**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE BÁN KHÓA HỌC VĂN ANH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Nguyễn Lan Anh** |
| **Sinh viên thực hiện:** | **Kiều Văn Anh** |
| **Mã số sinh viên:** | **2020606145** |
| **Khóa:** | **15** |
| **Lớp:** | **2020DHCNTT05** |

**Hà Nội- 2024**

# MỤC LỤC

[MỤC LỤC 3](#_Toc165840280)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT 6](#_Toc165840281)

[DANH MỤC HÌNH ẢNH, ĐỒ THỊ 7](#_Toc165840282)

[DANH MỤC BẢNG BIỂU 10](#_Toc165840283)

[LỜI CẢM ƠN 11](#_Toc165840284)

[MỞ ĐẦU 12](#_Toc165840285)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 13](#_Toc165840286)

[1.1. Giới thiệu về đề tài 13](#_Toc165840287)

[1.1.1. Lý do chọn đề tài 13](#_Toc165840288)

[1.1.2. Mục tiêu đề tài 13](#_Toc165840289)

[1.1.3. Đối tượng nghiên cứu 14](#_Toc165840290)

[1.1.4. Phạm vi nghiên cứu 15](#_Toc165840291)

[1.1.5. Kết quả mong muốn 16](#_Toc165840292)

[1.2. Tìm hiểu ngôn ngữ, framework, môi trường, công nghệ, hệ quản trị cơ sở dữ liệu 17](#_Toc165840293)

[1.2.1. Ngôn ngữ 17](#_Toc165840294)

[1.2.2. Framework 21](#_Toc165840295)

[1.2.3. Môi trường 23](#_Toc165840296)

[1.2.4. Công nghệ RESTful API 24](#_Toc165840297)

[1.2.5. Hệ quản trị cơ sở dữ liệu 25](#_Toc165840298)

[CHƯƠNG 2: TÀI LIỆU ĐẶC TẢ PHẦN MỀM 27](#_Toc165840299)

[2.1. Khảo sát hệ thống, xác định yêu cầu phần mềm 27](#_Toc165840300)

[2.1.1. Khảo sát yêu cầu người dùng 27](#_Toc165840301)

[2.1.2. Khảo sát một số trung tâm, giảng viên 28](#_Toc165840302)

[2.1.3. Mô tả trang web 28](#_Toc165840303)

[2.1.4. Các yêu cầu chức năng 28](#_Toc165840304)

[2.1.5. Các yêu cầu phi chức năng 30](#_Toc165840305)

[2.2. Phân tích yêu cầu website 30](#_Toc165840306)

[2.2.1. Biểu đồ use case 31](#_Toc165840307)

[2.2.2. Mô tả use case 31](#_Toc165840308)

[2.2.3. Biểu đồ tuần tự 31](#_Toc165840309)

[2.2.4. Biểu đồ quan hệ 31](#_Toc165840310)

[CHƯƠNG 3: THIẾT KẾ PHẦN MỀM 31](#_Toc165840311)

[3.1. Giao diện khách 31](#_Toc165840312)

[3.2. Giao diện trang Admin 31](#_Toc165840313)

[3.3. Giao diện trang Student 31](#_Toc165840314)

[3.4. Giao diện trang Tutor 31](#_Toc165840315)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 31](#_Toc165840316)

[4.1. API 31](#_Toc165840317)

[4.1.1. Cấu trúc source code 31](#_Toc165840318)

[4.1.2. Tạo CSDL bằng Code first với Migrations 33](#_Toc165840319)

[4.1.3. Viết API 34](#_Toc165840320)

[4.1.4. Demo kết quả API 36](#_Toc165840321)

[4.2. Trang front end 40](#_Toc165840322)

[4.3. Kiểm thử phần mềm 42](#_Toc165840323)

[CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ 43](#_Toc165840324)

[5.1. Kết quả đạt được 43](#_Toc165840325)

[5.2. Khó khăn gặp phải 43](#_Toc165840326)

[5.3. Kiến nghị 44](#_Toc165840327)

[TÀI LIỆU THAM KHẢO 46](#_Toc165840328)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt, ký hiệu** | **Giải thích** |
| HTML | Hyper Text Markup Language (Ngôn ngữ đánh dấu siêu văn bản) |
| CSS | Cascading Style Sheets |
| CSDL | Cơ sở dữ liệu |
| DOM | Document Object Model |
| LINQ | Language Integrated Query (Ngôn ngữ truy vấn tích hợp) |
| MSSQL Server | Microsoft SQL Server |
| RDBMS | Relational Database Management System |
| 3NF | Third Normal Form |
| AI | Artificial Intelligence |

# DANH MỤC HÌNH ẢNH, ĐỒ THỊ

[Hình 1.1. Cách thức thiết kế theo mô hình RESTfull API 24](#_Toc165840374)

[Hình 2.1. Biểu đồ use case tổng quát 31](#_Toc165840375)

[Hình 2.2. Biểu đồ use case front end 31](#_Toc165840376)

[Hình 2.3. Biểu đồ use case front end hướng học viên 31](#_Toc165840377)

[Hình 2.4. Biểu đồ use case front end hướng giảng viên 31](#_Toc165840378)

[Hình 2.5. Biểu đồ use case back end 31](#_Toc165840379)

[Hình 2.6. Sơ đồ CSDL tổng quát 31](#_Toc165840380)

[Hình 2.7. Bảng \_EFMigrationsHistory 31](#_Toc165840381)

[Hình 2.8. Bảng Accounts 31](#_Toc165840382)

[Hình 2.9. Bảng Answers 31](#_Toc165840383)

[Hình 2.10. Bảng CourseParts 31](#_Toc165840384)

[Hình 2.11. Bảng Courses 31](#_Toc165840385)

[Hình 2.12. Bảng Decentralizations 31](#_Toc165840386)

[Hình 2.13. Bảng Enrollments 31](#_Toc165840387)

[Hình 2.14. Bảng Exams 31](#_Toc165840388)

[Hình 2.15. Bảng ExamTypes 31](#_Toc165840389)

[Hình 2.16. Bảng Fees 31](#_Toc165840390)

[Hình 2.17. Bảng Lectures 31](#_Toc165840391)

[Hình 2.18. Bảng PaymentHistorys 31](#_Toc165840392)

[Hình 2.19. Bảng PaymentTypes 31](#_Toc165840393)

[Hình 2.20. Bảng Questions 31](#_Toc165840394)

[Hình 2.21. Bảng StatusTypes 31](#_Toc165840395)

[Hình 2.22. Bảng Students 31](#_Toc165840396)

[Hình 2.23. Bảng Submissions 31](#_Toc165840397)

[Hình 2.24. Bảng TutorAssignments 31](#_Toc165840398)

[Hình 2.25. Bảng Tutors 31](#_Toc165840399)

[Hình 2.26. Bảng VerifyCodes 31](#_Toc165840400)

[Hình 3.1. Giao diện trang Đăng nhập 32](#_Toc165840401)

[Hình 3.2. Giao diện trang Đăng ký 32](#_Toc165840402)

[Hình 3.3. Giao diện trang Quên mật khẩu 32](#_Toc165840403)

[Hình 3.4. Giao diện trang chủ Admin 32](#_Toc165840404)

[Hình 3.5. Giao diện trang Quản lý tài khoản 32](#_Toc165840405)

[Hình 3.6. Giao diện trang Quản lý học viên 32](#_Toc165840406)

[Hình 3.7. Giao diện trang Chi tiết học viên 32](#_Toc165840407)

[Hình 3.8. Giao diện Thêm tiền 32](#_Toc165840408)

[Hình 3.9. Giao diện trang Quản lý giảng viên 32](#_Toc165840409)

[Hình 3.10. Giao diện trang Chi tiết giảng viên 32](#_Toc165840410)

[Hình 3.11. Giao diện trang Phân công khóa học 32](#_Toc165840411)

[Hình 3.12. Giao diện trang Xem số học viên đăng kí khóa học 32](#_Toc165840412)

[Hình 3.13. Giao diện trang Khóa học 32](#_Toc165840413)

[Hình 3.14. Giao diện trang Chi tiết khóa học 32](#_Toc165840414)

[Hình 3.15. Giao diện trang chủ Student 32](#_Toc165840415)

[Hình 3.16. Giao diện trang Khóa học 32](#_Toc165840416)

[Hình 3.17. Giao diện trang Xem chi tiết khóa học 32](#_Toc165840417)

[Hình 3.18. Giao diện Giỏ hàng 32](#_Toc165840418)

[Hình 3.19. Giao diện thanh toán 32](#_Toc165840419)

[Hình 3.20. Giao diện thanh toán thành công 32](#_Toc165840420)

[Hình 3.21. Giao diện trang Xem khóa học đã mua 32](#_Toc165840421)

[Hình 3.22. Giao diện trang Vào học 32](#_Toc165840422)

[Hình 3.23. Giao diện trang Quản lý thông tin cá nhân 32](#_Toc165840423)

[Hình 3.24. Giao diện trang Đổi mật khẩu 32](#_Toc165840424)

[Hình 3.25. Giao diện trang Tutor 32](#_Toc165840425)

[Hình 3.26. Giao diện trang Phân công 32](#_Toc165840426)

[Hình 3.27. Giao diện trang Chi tiết phân công 32](#_Toc165840427)

[Hình 4.1. Cấu trúc source code phần API 32](#_Toc165840428)

[Hình 4.2. Bật Package Manager Console 34](#_Toc165840429)

[Hình 4.3. Package Manager Console 34](#_Toc165840430)

[Hình 4.4. Package Swashbuckle.AspNetCore 35](#_Toc165840431)

[Hình 4.5. Code cài đặt Swagger trong Program.cs 35](#_Toc165840432)

[Hình 4.6. Cài đặt cấu hình chạy API 36](#_Toc165840433)

[Hình 4.7. Giao diện API website 36](#_Toc165840434)

[Hình 4.8. Truy vấn lấy danh sách tài khoản 37](#_Toc165840435)

[Hình 4.9. Truy vấn lấy danh sách khóa học 37](#_Toc165840436)

[Hình 4.10. Truy vấn lấy giỏ hàng cho học viên 38](#_Toc165840437)

[Hình 4.11. Truy vấn lấy lịch sử giao dịch cho học viên 38](#_Toc165840438)

[Hình 4.12. Truy vấn lấy danh sách học viên cho Admin 39](#_Toc165840439)

[Hình 4.13. Truy vấn lấy danh sách giảng viên cho Admin 39](#_Toc165840440)

[Hình 4.14. Cấu trúc source code front end 40](#_Toc165840441)

[Hình 4.15. Cấu trúc file components 41](#_Toc165840442)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1. Bảng Danh sách use case 31](#_Toc165840443)

[Bảng 2.2. Bảng Đặc tả use case Đăng ký 31](#_Toc165840444)

[Bảng 2.3. Bảng Đặc tả use case Đăng nhập 31](#_Toc165840445)

[Bảng 2.4. Bảng Đặc tả use case Tìm kiếm tài khoản 31](#_Toc165840446)

[Bảng 2.5. Bảng Đặc tả use case Xem chi tiết khóa học 31](#_Toc165840447)

[Bảng 2.6. Bảng Đặc tả use case Thanh toán online 31](#_Toc165840448)

[Bảng 2.7. Bảng Đặc tả use case Mua khóa học 31](#_Toc165840449)

[Bảng 2.8. Bảng Đặc tả use case Vào học 31](#_Toc165840450)

[Bảng 2.9. Bảng Đặc tả use case Gửi feedback 31](#_Toc165840451)

[Bảng 2.10. Bảng Đặc tả use case Xem lịch sử thanh toán 31](#_Toc165840452)

[Bảng 2.11. Bảng Đặc tả use case Xem phân công 31](#_Toc165840453)

[Bảng 2.12. Bảng Đặc tả use case Quản lý tài khoản 31](#_Toc165840454)

[Bảng 2.13. Bảng Đặc tả use case Quản lý khóa học 31](#_Toc165840455)

[Bảng 2.14. Bảng Đặc tả use case Quản lý feedback 31](#_Toc165840456)

[Bảng 2.15. Bảng Đặc tả use case Quản lý học viên 31](#_Toc165840457)

[Bảng 2.16. Bảng Đặc tả use case Quản lý giảng viên 31](#_Toc165840458)

[Bảng 2.17. Bảng Đặc tả use case Xem thống kê 31](#_Toc165840459)

# LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được rất nhiều sự giúp đỡ, chỉ bảo cũng như những lời động viên đến từ thầy cô cũng như từ phía gia đình và bạn bè. Đầu tiên, em xin gửi lời cảm ơn chân thành đến cô Nguyễn Lan Anh giảng viên khoa Công Nghệ Thông Tin – trường Đại học Công nghiệp Hà Nội, người đã nhận đồ án của em. Không chỉ vậy, cô còn tận tình hướng dẫn chỉ bảo em trong suốt quá trình làm đồ án. Thật sự em không biết có thể nói gì hơn những lời cảm ơn cô, em chúc cô luôn thành công trong sự nghiệp cũng như trong việc giảng dạy. Em cũng được gửi lời cảm ơn đến các thầy cô trong trường Đại học Công nghiệp Hà Nội nói chung và các thầy cô khoa Công Nghệ Thông Tin nói riêng. Cảm ơn thầy cô đã dành tất cả nhiệt huyết cũng như công sức dạy dỗ. Nhờ có sự dạy dỗ tận tình đó em đã có được những kiến thức là cơ sở lý thuyết cũng như kỹ năng vững vàng tạo điều kiện, giúp đỡ em trong suốt quá trình học tập tại trường cũng như quá trình thực hiện đồ án. Cuối cùng em xin được phép cảm ơn gia đình cũng như bạn bè. Gia đình và bạn bè đã luôn quan tâm giúp đỡ, động viên trong suốt quá trình học tập và hoàn thành đồ án. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy cô để em có thể bổ sung, nâng cao kiến thức của mình. Em xin chân thành cảm ơn.

# MỞ ĐẦU

Sự bùng nổ Công nghệ thông tin (CNTT), trong những năm gần đây đã tác động đến mọi mặt của đời sống xã hội từ khoa học kỹ thuật đến kinh tế xã hội. Nhờ có tin học các hoạt động sản xuất, quản lý trở nên phong phú, thuận tiện hơn và mang lại nhiều lợi ích hơn. CNTT đã xâm nhập vào các hoạt động của con người và ngày càng khẳng định vai trò của nó. Trong những năm gần đây, công nghệ thông tin ở nước ta đã có những tiến bộ vượt bậc. Nó không còn xa lạ với mọi người và ngày càng được ứng dụng rộng rãi trong tất cả các lĩnh vực của đời sống xã hội cũng như trong việc học tập, mua bán, trao đổi. Dưới sự tác động của Covid 19, việc học tập online đã trở thành xu hướng, phát triển mạnh trong thời gian qua. Đồng thời, thương mại điện tử là một hình thức mua bán và trao đổi thông tin qua internet. Giảng viên, trung tâm dạy học có thể giới thiệu tất cả sản phẩm của mình bằng hình ảnh và thông tin trên trang web để khách hàng có thể xem, tham khảo và lựa chọn. Đây là hình thức kinh doanh tiện lợi, dễ dàng đang phát triển, sẽ có cơ hội thuận lợi lớn cho các công ty, doanh nghiệp đang trên đường phát triển tạo nên thành công và danh tiếng cho mình. Trong bất cứ lĩnh vực nào, thương mại điện tử cũng có lợi thế. Nó không những làm giảm thời gian mà còn làm giảm chi phí cho mọi vấn đề giao dịch của con người. Việc kinh doanh qua mạng không chỉ đáp ứng nhu cầu của con người mà còn giúp cho người sử dụng so sánh giá ở nhiều nơi để mua những mặt hàng tốt, giá hợp lý. Với những kiến thức đã học và thời gian thực tế, em đã lựa chọn đề tài “Xây dựng website bán khóa học Văn Anh”.

# TỔNG QUAN VỀ ĐỀ TÀI

* 1. Giới thiệu về đề tài

### Lý do chọn đề tài

Việc học tập là nhu cầu không thể thiểu đối với mỗi chúng ta. Trong thời đại công nghệ hiện nay, việc tự học đã được đề cao mạnh mẽ, càng ngày càng nhiều người có nhu cầu với việc học tập online. Nhận thấy xu thế đó cùng với việc tìm hiểu về nghiệp vụ của các trung tâm, giảng viên, việc giới thiệu khóa học đến từng cá nhân với chi phí thấp, hiệu quả cao là một vấn đề nan giải của các giảng viên, trung tâm.

Phổ biến các khóa học đến khách hàng thông qua các bảng báo giá, bài viết truyền thông có chi phí khá cao vì số lượng sản phẩm ngày một đa dạng và giá cả thay đổi liên tục và tính phổ biến không cao chưa đáp ứng được nhu cầu người dùng.

Mặt khác, khách hàng còn gặp nhiều khó khăn như chưa có nhiều thời gian rảnh, thời khóa biểu hay thay đổi, khóa học quảng cáo tràn lan, nhiều vụ việc lừa đảo xảy ra giữa trung tâm và học viên… Vì vậy, em đã làm về đề tài “Xây dựng website bán khóa học Văn Anh” để giúp cho các giảng viên, trung tâm có thể dễ dàng giới thiệu các khóa học của mình đến mọi người, học viên thì có thể dễ dàng tìm kiếm, sắp xếp học vào thời gian rảnh nhưng vẫn tiếp thu được kiến thức.

### Mục tiêu đề tài

Quảng cáo không giới hạn với chi phí thấp nhất.

Có cơ hội liên kết, hợp tác với các giảng viên, trung tâm giảng dạy hoặc các công ty, tổ chức. Nhận thông tin phản hồi nhanh của khách hàng, đối tác nhanh nhất.

Tiết kiệm chi phí, hoạt động không nghỉ 24/24 mà không cần đội ngũ giảng viên phục vụ.

Làm cho việc kinh doanh của các giảng viên, trung tâm giảng dạy, công ty, tổ chức được phát triển hơn.

Thuận tiện hơn cho người học về thời gian, đi lại, chi phí.

### Đối tượng nghiên cứu

Các đối tượng nghiên cứu:

* Người sử dụng cuối (người học): Nghiên cứu về nhu cầu, mong muốn và hành vi của người học khi sử dụng website để tìm kiếm, đăng ký và tham gia các khóa học.
* Quản trị viên website: Nghiên cứu về cách quản trị viên website quản lý và cập nhật nội dung, quản lý học viên, xử lý các giao dịch thanh toán và duy trì bảo mật cho hệ thống.
* Giảng viên và nhà phát triển nội dung: Nghiên cứu về cách giảng viên và nhà phát triển nội dung tạo ra và quản lý nội dung giảng dạy trên website, bao gồm việc tạo bài giảng, tài liệu học và bài kiểm tra.
* Nhà quảng cáo và đối tác hợp tác: Nghiên cứu về cách hợp tác với nhà quảng cáo và các đối tác khác để quảng cáo và tiếp thị các khóa học trên website.
* Chuyên gia trong lĩnh vực công nghệ thông tin và e-commerce: Nghiên cứu về các chuyên gia trong lĩnh vực công nghệ thông tin và e-commerce để tìm hiểu về các công nghệ mới, xu hướng và phương pháp tiên tiến để áp dụng vào việc xây dựng và phát triển website bán khóa học.
* Đối thủ cạnh tranh: Nghiên cứu về các website bán khóa học khác để hiểu về cơ cấu, chiến lược kinh doanh và điểm mạnh, yếu của họ để có thể cạnh tranh và phát triển website của mình một cách hiệu quả hơn.
* Nhà đầu tư và nhà tài trợ: Nghiên cứu về cách thu hút và hợp tác với nhà đầu tư và nhà tài trợ để hỗ trợ tài chính cho việc phát triển và quảng bá website bán khóa học.

### Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài “Xây dựng website bán khóa học Văn Anh” bao gồm các khía cạnh sau:

* Thiết kế giao diện và trải nghiệm người dùng: Nghiên cứu về cách thiết kế giao diện và trải nghiệm người dùng để tối ưu hóa sự thu hút và tiện lợi khi người dùng tìm kiếm và mua các khóa học trên website.
* Phát triển hệ thống quản lý nội dung (CMS): Nghiên cứu về việc xây dựng hệ thống quản lý nội dung linh hoạt và dễ dàng để quản lý các khóa học, nội dung giảng dạy, thông tin học viên, thanh toán và các tính năng khác.
* Tích hợp hệ thống thanh toán trực tuyến: Nghiên cứu về cách tích hợp các cổng thanh toán trực tuyến an toàn và thuận tiện để người dùng có thể mua khóa học một cách dễ dàng.
* Tối ưu hóa công cụ tìm kiếm (SEO): Nghiên cứu về cách tối ưu hóa website để nó có thể xuất hiện cao trong kết quả tìm kiếm của các công cụ tìm kiếm như Google, giúp tăng khả năng tiếp cận của người dùng.
* Phân tích dữ liệu và đánh giá hiệu suất: Nghiên cứu về cách thu thập và phân tích dữ liệu để đánh giá hiệu suất của website bán khóa học, từ đó cải thiện các chiến lược kinh doanh và marketing.
* Bảo mật thông tin và thanh toán: Nghiên cứu về cách đảm bảo tính bảo mật cho thông tin cá nhân của người dùng và các giao dịch thanh toán trên website.
* Xây dựng cộng đồng học viên: Nghiên cứu về cách tạo ra một cộng đồng học viên tích cực trên website, bao gồm việc tạo ra diễn đàn, hệ thống phản hồi, và các hoạt động tương tác khác.
* Tích hợp các tính năng học trực tuyến: Nghiên cứu về cách tích hợp các tính năng học trực tuyến như video học, bài tập trực tuyến, hệ thống đánh giá và phản hồi, để cung cấp một trải nghiệm học tập toàn diện cho người dùng.

### Kết quả mong muốn

Website hoạt động hiệu quả: Một website hoạt động một cách mượt mà, nhanh chóng và ổn định, cung cấp trải nghiệm người dùng tốt và thu hút được số lượng lớn người dùng.

Tăng doanh số bán hàng: Website đạt được mục tiêu tăng doanh số bán hàng thông qua việc tăng lượng truy cập, tỷ lệ chuyển đổi từ lượt xem thành giao dịch mua hàng, và tăng giá trị đơn hàng trung bình.

Tăng sự hài lòng của khách hàng: Khách hàng hài lòng với trải nghiệm mua sắm trên website, từ việc tìm kiếm thông tin, đăng ký khóa học, thanh toán đến việc tham gia và hoàn thành khóa học.

Xây dựng thương hiệu: Website giúp xây dựng và củng cố thương hiệu của doanh nghiệp trong lĩnh vực giáo dục và đào tạo trực tuyến, từ đó tạo niềm tin và lòng tin của người dùng.

Tăng cường cộng đồng học viên: Website góp phần tạo ra một cộng đồng học viên tích cực, nơi mà họ có thể trao đổi kiến thức, kinh nghiệm và hỗ trợ lẫn nhau trong quá trình học tập.

Tối ưu hóa hiệu suất hoạt động: Website được tối ưu hóa về mặt kỹ thuật, nội dung và marketing để đạt được hiệu suất hoạt động tốt nhất và giảm thiểu các chi phí không cần thiết.

Thu thập và phân tích dữ liệu: Website có cơ chế thu thập và phân tích dữ liệu một cách chính xác và có ý nghĩa, từ đó cung cấp thông tin hữu ích để điều chỉnh chiến lược kinh doanh và marketing.

Bảo mật thông tin và thanh toán: Website đảm bảo tính bảo mật cho thông tin cá nhân của người dùng và các giao dịch thanh toán, giúp tạo niềm tin và lòng tin từ phía khách hàng.

* 1. Tìm hiểu ngôn ngữ, framework, môi trường, công nghệ, hệ quản trị cơ sở dữ liệu

1. Ngôn ngữ

* HTML (Hypertext Markup Language - Ngôn ngữ đánh dấu siêu văn bản):

HTML là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng nội dung trên trang web. HTML định nghĩa cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. HTML là ngôn ngữ cơ bản nhưng rất quan trọng trong việc xây dựng một trang web, và nó thường được kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trang web đa dạng và tương tác.

Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Các phần tử HTML là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists, links, quotes và các mục khác. Các phần tử HTML được phân định bằng các tags, được viết bằng dấu ngoặc nhọn. Các tags như <img/> và <input/> giới thiệu trực tiếp nội dung vào trang. Các tags khác như <p> bao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn giải nội dung của trang.

HTML là một ngôn ngữ đánh dấu mà các trình duyệt web sử dụng để giải thích và soạn văn bản, hình ảnh và các tài liệu khác thành các trang web trực quan hoặc nghe được. Các đặc điểm mặc định cho mọi mục của đánh dấu HTML được xác định trong trình duyệt và các đặc điểm này có thể được thay đổi hoặc nâng cao bằng cách sử dụng thêm CSS của nhà thiết kế trang web. Nhiều thành phần văn bản được tìm thấy trong báo cáo kỹ thuật 1988 ISO TR 9537 Techniques for using SGML, lần lượt đề cập đến các tính năng của các ngôn ngữ định dạng văn bản ban đầu, chẳng hạn như được sử dụng bởi lệnh RUNOFF được phát triển vào đầu những năm 1960 cho hệ điều hành CTSS (Compatible Time-Sharing System): các lệnh định dạng này bắt nguồn từ các lệnh được sử dụng bởi các bộ sắp chữ để định dạng tài liệu theo cách thủ công. Tuy nhiên, khái niệm SGML về đánh dấu tổng quát dựa trên các phần tử (các phạm vi được chú thích lồng nhau với các thuộc tính) chứ không chỉ đơn thuần là các hiệu ứng in, với sự phân tách của cấu trúc và đánh dấu, HTML đã được chuyển dần theo hướng này với CSS.

Cấu trúc cơ bản của một file HTML:

<HTML>

<HEAD>

<TITLE></TITLE>

</HEAD>

<BODY>

<H1>Tiêu đề</H1>

…

</BODY>

</HTML>

* CSS (Cascading Style Sheets):

CSS là viết tắt của Cascading Style Sheets. Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn. Bạn có thể hiểu là nếu HTML là tường gạch thô thì CSS là sơn để trang trí cho tường gạch đó. Các ngôn ngữ này được giữ riêng biệt để đảm bảo các trang web được xây dựng chính xác trước khi chúng được định dạng lại.

* JavaScript:

JavaScript theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là ECMAScript 7. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta1có hỗ trợ không đầy đủ cho E4X-phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

Cùng thời điểm Netscape bắt đầu sử dụng Java trên trình duyệt Netscape, LiveScript đã được đổi tên thành JavaScript để được chú ý hơn bởi ngôn ngữ lập trình Java lúc đó đang được coi là một hiện tượng. JavaScript được bổ sung vào trình duyệt Netscape bắt đầu từ phiên bản 2.0b3 của trình duyệt này vào tháng 12 năm 1995. Trên thực tế, JavaScript không được phát triển dựa từ Java. Do đó JavaScript chỉ dựa trên các cách đặt tên của Java. Java Script gồm 2 mảng là client-server thực hiện lệnh trên máy của end-user và web-server. Sau thành công của JavaScript, Microsoft bắt đầu phát triển JScript, một ngôn ngữ có cùng ứng dụng và tương thích với JavaScript. JScript được bổ sung vào trình duyệt Internet Explorer bắt đầu từ Internet Explorer phiên bản 3.0 được phát hành tháng 8 năm 1996.

DOM (Document Object Model), một khái niệm thường được nhắc đến với JavaScript trên thực tế không phải là một phần của chuẩn ECMAScript, DOM là một chuẩn riêng biệt có liên quan chặt chẽ với XML.

Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh, ... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng JavaScript để tương tác với DOM bao gồm DHTML, Ajax và SPA.

Bên ngoài trình duyệt, JavaScript có thể được sử dụng trong tập tin PDF của Adobe Acrobat và Adobe Reader. Điều khiển Dashboard trên hệ điều hành Mac OSX phiên bản 10.4 cũng có sử dụng JavaScript. Công nghệ kịch bản linh động (active scripting) của Microsoft có hỗ trợ ngôn ngữ JScript làm một ngôn ngữ kịch bản dùng cho hệ điều hành. JScript.NET là một ngôn ngữ tương thích với CLI gần giống Jscript nhưng có thêm nhiều tính năng lập trình hướng đối tượng.

Từ khi Nodejs ra đời vào năm 2009, Javascript được biết đến nhiều hơn là một ngôn ngữ đa nền tảng khi có thể chạy trên cả môi trường máy chủ cũng như môi trường nhúng.

Mỗi ứng dụng này đều cung cấp mô hình đối tượng riêng cho phép tương tác với môi trường chủ, với phần lõi là ngôn ngữ lập trình JavaScript gần như giống nhau.

* C# (C Sharp):

C# là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Entity Frammework core-Entity Framework Core (EF Core) là một framework ánh xạ cơ sở dữ liệu đối tượng mới cho .NET. Nó hỗ trợ các truy vấn LINQ, theo dõi thay đổi, cập nhật và thay đổi cấu trúc.

EF Core hoạt động với SQL Server/ SQL Azure, SQLite, Azure Cosmos DB, MySQL, PostgreSQL và nhiều cơ sở dữ liệu khác thông qua mô hình plugin của cơ sở dữ liệu.

* LINQ (Language Integrated Query - Ngôn ngữ truy vấn tích hợp):

LINQ là cách thức truy vấn dữ liệu từ một tập hợp dữ liệu. Hôm nay, chúng ta sẽ cùng tìm hiểu về LINQ và cách thức hoạt động của LINQ nhé. Nói đến lập trình, chúng ta không thể không nhắc đến dữ liệu.

Dữ liệu ở đây có thể được bắt nguồn từ bộ nhớ, XML, cơ sở dữ liệu (Mysql, Sql Server...) ... hay bất cứ nơi lưu trữ nào khác. Bất cứ dữ liệu đó bắt nguồn từ đâu, chúng ta phải luôn luôn thực hiện việc truy vấn dữ liệu đến nó. Với mỗi nguồn dữ liệu khác nhau sẽ có những cách khác nhau để thực hiện truy vấn.

1. Framework

#### React

React là một thư viện JavaScript được sử dụng phổ biến trong việc xây dựng giao diện người dùng (UI) cho các ứng dụng web. Nó được phát triển bởi Facebook và được công bố dưới dạng mã nguồn mở.

Dưới đây là một số điểm nổi bật của React:

* + **Components (Các thành phần)**: React sử dụng các thành phần để xây dựng giao diện người dùng. Mỗi thành phần đại diện cho một phần của giao diện và có thể chứa mã HTML, CSS và JavaScript.
  + **Virtual DOM (DOM ảo)**: React sử dụng Virtual DOM để cải thiện hiệu suất của ứng dụng. Thay vì cập nhật trực tiếp DOM mỗi khi có thay đổi, React so sánh Virtual DOM với DOM thực tế và chỉ cập nhật các phần tử đã thay đổi, giúp tối ưu hóa hiệu suất và tăng trải nghiệm người dùng.
  + **One-way data binding (Ràng buộc dữ liệu một chiều)**: React thúc đẩy mô hình ràng buộc dữ liệu một chiều, điều này có nghĩa là dữ liệu chuyển từ thành phần cha đến thành phần con. Điều này giúp tạo ra mã nguồn dễ đọc và dễ hiểu, cũng như giảm thiểu rủi ro về lỗi trong ứng dụng.
  + **Reusable Components (Các thành phần có thể tái sử dụng)**: React khuyến khích việc sử dụng các thành phần có thể tái sử dụng để tối ưu hóa quản lý mã nguồn và tăng cường sự linh hoạt của ứng dụng.
  + **JSX (JavaScript XML)**: React sử dụng JSX, một cú pháp mở rộng của JavaScript, để viết mã HTML trong các thành phần. JSX kết hợp giữa JavaScript và HTML, giúp tạo ra mã nguồn dễ đọc và dễ hiểu hơn.
  + **Ecosystem (Hệ sinh thái)**: React có một hệ sinh thái phong phú của các thư viện và công cụ bổ sung, như React Router, Redux, và Material-UI, giúp phát triển ứng dụng một cách linh hoạt và nhanh chóng.

#### Tailwind CSS

Tailwind CSS là một thư viện CSS được thiết kế để giúp bạn xây dựng giao diện người dùng (UI) cho các ứng dụng web một cách nhanh chóng và hiệu quả. Khác với các framework CSS truyền thống như Bootstrap, Tailwind không cung cấp các thành phần UI sẵn có mà thay vào đó tập trung vào việc cung cấp một bộ các lớp CSS tái sử dụng để định dạng giao diện.

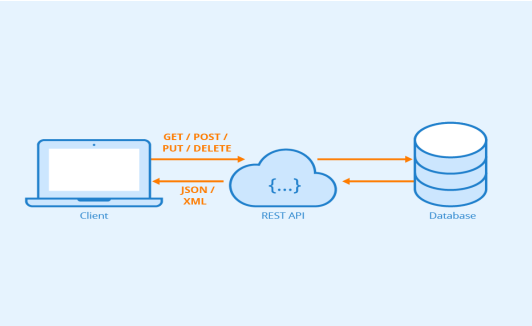
Dưới đây là một số điểm nổi bật của Tailwind CSS:

* **Reusable CSS class (Lớp CSS tái sử dụng)**: Tailwind cung cấp một bộ các lớp CSS có tên dễ nhớ và có ý nghĩa, giúp bạn áp dụng kiểu dáng và định dạng cho các phần tử HTML một cách nhanh chóng. Ví dụ, bạn có thể sử dụng lớp “**bg-blue-500**” để đặt màu nền cho một phần tử thành màu xanh dương.
* **Flexibitily (Linh hoạt)**: T ailwind không đặt ra các quy tắc cứng nhắc về cách bạn nên xây dựng giao diện. Thay vào đó, nó cung cấp các công cụ để bạn tạo ra các kiểu dáng và giao diện một cách linh hoạt theo ý muốn của bạn.
* **Customization (Tùy chỉnh)**: Tailwind cho phép bạn tùy chỉnh các cài đặt của nó bằng cách sử dụng file cấu hình, từ việc thay đổi màu sắc, phông chữ đến việc thêm hoặc loại bỏ các lớp CSS.
* **Utility-first approach (Tiếp cận tiện ích trước)**: Tailwind sử dụng tiếp cận “tiện ích trước” bằng cách cung cấp các lớp CSS cho mỗi thuộc tính CSS khác nhau. Điều này cho phép bạn tạo ra các kiểu dáng tùy chỉnh mà không cần phải viết CSS tùy chỉnh.
* **Library (Thư viện mở rộng)**: Tailwind có một cộng đồng lớn và sôi động, với nhiều plugin và công cụ mở rộng được phát triển bởi cộng đồng, giúp bạn mở rộng và tùy chỉnh Tailwind theo nhu cầu của dự án của mình.

1. Môi trường

Node.js là một môi trường chạy mã JavaScript trên máy chủ (server-side) được xây dựng trên nền tảng JavaScript Engine của Chrome, được gọi là V8 Engine. Node.js cho phép bạn thực thi mã JavaScript ở phía máy chủ, không chỉ ở phía trình duyệt web. Điều này mở ra khả năng phát triển ứng dụng web đa năng và hiệu quả.

1. Công nghệ RESTful API



Hình 1.1. Cách thức thiết kế theo mô hình RESTfull API

API (Application Programming Interface) 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 (REpresentational State Transfer) 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. 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.

Lý do chọn thiết kế theo mô hình RestfullAPI:

• Tiếp cận với xu thế phát triển phần mềm.

• Có hướng phát triển ứng dụng đa nền tảng web và mobile app.

1. Hệ quản trị cơ sở dữ liệu

MSSQL Server (Microsoft SQL Server) là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (TransactSQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server.

Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. SQL Server được phát triển và tiếp thị bởi Microsoft.

Dưới đây là một số điểm nổi bật về Microsoft SQL Server:

* **Các tính năng RDBMS**: SQL Server cung cấp các tính năng cơ bản của một hệ quản trị cơ sở dữ liệu quan hệ, bao gồm quản lý dữ liệu, truy vấn, chỉnh sửa, xóa và cập nhật dữ liệu, cũng như quản lý quyền truy cập.
* **Tính an toàn cao**: SQL Server cung cấp các tính năng an toàn để bảo vệ dữ liệu, bao gồm mã hóa dữ liệu, kiểm soát truy cập, và quản lý bảo mật dựa trên vai trò.
* **Hiệu suất cao**: SQL Server được tối ưu hóa để đảm bảo hiệu suất cao cho các ứng dụng và dịch vụ yêu cầu xử lý dữ liệu lớn và truy vấn phức tạp.
* **Tích hợp với hệ thống Microsoft**: SQL Server tích hợp tốt với các công nghệ khác của Microsoft như .NET Framework, Azure, Power BI và SharePoint, tạo ra một hệ sinh thái mạnh mẽ cho việc phát triển và triển khai ứng dụng.
* **Hỗ trợ quy mô lớn**: SQL Server có thể mở rộng để xử lý các tải công việc lớn và có khả năng quản lý cơ sở dữ liệu có kích thước lớn.
* **Các phiên bản và phiên bản miễn phí**: SQL Server có các phiên bản phong phú từ phiên bản cơ bản đến các phiên bản cao cấp hơn như Enterprise và Datacenter. Ngoài ra, Microsoft cũng cung cấp phiên bản SQL Server Express miễn phí cho các dự án và ứng dụng nhỏ.

# TÀI LIỆU ĐẶC TẢ PHẦN MỀM

## Khảo sát hệ thống, xác định yêu cầu phần mềm

### Khảo sát yêu cầu người dùng

- Web cần chọn các hình ảnh hấp dẫn, thu hút khách hàng. Web cần ít chữ và nhiều hình ảnh.

- Nên có sologan ngắn để tạo điểm nhấn cho trang web để tạo điểm nhấn cho khác hàng và người dùng

- Thiết kế không được quá sát nhau, tránh gây mỏi mắt khi tìm chi tiết cũng như cỡ chữ phông chữ phải phù hợp với kích cỡ màn hình tránh gây mỏi mắt khi chữ nhỏ hay cảm giác lệch lạc khi cỡ chữ quá to, không phù hợp khung hình.

- Có hiệu ứng animation để tạo hiệu ứng sinh động cho trang web để khi vừa truy cập vào trang web thì người dùng sẽ cảm thấy hứng thú, không thấy nhàm chán để không rời website sớm. Kết hợp với sologan để tạo trải nhiệm tốt cho người dùng

- Phân loại khóa học theo giá, danh mục để dễ tìm kiếm.

- Nên có banner quảng cáo cho website cũng như các trung tâm ngăn cách giữa các phần của website để làm ngắt quãng thông tin đỡ gây nhàm chán trong quá trình tìm tin tức.

- Trang chủ nên để tầm 3 khóa học 1 hàng để tránh làm tràn thông tin quá nhiều còn toàn bộ khóa học thì để ở 1 trang riêng để khách hàng nào cần thiết thì sẽ vào và tìm hiểu chi tiết từng khóa học. Khóa học cũng nên để những khóa học hot, khóa học mới lên đầu và ở trang chủ, các khóa học cũ thì sắp xếp dưới dần.

- Có form điền thông tin hiện ra ở cuối trang web để website có thể lấy thông tin khách hàng trong quá trình mua hàng để có thể tư vấn cũng như quảng cáo khóa học hay thông báo về các sự kiện của website.

- Có lối tắt để liên kết đến các trang mạng xã hội khác của website như youtube, facebook, instagram, pinterest,… để dễ sử dụng.

- Trang mua hàng phải trực quan và dễ hiểu để người dùng có thể dễ dàng sử dụng.

- Khóa học phải có nhiều ảnh demo giới thiệu để có thể nhìn được nhiều góc độ khác nhau.

### Khảo sát một số trung tâm, giảng viên

- Trang quản lý thiết kế đơn giản, dễ sử dụng.

- Các chức năng tương ứng với các use case được thiết kế thành Menu hiển thị ở thanh header.

### Mô tả trang web

Học viên đăng ký hoặc đăng nhập vào website bằng tên tài khoản, mật khẩu để thực hiện quản lý tài khoản, xem, tìm kiếm thông tin khóa học, mua khóa học, vào học, xem lịch sử mua hàng, thanh toán online, gửi feedback.

Giảng viên đăng ký hoặc đăng nhập vào website bằng tên tài khoản, mật khẩu để thực hiện xem thông tin khóa học được phân công, xem các học viên đã đăng kí khóa học được phân công.

Trang web yêu cầu đăng nhập để thực hiện bất kỳ chức năng nào. Hệ thống lưu lại các phiếu mua hàng của người dùng. Người dùng thanh toán bằng Internet banking hoặc dùng tiền có sẵn trong tài khoản. Hệ thống lưu lại thông tin đơn hàng.

### Các yêu cầu chức năng

* Học viên:

+ Đăng ký: Khi chưa có tài khoản học viên có thể đăng ký tài khoản.

+ Đăng nhập: Học viên có thể đăng nhập vào hệ thống bằng tên tài khoản và mật khẩu.

+ Quản lý thông tin cá nhân: Học viên có thể chỉnh sửa thông tin cá nhân của mình.

+ Tìm kiếm khóa học: Học viên có thể tìm kiếm khóa học theo tên.

+ Xem chi tiết khóa học: Xem chi tiết khóa học có trên website và thêm vào giỏ hàng.

+ Thanh toán online: Học viên có thể nạp tiền vào tài khoản qua mã QR ngân hàng.

+ Mua khóa học: Học viên có thể mua bất kì khóa học nào trong giỏ hàng.

+ Vào học: Sau khi mua 1 khóa học người dùng có thể vào học khóa học đó.

+ Xem lịch sử mua hàng: Học viên có thể xem lịch sử nạp tiền, mua khóa học trên trang web.

+ Gửi feedback: Học viên có thể gửi feedback về các lỗi gặp phải cho Admin.

* Giảng viên:

+ Đăng ký: Khi chưa có tài khoản giảng viên có thể đăng ký tài khoản.

+ Đăng nhập: Giảng viên có thể đăng nhập vào hệ thống bằng tên tài khoản và mật khẩu.

+ Xem chi tiết khóa học: Xem danh sách các khóa học có trên website.

+ Xem phân công: Xem danh sách các khóa học được phân công và các học viên đăng kí khóa học.

* Admin:

+ Đăng nhập: Admin có thể đăng nhập vào hệ thống bằng tên tài khoản và mật khẩu.

+ Tìm kiếm tài khoản: Tìm kiếm tài khoản theo Id hoặc Email, lọc tài khoản theo danh sách chặn hoặc bỏ chặn.

+ Quản lý tài khoản: Admin có thể quản lý danh sách tài khoản: thêm, xóa tài khoản, chặn hoặc bỏ chặn tài khoản.

+ Quản lý khóa học: Admin có thể quản lý danh sách khóa học: thêm, xóa khóa học, chỉnh sửa thông tin.

+ Quản lý học viên: Admin có thể quản lý danh sách học viên: thêm, xóa học viên, thêm tiền cho học viên.

+ Quản lý giảng viên: Admin có thể quản lý danh sách giảng viên: thêm, xóa giảng viên, phân công khóa học cho giảng viên, xem danh sách học viên đăng kí khóa học đã phân công.

+ Quản lý feedback: Admin có thể quản lý danh sách feedback của học viên: thêm, xóa feedback, sửa trạng thái feedback.

+ Xem thống kê tổng quan: Admin có thể xem thống kê về doanh thu, số học viên, số giảng viên, số khóa học.

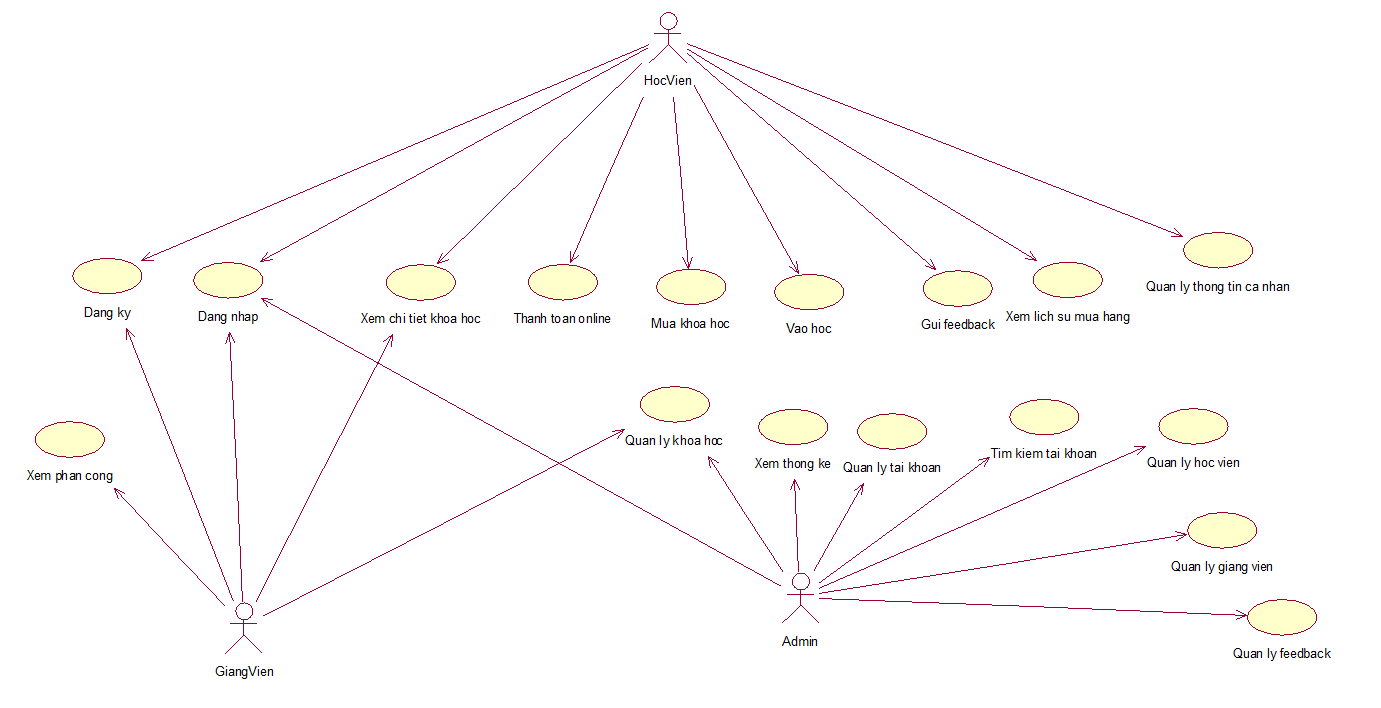
### Các yêu cầu phi chức năng

* Trang web được áp dụng rộng rãi, phổ biến cho mọi đối tượng
* Giao diện đơn giản, thân thiện, đẹp và dễ nhìn, dễ sử dụng cho mọi đối tượng mà không cần trình độ cao.
* Phải có tính bảo mật cao.
* Thao tác nhanh chóng, hợp lý, hiệu quả, chính xác.
* Cập nhật, phục hồi và sao lưu dữ liệu.

## Phân tích yêu cầu website

### Biểu đồ use case

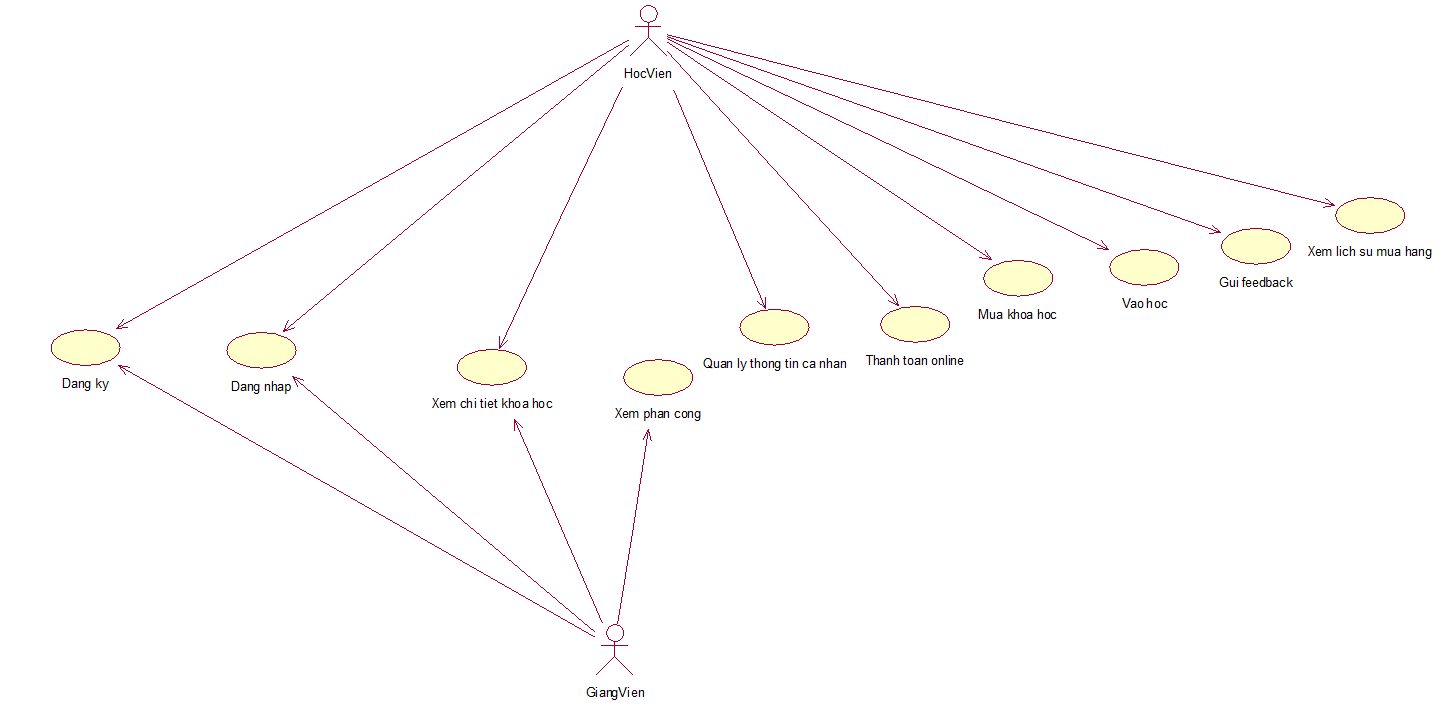
* Biểu đồ use case tổng quát:



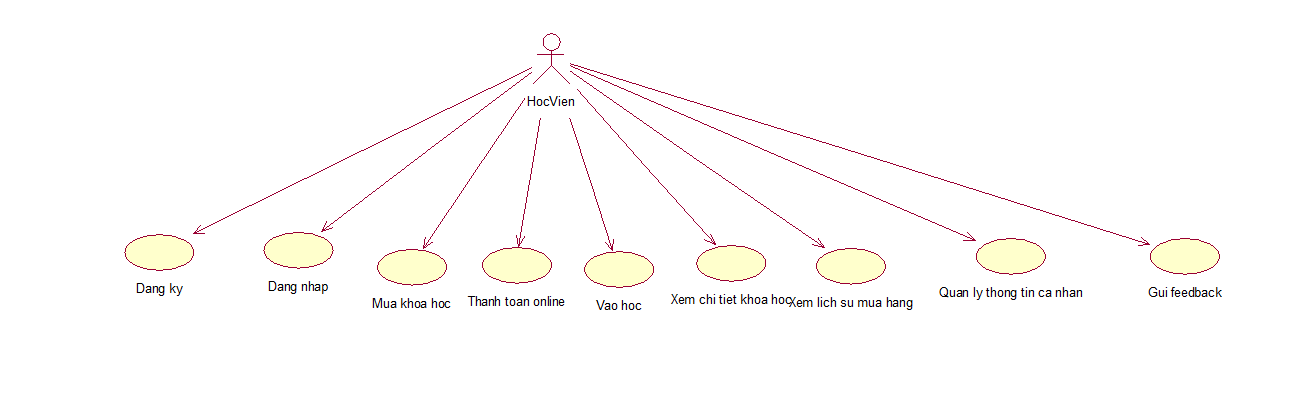
Hình 2.1. Biểu đồ use case tổng quát

* Biểu đồ use case phân rã:

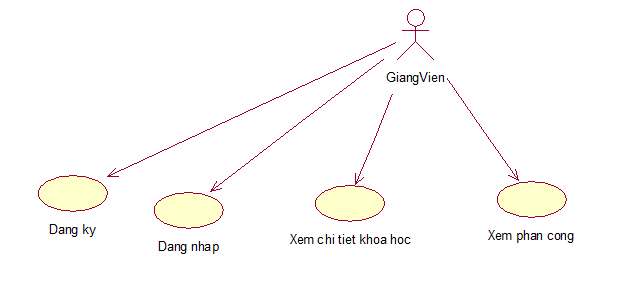
+ Biểu đồ front end:



Hình 2.2. Biểu đồ use case front end

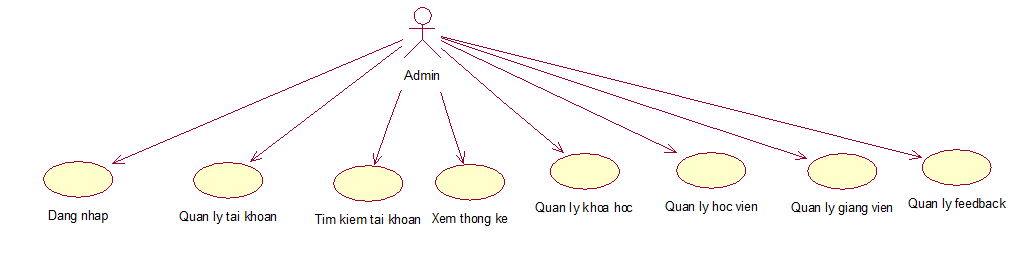


Hình 2.3. Biểu đồ use case front end hướng học viên



Hình 2.4. Biểu đồ use case front end hướng giảng viên

+ Biểu đồ back end:



Hình 2.5. Biểu đồ use case back end

### Mô tả use case

Bảng 2.1. Bảng Danh sách use case

|  |  |
| --- | --- |
| **STT** | **Tên use case** |
| UC01 | Đăng ký |
| UC02 | Đăng nhập |
| UC03 | Tìm kiếm tài khoản |
| UC04 | Xem chi tiết khóa học |
| UC05 | Thanh toán online |
| UC06 | Mua khóa học |
| UC07 | Vào học |
| UC08 | Gửi feedback |
| UC09 | Xem lịch sử thanh toán |
| UC10 | Xem phân công |
| UC11 | Quản lý tài khoản |
| UC12 | Quản lý khóa học |
| UC13 | Quản lý feedback |
| UC14 | Quản lý học viên |
| UC15 | Quản lý giảng viên |
| UC16 | Xem thống kê |

* Use case Đăng ký:

Bảng 2.2. Bảng Đặc tả use case Đăng ký

|  |  |
| --- | --- |
| **Mã use case** | **UC01** |
| **Tên use case** | **Đăng ký** |
| Tác nhân chính | Học viên, Giảng viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên, Giảng viên (Người dùng) |
| Tiền điều kiện | Không |
| Đảm bảo tối thiểu | Thông báo lỗi và quay lại bước trước |
| Đảm bảo thành công | Tác nhân đăng ký thành công và tài khoản được lưu vào CSDL |
| Kích hoạt | Người dùng kích vào nút “Đăng ký” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị màn hình đăng ký 2. Người dùng nhập các thông tin cần thiết theo form và kích nút “Đăng ký” 3. Hệ thống xác thực thông tin thành công và tạo tài khoản cho Người dùng 4. Hệ thống lưu tài khoản của Người dùng vào CSDL 5. Hệ thống thông báo đăng ký thành công. UC kết thúc. | |
| Ngoại lệ:  - Tại bước 2, nếu Người dùng cung cấp các thông tin yêu cầu không đúng, tên đăng nhập, email, mật khẩu không hợp lệ hay đã tồn tại, Hệ thống hiển thị thông báo lỗi, quay lại bước 2.  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Nếu use case này thành công thì Người dùng đăng ký được tài khoản. |

* Use case Đăng nhập:

Bảng 2.3. Bảng Đặc tả use case Đăng nhập

|  |  |
| --- | --- |
| **Mã use case** | **UC02** |
| **Tên use case** | **Đăng nhập** |
| Tác nhân chính | Học viên, Giảng viên (Người dùng), Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên, Giảng viên (Người dùng), Admin |
| Tiền điều kiện | Người dùng đã đăng ký tài khoản |
| Đảm bảo tối thiểu | Thông báo lỗi và quay lại bước trước |
| Đảm bảo thành công | Tác nhân đăng nhập thành công |
| Kích hoạt | Kích vào nút “Đăng nhập” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị màn hình đăng nhập. 2. Người dùng, Admin nhập tài khoản, mật khẩu và kích vào nút “Đăng nhập” 3. Hệ thống xác thực thông tin thành công và cho phép Người dùng, Admin truy cập vào trang chủ. UC kết thúc. | |
| Ngoại lệ:  - Tại bước 2, nếu Người dùng nhập sai mật khẩu hoặc tên tài khoản. Hệ thống hiển thị thông báo lỗi, quay lại bước 2.  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Nếu use case này thành công thì Người dùng vào được hệ thống. |

* Use case Tìm kiếm tài khoản:

Bảng 2.4. Bảng Đặc tả use case Tìm kiếm tài khoản

|  |  |
| --- | --- |
| **Mã use case** | **UC03** |
| **Tên use case** | **Tìm kiếm tài khoản** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Không có |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Admin có thể Tìm kiếm tài khoản theo AccountID hoặc Email  Admin có thể lọc danh sách tài khoản theo Lock/Unlock |
| Kích hoạt | Admin kích vào nút “Tìm kiếm tài khoản” |
| Chuỗi sự kiện chính:   1. Admin nhập từ khóa tìm kiếm vào ô tìm kiếm trên giao diện. 2. Hệ thống xác nhận từ khóa tìm kiếm và tiến hành tìm kiếm trong cơ sở dữ liệu bảng Account. 3. Hệ thống hiển thị kết quả tìm kiếm cho Admin. 4. Admin có thể lọc danh sách tài khoản bằng cách chọn theo Lock hoặc Unlock. | |
| Ngoại lệ:  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Hệ thống chỉ tìm kiếm trong phạm vi dữ liệu được phép truy cập của Admin. |

* Use case Xem chi tiết khóa học:

Bảng 2.5. Bảng Đặc tả use case Xem chi tiết khóa học

|  |  |
| --- | --- |
| **Mã use case** | **UC04** |
| **Tên use case** | **Xem chi tiết khóa học** |
| Tác nhân chính | Học viên, Giảng viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên, Giảng viên (Người dùng) |
| Tiền điều kiện | Không có |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Tác nhân xem thông tin các khóa học |
| Kích hoạt | Người dùng kích vào nút “Xem chi tiết khóa học” |
| Chuỗi sự kiện chính:   1. Khi Người dùng kích vào nút “Trang chủ” trong 1 trang bất kỳ. Hệ thống sẽ điều hướng url hiện tại của trang web về trang chủ. UC kết thúc. 2. Người dùng kích vào 1 khóa học bất kì. Hệ thống hiển thị ra màn hình chi tiết khóa học. UC kết thúc. | |
| Ngoại lệ:  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Không có |

* Use case Thanh toán online:

Bảng 2.6. Bảng Đặc tả use case Thanh toán online

|  |  |
| --- | --- |
| **Mã use case** | **UC05** |
| **Tên use case** | **Thanh toán online** |
| Tác nhân chính | Học viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Số dư trong tài khoản của Người dùng được cập nhật và giao dịch thanh toán được ghi nhận trong hệ thống. |
| Kích hoạt | Người dùng đã chọn chức năng nạp tiền. |
| Chuỗi sự kiện chính:   1. Người dùng chọn phương thức thanh toán trực tuyến. 2. Hệ thống chuyển hướng Người dùng đến cổng thanh toán trực tuyến và yêu cầu nhập thông tin thanh toán. 3. Người dùng nhập thông tin thanh toán, bao gồm số thẻ, ngày hết hạn, mã bảo mật, và một số thông tin khác cần thiết. 4. Hệ thống thanh toán trực tuyến xác nhận thông tin thanh toán và yêu cầu xác nhận từ Người dùng (nếu cần). 5. Người dùng xác nhận giao dịch thanh toán. 6. Hệ thống thanh toán trực tuyến xử lý giao dịch và chuyển trạng thái thanh toán thành công hoặc thất bại. 7. Hệ thống cập nhật số dư trong tài khoản của Người dùng và ghi nhận giao dịch trong lịch sử giao dịch. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo lỗi cho Học viên và yêu cầu thử lại hoặc chọn phương thức thanh toán khác. | |
| Hậu điều kiện: | Hệ thống cần tích hợp với các cổng thanh toán trực tuyến phổ biến để cho phép người dùng chọn lựa.  Các thông tin thanh toán của người dùng cần được mã hóa và bảo mật để đảm bảo tính an toàn.  Hệ thống cần hỗ trợ xử lý các lỗi và xác nhận giao dịch để đảm bảo tính chính xác và minh bạch trong quá trình thanh toán. |

* Use case Mua khóa học:

Bảng 2.7. Bảng Đặc tả use case Mua khóa học

|  |  |
| --- | --- |
| **Mã use case** | **UC06** |
| **Tên use case** | **Mua khóa học** |
| Tác nhân chính | Học viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Người dùng đã mua thành công khóa học và số dư trong tài khoản của họ đã được cập nhật. |
| Kích hoạt | Người dùng chọn khóa học mình muốn mua và kích vào nút “Mua” |
| Chuỗi sự kiện chính:   1. Người dùng chọn khóa học mà họ muốn mua trên trang web. 2. Hệ thống hiển thị thông tin chi tiết về khóa học, bao gồm giá cả và mô tả. 3. Người dùng xác nhận muốn mua khóa học bằng tiền có sẵn trong tài khoản của họ. 4. Hệ thống xác nhận số dư trong tài khoản của người dùng có đủ để mua khóa học hay không. 5. Nếu số dư đủ, hệ thống trừ số tiền tương ứng từ tài khoản của người dùng và ghi nhận giao dịch mua khóa học. 6. Hệ thống cập nhật tài khoản của người dùng để cho phép họ truy cập vào khóa học đã mua. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo lỗi cho người dùng, thông báo rằng họ không có đủ tiền trong tài khoản để mua khóa học và hướng dẫn họ nạp tiền vào tài khoản. | |
| Hậu điều kiện: | Hệ thống cần kiểm tra số dư trong tài khoản của người dùng trước khi thực hiện giao dịch mua khóa học để đảm bảo rằng họ có đủ tiền.  Quá trình mua khóa học bằng tiền trong tài khoản cần được xử lý một cách an toàn và bảo mật để đảm bảo tính toàn vẹn của giao dịch. |

* Use case Vào học:

Bảng 2.8. Bảng Đặc tả use case Vào học

|  |  |
| --- | --- |
| **Mã use case** | **UC07** |
| **Tên use case** | **Vào học** |
| Tác nhân chính | Học viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web.  Người dùng đã mua một khóa học trước đó trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Người dùng đã truy cập thành công vào nội dung giáo trình của khóa học. |
| Kích hoạt | Người dùng chọn khóa học mình muốn học và kích vào nút “Học” |
| Chuỗi sự kiện chính:   1. Người dùng chọn khóa học mà họ đã mua trước đó từ danh sách các khóa học có sẵn trong tài khoản của họ. 2. Hệ thống chuyển hướng người dùng đến trang chi tiết của khóa học đã mua. 3. Hệ thống hiển thị nội dung giáo trình của khóa học, bao gồm các bài học, tài liệu, video, bài tập, hoặc bất kỳ tài nguyên học tập nào khác. 4. Người dùng có thể chọn bất kỳ bài học cụ thể nào để bắt đầu học. | |
| Ngoại lệ:  - Nếu khóa học không có nội dung nào, hệ thống thông báo cho người dùng biết và cung cấp hướng dẫn cụ thể hoặc liên kết để họ có thể tiếp tục quá trình học. | |
| Hậu điều kiện: | Hệ thống cần đảm bảo rằng chỉ có người dùng đã mua khóa học mới có quyền truy cập vào nội dung giáo trình của khóa học đó.  Các tài nguyên giáo trình cần được hiển thị một cách rõ ràng và dễ dàng truy cập để người dùng có thể bắt đầu học một cách thuận tiện. |

* Use case Gửi feedback:

Bảng 2.9. Bảng Đặc tả use case Gửi feedback

|  |  |
| --- | --- |
| **Mã use case** | **UC08** |
| **Tên use case** | **Gửi feedback** |
| Tác nhân chính | Học viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Người dùng đã hoàn thành bài kiểm tra và nhận được kết quả hoặc phản hồi từ hệ thống. |
| Kích hoạt | Người dùng chọn bài kiểm tra mình muốn làm và kích vào nút “Làm” |
| Chuỗi sự kiện chính:   1. Người dùng truy cập vào trang chủ hoặc trang khóa học trên trang web. 2. Người dùng chọn mục "Gửi feedback" trong phần Profile hoặc biểu tượng tương ứng. 3. Hệ thống mở một biểu mẫu hoặc trang để học viên nhập thông tin feedback. 4. Người dùng nhập nội dung feedback của họ vào biểu mẫu, bao gồm những ý kiến, đề xuất hoặc các vấn đề mà họ muốn chia sẻ với nhà cung cấp khóa học. 5. Người dùng nhấn nút "Gửi" để hoàn tất quá trình gửi feedback. 6. Hệ thống xác nhận việc gửi feedback thành công và hiển thị thông báo cho học viên. | |
| Ngoại lệ:  - Nếu bài kiểm tra hoặc bài tập không được điểm, hệ thống chỉ cung cấp phản hồi chi tiết cho người dùng sau khi họ đã gửi bài làm. | |
| Hậu điều kiện: | Feedback đã được gửi thành công và được lưu trữ trong hệ thống. |

* Use case Xem lịch sử thanh toán:

Bảng 2.10. Bảng Đặc tả use case Xem lịch sử thanh toán

|  |  |
| --- | --- |
| **Mã use case** | **UC09** |
| **Tên use case** | **Xem lịch sử thanh toán** |
| Tác nhân chính | Học viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Học viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Người dùng đã xem thành công lịch sử các giao dịch mua khóa học và nạp tiền. |
| Kích hoạt | Người dùng chọn Xem lịch sử thanh toán trên cửa sổ pop-up khi di chuột vào Avatar tài khoản. |
| Chuỗi sự kiện chính:   1. Người dùng truy cập vào phần quản lý thông tin cá nhân và lịch sử giao dịch trên trang web. 2. Người dùng chọn chức năng xem lịch sử mua khóa học và nạp tiền. 3. Hệ thống hiển thị danh sách các giao dịch mua khóa học và nạp tiền gần đây của người dùng, bao gồm các thông tin như thời gian, số tiền, và trạng thái của giao dịch. 4. Người dùng có thể xem chi tiết về từng giao dịch bằng cách nhấp vào mỗi mục trong danh sách. | |
| Ngoại lệ:  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Hệ thống cần lưu trữ thông tin chi tiết về các giao dịch mua khóa học và nạp tiền của người dùng để có thể hiển thị lịch sử cho họ.  Thông tin trong lịch sử giao dịch cần được hiển thị một cách rõ ràng và dễ hiểu cho người dùng. |

* Use case Xem phân công:

Bảng 2.11. Bảng Đặc tả use case Xem phân công

|  |  |
| --- | --- |
| **Mã use case** | **UC10** |
| **Tên use case** | **Xem phân công** |
| Tác nhân chính | Giảng viên (Người dùng) |
| Mức | 1 |
| Người chịu trách nhiệm | Giảng viên (Người dùng) |
| Tiền điều kiện | Người dùng đã đăng nhập vào tài khoản trên trang web.  Có giao dịch mua khóa học đã được thực hiện trên trang web, từ đó tạo ra doanh thu. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Hiển thị danh sách các khóa học được phân công và học viên đã đăng ký khóa học đó. |
| Kích hoạt | Người dùng chọn Xem phân công trên bảng thông tin chi tiết khóa học. |
| Chuỗi sự kiện chính:   1. Người dùng truy cập vào trang Assignment trên trang web giáo dục. 2. Người dùng chọn một khóa học trong những khóa học được phân công. 3. Hệ thống hiển thị lên thông tin chi tiết khóa học và những học sinh đã đăng ký khóa học đó. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho người dùng biết rằng không có phân công nào được tìm thấy. | |
| Hậu điều kiện: | Danh sách các phân công cần được hiển thị một cách rõ ràng và dễ dàng hiểu để người dùng có thể quản lý thời gian và hoàn thành công việc một cách hiệu quả.  Thông tin về các phân công cần được cập nhật đều đặn và chính xác để đảm bảo tính toàn vẹn và đáng tin cậy của hệ thống phân công. |

* Use case Quản lý tài khoản:

Bảng 2.12. Bảng Đặc tả use case Quản lý tài khoản

|  |  |
| --- | --- |
| **Mã use case** | **UC11** |
| **Tên use case** | **Quản lý tài khoản** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Admin đã đăng nhập thành công |
| Đảm bảo thành công | Chức năng thực hiện thành công sẽ lưu vào CSDL |
| Kích hoạt | Admin chọn chức năng quản lý tài khoản. |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị giao diện quản lý tài khoản. 2. Admin kích vào nút “Sửa” để chỉnh sửa tài khoản    * Hệ thống hiển thị form chỉnh sửa tài khoản    * chỉnh sửa các thông tin cần thiết và kích vào nút “Save”    * Hệ thống cập nhật lại dữ liệu và tự động quay trở lại màn hình danh sách thành viên trước đó 3. Admin kích vào nút “Xoá” để xoá danh mục    * Hệ thống hiển thị màn hình xác nhận Xóa.    * Admin xác nhận Xóa.    * Hệ thống xoá thành viên đó trong CSDL. 4. Admin thoát khỏi chức năng quản lý thành viên. UC kết thúc. | |
| Ngoại lệ:  - Tại bước 3, nếu admin không xác nhận xoá thì hệ thống quay trở lại màn hình danh sách thành viên trước đó.  - Tại 1 thời điểm bất kì, nếu hệ thống không kết nối được với CSDL. Hệ thống thông báo lỗi. UC kết thúc. | |
| Hậu điều kiện: | Hệ thống cập nhật lại CSDL |

* Use case Quản lý khóa học:

Bảng 2.13. Bảng Đặc tả use case Quản lý khóa học

|  |  |
| --- | --- |
| **Mã use case** | **UC12** |
| **Tên use case** | **Quản lý khóa học** |
| Tác nhân chính | Giảng viên (Người dùng), Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Giảng viên (Người dùng), Admin |
| Tiền điều kiện | Người dùng, Admin đã đăng nhập vào tài khoản trên trang web.  Người dùng đã tạo khóa học trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Các thay đổi về quản lý khóa học đã được áp dụng thành công. |
| Kích hoạt | Người dùng, Admin chọn Quản lý khóa học. |
| Chuỗi sự kiện chính:   1. Người dùng, Admin truy cập vào trang quản lý khóa học trên trang web. 2. Hệ thống hiển thị danh sách các khóa học mà người dùng, Admin đã tạo. 3. Người dùng, Admin có thể thực hiện các hành động quản lý sau:  * Xem thông tin chi tiết về mỗi khóa học. * Chỉnh sửa thông tin khóa học. * Xóa khóa học. * Gán quyền truy cập cho người dùng khác. * Quản lý học viên đã đăng ký. * Tạo bài kiểm tra hoặc bài tập mới.  1. Sau khi hoàn thành các thao tác, người dùng, Admin lưu các thay đổi. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho họ biết rằng không có khóa học nào được tìm thấy và cung cấp hướng dẫn cụ thể về cách mua hoặc tạo khóa học mới. | |
| Hậu điều kiện: | Admin có quyền truy cập và quản lý tất cả các khóa học trên trang web, trong khi người dùng chỉ có quyền truy cập và quản lý các khóa học họ đã tạo.  Các thao tác quản lý khóa học cần phải được bảo mật và kiểm soát truy cập đúng đắn để đảm bảo tính toàn vẹn và an toàn của dữ liệu. |

* Use case Quản lý feedback:

Bảng 2.14. Bảng Đặc tả use case Quản lý feedback

|  |  |
| --- | --- |
| **Mã use case** | **UC13** |
| **Tên use case** | **Quản lý feedback** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Các thay đổi về Quản lý feedback đã được áp dụng thành công. |
| Kích hoạt | Admin chọn Quản lý feedback. |
| Chuỗi sự kiện chính:   1. Admin truy cập vào trang Quản lý feedback trên trang web. 2. Hệ thống hiển thị danh sách các feedback được gửi bởi người dùng, bao gồm các thông tin như tên người gửi, ngày gửi, nội dung của feedback. 3. Admin có thể thực hiện các hành động quản lý sau:  * Xem thông tin chi tiết về feedback. * Đánh dấu feedback là đã đọc hoặc chưa đọc để theo dõi quá trình xử lý. * Xóa feedback khỏi hệ thống nếu nội dung không phù hợp hoặc không cần thiết.  1. Sau khi hoàn thành các thao tác, Admin lưu các thay đổi. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho Admin biết rằng không có feedback nào được gửi và không có hành động quản lý cần thực hiện. | |
| Hậu điều kiện: | Admin có quyền truy cập và quản lý tất cả các feedback trên trang web.  Các thao tác quản lý feedback cần phải được bảo mật và kiểm soát truy cập đúng đắn để đảm bảo tính toàn vẹn và an toàn của dữ liệu. |

* Use case Quản lý học viên:

Bảng 2.15. Bảng Đặc tả use case Quản lý học viên

|  |  |
| --- | --- |
| **Mã use case** | **UC14** |
| **Tên use case** | **Quản lý học viên** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Các thay đổi về Quản lý học viên đã được áp dụng thành công. |
| Kích hoạt | Admin chọn Quản lý học viên. |
| Chuỗi sự kiện chính:   1. Admin truy cập vào trang quản lý học viên trên trang web. 2. Hệ thống hiển thị danh sách các học viên đã được tạo trên trang web, bao gồm cả thông tin chi tiết về học viên. 3. Admin có thể thực hiện các hành động quản lý sau:  * Xem thông tin chi tiết về mỗi học viên. * Thêm tiền vào tài khoản cho học viên. * Xem lịch sử giao dịch của học viên. * Xóa học viên khỏi hệ thống.  1. Sau khi hoàn thành các thao tác, Admin lưu các thay đổi và thoát khỏi trang quản lý học viên. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho quản trị viên biết rằng không có học viên nào được tìm thấy . | |
| Hậu điều kiện: | Admin có quyền truy cập và quản lý tất cả các học viên trên trang web.  Các thao tác quản lý học viên cần phải được bảo mật và kiểm soát truy cập đúng đắn để đảm bảo tính toàn vẹn và an toàn của dữ liệu. |

* Use case Quản lý giảng viên:

Bảng 2.16. Bảng Đặc tả use case Quản lý giảng viên

|  |  |
| --- | --- |
| **Mã use case** | **UC15** |
| **Tên use case** | **Quản lý giảng viên** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Các thay đổi về Quản lý giảng viên đã được áp dụng thành công. |
| Kích hoạt | Admin chọn Quản lý giảng viên. |
| Chuỗi sự kiện chính:   1. Admin truy cập vào trang quản lý giảng viên trên trang web. 2. Hệ thống hiển thị danh sách các giảng viên đã được tạo trên trang web, bao gồm cả thông tin chi tiết về giảng viên. 3. Admin có thể thực hiện các hành động quản lý sau:  * Xem thông tin chi tiết về mỗi giảng viên. * Thêm khóa học vào danh sách phân công cho giảng viên. * Xem các khóa học đã phân công và danh sách học viên đã đăng ký khóa học đó. * Xóa giảng viên khỏi hệ thống.  1. Sau khi hoàn thành các thao tác, Admin lưu các thay đổi và thoát khỏi trang quản lý giảng viên. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho quản trị viên biết rằng không có giảng viên nào được tìm thấy . | |
| Hậu điều kiện: | Admin có quyền truy cập và quản lý tất cả các giảng viên trên trang web.  Các thao tác quản lý giảng viên cần phải được bảo mật và kiểm soát truy cập đúng đắn để đảm bảo tính toàn vẹn và an toàn của dữ liệu. |

* Use case Xem thống kê:

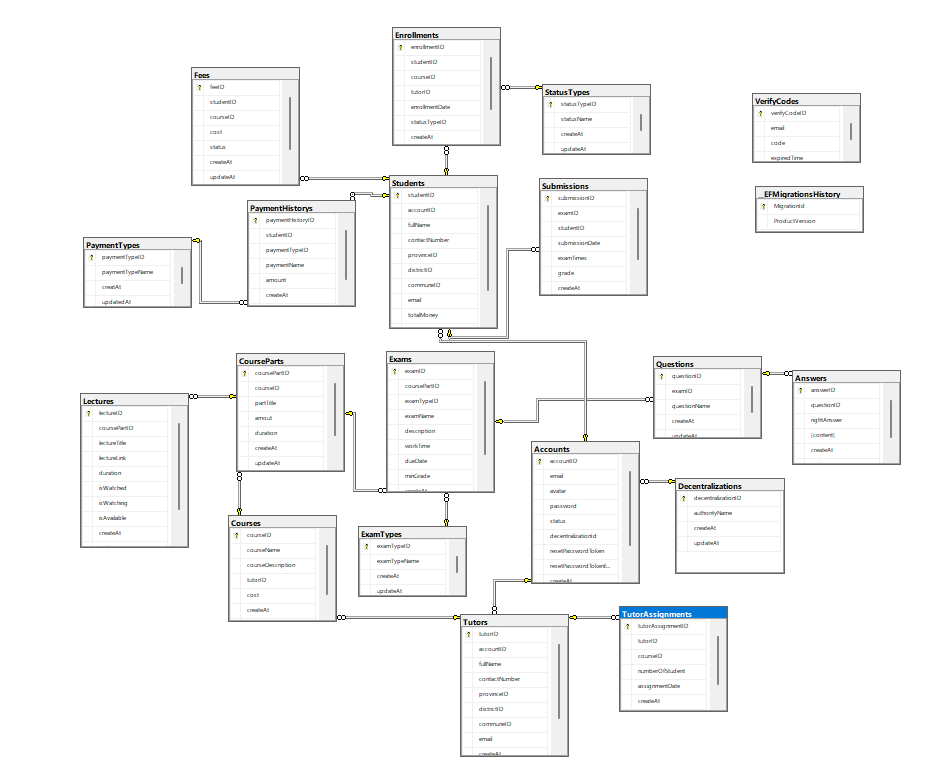
Bảng 2.17. Bảng Đặc tả use case Xem thống kê

|  |  |
| --- | --- |
| **Mã use case** | **UC16** |
| **Tên use case** | **Xem thống kê** |
| Tác nhân chính | Admin |
| Mức | 1 |
| Người chịu trách nhiệm | Admin |
| Tiền điều kiện | Admin đã đăng nhập vào tài khoản trên trang web. |
| Đảm bảo tối thiểu | Không có |
| Đảm bảo thành công | Admin đã xem được các thống kê và báo cáo theo yêu cầu. |
| Kích hoạt | Admin chọn Dashboard. |
| Chuỗi sự kiện chính:   1. Admin truy cập vào trang thống kê hoặc báo cáo trên trang web. 2. Hệ thống hiển thị danh sách các thống kê có sẵn:  * Doanh thu từ việc bán khóa học hoặc các sản phẩm khác trên trang web. * Số lượng khóa học đã được tạo hoặc bán. * Số lượng người dùng đăng ký mới trong một khoảng thời gian nhất định. * Tỷ lệ các khóa học được mua.  1. Sau khi hoàn thành các thao tác, Admin lưu các thay đổi và thoát khỏi trang quản lý học viên. | |
| Ngoại lệ:  - Hệ thống hiển thị thông báo cho quản trị viên biết rằng không có học viên nào được tìm thấy . | |
| Hậu điều kiện: | Admin có quyền truy cập và quản lý tất cả các học viên trên trang web.  Các thao tác quản lý học viên cần phải được bảo mật và kiểm soát truy cập đúng đắn để đảm bảo tính toàn vẹn và an toàn của dữ liệu. |

### Biểu đồ tuần tự

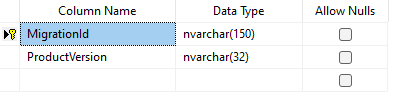
* Use case Đăng ký:
* Use case Đăng nhập:
* Use case Tìm kiếm tài khoản:
* Use case Xem chi tiết khóa học:
* Use case Thanh toán Online:
* Use case Mua khóa học:
* Use case Vào học:
* Use case Gửi feedback:
* Use case Xem lịch sử thanh toán:
* Use case Xem phân công:
* Use case Quản lý tài khoản:
* Use case Quản lý khóa học:
* Use case Quản lý feedback:
* Use case Quản lý học viên:
* Use case Quản lý giảng viên:
* Use case Xem thống kê:

### Biểu đồ quan hệ

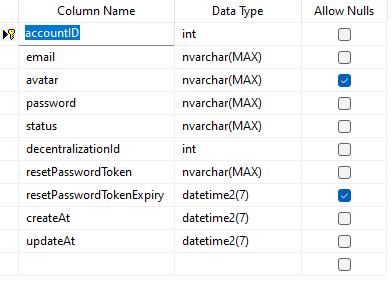


Hình 2.6. Sơ đồ CSDL tổng quát

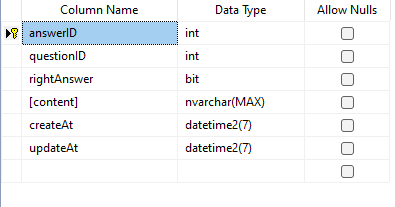
CSDL của website được thiết kế theo chuẩn 3NF, giúp CSDL súc tích, dễ hiểu với người dùng.



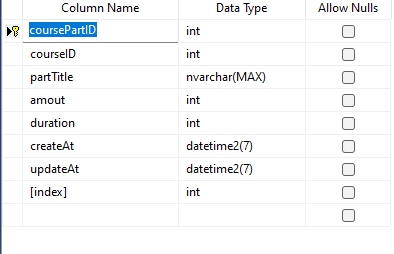
Hình 2.7. Bảng \_EFMigrationsHistory



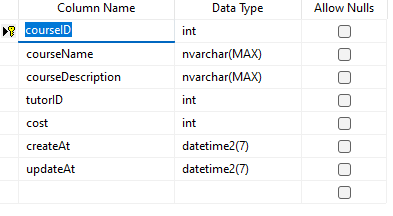
Hình 2.8. Bảng Accounts



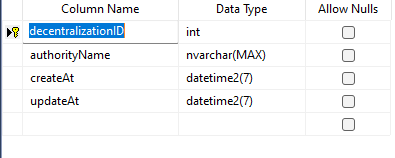
Hình 2.9. Bảng Answers



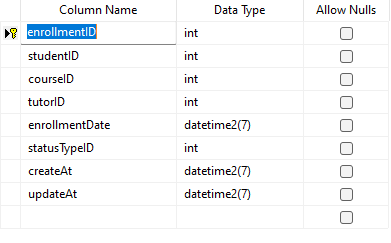
Hình 2.10. Bảng CourseParts



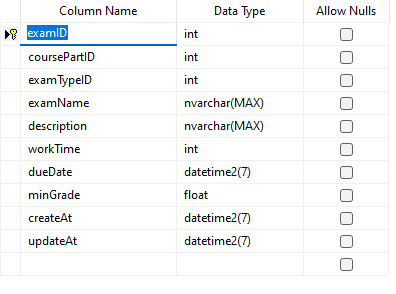
Hình 2.11. Bảng Courses



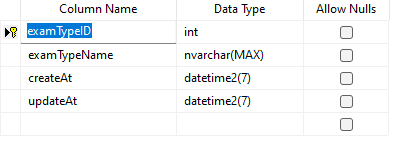
Hình 2.12. Bảng Decentralizations



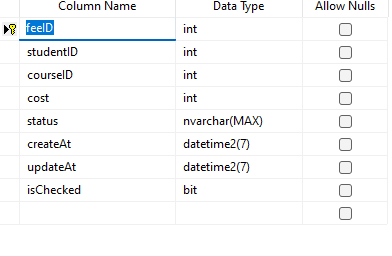
Hình 2.13. Bảng Enrollments



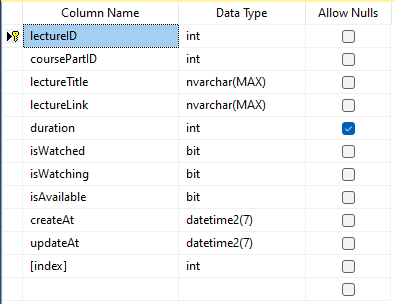
Hình 2.14. Bảng Exams



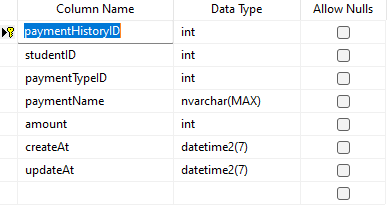
Hình 2.15. Bảng ExamTypes



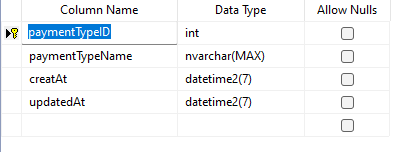
Hình 2.16. Bảng Fees



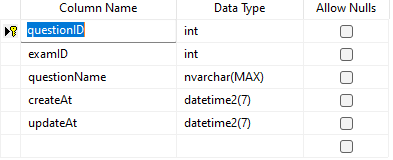
Hình 2.17. Bảng Lectures



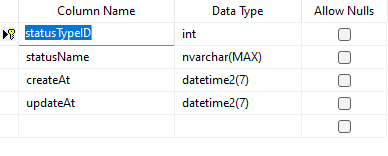
Hình 2.18. Bảng PaymentHistorys



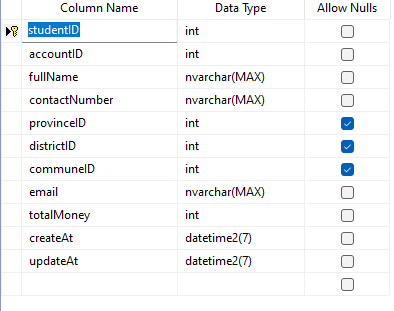
Hình 2.19. Bảng PaymentTypes



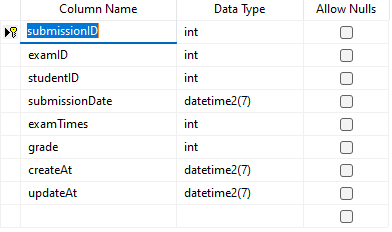
Hình 2.20. Bảng Questions



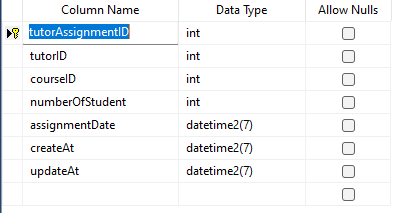
Hình 2.21. Bảng StatusTypes



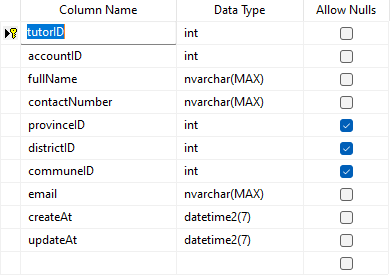
Hình 2.22. Bảng Students



Hình 2.23. Bảng Submissions



Hình 2.24. Bảng TutorAssignments



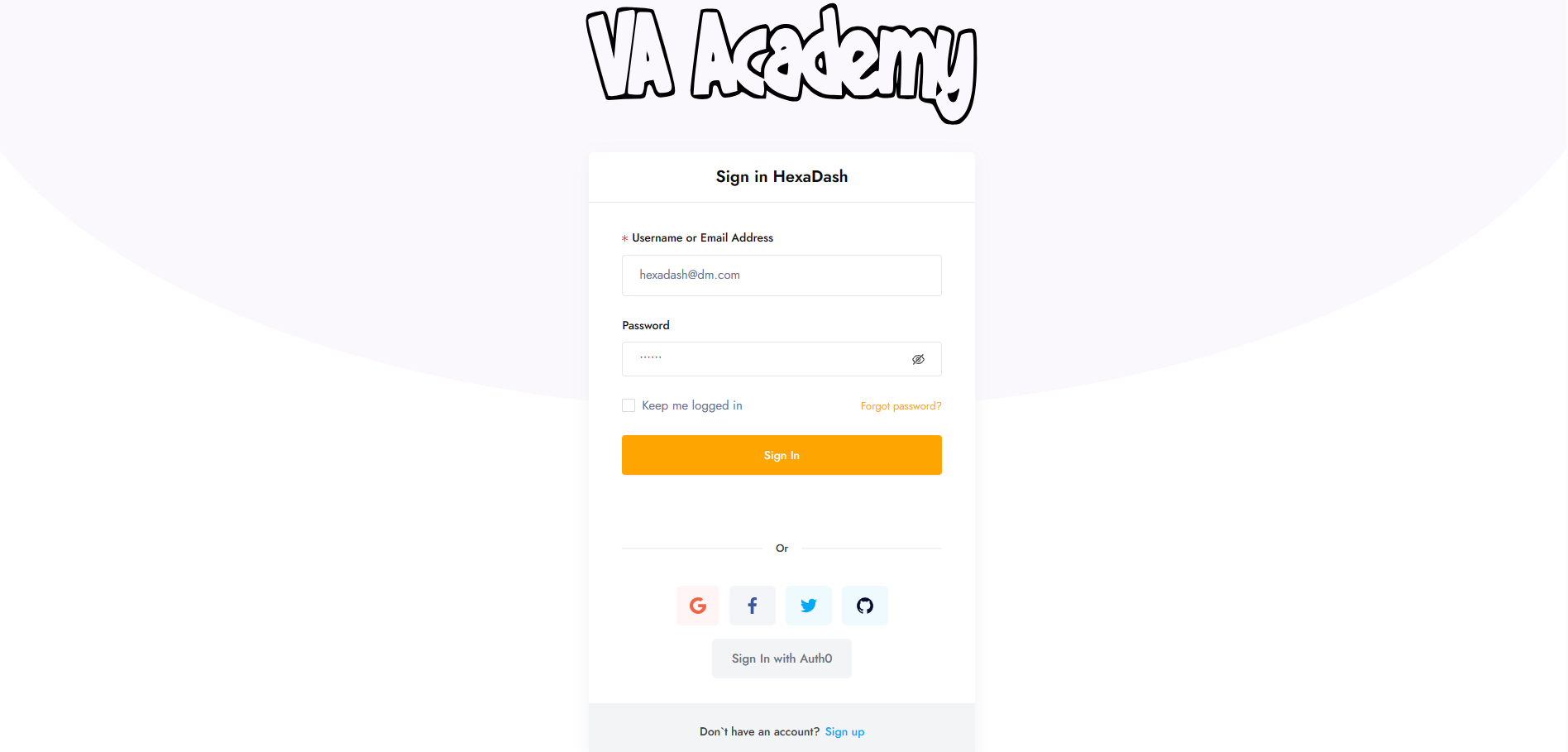
Hình 2.25. Bảng Tutors



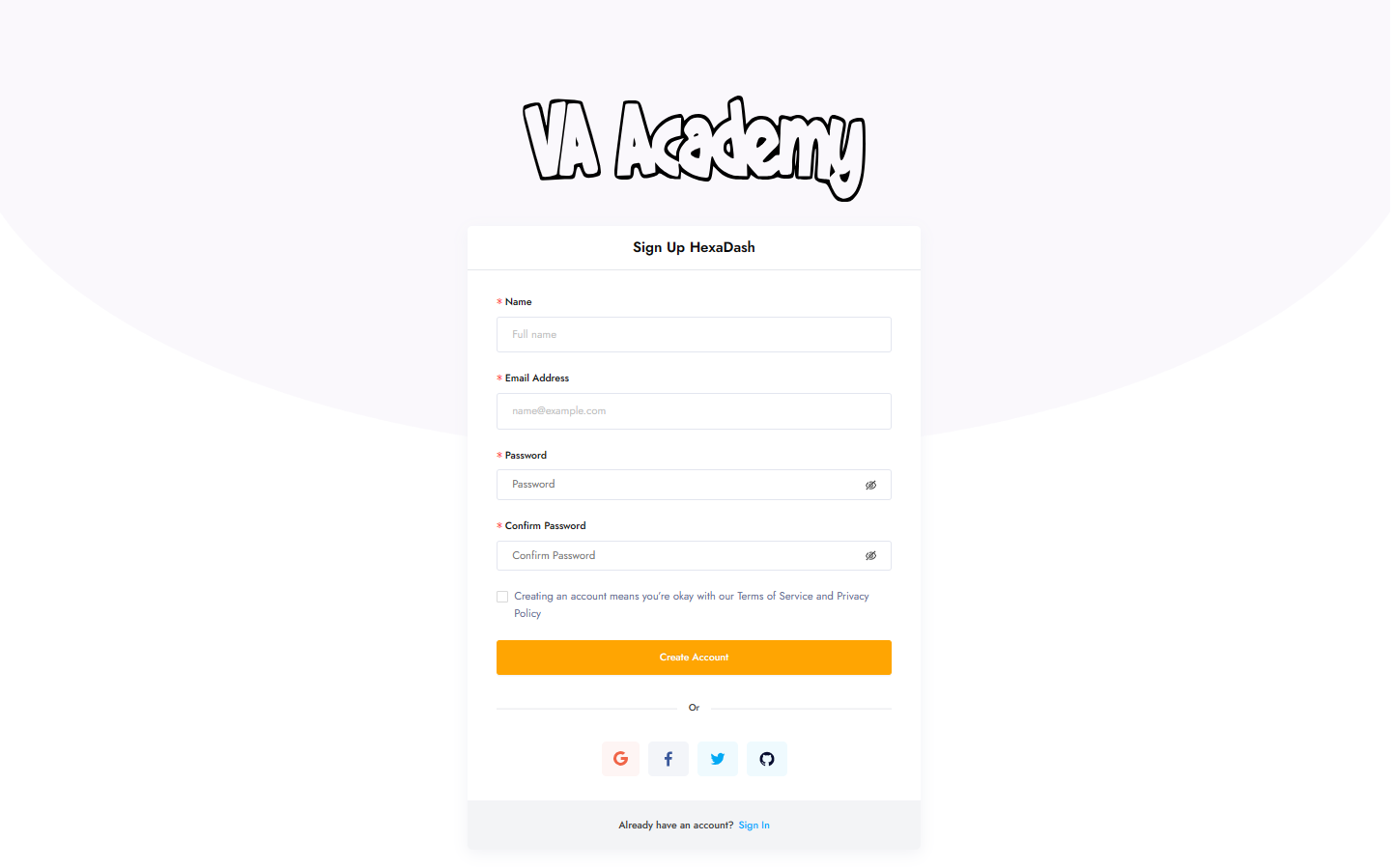
Hình 2.26. Bảng VerifyCodes

# THIẾT KẾ PHẦN MỀM

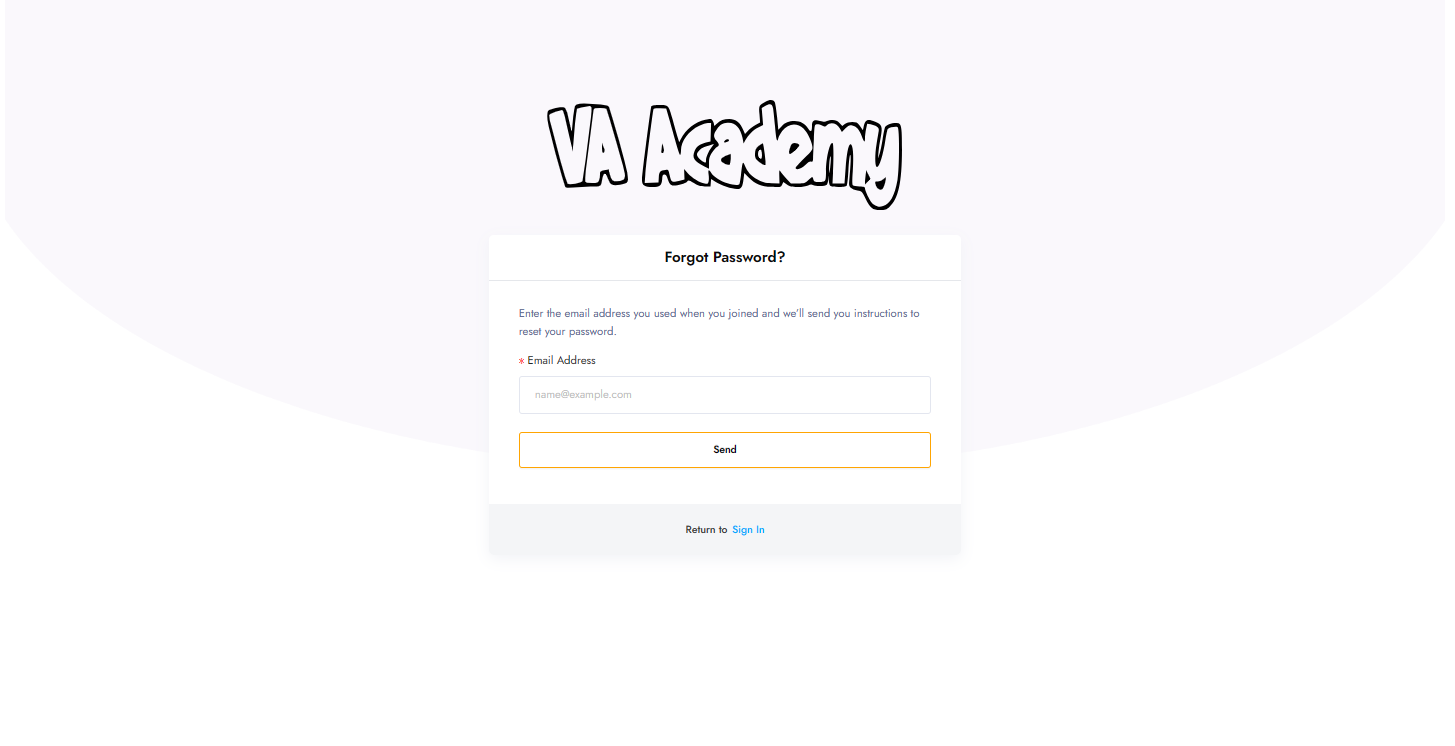
## Giao diện khách



Hình 3.1. Giao diện trang Đăng nhập

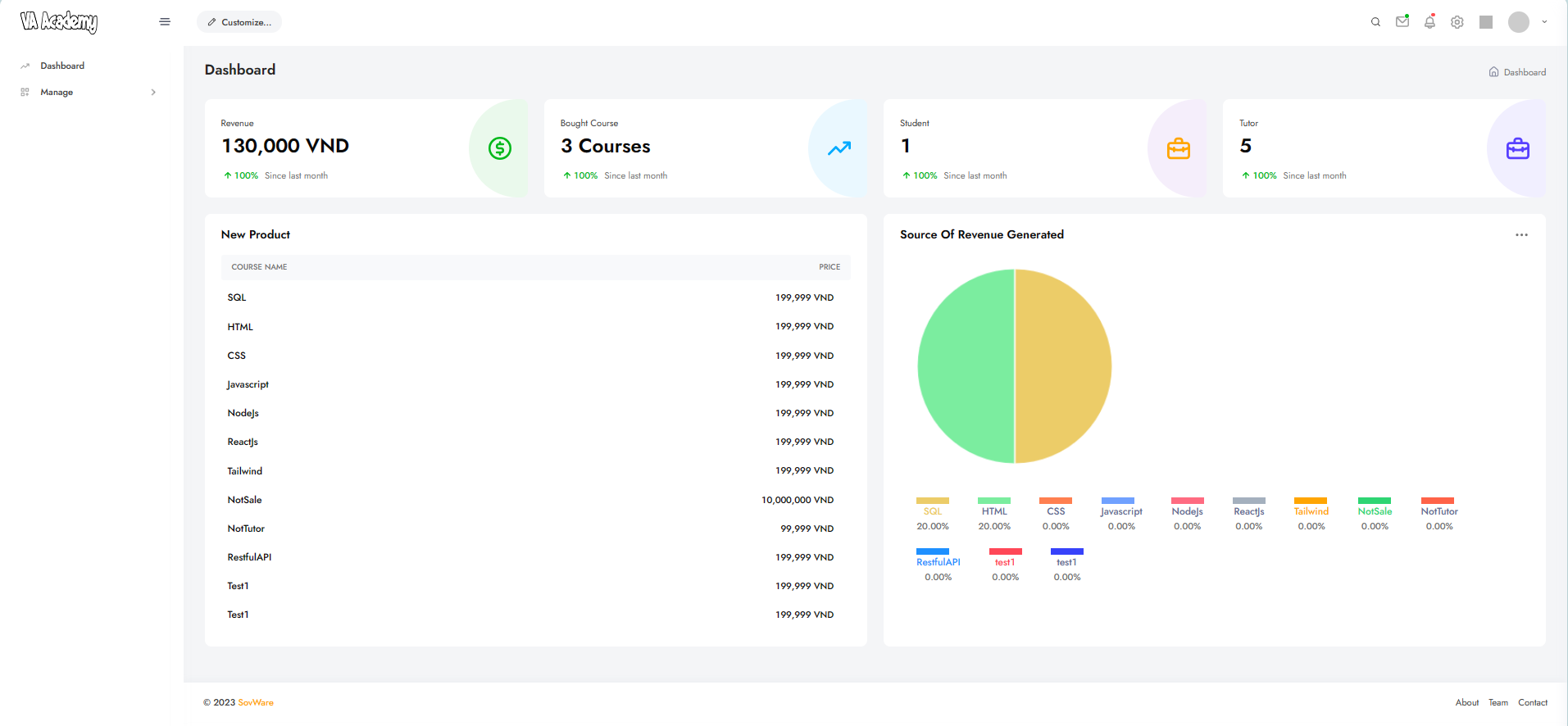


Hình 3.2. Giao diện trang Đăng ký

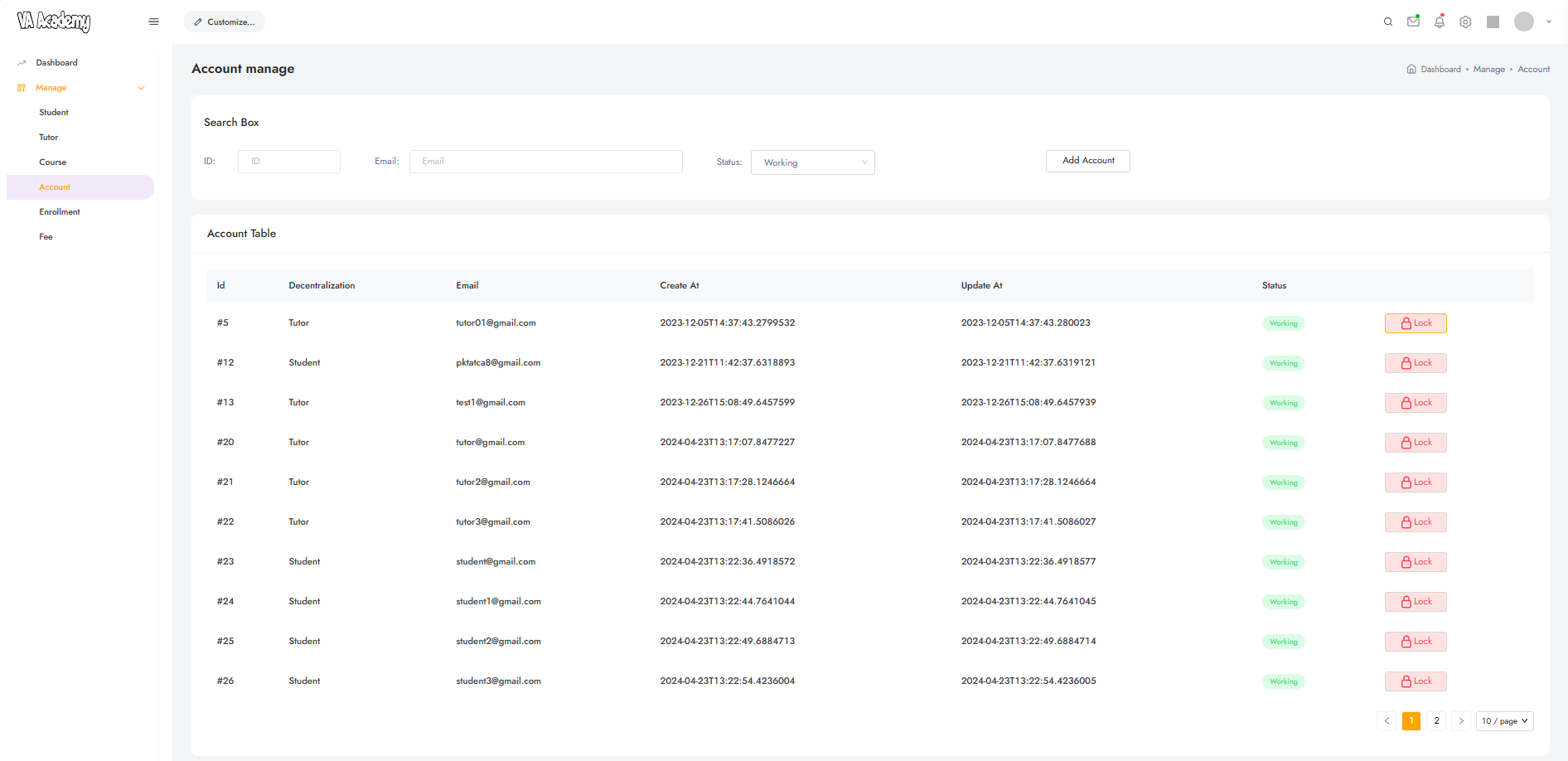


Hình 3.3. Giao diện trang Quên mật khẩu

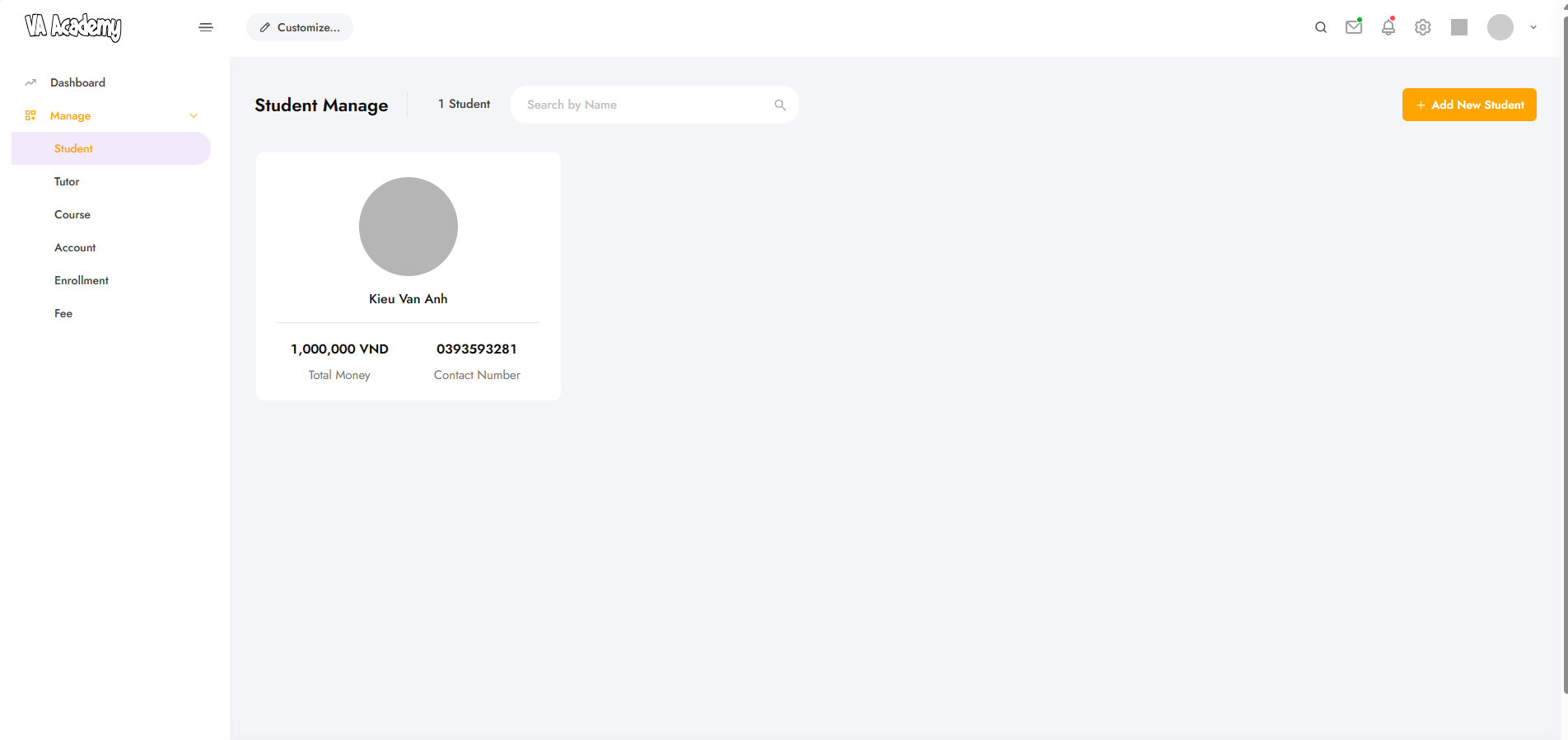
## Giao diện trang Admin



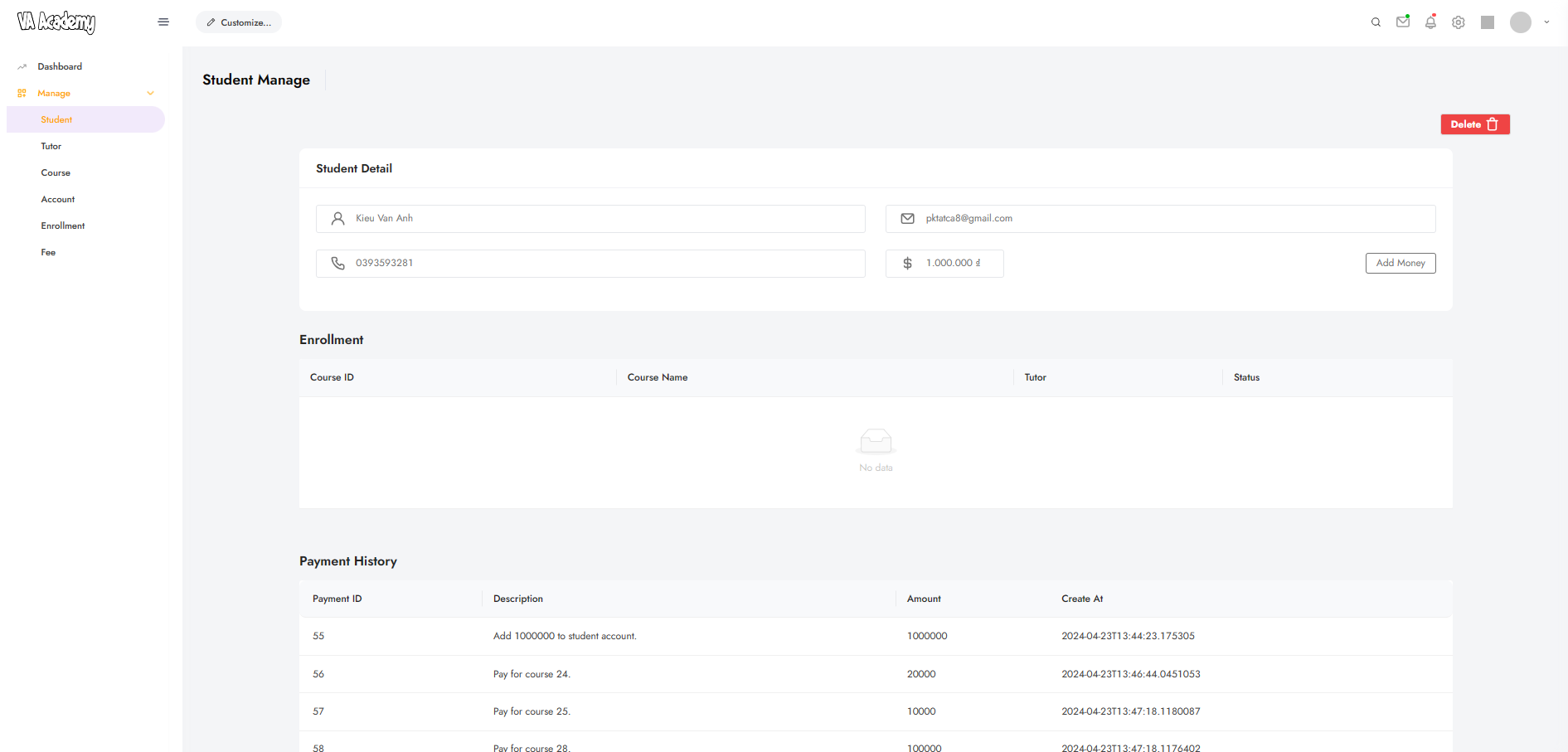
Hình 3.4. Giao diện trang chủ Admin



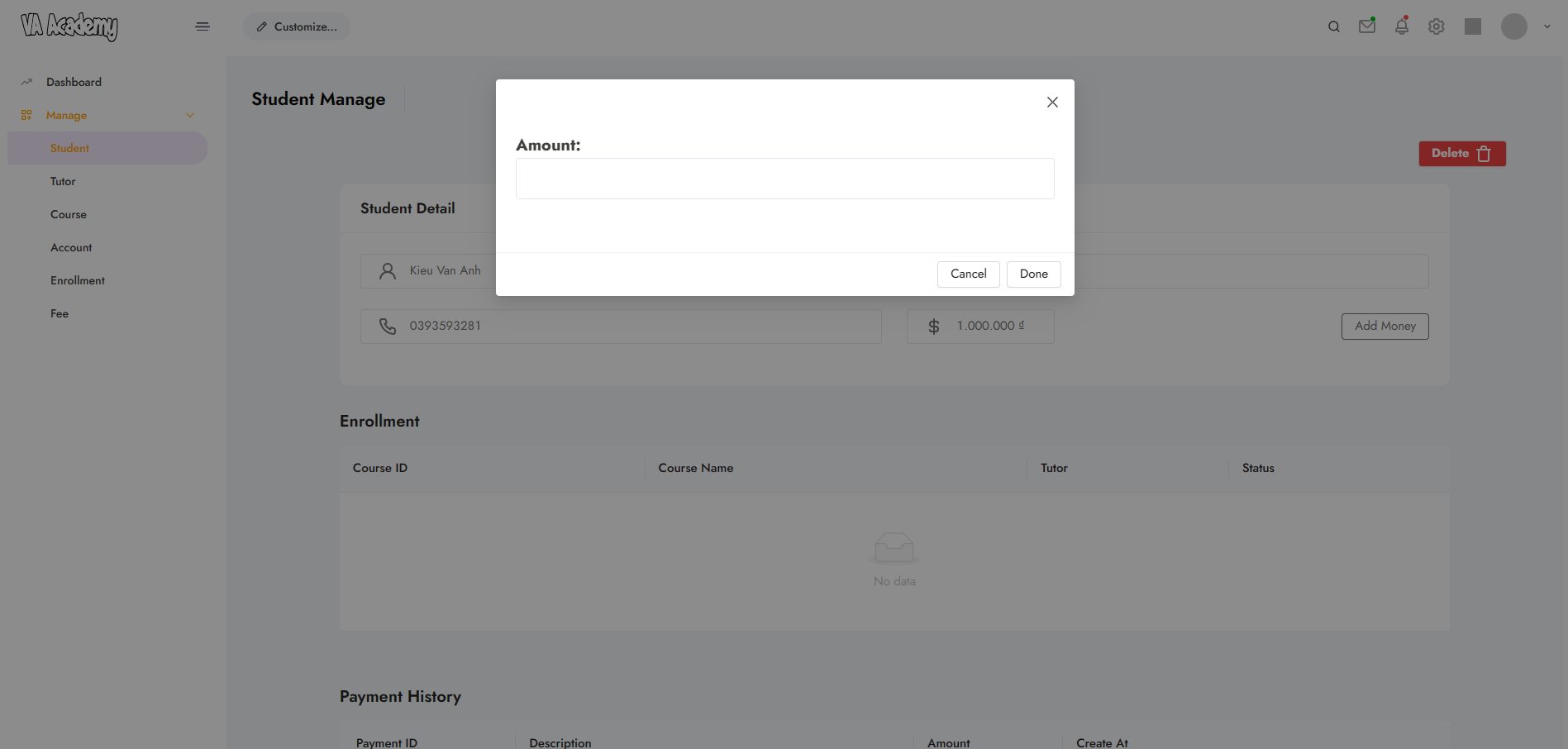
Hình 3.5. Giao diện trang Quản lý tài khoản



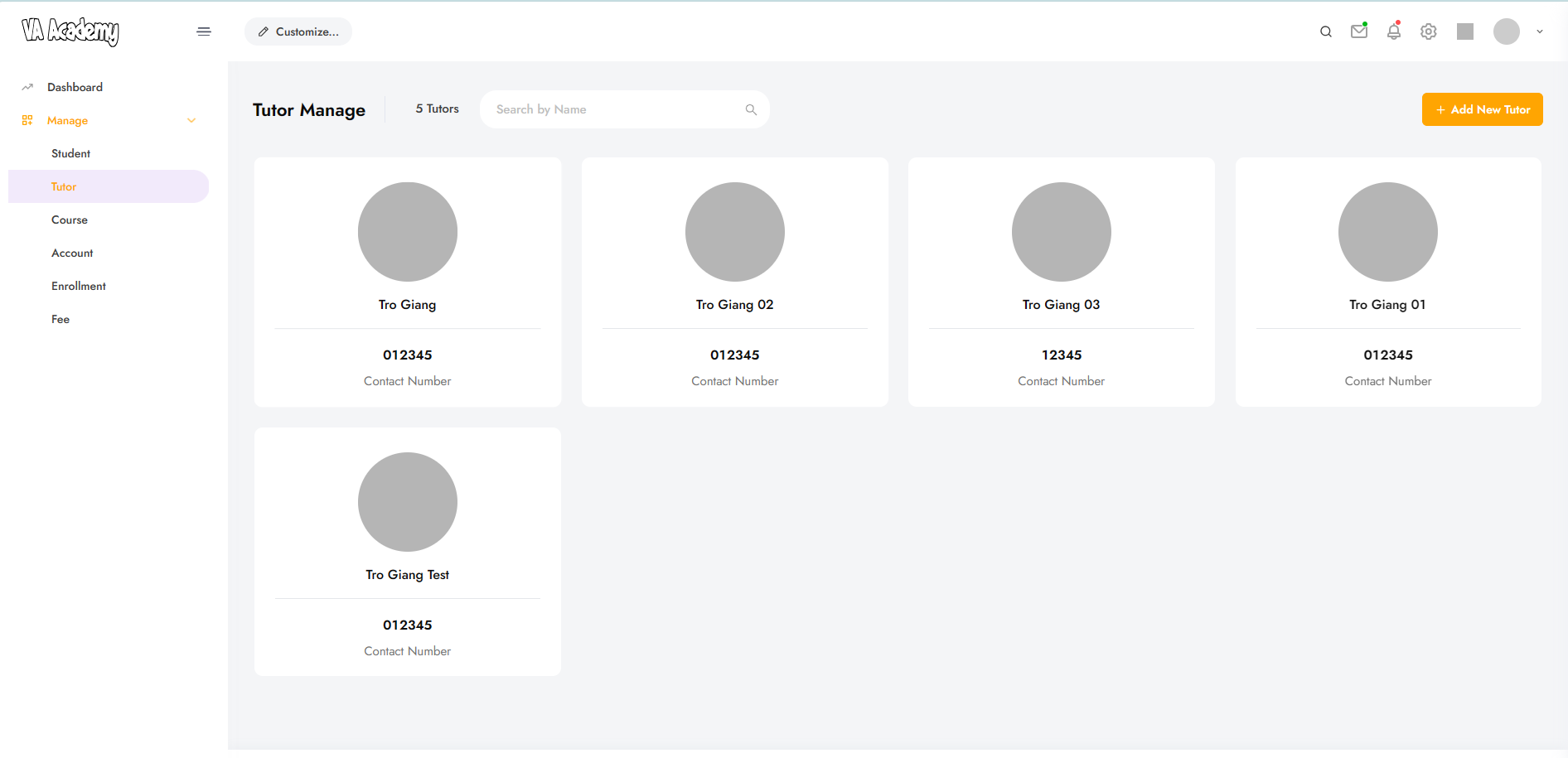
Hình 3.6. Giao diện trang Quản lý học viên



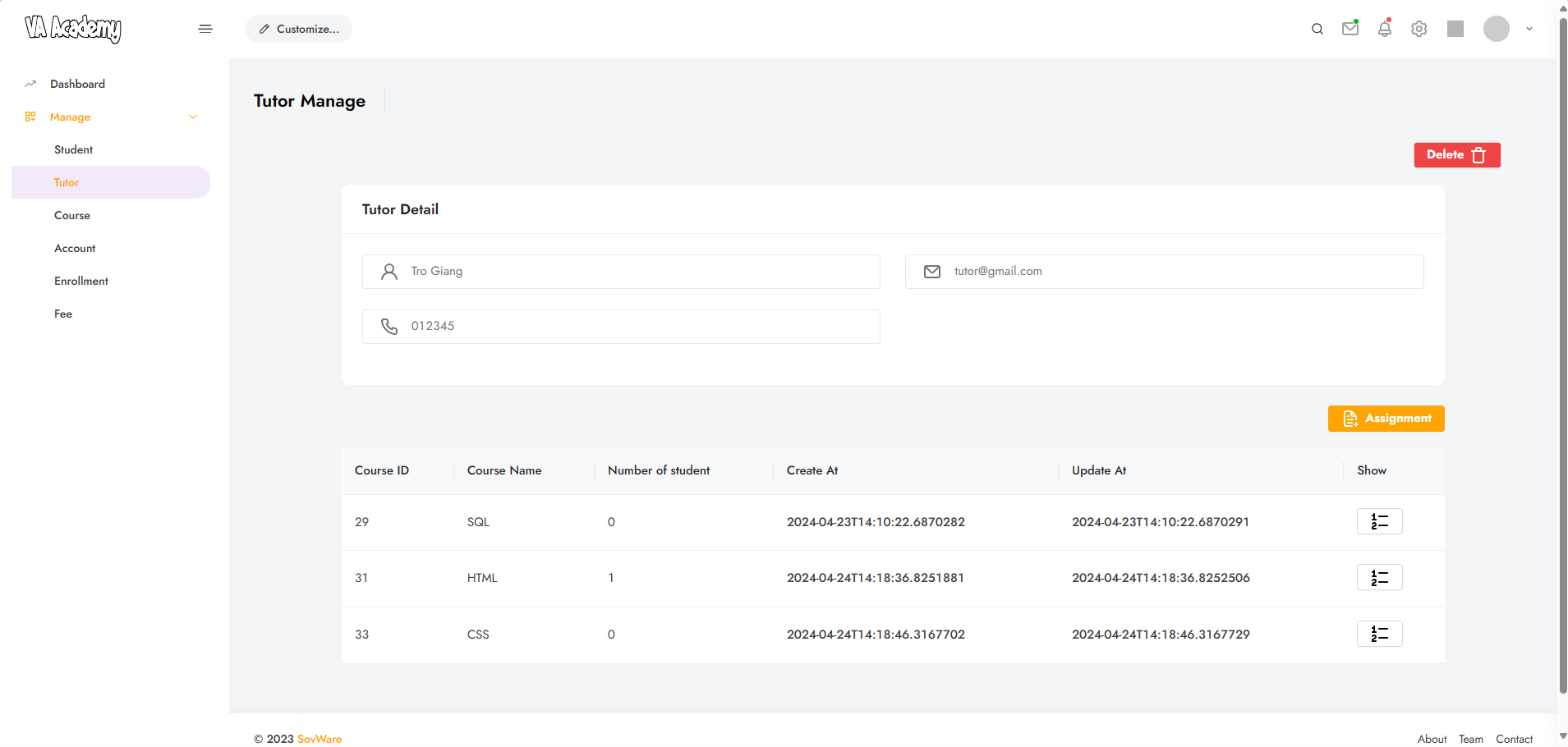
Hình 3.7. Giao diện trang Chi tiết học viên



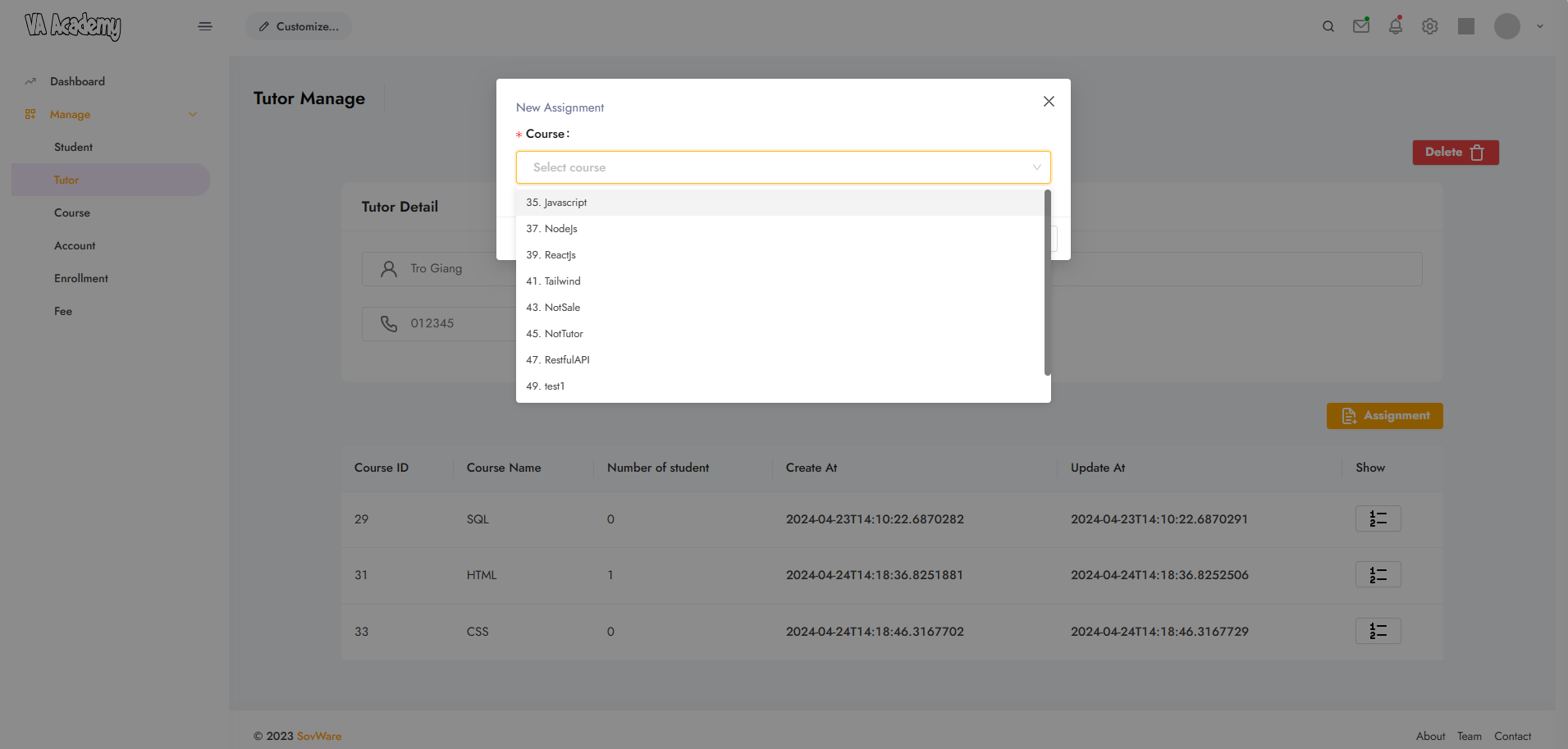
Hình 3.8. Giao diện Thêm tiền



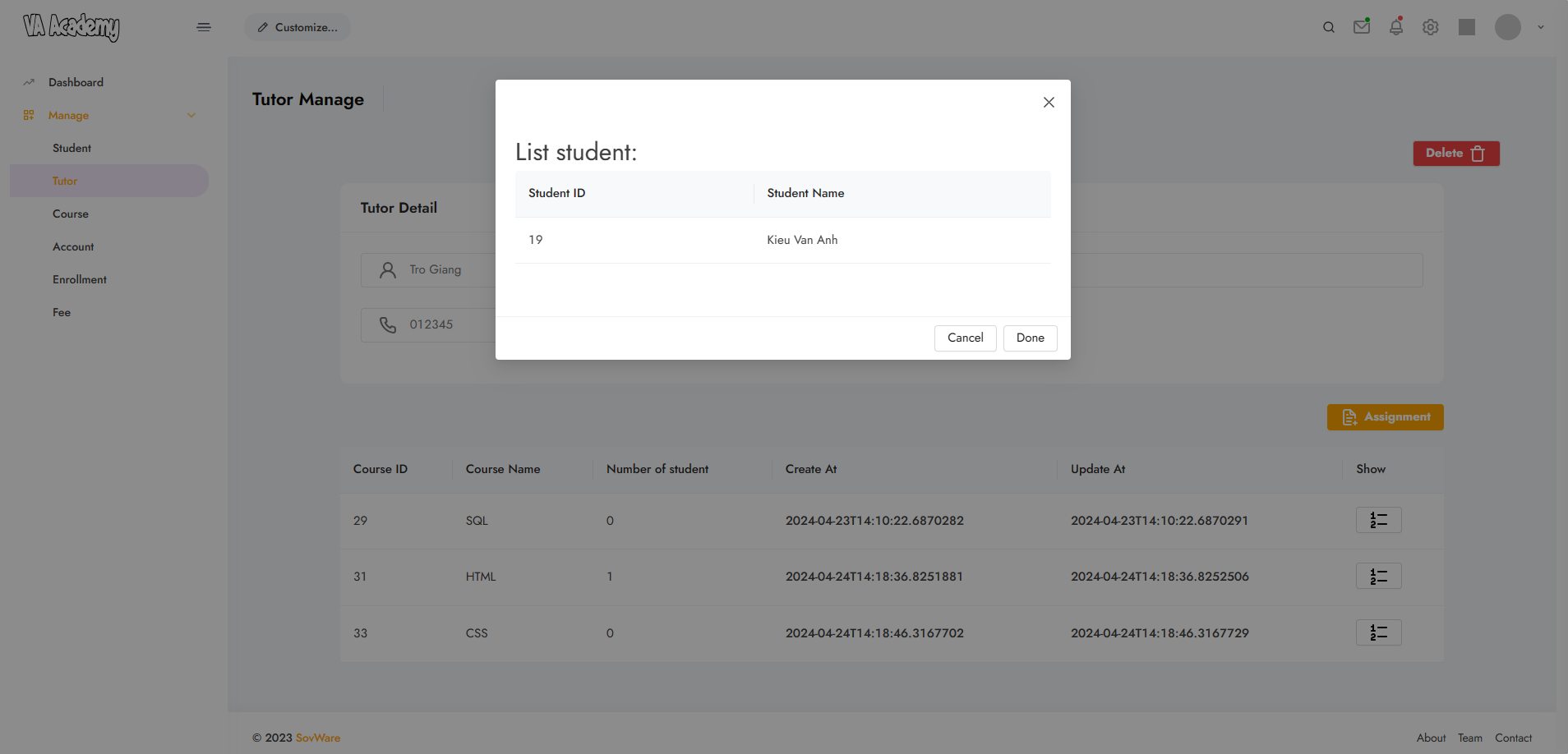
Hình 3.9. Giao diện trang Quản lý giảng viên



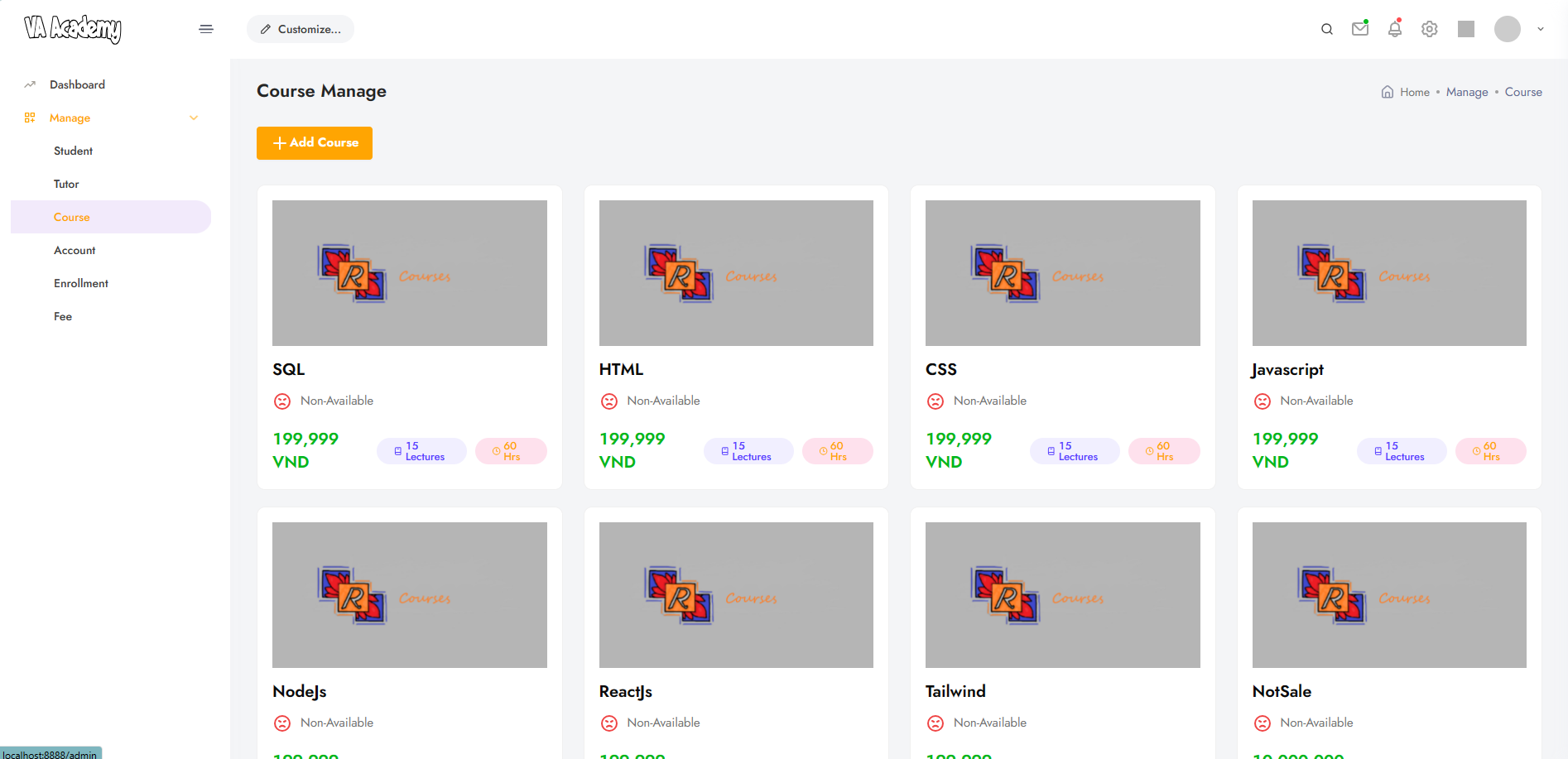
Hình 3.10. Giao diện trang Chi tiết giảng viên



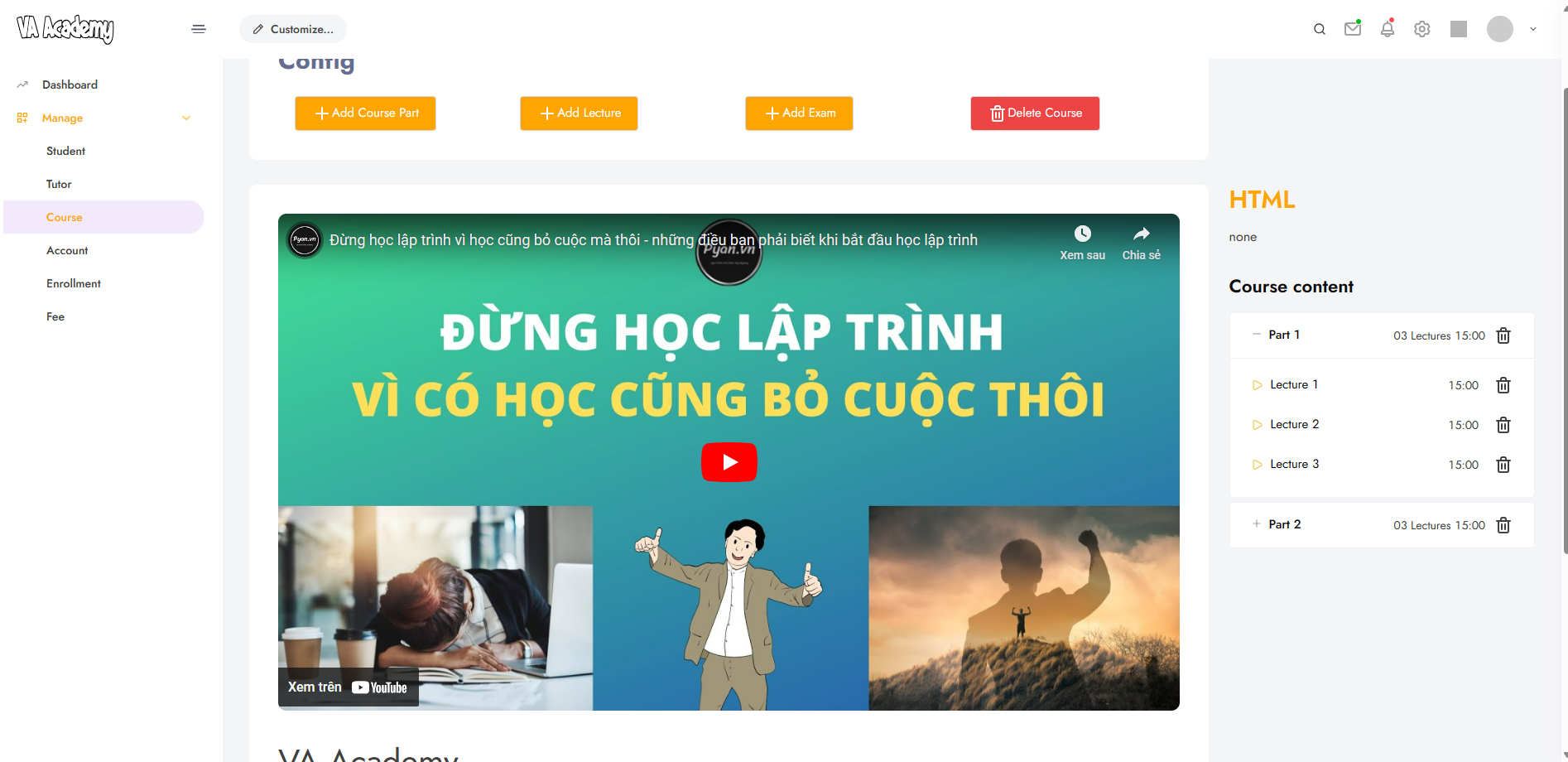
Hình 3.11. Giao diện trang Phân công khóa học



Hình 3.12. Giao diện trang Xem số học viên đăng kí khóa học

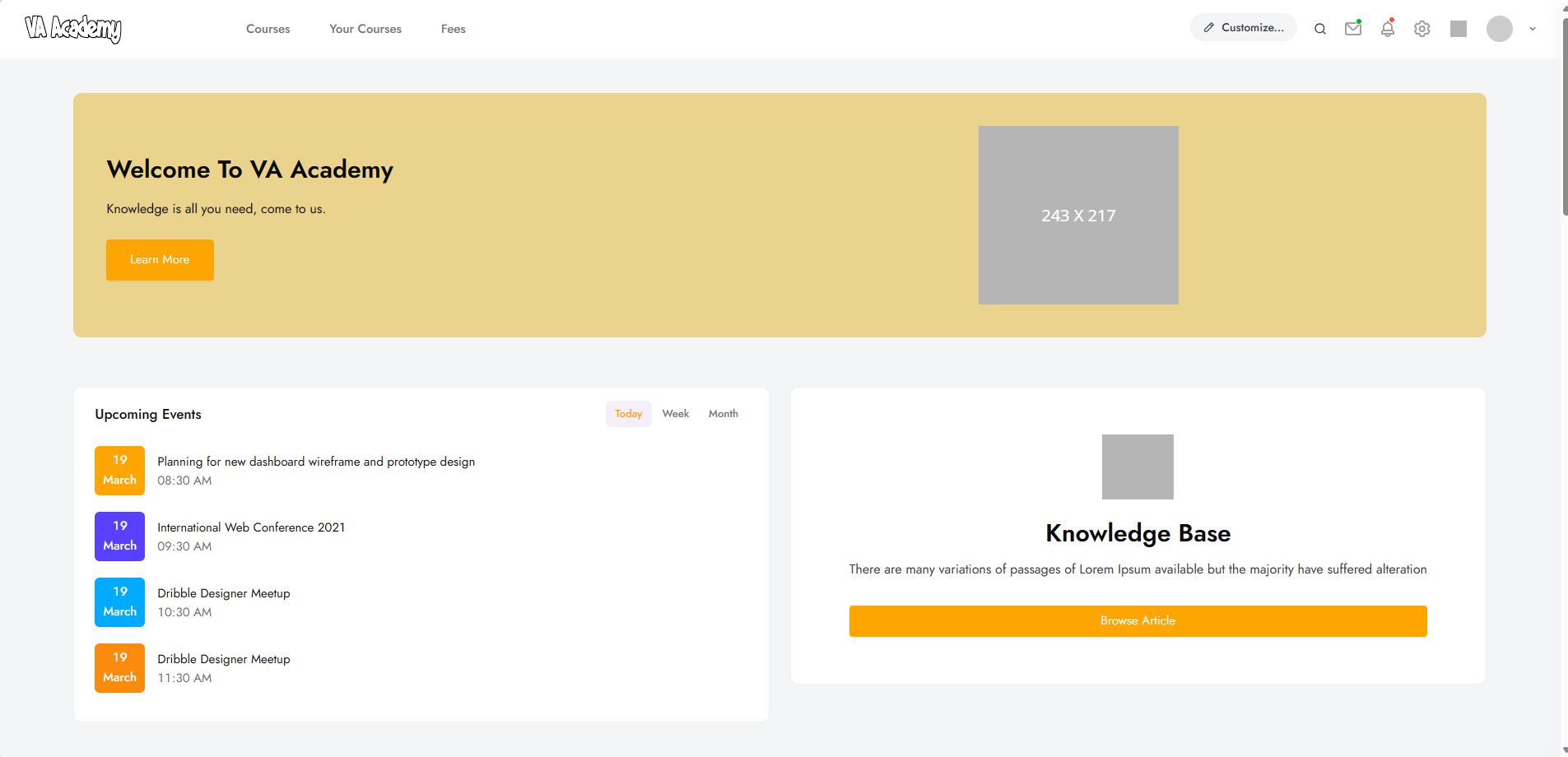


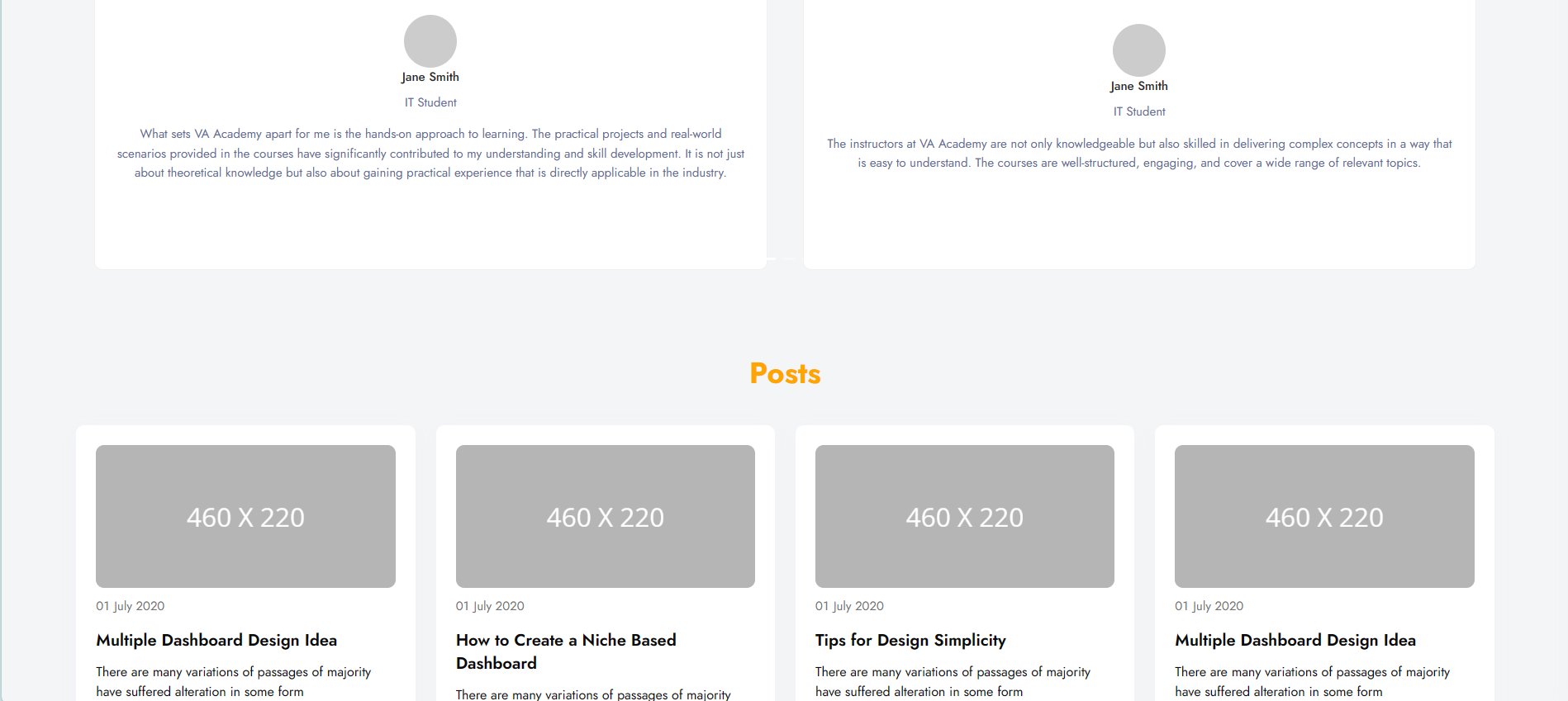
Hình 3.13. Giao diện trang Khóa học



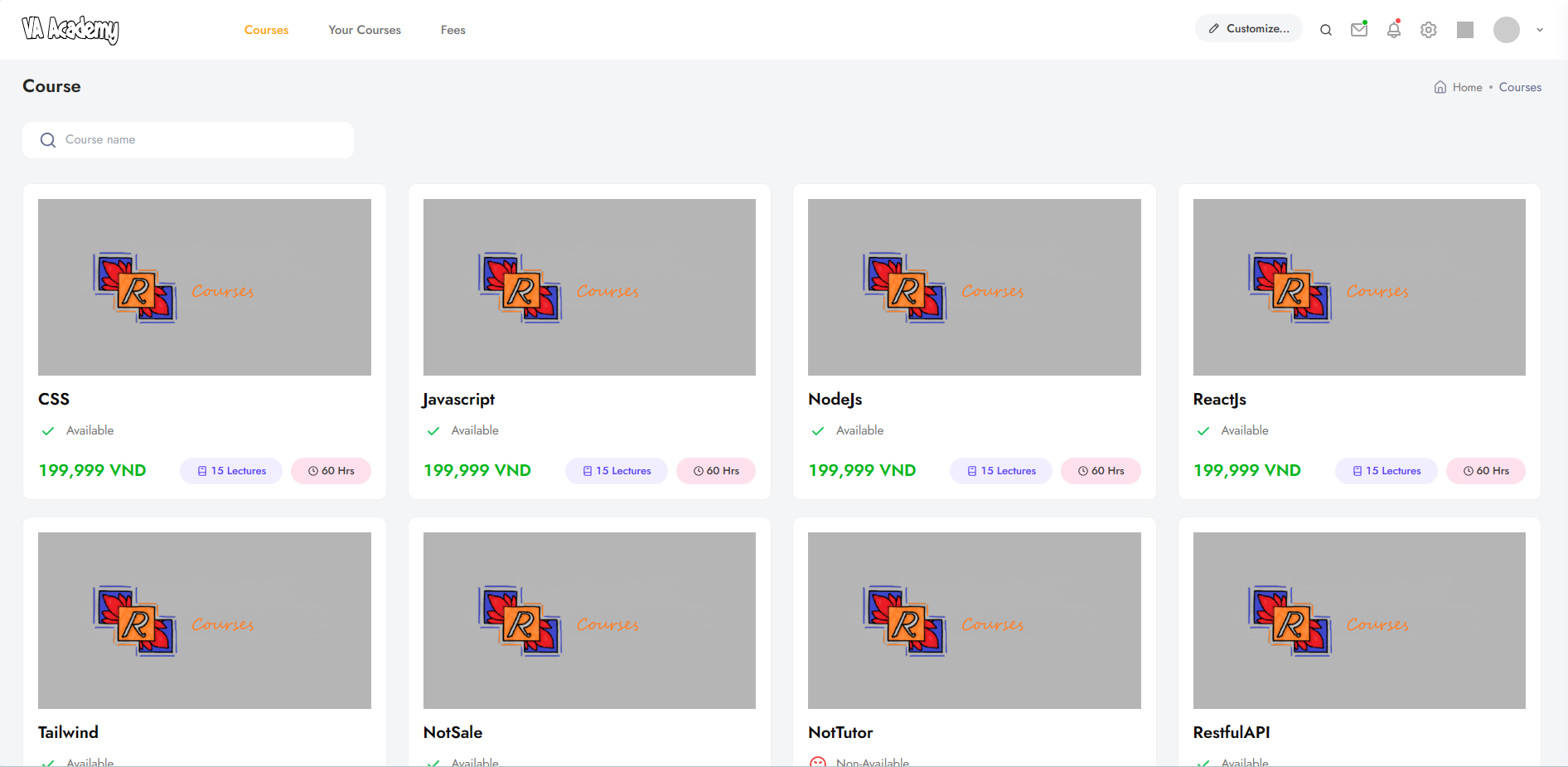
Hình 3.14. Giao diện trang Chi tiết khóa học

## Giao diện trang Student

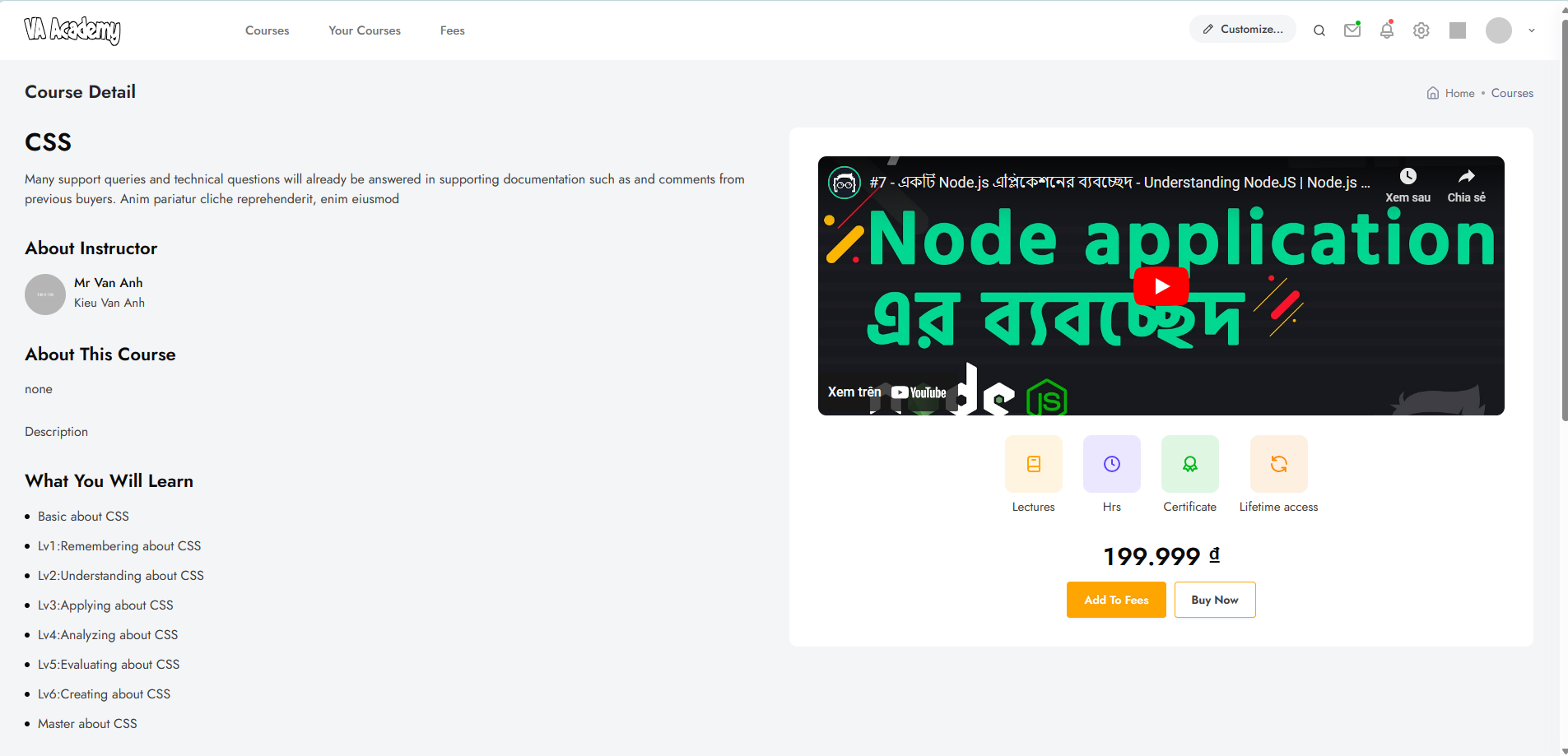
 



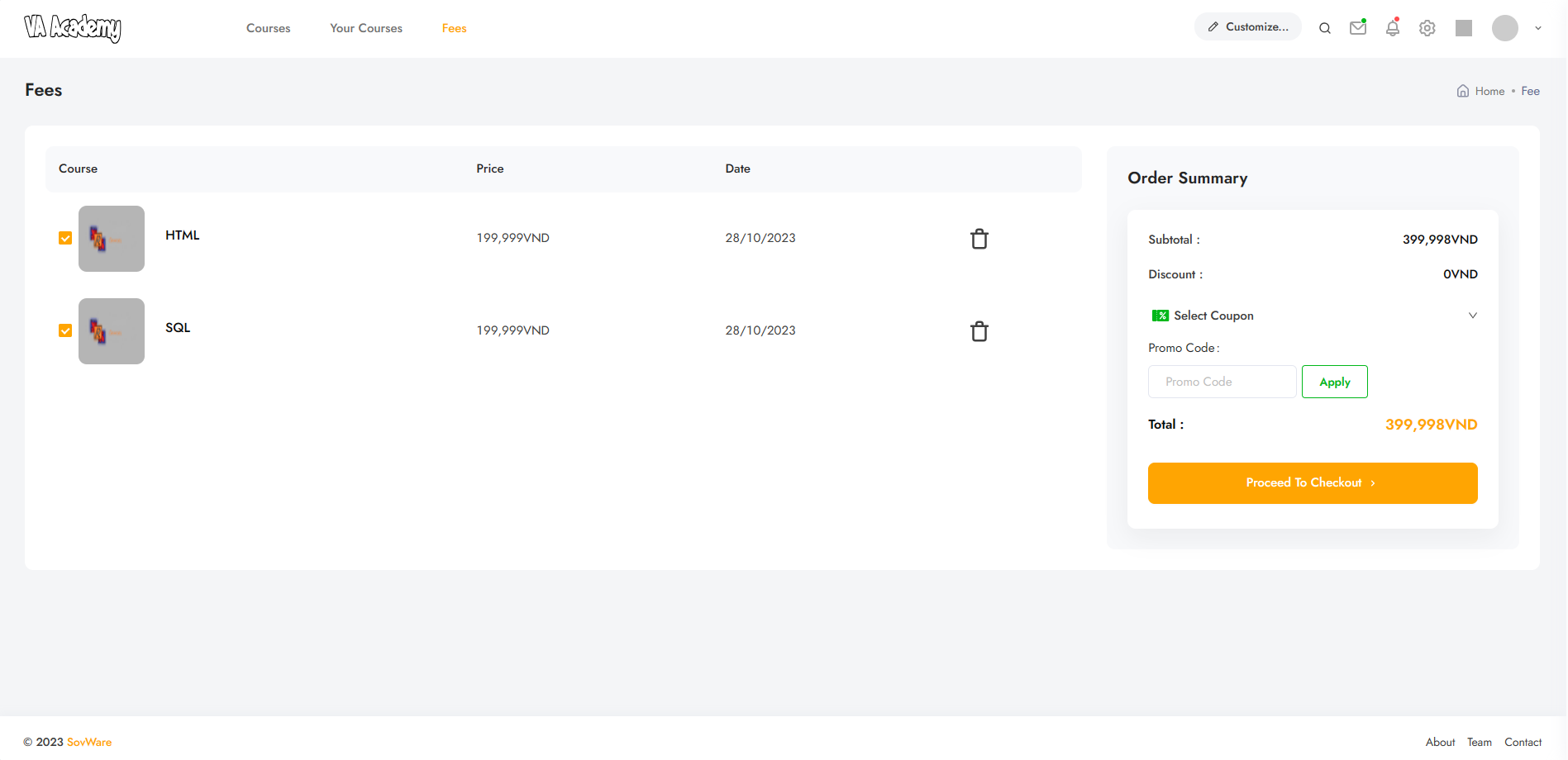
Hình 3.15. Giao diện trang chủ Student



Hình 3.16. Giao diện trang Khóa học



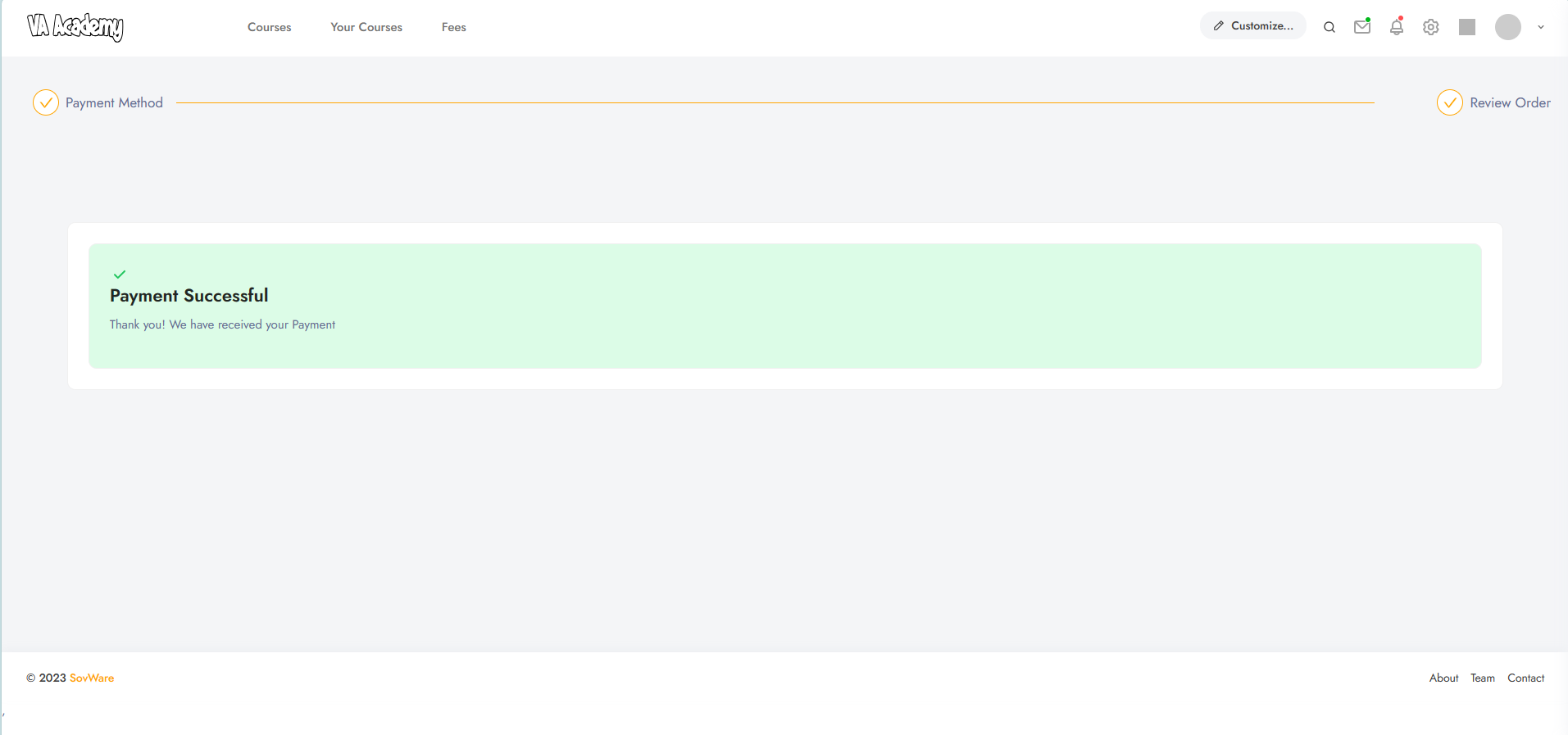
Hình 3.17. Giao diện trang Xem chi tiết khóa học



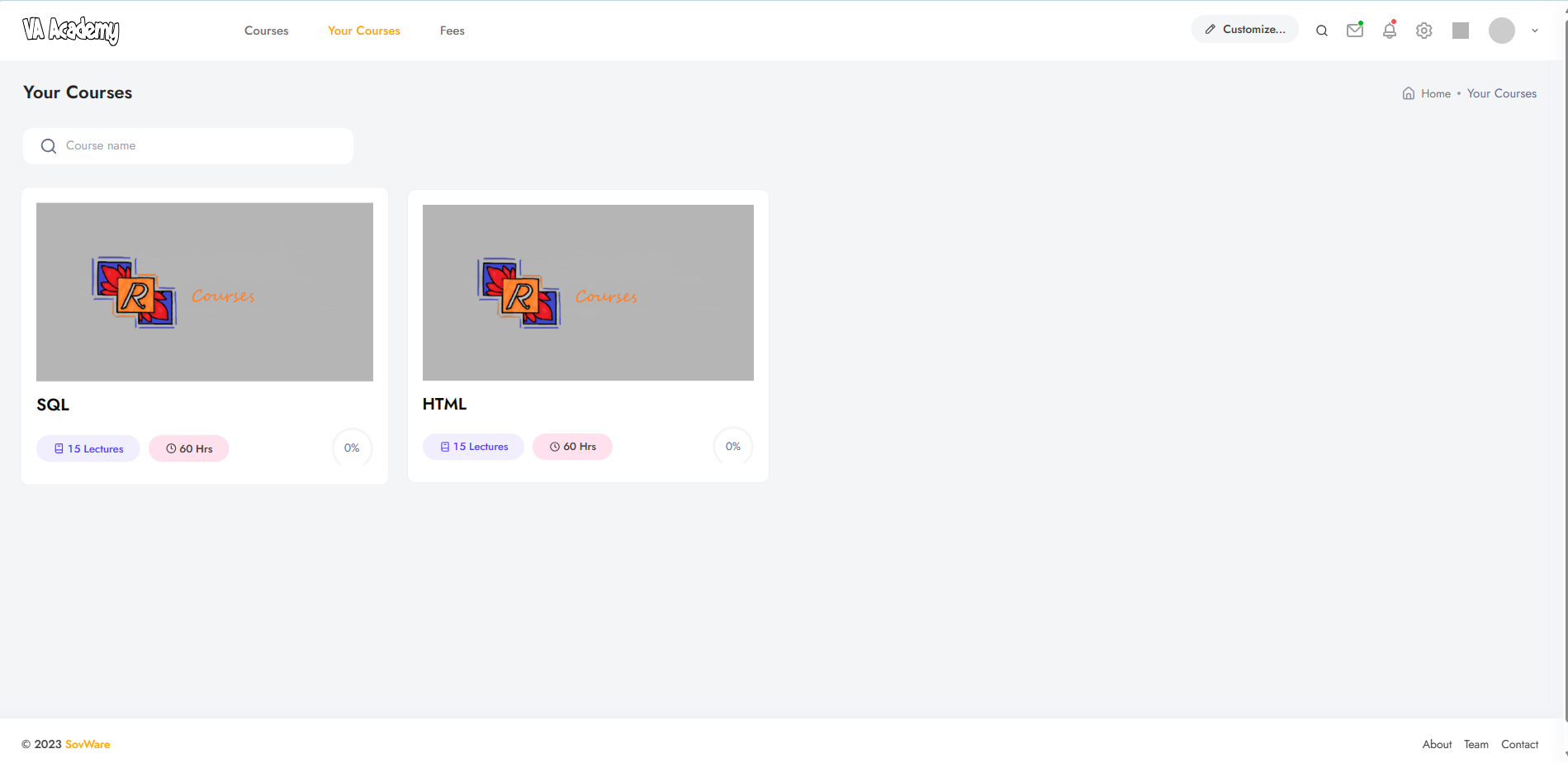
Hình 3.18. Giao diện Giỏ hàng



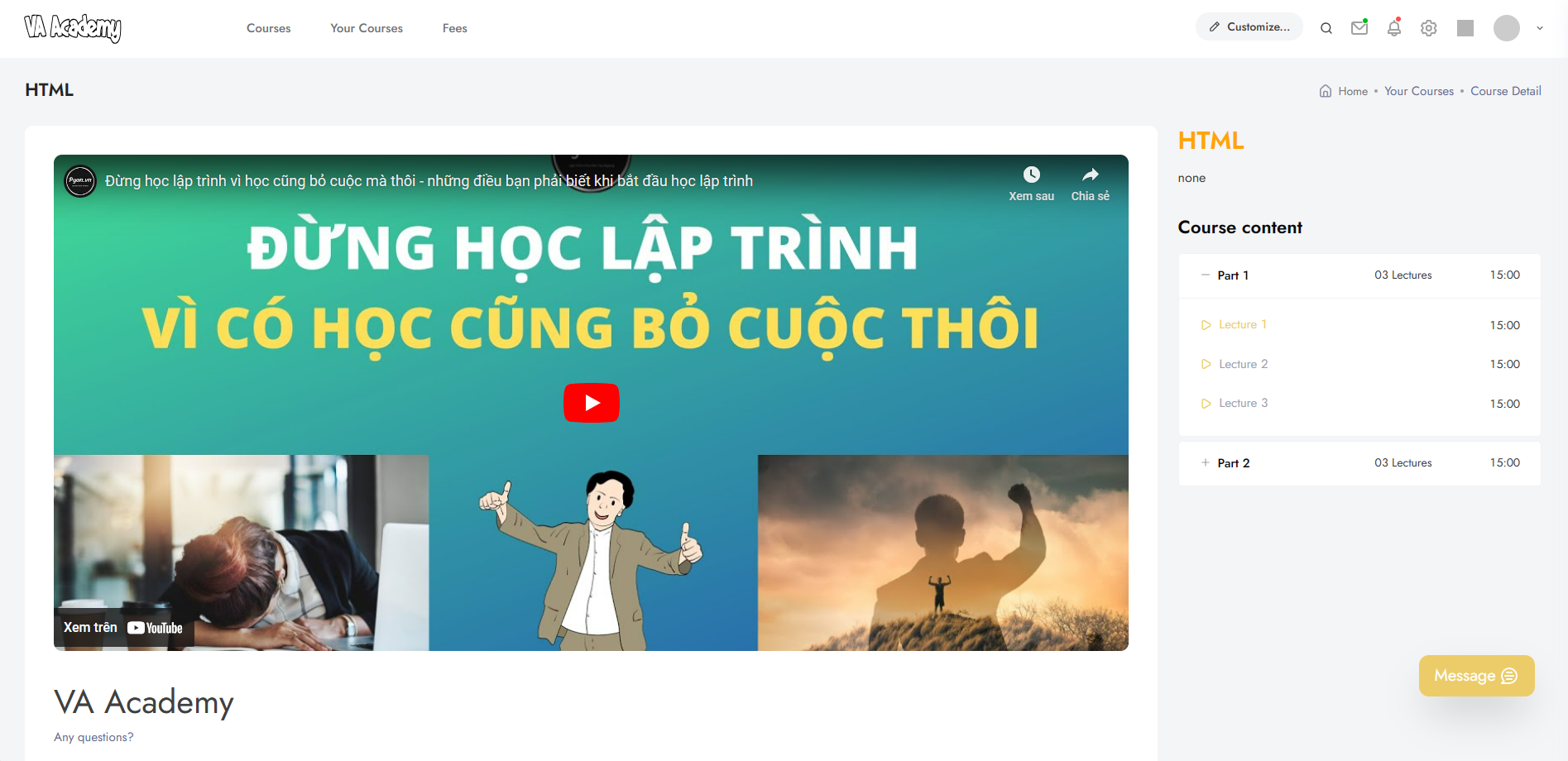
Hình 3.19. Giao diện thanh toán



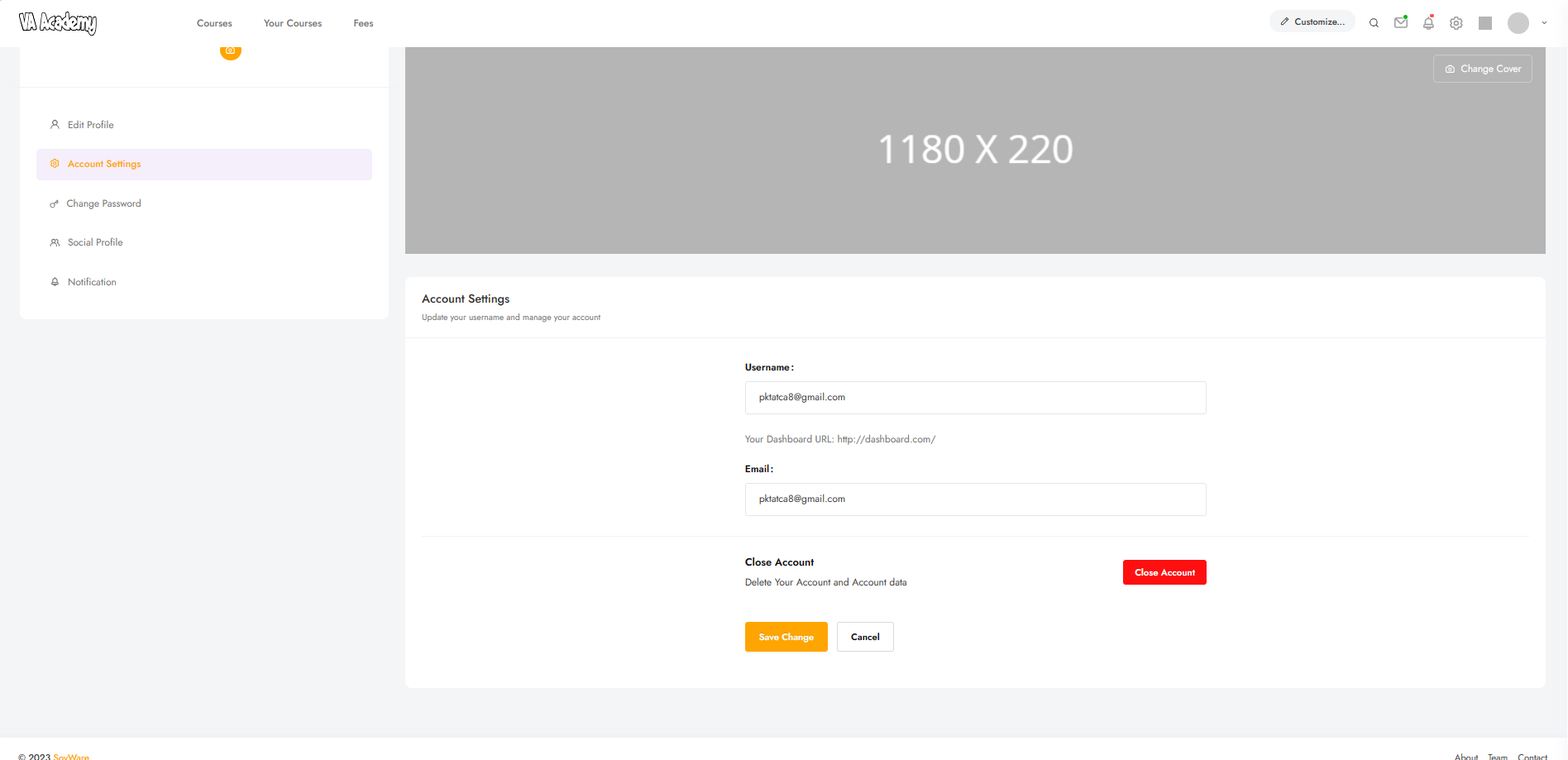
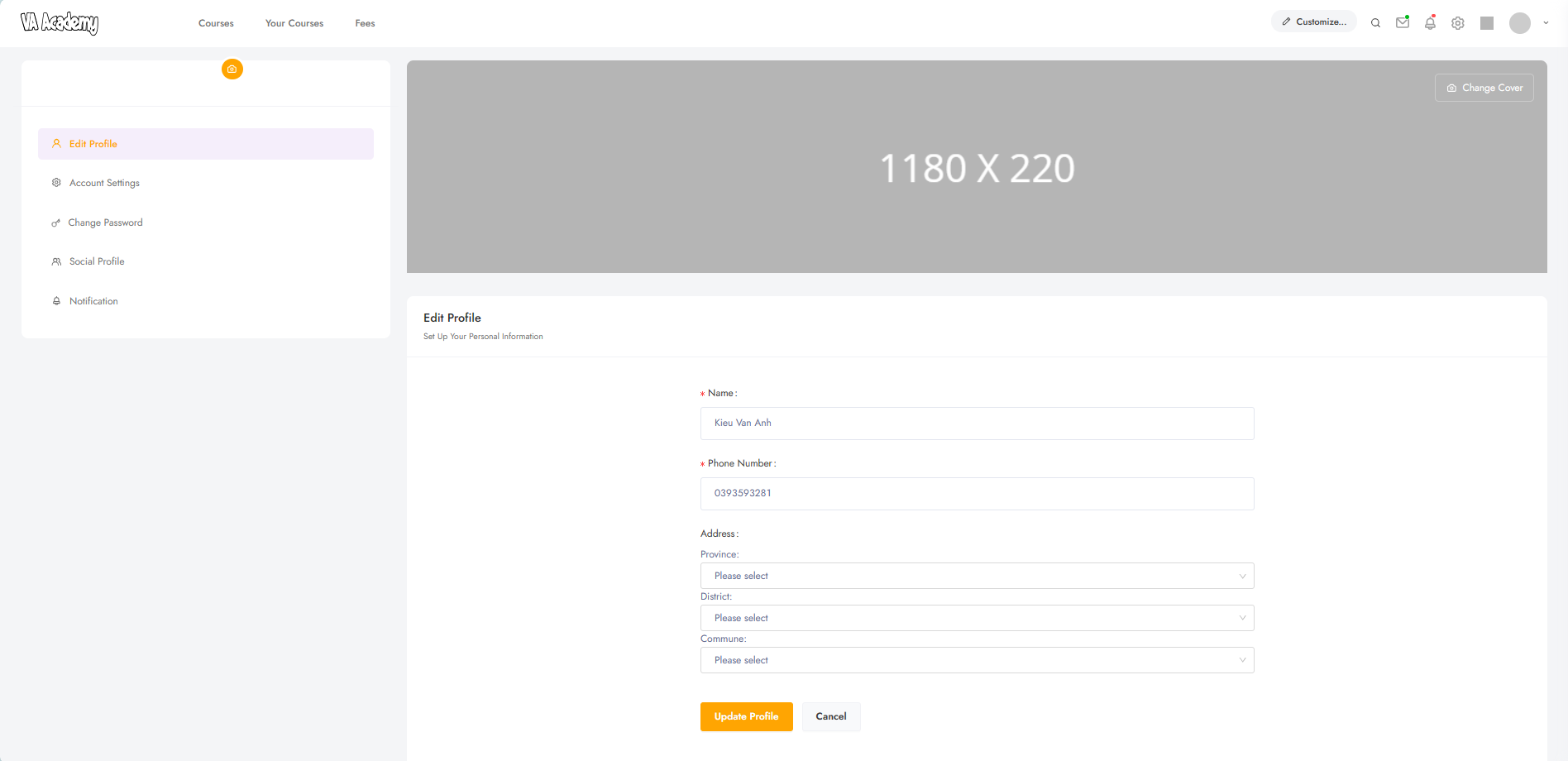
Hình 3.20. Giao diện thanh toán thành công



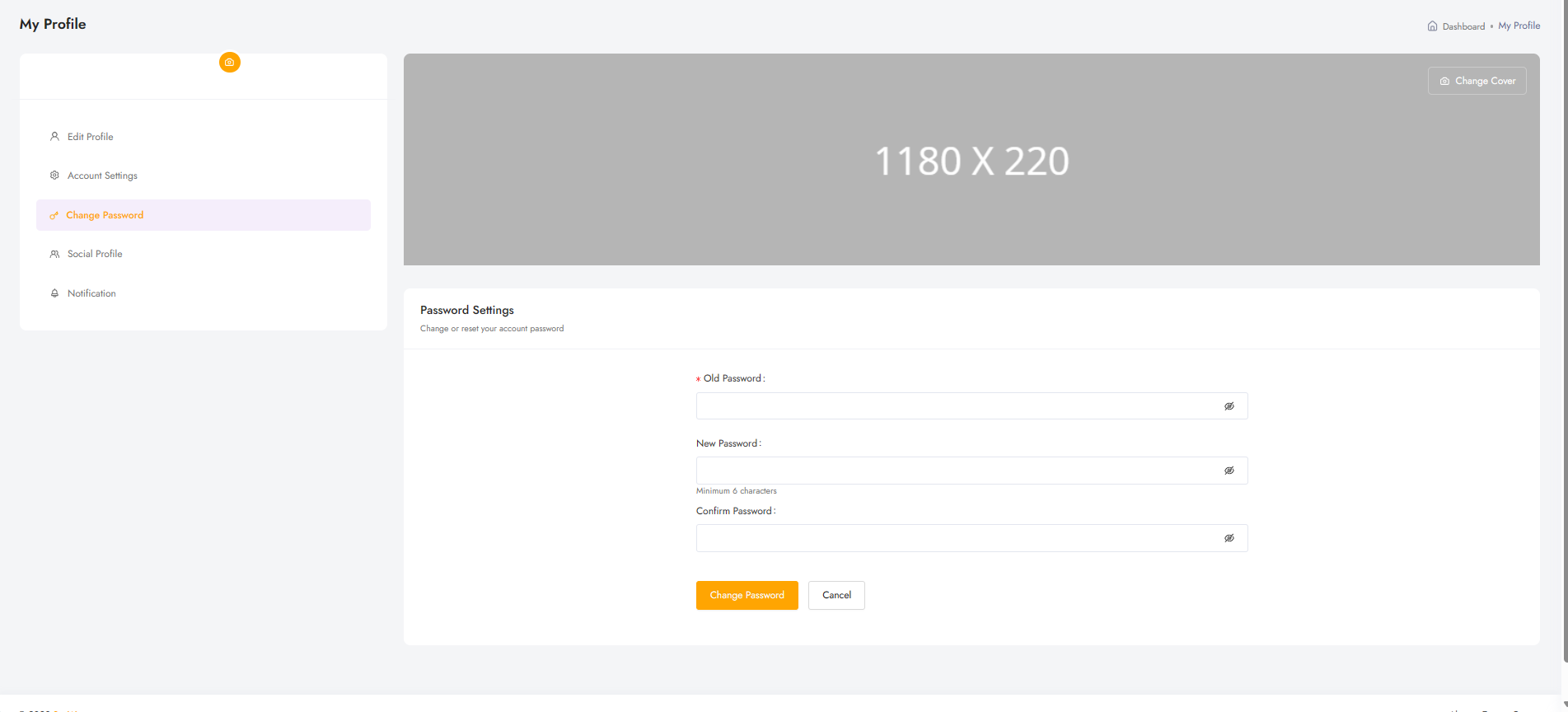
Hình 3.21. Giao diện trang Xem khóa học đã mua



Hình 3.22. Giao diện trang Vào học

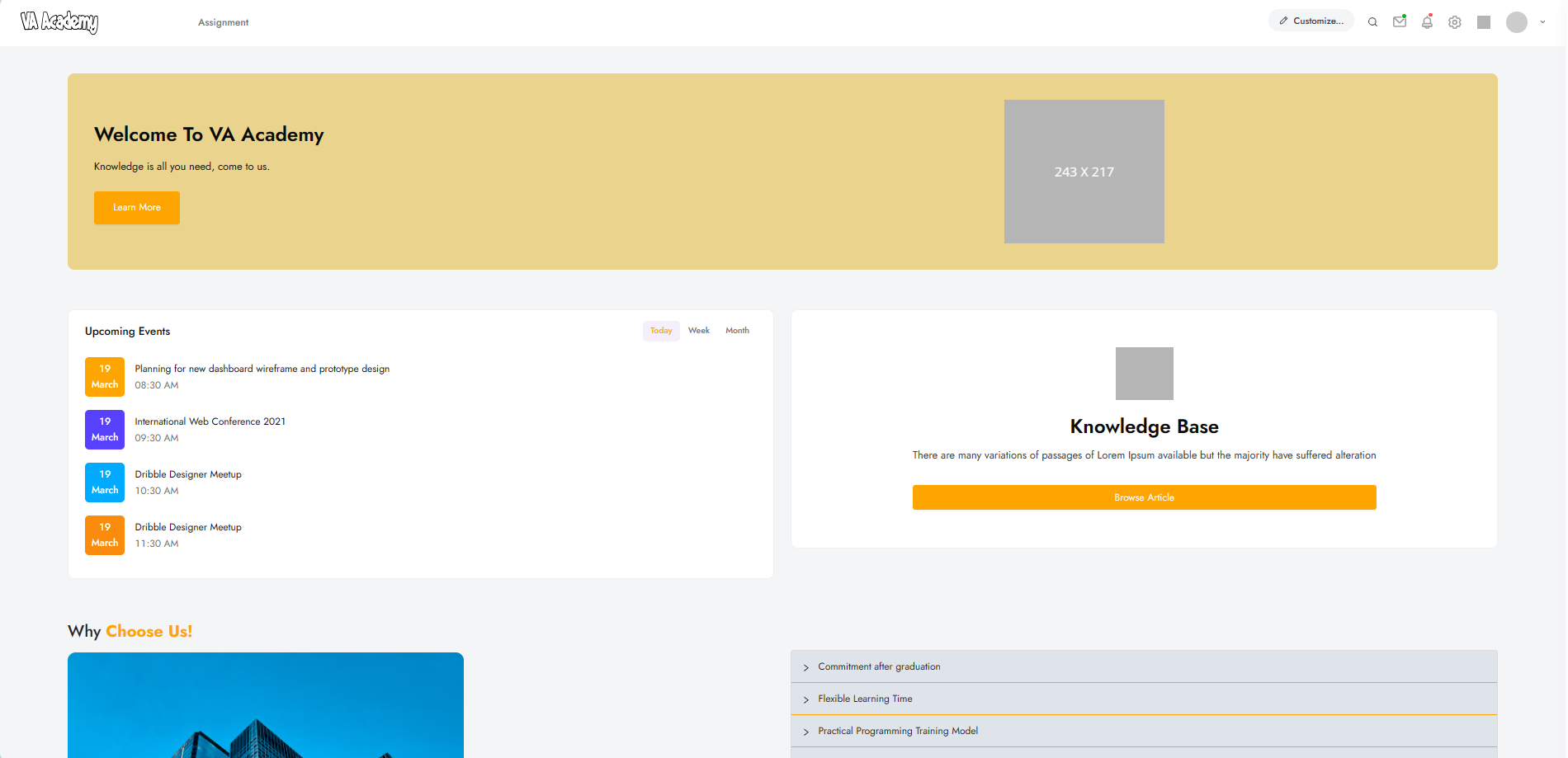


Hình 3.23. Giao diện trang Quản lý thông tin cá nhân

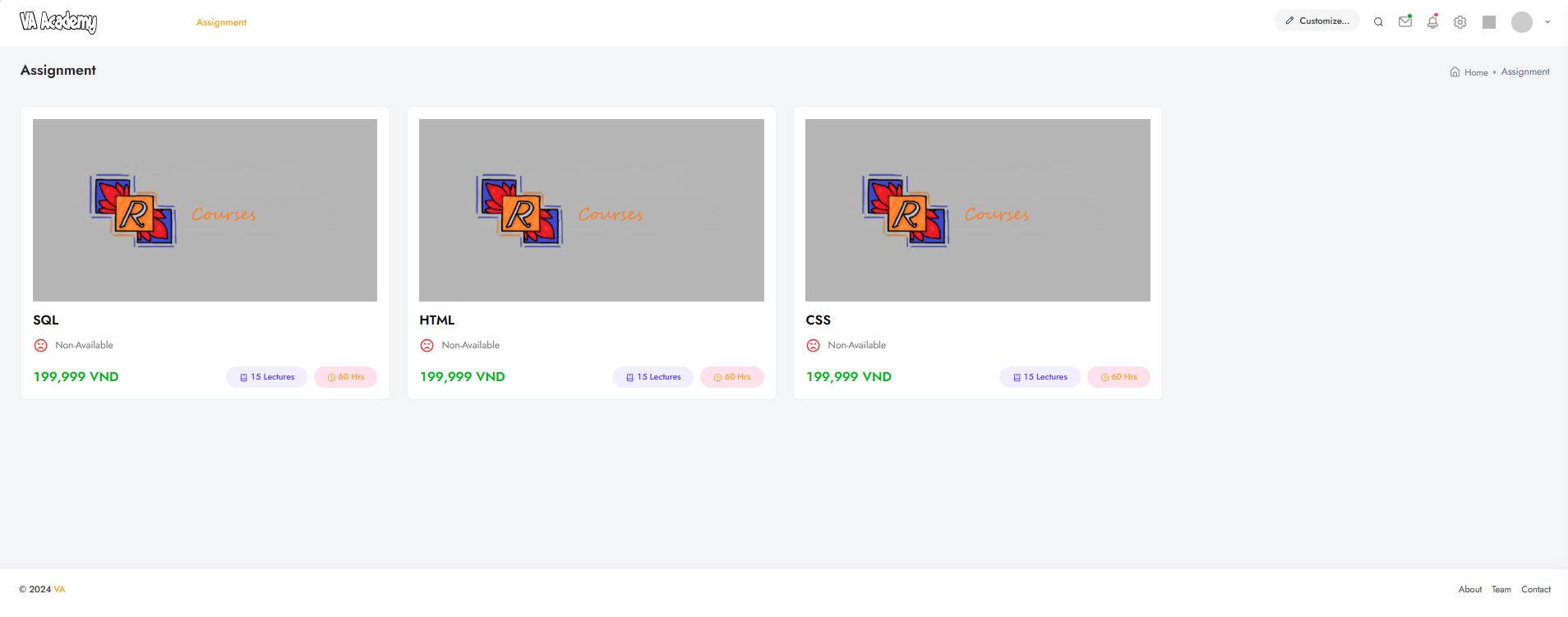


Hình 3.24. Giao diện trang Đổi mật khẩu

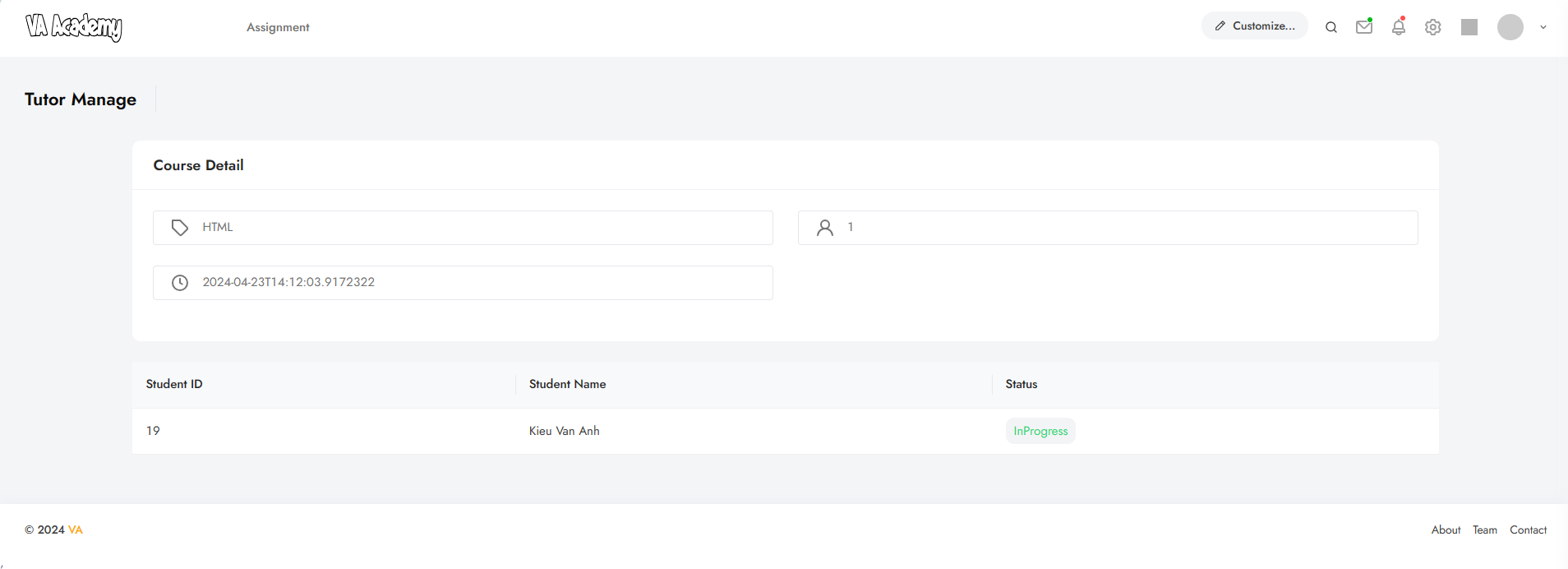
## Giao diện trang Tutor



Hình 3.25. Giao diện trang Tutor



Hình 3.26. Giao diện trang Phân công



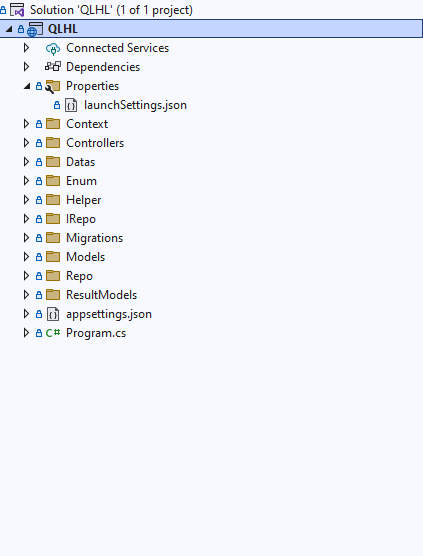
Hình 3.27. Giao diện trang Chi tiết phân công

# CÀI ĐẶT VÀ KIỂM THỬ

## API

1. Cấu trúc source code

Em đã lựa chọn Framework Web đó là Asp.net Core 6.0 với ngôn ngữ C# để làm phần backend viết API.



Hình 4.1. Cấu trúc source code phần API

Mô tả cấu trúc source code:

* Controllers: Là nơi chứa các controller thực hiện nhiệm vụ tương tác với request , xử lý và trả response cho frontend.
* Dependencies: Là nơi chứa các Packages, và các thành phần liên quan của ứng dụng của ứng dụng.
* Context: Là nơi chứa các dòng khai báo mô hình dữ liệu để liên kết với CSDL.
* Datas: Là nơi chứa các class làm việc với EF Core để làm việc với cơ sở dữ liệu.
* Enum: Là nơi chứa các class dạng Enum.
* Helpers: Là nơi chứa các class hỗ trợ cho Project.
* Migrations: Là nơi chứa các Migrations khi kết nối với CSDL MSSQL theo dạng code-first.
* Models: Là nơi chứa các Entities.
* IRepo: Là nơi chứa các class Interface của Service.
* Repo: Là nơi chứa các Service của Website.
* ResultModel: Là nơi chứa các class Model của dữ liệu trả về.
* appsettings: Là nơi cấu hình cho project.

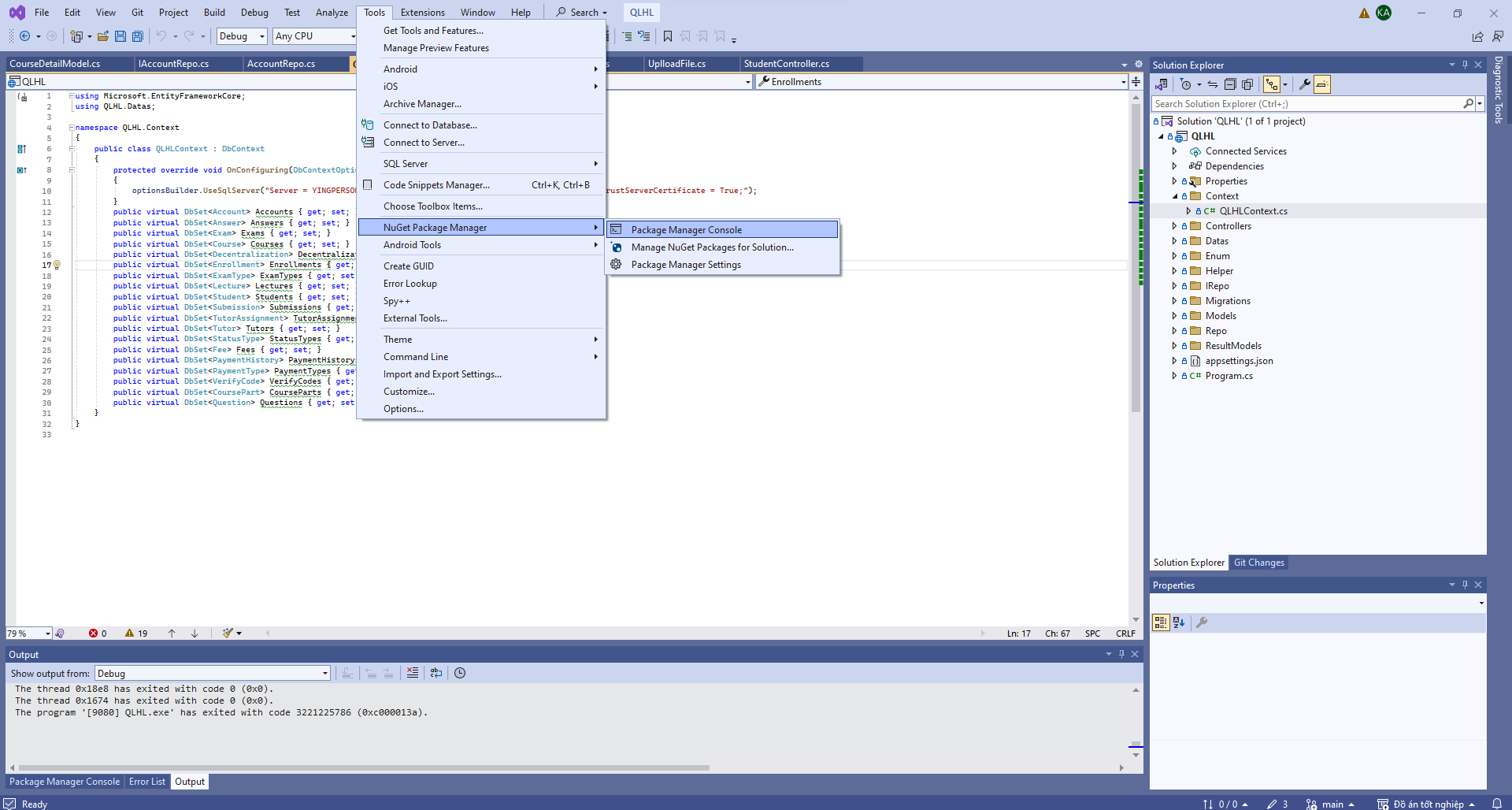
1. Tạo CSDL bằng Code first với Migrations

Từ phần phân tích thiết kế vừa xây dựng ta tiến hành tạo các class cho các lớp tương ứng.

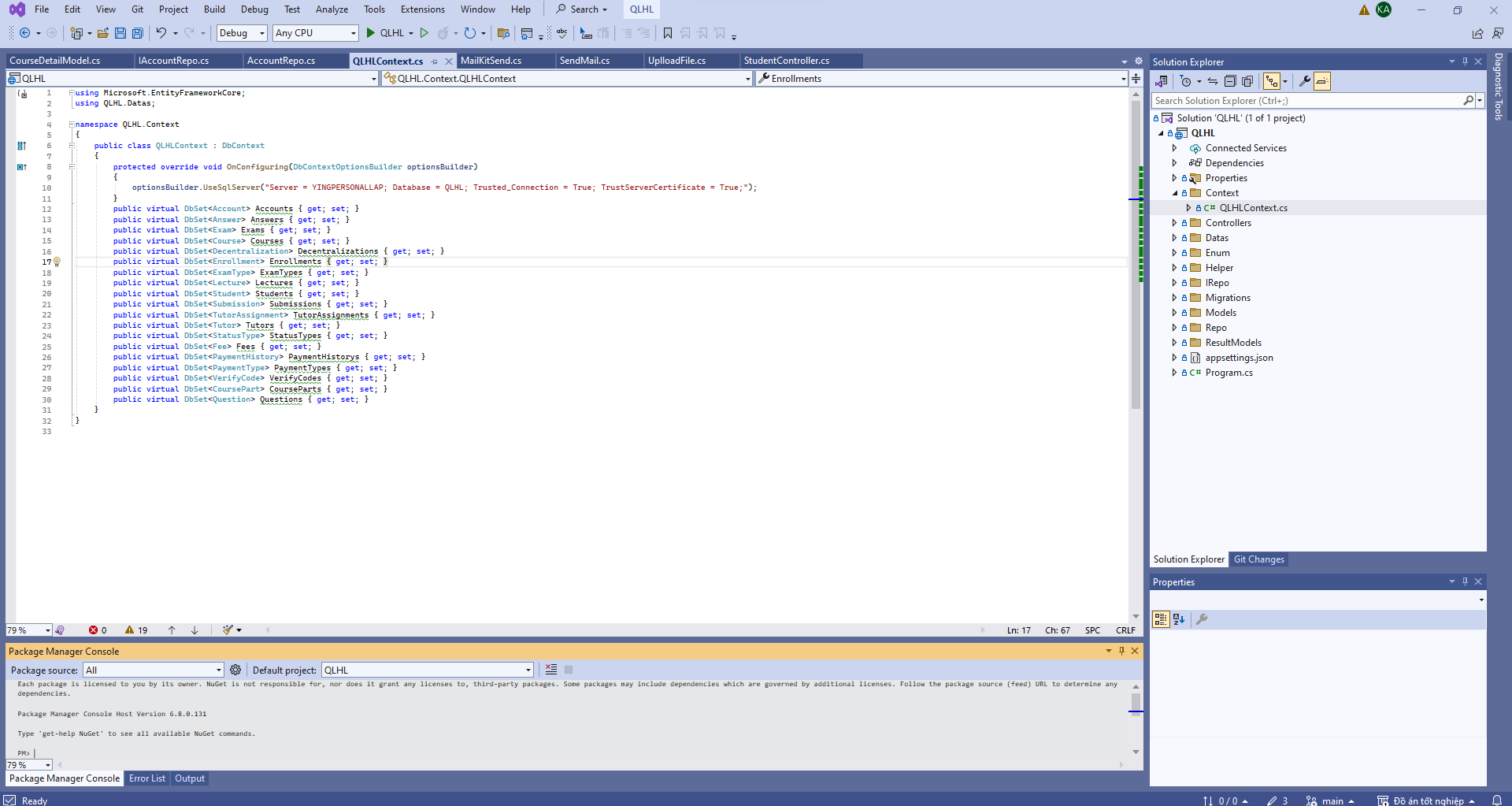
Từ các models vừa tạo, ta sẽ cài đặt các thư viện cần thiết, cấu hình chuỗi kết nối, đăng kí nó trong service và tiến hành tạo lớp DBContext để tiến hành tạo CSDL.

Em đã sử sử dụng Entity Framework của .Net hỗ trợ cho việc cấu hình các bảng liên quan đến người dùng.

Sau đó, ta bật Package Manager Console để tạo Migrations kết nối với MSSQL. Chọn Tools -> NuGet Package Manager -> Package Manager Console.



Hình 4.2. Bật Package Manager Console



Hình 4.3. Package Manager Console

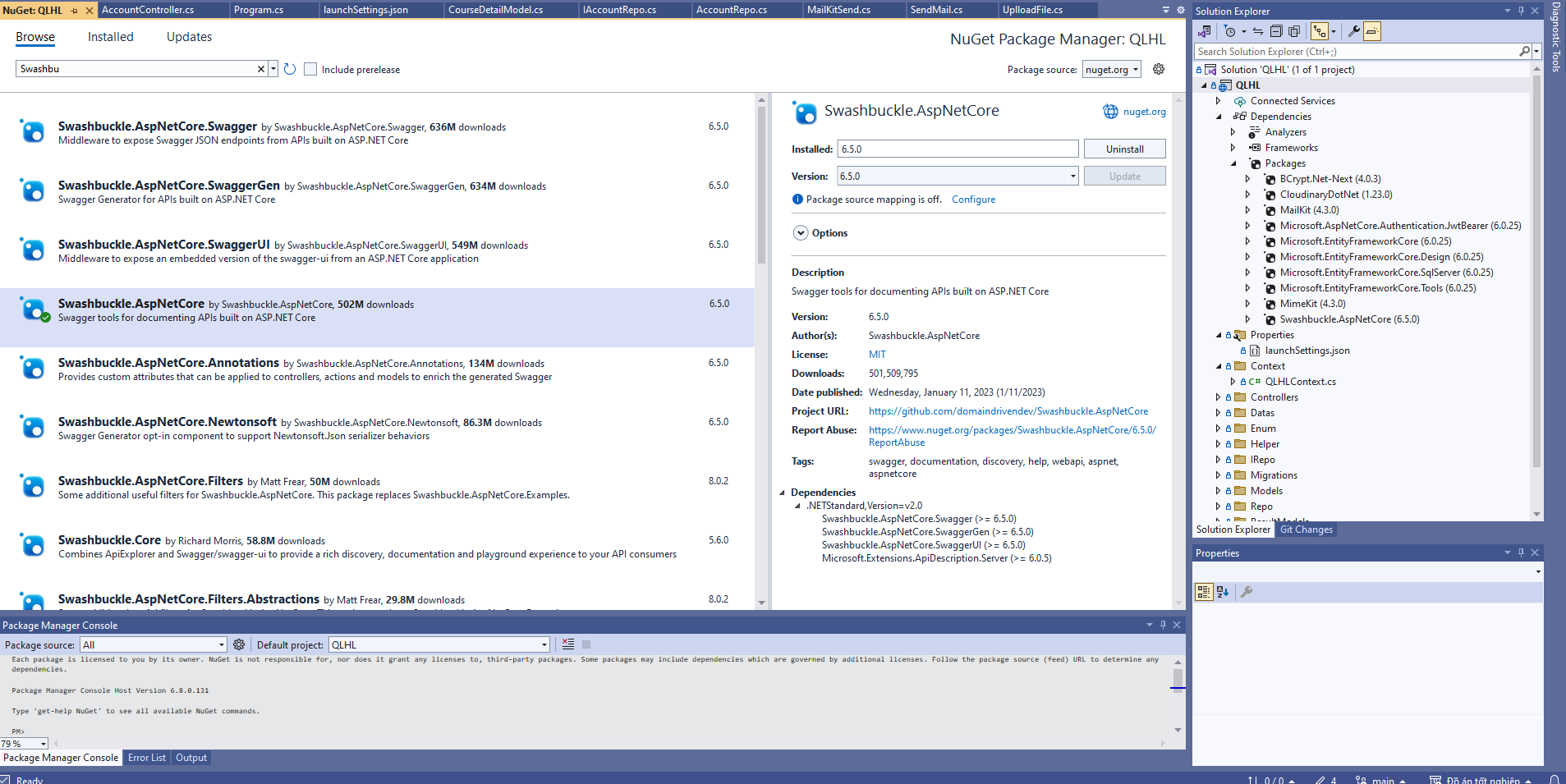
Đầu tiên, ta ghi lệnh “Enable-Migrations”, rồi đến “Add-Migration” và “Update-Database”, ta được CSDL.

1. Viết API

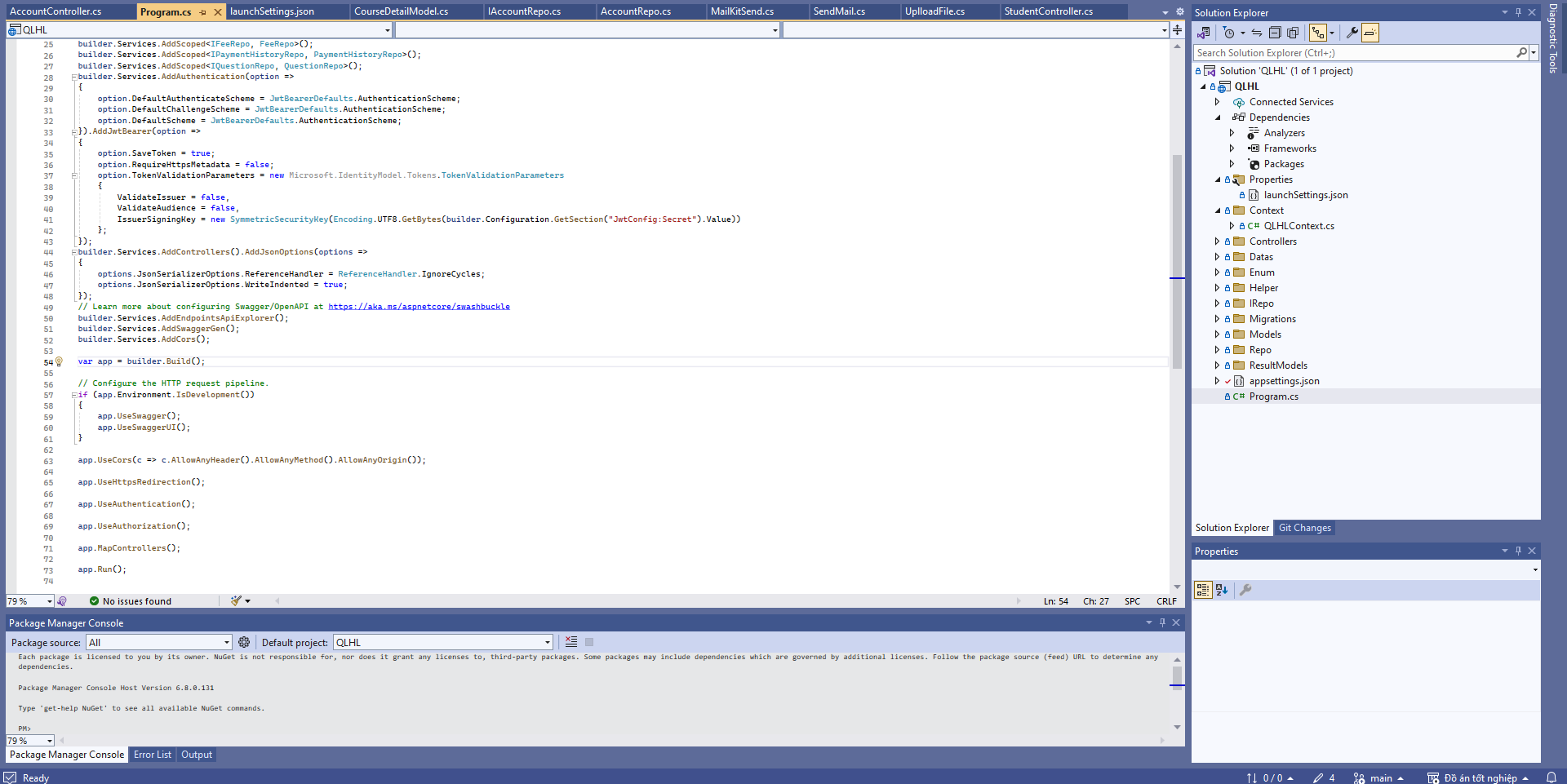
Ở Controllers, ta sẽ khởi tạo các lớp Controller cho các Model bằng phương thức khởi tạo. Ở mỗi hàm, tùy vào yêu cầu chức năng của trang web muốn hiển thị những gì mà ta viết các hàm tương ứng. Vì là giao tiếp với ứng dụng front-end nên ta sẽ sử dụng từ khóa async của .Net để biến hàm của chúng ta trở thành bất đồng bộ.

Về phần truy vấn dữ liệu, ta sẽ dùng LINQ để truy vấn.

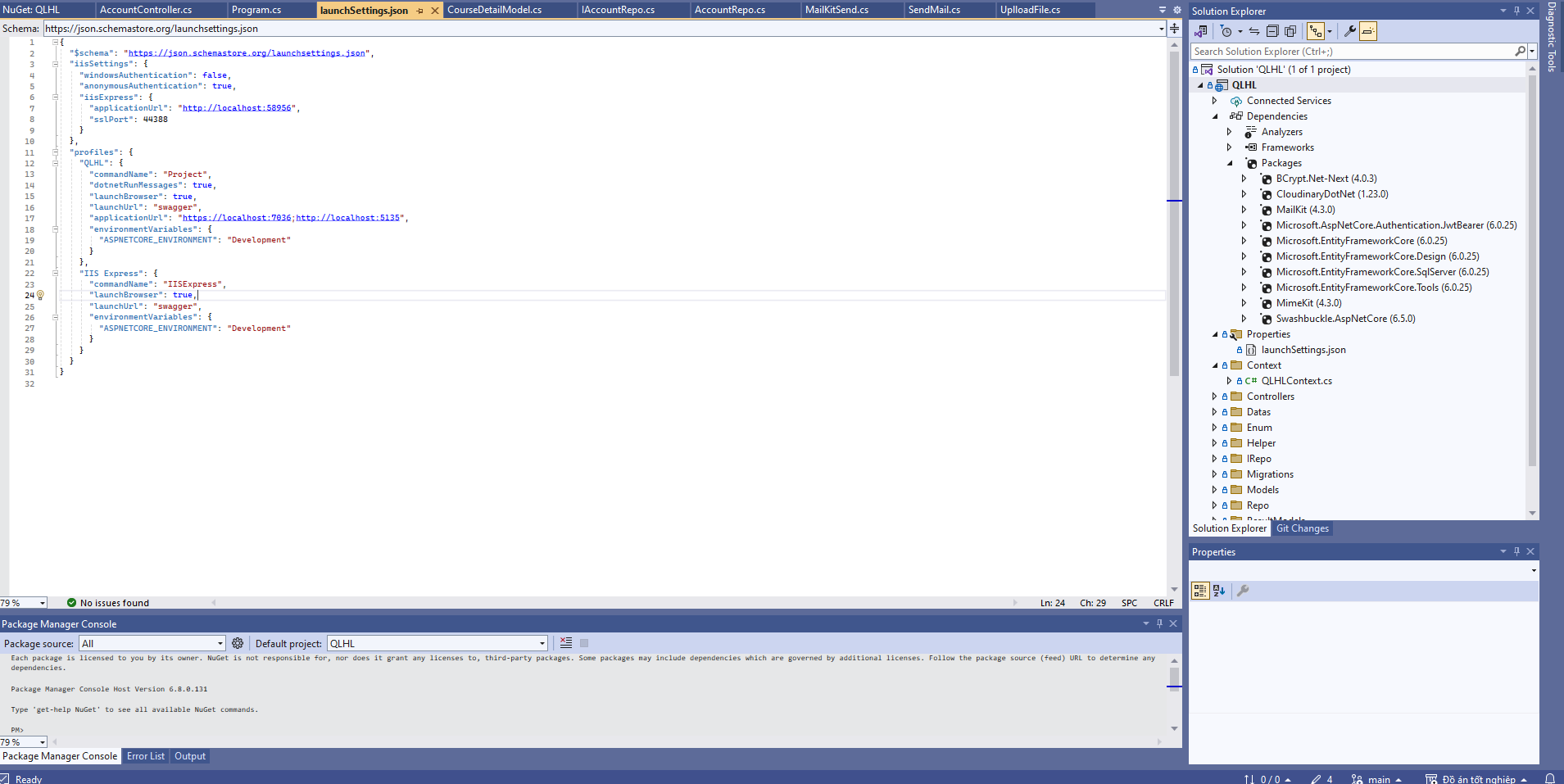
Về phần hiển thị API thì ta sử dụng package Swashbuckle.AspNetCore để sử dụng Swagger UI vào API.



Hình 4.4. Package Swashbuckle.AspNetCore

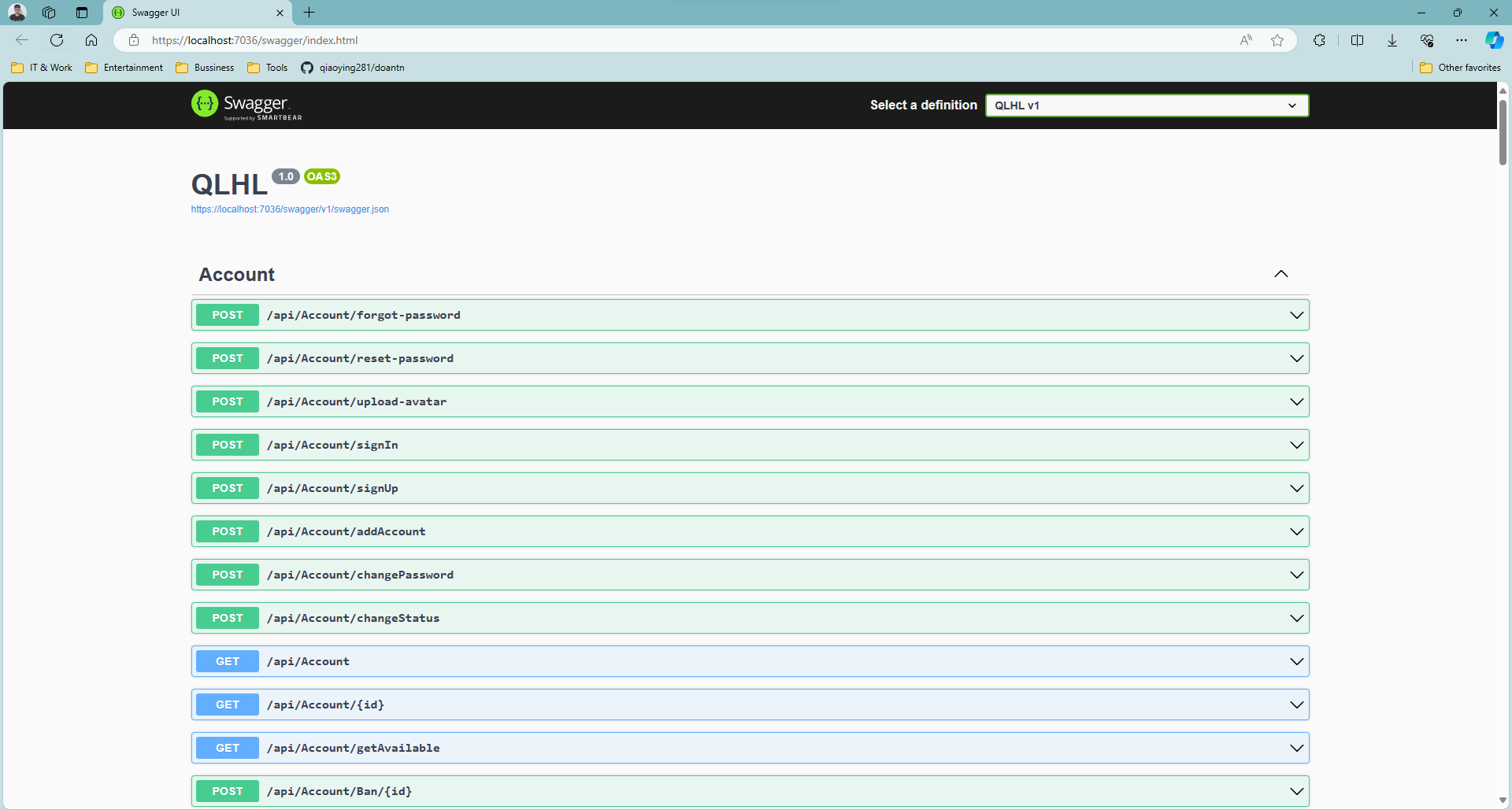


Hình 4.5. Code cài đặt Swagger trong Program.cs

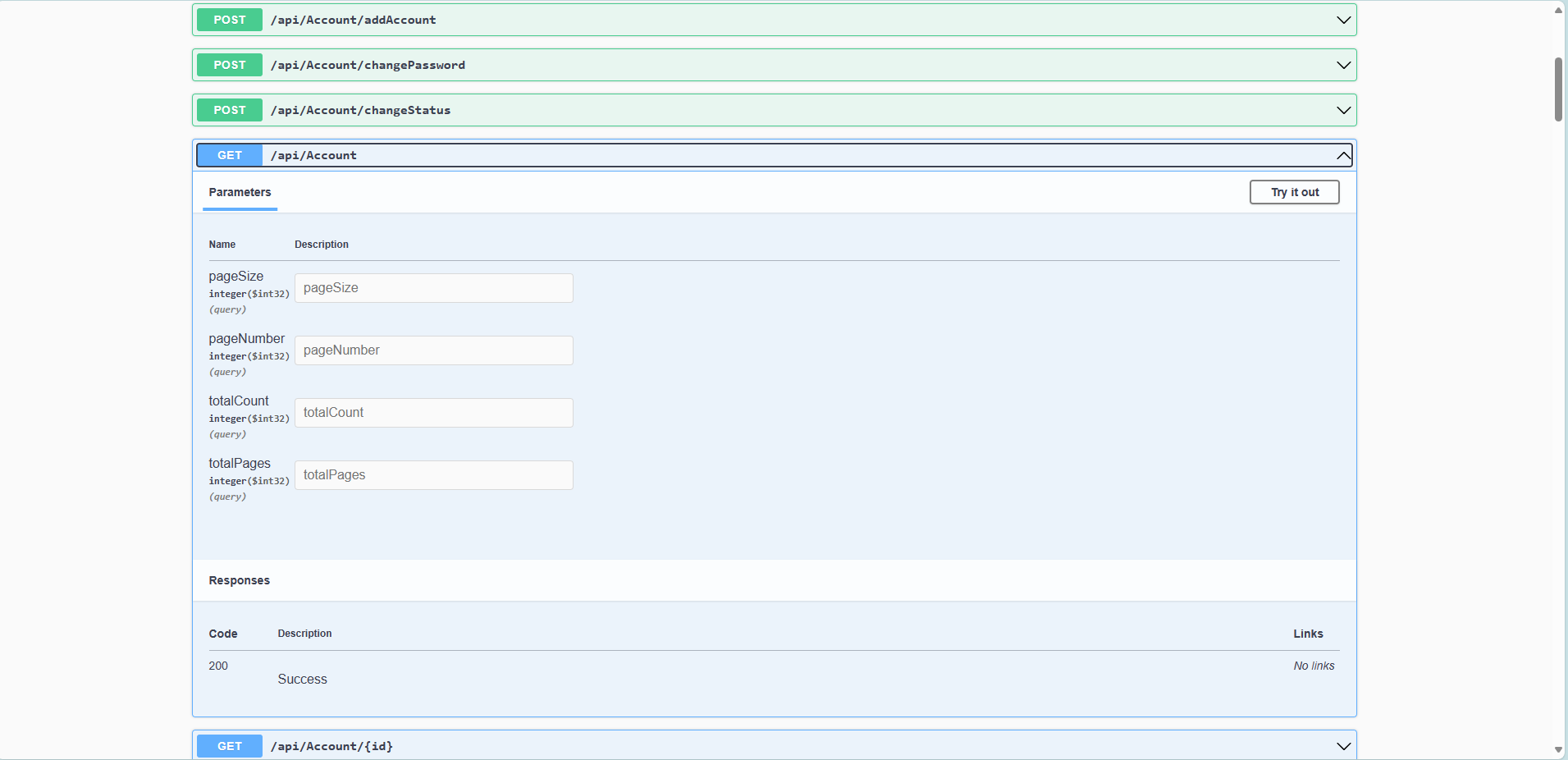


Hình 4.6. Cài đặt cấu hình chạy API

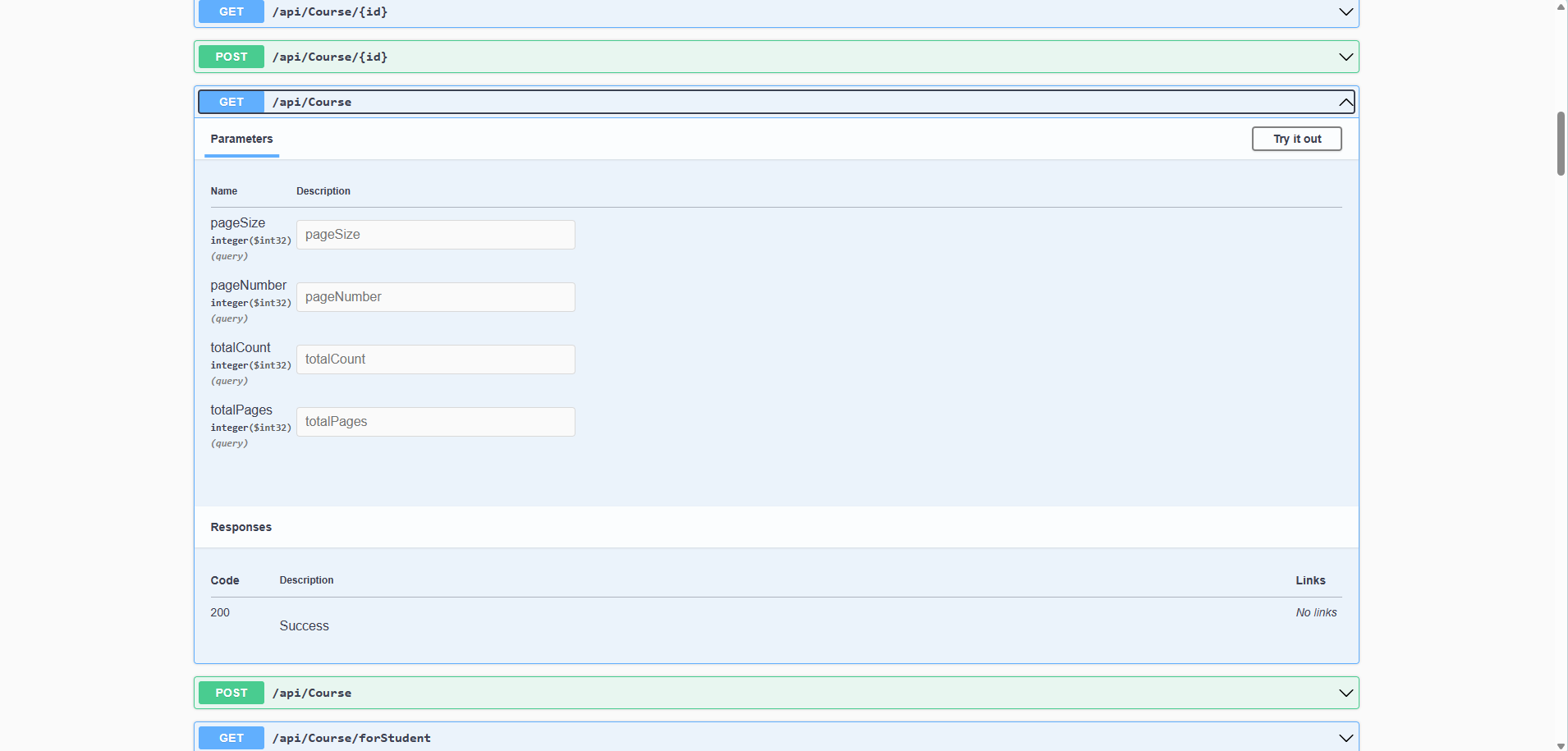
1. Demo kết quả API



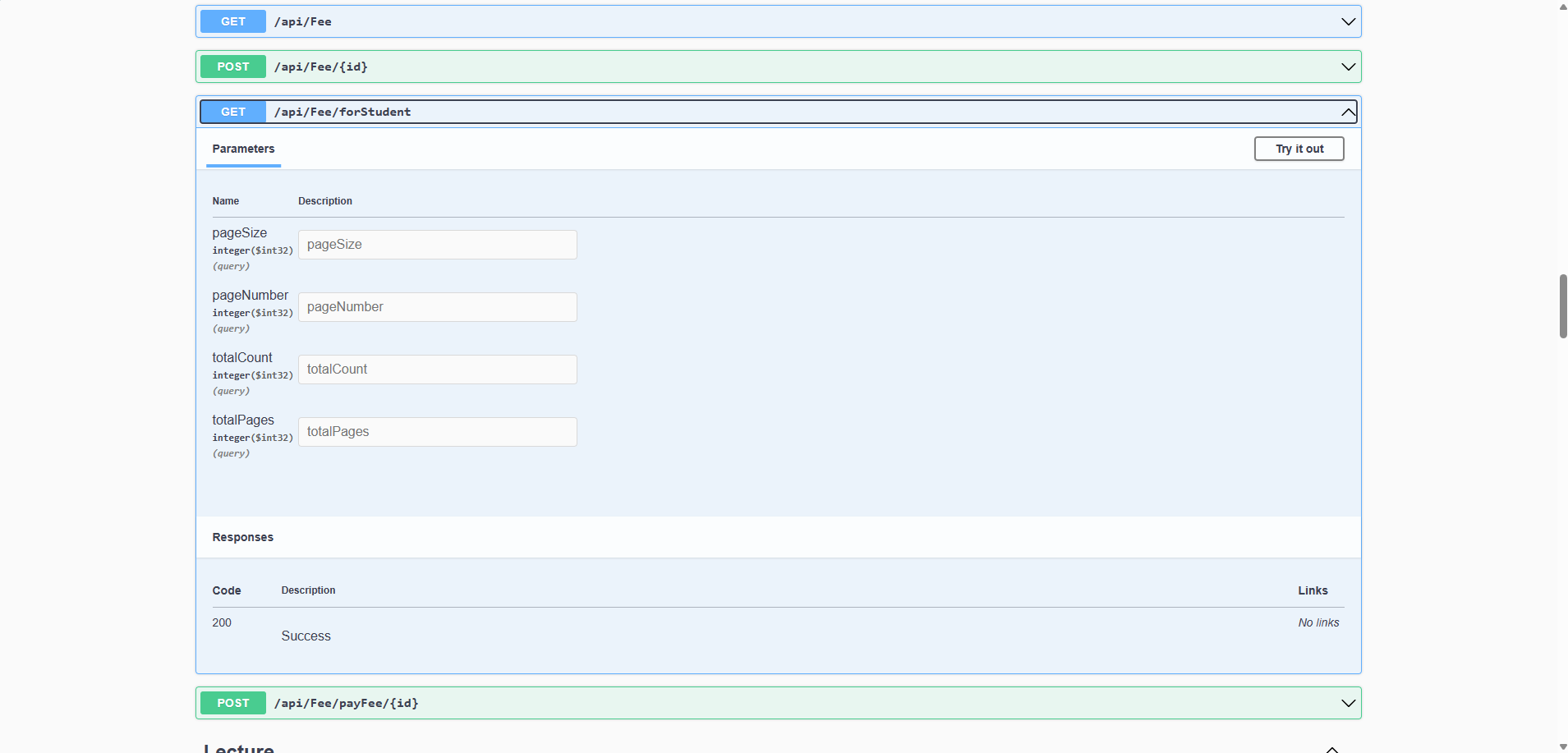
Hình 4.7. Giao diện API website



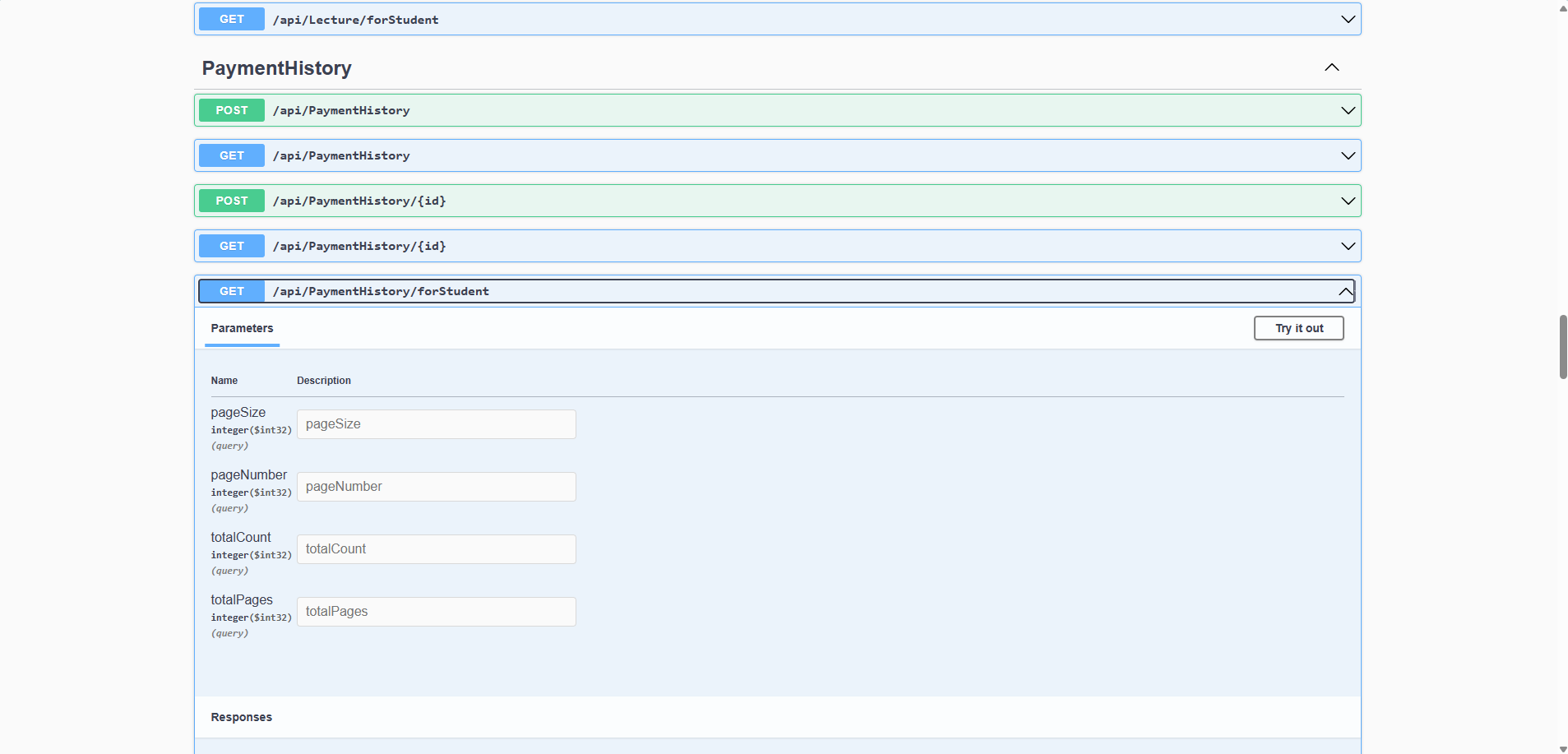
Hình 4.8. Truy vấn lấy danh sách tài khoản



Hình 4.9. Truy vấn lấy danh sách khóa học



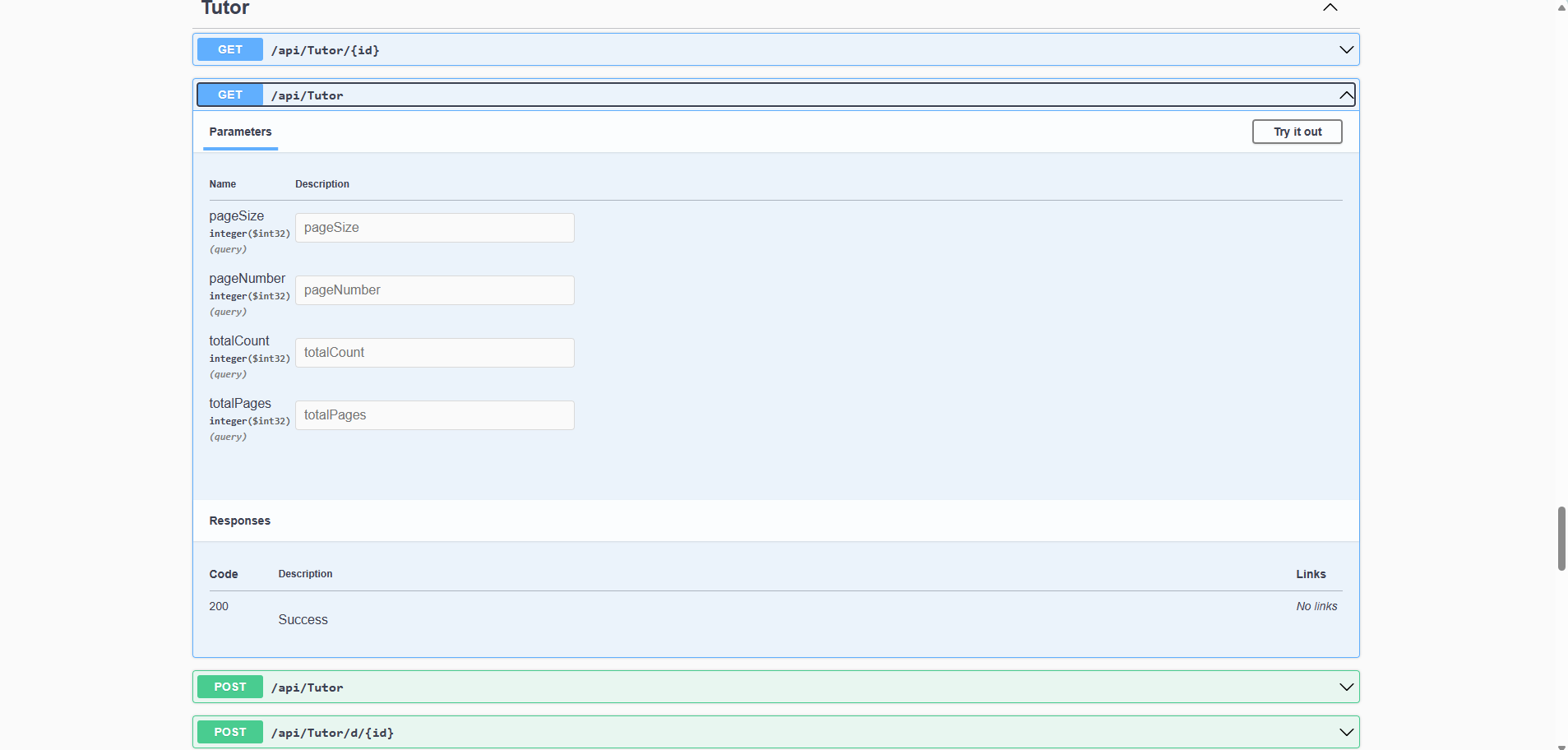
Hình 4.10. Truy vấn lấy giỏ hàng cho học viên



Hình 4.11. Truy vấn lấy lịch sử giao dịch cho học viên

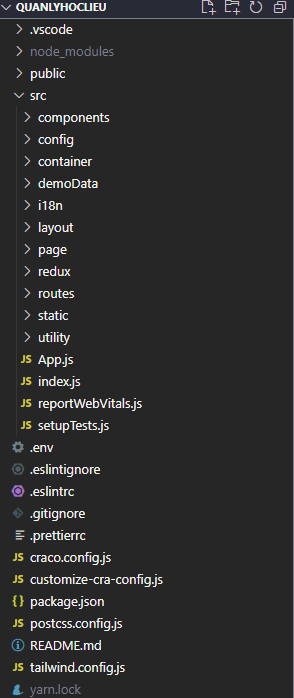


Hình 4.12. Truy vấn lấy danh sách học viên cho Admin



Hình 4.13. Truy vấn lấy danh sách giảng viên cho Admin

## Trang front end

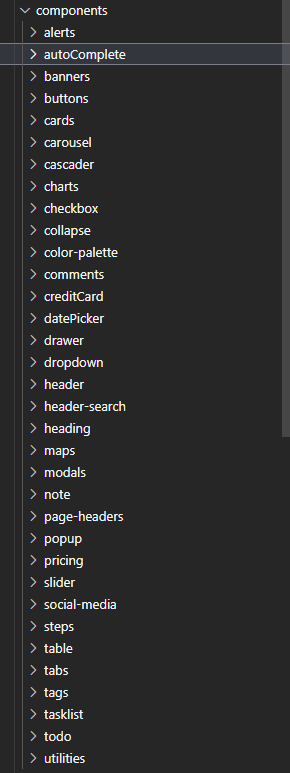


Hình 4.14. Cấu trúc source code front end

Ở cấu trúc source code phần page này, ta chia ra làm các component khác nhau tương ứng chứa các file code theo chức năng:

* node\_modules: chứa các thư viện phục vụ cho dự án.
* package.json: chứa thông tin, đường dẫn cài các package cho dự án.
* README.md: chứa hướng dẫn build, run dự án.
* src: chứa code chính của dự án và các tài nguyên của dự án.

+ components: chứa các thành phần hay được sử dụng trong trang web: alerts, banners, …



Hình 4.15. Cấu trúc file components

## Kiểm thử phần mềm

# KẾT LUẬN VÀ KIẾN NGHỊ

## Kết quả đạt được

Sau khi thời gian thực hiện đề tài, em cảm thấy đồ án tốt nghiệp giúp cho em nâng cao được nhiều kĩ năng cùng với đó là một số kết quả đạt được:

* Hiểu rõ quy trình bán khóa học trực tuyến.
* Xây dựng thành công website bán hàng quần áo đáp ứng nhu cầu đặt ra của người dùng.
* Tìm hiểu và nắm rõ công cụ phân tích thiết kế và xây dựng website.

+ Giao diện của chương trình thân thiện, dễ sử dụng.

+ Website đã giúp người dùng tiết kiệm thời gian công sức để có được một sản phẩm ưng ý. Dễ dàng quản lý thông tin chi tiết của người dùng, các thông tin về sản phẩm và những đơn hàng của người mua sản phẩm từ đó sẽ thuận tiện cho việc thanh toán và mua hàng.

* Đặc biệt là giúp cho em nâng cao khả năng lập trình, củng cố kiến thức các môn học trong những học kì vừa qua mà thầy cô tâm huyết giảng dạy như môn Cơ Sở Dữ Liệu, Lập trình hướng đối tượng, Lập trình web asp.net…
* Thêm vào đó là khả năng tự học, tự tìm hiểu framework mới như TailwindCSS, ReactJs.
* Với một người lập trình viên, khả năng tự học, đọc là liệu tiếng anh là một điều rất quan trọng, em sẽ cố gắng củng cố điều đó.

## Khó khăn gặp phải

Chạy ứng dụng website trên môi trường localhost chưa được thử nghiệm trên mạng internet.

Chưa thể cập nhập được hết những khóa học nổi bật hiện đang có mặt trên thị trường.

Chưa có nhiều loại khóa học với thời gian, dung lượng, yêu cầu khác nhau.

Về phần nghiệp vụ, website chỉ đáp ứng nhiệp vụ mua hàng, quản lý khóa học, chưa có các chức năng bổ sung như làm bài kiểm tra, quản lí bài kiểm tra, đáp án,…

Chưa xử lý tốt vấn đề bảo mật dữ liệu, bảo mật tài khoản.

## Kiến nghị

Để website được phát triển hơn trong tương lai em cần tiếp tục cố gắng trau dồi kiến thức của bản thân và hoàn thiện bài báo cáo thực nghiệm tốt hơn, giúp hoàn thành yêu cầu đề tài tốt hơn.

Hướng phát triển của website, cần hoản thiện hơn:

* Về mặt chức năng:

+ Thêm chức năng Làm bài kiểm tra cho học viên.

+ Thêm chức năng Quản lý bài kiểm tra, đáp án và danh sách kết quả kiểm tra cho Admin.

+ Thêm chức năng Xem danh sách kết quả kiểm tra cho giảng viên.

+ Quản lý đăng nhập của tài khoản học viên, tránh cho học viên mua, bán hay thuê tài khoản bằng cách quản lý thiết bị đăng nhập hoặc đăng nhập yêu cầu xác minh mã code gửi về email đăng ký của học viên.

+ Quản lý thời gian của khóa học sau khi học viên mua nhằm đốc thúc quá trình học của học viên, nâng cao hiệu suất học tập, đồng thời cũng giảm thiểu vấn đề học viên mua, bán hay thuê tài khoản.

+ Tăng cường vấn đề bảo mật dữ liệu cho website.

+ Áp dụng AI, Machine Learning vào giải đáp các vấn đề của học viên qua chức năng live chat.

* Về mặt kỹ thuật:

+ Code cần clean, dễ hiểu hơn.

+ Áp dụng một số mô hình xây dựng website khác ngoài Mô hình thác nước (Waterfall) vào quá trình phát triển phần mềm: Mô hình xoắn ốc (Spiral model), Mô hình Agile, Mô hình tiếp cận lặp (Iterative model), Mô hình tăng trưởng (Incremental model), Mô hình Scrum,…

+ Sử dụng nhiều tiếng Anh hơn trong quá trình xây dựng trang web.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | N. T. Phú, Giáo trình Thiết kế web, Đại học Công nghiệp Hà nội. |
| [2] | Trinh Thanh Tung, Giáo trình Lập trình web, Đại học Bách Khoa Hà nội. |
| [3] | http://freetuts.net/hoc-php. |
| [4] | https://tailwindcss.com/docs/installation |
| [5] | https://react.dev/learn |
| [6] | https://www.youtube.com/@evondevfrontend |
| [7] | https://stackoverflow.com/ |
| [8] | https://viblo.asia/. |