

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN



ĐỒ ÁN 4

**XÂY DỰNG FRAMEWORK KIỂM THỦ TỰ ĐỘNG CHO
ỨNG DỤNG WEB VÀ ỨNG DỤNG KIỂM THỦ CHO WEBSITE
SAVOUREBAKERY.COM**

**NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM**

SINH VIÊN: TRẦN THỊ THU HÀ

MÃ LỚP: 12522T.1

HƯỚNG DẪN: ĐÀO ANH HIỀN

HƯNG YÊN - 2026

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

LỜI CAM ĐOAN

Em xin cam đoan đồ án “Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Đào Anh Hiển

Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong đồ án và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước Khoa và Nhà trường.

Hưng Yên, ngày ... tháng ... năm.....

Sinh viên

.....

LỜI CẢM ƠN

Để có thể hoàn thành đồ án này, đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin - Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện đồ án môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Đào Anh Hiển đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đồ án vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được đồ án này.

Mặc dù em đã có cố gắng, nhưng với kiến thức còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong đồ án.

Em xin trân trọng cảm ơn!

MỤC LỤC

NHẬN XÉT	2
DANH MỤC CÁC THUẬT NGỮ	7
DANH MỤC CÁC BẢNG	8
DANH MỤC CÁC HÌNH VẼ	9
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	10
1.1 Lý do chọn đề tài	10
1.2 Mục tiêu của đề tài	10
1.2.1 Mục tiêu tổng quát	10
1.2.2 Mục tiêu cụ thể	11
1.3 Giới hạn và phạm vi của đề tài	12
1.3.1 Đối tượng nghiên cứu	12
1.3.2 Phạm vi nghiên cứu	12
1.4 Nội dung thực hiện	12
1.5 Phương pháp tiếp cận	13
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	14
2.1 Tổng quan về kiểm thử tự động	14
2.2 Giới thiệu các công cụ kiểm thử	15
2.2.1 Playwright	15
2.2.2 Pytest	16
2.2.3 OpenPyXL và các định dạng dữ liệu	16
2.2.4 Allure	17
2.3 Kiến trúc của Framework kiểm thử tự động	17
2.3.1 Vai trò trung tâm của conftest.py	17
2.3.2 Kiến trúc Page Object Model (POM)	18
2.3.3 Data Driven Testing (DDT)	19

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

2.3.4 Kiến trúc tổng thể của Framework	19
CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM	21
3.1 Giới thiệu chung	21
3.2 Các yêu cầu chức năng	21
3.3 Các yêu cầu phi chức năng	28
3.4 Tính ứng dụng	29
3.5 Tính toàn vẹn dữ liệu	30
CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG	31
4.1 Đặt vấn đề	31
4.2 Xây dựng framework kiểm thử tự động	31
4.2.1 Mô hình framework kiểm thử	31
4.2.2. Cấu trúc thư mục framework	32
4.3 Thực thi và báo cáo kiểm thử	35
4.3.3 Xây dựng kịch bản kiểm thử	57
CHƯƠNG 5: KẾT LUẬN	67
5.1. Kết quả đạt được	67
5.2. Những hạn chế của đề tài	67
5.3. Hướng phát triển của đề tài	68

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng anh	Điễn giải
1	SRS	Software Requirement Specification	Đặc tả yêu cầu phần mềm
2	POM	Page Object Model	

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

DANH MỤC CÁC BẢNG

Bảng 1: Kiến trúc tổng thể Framework	19
Bảng 2: Các yêu cầu chức năng	22
Bảng 3: Các yêu cầu phi chức năng	29
Bảng 4: Tính ứng dụng	30
Bảng 5: Tính toàn vẹn dữ liệu	30
Bảng 6: Testcase chức năng đăng nhập	39
Bảng 7: Testcase chức năng tìm kiếm sản phẩm	42
Bảng 8: Testcase chức năng xem chi tiết sản phẩm	44
Bảng 9: Testcase chức năng quản lý giỏ hàng	46
Bảng 10: Testcase chức năng cập nhật thông tin cá nhân	49

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

DANH MỤC CÁC HÌNH VẼ

Hình 1: Giao diện đăng nhập	22
Hình 2: Giao diện tìm kiếm sản phẩm	23
Hình 3: Giao diện xem chi tiết sản phẩm	24
Hình 4: Giao diện quản lý giỏ hàng	25
Hình 5: Giao diện cập nhật thông tin	27

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài

Trong những năm gần đây, sự phát triển mạnh mẽ của thương mại điện tử đã thúc đẩy các doanh nghiệp, cửa hàng kinh doanh thực phẩm và bánh ngọt đầy mạnh hoạt động bán hàng thông qua website. Website Savourebakery.com là một nền tảng giới thiệu và kinh doanh các sản phẩm bánh, cho phép người dùng dễ dàng tìm kiếm sản phẩm, xem thông tin chi tiết và thực hiện đặt hàng trực tuyến.

Tuy nhiên, trong quá trình vận hành và nâng cấp website, nếu không được kiểm thử đầy đủ và thường xuyên, hệ thống rất dễ phát sinh lỗi như lỗi hiển thị giao diện, lỗi chức năng đặt hàng, giỏ hàng, đăng nhập hoặc các vấn đề liên quan đến hiệu năng và bảo mật. Việc kiểm thử thủ công truyền thống tuy đơn giản nhưng lại tiêu tốn nhiều thời gian, phụ thuộc vào con người và khó đảm bảo tính nhất quán khi hệ thống liên tục thay đổi.

Kiểm thử tự động được xem là giải pháp hiệu quả giúp giảm thiểu rủi ro, tăng tốc độ kiểm thử và nâng cao chất lượng phần mềm. Tuy nhiên, việc áp dụng kiểm thử tự động một cách bài bản đòi hỏi phải có một framework kiểm thử rõ ràng, dễ mở rộng và dễ bảo trì, phù hợp với đặc thù của từng website.

Xuất phát từ nhu cầu thực tế đó, đề tài “Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com” được lựa chọn nhằm xây dựng một framework kiểm thử tự động có tính ứng dụng cao, góp phần đảm bảo website hoạt động ổn định, chính xác và nâng cao trải nghiệm người dùng.

1.2 Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

Mục tiêu tổng quát của đề tài là xây dựng một framework kiểm thử tự động cho ứng dụng Web và triển khai kiểm thử thực tế trên website Savourebakery.com, nhằm

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

đảm bảo hệ thống hoạt động ổn định, đúng yêu cầu nghiệp vụ và nâng cao chất lượng tổng thể của website bán hàng trực tuyến.

1.2.2 Mục tiêu cụ thể

Trên cơ sở mục tiêu tổng quát, đề tài hướng tới các mục tiêu cụ thể sau:

- Phân tích các chức năng nghiệp vụ chính của website Savourebakery.com như: đăng nhập, xem sản phẩm, tìm kiếm, giỏ hàng và trang thông tin cá nhân.
- Xây dựng các kịch bản kiểm thử nhằm đảm bảo tính chính xác và đầy đủ của các chức năng cốt lõi.
- Thực hiện kiểm thử chức năng cho các module chính:
 - + Người dùng truy cập website, đăng nhập, xem danh sách sản phẩm và chi tiết sản phẩm
 - + Thao tác thêm sản phẩm vào giỏ hàng và cập nhật thông tin cá nhân.
- Thực hiện kiểm thử phi chức năng, bao gồm:
 - + Bảo mật: kiểm tra các vấn đề cơ bản liên quan đến đăng nhập, dữ liệu người dùng và nguy cơ tấn công phổ biến
 - + Hiệu năng: đánh giá thời gian tải trang và khả năng đáp ứng của website khi có nhiều người truy cập
 - + Tính tương thích: kiểm tra khả năng hoạt động trên các trình duyệt phổ biến và thiết bị khác nhau
- Áp dụng kiểm thử tự động để tự động hóa các kịch bản kiểm thử lặp đi lặp lại, giúp tiết kiệm thời gian và giảm sai sót.
- Ghi nhận và báo cáo lỗi một cách có hệ thống, bao gồm mô tả lỗi, mức độ nghiêm trọng và đề xuất hướng khắc phục.
- Đánh giá chất lượng tổng thể của website dựa trên kết quả kiểm thử và đề xuất cải tiến cho các phiên bản tiếp theo.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

1.3 Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

- Đối tượng nghiên cứu: Đề tài tập trung vào việc nghiên cứu và xây dựng một framework kiểm thử tự động cho các ứng dụng web, với trọng tâm là ứng dụng công cụ Playwright.

- Khách thể nghiên cứu: Các đối tượng được khảo sát và chịu tác động trực tiếp bao gồm:

+ Người dùng của website Savourebakery.com, những người sẽ hưởng lợi từ chất lượng sản phẩm được cải thiện.

+ Đội ngũ phát triển phần mềm và kiểm thử viên, những người sẽ trực tiếp sử dụng framework này để nâng cao hiệu quả công việc.

1.3.2 Phạm vi nghiên cứu

- Phạm vi không gian: Đề tài được triển khai và áp dụng trực tiếp trên hệ thống website chính thức Savourebakery.com,.

- Phạm vi thời gian: Quá trình nghiên cứu và triển khai được thực hiện trong khoảng thời gian từ tháng 8/2025 đến tháng 1/2026.

- Ý nghĩa khoa học và thực tiễn:

+ Ý nghĩa khoa học: Đề tài góp phần hệ thống hóa kiến thức về kiểm thử tự động, cung cấp một nghiên cứu thực tế về việc áp dụng công cụ Playwright vào quy trình kiểm thử.

+ Ý nghĩa thực tiễn: Framework được xây dựng có thể được áp dụng trực tiếp để cải thiện chất lượng kiểm thử cho website Savourebakery.com. Đồng thời, đây cũng là một mô hình tham khảo hữu ích cho việc kiểm thử các website thương mại điện tử tương tự, giúp tiết kiệm đáng kể thời gian và chi phí.

1.4 Nội dung thực hiện

- ❖ Tìm hiểu lý thuyết về kiểm thử phần mềm và kiểm thử tự động.
- ❖ Khảo sát website Savourebakery.com, xác định các chức năng cần kiểm thử.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

- ❖ Thiết kế kịch bản và test case cho các chức năng chính.
- ❖ Xây dựng framework kiểm thử tự động với Playwright.
- ❖ Thực thi test case, ghi nhận kết quả và phân tích.
- ❖ Đánh giá ưu điểm, hạn chế và đề xuất hướng phát triển trong tương lai.

1.5 Phương pháp tiếp cận

- ❖ Khảo sát hệ thống: Trực tiếp phân tích và sử dụng website Savourebakery.com để nắm bắt quy trình hoạt động.
- ❖ Tiếp cận thực nghiệm: Triển khai framework kiểm thử tự động, chạy thử nghiệm trên các chức năng chính của website.
- ❖ So sánh, đánh giá: Đôi chiêu hiệu quả giữa kiểm thử thủ công và kiểm thử tự động để rút ra nhận xét.
- ❖ Tổng hợp, phân tích tài liệu: Tham khảo các tài liệu nghiên cứu, hướng dẫn sử dụng Playwright và các mô hình kiểm thử phổ biến.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về kiểm thử tự động

Kiểm thử tự động (Automated Testing) là phương pháp sử dụng các công cụ và chương trình để thực thi các kịch bản kiểm thử thay cho thao tác thủ công của con người. Các hành vi của người dùng như truy cập website, đăng nhập, tìm kiếm, thêm sản phẩm vào giỏ hàng, cập nhật thông tin cá nhân... được mô phỏng thông qua mã nguồn kiểm thử.

Trong bối cảnh các website thương mại điện tử như Savourebakery.com thường xuyên thay đổi giao diện, cập nhật sản phẩm và tính năng, kiểm thử tự động đóng vai trò quan trọng trong việc đảm bảo chất lượng hệ thống. Việc tự động hóa giúp phát hiện lỗi sớm, giảm thiểu rủi ro phát sinh lỗi sau mỗi lần cập nhật và nâng cao độ ổn định của website.

Những lợi ích chính của kiểm thử tự động bao gồm:

- Tiết kiệm thời gian và công sức: Các kịch bản kiểm thử có thể chạy lặp lại nhiều lần mà không cần thao tác thủ công.
- Tăng tính chính xác: Hạn chế sai sót do con người khi thực hiện các bước kiểm thử lặp đi lặp lại.
- Hỗ trợ kiểm thử hồi quy: Dễ dàng xác minh các chức năng cũ vẫn hoạt động đúng sau khi có thay đổi.
- Nâng cao độ phủ kiểm thử: Cho phép chạy test trên nhiều trình duyệt và dữ liệu khác nhau.
- Phù hợp với quy trình phát triển hiện đại: Dễ tích hợp vào CI/CD để kiểm tra chất lượng hệ thống liên tục.

Trong đề tài này, kiểm thử tự động tập trung chủ yếu vào kiểm thử giao diện người dùng (UI Testing) và kiểm thử hồi quy, đảm bảo các chức năng chính của website Savourebakery hoạt động ổn định.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

2.2 Giới thiệu các công cụ kiểm thử

2.2.1 *Playwright*

Playwright là thư viện mã nguồn mở do Microsoft phát triển, chuyên dùng để kiểm thử tự động giao diện web. Công cụ này hỗ trợ nhiều trình duyệt phổ biến như Chromium, Firefox và WebKit, đồng thời hoạt động ổn định trên các hệ điều hành Windows, macOS và Linux.

Trong đề tài, Playwright được sử dụng để mô phỏng các thao tác thực tế của người dùng trên website Savourebakery như:

- Đăng nhập tài khoản
- Tìm kiếm sản phẩm
- Xem chi tiết sản phẩm
- Quản lý giỏ hàng
- Cập nhật thông tin cá nhân

Các thành phần chính của Playwright bao gồm:

- Browser: đại diện cho trình duyệt được khởi chạy.
- Browser Context: mô phỏng một phiên làm việc độc lập.
- Page: đại diện cho một tab trình duyệt.
- Locator: xác định và thao tác với các phần tử trên trang.
- Screenshot & Video: hỗ trợ ghi lại bằng chứng khi test thất bại.

Ưu điểm:

- Tốc độ thực thi nhanh, ổn định.
- Cơ chế auto-wait giúp hạn chế lỗi do phần tử chưa load.
- Hỗ trợ chụp ảnh khi test fail, thuận tiện cho phân tích lỗi.

Hạn chế:

- Yêu cầu người dùng có kiến thức lập trình Python.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

- Là công cụ tương đối mới nên tài liệu tiếng Việt còn hạn chế.

2.2.2 Pytest

Pytest là framework kiểm thử phổ biến nhất trong Python, được sử dụng để tổ chức, thực thi và quản lý các test case trong dự án.

Trong framework của đê tài, Pytest đảm nhiệm các vai trò:

- Quản lý vòng đời kiểm thử thông qua fixture
- Tổ chức test theo module chức năng
- Kết hợp với Playwright để kiểm thử giao diện
- Tích hợp với Allure để sinh báo cáo

Pytest cho phép xây dựng các fixture dùng chung cho toàn bộ project, giúp giảm trùng lặp mã và dễ bảo trì.

Ưu điểm:

- Cú pháp đơn giản, dễ đọc.
- Fixture linh hoạt, có thể tái sử dụng.
- Dễ dàng mở rộng và tích hợp với công cụ khác.

Nhược điểm:

- Không có giao diện đồ họa.
- Người mới cần thời gian làm quen với fixture nâng cao.

2.2.3 OpenPyXL và các định dạng dữ liệu

Để phục vụ Data Driven Testing, dự án sử dụng nhiều định dạng dữ liệu khác nhau:

- Excel (.xlsx): lưu dữ liệu đăng nhập, cập nhật hồ sơ.
- CSV (.csv): dữ liệu đăng nhập nhanh.
- JSON (.json): dữ liệu tìm kiếm sản phẩm.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

Thư viện OpenPyXL được sử dụng để đọc và xử lý dữ liệu Excel, giúp test case có thể chạy với nhiều bộ dữ liệu khác nhau mà không cần sửa mã nguồn.

2.2.4 Allure

Allure là công cụ tạo báo cáo kiểm thử trực quan dưới dạng HTML. Sau mỗi lần chạy test, Allure hiển thị:

- Danh sách test case
- Trạng thái Pass / Fail
- Thời gian thực thi
- Ảnh chụp màn hình khi lỗi xảy ra

Trong project, kết quả kiểm thử được lưu tại:

- reports/allure-results
- Báo cáo HTML tại reports/allure-report

Allure giúp việc đánh giá chất lượng website Savourebakery trở nên trực quan và dễ theo dõi.

2.3 Kiến trúc của Framework kiểm thử tự động

2.3.1 Vai trò trung tâm của conftest.py

Trong project, conftest.py đóng vai trò là trung tâm cấu hình của toàn bộ framework, thay thế cho việc sử dụng thư mục config/.

File conftest.py chịu trách nhiệm:

- Khởi tạo Playwright
- Quản lý vòng đời trình duyệt
- Tạo context và page cho từng test
- Cung cấp base URL cho website
- Chuẩn bị trạng thái đặc biệt như giỏ hàng có sẵn sản phẩm hoặc tài khoản đã đăng nhập

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

- Tự động chụp screenshot khi test thất bại

Các fixture chính bao gồm:

- playwright_instance
- browser
- context
- page
- base_url
- cart_with_product
- login_for_profile

Cách tổ chức này giúp toàn bộ test script không cần lặp lại mã khởi tạo trình duyệt, đồng thời đảm bảo tính nhất quán giữa các test case.

2.3.2 Kiến trúc Page Object Model (POM)

Framework áp dụng mô hình Page Object Model, trong đó mỗi trang web được biểu diễn bằng một class Python riêng biệt trong thư mục pages/.

Ví dụ:

```
home_page.py  
login_page.py  
product_page.py  
cart_page.py  
profile_page.py  
search_page.py
```

Mỗi class chứa:

- Locator của các phần tử giao diện
- Các phương thức thao tác tương ứng với hành vi người dùng

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

Việc áp dụng POM giúp mã kiểm thử:

- Dễ đọc
- Dễ bảo trì
- Dễ mở rộng khi giao diện thay đổi

2.3.3 Data Driven Testing (DDT)

Framework tích hợp DDT bằng cách tách dữ liệu kiểm thử ra khỏi test script. Dữ liệu được lưu trong thư mục data/ dưới nhiều định dạng khác nhau.

Các file tiêu biểu:

login_data.csv
profile_update_data.xlsx
search_data.json

Việc này cho phép chạy cùng một test case với nhiều bộ dữ liệu khác nhau, đặc biệt phù hợp cho kiểm thử hồi quy.

2.3.4 Kiến trúc tổng thể của Framework

Thư mục	Chức năng
pages	Chứa các lớp Page Object
tests	Các test script Pytest
data	Dữ liệu kiểm thử
utils	Hàm đọc dữ liệu
reports	Allure report, screenshot
conftest.py	Quản lý fixture và cấu hình
pytest.ini	Cấu hình Pytest

Bảng 1: Kiến trúc tổng thể Framework

Kiến trúc này giúp framework gọn gàng, dễ hiểu và phù hợp cho việc mở rộng trong tương lai.

2.4 Test Script và phương pháp xây dựng Test Script

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

2.4.1 Ngôn ngữ và công cụ

- Ngôn ngữ: Python
- Công cụ: Playwright + Pytest + Allure
- Dữ liệu: Excel, CSV, JSON

Mỗi file test trong thư mục tests/ tương ứng với một nhóm chức năng của website.

2.4.2 Quy trình xây dựng Test Script

Bước 1: Chuẩn bị dữ liệu kiểm thử trong thư mục data/

Bước 2: Xây dựng Page Object cho từng trang

Bước 3: Viết test case sử dụng fixture từ conftest.py

Bước 4: Chạy test bằng Pytest

Bước 5: Sinh báo cáo Allure và phân tích kết quả

2.4.3 Ưu điểm của phương pháp

Cấu trúc rõ ràng, dễ bảo trì

Tái sử dụng mã nguồn hiệu quả

Phân tách rõ ràng giữa dữ liệu và logic

Báo cáo trực quan, hỗ trợ phân tích lỗi nhanh

Phù hợp làm việc nhóm và mở rộng lâu dài

CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM

3.1 Giới thiệu chung

Hệ thống website Savoure Bakery là một website thương mại điện tử cho phép người dùng tìm kiếm, xem thông tin sản phẩm bánh, thực hiện mua hàng và quản lý thông tin cá nhân.

Hệ thống hỗ trợ các chức năng cơ bản như đăng nhập, tìm kiếm sản phẩm, xem chi tiết sản phẩm, quản lý giỏ hàng và cập nhật thông tin người dùng, nhằm mang lại trải nghiệm mua sắm thuận tiện cho khách hàng.

3.2 Các yêu cầu chức năng

STT	Tên chức năng	Mô tả
1	Đăng nhập	<p>Hệ thống website Savoure Bakery là một website thương mại điện tử cho phép người dùng tìm kiếm, xem thông tin sản phẩm bánh, thực hiện mua hàng và quản lý thông tin cá nhân.</p> <p>Hệ thống hỗ trợ các chức năng cơ bản như đăng nhập, tìm kiếm sản phẩm, xem chi tiết sản phẩm, quản lý giỏ hàng và cập nhật thông tin người dùng, nhằm mang lại trải nghiệm mua sắm thuận tiện cho khách hàng.</p>
2	Tìm kiếm sản phẩm	Cho phép người dùng tìm kiếm sản phẩm theo từ khóa. Hệ thống hiển thị danh sách các sản phẩm phù hợp với nội dung tìm kiếm của người dùng.
3	Xem chi tiết sản phẩm	Cho phép người dùng xem thông tin chi tiết của sản phẩm bao gồm tên sản phẩm, hình ảnh, giá bán và mô tả sản phẩm.
4	Quản lý giỏ hàng	Cho phép người dùng thêm sản phẩm vào giỏ hàng, xem danh sách sản phẩm trong giỏ hàng và cập nhật số lượng sản phẩm trước khi thanh toán.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

5	Cập nhật thông tin người dùng	Cho phép người dùng thêm sản phẩm vào giỏ hàng, xem danh sách sản phẩm trong giỏ hàng và cập nhật số lượng sản phẩm trước khi thanh toán.
---	-------------------------------	---

Bảng 2: Các yêu cầu chức năng

* **Đặc tả chức năng Đăng nhập:**

Dăng Nhập

Địa chỉ email *

Mật khẩu *

Ghi nhớ đăng nhập [Quên mật khẩu?](#)

Đăng Nhập

Chưa có tài khoản? [Tạo Tài Khoản](#)

Hình 1: Giao diện đăng nhập

Chức năng Đăng nhập cho phép người dùng truy cập vào hệ thống Savourebakery thông qua tài khoản đã đăng ký. Người dùng truy cập website tại địa chỉ <https://savourebakery.com/>, từ trang chủ chọn chức năng Đăng nhập, hệ thống sẽ điều hướng đến giao diện đăng nhập.

Giao diện Đăng nhập bao gồm hai trường nhập liệu chính là Email và Mật khẩu, kèm theo các thành phần hỗ trợ như tùy chọn Ghi nhớ đăng nhập, liên kết Quên mật khẩu và liên kết Tạo tài khoản.

Trường Email bắt buộc phải được nhập và đúng định dạng email hợp lệ. Trường Mật khẩu không được để trống và dữ liệu nhập vào được hiển thị dưới dạng ký tự ẩn. Khi người dùng nhấn nút Đăng Nhập, hệ thống tiến hành kiểm tra tính hợp lệ của dữ liệu và xác thực thông tin với cơ sở dữ liệu.

Trong trường hợp thông tin đăng nhập hợp lệ, hệ thống xác thực thành công và chuyển người dùng sang giao diện tài khoản cá nhân. Ngược lại, nếu dữ liệu không

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

hợp lệ như bỏ trống Email hoặc Mật khẩu, nhập sai định dạng Email hoặc thông tin đăng nhập không đúng với dữ liệu trong hệ thống, hệ thống sẽ hiển thị thông báo lỗi phù hợp để người dùng thực hiện nhập lại.

Chức năng Ghi nhớ đăng nhập cho phép hệ thống lưu trạng thái đăng nhập của người dùng trong một khoảng thời gian nhất định, giúp người dùng không cần đăng nhập lại khi truy cập website lần sau. Liên kết Quên mật khẩu hỗ trợ người dùng khôi phục mật khẩu trong trường hợp quên thông tin đăng nhập, trong khi liên kết Tạo tài khoản cho phép người dùng đăng ký tài khoản mới nếu chưa có tài khoản trong hệ thống.

* Đặc tả chức năng tìm kiếm sản phẩm:



Hình 2: Giao diện tìm kiếm sản phẩm

Chức năng tìm kiếm sản phẩm cho phép người dùng nhanh chóng tìm kiếm các sản phẩm có trong hệ thống thông qua từ khóa. Chức năng này hỗ trợ người dùng rút ngắn thời gian duyệt sản phẩm và nâng cao trải nghiệm mua sắm trên website Savourebakery.

Trên giao diện website, người dùng ánh chọn icon tìm kiếm, nhập từ khóa cần tìm vào ô tìm kiếm sản phẩm. Từ khóa tìm kiếm có thể là tên sản phẩm hoặc một phần tên sản phẩm. Sau khi nhập từ khóa và thực hiện thao tác tìm kiếm, hệ thống sẽ tiến hành truy vấn dữ liệu và hiển thị danh sách các sản phẩm phù hợp với nội dung tìm kiếm.

Trong trường hợp từ khóa tìm kiếm hợp lệ và tồn tại sản phẩm tương ứng trong hệ thống, website sẽ hiển thị danh sách sản phẩm liên quan kèm theo các thông tin cơ

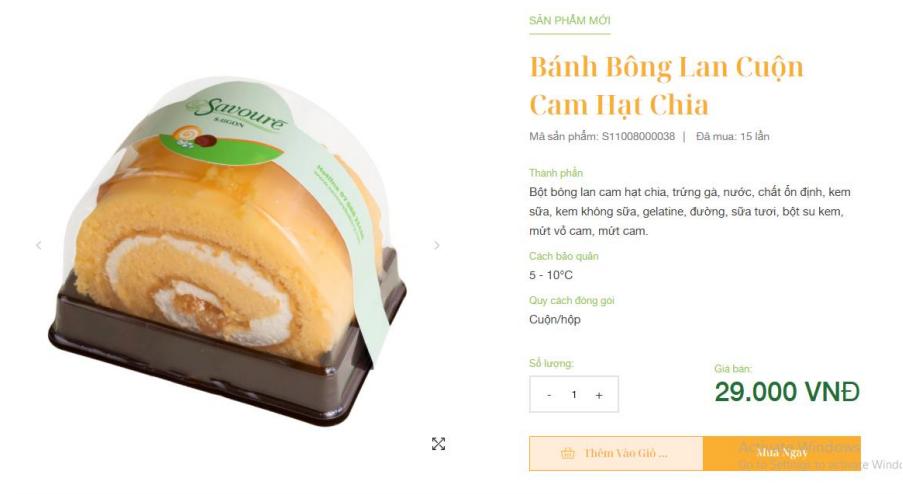
Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

bản như tên sản phẩm, hình ảnh và giá bán. Người dùng có thể tiếp tục chọn sản phẩm để xem chi tiết hoặc thực hiện các thao tác mua hàng.

Ngược lại, nếu người dùng không nhập từ khóa, nhập từ khóa không hợp lệ hoặc hệ thống không tìm thấy sản phẩm phù hợp, website sẽ hiển thị thông báo hoặc giao diện kết quả rỗng nhằm thông báo cho người dùng rằng không có sản phẩm nào được tìm thấy.

Chức năng tìm kiếm được thiết kế để hoạt động nhanh chóng, chính xác và ổn định, đảm bảo đáp ứng nhu cầu tra cứu sản phẩm của người dùng trong quá trình sử dụng website.

* Đặc tả chức năng xem chi tiết sản phẩm:



Hình 3: Giao diện xem chi tiết sản phẩm

Chức năng xem chi tiết sản phẩm cho phép người dùng xem đầy đủ thông tin của một sản phẩm cụ thể trên website Savourebakery. Chức năng này hỗ trợ người dùng đưa ra quyết định mua hàng chính xác dựa trên các thông tin chi tiết về sản phẩm.

Từ danh sách sản phẩm trên trang chủ hoặc trang kết quả tìm kiếm, người dùng có thể click vào một sản phẩm bất kỳ. Sau khi thực hiện thao tác này, hệ thống sẽ điều hướng người dùng đến trang Chi tiết sản phẩm tương ứng.

Tại trang chi tiết sản phẩm, hệ thống hiển thị các thông tin quan trọng bao gồm hình ảnh sản phẩm, tên sản phẩm, mã sản phẩm, số lượt mua, giá bán, thành phần, cách

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

bảo quản, quy cách đóng gói và số lượng sản phẩm. Người dùng có thể thay đổi số lượng sản phẩm mong muốn thông qua nút tăng hoặc giảm số lượng.

Ngoài ra, trang chi tiết sản phẩm cung cấp các chức năng hỗ trợ mua hàng như Thêm vào giỏ hàng và Mua ngay, giúp người dùng dễ dàng tiếp tục quá trình đặt hàng. Khi người dùng lựa chọn chức năng thêm sản phẩm vào giỏ hàng, hệ thống sẽ ghi nhận thông tin sản phẩm và số lượng tương ứng vào giỏ hàng của người dùng.

Trong trường hợp sản phẩm không tồn tại, bị xóa hoặc không còn khả dụng, hệ thống sẽ hiển thị thông báo phù hợp nhằm đảm bảo người dùng nhận được phản hồi rõ ràng trong quá trình sử dụng.

Chức năng xem chi tiết sản phẩm được thiết kế nhằm cung cấp đầy đủ thông tin cần thiết, đảm bảo tính trực quan và thuận tiện cho người dùng trong quá trình tìm hiểu và lựa chọn sản phẩm trên website.

* Đặc tả chức năng quản lý giỏ hàng:

The screenshot shows a shopping cart page with the following details:

SẢN PHẨM	BỘN GIÁ	SỐ LƯỢNG	TỔNG CỘNG
	335.000 VNĐ	<input type="button" value="-"/> <input type="text" value="2"/> <input type="button" value="+"/>	670.000 VNĐ

Below the table, there is a summary table:

Thành tiền	670.000 VNĐ
Giảm giá	0 VNĐ
Tổng đơn hàng	670.000 VNĐ

* Lưu ý: Phí vận chuyển được tính khi thanh toán

Hình 4: Giao diện quản lý giỏ hàng

Chức năng quản lý giỏ hàng cho phép người dùng theo dõi và điều chỉnh các sản phẩm đã lựa chọn trước khi tiến hành đặt hàng trên website Savourebakery. Chức năng này đóng vai trò trung gian giữa việc lựa chọn sản phẩm và quá trình thanh toán.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

Sau khi người dùng thêm sản phẩm từ trang chi tiết sản phẩm vào giỏ hàng, hệ thống sẽ tự động cập nhật giỏ hàng và điều hướng người dùng đến trang Giỏ hàng hoặc hiển thị thông báo xác nhận thêm sản phẩm thành công. Tại trang giỏ hàng, hệ thống hiển thị danh sách các sản phẩm đã được thêm vào, bao gồm hình ảnh sản phẩm, tên sản phẩm, giá bán, số lượng và tổng tiền tương ứng cho từng sản phẩm.

Người dùng có thể thay đổi số lượng sản phẩm trong giỏ hàng bằng cách tăng hoặc giảm số lượng. Khi số lượng sản phẩm được thay đổi, hệ thống sẽ tự động cập nhật lại tổng tiền của từng sản phẩm cũng như tổng giá trị đơn hàng trong giỏ.

Ngoài ra, chức năng giỏ hàng cho phép người dùng xóa sản phẩm khỏi giỏ hàng trong trường hợp không còn nhu cầu mua sản phẩm đó. Sau khi xóa, hệ thống sẽ cập nhật lại danh sách sản phẩm và tổng giá trị giỏ hàng theo thời gian thực.

Trong trường hợp giỏ hàng trống, hệ thống sẽ hiển thị thông báo phù hợp nhằm hướng dẫn người dùng tiếp tục mua sắm. Khi người dùng đã hoàn tất việc kiểm tra giỏ hàng, có thể tiếp tục quá trình đặt hàng bằng cách chuyển sang bước thanh toán theo quy trình của hệ thống.

Chức năng quản lý giỏ hàng giúp người dùng kiểm soát được nội dung đơn hàng trước khi thanh toán, đảm bảo tính linh hoạt và thuận tiện trong quá trình mua sắm trực tuyến trên website Savourebakery.

*** Đặc tả chức năng cập nhật thông tin người dùng:**

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

CẬP NHẬT THÔNG TIN

Địa chỉ email
meomeo40021140@gmail.com

Họ
Meo

Tên
Mimi

Điện thoại
0999999999

Tỉnh/ Thành phố
TP. Hồ Chí Minh

Quận/ Huyện
Quận Tân Bình

Phường/ Xã
Phường 4

Địa chỉ
Somewhere

Cập Nhật

Hình 5 Giao diện cập nhật thông tin

Chức năng cập nhật thông tin người dùng cho phép người dùng đã đăng nhập vào hệ thống chỉnh sửa và cập nhật các thông tin cá nhân của mình nhằm đảm bảo dữ liệu tài khoản luôn chính xác và đầy đủ trong quá trình sử dụng website Savourebakery.

Sau khi đăng nhập thành công, người dùng có thể truy cập chức năng Cập nhật thông tin cá nhân thông qua khu vực tài khoản thành viên. Hệ thống sẽ điều hướng người dùng đến trang cập nhật thông tin, tại đó hiển thị các trường thông tin cá nhân như họ tên, số điện thoại và địa chỉ liên hệ.

Người dùng có thể chỉnh sửa các trường thông tin này và nhấn nút xác nhận cập nhật để lưu thay đổi. Khi người dùng thực hiện cập nhật, hệ thống sẽ tiếp nhận dữ liệu đầu vào và tiến hành xử lý lưu trữ thông tin mới.

Trong trường hợp dữ liệu nhập vào hợp lệ, hệ thống sẽ cập nhật thông tin thành công và hiển thị thông báo xác nhận cho người dùng. Ngược lại, nếu dữ liệu không hợp lệ hoặc thiếu thông tin cần thiết, hệ thống sẽ hiển thị thông báo phù hợp nhằm yêu cầu người dùng kiểm tra và nhập lại thông tin.

Chức năng cập nhật thông tin cá nhân giúp người dùng chủ động quản lý dữ liệu cá nhân của mình, đồng thời hỗ trợ hệ thống sử dụng thông tin chính xác trong các hoạt động liên quan như liên hệ, giao hàng và chăm sóc khách hàng.

3.3 Các yêu cầu phi chức năng

Tên yêu cầu	Mô tả
Hiệu năng	<ul style="list-style-type: none"> - Hệ thống phản hồi trong vòng ≤ 3 giây cho các thao tác chính (tìm kiếm sản phẩm, thêm vào giỏ hàng, đăng nhập). - Hỗ trợ tối thiểu 100 người dùng truy cập đồng thời
Bảo mật	<ul style="list-style-type: none"> - Thông tin tài khoản và mật khẩu phải được mã hóa. - Hệ thống chỉ cho phép người dùng đã xác thực thực hiện các chức năng liên quan đến tài khoản cá nhân và đặt hàng. - Có cơ chế phân quyền quản trị viên và người dùng thông thường.
Khả năng sử dụng	<ul style="list-style-type: none"> - Giao diện thân thiện, dễ thao tác, hiển thị rõ ràng trên cả máy tính và thiết bị di động. - Các nút chức năng và thông báo lỗi phải trực quan, dễ hiểu.
Tương thích	<ul style="list-style-type: none"> - Hoạt động ổn định trên các trình duyệt phổ biến (Chrome, Firefox, Edge, Safari). - Tương thích với cả thiết bị di động và máy tính để bàn.
Bảo trì & Mở rộng	<ul style="list-style-type: none"> - Mã nguồn được thiết kế theo mô-đun, dễ dàng mở rộng thêm chức năng mới. - Có thể nâng cấp hệ thống để đáp ứng lượng truy cập tăng gấp 3 lần mà không ảnh hưởng hiệu năng.
Độ tin cậy	<ul style="list-style-type: none"> - Hệ thống hoạt động ổn định $\geq 99\%$ thời gian. - Có cơ chế sao lưu dữ liệu định kỳ (hàng ngày/tuần) để tránh mất mát dữ liệu.
Khả dụng	<ul style="list-style-type: none"> - Hệ thống sẵn sàng hoạt động 24/7. - Thời gian bảo trì được thông báo trước tối thiểu 24 giờ.
Khả chuyển	<ul style="list-style-type: none"> - Có thể triển khai trên nhiều môi trường (Windows, Linux). - Dữ liệu có thể di chuyển sang hệ thống khác mà không bị mất mát.
Tuân thủ	<ul style="list-style-type: none"> - Đáp ứng quy định pháp luật Việt Nam về bảo vệ dữ liệu cá nhân

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

	(Nghị định 13/2023/NĐ-CP). - Tuân thủ quy định về thương mại điện tử.
Hỗ trợ & Tài liệu	- Cung cấp tài liệu hướng dẫn sử dụng cho khách hàng. - Có tài liệu kỹ thuật để hỗ trợ đội phát triển và bảo trì hệ thống.

Bảng 3: Các yêu cầu phi chức năng

3.4 Tính ứng dụng

Đối tượng	Nội dung ứng dụng
Doanh nghiệp	<ul style="list-style-type: none"> - Phát hiện nhanh lỗi trong quá trình vận hành website. - Đảm bảo hệ thống hoạt động ổn định, nâng cao trải nghiệm người dùng. - Giảm thiểu chi phí nhân lực, tiết kiệm thời gian so với kiểm thử thủ công.
Người dùng	<ul style="list-style-type: none"> - Các chức năng (đăng nhập, đăng ký, tìm kiếm, đặt hàng, thanh toán, quản lý tài khoản) hoạt động ổn định. - Trải nghiệm mua sắm trực tuyến thuận tiện, nhanh chóng và an toàn.
Nghiên cứu và đào tạo	<ul style="list-style-type: none"> - Làm tài liệu tham khảo trong giảng dạy và học tập về kiểm thử phần mềm. - Minh chứng thực tế cho việc áp dụng Cypress vào kiểm thử website thương mại điện tử.
Khả năng mở rộng	<ul style="list-style-type: none"> - Framework có thể tùy chỉnh và áp dụng cho nhiều website thương mại điện tử khác có cấu trúc tương tự.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

Bảng 4: Tính ứng dụng

3.5 Tính toàn vẹn dữ liệu

Khía cạnh	Nội dung đảm bảo
Toàn vẹn cú pháp	<ul style="list-style-type: none"> - Các trường dữ liệu (Email, Số điện thoại, Mật khẩu, Địa chỉ) được kiểm tra đúng định dạng trước khi lưu vào hệ thống. - Tránh tình trạng nhập sai, nhập thiếu hoặc ký tự không hợp lệ.
Toàn vẹn tham chiếu	<ul style="list-style-type: none"> - Mỗi bản ghi dữ liệu (người dùng, sản phẩm, đơn hàng) được liên kết chặt chẽ với các bảng liên quan trong cơ sở dữ liệu. - Ví dụ: Một đơn hàng phải tham chiếu đúng đến khách hàng và sản phẩm tương ứng.
Toàn vẹn giao dịch	<ul style="list-style-type: none"> - Các thao tác như đặt hàng, thanh toán, cập nhật thông tin tài khoản được thực hiện theo cơ chế tất cả hoặc không (Atomic). - Đảm bảo không có tình trạng đơn hàng lưu thiếu thông tin hoặc thanh toán không khớp.
Toàn vẹn bảo mật	<ul style="list-style-type: none"> - Thông tin nhạy cảm (mật khẩu, dữ liệu thanh toán) được mã hóa và bảo vệ khỏi truy cập trái phép. - Quyền truy cập dữ liệu được phân quyền theo vai trò (người dùng, quản trị viên).
Toàn vẹn nhất quán	<ul style="list-style-type: none"> - Dữ liệu được cập nhật đồng bộ giữa các module trong hệ thống. - Ví dụ: Khi sản phẩm đã hết hàng, hệ thống sẽ tự động cập nhật trong kho và ngăn chặn đặt hàng vượt quá số lượng.

Bảng 5: Tính toàn vẹn dữ liệu

CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG

4.1 Đặt vấn đề

Website Savourebakery.com là một ứng dụng web thương mại điện tử với nhiều chức năng quan trọng như đăng nhập, tìm kiếm sản phẩm, xem chi tiết sản phẩm, quản lý giỏ hàng và cập nhật thông tin cá nhân. Để đảm bảo các chức năng này hoạt động đúng yêu cầu và ổn định trong quá trình vận hành, việc kiểm thử phần mềm là cần thiết.

Trong thực tế, kiểm thử thủ công gặp nhiều hạn chế như tốn thời gian, khó lặp lại và khó kiểm soát khi số lượng trường hợp kiểm thử tăng lên. Do đó, kiểm thử tự động được lựa chọn nhằm nâng cao hiệu quả kiểm thử và giảm thiểu sai sót.

Đề tài “Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com” tập trung vào việc xây dựng một framework kiểm thử tự động dựa trên công cụ Playwright kết hợp với Pytest, áp dụng phương pháp kiểm thử hướng dữ liệu (Data Driven Testing) thông qua file Excel. Framework cho phép tổ chức mã nguồn rõ ràng, dễ mở rộng và tái sử dụng cho nhiều chức năng khác nhau của website.

Việc triển khai kiểm thử tự động giúp thực thi các kịch bản kiểm thử một cách nhanh chóng, ghi nhận kết quả kiểm thử và hỗ trợ phát hiện lỗi trong hệ thống, từ đó góp phần đánh giá chất lượng và độ ổn định của website Savourebakery.com.

4.2 Xây dựng framework kiểm thử tự động

4.2.1 Mô hình framework kiểm thử

Framework kiểm thử tự động trong đề tài được xây dựng theo mô hình Page Object Model (POM) kết hợp với Pytest Framework và Playwright.

Trong mô hình này:

- Mỗi trang hoặc chức năng của website được biểu diễn bằng một Page Object.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

- Các test case chỉ tập trung vào kịch bản kiểm thử, không phụ thuộc trực tiếp vào locator hay chi tiết giao diện.
- Dữ liệu kiểm thử được tách riêng và quản lý bằng file Excel, áp dụng phương pháp Data Driven Testing (DDT).

Mô hình framework giúp:

- Tăng khả năng tái sử dụng mã nguồn
- Dễ bảo trì khi giao diện website thay đổi
- Dễ mở rộng thêm test case và chức năng mới

4.2.2. Cấu trúc thư mục framework

Framework kiểm thử tự động được tổ chức theo cấu trúc thư mục như sau:

DOAN4/

```
|  
|   __pycache__/      # Cache của Python  
|   .pytest_cache/    # Cache của PyTest  
|  
|   data/             # Dữ liệu test (test data)  
|       |   login_data.csv      # Dữ liệu đăng nhập  
|       |   profile_update_data.xlsx # Dữ liệu cập nhật hồ sơ  
|       |   search_data.json     # Dữ liệu tìm kiếm  
|  
|   pages/            # Các Page Object  
|       |   __pycache__/  
|       |   cart_page.py      # Trang giỏ hàng  
|       |   home_page.py      # Trang trang chủ
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
|   ├── login_page.py      # Trang đăng nhập
|   ├── product_detail_page.py  # Trang chi tiết sản phẩm
|   ├── product_page.py      # Trang danh sách sản phẩm
|   ├── profile_page.py      # Trang hồ sơ người dùng
|   └── search_page.py      # Trang tìm kiếm
|
|   └── reports/          # Báo cáo kết quả test
|
|   └── templates/        # Template báo cáo (HTML, v.v.)
|
|   └── tests/            # Các test case
|       ├── __pycache__/
|       ├── test_cart.py    # Test chức năng giỏ hàng
|       ├── test_demo.py    # Test demo / thử nghiệm
|       ├── test_home.py    # Test trang chủ
|       ├── test_login.py   # Test đăng nhập
|       ├── test_product_detail.py  # Test chi tiết sản phẩm
|       ├── test_profile.py  # Test hồ sơ người dùng
|       └── test_search.py   # Test tìm kiếm
|
|   └── utils/             # Tiện ích dùng chung
|       ├── __pycache__/
|       └── data_loader.py  # Hàm đọc dữ liệu (CSV, Excel, JSON)
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
|── venv/          # Môi trường ảo Python  
|  
|── conftest.py    # Fixture PyTest (setup/teardown, driver)  
|── pytest.ini     # Cấu hình PyTest  
|── README.md      # Tài liệu mô tả framework  
└── requirements.txt # Danh sách thư viện cần cài
```

4.2.3. Giải thích chức năng từng thư mục và file

* Thư mục pages/

Chứa các file Page Object, mỗi file đại diện cho một trang hoặc chức năng của website.

cart_page.py: Định nghĩa locator và các hành động liên quan đến chức năng quản lý giỏ hàng.

profile_page.py: Định nghĩa locator và hành động cho chức năng cập nhật thông tin cá nhân.

product_page.py: Quản lý các thao tác liên quan đến sản phẩm và trang chi tiết sản phẩm.

* Thư mục tests/

Chứa các file test case, mỗi file tương ứng với một chức năng của hệ thống.

test_cart.py: Thực hiện kiểm thử tự động cho chức năng giỏ hàng.

test_profile.py: Kiểm thử chức năng cập nhật thông tin cá nhân bằng phương pháp Data Driven Testing.

test_product.py: Kiểm thử các chức năng liên quan đến sản phẩm.

* Thư mục utils/

Chứa các hàm tiện ích dùng chung cho toàn bộ framework.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

data_loader.py: Đọc dữ liệu kiểm thử từ file Excel và trả về cho Pytest sử dụng trong @pytest.mark.parametrize.

* Thư mục test_data/

Chứa dữ liệu kiểm thử tách biệt khỏi mã nguồn.

cart_data.xlsx: Dữ liệu test cho chức năng giỏ hàng.

profile_update_data.xlsx: Dữ liệu test cho chức năng cập nhật thông tin cá nhân.

* Thư mục reports/

Lưu trữ kết quả kiểm thử.

screenshots/: Tự động lưu ảnh chụp màn hình khi test case thất bại (FAIL).

* Thư mục templates/

Dùng để lưu các mẫu tài liệu báo cáo hoặc template hỗ trợ việc tổng hợp và log bug từ kết quả kiểm thử.

* File conftest.py

Chứa các fixture dùng chung cho toàn bộ test case, bao gồm:

Khởi tạo Playwright, browser và page

Fixture đăng nhập cho chức năng cập nhật thông tin cá nhân

Hook chụp screenshot khi test case FAIL

* File pytest.ini

Cấu hình Pytest, hỗ trợ quản lý marker, log và tùy chỉnh khi thực thi kiểm thử.

* File requirements.txt

Khai báo các thư viện cần thiết để cài đặt và chạy framework kiểm thử tự động.

4.3 Thực thi và báo cáo kiểm thử

4.3.1 Thiết kế kiểm thử

Việc thiết kế kiểm thử trong đề tài được thực hiện theo các bước sau:

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

❖ Phân tích yêu cầu (Requirement Analysis):

Dựa trên yêu cầu nghiệp vụ của website Savourebakery.com, các chức năng chính được lựa chọn để kiểm thử bao gồm:

- Đăng nhập hệ thống
- Tìm kiếm sản phẩm
- Xem chi tiết sản phẩm
- Quản lý giỏ hàng
- Cập nhật thông tin cá nhân

Các chức năng này là những chức năng cốt lõi, ảnh hưởng trực tiếp đến trải nghiệm người dùng trên website.

❖ Xác định phạm vi kiểm thử (Scope Definition):

Phạm vi kiểm thử tập trung vào:

- Kiểm thử giao diện người dùng (UI Testing)
- Kiểm thử chức năng (Functional Testing)

Để tài không đi sâu vào kiểm thử hiệu năng, bảo mật hay kiểm thử API, mà tập trung đánh giá tính đúng đắn của các chức năng chính thông qua thao tác trên trình duyệt.

❖ Xác định mục tiêu kiểm thử (Test Objectives):

Mục tiêu của quá trình kiểm thử là:

- Đảm bảo người dùng có thể thao tác đúng các chức năng trên website
- Kiểm tra tính hợp lệ của dữ liệu nhập vào các form
- Xác minh hệ thống phản hồi đúng với các trường hợp dữ liệu hợp lệ và không hợp lệ
- Phát hiện lỗi phát sinh trong quá trình sử dụng thực tế để làm cơ sở ghi nhận bug

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

❖ Xác định công cụ kiểm thử (Tool Selection):

Các công cụ được sử dụng trong quá trình kiểm thử tự động bao gồm:

- Playwright: điều khiển trình duyệt, mô phỏng hành vi người dùng khi thao tác trên website.

- Pytest: tổ chức và thực thi các test case, hỗ trợ tham số hóa dữ liệu kiểm thử theo mô hình Data Driven Testing.

- Allure Report: tạo báo cáo kết quả kiểm thử, hiển thị trạng thái PASS/FAIL và ảnh chụp màn hình khi test thất bại.

- OpenPyXL: đọc dữ liệu kiểm thử từ file Excel phục vụ cho việc xây dựng test case.

Kết quả là framework kiểm thử có khả năng tự động thực thi các kịch bản kiểm thử, so sánh kết quả thực tế với kết quả mong đợi và hỗ trợ ghi nhận lỗi một cách trực quan.

4.3.2 Xây dựng trường hợp kiểm thử

a. Testcase cho chức năng Đăng nhập

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TC_01	Đăng nhập khi bỏ trống Email	Người dùng đang ở trang Đăng nhập	1. Truy cập trang https://savourebake ry.com/thanh-vien/ 2. Đề trống trường Email 3. Nhập mật khẩu bất kỳ 4. Click nút Đăng Nhập	Hệ thống không cho đăng nhập Trường Email hiển thị trạng thái không hợp lệ (validation error)

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

TC_02	Đăng nhập với Email sai định dạng (không có @)	Người dùng đang ở trang Đăng nhập	1. Nhập Email: abc 2. Nhập mật khẩu bất kỳ 3. Click Đăng Nhập	Hệ thống không cho đăng nhập Trường Email hiển thị lỗi định dạng
TC_03	Đăng nhập với Email sai định dạng (thiếu domain)	Người dùng đang ở trang Đăng nhập	1. Nhập Email: abc@gmail 2. Nhập mật khẩu bất kỳ 3. Click Đăng Nhập	Hệ thống không cho đăng nhập Trình duyệt hiển thị lỗi xác thực Email
TC_04	Đăng nhập với Email không tồn tại trong hệ thống	Người dùng đang ở trang Đăng nhập	1. Nhập Email: notexist@gmail.com 2. Nhập mật khẩu hợp lệ 3. Click Đăng Nhập	Hệ thống hiển thị thông báo lỗi: Email không tồn tại
TC_05	Đăng nhập với Email đúng nhưng mật khẩu sai	Người dùng đang ở trang Đăng nhập	1. Nhập Email hợp lệ đã đăng ký 2. Nhập mật khẩu sai 3. Click Đăng Nhập	Hệ thống hiển thị thông báo lỗi: Password không đúng

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_06	Đăng nhập thành công với Email và mật khẩu hợp lệ	Người dùng đã có tài khoản hợp lệ	1. Nhập Email hợp lệ 2. Nhập đúng mật khẩu 3. Click Đăng Nhập	Hệ thống đăng nhập thành công Người dùng được chuyển đến trang tài khoản cá nhân
-------	---	-----------------------------------	---	--

Bảng 6: Testcase chức năng đăng nhập

b. Testcase cho chức năng Tìm kiếm sản phẩm

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TC_01	Tìm kiếm với từ khóa hợp lệ có dấu	Người dùng đang ở trang chủ website	1. Truy cập trang https://savourebakery.com 2. Click biểu tượng tìm kiếm 3. Nhập từ khóa “bắp” 4. Click nút tìm kiếm	Hệ thống hiển thị danh sách sản phẩm liên quan đến từ khóa
TC_02	Tìm kiếm với từ khóa viết hoa	Người dùng đang ở trang chủ website	1. Mở popup tìm kiếm 2. Nhập từ khóa “BẮP” 3. Thực hiện tìm kiếm	Hệ thống không phân biệt hoa/thường Kết quả sản phẩm được hiển thị

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_03	Tìm kiếm với từ khóa viết hoa – thường xen kẽ	Người dùng đang ở trang chủ website	1. Nhập từ khóa “bAp” 2. Click tìm kiếm	Hệ thống trả về kết quả sản phẩm phù hợp
TC_04	Tìm kiếm với tên sản phẩm đầy đủ	Người dùng đang ở trang chủ website	1. Nhập từ khóa “bánh kem báp” 2. Thực hiện tìm kiếm	Hiển thị chính xác sản phẩm tương ứng
TC_05	Tìm kiếm với từ khóa không dấu	Người dùng đang ở trang chủ website	1. Nhập từ khóa “banh kem bap” 2. Click tìm kiếm	Hệ thống vẫn hiển thị sản phẩm liên quan
TC_06	Tìm kiếm với từ khóa tiếng Anh không dấu	Người dùng đang ở trang chủ website	1. Nhập từ khóa “bap” 2. Thực hiện tìm kiếm	Kết quả tìm kiếm được hiển thị
TC_07	Tìm kiếm với từ khóa chung	Người dùng đang ở trang chủ website	1. Nhập từ khóa “bánh” 2. Click tìm kiếm	Hiển thị nhiều sản phẩm phù hợp
TC_08	Tìm kiếm với từ khóa phổ biến	Người dùng đang ở trang chủ website	1. Nhập từ khóa “kem” 2. Thực hiện tìm kiếm	Danh sách sản phẩm được hiển thị

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

TC_09	Tìm kiếm với từ khóa không tồn tại	Người dùng đang ở trang chủ website	1. Nhập từ khóa “abcdefxyz” 2. Click tìm kiếm	Không hiển thị sản phẩm nào
TC_10	Tìm kiếm với từ khóa chỉ gồm số	Người dùng đang ở trang chủ website	1. Nhập từ khóa “123456” 2. Thực hiện tìm kiếm	Không có sản phẩm được hiển thị
TC_11	Tìm kiếm với ký tự đặc biệt	Người dùng đang ở trang chủ website	1. Nhập từ khóa “@#\$%^&*” 2. Click tìm kiếm	Hệ thống không trả về kết quả
TC_12	Tìm kiếm với từ khóa pha chữ và ký tự đặc biệt	Người dùng đang ở trang chủ website	1. Nhập từ khóa “băp@123” 2. Thực hiện tìm kiếm	Không có sản phẩm phù hợp
TC_13	Tìm kiếm với từ khóa chỉ gồm khoảng trắng	Người dùng đang ở trang chủ website	1. Nhập khoảng trắng 2. Click tìm kiếm	Hệ thống hiển thị toàn bộ danh sách sản phẩm
TC_14	Tìm kiếm khi không nhập từ khóa	Người dùng đang ở trang chủ website	1. Để trống ô tìm kiếm 2. Click tìm kiếm	Hiển thị toàn bộ sản phẩm

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_15	Tìm kiếm với từ khóa dài, lặp lại	Người dùng đang ở trang chủ website	1. Nhập từ khóa “bắp bắp bắp bắp bắp” 2. Thực hiện tìm kiếm	Không tìm thấy sản phẩm phù hợp
-------	-----------------------------------	-------------------------------------	--	---------------------------------

Bảng 7: Testcase chức năng tìm kiếm sản phẩm

c. Testcase cho chức năng Xem chi tiết sản phẩm

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TC_01	Truy cập trang chi tiết sản phẩm thành công	Website hoạt động bình thường	1. Truy cập URL chi tiết sản phẩm https://savourebakery.com/san-pham/banh-bong-lan-cuon-tra-ba-tuoc-224.html	Trang chi tiết sản phẩm được hiển thị
TC_02	Hiển thị tên sản phẩm	Đang ở trang chi tiết sản phẩm	1. Quan sát tiêu đề sản phẩm	Tên sản phẩm được hiển thị và không rỗng
TC_03	Hiển thị mã sản phẩm	Đang ở trang chi tiết sản phẩm	1. Kiểm tra thông tin mã sản phẩm	Mã sản phẩm được hiển thị
TC_04	Hiển thị số lượt mua	Đang ở trang chi tiết sản phẩm	1. Quan sát thông tin số lượt mua	Thông tin “đã mua” được hiển thị

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_05	Hiển thị hình ảnh sản phẩm	Đang ở trang chi tiết sản phẩm	1. Quan sát khu vực hình ảnh sản phẩm	Hình ảnh sản phẩm hiển thị đầy đủ
TC_06	Hiển thị giá sản phẩm hợp lệ	Đang ở trang chi tiết sản phẩm	1. Quan sát giá sản phẩm	Giá sản phẩm được hiển thị và chứa giá trị số
TC_07	Số lượng mặc định khi mở trang	Đang ở trang chi tiết sản phẩm	1. Quan sát ô số lượng	Giá trị mặc định là 1
TC_08	Tăng số lượng sản phẩm	Đang ở trang chi tiết sản phẩm	1. Click nút tăng số lượng (+)	Số lượng tăng lên tương ứng
TC_09	Giảm số lượng sản phẩm	Đang ở trang chi tiết sản phẩm	1. Click nút giảm số lượng (-)	Số lượng giảm xuống tương ứng
TC_10	Không cho phép số lượng nhỏ hơn 1	Đang ở trang chi tiết sản phẩm	1. Click nút giảm số lượng khi đang là 1	Số lượng vẫn giữ nguyên là 1
TC_11	Thêm sản phẩm vào giỏ hàng	Đang ở trang chi tiết sản phẩm	1. Click nút “Thêm vào giỏ hàng”	Thao tác được thực hiện thành công, không lỗi
TC_12	Giữ nguyên trang sau khi thêm vào giỏ	Đã click “Thêm vào giỏ hàng”	1. Quan sát URL sau khi thao tác	Người dùng vẫn ở trang chi tiết sản phẩm

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_13	Kiểm tra nút “Mua ngay” hiển thị	Đang ở trang chi tiết sản phẩm	1. Quan sát nút “Mua ngay”	Nút “Mua ngay” hiển thị và có thể thao tác
-------	----------------------------------	--------------------------------	----------------------------	--

Bảng 8: Testcase chức năng xem chi tiết sản phẩm

d. Testcase cho chức năng Quản lý giỏ hàng

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TC_01	Truy cập trang giỏ hàng thành công	Người dùng đã thêm ít nhất 1 sản phẩm vào giỏ hàng	1. Truy cập trang giỏ hàng 2. Chờ bảng giỏ hàng hiển thị	Bảng giỏ hàng hiển thị thành công
TC_02	Giỏ hàng có sản phẩm	Giỏ hàng đã có sản phẩm	1. Quan sát danh sách sản phẩm trong giỏ hàng	Có ít nhất 1 sản phẩm trong giỏ
TC_03	Hiển thị tên sản phẩm trong giỏ	Giỏ hàng có sản phẩm	1. Quan sát cột tên sản phẩm	Tên sản phẩm hiển thị và không rỗng
TC_04	Hiển thị giá sản phẩm	Giỏ hàng có sản phẩm	1. Quan sát cột giá sản phẩm	Giá sản phẩm hiển thị đúng định dạng tiền tệ (VND)
TC_05	Số lượng mặc định của sản phẩm	Giỏ hàng có sản phẩm	1. Quan sát ô số lượng	Số lượng mặc định là 1

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

TC_06	Tăng số lượng sản phẩm	Giỏ hàng có sản phẩm	1. Click nút tăng số lượng (+)	Số lượng sản phẩm tăng lên
TC_07	Giảm số lượng sản phẩm	Số lượng sản phẩm lớn hơn 1	1. Click nút giảm số lượng (-)	Số lượng sản phẩm giảm xuống
TC_08	Không cho phép số lượng nhỏ hơn 1	Số lượng sản phẩm đang là 1	1. Click nút giảm số lượng	Số lượng không nhỏ hơn 1
TC_09	Thành tiền thay đổi khi thay đổi số lượng	Giỏ hàng có sản phẩm	1. Ghi nhận thành tiền ban đầu 2. Tăng số lượng sản phẩm	Thành tiền được cập nhật theo số lượng mới
TC_10	Hiển thị tổng tiền đơn hàng	Giỏ hàng có sản phẩm	1. Quan sát tổng tiền đơn hàng	Tổng tiền hiển thị đúng định dạng VNĐ
TC_11	Tổng tiền bằng thành tiền khi chỉ có 1 sản phẩm	Giỏ hàng chỉ có 1 sản phẩm	1. So sánh tổng tiền và thành tiền	Tổng tiền bằng thành tiền
TC_12	Xóa sản phẩm khỏi giỏ hàng	Giỏ hàng có sản phẩm	1. Click nút xóa sản phẩm	Sản phẩm bị xóa khỏi giỏ

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_13	Giỏ hàng trống sau khi xóa sản phẩm	Sản phẩm đã bị xóa	1. Quan sát giỏ hàng	Không còn sản phẩm, tổng tiền = 0 VNĐ
-------	-------------------------------------	--------------------	----------------------	---------------------------------------

Bảng 9: Testcase chức năng quản lý giỏ hàng

e. Testcase cho chức năng Cập nhật thông tin người dùng

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TC_01	Cập nhật thông tin với dữ liệu hợp lệ đầy đủ	Người dùng đã đăng nhập	1. Truy cập trang cập nhật thông tin 2. Nhập họ, tên, số điện thoại, địa chỉ hợp lệ 3. Click Cập nhật	Thông tin được cập nhật thành công, không hiển thị lỗi
TC_02	Bỏ trống họ	Người dùng đã đăng nhập	1. Để trống trường họ 2. Nhập các trường còn lại 3. Click Cập nhật	Hệ thống vẫn cho phép cập nhật
TC_03	BỎ trống tên	Người dùng đã đăng nhập	1. Để trống trường tên 2. Nhập các trường còn lại 3. Click Cập nhật	Thông tin được cập nhật, không báo lỗi

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

TC_04	BỎ TRỐNG SỐ ĐIỆN THOẠI	Người dùng đã đăng nhập	1. Đèn trống số điện thoại 2. Nhập các trường khác 3. Click Cập nhật	Cập nhật thành công
TC_05	BỎ TRỐNG ĐỊA CHỈ	Người dùng đã đăng nhập	1. Đèn trống địa chỉ 2. Nhập các trường khác 3. Click Cập nhật	Cập nhật thành công
TC_06	NHẬP SỐ ĐIỆN THOẠI KHÔNG CHỮ	Người dùng đã đăng nhập	1. Nhập số điện thoại chứa ký tự chữ 2. Click Cập nhật	Hệ thống không hiển thị lỗi
TC_07	NHẬP SỐ ĐIỆN THOẠI CHỮ VÀ SỐ	Người dùng đã đăng nhập	1. Nhập số điện thoại pha chữ và số 2. Click Cập nhật	Thông tin được lưu
TC_08	NHẬP SỐ ĐIỆN THOẠI NGẮN HƠN CHUẨN	Người dùng đã đăng nhập	1. Nhập số điện thoại ít chữ số 2. Click Cập nhật	Hệ thống vẫn chấp nhận
TC_09	NHẬP SỐ ĐIỆN THOẠI DÀI HƠN CHUẨN	Người dùng đã đăng nhập	1. Nhập số điện thoại nhiều chữ số 2. Click Cập nhật	Cập nhật thành công

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử
cho website Savourebakery.com

TC_10	Nhập số điện thoại có khoảng trắng	Người dùng đã đăng nhập	1. Nhập số điện thoại chứa khoảng trắng 2. Click Cập nhật	Không hiển thị lỗi
TC_11	Nhập số điện thoại ký tự đặc biệt	Người dùng đã đăng nhập	1. Nhập số điện thoại bằng ký tự đặc biệt 2. Click Cập nhật	Hệ thống vẫn cho cập nhật
TC_12	Nhập số điện thoại toàn số 0	Người dùng đã đăng nhập	1. Nhập số điện thoại 0000000000 2. Click Cập nhật	Không có lỗi xác thực
TC_13	Nhập họ có số	Người dùng đã đăng nhập	1. Nhập họ chứa số 2. Click Cập nhật	Thông tin được cập nhật
TC_14	Nhập tên có số	Người dùng đã đăng nhập	1. Nhập tên chứa số 2. Click Cập nhật	Không hiển thị lỗi
TC_15	Nhập họ có dấu tiếng Việt	Người dùng đã đăng nhập	1. Nhập họ có dấu 2. Click Cập nhật	Cập nhật thành công
TC_16	Nhập tên có khoảng trắng	Người dùng đã đăng nhập	1. Nhập tên có khoảng trắng 2. Click Cập nhật	Thông tin được lưu

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC_17	Nhập địa chỉ tiếng Việt	Người dùng đã đăng nhập	1. Nhập địa chỉ tiếng Việt 2. Click Cập nhật	Cập nhật thành công
TC_18	Nhập địa chỉ khác khu vực	Người dùng đã đăng nhập	1. Nhập địa chỉ khác tỉnh/thành 2. Click Cập nhật	Không xảy ra lỗi
TC_19	Nhập địa chỉ tiếng Anh	Người dùng đã đăng nhập	1. Nhập địa chỉ tiếng Anh 2. Click Cập nhật	Cập nhật thành công
TC_20	Nhập địa chỉ bất kỳ	Người dùng đã đăng nhập	1. Nhập địa chỉ bất kỳ 2. Click Cập nhật	Hệ thống không hiển thị lỗi

Bảng 10: Testcase chức năng cập nhật thông tin cá nhân

* Dữ liệu kiểm thử

- File CSV:

```
email, password, expected, type
, test123, invalid, validation
abc,123456, invalid, validation
abc@gmail,123456, invalid, validation
notexist@gmail.com,123456, Email không tồn tại!, backend
meomeo40021140@gmail.com, wrongpass, Password không đúng!, backend
```

- File excel:

tc_id	first_name	last_name	phone	address	expect_valid
TC01	Meo	Mimi	0987654321	Trên trái đất	PASS
TC02		Mimi	0987654321	Trên trái đất	PASS

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TC03	Meo		0987654321	Trên trái đất	PASS
TC04	Meo	Mimi		Trên trái đất	PASS
TC05	Meo	Mimi	0987654321		PASS
TC06	Meo	Mimi	abc123	Trên trái đất	PASS
TC07	Meo	Mimi	09876abc	Trên trái đất	PASS
TC08	Meo	Mimi	098765432	Trên trái đất	PASS
TC09	Meo	Mimi	0987654321123	Trên trái đất	PASS
TC10	Meo	Mimi	09876 54321	Trên trái đất	PASS
TC11	Meo	Mimi	@#\$%^&*	Trên trái đất	PASS
TC12	Meo	Mimi	0000000000	Trên trái đất	PASS
TC13	Meo123	Mimi	0987654321	Trên trái đất	PASS
TC14	Meo	Mimi123	0987654321	Trên trái đất	PASS
TC15	Mèo	Mimi	0987654321	Trên trái đất	PASS
TC16	Meo	Mi mi	0987654321	Trên trái đất	PASS
TC17	Meo	Mimi	0987654321	Hà Nội	PASS
TC18	Meo	Mimi	0912345678	Sài Gòn	PASS
TC19	Meo	Mimi	0900000000	Earth	PASS
TC20	Meo	Mimi	0999999999	Somewhere	PASS

- File JSON:

```
[  
  {  
    "tc_id": "SEARCH_01",  
    "keyword": "bắp",  
    "expected": "found",  
    "description": "Từ khóa hợp lệ, có nhiều sản phẩm"  
  },  
  {  
    "tc_id": "SEARCH_02",  
    "keyword": "BẮP",  
    "expected": "found",  
    "description": "Từ khóa viết hoa"  
  },  
  {  
    "tc_id": "SEARCH_03",  
    "keyword": "bAp",  
    "expected": "found",  
    "description": "Từ khóa mix kí tự thường và hoa"  
  }]
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
"expected": "found",
"description": "Từ khóa viết hoa – thường lẫn lộn"
},
{
"tc_id": "SEARCH_04",
"keyword": "bánh kem bắp",
"expected": "found",
"description": "Từ khóa đầy đủ tên sản phẩm"
},
{
"tc_id": "SEARCH_05",
"keyword": "banh kem bap",
"expected": "found",
"description": "Từ khóa không dấu"
},
{
"tc_id": "SEARCH_06",
"keyword": "bap",
"expected": "found",
"description": "Từ khóa tiếng Anh không dấu"
},
{
"tc_id": "SEARCH_07",
"keyword": "bánh",
"expected": "found",
"description": "Từ khóa chung, nhiều sản phẩm"
},
{
"tc_id": "SEARCH_08",
"keyword": "kem",
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
"expected": "found",
"description": "Từ khóa phổ biến"
},
{
"tc_id": "SEARCH_09",
"keyword": "abcdefxyz",
"expected": "not_found",
"description": "Từ khóa không tồn tại"
},
{
"tc_id": "SEARCH_10",
"keyword": "123456",
"expected": "not_found",
"description": "Từ khóa chỉ gồm số"
},
{
"tc_id": "SEARCH_11",
"keyword": "@#$%^&*",
"expected": "not_found",
"description": "Từ khóa ký tự đặc biệt"
},
{
"tc_id": "SEARCH_12",
"keyword": "bắp@123",
"expected": "not_found",
"description": "Từ khóa pha chữ + ký tự đặc biệt"
},
{
"tc_id": "SEARCH_13",
"keyword": "    ",
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
"expected": "found",
"description": "Từ khóa chỉ gồm khoảng trắng → hiển thị toàn bộ sản phẩm"
},
{
"tc_id": "SEARCH_14",
"keyword": "",
"expected": "found",
"description": "Không nhập từ khóa → hiển thị toàn bộ sản phẩm"
},
{
"tc_id": "SEARCH_15",
"keyword": "bắp bắp bắp bắp bắp",
"expected": "not_found",
"description": "Từ khóa dài, lặp lại, không khớp dữ liệu"
}

]
```

* Hàm đọc dữ liệu kiểm thử

```
import os

import csv

import json

from pathlib import Path

from typing import Any, List, Dict, Optional, Tuple, Union

from openpyxl import load_workbook

import pandas as pd

# =====

# HÀM CHÍNH: load_data() - Đọc đa định dạng cho Data-Driven Test
# =====

def load_data(filename: str) -> List[Tuple]:
    """
    """
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

Loader đa năng: đọc dữ liệu từ .md, .csv, .json, .xlsx

Trả về list[tuple] để dùng cho pytest.mark.parametrize hoặc test DDT.

Args:

filename (str): Tên file (nằm trong thư mục ../data/)

Returns:

List[Tuple]: Dữ liệu đọc được

"""

data: List[Tuple] = []

base_dir = Path(__file__).resolve().parent.parent / "data"

file_path = base_dir / filename

if not file_path.exists():

 raise FileNotFoundError(f" Không tìm thấy file: {file_path}")

Markdown (.md)

if filename.endswith(".md"):

 with open(file_path, "r", encoding= "utf-8") as f:

 lines = f.readlines()

 for line in lines[2:]: # bỏ 2 dòng đầu (header + separator)

 row = [x.strip() for x in line.strip().split("|") if x.strip()]

 if row:

 data.append(tuple(row))

CSV (.csv)

elif filename.endswith(".csv"):

 with open(file_path, newline= "", encoding= "utf-8") as f:

 reader = csv.reader(f)

 next(reader, None) # bỏ header

 for row in reader:

 if any(row):

 data.append(tuple(row))

JSON (.json)

elif filename.endswith(".json"):

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
with open(file_path, "r", encoding= "utf-8") as f:
    json_data = json.load(f)
    if isinstance(json_data, list):
        for item in json_data:
            data.append(tuple(item.values()))
    elif isinstance(json_data, dict):
        data.append(tuple(json_data.values()))
    # Excel (.xlsx)
    elif filename.endswith(".xlsx"):
        wb = load_workbook(file_path)
        sheet = wb.active
        if sheet is None:
            raise ValueError(" Workbook không chứa sheet nào (sheet = None).")
        if sheet.max_row < 2:
            raise ValueError(" File Excel không có dữ liệu (ít hơn 2 dòng).")
        for row in sheet.iter_rows(min_row = 2, values_only = True):
            if any(row):
                data.append(tuple(row))
        else:
            raise ValueError(" Định dạng file không được hỗ trợ (.md/.csv/.json/.xlsx).")
    # Kiểm tra dữ liệu rỗng
    if not data:
        print(f" Không có dữ liệu hợp lệ trong file {filename}")
    else:
        print(f" Đã load {len(data)} dòng dữ liệu từ {filename}")
return data # luôn return, giúp Pylance hết cảnh báo
# =====
# HÀM PHỤ: load_excel_data() - Dành cho Keyword-Driven Framework
# =====
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
def load_excel_data(filepath: str, sheet_name: Optional[str] = None) ->
pd.DataFrame:

    """
    Đọc dữ liệu từ Excel (.xlsx) cho Framework Keyword-Driven.

    Trả về DataFrame chứa cột như TS_ID, Action Keyword, Locator, ...

    """

    file_path_obj = Path(filepath)

    if not file_path_obj.is_file():

        raise FileNotFoundError(f" Không tìm thấy file Excel: {filepath}")

    try:

        df = pd.read_excel(filepath, sheet_name = sheet_name or 0, header = 0)

        if df.empty:

            raise ValueError(" File Excel rỗng.")

        print(f" Đã load {len(df)} dòng và {len(df.columns)} cột từ
{filepath}")

        return df

    except Exception as e:

        raise Exception(f" Lỗi đọc Excel: {e}")

# =====

# HÀM PHỤ: load_json_config() – Đọc file cấu hình

# =====

def load_json_config(filepath: str) -> Dict[str, Any]:

    """
    Đọc file JSON cấu hình và trả về dict.

    """

    file_path = Path(filepath)

    if not file_path.is_file():

        raise FileNotFoundError(f" Không tìm thấy file cấu hình: {filepath}")

    with open(file_path, encoding= "utf-8") as f:

        data = json.load(f)
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
print(f" Đã load config từ {filepath}")

return data

# =====

# DEMO (chạy thử nếu file được chạy trực tiếp)
# =====

if __name__ == "__main__":
    print("\n Demo Data-Driven (.csv/.json/.xlsx/.md)")

    demo_data = load_data("login_data.xlsx")
    print(demo_data[:2])

    print("\n Demo Keyword-Driven (.xlsx)")
    df = load_excel_data("data/login_keywords.xlsx")
    print(df.head())
```

Nhờ sử dụng DDT, framework có thể chạy cùng một kịch bản kiểm thử với nhiều bộ dữ liệu khác nhau mà không cần sửa mã nguồn, giúp tiết kiệm thời gian và đảm bảo tính bao phủ kiểm thử cao.

4.3.3 Xây dựng kịch bản kiểm thử

Sau khi hoàn thành việc thiết kế và xây dựng các test case cho từng chức năng, kịch bản kiểm thử tự động được triển khai trực tiếp trong framework bằng cách xây dựng các test script trong thư mục tests/. Mỗi kịch bản kiểm thử được hiện thực hóa dưới dạng các hàm kiểm thử (test function) theo chuẩn của Pytest, kết hợp với Playwright để mô phỏng hành vi người dùng trên trình duyệt.

Các kịch bản kiểm thử được tổ chức rõ ràng theo từng chức năng chính của website Savourebakery.com, bao gồm:

- test_login.py:

```
import pytest

from pages.login_page import LoginPage
from utils.data_loader import load_data

@pytest.mark.parametrize("email,password,expected,type",
load_data("login_data.csv"))

def test_login_data_driven(page, base_url, email, password, expected, type):
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
login_page = LoginPage(page, base_url)
login_page.open()
login_page.login(email, password)

if type == "validation":
    assert login_page.is_email_invalid() or
login_page.is_password_invalid()

elif type == "backend":
    actual = login_page.get_error_text()

    assert expected in actual
```

- test_search.py:

```
import json
import pytest
from pages.search_page import SearchPage
def load_search_data():

    with open("data/search_data.json", encoding= "utf-8") as f:
        return json.load(f)

search_data = load_search_data()

@pytest.mark.parametrize("data", search_data)
def test_search_ddt(page, base_url, data):

    search_page = SearchPage(page, base_url)

    tc_id = data["tc_id"]

    keyword = data["keyword"]

    expected = data["expected"]

    # Step 1: Open homepage
    search_page.open_home()

    # Step 2: Open search popup
    search_page.open_search_popup()

    # Step 3: Perform search
    search_page.search(keyword)

    page.wait_for_timeout(2000)
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
# Step 4: Assertion theo behavior thực tế

if expected == "found":
    assert search_page.has_search_result(), (
        f"{tc_id} FAILED - Expected products but found none"
    )
else:
    assert not search_page.has_search_result(), (
        f"{tc_id} FAILED - Expected no products but found some"
    )
```

- test_product_detail.py:

```
import pytest

from pages.product_detail_page import ProductDetailPage

PRODUCT_URL = "https://savourebakery.com/san-pham/banh-bong-lan-cuon-tra-ba-tuoc-224.html"

def test_product_detail_display(page):
    detail = ProductDetailPage(page, PRODUCT_URL)
    detail.open()

    # === Content Assertions ===
    assert detail.get_name(), "Tên sản phẩm không được để trống"
    assert detail.get_code(), "Không thấy mã sản phẩm"
    assert detail.get_bought_count(), "Không thấy thông tin đã mua"
    assert detail.has_image(), "Ảnh sản phẩm không hiển thị"

    price_text = detail.get_price_text()

    assert price_text != "" and any(c.isdigit() for c in price_text), "Giá sản phẩm không hợp lệ"

    # === Quantity default ===
    qty = detail.get_quantity()

    assert qty == "1", f"Quantity mặc định phải là 1, nhưng là {qty}"

def test_product_detail_quantity_controls(page):
    detail = ProductDetailPage(page, PRODUCT_URL)
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
detail.open()

# Default 1

assert detail.get_quantity() == "1"

# Increase

detail.increase_quantity()

assert detail.get_quantity() == "2"

# Reduce

detail.reduce_quantity()

assert detail.get_quantity() == "1"

# Reduce không < 1

detail.reduce_quantity()

assert detail.get_quantity() == "1"

def test_product_detail_add_to_cart(page):

    detail = ProductDetailPage(page, PRODUCT_URL)

    detail.open()

    # Click "Thêm vào giỏ hàng"

    detail.click_add_to_cart()

    page.wait_for_timeout(2000)

    # Assert vẫn ở trang chi tiết sản phẩm

    assert "san-pham" in page.url

    # Assert nút thêm giỏ vẫn tồn tại (không lỗi)

    assert page.locator(".jsSevenAddToCard").is_visible()
```

- test_cart.py:

```
Import pytest

from pages.cart_page import CartPage

@pytest.mark.usefixtures("cart_with_product")

class TestCart :

    def test_01_open_cart_page(self, page):

        cart = CartPage(page)
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
cart.wait_cart_loaded()

assert cart.cart_table.is_visible()

def test_02_cart_has_product(self, page):
    cart = CartPage(page)
    assert cart.get_item_count() >= 1

def test_03_show_product_name(self, page):
    cart = CartPage(page)
    assert cart.product_name.inner_text().strip() != ""

def test_04_show_product_price(self, page):
    cart = CartPage(page)
    assert "VND" in cart.product_price.inner_text()

def test_05_default_quantity_is_1(self, page):
    cart = CartPage(page)
    assert cart.get_quantity() == 1

def test_06_increase_quantity(self, page):
    cart = CartPage(page)
    cart.increase_quantity()
    assert cart.get_quantity() == 2

def test_07_decrease_quantity(self, page):
    cart = CartPage(page)
    cart.decrease_quantity()
    assert cart.get_quantity() == 1

def test_08_quantity_not_less_than_1(self, page):
    cart = CartPage(page)
    cart.decrease_quantity()
    assert cart.get_quantity() >= 1

def test_09_subtotal_change_when_qty_change(self, page):
    cart = CartPage(page)
    before = cart.get_subtotal_text()
    cart.increase_quantity()
```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
    after = cart.get_subtotal_text()

    assert before != after

def test_10_show_total_price(self, page):
    cart = CartPage(page)

    assert "VND" in cart.get_total_text()

def test_11_total_equal_subtotal_single_item(self, page):
    cart = CartPage(page)

    assert cart.get_total_text() == cart.get_subtotal_text()

def test_12_remove_product_from_cart(self, page):
    cart = CartPage(page)

    cart.remove_item()

    assert cart.get_total_text() == "0 VND"

● test_profile_update.py:
import pytest

from pages.profile_page import ProfilePage
from utils.data_loader import load_data
@pytest.mark.parametrize(
    "tc_id, first_name, last_name, phone, address, expect_valid",
    load_data("profile_update_data.xlsx")
)

def test_profile_update(
    login_for_profile,
    tc_id,
    first_name,
    last_name,
    phone,
    address,
    expect_valid
):
    """

```

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

```
PROFILE UPDATE – DATA DRIVEN TEST (EXCEL)

"""

page = login_for_profile

profile = ProfilePage(page)

# Mở trang cập nhật thông tin

profile.open_update_profile_page()

# Nhập dữ liệu theo Excel

profile.fill_profile_form(

    first_name,
    last_name,
    phone,
    address

)

# Submit

profile.submit_update()

# Assert

if expect_valid == "PASS":

    assert profile.has_validation_error() is False, f"{tc_id} expected PASS"

else:

    assert profile.has_validation_error() is True, f"{tc_id} expected FAIL"
```

Mỗi kịch bản kiểm thử được xây dựng theo hướng Data-Driven Testing (DDT), trong đó dữ liệu kiểm thử được tách riêng và lưu trữ trong các file Excel hoặc JSON. Khi thực thi, Pytest sử dụng cơ chế parametrize để tự động lặp qua nhiều bộ dữ liệu khác nhau cho cùng một kịch bản kiểm thử, giúp tăng độ bao phủ và giảm trùng lặp mã nguồn.

Quá trình thực thi kịch bản kiểm thử được framework tự động hóa hoàn toàn. Khi chạy lệnh kiểm thử, hệ thống sẽ tự động mở trình duyệt bằng Playwright, điều

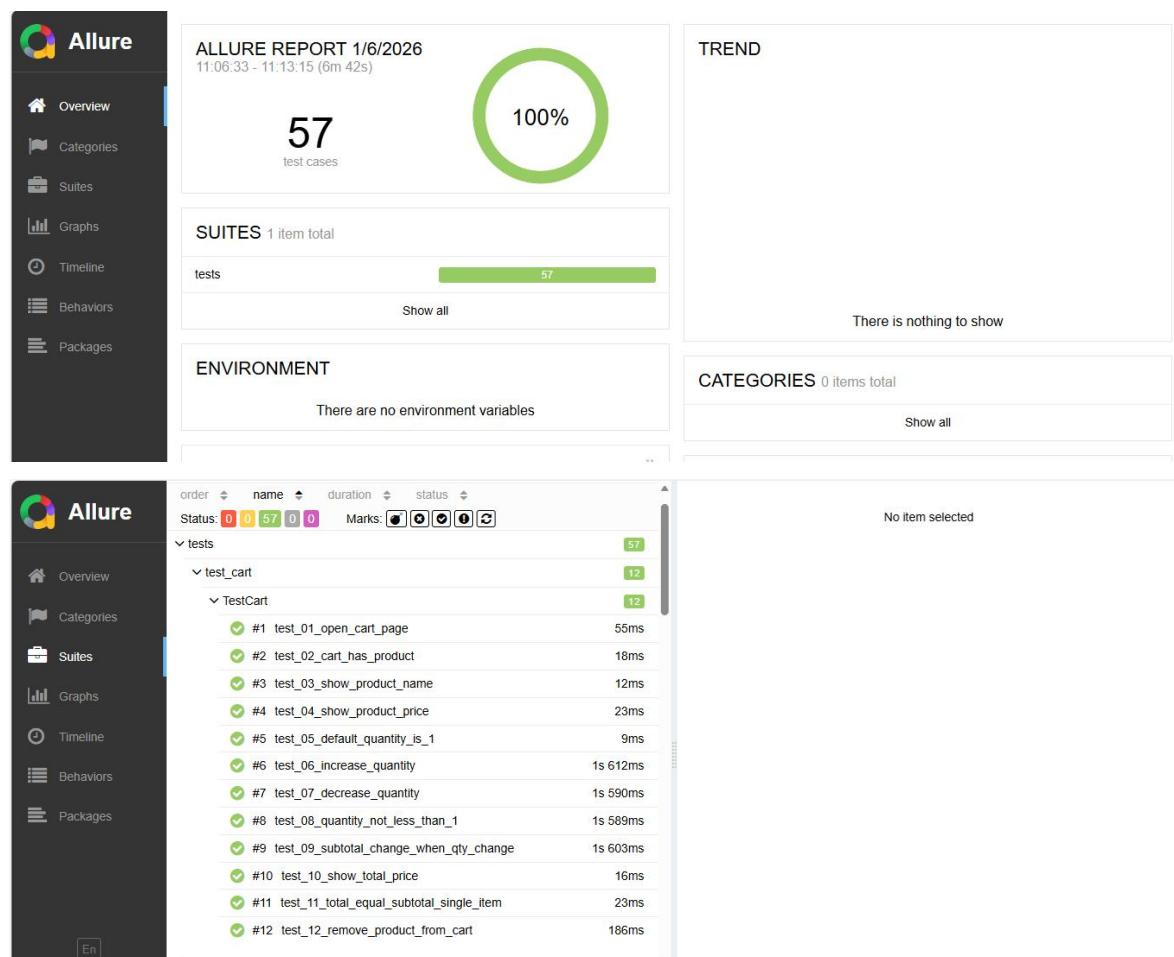
Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

hướng đến website Savourebakery.com, thực hiện các thao tác như nhập liệu, click nút, thay đổi số lượng sản phẩm, và so sánh kết quả thực tế với kết quả mong đợi đã được định nghĩa trong test case.

Trong trường hợp kịch bản kiểm thử thất bại, framework sẽ tự động ghi nhận bằng cách chụp ảnh màn hình và quay video quá trình thực thi. Các dữ liệu này được lưu trữ phục vụ cho việc phân tích và truy vết lỗi. Sau khi quá trình kiểm thử kết thúc, kết quả được tổng hợp và sinh báo cáo trực quan thông qua công cụ Allure, cho phép theo dõi trạng thái Pass/Fail của từng test case, thời gian thực thi, cũng như chi tiết lỗi kèm hình ảnh và video minh họa.

Nhờ việc xây dựng kịch bản kiểm thử theo cấu trúc rõ ràng và tự động hóa toàn bộ quy trình, framework kiểm thử giúp đảm bảo các chức năng chính của website Savourebakery.com được kiểm tra một cách nhất quán, dễ bảo trì và có khả năng mở rộng trong tương lai.

* Báo cáo kiểm thử

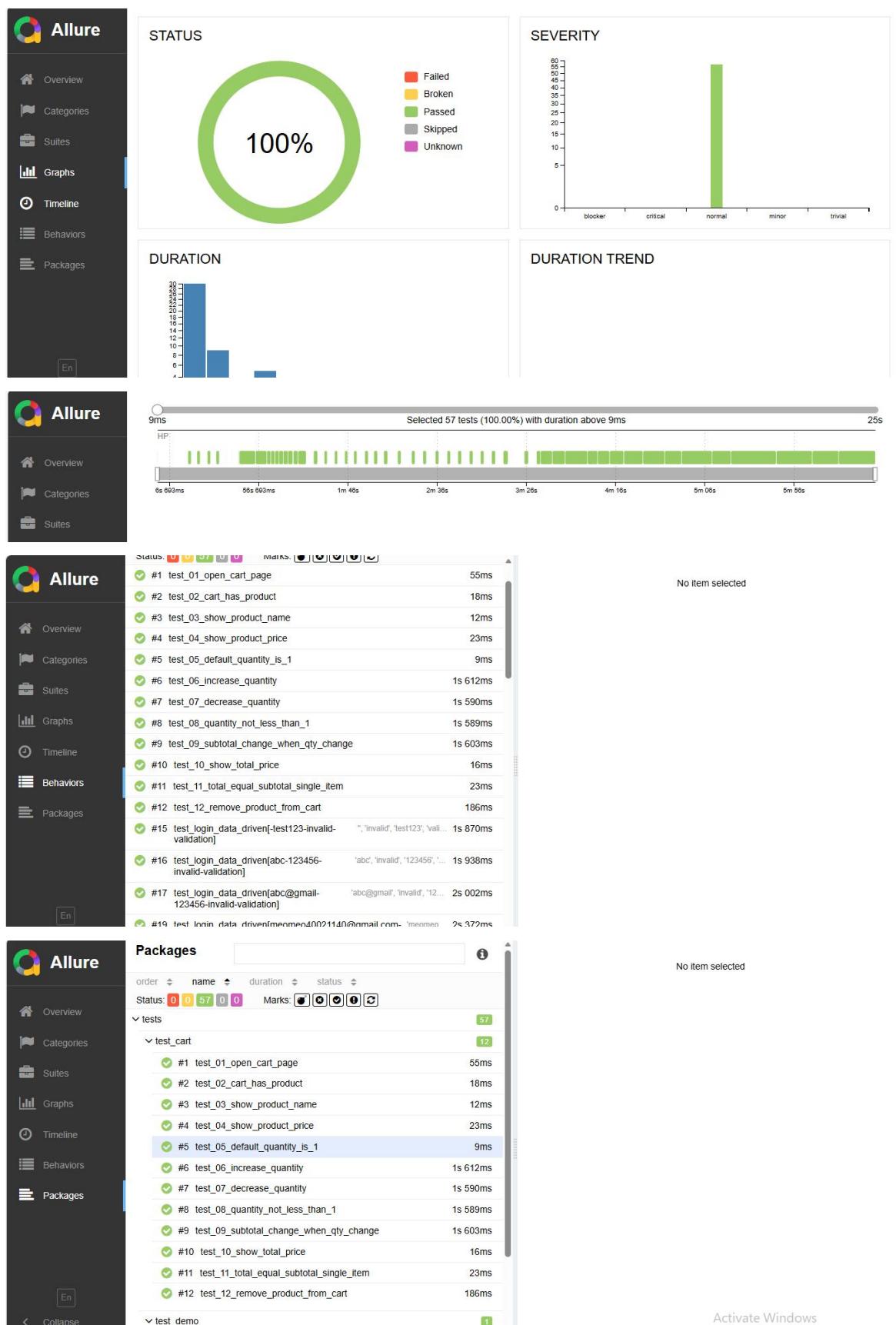


Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

The image displays three separate screenshots of the Allure test report interface, each showing a list of test cases with their status, ID, name, and execution time.

- Screenshot 1 (Top):** Shows results for the `test_login` scenario. It includes five test cases:
 - #1 `test_login_data_driven[-test123-invalid-validation]`: Status green, duration 1s 870ms.
 - #2 `test_login_data_driven[abc-123456-invalid-validation]`: Status green, duration 1s 938ms.
 - #3 `test_login_data_driven[abc@gmail-123456-invalid-validation]`: Status green, duration 2s 002ms.
 - #5 `test_login_data_driven[meomeo40021140@gmail.com-wrongpass-Password khxf4ng lu0111xangl-backend]`: Status green, duration 2s 372ms.
 - #4 `test_login_data_driven[notexist@gmail.com-notexist@gmail-123456-Email khxf4ng lu01ed3n lu1ea1l-backend]`: Status green, duration 2s 180ms.
- Screenshot 2 (Middle):** Shows results for the `test_product_detail` scenario. It includes three test cases:
 - #3 `test_product_detail_add_to_cart`: Status green, duration 4s 301ms.
 - #1 `test_product_detail_display`: Status green, duration 2s 475ms.
 - #2 `test_product_detail_quantity_controls`: Status green, duration 2s 300ms.
- Screenshot 3 (Bottom):** Shows results for the `test_profile` scenario. It includes eight test cases:
 - #1 `test_profile_update[TC01-Meo-Mimi-0987654321-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 2s 003ms.
 - #2 `test_profile_update[TC02-None-Mimi-0987654321-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 425ms.
 - #3 `test_profile_update[TC03-Meo-None-0987654321-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 443ms.
 - #4 `test_profile_update[TC04-Meo-Mimi-None-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 433ms.
 - #5 `test_profile_update[TC05-Meo-Mimi-0987654321-None-PASS]`: Status green, duration 1s 531ms.
 - #6 `test_profile_update[TC06-Meo-Mimi-abc123-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 704ms.
 - #7 `test_profile_update[TC07-Meo-Mimi-098765432-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 806ms.
 - #8 `test_profile_update[TC08-Meo-Mimi-098765432-Trxean trxe1i lu0111u1ea5t-PASS]`: Status green, duration 1s 973ms.
- Screenshot 4 (Bottom-most):** Shows results for the `test_search` scenario. It includes 15 test cases:
 - #1 `test_search_ddt[data0]`: Status green, duration 6s 332ms.
 - #11 `test_search_ddt[data10]`: Status green, duration 10s 347ms.
 - #12 `test_search_ddt[data11]`: Status green, duration 25s 114ms.
 - #13 `test_search_ddt[data12]`: Status green, duration 19s 942ms.
 - #14 `test_search_ddt[data13]`: Status green, duration 14s 542ms.
 - #15 `test_search_ddt[data14]`: Status green, duration 20s 311ms.
 - #2 `test_search_ddt[data1]`: Status green, duration 6s 865ms.
 - #3 `test_search_ddt[data2]`: Status green, duration 12s 243ms.
 - #4 `test_search_ddt[data3]`: Status green, duration 5s 455ms.
 - #5 `test_search_ddt[data4]`: Status green, duration 6s 458ms.
 - #6 `test_search_ddt[data5]`: Status green, duration 7s 647ms.
 - #7 `test_search_ddt[data6]`: Status green, duration 10s 429ms.
 - #8 `test_search_ddt[data7]`: Status green, duration 11s 848ms.
 - #10 `test_search_ddt[data9]`: Status green, duration 10s 420ms.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com



CHƯƠNG 5: KẾT LUẬN

5.1. Kết quả đạt được

Trong quá trình thực hiện đề tài “Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com”, nhóm đã đạt được một số kết quả sau:

Về mặt kiến thức, nhóm đã nắm được các khái niệm và quy trình cơ bản của kiểm thử phần mềm, đặc biệt là kiểm thử tự động cho ứng dụng web. Nhóm hiểu và áp dụng được mô hình kiểm thử tự động, cách thiết kế test case, xây dựng kịch bản kiểm thử và tổ chức thực thi kiểm thử theo hướng Data-Driven Testing. Ngoài ra, nhóm cũng làm quen với các công cụ kiểm thử hiện đại như Playwright, Pytest và Allure.

Về mặt sản phẩm, nhóm đã xây dựng thành công một framework kiểm thử tự động hoàn chỉnh cho website Savourebakery.com. Framework cho phép tự động kiểm thử các chức năng chính của website như đăng nhập, tìm kiếm sản phẩm, xem chi tiết sản phẩm, quản lý giỏ hàng và cập nhật thông tin cá nhân. Hệ thống có khả năng đọc dữ liệu kiểm thử từ file Excel và JSON, tự động thực thi test case, ghi nhận kết quả Pass/Fail, chụp ảnh màn hình khi xảy ra lỗi và sinh báo cáo kiểm thử trực quan thông qua Allure.

Framework được xây dựng theo cấu trúc rõ ràng, dễ mở rộng và có thể tái sử dụng cho các dự án kiểm thử web khác trong tương lai.

5.2. Những hạn chế của đề tài

Bên cạnh những kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định.

Về sản phẩm, framework hiện tại chủ yếu tập trung vào kiểm thử chức năng (Functional Testing) và giao diện người dùng, chưa triển khai kiểm thử hiệu năng (Performance Testing), kiểm thử bảo mật (Security Testing) và kiểm thử khả năng tương thích trên nhiều trình duyệt khác nhau. Một số chức năng nâng cao như quản lý đơn hàng, thanh toán trực tuyến và phân quyền người dùng chưa được đưa vào

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

phạm vi kiểm thử. Ngoài ra, việc log bug vẫn được thực hiện thủ công dựa trên kết quả kiểm thử thay vì tích hợp trực tiếp với các công cụ quản lý lỗi chuyên nghiệp.

Về kỹ năng, nhóm còn hạn chế trong việc phân tích và thiết kế hệ thống kiểm thử ở mức độ nâng cao. Kỹ năng tối ưu hóa kịch bản kiểm thử, xử lý các tình huống bất đồng bộ phức tạp và xây dựng các assertion nâng cao trong Playwright vẫn chưa thực sự hoàn thiện. Kinh nghiệm thực tế trong việc triển khai kiểm thử tự động cho các hệ thống lớn còn chưa nhiều.

5.3. Hướng phát triển của đế tài

Trong thời gian tới, đế tài có thể được phát triển và hoàn thiện hơn theo các hướng sau:

Mở rộng phạm vi kiểm thử sang các loại kiểm thử khác như kiểm thử hiệu năng, kiểm thử bảo mật và kiểm thử tương thích đa trình duyệt nhằm đánh giá toàn diện chất lượng website. Bổ sung thêm các kịch bản kiểm thử cho các chức năng nâng cao như thanh toán, quản lý đơn hàng và xử lý lỗi phía server.

Nâng cấp framework bằng cách tích hợp với các công cụ quản lý bug như Jira hoặc Redmine để tự động ghi nhận lỗi từ kết quả kiểm thử. Đồng thời, cải thiện việc quản lý dữ liệu kiểm thử và tối ưu thời gian thực thi bằng cách chạy song song (parallel execution).

Ngoài ra, framework có thể được đóng gói và chuẩn hóa để áp dụng cho các website thương mại điện tử khác, góp phần nâng cao hiệu quả và độ tin cậy của quá trình kiểm thử phần mềm.

Xây dựng framework kiểm thử tự động cho ứng dụng Web và ứng dụng kiểm thử cho website Savourebakery.com

TÀI LIỆU THAM KHẢO

- [1] K. CNTT, Đề cương Kiểm thử ứng dụng Web & Mobile, Trường Đại học SPKT Hưng Yên.
- [2] K. CNTT, Giáo trình môn Kiểm thử phần mềm, Trường Đại học SPKT Hưng Yên.
- [3] <https://playwright.dev/python/docs/intro/>
- [4] <https://allurereport.org/docs/>
- [1] Tên tác giả (năm XB), Tên sách, NXB, Nơi XB
- [2]