TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**AO THUỴ NGỌC TRÂN – 52100937**

**KIỂM THỬ CÁC CHỨC NĂNG CỦA TRANG WEB SPA ĐẶT ĐỒ ĂN BẰNG CYPRESS**

**BÁO CÁO CUỐI KỲ**

**KIỂM THỬ PHẦN MỀM TỰ ĐỘNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**AO THUỴ NGỌC TRÂN – 52100937**

**KIỂM THỬ CÁC CHỨC NĂNG CỦA TRANG WEB SPA ĐẶT ĐỒ ĂN BẰNG CYPRESS**

**BÁO CÁO CUỐI KỲ**

**KIỂM THỬ PHẦN MỀM TỰ ĐỘNG**

Người hướng dẫn

**GV. Nguyễn Thanh Quân**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn thầy Nguyễn Thanh Quân đã tạo điều kiện tốt nhất cho chúng em hoàn thành bài báo cáo này, dù có đôi lúc chúng em gặp khó khăn dẫn đến gần như làm không kịp bài nhưng thầy vẫn hỗ trợ hết mình để chúng em có thể nộp bài tập đúng hạn, và những feed back chi tiết của thầy để chúng em cải thiện dần những sai sót cho bài tập cuối kỳ

*TP. Hồ Chí Minh, ngày 18 tháng 05 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Ao Thuỵ Ngọc Trân*

**KIỂM THỬ CÁC CHỨC NĂNG CỦA TRANG WEB SPA ĐẶT ĐỒ ĂN BẰNG CYPRESS**

**TÓM TẮT**

Hầu hết các trang web hiện nay đều sử dụng MPA (Multiple-page applications), mỗi khi 1 request đến trang MPA, server sẽ tính toán và trả về toàn bộ các thành phần của 1 trang web dưới dạng mã html. Trong các tác vụ về lướt bảng tin, tìm kiếm, … yêu cầu trang web phải load lại liên tục thì, thay vì chỉ load phần nội dung thì MPA lại load lại toàn bộ từ layout cho tới nội dung, điều này làm cho thời gian trễ dài hơn, hiệu suất bị giảm đi đáng kể.

SPA (Single-page Application) được sinh ra để tối ưu việc chỉ thay đổi nội dung bên trong trang web, nó sẽ giữ lại layout cố định và dùng các đoạn mã javascript để thay đổi các thành phần nội dung bên trong, làm cho server chỉ cần tính toán và trả về phần nội dung yêu cầu trên chính trang web đó, mặc dù còn khá nhiều khuyết điểm nhưng vẫn là một application đáng tin cậy, hiện nay SPA được tích hợp nhiều vào trong các MPA để nâng cao hiệu suất hệ thống, giảm thiểu độ trễ đi rất nhiều so với MPA truyền thống.

Cypress framework là một framework thử nghiệm end-to-end (e2e) dựa trên JavaScript được xây dựng trên Mocha, hoạt động với máy chủ Node js, có hỗ trợ CI/CD giúp ích cho việc kiểm tra và phát hiện lỗi nhanh trong thời gian thực. Trong bài báo cáo này nhóm em lựa chọn kiểm thử e2e để tiến hành kiểm tra 1 web SPA đặt đồ ăn đơn giản bao gồm các chức năng:

- Chọn loại món ăn, thức uống

- Tìm kiếm món ăn, thức uống

- Xem thông tin chi tiết sản phẩm

- Đánh giá sản phẩm

- Tiến hành chọn sản phẩm

- Kiểm tra giỏ hàng

- Cập nhật giỏ hàng

**MỤC LỤC**

[DANH MỤC HÌNH VẼ iv](#_Toc167025842)

[DANH MỤC BẢNG BIỂU iv](#_Toc167025843)

[DANH MỤC CÁC CHỮ VIẾT TẮT iv](#_Toc167025844)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 1](#_Toc167025845)

[1.1 Các công cụ được sử dụng để hoàn thành đề tài 1](#_Toc167025846)

[1.1.1 Tổng quan về e2e testing 1](#_Toc167025847)

[1.1.2 Giới thiệu về webpack 2](#_Toc167025848)

[1.1.3 Giới thiệu về cypress 9](#_Toc167025849)

[1.2 Đặc tả use case các chức năng của trang web 15](#_Toc167025850)

[1.2.1 Usecase tổng quát 15](#_Toc167025851)

[1.2.2 Bảng đặc tả usecase chi tiết của trang web 16](#_Toc167025852)

[1.3 Test plan 27](#_Toc167025853)

[1.4 Các trường hợp kiểm thử 28](#_Toc167025854)

[CHƯƠNG 2. BÁO CÁO KIỂM THỬ 29](#_Toc167025855)

[2.1 Thông tin dự án 29](#_Toc167025856)

[2.2 Các thông số kiểm thử 29](#_Toc167025857)

[2.2.1 Các test case 29](#_Toc167025858)

[2.2.2 Các bước vận hành kiểm thử Cypress 34](#_Toc167025859)

[2.2.3 Các results và các defects 35](#_Toc167025860)

[2.3 Kết quả kiểm thử 38](#_Toc167025861)

[2.4 Các đề xuất cải tiến 40](#_Toc167025862)

[2.5 Tổng kết 40](#_Toc167025863)

[TÀI LIỆU THAM KHẢO 41](#_Toc167025864)

# DANH MỤC HÌNH VẼ

[Hình 1 Usecase tổng quát 16](#_Toc3137)

# DANH MỤC BẢNG BIỂU

[Bảng 1 Bảng đặc tả Usecase Xem thông tin mặt hàng 16](#_Toc4654)

[Bảng 2 Bảng đặc tả Usecase Chọn sản phẩm 17](#_Toc22567)

[Bảng 3 Bảng đặc tả Usecase Chỉnh sửa giỏ hàng 18](#_Toc5741)

[Bảng 4 Bảng đặc tả Usecase Xóa sản phẩm 19](#_Toc17477)

[Bảng 5 Bảng đặc tả Usecase Xem giỏ hàng 20](#_Toc15769)

[Bảng 6 Bảng đặc tả Usecase Xem các sản phẩm mới 21](#_Toc23793)

[Bảng 7 Bảng đặc tả Usecase Xem sản phẩm khuyến mãi 22](#_Toc2514)

[Bảng 8 Bảng đặc tả Usecase Đánh giá sản phẩm 24](#_Toc1903)

[Bảng 9 Bảng đặc tả Usecase Điều chỉnh số lượng sản phẩm muốn chọn 25](#_Toc28051)

[Bảng 10 Bảng đặc tả Usecase Tìm kiếm sản phẩm 26](#_Toc17965)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CMD | Command Prompt |
| MPA | Multiple-page Application |
| SPA | Single-page Application |
| E2e | end-to-end |
| CI | Continuous Integration |
| CD | Continuous Delivery/Deployment |

# GIỚI THIỆU CHUNG

## Các công cụ được sử dụng để hoàn thành đề tài

### Tổng quan về e2e testing

E2e testing (end-to-end testing) là phương pháp thử nghiệm đánh giá toàn bộ luồng ứng dụng, từ đầu đến cuối. Nó đảm bảo rằng tất cả các thành phần hoạt động như mong đợi và ứng dụng phần mềm hoạt động chính xác trong các tình huống thực tế.

Trong e2e testing, phần mềm được kiểm tra từ góc độ của người dùng cuối, mô phỏng kịch bản người dùng thực, bao gồm giao diện người dùng, dịch vụ phụ trợ, cơ sở dữ liệu và giao tiếp mạng. Mục đích của e2e testing là xác thực hành vi tổng thể của ứng dụng, bao gồm chức năng, độ tin cậy, hiệu suất và bảo mật.

E2e testing nhằm mục đích xác định bất kỳ lỗi hoặc vấn đề nào có thể phát sinh khi các phần khác nhau của ứng dụng tương tác với nhau. Bằng cách thử nghiệm toàn bộ hệ thống, e2e testing đảm bảo rằng tất cả các thành phần hoạt động liền mạch với nhau để đáp ứng mong đợi của người dùng.

E2e testing thường được thực hiện sau integration testing (tích hợp), thử nghiệm các mô-đun riêng lẻ và trước thử nghiệm user aceptance testing (tiếp nhận), nhằm đảm bảo rằng ứng dụng đáp ứng các yêu cầu của người dùng.

Các bước thực hiện e2e testing:

* Phân tích các yêu cầu. Có ý tưởng rõ ràng về cách ứng dụng hoạt động ở mọi khía cạnh.
* Thiết lập môi trường thử nghiệm phù hợp với tất cả các yêu cầu
* Phân tích yêu cầu phần mềm và phần cứng
* Liệt kê cách mọi hệ thống cần phản hồi
* Liệt kê các phương pháp kiểm tra cần thiết để kiểm tra những phản hồi này. Bao gồm các mô tả rõ ràng về các tiêu chuẩn (ngôn ngữ, công cụ, v.v.) cần tuân thủ trong mỗi bài kiểm tra.
* Thiết kế các trường hợp thử nghiệm
* Chạy thử nghiệm, nghiên cứu và lưu kết quả

### Giới thiệu về webpack

#### Webpack là gì?

Khái niệm:

Webpack là một module bundler dành cho JavaScript, chủ yếu được sử dụng để gói gọn các module JavaScript với tất cả các dependencies của chúng thành một hoặc nhiều file bundle, giúp cho quá trình tải trang nhanh hơn và hiệu quả hơn. Nó được phát triển và duy trì bởi Tobias Koppers và cộng đồng mã nguồn mở.

Mục đích của Webpack:

* Đóng gói (Bundling): Tự động gói gọn các module JavaScript và các tài nguyên khác (CSS, hình ảnh, fonts, v.v.) thành các file riêng lẻ.
* Quản lý dependencies: Webpack tự động xử lý và giải quyết các dependencies giữa các module, giúp tránh các vấn đề như vòng lặp phụ thuộc.
* Tối ưu hóa tài nguyên: Nén và tối ưu hóa các file JavaScript, CSS, hình ảnh để giảm kích thước file và tăng tốc độ tải trang.
* Hot Module Replacement (HMR): Hỗ trợ cập nhật module trực tiếp mà không cần tải lại toàn bộ trang web, rất hữu ích cho quá trình phát triển.
* Hỗ trợ các module hiện đại: Tích hợp dễ dàng với các module ES6, ES7 và các module khác thông qua Babel hoặc các trình biên dịch khác.

Lợi ích của Webpack:

1. Tối ưu hóa tài nguyên:

* Minification: Giảm kích thước file bằng cách loại bỏ khoảng trắng, chú thích và sử dụng các tên biến ngắn hơn.
* Tree Shaking: Loại bỏ mã không sử dụng khỏi bundle cuối cùng, giúp tối ưu hóa hiệu suất và giảm kích thước file.
* Code Splitting: Chia nhỏ ứng dụng thành nhiều file bundle nhỏ hơn, cho phép tải các phần của ứng dụng khi cần thiết (lazy loading).

1. Quản lý dependencies:

* Static Analysis: Phân tích và xây dựng biểu đồ phụ thuộc (dependency graph) từ các module được sử dụng trong dự án.
* Module Resolution: Tự động tìm kiếm và gói gọn các module từ các nguồn khác nhau, bao gồm node\_modules và các đường dẫn tương đối.

1. Hỗ trợ các module hiện đại:

* ES6/ES7 Modules: Hỗ trợ native module syntax của JavaScript, giúp phát triển với các tính năng hiện đại nhất.
* Loaders và Plugins: Hỗ trợ các định dạng file khác nhau và mở rộng khả năng của Webpack thông qua hệ thống loaders và plugins phong phú.

1. Tích hợp dễ dàng:

* Frameworks và Libraries: Dễ dàng tích hợp với các framework và thư viện phổ biến như React, Angular, Vue.js.
* Configuration Flexibility: Cấu hình Webpack linh hoạt, có thể tùy chỉnh để phù hợp với các yêu cầu cụ thể của dự án.

1. Development Enhancements:

* Hot Module Replacement (HMR): Cho phép thay đổi module trực tiếp mà không cần tải lại trang, giúp cải thiện trải nghiệm phát triển.
* Source Maps: Hỗ trợ source maps giúp debug dễ dàng hơn bằng cách ánh xạ mã nguồn đã biên dịch về mã nguồn gốc.

#### Cách hoạt động của webpack

*Bước 1: Xác định Entry Point*

Trong tệp cấu hình của Webpack, bạn xác định entry point của ứng dụng. Đây là điểm bắt đầu mà từ đó Webpack sẽ bắt đầu phân tích và xây dựng sơ đồ phụ thuộc (dependency graph) của ứng dụng.

*Bước 2: Xây dựng Dependency Graph*

Webpack phân tích mã nguồn từ entry point và xác định tất cả các module và sự phụ thuộc giữa chúng. Mỗi module có thể là một tệp JavaScript, CSS, hình ảnh hoặc bất kỳ tệp nào khác mà ứng dụng của bạn cần. Dependency graph là biểu đồ thể hiện các module và các liên kết phụ thuộc giữa chúng.

*Bước 3: Xử lý các Module*

Webpack bắt đầu tải và xử lý các module. Đối với mỗi module, Webpack sẽ sử dụng các loader tương ứng để chuyển đổi module đó thành mã JavaScript có thể sử dụng được trong trình duyệt.

*Bước 4: Áp dụng Plugins*

Sau khi các module được xử lý, Webpack áp dụng các plugin mà bạn đã cấu hình. Plugins thực hiện các tác vụ bổ sung như tối ưu hóa mã, tạo file HTML từ template, và nhiều công việc khác để tùy chỉnh quá trình build.

*Bước 5: Tạo các Bundle*

Dựa trên dependency graph và kết quả của quá trình xử lý module, Webpack tạo ra các bundle. Mỗi bundle chứa một nhóm các module có liên quan.

*Bước 6: Tối ưu hóa*

Webpack cung cấp các công cụ tối ưu hóa để giảm kích thước file và tăng tốc độ tải trang. Nó có thể gộp các file lại, nén mã nguồn, tối ưu hóa hình ảnh và áp dụng nhiều biện pháp tối ưu hóa khác. Sau khi tối ưu hóa xong, các bundle được tạo ra sẽ sẵn sàng để triển khai ứng dụng.

#### Các loại File và Module của webpack

1. Các loại module của webpack

*JavaScript Modules*

* Khái niệm: JavaScript modules là các khối mã độc lập mà bạn có thể xuất (export) và nhập (import) để sử dụng ở các phần khác nhau trong ứng dụng. Có nhiều định dạng module khác nhau như ES6 modules, CommonJS, và AMD.
* Cách dùng: Không cần cấu hình đặc biệt trong Webpack, chỉ cần sử dụng cú pháp import và export phù hợp với định dạng module của bạn.

*CSS Modules*

* Khái niệm: CSS modules giúp bạn quản lý và đóng gói các file CSS. CSS modules đảm bảo rằng các lớp CSS chỉ áp dụng trong phạm vi module của chúng, ngăn xung đột tên giữa các module khác nhau.
* Cách dùng: Webpack có thể xử lý CSS modules thông qua các loader mà bạn cấu hình, giúp bạn dễ dàng bao gồm CSS trong bundle cuối cùng.

*Image Modules*

* Khái niệm: Image modules cho phép bạn quản lý và đóng gói các file hình ảnh như PNG, JPEG, GIF, SVG. Điều này giúp bạn tối ưu hóa và bao gồm các hình ảnh trong ứng dụng của mình.
* Cách dùng: Webpack sử dụng các loader để xử lý các file hình ảnh, chuyển đổi chúng thành các URL hoặc Base64, và bao gồm chúng trong bundle.

*Font Modules*

* Khái niệm: Font modules giúp bạn quản lý và đóng gói các file font như WOFF, WOFF2, EOT, TTF, và OTF. Điều này đảm bảo rằng các phông chữ được bao gồm và sử dụng đúng cách trong ứng dụng của bạn.
* Cách dùng: Webpack sử dụng các loader để xử lý và bao gồm các file font trong bundle.

*HTML Modules*

* Khái niệm: HTML modules cho phép bạn quản lý và đóng gói các file HTML. Thường sử dụng các plugin để tự động tạo và chèn các bundle script vào trong file HTML.
* Cách dùng: Webpack có các plugin hỗ trợ để tạo các file HTML mới hoặc sửa đổi các file HTML hiện có.

*JSON Modules*

* Khái niệm: JSON modules giúp bạn quản lý và đóng gói các file JSON, cho phép bạn dễ dàng nhập dữ liệu JSON vào mã JavaScript của mình.
* Cách dùng: Không cần cấu hình đặc biệt, Webpack có thể xử lý JSON modules mặc định.

1. Quản Lý Tài Nguyên (Assets Management)

Khái niệm: Quản lý tài nguyên (assets) trong Webpack bao gồm việc quản lý và tối ưu hóa các file tĩnh như hình ảnh, phông chữ, và các file khác. Điều này giúp giảm kích thước bundle và cải thiện hiệu suất tải trang.

*File Loader*

* Khái niệm: File loader là công cụ cho phép bạn chuyển đổi các file (hình ảnh, phông chữ) thành các URL và bao gồm chúng trong bundle.
* Cách dùng: File loader xử lý các file tĩnh bằng cách di chuyển chúng vào thư mục output và trả về URL tương ứng để sử dụng trong mã nguồn.

*URL Loader*

* Khái niệm: URL loader là một công cụ chuyển đổi các file thành Base64 URL nếu file nhỏ hơn một kích thước định trước. Điều này giúp giảm số lượng yêu cầu HTTP bằng cách nhúng các file nhỏ trực tiếp vào bundle JavaScript.
* Cách dùng: URL loader xử lý các file tĩnh nhỏ và trả về Base64 URL hoặc URL thông thường nếu file lớn hơn giới hạn định trước.

*Image Minimizer Plugin*

* Khái niệm: Image Minimizer Plugin giúp tối ưu hóa kích thước file hình ảnh, giảm dung lượng mà không làm giảm chất lượng đáng kể.
* Cách dùng: Plugin này áp dụng các kỹ thuật nén và tối ưu hóa khác nhau để giảm kích thước file hình ảnh, giúp cải thiện hiệu suất tải trang.

1. Cách Webpack Tìm Kiếm và Xử Lý Các Module (Module Resolution)

Khái niệm: Module resolution là quá trình Webpack tìm kiếm và xác định vị trí các module để bao gồm trong bundle. Quá trình này bao gồm việc tìm kiếm trong các thư mục, áp dụng các alias, và xử lý các phần mở rộng file.

*Path Aliases*

* Khái niệm: Path aliases cho phép bạn định nghĩa các tên thay thế (alias) cho các đường dẫn dài và phức tạp. Điều này giúp mã nguồn trở nên ngắn gọn và dễ hiểu hơn.
* Cách dùng: Bạn có thể cấu hình các alias để dễ dàng truy cập các thư mục và file thường dùng mà không cần viết đường dẫn đầy đủ.

*Extensions*

* Khái niệm: Extensions là các phần mở rộng file mà Webpack sẽ tự động thêm vào khi tìm kiếm các module. Điều này giúp đơn giản hóa quá trình nhập các module.
* Cách dùng: Bạn có thể cấu hình các phần mở rộng mà Webpack sẽ sử dụng để tự động tìm kiếm và nhập các module phù hợp.

*Modules*

* Khái niệm: Modules trong Webpack bao gồm các thư mục mà Webpack sẽ tìm kiếm khi resolve các module. Điều này giúp xác định vị trí các module trong cấu trúc dự án.
* Cách dùng: Bạn có thể cấu hình các thư mục mà Webpack sẽ sử dụng để tìm kiếm các module, giúp quản lý và tổ chức mã nguồn hiệu quả hơn.

#### Tích hợp với các công cụ và framework

*Babel: Tích hợp Webpack với Babel để Chuyển đổi (Transpile) ES6/ES7 sang ES5*

* Khái niệm:

Babel là một công cụ biên dịch JavaScript được sử dụng để chuyển đổi mã nguồn từ ES6/ES7 sang ES5, giúp mã JavaScript có thể chạy trên các trình duyệt không hỗ trợ các tính năng mới nhất của JavaScript.

* Lợi ích:

Tương thích trình duyệt: Đảm bảo mã JavaScript có thể chạy trên các trình duyệt cũ không hỗ trợ các tính năng mới của ES6/ES7.

Sử dụng tính năng mới: Cho phép sử dụng các tính năng mới nhất của JavaScript mà không phải lo lắng về khả năng tương thích với trình duyệt.

* Cách dùng:

Cài đặt Babel và các loader: Cài đặt các gói Babel như @babel/core, babel-loader, và các preset như @babel/preset-env.

Cấu hình Webpack: Sử dụng babel-loader để Webpack xử lý các file JavaScript qua Babel trước khi đóng gói.

*React*

* Khái niệm:

React là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng, đặc biệt là các ứng dụng một trang (SPA).

* Lợi ích:

Component-based: React cho phép xây dựng giao diện người dùng bằng cách chia nhỏ ứng dụng thành các component có thể tái sử dụng.

Hiệu suất cao: Virtual DOM của React giúp tối ưu hóa việc cập nhật giao diện, cải thiện hiệu suất.

* Cách dùng:

Cài đặt các gói cần thiết: Bao gồm react, react-dom, babel-loader, và @babel/preset-react.

Cấu hình Webpack và Babel: Sử dụng babel-loader với preset @babel/preset-react để biên dịch mã JSX và ES6/ES7.

*Vue.js*

* Khái niệm:

Vue.js là một framework JavaScript tiến bộ được sử dụng để xây dựng giao diện người dùng. Nó dễ dàng tích hợp với các dự án hiện có và có thể mở rộng để xây dựng các ứng dụng phức tạp.

* Lợi ích:

Dễ học: Vue.js có một cú pháp đơn giản và dễ học, giúp các nhà phát triển nhanh chóng bắt đầu.

Tích hợp tốt: Có thể dễ dàng tích hợp Vue.js vào các dự án hiện có mà không cần thay đổi cấu trúc mã nguồn.

* Cách dùng:

Cài đặt các gói cần thiết: Bao gồm vue, vue-loader, vue-template-compiler, và các loader khác như css-loader và vue-style-loader.

Cấu hình Webpack: Sử dụng vue-loader để xử lý các file .vue và tích hợp các công cụ liên quan.

*Angular*

* Khái niệm:

Angular là một framework phát triển ứng dụng web toàn diện được phát triển bởi Google. Angular sử dụng TypeScript và cung cấp một hệ thống module mạnh mẽ để xây dựng các ứng dụng lớn và phức tạp.

* Lợi ích:

Mạnh mẽ và toàn diện: Angular cung cấp đầy đủ các công cụ và chức năng để xây dựng các ứng dụng web phức tạp, từ quản lý state đến routing và xử lý HTTP.

TypeScript: Sử dụng TypeScript giúp viết mã rõ ràng và dễ bảo trì, với các tính năng của ngôn ngữ mạnh mẽ như typing và các tính năng mới nhất của JavaScript.

* Cách dùng:

Cài đặt các gói cần thiết: Bao gồm @angular/core, @angular/cli, typescript, ts-loader, và các gói liên quan khác.

Cấu hình Webpack: Sử dụng ts-loader để biên dịch TypeScript và cấu hình Webpack để xử lý các file Angular.

### Giới thiệu về cypress

Cypress là một framework e2e testing dựa trên JavaScript được xây dựng trên Mocha

Cypress có hỗ trợ tốt 2 tính năng Automatic waiting và Real-Time Reloads

- Automatic waiting: Cypress tự động đợi các phần tử hiển thị, hoạt ảnh hoàn tất, tải DOM, các lệnh gọi XHR và AJAX kết thúc

- Real-Time Reloads: Cypress tự động chạy file test sau khi lưu lại các thay đổi với file đó

Thời gian để viết test bằng cypress tương đối ngắn, dù ban đầu hơi tốn một chút thời gian để làm quen nhưng sau khi đã setup hoàn tất thì công đoạn test hoàn thành nhanh hơn

#### Lợi ích của Cypress

* Tốc độ: Cypress được biết đến với tốc độ thực thi kiểm thử nhanh chóng. Nhờ vào kiến trúc độc đáo của nó, Cypress có thể chạy các bài kiểm thử một cách hiệu quả và nhanh chóng hơn so với các công cụ kiểm thử khác.
* Dễ sử dụng: Cypress được thiết kế để dễ sử dụng với một giao diện người dùng thân thiện và trực quan. Người dùng có thể viết các bài kiểm thử một cách dễ dàng bằng cách sử dụng các API rõ ràng và dễ hiểu.
* Khả năng debug mạnh mẽ: Cypress cung cấp các công cụ debug mạnh mẽ, cho phép người dùng dễ dàng theo dõi, ghi lại, và phân tích các bước trong quá trình kiểm thử. Điều này giúp phát hiện và sửa lỗi một cách nhanh chóng và dễ dàng.
* Tích hợp tốt với các công cụ khác: Cypress có khả năng tích hợp tốt với các công cụ CI/CD như Jenkins, CircleCI, và GitHub Actions, giúp tự động hóa quy trình kiểm thử và tích hợp vào quy trình phát triển liên tục của dự án.
* Cộng đồng và tài liệu phong phú: Cypress có một cộng đồng lớn và nhiều tài liệu hữu ích, bao gồm hướng dẫn, video tutorials, và các khóa học trực tuyến, giúp người dùng dễ dàng học và sử dụng công cụ này.

#### Cài đặt và cấu hình trong Cypress

a) Cài Đặt Cypress

Cypress có thể được cài đặt dễ dàng thông qua npm hoặc yarn, hai công cụ quản lý gói phổ biến trong cộng đồng phát triển phần mềm:

- Cài Đặt qua npm: Sử dụng npm để cài đặt Cypress bằng cách chạy lệnh npm install cypress --save-dev trong terminal. Quá trình này sẽ cài đặt Cypress và thêm nó vào danh sách các gói phụ thuộc của dự án.

- Cài Đặt qua yarn: Nếu bạn ưa thích yarn, bạn cũng có thể cài đặt Cypress bằng lệnh yarn add cypress --dev.

Trước khi cài đặt Cypress, đảm bảo rằng máy tính đã cài đặt Node.js và npm hoặc yarn.

b) Cấu Hình Cypress

Sau khi cài đặt, Cypress sẽ tạo một thư mục cấu hình mặc định và một tệp cypress.json trong thư mục gốc của dự án.

Cấu Hình cơ bản: Tệp cypress.json chứa các cài đặt cơ bản của Cypress, như URL mặc định cho các bài kiểm thử. Có thể chỉnh sửa tệp này để cấu hình Cypress theo nhu cầu.

Tùy chọn cấu hình khác: Ngoài cypress.json, cũng có thể sử dụng các tệp cấu hình khác như cypress.env.json để lưu trữ các biến môi trường hoặc plugins/index.js để cấu hình các plugins tùy chỉnh.

c) Sử Dụng Cypress

Khởi Động Cypress: Có thể bắt đầu Cypress bằng cách chạy lệnh npx cypress open trong terminal. Điều này sẽ mở Cypress Test Runner và cho phép chạy các bài kiểm thử, quản lý dự án, và xem kết quả kiểm thử.

Viết Bài Kiểm Thử: Cypress Test Runner cho phép viết và chạy các bài kiểm thử một cách dễ dàng. Có thể tạo các tệp kiểm thử mới trong thư mục cypress/integration và sử dụng các API của Cypress để viết các ca kiểm thử.

d) Mở Rộng Cấu Hình

Tùy Chỉnh Cài Đặt: Tệp cypress.json cho phép tùy chỉnh cài đặt của Cypress, bao gồm cả việc đặt đường dẫn cho các thư mục lưu trữ kiểm thử và cấu hình browser mặc định.

Sử Dụng Plugins và Thư Viện Bên Ngoài: Cypress hỗ trợ sử dụng các plugins và thư viện bên ngoài để mở rộng tính năng của nó. Có thể thêm các plugins bằng cách cài đặt chúng thông qua npm và thêm cấu hình tương ứng vào tệp cypress.json.

#### Cấu trúc thư mục và tệp tin trong Cypress

1. Cấu Trúc Thư Mục Mặc Định

Cypress tổ chức thư mục của dự án kiểm thử một cách có tổ chức, giúp dễ dàng quản lý và hiểu cách tổ chức của các tệp và thư mục.

cypress/: Đây là thư mục chính của dự án Cypress, nơi bạn sẽ thấy tất cả các tệp liên quan đến việc viết và chạy các bài kiểm thử.

cypress/integration/ (hoặc cypress/e2e/, cypress/component/, …): Thư mục này chứa tất cả các tệp kiểm thử (test files). Mỗi tệp trong thư mục này thường tương ứng với một nhóm kiểm thử hoặc một tính năng cụ thể trong ứng dụng.

cypress/support/: Thư mục này chứa các tệp hỗ trợ, bao gồm các tệp JavaScript và các plugin mở rộng cho Cypress. Một tệp quan trọng trong thư mục này là index.js, nơi có thể định nghĩa các hàm và lớp tùy chỉnh để tái sử dụng trong các bài kiểm thử.

cypress/plugins/: Nếu sử dụng các plugins mở rộng cho Cypress, các tệp cấu hình cho những plugins đó sẽ được đặt trong thư mục này.

1. Tệp Tin Quan Trọng

Các tệp tin quan trọng trong dự án Cypress giúp quản lý và chạy các bài kiểm thử một cách hiệu quả.

cypress.config.json: Đây là tệp cấu hình cho Cypress. Có thể sử dụng tệp này để đặt các cài đặt cấp độ dự án như đường dẫn mặc định cho các tệp kiểm thử, cài đặt URL mặc định và nhiều hơn nữa.

cypress/support/index.js (hoặc cypress/support/e2e.js đối với e2e): Tệp này chứa các hàm và cấu hình tùy chỉnh mà bạn muốn sử dụng trong các bài kiểm thử. Có thể định nghĩa các hàm tiện ích, cài đặt trước cho các bài kiểm thử, và nhiều hơn nữa trong tệp này.

Tệp Kiểm Thử (.cy.js): Các tệp kiểm thử là những tệp chứa các ca kiểm thử cụ thể cho ứng dụng của bạn. Mỗi tệp .cy.js thường chứa một hoặc nhiều ca kiểm thử và được đặt trong thư mục cypress/integration/.

c) Tùy Chỉnh Thư Mục và Tệp Tin

Ngoài cấu trúc thư mục và tệp tin mặc định, Cypress cũng cho phép tùy chỉnh cấu trúc của dự án và sử dụng các tệp mở rộng theo nhu cầu.

Tùy chỉnh cấu trúc thư mục: Có thể tự do tổ chức các thư mục và tệp tin trong dự án Cypress theo cách mà bạn muốn. Điều này giúp tạo ra một cấu trúc dự án phù hợp với quy trình phát triển của bạn.

Sử dụng các plugin mở rộng: Cypress hỗ trợ sử dụng các plugin mở rộng để mở rộng tính năng của nó. Có thể thêm các plugins bằng cách cài đặt chúng thông qua npm và thêm cấu hình tương ứng vào tệp cypress.json.

#### Viết và quản lý kiểm thử với Cypress

1. Quy Trình Kiểm Thử với Cypress

* Xác Định Mục Tiêu Kiểm Thử: Trước khi bắt đầu viết bất kỳ bài kiểm thử nào, điều quan trọng là xác định rõ mục tiêu của chúng. Mục tiêu này có thể là kiểm tra một tính năng cụ thể, xác nhận hành vi của ứng dụng trong một trường hợp cụ thể, hoặc đảm bảo tính nhất quán của giao diện người dùng.
* Viết Ca Kiểm Thử: Sử dụng Cypress Test Runner để viết các ca kiểm thử. Mỗi ca kiểm thử nên tập trung vào một tính năng hoặc chức năng cụ thể và kiểm tra một trường hợp cụ thể của tính năng đó. Sử dụng các phương thức của Cypress như cy.visit, cy.get, cy.contains để tương tác với các phần tử trên giao diện người dùng và kiểm tra chúng.
* Sử Dụng Assertions: Sử dụng các assertions để kiểm tra kết quả của các hoạt động trong bài kiểm thử. Cypress cung cấp một loạt các assertions tích hợp như cy.should, cy.expect để kiểm tra các điều kiện mong đợi và xác nhận hành vi của ứng dụng.
* Sử Dụng Fixtures: Fixtures là các tập dữ liệu mẫu được sử dụng trong bài kiểm thử để tạo điều kiện ban đầu hoặc dữ liệu đầu vào cho các kiểm thử. Sử dụng fixtures để đọc dữ liệu từ các tệp JSON hoặc CSV và sử dụng chúng trong bài kiểm thử của bạn.
* Sử Dụng Custom Commands: Cypress cho phép bạn tạo ra các lệnh tùy chỉnh để tái sử dụng trong các bài kiểm thử. Điều này giúp giảm sự lặp lại mã và tăng tính module của mã của bạn. Tùy chỉnh các lệnh cho phép bạn viết mã kiểm thử một cách dễ đọc hơn và dễ bảo trì hơn.
* Sử Dụng Hooks: Cypress hỗ trợ các hooks như before, beforeEach, after, và afterEach để thiết lập trạng thái ban đầu, làm sạch dữ liệu, hoặc thực hiện các tác vụ khác trước và sau mỗi bài kiểm thử. Sử dụng hooks để tự động hóa các bước chuẩn bị và làm sạch trong quá trình kiểm thử.

1. Quản Lý Kiểm Thử

Tổ Chức Thư Mục: Sử dụng cấu trúc thư mục hợp lý để tổ chức các tệp kiểm thử theo tính năng, chức năng hoặc loại kiểm thử. Tổ chức thư mục giúp dễ dàng tìm và hiểu các kiểm thử trong dự án của mình.

Chia Nhỏ Bài Kiểm Thử: Chia nhỏ các bài kiểm thử thành các ca kiểm thử nhỏ hơn để dễ bảo trì và hiểu rõ hơn. Các ca kiểm thử nhỏ hơn giúp tăng tính module của mã kiểm thử và giảm nguy cơ lỗi.

Sử Dụng Nhóm Kiểm Thử: Sử dụng nhóm kiểm thử để nhóm các kiểm thử liên quan hoặc có chung mục tiêu. Nhóm kiểm thử giúp dễ dàng quản lý và thực thi các kiểm thử có liên quan một cách hiệu quả.

Bảo Trì và Cập Nhật: Theo dõi các thay đổi trong ứng dụng và cập nhật các bài kiểm thử tương ứng để đảm bảo rằng chúng vẫn hoạt động đúng và phản ánh chính xác các tính năng mới hoặc thay đổi. Đảm bảo rằng bạn thực hiện kiểm tra định kỳ và cập nhật các kiểm thử theo nhu cầu.

#### Debugging và Quản Lý Lỗi trong Cypress

1. Debugging với Cypress

cy.log và console.log: Cypress cung cấp cy.log() để ghi thông điệp log vào kết quả kiểm thử. Ngoài ra, có thể sử dụng console.log() để ghi log vào console trong quá trình kiểm thử. Điều này giúp theo dõi và hiểu rõ hơn về quá trình thực thi kiểm thử.

Snapshot và Time Travel: Cypress cho phép xem các snapshot của ứng dụng trong quá trình kiểm thử, giúp dễ dàng phát hiện lỗi và hiểu rõ hơn về tình trạng của ứng dụng tại mỗi bước của kiểm thử. Ngoài ra, tính năng Time Travel của Cypress cho phép quay lại và xem lại các thao tác kiểm thử trước đó để phân tích và debug.

Breakpoints và Debugging Trong DevTools: Cypress tích hợp sẵn với DevTools của trình duyệt, cho phép dễ dàng debug và phân tích các lỗi trong kiểm thử của mình bằng cách sử dụng các công cụ mạnh mẽ của DevTools. Có thể sử dụng breakpoints để dừng thực thi của kiểm thử tại các điểm nhất định và tiến hành debug từ đó.

1. Quản Lý Lỗi trong Cypress

Xử Lý Lỗi: Khi một lỗi xảy ra trong quá trình kiểm thử, Cypress cung cấp thông báo chi tiết về lỗi, bao gồm vị trí chính xác của lỗi và thông tin về môi trường kiểm thử. Bạn có thể sử dụng thông báo lỗi này để nhanh chóng xác định và giải quyết vấn đề.

Tự Động Thử Lại (Retry): Cypress có khả năng tự động thử lại các bài kiểm thử khi chúng gặp lỗi, giúp giảm thiểu các lỗi do flaky tests (kiểm thử không đáng tin cậy) và đảm bảo sự ổn định của quá trình kiểm thử. Có thể cấu hình số lần thử lại và khoảng thời gian giữa các lần thử lại để tối ưu hóa quá trình này.

#### Tích Hợp Cypress với Các Công Cụ Khác

1. Tích Hợp với Công Cụ CI/CD

Tích Hợp với Jenkins, GitLab CI/CD, CircleCI, vv.: Cypress có thể tích hợp dễ dàng với các công cụ CI/CD phổ biến như Jenkins, GitLab CI/CD, CircleCI, vv. để tự động chạy các bài kiểm thử trong quy trình CI/CD của bạn. Điều này giúp đảm bảo rằng các kiểm thử được thực hiện một cách liên tục và tự động.

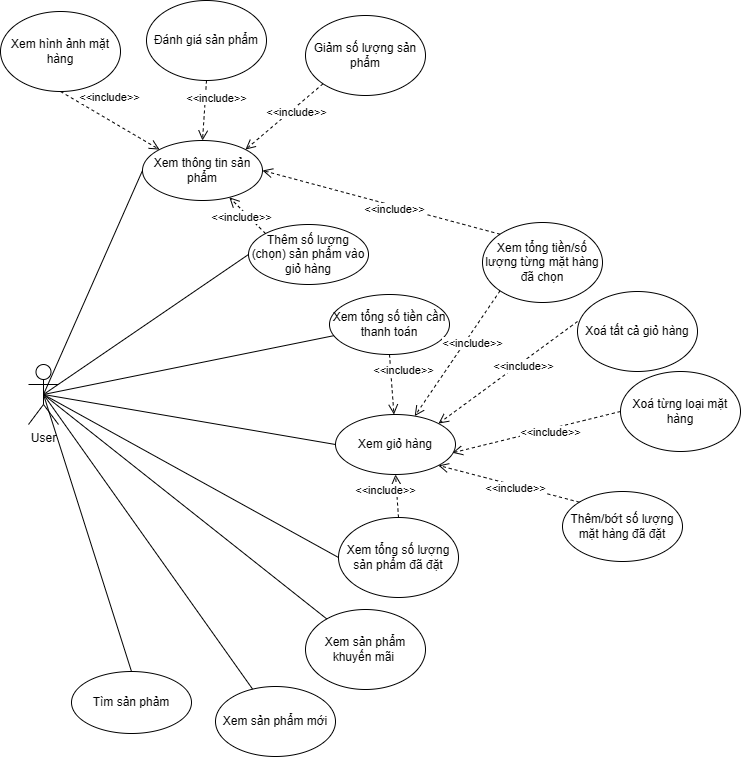
1. Sử Dụng Cypress Dashboard

Quản Lý và Báo Cáo Kiểm Thử: Cypress Dashboard cung cấp một giao diện trực quan để quản lý và theo dõi các kết quả kiểm thử của bạn. Bạn có thể xem các báo cáo chi tiết, thống kê và đánh giá hiệu suất của các bài kiểm thử của mình. Ngoài ra, bạn cũng có thể tạo và chia sẻ các báo cáo kiểm thử với đồng đội của mình.

Tích Hợp với Các Dịch Vụ Cloud: Cypress Dashboard cũng tích hợp với các dịch vụ cloud như GitHub, GitLab, Bitbucket để tự động kích hoạt và chạy các bài kiểm thử khi có các thay đổi trong mã nguồn của bạn. Điều này giúp bạn dễ dàng quản lý và theo dõi quá trình kiểm thử của mình trên các nền tảng phổ biến.

## Đặc tả use case các chức năng của trang web

### Usecase tổng quát



Hình 1 Usecase tổng quát

### Bảng đặc tả usecase chi tiết của trang web

Bảng 1 Bảng đặc tả Usecase Xem thông tin mặt hàng

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 1 | |
| Usecase Name | Xem thông tin mặt hàng | |
| Mô tả | Xem các thông tin của sản phẩm muốn xem | |
| Ngữ cảnh | Khi người dùng muốn biết những thông tin liên quan đến sản phẩm | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi xem thông tin. | |
| Điều kiện sau | Hiển thị đầy đủ các thông tin cần thiết cho người dùng | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút “+info” để xem những thông tin cần biết. | * Hiển thị các thông tin có sẵn lên giao diện. |
| Ngoại lệ | không | |

Bảng 2 Bảng đặc tả Usecase Chọn sản phẩm

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 2 | |
| Usecase Name | Chọn sản phẩm. | |
| Mô tả | Chọn các sản phẩm cần mua trong trang web | |
| Ngữ cảnh | Khi người dùng muốn chọn những sản phẩm mong muốn trên web | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi chọn sản phẩm. | |
| Điều kiện sau | Thêm đẩy đủ các sản phẩm người dùng chọn vào giỏ hàng | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút thêm sản phẩm vào giỏ hàng để mua những sản phẩm muốn mua. | * Hiển thị các thông tin có sẵn lên giao diện. * Cộng các sản phẩm khách hàng đã chọn vào giỏ hàng. * Tổng kết số lượng sản phẩm và số tiền cần thanh toán. |
| Ngoại lệ | không | |

Bảng 3 Bảng đặc tả Usecase Chỉnh sửa giỏ hàng

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 3 | |
| Usecase Name | Chỉnh sửa giỏ hàng | |
| Mô tả | Chỉnh sửa số lượng và loại mặt hàng đã có trong giỏ hàng | |
| Ngữ cảnh | Khi người dùng muốn chỉnh sửa số lượng và loại sản phẩm mình đã chọn và lưu trong giỏ hàng | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi chỉnh sửa. * Người dùng đã thêm mặt hàng vào giỏ hàng. | |
| Điều kiện sau | Đơn hàng được điều chỉnh thành công | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút giỏ hàng trên trang web để truy cập giỏ hàng. * Người dùng chỉnh sửa số lượng của các sản phẩm mình đã chọn bằng cách ấn các mũi tên lên/xuống tại vị trí số lượng mặt hàng. | * Ghi nhận hoạt động chuyển trang của người dùng và chuyển sang giao diện hiển thị giỏ hàng. * Hiển thị các thông tin có sẵn lên giao diện giỏ hàng. * Ghi nhận hoạt động chỉnh sửa của người dùng. * Tự động lưu lại số lượng của sản phẩm đã được người dùng điều chỉnh. |
| Ngoại lệ | Người dùng chưa thêm sản phẩm nào trong giỏ hàng | |

Bảng 4 Bảng đặc tả Usecase Xóa sản phẩm

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 4 | |
| Usecase Name | Xoá sản phẩm | |
| Mô tả | Xoá các sản phẩm trong giỏ hàng | |
| Ngữ cảnh | Khi người dùng muốn xoá các sản phẩm trong giỏ hàng. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện xoá sản phẩm. * Người dùng đã thêm mặt hàng vào giỏ hàng. | |
| Điều kiện sau | Sản phẩm được xóa thành công | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút giỏ hàng trên trang web để truy cập giỏ hàng. * Người dùng chọn phương thức xoá từng loại mặt hàng theo mong muốn hoặc chọn nút xoá tất cả các sản phẩm có trong giỏ hàng. | * Ghi nhận hoạt động chuyển trang của người dùng và chuyển sang giao diện hiển thị giỏ hàng. * Hiển thị các thông tin có sẵn lên giao diện giỏ hàng. * Ghi nhận hoạt động chỉnh sửa của người dùng. * Tự động lưu lại thông tin giỏ hàng sau khi người dùng thực hiện thao tác chỉnh sửa. |
| Ngoại lệ | Người dùng chưa thêm sản phẩm nào trong giỏ hàng | |

Bảng 5 Bảng đặc tả Usecase Xem giỏ hàng

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 5 | |
| Usecase Name | Xem giỏ hàng. | |
| Mô tả | Xem các thông tin có trong giỏ hàng | |
| Ngữ cảnh | Khi người dùng muốn xem lại những thông tin liên quan đến mặt hàng mình đã chọn trong giỏ hàng | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện xem giỏ hàng. * Người dùng đã thêm sản phẩm vào giỏ hàng. | |
| Điều kiện sau | Các thông tin trong giỏ hàng được thể hiện rõ ràng, chi tiết. | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút giỏ hàng trên trang web để truy cập giỏ hàng. | * Ghi nhận hoạt động chuyển trang của người dùng và chuyển sang giao diện hiển thị giỏ hàng. * Hiển thị các thông tin đã được lưu trong giỏ hàng lên giao diện. |
| Ngoại lệ | Không | |

Bảng 6 Bảng đặc tả Usecase Xem các sản phẩm mới

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 6 | |
| Usecase Name | Xem các sản phẩm mới. | |
| Mô tả | Xem các sản phẩm mới trên trang web | |
| Ngữ cảnh | Khi người dùng muốn xem các mặt hàng mới được thêm trên web. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện xem sản phẩm. | |
| Điều kiện sau | Thông tin các sản phẩm mới được thể hiện rõ ràng | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút “Productors Neuvos” để chuyển sang giao diện hiển thị những sản phẩm mới * Người dùng chọn nút “+info” để biết thêm thông tin cụ thể của các sản phẩm mới trên web. | * Hiển thị các thông tin có sẵn lên giao diện. * Ghi nhận hoạt động chuyển trang của người dùng và chuyển sang giao diện hiển thị các mặt hàng mới. |
| Ngoại lệ | Không | |

Bảng 7 Bảng đặc tả Usecase Xem sản phẩm khuyến mãi

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 7 | |
| Usecase Name | Xem các sản phẩm khuyến mãi. | |
| Mô tả | Xem các sản phẩm khuyến mãi trên trang web | |
| Ngữ cảnh | Khi người dùng muốn xem các mặt hàng được khuyến mãi trên web. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện xem sản phẩm. | |
| Điều kiện sau | Thông tin các sản phẩm khuyến mãi được thể hiện rõ ràng. | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút “Promociones” để chuyển sang giao diện hiển thị những sản phẩm mới. * Người dùng chọn nút “+info” để biết thêm thông tin cụ thể của các sản phẩm mới trên web. | * Hiển thị các thông tin có sẵn lên giao diện. * Ghi nhận hoạt động chuyển trang của người dùng và chuyển sang giao diện hiển thị các sản phẩm được khuyến mãi. |
| Ngoại lệ | Không | |

Bảng 8 Bảng đặc tả Usecase Đánh giá sản phẩm

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 8 | |
| Usecase Name | Đánh giá sản phẩm | |
| Mô tả | Đánh giá sản phẩm theo ý muốn. | |
| Ngữ cảnh | Khi người dùng muốn đánh giá các mặt hàng có trên web. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện đánh giá sản phẩm. | |
| Điều kiện sau | Các đánh giá sản phẩm của người dùng được lưu lại | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút “+info” để xem thông tin chi tiết của sản phẩm muốn đánh giá * Người dùng chọn phương thức thích hoặc không thích sản phẩm trong thông tin để đánh giá sản phẩm. | * Hiển thị các thông tin có sẵn lên giao diện. * Hiển thị thông tin chi tiết sản phẩm * Tự động lưu lại các đánh giá của người dùng về sản phẩm. |
| Ngoại lệ | Người dùng chưa thêm sản phẩm nào trong giỏ hàng | |

Bảng 9 Bảng đặc tả Usecase Điều chỉnh số lượng sản phẩm muốn chọn

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 9 | |
| Usecase Name | Điều chỉnh số lượng mặt hàng muốn chọn. | |
| Mô tả | Trực tiếp lựa chọn số lượng mặt hàng muốn thêm vào giỏ hàng. | |
| Ngữ cảnh | Khi người dùng muốn trực tiếp tùy chỉnh số lượng mặt hàng cần mua để thêm vào giỏ hàng. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện mua sản phẩm. | |
| Điều kiện sau | Các sản phẩm được thêm vào giỏ hàng theo số lượng khách hàng mong muốn. | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng chọn nút “+info” để đến giao diện thể hiện chi tiết thông tin mặt hàng cần chọn. * Người dùng chọn phương thức thêm hoặc giảm số lượng mặt hàng mình muốn chọn. | * Hiển thị các thông tin có sẵn lên giao diện. * Hiển thị các thông tin chi tiết sản phẩm người dùng vừa chọn. * Tự động lưu lại số lượng sản phẩm người dùng đã chọn và tổng kết số lượng, giá tiền. |
| Ngoại lệ | Không. | |

Bảng 10 Bảng đặc tả Usecase Tìm kiếm sản phẩm

|  |  |  |
| --- | --- | --- |
| Usecase ID | Usecase 10 | |
| Usecase Name | Tìm kiếm sản phẩm. | |
| Mô tả | Tìm kiếm sản phẩm muốn chọn | |
| Ngữ cảnh | Khi người dùng muốn tìm kiếm những mặt hàng muốn mua hoặc thêm vào giỏ hàng. | |
| Điều kiện tiên quyết | * Những mặt hàng phải có sẵn trên trang web. * Thiết bị của người dùng có kết nối internet khi thực hiện tìm kiếm sản phẩm. | |
| Điều kiện sau | Người dùng tìm được những sản phẩm mong muốn | |
| Tác nhân | Khách hàng | |
| Luồng hoạt động | Tác nhân | Hệ thống |
| * Người dùng truy cập vào trang web bán đồ ăn. * Người dùng nhập tên sản phẩm cần tìm trên thanh tìm kiếm và * bắt đầu tìm kiếm sản phẩm. | * Hiển thị các thông tin có sẵn lên giao diện. * Hiển thị các sản phẩm có thông giống với thông tin người dùng vừa tìm kiếm. |
| Ngoại lệ | Không có mặt hàng người đang muốn tìm trên trang web | |

## Test plan

* Mục đích:
* Tìm hiểu các tính năng, cơ chế hoạt động của một trang web SPA
* Hiểu được các vận hành một hệ thống sử dụng các module Webpack
* Học và áp dụng kiểm thử e2e bằng công cụ Cypress, từ đó hiểu được hoạt động chi tiết của CI/CD
* Phạm vi:

Tất cả chức năng của trang web SPA đặt đồ ăn bằng cypress

* Môi trường kiểm thử:

Hệ điều hành: windows 11

Trình duyệt web: Electron v118

Cơ sở dữ liệu: src/utils/data.js (data, không thể truy cập được khi dùng global host), Session storage (cartContent, cartTotals), tất cả đều lưu dưới dạng JSON, trong đó:

* data {id, name, brand, likes, dislikes, priceInitial, newRelease, promotions, discount, price, photo1, photo2, photo3, description}
* cartContent {id, image, name, brand, price, counter, total, photo1}
* cartTotals {counter, total}

⇒ Tất cả đều trở về trạng thái ban đầu khi tắt trình duyệt

Công cụ kiểm thử: cypress (13.9.0), sử dụng reporter cypress-mochawesome-reporter (3.8.2) để xuất kết quả thành file html (cypress/results/result.html)

Các công cụ hỗ trợ: trình biên dịch Node js (21.7.1), trình quản lý gói yarn (1.12.19)

Môi trường vận hành web:

* Local: [http://localhost:8080](http://localhost:8080/) (chạy lệnh yarn install, yarn start để khởi động)
* Global: <https://final-swat-2324.netlify.app> (đã deploy lên Netlify)
* Các chức năng và phi chức năng:
* Chức năng: hiện tất cả mặt hàng, hiện mặt hàng mới, hiện mặt hàng có khuyến mãi, tìm kiếm mặt hàng theo tên, mua hàng, thay đổi thông tin giỏ hàng, xoá giỏ hàng
* Phi chức năng: kiểm các các thông tin hiển thị trên giao diện, các thông báo và lỗi trả về, các thao tác nhấp vào các phần tử
* Ước lượng nguồn lực và thời gian cần cho việc kiểm thử: nhóm 3 người, thời gian từ 18/04/2024 đến 18/05/2024
* Thành lập nhóm: Nguyễn Anh Tú, Lê Hoàng Khang, Ao Thuỵ Ngọc Trân
* Phân công:
* Nguyễn Anh Tú: Tìm hiểu về SPA, kiểm thử các phi chức năng, các chức năng tìm kiếm, xem thông tin chi tiết sản phẩm, đánh giá sản phẩm
* Lê Hoàng Khang: Tìm hiểu về cách vận hành webpack module, deploy lên global host, lập test plan, kiểm thử các chức năng còn lại
* Ao Thuỵ Ngọc Trân: Tìm hiểu cypress, webpack, review các chức năng, đặc tả use case, hoàn thiện báo cáo
* Rủi ro cần quản lý: thời gian làm bài tập bị giảm đi ra khá nhiều do cả nhóm đều học khá nhiều môn cho học kỳ 2 2023-2024

## Các trường hợp kiểm thử

Kiểm tra trang chủ, bố cục sắp xếp danh sách sản phẩm, các nút chuyển trang

Xem chi tiết các sản phẩm

Tìm kiếm sản phẩm theo tên

Đánh giá sản phẩm

Chọn hàng (thêm vào giỏ hàng)

Kiểm tra giỏ hàng

Cập nhật giỏ hàng (chỉnh sửa số lượng, xoá một sản phẩm, xoá toàn bộ giỏ hàng)

# BÁO CÁO KIỂM THỬ

**Tiêu đề:** Báo cáo kiểm thử các ứng dụng web SPA Mua đồ ăn, thức uống bằng cypress

## Thông tin dự án

* Tên dự án: Ứng dụng web SPA Mua đồ ăn, thức uống
* Thời gian: từ 18/04/2024 đến 18/05/2024
* Thành viên: Nguyễn Anh Tú, Lê Hoàng Khang, Ao Thuỵ Ngọc Trân
* Mục tiêu: Đánh giá được kết quả hoạt động các chức năng của trang web SPA Mua đồ ăn, thức uống
* Phạm vi kiểm thử: Tất cả các chức năng và phi chức năng của của trang web SPA Mua đồ ăn, thức uống

## Các thông số kiểm thử

### Các test case

#### TC1: Kiểm tra trang chủ/store, các button dẫn đến các trang trong header

Step 1: Kiểm tra trang chủ khi mới mở lên có đủ các phần header, main gồm có các section, có footer, trong đó không có modal nào tự động hiện lên

Step 2: Kiểm tra 4 button denStore, Productos Nuevos, Promociones, Carrito có tồn tại và đang hiển thị trên web, đồng thời 4 button này mang các text đại diện denStore, Productos Nuevos, Promociones, Carrito

Step 3: Click điều hướng 4 button denStore, Productos Nuevos, Promociones, Carrito, đảm bảo điều hướng đúng url có các path /#/inicio, /#/nuevosproductos, /#/promociones, /#/cart (baseUrl: [https://final-swat-2324.netlify.app](https://final-swat-2324.netlify.app/)), đồng thời cả 4 trang đều có đủ 3 phần header, main, footer, không có modal xuất hiện, và title đều là “Den Store”

Step 4: Click điều hướng 3 button denStore, Productos Nuevos, Promociones, đảm bảo điều hướng đúng nội dung có các thông tin category lần lượt là Catalogo, Nuevos, Promociones, cùng lúc đó đảm bảo cả 3 trang đều có thanh tìm kiếm

Step 5: Filter data trong cơ sở dữ liệu theo tiêu chí

* Tại trang Home/Store (denStore), tất cả data
* Tại trang new products (Productos Nuevos), data có newRelease=true
* Tại trang promotions (Promociones), data có promotions=true

Điều hướng đến các trang denStore, Productos Nuevos, Promociones, đảm bảo số lượng sản phẩm (số phần tử article) đúng bằng số lượng các data đã filter

#### TC2: Xem chi tiết các sản phẩm

Step 1: Tại trang Store, click vào button +info của 1 sản phẩm bất kỳ, đảm bảo modal được hiện ra, press phím Esc, đảm bảo modal biến mất

Step 2: Tại trang Store, click vào button +info của các sản phẩm để modal hiện ra, đảm bảo thông tin trên modal khớp với thông tin từ article truyền chính xác vào modal bao gồm: name, price, photo1

Step 3: Filter data trong cơ sở dữ liệu theo tiêu chí

* Tại trang Home/Store (denStore), tất cả data
* Tại trang new products (Productos Nuevos), data có newRelease=true
* Tại trang promotions (Promociones), data có promotions=true

Điều hướng đến các trang denStore, Productos Nuevos, Promociones. Đảm bảo đúng thông tin chi tiết mỗi sản phẩm về giá trị của id, name, brand, likes, dislikes, price, photo1, photo2, photo3, description

Step 4: Tại trang Store, chọn 1 sản phẩm, click vào button +info, click lần lượt vào các hình con của sản phẩm, đảm bảo ô hiển thị hình ảnh hiển thị chính xác với hình ảnh con đã click (verify thuộc tính src của ảnh)

#### TC3: Tìm kiếm sản phẩm theo tên

Step 1: Tại trang Store (vì muốn đầy đủ dữ liệu), tìm kiếm sản phẩm với kết quả không tồn tại bằng cách nhập vào thanh tìm kiếm và bấm phím Enter với lần lượt các text: “fsf”, “35253fas”, “jg”. Đảm bảo section chỉ hiện ra nội dung “Sin Resultados” với background color là rgba(255, 0, 0, 0.3)

Step 2: Tại trang Store, tìm kiếm sản phẩm với kết quả tồn tại ít nhất 1 sản phẩm với các text lần lượt là: “m”, “a”, “e”, filter các text tìm kiếm này vào gốc để lấy các data kiểm tra. Đảm bảo số lượng item tìm được >=1, các item tìm kiếm được khớp với với data kiểm tra

Step 3: Tại trang Store, tìm kiếm sản phẩm với kết quả yêu cầu tồn tại đúng 1 sản phẩm với các text hiển thị giống như trên web hiển thị lần lượt là: “Cafe”, “Refresco”, “Pizza”

Step 4: Tại trang Store, tìm kiếm sản phẩm với kết quả yêu cầu tồn tại đúng 1 sản phẩm với các text đã viết thường lần lượt là: “cafe”, “refresco”, “pizza”

Step 5: Filter data trong cơ sở dữ liệu theo tiêu chí

* Tại trang Home/Store (denStore), tất cả data
* Tại trang new products (Productos Nuevos), data có newRelease=true
* Tại trang promotions (Promociones), data có promotions=true

Điều hướng đến các trang denStore, Productos Nuevos, Promociones. Tìm kiếm với text bất kỳ rồi xoá đi, đảm bảo nội dung trả về có số lượng đúng như khi mới mở trang (ví dụ promotions là 3 sản phẩm, tìm kiếm trả về 1 sản phẩm, xoá tìm kiếm thì trở về lại 3 sản phẩm)

#### TC4: Đánh giá sản phẩm

Step 1: Tại trang Store, click vào button +info của các sản phẩm Cafe (sản phẩm đầu tiên), ghi lại các lượt likes và dislikes ban đầu, click button like 3 lần và button dislike 3 lần. Verify dữ liệu theo các bước:

* Các lượt đánh giá hiển thị khớp với số lượt kỳ vọng (like+3, dislike+3)
* Điều hướng tới trang new products (Productos Nuevos), mở modal chi tiết của Cafe lên và đảm bảo đúng với đánh giá đã lưu ở trang Store
* Tiếp tục điều hướng về trang Store và kiểm tra các lượt đánh giá của Cafe

Step 2: Thực hiện các thao tác ở step 1 và verify với like 4 lần và dislike 1 lần cho mặt hàng Hamburguesa

Step 3: Thực hiện các thao tác ở step 1 với lượt like và dislike lần lượt là giá trị random trong khoảng từ 1 đến 10 cho mặt hàng Rosquilla

#### TC5: Chọn hàng (thêm vào giỏ hàng)

Step 1: Tại trang Store (vì đầy đủ sản phẩm), thực hiện các thao tác:

* Click duy nhất 1 lần button add của cả 6 sản phẩm
* Ghi lại các counter=số lần click, price=giá sản phẩm, tính total từng sản phẩm
* Tính tổng counter, tổng total của tất cả sản phẩm đã click.

Đảm bảo tổng counter, tổng total tại nút giỏ hàng hiện đúng tổng counter, tổng total đã tính được

Step 2: Lặp lại các thao tác thêm giỏ hàng của step 1, và đảm bảo các tiêu chí sau:

* cartContent có đủ 6 sản phẩm, mỗi sản phẩm đảm bảo đúng id, name, brand, photo1, price, counter, total đúng với các thông tin (id, name, brand, photo1, price, counter, total) đã ghi lại
* cartTotals có counter, total đúng với tổng counter, tổng total đã tính được

Step 3: Lặp lại step 2 nhưng với số lần click button add của mỗi sản phẩm là random trong khoảng 1 tới 10

Step 4: Tại trang Store, thêm tất cả sản phẩm vào giỏ hàng, mỗi mặt hàng thực hiện theo các bước:

* Click vào button +info mở modal chi tiết
* Click button + 3 lần, button – 2 lần, ghi lại các thông tin id, name, brand, photo1, price, counter, total, tổng counter, tổng total

Đảm bảo thông tin đúng theo các tiêu chí:

* Kiểm tra cartContent và cartTotals như step 2
* Kiểm tra tổng counter, tổng total tại nút giỏ hàng hiện đúng tổng counter, tổng total đã tính được

Step 5: Thay đổi số lần click của button + là một số random trong khoảng từ 1 tới 10, và button – là random khoảng 0 tới số lần click của button + trừ đi 1. Lặp lại việc ghi thông tin và kiểm tra của step 4

#### TC6: Kiểm tra giỏ hàng

Step 1: Tại trang Store, thêm tất cả sản phẩm vào giỏ hàng, mỗi mặt hàng thực hiện theo các bước:

* Click vào button +info mở modal chi tiết
* Chọn số lần click của button + là một số random trong khoảng từ 1 tới 10, và button – là random khoảng 0 tới số lần click của button + trừ đi 1
* Click button + và button – theo số lần đã tạo, ghi lại các thông tin id, name, brand, photo1, price, counter, total, tổng counter, tổng total

Đảm bảo thông tin đúng theo các tiêu chí:

* Kiểm tra cartContent và cartTotals đúng với thông tin đã ghi lại
* Kiểm tra tổng counter, tổng total tại nút giỏ hàng hiện đúng tổng counter, tổng total đã tính được
* Điều hướng đến trang Cart, đảm bảo các thông tin hiển thị đúng như các thông tin đã lưu bao gồm: id (lấy tại button -), photo1, name, brand, price, counter, total, tổng counter, tổng total

Step 2: Lặp lại step 1 nhưng thay đổi số lần click button + là 5, số lần click button – là 5

Step 3: Lặp lại step 1 nhưng thay đổi số lần click button + là 2, số lần click button – là 5

#### TC7: Cập nhật giỏ hàng (chỉnh sửa số lượng, xoá một sản phẩm, xoá toàn bộ giỏ hàng)

Step 1: Tại trang Store, thêm tất cả sản phẩm vào giỏ hàng, mỗi mặt hàng được thêm bằng cách click button add với số lần click là một số random trong khoảng 1 tới 10, sau khi click hoàn tất thì ghi lại các thông tin id, name, brand, photo1, price, counter, total của từng mặt hàng, và tổng counter, tổng total. Điều hướng đến trang Cart, thay đổi số lượng Hamburguesa thành 3, Helado 2 và ghi lại các thông tin mới, đảm bảo các thông tin đã thay đổi trên giao diện và trong cartContent, cartTotals đúng với tính toán

Step 2: Lặp lại việc thêm giỏ hàng ở step 1, thay đổi số lượng và ghi lại thông tin ở step 1, kiểm tra khớp thông tin ở trang Cart. Điều hướng về trang Store, click vào button +info mở modal của các mặt hàng Hamburguesa, Helado, đảm bảo số lượng hiển thị và tổng tiền đã thay đổi khớp với trên modal chi tiết

Step 3: Lặp lại thêm giỏ hàng ở step 1 rồi điều hướng đến trang Cart, click button – của Pizza và Cafe để xoá 2 mặt hàng này, kiểm tra và đảm bảo 2 mặt hàng đã bị xoá trên cartContent, cartTotals, button Cart và modal hiển thị chi tiết của 2 mặt hàng ở trang Store có counter và total là 0

Step 4: Thực hiện các thao tác thêm giỏ hàng ở step 1. Điều hướng đến trang Cart, kiểm tra so khớp thông tin rồi click nút Empty cart, kiểm tra cartContent rỗng, cartTotals có counter và total đều bằng 0, tổng counter và tổng total ở button Cart cũng như ở cuối bảng item cũng bằng 0. Điều hướng về trang Store và kiểm tra tất cả mặt hàng đều có counter, và total bằng 0

### Các bước vận hành kiểm thử Cypress

Step 1: Nếu chưa có node js và yarn thì cài đặt node js và yarn

- Node JS: https://nodejs.org/en (phiên bản đang sử dung 21.7.1)

- Yarn: https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable (phiên bản đang sử dung 1.22.19)

Step 2: Mở thư mục source, mở CMD, chạy lệnh "cd SPA-ecommerce" để vào souce code của web

Step 3: Chạy lệnh "yarn install" để cài các package và lệnh "yarn start" để chạy web

(Step 3 sẽ được bỏ qua vì đã deploy lên host Netlify. Link: https://final-swat-2324.netlify.app/)

Step 4: Chạy lệnh "cd ../Cypress" để điều hướng đến thư mục Cypress

Step 5: Nếu muốn chạy thông qua Test Runner thì chạy lệnh "yarn run cy:open", sau khi cửa sổ bật lên thì bấm vào "E2E Testing", chọn trình duyệt "Chrome", khi này sẽ hiện ra các spec, mỗi spec là 1 file \*.cy.js để viết các test case, chọn từng file spec để kiểm tra kết quả

Step 6: Nếu muốn xuất kết quả thì quay lại CMD, chạy lệnh "yarn run cy:run" để chạy chế độ headless, khi chạy xong thư mục Cypress sẽ xuất hiện file result/result.html chứa báo cáo kết quả, và thư mục screenshots (nếu có fail) chứa các ảnh chụp màn hình các test fail

### Các results và các defects

#### TC1: Kiểm tra trang chủ/store, các button dẫn đến các trang trong header

* Result 1: Trang chủ/Store có đầy đủ các phần header, main, section, footer, không có modal nào hiện lên
* Result 2: 4 button denStore, Productos Nuevos, Promociones, Carrito có tồn tại và đang hiển thị trên web, và mang lần lượt các text đại diện là denStore, Productos Nuevos, Promociones, Carrito
* Result 3: 4 button denStore, Productos Nuevos, Promociones, Carrito dẫn đúng đến các trang có path lần lượt là /#/inicio, /#/nuevosproductos, /#/promociones, /#/cart (baseUrl: [https://final-swat-2324.netlify.app](https://final-swat-2324.netlify.app/)), cả 4 trang đều có đủ 3 phần header, main, footer, không có modal xuất hiện, và title đều là “Den Store”
* Result 4: 3 button denStore, Productos Nuevos, Promociones điều hướng tới các trang có nội dung category lần lượt là Catalogo, Nuevos, Promociones, cả 3 trang đều có thanh tìm kiếm
* Result 5: Các trang denStore, Productos Nuevos, Promociones đều hiển thị số lượng sản phẩm đúng với dữ liệu đã filter

#### TC2: Xem chi tiết các sản phẩm

* Result 6: Modal hiển thị khi click vào button +info của 1 sản phẩm, và biến mất khi press phím Esc
* Result 7: Thông tin name, price, photo1 được modal hiển thị từ article gửi lên đúng với thông tin hiển thị trên article chứa button +info được click
* Result 8: Các thông tin chi tiết: id, name, brand, likes, dislikes, price, photo1, photo2, photo3, description của các article item tại trang denStore, Productos Nuevos, Promociones hiển thị đúng với các data, data có newRelease=true, data có promotions=true
* Result 9: Trên modal, ô hình lớn hiển thị đúng các hình con đã click

#### TC3: Tìm kiếm sản phẩm theo tên

* Result 10: Khi tìm kiếm với tên không tồn tại: “fsf”, “35253fas”, “jg”, trang web hiển thị ra 1 div có background color là rgba(255, 0, 0, 0.3), mang nội dung “Sin Resultados”
* Result 11: Tìm kiếm với các text: “m”, “a”, “e”, các article item trả về có số lượng >= 1, các item này khớp và đúng thứ tự với data đã filter
* Defect 1 (Step 3): Tìm kiếm theo tên không khả dụng khi người dùng copy paste tên sản phẩm để tìm kiếm. Data có name là chữ viết thường, trong khi hiển thị trên web là sau khi áp dụng css capitalize. Data hệ thống chỉ áp dụng tìm kiếm với trường hợp match text
* Result 12: Trả về đúng 1 article item, item này có thông tin chi tiết khớp với data đã filter
* Defect 2 (Step 5): Trang Productos Nuevos ban đầu chỉ hiển thị các data newRelease=true (đúng), sau khi tìm kiếm và clear thanh tìm kiếm thì lại hiển thị toàn bộ sản phẩm

#### TC4: Đánh giá sản phẩm

* Result 13: Các lượt likes và dislikes mới đúng với số lần click + lượt likes/dislikes cũ, khi chuyển qua trang khác thì lượt likes mới và dislikes mới đã được lưu lại, lượt likes và dislikes mới vẫn được lưu khi chuyển về trang Store
* Result 14: Các lượt likes và dislikes mới của mặt hàng Hamburguesa được lưu lại sau khi đánh giá và chuyển trang
* Result 15: Các lượt likes và dislikes mới vẫn lưu chính xác với trường hợp random của mặt hàng Rosquilla

#### TC5: Chọn hàng

* Result 16: Sau khi chọn sản phẩm thì tại button Cart hiển thị chính xác tổng số lượng hàng và tổng số tiền
* Result 17: cartContent có đủ 6 sản phẩm, mỗi sản phẩm khớp các thông tin id, name, brand, photo1, price, counter, total thêm từ các article item gửi lên. Còn cartTotals lưu chính xác tổng số lượng hàng đã chọn và tổng số tiền
* Result 18: cartContent có đủ các sản phẩm đã thêm, và thông tin từng sản phẩm đều chính xác, cũng như thông tin trong cartTotals
* Result 19: Thông tin cartContent và cartTotals hiển thị chính xác với các sản phẩm đã thêm thông qua các button + và – trong modal chi tiết
* Result 20: Thông tin cartContent và cartTotals vẫn hiển thị chính xác trong trường hợp số lần click button + và – là random

#### TC6: Kiểm tra giỏ hàng

* Result 21: Thông tin trên cartContent, cartTotals, tổng số lượng hàng và tổng tiền trên button Cart được lưu chính xác, kể cả những thông tin trên trang Cart
* Result 22: Các thông tin cartContent, cartTotals, button Cart và trang Cart được lưu hiển thị đúng với dữ liệu của các article item được thêm vào
* Result 23: Các thông tin article item đều được thêm vào từ lúc click button + trên modal, khi này dù có click button – đến 0 thì thông tin các mặt hàng đều được lưu lại và có số lượng là 0

#### TC7: Cập nhật giỏ hàng

* Result 24: Thông tin về tổng counter, tổng total thay đổi đúng trên giao diện, các thông tin sau khi cập nhật ở cartContent và cartTotal đúng với các thông tin tính toán đã thay đổi
* Result 25: Các thông tin ở trang Cart đúng thông tin đã tính toán khi mới thêm vào và khi đã thay đổi, về trang Store, các thông tin counter và total của các mặt hàng đã thay đổi hiển thị đúng như thông tin trong giỏ hàng
* Result 26: Thông tin cartContent và cartTotals sau khi xoá các mặt hàng được lưu đúng với trên giao diện hiển thị và đúng với tính toán, các giá trị tổng counter, tổng totals trên button Cart và dưới bảng item hiển thị giống nhau. Các mặt hàng đã bị xoá khi xem chi tiết tại trang Store có counter và total bằng 0
* Result 27: Tất cả mặt hàng trên giỏ hàng đều bị xoá, tổng counter và tổng total trên button Cart và dưới bảng item đều bằng 0. Về trang Store, tất cả các mặt hàng đều có counter và total bằng 0

## Kết quả kiểm thử

* Các results:

Tổng số lượng testcases: 7

Tổng số lượng defects tìm được: 2

Testing coverage: 90%

* Các defects tìm được:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tên defect | Mô tả | Độ ưu tiên | Trạng thái |
| 1 | Chức năng tìm kiếm không hoạt động tốt khi người ta sử dụng text trên giao diện để tìm kiếm (Test case 3 Step 3) | Thông qua thuộc tính css capitalize, tên sản phẩm hiển thị trên giao diện có viết hoa chữ cái đầu tiên, người dùng muốn tìm kiếm chính xác sản phẩm bằng việc gõ/copy chính xác tên sản phẩm trên giao diện thì lại thông báo không tìm thấy. Trong khi vẫn là tên sản phẩm đó nhưng nhập toàn bộ là chữ thường thì tìm thấy | medium | accepted |
| 2 | Số sản phẩm được trả lại sau khi kết thúc tìm kiếm ở trang Productos Nuevos không đúng với ban đầu (Test case 3 Step 5) | Số sản phẩm ban đầu của Productos Nuevos là 3, đúng với cơ sở dữ liệu, sau khi các mặt hàng thay đổi thông qua tìm kiếm, xoá thanh tìm kiếm đi thì số sản phẩm hiển thị là 6 tương tự trang Store mà không phải là 3 như ban đầu | high | accepted |

* Đánh giá kiểm thử: trang web hoạt động khá trơn tru, tốc độ hiển thị nhanh, tuy ít defect nhưng các defects này khá nguy hiểm, gây ảnh hưởng trải nghiệm người dùng

## Các đề xuất cải tiến

* Bổ sung chức năng tìm kiếm theo brand
* Chỉnh sửa lại chức năng tìm kiếm theo tên, có thể xét trường hợp isCapitalized
* Chỉnh sửa lại việc hiển thị sản phẩm ở trang Productos Nuevos và trang Promociones đúng với ban đầu sau khi xoá thanh tìm kiếm
* Bổ sung tính năng check out
* Bổ sung việc phân trang cho trường hợp và sắp xếp các mặt hàng theo nhiều tiêu chí

## Tổng kết

* Tóm tắt kết quả kiểm thử:
* Nhìn chung web SPA này hoạt động tốt, gần như không có độ trễ khi load (trường hợp mạng tốt), còn thiếu tính năng quan trọng nhất đó là check out, một số tính năng có lỗi khá nặng là tìm kiếm và hiển thị sau tìm kiếm
* Việc cài đặt kiểm thử với Cypress mất khá nhiều thời gian từ config cho tới giải quyết các tình huống bất đồng bộ không mong muốn, tuy nhiên điểm cộng so với Robot Framework Selenium là có tích hợp Test Runner thuận tiện cho việc tự động chạy các bài kiểm thử sau khi lưu file, thuận tiện cho việc gỡ lỗi, giúp tốc độ cài đặt các test case nhanh hơn
* Chữ ký thành viên

Ao Thuỵ Ngọc Trân

# TÀI LIỆU THAM KHẢO

Tiếng Việt

Single Page Application Concept. (2018, March 25). Retrieved May 16, 2024, from https://viblo.asia/p/single-page-application-concept-LzD5dDvo5jY

Webpack là gì? Những điều bạn có thể chưa biết về Webpack. (n.d.). Retrieved May 18, 2024, from https://stringee.com/vi/blog/post/webpack-la-gi

Tiếng Anh

Comprehensive Cypress Test Automation Guide | Cypress Documentation. (2024, April 11). Retrieved May 16, 2024, from https://docs.cypress.io/guides/overview/why-cypress

Concepts. (n.d.). Retrieved May 18, 2024, from Webpack website: https://webpack.js.org/concepts/

Cypress Web Testing Framework: Getting Started. (n.d.). Retrieved May 16, 2024, from BrowserStack website: https://browserstack.wpengine.com/guide/cypress-framework-tutorial/

What is End To End Testing? (n.d.). Retrieved May 16, 2024, from BrowserStack website: https://browserstack.wpengine.com/guide/end-to-end-testing/