

TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ



ÂU KIM HÀ

**XÂY DỰNG CÔNG CỤ ĐÁNH GIÁ
CHẤT LƯỢNG MÃ HTML, CSS,
JAVASCRIPT TỰ ĐỘNG**

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

VĨNH LONG, NĂM 2025

TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ



XÂY DỰNG CÔNG CỤ ĐÁNH GIÁ
CHẤT LƯỢNG MÃ HTML, CSS,
JAVASCRIPT TỰ ĐỘNG

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn : ThS NGUYỄN NGỌC ĐAN THANH

Sinh viên thực hiện: ÂU KIM HÀ

Mã số sinh viên: 110121020

Lớp: DA21TTC

Khoá: 2021 - 2025

VĨNH LONG, NĂM 2025

LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, các trang web đã trở thành một phần quen thuộc trong đời sống hằng ngày, từ việc học tập, làm việc đến giải trí. Để xây dựng một trang web, các lập trình viên thường sử dụng ba ngôn ngữ chính là HTML, CSS và JAVASCRIPT. Tuy nhiên, việc đảm bảo chất lượng mã nguồn của các ngôn ngữ này là một thách thức, đặc biệt khi dự án ngày càng phức tạp và quy mô lớn. Những lỗi cú pháp, sai chính tả thuộc tính, giá trị không hợp lệ hay cấu trúc mã không tối ưu có thể dẫn đến hiệu suất thấp, trải nghiệm người dùng kém, thậm chí gây ra các lỗi nghiêm trọng trên trang web.

Xuất phát từ nhu cầu thực tiễn đó, tôi đã lựa chọn đề tài “Xây dựng công cụ đánh giá chất lượng mã HTML, CSS, JAVASCRIPT tự động” nhằm mục tiêu phát triển một công cụ hỗ trợ lập trình viên trong việc kiểm tra và sửa lỗi mã nguồn một cách nhanh chóng, hiệu quả. Công cụ không chỉ giúp phát hiện các lỗi phổ biến mà còn đưa ra gợi ý sửa lỗi, từ đó nâng cao chất lượng mã và giảm thiểu thời gian kiểm tra thủ công.

Báo cáo trình bày chi tiết quá trình nghiên cứu, thiết kế và phát triển công cụ, bao gồm các phương pháp, công nghệ được sử dụng, cũng như những kết quả đạt được và hướng phát triển trong tương lai. Tôi hy vọng rằng sản phẩm này sẽ mang lại giá trị thiết thực cho cộng đồng lập trình viên và góp phần vào sự phát triển của lĩnh vực phát triển web.

LỜI CAM ĐOAN

Tôi xin cam đoan toàn bộ nội dung trong khóa luận tốt nghiệp này là kết quả của quá trình nghiên cứu và tìm hiểu của tôi. Các dữ liệu, hình ảnh, bảng biểu và kết quả được trình bày một cách trung thực. Nếu có bất kỳ hành vi nào sao chép hay vi phạm bản quyền được phát hiện tôi hoàn toàn chịu toàn bộ trách nhiệm.

Vĩnh Long, ngày tháng năm 20....

Sinh viên thực hiện

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Lời nói đầu tiên tôi xin gửi lời cảm ơn chân thành đến quý thầy, cô trong Khoa Công nghệ Thông tin những người đã tận tình giảng dạy và trang bị cho tôi những kiến thức nền tảng quý báu trong suốt quá trình học tập. Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc đến giảng viên Nguyễn Ngọc Đan Thanh người đã trực tiếp hướng dẫn, hỗ trợ và định hướng tôi trong quá trình thực hiện đề tài này.

Mặc dù đã cố gắng hết sức, báo cáo này khó tránh khỏi những thiếu sót, tôi rất mong nhận được những ý kiến đóng góp để hoàn thiện hơn.

Một lần nữa, tôi xin chân thành cảm ơn!

Sinh viên thực hiện

Âu Kim Hà

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN.....	1
1.1 Lý do chọn đề tài.....	1
1.2 Mục tiêu nghiên cứu.....	1
1.3 Phạm vi nghiên cứu.....	1
1.4 Đối tượng nghiên cứu.....	2
1.5 Phương pháp nghiên cứu.....	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 Tổng quan về HTML, CSS và JAVASCRIPT.....	4
2.1.1 Ngôn ngữ HTML	4
2.1.2 Bảng định kiểu CSS	4
2.1.3 Ngôn ngữ JavaScript.....	5
2.2 Phân loại lỗi trong kiểm tra mã.....	6
2.2.1 Lỗi cú pháp.....	6
2.2.2 Lỗi ngữ nghĩa	7
2.2.3 Cảnh báo.....	8
2.3 Các công cụ kiểm tra mã.....	9
2.3.1 HTMLHint	9
2.3.2 JSHint.....	10
2.3.3 Stylelint	11
2.3.4 ESLint	12
2.3.5 W3C Validator	13
2.4 Các thư viện kiểm tra mã	15
2.4.1 Thư viện Cssutils.....	15

2.4.2 Thư viện requests	16
2.4.3 Thư viện BeautifulSoup	18
2.4.4 Thư viện Lxml.....	20
2.4.5 Thư viện re	21
2.4.6 Thư viện os.....	22
2.5 Công nghệ và công cụ sử dụng	23
2.5.1 Công nghệ nền tảng.....	23
2.5.2 Ngôn ngữ lập trình	24
2.5.3 Công cụ hỗ trợ phát triển	24
2.6 Cơ sở dữ liệu	25
2.7 Kiến trúc hệ thống.....	25
2.8 Nhận xét và định hướng phát triển công cụ	26
2.8.1 Những khó khăn và thách thức	26
2.8.2 Phương pháp phát triển và tích hợp công cụ.....	27
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	28
3.1 Mô tả đề tài.....	28
3.2 Đặc tả yêu cầu hệ thống	28
3.2.1 Yêu cầu chức năng	28
3.2.2 Yêu cầu phi chức năng	29
3.3. Chức năng của người dùng	29
3.3.1 Quyền quản trị.....	29
3.3.2 Quyền giảng viên	29
3.3.3 Quyền sinh viên.....	29
3.3 Thiết kế dữ liệu	30
3.3.1 Mô hình cơ sở dữ liệu	30
3.3.2 Chi tiết các thực thể.....	30

3.4 Thiết kế xử lý	35
3.4.1 Mô hình Use Case	35
3.4.2 Sơ đồ hoạt động.....	38
3.5 Thiết kế giao diện.....	46
3.5.1 Thiết kế giao diện đăng nhập	46
3.5.2 Thiết kế giao diện trang đăng ký.....	47
3.5.3 Thiết kế giao diện chủ	48
3.5.4 Thiết kế giao diện trang thêm giảng viên.....	49
3.5.5 Thiết kế giao diện trang thêm sinh viên.....	50
3.5.6 Thiết kế giao diện trang thông tin giảng viên	50
3.5.7 Thiết kế giao diện trang thông tin sinh viên.....	51
3.5.8 Thiết kế giao diện trang tạo lớp học.....	52
3.5.9 Thiết kế giao diện trang tìm kiếm lớp học	53
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	54
4.1 Chức năng hệ thống.....	54
4.1.1 Giao diện đăng ký	54
4.1.2 Giao diện đăng nhập.....	55
4.1.3 Giao diện trang chủ	55
4.1.4 Giao diện kiểm tra mã HTML.....	56
4.1.5 Giao diện kiểm tra mã CSS.....	57
4.1.6 Giao diện kiểm tra JAVASCRIPT	57
4.2 Chức năng quản trị, giảng viên	58
4.2.1 Giao diện xem giảng viên	58
4.2.2 Giao diện xem sinh viên.....	58
4.2.3 Giao diện tạo bài tập	59
4.2.4 Giao diện xem danh sách bài tập	59

4.2.5 Giao diện sửa bài tập.....	60
4.2.6 Giao diện tạo lớp học	60
4.2.7 Giao diện xem thông tin lớp học.....	61
4.2.8 Giao diện chi tiết lớp học	61
4.2.9 Giao diện sửa lớp học.....	62
4.2.10 Giao diện thêm giảng viên	62
4.2.11 Giao diện thêm sinh viên	63
4.3 Chức năng sinh viên.....	63
4.3.1 Giao diện trang chủ	63
4.3.2 Giao diện trang tìm kiếm lớp học	64
4.3.3 Giao diện trang lớp học.....	64
4.3.4 Giao diện trang chi tiết lớp học.....	65
4.4 Ưu điểm và hạn chế.....	65
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	67
5.1 Kết Luận.....	67
5.1.1 Kết luận kiểm tra mã HTML.....	67
5.1.2 Kết luận kiểm tra mã CSS.....	67
5.1.3 Kết luận kiểm tra mã JAVASCRIPT	68
5.2. Hướng phát triển	68

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

STT	Từ viết tắt	Giải nghĩa
1	API	Application Programming Interface
2	CD/ID	Compact Disc/Identifier
3	CSS	Cascading Style Sheets
4	ES	ECMAScript
5	HTML	HyperText Markup Language
6	HTTP	Hypertext Transfer Protocol
7	HTTPS	Hypertext Transfer Protocol Secure
8	JS	JavaScript
9	W3C	World Wide Web Consortium
10	XML	eXtensible Markup Language
11	XSLT	eXtensible Stylesheet Language Transformations

DANH MỤC CÁC BẢNG

Bảng 3.1 Chi tiết thực thể nguoi_dung	30
Bảng 3.2 Bảng chi tiết thực thể lop_hoc	31
Bảng 3.3 Bảng chi tiết thực thể sinh_vien_lop_hoc	32
Bảng 3.4 Bảng chi tiết thực thể bai_tap	32
Bảng 3.5 Bảng chi tiết thực thể bai_tap_lop_hoc	33
Bảng 3.6 Bảng chi tiết thực thể bai_nop	33
Bảng 3.8 Bảng chi tiết thực thểkiem_tra_ma.....	34

DANH MỤC HÌNH ẢNH

Hình 2.1 Giao diện w3c Validation	13
Hình 2.2 Thư viện cssutils	15
Hình 2.3 Thư viện Requests.....	17
Hình 2.4 Thư viện BeautifulSoup	18
Hình 2.5 Thư viện Lxml.....	20
Hình 3.1 Mô hình cơ sở dữ liệu	30
Hình 3.2 Mô hình Use Case quản trị.....	35
Hình 3.3 Mô hình Use Case giảng viên	36
Hình 3.4 Mô hình Use Case sinh viên	37
Hình 3.5 Sơ đồ hoạt động quản trị	38
Hình 3.6 Sơ đồ hoạt động giảng viên.....	39
Hình 3.7 Mô hình hoạt động quyền sinh viên.....	40
Hình 3.8 Sơ đồ hoạt động kiểm tra mã HTML, CSS, JAVASCRIPT	41
Hình 3.9 Sơ đồ hoạt động tạo bài tập.....	42
Hình 3.10 Sơ đồ hoạt động tạo lớp học.....	43
Hình 3.11 Sơ đồ hoạt động tham gia lớp học.....	44
Hình 3.12 Sơ đồ hoạt động nộp bài tập.....	45
Hình 3.13 Phác thảo giao diện trang đăng nhập	46
Hình 3.14 Phác thảo giao diện trang đăng ký	47
Hình 3.15 Phác thảo giao diện trang chủ	48
Hình 3.16 Phác thảo giao diện trang thêm giảng viên	49
Hình 3.17 Phác thảo giao diện trang thêm sinh viên	50
Hình 3.18 Phác thảo giao diện trang thông tin giảng viên.....	50
Hình 3.19 Phác thảo giao diện trang thông tin sinh viên	51

Hình 3.20 Phác thảo giao diện trang tạo lớp học	52
Hình 3.21 Phác thảo giao diện trang tìm kiếm lớp học.....	53
Hình 4.1 Giao diện trang đăng ký	54
Hình 4.2 Giao diện trang đăng nhập	55
Hình 4.3 Giao diện trang chủ	55
Hình 4.4 Giao diện trang kiểm tra mã HTML	56
Hình 4.5 Giao diện kiểm tra mã CSS.....	57
Hình 4.6 Giao diện kiểm tra mã JAVASCRIPT	57
Hình 4.7 Giao diện xem giảng viên	58
Hình 4.8 Giao diện xem sinh viên.....	58
Hình 4.9 Giao diện tạo bài tập	59
Hình 4.10 Giao diện xem danh sách bài tập	59
Hình 4.11 Giao diện sửa bài tập.....	60
Hình 4.12 Giao diện tạo lớp học	60
Hình 4.13 Giao diện xem danh sách lớp học	61
Hình 4.14 Giao diện chi tiết lớp học	61
Hình 4.15 Giao diện sửa thông tin lớp học	62
Hình 4.16 Giao diện thêm giảng viên	62
Hình 4.17 Giao diện thêm sinh viên	63
Hình 4.18 Giao diện trang chủ quyền sinh viên.....	63
Hình 4.19 Giao diện trang tìm kiếm lớp học	64
Hình 4.20 Giao diện trang lớp học	64
Hình 4.21 Giao diện trang chi tiết lớp học.....	65

NHẬN XÉT

(Của giảng viên chấm trong đồ án, khoá luận của sinh viên)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giảng viên chấm

(ký và ghi rõ họ tên)

NHẬN XÉT

(Của giảng viên chấm trong đề án, khoá luận của sinh viên)

Giảng viên chấm

(ký và ghi rõ họ tên)

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên hướng dẫn)

Họ và tên sinh viên: Âu Kim Hà

MSSV: 110121020

Ngành: Công nghệ Thông tin.

Khóa: 2021

Tên đề tài: Xây dựng công cụ đánh giá chất lượng mã HTML, CSS, JAVASCRIPT tự động

Họ và tên Giáo viên hướng dẫn: Nguyễn Ngọc Đan Thanh

Chức danh: Giảng viên

Học vị: Thạc sĩ

NHẬN XÉT

1. Nội dung đề tài:

Đề tài tập trung phát triển một hệ thống hỗ trợ luyện tập và đánh giá tự động mã HTML, CSS, JAVASCRIPT dành cho sinh viên trong các lớp học lập trình web. Hệ thống cho phép giảng viên tạo bài tập, quản lý lớp học và phân bổ bài tập theo lớp, trong khi sinh viên có thể làm bài trực tuyến và được phản hồi tự động. Đây là công cụ mang tính hỗ trợ giảng dạy và tự học hiệu quả, hướng đến việc cải thiện kỹ năng lập trình giao diện web cho sinh viên.

2. Ưu điểm:

Giao diện hệ thống thân thiện, phù hợp cho cả giảng viên và sinh viên.

Tích hợp đầy đủ các chức năng quản lý lớp học, bài tập và thông tin người dùng.

Cho phép giảng viên phân bổ bài tập theo từng lớp học, tạo sự linh hoạt và kiểm soát tốt tiến trình học tập.

Sinh viên có thể luyện tập trực tiếp và nhận phản hồi tự động, từ đó tự cải thiện chất lượng mã.

Phân quyền người dùng rõ ràng (sinh viên, giảng viên, quản trị viên), đảm bảo an toàn và hiệu quả trong sử dụng.

3. Khuyết điểm:

Hệ thống đánh giá mã mới chỉ dừng ở mức cơ bản, chưa kiểm tra được các lỗi logic phức tạp như vòng lặp vô hạn, cú pháp lồng nhau sai, hay selector CSS sai.

Thông báo lỗi trong quá trình kiểm tra mã còn sơ sài, chưa đủ rõ ràng để người học hiểu và sửa.

Chưa có cơ chế phản hồi hướng dẫn cụ thể từng lỗi, điều này làm giảm tính giáo dục của hệ thống.

Chưa hỗ trợ tính năng thống kê tiến độ học tập và mức độ hoàn thành bài tập để giảng viên dễ theo dõi.

4. Điểm mới đề tài:

Hệ thống không chỉ kiểm tra mã mà còn kết hợp mô hình lớp học và quản lý bài tập theo lớp, tạo ra môi trường mô phỏng gần với lớp học thật.

Tích hợp bài tập tự động chấm và phản hồi, giúp sinh viên luyện tập chủ động, đồng thời giảm tải cho giảng viên.

Hướng đến mô hình tự học có hướng dẫn, kết hợp giữa tự động hóa và quản lý lớp học – điều mà ít công cụ mã nguồn mở hiện tại thực hiện đầy đủ.

5. Giá trị thực trên đề tài:

Hỗ trợ giảng viên dễ dàng tạo, phân phối và chấm bài tập lập trình web, tiết kiệm thời gian và tăng tính công bằng.

Giúp sinh viên luyện tập mã trực tiếp, tự kiểm tra và cải thiện kỹ năng lập trình HTML, CSS, JAVASCRIPT – các công nghệ thiết yếu trong phát triển web.

Có thể triển khai tại các trường đại học, trung tâm đào tạo lập trình hoặc lớp học trực tuyến.

Nền tảng mở có thể phát triển thêm để áp dụng vào nhiều môn học khác trong khối Công nghệ thông tin.

6. Đề nghị sửa chữa bổ sung:

Nâng cấp khả năng phân tích mã để phát hiện lỗi logic nâng cao, bao gồm: selector sai, thiếu đóng thẻ, sai cú pháp trong vòng lặp hoặc khai báo hàm.

Cải tiến thông báo lỗi rõ ràng, có gợi ý hướng sửa, đặc biệt hữu ích cho sinh viên mới học lập trình.

Tích hợp chức năng chấm điểm tự động có trọng số, giúp giảng viên linh hoạt trong đánh giá.

Bổ sung báo cáo thống kê tiến độ học tập và số lần làm bài để hỗ trợ theo dõi lớp học hiệu quả hơn.

Xây dựng kho bài tập mẫu, phân loại theo mức độ và kỹ năng để sử dụng lại và mở rộng hệ thống.

7. Đánh giá:

Đề tài được triển khai nghiêm túc, định hướng rõ ràng và mang tính ứng dụng thực tiễn cao trong môi trường giáo dục lập trình web. Việc kết hợp giữa quản lý lớp học, phân phối bài tập và tự động đánh giá mã giúp nâng cao hiệu quả dạy – học, đồng thời giảm tải công việc cho giảng viên. Tuy hệ thống vẫn còn một số tính năng cần hoàn thiện, đặc biệt là ở phần đánh giá chuyên sâu và phản hồi lỗi, nhưng về tổng thể, đây là một đề tài có giá trị cao trong giảng dạy lập trình hiện đại và xứng đáng được đánh giá Tốt.

....., ngày tháng năm 20....

Giảng viên hướng dẫn

(Ký & ghi rõ họ tên)

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét:

Chức danh: Học vị:

Chuyên ngành:

Cơ quan công tác:

Tên sinh viên:

Tên đề tài đồ án, khóa luận tốt nghiệp:

.....

.....

I. Ý KIẾN NHẬN XÉT

1. Nội dung:

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Điểm mới các kết quả của đồ án, khóa luận:

.....

3. Ứng dụng thực tế:

(Các câu hỏi của giáo viên phản biện)

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

.....

.....

III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đề án khóa luận tốt nghiệp)

.....

.....

.....

.....

.....

....., ngày tháng năm 20...

Người nhận xét

(Ký & ghi rõ họ tên)

CHƯƠNG 1: TỔNG QUAN

1.1 Lý do chọn đề tài

Trong lĩnh vực phát triển web, việc đảm bảo chất lượng mã nguồn là một yếu tố quan trọng nhằm đảm bảo hiệu suất và tính ổn định của sản phẩm. Tuy nhiên, các lập trình viên thường gặp khó khăn trong việc kiểm tra thủ công mã HTML, CSS và JAVASCRIPT. Những lỗi như cú pháp sai, thuộc tính không hợp lệ, hoặc giá trị không phù hợp có thể gây ra các vấn đề nghiêm trọng, từ việc hiển thị sai giao diện đến làm chậm tốc độ tải trang.

Hiện nay, mặc dù đã có một số công cụ kiểm tra mã tự động như W3C Validator hay Stylelint, nhưng yêu cầu môi trường cài đặt phức tạp. Điều này khiến cho việc kiểm tra và sửa lỗi trở nên tốn thời gian và không hiệu quả đối với các lập trình viên, đặc biệt là những người mới bắt đầu.

Vì thế, tôi quyết định nghiên cứu và phát triển một công cụ đánh giá chất lượng mã HTML, CSS, JAVASCRIPT tự động, với mục tiêu cung cấp một giải pháp đơn giản, hiệu quả, có khả năng phát hiện lỗi và đưa ra gợi ý sửa lỗi ngay lập tức. Công cụ này sẽ giúp lập trình viên tiết kiệm thời gian, nâng cao chất lượng mã và cải thiện trải nghiệm phát triển web.

1.2 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một công cụ web tự động hỗ trợ đánh giá chất lượng mã HTML, CSS và JAVASCRIPT.

Phát triển một công cụ web cho phép người dùng nhập mã HTML, CSS, JAVASCRIPT hoặc tải file mã nguồn và kiểm tra.

Phát hiện các lỗi phổ biến, đánh dấu vị trí và cung cấp gợi ý sửa lỗi.

Tích hợp các công cụ kiểm tra mã hiện đại như Stylelint và W3C Validator để nâng cao độ chính xác.

1.3 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào ba ngôn ngữ phổ biến trong lập trình web là HTML, CSS và JAVASCRIPT.

HTML: Kiểm tra lỗi thẻ không đóng, thẻ không hợp lệ, sai cú pháp và thuộc tính không chuẩn.

CSS: Phát hiện lỗi cú pháp, thuộc tính sai, giá trị không hợp lệ, lỗi thiếu dấu ;.

JAVASCRIPT: Kiểm tra lỗi cú pháp cơ bản, sử dụng biến chưa khai báo, đoạn mã dễ gây lỗi hoặc vòng lặp vô hạn.

1.4 Đối tượng nghiên cứu

Mã nguồn HTML, CSS, JAVASCRIPT: Là đầu vào chính của hệ thống, bao gồm cả mã đúng và mã có lỗi được kiểm tra, phân tích và xử lý.

Lập trình viên và sinh viên công nghệ thông tin: Là nhóm người dùng mục tiêu, đặc biệt là những người mới học lập trình web, có nhu cầu kiểm tra và sửa mã một cách nhanh chóng, trực quan.

Các công cụ phân tích mã nguồn tự động hiện có: Bao gồm W3C Validator, HTMLHint, Stylelint, ESLint... để khảo sát, so sánh và tích hợp nhằm nâng cao hiệu quả kiểm tra.

Các kỹ thuật kiểm tra và gợi ý sửa lỗi mã nguồn: Nghiên cứu các phương pháp như phân tích cú pháp, biểu thức chính quy, phát hiện mẫu lỗi để áp dụng vào công cụ.

1.5 Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

Sử dụng tài liệu tham khảo từ các nguồn đáng tin cậy như sách, bài báo khoa học, tài liệu chính thức của W3C, và tài liệu hướng dẫn của các công cụ như Stylelint, ESLint, và HTMLHint. Phương pháp này tập trung vào việc phân tích các tiêu chuẩn lập trình web (HTML5, CSS3), các loại lỗi phổ biến trong mã nguồn, cũng như các thuật toán và phát hiện lỗi. Từ đó, xây dựng nền tảng lý thuyết và quy trình kiểm tra mã hiệu quả.

Phương pháp nghiên cứu thực nghiệm:

Áp dụng các thử nghiệm thực tế trên tập dữ liệu mã nguồn mẫu bao gồm HTML, CSS, và JavaScript với các lỗi cố ý được chèn vào (ví dụ: thẻ không đóng, thuộc tính sai, biến chưa khai báo). Sử dụng các công cụ hiện có (W3C Validator, Stylelint, ESLint) để so sánh kết quả và đánh giá độ chính xác của công cụ tự phát triển. Quá trình thực nghiệm bao gồm việc viết mã kiểm tra, tích hợp API hoặc thư viện phù hợp, và tối ưu hóa giao diện người dùng dựa trên phản hồi từ người dùng. Kết quả thực nghiệm sẽ được phân tích để cải thiện thuật toán phát hiện lỗi và gợi ý sửa chữa, đảm bảo công cụ đáp ứng nhu cầu thực tế.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về HTML, CSS và JAVASCRIPT

2.1.1 Ngôn ngữ HTML

HTML là ngôn ngữ đánh dấu chuẩn được sử dụng để định nghĩa cấu trúc và nội dung của một trang web. Một tài liệu HTML hợp lệ thường bắt đầu với `<!DOCTYPE html>` và bao gồm các phần tử cơ bản như `<html>`, `<head>`, và `<body>`. Mỗi phần tử HTML được định nghĩa bằng các thẻ (tags), ví dụ như `<p>` để tạo đoạn văn, `<a>` để tạo liên kết, hoặc `` để hiển thị hình ảnh. [5]

Ví dụ:

```
<!DOCTYPE html>

<html>

  <head><title>Trang chủ</title></head>

  <body>

    <h1>Chào mừng!</h1>

    <p>Đây là một đoạn văn.</p>

  </body>

</html>
```

Các lỗi sai phổ biến trong HTML

- Thiếu thẻ đóng
- Lòn thẻ sai thứ tự
- Sử dụng thuộc tính không hợp lệ

2.1.2 Bảng định kiểu CSS

CSS là ngôn ngữ dùng để định nghĩa cách trình bày của các phần tử HTML. bao gồm màu sắc, bố cục, kích thước, phông chữ, và hiệu ứng.

Ví dụ:

```
p {  
    color: blue;  
    font-size: 16px;  
}
```

Các lỗi phổ biến trong CSS

- Sai chính tả thuộc tính: Viết colour thay vì color hoặc fomt-size thay vì font-size
- Giá trị không hợp lệ: Sử dụng giá trị không được hỗ trợ, như display: invalid thay vì display: block
- Thiếu dấu chấm phẩy: Quên dấu ; giữa các thuộc tính, như p { color: blue font-size: 16px }, dẫn đến lỗi phân tích cú pháp.

2.1.3 Ngôn ngữ JavaScript

JavaScript là ngôn ngữ lập trình động, được sử dụng để thêm tính tương tác cho trang web, chẳng hạn như xử lý sự kiện (click, hover), gọi API, hoặc thay đổi nội dung trang theo thời gian thực. JavaScript thường được nhúng trong thẻ <script> hoặc thông qua tệp .js bên ngoài

Ví dụ:

```
document.getElementById("btn").onclick = function ()  
{  
    alert("Bạn vừa nhấn nút!");  
}
```

Các lỗi phổ biến trong JavaScript

- Lỗi cú pháp: Thiếu dấu ngoặc đóng, ví dụ function example() { return true;

-
- Biến chưa khai báo: Sử dụng biến mà không khai báo trước, như `x = 5`; thay vì `let x = 5`;, gây lỗi trong chế độ strict mode
 - Sự kiện không được xử lý đúng: Gán sự kiện không hợp lệ, như `element.onclick = "invalidFunction"`;

2.2 Phân loại lỗi trong kiểm tra mã

2.2.1 Lỗi cú pháp

Lỗi cú pháp là loại lỗi cơ bản nhất, xảy ra khi mã nguồn không tuân thủ đúng quy tắc cú pháp của ngôn ngữ lập trình, khiến trình duyệt hoặc trình biên dịch không thể hiểu và thực thi mã đúng cách. Trong HTML, lỗi cú pháp thường gặp như thiếu thẻ đóng hoặc viết sai cú pháp thuộc tính.

HTML: Thiếu thẻ đóng (`<div>` không có `</div>`), hoặc viết sai cú pháp thuộc tính (`` thay vì ``).

Ví dụ: Lỗi thiếu giá trị thuộc tính

```
<img src>
```

Cú pháp đúng:

```

```

CSS: Thiếu dấu chấm phẩy giữa các thuộc tính (`color: blue font-size: 16px`), hoặc viết sai cú pháp selector (`.class#id` không hợp lệ).

Ví dụ: Lỗi thiếu dấu chấm phẩy giữa các thuộc tính

```
p { color: blue font-size: 16px }
```

Cú pháp đúng:

```
p { color: blue; font-size: 16px; }
```

JavaScript: Thiếu dấu ngoặc đóng (`function example() {}`), hoặc quên dấu phẩy trong khai báo object (`{a: 1 b: 2}`).

Ví dụ: Lỗi thiếu dấu ngoặc, thiếu dấu phẩy trong khai báo object

```
function example() {  
    let person = { name: "Alice" age: 25 }; // Thiếu dấu phẩy  
}
```

Cú pháp đúng:

```
let person = { name: "Alice", age: 25 };
```

2.2.2 Lỗi ngữ nghĩa

Lỗi ngữ nghĩa xảy ra khi mã viết đúng cú pháp nhưng không thực hiện đúng mục đích mong muốn, hoặc gây ra hành vi sai lệch trong quá trình thực thi. Trong HTML, một ví dụ điển hình là trùng lặp id trên nhiều phần tử, điều này gây xung đột khi truy xuất bằng CSS hoặc JavaScript:

HTML: Sử dụng id trùng lặp trên nhiều thẻ, gây xung đột khi truy cập bằng JavaScript hoặc CSS.

Ví dụ: Lỗi trùng lặp id

```
<div id="header"></div>  
  
<div id="header"></div> <!-- Trùng lặp id -->
```

CSS: Đặt selector không áp dụng được, như `.non-existent-class`, hoặc sử dụng thuộc tính không phù hợp (`font-size: block`).

Ví dụ: Lỗi do selector không áp dụng được hoặc giá trị không phù hợp

```
font-size: block; /* Sai vì block không phải là giá trị hợp lệ của  
font-size */
```

JavaScript: Gán nhầm biến hoặc gọi hàm không tồn tại, như `undefinedFunction()`.

Ví dụ:

```
undefinedFunction(); // Gây lỗi runtime vì hàm chưa được định nghĩa
```

Mặc dù đoạn mã có thể không báo lỗi cú pháp, nhưng khi chạy sẽ dẫn đến lỗi thực thi hoặc không có tác dụng như mong đợi.

2.2.3 Cảnh báo

Cảnh báo không phải là lỗi nghiêm trọng, nhưng chúng ảnh hưởng đến hiệu suất, khả năng bảo trì hoặc không tuân thủ chuẩn. Trong HTML, có thể là việc sử dụng thuộc tính không được khuyến nghị.

Ví dụ: Sử dụng thuộc tính cũ

```
<p align="center">Nội dung</p>
```

Việc canh lề giữa nên được thực hiện bằng CSS *text-align: center*; thay vì dùng thuộc tính *align* đã lỗi thời.

CSS: Khai báo thuộc tính dư thừa, như `color: red; color: blue;` trong cùng một quy tắc.

Ví dụ: khai báo dư thừa

```
h1 { color: red; color: blue; }
```

JavaScript: Khai báo biến nhưng không sử dụng hoặc không tuân thủ quy ước đặt tên (ví dụ quy tắc camelCase)

```
let unusedVar = 10; // Không được dùng đến  
let User_Name = "John"; // Không theo quy ước tên biến
```

Mặc dù những lỗi này không ngăn mã hoạt động nhưng về lâu dài sẽ ảnh hưởng đến chất lượng và khả năng mở rộng hệ thống.

2.3 Các công cụ kiểm tra mã

2.3.1 HTMLHint

HTMLHint là một công cụ kiểm tra mã nguồn dành riêng cho ngôn ngữ HTML. Được thiết kế giúp lập trình viên phát hiện các lỗi cú pháp, lỗi quy tắc, các vấn đề về cấu trúc cũng như rủi ro bảo mật trong quá trình phát triển giao diện web.

HTMLHint được sử dụng để phát hiện một số lỗi phổ biến như:

- Thẻ HTML không được đóng đúng cách
- Sử dụng các thuộc tính không hợp lệ hoặc sai chính tả.
- Cách đặt tên không chuẩn hoặc cấu trúc lồng thẻ không đúng quy ước
- Thuộc tính không hợp lệ

Ưu điểm:

- Dễ tích hợp vào các hệ thống kiểm thử tự động hoặc trình soạn thảo như VSCode.
- Hỗ trợ cấu hình linh hoạt thông qua tệp .htmlhintc
- Tốc độ kiểm tra nhanh, phù hợp cho các dự án vừa và nhỏ

Nhược điểm:

- Chỉ hỗ trợ HTML tĩnh
- Không kiểm tra sâu logic nội dung
- Cần cấu hình thủ công để đạt hiệu quả cao

Ví dụ:

```
<!-- Mã có lỗi -->

<div clas="box">

    <p>Xin chào

</div>
```

Kết quả phản hồi:

Sai chính tả thuộc tính: `clas` và đề xuất `class`

Thiếu thẻ đóng `</p>`

2.3.2 JSHint

JSHint là một công cụ kiểm tra mã dành riêng cho ngôn ngữ lập trình JavaScript, công cụ được phát triển bởi Douglas Crockford. Mục tiêu chính của JSHint là giúp lập trình viên phát hiện lỗi sớm trong quá trình viết mã, từ đó nâng cao chất lượng mã nguồn, giảm thiểu các lỗi gây ra trong quá trình chạy chương trình.

Các chức năng chính của JSHint

- Phát hiện lỗi cú pháp: thiếu dấu ngoặc, thiếu dấu chấm phẩy (;), từ khóa sai.
- Kiểm tra khai báo biến: Biến chữ khai báo, khai báo biến bị đè, biến không sử dụng.

Ưu điểm:

- Nhẹ và dễ sử dụng
- Tích hợp tốt trong hệ thống CI/CD đơn giản
- Tập trung vào lỗi cơ bản

Nhược điểm:

- Không hỗ trợ tốt các phiên bản ECMAScript hiện đại (ES6+)
- Không hỗ trợ sửa lỗi tự động
- Không phân tích sâu logic hoặc flow chương trình

Ví dụ:

```
function greet() {  
    message = "Hello, world!"  
    console.log(message)  
}
```

Kết quả phản hồi:

Biến `message` chưa được khai báo (thiếu `let`, `var`, hoặc `const`)
Thiếu dấu `;` cuối dòng

2.3.3 Stylelint

Stylelint là một công cụ kiểm tra mã dành cho CSS và các ngôn ngữ mở rộng của CSS. Đây là công cụ phổ biến trong các dự án phát triển giao diện, với mục tiêu chính là giúp phát hiện lỗi cú pháp, lỗi định dạng và vi phạm quy tắc code style, từ đó cải thiện chất lượng và tính nhất quán của mã CSS.

Các chức năng chính của JSHint

- Kiểm tra cú pháp CSS
- Phát hiện giá trị không hợp lệ
- Kiểm tra quy tắc đặt tên, selector

Ưu điểm:

- Tích hợp tốt với công cụ hiện đại
- Hỗ trợ cộng đồng mạnh
- Linh hoạt sử dụng được nhiều

Nhược điểm:

- Cần môi trường Node.js
- Cấu hình ban đầu có thể phức tạp

-
- Không kiểm tra logic CSS

Ví dụ:

```
/* Mã có lỗi */  
  
.box {  
  
    colour: red  
  
    font-size: 16px  
  
}
```

Kết quả phản hồi:

```
colour không phải thuộc tính hợp lệ và đề xuất color  
  
Thiếu dấu ; giữa các thuộc tính
```

2.3.4 ESLint

ESLint là một công cụ kiểm tra mã nguồn dành cho JavaScript và các ngôn ngữ dựa trên JavaScript. Giúp phát hiện lỗi cú pháp, vi phạm quy tắc lập trình, cũng như các đoạn mã tiềm ẩn lỗi logic hoặc không tuân thủ coding style.

Chức năng chính của ESLint

- Phát hiện lỗi cú pháp
- Phân tích mã tĩnh
- Hỗ trợ plugin và môi trường hiện đại
- Tích hợp trình sửa lỗi tự động

Ưu điểm:

- Công cụ phổ biến và mạnh mẽ
- Cấu hình linh hoạt
- Tích hợp tốt với công cụ hiện đại

Nhược điểm:

- Yêu cầu môi trường Node.js

- Cấu hình ban đầu có thể phức tạp

Ví dụ:

```
let num = 5

if(num = 10){

    console.log("Lỗi rồi!")

}
```

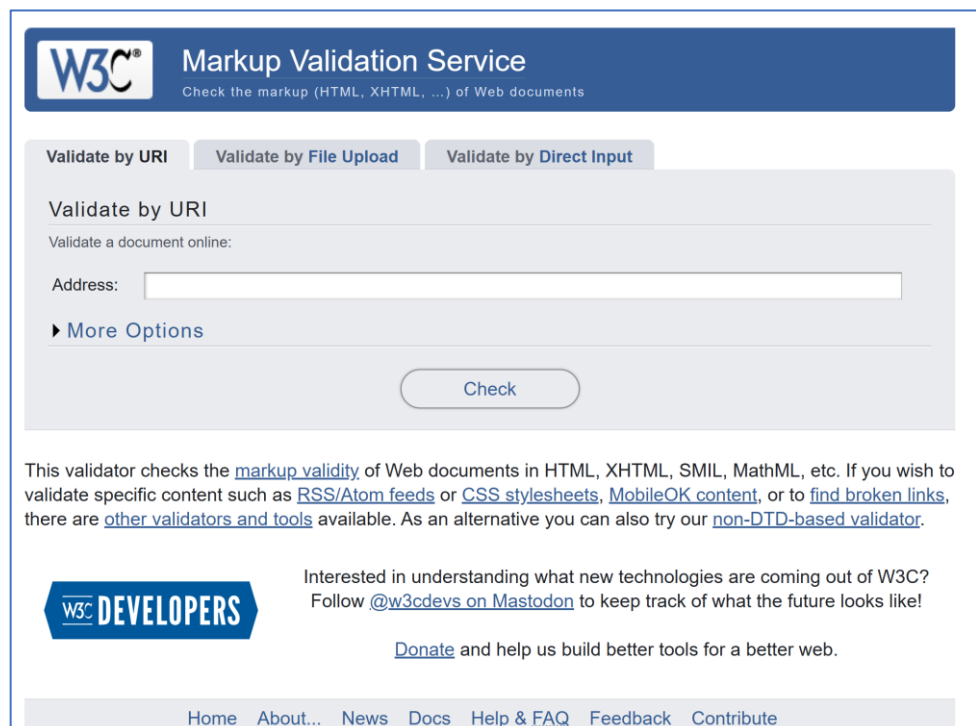
Kết quả phản hồi:

Gán nhầm = trong điều kiện (nên dùng ===)

Thiếu dấu ;

2.3.5 W3C Validator

W3C Validator là công cụ chính thức do World Wide Web Consortium (W3C) cung cấp, dùng để kiểm tra mã HTML và CSS có tuân thủ các tiêu chuẩn web hiện hành. Công cụ này giúp đảm bảo rằng mã nguồn web hợp lệ, có khả năng tương thích tốt trên nhiều trình duyệt và thiết bị khác nhau.



The image shows the W3C Markup Validation Service interface. At the top, there's a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" tab is selected. Under this tab, there's a section "Validate by URI" with the text "Validate a document online:". Below this, there's a label "Address:" followed by a text input field. Below the input field, there's a link "More Options". At the bottom of this section, there's a "Check" button. Below the "Check" button, there's a paragraph of text explaining the validator's purpose and providing links to other resources. At the bottom of the page, there's a "W3C DEVELOPERS" logo and a section titled "Interested in understanding what new technologies are coming out of W3C?" with a link to follow @w3cdevs on Mastodon. At the very bottom, there's a footer with links: Home, About..., News, Docs, Help & FAQ, Feedback, and Contribute.

Hình 2.1 Giao diện w3c Validation

Chức năng chính của W3C Validator

- Phân tích cú pháp HTML/CSS
- Phát hiện lỗi định dạng và thẻ HTML/CSS sai chuẩn
- Kiểm tra theo chuẩn cụ thể
- Hiện thị chi tiết lỗi theo dòng

Ưu điểm:

- Không cần cài đặt
- Kiểm tra HTML/CSS toàn diện
- Hiện thị rõ lỗi và gợi ý

Nhược điểm:

- Không linh hoạt cho hệ thống kiểm thử tự động
- Không kiểm tra JavaScript hoặc logic động

Ví dụ:

```
<html>

  <head><title>Trang mẫu</title></head>

  <body>

    <center>Chào bạn</center>

  </body>

</html>
```

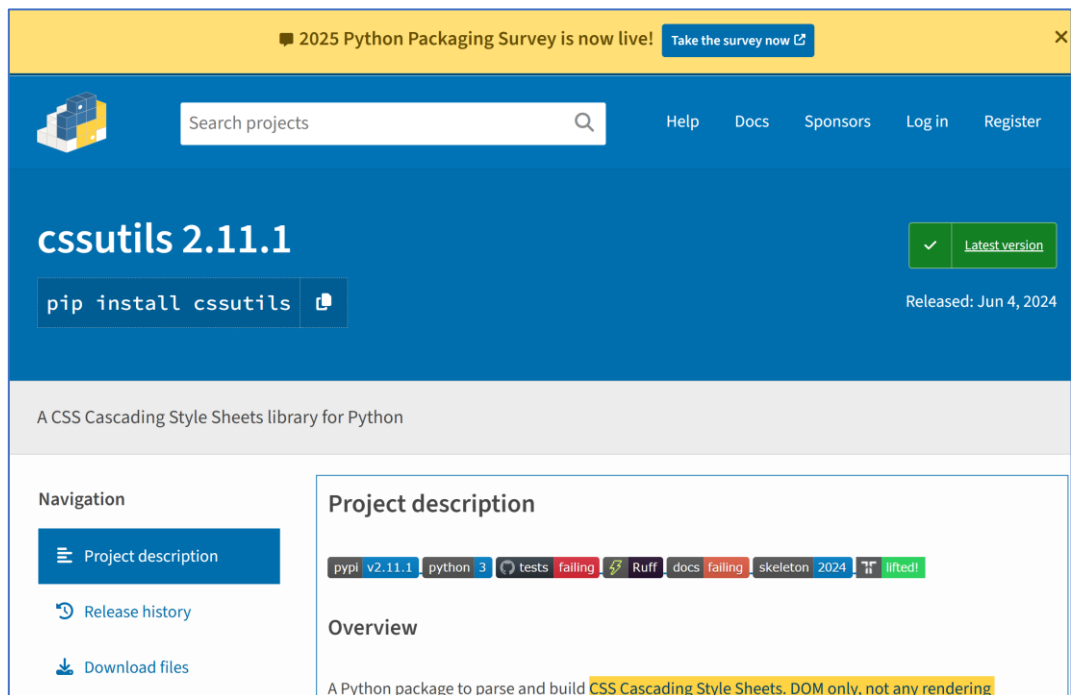
Kết quả phản hồi:

Cảnh báo: <center> là thẻ không còn được khuyến nghị, nên thay bằng CSS (text-align: center)

2.4 Các thư viện kiểm tra mã

2.4.1 Thư viện Cssutils

Cssutils là một thư viện Python dùng để phân tích cú pháp, chỉnh sửa và chuẩn hóa mã CSS. Thư viện này có khả năng trích xuất các thuộc tính CSS, kiểm tra lỗi cú pháp, đồng thời giúp chuyển đổi hoặc tái cấu trúc lại mã CSS theo chuẩn W3C. Trong công cụ kiểm tra mã, cssutils được sử dụng để phát hiện các lỗi phổ biến như thuộc tính sai chính tả, giá trị không hợp lệ hoặc thiếu dấu chấm phẩy. Đây là một thư viện lý tưởng để xây dựng trình phân tích CSS tự động.



Hình 2.2 Thư viện cssutils

Ví dụ:

```
import cssutils

css_code = "body { colour: red; font-size: 16px }"
sheet = cssutils.parseString(css_code)

for rule in sheet:
    for property in rule.style:
        print(f"{property.name} = {property.value}")
```

Kết quả:

```
colour = red
font-size = 16px
```

Lưu ý: *colour* là thuộc tính sai chính tả, thư viện vẫn phân tích được nhưng có thể đánh dấu là không chuẩn.

2.4.2 Thư viện requests

Requests là một trong những thư viện HTTP phổ biến và dễ dùng nhất trong Python, được sử dụng để gửi các yêu cầu HTTP/HTTPS như GET, POST, PUT, DELETE,... và nhận phản hồi từ các máy chủ web. Trong các hệ thống kiểm tra mã hoặc thu thập dữ liệu web, requests thường được dùng để tải mã nguồn HTML của trang web, gửi dữ liệu bài làm, hoặc lấy nội dung từ API. Đây là công cụ quan trọng nếu hệ thống cần tương tác với website từ xa hoặc máy chủ RESTful.



Requests: HTTP for Humans™

Release v2.32.4. ([Installation](#))

downloads/month 682M | license Apache-2.0 | wheel yes | python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | 3.13

Requests is an elegant and simple HTTP library for Python, built for human beings.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type": "User"...}'
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

See [similar code](#), sans [Requests](#).

Requests allows you to send HTTP/1.1 requests extremely easily. There's no need to add query strings to your URLs, or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

[Star](#) 52,966

Requests is an elegant and simple HTTP library for Python, built for human beings.

Useful Links

- Quickstart
- Advanced Usage
- API Reference
- Release History
- Contributors Guide
- Recommended Packages and Extensions

Fork me on GitHub

Hình 2.3 Thư viện Requests

Ví dụ:

```
import requests

url = "https://example.com"

response = requests.get(url)

if response.status_code == 200:

    print("Tải trang thành công!")

    print("Nội dung trang:", response.text[:100]) # In 100 ký tự
đầu

else:

    print("Lỗi khi truy cập trang.")
```

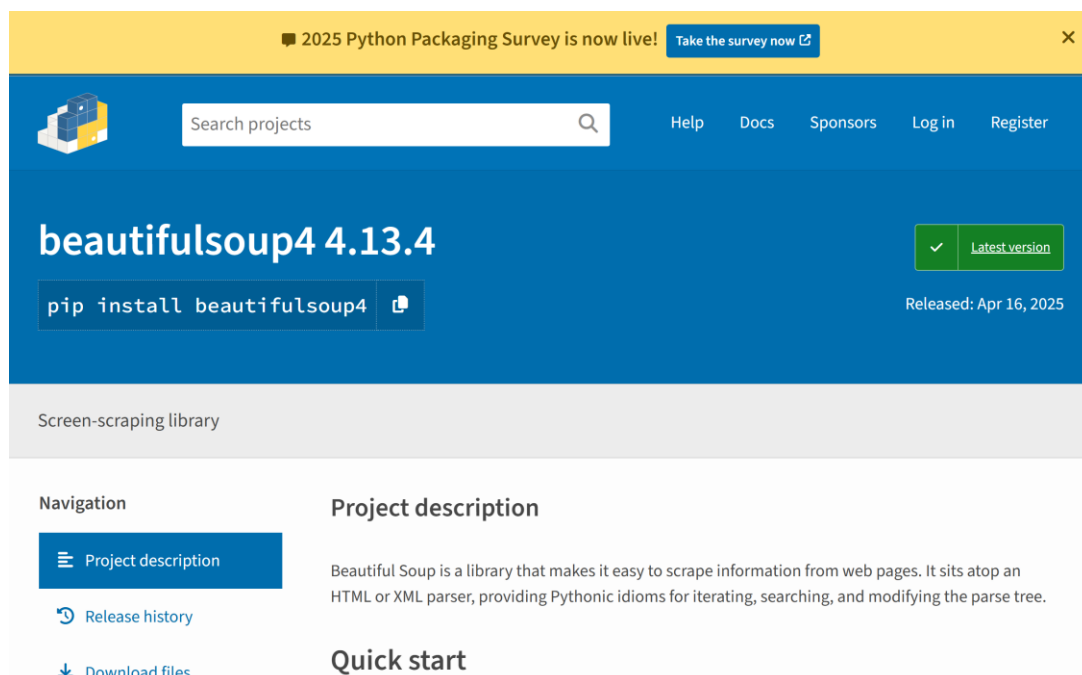
Kết quả:

```
Tải trang thành công!

Nội dung trang: <!doctype html><html><head><title>Example
Domain</title></head><body>...
```

2.4.3 Thư viện BeautifulSoup

BeautifulSoup là một thư viện phổ biến trong Python được sử dụng để phân tích cú pháp và trích xuất dữ liệu từ các tài liệu HTML hoặc XML. Thư viện này cho phép truy cập nhanh đến các thẻ HTML, thuộc tính, văn bản,... bằng cách sử dụng cú pháp đơn giản và thân thiện. Trong các công cụ kiểm tra mã HTML, BeautifulSoup đặc biệt hữu ích để trích xuất các phần tử, kiểm tra cấu trúc thẻ, và phát hiện các lỗi định dạng như thẻ chưa đóng hoặc lồng sai.



Hình 2.4 Thư viện BeautifulSoup

Ví dụ:

```
from bs4 import BeautifulSoup

html_code = "<html><body><div><p>Hello</div></body></html>"
soup = BeautifulSoup(html_code, 'html.parser')

# Tìm tất cả thẻ <div>
div_tags = soup.find_all('div')
print("Số lượng thẻ <div>:", len(div_tags))

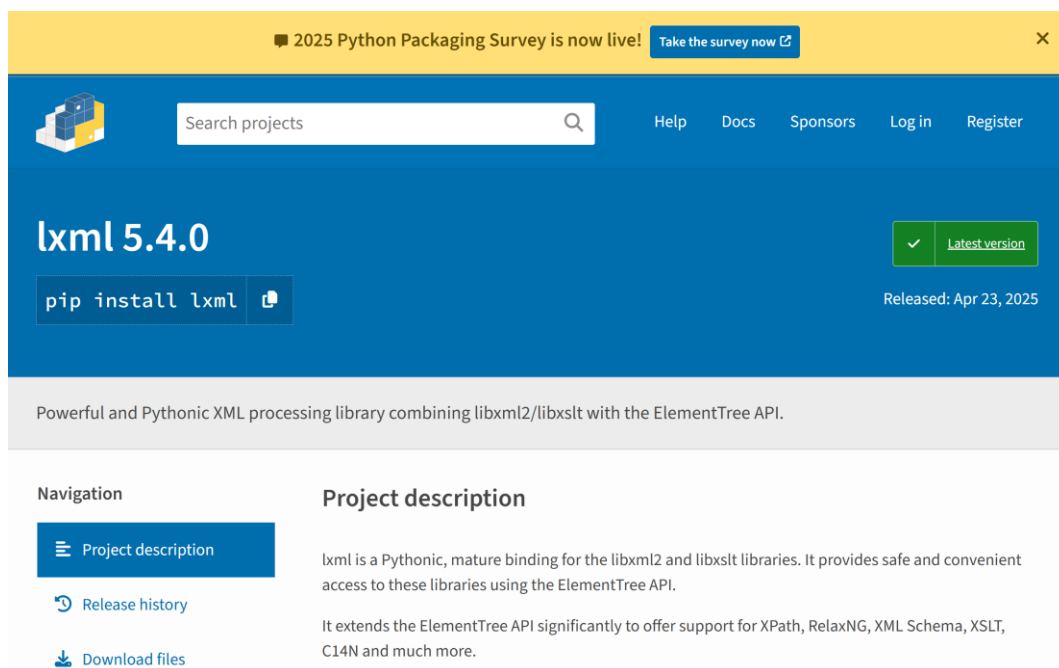
# In ra nội dung đầu tiên trong <p>
p_tag = soup.find('p')
print("Nội dung trong <p>:", p_tag.text)
```

Kết quả:

```
Số lượng thẻ <div>: 1
Nội dung trong <p>: Hello
```


2.4.4 Thư viện Lxml

Thư viện lxml là một thư viện mạnh mẽ trong Python được xây dựng trên nền tảng C, chuyên dùng để phân tích cú pháp và xử lý các tài liệu HTML và XML với hiệu năng cao. Khác với BeautifulSoup thiên về dễ dùng, lxml nổi bật với khả năng xử lý nhanh, hỗ trợ XPath và XSLT, phù hợp cho những hệ thống cần kiểm tra nhiều tệp HTML hoặc XML phức tạp. Trong công cụ đánh giá mã, lxml thường được dùng để kiểm tra tính đúng đắn của cấu trúc thẻ và truy vấn sâu bên trong tài liệu.



Hình 2.5 Thư viện Lxml

Ví dụ:

```
from lxml import etree

html_code = '''
<html>

  <body>

    <div class="note">Lưu ý 1</div>

    <div class="note">Lưu ý 2</div>

  </body>

</html>
'''

tree = etree.HTML(html_code)
notes = tree.xpath('//div[@class="note"]')

for n in notes:
    print("Ghi chú:", n.text)
```

Kết quả:

```
Ghi chú: Lưu ý 1
Ghi chú: Lưu ý 2
```

2.4.5 Thư viện re

Thư viện re là một thư viện tích hợp sẵn trong Python, chuyên dùng để xử lý các biểu thức chính quy (regular expressions), cho phép người dùng tìm kiếm, trích xuất và thay thế các mẫu chuỗi trong văn bản một cách linh hoạt. Trong bối cảnh kiểm tra mã HTML, CSS, JavaScript, thư viện này được ứng dụng để phát hiện lỗi như thẻ không đóng, sai chính tả thuộc tính (ví dụ viết nhầm class thành clas), hoặc trích xuất các thành phần như tên thẻ và thuộc tính. Ngoài ra, re còn có thể hỗ trợ sửa

lỗi đơn giản bằng cách thay thế chuỗi thông qua biểu thức chính quy. Với ưu điểm là được tích hợp sẵn trong Python, dễ tùy chỉnh và có tính linh hoạt cao, re trở thành công cụ hữu ích trong nhiều bài toán xử lý chuỗi. Tuy nhiên, thư viện này cũng có một số nhược điểm, đặc biệt là khi xử lý các mẫu phức tạp hoặc dữ liệu có cấu trúc sâu, dễ gây lỗi và khó kiểm soát nếu biểu thức không được viết rõ ràng.

Ví dụ:

```
import re

html_code = "<div clas='box'>Nội dung</div>"

# Phát hiện lỗi chính tả thuộc tính 'clas'
pattern = r"clas="
result = re.search(pattern, html_code)

if result:
    print("Phát hiện lỗi chính tả: 'clas' nên sửa thành 'class'")
    # Gợi ý sửa lỗi
    fixed_code = re.sub(pattern, "class=", html_code)
    print("Mã đã sửa:", fixed_code)
```

Kết quả:

```
Phát hiện lỗi chính tả: 'clas' nên sửa thành 'class'
Mã đã sửa: <div class='box'>Nội dung</div>
```

2.4.6 Thư viện os

Thư viện os là một thư viện chuẩn khác trong Python, cung cấp các chức năng quan trọng để tương tác với hệ điều hành, chẳng hạn như quản lý tệp và thư mục, thao tác với đường dẫn, và truy cập các biến môi trường. Trong các ứng dụng xử lý mã nguồn, os thường được sử dụng để truy cập thư mục chứa bài tập, kiểm tra sự tồn tại

của tệp, tạo thư mục lưu kết quả phân tích, hoặc xóa các tệp tạm sau khi xử lý xong. Ưu điểm của `os` là có khả năng đa nền tảng, hỗ trợ tốt cho việc cấu hình và tương tác hệ thống cơ bản, đồng thời không cần cài đặt bổ sung do đã được tích hợp sẵn trong Python. Tuy vậy, thư viện này cũng có những hạn chế nhất định như phụ thuộc vào hệ thống tệp cụ thể của từng nền tảng, và còn thiếu các tính năng nâng cao nếu không kết hợp thêm với các thư viện khác như `shutil` hay `pathlib`.

Ví dụ: Tạo thư mục và tạo file `log.txt` để lưu kết quả ghi file

```
import os

# Kiểm tra xem thư mục 'ketqua' đã tồn tại chưa
if not os.path.exists('ketqua'):

    os.mkdir('ketqua') # Tạo thư mục nếu chưa có

# Tạo một file lưu kết quả
with open(os.path.join('ketqua', 'log.txt'), 'w') as f:

    f.write("Kết quả kiểm tra bài tập HTML.\n")
```

2.5 Công nghệ và công cụ sử dụng

2.5.1 Công nghệ nền tảng

Bộ công cụ được xây dựng trên nền tảng web theo mô hình client-server, nhằm cung cấp giao diện thân thiện với người dùng cùng khả năng xử lý hiệu quả.

Flask: Là một framework nhẹ của Python, được sử dụng để xây dựng phần backend. Flask hỗ trợ tiếp nhận và xử lý yêu cầu từ người dùng, quản lý dữ liệu đầu vào (mã HTML, CSS, JavaScript), đồng thời tích hợp dễ dàng với các thư viện kiểm tra mã. Ưu điểm nổi bật của Flask là cấu hình đơn giản, dễ mở rộng, phù hợp với các dự án quy mô nhỏ và trung bình, đặc biệt tương thích tốt với các thư viện như `cssutils`, `BeautifulSoup`, và `re`.

HTML5, CSS3, JavaScript (ES6+): Đây là các công nghệ nền tảng cho phần giao diện người dùng (frontend). HTML5 định nghĩa cấu trúc trang, CSS3 đảm bảo

bố cục và hiển thị hiện đại, trong khi JavaScript (ES6+) giúp tạo ra các tính năng tương tác như kéo-thả tập tin, hiển thị lỗi theo dòng, hoặc cập nhật kết quả kiểm tra thời gian thực.

WebSocket (tùy chọn): Đối với các tình huống cần phản hồi nhanh hoặc xử lý tập mã lớn, WebSocket có thể được tích hợp để hỗ trợ truyền tải dữ liệu hai chiều theo thời gian thực giữa client và server, giúp giảm độ trễ so với giao thức HTTP truyền thống.

2.5.2 Ngôn ngữ lập trình

Python: Là ngôn ngữ chính của backend, được lựa chọn nhờ cú pháp đơn giản, khả năng phát triển nhanh và hệ sinh thái thư viện đa dạng như `cssutils`, `BeautifulSoup`, `lxml`, `re`, và `requests`. Python giúp xử lý logic kiểm tra mã một cách linh hoạt và dễ mở rộng.

JavaScript: Sử dụng cho phía frontend để xử lý tương tác người dùng. Phiên bản ES6+ được áp dụng nhằm tận dụng các tính năng hiện đại như arrow functions, promises, `async/await`, hỗ trợ lập trình bất đồng bộ và tối ưu hóa trải nghiệm người dùng.

SQL (SQLite): Hệ quản trị cơ sở dữ liệu nhẹ, dùng để lưu trữ tạm thời các kết quả kiểm tra, lịch sử thao tác người dùng. SQLite phù hợp cho các ứng dụng nhỏ, dễ tích hợp với Python thông qua module `sqlite3` mà không cần cài đặt thêm máy chủ cơ sở dữ liệu.

2.5.3 Công cụ hỗ trợ phát triển

Git & GitHub: Hệ thống quản lý phiên bản giúp theo dõi thay đổi mã nguồn, hỗ trợ làm việc nhóm hiệu quả và triển khai liên tục (CI/CD) thông qua các workflow tự động.

Postman: Dùng để kiểm thử các API nội bộ giữa frontend và backend, giúp đảm bảo các thao tác như gửi mã nguồn, nhận kết quả, lưu lịch sử đều được thực thi chính xác.

Git và GitHub: Sử dụng để quản lý phiên bản mã nguồn, cho phép theo dõi thay đổi, hợp tác phát triển, và lưu trữ dự án một cách an toàn. GitHub cũng hỗ trợ tích hợp CI/CD để tự động kiểm tra mã trước khi triển khai.

2.6 Cơ sở dữ liệu

Cơ sở dữ liệu đóng vai trò quan trọng trong việc lưu trữ, quản lý, và truy xuất dữ liệu của hệ thống kiểm tra mã nguồn. Trong đề tài, cơ sở dữ liệu được thiết kế để hỗ trợ các chức năng như lưu trữ thông tin người dùng, bài tập, lớp học, và kết quả kiểm tra mã nguồn.

SQLite một hệ quản trị cơ sở dữ liệu nhẹ, không cần máy chủ riêng, phù hợp với các ứng dụng web quy mô nhỏ. SQLite tích hợp tốt với Python thông qua module `sqlite3`, giúp lưu trữ dữ liệu một cách hiệu quả.

2.7 Kiến trúc hệ thống

Kiến trúc được thiết kế theo mô hình client-server

Lớp client (Frontend)

Giao diện người dùng được xây dựng bằng ngôn ngữ HTML, CSS, JAVASCRIPT. Cho phép nhập mã nguồn, gửi yêu cầu, kiểm tra, hiển thị lỗi và gợi ý sửa lỗi.

Lớp Server (Backend)

Phát triển bằng Flask (Python), xử lý các quy trình kiểm tra, xác thực người dùng.

Lớp dữ liệu

Dùng Sqlite để lưu trữ người dùng, lớp học, bài tập, và kết quả kiểm tra

Lớp xử lý mã nguồn

Gồm các công cụ phân tích như HTMLHint, Stylelint, ESLint, với các thư viện Python (BeautifulSoup, lxml..) để kiểm tra và chuẩn hóa lỗi

2.8 Nhận xét và định hướng phát triển công cụ

Trong quá trình xây dựng công cụ kiểm tra tự động chất lượng mã HTML, CSS và JavaScript, tôi nhận thấy có nhiều vấn đề kỹ thuật và triển khai phức tạp cần được xử lý một cách có hệ thống. Các thách thức không chỉ đến từ sự đa dạng và đặc thù của lỗi trong từng ngôn ngữ mà còn liên quan đến khả năng tích hợp nhiều công cụ khác nhau, đảm bảo hiệu suất xử lý khi kiểm tra khối lượng lớn mã, và cung cấp phản hồi hữu ích, chính xác cho người dùng. Từ đó, tôi rút ra được hai nhóm nội dung chính: các khó khăn, thách thức cần vượt qua, và những phương pháp tôi đã lựa chọn để phát triển và tích hợp công cụ một cách hiệu quả.

2.8.1 Những khó khăn và thách thức

Một trong những khó khăn lớn đầu tiên tôi gặp phải là sự đa dạng và đặc thù của lỗi trong từng ngôn ngữ. HTML thường mắc lỗi thẻ không đóng, thẻ lồng sai hoặc thuộc tính không hợp lệ; CSS lại dễ bị lỗi định dạng, sai giá trị hoặc không tuân thủ quy tắc đặt tên; trong khi JavaScript thường gặp lỗi logic như gán nhầm, gọi hàm không tồn tại, hoặc sai lệch trong xử lý bất đồng bộ. Việc phân loại và xây dựng tiêu chí kiểm tra cho từng loại lỗi đòi hỏi sự nghiên cứu kỹ lưỡng về chuẩn ngôn ngữ và thực tiễn lập trình.

Vấn đề hiệu suất cũng là một trở ngại đáng kể. Khi công cụ cần kiểm tra nhiều bài làm sinh viên hoặc mã nguồn từ các dự án lớn, thời gian xử lý trở thành yếu tố quan trọng. Tôi cần lựa chọn thuật toán và cách xử lý dữ liệu đủ nhẹ, nhanh nhưng vẫn đảm bảo độ chính xác và khả năng phản hồi tức thời cho người dùng.

Việc tích hợp nhiều công cụ kiểm tra mã như ESLint, Stylelint, HTMLHint, hay W3C Validator vào cùng một hệ thống kiểm tra cũng phát sinh nhiều khó khăn. Các công cụ này yêu cầu cấu hình môi trường khác nhau (ví dụ: Node.js và các gói phụ thuộc), định dạng đầu ra không đồng nhất, và đôi khi cho kết quả đánh giá mâu thuẫn – một công cụ đánh giá là lỗi, công cụ khác chỉ cảnh báo. Tôi phải xây dựng một lớp trung gian để xử lý, đồng bộ, và chuẩn hóa kết quả nhằm đưa ra phản hồi rõ ràng và nhất quán cho người dùng cuối.

Một thách thức quan trọng khác là việc đưa ra gợi ý sửa lỗi và sửa lỗi tự động. Đối với các lỗi cú pháp đơn giản, việc đề xuất hoặc sửa tự động có thể thực hiện khá hiệu quả. Tuy nhiên, với những lỗi phức tạp hơn – đặc biệt là lỗi logic trong JavaScript hoặc lỗi trình bày nâng cao trong CSS – tôi nhận thấy rất khó để đưa ra hướng sửa hợp lý nếu thiếu ngữ cảnh đầy đủ. Các phương pháp phát triển công cụ.

2.8.2 Phương pháp phát triển và tích hợp công cụ

Để giải quyết các vấn đề nêu trên, tôi đã xây dựng hệ thống kiểm tra dựa trên các tập hợp quy tắc kiểm tra riêng biệt cho từng ngôn ngữ, đồng thời tuân thủ theo các tiêu chuẩn quốc tế và các khuyến nghị của cộng đồng phát triển. Cụ thể, tôi áp dụng tiêu chuẩn W3C cho HTML để đảm bảo khả năng tương thích và truy cập tốt; sử dụng đặc tả CSS3/CSS4 để kiểm tra thuộc tính, thứ tự và định dạng trong CSS; và tuân thủ chuẩn ECMAScript 6 trở lên cho việc kiểm tra mã JavaScript, nhằm phát hiện lỗi hiện đại và đảm bảo tính cập nhật.

Bên cạnh đó, tôi sử dụng các công cụ kiểm tra mã nguồn phổ biến hiện có như HTMLHint, Stylelint, ESLint và W3C Validator. Việc tích hợp các công cụ này được thực hiện thông qua hai cơ chế: gọi thực thi trực tiếp thông qua subprocess trong Python với các công cụ cài đặt cục bộ, và sử dụng API trực tuyến (như W3C Validator) để kiểm tra mã không cần cài đặt thư viện. Tôi xây dựng thành phần trung gian để chuẩn hóa đầu ra, gom kết quả, và hiển thị theo định dạng thống nhất, giúp người dùng dễ hiểu và giảng viên dễ theo dõi lỗi.

Việc tích hợp đa công cụ mang lại tính chính xác cao hơn trong kiểm tra, đồng thời tiết kiệm thời gian phát triển khi có thể tận dụng các bộ kiểm tra mã đã được cộng đồng phát triển mạnh mẽ. Tuy nhiên, để hệ thống hoạt động hiệu quả, tôi phải đảm bảo môi trường đồng nhất, xử lý tốt các lỗi xung đột giữa các công cụ, và dự kiến sẽ tiếp tục mở rộng khả năng cấu hình quy tắc kiểm tra, bổ sung tính năng sửa lỗi tự động thông minh hơn và cải thiện hiệu suất trong các lần phát triển tiếp theo.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả đề tài

Đề tài “Xây dựng công cụ đánh giá chất lượng mã nguồn HTML, CSS, JAVASCRIPT tự động” hướng đến việc thiết kế và phát triển một hệ thống web hỗ trợ kiểm tra và đánh giá chất lượng mã nguồn ba ngôn ngữ phổ biến trong phát triển giao diện web, bao gồm HTML, CSS và JavaScript. Hệ thống được thiết kế hướng đến môi trường giáo dục, đặc biệt là phục vụ giảng viên và sinh viên trong việc học tập, giảng dạy và thực hành lập trình web.

Mục tiêu chính của hệ thống là phát hiện lỗi mã nguồn, kiểm tra tuân thủ tiêu chuẩn lập trình, đồng thời đưa ra gợi ý và sửa lỗi tự động, góp phần nâng cao chất lượng và hiệu quả học tập – giảng dạy trong lĩnh vực phát triển web.

3.2 Đặc tả yêu cầu hệ thống

3.2.1 Yêu cầu chức năng

<i>STT</i>	<i>Chức năng</i>	<i>Quản trị viên</i>	<i>Giảng viên</i>	<i>Sinh viên</i>
1	Quản lý người dùng giảng viên (Thêm, xóa,sửa)	✓		
2	Quản lý người dùng sinh viên (Thêm, xóa,sửa)	✓	✓	
3	Đăng nhập, Đăng ký	✓	✓	
4	Quản lý lớp học (Thêm, xóa,sửa)	✓	✓	
5	Quản lý bài tập (Thêm, xóa,sửa)	✓	✓	
6	Kiểm tra mã	✓	✓	✓
7	Xem và thay đổi thông tin cá nhân	✓	✓	✓
8	Tham gia lớp học			✓
9	Làm bài, nộp bài, chỉnh sửa bài tập			✓

3.2.2 Yêu cầu phi chức năng

<i>STT</i>	<i>Chức năng</i>	<i>Quản trị viên</i>	<i>Giảng viên</i>	<i>Sinh viên</i>
1	Giao diện thân thiện	✓	✓	✓
2	Truy xuất dữ liệu ổn định	✓	✓	✓
3	Hệ thống bảo mật tốt	✓	✓	✓

3.3. Chức năng của người dùng

3.3.1 Quyền quản trị

- Quản lý tài khoản người dùng toàn hệ thống (sửa, xóa, gán quyền).
- Xem thống kê hệ thống: số lượng người dùng, lượt kiểm tra mã, số bài tập giao...
- Quản lý các lớp học và theo dõi hoạt động

3.3.2 Quyền giảng viên

- Tạo và quản lý các lớp học mà mình phụ trách
- Giao bài tập cho từng lớp, quy định thời hạn, nội dung mã cần viết.
- Xem bài làm của sinh viên, lịch sử sửa lỗi, kết quả tự động đánh giá và điểm.

3.3.3 Quyền sinh viên

- Tham gia lớp học bằng mã lớp hoặc qua duyệt của giảng viên.
- Làm bài tập trực tiếp trên giao diện web có tích hợp Monaco Editor.
- Kiểm tra mã nguồn HTML, CSS, JS tự động với phản hồi lỗi và gợi ý sửa.
- Gửi bài đúng hạn và xem phản hồi đánh giá từ hệ thống/giảng viên.

3.3 Thiết kế dữ liệu

3.3.1 Mô hình cơ sở dữ liệu



Hình 3.1 Mô hình cơ sở dữ liệu

3.3.2 Chi tiết các thực thể

3.3.2.1 Thực thể nguoi_dung

Mô tả: Lưu trữ thông tin của người dùng bao gồm mã người dùng, tên đăng nhập, mật khẩu, vai trò, trạng thái tài khoản, thời gian tạo, ảnh đại diện, email.

Bảng 3.1 Chi tiết thực thể nguoi_dung

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
ma_nguoi_dung	Định danh duy nhất cho mỗi người dùng	INTEGER	Khóa chính
ten_dang_nhap	Tên đăng nhập của người dùng	TEXT	Khóa ngoại
mat_khau	Mật khẩu của người dùng	TEXT	
vai_tro	Vai trò của người dùng	TEXT	
trang_thai	Trạng thái tài khoản	TEXT	

<i>Tên thuộc tính</i>	<i>Diễn giải</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
thoi_gian_tao	Thời gian tạo tài khoản	DATETIME	
anh_dai_dien	Đường dẫn ảnh đại diện	TEXT	
email	Địa chỉ email của người dùng	TEXT	

3.3.2.2 Thực thể lop_hoc

Mô tả: Bảng lớp học lưu trữ thông tin về các lớp học, bao gồm mã lớp, mã giảng viên phụ trách, tên lớp, mã số lớp, thời gian tạo, trạng thái, ngày bắt đầu, ngày kết thúc, và mô tả.

Bảng 3.2 Bảng chi tiết thực thể lop_hoc

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_lop	Định danh duy nhất cho lớp học	INTEGER	Khóa chính
ma_giang_vien	Mã giảng viên phụ trách lớp	INTEGER	Khóa ngoại
ten_lop	Tên của lớp học	TEXT	
ma_so_lop	Mã số lớp	TEXT	
thoi_gian_tao	Thời gian tạo lớp học	DATETIME	
trang_thai	Trạng thái lớp học	TEXT	
ngay_bat_dau	Ngày bắt đầu lớp	DATE	
ngay_ket_thuc	Ngày kết thúc lớp	DATE	
mo_ta	Mô tả lớp học	TEXT	

3.3.2.3 Thực thể sinh_vien_lop_hoc

Mô tả: Bảng sinh viên lớp học lưu trữ thông tin của người dùng đã tham gia lớp học bao gồm mã sinh viên lớp học, mã lớp, mã sinh viên, ngày đăng ký, và trạng thái tham gia.

Bảng 3.3 Bảng chi tiết thực thể sinh_vien_lop_hoc

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_sinh_vien_lop_hoc	ID duy nhất cho mỗi bản ghi	INTEGER	Khóa chính
ma_lop	Mã lớp học	INTEGER	Khóa ngoại
ma_sinh_vien	Mã sinh viên	TEXT	Khóa ngoại
ngay_dang_ky	Ngày sinh viên đăng ký lớp	DATETIME	
trang_thai	Trạng thái tham gia lớp	TEXT	

3.3.2.4 Thực thể bai_tap

Mô tả: Bảng bài tập lưu trữ thông tin của bài tập bao gồm mã bài tập, mã giảng viên, tiêu đề, nội dung, hạn nộp, ngày bắt đầu, ngày kết thúc, thời gian tạo, mô tả, ngôn ngữ, tập tin bài tập, và trạng thái.

Bảng 3.4 Bảng chi tiết thực thể bai_tap

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_bai_tap	Định danh duy nhất cho mỗi bài tập	INTEGER	Khóa chính
ma_giang_vien	Mã giảng viên tạo bài	INTEGER	Khóa ngoại
tieu_de	Tiêu đề của bài tập	TEXT	
noi_dung	Nội dung của bài tập	TEXT	
han_nop	Hạn nộp bài tập	DATETIME	
ngay_bat_dau	Ngày bắt đầu nộp	DATE	
ngay_ket_thuc	Ngày kết thúc nộp	DATE	

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
thoi_gian_tao	Thời gian tạo bài	DATETIME	
mo_ta	Mô tả bài tập	TEXT	
ngon_ngu	Ngôn ngữ kiểm tra	TEXT	
tap_tin_bai_tap	Tệp bài tập đính kèm	TEXT	
trang_thai	Trạng thái bài	TEXT	

3.3.2.5 Thực thể bai_tap_lop_hoc

Mô tả: Bài tập lớp học lưu trữ thông tin của bài tập có trong lớp học bao gồm mã bài tập, mã lớp, ngày gán bài tập, và trạng thái bài tập trong lớp

Bảng 3.5 Bảng chi tiết thực thể bai_tap_lop_hoc

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_bai_tap	Mã bài tập	INTEGER	Khóa ngoại
ma_lop	Mã lớp học	INTEGER	Khóa ngoại
ngay_gan	Ngày gán bài cho lớp	DATETIME	
trang_thai	Trạng thái bài tập trong lớp	TEXT	

3.3.2.6 Thực thể bai_nop

Mô tả: Bảng bài nộp lưu trữ thông tin của bài tập mà sinh viên đã nộp bao gồm mã bài nộp, mã bài tập, mã sinh viên nộp bài, nội dung nộp, tệp tin nộp, thời gian nộp, trạng thái bài nộp, và số lần nộp.

Bảng 3.6 Bảng chi tiết thực thể bai_nop

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_bai_nop	Mã bài nộp	INTEGER	Khóa chính
ma_bai_tap	Mã bài tập liên quan	INTEGER	Khóa ngoại
ma_sinh_vien	Mã sinh viên nộp bài	INTEGER	Khóa ngoại

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
noi_dung_nop	Nội dung nộp	TEXT	
tap_tin_nop	File đính kèm	TEXT	
thoi_gian_nop	Thời điểm nộp	DATETIME	
trang_thai	Trạng thái bài nộp	TEXT	
lan_nop	Số lần nộp	INTEGER	

3.3.2.7 Thực thể người_dung

Mô tả: Bảng kiểm tra mã lưu trữ thông tin của mã đã kiểm tra bao gồm mã kiểm tra, mã sinh viên thực hiện kiểm tra, mã bài tập được kiểm tra, nội dung mã nguồn kiểm tra, ngôn ngữ, lỗi, và thời gian kiểm tra

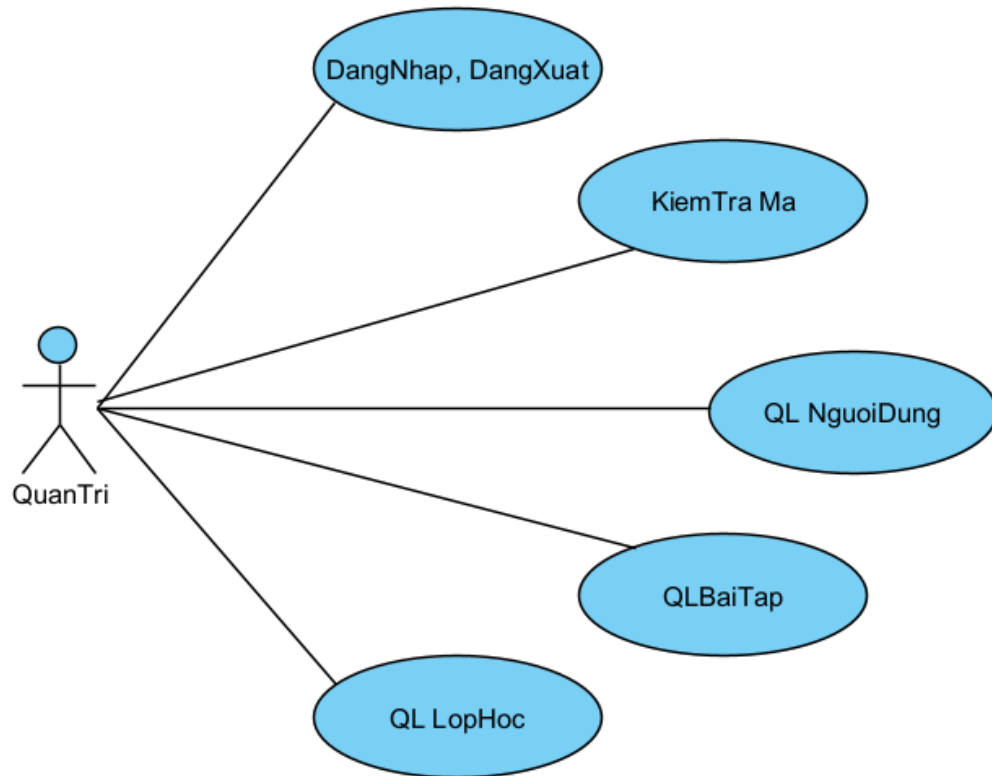
Bảng 3.7 Bảng chi tiết thực thể kiểm_tra_ma

<i>Tên thuộc tính</i>	<i>Giải thích</i>	<i>Kiểu dữ liệu</i>	<i>Ràng buộc</i>
ma_kiem_tra	Mã kiểm tra	INTEGER	Khóa chính
ma_sinh_vien	Mã sinh viên thực hiện kiểm tra	INTEGER	Khóa ngoại
ma_bai_tap	Mã bài tập được kiểm tra	INTEGER	Khóa ngoại
ma_nguon	Nội dung mã nguồn kiểm tra	TEXT	
ngon_ngu	Ngôn ngữ kiểm tra	TEXT	
loi	Các lỗi được phát hiện	TEXT	
thoi_gian_kiem_tra	Thời gian kiểm tra	DATETIME	

3.4 Thiết kế xử lý

3.4.1 Mô hình Use Case

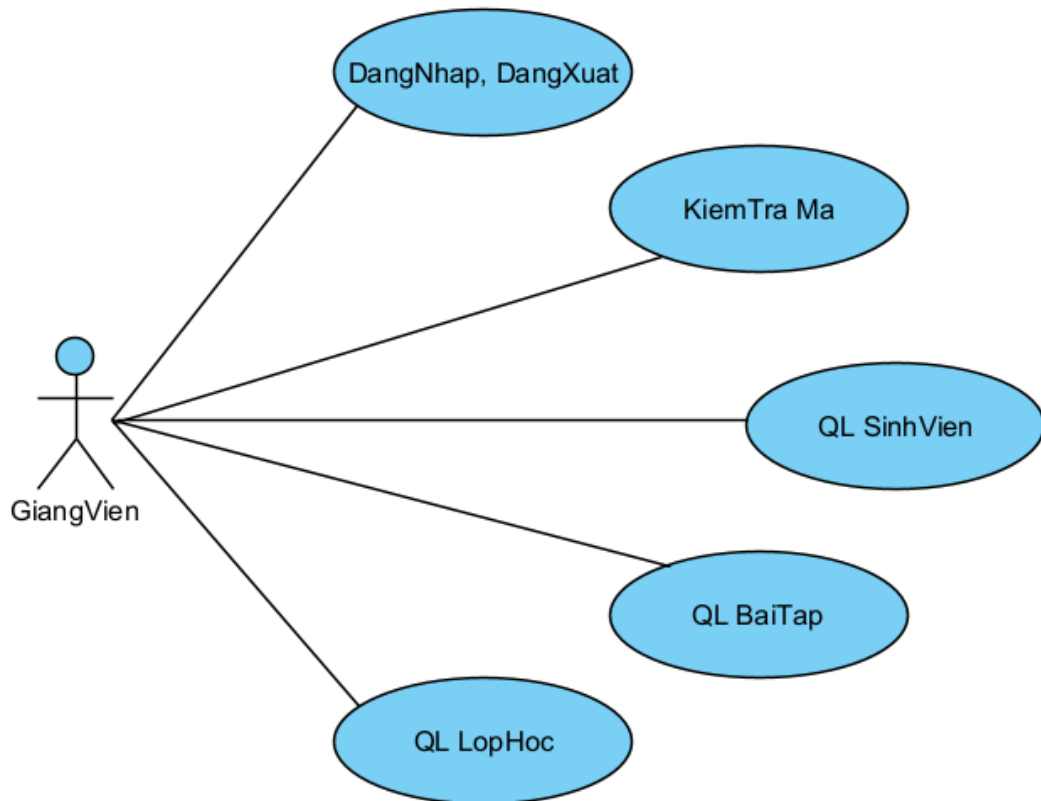
3.4.1.1 Mô hình Use Case quyền quản trị



Hình 3.2 Mô hình Use Case quản trị

Mô tả: Mô hình Use Case quản trị thể hiện quản trị có các chức năng quản lý người dùng, quản lý bài tập, quản lý lớp học, đăng nhập và đăng xuất hệ thống

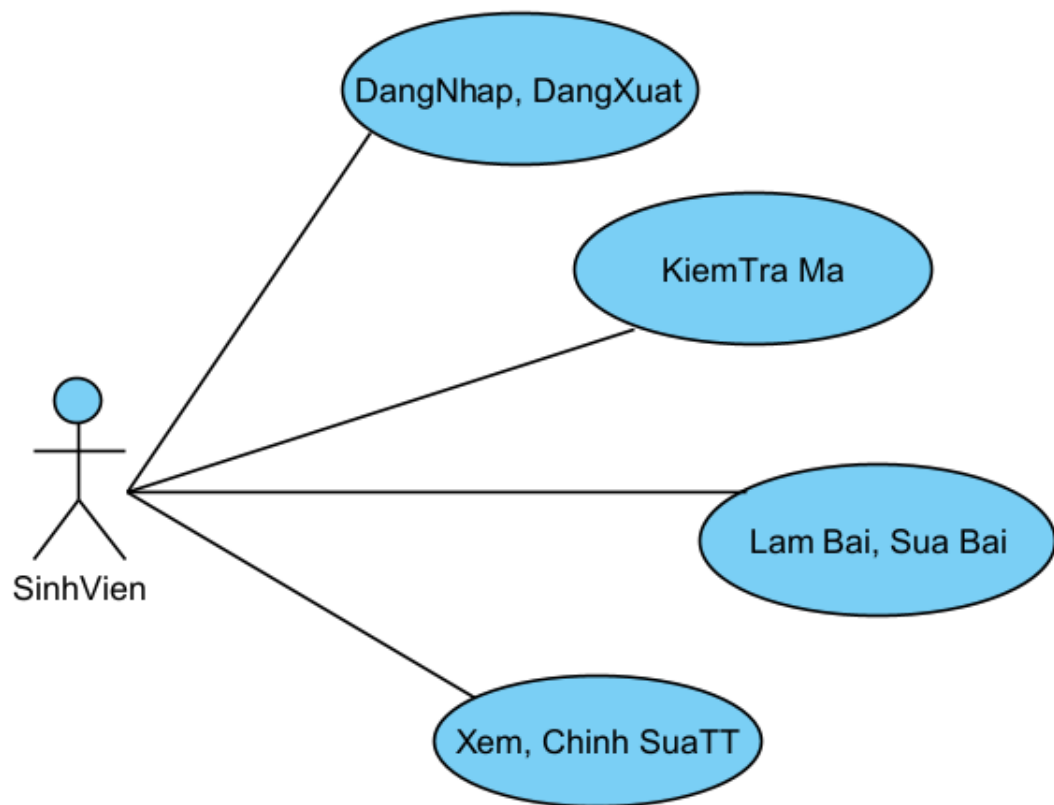
3.4.1.2 Mô hình Use Case quyền giảng viên



Hình 3.3 Mô hình Use Case giảng viên

Mô tả: Mô hình Use Case giảng viên thể hiện giảng viên có các chức năng quản lý sinh viên, quản lý bài tập, quản lý lớp học, đăng nhập và đăng xuất hệ thống

3.4.1.3 Mô hình Use Case quyền giảng viên

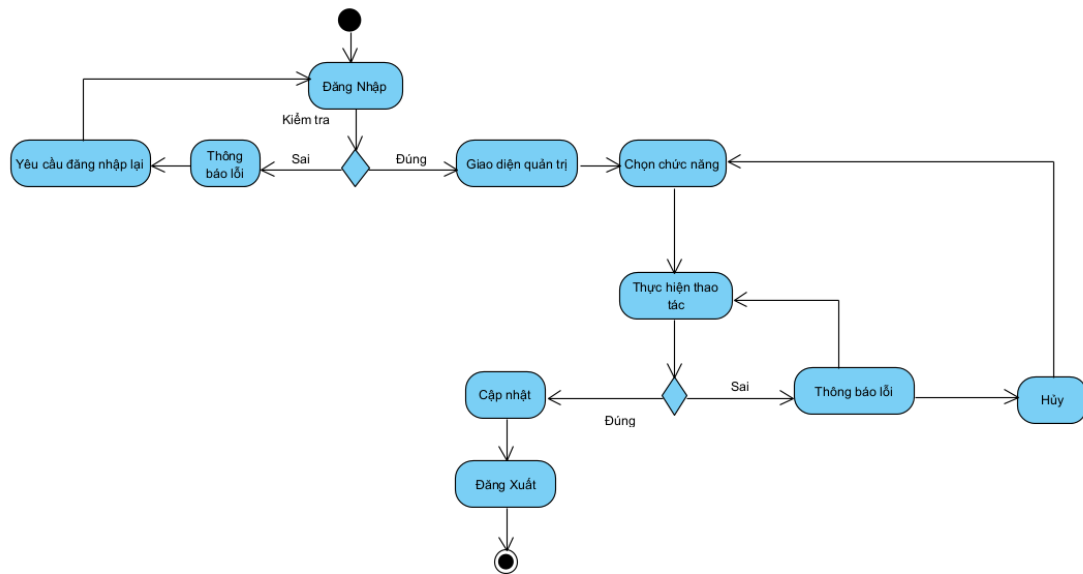


Hình 3.4 Mô hình Use Case sinh viên

Mô tả: Mô hình Use Case sinh viên thể hiện sinh viên có các chức năng đăng nhập đăng xuất hệ thống, kiểm tra mã, làm bài, sửa bài, xem và chỉnh sửa thông tin cá nhân.

3.4.2 Sơ đồ hoạt động

3.4.2.1 Sơ đồ hoạt động quyền quản trị



Hình 3.5 Sơ đồ hoạt động quản trị

Mô tả:

- Quản trị chọn đăng nhập
- Hệ thống yêu cầu nhập thông tin đăng nhập (Tên đăng nhập, mật khẩu)

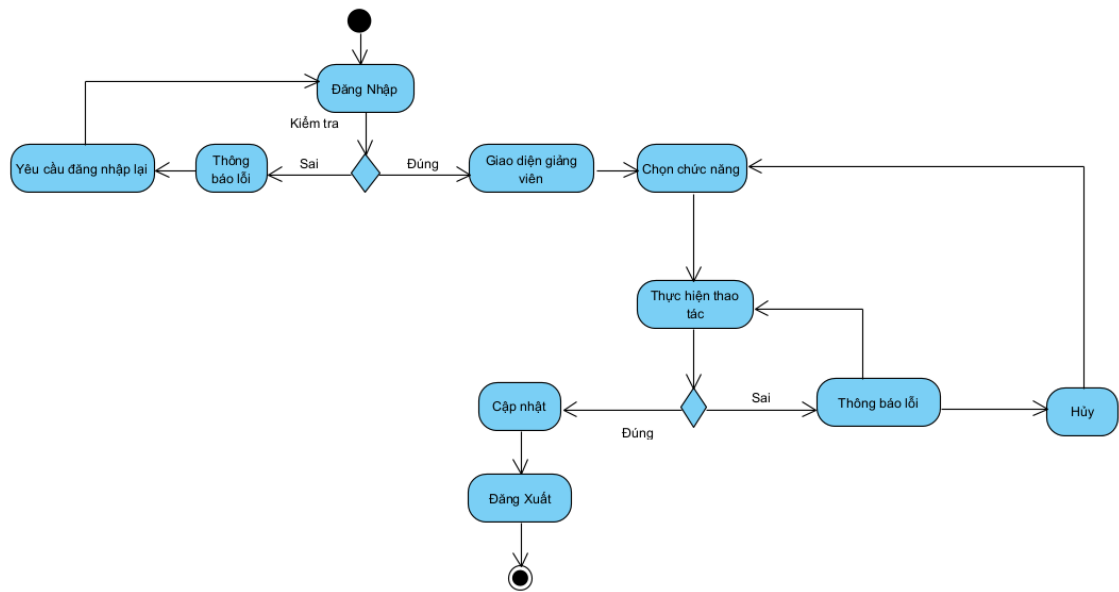
Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu đăng nhập lại

Nếu đúng: Chuyển sang giao diện quản trị

• Sau khi đăng nhập thành công hệ thống sẽ hiển thị giao diện trang quản trị. Quản trị có thể chọn các chức năng (quản lý lớp, quản lý bài tập, quản lý người dùng..)

• Quản trị thực hiện các thao tác tương ứng với chức năng đã chọn. Hệ thống sẽ kiểm tra nếu thành công sẽ cập nhật vào sơ sở dữ liệu, nếu sai sẽ thông báo lỗi và kết thúc

3.4.2.2 Sơ đồ hoạt động quyền quản trị



Hình 3.6 Sơ đồ hoạt động giảng viên

Mô tả:

- Quản trị chọn đăng nhập
- Hệ thống yêu cầu nhập thông tin đăng nhập (Tên đăng nhập, mật khẩu)

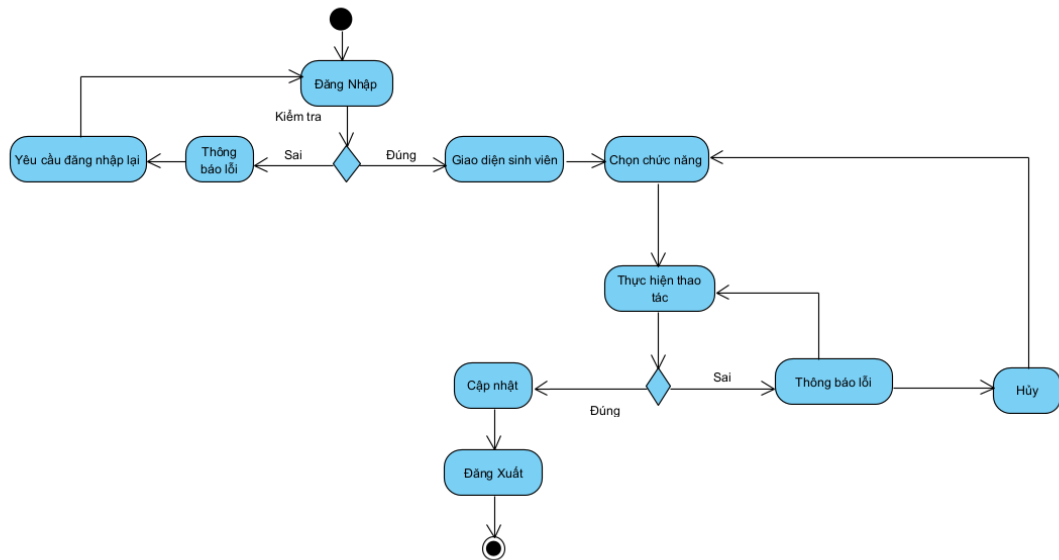
Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu đăng nhập lại

Nếu đúng: Chuyển sang giao diện quản trị

• Sau khi đăng nhập thành công hệ thống sẽ hiển thị giao diện trang quản trị. Quản trị có thể chọn các chức năng (quản lý lớp, quản lý bài tập, quản lý người dùng..)

• Quản trị thực hiện các thao tác tương ứng với chức năng đã chọn. Hệ thống sẽ kiểm tra nếu thành công sẽ cập nhật vào sơ sở dữ liệu, nếu sai sẽ thông báo lỗi và kết thúc

3.4.2.3 Sơ đồ hoạt động quyền sinh viên



Hình 3.7 Mô hình hoạt động quyền sinh viên

Mô tả:

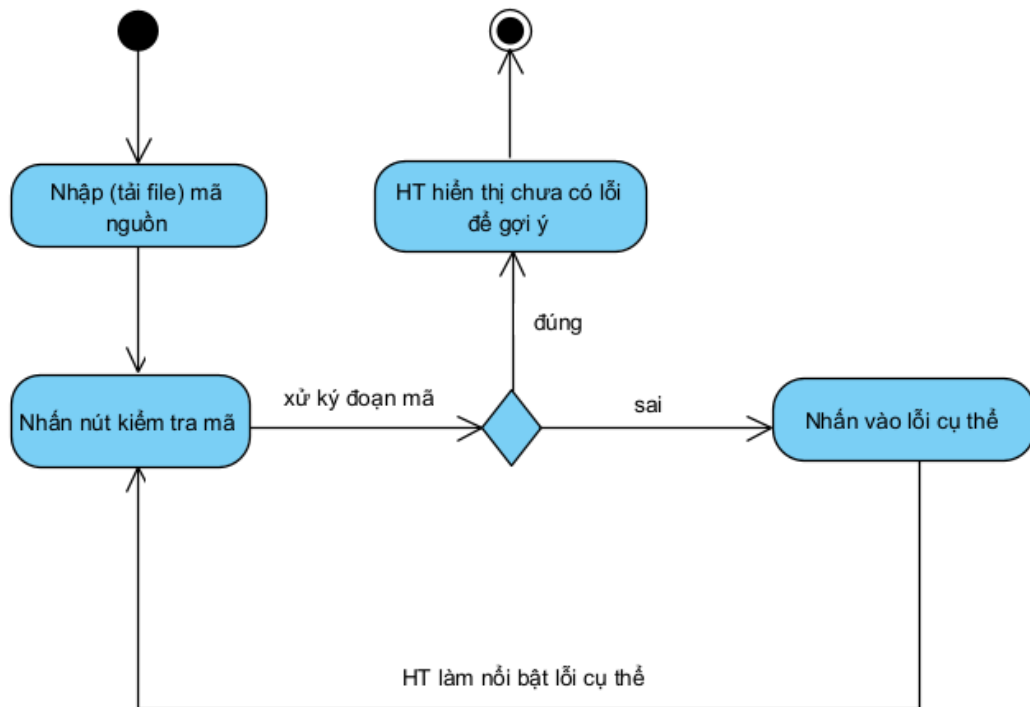
- Quản trị chọn đăng nhập
- Hệ thống yêu cầu nhập thông tin đăng nhập (Tên đăng nhập, mật khẩu)

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu đăng nhập lại

Nếu đúng: Chuyển sang giao diện quản trị

- Sau khi đăng nhập thành công hệ thống sẽ hiển thị giao diện trang quản trị. Quản trị có thể chọn các chức năng (quản lý lớp, quản lý bài tập, quản lý người dùng..)
- Quản trị thực hiện các thao tác tương ứng với chức năng đã chọn. Hệ thống sẽ kiểm tra nếu thành công sẽ cập nhật vào cơ sở dữ liệu, nếu sai sẽ thông báo lỗi và kết thúc

3.4.2.4 Sơ đồ hoạt động kiểm tra mã HTML, CSS, JAVASCRIPT



Hình 3.8 Sơ đồ hoạt động kiểm tra mã HTML, CSS, JAVASCRIPT

Mô tả:

- Người dùng nhập mã (Tải file) vào giao diện IDE tương ứng với mã nguồn
- Người dùng nhấn nút “Kiểm tra mã” để gửi yêu cầu kiểm tra

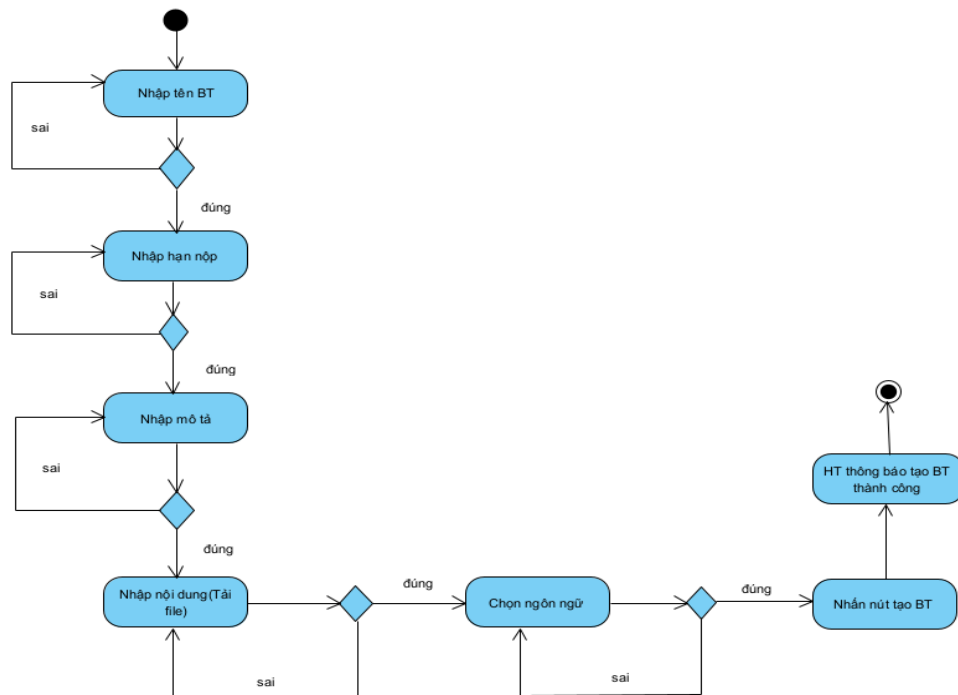
Nếu sai: Hệ thống sẽ thông báo lỗi và đưa ra gợi ý sửa lỗi

Nếu đúng: Hệ thống sẽ thông báo chưa có lỗi để gợi ý

• Người dùng có thể nhấn vào lỗi cụ thể để xem dòng lỗi đó nằm ở vị trí nào trong giao diện IDE, hệ thống sẽ làm nổi bật dòng lỗi cho người dùng để quan sát.

- Kết thúc

3.4.2.5 Sơ đồ hoạt động tạo bài tập



Hình 3.9 Sơ đồ hoạt động tạo bài tập

Mô tả: Chức năng tạo bài tập dành cho quản trị và giảng viên

- Quản trị hay giảng viên nhập tên bài tập

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại tên bài tập

- Nhập hạn nộp bài tập

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại hạn nộp bài tập

- Nhập mô tả bài tập (Có thể bỏ qua)

- Nhập nội dung bài tập hoặc tải file bài tập

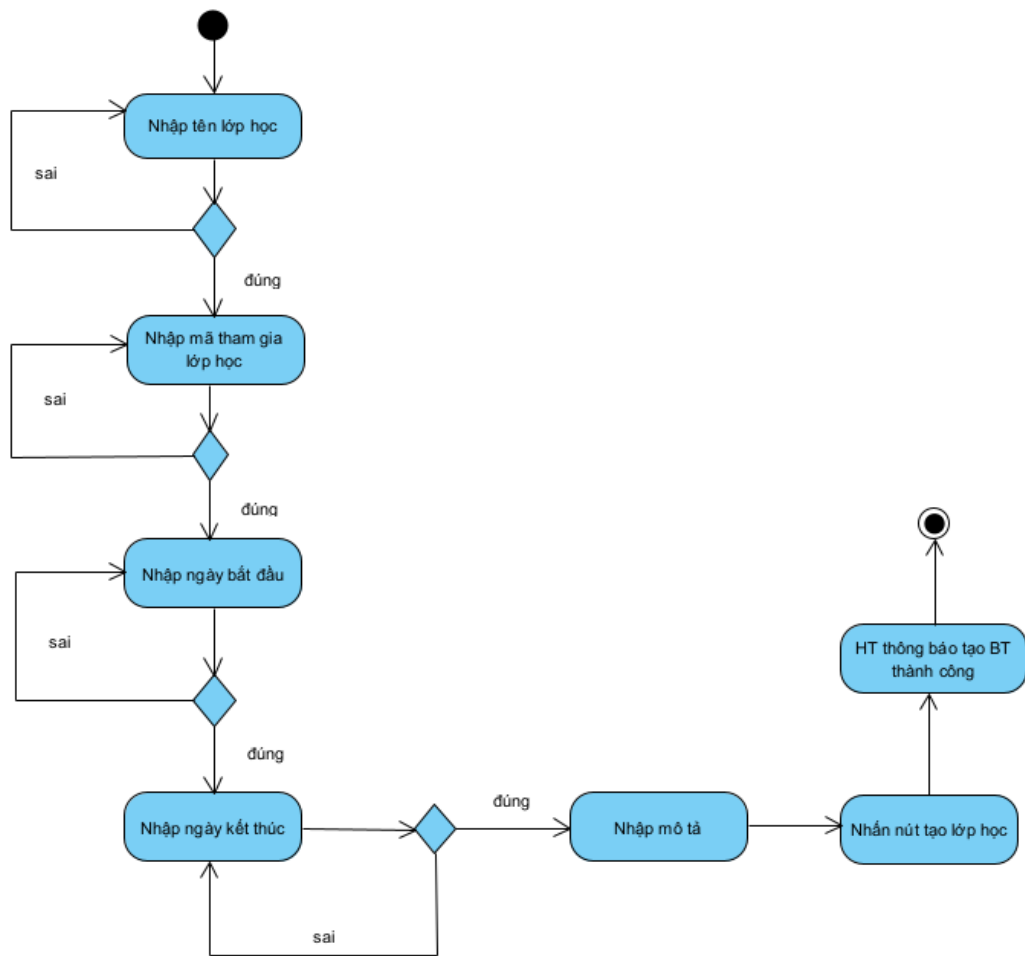
Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại tên bài tập hoặc tải file đúng định dạng

- Chọn ngôn ngữ kiểm tra (HTML, CSS, JAVASCRIPT)

- Nhấn nút tạo bài tập

- Kết thúc

3.4.2.6 Sơ đồ hoạt động tạo lớp học



Hình 3.10 Sơ đồ hoạt động tạo lớp học

Mô tả: Chức năng tạo bài tập dành cho quản trị và giảng viên

- Quản trị hay giảng viên nhập tên lớp học

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại tên lớp học

- Nhập mã tham gia lớp học

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại mã tham gia lớp học

- Nhập ngày bắt đầu lớp học

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại ngày bắt đầu lớp học

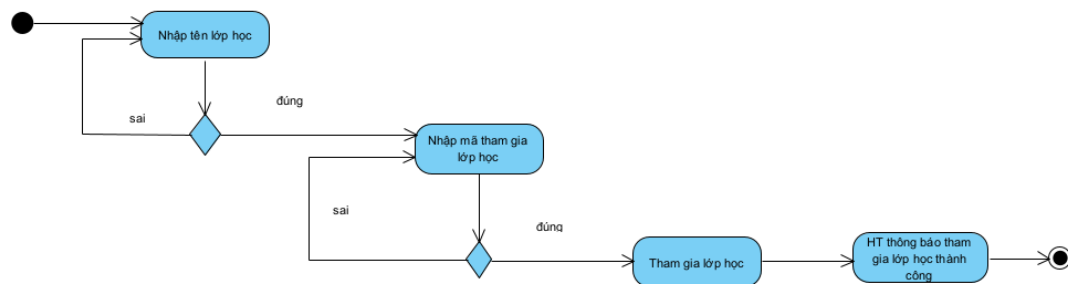
- Nhập ngày kết thúc lớp học

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại ngày kết thúc lớp học

- Nhập mô tả bài tập (Có thể bỏ qua)

- Kết thúc

3.4.2.7 Sơ đồ hoạt động tham gia lớp học



Hình 3.11 Sơ đồ hoạt động tham gia lớp học

Mô tả: Chức năng tham gia lớp học dành cho sinh viên

- Sinh viên nhập tên lớp học cần tham gia

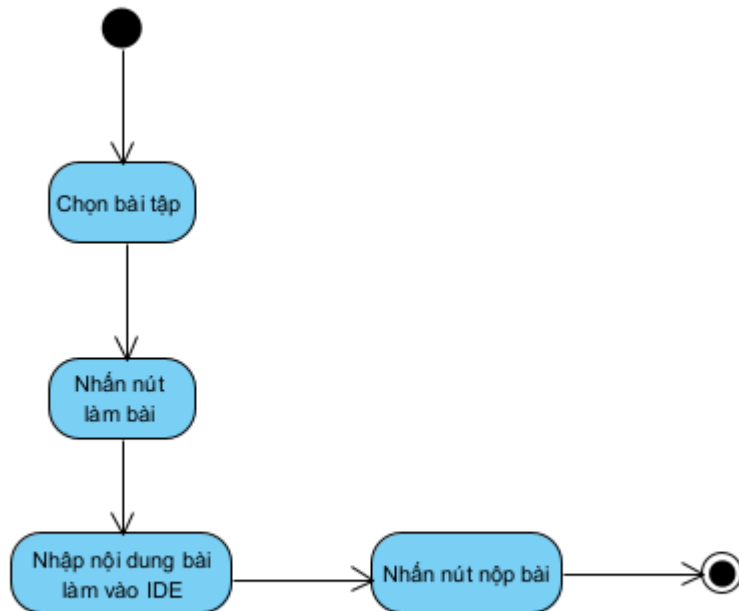
Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại tên lớp học

- Nhập mã tham gia lớp học

Nếu sai: Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại mã tham gia lớp học

-
- Nhấn nút tham gia lớp học, tham gia lớp học thành công
 - Kết thúc

3.4.2.8 Sơ đồ hoạt động nộp bài tập



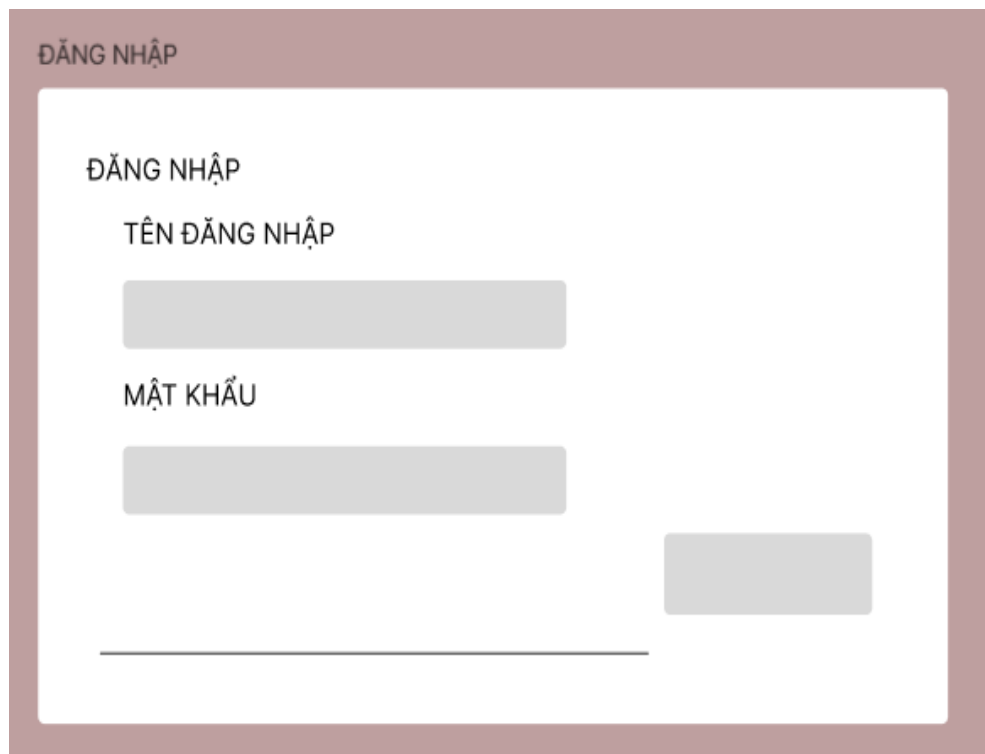
Hình 3.12 Sơ đồ hoạt động nộp bài tập

Mô tả: Chức nộp bài tập dành cho sinh viên

- Sinh viên chọn bài tập cần làm và nhấn nút làm bài
- Nhập nội dung bài làm vào giao diện IDE
- Nhấn nút nộp bài, nộp bài tập thành công
- Kết thúc

3.5 Thiết kế giao diện

3.5.1 Thiết kế giao diện đăng nhập



The image shows a wireframe of a login form. It is titled "ĐĂNG NHẬP" (Login) in a dark grey font at the top left. Below the title, there are two input fields: one for "TÊN ĐĂNG NHẬP" (Username) and one for "MẬT KHẨU" (Password). The password field has a small icon on its right side, likely for toggling password visibility. Below the password field, there is a horizontal line, possibly for a separator or a link. The entire form is enclosed in a light grey border.

Hình 3.13 Phác thảo giao diện trang đăng nhập

Giao diện trang đăng nhập được thiết kế sẽ có các ô nhập tên đăng nhập và mật khẩu ứng với tiêu đề, có nút xác nhận đăng nhập.

3.5.2 Thiết kế giao diện trang đăng ký

The image shows a wireframe of a registration form. The form is titled "ĐĂNG KÝ" (Register) and is contained within a light gray border. The form itself has a white background. It includes the following elements:

- A title "ĐĂNG KÝ" at the top left of the form area.
- A label "EMAIL" above a single-line text input field.
- A label "MẬT KHẨU" (Password) above a single-line text input field.
- A label "MÃ XÁC NHẬN" (Confirmation Code) above a two-line input area. The first line is a single-line text input field, and the second line consists of a small square input field followed by a single-line text input field.
- A single-line text input field to the right of the first line of the confirmation code input area.
- A single-line text input field to the right of the second line of the confirmation code input area.
- A horizontal line at the bottom of the form area.

Hình 3.14 Phác thảo giao diện trang đăng ký

Giao diện trang đăng ký được thiết kế sẽ có các ô nhập tên đăng nhập email, ô nhập mật khẩu ô xác nhận mã song song là ô gửi mã để người dùng nhận được mã về email khi nhấn và nút xác nhận đăng ký.

3.5.3 Thiết kế giao diện chủ



Hình 3.15 Phác thảo giao diện trang chủ

Giao diện trang chủ được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa giao diện là 3 mã dùng để nhập mã HTML, CSS, JAVASCRIPT tương ứng, bên phải là giao diện hiển thị kết quả khi kiểm tra mã.

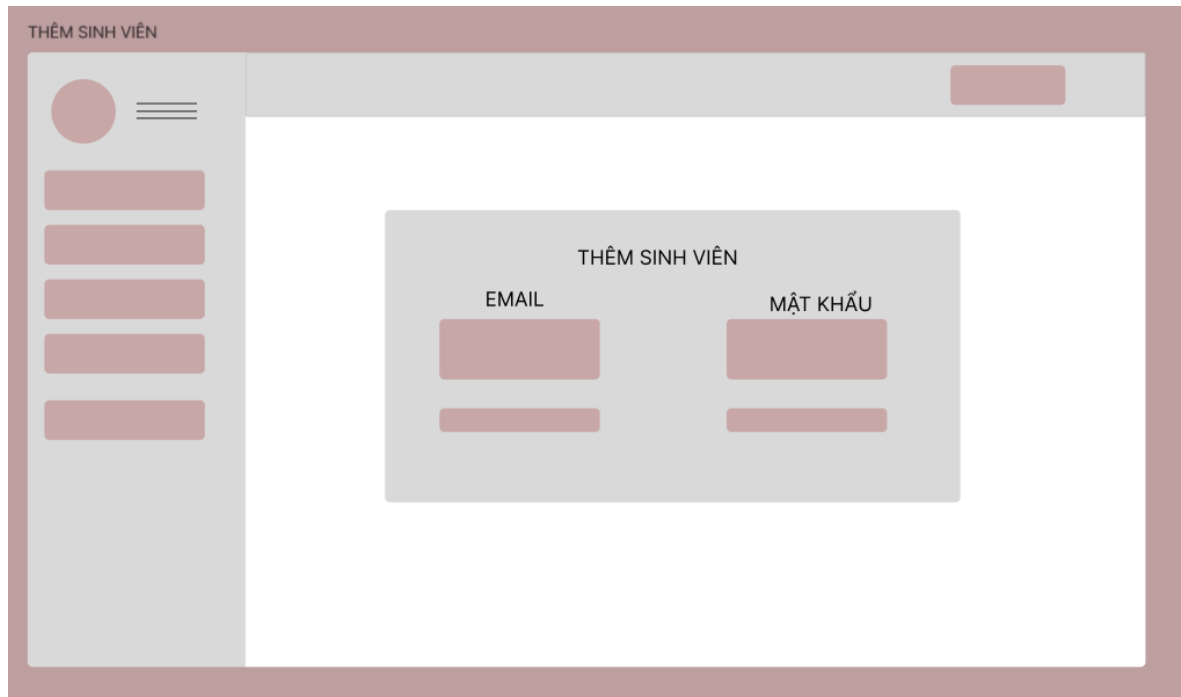
3.5.4 Thiết kế giao diện trang thêm giảng viên



Hình 3.16 Phác thảo giao diện trang thêm giảng viên

Giao diện trang thêm giảng viên được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để thêm thông tin giảng viên.

3.5.5 Thiết kế giao diện trang thêm sinh viên



Hình 3.17 Phác thảo giao diện trang thêm sinh viên

Giao diện trang thêm sinh viên được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để thêm thông tin sinh viên.

3.5.6 Thiết kế giao diện trang thông tin giảng viên



Hình 3.18 Phác thảo giao diện trang thông tin giảng viên

Giao diện trang xem thông tin giảng viên được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để xem thông tin của giảng viên.

3.5.7 Thiết kế giao diện trang thông tin sinh viên



Hình 3.19 Phác thảo giao diện trang thông tin sinh viên

Giao diện trang xem thông tin sinh viên được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để xem thông tin của sinh viên.

3.5.8 Thiết kế giao diện trang tạo lớp học

The wireframe shows a web interface for creating a class. The header is a dark red bar with the title 'TẠO LỚP HỌC' in white. Below the header is a light gray sidebar with a circular profile icon and a hamburger menu. The main content area is white and contains a gray box with the title 'TẠO LỚP HỌC'. Inside this box are five labeled input fields: 'Tên lớp học', 'Mã tham gia lớp học', 'Ngày bắt đầu', 'Ngày kết thúc', and 'Mô tả'. A red button labeled 'TẠO LỚP HỌC' is positioned at the bottom right of the form.

Hình 3.20 Phác thảo giao diện trang tạo lớp học

Giao diện trang tạo lớp học được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để tạo lớp học.

3.5.9 Thiết kế giao diện trang tìm kiếm lớp học



Hình 3.21 Phác thảo giao diện trang tìm kiếm lớp học

Giao diện trang tìm kiếm lớp học được thiết kế sẽ có các ô trên thanh menu tương ứng với các chức năng, chính giữa là giao diện dùng để tìm kiếm lớp học.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Chức năng hệ thống

4.1.1 Giao diện đăng ký

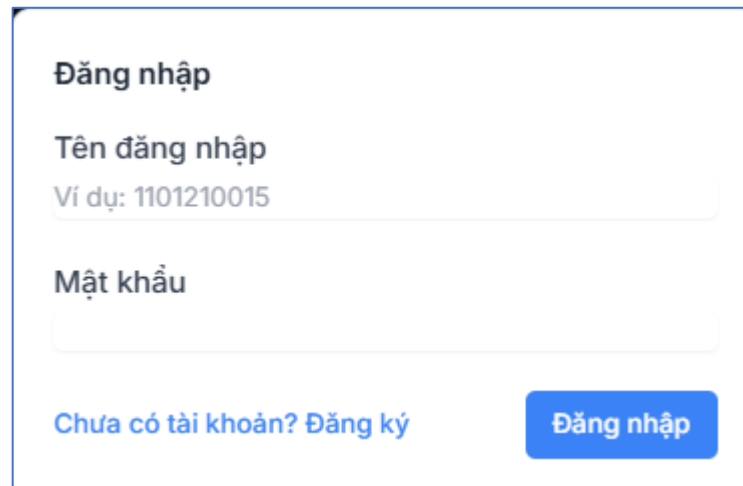
The screenshot displays a registration form titled "Đăng ký". It includes the following elements:

- Email:** A text input field with a placeholder "Ví dụ: example@gmail.com" and a note "Đây sẽ là email đăng nhập của bạn."
- Mật khẩu:** A password input field.
- Mã xác nhận:** A section for a confirmation code, including a text input field labeled "Nhập mã từ email" and a "Gửi mã" button. A note states: "Mã sẽ được gửi từ email cá nhân của quản trị viên đến email bạn vừa nhập."
- Role Selection:** A checkbox labeled "Đăng ký với vai trò Giảng viên". Below it, a note says: "Nếu không chọn, bạn sẽ được đăng ký với vai trò Sinh viên."
- Navigation:** A link "Đã có tài khoản? Đăng nhập" and a blue "Đăng ký" button.

Hình 4.1 Giao diện trang đăng ký

Với người dùng là sinh viên và giảng viên sẽ đăng ký tài khoản qua email cá nhân. Sinh viên và giảng viên phải nhập mail và mật khẩu để đăng ký, sau khi hệ thống xác nhận sẽ gửi về mail sinh viên và giảng viên mã xác nhận dùng để đăng ký. Trường hợp người dùng là giảng viên khi đăng ký sẽ tích vào ô đăng ký với quyền giảng viên.

4.1.2 Giao diện đăng nhập



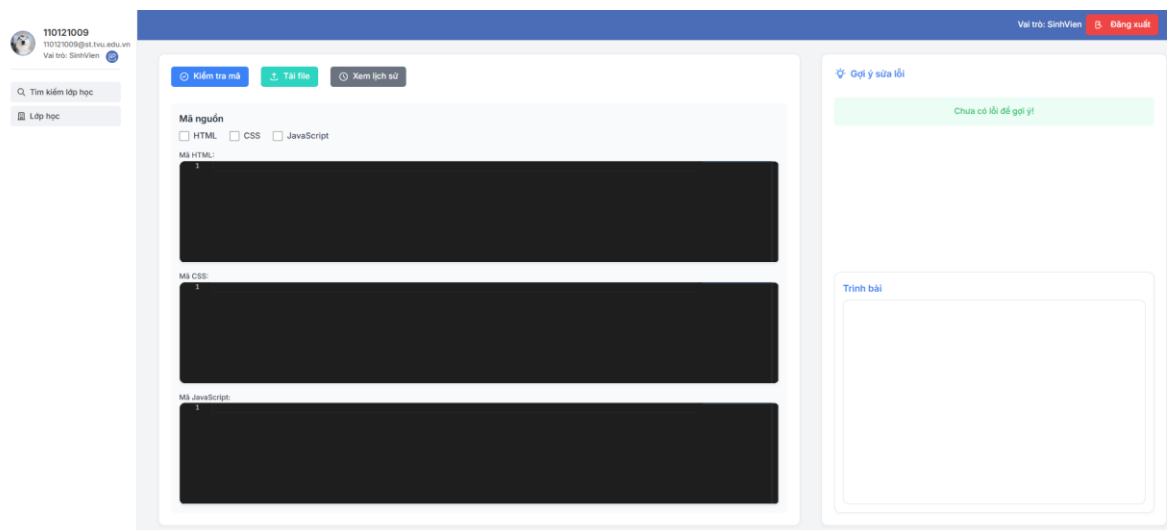
Hình 4.2 Giao diện trang đăng nhập

Với người dùng là sinh viên cần nhập chính xác thông tin đăng nhập gồm tên đăng nhập mặc định là mã số sinh viên và mật khẩu mặc định là mã số sinh viên. Tương tự với giảng viên cũng thế.

Trường hợp sinh viên đăng nhập bằng gmail cần nhập chính xác tên phía trước @gmail.com và mật khẩu lúc đăng ký.

Nếu người dùng nhập đúng, hệ thống sẽ kiểm tra quyền truy cập của tài khoản và chuyển đến trang chủ. Nếu sai sẽ thông báo lỗi

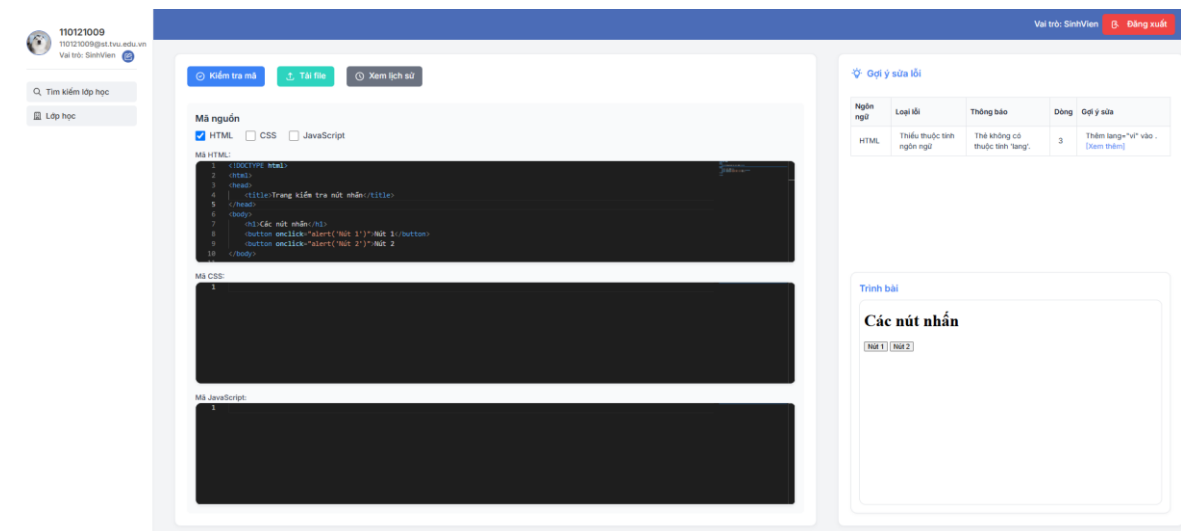
4.1.3 Giao diện trang chủ



Hình 4.3 Giao diện trang chủ

Quản trị, giảng viên hay sinh viên khi vừa đăng nhập thành công sẽ hiển thị giao diện trang chủ cho phép thực hiện kiểm tra mã bằng cách nhập mã trực tiếp hoặc tải file. Khi nhấn nút kiểm tra mã hệ thống sẽ xử lý và trả về kết quả nếu đoạn mã có lỗi hệ thống sẽ thông báo lỗi, nếu đoạn báo đúng hệ thống sẽ thông báo đoạn mã trên không có lỗi.

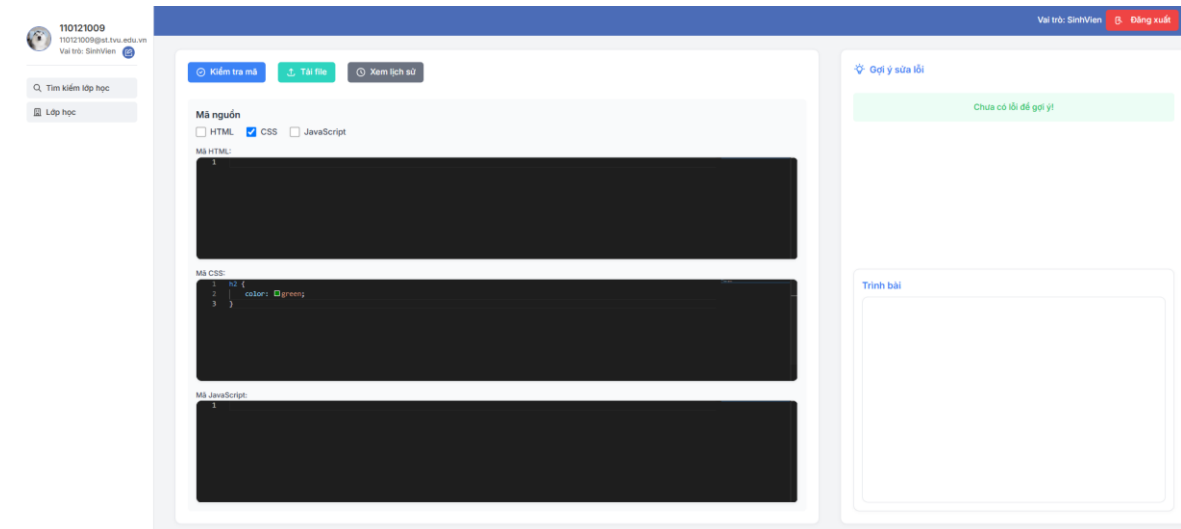
4.1.4 Giao diện kiểm tra mã HTML



Hình 4.4 Giao diện trang kiểm tra mã HTML

Người dùng nhập mã nguồn HTML vào giao diện Monaco, hệ thống sẽ xác nhận và thực hiện kiểm tra mã nguồn mà người dùng vừa nhập vào nếu có lỗi hệ thống sẽ thông báo lỗi và đưa ra gợi ý giúp người dùng phát hiện và chỉnh sửa lỗi.

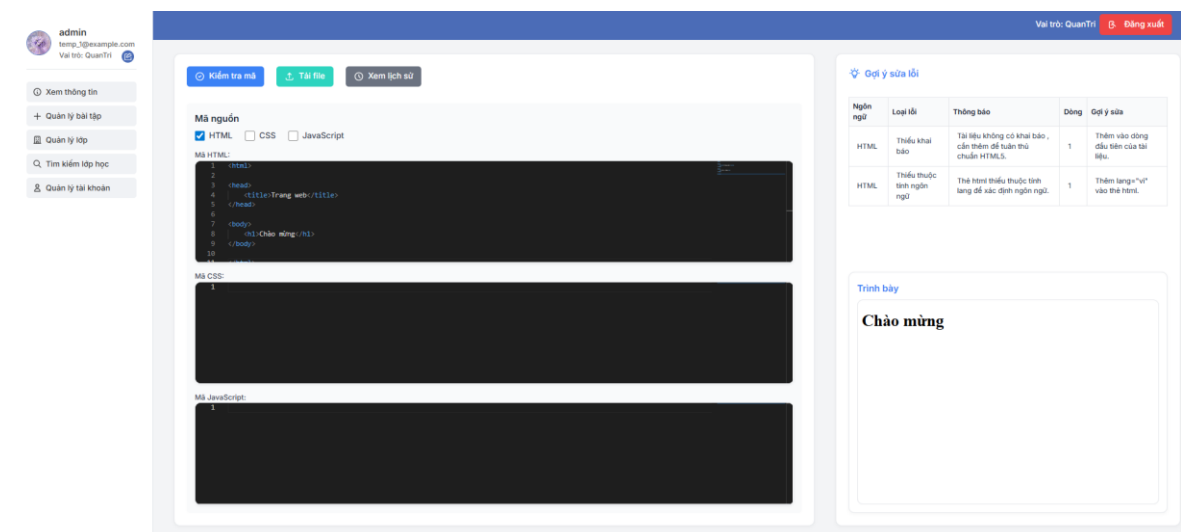
4.1.5 Giao diện kiểm tra mã CSS



Hình 4.5 Giao diện kiểm tra mã CSS

Người dùng nhập mã nguồn CSS vào giao diện Monaco, hệ thống sẽ xác nhận và thực hiện kiểm tra mã nguồn mà người dùng vừa nhập vào nếu có lỗi hệ thống sẽ thông báo lỗi và đưa ra gợi ý giúp người dùng phát hiện và chỉnh sửa lỗi.

4.1.6 Giao diện kiểm tra JAVASCRIPT



Hình 4.6 Giao diện kiểm tra mã JAVASCRIPT

Người dùng nhập mã nguồn JAVASCRIPT vào giao diện Monaco, hệ thống sẽ xác nhận và thực hiện kiểm tra mã nguồn mà người dùng vừa nhập vào nếu có lỗi hệ thống sẽ thông báo lỗi và đưa ra gợi ý giúp người dùng phát hiện và chỉnh sửa lỗi.

4.2 Chức năng quản trị, giảng viên

4.2.1 Giao diện xem giảng viên

TÊN ĐĂNG NHẬP	NGÀY TẠO	HÀNH ĐỘNG
anh	15/06/2025	<button>Sửa</button> <button>Xóa</button>
QuangHuy	16/06/2025	<button>Sửa</button> <button>Xóa</button>
ThiHong	16/06/2025	<button>Sửa</button> <button>Xóa</button>
MinhTuan	16/06/2025	<button>Sửa</button> <button>Xóa</button>
AnhDung	16/06/2025	<button>Sửa</button> <button>Xóa</button>
ThaoNguyen	16/06/2025	<button>Sửa</button> <button>Xóa</button>
ThaoAn	16/06/2025	<button>Sửa</button> <button>Xóa</button>
KhanhLinh	16/06/2025	<button>Sửa</button> <button>Xóa</button>
an	17/06/2025	<button>Sửa</button> <button>Xóa</button>

Hình 4.7 Giao diện xem giảng viên

Quản trị viên có quyền xem thông tin của từng giảng viên thông qua nút “Xem thông tin - giảng viên” trong giao diện hệ thống

4.2.2 Giao diện xem sinh viên

TÊN ĐĂNG NHẬP	NGÀY TẠO	HÀNH ĐỘNG
110121000	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121001	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121002	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121004	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121003	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121005	15/06/2025	<button>Sửa</button> <button>Xóa</button>
110121006	16/06/2025	<button>Sửa</button> <button>Xóa</button>
110121007	16/06/2025	<button>Sửa</button> <button>Xóa</button>
110121008	16/06/2025	<button>Sửa</button> <button>Xóa</button>
110121009	16/06/2025	<button>Sửa</button> <button>Xóa</button>

Hình 4.8 Giao diện xem sinh viên

Quản trị viên có quyền xem thông tin của từng sinh viên thông qua nút “Xem thông tin - sinh viên” trong giao diện hệ thống

4.2.3 Giao diện tạo bài tập

The screenshot shows the 'Tạo Bài Tập Mới' (Create New Exercise) form. The form has the following fields:

- Tên bài tập: Nhập tên bài tập
- Hạn nộp (bắt buộc): mm/dd/yyyy --:-- --
- Mô tả (bắt buộc): Nhập mô tả bài tập
- Nội dung bài tập: Ms HTML, Nhập mã HTML (nếu có)
- Mã CSS: Nhập mã CSS (nếu có)
- Mã JavaScript: Nhập mã JavaScript (nếu có)

Hình 4.9 Giao diện tạo bài tập

Quản trị có quyền tạo bài tập cho sinh viên sau khi tạo bài tập thành công sẽ chuyển sang giao diện danh sách bài tập

4.2.4 Giao diện xem danh sách bài tập

The screenshot shows the 'Danh sách Bài tập' (Exercise List) page. The table displays the following data:

Tên bài tập	HẠN NỘP	NGÔN NGỮ	LỚP HỌC	TRẠNG THÁI	XEM CHI TIẾT	HÀNH ĐỘNG
Sửa Lỗi Mã Javascript	23:00:00 31/07/2025	js	Không có lớp	Còn hạn	Xem chi tiết	Sửa Xóa
Sửa Lỗi Mã CSS	23:00:00 12/08/2025	css	Không có lớp	Còn hạn	Xem chi tiết	Sửa Xóa
Sửa lỗi sai tên	23:00:00 31/07/2025	html	4	Còn hạn	Xem chi tiết	Sửa Xóa
Sửa lỗi thiếu thẻ đóng đoạn HTML	23:00:00 15/07/2025	html	Không có lớp	Còn hạn	Xem chi tiết	Sửa Xóa
Sửa Lỗi Mã HTML	02:30:00 15/07/2025	html	1	Còn hạn	Xem chi tiết	Sửa Xóa

Hình 4.10 Giao diện xem danh sách bài tập

Quản trị có thể xem danh sách bài tập sau khi tạo bài tập xong. Tại giao diện danh sách bài tập quản trị có thể thực hiện thêm các thao tác xem chi tiết bài tập, sửa bài tập, xóa bài tập.

4.2.5 Giao diện sửa bài tập

The screenshot shows the 'Sửa Thông Tin Bài Tập' (Edit Exercise Information) form. The form is titled 'Sửa Thông Tin Bài Tập' and has a 'Quay lại Trang chính' (Go back to Home) button on the left and a 'Đăng xuất' (Logout) button on the right. The form contains several input fields: 'Tiêu đề' (Title) with the value 'Sửa Lỗi Mã Javascript', 'Hạn nộp' (Deadline) with the value '07/31/2025 04:00 PM', 'Mã tài' (Code), 'Ngôn ngữ' (Language) with the value 'js', 'HTML', 'CSS', and 'JavaScript' (containing a code snippet). At the bottom right, there are 'Hủy' (Cancel) and 'Lưu' (Save) buttons.

Hình 4.11 Giao diện sửa bài tập

Quản trị và sinh viên có thể sửa bài tập khi tạo xong trường hợp bài tập bị lỗi....

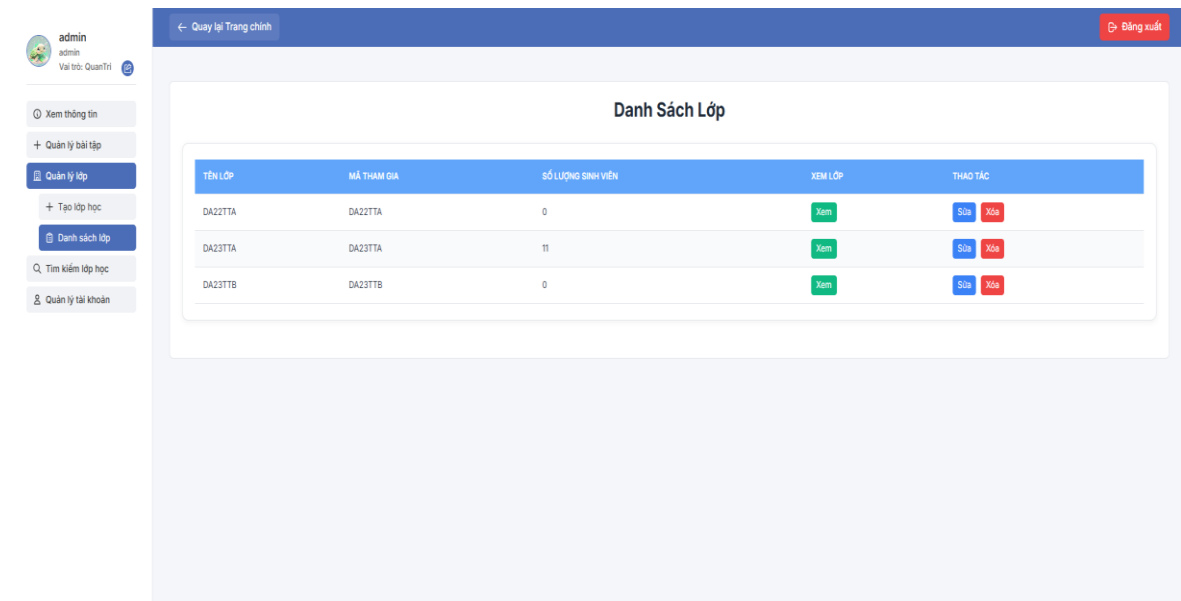
4.2.6 Giao diện tạo lớp học

The screenshot shows the 'Tạo Lớp Học' (Create Class) form. The form is titled 'Tạo Lớp Học' and has a 'Quay lại Trang chính' (Go back to Home) button on the left and a 'Đăng xuất' (Logout) button on the right. The form contains several input fields: 'Tên Lớp Học' (Class Name) with the placeholder 'Nhập tên lớp học', 'Mã Tham Gia Lớp Học' (Class Participation Code) with the placeholder 'Nhập mã tham gia (tối nhất 3 ký tự)', 'Ngày Bắt Đầu' (Start Date) with the placeholder 'mm/dd/yyyy', 'Ngày Kết Thúc' (End Date) with the placeholder 'mm/dd/yyyy', and 'Mô Tả' (Description) with the placeholder 'Nhập mô tả lớp học (tùy chọn)'. At the bottom, there is a 'Tạo Lớp Học' (Create Class) button.

Hình 4.12 Giao diện tạo lớp học

Quản trị và giảng viên có thể tạo lớp học, sau khi tạo lớp học xong sẽ chuyển sang giao diện danh sách lớp.

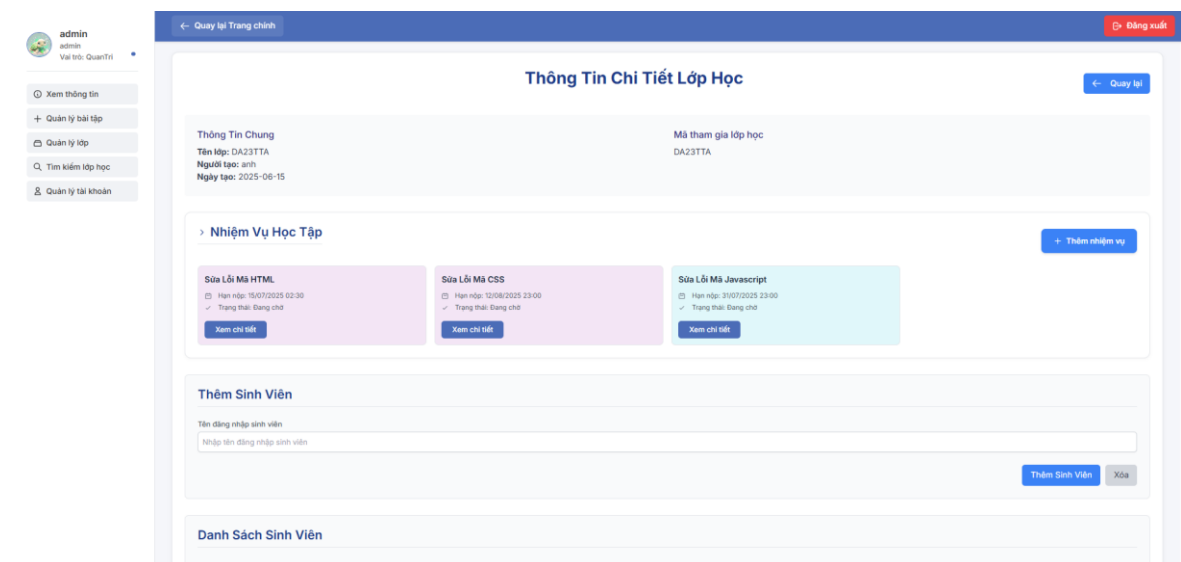
4.2.7 Giao diện xem thông tin lớp học



Hình 4.13 Giao diện xem danh sách lớp học

Quản trị và giảng viên có thể xem danh sách lớp.

4.2.8 Giao diện chi tiết lớp học



Hình 4.14 Giao diện chi tiết lớp học

Đối với quản trị có thể xem chi tiết các lớp do giảng viên tạo. Tại giao diện chi tiết lớp học sẽ hiển thị những thông tin cơ bản của giảng viên tạo lớp, có thể gán nhiệm vụ cho lớp học và thêm sinh viên vào lớp.

Đối với giảng viên tương tự như quản trị có thể thêm nhiệm vụ thêm sinh viên nhưng chỉ đối với lớp học do mình tạo.

4.2.9 Giao diện sửa lớp học

Hình 4.15 Giao diện sửa thông tin lớp học

Quản trị và giảng viên có thể sửa thông tin lớp học. Với giảng viên chỉ chỉnh sửa được thông tin của lớp học do mình tạo ra.

4.2.10 Giao diện thêm giảng viên

Hình 4.16 Giao diện thêm giảng viên

Quản trị có thêm thêm trực tiếp thông tin giảng viên.

4.2.11 Giao diện thêm sinh viên

The screenshot shows the 'Thêm Sinh Viên' (Add Student) form in the admin interface. The form is titled 'Thêm Sinh Viên' and has two input fields: 'Email' with a placeholder 'Nhập email sinh viên' and 'Mật Khẩu' with a placeholder 'Nhập mật khẩu (tối thiểu 6 ký tự)'. Below the input fields are two buttons: 'Thêm trường nhập liệu' (Add input field) and 'Thêm Sinh Viên' (Add Student). Below the buttons is a label 'Danh sách sinh viên mới:' (New student list:). The interface also shows a sidebar with navigation links and a top bar with a 'Đăng xuất' (Logout) button.

Hình 4.17 Giao diện thêm sinh viên

Quản trị và giảng viên có thêm thêm trực tiếp thông tin giảng viên.

4.3 Chức năng sinh viên

4.3.1 Giao diện trang chủ

The screenshot shows the student's main page (trang chủ) interface. The page has a top bar with a 'Đăng xuất' (Logout) button. Below the top bar is a sidebar with navigation links. The main content area is divided into two sections: 'Mã nguồn' (Source code) and 'Trình bài' (Article editor). The 'Mã nguồn' section has three tabs: 'HTML', 'CSS', and 'JavaScript'. Each tab has a text area for entering code. The 'Trình bài' section has a text area for writing an article. The interface also shows a sidebar with navigation links and a top bar with a 'Đăng xuất' (Logout) button.

Hình 4.18 Giao diện trang chủ quyền sinh viên

Đối với vai trò sinh viên khi đăng nhập thành công sẽ hiện ra giao diện trang chủ, sinh viên có thể trực tiếp sử dụng giao diện để kiểm tra lỗi.

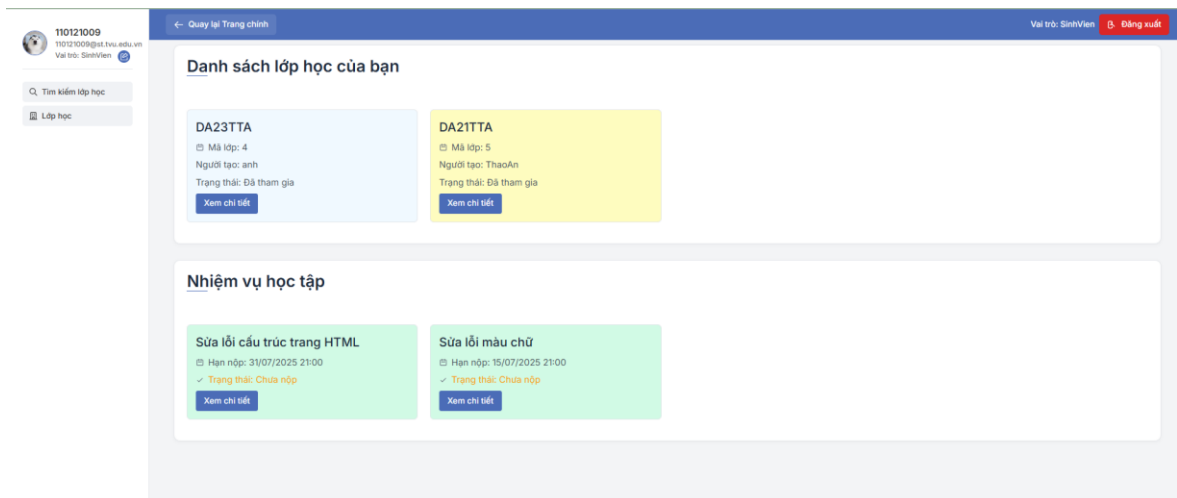
4.3.2 Giao diện trang tìm kiếm lớp học



Hình 4.19 Giao diện trang tìm kiếm lớp học

Giao diện tìm kiếm lớp học dành cho quản trị giảng viên và sinh viên để tìm kiếm lớp học.

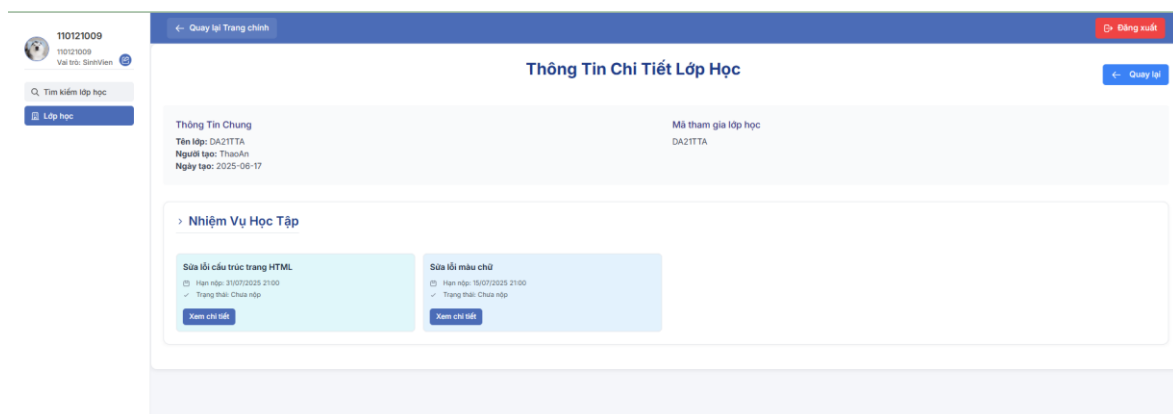
4.3.3 Giao diện trang lớp học



Hình 4.20 Giao diện trang lớp học

Với giao diện trang lớp học sinh viên sẽ thấy được những lớp học mình đã tham gia và nhiệm vụ học tập của các lớp mình đã tham gia.

4.3.4 Giao diện trang chi tiết lớp học



Hình 4.21 Giao diện trang chi tiết lớp học

Sinh viên có thể xem thông tin chi tiết lớp học đã tham gia bao gồm thông tin giảng viên dạy và các nhiệm vụ học tập của lớp.

4.4 Ưu điểm và hạn chế

Công cụ đã được triển khai thành công với nhiều tính năng nổi bật, hỗ trợ kiểm tra và đánh giá mã HTML, CSS và JavaScript một cách hiệu quả. Trong quá trình phát hiện lỗi, hệ thống có thể xác định các lỗi phổ biến như: thẻ HTML không đóng, thuộc tính không hợp lệ (ví dụ: viết sai class thành clas); trong CSS, phát hiện lỗi sai chính tả thuộc tính (như colour thay vì color), giá trị không hợp lệ (như display: invalid) hoặc thiếu dấu chấm phẩy ở cuối dòng. Đối với JavaScript, hệ thống phát hiện được các lỗi như biến chưa khai báo hoặc lỗi cú pháp cơ bản (thiếu dấu ngoặc,...). Ngoài việc phát hiện lỗi, hệ thống còn cung cấp gợi ý sửa lỗi chi tiết, giúp người dùng dễ hiểu và sửa nhanh, ví dụ: “Thuộc tính ‘colour’ sai chính tả, nên sửa thành ‘color’”. Một số lỗi đơn giản có thể được sửa tự động, như thêm dấu chấm phẩy hoặc thay thế từ sai bằng từ đúng. Về hiệu suất, công cụ hoạt động nhanh với đoạn mã dưới 1000 dòng, thời gian kiểm tra trung bình chỉ dưới 7 giây.

Bên cạnh đó, hệ thống được thiết kế với giao diện thân thiện, dễ sử dụng, sử dụng trình soạn thảo mã Monaco Editor và bố cục hợp lý với thanh điều hướng sidebar cùng bảng dữ liệu rõ ràng. Hệ thống cũng tích hợp cơ chế xác thực và phân quyền theo vai trò (Sinh viên, Giảng viên, Quản trị viên), đảm bảo hoạt động đúng quyền hạn của từng người dùng. Chức năng quản lý bài tập cho phép giảng viên tạo, chỉnh

sửa, xóa và phân bổ bài tập đến các lớp học, với giao diện hỗ trợ phân trang, lọc dữ liệu và tích hợp API giúp thao tác nhanh chóng và bảo mật.

Về ưu điểm, hệ thống nổi bật nhờ giao diện trực quan, khả năng tự động sửa lỗi, đồng thời tích hợp nhiều công cụ kiểm tra như `cssutils`, `Stylelint` và `W3C Validator` giúp tăng độ chính xác trong quá trình đánh giá mã. Tuy nhiên, công cụ vẫn còn một số hạn chế nhất định: hiện tại chưa xử lý được triệt để các lỗi phức tạp như vòng lặp vô hạn trong JavaScript hoặc các selector CSS quá phức tạp. Ngoài ra, hiệu suất hệ thống có thể giảm khi xử lý những đoạn mã lớn trên 2000 dòng.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết Luận

Đề tài xây dựng công cụ đánh giá chất lượng mã HTML, CSS, JAVASCRIPT tự động đã được xây dựng thành công, đáp ứng các yêu cầu cơ bản về quản lý bài tập, lớp học và thông tin người dùng. Giao diện thân thiện, tích hợp API và phân quyền rõ ràng đã tạo điều kiện thuận lợi cho các nhóm người dùng (Sinh viên, Giảng viên, Quản trị viên) trong việc sử dụng hệ thống. Các chức năng chính như đăng nhập, quản lý bài tập, quản lý lớp học và tìm kiếm lớp học đã được triển khai đầy đủ.

5.1.1 Kết luận kiểm tra mã HTML

Công cụ kiểm tra mã HTML hoạt động hiệu quả trong việc phát hiện các lỗi phổ biến như:

- Thiếu khai báo `<!DOCTYPE>`
- Thẻ chưa đóng (ví dụ: `<div>` không có thẻ đóng tương ứng `</div>`)
- Thẻ đóng không có thẻ mở (ví dụ: `</div>` không có thẻ mở tương ứng)
- Thẻ không hợp lệ
- Thuộc tính không hợp lệ (ví dụ: `clas` thay vì `class`)
- Thuộc tính trùng lặp
- Thẻ sai vị trí (ví dụ: `<h1>` nằm ngoài `<body>`)
- Thiếu thuộc tính `alt` trong thẻ ``

Sử dụng các thư viện BeautifulSoup và lxml để phân tích các lỗi cơ bản và làm đẹp mã nguồn. Hiển thị rõ ràng vị trí lỗi, mô tả và gợi ý sửa lỗi giúp người dùng dễ dàng cải thiện mã nguồn.

5.1.2 Kết luận kiểm tra mã CSS

Công cụ kiểm tra mã CSS hoạt động hiệu quả trong việc phát hiện các lỗi phổ biến như:

- Thiếu dấu `;`, dấu `{` và `}`.

-
- Thuộc tính sai (ví dụ: colour thay vì color)
 - Giá trị không hợp lệ (ví dụ: font-size: block)
 - Thiếu giá trị cho thuộc tính

Sử dụng thư viện cssutils để phân tích các lỗi cơ bản. Hiện thị rõ ràng vị trí lỗi, mô tả và gợi ý sửa lỗi giúp người dùng dễ dàng cải thiện mã nguồn.

5.1.3 Kết luận kiểm tra mã JAVASCRIPT

Công cụ kiểm tra mã CSS hoạt động hiệu quả trong việc phát hiện các lỗi phổ biến như:

- Thiếu dấu (,) , ; .
- Khai báo biến nhưng không sử dụng
- Khai báo hàm nhưng không sử dụng
- Gọi hàm không tồn tại

Áp dụng node.js và ESLint. Hiện thị rõ ràng vị trí lỗi, mô tả và gợi ý sửa lỗi giúp người dùng dễ dàng cải thiện mã nguồn.

5.2. Hướng phát triển

Trong giai đoạn tiếp theo, hệ thống có thể được mở rộng và nâng cấp theo các hướng sau:

Cải tiến cơ chế đánh giá mã, bổ sung khả năng phát hiện và phản hồi các lỗi logic phức tạp như selector sai, thẻ HTML không đóng đúng, lỗi lặp vô hạn trong JavaScript, hoặc cách sử dụng không tối ưu về cú pháp và cấu trúc mã.

Tích hợp hệ thống chấm điểm tự động theo tiêu chí linh hoạt, cho phép giảng viên thiết lập thang điểm dựa trên cấu trúc, tính đúng đắn và khả năng hiển thị của mã.

Cung cấp phản hồi gợi ý sửa lỗi có định hướng học tập, giúp sinh viên không chỉ biết lỗi mà còn hiểu cách cải thiện kỹ năng viết mã.

Xây dựng hệ thống thống kê học tập theo cá nhân và lớp học, hỗ trợ giảng viên theo dõi tiến độ, mức độ hoàn thành và chất lượng bài làm của từng sinh viên.

Phát triển thư viện bài tập mẫu đa dạng theo chủ đề và cấp độ, giúp giảng viên dễ dàng phân phối bài luyện tập phù hợp với nội dung môn học.

Tối ưu hóa giao diện người dùng và phản hồi tương tác theo thời gian thực, nâng cao trải nghiệm học tập và đánh giá.

Mở rộng hệ thống thành nền tảng học lập trình web mô phỏng lớp học trực tuyến, có thể tích hợp với LMS (Learning Management System).

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Nguyễn, T. A. (2020). Giới thiệu về lập trình Python. Nhà xuất bản Giáo dục Việt Nam
- [2] Lê, Q. T., Nguyễn, H. L., & Phạm, T. M. (2019). Lập trình web hiện đại với HTML, CSS và JavaScript
- [3] Welling, L., & Thomson, L. (2016). PHP and MySQL web development (5th ed.). Addison-Wesley Professional
- [4] Tailwind CSS. (2024). Tailwind CSS Documentation. Truy cập từ <https://tailwindcss.com/docs>
- [5] W3C. (2023). Web accessibility guidelines 2.2.