|  |  |
| --- | --- |
| HÀ QUANG HUY |  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC |
| NGÀNH CÔNG NGHỆ THÔNG TIN |
| NGÀNH CÔNG NGHỆ THÔNG TIN | **ĐỀ TÀI: XÂY DỰNG HỆ THỐNG HỖ TRỢ HỌC TRỰC TUYẾN ACADEMY** |
|  |
|  |
| **CBHD : Ths. Nguyễn Lan Anh** |
| **Sinh viên : HÀ QUANG HUY** |
| **Mã số sinh viên: 2018603648** |
|  |
|  |
| ***Hà Nội – Năm 2022*** |

# LỜI CẢM ƠN

Để hoàn thành báo cáo cho đề tài đồ án tốt nghiệp này, trước hết em xin được gửi lời cảm ơn tới trường Đại học Công nghiệp Hà Nội, khoa Công nghệ thông tin đã tạo điều kiện cho em được học tập và nghiên cứu để hoàn thành đồ án tốt nghiệp của mình. Em xin gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin đã giảng dạy bằng tất cả tâm huyết, sự nhiệt tình, truyền đạt cho chúng em những tri thức, kinh nghiệm quý báu của mình để từ đó chúng em có một hành trang tốt nhất bước vào con đường lập nghiệp của riêng mình. Đặc biệt em xin gửi lời cảm ơn sâu sắc đến cô giáo **ThS. Nguyễn Lan Anh**, cô đã luôn tận tình chỉ dạy, định hướng và giúp đỡ em trong trong suốt quá trình nghiên cứu và thực hiện đồ án tốt nghiệp của em.  
Em đã cố gắng hoàn thiện báo cáo đồ án tốt nghiệp một cách tốt nhất, tuy nhiên không thể tránh được những thiếu sót. Em rất mong nhận được sự góp ý của các thầy cô để đồ án tốt nghiệp này của em được hoàn thiện hơn nữa. Lời cuối cùng em xin chúc thầy cô luôn dồi dào sức khỏe và thành công trong cuộc sống.

Em xin chân thành cảm ơn!

# MỤC LỤC

[DANH MỤC BẢNG BIỂU vi](#_Toc103282430)

[DANH MỤC HÌNH VẼ vii](#_Toc103282431)

[MỞ ĐẦU 1](#_Toc103282432)

[1. Lý do chọn đề tài 1](#_Toc103282433)

[2. Đối tượng, mục tiêu phạm vi nghiên cứu 1](#_Toc103282434)

[2.1 Mục tiêu nghiên cứu 1](#_Toc103282435)

[2.2 Đối tượng nghiên cứu 1](#_Toc103282436)

[3. Ý nghĩa khoa học và thực tiễn của đề tài 2](#_Toc103282437)

[3.1 Ý nghĩa khoa học, thực tiễn 2](#_Toc103282438)

[Chương 1 - TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU 4](#_Toc103282439)

[1.1 Giới thiệu chung 4](#_Toc103282440)

[1.2 Cơ sở lý thuyết 4](#_Toc103282441)

[1.2.1 Dạy học trực tuyến E – Learning 4](#_Toc103282442)

[1.2.2 Giới thiệu về Angular 6](#_Toc103282443)

[1.2.3 Giới thiệu về FastAPI 12](#_Toc103282444)

[1.2.4 MongoDB 13](#_Toc103282445)

[Chương 2 - PHÂN TÍCH THIẾT KẾ HỆ THỐNG 17](#_Toc103282446)

[2.1 Khảo sát bài toán 17](#_Toc103282447)

[2.2 Giới thiệu về hệ thống 17](#_Toc103282448)

[2.3 Phân tích yêu cầu hệ thống 17](#_Toc103282449)

[2.3.1 Hệ thống 17](#_Toc103282450)

[2.3.2 Người sử dụng 17](#_Toc103282451)

[2.3.3 Yêu cầu về chức năng 18](#_Toc103282452)

[2.4 Thiết kế hệ thống 18](#_Toc103282453)

[2.4.1 Biểu đồ Use case: 18](#_Toc103282454)

[2.4.2 Biểu đồ lớp: 19](#_Toc103282455)

[2.4.3 Use case đăng ký 20](#_Toc103282456)

[2.4.4 Use case đăng nhập 21](#_Toc103282457)

[2.4.5 Use case xem danh sách khóa học 23](#_Toc103282458)

[2.2.6 Use case quên mật mật khẩu 24](#_Toc103282459)

[2.4.7 Use case xem chi tiết khóa học 25](#_Toc103282460)

[2.4.8 Use case quản lý khóa học 28](#_Toc103282461)

[2.4.9 Use case quản lý bài tập, bài giảng 30](#_Toc103282462)

[2.4.10 Use case thống kê khóa học 31](#_Toc103282463)

[2.4.11 Use case quản lý danh mục 32](#_Toc103282464)

[2.5 Thiết kế giao diện 34](#_Toc103282465)

[2.5.1 Màn hình trang chủ 34](#_Toc103282466)

[2.5.2 Màn hình chi tiết khóa học 35](#_Toc103282467)

[2.5.3 Màn hình đăng ký 36](#_Toc103282468)

[2.5.5 Màn hình quên mật khẩu 38](#_Toc103282469)

[2.5.6 Màn hình quản lý khóa học 39](#_Toc103282470)

[2.5.7 Màn hình quản lý bài tập, bài giảng 40](#_Toc103282471)

[2.5.8 Màn hình giỏ hàng 41](#_Toc103282472)

[2.5.9 Màn hình thanh toán 42](#_Toc103282473)

[2.6 Thiết kế cơ sở dữ liệu 43](#_Toc103282474)

[2.6.1 User 43](#_Toc103282475)

[2.6.2 ConfirmToken 45](#_Toc103282476)

[2.6.3 Content 45](#_Toc103282477)

[2.6.4 Parameter 46](#_Toc103282478)

[2.6.5 Function 47](#_Toc103282479)

[2.6.6 OutputTestcase 48](#_Toc103282480)

[2.6.7 Testcase 49](#_Toc103282481)

[2.6.8 CodingExamHistory 50](#_Toc103282482)

[2.6.9 Course 50](#_Toc103282483)

[2.6.10 CourseRating 53](#_Toc103282484)

[2.6.11 ContentDiscussion 53](#_Toc103282485)

[2.6.12 Resourse 54](#_Toc103282486)

[2.6.13 Answer 55](#_Toc103282487)

[2.6.14 LeanerAnswerOfQuestion 55](#_Toc103282488)

[2.6.15 Question 56](#_Toc103282489)

[2.6.16 MultipleChoiceExamHistory 57](#_Toc103282490)

[Chương 3 - KẾT QUẢ ĐẠT ĐƯỢC 59](#_Toc103282491)

[3.1 Cài đặt chương trình 59](#_Toc103282492)

[3.2 Kết quả thu được 62](#_Toc103282493)

[3.3 Một số giao diện của hệ thống 63](#_Toc103282494)

[3.3.1 Giao diện đăng nhập 63](#_Toc103282495)

[3.3.2 Giao diện đăng ký thành viên 64](#_Toc103282496)

[3.3.3 Giao diện trang chủ 65](#_Toc103282497)

[3.3.4 Giao diện xem chi tiết khóa học 65](#_Toc103282498)

[3.3.5 Giao diện danh sách khóa học đã tạo 66](#_Toc103282499)

[3.3.6 Giao diện tạo tên khóa học 66](#_Toc103282500)

[3.3.7 Giao diện lựa chọn thể loại khóa học 67](#_Toc103282501)

[3.3.8 Giao diện tạo mục đích của môn học 68](#_Toc103282502)

[3.3.9 Giao diện tạo nội dung khóa học 69](#_Toc103282503)

[3.3.10 Giao diện tạo bài học 69](#_Toc103282504)

[3.3.11 Giao diện tạo bài tập trắc nghiệm cho khóa học 72](#_Toc103282505)

[3.3.12 Giao diện tạo bài tập lập trình 73](#_Toc103282506)

[3.3.13 Giao diện tạo học phí khóa học 74](#_Toc103282507)

[3.3.14 Giao diện tạo trang đích của khóa học 75](#_Toc103282508)

[3.3.15 Giao diện thêm vào giỏ hàng 76](#_Toc103282509)

[3.3.16 Giao diện thanh toán 76](#_Toc103282510)

[KẾT LUẬN 78](#_Toc103282511)

[TÀI LIỆU THAM KHẢO 79](#_Toc103282512)

# DANH MỤC BẢNG BIỂU

[Bảng 2.2: Use case đăng nhập 38](#_Toc103095596)

[Bảng 2.3: Use case xem danh sách khóa học 40](#_Toc103095597)

[Bảng 2.4 Use case quên mật khẩu 41](#_Toc103095598)

[Bảng 2.7 : Use case xem chi tiết khóa học 42](#_Toc103095599)

[Bảng 2.6: Use case quản lý giỏ hàng 43](#_Toc103095600)

[Bảng 2.7: Use case thanh toán hóa học 44](#_Toc103095601)

[Bảng 2.8: Use case quản lý khóa học 46](#_Toc103095602)

[Bảng 2.9: Use case quản lý bài tập, bài giảng 47](#_Toc103095603)

[Bảng 2.10: Use case thống kê khóa học 49](#_Toc103095604)

[Bảng 2.11: Use case thanh toán cho người dùng 50](#_Toc103095605)

[Bảng 2.12: User 61](#_Toc103095606)

[Bảng 2.13: ConfirmToken 62](#_Toc103095607)

[Bảng 2.14: Content 62](#_Toc103095608)

[Bảng 2.15: Parameter 63](#_Toc103095609)

[Bảng 2.16: Function 64](#_Toc103095610)

[Bảng 2.17: OutputTestcase 64](#_Toc103095611)

[Bảng 2.18: Testcase 65](#_Toc103095612)

[Bảng 2.19: CodingExamHistory 65](#_Toc103095613)

[Bảng 2.20: Course 66](#_Toc103095614)

[Bảng 2.21: CourseRating 67](#_Toc103095615)

[Bảng 2.22: ContentDiscussion 68](#_Toc103095616)

[Bảng 2.23 Resourse 69](#_Toc103095617)

[Bảng 2.24: Answer 69](#_Toc103095618)

[Bảng 2.25: LeanerAnswerOfQuestion 70](#_Toc103095619)

[Bảng 2.26: Question 71](#_Toc103095620)

[Bảng 2.27: MultipleChoiceExamHistory 73](#_Toc103095621)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Biểu đồ Use case 37](#_Toc103095772)

[Hình 2.2: Biểu đồ lớp 37](#_Toc103095773)

[Hình 2.3: Biểu đồ trình tự đăng ký 38](#_Toc103095774)

[Hình 2.4: Biểu đồ trình tự đăng nhập 40](#_Toc103095775)

[Hình 2.5: Biểu đồ trình tự xem danh sách khóa học 41](#_Toc103095776)

[Hình 2.6: Biểu đồ trình tự quên mật khẩu 42](#_Toc103095777)

[Hình 2.7: Biểu đồ trình tự xem chi tiết sản phẩm 43](#_Toc103095778)

[Hình 2.8: Biểu đồ trình tự quản lý giỏ hàng 45](#_Toc103095779)

[Hình 2.9: Biểu đồ trình tự thanh toán 46](#_Toc103095780)

[Hình 2.10: Biểu đồ trình tự quản lý khóa học 47](#_Toc103095781)

[Hình 2.11: Biểu đồ trình tự quản lý bài tập bài giảng 49](#_Toc103095782)

[Hình 2.12: Biểu đồ trình tự thống kê khóa học 50](#_Toc103095783)

[Hình 2.13 Biểu đồ trình tự thanh toán cho người dùng 52](#_Toc103095784)

[Hình 2.14: Màn hình trang chủ 52](#_Toc103095785)

[Hình 2.15: Màn hình chi tiết khóa học 53](#_Toc103095786)

[Hình 2.16: Màn hình đăng ký 54](#_Toc103095787)

[Hình 2.17: Màn hình đăng nhập 55](#_Toc103095788)

[Hình 2.18: Màn hình quên mật khẩu 56](#_Toc103095789)

[Hình 2.19 Màn hình quản lý khóa học 57](#_Toc103095790)

[Hình 2.20: Màn hình quản lý bài tập bài giảng 58](#_Toc103095791)

[Hình 2.21: Màn hình giỏ hàng 59](#_Toc103095792)

[Hình 2.22: Màn hình thanh toán 61](#_Toc103095793)

[Hình 3.1: Câu lệnh cài đặt Angular CLI 75](#_Toc103095794)

[Hình 3.2: Kiểm tra phiên bản Angular CLI 75](#_Toc103095795)

[Hình 3.3: Tạo dự án Angular 75](#_Toc103095796)

[Hình 3.4: Tạo routing cho dự án 76](#_Toc103095797)

[Hình 3.5: Chọn chương trình tiền xử lý SCSS 76](#_Toc103095798)

[Hình 3.6: Giao diện ban đầu của dự án 77](#_Toc103095799)

[Hình 3.7: Giao diện đăng nhập 78](#_Toc103095800)

[Hình 3.8: Giao diện đăng ký thành viên 79](#_Toc103095801)

[Hình 3.9: Giao diện trang chủ 80](#_Toc103095802)

[Hình 3.10: Giao diện xem chi tiết khóa học 80](#_Toc103095803)

[Hình 3.11: Giao diện danh sách khóa học đã tạo 81](#_Toc103095804)

[Hình 3.12: Giao diện tạo tên khóa học 81](#_Toc103095805)

[Hình 3.13: Giao diện lựa chọn thể loại khóa học 82](#_Toc103095806)

[Hình 3.14: Giao diện tạo mục đích của môn học 83](#_Toc103095807)

[Hình 3.15: Giao diện 84](#_Toc103095808)

[Hình 3.16: Giao diện nhập tiêu đề bài học 85](#_Toc103095809)

[Hình 3.17: Giao diện thêm video cho bài học 85](#_Toc103095810)

[Hình 3.18: Giao diện tạo mô tả cho bài học 85](#_Toc103095811)

[Hình 3.19: Giao diện thêm file tài liệu cho bài học 86](#_Toc103095812)

[Hình 3.20: Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài 87](#_Toc103095813)

[Hình 3.21: Giao diện tạo câu hỏi cho bài tập trách nghiệm 87](#_Toc103095814)

[Hình 3.22: Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài 88](#_Toc103095815)

[Hình 3.23: Giao diện nhập nội dung cho bài tập trắc nghiệm 88](#_Toc103095816)

[Hình 3.24: Giao diện tạo học phí cho khóa học 89](#_Toc103095817)

[Hình 3.25: Giao diện tạo trang đích khóa học 90](#_Toc103095818)

[Hình 3.26: Giao diện thêm vào giỏ hàng 91](#_Toc103095819)

[Hình 3.27: Giao diện giỏ hàng 91](#_Toc103095820)

[Hình 3.28: Giao diện cổng thanh toán 92](#_Toc103095821)

# 

# MỞ ĐẦU

## Lý do chọn đề tài

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 đang lây lan 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.

## Đối tượng, mục tiêu phạm vi nghiên cứu

### Mục tiêu nghiên cứu

* Xác định và làm rõ các vấn đề liên quan đến đề tài.
* Khảo sát phân tích thiết kế hệ thông cho website.
* Xây dựng website hỗ trợ học trực tuyến Academy đáp ứng đủ các nhu cầu được đặt ra.
* Về phía người dùng, đảm bảo quá trình xem, phẩn hổi và học tập online một cách thuận tiện hiệu quả.
* Về phía người giảng dạy giúp quản lý học viên, bài học nội dung kiểm tra một cách hiệu quả.

### Đối tượng nghiên cứu

* Dạy học trực tuyến E – Learning
* Phần mềm Visual Studio Code 2019
* Framework Angular
* Framework FastAPI
* Hệ quản trị cơ sở dữ liệu MongoDB

## Ý nghĩa khoa học và thực tiễn của đề tài

Ý nghĩa khoa học, thực tiễn

* + - * 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.
* 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.
* Kết quả mong muốn
* Xác định và làm rõ các vấn đề liên quan đến đề tài.
* Khảo sát, phân tích thiết kế hệ thống cho website.
* Xây dựng được website giúp các trung tâm có thể xây dựng hệ thống khóa học trực tuyến
* Xây dựng hệ thống kiểm tra, đánh giá chất lượng trong mỗi khóa học để đảm bảo chất lượng đầu vào và đầu ra của học viên.

# Chương 1 - TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## Giới thiệu chung

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.

Đó là những lý do để em chọn đề tài **“Xây dựng hệ thống hỗ trợ học trực truyến Academy”**

## Cơ sở lý thuyết

### Dạy học trực tuyến E – Learning

#### Khái niệm

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,...).

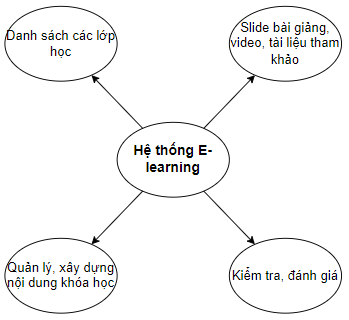
Các bài giảng, tài liệu (dưới dạng văn bản, hình ảnh, video…) được đưa lên các nền tảng và người dùng có thể dễ dàng truy cập và học mọi lúc mọi nơi. Bên cạnh đó còn có các khóa học cùng thời gian thực có sự tham gia và tương tác giữa giáo viên và học viên.

#### 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.

#### Mô hình hệ thống E – Learning



Hình 1.1 Mô hình hệ thống E-Learning

### Giới thiệu về 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.

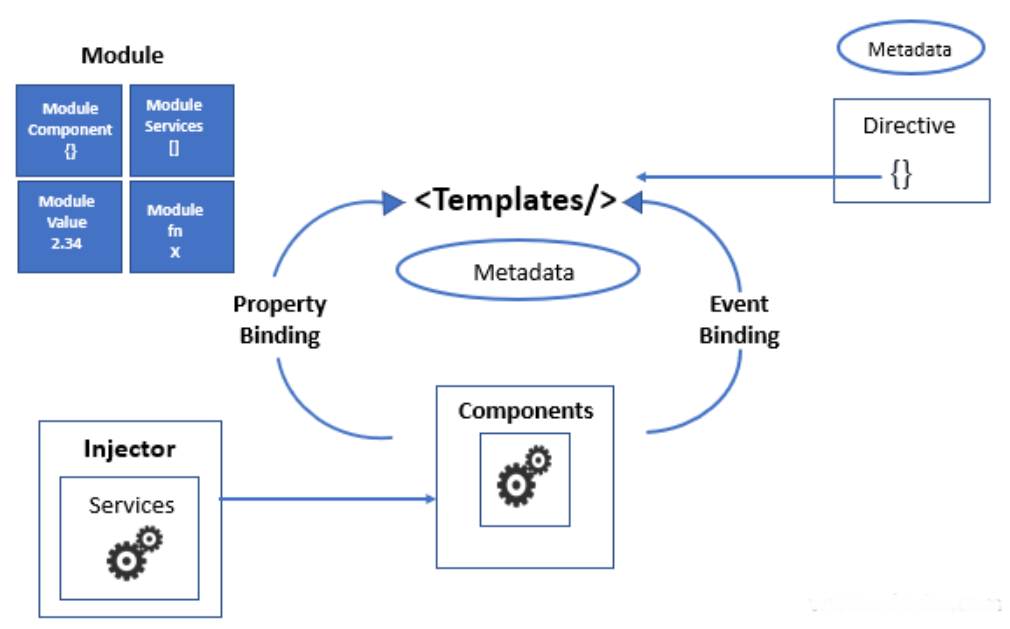
##### Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất

* + 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 (Comunity)

* + 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 1.2: Cơ chế hoạt động 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.

### Giới thiệu về FastAPI

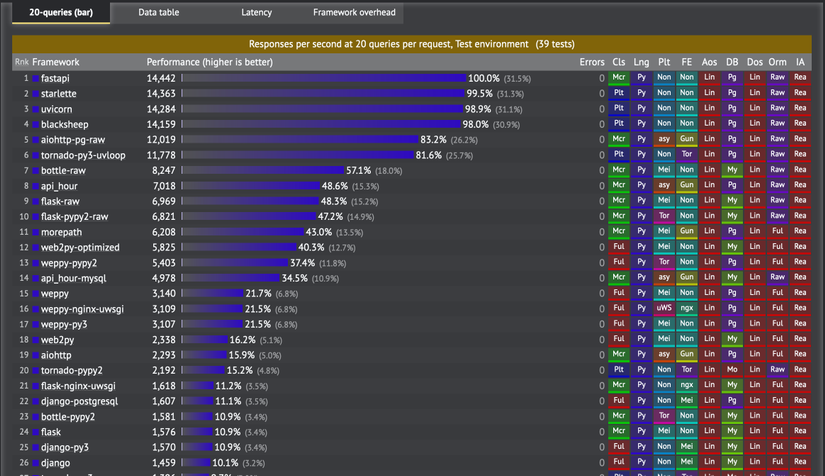
#### Khái niệm

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 1.3: Biểu đồ lượt sử dung FastAPI

* 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.

### MongoDB

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

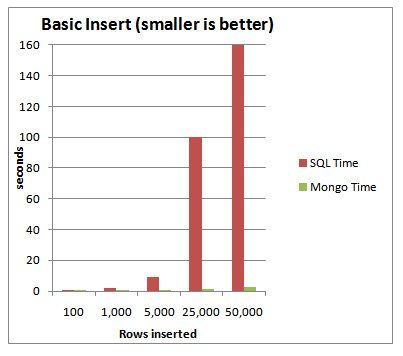
* NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.
* NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.
* NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng 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.

#### 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 1.4: Hiệu năng tốc độ truy vấn giữa MongDB và SQL

#### 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...

# Chương 2 - PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Khảo sát bài toán

Với mục tiêu phát triển hệ thống hộ trợ học trực tuyến nhằm mục đích tạo một môi trường học tập thuận lợi dễ dàng tiếp cận với người học và người giảng dạy, gồm những chức năng xem video, xem bài giảng, làm kiểm tra trắc nghiệm, kiểm tra lập trình trực tuyến. Người học có thể học tập một cách tự do và chủ động có thể tự đánh giá năng lực của bản thân. Người dạy học có thể trực tiếp tạo bài giảng cũng như đăng tải nhưng video và tài liệu. Đáp ứng nhu cầu học tập và giảng dạy trực tuyến đang được rất quan tâm hiện tại.

Hệ thống website hỗ trợ học tập trực tuyến Academy giúp cho người học trực tuyến có một lộ trình học cụ thể, đánh giá năng lực một cách khách quan. Người giảng dạy cũng có thể tạo khóa học một cách dễ dàng và chuyên ngiệp, có thể đăng tải tài liệu cũng như video để giúp người học học tập hiệu quả hơn.

## 2.2 Giới thiệu về hệ thống

Để đẩy mạnh phát triển việc học trực tuyến. Hệ thống website hỗ trợ học trực tuyến Academy đã được xây dựng. Hệ thống đã sử dụng công nghệ Angular, FastAPI cùng với hệ cơ sở dữ liệu MongoDB, là một website hỗ trợ về việc học trực tuyến, giúp cho người sử dụng có thể học trực tuyến thông qua các tài liệu, video, slider bài giảng. Đồ thời giúp cho người dùng có thể chia sẻ khóa học do chính mình xây dựng.

## 2.3 Phân tích yêu cầu hệ thống

### 2.3.1 Hệ thống

Xây dựng hệ thống có giao diện đẹp mắt, thân thiện, dễ sử dụng và có bố cục cũng như trang trí phù hợp với người dùng.

### 2.3.2 Người sử dụng

Người sử dụng có thể năm bắt được các thông tin cơ bản về khóa học như: nội dung, kết quả đạt được,… Ngoài việc có thể xem danh sách các khóa học và chi tiết của từng khóa học, người dùng có thể mua khóa học và thanh toán trực tuyến trên website và có thể chia sẻ những khóa học của mình trên chính website với những người dùng khác.

### 2.3.3 Yêu cầu về chức năng

#### Các chức năng chính của hệ thống:

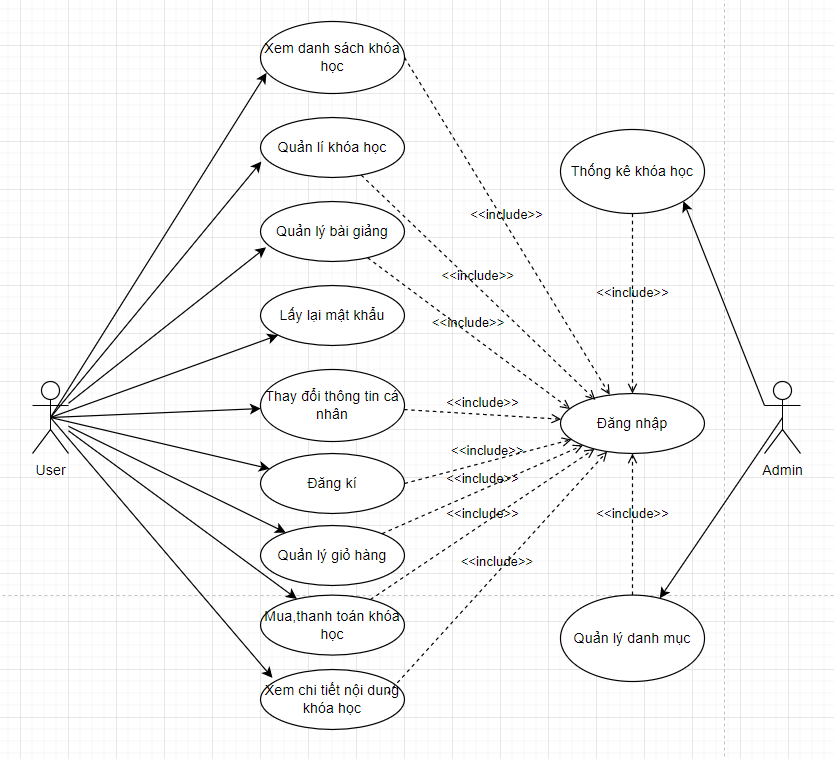
* + Đăng nhập/ Đăng ký
  + Đổi mật khẩu/ Quên mật khẩu
  + Thay đổi thông tin
  + Xem danh sách khóa học/ xem chi tiết khóa học
  + Thanh toán khóa học
  + Tạo khóa học
  + Xem danh sách khóa học đã mua
  + Xem danh sách khóa học đã tạo
  + Làm bài tập trắc nghiệm/ coding
  + Thống kê khóa học
  + Quản lý danh mục

#### Yêu cầu phi chức năng:

* + Giao điện đồ họa
  + Hoạt động của hệ thông
  + Độ tin cậy của hệ thống
  + Hiệu năng của hệ thống

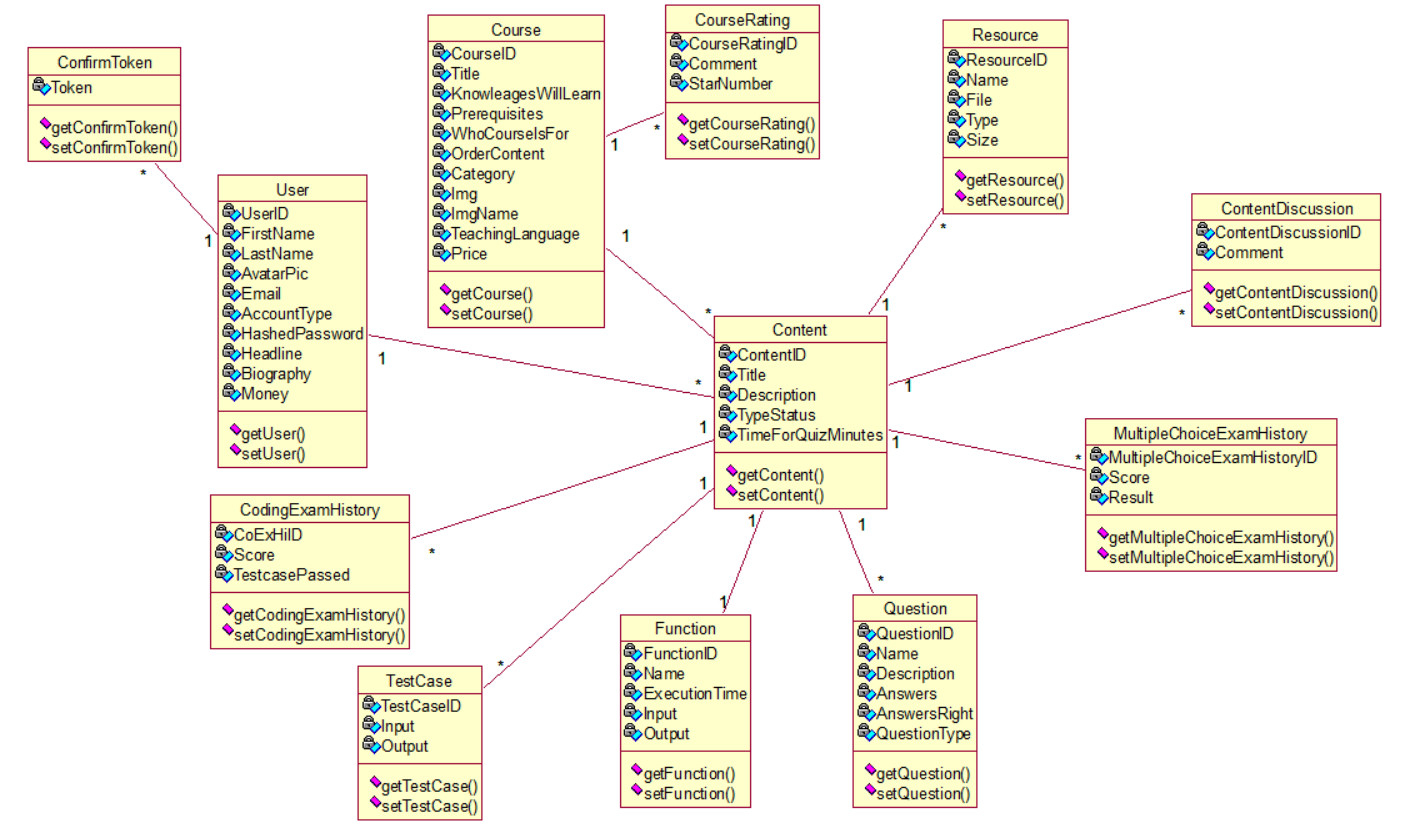
## 2.4 Thiết kế hệ thống

### 2.4.1 Biểu đồ Use case:



Hình 2.1: Biểu đồ Use case

### 2.4.2 Biểu đồ lớp:



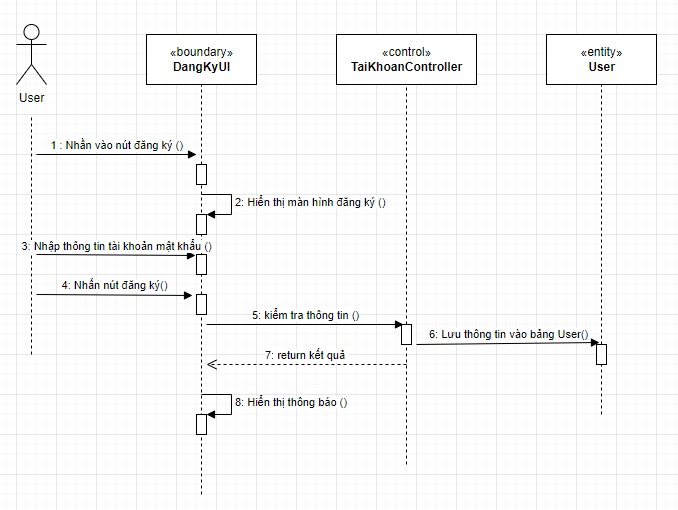
Hình 2.2: Biểu đồ lớp

### 2.4.3 Use case 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 | User |
| Điều kiện | Chưa đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi User nhấn vào ô “Đăng ký ”. Hệ thống sẽ hiển thị màn hình để nhập các thông tin cần thiết. * User nhập các thông tin theo yêu cầu của hệ thống và kích vào nút “Đăng ký”. Hệ thống sẽ lưu thông tin khách hàng vào bảng “User” và hiển thị lên màn hình thông báo đăng kyys thành công. Use case kết thúc. |
| 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 |

Bảng 2.1: Use case đăng ký

Biểu đồ trình tự:



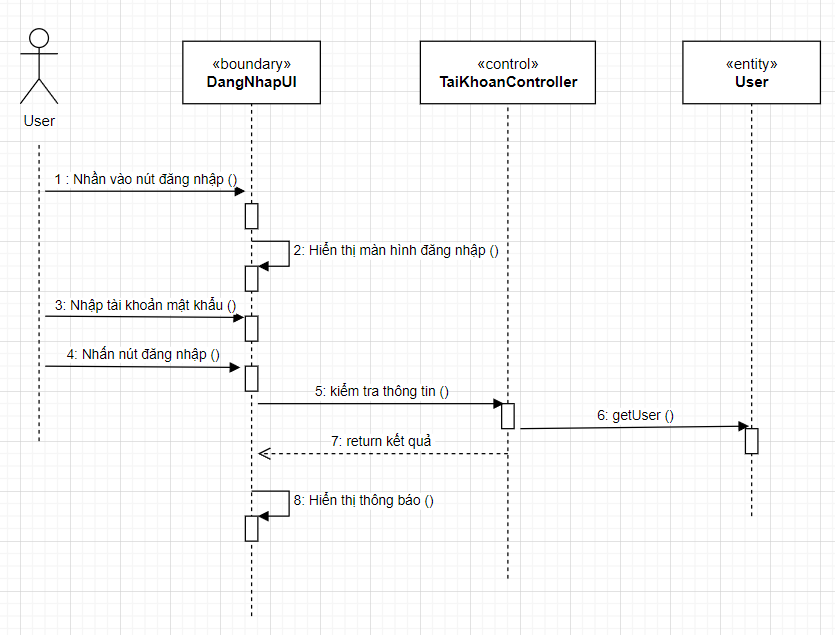
Hình 2.3: Biểu đồ trình tự đăng ký

### 2.4.4 Use case 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 nhập tài khoản để có thể sử dụng đầy đủ các chức năng của hệ thống |
| Tác nhân | User, Admin |
| Điều kiện | Đã đăng ký tài khoản |
| Luồng sự kiện chính | * Use case này bắt đầu khi User nhấn vào nút “Đăng nhập” trên giao diện của trang web. Hệ thống hiển thị một màn hình đăng nhập * User nhập tài khoản vào mật khẩu vào nhấn nút “Đăng nhập”. Hệ thống sẽ kiểm tra thông tin trong bảng “User”. Use case kết thúc. |
| 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 |

###### Bảng 2.2: Use case đăng nhập

Biểu đồ trình tự:



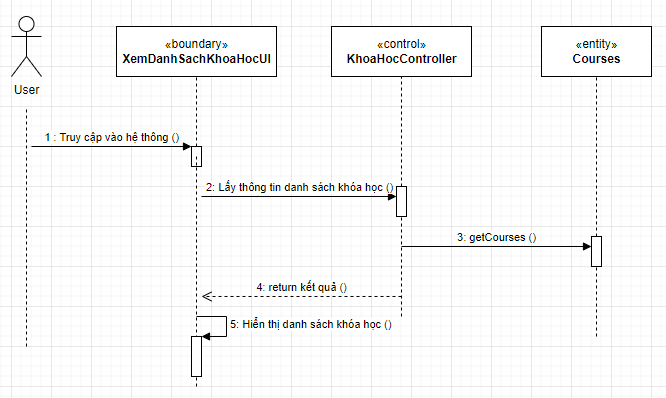
Hình 2.4: Biểu đồ trình tự đăng nhập

### 2.4.5 Use case 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 | User |
| Điều kiện | Truy cập vào phần mềm kiểm tra và đánh giá chất lượng |
| Luồng sự kiện chính | * Use case này bắt đầu khi khách hàng truy cập vào hệ thống. Hệ thống sẽ lấy ra danh sách khóa học có trong bảng “Course” bao gồm hình ảnh, tên, giá tiền, đánh giá của khóa học. Use case kết thúc |
| Luồng sự kiện phụ | Người dùng có thể xem danh sách khóa học đã sở hữu sau khi đăng nhập. Use case kết thúc |
| Điều kiện sau | Hiển thị tất cả danh sách khóa học của hệ thống |

###### Bảng 2.3: Use case xem danh sách khóa học

Biểu đồ trình tự:



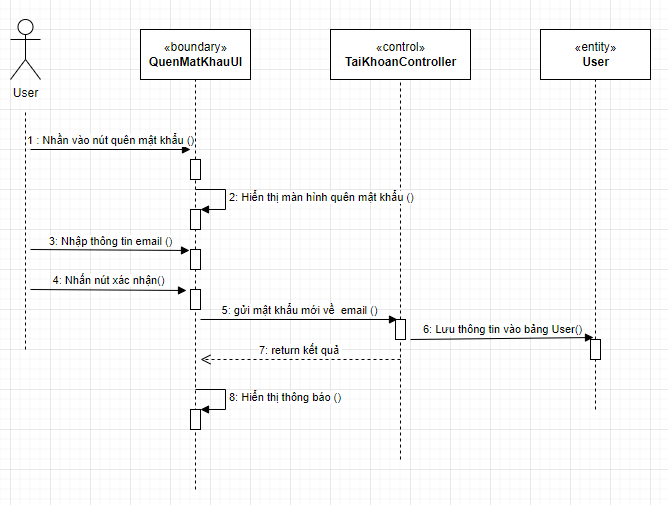
Hình 2.5: Biểu đồ trình tự xem danh sách khóa học

### 2.2.6 Use case quên mật 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 | User, Admin |
| Điều kiện | Truy cập vào trang đăng nhập |
| Luồng sự kiện chính | * Use case này bắt đầu khi User nhấn vào nút “Quên mật khẩu” trên màn hình hiển thị. Hệ thống sẽ hiển thị màn hình để User nhập Email đăng nhập, mật khẩu mới sẽ được gửi vào gmail của User. |
| Luồng sự kiện phụ | * Khi người dùng nhập sai thông tin gmail hệ thống sẽ không gửi được mật khẩu về email |
| Điều kiện sau | Người dùng nhận được mật khẩu mới và đăng nhập thành công với mật khẩu mới |

###### Bảng 2.4 Use case quên mật khẩu

Biểu đồ trình tự:



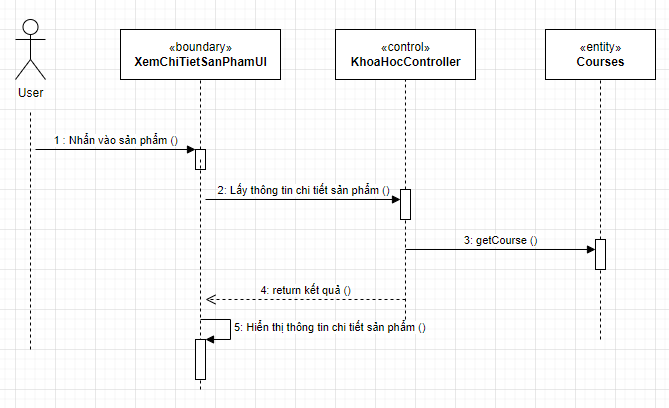
Hình 2.6: Biểu đồ trình tự quên mật khẩu

### 2.4.7 Use case 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 | User |
| Điều kiện | * Truy cập vào trang chi tiết khóa học |
| Luồng sự kiện chính | * Use case này bắt đầu khi User nhấn vào một khóa học trên danh sách khóa học. Hệ thống sẽ hiển thị nội dung cơ bản của khóa học, mô tả, thông tin người tạo khóa học, xem số người đã mua khóa học, giá, đánh giá của khóa học. * Khi người dùng đã mua khóa học thì có thể tất cả nội dung khóa học gồm bài giảng, bài tập, video, tài liệu. |
| 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 |

###### Bảng 2.7 : Use case xem chi tiết khóa học

Biểu đồ trình tự:



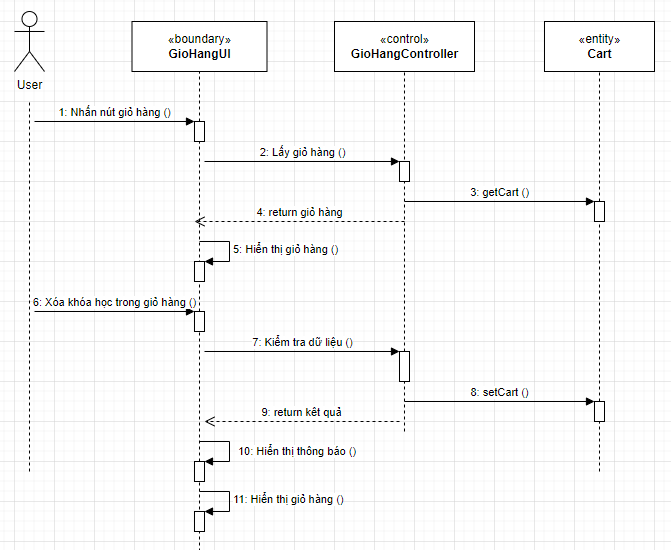
Hình 2.7: Biểu đồ trình tự xem chi tiết sản phẩm

2.4.6 Use case quản lý 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 | User |
| Điều kiện | Đã chọn sản phẩm vào trong giỏ hàng (ít nhất 1 sản phẩm) |
| Luồng sự kiện chính | * Use case bắt đầu khi khách hàng nhấn vào nút “Giỏ hàng”. * Hệ thống lấy thông tin từ bảng “Cart” hiển thị danh sách những mặt hàng đã được thêm vào giỏ hàng trước đó. * Khách hàng có thể thực hiện xóa sản phẩm ra khỏi giỏ hàng. * Hệ thống cập nhật lại giỏ hàng nếu nếu có sự thay đổi của người dùng. Use case kết thúc |
| Luồng sự kiện phụ | * Nếu trong giỏ hàng chưa có sản phẩm nào được chọn thì hệ thống sẽ hiển thị thông báo chưa có khóa học nào trong giỏ hàng. Use case kết thúc. |
| Điều kiện sau | Thêm khóa học vào giỏ hàng thành công |

###### Bảng 2.6: Use case quản lý giỏ hàng

Biểu đồ trình tự:



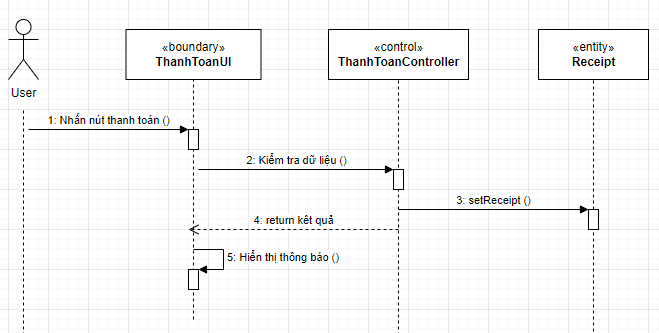
Hình 2.8: Biểu đồ trình tự quản lý giỏ hàng

2.4.7 Use case thanh toán khóa học

|  |  |
| --- | --- |
| 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 | User |
| Điều kiện | Đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi người dùng chọn nút “Thanh toán” trong màn hình giỏ hàng. Hệ thống sẽ lấy ra thông tin danh sách các khóa học có trong giỏ hàng. Thông tin người dùng. * Người dùng sẽ thanh toán trực tiếp bằng tài khoản Paypal hoặc thẻ visa. Khi thanh toán xong hệ thống sẽ hiển thị thông báo đã thanh toán thành công. Use case kết thúc |
| Luồng sự kiện phụ | * Nếu thanh toán không thành công thì hệ thống sẽ hiển thị thông báo thanh toán không thành công. Use case kết thúc. |
| Điều kiện sau | Đăng ký thành công và có thể đăng nhập vào hệ thống |

###### Bảng 2.7: Use case thanh toán hóa học

Biểu đồ trình tự:



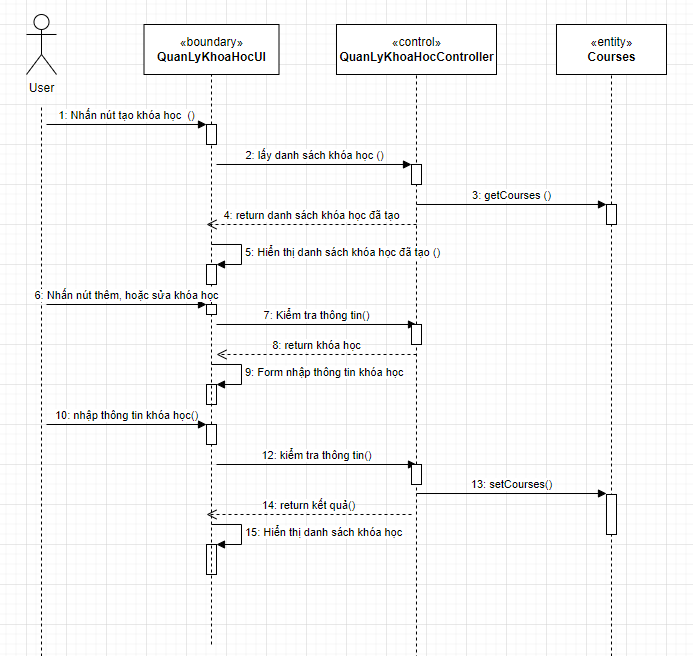
Hình 2.9: Biểu đồ trình tự thanh toán

### Use case quản lý 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 | User |
| Điều kiện | Đăng nhập thành công vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi người dùng nhấn vào nút “Tạo khóa học” trên màn hình. Hệ thống sẽ hiển thị danh sách các khóa học trong bảng “CreateCourse”. Mỗi khóa học sẽ có nút sửa. Đồng thời có nút thêm khóa học mới. * Người dùng chọn hành động cần thực hiện (thêm, sửa khóa học). Hệ thống sẽ hiển thị ra màn hình gồm các trường chứa thông tin của khóa học để người dùng thực hiện thao tác cần thiết. * Khi người dùng nhấn nút “Sửa khóa học” hoặc “Tọa khóa học” . Hệ thống sẽ kiểm tra tính hợp lệ của các trường tương ứng và lưu vào bảng “CreateCourse” trong cơ sở dữ liệu. Hệ thống sẽ hiển thị thông báo thành công. Use case kết thúc. |
| Luồng sự kiện phụ | * Nếu nhập không hợp lệ các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, người dùng nhập lại trường đó. |
| Điều kiện sau | Thêm mới khóa học thành công |

###### Bảng 2.8: Use case quản lý khóa học

Biểu đồ trình tự:



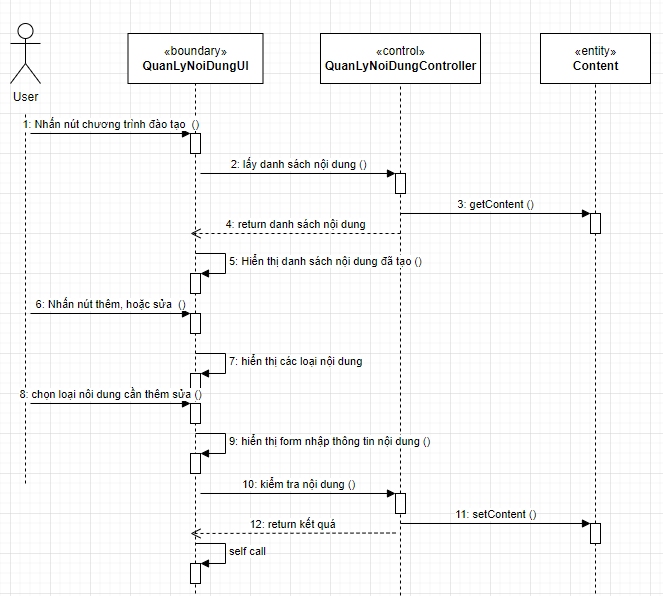
Hình 2.10: Biểu đồ trình tự quản lý khóa học

### 2.4.9 Use case quản lý bài tập, bài giảng

|  |  |
| --- | --- |
| Mục đích | Người dùng tạo nội dung bài tập, bài giảng cho khóa học |
| Mô tả | Người dùng tạo nội dung bài tập, bài giảng cho khóa học |
| Tác nhân | User |
| Điều kiện | Đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi người dùng nhấn vào nút chương trình giảng dạy trên giao diện tạo khóa học. * Hệ thống sẽ chuyển sang trang mới và lấy dữ liệu từ bảng “Courses”. Hệ thống sẽ hiển thị danh sách các bài giảng, bài tập của khóa học mỗi sản phẩm có nút sửa, xóa, di chuyển thứ tự khóa học, đồng thời cả nút thêm nội dung bài giảng. * Người dùng có thể chọn hành đong cần thực hiện (Thêm, sửa, xóa nội dung khóa học). Hệ thống hiện thị các mục lựa chọn cho người dùng chọn tạo bài tập trắc nghiệm, bài tập coding, bài giảng thông thường và hiển thị các trường thông tin tương ứng với mỗi loại nội dung cần thêm hoặc sửa. * Người dùng nhập thông tin vào các trường tương ứng và nhấn vào nội dung để sửa (hoặc thêm). Hệ thống sẽ kiểm tra các trường thông tin nếu hợp lệ sẽ lưu vào cơ sở dữ liệu. Use case kết thúc |
| Luồng sự kiện phụ |  |
| Điều kiện sau | * Tạo, sửa xóa thành công nội dung khóa học |

###### Bảng 2.9: Use case quản lý bài tập, bài giảng

Biểu đồ trình tự:



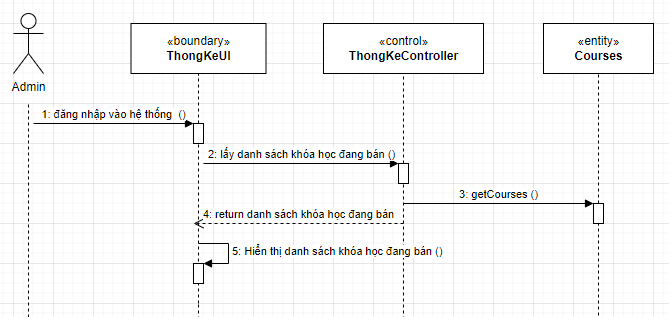
Hình 2.11: Biểu đồ trình tự quản lý bài tập bài giảng

### 2.4.10 Use case thống kê khóa học

|  |  |
| --- | --- |
| Mục đích | Admin thống kê khóa học lượt đánh giá người tham gia khóa học |
| Mô tả | Admin thống kê khóa học lượt đánh giá người tham gia khóa học |
| Tác nhân | Admin |
| Điều kiện | Đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi Admin đăng nhập vào hệ thống. Hệ thống sẽ lấy ra thông tin danh sách các khóa học đang có trong hệ thống và các khóa học đang được quan tâm nhất |
| Luồng sự kiện phụ | * Khi không lấy được cơ sở dữ liệu hệ thống sẽ báo lỗi. |
| Điều kiện sau | * Hiển thị thành công danh sách khóa học |

###### Bảng 2.10: Use case thống kê khóa học

Biểu đồ trình tự:



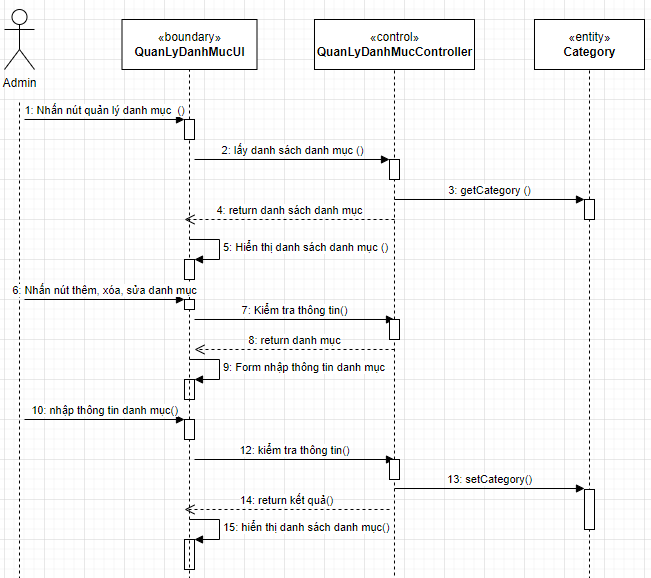
Hình 2.12: Biểu đồ trình tự thống kê khóa học

### 2.4.11 Use case quản lý danh mục

|  |  |
| --- | --- |
| Mục đích | Admin có thể thực hiện chức năng thêm sửa xóa danh mục của khóa học |
| Mô tả | Admin có thể thực hiện chức năng thêm sửa xóa danh mục của khóa học |
| Tác nhân | Admin |
| Điều kiện | Admin đăng nhập vào hệ thống |
| Luồng sự kiện chính | * Use case này bắt đầu khi người dùng nhấn vào nút quản lý danh mục. * Hệ thống sẽ chuyển sang trang mới và lấy dữ liệu từ bảng “Categogy”. Hệ thống sẽ hiển thị danh sách các danh mục đang có trên hệ thống, và có nút thêm để có thể thêm danh mục. * Người dùng có thể chọn hành động cần thực hiện (Thêm, sửa, xóa danh mục ).   Người dùng nhập thông tin vào các trường tương ứng và nhấn vào nội dung để sửa (hoặc thêm). Hệ thống sẽ kiểm tra các trường thông tin nếu hợp lệ sẽ lưu vào cơ sở dữ liệu. Use case kết thúc |
| Luồng sự kiện phụ | * Nếu đã tồn tại danh mục hệ thống sẽ hiển thị thông báo đã tồn tại danh mục. use case kết thúc |
| Điều kiện sau | * Tạo, sửa, xóa thành công danh mục |

###### Bảng 2.11: Use case quản lý danh mục

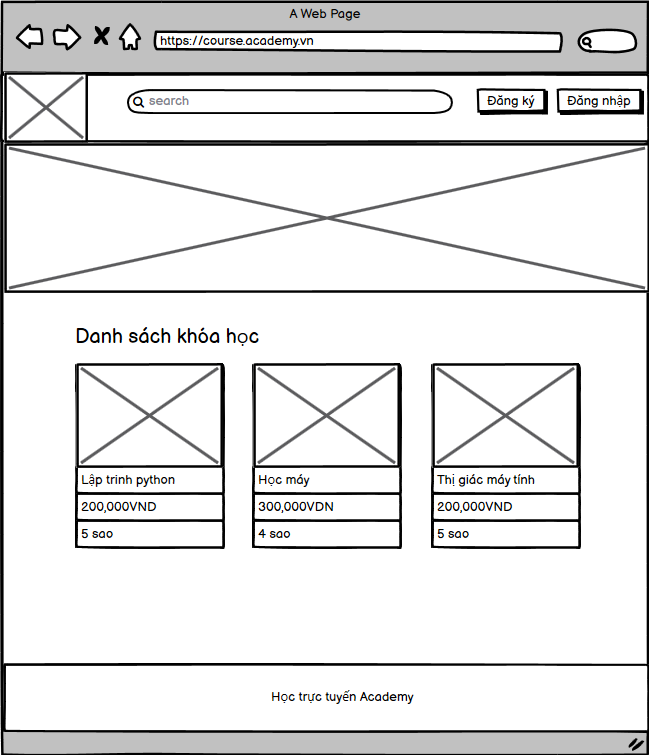
Biểu đồ trình tự



Hình 2.13 Biểu đồ trình tự quản lý danh mục

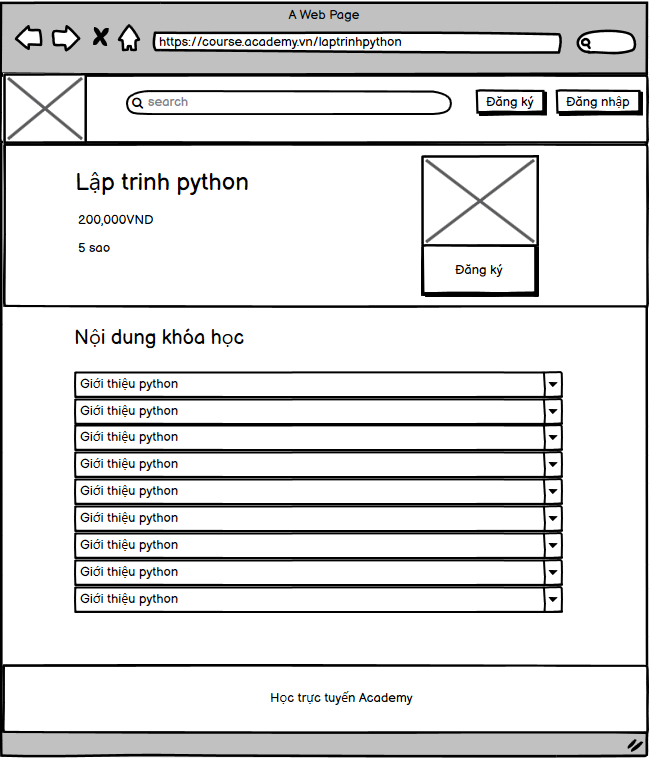
## Thiết kế giao diện

### 2.5.1 Màn hình trang chủ



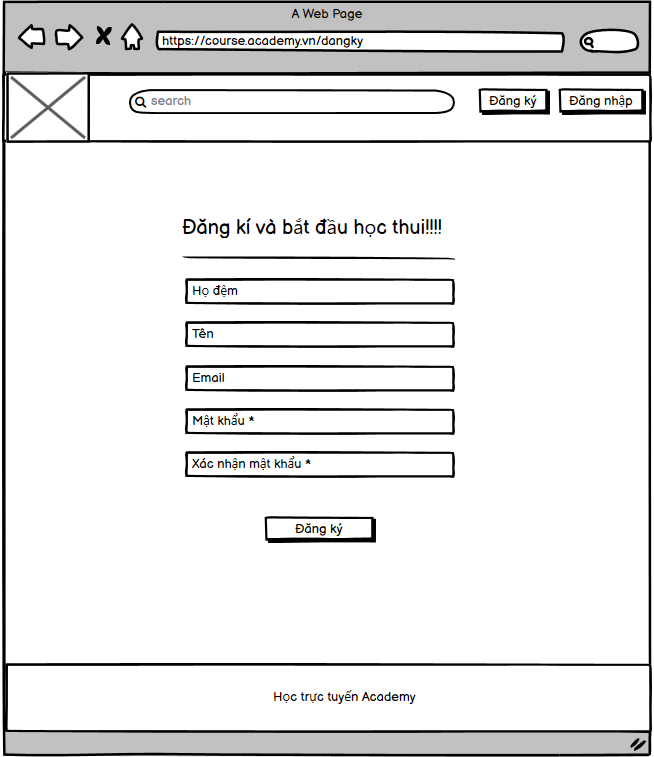
Hình 2.14: Màn hình trang chủ

### 2.5.2 Màn hình chi tiết khóa học

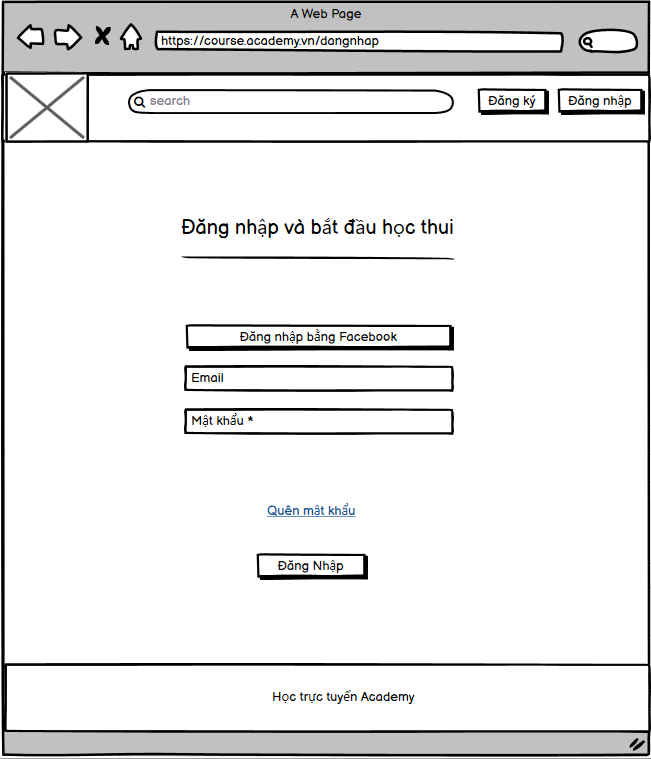


Hình 2.15: Màn hình chi tiết khóa học

### 2.5.3 Màn hình đăng ký

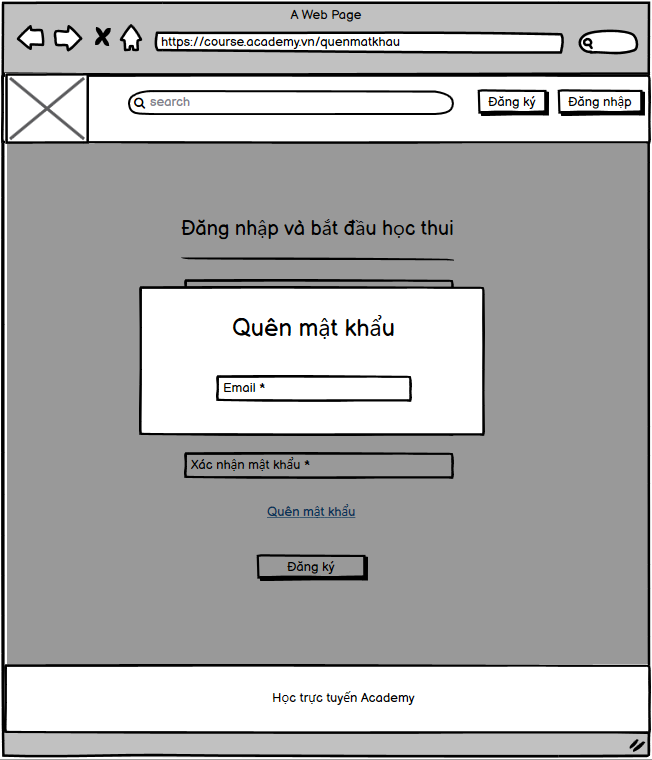


Hình 2.16: Màn hình đăng ký

2.5.4 Màn hình đăng nhập

Hình 2.17: Màn hình đăng nhập

### 2.5.5 Màn hình quên mật khẩu



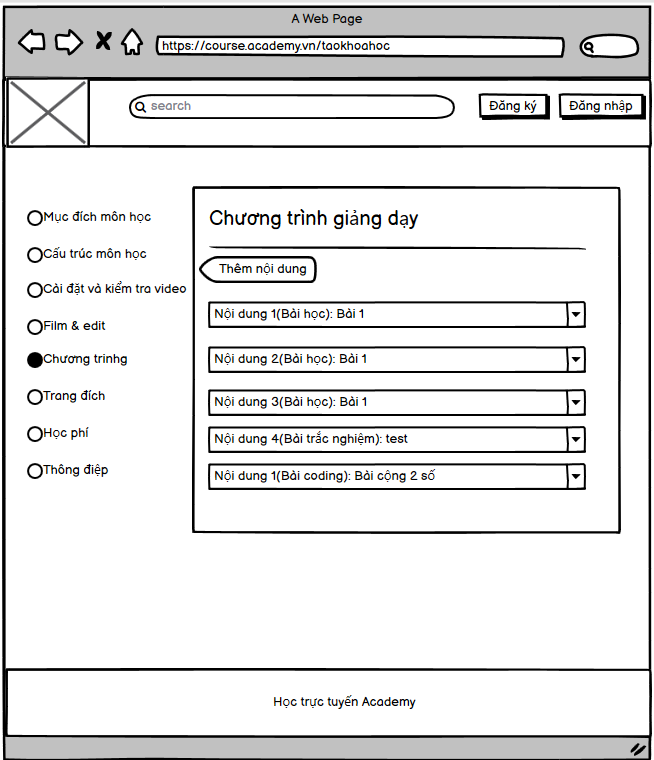
Hình 2.18: Màn hình quên mật khẩu

### 2.5.6 Màn hình quản lý khóa học



Hình 2.19 Màn hình quản lý khóa học

### 2.5.7 Màn hình quản lý bài tập, bài giảng



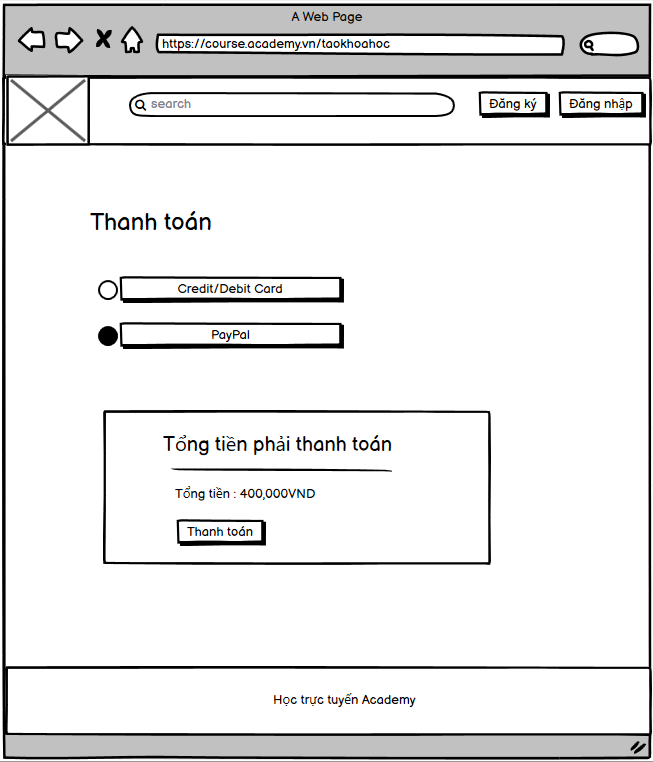
Hình 2.20: Màn hình quản lý bài tập bài giảng

### 2.5.8 Màn hình giỏ hàng

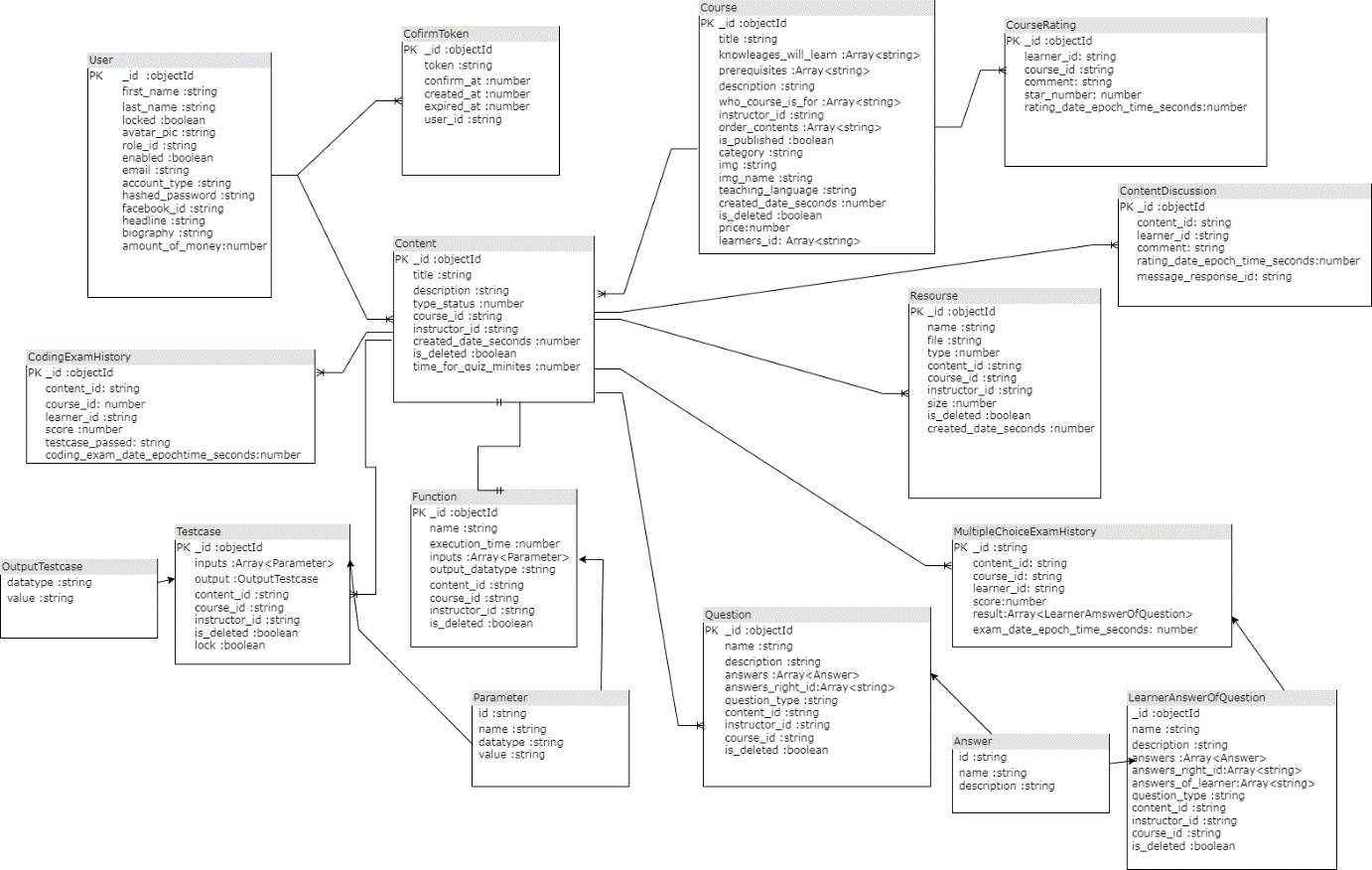


Hình 2.21: Màn hình giỏ hàng

### 2.5.9 Màn hình thanh toán

Hình 2.22: Màn hình thanh toán

## Thiết kế cơ sở dữ liệu



### 2.6.1 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 |

###### Bảng 2.12: User

### 2.6.2 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 |  |

###### Bảng 2.13: ConfirmToken

### 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 |  |

###### Bảng 2.14: Content

### 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 |

###### Bảng 2.15: Parameter

### 2.6.5 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 |  |

###### Bảng 2.16: Function

### 2.6.6 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 |

###### Bảng 2.17: OutputTestcase

### 2.6.7 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 |  |

###### Bảng 2.18: Testcase

### 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 |  |

###### Bảng 2.19: CodingExamHistory

### 2.6.9 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 |  |

###### Bảng 2.20: Course

### 2.6.10 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 |  |

###### Bảng 2.21: CourseRating

### 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 |  |

###### Bảng 2.22: ContentDiscussion

### 2.6.12 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 |  |

###### Bảng 2.23 Resourse

### 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 |  |

###### Bảng 2.24: Answer

### 2.6.14 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 |  |

###### Bảng 2.25: LeanerAnswerOfQuestion

### 2.6.15 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 |  |

###### Bảng 2.26: Question

### 2.6.16 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 |  |

###### Bảng 2.27: MultipleChoiceExamHistory

# Chương 3 - KẾT QUẢ ĐẠT ĐƯỢC

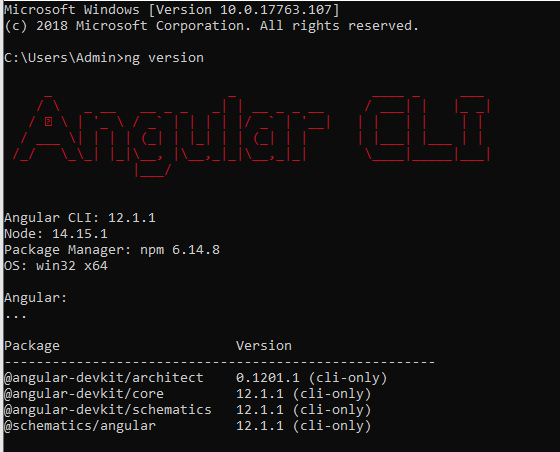
## 3.1 Cài đặt chương trình

* Tạo ứng dụng Angular CLI bằng terminal ta dùng câu lệnh



Hình 3.1: Câu lệnh cài đặt Angular CLI

* Sau khi cài đặt kiểm tra phiên bản của Angular CLI bằng câu lệnh “ng version”.



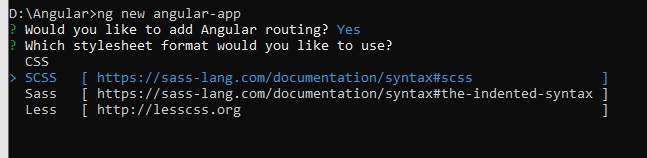
Hình 3.2: Kiểm tra phiên bản Angular CLI

* Tạo dự án mới bằng câu lệnh “ng new + tên dự án”



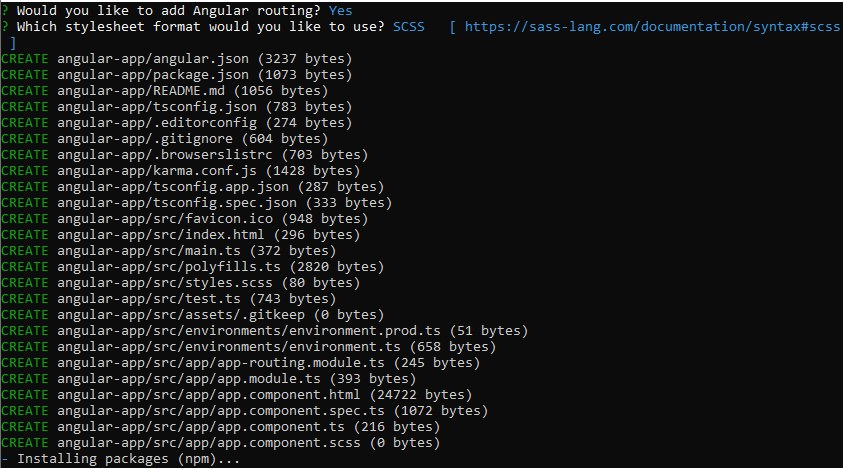
Hình 3.3: Tạo dự án Angular

* Thực hiện câu lênh “y” để đồng ý tạo routing cho dự án Angular



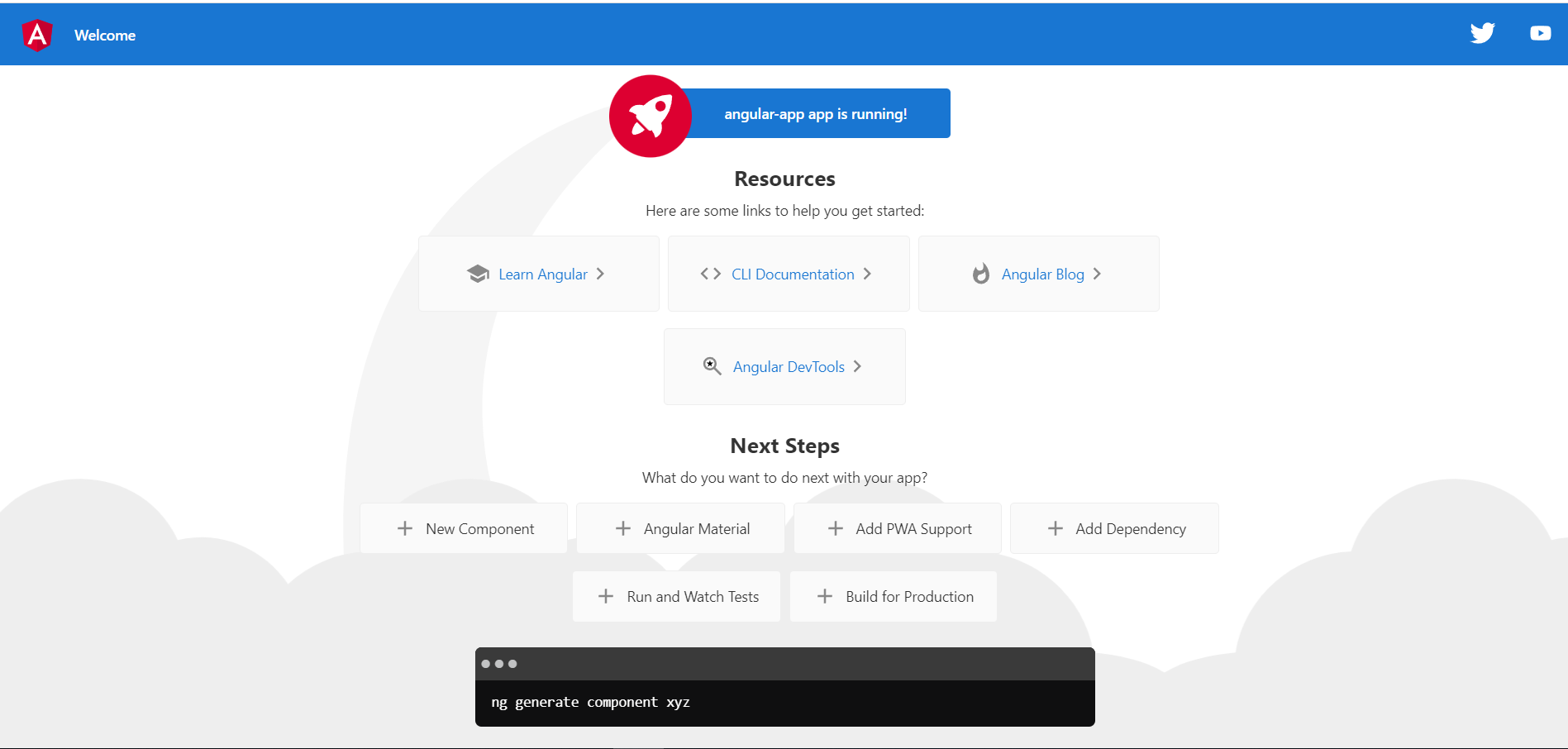
Hình 3.4: Tạo routing cho dự án

* Chọn chương trình tiền xử lý SCSS cho dự án



Hình 3.5: Chọn chương trình tiền xử lý SCSS

* Chạy dự án bằng câu lệnh “yarn start” hoặc “ng serve”



Hình 3.6: Giao diện ban đầu của dự án

* Cấu trúc của 1 dự án Angular bao gồm:
* angular.json: cấu hình CLI của dự án cho các tuỳ chọn xây dựng, chạy và kiểm thử.
* package.json: cấu hình các thư viện NPM phụ thuộc và các scripts trong dự án.
* src: thư mục gốc chứa source code dự án.
* node\_modules: chứa các thư viện NPM cho toàn dự án.
* tsconfig.json: cấu hình TypeScript cho dự án.
* tslint.json: cấu hình TSLint kiểm tra mã TypeScript.
* index.html: HTML Layout Master
* styles.scss: Global Stylesheet
* assets: chứa tài nguyên toàn bộ dự án
* main.ts: bootstrap root module
* app: thư mục này tương ứng với root module được bootstrap tại
* main.js.
* Về sau khi phát triển thêm các modules và thành phần khác sẽ nằm trong thư mục này.

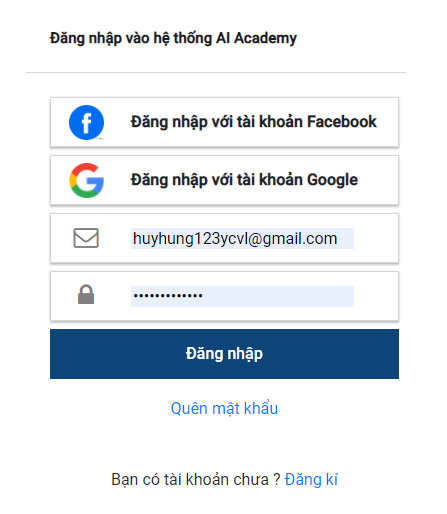
## Kết quả thu được

Hệ thống có những chức năng:

* Đăng nhập/ Đăng ký
* Quên mật khẩu
* Xem danh sách khóa học/ xem chi tiết khóa học
* Quản lý khóa học
* Quản lý giỏ hàng
* Thanh toán
* Quản lý bài tập, bài giảng
* Thống kê khóa học

## Một số giao diện của hệ thống

### Giao diện đăng nhập

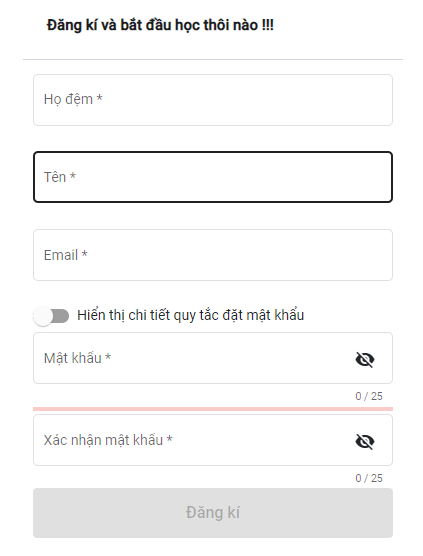


Hình 3.7: Giao diện đăng nhập

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 3.8: 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

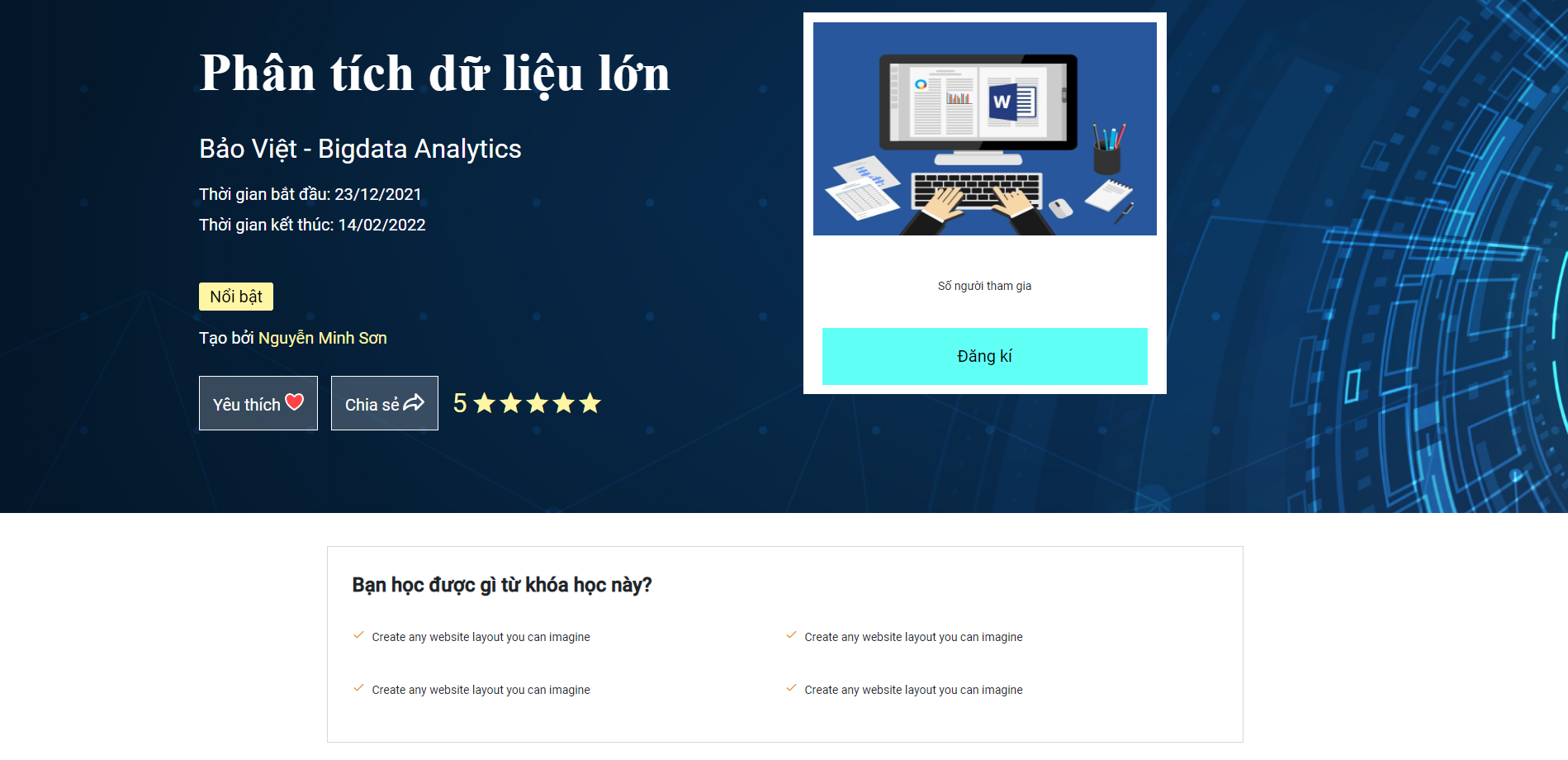


Hình 3.9: Giao diện trang chủ

Chức năng

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

### Giao diện xem chi tiết khóa học

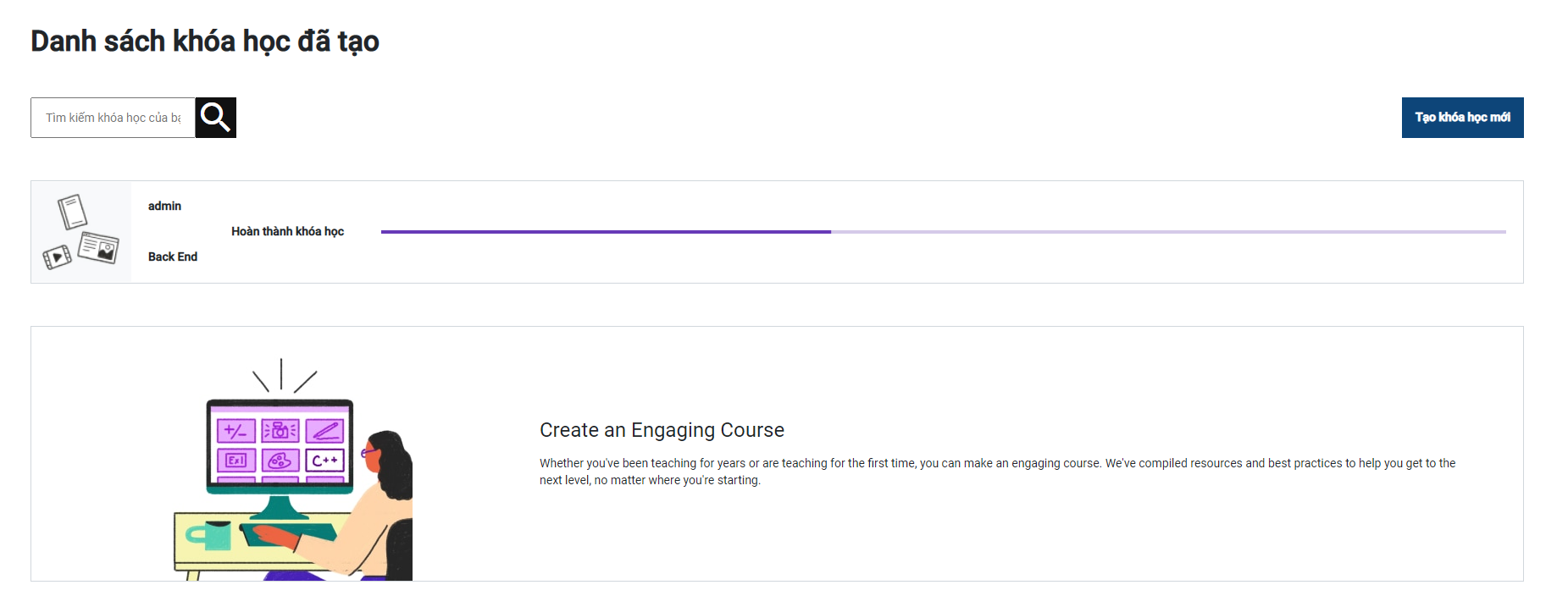


Hình 3.10: Giao diện xem chi tiết khóa học

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

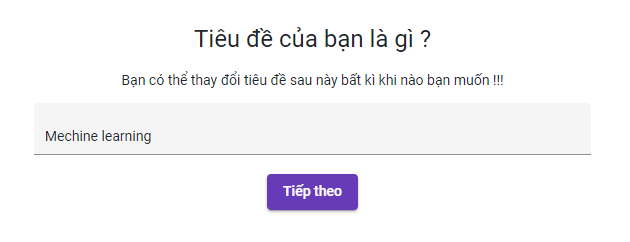
****

Hình 3.11: 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

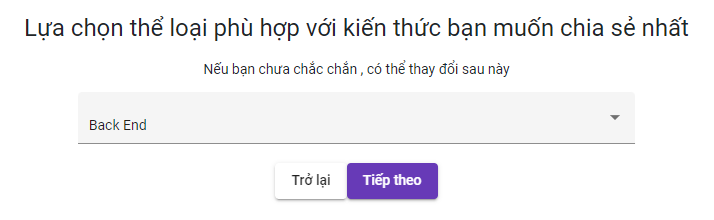


Hình 3.12: Giao diện tạo tên khóa học

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

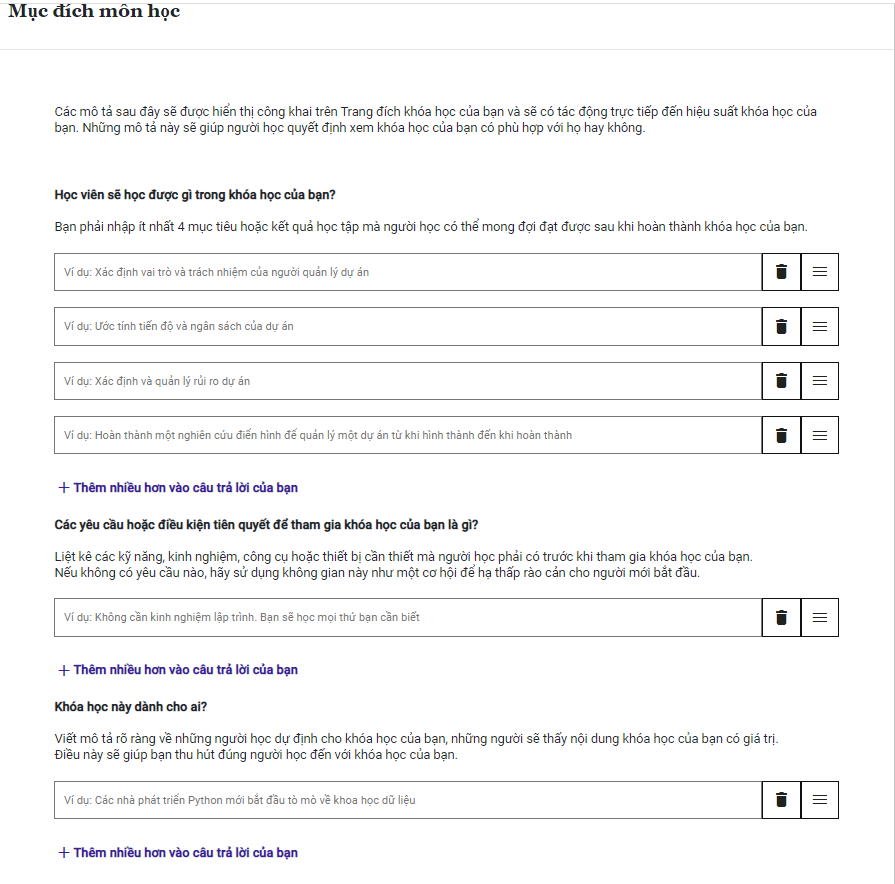


Hình 3.13: Giao diện lựa chọn thể loại khóa học

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

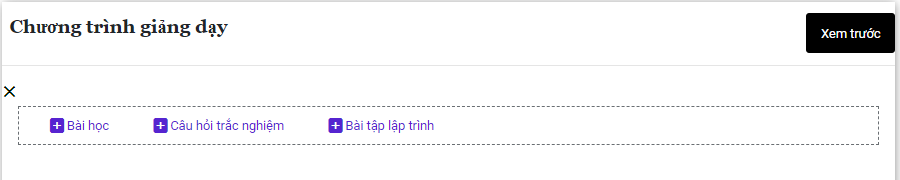


Hình 3.14: Giao diện tạo mục đích của môn học

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

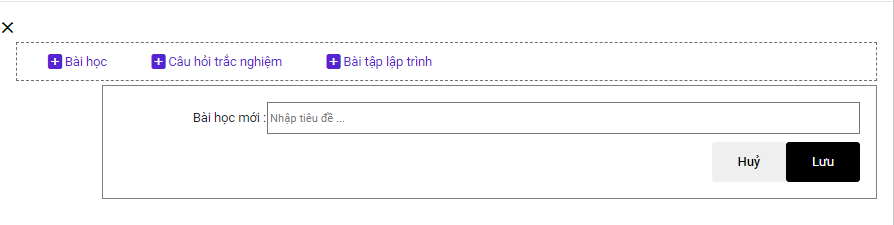
****

Hình 3.15: 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.

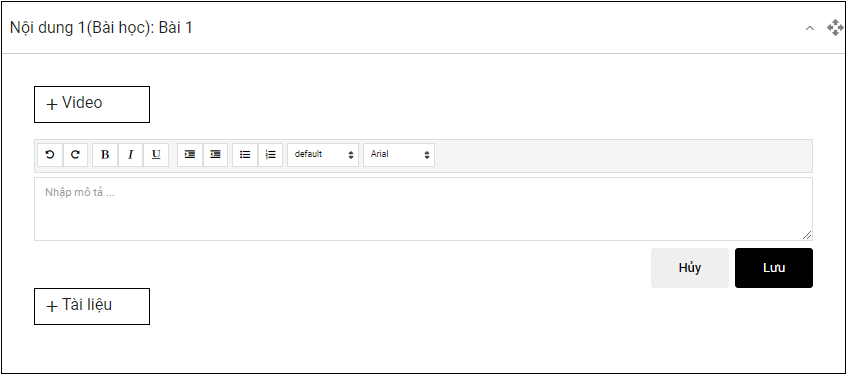
### Giao diện tạo bài học



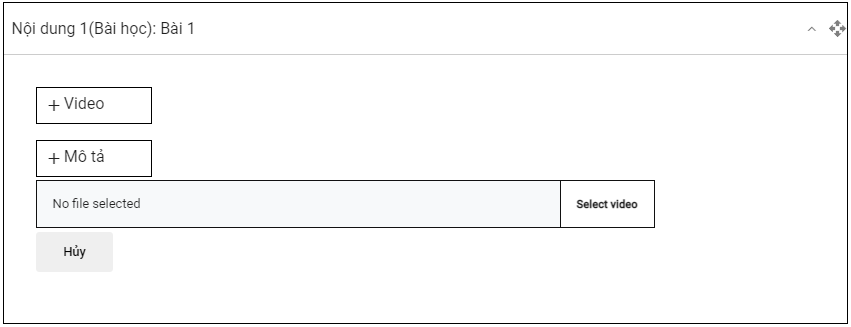
Hình 3.16: Giao diện nhập tiêu đề bài học



Hình 3.17: Giao diện thêm video cho bài học



Hình 3.18: Giao diện tạo mô tả cho bài học



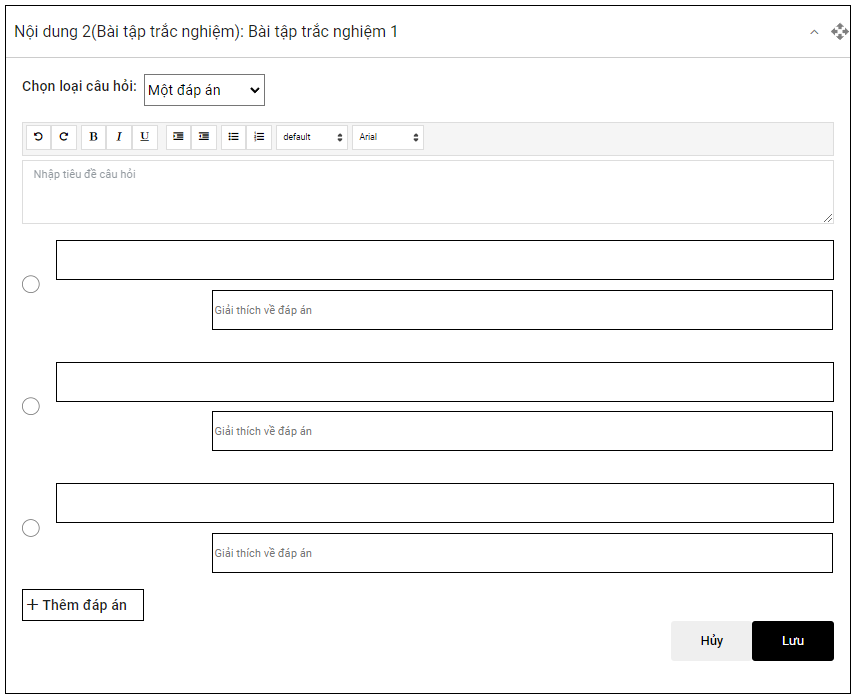
Hình 3.19: Giao diện thêm file tài liệu cho bài học

* 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 3.20: Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài



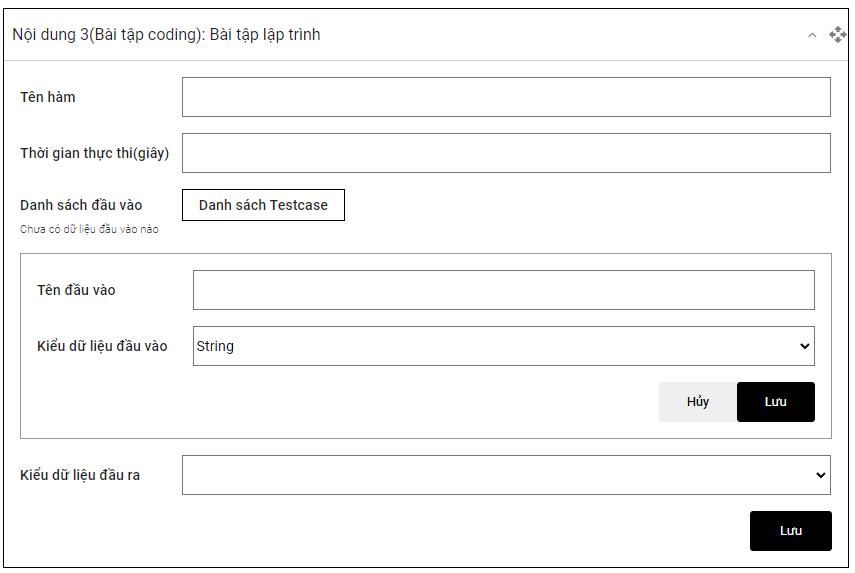
Hình 3.21: Giao diện tạo câu hỏi cho bài tập trách nghiệm

* 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



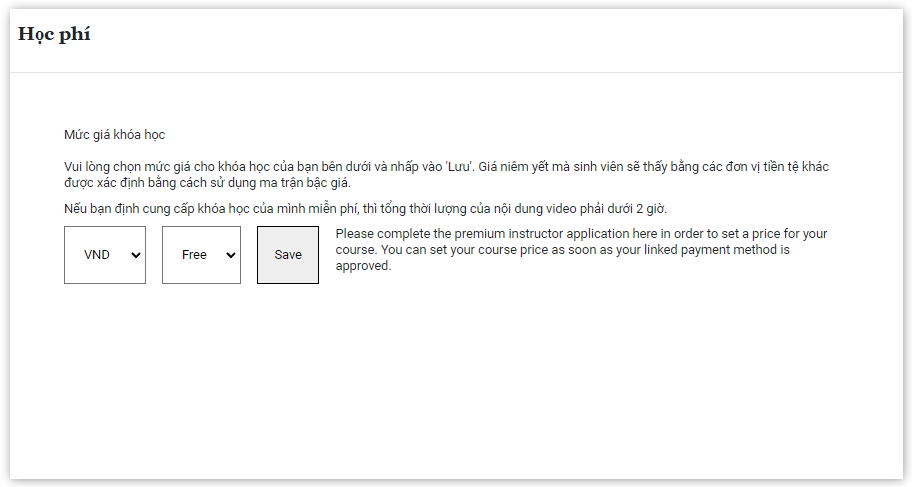
Hình 3.22: Giao diện nhập Tiêu đề, Mô tả, Thời gian làm bài



Hình 3.23: Giao 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 3.24: Giao diện tạo học phí cho khóa học

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

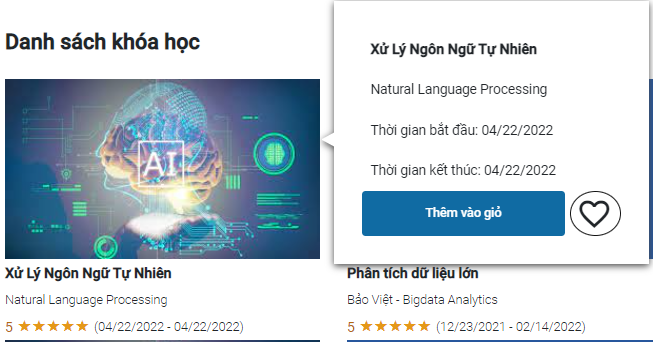
### Giao diện tạo trang đích của khóa học



Hình 3.25: 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

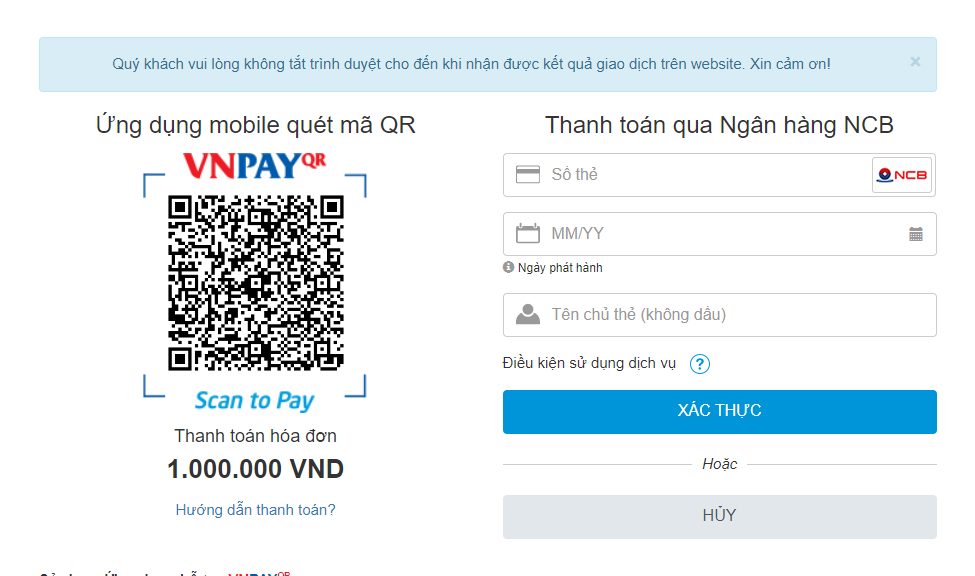


Hình 3.26: Giao diện thêm vào giỏ hàng

### Giao diện thanh toán



Hình 3.27: Giao diện giỏ hàng



Hình 3.28: Giao diện cổng thanh toán

# KẾT LUẬN

Sau thời gian nghiên cứu cùng với sự giúp đỡ của cô giáo và sự góp ý của các bạn, tôi đã hoàn thành đồ án tốt nghiệp với đề tài: ***Xây dựng hệ thống hỗ trợ học trực tuyến Academy***. 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