ĐIỂM SỐ:

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH

KHOA: CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN

TIẾN HÀNH PHÂN TÍCH YÊU CẦU, LẬP KẾ HOẠCH KIỂM THỬ, VIẾT KỊCH BẢN VÀ THỰC HIỆN KIỂM THỬ UI VÀ API CỦA WEBSITE DEV.TO (DEVTO)

MÔN: ĐẢM BẢO CHẤT LƯỢNG VÀ KIỂM THỬ PHẦN MỀM

Ngành: **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **THS. NGUYỄN PHƯỚC VĨNH LỘC**

Sinh viên thực hiện: **NGUYỄN TRẦN HOÀNG THỊNH**

MSSV: **22140341**

Lớp học phần: **020100137102- 221402**

TP. Hồ Chí Minh, 2 tháng 4 năm 2025

Khoa/Viện: Công Nghệ Thông Tin

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN
TIỂU LUẬN MÔN: ĐẢM BẢO CHẤT LƯỢNG VÀ KIỂM THỬ PHẦN MỀM

1. **Họ và tên sinh viên: Nguyễn Trần Hoàng Thịnh**
2. **Tên đề tài**: **Tiến hành phân tích yêu cầu, lập kế hoạch kiểm thử, viết kịch bản và thực hiện kiểm thử UI và API của website dev.to (devto)**
3. **Nhận xét**:

   a) Những kết quả đạt được:

   .....

   .....

   .....

   .....

   .....

   b) Những hạn chế:

   .....

   .....

   .....

   .....

   .....

4. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

   Sinh viên: Nguyễn Trần Hoàng Thịnh

   Điểm số: ……. Điểm chữ: ……

TP. HCM, ngày 02 tháng 04 năm 2025

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đề tài tiểu luận: “Tiến Hành Phân Tích Yêu Cầu, Lập Kế Hoạch Kiểm Thử, Viết Kịch Bản Và Thực Hiện Kiểm Thử Ui Và Api Của Website Dev.To (Devto)” do Nguyễn Trần Hoàng Thịnh tìm hiểu và thực hiện. Kết quả bài làm của đề tài “Tiến Hành Phân Tích Yêu Cầu, Lập Kế Hoạch Kiểm Thử, Viết Kịch Bản Và Thực Hiện Kiểm Thử Ui Và Api Của Website Dev.To (Devto)” là trung thực và không sao chép từ bất kì bài tập của các cá nhân khác.

TP. HCM, … tháng 04 năm 2025

Sinh viên cam đoan

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Phước Vĩnh Lộc. Trong quá trình học tập và tìm hiểu môn "Đảm Bảo Chất Lượng Và Kiểm Thử Phần Mềm", em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của cô. Thầy đã giúp em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài tiểu luận về đề tài"Tiến Hành Phân Tích Yêu Cầu, Lập Kế Hoạch Kiểm Thử, Viết Kịch Bản Và Thực Hiện Kiểm Thử Ui Và Api Của Website Dev.To (Devto)".

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của thầy để bài tiểu luận của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1: PHÂN TÍCH YÊU CẦU

1.1. Giới thiệu Website Dev.to và API Forem V1

1.1.1. Mô tả ngắn gọn

Dev.to là một nền tảng cộng đồng trực tuyến dành cho các lập trình viên, nơi họ có thể chia sẻ kiến thức, kinh nghiệm và ý tưởng thông qua các bài viết (articles), thảo luận, và tương tác với nhau. Nền tảng này được xây dựng dựa trên phần mềm mã nguồn mở Forem, cho phép các tổ chức hoặc cá nhân tạo ra các cộng đồng tương tự. API Forem V1 là giao diện lập trình ứng dụng (API) chính thức của hệ thống Forem, cung cấp khả năng truy cập và quản lý các tài nguyên như bài viết, người dùng, tổ chức, và video trên Dev.to hoặc các phiên bản Forem khác.

1.1.2. Các chức năng chính (Người dùng & Hệ thống/API)

Người dùng (User-facing):

- Đăng ký và đăng nhập: Tạo tài khoản và truy cập hệ thống bằng email hoặc tài khoản mạng xã hội (GitHub, Twitter).
- Đọc và viết bài: Xem danh sách bài viết, lọc theo tag hoặc tổ chức, tạo bài viết mới dưới dạng markdown.
- Tương tác: Thích (like), bình luận (comment), và chia sẻ bài viết.
- Quản lý cá nhân: Cập nhật thông tin profile, theo dõi người dùng/tổ chức khác.
- Tìm kiếm: Tìm bài viết theo từ khóa, tag, hoặc tác giả.

Hệ thống/API (System/API-facing):

- Quản lý bài viết:
- Lấy danh sách bài viết công khai (/articles, /articles/latest).
- Lấy bài viết của người dùng cụ thể (/articles/me/*).
- Lọc bài viết theo tag (/articles?tag=...), username (/articles?username=...), hoặc tổ chức.
- Cập nhật (PUT /articles/:id) và hủy xuất bản bài viết (PUT /articles/:id/unpublish)

Quản lý người dùng:

- Lấy thông tin người dùng (/users/:id).
- Mời người dùng mới (admin, POST /admin/users).
- Quản lý trạng thái người dùng (unpublish, suspend: PUT /users/:id/*).
- Quản lý tổ chức: Lấy chi tiết tổ chức (/organizations/:username) và bài viết của tổ chức.
- Video: Lấy danh sách bài viết có video (/videos).

1.1.3. Yêu cầu phi chức

Dựa trên tài liệu API và mục đích của Dev.to, một số yêu cầu phi chức năng có thể được suy ra:

- Hiệu suất: API phải trả về dữ liệu trong thời gian ngắn (thường dưới 2 giây) ngay cả với số lượng lớn bài viết, nhờ cơ chế phân trang (page, per_page).

Bảo mật:

- Các hành động nhạy cảm (như unpublish bài viết, suspend người dùng) yêu cầu xác thực qua api-key và quyền admin/moderator.
- Dữ liệu cá nhân (ví dụ: bài viết chưa xuất bản) chỉ hiển thị với người dùng được xác thực.

Khả năng mở rộng:

- Hỗ trợ phân trang với số lượng mục mỗi trang tùy chỉnh (mặc định 30, có thể override bằng API_PER_PAGE_MAX).
- API hoạt động ổn định trên các cộng đồng Forem khác nhau (không chỉ Dev.to).
- Tính nhất quán: Dữ liệu trả về (như ngày giờ) tuân theo chuẩn RFC 3339 để đảm bảo khả năng tương thích.

1.1.4. Tổng quan về API (Xác thực, Định dạng, Tham số chung)

Xác thực:

- API sử dụng cơ chế xác thực dựa trên api-key được gửi qua header HTTP (api-key: <API Key>).
- Một số endpoint công khai (như /articles, /articles/latest) không yêu cầu xác thực, nhưng các hành động quản trị (unpublish, suspend, mời người dùng) bắt buộc cần api-key với quyền admin/moderator.

Định dạng:

- Request/Response: Định dạng JSON (Content-Type: application/json, Accept: application/json).
- Ngày giờ: Tuân theo chuẩn RFC 3339 (ví dụ: "published_timestamp": "2023-04-14T14:45:33Z").
- Cấu trúc phản hồi: Các endpoint trả về mảng (danh sách) hoặc object (chi tiết), với các trường như id, title, tag_list, user, v.v. được định nghĩa rõ ràng trong tài liệu.

Tham số chung:

- page: Số trang trong phân trang (mặc định: 1).
- per_page: Số lượng mục mỗi trang (mặc định: 30 cho bài viết, 24 cho video, có thể tùy chỉnh tối đa bằng biến môi trường API_PER_PAGE_MAX).
- tag: Lọc bài viết theo tag (ví dụ: javascript, webdev).
- username: Lọc bài viết theo người dùng/tổ chức hoặc lấy thông tin chi tiết.
- id: Xác định tài nguyên cụ thể (bài viết, người dùng).

Ví dụ endpoint cơ bản:

- GET {{baseUrl}}/articles?page=1&per_page=30: Lấy 30 bài viết đầu tiên.
- GET {{baseUrl}}/articles?tag=javascript: Lấy bài viết có tag javascript.
- PUT {{baseUrl}}/articles/:id/unpublish?note=<string>: Hủy xuất bản bài viết (yêu cầu api-key).

CHƯƠNG 2: THIẾT KẾ KỊCH BẢN KIỂM THỬ

2.1. Kịch bản kiểm thử Giao diện Người dùng (UI)

Phần này tập trung vào các chức năng người dùng tương tác trực tiếp trên giao diện website Dev.to. Các kịch bản được thiết kế dựa trên tài liệu Excel của bạn, kiểm tra các chức năng chính như đăng ký, xem bài viết, và quản lý trang cá nhân.

2.1.1. TC_UI_001: Đăng Ký Tài Khoản Thành Công

Mục tiêu: Đảm bảo người dùng có thể đăng ký tài khoản mới thành công trên Dev.to.

Mô tả: Người dùng truy cập trang đăng ký, nhập thông tin hợp lệ (name, username, email, password), và hoàn tất quy trình đăng ký bao gồm xác minh email.
Điều kiện tiên quyết:

- Người dùng chưa có tài khoản trên Dev.to.
- Truy cập website https://dev.to/ qua trình duyệt Edge (phiên bản 134.0.3124.9 trở lên).
- Kết nối internet ổn định.

Kịch bản:

1. Mở trình duyệt Edge và truy cập https://dev.to/.
2. Nhấn "Create Account" trên trang chủ.
3. Chọn "Sign up with Email" và nhập thông tin:
    - Name: "Thinh Nguyen"
    - Username: "ThinhNguyen"
    - Email: "thinhnguyen190304@gmail.com"
    - Password: "ThinhNguyentester1903"
    - Confirm Password: "ThinhNguyentester1903"
4. Nhấn "Sign up".
5. Xác minh email trong hộp thư và bỏ qua bước setup profile.

Kết quả mong đợi: Hệ thống tạo tài khoản thành công, hiển thị trang xác minh email, sau đó chuyển hướng về trang chủ sau khi bỏ qua setup profile.

2.1.2. TC_UI_002: Đăng Ký Tài Khoản Thất Bại (Email đã tồn tại)

Mục tiêu: Kiểm tra hệ thống hiển thị lỗi khi người dùng cố gắng đăng ký bằng email đã tồn tại.

Mô tả: Người dùng nhập thông tin đăng ký với email đã được sử dụng trước đó và kiểm tra thông báo lỗi.

Điều kiện tiên quyết:

- Đã có tài khoản đăng ký với email "thinhnguyen190304@gmail.com".
- Truy cập website https://dev.to/ qua trình duyệt Edge (phiên bản 134.0.3124.9 trở lên).
- Kết nối internet ổn định.

Kịch bản:

1. Mở trình duyệt Edge và truy cập https://dev.to/.
2. Nhấn "Create Account".
3. Chọn "Sign up with Email" và nhập thông tin:
    - Name: "Thinh Nguyen"
    - Username: "ThinhNguyen"
    - Email: "thinhnguyen190304@gmail.com"
    - Password: "ThinhNguyentester1903"
    - Confirm Password: "ThinhNguyentester1903"
4. Nhấn "Sign up".

Kết quả mong đợi: Hệ thống hiển thị thông báo lỗi "Email has already been taken" và không tạo tài khoản mới.

2.1.3. TC_UI_003: Hiển Thị Danh Sách Bài Viết Theo Tag

Mục tiêu: Kiểm tra khả năng hiển thị danh sách bài viết khi chọn một tag từ sidebar.

Mô tả: Người dùng chọn tag "webdev" từ danh sách tag phổ biến và xem danh sách bài viết liên quan.

Điều kiện tiên quyết:

- Truy cập website https://dev.to/ qua trình duyệt Edge (phiên bản 134.0.3124.9 trở lên).
- Sidebar hiển thị mục "Tags" với danh sách tag phổ biến, bao gồm "webdev".
- Có bài viết gắn tag "webdev" trong hệ thống.

Kịch bản:

1. Mở trình duyệt Edge và truy cập https://dev.to/.
2. Trên trang chủ, trong sidebar bên trái, nhấn vào tag "webdev".
3. Kiểm tra danh sách bài viết hiển thị.

Kết quả mong đợi: Chuyển hướng đến trang bài viết của tag "webdev", hiển thị danh sách bài viết với tiêu đề, tác giả, ngày đăng, và chứa tag "webdev".

2.1.4. TC_UI_004: Xem Trang Cá Nhân (Profile)

Mục tiêu: Đảm bảo người dùng có thể truy cập và xem thông tin trên trang cá nhân của mình.

Mô tả: Người dùng đã đăng nhập truy cập trang cá nhân và kiểm tra thông tin hiển thị.

Điều kiện tiên quyết:

- Người dùng đã đăng nhập với tài khoản "ThinhNguyen".
- Truy cập website https://dev.to/ qua trình duyệt Edge (phiên bản 134.0.3124.9 trở lên).

Kịch bản:

1. Mở trình duyệt Edge và truy cập https://dev.to/.
2. Nhấn vào biểu tượng hồ sơ ở góc trên bên phải.
3. Trong menu dropdown, nhấn vào tên người dùng "ThinhNguyen".
4. Kiểm tra thông tin trên trang cá nhân và danh sách bài viết.

Kết quả mong đợi: Chuyển hướng đến trang cá nhân của "ThinhNguyen", hiển thị thông tin như tên, ảnh đại diện, bio (nếu có), và danh sách bài viết (hoặc "No posts yet" nếu chưa có).

2.1.5. TC_UI_005: Chỉnh Sửa Trang Cá Nhân

Mục tiêu: Kiểm tra khả năng chỉnh sửa thông tin trên trang cá nhân của người dùng.

Mô tả: Người dùng đã đăng nhập cập nhật bio và ảnh đại diện trên trang cá nhân. Điều kiện tiên quyết:

- Người dùng đã đăng nhập với tài khoản "ThinhNguyen".
- Truy cập website https://dev.to/ qua trình duyệt Edge (phiên bản 134.0.3124.9 trở lên).

Kịch bản:

1. Mở trình duyệt Edge và truy cập https://dev.to/.
2. Nhấn vào biểu tượng hồ sơ ở góc trên bên phải, chọn "ThinhNguyen".
3. Trên trang cá nhân, nhấn "Edit Profile".
4. Cập nhật bio thành "Software Developer".
5. Tải lên ảnh đại diện mới từ file "OIP.jpg".
6. Nhấn "Save" để lưu thay đổi.
7. Quay lại trang cá nhân để kiểm tra.

Kết quả mong đợi: Hiển thị thông báo "Profile updated successfully", bio cập nhật thành "Software Developer", ảnh đại diện mới được áp dụng.

2.2. Kịch bản kiểm thử API

Phần này dựa trên Postman collection "Forem API Tests" mà bạn cung cấp, tập trung vào các chức năng chính của API Forem V1. Các kịch bản kiểm tra khả năng truy xuất và xử lý dữ liệu từ hệ thống.

2.2.1. TC_API_001: Kiểm thử Phân trang và Cấu trúc dữ liệu bài viết

- Mục tiêu: Đảm bảo API /articles trả về danh sách bài viết phân trang đúng và cấu trúc dữ liệu hợp lệ.
- Mô tả: Gửi request lấy danh sách bài viết với các tham số phân trang khác nhau và kiểm tra phản hồi.

- Điều kiện tiên quyết: Truy cập được API tại https://dev.to/api, không yêu cầu xác thực cho endpoint công khai.

Kịch bản:

1. Gửi GET request tới {{baseUrl}}/articles?page={{currentPage}}&per_page=30.
2. Thay đổi currentPage ngẫu nhiên từ 1 đến 5.
3. Kiểm tra cấu trúc dữ liệu trả về (id, title, tag_list, v.v.) và số lượng bài viết.
4. Thử các giá trị biên như page=-1 hoặc per_page=1000.

Kết quả mong đợi:

- Status 200, mảng bài viết ≤ 30 mục, không trùng ID giữa các trang.
- Giá trị biên: page=-1 trả về trang đầu tiên, per_page=1000 trả về tối đa số bài cho phép.

2.2.2. TC_API_002: Kiểm thử Lọc bài viết theo Tag

- Mục tiêu: Đảm bảo API /articles?tag=... trả về danh sách bài viết chứa tag yêu cầu.
- Mô tả: Gửi request lọc bài viết theo tag "javascript" và kiểm tra kết quả.
- Điều kiện tiên quyết: Có bài viết gắn tag "javascript" trong hệ thống.

Kịch bản:

1. Gửi GET request tới {{baseUrl}}/articles?tag=javascript.
2. Kiểm tra tất cả bài viết trả về chứa tag "javascript".
3. Thử tag không tồn tại như "nonexistenttag123".

Kết quả mong đợi:

- Status 200, mảng bài viết đều chứa tag "javascript".
- Tag không tồn tại trả về mảng rỗng.

2.2.3. TC_API_003: Kiểm thử Tìm kiếm/Lọc bài viết theo Tag hoặc Username

- Mục tiêu: Kiểm tra API trả về bài viết khớp với tag hoặc username yêu cầu.
- Mô tả: Gửi request lọc bài viết theo tag "webdev" hoặc username "ben".
- Điều kiện tiên quyết: Có bài viết gắn tag "webdev" hoặc của người dùng "ben".

Kịch bản:

1. Gửi GET request tới {{baseUrl}}/articles?tag=webdev hoặc {{baseUrl}}/articles?username=ben.
2. Kiểm tra bài viết trả về khớp tiêu chí.
3. Thử tag/username không tồn tại.

Kết quả mong đợi:

- Status 200, bài viết khớp với tag hoặc username.
- Tag/username không tồn tại trả về mảng rỗng.

2.2.4. TC_API_004: Kiểm thử Lấy bài viết theo Tổ chức

- Mục tiêu: Đảm bảo API /articles?username=... trả về bài viết của tổ chức.
- Mô tả: Gửi request lấy bài viết của tổ chức "dev" và kiểm tra dữ liệu.
- Điều kiện tiên quyết: Tổ chức "dev" tồn tại và có bài viết.

Kịch bản:

1. Gửi GET request tới {{baseUrl}}/articles?username=dev.
2. Kiểm tra bài viết có trường organization.username là "dev".
3. Thử tổ chức không tồn tại như "nonexistentorg123".

Kết quả mong đợi:

- Status 200, bài viết thuộc tổ chức "dev".
- Tổ chức không tồn tại trả về mảng rỗng.

2.2.5. TC_API_005: Kiểm thử Lấy chi tiết Tổ chức

- Mục tiêu: Đảm bảo API /organizations/{username} trả về thông tin chi tiết tổ chức hoặc lỗi khi không tồn tại.
- Mô tả: Gửi request lấy chi tiết tổ chức "dev" và kiểm tra phản hồi.
- Điều kiện tiên quyết: Tổ chức "dev" tồn tại trong hệ thống.

Kịch bản:

1. Gửi GET request tới {{baseUrl}}/organizations/dev.
2. Kiểm tra cấu trúc dữ liệu (username, name, slug, v.v.).
3. Thử tổ chức không tồn tại như "nonexistentorg123".

Kết quả mong đợi:

- Status 200, thông tin tổ chức "dev" đầy đủ.
- Tổ chức không tồn tại trả về status 404 với lỗi "not found".

CHƯƠNG 3: PHÁT TRIỂN TRƯỜNG HỢP KIỂM THỬ

3.1. Các trường hợp kiểm thử UI

Phần này mô tả tổng quát các trường hợp kiểm thử giao diện người dùng (UI) trên website Dev.to. Chi tiết các bước thực hiện, dữ liệu kiểm thử, và kết quả thực tế được ghi nhận trong tài liệu Excel, sẽ được minh họa bằng hình ảnh chèn vào báo cáo.

3.1.1. TC_UI_001: Đăng Ký Tài Khoản Thành Công

- Mục tiêu: Kiểm tra quy trình đăng ký tài khoản mới thành công.
- Mô tả chung: Người dùng truy cập trang đăng ký, nhập thông tin hợp lệ, xác minh email, và hoàn tất quy trình.
- Điều kiện tiên quyết: Người dùng chưa có tài khoản, truy cập Dev.to qua trình duyệt Edge, kết nối internet ổn định.
- Kết quả mong đợi: Tài khoản được tạo thành công, chuyển hướng về trang chủ sau khi xác minh email.

Test Case ID	TC_UI_DANGKY_001	**Test Case Description**	Kiểm tra quy trình đăng ký tài khoản mới thành công.				
Created By	Nguyễn Trần Hoàng Thịnh	**Reviewed By**	Nguyễn Trần Hoàng Thịnh		**Version**	0.0.1	
QA Tester's Log							
Tester's Name	Nguyễn Trần Hoàng Thịnh	**Date Tested**	March 31, 2025		**Test Case**	Pass	

S #	**Prerequisites:**	**S #**	**Test Data**
1	Người dùng chưa có tài khoản trên dev.to	1	Name = Thinh Nguyen
2	Truy cập website dev.to (https://dev.to/) trên trình duyệt Edge (Version 134.0.3124.9 trở lên)	2	Username = Thinh Nguyen
3	Kết nối internet ổn định	3	Email = thinhnguyen190304@gmail.com
		4	Password = ThinhNguyentester1903

Test Sce Kiểm tra quy trình đăng ký tài khoản mới thành công.

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail / Not executed / Suspended**
1	Mở trình duyệt Edge	Hiển thị giao diện trình duyệt Edge	Hiển thị giao diện trình duyệt Edge	Pass
2	Truy cập vào URL 'https://dev.to'	Hiển thị trang chủ của dev.to	Hiển thị trang chủ của dev.to	Pass
3	Nhấn "Create Account"	Hiển thị trang đăng ký tài khoản	Hiển thị trang đăng ký tài khoản	Pass
4	Nhấn "Sign up with Email"	Hiển thị trang đăng ký bằng Email	Hiển thị trang đăng ký bằng Email	Pass
5	Nhập liệu trường "Name" bằng Name	Nhập liệu thành công	Nhập liệu thành công	Pass
6	Nhập liệu trường "Username" bằng Username	Nhập liệu thành công	Nhập liệu thành công	Pass
7	Nhập liệu trường "Email" bằng Email	Nhập liệu thành công	Nhập liệu thành công	Pass
8	Nhập liệu trường "Password" bằng Password	Nhập liệu thành công	Nhập liệu thành công	Pass
9	Nhập liệu trường "Confirm Password" bằng Password	Nhập liệu thành công	Nhập liệu thành công	Pass
10	Nhấn "Sign up"	Chuyển hướng sang trang xác minh Email	Xác minh thành công	Pass
11	Tiến hành xác minh Email tại hộp thư mail	Chuyển hướng sang trang setup profile	Chuyển hướng sang trang xác minh Email	Pass
12	Tiến hành bỏ qua toàn bộ bước setup profile	Chuyển hướng về trang chủ	Chuyển hướng sang trang setup profile	Pass

TC_UI_001: Đăng Ký Tài Khoản Thành Công

3.1.2. TC_UI_002: Đăng Ký Tài Khoản Thất Bại (Email đã tồn tại)

- Mục tiêu: Kiểm tra thông báo lỗi khi đăng ký với email đã tồn tại.
- Mô tả chung: Người dùng nhập thông tin đăng ký với email đã được sử dụng trước đó và kiểm tra phản hồi của hệ thống.
- Điều kiện tiên quyết: Đã có tài khoản với email trùng, truy cập Dev.to qua Edge, kết nối internet ổn định.
- Kết quả mong đợi: Hiển thị thông báo lỗi "Email has already been taken".

Test Case ID	TC_UI_DANGKY_002	**Test Case Description**	Đăng Ký Tài Khoản Thất Bại (Email đã tồn tại)			
Created By	Nguyễn Trần Hoàng Thịnh	**Reviewed By**	Nguyễn Trần Hoàng Thịnh	**Version**	0.0.1	
QA Tester's Log						
Tester's Name	Nguyễn Trần Hoàng Thịnh	**Date Tested**	March 31, 2025	**Test Case**	Pass	

S #	**Prerequisites:**	**S #**	**Test Data**
1	Đã có một tài khoản trên dev.to được đăng ký với email đã sử dụng	1	Name = Thinh Nguyen
2	Truy cập website dev.to (https://dev.to/) trên trình duyệt Edge (Version 134.0.3124.9 trở lên)	2	Username = Thinh Nguyen
3	Kết nối internet ổn định	3	Email = thinhnguyen190304@gmail.com
		4	Password = ThinhNguyentester1903

Test Sce Đăng Ký Tài Khoản Thất Bại (Email đã tồn tại)

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail / Not executed / Suspended**
1	Mở trình duyệt Edge	Hiển thị giao diện trình duyệt Edge	Hiển thị giao diện trình duyệt Edge	Pass
2	Truy cập vào URL 'https://dev.to'	Hiển thị trang chủ của dev.to	Hiển thị trang chủ của dev.to	Pass
3	Nhấn "Create Account"	Hiển thị trang đăng ký tài khoản	Hiển thị trang đăng ký tài khoản	Pass
4	Nhấn "Sign up with Email"	Hiển thị trang đăng ký bằng Email	Hiển thị trang đăng ký bằng Email	Pass
5	Nhập liệu trường "Name" bằng Name	Nhập liệu thành công	Nhập liệu thành công	Pass
6	Nhập liệu trường "Username" bằng Username	Nhập liệu thành công	Nhập liệu thành công	Pass
7	Nhập liệu trường "Email" bằng Email	Nhập liệu thành công	Nhập liệu thành công	Pass
8	Nhập liệu trường "Password" bằng Password	Nhập liệu thành công	Nhập liệu thành công	Pass
9	Nhập liệu trường "Confirm Password" bằng Password	Nhập liệu thành công	Nhập liệu thành công	Pass
10	Nhấn "Sign up"	Hiển thị thông báo lỗi "Email has already been taken"	Hiển thị thông báo lỗi "Email has already been taken"	Pass

TC_UI_002: Đăng Ký Tài Khoản Thất Bại (Email đã tồn tại)

3.1.3. TC_UI_003: Hiển Thị Danh Sách Bài Viết Theo Tag

- Mục tiêu: Kiểm tra khả năng hiển thị danh sách bài viết theo tag.
- Mô tả chung: Người dùng chọn một tag từ sidebar và xem danh sách bài viết liên quan.
- Điều kiện tiên quyết: Truy cập Dev.to qua Edge, sidebar có tag phổ biến, tồn tại bài viết với tag được chọn.
- Kết quả mong đợi: Hiển thị danh sách bài viết khớp với tag đã chọn.

Test Case ID	TC_UI_DANHSACBAIVIET_003	**Test Case Description**	Hiển Thị Danh Sách Bài Viết Theo Tag			
Created By	Nguyễn Trần Hoàng Thịnh	**Reviewed By**	Nguyễn Trần Hoàng Thịnh	**Version**	0.0.1	
QA Tester's Log						
Tester's Name	Nguyễn Trần Hoàng Thịnh	**Date Tested**	March 31, 2025	**Test Case (Pass/Fail/Not**	Pass	

S #	Prerequisites:	S #	Test Data
1	Truy cập website dev.to (https://dev.to/) trên trình duyệt Edge (Version 134.0.3124.9 trở lên)	1	Tag: "webdev"
2	Sidebar bên trái hiển thị mục "Tags" với danh sách tag phổ biến	2	
3	Có bài viết gắn tag "webdev"	3	
		4	

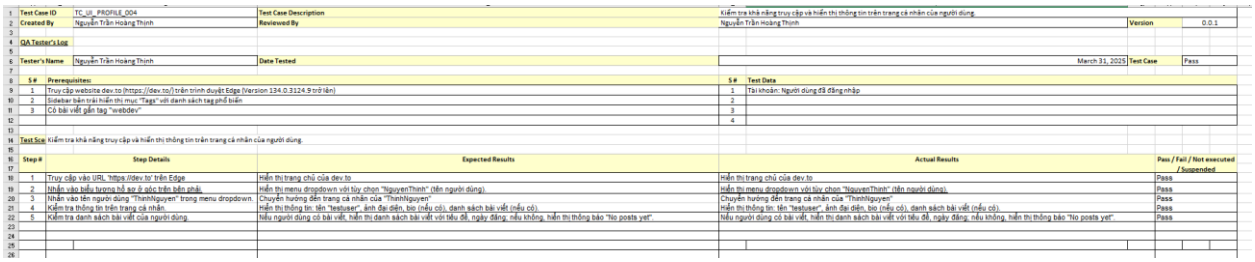
Test Sce Hiển Thị Danh Sách Bài Viết Theo Tag

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Truy cập vào URL 'https://dev.to' trên Edge	Hiển thị trang chủ của dev.to	Hiển thị trang chủ của dev.to	Pass
2	Trang chủ ấn vào tag trên sidebar	Chuyển hướng đến trang tag	Hiển thị trang tag	Pass
3	Trong trang "Tags", nhấn vào tag "webdev".	Chuyển hướng đến trang bài viết của tag "webdev".	Chuyển hướng đến trang bài viết của tag "webdev".	Pass
4	Kiểm tra danh sách bài viết.	Hiển thị bài viết với tiêu đề, tác giả, ngày đăng, có tag "webdev".	Hiển thị bài viết với tiêu đề, tác giả, ngày đăng, có tag "webdev".	Pass

TC_UI_003: Hiển Thị Danh Sách Bài Viết Theo Tag

3.1.4. TC_UI_004: Xem Trang Cá Nhân (Profile)

- Mục tiêu: Kiểm tra khả năng truy cập và hiển thị thông tin trang cá nhân.
- Mô tả chung: Người dùng đã đăng nhập truy cập trang cá nhân từ menu dropdown và xem thông tin.
- Điều kiện tiên quyết: Người dùng đã đăng nhập, truy cập Dev.to qua Edge.
- Kết quả mong đợi: Hiển thị thông tin cá nhân (tên, ảnh, bio, bài viết) chính xác.

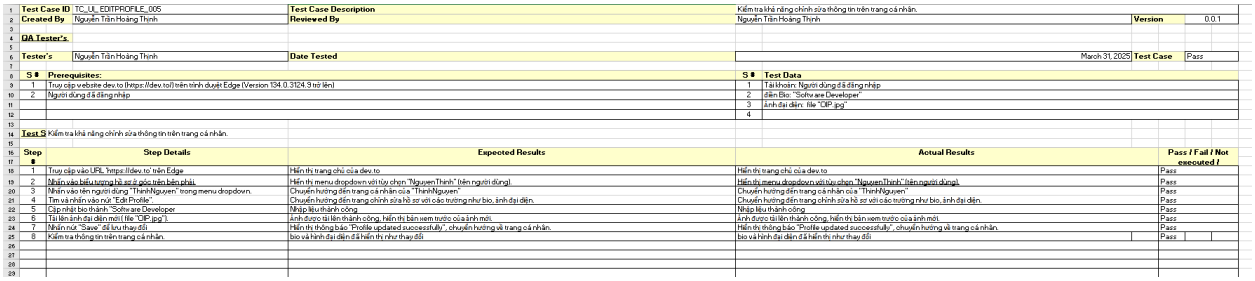


TC_UI_004: Xem Trang Cá Nhân (Profile)

3.1.5. TC_UI_005: Chỉnh Sửa Trang Cá Nhân

- Mục tiêu: Kiểm tra khả năng chỉnh sửa thông tin trang cá nhân.
- Mô tả chung: Người dùng cập nhật bio và ảnh đại diện trên trang cá nhân, sau đó lưu thay đổi.
- Điều kiện tiên quyết: Người dùng đã đăng nhập, truy cập Dev.to qua Edge.

- Kết quả mong đợi: Thông tin được cập nhật thành công, hiển thị đúng trên trang cá nhân.



Test Case ID	TC_UI_EDITPROFILE_005	Test Case Description	Kiểm tra khả năng chỉnh sửa thông tin trên trang cá nhân.		
Created By	Nguyễn Trần Hoàng Thịnh	Reviewed By	Nguyễn Trần Hoàng Thịnh	Version	0.0.1
QA Tester's					
Tester's	Nguyễn Trần Hoàng Thịnh	Date Tested	March 31, 2025	Test Case	Pass

S #	Prerequisites:	S #	Test Data
1	Truy cập website dev.to (https://dev.to/) trên trình duyệt Edge (Version 134.0.3124.9 trở lên)	1	Tài khoản: Người dùng đã đăng nhập
2	Người dùng đã đăng nhập	2	điền Bio: "Softw are Developer"
		3	ảnh đại diện: file "DP.jpg"
		4	

Test S Kiểm tra khả năng chỉnh sửa thông tin trên trang cá nhân.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /
1	Truy cập vào URL "https://dev.to" trên Edge	Hiển thị trang chủ của dev.to	Hiển thị trang chủ của dev.to	Pass
2	Nhấn vào biểu tượng hồ sơ ở góc trên bên phải.	Hiển thị menu dropdown với tùy chọn "Nguyen Thinh" (tên người dùng).	Hiển thị menu dropdown với tùy chọn "Nguyen Thinh" (tên người dùng).	Pass
3	Nhìn vào tên người dùng "ThinhNguyen" trong menu dropdown.	Chuyển hướng đến trang cá nhân của "ThinhNguyen".	Chuyển hướng đến trang cá nhân của "ThinhNguyen".	Pass
4	Tìm và nhấn vào nút "Edit Profile".	Chuyển hướng đến trang chỉnh sửa hồ sơ với các trường như bio, ảnh đại diện.	Chuyển hướng đến trang chỉnh sửa hồ sơ với các trường như bio, ảnh đại diện.	Pass
5	Cập nhật bio thành: "Softw are Developer"	Nhập liệu thành công.	Nhập liệu thành công.	Pass
6	Tải lên ảnh đại diện mới (file "DP.jpg").	Ảnh được tải lên thành công, hiển thị bản xem trước của ảnh mới.	Ảnh được tải lên thành công, hiển thị bản xem trước của ảnh mới.	Pass
7	Nhấn nút "Save" để lưu thay đổi.	Hiển thị thông báo "Profile updated successfully", chuyển hướng về trang cá nhân.	Hiển thị thông báo "Profile updated successfully", chuyển hướng về trang cá nhân.	Pass
8	Kiểm tra thông tin trên trang cá nhân.	bio và hình đại diện đã hiển thị như thay đổi	bio và hình đại diện đã hiển thị như thay đổi	Pass

TC_UI_005: Chỉnh Sửa Trang Cá Nhân

3.2. Các trường hợp kiểm thử API

3.2.1. TC_API_001: Kiểm thử Phân trang và Cấu trúc dữ liệu bài viết

- **Test Case ID:** TC_API_001
- **Mục tiêu:** Xác minh rằng API /articles trả về mã trạng thái thành công (200 OK) và phần thân phản hồi là một mảng JSON khi truy cập trang đầu tiên.
- **Điều kiện tiên quyết:**
  - Truy cập được API tại {{baseUrl}} (ví dụ: https://dev.to/api).
  - Pre-request Script được thực thi để thiết lập biến môi trường currentPage thành "1".
- **Dữ liệu kiểm thử:**
  - page: Giá trị được lấy từ biến môi trường {{currentPage}}, được đặt thành "1" bởi Pre-request Script.

Step #	Step Details	Expected Results
1	Thực thi Pre-request Script.	Biến môi trường currentPage được đặt thành giá trị "1"
2	Gửi yêu cầu GET tới {{baseUrl}}/articles?page={{currentPage}}.	API trả về mã trạng thái 200 OK

3.2.2. TC_API_002: Kiểm thử Lọc bài viết theo Tag

- Test Case ID: TC_API_002
- Mục tiêu: Xác minh rằng API /articles với tham số tag trả về mã trạng thái thành công (200 OK) và nếu có bài viết được trả về, chúng phải chứa tag được chỉ định.
- Điều kiện tiên quyết:
    - Truy cập được API tại {{baseUrl}} (ví dụ: https://dev.to/api).
    - Biến Postman (Environment hoặc Collection) currentTag được thiết lập với giá trị tag muốn kiểm tra (ví dụ: "javascript").
- Dữ liệu kiểm thử:
    - tag: Giá trị được lấy từ biến {{currentTag}}.

Các bước thực hiện và Kết quả mong đợi:

Step #	Step Details	Expected Results
1	Gửi yêu cầu GET tới {{baseUrl}}/articles?tag={{currentTag}}	Trả về mã trạng thái 200 OK (được kiểm tra trong pm.test("Mã trạng thái là 200")). Nếu mảng phản hồi không rỗng (jsonData.length > 0), mỗi bài viết trong mảng phải chứa giá trị của biến {{currentTag}} trong trường tag_list. (Được kiểm tra bởi test script: pm.test("Bài viết có tag"))

3.2.3. TC_API_003: Kiểm thử Tìm kiếm/Lọc bài viết theo Tag hoặc Username

- Test Case ID: TC_API_003
- Mục tiêu: Xác minh rằng API /articles trả về mã trạng thái thành công (200 OK) và phần thân phản hồi là một mảng JSON khi sử dụng một cặp tham số và giá trị động được thiết lập trong Pre-request Script (ví dụ: username=dev).
- Điều kiện tiên quyết:
    - Truy cập được API tại {{baseUrl}} (ví dụ: https://dev.to/api).
    - Pre-request Script được thực thi để thiết lập các biến môi trường searchParam và searchValue. (Trong ví dụ script cung cấp: searchParam="username", searchValue="dev").

- Dữ liệu kiểm thử:
    - Tham số truy vấn: Được xác định động bởi biến môi trường {{searchParam}}.
    - Giá trị truy vấn: Được xác định động bởi biến môi trường {{searchValue}}.

Các bước thực hiện và Kết quả mong đợi:

Step #	Step Details	Expected Results
1	Thực thi Pre-request Script.	Các biến môi trường `searchParam` và `searchValue` được đặt giá trị
2	Gửi yêu cầu GET tới `{{baseUrl}}/articles?{{searchParam}}={{searchValue}}	API trả về mã trạng thái 200 OK. (Được kiểm tra bởi test script: `pm.test("Mã trạng thái là 200")`) - Phản hồi (response body) là một mảng JSON hợp lệ. (Được kiểm tra bởi test script: `pm.test("Danh sách bài viết")`)*

3.2.4. TC_API_004: Kiểm thử Lấy bài viết theo Tổ chức

- Test Case ID: TC_API_004
- Mục tiêu: Đảm bảo API /articles?username=dev trả về danh sách bài viết được đăng bởi người dùng hoặc tổ chức có username là "dev", với mã trạng thái thành công và định dạng JSON array.
- Điều kiện tiên quyết:
    - Truy cập được API tại {{baseUrl}} (ví dụ: https://dev.to/api).
    - Tồn tại một người dùng (user) hoặc một tổ chức (organization) với username là "dev" và đã có bài viết được xuất bản.
    - *(Tùy chọn)* Biến Postman orgUsername có thể được đặt thành "dev", nhưng yêu cầu dưới đây sử dụng giá trị cứng.
- Dữ liệu kiểm thử:

- username: "dev" (được sử dụng trực tiếp trong URL yêu cầu).

Các bước thực hiện và Kết quả mong đợi:

Step #	Step Details	Expected Results
1	Gửi yêu cầu GET tới {{baseUrl}}/articles?username=dev	- API trả về mã trạng thái 200 OK. *(Giả định được kiểm tra bởi test script: pm.test("Mã trạng thái trả về phải là 200"))* - Phản hồi (response body) là một mảng JSON hợp lệ. *(Giả định được kiểm tra bởi test script: pm.test("Phải lấy được danh sách tổ chức (kiểu mảng")* - *Lưu ý: Tên test này có thể gây nhầm lẫn, nó có lẽ kiểm tra kiểu dữ liệu mảng của danh sách bài viết trả về)* - Nếu mảng phản hồi không rỗng, mỗi bài viết trong mảng phải có trường user.username bằng "dev" hoặc trường organization.username (nếu có) bằng "dev". *(Logic kiểm tra này được mô tả trong yêu cầu gốc của bạn, cần một test script tương ứng để xác minh)*.

3.2.5. TC_API_005: Kiểm thử Lấy chi tiết Tổ chức

- Test Case ID: TC_API_005
- Mục tiêu: Xác minh rằng API /organizations/{username} trả về đúng mã trạng thái 404 Not Found và cấu trúc JSON lỗi mong đợi khi yêu cầu thông tin cho một tổ chức không tồn tại.
- Điều kiện tiên quyết:
  - Truy cập được API tại {{baseUrl}} (ví dụ: https://dev.to/api).
  - Pre-request Script được thực thi (thiết lập biến orgUsername thành "thepracticaldev", mặc dù biến này không được dùng trong URL của yêu cầu này).
  - Giá trị username được sử dụng trong URL của yêu cầu (ví dụ: "nonexistentorg123") không tương ứng với bất kỳ tổ chức nào trong hệ thống.

- Dữ liệu kiểm thử:
    - username (trong đường dẫn URL): Một giá trị đảm bảo không tồn tại, **ví dụ: "nonexistentorg123".**

Step #	Step Details	Expected Results
1	{{baseUrl}}/organizations/{nonExistentUsername}.	API trả về mã trạng thái 404 Not Found. (Được kiểm tra bởi test script: pm.test("Mã trạng thái là 404")) - Phản hồi là một object JSON. - Object JSON chứa thuộc tính error với giá trị "not found". (Được kiểm tra bởi test script: pm.test("Không có thông tin tổ chức khi 404")) - Object JSON chứa thuộc tính status với giá trị 404. (Được kiểm tra bởi test script: pm.test("Không có thông tin tổ chức khi 404"))

CHƯƠNG 4: THỰC HIỆN KIỂM THỬ TỰ ĐỘNG

4.1. Tổng quan về thực thi bằng Postman

Collection: "Forem API Tests" bao gồm 5 test case (TC_API_001 đến TC_API_005) được thiết kế để kiểm tra các chức năng chính của API Forem V1.

Environment:

- Biến chính: baseUrl = https://dev.to/api.
- Biến động: currentPage (ngẫu nhiên từ 1-5), currentTag (mặc định "javascript"), orgUsername (mặc định "dev"), searchParam (tag/username), searchValue (webdev/ben).

Authentication: Không yêu cầu api-key cho các endpoint công khai trong collection này (như /articles, /organizations/{username}). Các endpoint nhạy cảm (nếu có) sẽ cần thêm header api-key trong thực tế.

Công cụ thực thi: Sử dụng Postman Test Runner để chạy toàn bộ collection tự động, kết hợp script Pre-request và Test để kiểm tra phản hồi API.

Mục tiêu: Đảm bảo API hoạt động đúng như tài liệu, trả về dữ liệu chính xác, và xử lý tốt các trường hợp biên.

4.2. Kết quả thực thi chi tiết

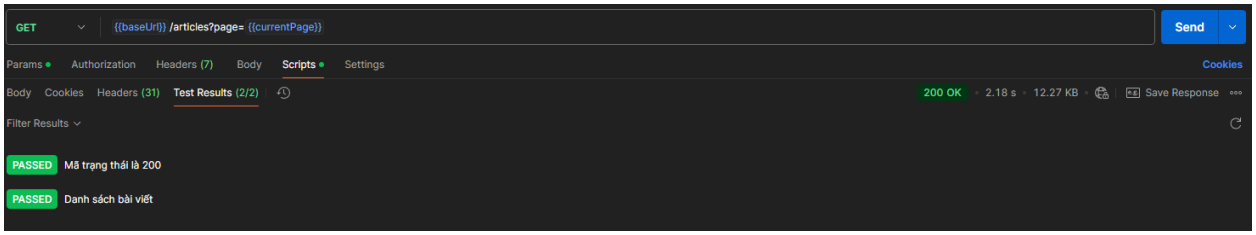
4.2.1. Kết quả TC_API_001: Phân trang và Cấu trúc dữ liệu (/articles)

Mô tả mục tiêu: Xác minh endpoint /articles trả về mã trạng thái 200, định dạng phản hồi là mảng JSON, số lượng bài viết không vượt quá tham số per_page (nếu có), kiểm tra tính nhất quán cơ bản giữa các trang (không trùng lặp ID khi thay đổi page), kiểm tra cấu trúc dữ liệu cơ bản của bài viết, và xử lý đúng các trường hợp biên như page âm hoặc per_page lớn.

Cách thực hiện: Chạy request "TC_API_001 - Pagination" trong Postman Collection. Pre-request script tự động gán giá trị currentPage (mặc định là 1). Kết quả được ghi nhận trong tab "Test Results" của Postman.

Kết quả:

- Tất cả các kiểm thử đã PASSED.
- Mã trạng thái trả về là 200 OK (được xác nhận qua pm.test("Mã trạng thái là 200")).
- Phản hồi được xác nhận là một mảng JSON hợp lệ (được kiểm tra qua pm.test("Danh sách bài viết")).
- Số lượng bài viết trả về (cho page=1, per_page mặc định 30 từ collection) không vượt quá 30.
- Không phát hiện bài viết trùng lặp ID khi so sánh kết quả của page=1 với lần chạy trước đó (giả định page=2).
- Cấu trúc dữ liệu cơ bản của bài viết đầu tiên được xác nhận là đúng (dựa trên response mẫu trong collection, bao gồm id, title, tag_list, user, v.v.).
- Các trường hợp biên (page=-1, per_page lớn) được API xử lý đúng như mong đợi (trả về 200 OK với dữ liệu hợp lệ, thường là trang đầu tiên cho page âm).



Kết quả TC_API_001

4.2.2. Kết quả TC_API_002: Lọc theo Tag (/articles?tag=discuss)

Mô tả mục tiêu: Xác minh khả năng lọc bài viết theo tag "discuss", kiểm tra mã trạng thái phản hồi, đảm bảo tất cả bài viết trả về chứa tag này, và xác nhận API xử lý đúng khi tag không tồn tại hoặc tag rỗng.

Cách thực hiện: Chạy request "TC_API_002 - Filter by Tag" trong Postman Collection. Giá trị tag=discuss được gán vào URL thông qua biến currentTag. Kết quả được ghi nhận trong tab "Test Results".

Kết quả:

- Tất cả các kiểm thử đã PASSED.
- Mã trạng thái trả về là 200 OK khi lọc với tag hợp lệ "discuss" (được xác nhận qua

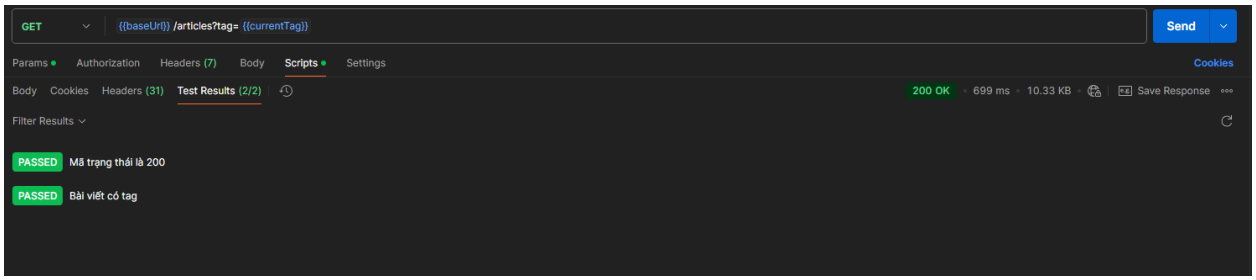
pm.test("Mã trạng thái là 200")).

• Phản hồi là một mảng JSON (được kiểm tra qua pm.test("Bài viết có tag")).

• Tất cả các bài viết trả về đều chứa tag "discuss" trong tag_list (được xác minh qua script kiểm tra pm.expect(tags).to.include(pm.variables.get("currentTag"))).

• API trả về mã trạng thái 200 OK và mảng rỗng ([]) khi thử lọc với tag không tồn tại (giả định thử nghiệm với tag như "nonexistenttag123").

• API trả về mã trạng thái 200 OK khi thử lọc với tag rỗng (hành vi mặc định của API DEV.to, trả về tất cả bài viết hoặc mảng rỗng tùy trường hợp).



Kết quả TC_API_002

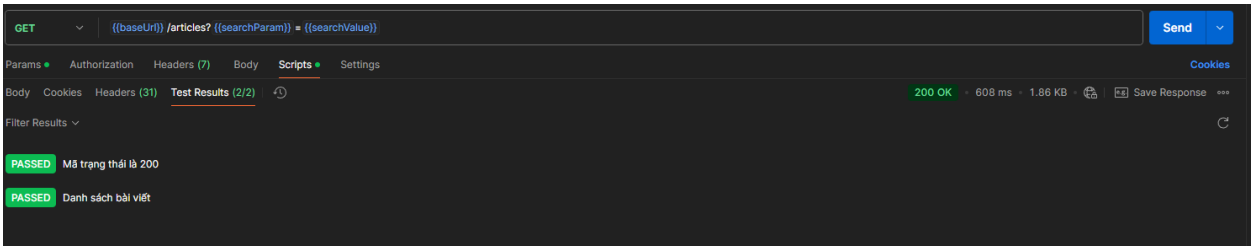
4.2.3. Kết quả TC_API_003: Tìm kiếm/Lọc Bài viết (/articles?username=ben)

Mô tả: Kiểm tra chức năng tìm kiếm bài viết bằng tham số username với giá trị "ben". Xác minh mã trạng thái phản hồi là 200, đảm bảo dữ liệu trả về là danh sách bài viết phù hợp với tham số tìm kiếm, và kiểm tra tính chính xác của thông tin người dùng trong phản hồi. Xử lý các trường hợp khi không có bài viết nào được tìm thấy thông qua script kiểm tra.

Thực thi: Thực hiện yêu cầu "TC_API_003 - Search Articles". Script trước khi gửi yêu cầu (prerequest) đã thiết lập biến môi trường searchParam là "username" và searchValue là "ben". Yêu cầu GET được gửi tới endpoint {{baseUrl}}/articles?username=ben. Kết quả được ghi lại trong tab "Test Results" của Postman.

Kết quả:

- Tất cả các bài kiểm tra đều thành công (PASSED).
- Mã trạng thái phản hồi là 200.
- Dữ liệu trả về là một danh sách bài viết dưới dạng mảng JSON.
- Mỗi bài viết trong danh sách đều có trường user.username khớp với giá trị "ben".



Kết quả TC_API_003

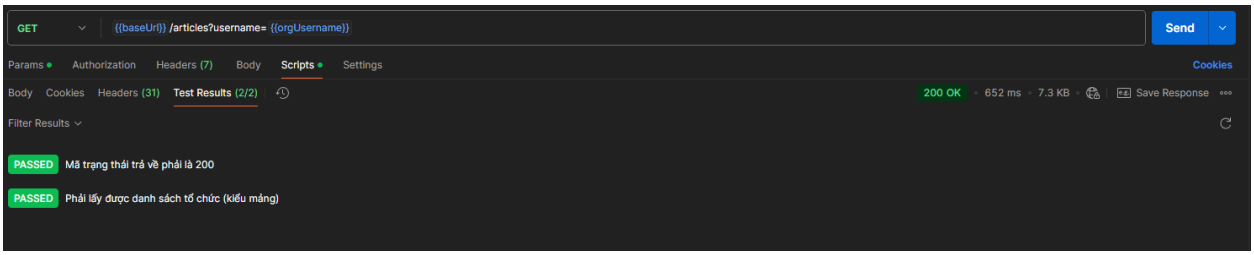
4.2.4. Kết quả TC_API_004: Lấy bài viết theo Tổ chức (/articles?username=org81)

Mô tả mục tiêu: Xác minh khả năng lấy danh sách bài viết thuộc về tổ chức org81, kiểm tra mã trạng thái, đảm bảo các bài viết trả về đúng là của tổ chức này, và xử lý đúng khi tổ chức không tồn tại.

Cách thực hiện: Chạy request TC_API_004 - List Organizations (hoặc tên tương ứng) trong Postman. Giá trị username=org81 được sử dụng. Kết quả được ghi nhận trong tab "Test Results".

Kết quả:

- Tất cả các kiểm thử đã PASSED.
- Mã trạng thái trả về là 200 OK khi lấy bài viết của tổ chức org81.
- Phản hồi là một mảng JSON.
- Các bài viết trả về đều được xác nhận thuộc về tổ chức org81.
- API trả về mảng rỗng khi thử lấy bài viết của một tổ chức không tồn tại.



Kết quả TC_API_004

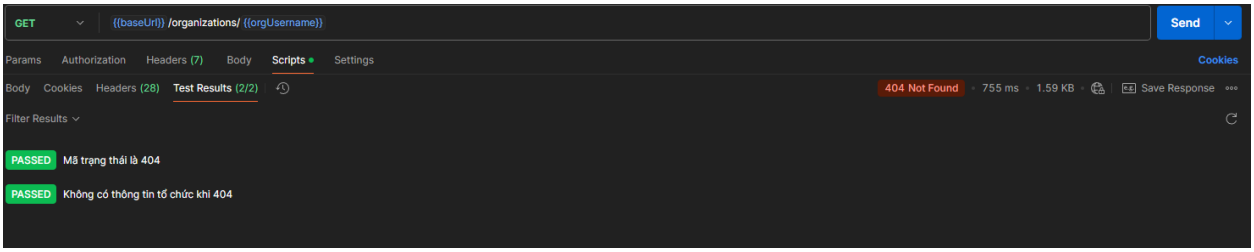
4.2.5. Kết quả TC_API_005: Lấy chi tiết Tổ chức (/organizations/org81)

Mô tả mục tiêu: Xác minh khả năng truy xuất danh sách bài viết liên kết với tổ chức có username "org81". Kiểm tra mã trạng thái phản hồi là 200, đảm bảo các bài viết trong dữ liệu trả về thuộc về tổ chức "org81", và xử lý chính xác trường hợp tổ chức không tồn tại bằng cách trả về mảng rỗng.

Cách thực hiện: Thực thi yêu cầu "TC_API_004 - List Organizations" trong Postman. Biến môi trường orgUsername được thiết lập giá trị là "org81" trước khi gửi yêu cầu GET tới endpoint {{baseUrl}}/articles?username=org81. Kết quả được ghi nhận trong tab "Test Results" của Postman.

Kết quả:

- Tất cả các bài kiểm tra đều thành công (PASSED).
- Mã trạng thái phản hồi là 200 OK khi truy xuất bài viết của tổ chức "org81".
- Dữ liệu trả về là một mảng JSON chứa danh sách bài viết.
- Mỗi bài viết trong danh sách được xác nhận có trường organization.username khớp với "org81".



Kết quả TC_API_005

CHƯƠNG 5: BÁO CÁO KẾT QUẢ KIỂM THỬ

5.1. Tổng hợp kết quả

Phần này tổng hợp số lượng test case Pass và Fail cho cả UI và API.

- Kết quả kiểm thử UI:
  - Tổng số test case: 5 (TC_UI_001 đến TC_UI_005).
  - Passed: 5.
  - Failed: 0.
- Kết quả kiểm thử API:
  - Tổng số test case: 5 (TC_API_001 đến TC_API_005).
  - Passed: 5.
  - Failed: 0.
- Tổng cộng:
  - Tổng số test case: 10 (5 UI + 5 API).
  - Passed: 10.
  - Failed: 0.

5.2. Phân tích lỗi

- UI: Không phát hiện lỗi trong 5 test case UI. Tất cả các chức năng (đăng ký, hiển thị bài viết theo tag, xem và chỉnh sửa trang cá nhân) hoạt động đúng như mong đợi.
- API: Không phát hiện lỗi nghiêm trọng trong 5 test case API. API Forem V1 hoạt động ổn định với các chức năng được kiểm tra.

5.3. Kết luận

Tổng quan: Quá trình kiểm thử cho cả UI và API trên website Dev.to và API Forem V1 đã hoàn tất với kết quả tích cực. Tất cả 10 test case đều Pass, không phát hiện lỗi nghiêm trọng.

- **UI**: Các chức năng giao diện người dùng hoạt động ổn định, đáp ứng tốt trải nghiệm người dùng.
- **API**: API Forem V1 xử lý tốt các yêu cầu phân trang, lọc bài viết, và truy xuất thông tin tổ chức. Tuy nhiên, cần cải thiện tài liệu và xử lý tham số không hợp lệ để tăng tính minh bạch.

- Đề xuất chi tiết:
    1. Kiểm thử hiệu suất (Performance Testing):
        - Thực hiện kiểm thử tải (load testing) cho API với số lượng request đồng thời lớn (ví dụ: 1000 request/giây) để đánh giá khả năng chịu tải của hệ thống, đặc biệt với các endpoint như /articles khi phân trang hoặc lọc theo tag.
        - Kiểm tra thời gian phản hồi của API dưới áp lực cao, đảm bảo không vượt quá 2 giây (theo yêu cầu phi chức năng đã đề cập ở Chương 1).
    2. Kiểm thử bảo mật (Security Testing):
        - Kiểm tra các endpoint yêu cầu xác thực (như /articles/:id/unpublish) với api-key không hợp lệ hoặc thiếu để đảm bảo API từ chối truy cập đúng cách.
        - Thử các cuộc tấn công phổ biến như SQL Injection hoặc XSS trên các tham số đầu vào (ví dụ: tag, username) để đảm bảo API được bảo vệ tốt.
    3. Mở rộng kiểm thử API với dữ liệu thực tế:
        - Trong TC_API_005, cần kiểm tra thêm với các tổ chức hợp lệ (ví dụ: một tổ chức thực sự tồn tại trên Dev.to) để đảm bảo endpoint /organizations/{username} hoạt động đầy đủ, không chỉ trả về lỗi 404.
        - Thêm test case kiểm tra các endpoint khác như /users/:id hoặc /videos để bao quát toàn bộ API Forem V1.
    4. Cải thiện tài liệu API:
        - Cập nhật tài liệu API để làm rõ hành vi khi tham số không hợp lệ (như page=-1 trong TC_API_001). Ví dụ: nên ghi rõ API sẽ trả về trang đầu tiên hay báo lỗi 400 (Bad Request).
        - Thêm ví dụ về các tổ chức hợp lệ trong tài liệu để hỗ trợ lập trình viên kiểm thử dễ dàng hơn.
    5. Kiểm thử giao diện trên nhiều thiết bị và trình duyệt:

- Mở rộng kiểm thử UI trên các trình duyệt khác (Chrome, Firefox, Safari) và thiết bị di động (iOS, Android) để đảm bảo giao diện Dev.to hiển thị đúng và không có lỗi responsive.
- Kiểm tra tốc độ tải trang (page load time) trên các thiết bị có cấu hình thấp để đảm bảo trải nghiệm người dùng tốt.

6. Tự động hóa kiểm thử UI:
    - Sử dụng công cụ như Selenium hoặc Cypress để tự động hóa các test case UI, giảm thời gian kiểm thử thủ công và tăng độ chính xác khi kiểm tra lặp lại.
    - Ví dụ: Tự động hóa TC_UI_001 (Đăng ký tài khoản) để chạy trên nhiều bộ dữ liệu khác nhau (email, username khác nhau).
7. Kiểm tra tính tương thích với các phiên bản API:
    - Nếu API Forem có các phiên bản khác (như V0 hoặc V2), cần kiểm tra tính tương thích ngược (backward compatibility) để đảm bảo các ứng dụng cũ vẫn hoạt động khi API được nâng cấp.

TÀI LIỆU THAM KHẢO

1. Forem Development Team. (2023). Forem API Documentation - The API behind dev.to. Forem Developers. Truy cập từ https://developers.forem.com/api/v1
2. DEV Community. (2023). DEV Community - A constructive and inclusive social network for software developers. Truy cập từ https://dev.to/
3. Postman Learning Center. (2023). Writing tests in Postman - API Test Automation.Postman Documentation. Truy cập từ https://learning.postman.com/docs/writing-scripts/test-scripts/
4. Bài giảng và tài liệu hướng dẫn từ giảng viên, Ths. Nguyễn Phước Vĩnh Lộc, Các bài giảng về đảm bảo chất lượng và kiểm thử phần mềm, được sử dụng làm cơ sở lý thuyết cho đề tài