

ĐỒ ÁN MÔN HỌC: Kiểm thử và đảm bảo chất lượng phần mềm**BÀI TIỂU LUẬN KIỂM THỬ WEBSITE TIN TỨC**Ngành: **CÔNG NGHỆ THÔNG TIN**

Lớp: 21DTHE1

Giảng viên hướng dẫn: Trần Thị Vân Anh

Sinh viên thực hiện:

Nguyễn Nhật Huy	MSSV: 218060
Nguyễn Đức Anh	MSSV: 218060
Đỗ Công Chiến	MSSV: 2180607327
Nguyễn Hồng Hải Dương	MSSV: 2180608795
Nguyễn Trung Kiên	MSSV: 218060
Trần Lê Minh Hiếu	MSSV: 2180608369
Nguyễn Thế Vũ	MSSV: 2180608244
Nguyễn Ngô Trúc Chi	MSSV: 2180607325
Nguyễn Xuân Bắc	MSSV: 218060

TP. Hồ Chí Minh, ngày 4 tháng 04 năm 2025

NHẬN XÉT

[illegible]

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến cô Trần Thị Vân Anh, người đã tận tâm hướng dẫn và chia sẻ những kiến thức quý báu trong suốt quá trình thực hiện đồ án chuyên ngành. Cô đã giúp em hiểu rõ hơn về các phương pháp và công nghệ trong việc phát triển các ứng dụng thực tế, đặc biệt là trong việc xây dựng website quản lý phòng khám. Những lời khuyên của cô không chỉ là kim chỉ nam giúp em vượt qua những khó khăn trong quá trình thực hiện đồ án. Em cũng xin gửi lời cảm ơn đến các cô trong khoa Công Nghệ Thông Tin, những người đã tạo điều kiện và cung cấp nền tảng kiến thức vững chắc, giúp em trang bị đầy đủ các kỹ năng cần thiết để hoàn thành đồ án này. Trong suốt quá trình thực hiện đồ án, em cũng nhận được sự giúp đỡ và sự chia sẻ nhiệt tình từ những người bạn đồng hành, những người đã không ngừng hỗ trợ em trong việc nghiên cứu, tìm kiếm tài liệu và giải quyết các vấn đề kỹ thuật. Em vô cùng trân trọng sự động viên và sự giúp đỡ quý báu này, vì chính những hỗ trợ ấy đã tạo điều kiện để em hoàn thành đồ án một cách tốt nhất. Mặc dù đã cố gắng hết sức để hoàn thiện đồ án, nhưng em nhận thức rằng vẫn còn nhiều thiếu sót và hạn chế. Em rất mong nhận được sự đóng góp và ý kiến từ thầy cô, các chuyên gia, và các bạn để bài báo cáo của em có thể được hoàn thiện hơn nữa. Những góp ý này sẽ là động lực để em tiếp tục phấn đấu, học hỏi và phát triển trong nghề lập trình. Cuối cùng, em xin chân thành cảm ơn tất cả những người đã đồng hành cùng em trong suốt quá trình thực hiện đồ án chuyên ngành. Em hy vọng rằng với nền tảng kiến thức và kinh nghiệm đã tích lũy, em sẽ có thể áp dụng vào công việc thực tế trong tương lai và đóng góp vào sự phát triển của ngành Công Nghệ Thông Tin.

Xin chân thành cảm ơn!

MỤC LỤC

NHẬN XÉT	2
LỜI CẢM ƠN	3
MỤC LỤC	4
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	6
1.1 GIỚI THIỆU SẢN PHẨM.....	6
1.2 CÁC CHỨC NĂNG TRONG SẢN PHẨM	7
CHƯƠNG 2: ÁP DỤNG TEST CASE BLACK BOX VÀ WHITE BOX.....	9
2.1 CÁCH ÁP DỤNG BLACK BOX VÀ WHITE BOX.....	10
2.2 VIẾT CÁC TEST CASE CHÍNH.....	10
2.3 VIẾT CÁC CHỨC NĂNG CHÍNH TRONG UT CHO FUNCTION	12
CHƯƠNG 3: THỰC NGHIỆM	14
CHƯƠNG 4: KẾT LUẬN	17
4.1 KẾT QUẢ THỰC HIỆN	17
4.2 HƯỚNG PHÁT TRIỂN.....	18
TÀI LIỆU THAM KHẢO	24

Danh Mục Bảng

Bảng 1 : Bảng workflow quản lý tất cả tài khoản	22
Bảng 2 : Bảng workflow đặt lịch khám bệnh.....	24
Bảng 3 : Bảng workflow quản lý phòng khám	26
Bảng 4 :Bảng workflow tạo tài khoản.....	28
Bảng 6 :Bảng workflow quên mật khẩu.....	33
Bảng 7 : Bảng quản lý bệnh nhân	36
Bảng 8 : Bảng workflow quản lý chuyên khoa khám bệnh	39
Bảng 9 : Bảng workflow thống kê	40

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 GIỚI THIỆU SẢN PHẨM

Đề tài này trình bày quá trình phát triển một ứng dụng web hoàn chỉnh, hoạt động như một Hệ thống Quản lý Nội dung (CMS) hoặc một trang blog hiện đại. Sản phẩm được xây dựng trên nền tảng công nghệ PHP, sử dụng framework Laravel để tối ưu hóa quy trình phát triển và đảm bảo tính ổn định, bảo mật. Mục tiêu chính của ứng dụng là tạo ra một không gian trực tuyến nơi người quản trị có thể dễ dàng xuất bản, quản lý các bài viết, tin tức hoặc bất kỳ dạng nội dung số nào. Đồng thời, ứng dụng cung cấp một giao diện thân thiện cho người dùng cuối để tiếp cận, tương tác với các nội dung đó.

Các khối chức năng chính được xây dựng trong ứng dụng bao gồm:

- **Hiển thị và Tương tác Người dùng (Frontend):**
 - Trình bày danh sách bài viết (mới nhất, nổi bật, theo chủ đề).
 - Chi tiết bài viết với đầy đủ nội dung và hình ảnh.
 - Cho phép người đọc bình luận, đánh giá.
 - Cung cấp công cụ tìm kiếm, lọc bài viết theo danh mục, thẻ (tags).
 - Các trang tĩnh như Giới thiệu (About), Liên hệ (Contact).
 - Tính năng đăng ký/đăng nhập tài khoản người dùng.
 - Hệ thống thăm dò ý kiến (Polls).
- **Quản trị Hệ thống (Backend/Admin Dashboard):**
 - Quản lý toàn diện bài viết (thêm, sửa, xóa, duyệt, lên lịch).
 - Quản lý các danh mục, thẻ (tags) để tổ chức nội dung.
 - Quản lý người dùng, phân chia vai trò và quyền hạn truy cập chi tiết.
 - Quản lý các bình luận, liên hệ từ người dùng.
 - Cấu hình các thông số, cài đặt chung cho website.
 - Theo dõi thống kê lượt truy cập.
 - Quản lý hệ thống thăm dò ý kiến.

Chương này đóng vai trò giới thiệu tổng thể về sản phẩm và phạm vi chức năng của nó. Các khía cạnh về phân tích yêu cầu, thiết kế cơ sở dữ liệu, kiến trúc hệ thống và chi tiết triển khai các chức năng sẽ được làm rõ trong các chương sau.

1.2 CÁC CHỨC NĂNG TRONG SẢN PHẨM

Ứng dụng web được xây dựng bao gồm hai nhóm chức năng chính, phục vụ cho các đối tượng người dùng khác nhau: nhóm chức năng dành cho người dùng cuối (Frontend) và nhóm chức năng dành cho quản trị viên (Backend).

1.2.1 Chức năng Frontend (Dành cho người dùng cuối)

Đây là các chức năng mà khách truy cập và người dùng đã đăng ký có thể tương tác trực tiếp trên giao diện website:

- **Xem và duyệt nội dung:**
 - **Trang chủ:** Hiện thị các bài viết mới nhất, có thể có các mục bài viết nổi bật (hot posts), danh sách danh mục, thẻ phổ biến, và các cuộc thăm dò ý kiến (polls).
 - **Xem chi tiết bài viết:** Hiện thị đầy đủ nội dung, hình ảnh, thông tin tác giả, ngày đăng, danh mục, thẻ liên quan và khu vực bình luận.
 - **Xem bài viết theo Danh mục:** Liệt kê tất cả bài viết thuộc một danh mục cụ thể.
 - **Xem bài viết theo Thẻ (Tag):** Liệt kê tất cả bài viết được gắn một thẻ cụ thể.
- **Tìm kiếm:** Cho phép người dùng tìm kiếm bài viết dựa trên từ khóa.
- **Tương tác:**
 - **Bình luận:** Người dùng (thường là đã đăng nhập) có thể để lại bình luận dưới mỗi bài viết.
 - **Liên hệ:** Cung cấp một biểu mẫu (form) để người dùng gửi ý kiến, câu hỏi đến quản trị viên. Email liên hệ sẽ được gửi đi.
 - **Bỏ phiếu (Polls):** Người dùng có thể tham gia bình chọn cho các cuộc thăm dò ý kiến đang hoạt động.
 - **Đăng ký nhận bản tin (Newsletter):** Cho phép người dùng nhập email để nhận các thông tin cập nhật.
- **Xác thực và Quản lý tài khoản:**
 - **Đăng ký:** Tạo tài khoản người dùng mới.
 - **Đăng nhập/Đăng xuất:** Truy cập vào hệ thống với tài khoản đã có.
 - **Quên mật khẩu:** Chức năng cho phép người dùng đặt lại mật khẩu qua email.

- **Xác thực Email:** (Có thể có) Yêu cầu người dùng xác nhận địa chỉ email sau khi đăng ký.
- **Trang cá nhân (Profile):** (Có thể có) Cho phép người dùng xem và chỉnh sửa thông tin cá nhân.
- **Trang tĩnh:**
 - **Giới thiệu (About):** Hiển thị thông tin giới thiệu về website hoặc tổ chức.

1.2.2 Chức năng Backend (Khu vực Quản trị - Admin Dashboard)

Đây là các chức năng dành riêng cho quản trị viên và những người dùng được cấp quyền, truy cập thông qua một giao diện quản lý riêng biệt:

- **Bảng điều khiển (Dashboard):**
 - Hiển thị thông tin tổng quan, thống kê nhanh về số lượng bài viết, người dùng, bình luận, lượt truy cập (visitors), v.v.
- **Quản lý Nội dung:**
 - **Quản lý Bài viết:** Tạo mới, xem danh sách, chỉnh sửa, xóa bài viết. Tích hợp trình soạn thảo văn bản (TinyMCE) để định dạng nội dung, tải lên hình ảnh, đặt trạng thái (xuất bản, nháp), chọn danh mục, gắn thẻ, lên lịch đăng bài (schedule).
 - **Quản lý Danh mục:** Tạo mới, xem danh sách, chỉnh sửa, xóa các danh mục bài viết.
 - **Quản lý Thẻ (Tags):** Tạo mới, xem danh sách, chỉnh sửa, xóa các thẻ.
- **Quản lý Tương tác:**
 - **Quản lý Bình luận:** Xem danh sách, phê duyệt, xóa các bình luận của người dùng.
 - **Quản lý Liên hệ:** Xem danh sách và nội dung các thư liên hệ được gửi từ người dùng, có thể xóa liên hệ.
 - **Quản lý Thăm dò ý kiến (Polls):** Tạo, sửa, xóa các cuộc thăm dò và các lựa chọn trả lời (poll options).
- **Quản lý Người dùng và Phân quyền:**
 - **Quản lý Người dùng:** Xem danh sách, tạo mới, chỉnh sửa thông tin, xóa tài khoản người dùng.

- **Quản lý Vai trò (Roles):** Tạo, sửa, xóa các vai trò (ví dụ: Admin, Editor, User). Gán quyền hạn cụ thể cho từng vai trò.
- **Quản lý Quyền hạn (Permissions):** (Thường được định nghĩa sẵn trong code hoặc quản lý thông qua giao diện gán cho vai trò) Xác định các hành động mà người dùng có thể thực hiện.
- **Quản lý Hệ thống:**
 - **Cài đặt (Settings):** Cho phép chỉnh sửa các thông tin chung của website như tên website, mô tả, logo, nội dung trang Giới thiệu, v.v.
 - **Quản lý Newsletter:** (Có thể có) Xem danh sách email đã đăng ký nhận tin.
- **Tiện ích:**
 - **Quản lý Hình ảnh (thông qua TinyMCE):** Tải và quản lý hình ảnh khi soạn thảo bài viết.

CHƯƠNG 2: ÁP DỤNG TEST CASE BLACK BOX VÀ WHITE BOX

2.1 CÁCH ÁP DỤNG BLACK BOX VÀ WHITE BOX

2.1.1 Kiểm thử hộp đen (Black Box Testing)

Cách áp dụng:

- Xác định yêu cầu và chức năng của phần mềm: Dựa trên tài liệu đặc tả yêu cầu (SRS) để hiểu rõ về các chức năng cần kiểm thử.
- Thiết kế test case: Dựa trên các kỹ thuật như Equivalence Partitioning (Phân vùng tương đương), Boundary Value Analysis (Phân tích giá trị biên) hoặc State Transition Testing (Kiểm thử chuyển trạng thái).
- Thực hiện kiểm thử: Nhập dữ liệu kiểm thử vào hệ thống và quan sát kết quả đầu ra. So sánh với kết quả mong đợi để xác định lỗi.
- Ghi nhận và báo cáo lỗi: Nếu có sự sai lệch giữa kết quả thực tế và mong đợi, tester cần ghi nhận lỗi và gửi báo cáo cho nhóm phát triển để sửa chữa.

2.1.2 Kiểm thử hộp trắng (White Box Testing)

Cách áp dụng:

- Phân tích mã nguồn: Xem xét mã nguồn để xác định các thành phần quan trọng cần kiểm thử.
- Thiết kế test case: Sử dụng các kỹ thuật như Control Flow Testing (Kiểm thử luồng điều khiển), Data Flow Testing (Kiểm thử luồng dữ liệu) hoặc Branch Coverage (Bao phủ nhánh) để đảm bảo mọi đường đi trong mã đều được kiểm tra.
- Viết test script: Tạo các đoạn mã hoặc sử dụng công cụ kiểm thử tự động để kiểm tra từng thành phần của chương trình.
- Thực hiện kiểm thử và phân tích lỗi: Chạy test case, xem xét kết quả và điều chỉnh nếu cần.

2.2 VIẾT CÁC TEST CASE CHÍNH

- Chức năng đăng nhập
 - + Kiểm tra đăng nhập thành công
 - + Kiểm tra đăng nhập với mật khẩu sai
 - + Kiểm tra đăng nhập với email sai (không tồn tại)

- + Kiểm tra đăng nhập khi bỏ trống email
- + Kiểm tra đăng nhập khi bỏ trống mật khẩu
- + Kiểm tra đăng nhập khi bỏ trống cả email và mật khẩu
- + Kiểm tra đăng nhập với email sai định dạng
- Chức năng liên hệ sử dụng kỹ thuật phân vùng tương đương:
 - + Kiểm tra gửi đi với dữ liệu hợp lệ
 - + Kiểm tra gửi đi khi bỏ trống tất cả các trường.
 - + Kiểm tra nhập email không hợp lệ.
 - + Kiểm tra bỏ trống trường Họ và Tên.
 - + Kiểm tra nhập tiêu đề quá dài.
 - + Kiểm tra bỏ trống trường nội dung.
 - + Kiểm tra nhập ký tự đặc biệt vào các trường.
- Chức năng tìm kiếm
 - + Tìm kiếm tin tức theo tiêu đề chính xác
 - + Tìm kiếm tin tức theo ký tự
 - + Tìm kiếm tin tức với từ khóa chỉ chứa khoảng trắng
 - + Tìm kiếm tin tức với từ khóa viết hoa/thường khác nhau
 - + Tìm kiếm tin tức với link bài viết
 - + Tìm kiếm tin tức với từ khóa trùng lặp
 - + Tìm kiếm với từ khóa trống
- Chức năng đăng ký
 - + Đăng ký thành công với thông tin hợp lệ
 - + Đăng ký bằng email đã được sử dụng
 - + Đăng ký với các trường để trống
 - + Đăng ký với mật khẩu không hợp lệ
 - + Đăng ký với mật khẩu không trùng khớp
 - + Đăng ký với tên đăng nhập có kí tự đặc biệt Đăng ký với tên đăng nhập có kí tự đặc biệt
 - + Đăng ký với email sai định dạng
- Chức năng bình luận sử dụng kỹ thuật phân vùng tương đương:
 - + Phải đăng nhập mới gửi được bình luận.

- + Người dùng đã đăng nhập, để trống trường nhập nội dung bình luận và nhấn gửi.
- + Người dùng đã đăng nhập, nhập bình luận quá ít ký tự hoặc vượt quá giới hạn ký tự cho phép.
- + Người dùng đã đăng nhập, nhập nội dung bình luận hợp lệ và nhấn gửi.
- + Người dùng đã đăng nhập, nhập nội dung bình luận chứa các ký tự đặc biệt như "!!!@@@###\$\$\$".
- + Người dùng đã đăng nhập, nhập nội dung bình luận có chứa URL như "https://www.google.com/".
- + Người dùng đã đăng nhập, gửi bình luận có nội dung giống với bình luận trong một bài viết khác.

2.3 VIẾT CÁC CHỨC NĂNG CHÍNH TRONG UT CHO FUNCTION

Các chức năng White Box Testing bao gồm:

- Thêm tài khoản
- Bình luận
- Cập nhật tài khoản
- Tạo bài viết
- Quên mật khẩu

2.3.1. Thêm tài khoản

- Kiểm thử dòng điều khiển (Control Flow Testing): Phân tích luồng thực thi của hàm.
- Kiểm thử đường dẫn (Path Testing): Xác định các đường dẫn cơ sở để đảm bảo kiểm thử đầy đủ.
- Kiểm thử điều kiện (Condition Testing): Kiểm tra các nhánh điều kiện như `if ($request->has('image'))`.
- Tính toán độ phức tạp chu trình (Cyclomatic Complexity - $V(G)$): Xác định số lượng test case tối thiểu cần thiết.

2.3.2. Chức năng bình luận

- Kiểm thử dòng điều khiển (Control Flow Testing): Phân tích luồng thực thi của hàm thông qua đồ thị điều khiển (CFG).
- Kiểm thử đường dẫn (Path Testing): Xác định các đường dẫn cơ sở để đảm bảo kiểm thử đầy đủ các tình huống.
- Kiểm thử điều kiện (Condition Testing): Kiểm tra các nhánh điều kiện như `if ($validated->fails())` và `if ($post)`.

- Tính toán độ phức tạp chu trình (Cyclomatic Complexity - $V(G)$): Xác định số lượng test case tối thiểu cần thiết để kiểm tra tất cả các luồng điều khiển.

2.3.3. Chức năng cập nhật tài khoản

- Kiểm thử dòng điều khiển (Control Flow Testing): Phân tích luồng thực thi của hàm thông qua đồ thị điều khiển (CFG).
- Kiểm thử đường dẫn (Path Testing): Xác định các đường dẫn cơ sở để đảm bảo kiểm thử đầy đủ các trường hợp cập nhật tài khoản.
- Kiểm thử điều kiện (Condition Testing): Kiểm tra các nhánh điều kiện như `if (!$valid['password'] == null)` và `if ($request->has('image'))`.
- Tính toán độ phức tạp chu trình (Cyclomatic Complexity - $V(G)$): Xác định số lượng test case tối thiểu cần thiết để kiểm tra tất cả các nhánh điều khiển.
- Phủ nhánh (Branch Coverage): Đảm bảo tất cả các nhánh trong cấu trúc điều kiện (if-else) đều được kiểm thử.

2.3.4. Chức năng đăng bài viết

- Kiểm thử dòng điều khiển (Control Flow Testing): Phân tích luồng thực thi của hàm thông qua đồ thị điều khiển (CFG).
- Kiểm thử đường dẫn (Path Testing): Xác định các đường dẫn cơ sở để đảm bảo kiểm thử đầy đủ các trường hợp đăng bài.
- Kiểm thử điều kiện (Condition Testing): Kiểm tra các nhánh điều kiện như `if ($request->has('thumbnail'))` và `if (count($tags_id) > 0)`.
- Phủ nhánh (Branch Coverage): Đảm bảo tất cả các nhánh trong cấu trúc điều kiện (if-else) đều được kiểm thử, bao gồm các trường hợp có ảnh, không có ảnh, có tags, không có tags.

2.3.5. Chức năng quên mật khẩu

- Kiểm thử dòng điều khiển (Control Flow Testing): Phân tích luồng thực thi của hàm đặt lại mật khẩu bằng sơ đồ điều khiển dòng chảy (CFG).
- Kiểm thử đường dẫn (Path Testing): Xác định các đường dẫn cơ sở để đảm bảo kiểm thử đầy đủ các trường hợp nhập email hợp lệ, không hợp lệ, email không tồn tại, hoặc lỗi trong quá trình gửi mail.
- Kiểm thử điều kiện (Condition Testing): Kiểm tra các nhánh điều kiện `if ($request->validate(['email' => 'required', 'email']))` và `if ($status == Password::RESET_LINK_SENT)`.
- Tính toán độ phức tạp chu trình ($V(G)$): Xác định số lượng test case tối thiểu cần thiết để kiểm tra tất cả các nhánh điều khiển.
- Phủ đường (Path Coverage): Đảm bảo tất cả các đường dẫn có thể xảy ra trong quá trình xử lý.

CHƯƠNG 3: Tổng quan về Robot Framework

3.1. Khái niệm Robot Framework

Robot Framework là một công cụ kiểm thử tự động mã nguồn mở, được thiết kế để hỗ trợ kiểm thử chấp nhận và phát triển dựa trên kiểm thử. Điểm nổi bật của Robot Framework nằm ở cú pháp dựa trên từ khóa, giúp đơn giản hóa việc tạo kịch bản kiểm thử, phù hợp cho cả người không có nền tảng lập trình sâu và các chuyên gia kiểm thử. Framework này linh hoạt trong việc kiểm thử nhiều loại ứng dụng như web, di động, API, hoặc các hệ thống phần mềm phức tạp, nhờ khả năng hoạt động trên nhiều nền tảng và tích hợp với các công cụ khác.

3.2. Các đặc trưng chính của Robot Framework

Robot Framework được đánh giá cao nhờ những đặc điểm sau:

- **Dựa trên từ khóa:** Sử dụng từ khóa để mô tả các bước kiểm thử, giúp kịch bản dễ hiểu và dễ bảo trì.
- **Tích hợp thư viện phong phú:** Hỗ trợ các thư viện như SeleniumLibrary (kiểm thử web), AppiumLibrary (kiểm thử di động), đồng thời cho phép mở rộng bằng Python hoặc Java.
- **Báo cáo trực quan:** Tạo báo cáo và nhật ký chi tiết dưới dạng HTML sau mỗi lần chạy, hỗ trợ phân tích kết quả hiệu quả.
- **Khả năng tương thích:** Dễ dàng kết nối với các hệ thống CI/CD như Jenkins hoặc các công cụ quản lý kiểm thử.
- **Miễn phí và mở:** Là phần mềm mã nguồn mở, người dùng có thể tùy chỉnh theo nhu cầu mà không tốn chi phí.

3.3. Cách tổ chức của Robot Framework

Một kịch bản kiểm thử trong Robot Framework thường bao gồm các thành phần chính:

- **Settings:** Cấu hình ban đầu như import thư viện, khai báo tài nguyên hoặc biến toàn cục.
- **Variables:** Định nghĩa các biến để lưu trữ dữ liệu, tăng tính tái sử dụng.
- **Test Cases:** Các trường hợp kiểm thử cụ thể, được xây dựng từ các từ khóa.
- **Keywords:** Tập hợp các bước thực thi được định nghĩa để sử dụng lại trong nhiều test case.

3.4. Ưu điểm khi sử dụng Robot Framework

- **Dễ tiếp cận:** Cú pháp đơn giản giúp người mới nhanh chóng làm quen.
- **Tính linh hoạt:** Hỗ trợ nhiều loại kiểm thử từ UI đến API.
- **Tái sử dụng tốt:** Từ khóa và biến giúp giảm thiểu lặp lại trong kịch bản.

- **Cộng đồng lớn:** Tài liệu phong phú và sự hỗ trợ từ cộng đồng giúp giải quyết vấn đề nhanh chóng.

3.5. Nhược điểm của Robot Framework

Bên cạnh các lợi ích, Robot Framework cũng có một số hạn chế:

- **Hiệu suất hạn chế:** Không tối ưu cho các bài kiểm thử phức tạp hoặc yêu cầu tốc độ cao.
- **Phụ thuộc thư viện:** Một số chức năng nâng cao đòi hỏi kỹ năng lập trình để phát triển thêm.
- **Tùy chỉnh khó:** Xử lý các trường hợp đặc biệt có thể phức tạp và tốn thời gian.

3.6. Ứng dụng thực tiễn của Robot Framework

Robot Framework được sử dụng rộng rãi trong nhiều lĩnh vực, từ doanh nghiệp nhỏ đến tổ chức lớn. Một số ví dụ bao gồm:

- Tự động hóa kiểm thử giao diện người dùng cho các ứng dụng web.
- Kiểm thử API trong các hệ thống phân tán.
- Hỗ trợ quy trình kiểm thử trong môi trường Agile hoặc DevOps.

CHƯƠNG 4: KẾT LUẬN

4.1 KẾT QUẢ THỰC HIỆN

4.2 HƯỚNG PHÁT TRIỂN

Sau quá trình thực hiện kiểm thử website tin tức và phân tích kết quả, nhóm nhận thấy rằng mặc dù website đã đáp ứng được các yêu cầu cơ bản, vẫn còn nhiều tiềm năng để cải thiện chất lượng, hiệu suất, bảo mật và trải nghiệm người dùng. Để đạt được những mục tiêu này, đồng thời tối ưu hóa quy trình kiểm thử trong tương lai, nhóm đề xuất các hướng phát triển chiến lược sau đây:

1. Tối ưu hóa và Mở rộng Chiến lược Kiểm thử Tự động (Automation Testing)

- **Mục tiêu:** Giảm thiểu công sức kiểm thử thủ công cho các tác vụ lặp đi lặp lại, tăng tốc độ phản hồi về chất lượng sản phẩm, và đảm bảo độ ổn định của các chức năng cốt lõi qua các bản cập nhật.
- **Hướng phát triển cụ thể:**
 - **Mở rộng phạm vi bao phủ (Test Coverage):** Không chỉ dừng lại ở các luồng cơ bản, cần xây dựng các bộ kiểm thử tự động cho các kịch bản phức tạp hơn như:
 - Quy trình quản lý tài khoản người dùng nâng cao (đổi mật khẩu, cập nhật thông tin, quản lý đăng ký nhận tin, v.v.).
 - Quy trình đăng tải, biên tập và duyệt bài viết (nếu website có hệ thống CMS nội bộ cho phóng viên/biên tập viên).
 - Kiểm thử các tính năng cá nhân hóa nội dung dựa trên hành vi hoặc sở thích người dùng (nếu có).
 - Kiểm thử tích hợp với các dịch vụ bên thứ ba (ví dụ: đăng nhập qua mạng xã hội, hệ thống thanh toán cho gói premium, v.v.).
 - **Tối ưu hóa và Chuẩn hóa Test Script:**
 - Áp dụng các mẫu thiết kế (design patterns) phổ biến trong kiểm thử tự động như Page Object Model (POM) để tăng tính tái sử dụng, dễ đọc và dễ bảo trì của mã kiểm thử.

- Thực hiện refactor định kỳ các test script để loại bỏ mã dư thừa, cải thiện hiệu suất thực thi và tăng độ tin cậy (giảm flaky tests).
- **Đẩy mạnh Kiểm thử API Tự động:** Tập trung kiểm thử vào tầng giao diện lập trình ứng dụng (API) vì đây là nơi xử lý logic nghiệp vụ chính. Lợi ích bao gồm:
 - Phát hiện lỗi sớm hơn trong chu trình phát triển, trước khi chúng ảnh hưởng đến giao diện người dùng.
 - Tốc độ thực thi nhanh hơn đáng kể so với kiểm thử UI.
 - Ít bị ảnh hưởng bởi các thay đổi nhỏ về giao diện.
- **Tích hợp Kiểm thử Hồi quy Giao diện (Visual Regression Testing):** Sử dụng các công cụ chuyên dụng (ví dụ: Percy, Appliflow, hoặc các thư viện mã nguồn mở) để tự động chụp ảnh màn hình giao diện và so sánh với phiên bản chuẩn (baseline). Điều này giúp phát hiện nhanh chóng các lỗi không mong muốn về bố cục, font chữ, màu sắc sau mỗi lần thay đổi code, đảm bảo tính nhất quán về mặt thẩm mỹ.

2. Chú trọng và Hệ thống hóa Kiểm thử Hiệu năng và Khả năng chịu tải (Performance & Load Testing)

- **Mục tiêu:** Đảm bảo website luôn hoạt động nhanh chóng, ổn định và đáp ứng tốt ngay cả khi có lượng truy cập lớn, đặc biệt trong các sự kiện tin tức nóng hoặc chiến dịch marketing.
- **Hướng phát triển cụ thể:**
 - **Lên kế hoạch và Thực hiện Định kỳ:** Xây dựng một lịch trình kiểm thử hiệu năng (đo thời gian tải trang, tốc độ phản hồi của máy chủ, tài nguyên tiêu thụ phía client) và kiểm thử chịu tải (stress test để tìm điểm giới hạn, load test để kiểm tra hoạt động dưới tải dự kiến, spike test để kiểm tra khả năng phục hồi sau khi tải tăng đột ngột) một cách thường xuyên, không chỉ trước các sự kiện lớn mà còn sau mỗi bản phát hành quan trọng.
 - **Mô phỏng Kịch bản Người dùng Đa dạng và Thực tế:** Thiết kế các kịch bản kiểm thử mô phỏng sát với hành vi người dùng thực tế, bao gồm:

- Sử dụng các cấu hình mạng khác nhau (Wifi tốc độ cao, Wifi yếu, 3G, 4G).
- Thực hiện trên nhiều loại thiết bị (desktop, laptop, tablet, mobile) và hệ điều hành/trình duyệt phổ biến.
- Kết hợp nhiều hành động người dùng khác nhau (đọc bài, xem video, bình luận, tìm kiếm, v.v.).
- **Phân tích Kết quả Chuyên sâu và Tối ưu hóa:** Không chỉ ghi nhận kết quả pass/fail, cần sử dụng các công cụ phân tích hiệu năng phía client (Google Lighthouse, WebPageTest, trình duyệt DevTools) và phía server (Application Performance Monitoring - APM tools, logs) để xác định chính xác các "điểm nghẽn" (bottlenecks) như truy vấn cơ sở dữ liệu chậm, mã nguồn không hiệu quả, cấu hình máy chủ chưa tối ưu, thiếu cơ chế caching, hình ảnh/video chưa nén. Từ đó, đề xuất các giải pháp cụ thể cho đội ngũ phát triển và hạ tầng.

3. Tăng cường và Chủ động trong Kiểm thử Bảo mật (Security Testing)

- **Mục tiêu:** Bảo vệ website khỏi các cuộc tấn công mạng, đảm bảo an toàn dữ liệu người dùng (nếu có) và duy trì uy tín của trang tin tức.
- **Hướng phát triển cụ thể:**
 - **Thực hiện Kiểm thử Xuyên nhập (Penetration Testing) Định kỳ:** Thuê các chuyên gia bảo mật bên ngoài hoặc sử dụng đội ngũ nội bộ có chuyên môn để thực hiện các cuộc kiểm thử xuyên nhập định kỳ (ví dụ: hàng năm hoặc sau các thay đổi lớn về kiến trúc). Mục tiêu là mô phỏng các cuộc tấn công thực tế để phát hiện các lỗ hổng bảo mật tiềm ẩn, đặc biệt là các lỗ hổng trong danh sách OWASP Top 10 (như SQL Injection, Cross-Site Scripting - XSS, Broken Authentication).
 - **Tích hợp Quét Lỗ hổng Tự động vào Quy trình Phát triển:** Sử dụng các công cụ Static Application Security Testing (SAST - quét mã nguồn) và Dynamic Application Security Testing (DAST - quét ứng dụng đang chạy) tích hợp vào quy trình CI/CD. Việc này giúp phát hiện sớm các vấn đề bảo mật ngay từ giai đoạn code hoặc build, giảm chi phí khắc phục.

- **Rà soát và Đảm bảo Tuân thủ Quy định:** Kiểm tra và đảm bảo website tuân thủ các quy định pháp luật hiện hành về bảo vệ dữ liệu cá nhân (như Nghị định 13/2023/NĐ-CP của Việt Nam về bảo vệ dữ liệu cá nhân, hoặc GDPR nếu có độc giả quốc tế), đặc biệt là các khía cạnh liên quan đến thu thập, lưu trữ và xử lý thông tin người dùng (ví dụ: qua form bình luận, đăng ký nhận tin).

4. Nâng cao Trải nghiệm Người dùng thông qua Kiểm thử Khả năng Sử dụng và Tiếp cận (Usability & Accessibility Testing)

- **Mục tiêu:** Đảm bảo website không chỉ hoạt động đúng chức năng mà còn dễ dàng, thuận tiện và thân thiện cho mọi đối tượng người dùng, bao gồm cả người khuyết tật.
- **Hướng phát triển cụ thể:**
 - **Tổ chức User Testing Thường xuyên:** Định kỳ mời các nhóm người dùng đại diện cho độc giả mục tiêu (khác nhau về độ tuổi, trình độ công nghệ, thói quen đọc tin) tham gia các buổi kiểm thử khả năng sử dụng. Quan sát trực tiếp cách họ tương tác với website, thực hiện các tác vụ cụ thể (tìm tin, đọc bài, chia sẻ, bình luận) và thu thập phản hồi chi tiết về những khó khăn, vướng mắc hoặc điểm chưa hài lòng.
 - **Kiểm tra và Cải thiện Tính Tiếp cận (Accessibility):** Chủ động kiểm tra mức độ tuân thủ của website với các tiêu chuẩn quốc tế về tiếp cận nội dung web (WCAG - Web Content Accessibility Guidelines), hướng tới mức AA hoặc cao hơn. Điều này bao gồm việc đảm bảo:
 - Người dùng có thể điều hướng bằng bàn phím.
 - Tương phản màu sắc đủ tốt.
 - Hình ảnh có văn bản thay thế (alt text).
 - Cấu trúc trang web rõ ràng (sử dụng đúng thẻ heading).
 - Tương thích với các công nghệ hỗ trợ như trình đọc màn hình (screen readers).
 - **Áp dụng A/B Testing cho Tối ưu hóa Giao diện/Luồng:** Đối với các thay đổi quan trọng về thiết kế giao diện (ví dụ: vị trí nút kêu gọi hành động, bố cục trang chủ, cách hiển thị tin liên

quan) hoặc luồng chức năng (ví dụ: quy trình đăng ký), triển khai A/B testing để so sánh hiệu quả của các phiên bản khác nhau dựa trên dữ liệu thực tế (tỷ lệ nhấp chuột, thời gian trên trang, tỷ lệ hoàn thành tác vụ).

5. Tích hợp Sâu hơn Hoạt động Kiểm thử vào Quy trình CI/CD và Văn hóa DevOps

- **Mục tiêu:** Xây dựng một quy trình phát triển và triển khai phần mềm liên tục, nơi kiểm thử là một phần không thể tách rời, giúp cung cấp phản hồi nhanh chóng và đảm bảo chất lượng ở mọi giai đoạn.
- **Hướng phát triển cụ thể:**
 - **Thực thi "Shift-left Testing":** Đẩy mạnh việc thực hiện các hoạt động kiểm thử sớm hơn trong vòng đời phát triển phần mềm. Khuyến khích và hỗ trợ nhà phát triển viết unit test và integration test. Tích hợp chặt chẽ các bộ kiểm thử tự động (unit, integration, API, UI) vào pipeline Tích hợp Liên tục (Continuous Integration - CI) để chúng được thực thi tự động sau mỗi lần commit code.
 - **Thiết lập "Quality Gates" trong Pipeline:** Định nghĩa các tiêu chí chất lượng (ví dụ: tỷ lệ bao phủ mã nguồn tối thiểu, tất cả các test case quan trọng phải pass) làm "cổng kiểm soát" trong pipeline CI/CD. Chỉ khi các tiêu chí này được đáp ứng, mã nguồn mới được phép merge vào nhánh chính hoặc triển khai lên các môi trường tiếp theo (staging, production).

6. Nghiên cứu và Thử nghiệm Ứng dụng Trí tuệ Nhân tạo (AI) và Học máy (ML) trong Kiểm thử

- **Mục tiêu:** Khám phá tiềm năng của các công nghệ mới để làm cho quy trình kiểm thử trở nên thông minh hơn, hiệu quả hơn và có khả năng dự đoán tốt hơn.
- **Hướng phát triển cụ thể (mang tính dài hạn và nghiên cứu):**
 - **Phân tích và Dự đoán Lỗi:** Sử dụng các thuật toán ML để phân tích dữ liệu lịch sử về lỗi, các thay đổi mã nguồn, và kết quả kiểm thử nhằm dự đoán các module hoặc khu vực code có

nguy cơ cao chứa lỗi, từ đó giúp tập trung nỗ lực kiểm thử hiệu quả hơn.

- **Tối ưu hóa hoặc Tự động Tạo Test Case:** Nghiên cứu các công cụ dựa trên AI có khả năng phân tích yêu cầu hoặc hành vi người dùng để tự động đề xuất hoặc tạo ra các test case mới, hoặc giúp tối ưu hóa bộ test case hiện có bằng cách xác định các trường hợp dư thừa hoặc thiếu sót.
- **Phát hiện Bất thường Tự động (Anomaly Detection):** Áp dụng AI/ML để giám sát log hệ thống hoặc các chỉ số hiệu năng trong quá trình kiểm thử (đặc biệt là kiểm thử hiệu năng hoặc kiểm thử dài hạn) nhằm tự động phát hiện các hành vi bất thường có thể là dấu hiệu của lỗi tiềm ẩn mà các test case truyền thống bỏ sót.

Kết luận:

Việc đầu tư vào các hướng phát triển trên không chỉ đơn thuần là cải thiện hoạt động kiểm thử. Đây là một chiến lược toàn diện nhằm nâng cao chất lượng tổng thể của website tin tức [Tên Website Cụ Thể] trên mọi phương diện: từ tính năng, hiệu suất, bảo mật đến trải nghiệm người dùng cuối.

Bằng cách áp dụng tự động hóa một cách thông minh, chú trọng vào hiệu năng và bảo mật, lắng nghe người dùng, tích hợp chặt chẽ vào quy trình phát triển hiện đại và sẵn sàng khám phá công nghệ mới, website sẽ có nền tảng vững chắc để phát triển bền vững, giữ chân độc giả và khẳng định vị thế trong thị trường tin tức cạnh tranh. Quy trình kiểm thử cũng sẽ trở nên hiệu quả hơn, tiết kiệm chi phí và thời gian, đồng thời đóng góp tích cực hơn vào thành công chung của sản phẩm.

TÀI LIỆU THAM KHẢO

- [1] React team. Getting Started, <<https://reactjs.org/docs/getting-started.html>>
- [2] Sequelize team. Associations, <<https://sequelize.org/docs/v6/core-concepts/assocs/>>
- [3] ExpressJS team. Routing, <<https://expressjs.com/en/guide/routing.html>>
- [4] Do Khanh Toan (2017). Giới thiệu về ReactJS - Phần I (Các khái niệm cơ bản), <<https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7>>