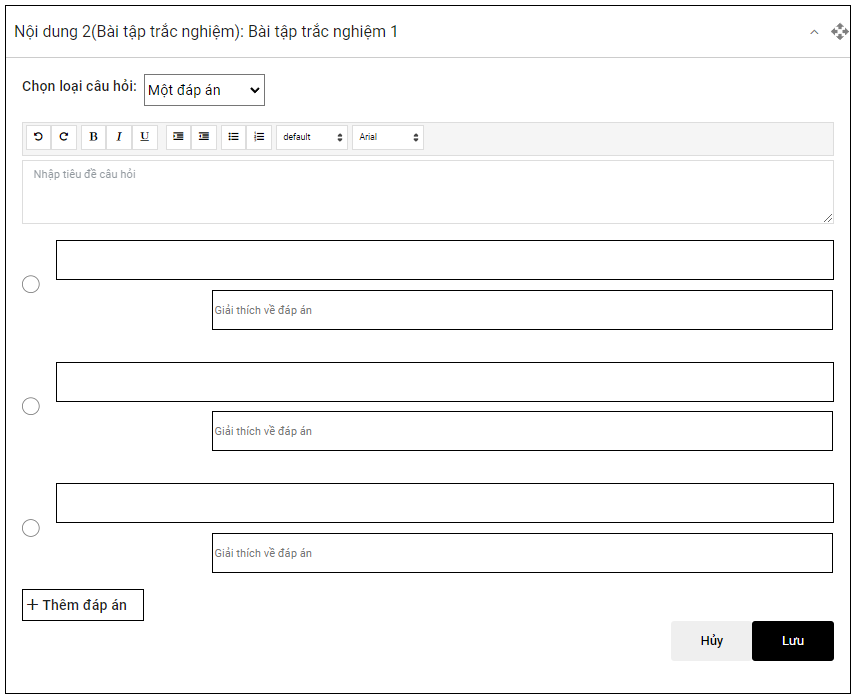
Hình 3.16 Giao diện thêm file tài liệu cho bài học

* Goals: Tạo thành công bài học
* Methods: Chọn Bài học và nhập các thông tin cần thiết

### Giao diện tạo bài tập trắc nghiệm cho khóa học



Hình 3.17Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài



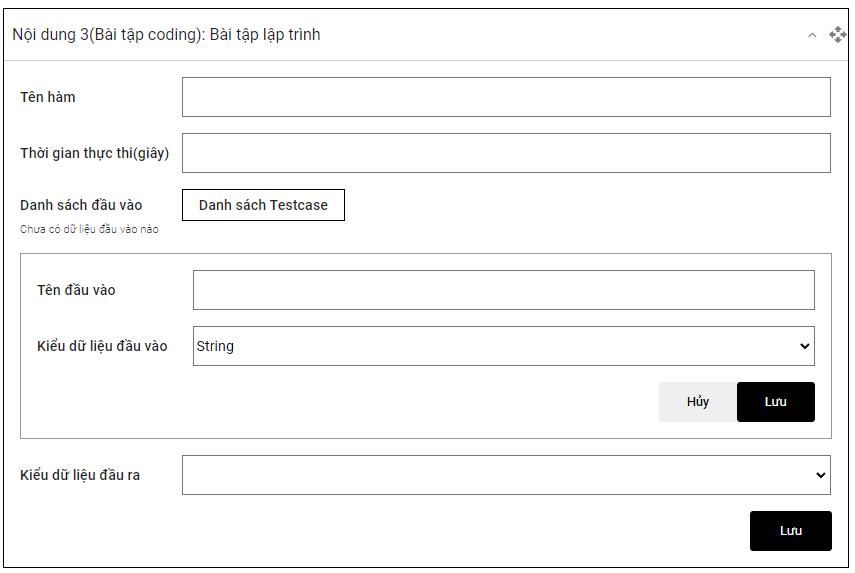
Hình 3.18Giao diện tạo câu hỏi cho bài tập trách nghiệm

* Goals: Tạo thành công bài tập trắc nghiệm
* Methods: chọn Bài tập trắc nghiệm và nhập các thông tin cần thiết.

### Giao diện tạo bài tập lập trình



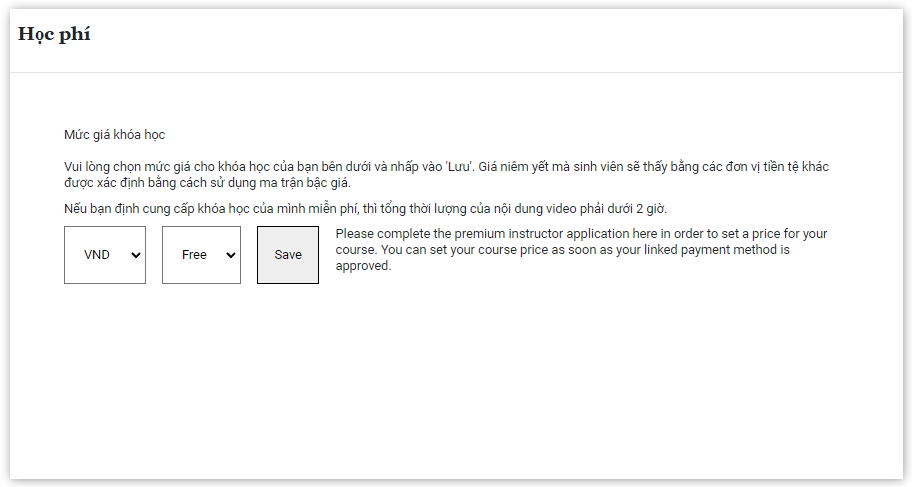
Hình 3.19Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài



Hình 3.20Giao diện nhập nội dung cho bài tập trắc nghiệm

* Goals: Tạo thành công bài tập lập trình
* Methods: Chọn bài tập lập trình và nhập các thông tin cần thiết

### Giao diện tạo học phí khóa học



Hình 3.21 Giao diện tạo học phí cho khóa học

* Goals: tạo học phí cho khóa học
* Methods: Chọn học phí bên trang menu.

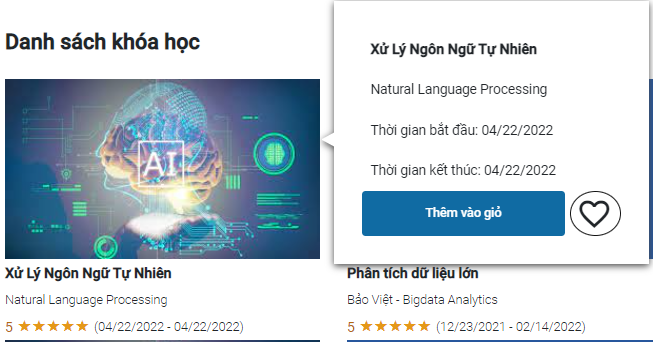
### Giao diện tạo trang đích của khóa học



Hình 3.22 Giao diện tạo trang đích khóa học

* Goals: Thông tin trang đích khóa học
* Methods: Chọn Trang đích khóa học ở thanh menu

### Giao diện thêm vào giỏ hàng

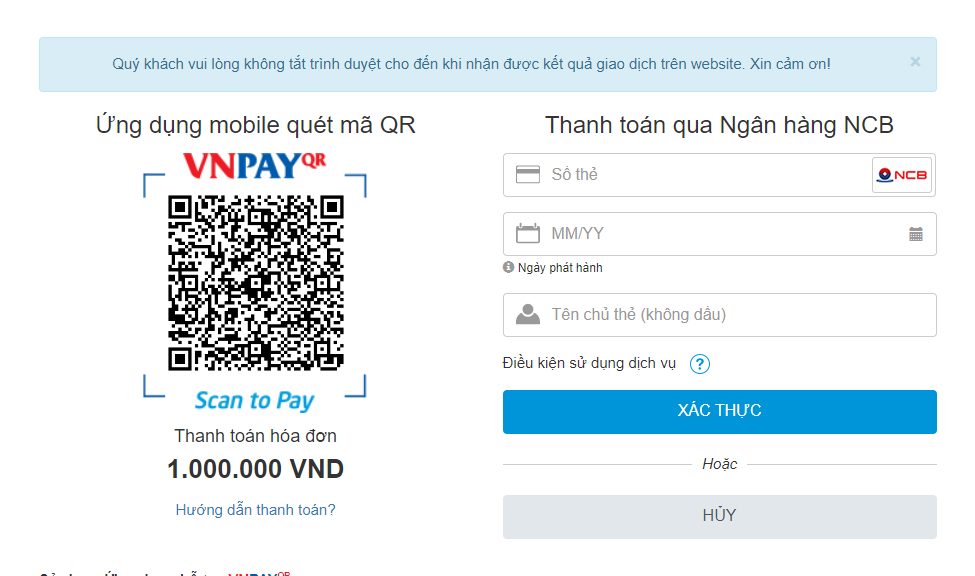


Hình 3.23 Giao diện thêm vào giỏ hàng

### Giao diện thanh toán



Hình 3.24 Giao diện giỏ hàng



Hình 3.25Giao diện cổng thanh toán

# CHƯƠNG 4

**CÀI ĐẶT HỆ THỐNG**

## Công nghệ sử dụng

### 4.1.1. FastAPI

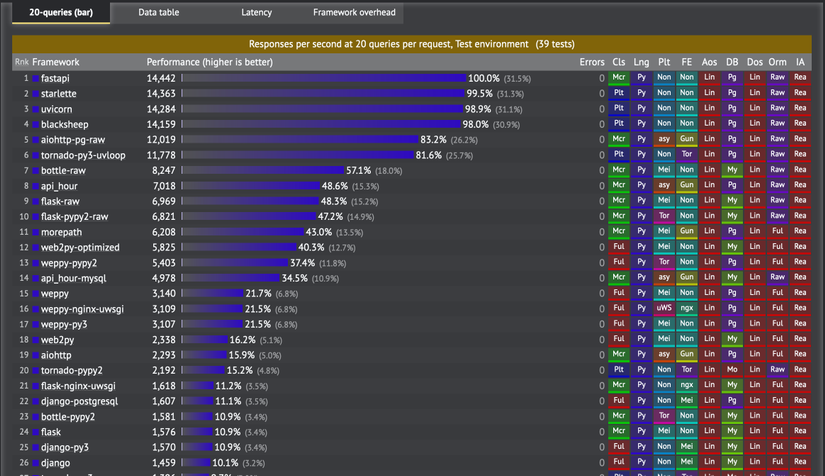
#### 4.1.1.1 Khái niệm

FastAPI là nền tảng thiết kế, lập trình xây dựng API cực kỳ nhanh trên cả 2 phương diện phát triển và thực thi trên Python 3.6+.Từ Python 3.6+ thì bạn đã có thể sử dụng cú pháp await/async để chạy code bất đồng bộ, vì lý do này các framework trên Python sẽ đạt được hiệu năng cao, FastAPI là một trong số Python framework nhanh nhất hiện nay.

#### Lý do chọn FastAPI

FastAPI là một micro framework khá mới, chỉ vừa được release năm 2018. Github của framework này hiện tính đến tháng 1/2021 đang đạt 25,4k star, tuy nhiên do được áp dụng khá nhiều công nghệ mới nên có FastAPI có vài điểm mạnh mà mình cảm thấy khá phù hợp để sử dụng phát triển project:

High performance Do được base trên 2 lib khá mạnh ở thời điểm hiện tại của python là Pydantic và Starlette nên FastAPI sở hữu hiệu suất cao nhất trong tất cả các framework Python hiện nay, bạn có thể tham khảo so sánh hiệu năng giữa các framework tại [https://www.techempower.com/](https://www.techempower.com/benchmarks/#section=test&runid=7464e520-0dc2-473d-bd34-dbdfd7e85911&hw=ph&test=query&l=zijzen-7).



Hình 4.1 Bảng so sánh hiệu năng giữa các framwork Python

Development Speed Được hỗ trợ tích hợp sẵn giao diện Swagger – OpenAPI kèm theo cách code khá đơn giản nên lập trình có thể release function rất nhanh mà vẫn có document đầy đủ, đây là lợi thế có thể nói là quan trọng nhất của FastAPI so với các Framework khác. Dưới đây là một đoạn code in ra dòng text healthcheck.

Bất đồng bộ Hiện tại bất đồng bộ đã được hỗ trợ từ phiên bản Django 3.x nhưng ngay từ khi release, FastAPI mặc định đã hỗ trợ developer Async, cũng vì vậy mà FastAPI chỉ có thể sử dụng với python3.6 trở lên.

So sánh 3 Framework Python là Django, Flask and FastAPI

Dưới đây là bảng so sánh giữa 3 Framework Django, Flask và FastAPI:

|  | **Django** | **Flask** | **FastAPI** |
| --- | --- | --- | --- |
| **Community** | Cộng đồng của Django hiện tại khá lớn và lâu đời, vì vậy số lượng lib hỗ trợ cho Django có thể nói là nhiều nhất ở thời điểm hiện tại | Cộng đồng của Flask hiện tại cũng lớn nhưng không bằng Django, bởi vậy số lượng lib cũng hạn chế hơn Django | Cộng đồng của FastAPI là khá mới, do vậy số lượng lib hỗ trợ khá ít |
| **Python version** | All version: 2x, 3x | All version: 2x, 3x | Only python3.x |
| **Performance** | Do hỗ trợ nhiều feature nên perfomance của Django không được cao | Performance khá cao | Là framework có hiệu năng cao nhất trong các framework python hiện tại |
| **ORM** | Django ORM | Flask-sqlachemy | Sqlachemy |
| **Async** | Hỗ trợ từ Django 3.x | Chưa hỗ trợ | Tương thích hoàn toàn |
| **Builtin Admin UI** | Có | Không | Không |
| **Project Frame** | Có, chỉ cần run python manage startapp yourapp | Không | Không |
| **Python shell** | Có | Có | Không – đây là hạn chế của một microframework so với các framework thực thụ |
| **OpenAPI document** | Phổ biến nhất là sử dụng Django-rest-framework, với Swagger thì dùng django-rest-swagger với python2.x, dùng drf-yasg với python3.x | Sử dụng Flask-restplus | Mặc định đã tích hợp sẵn Swagger và Redoc |

### MongoDB

#### NoSql - cơ sở dữ liệu phi quan hệ:

NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.

NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...

Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...

NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.

NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...

#### MongoDB là gì?

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng.

So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

#### Một số câu lệnh cơ bản trên MongoDB

| CSDL | **MySQL** | **MongoDB** |
| --- | --- | --- |
| **Tạo csdl** | CREATE DATABASE test; | use test; |
| **Tạo bảng** | CREATE TABLE students (ten\_cot - kieu\_du\_lieu); | db.createCollection('students'); |
| **Tạo bản ghi** | INSERT INTO studetns ('name', 'gender') VALUES('thanh', 'male'); | db.students.insert({ name:'thanh', gender: 'male'}); |
| **Cập nhật** | UPDATE students SET name = 'thanh update' WHERE id = 1; | db.students.update({ \_id: 1 },{$set:{ name: 'thanh update' }}); |
| **Xóa bản ghi** | DELETE FROM students Where id = 1; | db.students.remove({ \_id: 1}); |
| **Tìm kiếm all** | SELECT \* FROM students; | db.students.find({}); |
| **Tìm kiếm** | SELECT \* FROM students WHERE name = 'thanh'; | db.students.find({ name: 'thanh' }); |

#### Ưu điểm của mongoDB.

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.

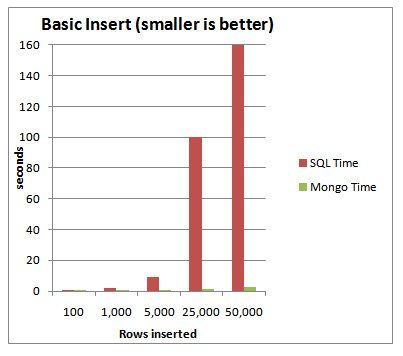
Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.



Hình 4.2 Bảng so sánh tốc độ truy vấn SQL và Mongo

#### Nhược điểm của mongoDB

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

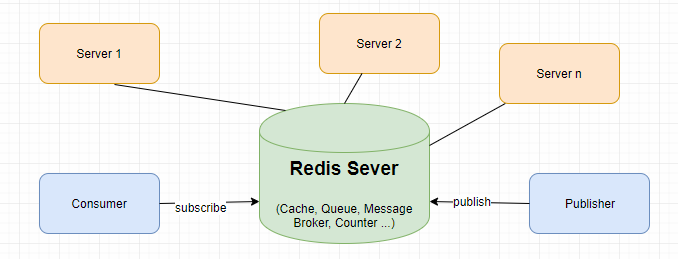
#### Khi nào nên dùng MongoDB

Ví dụ như các hệ thống realtime (thời gian thực) yêu cầu phản hồi nhanh, Các hệ thống bigdata với yêu cầu truy vấn nhanh hay các hệ thống có lượng request lớn thì MongoDB sẽ là sự lựa chọn ưu tiên hơn CSDL quan hệ. Tùy theo dự án và trường hợp cụ thể để sử dụng CSDL quan hệ hay sử dụng MongoDB đem lại hiệu quả cao.

### Redis

#### Redis là gì?

**Redis (REmote DIctionary Server)** là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker.



Hình 4.3 Sơ đồ Redis server

#### **Các ứng dụng của Redis**

Ngoài tính năng lưu trữ KEY-VALUE trên RAM thì Redis còn hỗ trợ tính năng xắp xếp, query, backup dữ liệu trên đĩa cứng cho phép bạn có thể phục hồi dữ liệu khi hệ thống gặp sự cố…và có thể nhân bản (Chạy nhiều Server Redis cùng lúc).

**Caching:**Sử dụng làm bộ nhớ đệm. Chính tốc độ đọc ghi nhanh mà Redis có thể làm bộ nhớ đệm, nơi chia sẻ dữ liệu giữa các ứng dụng hoặc làm database tạm thời. Ngoài ra Redis có thể sử dụng để làm Full Page Cache cho website. Cũng vì tính nhất quán của Redis, cho dù restart Redis thì người dùng cũng không có cảm nhận chậm khi tải trang.

**Counter:** Sử dụng làm bộ đếm. Với thuộc tính tăng giảm thông số rất nhanh trong khi dữ liệu được lưu trên RAM, sets và sorted sets được sử dụng thực hiện đếm lượt view của một website, các bảng xếp hạng trong game chẳng hạng. Redis hỗ trợ thread safe do đó nó có thể đồng bộ dữ liệu giữa các request.

**Publish/Suscribe (Pub/Sub):** Tạo kênh chia sẻ dữ liệu. Redis hỗ trợ tạo các channel để trao đổi dữ liệu giữa publisher và subscriber giống như channel trong **Socket Cluster** hay topic trong **Apache Kafka. Ví dụ:**Pub/Sub được sử dụng theo dõi các kết nối trong mạng xã hội hoặc các hệ thống chat.

**Queues:** Tạo hàng đợi để xử lý lần lượt các request. Redis cho phép lưu trữ theo list và cung cấp rất nhiều thao tác với các phần tử trong list, vì vậy nó còn được sử dụng như một message queue.

#### **Các kiểu dữ liệu trong Redis**

Khác với RDMS như MySQL, hay PostgreSQL, Redis không có table (bảng). Redis lưu trữ data dưới dạng key-value. Thực tế thì memcache cũng làm vậy, nhưng kiểu dữ liệu của memcache bị hạn chế, không đa dạng được như Redis, do đó không hỗ trợ được nhiều thao tác từ phía người dùng. Dưới đây là sơ lược về các kiểu dữ liệu Redis dùng để lưu value.

**– STRING**: string, integer hoặc float. Redis có thể làm việc với cả string, từng phần của string, cũng như tăng/giảm giá trị của integer, float.

**– LIST**: List là một danh sách của strings, sắp xếp theo thứ tự insert. Redis có thể thêm một phần tử vào đầu hoặc cuối list. List phù hợp cho các bài toán cần thao tác với các phần tử gần đầu và cuối vì việc truy xuất này là cực nhanh, cho dù insert cả triệu phần tử. Tuy nhiên nhược điểm là việc truy cập vào các phần tử ở giữa list rất chậm.

**– SET**: tập hợp các string (không được sắp xếp). Redis hỗ trợ các thao tác thêm, đọc, xóa từng phần tử, kiểm tra sự xuất hiện của phần tử trong tập hợp. Ngoài ra Redis còn hỗ trợ các phép toán tập hợp, gồm intersect/union/difference.

**– HASH**: lưu trữ hash table của các cặp key-value, trong đó key được sắp xếp ngẫu nhiên, không theo thứ tự nào cả. Redis hỗ trợ các thao tác thêm, đọc, xóa từng phần tử, cũng như đọc tất cả giá trị.

**– SORTED SET (ZSET)**: là 1 danh sách, trong đó mỗi phần tử là map của 1 string (member) và 1 floating-point number (score), danh sách được sắp xếp theo score này. Các phần tử của zset được sắp xếp theo thứ tự từ score nhỏ tới lớn.

Ngoài ra, Redis còn hỗ trợ các data types khác như: Bit arrays, HyperLogLogs, Streams.

#### **Persistent redis là gì**

Bên cạnh việc lưu key-value trên bộ nhớ RAM, Redis có 2 background threads chuyên làm nhiệm vụ định kỳ ghi dữ liệu lên đĩa cứng.

Có 2 loại file được ghi xuống đĩa cứng:

* + **RDB (Redis DataBase file)**

RDB thực hiện tạo và sao lưu snapshot của DB vào ổ cứng sau mỗi khoảng thời gian nhất định.

* + - **Ưu điểm**

RDB cho phép người dùng lưu các version khác nhau của DB, rất thuận tiện khi có sự cố xảy ra.

Bằng việc lưu trữ data vào 1 file cố định, người dùng có thể dễ dàng chuyển data đến các data centers, máy chủ khác nhau.

Khi restart server, dùng RDB làm việc với lượng data lớn sẽ có tốc độ cao hơn là dùng AOF.

* + - **Nhược điểm**

RDB không phải là lựa chọn tốt nếu muốn giảm thiểu tối đa nguy cơ mất mát dữ liệu.

Thông thường người dùng sẽ set up để tạo RDB snapshot 5 phút 1 lần (hoặc nhiều hơn). Do vậy, trong trường hợp có sự cố, Redis không thể hoạt động, dữ liệu trong những phút cuối sẽ bị mất.

RDB cần dùng fork() để tạo tiến trình con phục vụ cho thao tác disk I/O. Trong trường hợp dữ liệu quá lớn, quá trình fork() có thể tốn thời gian và server sẽ không thể đáp ứng được request từ client trong vài milisecond hoặc thậm chí là 1 second tùy thuộc vào lượng data và hiệu năng CPU.

* + **AOF (Append Only File)**

AOF lưu lại tất cả các thao tác write mà server nhận được, các thao tác này sẽ được chạy lại khi restart server hoặc tái thiết lập dataset ban đầu.

* + - **Ưu điểm**

Sử dụng AOF sẽ giúp đảm bảo dataset được bền vững hơn so với dùng RDB. Người dùng có thể config để Redis ghi log theo từng câu query hoặc mỗi giây 1 lần.

Redis ghi log AOF theo kiểu thêm vào cuối file sẵn có, do đó tiến trình seek trên file có sẵn là không cần thiết. Ngoài ra, kể cả khi chỉ 1 nửa câu lệnh được ghi trong file log (có thể do ổ đĩa bị full), Redis vẫn có cơ chế quản lý và sửa chữa lối đó (redis-check-aof).

Redis cung cấp tiến trình chạy nền, cho phép ghi lại file AOF khi dung lượng file quá lớn.

* + - **Nhược điểm**

File AOF thường lớn hơn file RDB với cùng 1 dataset.

AOF có thể chậm hơn RDB tùy theo cách thức thiết lập khoảng thời gian cho việc sao lưu vào ổ cứng. Tuy nhiên, nếu thiết lập log 1 giây 1 lần có thể đạt hiệu năng tương đương với RDB.

Developer của Redis đã từng gặp phải bug với AOF (mặc dù là rất hiếm), đó là lỗi AOF không thể tái tạo lại chính xác dataset khi restart Redis. Lỗi này chưa gặp phải khi làm việc với RDB bao giờ.

### Websocket

#### Socket là gì?

Socket là gì? Socket là một điểm cuối (end-point) của liên kết giao tiếp hai chiều (two-way communication) giữa hai chương trình chạy trên mạng. Nghĩa là một socket được sử dụng để cho phép 1 process nói chuyện với 1 process khác.

Các lớp Socket được sử dụng để tiến hành kết nối giữa client và server. Nó được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.

***Socket sử dụng như thế nào?***

Sau khái niệm socket là gì? Thì mình sẽ giới thiệu đến chức năng của socket, nó được áp dụng trong quá trình làm việc, nếu có thể chạy nhiều socket cùng một lúc nên công việc của bạn sẽ nhanh hơn, nâng cao hiệu suất làm việc.

***Socket hoạt động như thế nào?***

Socket giúp lập trình viên kết nối các ứng dụng để truyền và nhận dữ liệu trong môi trường có kết nối Internet bằng cách sử dụng phương thức TCPIP và UDP.

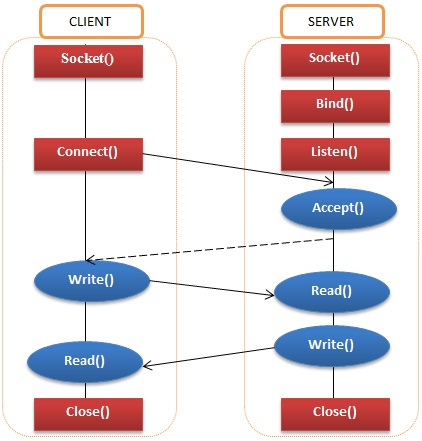
Khi cần trao đổi dữ liệu cho nhau thì 2 ứng dụng cần phải biết thông tin IP và port bao nhiêu của ứng dụng kia.

Có rất nhiều dạng socket khác nhau phụ thuộc vào sự khác biệt giữa cách truyền dữ liệu (protocol). Dạng phổ biến nhất là TCP và UDP.

***Phân loại Socket***

*Stream Socket là gì?*

Dựa trên giao thức TCP( Tranmission Control Protocol), stream socket thiết lập giao tiếp 2 chiều theo mô hình client và server. Được gọi là socket hướng kết nối.



Hình 4.4 Giao thức TCP

Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng tuần tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.

**Đặc điểm tóm gọn:**

Có một đường kết nối (địa chỉ IP) giữa 2 tiến trình.

Một trong hai tiến trình kia phải đợi tiến trình này yêu cầu kết nối.

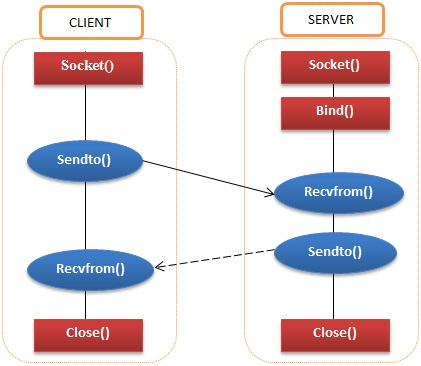
Mô hình client /sever thì sever lắng nghe và chấp nhận từ client.

Mỗi thông điệp gửi phải có xác nhận trả về.

Các gói tin chuyển đi tuần tự.

*Datagram Socket*

Dựa trên giao thức UDP( User Datagram Protocol) việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 process. Tức là nó cung cấp connection-less point cho việc gửi và nhận packets. Gọi là socket không hướng kết nối



Hình 4.5 Giao thức UDP

Do không yêu cầu thiết lập kết nối, không phải có những cơ chế phức tạp. Nên tốc độ giao thức khá nhanh, thuận tiện cho các ứng dụng truyền dữ liệu nhanh như chat, game online…

**Đặc điểm tóm gọn:**

Hai tiến trình liên lạc với nhau không kết nối trực tiếp

Thông điệp gửi đi phải kèm theo thông điệp người nhận

Thông điệp có thể gửi nhiều lần

Người gửi không chắc chắn thông điệp đến tay người nhận.

Thông điệp gửi sau có thể đến trước và ngược lại.

Để có thể thực hiện các cuộc giao tiếp, một trong 2 quá trình phải công bố port của socket mà mình đang sử dụng.

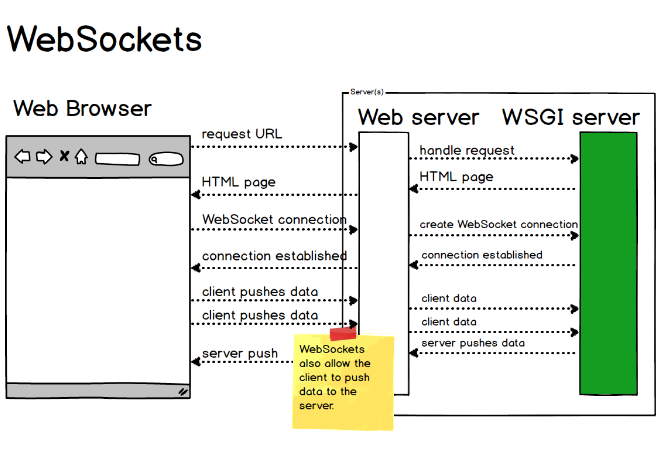
#### Web socket là gì?

**Websocket** là giao thức hỗ trợ giao tiếp hai chiều giữa client và server để tạo một kết nối trao đổi dữ liệu. Giao thức này không sử dụng HTTP mà thực hiện nó qua TCP. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

*Ưu điểm*

WebSocket cung cấp giao thức giao tiếp hai chiều mạnh mẽ. No có độ trễ thấp và dễ xử lý lỗi. Websocket thường được sử dụng cho những trường hợp yêu cầu real time như chat, hiển thị biểu đồ hay thông tin chứng khoán.

*Cấu trúc WebSocket*



Hình 4.6 Cấu trúc Websocket

Giao thức chuẩn thông thường của WebSocket là ws:// , giao thức secure là wss:// . Chuẩn giao tiếp là String và hỗ trợ buffered arrays và blobs.

***Các thuộc tính của WebSocket***

|  |  |
| --- | --- |
| **THUỘC TÍNH** | **MÔ TẢ** |
| readyState | Diễn tả trạng thái kết nối. Nó có các giá trị sau:  Giá trị 0: kết nối vẫn chưa được thiết lập (WebSocket.CONNECTING)  Giá trị 1: kết nối đã thiết lập và có thể giao tiếp (WebSocket.OPEN)  Giá trị 2: kết nối đang qua handshake đóng (WebSocket.CLOSING)  Giá trị 3: kết nối đã được đóng (WebSocket.CLOSED) |
| bufferedAmount | Biểu diễn số byte của UTF-8 mà đã được xếp hàng bởi sử dụng phương thức send() |

***Các sự kiện WebSocket***

|  |  |  |
| --- | --- | --- |
| **SỰ KIỆN** | **EVENT HANDLER** | **MÔ TẢ** |
| open | onopen | Khi một WebSocket chuyển sang trạng thái mở, “onopen” sẽ được gọi. |
| message | onmessage | Khi WebSocket nhận dữ liệu từ Server. |
| error | onerror | Có bất kỳ lỗi nào trong giao tiếp. |
| close | onclose | Kết nối được đóng. Những sự kiện được truyền cho “onclose” có ba tham số là “code”, “reason”, và “wasClean”. |

Event handlers có thể được tạo ra bằng cách sử dụng phương thức **addEventListener()**.

***Các phương thức của WebSocket***

|  |  |
| --- | --- |
| PHƯƠNG THỨC | MÔ TẢ |
| send() | send(data) gửi dữ liệu tới server. Message data là string, ArrayBuffer, blob. |
| close() | Đóng kết nối đang tồn tại. |

### Docker

**Docker** là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa). Ban đầu viết bằng Python, hiện tại đã chuyển sang Golang.

#### Container trong Docker là gì?

Các containers cho phép lập trình viên đóng gói một ứng dụng với tất cả các phần cần thiết, chẳng hạn như thư viện và các phụ thuộc khác, và gói tất cả ra dưới dạng một package.

Bằng cách đó, nhờ vào container, ứng dụng sẽ chạy trên mọi máy Linux khác bất kể mọi cài đặt tùy chỉnh mà máy có thể có khác với máy được sử dụng để viết code.

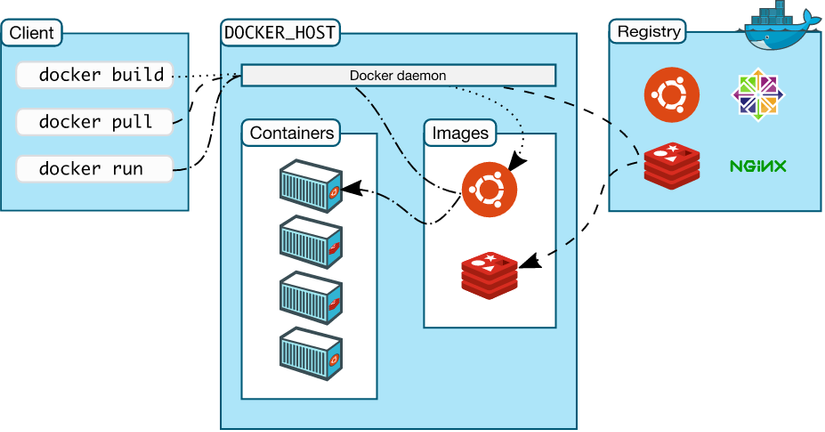
Theo một cách nào đó, Docker khá giống virtual machine. Nhưng tại sao Docker lại phát triển, phổ biến nhanh chóng? Đây là những nguyên nhân:

**Tính dễ ứng dụng:** Docker rất dễ cho mọi người sử dụng từ lập trình viên, sys admin… nó tận dụng lợi thế của container để build, test nhanh chóng. Có thể đóng gói ứng dụng trên laptop của họ và chạy trên public cloud, private cloud… Câu thần chú là “Build once, run anywhere”.

**Tốc độ:** Docker container rất nhẹ và nhanh, bạn có thể tạo và chạy docker container trong vài giây.

**Môi trường chạy và khả năng mở rộng:** Bạn có thể chia nhỏ những chức năng của ứng dụng thành các container riêng lẻ. Ví dụng Database chạy trên một container và Redis cache có thể chạy trên một container khác trong khi ứng dụng Node.js lại chạy trên một cái khác nữa. Với Docker, rất dễ để liên kết các container với nhau để tạo thành một ứng dụng, làm cho nó dễ dàng scale, update các thành phần độc lập với nhau.

#### Các khái niệm liên quan



Hình 4.7 Docker

**Docker Engine** : là thành phần chính của Docker, như một công cụ để đóng gói ứng dụng

**Docker Hub** : là một “github for docker images”. Trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng cho phép bạn dễ dàng tìm thấy những image mà bạn cần. Và chỉ cần pull về và sử dụng với một số config mà bạn mong muốn.

**Images**: là một khuôn mẫu để tạo một container. Thường thì image sẽ dựa trên 1 image có sẵn với những tùy chỉnh thêm. Ví dụ bạn build 1 image dựa trên image Centos mẫu có sẵn để chạy Nginx và những tùy chỉnh, cấu hình để ứng dụng web của bạn có thể chạy được. Bạn có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẽ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile.

**Container**: là một instance của một image. Bạn có thể create, start, stop, move or delete container dựa trên Docker API hoặc Docker CLI.

**Docker Client**: là một công cụ giúp người dùng giao tiếp với Docker host.

**Docker Daemon**: lắng nghe các yêu cầu từ Docker Client để quản lý các đối tượng như Container, Image, Network và Volumes thông qua REST API. Các Docker Daemon cũng giao tiếp với nhau để quản lý các Docker Service.

**Dockerfile**: là một tập tin bao gồm các chỉ dẫn để build một image .

**Volumes**: là phần dữ liệu được tạo ra khi container được khởi tạo.

Trên đây là những khái niệm cơ bản nhất về Docker. Ngoài ra còn nhiều khái niệm nữa như swarm, compose…

#### Quy trình thực thi của một hệ thống sử dụng Docker



Hình 4.8 Quy trình thực thi

Như trong hình vẽ, một hệ thống Docker được thực thi với 3 bước chính :

*Build -> Push -> Pull,Run*

* **Build**: Đầu tiên tạo một dockerfile, trong dockerfile này chính là code của chúng ta. Dockerfile này sẽ được Build tại một máy tính đã cài đặt Docker Engine. Sau khi build ta sẽ có được Container, trong Container này chứa ứng dụng kèm bộ thư viện của chúng ta.
* **Push:** Sau khi có được Container, chúng ta thực hiện push Container này lên cloud và lưu tại đó.
* **Pull, Run:** Nếu một máy tính khác muốn sử dụng Container chúng ta thì bắt buộc máy phải thực hiện việc Pull container này về máy, tất nhiên máy này cũng phải cài Docker Engine. Sau đó thực hiện Run Container này.

*Khi nào sử dụng Docker?*

Triển khai kiến trúc [Microservices](https://topdev.vn/blog/microserices-la-gi/).

Khi xây dựng ứng dụng và cần scale một cách linh hoạt.

Khi bạn muốn không tốn khá nhiều thời gian để config máy local và server cùng một môi trường để chạy được ứng dụng. Bạn chỉ cần build 1 lần chạy ở nhiều nơi mà thôi.

Sản phẩm của công ty bạn cần một cách tiếp cận mới về xây dựng, đẩy lên server, thực thi ứng dụng một cách nhanh chóng dễ dàng.

### Angular

#### Khái niệm

Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation, http service và có các tính năng như auto-complete , navigation , toolbar , menus ,… Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

Để học được angular thì cần biết các kiển thức cơ bản sau: HTML, CSS, JavaScript, TypeScript, Document Object Model (DOM)

#### Tại sao chúng ta nên chọn Angular

***Angular giúp nâng cao năng suất của các lập trình viên***

Việc phát triển Web đã có bước thay đổi đáng kể trong vài năm qua. Với phiên bản ECMAScript (ES) 2015 – chúng ta quen thuộc với cái tên ES6, với những class hay arrow function. Angular 2+ ứng dụng những tính năng mới này giúp việc code với Angular trở nên rõ ràng và dễ học hơn rất nhiều.

Thêm vào đó, với việc ứng dụng Typescript – một ngôn ngữ – hay là một bản nâng cấp đáng giá của Javascript, Angular kết hợp với Typescript, chúng ta có một công cụ tuyệt vời giúp xử lý các vấn đề hạn chế của JS như kiểm tra kiểu dữ liệu, refactor code an toàn hơn,… từ đó cũng hỗ trợ tốt hơn cho việc Debug cũng như giúp các Dev thực sự hiểu rõ mã nguồn của họ hơn.

***Cấu trúc phát triển rõ ràng***

Điều quan trọng của một Framework đối với lập trình viên đó là cấu trúc phát triển ứng dụng của nó, và Angular mang đến một kiến trúc rất rõ ràng, dựa trên ba yếu tố chính: class, các dependency được thêm vào và mô hình MVVM (model-view-view/model).

Angular sử dụng class trong ES6 với một loạt các thuộc tính để xây dựng toàn bộ các cấu trúc chủ chốt, giả sử bạn muốn tạo một Angular component – Tạo một class và thêm vào các thuộc tính cần thiết. Hay bạn muốn tạo một Angular module – Hãy tạo một class và thêm vào đó các thuộc tính cần thiết. Về cơ bản sẽ là như vậy, Angular cung cấp một cấu trúc rõ ràng để xây dựng từng tính năng cho ứng dụng của bạn.

Các dependency mạnh mẽ được sử dụng trong ứng dụng khi cần thiết, và khi cần tích hợp bất kì dependency nào, như HTTP hay Router, chúng ta chỉ cần thêm nó vào bên trong constructor của class.

Mô hình MVVM cũng giúp Angular chiếm lợi thế trong xây dựng ứng dụng client-side, thường ta sẽ có 3 điều cần quan tâm chính: đó là giao diện người dùng, mã nguồn điều khiển giao diện và mô hình dữ liệu (data) cho giao diện. Angular với MVVM phân biệt hoàn toàn rõ ràng các yếu tố trên nhờ mô hình MVVM.

Phần giao diện (view) được định nghĩa trong một template bao hàm HTML dành cho một component nhất định. Template có thể là toàn bộ Layout hoặc bất cứ mảnh ghép nào trong Layout đó.

Model được định nghĩa như là các thuộc tính của component class. Có thể hiểu là dữ liệu, dựa vào đó để phần View sử dụng để thực thi.

View/model là class quản lý cả view cũng như model. Là phần code sẽ xử lý việc truy xuất dữ liệu, đồng thời thực thi các tương tác của người dùng trên view.

Với việc ứng dụng các điểm tích cực của các thành phần trên, Angular khiến việc phát triển ứng dụng trở nên dễ dàng và hiệu quả hơn.

***Extensive binding***

Rất nhiều ứng dụng Web làm việc với dữ liệu (data). App sẽ truy xuất dữ liệu từ Server và hiển thị dữ liệu đó tới người dùng trên view, sử dụng template. Và các tương tác của người dùng sẽ được khiến dữ liệu thay đổi, được view ghi nhận và lưu lại trên server. Data Binding trong Angular giúp bạn thực thi tiến trình trên rất dễ dàng. Đơn thuần từ việc ràng buộc thành phần HTML trong template với các thuộc tính trong class và dữ liệu sẽ tự động xuất hiện trên màn hình. Với các tương tác của người dùng đòi hỏi thay đổi dữ liệu, Angular sử dụng phương pháp two-way binding. Bất kì thay đổi dữ liệu đến từ view sẽ tự động cập nhật thuộc tính “model” bên trong class.

Thêm vào đó, Angular cũng hỗ trợ property binding – cho phép chúng ta điều khiển DOM bằng cách ràng buộc thuộc tính HTML với thuộc tính của component class, data sẽ tự động xuất hiện bên trong view. Ví dụ, chúng ta ràng buộc thuộc tính hidden đối với một thẻ img với thuộc tính hideImg bên trong class. Khi thuộc tính hideImg nhận giá trị true, img sẽ tự động hidden và ngược lại khi hideImg nhận giá trị false, thẻ img sẽ tự động hiển thị tới người dùng.

Cuối cùng, Angular hỗ trợ event binding, có nghĩa là chúng ta có thể xử lí bất kì event nào từ phía view, như HTML event. Về cơ bản chúng ta sẽ gắn event với một method bên trong class. Mỗi khi event xuất hiện, method tương ứng sẽ được thực thi.

Extensive binding giúp quá trình hiển thị dữ liệu, điều khiển DOM, thực thi các event một cách trơn tru và dễ dàng.

***Hỗ trợ đầy đủ tính năng điều hướng (routing)***

Đa số các ứng dụng Web không chỉ có 1 view hay một page duy nhất, mà sẽ cung cấp nhiều view khác nhau tương ứng với với các chức năng chính. Ví dụ như một trang web với các trang giới thiệu, trang nội dung, trang chi tiết, trang đăng nhập, đăng ký,… Chúng ta sẽ cần hiển thị đúng view vào đúng thời điểm. Đó là mục đích của điều hướng (routing). Và Angular cung cấp đầy đủ tính năng cho việc này, chúng ta định nghĩa các đường dẫn (route) cho mỗi page view của ứng dụng. Và chúng ta sẽ kích hoạt route dựa trên tương tác của người dùng (user). Chúng ta có thể truyền thêm dữ liệu vào các route, giúp view hiển thị nội dung một cách dynamic, có thể bảo vệ route để người dùng chỉ có thể truy cập sau khi đã đăng nhập hoặc có quyền truy cập, có thể ngăn chặn việc người dùng ngay lập tức rời một trang khi các thao tác còn dang dở cho đến khi họ thực sự xác nhận việc rời đi hoặc lưu lại tiến trình sử dụng,…Angular đồng thời cũng hỗ trợ child-route cho việc điều hướng bên trong một route. Việc điều hướng giữa các view bên trong ứng dụng Angular thực sự rất linh hoạt và mạnh mẽ.

***Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất của ứng dụng.***

Kích thước và hiệu năng có mối liên quan mật thiết khi chúng ta làm việc trên nền tảng Web. Một component nhỏ hơn sẽ giúp nâng cao hiệu suất khởi động – giảm cả thời gian download cũng như thời gian cũng như thời gian compile trên trình duyệt. Giảm kích thước component và giúp tăng hiệu suất là một ưu điểm cũng như mục tiêu mà Angular mong muốn mang đến cho các lập trình viên.

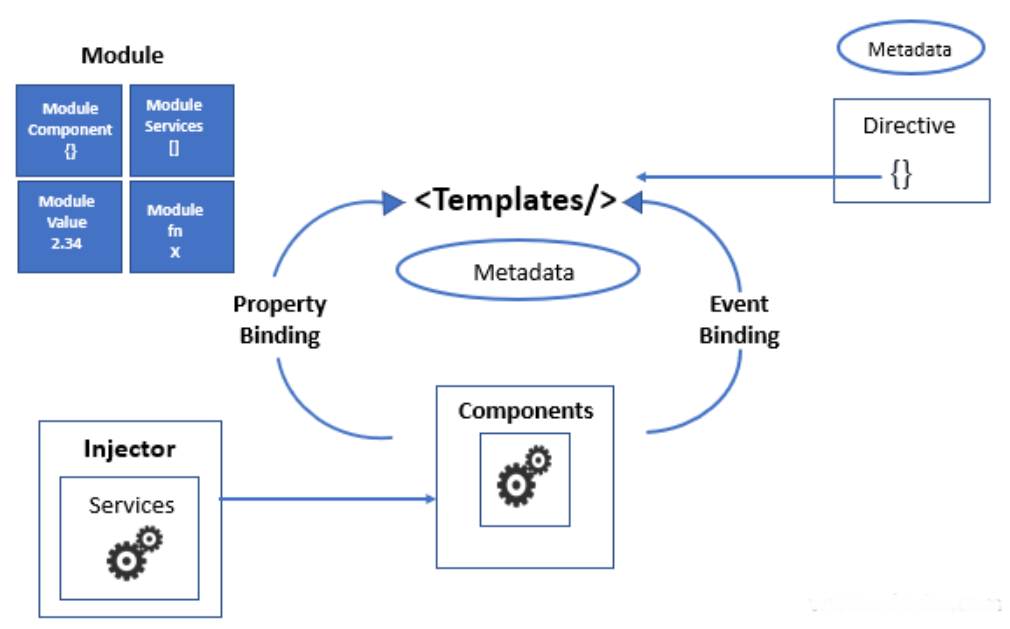
Giảm kích thước ứng dụng có thể thực hiện bằng nhiều cách. Đầu tiên chúng ta có thể giảm tối đa kích thước của từng component tới mức tối thiểu có thể. Tiếp theo các component sẽ được sắp xếp bên trong Angular Module bằng 1 cách đề cho các nhóm logic có liên quan đến nhau sẽ được download cùng với nhau. Và bước thứ ba, lazy loading bên trong các route sẽ chỉ downloaad những module cần thiết cho việc hiển thị nội dung cần thiết tới người dùng, và sẽ không bao giờ download những nội dung không cần thiết.

Chúng ta có một trình biên dịch tên là AOT, trình biên dịch này sẽ chạy một lần trong thời gian build ứng dụng. Trình duyệt sau đó sẽ download phiên bản chưa được biên dịch của ứng dụng và render ứng dụng tới người dùng ngay lập tức mà không cần biên dịch nó lần đầu trong trình duyệt. Thêm nữa là sẽ không cần download trình biên dịch Angular, giúp làm giảm đáng kể kích thước (size) của ứng dụng cần tải về.

***Document và cộng đồng (community)***

Document cho Angular 2+ (angular.io) rất đầy đủ và chi tiết, bao hàm giới thiệu cơ bản giúp bạn làm quen nhanh chóng với Angular, giới thiệu chi tiết, từ cơ bản đến nâng cao các API của Angular, cũng như có hẳn một Tutorial Basic được xây dựng nên bởi Angular team, cung cấp cho bạn nhanh chóng nắm bắt các thuộc tính cơ bản của Framework.

#### Cơ chế hoạt động của Angular



Hình 4.9 Cơ chế hoạt động của Angular

* Modules: Trong Angular một modules là một nhóm các khối xây dựng như components, services, templates, directive, pipes kết hợp để xây dựng một đơn vị chức năng, ví dụ điển hình của modules là modules httpclient của Angular, vì nó giúp chúng ta thực hiện truy cập mạng. Nhiều modules kết hợp với nhau để xây dựng một ứng dụng angular.
* Component: Component là các thành phần xây dựng nên; chúng hoàn toàn có thể tái sử dụng trên modules và ứng dụng. Một component trong Angular bao gồm một file class, một template (file HTML) và ít nhất một file css/scss để tạo kiểu cho thành phần.
* Services: Các service là các lớp đối tượng singleton trong ứng dụng Angular. Chèn các dịch vụ trong các thành phần là một mẫu phổ biến để chia sẻ dữ liệu giữa các thành phần trong Angular.
* Template: Template là dạng xem HTML với sự trợ giúp của file class, khối xây dựng tạo mẫu của dạng xem người dùng.
* Metadata: Trong Angular, metadata được sử dụng để trang trí một lớp để tạo cho nó một cấu hình cơ bản.
* Data binding: Đối với data-binding Angular hỗ trợ liên kết thuộc tính, sự kiện và thuộc tính từ directives thành phần đến template, với chỉ thị ngModel Angular cũng cung cấp liên kết dữ liệu hai chiều.
* Directives: Trong Angular, directives là cách để hướng dẫn DOM. Các chỉ thị đính kèm hành vi cụ thể với DOM và giúp chúng ta chuyển đổi giao diện người dùng theo logic nghiệp vụ hoặc các tương tác của người dùng.

## Đánh giá

Đã triển khai thành công ở cả tại máy và cho lên server. Việc sử dụng docker để deploy remote rất tiện và nhanh, không phải thao tác quá nhiều bước, tuy nhiên phải cài thêm docker trên server.

# KẾT LUẬN

Sau thời gian nghiên cứu cùng với sự giúp đỡ của thầy giáo và sự góp ý của các bạn, tôi đã hoàn thành đồ án tốt nghiệp với đề tài: ***Xây dựng hệ thống học trực tuyến, kiểm tra, đánh giá online***. Qua đó nắm bắt được quá trình và các kỹ năng cần thiết để xây dựng một hệ thống đào tạo. Hệ thống được thiết kế để có khả năng mở rộng sau này và được chia làm các phần rõ ràng.

Trong quá trình thực hiện đồ án do phải tiếp cận với nhiều lĩnh vực mới nên tôi đã gặp phải không ít khó khăn, tuy nhiên với sự giúp đỡ nhiệt tình của thầy, những người đi trước và bạn bè cùng với nỗ lực của bản thân, đề tài đã được hoàn thành đúng hạn và thu được kết quả khả quan. Bên cạnh đó, thời gian thực hiện có hạn nên một vài phần của đề tài có kết quả chưa được hoàn toàn như mong đợi.

## Kết quả đạt được

* Đã xây dựng thành công cơ bản website dạy học trực tuyến và quản lý khóa học.
* Triển khai hệ thống thành công trên cả môi trường thử nghiệm và server.

## Hướng phát triển

* Hoàn thiện chương trình.
* Thêm và nâng cấp các tính năng mới.
* Ứng dụng học thuật, khai phá dữ liệu nhiều hơn nữa vào hệ thống, giúp hệ thống trở nên thông minh hơn.

# TÀI LIỆU THAM KHẢO

1. B Akkoyunlu, M Y Soylu, ‘*A Study on Students' Views About Blended Learning Environment*’, Turkish Online Journal of Distance Education-TOJDE, volume 7, issue 3
2. F Z Azizan, ‘*Blended Learning in Higher Education Institution in Malaysia*’, Proceedings of Regional Conference on Knowledge Integration in ICT
3. M F Badawi, ‘*Using Blended Learning for Enhancing EFL Prospective Teachers' Pedagogical Knowledge and* Performance’, *Conference paper: Learning & Language -The Spirit of the Age*’
4. C C Chen, K T Jones, ‘*Blended Learning vs. Traditional Classroom Settings: Assessing Effectiveness and Student Perceptions in an MBA Accounting Course*’, The Journal of Educators Online, volume 4, issue 1
5. https://vi.wikipedia.org/wiki/Giáo\_dục\_trực\_tuyến