

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2025-2026

PHÁT TRIỂN GIAO DIỆN FRONTEND  
CHO NỀN TẢNG HỌC TRỰC TUYẾN  
CÓ YẾU TỐ TRÒ CHƠI

Giáo viên hướng dẫn

ThS. Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Nguyễn Huỳnh Phú Vinh

MSSV: 110122203

Lớp: DA22TTC

Vĩnh Long, Tháng 12 Năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2025-2026

PHÁT TRIỂN GIAO DIỆN FRONTEND  
CHO NỀN TẢNG HỌC TRỰC TUYẾN  
CÓ YẾU TỐ TRÒ CHƠI

Giáo viên hướng dẫn

ThS. Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Nguyễn Huỳnh Phú Vinh

MSSV: 110122203

Lớp: DA22TTC

Vĩnh Long, Tháng 12 Năm 2025

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Vĩnh Long, ngày ... tháng ... năm 2025

## **Giáo viên hướng dẫn**

(Ký và ghi rõ họ tên)

## **NHÂN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

Vĩnh Long, ngày ... tháng ... năm 2025

## Thành viên hội đồng

(Ký và ghi rõ ho tên)

## LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến quý thầy cô, các bạn bè đã luôn quan tâm, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài này.

Đề tài này là một thử thách lớn đối với em, đòi hỏi em phải có kiến thức chuyên môn vững vàng và kỹ năng lập trình thành thạo. Tuy nhiên, nhờ sự hướng dẫn tận tình của cô, sự động viên của gia đình và bạn bè, em đã có thể hoàn thành đề tài đúng tiến độ và đạt được kết quả tốt.

Em xin chân thành cảm ơn cô đã luôn dành thời gian quý báu để hướng dẫn, giải đáp thắc mắc của em trong suốt quá trình thực hiện đề tài. Cô đã giúp em hiểu rõ hơn về đề tài, từ đó có thể xây dựng được đề tài đáp ứng được nhu cầu của người dùng.

Em cũng xin gửi lời cảm ơn đến các bạn bè trong lớp đã luôn giúp đỡ và động viên em trong suốt quá trình học tập và thực hiện đề tài. Sự giúp đỡ của các bạn đã giúp em có thêm động lực để hoàn thành đề tài một cách tốt nhất.

Một lần nữa, em xin gửi lời cảm ơn chân thành đến quý thầy cô, các bạn bè đã luôn quan tâm, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài.

Em xin chân thành cảm ơn!

Trân trọng,

## MỤC LỤC

<b>NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN .....</b>	<b>i</b>
<b>NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG.....</b>	<b>ii</b>
<b>LỜI CẢM ƠN .....</b>	<b>iii</b>
<b>MỤC LỤC.....</b>	<b>iv</b>
<b>DANH MỤC HÌNH ẢNH – BẢNG BIỂU .....</b>	<b>ix</b>
<b>CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU.....</b>	<b>1</b>
1.1. Giới thiệu đề tài.....	1
1.2. Mục đích nghiên cứu.....	2
1.3. Đối tượng nghiên cứu .....	3
1.4. Phạm vi nghiên cứu.....	3
1.5. Phương pháp nghiên cứu.....	4
<b>CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT.....</b>	<b>5</b>
2.1. Tổng quan về công nghệ phát triển web .....	5
2.1.1 Khái niệm về website và hệ thống web động .....	5
2.1.2 Kiến trúc Client – Server .....	5
2.1.3 Công nghệ nền tảng.....	6
2.1.4 Môi trường phát triển .....	6
2.2. Ngôn ngữ lập trình và Framework sử dụng .....	7
2.2.1 Next.js .....	7
2.2.1.1 Giới thiệu Next.js .....	7
2.2.1.2 Cấu trúc thư mục dự án .....	8
2.2.1.3 Cơ chế Rendering.....	10
2.2.1.4 Routing và Điều hướng .....	11

---

2.2.1.5 Quản lý tài nguyên tĩnh .....	12
2.2.1.6 SEO và tối ưu hiệu năng .....	13
2.2.1.7 Triển khai và vận hành.....	14
2.2.2 Phaser.js .....	15
2.2.2.1 Giới thiệu Phaser.js .....	15
2.2.2.2 Các thành phần cấu trúc dự án Phaser .....	16
2.2.2.3 Quản lý Scene.....	17
2.2.2.4 Tải và quản lý tài nguyên.....	18
2.2.2.5 Hệ thống vật lý .....	19
2.2.2.6 Đầu vào và điều khiển.....	19
2.2.2.7 Âm thanh và animation .....	20
2.2.2.8 Triển khai và xuất bản .....	21
2.3. API .....	22
2.3.1 Khái niệm về API.....	22
2.3.2 Cấu trúc request và response trong API.....	23
2.3.3 Tương tác API HTTP với axios .....	24
2.3.4 Tương tác realtime với Socket.IO Client.....	26
2.4. Công nghệ giao diện và thiết kế Responsive .....	27
2.4.1 Tailwind CSS .....	27
2.4.2 Shadcnui.....	29
2.4.3 Chart.js .....	30
2.5. Bảo mật và hiệu năng trong ứng dụng web .....	32
2.5.1 Xác thực và phân quyền.....	32
2.5.2 Quản lý session, cookie.....	33

---

2.5.3 Tối ưu truy vấn, caching với TanStack Query .....	33
2.6. Tổng quan các công cụ hỗ trợ phát triển và triển khai.....	35
2.6.1 Git/GitHub.....	35
2.6.2 Browser DevTools .....	36
2.7. Tổng kết chương .....	38
<b>CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU.....</b>	<b>39</b>
3.1. Mô tả bài toán.....	39
3.2. Phân tích thiết kế hệ thống .....	40
3.2.1 Đặc tả yêu cầu hệ thống .....	40
3.2.1.1 Yêu cầu chức năng .....	40
3.2.1.2 Yêu cầu phi chức năng .....	41
3.2.2 Kiến trúc hệ thống.....	42
3.2.3 Kịch bản tương tác người dùng.....	44
3.2.3.1 Quy trình đăng nhập hệ thống .....	45
3.2.3.2 Quy trình tạo và quản lý bài kiểm tra .....	46
3.2.3.3 Quy trình tham gia bài kiểm tra của sinh viên.....	47
3.2.3.4 Quy trình nghiệp vụ quản lý của admin.....	48
3.2.4 Tương tác giữa frontend và backend.....	49
3.2.4.1 Tiến trình đăng nhập tài khoản.....	49
3.2.4.2 Tiến trình tạo bài quiz mới.....	50
3.2.4.3 Tiến trình tham gia bài quiz.....	51
3.2.4.4 Tiến trình quản lý chương trình đào tạo.....	52
3.2.5 Thiết kế giao diện.....	54
3.2.5.1 Sơ đồ website.....	54

---

3.2.5.2 Phác thảo giao diện đăng nhập .....	56
3.2.5.3 Phác thảo giao diện trang tổng quan hệ thống.....	57
3.2.5.4 Phác thảo giao diện quản lý danh sách khóa đào tạo .....	57
3.2.5.5 Phác thảo giao diện quản lý danh sách chương trình đào tạo .....	58
3.2.5.6 Phác thảo giao diện quản lý danh sách môn học .....	59
3.2.5.7 Phác thảo giao diện quản lý tài khoản người dùng.....	59
3.2.5.8 Phác thảo giao diện môn học phân công.....	60
3.2.5.9 Phác thảo giao diện khóa học của giảng viên.....	61
3.2.5.10 Phác thảo giao diện quản lý ngân hàng câu hỏi .....	61
3.2.5.11 Phác thảo giao diện tạo bài kiểm tra.....	62
3.2.5.12 Phác thảo giao diện báo cáo ma trận kiến thức .....	63
3.2.5.13 Phác thảo giao diện bài kiểm tra theo phong cách trò chơi.....	63
3.2.5.14 Phác thảo giao diện bảng xếp hạng thành tích sinh viên .....	64
3.2.5.15 Phác thảo giao diện cửa hàng vật phẩm.....	65
<b>CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU.....</b>	<b>66</b>
4.1. Kết quả triển khai hệ thống .....	66
4.2. Kết quả các kịch bản triển khai .....	67
4.2.1 Đăng nhập và phân quyền .....	67
4.2.2 Giảng viên tạo bài kiểm tra .....	69
4.2.3 Sinh viên làm bài dạng Game .....	76
4.2.4 Sinh viên xem kết quả và xếp hạng.....	78
4.2.5 Quản trị viên quản lý hệ thống.....	80
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>89</b>
5.1. Kết quả đạt được .....	89

---

5.2. Hướng phát triển .....	89
-----------------------------	----

<b>TÀI LIỆU THAM KHẢO .....</b>	<b>90</b>
---------------------------------	-----------

## **DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

Hình 2.1 Mô hình Client – Server [2] .....	5
Hình 2.2 Cấu trúc thư mục Next.js 16 .....	8
Hình 3.1 Kiến trúc hệ thống.....	42
Hình 3.2 Sơ đồ hoạt động đăng nhập .....	45
Hình 3.3 Sơ đồ hoạt động tạo Quiz mới .....	46
Hình 3.4 Sơ đồ hoạt động tham gia Quiz.....	47
Hình 3.5 Sơ đồ hoạt động quản lý của Admin.....	48
Hình 3.6 Sơ đồ tuần tự quá trình đăng nhập tài khoản .....	50
Hình 3.7 Sơ đồ tuần tự quá trình tạo quiz mới.....	51
Hình 3.8 Sơ đồ tuần tự quá trình tham gia quiz .....	52
Hình 3.9 Sơ đồ tuần tự quá trình quản lý Chương trình đào tạo.....	53
Hình 3.10 Sơ đồ website quyền admin phần Tổng quan và Quản lý Tài khoản.....	54
Hình 3.11 Sơ đồ website quyền giảng viên phần cấu trúc tổng quan .....	55
Hình 3.12 Sơ đồ website quyền sinh viên phần Hoạt động Học tập và Kiểm tra.....	55
Hình 3.13 Sơ đồ website quyền sinh viên phần Trò chơi hóa .....	56
Hình 3.14 Phác thảo giao diện đăng nhập.....	56
Hình 3.15 Phác thảo giao diện trang tổng quan hệ thống .....	57
Hình 3.16 Phác thảo giao diện quản lý danh sách khóa đào tạo .....	58
Hình 3.17 Phác thảo giao diện quản lý danh sách chương trình đào tạo .....	58
Hình 3.18 Phác thảo giao diện quản lý danh sách môn học.....	59
Hình 3.19 Phác thảo giao diện quản lý tài khoản người dùng .....	60
Hình 3.20 Phác thảo giao diện danh sách môn học được phân công.....	60
Hình 3.21 Phác thảo giao diện quản lý khóa học giảng dạy .....	61

---

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

Hình 3.22 Phác thảo giao diện danh sách ngân hàng câu hỏi .....	62
Hình 3.23 Phác thảo giao diện thiết lập thông tin cơ bản bài kiểm tra .....	62
Hình 3.24 Phác thảo giao diện báo cáo phân bổ kiến thức bài kiểm tra .....	63
Hình 3.25 Phác thảo giao diện làm bài kiểm tra dạng game.....	64
Hình 3.26 Phác thảo giao diện bảng xếp hạng thi đấu .....	64
Hình 3.27 Phác thảo giao diện cửa hàng đổi quà .....	65
Hình 4.1 Giao diện đăng nhập.....	67
Hình 4.2 Giao diện tổng quan dành cho quản trị viên .....	68
Hình 4.3 Giao diện tổng quan môn học phân công dành cho giảng viên .....	68
Hình 4.4 Giao diện tổng quan khóa học dành cho sinh viên .....	69
Hình 4.5 Giao diện quản lý ngân hàng câu hỏi .....	70
Hình 4.6 Giao diện tạo mới câu hỏi .....	71
Hình 4.7 Giao diện tạo mới bài kiểm tra - Bước 1 .....	71
Hình 4.8 Giao diện tạo mới bài kiểm tra - Bước 2.....	72
Hình 4.9 Giao diện tạo mới bài kiểm tra - Bước 3.....	73
Hình 4.10 Giao diện tạo mới bài kiểm tra - Bước 4.....	74
Hình 4.11 Giao diện quản lý danh sách bài kiểm tra .....	74
Hình 4.12 Chi tiết bài kiểm tra.....	75
Hình 4.13 Màn hình sảnh chờ bài kiểm tra .....	76
Hình 4.14 Màn hình chờ tham gia của sinh viên .....	76
Hình 4.15 Giao diện bài kiểm tra theo phong cách trò chơi .....	77
Hình 4.16 Giao diện hoàn thành bài kiểm tra chế độ đánh giá .....	77
Hình 4.17 Lịch sử kết quả bài kiểm tra .....	78
Hình 4.18 Chi tiết kết quả bài kiểm tra và phân tích so sánh .....	79

---

---

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

Hình 4.19 Bảng xếp hạng thành tích sinh viên .....	80
Hình 4.20 Giao diện quản lý chương trình đào tạo .....	81
Hình 4.21 Giao diện chi tiết chương trình đào tạo .....	81
Hình 4.22 Quản lý Mục tiêu chương trình .....	82
Hình 4.23 Quản lý Chuẩn đầu ra chương trình .....	82
Hình 4.24 Giao diện ma trận liên kết PO – PLO .....	83
Hình 4.25 Quản lý danh sách môn học thuộc chương trình .....	83
Hình 4.26 Giao diện chi tiết học phần .....	84
Hình 4.27 Quản lý chuẩn đầu ra học phần .....	85
Hình 4.28 Quản lý cấu trúc chương học .....	85
Hình 4.29 Chi tiết nội dung chương học .....	86
Hình 4.30 Ma trận liên kết Chương học và Chuẩn đầu ra .....	86
Hình 4.31 Giao diện quản lý khóa đào tạo .....	87
Hình 4.32 Giao diện chi tiết khóa đào tạo .....	87
Hình 4.33 Giao diện quản lý học kỳ và phân công giảng dạy .....	88
Hình 4.34 Giao diện quản lý tài khoản người dùng .....	88

## CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU

### 1.1. Giới thiệu đề tài

Trong bối cảnh xã hội ngày càng chú trọng tới chuyển đổi số trong giáo dục, các nền tảng học tập trực tuyến dần trở thành công cụ không thể thiếu nhằm đáp ứng yêu cầu đổi mới phương pháp giảng dạy, mở rộng cơ hội tiếp cận tri thức cho mọi đối tượng người học. Tuy nhiên, quá trình số hóa giáo dục cũng đặt ra thách thức lớn về việc giữ chân học viên, duy trì động lực và đảm bảo hiệu quả đào tạo xuyên suốt quá trình học tập. Sự thiếu hứng thú, giảm tính tương tác và động lực có thể dẫn tới kết quả học tập không như mong muốn, làm hạn chế tiềm năng phát triển của từng sinh viên cũng như giảm hiệu quả vận hành hệ thống giáo dục nói chung.

Nhằm giải quyết vấn đề này, đề tài “**Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi**” được xây dựng với mục tiêu đổi mới toàn diện trải nghiệm học trực tuyến, lấy việc tích hợp các yếu tố trò chơi vào giao diện người dùng làm trung tâm sáng tạo. Việc bổ sung yếu tố trò chơi giúp tăng sự hấp dẫn, tạo động lực cho sinh viên thông qua những thử thách, phần thưởng, bảng xếp hạng hay hệ thống điểm thưởng, từ đó thúc đẩy sự chủ động tiếp cận và duy trì tiến độ học tập của từng cá nhân. Đồng thời, yếu tố trò chơi còn tạo môi trường cạnh tranh lành mạnh, khuyến khích sinh viên không ngừng nâng cao thành tích và phát triển bản thân. Việc phát triển giao diện frontend cho hệ thống học tập trực tuyến cũng đóng vai trò quan trọng trong việc tạo môi trường học tập trực quan, dễ sử dụng, đồng thời tối ưu hiệu quả cho việc hỗ trợ quản lý đào tạo và kiểm soát chất lượng dạy và học.

Về cấu trúc chức năng, giao diện người dùng phát triển trong khuôn khổ đề tài bao gồm hai phần chính, phục vụ từng nhóm đối tượng sử dụng khác nhau:

Giao diện quản lý học thuật: Đây là khu vực dành riêng cho quản trị viên và giảng viên, cung cấp đầy đủ các công cụ quản trị toàn diện trên hệ thống. Tại đây, người dùng có thể thực hiện quản lý, kiểm soát các đối tượng như chương trình đào tạo, khóa đào tạo, phân công giảng dạy, tài khoản người học và người dạy, mục tiêu đào tạo, chuẩn đầu ra, danh mục môn học, chương trình giảng dạy, ma trận liên kết nội dung, ngân hàng

câu hỏi, danh sách sinh viên, quá trình đánh giá, bài tập luyện tập..., đảm bảo mọi hoạt động học thuật diễn ra suôn sẻ, minh bạch và khoa học.

Giao diện dành cho tiến trình luyện tập, đánh giá học viên: Được thiết kế hiện đại, thân thiện cùng sự tích hợp các yếu tố trò chơi, giao diện này mang đến cho sinh viên môi trường tương tác sinh động. Người học có thể dễ dàng theo dõi tiến độ, thành tích cá nhân qua từng bài luyện tập, bài kiểm tra đánh giá định kỳ; đồng thời, kết quả học tập còn được so sánh với mức trung bình hoặc điểm số xuất sắc nhất lớp, tạo động lực phấn đấu và điều chỉnh kế hoạch học tập kịp thời.

Không dừng lại ở đó, hệ thống được tích hợp thêm công cụ trực quan hóa dữ liệu học tập dưới nhiều dạng biểu đồ, thống kê, giúp cả sinh viên lẫn giảng viên tiếp cận số liệu tổng hợp một cách trực quan, dễ hiểu. Đối với sinh viên, những biểu đồ này cung cấp cái nhìn rõ nét về quá trình tiến bộ của bản thân, xác định điểm mạnh và điểm yếu, đổi chiếu với trung bình lớp để đặt mục tiêu phấn đấu phù hợp hơn. Đối với giảng viên, tính năng này hỗ trợ theo dõi toàn cảnh chất lượng lớp học, phát hiện sớm những vấn đề chung, từ đó có giải pháp kịp thời trong điều chỉnh nội dung, phương pháp giảng dạy, góp phần nâng cao kết quả đào tạo chung cho toàn hệ thống.

Tóm lại, đề tài không chỉ góp phần hoàn thiện kỹ thuật xây dựng giao diện người dùng cho một nền tảng học trực tuyến hiện đại mà còn hướng đến việc nâng cao trải nghiệm, tăng động lực học tập và thúc đẩy hiệu quả thực tiễn của mô hình giáo dục số hóa hiện nay.

## 1.2. Mục đích nghiên cứu

Đề tài hướng đến việc nghiên cứu và ứng dụng các công nghệ hiện đại trong phát triển giao diện người dùng cho nền tảng học trực tuyến. Thông qua đó, xây dựng một hệ thống giao diện trực quan, thân thiện và hiệu quả, tích hợp các yếu tố trò chơi nhằm tạo động lực học tập, tăng tính tương tác cũng như nâng cao trải nghiệm người dùng cho sinh viên, giảng viên và quản trị viên. Bên cạnh mục tiêu phát triển sản phẩm để ứng dụng vào thực tiễn, việc thực hiện đề tài còn giúp người viết mở rộng kiến thức, rèn

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

luyện kỹ năng về thiết kế, xây dựng giao diện cũng như nâng cao hiểu biết chuyên sâu về các xu hướng mới trong phát triển phần mềm giáo dục.

### 1.3. Đối tượng nghiên cứu

Các công nghệ frontend hiện đại: Tìm hiểu chuyên sâu về các framework, thư viện như Next.js, Tailwind CSS, Shadcn/ui, TanStack Query, Chart.js, Phaser.js và Socket.IO Client, qua đó ứng dụng vào xây dựng giao diện học trực tuyến tích hợp yếu tố trò chơi.

Các mô hình giao diện học trực tuyến hiện có: Phân tích, khảo sát ưu và nhược điểm của các nền tảng phổ biến, từ đó chọn lọc yếu tố ưu việt để đưa vào thực hiện.

Nhu cầu và hành vi người dùng: Tìm hiểu xu hướng học tập trực tuyến, thói quen, mức độ tương tác với hệ thống trò chơi của sinh viên để định hình đúng trải nghiệm người dùng.

Bộ dữ liệu học tập, tiến trình và kết quả: Thu thập và trực quan hóa dữ liệu học tập, bài kiểm tra và đánh giá, kết quả, giúp sinh viên, giảng viên chủ động theo dõi và điều chỉnh kế hoạch học tập, giảng dạy.

### 1.4. Phạm vi nghiên cứu

Về phạm vi kỹ thuật: Đề tài giới hạn trong quy trình phát triển giao diện frontend, bao gồm các bước từ thiết kế giao diện, triển khai lập trình các chức năng cho từng vai trò sinh viên, giảng viên và quản trị viên, kiểm thử, đánh giá trải nghiệm người dùng.

Về phạm vi chuyên môn: Sản phẩm tập trung thực hiện các hoạt động học tập có tích hợp trò chơi: luyện tập, đánh giá, quản lý, đồng thời phát triển giao diện trực quan hóa dữ liệu học tập bằng các biểu đồ, thống kê.

Đối tượng phục vụ: Chủ yếu dành cho sinh viên, giảng viên, quản trị viên tại các trường đại học ở Việt Nam, với khả năng mở rộng thêm cho các môi trường học trực tuyến khác.

### 1.5. Phương pháp nghiên cứu

*Phương pháp nghiên cứu lý thuyết:* Tham khảo tài liệu, giáo trình, các nền tảng học trực tuyến có yếu tố trò chơi. Tìm hiểu sâu về framework, thư viện frontend và các nguyên lý thiết kế UI/UX.

*Phương pháp nghiên cứu thực nghiệm:* Tiến hành cài đặt môi trường, lập trình các tính năng cho giao diện, tích hợp các yếu tố trò chơi và công cụ trực quan hóa dữ liệu. Thực hiện kiểm thử cho nhiều trường hợp sử dụng thực tế với từng vai trò, lấy ý kiến phản hồi cải tiến nhằm tối ưu sản phẩm cuối cùng.

## CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

### 2.1. Tổng quan về công nghệ phát triển web

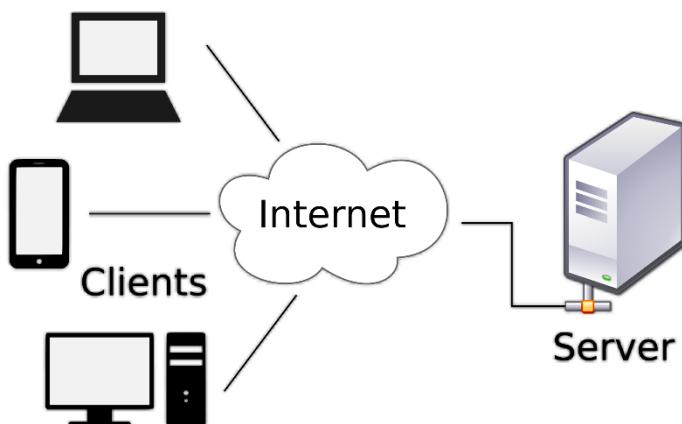
#### 2.1.1 Khái niệm về website và hệ thống web động

Website là tập hợp các trang thông tin gồm văn bản, hình ảnh, video, liên kết,... được hiển thị qua trình duyệt thông qua kết nối Internet. Theo cấu trúc và chức năng, website được chia thành hai loại chủ yếu: website tĩnh và website động. Website động là nền tảng ứng dụng web phổ biến hiện nay với khả năng tương tác với người dùng, cho phép cập nhật, thay đổi nội dung liên tục thông qua cơ sở dữ liệu hoặc hệ thống quản trị. Website động đóng vai trò trọng yếu trong quản lý hoạt động, giới thiệu sản phẩm, giao dịch trực tuyến và cung cấp dịch vụ trực tiếp đến người dùng [1].

#### 2.1.2 Kiến trúc Client – Server

Kiến trúc Client – Server là mô hình nền tảng của hầu hết các hệ thống web hiện đại. Trong đó, Client thường là trình duyệt web người dùng đảm nhiệm vai trò gửi yêu cầu về tài nguyên hoặc dịch vụ tới Server. Máy chủ xử lý các yêu cầu, truy xuất dữ liệu từ hệ quản trị cơ sở dữ liệu, sau đó trả về kết quả cho client [2]. Kiến trúc này thường triển khai dưới dạng mô hình 3 lớp:

- Lớp trình bày: giao diện người dùng.
- Lớp logic nghiệp vụ: xử lý dữ liệu và hoạt động hệ thống.
- Lớp dữ liệu: truy xuất, lưu trữ dữ liệu.



Hình 2.1 Mô hình Client – Server [2]

Giao thức trao đổi phổ biến nhất hiện nay là HTTP hoặc HTTPS, cho phép truyền tải thông tin bảo mật và ổn định.

### 2.1.3 Công nghệ nền tảng

Công nghệ nền tảng trong phát triển frontend web hiện đại bao gồm nhiều thành phần liên kết chặt chẽ để tạo thành giao diện và chức năng động cho người dùng. HTML là ngôn ngữ đánh dấu tiêu chuẩn, đóng vai trò xác định cấu trúc nội dung của trang web. Bên cạnh đó, CSS thực hiện nhiệm vụ định dạng, thiết kế giao diện, quản lý layout và các hiệu ứng trực quan. JavaScript là ngôn ngữ lập trình phía client giúp bổ sung khả năng tương tác, xử lý logic, kết nối dữ liệu không đồng bộ và quản lý các tác vụ thời gian thực [3].

Các framework hiện đại như React, Vue.js, Angular cho phép xây dựng ứng dụng web theo kiến trúc component hóa, giúp quản trị mã nguồn tối ưu, dễ dàng bảo trì và mở rộng. Ngoài ra, sự kết hợp với các thư viện của JavaScript hoặc CSS như Bootstrap, Tailwind CSS, Material UI,... nâng cao tốc độ hoàn thiện giao diện cũng như khả năng tương thích trên nhiều loại thiết bị.

Quản lý trạng thái ứng dụng và dữ liệu càng trở nên quan trọng đối với các hệ thống quy mô lớn, do đó các giải pháp quản lý như Redux, Context API, hay các thư viện truy vấn dữ liệu như TanStack Query được ưu tiên cao. Với các ứng dụng yêu cầu tính đồng bộ theo thời gian thực thì các công nghệ như WebSocket hoặc Socket.IO thường được lựa chọn. Ngoài ra, còn có các thư viện như Chart.js, D3.js giúp hỗ trợ hiển thị và phân tích dữ liệu trực quan.

Việc ứng dụng các công nghệ nền tảng này giúp sản phẩm không những đạt được chất lượng về mặt giao diện, hiệu năng, mà còn đảm bảo khả năng bảo trì, mở rộng lâu dài, đáp ứng tiêu chuẩn công nghiệp hiện hành.

### 2.1.4 Môi trường phát triển

Môi trường phát triển frontend là tập hợp các công cụ, quy trình và hệ thống hỗ trợ lập trình viên trong suốt vòng đời dự án. Thông thường, các trình soạn thảo mã nguồn

hiện đại như Visual Studio Code, WebStorm, Sublime Text được sử dụng rộng rãi nhờ tích hợp nhiều chức năng kiểm tra cú pháp, gỡ lỗi, quản lý dự án và mở rộng tiện ích.

Hệ thống quản lý mã nguồn như Git kết hợp với nền tảng lưu trữ trực tuyến như GitHub, GitLab giúp kiểm soát phiên bản, hỗ trợ làm việc nhóm hiệu quả, đồng thời tích hợp dễ dàng với quy trình CI/CD để tự động kiểm thử và triển khai. Các công cụ kiểm thử như Chrome DevTools, Lighthouse, Postman, Jest hoặc Cypress góp phần vào việc kiểm tra, tối ưu hiệu suất, đánh giá mức độ tương thích, bảo mật và độ chính xác của mã nguồn [1].

Quản lý thư viện và môi trường thông qua npm, yarn hoặc pnpm giúp việc cài đặt, cập nhật các thành phần phụ trợ thuận tiện, tránh xung đột và đảm bảo sự nhất quán của mã nguồn. Ngoài ra, môi trường triển khai và thử nghiệm, như Docker, Vercel, Netlify, Heroku cung cấp giải pháp đóng gói, kiểm thử và vận hành sản phẩm một cách nhanh chóng và an toàn.

Vì vậy, việc lựa chọn môi trường phát triển phù hợp là yếu tố quan trọng để đảm bảo năng suất, chất lượng và giúp quá trình xây dựng giao diện web đáp ứng các tiêu chuẩn kỹ thuật cũng như yêu cầu thực tế của các dự án hiện đại.

## 2.2. Ngôn ngữ lập trình và Framework sử dụng

### 2.2.1 Next.js

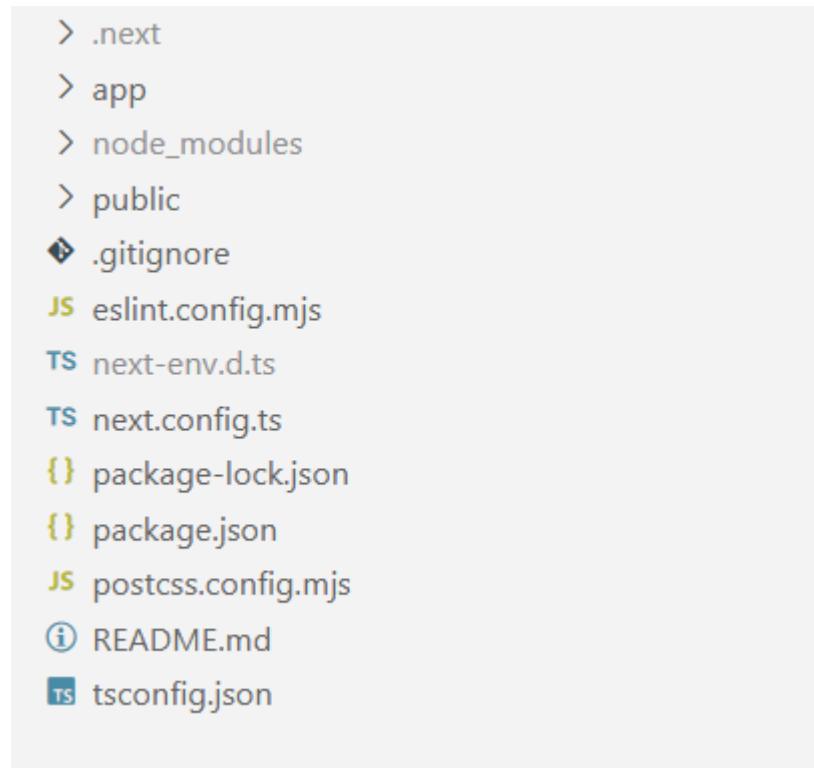
#### 2.2.1.1 Giới thiệu Next.js

Next.js tự động cấu hình các công cụ nền tảng thấp như trình biên dịch, giúp đơn giản hóa quy trình thiết lập và phát triển dự án, tạo điều kiện cho lập trình viên tập trung chủ yếu vào việc xây dựng sản phẩm và triển khai tính năng mới một cách nhanh chóng.

Framework này phù hợp cho cả cá nhân phát triển lẫn các nhóm lớn nhờ khả năng tạo ra các ứng dụng giàu tính tương tác, động và có hiệu suất cao. Next.js nổi bật với các chức năng như rendering linh hoạt như server-side, static, incremental static, client-side, routing dựa trên cấu trúc file, tối ưu hóa SEO, hỗ trợ dữ liệu thời gian thực và tích hợp tốt với hệ sinh thái React [4].

Nhờ những đặc điểm này, Next.js ngày càng được lựa chọn rộng rãi để xây dựng ứng dụng web hiệu quả, mở rộng và thân thiện với người dùng hiện đại.

#### 2.2.1.2 Cấu trúc thư mục dự án



Hình 2.2 Cấu trúc thư mục Next.js 16

Dự án Next.js hiện đại có cách sắp xếp thư mục rõ ràng, giúp dễ quản lý code, mở rộng và bảo trì. Cấu trúc này dựa trên quy ước sẵn có, tập trung vào routing và tách biệt chức năng.

##### - **Thư mục app/**

Đây là thư mục chính cho hệ thống định tuyến mới. Mỗi thư mục con tương ứng với một đường dẫn URL. Các file đặc biệt bao gồm:

page.js hoặc page.tsx: Nội dung chính của trang.

layout.js hoặc layout.tsx: Bộ cục chung cho route hoặc phần con.

loading.js: Màn hình tải khi dữ liệu đang xử lý.

error.js: Xử lý lỗi tùy chỉnh cho route.

route.js hoặc route.ts: Tạo API endpoint ngay trong app.

- **Thư mục public/**

Nơi lưu tài nguyên tĩnh như hình ảnh, icon, font chữ. File ở đây truy cập trực tiếp qua URL gốc, ví dụ /image.png.

- **Thư mục node\_modules/**

Chứa các thư viện và gói cài đặt qua npm, pnpm hoặc yarn. Không chỉnh sửa thủ công.

- **Thư mục components/**

Lưu các phần giao diện tái sử dụng, như nút bấm, form hoặc header. Giúp tránh lặp code.

- **Thư mục styles/**

Chứa file CSS, SCSS hoặc kiểu dáng toàn cục. Có thể dùng module cho từng phần tử riêng.

- **Các file cấu hình ở gốc dự án**

- next.config.js: Thiết lập tùy chỉnh cho Next.js, như môi trường hoặc plugin.
- package.json: Quản lý gói phụ thuộc và lệnh chạy (build, start).
- .env, .env.local, .env.production, .env.development: Lưu biến môi trường như API key.
- tsconfig.json hoặc jsconfig.json: Cấu hình TypeScript hoặc JavaScript.
- .gitignore: Bỏ qua file không cần theo dõi (như node\_modules).
- Các file khác: next-env.d.ts, README.md.

## Kết luận

Cấu trúc này giúp dự án Next.js dễ mở rộng, kiểm thử và triển khai, phù hợp cho ứng dụng web từ nhỏ đến lớn.

### 2.2.1.3 Cơ chế Rendering

Next.js là framework dựa trên React, giúp tối ưu tốc độ và trải nghiệm người dùng cho trang web bằng cách kết hợp nhiều cách hiển thị nội dung. Các cách này cân bằng giữa tải nhanh, dữ liệu mới và khả năng mở rộng dự án.

#### 1. Server-side Rendering (SSR)

SSR tạo nội dung trang trên máy chủ mỗi khi người dùng yêu cầu. Máy chủ lấy dữ liệu mới nhất, tạo HTML hoàn chỉnh và gửi về trình duyệt.

Cách này tốt cho trang cần cập nhật thường xuyên, cải thiện tìm kiếm vì nội dung sẵn sàng ngay lập tức.

#### 2. Static Site Generation (SSG)

SSG tạo toàn bộ trang thành file HTML tĩnh ngay lúc xây dựng dự án. File này phục vụ trực tiếp khi người dùng truy cập, giúp tải rất nhanh.

Phù hợp cho trang ít thay đổi, như blog hoặc trang giới thiệu, nhưng không lý tưởng cho dữ liệu động.

#### 3. Incremental Static Regeneration (ISR)

ISR kết hợp SSG và cập nhật, tái tạo trang tĩnh theo thời gian định sẵn hoặc khi dữ liệu thay đổi.

Nó giữ tốc độ nhanh như tĩnh nhưng vẫn cho phép nội dung mới mà không cần xây dựng lại toàn bộ.

#### 4. Client-side Rendering (CSR)

CSR tạo nội dung và tương tác trên trình duyệt sau khi tải trang ban đầu. JavaScript xử lý dữ liệu và cập nhật giao diện.

Tốt cho phần cần nhiều hành động người dùng, như ứng dụng tương tác, nhưng có thể chậm hơn cho SEO.

#### 5. Partial Prerendering (PPR) và Streaming

PPR tạo trước phần giao diện tĩnh trên máy chủ, trong khi phần động tải dần qua React Suspense.

Streaming gửi dữ liệu từng phần, cập nhật trang liên tục mà không chờ toàn bộ, giúp trải nghiệm mượt mà hơn.

## Kết luận

Next.js cho phép kết hợp các cách hiển thị này để tạo ứng dụng nhanh, an toàn, dễ mở rộng và thân thiện với người dùng.

### 2.2.1.4 Routing và Điều hướng

Next.js dùng cách định tuyến dựa trên cấu trúc thư mục và file, giúp dễ dàng tạo đường dẫn URL và sắp xếp chức năng ứng dụng một cách rõ ràng. Cách này đơn giản cho dự án nhỏ, nhưng cũng linh hoạt cho dự án lớn, hỗ trợ cả đường dẫn cố định và động.

#### - Định tuyến dựa trên file và thư mục

Mỗi thư mục hoặc file trong thư mục app/ sẽ tương ứng với một phần của URL.

File page.js hoặc page.tsx đại diện cho một trang tại đường dẫn cụ thể.

Ví dụ: File app/dashboard/settings/page.js sẽ tạo đường dẫn /dashboard/settings. Thư mục lồng nhau giúp tạo đường dẫn phức tạp hơn.

#### - Định tuyến động

Để tạo đường dẫn thay đổi theo dữ liệu, dùng dấu ngoặc vuông [ ].

Ví dụ: Thư mục app/blog/[slug]/page.js sẽ tạo đường dẫn như /blog/abc hoặc /blog/xyz.

Trong code trang, lấy giá trị động qua props params để sử dụng.

#### - Bổ cục lồng nhau

Next.js cho phép tạo bổ cục lồng nhau để chia sẻ giao diện chung cho toàn bộ hoặc một phần ứng dụng.

File layout.js trong mỗi thư mục giữ cho giao diện nhất quán và dễ tái sử dụng, tránh lặp code.

– **Điều hướng giữa các trang**

Sử dụng thành phần <Link> của Next.js để chuyển trang mượt mà, tự động tải trước dữ liệu để tránh tải lại toàn bộ.

Với trường hợp phức tạp, dùng hook useRouter() để đẩy, thay thế, quay lại trang, hoặc tùy chỉnh hành vi.

Next.js còn tự động tải trước liên kết hiển thị trên màn hình để tăng tốc độ.

Tham số tìm kiếm và props đường dẫn

Trong trang phía máy chủ, lấy tham số URL qua prop searchParams.

Sử dụng helpers như PageProps hoặc LayoutProps để đảm bảo code an toàn kiểu dữ liệu và dễ quản lý props từ định tuyến.

– **API Routes và xử lý đường dẫn**

Next.js cho phép tạo điểm cuối API ngay trong hệ thống định tuyến bằng file route.js hoặc route.ts.

Điều này giúp xử lý yêu cầu từ phía máy chủ mà không cần server riêng.

## Kết luận

Định tuyến của Next.js giúp sắp xếp code rõ ràng, dễ mở rộng, tối ưu phát triển cho cả trang web tĩnh và ứng dụng động. Nó mang lại trải nghiệm chuyển trang nhanh và liền mạch cho dự án hiện đại.

### 2.2.1.5 Quản lý tài nguyên tĩnh

Tài nguyên tĩnh là các file không thay đổi như hình ảnh, font chữ, văn bản hoặc media, giúp xây dựng ứng dụng web hiệu quả. Next.js quản lý chúng qua thư mục public/ ở gốc dự án, tách biệt rõ ràng với code động để dễ bảo trì và mở rộng.

– **Cách đặt và truy cập**

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

Đặt tất cả file tĩnh vào public/. Máy chủ phục vụ trực tiếp mà không qua xử lý routing, giúp phản hồi nhanh và tiết kiệm băng thông.

Ví dụ: File public/assets/img.png có thể truy cập qua /assets/img.png trong code, như <Image src="/assets/img.png" />.

Điều này hỗ trợ tích hợp SEO, như robots.txt hoặc favicon.ico, mà không cần rebuild toàn bộ khi cập nhật.

### - **Tối ưu hóa**

File trong public/ không bị rebuild khi triển khai, nên cập nhật độc lập. Next.js đặt header cache mặc định để tránh lưu cache không an toàn, nhưng bạn có thể tùy chỉnh cho hiệu suất tốt hơn.

## Kết luận

Cách quản lý tài nguyên tĩnh của Next.js đảm bảo ổn định, nhanh chóng và dễ mở rộng, nâng cao trải nghiệm người dùng và hiệu năng dự án web.

### 2.2.1.6 SEO và tối ưu hiệu năng

Next.js giúp dễ dàng kiểm soát tìm kiếm và tốc độ trang web ngay từ đầu dự án, kết hợp hai yếu tố quan trọng để ứng dụng chạy mượt mà và dễ tìm thấy trên công cụ tìm kiếm.

### - **Tối ưu SEO với metadata**

Next.js cho phép đặt thông tin mô tả cho từng trang như tiêu đề, mô tả ngắn, từ khóa và dữ liệu chia sẻ mạng xã hội.

Bạn có thể khai báo trực tiếp trong file page.js hoặc dùng hàm generateMetadata để tạo động dựa trên dữ liệu cụ thể cho từng trang.

Tất cả thông tin này tự động thêm vào phần head của HTML, giúp bot tìm kiếm dễ dàng lập chỉ mục mà không bỏ sót chi tiết.

### - **File đặc biệt và trình duyệt**

Đặt file như favicon.ico, robots.txt hoặc sitemap.xml theo quy tắc Next.js để tăng khả năng hiển thị và chỉ mục tốt hơn.

Next.js còn hỗ trợ metadata cho chia sẻ trên mạng xã hội, tạo bản xem trước hấp dẫn khi liên kết được chia sẻ.

- **Tối ưu hiệu năng**

Sử dụng thành phần Image để tự động nén, thay đổi kích thước và tải chậm hình ảnh, giúp trang tải nhanh hơn.

Tối ưu font chữ bằng cách tải trước và chỉ dùng phần cần thiết, giảm thời gian chờ hiển thị chữ.

Next.js tự động tải trước dữ liệu cho liên kết hiển thị trên màn hình, làm chuyển trang nhanh như chớp.

Kết hợp các cách render như SSR, SSG, ISR và streaming để giảm thời gian tương tác, giữ trang phản hồi tốt.

- **Lưu trữ tạm và truyền dữ liệu**

Next.js dùng cache, streaming và ghi nhớ để lưu metadata và nội dung trang, giảm tải dữ liệu và tăng tốc độ tải.

Điều này giúp trang web tiết kiệm băng thông và cải thiện trải nghiệm trên nhiều thiết bị.

## Kết luận

Với API metadata, file chuẩn SEO, tối ưu hình ảnh/font và tải trước thông minh, Next.js làm trang web thân thiện với tìm kiếm và chạy mượt mà trong môi trường hiện đại.

### 2.2.1.7 Triển khai và vận hành

Next.js hỗ trợ nhiều cách triển khai ứng dụng, phù hợp với các môi trường khác nhau, từ máy chủ đơn giản đến hệ thống lớn, giúp dễ dàng đưa dự án lên sản xuất.

- **Triển khai trên máy chủ Node.js**

Bạn có thể chạy ứng dụng trên bất kỳ dịch vụ hỗ trợ Node.js, như VPS hoặc hosting. Quy trình: Chạy npm run build để tạo phiên bản sản xuất, rồi npm run start để khởi động máy chủ. Tất cả tính năng như render phía máy chủ hoặc tái tạo dàn đều hoạt động đầy đủ.

- **Triển khai với Docker**

Next.js dễ đóng gói thành container Docker để chạy trên Kubernetes, Cloud Run hoặc dịch vụ đám mây như Google Cloud, DigitalOcean. Ưu điểm: Môi trường nhất quán, dễ mở rộng và tích hợp tự động hóa (CI/CD), giúp triển khai nhanh và ổn định.

- **Xuất tĩnh**

Next.js có thể tạo file HTML/CSS/JS tĩnh, phù hợp cho trang web đơn giản hoặc ứng dụng một trang. Triển khai dễ dàng lên AWS S3, Nginx hoặc Apache. Tuy nhiên, tính năng động như API hoặc SSR sẽ không dùng được.

- **Thích ứng với các nền tảng**

Next.js linh hoạt với Vercel, Netlify, Cloudflare, AWS Amplify hoặc Deno Deploy, qua hướng dẫn riêng hoặc bộ chuyển đổi triển khai. Mỗi nền tảng có mẫu sẵn để tối ưu hóa quy trình.

## Kết luận

Với nhiều lựa chọn triển khai và công cụ hỗ trợ, Next.js phù hợp từ dự án cá nhân đến doanh nghiệp lớn, đảm bảo hệ thống ổn định, dễ bảo trì và hiệu quả lâu dài.

### 2.2.2 Phaser.js

#### 2.2.2.1 Giới thiệu Phaser.js

Phaser.js là một framework mã nguồn mở dựa trên JavaScript, chuyên phát triển trò chơi 2D trên nền tảng web HTML5. Framework này hỗ trợ đa nền tảng bao gồm desktop và mobile, sử dụng renderer WebGL hoặc Canvas để tối ưu hiệu suất tùy theo thiết bị [5].

Phaser cung cấp các thành phần cốt lõi như quản lý đối tượng game, cảnh, tương tác đầu vào, engine vật lý, âm thanh, hoạt họa, tài nguyên và tilemap, được tổ chức qua

---

Nguyễn Huỳnh Phú Vinh

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

API hướng đối tượng trực quan. Một dự án cơ bản bắt đầu bằng đối tượng config để định nghĩa renderer, kích thước canvas, engine vật lý, scene và tài nguyên. Sau đó, lớp Phaser.Game khởi tạo vòng đời game với ba phương thức chính:

- **preload()**: Tải tài nguyên vào bộ nhớ.
- **create()**: Khởi tạo đối tượng, môi trường và logic game.
- **update()**: Xử lý tương tác, điều khiển và cập nhật theo frame.

Các đặc trưng nổi bật của Phaser.js bao gồm: mã nguồn mở với cộng đồng hỗ trợ rộng lớn. Xây dựng nhanh chóng trò chơi 2D chất lượng cao nhờ module hóa. Hỗ trợ toàn diện engine vật lý, animation, tilemap, loading asset và sự kiện đầu vào. Đảm bảo trải nghiệm nhất quán trên đa thiết bị và trình duyệt cùng tài liệu với nhiều ví dụ minh họa.

Nhờ thiết kế ưu tiên hiệu suất, tính mô-đun và API thân thiện, Phaser.js là lựa chọn ưu tiên cho lập trình viên phát triển game web hoặc ứng dụng tương tác cao.

### 2.2.2 Các thành phần cấu trúc dự án Phaser

Dự án Phaser.js được tổ chức theo mô hình mô-đun hóa, với cấu trúc thư mục rõ ràng để hỗ trợ quản lý mã nguồn, tài nguyên và mở rộng, thường sử dụng công cụ như npm, Grunt hoặc Webpack. Cấu trúc cơ bản bao gồm tệp khởi tạo, config game, scene, tài nguyên và module quản lý, tối ưu cho phát triển game 2D trên web.

#### Các thành phần chính:

- **Tệp khởi tạo**: Tệp HTML chính nhúng script Phaser và canvas. Tệp JavaScript khởi tạo instance Phaser.Game, thường sử dụng generator như Yeoman để tạo dự án mẫu.
- **Đối tượng cấu hình game**: Định nghĩa thông số cốt lõi như renderer, kích thước, scene danh sách, physics, và preload asset.
- **Scene**: Các trạng thái game như Boot, Preload, Menu, Play, GameOver. Mỗi scene có preload(), create(), update().

- **Tài nguyên và asset:** Thư mục assets/ chứa sprites, sounds, fonts, tilemaps; preload qua lệnh như this.load.image('key', 'path') hoặc atlas cho sprite sheets.
- **Quản lý đối tượng và logic:** Module xử lý sprite, groups, physics, input, animation. Tổ chức trong thư mục game/states/ hoặc prefabs/, hỗ trợ ES6 modules cho dự án lớn.
- **Phân chia mã nguồn:** Thư mục như game/, css/, config.json; tách file cho từng scene hoặc hệ thống để dễ bảo trì và tích hợp bundler.

Tóm lại, cấu trúc Phaser.js nhẫn nại với config trung tâm và scene độc lập, phù hợp cho dự án web game từ cơ bản đến phức tạp.

#### 2.2.2.3 Quản lý Scene

Scene là thành phần cốt lõi của Phaser.js, đại diện cho các màn hình logic như gameplay, menu hoặc UI, giúp phân chia và quản lý vòng đời game một cách hiệu quả. Quản lý scene bao gồm khởi tạo, chuyển đổi và giao tiếp giữa các scene, được xử lý qua Scene Manager để tối ưu hóa hiệu suất và trải nghiệm người dùng.

Các khía cạnh chính:

- **Cấu trúc và khởi tạo:** Mỗi scene là một class kế thừa Phaser.Scene, với phương thức preload() tải tài nguyên, create() khởi tạo đối tượng và tương tác, update() xử lý cập nhật frame, vật lý và sự kiện thời gian thực. Scene được thêm vào game config qua mảng hoặc scene.add(key, SceneClass).
- **Vòng đời scene:** Scene trải qua các trạng thái: khởi tạo, tải, tạo, chạy, tạm dừng, tiếp tục và kết thúc. Scene Manager kiểm soát bằng phương thức như start(), stop(), pause(), resume() để chuyển đổi mượt mà.
- **Quản lý nhiều scene:** Hỗ trợ chạy đồng thời nhiều scene, xếp chồng hoặc lồng ghép. Sử dụng moveAbove(), moveBelow() hoặc swapPosition() để sắp xếp thứ tự render. Có thể chạy scene mới mà không dừng cũ qua launch().
- **Sự kiện và giao tiếp:** Scene sử dụng Event Emitter để gửi hay nhận dữ liệu, như truyền điểm số giữa gameplay và leaderboard qua events.on('eventName',

callback) hoặc data parameter trong start(). Điều này hỗ trợ chia sẻ trạng thái mà không cần global variables.

- **Ứng dụng thực tế:** Phân chia logic thành scene độc lập đơn giản hóa mã nguồn, dễ mở rộng và kiểm thử, đặc biệt cho game đa màn hình.

Tóm lại, quản lý scene trong Phaser.js đảm bảo cấu trúc mô-đun, hiệu suất cao và trải nghiệm mượt mà, là nền tảng cho phát triển game 2D hiện đại.

#### 2.2.2.4 Tải và quản lý tài nguyên

Việc tải và quản lý tài nguyên là bước quan trọng trong Phaser.js, đảm bảo dữ liệu được chuẩn bị đầy đủ trước khi chạy logic game, sử dụng phương thức preload() trong vòng đời scene để tránh lỗi thiếu file. Framework cung cấp hệ thống loader linh hoạt cho các loại asset như hình ảnh, âm thanh, sprite sheet và tilemap, với cache nội bộ để tối ưu hiệu suất.

Quy trình tải asset: Đăng ký và tải trong preload(): Sử dụng các hàm như this.load.image('key', 'path/to/image.png'), this.load.audio('key', 'path/to/sound.mp3'), hoặc this.load.tilemapTiledJSON('key', 'path/to/map.json') để xác định tài nguyên cần thiết. Quy trình này diễn ra trước create(), giúp game sẵn sàng render mà không gián đoạn.

Vận hành asset sau tải: Truy cập qua key trong create(): Tạo sprite (this.add.image(x, y, 'key')), animation hoặc đối tượng vật lý, tận dụng cache để tái sử dụng mà không tải lại, giảm tải cho trình duyệt khi chuyển scene. Hệ thống cache tự động quản lý bộ nhớ, hỗ trợ lặp lại nội dung hiệu quả.

Xử lý nâng cao: Kiểm tra trạng thái: Sử dụng sự kiện load.onLoadComplete, onFileComplete để theo dõi tiến độ, xử lý lỗi tải thất bại; có thể xóa asset khỏi cache qua destroy() để tiết kiệm bộ nhớ trên thiết bị yếu. Điều này cho phép reload động khi cần, tối ưu hiệu suất.

Tóm lại, cách tiếp cận tải và quản lý asset của Phaser.js nhấn mạnh tính chủ động, hiệu quả bộ nhớ và dễ mở rộng, hỗ trợ lập trình viên kiểm soát toàn bộ quy trình.

### 2.2.2.5 Hệ thống vật lý

Phaser.js tích hợp hệ thống vật lý đa dạng cho game 2D, từ arcade cơ bản đến mô phỏng thực tế, với ba engine chính: Arcade, Matter.js và Impact, được cấu hình qua config game để tối ưu hiệu suất trên web. Hệ thống hỗ trợ va chạm, trọng lực và động học, kích hoạt trên đối tượng qua `this.physics.add.sprite()` hoặc `enableBody`.

#### Các hệ vật lý chính:

- **Arcade Physics:** Engine nhẹ, phù hợp game retro, platformer hoặc puzzle, xử lý va chạm hình chữ nhật hoặc tròn với tốc độ cao. Hỗ trợ vận tốc, gia tốc, trọng lực, phản lực và event collision. API đơn giản như `body.velocity.x = 100`, `collideWorldBounds = true`.
- **Matter.js Physics:** Tích hợp thư viện Matter.js cho mô phỏng thực tế, hỗ trợ shape phức tạp, composite body, constraints, joints, restitution và raycasting. Xử lý va chạm đa giai đoạn, stacking và động học bảo toàn năng lượng. Phù hợp puzzle sâu hoặc simulation, với tính năng scaling, zoom và debug inspector.
- **Impact Physics:** Dành cho tương thích với engine khác, cung cấp va chạm cơ bản và rigid body, dễ chuyển đổi từ dự án cũ.

Tóm lại, hệ thống vật lý của Phaser.js linh hoạt, tùy biến cao và hiệu suất tối ưu, hỗ trợ từ game casual đến simulation phức tạp, nâng cao sáng tạo trong phát triển web game.

### 2.2.2.6 Đầu vào và điều khiển

Phaser.js cung cấp hệ thống đầu vào thống nhất, hỗ trợ đa nền tảng bao gồm desktop, mobile và tablet, xử lý sự kiện chuột, bàn phím, cảm ứng và gamepad qua Input Manager và Input Plugin để đảm bảo tính tương thích và mở rộng. Hệ thống này lắng nghe sự kiện DOM gốc, chuyển tiếp đến scene cụ thể, tối ưu hóa bằng cách chỉ kích hoạt input trên đối tượng tương tác.

### Các cơ chế nhập liệu chính:

- **Chuột và cảm ứng:** Hợp nhất mouse,touch thành pointer, xử lý sự kiện pointerdown, pointerup, pointermove. Kích hoạt tương tác qua setInteractive() trên Game Object như sprite hoặc text. Hỗ trợ đa điểm với maxPointers config cho UI phức tạp.
- **Bàn phím:** Theo dõi phím, kiểm tra trạng thái isDown, justDown hoặc event onKeyDown để xử lý điều khiển. Tích hợp dễ dàng cho game platformer hoặc puzzle.
- **Gamepad:** Kích hoạt qua config.input.gamepad = true. Quản lý nút, analog stick và shoulder button với event gamepad.buttonDown, axis value. Hỗ trợ kết nối/mất kết nối tự động. Phù hợp cho game console-like trên web.
- **Xử lý đa điểm:** Cấu hình số pointer,touch, kiểm tra activePointer.id để xử lý chạm riêng biệt trong mini-game hoặc UI đa chạm.

Tóm lại, hệ thống đầu vào của Phaser.js mạnh mẽ, dễ mở rộng và tương thích cao, tạo nền tảng cho tương tác tinh vi trong game web 2D.

#### 2.2.2.7 Âm thanh và animation

Phaser.js tích hợp hệ thống âm thanh và animation mạnh mẽ, sử dụng Sound Manager và Animation Manager để tạo trải nghiệm sống động, hỗ trợ đa nền tảng qua Web Audio API hoặc HTML5 Audio, đảm bảo hiệu suất cao trong game 2D. Các hệ thống này được quản lý toàn cục, dễ tái sử dụng giữa scene và tối ưu hóa tải tài nguyên.

##### - Âm thanh:

Sound Manager tự động chọn Web Audio API nếu hỗ trợ, fallback sang HTML5 Audio cho tương thích rộng. Hỗ trợ tải và phát audio qua key (this.sound.add('key')), với kiểm soát volume, pitch, loop, fade in,out và marker cho audio sprite. Event system xử lý complete, loop hoặc end, cho phép phát đồng thời nhạc nền và hiệu ứng.

##### - Animation:

Animation Manager quản lý global frame-based animation từ sprite sheet hoặc atlas, định nghĩa frames, frameRate, repeat và yoyo cho sprite hoặc image. Tích hợp event như onComplete để chuyển trạng thái, kết hợp Tween cho hiệu ứng mượt mà như move, scale, shake hoặc timeline cho chuỗi động tác phức tạp.

Tóm lại, hệ thống âm thanh và animation của Phaser.js linh hoạt, hiệu quả và dễ mở rộng, tạo nền tảng vững chắc cho game tương tác cao.

#### 2.2.2.8 *Triển khai và xuất bản*

Phaser.js hỗ trợ triển khai game đa nền tảng nhờ nền tảng HTML5, JavaScript và CSS, cho phép xuất bản nhanh chóng lên web server, đóng gói thành app desktop/mobile hoặc tích hợp với dịch vụ như Discord, Steam và Twitch. Quy trình tận dụng bundler như Webpack, Vite hoặc CLI tools để tối ưu hóa, đảm bảo hiệu suất và tương thích rộng.

##### Các hình thức xuất bản phổ biến:

- **Web game (HTML5):** Deploy trực tiếp lên Apache, Nginx, GitHub Pages, Itch.io, Netlify hoặc Vercel bằng cách upload folder dự án, hỗ trợ SCORM cho nền tảng giáo dục như Moodle để lưu trạng thái. Game chạy ngay trên trình duyệt mà không cần plugin.
- **Đóng gói desktop, mobile:** Sử dụng Electron cho .exe/.app, Cordova/Capacitor cho APK/IPA, tích hợp payment, login và lưu trữ cục bộ. Dễ chuyển đổi từ web sang native app.
- **Tích hợp nền tảng, cloud:** Triển khai lên Discord Activities, Twitch Overlay, YouTube Playables hoặc streaming services. Sử dụng bundler để tùy chỉnh metadata và analytics. Hỗ trợ framework như React, Vue cho tích hợp site hiện đại.

Tóm lại, nhờ hệ sinh thái web hiện đại và cộng đồng hỗ trợ, triển khai Phaser.js linh hoạt, nhanh chóng và dễ bảo trì, phù hợp từ dự án cá nhân đến thương mại.

## 2.3. API

### 2.3.1 Khái niệm về API

API là cơ chế cho phép hai thành phần phần mềm khác nhau giao tiếp và trao đổi dữ liệu thông qua một tập hợp các định nghĩa và giao thức chuẩn. API đóng vai trò như một hợp đồng dịch vụ, định nghĩa rõ cách thức một phần mềm có thể gửi yêu cầu đến một phần mềm hoặc dịch vụ khác, và nhận về phản hồi phù hợp.

Ví dụ điển hình là ứng dụng thời tiết trên điện thoại thông minh, sử dụng API để kết nối với hệ thống dữ liệu dự báo thời tiết, gửi các yêu cầu về vị trí địa lý, ngày giờ và nhận về thông tin dự báo cập nhật theo thời gian thực.

Kiến trúc API hiện đại thường chia thành các loại chính như:

- **API REST:** Giao tiếp qua giao thức HTTP, sử dụng các phương thức GET, POST, PUT, DELETE... Dữ liệu truyền thường là định dạng JSON, nhấn mạnh tính không trạng thái và tương thích tốt với môi trường web, mobile.
- **API WebSocket:** Hỗ trợ giao tiếp hai chiều realtime giữa các ứng dụng, hiệu quả cho các dịch vụ chat, game, notification.
- **API SOAP, RPC:** Các kiểu API truyền thống, sử dụng XML, được dùng cho hệ thống yêu cầu tính đồng bộ khóa chặt.

API giúp tích hợp nhanh các ứng dụng mới với hệ thống hiện tại, đẩy nhanh đổi mới công nghệ, mở rộng khả năng kết nối nhiều nền tảng, đồng thời tăng khả năng bảo trì phần mềm nhờ tách biệt rõ giữa giao diện và logic nội bộ. Các API còn được phân loại theo phạm vi sử dụng: API riêng, API công cộng, API đối tác và API tổng hợp.

#### Kết luận:

API là nền tảng cho mọi kết nối dữ liệu trong thời đại số hóa, thúc đẩy khả năng tích hợp, mở rộng và tối ưu hóa trải nghiệm người dùng ở mọi cấp độ phát triển phần mềm.

### 2.3.2 Cấu trúc request và response trong API

Trong môi trường giao tiếp API hiện đại, nhất là API RESTful, mọi hoạt động cần tuân thủ các nguyên tắc cấu trúc request – response chuẩn, đảm bảo tính nhất quán và khả năng tích hợp hệ thống dễ dàng.

#### – Request:

**Endpoint:** URL đại diện cho điểm cuối API, ví dụ: /api/v1/users, thường đi kèm tham số truy vấn hoặc ID tài nguyên.

**Phương thức HTTP:** Các verb chuẩn như GET, POST, PUT/PATCH, DELETE.

**Header:** Phần khai báo thông tin metadata yêu cầu.

**Body:** Dữ liệu gửi kèm ở dạng JSON, XML... với các trường dữ liệu phù hợp yêu cầu endpoint.

#### – Response:

**Status Code:** Mã trạng thái HTTP phản ánh kết quả xử lý: 200/201, 400, 401, 403, 404, 500.

**Header:** Metadata phản hồi, thông tin xác thực lại.

**Body:** Nội dung phản hồi ở dạng JSON/XML bao gồm dữ liệu tài nguyên trả về, thông báo lỗi, hoặc thông tin bổ sung cho client.

#### – Ví dụ minh họa:

##### Client gửi request:

```
POST /api/v1/users
Content-Type: application/json
Authorization: Bearer <token>
{
  "username": "exampleuser",
  "email": "mail@example.com"
}
```

### Server phản hồi:

```
HTTP/1.1 201 Created
Content-Type: application/json
{
  "id": 123,
  "username": "exampleuser",
  "email": "mail@example.com",
  "createdAt": "2025-11-10T20:04:05Z"
}
```

### Kết luận:

Việc tuân thủ cấu trúc chuẩn request và response API giúp đảm bảo hệ thống backend–frontend giao tiếp ổn định, dễ tích hợp, dễ quản lý và tăng khả năng bảo trì và kiểm thử toàn hệ thống.

#### 2.3.3 Tương tác API HTTP với axios

Axios là một thư viện JavaScript dựa trên cơ chế bất đồng bộ, giúp gửi các yêu cầu mạng từ phía người dùng hoặc máy chủ một cách nhất quán và hiệu quả. Thư viện này hoạt động tốt trên nhiều nền tảng, dễ dàng thêm vào các ứng dụng web hiện đại nhờ cách viết code đơn giản, dễ mở rộng, và khả năng kiểm soát chi tiết quá trình gửi/nhận dữ liệu.

##### - Các tính năng chính của Axios

Gửi yêu cầu HTTP đến các dịch vụ API backend, hỗ trợ cả công cụ XMLHttpRequest và module http.

Hoạt động theo chuẩn bất đồng bộ, giúp xử lý các tác vụ không đồng bộ mượt mà, dễ kết hợp với lệnh async/await.

Tự động chuyển đổi dữ liệu gửi đi hoặc nhận về thành định dạng JSON hoặc các loại phù hợp khác.

Có trình chặn để can thiệp vào quá trình: ghi nhật ký, thêm/xóa tiêu đề, xử lý xác thực, hoặc sửa lỗi khi gửi/nhận.

Hỗ trợ hủy yêu cầu, đặt thời gian chờ, và tự động phát hiện lỗi kết nối mạng.

Theo dõi tiến độ tải lên/tải xuống, giới hạn tốc độ băng thông trên Node.js.

Bảo vệ chống tấn công giả mạo yêu cầu chéo trang web khi dùng trình duyệt.

### - **Cách sử dụng cơ bản**

Dưới đây là các ví dụ đơn giản để bắt đầu:

```
// Gửi yêu cầu lấy dữ liệu (GET)
axios.get('/api/users')
  .then(response => {
    // Xử lý dữ liệu nhận được từ server
    console.log(response.data);
  })
  .catch(error => {
    // Xử lý nếu có lỗi
    console.error('Lỗi:', error);
  });

// Gửi yêu cầu thêm dữ liệu mới (POST)
axios.post('/api/users', {
  username: 'abc',
  email: 'mail@test.com'
})
  .then(response => {
    // Nhận và xử lý phản hồi từ server
    console.log('Thành công:', response.data);
  });

// Tạo phiên bản Axios tùy chỉnh
const instance = axios.create({
  baseURL: 'https://api.example.com', // Địa chỉ cơ sở cho tất cả yêu cầu
  timeout: 5000, // Thời gian chờ tối đa 5 giây
})
```

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

```
headers: { 'Authorization': 'Bearer <token>' } // Thêm tiêu đề xác thực
});
```

## Kết luận

Axios là lựa chọn phổ biến và đáng tin cậy để giao tiếp với API HTTP trong các dự án web hiện đại. Nó đảm bảo tính tương thích cao, tăng cường bảo mật, tối ưu hóa quy trình gọi dữ liệu, và dễ dàng mở rộng cho phần giao diện người dùng.

### 2.3.4 Tương tác realtime với Socket.IO Client

Socket.IO Client là thư viện JavaScript giúp tạo kết nối thời gian thực hai chiều giữa phía người dùng và máy chủ. Nó sử dụng công nghệ WebSocket hoặc kết nối chờ dài qua HTTP để truyền dữ liệu nhanh chóng. Socket.IO nổi bật với khả năng gửi nhận sự kiện nhanh, tự động kết nối lại khi mất mạng, và cách thiết kế dựa trên sự kiện dễ dàng thêm vào các ứng dụng web hoặc game hiện đại.

#### Các đặc điểm và chức năng chính

- **Khởi tạo kết nối:**

Sử dụng hàm io() để tạo kết nối từ phía người dùng. Bạn có thể chỉ định địa chỉ máy chủ và các tùy chọn như xác thực, tham số truy vấn, thời gian chờ kết nối lại.

Ví dụ:

```
import { io } from "socket.io-client";
const socket = io("ws://example.com/my-namespace", {
  reconnectionDelayMax: 10000,
  auth: { token: "123" },
  query: { "my-key": "my-value" }
});
```

Thư viện hỗ trợ các cách nhập code như ESM, CommonJS hoặc tải trực tiếp qua CDN trên trình duyệt.

- **Quản lý trạng thái kết nối:**

Kết nối có các sự kiện như connect, connect\_error, disconnect, reconnect,... Có thể kiểm tra trạng thái qua thuộc tính socket.connected. Thư viện tự động xử lý kết nối lại khi mất mạng hoặc máy chủ khởi động lại.

- **Truyền và nhận sự kiện:**

Dùng socket.emit(eventName, ...args) để gửi sự kiện từ phía người dùng lên máy chủ. Dùng socket.on(eventName, callback) để lắng nghe và phản hồi sự kiện từ máy chủ, cập nhật giao diện thời gian thực cho web hoặc game. Hỗ trợ hàm xác nhận (ack callback) để nhận phản hồi từ máy chủ, và truyền dữ liệu linh hoạt dưới dạng JSON hoặc dữ liệu nhị phân.

- **Tính năng nâng cao:**

Hỗ trợ phòng chat và không gian tên riêng để phân chia, phát sóng sự kiện đến nhiều người dùng cùng lúc. Tích hợp lớp trung gian để xác thực mã token, thêm tiêu đề tùy chỉnh nhằm tăng bảo mật và phân quyền.

- **Quản lý phiên người dùng:**

Mỗi kết nối có mã định danh tạm thời socket.id, có thể thay đổi sau khi kết nối lại. Sử dụng phiên trên máy chủ để nhận dạng lâu dài, đồng bộ trạng thái người dùng hoặc game.

## Kết luận

Socket.IO Client là lựa chọn lý tưởng cho dự án cần truyền dữ liệu thời gian thực, sự kiện hai chiều nhanh chóng, bảo mật cao, và dễ dàng tích hợp mở rộng cho phần giao diện web hoặc game hiện đại.

## 2.4. Công nghệ giao diện và thiết kế Responsive

### 2.4.1 Tailwind CSS

Tailwind CSS là một bộ công cụ CSS hiện đại, tập trung vào việc dùng các lớp nhỏ đơn giản để xây dựng giao diện web nhanh chóng và linh hoạt. Thay vì viết code

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

CSS riêng cho từng phần, thì chỉ cần thêm các lớp sẵn có trực tiếp vào HTML để tạo thiết kế đẹp [6].

### Đặc điểm chính

- **Ưu tiên lớp tiện ích:** Mỗi lớp chỉ làm một việc nhỏ, như bg-blue-500 để đổi màu nền xanh, p-4 để thêm khoảng cách bên trong, font-bold để chữ in đậm. Kết hợp chúng để tạo thành phần giao diện hoàn chỉnh mà không cần file CSS riêng.
- **Tùy chỉnh dễ dàng:** Dùng file tailwind.config.js để thay đổi màu sắc, kích thước chữ, khoảng cách, và các điểm ngắt cho màn hình khác nhau. Bạn có thể thêm lớp mới hoặc dùng phần mở rộng.
- **Hỗ trợ màn hình đa dạng:** Có các lớp cho màn hình nhỏ, trung bình, lớn, và biến thể như hover hoặc focus để giao diện thay đổi phù hợp trên điện thoại, máy tính bảng hay máy tính.
- **Tiết kiệm tài nguyên:** Tự động loại bỏ các lớp không dùng để file CSS nhỏ hơn, giúp trang web tải nhanh.
- **Dễ kết hợp:** Làm việc tốt với React, Vue, Next.js. Có nhiều mẫu sẵn và cộng đồng hỗ trợ lớn.

### Cách sử dụng

Cài đặt qua npm: Chạy lệnh npm install tailwindcss, rồi tạo file cấu hình bằng npx tailwindcss init.

Thêm lớp vào HTML:

```
<button class="bg-blue-600 text-white px-4 py-2 rounded shadow-lg hover:bg-blue-700">  
    Click Me  
</button>
```

Chỉnh sửa màu, khoảng cách, hoặc lớp cho màn hình khác nhau trong file cấu hình theo nhu cầu dự án.

Dùng công cụ gợi ý lớp trong VS Code để viết code nhanh hơn.

Ví dụ cho giao diện thay đổi theo màn hình:

```
<div class="w-full md:w-1/2 lg:w-1/3 p-4">  
    Hộp thay đổi kích thước  
</div>
```

## Kết luận

Tailwind CSS giúp bạn tiết kiệm thời gian thiết kế, giữ giao diện nhất quán, tải trang nhanh và dễ thích ứng với mọi thiết bị. Đây là lựa chọn phổ biến cho lập trình viên làm phần giao diện web.

### 2.4.2 Shadcnui

Shadcn/ui là bộ công cụ và nền tảng giao diện hiện đại, giúp lập trình viên xây dựng các phần tử giao diện cho dự án React hoặc Next.js. Nó theo cách tiếp cận mã nguồn mở, cho phép kiểm soát hoàn toàn, dễ dàng thay đổi và mở rộng, thay vì dùng các bộ thư viện giao diện cố định [7].

#### Đặc điểm chính

- **Không giống thư viện thông thường:** Không tải gói qua npm. Thay vào đó, dùng lệnh CLI như npx shadcn@latest add component để sao chép trực tiếp mã nguồn của từng phần tử vào dự án. Có thể sửa đổi chức năng và giao diện theo ý muốn.
- **Quản lý mã nguồn tại chỗ:** Giữ code phần tử ngay trong dự án, dễ chỉnh sửa từng dòng. Điều này giúp công cụ AI như Copilot hoặc GPT hiểu và hỗ trợ tự động hóa việc phát triển giao diện nhanh hơn.
- **Dựa trên Radix UI và Tailwind CSS:** Các phần tử dùng nền tảng cơ bản của Radix UI để đảm bảo dễ tiếp cận, logic chuẩn và hoạt động tốt trên mọi trình duyệt. Chúng được thiết kế bằng lớp tiện ích của Tailwind CSS. Sử dụng chung các yếu tố thiết kế như màu chính, nền, giúp toàn bộ giao diện nhất quán và dễ thay đổi chủ đề.

- **Dễ mở rộng và kết hợp:** Hỗ trợ Next.js, Vite, Remix. Dễ dàng hòa hợp với cấu trúc code mới nhất, phù hợp cho dự án lớn nhỏ. Khi thêm phần tử, có thể cấu hình biến thể, chủ đề và logic động qua code hoặc file dễ đọc.
- **Cách sử dụng**

Chạy lệnh để thêm phần tử:

```
npx shadcn@latest add button
```

Kết quả: File nút bấm được thêm vào thư mục components. Mở file, chỉnh sửa kiểu dáng, thêm biến thể hoặc thay đổi cách hoạt động trực tiếp.

### Kết luận

Shadcn/ui giúp lập trình viên phần giao diện thoát khỏi các phần tử đóng gói sẵn. Chủ động kiểm soát, tùy chỉnh hệ thống giao diện và trải nghiệm người dùng, giữ chủ đề nhất quán, mở rộng dễ dàng, bảo trì đơn giản, và tận dụng lợi ích từ Radix UI cùng Tailwind CSS. Đây là cách tiếp cận mới cho hệ thống thiết kế trên React, Next.js và công cụ AI trong phát triển chuyên nghiệp.

## 2.4.3 Chart.js

Chart.js là thư viện JavaScript mã nguồn mở, dùng để hiển thị dữ liệu dưới dạng biểu đồ động trên thẻ canvas của HTML5. Nó nổi bật nhờ cách dùng đơn giản, dễ tùy chỉnh, và hỗ trợ nhiều loại biểu đồ khác nhau.

### Đặc điểm chính

- **Nhiều loại biểu đồ:** Hỗ trợ biểu đồ đường, cột, tròn hoặc bánh, bong bóng, radar, phân tán, vùng, cực, và kết hợp nhiều loại.
- **Dựa trên canvas:** Vẽ biểu đồ trên thẻ canvas, giúp chạy mượt mà với dữ liệu lớn và hoạt động tốt trên mọi trình duyệt hiện đại.
- **Tùy chỉnh dễ dàng:** Thay đổi màu sắc, chữ viết, gợi ý khi di chuột, chú thích, hiệu ứng động qua các tùy chọn rõ ràng.
- **Thích ứng màn hình:** Biểu đồ tự động điều chỉnh kích thước và bố cục cho máy tính, máy tính bảng hoặc điện thoại.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

- **Tương tác cao:** Cho phép di chuột hiện thông tin, click chuột thích, lọc dữ liệu, và xử lý sự kiện như kéo thả.
- **Nhỏ và nhanh:** Kích thước nhỏ, vẽ nhanh, phù hợp cho bảng điều khiển, báo cáo hoặc ứng dụng web.
- **Hỗ trợ cộng đồng:** Có nhiều phần mở rộng, hướng dẫn chi tiết, và ví dụ thực tế. Dễ tải qua CDN, npm hoặc dùng với React, Vue, Next.js.
- **Cách sử dụng**

Tạo thẻ canvas trong HTML:

```
<canvas id="myChart"></canvas>
```

Thêm script từ CDN:

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Viết code JavaScript để vẽ biểu đồ:

```
<script>
const ctx = document.getElementById('myChart').getContext('2d');
const myChart = new Chart(ctx, {
    type: 'bar', // Loại biểu đồ cột
    data: {
        labels: ['A', 'B', 'C'], // Nhãn
        datasets: [{
            label: 'Giá trị',
            data: [10, 20, 30],
            backgroundColor: ['#4facfe', '#00f2fe', '#1e3c72'] // Màu nền
        }]
    },
    options: { responsive: true } // Thích ứng màn hình
});
</script>
```

## Kết luận

Chart.js là lựa chọn tốt để tạo bảng điều khiển dữ liệu, báo cáo hoặc ứng dụng tương tác trên web. Nó kết hợp tốc độ cao, dễ dùng, tùy chỉnh linh hoạt và thích ứng màn hình trong một gói nhỏ gọn, phù hợp cho lập trình viên phần giao diện.

## 2.5. Bảo mật và hiệu năng trong ứng dụng web

### 2.5.1 Xác thực và phân quyền

Xác thực là bước kiểm tra danh tính người dùng để cho phép vào hệ thống, còn phân quyền là bước quyết định người dùng đã xác thực được làm gì và truy cập tài nguyên nào trong hệ thống.

#### Xác thực

Cách phổ biến gồm tên đăng nhập và mật khẩu, có thể tăng cường bằng mã một lần, nhiều lớp bảo vệ, hoặc sinh trắc học khi cần thiết.

Hệ thống thường dùng phiên làm việc hoặc dạng dấu hiệu như token để ghi nhớ trạng thái đăng nhập qua các lần gửi yêu cầu.

Ứng dụng kiểu một trang hoặc cần gọi nhiều dịch vụ thường ưu tiên token vì linh hoạt hơn so với phiên phụ thuộc một máy chủ.

#### Phân quyền

Sau khi xác thực, hệ thống kiểm tra vai trò và quyền để cho phép hoặc chặn truy cập vào chức năng hay tài nguyên cụ thể.

Nên chỉ cấp đúng phần cần thiết cho mỗi người dùng để giảm rủi ro và giữ an toàn dữ liệu.

## Kết luận

Xác thực trả lời “bạn là ai”, phân quyền trả lời “bạn được phép làm gì”, cả hai kết hợp để ngăn truy cập trái phép và bảo vệ tài nguyên của ứng dụng web.

## 2.5.2 Quản lý session, cookie

Session và cookie là hai cách cơ bản để giữ trạng thái đăng nhập, nhận biết người dùng, bảo vệ quá trình xác thực và cá nhân hóa trải nghiệm trên trang web.

### Session

Session lưu thông tin trạng thái trên máy chủ, như chi tiết đăng nhập, vai trò hoặc giỏ hàng. Sau khi đăng nhập, máy chủ tạo mã định danh session với dãy số ngẫu nhiên và gửi về trình duyệt qua cookie. Mỗi lần gửi yêu cầu mới, trình duyệt gửi mã này để máy chủ nhận diện và tiếp tục phiên làm việc mà không cần đăng nhập lại. Session cần đặt thời gian hết hạn, tự động xóa khi đăng xuất hoặc không hoạt động, để tránh bị đánh cắp hoặc lặp lại tấn công. Dữ liệu thực chỉ lưu trên máy chủ, cookie chỉ chứa mã định danh để tăng an toàn.

### Cookie

Cookie là file nhỏ lưu trên trình duyệt, chứa mã session hoặc dữ liệu nhẹ như chủ đề trang hoặc ngôn ngữ. Nó được gửi kèm mỗi yêu cầu đến máy chủ để duy trì thông tin. Các thiết lập bảo mật quan trọng: Secure, HttpOnly, SameSite, và thời hạn hết hạn rõ ràng. Cookie phù hợp cho ứng dụng không phụ thuộc máy chủ, nhưng dễ bị khai thác nếu không cấu hình đúng, như qua tấn công chèn mã hoặc giả mạo.

### Kết luận

Quản lý session và cookie đúng cách giúp bảo vệ trang web khỏi rủi ro phổ biến như đánh cắp phiên hoặc tấn công chèn mã, đồng thời giữ trải nghiệm người dùng mượt mà và an toàn lâu dài.

## 2.5.3 Tối ưu truy vấn, caching với TanStack Query

TanStack Query là thư viện quản lý dữ liệu không đồng bộ và trạng thái từ máy chủ cho ứng dụng web hiện đại, đặc biệt với React. Nó cũng hỗ trợ Vue, Solid, Svelte, Angular. Thư viện giúp tối ưu hóa việc gọi API, lưu trữ tạm thời dữ liệu, cập nhật ngầm, hiển thị giao diện, và giảm code lặp lại khi làm việc với dữ liệu thay đổi [8].

## Đặc điểm chính

- **Lưu trữ tạm thời thông minh:** Tự động giữ dữ liệu từ các lần gọi theo mã định danh, giảm số lần gửi yêu cầu lặp lại, giúp trang tải nhanh hơn khi chuyển trang.
- **Kết hợp yêu cầu:** Ghép nhiều yêu cầu giống nhau thành một, giảm tải cho máy chủ khi nhiều phần giao diện cùng gọi cùng API.
- **Cập nhật ngầm:** Khi dữ liệu máy chủ thay đổi, thư viện tự động kiểm tra và cập nhật dữ liệu cũ theo thời gian định sẵn hoặc sự kiện như kiểm tra định kỳ.
- **Quản lý trạng thái dễ dàng:** Xử lý tự động trạng thái đang tải, thành công, lỗi, giúp giao diện hiển thị thông báo mà không cần viết nhiều code.
- **Cập nhật giao diện nhanh:** Hỗ trợ thay đổi dữ liệu ngay lập tức khi thêm, sửa, xóa, quay lại nếu máy chủ báo lỗi, và làm mới lưu trữ tạm thời.
- **Dọn dẹp tự động:** Dữ liệu không dùng sẽ bị xóa sau thời gian nhất định, giữ ứng dụng nhẹ và nhanh.
- **Hỗ trợ render phía máy chủ:** Làm việc tốt với Next.js hoặc Remix, truyền dữ liệu lưu trữ từ máy chủ sang phía người dùng để tăng tốc độ hiển thị.
- **Cách sử dụng cơ bản**

Thêm thư viện và dùng hook useQuery để gọi dữ liệu:

```
import { useQuery } from '@tanstack/react-query';

function DanhSachCongViec() {
  const { data, isPending, error } = useQuery({
    queryKey: ['todos'], // Mã định danh cho dữ liệu
    queryFn: () => fetch('/api/todos').then(r => r.json()) // Hàm gọi API
  });

  if (isPending) return <span>Đang tải...</span>;
  if (error) return <span>Có lỗi!</span>;
  return (
    <ul>
      {data.map(item => (
        <li>{item}</li>
      ))}
    </ul>
  );
}
```

```
<ul>
  {data.map(t => <li key={t.id}>{t.title}</li>)}
</ul>
);
}
```

- **Tối ưu hóa hiệu năng:** Điều chỉnh thời gian lưu trữ tạm thời để dữ liệu tươi mới, tự động cập nhật khi tab được chọn lại, hỗ trợ phân trang hoặc cuộn vô tận. Kết hợp hook useMutation để thay đổi dữ liệu, làm mới lưu trữ và đồng bộ giao diện.

## Kết luận

TanStack Query là công cụ lý tưởng để quản lý gọi API, lưu trữ tạm thời và trạng thái dữ liệu cho ứng dụng web hiện đại. Nó nâng cao tốc độ, giảm tải máy chủ, cải thiện trải nghiệm người dùng, và giữ code dự án sạch sẽ dễ bảo trì.

## 2.6. Tổng quan các công cụ hỗ trợ phát triển và triển khai

### 2.6.1 Git/GitHub

Git là hệ thống quản lý phiên bản mã nguồn phân tán, giúp theo dõi lịch sử thay đổi dự án từ nhỏ đến lớn. GitHub là nền tảng trực tuyến dựa trên Git, hỗ trợ lưu trữ, cộng tác, kiểm tra code và tự động hóa quy trình cho lập trình viên.

- **Đặc điểm chính của Git**

Theo dõi toàn bộ lịch sử: Ghi lại mọi thay đổi file, người thực hiện, thời gian và lý do, dễ quay lại phiên bản cũ nếu lỗi.

Làm việc phân tán: Mỗi người có bản sao đầy đủ trên máy cá nhân, không phụ thuộc máy chủ trung tâm, tăng an toàn và tốc độ.

Nhánh và gộp mạnh mẽ: Tạo nhánh riêng cho tính năng mới, dễ gộp thay đổi và giải quyết xung đột.

Quy trình linh hoạt: Hỗ trợ nhiều cách làm việc như Git Flow hoặc GitHub Flow.

- **Các lệnh Git cơ bản**

git init: Tạo kho lưu trữ mới.

git clone <url>: Tải kho từ xa về máy.

git add <file>: Thêm file vào danh sách chuẩn bị lưu.

git commit -m "msg": Lưu thay đổi với ghi chú.

git status: Kiểm tra trạng thái file.

git branch: Xem hoặc tạo nhánh.

git merge <branch>: Gộp nhánh.

git pull: Lấy và cập nhật code từ xa.

git push: Đẩy code lên kho từ xa.

- **Đặc điểm chính của GitHub**

Lưu trữ code: Quản lý kho riêng tư hoặc công khai, sao lưu và truy cập mọi nơi.

Yêu cầu kéo và kiểm tra: Thảo luận thay đổi, duyệt code trước khi gộp vào nhánh chính.

Quản lý nhóm và nhiệm vụ: Giao việc, theo dõi tiến độ qua issue và mốc thời gian.

Tự động hóa: Kiểm tra, xây dựng và triển khai với GitHub Actions hoặc công cụ khác.

Bảo mật: Hỗ trợ xác thực hai lớp và kiểm soát quyền truy cập.

**Kết luận**

Git kết hợp GitHub là công cụ thiết yếu cho mọi dự án phần mềm, giúp quản lý code an toàn, cộng tác chuyên nghiệp và hiện đại cho lập trình viên.

**2.6.2 Browser DevTools**

Browser DevTools là bộ công cụ săn sóc trong các trình duyệt hiện đại như Chrome, Firefox, Edge, Safari. Nó giúp lập trình viên và người kiểm thử theo dõi, sửa lỗi, tối ưu hóa và thử nghiệm trang web ngay trên trình duyệt mà không cần thay đổi code gốc.

## Các thành phần chính

- **Elements:** Xem và chỉnh sửa nhanh cấu trúc HTML và kiểu dáng CSS của trang. Bạn có thể thử thay đổi bối cảnh, thêm lớp hoặc kiểm tra lỗi hiển thị, đặc biệt hữu ích cho thiết kế đáp ứng trên nhiều màn hình.
- **Console:** Chạy lệnh JavaScript trực tiếp, ghi nhật ký dữ liệu, kiểm tra biến, xem lỗi hoặc cảnh báo. Dùng để thử code nhỏ và gỡ lỗi hàm nhanh chóng.
- **Network:** Theo dõi mọi yêu cầu gửi và nhận dữ liệu. Kiểm tra thời gian tải, lỗi kết nối, đo tốc độ trang và thử dữ liệu giả.
- **Sources:** Quản lý file code, đặt điểm dừng để theo dõi từng bước thực thi, xem lịch sử gọi hàm và sửa code tạm thời.
- **Performance:** Phân tích tốc độ tải và chạy trang, xác định phần chậm. Thu thập dữ liệu về hiệu suất tổng thể.
- **Application:** Quản lý lưu trữ dữ liệu như localStorage, cookie, bộ nhớ đệm. Kiểm tra trạng thái ngoại tuyến và tài nguyên cho ứng dụng web tiến bộ.
- **Security:** Kiểm tra chứng chỉ an toàn, phát hiện nội dung không mã hóa và đảm bảo dữ liệu truyền tải bảo mật.
- **Chế độ thiết bị:** Giả lập màn hình di động, máy tính bảng, xoay chiều hoặc tốc độ mạng chậm để thử giao diện trên nhiều thiết bị.
- **Lighthouse:** Chạy kiểm tra chất lượng trang, tiếp cận dễ dàng, thực hành tốt và tối ưu hóa tìm kiếm.

## Kết luận

Browser DevTools là công cụ thiết yếu cho lập trình viên phần giao diện, giúp rút ngắn thời gian phát triển, kiểm thử, sửa lỗi và nâng cao chất lượng dự án web một cách toàn diện.

## 2.7. Tổng kết chương

Chương này cung cấp cái nhìn toàn diện về các yếu tố nền tảng trong phát triển ứng dụng web hiện đại, từ thiết kế giao diện responsive, công nghệ bảo mật đến các công cụ hỗ trợ phát triển và triển khai. Các phần trình bày đã phân tích sâu vai trò của HTML5 và CSS3, các framework nổi bật như Tailwind CSS, Shadcn/ui, Chart.js, cùng với các phương pháp xác thực, quản lý session, tối ưu hóa truy vấn và caching sử dụng TanStack Query. Ngoài ra, các công cụ thực tiễn như Git/GitHub, Node.js và Browser DevTools được giới thiệu nhằm nâng cao hiệu suất lập trình, hỗ trợ làm việc nhóm và đảm bảo chất lượng sản phẩm. Thông qua nội dung chương, người đọc có nền tảng nhận thức vững chắc về cơ sở lý thuyết lẫn thực hành để hiện thực hóa các dự án web ứng dụng an toàn, hiệu quả, dễ mở rộng và bảo trì.

## CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô tả bài toán

Trong bối cảnh giáo dục hiện đại và xu hướng chuyển đổi số, việc xây dựng một nền tảng học tập trực tuyến hiệu quả là vô cùng cần thiết. Mức độ hứng thú của người học là yếu tố quan trọng để duy trì tiến trình học tập, góp phần giúp họ đạt được thành tích tốt, đồng thời phản ánh sự thành công của hệ thống.

Bài toán đặt ra là phát triển giao diện Frontend cho nền tảng học trực tuyến có tích hợp yếu tố trò chơi hóa, giúp nâng cao trải nghiệm học tập, tăng động lực cho sinh viên và cung cấp công cụ quản lý, phân tích hiệu quả cho giảng viên cùng quản trị viên.

Cụ thể, giao diện gồm hai phần chính:

Giao diện quản lý học thuật: dành cho quản trị viên và giảng viên, hỗ trợ quản lý chương trình đào tạo, khóa học, phân công, ngân hàng câu hỏi, sinh viên, bài kiểm tra, luyện tập... với các công cụ trực quan, chức năng quản lý mạnh mẽ, thực hiện các thao tác CRUD cần thiết.

Giao diện luyện tập, đánh giá có tích hợp trò chơi: dành cho sinh viên, giúp hiển thị tiến trình học tập, kết quả các bài kiểm tra/luyện tập, so sánh năng lực cá nhân với trung bình lớp và học sinh xuất sắc, đồng thời tăng sự tương tác và hấp dẫn nhờ các yếu tố game như cấp độ, vật phẩm, cửa hàng, thành tích.

Ngoài ra, hệ thống hỗ trợ trực quan hóa dữ liệu bằng nhiều loại biểu đồ giúp người dùng dễ dàng nhận biết, phân tích tiến độ, hiệu quả học tập của cá nhân và lớp học, từ đó thúc đẩy động lực, cải thiện chất lượng học tập.

**Yếu tố công nghệ:** Ứng dụng các framework, thư viện hiện đại như Next.js, Tailwind CSS, Shadcn/ui, TanStack Query, Phaser.js, Chart.js và Socket.IO để tối ưu tốc độ, trải nghiệm, khả năng mở rộng cũng như tích hợp các tính năng thời gian thực phù hợp với yêu cầu.

### 3.2. Phân tích thiết kế hệ thống

#### 3.2.1 Đặc tả yêu cầu hệ thống

##### 3.2.1.1 Yêu cầu chức năng

Hệ thống cung cấp các chức năng riêng biệt, phù hợp với ba vai trò người dùng chính: Quản trị viên, Giảng viên và Sinh viên.

##### **Chức năng Quản lý (Quản trị viên):**

Quản lý chương trình đào tạo: Cho phép thêm, sửa, xóa và xem chi tiết các chương trình đào tạo. Quản lý các mục tiêu đào tạo, chuẩn đầu ra chương trình và các môn học thuộc mỗi chương trình.

Quản lý khóa đào tạo: Thiết lập các khóa đào tạo theo từng năm học, quản lý danh sách các học kỳ trong mỗi khóa.

Quản lý phân công giảng dạy: Phân công giảng viên phụ trách các môn học trong từng học kỳ cụ thể thông qua ma trận trực quan.

##### **Chức năng Giảng dạy (Giảng viên):**

Quản lý khóa học và sinh viên: Xem danh sách các khóa học được phân công, quản lý danh sách sinh viên trong từng lớp, bao gồm thêm mới, import hàng loạt từ file Excel và xóa sinh viên khỏi lớp.

Quản lý ngân hàng câu hỏi: Thêm mới, sửa, xóa và import câu hỏi từ file Excel/ZIP, bao gồm cả các tệp media đi kèm.

Quản lý bài kiểm tra: Tạo các bài kiểm tra mới với hai chế độ đánh giá và luyện tập, thiết lập tiêu chí sinh câu hỏi tự động từ ngân hàng câu hỏi dựa trên chuẩn đầu ra và độ khó.

Theo dõi và báo cáo: Theo dõi tiến độ làm bài của sinh viên trong thời gian thực, xem các cảnh báo về những sinh viên gặp khó khăn, và xem báo cáo, phân tích kết quả chi tiết sau khi bài kiểm tra kết thúc.

### **Chức năng Học tập (Sinh viên):**

Luyện tập và đánh giá: Tham gia các bài kiểm tra dưới dạng một trò chơi, trả lời câu hỏi để vượt qua chướng ngại vật và hoàn thành màn chơi.

Theo dõi tiến độ và kết quả: Xem lại lịch sử các bài kiểm tra đã hoàn thành, xem phân tích chi tiết kết quả học tập của bản thân qua các biểu đồ trực quan, so sánh năng lực với trung bình lớp và nhận các gợi ý cải thiện.

Tương tác và tùy chỉnh: Tham gia bảng xếp hạng, theo dõi tiến trình cấp độ, quản lý kho đồ và mua sắm vật phẩm mới trong cửa hàng bằng tiền tệ trong trò chơi.

#### *3.2.1.2 Yêu cầu phi chức năng*

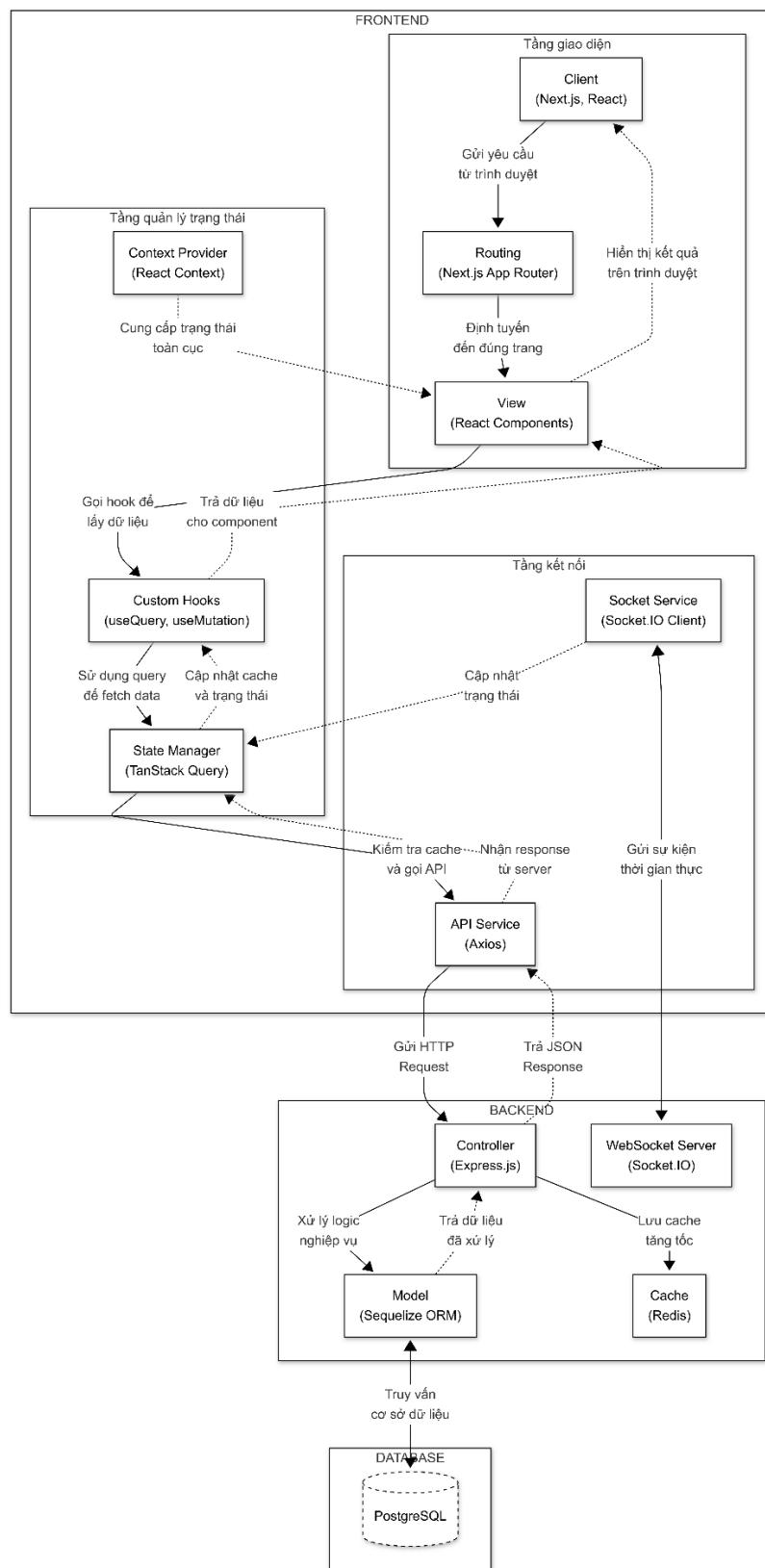
Giao diện trực quan và thân thiện: Giao diện được thiết kế hiện đại, nhất quán, dễ sử dụng cho cả ba vai trò người dùng. Các luồng thao tác được tối ưu hóa để người dùng có thể thực hiện công việc một cách nhanh chóng. Giao diện phải tương thích trên nhiều thiết bị.

Hiệu suất và tốc độ: Ứng dụng phải có thời gian tải nhanh và phản hồi mượt mà. Nhờ Next.js, các trang được tối ưu hóa cho việc kết xuất phía máy chủ. TanStack Query giúp quản lý dữ liệu hiệu quả, giảm thiểu các yêu cầu không cần thiết. Trò chơi xây dựng bằng Phaser.js phải đảm bảo hiệu suất ổn định trên các trình duyệt phổ biến.

Bảo mật và phân quyền: Hệ thống phải có cơ chế xác thực người dùng và phân quyền truy cập rõ ràng. Chỉ những người dùng có vai trò phù hợp mới có thể truy cập các chức năng tương ứng.

Khả năng mở rộng: Kiến trúc hệ thống được xây dựng theo dạng module hóa, cho phép dễ dàng bảo trì, nâng cấp và tích hợp các tính năng mới trong tương lai mà không ảnh hưởng lớn đến các thành phần hiện có.

### **3.2.2 Kiến trúc hệ thống**



Hình 3.1 Kiến trúc hệ thống

Kiến trúc hệ thống bao gồm các thành phần và luồng dữ liệu sau:

**Client (Next.js, React):** Giao diện người dùng được xây dựng bằng Next.js và React, cho phép người dùng tương tác với hệ thống thông qua trình duyệt web. Yêu cầu từ người dùng sẽ được gửi đến Routing để điều hướng.

**Routing (Next.js App Router):** Chịu trách nhiệm tiếp nhận các yêu cầu từ người dùng thông qua các đường dẫn và định hướng chúng đến trang phù hợp. Hệ thống sử dụng cơ chế điều hướng dựa trên cấu trúc thư mục, giúp tổ chức các trang một cách trực quan và dễ quản lý.

**View (React Components):** Có nhiệm vụ tạo ra giao diện người dùng dựa trên dữ liệu nhận được. Các thành phần giao diện được xây dựng theo hướng module hóa, giúp tái sử dụng mã nguồn và dễ bảo trì.

**Custom Hooks (useQuery, useMutation):** Đây là các hàm tùy chỉnh được xây dựng để đóng gói logic lấy dữ liệu và thao tác với máy chủ. Giúp tách biệt logic nghiệp vụ khỏi giao diện, làm cho mã nguồn dễ đọc và dễ kiểm thử hơn.

**State Manager (TanStack Query):** Thành phần này đảm nhận việc quản lý trạng thái dữ liệu từ máy chủ, bao gồm lưu đệm, đồng bộ hóa dữ liệu và xử lý các trạng thái đang tải hoặc lỗi. Hệ thống tự động lưu đệm dữ liệu và chỉ gọi máy chủ khi cần thiết, giúp tối ưu hiệu suất ứng dụng.

**Context Provider (React Context):** Cung cấp trạng thái toàn cục cho ứng dụng như thông tin xác thực người dùng, cài đặt hệ thống. Giúp truyền dữ liệu xuống các thành phần con mà không cần truyền qua từng cấp.

**API Service (Axios):** Tầng kết nối chịu trách nhiệm gửi các yêu cầu đến máy chủ. Được cấu hình để tự động thêm mã xác thực và xử lý lỗi một cách thống nhất.

**Socket Service (Socket.IO Client):** Cung cấp kết nối thời gian thực giữa trình duyệt và máy chủ, cho phép nhận các sự kiện cập nhật như điểm số, thông báo mới mà không cần tải lại trang.

**Controller (Express.js):** Xử lý các yêu cầu đến từ giao diện người dùng, sau đó tương tác với Model để thực hiện các hành động hoặc truy xuất dữ liệu từ cơ sở dữ liệu.

**WebSocket Server (Socket.IO):** Xử lý các kết nối thời gian thực từ trình duyệt, cho phép gửi sự kiện đến người dùng như cập nhật bảng xếp hạng, thông báo khi có người tham gia bài kiểm tra.

**Model (Sequelize ORM):** Thành phần này đảm nhận việc xử lý dữ liệu và logic nghiệp vụ. Giúp tương tác với cơ sở dữ liệu thông qua các đối tượng, hỗ trợ các thao tác thêm, sửa, xóa và quan hệ giữa các bảng.

**Cache (Redis):** Hệ thống lưu trữ tạm thời được sử dụng để lưu các dữ liệu thường xuyên truy cập, giúp giảm tải cho cơ sở dữ liệu và tăng tốc độ phản hồi của hệ thống.

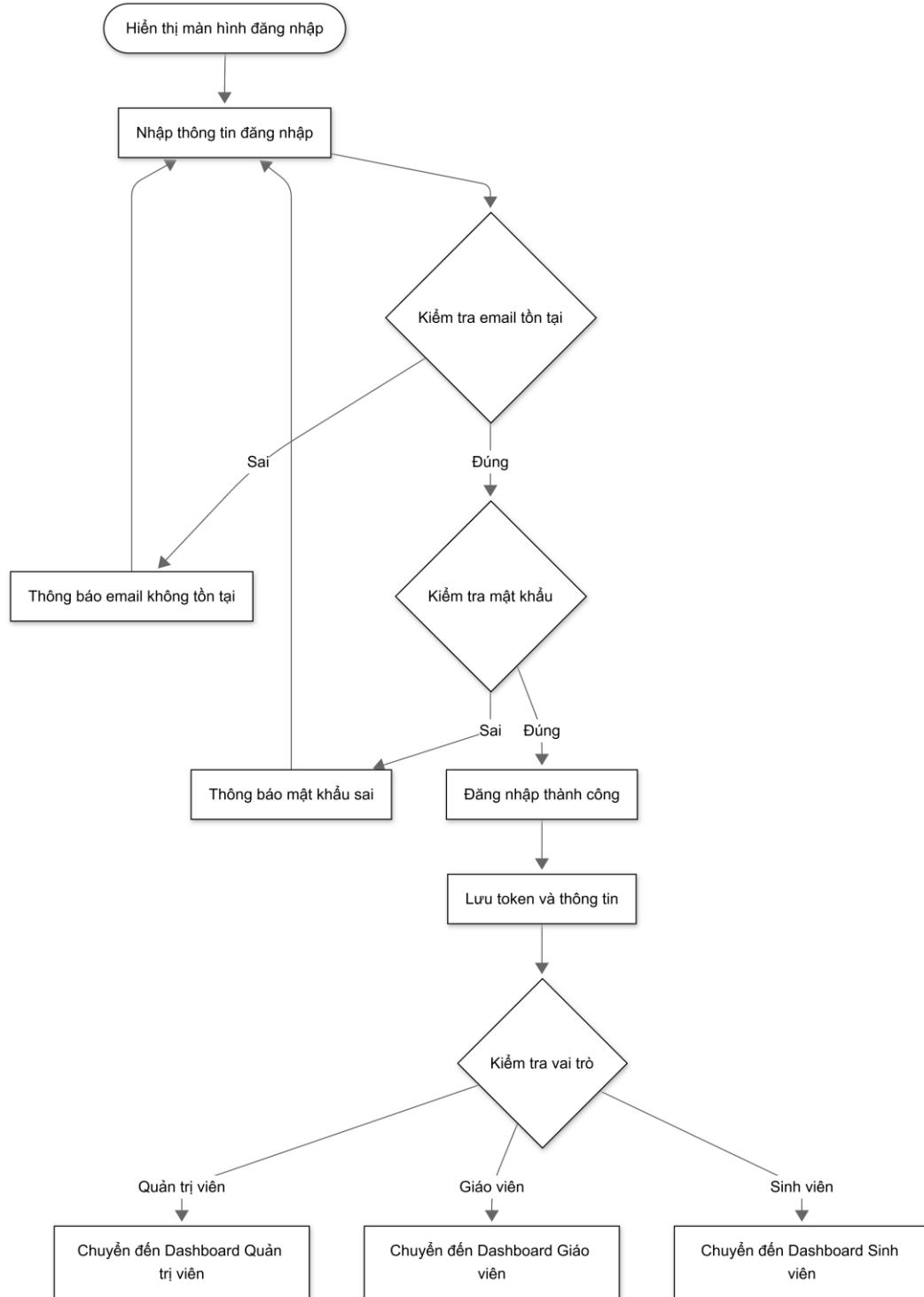
**Database (PostgreSQL):** Đây là nơi lưu trữ tất cả dữ liệu của ứng dụng. PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, hỗ trợ các tính năng nâng cao như lưu trữ dữ liệu dạng JSON, tìm kiếm toàn văn và giao dịch.

**Luồng dữ liệu từ máy chủ đến giao diện:** Khi người dùng thao tác trên giao diện, View sẽ gọi Custom Hooks tương ứng. Hook này sử dụng State Manager để kiểm tra bộ nhớ đệm, nếu dữ liệu chưa có hoặc đã cũ thì gọi API Service. API Service gửi yêu cầu đến Controller, Controller xử lý và trả về dữ liệu dạng JSON. Dữ liệu được API Service nhận và chuyển cho State Manager để cập nhật bộ nhớ đệm. Cuối cùng, Hook trả dữ liệu cho View để hiển thị kết quả lên giao diện người dùng.

### 3.2.3 Kịch bản tương tác người dùng

Trong nền tảng học trực tuyến có tích hợp yếu tố trò chơi, kịch bản tương tác người dùng đóng vai trò quan trọng trong việc mô tả cách người dùng thực hiện các thao tác và tương tác với hệ thống trong quá trình sử dụng. Nội dung mục này tập trung phân tích các quy trình nghiệp vụ từ góc nhìn người dùng, bao gồm đăng nhập hệ thống, tạo và quản lý bài kiểm tra, tham gia bài kiểm tra và các hoạt động quản lý của quản trị viên. Các kịch bản được mô hình hóa bằng sơ đồ hoạt động nhằm làm rõ luồng hành vi, các bước thao tác và các nhánh xử lý chính trong quá trình sử dụng hệ thống.

### 3.2.3.1 Quy trình đăng nhập hệ thống



Hình 3.2 Sơ đồ hoạt động đăng nhập

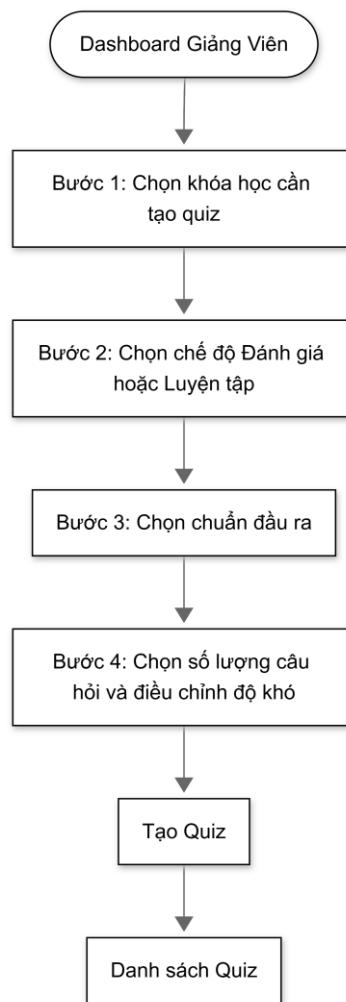
Đầu tiên, hệ thống hiển thị giao diện đăng nhập để người dùng nhập email và mật khẩu. Sau khi nhận thông tin, hệ thống kiểm tra xem email có tồn tại trong cơ sở dữ liệu

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

hay không. Nếu email không tồn tại, hệ thống thông báo “Email không tồn tại” và yêu cầu nhập lại. Nếu email hợp lệ, hệ thống tiếp tục kiểm tra mật khẩu. Trong trường hợp mật khẩu không chính xác, hệ thống hiển thị thông báo “Mật khẩu sai” và yêu cầu nhập lại. Ngược lại, nếu mật khẩu đúng, hệ thống sẽ lưu token xác thực và thông tin người dùng vào bộ nhớ cục bộ. Sau đó, hệ thống kiểm tra vai trò của người dùng và chuyển hướng đến dashboard tương ứng: Admin được chuyển đến Dashboard Admin, Giáo viên đến Dashboard Giáo viên, và Sinh viên đến Dashboard Sinh viên.

### 3.2.3.2 Quy trình tạo và quản lý bài kiểm tra



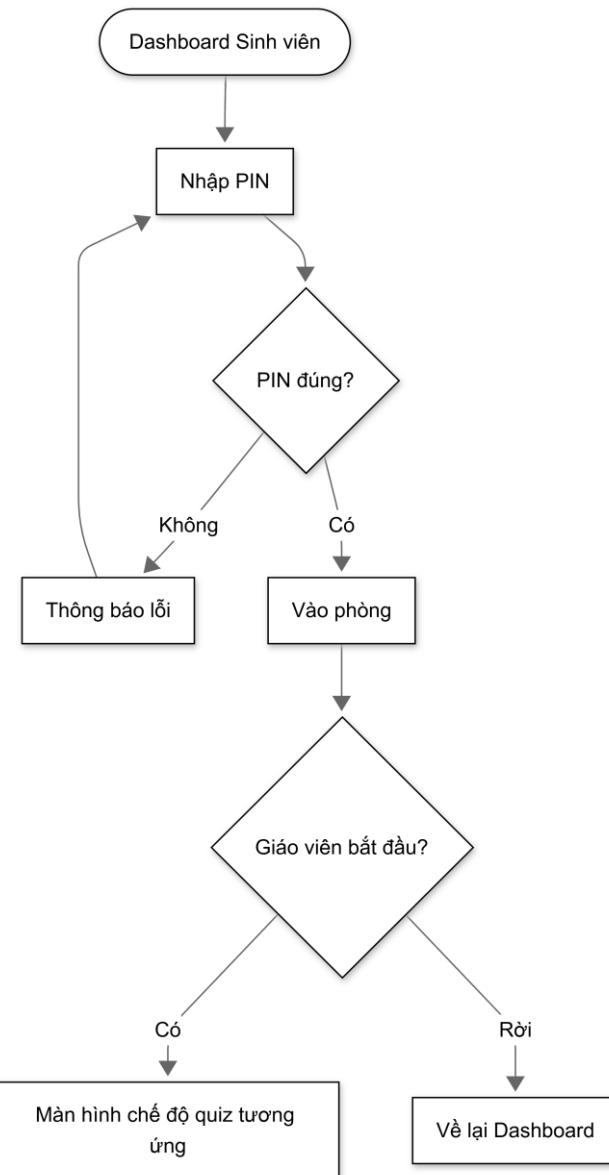
Hình 3.3 Sơ đồ hoạt động tạo Quiz mới

Sơ đồ mô tả quy trình tạo và quản lý bài kiểm tra của giáo viên theo 4 bước: Bước 1: Giáo viên chọn khóa học, nhập tên bài kiểm tra và thời gian làm bài. Bước 2: Chọn chế độ đánh giá hoặc luyện tập. Bước 3: Chọn các chuẩn đầu ra từ danh sách được tải

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

tự động theo môn học. Bước 4: Nhập số lượng câu hỏi và điều chỉnh tỷ lệ độ khó bằng thanh trượt, hệ thống tự động cân bằng tổng sao cho đủ 100%. Sau khi tạo thành công, hệ thống tự động chọn câu hỏi từ ngân hàng và chuyển về danh sách Quiz.

### 3.2.3.3 Quy trình tham gia bài kiểm tra của sinh viên



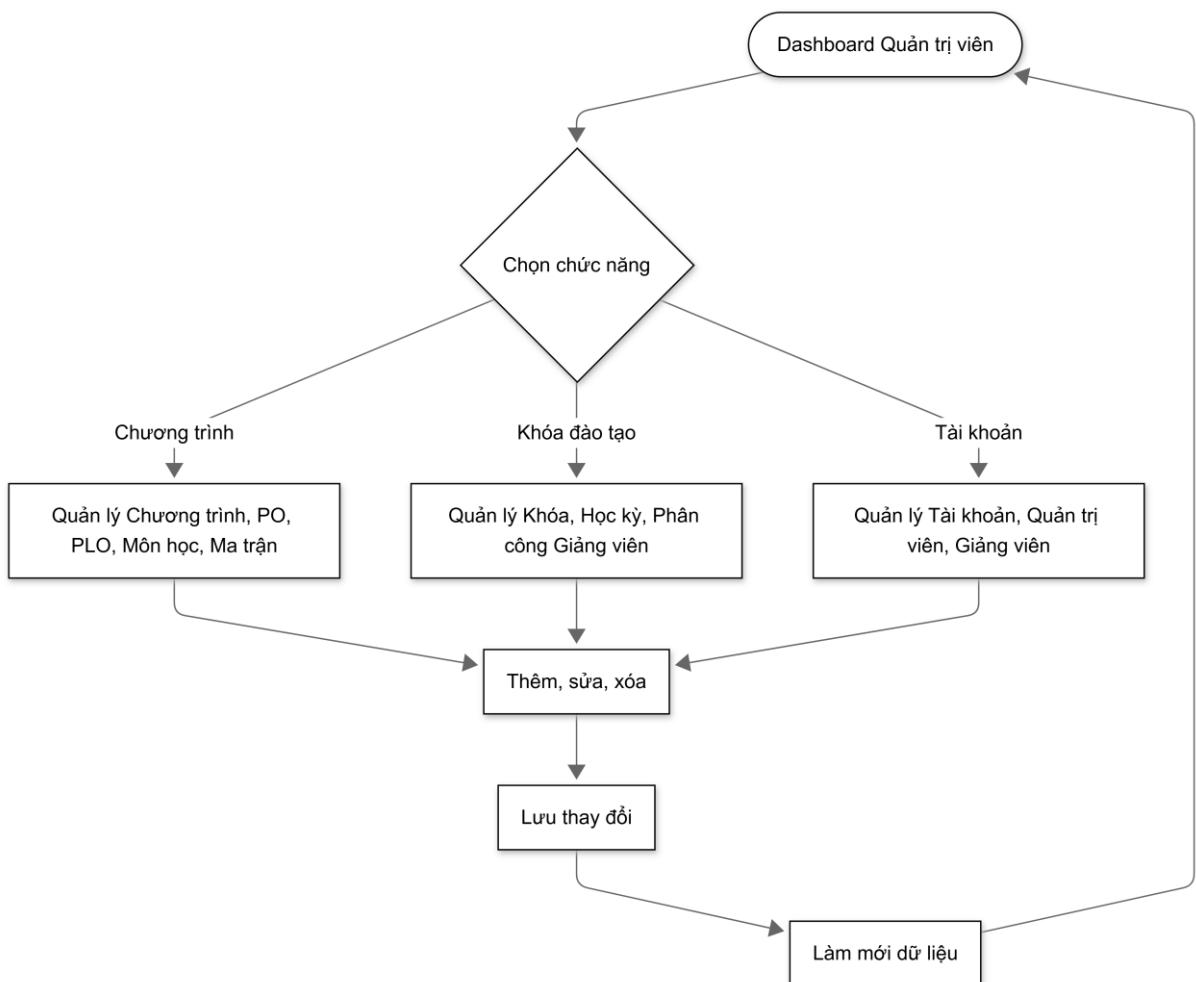
Hình 3.4 Sơ đồ hoạt động tham gia Quiz

Sơ đồ sau mô tả quy trình tham gia bài kiểm tra của sinh viên. Từ Dashboard Sinh viên, nhấn vào nút nhập mã PIN và nhập mã do giáo viên cung cấp. Sau khi nhập PIN, hệ thống kiểm tra tính hợp lệ của mã. Nếu PIN không đúng, hệ thống hiển thị thông báo “PIN không đúng” và yêu cầu nhập lại. Nếu PIN hợp lệ, sinh viên được tham gia vào

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

phòng chờ Quiz. Trong phòng chờ, sinh viên có thể xem danh sách những người tham gia khác. Hệ thống sử dụng Socket.IO để cập nhật danh sách người tham gia theo thời gian thực. Sinh viên chờ đợi cho đến khi giáo viên bắt đầu Quiz. Trong thời gian chờ, sinh viên có thể chọn rời khỏi phòng chờ và quay lại Dashboard. Khi giáo viên bắt đầu Quiz, hệ thống tự động chuyển tất cả sinh viên trong phòng chờ đến màn hình chế độ quiz tương ứng để bắt đầu làm bài kiểm tra.

### 3.2.3.4 Quy trình nghiệp vụ quản lý của admin



Hình 3.5 Sơ đồ hoạt động quản lý của Admin

Sơ đồ mô tả quy trình quản trị hệ thống. Admin chọn chức năng: Quản lý Chương trình Đào tạo gồm thêm xóa sửa chương trình, quản lý PO, PLO, Môn học, Ma trận liên kết, Quản lý Khóa đào tạo gồm thêm xóa sửa khóa, quản lý Học kỳ và Phân công giảng viên, Quản lý Tài khoản gồm thêm xóa sửa tài khoản cho Admin và Giảng viên. Trong

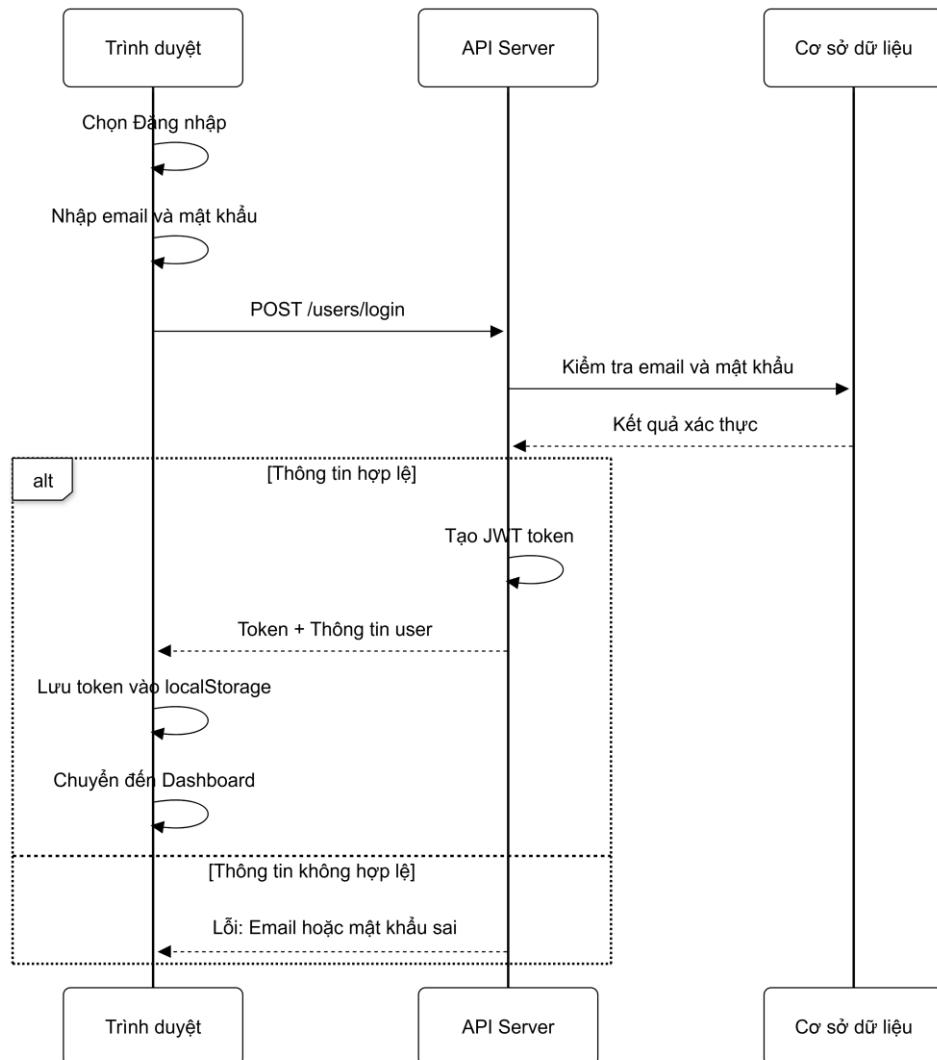
Môn học có quản lý Chapter và LO với ma trận liên kết. Sau mỗi hành động, hệ thống lưu thay đổi và làm mới dữ liệu hiển thị.

### 3.2.4 Tương tác giữa frontend và backend

Bên cạnh việc mô tả hành vi người dùng, việc phân tích tương tác giữa giao diện frontend và hệ thống backend là cần thiết để làm rõ cơ chế trao đổi dữ liệu và xử lý nghiệp vụ trong hệ thống. Nội dung mục này trình bày các tiến trình tương tác chính giữa frontend và backend trong các chức năng cốt lõi của nền tảng, bao gồm đăng nhập tài khoản, tạo bài quiz, tham gia bài quiz và quản lý chương trình đào tạo. Các tiến trình được mô hình hóa bằng sơ đồ tuần tự, qua đó thể hiện rõ trình tự gọi - phản hồi giữa các thành phần hệ thống và vai trò của frontend trong việc hiển thị, cập nhật và phản hồi thông tin đến người dùng.

#### 3.2.4.1 Tiến trình đăng nhập tài khoản

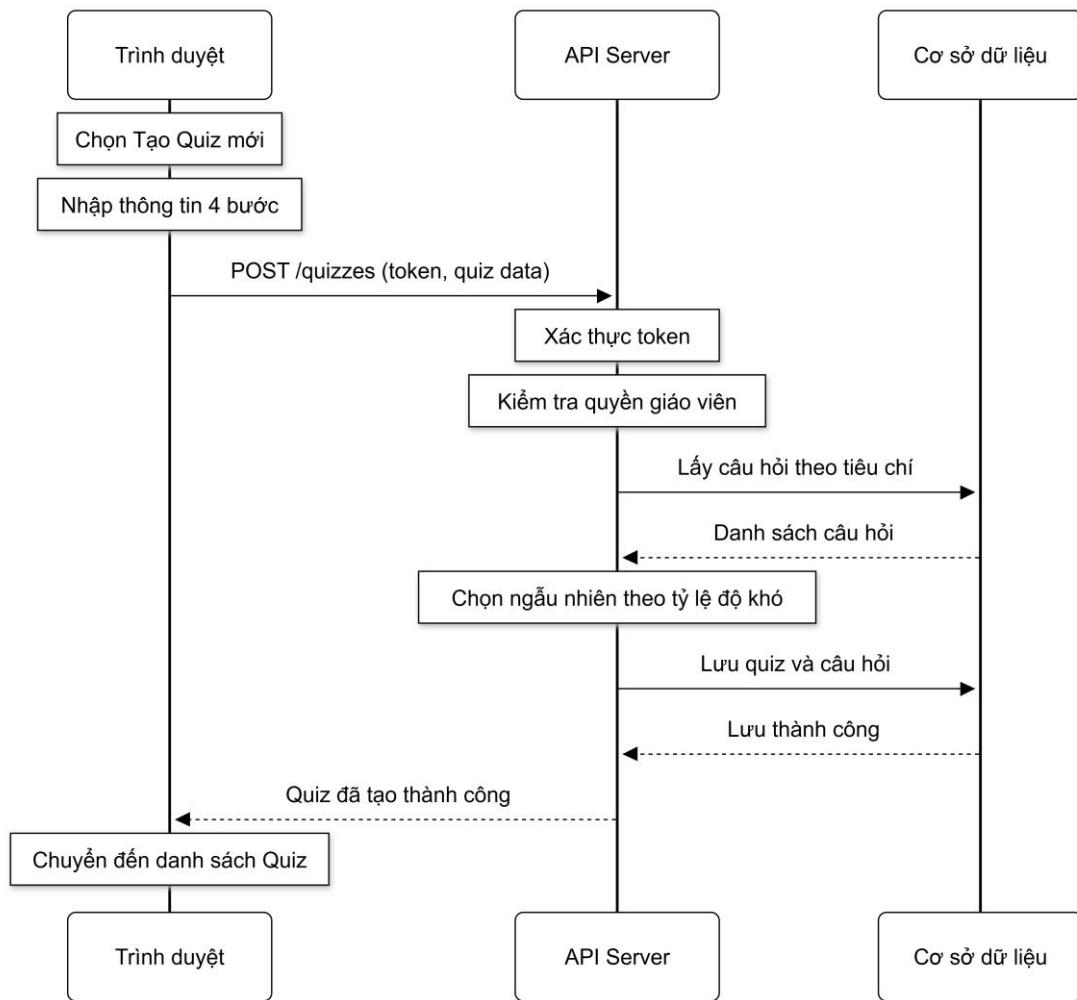
Người dùng chọn “Đăng nhập” và nhập email và mật khẩu. Trình duyệt gửi thông tin đến API Server. Server kiểm tra thông tin đăng nhập bằng cách truy vấn cơ sở dữ liệu để xác minh email và mật khẩu có khớp không. Nếu thông tin hợp lệ, Server tạo JWT token và gửi về trình duyệt cùng thông báo “Đăng nhập thành công”. Trình duyệt lưu token và chuyển hướng người dùng đến dashboard tương ứng với vai trò. Nếu thông tin không chính xác, trả về thông báo lỗi.



Hình 3.6 Sơ đồ tuần tự quá trình đăng nhập tài khoản

#### 3.2.4.2 Tiến trình tạo bài quiz mới

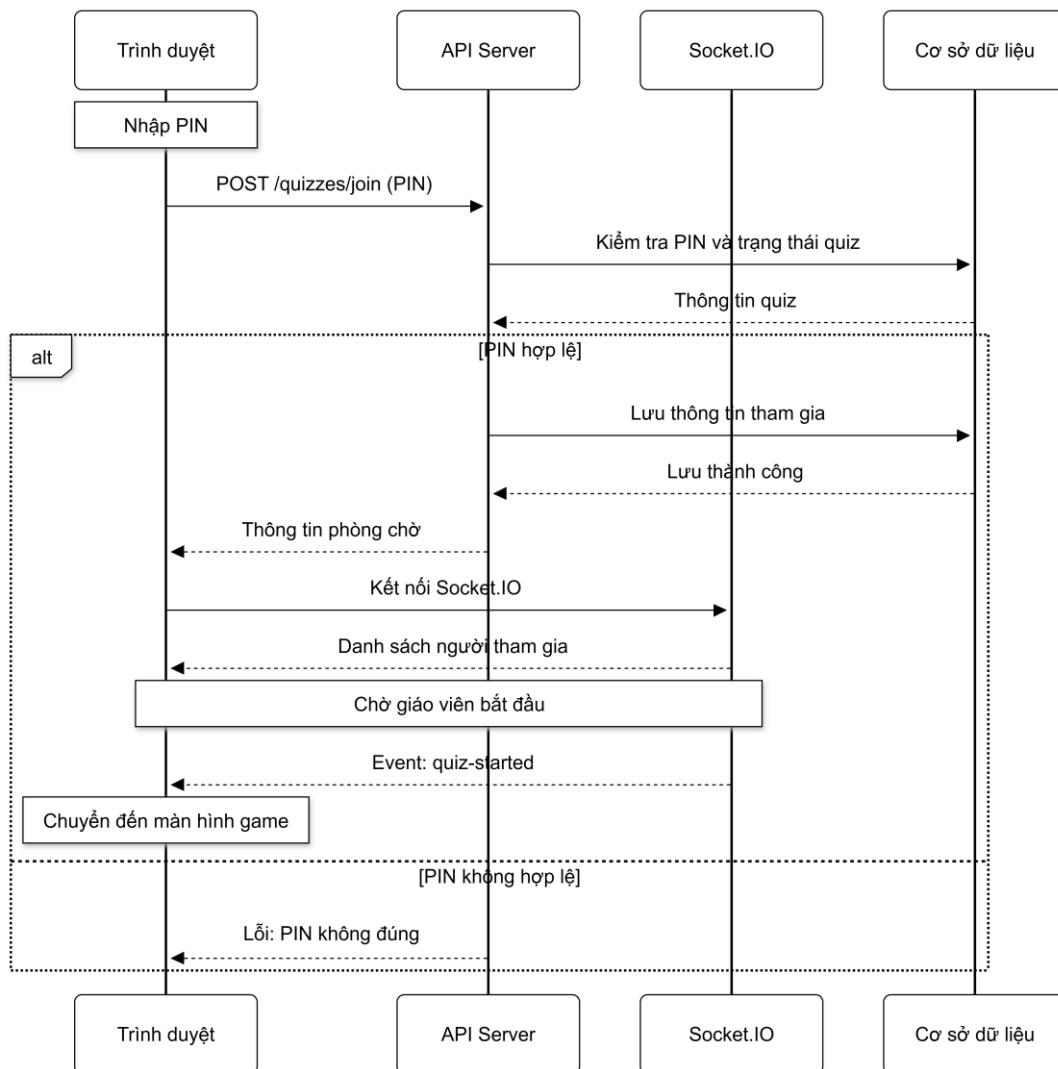
Giáo viên chọn “Tạo Quiz mới” và nhập thông tin qua 4 bước. Trình duyệt gửi yêu cầu POST đến API Server với thông tin quiz. Server xác thực token, kiểm tra quyền giáo viên, sau đó tự động chọn câu hỏi từ cơ sở dữ liệu dựa trên tiêu chí LO, số lượng, tỷ lệ độ khó. Server lưu quiz và danh sách câu hỏi vào cơ sở dữ liệu, trả về thông báo thành công. Trình duyệt chuyển hướng đến danh sách Quiz.



Hình 3.7 Sơ đồ tuần tự quá trình tạo quiz mới

#### 3.2.4.3 Tiến trình tham gia bài quiz

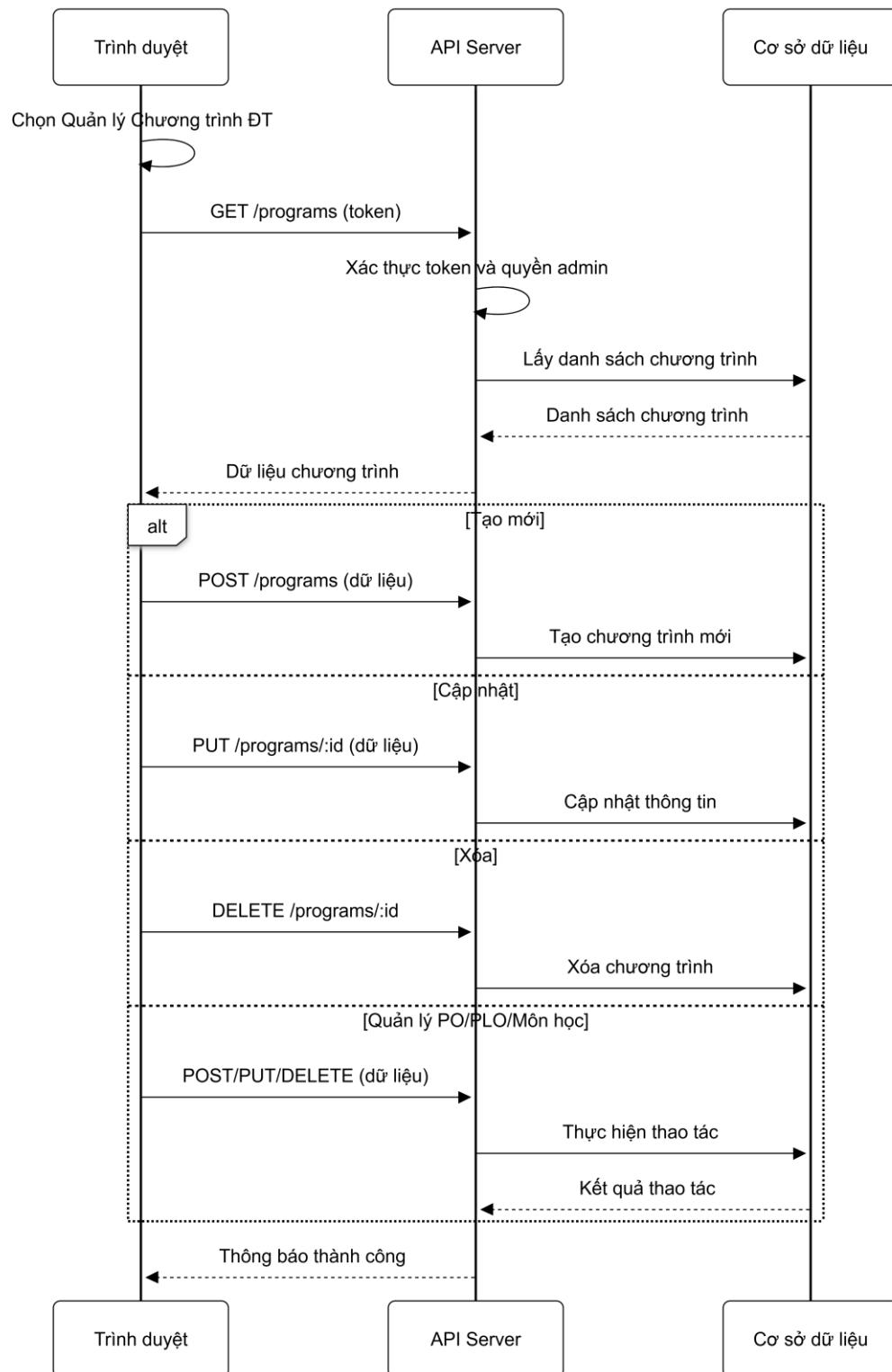
Sinh viên nhập mã PIN. Trình duyệt gửi yêu cầu POST đến API Server để tham gia quiz. Server kiểm tra PIN hợp lệ và trạng thái quiz. Nếu hợp lệ, lưu thông tin tham gia vào database và trả về thông tin phòng chờ. Trình duyệt kết nối Socket.IO để nhận cập nhật realtime về danh sách người tham gia. Khi giáo viên bắt đầu quiz, Server phát sự kiện “quiz-started” qua Socket.IO, trình duyệt nhận sự kiện và chuyển đến màn hình game.



Hình 3.8 Sơ đồ tuần tự quá trình tham gia quiz

#### 3.2.4.4 Tiến trình quản lý chương trình đào tạo

Admin chọn “Quản lý Chương trình ĐT” và thực hiện hành động thêm xóa sửa. Trình duyệt gửi yêu cầu đến API Server với token xác thực. Server kiểm tra quyền admin, thực hiện thao tác trên database. Khi xem chi tiết chương trình, có thể quản lý PO, PLO, Môn học và Ma trận liên kết. Mỗi thao tác đều được lưu vào database và trả về kết quả. Trình duyệt làm mới dữ liệu hiển thị.

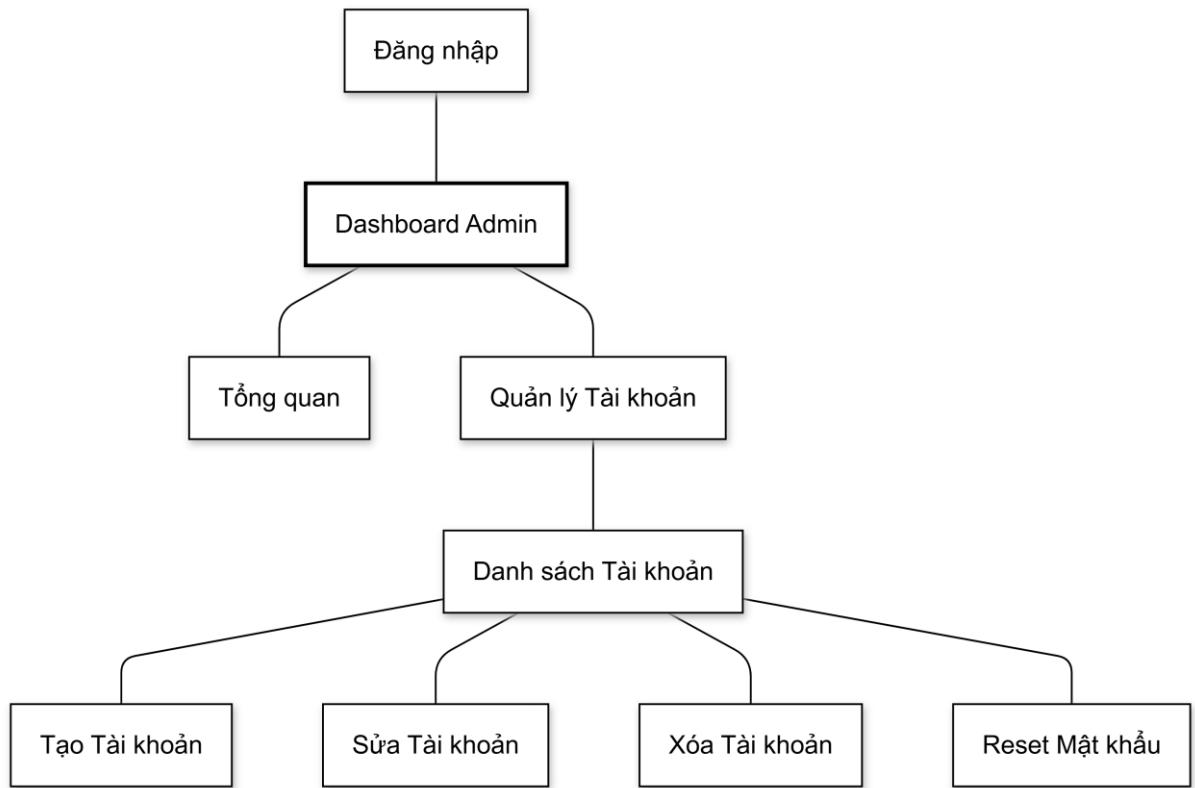


Hình 3.9 Sơ đồ tuần tự quá trình quản lý Chương trình đào tạo

### 3.2.5 Thiết kế giao diện

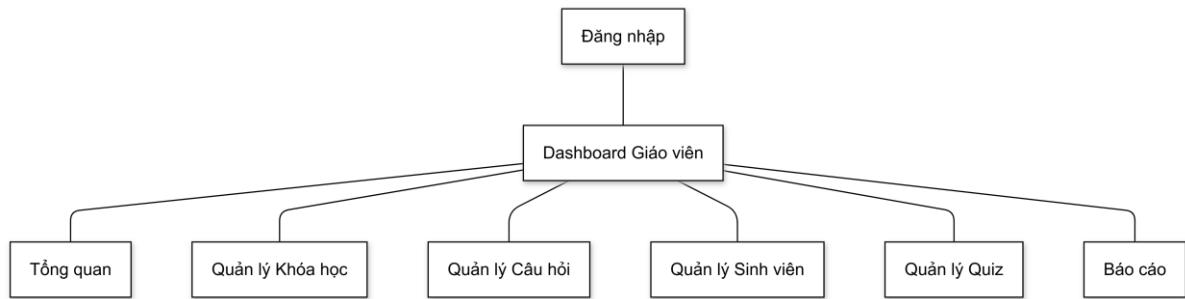
#### 3.2.5.1 Sơ đồ website

Sơ đồ website Admin phần Tổng quan và Quản lý Tài khoản mô tả cấu trúc trang web dành cho quản trị viên. Từ trang đăng nhập, Admin truy cập Dashboard để xem tổng quan hệ thống và quản lý Tài khoản.



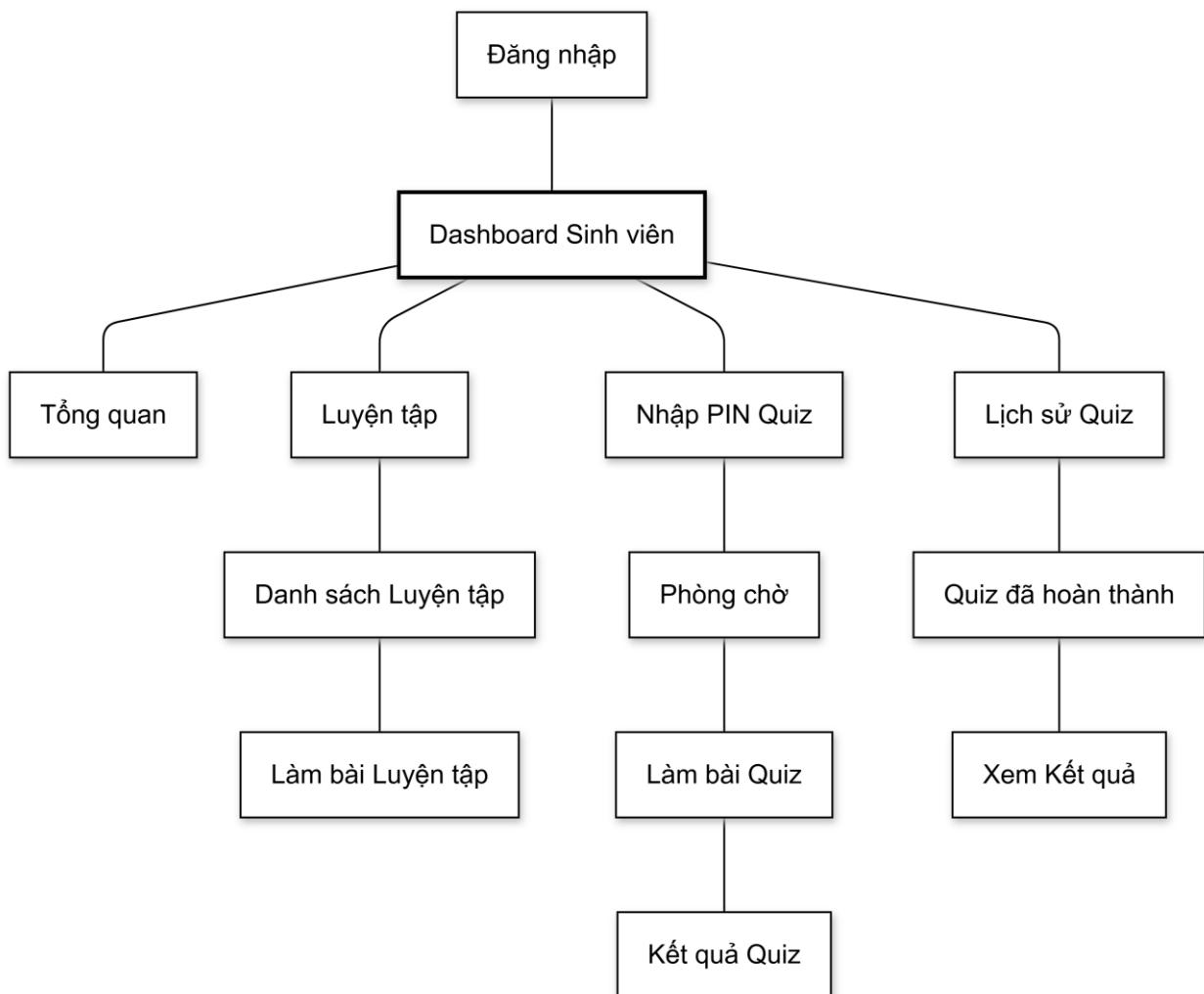
Hình 3.10 Sơ đồ website quyền admin phần Tổng quan và Quản lý Tài khoản

Sơ đồ website Giảng viên phần cấu trúc tổng quan mô tả các chức năng chính trên Dashboard. Từ trang đăng nhập, Giáo viên truy cập Dashboard để xem tổng quan và các menu chức năng: Quản lý Khóa học, Câu hỏi, Sinh viên, Quiz và Báo cáo.



Hình 3.11 Sơ đồ website quyền giảng viên phần cấu trúc tổng quan

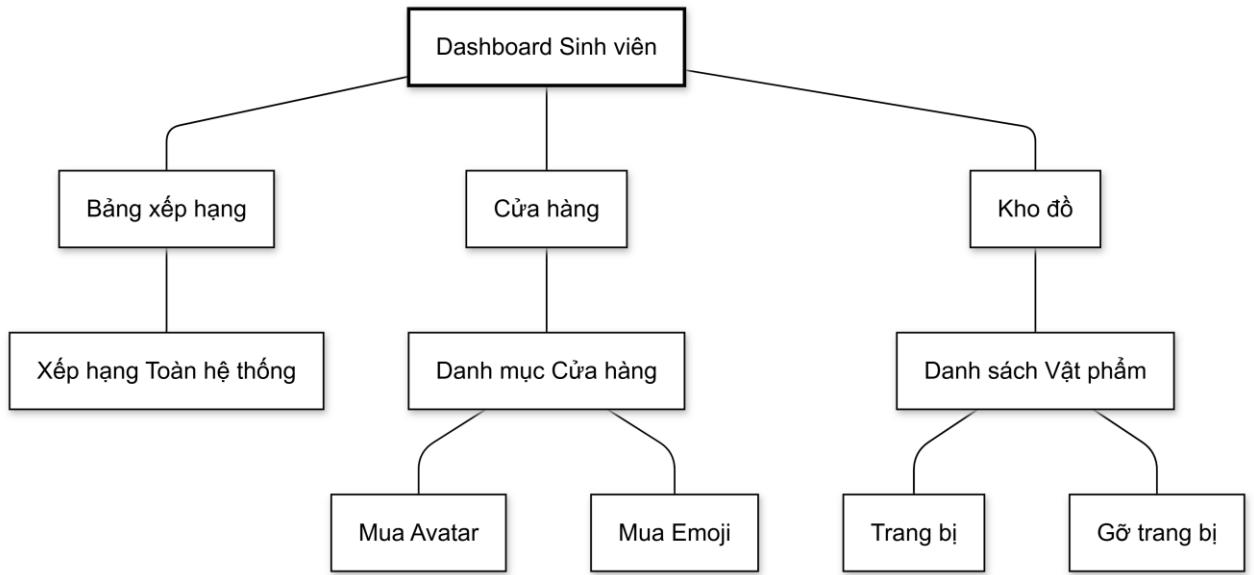
Sơ đồ website Sinh viên phần Hoạt động Học tập và Kiểm tra mô tả các chức năng học tập chính. Sinh viên có thể truy cập Danh sách luyện tập để ôn bài, nhập PIN để vào Phòng chờ làm Quiz và xem lại Lịch sử các bài Quiz đã hoàn thành.



Hình 3.12 Sơ đồ website quyền sinh viên phần Hoạt động Học tập và Kiểm tra

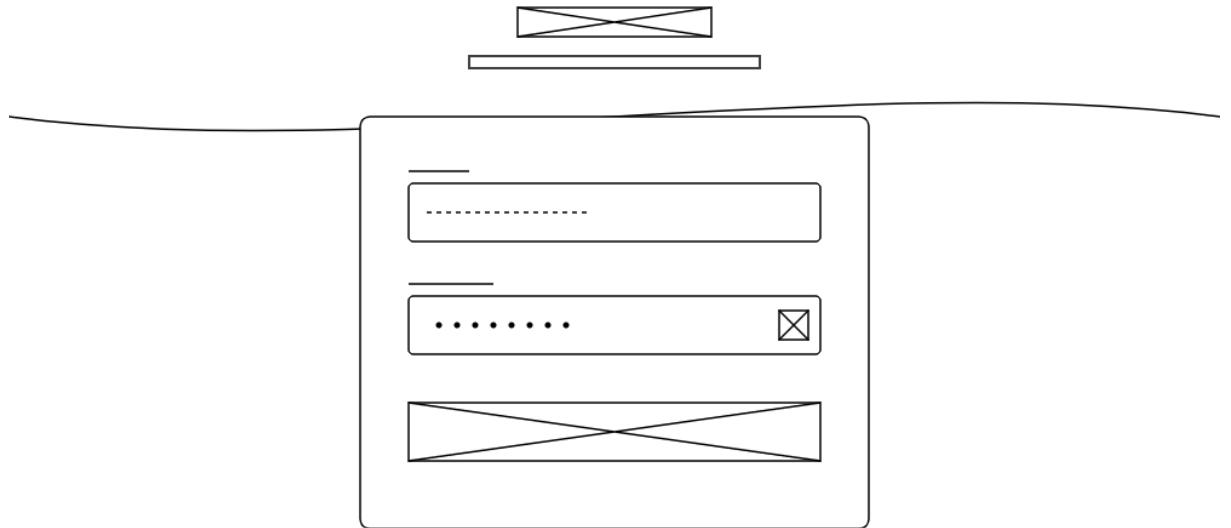
## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sơ đồ website Sinh viên phần Trò chơi hóa mô tả hệ thống trò chơi hóa. Sinh viên có thể xem Bảng xếp hạng toàn hệ thống, truy cập Cửa hàng để mua vật phẩm và vào Kho đồ để quản lý, trang bị các vật phẩm đã mua.



Hình 3.13 Sơ đồ website quyển sinh viên phần Trò chơi hóa

### 3.2.5.2 Phác thảo giao diện đăng nhập



Hình 3.14 Phác thảo giao diện đăng nhập

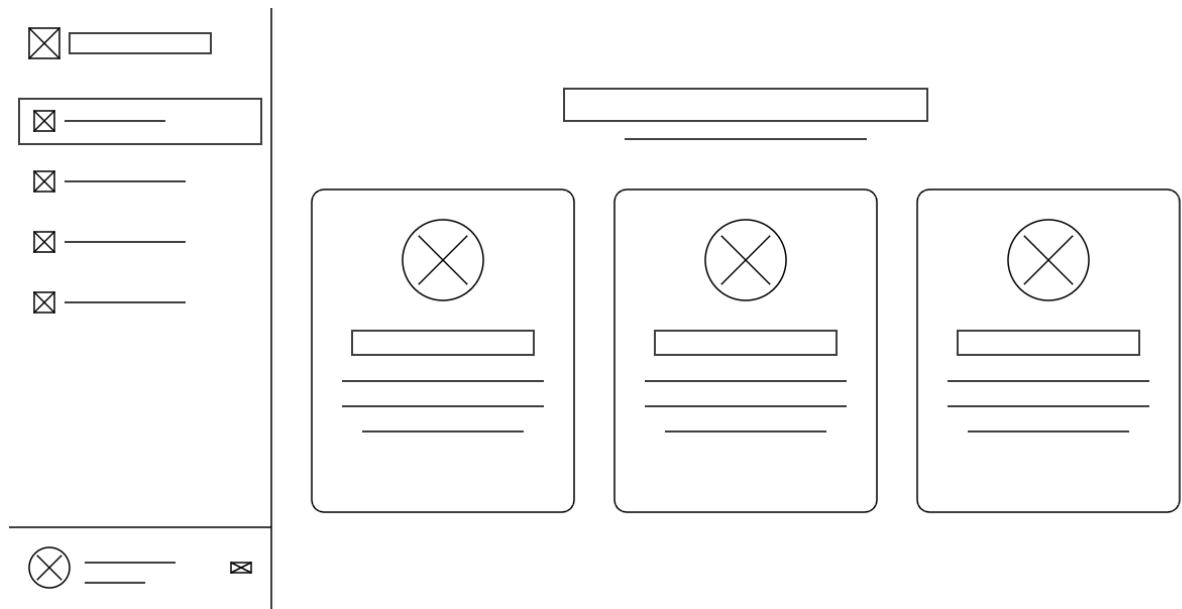
Bản phác thảo mô tả bố cục màn hình đăng nhập với thiết kế tối giản, tập trung vào khu vực nhập liệu trung tâm. Phần tiêu đề trang được đặt phía trên cùng để định hướng người dùng. Khung đăng nhập nằm chính giữa màn hình, bao gồm hai trường

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

nhập liệu cơ bản dành cho email và mật khẩu. Các nhãn thông tin được bố trí ngay trên mỗi ô nhập liệu để chủ thích rõ ràng. Phía cuối khung là nút bấm chức năng dạng khối chữ nhật, đại diện cho hành động xác nhận đăng nhập vào hệ thống.

### 3.2.5.3 Phác thảo giao diện trang tổng quan hệ thống



Hình 3.15 Phác thảo giao diện trang tổng quan hệ thống

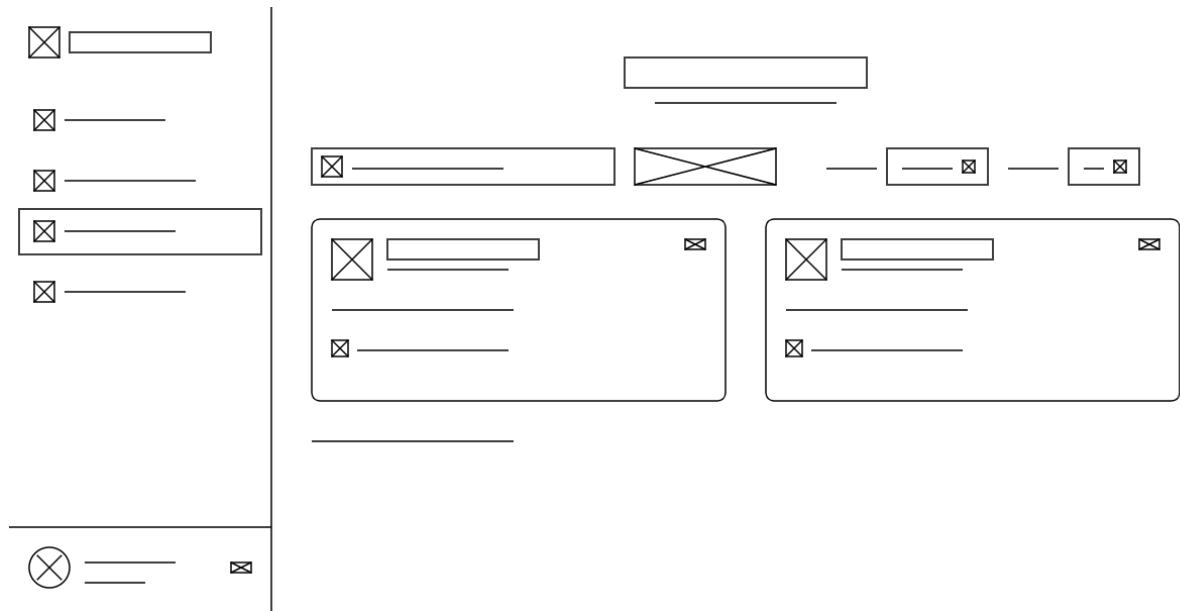
Bản phác thảo mô tả bố cục trang quản trị với thiết kế chia hai cột rõ ràng. Cột bên trái là thanh điều hướng cố định chứa logo thương hiệu, danh sách các chức năng quản lý và thông tin người dùng ở góc dưới. Khu vực nội dung chính bên phải hiển thị tiêu đề chào mừng ở vị trí trung tâm. Phía dưới tiêu đề là ba khung chức năng lớn được bố trí ngang hàng, đại diện cho các phân hệ quan trọng. Mỗi khung bao gồm biểu tượng minh họa, tên chức năng và các dòng mô tả ngắn gọn giúp người quản trị dễ dàng nắm bắt và thao tác.

### 3.2.5.4 Phác thảo giao diện quản lý danh sách khóa đào tạo

Bản phác thảo mô tả giao diện quản lý khóa đào tạo với bố cục hai phần chính. Thanh điều hướng bên trái cung cấp truy cập nhanh vào các chức năng quản trị hệ thống. Khu vực nội dung chính hiển thị tiêu đề trang lớn ở phía trên cùng. Ngay bên dưới là thanh công cụ bao gồm ô tìm kiếm, nút thêm mới khóa đào tạo và các tùy chọn sắp xếp, lọc dữ liệu. Phần danh sách các khóa đào tạo được trình bày dưới dạng lưới các thẻ

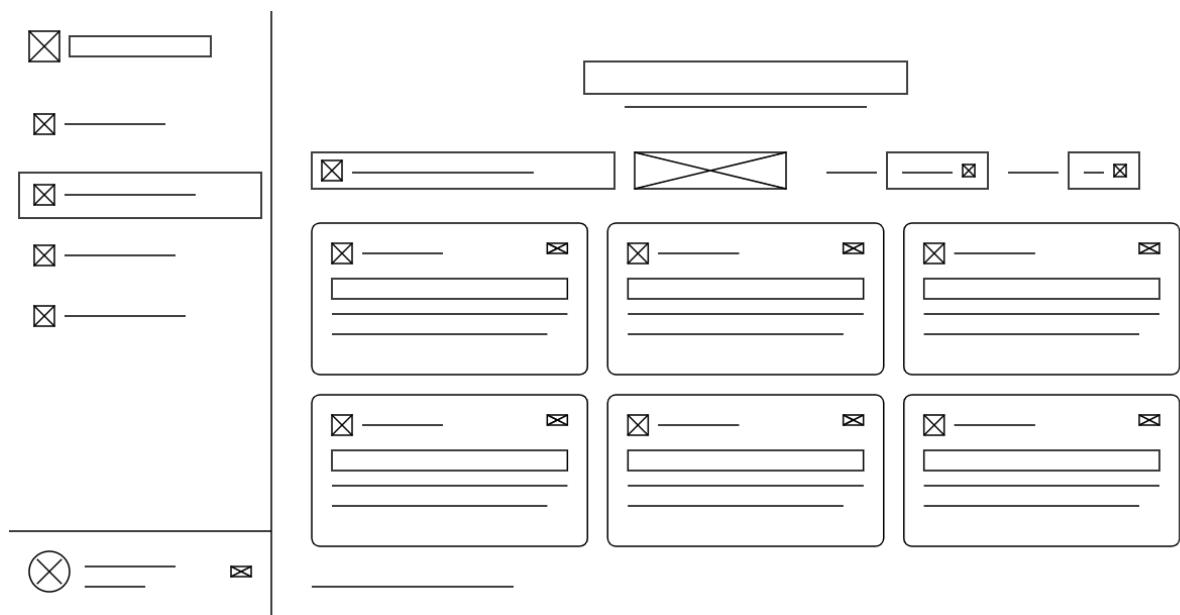
## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

thông tin. Mỗi thẻ hiển thị tóm tắt về tên khóa, chương trình đào tạo liên quan, niên khóa và biểu tượng trạng thái, giúp người quản trị dễ dàng theo dõi và thao tác.



Hình 3.16 Phác thảo giao diện quản lý danh sách khóa đào tạo

### 3.2.5.5 Phác thảo giao diện quản lý danh sách chương trình đào tạo



Hình 3.17 Phác thảo giao diện quản lý danh sách chương trình đào tạo

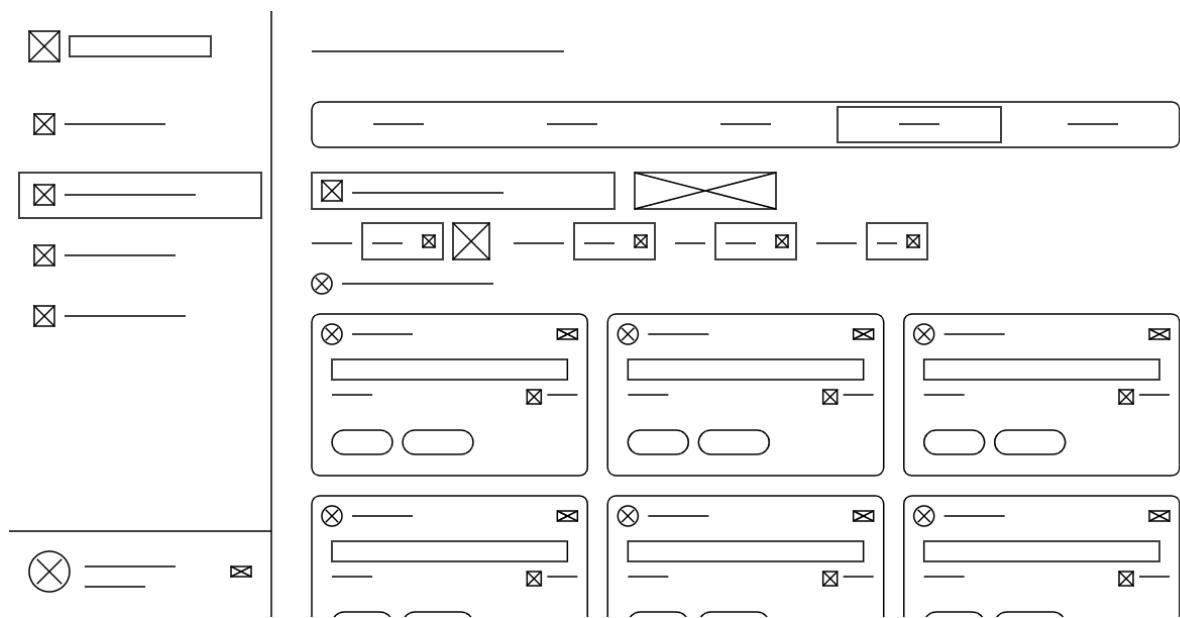
Bản phác thảo mô tả giao diện quản lý chương trình đào tạo với bố cục hai cột đặc trưng. Thanh điều hướng bên trái cung cấp truy cập nhanh vào các chức năng quản trị. Khu vực nội dung chính bên phải hiển thị danh sách các chương trình đào tạo dưới

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

---

dạng lưới thẻ trực quan. Mỗi thẻ chương trình bao gồm biểu tượng phân loại, tên chương trình, mô tả tóm tắt và nút chức năng mở rộng. Phía trên danh sách là thanh công cụ tích hợp các chức năng tìm kiếm, thêm mới chương trình và các tùy chọn sắp xếp, lọc dữ liệu giúp tối ưu hóa trải nghiệm quản lý.

### 3.2.5.6 Phác thảo giao diện quản lý danh sách môn học



Hình 3.18 Phác thảo giao diện quản lý danh sách môn học

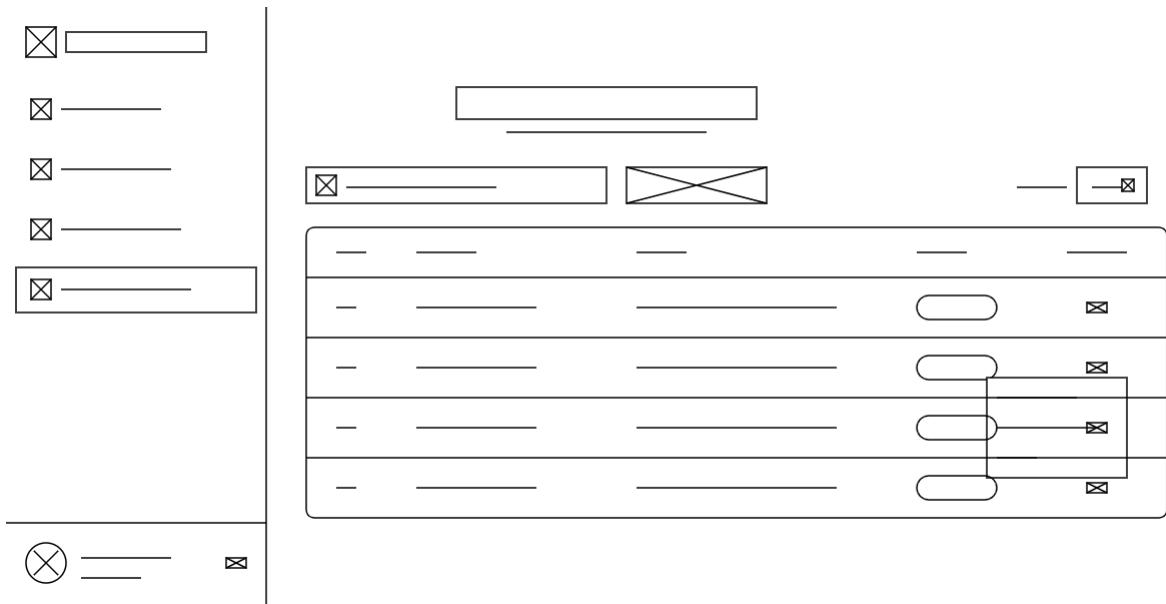
Bản phác thảo mô tả giao diện quản lý danh sách môn học trong một chương trình đào tạo. Thanh tab chức năng phía trên hiển thị tab "Môn học" đang được chọn. Khu vực công cụ bao gồm thanh tìm kiếm, nút tạo mới môn học và hàng loạt bộ lọc nâng cao như Sắp xếp, Trạng thái, Loại môn học và Số lượng hiển thị. Phần hiển thị chính được tổ chức dưới dạng lưới các thẻ môn học. Mỗi thẻ cung cấp thông tin tóm tắt gồm tên môn, số thứ tự, học kỳ, loại hình và trạng thái hoạt động, giúp người quản trị dễ dàng quan sát và quản lý tổng thể.

### 3.2.5.7 Phác thảo giao diện quản lý tài khoản người dùng

Bản phác thảo mô tả giao diện quản lý danh sách tài khoản người dùng trong hệ thống. Menu Quản lý tài khoản trên thanh bên trái được kích hoạt. Khu vực chính hiển thị tiêu đề trang lớn và bảng danh sách người dùng bao gồm: Mã số, Họ tên, Vai

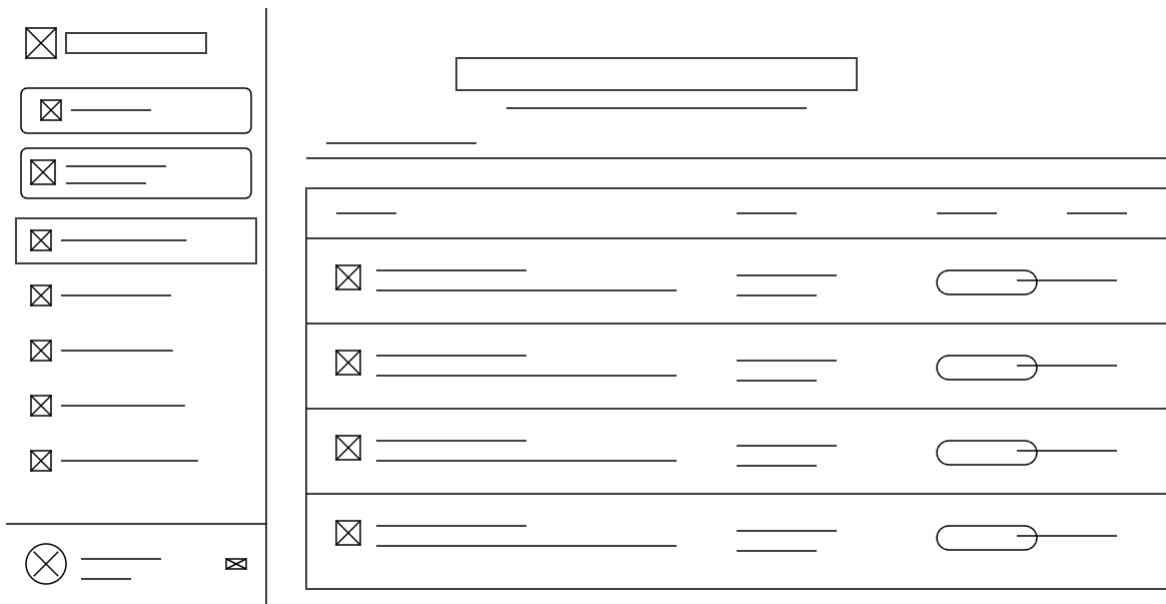
## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

trò. Mỗi dòng dữ liệu có menu thao tác nhanh cho phép chỉnh sửa thông tin, đặt lại mật khẩu hoặc xóa tài khoản.



Hình 3.19 Phác thảo giao diện quản lý tài khoản người dùng

### 3.2.5.8 Phác thảo giao diện môn học phân công

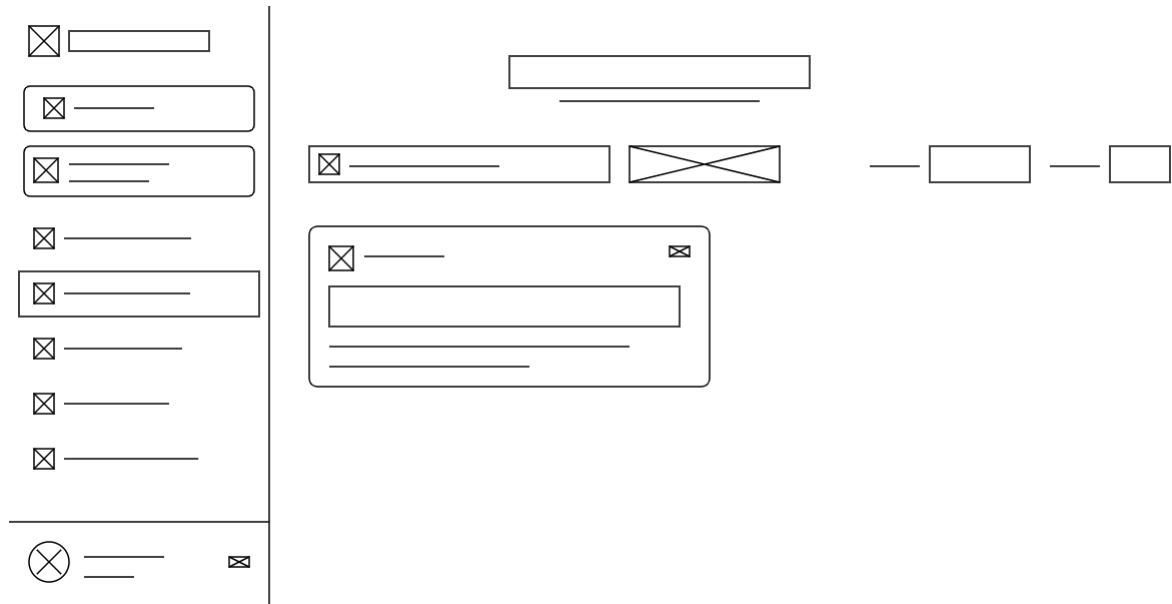


Hình 3.20 Phác thảo giao diện danh sách môn học được phân công

Bản phác thảo mô tả giao diện tổng quan dành cho giảng viên, hiển thị danh sách các môn học được phân công giảng dạy. Khu vực tiêu đề lớn chào mừng người dùng. Phía dưới là thanh tìm kiếm giúp lọc nhanh môn học theo tên hoặc học kỳ. Danh sách môn học được trình bày dạng danh sách chi tiết với các cột thông tin: Tên môn học và Nguyễn Huỳnh Phú Vinh

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi mô tả, Học kỳ áp dụng, Trạng thái hoạt động và Ghi chú. Các biểu tượng trạng thái giúp giảng viên dễ dàng nhận biết môn học nào đang diễn ra.

### 3.2.5.9 Phác thảo giao diện khóa học của giảng viên

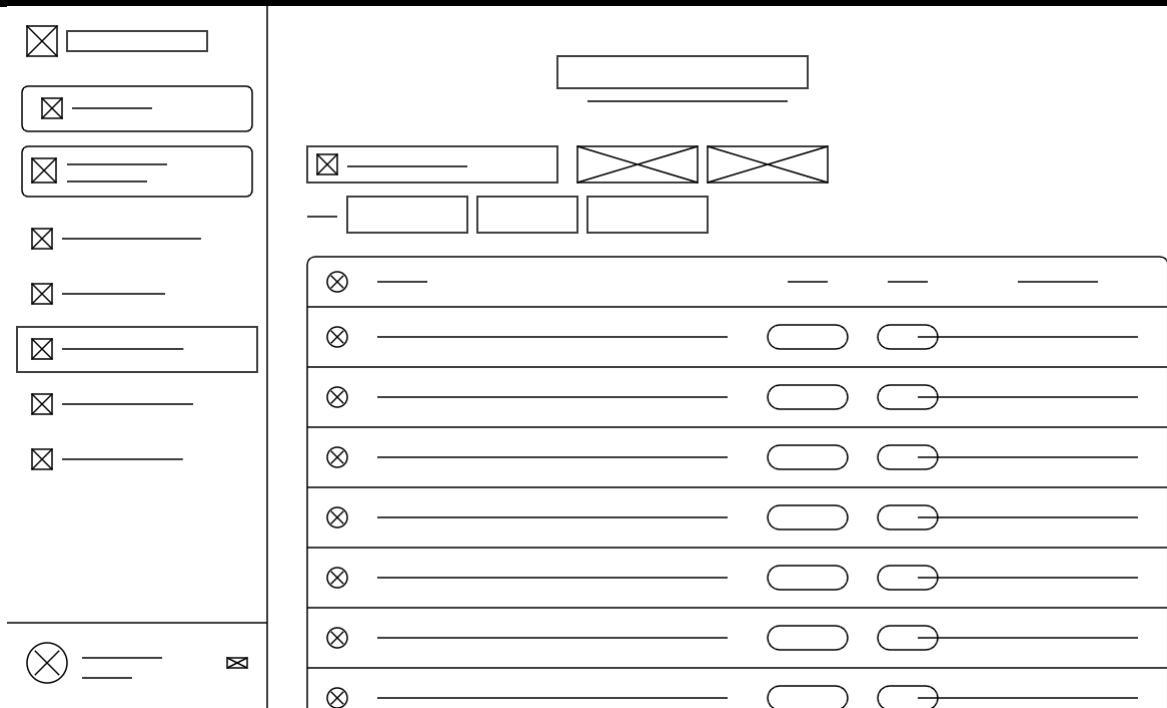


Hình 3.21 Phác thảo giao diện quản lý khóa học giảng dạy

Bản phác thảo mô tả giao diện quản lý các lớp học phần mà giảng viên đang phụ trách. Tab "Khóa học" trên thanh điều hướng bên trái được kích hoạt. Giao diện chính bao gồm tiêu đề lớn, thanh công cụ với ô tìm kiếm, nút "Thêm khóa học mới" và bộ lọc sắp xếp. Danh sách khóa học hiển thị dưới dạng thẻ chứa thông tin mã lớp, tên môn học và giảng viên phụ trách.

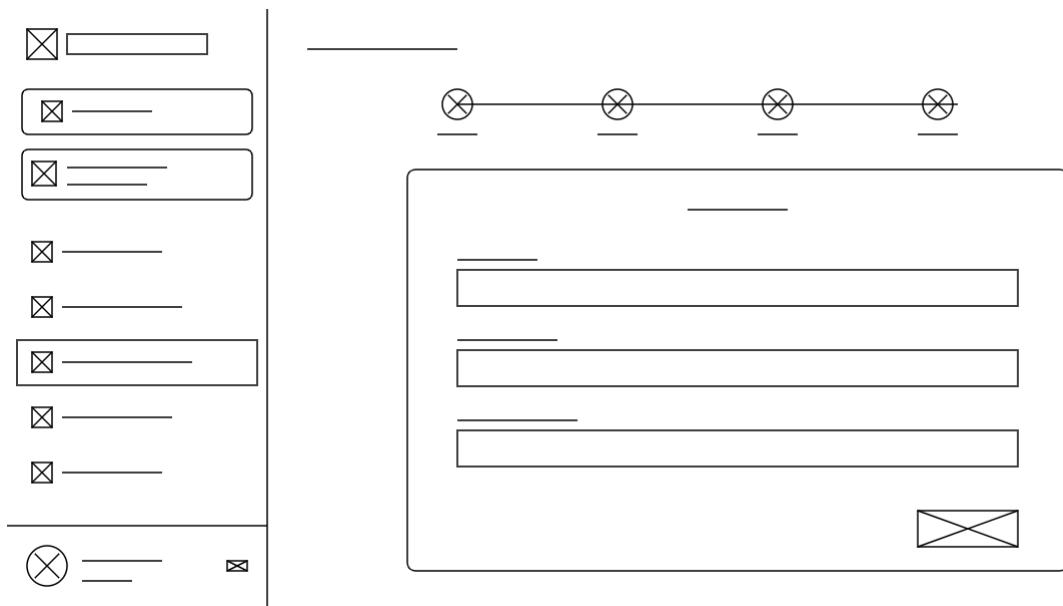
### 3.2.5.10 Phác thảo giao diện quản lý ngân hàng câu hỏi

Bản phác thảo mô tả giao diện quản lý ngân hàng câu hỏi trắc nghiệm. Menu "Quản lý câu hỏi" trên thanh bên được kích hoạt. Khu vực chính cung cấp các công cụ lọc mạnh mẽ theo loại câu hỏi, độ khó và chuẩn đầu ra. Nút "Thêm câu hỏi mới" và "Import câu hỏi" giúp giảng viên xây dựng ngân hàng đề thi nhanh chóng. Danh sách câu hỏi hiển thị chi tiết nội dung, loại hình, nhãn độ khó và chuẩn đầu ra tương ứng.



Hình 3.22 Phác thảo giao diện danh sách ngân hàng câu hỏi

### **3.2.5.11 Phác thảo giao diện tạo bài kiểm tra**



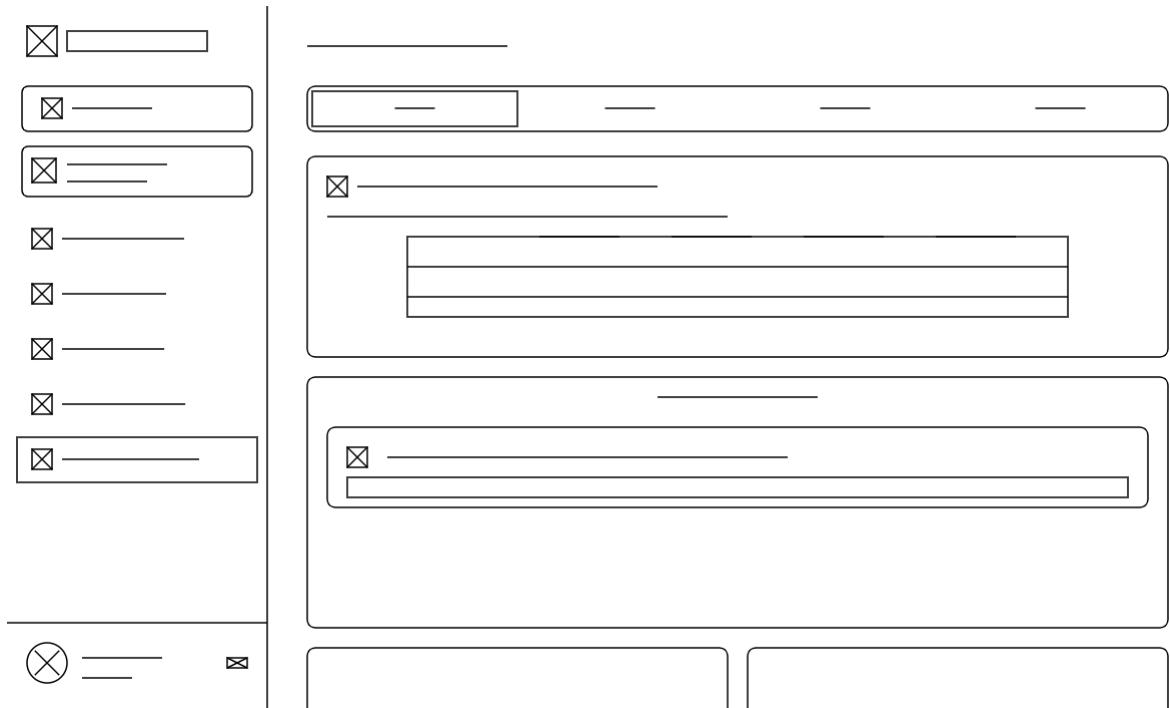
Hình 3.23 Phác thảo giao diện thiết lập thông tin cơ bản bài kiểm tra

Bản phác thảo mô tả bước đầu tiên trong quy trình tạo bài kiểm tra. Thành phần trình phía trên hiển thị 4 bước thực hiện, với bước "Thông tin cơ bản" đang được kích hoạt. Tại đây, giảng viên cần chọn khóa học áp dụng, đặt tên cho bài kiểm tra và thiết

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

lập thời gian làm bài. Giao diện đơn giản giúp người dùng tập trung vào các thông tin thiết yếu nhất để khởi tạo bài thi.

### 3.2.5.12 Phác thảo giao diện báo cáo ma trận kiến thức

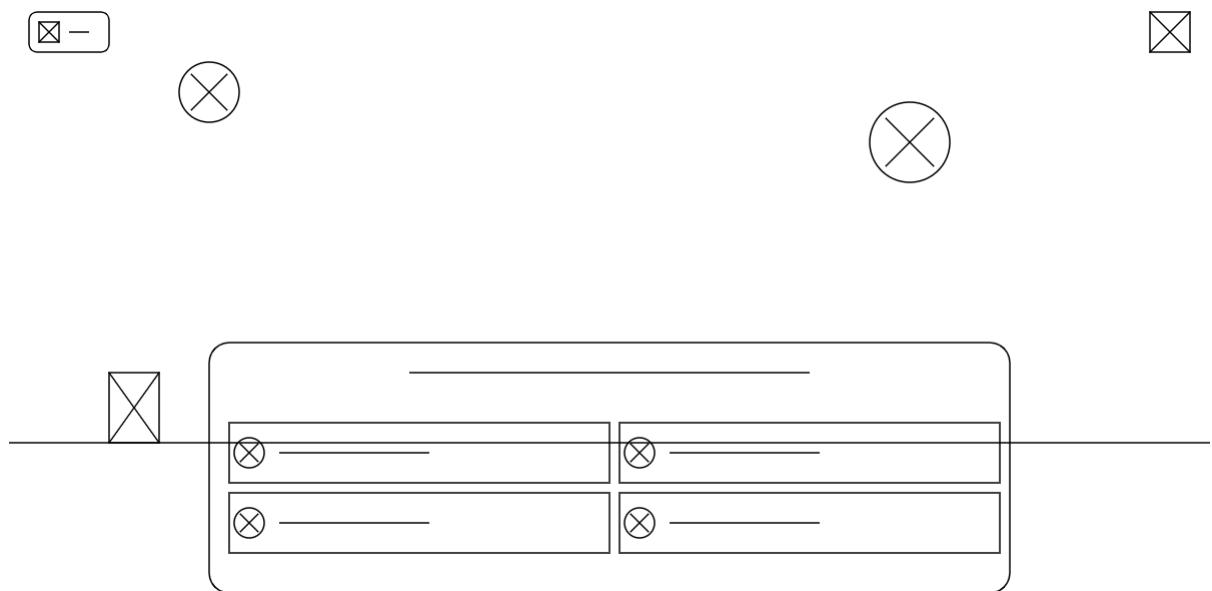


Hình 3.24 Phác thảo giao diện báo cáo phân bổ kiến thức bài kiểm tra

Bản phác thảo mô tả giao diện báo cáo phân tích cấu trúc đề thi. Tab "Ma trận" được chọn. Biểu đồ nhiệt thể hiện sự phân bố số lượng câu hỏi dựa trên hai trục: Độ khó và Chuẩn đầu ra. Phía dưới là danh sách chi tiết các câu hỏi trong đề thi. Ngoài ra, giao diện còn cung cấp các biểu đồ thanh ngang thống kê tỷ lệ cân bằng độ khó và phân bố chuẩn đầu ra, giúp giảng viên đánh giá chất lượng đề thi.

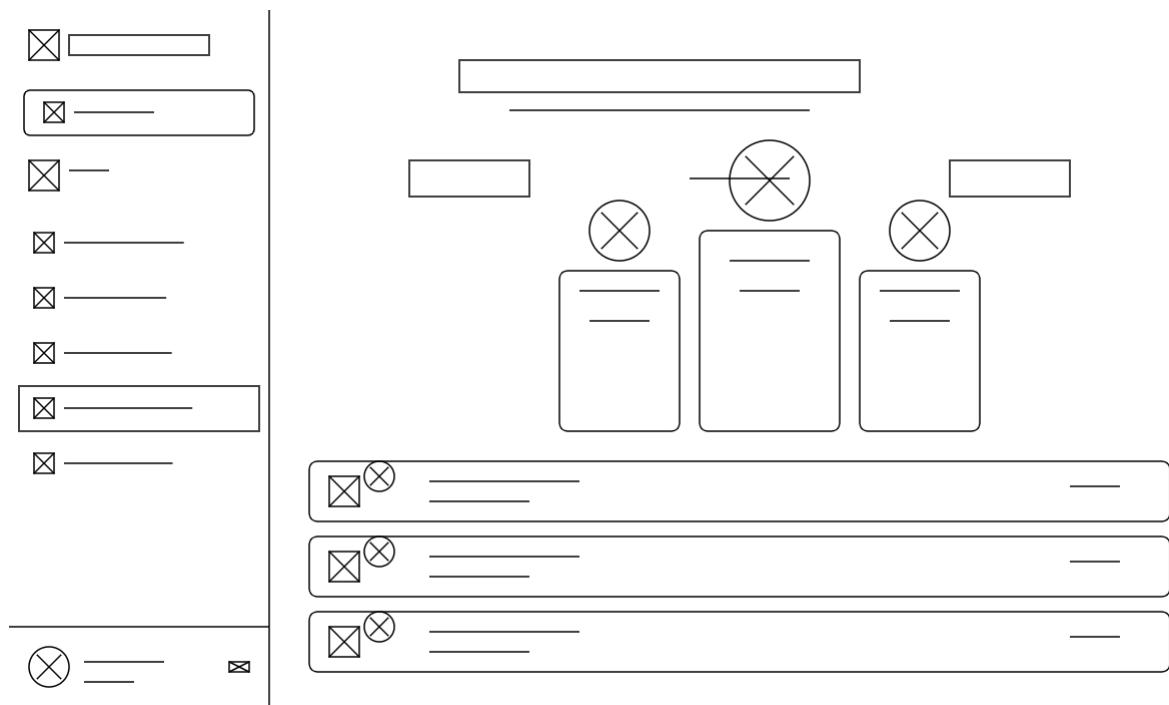
### 3.2.5.13 Phác thảo giao diện bài kiểm tra theo phong cách trò chơi

Bản phác thảo mô tả giao diện làm bài kiểm tra được thiết kế theo phong cách trò chơi. Màn hình mô phỏng bối cảnh nhân vật di chuyển vượt chướng ngại vật. Câu hỏi trắc nghiệm hiển thị trong khung nổi bật ở giữa màn hình cùng 4 phương án lựa chọn. Các yếu tố như điểm số, nút gợi ý và điều khiển âm thanh được bố trí ở các góc, tạo trải nghiệm học tập thú vị và tương tác cao.



Hình 3.25 Phác thảo giao diện làm bài kiểm tra dạng game

### 3.2.5.14 Phác thảo giao diện bảng xếp hạng thành tích sinh viên



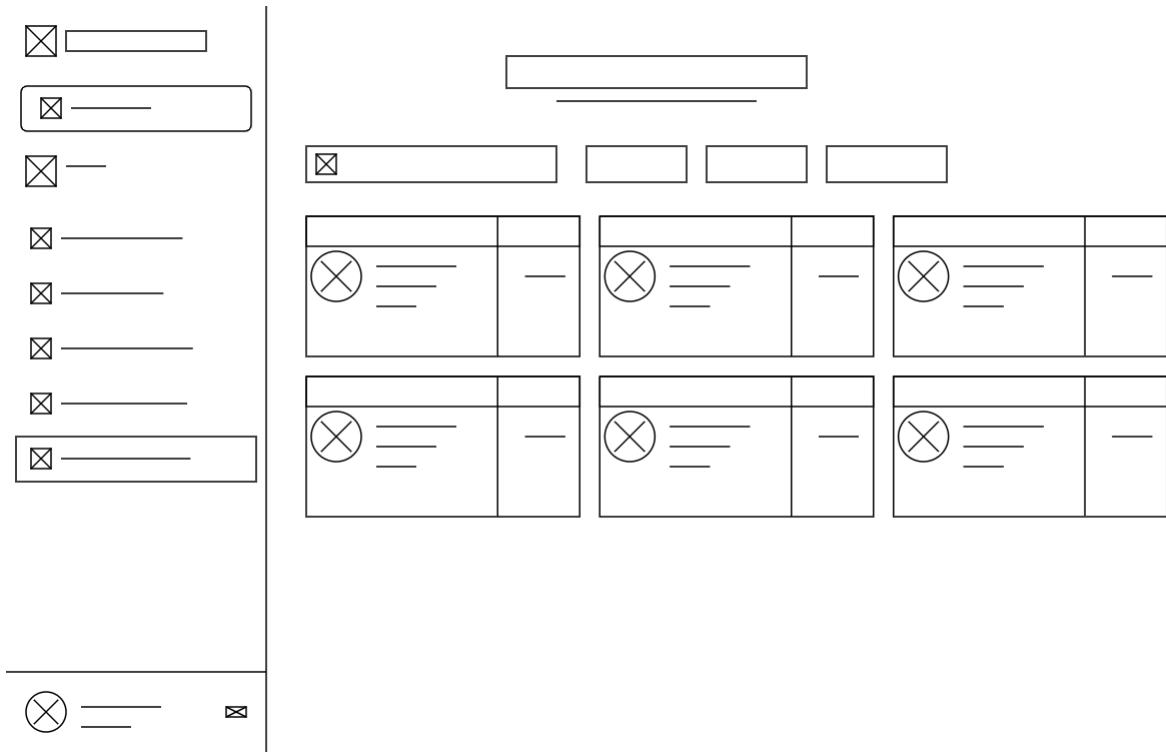
Hình 3.26 Phác thảo giao diện bảng xếp hạng thi đua

Bản phác thảo mô tả giao diện Bảng xếp hạng thành tích học tập. Menu "Bảng xếp hạng" trên thanh điều hướng được chọn. Phần đầu trang vinh danh Top 3 sinh viên xuất sắc nhất bằng các bức ảnh trực quan với hình đại diện và tổng điểm nổi bật. Phía

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

dưới là danh sách xếp hạng chi tiết của các sinh viên còn lại. Bộ lọc thời gian giúp người dùng theo dõi sự cạnh tranh trong các khoảng thời gian khác nhau.

### 3.2.5.15 Phác thảo giao diện cửa hàng vật phẩm



Hình 3.27 Phác thảo giao diện cửa hàng đổi quà

Bản phác thảo mô tả giao diện Cửa hàng nơi sinh viên sử dụng điểm tích lũy để đổi vật phẩm. Menu "Cửa hàng" được kích hoạt. Khu vực tiêu đề chào mừng lớn ở trên cùng. Thanh công cụ bao gồm ô tìm kiếm và các bộ lọc theo Loại vật phẩm, Độ hiếm và Khoảng giá. Danh sách vật phẩm được trình bày dưới dạng lưới các thẻ, hiển thị hình ảnh, tên, độ hiếm và giá bán. Hiệu ứng "Kéo để mua" được mô phỏng trên từng thẻ.

## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1. Kết quả triển khai hệ thống

Sau quá trình phân tích, thiết kế và hiện thực, hệ thống nền tảng học trực tuyến có tích hợp yếu tố trò chơi hóa đã được triển khai thành công dưới dạng một ứng dụng web. Hệ thống cho phép người dùng truy cập thông qua trình duyệt và hỗ trợ đầy đủ các vai trò chính, bao gồm quản trị viên, giảng viên và sinh viên. Các chức năng cốt lõi được xác định trong giai đoạn phân tích yêu cầu đã được hiện thực và vận hành ổn định trong môi trường triển khai thử nghiệm.

Về phạm vi triển khai, hệ thống đáp ứng đầy đủ các nghiệp vụ cơ bản của một nền tảng học trực tuyến, đồng thời tích hợp các yếu tố trò chơi nhằm nâng cao mức độ tương tác của người học. Đối với vai trò quản trị viên, hệ thống cho phép quản lý tài khoản người dùng, quản lý các khóa đào tạo và chương trình đào tạo, cũng như thực hiện phân quyền truy cập phù hợp với từng đối tượng sử dụng. Các chức năng này giúp đảm bảo tính nhất quán và khả năng kiểm soát trong quá trình vận hành hệ thống.

Đối với vai trò giảng viên, hệ thống hỗ trợ quản lý các khóa học giảng dạy, xây dựng và quản lý ngân hàng câu hỏi, tạo và cấu hình các bài kiểm tra (quiz), cũng như theo dõi và xem báo cáo kết quả học tập của sinh viên. Các chức năng này được triển khai thông qua giao diện trực quan, giúp giảng viên thao tác thuận tiện trong quá trình tổ chức giảng dạy và đánh giá người học.

Đối với sinh viên, hệ thống cho phép tham gia các khóa học, thực hiện các bài kiểm tra trực tuyến và theo dõi kết quả học tập cá nhân. Đặc biệt, các yếu tố trò chơi như làm bài kiểm tra dưới dạng trò chơi, tích lũy điểm số, bảng xếp hạng và phần thưởng đã được tích hợp trực tiếp vào giao diện người dùng. Các yếu tố này góp phần tạo động lực học tập và nâng cao mức độ hứng thú của sinh viên khi tham gia hệ thống.

Về mặt giao diện frontend, hệ thống được triển khai theo hướng trực quan, nhất quán và thân thiện với người dùng. Các thành phần giao diện hiển thị đúng chức năng, phản hồi kịp thời các thao tác của người dùng và hỗ trợ điều hướng rõ ràng giữa các

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi màn hình. Giao diện hoạt động ổn định trên các trình duyệt phổ biến, đáp ứng tốt yêu cầu sử dụng trong môi trường học tập trực tuyến.

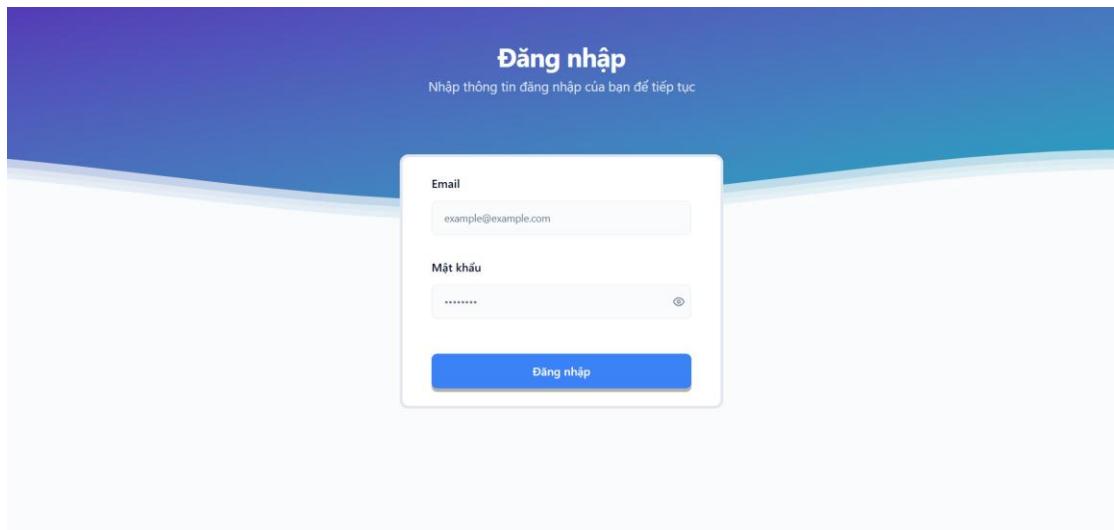
Bên cạnh đó, quá trình tích hợp giữa frontend và backend được thực hiện thông qua các API, đảm bảo việc trao đổi dữ liệu và xử lý nghiệp vụ diễn ra chính xác và đồng bộ. Frontend có khả năng gửi yêu cầu, tiếp nhận dữ liệu phản hồi và cập nhật giao diện theo thời gian thực, góp phần nâng cao trải nghiệm người dùng khi sử dụng hệ thống.

Nhìn chung, kết quả triển khai cho thấy hệ thống đã đáp ứng được các mục tiêu đề ra ban đầu, bao gồm việc xây dựng một nền tảng học trực tuyến có giao diện thân thiện, hỗ trợ đầy đủ các chức năng học tập và giảng dạy, đồng thời tích hợp hiệu quả các yếu tố trò chơi hóa nhằm nâng cao trải nghiệm và mức độ tương tác của người học.

## 4.2. Kết quả các kịch bản triển khai

### 4.2.1 Đăng nhập và phân quyền

Quy trình bắt đầu với việc xác thực người dùng. **Hình 4.1** minh họa giao diện đăng nhập được thiết kế đơn giản để người dùng dễ dàng truy cập. Tại đây, người dùng sẽ nhập địa chỉ email và mật khẩu vào các ô tương ứng. Hệ thống có hỗ trợ tính năng hiển thị mật khẩu để người dùng kiểm tra lại độ chính xác của các ký tự đã nhập trước khi nhấn nút "Đăng nhập".



Hình 4.1 Giao diện đăng nhập

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sau khi đăng nhập thành công, hệ thống sẽ điều hướng người dùng dựa trên vai trò. Với quyền **Quản trị viên**, giao diện tổng quan hiển thị tại **Hình 4.2**. Trang này liệt kê các nhóm chức năng chính bao gồm quản lý chương trình đào tạo, quản lý khóa đào tạo và quản lý tài khoản người dùng dưới dạng các thẻ lớn, giúp quản trị viên dễ dàng lựa chọn tác vụ.

The screenshot shows the Synlearnia system's general management interface. On the left, there is a sidebar with the Synlearnia logo and navigation links: 'Tổng quan' (General), 'Chương trình ĐT' (Program DT), 'Khóa đào tạo' (Training Course), and 'Quản lý tài khoản' (Account Management). Below these are user roles: 'admin Quản trị viên' and 'teacher Giảng viên'. On the right, the main area is titled 'Quản trị hệ thống' (System Management) with the sub-instruction 'Chọn một mục để bắt đầu quản lý' (Select a category to start managing). It features three large cards: 'Chương trình Đào tạo' (Training Program) with a graduation cap icon, 'Khóa Đào tạo' (Training Course) with a briefcase icon, and 'Quản lý Tài khoản' (Account Management) with a user icon. Each card has a brief description below it.

Hình 4.2 Giao diện tổng quan dành cho quản trị viên

The screenshot shows the Synlearnia system's course distribution management interface. On the left, there is a sidebar with the Synlearnia logo and navigation links: 'Tạo Quiz' (Create Quiz), 'Thiết kế web Học kỳ II - Năm 1 ...', 'Tổng quan' (General), 'Khóa học', 'Quản lý câu hỏi', 'Quản lý Sinh viên', and 'Quản lý Quiz'. Below these are user roles: 'teacher Giảng viên' and '...'. On the right, the main area is titled 'Môn học phân công' (Subject Allocation) with the sub-instruction 'Quản lý và theo dõi các môn học được phân công của bạn' (Manage and track the subjects assigned to you). It features a search bar 'Q. Tìm môn học, học kỳ...' and a table listing subjects. The table columns are 'Môn học', 'Học kỳ', 'Hoạt động', and 'Ghi chú'. There are four rows of data:

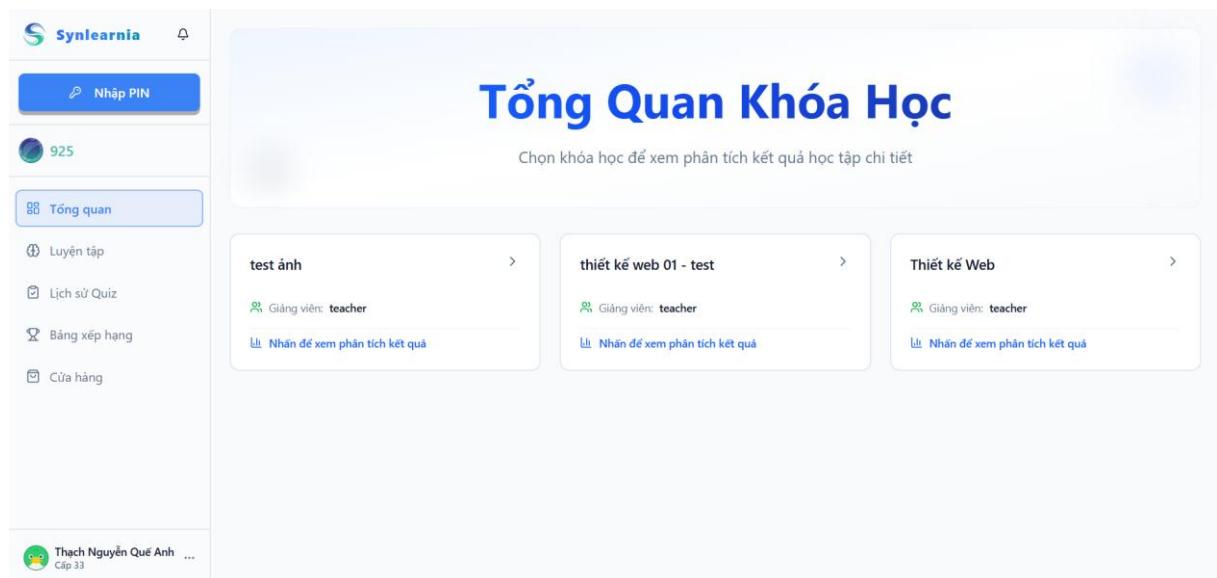
Môn học	Học kỳ	Hoạt động	Ghi chú
Kỹ thuật lập trình	Học kỳ II - Năm 1 Năm học 2025-2026	Đang hoạt động	Không có ghi chú
Thiết kế web	Học kỳ II - Năm 1 Năm học 2025-2026	Đang hoạt động	Không có ghi chú
Kỹ thuật lập trình	Học kỳ I - Năm 1 Năm học 2025-2026	Không hoạt động	Không có ghi chú
Thiết kế web	Học kỳ I - Năm 1 Năm học 2025-2026	Không hoạt động	Không có ghi chú

Hình 4.3 Giao diện tổng quan môn học phân công dành cho giảng viên

Với quyền **Giảng viên**, hệ thống hiển thị danh sách các lớp học phần được phân công như **Hình 4.3**. Giao diện cung cấp thông tin chi tiết về tên môn học, học kỳ và Nguyễn Huỳnh Phú Vinh

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi trạng thái hoạt động của lớp. Giảng viên có thể dùng thanh tìm kiếm để lọc nhanh môn học hoặc nhấn vào từng lớp để truy cập các chức năng quản lý chuyên sâu.

Với quyền **Sinh viên**, **Hình 4.4** hiển thị danh sách các khóa học đang tham gia. Mỗi thẻ khóa học cung cấp thông tin về tên môn, giảng viên phụ trách và đường dẫn nhanh để xem báo cáo phân tích kết quả học tập chi tiết.



Hình 4.4 Giao diện tổng quan khóa học dành cho sinh viên

#### 4.2.2 Giảng viên tạo bài kiểm tra

Để tạo bài kiểm tra, trước hết giảng viên cần chuẩn bị ngân hàng câu hỏi tại giao diện **Hình 4.5**. Trang này cho phép xây dựng ngân hàng đề thi phong phú với các bộ lọc nâng cao theo loại câu, độ khó và chuẩn đầu ra. Giảng viên có thể thêm thủ công hoặc nhập hàng loạt từ file Excel/ZIP thông qua tính năng Import.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

The screenshot shows the Synlearnia platform's question management interface. On the left, there is a sidebar with navigation links: 'Tạo Quiz' (Create Quiz), 'Thiết kế web' (Web Design), 'Tổng quan' (Overview), 'Khóa học' (Course), 'Quản lý câu hỏi' (Manage Questions) (which is highlighted in blue), 'Quản lý Sinh viên' (Manage Students), and 'Quản lý Quiz'. Below this is a user profile for 'teacher Giảng viên'. The main area is titled 'Quản lý câu hỏi' (Manage Questions) and contains a search bar with 'Tim kiếm câu hỏi...' and buttons for '+ Thêm câu hỏi mới' (Add new question) and 'Import câu hỏi' (Import question). There are also filtering options: 'Lọc: Tất cả loại câu hỏi', 'Tất cả độ khó', and 'Tất cả chuẩn đầu ra'. A table lists 10 questions, each with a checkbox, title, category, difficulty level, and description. The columns are 'Câu hỏi' (Question), 'Loại' (Type), 'Độ khó' (Difficulty), and 'Chuẩn đầu ra' (Learning outcome). At the bottom, it says 'Hiển thị 1 - 10 trong tổng số 395 kết quả (Trang 1 / 40)' and has a page navigation bar with buttons for 'Trước' (Previous), 'Sau' (Next), and numbers 1, 2, 3, 4, 5, ..., 40.

Hình 4.5 Giao diện quản lý ngân hàng câu hỏi

Khi thêm mới một câu hỏi, giao diện nhập liệu hiện ra như **Hình 4.6**. Biểu mẫu chi tiết cho phép nhập nội dung đề bài, chọn loại câu hỏi, mức độ khó và gắn thẻ chuẩn đầu ra. Hệ thống hỗ trợ cấu hình phương án trả lời, đánh dấu đáp án đúng và đính kèm tệp đa phương tiện minh họa.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

The screenshot shows the Synlearnia platform's interface for creating a new question. On the left sidebar, there are navigation links: 'Tổng quan', 'Khóa học', 'Quản lý câu hỏi' (highlighted in blue), 'Quản lý Sinh viên', 'Quản lý Quiz', and 'teacher Giảng viên'. The main content area is titled 'Câu hỏi > Tạo mới'. It contains several input fields and dropdown menus:

- Thông tin câu hỏi:** 'Nhập thông tin cơ bản cho câu hỏi mới'.
- Nội dung câu hỏi \***: A large text area for entering the question content.
- Loại câu hỏi \***: A dropdown menu set to 'Trắc nghiệm'.
- Độ khó \***: A dropdown menu set to 'Đèn'.
- Chuẩn đầu ra \***: A dropdown menu set to 'LO1 - Xác định nguyên tắc hoạt động của Xác định nguyên tắc hoạt động của We...'.  
A note below says: 'Thiết lập các câu trả lời cho câu hỏi. Phải có đúng 1 câu trả lời đúng.'
- Giải thích**: A text area for explaining the question.
- Media cho câu hỏi (Tùy chọn - Tối đa 1 file)**: A file upload button labeled 'Tải lên'.

Below this is a table for defining answers:

STT	Nội dung câu trả lời	Đáp án đúng	Media	Thao tác
1	Nhập nội dung câu trả lời...	<input checked="" type="radio"/>	<input type="button" value=""/>	<input type="button" value=""/>
2	Nhập nội dung câu trả lời...	<input type="radio"/>	<input type="button" value=""/>	<input type="button" value=""/>

A 'Thêm câu trả lời' button is located at the bottom of the table.

At the bottom right are 'Hủy bỏ' and 'Tạo câu hỏi' buttons.

Hình 4.6 Giao diện tạo mới câu hỏi

The screenshot shows the 'Create New Exam' process. On the left sidebar, there are navigation links: 'Tổng quan', 'Khóa học', 'Quản lý câu hỏi' (highlighted in blue), 'Quản lý Sinh viên', 'Quản lý Quiz', and 'teacher Giảng viên'. The main content area is titled 'Quản lý Quiz > Tạo mới'. It shows a four-step process:

1. Thông tin cơ bản (Tên và thời gian)
2. Chế độ Quiz (Đánh giá hoặc Luyện tập)
3. Mục tiêu học tập (Chọn nội dung)
4. Tiêu chí câu hỏi (Số lượng và độ khó)

The first step, 'Thông tin cơ bản', is expanded. It contains:

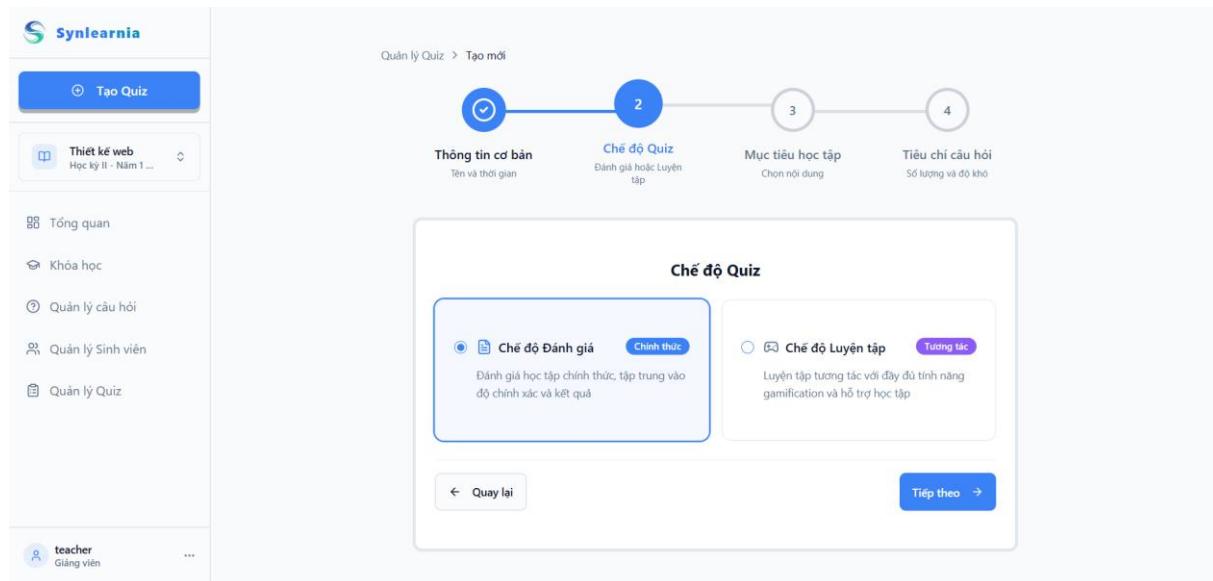
- Khóa học:** A dropdown menu showing 'Chưa có khóa học nào'.
- Tên bài kiểm tra:** An input field with placeholder 'Nhập tên bài kiểm tra'.
- Thời gian làm bài (phút):** An input field with placeholder '30'.
- Tiếp theo →** A blue button at the bottom right of the step panel.

Hình 4.7 Giao diện tạo mới bài kiểm tra - Bước 1

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sau khi có dữ liệu, giảng viên tiến hành tạo bài kiểm tra theo quy trình 4 bước: Tại **Bước 1 (Hình 4.7)**, giao diện hiển thị thanh tiến trình phía trên. Giảng viên chọn khóa học áp dụng, đặt tên bài kiểm tra và thiết lập thời gian làm bài tại bước "Thông tin cơ bản".

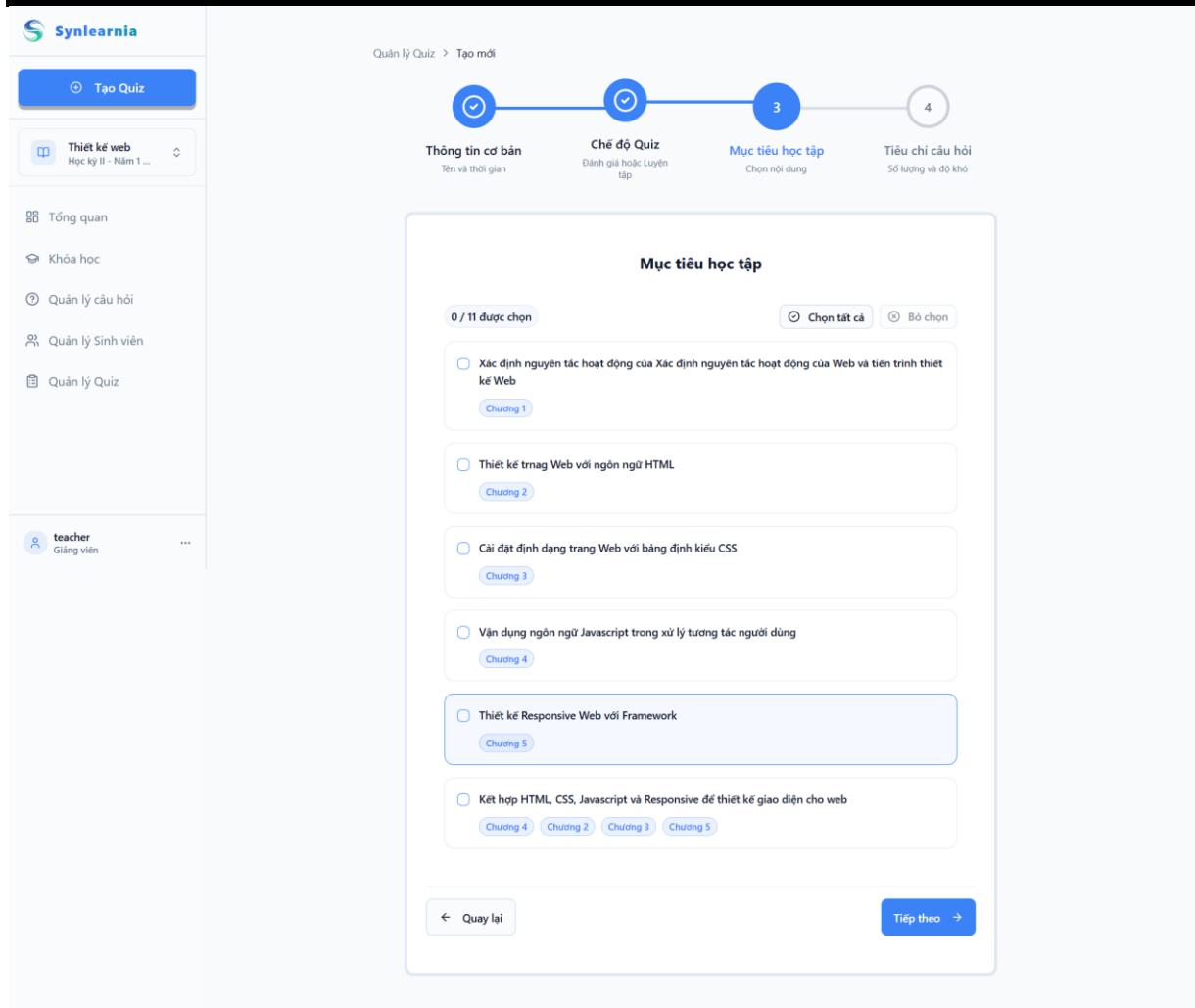
Tại **Bước 2 (Hình 4.8)**, giảng viên chọn "Chế độ Quiz". Hệ thống cung cấp hai tùy chọn: "Chế độ Đánh giá" cho bài thi chính thức và "Chế độ Luyện tập" tích hợp yếu tố trò chơi hóa để tăng tương tác.



Hình 4.8 Giao diện tạo mới bài kiểm tra - Bước 2

Tại **Bước 3 (Hình 4.9)**, giảng viên xác định phạm vi kiến thức bằng cách chọn "Mục tiêu học tập". Danh sách các chuẩn đầu ra được hiển thị kèm thông tin chương học liên quan, hỗ trợ tính năng chọn tất cả hoặc chọn lẻ từng mục.

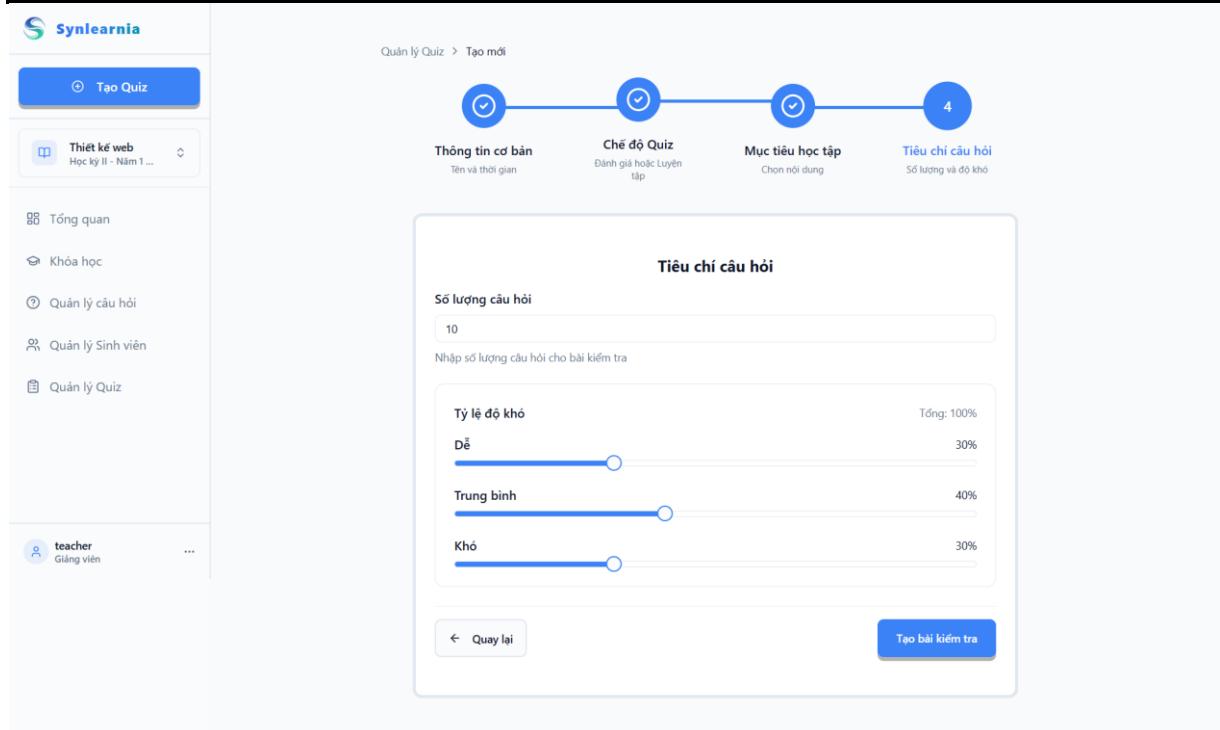
## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi



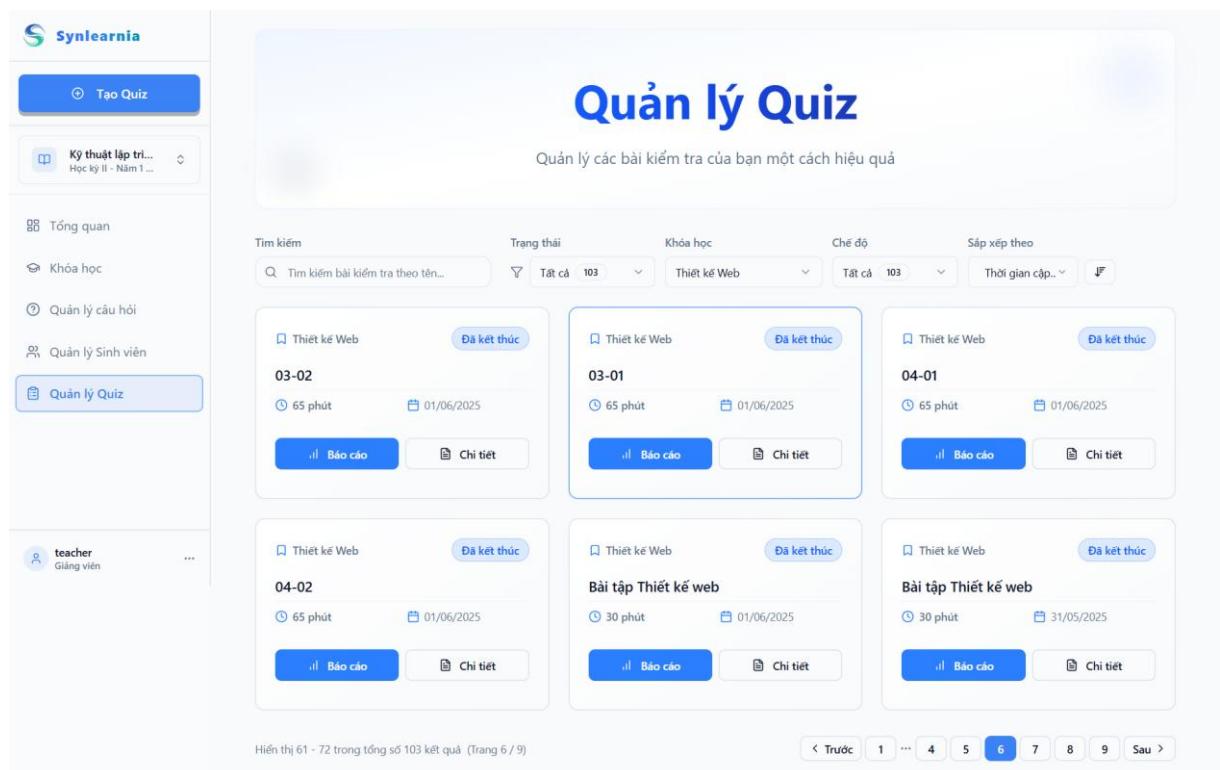
Hình 4.9 Giao diện tạo mới bài kiểm tra - Bước 3

Tại **Bước 4 (Hình 4.10)**, bước cuối cùng là thiết lập "Tiêu chí câu hỏi". Giảng viên nhập tổng số lượng câu hỏi và điều chỉnh thanh trượt tỷ lệ độ khó. Hệ thống tự động tính toán để đảm bảo tổng tỷ lệ là 100% trước khi tạo đề.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi



Hình 4.10 Giao diện tạo mới bài kiểm tra - Bước 4



Hình 4.11 Giao diện quản lý danh sách bài kiểm tra

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sau khi hoàn tất các bước tạo, hệ thống chuyển hướng về trang Quản lý danh sách bài kiểm tra như **Hình 4.11**. Tại đây liệt kê toàn bộ các bài thi đã tạo trong khóa học dưới dạng lưới thẻ. Mỗi thẻ hiển thị thông tin tóm tắt gồm tên bài, thời gian làm bài và trạng thái.

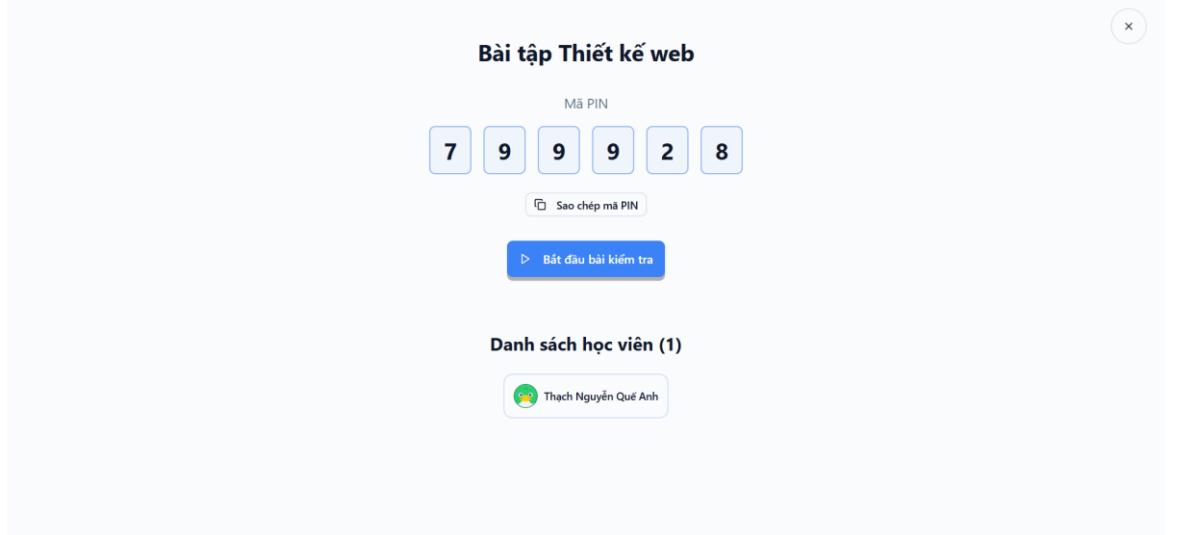
Nếu cần kiểm tra lại nội dung đề thi trước khi tổ chức, giảng viên truy cập vào giao diện Chi tiết bài kiểm tra tại **Hình 4.12**. Trang này hiển thị danh sách các câu hỏi đã được hệ thống chọn lọc ngẫu nhiên kèm mã PIN truy cập. Giảng viên có thể sử dụng tính năng trộn câu hỏi hoặc chỉnh sửa lại nếu cần thiết.

The screenshot shows the Synlearnia platform interface. On the left, there's a sidebar with navigation links: 'Tổng quan', 'Khóa học', 'Quản lý câu hỏi', 'Quản lý sinh viên', and 'Quản lý Quiz'. The main area is titled 'Chi tiết bài kiểm tra' and shows details for '04-02'. It includes a timestamp 'Thứ ba, 05/06/2025', a duration '65 phút', a PIN '266876', and start/end times '05/06/2025 07:37' and '05/06/2025 08:42'. Below this, a section titled 'Danh sách câu hỏi (54)' lists two questions:

- Question 1: Trắc nghiệm (Đúng) - 'HTTP là gì?'
- Question 2: Trắc nghiệm (Trung bình) - 'Thẻ <input type="Submit"> dùng để làm gì?'

Hình 4.12 Chi tiết bài kiểm tra

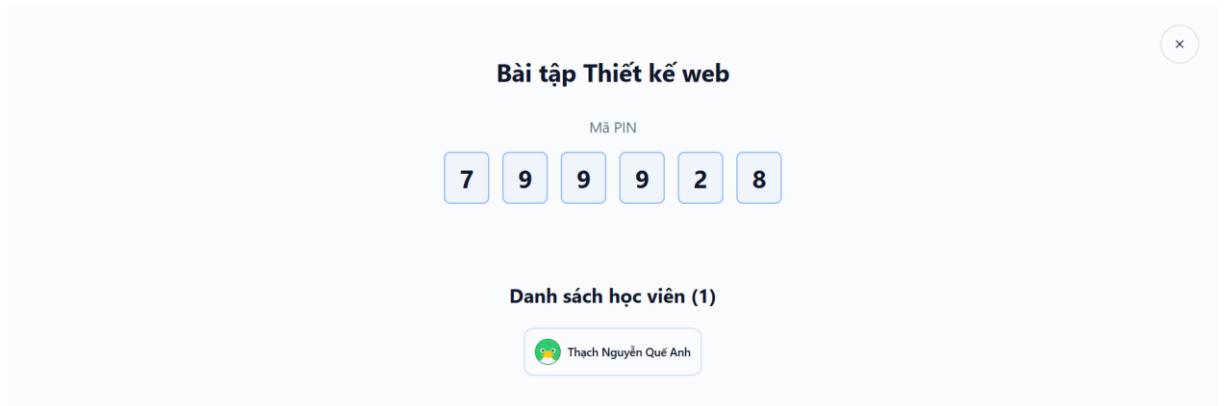
Để tiến hành thi, từ danh sách bài kiểm tra, giảng viên nhấn nút chức năng để vào phòng, mở Màn hình sảnh chờ như **Hình 4.13**. Mã PIN tham gia được hiển thị rất lớn ở trung tâm. Danh sách sinh viên đã vào phòng chờ sẽ xuất hiện theo thời gian thực ở phía dưới, giúp giảng viên kiểm soát số lượng trước khi nhấn nút "Bắt đầu".



Hình 4.13 Màn hình sẵn chờ bài kiểm tra

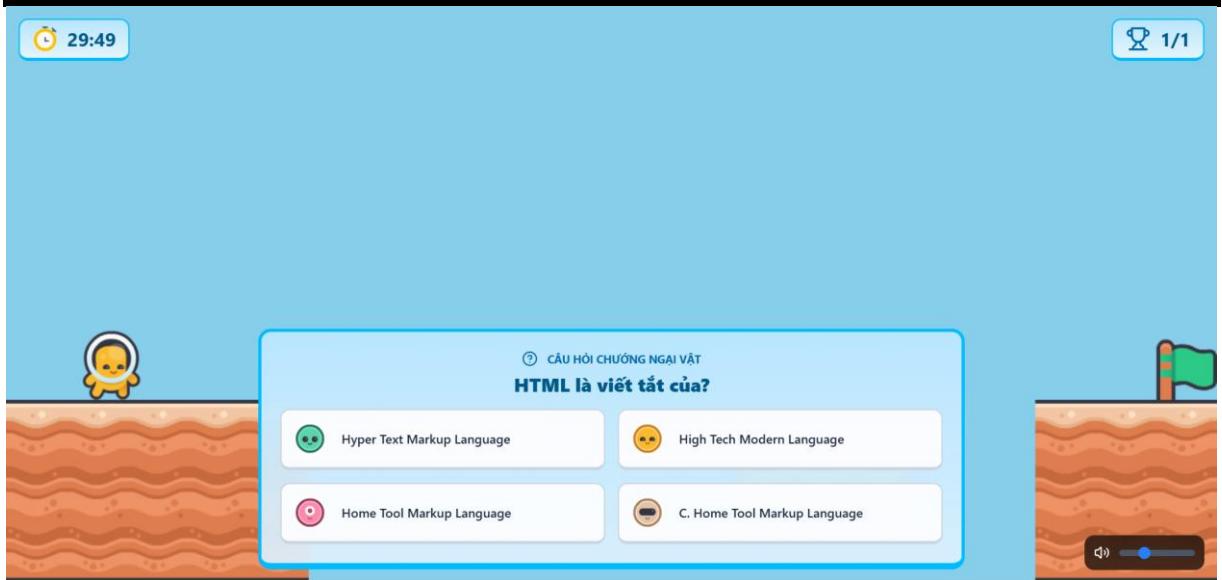
#### 4.2.3 Sinh viên làm bài dạng Game

Từ phía sinh viên, sau khi nhập mã PIN, hệ thống chuyển đến màn hình chờ như **Hình 4.14**. Màn hình hiển thị trạng thái "Đang chờ giảng viên bắt đầu" cùng thông tin cá nhân. Sinh viên có thể sử dụng nút biểu tượng cảm xúc ở góc dưới để tương tác trong lúc chờ đợi.

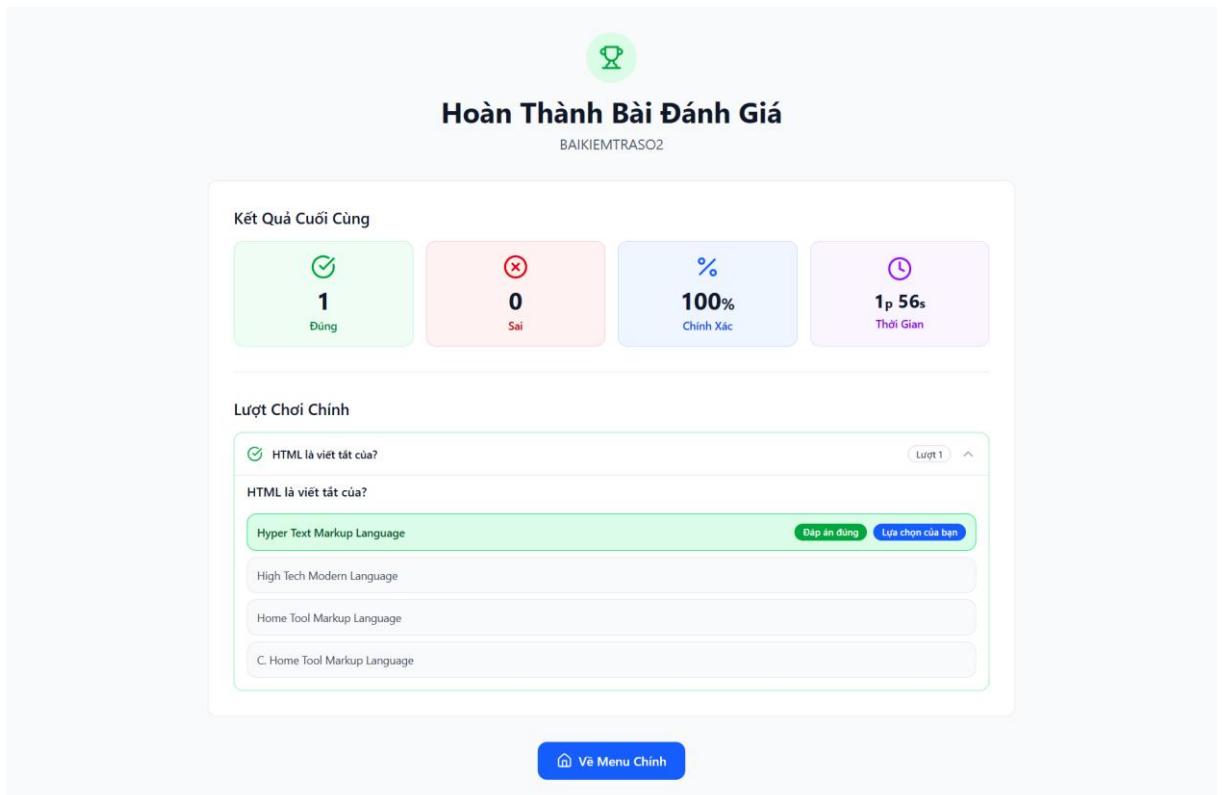


Hình 4.14 Màn hình chờ tham gia của sinh viên

Khi bài thi bắt đầu, giao diện chuyển sang phong cách trò chơi như **Hình 4.15**. Màn hình mô phỏng bối cảnh nhân vật vượt chướng ngại vật. Câu hỏi trắc nghiệm hiển thị nổi bật ở giữa cùng 4 phương án. Các yếu tố như thời gian, bảng xếp hạng và điều khiển âm thanh được bố trí ở các góc, tạo trải nghiệm thi đấu thú vị.



Hình 4.15 Giao diện bài kiểm tra theo phong cách trò chơi



Hình 4.16 Giao diện hoàn thành bài kiểm tra chế độ đánh giá

Ngay khi hoàn thành tất cả các câu hỏi, hệ thống sẽ hiển thị giao diện **Tổng kết bài làm** như **Hình 4.16**. Tại đây, sinh viên nhận được phản hồi tức thì về hiệu suất của mình, bao gồm các chỉ số quan trọng: số câu đúng, số câu sai, tỷ lệ chính xác và tổng thời gian hoàn thành. Phần "Lượt chơi chính" cho phép sinh viên xem lại ngay lập tức

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

nội dung câu hỏi và đối chiếu đáp án mình đã chọn với đáp án đúng, giúp rút kinh nghiệm nhanh trước khi nhấn nút "Về Menu Chính" để thoát ra.

#### 4.2.4 Sinh viên xem kết quả và xếp hạng

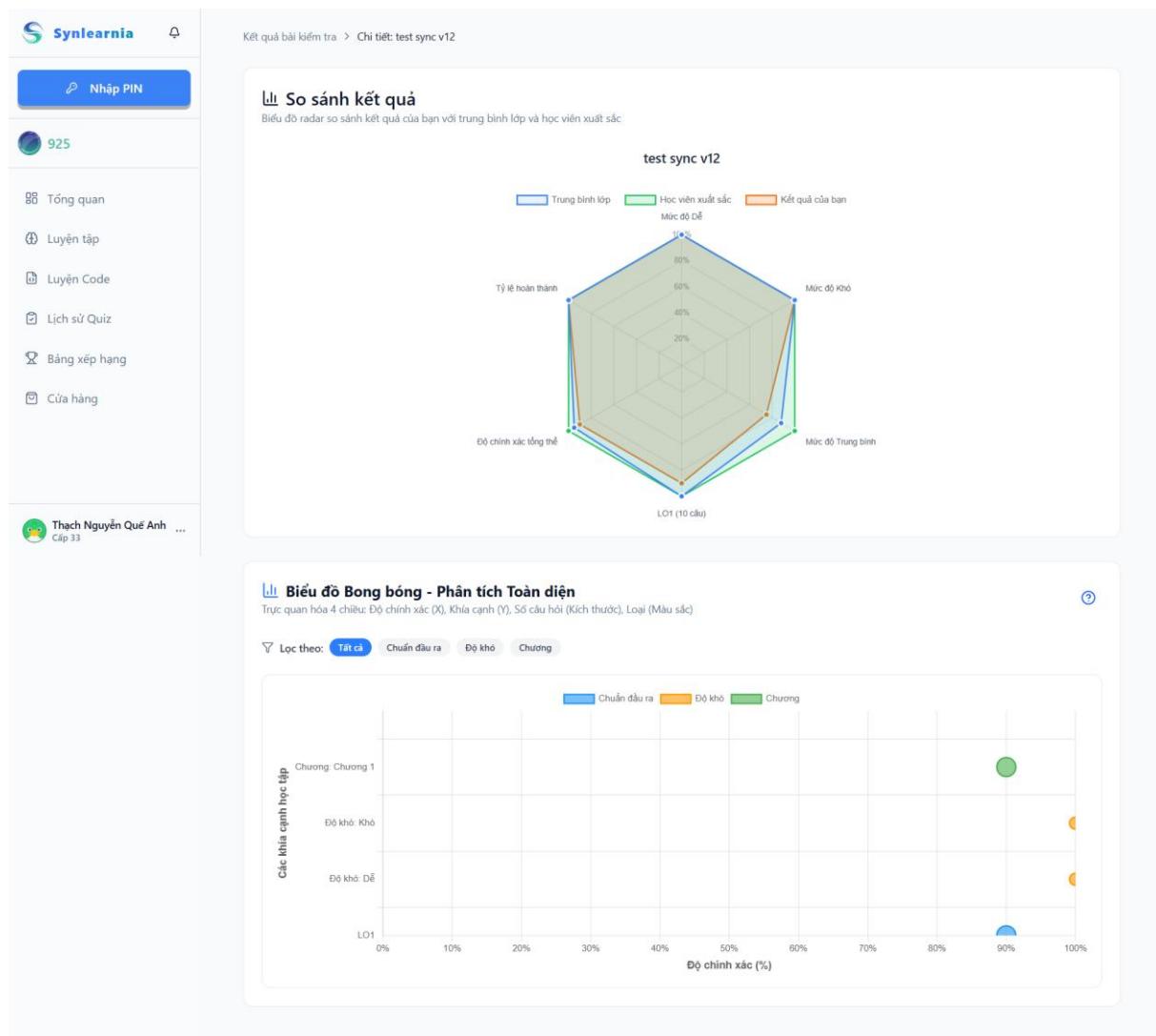
Sau khi hoàn thành bài thi và quay trở lại màn hình chính, để xem lại thành tích của mình, sinh viên truy cập vào trang Lịch sử kết quả bài kiểm tra như **Hình 4.17**. Giao diện này lưu trữ toàn bộ các bài thi đã thực hiện, hiển thị danh sách tóm tắt gồm tên bài, điểm số đạt được, thời gian hoàn thành và trạng thái được thể hiện bằng màu sắc trực quan.

The screenshot shows the Synlearn platform interface. On the left sidebar, there are navigation links: Nhập PIN, 925, Tổng quan, Luyện tập, Luyện Code, Lịch sử Quiz (which is highlighted in blue), Bảng xếp hạng, and Cửa hàng. Below this is the user's profile: Thạch Nguyễn Quế Anh, Cấp 33. The main content area is titled 'Kết Quả Bài Kiểm Tra' and subtitle 'Xem tất cả kết quả bài kiểm tra của bạn'. It includes a search bar and a message indicating 42 results found. The results are listed in a table with columns for score (9.6, 9.6, 3.5, 3.5, 10, 9), name (test, test code, test phân tích v2, test, test điểm v4, test sync v12), date (30/11/2025 16:03, 30/11/2025 08:53, 28/11/2025 14:23, 28/11/2025 13:44, 09/10/2025 15:25, 03/10/2025 15:32), and a 'Chi tiết' button for each row. At the bottom, it says 'Hiển thị 19 - 24 trong tổng số 42 kết quả (Trang 4 / 7)' and shows a page navigation bar with buttons for Trước, 1, 2, 3, 4, 5, 6, 7, Sau.

Hình 4.17 Lịch sử kết quả bài kiểm tra

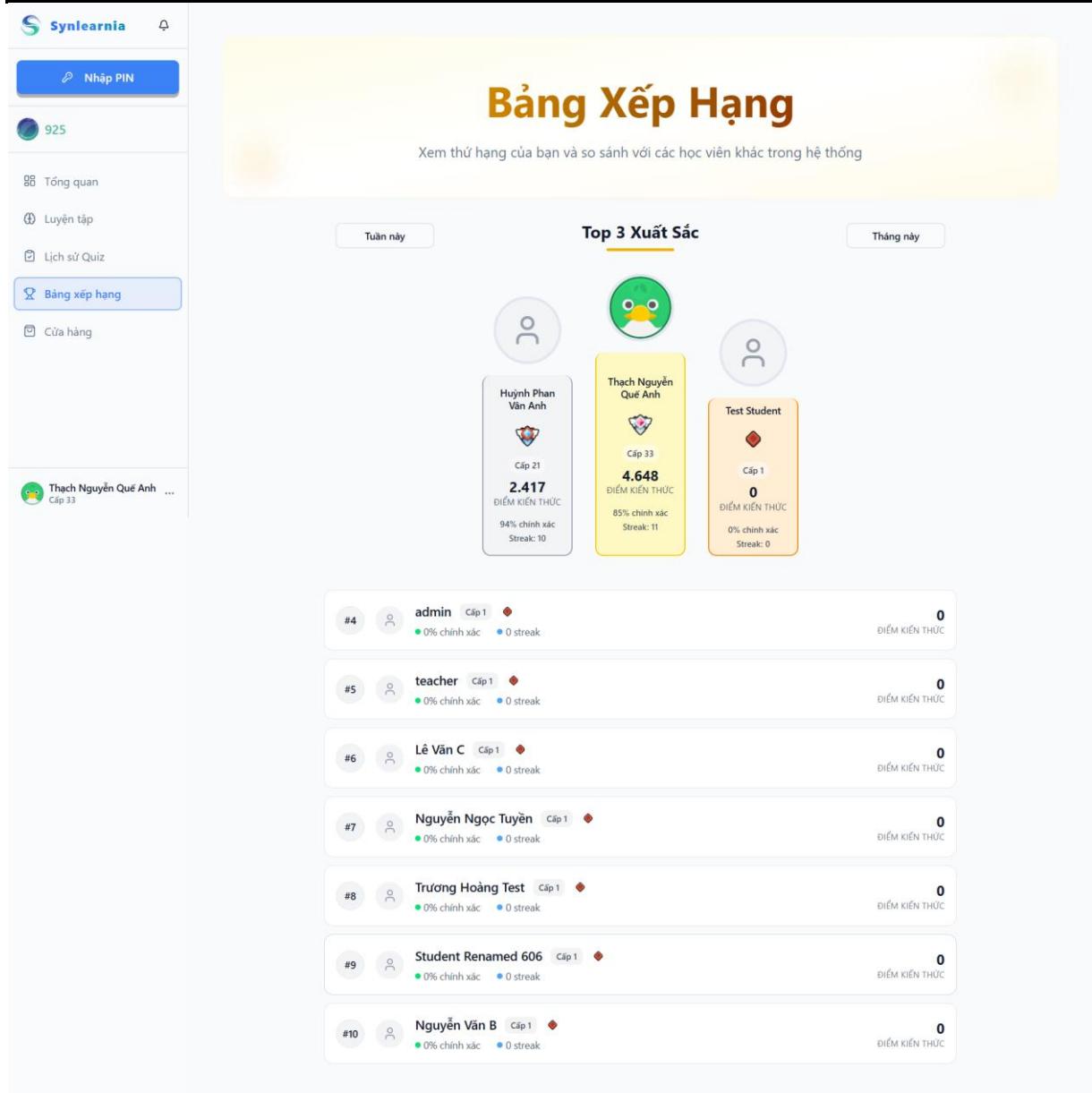
Khi nhấn vào một bài kiểm tra cụ thể trong danh sách lịch sử, hệ thống sẽ điều hướng đến giao diện Chi tiết kết quả và phân tích so sánh tại **Hình 4.18**. Đây là màn hình phân tích chuyên sâu dành cho sinh viên. Biểu đồ Radar trung tâm cho phép so sánh trực quan hiệu suất của cá nhân với mức trung bình lớp và nhóm sinh viên xuất

Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi sắc. Bên dưới, biểu đồ Bong bóng giúp sinh viên nhận diện mối tương quan giữa độ khó câu hỏi, chuẩn đầu ra và kết quả thực tế để có kế hoạch cải thiện phù hợp.



Hình 4.18 Chi tiết kết quả bài kiểm tra và phân tích so sánh

Sau khi tự đánh giá năng lực, sinh viên truy cập vào Bảng xếp hạng tại **Hình 4.19** để xem vị trí thi đua của mình trong lớp. Hệ thống vinh danh Top 3 sinh viên dẫn đầu trên bục hạng trang trọng, đồng thời liệt kê chi tiết thứ hạng của các thành viên còn lại, hỗ trợ bộ lọc thời gian theo tuần hoặc tháng.



Hình 4.19 Bảng xếp hạng thành tích sinh viên

#### 4.2.5 Quản trị viên quản lý hệ thống

Quy trình quản trị bắt đầu từ việc xây dựng khung chương trình. Tại giao diện **Quản lý chương trình đào tạo** (Hình 4.20), Admin theo dõi danh sách các ngành học hiện có. Mỗi chương trình được hiển thị dưới dạng thẻ tóm tắt, cho phép tìm kiếm, thêm mới hoặc truy cập vào chi tiết.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Hình 4.20 Giao diện quản lý chương trình đào tạo

Khi truy cập vào một chương trình cụ thể, giao diện **Chi tiết chương trình** (Hình 4.21) hiện ra với các tab chức năng chuyên biệt. Admin có thể xem mô tả tổng quan và thực hiện các thiết lập sâu hơn thông qua thanh điều hướng phía trên.

Hình 4.21 Giao diện chi tiết chương trình đào tạo

Đầu tiên, Admin thiết lập mục tiêu tại giao diện **Quản lý Mục tiêu chương trình** (Hình 4.22). Danh sách các mục tiêu được liệt kê rõ ràng kèm mô tả chi tiết, Admin có thể thêm mới hoặc chỉnh sửa để định hướng đầu ra cho ngành học.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

The screenshot shows the Synlearnia platform's interface for managing programme details. The left sidebar includes links for 'Tổng quan', 'Chương trình ĐT' (selected), 'Khóa đào tạo', and 'Quản lý tài khoản'. The right panel displays a table of learning outcomes (POs) with their descriptions and associated PLOs. The 'POs' tab is active, showing PO1 through PO4. Each row includes a checkbox, the PO name, a description, associated PLOs, and a 'Thao tác' (Action) column.

Tên PO ↑	Mô tả	PLOs	Thao tác
PO1	Có kiến thức cơ bản về khoa học xã hội, khoa học chính trị, pháp luật, ...	PLO1 PLO6 +2	...
PO3	Có khả năng làm việc hiệu quả trong lĩnh vực công nghệ thông tin, tạo...	PLO1 PLO2 +7	...
PO2	Có kiến thức và kỹ năng chuyên môn trong lĩnh vực công nghệ thông t...	PLO2 PLO3 +2	...
PO5	Có khả năng khám phá tri thức, giải quyết vấn đề, tư duy hệ thống, ph...	PLO2 PLO3 +5	...
PO4	Có khả năng làm việc độc lập hoặc làm việc theo nhóm, hướng dẫn, gi...	PLO7 PLO8 +1	...
PO1: Kiến thức chuyên môn	Sinh viên có kiến thức chuyên môn vững vàng	—	...

Hình 4.22 Quản lý Mục tiêu chương trình

Tiếp theo là quản lý **Chuẩn đầu ra chương trình** tại **Hình 4.23**. Bảng dữ liệu hiển thị chi tiết mã PLO, nội dung mô tả và danh sách các mục tiêu liên quan. Đây là cơ sở quan trọng để đánh giá năng lực sinh viên sau này.

The screenshot shows the Synlearnia platform's interface for managing programme details. The left sidebar includes links for 'Tổng quan', 'Chương trình ĐT' (selected), 'Khóa đào tạo', and 'Quản lý tài khoản'. The right panel displays a table of learning outcomes (PLOs) with their descriptions, associated POs, and associated subjects. The 'PLOs' tab is active, showing PLO1 through PLO10. Each row includes a checkbox, the PLO name, a description, associated POs, associated subjects, and a 'Thao tác' (Action) column.

Tên PLO ↑	Mô tả	PO liên quan	Môn học	Thao tác
PLO1	Vận dụng kiến thức cơ bản về khoa học tự nhiên, ...	PO3 PO1	Kỹ thuật lập trình Lập trình Web +70	...
PLO2	Kết hợp kiến thức chuyên môn để giải quyết các v...	PO5 PO3 +1	Kỹ thuật lập trình Thiết kế web +1	...
PLO3	Phân tích các vấn đề tính toán phức tạp để xác đị...	PO2 PO3 +1	Kỹ thuật lập trình Thiết kế web	...
PLO4	Triển khai các giải pháp công nghệ thông tin bả...	PO5 PO3 +1	Kỹ thuật lập trình Thiết kế web	...
PLO5	Kết hợp các giải pháp có tính hệ thống để lựa chọ...	PO2 PO3 +1	Kỹ thuật lập trình Thiết kế web	...
PLO6	Sử dụng tốt tiếng Anh trong soạn thảo, đọc tài liệ...	PO5 PO3 +1	Kỹ thuật lập trình Thiết kế web	...
PLO7	Phối hợp tốt các kỹ năng trình bày và giao tiếp tr...	PO4 PO3	Kỹ thuật lập trình Thiết kế web	...
PLO8	Phối hợp hiệu quả các kỹ năng làm việc nhóm với...	PO4 PO3 +1	Kỹ thuật lập trình Thiết kế web	...
PLO9	Tôn trọng trách nhiệm nghề nghiệp để đưa ra các...	PO5 PO4 +2	Kỹ thuật lập trình Thiết kế web	...
PLO10	Tôn trọng sự cần thiết của quá trình học tập suốt ...	PO5	Thiết kế web Kỹ thuật lập trình	...

Hình 4.23 Quản lý Chuẩn đầu ra chương trình

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Để đảm bảo tính liên kết chặt chẽ, Admin sử dụng **Ma trận PO - PLO** tại **Hình 4.24**. Giao diện dạng bảng cho phép tích chọn các ô giao nhau để xác định mối quan hệ đóng góp giữa từng Chuẩn đầu ra với Mục tiêu chương trình tương ứng.

The screenshot shows a 'Ma trận liên kết PO - PLO' (Matrix linking PLOs to POs) interface. On the left, a sidebar menu includes 'Tổng quan', 'Chương trình ĐT' (selected), 'Khóa đào tạo', and 'Quản lý tài khoản'. The main area has tabs for 'Chi tiết', 'POs', 'PLOs', 'Môn học', and 'Ma trận' (selected). Below is a section titled 'Ma trận liên kết PO - PLO' with a note: 'Đánh dấu vào các ô để tạo liên kết giữa Mục tiêu đào tạo (cột) và Chuẩn đầu ra (hàng). Nhấn "Lưu thay đổi" để áp dụng.' A table lists PLOs (PLO1-PLO7) as rows and POs (PO1-PO4) as columns. Checkmarks indicate the relationship between specific PLOs and POs.

Chuẩn đầu ra (PLO)	PO1	PO3	PO2	PO5	PO4	PO1: Kĩ chuyê
PLO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PLO2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLO3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLO4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLO5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLO6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLO7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Hình 4.24 Giao diện ma trận liên kết PO – PLO

The screenshot shows the 'Quản lý danh sách môn học' (Manage course list) interface. The sidebar includes 'Tổng quan', 'Chương trình ĐT' (selected), 'Khóa đào tạo', and 'Quản lý tài khoản'. The main area has tabs for 'Chi tiết', 'POs', 'PLOs', 'Môn học' (selected), and 'Ma trận'. It features a search bar ('Tim kiem theo tên môn học...'), a 'Tạo môn học mới' button, and filters for 'Sắp xếp: Tên', 'Trạng thái: Tất cả', 'Loại: Tất cả', and 'Hiển thị: 10'. A checkbox 'Chọn tất cả (10 môn học)' is present. Below is a grid of course cards categorized by program:

Môn học	Thiết kế web	An toàn và bảo mật	Anh văn chuyên ngành
Môn học	# STT: 1	HK: 1	HK: 5
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Anh văn không chuyên 1			
Môn học	HK: 1	HK: 2	HK: 3
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Anh văn không chuyên 2			
Môn học	HK: 4	HK: 2	HK: 4
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Anh văn không chuyên 3			
Môn học	HK: 4	HK: 3	HK: 4
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Anh văn không chuyên 4			
Môn học	HK: 4	Cấu trúc dữ liệu và giải thuật	Chủ nghĩa xã hội khoa học
Bất buộc	Hoạt động	HK: 2	HK: 4
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Cấu trúc dữ liệu và giải thuật			
Môn học	HK: 2	Bất buộc	Bất buộc
Bất buộc	Hoạt động	Bất buộc	Hoạt động
Chủ nghĩa xã hội khoa học			

Hình 4.25 Quản lý danh sách môn học thuộc chương trình

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sau khi hoàn tất khung chương trình, Admin quản lý các học phần tại giao diện **Danh sách môn học (Hình 4.25)**. Các môn học được hiển thị dưới dạng lưới thẻ với thông tin tên môn, số tín chỉ và trạng thái, giúp Admin dễ dàng sắp xếp kế hoạch giảng dạy.

Đi sâu vào cấu trúc từng môn, giao diện **Chi tiết học phần (Hình 4.26)** cung cấp thông tin mô tả và các công cụ quản lý nội dung. Tại đây, Admin có thể cập nhật thông tin chung hoặc điều hướng sang các phần cấu hình khác của môn học.

The screenshot shows the Synlearn platform's 'Chi tiết học phần' (Detail course) page for the 'Thiết kế web' course. The page has a header with tabs: Thông tin (selected), CDR, Chương, Ma trận, and Khóa học. Below the tabs, there's a sub-header 'Thiết kế web' and 'Mã học phần: #1'. A 'Chỉnh sửa' button is visible. The main content area includes a 'Mô tả học phần' section with a detailed description of the course goals and objectives. A 'Sau khi hoàn thành học phần này, sinh viên có thể:' section lists 10 learning outcomes (LO1 to LO10) and a 'Hiểu cơ bản về HTML' section. On the left sidebar, there are links for Tổng quan, Chương trình ĐT (selected), Khóa đào tạo, and Quản lý tài khoản. The top navigation bar shows 'Chương trình ĐT > Công nghệ thông tin > Thiết kế web'. The bottom left corner shows the user 'admin' (Quản trị viên).

Hình 4.26 Giao diện chi tiết học phần

Admin định nghĩa các yêu cầu cụ thể đối với sinh viên tại giao diện **Quản lý chuẩn đầu ra học phần (Hình 4.27)**. Các chuẩn đầu ra được liệt kê chi tiết, gắn kết trực tiếp với nội dung giảng dạy để đảm bảo chất lượng đào tạo.

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

The screenshot shows the Synlearnia platform's 'Chương trình ĐT' (Mobile Program) section. The 'Chương' tab is active, showing a list of learning outcomes (LO1-LO10) with their descriptions and associated chapters. Admin 'admin' is logged in.

Tên chuẩn đầu ra	Mô tả	Chương	Hành động
LO1	Xác định nguyên tắc hoạt động của Xác định nguyên tắc ho...	Chương 1	...
LO2	Thiết kế trang Web với ngôn ngữ HTML.	Chương 2	...
LO3	Cài đặt định dạng trang Web với bảng định kiểu CSS	Chương 3	...
LO4	Vận dụng ngôn ngữ Javascript trong xử lý tương tác người d...	Chương 4	...
LO5	Thiết kế Responsive Web với Framework	Chương 5	...
LO6	Kết hợp HTML, CSS, Javascript và Responsive để thiết kế giao...	Chương 4, Chương 2, Chương 3 +1	...
LO7	Thể hiện khả năng giao tiếp, trình bày vấn đề	Chưa liên kết chương	...
LO8	Thể hiện khả năng làm việc nhóm	Chưa liên kết chương	...
LO9	Thể hiện đạo đức nghề nghiệp trong xây dựng Website	Chưa liên kết chương	...
LO10	Xác định vai trò của công nghệ Web đối với đời sống xã hội	Chưa liên kết chương	...

Hình 4.27 Quản lý chuẩn đầu ra học phần

Nội dung học phần được tổ chức thông qua giao diện **Quản lý cấu trúc chương (Hình 4.28)**. Admin xây dựng đề cương chi tiết, sắp xếp thứ tự các chương học để tạo ra lộ trình kiến thức logic cho sinh viên.

The screenshot shows the Synlearnia platform's 'Chương' section. The 'Chương' tab is active, showing a list of chapters (Chương 1-5) with their descriptions and associated learning outcomes (LO1-LO6).

Tên chương	Mô tả	Chuẩn đầu ra	Hành động
Chương 1	Tổng quan về thiết kế web	LO1	...
Chương 2	Tìm hiểu ngôn ngữ HTML	LO2, LO6	...
Chương 3	Bảng định kiểu CSS	LO6, LO3	...
Chương 4	Ngôn ngữ Javascript	LO4, LO6	...
Chương 5	Thiết kế Responsive Web	LO5, LO6	...

Hình 4.28 Quản lý cấu trúc chương học

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Để quản lý nội dung sâu hơn, Admin truy cập vào **Chi tiết nội dung chương (Hình 4.29)**. Giao diện này cho phép thêm mới hoặc chỉnh sửa các mục bài học bên trong mỗi chương, đảm bảo nội dung phong phú và đầy đủ.

The screenshot shows the 'Chi tiết nội dung chương' (Chapter Content Details) page. On the left is a sidebar with navigation links: Tổng quan (General), **Chương trình ĐT** (DT Program), Khóa đào tạo (Training Course), and Quản lý tài khoản (Account Management). The main content area shows the 'Mô tả chương' (Chapter Description) and 'Nội dung chương' (Chapter Content). The content includes three sections: 1.1. Nguyên tắc hoạt động của Web, 1.2. Tiến trình thiết kế web, and 1.3. Công cụ phân tích, kiểm lỗi và tối ưu hóa mã nguồn tự động (Linting tools). A 'Thêm Section' (Add Section) button is located at the top right of the content list. To the right of the content is a sidebar with 'Thông tin' (Information): MÔN HỌC (Subject), Thiết kế web (Web Design), CHƯƠNG TRÌNH (Program), Công nghệ thông tin (Information Technology), SỐ SECTIONS (Number of Sections) 3, and CHUẨN ĐẦU RA (Learning Outcomes) 1. Below this is a 'Lợi' (Benefit) button. At the bottom right of the content area is a 'Chỉnh sửa' (Edit) button.

Hình 4.29 Chi tiết nội dung chương học

Để hoàn tất thiết kế môn học, Admin thiết lập **Ma trận Chương - Chuẩn đầu ra** tại **Hình 4.30**. Việc tích chọn các ô trong ma trận giúp xác định rõ chương học nào sẽ đóng góp vào việc đạt được chuẩn đầu ra nào của học phần.

The screenshot shows the 'Ma trận liên kết Chapter - LO' (Chapter-LO Matrix) page. On the left is a sidebar with navigation links: Tổng quan (General), **Chương trình ĐT** (DT Program), Khóa đào tạo (Training Course), and Quản lý tài khoản (Account Management). The main content area shows a table titled 'Ma trận liên kết Chapter - LO'. The table has columns for 'Chương học (Chapter)' (Chương 1, Chương 2, Chương 3, Chương 4, Chương 5) and rows for 'LO1', 'LO2', 'LO3', 'LO4', 'LO5'. Checkmarks indicate which chapters are linked to which learning outcomes. A 'Lưu thay đổi' (Save changes) button is located at the bottom right of the matrix.

Chương học (Chapter)	LO1	LO2	LO3	LO4	LO5
Chương 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chương 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chương 3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chương 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Chương 5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Hình 4.30 Ma trận liên kết Chương học và Chuẩn đầu ra

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Sau khi hoàn thiện nội dung đào tạo, Admin chuyển sang công tác tổ chức tại giao diện **Quản lý Khóa đào tạo (Hình 4.31)**. Tại đây, Admin tạo các khóa học mới theo niên khóa để bắt đầu chu kỳ giảng dạy.

The screenshot shows the Synlearnia platform's course management section. The sidebar has links for 'Tổng quan', 'Chương trình ĐT', 'Khóa đào tạo' (which is highlighted in blue), and 'Quản lý tài khoản'. The main content area is titled 'Khóa đào tạo' and subtitle 'Quản lý các khóa đào tạo trong hệ thống'. It includes a search bar, a 'Thêm khóa đào tạo' button, and sorting/filtering options. Two course cards are listed: 'CNTT 2023-2025' (Category: Công nghệ thông tin, Status: Khóa đào tạo CNTT, Year: Niên khóa: 2023 - 2025) and 'K2024' (Category: Công nghệ thông tin, Status: Niên khóa: 2024 - 2028). A message at the bottom says 'Hiển thị 1 - 2 trong tổng số 2 kết quả'.

Hình 4.31 Giao diện quản lý khóa đào tạo

Trong mỗi khóa đào tạo, Admin quản lý thông tin tại giao diện **Chi tiết khóa đào tạo (Hình 4.32)**. Trang này cung cấp các thông kê quan trọng như số lượng học kỳ, số lượt phân công và tổng số lớp học hiện có trong khóa.

The screenshot shows the detailed view of the 'CNTT 2023-2025' course. The sidebar is identical to the previous screenshot. The main content area has tabs for 'Chi tiết' (selected) and 'Học kỳ & Phân công'. Under 'Chi tiết khóa đào tạo: CNTT 2023-2025', it shows 'Học kỳ 9', 'Phân công 2', and 'Lớp học 3'. Below this are sections for 'Thông tin cơ bản' (Course name: CNTT 2023-2025, ID: #1, Year: 2023 - 2025) and 'Chương trình đào tạo' (Program: Công nghệ thông tin). At the bottom, there's a 'Danh sách lớp học' section with three entries: 'Lập trình Java' (ID: 13), 'thiết kế web 01 - test' (ID: 14), and 'test ảnh' (ID: 15).

Hình 4.32 Giao diện chi tiết khóa đào tạo

## Phát triển Giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi

Bước quan trọng tiếp theo là **Quản lý học kỳ và Phân công giảng dạy (Hình 4.33)**. Admin sử dụng bảng ma trận trực quan để gán giảng viên phụ trách cho từng môn học trong học kỳ cụ thể một cách nhanh chóng.

The screenshot shows a user interface for managing academic periods and assignments. On the left sidebar, there are links for 'Tổng quan', 'Chương trình ĐT', 'Khóa đào tạo' (highlighted in blue), and 'Quản lý tài khoản'. The main content area has a header 'Khóa đào tạo > CNTT 2023-2025'. Below it is a navigation bar with tabs: 'Chi tiết' (selected), 'Học kỳ & Phân công' (highlighted in blue), 'Học kỳ II - Năm 1' (highlighted in blue), 'Học kỳ I - Năm 1', 'Học kỳ test', 'Học kỳ I - Năm 2', 'Học kỳ II - Năm 2', 'Học kỳ I - Năm 3', 'Học kỳ II - Năm 3', 'Học kỳ I - Năm 4', and '+ Tạo học kỳ'. A sub-section titled 'Học kỳ II - Năm 1' (highlighted in green) is currently active. It displays a table where subjects are mapped to teachers. The columns are 'Môn học' (Subject), 'Teacher John' (Teacher assigned), and 'teacher' (Original teacher). The subjects listed are 'Thiết kế web', 'Kỹ thuật lập trình', 'Test Subject - Updated', 'Lý thuyết đồ thị', 'Subject Fix 319', and 'Phân tích và thiết kế hệ thống thông tin'. Each subject row has two checkboxes: one for the assigned teacher and one for the original teacher.

Hình 4.33 Giao diện quản lý học kỳ và phân công giảng dạy

Cuối cùng, để vận hành hệ thống, Admin sử dụng giao diện **Quản lý tài khoản (Hình 4.34)**. Danh sách toàn bộ người dùng được hiển thị, cho phép Admin thực hiện cấp mới, chỉnh sửa hoặc đặt lại mật khẩu cho tài khoản.

The screenshot shows a user interface for managing accounts. On the left sidebar, there are links for 'Tổng quan', 'Chương trình ĐT', 'Khóa đào tạo' (highlighted in blue), and 'Quản lý tài khoản' (highlighted in blue). The main content area has a title 'Quản lý tài khoản' and a subtitle 'Quản lý tài khoản Admin và Giảng viên trong hệ thống'. Below is a search bar 'Q. Tìm kiếm tài khoản...' and a 'Thêm tài khoản' button. A filter 'Lọc vai trò: Tất cả' is also present. A table lists users with columns: 'Mã' (ID), 'Họ tên' (Name), 'Email', 'Vai trò' (Role), and 'Thao tác' (Actions). The roles shown are 'Giảng viên' (lecturer), 'Admin', and another 'Giảng viên'. For each user, there is a '... More' button and a context menu with options: 'Chỉnh sửa' (Edit), 'Đặt lại mật khẩu' (Reset password), and 'Xóa' (Delete). The users listed are Teacher John (ID 2), admin (ID 7), teacher (ID 8), and Lê Văn C (ID 257).

Hình 4.34 Giao diện quản lý tài khoản người dùng

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết quả đạt được

Sau quá trình nghiên cứu và triển khai, đề tài “Phát triển giao diện Frontend cho Nền tảng học trực tuyến có yếu tố trò chơi” đã đạt được các kết quả sau:

Xây dựng thành công giao diện frontend hiện đại, trực quan, dễ sử dụng, phân biệt rõ ba vai trò quản trị viên, giảng viên và sinh viên.

Tích hợp các yếu tố trò chơi như hệ thống điểm thưởng, bảng xếp hạng, thành tích cá nhân và các phần thưởng nhằm thúc đẩy động lực, duy trì sự tương tác và nâng cao trải nghiệm học tập của sinh viên.

Hỗ trợ các chức năng quản lý mạnh mẽ cho quản trị viên, giảng viên và sinh viên.

Áp dụng các công nghệ hiện đại: Next.js cho phát triển web, Tailwind CSS và Shadcnui cho thiết kế giao diện thống nhất, Chart.js để trực quan hóa số liệu, TanStack Query quản lý truy vấn dữ liệu, Socket.IO cho tương tác thời gian thực, và Phaser.js cho các mini-game luyện tập.

Đảm bảo khả năng mở rộng, bảo mật, hiệu năng tối ưu cho các tác vụ realtime và trải nghiệm đồng nhất trên nhiều thiết bị.

### 5.2. Hướng phát triển

Mở rộng các tính năng mới cho website như: tích hợp AI, nhiều loại câu hỏi,...

Tối ưu hóa trang web cho trải nghiệm người dùng tốt hơn.

## TÀI LIỆU THAM KHẢO

- [1] Mozilla Developer Network, “Web Technology for Developers.” 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web>
- [2] IBM, “Client–Server Architecture.” 2023. [Online]. Available: <https://www.ibm.com/docs/en>
- [3] World Wide Web Consortium, “Architecture of the World Wide Web.” 2004. [Online]. Available: <https://www.w3.org/TR/webarch/>
- [4] Vercel, “Next.js Documentation.” Accessed: Jan. 15, 2025. [Online]. Available: <https://nextjs.org/docs>
- [5] Phaser.io, “What is Phaser?” Accessed: Jan. 15, 2025. [Online]. Available: <https://docs.phaser.io/phaser/getting-started/what-is-phaser>
- [6] Tailwind Labs, “Tailwind CSS Documentation.” Accessed: Jan. 15, 2025. [Online]. Available: <https://tailwindcss.com/docs>
- [7] shadcn, “shadcn/ui Documentation.” Accessed: Jan. 15, 2025. [Online]. Available: <https://ui.shadcn.com/docs>
- [8] TanStack, “TanStack Query Documentation for React.” Accessed: Jan. 15, 2025. [Online]. Available: <https://tanstack.com/query/latest/docs/framework/react/overview>