

TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



ISO 9001:2015

CAO KA KA

NGHIÊN CỨU QUY TRÌNH CI/CD TRONG
GITHUB ACTIONS ĐỂ TẠO HỆ THỐNG
ĐÁNH GIÁ LẬP TRÌNH TỰ ĐỘNG
CHO SINH VIÊN CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

TRÀ VINH, NĂM 2024

TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**NGHIÊN CỨU QUY TRÌNH CI/CD TRONG
GITHUB ACTIONS ĐỂ TẠO HỆ THỐNG
ĐÁNH GIÁ LẬP TRÌNH TỰ ĐỘNG
CHO SINH VIÊN CÔNG NGHỆ THÔNG TIN**

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

Sinh viên: **Cao Ka Ka**
Lớp: **DA20TTB**
MSSV: **110120138**
GVHD: **TS. Nguyễn Bảo An**

TRÀ VINH, NĂM 2024

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển không ngừng, việc tự động hóa các quy trình lặp đi lặp lại được ưu tiên phát triển. Đặc biệt trong ngành Công nghệ thông tin, các hệ thống đánh giá tự động dần trở nên cần thiết để cải thiện năng suất và chất lượng đào tạo trong lĩnh vực này. Quy trình CI/CD là một lựa chọn phù hợp với các yêu cầu đặt ra này. Hơn nữa, việc áp dụng quy trình CI/CD trong các dự án phát triển phần mềm giúp tăng tính nhất quán, giảm thiểu lỗi phát sinh trong quá trình phát triển và triển khai.

Bên cạnh đó, GitHub Actions là một tính năng miễn phí, được cung cấp bởi GitHub, với khả năng tích hợp mạnh mẽ, là một công cụ hữu ích hỗ trợ quá trình kiểm tra và đánh giá mã nguồn một cách tự động. Do đó, việc nghiên cứu và phát triển một hệ thống đánh giá lập trình tự động dựa trên GitHub Actions là một giải pháp khả thi không chỉ giúp sinh viên nâng cao kỹ năng lập trình mà còn giúp sinh viên nắm được cách hoạt động, cách sử dụng Git, đặc biệt là GitHub trong quá trình học tập.

Đề tài này hướng tới việc xây dựng hệ thống giúp sinh viên dễ dàng kiểm tra và đánh giá chất lượng bài làm của mình, tạo điều kiện thuận lợi cho việc học tập và giảng dạy của sinh viên, giảng viên ngành Công nghệ thông tin.

Ngoài ra, cùng với sự phát triển gần đây của các mô hình ngôn ngữ lớn, việc tích hợp các mô hình này vào quá trình đánh giá mã nguồn để nhanh chóng đưa ra các phản hồi, đánh giá nhằm cải thiện chất lượng mã nguồn cũng được ưu tiên trong đề tài này.

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn đến toàn thể quý thầy cô, giảng viên Trường Đại học Trà Vinh, đặc biệt là các thầy cô ở Khoa Kỹ thuật & Công nghệ, bộ môn Công nghệ thông tin đã tận tâm chỉ dạy trong suốt quá trình học tập tại trường và đã tạo điều kiện tốt nhất để em hoàn thành trọn vẹn bài báo cáo này.

Tiếp theo, em xin tỏ lòng biết ơn sâu sắc đến Thầy, TS. Nguyễn Bảo Ân – Giảng viên Khoa Kỹ thuật & Công nghệ, Trường Đại học Trà Vinh, trong quá trình hướng dẫn đồ án đã vô cùng tâm huyết trong việc truyền đạt kiến thức, đưa ra các hướng đi phù hợp và luôn tận tình, sẵn sàng hướng dẫn, chỉ bảo trong suốt quá trình thực hiện đồ án này.

Trong bài báo cáo, do lượng kiến thức và kinh nghiệm còn khiêm tốn, vẫn còn một số sai sót nhỏ không đáng kể. Do đó, em kính mong quý thầy cô thông cảm, góp ý để em có thể tiếp thu và cải thiện cho những nghiên cứu trong tương lai. Sau tất cả, kính chúc các thầy cô luôn dồi dào sức khỏe.

Em xin chân thành cảm ơn!

Trà Vinh, ngày tháng 10 năm 2024

Sinh viên thực hiện

Cao Ka Ka

NHẬN XÉT

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

Giảng viên hướng dẫn

**UBND TỈNH TRÀ VINH
TRƯỜNG ĐẠI HỌC TRÀ VINH**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh Phúc

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên hướng dẫn)

Họ và tên sinh viên: MSSV:

Ngành: Khóa:

Tên đề tài:

Ho và tên Giáo viên hướng dẫn:

Chức danh: Hoc vi:

NHÂN XÉT

1. Nội dung đề tài:

2. Ưu điểm:

.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Điểm mới đề tài:

.....
.....
.....
.....
.....

5. Giá trị thực trên đề tài:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

7. Đề nghị sửa chữa bổ sung:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

8. Đánh giá:

.....
.....
.....
.....
.....

Trà Vinh, ngày tháng năm 2024
Giảng viên hướng dẫn
(Ký & ghi rõ họ tên)

NHẬN XÉT

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

Giảng viên chấm

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét: Nguyễn Ngọc Đan Thanh

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Bộ môn CNTT, Khoa Kỹ thuật & Công nghệ

Tên sinh viên: Cao Ka Ka

Tên đề tài đồ án, khóa luận tốt nghiệp: Nghiên cứu quy trình CI/CD trong GitHub

Actions để tạo hệ thống đánh giá lập trình tự động cho sinh viên Công nghệ thông tin

I. Ý KIẾN NHẬN XÉT

1. Nội dung:

- Gồm 5 chương và nội dung đáp ứng theo đề cương chi tiết. Trong đó, chương 2 trình bày cở sở lý thuyết ở mức cơ bản.
 - Nội dung trình bày có tính logic cao

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

Có kết hợp mô hình ngôn ngữ lớn trong việc đánh giá chất lượng mã

3. Ứng dụng thực tế:

Mang tính thực tế cao, áp dụng phục vụ cho quá trình học tập và rèn luyện các học phần cơ sở.

II. CÁC VẤN ĐỀ CẦN LÀM RÕ

(Các câu hỏi của giáo viên phản biện)

III. KẾT LUẬN

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

Đồng ý.

....., ngày tháng năm 2024

Người nhận xét

(Ký & ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. ĐẶT VĂN ĐỀ	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu	1
1.3. Nội dung.....	1
1.4. Đối tượng và phạm vi nghiên cứu	2
1.5. Phương pháp nghiên cứu	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1. Tổng quan về CI/CD	3
2.1.1. <i>Tìm hiểu về CI</i>	3
2.1.2. <i>Tìm hiểu về CD</i>	6
2.1.3. <i>Quy trình CI/CD</i>	9
2.2. Các phương pháp kiểm tra và đánh giá chất lượng mã nguồn	12
2.2.1. <i>Kiểm thử đơn vị</i>	12
2.2.2. <i>Kiểm thử tích hợp</i>	14
2.2.3. <i>Kiểm thử chức năng</i>	15
2.2.4. <i>Kiểm thử hồi quy</i>	17
2.2.5. <i>Kiểm thử bảo mật</i>	18
2.2.6. <i>Kiểm thử hiệu suất</i>	19
2.3. Tìm hiểu về GitHub Actions.....	20
2.4. Tìm hiểu về GitHub REST API	22
2.5. Tổng quan về các nền tảng đánh giá lập trình tự động	26
2.6. Tìm hiểu về mô hình ngôn ngữ lớn và Gemini.....	30
2.6.1. <i>Tìm hiểu về mô hình ngôn ngữ lớn</i>	30
2.6.2. <i>Tìm hiểu về Gemini</i>	31
2.7. Các công nghệ xây dựng trang web	32
2.7.1. <i>Ngôn ngữ HTML</i>	32
2.7.2. <i>Bảng định kiểu CSS</i>	35
2.7.3. <i>Ngôn ngữ Javascript</i>	36
2.7.4. <i>React</i>	38
2.7.5. <i>TailwindCSS</i>	40
2.7.6. <i>Shadcn/ui</i>	42
2.7.7. <i>TypeScript</i>	44
2.7.8. <i>Node.js</i>	45
2.7.9. <i>Express</i>	47
2.7.10. <i>Socket.IO</i>	50

2.7.11. MySQL	52
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU.....	55
3.1. Mô tả bài toán	55
3.2. Phân tích yêu cầu và lựa chọn công nghệ	56
3.2.1. Phân tích yêu cầu	56
3.2.2. Lựa chọn công nghệ.....	57
3.3. Thiết kế kiến trúc hệ thống	60
3.3.1. Thiết kế kiến trúc phần mềm	60
3.3.2. Thiết kế cơ sở dữ liệu	63
3.4. Thiết kế workflow cho GitHub Actions	73
3.5. Xây dựng API phía máy chủ (Back-end).....	85
3.6. Các API xác thực (Auth).....	85
3.7. Các API chức năng người dùng (Users)	85
3.8. Các API chức năng bài viết (Posts)	86
3.9. Các API chức năng khoá học (Courses)	88
3.10. Các API chức năng bài tập (Problems).....	90
3.11. Các API chức năng thảo luận (Discussions).....	93
3.12. Các API chức năng nộp bài (Submissions)	95
3.13. Các API chức năng cuộc thi (Contests)	95
3.14. Thiết kế giao diện và chức năng phía sinh viên, giảng viên (Front-end)	98
3.14.1. Thiết kế khung giao diện làm việc chính	98
3.14.2. Chức năng Đăng nhập	99
3.14.3. Chức năng Bài tập	100
3.14.4. Chức năng Thảo luận	101
3.14.5. Chức năng Cuộc thi	102
3.14.6. Chức năng Khoá học	104
3.14.7. Chức năng Bảng xếp hạng	106
3.14.8. Chức năng Diễn đàn	106
3.14.9. Chức năng Quản lý khoá học (dành cho giảng viên)	108
3.15. Thiết kế giao diện và chức năng phía quản trị viên (Front-end)	109
3.15.1. Chức năng Tổng quan.....	109
3.15.2. Chức năng Quản lý người dùng.....	110
3.15.3. Chức năng Quản lý khoá học	111
3.15.4. Chức năng Quản lý bài tập	111
3.15.5. Chức năng Quản lý cuộc thi	112
3.15.6. Chức năng Quản lý bài viết	112

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	114
4.1. Bộ dữ liệu thử nghiệm	114
4.1.1. Dữ liệu thử nghiệm chức năng Khoá học	114
4.1.2. Dữ liệu thử nghiệm chức năng Luyện tập	136
4.1.3. Dữ liệu thử nghiệm chức năng Cuộc thi	140
4.3. Kết quả thực nghiệm với vai trò sinh viên.....	154
4.3.1. Chức năng Đăng nhập	154
4.3.2. Chức năng Trang chủ	155
4.3.3. Chức năng Khoá học	155
4.3.4. Chức năng Bài tập	159
4.3.5. Chức năng Luyện tập	170
4.3.6. Chức năng Cuộc thi	172
4.3.7. Chức năng Diễn đàn	176
4.4. Kết quả thực nghiệm với vai trò giảng viên	177
4.5. Kết quả thực nghiệm với vai trò quản trị viên (Admin)	180
4.5.1. Chức năng Tổng quan.....	180
4.5.2. Chức năng Quản lý người dùng.....	180
4.5.3. Chức năng Quản lý khoá học	181
4.5.4. Chức năng Quản lý bài tập	181
4.5.5. Chức năng Quản lý cuộc thi	183
4.5.6. Chức năng Quản lý bài viết	186
4.6. Các chức năng giao diện khác	187
4.7. Kết quả thực nghiệm ứng dụng HiCommit CLI	188
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	193
5.1. Kết luận.....	193
5.2. Hướng phát triển	193
DANH MỤC TÀI LIỆU THAM KHẢO.....	194
PHỤ LỤC 1: CÁC CÔNG CỤ VÀ PHẦN MỀM HỖ TRỢ XÂY DỰNG TRANG WEB	195
1. Visual Studio Code	195
2. Xampp	195
3. Postman	196
4. Ngrok	198

DANH MỤC CÁC HÌNH ẢNH

Hình 2.1. Quy trình của Test Driven Development (TDD)	13
Hình 2.2. Quy trình cơ bản của các nền tảng Online Judge	27
Hình 3.1. Kiến trúc phần mềm của HiCommit	60
Hình 3.2. Lược đồ cơ sở dữ liệu của HiCommit	63
Hình 3.3. Bản phác thảo khung giao diện chính	99
Hình 3.4. Bản phác thảo giao diện Đăng nhập	99
Hình 3.5. Bản phác thảo giao diện Bài tập	101
Hình 3.6. Bản phác thảo giao diện Chi tiết bài tập	101
Hình 3.7. Bản phác thảo giao diện Thảo luận	102
Hình 3.8. Bản phác thảo giao diện Danh sách cuộc thi	103
Hình 3.9. Bản phác thảo giao diện Chi tiết cuộc thi	103
Hình 3.10. Bản phác thảo giao diện Bảng xếp hạng cuộc thi	104
Hình 3.11. Bản phác thảo giao diện Khoá học	105
Hình 3.12. Bản phác thảo giao diện Chi tiết khoá học	105
Hình 3.13. Bản phác thảo giao diện Bảng xếp hạng	106
Hình 3.14. Bản phác thảo giao diện Diễn đàn	107
Hình 3.15. Bản phác thảo giao diện Xem bài viết	107
Hình 3.16. Bản phác thảo giao diện Quản lý khoá học	108
Hình 3.17. Bản phác thảo giao diện Quản lý chi tiết khoá học	109
Hình 3.18. Bản phác thảo giao diện Admin – Tổng quan	110
Hình 3.19. Bản phác thảo giao diện Admin – Quản lý người dùng	111
Hình 3.20. Bản phác thảo giao diện Admin – Quản lý khoá học	111
Hình 3.21. Bản phác thảo giao diện Admin – Quản lý bài tập	112
Hình 3.22. Bản phác thảo giao diện Admin – Quản lý cuộc thi	112
Hình 3.23. Bản phác thảo giao diện Admin – Quản lý bài viết	113
Hình 4.1. Giao diện thực tế chức năng Đăng nhập	154
Hình 4.2. Giao diện chọn tài khoản đăng nhập GitHub	154
Hình 4.3. Giao diện thực tế chức năng Trang chủ	155
Hình 4.4. Giao diện thực tế chức năng Khoá học	155
Hình 4.5. Giao diện khi chưa tham gia khoá học	156
Hình 4.6. Giao diện khi được yêu cầu nhập mật khẩu khoá học	156
Hình 4.7. Giao diện chi tiết trong một khoá học	157
Hình 4.8. Thông tin các chuẩn đầu ra của một khoá học	157
Hình 4.9. Giao diện phần thực hành trong một khoá học	157
Hình 4.10. Chức năng phân tích số liệu khoá học	158
Hình 4.11. Giao diện chức năng phân tích số liệu khoá học – Tần suất nộp bài	158
Hình 4.12. Giao diện chức năng phân tích số liệu khoá học – Mức độ hoàn thành	158
Hình 4.13. Giao diện thực tế một bài tập trên HiCommit – Nội dung bài tập	159
Hình 4.14. Các ví dụ của bài tập và chức năng nộp bài	159
Hình 4.15. Chức năng nộp bài thông qua giao diện HiCommit	160
Hình 4.16. Commit thay đổi trước khi nộp bài	160
Hình 4.17. Trạng thái đang chấm bài	162
Hình 4.18. Giao diện thực tế một bài tập trên HiCommit – Thảo luận	163
Hình 4.19. Tạo cuộc thảo luận mới	163
Hình 4.20. Tạo cuộc thảo luận thành công	164
Hình 4.21. Chi tiết về cuộc thảo luận	164
Hình 4.22. Giao diện thực tế một bài tập trên HiCommit – Các bài nộp	165

Hình 4.23. Kiểm tra đơn vị của một bài nộp – Trường hợp kết quả chính xác	165
Hình 4.24. Trường hợp sai kết quả	166
Hình 4.25. Chi tiết về lỗi của một testcase chưa hoàn thành	166
Hình 4.26. Gợi ý hoàn thành testcase	166
Hình 4.27. Trường hợp lỗi trong quá trình thực hiện	167
Hình 4.28. Trường hợp lỗi biên dịch chương trình.....	167
Hình 4.29. Chức năng xem lại mã nguồn đã nộp	167
Hình 4.30. Đánh giá chất lượng mã nguồn của một bài nộp	168
Hình 4.31. Các gợi ý chỉnh sửa mã nguồn.....	168
Hình 4.32. Chi tiết quá trình chấm bài trên GitHub Actions	169
Hình 4.33. Giao diện thực tế chức năng bài tập trong khoá học	169
Hình 4.34. Giao diện thực tế chức năng bài tập tự do	170
Hình 4.35. Giao diện thực tế chức năng bài tập trong cuộc thi	170
Hình 4.36. Giao diện thực tế chức năng Luyện tập	171
Hình 4.37. Giao diện thực tế chức năng Bảng xếp hạng (Luyện tập)	171
Hình 4.38. Giao diện thực tế chức năng Trang cá nhân	172
Hình 4.39. Giao diện thực tế chức năng Cuộc thi.....	172
Hình 4.40. Giao diện thực tế trước khi diễn ra cuộc thi	173
Hình 4.41. Xác nhận tham gia cuộc thi.....	173
Hình 4.42. Giao diện thực tế trong thời gian làm bài của cuộc thi	174
Hình 4.43. Giao diện thực tế Bảng xếp hạng cuộc thi	174
Hình 4.44. Giao diện thực tế các bài nộp trong cuộc thi	175
Hình 4.45. Giao diện thực tế sau khi cuộc thi kết thúc	175
Hình 4.46. Giao diện thực tế chức năng Diễn đàn.....	176
Hình 4.47. Giao diện khi xem một bài viết.....	176
Hình 4.48. Nút đăng tải bài viết mới lên diễn đàn.....	177
Hình 4.49. Tạo bài viết mới đăng tải lên diễn đàn.....	177
Hình 4.50. Giao diện thực tế chức năng Quản lý khoá học	177
Hình 4.51. Giao diện thực tế chức năng tạo khoá học mới	178
Hình 4.52. Giao diện thực tế quản lý một khoá học	178
Hình 4.53. Cài đặt khoá học	179
Hình 4.54. Quản lý thành viên trong khoá học	179
Hình 4.55. Quản lý các chương, bài tập trong khoá học	179
Hình 4.56. Giao diện thực tế chức năng Quản lý tổng quan.....	180
Hình 4.57. Giao diện thực tế chức năng Quản lý người dùng	180
Hình 4.58. Giao diện thực tế chức năng Quản lý khoá học	181
Hình 4.59. Giao diện thực tế chức năng Quản lý bài tập.....	181
Hình 4.60. Tạo bài tập mới (Thông tin cơ bản)	182
Hình 4.61. Tạo bài tập mới (Thông tin chi tiết).....	182
Hình 4.62. Tạo bài tập mới (Các ví dụ và các Testcase)	183
Hình 4.63. Chức năng chỉnh sửa bài tập	183
Hình 4.64. Giao diện thực tế chức năng Quản lý cuộc thi.....	184
Hình 4.65. Giao diện chức năng tạo cuộc thi mới	184
Hình 4.66. Giao diện chi tiết quản lý một cuộc thi.....	185
Hình 4.67. Giao diện chỉnh sửa một cuộc thi	185
Hình 4.68. Quản lý danh sách tham gia cuộc thi	186
Hình 4.69. Giao diện thực tế chức năng Quản lý bài viết (Đã duyệt)	186
Hình 4.70. Giao diện thực tế chức năng Quản lý bài viết (Đang chờ duyệt)	186

Hình 4.71. Giao diện xem trước bài viết	187
Hình 4.72. Chế độ tối trên hệ thống HiCommit.....	187
Hình 4.73. Quá trình cài đặt HiCommit CLI	188
Hình 4.74. Đăng nhập với HiCommit CLI	189
Hình 4.75. Lựa chọn tài khoản để đăng nhập	189
Hình 4.76. Đăng nhập thành công với HiCommit CLI	189
Hình 4.77. Kiểm tra trạng thái đăng nhập trong HiCommit CLI	190
Hình 4.78. Khởi tạo với HiCommit CLI.....	190
Hình 4.79. Ví dụ tạo bài tập FIBONACCI_NUMBER	191
Hình 4.80. Lỗi khi chưa có nhánh tương ứng trên remote repo	192
Hình 4.81. Nộp bài sử dụng Git thành công	192
Hình 4.82. Kết quả chấm bài được hiển thị trên giao diện web của HiCommit.....	192

DANH MỤC CÁC BẢNG, SƠ ĐỒ

Bảng 2.1. Các kết quả chấm bài trong một hệ thống Online Judge.....	28
Bảng 2.2. Các thẻ HTML thông dụng	34
Bảng 3.1. Các chức năng trong hệ thống HiCommit.....	56
Bảng 3.2. Các công nghệ được lựa chọn sử dụng.....	58
Bảng 3.3. Bảng mô tả bảng Users.....	64
Bảng 3.4. Bảng mô tả bảng Posts	65
Bảng 3.5. Bảng mô tả bảng Courses	65
Bảng 3.6. Bảng mô tả bảng Users_Courses.....	66
Bảng 3.7. Bảng mô tả bảng Units	67
Bảng 3.8. Bảng mô tả bảng Problems.....	67
Bảng 3.9. Bảng mô tả bảng Examples	68
Bảng 3.10. Bảng mô tả bảng Testcases	69
Bảng 3.11. Bảng mô tả bảng Submissions.....	69
Bảng 3.12. Bảng mô tả bảng Discussions.....	70
Bảng 3.13. Bảng mô tả bảng Comments	71
Bảng 3.14. Bảng mô tả bảng Contests	72
Bảng 3.15. Bảng mô tả bảng Users_Contests	73
Bảng 3.16. Bảng mô tả API lấy thông tin người dùng hiện tại	85
Bảng 3.17. Bảng mô tả API đăng nhập.....	85
Bảng 3.18. Bảng mô tả API đăng xuất	85
Bảng 3.19. Bảng mô tả API lấy thông tin trang cá nhân	85
Bảng 3.20. Bảng mô tả API lấy thông tin bảng xếp hạng	86
Bảng 3.21. Bảng mô tả API lấy danh sách bài viết	86
Bảng 3.22. Bảng mô tả API lấy bài viết theo ID	86
Bảng 3.23. Bảng mô tả API lấy danh sách bài viết theo tag.....	86
Bảng 3.24. Bảng mô tả API tạo bài viết mới	86
Bảng 3.25. Bảng mô tả API công bố bài viết	87
Bảng 3.26. Bảng mô tả API duyệt bài viết	87
Bảng 3.27. Bảng mô tả API chỉnh sửa bài viết.....	87
Bảng 3.28. Bảng mô tả API xoá bài viết	87
Bảng 3.29. Bảng mô tả API lấy danh sách các khoá học	88
Bảng 3.30. Bảng mô tả API lấy danh sách khoá học đã tham gia	88
Bảng 3.31. Bảng mô tả API lấy danh sách khoá học đã tạo	88
Bảng 3.32. Bảng mô tả API lấy thông tin khoá học theo ID (Slug)	88
Bảng 3.33. Bảng mô tả API lấy số liệu thống kê khoá học	88
Bảng 3.34. Bảng mô tả API lấy thông tin khoá học dành cho người quản trị	89
Bảng 3.35. Bảng mô tả API tạo khoá học mới	89
Bảng 3.36. Bảng mô tả API tham gia khoá học.....	89
Bảng 3.37. Bảng mô tả API lấy ra danh sách chương và bài tập trong khoá học	89
Bảng 3.38. Bảng mô tả API tạo chương (phần) mới trong khoá học	89
Bảng 3.39. Bảng mô tả API chỉnh sửa khoá học	90
Bảng 3.40. Bảng mô tả API xoá khoá học (xoá mềm)	90
Bảng 3.41. Bảng mô tả API lấy danh sách bài tập tự do	90
Bảng 3.42. Bảng mô tả API lấy thông tin bài tập theo đường dẫn	90
Bảng 3.43. Bảng mô tả API lấy danh sách testcases của bài tập	91
Bảng 3.44. Bảng mô tả API lấy các phân tích số liệu của bài tập	91
Bảng 3.45. Bảng mô tả API tạo khoá học mới	91

Bảng 3.46. Bảng mô tả API ghi kết quả chấm bài.....	92
Bảng 3.47. Bảng mô tả API chỉnh sửa bài tập	92
Bảng 3.48. Bảng mô tả API xoá bài tập (xoá mềm)	93
Bảng 3.49. Bảng mô tả API lấy danh sách các cuộc thảo luận theo đường dẫn bài tập....	93
Bảng 3.50. Bảng mô tả API lấy thông tin cuộc thảo luận theo ID	93
Bảng 3.51. Bảng mô tả API lấy danh sách bình luận của một cuộc thảo luận	93
Bảng 3.52. Bảng mô tả API tạo cuộc thảo luận mới	93
Bảng 3.53. Bảng mô tả API tạo bình luận mới.....	94
Bảng 3.54. Bảng mô tả API đóng/mở cuộc thảo luận	94
Bảng 3.55. Bảng mô tả API chỉnh sửa thông tin cuộc thảo luận	94
Bảng 3.56. Bảng mô tả API xoá cuộc thảo luận (xoá mềm)	94
Bảng 3.57. Bảng mô tả API lấy danh sách nộp bài của người dùng	95
Bảng 3.58. Bảng mô tả API lấy danh sách nộp bài của bài tập	95
Bảng 3.59. Bảng mô tả API lấy thông tin nộp bài theo ID	95
Bảng 3.60. Bảng mô tả API công khai/ẩn mã nguồn.....	95
Bảng 3.61. Bảng mô tả API lấy danh sách các cuộc thi	95
Bảng 3.62. Bảng mô tả API lấy thông tin cuộc thi theo ID	96
Bảng 3.63. Bảng mô tả API lấy danh sách nộp bài của cuộc thi	96
Bảng 3.64. Bảng mô tả API lấy danh sách Bảng xếp hạng	96
Bảng 3.65. Bảng mô tả API lấy đề bài của cuộc thi	96
Bảng 3.66. Bảng mô tả API yêu cầu tham gia cuộc thi	96
Bảng 3.67. Bảng mô tả API rời khỏi cuộc thi.....	96
Bảng 3.68. Bảng mô tả API tạo cuộc thi mới	97
Bảng 3.69. Bảng mô tả API ghim/bỏ ghim cuộc thi.....	97
Bảng 3.70. Bảng mô tả API công bố/ẩn cuộc thi.....	97
Bảng 3.71. Bảng mô tả API thay đổi trạng thái cuộc thi	97
Bảng 3.72. Bảng mô tả API chỉnh sửa cuộc thi	98
Bảng 3.73. Bảng mô tả API xoá bài tập trong cuộc thi (xoá mềm)	98
Bảng 3.74. Bảng mô tả API xoá cuộc thi (xoá mềm)	98
Bảng 3.75. Các trạng thái trong Bảng xếp hạng cuộc thi	104
Bảng 4.1. Dữ liệu thử nghiệm chức năng khoá học.....	114
Bảng 4.2. Dữ liệu thử nghiệm bài tập 1.1	115
Bảng 4.3. Dữ liệu thử nghiệm bài tập 1.2	116
Bảng 4.4. Dữ liệu thử nghiệm bài tập 2.1	117
Bảng 4.5. Dữ liệu thử nghiệm bài tập 2.2	119
Bảng 4.6. Dữ liệu thử nghiệm bài tập 2.3	121
Bảng 4.7. Dữ liệu thử nghiệm bài tập 3.1	122
Bảng 4.8. Dữ liệu thử nghiệm bài tập 3.2	124
Bảng 4.9. Dữ liệu thử nghiệm bài tập 4.1	126
Bảng 4.10. Dữ liệu thử nghiệm bài tập 4.2	127
Bảng 4.11. Dữ liệu thử nghiệm bài tập 5.1	129
Bảng 4.12. Dữ liệu thử nghiệm bài tập 5.2	131
Bảng 4.13. Dữ liệu thử nghiệm bài tập 6.1	133
Bảng 4.14. Dữ liệu thử nghiệm bài tập 6.2	134
Bảng 4.15. Dữ liệu thử nghiệm bài tập Chuỗi đối xứng.....	136
Bảng 4.16. Dữ liệu thử nghiệm bài tập Tổng các chữ số.....	137
Bảng 4.17. Dữ liệu thử nghiệm bài tập Kiểm tra số nguyên tố	139
Bảng 4.18. Dữ liệu thử nghiệm chức năng Cuộc thi	140

Bảng 4.19. Dữ liệu thử nghiệm bài tập Tổng ước số nguyên tố	141
Bảng 4.20. Dữ liệu thử nghiệm bài tập Tiền thừa của Hiệp	142
Bảng 4.21. Dữ liệu thử nghiệm bài tập Con éch thông minh	144
Bảng 4.22. Dữ liệu thử nghiệm bài tập Xây dựng mạng lưới viễn thông.....	146
Bảng 4.23. Sơ đồ hoạt động quá trình nộp bài.....	161
Bảng 4.24. Nhánh tương ứng với bài tập trên GitHub	162
Bảng 4.25. Các lượt chạy của GitHub Actions.....	163

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
API	Application Programming Interface
AWS	Amazon Web Service
CI/CD	Continuous Integration/Continuous Delivery
CLI	Command Line Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
ICPC	International Collegiate Programming Contest
LLM	Large Language Model
RDBMS	Relational Database Management Systems
TDD	Test Driven Development
VPS	Virtual Private Server

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển không ngừng, các hệ thống đánh giá tự động trở nên cần thiết để cải thiện năng suất và chất lượng đào tạo trong lĩnh vực Công nghệ thông tin. Việc áp dụng quy trình Continuous Integration/Continuous Delivery (CI/CD) trong các dự án phát triển phần mềm giúp tăng tính nhất quán, giảm thiểu lỗi phát sinh trong quá trình phát triển và triển khai.

Bên cạnh đó, GitHub Actions là một tính năng miễn phí, được cung cấp bởi GitHub, với khả năng tích hợp mạnh mẽ, là một công cụ hữu ích hỗ trợ quá trình kiểm tra và đánh giá mã nguồn một cách tự động. Do đó, việc nghiên cứu và phát triển một hệ thống đánh giá lập trình tự động dựa trên GitHub Actions là một giải pháp khả thi không chỉ giúp sinh viên nâng cao kỹ năng lập trình mà còn giúp sinh viên nắm được cách hoạt động, cách sử dụng Git, đặc biệt là GitHub trong quá trình học tập.

Đề tài này hướng tới việc xây dựng hệ thống giúp sinh viên dễ dàng kiểm tra và đánh giá chất lượng bài làm của mình, tạo điều kiện thuận lợi cho việc học tập và giảng dạy của sinh viên, giảng viên ngành Công nghệ thông tin. Ngoài ra, đây cũng là nền tảng giúp phát triển tư duy lập trình thi đấu, tạo nguồn lực mạnh mẽ cho các cuộc thi Olympic Tin học, ICPC hàng năm.

1.2. Mục tiêu

Nghiên cứu quy trình CI/CD và các tính năng được hỗ trợ trong GitHub Actions kết hợp với GitHub REST API và các công nghệ phù hợp để tạo ra hệ thống đánh giá lập trình tự động, hỗ trợ sinh viên, giảng viên Công nghệ thông tin trong học tập và thi đấu.

1.3. Nội dung

- Tìm hiểu về quy trình CI/CD.
- Tìm hiểu các phương pháp kiểm tra và đánh giá chất lượng mã nguồn.
- Tìm hiểu các tính năng của GitHub Actions.
- Tìm hiểu về GitHub REST API.
- Vận dụng các kiến thức trên để thiết kế và xây dựng hệ thống LMS đánh giá lập

trình tự động dựa trên nền tảng GitHub.

1.4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Quy trình CI/CD và các công cụ hỗ trợ trong GitHub Actions.
- Các phương pháp kiểm tra và đánh giá mã nguồn trong các dự án phần mềm.
- Các API của GitHub, đặc biệt là GitHub REST API, phục vụ cho việc quản lý và tích hợp hệ thống đánh giá.

Phạm vi nghiên cứu:

- Tập trung vào việc áp dụng CI/CD và GitHub Actions trong quá trình đánh giá mã nguồn tự động, hỗ trợ các bài tập lập trình của sinh viên.
- Đề tài sẽ giới hạn trong phạm vi việc sử dụng GitHub như nền tảng chính cho việc lưu trữ mã nguồn và thực thi quy trình đánh giá.
- Hệ thống được xây dựng sẽ hỗ trợ các ngôn ngữ lập trình phổ biến trong giáo dục như C, C++ và Java.

1.5. Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

- Nghiên cứu các tài liệu về quy trình CI/CD.
- Nghiên cứu các tài liệu về phương pháp kiểm tra và đánh giá chất lượng mã nguồn.
- Nghiên cứu các tài liệu về tính năng của GitHub Actions.
- Nghiên cứu các tài liệu về GitHub REST API.
- Nghiên cứu các tài liệu cần thiết về thiết kế và xây dựng website.

Phương pháp thực nghiệm:

- Xây dựng mô hình cơ sở dữ liệu và kiến trúc hệ thống.
- Cài đặt các nền tảng và công cụ cần thiết.
- Từng bước phát triển các tính năng và giao diện cho hệ thống.
- Kiểm thử và triển khai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về CI/CD

2.1.1. *Tìm hiểu về CI*

Tích hợp liên tục (Continuous Integration - CI) là một phương pháp phát triển phần mềm phổ biến nhằm cải thiện chất lượng và hiệu quả của quy trình phát triển thông qua việc tự động hóa và tích hợp mã nguồn từ các nhà phát triển vào hệ thống chính một cách liên tục và thường xuyên. CI được áp dụng rộng rãi trong các dự án phần mềm hiện đại, đặc biệt là trong các mô hình phát triển theo phương pháp Agile, nơi sự thay đổi và cập nhật mã diễn ra nhanh chóng và liên tục.

CI bắt nguồn từ việc giải quyết một trong những thách thức lớn của quá trình phát triển phần mềm: việc tích hợp mã nguồn từ nhiều nhà phát triển vào một hệ thống lớn. Khi có nhiều nhà phát triển làm việc đồng thời, khả năng xảy ra xung đột mã nguồn hoặc lỗi phát sinh do sự thay đổi của các thành phần khác nhau rất lớn. Việc tích hợp mã thủ công hoặc vào cuối một chu kỳ phát triển thường dẫn đến khủng hoảng tích hợp (Integration Hell), khi mã của các nhà phát triển xung đột nhau hoặc không hoạt động đúng như mong đợi khi hợp nhất vào hệ thống chung.

Để khắc phục tình trạng này, CI khuyến khích việc tích hợp mã thường xuyên – có thể là vài lần một ngày. Mỗi khi có thay đổi mã được đẩy lên kho mã nguồn, một quy trình tự động sẽ được kích hoạt, bao gồm việc kiểm tra và xây dựng mã, chạy các bộ kiểm thử tự động nhằm phát hiện sớm các lỗi hoặc vấn đề có thể xảy ra.

Quy trình hoạt động của CI

- Phát triển mã:* Trong một môi trường CI, các nhà phát triển viết mã và làm việc trên các nhánh phát triển riêng của họ (Feature Branches) hoặc làm việc trực tiếp trên nhánh chính (main branch). Họ thực hiện các thay đổi nhỏ và thường xuyên.

- Đẩy mã lên kho lưu trữ:* Sau khi hoàn thành một thay đổi, nhà phát triển đẩy mã của họ lên kho lưu trữ mã nguồn (thường sử dụng Git). CI hoạt động chặt chẽ với các hệ thống quản lý mã nguồn như GitHub, GitLab hoặc Bitbucket.

- Tự động hóa kiểm thử và xây dựng:* Ngay khi mã được đẩy lên, một hệ thống CI sẽ tự động kích hoạt quy trình xây dựng và kiểm thử. Hệ thống CI sẽ lấy mã từ kho lưu trữ,

xây dựng mã và chạy các bộ kiểm thử đã định nghĩa. Điều này bao gồm:

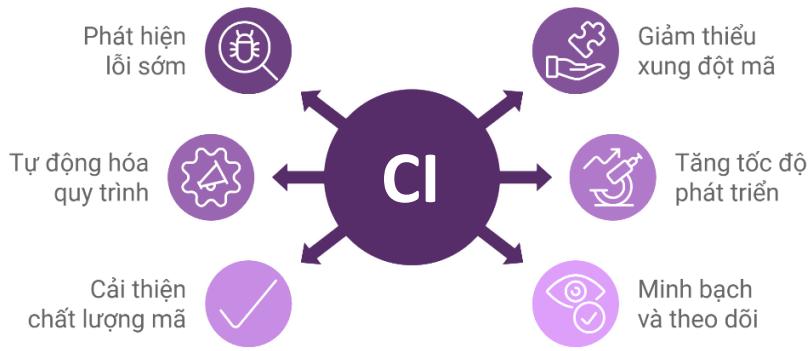
- + *Xây dựng mã nguồn*: Hệ thống CI sẽ biên dịch hoặc xây dựng mã để đảm bảo không có lỗi biên dịch, thiếu phụ thuộc hoặc vấn đề xây dựng.
- + *Kiểm thử đơn vị (Unit Tests)*: Các kiểm thử đơn vị được chạy để kiểm tra các thành phần riêng lẻ của phần mềm, đảm bảo rằng chúng hoạt động như mong đợi.
- + *Kiểm thử tích hợp (Integration Tests)*: Các kiểm thử tích hợp giúp đảm bảo rằng các thành phần khác nhau của hệ thống hoạt động tốt khi chúng được tích hợp với nhau.
- + *Kiểm thử hệ thống*: Các kiểm thử toàn bộ hệ thống được thực hiện nhằm đảm bảo rằng phần mềm hoạt động ổn định trong môi trường thực tế.

4. *Phản hồi nhanh chóng*: Nếu có lỗi xảy ra trong quá trình xây dựng hoặc kiểm thử, hệ thống CI sẽ thông báo ngay lập tức cho nhà phát triển qua email hoặc các công cụ quản lý thông báo khác như Slack, Teams. Điều này cho phép các nhà phát triển sửa lỗi sớm, khi sự thay đổi còn mới và dễ dàng xác định nguồn gốc vấn đề.

5. *Lưu trữ kết quả và báo cáo*: Kết quả của các quá trình kiểm thử, bao gồm số lượng kiểm thử đã chạy, kiểm thử nào đã thất bại và báo cáo lỗi chi tiết, được lưu trữ và cung cấp cho nhóm phát triển để xem xét. Nhiều hệ thống CI còn cung cấp các dashboard theo dõi tình trạng của các lần tích hợp gần đây.

6. *Triển khai tự động (Optional)*: Trong nhiều dự án CI, quy trình tích hợp liên tục được mở rộng thành Continuous Delivery (CD) hoặc Continuous Deployment, trong đó phần mềm sau khi xây dựng và kiểm thử thành công sẽ tự động được triển khai lên môi trường staging hoặc thậm chí là môi trường production. Điều này giúp rút ngắn thời gian từ khi thay đổi mã đến khi triển khai lên hệ thống thực tế, đồng thời giảm thiểu nguy cơ lỗi trong quá trình triển khai.

Lợi ích của Continuous Integration



- *Phát hiện lỗi sớm*: CI giúp phát hiện và sửa lỗi sớm, ngay khi mã được tích hợp. Điều này giúp tránh được việc lỗi nhỏ tích lũy thành các vấn đề lớn khó giải quyết sau này. Việc chạy kiểm thử tự động cho mỗi lần tích hợp giúp phát hiện ngay khi có lỗi mới phát sinh, giảm thiểu rủi ro cho hệ thống.

- *Giảm thiểu xung đột mã*: Khi các nhà phát triển tích hợp mã thường xuyên, CI giúp tránh việc mã của các nhà phát triển xung đột nhau hoặc không tương thích khi tích hợp vào hệ thống chính. Điều này giúp duy trì tính toàn vẹn của hệ thống và giảm thiểu công sức gỡ rối khi các lỗi liên quan đến tích hợp phát sinh.

- *Tự động hóa quy trình*: CI giúp tự động hóa các tác vụ như kiểm thử, xây dựng và triển khai. Việc này giúp tiết kiệm thời gian và công sức của các nhà phát triển, đồng thời giảm thiểu sai sót khi thực hiện các công đoạn này thủ công. Sự tự động hóa cũng giúp tăng cường tính ổn định và đáng tin cậy của quy trình phát triển.

- *Tăng tốc độ phát triển*: CI giúp tăng tốc độ phát triển bằng cách giảm thời gian phản hồi về lỗi và tăng cường sự hợp tác giữa các nhà phát triển. Khi các lỗi được phát hiện và xử lý ngay lập tức, các nhóm phát triển có thể tiếp tục công việc của mình mà không bị gián đoạn.

- *Cải thiện chất lượng phần mềm*: Khi kiểm thử được thực hiện liên tục trong suốt chu kỳ phát triển, phần mềm sẽ được kiểm tra chất lượng ở mọi giai đoạn. CI đảm bảo rằng phần mềm luôn được kiểm tra giúp cải thiện chất lượng tổng thể và giảm thiểu rủi ro lỗi khi phát hành.

- *Cải thiện tính minh bạch và khả năng theo dõi*: CI cung cấp các báo cáo chi tiết về tình trạng của các lần tích hợp, từ đó cung cấp cho nhóm phát triển một cái nhìn toàn diện về tình trạng của dự án. Điều này giúp các nhóm dễ dàng theo dõi tiến độ và phát hiện sớm

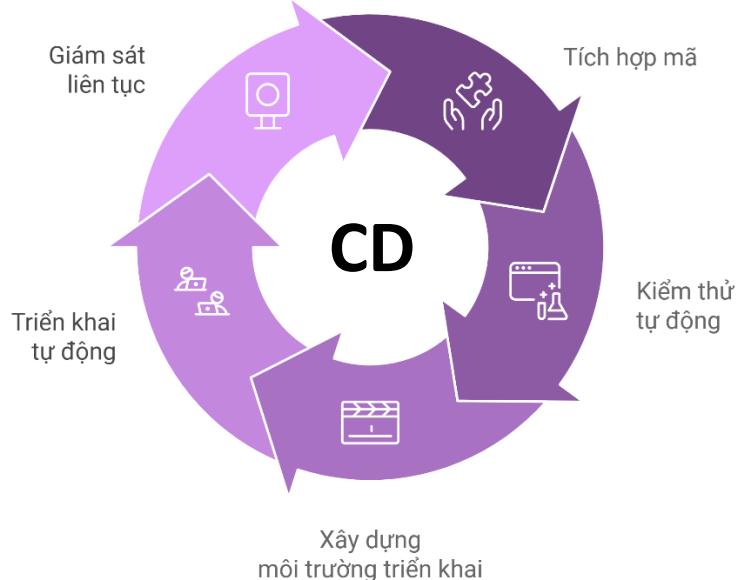
các vấn đề cần giải quyết.

CI đã trở thành một phần không thể thiếu của quy trình phát triển phần mềm hiện đại. Việc tích hợp mã liên tục không chỉ giúp phát hiện và sửa lỗi sớm mà còn cải thiện tốc độ phát triển, tính ổn định và chất lượng phần mềm tổng thể. CI không chỉ là một công cụ kỹ thuật mà còn là một phương pháp giúp tạo ra một môi trường phát triển hợp tác, nhanh chóng và minh bạch.

2.1.2. Tìm hiểu về CD

Phân phối liên tục (Continuous Delivery - CD) là một phương pháp và thực tiễn trong phát triển phần mềm nhằm đảm bảo rằng phần mềm có thể được phát hành một cách nhanh chóng, đáng tin cậy và tự động đến môi trường sản xuất hoặc các môi trường khác. CD xây dựng trên nền tảng của Continuous Integration (CI), tiếp tục mở rộng và tự động hóa quá trình triển khai phần mềm sau khi mã nguồn đã được tích hợp và kiểm thử thành công. Mục tiêu chính của CD là đảm bảo rằng các thay đổi đối với mã nguồn – từ các tính năng mới đến các bản vá lỗi nhỏ – có thể được triển khai vào môi trường sản xuất một cách an toàn và nhanh chóng.

Trong quá trình phát triển phần mềm truyền thống, việc phát hành phần mềm thường diễn ra vào cuối mỗi chu kỳ phát triển dài hạn và đi kèm với đó là quá trình kiểm thử, triển khai và đánh giá tốn nhiều thời gian. Với Continuous Delivery, thay vì đợi đến một thời điểm nhất định mới phát hành phần mềm, các nhóm phát triển có thể phát hành bất kỳ lúc nào vì mã nguồn luôn sẵn sàng để triển khai. Điều này giúp giảm thiểu thời gian giữa các lần phát hành, tạo điều kiện cho các bản cập nhật và cải tiến nhanh chóng, tăng tính linh hoạt cho doanh nghiệp trong việc đáp ứng nhu cầu của người dùng.



CD được thiết kế để tự động hóa tất cả các giai đoạn của chuỗi cung ứng phần mềm, từ quá trình tích hợp mã (Continuous Integration) đến quá trình kiểm thử và triển khai. Quy trình chung bao gồm các bước chính sau:

- *Tích hợp mã liên tục (CI)*: Mã nguồn từ nhiều nhà phát triển được liên tục tích hợp vào kho lưu trữ chung và kiểm thử tự động (unit test, integration test) được thực hiện để đảm bảo rằng không có lỗi hoặc sự cố nào phát sinh từ quá trình tích hợp.
- *Kiểm thử tự động và đảm bảo chất lượng*: Sau khi mã đã được tích hợp, quá trình kiểm thử tự động hóa sẽ được kích hoạt. Điều này bao gồm nhiều loại kiểm thử khác nhau như kiểm thử tích hợp (integration tests), kiểm thử hệ thống (system tests), kiểm thử giao diện người dùng (UI tests) và kiểm thử hiệu năng (performance tests). Mục tiêu là đảm bảo rằng mã không chỉ hoạt động đúng mà còn tương thích và an toàn trong môi trường sản xuất thực tế.
- *Xây dựng môi trường triển khai (staging environment)*: Một môi trường staging hoặc pre-production được thiết lập để mô phỏng môi trường sản xuất thực tế. Đây là nơi các tính năng mới và các bản cập nhật sẽ được kiểm thử để đảm bảo chúng hoạt động như mong đợi trước khi triển khai vào sản xuất.
- *Triển khai tự động vào các môi trường khác nhau*: Với CD, việc triển khai phần mềm vào các môi trường khác nhau (như môi trường staging, môi trường kiểm thử hoặc môi trường sản xuất) được tự động hóa hoàn toàn. Khi mã đã vượt qua tất cả các kiểm thử và

kiểm duyệt cần thiết, nó có thể được triển khai trực tiếp mà không cần sự can thiệp của con người. Điều này giúp giảm thiểu lỗi triển khai và đảm bảo sự nhất quán giữa các môi trường.

- *Giám sát và phản hồi liên tục*: Sau khi phần mềm được triển khai, quá trình giám sát tự động sẽ tiếp tục theo dõi hiệu năng và hành vi của hệ thống. Bất kỳ sự cố nào phát sinh sẽ được phát hiện và phản hồi nhanh chóng giúp đội ngũ phát triển có thể khắc phục sự cố ngay lập tức.

Continuous Delivery mang lại nhiều lợi ích to lớn cho quá trình phát triển phần mềm, cả về mặt kỹ thuật lẫn mặt kinh doanh:

- *Giảm thiểu rủi ro*: CD giúp giảm thiểu rủi ro của quá trình triển khai phần mềm bằng cách chia nhỏ các thay đổi và phát hành liên tục. Thay vì triển khai một khối lượng lớn thay đổi trong một lần, CD cho phép triển khai từng thay đổi nhỏ lẻ giúp dễ dàng xác định và sửa chữa lỗi nếu có phát sinh. Điều này cũng giúp tránh tình trạng lỗi phát sinh từ các thay đổi lớn và phức tạp.

- *Phản hồi nhanh chóng từ người dùng*: Với CD, các tính năng mới hoặc các thay đổi có thể được triển khai nhanh chóng đến người dùng. Điều này giúp doanh nghiệp thu thập phản hồi từ người dùng trong thời gian ngắn, từ đó cải thiện sản phẩm và dịch vụ dựa trên nhu cầu thực tế.

- *Tăng tính linh hoạt trong phát triển*: Continuous Delivery giúp tăng tính linh hoạt trong phát triển phần mềm. Các nhóm phát triển có thể nhanh chóng thử nghiệm các tính năng mới, loại bỏ những thay đổi không hiệu quả và liên tục điều chỉnh để đáp ứng nhu cầu của người dùng hoặc thay đổi của thị trường.

- *Cải thiện chất lượng phần mềm*: CD khuyến khích việc kiểm thử liên tục ở mọi giai đoạn của quá trình phát triển giúp phát hiện lỗi sớm và ngăn chặn sự cố trước khi chúng ảnh hưởng đến người dùng cuối. Quá trình kiểm thử tự động giúp đảm bảo rằng phần mềm luôn đáp ứng được các tiêu chuẩn chất lượng cao nhất.

- *Tăng năng suất phát triển*: Với CD, các tác vụ thủ công như triển khai và kiểm thử, được tự động hóa hoàn toàn. Điều này giúp các nhà phát triển có thể tập trung vào việc phát triển và sáng tạo thay vì phải xử lý các công việc tồn thời gian và có thể gây lỗi.

- *Rút ngắn thời gian phát hành*: CD cho phép các đội ngũ phát triển rút ngắn thời gian từ khi một tính năng mới được phát triển đến khi nó được phát hành ra ngoài thị trường. Điều này mang lại lợi thế cạnh tranh cho doanh nghiệp, đặc biệt trong các thị trường có tốc độ thay đổi nhanh chóng.

- *Tính nhất quán và đáng tin cậy*: Nhờ quá trình triển khai tự động và kiểm thử liên tục, CD giúp đảm bảo rằng phần mềm luôn được triển khai một cách nhất quán và đáng tin cậy giữa các môi trường khác nhau. Điều này giúp tránh những lỗi phát sinh do sự khác biệt giữa môi trường phát triển, kiểm thử và sản xuất.

Continuous Delivery (CD) là một phần quan trọng của quy trình phát triển phần mềm hiện đại giúp tăng cường hiệu quả và tính linh hoạt của việc phát hành phần mềm. Bằng cách tự động hóa quá trình kiểm thử và triển khai, CD giúp giảm thiểu rủi ro, tăng tốc độ phản hồi từ người dùng, cải thiện chất lượng phần mềm và mang lại lợi ích to lớn cho cả các nhóm phát triển và doanh nghiệp. Việc triển khai CD không chỉ yêu cầu các công cụ phù hợp mà còn cần một chiến lược phát triển phần mềm toàn diện, trong đó nhấn mạnh việc kiểm thử tự động và hợp tác liên tục giữa các thành viên trong nhóm.

2.1.3. Quy trình CI/CD

Quy trình CI/CD (Continuous Integration và Continuous Delivery/Deployment) là một phương pháp phát triển phần mềm hiện đại giúp tối ưu hóa và tự động hóa quá trình phát triển, kiểm thử và triển khai phần mềm. CI/CD giúp giảm thiểu thời gian phát triển, cải thiện chất lượng phần mềm và đảm bảo các bản phát hành nhanh chóng và ổn định hơn. Quy trình này được chia thành hai giai đoạn chính: Continuous Integration (CI) và Continuous Delivery/Deployment (CD).

CI/CD mang lại nhiều lợi ích đáng kể cho quy trình phát triển phần mềm, đặc biệt trong các dự án có chu kỳ phát triển liên tục và yêu cầu sự linh hoạt cao. Dưới đây là những lợi ích chính:

- *Tăng cường tốc độ phát triển*: Với CI/CD, các bước kiểm thử và triển khai phần mềm đều được tự động hóa giúp giảm thiểu thời gian chờ đợi giữa các giai đoạn phát triển. Điều này cho phép các nhà phát triển nhanh chóng đưa mã mới vào hệ thống và triển khai các tính năng mới hoặc sửa lỗi với tốc độ nhanh hơn. Ngoài ra, việc triển khai các thay đổi nhỏ thường xuyên thay vì gộp lại một bản cập nhật lớn giúp giảm bớt rủi ro và tăng tính linh

hoạt.

- *Phát hiện lỗi sớm*: CI/CD cho phép phát hiện lỗi ngay từ giai đoạn đầu khi mã mới được đẩy lên hệ thống. Khi có sự cố xảy ra, các bài kiểm thử tự động sẽ giúp phát hiện lỗi kịp thời và thông báo cho đội ngũ phát triển. Điều này giúp giảm thời gian sửa chữa lỗi và ngăn chặn lỗi lây lan sang các phần khác của hệ thống. Bằng cách chạy các bài kiểm thử tự động trong suốt quá trình phát triển, các vấn đề về tính năng, hiệu suất hoặc bảo mật có thể được phát hiện và giải quyết sớm hơn.

- *Cải thiện chất lượng phần mềm*: Nhờ vào các bài kiểm thử tự động và quy trình kiểm thử liên tục, phần mềm được duy trì ở trạng thái ổn định và chất lượng cao hơn. CI/CD đảm bảo rằng mỗi thay đổi đều được kiểm tra kỹ lưỡng trước khi triển khai vào môi trường sản xuất. Kiểm thử tự động không chỉ giúp phát hiện lỗi mà còn cung cấp thông tin về mức độ ảnh hưởng của mỗi thay đổi giúp giảm thiểu tác động tiêu cực và giữ cho hệ thống luôn ổn định.

- *Giảm thiểu rủi ro*: Continuous Deployment (triển khai liên tục) giúp giảm thiểu rủi ro bằng cách triển khai các thay đổi nhỏ và thường xuyên thay vì gộp lại nhiều thay đổi lớn. Điều này làm giảm khả năng các lỗi lớn xuất hiện và dễ dàng xác định nguyên nhân nếu có sự cố xảy ra. Ngoài ra, các quy trình rollback (khôi phục phiên bản trước) được tích hợp sẵn trong nhiều công cụ CI/CD giúp đảm bảo rằng nếu có lỗi nghiêm trọng, hệ thống có thể quay trở lại phiên bản hoạt động ổn định trước đó mà không gây gián đoạn dịch vụ.

Việc triển khai CI/CD đòi hỏi sự hỗ trợ của các công cụ tự động hóa để quản lý quá trình xây dựng, kiểm thử và triển khai phần mềm. Hiện nay có nhiều công cụ khác nhau, từ mã nguồn mở đến các dịch vụ đám mây, phục vụ cho các nhu cầu khác nhau của dự án. Một số công cụ phổ biến:

- *Jenkins*: Jenkins là một công cụ mã nguồn mở hàng đầu trong lĩnh vực CI/CD, nổi tiếng với khả năng mở rộng và tính tùy chỉnh cao. Jenkins hỗ trợ các dự án quy mô từ nhỏ đến lớn, cung cấp khả năng tích hợp với hàng nghìn plugin giúp tự động hóa gần như mọi bước trong quy trình phát triển phần mềm. Jenkins được biết đến với khả năng hỗ trợ nhiều loại môi trường xây dựng và triển khai khác nhau, bao gồm Docker, Kubernetes và các dịch vụ đám mây như AWS, Azure. Tuy nhiên, do tính linh hoạt cao, Jenkins yêu cầu cấu hình phức tạp hơn và thường cần đội ngũ kỹ thuật giàu kinh nghiệm để quản lý.

- *GitLab CI/CD*: GitLab CI/CD là một tính năng được tích hợp sẵn trong GitLab, một nền tảng quản lý mã nguồn phổ biến. Với GitLab CI/CD, các nhà phát triển có thể dễ dàng cấu hình quy trình xây dựng, kiểm thử và triển khai phần mềm mà không cần sử dụng các công cụ bên ngoài. GitLab CI/CD cung cấp nhiều tính năng như pipeline trực quan, quản lý các stages (giai đoạn) của pipeline và hỗ trợ nhiều môi trường triển khai khác nhau. Tích hợp trực tiếp với hệ thống quản lý mã nguồn, GitLab CI/CD rất thuận tiện cho việc theo dõi tiến độ phát triển phần mềm. Một lợi thế lớn của GitLab CI/CD là nó có thể triển khai ngay cả trên cơ sở hạ tầng của tổ chức (self-hosted), cho phép kiểm soát tốt hơn về bảo mật và tài nguyên.

- *GitHub Actions*: GitHub Actions là một công cụ CI/CD tích hợp với GitHub giúp các nhà phát triển tự động hóa quy trình làm việc của họ từ giai đoạn phát triển đến triển khai. Nó được thiết kế để hoạt động liền mạch với GitHub repository, cho phép dễ dàng tạo ra các workflows (quy trình làm việc) dựa trên các sự kiện như push code, mở pull request hay release phần mềm. GitHub Actions có lợi thế là không yêu cầu cấu hình phức tạp, với các template sẵn có và dễ sử dụng, đặc biệt phù hợp với các dự án có quy mô vừa và nhỏ. Công cụ này cung cấp các workflows đa dạng, từ kiểm thử đơn giản đến các quy trình tích hợp phức tạp. Một điểm mạnh khác là khả năng mở rộng và tích hợp với các dịch vụ đám mây, cho phép triển khai dễ dàng lên các nền tảng như AWS, Azure hoặc Google Cloud.

- *CircleCI*: CircleCI là một dịch vụ CI/CD đám mây được ưa chuộng nhờ khả năng tự động hóa quy trình xây dựng, kiểm thử và triển khai với độ tin cậy cao. Công cụ này cho phép nhà phát triển dễ dàng thiết lập pipeline phức tạp và chạy nhiều tác vụ song song giúp giảm thời gian xây dựng và kiểm thử. CircleCI cung cấp khả năng tích hợp tốt với Docker và Kubernetes và có thể cấu hình dễ dàng để chạy trên các dịch vụ đám mây lớn. Ngoài ra, CircleCI cung cấp một giao diện người dùng đơn giản nhưng mạnh mẽ, cho phép theo dõi quá trình pipeline và kết quả kiểm thử một cách trực quan.

- *Travis CI*: Travis CI là một công cụ CI/CD dễ sử dụng, đặc biệt phổ biến trong cộng đồng mã nguồn mở, với khả năng tích hợp chặt chẽ với GitHub. Travis CI nổi bật với quy trình thiết lập đơn giản, chỉ yêu cầu một file cấu hình .travis.yml để định nghĩa các bước trong pipeline. Travis CI hỗ trợ nhiều ngôn ngữ lập trình và nền tảng khác nhau, đồng thời cung cấp cả dịch vụ miễn phí cho các dự án mã nguồn mở. Đối với các dự án có quy mô lớn hơn hoặc yêu cầu độ tùy chỉnh cao, Travis CI cũng cung cấp các gói dịch vụ cao cấp

có thể tự host hoặc chạy trên cơ sở hạ tầng của tổ chức.

2.2. Các phương pháp kiểm tra và đánh giá chất lượng mã nguồn

Trong phát triển phần mềm, việc kiểm tra và đánh giá chất lượng mã nguồn là một bước quan trọng để đảm bảo tính ổn định, hiệu suất và khả năng mở rộng của sản phẩm. Mã nguồn chất lượng cao giúp giảm thiểu lỗi, tối ưu hóa hiệu suất và cải thiện khả năng bảo trì. Có nhiều phương pháp và công cụ hỗ trợ kiểm tra và đánh giá chất lượng mã nguồn, từ kiểm thử đơn vị (unit testing) đến các công cụ phân tích tĩnh (static analysis) và kiểm thử tự động (automated testing),...

2.2.1. Kiểm thử đơn vị

Kiểm thử đơn vị (Unit Testing) là một phương pháp kiểm thử phần mềm trong đó từng thành phần nhỏ nhất của chương trình, gọi là "đơn vị" (unit), được kiểm tra một cách riêng lẻ. Một đơn vị thường là một hàm, phương thức hoặc một đoạn mã riêng biệt có chức năng cụ thể trong ứng dụng. Kiểm thử đơn vị là bước đầu tiên trong quy trình kiểm thử và giúp phát hiện lỗi ngay từ giai đoạn sớm nhất trong phát triển phần mềm.

Mục tiêu của kiểm thử đơn vị là đảm bảo rằng mỗi đơn vị nhỏ của chương trình hoạt động đúng theo yêu cầu và không có lỗi. Các nhà phát triển sẽ viết các bài kiểm thử nhỏ, thường gọi là "test case", để xác minh rằng chức năng của từng đơn vị hoạt động chính xác dưới các điều kiện khác nhau, bao gồm cả các trường hợp đặc biệt.

Lợi ích của kiểm thử đơn vị:

- *Phát hiện lỗi sớm*: Kiểm thử đơn vị cho phép phát hiện lỗi ngay trong giai đoạn đầu của quá trình phát triển phần mềm giúp giảm thiểu chi phí và thời gian sửa chữa so với việc phát hiện lỗi sau này.
- *Tài liệu hóa mã nguồn*: Các bài kiểm thử đơn vị không chỉ giúp kiểm tra mã mà còn đóng vai trò như một loại tài liệu. Khi các nhà phát triển khác xem lại mã, các bài kiểm thử sẽ giúp họ hiểu rõ cách các hàm hoặc phương thức hoạt động.
- *Giảm thiểu lỗi liên quan*: Thực hiện kiểm thử đơn vị đều đặn giúp giảm thiểu rủi ro các lỗi nhỏ có thể dẫn đến sự cố lớn hơn trong ứng dụng khi các thành phần khác tương tác với nhau.
- *Tăng cường sự tin tưởng trong việc refactor mã*: Khi một đoạn mã có kiểm thử đơn vị

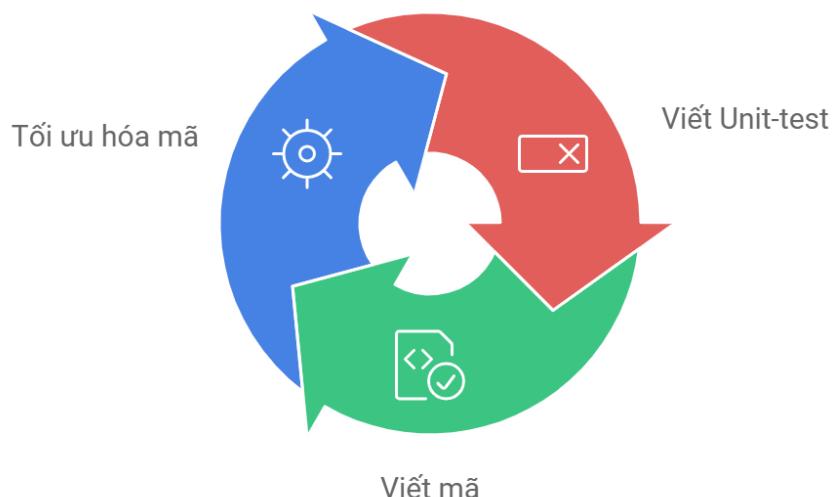
tốt, các nhà phát triển có thể dễ dàng refactor hoặc thay đổi mã mà không sợ làm hỏng chức năng hiện có, vì họ đã có bài kiểm thử để đảm bảo mọi thứ hoạt động đúng.

- *Cải thiện chất lượng mã nguồn*: Kiểm thử đơn vị giúp nhà phát triển nghĩ kỹ hơn về thiết kế và cấu trúc mã nguồn, dẫn đến mã có chất lượng tốt hơn và dễ bảo trì.

Có nhiều công cụ và framework hỗ trợ kiểm thử đơn vị cho các ngôn ngữ lập trình khác nhau. Một số framework phổ biến bao gồm:

- + *JUnit*: Framework kiểm thử đơn vị phổ biến cho Java.
- + *NUnit*: Framework kiểm thử đơn vị cho .NET.
- + *PyTest*: Framework kiểm thử đơn vị cho Python.
- + *Mocha* và *Jest*: Các framework kiểm thử đơn vị phổ biến cho JavaScript/TypeScript.
- + *RSpec*: Framework kiểm thử cho Ruby.

Test Driven Development (TDD) là một phương pháp phát triển phần mềm trong đó các bài kiểm thử đơn vị được viết trước khi mã được phát triển. Quy trình TDD thường tuân theo ba bước chính:



Hình 2.1. Quy trình của Test Driven Development (TDD)

- *Viết bài kiểm thử*: Viết một bài kiểm thử đơn vị cho chức năng mới mà chúng ta muốn phát triển. Ban đầu, bài kiểm thử sẽ thất bại vì mã chưa được triển khai.

- *Viết mã để kiểm thử thành công*: Viết đủ mã để bài kiểm thử đơn vị có thể thành công.

- *Tối ưu hóa mã*: Sau khi bài kiểm thử đơn vị thành công, chúng ta có thể refactor và

cải tiến mã mà không lo lắng về việc phá vỡ chức năng hiện có.

TDD giúp tăng cường tính chắc chắn của mã nguồn, giảm lỗi và thúc đẩy việc phát triển phần mềm có chất lượng cao.

Mặc dù kiểm thử đơn vị rất hữu ích, nó cũng có một số hạn chế:

- *Không phát hiện được lỗi tích hợp*: Kiểm thử đơn vị chỉ kiểm tra các thành phần riêng lẻ và không đảm bảo rằng các thành phần đó sẽ hoạt động tốt khi tích hợp vào hệ thống.

- *Phụ thuộc vào chất lượng test case*: Nếu các test case không đủ toàn diện, kiểm thử đơn vị sẽ không phát hiện được hết các lỗi tiềm ẩn.

- *Tốn thời gian để viết test case*: Việc viết các test case cho từng thành phần mã nguồn có thể tốn nhiều thời gian, đặc biệt là với các dự án lớn.

2.2.2. Kiểm thử tích hợp

Kiểm thử tích hợp (Integration Testing) là một phương pháp kiểm thử phần mềm nhằm kiểm tra sự tương tác giữa các thành phần, mô-đun hoặc hệ thống riêng lẻ khi chúng được tích hợp lại với nhau. Mục tiêu chính của kiểm thử tích hợp là xác minh rằng các mô-đun hoạt động đúng đắn khi kết hợp lại, đảm bảo rằng sự tích hợp không tạo ra lỗi trong hệ thống tổng thể.

Trong một ứng dụng, các thành phần hoặc mô-đun riêng biệt thường được phát triển và kiểm thử đơn lẻ thông qua kiểm thử đơn vị. Tuy nhiên, khi các mô-đun này được tích hợp lại với nhau, có thể xuất hiện lỗi do sự không tương thích hoặc sai sót trong giao tiếp giữa các thành phần. Kiểm thử tích hợp giúp phát hiện các lỗi liên quan đến sự tương tác giữa các mô-đun, ví dụ như:

- Sai lệch trong cách các mô-đun trao đổi dữ liệu.
- Các vấn đề về giao tiếp giữa mô-đun và cơ sở dữ liệu.
- Các lỗi liên quan đến API hoặc dịch vụ web khi chúng tương tác với nhau.
- Sự không tương thích giữa các giao thức hoặc định dạng dữ liệu.

Lợi ích của kiểm thử tích hợp:

- *Phát hiện sớm các lỗi tích hợp*: Các lỗi thường xuất hiện khi các mô-đun, đặc biệt là những mô-đun do các nhóm khác nhau phát triển, được kết hợp với nhau. Kiểm thử tích

hợp giúp phát hiện và khắc phục lỗi trong quá trình tích hợp, trước khi chúng gây ra các vấn đề lớn hơn ở các giai đoạn sau.

- *Đảm bảo tính tương thích*: Kiểm thử tích hợp xác minh rằng các mô-đun khác nhau hoạt động tốt khi chúng tương tác với nhau, đảm bảo rằng giao tiếp giữa các thành phần hệ thống diễn ra mượt mà và chính xác.

- *Cải thiện sự ổn định của hệ thống*: Bằng cách thực hiện kiểm thử tích hợp thường xuyên, các nhà phát triển có thể đảm bảo rằng toàn bộ hệ thống duy trì tính ổn định và hoạt động đúng khi có sự thay đổi hoặc bổ sung các thành phần mới.

- *Giảm thiểu rủi ro*: Thực hiện kiểm thử tích hợp sớm giúp giảm thiểu rủi ro của các lỗi tích hợp nghiêm trọng ảnh hưởng đến chất lượng hệ thống tổng thể hoặc gây ra sự cố trong môi trường sản xuất.

Kiểm thử tích hợp có thể được thực hiện theo nhiều cách khác nhau, tùy thuộc vào cách các thành phần được tích hợp và độ phức tạp của hệ thống. Dưới đây là một số phương pháp kiểm thử tích hợp phổ biến:

- *Big Bang Integration Testing (Kiểm thử tích hợp toàn bộ)*: Trong phương pháp này, tất cả các mô-đun hoặc thành phần của hệ thống được tích hợp cùng lúc và kiểm thử như một hệ thống hoàn chỉnh.

- *Incremental Integration Testing (Kiểm thử tích hợp gia tăng)*: Trong phương pháp này, các mô-đun được tích hợp dần dần, từng phần một và mỗi phần sẽ được kiểm thử sau khi tích hợp.

- *Hybrid Integration Testing (Kiểm thử tích hợp lai)*: Phương pháp này kết hợp giữa kiểm thử tích hợp từ trên xuống và từ dưới lên. Các mô-đun cấp cao và cấp thấp có thể được tích hợp và kiểm thử đồng thời.

- *Sandwich Integration Testing (Kiểm thử tích hợp kiểu sandwich)*: Đây là một dạng của kiểm thử tích hợp lai, trong đó các mô-đun cấp cao và cấp thấp được kiểm thử đồng thời, trong khi các mô-đun ở giữa được tích hợp và kiểm thử cuối cùng.

2.2.3. Kiểm thử chức năng

Kiểm thử chức năng (Functional Testing) là một kỹ thuật kiểm thử phần mềm nhằm xác nhận rằng các tính năng hoặc chức năng của một hệ thống hoạt động đúng như yêu cầu

đã xác định. Mục tiêu của kiểm thử chức năng là đảm bảo rằng hệ thống đáp ứng được các yêu cầu chức năng của người dùng và các quy trình kinh doanh.

Kiểm thử chức năng kiểm tra các hành vi bên ngoài của hệ thống bằng cách cung cấp các đầu vào cụ thể và so sánh với kết quả đầu ra mong đợi mà không quan tâm đến cách hệ thống thực hiện chúng bên trong. Cụ thể, kiểm thử chức năng tập trung vào các yếu tố sau:

- *Xác nhận các tính năng của hệ thống*: Đảm bảo rằng tất cả các tính năng của ứng dụng hoạt động đúng với mô tả và yêu cầu.

- *Kiểm tra sự tương tác của người dùng với hệ thống*: Xác minh rằng giao diện người dùng (UI) hoạt động chính xác và đáp ứng đúng các thao tác của người dùng.

- *Đảm bảo tính chính xác của đầu ra*: Kiểm tra rằng kết quả đầu ra từ hệ thống chính xác và phù hợp với các thông số đầu vào đã cung cấp.

Quy trình kiểm thử chức năng thường bao gồm các bước sau:

Bước 1. Phân tích yêu cầu: Đầu tiên, nhóm kiểm thử sẽ phân tích các yêu cầu chức năng của hệ thống để hiểu rõ cách hệ thống nên hoạt động và các tính năng cần được kiểm tra. Đây là giai đoạn rất quan trọng để đảm bảo rằng các bài kiểm thử được thiết kế đúng với yêu cầu của người dùng.

Bước 2. Xây dựng các trường hợp kiểm thử (Test case): Dựa trên các yêu cầu chức năng, các trường hợp kiểm thử sẽ được viết ra. Mỗi trường hợp kiểm thử sẽ mô tả các bước cần thực hiện, dữ liệu đầu vào và kết quả đầu ra mong đợi.

Bước 3. Chuẩn bị môi trường kiểm thử: Xây dựng môi trường kiểm thử với các công cụ, cơ sở dữ liệu và hệ thống cần thiết để thực hiện các bài kiểm thử.

Bước 4. Thực hiện kiểm thử: Tiến hành chạy các trường hợp kiểm thử theo kế hoạch, cung cấp các đầu vào và kiểm tra kết quả so với mong đợi.

Bước 5. So sánh kết quả: Kết quả đầu ra từ các bài kiểm thử sẽ được so sánh với kết quả mong đợi để xác định xem hệ thống có hoạt động đúng hay không.

Bước 6. Báo cáo lỗi: Nếu phát hiện lỗi trong quá trình kiểm thử, các lỗi sẽ được ghi lại và báo cáo cho nhóm phát triển để sửa chữa. Sau đó, quá trình kiểm thử lại (retesting) sẽ

được thực hiện để đảm bảo lỗi đã được khắc phục.

Ưu điểm của kiểm thử chức năng:

- *Dảm bảo tính đúng đắn của hệ thống*: Kiểm thử chức năng giúp xác minh rằng hệ thống hoạt động theo đúng yêu cầu đã xác định, đảm bảo rằng người dùng có thể sử dụng ứng dụng một cách hiệu quả.

- *Dễ dàng phát hiện lỗi*: Phương pháp này giúp dễ dàng phát hiện các lỗi liên quan đến logic nghiệp vụ, đặc biệt là các lỗi mà người dùng có thể gặp phải trong quá trình tương tác với hệ thống.

- *Kiểm tra các kịch bản thực tế*: Kiểm thử chức năng tập trung vào việc kiểm tra các kịch bản sử dụng thực tế, điều này giúp đảm bảo rằng ứng dụng hoạt động như mong đợi khi được triển khai.

2.2.4. Kiểm thử hồi quy

Kiểm thử hồi quy (Regression Testing) là một kỹ thuật kiểm thử phần mềm được thực hiện để đảm bảo rằng những thay đổi trong mã nguồn của hệ thống (như sửa lỗi, cập nhật tính năng hoặc tối ưu hóa) không gây ra lỗi mới hoặc làm ảnh hưởng đến các phần chức năng khác đã hoạt động đúng trước đó.

Quy trình thực hiện kiểm thử hồi quy:

- *Xác định các trường hợp kiểm thử cần hồi quy*: Lựa chọn các trường hợp kiểm thử cần được thực hiện lại. Không phải tất cả các bài kiểm thử đều cần phải hồi quy mà chỉ những phần liên quan đến thay đổi trong mã nguồn hoặc các chức năng có liên quan đến sự thay đổi đó.

- *Ưu tiên các trường hợp kiểm thử*: Xác định thứ tự ưu tiên của các trường hợp kiểm thử hồi quy. Các trường hợp liên quan trực tiếp đến thay đổi hoặc có khả năng ảnh hưởng cao sẽ được ưu tiên kiểm thử trước.

- *Thực hiện kiểm thử*: Tiến hành thực hiện các trường hợp kiểm thử đã được lựa chọn. Nếu là kiểm thử tự động, các kịch bản kiểm thử tự động sẽ được chạy trên hệ thống.

- *So sánh kết quả*: So sánh kết quả kiểm thử với kết quả mong đợi để phát hiện lỗi. Nếu có sự khác biệt, lỗi sẽ được báo cáo và sửa chữa.

- *Lặp lại quá trình*: Sau khi lỗi được sửa, quy trình kiểm thử hồi quy lại được thực hiện để đảm bảo rằng các lỗi đã được khắc phục và không có lỗi mới phát sinh.

Lợi ích của kiểm thử hồi quy:

- *Phát hiện lỗi sớm*: Kiểm thử hồi quy giúp phát hiện các lỗi mới phát sinh sau khi có thay đổi trong hệ thống, từ đó giúp giảm thiểu rủi ro lỗi xảy ra ở giai đoạn triển khai sản phẩm.

- *Đảm bảo tính nhất quán*: Kiểm thử hồi quy đảm bảo rằng tất cả các chức năng của hệ thống vẫn hoạt động như mong đợi sau mỗi lần cập nhật hoặc thay đổi.

- *Tăng độ tin cậy của sản phẩm*: Việc liên tục thực hiện kiểm thử hồi quy giúp nâng cao độ tin cậy của hệ thống, giảm thiểu các lỗi tiềm ẩn và cải thiện chất lượng sản phẩm.

- *Hỗ trợ phát triển liên tục*: Trong các mô hình phát triển Agile hoặc DevOps, kiểm thử hồi quy là yếu tố quan trọng giúp đảm bảo tính ổn định của sản phẩm trong suốt quá trình phát triển liên tục.

2.2.5. Kiểm thử bảo mật

Kiểm thử bảo mật (Security Testing) là một quá trình kiểm thử phần mềm được thực hiện nhằm phát hiện các lỗ hổng, điểm yếu và rủi ro bảo mật tiềm ẩn trong ứng dụng hoặc hệ thống. Mục đích của kiểm thử bảo mật là đảm bảo rằng phần mềm được bảo vệ khỏi các cuộc tấn công từ bên ngoài cũng như bên trong, đồng thời bảo vệ tính toàn vẹn, tính bí mật và tính sẵn sàng của dữ liệu.

Mục tiêu chính của kiểm thử bảo mật là:

- *Xác định lỗ hổng bảo mật*: Phát hiện các lỗ hổng hoặc điểm yếu trong hệ thống có thể bị khai thác bởi hacker hoặc kẻ tấn công.

- *Đảm bảo tính toàn vẹn dữ liệu*: Bảo vệ dữ liệu khỏi sự sửa đổi không được phép hoặc bị thay đổi bởi các tác nhân độc hại.

- *Bảo vệ tính bí mật*: Ngăn chặn việc truy cập trái phép vào thông tin nhạy cảm hoặc dữ liệu cá nhân.

- *Bảo vệ tính sẵn sàng của hệ thống*: Đảm bảo rằng hệ thống không bị gián đoạn hoặc bị từ chối dịch vụ (DoS) bởi các cuộc tấn công.

- *Kiểm tra tuân thủ*: Đảm bảo rằng ứng dụng tuân thủ các quy định và tiêu chuẩn bảo mật (ví dụ như GDPR, HIPAA, PCI-DSS).

Kiểm thử bảo mật thường được phân loại dựa trên phương thức kiểm thử và loại tấn công được kiểm tra. Một số loại kiểm thử bảo mật phổ biến bao gồm:

- *Kiểm thử xâm nhập (Penetration Testing)*: là một hình thức kiểm thử bảo mật giả lập các cuộc tấn công từ bên ngoài để đánh giá tính bảo mật của hệ thống.

- *Kiểm thử hộp trắng (White-box Testing)*: là kiểm thử bảo mật với quyền truy cập đầy đủ vào mã nguồn và kiến trúc của hệ thống.

- *Kiểm thử hộp đen (Black-box Testing)*: là phương pháp kiểm thử bảo mật mà tester không có quyền truy cập vào mã nguồn hoặc kiến trúc hệ thống, chỉ kiểm thử từ góc nhìn của người dùng cuối hoặc hacker.

- *Kiểm thử hộp xám (Gray-box Testing)*: Gray-box Testing kết hợp giữa kiểm thử hộp trắng và hộp đen, trong đó tester có một phần kiến thức về hệ thống nội bộ nhưng không đầy đủ.

2.2.6. Kiểm thử hiệu suất

Kiểm thử hiệu suất (Performance Testing) là quá trình kiểm tra phần mềm nhằm đánh giá cách mà hệ thống, ứng dụng hoặc phần mềm hoạt động dưới một khối lượng công việc cụ thể. Mục tiêu chính là xác định tính ổn định, độ tin cậy và khả năng mở rộng của hệ thống khi phải xử lý khối lượng truy cập hoặc dữ liệu lớn.

Mục tiêu của kiểm thử hiệu suất là:

- *Đo lường tốc độ*: Xác định thời gian phản hồi của hệ thống cho các tác vụ cụ thể, chẳng hạn như tải một trang web, truy xuất cơ sở dữ liệu, hay xử lý giao dịch.

- *Xác định khả năng chịu tải*: Kiểm tra hệ thống để xem nó có thể xử lý được bao nhiêu người dùng hoặc yêu cầu đồng thời mà không ảnh hưởng đến hiệu suất.

- *Đánh giá tính mở rộng*: Đo lường hiệu suất của hệ thống khi khối lượng công việc tăng lên và đánh giá khả năng mở rộng mà không bị suy giảm chất lượng.

- *Kiểm tra tính ổn định*: Đảm bảo hệ thống có thể chạy liên tục trong thời gian dài mà không bị crash hoặc gặp sự cố hiệu suất, ngay cả dưới tải cao.

Có nhiều loại kiểm thử hiệu suất, mỗi loại tập trung vào các khía cạnh khác nhau của hiệu suất hệ thống:

- *Kiểm thử tải (Load Testing)*: là quá trình kiểm tra hệ thống bằng cách đặt nó dưới một tải công việc cụ thể (ví dụ: một lượng người dùng nhất định hoặc số lượng yêu cầu trong khoảng thời gian xác định).
- *Kiểm thử sức chịu đựng (Stress Testing)*: là quá trình kiểm tra hệ thống bằng cách đưa nó vào tình trạng quá tải nhằm đánh giá khả năng chịu đựng của hệ thống khi bị áp lực lớn hơn mức thông thường.
- *Kiểm thử tải đột biến (Spike Testing)*: là kiểm thử hiệu suất bằng cách tăng đột ngột tải lên hệ thống trong một khoảng thời gian ngắn, nhằm kiểm tra khả năng hệ thống phản ứng với các tải đột biến.
- *Kiểm thử độ bền (Endurance Testing)*: là kiểm thử hệ thống trong một khoảng thời gian dài dưới tải bình thường hoặc tải lớn để kiểm tra tính ổn định.
- *Kiểm thử khả năng mở rộng (Scalability Testing)*: Scalability Testing đánh giá khả năng của hệ thống khi khôi phục công việc hoặc số lượng người dùng tăng lên, nhằm xác định cách hệ thống phản ứng và thích ứng với sự tăng trưởng về tài nguyên.

2.3. Tìm hiểu về GitHub Actions

GitHub Actions là một công cụ mạnh mẽ được tích hợp sẵn trong GitHub, cho phép các nhà phát triển tự động hóa các quy trình phát triển phần mềm như kiểm thử, xây dựng, triển khai ứng dụng và quản lý CI/CD (Continuous Integration và Continuous Delivery). Bằng cách tận dụng GitHub Actions, các nhóm phát triển có thể dễ dàng tạo ra các workflow tùy chỉnh cho dự án của mình mà không cần phải dựa vào các dịch vụ CI/CD bên ngoài.^[4]

GitHub Actions được giới thiệu lần đầu vào năm 2018 và nhanh chóng trở thành một công cụ phổ biến trong cộng đồng phát triển phần mềm nhờ sự tích hợp chặt chẽ với nền tảng GitHub và linh hoạt trong việc tự động hóa các quy trình công việc. GitHub Actions cho phép các nhà phát triển định nghĩa các hành động tự động (action) khi xảy ra một sự kiện nhất định (event) trong repository, chẳng hạn như khi có một commit mới, tạo pull request hoặc khi có một bản phát hành được xuất bản.^[2]

Mỗi workflow (quy trình công việc) trong GitHub Actions bao gồm các bước cụ thể và các action cần thực hiện. Các workflow này được định nghĩa thông qua các file YAML và được lưu trữ trực tiếp trong repository của dự án (.github/workflow).^[4]

Một workflow trong GitHub Actions thường bao gồm các thành phần chính sau:

- *Event (Sự kiện)*: Là các tác nhân kích hoạt workflow, ví dụ như push, pull_request hoặc schedule (theo lịch trình).

- *Job (Công việc)*: Một workflow có thể bao gồm một hoặc nhiều job, mỗi job đại diện cho một chuỗi các bước cần thực hiện. Các job có thể chạy song song hoặc tuân tự.

- *Step (Bước)*: Mỗi job bao gồm một hoặc nhiều step. Một step có thể thực hiện một script hoặc sử dụng một action có sẵn để thực hiện một tác vụ cụ thể.

- *Action (Hành động)*: Đây là các thành phần có thể tái sử dụng, thực hiện các tác vụ đơn giản như chạy một lệnh shell, thiết lập môi trường hoặc triển khai ứng dụng. GitHub Actions có một kho lưu trữ các action có sẵn mà người dùng có thể sử dụng.

Các tính năng nổi bật của GitHub Actions:

- *Tích hợp sẵn với GitHub*: GitHub Actions được tích hợp trực tiếp vào nền tảng GitHub, điều này có nghĩa là người dùng có thể quản lý toàn bộ quy trình CI/CD ngay trong repository của mình mà không cần thêm bất kỳ dịch vụ bên ngoài nào.

- *Hỗ trợ đa nền tảng*: GitHub Actions hỗ trợ nhiều môi trường khác nhau như Linux, macOS và Windows. Điều này giúp các nhà phát triển dễ dàng kiểm thử ứng dụng trên nhiều hệ điều hành khác nhau.

- *Khả năng mở rộng với Action Marketplace*: GitHub cung cấp một marketplace với hàng ngàn action được phát triển sẵn giúp người dùng dễ dàng tích hợp các tác vụ phổ biến như triển khai, kiểm thử, kiểm tra bảo mật vào workflow của mình mà không cần viết mã từ đầu.

- *Khả năng tùy chỉnh mạnh mẽ*: GitHub Actions cho phép các nhà phát triển tạo ra các workflow rất linh hoạt với các điều kiện cụ thể. Ví dụ, một workflow có thể chỉ chạy khi một loại file nhất định thay đổi hoặc khi có tag mới được tạo.

- *Bảo mật và kiểm soát*: GitHub Actions cho phép quản lý các secrets (bí mật) như

khóa API, thông tin đăng nhập, để bảo vệ các thông tin nhạy cảm khi thực hiện các quy trình CI/CD. Người dùng cũng có thể kiểm soát chi tiết các quyền truy cập vào workflow.

GitHub Actions được sử dụng rộng rãi trong các quy trình Continuous Integration (CI) và Continuous Delivery (CD) nhờ khả năng tự động hóa linh hoạt. Các nhà phát triển có thể thiết lập CI/CD pipeline một cách dễ dàng và nhanh chóng.^[4]

- *Continuous Integration (CI)*: GitHub Actions tự động kiểm thử mã nguồn khi có sự thay đổi, đảm bảo rằng mỗi lần thay đổi đều không phá vỡ tính năng hiện có. CI workflow thường bao gồm các bước như xây dựng ứng dụng, kiểm thử đơn vị, kiểm tra kiểu mã (linting) và chạy các bài kiểm thử tích hợp.

- *Continuous Delivery (CD)*: Với GitHub Actions, các thay đổi đã được kiểm thử có thể được tự động triển khai lên môi trường staging hoặc production. Workflow CD có thể bao gồm việc đóng gói ứng dụng, triển khai lên dịch vụ đám mây (như AWS, Azure hoặc Google Cloud) và gửi thông báo về trạng thái triển khai.

Lợi ích của GitHub Actions:

- *Tự động hóa liền mạch*: GitHub Actions loại bỏ nhu cầu về các dịch vụ CI/CD bên ngoài giúp các nhà phát triển quản lý quy trình phát triển phần mềm một cách trọn vẹn và hiệu quả ngay trong GitHub.

- *Tiết kiệm thời gian và nguồn lực*: Với khả năng tự động hóa từ kiểm thử đến triển khai, GitHub Actions giúp giảm thiểu công sức thủ công và tối ưu hóa quy trình phát triển giúp các nhóm tập trung hơn vào việc cải tiến và phát triển sản phẩm.

- *Cộng đồng và hỗ trợ mạnh mẽ*: GitHub Actions có một cộng đồng lớn và tài liệu phong phú, với hàng ngàn action có sẵn trên marketplace giúp nhà phát triển dễ dàng tìm kiếm và tích hợp các công cụ phù hợp vào quy trình của mình.

2.4. Tìm hiểu về GitHub REST API

GitHub REST API là một giao diện lập trình ứng dụng (API) mạnh mẽ do GitHub cung cấp, cho phép các nhà phát triển tương tác với các tính năng và dịch vụ của GitHub từ bên ngoài thông qua các yêu cầu HTTP tiêu chuẩn. Với GitHub REST API, người dùng có thể tự động hóa các tác vụ liên quan đến quản lý kho lưu trữ, theo dõi vấn đề (Issues), quản lý yêu cầu kéo (Pull request) và nhiều chức năng khác mà GitHub cung cấp. API này hỗ trợ

định dạng JSON và có thể được sử dụng với bất kỳ ngôn ngữ lập trình nào có khả năng gửi các yêu cầu HTTP.^[3]

GitHub REST API bao gồm một loạt các chức năng đa dạng, từ quản lý người dùng đến thao tác trực tiếp với mã nguồn. Dưới đây là các chức năng chính của GitHub REST API:

- *Quản lý kho lưu trữ (Repository Management)*: GitHub REST API cho phép người dùng tạo, cập nhật, xóa và sao chép các kho lưu trữ. Các thao tác này giúp tự động hóa việc quản lý mã nguồn, đặc biệt trong các dự án lớn.

- *Theo dõi và quản lý vấn đề (Issues)*: API cung cấp các chức năng để tạo, theo dõi, bình luận và đóng các vấn đề. Điều này giúp cải thiện quy trình quản lý lỗi và công việc trong các dự án phần mềm.

- *Pull Request*: GitHub REST API cho phép tạo và quản lý các pull request, bao gồm việc theo dõi trạng thái, thêm nhận xét và hợp nhất các yêu cầu kéo.

- *Tương tác với Branch và Commit*: API có thể được sử dụng để tạo và xóa branch, cũng như truy cập thông tin về các commit cụ thể. Điều này hỗ trợ việc theo dõi các thay đổi trong lịch sử mã nguồn và xử lý phân nhánh trong quy trình phát triển.

- *Webhooks*: GitHub API hỗ trợ thiết lập và quản lý các webhook giúp tự động hóa quá trình nhận thông báo khi có sự kiện xảy ra trên kho lưu trữ như khi có commit mới hoặc pull request được tạo ra.

- *Quản lý tổ chức và nhóm (Organization and Team Management)*: GitHub REST API cũng hỗ trợ việc quản lý các tổ chức và đội nhóm trên GitHub cho phép người dùng thêm hoặc xóa thành viên, gán quyền truy cập và quản lý các kho lưu trữ của tổ chức.

Đặc biệt, GitHub Actions API cũng là một phần của GitHub REST API, cung cấp các phương thức để quản lý, theo dõi và tích hợp quy trình tự động hóa CI/CD trong GitHub Actions. Nó cho phép các nhà phát triển kiểm soát các workflow (quy trình công việc), chạy các tác vụ (jobs) và kiểm tra trạng thái các bản build một cách tự động thông qua các yêu cầu HTTP. GitHub Actions API đặc biệt hữu ích trong việc thiết lập, quản lý các pipeline phát triển phần mềm và hỗ trợ các quy trình Continuous Integration (CI) và Continuous Deployment (CD) một cách liên tục.

GitHub Actions API cung cấp các chức năng để quản lý toàn bộ quy trình GitHub Actions, bao gồm:

- *Workflow Management (Quản lý quy trình công việc)*: API cho phép liệt kê, kích hoạt, dừng hoặc khởi động lại các workflow từ xa. Người dùng có thể thiết lập các workflow tương ứng với các sự kiện, chẳng hạn như khi có commit mới, tạo pull request hoặc cập nhật mã nguồn trong kho lưu trữ.

- *Jobs and Runs (Chạy các công việc và tác vụ)*: Người dùng có thể truy cập thông tin chi tiết về các job (tác vụ) và run (lượt chạy) trong workflow, bao gồm trạng thái, thời gian chạy và log của các quá trình build, test. API cho phép kiểm tra từng bước trong pipeline và xem trạng thái chi tiết của từng job.

- *Artifacts Management (Quản lý kết quả và hiện vật)*: GitHub Actions API cho phép người dùng tải về hoặc xóa các artifacts (kết quả) được tạo ra từ các job, chẳng hạn như báo cáo kiểm thử, bản dựng ứng dụng hoặc file log. Điều này hỗ trợ việc lưu trữ và kiểm tra kết quả của các tác vụ trong quá trình chạy CI/CD.

- *Secret Management (Quản lý biến bí mật)*: API hỗ trợ quản lý các biến bí mật (secrets) được sử dụng trong các workflow. Người dùng có thể thêm, cập nhật hoặc xóa các secret một cách bảo mật giúp bảo vệ thông tin nhạy cảm như token truy cập, khóa API trong quy trình build.

- *Self-hosted Runners Management (Quản lý runner tự quản lý)*: GitHub Actions API cho phép quản lý các runner tự quản lý, bao gồm việc đăng ký, hủy đăng ký hoặc theo dõi tình trạng các runner đang được sử dụng trong quá trình thực hiện công việc.

GitHub REST API hoạt động theo mô hình RESTful, một tiêu chuẩn cho các giao diện lập trình web^[3]. Điều này có nghĩa là các thao tác với API được thực hiện thông qua các phương thức HTTP tiêu chuẩn như:

- *GET*: Lấy thông tin từ GitHub, chẳng hạn như danh sách kho lưu trữ, vấn đề hoặc thông tin người dùng.

- *POST*: Tạo mới các đối tượng như kho lưu trữ, pull request hoặc vấn đề.

- *PUT*: Cập nhật thông tin của một đối tượng hiện có, chẳng hạn như chỉnh sửa một vấn đề hoặc repo.

- *DELETE*: Xóa các đối tượng như kho lưu trữ hoặc webhook.

Mỗi yêu cầu đến GitHub REST API phải bao gồm các thông tin như phương thức HTTP, URL endpoint và các tham số hoặc dữ liệu cần thiết trong yêu cầu. API sử dụng JSON làm định dạng trả về giúp dễ dàng phân tích và xử lý dữ liệu.

GitHub REST API yêu cầu xác thực để truy cập vào các chức năng liên quan đến người dùng và kho lưu trữ riêng tư. Điều này đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể thực hiện các thay đổi hoặc truy cập dữ liệu nhạy cảm^[3]. Có hai phương thức chính để xác thực:

- *Token-based Authentication*: Người dùng có thể tạo token cá nhân để xác thực với GitHub API. Đây là cách phổ biến và an toàn nhất để truy cập API.
- *OAuth*: GitHub API hỗ trợ xác thực OAuth, cho phép ứng dụng của bên thứ ba truy cập vào GitHub mà không cần chia sẻ thông tin đăng nhập của người dùng.

GitHub REST API có cơ chế giới hạn số lượng yêu cầu API (Rate Limit) mà một người dùng hoặc ứng dụng có thể thực hiện trong một khoảng thời gian nhất định. Điều này giúp đảm bảo hiệu suất và bảo mật cho hệ thống của GitHub, tránh tình trạng quá tải từ các yêu cầu tự động hóa. Người dùng đã xác thực có thể thực hiện nhiều yêu cầu hơn so với người dùng không xác thực.

- *Người dùng không xác thực*: Giới hạn khoảng 60 yêu cầu mỗi giờ.
- *Người dùng đã xác thực*: Có thể gửi lên đến 5000 yêu cầu mỗi giờ.

Thông tin về trạng thái hạn chế tốc độ được bao gồm trong tiêu đề (header) của mỗi phản hồi API giúp người dùng theo dõi số lượng yêu cầu còn lại.

GitHub REST API được sử dụng rộng rãi trong các ứng dụng thực tế, bao gồm:

- *Tự động hóa quy trình phát triển*: GitHub API cho phép tạo các kho lưu trữ, theo dõi vấn đề và tạo pull request tự động, hỗ trợ các công việc CI/CD và DevOps.
- *Tích hợp với các công cụ khác*: GitHub API có thể được tích hợp với các hệ thống bên ngoài như hệ thống quản lý dự án (Jira, Trello) hoặc các dịch vụ CI/CD (Jenkins, Travis CI).
- *Quản lý tổ chức*: Tạo và quản lý người dùng, nhóm và quyền truy cập trong các tổ

chức trên GitHub.

GitHub cung cấp tài liệu phong phú cho GitHub REST API, bao gồm các ví dụ mã nguồn, mô tả chi tiết từng endpoint và các hướng dẫn về cách xác thực và xử lý lỗi. Cộng đồng GitHub cũng đóng vai trò quan trọng trong việc cung cấp hỗ trợ, với nhiều ví dụ thực hành, bài viết và hướng dẫn từ các nhà phát triển khắp nơi trên thế giới. Với sự hỗ trợ nhiều chức năng quan trọng và tính linh hoạt trong việc tương tác, GitHub REST API giúp các nhà phát triển tối ưu hóa quy trình làm việc, từ quản lý kho lưu trữ đến triển khai các giải pháp CI/CD.

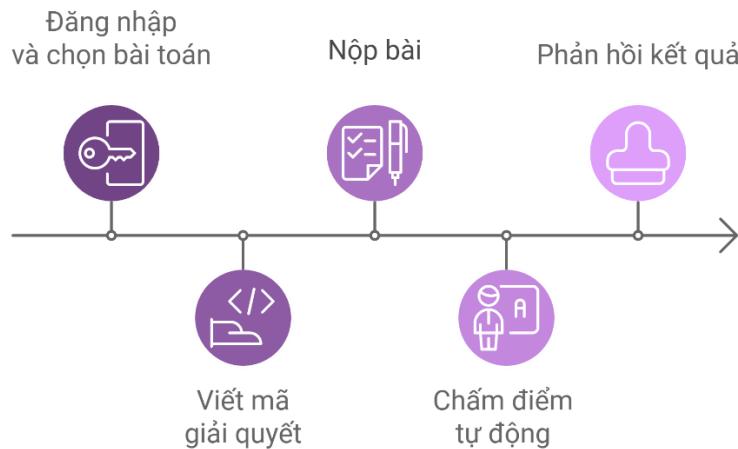
2.5. Tổng quan về các nền tảng đánh giá lập trình tự động

Nền tảng đánh giá lập trình tự động (Online Judge) là những hệ thống trực tuyến cho phép người dùng, thường là lập trình viên hoặc sinh viên công nghệ thông tin, thực hiện các bài tập lập trình và kiểm tra kết quả một cách tự động. Những nền tảng này đóng vai trò quan trọng trong giáo dục, rèn luyện kỹ năng lập trình và tổ chức các cuộc thi lập trình, đồng thời cũng được sử dụng rộng rãi trong các doanh nghiệp để kiểm tra năng lực của các ứng viên.

Các nền tảng đánh giá lập trình tự động (Online Judge systems, hay OJs) là những hệ thống trực tuyến cung cấp một môi trường để lập trình viên giải quyết các vấn đề theo yêu cầu, sau đó nộp bài giải (thường là mã nguồn của chương trình) để được đánh giá tự động. Mục tiêu chính của các nền tảng này là kiểm tra khả năng giải quyết vấn đề và kỹ năng lập trình của người dùng thông qua việc đánh giá:

- *Độ chính xác*: Bài giải có trả về kết quả đúng theo yêu cầu của bài toán không.
- *Hiệu suất*: Bài giải có thực hiện trong giới hạn thời gian và bộ nhớ đã định không.
- *Tính tối ưu*: Bài giải có sử dụng tài nguyên hệ thống một cách hiệu quả không, đặc biệt là trong những bài toán phức tạp hoặc yêu cầu xử lý khối lượng lớn dữ liệu.

Các nền tảng OJ không chỉ hỗ trợ cho giáo dục và rèn luyện cá nhân mà còn là công cụ quan trọng trong tuyển dụng, thi đấu lập trình và các cuộc thi quốc tế như ACM ICPC, Google Code Jam, Facebook Hacker Cup hay VNOI (Việt Nam).



Hình 2.2. Quy trình cơ bản của các nền tảng Online Judge

Quy trình cơ bản của các nền tảng Online Judge bao gồm các bước sau:

Bước 1: Đăng nhập và chọn bài toán: Người dùng đăng nhập vào hệ thống, sau đó chọn một bài toán lập trình để giải quyết. Bài toán thường được mô tả chi tiết về yêu cầu, các ràng buộc (ví dụ: giới hạn thời gian, giới hạn bộ nhớ) và các bộ dữ liệu đầu vào, đầu ra mẫu.

Bước 2: Viết mã giải quyết: Người dùng viết chương trình của mình (sử dụng một trong nhiều ngôn ngữ lập trình được hỗ trợ như C, C++, Java, Python, v.v.) để giải quyết bài toán. Chương trình phải đảm bảo nhận đầu vào theo yêu cầu và trả về kết quả chính xác theo định dạng đầu ra đã cho.

Bước 3: Nộp bài: Sau khi viết mã, người dùng nộp chương trình của mình lên hệ thống để được chấm điểm tự động.

Bước 4: Chấm điểm tự động: Hệ thống sẽ chạy chương trình trên một tập các bộ test case được chuẩn bị sẵn, bao gồm các bộ dữ liệu mẫu và các tình huống thử nghiệm đặc biệt để kiểm tra tính đúng đắn và hiệu quả của chương trình. Kết quả được đánh giá dựa trên các tiêu chí như:

- + *Kết quả đúng*: Kết quả chương trình có khớp với kết quả mong đợi không.
 - + *Thời gian thực thi*: Chương trình có chạy trong giới hạn thời gian quy định không.
 - + *Bộ nhớ sử dụng*: Chương trình có vượt quá giới hạn bộ nhớ không.
 - *Phản hồi kết quả*: Sau khi hoàn tất, hệ thống sẽ cung cấp phản hồi cho người dùng,

bao gồm trạng thái bài giải, điểm số và đôi khi là thông tin chi tiết về các test case đã không đạt yêu cầu. Kết quả có thể là một trong các trường hợp được mô tả trong bảng 2.1:

Bảng 2.1. Các kết quả chấm bài trong một hệ thống Online Judge

Mã trạng thái	Diễn giải
AC (Accepted)	Chương trình chạy đúng với toàn bộ test của đề bài.
WA (Wrong Answer)	Chương trình đưa ra output không giống với output chuẩn, kết quả sai.
RTE (Runtime Exception hoặc Runtime Error)	Chương trình sinh lỗi và trả về một giá trị khác, thường chỉ xuất hiện khi sử dụng ngôn ngữ như C hoặc C++.
IR (Invalid Return)	Chương trình trả về một giá trị khác 0, tương tự RTE, thường xuất hiện ở các ngôn ngữ Python hoặc Java khi xử lý ngoại lệ (exception).
OLE (Output Limit Exceeded)	Chương trình in ra quá nhiều thông tin ra output gây tràn bộ nhớ (giới hạn bộ nhớ thường là 256MB).
TLE (Time Limit Exceeded)	Chương trình chạy quá thời gian quy định.
IE (Internal Error)	Lỗi hệ thống.

Các tính năng chính của nền tảng đánh giá lập trình tự động

- *Kho bài tập đa dạng*: Các nền tảng OJ thường cung cấp một lượng lớn bài tập thuộc nhiều chủ đề khác nhau, từ những vấn đề đơn giản cho người mới bắt đầu đến các bài toán khó hơn yêu cầu sử dụng các thuật toán phức tạp như đồ thị, quy hoạch động hoặc lý thuyết số.

- *Hỗ trợ nhiều ngôn ngữ lập trình*: Đa số các hệ thống OJ hỗ trợ nhiều ngôn ngữ lập trình khác nhau, cho phép người dùng lựa chọn ngôn ngữ mà họ quen thuộc hoặc muốn học.

- *Chấm điểm tự động và tức thì*: Các bài nộp được chấm điểm tự động, thường là ngay

lập tức sau khi nộp, với phản hồi rõ ràng về các test case không thành công hoặc các lỗi phát sinh.

- *Dánh giá hiệu suất*: Ngoài việc kiểm tra tính chính xác, hệ thống cũng đánh giá hiệu suất của bài giải. Điều này đảm bảo rằng người dùng không chỉ cần một bài giải đúng mà còn phải tối ưu về thời gian và bộ nhớ.

- *Cộng đồng và xếp hạng*: Một số nền tảng cho phép người dùng cạnh tranh với nhau thông qua hệ thống xếp hạng hoặc thảo luận về các bài giải trong cộng đồng. Điều này giúp thúc đẩy việc học tập và chia sẻ kiến thức.

Các nền tảng Online Judge phổ biến:

1. *HackerRank*: Là một nền tảng phổ biến dành cho các lập trình viên, HackerRank cung cấp các thử thách lập trình từ dễ đến khó và hỗ trợ nhiều ngôn ngữ lập trình. Nó còn tích hợp các bài kiểm tra kỹ năng, được sử dụng rộng rãi trong tuyển dụng.

2. *LeetCode*: Chuyên về các bài toán liên quan đến cấu trúc dữ liệu và thuật toán, LeetCode là lựa chọn hàng đầu cho những ai muốn chuẩn bị cho các cuộc phỏng vấn lập trình.

3. *Codeforces*: Đây là một nền tảng nổi tiếng trong cộng đồng thi đấu lập trình, tổ chức nhiều cuộc thi lập trình trực tuyến và cung cấp các bài toán thử thách.

4. *AtCoder*: Được biết đến với các cuộc thi lập trình chuyên nghiệp, AtCoder cung cấp một môi trường thử thách thú vị cho các lập trình viên và nhà khoa học máy tính.

5. *SPOJ (Sphere Online Judge)*: Cung cấp hơn 20.000 bài toán, SPOJ là một nền tảng OJ lâu đời, cho phép người dùng rèn luyện và phát triển kỹ năng lập trình ở nhiều mức độ khác nhau.

Ở Việt Nam, cũng có một số nền tảng đánh giá lập trình tự động nổi bật, điển hình là VNOI (Vietnamese Olympiad in Informatics), được xây dựng với mục tiêu phát triển cộng đồng lập trình viên Việt Nam và hỗ trợ luyện tập cho các kỳ thi Tin học quốc tế. Thông qua việc luyện tập trên VNOI, nhiều học sinh đã đạt thành tích cao tại các kỳ thi Olympic Tin học và ACM ICPC.

Lợi ích của nền tảng Online Judge:

- *Cải thiện kỹ năng lập trình*: Các lập trình viên có thể luyện tập và nâng cao kỹ năng giải quyết vấn đề của mình thông qua việc giải quyết nhiều dạng bài toán khác nhau. Đây là cách tuyệt vời để học các thuật toán và cấu trúc dữ liệu một cách thực tế.

- *Chuẩn bị cho phỏng vấn lập trình*: Nhiều công ty công nghệ lớn sử dụng các bài toán lập trình trong các cuộc phỏng vấn. Việc luyện tập trên các nền tảng OJ là cách hiệu quả để chuẩn bị cho các câu hỏi phỏng vấn lập trình.

- *Thi đấu*: Các cuộc thi lập trình online trên các nền tảng như Codeforces hay AtCoder giúp lập trình viên kiểm tra kỹ năng của mình trong môi trường thi đấu, phát triển tư duy thuật toán và khả năng giải quyết vấn đề nhanh chóng.

- *Hỗ trợ tuyển dụng*: Nhiều nền tảng như HackerRank, LeetCode và CodeSignal cung cấp các dịch vụ hỗ trợ tuyển dụng. Nhà tuyển dụng có thể sử dụng các bài toán lập trình trên những nền tảng này để kiểm tra năng lực của ứng viên một cách khách quan.

Thách thức trong việc sử dụng nền tảng đánh giá lập trình tự động:

- *Hạn chế về ngôn ngữ*: Một số nền tảng chỉ hỗ trợ một số ngôn ngữ lập trình nhất định, có thể giới hạn khả năng thực hành của người dùng.

- *Giới hạn thời gian và bộ nhớ*: Các bài toán trên OJ thường có yêu cầu rất khắt khe về thời gian và bộ nhớ, điều này có thể làm khó người dùng, đặc biệt là những người mới học lập trình.

- *Cạnh tranh và áp lực*: Trong các cuộc thi lập trình, thời gian và mức độ cạnh tranh cao có thể tạo áp lực lớn cho người tham gia, đòi hỏi họ phải có kỹ năng giải quyết vấn đề và lập trình nhanh nhẹn.

2.6. Tìm hiểu về mô hình ngôn ngữ lớn và Gemini

2.6.1. Tìm hiểu về mô hình ngôn ngữ lớn

Mô hình ngôn ngữ lớn (LLM) là một loại mạng thần kinh nhân tạo được huấn luyện trên một lượng lớn văn bản. Chúng được thiết kế để hiểu, tạo ra và giải thích thông tin bằng ngôn ngữ tự nhiên^[7]. LLM đã trở thành một trong những tiến bộ đáng chú ý nhất trong lĩnh vực trí tuệ nhân tạo (AI) trong những năm gần đây, với khả năng thực hiện nhiều nhiệm vụ phức tạp như:

- *Tạo văn bản*: LLM có thể tạo ra các văn bản chất lượng cao, từ các bài báo, bài thơ, kịch bản, đến các đoạn mã máy tính.

- *Tóm tắt văn bản*: LLM có thể tự động tóm tắt các đoạn văn bản dài thành những đoạn ngắn, súc tích mà vẫn giữ nguyên được ý nghĩa trọng tâm.

- *Dịch thuật*: LLM có thể dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác một cách chính xác và lưu loát.

- *Trả lời câu hỏi*: LLM có thể trả lời các câu hỏi phức tạp bằng cách tìm kiếm thông tin từ một lượng lớn dữ liệu.

- *Tương tác với người dùng*: LLM có thể tham gia vào các cuộc hội thoại tự nhiên với con người, tạo ra các chatbot thông minh và trợ lý ảo.

LLM hoạt động dựa trên kiến trúc transformer, một loại mạng thần kinh được thiết kế đặc biệt để xử lý dữ liệu tuần tự như văn bản.^[7] Transformer cho phép LLM hiểu ngữ cảnh của các từ trong một câu, từ đó tạo ra các văn bản có ý nghĩa và mạch lạc.

Các yếu tố quan trọng của LLM:

- *Kích thước*: LLM thường có số lượng tham số rất lớn, cho phép chúng học được các biểu diễn ngôn ngữ phức tạp.

- *Dữ liệu huấn luyện*: Chất lượng và số lượng dữ liệu huấn luyện có ảnh hưởng lớn đến hiệu suất của LLM.

- *Kiến trúc*: Kiến trúc transformer là một trong những kiến trúc phổ biến nhất cho LLM nhưng cũng có nhiều kiến trúc khác đang được nghiên cứu.

2.6.2. Tìm hiểu về Gemini

Gemini là một trong những LLM mạnh mẽ nhất được phát triển bởi Google AI. Gemini được thiết kế để vượt trội trong nhiều nhiệm vụ, bao gồm hiểu ngôn ngữ, lập luận và giải quyết các vấn đề phức tạp^[7]. Một số điểm nổi bật của Gemini:

- *Khả năng đa nhiệm*: Gemini có thể thực hiện nhiều nhiệm vụ khác nhau một cách đồng thời, từ dịch thuật, viết mã, đến tạo ra các hình ảnh và video.

- *Hiểu biết sâu sắc về thế giới*: Gemini được huấn luyện trên một lượng lớn dữ liệu, cho phép nó có một hiểu biết sâu sắc về thế giới và các mối quan hệ giữa các sự vật.

- *Khả năng thích ứng*: Gemini có khả năng thích ứng với các nhiệm vụ mới và các loại dữ liệu mới một cách nhanh chóng.

Ứng dụng của Gemini:

1. *Tìm kiếm thông tin*: Gemini có thể được sử dụng để tìm kiếm thông tin một cách hiệu quả hơn, bằng cách hiểu ý định của người dùng và cung cấp các kết quả chính xác hơn.

2. *Tạo nội dung*: Gemini có thể được sử dụng để tạo ra nhiều loại nội dung khác nhau, từ các bài báo, bài viết blog, đến các kịch bản phim.

3. *Phát triển phần mềm*: Gemini có thể được sử dụng để hỗ trợ các nhà phát triển phần mềm trong việc viết mã, kiểm tra lỗi và tối ưu hóa mã nguồn.

4. *Dịch vụ khách hàng*: Gemini có thể được sử dụng để tạo ra các chatbot thông minh, có khả năng tương tác với khách hàng một cách tự nhiên và hiệu quả.

Gemini cung cấp một số API để có thể sử dụng mô hình này một cách chủ động hơn. Với API từ Gemini, việc đánh giá chất lượng mã nguồn có thể được tự động hóa dựa vào trí tuệ nhân tạo (AI) theo các tiêu chuẩn được yêu cầu từ dữ liệu đầu vào.

2.7. Các công nghệ xây dựng trang web

2.7.1. Ngôn ngữ HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chuẩn được sử dụng để xây dựng và định dạng nội dung trên các trang web. HTML lần đầu được phát triển vào năm 1993 bởi Tim Berners-Lee, người sáng tạo ra World Wide Web. HTML đã trải qua nhiều phiên bản và cập nhật, từ HTML 1.0 cho đến phiên bản hiện tại là HTML5 giúp cải thiện tính năng, khả năng tương tác và hỗ trợ các công nghệ hiện đại hơn trên web.

HTML rất quan trọng vì nó giúp các trình duyệt web hiểu và hiển thị nội dung theo đúng cấu trúc mà nhà phát triển mong muốn. Nó cũng hỗ trợ việc tạo ra các siêu liên kết (hyperlinks), cho phép người dùng điều hướng giữa các trang web một cách dễ dàng. Nhờ sự đơn giản và tính phổ biến, HTML là ngôn ngữ cốt lõi mà mọi nhà phát triển web đều phải nắm vững khi xây dựng trang web hoặc ứng dụng trực tuyến.

Vai trò chính của HTML là:

- *Tạo cấu trúc nội dung*: HTML xác định cách các phần tử (như đoạn văn, tiêu đề, liên kết, hình ảnh, v.v.) được tổ chức và hiển thị trên một trang web.

- *Kết hợp với CSS và JavaScript*: HTML cung cấp cấu trúc cơ bản, trong khi CSS định nghĩa về hình thức và JavaScript thêm các tính năng tương tác.

Một tài liệu HTML cơ bản thường có cấu trúc sau:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <h1>Example Heading</h1>
  </body>
</html>
```

Các phần chính:

<!DOCTYPE html>: Khai báo kiểu tài liệu, thông báo cho trình duyệt biết rằng tài liệu này sử dụng HTML5.

<html>: Thẻ gốc chứa toàn bộ nội dung của tài liệu HTML.

<head>: Chứa thông tin siêu dữ liệu về tài liệu như tiêu đề (hiển thị trên tab trình duyệt), charset và các liên kết đến tài nguyên như CSS.

<body>: Chứa nội dung thực sự hiển thị trên trang web, bao gồm các phần tử như văn bản, hình ảnh, liên kết, biểu mẫu,...

Các thẻ HTML thông dụng được trình bày trong bảng 2.2:

Bảng 2.2. Các thẻ HTML thông dụng

Tên thẻ	Mô tả, chức năng
<h1>...<h6>	Tiêu đề, từ h1 (tiêu đề chính) đến h6 (tiêu đề phụ nhỏ nhất)
<p>	Đoạn văn
<a>	Thẻ liên kết (anchor), cho phép tạo liên kết đến các trang khác hoặc phần khác của cùng một trang
	Thẻ nội tuyến dùng để gói một đoạn văn bản mà không ảnh hưởng đến cách dàn trang
<div>	Thẻ khói dùng để tạo một khói hoặc vùng trên trang
	Thẻ hiển thị hình ảnh
<video>	Thẻ nhúng video
	Danh sách không có thứ tự (bullet points)
	Danh sách có thứ tự (số hoặc chữ cái)
	Mục trong danh sách
<form>	Tạo biểu mẫu để nhập dữ liệu
<input>	Thẻ nhập liệu trong biểu mẫu như trường văn bản, nút radio, hộp kiểm tra
<textarea>	Vùng nhập liệu nhiều dòng
<button>	Nút bấm
<table>	Tạo bảng
<tr>	Hàng trong bảng
<td>	Ô dữ liệu trong bảng
<th>	Ô tiêu đề cột hoặc hàng
<header>	Phần đầu trang hoặc đầu của một phần
<footer>	Phần cuối trang hoặc cuối của một phần

<article>	Một bài viết hoặc khối nội dung độc lập
<section>	Một phần hoặc khối nội dung có chủ đề riêng
<nav>	Điều hướng, thường chứa liên kết đến các phần khác của trang hoặc trang web
<canvas>	Tạo không gian đồ họa động, hỗ trợ vẽ đồ họa 2D và 3D thông qua JavaScript

2.7.2. *Bảng định kiểu CSS*

CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để mô tả cách trình bày, định dạng và bố cục của các phần tử trong một tài liệu HTML. Trong khi HTML định nghĩa cấu trúc và nội dung của trang web, CSS được sử dụng để kiểm soát giao diện của trang, bao gồm màu sắc, phông chữ, khoảng cách, kích thước, vị trí của các phần tử và nhiều thuộc tính khác.

CSS hoạt động bằng cách áp dụng các quy tắc (rules) lên các phần tử HTML. Mỗi quy tắc bao gồm một bộ chọn (selector) để chọn các phần tử cụ thể và một tập hợp các thuộc tính (properties) cùng giá trị (values) để định nghĩa kiểu dáng cho các phần tử đó.

Một số đặc điểm chính của CSS:

- *Tách biệt nội dung và trình bày*: CSS giúp tách riêng phần trình bày ra khỏi nội dung giúp việc quản lý và thay đổi giao diện trang web trở nên dễ dàng hơn mà không ảnh hưởng đến cấu trúc HTML.

- *Thùa kế và phân tầng (Cascading)*: CSS tuân theo nguyên tắc phân tầng, nghĩa là khi có nhiều quy tắc được áp dụng lên cùng một phần tử, các quy tắc có độ ưu tiên cao hơn sẽ được áp dụng. CSS cũng hỗ trợ thừa kế, cho phép các thuộc tính của một phần tử cha được kế thừa bởi các phần tử con.

- *Định dạng đa dạng*: CSS cho phép định dạng phong phú và linh hoạt như màu sắc, phông chữ, căn lề, khoảng cách giữa các phần tử, độ rộng, độ cao, bo tròn góc, hiệu ứng đổ bóng, hiệu ứng chuyển động và nhiều thuộc tính khác.

- *Khả năng đáp ứng (Responsive)*: CSS hỗ trợ các kỹ thuật tạo ra giao diện linh hoạt, phù hợp với nhiều kích thước màn hình khác nhau, từ điện thoại di động đến máy tính để

bàn, thông qua việc sử dụng các đơn vị đo lường linh hoạt và các truy vấn phuong tiện (media queries).

Có ba cách chính để tích hợp CSS vào tài liệu HTML:

- *Inline CSS*: Được định nghĩa trực tiếp trong thuộc tính style của một thẻ HTML.
- *Internal CSS*: Được đặt trong thẻ `<style>` trong phần `<head>` của tài liệu HTML.
- *External CSS*: CSS được viết trong một file riêng biệt với đuôi `.css` và liên kết với tài liệu HTML bằng thẻ `<link>`.

2.7.3. Ngôn ngữ Javascript

JavaScript là một ngôn ngữ lập trình động, phổ biến và mạnh mẽ được sử dụng rộng rãi để phát triển các ứng dụng web, ứng dụng di động, trò chơi và nhiều loại phần mềm khác. Ban đầu, JavaScript được phát triển vào năm 1995 bởi Brendan Eich tại Netscape nhằm bổ sung khả năng tương tác cho các trang web nhưng kể từ đó, nó đã phát triển và trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Tính năng nổi bật của JavaScript

- *Ngôn ngữ lập trình phía client*: JavaScript chủ yếu được sử dụng để lập trình phía client (trên trình duyệt web) giúp tạo ra các tương tác động trên giao diện người dùng mà không cần phải tải lại toàn bộ trang web. Nó cho phép các nhà phát triển tạo ra các ứng dụng web phong phú và tương tác từ việc kiểm tra đầu vào của người dùng đến thay đổi giao diện một cách động.

- *Tính năng đa nền tảng*: JavaScript hoạt động trên hầu hết các trình duyệt web hiện đại mà không cần phải cài đặt thêm bất kỳ phần mềm hoặc plugin nào. Điều này giúp nó trở thành một ngôn ngữ đa nền tảng, cho phép viết một lần và chạy trên nhiều hệ điều hành và trình duyệt khác nhau.

- *Hỗ trợ lập trình hướng đối tượng*: JavaScript là một ngôn ngữ lập trình hỗ trợ lập trình hướng đối tượng, cho phép các nhà phát triển tạo ra các đối tượng, kế thừa và tái sử dụng mã. Các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance) đều có thể được áp dụng trong JavaScript thông qua phiên bản ES6 trở về sau.

- *Lập trình sự kiện*: JavaScript cung cấp khả năng lập trình sự kiện giúp các ứng dụng

web có thể phản hồi lại các hành động của người dùng như nhấp chuột, di chuyển chuột, nhập liệu trên bàn phím hoặc các sự kiện khác. Điều này giúp tạo ra trải nghiệm người dùng linh hoạt và tương tác.

- *Ngôn ngữ không đồng bộ*: JavaScript có khả năng xử lý các tác vụ không đồng bộ một cách hiệu quả thông qua các tính năng như callback, promises và async/await. Điều này giúp nó xử lý tốt các tác vụ I/O (Input/Output) như gọi API, đọc file mà không làm gián đoạn các tác vụ khác.

- *Khả năng mở rộng mạnh mẽ*: Ngoài việc lập trình phía client, với sự ra đời của Node.js, JavaScript đã mở rộng sang lĩnh vực lập trình phía server, cho phép xây dựng các ứng dụng server-side mạnh mẽ và hiệu quả. Node.js cũng cung cấp một môi trường thực thi JavaScript trên máy chủ giúp tạo ra các ứng dụng thời gian thực như chat và streaming.

Ưu điểm của JavaScript

- *Tốc độ*: Vì JavaScript là ngôn ngữ được thực thi trực tiếp trên trình duyệt mà không cần biên dịch trước nên nó rất nhanh trong việc thực thi mã và phản hồi lại các hành động của người dùng.

- *Dễ học*: Cú pháp của JavaScript tương đối đơn giản và dễ tiếp cận, đặc biệt là đối với những người mới học lập trình.

- *Khả năng tương tác mạnh mẽ*: JavaScript dễ dàng tương tác với các ngôn ngữ và công nghệ khác như HTML và CSS để tạo ra các ứng dụng web phức tạp.

- *Hệ sinh thái phong phú*: JavaScript có một hệ sinh thái phong phú với nhiều thư viện và framework nổi tiếng như React, Angular, Vue.js giúp đơn giản hóa việc phát triển các ứng dụng lớn.

Một số thách thức của JavaScript

- *Hiệu năng trong các ứng dụng lớn*: Khi ứng dụng web lớn dần, quản lý mã JavaScript trở nên phức tạp, điều này có thể ảnh hưởng đến hiệu năng nếu không được tối ưu hóa đúng cách.

- *Tính không đồng bộ*: Mặc dù JavaScript có khả năng xử lý các tác vụ không đồng bộ, việc quản lý các tác vụ này đôi khi có thể phức tạp, đặc biệt là với những người mới làm quen.

Với khả năng lập trình linh hoạt và đa năng, JavaScript không chỉ giúp tạo ra các trang web tương tác mà còn mở rộng sang lĩnh vực phát triển di động, desktop và server. Hệ sinh thái JavaScript không ngừng phát triển giúp các nhà phát triển có thêm nhiều công cụ mạnh mẽ để xây dựng các ứng dụng phức tạp, hiệu quả và mang tính tương tác cao.

2.7.4. React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI). React tập trung vào việc phát triển các thành phần (components) có thể tái sử dụng giúp quản lý trạng thái của ứng dụng một cách hiệu quả và linh hoạt. Mục tiêu chính của React là tạo ra các giao diện tương tác một cách mượt mà và hiệu quả, nhờ vào việc sử dụng cơ chế "ảo hóa" DOM (Virtual DOM). Khi trạng thái của ứng dụng thay đổi, React sẽ so sánh phiên bản DOM ảo với phiên bản DOM thực tế và chỉ cập nhật những phần cần thiết giúp cải thiện hiệu suất. React cũng mang đến khái niệm về "component-based architecture", cho phép xây dựng giao diện từ những thành phần nhỏ gọn, dễ quản lý và tái sử dụng. Điều này giúp đơn giản hóa việc phát triển các ứng dụng phức tạp và tăng khả năng mở rộng. React không chỉ hỗ trợ việc phát triển các ứng dụng web mà còn được mở rộng để phát triển các ứng dụng di động thông qua React Native. Với cộng đồng lớn mạnh và hệ sinh thái phong phú, React là một trong những công cụ phổ biến và quan trọng trong việc phát triển giao diện web hiện đại.

Các đặc điểm chính của React:

- *Component-based (Dựa trên thành phần)*: React tổ chức giao diện người dùng thành các component (thành phần). Một component có thể coi như một khối xây dựng nhỏ, chứa logic và giao diện liên quan đến một phần cụ thể của ứng dụng. Các component này có thể được lồng ghép vào nhau để tạo ra các giao diện phức tạp hơn. Mỗi component có thể tái sử dụng nhiều lần trong toàn bộ ứng dụng giúp mã nguồn dễ quản lý và bảo trì.

- *Virtual DOM (DOM ảo)*: Một trong những tính năng nổi bật của React là việc sử dụng Virtual DOM để tối ưu hóa hiệu suất. Khi có sự thay đổi trong giao diện, React sẽ tạo ra một bản sao DOM ảo của ứng dụng, sau đó so sánh (diff) với phiên bản trước. Chỉ những phần tử cần thiết mới được cập nhật trong DOM thực giúp giảm thiểu số lượng thao tác trên DOM và cải thiện tốc độ render.

- *One-way Data Binding (Ràng buộc dữ liệu một chiều)*: Dữ liệu trong React được

truyền từ component cha xuống component con thông qua các props (thuộc tính). Điều này đảm bảo rằng dữ liệu được kiểm soát một cách rõ ràng và giảm thiểu các tình huống khó đoán khi dữ liệu có thể thay đổi từ nhiều nguồn khác nhau. Với luồng dữ liệu một chiều, việc kiểm soát và theo dõi luồng dữ liệu trở nên đơn giản và dễ dàng hơn.

- *JSX (JavaScript XML)*: React sử dụng một cú pháp mở rộng của JavaScript gọi là JSX, cho phép các nhà phát triển viết mã HTML trực tiếp trong JavaScript. JSX giúp mô tả giao diện một cách rõ ràng và dễ đọc hơn. Mặc dù không bắt buộc nhưng JSX trở thành một phần quan trọng trong việc phát triển ứng dụng React vì nó kết hợp mã giao diện và logic một cách mạch lạc.

- *State (Trạng thái)*: Mỗi component trong React có thể có một trạng thái riêng (state) và khi trạng thái này thay đổi, React sẽ tự động render lại component tương ứng để phản ánh thay đổi đó. State thường được sử dụng để quản lý các thông tin động và sự tương tác của người dùng trong một phần tử giao diện.

- *Hooks (Các móc hàm)*: Hooks là một tính năng mới được giới thiệu trong React 16.8, cho phép các nhà phát triển sử dụng state và các tính năng khác của React mà không cần phải sử dụng các class component. Các hook như useState, useEffect giúp đơn giản hóa việc quản lý trạng thái và vòng đời component trong các component hàm, từ đó giúp mã dễ hiểu và gọn gàng hơn.

- *Ecosystem (Hệ sinh thái)*: React có một hệ sinh thái phát triển rất phong phú với hàng ngàn thư viện và công cụ hỗ trợ. Các công cụ như React Router giúp quản lý điều hướng trong ứng dụng hoặc Redux giúp quản lý trạng thái toàn cục. Nhờ hệ sinh thái này, React không chỉ hỗ trợ phát triển giao diện người dùng mà còn có thể mở rộng để xây dựng các ứng dụng phức tạp hơn.

Lợi ích của việc sử dụng React:

- *Hiệu suất cao*: Việc sử dụng Virtual DOM giúp tăng tốc độ cập nhật giao diện, giảm thiểu thao tác trực tiếp trên DOM, từ đó tối ưu hiệu suất của ứng dụng.

- *Khả năng tái sử dụng*: Các component có thể được tái sử dụng nhiều lần trong nhiều phần của ứng dụng giúp tiết kiệm thời gian phát triển và giảm thiểu lỗi.

- *Quản lý giao diện dễ dàng*: React giúp chia nhỏ giao diện phức tạp thành các

component nhỏ hơn giúp việc phát triển và bảo trì trở nên đơn giản hơn.

- *Cộng đồng lớn và nhiều tài liệu*: Do được sử dụng rộng rãi, React có một cộng đồng lớn với nhiều tài liệu, công cụ và thư viện hỗ trợ giúp việc học tập và phát triển dễ dàng hơn.

React hiện tại là một trong những thư viện hàng đầu cho việc xây dựng giao diện người dùng trên web nhờ vào sự linh hoạt, hiệu suất cao và khả năng tái sử dụng component. Với sự hỗ trợ từ cộng đồng lớn và một hệ sinh thái phong phú, React tiếp tục là lựa chọn ưu tiên cho các nhà phát triển trong việc xây dựng các ứng dụng web hiện đại.

2.7.5. TailwindCSS

TailwindCSS là một framework CSS hiện đại được thiết kế để cung cấp một phương pháp tiếp cận mới trong việc xây dựng giao diện người dùng (UI) trên web. Khác với các framework truyền thống như Bootstrap hay Foundation, vốn dựa vào các thành phần giao diện được xây dựng sẵn, TailwindCSS áp dụng triết lý "utility-first" (ưu tiên lớp tiện ích), cho phép nhà phát triển định dạng trực tiếp các phần tử HTML thông qua các lớp CSS nhỏ và có thể tái sử dụng.^[5]

Tổng quan về TailwindCSS:

- *Triết lý "utility-first"*: TailwindCSS cung cấp các lớp tiện ích độc lập tương ứng với các thuộc tính CSS phổ biến như căn lề, kích thước màu sắc và căn chỉnh. Thay vì phải viết các khối CSS phức tạp cho từng phần tử, nhà phát triển chỉ cần kết hợp các lớp tiện ích này để tạo nên giao diện tùy chỉnh. Cách tiếp cận này giúp việc phát triển trở nên nhanh chóng hơn, giảm thiểu lượng CSS tùy chỉnh cần viết và duy trì.

- *Phân tầng và tính kế thừa*: Cũng như các framework khác, TailwindCSS sử dụng nguyên tắc phân tầng (cascading), trong đó các lớp CSS có thể được ghi đè dựa trên độ ưu tiên. Ngoài ra, các lớp tiện ích trong Tailwind được tổ chức theo cách cho phép sự kế thừa, nghĩa là các thiết lập cơ bản có thể được mở rộng và tùy chỉnh dễ dàng thông qua các lớp tiện ích mới.

- *Tùy chỉnh linh hoạt*: TailwindCSS được đánh giá cao về khả năng tùy biến. Nó cho phép người dùng tùy chỉnh toàn bộ cấu trúc hệ thống lớp tiện ích thông qua tập tin cấu hình. Các thông số như màu sắc, khoảng cách, độ rộng, độ cao và nhiều thuộc tính khác

đều có thể được điều chỉnh sao cho phù hợp với yêu cầu cụ thể của từng dự án. Điều này mang lại khả năng thích ứng cao cho các nhà phát triển khi cần xây dựng giao diện độc đáo.^[5]

- *Thiết kế theo hướng đáp ứng (Responsive Design)*: TailwindCSS được tích hợp sẵn các công cụ hỗ trợ phát triển giao diện đáp ứng (responsive) giúp dễ dàng điều chỉnh giao diện cho nhiều loại thiết bị khác nhau, từ máy tính bàn đến điện thoại di động. Cách tiếp cận này không chỉ giảm thời gian phát triển mà còn đảm bảo tính nhất quán trong trải nghiệm người dùng trên các màn hình với kích thước khác nhau.

- *Khả năng tối ưu hiệu suất*: Một trong những lợi thế nổi bật của TailwindCSS là khả năng tối ưu hiệu suất thông qua việc sử dụng các công cụ như PurgeCSS để loại bỏ các lớp CSS không được sử dụng khi triển khai ứng dụng. Kết quả là kích thước tập tin CSS cuối cùng sẽ được giảm đáng kể, từ đó cải thiện tốc độ tải trang và nâng cao trải nghiệm người dùng. Điều này đặc biệt hữu ích trong các dự án lớn với nhiều lớp CSS phức tạp.

- *Phát triển nhanh chóng*: Một trong những lợi ích lớn nhất của TailwindCSS là khả năng tăng tốc quá trình phát triển giao diện. Việc sử dụng các lớp tiện ích sẵn có giúp giảm thiểu sự phụ thuộc vào các lớp CSS tùy chỉnh phức tạp, từ đó giảm thời gian xây dựng và bảo trì giao diện. Đồng thời, tính nhất quán trong việc sử dụng các lớp tiện ích cũng giúp đảm bảo tính đồng nhất của giao diện trên toàn bộ dự án.

Lợi ích của TailwindCSS:

- *Tính tùy chỉnh cao*: Với TailwindCSS, nhà phát triển có toàn quyền kiểm soát giao diện mà không cần phải dựa vào các thành phần giao diện có sẵn. Framework này cho phép tùy chỉnh toàn bộ giao diện mà vẫn đảm bảo tính linh hoạt và khả năng mở rộng.

- *Hỗ trợ phát triển giao diện đáp ứng*: TailwindCSS giúp đơn giản hóa quá trình xây dựng giao diện thích ứng với nhiều kích thước màn hình khác nhau. Nhà phát triển có thể dễ dàng áp dụng các lớp tiện ích tương ứng để điều chỉnh thiết kế cho các thiết bị khác nhau mà không cần phải viết nhiều mã CSS phức tạp.

- *Tối ưu hóa hiệu suất*: Khả năng tự động loại bỏ các lớp CSS không sử dụng của TailwindCSS giúp giảm thiểu lượng mã không cần thiết, từ đó tăng tốc độ tải trang và nâng cao hiệu quả hoạt động của ứng dụng.

- *Thúc đẩy sự sáng tạo*: Thay vì bị hạn chế bởi các thành phần giao diện dựng sẵn, nhà phát triển có thể tự do sáng tạo và tùy chỉnh giao diện theo yêu cầu cụ thể của dự án, từ đó mang lại tính độc đáo và khác biệt cho sản phẩm.

TailwindCSS đang là một lựa chọn phổ biến trong cộng đồng phát triển web nhờ vào sự linh hoạt, khả năng tùy chỉnh và hiệu suất tối ưu mà nó mang lại. Với việc tập trung vào các lớp tiện ích, framework này cung cấp một phương pháp tiếp cận hiện đại và hiệu quả để xây dựng giao diện người dùng, từ đó giúp tăng cường tính nhất quán, giảm thiểu sự phức tạp và tối ưu hóa quy trình phát triển. TailwindCSS không chỉ là một công cụ hữu ích cho các dự án lớn mà còn phù hợp với các dự án nhỏ nhờ khả năng tối ưu hiệu suất và dễ dàng tùy biến.

2.7.6. Shadcn/ui

Shadcn/UI là một thư viện giao diện người dùng (UI) cho các ứng dụng web, được xây dựng với mục tiêu mang lại sự linh hoạt, khả năng tùy chỉnh cao và dễ dàng tích hợp vào các dự án. Thay vì là một framework toàn diện, Shadcn/UI tập trung vào việc cung cấp các thành phần UI dưới dạng mã nguồn sẵn sàng để chỉnh sửa và sử dụng giúp các nhà phát triển có thể dễ dàng quản lý giao diện người dùng mà không bị ràng buộc vào một hệ thống cấu trúc cố định. Dưới đây là mô tả chi tiết về các tính năng và lợi ích của Shadcn/UI:

- *Cung cấp mã nguồn trực tiếp*: Thay vì cung cấp các thành phần UI dưới dạng gói phần mềm đóng gói sẵn, Shadcn/UI cung cấp mã nguồn hoàn chỉnh của các thành phần. Điều này cho phép nhà phát triển có toàn quyền kiểm soát và chỉnh sửa các thành phần này theo nhu cầu của dự án. Người dùng không bị phụ thuộc vào các quyết định thiết kế hay hạn chế từ nhà cung cấp thư viện và có thể thoải mái tùy chỉnh mọi yếu tố của giao diện, từ màu sắc, kiểu dáng cho đến hành vi.

- *Khả năng tùy chỉnh cao*: Với mã nguồn có sẵn và khả năng chỉnh sửa linh hoạt, Shadcn/UI cho phép các nhà phát triển tạo ra các giao diện người dùng theo đúng mong muốn của họ mà không bị giới hạn bởi các thiết kế mặc định. Điều này rất quan trọng trong việc xây dựng các ứng dụng web yêu cầu giao diện người dùng độc đáo và không muốn phụ thuộc vào các thư viện UI có tính gò bó.

- *Sử dụng tốt với Tailwind CSS*: Shadcn/UI được xây dựng với khả năng tích hợp tốt với Tailwind CSS, một framework phổ biến giúp dễ dàng tùy chỉnh giao diện thông qua

việc sử dụng các lớp tiện ích (utility classes). Điều này tạo ra sự kết hợp mạnh mẽ giúp nhà phát triển không chỉ có khả năng quản lý thành phần UI mà còn dễ dàng kiểm soát và tối ưu hóa giao diện theo phong cách mà họ mong muốn.

- *Tối ưu hóa hiệu suất*: Một ưu điểm của Shadcn/UI là tối ưu hóa cho hiệu suất cao trong các ứng dụng web hiện đại. Vì các thành phần UI được cung cấp dưới dạng mã nguồn và không yêu cầu các phụ thuộc nặng nề, các ứng dụng sử dụng Shadcn/UI có thể chạy nhanh hơn và tải trang hiệu quả hơn. Điều này giúp cải thiện trải nghiệm người dùng, đặc biệt khi ứng dụng yêu cầu tải nhiều thành phần UI hoặc có lượng truy cập lớn.

- *Không có phụ thuộc nặng nề*: Shadcn/UI không ép buộc người dùng phải tuân theo một hệ thống cấu trúc hoặc các gói phần mềm phụ thuộc cố định như nhiều thư viện UI khác. Điều này mang lại sự linh hoạt trong việc xây dựng ứng dụng mà không cần lo lắng về sự phụ thuộc vào các phiên bản của bên thứ ba hoặc nguy cơ gặp các vấn đề về tương thích trong tương lai. Nhà phát triển có thể tự do chọn các thư viện hoặc công cụ khác phù hợp với dự án mà không lo về sự xung đột.

- *Hỗ trợ cộng đồng và phát triển mã nguồn mở*: Là một thư viện mã nguồn mở, Shadcn/UI được xây dựng và phát triển dựa trên sự đóng góp của cộng đồng. Điều này mang lại nhiều lợi ích như khả năng cập nhật nhanh chóng, các tính năng mới liên tục được cải tiến và mở rộng. Đồng thời, cộng đồng sử dụng Shadcn/UI cũng chia sẻ nhiều hướng dẫn, ví dụ cụ thể giúp người dùng dễ dàng triển khai và tối ưu hóa các thành phần UI cho dự án của mình.

- *Thích hợp cho các dự án có yêu cầu cao về tùy biến UI*: Shadcn/UI là lựa chọn lý tưởng cho các dự án yêu cầu tính tùy biến cao và không muốn bị giới hạn bởi các giao diện UI cố định. Với việc cung cấp các thành phần dưới dạng mã nguồn và hỗ trợ tùy chỉnh dễ dàng, các nhà phát triển có thể tự do thiết kế giao diện theo đúng yêu cầu của khách hàng hoặc dự án.

Nói tóm lại, Shadcn/UI là một thư viện giao diện người dùng tập trung vào sự linh hoạt, khả năng tùy chỉnh cao và tối ưu hóa hiệu suất. Với mã nguồn mở và tích hợp tốt với Tailwind CSS, nó mang lại cho nhà phát triển khả năng kiểm soát hoàn toàn các thành phần UI giúp dễ dàng tùy chỉnh và mở rộng theo nhu cầu của dự án. Shadcn/UI phù hợp với các dự án có yêu cầu cao về giao diện người dùng độc đáo và hiệu suất, đồng thời hỗ trợ mạnh

mẽ từ cộng đồng mã nguồn mở.

2.7.7. *TypeScript*

TypeScript là một ngôn ngữ lập trình mã nguồn mở được phát triển bởi Microsoft, là một phiên bản mở rộng của JavaScript. TypeScript bổ sung thêm các tính năng mạnh mẽ như hệ thống kiểu tĩnh (static typing), khả năng hỗ trợ lập trình hướng đối tượng và các công cụ phát triển mạnh mẽ hơn giúp làm việc với JavaScript trở nên dễ dàng, an toàn và có cấu trúc hơn. Mục tiêu chính của TypeScript là giúp các nhà phát triển xây dựng các ứng dụng lớn, dễ bảo trì và mở rộng hơn so với JavaScript thuận tủy.

Tính năng chính của TypeScript:

- *Siêu ngôn ngữ của JavaScript*: TypeScript mở rộng cú pháp và chức năng của JavaScript, đồng thời vẫn tương thích hoàn toàn với mã JavaScript. Nhờ khả năng chuyển đổi mã TypeScript về JavaScript thông qua quá trình biên dịch, ngôn ngữ này có thể sử dụng trong mọi môi trường mà JavaScript được hỗ trợ. Điều này cho phép các dự án hiện có chuyển đổi dần dần sang TypeScript mà không gây gián đoạn lớn.

- *Hệ thống kiểu tĩnh (Static Typing)*: Một trong những đặc điểm cốt lõi của TypeScript là hệ thống kiểu tĩnh. Khác với JavaScript, nơi kiểu của biến được xác định tại thời gian chạy (runtime), TypeScript cho phép các lập trình viên chỉ định kiểu ngay tại thời gian biên dịch. Việc này giúp giảm thiểu các lỗi liên quan đến kiểu, nâng cao độ tin cậy của mã nguồn, đặc biệt trong các dự án lớn, phức tạp. Điều này cũng cho phép các công cụ phát triển hỗ trợ tốt hơn thông qua việc cung cấp IntelliSense và tự động hoàn thành.

- *Lập trình hướng đối tượng*: TypeScript bổ sung các khái niệm của lập trình hướng đối tượng như lớp (classes), giao diện (interfaces), kế thừa (inheritance) và tính đóng gói (encapsulation). Điều này tạo điều kiện cho việc thiết kế phần mềm có tính cấu trúc tốt hơn và dễ bảo trì hơn, đặc biệt trong các hệ thống lớn. So với JavaScript, vốn chỉ hỗ trợ lập trình hướng đối tượng thông qua cú pháp dựa trên prototype, TypeScript mang đến mô hình lập trình hướng đối tượng rõ ràng và quen thuộc hơn với các lập trình viên từ các ngôn ngữ như Java hay C#.

- *Generics*: TypeScript cung cấp Generics, một tính năng mạnh mẽ cho phép lập trình viên định nghĩa các hàm, lớp và giao diện với các tham số kiểu (type parameters). Tính năng này cho phép mã linh hoạt hơn mà không đánh mất tính an toàn về kiểu. Generics

giúp các thành phần có thể hoạt động với nhiều kiểu dữ liệu khác nhau mà không cần viết lại mã cho từng loại cụ thể.

Một số thách thức của TypeScript:

- *Độ phức tạp trong quá trình thiết lập*: Mặc dù TypeScript mang lại nhiều lợi ích trong việc phát triển các ứng dụng lớn, việc cấu hình và thiết lập ban đầu cho TypeScript có thể đòi hỏi nhiều công sức hơn so với JavaScript thuận túy. Quá trình biên dịch TypeScript về JavaScript, quản lý các tệp khai báo (declaration files) và cấu hình môi trường phát triển có thể là những thách thức cho các dự án nhỏ hoặc với các lập trình viên mới làm quen.

- *Phù hợp với các dự án nghiêm ngặt*: Trong những dự án nhỏ, nơi sự phức tạp của mã nguồn không lớn, việc sử dụng TypeScript có thể làm tăng độ phức tạp không cần thiết. TypeScript yêu cầu viết nhiều mã hơn so với JavaScript thuận và cần thêm bước biên dịch, điều này có thể không mang lại lợi ích đáng kể đối với các dự án ngắn hạn hoặc ít tính phức tạp.

TypeScript đặc biệt hữu ích trong các dự án phát triển ứng dụng web quy mô lớn. Nó được sử dụng rộng rãi trong các framework và thư viện hiện đại như Angular, React và Vue.js. Trong Angular, TypeScript là ngôn ngữ chính thức giúp phát triển các ứng dụng web phức tạp với sự hỗ trợ mạnh mẽ từ hệ thống kiểu. Với React và Vue.js, TypeScript cũng được sử dụng để tăng cường tính an toàn về kiểu và giảm thiểu lỗi phát sinh trong quá trình phát triển.

Ngoài ra, TypeScript cũng phù hợp với các dự án back-end, nơi cần tính ổn định cao và sự hỗ trợ của các tính năng lập trình hướng đối tượng. Các môi trường như Node.js hoàn toàn có thể sử dụng TypeScript để cải thiện chất lượng và khả năng bảo trì mã nguồn.

2.7.8. *Node.js*

Node.js là một môi trường chạy JavaScript phía máy chủ (server-side) dựa trên nền tảng V8 JavaScript Engine của Google Chrome. Được phát triển bởi Ryan Dahl vào năm 2009, Node.js cho phép các nhà phát triển viết mã JavaScript để chạy trên máy chủ, mở rộng khả năng của JavaScript từ việc chỉ chạy trên trình duyệt web sang việc xây dựng các ứng dụng máy chủ (server-side applications).

Đặc điểm chính của Node.js:

- *Nền tảng phi đồng bộ và không chặn (Asynchronous and Non-blocking)*: Node.js sử dụng mô hình phi đồng bộ và không chặn, cho phép xử lý nhiều yêu cầu đồng thời mà không phải đợi các hoạt động I/O (như đọc/ghi tệp tin, truy vấn cơ sở dữ liệu) hoàn tất. Điều này giúp tăng cường hiệu suất và khả năng mở rộng của các ứng dụng web, đặc biệt là trong các ứng dụng cần xử lý lượng lớn dữ liệu hoặc nhiều kết nối đồng thời.

- *Mô hình sự kiện (Event-driven)*: Node.js dựa trên mô hình sự kiện, trong đó các hoạt động như xử lý yêu cầu HTTP, đọc tệp tin và truy vấn cơ sở dữ liệu được thực hiện qua các sự kiện và callback functions. Điều này giúp quản lý hiệu quả các tác vụ bất đồng bộ và xử lý nhiều kết nối đồng thời mà không làm giảm hiệu suất của ứng dụng.

- *Cung cấp một môi trường chạy JavaScript trên máy chủ*: Trước khi Node.js ra đời, JavaScript chủ yếu được sử dụng trên trình duyệt web. Node.js mở rộng khả năng của JavaScript để chạy trên máy chủ, cho phép phát triển các ứng dụng web, dịch vụ API và các ứng dụng mạng một cách đồng nhất bằng một ngôn ngữ lập trình duy nhất.

- *Quản lý gói qua npm (Node Package Manager)*: Node.js đi kèm với npm, một hệ thống quản lý gói mạnh mẽ cho phép các nhà phát triển dễ dàng cài đặt, quản lý và chia sẻ các thư viện và công cụ. npm có một kho lưu trữ khổng lồ với hàng triệu gói phần mềm giúp tiết kiệm thời gian phát triển và tăng cường khả năng mở rộng của các ứng dụng.

- *Mô hình đơn luồng (Single-threaded)*: Node.js sử dụng mô hình đơn luồng với vòng lặp sự kiện (event loop) để xử lý các tác vụ. Mặc dù Node.js chỉ sử dụng một luồng chính nhưng nó có thể xử lý hàng triệu kết nối đồng thời nhờ vào cơ chế phi đồng bộ và không chặn.

- *Hiệu suất cao với V8 Engine*: Node.js dựa trên V8 JavaScript Engine của Google, nổi tiếng với hiệu suất cao trong việc thực thi mã JavaScript. V8 thực hiện biên dịch mã JavaScript thành mã máy (machine code) giúp tăng tốc việc thực thi và cải thiện hiệu suất tổng thể của ứng dụng.

Các ứng dụng phổ biến của Node.js:

- *Ứng dụng web*: Node.js rất phù hợp để phát triển các ứng dụng web, bao gồm cả các trang web tĩnh và động, nhờ vào khả năng xử lý nhiều yêu cầu đồng thời và hỗ trợ các thư viện web như Express.js.

- *Dịch vụ API*: Node.js được sử dụng để xây dựng các dịch vụ API RESTful hoặc GraphQL, nhờ vào khả năng xử lý các yêu cầu HTTP và tương tác với cơ sở dữ liệu một cách hiệu quả.

- *Ứng dụng mạng thời gian thực*: Node.js thường được sử dụng trong các ứng dụng mạng thời gian thực như chat rooms, gaming servers và các ứng dụng video streaming, nhờ vào khả năng xử lý các kết nối đồng thời và sự kiện.

- *Công cụ dòng lệnh (CLI tools)*: Node.js cũng được sử dụng để xây dựng các công cụ dòng lệnh mạnh mẽ, nhờ vào khả năng tương tác với hệ thống tệp tin và thực thi các lệnh từ dòng lệnh.

Node.js đã thay đổi cách các nhà phát triển xây dựng và triển khai ứng dụng máy chủ bằng cách mang lại một môi trường phát triển đồng nhất và hiệu suất cao. Với khả năng xử lý đồng thời tốt, mô hình sự kiện mạnh mẽ và hệ sinh thái phong phú, Node.js là một lựa chọn lý tưởng cho việc phát triển các ứng dụng web hiện đại, dịch vụ API và nhiều ứng dụng khác trong môi trường máy chủ.

2.7.9. Express

Express hay Express.js là một framework ứng dụng web cho Node.js, được phát triển để tạo ra các ứng dụng web và dịch vụ API một cách dễ dàng và nhanh chóng. Được phát hành lần đầu tiên vào năm 2010, Express đã trở thành một trong những framework phổ biến nhất cho Node.js nhờ vào sự đơn giản, hiệu quả và tính mở rộng của nó.

Đặc điểm nổi bật của Express.js:

- *Thiết kế đơn giản và nhẹ nhàng*: Express cung cấp một cấu trúc nhẹ nhàng và đơn giản giúp các nhà phát triển nhanh chóng thiết lập và triển khai các ứng dụng web và dịch vụ API. Với ít cấu hình cần thiết, Express giúp giảm bớt sự phức tạp trong việc phát triển và quản lý ứng dụng, tập trung vào việc cung cấp các tính năng cơ bản và linh hoạt để phát triển ứng dụng.

- *Mô hình Middleware mạnh mẽ*: Một trong những tính năng mạnh mẽ của Express là hệ thống middleware. Middleware là các hàm xử lý yêu cầu HTTP (request) và phản hồi (response) mà có thể được chèn vào giữa các bước xử lý của một ứng dụng. Điều này cho phép các nhà phát triển thực hiện các tác vụ như xác thực, xử lý lỗi, phân tích dữ liệu và

nhiều chức năng khác một cách dễ dàng và có tổ chức. Middleware có thể được sử dụng toàn cục hoặc cho các tuyến đường (routes) cụ thể, mang lại khả năng kiểm soát và tùy chỉnh cao.

- *Quản lý tuyến đường (Routing)*: Express cung cấp một hệ thống quản lý tuyến đường linh hoạt, cho phép định nghĩa các tuyến đường HTTP cho các phương thức như GET, POST, PUT, DELETE và các phương thức khác. Chúng ta có thể định nghĩa các tuyến đường với các tham số động và xử lý các yêu cầu từ người dùng một cách rõ ràng và có tổ chức. Express hỗ trợ việc phân chia các tuyến đường thành các tệp riêng biệt giúp quản lý mã nguồn trở nên dễ dàng hơn.

- *Tính mở rộng cao*: Express có khả năng mở rộng tốt thông qua việc tích hợp với các thư viện và công cụ khác. Chúng ta có thể dễ dàng kết hợp Express với các thư viện như Mongoose để tương tác với MongoDB, Passport để xử lý xác thực người dùng hoặc các công cụ khác để tăng cường chức năng của ứng dụng. Điều này giúp chúng ta xây dựng các ứng dụng phức tạp hơn mà vẫn giữ cho mã nguồn dễ quản lý và bảo trì.

- *Hỗ trợ nhiều engine template*: Express hỗ trợ nhiều engine template khác nhau như Pug (trước đây là Jade), EJS và Handlebars. Điều này giúp người dùng dễ dàng tạo ra các trang web động bằng cách kết hợp dữ liệu từ máy chủ với các mẫu giao diện. Việc tích hợp các engine template giúp tăng cường khả năng phát triển giao diện người dùng một cách linh hoạt và hiệu quả.

- *Quản lý yêu cầu và phản hồi (Request/Response)*: Express cung cấp các đối tượng yêu cầu (request) và phản hồi (response) để sử dụng giúp xử lý dữ liệu gửi từ người dùng và gửi dữ liệu phản hồi một cách hiệu quả. Chúng ta có thể dễ dàng truy cập các tham số của yêu cầu, xử lý dữ liệu từ biểu mẫu và gửi phản hồi với các mã trạng thái HTTP khác nhau, tiêu đề và nội dung.

- *Hỗ trợ xử lý lỗi (Error Handling)*: Express cung cấp cơ chế xử lý lỗi tích hợp, cho phép người dùng dễ dàng quản lý và phản hồi các lỗi xảy ra trong ứng dụng. Chúng ta có thể định nghĩa các hàm xử lý lỗi middleware để xử lý các lỗi phát sinh trong các bước xử lý yêu cầu, từ đó cung cấp thông tin lỗi chi tiết và thân thiện với người dùng.

- *Tích hợp với cơ sở dữ liệu*: Express có thể được tích hợp với nhiều hệ quản trị cơ sở dữ liệu (DBMS) khác nhau như MongoDB, MySQL, PostgreSQL và SQLite thông qua các

thư viện và ORM (Object-Relational Mapping). Điều này giúp dễ dàng kết nối và tương tác với cơ sở dữ liệu, thực hiện các thao tác như lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả.

Lợi ích của Express:

- *Nhanh chóng và dễ dàng để bắt đầu*: Với một cấu trúc đơn giản, Express.js được thiết kế với cấu trúc tối giản và linh hoạt, không yêu cầu cấu hình phức tạp hoặc thiết lập môi trường nặng nề. Các nhà phát triển có thể dễ dàng tạo ra một ứng dụng web cơ bản chỉ với vài dòng mã. Điều này giúp giảm thời gian học tập và triển khai cho những người mới bắt đầu, đồng thời giúp các dự án nhỏ được triển khai nhanh chóng.

- *Tài liệu phong phú*: Express cung cấp tài liệu chi tiết và dễ hiểu, bao gồm hướng dẫn sử dụng, các ví dụ mã nguồn và các bài viết giải thích về các tính năng chính. Điều này giúp các nhà phát triển nắm bắt nhanh chóng cách sử dụng các tính năng của Express và giải quyết các vấn đề mà họ gặp phải trong quá trình phát triển. Tài liệu phong phú giúp các nhà phát triển, đặc biệt là những người mới, có thể nhanh chóng làm quen và áp dụng Express vào dự án của mình.

- *Hỗ trợ nhiều middleware*: Express.js cho phép sử dụng các middleware để xử lý yêu cầu và phản hồi, thực hiện các tác vụ như xác thực, xử lý lỗi và phân tích dữ liệu. Các middleware có thể được sử dụng để thêm chức năng mới vào ứng dụng mà không cần phải thay đổi mã nguồn chính. Điều này giúp ứng dụng có thể mở rộng dễ dàng theo nhu cầu, chẳng hạn như thêm các tính năng bảo mật hoặc phân tích người dùng.

- *Dễ dàng tích hợp với thư viện và công cụ khác*: Express có khả năng tích hợp với nhiều thư viện và công cụ khác, chẳng hạn như cơ sở dữ liệu, hệ thống xác thực và các công cụ phân tích. Chúng ta có thể sử dụng các thư viện bên ngoài như Mongoose để kết nối và tương tác với MongoDB, Passport để xử lý xác thực người dùng hoặc các công cụ khác để nâng cao chức năng của ứng dụng. Điều này giúp người dùng có thể xây dựng các ứng dụng phức tạp và tùy chỉnh theo nhu cầu cụ thể của dự án.

- *Quản lý yêu cầu và phản hồi hiệu quả*: Nhờ vào sự đơn giản trong cách Express xử lý các yêu cầu và phản hồi, việc xây dựng các API và ứng dụng web trở nên trực quan và dễ quản lý. Chúng ta có thể dễ dàng định nghĩa các tuyến đường (routes) và các hàm xử lý (handlers) để xử lý các yêu cầu từ người dùng giúp giảm thiểu sự phức tạp và tăng hiệu

qua phát triển ứng dụng.

- *Cộng đồng lớn*: Express.js có một cộng đồng người dùng và nhà phát triển rộng lớn và năng động. Cộng đồng này đóng góp vào việc phát triển và duy trì Express, đồng thời cung cấp hỗ trợ qua các diễn đàn, nhóm thảo luận và các mạng xã hội. Việc có một cộng đồng lớn giúp đảm bảo rằng có thể dễ dàng tìm kiếm sự trợ giúp và giải pháp cho các vấn đề người dùng gặp phải.

- *Tài liệu phong phú và hỗ trợ*: Tài liệu của Express bao gồm hướng dẫn chi tiết, các ví dụ mã nguồn và các bài viết hướng dẫn từ cơ bản đến nâng cao. Ngoài ra, nhiều trang web, blog và khóa học trực tuyến cung cấp các hướng dẫn và mẹo bổ sung để giúp lập trình viên khai thác tối đa các tính năng của Express. Việc có tài liệu phong phú và sự hỗ trợ từ cộng đồng giúp giảm bớt khó khăn trong việc học và phát triển ứng dụng, đồng thời cung cấp nhiều nguồn tài nguyên để giải quyết các vấn đề kỹ thuật.

Express phát huy được sức mạnh và sự linh hoạt của mình trong việc xây dựng các ứng dụng web và dịch vụ API bằng Node.js. Với thiết kế đơn giản, khả năng mở rộng cao và tính dễ sử dụng, Express là một lựa chọn lý tưởng cho các nhà phát triển khi xây dựng các ứng dụng từ nhỏ đến lớn. Việc kết hợp Express với Node.js cung cấp một nền tảng mạnh mẽ và hiệu quả cho việc phát triển các ứng dụng máy chủ hiện đại.

2.7.10. *Socket.IO*

Socket.IO là một thư viện mạnh mẽ được thiết kế để tạo ra các ứng dụng thời gian thực (real-time applications) với giao tiếp hai chiều (bi-directional communication) giữa máy khách và máy chủ. Nó dựa trên cơ chế WebSocket nhưng bổ sung thêm nhiều tính năng khác để hỗ trợ nhiều trường hợp sử dụng, bao gồm độ tin cậy cao hơn, tự động khôi phục kết nối và khả năng mở rộng (scalability). Socket.IO có thể được sử dụng với nhiều ngôn ngữ lập trình và trong nhiều môi trường như Node.js, Python, Java và các ứng dụng web.

Các thành phần chính của Socket.IO:

- *Server-side (Máy chủ)*: Socket.IO chạy trên máy chủ và quản lý tất cả các kết nối giữa máy khách và máy chủ. Trong môi trường Node.js, Socket.IO được cài đặt và tích hợp dễ dàng với các framework như Express.

- *Client-side (Máy khách)*: Socket.IO cung cấp một thư viện trên phía máy khách,

thường được nhúng vào trang web thông qua một tập tin JavaScript, để kết nối với máy chủ và gửi nhận dữ liệu theo thời gian thực.

Tính năng của Socket.IO:

- *Giao tiếp thời gian thực*: Socket.IO cung cấp giao tiếp theo thời gian thực giữa máy chủ và máy khách, cho phép dữ liệu được truyền ngay lập tức mà không cần phải tải lại trang web.

- *Kết nối hai chiều (Bidirectional communication)*: Cả máy khách và máy chủ đều có thể gửi dữ liệu cho nhau một cách chủ động. Điều này có nghĩa là không chỉ máy khách có thể yêu cầu dữ liệu từ máy chủ mà máy chủ cũng có thể đẩy thông tin đến máy khách khi cần thiết.

- *Đa giao thức (Multiprotocol)*: Mặc dù Socket.IO sử dụng WebSocket làm giao thức chính, nó còn hỗ trợ các phương pháp dự phòng như long polling (lấy dữ liệu liên tục bằng HTTP). Điều này đảm bảo rằng ngay cả khi trình duyệt hoặc mạng không hỗ trợ WebSocket, ứng dụng vẫn có thể hoạt động.

- *Tự động khôi phục kết nối (Auto-reconnection)*: Nếu kết nối giữa máy khách và máy chủ bị gián đoạn, Socket.IO sẽ tự động cố gắng khôi phục kết nối mà không cần sự can thiệp của người dùng.

- *Phòng và không gian tên (Rooms and Namespaces)*: Namespaces giúp phân chia không gian giữa các nhóm kết nối khác nhau, cho phép ứng dụng tổ chức các logic phức tạp hơn như chat, trò chơi đa người chơi hoặc các ứng dụng công cộng và riêng tư. Rooms là các nhóm con trong một namespace. Mỗi client có thể tham gia một hoặc nhiều phòng và các tin nhắn chỉ được gửi tới các client trong cùng phòng.

- *Sự kiện tùy chỉnh (Custom events)*: Socket.IO hỗ trợ việc định nghĩa các sự kiện tùy chỉnh trên cả máy khách và máy chủ, cho phép người dùng dễ dàng mở rộng giao tiếp với những kiểu dữ liệu khác nhau.

- *Tính năng phát sóng (Broadcasting)*: Socket.IO cho phép một client gửi một thông điệp tới tất cả các client khác được kết nối cùng máy chủ (trong cùng một phòng hoặc toàn bộ hệ thống).

Socket.IO hoạt động trên mô hình client-server. Máy khách sẽ tạo một kết nối tới máy

chủ bằng cách gửi yêu cầu kết nối và máy chủ sẽ lắng nghe và quản lý các kết nối này.

- *Kết nối*: Khi một client kết nối với máy chủ, nó sẽ thiết lập một kết nối vĩnh viễn bằng cách sử dụng WebSocket hoặc các phương pháp dự phòng.

- *Trao đổi sự kiện*: Sau khi kết nối được thiết lập, máy chủ và máy khách có thể trao đổi sự kiện với nhau. Mỗi sự kiện có thể chứa dữ liệu kèm theo.

- *Ngắt kết nối*: Nếu kết nối bị mất (do mạng hoặc do lỗi), Socket.IO có thể tự động khôi phục kết nối.

2.7.11. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp để quản lý và lưu trữ dữ liệu. Được phát triển từ đầu những năm 1990 bởi MySQL AB, MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất nhờ hiệu suất cao, tính ổn định và dễ dàng sử dụng. Hiện tại, MySQL thuộc sở hữu của Oracle Corporation sau khi mua lại Sun Microsystems vào năm 2010. Dưới đây là mô tả chi tiết về các tính năng và lợi ích của MySQL:

- *Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS)*: MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, nghĩa là nó sử dụng bảng (table) để lưu trữ dữ liệu và các bảng này có thể liên kết với nhau thông qua các khóa (keys) và quan hệ (relationships). Mô hình quan hệ giúp cho việc tổ chức và truy vấn dữ liệu trở nên dễ dàng và hiệu quả hơn. Cấu trúc dữ liệu dạng bảng rất phù hợp cho việc quản lý dữ liệu có cấu trúc rõ ràng, chẳng hạn như thông tin khách hàng, đơn đặt hàng hoặc dữ liệu sản phẩm trong các hệ thống thương mại điện tử.

- *Khả năng mở rộng và hiệu suất cao*: MySQL nổi tiếng với khả năng xử lý khối lượng dữ liệu lớn và đáp ứng các yêu cầu truy vấn phức tạp một cách nhanh chóng. Nhờ khả năng tối ưu hóa mạnh mẽ, MySQL có thể hoạt động tốt trên các ứng dụng từ quy mô nhỏ đến quy mô lớn, từ các trang web cá nhân đến các hệ thống thương mại điện tử lớn và dịch vụ tài chính phức tạp. Ngoài ra, MySQL cũng có thể được sử dụng trong môi trường phân tán giúp tăng khả năng mở rộng và cung cấp hiệu suất tốt hơn khi xử lý các yêu cầu từ nhiều người dùng.

- *Hỗ trợ đa nền tảng*: MySQL có khả năng hoạt động trên nhiều hệ điều hành khác

nhau, bao gồm Linux, Windows, macOS và các hệ điều hành khác. Điều này giúp MySQL dễ dàng tích hợp vào các hệ thống hiện có mà không cần phải thay đổi nhiều về cơ sở hạ tầng. Khả năng đa nền tảng của MySQL là một yếu tố quan trọng làm tăng tính linh hoạt và tiện lợi trong việc triển khai các giải pháp phần mềm.

- *Tính toàn vẹn dữ liệu và bảo mật cao*: MySQL cung cấp các tính năng mạnh mẽ để đảm bảo tính toàn vẹn của dữ liệu, bao gồm việc hỗ trợ các giao dịch (transactions), khóa ngoại (foreign keys) và các quy tắc ràng buộc (constraints). Ngoài ra, MySQL cung cấp cơ chế bảo mật mạnh mẽ để kiểm soát quyền truy cập vào cơ sở dữ liệu, bao gồm việc sử dụng xác thực người dùng và phân quyền. Điều này đảm bảo rằng chỉ có những người dùng được ủy quyền mới có thể truy cập và thao tác trên dữ liệu nhạy cảm.

- *Hỗ trợ SQL và các tính năng nâng cao*: MySQL sử dụng Structured Query Language (SQL), ngôn ngữ tiêu chuẩn cho việc truy vấn và quản lý cơ sở dữ liệu. SQL giúp người dùng dễ dàng thực hiện các thao tác như thêm, xóa, sửa đổi và truy xuất dữ liệu. MySQL cũng hỗ trợ các tính năng nâng cao của SQL như các hàm tổng hợp (aggregate functions), chỉ mục (indexes), views, stored procedures và triggers giúp tối ưu hóa quá trình xử lý dữ liệu và thực hiện các tác vụ phức tạp hơn.

- *Sao lưu và phục hồi dữ liệu*: MySQL cung cấp nhiều công cụ và tính năng để sao lưu và phục hồi dữ liệu một cách hiệu quả. Các công cụ như mysqldump hoặc các giải pháp phục hồi từ tập tin sao lưu giúp bảo đảm tính liên tục của dịch vụ trong trường hợp xảy ra lỗi hệ thống hoặc mất dữ liệu. Khả năng sao lưu và phục hồi dễ dàng giúp MySQL trở thành lựa chọn tin cậy cho các ứng dụng yêu cầu tính sẵn sàng cao.

- *Cộng đồng và hỗ trợ rộng rãi*: MySQL là một dự án mã nguồn mở với cộng đồng người dùng và nhà phát triển rộng lớn. Điều này giúp cung cấp nhiều tài nguyên hỗ trợ bao gồm tài liệu, hướng dẫn, diễn đàn và các ví dụ thực hành. Sự hỗ trợ từ cộng đồng giúp người dùng dễ dàng tìm giải pháp cho các vấn đề gặp phải và liên tục cải tiến dự án dựa trên phản hồi của người dùng.

- *Tính tương thích cao với các ngôn ngữ lập trình*: MySQL có thể dễ dàng tích hợp với nhiều ngôn ngữ lập trình phổ biến như PHP, Java, Python, Ruby và .NET. Điều này giúp các nhà phát triển dễ dàng xây dựng và triển khai các ứng dụng web và di động mà không gặp khó khăn trong việc tích hợp cơ sở dữ liệu. Khả năng tương thích này làm cho MySQL

trở thành một trong những sự lựa chọn phổ biến nhất cho các dự án phát triển phần mềm hiện nay.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả bài toán

Hệ thống HiCommit là một hệ thống đánh giá lập trình trực tuyến, được tích hợp nhiều chức năng như khoá học, luyện tập, cuộc thi, diễn đàn. Hệ thống sử dụng GitHub Actions làm nền tảng trong việc thực hiện các tác vụ tự động như chấm và đưa ra kết quả.

Hệ thống sử dụng phương thức đăng nhập duy nhất là sử dụng tài khoản GitHub để đăng nhập. Sau khi đăng nhập, giao diện sẽ hiển thị các chức năng theo từng vai trò người dùng. Có ba vai trò người dùng: quản trị viên, giảng viên, sinh viên. Mỗi người dùng sẽ chỉ được sử dụng các chức năng tương ứng với từng vai trò.

Người dùng có thể tham gia các khoá học trên hệ thống. Với mỗi khoá học, có nhiều bài tập thực hành được xây dựng theo lộ trình của khoá học, học viên cần hoàn thành các bài tập này để hoàn thành khóa học. Các bài tập đều có các thông tin như mô tả đề bài, mô tả dữ liệu đầu vào, đầu ra, các ví dụ,... Nếu người dùng có bất kỳ thắc mắc hay câu hỏi nào, có thể sử dụng chức năng thảo luận mọi người hỗ trợ lẫn nhau.

Ngoài các bài tập trong khoá học, hệ thống cũng cung cấp một khu vực dành riêng để sinh viên có thể luyện tập tự do. Sau khi hoàn thành các bài tập này, kết quả sẽ được ghi nhận, điểm của bài tập này sẽ được cộng dồn vào điểm tích luỹ của người dùng. Các bài tập khác nhau có mức điểm số khác nhau, được phân loại thành ba mức độ khó: dễ, trung bình, khó. Bài có mức độ khó càng cao, điểm sẽ càng lớn. Người dùng có thể theo dõi thứ hạng của mình thông qua chức năng Bảng xếp hạng.

Chức năng cuộc thi được xây dựng nhằm đáp ứng nhu cầu của các cuộc thi về lập trình. Trong tự khoá học, mỗi cuộc thi sẽ có nhiều bài toán, người tham gia cần giải các bài toán này để tối ưu hoá thành tích. Cuộc thi chỉ tồn tại trong một khoảng thời gian, từ thời gian bắt đầu và diễn ra một khoảng thời gian nhất định. Sau khoảng thời gian này, người tham gia không thể truy cập vào đề bài. Trong thời gian diễn ra cuộc thi, thông tin về thành tích của những người tham gia sẽ được cập nhật liên tục lên bảng xếp hạng giúp người tham gia và các người dùng khác dễ dàng nắm được tình hình cuộc thi.

Người dùng có thể đăng các bài viết về các chủ đề liên quan đến công nghệ, thuật toán, kiến thức,... hoặc các chủ đề phù hợp lên hệ thống. Bài viết này cần được người quản trị duyệt trước khi xuất hiện trên phần diễn đàn. Khi bài viết đã được thông qua, những người

dùng khác có thể xem và đọc được bài viết này.

Ngoài ra, cần có giao diện để người quản trị hệ thống có thể dễ dàng theo dõi và quản trị toàn bộ hệ thống.

3.2. Phân tích yêu cầu và lựa chọn công nghệ

3.2.1. Phân tích yêu cầu

Dựa vào yêu cầu của đề bài, hệ thống sẽ có ba vai trò người dùng là quản trị viên, giảng viên, sinh viên. Các chức năng ứng với từng phân quyền được ghi rõ trong bảng 3.1.

Bảng 3.1. Các chức năng trong hệ thống HiCommit

STT	Chức năng		Sinh viên	Giảng viên	Quản trị
1	Đăng nhập		✓	✓	✓
2	Luyện tập		✓	✓	✓
3	Thảo luận		✓	✓	✓
4	Cuộc thi		✓	✓	✓
5	Khoa học		✓	✓	✓
6	Bảng xếp hạng		✓	✓	✓
7	Diễn đàn		✓	✓	✓
8	Đăng bài viết (chờ duyệt)		✓	✓	✓
9	Quản lý khóa học	Đã tạo		✓	✓
		Tất cả			✓
10	Quản lý bài tập	Đã tạo trong khóa học		✓	✓
		Tất cả			✓
11	Quản lý người dùng				✓
12	Quản lý bài viết				✓
13	Quản lý cuộc thi				✓

14	Thống kê hệ thống				✓
----	-------------------	--	--	--	---

3.2.2. *Lựa chọn công nghệ*

Các phần chính của hệ thống bao gồm:

- *Front-end*: Giao diện người dùng của trang web, hiển thị thông tin và xử lý các thao tác của người dùng.
- *Back-end*: Cung cấp API, là trung tâm xử lý của hệ thống.
- *Database*: Nơi lưu trữ toàn bộ dữ liệu của trang web.
- *HiCommit CLI (Local)*: Ứng dụng dòng lệnh chạy trên môi trường cục bộ trên máy tính của người dùng, hỗ trợ tương tác với hệ thống HiCommit và Git.
- *GitHub Actions*: Nơi xử lý tự động hóa các tác vụ CI/CD, được quản lý và tương tác thông qua GitHub REST API.
- *LLM*: Mô hình ngôn ngữ lớn, đưa ra các đánh giá và cải thiện về chất lượng mã nguồn. Hoạt động thông qua API được cung cấp.
- Một số thành phần khác.

Các công nghệ được ưu tiên lựa chọn để phù hợp thoả mãn các tiêu chí:

- Mã nguồn mở/Miễn phí.
- Phát triển nhanh chóng, dễ làm quen.
- Hỗ trợ nhiều thư viện, công cụ.
- Cộng đồng người dùng lớn.
- Tài liệu phát triển đầy đủ và rõ ràng.
- Dễ cập nhật, bảo trì.
- Khả năng mở rộng.
- Tối ưu chi phí khi triển khai ở quy mô nhỏ và vừa.

Dựa theo các tiêu chí trên, các công nghệ phù hợp được lựa chọn trong bảng 3.2.

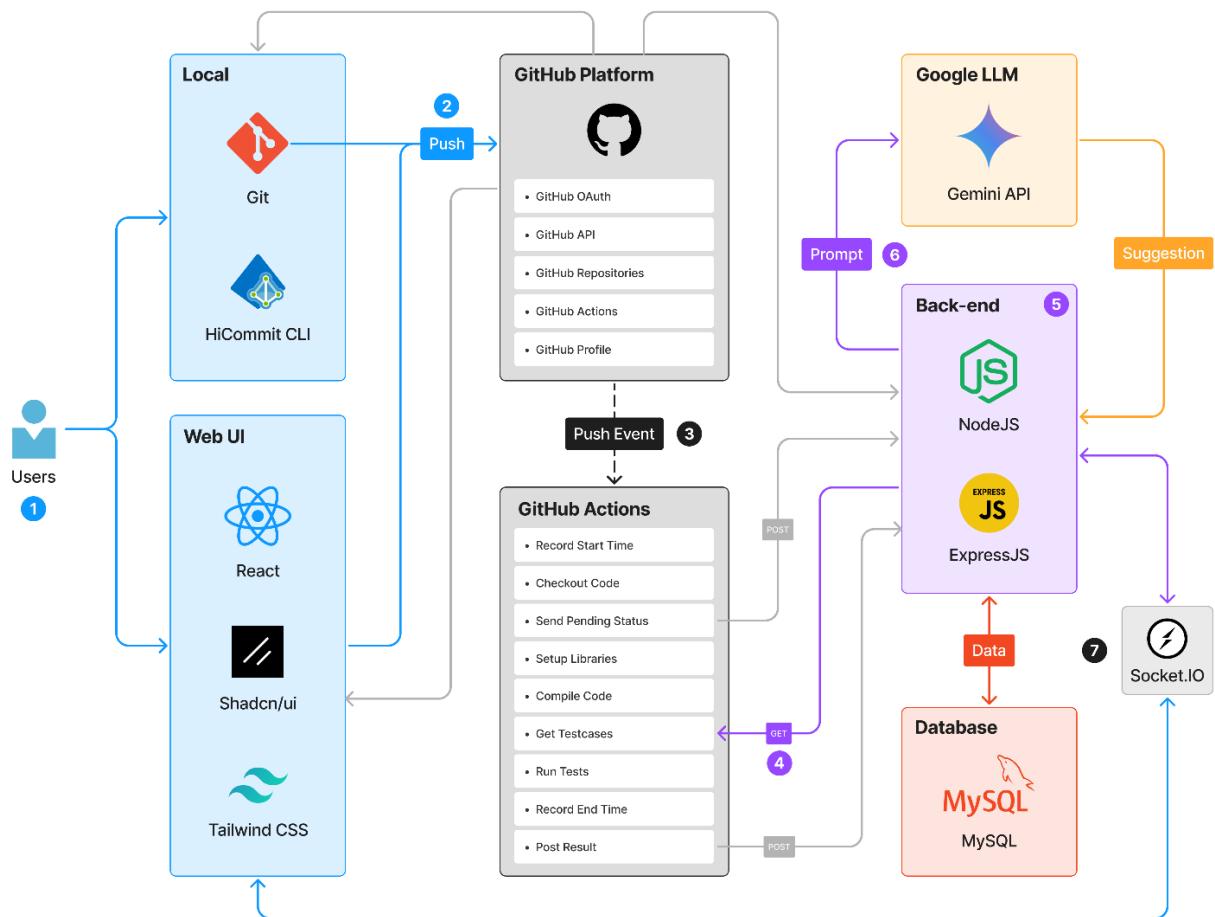
Bảng 3.2. Các công nghệ được lựa chọn sử dụng

STT	Module	Công nghệ/Giải pháp	Diễn giải
1	Front-end	React	Framework JavaScript, tạo khung giao diện nhanh chóng và mượt mà.
		TypeScript	Ngôn ngữ lập trình được phát triển mở rộng từ JavaScript, có cấu trúc chặt chẽ, dễ bảo trì và mở rộng.
		TailwindCSS	Framework CSS giúp đơn giản hoá việc tạo kiểu cho trang web.
		Shadcn/ui	Thư viện UI với các thành phần đẹp mắt được xây dựng sẵn bằng TypeScript, dễ dàng tùy biến với TailwindCSS.
2	Back-end	NodeJS	JavaScript Runtime phía máy chủ, hỗ trợ nhiều thư viện.
		ExpressJS	Thư viện JavaScript dành cho NodeJS để phát triển các API.
		Socket.IO	Thư viện cho phép giao tiếp thời gian thực giữa máy chủ và máy khách.
3	Database	MySQL	Là cơ sở dữ liệu mã nguồn

			mở miễn phí, phổ biến, thường được tích hợp sẵn trong các dịch vụ Cloud như Hosting, VPS,...
4	LLM	Gemini	Mô hình ngôn ngữ lớn được cung cấp miễn phí bởi Google, cho phép tương tác và tạo sinh thông qua Gemini API.
5	GitHub Platform	GitHub REST API	Giao diện lập trình ứng dụng (API) được cung cấp bởi GitHub, cho phép tương tác với các tính năng và dịch vụ của GitHub từ bên ngoài.
		GitHub OAuth	Tính năng cho phép xác thực đăng nhập bằng tài khoản GitHub.
		GitHub Actions	Tính năng hỗ trợ CI/CD được GitHub cung cấp miễn phí.
6	HiCommit CLI (Local)	NodeJS	Ngoài vai trò là ngôn ngữ phía máy chủ, NodeJS còn có thể tạo ra các ứng dụng dòng lệnh, tương tác trực tiếp với hệ thống tệp, tài nguyên của máy tính,...

3.3. Thiết kế kiến trúc hệ thống

3.3.1. Thiết kế kiến trúc phần mềm



Hình 3.1. Kiến trúc phần mềm của HiCommit

Chi tiết cách hoạt động của hệ thống:

Bước 1. Người dùng có thể sử dụng thông qua giao diện web của HiCommit (Web UI) hoặc sử dụng trên cục bộ thông qua HiCommit CLI.

Bước 2. Khi nộp mã nguồn các bài tập:

- *Web UI*: Mã nguồn sẽ được Push lên GitHub Repo thông qua GitHub REST API.
- *Local*: Người dùng Push lên GitHub bằng cách sử dụng lần lượt các câu lệnh Git

```
git add .
git commit -m "Commit Message"
git push
```

Bước 3. Khi Push thành công, sự kiện Push ở nhánh tương ứng sẽ được kích hoạt, các file trong .github/workflows sẽ tự động thực thi.

Bước 4. Trong quá trình thực thi workflow, GitHub Actions sẽ lấy các Test-case tương ứng với bài tập thông qua API từ Back-end. Đồng thời, các dữ liệu về lần nộp bài cũng sẽ được gửi đến và xử lý tại Back-end. Sau khi đã thực thi xong, kết quả của lần chấm bài sẽ được gửi đến Back-end lần nữa để ghi nhận.

Bước 5. Back-end đóng vai trò tương tác và xử lý dữ liệu giữa Front-end, GitHub, Database, LLM. Khi nhận được yêu cầu từ GitHub Actions về mã nguồn cần đánh giá, Back-end gửi mã nguồn này đến LLM, theo cấu trúc Prompt được chỉ định. Các đánh giá từ LLM phản hồi sẽ được lưu trữ vào Database ở dạng JSON để thuận tiện trong việc xử lý sau.

Bước 6. Prompt mẫu được sử dụng với mô hình Gemini nhằm đưa ra các đánh giá về chất lượng mã nguồn và các góp ý có dạng như sau:

[Hãy trả lời toàn bộ bằng tiếng Việt]

[Chỉ làm những gì tôi yêu cầu]

Đưa ra các đánh giá cho đoạn code bên dưới về:

1. Các lỗi cú pháp (nếu có)

2. Các vấn đề trong mã nguồn

3. Style check [Indentation, Whitespace, Newline, Variable and Function Names, Comments, Code Organization, Modularity, Code Style Compliance, Automated Code Checking Tools, Dead Code Elimination, Code Optimization, Consistency Check]

4. Các góp ý cải thiện

Tôi chỉ cần đầu ra có dạng JSON như sau:

{

```
"syntax": [
  {
    "issue": <string><Vietnamese>
    "category": <string>
    "severity": (thang điểm 10)
```

```
},  
{  
    "issue": <string><Vietnamese>  
    "category": <string>  
    "severity": (thang điểm 10)  
,  
...],  
  
"evaluation": [  
{  
    "issue": <string><Vietnamese>  
    "category": <string>  
    "severity": (thang điểm 10)  
,  
{  
    "issue": <string><Vietnamese>  
    "category": <string>  
    "severity": (thang điểm 10)  
,  
...],  
  
"style_check": [  
{  
    "issue": <string><Vietnamese>  
    "category": <string>  
    "severity": (thang điểm 10)  
,  
{
```

```

    "issue": <string><Vietnamese>

    "category": <string>

    "severity": (thang điểm 10)

  },
  ...],
}

"suggestions": [
  <string><Vietnamese>,
  <string><Vietnamese>,
  ...
]

}

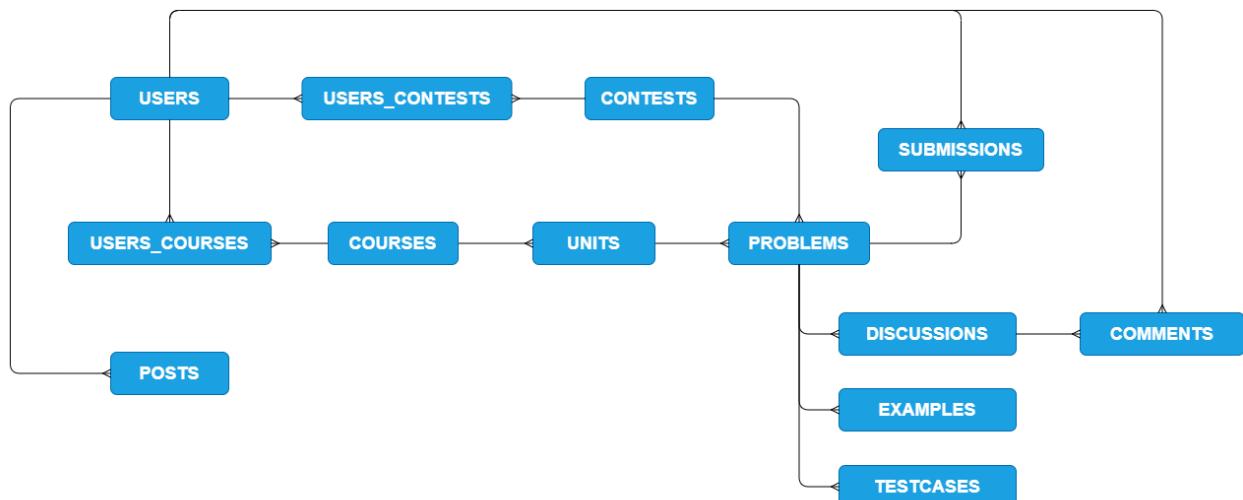
```

CODE:

{My_Code}

Bước 7. Các dữ liệu từ Back-end được cập nhật thời gian thực (realtime) thông qua kết nối giữa Client-Server từ Socket.IO.

3.3.2. Thiết kế cơ sở dữ liệu



Hình 3.2. Lược đồ cơ sở dữ liệu của HiCommit

Chi tiết mỗi bảng sẽ có các thuộc tính, kiểu dữ liệu, ràng buộc được nêu rõ trong các

bảng bên dưới (*từ Bảng 3.3 đến Bảng 3.15*).

* Mô tả bảng USERS (Người dùng):

Bảng 3.3. Bảng mô tả bảng Users

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Điễn giải
1	id	String	Khoá chính	Mã người dùng
2	email	String		Email
3	username	String		Tên người dùng
4	avatar_url	String		Ảnh đại diện của người dùng
5	uid	String		Mã định danh từ GitHub
6	role	ENUM('ADMIN', 'TEACHER', 'STUDENT')		Phân quyền của người dùng
7	status	ENUM('ACTIVE', 'INACTIVE', 'PENDING')		Trạng thái hoạt động của người dùng
8	favourite_post	JSON		Danh sách bài viết yêu thích
9	favourite_course	JSON		Danh sách khóa học yêu thích
10	favourite_problem	JSON		Danh sách bài tập yêu thích

* Mô tả bảng POSTS (Bài viết):

Bảng 3.4. Bảng mô tả bảng Posts

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã bài viết
2	created_by	String	Khoá ngoại	Người tạo bài viết
3	title	String		Tiêu đề bài viết
4	description	String		Mô tả bài viết
5	content	String		Nội dung của bài viết
6	slug	String		Đường dẫn tùy chỉnh của bài viết
7	thumbnail	String		Ảnh thu nhỏ của bài viết
8	tags	JSON		Danh sách chủ đề của bài viết
9	publish	BOOLEAN		Trạng thái công bố của bài viết
10	status	ENUM('ACTIVE', 'INACTIVE')		Trạng thái duyệt của bài viết

* Mô tả bảng COURSES (Khoá học):

Bảng 3.5. Bảng mô tả bảng Courses

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã khoá học
2	created_by	String	Khoá ngoại	Người tạo khoá học
3	name	String		Tên khoá học

4	class_name	String		Mã lớp
5	slug	String		Đường dẫn tùy chỉnh của khoá học
6	desciption	String		Mô tả khoá học
7	join_key	String		Mật khẩu tham gia khoá học
8	thumbnail	String		Ảnh thu nhỏ của khoá học
9	public	BOOLEAN		Trạng thái công khai/riêng tư của khoá học
10	publish	BOOLEAN		Trạng thái công bố của khoá học
11	auto_join	BOOLEAN		Trạng thái tự động duyệt thành viên vào khoá học
12	units	JSON		Danh sách các chương của khoá học

* Mô tả bảng USERS_COURSES (Tham gia khoá học):

Bảng 3.6. Bảng mô tả bảng Users_Courses

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã tham gia khoá học
2	course_id	String	Khoá ngoại	Mã khoá học
3	email	String		Email của người dùng

4	status	ENUM('ACTIVE', 'INACTIVE', 'BANNED')		Trạng thái tham gia khoá học
---	--------	--------------------------------------	--	------------------------------

* Mô tả bảng UNITS (Chương):

Bảng 3.7. Bảng mô tả bảng Units

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã chương
2	course_id	String	Khoá ngoại	Mã khoá học
3	name	String		Tên chương
4	children	JSON		Danh sách mã các bài tập có trong chương

* Mô tả bảng PROBLEMS (Bài tập):

Bảng 3.8. Bảng mô tả bảng Problems

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã bài tập
2	created_by	String	Khoá ngoại	Mã người dùng của người tạo bài tập
3	name	String		Tên bài tập
4	description	String		Mô tả bài tập
5	slug	String		Đường dẫn tùy chỉnh của bài tập

6	tags	JSON		Danh sách tag của bài tập
7	language	ENUM('C', 'CPP', 'JAVA')		Ngôn ngữ lập trình của bài tập
8	input	String		Mô tả dữ liệu đầu vào
9	output	String		Mô tả dữ liệu đầu ra
10	limit	String		Mô tả giới hạn
11	examples	JSON		Danh sách các ví dụ
12	testcases	JSON		Danh sách các testcases
13	type	ENUM('COURSE', 'CONTEST', 'FREE')		Loại bài tập: Bài tập trong khoa học, trong cuộc thi hoặc bài tập tự do
14	level	ENUM('EASY', 'MEDIUM', 'HARD')		Mức độ của bài tập: dễ, trung bình, khó
15	score	Integer		Điểm số

* Mô tả bảng EXAMPLES (Ví dụ):

Bảng 3.9. Bảng mô tả bảng Examples

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã ví dụ
2	input	String		Dữ liệu đầu vào

3	output	String		Dữ liệu đầu ra
4	note	String		Ghi chú

* Mô tả bảng TESTCASES (Ví dụ):

Bảng 3.10. Bảng mô tả bảng Testcases

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã testcase
2	input	String		Dữ liệu đầu vào
3	output	String		Dữ liệu đầu ra
4	suggestion	String		Gợi ý hoàn thành testcase

* Mô tả bảng SUBMISSIONS (Bài nộp):

Bảng 3.11. Bảng mô tả bảng Submissions

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã bài nộp
2	problem_slug	String		Đường dẫn của bài tập
3	username	String		Tên người dùng nộp bài
4	sha	String		Mã thay đổi trên GitHub
5	commit	String		Thông điệp của người dùng

6	run_id	String		Mã lần chạy GitHub Actions trên GitHub
7	code	String		Mã nguồn được mã hoá ở dạng Base64
8	public	BOOLEAN		Công khai/Ẩn mã nguồn với người khác
9	duration	Integer		Thời gian chấm bài nộp
10	result	JSON		Mảng kết quả chấm với từng testcase
11	review	JSON		Đánh giá mã nguồn từ Gemini
12	status	ENUM('PENDING', 'PASSED', 'FAILED', 'ERROR', 'COMPILE_ERROR')		Mã trạng thái của lần nộp bài
13	pass_count	Integer		Số testcase đạt
14	total_count	Integer		Tổng số testcase

* Mô tả bảng DISCUSSIONS (Thảo luận):

Bảng 3.12. Bảng mô tả bảng Discussions

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã cuộc thảo luận
2	username	String		Tên người dùng
3	problem_slug	String		Đường dẫn đến bài tập
4	title	String		Tiêu đề cuộc thảo luận
5	description	String		Mô tả của cuộc thảo luận
6	status	ENUM('OPEN', 'CLOSED')		Trạng thái Đóng/mở của cuộc thảo luận

* Mô tả bảng COMMENTS (Bình luận):

Bảng 3.13. Bảng mô tả bảng Comments

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã bình luận
2	discussion_id	String	Khoá ngoại	Mã cuộc thảo luận
3	username	String		Tên người dùng
4	description	String		Nội dung bình luận
5	liked_by	JSON		Danh sách lượt thích của bình luận

* Mô tả bảng CONTESTS (Cuộc thi):

Bảng 3.14. Bảng mô tả bảng Contests

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Điễn giải
1	id	String	Khoá chính	Mã cuộc thi
2	created_by	String	Khoá ngoại	Người tạo cuộc thi
3	name	String		Tên cuộc thi
4	slug	String		Đường dẫn tùy chỉnh của cuộc thi
5	desciption	String		Mô tả cuộc thi
6	join_key	String		Mật khẩu tham gia cuộc thi
7	public	BOOLEAN		Trạng thái công khai/riêng tư của cuộc thi
8	publish	BOOLEAN		Trạng thái công bố của cuộc thi
9	pinned	BOOLEAN		Trạng thái ghim/bỏ ghim bài viết
10	start_time	Timestamp		Thời gian bắt đầu cuộc thi
11	end_time	Timestamp		Thời gian kết thúc cuộc thi
12	duration	Timestamp		Thời lượng của cuộc thi

13	problems	JSON		Danh sách các bài tập trong cuộc thi
----	----------	------	--	--------------------------------------

* Mô tả bảng USERS_CONTESTS (Tham gia cuộc thi):

Bảng 3.15. Bảng mô tả bảng Users_Contests

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	id	String	Khoá chính	Mã tham gia cuộc thi
2	course_id	String	Khoá ngoại	Mã cuộc thi
3	user_id	String	Khoá ngoại	Mã người dùng
4	status	ENUM('ACTIVE', 'EXITED', 'BANNED')		Trạng thái tham gia cuộc thi: hoạt động, đã rời khỏi, bị cấm

3.4. Thiết kế workflow cho GitHub Actions

Để sử dụng tính năng GitHub Actions, cần đặt trạng thái của Repository thành “Công khai” và file workflow (viết bằng ngôn ngữ yaml) được đặt trong đường dẫn như sau:

```
.github/workflows/ci.yml
```

Tên workflow được đặt là CI, nhằm chỉ quá trình kiểm thử và đánh giá mã nguồn.

```
name: CI
```

Với yêu cầu của đề bài, chúng ta cần kích hoạt GitHub Actions mỗi khi sinh viên nộp bài (through qua Git/Web UI), do đó, cần dựa vào sự kiện Push để chạy workflow trên chính nhánh đã kích hoạt sự kiện Push.

```
on:  
  push:  
    branches:  
      - '**'
```

GitHub Actions cung cấp các môi trường chạy Ubuntu, Windows, macOS. Để tối ưu về tốc độ và sự ổn định, chúng ta chọn môi trường Ubuntu để thực hiện.

```
jobs:  
  build:  
    runs-on: ubuntu-latest  
  
    steps:  
      - name: Record Start Time  
      - name: Checkout Code  
      - name: Send Pending Status  
      - name: Set up Compiler and Libraries  
      - name: Compile Code  
      - name: Handle Compile Failure  
      - name: Get Test Cases  
      - name: Run Tests  
      - name: Record End Time  
      - name: Post Test Results
```

Để thực hiện kiểm thử và đánh giá, chúng ta cần thực hiện lần lượt các bước (step) với các chức năng như sau:

1. Record Start Time: Ghi lại thời gian bắt đầu thực hiện workflow, nhằm xác định thời gian chạy của tiến trình.

```
- name: Record start time  
  id: start_time
```

```
run: echo "START_TIME=$(date +%s)" >> $GITHUB_ENV
```

2. Checkout Code: Chuẩn bị mã nguồn cho các bước sau.

```
- name: Checkout code  
  uses: actions/checkout@v2
```

3. Send Pending Status: Gửi một yêu cầu POST đến Back-end để thông báo rằng, có một tiến trình đang chạy, đồng thời gửi đi mã nguồn để đánh giá chất lượng.

```
- name: Send Pending Status to API  
  run: |  
    branch_name=$(echo ${github.ref} | sed 's|refs/heads/||')  
  
    api_url="https://proxy.kakanie.com/problems/$branch_name/submit/result"  
    encoded_code=$(base64 -w 0 main.c)
```

```
    json_payload=$(jq -n \  
      --arg branch "$branch_name" \  
      --arg actor "${github.actor}" \  
      --arg job_name "${github.job}" \  
      --arg run_id "${github.run_id}" \  
      --arg sha "${github.sha}" \  
      --arg status "pending" \  
      --arg code "$encoded_code" \  
      '{  
        problem: $branch,  
        actor: $actor,  
        job_name: $job_name,  
        run_id: $run_id,
```

```

        sha: $sha,
        status: $status,
        code: $code
    }')

echo "JSON Payload: $json_payload"

curl -X POST "$api_url" \
-H "Content-Type: application/json" \
-d "$json_payload"

```

4. Set up Compiler and Libraries: Chuẩn bị các trình biên dịch và thư viện cần thiết cho các bước tiếp theo. Các trình biên dịch và thư viện sẽ khác nhau tùy thuộc vào ngôn ngữ lập trình.

Với ngôn ngữ C:

```

- name: Set up C Compiler and Libraries
  run: |
    sudo apt-get update
    sudo apt-get install -y gcc cmake git curl
    # Download and install jq-1.5-1
    curl -LO https://github.com/stedolan/jq/releases/download/jq-
1.5/jq-linux64
    chmod +x jq-linux64
    sudo mv jq-linux64 /usr/local/bin/jq

```

Với ngôn ngữ C++:

```

- name: Set up C++ Compiler and Libraries
  run: |
    sudo apt-get update
    sudo apt-get install -y g++-10 cmake git curl

```

```

        sudo update-alternatives --install /usr/bin/g++ g++
/usr/bin/g++-10 90

        echo "#include <bits/stdc++.h>" > stdc++.h
        echo "using namespace std;" >> stdc++.h
        # Download and install jq-1.5-1
        curl -LO https://github.com/stedolan/jq/releases/download/jq-
1.5/jq-linux64
        chmod +x jq-linux64
        sudo mv jq-linux64 /usr/local/bin/jq

```

Với ngôn ngữ Java:

```

- name: Set up Java and Utilities
  run: |
    sudo apt-get update
    sudo apt-get install -y openjdk-17-jdk git curl
    # Download and install jq-1.5-1
    curl -LO https://github.com/stedolan/jq/releases/download/jq-
1.5/jq-linux64
    chmod +x jq-linux64
    sudo mv jq-linux64 /usr/local/bin/jq

```

5. Compile Code: Cần chỉ định file và bắt đầu biên dịch mã nguồn sang mã máy để máy tính có thể hiểu và thực thi. Bước này cũng khác nhau tùy thuộc vào từng ngôn ngữ lập trình:

Với ngôn ngữ C (*file main.c, trình biên dịch gcc với chuẩn C99*):

```

- name: Compile Code
  id: compile
  run: |
    gcc -o my_program main.c -std=c99

```

Với ngôn ngữ C++ (*file main.cpp, trình biên dịch g++ với chuẩn C++17*):

```

- name: Compile Code

  id: compile

  run: |

    g++ -o my_program main.cpp -std=c++17

```

Với ngôn ngữ Java (*file Main.java, trình biên dịch javac*):

```

- name: Compile Code

  id: compile

  run: javac Main.java

```

6. Handle Compile Failure: Bước này chỉ hoạt động nếu quá trình Compile Code ở bước trên thất bại (do sai cú pháp, gặp lỗi,...). GitHub Actions sẽ gửi một yêu cầu POST đến Back-end để ghi nhận kết quả này (*Mã trạng thái: COMPILE_ERROR*).

```

- name: Handle Compile Failure

  if: failure()

  run: |

    end_time=$(date +%s)

    start_time=${{ env.START_TIME }}

    duration=$((end_time - start_time))

    echo "DURATION=$duration" >> $GITHUB_ENV

    branch_name=$(echo ${{ github.ref }} | sed 's|refs/heads/||')

api_url="https://proxy.kakanie.com/problems/$branch_name/submit/result"
encoded_code=$(base64 -w 0 main.cpp)

json_payload=$(jq -n \
  --arg branch "$branch_name" \
  --arg actor "${{ github.actor }}" \

```

```

    --arg job_name "${{ github.job }}" \
    --arg run_id "${{ github.run_id }}" \
    --arg sha "${{ github.sha }}" \
    --arg status "compile_error" \
    --arg duration "$duration" \
    --arg code "$encoded_code" \
    '{

        problem: $branch,
        actor: $actor,
        job_name: $job_name,
        run_id: $run_id,
        sha: $sha,
        status: $status,
        duration: ($duration | tonumber),
        code: $code
    }')
}

echo "JSON Payload: $json_payload"

curl -X POST "$api_url" \
-H "Content-Type: application/json" \
-d "$json_payload"

```

7. Get Test Cases: Nếu biên dịch chương trình thành công, GitHub Actions sẽ gửi một yêu cầu GET đến Back-end để lấy thông tin Test-cases của bài tập này để chuẩn bị cho quá trình Test.

```

- name: Fetch Test Cases from API

  if: success()

  run: |

```

```

branch_name=$(echo ${github.ref} | sed 's|refs/heads/||')
api_url="https://proxy.kakanie.com/problems/$branch_name/testcases"
curl -s $api_url -o test_cases.json

```

8. Run Tests: Chạy các Testcases với chương trình đã được biên dịch, ghi nhận các kết quả, gắn các Mã trạng thái tương ứng:

- *PASSED*: Test-case được thông qua.
- *FAILED*: Test-case không được thông qua.
- *ERROR*: Có lỗi xảy ra trong quá trình thực hiện Test-case (Quá thời gian, quá bộ nhớ, chạy vô hạn, in ra các giá trị không xác định,...).

```

- name: Run Tests
  if: success()
  id: run_tests
  run: |
    pass_count=0
    total_count=0
    results="[]"
    overall_status="passed"

    jq -c '.[]' test_cases.json > test_cases_list.txt
    while IFS= read -r test_case; do
      id=$(echo "$test_case" | jq -r '.id')
      input=$(echo "$test_case" | jq -r '.input')
      expected_output=$(echo "$test_case" | jq -r '.output')

      echo "Running test with ID: $id"
      echo "Input: $input"
      echo "Expected Output: $expected_output"

```

```

# Run the program and handle any errors

if ! echo "$input" | ./my_program > output.txt; then

    echo "Test $id encountered an error!"

    result_status="error"

    actual_output="N/A"      # Do not record actual output if
there's an error

    overall_status="error"

else

    # Ensure there's a newline at the end of both output files

    sed -i -e '$a\' output.txt

    actual_output=$(cat output.txt)

echo "Actual Output: $actual_output"

if [ "$actual_output" == "$expected_output" ]; then

    echo "Test $id passed!"

    result_status="passed"

    pass_count=$((pass_count + 1))

else

    echo "Test $id failed!"

    result_status="failed"

    overall_status="failed"

fi

fi

echo "-----"

# Append the new result to the JSON array

```

```

        results=$(echo "$results" | jq --arg id "$id" --arg status
"$result_status" --arg actual_output "$actual_output" \
        '. += [{"id": $id, "status": $status, "actual_output": $actual_output}])'

        total_count=$((total_count + 1))

done < test_cases_list.txt

# Log results for debugging
echo "Results JSON: $results"
echo "Overall Status: $overall_status"
echo "Total: $total_count"
echo "Pass Count: $pass_count"

# Encode JSON for safe export to GitHub Actions environment
encoded_results=$(echo "$results" | jq -c .)
echo "RESULTS_JSON=$encoded_results" >> $GITHUB_ENV
echo "OVERALL_STATUS=$overall_status" >> $GITHUB_ENV
echo "PASS_COUNT=$pass_count" >> $GITHUB_ENV
echo "TOTAL_COUNT=$total_count" >> $GITHUB_ENV

```

9. Record End Time: Kết thúc quá trình Run Testcases, ghi nhận thời gian thực hiện.

```

- name: Record end time and calculate duration
  id: end_time
  if: success()
  run: |
    end_time=$(date +%s)
    start_time=${{ env.START_TIME }}
    duration=$((end_time - start_time))

```

```
echo "DURATION=$duration" >> $GITHUB_ENV
```

10. Post Test Results: Chuẩn bị kết quả và gửi kết quả tổng kết về cho Back-end. Kết thúc quá trình chạy workflow.

```
- name: Post Test Results

  if: success()

  run: |

    branch_name=$(echo ${{ github.ref }} | sed 's|refs/heads/|||')

api_url="https://proxy.kakanie.com/problems/$branch_name/submit/result"
"

encoded_code=$(base64 -w 0 main.cpp)

json_payload=$(jq -n \
  --arg branch "$branch_name" \
  --arg actor "${{ github.actor }}" \
  --arg job_name "${{ github.job }}" \
  --arg run_id "${{ github.run_id }}" \
  --arg sha "${{ github.sha }}" \
  --arg status "$OVERALL_STATUS" \
  --arg result "$RESULTS_JSON" \
  --arg pass_count "$PASS_COUNT" \
  --arg total_count "$TOTAL_COUNT" \
  --arg duration "$DURATION" \
  --arg code "$encoded_code" \
  '{

  problem: $branch,
  actor: $actor,
  job_name: $job_name,
```

```

    run_id: $run_id,
    sha: $sha,
    status: $status,
    result: ($result | fromjson),
    pass_count: ($pass_count | tonumber),
    total_count: ($total_count | tonumber),
    duration: ($duration | tonumber),
    code: $code
}

echo "JSON Payload: $json_payload"

curl -X POST "$api_url" \
-H "Content-Type: application/json" \
-d "$json_payload"

```

Ngoài ra, còn có các biến môi trường được sử dụng trong workflow như sau:

```

env:
  RESULTS_JSON: ${{ env.RESULTS_JSON }}
  OVERALL_STATUS: ${{ env.OVERALL_STATUS }}
  PASS_COUNT: ${{ env.PASS_COUNT }}
  TOTAL_COUNT: ${{ env.TOTAL_COUNT }}
  DURATION: ${{ env.DURATION }}

```

3.5. Xây dựng API phía máy chủ (Back-end)

3.6. Các API xác thực (Auth)

Bảng 3.16. Bảng mô tả API lấy thông tin người dùng hiện tại

Phương thức	GET
Endpoint	{API_DOMAIN}/auth/me
Diễn giải	API cho phép lấy ra thông tin của người dùng đang đăng nhập

Bảng 3.17. Bảng mô tả API đăng nhập

Phương thức	POST	
Endpoint	{API_DOMAIN}/auth/login	
Thuộc tính	access_token	Access Token hợp lệ của tài khoản GitHub
	uid	Mã người dùng của tài khoản GitHub
	email	Email của người dùng cần đăng nhập
Diễn giải	API cho phép đăng nhập hệ thống với dữ liệu đăng nhập OAuth từ GitHub. Nếu người dùng chưa tồn tại (đăng nhập lần đầu), hệ thống sẽ tạo mới thông tin người dùng.	

Bảng 3.18. Bảng mô tả API đăng xuất

Phương thức	GET
Endpoint	{API_DOMAIN}/auth/logout
Diễn giải	API cho phép đăng xuất khỏi hệ thống

3.7. Các API chức năng người dùng (Users)

Bảng 3.19. Bảng mô tả API lấy thông tin trang cá nhân

Phương thức	GET	
Endpoint	{API_DOMAIN}/users/profile/{:username}	
Thuộc tính	username	Tên người dùng cần lấy thông tin

Diễn giải	API cho phép lấy thông tin người dùng bằng username
------------------	---

Bảng 3.20. Bảng mô tả API lấy thông tin bảng xếp hạng

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/users/analysis/leaderboard</code>
Diễn giải	API cho phép lấy ra bảng xếp hạng người dùng

3.8. Các API chức năng bài viết (Posts)

Bảng 3.21. Bảng mô tả API lấy danh sách bài viết

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/posts/list</code>
Diễn giải	API cho phép lấy ra danh sách các bài viết

Bảng 3.22. Bảng mô tả API lấy bài viết theo ID

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/posts/{:slug}</code>
Diễn giải	API cho phép lấy ra bài viết cụ thể theo đường dẫn

Bảng 3.23. Bảng mô tả API lấy danh sách bài viết theo tag

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/posts/tag/{:tag}</code>
Diễn giải	API cho phép lấy ra danh sách các bài viết theo tag

Bảng 3.24. Bảng mô tả API tạo bài viết mới

Phương thức	POST	
Endpoint	<code>{API_DOMAIN}/posts/create</code>	
Thuộc tính	title	Tiêu đề bài viết
	description	Mô tả bài viết
	content	Nội dung bài viết

	thumbnail	Ảnh thu nhỏ
	tags	Danh sách các tag
Diễn giải	API cho phép chỉnh sửa một chương	

Bảng 3.25. Bảng mô tả API công bố bài viết

Phương thức	PUT
Endpoint	{API_DOMAIN}/posts/edit.publish/{:id}
Diễn giải	API cho phép chỉnh sửa công bố/ẩn bài viết

Bảng 3.26. Bảng mô tả API duyệt bài viết

Phương thức	PUT
Endpoint	{API_DOMAIN}/admin/posts/edit.status/{:id}
Diễn giải	API cho phép duyệt bài viết

Bảng 3.27. Bảng mô tả API chỉnh sửa bài viết

Phương thức	PUT	
Endpoint	{API_DOMAIN}/admin/posts/edit/{:id}	
Thuộc tính	title	Tiêu đề bài viết
	description	Mô tả bài viết
	content	Nội dung bài viết
	thumbnail	Ảnh thu nhỏ
	tags	Danh sách các tag
Diễn giải	API cho phép chỉnh sửa bài viết	

Bảng 3.28. Bảng mô tả API xoá bài viết

Phương thức	DELETE	
Endpoint	{API_DOMAIN}/admin/posts/destroy	
Thuộc tính	ids	Danh sách các id cần xoá

Diễn giải	API cho phép các bài viết theo danh sách mã bài viết
------------------	--

3.9. Các API chức năng khoá học (Courses)

Bảng 3.29. Bảng mô tả API lấy danh sách các khoá học

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/list
Diễn giải	API cho phép lấy ra danh sách các khoá học trên hệ thống

Bảng 3.30. Bảng mô tả API lấy danh sách khoá học đã tham gia

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/joined
Diễn giải	API cho phép lấy ra danh sách các khoá học đã tham gia

Bảng 3.31. Bảng mô tả API lấy danh sách khoá học đã tạo

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/created
Diễn giải	API cho phép lấy ra danh sách các khoá học đã tạo

Bảng 3.32. Bảng mô tả API lấy thông tin khoá học theo ID (Slug)

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/{:slug}
Diễn giải	API cho phép lấy ra thông tin chi tiết khoá học theo đường dẫn

Bảng 3.33. Bảng mô tả API lấy số liệu thống kê khoá học

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/{:slug}/analysis
Diễn giải	API cho phép lấy ra thống kê tổng quan của khoá học

Bảng 3.34. Bảng mô tả API lấy thông tin khoá học dành cho người quản trị

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/admin/{:id}
Diễn giải	API cho phép lấy ra thông tin chi tiết khoá học (dành cho người quản trị) theo đường dẫn

Bảng 3.35. Bảng mô tả API tạo khoá học mới

Phương thức	POST	
Endpoint	{API_DOMAIN}/courses/create	
Thuộc tính	name	Tên khoá học
	description	Mô tả khoá học
	class_name	Mã lớp
	join_key	Mật khẩu tham gia
	thumbnail	Ảnh bìa của khoá học (Base64)
Diễn giải	API cho phép tạo một khoá học mới	

Bảng 3.36. Bảng mô tả API tham gia khoá học

Phương thức	POST	
Endpoint	{API_DOMAIN}/courses/join/{:id}	
Thuộc tính	join_key	Mật khẩu tham gia
Diễn giải	API cho phép tham gia vào một khoá học	

Bảng 3.37. Bảng mô tả API lấy ra danh sách chương và bài tập trong khoá học

Phương thức	GET
Endpoint	{API_DOMAIN}/courses/{:id}/units/list
Diễn giải	API cho phép lấy ra các chương của một khoá học

Bảng 3.38. Bảng mô tả API tạo chương (phân) mới trong khoá học

Phương thức	POST	
Endpoint	{API_DOMAIN}/courses/{:id}/units/create	
Thuộc tính	name	Tên chương
Diễn giải	API cho phép tạo một chương mới trong khoá học	

Bảng 3.39. Bảng mô tả API chỉnh sửa khoá học

Phương thức	PUT	
Endpoint	{API_DOMAIN}/courses/{:id}/units/{:unit_id}	
Thuộc tính	name	Tên chương (đã sửa)
Diễn giải	API cho phép chỉnh sửa một chương	

Bảng 3.40. Bảng mô tả API xoá khoá học (xoá mềm)

Phương thức	DELETE	
Endpoint	{API_DOMAIN}/courses/{:id}	
Diễn giải	API cho phép xoá khoá học	

3.10. Các API chức năng bài tập (Problems)

Bảng 3.41. Bảng mô tả API lấy danh sách bài tập tự do

Phương thức	GET	
Endpoint	{API_DOMAIN}/problems/list	
Diễn giải	API cho phép lấy ra các bài tập tự do trên hệ thống	

Bảng 3.42. Bảng mô tả API lấy thông tin bài tập theo đường dẫn

Phương thức	GET	
Endpoint	{API_DOMAIN}/problems/:slug	
Diễn giải	API cho phép lấy ra thông tin bài tập	

Bảng 3.43. Bảng mô tả API lấy danh sách testcases của bài tập

Phương thức	GET
Endpoint	{API_DOMAIN}/problems/{:slug}/testcases
Điễn giải	API cho phép lấy ra các testcase của bài tập

Bảng 3.44. Bảng mô tả API lấy các phân tích số liệu của bài tập

Phương thức	GET
Endpoint	{API_DOMAIN}/problems/{:slug}/analysis/submissions
Điễn giải	API cho phép lấy ra các phân tích số liệu của bài tập (dùng để vẽ các biểu đồ)

Bảng 3.45. Bảng mô tả API tạo khoá học mới

Phương thức	POST	
Endpoint	{API_DOMAIN}/problems/create	
Thuộc tính	name	Tên bài tập
	slug	Đường dẫn bài tập
	tags	Mảng các tag của bài tập
	language	Ngôn ngữ lập trình
	description	Mô tả bài tập
	input	Đầu vào
	output	Đầu ra
	limit	Giới hạn
	examples	Mảng các ví dụ
	testcases	Mảng các testcase
	score	Điểm số
	type	Loại bài tập (FREE, CONTEST, COURSE)

Diễn giải	API cho phép tạo một bài tập mới
------------------	----------------------------------

Bảng 3.46. Bảng mô tả API ghi kết quả chấm bài

Phương thức	POST	
Endpoint	{API_DOMAIN}/problems/{:problem_slug} /submissions/result	
Thuộc tính	problem_slug	Đường dẫn của bài tập
	actor	Người nộp bài
	run_id	Mã run trong GitHub Actions
	sha	Mã thay đổi trong GitHub Repo
	status	Trạng thái kết quả của lần chấm
	code	Mã nguồn (Base64)
Diễn giải	API cho phép ghi kết quả chấm bài	

Bảng 3.47. Bảng mô tả API chỉnh sửa bài tập

Phương thức	PUT	
Endpoint	{API_DOMAIN}/problems/{:id}	
Thuộc tính	name	Tên bài tập
	slug	Đường dẫn bài tập
	tags	Mảng các tag của bài tập
	language	Ngôn ngữ lập trình
	description	Mô tả bài tập
	input	Đầu vào
	output	Đầu ra
	limit	Giới hạn
	examples	Mảng các ví dụ
	testcases	Mảng các testcase

	score	Điểm số
Diễn giải	API cho phép chỉnh sửa bài tập	

Bảng 3.48. Bảng mô tả API xoá bài tập (xoá mềm)

Phương thức	DELETE
Endpoint	{API_DOMAIN}/problems/{:id}
Diễn giải	API cho phép xoá bài tập

3.11. Các API chức năng thảo luận (Discussions)

Bảng 3.49. Bảng mô tả API lấy danh sách các cuộc thảo luận theo đường dẫn bài tập

Phương thức	GET
Endpoint	{API_DOMAIN}/discussions/{:problem_slug}
Diễn giải	API cho phép lấy ra các cuộc thảo luận của một bài tập

Bảng 3.50. Bảng mô tả API lấy thông tin cuộc thảo luận theo ID

Phương thức	GET
Endpoint	{API_DOMAIN}/discussions/p/{:id}
Diễn giải	API cho phép lấy ra thông tin cụ thể của một cuộc thảo luận

Bảng 3.51. Bảng mô tả API lấy danh sách bình luận của một cuộc thảo luận

Phương thức	GET
Endpoint	{API_DOMAIN}/discussions/p/{:id}/comments
Diễn giải	API cho phép lấy ra danh sách các bình luận của cuộc thảo luận

Bảng 3.52. Bảng mô tả API tạo cuộc thảo luận mới

Phương thức	POST	
Endpoint	{API_DOMAIN}/discussions/{:slug}	
Thuộc tính	title	Tiêu đề bài viết

	description	Mô tả bài viết
Diễn giải	API cho phép tạo cuộc thảo luận mới	

Bảng 3.53. Bảng mô tả API tạo bình luận mới

Phương thức	POST	
Endpoint	{API_DOMAIN}/discussions/{:slug}/comments	
Thuộc tính	description	Mô tả bài viết
Diễn giải	API cho phép tạo bình luận mới	

Bảng 3.54. Bảng mô tả API đóng/mở cuộc thảo luận

Phương thức	PUT	
Endpoint	{API_DOMAIN}/discussions/{:slug}/status	
Diễn giải	API cho phép đóng/mở cuộc thảo luận	

Bảng 3.55. Bảng mô tả API chỉnh sửa thông tin cuộc thảo luận

Phương thức	PUT	
Endpoint	{API_DOMAIN}/discussions/{:slug}	
Thuộc tính	title	Tiêu đề bài viết
	description	Mô tả bài viết
Diễn giải	API cho phép chỉnh sửa thông tin cuộc thảo luận	

Bảng 3.56. Bảng mô tả API xoá cuộc thảo luận (xoá mềm)

Phương thức	DELETE	
Endpoint	{API_DOMAIN}/discussions/{:problem_slug}	
Diễn giải	API cho phép xoá bài tập	

3.12. Các API chức năng nộp bài (Submissions)

Bảng 3.57. Bảng mô tả API lấy danh sách nộp bài của người dùng

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/submissions/me</code>
Điễn giải	API cho phép lấy ra các lượt nộp bài của người dùng hiện tại

Bảng 3.58. Bảng mô tả API lấy danh sách nộp bài của bài tập

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/submissions/problem/:slug</code>
Điễn giải	API cho phép lấy ra lượt nộp bài của một bài tập cụ thể

Bảng 3.59. Bảng mô tả API lấy thông tin nộp bài theo ID

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/submissions/:id</code>
Điễn giải	API cho phép lấy ra thông tin một lượt nộp bài cụ thể

Bảng 3.60. Bảng mô tả API công khai/ẩn mã nguồn

Phương thức	PUT
Endpoint	<code>{API_DOMAIN}/submissions/:id/public</code>
Điễn giải	API cho phép công khai/ẩn mã nguồn

3.13. Các API chức năng cuộc thi (Contests)

Bảng 3.61. Bảng mô tả API lấy danh sách các cuộc thi

Phương thức	GET
Endpoint	<code>{API_DOMAIN}/contests/list</code>
Điễn giải	API cho phép lấy ra danh sách các cuộc thi trên hệ thống

Bảng 3.62. Bảng mô tả API lấy thông tin cuộc thi theo ID

Phương thức	GET
Endpoint	{API_DOMAIN}/contests/{:id}
Diễn giải	API cho phép lấy ra thông tin một cuộc thi theo ID

Bảng 3.63. Bảng mô tả API lấy danh sách nộp bài của cuộc thi

Phương thức	GET
Endpoint	{API_DOMAIN}/contests/{:id}/submissions
Diễn giải	API cho phép lấy ra các lượt nộp bài của cuộc thi theo ID

Bảng 3.64. Bảng mô tả API lấy danh sách Bảng xếp hạng

Phương thức	GET
Endpoint	{API_DOMAIN}/contests/{:id}/members
Diễn giải	API cho phép lấy ra danh sách những người tham gia và thành tích của họ trong cuộc thi

Bảng 3.65. Bảng mô tả API lấy đề bài của cuộc thi

Phương thức	GET
Endpoint	{API_DOMAIN}/contests/{:id}/problems
Diễn giải	API cho phép lấy ra đề bài của cuộc thi

Bảng 3.66. Bảng mô tả API yêu cầu tham gia cuộc thi

Phương thức	POST	
Endpoint	{API_DOMAIN}/contests/{:id}/join	
Thuộc tính	join_key	Mật khẩu tham gia
Diễn giải	API cho phép người dùng tham gia cuộc thi bằng mật khẩu.	

Bảng 3.67. Bảng mô tả API rời khỏi cuộc thi

Phương thức	POST
Endpoint	<code>{API_DOMAIN}/contests/{:id}/exit</code>
Diễn giải	API cho phép người tham gia rời khỏi cuộc thi.

Bảng 3.68. Bảng mô tả API tạo cuộc thi mới

Phương thức	POST	
Endpoint	<code>{API_DOMAIN}/admin/contests/create</code>	
Thuộc tính	name	Tên cuộc thi
	description	Mô tả cuộc thi
	start_time	Thời gian bắt đầu (timestamp)
	duration	Thời gian làm bài (timestamp)
	join_key	Mật khẩu tham gia cuộc thi (nếu có)
	slug	Đường dẫn cuộc thi (nếu có)
Diễn giải	API cho phép tạo cuộc thi mới	

Bảng 3.69. Bảng mô tả API ghim/bỏ ghim cuộc thi

Phương thức	PUT
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/pinned</code>
Diễn giải	API cho phép ghim/bỏ ghim cuộc thi

Bảng 3.70. Bảng mô tả API công bố/ẩn cuộc thi

Phương thức	PUT
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/publish</code>
Diễn giải	API cho phép công bố/ẩn cuộc thi

Bảng 3.71. Bảng mô tả API thay đổi trạng thái cuộc thi

Phương thức	PUT
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/public</code>

Điễn giải	API cho phép thay đổi cuộc thi giữa công khai/riêng tư
------------------	--

Bảng 3.72. Bảng mô tả API chỉnh sửa cuộc thi

Phương thức	PUT
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/update</code>
Điễn giải	API cho phép chỉnh sửa thông tin cuộc thi

Bảng 3.73. Bảng mô tả API xoá bài tập trong cuộc thi (xoá mềm)

Phương thức	DELETE
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/problems/{:problem_id}</code>
Điễn giải	API cho phép xoá một bài tập trong cuộc thi theo ID bài tập

Bảng 3.74. Bảng mô tả API xoá cuộc thi (xoá mềm)

Phương thức	DELETE
Endpoint	<code>{API_DOMAIN}/admin/contests/{:id}/delete</code>
Điễn giải	API cho phép xoá cuộc thi theo ID

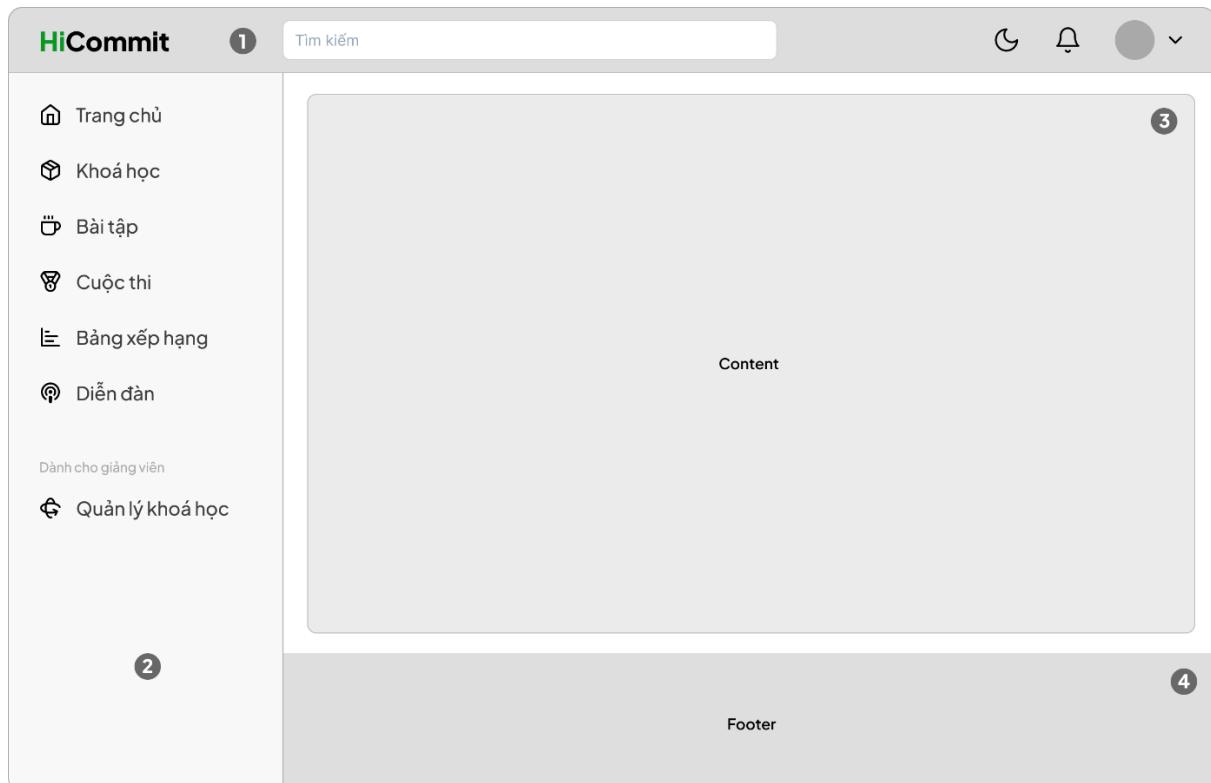
3.14. Thiết kế giao diện và chức năng phía sinh viên, giảng viên (Front-end)

3.14.1. Thiết kế khung giao diện làm việc chính

Khung giao diện chính được thiết kế với bốn thành phần chính:

- *Header* (1): Cho phép người dùng dễ dàng truy cập các tính năng toàn cục như tìm kiếm, chuyển đổi giao diện, xem thông báo và quản lý hồ sơ cá nhân.
- *Navbar* (2): Chứa danh sách các liên kết điều hướng (menu) đến các trang quan trọng như: Trang chủ, Khóa học, Bài tập, Cuộc thi, Bảng xếp hạng, Diễn đàn,... Giúp người dùng dễ dàng truy cập các phần khác nhau của nền tảng.
- *Content* (3): Vùng hiển thị nội dung chính của trang web, Hiển thị thông tin chính theo từng trang mà người dùng đã chọn từ Navbar, ví dụ: danh sách bài tập, thông tin khóa học, cuộc thi,...
- *Footer* (4): Phần chân trang, có thể chứa thông tin liên hệ, bản quyền hoặc các liên

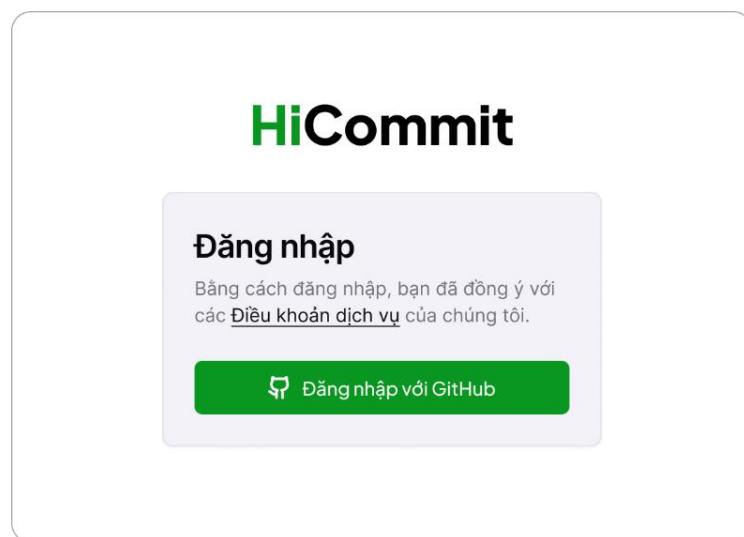
kết đến chính sách và điều khoản sử dụng.



Hình 3.3. Bản phác thảo khung giao diện chính

3.14.2. Chức năng Đăng nhập

Do hệ thống sử dụng nền tảng GitHub để xử lý một phần tiến trình nên người dùng bắt buộc phải đăng nhập bằng tài khoản GitHub để sử dụng được các chức năng trên HiCommit.



Hình 3.4. Bản phác thảo giao diện Đăng nhập

Hệ thống sẽ yêu cầu người dùng cấp quyền để sử dụng các chức năng sau của GitHub:

- GitHub Repositories

- GitHub Actions

- GitHub Profile

Người dùng cần phê duyệt để HiCommit sử dụng được các tính năng này.

3.14.3. Chức năng Bài tập

Các bài tập trên HiCommit được chia làm 3 loại:

- *Bài tập tự do*: Là loại bài tập ôn luyện chung cho toàn bộ người dùng trên hệ thống, được chia làm 3 cấp độ: dễ, trung bình, khó. Mỗi mức độ có mức điểm khác nhau. Khi hoàn thành, điểm số sẽ được ghi nhận và hiển thị trên bảng xếp hạng. Với loại bài tập này, người dùng nào cũng có thể truy cập và sử dụng.

- *Bài tập thuộc khoá học*: Là loại bài tập nằm trong một khoá học cụ thể, chỉ những người tham gia khoá học mới có thể truy cập và sử dụng.

- *Bài tập thuộc cuộc thi*: Là loại bài tập nằm trong một cuộc thi, chỉ những người tham gia cuộc thi mới có thể truy cập và sử dụng. Tính năng thảo luận và đề xuất gợi ý trên loại bài tập này bị vô hiệu hoá để đảm bảo tính công bằng cho những người tham gia cuộc thi.

Giao diện Bài tập hiển thị danh sách các bài tập tự do trên hệ thống, với các thông tin kèm theo như mức độ, điểm, tỉ lệ hoàn thành, dạng bài tập,... Các chức năng được hỗ trợ như tìm kiếm, lọc, tìm theo dạng bài,...

Hình 3.5. Bản phác thảo giao diện Bài tập

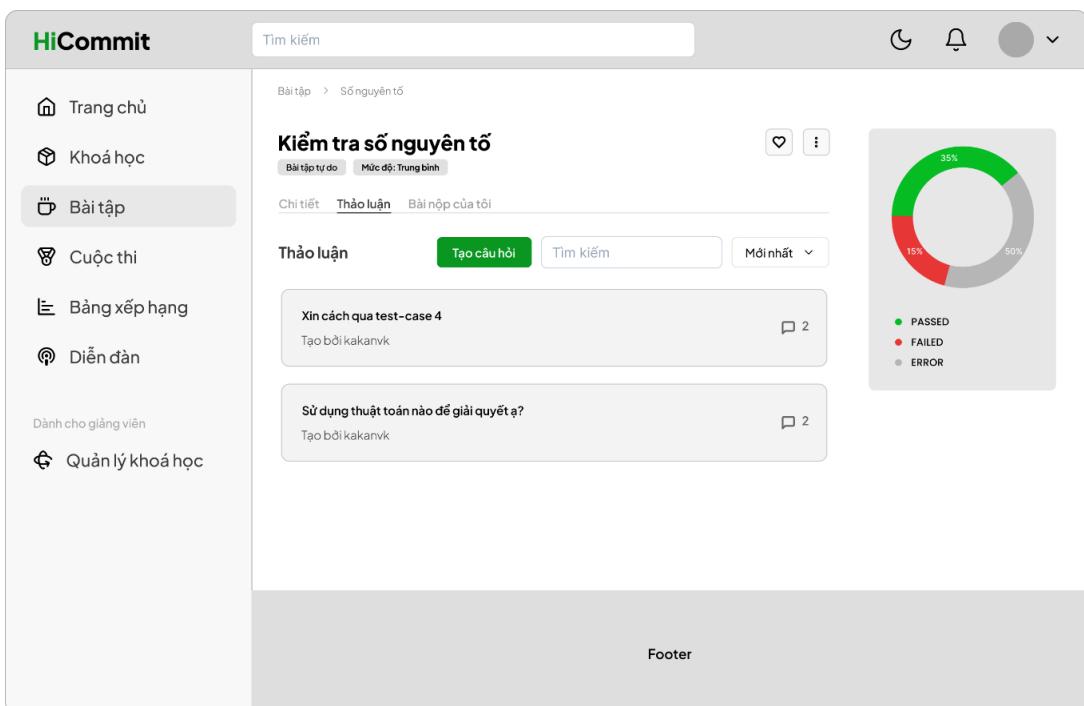
Ở từng bài tập cụ thể, giao diện hiển thị các thông tin như mô tả bài toán, input, output,... Người dùng có thể bấm nộp bài để đưa mã nguồn của mình lên.

Hình 3.6. Bản phác thảo giao diện Chi tiết bài tập

3.14.4. Chức năng Thảo luận

Ở mỗi bài tập, người dùng có thể chuyển sang tab thảo luận để tham gia vào các cuộc thảo luận. Chức năng này cung cấp một không gian để mọi người có thể trao đổi hay đưa

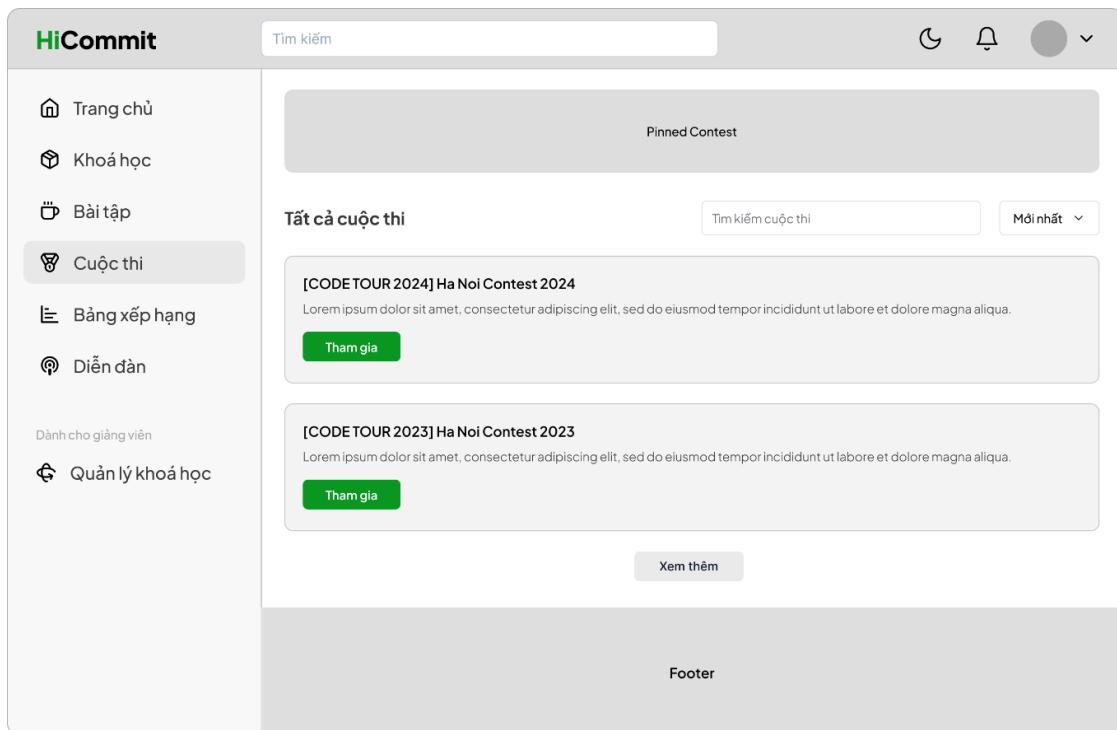
ra các bình luận, chia sẻ, trao đổi về bài toán. Người dùng có thể tạo cuộc thảo luận mới và miêu tả các vấn đề của mình gấp phải. Giao diện này cũng cung cấp một số tính năng cần thiết như tìm kiếm, lọc để dễ dàng tìm ra các thảo luận một cách nhanh chóng.



Hình 3.7. Bản phác thảo giao diện Thảo luận

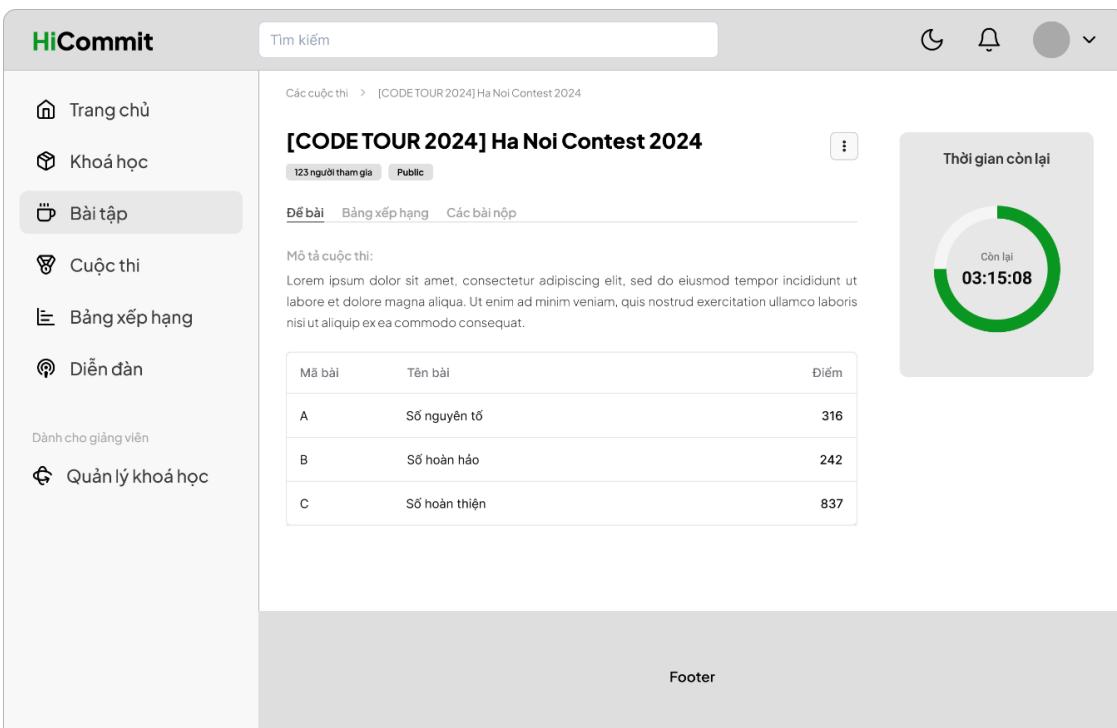
3.14.5. Chức năng Cuộc thi

Chức năng Cuộc thi là chức năng được phát triển dựa trên cách hoạt động của các cuộc thi lớn trong cộng đồng Tin học như Olympic Tin học, ICPC. Chức năng này cho phép người dùng có thể tham gia vào một cuộc thi và giải các bài toán trong cuộc thi đó. Một số cuộc thi chỉ có thể tham gia khi có mật khẩu riêng, điều này đảm bảo sự riêng tư cho các cuộc thi yêu cầu tính bảo mật.



Hình 3.8. Bản phác thảo giao diện Danh sách cuộc thi

Khi tham gia cuộc thi, người tham gia cần có gắng giải được càng nhiều bài toán càng tốt. Mỗi cuộc thi sẽ có thời gian làm bài khác nhau, người tham gia có thể dễ dàng quan sát trên giao diện của cuộc thi. Khi hết giờ làm bài, đề bài sẽ được ẩn đi và hệ thống chấm bài sẽ được đóng lại.



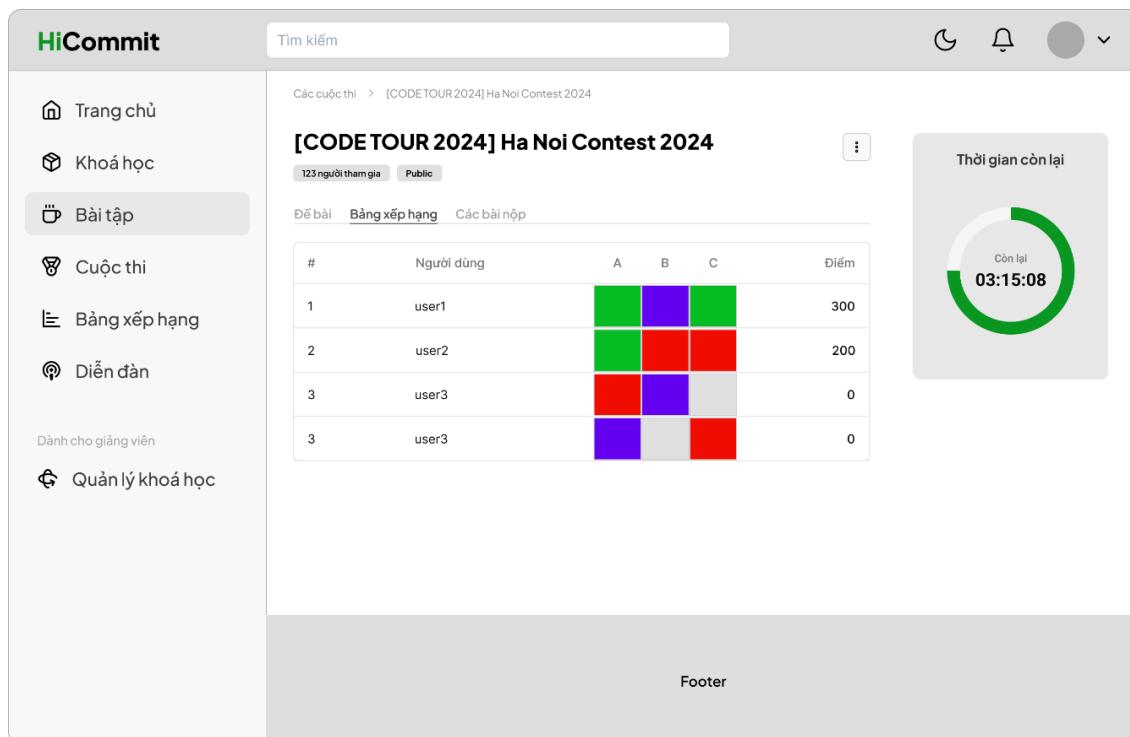
Hình 3.9. Bản phác thảo giao diện Chi tiết cuộc thi

Người tham gia và những người theo dõi bên ngoài cuộc thi có thể dễ dàng theo dõi thứ hạng, kết quả thi đấu của toàn bộ người tham gia thông qua giao diện Bảng xếp hạng. Bảng này được cập nhật liên tục với các trạng thái màu sắc như sau:

Bảng 3.75. Các trạng thái trong Bảng xếp hạng cuộc thi

Màu sắc	Mã trạng thái	Điễn giải
■ Màu xanh lá	PASSED	Bài nộp có kết quả chính xác
■ Màu đỏ	FAILED	Bài nộp có kết quả không chính xác
■ Màu tím	PENDING	Bài nộp đang được xử lý
■ Màu trắng/xám	NONE	Chưa nộp bài

Ngoài ra, khi số lượt sai càng nhiều, thứ hạng cũng sẽ bị ảnh hưởng.

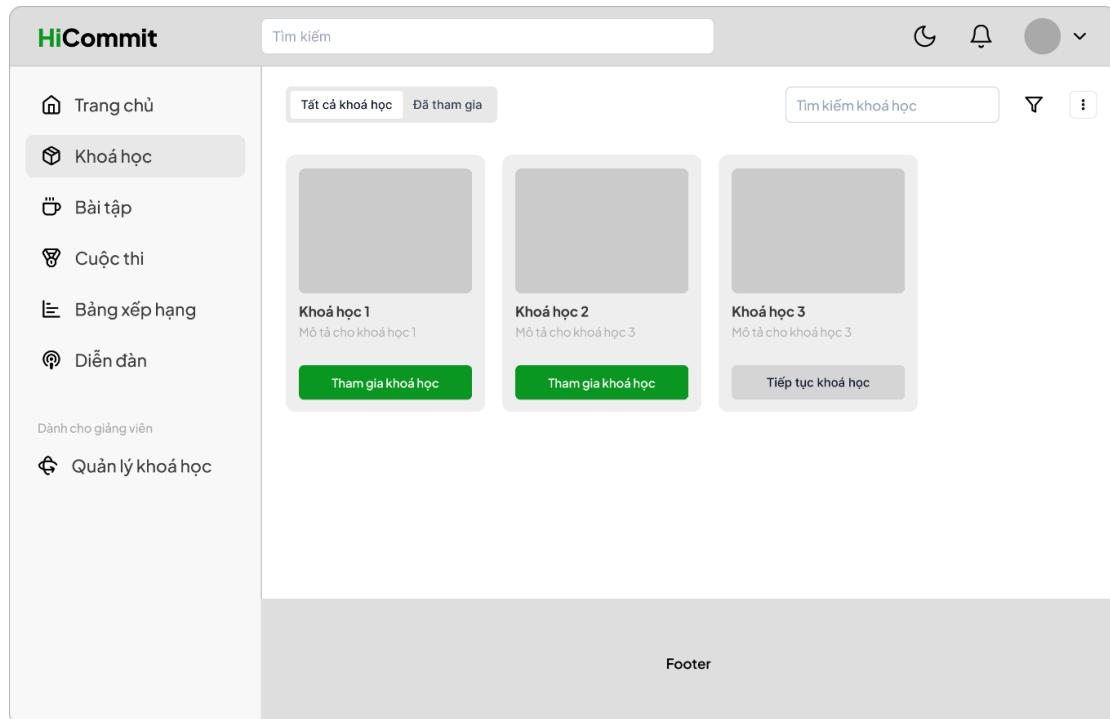


Hình 3.10. Bản phác thảo giao diện Bảng xếp hạng cuộc thi

3.14.6. Chức năng Khoá học

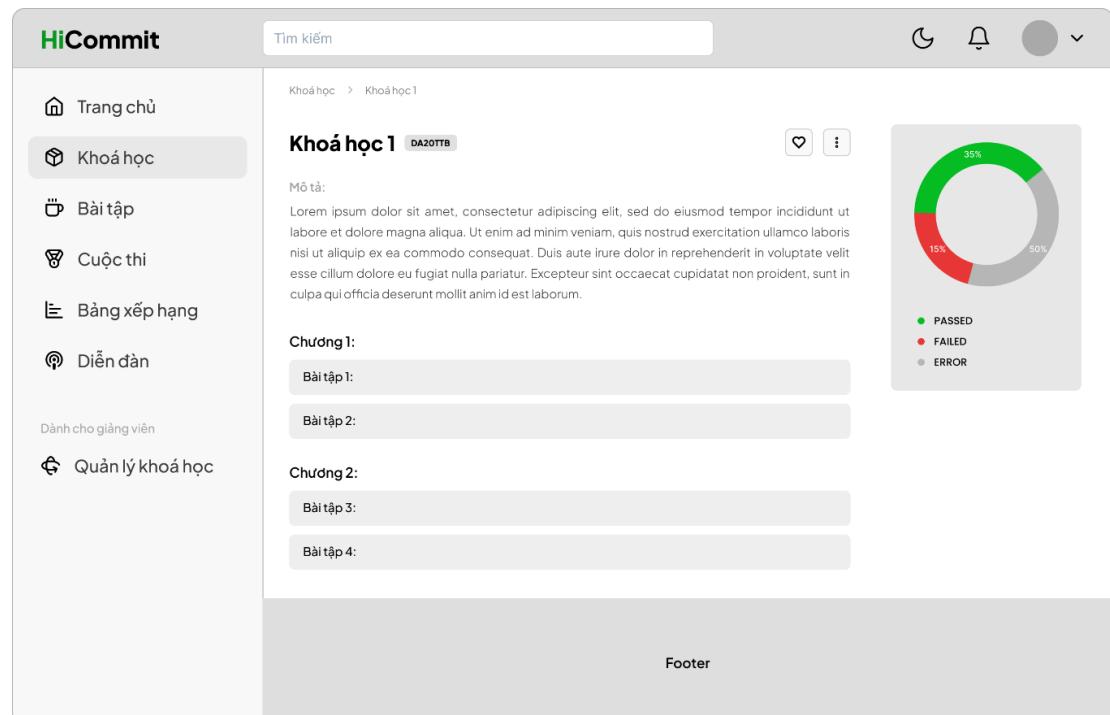
Chức năng này cho phép người dùng, sinh viên có thể tham gia vào các khoá học và hoàn thành những khoá học này. Giao diện này hiển thị các khoá học khả dụng trên hệ thống. Các khoá học được chia làm hai loại là công khai và riêng tư. Các khoá học riêng

tư yêu cầu người tham gia cung cấp mật khẩu tham gia khoá học để xác minh. Ngược lại, các khoá học công khai cho phép người dùng tham gia mà không cần xác thực.



Hình 3.11. Bản phác thảo giao diện Khoa học

Sau khi tham gia, học viên có thể theo dõi tiến độ khoá học, giải các bài tập được đưa ra trong khoá học này để đảm bảo tiến độ học tập.



Hình 3.12. Bản phác thảo giao diện Chi tiết khoa học

3.14.7. Chức năng Bảng xếp hạng

Khác với Bảng xếp hạng của cuộc thi, chức năng này thống kê thứ hạng của toàn bộ người dùng trên hệ thống với các thông tin như tên người dùng, số bài đã giải, tỉ lệ giải đúng bài tập, tổng điểm,... Tại đây cũng hỗ trợ các chức năng như tìm kiếm, sắp xếp.

The screenshot shows the HiCommit application interface. On the left is a sidebar with icons for Trang chủ (Home), Khoá học (Course), Bài tập (Assignment), Cuộc thi (Contest), Bảng xếp hạng (Ranking Table), Diễn đàn (Forum), and Quản lý khoá học (Course Management). The 'Cuộc thi' icon is highlighted with a grey box. The main area is titled 'Bảng xếp hạng' (Ranking Table) and displays a table of user statistics. The table has columns for # (Rank), Tên người dùng (User Name), Số bài đã giải (Solved Problems), Tỉ lệ AC (AC Ratio), and Điểm (Score). The data is as follows:

#	Tên người dùng	Số bài đã giải	Tỉ lệ AC	Điểm
1	kakanvk	15	55%	540
2	user1	10	76%	300
3	user2	5	45%	140
4	user3	1	100%	40

Below the table are search and sort filters: 'Tim kiem người dùng' (Search user) and 'Mới nhất' (Newest). At the bottom are 'Previous' and 'Next' buttons. A footer bar at the bottom of the main content area contains the word 'Footer'.

Hình 3.13. Bản phác thảo giao diện Bảng xếp hạng

3.14.8. Chức năng Diễn đàn

Khu vực này cho phép người dùng có thể đăng các bài viết về các chủ đề khác nhau. Các bài viết được đăng cần phải qua bước kiểm duyệt trước khi hiển thị lên diễn đàn, nhằm quản lý chặt chẽ về mặt nội dung.

The screenshot shows the HiCommit platform interface. On the left sidebar, there are several menu items: Trang chủ, Khoa học, Bài tập, Cuộc thi, Bảng xếp hạng, and Diễn đàn, which is highlighted with a grey background. Other items include Dành cho giảng viên and Quản lý khóa học. The main content area is titled "Các bài viết" (List of posts). It displays three sample posts, each with a title, a short preview of the content, and three associated tags (Tag1, Tag2, Tag3). The first post's preview reads: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua." The second and third posts have similar previews. At the top right of the main area are buttons for "Tạo bài viết" (Create post), "Tim kiếm" (Search), and "Mới nhất" (Newest). Below the main content is a "Footer" section.

Hình 3.14. Bản phác thảo giao diện Diễn đàn

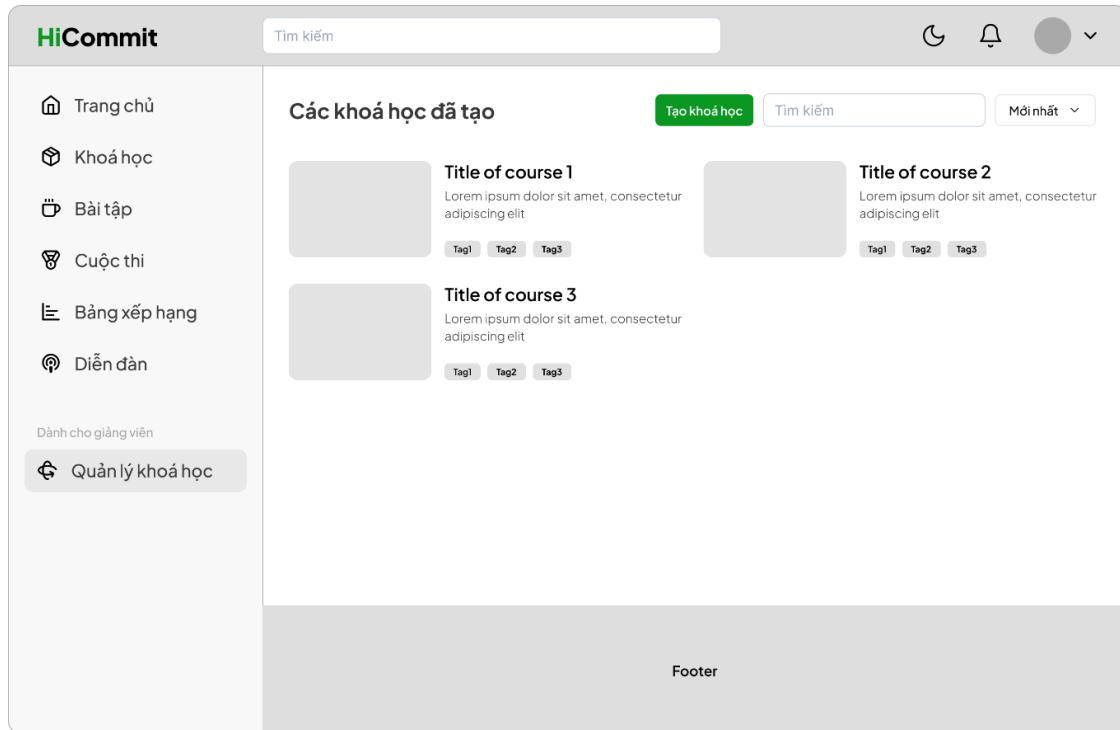
Các bài viết đã được duyệt sẽ xuất hiện trên diễn đàn. Người dùng có thể xem các bài viết để bổ sung kiến thức hoặc hiểu rõ hơn về vấn đề mà tác giả chia sẻ trong bài viết.

This screenshot shows a detailed view of a post on the HiCommit platform. The left sidebar is identical to the one in the previous screenshot, with "Diễn đàn" selected. The main content area shows a single post titled "[CODETOUR 2024] Ha Noi Contest 2024". Below the title, it says "Đăng bởi kakanvk, 2 ngày trước". Underneath the post, there are three tags: Tag1, Tag2, and Tag3. The post content itself is a block of placeholder text: "What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum." A section titled "Why do we use it?" follows, explaining that it's used because it has a more-or-less normal distribution of letters, making it look like readable English. The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham. The bottom of the screen features a "Footer" section.

Hình 3.15. Bản phác thảo giao diện Xem bài viết

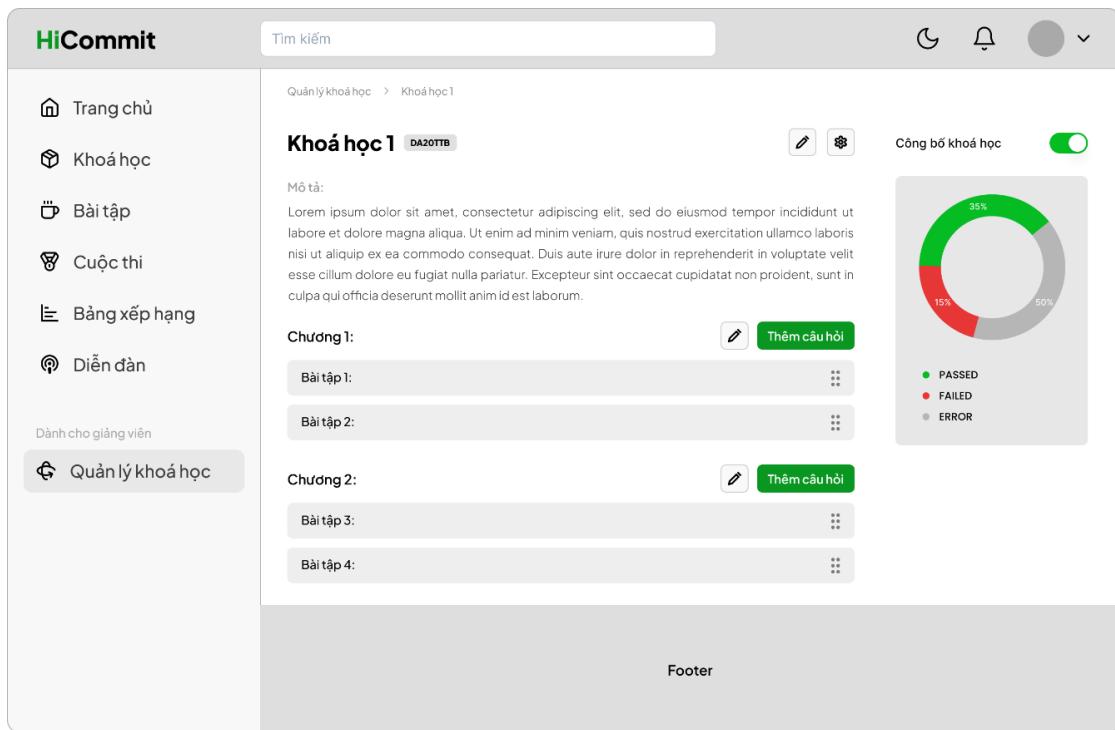
3.14.9. Chức năng Quản lý khoá học (dành cho giảng viên)

Chức năng này giúp giảng viên dễ dàng quản lý được các khoá học mà họ đã tạo. Giảng viên có thể tạo khoá học mới, tìm kiếm, sắp xếp ở giao diện này.



Hình 3.16. Bản phác thảo giao diện Quản lý khoá học

Trong từng khoá học cụ thể, giảng viên có thể chỉnh sửa chi tiết các thông tin, quản lý các bài tập trong khoá học đó hoặc điều chỉnh các thông tin nâng cao hơn. Ngoài ra, giao diện này còn cung cấp một số biểu đồ để giảng viên dễ dàng đánh giá tổng quan khoá học.

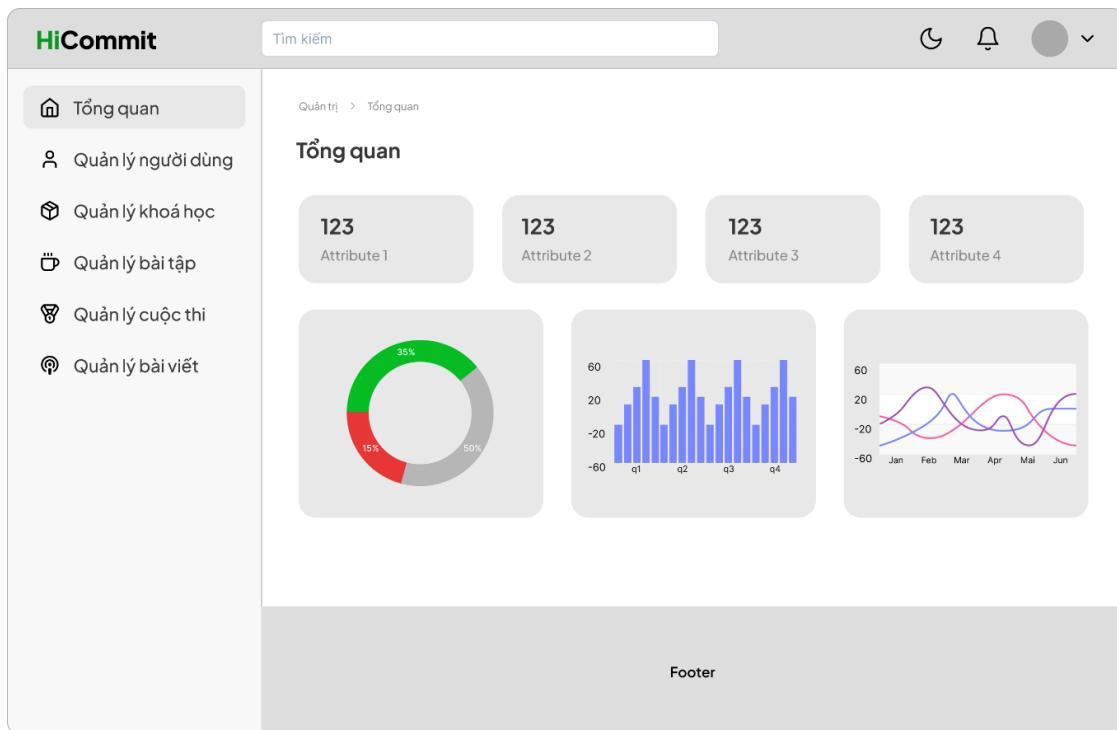


Hình 3.17. Bản phác thảo giao diện Quản lý chi tiết khóa học

3.15. Thiết kế giao diện và chức năng phía quản trị viên (Front-end)

3.15.1. Chức năng Tổng quan

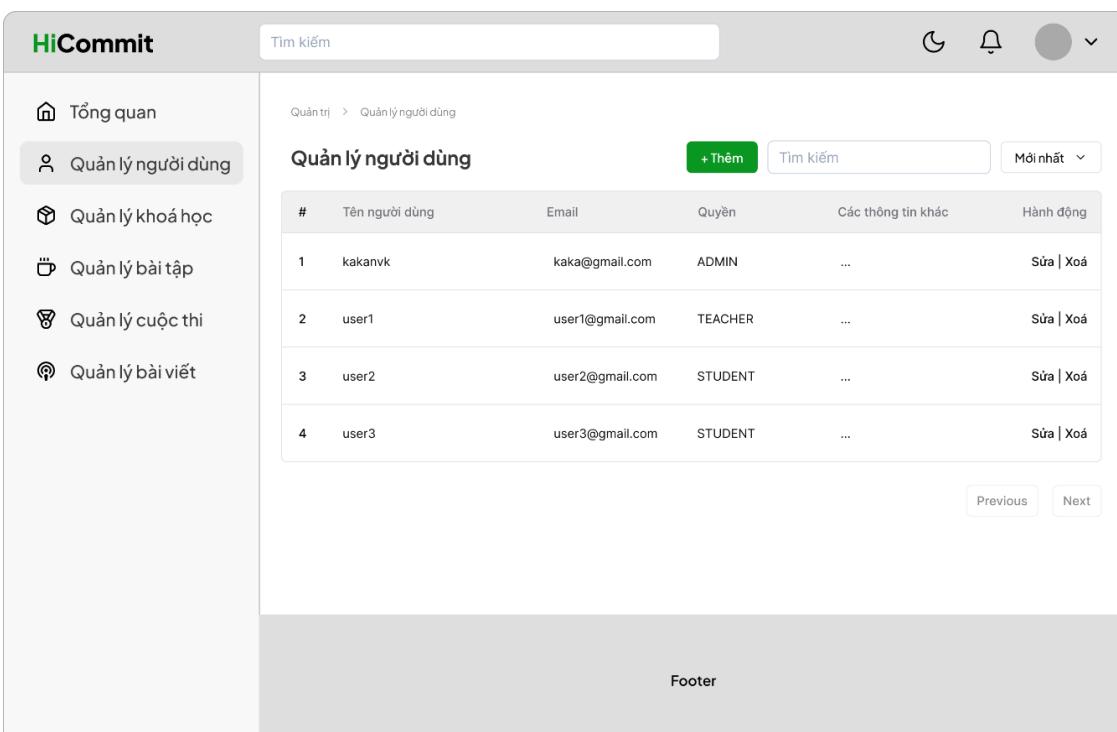
Giao diện này cho phép người quản trị nắm được một số thông tin cơ bản của hệ thống như số lượt nộp bài trong ngày, tổng số người dùng, tổng số bài viết. Một số biểu đồ cũng được thể hiện, biểu diễn trực quan các thông tin như số liệu các lượt nộp bài hoàn thành, không hoàn thành, lỗi, lỗi biên dịch,...



Hình 3.18. Bản phác thảo giao diện Admin – Tổng quan

3.15.2. Chức năng Quản lý người dùng

Ở giao diện này, người quản trị có thể thực hiện các thao tác với danh sách người dùng như thêm người dùng, tìm kiếm, chỉnh sửa thông tin người dùng. Ngoài ra, người quản trị cũng có thể phân quyền cho các người dùng khác bao gồm ba quyền: Quản trị viên, giáo viên, sinh viên.



Hình 3.19. Bản phác thảo giao diện Admin – Quản lý người dùng

3.15.3. Chức năng Quản lý khoá học

Chức năng này cho phép quản trị viên quản lý các khoá học trên hệ thống. Các chức năng được hỗ trợ bao gồm: thêm khoá học, chỉnh sửa, xoá, công bố,... Ngoài ra, có thể xem các phân tích số liệu của từng khoá học.

The screenshot shows the HiCommit Admin dashboard. On the left sidebar, under the 'Quản lý người dùng' section, the 'Quản lý khoá học' option is selected, indicated by a highlighted box around it. The main content area is titled 'Quản lý khoá học'. It displays a table with four rows of course data:

#	Tên khoá học	Tác giả	Tham gia	Các thông tin khác	Hành động
1	Kỹ thuật lập trình	kaka@gmail.com	10	...	Sửa Xoá
2	Lập trình hướng đối tượng	user1@gmail.com	20	...	Sửa Xoá
3	Khoa học 3	user2@gmail.com	30	...	Sửa Xoá
4	Khoa học 4	user3@gmail.com	40	...	Sửa Xoá

At the bottom right of the main content area, there are 'Previous' and 'Next' navigation buttons. The footer of the page is labeled 'Footer'.

Hình 3.20. Bản phác thảo giao diện Admin – Quản lý khoá học

3.15.4. Chức năng Quản lý bài tập

Chức năng này hiển thị danh sách của các bài tập tự do trên hệ thống và cho phép người quản trị thêm, chỉnh sửa, xoá bài tập và các tùy chỉnh khác liên quan đến bài tập.

Tên bài tập	Mức độ ↑	Các thông tin khác	Điểm
A + B	Dễ	...	242 ...
A - B	Dễ	...	837 ...
Kiểm tra số nguyên tố	Trung bình	...	316 ...
Số hoàn thiện	Khó	...	316 ...

Hình 3.21. Bản phác thảo giao diện Admin – Quản lý bài tập

3.15.5. Chức năng Quản lý cuộc thi

Chức này hiển thị danh sách các cuộc thi trên hệ thống và cho phép người quản trị tuỳ chỉnh các thông tin của cuộc thi, thêm hoặc xoá một cuộc thi, công bố hoặc ẩn đi một cuộc thi, tuỳ chỉnh các bài tập trong cuộc thi,...

#	Tên cuộc thi	Ngày bắt đầu	Ngày kết thúc	Các thông tin khác	Hành động
1	Contest 1	24/8/2024	25/8/2024	...	Sửa Xoá
2	Contest 2	23/8/2024	24/8/2024	...	Sửa Xoá
3	Contest 3	22/8/2024	23/8/2024	...	Sửa Xoá
4	Contest 4	21/8/2024	22/8/2024	...	Sửa Xoá

Hình 3.22. Bản phác thảo giao diện Admin – Quản lý cuộc thi

3.15.6. Chức năng Quản lý bài viết

Chức năng này cho phép người quản trị xem danh sách các bài viết trên Diễn đàn và

tùy chỉnh chúng như thêm, chỉnh sửa, xoá, ẩn hoặc công bố,... Ngoài ra, khi người dùng đăng bài viết, các bài viết này sẽ vào hàng đợi và chờ duyệt. Người quản trị có thể xem xét để duyệt các bài viết này hoặc loại bỏ chúng nếu có vấn đề về nội dung.

The screenshot shows the HiCommit Admin interface. On the left, there's a sidebar with icons for 'Tổng quan', 'Quản lý người dùng', 'Quản lý khóa học', 'Quản lý bài tập', 'Quản lý cuộc thi', and 'Quản lý bài viết' (which is highlighted). The main area is titled 'Quản lý bài viết' and shows a list of four posts with columns for ID, post title, tags, author, other info, and actions (Edit | Delete). At the bottom right of the main area, there are 'Previous' and 'Next' buttons. The footer of the page is labeled 'Footer'.

#	Tên bài viết	Tags	Tác giả	Các thông tin khác	Hành động
1	Tiêu đề bài viết 1	tag1, tag2	kakanvk	...	Sửa Xoá
2	Tiêu đề bài viết 2	tag1, tag2	kakanvk	...	Sửa Xoá
3	Tiêu đề bài viết 3	tag1, tag2	kakanvk	...	Sửa Xoá
4	Tiêu đề bài viết 4	tag1, tag2	kakanvk	...	Sửa Xoá

Hình 3.23. Bản phác thảo giao diện Admin – Quản lý bài viết

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Bộ dữ liệu thử nghiệm

4.1.1. Dữ liệu thử nghiệm chức năng Khoa học

Bảng 4.1. Dữ liệu thử nghiệm chức năng khoa học

Tên khoa học	Mô tả khoa học
Kỹ thuật lập trình	<p>Thông tin chung:</p> <p>Giảng viên phụ trách học phần: Họ và tên: Nguyễn Ngọc Đan Thanh Nơi làm việc: Bộ môn Công nghệ thông tin, Khoa Kỹ thuật & Công nghệ Email: ngocdanhanhdt@tvu.edu.vn Điện thoại: 0916741252</p> <p>Cách liên lạc với giảng viên: Sinh viên có thể liên hệ để trao đổi các vấn đề liên quan đến bài học qua các hình thức sau: email, điện thoại, trao đổi trên diễn đàn của Khóa học ELearning hoặc gặp trực tiếp tại phòng C71.102 vào các buổi chiều thứ năm.</p> <p>Loại học phần: Cơ sở ngành (CSN)</p> <p>Số tín chỉ: 2 Lý thuyết, 2 Thực hành</p> <p>Đối tượng học: Sinh viên ngành Công nghệ Thông tin</p> <p>CHƯƠNG 1: Mô tả khoa học: Học phần giúp trang bị cho sinh viên các kiến thức cơ bản về kỹ thuật lập trình giải quyết bài toán trên máy tính. Đồng thời học phần cũng nhằm rèn luyện cho sinh viên các kỹ năng phân tích bài toán, lập trình giải các bài toán trên máy tính bằng ngôn ngữ lập trình C với giải thuật hiệu quả. Học phần cũng giúp hình thành cho sinh viên thái độ và nhận thức đúng đắn về tầm quan trọng của học phần và phương pháp lập trình trên máy tính.</p> <p>CHƯƠNG 2: Các chuẩn đầu ra của khoa học: Về kiến thức: - LO1: Vận dụng các cách mô tả thuật toán để diễn đạt ý tưởng giải quyết cho từng bài toán cụ thể - LO2: Tổng hợp các kiến thức về ngôn ngữ lập trình để giải bài toán trên máy tính</p>

	<ul style="list-style-type: none"> - LO3: Sử dụng biến con trỏ với kiểu dữ liệu mảng, chuỗi ký tự và kiểu cấu trúc - LO4: Vận dụng các kỹ thuật lập trình đệ quy để giải quyết bài toán - LO5: Quản lý dữ liệu với tập tin - LO6: Kiểm tra lỗi chương trình - LO7: Đánh giá kết quả chương trình <p>Về kỹ năng chuyên môn:</p> <ul style="list-style-type: none"> - LO8: Thiết kế các thành phần của hệ thống - LO9: Hiện thực hóa hệ thống dựa trên thiết kế - LO10: Kiểm chứng các thành phần hay toàn bộ đề tài <p>Về thái độ và kỹ năng mềm:</p> <ul style="list-style-type: none"> - LO11: Xác định và phát biểu vấn đề - LO12: Mô hình hóa vấn đề - LO13: Suy luận và giải quyết vấn đề - LO14: Tìm kiếm và thu thập thông tin - LO15: Tư duy tổng thể vấn đề - LO16: Học tập chủ động - LO17: Thể hiện tính đạo đức, trung thực và có trách nhiệm xã hội - LO18: Kỹ năng giao tiếp bằng văn bản
--	--

Bài tập thực hành:

Chương 1: Tổng quan về lập trình trên máy tính

Bảng 4.2. Dữ liệu thử nghiệm bài tập 1.1

Tên bài tập	Bài tập 1.1: Hello World
Mã bài tập	HELLO_WORLD
Tags	LO1
Ngôn ngữ lập trình	C
Mô tả đề bài	In ra màn hình dòng chữ "HELLO WORLD"
Input	Không có
Output	Dòng chữ "HELLO WORLD"
Giới hạn	

Các testcases	1	Input: (Trống) Output: HELLO WORLD
---------------	---	---

Bảng 4.3. Dữ liệu thử nghiệm bài tập 1.2

Tên bài tập	Bài tập 1.2: Tính tổng 2 số nguyên	
Mã bài tập	LO1_APB	
Tags	LO1	
Ngôn ngữ lập trình	C	
Mô tả đề bài	Viết một chương trình nhập hai số nguyên và tính tổng của chúng.	
Input	Hai số nguyên a và b.	
Output	Tổng của hai số nguyên a và b.	
Giới hạn		
Các ví dụ	1	Input: 2 3 Output: 5
	2	Input: 5 99 Output: 104

Các testcases	1	Input: 99 4 Output: 103
	2	Input: 2 3 Output: 5
	3	Input: 200000 12345 Output: 212345
	4	Input: 9999999 0 Output: 9999999

Chương 2: Ngôn ngữ lập trình C

Bảng 4.4. Dữ liệu thử nghiệm bài tập 2.1

Tên bài tập	Bài tập 2.1: Tính diện tích hình chữ nhật
Mã bài tập	BT1
Tags	LO2, LO3
Ngôn ngữ lập trình	C
Mô tả đề bài	Viết chương trình nhập vào 2 số thực dương ($a, b > 0$) tương ứng là chiều dài, chiều rộng của một hình chữ nhật. Hãy tính diện tích hình chữ nhật và in ra màn hình.

Input	2 số thực dương a, b ($a > 0, b > 0, a > b$), cách nhau bởi khoảng trắng
Output	Một số thực duy nhất là diện tích hình chữ nhật, làm tròn đến 2 chữ số thập phân
Giới hạn	$0 < b \leq a < 10^4$
Các ví dụ	<p>1 Input: 3 2 Output: 6</p>
	<p>2 Input: 2 3 Output: SAI DU LIEU</p>
Các testcases	<p>1 Input: -5 -2 Output: SAI DU LIEU</p>
	<p>2 Input: 8 7 Output: 56.00</p>
	<p>3 Input: 10001 10002 Output: SAI DU LIEU</p>

	4	Input: 3.5 1.5 Output: 5.25
	5	Input: 7 8 Output: SAI DU LIEU
	6	Input: 10000 0 Output: SAI DU LIEU

Bảng 4.5. Dữ liệu thử nghiệm bài tập 2.2

Tên bài tập	Bài tập 2.2: Tìm số lớn nhất
Mã bài tập	FIND_MAX
Tags	LO2
Ngôn ngữ lập trình	C
Mô tả đề bài	Viết chương trình nhập vào n số nguyên, sau đó tìm số lớn nhất trong các số đó.
Input	Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 100$). Dòng thứ hai chứa n số nguyên ($ a_i \leq 10^6$).
Output	Số nguyên lớn nhất trong dãy.
Giới hạn	$1 \leq n \leq 100$, $ a_i \leq 10^6$

Các ví dụ	1	Input: 3 -10 -5 0 Output: 0
	2	Input: 5 1 2 3 4 5 Output: 5
Các testcases	1	Input: 1 100 Output: 100
	2	Input: 4 -1 -2 -3 -4 Output: -1

	3	Input: 8 5 -5 5 -5 5 -5 Output: 5
	4	Input: 6 7 7 7 7 7 7 Output: 7

Bảng 4.6. Dữ liệu thử nghiệm bài tập 2.3

Tên bài tập	Bài tập 2.3: Tính tổng các số lẻ	
Mã bài tập	SUM_ODD_NUMBERS	
Tags	LO3	
Ngôn ngữ lập trình	C	
Mô tả đề bài	Viết một chương trình tính tổng các số lẻ từ 1 đến một số nguyên n (bao gồm cả n nếu n là số lẻ).	
Input	Một số nguyên dương n ($1 \leq n \leq 10^6$).	
Output	Tổng của các số lẻ từ 1 đến n.	
Giới hạn	$1 \leq n \leq 10^6$	
Các ví dụ	1	Input: 5 Output: 9

Các testcases	2	Input: 10 Output: 25
	1	Input: 6 Output: 9
	2	Input: 1000 Output: 250000
	3	Input: 1 Output: 1
	4	Input: 999999 Output: 250000000000

Chương 3: Thiết kế hàm

Bảng 4.7. Dữ liệu thử nghiệm bài tập 3.1

Tên bài tập	Bài tập 3.1: Kiểm tra số nguyên tố
Mã bài tập	PRIME_CHECK_FUNCTION

Tags	LO3, LO4, LO5				
Ngôn ngữ lập trình	C				
Mô tả đề bài	<p>Viết một hàm isPrime nhận vào một số nguyên dương n và kiểm tra xem n có phải là số nguyên tố hay không. Số nguyên tố là số lớn hơn 1 và chỉ chia hết cho 1 và chính nó.</p> <p>Hàm sẽ trả về giá trị true nếu n là số nguyên tố, ngược lại trả về false.</p>				
Input	Một số nguyên dương n ($1 \leq n \leq 10^6$).				
Output	In ra "YES" nếu n là số nguyên tố, ngược lại in "NO".				
Giới hạn	$1 \leq n \leq 10^6$				
Các ví dụ	<table> <tr> <td style="text-align: center;">1</td> <td>Input: 15 Output: NO</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Input: 29 Output: YES</td> </tr> </table>	1	Input: 15 Output: NO	2	Input: 29 Output: YES
1	Input: 15 Output: NO				
2	Input: 29 Output: YES				
<table> <tr> <td style="text-align: center;">1</td> <td>Input: 1000000 Output: NO</td> </tr> </table>	1	Input: 1000000 Output: NO			
1	Input: 1000000 Output: NO				
Các testcases					

	2	Input: 97 Output: YES
	3	Input: 541 Output: YES
	4	Input: 2 Output: YES

Bảng 4.8. Dữ liệu thử nghiệm bài tập 3.2

Tên bài tập	Bài tập 3.2: Tính ước chung lớn nhất (UCLN)
Mã bài tập	GCD
Tags	LO3, LO4
Ngôn ngữ lập trình	C
Mô tả đề bài	Viết một hàm tính ước chung lớn nhất (UCLN) của hai số nguyên dương a và b bằng thuật toán Euclid. UCLN của hai số là số lớn nhất chia hết cả hai số đó.
Input	Hai số nguyên dương a và b.
Output	UCLN của a và b.
Giới hạn	$1 \leq a, b \leq 10^6$

Các ví dụ	1	Input: 12 8 Output: 4
	2	Input: 100 25 Output: 25
Các testcases	1	Input: 48 18 Output: 6
	2	Input: 1000 250 Output: 250
	3	Input: 13 7 Output: 1
	4	Input: 1024 512 Output: 512

Chương 4: Kiểu dữ liệu mảng

Bảng 4.9. Dữ liệu thử nghiệm bài tập 4.1

Tên bài tập	Bài tập 4.1: Tìm phần tử lớn thứ hai
Mã bài tập	SECOND_LARGEST
Tags	LO4, LO5
Ngôn ngữ lập trình	C
Mô tả đề bài	Cho một mảng gồm n số nguyên dương. Nhiệm vụ của bạn là tìm phần tử lớn thứ hai trong mảng (phần tử khác với phần tử lớn nhất).
Input	Dòng đầu tiên chứa số nguyên n ($2 \leq n \leq 1000$). Dòng thứ hai chứa n số nguyên dương ($1 \leq a_i \leq 10^6$).
Output	In ra phần tử lớn thứ hai trong mảng.
Giới hạn	$2 \leq n \leq 1000$, $1 \leq a_i \leq 10^6$
Các ví dụ	<p>1 Input: 5 1 3 4 5 2 Output: 4</p>
	<p>2 Input: 4 9 2 7 7 Output: 7</p>

Các testcases	1	Input: 3 1000000 999999 1000000 Output: 999999
	2	Input: 7 10 15 15 20 20 10 5 Output: 15
	3	Input: 6 1 1 1 1 2 2 Output: 1
	4	Input: 10 1 9 7 7 3 5 8 9 10 10 Output: 9

Bảng 4.10. Dữ liệu thử nghiệm bài tập 4.2

Tên bài tập	Bài tập 4.2: Đếm số phần tử lẻ trong mảng
Mã bài tập	COUNT_ODD
Tags	LO4, LO5

Ngôn ngữ lập trình	C
Mô tả đề bài	Cho một mảng gồm n số nguyên. Nhiệm vụ của bạn là đếm xem có bao nhiêu số lẻ trong mảng.
Input	Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 1000$). Dòng thứ hai chứa n số nguyên ($ a_i \leq 10^6$).
Output	In ra số lượng phần tử lẻ trong mảng.
Giới hạn	$1 \leq n \leq 1000$, $ a_i \leq 10^6$
Các ví dụ	<p>Input: 5 1 2 3 4 5 Output: 3</p>
	<p>Input: 4 2 4 6 8 Output: 0</p>
Các testcases	<p>Input: 6 -4 -2 0 1 2 4 Output: 1</p>

	2	Input: 10 1 3 5 7 9 2 4 6 8 10 Output: 5
	3	Input: 3 7 7 7 Output: 3
	4	Input: 8 0 5 -3 7 9 -11 6 2 Output: 5

Chương 5: Kỹ thuật đệ quy

Bảng 4.11. Dữ liệu thử nghiệm bài tập 5.1

Tên bài tập	Bài tập 5.1: Tính số Fibonacci
Mã bài tập	FIBONACCI_RECUSION
Tags	LO6, LO7, LO8
Ngôn ngữ lập trình	C
Mô tả đề bài	Viết một hàm đệ quy để tính số Fibonacci thứ n. Số Fibonacci được định nghĩa như sau: $F(0) = 0, F(1) = 1$

	$F(n) = F(n-1) + F(n-2)$ với $n \geq 2$.
Input	Một số nguyên n ($0 \leq n \leq 30$).
Output	Số Fibonacci thứ n .
Giới hạn	$0 \leq n \leq 30$
Các ví dụ	<p>1 Input: 5 Output: 5</p>
	<p>2 Input: 10 Output: 55</p>
Các testcases	<p>1 Input: 1 Output: 1</p>
	<p>2 Input: 15 Output: 610</p>
	<p>3 Input: 30 Output: 832040</p>

	4	Input: 20 Output: 6765
--	---	---------------------------------

Bảng 4.12. Dữ liệu thử nghiệm bài tập 5.2

Tên bài tập	Bài tập 5.2: Tính giai thừa	
Mã bài tập	FACTORIAL_RECUSION	
Tags	LO5, LO6	
Ngôn ngữ lập trình	C	
Mô tả đề bài	<p>Viết một hàm đệ quy để tính giai thừa của một số nguyên n. Giai thừa của n được định nghĩa như sau:</p> $n! = n * (n-1) * (n-2) * \dots * 1 \text{ với } n \geq 1$ $0! = 1$	
Input	Một số nguyên n ($0 \leq n \leq 12$).	
Output	Giai thừa của n.	
Giới hạn	$0 \leq n \leq 12$	
Các ví dụ	1	Input: 5 Output: 120

Các testcases	2	Input: 0 Output: 1
	1	Input: 6 Output: 720
	2	Input: 1 Output: 1
	3	Input: 3 Output: 6
	4	Input: 12 Output: 479001600
	5	Input: 7 Output: 5040

Chương 6: Chuỗi ký tự và kiểu dữ liệu có cấu trúc

Bảng 4.13. Dữ liệu thử nghiệm bài tập 6.1

Tên bài tập	Bài tập 6.1: Đếm số ký tự trong chuỗi	
Mã bài tập	COUNT_CHARACTERS	
Tags	LO6, LO7, LO8, LO10	
Ngôn ngữ lập trình	C	
Mô tả đề bài	Viết một hàm nhận vào một chuỗi ký tự và đếm số lượng ký tự trong chuỗi đó. Hàm cần trả về số lượng ký tự không bao gồm các ký tự khoảng trắng.	
Input	Một chuỗi ký tự, độ dài tối đa là 1000 ký tự.	
Output	In ra số lượng ký tự không phải là khoảng trắng trong chuỗi.	
Giới hạn	Độ dài chuỗi ≤ 1000	
Các ví dụ	1	Input: Hello World Output: 10
	2	Input: Programming in C is fun Output: 24
Các testcases	1	Input: 123 456 Output: 6

	2	Input: Leading and trailing spaces Output: 29
	3	Input: A B C D E Output: 5
	4	Input: C++ Output: 3

Bảng 4.14. Dữ liệu thử nghiệm bài tập 6.2

Tên bài tập	Bài tập 6.2: Đảo ngược chuỗi
Mã bài tập	REVERSE_STRING
Tags	LO6, LO7, LO8, LO10
Ngôn ngữ lập trình	C
Mô tả đề bài	Viết một hàm nhận vào một chuỗi ký tự và trả về chuỗi đó đã được đảo ngược.
Input	Một chuỗi ký tự, độ dài tối đa là 1000 ký tự.
Output	In ra chuỗi đã được đảo ngược.
Giới hạn	Độ dài chuỗi ≤ 1000

Các ví dụ	1	Input: Hello Output: olleH
	2	Input: C Programming Output: gnimmargorP C
Các testcases	1	Input: The quick brown fox Output: xof nworb kciuq ehT
	2	Input: A B C Output: C B A
	3	Input: 12345 Output: 54321
	4	Input: abcdefghijklmnopqrstuvwxyz Output: zyxwvutsrqponmlkjihgfedcba

4.1.2. Dữ liệu thử nghiệm chức năng Luyện tập

Bảng 4.15. Dữ liệu thử nghiệm bài tập Chuỗi đối xứng

Tên bài tập	Chuỗi đối xứng		
Mã bài tập	PALINDROME_CHECK		
Tags	Chuỗi		
Ngôn ngữ lập trình	C++		
Cấp độ	Khó		
Mô tả đề bài	Viết một chương trình kiểm tra xem một chuỗi ký tự có phải là chuỗi đối xứng hay không.		
Input	Một chuỗi ký tự (độ dài ≤ 1000).		
Output	In ra "YES" nếu chuỗi là đối xứng, "NO" nếu không phải.		
Giới hạn			
Các ví dụ	1	Input: abc Output: NO	
	2	Input: abba Output: YES	
Các testcases	1	Input: racecar Output: NO	

	2	Input: 123454321 Output: YES
	3	Input: a Output: YES
	4	Input: hello Output: NO

Bảng 4.16. Dữ liệu thử nghiệm bài tập Tính tổng các chữ số

Tên bài tập	Tính tổng các chữ số
Mã bài tập	SUM_OF_DIGITS
Tags	Tổng
Ngôn ngữ lập trình	C++
Cấp độ	Trung bình
Mô tả đề bài	Viết một chương trình tính tổng các chữ số của một số nguyên dương.
Input	Một số nguyên dương n ($1 \leq n \leq 10^9$).
Output	Tổng các chữ số của n.
Giới hạn	$1 \leq n \leq 10^9$

Các ví dụ	1	Input: 987654 Output: 39
	2	Input: 12345 Output: 15
Các testcases	1	Input: 100000009 Output: 10
	2	Input: 100001 Output: 2
	3	Input: 9 Output: 9
	4	Input: 999999 Output: 54

Bảng 4.17. Dữ liệu thử nghiệm bài tập Kiểm tra số nguyên tố

Tên bài tập	Kiểm tra số nguyên tố	
Mã bài tập	PRIME2	
Tags	Số nguyên tố	
Ngôn ngữ lập trình	Java	
Cấp độ	Để	
Mô tả đề bài	<p>Viết một chương trình Java để kiểm tra xem một số nguyên dương có phải là số nguyên tố hay không. Số nguyên tố là một số nguyên lớn hơn 1 và chỉ chia hết cho 1 và chính nó.</p> <p>Viết một hàm isPrime(int number) nhận vào một số nguyên dương và trả về true nếu số đó là số nguyên tố, ngược lại trả về false.</p> <p>Trong hàm main, đọc một số nguyên từ đầu vào, kiểm tra nó có phải là số nguyên tố không và in ra kết quả.</p>	
Input	Một số nguyên dương n.	
Output	Một chuỗi "YES" nếu số đó là số nguyên tố, ngược lại in ra "NO".	
Giới hạn		
Các ví dụ	1	<p>Input:</p> <p>2</p> <p>Output:</p> <p>YES</p>

Các testcases	2	Input: 4 Output: NO
	1	Input: 17 Output: YES
	2	Input: 2 Output: YES
	3	Input: 4 Output: NO
	4	Input: 97 Output: YES

4.1.3. Dữ liệu thử nghiệm chức năng Cuộc thi

Bảng 4.18. Dữ liệu thử nghiệm chức năng Cuộc thi

Tên cuộc thi	HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024
---------------------	---

Mã cuộc thi	tvu_olympic_2024
Mô tả cuộc thi	[Replay] HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024 Địa điểm: Toà C7, Trường Đại học Trà Vinh Thời gian: 13h, 10.10.2024
Thời gian bắt đầu	10/10/2024, 13h00
Thời gian làm bài	02 giờ 00 phút
Mật khẩu tham gia	tvu_olympic_2024@

Các bài tập trong cuộc thi:

Bảng 4.19. *Dữ liệu thử nghiệm bài tập Tổng ước số nguyên tố*

Tên bài tập	Tổng ước số nguyên tố
Mã bài tập	tvu_olympic_2024_A
Tags	tvu_olympic_2024
Ngôn ngữ lập trình	C++
Điểm số	100
Mô tả đề bài	Cho một số nguyên dương N ($1 \leq N \leq 10^6$). Viết chương trình để tính tổng các ước số nguyên tố của N . Biết rằng một số nguyên $x > 1$ được gọi là số nguyên tố nếu x chỉ chia hết cho 1 và chính nó.
Input	Một số nguyên dương N ($1 \leq N \leq 10^6$)
Output	Tổng các ước số nguyên tố của N
Giới hạn	

Các testcases	Các ví dụ	1	Input: 12 Output: 5
		1	Input: 999999 Output: 71
		2	Input: 12345 Output: 831
		3	Input: 1234 Output: 619
		4	Input: 123 Output: 44

Bảng 4.20. Dữ liệu thử nghiệm bài tập Tiên thùa của Hiệp

Tên bài tập	Tiền thùa của Hiệp
Mã bài tập	tvu_olympic_2024_B
Tags	tvu_olympic_2024

Ngôn ngữ lập trình	C++
Điểm số	100
Mô tả đề bài	Bạn Hiệp được mẹ đưa cho tờ 500.000đ để đến cửa hàng thực phẩm TVU mua một số mặt hàng. Cửa hàng thực phẩm TVU có các mệnh giá tiền: 1.000đ, 5.000đ, 10.000đ, 50.000đ và 100.000đ; và đơn giá của các mặt hàng đều chia hết cho 1.000. Hiệp mua thực phẩm với tổng tiền là N ($1.000 \leq N \leq 499.000$). Hãy giúp cửa hàng trả lại số tiền K còn lại của Hiệp sao cho với số tờ ít nhất.
Input	Số nguyên dương N ($1.000 \leq N \leq 499.000$ và N chia hết cho 1.000) : là số tiền Hiệp đã mua
Output	Số nguyên dương K: là số lượng tờ tiền ít nhất mà cửa hàng TVU phải trả lại.
Giới hạn	
Các ví dụ	1 Input: 273000 Output: 7
Các testcases	1 Input: 499000 Output: 1

	2	Input: 391000 Output: 6
	3	Input: 150000 Output: 4
	4	Input: 266000 Output: 9
	5	Input: 100000 Output: 4

Bảng 4.21. Dữ liệu thử nghiệm bài tập Con éch thông minh

Tên bài tập	Con éch thông minh
Mã bài tập	tvu_olympic_2024_C
Tags	tvu_olympic_2024
Ngôn ngữ lập trình	C++
Điểm số	100
Mô tả đề bài	Có N hòn đá được đánh số từ 1 đến N. Với mỗi hòn đá thứ i ($1 \leq i \leq N$) có độ cao tương ứng là hi. Một chú éch

	<p>ban đầu đang ngồi ở hòn đá thứ nhất và chú éch sẽ thực hiện liên tục các hành động sau:</p> <p>Nếu chú éch ở hòn đá thứ i thì chú có thể nhảy tới hòn đá thứ $i+1$ hoặc $i+2$ và chú sẽ mất chi phí là $h_i - h_j$ với h_j là hòn đá chú nhảy đến.</p> <p>Hãy viết chương trình giúp chú éch nhảy từ hòn đá thứ nhất đến hòn đá thứ N với tổng chi phí thấp nhất.</p>		
Input	<ul style="list-style-type: none"> - Số nguyên dương N ($2 \leq N \leq 10^4$) : là số lượng hòn đá. - Dãy N số nguyên h_i ($1 \leq i \leq N$, $1 \leq h_i \leq 10^3$): là độ cao của hòn đá i. 		
Output	Số nguyên M : là tổng chi phí nhỏ nhất để chú éch nhảy từ hòn đá thứ nhất đến hòn đá thứ N .		
Giới hạn			
Các ví dụ	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;">1</td> <td style="width: 90%;"> <p>Input: 4 1 4 5 3</p> <p>Output: 4</p> </td> </tr> </table>	1	<p>Input: 4 1 4 5 3</p> <p>Output: 4</p>
1	<p>Input: 4 1 4 5 3</p> <p>Output: 4</p>		
Các testcases	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;">1</td> <td style="width: 90%;"> <p>Input: 3 1 3 2</p> <p>Output: 1</p> </td> </tr> </table>	1	<p>Input: 3 1 3 2</p> <p>Output: 1</p>
1	<p>Input: 3 1 3 2</p> <p>Output: 1</p>		

	2	Input: 7 3 3 4 4 3 4 4 Output: 1
	3	Input: 4 10 30 40 20 Output: 30
	4	Input: 6 30 10 60 10 60 50 Output: 40
	5	Input: 2 9 9 Output: 0

Bảng 4.22. Dữ liệu thử nghiệm bài tập Xây dựng mạng lưới viễn thông

Tên bài tập	Xây dựng mạng lưới viễn thông
Mã bài tập	tvu_olympic_2024_D
Tags	tvu_olympic_2024

Ngôn ngữ lập trình	C++
Điểm số	100
Mô tả đề bài	Sau trận bão lịch sử, hệ thống mạng lưới viễn thông của vương quốc Gamma bị phá hủy nặng nề. Chính phủ cần xây dựng lại hệ thống mạng lưới viễn thông với mục tiêu là thiết lập đường truyền kết nối được tất cả các thành phố lại với nhau (giữa 2 thành phố có thể được kết nối trực tiếp hoặc gián tiếp). Để kết nối N thành phố, M đường truyền giữa 2 thành phố được đề xuất xây dựng, gồm chi phí xây dựng và chi phí bảo trì trong 5 năm. Hãy giúp Chính phủ nước Gamma tính toán chi phí tối thiểu để xây dựng đường truyền liên thông giữa N thành phố đó.
Input	<ul style="list-style-type: none"> - Dòng đầu tiên chứa 2 số nguyên N và M ($1 \leq N \leq 10^3$, $1 \leq M \leq 499500$) cách nhau bởi khoảng trắng. - M dòng tiếp theo, mỗi dòng gồm 4 số nguyên dương u, v, c, r cách nhau bởi khoảng trắng. Trong đó u, v là 2 thành phố được xây dựng đường truyền, c là chi phí xây dựng, r là chi phí bảo trì.
Output	Danh sách các đường truyền được chọn và chi phí tối thiểu để xây dựng đường truyền liên thông giữa N thành phố.
Giới hạn	

Các ví dụ	1	<p>Input:</p> <p>4 5 1 2 3 2 1 3 4 3 2 3 2 1 2 4 5 2 3 4 1 1</p> <p>Output:</p> <p>1 2 2 3 3 4</p> <p>10</p>
-----------	---	--

		Input:
		8 14
		1 2 2 1
		1 4 1 2
		2 3 1 1
		2 4 4 2
		2 5 1 1
		3 5 3 1
		3 8 5 3
		4 5 2 2
		4 6 6 2
		4 7 1 1
Các testcases	1	5 7 2 1
		5 8 7 3
		6 7 4 1
		7 8 3 1
		Output:
		1 2
		2 3
		1 4
		2 5
		7 6
		4 7
		7 8

		21
	2	<p>Input:</p> <p>5 7</p> <p>1 2 2 1</p> <p>1 3 3 2</p> <p>2 3 1 3</p> <p>2 4 4 2</p> <p>3 4 5 1</p> <p>3 5 6 3</p> <p>4 5 2 2</p> <p>Output:</p> <p>1 2</p> <p>2 3</p> <p>2 4</p> <p>4 5</p> <p>17</p>

		Input:
		6 8
		1 2 4 1
		1 3 2 3
		1 4 5 2
		2 3 3 1
		2 5 4 2
		3 4 1 1
		4 5 2 3
	3	5 6 6 2
		Output:
		1 2
		2 3
		3 4
		4 5
		5 6
		24

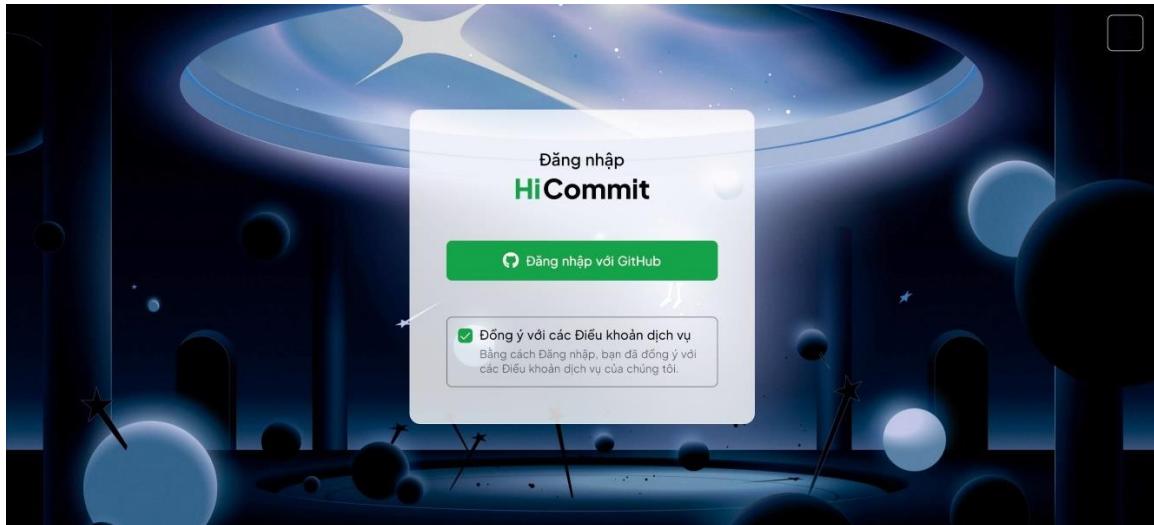
		Input:
		8 12
		1 2 3 2
		1 3 4 1
		1 4 5 2
		2 3 1 3
		2 5 4 1
		3 4 2 2
		3 6 6 3
		4 5 2 1
		5 6 1 1
	4	5 7 5 2
		6 8 3 1
		7 8 4 2
		Output:
		1 2
		2 3
		3 4
		4 5
		5 6
		8 7
		6 8
		28

		Input:
		7 10
		1 2 5 2
		1 3 3 1
		1 4 4 3
		2 3 2 1
		2 5 6 2
		3 4 2 1
		3 5 3 2
		4 6 7 1
5	5	5 6 5 2
		5 7 1 3
		Output:
		3 2
		1 3
		3 4
		3 5
		5 6
		5 7
		26

4.3. Kết quả thực nghiệm với vai trò sinh viên

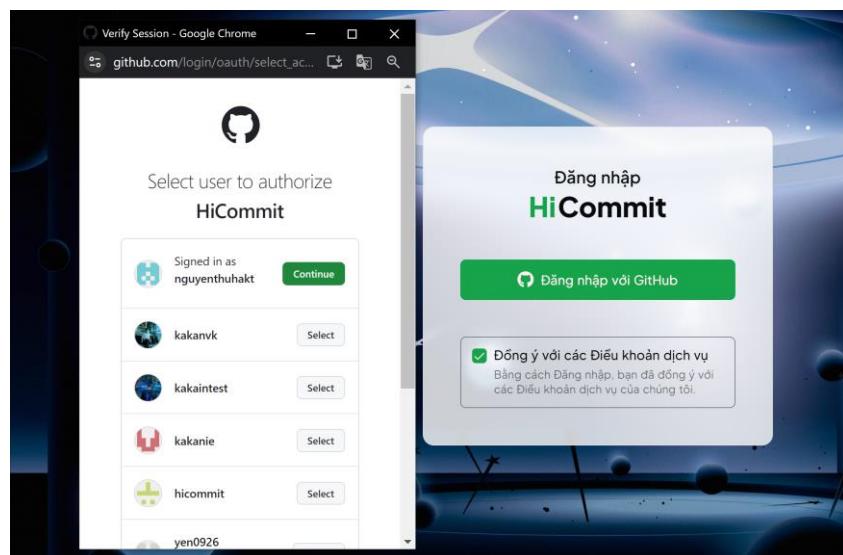
4.3.1. Chức năng Đăng nhập

Để bắt đầu sử dụng các tính năng trên hệ thống HiCommit, người dùng bắt buộc phải qua bước đăng nhập. Ở bước này, chỉ có một phương thức đăng nhập duy nhất là sử dụng tài khoản GitHub.



Hình 4.1. Giao diện thực tế chức năng Đăng nhập

Khi chọn Đăng nhập với GitHub, một cửa sổ hiện lên cho phép chọn các tài khoản GitHub để đăng nhập. Người dùng cần chọn một tài khoản để sử dụng, nếu chỉ có một tài khoản duy nhất, hệ thống sẽ tự động đăng nhập vào tài khoản này.



Hình 4.2. Giao diện chọn tài khoản đăng nhập GitHub

Sau khi đăng nhập thành công, hệ thống sẽ tự động chuyển hướng đến trang chủ.

4.3.2. Chức năng Trang chủ

Trang chủ hiển thị nhanh một số khoá học gần đây mà người dùng đã tương tác.

The screenshot shows the HiCommit homepage with a sidebar on the left containing links for Trang chủ, Khoa học, Luyện tập, Cuộc thi, and Diễn đàn. A search bar at the top right says "Tim kiếm khoá học, bài tập...". On the right, there's a user profile for "Ka Ka" with an email "caoka5265@gmail.com" and a dropdown menu with options: Dành cho quản trị viên, Tài khoản của tôi, Được đánh dấu, Thông báo, Cài đặt, and Đăng xuất. Below the profile are three course cards:

- Kỹ thuật lập trình**: Sau khi học khoá học này, sẽ có được các kỹ năng cần thiết để lập trình.
Vào khoá học >
- Lập trình hướng đối tượng**: Sau khi học khoá học này, sẽ có được các kỹ năng cần thiết để lập trình.
Vào khoá học >
- Công nghệ phần mềm**: Sau khi học khoá học này, sẽ có được các kỹ năng cần thiết để lập trình.
Vào khoá học >

A circular progress bar at the bottom right indicates "1/3".

Hình 4.3. Giao diện thực tế chức năng Trang chủ

4.3.3. Chức năng Khoa học

Chức năng này hiển thị danh sách các khoá học (đã được công bố) trên hệ thống, được chia thành hai tab: Tất cả khoá học (mặc định) và Đã tham gia. Với tab đầu tiên, tất cả các khoá học trên hệ thống sẽ được hiển thị. Với tab “Đã tham gia”, sẽ chỉ hiển thị các khoá học mà người dùng đang theo học. Tại đây cũng hỗ trợ một số tùy chọn như tìm kiếm, lọc để người dùng thuận tiện hơn trong việc chọn ra khoá học cần tìm.

The screenshot shows the "Khoa học" tab on the HiCommit homepage. The sidebar remains the same. The main area displays three course cards under the heading "Tất cả khoá học":

- Kỹ thuật lập trình**: Được tạo bởi kakanvk • DA2ITA
Thông tin chung:
Giảng viên phụ trách học phần: ...
1. 13 bài tập | 2. 2
Tham gia
- An toàn và bảo mật thông tin - Cơ bản**: Được tạo bởi kakanvk • DA2ITA
An toàn và Bảo mật Thông tin là một môn học chuyên sâu tập trung vào các phươn...
1. 9 bài tập | 2. 4
Tham gia
- Lập trình hướng đối tượng**: Được tạo bởi hicommit • DA2ITA
You can also use variant modifiers to target media queries like responsive breakpoint...
1. 0 bài tập | 2. 123
Tham gia

Hình 4.4. Giao diện thực tế chức năng Khoa học

Các khóa học > Kỹ thuật lập trình

Kỹ thuật lập trình DA24TTA

Được tạo bởi  kakanvk • 1 tuần trước A 4

Thông tin khoa học:

Thông tin chung:

Giảng viên phụ trách học phần:
Họ và tên: Nguyễn Ngọc Đan Thanh
Nơi làm việc: Bộ môn Công nghệ thông tin, Khoa Kỹ thuật & Công nghệ
Email: ngocdanhanhdkt@tvt.edu.vn
Điện thoại: 0916741252

Cách liên lạc với giảng viên: Sinh viên có thể liên hệ để trao đổi các vấn đề liên quan đến bài học qua các hình thức sau: email, điện thoại, trao đổi trên diễn đàn của Khóa học ELearning, hoặc gặp trực tiếp tại phòng C71.102 vào các buổi chiều thứ năm.



DA24TTA Kỹ thuật lập trình

Bạn chưa tham gia khóa học này

Tham gia ngay

Hình 4.5. Giao diện khi chưa tham gia khóa học

Khoa học được chia thành hai loại: khoa học công khai và khoa học riêng tư. Để tham gia một khoa học riêng tư, người dùng cần cung cấp mật khẩu tham gia (join key) để tham gia khoa học này. Mật khẩu này được giảng viên cung cấp. Với khoa học công khai, người dùng có thể tham gia một cách dễ dàng mà không cần cung cấp mật khẩu.

HiCommit

Tìm kiếm khóa học, bài tập

Trang chủ

Khoa học

Luyện tập

Cuộc thi

Điểm danh

Danh sách giáo viên

Quản lý khóa học

Các khóa học > Kỹ thuật lập trình

Được tạo bởi

Kỹ thuật lập trình

Đã xác nhận

Được tạo bởi

Thông tin khóa học

Thông tin chung

Giảng viên phụ trách

Họ và tên: Nguyễn Văn Hùng

Nơi làm việc: Khoa Công nghệ

Email: ngocdanthanhnd@hnu.edu.vn

Điện thoại: 0916741252

Xác nhận tham gia khóa học

Bạn có chắc chắn rằng bạn muốn tham gia khóa học này.

Mời tham gia

Đóng Xác nhận

Đã xác nhận

Bạn chưa thêm qua khóa học này

Tham gia ngay

Hình 4.6. Giao diện khi được yêu cầu nhập mật khẩu khoá học

Các thông tin chính của khoá học bao gồm thông tin khoá học, bài tập thực hành,...

Các khoa học > Kỹ thuật lập trình

Kỹ thuật lập trình DA24ITA

Được tạo bởi  kakanvk • 1 tuần trước  5

Thông tin khoa học:

Thông tin chung:

Giảng viên phụ trách học phần:

Họ và tên: Nguyễn Ngọc Đan Thanh
Nơi làm việc: Bộ môn Công nghệ thông tin, Khoa Kỹ thuật & Công nghệ
Email: ngocdanthanhd@tuv.edu.vn
Điện thoại: 0916741252

Cách liên lạc với giảng viên: Sinh viên có thể liên hệ để trao đổi các vấn đề liên quan đến bài học qua các hình thức sau: email, điện thoại, trao đổi trên diễn đàn của Khóa học ELearning, hoặc gặp trực tiếp tại phòng C71.102 vào các buổi chiều thứ năm.



Tiến độ khoa học



31%

-  Kết quả chính xác: 4/13
-  Sai kết quả: 9/13
-  Gặp vấn đề: 0/13
-  Lỗi biên dịch: 0/13
-  Chưa nộp bài: 0/13

Hình 4.7. Giao diện chi tiết trong một khoá học

Các chuẩn đầu ra của khoá học:

Về kiến thức:

- LO1: Vận dụng các cách mô tả thuật toán để diễn đạt ý tưởng giải quyết cho từng bài toán cụ thể
- LO2: Tổng hợp các kiến thức về ngôn ngữ lập trình để giải bài toán trên máy tính
- LO3: Sử dụng biến con trỏ với kiểu dữ liệu mảng, chuỗi ký tự và kiểu cấu trúc
- LO4: Vận dụng các kỹ thuật lập trình để quy để giải quyết bài toán
- LO5: Quản lý dữ liệu với tập tin
- LO6: Kiểm tra lỗi chương trình
- LO7: Đánh giá kết quả chương trình

Về kỹ năng chuyên môn:

- LO8: Thiết kế các thành phần của hệ thống
- LO9: Hiện thực hóa hệ thống dựa trên thiết kế
- LO10: Kiểm chứng các thành phần hay toàn bộ đề tài

Về thái độ và kỹ năng mềm:

- LO11: Xác định và phát biểu vấn đề
- LO12: Mô hình hóa vấn đề
- LO13: Suy luận và giải quyết vấn đề
- LO14: Tìm kiếm và thu thập thông tin
- LO15: Tư duy tổng thể vấn đề
- LO16: Học tập chủ động
- LO17: Thể hiện tính đạo đức, trung thực và có trách nhiệm xã hội
- LO18: Kỹ năng giao tiếp bằng văn bản

Hình 4.8. Thông tin các chuẩn đầu ra của một khoá học

Bài tập thực hành:

- Chương 1: Tổng quan về lập trình trên máy tính**
 - ✓ Bài tập 1.1: Hello World LO1
 - ✗ Bài tập 2.1: Tính diện tích hình chữ nhật LO2 CDR2
 - ✓ Bài tập 1.2: Tính tổng 2 số nguyên LO1
- Chương 2: Ngôn ngữ lập trình C**
 - ✓ Bài tập 2.2: Tìm số lớn nhất LO2
 - ✓ Bài tập 2.3: Tính tổng các số lẻ LO1
- Chương 3: Thiết kế hàm**
 - ✗ Bài tập 3.1: Kiểm tra số nguyên tố LO3 LO4 LO5
 - ✗ Bài tập 3.2: Tính ước chung lớn nhất (UCLN) LO3 LO4

Tiến độ khoá học



31%

✓ Kết quả chính xác: 4/13
✗ Sai kết quả: 9/13
⚠ Gặp vấn đề: 0/13
△ Lỗi biên dịch: 0/13
● Chưa nộp bài: 0/13

Hình 4.9. Giao diện phần thực hành trong một khoá học

Phần thực hành của khoá học bao gồm các chương và các bài tập (sẽ nói rõ ở phần Bài tập). Ngoài ra, người dùng có thể phân tích dữ liệu khoá học bằng cách bấm vào nút phân tích ngay cạnh tên khoá học:

Các khoá học > Kỹ thuật lập trình

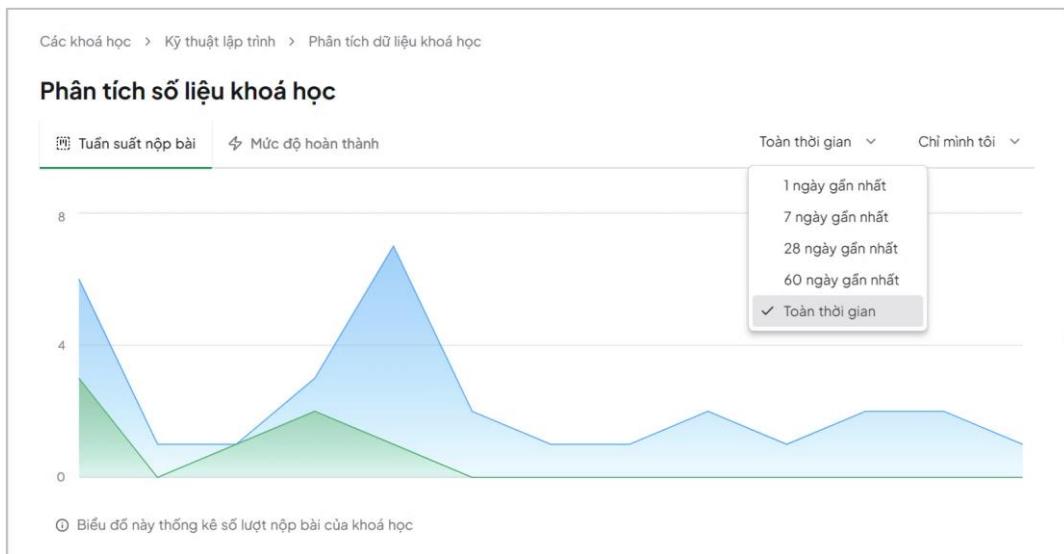
Kỹ thuật lập trình DA24TTA

↓ Được tạo bởi  kakanvk • 1 tuần trước 

Hình 4.10. Chức năng phân tích số liệu khoá học

Với chức năng này, người dùng có thể xem thống kê các lượt nộp bài và tỉ lệ đúng trong nhiều mốc thời gian, có thể chọn hiển thị dữ liệu của tất cả các thành viên hoặc chỉ duy nhất người dùng hiện tại. Điều này giúp người dùng dễ dàng theo dõi được quá trình học tập trên khoá học.



Hình 4.11. Giao diện chức năng phân tích số liệu khoá học – Tần suất nộp bài

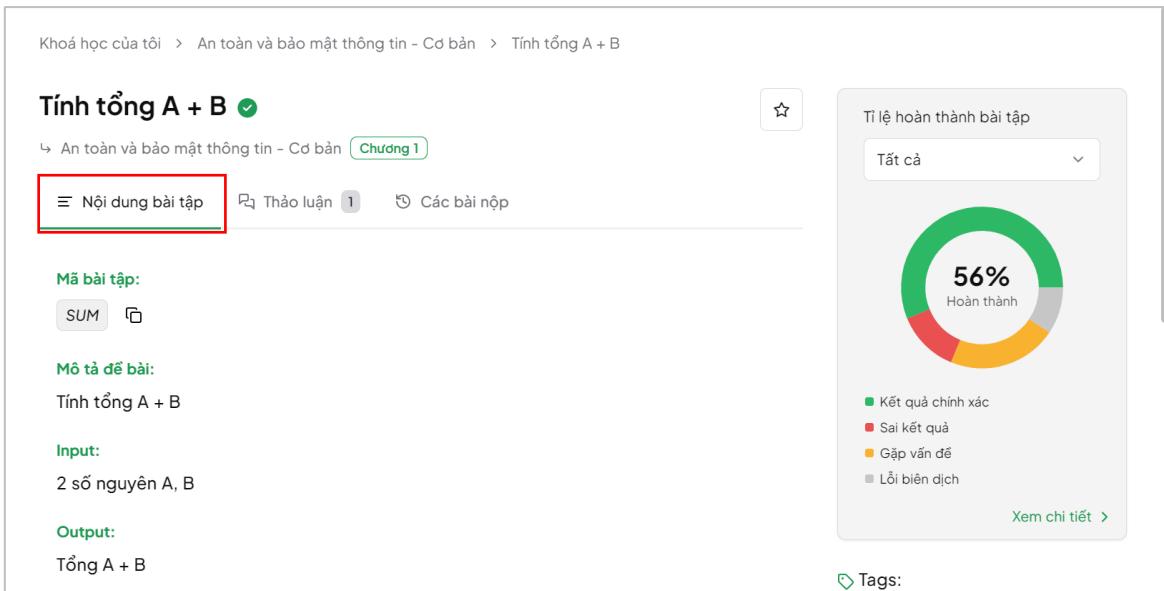


Hình 4.12. Giao diện chức năng phân tích số liệu khoá học – Mức độ hoàn thành

4.3.4. Chức năng Bài tập

Một bài tập sẽ có bố cục: bên trái là các thông tin bài tập và tab chức năng, bên phải là biểu đồ trực quan hoá thống kê về dữ liệu nộp bài (theo cá nhân hoặc toàn bộ người nộp).

Tab đầu tiên là “Nội dung bài tập” chứa các thông tin về bài tập như mã bài tập, mô tả đề bài, mô tả input, mô tả output, các ví dụ, hướng dẫn nộp bài,... và quan trọng nhất là nút nộp bài.



Hình 4.13. Giao diện thực tế một bài tập trên HiCommit – Nội dung bài tập

Bên dưới là các ví dụ, hướng dẫn nộp bài và nút gửi bài giải.

Hình 4.14. Các ví dụ của bài tập và chức năng nộp bài

Khi người dùng nhấn gửi bài giải, hệ thống sẽ chuyển sang trang nộp bài. Tại đây, mã nguồn sẽ tự động được khởi tạo theo ngôn ngữ của bài tập. Nhiệm vụ của người dùng là điền phần mã nguồn của mình vào khu vực soạn thảo. Hoặc có thể tải lên file mã

nguồn từ máy tính thông qua nút tải lên.

The screenshot shows the HiCommit platform's submission interface. On the left, there's a sidebar with navigation links: Trang chủ, Khoa học, Luyện tập, Cuộc thi, Diễn đàn, Dành cho giáo viên, and Quản lý khóa học. The main area displays a submission titled "Nộp bài: Tính tổng A + B" with a green checkmark. Below it, a breadcrumb trail shows: An toàn và bảo mật thông tin - Cơ bản > Tính tổng A + B > Nộp bài. A sub-breadcrumb indicates "Chương 1". A yellow warning box at the top right says "Vui lòng KHÔNG sử dụng tiếng Việt có dấu trong mã nguồn để tránh xảy ra lỗi". The code editor contains the following C++ code:

```
3 using namespace std;
4
5 v int main() {
6     ios_base::sync_with_stdio(false);
7     cin.tie(NULL);
8
9     int a, b;
10
11    cin >> a >> b;
12    cout << a + b;
13
14    return 0;
15 }
```

Below the code editor, there are buttons for "Quay lại", "C++" (language selector), and "Nộp bài" (submit button). The "Nộp bài" button has a small icon of a person running.

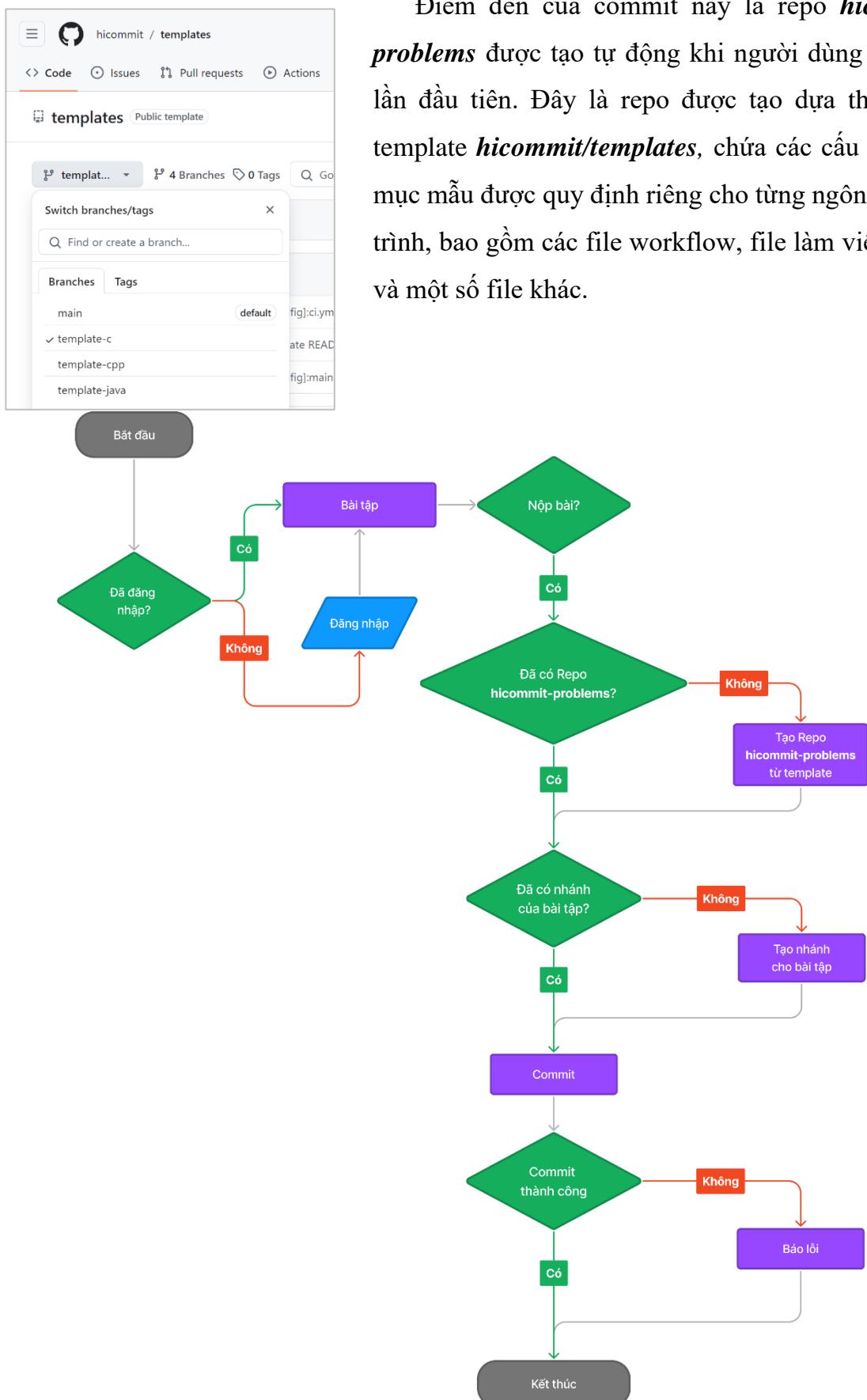
Hình 4.15. Chức năng nộp bài thông qua giao diện HiCommit

Khi mọi thứ đã sẵn sàng, chúng ta tiến hành nộp bài bằng cách nhấn vào nút nộp bài phía dưới. Một hộp thoại hiện lên yêu cầu nhập thông điệp của lần nộp bài để tiện cho việc quản lý. Sau khi nhập, nhấn Commit để hoàn tất quá trình nộp bài.

A modal dialog box titled "Commit thay đổi" (Commit change) is displayed. It contains the message: "Hãy để lại một thông điệp cho lần nộp bài này:" (Please leave a message for this submission). Below this is a text input field with the placeholder "Submit code by kakanvk - Test Program". At the bottom of the dialog are two buttons: "Đóng" (Close) and a green "Commit" button.

Hình 4.16. Commit thay đổi trước khi nộp bài

Điểm đến của commit này là repo ***hicommits-problems*** được tạo tự động khi người dùng nộp bài lần đầu tiên. Đây là repo được tạo dựa theo repo template ***hicommits/templates***, chứa các cấu trúc thư mục mẫu được quy định riêng cho từng ngôn ngữ lập trình, bao gồm các file workflow, file làm việc chính và một số file khác.



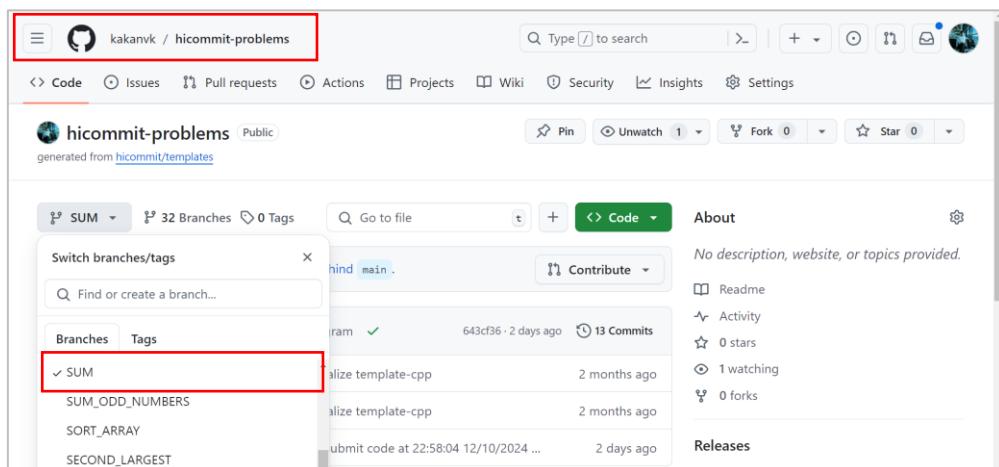
Bảng 4.23. Sơ đồ hoạt động quá trình nộp bài

Sau khi nhập thông điệp và nhấn “Commit”, quá trình nộp bài sẽ được diễn ra như sau:

- **Bước 1:** Hệ thống cần chắn chắn rằng người dùng đã đăng nhập, nếu chưa đăng nhập, hệ thống cần chuyển hướng đến trang đăng nhập và yêu cầu người dùng đăng nhập.

- **Bước 2:** Trước khi nộp bài, hệ thống cần kiểm tra repo **hicommit-problems** có tồn tại trên GitHub của người dùng hay không. Nếu không tồn tại, cần phải tạo mới trước khi nộp.

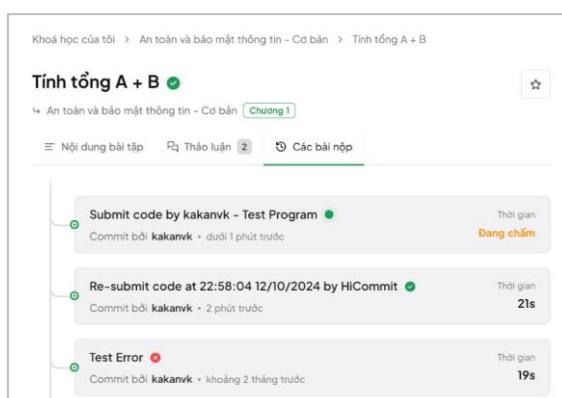
- **Bước 3:** Sau khi chắc chắn đã tồn tại repo **hicommit-problems**, tiếp theo cần kiểm tra xem nhánh tương ứng với bài tập đã được khởi tạo chưa. Mỗi bài tập trên hệ thống sẽ tương ứng với một nhánh trong repo **hicommit-problems**, các nhánh được đặt tên trùng với mã bài tập để dễ dàng quản lý. Khi nộp bài, mã nguồn sẽ được commit vào nhánh tương ứng. Nếu chưa tồn tại nhánh của bài tập, cần tạo mới nhánh để tiến hành nộp bài.



Bảng 4.24. Nhánh tương ứng với bài tập trên GitHub

- **Bước 4:** Khi mọi thứ đã sẵn sàng, mã nguồn sẽ được commit vào nhánh tương ứng với bài tập. Nếu quá trình này không thành công, hệ thống sẽ đưa ra các cảnh báo lỗi.

Khi nộp bài thành công, hệ thống sẽ tự động chuyển hướng về trang bài tập.



Hình 4.17. Trạng thái đang chấm bài

Đồng thời, GitHub Actions cũng tự động chạy các file workflow khi có commit mới

The screenshot shows the GitHub Actions interface for the repository 'kakanvk / hicommit-problems'. The 'Actions' tab is selected. On the left, there's a sidebar with 'All workflows' (CI, Management, Caches, Attestations, Runners). The main area shows 'All workflows' with '115 workflow runs'. Two specific runs are highlighted with red boxes: 'Submit code by kakanvk - Test Program' (CI #115) and 'Re-submit code at 22:58:04 12/10/2024 b...' (CI #114). Both runs show a green checkmark and the message 'Commit 643cf36 pushed by kakanvk'.

Bảng 4.25. Các lượt chạy của GitHub Actions

Tab tiếp theo là “Thảo luận”, cho phép tham gia, tạo các cuộc thảo luận về bài toán.

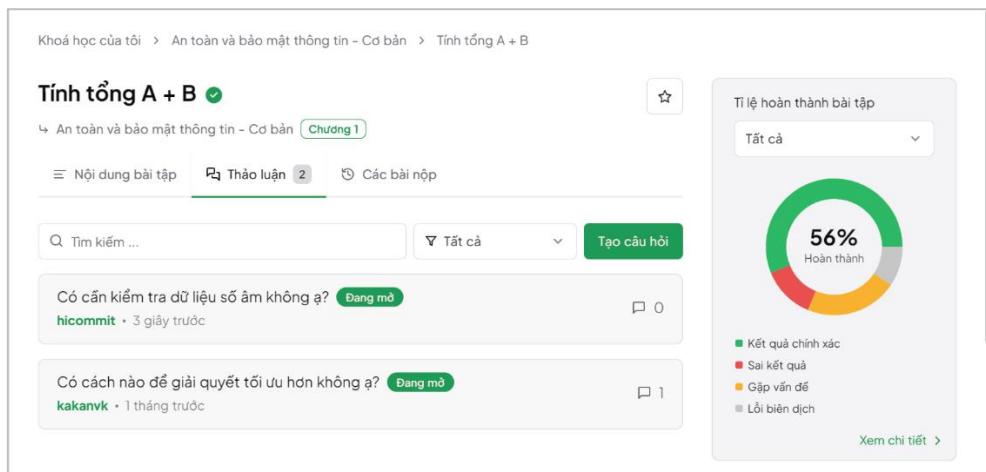
The screenshot shows the HiCommit platform interface for the course 'An toàn và bảo mật thông tin - Cơ bản' (Chapter 1). The current topic is 'Tính tổng A + B'. In the navigation bar, 'Thảo luận' is highlighted with a red box. Below it, there's a search bar, a dropdown for 'Tất cả', and a 'Tạo câu hỏi' button. A sidebar on the right displays a pie chart showing completion status: 56% (Hoàn thành), with categories: Kết quả chính xác (green), Sai kết quả (red), Gặp vấn đề (orange), and Lỗi biên dịch (grey). Below the chart is a link 'Xem chi tiết >'. At the bottom, there are 'Tags' buttons for 'Cơ Bản' and 'Tổng'.

Hình 4.18. Giao diện thực tế một bài tập trên HiCommit – Thảo luận

Người dùng có thể tạo câu hỏi thảo luận mới để giải quyết các vấn đề có liên quan đến bài tập, chỉ cần cung cấp tiêu đề và mô tả vấn đề đang gặp phải.

The screenshot shows the HiCommit platform interface for the course 'An toàn và bảo mật thông tin - Cơ bản' (Chapter 1). A modal window titled 'Tạo câu hỏi thảo luận' is open. It has a 'Tiêu đề' field containing 'Có cần kiểm tra dữ liệu số âm không?' and a larger 'Mô tả' field containing 'Để bài không nêu rõ về vấn đề số âm và số dương, vậy có cần kiểm tra vấn đề này không a?'. At the bottom of the modal are 'Đóng' and 'Đăng câu hỏi' buttons. The background shows the same course navigation and completion chart as in Figure 4.18.

Hình 4.19. Tạo cuộc thảo luận mới



Hình 4.20. Tạo cuộc thảo luận thành công

Những người dùng khác có thể tham gia vào cuộc thảo luận này, có thể bình luận và tương tác (thích) nếu như chúng hữu ích. Đồng thời, tác giả cũng có một số tùy chọn như đóng/mở cuộc thảo luận, chỉnh sửa, xoá cuộc thảo luận.

Hình 4.21. Chi tiết về cuộc thảo luận

Cuối cùng là tab “Các bài nộp”, đây là nơi hiển thị các lượt nộp bài tập này của người dùng, cùng với một số thông tin như thông điệp commit, người nộp, thời gian nộp, thời gian chấm bài, kết quả chấm bài,...

Tính tổng A + B ✓↳ An toàn và bảo mật thông tin - Cơ bản **Chương 1**

≡ Nội dung bài tập

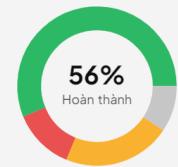
Thảo luận 2

☞ Các bài nộp



Tỉ lệ hoàn thành bài tập

Tất cả



- Kết quả chính xác
- Sai kết quả
- Gặp vấn đề
- Lỗi biên dịch

[Xem chi tiết >](#)[Taas:](#)*Hình 4.22. Giao diện thực tế một bài tập trên HiCommit – Các bài nộp*

Với mỗi bài nộp, thông tin chi tiết của lần nộp bài đó sẽ được hiển thị khi người dùng nhấp vào. Mỗi bài nộp có hai phần đánh giá: Kiểm tra đơn vị (Unit Test) và Đánh giá chất lượng mã nguồn (Code Quality).

Các bài tập > Kiểm tra số nguyên tố > Kết quả chấm bài

Kết quả chấm bài: Kiểm tra số nguyên tố ✓

☞ Test Review Code with Gemini

Commit bởi kakanvk • 28 ngày trước [Xem mã nguồn](#)

Kiểm tra đơn vị (Unit Test) Java

✓ Test case 1	Chính xác
✓ Test case 2	Chính xác
✓ Test case 3	Chính xác
✓ Test case 4	Chính xác
✓ Test case 5	Chính xác

Kết quả
✓ Kết quả chính xác

Kết quả chấm bài

100%

Tổng thời gian chấm: 23s

Unit Test • Chất lượng mã nguồn •

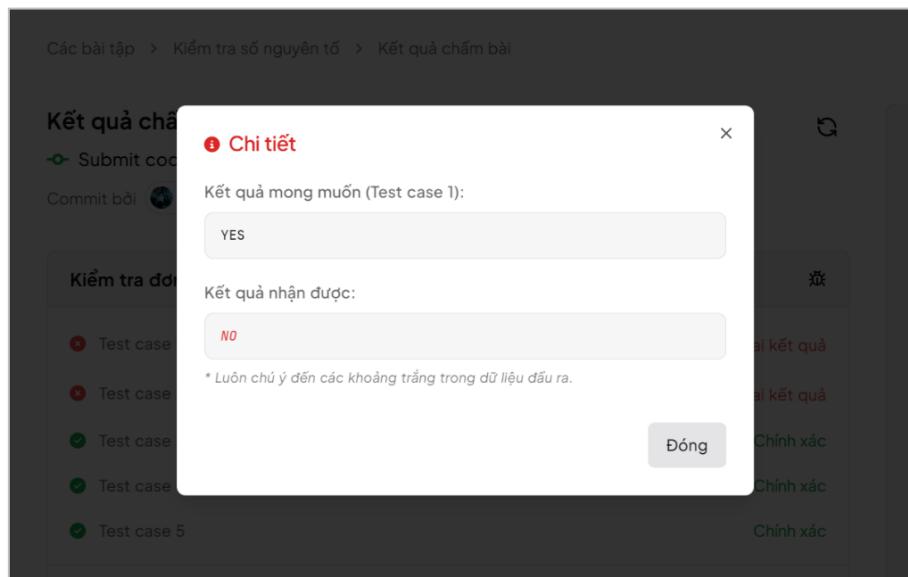
Kết quả chính xác

Hình 4.23. Kiểm tra đơn vị của một bài nộp – Trường hợp kết quả chính xác

Với phần Kiểm tra đơn vị, hệ thống sẽ đánh giá từng testcase của bài tập này và đưa ra phản hồi về các testcase đó. Nếu tất cả các testcase được thông qua, bài nộp được công nhận là có Kết quả chính xác. Ngược lại, nếu có bất kỳ sai sót nào, hệ thống sẽ đưa ra các cảnh báo.

Kiểm tra đơn vị (Unit Test)		C++	X
✓	Test case 1	Chính xác	
✓	Test case 2	Chính xác	
✓	Test case 3	Chính xác	
✗	Test case 4 	Sai kết quả	
✓	Test case 5	Chính xác	
Kết quả		Thời gian	19s
✗ Sai kết quả			

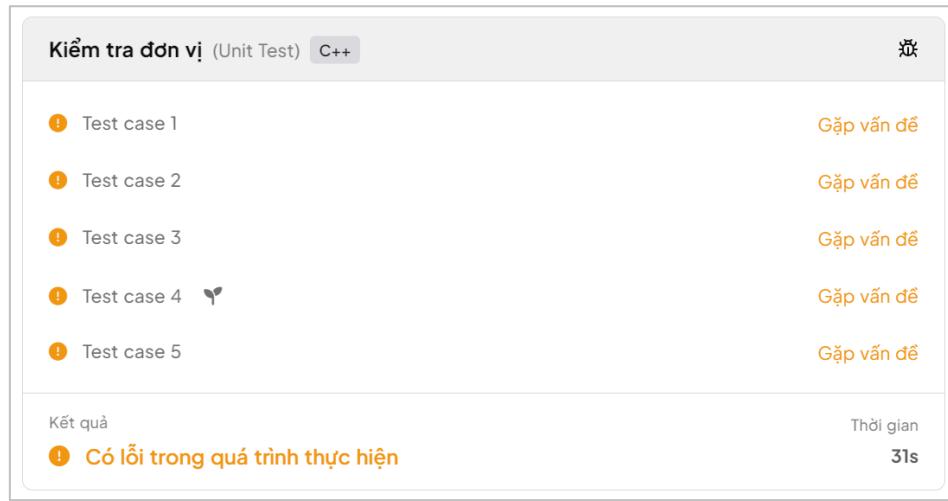
Hình 4.24. Trường hợp sai kết quả



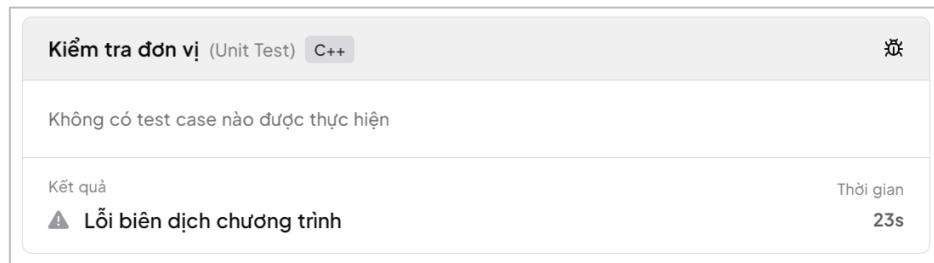
Hình 4.25. Chi tiết về lỗi của một testcase chưa hoàn thành



Hình 4.26. Gợi ý hoàn thành testcase

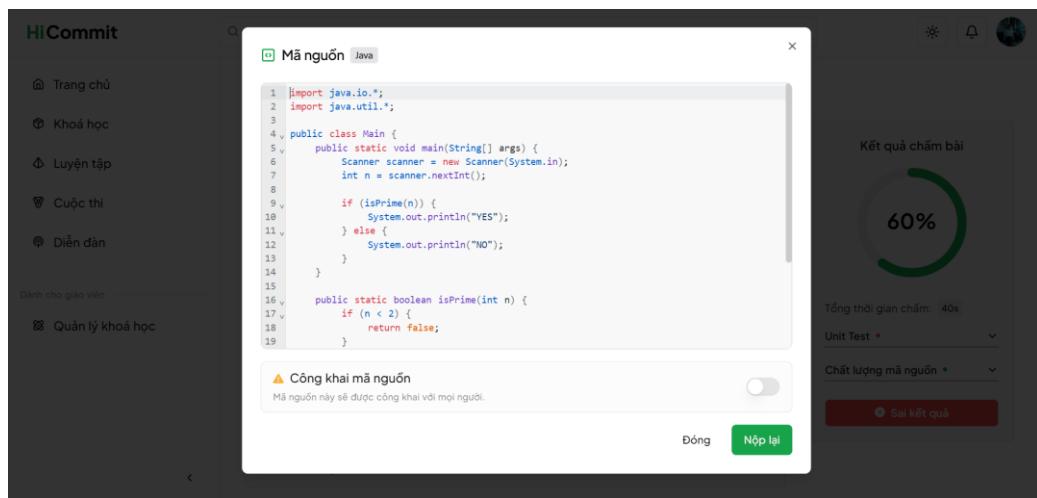


Hình 4.27. Trường hợp lỗi trong quá trình thực hiện

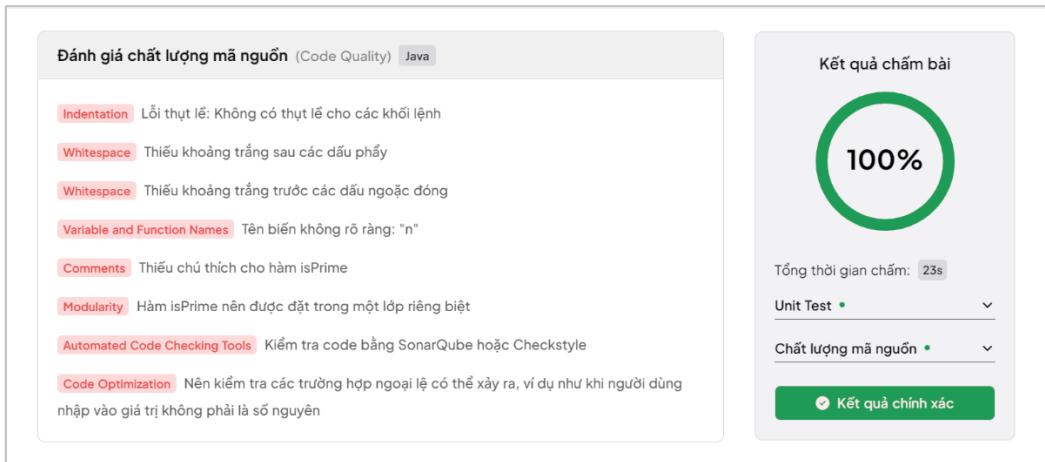


Hình 4.28. Trường hợp lỗi biên dịch chương trình

Người dùng có thể dễ dàng xem lại mã nguồn đã nộp và nộp lại nếu cần. Nút công khai mã nguồn cho phép những người dùng khác có thể xem được phần mã nguồn của lần nộp bài này (mặc định chức năng này được tắt đi).



Hình 4.29. Chức năng xem lại mã nguồn đã nộp



Hình 4.30. Đánh giá chất lượng mã nguồn của một bài nộp

Các gợi ý chỉnh sửa:

- Sử dụng thụt lề phù hợp để tăng tính dễ đọc cho code
- Thêm khoảng trắng sau các dấu phẩy và trước các dấu ngoặc đóng để tăng tính dễ đọc
- Đặt tên biến rõ ràng hơn
- Thêm chú thích cho hàm isPrime
- Phân tách hàm isPrime vào một lớp riêng biệt
- Sử dụng công cụ kiểm tra code tự động như SonarQube hoặc Checkstyle để phát hiện lỗi và cải thiện chất lượng code
- Kiểm tra các trường hợp ngoại lệ có thể xảy ra để code hoạt động ổn định hơn

Được thực hiện bởi GitHub Actions

Hình 4.31. Các gợi ý chỉnh sửa mã nguồn

Cuối mỗi bài chấm đều sẽ có phần “Được thực hiện bởi GitHub Actions”, người dùng có thể bấm vào đây để xem chi tiết quá trình chấm trên GitHub Actions.

build		Search logs
succeeded last month in 27s		↻ ⚙
>	✓ Set up job	0s
>	✓ Record start time	1s
>	✓ Checkout code	0s
>	✓ Send Pending Status to API	2s
>	✓ Set up Java and Utilities	17s
>	✓ Compile Code	1s
∅	Handle Compile Failure	0s
>	✓ Fetch Test Cases from API	2s
>	✓ Run Tests	0s
>	✓ Record end time and calculate duration	0s
>	✓ Post Test Results	2s
>	✓ Post Checkout code	0s
>	✓ Complete job	0s

Hình 4.32. Chi tiết quá trình chấm bài trên GitHub Actions

Các bài tập trên hệ thống HiCommit được chia làm ba dạng với cùng một cách thức hoạt động. Tuy nhiên, giao diện sẽ khác nhau ở một số điểm nhỏ để phân biệt.

Bài tập trong khoá học có một số liên kết neo đến khoá học chứa nó:

Khoa học của tôi > An toàn và bảo mật thông tin - Cơ bản > Tính tổng A + B

Tính tổng A + B ✓

An toàn và bảo mật thông tin - Cơ bản **Chương 1**

Nội dung bài tập Thảo luận 1 Các bài nộp

Mã bài tập: **SUM**

Mô tả bài tập: Tính tổng A + B

Input: 2 số nguyên A, B

Output: Tổng A + B

Tỉ lệ hoàn thành bài tập

56% Hoàn thành

- Kết quả chính xác
- Sai kết quả
- Gặp vấn đề
- Lỗi biên dịch

Xem chi tiết >

Hình 4.33. Giao diện thực tế chức năng bài tập trong khoá học

Bài tập tự do có liên kết neo đến mục luyện tập, các nhãn về độ khó và bài tập tự do:

Luyện tập > Tính A * B bản khó

Tính A * B bản khó ✓

Bài tập tự do Mức độ: Trung bình

Nội dung bài tập Thảo luận 0 Các bài nộp

Mã bài tập: AXB2

Mô tả đề bài: Tính A * B

Input:

- Dòng đầu tiên là số bộ test n
- n dòng tiếp theo, mỗi dòng có 2 số a, b cách nhau bởi khoảng trắng

Output:

In ra n dòng, mỗi dòng là tích a*b

Tỉ lệ hoàn thành bài tập

Tất cả

27% Hoàn thành

- Kết quả chính xác
- Sai kết quả
- Gặp vấn đề
- Lỗi biên dịch

Xem chi tiết >

Tags: Phép Nhân Vòng Lặp

Hình 4.34. Giao diện thực tế chức năng bài tập tự do

Bài tập trong cuộc thi có liên kết neo đến cuộc thi, chức năng thảo luận và thêm vào danh sách yêu thích được ẩn đi:

Các cuộc thi > HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024 > Tiền thừa của Hiệp

Tiền thừa của Hiệp

HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024

Nội dung bài tập Các bài nộp

Mã bài tập: tvu_olympic_2024_B

Mô tả đề bài:

Bạn Hiệp được mẹ đưa cho tờ 500.000đ để đến cửa hàng thực phẩm TVU mua một số mặt hàng. Cửa hàng thực phẩm TVU có các mệnh giá tiền: 1.000đ, 5.000đ, 10.000đ, 50.000đ và 100.000đ; và đơn giá của các mặt hàng đều chia hết cho 1.000. Hiệp mua thực phẩm với tổng tiền là N ($1.000 \leq N \leq 499.000$). Hãy giúp cửa hàng trả lại số tiền K còn lại của Hiệp sao cho với số tờ ít nhất.

Input:

Số nguyên dương N ($1.000 \leq N \leq 499.000$ và N chia hết cho 1.000) : là số tiền Hiệp đã mua.

Tỉ lệ hoàn thành bài tập

Chỉ mình tôi

Chưa có dữ liệu

Xem chi tiết >

Bài tập này nằm trong một cuộc thi

Hình 4.35. Giao diện thực tế chức năng bài tập trong cuộc thi

4.3.5. Chức năng Luyện tập

Chức năng này hiển thị danh sách các bài tập tự do trên hệ thống cùng với các thông tin của từng bài tập như tên, dạng bài, ngôn ngữ, tỉ lệ hoàn thành (AC), cấp độ, điểm,... Các tùy chọn khác nhằm giúp người dùng thao tác với dữ liệu như tìm kiếm, lọc theo dạng bài, các bài đã giải/chưa giải,... Với nút “Lấy ngẫu nhiên”, cho phép hệ thống chọn ra một bài ngẫu nhiên (chưa giải) và chuyển hướng người dùng đến bài tập đó.

The screenshot shows the HiCommit interface with the following elements:

- Main Area:** Displays a list of 7 exercises with columns for ID, Exercise Name, Language, Difficulty, AC Rate, Problem Type, and Points.
- Top Right:** A red box highlights the "Bảng xếp hạng" (Ranking Table) button, which leads to the ranking table shown in Figure 4.37.
- Sidebar:** Labeled "Các dạng bài tập" (Types of exercises) with a list of categories like Chuỗi, Palindrome, etc.
- Bottom Right:** A red box highlights the "Lấy ngẫu nhiên" (Randomize) button in the sidebar.

Hình 4.36. Giao diện thực tế chức năng Luyện tập

Khi hoàn thành các bài tập trong Luyện tập, điểm của bài tập đó sẽ được cộng dồn vào điểm tích luỹ của người dùng và được thống kê trên bảng xếp hạng. Chức năng Bảng xếp hạng được truy cập bằng cách vào mục “Bảng xếp hạng” trong chức năng Luyện tập.

The screenshot shows the ranking table for the practice section:

#	Tài khoản	Tỷ lệ AC	Số bài đã giải	Điểm
1	kakanvk (Bạn)	38.9%	4 bài	200
2	hicommitt	33.3%	2 bài	40
3	vchiepp	50.0%	1 bài	20
4	kakaintest	0.0%	0 bài	0
5	nguyenthuhakt	0.0%	0 bài	0
6	kakanie	0.0%	0 bài	0

Hình 4.37. Giao diện thực tế chức năng Bảng xếp hạng (Luyện tập)

Tại đây, danh sách các người dùng trên hệ thống được xếp hạng dựa vào điểm tích luỹ đạt được trong quá trình giải các bài tập (Luyện tập). Người dùng cũng có thể xem rõ thông tin từng người và thứ bậc của họ bằng cách bấm vào tên của người dùng tương ứng. Hệ thống sẽ chuyển hướng đến trang cá nhân của người dùng đó và hiển thị một số thông tin cơ bản như tên người dùng, điểm tích luỹ, thứ bậc trên Bảng xếp hạng, các bài tập đã giải,

các cuộc thi đã tham gia,...

The screenshot shows the HiCommit interface. On the left sidebar, there are navigation links: Trang chủ, Khoa học, Luyện tập, Cuộc thi, and Diễn đàn. Under 'Dành cho giáo viên', there is Quản lý khóa học. The main area shows a profile for 'kakanvk' with the email 'caoka5265@gmail.com'. A green banner at the top says '#T trên Bảng xếp hạng'. Below it, a box shows 'Điểm tích luỹ: 200' and 'Tỷ lệ hoàn thành: 38.9%'. A note says 'Đã tham gia 2 tháng trước'. Below this are tabs for 'Bài tập đã giải', 'Khoa học đã tham gia', 'Cuộc thi đã tham gia', and 'Bài viết'. Under 'Cuộc thi đã tham gia', there are four items: 'Trung bình Tính A * B bản khó', 'Trung bình Tính số Fibonacci thứ n', 'Để Kiểm tra số nguyên tố', and 'Khô Thu hoạch vụ mùa (Bản khó)'. A search bar at the top right says 'Tim kiếm khoá học, bài tập...'.

Hình 4.38. Giao diện thực tế chức năng Trang cá nhân

4.3.6. Chức năng Cuộc thi

Giao diện này hiển thị danh sách các cuộc thi sắp diễn ra, đang diễn ra hoặc đã kết thúc trên hệ thống. Với các cuộc thi công khai, người dùng có thể tham gia một cách thoải mái.

The screenshot shows the 'Contests' section of the HiCommit interface. On the left sidebar, 'Cuộc thi' is selected. The main area shows a contest titled 'Olympic Tin học sinh viên Việt Nam 2024' with a green banner saying 'Đang diễn ra'. It includes details: 'Bắt đầu từ 21:30 11.10.2024' and 'đến 00:30 12.10.2024'. Below it is another contest card for 'HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024' with a green banner 'Đang diễn ra'. It shows 'Được tạo bởi kakanvk • 1 giờ trước' and '2 người tham gia'. A button 'Tiếp tục làm bài >' is present. Below these is a concluded contest card for 'ICPC TEST OFFICIAL 2024 TVU' with a red banner 'Đã kết thúc'. It shows 'Bắt đầu từ 18:00 23.08.2024' and 'đến 02:00 24.08.2024'. It includes 'Được tạo bởi kakanvk • 1 tháng trước' and '1 người tham gia'. A button 'Xem kết quả >' is shown. A search bar at the top right says 'Tim kiếm cuộc thi' and a filter 'Tất cả' is visible.

Hình 4.39. Giao diện thực tế chức năng Cuộc thi

Với các cuộc thi sắp diễn ra, thời gian sẽ đếm ngược trước đến bắt đầu cuộc thi. Trong khoảng thời gian này, mọi thông tin về đề bài và bảng xếp hạng sẽ bị khoá. Mỗi người dùng chỉ được phép tham gia một cuộc thi vào một thời điểm nhất định.

HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024

☰ Chi tiết ⚑ Bảng xếp hạng ⏱ Các bài nộp

[Replay] HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024

Địa điểm: Toà C7, Trường Đại học Trà Vinh

Thời gian: 13h, 10.10.2024

Đề bài:

Bắt đầu sau

00:06:22

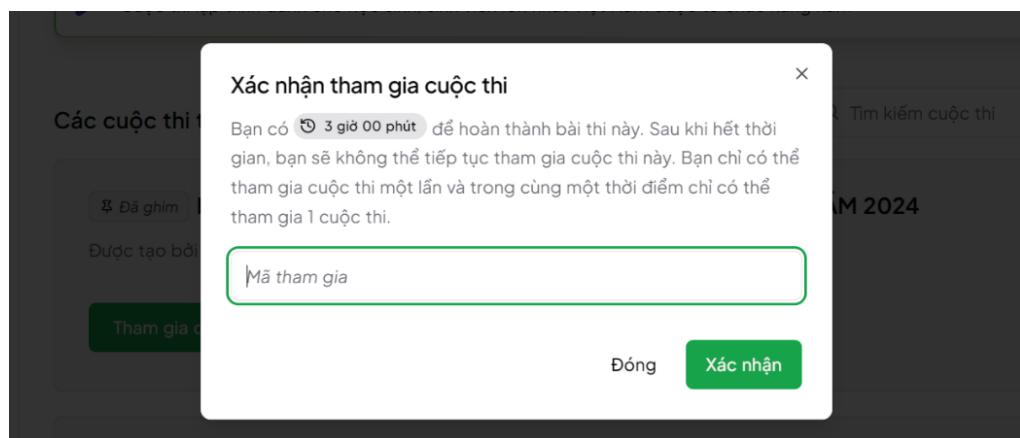
Diễn ra từ 10:15 13.10.2024

Kết thúc 13:15 13.10.2024

Chỉ hiển thị trong thời gian diễn ra cuộc thi

Hình 4.40. Giao diện thực tế trước khi diễn ra cuộc thi

Với các cuộc thi riêng tư, người dùng cần cung cấp mật khẩu tham gia để tham gia. Nếu là một cuộc thi công khai, người tham gia không cần phải cung cấp mật khẩu tham gia nhưng cũng cần xác nhận tham gia cuộc thi.



Hình 4.41. Xác nhận tham gia cuộc thi

Khi tham gia thành công, đề bài sẽ được mở ra để người tham gia có thể hoàn thành. Thời gian làm bài của cuộc thi sẽ đếm ngược đến khi hết giờ làm bài. Trong khoảng thời gian này, đề bài sẽ được mở ra và cho phép người tham gia cuộc thi được phép nộp các bài nộp lên hệ thống. Người tham gia được quyền rời khỏi cuộc thi nhưng sẽ không thể tham gia lại cuộc thi đó.

Tab “Chi tiết” chứa các thông tin của cuộc thi như mô tả, đề bài, ngôn ngữ, điểm,...

Hình 4.42. Giao diện thực tế trong thời gian làm bài của cuộc thi

Bảng xếp hạng của cuộc thi là nơi cập liên tục quá trình làm bài của các thí sinh tham gia. Thứ hạng của các thí sinh được xếp dựa theo kết quả làm bài của các thí sinh.

Hạng	Tài khoản	A (1 lần)	B (1 lần)	C (1 lần)	D (1 lần)	Điểm
1	kakaintest (Bản)		40	40		80
2	kakanvk		40 (2 lần)		0 (1 lần)	80
3	hicommitt			40 (1 lần)	0 (1 lần)	40
4	nguyenthuhakt					0
5	kakanie					0

Hình 4.43. Giao diện thực tế Bảng xếp hạng cuộc thi

Tab “Các bài nộp” hiển thị danh sách toàn bộ bài nộp trong cuộc thi của tất cả các thí sinh tham gia. Người dùng có thể dễ dàng theo dõi để đưa ra các chiến thuật phù hợp. Chức năng này cung cấp tùy chọn để có thể lọc ra các lượt nộp bài của từng người tham gia.

HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024

☰ Chi tiết ⚡ Bảng xếp hạng 🗂 Các bài nộp

Tất cả

Tất cả

kakaintest (*Bạn*)

kakanie

kakanvk

hicommit

nguyenthuhukt

Độc quyền ✅

kakaintest • 14 phút trước

Thời gian

21s

Độc quyền ❌

kanvk • 18 giờ trước

Thời gian

23s

Tiền thừa của Hiệp ✅

Commit bởi kakaintest • 18 giờ trước

Thời gian

22s

Thời gian còn lại

02:39:11

⌚ 03 giờ 00 phút

Rời khỏi cuộc thi →

Hình 4.44. Giao diện thực tế các bài nộp trong cuộc thi

Khi hết thời gian làm bài, đề bài sẽ được khoá lại, phía bên phải màn hình hiển thị và vinh danh ba người có thành tích tốt nhất cuộc thi.

The screenshot shows the HiCommit interface after the competition has ended. On the left, there's a sidebar with navigation links: Trang chủ, Khoa học, Luyện tập, **Cuộc thi**, and Diễn đàn. Below that are sections for Dành cho giáo viên and Quản lý khoá học. The main content area displays the competition details: "Cuộc thi này đã kết thúc" (The competition has ended), "HỘI THI OLYMPIC TIN HỌC TRƯỜNG ĐẠI HỌC TRÀ VINH NĂM 2024", and the replay information: "Địa điểm: Toà C7, Trường Đại học Trà Vinh" and "Thời gian: 13h, 10.10.2024". It also shows the competition results table:

Rank	Avatar	Tên	Điểm
1		kakaintest	120
2		kakanvk	80
3		hicommitt	40

A button "Xem tất cả" (View all) is located next to the results table. A note at the bottom says "Chỉ hiển thị trong thời gian diễn ra cuộc thi" (Only displayed during the competition period).

Hình 4.45. Giao diện thực tế sau khi cuộc thi kết thúc

4.3.7. Chức năng Diễn đàn

Giao diện này hiển thị các bài viết đã được công bố trên hệ thống, với đa dạng các chủ đề. Người dùng có thể dễ dàng tìm kiếm, lọc,...

Hình 4.46. Giao diện thực tế chức năng Diễn đàn

Có thẻ xem chi tiết thông tin của một bài viết khi nhấn vào bài viết tương ứng.

Hình 4.47. Giao diện khi xem một bài viết

Người dùng cũng có thể đăng tải bài viết mới lên diễn đàn thông qua nút thêm.

Hình 4.48. Nút đăng tải bài viết mới lên diễn đàn

Cần hoàn tất các thông tin cần thiết để tạo bài viết mới như: tiêu đề, đường dẫn, tags, mô tả bài viết, nội dung bài viết, ảnh bìa bài viết.

The screenshot shows the HiCommit interface for creating a new article. On the left sidebar, 'Diễn đàn' is highlighted. The main area is titled 'Tạo bài viết'. It contains two main sections: '1. Thông tin cơ bản' and '2. Thông tin chi tiết'. In '1. Thông tin cơ bản', there are fields for 'Tiêu đề bài viết' (with placeholder 'Nhập tiêu đề bài viết') and 'Tùy chỉnh đường dẫn (URL)' (with placeholder 'Nhập đường dẫn tùy chỉnh'). Below these is a 'Tag' field with placeholder 'Nhập tag mới, tối đa 5 tags' and a '+' button. In '2. Thông tin chi tiết', there is a 'Mô tả bài viết' field with placeholder 'Nhập mô tả bài viết'. At the top right of the main area are search and filter icons.

Hình 4.49. Tạo bài viết mới đăng tải lên diễn đàn

Bài viết sau khi đăng thành công sẽ được đợi duyệt trước khi hiển thị lên diễn đàn.

4.4. Kết quả thực nghiệm với vai trò giảng viên

Với vai trò giảng viên, giảng viên có thể tạo khoá học, quản lý các khoá học do mình tạo ra, tuỳ chỉnh, công bố/ẩn khoá học, xoá khoá học,...

The screenshot shows the 'Quản lý khoá học' section of the HiCommit platform. On the left sidebar, 'Quản lý khoá học' is selected. The main area displays three course cards. The first card is for 'DA20TTB | Kỹ thuật lập trình', showing a thumbnail of a city skyline at night, a description 'Thông tin chung: Giảng viên phụ trách học phần:...', 13 assignments, and a status 'Đã công bố' (Published). The second card is for 'DA20TTB | An toàn và bảo mật thông tin', showing a thumbnail of a computer setup, a description 'An toàn và Bảo mật Thông tin là một môn học chuyên sâu tập trung vào các phương pháp và kỹ thuật bảo vệ thông tin và hệ thống thông tin khỏi các mối đe dọa, xâm nhập trái phép và các nguy cơ khác. Môn học này cung cấp kiến thức cơ bản đến nâng cao về an...', 9 assignments, and a status 'Đã công bố'. The third card is for 'DA20TTB | Kỹ thuật lập trình', showing a thumbnail of a plane, a description 'Sau khi học khóa học này, sinh viên có thể nắm vững các kiến thức cơ bản về C, C++...', and a status 'Công khai' (Public).

Hình 4.50. Giao diện thực tế chức năng Quản lý khoá học

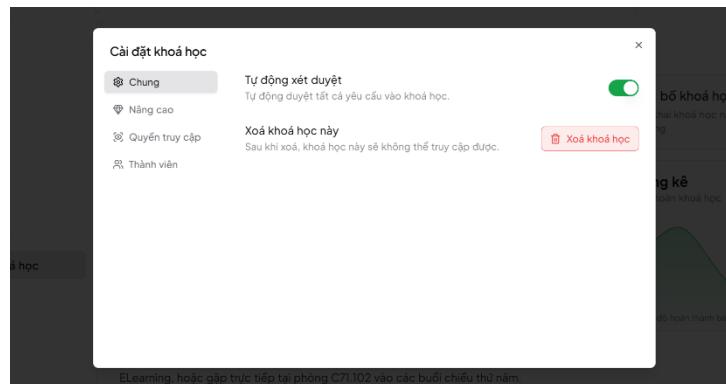
Các thông tin cần thiết khi tạo một khoá học mới bao gồm: tên khoá học, mô tả khoá học, mã lớp, đường dẫn tuỳ chỉnh, mật khẩu tham gia (nếu có), ảnh bìa.

Hình 4.51. Giao diện thực tế chức năng tạo khóa học mới

Khi nhấp vào từng khóa học, giao diện quản lý khóa học đó sẽ hiện ra với một số tùy chỉnh như chỉnh sửa thông tin, cài đặt khóa học, công bố, thống kê,...

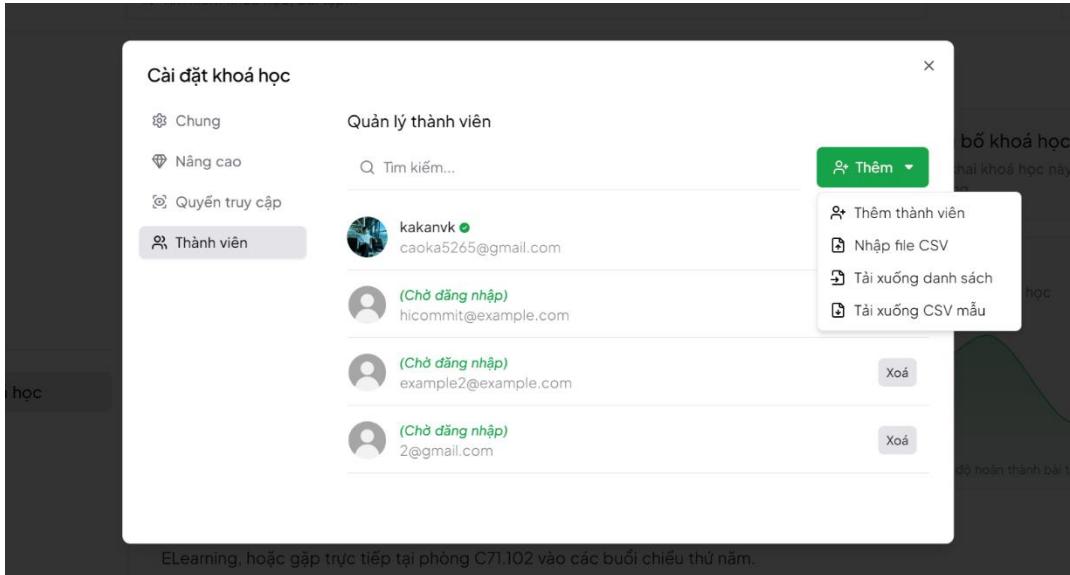
Hình 4.52. Giao diện thực tế quản lý một khóa học

Một số cài đặt được hỗ trợ như: tự động xét duyệt, xoá khóa học, quyền truy cập khóa học, liên kết tham gia,...



Hình 4.53. Cài đặt khoá học

Chức năng quản lý thành viên giúp quản lý các thành viên trong khoá học. Giảng viên có thể thêm thành viên bằng giao diện hoặc thêm bằng file CSV theo mẫu có sẵn. Chức năng cũng hỗ trợ xuất ra danh sách các thành viên trong khoá học.



Hình 4.54. Quản lý thành viên trong khoá học

Bên dưới là khu vực quản lý các chương, bài tập trong khoá học. Giảng viên có thể thêm, chỉnh sửa, xoá các chương, bài tập, di chuyển vị trí của các chương, bài tập.

Bài tập thực hành:

- 🕒 **Chương 1: Tổng quan về lập trình trên máy tính**
 - ⋮ Bài tập 1.1: Hello World LO1
 - ⋮ Bài tập 2.1: Tính diện tích hình chữ nhật LO2 CDR2
 - ⋮ Bài tập 1.2: Tính tổng 2 số nguyên LO1

🕒 Chương 2: Ngôn ngữ lập trình C

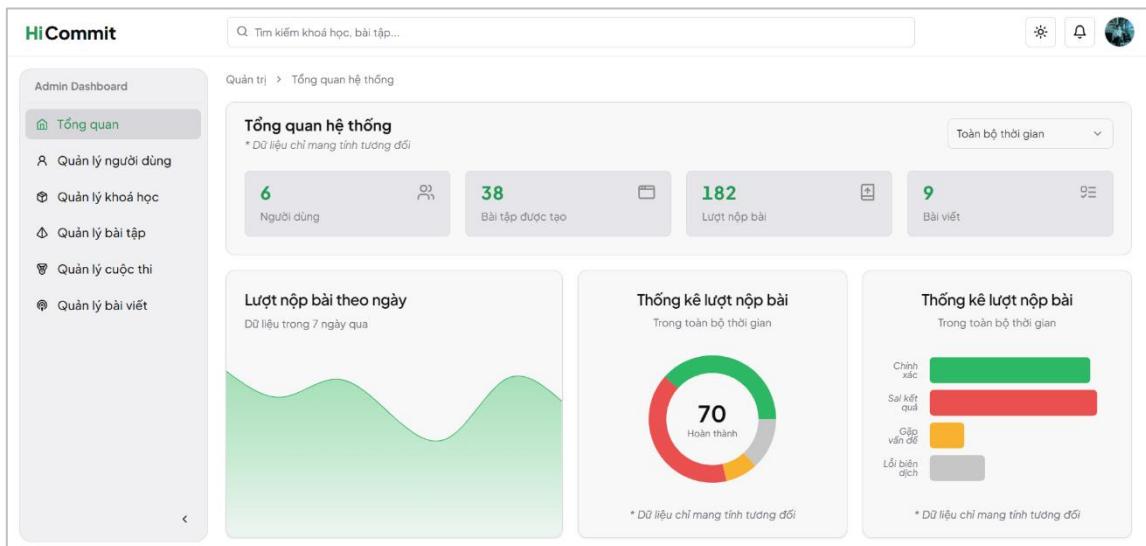
- ⋮ Bài tập 2.2: Tìm số lớn nhất LO2
- ⋮ Bài tập 2.3: Tính tổng các số lẻ LO1

Hình 4.55. Quản lý các chương, bài tập trong khoá học

4.5. Kết quả thực nghiệm với vai trò quản trị viên (Admin)

4.5.1. Chức năng Tổng quan

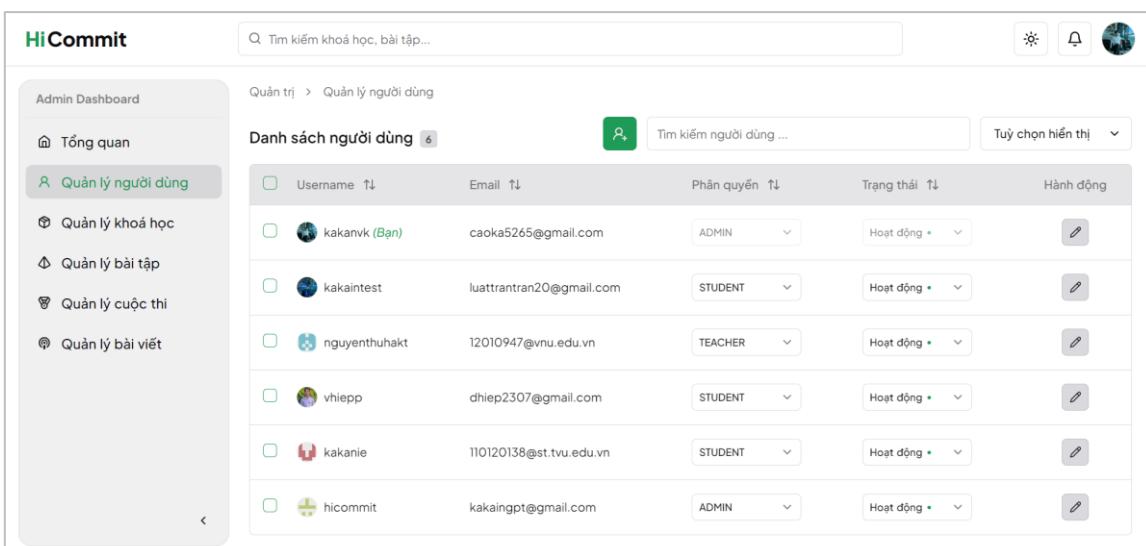
Giao diện này hiển thị một số thông tin tổng quan như số lượt nộp bài, số người dùng, số bài viết,... và các biểu đồ trực quan hóa các thông tin tổng quan.



Hình 4.56. Giao diện thực tế chức năng Quản lý tổng quan

4.5.2. Chức năng Quản lý người dùng

Chức năng cho phép quản lý toàn bộ người dùng với các thông tin như username, email, phân quyền, trạng thái,... Người quản trị có thể phân quyền trực tiếp vai trò của từng người dùng trong hệ thống ở giao diện này.



Hình 4.57. Giao diện thực tế chức năng Quản lý người dùng

4.5.3. Chức năng Quản lý khoá học

Chức năng này quản lý toàn bộ các khoá học trên hệ thống với các thông tin như tên khoá học, tác giả, bài tập, ngày tạo,... Với các chức năng quản lý tương tự như ở vai trò giảng viên. Tuy nhiên, quản trị viên có thể quản lý toàn bộ các khoá học trên hệ thống, thay vì các khoá học đã tạo như giảng viên.

Khoa học	Tác giả	Bài tập	Tham gia	Công bố	Ngày tạo
DAZTITA Kỹ thuật lập trình	kakanvk	7 bài tập	0	Đang chờ	Được tạo 1 tuần trước
DAZTITA An toàn và bảo mật thông...	kakanvk	3 bài tập	4	Đang chờ	Được tạo 2 tháng trước
DAZTITA Lập trình hướng đối tượng	hicommitt	0 bài tập	0	Đang chờ	Được tạo 2 tháng trước
DAZTITA Kỹ thuật lập trình	kakanvk	3 bài tập	1	Đang chờ	Được tạo 2 tháng trước

Hình 4.58. Giao diện thực tế chức năng Quản lý khoá học

4.5.4. Chức năng Quản lý bài tập

Chức năng này quản lý các bài tập tự do trên hệ thống. Quản trị viên có thể thêm mới, tùy chỉnh thông tin các bài tập, điều chỉnh cấp độ của bài tập, tìm kiếm, xoá bài tập,...

Mã bài tập	Tên bài tập	Dạng bài	Ngôn ngữ	Cấp độ	Tỷ lệ AC	Điểm
FIBONACCI_NUMBER	Tính số Fibonacci thứ n	Fibonacci	C	Trung bình	50%	40
PALINDROME_CHECK	Chuỗi đối xứng	Chuỗi	C++	Khó	0%	100
SUM_OF_DIGITS	Tổng các chữ số	Tổng	Chuỗi	Trung bình	0%	40
PRIME2	Kiểm tra số nguyên tố	Số nguyên tố	Java	Đá	60%	20
THUHOACH2	Thu hoạch vụ mùa (Bản...	Quy hoạch động	C++	Khó	50%	100
AXB2	Tính A * B bằn khô	Phép nhân	Vòng lặp	C++	Trung bình	27%

Hình 4.59. Giao diện thực tế chức năng Quản lý bài tập

Để tạo bài tập mới, cần cung cấp thông tin cơ bản của bài tập như: tên bài tập, mã bài

tập, ngôn ngữ lập trình, cấp độ của bài tập, điểm tuỳ chỉnh (mặc định theo cấp độ bài tập), các tag của bài tập.

The screenshot shows the 'Admin Dashboard' with a sidebar containing links like 'Tổng quan', 'Quản lý người dùng', 'Quản lý khóa học', 'Quản lý bài tập' (highlighted in green), 'Quản lý cuộc thi', and 'Quản lý bài viết'. The main content area is titled '1. Thông tin cơ bản *'. It includes fields for 'Tên bài tập *' (Input: Nhập tên bài tập), 'Mã bài tập *' (Input: Nhập đường dẫn tuỳ chỉnh), 'Ngôn ngữ lập trình *' (Select: C), 'Cấp độ *' (Select: Dễ), 'Điểm *' (Input: 20), 'Tag *' (Input: Nhập tag mới, tối đa 5 tags), and a 'Hicommit' button.

Hình 4.60. Tạo bài tập mới (Thông tin cơ bản)

Tiếp theo, cần cung cấp các thông tin chi tiết như: mô tả đề bài, input, output, giới hạn

The screenshot shows the 'Admin Dashboard' sidebar with the 'Quản lý bài tập' link highlighted. The main content area is titled '2. Thông tin chi tiết *'. It includes fields for 'Mô tả đề bài *' (Input: Nhập mô tả bài toán), 'Input *' (Input: Nhập mô tả cho dữ liệu đầu vào (input)), 'Output *' (Input: Nhập mô tả cho dữ liệu đầu ra (output)), and 'Giới hạn' (Input: Nhập các giới hạn về tài nguyên, thời gian (nếu có)).

Hình 4.61. Tạo bài tập mới (Thông tin chi tiết)

Cuối cùng, cần cung cấp các ví dụ với các testcase cho bài tập. Các ví dụ sẽ hiển thị cùng với đề bài, cho phép người dùng hình dung rõ ràng hơn về dữ liệu đầu vào và đầu ra của bài toán. Ngược lại, các testcase sẽ được ẩn với người dùng, được sử dụng trong việc kiểm tra đơn vị (Unit Test).

Cả ví dụ và testcase đều có mục tạo ghi chú để giải thích rõ hơn về các ví dụ hoặc testcase. Với testcase, gợi ý chỉnh sửa cho phép người dùng xem được gợi ý của testcase đó (nếu lỗi), để cải thiện mã nguồn.

The screenshot shows the Admin Dashboard with the 'Quản lý bài tập' (Test Management) section selected. Under '3. Các ví dụ', there is a 'Ví dụ 1' card with input '1' and output '2'. Below it is a note 'Ghi chú cho Ví dụ' with 'Ghi chú' and a '+ Thêm ví dụ' button. Under '4. Các Test-case', there is a 'Test-case 1' card with input '3' and output '4'. Below it is a '+ Thêm Test-case' button. At the bottom right is a green 'Tạo bài tập' button.

Hình 4.62. Tạo bài tập mới (Các ví dụ và các Testcase)

Khi tạo bài tập thành công, quản trị viên có thể tùy ý thay đổi các thông tin của bài tập này (nếu cần). Tuy nhiên, ngôn ngữ lập trình sẽ chỉ có thể thay đổi nếu chưa có lượt nộp bài nào liên quan đến bài toán này.

The screenshot shows the Admin Dashboard with the 'Quản lý bài tập' section selected. It displays the 'Chỉnh sửa bài tập' (Edit Test Case) form for a test case named 'Tính số Fibonacci thứ n' with ID 'FIBONACCI_NUMBER'. The form includes fields for 'Tên bài tập' (Title), 'Mã bài tập' (Code), 'Ngôn ngữ lập trình' (Programming Language) set to 'C', 'Cấp độ' (Difficulty) set to 'Trung bình' (Medium), 'Điểm' (Score) set to '40', and a 'Tag' field containing 'Fibonacci'. A note at the bottom states: '* Chỉ có thể thay đổi ngôn ngữ nếu chưa có người nộp bài' and 'Mặc định (Để: 20, Trung bình: 40, Khó: 100)'.

Hình 4.63. Chức năng chỉnh sửa bài tập

4.5.5. Chức năng Quản lý cuộc thi

Chức năng cho phép quản lý các cuộc thi trên hệ thống, người quản trị có thể thực hiện các tác vụ như tạo cuộc thi mới, ghim/bỏ ghim cuộc thi, xem chi tiết, chỉnh sửa cuộc thi, xoá cuộc thi.

Tên cuộc thi	Thời gian	Thời lượng	Tham gia	Trạng thái	Công bố
WARM-UP [Mirror] Code Tour 2024 - Warm-up... Đã kết thúc	Bắt đầu 10.10.2024 · 10:30 Kết thúc 10.10.2024 · 16:30	Làm bài trong 6 giờ	AI 5	Riêng tư	<input checked="" type="checkbox"/>
ICPC-TEST-TVU ICPC TEST OFFICIAL 2024 TVU Đã kết thúc	Bắt đầu 23.08.2024 · 18:00 Kết thúc 24.08.2024 · 02:00	Làm bài trong 8 giờ	AI 1	Riêng tư	<input checked="" type="checkbox"/>
KCODING-TOUR-2024 KCODING TOUR 2024 Đã kết thúc	Bắt đầu 21.08.2024 · 22:30 Kết thúc 22.08.2024 · 01:30	Làm bài trong 3 giờ	AI 0	Công khai	<input checked="" type="checkbox"/>
HICOMMIT-TOUR HICOMMIT ICPC Regional 2024	Bắt đầu 21.08.2024 · 03:00	Làm bài trong	AI 0	Riêng tư	<input checked="" type="checkbox"/>

Hình 4.64. Giao diện thực tế chức năng Quản lý cuộc thi

Người quản trị có thể tạo cuộc thi mới với các thông tin cần cung cấp như: tên cuộc thi, mã cuộc thi, mô tả, thời gian bắt đầu, thời gian làm bài, mật khẩu tham gia (nếu có).

Tên cuộc thi *
Nhập tiêu đề bài viết

Mã cuộc thi *
Nhập mã cuộc thi

Mô tả cuộc thi *
Nhập nội dung bài viết

2. Thông tin chi tiết *

Thời gian bắt đầu *
14 tháng 10, 2024 00:00

Thời gian làm bài *
02 giờ 00 phút

Hình 4.65. Giao diện chức năng tạo cuộc thi mới

Trong chi tiết mỗi cuộc thi, quản trị viên có thể chỉnh sửa các thông tin cuộc thi, thêm, chỉnh sửa hoặc xoá các bài tập trong đề bài. Với chức năng tạo bài tập trong cuộc thi tương tự như tạo trong khoá học hay bài tập tự do.

Hình 4.66. Giao diện chi tiết quản lý một cuộc thi

Khi nhấp vào nút chính sửa, hệ thống sẽ chuyển hướng sang chỉnh sửa cuộc thi.

Hình 4.67. Giao diện chỉnh sửa một cuộc thi

Ngoài ra, người quản trị cũng có thể quản lý danh sách tham gia và bảng xếp hạng.

Hình 4.68. Quản lý danh sách tham gia cuộc thi

4.5.6. Chức năng Quản lý bài viết

Chức năng này được chia làm hai phần, các bài viết cần duyệt và đã duyệt. Ở tab “Đã duyệt”, liệt kê danh sách các bài viết đã được duyệt, hỗ trợ các chức năng như tìm kiếm, công bố, chỉnh sửa, xem trước, xoá bài viết,...

Chức năng tạo bài viết tương tự như đăng tải bài viết của người dùng trong diễn đàn.

Bài viết	Tác giả	Tags	Công bố	Ngày tạo	Hành động
Quy hoạch động cơ bản (Phần 1)	kakanvk	Quy hoạch động (QHD) (Dynamic Programming)	<input checked="" type="checkbox"/>	Được tạo 11 giờ trước	
Mảng và danh sách liên kết	kakanvk	Con trỏ, Mảng, Danh sách liên kết	<input checked="" type="checkbox"/>	Được tạo 11 giờ trước	
Tâm Quan Trọng Của Thuật Toán	kakanvk	Algorithm	<input checked="" type="checkbox"/>	Được tạo 12 giờ trước	
Deploy NestJS with TypeScript on...	kakanvk	NestJS, TypeScript	<input checked="" type="checkbox"/>	Được tạo 1 tháng trước	

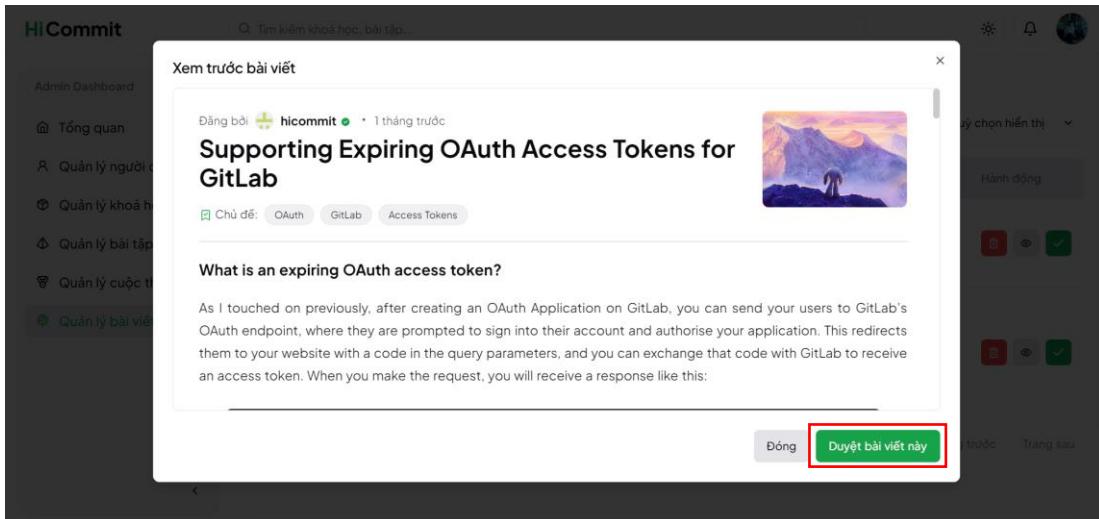
Hình 4.69. Giao diện thực tế chức năng Quản lý bài viết (Đã duyệt)

Ở tab “Đang chờ duyệt”, hiển thị danh sách bài viết được người dùng đăng tải, cần được kiểm tra trước khi công bố lên diễn đàn của hệ thống. Các chức năng được hỗ trợ như xem trước bài viết, xoá bài viết, duyệt bài viết.

Bài viết	Tác giả	Tags	Ngày tạo	Hành động
Supporting Expiring OAuth Access Tokens for GitLab	hicommit	OAuth, GitLab, Access Tokens	Được tạo 1 tháng trước	
How to Build a Scalable Software Architecture Part 1: Monolith vs....	kakanvk	Software, Architecture, Monolith, Microservices	Được tạo 2 tháng trước	

Hình 4.70. Giao diện thực tế chức năng Quản lý bài viết (Đang chờ duyệt)

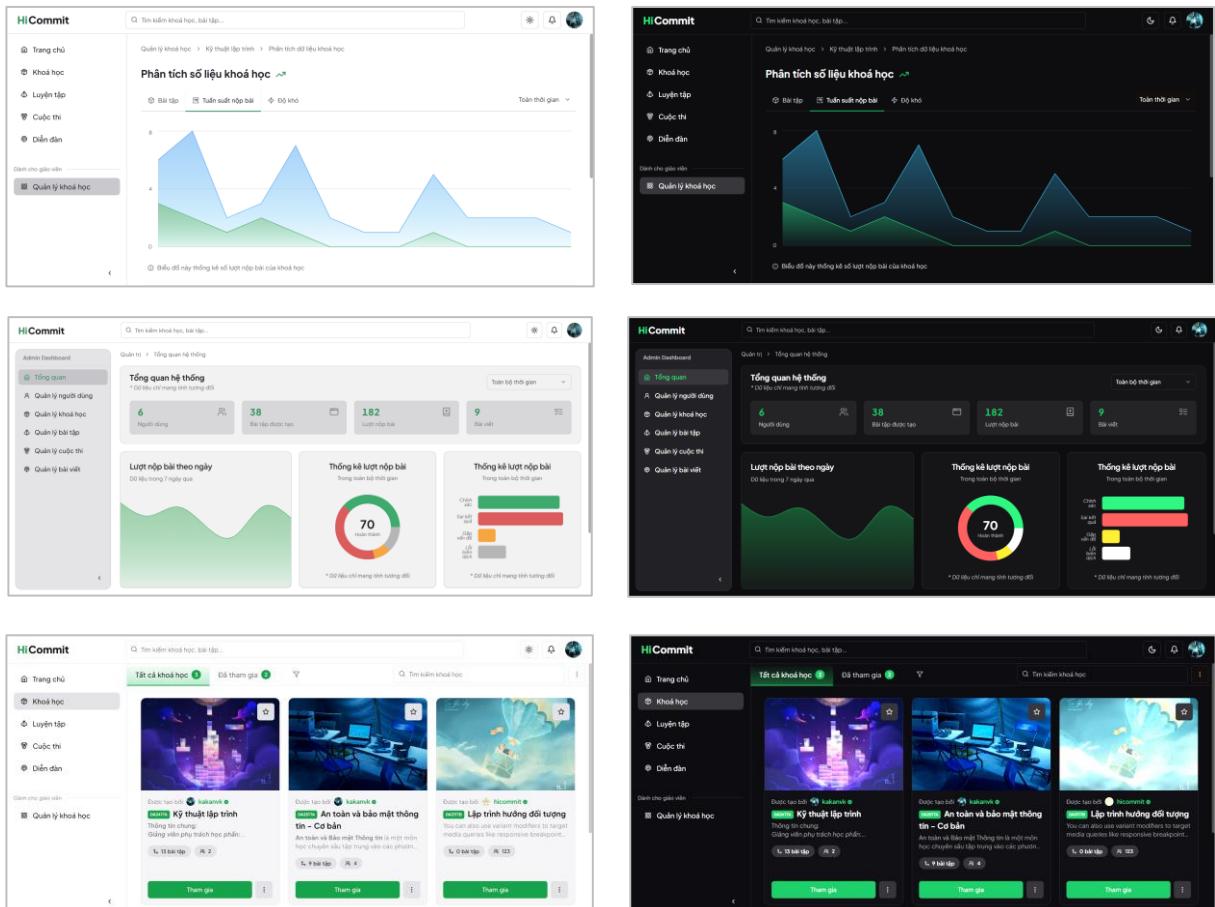
Khi xem trước bài viết, quản trị viên có thể trực tiếp duyệt bài viết cần thiết.



Hình 4.71. Giao diện xem trước bài viết

4.6. Các chức năng giao diện khác

Chế độ tối được hỗ trợ cho toàn bộ các giao diện trên hệ thống giúp phù hợp hơn trong điều kiện làm việc ở môi trường ánh sáng yếu.



Hình 4.72. Chế độ tối trên hệ thống HiCommit

4.7. Kết quả thực nghiệm ứng dụng HiCommit CLI

HiCommit CLI là một ứng dụng với giao diện dòng lệnh được xây dựng nhằm hỗ trợ kết nối giữa HiCommit và Git trên môi trường cục bộ (Local). Các chức năng chính mà HiCommit CLI có thể hỗ trợ:

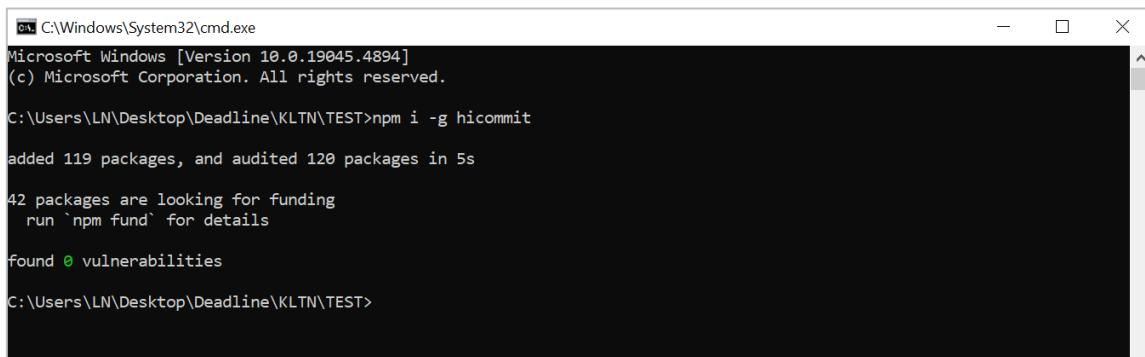
- Đăng nhập GitHub
- Hỗ trợ các khôi tạo cần thiết
- Hỗ trợ khôi tạo cho các bài tập trên hệ thống
- Một số chức năng khác

Với việc sử dụng NodeJS làm ngôn ngữ phát triển, các ứng dụng CLI đơn giản có thể được phát triển một cách nhanh chóng với nhiều thư viện hỗ trợ.

Ứng dụng có thể cài đặt trực tiếp thông qua trình quản lý gói **npm** của NodeJS:

```
npm i -g hicommitt
```

Lựa chọn **-g** cho phép ứng dụng được cài đặt ở mức toàn cục (global), vì thế có thể sử dụng ở mọi nơi trên máy tính.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LN\Desktop\Deadline\KLTN\TEST>npm i -g hicommitt
added 119 packages, and audited 120 packages in 5s

42 packages are looking for funding
  run `npm fund` for details

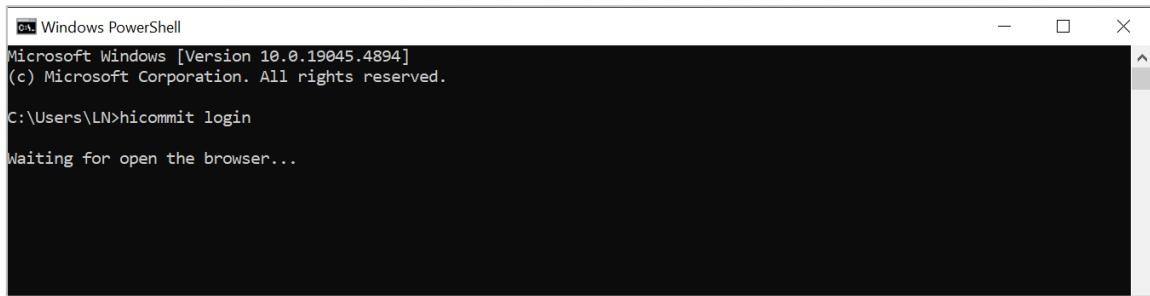
found 0 vulnerabilities

C:\Users\LN\Desktop\Deadline\KLTN\TEST>
```

Hình 4.73. Quá trình cài đặt HiCommit CLI

Sau khi cài đặt thành công, có thể đăng nhập tài khoản GitHub với câu lệnh:

```
hicommitt login
```

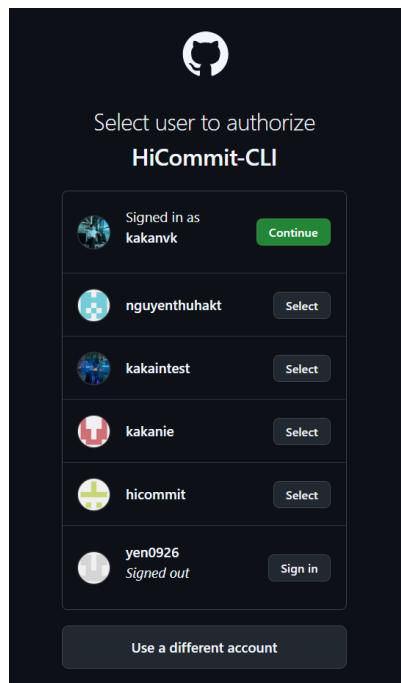


```
C:\Windows\system32\cmd.exe
Windows PowerShell [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LN>hicommmit login
Waiting for open the browser...
```

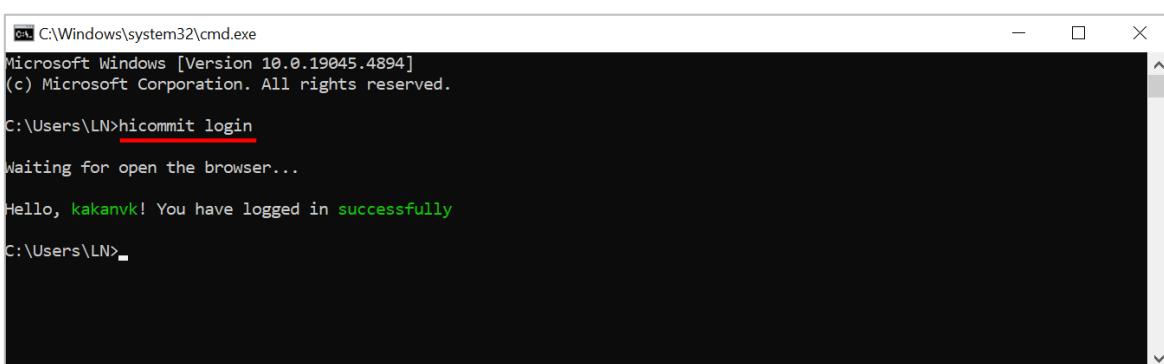
Hình 4.74. Đăng nhập với HiCommit CLI

Sau khi lệnh được thực thi, một cửa sổ trình duyệt web bật lên cho phép người dùng lựa chọn tài khoản muốn đăng nhập vào HiCommit CLI.



Hình 4.75. Lựa chọn tài khoản để đăng nhập

Sau khi người dùng chọn tài khoản và đồng ý các quyền, giao diện dòng lệnh sẽ thông báo đăng nhập thành công.



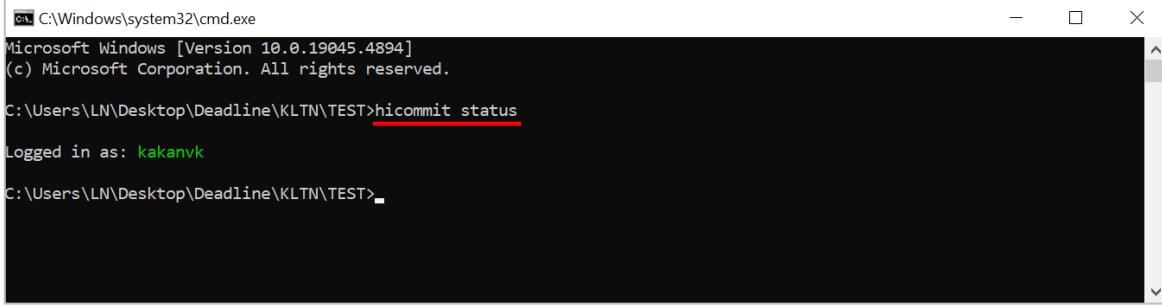
```
C:\Windows\system32\cmd.exe
Windows PowerShell [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LN>hicommmit login
Waiting for open the browser...
Hello, kakanvk! You have logged in successfully
C:\Users\LN>
```

Hình 4.76. Đăng nhập thành công với HiCommit CLI

Ngoài ra, người dùng có thể kiểm tra trạng thái đăng nhập với lệnh:

```
hicommit status
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

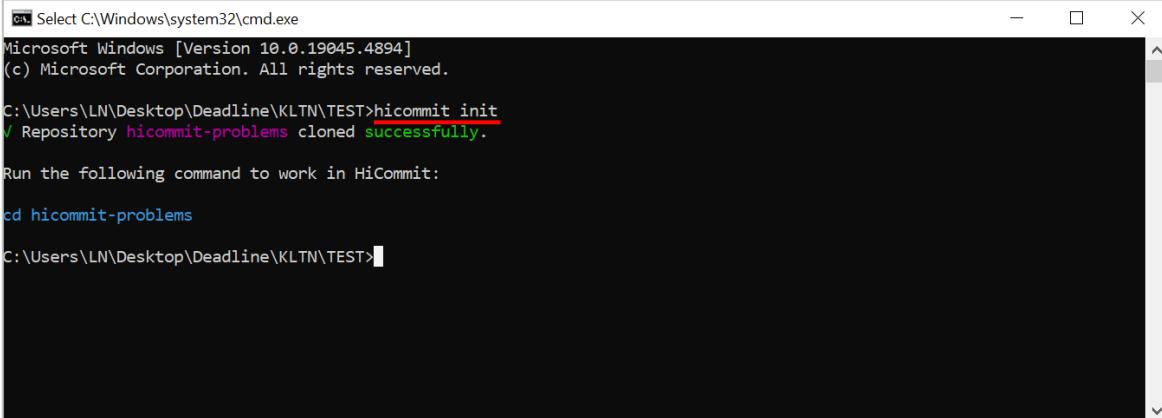
C:\Users\LN\Desktop\Deadline\KLTN\TEST>hicommit status
Logged in as: kakanvk

C:\Users\LN\Desktop\Deadline\KLTN\TEST>
```

Hình 4.77. Kiểm tra trạng thái đăng nhập trong HiCommit CLI

Để khởi tạo các thông tin và tạo thư mục làm việc, người dùng có thể sử dụng lệnh:

```
hicommit init
```



```
Select C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LN\Desktop\Deadline\KLTN\TEST>hicommit init
Repository hicommit-problems cloned successfully.

Run the following command to work in HiCommit:
cd hicommit-problems

C:\Users\LN\Desktop\Deadline\KLTN\TEST>
```

Hình 4.78. Khởi tạo với HiCommit CLI

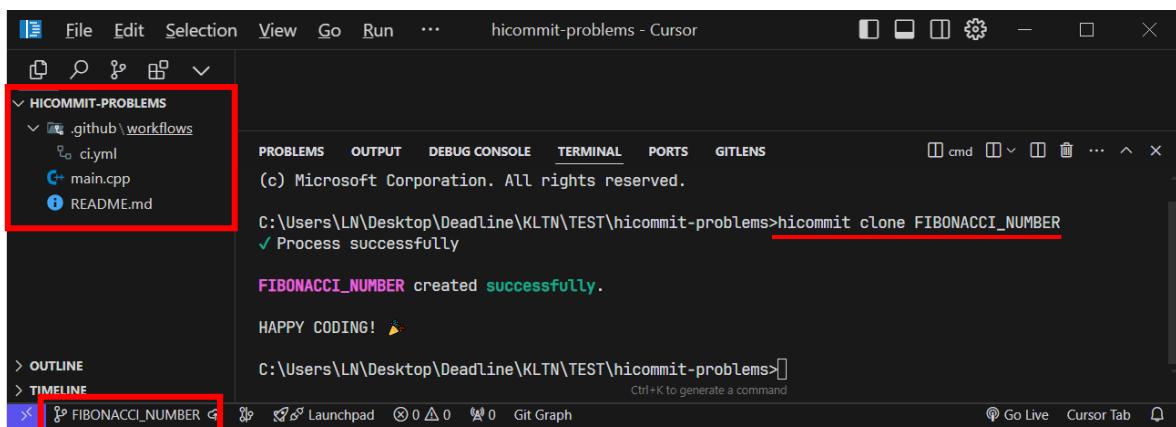
Sau khi thực thi thành công, thư mục `hicommit-problems` được tạo ra và được tự động clone từ GitHub về thư mục hiện tại. Thư mục này chính là repo GitHub được tạo riêng cho hệ thống HiCommit. Mỗi nhánh trong repo tương ứng với một bài tập mà người dùng đã thực hiện trên HiCommit. Chúng ta cần chuyển sang làm việc trong thư mục `hicommit-problems`:

```
cd hicommit-problems
```

Để khởi tạo dữ liệu cho một bài tập mới, người dùng cần thực hiện câu lệnh như sau:

```
hicommit clone <Problem_ID>
```

Với thông số <Problem_ID> là mã của bài tập cần tạo, có thể lấy dựa vào trang web HiCommit. Ví dụ bài tập “Tính số Fibonacci thứ n” có Problem_ID là “FIBONACCI_NUMBER”, cần tạo như sau:



Hình 4.79. Ví dụ tạo bài tập FIBONACCI_NUMBER

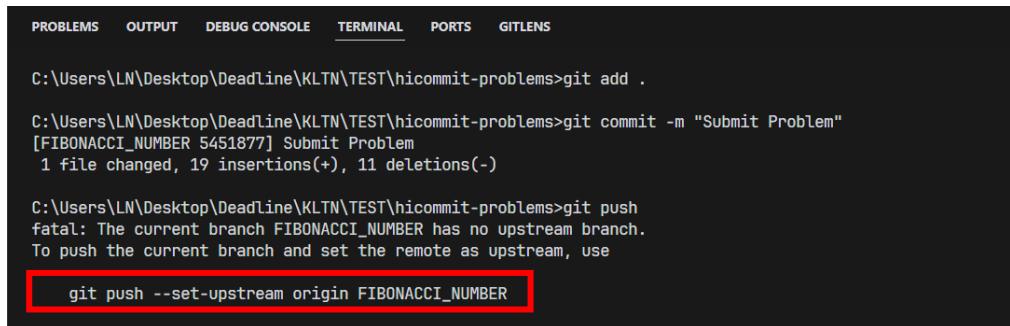
Sau khi thực thi câu lệnh thành công, một nhánh mới được tạo trong thư mục *hicommit-problems* với tên **FIBONACCI_NUMBER**. Đồng thời, hệ thống cũng tự động chuyển sang nhánh này để người dùng thuận tiện thao tác. Ứng dụng đã khởi tạo các file và thư mục cần thiết cho bài tập này theo đúng ngôn ngữ được quy định trên hệ thống HiCommit. Các file này bao gồm:

- Thư mục *.github/workflows*: chứa file *ci.yml*, là file cần thiết cho quá trình chấm tự động mỗi khi bài nộp được push lên GitHub.
- File **main.cpp**: file mã nguồn chính và duy nhất của nhánh, nơi người dùng trực tiếp làm việc để xử lý bài toán. Tuỳ vào ngôn ngữ lập trình, file này có thể là *main.c* (C), *main.cpp* (C++) hoặc *Main.java* (Java).
- File **README.md**: file chứa các thông tin của bài toán.

Người dùng cần chuẩn bị mã nguồn xử lý các yêu cầu của bài toán vào file *main.cpp* (*main.c* hoặc *Main.java*) để tiến hành nộp bài. Để nộp bài, có thể sử dụng lần lượt các câu lệnh Git như sau:

```
git add .
git commit -m "Submit Problem"
git push
```

Tuy nhiên, ở lần đầu tiên nộp bài với nhánh vừa tạo, Git sẽ báo rằng nhánh vừa tạo chưa tồn tại trên remote repo:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

C:\Users\LN\Desktop\Deadline\KLTN\TEST\hicommit-problems>git add .

C:\Users\LN\Desktop\Deadline\KLTN\TEST\hicommit-problems>git commit -m "Submit Problem"
[FIBONACCI_NUMBER 5451877] Submit Problem
1 file changed, 19 insertions(+), 11 deletions(-)

C:\Users\LN\Desktop\Deadline\KLTN\TEST\hicommit-problems>git push
fatal: The current branch FIBONACCI_NUMBER has no upstream branch.
To push the current branch and set the remote as upstream, use

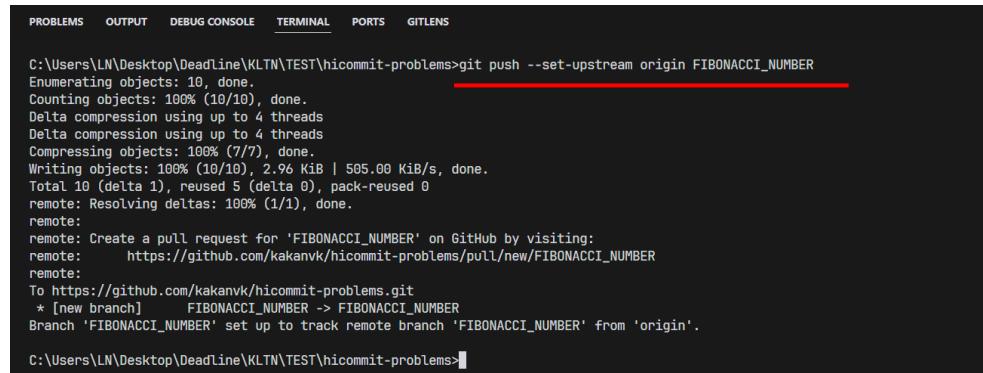
    git push --set-upstream origin FIBONACCI_NUMBER

```

Hình 4.80. Lỗi khi chưa có nhánh tương ứng trên remote repo

Để khắc phục, chúng ta cần sử dụng câu lệnh mà Git đã cung cấp ngay bên dưới:

```
git push --set-upstream origin <Problem_ID>
```



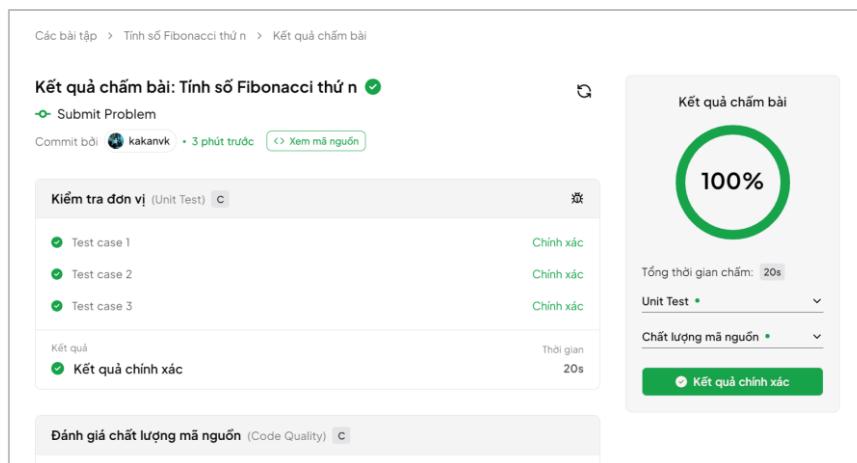
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

C:\Users\LN\Desktop\Deadline\KLTN\TEST\hicommit-problems>git push --set-upstream origin FIBONACCI_NUMBER
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 2.96 KiB | 505.00 KiB/s, done.
Total 10 (delta 1), reused 5 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'FIBONACCI_NUMBER' on GitHub by visiting:
remote:     https://github.com/kakanvk/hicommit-problems/pull/new/FIBONACCI_NUMBER
remote:
To https://github.com/kakanvk/hicommit-problems.git
 * [new branch]      FIBONACCI_NUMBER -> FIBONACCI_NUMBER
Branch 'FIBONACCI_NUMBER' set up to track remote branch 'FIBONACCI_NUMBER' from 'origin'.
C:\Users\LN\Desktop\Deadline\KLTN\TEST\hicommit-problems>

```

Hình 4.81. Nộp bài sử dụng Git thành công



Các bài tập > Tính số Fibonacci thứ n > Kết quả chấm bài

Kết quả chấm bài: Tính số Fibonacci thứ n ✓

- Submit Problem
- Commit bởi kakanvk • 3 phút trước
- Xem mã nguồn

Kiểm tra đơn vị (Unit Test)	
Test case 1	Chính xác
Test case 2	Chính xác
Test case 3	Chính xác
Kết quả	Thời gian 20s
Kết quả chính xác	

Đánh giá chất lượng mã nguồn (Code Quality)

Kết quả chấm bài

100%

Tổng thời gian chấm: 20s

Unit Test •

Chất lượng mã nguồn •

Kết quả chính xác

Hình 4.82. Kết quả chấm bài được hiển thị trên giao diện web của HiCommit

Khi không còn nhu cầu sử dụng ứng dụng, có thể thực hiện câu lệnh để gỡ cài đặt:

```
npm uninstall -g hicommitt
```

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Hệ thống HiCommit đã được xây dựng thành công và đáp ứng được các nhu cầu cần thiết của một hệ thống LMS tích hợp đánh giá lập trình tự động như:

- Chức năng khoá học
- Chức năng bài tập
- Chức năng cuộc thi
- Chức năng diễn đàn
- Các chức năng quản lý

Bằng việc sử dụng GitHub Actions, HiCommit đã tận dụng tối đa các tài nguyên được cung cấp miễn phí bởi nền tảng GitHub. Ngoài ra, mô hình ngôn ngữ lớn (LLM), cụ thể là Gemini cũng được tích hợp để đưa ra các đánh giá và các điều chỉnh cải thiện.

5.2. Hướng phát triển

Với hướng phát triển, chúng ta có thể mở rộng các chức năng theo chiều sâu (tối ưu hơn các tính năng hiện có) và theo chiều ngang (phát triển thêm các tính năng liên quan):

- *Theo chiều sâu:*

- + Cải thiện tốc độ xử lý của mỗi lần chấm bài: Với tốc độ mỗi lượt chấm khá chậm hiện tại (khoảng 20 giây/lượt), có thể nghiên cứu giải pháp để rút ngắn thời gian này.
 - + Giải quyết các vấn đề về việc tuỳ chỉnh các ngôn ngữ lập trình: Hiện tại, chỉ có ba ngôn ngữ lập trình được hỗ trợ trong HiCommit gồm C, C++ và Java. Việc thêm một ngôn ngữ mới vào hệ thống đòi hỏi quá trình kiểm thử khắt khe trước khi đưa vào sử dụng. Do đó, cần có giải pháp hợp lý để dễ dàng hơn trong việc thêm mới một ngôn ngữ.

- *Theo chiều ngang:*

- + Bổ sung tính năng chat để việc trao đổi giữa các người dùng với nhau được dễ dàng, thuận tiện hơn.
 - + Tính năng hỗ trợ với AI có thể được chú ý phát triển để giúp đưa ra các hỗ trợ hữu ích một cách nhanh chóng ngay từ giai đoạn ban đầu.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Iddon, Callum, Nasser Giacaman, and Valerio Terragni. "GradeStyle: GitHub-integrated and automated assessment of Java code style." 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). IEEE, 2023.
- [2] Chandrasekara, Chaminda, et al. "Introduction to GitHub Actions." Hands-on GitHub Actions: implement CI/CD with GitHub Actions workflows for your applications (2021): 1-8.
- [3] GitHub Docs, "GitHub REST API Documentation" [Online]. Available on: <https://docs.github.com/en/rest>
- [4] GitHub Docs, "GitHub Actions Documentation" [Online]. Available on: <https://docs.github.com/en/actions>
- [5] TailwindCSS Docs, "Get started with Tailwind CSS" [Online] Available on: <https://tailwindcss.com/docs>
- [6] Shadcn/ui Docs, "Introduction with Shadcn/ui" [Online] Available on: <https://ui.shadcn.com/docs>
- [7] Gemini Docs, "Get started with the Gemini API" [Online] Available on: <https://ai.google.dev/gemini-api/docs>

PHỤ LỤC 1: CÁC CÔNG CỤ VÀ PHẦN MỀM HỖ TRỢ XÂY DỰNG TRANG WEB

1. Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí và đa nền tảng, được phát triển bởi Microsoft. Được ra mắt lần đầu vào năm 2015, VS Code nhanh chóng trở thành một trong những công cụ phát triển phổ biến nhất trong cộng đồng lập trình viên nhờ vào sự linh hoạt, hiệu suất cao và khả năng tùy biến mạnh mẽ. Trình soạn thảo này hỗ trợ nhiều ngôn ngữ lập trình khác nhau như JavaScript, Python, C++, Java, HTML/CSS và nhiều ngôn ngữ khác thông qua hệ thống mở rộng phong phú.

Một trong những điểm mạnh lớn nhất của Visual Studio Code là sự tích hợp liền mạch với các hệ thống quản lý mã nguồn như Git, cho phép các nhà phát triển dễ dàng quản lý phiên bản, thực hiện commit, push và pull mà không cần rời khỏi trình soạn thảo.

Visual Studio Code có kho tiện ích mở rộng (extensions) phong phú, cho phép người dùng tùy chỉnh và mở rộng chức năng của trình soạn thảo theo nhu cầu. Người dùng có thể cài đặt các tiện ích hỗ trợ từ đơn giản như đổi màu giao diện, đến phức tạp hơn như tích hợp các công cụ phân tích mã, debugger hay hỗ trợ lập trình với framework cụ thể. VS Code cũng được tối ưu hóa để chạy mượt mà trên cả các máy tính có cấu hình thấp, làm cho nó trở thành lựa chọn lý tưởng cho cả những người mới học lập trình và các chuyên gia trong ngành.

Ngoài ra, với sự hỗ trợ mạnh mẽ từ cộng đồng, Visual Studio Code luôn được cập nhật và phát triển không ngừng giúp nó thích nghi với sự phát triển nhanh chóng của các công nghệ lập trình hiện đại. VS Code là sự kết hợp hoàn hảo giữa một trình soạn thảo mã nhẹ nhàng nhưng đầy đủ tính năng của một môi trường phát triển tích hợp (IDE), tạo điều kiện tối ưu cho việc phát triển phần mềm trên nhiều nền tảng khác nhau.

2. Xampp

XAMPP là một gói phần mềm mã nguồn mở được thiết kế để giúp các nhà phát triển dễ dàng triển khai và kiểm thử ứng dụng web trên máy tính cá nhân. Tên gọi "XAMPP" là viết tắt của các thành phần chính mà phần mềm này cung cấp: X (hệ điều hành bất kỳ), A (Apache - máy chủ web), M (MariaDB/MySQL - hệ quản trị cơ sở dữ liệu), P (PHP - ngôn ngữ lập trình phía máy chủ) và P (Perl - một ngôn ngữ lập trình khác). XAMPP được phát

triển và phân phối bởi Apache Friends và có sẵn trên nhiều nền tảng hệ điều hành như Windows, Linux và macOS.

Điểm nổi bật của XAMPP là tính đơn giản và tiện lợi, cho phép các lập trình viên nhanh chóng cài đặt và thiết lập một môi trường phát triển web đầy đủ trên máy tính cá nhân mà không cần phải cài đặt từng thành phần riêng lẻ. Nhờ đó, XAMPP thường được sử dụng trong quá trình phát triển và thử nghiệm ứng dụng web trước khi triển khai lên môi trường thực tế. XAMPP cung cấp một giao diện đồ họa thân thiện để quản lý các dịch vụ, cho phép người dùng khởi động hoặc dừng các thành phần như Apache, MariaDB và các công cụ khác chỉ với vài cú nhấp chuột.

Apache, một trong những thành phần chính của XAMPP, là một trong những máy chủ web phổ biến nhất trên thế giới, có khả năng xử lý hàng loạt yêu cầu HTTP và cung cấp nội dung web cho người dùng. MariaDB hoặc MySQL, hệ quản trị cơ sở dữ liệu tích hợp trong XAMPP, cung cấp giải pháp quản lý dữ liệu hiệu quả, cho phép các lập trình viên xây dựng và quản lý cơ sở dữ liệu cho các ứng dụng web. PHP và Perl là các ngôn ngữ lập trình phía máy chủ, hỗ trợ xây dựng các trang web động và thực thi các tác vụ phức tạp như xử lý form, tương tác với cơ sở dữ liệu và thực hiện logic nghiệp vụ.

Ngoài các thành phần cơ bản, XAMPP còn tích hợp nhiều công cụ hữu ích khác như phpMyAdmin, một giao diện web cho phép quản lý cơ sở dữ liệu MySQL/MariaDB trực quan và dễ dàng hơn. Công cụ này rất hữu ích cho việc quản lý, nhập/xuất dữ liệu và chạy các truy vấn SQL mà không cần phải thao tác qua dòng lệnh.

XAMPP là sự lựa chọn lý tưởng cho các nhà phát triển web, đặc biệt là những người mới bắt đầu hoặc những ai muốn nhanh chóng thiết lập môi trường phát triển trên máy tính cá nhân. Với XAMPP, quá trình triển khai và kiểm thử ứng dụng web trở nên dễ dàng và hiệu quả hơn giúp tiết kiệm thời gian và công sức trong việc cài đặt các thành phần cần thiết cho một máy chủ web hoàn chỉnh.

3. Postman

Postman là một công cụ mạnh mẽ và phổ biến được sử dụng để phát triển, kiểm thử và quản lý các API (Application Programming Interface). Được phát hành lần đầu vào năm 2012, Postman đã trở thành công cụ không thể thiếu đối với các lập trình viên và tester trong quá trình xây dựng và phát triển phần mềm dựa trên API. Với giao diện đồ họa trực

quan và dễ sử dụng, Postman cho phép người dùng tạo, gửi và phân tích các yêu cầu HTTP/HTTPS một cách nhanh chóng mà không cần phải viết mã hay dùng các công cụ dòng lệnh phức tạp.

Một trong những tính năng chính của Postman là hỗ trợ nhiều phương thức yêu cầu HTTP khác nhau như GET, POST, PUT, DELETE, PATCH và nhiều phương thức khác. Điều này cho phép các lập trình viên mô phỏng và kiểm thử các yêu cầu đến API một cách chính xác, cũng như kiểm tra cách thức API phản hồi với các dữ liệu đầu vào khác nhau. Giao diện của Postman cung cấp cho người dùng khả năng cấu hình chi tiết yêu cầu với các tham số, tiêu đề (headers), nội dung thân (body), cũng như quản lý cookies và xác thực người dùng.

Postman không chỉ dừng lại ở việc gửi yêu cầu và nhận phản hồi, nó còn cung cấp nhiều tính năng nâng cao giúp tối ưu hóa quy trình phát triển API. Một trong số đó là Postman Collections, cho phép người dùng tổ chức và lưu trữ các nhóm yêu cầu liên quan vào một tập hợp để tái sử dụng hoặc chia sẻ với đội nhóm. Postman cũng hỗ trợ environment variables (biến môi trường), cho phép người dùng tạo các môi trường khác nhau (như phát triển, thử nghiệm và sản xuất) với các giá trị biến cụ thể giúp dễ dàng chuyển đổi giữa các cấu hình khác nhau khi kiểm thử.

Postman còn có khả năng tự động hóa kiểm thử API. Người dùng có thể viết các đoạn mã kiểm thử bằng JavaScript trong Postman để tự động xác nhận phản hồi API có đúng như mong đợi hay không, từ đó giúp đảm bảo chất lượng API trong suốt quá trình phát triển. Tính năng này rất hữu ích cho việc thực hiện kiểm thử hồi quy và kiểm tra toàn vẹn của API sau mỗi lần cập nhật.

Ngoài ra, Postman hỗ trợ tích hợp CI/CD (Continuous Integration/Continuous Deployment) giúp tự động hóa quá trình kiểm thử API và triển khai trên các môi trường khác nhau một cách liền mạch. Điều này giúp cải thiện hiệu suất làm việc và đảm bảo các API luôn hoạt động đúng đắn trước khi triển khai lên môi trường sản xuất.

Với cộng đồng người dùng rộng lớn và tài liệu hỗ trợ phong phú, Postman không chỉ là công cụ hỗ trợ phát triển và kiểm thử API mà còn là một nền tảng hợp tác mạnh mẽ, cho phép các đội phát triển làm việc cùng nhau hiệu quả hơn trong quá trình xây dựng các giải pháp dựa trên API. Với những tính năng vượt trội và tính thân thiện với người dùng,

Postman đã trở thành lựa chọn hàng đầu cho các nhà phát triển API trên toàn thế giới.

4. Ngrok

Ngrok là một công cụ mạnh mẽ và tiện lợi được sử dụng để tạo ra các đường hầm bảo mật (secure tunnels) từ Internet tới máy tính cục bộ. Nó cho phép các lập trình viên dễ dàng phơi bày (expose) một máy chủ cục bộ hoặc ứng dụng đang chạy trên máy cá nhân lên internet thông qua một URL tạm thời mà không cần cấu hình phức tạp như chuyển tiếp cổng (port forwarding) hay thay đổi cấu hình mạng.

Ngrok được sử dụng rộng rãi trong quá trình phát triển và kiểm thử, đặc biệt là khi cần chia sẻ ứng dụng cục bộ với bên ngoài hoặc khi cần kiểm thử các webhook, API từ các dịch vụ bên ngoài. Khi một ứng dụng cục bộ được mở ra thông qua Ngrok, các đối tác hoặc dịch vụ từ bên ngoài có thể truy cập vào nó thông qua một URL ngẫu nhiên do Ngrok cung cấp mà không cần biết đến địa chỉ IP hay cấu hình nội bộ của người phát triển.

Một số tính năng chính của Ngrok:

- *Tạo đường hầm HTTPS bảo mật*: Ngrok cung cấp các URL HTTPS an toàn để truy cập ứng dụng cục bộ. Điều này đặc biệt hữu ích trong việc kiểm thử các dịch vụ yêu cầu kết nối bảo mật.

- *Hỗ trợ nhiều giao thức*: Ngrok không chỉ hỗ trợ giao thức HTTP/HTTPS mà còn có thể tạo đường hầm cho các giao thức khác như TCP. Điều này cho phép phát triển và kiểm thử nhiều loại ứng dụng khác nhau, từ web server đến các dịch vụ cơ sở dữ liệu hay ứng dụng thời gian thực.

- *Kiểm thử webhook*: Đối với các ứng dụng yêu cầu tích hợp với các dịch vụ bên ngoài qua webhook như Stripe, GitHub, hay PayPal, Ngrok là một công cụ lý tưởng. Nó cho phép nhận các sự kiện từ dịch vụ bên ngoài vào máy cục bộ mà không cần triển khai lên máy chủ thực.

- *Giao diện giám sát thời gian thực*: Ngrok cung cấp một giao diện web cho phép theo dõi toàn bộ các yêu cầu đến ứng dụng cục bộ, bao gồm tiêu đề (headers), thân (body) và các thông tin khác. Điều này giúp lập trình viên dễ dàng phân tích và gỡ lỗi các yêu cầu từ bên ngoài.